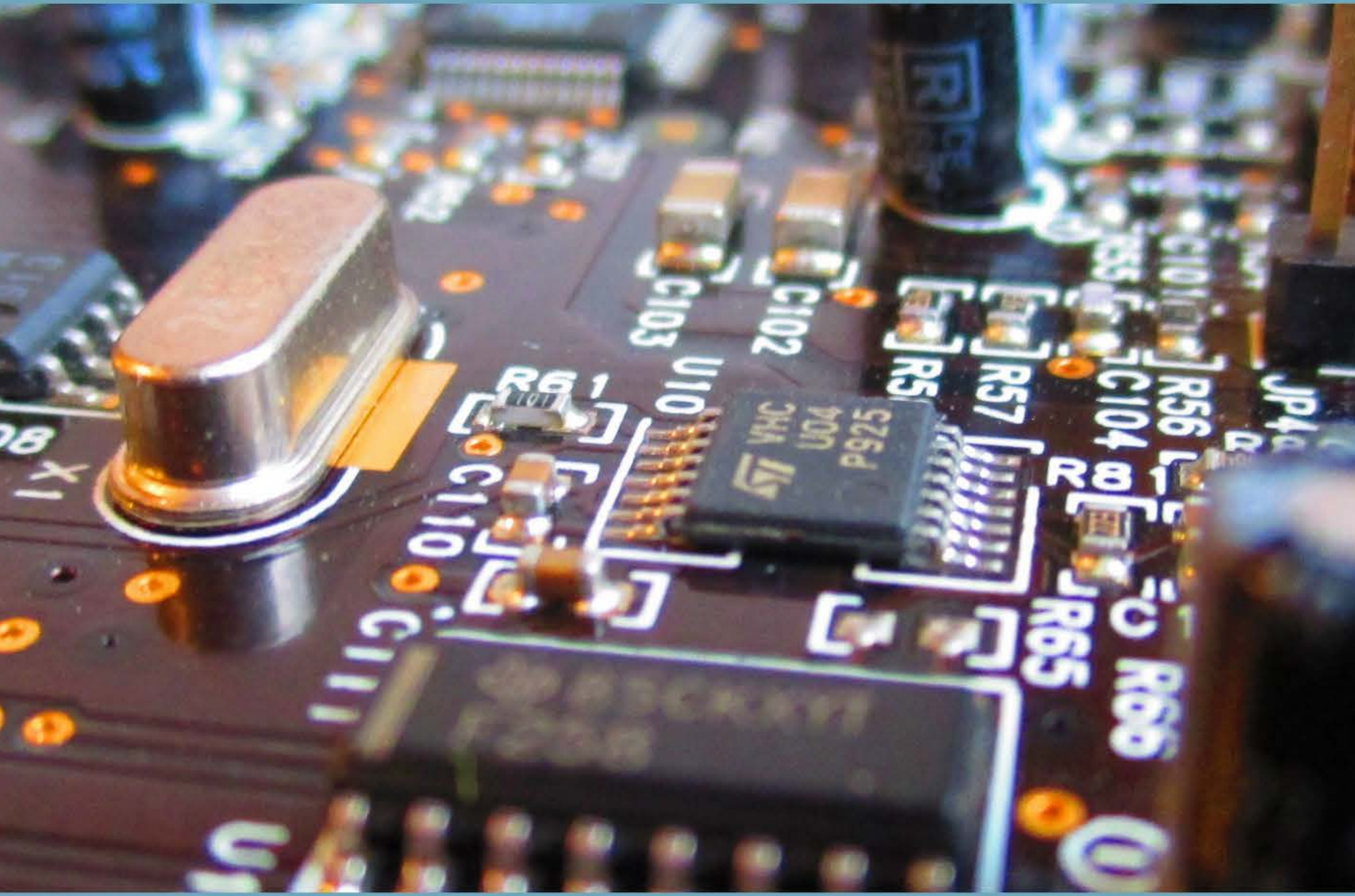
Programmable Smart Microcontroller Cards



EDITORS
Dr. Sabri KOCER
Dr. Ozgur DUNDAR
Resul BUTUNER

ISRES
Publishing

Programmable Smart Microcontroller Cards

EDITORS

Sabri KOCER
Ozgur DUNDAR
Resul BUTUNER

2021

ISRES
Publishing

Programmable Smart Microcontroller Cards

Editors

Sabri KOCER
Ozgur DUNDAR
Resul BUTUNER

Layout

Davut ALAN

This book was typeset in 10/12 pt. Times New Roman, Italic, Bold and Bold Italic.

Programmable Smart Microcontroller Cards

Published by ISRES Publishing, International Society for Research in Education and Science (ISRES).

Includes bibliographical references and index.

ISBN
978-605-74825-6-3

Date of Issue
December, 2021

E-mail
isresoffice@gmail.com

Address
Istanbul C. Cengaver S. No 2 Karatay/Konya/TURKEY

www.isres.org

PREFACE

Since the invention of the integrated circuit, the constant development of digital electronics has led to increasingly complex devices. These include microprocessors and microcontrollers, which are basic in electronic engineering, control systems careers.

Programmable smart microcontroller cards are a complete computer which is contained in the chip of an integrated circuit and is intended to govern a single task. The number of products that work on the basis of one or more microcontrollers increases exponentially. The computer industry accounts for a large part of the microcontrollers that are manufactured.

White-line appliances (washing machines, ovens, dishwashers, etc.) and brown-line appliances (televisions, videos, music devices, etc.) incorporate numerous microcontrollers. Likewise, the supervision, surveillance and alarm systems in buildings use these chips. They are also used to optimize the performance of elevators, heating, air conditioning, fire alarms, theft, etc. To test the (PLC) designed by means of a microcontroller, an electropneumatic circuit is designed that allows the number of inputs and outputs destined in the equipment to be put into use.

The applications of programmable smart microcontroller cards are vast, it can be said that they are only limited by the imagination of the user. It is common to find microcontrollers in fields such as robotics and automation, in the entertainment industry, in telecommunications, in instrumentation, in the automotive industry, etc.

Although the concept of controller has remained unchanged over time, its physical implementation has varied frequently. Three decades ago, electronic controllers were built exclusively with discrete logic components, later microprocessors were used, which were surrounded by memory chips and I/O on a printed circuit board. At present, all the elements of the controller have been able to be included in a single integrated circuit, which is called a microcontroller. It really consists of a simple but complete computer contained in an integrated circuit.

Today, smart electronic cards are used practically in almost every point that requires a control system.

In this book;

1. In the chapter, what is Arduino and what it does is briefly mentioned. Arduino is an electronic hardware and software based development platform designed to develop interactive projects. Arduino boards contain an Atmel AVR microcontroller and various electronic components for circuit connections. Arduino’s usage areas can be listed as robot programming, smart home, smart projects, automation, 3D printer. The reason why

it is used in these areas is that it is easily integrated and a started project can be revealed quickly.

2. In the chapter, information about BBC micro, one of the robotic coding tools, is given. BBC micro:bit is a small (4cm x 5cm) electronic board with a microcontroller on it. It was designed by the BBC for use for computer education (Ball at al., 2016). It is especially suitable for young children to do robotics and cod, it is an electronic circuit board for teaching basic cod logic in a block-based basis.

3. In the chapter, information about Raspberry Pi and its models is given. Raspberry Pi has been developed for young children all over the world to receive and learn to program. Raspberry Pi is used by everyone (children, educators, makers, software developers, etc.) due to its capacity to do most things a computer can do, its small size and affordable price.

4. In the chapter, the usage and advantages of PLCs today are explained. PLC (Programmable Logical Controller) is an automation device used in production departments in factories or in the control of processes such as the control of machines. PLC plays a major role in the prominence of factors such as producing more and high quality products of a short time, and producing with very low error rates. It is the biggest and most used application of PLCs, which is closest to relay systems with its "sequential operation" features. In terms of application, it is used in stand-alone machines or machine lines, conveyor and packaging machines, and even in modern elevator control systems.

5. In this chapter, the structure of STM32 Microcontrollers and their working logic is explained briefly. STM32 is a microcontroller using Arm Cortex-M processor produced by STMicroelectronics company. Basically, it consists of a microprocessor, ram memory, flash memory to save the program to run, and many interfaces depending on the model. By interfaces are meant programmable electronic structures.

6. In the chapter, the use of Nvidia Jetson in edge computing, artificial intelligence and image processing is mentioned. NVIDIA Jetso is the world's leading platform for advanced AI. The platform includes Jetson modules, which are small forms factor high-performance computers, the JetPack SDK to accelerate software, and an ecosystem of sensors, SDKs, services and products to accelerate development. Jetson is compatible with the same AI software and cloud-native workflows used in other NVIDIA platforms, delivering the performance and power efficiency customers need to build software-defined smart machines at the edge.

7. In the chapter, general information about LattePanda is given. In recent years, LattePanda has been increasing the use of Single Board Computers (SBC) as an embedded system, increasing its use in terms of ease of use, quick accessibility, low cost, increased R&D

studies, operating system on it, and popularizing its use academically and industrially. attePanda is used in many different project areas from robots to security systems, from system programming to gam. With the 4K support on the device, a wired or wireless keyboard or mouse can be connected and allows surfing the internet. In addition, since it has an Arduino board, it has the ability to do all the operations that can be done with this board.

8. In the chapter, the differences and advantages of ESP32 and ESP8266 are explained. ESP32 has been introduced to the market as an enhanced version of the ESP8266 module. Apart from the fact that the Wi-Fi connection is faster at the beginning, additional Bluetooth feature has been added to it. If we move on to the advantages of ESP8266, some of its libraries and features are better developed since it is an older module, apart from the fact that it will not be too complex and even take up less space for your projects, but it is lower in cost.

9. In the chapter, microcontroller architectures and the differences between them are explained and information about the PIC microcontroller is given. Harvard architecture is used in the architecture of PIC microcontrollers. In the Harvard architecture, the program and data storage memories are separate from each other. Although this means that they are a little more expensive than the Von Neumann architecture microcontrollers, which use the same structure to store programs and data, over time, developments in technology eliminate this price difference. Static RAM used in data memory is quite fast compared to flash memory used in program memory. Therefore, microcontrollers in the Harvard architecture are much faster. There are many uses.

10. In the chapter, ZigBee is mentioned. ZigBee is a customization of high-level communication protocols created using small, low-power digital radios according to an IEEE 802 standard used for personal area networks. Applications include short range wireless transfer of data at low rates to wireless light switches, in-home displays and electricity meters and other consumer and industrial equipment.

11. In the chapter, LilyPad Arduino board, which is especially designed for e-textile and wearable accessory projects, is mentioned. It is a very useful and versatile product that has a large audience in the field of sewn products. It was designed specifically for sewing on fabric by Leah Buechley and SparkFun. The board uses the ATmega168V or ATmega328V microcontroller.

12. In the chapter, the structure of the FPGA, the way it works and the programming methods are mentioned. FPGAs are semi-ready silicon devices that can be electrically programmed to be part of a digital circuit or system. Its structure can be defined in three main parts: programmable logic blocks, input-output blocks surrounding this block array, and interconnects. With FPGA, the functionality of basic logic gates and circuit elements

with more complex structure is increased. It is named field programmable because the logic blocks and interconnects can be programmed after the manufacturing process

13. In the chapter, the Deneyap Card is mentioned. This card is a camera and card set developed in Turkey and put on the market in 2020 in order to transfer the infrastructure of subjects related to the Internet of Things and electronic programming and to run certain artificial intelligence algorithms. “Deneyap Card and Camera” provides users with autonomous vehicles, smart cities, health, agriculture, energy, logistics, industry, etc. It is a product that provides the opportunity to develop projects in technological fields. It is mentioned that it was produced in 2020 with the cooperation of the Turkish Technology Team Foundation and its entrepreneurs.

Smart Electronic Cards for educational purposes; As a complete technology that brings all kinds of solutions to users from A to Z in general-purpose control and automation studies carried out in all branches of industrial applications, in aviation; It can be used in sensing and control systems, in the use of robotics and artificial intelligence, and as a source book in academia.

Best Regards,

Managing Editors

Sabri KOCER, PhD, He graduated from the Electrical Engineering Department of Selcuk University. He completed his graduate and his doctorate in Gazi University. Currently, Necmettin Erbakan University, Faculty of Engineering, Computer Engineering is working. Electronics, Computer, Telecommunication, Signal Processing and Biomedical studies in the area.

Email: skocer@erbakan.edu.tr , ORCID: 0000-0002-4849-747X

Ozgur DUNDAR, PhD, works at Necmettin Erbakan University, Department of Space and Satellite Engineering. He graduated from the Electrical and Electronics Engineering Department of Selcuk University. He worked as an Automation Engineer for a while. His master’s and doctorate degrees are from Selçuk University, Institute of Science and Technology, Department of Electrical and Electronics Engineering. Special fields of study are Automation, Robotic, Communication, Electromagnetic and Micro Strip Patch Antenna designs.

Email: ozdundar@erbakan.edu.tr , ORCID: 0000-0002-4142-4446

Resul BUTUNER, is a Computer Teacher at Adil Karaagac Vocational and Technical Anatolian High School in Konya, Turkey. He has a master’s degree in Computer Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, robotic coding, data mining and augmented reality applications. He is an instructor in the field of Robotic coding within TUBITAK. He continues to write a book in the field of robotic coding at the Ministry of National Education. He worked as a coordinator in budgeted projects related to student education.

E-mail: rbutuner@gmail. com, ORCID: 0000-0002-9778-2349

.

Arduino Microcontroller Card

Resul BUTUNER
Konya Adil Karaagac Vocational and Technical Anatolian High School

Yusuf UZUN
Necmettin Erbakan University

Introduction

Man continues to learn throughout his life. Education is immediately behind learning. Education strengthens its continuity by influencing people , those around them and the world they live in. Although individuals live with the knowledge they have acquired in the past , they continue to generate mental ideas of their future. Not just individuals Everything in the world is changing and evolving . For this reason , education continues to change and progress with new pursuits . Educators state that emerging new quests and approaches is related to 21st century skills (Gelen, 2017).

In the 21st century technology age, it has become very important to use the innovations required by the age in lessons in order to ensure the permanence of education. the most important of these innovations the concepts of robotics and robotics - coding are (Butuner, 2019).

Along with technological developments, the use of robots in daily life have increased day by day. In many areas, people renewal and develop themselves cognitively, socially and culturally with robotic applications. In many areas, people renew and have developed themselves cognitively, socially and culturally with robotic applications. However, the 21st century is called the age of technology. Many requirements have emerged in the name of education and training, and it has brought innovations with different structuring in education systems for meaningful and permanent learning. At the beginning of these innovations ; the name of which we hear a lot Concepts such as Coding and Robotic Coding , Arduino , STEM , Robot Kits are coming .

Today , in the studies of researchers in Turkey and abroad with integration of fields such as Science ,Technology , Engineering , Mathematics and Design thank to this , robotic applications have an important place . With the widespread use of STEM approach in education , robotic cod studies made it more common in education . At the beginning of these is the Arduino development board, which is most used in robotic coding. Due to its use of open source code and its low cost , it has become one of the most used cards in education . Therefore, robotic applications made with Arduino development card have taken their place in many academic studies.

Today , coding education means preparing software , websites and mobile applications through a code editor in any programming language .As learning and teaching strategy , it is shown as the use from easy about difficult and project based robots in programming education. (Akinci & Tuzun, 2016). In the use of robots in coding training, by loading certain tasks to the robot, the results are seen on the working system.

It is thought that arduino and similar cards for robotic coding will provide an important solution in concretizing abstract concepts and will help in physically observing the code (Ersoy et al., 2011).

Educational systems are tried to be developed by children learning coding-based robotic applications of an early age (Goksoy & Yılmaz, 2018).

In Turkey The information technologies and software lesson, which has been compulsory for the 5th and 6th grades since 2013, aims to enable students to start learning coding at an early age.(MEB, 2018).

In secondary school , the education program aimed for the student to acquire " Level 1 " and " Level 2 " robotic programming skills in applied computer lesson . (MEB, 2018).

Learning to code means by learned the working principles of existing technological devices and using them and producing for the better. By giving robotics and coding education to the students this is a very important step the technology developers of the future will be trained (Yildız et al., 2020).

In education, students need to be introduced science, mathematics, art, electronics, coding and mechanical skills.With Arduino and other electronic prototyping tools, it is easier and less costly to teach students science, mathematics, electronics, coding and mechanics.

Since the Arduino microcontroller board has a multidisciplinary structure, it is very useful for integrating it into education.

With Arduino training, students gain some skills. These;

- Analytical thinking ability,
- Design-oriented thinking,
- Teamwork,
- Giving and receiving feedback from peers,
- Problem solving with the help of project-based learning,

- Computational skills,

Thus, students can solve their own problems and become productive individuals rather than consuming. The Students, improve themselves on the subjects and applications they encounter at school with the robotic coding education given with Arduino microcontroller card and other robotic products.

These trainings enable them to achieve success in their future lives. The softwares aims to increase the motivation and interest of students who use and learn coding tools, rather than teaching coding directly. In the 21st century, coding software is ready for use of students from pre-school to university education today (Baz, 2018).

What is Arduino and Its History

Arduino, which is the most preferred for robotic studies, is a development board developed with 5 friends in Italy in 2005. It is a microcontroller board that started as university project. Name Arduino is named an from Italian medieval king. From this development card were produced Only 200 pieces for students. When the first 50 Arduino boards are bought by the students, five friends form a new company to sell the remaining 150. This company, which became famous for the name Arduino, has turned into one of the largest microcontroller companies in the world today.

Arduino’s; There are varieties such as Arduino Uno, Arduino Mega, Arduino Nano, Arduino Leonardo. Called Arduino Shield There are also parts that can be easily attached and removed from Arduino’s pins (Barman, 2014).

Arduino Cards and Features

Arduino Uno

Arduino has open source code. It has input and output units and microcontroller memory. The Arduino development card has been used in the development of robotic studies for many fields from education to engineering, making it popular. Arduino is a development board that has application developments in many different fields due to its wide library and continuous development. Due to the continuous development of the development card, varieties with different features have been developed according to different needs. At Figure 1 shows the Arduino development card and the description of the pins on it.

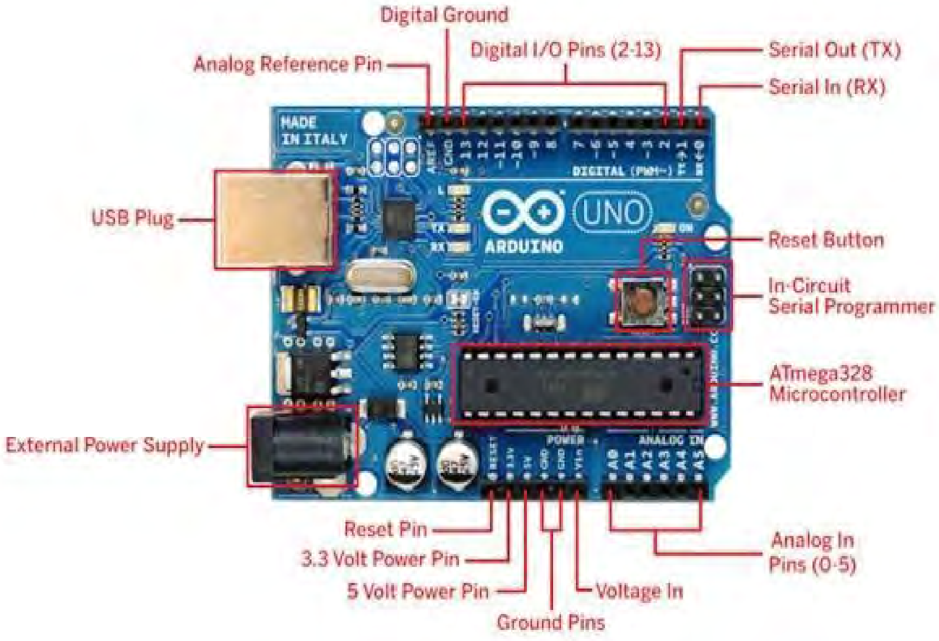


Figure 1. Arduino Development Card

There are 8 and 32 bit microcontrollers of Atmega company on Arduino card. for Arduino It provides the opportunity to easily program microcontrollers with ready-made libraries. Thanks to its analog and digital inputs, analog and digital data can be processed. Data from sensors can be used. According to the incoming data, sound, light, movement, etc. outputs can be produced. With the Arduino development board, many applications such as led circuits, automation systems, robotic projects, projects that interacts with the environment can be made. All Arduino models need a power supply. The Arduino microcontroller card provides its power via USB or from the power supply port, as in Figure 1. The pins on the Arduino development card are connected to the breadboard with the help of jumper cables. Each pin has different functions.

Pins and Indicators on Arduino Development Card

5V and 3.3V : Output voltages of 5 volts and 3.3 volts are taken from these pins.

GND: It is the abbreviation for ground. It is the grounding connection.

Analog pins: There are 6 analog pins including A0, A1, A2, A3, A4, A5. These pins can receive and give a total of 1024 numerical values between 0 and 1023. Figure 2 shows the line graph of the analog signal.

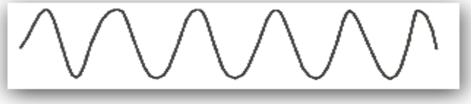


Figure 2. Analog Signal Line Graph

Digital pins: There are 14 digital pins between 0-13. These pins receive and gives high and low values. Front of them can be receivable Analog outputs from digital pins with a “~” sign in. These pins are called PWM pins. Figure 3 shows the digital signal line graph.

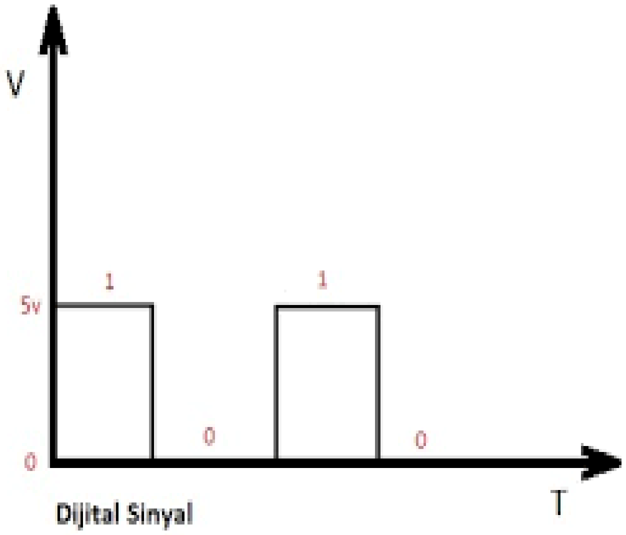


Figure 3. Digital Signal Line Graph

Aref pin: It is an analog reference pin. The Arduino regulator has 1023 steps. For a voltage of 3 volts, a voltage step with an accuracy of 3V/1023 = 0.00293V can be obtained.

Power indicator: This led shows whether the Arduino development board is working or not. There is a problem when this led is not lit even though there is a power connection. The circuit needs to be checked.

*TX and RX LEDs:*These led are blinking in the process of uploading software to the processor. It refers to serial communication. From the leds, TX means transmitter and RX means receiver.

0 and 1 pins: The letters with the TX and RX leds are seen on the 0 and 1 pins. These pins can be used for serial communication with the help of bluetooth module.

Processor: Atmega processor of ATMEL firm. Programs written are sent and run to this processor.

Regulator: It is used for protection from voltage values that may damage the circuit.

Arduino Mega 2560

Arduino Mega is a microcontroller card based on Atmega2560. Arduino Mega has 54 digital input/output pins. 15 of them can be used as PWM outputs. There are 16 analog inputs, 4 UARTs, 16Mhz crystal, usb socket, power socket, ICSP connector and reset button.

Figure 4. Arduino Mega 2560

Table 1 shows the features of Arduino Mega 2560. The operating voltage is 5 volts. The supply voltage is between 5-12 volts. It is recommended to using 7 to 12 volts for power supplies. Arduino Mega 2560 has 16 analog inputs and 10 bit resolution each. By default they operate in the range of 0-5V, but the AREF pin and the analogReference() function change the reference voltage range.

Table 1. Arduino Mega 2560 Features

Property name	Feature Description
Microcontroller	ATmega2560
Operating voltage	5V
Supply Voltage (Recommended)	7-12V
Supply Voltage (Limit)	6-20V
Digital I/O Pins	54 (14 of them PWM outputs)
Analog Input Pins	16
Current of I/O Pins	40 mA
3.3V Pin Current	50 mA
Flash Bellek	256 KB (It uses 8 kB of bootloader)
SRAM	8 KB
EEPROM	4 KB
Clock Frequency	16 MHz

Arduino Lilypad

Arduino Lilypad is Arduino development card. It was specially developed by Leah Buechley and SparkFun for sewing the of the card to fabric through the holes on the top. The microcontroller of Atmega168V or Atmega328V is used on the card. Arduino

Lilypad specifications are given in Table 2.

Table 2. Arduino Lilypad Features

Property name	Feature Description
Microcontroller	ATmega168V or ATmega328V
Operating voltage	2.7-5.5 V
Input voltage	2.7-5.5 V
Digital I/O Pins	14 (6 of them can be used as PWM output.)
Analog Input Pins	6
DC current of I/Os	40 mA
Flash Bellek	16 KB (2 KB of it is using bootloader.)
SRAM	1 KB
EEPROM	512 byte
Clock Frekansı	8 MHz

As seen in Figure 5, there are 6 analog inputs and 14 digital outputs on the Arduino Lilypad board. Arduino Lilypad card is mostly used in projects prepared for wearable technologies. The reason why it is called wearable technology is that Lilypad card is sewn with used on clothes, fabric and anywhere desired with conductive wire.

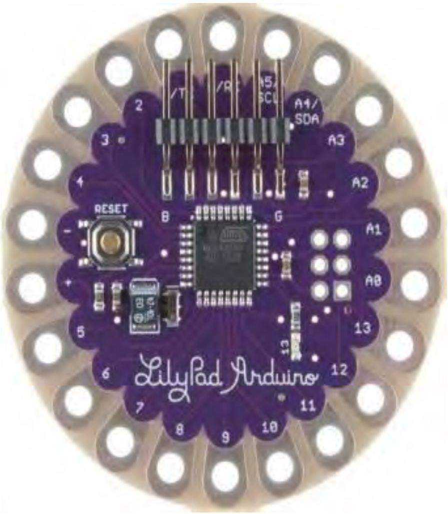


Figure 5. Arduino Lilypad

LilyPad has an ATmega328 processor on it. On the card Arduino bootloader is installed. Therefore, it is programmed as other Arduino boards are programmed. It can be coded from Arduino’s own program using an USB cable from A to B. These entrances the use of products that can be sewn to with conductive wire make things easier. The parts whose input pins can be connected to the first model of the same LilyPad are used in the same way. LilyPad is a circular development card. Its diameter is 50 mm and its thickness is 0.8 mm. One of the biggest features of the LilyPad is that it can come into contact with water if the power is disconnected.

Arduino Ethernet Shield

Ethernet module is used to connect Arduino to wired internet network. With the Ethernet module, Arduino can access websites such as web browsers and save data. There is Wiznet W5100 integrated into the card The Ethernet module is both TCP and UDP compatible. The advanced versions of the Ethernet module have the feature to use SD card. At Figure 6 shows the Arduino Ethernet module. In addition to the Arduino Uno, it has an internet connection.

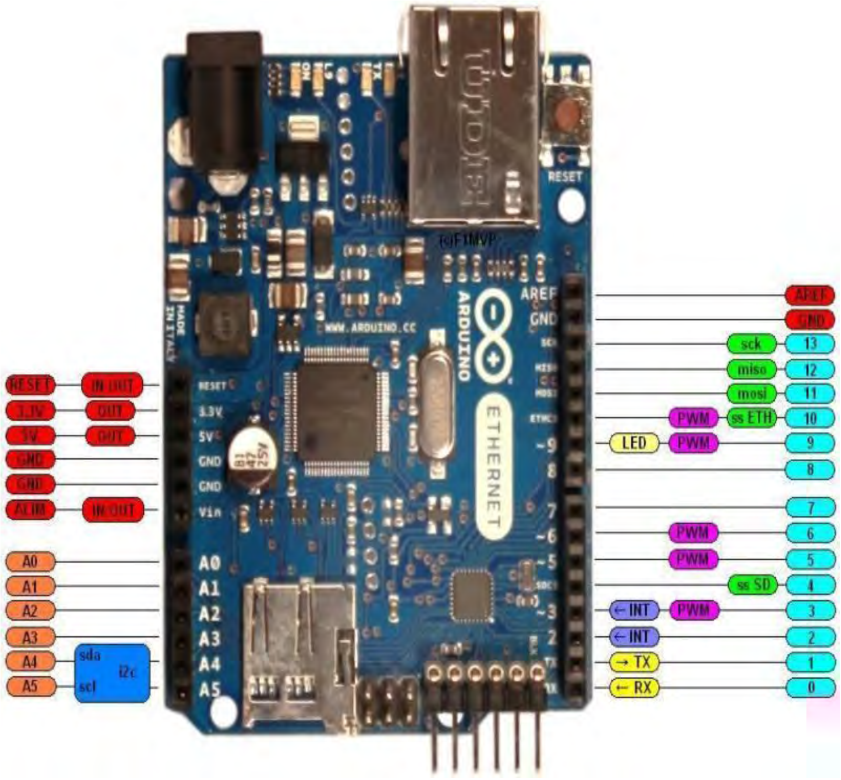


Figure 6. Arduino Ethernet

Arduino Bluetooth Module

It is an Arduino board designed for communication using bluetooth protocol as software. It is used with Arduino BT module. It works with the feature of being programmable via Bluetooth.

Arduino Pro Mini

Arduino Pro Mini mostly provides great convenience for advanced users. In addition , the fact that the header pins is not on top and the USB converter circuit is not integrated has made it small in size .Arduino pro mini card is much more useful for projects that are planned to be done on a small scale. Figure 7 shows the Arduino Pro mini. It has 14 digital pins, 8 analog pins, VCC, RST, GND pins.

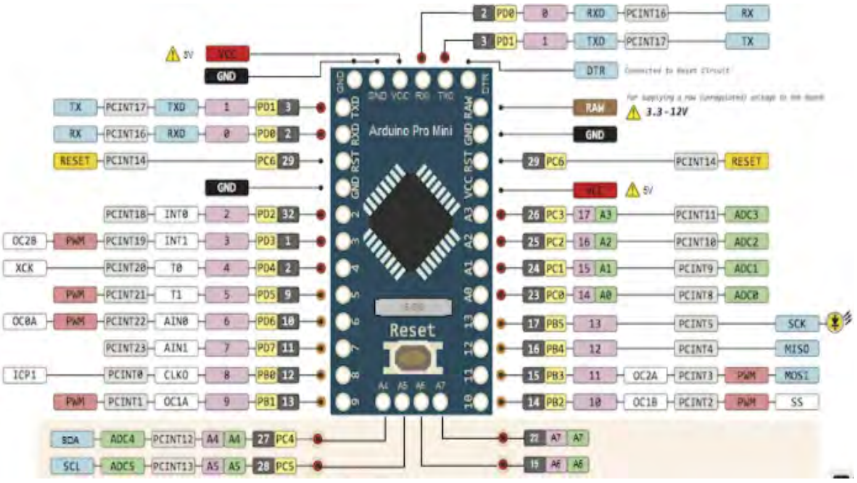


Figure 7. Arduino Pro Mini

Arduino Leonardo

It is a type of Arduino board that includes the ATmega32u4 microcontroller. The difference in the Arduino Leonardo card from other cards is that the ATmega32u4 microcontroller has its own internal USB communication feature. Therefore, a second processor is not required. Therefore, the Arduino Leonardo can appear as a mouse or keyboard or as a virtual CDC serial / COM port to a computer to which it is connected. There are 20 digital input / output pins on the Arduino Leonardo. 7 of them are used as PWM outputs and 12 of them are used as analog inputs. In addition, the board has a 16 MHz crystal oscillator, USB connection, power jack (2.1mm), ICSP header and reset button. Arduino Leonardo contains all the necessary components to support a microcontroller. In Figure 8, the Arduino Leonardo board is given with its explanations.

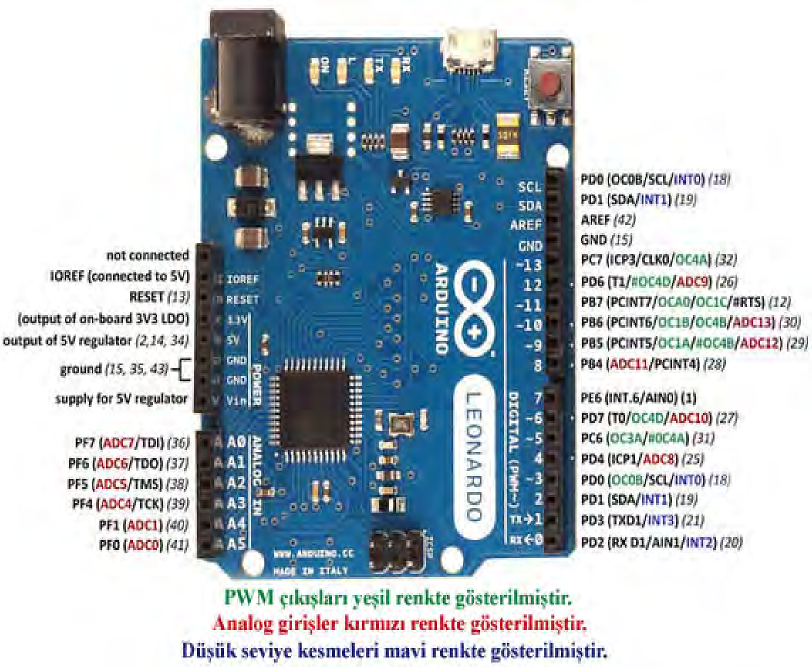


Figure 8. Arduino Pro Mini (Robotic Automation and Systems, 2018)

Arduino Nano

It is a handy Arduino board for small projects with an ATmega328 microcontroller or an Atmega168 microcontroller on it. It has many of the same functions as the Arduino Duemilanove. Arduino Nano is designed and used by Gravitech. In as width (18 mm x 45 mm) it is 12 mm larger than the Arduino Mini board (18 mm x 33 mm). As seen in Figure 9, there are 8 analog input pins, 14 digital input/output pins, reset button, 5-12 volt supply inputs, and gnd pins on the card. The Arduino Nano can be worked by connecting a computer with a type B mini USB cable. It can also be powered from an external power sourceAn external power supply in the range of 6 to 20 volt can be connected to pin 30. It can be fed from pin 27 with a regulated voltage of 5V.

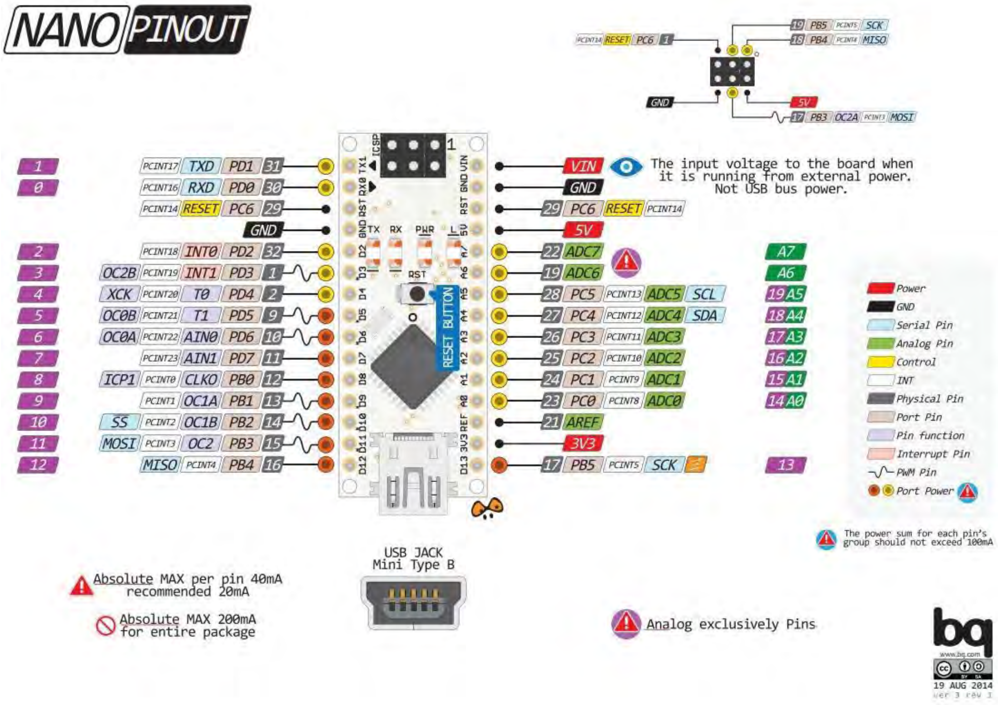


Figure 9. Arduino Nano

Arduino Compiler Programs

Arduino software consists of a development environment (IDE) and libraries. Libraries are written in C and C++ languages and include AVR-GCC and AVR Libc compiled with. Various compiler programs are used to program Arduino circuits prepared by the user.

These;

- Arduino Ide
- ArduinoBlocks
- Mblock

- S4A
- Eclipse
- Visual Studio

Arduino IDE

The development card works by installing via the usb interface developed with the wiring programming language. Arduino ide is a physical programming platform consisting of a development environment containing an application of the wiring language. Arduino ide is a physical programming platform consisting of a development environment containing an application of the wiring language. In terms of software, the Arduino IDE acts as a code editor and compiler. At the same time, it is an application written in the Java programming language, which can load the compiled program to the card and enable it to work on any platform (İbrahim Cayıroglu, 2015).

ArduinoBlocks

The Arduino IDE editor offers the opportunity to write code using the C++ programming language. One of the good alternatives that allows block-based coding is the ArduinoBlocks online editorThis application works through the browser. In addition, it offers the opportunity to be used in all systems without any problems. For the system to work, at the http://www.arduinoblocks.com/web/site/abconnector from web address, Windows 64 bits tab installer.exe package it is necessary to download and install the. After this package is installed, an application called abconnector is installed on the operating system. The development board can be coded by running this application, minimizing it and going to the Arduinoblocks website in your browser.

In Figure 10 web tool, there are sensors on the left, a worksheet where we can create block codes in the middle, and uploading to the Arduino development board, taking a picture of the environment and saving it to the outside on the upper right.

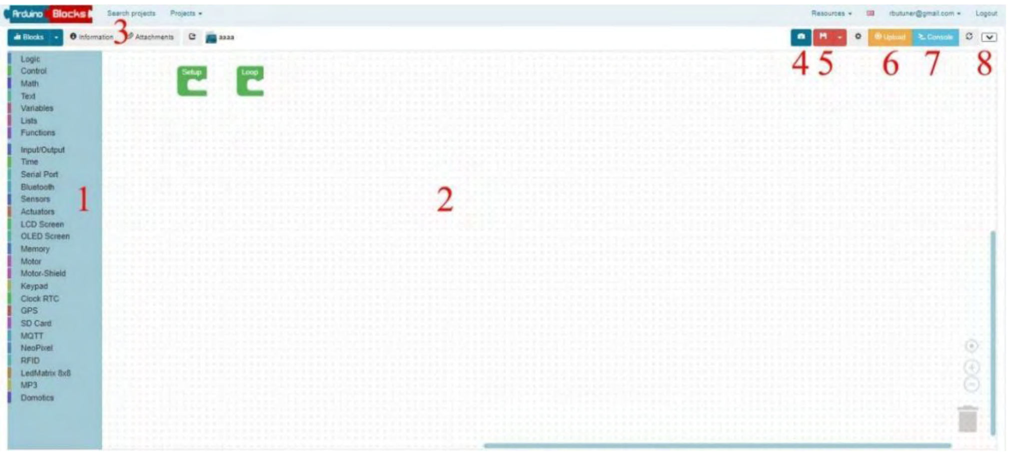


Figure 10. Arduinoblocks Web Tool

According to the Arduinoblocks Web Tool in Figure 10;

1. In order to do block coding, there are blocks of drag and drop logic. These are decision, control, calculation, text, variables, lists, sensors, etc. exists.
2. It is the necessary worksheet for block coding.
3. They are block coding menus. In these menus, the page information for block coding shows the text encoding.
4. It saves by taking a screenshot of the block coding on the worksheet.
5. It performs the process of registering the application for the online arduinoblocks web tool or external environment .
6. Uploads the block coding from the worksheet to the Arduino development card.
7. Open the serial communication console screen.
8. Indicate whether the Arduino board is connected to the web tool.

In order for the ArduinoBlocks web tool to upload to the development board, the ArduinoBlocks Connector application must be running. You can download this application for its website as in Figure 11.

```
>> ArduinoBlocks-Connector V8

>> by Juanjo Lopez
>> www.arduinoblocks.com
>> Listening on port 9987
>> (Ctrl+C to finish)
>> Ready...
Checking libraries...
Libraries version: 23
Downloading new libraries...
Libraries updated to version 25
```

Figure 11. ArduinoBlocks Connector Uygulaması

Mblock

Mblock is a block-based coding environment. The working logic is just like Scratch. Thanks to the drag and drop method, large code blocks can be created by placing the code blocks side by side, under each other, and the character can be moved as we want with these code blocks. Thanks to the drag and drop method, large code blocks can be created by placing the code blocks side by side, under each other, and the character can be moved as we want with these code blocks. The Mblock online editor screen is shown in Figure 12.

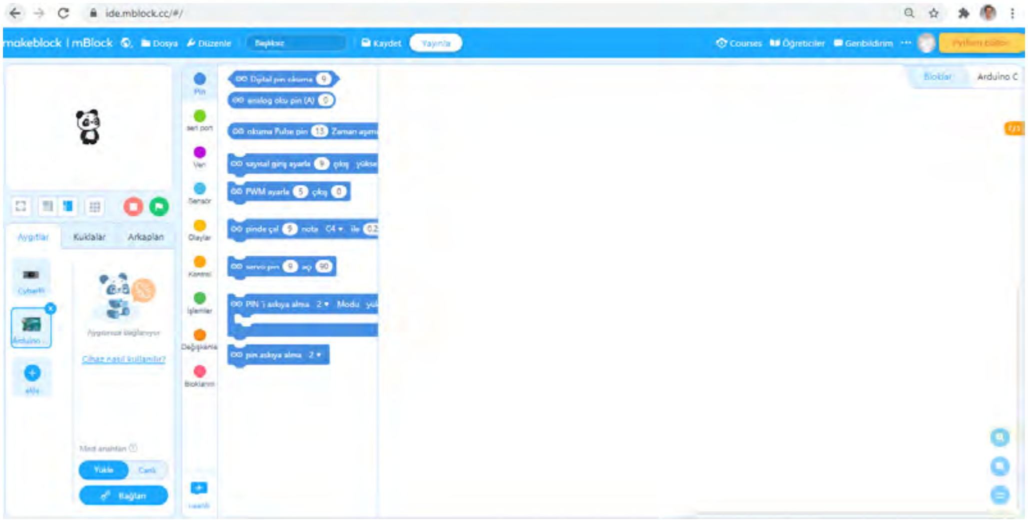


Figure 12. Mblock Online Interface

S4A

S4A program is a Scratch based programming language. There are some programming blocks of the editor that allows us to control the Arduino. Thanks to these program blocks, Arduino can be programmed completely visually without writing code. Apart from the fact that the program can be written in blocks, S4A's Turkish language support provides a separate convenience to programming. The S4A editor programs interface screen is given in Figure 13.

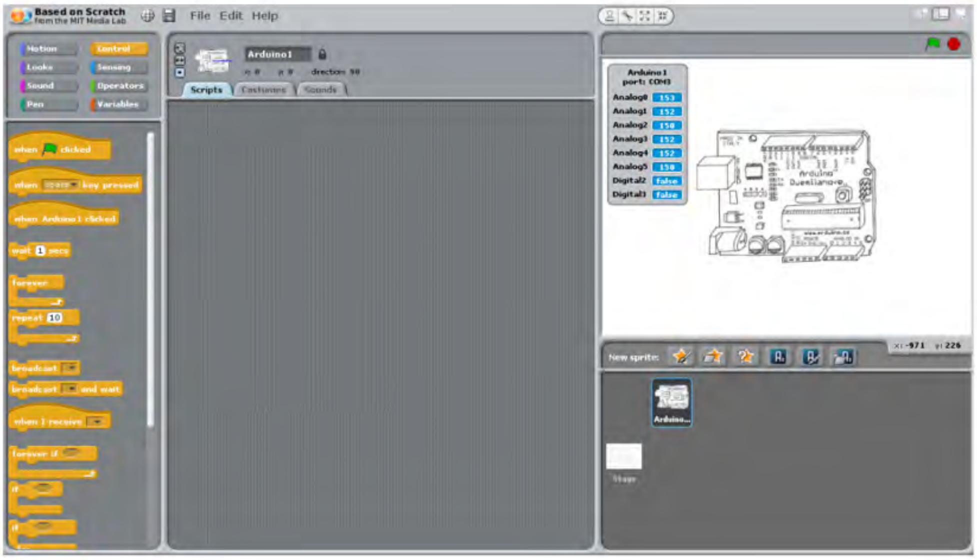


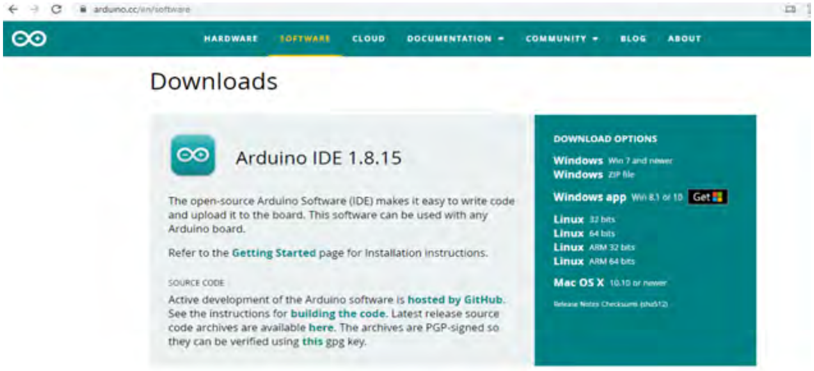
Figure 13. S4A Interface

Arduino Visual Studio

It is integrated into Microsoft's development environment, Visual Studio. The speed of writing programming with Visual Studio will increase by 2 times. The name of the plugin that transfers the Arduino compilation, upload system to the Visual Studio editor is Visual Micro.

Installing Arduino Ide Program

In order to install the Arduino Ide program, https://www.arduino.cc/en/software from the website address Arduino software is firstly downloaded. For Windows 10 users, the “Windows Installer, for Windows 7 and up” option is selected. If you want to download for free on the screen after the selection process, the “Just Download” option is selected. Also, if you want to contribute, “Contribute & Download” is selected and the download starts. Figure 14 shows the Arduino IDE program download screen.



The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the Getting Started page for installation instructions.

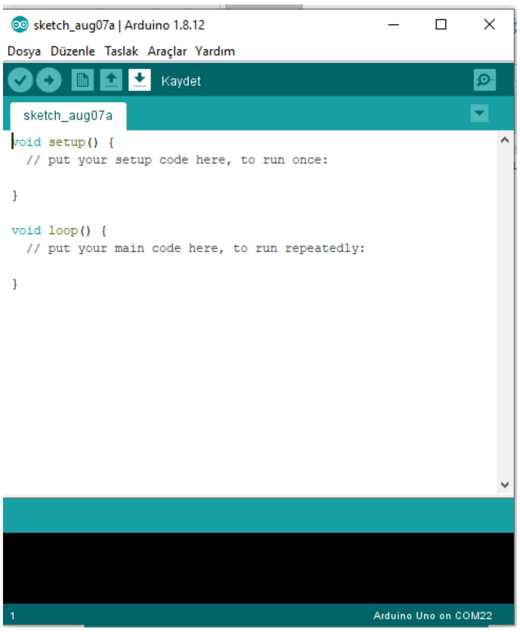
SOURCE CODE

Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

Figure 14. Arduino Ide Loading screen

Using Arduino Ide Interface

Arduino ide is an application designed to program development card. By writing the code of our program in the void setup() and void loop() sections in Figure 15, the installation process can be performed.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Şekil 15. Arduino Ide Program Interface

When we look at the functions of the buttons in Figure 15;

✓ It is for compiling the written program. Written information on orange background will be given when there is an error.

➜ If there is no problem/error in the program and Arduino is connected to the computer, uploading to the card can be done.

🗎 It is the shortcut given to create a new file.

↑ It is a given shortcut to open the previously saved file.

↓ Shortcut to save the work done (Altun, 2017).

Basic Arduino Functions

void setup(): This function run once when the program is opened with the first time and the necessary calibration and setup commands are written here.

void loop() : When the loop function runs, the commands of parentheses are repeated in an endless loop.

In Arduino programming schematic;

```
// At first we can add the libraries

// We can define variables of global type

// We can write our functions here

void setup()

{

// The codes we want to run at first are written here

}

void loop()

{

 // our main function in the form of an infinite loop

 // The program is written here.

}
```

Adding a Library

In order to add a new library to the Arduino ide application, the files of the library are moved to the 'libraries' folder under the Arduino program installed directory. After adding the library, the following code is used to call our Arduino ide program.

#include <libraryname.h> is in the form. After that, the functions of the library can be used.

PinMode: It is used to set the input and output status of the pins in Arduino. The pins to be used in the setup are set.

pinMode(10,OUTPUT); //Pin 10 is defined as output,

pinMode(11,INPUT); //Pin 11 is defined as input,

DigitalWrite: It is used to energize or stop energizing the pins defined as outputs. A value of 1 is given to energize the desired pin and 0 to stop it.

digitalWrite(9, HIGH); //Logic 1 is set to pin 9, or

digitalWrite(9,1);   //Logic 1 has been made to pin 9,

AnalogWrite: It enables the PWM signal to be generated on the selected pin, and the duty time of the PWM signal is determined by the value given to the function. When we think that the led is connected on the seventh pin, it is used to adjust the brightness of the led light with the PWM signal.

analogWrite(7, 180); // A 180 value analog signal is given to pin 7.

digitalRead: Logically reads the specified pin status. If the pin is logic 1, it returns 1, and if it is logic 0, it returns zero integer value.

int deger1= digitalRead(5);   // Read pin 5 and assign to variable value1

int değer2= digitalRead(8);   // Read pin 9 and assign to variable value2

delay(): It is used to give a wait between two codes . The value written into the delay function is milliseconds. The value 1000 must be written to the function of an one second wait.

delay(500); // Gives half a second pause,

Serial.begin(9600); // serial communication is started,

Serial.print("merhaba"); // The expression in quotes is sent to the serial monitor. The data is printed side by side on the screen.

References

Akinci, A., & Tuzun, H. (2016). Student Views on the Use of Robots in Programming Education.10th International Computer . Instructional Technologies Symposium, 135. Rize.

Altun, S. (2017). Adim Adim Arduino. Mardin.

Barman, G. (2014). Gorme Engellilere Yardimci Ultrasonik Cihaz. Trabzon: Karadeniz Technical University Computer Engineering Department.

Baz, F. (2018). A Comparative Analysis of Coding Software for Children. Curr Res Educ, 4(1),36-47.

Butuner, R. (2019). Effect of Coding and Robotic Coding Training on Students. Journal of Information Systems and Management Research, 24-30

Ersoy, H.,Gulbahar, Y., & Madran, R. O. (2011). A Model Proposal for Teaching Programming Languages: Robot Programming. XIII. Academic Informatics Conference,731-735. Malatya: Inonu University.

Gelen, I. (2017). P21- 21st Century Skill Frameworks in Curriculum and Instruction(USA Practices). Journal of Interdisciplinary Educational Research, 15-29.

Goksoy, S., & Yilmaz, I. (2018). The Opinions Of Information Relations Teacher And Their Students Wıth Regard To Lessons Of Robots And Coding. Duzce University Journal of Social Sciences Institute,8(1), 178-196.

Cayiroglu I. (2015). Arduino ile Park Sensoru Uygulamasi, http://www.ibrahimcayiroglu.com/Dokumanlar/MekatronikProjeUygulamasi/37-Makale-Arduino_Ile_Park_Sensoru_Uygulamasi-Mehmet_CIKAN.pdf

MEB. (2018). Bilgisayar bilimi dersi kur 1-2 ögretim programi. MEB Ogretim Programlarini Izleme ve Degerlendirme Sistemi: Access Adress: http://mufredat.meb.gov.tr/Dosyalar/2018120203611364

Robotic Automation and Systems. (2018). *Arduıno Leonardo*. Robotic Systems: Access Adress:http://www.robotiksistem.com/arduino_leonardo_ozellikleri.html

Yildız, R., Talaslioglu, S. S., & Yildirim, M. (2020). Determination of Situations of Extra-curricular Practices Carried About Robotics, Coding and Electronics. IBAD Journal of Social Sciences, 193-208.

Zohra, B. (2015). Arduino Ile Sensor Uygulamalari. Tokat: Gaziosmanpasa University.

About the Authors

Resul BUTUNER,i s a Computer Teacher at Adil Karaagac Vocational and Technical Anatolian High School in Konya, Turkey. He has a master's degree in Computer Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, robotic coding, data mining and augmented reality applications. He is an instructor in the field of Robotic coding within TUBITAK. He continues to write a book in the field of robotic coding at the Ministry of National Education. He worked as a coordinator in budgeted projects related to student education.

E-mail: rbutuner@gmail. com, Orcid: 0000-0002-9778-2349

Yusuf UZUN, PhD, is an Assistant Professor of Computer Engineering at Necmettin Erbakan University in Konya, Turkey. He holds a PhD in Mechanical Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, autonomous systems and augmented reality applications. He also works as the Rector's Advisor at Selcuk University.

E-mail: yuzun@erbakan.edu.tr, Orcid: 0000-0002-7061-8784.

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 9%.

To Cite This Chapter:

Butuner, R. & Uzun, Y. (2021). Arduino Microcontroller Card. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 1–18). ISRES Publishing.

Coding With BBC Mikro:bit

Zeynep SAGLAM
Zonguldak Çaycuma TSO Science High School

Rumeysa ERDOGAN
Bartın University

Introduction

Computer programs are the writing of a set of codes that allow algorithms to be solved by computer. The target audience of programming languages is generally considered as students studying in computer-related departments (Durak, Karaoğlan-Yılmaz, Yılmaz & Seferoğlu, 2017), learning these programs is an extra difficult and time-consuming process (Erol & Kurt, 2017), studies show that students perceive this process as difficult (Özmen & Altun, 2014). To learn a programming language, it is important to have basic maths knowledge. As artificial intelligence, internet of things, cyber security, future professions, dark factories, digital game technologies became widespread, people's perspective and interest in technology has changed. The rapid development of technology has brought together the concept of 21 st century skills (Cansoy, 2018) and in this century, the concept of education aims to bring skills and competence rather than to present knowledge to the abilities of individual (Uçak & Erdem, 2020). Skills such as problem solving, critical thinking, collaboration, communication, creativity must be gained in schools (National Research Council, 2012). For a developed society, qualified human power is very important (Sayın & Seferoğlu, 2016). Creating algorithms and solving problems with these algorithms is an important skill within the scope of 21st century skills. Another concept that is associated with the concept of algorithm is programming skill. The use of block coding tools has made it easier to learn this skill from an early age (Sırakaya, 2018) and many programs can be written even without basic mathematical knowledge. The widespread use of block coding tools and their use by large users has brought the concept of robotics to become widespread (Erten, 2019).

In the 21st century technology age, it has become very important to use the innovations required by the age in lessons in order to ensure the permanence of education, and the concepts of robotics and robotics - coding are the most important of these innovations (Butuner, 2019). It can be said that robotics is the programming of hardware to perform the desired task using coding tools. While the programmed hardware was coded with programming languages that require more electronic knowledge and are closer to machine language, in recent years they have turned into tools programmed with block coding tools. The emergence of tools that do not require much electronic knowledge, are practical to disassemble and install, and can be used repeatedly, has made it easier to learn the work done in the field of robotics from a younger age. It is said that robots will take over many jobs with the change of professions, in this case it is very important for children to learn how to code robots. Teachers who are educated in the field can provide education in this field by choosing robotics and coding tools suitable for the age level of the student. There are many robotics and coding tools developed according to the age group and student knowledge level. There are options such as Arduino, BBC Micro:bit, lego, makeblock, which are popularly known by many users and those who want to develop projects can find many sample projects.

Coding with BBC Micro:bit

BBC Micro:bit is a small size (4cm x 5cm) electronic card that contains a microcontroller on it. It is designed by the BBC for use in computer education (Ball et al., 2016). It is especially suitable for younger children to do robotics and coding and an electronic circuit board for teaching the basic coding logic on a block-based basis (Butuner & Dündar, 2018). Activities can be carried out to make primary school students familiar with concepts such as algorithmic thinking, programming, game development (Videnovik, Zdravevski, Lameski & Trajkovik, 2018). The United Kingdom made an attempt by distributing a BBC Micro:bit card to nearly one million 7th grade students (12-13 years old) due to the lack of students taking enough computer science courses at the university and the lack of qualified people in the field (Rogers et al., 2017; Schmidt, 2016). The United Kingdom made computing curriculum compulsory in schools in 2014, and in 2015 the BBC planned to motivate students in this area with its Micro:bit distribution initiative (Sentance, Waite, Hodges, MacLeod & Yeomans, 2017).

Features of BBC Micro:bit Card

Figure 1 shows the features of the first version of the BBC Micro:bit card. Figure 2 shows the latest version of the card with new features. As with every technological device produced, new features have been added to the device in the new version.

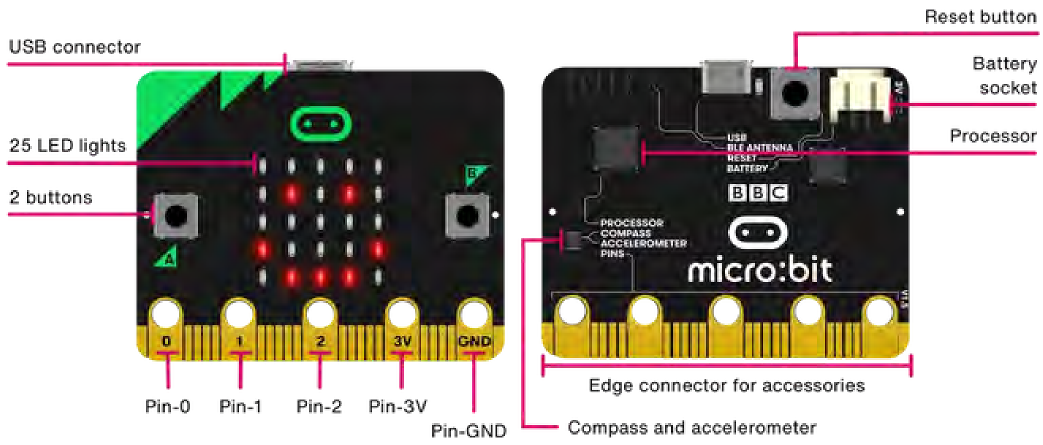


Figure 1. First Version of the BBC Mikro:Bit Card (Url1, 2021)

When Figure 1 is examined, it is seen that there are 2 buttons, a usb input, 25 LED outputs, 1 reset button, input-output pins, processor, power input, bluetooth, compass and accelerometer sensors on the BBC Micro:bit card.

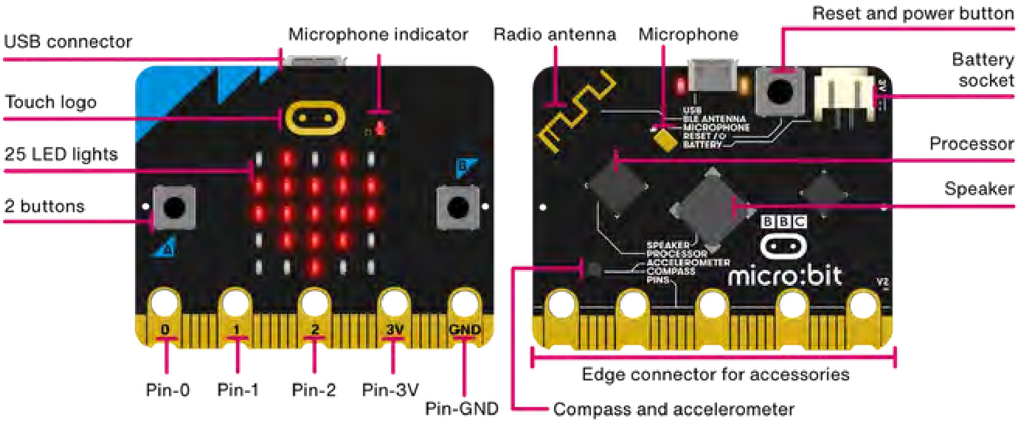


Figure 2. The Latest Version of the BBC Mikro:Bit Card (Url1, 2021

When Figure 2 is examined, it is seen that audio related features have been added to the new version of the BBC Micro:bit card. It is seen that unlike the previous version, a microphone, speaker and touch logo are added on the card. Examining the features of the hardware on the card will give an idea about what can be done with this card. The hardware built into the card is as follows:

- **Buttons:** There are two buttons named A and B on both sides of the card. The buttons can be programmed to work together as well as separately (Halfacree, 2017). The button can be used as an input device in many projects. The fact that the buttons are built into the card allows the development of projects using buttons without the need to explain the button binding business that requires electronic information to young age groups.

- **25 LEDs:** There are 25 LEDs arranged as 5 x 5 on the BBC Micro:bit card. It can be used as a screen to show words, numbers and symbols.

- **Light sensor:** The 25 LEDs on the card also function as a light sensor. With the light sensor, the amount of light falling on the device can be measured.

- **Processor:** It has a processor with 16K RAM running at 16 MHz (Schmidt, 2016). It can be described as the brain that runs the commands in the programs written.

- **Temperature sensor:** The processor also includes a temperature sensor, so projects that measure ambient temperature can be made.

- **Compass:** The BBC Micro:bit card contains the compass sensor embedded in it. This input sensor detects magnetic fields.

- **Accelerometer:** The acceleration sensor is a sensor that measures movement. It detects when the card is tilted right, left, up or down. Projects that alarm when the card is shaken or displaced can be done.

- **Bluetooth:** With the bluetooth receiver on the BBC Micro:bit, it can connect and communicate with another Micro:bit and other devices with bluetooth feature.

- **USB interface:** USB or universal serial bus used to connect computers, other devices and power. With the USB on the BBC Micro:bit, it can be connected to the computer and loaded into the program, at the same time it provides the power required by the microcontroller.

- **Audio output:** BBC Micro:bit, can be encoded by writing programs that produce sounds. However, if you are using the first version, you must use an external headset or speakers to hear these sounds.

New Version Features:

- **Built-in speaker:** There is a built-in speaker on the new version BBC Micro:bit. While in the previous version, sound is output by using headphones or speakers in sound related projects, there is no need for an extra sound output with the new version.

- **Microphone:** With the microphone built into it, projects related to sounds can be improved. The microphone also works as an audio sensor that measures the amount of sound.

- **Touch logo:** There is a touch sensor on the new BBC Micro:bit. This sensor on the logo enables data input by touch.

Pins on the BBC Micro:bit Card

In addition to the built-in inputs and outputs on the BBC Mikro:bit, new inputs-outputs can be added if desired. There are 25 pins on the board to add input outputs. Figure 3 shows the pins on the card. When Figure 3 is examined, it can be seen that there are 5 large pins shown with holes. These pins can be used by connecting equipment for input and output using alligator cables (Hodges, Sentance, Finney & Ball, 2020). Pins 0, 1, 2 are flexible pins used for general input-output purposes. They are named as GPIO, these pins have analog to digital converter (ADC) feature.

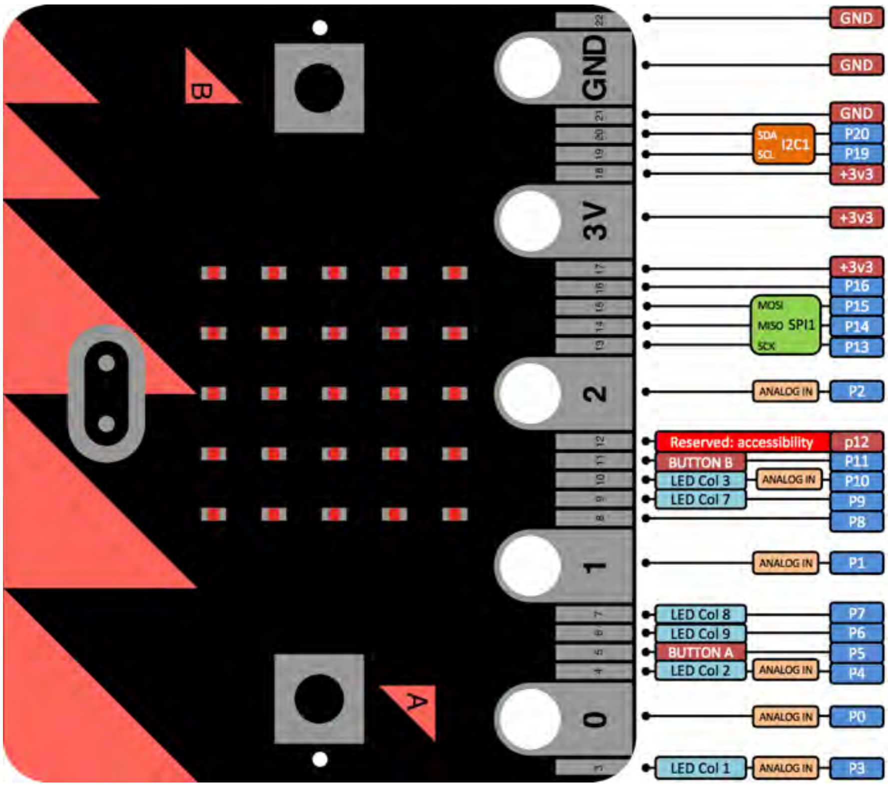


Figure 3. Pins on the BBC Micro:Bit Card (Url2, 2021)

When Figure 3 is examined, it can be seen for what purposes the pins can be used. It is possible to divide the pins into two groups as large pins and small pins.

Large pins:

- **P0:** GPIO pin, ADC
- **P1:** GPIO pin, ADC
- **P2:** GPIO pin, ADC
- **3V**: It can be used as a 3 Volt power input or output. It can be used as an output to power peripherals if the BBC is powered by a Micro:bit USB from a computer or is battery powered. If power is not connected via USB or battery, the 3V pin can be used to power the board externally.
- **GND:** It is the ground pin used to complete the electrical circuit.

Small pins:

There are 20 small pins on the board, these small pins are shared with some components on the board. When Figure 3 is examined, it is seen that these pins are named between 3 and 22 respectively. These pins can be operated using the expansion board.

- **Pin 3:** It is shared with column 1 of the sequential leds on the card, when these leds are turned off, they can be used as digital I / O.

- **Pin 4:** It is shared with column 2 of the LEDs that are sequentially on the card and these LEDs are used as digital I / O when they are turned off.
- **Pin 5:** Shared with the A button built into the card. Enables the feature of clicking the A button on the card to be triggered or to detect when clicked. An external button can be connected and operated via this pin.
- **Pin 6:** It is shared with column 9 from leds and can be used as digital I / O when these leds are not used.
- **Pin 7:** Shared with column 8 from leds and if these leds are not used, they can be used as digital I / O.
- **Pin 8:** GPIO used as digital I / O.
- **Pin 9:** Shared with column 7 of the built-in leds and these leds can be used as digital I / O when not used.
- **Pin 10:** Shared with column 3 from leds and can be used as digital I / O when these leds are not used.
- **Pin 11:** Shared with the B button, one of the built-in buttons. This allows the button to be triggered or detected externally.
- **Pin 12:** Reserved to provide accessibility support.
- **Pin 13:** Special reserved pin for SCK signal.
- **Pin 14:** Special reserved GPIO for Master in Slave Out.
- **Pin 15:** Special reserved GPIO for Master Out Slave In.
- **Pin 16:** Special GPIO, used for “Chip select” function.
- **Pin 17 and 18:** Functions the same as the large 3V pin.
- **Pins 19 and 20:** are the I2C bus. Multiple devices can be connected to the same bus with I2C.
- **Pin 21 and 22:** These pins are used for the same purpose as GND.

BBC Micro: bit Coding Tools

In BBC Micro:bit Educational Foundation, users and collaborators share the projects as open source and anyone who wants can access these codes. There are platforms available online to program the card. The coding tools available for programming the BBC Micro:bit are:

Microsoft Makecode Editor:

It allows programming the card with blocks or JavaScript. Figure 4 contains the editor's image. Block-based coding tools allow the young age group or people who are not interested in the field to easily encode the card. The editor also acts as a simulator and is important in that it allows people who do not yet physically have a card to program and operate the card. The program can be downloaded and recorded on the card connected to the computer with a USB cable.

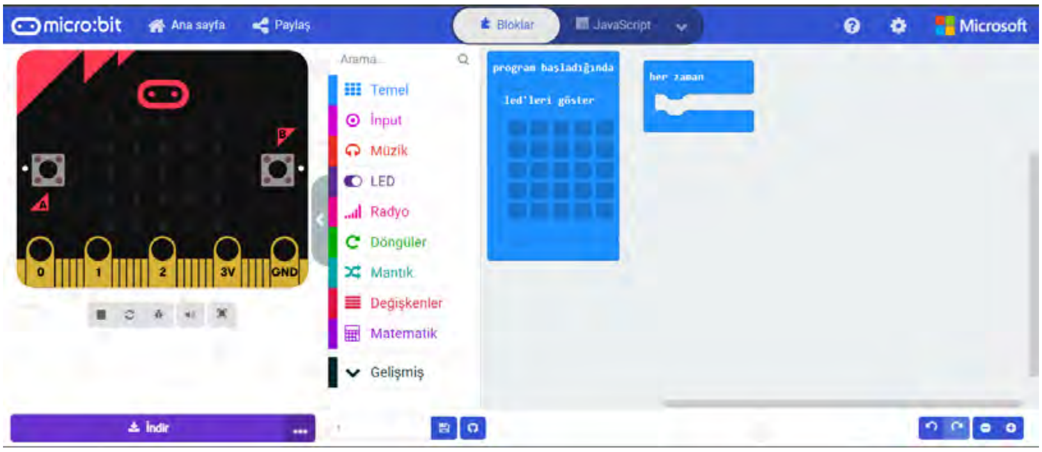
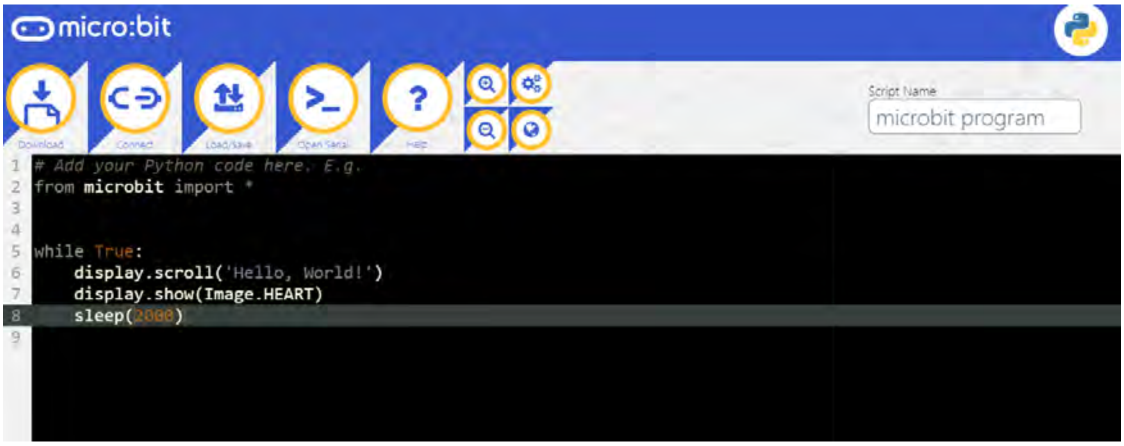


Figure 4. Microsoft Makecode Editor (Url3, 2021)

Python Editor:

There is a Micro:bit python editor developed for the use of the BBC Micro:bit card by users who know python programming language. Figure 5 shows the image of the editor.



Script Name: microbit program

```
# Add your Python code here. E.g.
from microbit import *


while True:
    display.scroll('Hello, World!')
    display.show(Image.HEART)
    sleep(2000)

```

Figure 5. Mikro:Bit Python Editor (Url4, 2021)

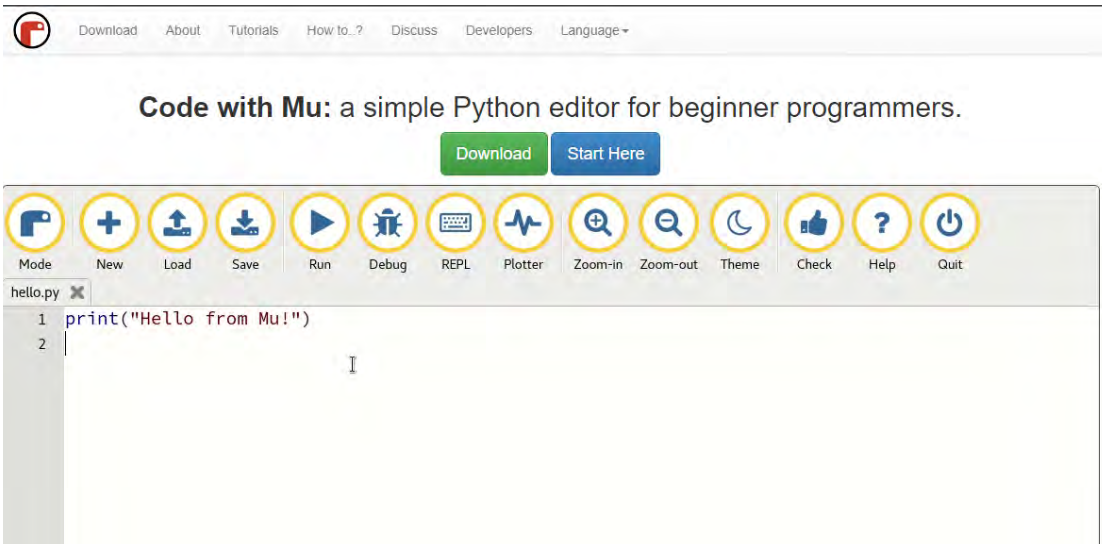
As seen in Figure 5, the editor works with python codes. The program written using the Python programming language is saved as a hex file with the download option on the Micro:bit card. The save option allows the written program to be saved to the computer as .py extension. Created by volunteers, the editor is maintained by Mikro:bit Education Foundation. Open source software continues to be developed by volunteers.

Coding with MikroPython:

It can be said that it is a simple version of Python 3 language. It is a version of python developed to work in microcontroller environments, containing a small subset of the Python standard library. It is written using the C programming language. Before the program is loaded to the Arduino board, the compilation process is performed and in case of faulty code, it is not uploaded to the card. However, in MicroPython, the codes are loaded into the microcontroller card first and the compilation process takes place inside the card. If the codes loaded into the card are incorrect, it gives a warning.

Mu Editor:

Online platforms require a constant connection to the internet, while the codes are being written, Mu, a simple editor that allows the codes to be written and uploaded to the card quickly, without the need to be constantly connected to the internet. The Mu Editor is developed using Python. It needs to be downloaded and installed on the computer. It runs on Windows, OSX, Linux, Raspberry Pi. Figure 6 is the image of the Mu Editor.



Code with Mu: a simple Python editor for beginner programmers.

```
print("Hello from Mu!")

```

Figure 6. Mu Editor

Scratch Editor:

It is a coding tool developed in MIT Media Lab so that beginner users can learn how to code (Meerbaum-Salant, Armoni & Ben-Ari, 2013). It is a frequently used tool in schools for younger age groups to learn coding. The coding tool can be accessed online or installed on the computer. It is an advantage that people who are particularly familiar with this tool can also use it while working with the BBC Micro:bit. In order to use BBC Mikro:bit on the Online Scratch platform, the plug-in must be installed. In order to install a plug-in, the plug-in in Figure 7 must be loaded under the Add extension menu in the Scratch interface.

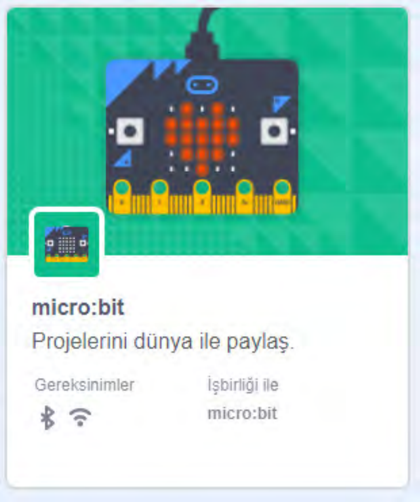


Figure 7. Scratch Micro:bit plug-in

When the plug-in is installed, the codes for programming the BBC Micro:bit tool in Figure 8 are displayed. This allows beginner users and students who are familiar with the scratch programming interface to easily program the tool.

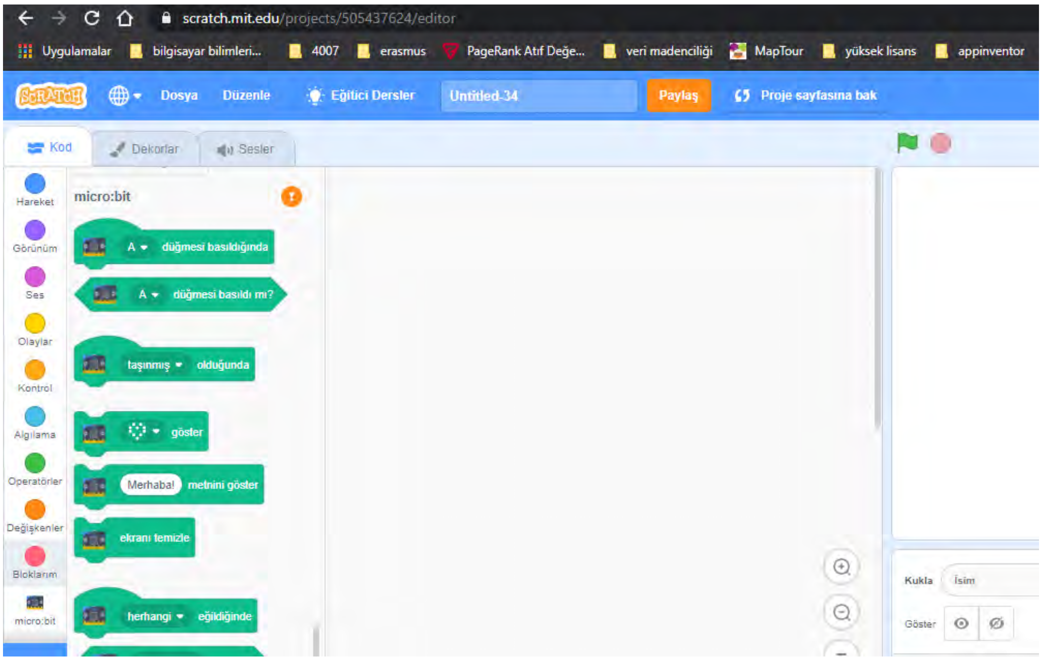


Figure 8. MIT Scratch Editor Micro:bit Codes

Codes written using the Scratch interface can be uploaded to the BBC Micro:bit tool via bluetooth.

MBlock Version5 Editor:

The Mblock coding tool allows the programming of different microcontrollers. Using the 5th version of the tool, the BBC Micro:bit can be encoded. Coding operations can be performed by adding the extension in Figure 9 by adding from the extensions section in the editor.

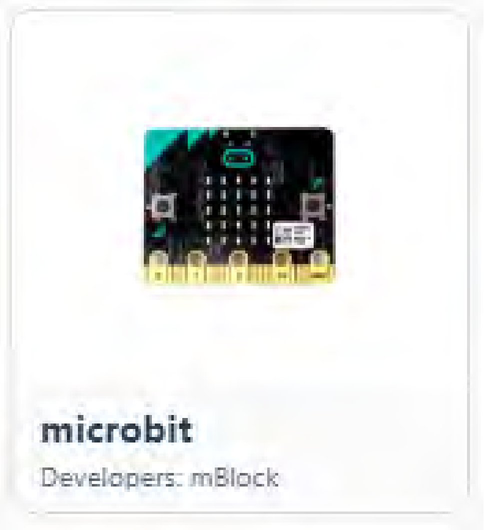
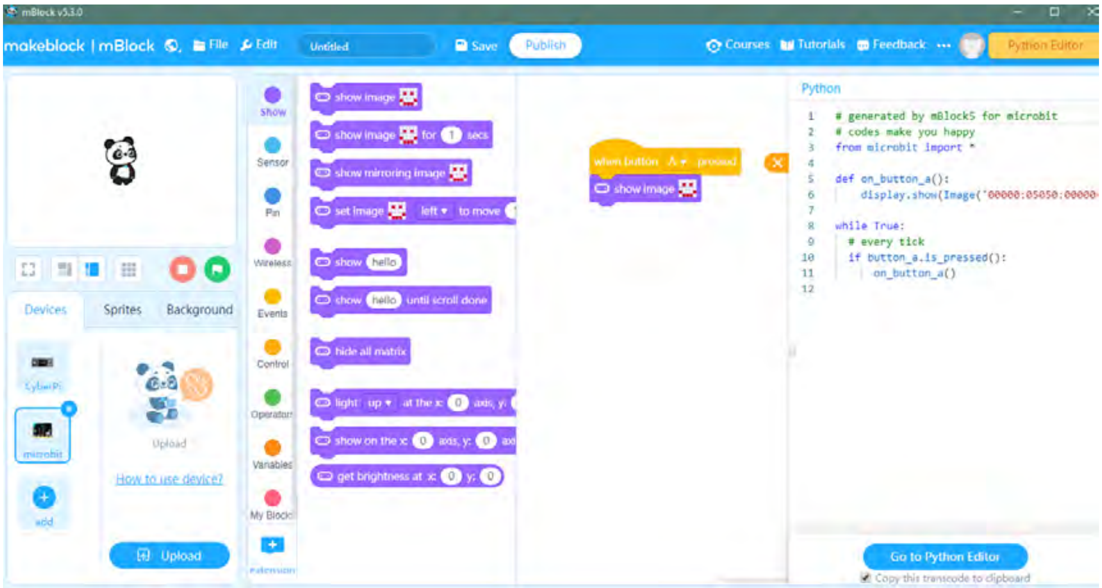


Figure 9. Adding Mikro:Bit with MBlock Plugin

Figure 10 shows the codes that come after adding the Micro:bit plugin in the Mblock editor.



```
# generated by mBlock5 for microbit
# codes make you happy
from microbit import *

def on_button_a():
    display.show(Image('00000:05050:00000

while True:
  # every tick
  if button_a.is_pressed():
    on_button_a()

```

Figure 10. MBlock Editor Micro:Bit Codes

After the desired program is written with block codes in the Mblock editor, it can also be looked at how these codes are written in the python programming language, if desired. When Figure 10 is examined, how the block codes look with the python editor can be seen on the right side.

BBC Micro:bit and Education in the World

The essence of programming is not in the line of code, but in the way of thinking, one should be able to imagine and implement it using commands (Lesničar, 2018). BBC Mikro:bit is a simple piece of hardware designed for students to bring their dreams to life. While programming tools output the codes by means of computers, robotic tools output by using hardware. STEM (Science, Technology, Engineering, Mathematics) is an abbreviation that connects basic sciences. It aims to bring together the acquisitions that students have learned in different disciplines with STEM activities. Education should provide students with problem-solving skills beyond rote knowledge. The most important of the 21st century skills is considered to be problem solving (Kırkıç, Derin & Aydın, 2018). The most important tools that provide STEM education are robotic tools. The product created by students bringing together many disciplines can be a solution to a problem. Croatia made a STEM revolution in 2017 on the grounds that there were not enough educated people about STEM in the country and used the BBC Micro:bit as the most important tool in this movement (Lesničar, 2018).

The Micro:bit Educational Foundation in the UK is a foundation established to ensure the continuity of the BBC Micro:bit project given to school children in England and to provide voluntary education (URL 5, 2021). It continues to support education around the world. No one is excluded from digital education in Singapore. With the Digital Maker program launched by Infocomm Media Development Authority in 2017, 100,000 Micro:bit cards were given, teachers from different branches had to participate in workshops and produce prototypes (URL6, 2021). Finland is a country that cares about being creative and learning by doing. Innokas, a network of educators and organizations built into the University of Helsinki, is a structure that encourages alternative work practices that enhance critical and creative thinking and is interested in using BBC Micro:bit when preparing children for future professions (URL7, 2021). Academic staff at the University of Duplin Technology in Ireland volunteered to educate more than 5000 students using BBC Micro:bit (URL8, 2021). Denmark Ultra:bit is a project that was introduced to 4th grades (8-9 years old) in 2018-2019, and to 5th and 6th grades in 2019-2020, established in 90% of schools and aims to make students love writing code (URL9, 2021). In many different countries around the world (27), new projects are started to encourage students' creativity, Micro:bit is used in these projects, and approximately 25 million children benefit from these projects (URL10, 2021).

Conclusion

While educational environments are organized to raise generations that produce and not consume, all over the world (Öz, 2019), one of the most important elements in these environments is robotic-coding tools (Şahin, 2019). These tools should be selected correctly in proportion to the student's age level and readiness in this regard. With these trainings starting from kindergarten, it is important to develop algorithmic thinking skills and problem-solving skills in students. The new generation students are a generation that is qualified as digital native and it is very important to use these tools correctly in the education of these students. If these tools are not suitable for the level of the student, they will be seen as difficult and complex and will decrease the motivation of the students (Ersoy, Madran & Gülbahar, 2011). In this context, tools that show the concrete results of the encoded algorithms and are not complicated for the student to use will motivate students to produce (Numanoğlu & Keser, 2017). BBC Micro:bit is a tool developed for this purpose, allowing students who are at the beginning of the road to do easy robotic-coding (Sentance et al., 2017), while allowing advanced students in this field to do their dream projects. It is more cost effective than many robotics tools (Ball et al., 2016). While it is possible for students who do not have a concrete tool to write codes for this tool with online simulators, it is a plus feature that this tool can be coded with many tools. The campaign, which was initiated by England, was distributed to 7th grade students by distributing BBC Micro:bit and encouraging them to their computer science, has become an example to the whole world and many countries have made their students enthusiastic with similar campaigns and projects. BBC Micro:bit is a tool with many plus features such as ease of use for newcomers to coding, ease of access to various sample projects thanks to the community established in its name, affordable cost, ability to recognize sensors developed for other tools, coding on different platforms, and the opportunity to experiment with online simulators. The release of the second version of the tool with new features indicates that it is open to development.

References

Ball, T., Protzenko, J., Bishop, J., Moskal, M., De Halleux, J., Braun, M., ... & Riley, C. (2016, May). Microsoft touch develop and the bbc micro: bit. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)* (pp. 637-640). IEEE.

Butuner, R. (2019). Effect of Coding and Robotic Coding Training on Students. Journal of. In Wu, W. & Alan, S.(Eds.), *Research Highlights in Education and Science*, 24-30.

Butuner, R., Dündar, Ö., (2018). Kodlama Eğitiminde Robot Kullanımı Ve Robotik Kodlama Eğitici Eğitiminde Öğretmenlerin Tecrübe Ve Görüşlerinin Alınması. In Yılmaz, E. & Sulak, S.A.(Eds.), *Human Society and Education in the Changing World*, 24-30. Palet Yayınları.

Cansoy, R. (2018). 21st Century Skills According to International Frameworks and Building *Them in the Education System. Journal of the Human and Social Sicence Researches*, *7*(4), 3112-3134.

Durak, H., Karaoğlan Yılmaz, F. G., Yılmaz, R. ve Seferoğlu, S. (2017). Erken yaşta programlama eğitimi: Araştırmalardaki güncel eğilimlerle ilgili bir inceleme. A. İşman, F. Odabaşı, ve B. Akkoyunlu (Ed.*), The Turkish Online Journal of Educational Techonogy* (ss. 119-137). Ankara: Pegem Yayınevi.

Erol, O., & Kurt, A. A. (2017). Investigation of CEIT Students' Attitudes towards Programming. *Mehmet Akif Ersoy* Üniversitesi *Eğitim Fakültesi Dergisi, 1*(41), 314–325.

Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011). A Model Proposed for Teaching Programming Languages: Robotic Programming. *Akademik Bilişim*, *11*, 731-736.

Erten, E. (2019). *A Case Study On Coding And Robotic Teaching* (Master's thesis, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü).

Halfacree, G. (2017). *The official BBC Micro: bit user guide.* Indianapolis, in: Wiley.

Hodges, S., Sentance, S., Finney, J., & Ball, T. (2020). Physical computing: A key element of modern computer science education. *Computer*, *53*(4), 20-30. doi:10.1109/MC.2019.2935058

Kırkıç, K.A., Derin, G. & Aydın, E. (2018). Yenilikçi Bir Öğretim Yaklaşımı Olarak Stem Eğitimi. Kırkıç, K.A. & Aydın, E. (Eds.), *Merhaba Stem Yenilikçi Bir Öğretim Yaklaşımı*, Bölüm 1, S.13.

Lesničar, M. (2018). *BBC micro:bit injegova primjena urazrednoj nastavi* (Doctoral dissertation, University of Pula. Faculty of Educational Sciences).

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, *23*(3), 239-264.

National Research Council (2012). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. Committee on Defining Deeper Learning and 21st Century Skills, James W. Pellegrino and Margaret L. Hilton, Editors. Board on Testing and Assessment and Board on Science Education, Division of Behavioral and Social Sciences and Education. Washington, DC: The National Academies Press.

Numanoğlu, M., & Keser, H. (2017). Robot Usage in Programmıng Teachıng - Mbot Example. *Bartin University Journal of Faculty of Education*, *6*(2), 497.

Öz, Ö. (2019). Digital Leadership: Being School Leader in Digital World. *International Journal of Leadership Studies: Theory and Practice*, *3*(1), 45-57.

Özmen, B., & Altun, A. (2014). Undergraduate students’ experiences in programming: Difficulties and obstacles. *Turkish Online Journal of Qualitative Inquiry, 5*(3), 1–27.

Rogers, Y., Shum, V., Marquardt, N., Lechelt, S., Johnson, R., Baker, H., & Davies, M. (2017). From the BBC Micro to micro: bit and Beyond: A British Innovation. *interactions*, *24*(2), 74-77.

Sayın, Z., & Seferoğlu, S. S. (2016). *Coding Education as a new 21st Century Skill and its Effect on Educational Policies.* Akademik Bilişim Konferansı. Aydın: Adnan Menderes Üniversitesi.

Schmidt, A. (2016). Increasing Computer Literacy with the BBC micro: bit. *IEEE Pervasive Computing*, *15*(2), 5-7. doi: 10.1109/MPRV.2016.23.

Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017, March). “ Creating Cool Stuff” Pupils’ Experience of the BBC micro: bit. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education* (pp. 531-536).

Sırakaya, M. (2018). Student Views on Coding Training. *Ondokuz Mayis University Journal of Education Faculty*, *37*(2), 79-90.

Şahin, E. (2019). *Application And Analysis Of Coding Teaching By Using Robotic Vehicles And Materials In 6-12 Age Groups* (Doctoral dissertation, Marmara Universitesi (Turkey)).

URL1: https://microbit.org/get-started/user-guide/overview/

URL2: https://makecode.microbit.org/device/pins

URL3: https://makecode.microbit.org/#editor

URL4: https://python.microbit.org/v/2

URL5: https://microbit.org/about/

URL6: https://microbit.org/impact/case-studies/imdas-digital-maker-programme/

URL7: https://www.innokas.fi/koulutus/

URL8: https://microbit.org/impact/case-studies/computer-science-inclusive/

URL9: https://microbit.org/impact/case-studies/dr-ultrabit/

URL10: https://microbit.org/impact/foundation-reports/

Uçak, S. & Erdem, H. H. (2020). On The Skills Of 21st Century And Philosophy Of Education In Terms Of Searching A New Aspect In Education. *Uşak Üniversitesi Eğitim Araştırmaları Dergisi*, *6*(1), 76-93.

Videnovik, M., Zdravevski, E., Lameski, P., & Trajkovik, V. (2018, April). The BBC micro: bit in the international classroom: learning experiences and first impressions. In *2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1-5). IEEE.

About the Authors

Zeynep SAGLAM, is ICT Teacher at Zonguldak Çaycuma TSO Science High School. She is receiving her master’s degree in Information Systems and Technologies at Bartın University. Her research interests include computer science, robotic coding, measurement and evaluation and learning. Her tasks are directorate of National Education Research and development, ICT Teacher, Directorate of National Education Robotic Coding Coordinator and Information Systems and Technologies Graduate student.

E-mail: zeynepelicora@gmail.com, Orcid: https://orcid.org/0000-0002-5790-6570

Rumeysa ERDOGAN, MSc, is research assistant in the Department of Computer Technology & Information Systems (CTIS) at Bartin University in Turkey. She started her master’s degree in Computer Engineering at Ankara Yıldırım Beyazıt University and is continuing. She is interested in machine learning, human computer interaction and computer science.

E-mail: rerdogan@bartin.edu.tr, Orcid: https://orcid.org/0000-0002-6218-7072

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 19%.

To Cite This Chapter:

Saglam, Z. & Erdogan, R. (2021). Coding With BBC Mikro:bit. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 19–33). ISRES Publishing.

Raspberry Pi and Models

Yusuf UZUN
Necmettin Erbakan University

Ozgur DUNDAR
Necmettin Erbakan University

What Is Raspbery Pi And History

In the 21st century technology age, it has become very important to use the innovations required by the age in lessons in order to ensure the permanence of education, and the concepts of robotics and robotics - coding are the most important of these innovations (Butuner, 2019). One of the most used boards in the field of robotic coding is the raspberry pi microcontroller card.

Raspberry Pi is a family of computers designed by the Raspberry Pi Foundation, a UK-based charity. The Raspberry Pi Foundation has released multiple Raspberry Pi models, each with slightly different features. Each Raspberry Pi model uses a System on Chip, a single microchip that contains all the basic components of a computer system; System on Chip for Raspberry Pi includes a processor and a graphics-processing unit (GPU) (Gay, 2017).

The Raspberry Pi is a small and inexpensive computer that was originally designed for educational use. Microcontroller is designed to help programmers improve their capabilities and better understand how programmable microcontrollers work. Raspberry Pi board; It looks like a computer designed on a single board, integrated with the CPU, ports and installed computer chips. It also has all the connectivity components required for the devices you want to connect to the card. The device does not come with any monitor, mouse, keyboard, internal storage. We can add components such as monitors, keyboards and mice to the device using these ports (Vijayan et al., 2015).

There are two different versions of Raspberry Pi on the market. These have been identified as Model A and Model B. The main difference between the two versions is an Ethernet and extra USB port added to the Model B. Other features and components are the same. These features also increase the price of the Model B. Both models have Arm CPU / GPU, GPIO, RCA, audio output, LEDs, USB, HDMI, power and an SD card slot. Features such as WiFi and audio input can be added to the device using the available USB ports on the board. In addition, an SD card and a power supply are required for the operating system and data storage (Sanchez, 2019).

Broadcom BCM2835 microchip containing the ARM1176JZF-S 700 MHz central processing unit was used in the first models of the Raspberry Pi. Broadcom used the BCM2836 microchip in the Raspberry Pi 2 model that was released later. As with the first model, the VideoCore IV GPU has a graphics-processing unit. It uses SD card for booting and data storage. It has USB 2.0 ports, HDMI video output, audio output, MIPI camera input, GPIO interface and 5V MicroUSB power input.

Raspbian (Debian Wheezy based), Pidora (Fedora based), Snappy Ubuntu Core or other supported operating systems can be downloaded from the official website of Raspberry Pi. In addition, the device supports Pardus ARM, Arch Linux ARM and Windows 10 IoT Core operating systems. The device can be programmed with the Python programming language as well as with the BBC Basic, C and Perl programming languages (McManus et al., 2017).

Raspberry Pi Models

Raspberry Pi has various models. While these models are basically the same, innovation, speed, etc. they differ in terms of (Monk, 2013).

Raspberry Pi Zero

Raspberry Pi Zero is a Raspberry Pi model that attracts attention due to its low price tag. In addition, it is the smallest Raspberry Pi model among the models. Its hardware (processor and memory) is exactly the same as the Raspberry Pi Model B. Due to its size, it has a micro USB-OTG port instead of a full size USB. Raspberry Pi Zero is half the size of the model A + with twice as many utilities. Raspberry Pi Zero model was shown in Figure 1.

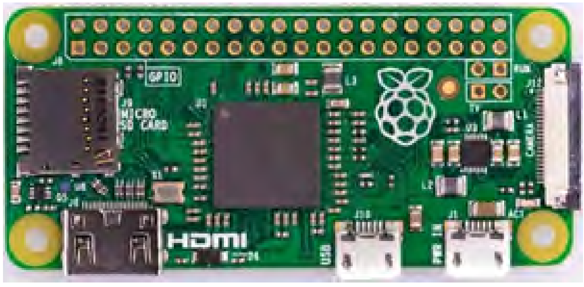


Figure 1. Raspberry Pi Zero

Technical features of Raspberry Pi Zero model;

- 1GHz single-core CPU
- 512MB RAM
- Mini HDMI port
- Micro USB OTG port

- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers
- CSI camera connector (v1.3 only)

Raspberry Pi Zero W

Raspberry Pi Zero W is an extension of the Pi Zero family and was additionally produced with wireless LAN and Bluetooth connectivity. Raspberry Pi Zero W model was shown in Figure 2.

Figure 2. Raspberry Pi Zero W

Launched at the end of February 2017, Pi Zero W has all the functionality of the original Pi Zero but is built with additional connectivity features consisting of the following.

Technical features of Raspberry Pi Zero W model;

- 802.11 b / g / n wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)
- Like the Pi Zero, it also has:
- 1GHz, single-core CPU
- 512MB RAM
- Mini HDMI and USB On-The-Go ports
- Micro USB power
- HAT-compatible 40-pin header
- Composite video and reset headers

- CSI camera connector

Raspberry Pi 1 Model A+

Model A + is a low cost model of the Raspberry Pi and replaced the original Model A in November 2014. It is the updated version of Model A. In this version, the 26-pin GPIO connector has been increased to 40 pins, the composite video output has been removed, and the micro SD card slot is used instead of the normal SD card. It was aimed to be used in projects that may have a lack of space by reducing the card sizes. Raspberry Pi 1 Model A + model was shown in Figure 3.

Figure 3. Raspberry Pi 1 Model A+

Compared to Model A:

- More GPIO pins designed. The 26-pin GPIO connector on Model A has been increased to 40 pins.
- Micro SD. The old friction SD card socket has been replaced with a much nicer push-and-push micro SD version.
- Lower power consumption. Linear regulators have been replaced by switches, reducing power consumption by 0.5W to 1W.
- Better sound. The audio circuitry includes a special low noise power supply.
- Smaller smooth form factor. The USB connector is aligned with the edge of the board, the composite video has been moved to the 3.5mm jack, and 4 square mounted mounting holes have been added. Model A + is approximately 2 cm shorter than Model A.

Raspberry Pi 1 Model B+

Model B + replaced Model B in July 2014 and handed over to Model B model Raspberry Pi 2. Raspberry Pi 1 Model B + model was shown in Figure 4.

Figure 4. Raspberry Pi 1 Model B+

Compared to Model B:

- More GPIO pins. The GPIO header has been increased to 40 pins from the 26-pin GPIO connector on Model A.
- More USB ports. It came with 4 USB 2.0 ports and better plug and play and over current behavior compared to the 2 USB ports on the Model B.
- Micro SD. The old friction SD card socket has been replaced with a much nicer push-and-push micro SD version.
- 100 Base Ethernet (same as original Model B)
- Lower power consumption. By replacing linear regulators with switches, power consumption has been reduced by 0.5 W to 1 W.
- Better sound. The audio circuitry includes a special low noise power supply.
- Smoother form factor. The USB connectors are aligned with the edge of the board, the composite video has been moved to the 3.5mm jack and added 4 square mounting holes.

Raspberry Pi 2 Model B

Raspberry Pi 2 Model B is the second generation Raspberry Pi model. It replaced the original Raspberry Pi 1 Model B + model in February 2015. Raspberry Pi 2 Model B model was shown in Figure 5.

Figure 5. Raspberry Pi 2 Model B

Compared to Raspberry Pi 1:

- 900 MHz quad core ARM Cortex-A7 CPU
- 1 GB RAM
- (Pi 1) Like Model B +, it also has:
- 100 Base Ethernet
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

Raspberry Pi 3 Model B

Raspberry Pi 3 Model B is the oldest model of the third generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016 and was defined as a single board computer with wireless LAN and Bluetooth connectivity. Raspberry Pi 3 Model B model was shown in Figure 6.

Figure 6. Raspberry Pi 3 Model B

Technical specifications;

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1 GB RAM

- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4-Pole stereo output and composite video port
- Full size HDMI
- CSI camera port to connect Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touch screen
- Micro SD port for loading your operating system and storing data
- Switched Micro USB power supply up to 2.5A

Raspberry Pi 3 Model B+

Raspberry Pi 3 Model B + is the latest revision of Raspberry Pi 3 series. Raspberry Pi 3 Model B + model was shown in Figure 7.

Figure 7. Raspberry Pi 3 Model B+

Technical specifications;

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1 GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b / g / n / ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full size HDMI

- 4 USB 2.0 ports
- CSI camera port to connect Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touch screen
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V / 2.5A DC power input
- Power over Ethernet (PoE) support (requires separate PoE HAT)

Raspberry Pi 3 Model A+

Raspberry Pi 3 Model A + extends the Raspberry Pi 3 series to A + board format. Raspberry Pi 3 Model A + model was shown in Figure 8.

Figure 8. Raspberry Pi 3 Model A+

Technical specifications;

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 512 MB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b / g / n / ac wireless LAN, Bluetooth 4.2 / BLE
- Extended 40-pin GPIO header
- Full size HDMI
- Single USB 2.0 ports
- CSI camera port to connect Raspberry Pi Camera Module
- DSI display port for connecting Raspberry Pi Touch Screen
- 4-pole stereo output and composite video port

- Micro SD port for loading your operating system and storing data
- 5V / 2.5A DC power input

Raspberry Pi 4 Model B

Raspberry Pi 4 Model B model is the latest version in the market for now. Raspberry Pi 4 Model B model was shown in Figure 9 (Sanchez, 2019).

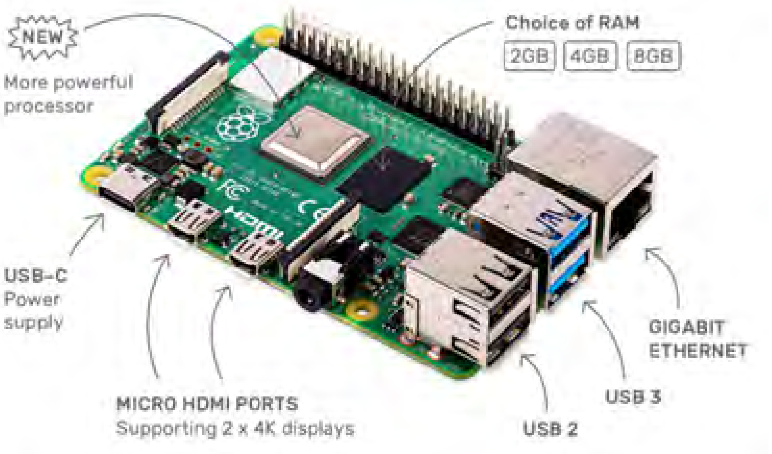


Figure 9. Raspberry Pi 4 Model B

Technical specifications;

- 1.5 GHz quad core ARM Cortex-A72 CPU 64-bit SoC Broadcom 2711
- 4Gb LPDDR4 RAM
- 4kp60 HEVC video
- VideoCore VI Graphics
- USB-C power input supporting 5V-3A operation
- True Gigabit Ethernet
- 2 micro HDMI connection ports 4K video (1 × 4K @ 60Hz or 2 × 4K @ 30Hz)
- 2 USB 3.0 and 2 × USB 2.0 ports
- Bluetooth 5.0 BLE
- Dimensions: 68.63 x 94.09 x 26.63 mm

Raspberry Pi OS

After choosing the Raspberry Pi model, you need to decide which operating system or systems will be run on the computers. Raspberry Pi models can run a wide variety of operating systems, from the customized Raspbian operating system provided by the Raspberry Pi Foundation to media center operating systems and custom RISC operating systems. Here, we will examine the operating systems that can be used in Raspberry Pi models (Pajankar et al., 2016).

Raspberry Pi Foundation provides an installer called NOOBS (short for New Out Of Box Software) to install one or more operating systems on Raspberry Pi computers. The NOOBS installer supports all four operating systems:

- Raspbian
- LibreELEC
- OSMC
- Windows 10 IoT Core

Raspbian

Raspbian is the official operating system of the Raspberry Pi microcomputer. This operating system was based on the Linux distribution Debian and includes features that allow us to make the most of the features of the microcomputer. Raspbian was optimized to run on ARM computers and comes with a large number of packages and applications. Raspberry Pi Operating System uses LXDE based PIXEL (Pi Enhanced X-Window Environment, Lightweight) desktop which is both useful and lightweight. Interface view of Raspbian operating system was shown in Figure 10.

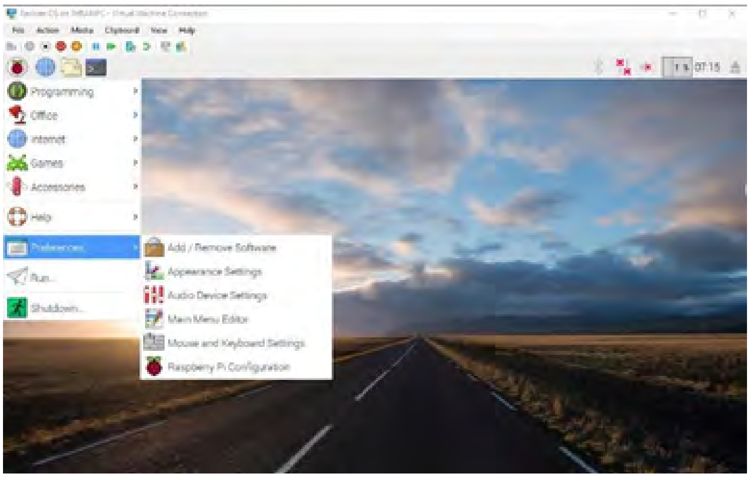


Figure 10. Raspbian Interface

Raspbian Lite

Raspbian Lite is a lightweight version of Raspbian. Raspbian Lite doesn't include a GUI, so it boots to the command line. Raspbian Lite can be used to get the best performance from an older Raspberry Pi computer. Raspbian Lite also doesn't include any apps, so the installation takes much less space than regular Raspbian.

RISC OS Pi

RISC OS Pi was designed for users who want to open applications individually. This may seem like a limitation, especially if we are used to using many programs at the same time, but it should be noted that the RISC OS Pi occupies only 16MB of memory and has a very simple learning curve.

Lakka

Lakka offers a Linux distribution specifically designed for us to run RetroArch emulators. It is one of the best distributions for retrogaming that we can install on microcomputer.

RetroPie

RetroPie is an operating system similar to the Lakka operating system and aims to transform our microcomputer into a Retro console. It is compatible with all versions of Raspberry Pi. This operating system can run under the full version of Raspbian. Therefore, besides being able to use all RetroArch emulators, we can also run any program on the Linux operating system.

Recalbox

Recalbox is another distribution of Raspberry Pi designed for Retrogaming. This distribution is located between Lakka and RetroPie. This distribution uses the Linux operating system.

Ubuntu Retro Remix

Ubuntu Retro Remix is actually a script that allows us to turn Ubuntu into a game distribution for the Raspberry Pi.

LibreELEC

LibreELEC, short for Libre Embedded Linux Entertainment Center, is a fully featured media center application running on Raspberry Pi cards. LibreELEC runs the Kodi media center, which provides quick access by dividing the home screen into five main categories such as Pictures, Videos, Music, Programs and Settings.

OSMC

OSMC (Open Source Media Center) is a customized version of the Debian Linux distribution to use the Kodi media center as its main application. When you start OSMC on the Raspberry Pi computer, you will see Kodi as the graphical interface.

Windows 10 IoT Core

Windows 10 IoT Core is a lightweight version of Windows 10 designed for developers, hackers, and hobbyists who want to use Raspberry Pi cards as a device to prototype IoT devices. Windows 10 IoT Core is almost completely different from full versions of Windows 10 running on tablets, laptops, desktops, and servers.

Karmbian OS

Kali Linux is an ethical hacking distribution for Karmbian OS, Raspberry Pi and other microcomputers.

OpenMediaVault

NAS is a device with which we can connect to the network and enable certain services (especially storage) to all computers on the network. While commercial NAS can be quite expensive, we can transform the Raspberry Pi into a fully functional NAS. This distribution allows us to enable or disable the processes and services we need and use certain protocols within the network. You can easily manage remotely from your web panel. It also supports RAID configurations.

Alpine Linux

Alpine is a very minimalist and standalone distribution that focuses on users building secure networks and infrastructures. It is a very small and simple distribution, does not take up more than 50 MB.

Lubuntu

Lubuntu is a lightweight Linux distribution that uses the LXDE desktop environment, which is the desktop environment Raspbian uses by default. Works on Lubuntu, Raspberry Pi 2 Model B, Raspberry Pi 3 Model B and later Raspberry Pi boards. However, it does not work on Raspberry Pi boards previously produced. Lubuntu typically uses about half the RAM of Xubuntu, so it's a good choice to get good performance on a full-featured version of Linux.

Xubuntu

Xubuntu is a version of Ubuntu Linux that uses the Xfce desktop environment instead of the Unity graphics shell that regular Ubuntu uses. As Xubuntu is a relatively demanding version of Linux, it is a better choice for the Raspberry Pi 3 Model B than the Raspberry Pi 2 Model B.

Ubuntu MATE

Ubuntu MATE is an official derivative of the widely used Ubuntu version of Linux. Ubuntu MATE differs from regular Ubuntu in that it uses the MATE desktop environment instead of the Unity graphics shell. Apart from this big difference, Ubuntu MATE provides a similar range of software to Raspbian and runs at similar speed.

RaspBSD, FreeBSD, and NetBSD

FreeBSD and NetBSD are two Unix-like operating systems derived from Berkeley Software Distribution (BSD), a derivative of Research Unix. Both are open source operating systems preferred by IT professionals for reliability and security.

Raspbery Pi Programming

Programming languages in Raspberry Pi operating system were examined on the Raspbian operating system (Monk, 2013).

Programming with Scratch

The Raspbian operating system includes the Scratch programming tool to teach the basics of programming, especially for elementary school children who are just beginning to program. The Scratch application was developed by the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) Media Lab. You can use the Scratch application as the first step to understanding programming basics without considering programming syntax constraints. The Scratch interface was shown in Figure 11.

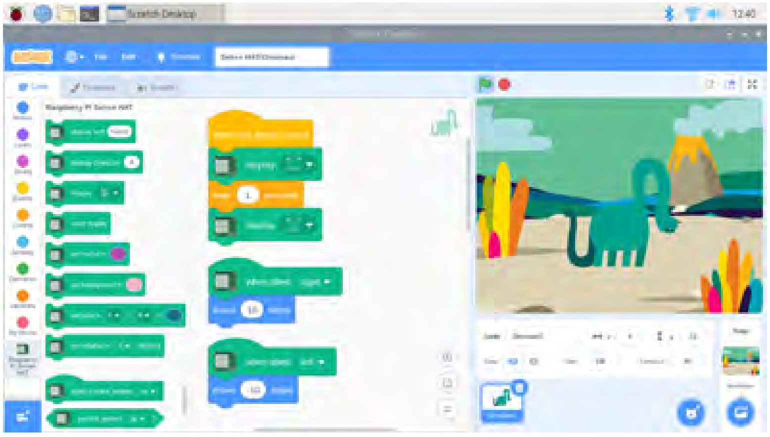
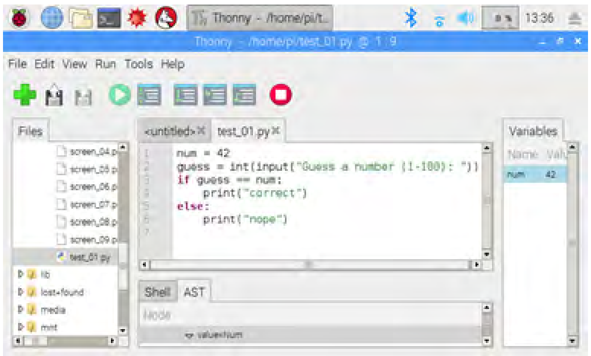


Figure 11. Scratch 3 Desktop for Raspbian

Programming in Python

The Raspbian operating system includes development environment (IDE) applications of the Python programming language to develop web applications, small databases and 2D games. Includes IDLE (Integrated Development and Learning Environment) IDE versions for Raspbian, Python 2 and Python 3. Figure 12 shows a Python language implementation in the IDLE IDE.



```
num = 42
guess = int(input("Guess a number (1-100): "))
if guess == num:
    print("correct")
else:
    print("nope")
```

Figure 12. Raspberry Pi Python IDE

Programming in Java

The Raspbian operating system includes BlueJ and Greenfoot applications for the use of the Java programming language. BlueJ is an editor developed for creating Java coding and working on objects. Figure 13 shows an example BlueJ interface (Bluej IDE, 2021).

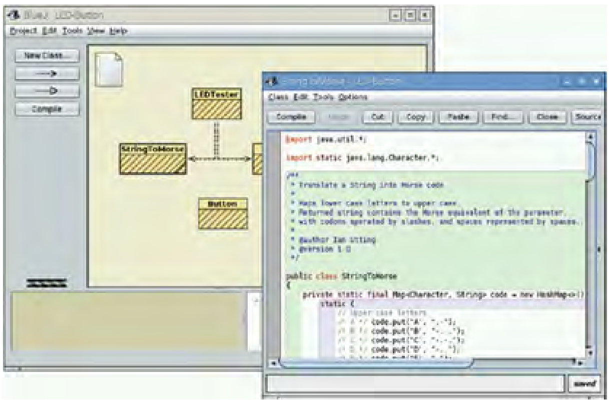


Figure 13. Bluej IDE

Greenfoot is an IDE application developed to create a number of Java scenarios. One of the Greenfoot sample applications was shown in Figure 14 (Greenfoot IDE, 2021).

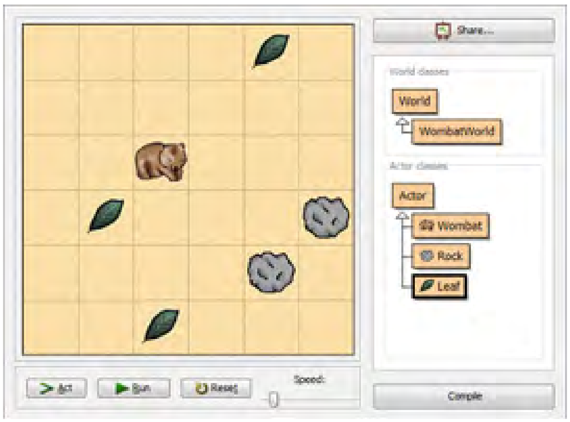


Figure 14. Greenfoot IDE

Programming in Other Languages

Includes Geany IDE for implementing programming languages such as Raspbian, C, PHP (PHP Hypertext Preprocessor), HTML, CSS (Cascading Style Sheets) and Perl. Geany has an accessible user interface. It includes features such as automatic completion of application programming terms and tags, syntax highlighting, and multiple document support. Geany IDE application was shown in Figure 15 (Geany IDE, 2021).

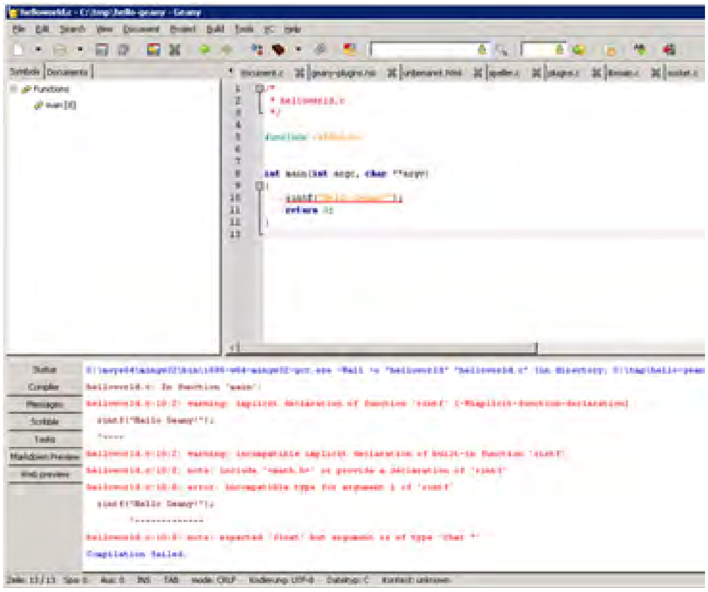


Figure 15. Geany IDE

References

Butuner, R. (2019). Effect of Coding and Robotic Coding Training on Students. Journal of Information Systems and Management Research, 24-30

Guy, H.D. (2017). Deploying Raspberry Pi in the Classroom. Apress, Durham, United Kingdom.

Vijayan, A. & Shaji, L. (2015). Raspberry Pi a Tiny Computer. International Journal of Engineering Research & Technology (IJERT), 3 (28), 1-3.

Sanchez, E. (2019). Raspberry Pi 4 User Guide.

Monk, S. (2013). Programming the Raspberry Pi, McGraw-Hill.

Pajankar, A. & Kakkar, A. (2016). Raspberry Pi By Example, Packt Publishing

McManus, S. & Cook, M. (2017). Raspberry Pi for Dummies, John Wiley & Sons

Raspberry Pi. (https://www.raspberrypi.org) , Accessed Date: 05.07.2021

SparkFun Electronics, (https://www.sparkfun.com), Accessed Date: 05.07.2021

Greenfoot IDE, (https://www.greenfoot.org), Accessed Date: 05.07.2021

Geany IDE, (https://www.geany.org), Accessed Date: 05.07.2021

Bluej IDE, (https://www.bluej.org)/, Accessed Date: 05.07.2021

About the Authors

Yusuf UZUN, PhD, is an Assistant Professor of Computer Engineering at Necmettin Erbakan University in Konya, Turkey. He holds a PhD in Mechanical Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, autonomous systems and augmented reality applications. He also works as the Rector's Advisor at Selcuk University.

E-mail: yuzun@erbakan.edu.tr, Orcid: 0000-0002-7061-8784.

Ozgur DUNDAR, PhD, works at Necmettin Erbakan University, Department of Space and Satellite Engineering. He graduated from the Electrical and Electronics Engineering Department of Selcuk University. He worked as an Automation Engineer for a while. His master's and doctorate degrees are from Selçuk University, Institute of Science and Technology, Department of Electrical and Electronics Engineering. Special fields of study are Automation, Robotic, Communication, Electromagnetic and Micro Strip Patch Antenna designs.

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 20%.

To Cite This Chapter:

Uzun, Y. & Dundar, O. (2021). Raspberry Pi and Models. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 34–49). ISRES Publishing.

Programmable Logic Controllers (PLC)

Ozgur DUNDAR
Necmettin Erbakan University

Sabri KOCER
Necmettin Erbakan University

Introduction

Today, it has become compulsory to compete by reducing the cost by increasing the production quality and speed in almost all sectors at the global level. This situation has revealed the concepts of automation or automatic control with the developing technology.

Automation provides automatic execution of human-mediated processes in the industrial field, management, informatics sector and all jobs where control operations can be performed. Automation means performing production or controlling systems using mechanical, electronic and computer-based systems. The main purpose in industrial automation is to minimize human errors by minimizing the manpower spent in production and to maximize productivity in production. As a result of automation, human needs in the production environment will be eliminated and the production speed will be increased.

Today, industrial automation systems are high quality, safe and independent from human influence, and they have become more competitive by increasing production efficiency and speed. Programmable Logic Controllers (PLC) is control elements that are widely used in automation systems today and form the basis of the system. PLCs have the ability to easily control even very complex systems, thanks to the fact that they can be easily programmed with computers or input panels and the program can be changed at any time.

General Motors company designed a computer-based, easily programmable and maintainable control system that could be used instead of relay systems in 1968. This device, defined as PLC, was a system that could only control open-closed contacts. With the replacement of discrete elements with integrated circuits in the 1970s and the development of microprocessor technology of 1974, PLCs using microprocessors as the basic element emerged. PLCs using microprocessors have expanded their application areas as they have the ability to perform more complex and flexible operations. PLCs, there were improvements in the hardware structure by increasing the high memory capacity and analog control until 1980. The high memory capacity enabled larger application programs to be run under changing conditions, and the use of built-in Analog-Digital and Digital-Analog converters enabled analog signals to be processed within the PLC.

The high memory capacity enabled larger application programs to be run under changing conditions, and the use of built-in Analog-Digital and Digital-Analog converters enabled analog signals to be processed within the PLC. As a result, languages that can easily provide command control in analog control have been developed. With the development of Bit-Slice technology in the 1980s, rapid scanning of the program in the memory of the PLC was provided. In these years, it has been realized that PLCs can exchange information among themselves with a common BUS system. This development has allowed control systems to be configured and programmed with great flexibility. With the addition of arithmetic processing capabilities to PLCs, these devices can also be used in feedback control systems.

Today, when automation is mentioned, control and follow-up come to mind. PLCs make the control part of the production, and Supervisory Control And Data Acquisition (SCADA) programs, which are interface programs, make the monitoring part. Performing data collection and control operations from a remote point with SCADA of the system, which works in line with the program loaded on the PLC, remote monitoring of industrial systems and rapid and effective control have made production processes more reliable and more efficient. With the effective use of the Internet today, PLC systems can be monitored much more remotely, intervening in the program at any time and system development can be done. It is in the form of collecting and storing the data obtained from the PLC on the server computer, and client computers accessing this server over the Internet to query, monitor and report the data.

Firstly, PLCs, Allen-Bradley PLCs, Texas, Square PLCs, which started their journey with Modikon PLCs specially produced for the General Motors Company, followed. Today, although PLCs produced by companies such as Hitachi, Siemens, Festo, Idec, Mitsubishi, Moller, Aeg, Omron, Toshiba, Telemecanique, Micron, Taian, General Electric are popular, many companies have offered solutions to the industrial environment by making PLC designs with various features. Various PLC examples are shown in Figure 1.

Figure1: PLC Types

It seems that control technology with PLC will continue to develop rapidly. This results in the use of industrial automation circuits of PLC-based control systems as an indispensable system and updating them with new features every day.

Hardware Structure of PLC

PLCs are devices that have advanced features of heavy working conditions such as shaky, humid, dusty and noisy industrial environments, and can operate in environments with 0-60 °C ambient temperatures and humidity between 0% and 95% (Kurtalan, 2001).

As seen in Figure 2, a PLC system; It consists of the Central Processing Unit (CPU), Input and Output (I/O) interfaces, power supply and programming unit.

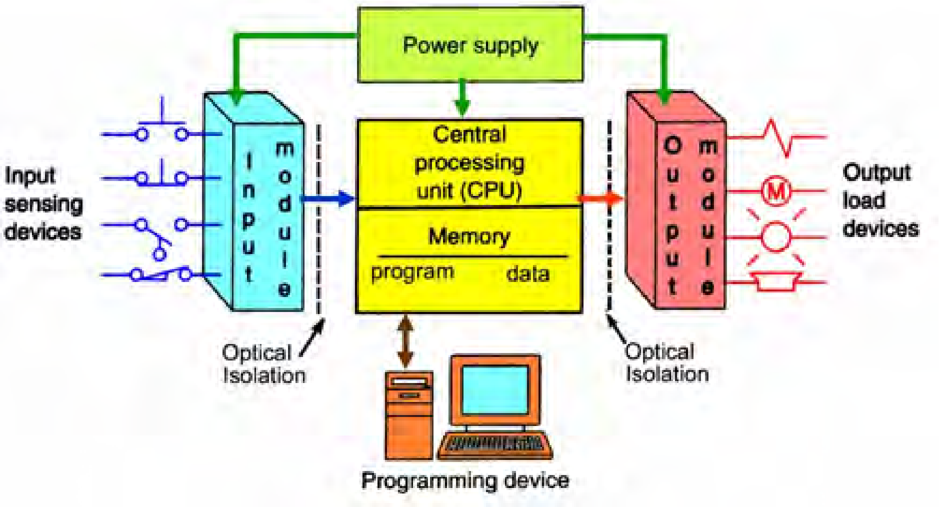


Figure 2: PLC System

As the operation of the system is shown in the block diagram, PLC is a microprocessor system that processes the information it receives from the inputs according to the programming and transfers the state resulting from the operation to the outputs. The power source is the unit that gives the power to operate the system elements.

Central Processing Unit (CPU)

Central processing units, together with the memory, are the part that gives processing capability to a PLC, and can be considered as the brain of the PLC system, which performs all the arithmetic, logic and data processing functions required by the program stored in the memory. The operation of the microprocessor is managed by a program system called the operating system in PLCs. The operating system is a PLC management software prepared by the manufacturer in a way that does not allow user intervention (Özcan and Özkan, 2004).

The CPU performs all the operations via a microprocessor. Today, these operations can be performed by more than one microprocessor. Usually, one of these microprocessors handles logic and the other handles control. The logic microprocessor performs a scan determined by the logic of the application program, along with operations such as timing, logic, and counting. The control microprocessor is the microprocessor that solves the equations in the control loop and performs more complex data and calculation operations such as interacting with operators.

During the execution of the program entered in the PLC, the CPU reads all the inputs and controls the output terminals by performing the operations required by the control logic. In the CPU, scanning is normally a continuous cycle of reading the state of the inputs, performing the control logic, and setting the outputs (Petruzella, 1995).

The CPU performs the next action, constantly browsing the program stored in memory to look for instructions and reference data. In addition, the CPU stores the data onto the inputs in memory for future use or makes use of the data onto memory for intermediate processing during some decision making .

PLC memory usually consists of Random Access Memory (RAM) and Read Only Memory (ROM). RAM memory is used for general program work. From here information can be read or written for temporary storage. ROM is a type of fixed memory and system information written to this memory once during production remains constant for the life of the device . PLC can only read information on ROM memory during its operation .

The selection of the CPU to be used in the PLC system is important. In order for it to perform its desired function properly, its processing speed, memory capacity and specific features must meet the minimum requirements of the process. The more powerful the CPU, the wider the user program that can be stored, and the shorter the processing of this program will be.

Input Output (I/O) Interfaces

Information from various field measuring devices of the controlled system (such as pressure, level, temperature) or signals to sensors are received via the input unit. The data processed by the processing unit is sent to the outputs where the machine control will be made via the output unit. In PLCs, we can generally divide the input-output interfaces into digital and analog I/O. However, parallel and serial inputs and outputs are also available in the system.

Digital Input Module

PLC can detect the status of the controlled object with the signal coming from this object to the digital input. For this, the contact that expresses the situation must be transmitting or not transmitting this voltage. Each input signal has two separate states. These are the cases based on the logic of yes-no, carrying the contact open-closed information and evaluated as “logic 1” and “logic 0”. (Özcan and Özkan, 2004).

Elements used as a source for PLC’s digital inputs: State Switches, Thumbwell Switch, Instant Contact-Push Type Buttons, Photoelectric Eyes (Sensors), Limit Switches, Proximity Switches, Liquid Level Switches (Floating Flow Switch), Contactor Contacts, Relay Contacts, Pressure Switches. The input signal is the non-PLC digital signal and in input modules they reduce it to the internal signal level of the PLC. 8, 16 or 32 bit digital field information can be read in a single input modules. There is an LED for each input on the modules and the level of the incoming signal can be understood from here. In order for the PLC to be able to read the input signals, these signals must be in the relevant range according to the type of card (Çalışkan, 2007).

Voltage levels that can be applied directly to the input module in accordance with the standard: 24 V, 48 V, 100 V- 120 V, 200 V- 240 V can be direct or DC or AC. A signal coming to PLC input circuit has a lower limit where logic-1 can be accepted and an upper limit value where logic-0 can be accepted. In order for the input information to be detected correctly, the signal voltage level must be between these values. The mentioned limit values can be learned from the product catalog of the company that produces the PLC. For example; In a digital input circuit taken from the manufacturer’s catalog, the voltage level of the incoming signal may need to be a minimum of 120 V to be accepted as “1”, and the voltage level to be considered to be “0” may need to be a maximum of 40 V (Özcan and Özkan, 2004).

Digital Output Module

The digital output module does the opposite of the input module. Digital output units are the units where PLC is used when digitally interfering with an element such as contactor, relay, solenoid in the field or in the process. Digital output modules are components that convert PLC internal signal levels to digital signal levels required by the process. By setting an output over these modules, any element in the field or in the control panel can be controlled. Elements that PLC drives as digital output: Control Relays, Alarms, Fans, Signal Lamps, Alarm Horns, Solenoid Valve, Contactors, Solenoids, Stepper Motor Control Units, Indicators. The digital output module can be expanded to create new outputs as needed.

The digital output module can be relay, triac or transistor output. Outputs with transistors in direct current and triacs in alternating current are preferred, especially in situations where high speed on-off are required such as driving a stepper motor during operation. A maximum current of 10-16 A can be drawn from the output module, but not large currents. The maximum current values of the output are learned from the manufacturer’s user manuals.

Since the output stages of PLCs are directly exposed to external influences such as high voltage and noise, they are the areas where the control system is the most sensitive and can be damaged. In the output stage, transistor or thyristor is used for DC switching and triac is used for AC switching. The thyristor in the output module turns on in most cases by applying a positive pulse to the gate terminal relative to the cathode on the microprocessor side. When a voltage value exceeding the reverse breakdown voltage is applied between the cathode and the anode of the thyristor, it may cause the thyristor to turn on even though the PLC does not send the turn-on signal to the digital output. This situation needs to be taken into account when realizing the application circuits.

A parallel varistor is connected to the thyristor, preventing the thyristor from turning on due to overvoltage or line transients. The varistor also prevents damage to the thyristor due to over voltage that may occur during transmission. If the voltage applied to the anode suddenly changes due to the rapid voltage change in the joint capacitance of the thyristor, the thyristor may turn off and turn on. In order to eliminate this negativity at the determined operating voltage, a series resistor-capacitor circuit called a stopper can be connected to parallel with the thyristor.

Analog Input Module

Analog input modules take analog currents and voltages from analog inputs and convert them into digital information form via an Analog to Digital Converters (ADC). Here, the conversion levels are expressed as a 12-bit binary or 3-bit BCD coded value proportional to the analog signal. Analog sensor elements are elements such as humidity sensors, thermocouples, potentiometers, flow sensors, pressure sensors that display temperature, light, speed, pressure, humidity, current, voltage data. All these sensors can be connected to the analog input.

An analog input module can be used for multiple data inputs with analog multiplexers. Generally, filtering and limiting operations can also be done in analog multiplexers. AC signals that are not at a certain level of limitation are prevented from reaching the A/D converter. Electrical isolation is done between the CPU and the analog input module. The timing and control processes within the module separately control the channel selection and the writing of the input data to the buffer during the CPU’s scan period. In this way, the temporal operation of write-to-memory processes and processes of reading data from memory by the CPU is prevented. Calibration of the A/D converter can be done by adjusting a reference voltage generated within the analog output module.

Analog Output Module

Analog output modules are used in situations where analog intervention is required to the controlled system. With these modules, an element in the field can be controlled proportionally with 0-10 V, 0-20 mA or 4-20 mA outputs. The actuator, which is an operator mechanism that provides automation of the valve's opening/closing process, can be managed with the analog outputs of the PLC. Output values decided by the CPU as a result of processing the input values are transmitted to the processor of the analog output card in digital form. These values are converted to analog voltage values by means of a digital-to-analog converter (ADC). In addition, output currents are generated with a voltage-current converter.

Signals that will affect analog elements appear as codes in the PLC. Since these signals have different values, the codes representing them must also be multi-bit. Thus, in addition to bit-based interfaces, word-based interfaces with multi-bit binary codes have also emerged. Interfaces usually output these codes in parallel. The analog output module converts these codes into analog current or voltage signals.

The code length used is usually a maximum of 10 bits. A typical output module can generally process and transmit 4 separate data. As with the analog input, precautions have been taken against writing and reading at the same time. The output of these modules can be DC current or voltage. Modules that provide 4-20 mA current for the current module and 0-10 V voltage for the voltage output are commonly used.

Parallel Input Output Communication Link

Communication input modules connect the CPU to multi-bit output elements in parallel format. A typical Parallel Input Output Communication Link module consists of 4 separate channels of 16 bits each. Each of these elements is controlled by a three-state semiconductor electronic gate with selector input. Otherwise, each input must be connected to a separate module.

As in bitwise I/O, the inputs are kept in a buffer register, passing through a filter and isolator under the control of the timing signals generated by the interface itself. The contents of this register are constantly refreshed regardless of the CPU's scanning rate. The delay of reads that may be made by the CPU while refreshing the buffer register is provided by the synchronization system.

An analog multiplexer is used in the parallel output module. When a multiplexer is used, the output element must be of the type that can receive information with a timer signal.

Serial Input Output Link

Serial Input Output Link interfaces to provide the connection between the elements with serial format data output or input with the CPU. An example of a serial input/output interface is spindle encoders. In the pulse sequence produced by such an encoder, each pulse must be counted without being missed. Therefore, there is a counter that can count fast pulses of the interface. The pulse frequency to enter this counter is 100 Hz. with 50 kHzs. may vary between The minimum pulse width should be between 10-20 ms.

Communication between the interface and the CPU is bidirectional. The module accepts default and other control data onto the CPU. It reports the counted value and the states of the index sign of the CPU. The CPU activates and resets the counter according to the application program.

Power Source

The power supply plays the main role in the operation of the whole system and is responsible for supplying the cards in the PLC, except for the I/O cards. Power supply in PLCs consists of two parts: Battery or Battery power supply, and voltage source fed from the mains.

Battery or cordless power supply ensures that the RAMs are fed uninterruptedly when the system is de-energized, preventing permanent application programs from being deleted, and the user program, permanent markers, counter and timer contents inside the CPU can be protected against voltage interruption (Webb and Reis, 2002).

On the other hand, the power source fed from the mains starts to be fed from the source when the PLC is connected to the supply voltage from the mains, and by reducing the supply voltage from the mains to the internal voltage levels of the PLC, the maximum output current of the source provides different levels of DC and AC voltages according to the power consumption of the cards in the PLC.

If the memory backup battery inside the power supply and this backup battery is to be replaced when there is no power, the power supply must be supplied from an external source.

Programming Units

The Programming Unit provides the link between the programmer and the PLC. By planning the necessary control operations and transferring them to the program format specific to the selected PLC, the programmer can enter the information via a programming tool into the PLC using the symbols, letters and numbers that are standard for a particular version of the programming language. The programming tool can be a microprocessor-based special handheld programming device, or it can be a personal computer or a tablet-shaped handheld computer. Figure 3 shows the programming of the PLC on a personal computer. This unit, during the writing of the program, transferring it to the PLC and when necessary, during the operation of the PLC; It provides the possibility to observe the input/output memory states or to change some parameters.

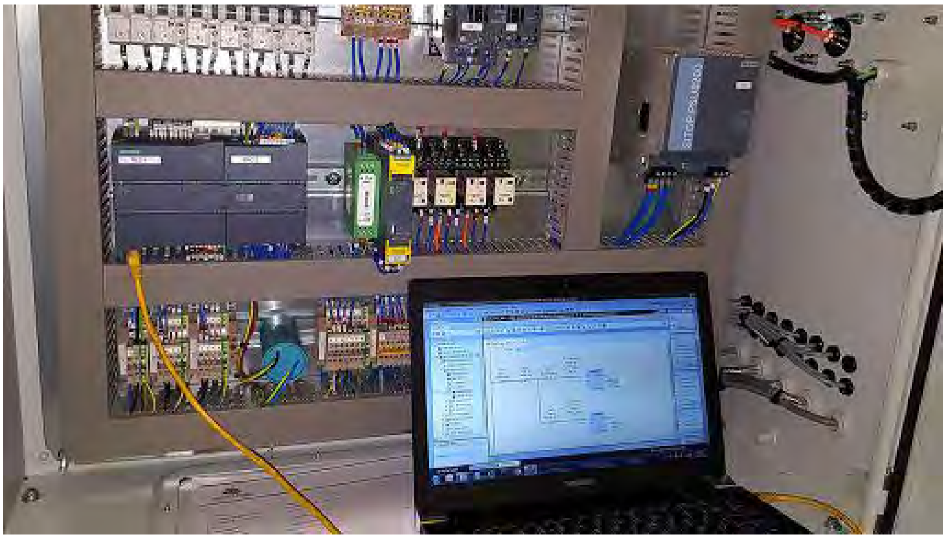


Figure 3: Programming the PLC

Although personal computers are more commonly used to program PLCs today, PLC programming can be carried out in a WEB-based internet environment through remote sensing systems. PLCs can be programmed easily with the help of an editor compiler to be installed on any computer. PLC manufacturer companies have developed editor compiler programs that users can easily use and belong to their companies.

Communication Unit

ASCII input/output module is generally used in the communication unit. The ASCII input/output module provides alphanumeric data exchange between the CPU and external units. The module usually consists of a RAM buffer and a microprocessor, together with all necessary communication units. Generally, this module is equipped with either RS 232C or RS 422 and one of two serial communication ports that allow a loop current of 20 mA. This module transfers the block information stored in its memory. First, when an information is entered from the device to the input, the data bus speed of the information to be transmitted to the module via the programmer is transferred in this module (Özcan and Özkan, 2004). All start parameters can be selected by parity coding, stop bit number, communication speed, software or hardware.

Remote Monitoring of Data Obtained from PLC

In addition to hardware, the system also needs software in order to collect, manage and control data by providing remote access to PLC systems. The fact that this software works on the Internet for remote access and is WEB-based ensures that the application can be accessed from anywhere. In addition, it can monitor data, query and receive various reports without installing any access program on Server computers. The general structure of the remote access system to the PLC is shown in Figure 4.

The server in the system can serve as both Application Server and Database Server. Therefore, since it is an Application Server, it can contain all program codes and screen structures in the application, and it can store all data in the system and all kinds of communication between them, as it is a database server (Birant and Kut, 2018).

When it is desired to query or monitor data in the system, client computers first reach the server over the internet and transmit user requests. The server, on the other hand, runs the modules suitable for these requests and transmits the data obtained from the PLC system back to the client computer. PLCs in the system, on the other hand, can transmit the data they obtain periodically to the server computer and ensure that the data is stored in the database.

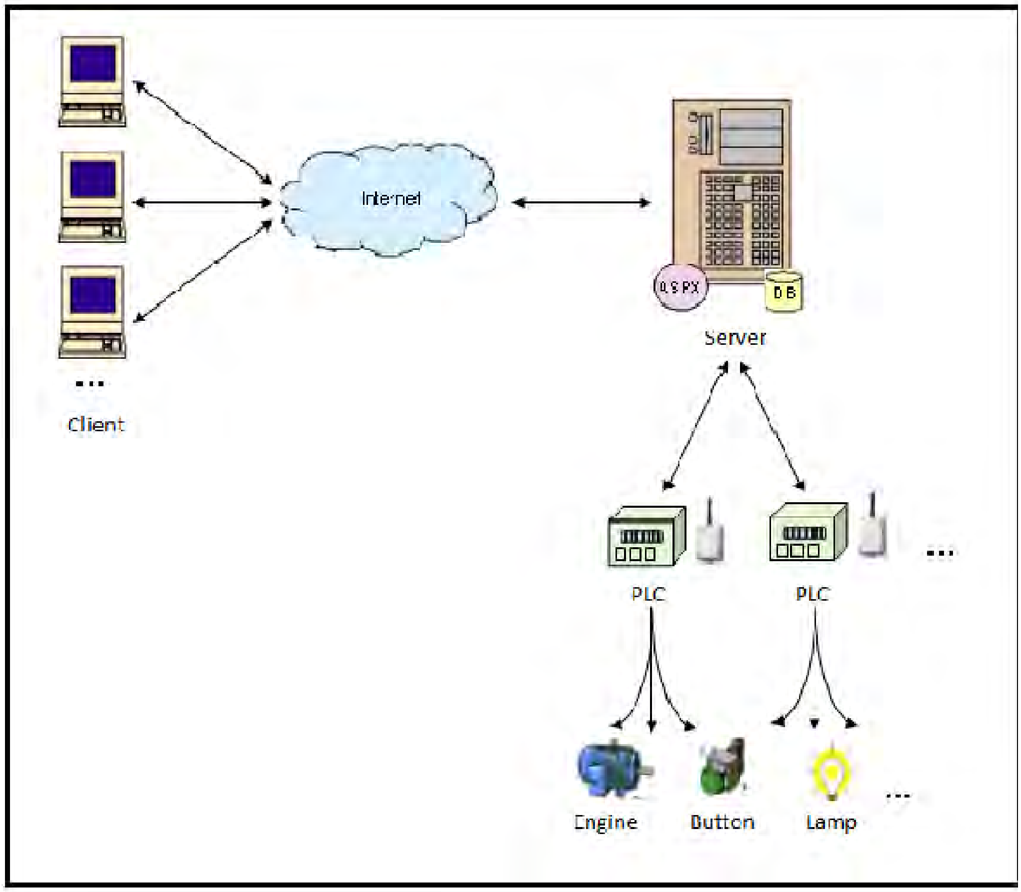


Figure 4: General Structure of the Remote Access System to the PLC

Programming the PLC

PLC programs can be prepared in one of several specially designed languages. We can examine these programming languages in two groups. The first group is Ladder Diagram and Boolean expressions as basic and common programming languages, and the second group is Function Blocks and Sequential Function Maps, which are high-level languages. In addition, programming techniques can be divided into two groups as Linear programming and Structured programming depending on how they are written

(Crispin,1996).

In linear programming, all statements are written sequentially, and all statements present in the program during a cycle are processed in the order in which they were written. In structured programming, each control operation is created independently in blocks and an execution program is written to determine which block or blocks will be executed in each cycle.

Classification of PLC Commands

Relay type commands, Timing and counting commands, Data transfer commands, Arithmetic operation commands, Data comparison commands, Program flow control commands are used in programming PLCs. In programming methods using ladder diagrams, only relay logic, timing and counting commands are used. In high-level languages, they contain large sets of commands for all the command groups listed above.

Relay Type Commands

Relay type commands contain the commands necessary to detect the status of the contact inputs connected to the input module from the outside and to control the output modules.

Normally Open Contact, NO (—| |—) : This command is used if an output is to be activated by closing a contact.

Normally Closed Contact, NC (—| / |—): It functions in the opposite way to a normally open contact. This command is used if an output is to be disabled by opening a contact.

Energize Coil (—( )—) : It is used to manage an internal element connected to the PLC, an internal output or an output located in the output module.

DE-Energize Coil (—(/)—) : It is used to manage an external element or an internal output connected to the PLC. If there is logic flow in the stair step, the specified output is disabled.

Latch Coil (—(L)— ; —(SET)—) : This command is used if an output is turned on once and it is desired to remain on even if the state of the contacts providing it is changed.

Unlatch Coil (—(U)— ; —(RES)—) : This command is used to remove an output that is locked to transmit from transmitting. If the logic flow is continuous in the ladder step, the transmission of the specified address is interrupted.

An example Ladder diagram using Relay type commands is given in Figure 5. In the Ladder diagram in Figure 5; In step 1, if one or both of the inputs 101 and 102 are transmitting and the input 103 is also transmitting, output 201 is transmitted. In this step, 101 and 102 form the logic “OR” gate. When the contact connected to address 201 and 104 on the 2nd step of the ladder is closed, the digital output 202 is de-energized. In the 3rd step, output 203 will be in transmission as input 105 is in transmission. Even if input 105 is no longer transmitting, output 203 will remain transmitting. If 106 is transmitting and 203 is transmitting, it will be de-energized (Özcan and Özkan, 2004).

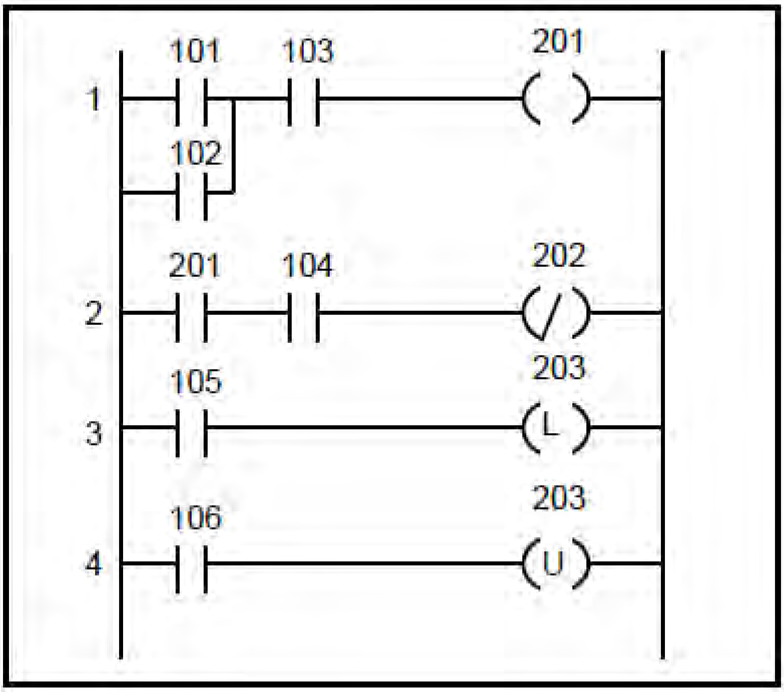


Figure 5: An Example Ladder Diagram Using Relay Type Commands

Timing and Counting Commands

These commands are used to turn an output on or off after a certain time or number of events. Basically, the counting operation and the timing operation are the same, both of them perform the counting operation.

Time Delay Energize (—(TON)—) : If there is logic continuity in a step, the timer starts counting with a certain time unit and the output is activated when the number reaches the specified value.

Time Delay De-energize (—(TOF)—) : Performs the interruption of an output from transmission with a certain time delay.

Retentive on Delay Timer (—(RTO)—) : This command is used if it is necessary to store logic continuity or the time that has elapsed when the supply is interrupted.

Retentive Timer Reset (—(RTR)—) : It is a special command used to reset the RTO. Resets the contents of the specified register register if the digit has logic continuity.

Up-Counter (—(CTU)—) : This command increments the contents of the specified register each time the logic continuity of the corresponding digit is interrupted and reopened.

Down Counter (—(CTD)—) : Counting is done down from the predicted number. When it reaches zero, the output turns on.

Counter Reset (—(CTR)—) : CTR command is used to reset counter content in CTU and CTD counters.

An example Ladder diagram using Count type commands is given in Figure 6. In some types of modular type PLCs, two addresses can be used for digital inputs and digital outputs.

If the 1st and 2nd Inputs of Module 101 of the Ladder diagram in Figure 6 are closed, the content of the up counter (CTU1) increases by one each time this operation occurs. If the content of this counter reaches the default value of 5, this counter closes its 3rd step open contact (CT1) and activates the 203/01 output. 2. In the ladder step, when the 101/01 contact and the 102/01 contact are both closed, the content of the counter decreases by one. When the 106/03 contact is closed, the content of the counter is reset (Özcan and Özkan, 2004).

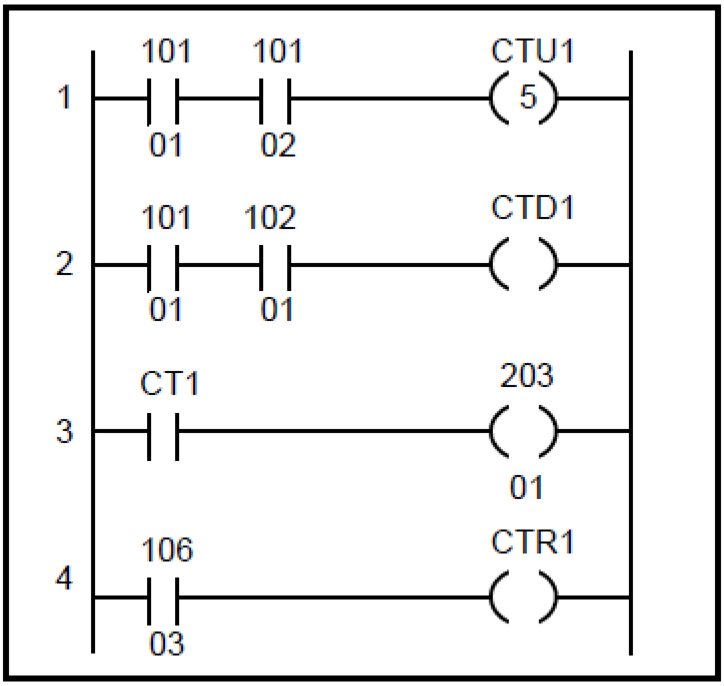


Figure 6: An example Ladder Diagram Using Counting type Commands

Data Transfer Commands

Get Word (—| GET |—) : This command takes the contents of the specified address and keeps it ready for further processing.

Put Word (—(PUT)—) : Puts the result of operations performed in a digit to the specified address.

Arithmetic Operation Commands

ADD (—(+)—) : Collects two data at the specified memory address.

SUB (—(-)—) : Performs subtraction of two data at the specified memory address.

MUL (—(*)—(*)—) : It takes the two data received with the GET command, multiplies it, and places the result at the specified memory address.

DIV (—(:)—(:)—) : It takes two data received with the GET command, performs division, and the integer part of the result is placed at the first address specified, and the fraction part is placed at the second address.

An example Ladder diagram about data transfer and arithmetic operation commands is given in Figure 7.

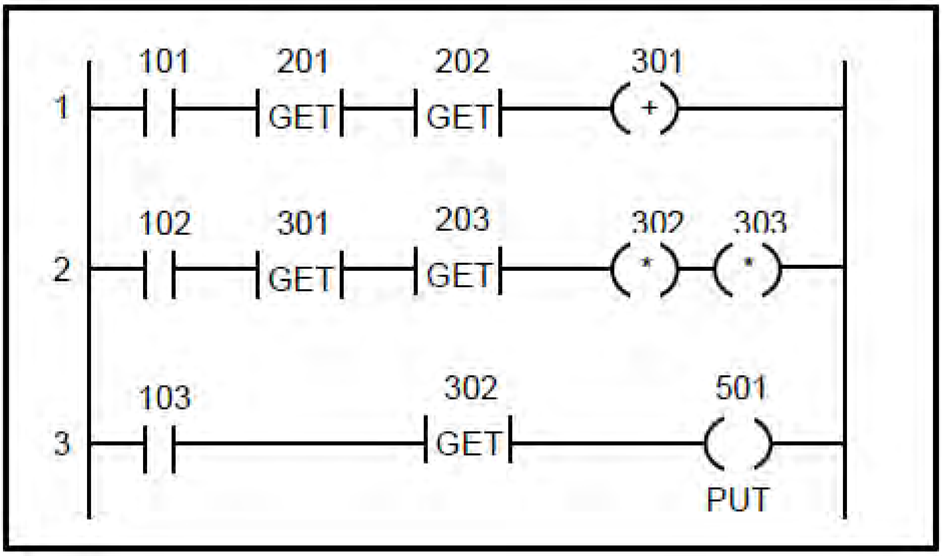


Figure 7: An Example Ladder Diagram About Data Transfer and Arithmetic Operation Commands

In the first step, if the contact number 101 is closed, the data in cells 201 and 202 are collected and the result is saved to 301 in the memory. In the second step, in 102 transmissions, the sum stored in 301 is multiplied with the data in 203 and the result is saved in memory compartments 302 and 303 as two bytes. In the last digit, in 103 transmission, the significant bits of the multiplication result are transferred to address 501 (Özcan and Özkan, 2004).

Data Comparison Operation Commands

Equal to (—| = |—) : Logic continuity is ensured if the contents of the two addresses specified by this comparison operation command are equal to each other or if the content of an address is equal to each other with a number to be compared.

Less than (—| < |—) : Logic continuity is ensured if the content of an address specified by this comparison operation command is less than another or if the content of an address is less than a number to be compared.

Greater than (—| > |—) : Logic continuity is ensured if the content of an address specified by this comparison operation command is greater than the other, or if the content of an address is greater than a number to be compared.

Program flow Control Commands

Master control relay (—(MCR)—) : This command is used to activate a group of ladder steps. If there is logic continuity in the step with the MCR instruction, the digits are scanned up to the digit carrying the END MCR instruction.

Label (—| LBL |—) : This command is made with the label command to define the digit in which the address is specified.

Jump to Label (—(JMP)—) : This command is used in situations where a change in the program flow needs to be created if some conditions are met.

Jump to Subroutine (—(JSB)—) : If there is logic continuity at the step where the instruction is located, it is jumped to the corresponding label step specified in the JSB instruction and the program flow continues from where it was until it is met with a RET instruction in order to execute the operations below this step.

Return Coil (—(RET)—) : When encountered, it is returned to the next step after the JSB command that caused the subroutine to be entered. A command to exit a subprogram.

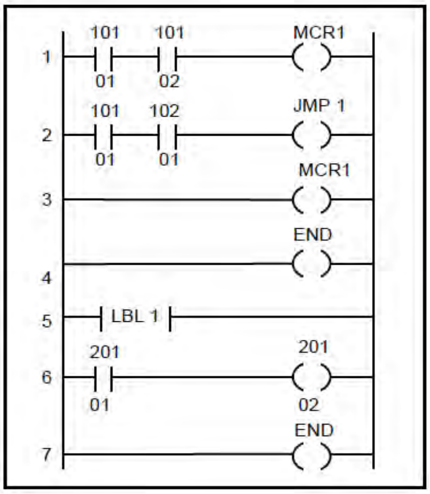


Figure 8: An Example Ladder Diagram Using Data Comparison Operation Instructions and Program Flow Control Instructions

An example Ladder diagram using Data comparison operation commands and Program flow control commands is given in Figure 8. In the first step of the ladder diagram; If the 101/01 and 101/02 contacts are closed, MCR1 will be active and the steps up to the digit where MCR1 is END are processed.

If step 1 is not active, step 4 is skipped. If 101/01 and 102/01 are active, the JMP 1 command is executed and jumps to the 5th step where the 1st tag command is located. If 201/01 contact is active, 201/02 output is active (Özcan and Özkan, 2004).

References

Birant D., Kut A., 2018. Monitoring of PLC Based Systems over the Internet. https://www.researchgate.net/publication/322754124

Crispin A. J.,1996. Programmable Logic Controllers and their Engineering Applications, Mc Graw Hill.

Çalışkan, Ö., 2007. Multıple Servoo Motors, Sensors And Texturebendıng Machıne Including Electropneumatic Element Automation, Institute of Natural Sciences, ITU.

Kurtalan, S., 2001. Industrial Automation with PLC, Birsen Publisher, İstanbul.

Özcan M., Özkan A.O., 2004. PLC Applications in Automation Systems. 975-6494-36-4, Atlas Publisher, Ankara.

Petruzella F. D., Programmable Logic Controllers, Macmillan/McGraw- Hill, Ohio, 1992.

Webb J. W., Reis R. A., 2002. Programmable Logic Controllers: Principles and Applications, Prentice Hall.

URL 1: https://www.electricalchile.cl/plcall1en.php

URL 2: https://www.mediaclick.com.tr/tr/blog/plc

URL3: https://www.linkedin.com/pulse/expanding-our-plc-programming-skills-pedro-barretto/

URL 4: https://maker.robotistan.com/plc-nedir-plc-programlama-teknikleri-ve-ozellikleri/

About the Authors

Ozgur DUNDAR, PhD, works at Necmettin Erbakan University, Department of Space and Satellite Engineering. He graduated from the Electrical and Electronics Engineering Department of Selcuk University. He worked as an Automation Engineer for a while. His master’s and doctorate degrees are from Selçuk University, Institute of Science and Technology, Department of Electrical and Electronics Engineering. Special fields of study are Automation, Robotic, Communication, Electromagnetic and Micro Strip Patch Antenna designs.

Email: ozdundar@erbakan.edu.tr , Orcid: 0000-0002-4142-4446

Sabri KOCER, PhD, He graduated from the Electrical Engineering Department of Selcuk University. He completed his graduate and his doctorate in Gazi University. Currently, Necmettin Erbakan University, Faculty of Engineering, Computer Engineering is working. Electronics, Computer, Telecommunication, Signal Processing and Biomedical studies in the area.

Email: skocer@erbakan.edu.tr , Orcid: 0000-0002-4849-747X

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 5%

To Cite This Chapter:

Dundar, O. & Kocer, S. (2021). Raspberry Pi and Models. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 50–66). ISRES Publishing.

STM32

Mohammed H. IBRAHIM
Necmettin Erbakan University

Introduction

ARM has named by the abbreviation of (Acorn RISC Machine). ARM processors are 32-bit and 64-bit RISC (Reduced instruction set computer) based processors. ARM is preferred in various embedded applications due to its ability to reach high speeds, has 64-bit architecture, consume very little energy, and rich peripheral hardware. Moreover, ARM architecture processors are used in many electronic devices such as mobile phones, computers, and tablets. (See Figure 1 Areas use of ARM processors).

Figure 1. Areas use of ARM Processors

Many companies are used ARM processors to design ARM-based microcomputers using their own software, libraries, and names. Most of these designed microcomputers are used very widely in education and embedded systems projects. In general, the programming of these ARM-based microcomputers is based on the C language. However, since each company shapes the library features according to the microcomputer it produces, each brand has its own libraries. According to this information, although programming ARM is basically based on the C language, programming will also vary with each company, due to the changing libraries and characteristics of the designed microcomputer. For example; when you learn the ARM processor-based STM32 Discovery programming produced by STM, you cannot program ARM processors produced by TI using the same libraries and features.

As given in Figure 2, the entire ARM architecture is classified into 3 classes according to application areas: classic ARM processors, embedded cortex processors, and application cortex processors. Classic ARM processors contain ARM, embedded cortex processors contain ARM cortex-M and ARM cortex-R, and application cortex processors contain ARM cortex-A (Ngabonziza et al., 2016).

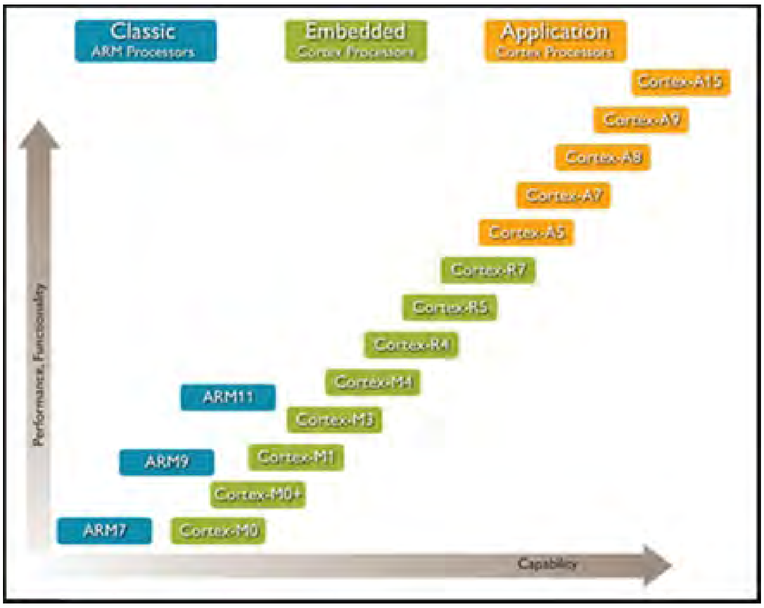


Figure 2. Classification of the ARM Processors

Cortex-A series: Cortex-A is a series developed for applications that require high speed due to its low power consumption and fast operation (Wan et al., 2012). Cortex-A5, Cortex-A7, Cortex-A8, Cortex-A9, and Cortex-A15 are types of ARM Cortex-A commonly used. In addition, the ARM Cortex-A series are commonly used in microcomputers such as Tablet, raspberry pi, Nvidia Jetson.

Cortex-R series: Cortex-R series are frequently used in real-time projects and technical devices. Cortex-R4, Cortex-R5, and Cortex-R7 are types of ARM Cortex-R commonly used. Cortex-R series are used in computer hardware devices (hard disk control, network connection equipment), automotive industry (airbag systems, brake and engine management), household electronic devices (food robots, washing machines, refrigerators, rugs), and enterprise printers. For example, Cortex-R4 is frequently used in the automotive industry (Iturbe et al., 2017).

Cortex-M series: Cortex-M series are a series developed to be used in the electronics industry projects where standard 8/16 bit microcontrollers are used, and projects that require a low cost / low power consumption. Moreover, due to its low cost, it is frequently preferred in educational control cards. Therefore, this article is described the STM32 control board with Cortex-M4 ARM processor, which is widely used today. Cortex-M0, Cortex-M1, Cortex-M3, and Cortex-M4 are types of ARM Cortex-M commonly used (Martin, 2016).

Classic ARM: Generally, classic ARM processors are used in many simpler and smaller-scale embedded systems. ARM7, ARM9, and ARM11 are families of ARM Cortex-A and ARM7 is still the highest shipping 32-bit processor (Fleisher & Bensoussan, 2015). Even though classic ARM processors can be used in many simpler and smaller-scale 32-bit devices, newer embedded systems are built using advanced ARM processors like Cortex-M processors and Cortex-R Processors.

ARM Processor Architecture

The ARM consists of a processor which is responsible for mathematical operations such as addition, subtraction, etc., as well as logical operations such as comparison. The ARM also contains a random access memory, which is called RAM, in which the program is stored temporarily while it is running and this memory is erased all its content from disconnect the power supply (Volatile), and it also contains an electronically erasable programmable read-only memory, which is called a EEPROM, as well as the general-purpose input/output and timers circuits. These aforementioned components interconnected with address, control, and data buses as illustrated in Figure 3.

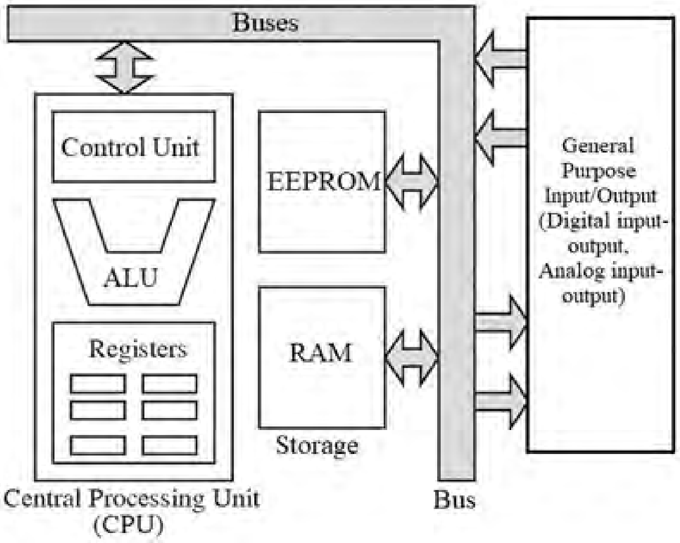


Figure 3. Components of ARM

The process of placing these components together and connecting them with each other will not lead to obtaining the required functions of the motherboard, which is represented in driving the computer as a whole, showing what is happening on the output and receiving various commands from the user through input terminals. There should be a certain methodology that organizes the flow of computer operations and all these tasks are fundamentally related to the concept of "architecture."

Architecture is the mechanism by which different functional units are organized within the computer system. Thus, the architecture determines how data and information transfer between computer units such as the central processing unit, memory, and input and output ports. To evaluate any architecture there are three main criteria that must be considered:

- Performance
- Cost
- The maximum size of the program and data

Besides, there are many other criteria by which we can evaluate the architecture such as power consumption, size and weight (two criteria concerned with the physical structure of the processor), and ease of programming. But, the three criteria above are important in all applications.

All processors today depend on the Instruction Set Architecture (ISA), where the wizard implements one instruction from the program's instructions through several cycles, and the program's instructions are executed one by one. The instruction is the simplest function that the wizard can perform, and it consists of Opcode (represents the required function) and Operand, which is the recorder on which the required function is applied, for example:

ADD r1, r2, r3;

You add the value of r3 with r2 and put the result in r1, which is equivalent to r1 = r2 + r3. You can learn more information about the instruction set of ARM from (www.st.com:, 2019).

Von Neumann and Harvard Architectures

Von Neumann: The design philosophy in this type of architecture is based on storing Instructions and data in the same memory (Mariantoni et al., 2011). In this case, the processor would need to implement the instruction consisting of one opcode and one parameter into three cycles - a cycle for fetching and reading the opcode (by the address bus), a cycle for reading the parameter, and a final cycle for implementing the instruction content - which is a waste of time. Therefore, there is a mainline bus to transfer data and control commands to all computer components, where the processor fetches the program instructions stored in the random memory to the CPU to execute it as well as the data is fetched from the input means or from random memory or vice versa data is taken out for output devices or stored in memory (Ben-Sasson et al., 2014). ARM Cortex-M23 is an example of Von Neumann architecture (Yiu, 2020). (See Figure 4).

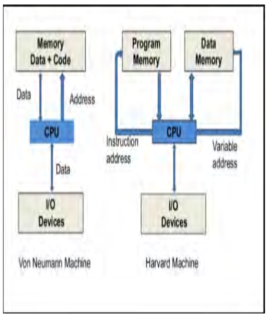


Figure 4. Von Neumann Architecture

Harvard: In this architecture, there are two separate memories, a memory for storing instructions called "Program Memory (PM") and another memory called "Data Memory (DM") for storing data. This is what needs an address bus and a special Data Bus for each memory CPU. This allows the processor to fetch information from program memory and access variables in the data memory at the same time. And since the instruction memory was separated from the data, the CPU can now write the opcode and parameter in one statement, as the CPU can read the entire instruction in only one session most of the time. This is the goal behind Harvard architecture, and this is what makes devices that rely on this architecture give much higher productivity than those based on von Neumann architecture. ARM Cortex-M4 is an example of Harvard architecture. (See Figure 5).

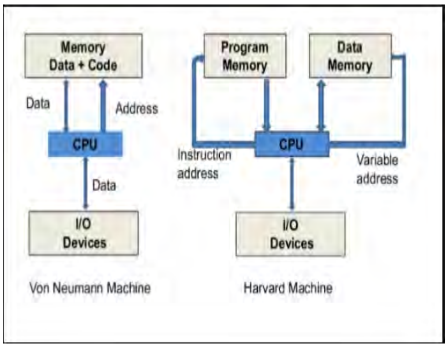


Figure 5. Harvard Architecture

Briefly, in microprocessors, instructions are stored in read-only memory (or fixed memory), that is, ROM, and variables are stored in random (or changeable) memories. RAM, so the memory in the von Neumann architecture is integrated between ROM, which saves the instructions and their operating codes, and RAM, to save the variables. The second reason ARM processors are fast is that they have RISC architecture. RISC architecture has a faster and lower cost than CISC architecture. On the other hand, CISC architectures have relatively more instruction sets than RISC architectures.

Cortex-M Family and STM32

Cortex-M family is used by many companies, especially companies that develop projects with embedded systems prefer this processor. As the usage areas of the Cortex-M series increased, companies such as ST (STMicroelectronics) and TI (Texas Instruments) started to manufacture ARM Cortex-M-based control boards with different properties. The application areas of ARM Cortex-M is given in Figure 6.

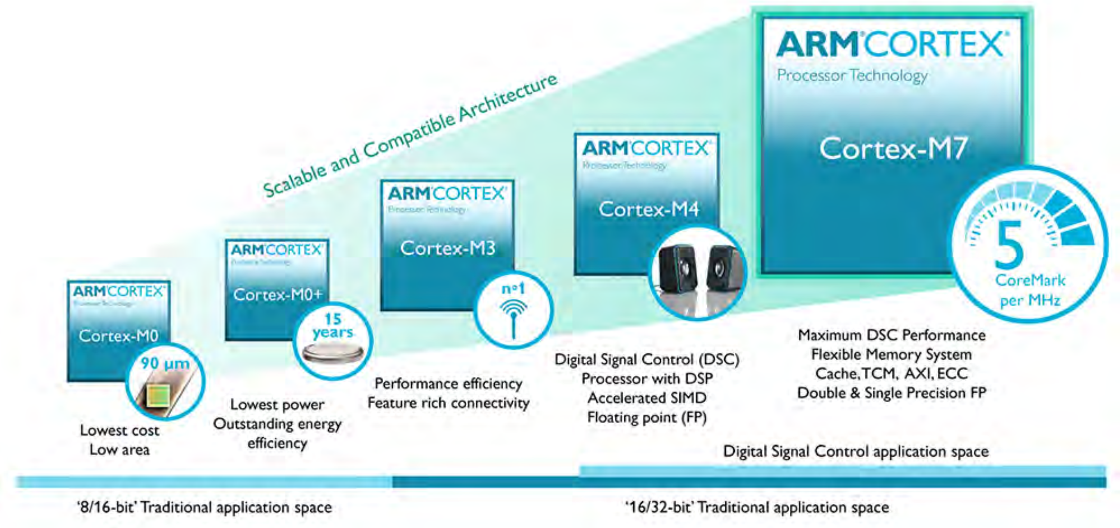


Figure 6. Application Areas of ARM Cortex-M

STMicroelectronics was founded as a result of the merger of two companies, one French and the other Italian in 1987 and since then the company has managed to occupy an important place among the major electronics companies. For instance, STMicroelectronics came in fifth place in 2002 after Intel, Samsung, Texas Instruments, and Toshiba, and today STMicroelectronics is the largest European manufacturer of chips. In terms of electronic profits, the STMicroelectronics Company started in 2007 with the production of its first product STM32F1 from the family of STM32 controllers based on ARM processors. The STM32 family of controllers features a wonderful 32-bit Cortex-M processor allowing the user to use software tools from ARM and full support from many developments and programming environments. Moreover, the STM32 family also facilitates the process of moving from one controller to another within the STM32 family, so you do not have to worry if you program and develop on a microcontroller and want to move to another controller, you can do this very easily. The STM32 family of controllers is also faster and cheaper than some 8-bit controllers.

As seen in Figure 7, the STM32 family of controllers contains 11 groups divided into 3 classes. First in terms of high-performance STM32H7, STM32F7, STM32F4, and STM32F2 have high computing power and are supported by ART Accelerator technology that enables direct execution of instructions from flash memory without waiting and speeds up to 400MHz, Therefor, it is suitable for multimedia, graphics, and digital signal processing applications.

In terms of medium performance, it includes STM32F0, STM32F1, and STM32F3. This type of STM is characterized by its cheap price and small size, as it is convenient for many commercial applications and has speeds of up to 72MHz.

In terms of low energy consumption, it includes STM32L4, STM32L4 +, STM32L0, and STM32L1 and it is of course characterized by its low energy consumption, which makes it suitable for applications that depend on battery for operation. This type of STM is characterized by its support for Dynamic Voltage Scaling Technology - DVS, which allows improving the processor's energy consumption based on the frequency used.

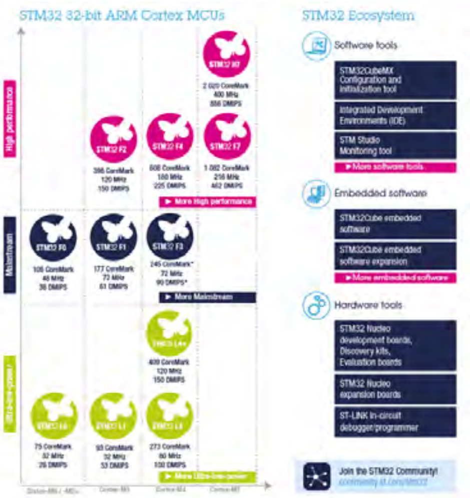


Figure 7. Class of the STM32 Cortex-M family

Technical properties of STM32

In this section, the STM32F4 discovery board, which is frequently used among STM32 boards, has examined in detail, the top layout of the STM32F4 board (www.st.com:, 2020) and the pinout of the STM32F4 board (Microcontrollerslab, 2021) are given in Figures 8 and 9, respectively.

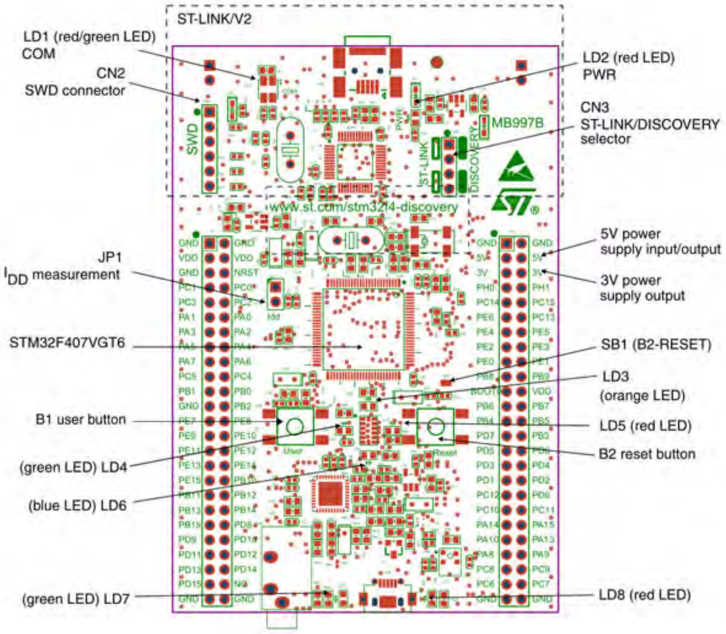


Figure 8. The top Layout of the STM32F4 Board

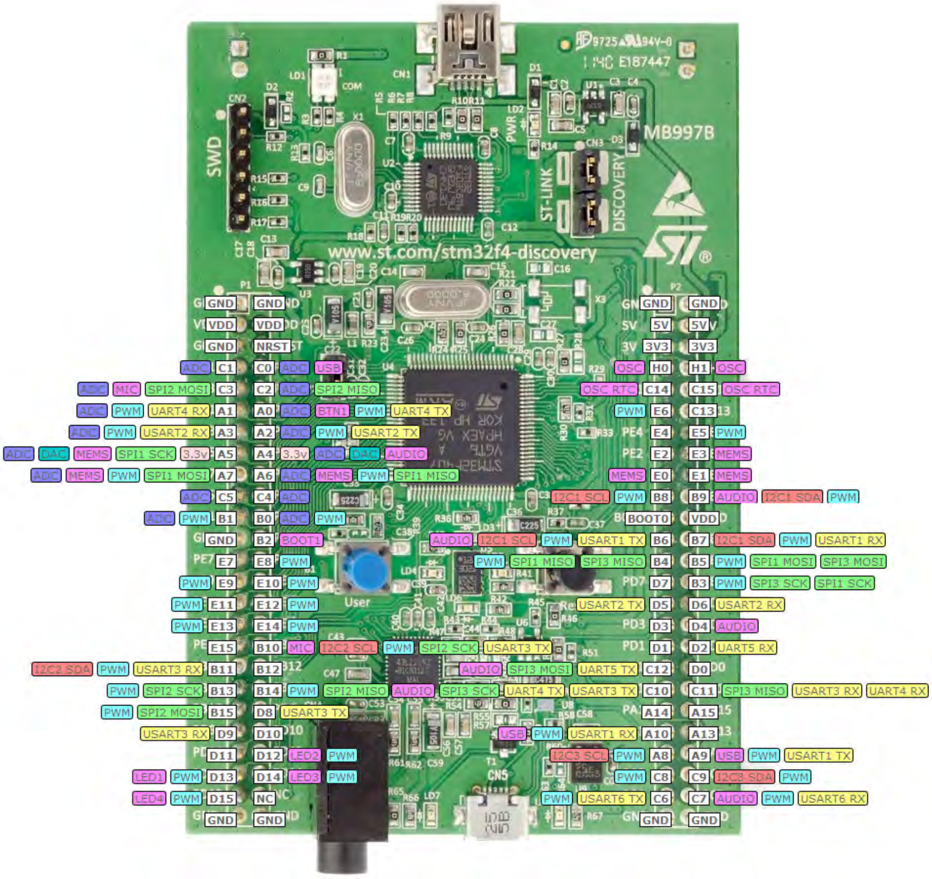


Figure 9. The Pinout of the STM32F4 Board

In addition, the technical properties of STM32F4 discovery which is commonly used has given bellow (www.st.com:, 2020).

Processor (Core): The STM32F4 discovery contains ARM Cortex-M4F core with a maximum clock rate of 84 / 100 / 168 / 180 MHz.

Oscillators: The oscillators in the STM32F4 discovery consists of internal (16 MHz, 32 kHz) and optional external (4 to 26 MHz, 32.768 to 1000 kHz).

Memory: The STM32F4 include flash memory and static RAM. The flash memory consists of up to 2 MB of dual bank and supporting read-while-write (RWW), 512 / 1024 / 2048 KB general-purpose, 512 bytes one-time programmable (OTP), and 30 KB system boot. Also contain dual bank option bytes for microcontroller configuration. The static RAM memory consists of up to 192 KB general-purpose and 64 KB core-coupled memory (CCM).

Peripherals:

Communication: The STM32F4 include Universal Serial Bus 2.0 On-The-Go (USB OTG), two Controller Area Network (CAN bus) 2.0B, one Serial Peripheral Interface (SPI) and two SPI or full-duplex Inter-IC Sound (I²S), three Inter-Integrated Circuit

(I²C), four universal synchronous and asynchronous receiver-transmitter (USART), two universal asynchronous receiver-transmitter (UART), Secure Digital Input Output (SDIO) for SD/MMC cards.

Timers: STM32F4 has a total of 17 timers with 16-bit and 32-bit including 10 general-purpose timers, two advanced control timers, two basic timers, one independent watchdog timer (IWDT), one window watchdog timer (WWDT), and one systematic timer.

General purpose timers: The general purpose timers include 16-bit or 32-bit timers up, down, up/down auto-reload counter. TIM3, TIM4, TIM9, TIM10, TIM11, TIM12, TIM13, and TIM14 are 16-bit otherwise TIM2 and TIM5 are 32-bit timers. These timers can be used for a diversity of purposes, including input capture or generating output waveforms (output compare and PWM). The block diagram of general purpose timers is given in Figure 10 (www.st.com:, 2021).

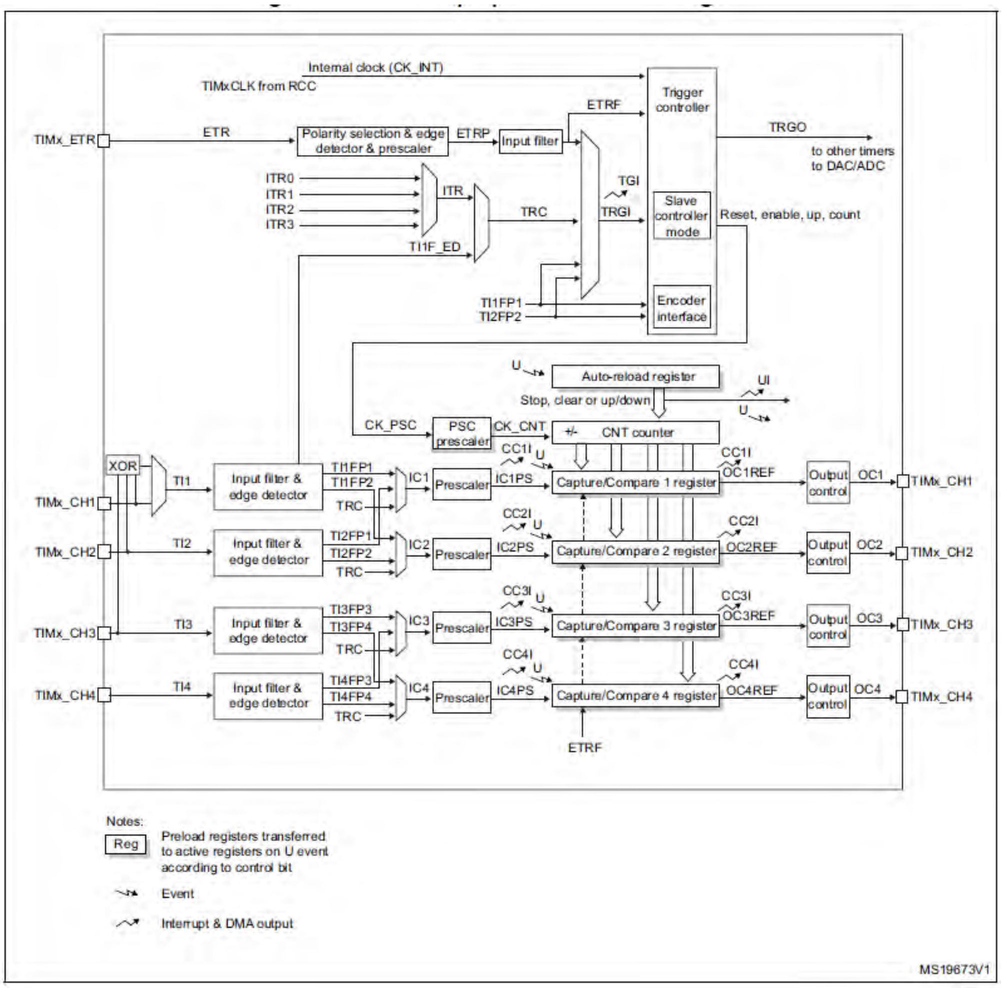


Figure 10. The Block Diagram of General Purpose Timers

The advanced-control timers include TIM1 and TIM8, theses timer are 16-bit auto-reload counter driven by a programmable prescaler and can be used for a diversity of purposes, including input capture or generating output waveforms (output compare and PWM).

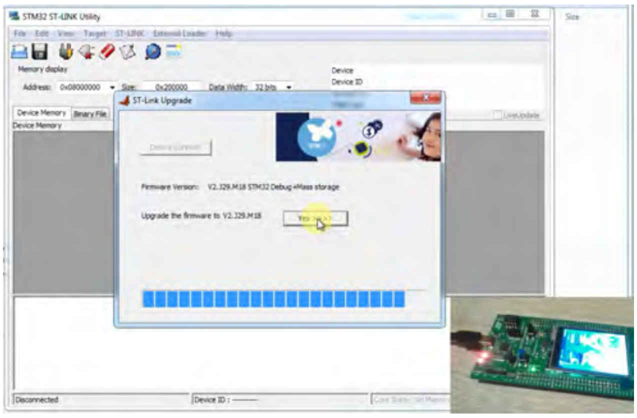
Basic timers include TIM6 and TIM7 timers which are 16-bit auto-reload counter driven by a programmable Prescaler. The TIM6 and TIM7 timers are able to drive digital-to-analog converter (DAC) through their trigger outputs because these timers internally connected to the DAC.

General-purpose input-output pins: The STM32F4 includes 51 to 140 general-purpose input-output pins, three ADCs with 12-bit, 10-bit, 8-bit, or 6-bit configurable resolution, and two Digital-to-analog converter (DAC) which can be configured in 8- or 12-bit mode and can be used in conjunction with the direct memory access (DMA) controller.

Important properties: The STM32F4 includes important properties such as direct memory access (DMA) that increases data transfer speed between memory and memory and between peripherals and memory, improved real-time clock (RTC), random number generator (RNG), cyclic redundancy check (CRC) engine, Ethernet MAC, LCD-TFT controller, and camera interface.

STM32 Programming

The STM32 board can be programmed in a low-level language like assembly and high-level language like c. Moreover, the STM32 board can be programmed with Cube which is used to graphically configure the processor of STM32. While ASM and C programming languages require sufficient knowledge of the processor, memory organization, and instruction set, programs such as Cube that make graphical settings do not require much knowledge about the processor, memory organization, and instruction set. There are many c based debuggers or editors such as Keil (Keil, 2021), MikroC (MikroC, 2021), CooCox (www.st.com, 2021a), and IAR developed (IAR, 2021) to program STM32; however, the TrueSTUDIO (www.st.com, 2021b) program a commercially enhanced C/C++ IDE built on Eclipse, GCC, and GDB is preferred because the TrueSTUDIO program is supported by ST. To get more information and for installing, you can visit the st.com website. As seen in Figure 11, after installing the TrueSTUDIO you should update the firmware of the device from the STM32 ST-LINK utility.



Address: 0x08000000 Size: 0x200000 Data Width: 32 bits

Firmware Version: V2.J29.M18 STM32 Debug+Mass storage

Upgrade the firmware to V2.J29.M18

Device ID : ---------

Figure 11. Update the Firmware of the STM32

Then we can easily create a project and assign functions to the STM32. The software development flow and the software compilation flow in the STM32 are given in figures 12 and 13, respectively.

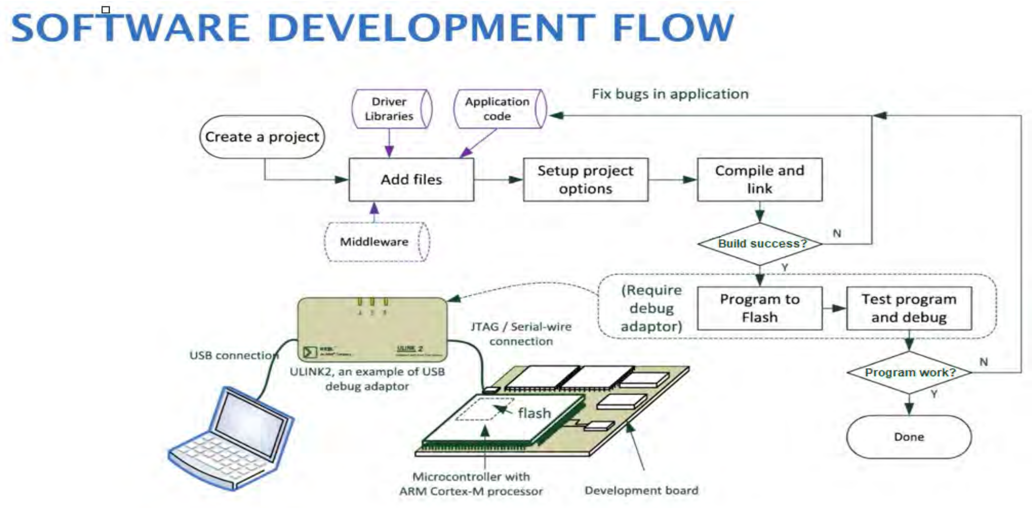


Figure 12. The Software Development Flow

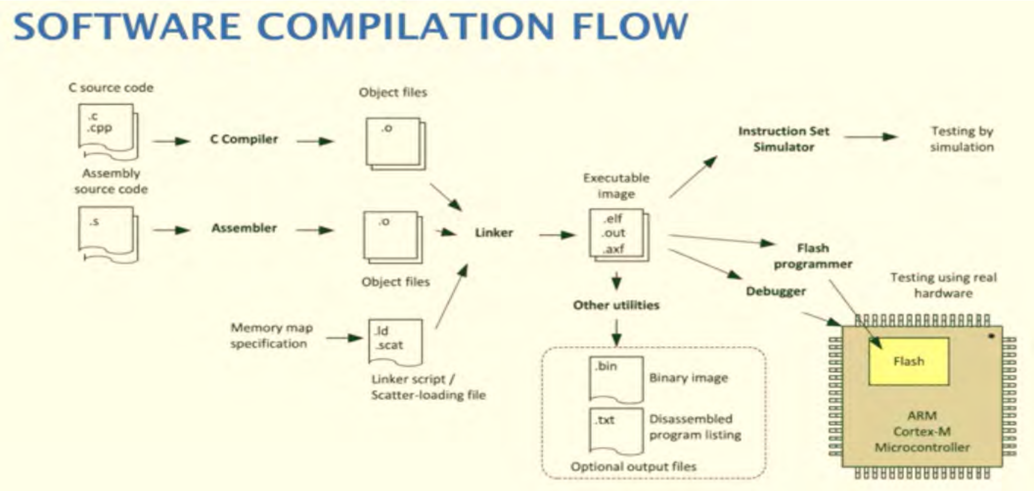


Figure 13. The Software Compilation Flow

As shown in Figures 12 and 13, the software written by the programmer can be embedded in the ARM microcontroller after debugging and linking by the compiler. In the debug process, a file suitable for the ARM processor is prepared by the compiler, this file contains instruction sets in the form of machine code. The instructions are processed sequentially by the ARM processor, and each instruction goes through at least three stages, fetch, decode and execute. Below is a short code about the general-purpose input/output ports of the ARM Microcontroller.

```
// Toggle output pins (D_Pin_0, D_Pin_1, D_Pin_2) with input pin (A_Pin_0)

#include “stm32f4xx.h”

#include “stm32f4_discovery.h”
```

```
GPIO_InitTypeDef GPIO_portstype;

int control = 0;

int main(void)

   {

           RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

           // For port D (output)

           GPIO_portstype.GPIO_Pin = GPIO_Pin_0|GPIO_Pin_1|GPIO_Pin_2;

           GPIO_portstype.GPIO_Mode = GPIO_Mode_OUT;

           GPIO_portstype.GPIO_OType = GPIO_OType_PP;

           GPIO_portstype.GPIO_PuPd = GPIO_PuPd_NOPULL;

           GPIO_portstype.GPIO_Speed = GPIO_Speed_50MHz;

           GPIO_Init(GPIOD, &GPIO_portstype);


           RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

           // For port A (input)

           GPIO_portstype.GPIO_Pin = GPIO_Pin_0;

           GPIO_portstype.GPIO_Mode = GPIO_Mode_IN;

           GPIO_portstype.GPIO_OType = GPIO_OType_PP;

           GPIO_portstype.GPIO_PuPd = GPIO_PuPd_DOWN;

           GPIO_portstype.GPIO_Speed = GPIO_Speed_50MHz;

           GPIO_Init(GPIOA, &GPIO_portstype);

 while (1)

    {

           if(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == Bit_SET)

           {
```

```
                    while(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0));
                    if(kontrol == 0)
                    {
                            kontrol = 1;
                            GPIO_WriteBits(GPIOD, GPIO_Pin_0, Bit_SET);
                            GPIO_WriteBits(GPIOD, GPIO_Pin_1, Bit_SET);
                            GPIO_WriteBits(GPIOD, GPIO_Pin_2, Bit_SET);
                    }
                    else
                    {
                            kontrol = 0;
                            GPIO_WriteBits(GPIOD, GPIO_Pin_0, Bit_RESET);
                            GPIO_WriteBits(GPIOD, GPIO_Pin_1, Bit_RESET);
                            GPIO_WriteBits(GPIOD, GPIO_Pin_2, Bit_RESET);
                    }
            }
    } // while
 } //Main
void EVAL_AUDIO_TransferComplete_CallBack(uint32_t pBuffer, uint32_t Size){
 return;
}
uint16_t EVAL_AUDIO_GetSampleCallBack(void){
 return -1;
}
```

References

Ben-Sasson, E., Chiesa, A., Tromer, E., & Virza, M. (2014). Succinct non-interactive zero knowledge for a von Neumann architecture. 23rd USENIX Security Symposium (USENIX Security 14),

Fleisher, C. S., & Bensoussan, B. E. (2015). Business and competitive analysis: effective application of new and classic methods. FT press.

IAR. (2021). https://www.iar.com/ewarm.

Iturbe, X., Venu, B., Penton, J., & Ozer, E. (2017). A” high resilience” mode to minimize soft error vulnerabilities in ARM cortex-R CPU pipelines: work-in-progress. Proceedings of the 2017 International Conference on Compilers, Architectures and Synthesis for Embedded Systems Companion,

Keil. (2021). https://www.keil.com/.

Mariantoni, M., Wang, H., Yamamoto, T., Neeley, M., Bialczak, R. C., Chen, Y., Sank, D. (2011). Implementing the quantum von Neumann architecture with superconducting circuits. Science, 334(6052), 61-65.

Martin, T. (2016). The designer’s guide to the Cortex-M processor family. Newnes.

Microcontrollerslab. (2021). https://microcontrollerslab.com/stm32f4-discovery-board-pinout-features-examples/.

MikroC. (2021). https://www.mikroe.com/mikroc-arm.

Ngabonziza, B., Martin, D., Bailey, A., Cho, H., & Martin, S. (2016). Trustzone explained: Architectural features and use cases. IEEE 2nd International Conference on Collaboration and Internet Computing.,

Wan, J., Wang, R., Lv, H., Zhang, L., Wang, W., Gu, C., . . . Gao, W. (2012). AVS video decoding acceleration on ARM Cortex-A with NEON. IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012),

www.st.com. (2021a). https://www.st.com/en/development-tools/coide.html.

www.st.com. (2021b). https://www.st.com/en/development-tools/truestudio.html.

www.st.com:. (2019). STM32 Cortex-M4 MCUs and MPUs programming manual.

www.st.com:. (2020). Discovery kit with STM32F407VG MCU (User manual UM1472).

www.st.com:. (2021). STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm-based 32-bit MCUs (Reference manual RM0090).

Yiu, J. (2020). Definitive Guide to Arm Cortex-M23 and Cortex-M33 Processors. Newnes.

About the Authors

Mohammed H. IBRAHIM received the B.S. degree in Software Engineering from Northern Technique University, Kirkuk, Iraq. He received M.S. and Ph.D. degrees in Computer Engineering from Selcuk University, Konya, Turkey. His research aims to develop and apply methods for data mining and machine learning and aims to use these methods in embedded systems and control. He has published more than 18 research papers in reputed international journals indexed in SCI, SCOPUS, and Web of Science.

E-mail: mibrahim@erbakan.edu.tr Orcid: 0000-0002-6093-6105

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 12%

To Cite This Chapter:

Ibrahim, M. H. (2021). STM32. In S. Kocer, O. Dundar & R. Butuner (Eds.), *Programmable Smart Microcontroller Cards* (pp. 67–81). ISRES Publishing.

Nvidia Jetson Nano Development Kit

Fatma Nur UZUN
Konya Technical University

Mehmet KAYRICI
Necmettin Erbakan University

Beyzanur AKKUZU
Necmettin Erbakan University

Introduction to NVIDIA Jetson Nano

Developed on object detection, one of the main problems in computer vision, Jetson Nano was designed by California-based NVIDIA Technology Company ("About NVIDIA," t.y.; Barba-Guaman, Eugenio Naranjo, Ortiz, 2020). NVIDIA Jetson family's announced the first Jetson Nano Developer Kit at the GPU Technology Conference (GTC) in March 2019 (Salih & Gh, 2020; Franklin, 2019).

NVIDIA's Jetson is a new integrated accelerator hardware created by the widespread use of automated learning algorithms. This kind of performance contributes to the development of applications in many areas, such as multiple sensor robots, intelligent IoT devices, and advanced Artificial Intelligence (AI) systems (Franklin, 2019). Originally designed for Internet of Things (IoT) manufacturers and creators, the Jetson Nano card is an AI computer that generates low power and is low-cost, small, and powerful, allowing a large number of AI algorithms to run in parallel ("Jetson Nano Developer Kit," t.y.; Kurniawan, 2021). However, achieving Jetson's full potential and achieving real-time performance involves an optimization phase for both hardware and different algorithms.

The fact that jetson nano is a small computer can be considered a daily computer. We can add sensor and actuator modules to NVIDIA Jetson Nano devices. The card provides a GPIO interface to connect to external device modules (Kurniawan, 2021).

Jetson Nano has three different models developed. These are defined as Jetson Nano A02 Developer Kit, Jetson Nano 4GB Developer Kit (B01) and Jetson Nano 2GB Developer Kit. The main difference between these two models, the A02, which was first released in March 2019, and the B01 (4GB), the second of which was released in January 2020, was that the A02 version could not boot with Intel 8260 WiFi installed. This problem has been updated with version B02 ("NVIDIA Jetson Linux," 2019). An extra camera slot has been added to the B02 (Hao, 2020). Other features and components are the same.

When all three models are examined, the ARM A57 CPU has a 128-core Maxwell GPU, GPIOs, I2C, SPI, UART, USB, HDMI, power and fan connectivity, as well as a microSD card slot for main storage. Other input and output interfaces such as serial communication, SPI and I2C can be accessed through the program.

The Jetson Nano Developer Kit is powered by the JetPack SDK and compatible with the artificial intelligence frameworks that are becoming popular today: we can run Pandas, Numpy, Tensorflow and Keras on an NVIDIA Jetson Nano card. By installing the OpenCV library, we can easily use NVIDIA Jetson Nano to create computer image programs. The OpenCV library provides a variety of image and video processing libraries. In addition, OpenCV consists of machine learning libraries such as facial recognition.

Programming platforms such as C/C ++ and Python are supported in NVIDIA Jetson Nano image. Since this image uses operating system-based operating systems, we can install various compilers and applications, including web applications and databases.

Jetson Nano Cards and Features

The first goal for NVIDIA Jetson Nano is to create IoT solutions. Jetson Nano cards developed over time have also become available for AI applications. The developed cards cover the shortcomings of the previous card, but also bring new features.

In general, some additional hardware will be needed to run the NVIDIA Jetson Nano Developer Kit. Additional devices are provided below:

- MicroSD card with a minimum storage size of 16 GB
- MicroSD card reader to retrieve data from computer
- USB wired mouse
- USB wired keyboard
- Power adapter 5V 2A
- Micro USB cable for power adapter
- HDMI connector monitor

You can easily find these products in electronics stores.

Jetson Nano A02 Developer Kit

Launched in March 2019, the A02 has become frequently used by developers in AI applications with its low price and strong performance.

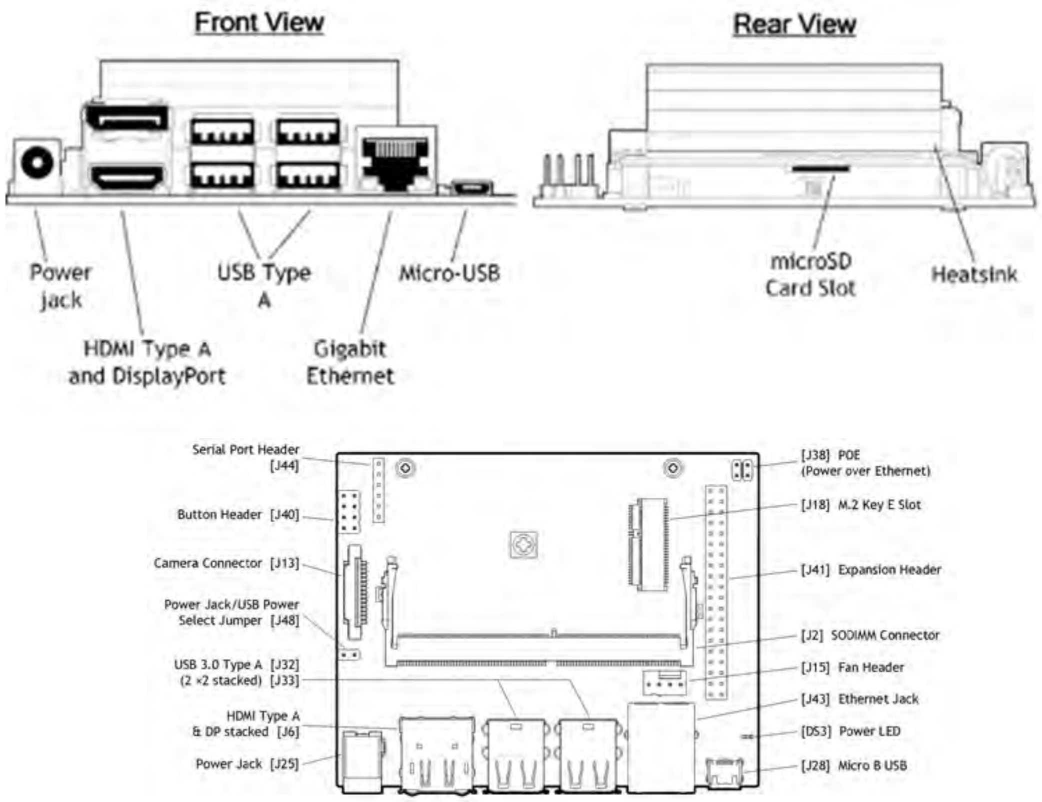


Figure1. NVIDIA Jetson Nano Developer Kit Module and Carrier Card

Specifications;

- Quad-core ARM A57 CPU
- 128-core Maxwell GPU
- 4GB 64-bit Memory | 25.6 GB/s
- 3 USB ports
- Power over Ethernet (PoE)
- 5V DC Power Input
- Fan Connector
- Camera Connector
- HDMI port
- DisplayPort

Jetson Nano B01 (4GB) Developer Kit

NVIDIA has just updated its Jetson Nano; In January 2020, it launched a new model, the

Jetson Nano B01 development card (Kurniawan, 2021; Jackson, 2020). With the updates made, the carrier card has been improved while supporting new interfaces and hardware.

Lending to the hardware ISP (Image Signal Processor) available in all Jetson Nano B01 variants, the NVIDIA libargus interface provides an application-engraved video stream with mosaic removal, color correction, and white balance algorithms ("Development Kits Nvidia," t.y.). Display Source provides a default configuration file for the ISP. Using the GStreamer interface for libargus, the app can interface with NVIDIA's unchanging Jetson ecosystem software buildup, which includes efficient integration of powerful image processing frameworks such as DeepStream, Visionworks, and others ("NVIDIA Jetson Nano," t.y.).

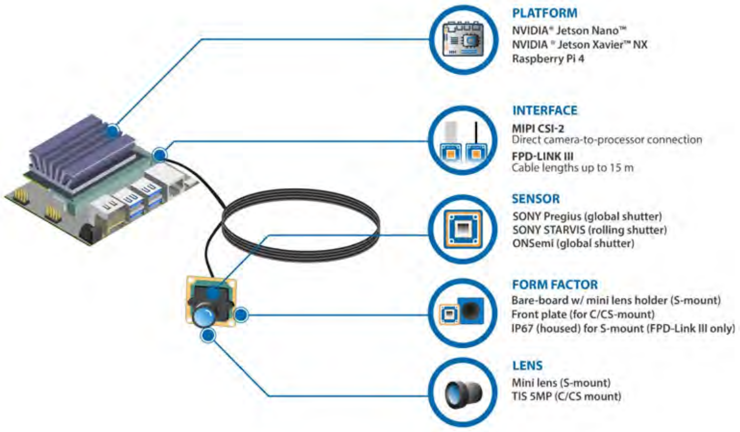


Figure 2. Jetson Nano 4GB

With updates to Jetson Nano, Jetpack 4.3 was launched into the economy market with TensorRT 6.0.1 and cuDNN 7.6.3 libraries that helped improve artificial intelligence inference performance by 25%.Using the GPU + CPU hardware encoder/decoder, VPI (Vision Programming Interface) accelerates 4K video or multiple 1080P video feeds (up to 8x at a time) and performs multiple tasks of ML (Machine Learning) algorithms, such as image detection recognition and monitoring (Jackson, 2020).

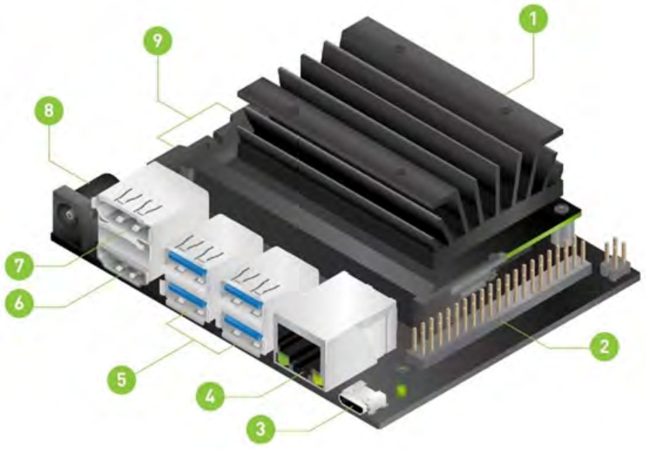


Figure 3. NVIDIA Jetson Nano B01 (4GB)

Specifications;

1) MicroSD card slot for main storage
2) 40-pin expansion head
3) Micro-USB port for 5V Power input or data transfer
4) Gigabit Ethernet port
5) USB 3.0 ports (x4)
6) HDMI output port
7) DisplayPort connector
8) DC Barrel jack for 5V power input
9) MIPI CSI camera connectors (x2) that create the appropriate environment for binocular applications such as stereo recording, depth detection, 3D object tracking and image consolidation

Jetson Nano 2GB Developer Kit

In October 2020, NVIDIA launched the more economically viable Jetson Nano 2GB developer kit (Hao, 2020). Jetson Nano 2GB provides an efficient environment for hands-on teaching and learning in the AI and robotics community.

Jetson Nano 2GB developer actively enables videos, open source projects, AI frameworks and models for applications such as image classification, object detection, partitioning and speech. These start with an introductory “Hello AI World”, continue with robotics projects such as the open source NVIDIA JetBot AI robot platform, and advance to the next stage of robotics development with NVIDIA Isaac™. These resources are supported by NVIDIA JetPack™, which includes the same CUDA-X™ software and tools used by experts from all over the world to each Jetson Nano 2GB developer. JetPack includes a known Linux environment and facilitates progress with its service to cloud-based technologies such as containerization and orchestration (“NVIDIA Jetson Nano,” t.y.). The Jetson Nano 4GB Development kit is shown in Figure 4:

Figure 4. NVIDIA Jetson Nano 2GB Developer Kit

Compared to Jetson Nano B01 (4GB);

- Camera slot (J49) removed.
- Dc barrel jack changed to USB Type-C port.
- USB Type-A ports have been changed.
- The display port has been removed.
- PoE - Power port over Ethernet (J38) removed.
- M.2 Key E slot (J18) removed.

Jetson Nano Module

The Jetson Nano module comes with the form factor and availability of users, the guarantees they need to put special carrier cards on the line. The B01 developer kit carrier motherboard is now compatible with the new Jetson Nano module for those who want to design a product (“Jetson Nano Module,” t.y.).

This carrier motherboard has been updated to support the production of the Jetson Xavier NX Module, which will go on sale in March 2020 (“Jetson Nano Module,” t.y.).

The Jetson Nano is a small, powerful computer for embedded applications and AI IoT that demonstrates the power of modern AI in this $99 (1KU+) module (“Jetson Nano Developer,” 2019).

Figure 5. Jetson Nano Module

Chapter 3. Installation and Operation

The NVIDIA Jetson Nano can be thought of as a minicomputer. Some hardware and software items such as mouse, keyboard and monitor are needed.

3.1. Hardware Preparation

The Jetson Nano Developer Kit needs a microSD card as the boot device and for main storage. For projects it is necessary to have a card large enough and fast; The recommended minimum is 32 GB UHS-1 card. A microSD card reader is required to work with the computer to read and write files. Figure 6.

Figure 6.. MicroSD card and NVIDIA Jetson Nano

Generally, some additional hardware will be needed to operate the computer. Power adapter 5V, 2A, Micro USB cable for power adapter, monitor with HDMI connector are some of them.

After the hardware needs are met, the necessary software can be installed to make the NVIDIA Jetson Nano work.

Installing Software

NVIDIA Jetson Nano uses its own operating system to run its applications. This operating system is based on Ubuntu Linux.

It needs the Etcher tool to upload the NVIDIA image file to a microSD card. This tool is available for Windows, Linux and macOS.

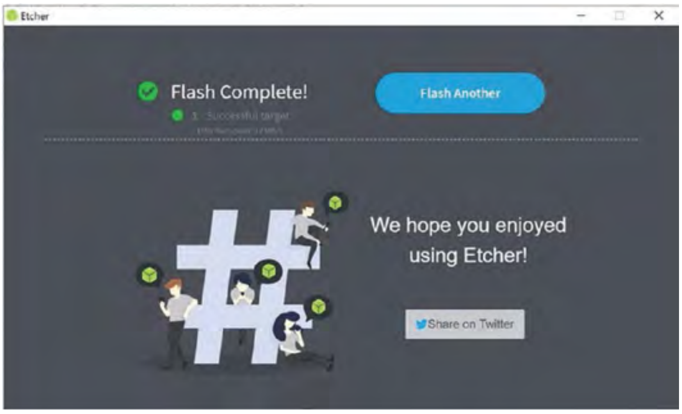
Installation Steps

1. Insert your microSD card with the reader into your computer.

2. Start flashing the image file by selecting the image file. Select the NVIDIA image file (ZIP file). Then choose your microSD card.

3. You can now start transferring the NVIDIA Jetson Nano image to your microSD card. Click Flash. You will be asked to give permission to the administrator to flash the image. This process takes a few minutes to complete.

4. You will get confirmation after it finishes flashing. Figure 3-2 shows the completion of the flashing image with the Etcher application. If you have finished flashing the image file, you can remove your microSD card from your computer. Then you can put it on the NVIDIA Jetson Nano device.



Flash Complete!

We hope you enjoyed using Etcher!

Figure 7. Flashing Jetson Nano Developer Kit SD Card Image

Run NVIDIA Jetson Nano

After transferring the NVIDIA Jetson Nano image to the microSD card, you can insert the card into the NVIDIA Jetson Nano card. Figure 8 shows a microSD card installed in an NVIDIA Jetson Nano.

Figure 8. Installing microSD Card in NVIDIA Jetson Nano

With it, the keyboard, mouse and monitor are attached to the NVIDIA Jetson Nano device. It uses DC jack power or micro USB power with a 5V, 2A power adapter to pump power to the NVIDIA Jetson Nano. Figure 9

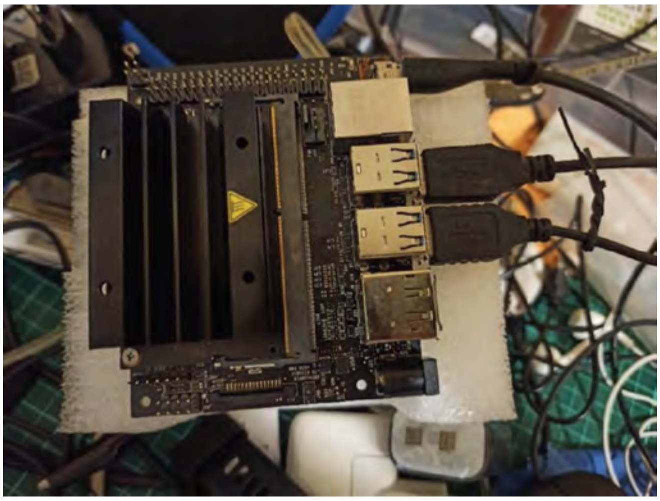


Figure 9. Plug in Keyboard, Mouse, Monitor and Power Adapter

Configure NVIDIA Jetson Nano Software

You can configure NVIDIA Jetson Nano for the first time after plugging in the power adapter. You must accept the required agreement to continue using the NVIDIA Jetson Nano device

After accepting the agreement, click the continue button. However, you will be asked to select a language for all text on the screen. You can also set the keyboard type and time zone for the local area. Finally, you must create your required account for NVIDIA

Jetson Nano. Enter the full name, username and password. Also set the authentication model. It is recommended that you use the Require my password to login option.

If your NVIDIA Jetson Nano is connected to the internet via a LAN cable or Wi-Fi module, you can connect to the internet. Then you can update the NVIDIA Jetson Nano software.

After performing all the operations, you will see the NVIDIA Jetson Nano desktop. This desktop is based on Ubuntu Linux. Figure 10. With the end of the process, you can create and edit files, surf the Internet, chat, etc. You can perform activities you do on any computer, such as.

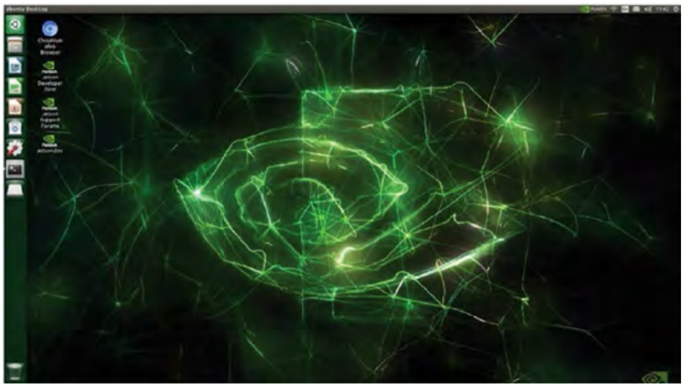


Figure 10. NVIDIA Jetson Nano Desktop

Since the NVIDIA Jetson Nano image is built from Ubuntu, you can use Terminal for administrative tasks such as creating/editing files and folders or compiling and executing programs.

You can find the NVIDIA Jetson Terminal by clicking Search at the top left. Type "Terminal" to see the terminal application. After clicking on the Terminal icon, you will get the Terminal application as seen in Figure 11.

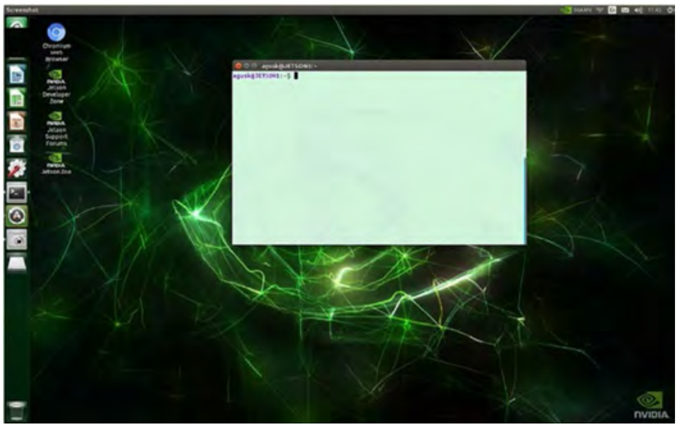


Figure 11. Terminal Application

Thus, you can perform administrative tasks using Terminal. You can manually reboot the NVIDIA Jetson Nano Operating System. You can click the Settings icon in the upper right corner of the NVIDIA Jetson Nano desktop.

You can also restart NVIDIA Jetson Nano via Terminal. You can open Terminal by pressing the CTRL and T keys at the same time. After opening Terminal you can type the command: sudo reboot . The NVIDIA Jetson Nano device will automatically reboot. You should make sure to save all data before rebooting.

If you no longer need to use NVIDIA Jetson Nano, you can shut down using the shutdown command, either from the Shutdown option in the confirmation dialog, or using Terminal if needed.

NVIDIA Jetson Nano Programming

NVIDIA provides the option to use the Jetson Nano card in two ways. Projects can be developed by trading with the Linux-based Ubuntu interface or transactions can be made via Jupyter Notebook IDLE using a computer.

There are SDKs required to install software on Jetson Nano:

- JetPack SDK

- DeepStream SDK

JetPack SDK

The NVIDIA JetPack SDK is one of the most comprehensive systems used to build AI applications. It also includes samples, documentation, and developer tools for both the host and developer kit, and supports higher-level SDKs such as DeepStream for video analysis and Isaac for robotics.

There are two options for using the JetPack SDK: in the first option; Without using equipment such as screens and keyboards, jetson nano card can be connected to computer via USB and projects can be realized with Jupyter Notebook interface. In the second option, after installing the JetPack system, jetson nano development card can perform projects by connecting the screen to the HDMI input, keyboard and mouse to the USB ports.

DeepStream SDK

Built for both developers and businesses, deepstream SDK offers seamless development for AI-based video, audio and video analytics with widely used segmentation models such as state-of-the-art SSD, FasterRCNN, YOLO, MaskRCNN.

With Deepstream, multi-GPU, multi-stream, and grouping support, you can get the best possible performance using NVIDIA TensorRT for high efficiency. It offers the possibility to run locally on PyTorch and TensorFlow and PyTorch with DeepStream on models. DeepStream can be developed in languages such as C/C ++ or Python.

Editor Tools

Jetson Developer Kits support a wide range of development environments. Most developers on a Linux platform work with IDE or can be moved to Jetson. One of the most popular IDE's is Visual Studio Code from Microsoft. Visual Studio Code is commonly referred to as VSCode. VSCode offers a free-source version to professional developers and there are many programming language options.

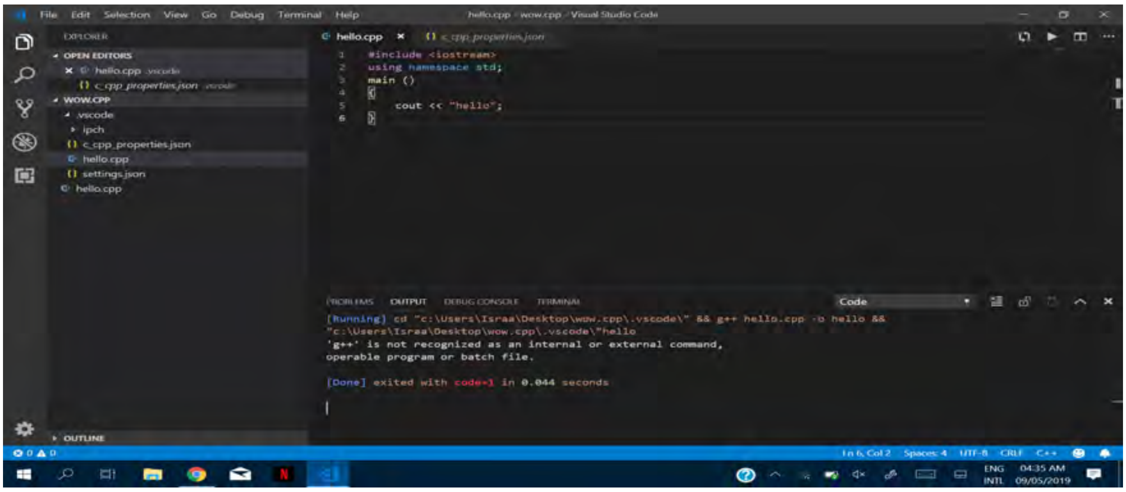
To set up VSCode, you can set up by opening Terminal and typing the following codes:

```
$ sudo -s

$ . <( wget -O - https://code.visualstudio.com/download

)

exit
```



```
#include <iostream>
using namespace std;
main ()
{
    cout << "hello";
}
```

```
[Running] cd "c:\Users\Israa\Desktop\wow.cpp\.vscode\" && g++ hello.cpp -o hello &&
"c:\Users\Israa\Desktop\wow.cpp\.vscode\"hello
'g++' is not recognized as an internal or external command,
operable program or batch file.

[Done] exited with code=1 in 0.044 seconds
```

Figure 12. Visual Studio Vode

C/C++ Programming

You can create programs for C/C ++ on your NVIDIA Jetson Nano. The NVIDIA Jetson Nano image includes the GCC compiler, so you can use it to compile C/C++.

Python Programming

Python is one of the programming languages frequently used by developers in data processing and data science applications. You can also create hardware programs using

Python libraries if necessary.

To write Python scripts in Jetson Nano, you can use visual studio code or jupyter notebook compiler programs after the required Python version is installed.

Node.js Programming

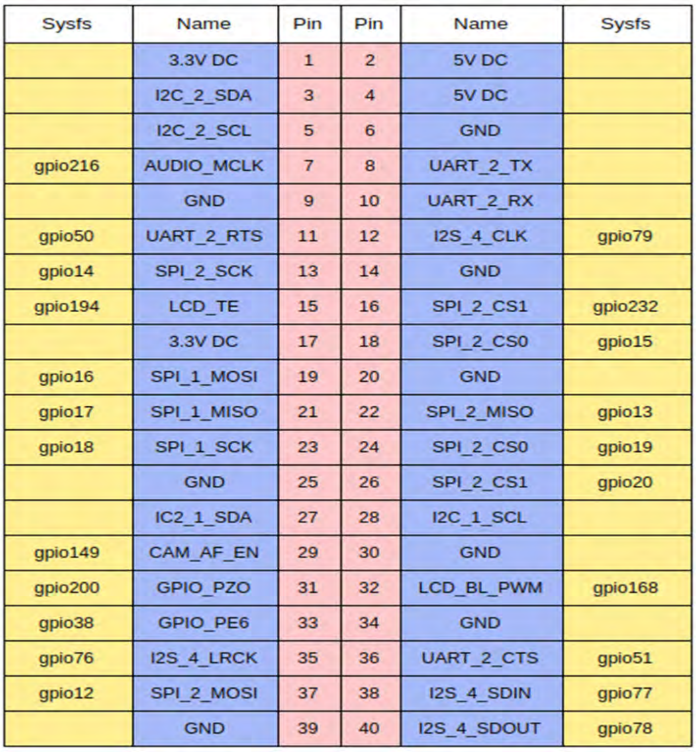
Node.js, a JavaScript-based programming language, is generally used in web applications and is a cross-platform. Nvidia Jetson Nano is easy to use with Node.js.

Jetson Nano I/O Programming

The Jetson Nano Development Kit has a 40-pin expansion head that carries several GPIO pins that can be used to perform microcontroller-level tasks similar to Raspberry Pi boards. These digital pins can give 20 Microampers (µA). Therefore, a transistor is needed to burn an Led. We can add pins and necessary devices and sensors and actuators to NVIDIA Jetson Nano devices. At NVIDIA Jetson Nano, we also mentioned in the introduction that it provides GPIO pins for UART, PWM, SPI, I2S and I2C.

GPIO Support on Jetson Nano Developer Kit

The Jetson Nano Developer Kit has a J41 expansion head with pins similar to the Rasberry Pi. The following pinout diagram shows different uses of pins.



Sysfs	Name	Pin	Pin	Name	Sysfs
	3.3V DC	1	2	5V DC	
	I2C_2_SDA	3	4	5V DC	
	I2C_2_SCL	5	6	GND	
gpio216	AUDIO_MCLK	7	8	UART_2_TX	
	GND	9	10	UART_2_RX	
gpio50	UART_2_RTS	11	12	I2S_4_CLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3V DC	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_2_CS0	gpio19
	GND	25	26	SPI_2_CS1	gpio20
	IC2_1_SDA	27	28	I2C_1_SCL	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

Figure 12. Pinout Diagram for J41 Expansion Head in Jetson Nano Developer Kit

As the diagram shows, other pins except I2C communication pins are directly connected to the Jetson Nano module. I2C pins are connected to a mid-range gear shifter to change the voltages in the module from 1.8V to 3.3V in the standard I2C interface. These are 3 and 5 pins (I2C SDA pins) and 27 and 28 pins (I2C SCL pins). The 8 and 10 pins are the UART transmitter (TX) and receiver (RX) pins, respectively.

The card also provides two 3.3V (pin 1 and 17) and two 5V (pin 2 and 4) power supplies. The expansion head also has multiple floor pins.

Some sources have given that the expansion header only has 2 PWM channels directly connected to hardware PWM controllers, but the Jetson Nano module is not naturally configured to identify the connection to the PWM hardware. Therefore, to use PWM hardware, the system must be configured to provide functionality to PinMux PWM pins. The documentation provided by NVIDIA describes the relevant steps.

References

About NVIDIA, (t.y.). https://www.nvidia.com/tr-tr/about-nvidia/

Barba-Guaman, L., Eugenio Naranjo, J., & Ortiz, A. (2020). Deep Learning Framework for Vehicle and Pedestrian Detection in Rural Roads on an Embedded GPU. *Electronics*, 9(4), 589. http://dx.doi.org/10.3390/electronics9040589.

Salih, T.A., Gh. M.B. (2020). A novel Face Recognition System based on Jetson Nano developer kit, *IOP Conf. Ser.: Mater. Sci. Eng. 928 032051.*

Franklin, D. (2019). Jetson Nano Brings AI Computing to Everyone. Erişim adresi https://developer.nvidia.com/blog/jetson-nano-ai-computing

Jetson Nano Developer Kit. (t.y.). https://developer.nvidia.com/embedded/jetson-nano-developer-kit

Kurniawan, A. (2021). IoT Projects with NVIDIA Jetson Nano: AI-Enabled Internet of Things Projects for Beginners, Apress.

NVIDIA Jetson Linux Driver Package, (2019, 9 October) https://docs.nvidia.com/jetson/l4t/

Hao, T.K. (2020, 14 October). Nvidia Jetson Nano Developer Kit A02 vs B01 vs 2GB. https://tutorial.cytron.io/2020/10/14/nvidia-jetson-nano-developer-kit-a02-vs-b01-vs-2gb/

Jackson, L. (2020, 17 March). Jetson Nano B01 vs A02: What's New for the Compute on Module (CoM) and Carrier Board. https://www.arducam.com/nvidia-jetson-nano-b01-update-dual-camera/

Development Kits Nvidia Jetson Nano. (t.y.). https://www.theimagingsource.com/embedded-vision/development-kits/nvidia-jetson-nano/

NVIDIA Jetson Nano 2GB Developer Kit WiFi. (t.y.). https://openzeka.com/urun/nvidia-jetson-nano-2gb-developer-kit/

NVIDIA Jetson Nano 2GB Developer Kit. (t.y.). https://tr.aliexpress.com/i/1005001827026370.html

Jetson Nano Module. (t.y.). https://developer.nvidia.com/embedded/jetson-nano

Jetson Nano Developer Kit User Guide (2019, 17 November). https://www.veribilimiokulu.com/yapay-zeka-uygulama-gelistirme-karti-nvidia-jetson-nano-2/

DeepStream SDK. (t.y.). https://developer.nvidia.com/deepstream-sdk

How to Use GPIO Pins on Jetson Nano Developer Kit. (2020, 23 June). Erişim adresi https://maker.pro/nvidia-jetson/tutorial/how-to-use-gpio-pins-on-jetson-nano-developer-kit

Nvidia Jetson Nano. (t.y.). https://www.nvidia.com/tr-tr/autonomous-machines/embedded-systems/jetson-nano/

About the Authors

Fatma Nur UZUN, Konya Technical University, Faculty of Engineering and Natural Sciences, Department of Computer Engineering,

E-Mail: fnuruzun4@gmail.com , Orcid: 0000-0003-0153-3442 ,

Mehmet KAYRICI, Necmettin Erbakan University, Faculty of Seydişehir Ahmet Cengiz Engineering, Department of Mechanical Engineering

E-Mail: mkayrici@erbakan.edu.tr, Orcid: 0000-0001-8553-1166

Beyzanur AKKUZU, Necmettin Erbakan University, Faculty of Seydişehir Ahmet Cengiz Engineering, Department of Computer Engineering

E-Mail: beyzaakkuzu@gmail.com, Orcid: 0000-0002-7299-5276

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 15%

To Cite This Chapter:

Uzun, F. N, Kayrici, M. & Akkuzu, B. (2021). Nvidia Jetson Nano Development Kit. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 82–96). ISRES Publishing.

Lattepanda

Mehmet KAYRICI
Necmettin Erbakan University

Ahmet Esref ARIKAN
Konya Technical University

Merve HATIPOGLU
Necmettin Erbakan University

Introduction

Origins of microprocessor and microcontroller, It can be traced back to the MOS integrated circuit, an integrated circuit chip developed in the early 1960s. By 1964, MOS chips achieved higher transistor density and lower manufacturing costs than bipolar chips. The application of MOS LSI chips to computing formed the basis of the first microprocessors as engineers began to realize that a complete computer processing system can be found in several MOS LSI chips.

The first multi-chip microprocessors were developed with more than one MOS LSI chip. The first single-chip microprocessor was the Intel 4004, released in 1971. One of the first recognizable modern embedded systems was developed for the Apollo spacecraft and missiles.

Since these first implementations in the 1960s, the price of embedded systems has fallen and there has been a dramatic increase in processing power and functionality. The first microprocessor, the Intel 4004 (released in 1971), was designed for calculators and other small systems, however, it still needed external memory and support chips. In the early 1980s, memory, input and output system components were integrated into the same chip as the processor that created a microcontroller. Microcontrollers find applications where a general purpose computer would be very costly. As the cost of microprocessors and microcontrollers fell, the prevalence of embedded systems increased ("LattePanda alpha," 2018).

In the historical process of microcomputer production, the 1960s passed with the use of electronic circuit elements and in the 1970s, computers became very fast with the use of microchips created by combining integrated circuits.

There has been a serious increase in applications on embedded systems in recent years. Single board computer systems are frequently preferred in embedded system applications due to their easy-to-use, low cost and small size features. They can communicate with other devices or sensors through communication interfaces. Thus, they are used in robotics, internet of things, health, computer vision and smart home systems. They are also selected for product development in the industrial field and prototype development in the academic field.

After ARM (Advanced RISC Machines) processors, LattePanda development board entered the mini computer market. It is a single board computer technology. This board offers the most advanced possibilities of mini computers. Although there are all kinds of necessary entries on a computer, Windows 10 is installed and operations that cannot be done with Raspberry pi 3 can be done with this card of the same size.

LattePanda card development team started work in 2015. It reached thousands of supporters via the Kicksarter platform in December 2015. Supported by the production of DFRobot, LattePanda team delivered the first generation products in March 2016. LattePanda R&D team continues to support further technology development possibilities. So much so that in 2017, it launched the LattePanda Delta and then the LattePanda Alpha. Today, LattePanda team continues to develop versions of Alpha ("Documentation," t.y.).

While LattePanda only supports Windows operating system in its first version, later versions Alpha 800 and Alpa 864 support Linux operating system. In addition, Alpha, which is released today, uses the same Intel processor class found in the MacBook.

Windows 10 comes licensed and installed. Since LattePanda contains Arduino board, it is a combination of a computer with Windows and an Arduino board. Thus, when there is a need for multimedia and performance, an atom processor, precise timing, PWM, and an Arduino microcontroller for analog input and hardware needs are used (Aydemir, 2018).

It is a combination of the operating systems supported in LattePanda and the Arduino board. It uses the Arduino IDE to program the Arduino and comes with the Arduino program installed. It also enables the use of large applications such as Visual Studio and Office to program the system.

LattePanda is used in many different project areas from robots to security systems, from system programming to playing games. With the 4K support on the device, a wired or wireless keyboard or mouse can be connected and allows surfing on the internet. In addition, since it has an Arduino board, it has the ability to do all the operations that can be done with this card.

By using LattePanda, there is an opportunity to become a Windows developer, IoT developer, interactive designer, robotics expert. In addition to the internal graphics card in the card, the desired card can be installed. The ability to install a second operating system on the card is another advantage of LattePanda. It supports the full version of Ubuntu as well as the full version of Windows 10 ("LattePanda alpha," 2018).

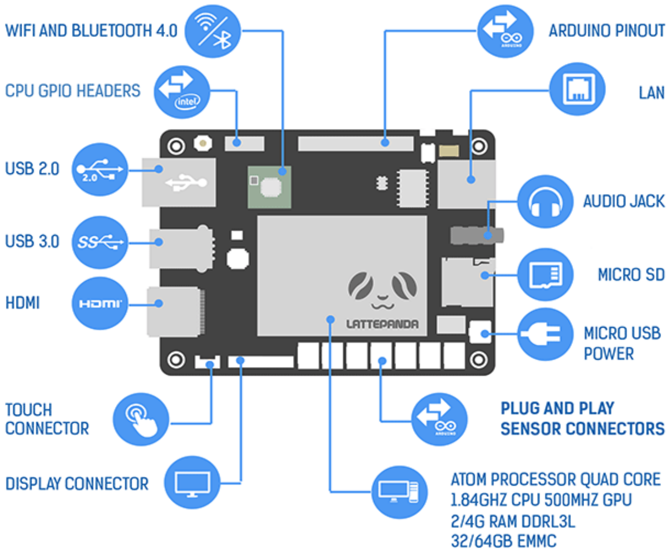


Figure 1. LattePanda Components

LattePanda Cards and Features

Figure 2. LattePanda Alpha

LattePanda is a complete Windows 10 single board computer. It has everything a normal computer has and can do everything a normal computer does. It is compatible with almost every device you know: printers, joysticks, cameras and more. All peripherals running on computers will run on a LattePanda.

LattePanda is a high performance palm-sized single board computer with low power consumption, running the full Windows 10 or Linux operating system. Edge computing, vending machine, advertising machine, industrial automation etc. It is widely used in its fields.

Whether you are a Windows developer, Internet of Things developer, system integrator or solution provider, LattePanda is the powerful development board that can speed up your production rate.

LattePanda Alpha and Delta are the latest versions. While the former has top performance, the latter has a perfect balance of design, performance and price. Besides the latest motherboard, the LattePanda V1 is a good choice for industrial customers given its maturity and stability. In addition, various accessories such as touch screens, cables, cooling fans, cases and more can be supported.

LattePanda comes preinstalled with the full version of Windows 10 Home Edition so it can run on powerful tools like Visual Studio, NodeJS, Java, Processing and more. With existing APIs, original software and hardware projects can be developed on a LattePanda just like on a normal PC. (C#, Javascript, Ruby, etc.)

A LattePanda also includes an integrated Arduino compatible co-processor, which means it can be used to control and sense the physical world. Whether you are a Windows developer, IoT developer, DIY fanatic, interactive designer, robotics expert or maker, a LattePanda single board computer can assist in creative processes.

In terms of processor, all versions of LattePanda now come with an upgraded CPU. (Intel Z8350 - up to 1.92GHz)

LattePanda has three different versions. These versions are 2G / 32GB, 2GB / 32GB and 4G / 64 GB respectively. In addition, the 4G / 64GB version offers support for 64-bit Windows 10 used on personal computers. Due to the 4K support of the device, it can also be used with a wired or wireless keyboard and mouse. While Windows cannot be installed on the Raspberry PI, which is the closest Arduino board, this is the biggest privilege of this device. However, despite all this, of course, it would be wrong to expect a performance like a normal computer from this device. However, it can be used easily in many different projects from robots to security systems. If desired, different operating systems such as Linux can be installed. It even supports the installation of a second operating system.

Technical features of LattePanda Alpha;

- Processor Intel Core 7 generations M3-7Y30, 2.6GHz
- 8GB dual channel RAM
- Compatibility with NVMe SSDs
- Supports Intel Dual Band Wireless-AC 3165 2.4G / 5G Wifi and Bluetooth V4.2
- 1.6-2.6GHz Dual Core

- Intel HD Graphics 615, 300-900MHz
- 1x M.2 M Key, supports PCIe 4x, NVMe SSD and SATA SSD
- 1x M.2 E Key, supports PCIe 2x, USB2.0, UART, PCM

Delta Edition uses 8th generation Celeron N4100 processor, features as the main robot controller, interactive project core, IoT edge device or AI brain. LattePanda Delta is an x86 based SBC design. It has dual operating system support. It allows efficient code writing with the advantage of more than one operating system. Thanks to its ultra-thin design, it minimizes the space holding. It also has superior networking ability in the cloud and objects.

Technical features of LattePanda Delta;

- CPU : Intel 8th Generation Celeron Processor N4100
- Core : 1.1-2.4GHz Quad Core , Four Thread
- Benchmark (PassMark): up to 1800+
- Graphics :Intel UHD Graphics 600, 200-700MHz
- RAM : 4G LPDDR4 2400MHz Dual Channel
- Memory : 32GB eMMC V5.0l
- External memory:
- 1x M.2 M Key, supports PCIe 2x, NVMe and PCIe SSD
- 1x M.2 E Key, supports PCIe 2x , USB2.0, UART
- Connectivity , WIFI 802.11 AC, 2.4G and 5G
- Dual Band Bluetooth 5.0
- Gigabyte Ethernet
- USB Ports :3x USB 3.0 Type A
- Supports 1x USB Type C, PD, DP, USB 3.0
- Display : HDMI Output
- Type-C DP Support

- Expandable eDP touch screens
- Co-processor :Arduino Leonardo
- Operating System Support : Windows 10 Pro Linux Ubuntu

Picture					
Name	LattePanda Delta 432	LattePanda Delta 432	LattePanda Alpha 800s	LattePanda Alpha 864s	LattePanda Alpha 864s
Storage	4GB+32GB	4GB+32GB	8GB+0GB	8GB+64GB	8GB+64GB
License	NO	YES	NO	NO	YES
CPU	Intel Celeron N4100	Intel Celeron N4100	Intel Core M3-8100Y	Intel Core M3-8100Y	Intel Core M3-8100Y
Price	**$188**	**$228**	**$379**	**$409**	**$449**

Figure 3. Comparison Table of LattePanda Models

1. LattePanda Setup (“Documentation,” t.y.)

There are two different power supply interfaces to power the LattePanda 1st Generation. The first is the widely used microUSB port. The other is any of the 5V GND (ground) pins, all labeled “CN2 Header Pins” on the board.

MicroUSB port is more suitable for desktop development scenarios.

The CN2 Header Pin is a good power supply for embedded applications that have limited space but provide a more stable, more powerful power supply for the system (with higher amps to supply enough current at about 3 ~ 4 Amps to sense the system simultaneously).

The current required for a LattePanda with full compute load is about 1.6 Amps. However, when restarting the system, the required peak current is about 2 Amps.

Requiered External Hardware

- USB Wall Adapter up to 2A Output
- Quality MicroUSB Cable (capable of 2A current)
- HDMI cable
- Cooler or Active Cooling Fan (required for advanced applications)
- Other General Purpose Computer Peripherals

- Monitor with HDMI Port or MIPI Display from LattePanda
- Keyboard and Mouse (Windows Ease of Access keyboard that comes preinstalled on your Windows OS device can also be used if it has a touchscreen)
- The LattePanda is powered by the microUSB port. Any standard USB adapter (such as mobile phone wall charger) with at least 2A current can be used as a power source for the LattePanda. A power adapter is not included with the motherboard.

Plug the USB into the USB power adapter and the microUSB into the LattePanda's microUSB port. (The microUSB port is next to the SD card slot).

1. If there is a need for IPS screen and touch panel, please install it first. Figure 3.

Figure 4. Installing the Display Plug

2. When inserted, you should see the red LED indicator light on the underside of the board. This means LattePanda is launched. Wait patiently for a few seconds until the LED goes out. Figure 4.

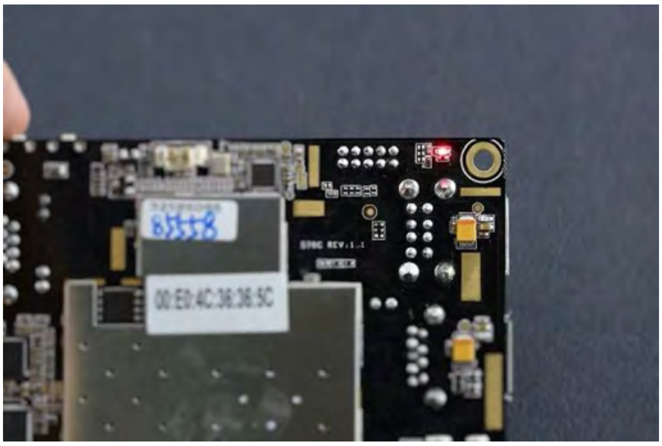


Figure 5. Launching LattePanda

3. When the LED is off, press and hold the power button for one second to turn on the LattePanda. You should see the LED light up again. Figure 5.

Figure 6. Opening LattePanda

4. Connect peripheral device to your device.

LattePanda is compatible with a wide variety of peripheral devices. You can connect any USB powered device such as flash drives, mouse and keyboard, or a webcam to USB 3.0 and 2.0 ports. SD card socket supports extra storage from miniSD card. You can also connect the LattePanda to an external speaker device via the 3.5mm audio jack.

LattePanda's Arduino-compatible co-processor with plug-and-play heads and GPIO pins supports standard 5V sensors and actuators that allow it to interact with the physical world.

Connect to the Wi-Fi network:

1. Install the Wi-Fi antenna by inserting the rounded tip into the socket labeled "ANT" located next to the GPIO pins on the board.

2. In Windows, select a Wi-Fi connection by clicking the Wi-Fi icon in the system tray at the bottom right of the screen. Follow the wizard to make a connection.

3. Connect via 7" Screen and Touch Panel Layout. Figure 7.

Figure 7. Connecting to Windows

Contacts in the FPC are very compact and sequential. Please note that any dislocation connection may cause LattePanda short circuit and the IPS may cause abnormal display such as ghosting or flickering.

1. Lift the actuator. Figure 8

Figure 8. Actuator Connection

2. Place the display FPC.

3. Place the Gold Finger side down! Figure 9

Figure 9. FPC Connection

4. Turn the actuator down until it is tightly closed. Figure 10

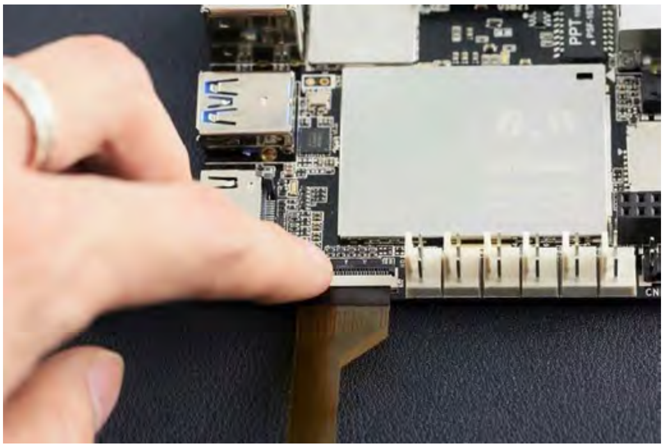


Figure 10. Socket Locking

5. Install the FPC of the touch panel same as the screen. Figure 11

Figure 11. Display and Panel FPC Connection

5. LattePanda Applications

Analyzes and experiments need to be portable to meet mobile data collection needs in the medical field. One of the key advantages here is LattePanda, a Windows 10 computer. Data from this device provided the same reliability as laptops. This device is inexpensive compared to notebook computers and provides a richer variety of media for experiments and analysis (Kuziek et al., 2018).

In recent years, data sharing with the IoT technique has become more attractive in the healthcare industry. Single board computers play a big role in data collection and sensor management. LattePanda is the single board computer that is cheaper, faster and capable of large data storage (HadiShanoer et al., 2020).

It can be used in LattePanda as a high performance card that will support the operations required by security and object recognition technology (Nacipucha et al., 2020).

A single board computer is a complete computer built on a single circuit board with microprocessor, memory, input / output (I / O) and other features required by a fully functional computer. Single board computers can be used to create rapid development systems, various education systems and as embedded robot controllers. Most single board computers have great connectivity options and peripherals not common for desktop / server PC systems.

LattePanda performs well but has various setup and stability issues. The multimedia playback experience is far behind other cards. Also, high heat dissipation and fan requirements are disadvantage for this SBC. Of course, it can fit some niche applications when computing requirements are important, the X86-64 architecture allows all PC-based operating system software packages to be used (Paunski & Angelov, 2019).

The LattePanda motherboard is a Windows-based edge computer designed to run

Windows 1064 bits on a 5inch motherboard. It has an Intel Atom microprocessor and 4GB of RAM memory that allows it to run different applications such as Matlab or run as a server for networked system applications. For the remote lab platform, the LattePanda board is used not only to run Matlab in Loop-in-Hardware configuration (HIL) for the system's local control scenario, but also as a installer for the remote control scenario. In addition, the Arduino card connected to the LattePanda provides great convenience and performance in applications (Viola et al., 2020).

The data obtained from the image and distance sensors used in object recognition technology was developed on a single board computer (LattePanda), which is the most useful for mobile applications, and the software was developed on a C ++ platform (Visual Studio 2017, Microsoft, USA). This is a great advantage for agricultural applications (SooKim et al., 2021).

Devices such as Intel LattePanda, which allow a large number of weak computing devices to share the workload in simulation applications, generally have lower cost and higher power efficiency (Rui et al., 2019).

Results

In a time when mobility is very high in the world of single board computers, the relatively small but high performance LattePanda card has become important in all industrial applications. The most important output of this board is its price advantage, performance and powerful output pins, as well as the ability to manage computers by connecting remotely and to make additional and complex applications in a simple way thanks to the Arduino board on it. Of course, it is obvious that it will not be sufficient for every application, as is the case with every card. However, as the studies on it increase, the card can be improved. It has high application performance for now.

References

What is LattePanda alpha? (2018, 1 March). https://www.kaizen40.com/LattePanda-alpha-nedir/

Single-board computer. (2021, 20 April). https://en.wikipedia.org/wiki/Single-board_computer

Doğru Bolat, E., Solak, S., Yakut, Ö. (2017). Yaygın kullanılan ARM tabanlı tek kart bilgisayar sistemleri ve kullanım alanları. *El-Cezeri*, *4* (1), 11-24. https://Doi:10.31202/ecjse.289633

Aydemir, E. (2018). *LattePanda ile arduino ve pc kodlama*. Konya: Eğitim Yayınevi

Documentation. (t.y.). http://docs.LattePanda.com/

LattePanda Alpha and Delta Series. (t.y.). https://www.LattePanda.com/products/LattePanda-delta-432.html

Kuziek, J.W.P., Redman, E.X., Splinter, G.D., Mathewson, K.E. (2018). Increasing the mobility of EEG data collection using a Latte Panda computer. *Journal of Neuroscience Methods*, 308 (1), 34-47.

HadiShanoer, H., Hassan, H.S., A. Abdul-Rahaim, L. (2020) Performance Analysis of IoT based Health Monitoring System using LattePanda Single Board Computer. *Solid State Technology*, 63 (1).

Nacipucha, N., César, A., Frías, P., Joel, J. (2020). *Diseño de un Prototipo de Control de Acceso a través de Reconocimiento Facial Mediante la Creación de un Algoritmo Basado en Software Libre Utilizando lattepanda*. Repositorio Institucional de la Universidad Politécnica Salesiana. http://dspace.ups.edu.ec/handle/123456789/19626.

Paunski, Y.K., Angelov, G.T. (2019). Performance and power consumption analysis of low-cost single board computers in educational robotics. *IFAC-PapersOnLine*, 52 (25), 424-428.

Viola, J., Oziablo, P., Chen, YQ., (2020). A Portable and Affordable Networked Temperature Distribution Control Platform for Education and Research. *IFAC-PapersOnLine*, 53 (2), 17530-17535.

SooKim, V., HyunLee, D., JooKim, Y., Kim, T., SukLee, W., HyunChoi, C. (2021). Stereo-vision-based crop height estimation for agricultural robots. *Computers and Electronics in Agriculture*, 181, 105937.

Rui, M., Chuanzhi, G., Jerry, Y.Z., Jiayu, C. (2019). Modeling city-scale building energy dynamics through inter-connected distributed adjacency blocks. *Energy and Buildings*, 202, 109391.

About the Authors

Mehmet KAYRICI, Necmettin Erbakan University, Faculty of Seydişehir Ahmet Cengiz Engineering, Department of Mechanical Engineering, Orcid No: 0000-0001-8553-1166 , E-Mail: mkayrici@erbakan.edu.tr

Ahmet Esref ARIKAN, Konya Technical University, Engineering and Natural Sciences, Email: ahmetarikan.r@gmail.com

Merve HATIPOGLU, Necmettin Erbakan University, Seydişehir Ahmet Cengiz Engineering

E-mail: ahmetarikan.r@gmail.com, Orcid No: 0000-0002-7299-5276

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 15%.

To Cite This Chapter:

Kayrıcı, M. , Arıkan, A. E. & Hatipoglu, M. (2021). Lattepanda. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 97–109). ISRES Publishing.

ESP8266 and ESP32 Series of SoC Microcontrollers

Hakkı SOY
Necmettin Erbakan University

Introduction

Nowadays, Internet of Things (IoT) heavily permeates the daily lives of humans through smart devices that can be monitored or controlled remotely. Basically, IoT contains the embedded devices can be accessed through Internet to meet the users' demand coming from separate applications (Al-Fuqaha et al., 2015). In parallel to increasing penetration of IoT-enabled devices in various domains, several manufacturers have started to produce the RF transceiver chips and modules with different protocols. Espressif Systems is one of the main actors in IoT market which was founded in 2008, based in Shanghai, China. Espressif offers well-integrated, low-power, low-cost, high performance, energy-efficient wireless IoT chips and modules that are widely used in mobile devices, home appliances and industrial applications. The well-known products from Espressif are the ESP8266 and ESP32 series chips, modules and development boards. Espressif uses a system-on-chip (SoC) architecture, which refers a single integrated circuit (IC) composed of multiple components. Table 1 shows the specifications of the ESP8266 and ESP32 Series SoCs (Espressif, 2021a).

Table 1. The Technical Specifications of the SoCs from Espressif Systems.

SoC	Core	MCU	Wi-Fi	BLE	Pins	RAM	ROM	Flash
ESP32-C6	Single	160 MHz	2.4 GHz	5.0	32	400 KB RAM	384KB	--
ESP32-S3	Dual	240 MHz	2.4 GHz	5.0	56	512 KB SRAM	384 KB	--
ESP32-S2	Single	240 MHz	2.4 GHz	--	56	320 KB SRAM	128 KB	--
ESP32-C3	Single	160 MHz	2.4 GHz	5.0	32	400 KB RAM	384 KB	4 MB
ESP32-D0WD-V3	Dual	240 MHz	2.4 GHz	4.2	48	520 KB SRAM	448 KB	--
ESP32-D2WD	Dual	160 MHz						2 MB
ESP32-U4WDH	Single	160 MHz						4 MB
ESP32-S0WD	Single	160 MHz						--
ESP32-PICO-V3	Dual	240 MHz						4 MB
ESP32-PICO-D4	Dual	240 MHz						4 MB
ESP8266EX	Single	160 MHz	2.4 GHz	--	32	160 KB RAM	--	--

ESP8266

ESP8266 (2014), officially known as ESP8266EX, is a Wi-Fi enabled SoC that integrates Cadence Tensilica L106 32-bit RISC processor with a TCP/IP stack. It is also embedded with memory controller, including on-chip SRAM and ROM. Figure 1 shows the building blocks of the ESP8266 SoC (Espressif, 2021b; Kolban, 2016).

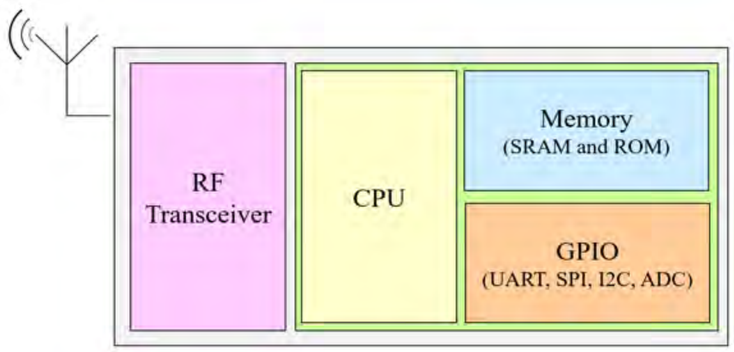


Figure 1. Building Blocks of the ESP8266 SoC.

ESP8266 SoC has a quad-flat no-leads (QFN)-32 package with 5 x 5 mm size, which means that there are no external pins, but instead there are pads at the bottom of the chip. ESP8266 SoC pinout diagram is shown by Figure 2. The processor communicates with the peripherals through standard interfaces, i.e., UART, SPI (Espressif, 2020a).

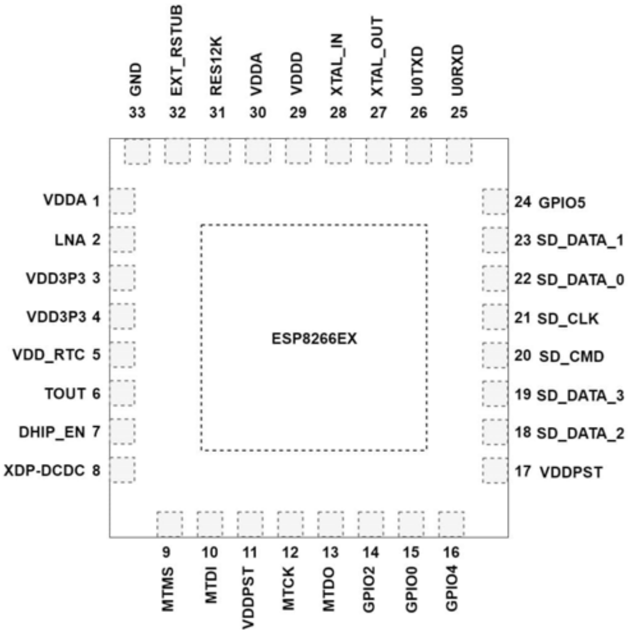


Figure 2. ESP8266 SoC Pinout Diagram

ESP8266 SoC typically draws about 80 mA in idle state and 170 mA while in operation. The operating voltage ranges from 2.5 V to 3.6 V DC. It also supports the light sleep, modem sleep and deep sleep modes with current of 0.5 mA, 15 mA and 0.1 mA, respectively. The peak operating current can reach to 320 mA in “boot up” mode or ‘wakes up’ from a sleep mode. When compared with ordinary Wi-Fi adapters, the ESP8266 SoC has a relatively lower power consumption that ensures the battery life can be typically up to 3 months (Espressif, 2020b).

Although ESP8266 is the name of the basic chip, it can be also bought in two different varieties including the module and development board, as shown in Figure 3. ESP8266

SoC is manufactured unshielded and its package is usually used by soldering onto a circuit board. So, it is unsuitable for hobbyists and it is recommended to be used as a module when mass production is not required. The ESP8266 modules are surface-mountable and enable the microcontrollers to connect to the Internet by using IEEE 802.11 b/g/n Wi-Fi standards with WPA/WPA2 certifications. On the other hand, the development boards are produced by several manufacturers with different specifications to develop IoT projects rapidly.

Figure 3. ESP8266 SoC, Module (ESP-01) and Development Board (NodeMCU v3).

Today, Espressif produces the ESP-WROOM-02 (Espressif, 2020c) and ESP-WROOM-S2 (Espressif, 2020d) modules, based on ESP8266 SoC. Besides, several third party manufacturers started to produce custom modules and boards based on the Espressif's ESP8266 SoC as the main controller. Ai-Thinker is an officially licensed manufacturer of Espressif modules that are labeled ESP-XX. The ESP-01 is the most basic and cheapest Wi-Fi SoC made by Ai–Thinker in August 2014 (Ai-Thinker, 2018a). Thanks to small form factor, it fits into any enclosure that has limited spacing. However ESP-01 module has only two digital general-purpose input/output pins (GPIO0 and GPIO2) and no analog pin. It can be integrated with the sensors and other application specific devices via GPIO pins. Besides, two additional pins (GPIO1-Tx and GPIO3-Rx) are available for serial communication. The pinout diagram of ESP-01 module and its breadboard adapters are shown by Figure 4.

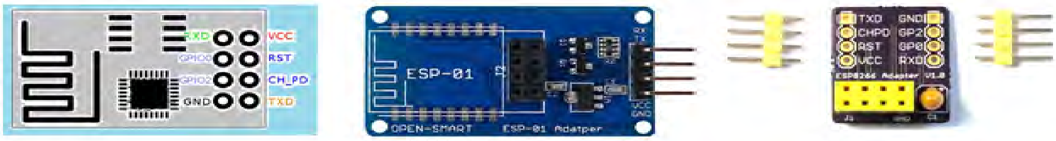


Figure 4. ESP-01 Module Pinout Diagram and Two Different Breadboard Adapters.

In response to high demand for ESP-01 module, updated versions from ESP-02 to ESP-14 were released by Ai-Thinker. All of these modules are designed for the needs of IoT applications and they are working on the ESP8266 SoC. A typical ESP-XX module enables Internet connectivity that can be required by host applications and to create a Wi-Fi network that can be used to exchange data and instructions. But, the main differences are the number of GPIO pins, pitch (space between pins), form factor and antenna connection type. The specifications of available modules are given in Table 2.

Table 2. The ESP8266 Modules Based on ESP8266 SoC.

Vendor	Module	GPIO	Pitch	Form factor	Antenna
Espressif	ESP-WROOM-02	18	1.5 mm	2×9 castellated	PCB trace
	ESP-WROOM-02D	18	1.5 mm	2×9 castellated	PCB trace
	ESP-WROOM-02U	18	1.5 mm	2×9 castellated	U.FL socket
	ESP-WROOM-S2	20	1.5 mm	2×10 castellated	PCB trace
Ai-Thinker	ESP-01	6	0.1 in	2×4 DIL	PCB trace
	ESP-02	6	0.1 in	2×4 castellated	U.FL socket
	ESP-03	10	2 mm	2×7 castellated	Ceramic
	ESP-04	10	2 mm	2×4 castellated	External
	ESP-05	3	0.1 in	1×5 SIL	U.FL socket
	ESP-06	11	misc	4×3 dice	External
	ESP-07	14	2 mm	2×8 pinhole	Ceramic + U.FL socket
	ESP-08	10	2 mm	2×7 castellated	External
	ESP-09	10	misc	4×3 dice	External
	ESP-10	3	2 mm	1×5 castellated	External
	ESP-11	6	1.27 mm	1×8 pinhole	Ceramic
	ESP-12	14	2 mm	2×8 castellated	PCB trace
	ESP-13	16	1.5 mm	2×9 castellated	PCB trace
	ESP-14	22	2 mm	2×8 castellated +6	PCB trace

ESP-07 is an improved version of the ESP-01 module with more functionalities (Ai-Thinker, 2018b). It comes with a UFL connector that enables to connect an external antenna in case of need to boost the received Wi-Fi signal. ESP-07 module has 16 pins and 11 of them operate as a GPIO pins. It has also an analog input pin with an input range of 0 to 1V. GPIO 1 and GPIO 3 pins are used for TX and RX of serial communication interface. Besides, SPI interface works on GPIO12 (MISO), GPIO13 (MOSI) and GPIO14 (CLK) pins. Figure 5 shows the ESP-07 and its pinout diagram. Since the ESP-07 has a 2.0mm spacing between the pads, it is not breadboard friendly. But, it can be transformed to a breadboard compatible form through a PCB shield with 2.54 mm spacing between the pins. This makes the prototyping easier for lots of cases. Also, it is possible to program the ESP-07 module via FTDI programmer based on USB to TTL serial converter module, i.e., FT232RL chip.

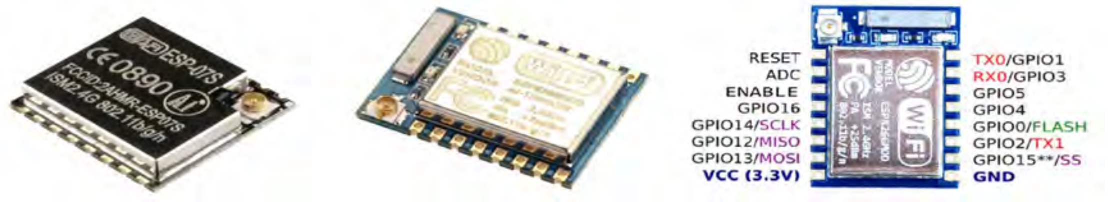


Figure 5. ESP-07 Module and Its Pinout Diagram.

When more pins are needed in the application to be implemented, ESP-12 module is an ideal choice to provide reliable Wi-Fi connectivity. ESP-12 is now outdated, although it rarely exists on the market. ESP-12E (Ai-Thinker, 2015) and ESP-12F (Ai-Thinker, 2018c) modules are the improved versions of the ESP8266-12. They are basically identical to each other, with the difference that ESP-12F has a better antenna design which gives a longer range. ESP-12E includes 11 GPIO pins, analog input serial communication and SPI interfaces. Besides, ESP-12S (Ai-Thinker, 2019a) module is a compact version of the ESP-12F, which has only 9 GPIO pins. Figure 6 shows the ESP-12 series modules and the pinout diagram of popular ESP-12E module. As seen in ESP-01 module, GPIO0 and GPIO2 pins are used to control the bootloader mode. These pins should be taken into account when preparing the application firmware.

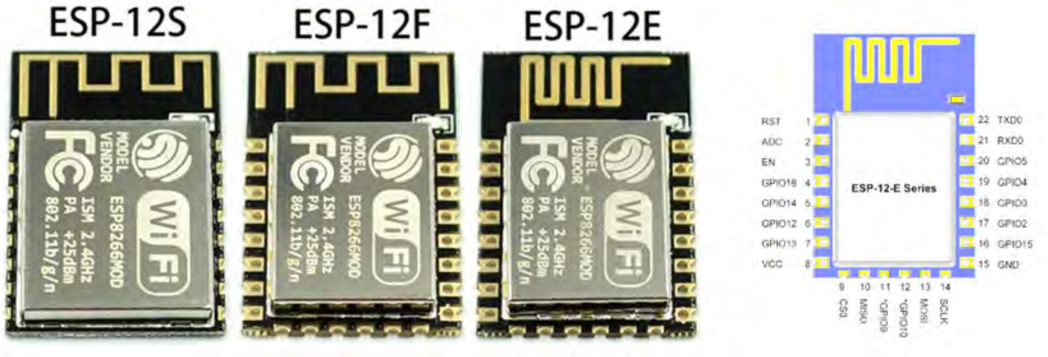


Figure 6. ESP-12 Series Modules and ESP-12E Pinout Diagram.

When a developer does not have enough experience working with ESP8266 module, employment of a development board is a more convenient for complex IoT projects. There are wide variety of ESP8266 development board options are available in the market. NodeMCU is a widely used development board, which is based on ESP8266 as well as ESP32 SoC. It makes the prototyping stage much easier. As shown in Figure 7, NodeMCU boards currently produced by three different producers, namely Amica, LoLin and DOIT.

Figure 7. NodeMCU Development Boards from Amica, DOIT and Lolin.

A typical NodeMCU board has got two more chips for USB to serial converter and voltage regulator from 5V to 3.3V. According to their hardware configurations, NodeMCU development boards have been separated into two different generations and three different versions, currently ranging from v1 to v3. The first generation NodeMCU v1 board has CP2102 USB to UART bridge chip, but it is currently outdated. The second generation NodeMCU v2 and NodeMCU v3 boards are produced by Amica/DOIT and LoLin, respectively. Note that, while the breadboard friendly NodeMCU v2 boards have CP210x, the larger sized NodeMCU v3 is equipped with CH34x USB to serial chip

(Stör, 2015). The pinout diagrams of different version NodeMCU boards are shown by Figure 8.

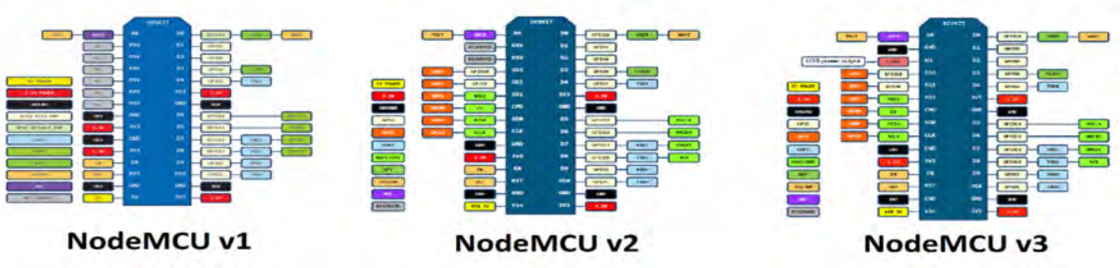


Figure 8. Pinout Diagrams of Different Nodemcu Development Board Versions.

NodeMCU boards can be supplied with the on-board USB connector. They have an onboard LED (connected to GPIO2 pin) and two buttons for Flash and Reset operations. The build-in blue LED is programmable by users to test the certain routines in the implemented algorithm. Flash button is used to download a new program and reset button restarts the microcontroller operation. It is easy to program the NodeMCU board by directly plugging it into the computer via Micro-B type USB connector. Additionally, it is possible to update the firmware of ESP8266 modules wirelessly by using Over-The-Air (OTA) programming feature. Clearly, the OTA functionality also allows uploading a new program to ESP8266 module by using Wi-Fi connection without requiring to connect the NodeMCU board to a computer via USB cable. As long as they share the same network, the firmware of multiple ESP8266 modules can be updated simultaneously from a central server (Bakolia, 2019).

ESP32

ESP32 (2016) is a hybrid Wi-Fi and Bluetooth SoC that integrates Tensilica Xtensa LX series 32-bit RISC processor with a TCP/IP stack. As a successor of the ESP8266, ESP32 series SoCs have dual core processor, more GPIO pins, faster Wi-Fi and BLE protocol support (Kurniawan, 2019). The processor communicates with the peripherals through standard interfaces, i.e., UART, SPI and I2C. In normal (active) mode, ESP32 SoC requires 160-260 mA current to operate at 3.3 V DC supply voltage. When Wi-Fi and Bluetooth functions are open, the current drawn can spike up to 790 mA. Although the ESP32 SoC is relatively power-hungry, the current drawn can be cut down by leveraging one of its sleep modes (modem sleep, light sleep, deep sleep and hibernation). Figure 9 shows the ESP32 SoC housed in QFN-49 package and its pinout diagram (Kolban, 2018).

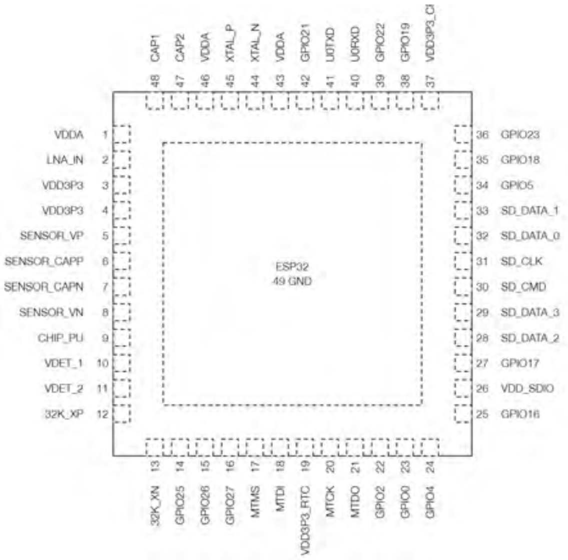


Figure 9. ESP32 SoC Pinout Diagram

ESP32 can be bought as a chip as well as module or development board forms, as shown in Figure 10. There are a number of variants of basic ESP32 SoC (ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD and ESP32-U4WDH) have been introduced to meet the different needs of embedded system developers (Espressif, 2021c). Nevertheless, Espressif Systems has been upgraded the ESP32 SoC in 2020. The latest ESP32 V3 SoC, is known as ECO V3, offers a new silicon wafer as well as the secure boot and flash encryption. The ESP32-D0WD-V3, ESP32-D0WDQ6-V3 and ESP32-U4WDH SoCs are produced based on ECO V3 wafer (Espressif, 2020e). It is noteworthy that all of the ESP32 SoC variants share the same software development tools and they are largely code-compatible. So, the firmware developed for one particular ESP32 SoC can be run on all others without significant modification.

Figure 10. ESP32 SoC, Module (ESP32-WROOM-32) and Development Board (NodeMCU ESP32-S).

The specifications of the advanced ESP32 SoC versions can be summarized as follows:

- ESP32-S2 (2019, 43 programmable GPIOs) is a single-core Xtensa LX7 processor which has only Wi-Fi 4 (802.11 b/g/n) connectivity (Espressif, 2021d).
- ESP32-S3 (2020, 44 programmable GPIOs) has a dual core XTensa LX7 processor with integrated Wi-Fi 4 and Bluetooth 5 (LE) connectivity. It has been designed to address the needs of the Artificial Intelligence of Things (AIoT) applications

(Espressif, 2021e).

- ESP32-C3 (2020, 22 programmable GPIOs) is a reduced version of the ESP32 and it has single-core RISC-V processor with Wi-Fi 4 and Bluetooth 5 (LE) connectivity (Espressif, 2021f).

- ESP32-C6 (2021, 22 programmable GPIOs) integrates a single core RISC-V processor with Wi-Fi 6 (802.11ax) and Bluetooth 5 (LE) connectivity. Thanks to the 802.11ax mode support, ESP32-C6 offers longer transmission range and lower power consumption for battery-driven IoT devices. Furthermore, it supports both the OFDMA (Orthogonal Frequency Division Multiple Access) and MU-MIMO (Multi User, Multiple Input Multiple Output) antenna techniques to provide the high spectral efficiency and low transmission latency in congested environments (Espressif, 2021g).

ESP32 modules offer an extended flexibility and customization opportunity in the revision of firmware for embedded system applications. ESP32-WROOM-32 is a popular module that can be used a wide variety of IoT applications. It is equipped with Wi-Fi and Bluetooth capabilities powered with ESP32-D0WDQ6 SoC and 4 MB integrated SPI Flash memory. It integrates a rich set of peripherals through ADC, DAC, SPI, UART, I²S, I²C, PWM, SD card and Ethernet interfaces (Espressif, 2021h). ESP32-WROOM-32 module is produced with different antenna types. While ESP32-WROOM-32U has a connector that allows us to connect an external antenna, ESP32-WROOM-32D, ESP32-WROOM-32E and ESP32-WROOM-32UE modules come with MIFA, PCB and IPEX antennas, respectively. Different ESP32-WROOM-32 module variants are shown in Figure 11.

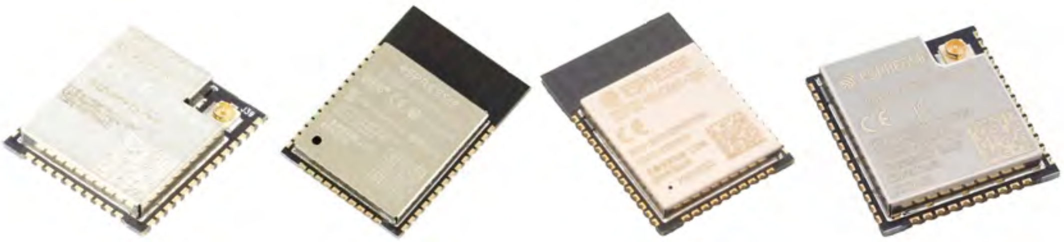


Figure 11. U, D, E and UE Variants of ESP32-WROOM-32 Module.

There is also another ESP32 module variant from Espressif Systems, namely ESP32-WROVER, which has dual core ESP32-D0WDQ6 SoC and its clock frequency can be adjustable from 80 MHz to 240 MHz. The distinctive feature is that it has an additional 8 MB Pseudo-Static RAM (PSRAM) besides the 4 MB external SPI Flash memory. Note that the ESP32-WROVER-B version is equipped with ESP32-D0WD SoC and it offers 8 MB or 16 MB Flash memory options when the customer needs more resources (Espressif, 2021i). The front and back views of ESP32-WROVER and WROVER-B modules are shown by Figure 12.

Figure 12. ESP32-WROVER and WROVER-B Modules.

On the other side, Ai-Thinker produces the ESP32-S which is an original ESP32 series module with dual core Xtensa 32-bit LX6 processor. The clock frequency is adjustable from 80 MHz to 240 MHz. It supports UART, SPI, SDIO, I2C, PWM, I2S, IR, ADC and DAC interfaces with 22 GPIO pins. It also supports the Wi-Fi (802.11 b/g/n/e/i) and Bluetooth (dual mode BR/EDR + BLE v4.2) connectivity. In terms of hardware capabilities, ESP32-S can be considered as equivalent to ESP-WROOM-32 module of Espressif Systems. Note that ESP32-S module comes with on-board PCB antenna and IPEX connector for external antenna (Ai-Thinker, 2016). The front and back views of ESP32-S module are shown by Figure 13.

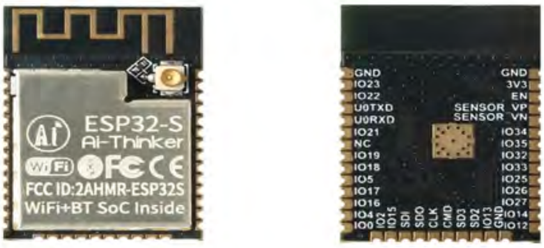


Figure 13. The front and back Views of Ai-Thinker ESP32-S Module.

When considering the challenges of working with chips and modules, the ESP32 SoC based development boards help embedded designers to rapidly prototype and test their IoT applications. Ai-Thinker NodeMCU ESP32-S is a compact prototyping board, which has 38 programmable GPIOs and built-in 2-channel 12-bit high-precision ADC with up to 18 channels (Ai-Thinker, 2019b). The pinout of the NodeMCU ESP32-S board can be seen in Figure 14.

Figure 14. Ai-Thinker NodeMCU ESP32-S Development Board Pinout Diagram.

Programming the ESP8266 and ESP32 SoC Microcontrollers

Espressif provides an official software development framework, called as Espressif ESP-IDF (IoT Development Framework), to develop the IoT applications with wireless connectivity (Espressif, 2021j). The ‘Getting Started Guide’ can be seen from the below link:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started

It is possible to download the ESP-IDF from GitHub repository:

https://github.com/espressif/esp-idf

Although it has complex programming environment, ESP-IDF offers maximum functionality and compatibility. ESP-IDF contains the ESP32-specific API (software libraries and source code), but it requires a toolchain to build the application. Eclipse is an ideal development environment. ESP-IDF also requires some prerequisite tools to build firmware for supported SoCs. ESP-IDF Tools Installer is simple solution to install the prerequisites. It can be downloadable from the following link:

https://dl.espressif.com/dl/esp-idf/

Espressif has also been developed the ESP-AT firmware based on ESP-IDF, which contains a set of AT commands to integrate connectivity for IoT applications. AT commands is the most primitive way to program the Espressif SoC microcontrollers (Espressif, 2021k). The ESP-AT firmware has in-built TCP/IP stack and data buffering functions. By this way, it is possible to quickly join the wireless network, connect to the cloud platform, realize data transmission and remote control functions and realize the interconnection of everything through wireless communication easily. The host microcontroller is connected to the ESP SoC via UART interface to sends/receives AT commands/responses. The following link contains several firmware files in binary format:

https://www.espressif.com/en/support/download/at

Each ESP-AT-Bin file contains several binaries for some specific functions and the binary file in factory folder is the combination of all binaries. The guide in (Espressif, 2020f) demonstrates how to download AT firmware and flash it into an ESP SoC microcontroller.

Furthermore, ESP-IDF allows the FreeRTOS to efficiently use of the processor(s) and manage the built-in peripherals. Both the ESP8266 and ESP32 SoCs are capable of support the FreeRTOS, which is a light-weight RTOS (Real-Time Operating System). Thanks to the RTOS scheduler, multiple threads are executed simultaneously by switching between tasks. So, multitasking can be performed effectively (Espressif, 2021m).

ESP8266 and ESP32 SoC microcontrollers can be programmed in various programming languages: Arduino, MicroPython, LUA and JavaScript.

Arduino Programming

Both ESP8266 and ESP32-based boards can be programmed with the Arduino IDE based on C++ programming language. The 'Getting Started Guide' can be seen from the below link:

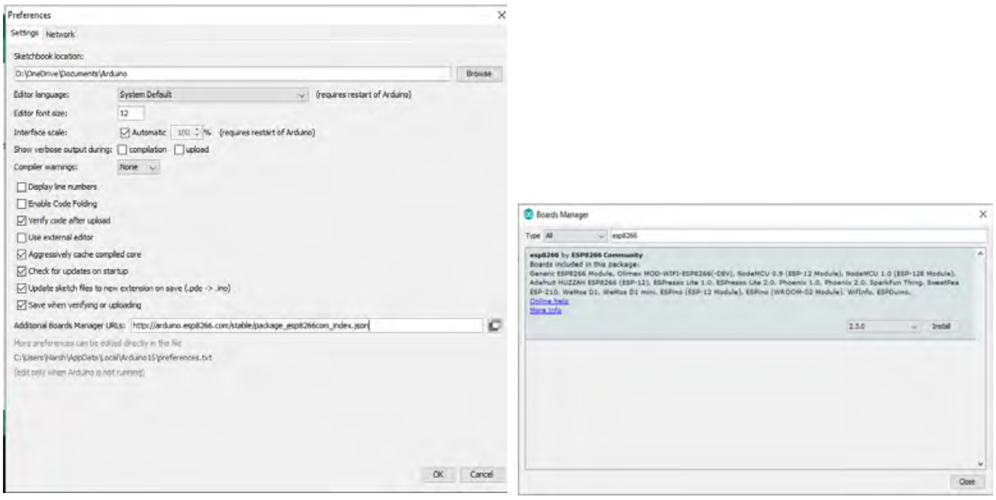
https://github.com/espressif/arduino-esp32

In order to code by using Arduino IDE, firstly the NodeMCU should be installed in supported boards. For this, the following links are added to File > Preferences window of Arduino IDE:

https://dl.espressif.com/dl/package_esp32_index.json

http://arduino.esp8266.com/stable/package_esp8266com_index.json

After that, NodeMCU boards are updated using Tools > Board > Boards Manager window. When clicked the Install button, Arduino IDE downloads all the required libraries for development boards. So, ESP32 boards are listed in the Tools > Board menu. Figure 15 shows the installing procedure of the ESP8266 package into Arduino IDE.



Sketchbook location: D:\OneDrive\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file
C:\Users\Harsh\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

Type All esp8266

esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.
Online help
More info

2.3.0

Figure 15. Installing the ESP8266 Package into Arduino IDE.

MicroPython Programming

MicroPython is a programming language written in C language, which is optimized to run on popular microcontrollers. As a lightweight implementation of Python 3, it includes a small subset of the Python standard library (Magda, 2017). MicroPython aims to simplify the programming of microcontrollers while simultaneously improving the performance and efficiency of code blocks. MicroPython runs on both the ESP8266 and ESP32 SoC microcontrollers. When programming the microcontrollers with MicroPython firmware, the programmer needs an IDE as an essential tool which is used for writing, testing, debugging and compiling program codes. Thonny is an open-source IDE which is used to write and upload the MicroPython codes to microcontroller. It can be downloaded from https://thonny.org link. The latest and stable versions of the firmware for ESP8266 and ESP32-based boards can be downloaded from following links:

https://micropython.org/download/esp8266

https://micropython.org/download/esp32

NodeMCU Firmware with LUA Programming

LUA is an open source scripting language which has many application programming interface (API) for network programming. LUA includes a stand-alone interpreter and works embedded in a host application within the application. The further details can be seen from the below link:

http://www.lua.org

NodeMCU is an LUA based firmware for the Espressif SoC microcontrollers. It can be downloadable from GitHub repository:

https://github.com/nodemcu/nodemcu-firmware

NodeMCU supports many Espressif modules, but only the required module is selected due to resource constraints. Also, the automated custom firmware build service can be used to get the specific firmware configuration. This service can be accessible via the below link:

https://nodemcu-build.com

ESPlorer is a Lua based IDE used to develop applications for NodeMCU. It allows to establish serial communication with Espressif modules, send commands and upload code. Note that, ESPlorer requires a Java Runtime Environment (JRE) to run. The latest ESPlorer IDE can be found in the following GitHub repository:

https://github.com/4refr0nt/ESPlorer

NodeMCU Flasher is a firmware programmer to program the Espressif modules. It enables the easily transferring the firmware files to the Espressif modules. The download link is given below:

https://github.com/nodemcu/nodemcu-flasher

The Arduino IDE uses the GPIO number to address each pin. However, GPIO pins renamed for NodeMCU firmware on LUA language. The GPIO pin mapping from NodeMCU to Arduino is shown by Table 3.

Table 3. NodeMCU to Arduino Pin Mapping.

NodeMCU	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
Arduino	16	5	4	0	2	14	12	13	5	3	1	9	10

JavaScript Programming

JavaScript is another language option to program the ESP8266 and ESP32-based boards. But, as in the use of MicroPython and LUA programming, JavaScript requires a firmware that defines the code blocks to microcontroller. Clearly, the firmware run like as an operating system of the computer. Espruino is a JavaScript interpreter which is designed to run on microcontrollers with constrained memory limits, i.e., 128 kB Flash and 8 kB RAM. The latest version of Espruino IDE (also Espruino Web IDE) can be found in following link:

https://www.espruino.com/

But, the setup of Espruino IDE depends on the node.js with NPM (Node Package Manager). Node.js is an open source and cross-platform runtime environment that executes JavaScript code blocks outside a web browser. Besides, NPM is the official package manager for Node.js environment. The details can be seen from https://nodejs.org link. Also, a detailed guideline can be read on https://www.espruino.com/programming web site.

Conclusion

This chapter has been discussed the technical specifications and programming abilities of ESP8266 and ESP32 series of SoC microcontrollers from Espressif Systems. Thanks to Wi-Fi and Bluetooth based wireless connectivity capabilities, these microcontrollers can connect the smart things to the Internet. Clearly, IoT projects can be easily prototyped with the help of ESP8266 and ESP32 SoC microcontrollers. This chapter has also presented an overview of the different available IDEs. Although there are so many programming environments, all of them have generally similar characteristics. Hence, embedded developers can adapt themselves to new environments. So they can rapidly evolve their suite of IoT devices and services.

References

Ai-Thinker. (2015). ESP-12E Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp12e_datasheet.pdf

Ai-Thinker. (2016). ESP-32S Datasheet. https://cdn.ozdisan.com/ETicaret_Dosya/632827_168142.pdf

Ai-Thinker. (2018a). ESP-01E Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-01e_product_specification_en.pdf

Ai-Thinker. (2018b). ESP-07S Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-07s_product_specification_en.pdf

Ai-Thinker. (2018c). ESP-12F Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf

Ai-Thinker. (2019a). ESP-12S Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-12s_product_specification_en.pdf

Ai-Thinker. (2019b). Nodemcu-32s Datasheet v1. https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf

Al-Fuqaha, A., Guizani, M., Mohammadi, M., & Aledhari, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347 - 2376. https://doi.org/10.1109/COMST.2015.2444095

Bakolia, A. (2019). Programming NodeMCU ESP8266 Over-the-Air (OTA) using Arduino IDE. https://circuitdigest.com/microcontroller-projects/esp8266-ota-update-programming-using-arduino-ide

Espressif Systems. (2020a). ESP8266 Technical Reference. https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

Espressif Systems. (2020b). ESP8266EX Datasheet v6.6. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Espressif Systems. (2020c). ESP-WROOM-02 Datasheet v3.2. https://www.espressif.com/sites/default/files/documentation/0c-esp-wroom-02_datasheet_en.pdf

Espressif Systems. (2020d). ESP-WROOM-S2 Datasheet v2.4. https://www.espressif.com/sites/default/files/documentation/esp-wroom-s2_datasheet__en.pdf

Espressif Systems. (2020e). ESP32 ECO V3 User Guide v1.1. https://www.espressif.com/sites/default/files/documentation/ESP32_ECO_V3_User_Guide__EN.pdf

Espressif Systems. (2020f). Downloading Guide. https://docs.espressif.com/projects/esp-at/en/latest/Get_Started/Downloading_guide.html

Espressif Systems. (2021a). SoCs. https://www.espressif.com/en/products/socs

Espressif Systems. (2021b). ESP8266. https://www.espressif.com/en/products/socs/esp8266

Espressif Systems. (2021c). ESP32 Series Datasheet v3.7. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Espressif Systems. (2021d). ESP32-S2 Family Datasheet v1.3. https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf

Espressif Systems. (2021e). ESP32-S3. https://www.espressif.com/en/products/socs/esp32-s3

Espressif Systems. (2021f). ESP32C3 Family Datasheet v1.0. https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf

Espressif Systems. (2021g). Announcing ESP32-C6: a Wi-Fi 6 + Bluetooth 5 (LE) SoC. https://www.espressif.com/en/news/ESP32_C6

Espressif Systems. (2021h). ESP32WROOM32 Datasheet v3.1. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

Espressif Systems. (2021i). ESP32WROVER Datasheet v2.4. https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf

Espressif Systems. (2021j). Official IoT Development Framework. https://www.espressif.com/en/products/sdks/esp-idf

Espressif Systems. (2021k). ESP-AT. https://www.espressif.com/en/products/sdks/esp-at/overview

Espressif Systems. (2021m). FreeRTOS. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html

Kolban, N. (2016). Kolban's book on the ESP8266. https://leanpub.com/ESP8266_ESP32

Kolban, N. (2018). Kolban's book on ESP32. https://leanpub.com/kolban-ESP32

Kurniawan, A. (2019). Internet of Things Projects with ESP32. Packt Publishing.

Magda, Y. (2017). Programming ESP8266-based Wireless Systems in MicroPython.

Stör, M. (2015). Comparison of ESP8266 NodeMCU development boards. https://frightanic.com/iot/comparison-of-esp8266-nodemcu-development-boards

About the Authors

Hakkı SOY was born in Konya, Turkey in 1978. He received the B.S. degree in Electronics Engineering from Uludağ University, Bursa, Turkey in 1999 and the Ph.D. degree in Electrical and Electronics Engineering from Selçuk University, Konya, Turkey in 2013. He joined the Department of Electrical and Electronics Engineering, Necmettin Erbakan University, Konya, Turkey, where he is currently an Assistant Professor. His research interests include wireless networks, embedded systems, smart sensors, cyber-physical systems, mobile communication and IoT applications.

E-mail: hakkisoy@erbakan.edu.tr, Orcid: 0000-0003-3938-038

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 14%.

To Cite This Chapter:

Soy, H. (2021). ESP8266 and ESP32 Series of SoC Microcontrollers. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 110 –125). ISRES Publishing.

Pic Microcontroller

Sabri KOCER
Necmettin Erbakan University

Resul BUTUNER
Konya Adil Karaagac Vocational and Technical Anatolian High School

Introduction

In simple applications, non-programmable solutions that do not require software development are preferable. Writing software is time consuming so it is more expensive and in simple and / or short run applications it is often more reasonable to perform tasks in hardware. However, as the complexity of the system increases, the benefits of using programmable systems increase. One of the main advantages of programmable systems is their flexibility, allowing you to update the operation of a system just by changing the program without having to redesign the hardware. This flexibility is very important, allowing products to be updated easily and economically.

The impressive effort of Microchip dedicated to updating its products has led to a massive commercialization of new models that have raised it to first place in the ranking 8-bit microcontroller world since 2002.

It is now possible to program microchips. Thanks to this technology, complex circuits can be made smaller by the use of these micro-controllers, of which pic is an excellent example. However, cheaper and smaller circuits would be possible using the PIC microcontroller. However, even complex logic circuits can benefit greatly from the use of PIC microcontrollers. In addition, by making changes to a PIC program, it will be easier to create prototypes without changing circuit designs and electronic components.

What has made the PIC MCU successful is:

- The availability of excellent, low-cost (free) development tools;
- The largest and strongest user Internet based community of probably any silicon chip family;
- An outstanding distributor network with a wide variety of parts available in very short notice;
- A wide range of devices with various features that just about guarantees that there is a PIC microcontroller suitable for any application;
- Microchip's efforts at continually improving and enhancing the PIC MCU family

based on customer’s needs.

This new updated and expanded edition includes, in addition to the PIC16F87X mid-range microcontrollers, the improved PIC18F range that are having a great acceptance and contains an introduction to modern 16-bit microcontrollers of the PIC24F and PIC24H families, as well as Digital Signal Controllers (DSC) embodied in the dsPIC families.

Importance of PIC Microcontrollers

Microchip, manufacturer of PIC microcontrollers, has remained a leader since 2002 8-bit microcontroller sales worldwide. MCU sales and units shipments driven by the spread of embedded control in systems, more sensors, and the rush to connect end-use applications to the Internet of Things (IoT).

PIC is a family of microcontrollers manufactured by Microchip Technology, derived from the PIC1650 originally developed by the Microelectronics Division of General Instrument. The name PIC initially referred to Peripheral Interface Controller, and is currently being expanded as Programmable Intelligent Computer. The first parts of the family were available in 1976; By 2013, the company had shipped more than twelve billion individual parts, used in a wide variety of embedded systems.

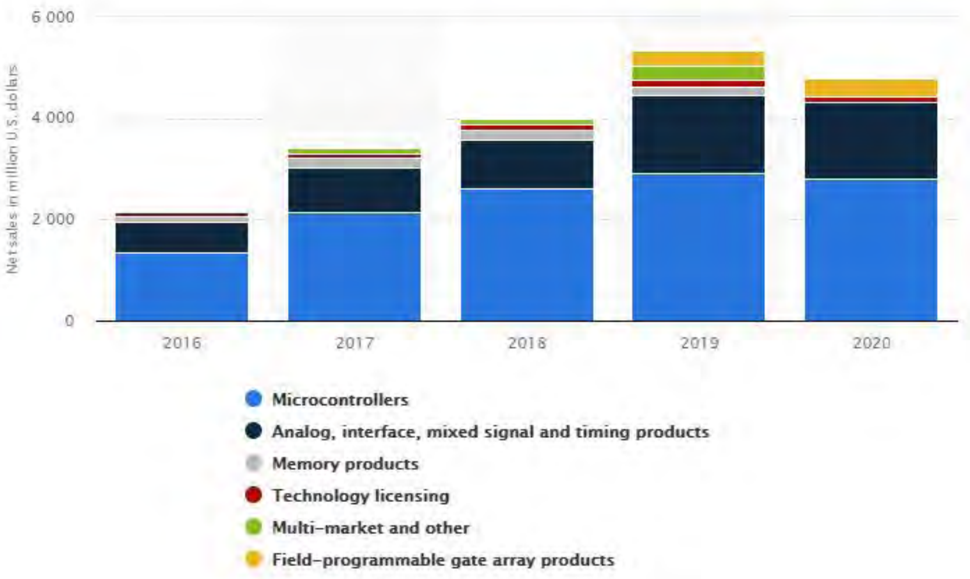


Figure 1. Microchip Technology Net Sales 2016-2020, Based on Product Line Published by Thomas Alsop, July 1, 2020

Figure 1 shows Microchip Technology’s net sales from 2016 to 2020, separated by product line. In 2020, Microchip Technology’s net microcontroller sales totaled approximately US $ 2.82 billion. Sales of Microchip products are absorbed by 43% by Asia and Japan, while that Europe supports 28% and America 29%. Regarding the market segments to which the PIC microcontrollers are destined, it stands out the generic area of Consumer Electronics with 35%, followed by the automotive industry with 18%.

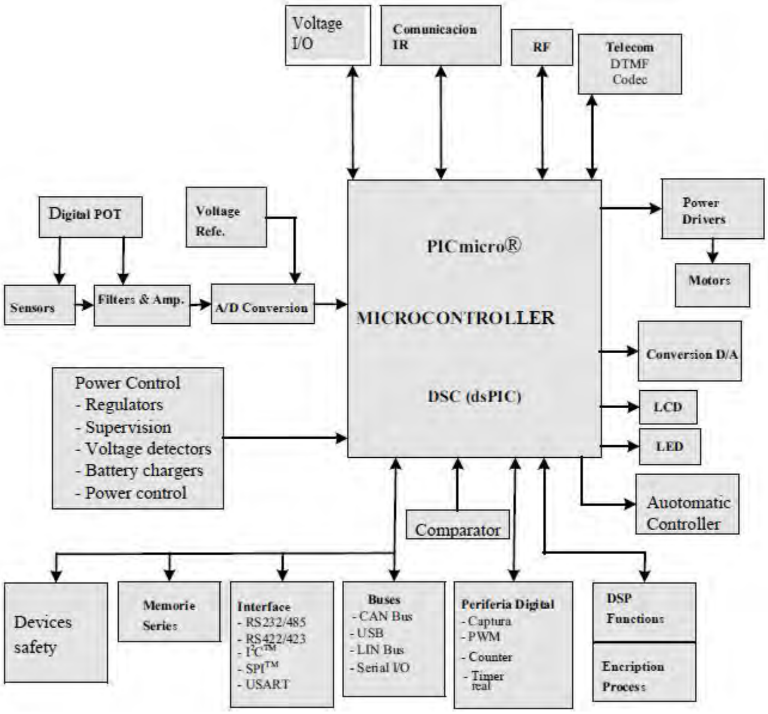


Figure 2. PIC Microcontroller Application Area

The great diversity of microcontroller models allows the designer to find the one that contains all the resources and memory capacities that he needs for his application (Figure 2.).

The technology used in the manufacture of PIC devices has gone from 0.7 microns in 1998 to 0.22 microns in 2006. This has meant a reduction in the supply voltage, which has gone from being between 2 and 5 , 5 V, to operate with 5V, at a range between 2 and 3.6 V, being the nominal of 3.3 V although the I / O continues to work with 5V.

One of the great advantages of Microchip microcontrollers is their “migrability”, which means the possibility of changing the MCU model and moving to a more powerful one with more memory and peripheral capacity, maintaining the distribution of the pins in the package or Pinout, software and instruction set compatibility, peripheral compatibility, and development tool compatibility.

Classification of PIC Microcontrollers

These microcontrollers or MCUs are characterized by their Harvard architecture with independent program and data memories, which allows simultaneous accessibility and diversity in the length of the positions and the size of both memories.

Microcontrollers are generically classified according to the size of the data handled by the instruction set and there are 4 large groups: 4, 8, 16 and 32 bits. Microchip only manufactures 8-bit and 16-bit microcontrollers and as noted is the world leader in first group sales.

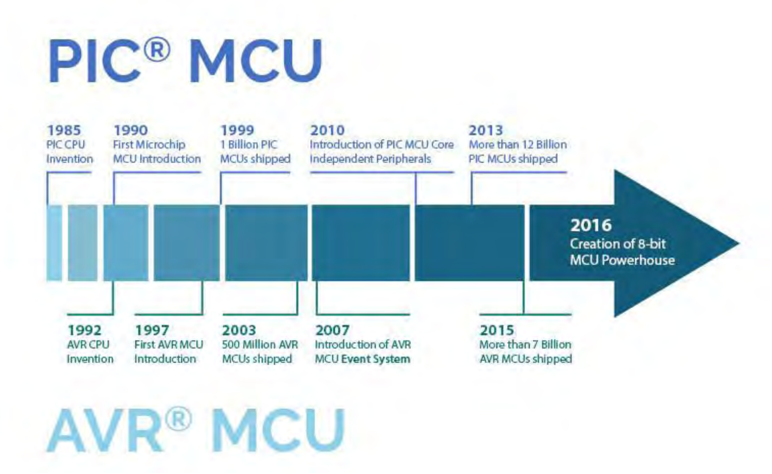


Figure 3. Historical Milestones in 8-bit AVR and PIC MCU Development

The AVR and PIC microcontroller brands, embedded design also represents the most effective architecture. Figure 3 shows historical milestones in 8-bit AVR and PIC MCU development. With 45 years of combined experience in developing commercially available and cost-effective 8-bit MCUs, Microchip is the supplier of choice for many due to its strong legacy and history of innovation in 8-bit.

Most current microprocessors are standard Von Neumann machines. The biggest deviation from this is the Harvard architecture, where instructions and data have separate address, data, and control buses for each memory area, and different memory areas in Figure 4. This directive and data returns can occur at the same time and has the advantages of not having the size of an instruction set relative to the standard data unit (word) size.

The PIC is based on the Harvard architecture in which the program and data buses are kept separate. Early versions of PIC microcontrollers use EEPROM to store program instructions, but have adopted flash memory since 2002 to enable better erasing and storage of code.

Thanks to their simplicity in architecture and ease of use, PIC microcontrollers have proven to be a hit among hobbyists, students and professionals. PIC16F84 and PIC16F877 were some of the most popular PIC microcontrollers with basic functions. Applications that require richer peripherals, higher performance or memory can rely on the PIC18F family.

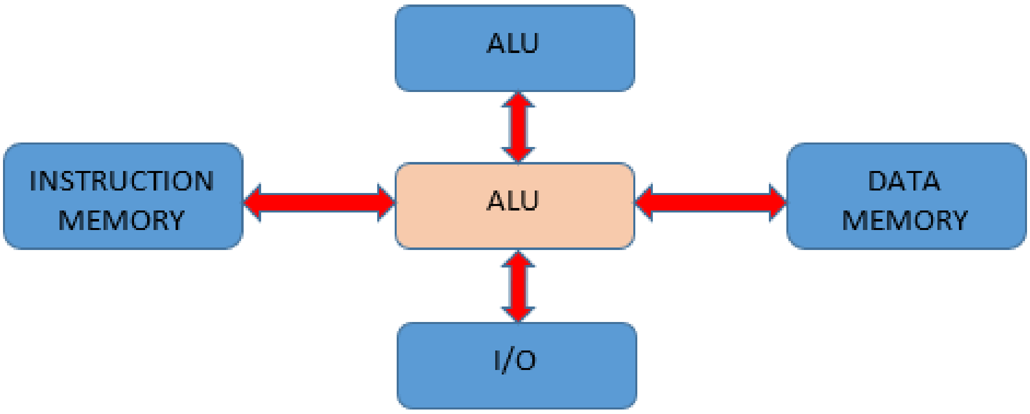


Figure 4. The Harvard Architecture used by PIC Microcontrollers

Instruction bus: Program instructions are fed from the FLASH program memory to the ALU via the 14-bit instruction bus. A 14-bit program word is read to the ALU in each instruction clock cycle

Bus: An 8-bit bus connects the ALU to the data memory area. During each instruction, the ALU can read data from the data memory location, modify the data, then write the data back to the memory.

Instruction Pipeline: The advanced mid-range PIC's dual-bus architecture provides a two-stage instruction pipeline. One each clock cycle execute two instruction phases:

The next instruction is "taken" from Program memory

The current instruction is "executed" and reads/modifies/writes the data memory (if needed)

8-Bit PIC Microcontrollers

They are distinguished by the length of the native data handled by the instructions is 8 bits, which corresponds to the size of the data bus and that of the CPU registers.

8-bit PIC microcontrollers are classified into three broad ranges: Base, Medium and Advanced, with a total of about 300 different models that contain different capacities of memory, peripherals and different types of packages (Figure 5.)

8-bit PIC® microcontrollers come in several core architectures. This can be confusing for someone trying to decide the best option for their product or project. This summary will describe the different options to help you make the right decision.

Microchip offers different ranges of microcontrollers.

- PIC12CXXX base line PIC (8-pin, 12-bit / 14-bit program word):

Low consumption, EEPROM data memory.

- PIC16C5X, low-end or classic (12-bit program word):

Encapsulated 14, 18, 20 and 28 pins, Optimal for applications that work with batteries (low consumption).

- PIC16CXXX, midrange (14-bit program word).

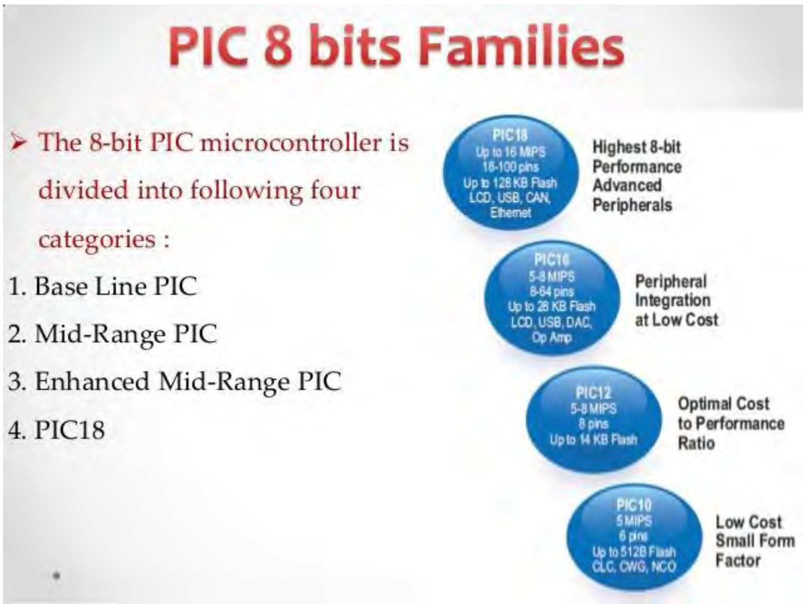
A / D converters and serial port, Encapsulated from 18 to 68 pins.

- PIC17CXXX, Enhanced mid range (16 bit program word).

Open architecture, expandable memory,

- PIC18XXX, high range (16 bit program word).

ny6



PIC 8 bits Families

➢ The 8-bit PIC microcontroller is divided into following four categories :

1. Base Line PIC
2. Mid-Range PIC
3. Enhanced Mid-Range PIC
4. PIC18

PIC18
Up to 16 MIPS
16-100 pins
Up to 128 KB Flash
LCD, USB, CAN, Ethernet

Highest 8-bit Performance Advanced Peripherals

PIC16
5-8 MIPS
8-64 pins
Up to 28 KB Flash
LCD, USB, DAC, Op Amp

Peripheral Integration at Low Cost

PIC12
5-8 MIPS
8 pins
Up to 14 KB Flash

Optimal Cost to Performance Ratio

PIC10
5 MIPS
6 pins
Up to 512B Flash
CLC, CWG, NCO

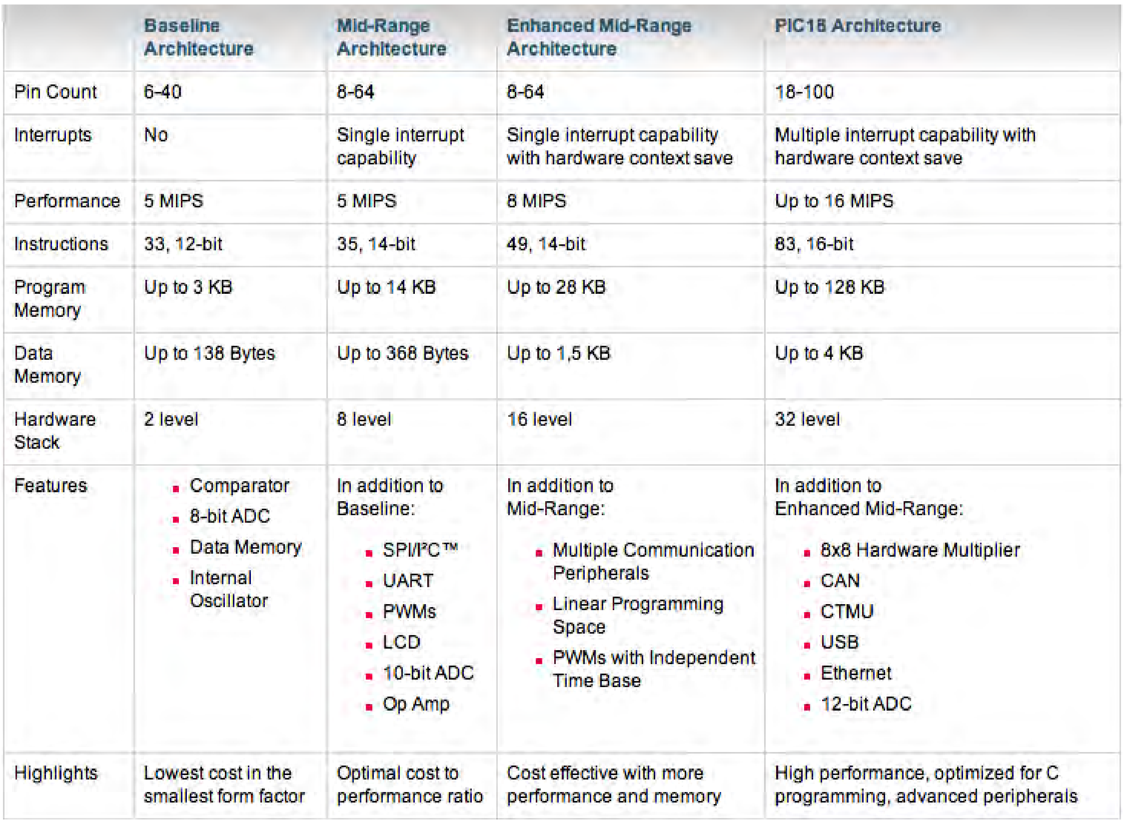
Low Cost Small Form Factor

Figure 5. 8-bit PIC Microcontrollers are Classified: Base, Mid-range and Advanced and PIC18

There are currently three main ranges of PIC MCUs: The basic ones (Linebase), the mid-range ones (Mid Range) and the high performance ones (high performance). The PIC18s are considered high performance and have among their members PICs with communication modules and advanced protocols (USB, Ethernet, Zigbee for example).

The dsPICs are Microchip's penultimate release, starting large-scale production in late 2004. They are the first 16-bit inherent data bus PICs. They incorporate all the possibilities of the previous PICs and add several DSP operations implemented in hardware, such as multiplication with accumulator addition (multiply-accumulate, or MAC), barrel shifting, bit reversion or 16x16-bit multiplication.

Microchip Technology launched in November 2007 the new 32-bit microcontrollers with a processing speed of up to 1.6 DMIPS / MHz with USB HOST capability. Its clock frequencies can reach 80MHz from standard 4 to 5MHz quartz thanks to an internal PLL. They operate at 3.3V in their input and output ports, although the manufacturer indicates that except for the pins with analogue function, most voltages of up to 5V are tolerated. They have an optimized architecture with a high degree of parallelism and an M4K-type core and a high capacity of RAM and FLASH ROM memory. All this means that these MCUs allow high information processing.



	Baseline Architecture	Mid-Range Architecture	Enhanced Mid-Range Architecture	PIC18 Architecture
Pin Count	6-40	8-64	8-64	18-100
Interrupts	No	Single interrupt capability	Single interrupt capability with hardware context save	Multiple interrupt capability with hardware context save
Performance	5 MIPS	5 MIPS	8 MIPS	Up to 16 MIPS
Instructions	33, 12-bit	35, 14-bit	49, 14-bit	83, 16-bit
Program Memory	Up to 3 KB	Up to 14 KB	Up to 28 KB	Up to 128 KB
Data Memory	Up to 138 Bytes	Up to 368 Bytes	Up to 1,5 KB	Up to 4 KB
Hardware Stack	2 level	8 level	16 level	32 level
Features	▪ Comparator ▪ 8-bit ADC ▪ Data Memory ▪ Internal Oscillator	In addition to Baseline: ▪ SPI/I²C™ ▪ UART ▪ PWMs ▪ LCD ▪ 10-bit ADC ▪ Op Amp	In addition to Mid-Range: ▪ Multiple Communication Peripherals ▪ Linear Programming Space ▪ PWMs with Independent Time Base	In addition to Enhanced Mid-Range: ▪ 8x8 Hardware Multiplier ▪ CAN ▪ CTMU ▪ USB ▪ Ethernet ▪ 12-bit ADC
Highlights	Lowest cost in the smallest form factor	Optimal cost to performance ratio	Cost effective with more performance and memory	High performance, optimized for C programming, advanced peripherals

Figure 6. The Differences Among the 8-bit PIC® MCU Sub-Families

The Base Line Range

The low cost and small size Baseline family contain PIC 10, sum PIC 10, sum PIC 12 devices. Baseline microcontrollers use a 12-bit instruction word and provide the right amount of features and options to minimize expense and get the job done right. The baseline has the simplest architecture of the 8-bit family and is therefore the easiest to use and understand. (Figure 6)

To favor hardware migrability and allow the transition to more powerful PIC models with more legs, the distribution of the functions assigned to the legs is maintained, so that connection and track layout changes are minimal. Figure 7 is a table of peripherals available on the latest 8-bit PIC® MCU devices, followed by an alphabetical list of links to the pages that describe these peripherals in detail.

The increase in the number of pins means the increase in memory capacities and in the number of peripherals and resources integrated in the device.

					Peripheral Function Focus																																							
					Intelligent Analog								Waveform Control							Timing and Measurements[illegible]							Logic and Math			Safety and Monitoring			[illegible]					User Interface			Low Power and System Flexibility			
Product Family	Pin Count	Program Flash Memory (KB)	RAM (B)	Data EE (B)	ADC (# of bits)	Comp	HSComp	DAC (# of bits)	OPA	PRG	SlopeComp	ZCD	CCP/ECCP	10-bit PWM	16-bit PWM	COG	CWG	NCO	DSM	AngTMR	HLT (8-bit)	16-bit PWM (16-bit)	NCO (20-bit)	SMT (24-bit)	RTCC	TEMP/TS	CLC	MULT	MathACC	CRC/SCAN	HLT	WWDT	EUSART/AUSART	UART with Protocols	I²C/SPI	USB with ACT	LIN Capable	mTouch® Sensing	HCVD	LCD	PPS	IDLE/DOZE/PMD	DMA/VI	DIA/MAP
PIC10(L)F3XX	6	384–896	64	HEF	8									✓			✓	✓					✓			✓	✓											✓						
PIC12LF1552	8	3.5	256	HEF	10																					✓									✓			✓	✓					
PIC16LF155X/6X	14–20	7–14	1024	HEF	10[illegible]									✓												✓							✓		2		✓	✓	✓					
PIC16(L)F145X	14–20	14	1024	HEF	10	✓								✓			✓									✓							✓		✓	✓	✓	✓						
PIC1X(L)F157X	8–20	1.75–14	1024	HEF	10	✓		5							✓		✓					✓				✓							✓				✓	✓			✓			
PIC16(L)F153XX	8–48	3.5–28	2048	HEF	10	✓		5				✓	✓	4			✓	✓			✓		✓			✓	✓				✓	✓	2		2		✓	✓			✓	✓		✓
PIC1X(HV)F752/53	8–14	1.75–3.5	128	–	10		✓	5/9	✓		✓		✓			✓					✓										✓							✓						
PIC1X(L)F1612/3	8–14	3.5	256	HEF	10	✓		8				✓	✓				✓				✓			✓		✓				✓	✓	✓						✓						
PIC16(L)F161X	14–20	7–14	1024	HEF	10	✓		8				✓	✓	✓			✓			✓	✓			✓		✓	✓		✓	✓	✓	✓	✓		✓		✓	✓			✓			
PIC16(L)F170X	14–20	3.5–14	1024	HEF	10		✓	5/8	✓			✓	✓	✓		✓										✓	✓						✓		✓		✓	✓	✓		✓			
PIC16(L)F171X	28–40	7–28	2048	HEF	10		✓	5/8	✓			✓	✓	✓		✓		✓					✓			✓	✓						✓		✓		✓	✓	✓		✓			
PIC16(L)F176X/7X	14–40	7–28	2048	HEF	10		✓	5/10	✓	✓	✓	✓	✓	✓	✓	✓			✓		✓	✓				✓	✓				✓		✓		✓		✓	✓			✓			
PIC16(L)F183XX	8–20	3.5–14	2048	256	10	✓		5					✓	✓			✓	✓	✓				✓			✓	✓						✓		2		✓	✓			✓	✓		
PIC16(L)F184XX	14–28	7–28	2048	256	12[illegible]	✓		5				✓	✓	✓			✓	✓			✓		✓	✓		✓	✓					✓	✓		2			✓	✓		✓	✓		✓
PIC16(L)F188XX	28–40	7–56	4096	256	10[illegible]	✓		5				✓	✓	✓			✓	✓	✓		✓		✓	✓		✓	✓			✓	✓	✓	✓		2		✓	✓	✓		✓	✓		
PIC16(L)F191XX	28–64	14–56	4096	256	12[illegible]	✓		5				✓	✓	✓			✓				✓			✓	✓	✓	✓				✓	✓	✓		✓		✓	✓	✓	✓[illegible]	✓	✓		✓
PIC18(L)FXXK40	28–64	16–128	3728	256–1K	10[illegible]	✓		5				✓	✓	✓			✓		✓		✓					✓		✓		✓	✓	✓	5		2		✓	✓	✓		✓	✓		
PIC18(L)FXXK42	28–48	16–128	8192	256–1K	12[illegible]	✓		5				✓	✓	4			✓	✓	✓		✓		✓	✓		✓	✓	✓		✓	✓	✓	1	1	2		✓	✓	✓		✓	✓	✓	✓
PIC18(L)FXXJ94	64–100	32–128	4096	–	12	✓							✓												✓			✓					4		2	✓	✓	✓		✓[illegible]	✓	✓		
PIC18(L)FXXK83	28	32–64	4096	1K	12[illegible]	✓		5				✓	✓	✓			✓	✓	✓		✓		✓	✓		✓	✓	✓		✓	✓	✓		2	✓		✓	✓	✓		✓	✓	✓	✓
PIC18FXXQ10	28–40	128	3728	1024	10[illegible]	✓		5				✓	✓	✓			✓		✓		✓					✓	✓			✓	✓	✓	2		2			✓	✓		✓			

Figure 7. 8-bit Peripheral Summary

The Mid Range

The microcontrollers that make up this range respond to a repertoire of 35 instructions with a format of 14 bits in length each, the Stack being 8 levels deep and having an interrupt vector. It is a large range that currently reaches 71 different models, starting with those that are encapsulated with 8 legs and reaching those with 68 legs. Figure 8 shows the most relevant aspects of some PICs of the 8-leg Medium range and others with more.

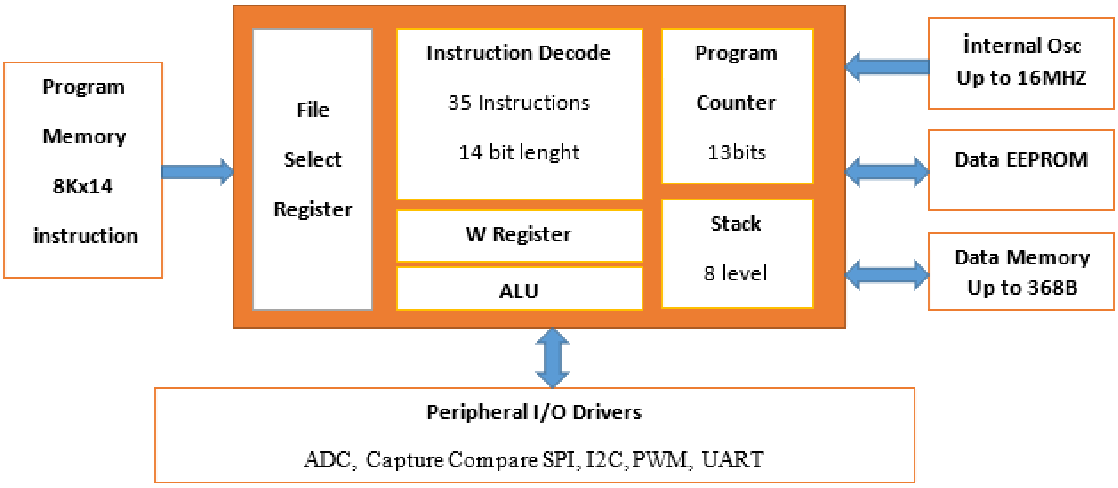


Figure 8. Mid Range Core

Enhanced Mid-Range

To improve the performance of two PIC Mid-Range microcontrollers, the PIC Enhaced Mid-Range family was raised. This family increased its performance by more than 50% compared to mid-range microcontrollers. Its core has a multiplier for Hardware and optimized for programming in C. Or code created by compiler C is reduced by 40% compared to the Mid-Range family in Figure 9. It has a lower cost than the PIC18 microcontrollers.

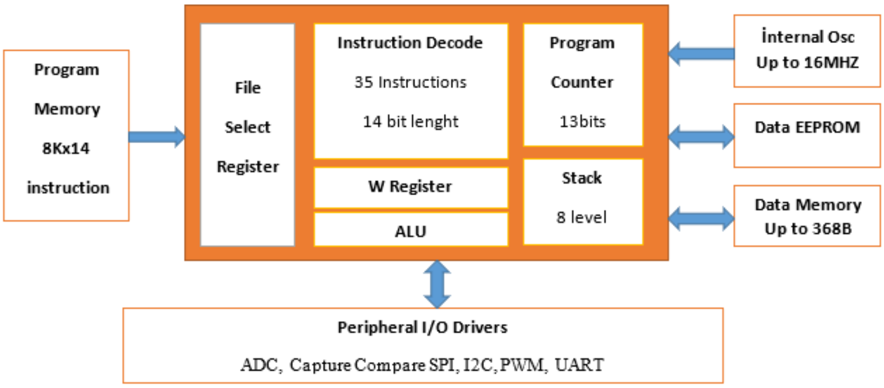


Figure 9. Enhanced Mid-Range Core

Advanced feature set, multiple serial communication and motor control capability

High-End

To the PIC18 family it presents a higher level of performance and integration of two 8-bit microcontrollers from Microchip. Possuem 16-bit instructions with more than 16 MIPS processing, 32-level hardware pile, 8 × 8 multiplier for Hardware, or that makes linguagem the best option for programming the architecture.

These microcontrollers present various advanced peripherals such as: CAN, USB, Ethernet, LCD among others.

To follow are presented its main characteristics:

- 83 instructions with optimization for programming in C language;
- I attached 2 MB program memory;
- 4KB of RAM;
- 32 snow hardware pilot;
- 8 × 8 multiplier by hardware;

The great interest of the manufacturer in this powerful model range can be felt, as it has the largest number of different devices. The capacity of program memory can reach 128 KB, data memory up to 3963 bytes and EEPROM up to 1 KB. It has highly specialized peripherals, among which a 10-bit AD converter stands out, up to 5 timers, interfaces for communication with I2C bus, SPI, USART, can 2.0 b, etc. Among the important resources is a fast hardware multiplier that allows this operation to be performed in one instruction cycle.

PIC 16F84 Architecture

Since simple applications require few resources and more complex applications require numerous and powerful resources, Microchip builds various models of microcontrollers geared to meet the needs of each project.

The architecture and programming of all PICs is very similar. In this topic we will focus on the PIC16F84 microcontroller, which is one of the mid-range microcontrollers. Its main characteristics are the following:

- Part of the data memory is EEPROM type (64 8-bit registers).
- Flash-type program memory (1024 registers of 14 bits), with the same performance as EEPROM but better performance.
- Two timers: TMR0 and watchdog. The TMR0 can act as a timer or a counter.
- Four possible interrupt sources that can be enabled or disabled by software.
- System reinitialization or RESET for five different causes.
- State of operation in low consumption or Sleep, with a consumption of 40 mA.
- Maximum working frequency can be 10 MHz.

Block Diagram

A somewhat simplified internal architecture (Fig. 10) has been derived from the block diagram given in the data sheet.This architecture applied by Microchip in its microcontrollers is characterized by the independence between code and data memory. Thus, both the capacity and the size of the buses of each memory are strictly adapted to the needs of the design, facilitating the parallel work of the two memories, which allows obtaining high levels of performance. The RISC philosophy is evident in the small number of instructions that make up its repertoire. It only consists of 35 instructions, which are executed in an instruction cycle, equivalent to four clock periods, except for jump instructions that require two cycles.

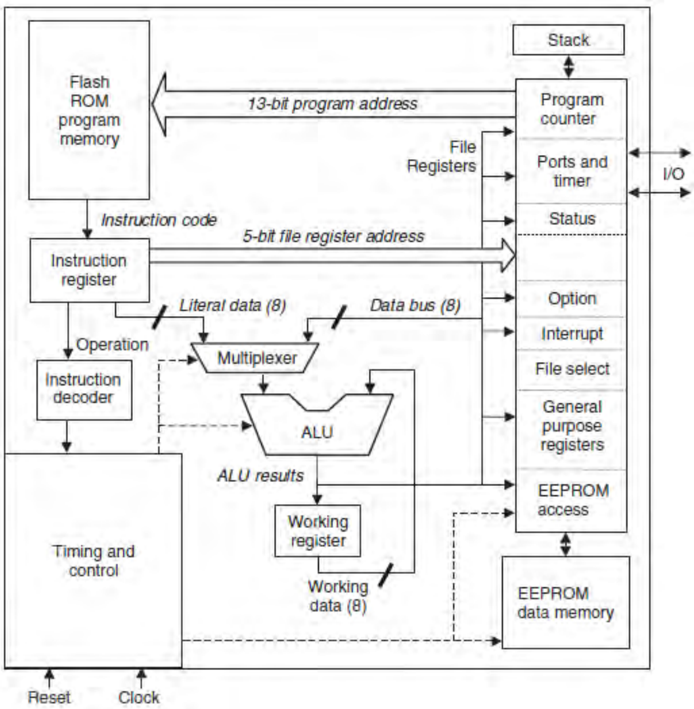


Figure 10. PIC 16F84 Internal Architecture.

The file recordset contains the port registers and the program counter, as well as various audit and status registers. The most commonly used ports (A and B), status register (status), real time clock counter (TMR0) and interrupt control (INTCON). There are also a number of redundant General Purpose registers (GPRs) that can be used as data registers, counters, etc. File records are numbered 00–4f, but appropriate tags are usually given in the program source code. File registers also provide access to an EEPROM block, which is a non-volatile data memory.

Thanks to its ports, the microcontroller can respond to external events and perform a certain action, such as varying the output signals according to the value of the inputs. To respond to external events, the microcontroller has a resource known as interrupts. Interruptions are random events, which are generated inside or outside the microcontroller, they stop the execution of the main program, to attend a secondary program or subprogram.

A real application is for example a push button connected to an input pin. Once pressed, an interrupt signal is generated, which will start the execution of the interrupt subroutine, which can be a timer, which at the end of the programmed time activates an output signal, this signal could turn off an oven for example or simply turn on a led.

To determine the input and output needs of the system, it is convenient to draw a block diagram of the system, in such a way that it is easy to identify the amount and type of signals to be controlled. Once this analysis has been carried out, it may be necessary to add peripherals, external hardware or change to another microcontroller more suitable for that system.

Ports A and B

The ports of the Microchip 16F84 microcontroller (Figure 11) are a set of pins or lines that are grouped together, each group of pins or lines makes up a complete port, the 16F84 has two ports implemented, port A and port B, both ports are bidirectional and They can be configured as input or output, in this way the ports allow the input or output of data to or from the internal part of the microcontroller, port A is made up of five pins or lines, ranging from RA0 to RA4, as It is observed in figure 1.5, pin 3 is multiplexed, that is, it fulfills a double function, which corresponds to RA4 / T0CKI, this is the only pin of port A that is multiplexed, and it can work as a RA4 port or as an input of the counter external events T0CKI, for the latter, pin 3 must be configured as an input.

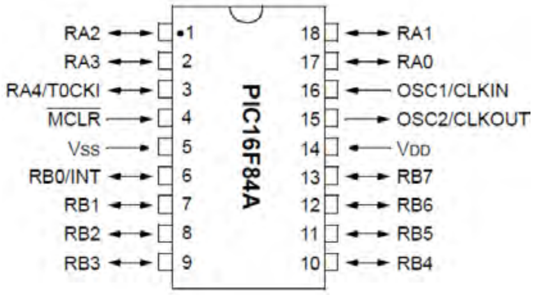


Figure 11. The Peripheral Diagram Shows in Great Detail the Function of each Pin.

Pin 3 fulfills double function, and you cannot use both functions at the same time, through the configuration you choose which function that pin will fulfill, and if it will be input or output, the other port with more pins is port B, and it is made up of eight pins or lines, which go from RB0 to RB7, as shown in figure 1.5, pin 6 is multiplexed, that is, it fulfills a double function, which corresponds to RB0 / INT, this is the only pin or line that is multiplexed, and can work as RB0 port or as external interrupt input INT, but never as port and external interrupt input at the same time, through the configuration you choose the function that the pin or line will fulfill and if it will be input or output.

The ports A and B of the microcontroller can be configured as input or output, according to the requirements of the application. If the port is configured as an output, the pins of the port work in source mode, supply current, and the maximum current that a line or pin is 20mA. If the port is configured as input, the pins work in sink mode, they receive current, and the maximum current that can enter through a line or pin is 25mA. If in the application there are loads that consume higher currents, such as: stepper motors, relays, various LEDs and other peripherals that require more current than indicated by the manufacturer, a matching coupler circuit must be applied. Pull-up resistors must be activated every time it is necessary, and not leave pins or input lines in a floating state, in digital electronics two logical states are handled. This problem is solved by activating an internal resistance between the input pin and Vdd, called a pull-up resistor, its objective is to ensure a logic 1 when the switch is open. If what is desired is to ensure a logical 0, the resistor is connected to Vss and is called a pull-down resistor.

Registers Sets

It is necessary to indicate that there are two internal registers associated with the ports, register with port should not be confused, just as there is a port A with 5 pins, there is an internal register called PORTA, port A with the PORTA register are closely related. These registers in some cases are called words or bytes, which means the same thing, in the registers the data reading or writing operations can be carried out, applying the MOV data movement instructions.

All the file registers are 8-bits wide. Figure 12 shows PIC 16F84 file register set. They are divided into two main blocks Special Function Registers (SFR), which are reserved for specific purposes, and General Function Registers (GPR) which can be used for temporary storage of any data byte.

Address	Page 0	Page 1	Address
0	IND0	▨	
1	**TMR0**	**OPTION**	81
2	**PCL**	▨	
3	**STATUS**	▨	
4	FSR	▨	
5	**PORTA**	**TRISA**	85
6	**PORTB**	**TRISB**	86
7	▨	▨	
8	EEDATA	EECON1	88
9	EEADR	EECON2	89
A	PCLATH	▨	
B	**INTCON**	▨	
C	**GPR1**	▨	
D	**GPR2**	▨	
E	**GPR3**	▨	
F	**GPR4**	▨	
10	**GPR5**	▨	
: : : : :	| GPRs |	▨	
4F	**GPR68**	▨	

Figure 12. PIC 16F84 File Register Set.

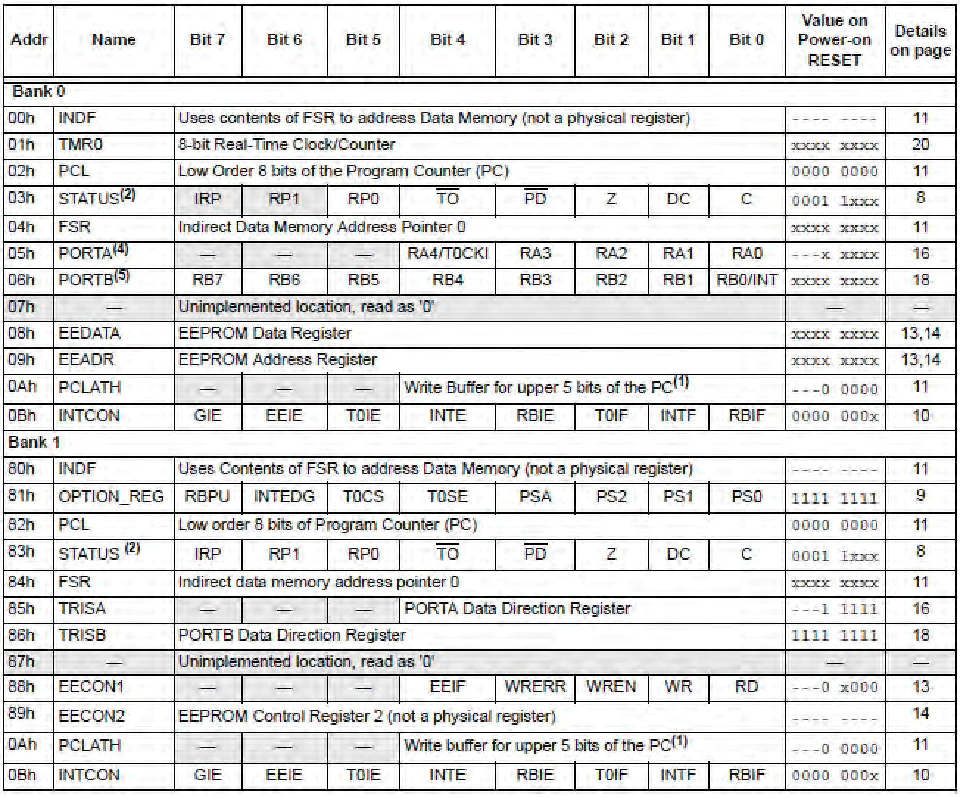
In the PIC there are many internal registers, these registers are contained in memory banks, a memory bank is nothing more than a set of registers ordered sequentially. Each register has a technical and unique name. There are microcontrollers that have two up to four memory banks, the 16F84 has two memory banks, each bank is assigned a name, bank 0 and bank 1.

In the memory banks there is a space of 68 bytes that the manufacturer left free, this space is known as RAM memory and is occupied by the program variables, this space starts at address 0x0C and ends at address 0x4F of bank 0 , according to the free space, up to 68 variables can be declared, which is a sufficient amount for application development, The procedure to address the memory banks will be explained in the following points. Figure 1.9 shows the graphic representation of the two memory banks of the PIC 16F84, which was copied from the manufacturer's manual.

Special Function Registers

The SFR Special Function Registers (Figure 12 and 13) are used by the CPU and peripheral functions to control the operation of the device. These registers are the static RAM.

The special function registers can be classified into two sets, central and peripheral. Those associated with the main functions are described in this section. Those related to the operation of peripheral features are described in the section for that specific feature.



Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
Bank 0											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS(2)	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA(4)	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB(5)	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10
Bank 1											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS (2)	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	16
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC(1)					---0 0000	11
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	10

Figure 13. Special Function Register File Summary

Interrupts on PIC 16F84:

Mid-range PICs are capable of handling 4 sources of interrupts:

- External (RB0 / INT pin).
- A change in bits RB4 to RB7 of Port B.
- Timer interruption (terminal count).

- Completion of writing to the EEPROM memory.
- There is a single interrupt vector at address 0004h.
- The INTCON register (position 0Bh, both banks) allows each interrupt to be enabled (equivalent to individual masks).
- This same register contains flags that serve to indicate (to the service subroutine) which was the origin of the interruption. INTCON also has a general enablement bit (GIE) that affects all sources.

Mid-Range Instruction Set

Special features of the assembler for PIC

- Uses the mnemonics of the instructions of the µC to which it is directed. In our case, it will correspond to the instruction set of the mid-range PICs (only 35 instructions).
- Allows the use of labels, both to point to addresses and to define constant values.
- It is possible to express numerical quantities in different bases, the default base being hexadecimal.
- There are a series of pseudo-instructions or directives (in addition to the well-known ORG, EQU and END) specific to these µCs.

Each PIC16CXX instruction is a 14-bit word divided into an opcode that indicates the type of instruction and one or more operands that further indicate the instruction's execution. The pic16cxx instruction set summary in Figure 13 lists byte-oriented, bit-oriented, and literal and control operations. Figure 14 shows the opcode field descriptions.

To better understand each instruction, the meaning of some parameters will be explained below:

f: Register affected by the instruction

w: Accumulator (Working register)

b: Bit number (there are instructions that affect a single bit)

k: Constant (a number)

d: Selection of the destination of the result of the instruction, it can be “0” or “1”, if it is “0” the result is saved in the accumulator (w) and if it is “1” it is saved in the register f to which affects instruction.

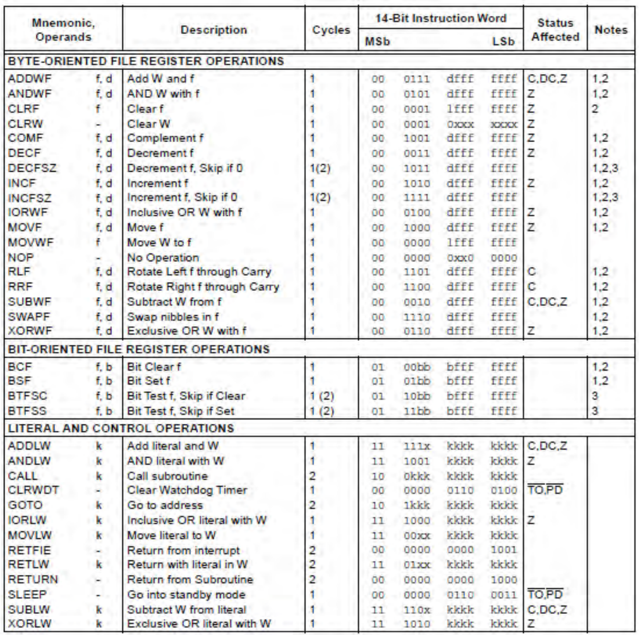
Data Transfer Instructions

Data transfer within the microcontroller takes place between the working register and a register that represents anywhere in the internal RAM, whether it relates to special function or general purpose registers.

Move the first three instruction literals to the W register, move the data from the W register to RAM and from RAM to the W register. The CLRF instruction clears the f register, while the CLRW clears the W register. The swap instruction changes nibbles within the F register.

Arithmetic-Logic Instructions

Similar to most microcontrollers, the PIC only supports two arithmetic instructions - addition and subtraction. Flags C, DC, z are automatically set depending on the results of addition or subtraction. The only exception is the C flag. The C flag is inverted after it is subtracted, since the subtraction is performed by an addition with a negative value. This means that the C flag is set if it is possible to perform the operation and cleared if the larger number is subtracted from the smaller one. The logic can perform one (1) operation of the PIC and or, EX-or, inversion (COMF) and rotation (RLF and RRF). Instructions that return a register actually rotate their bits one bit left (toward bit 7) or right (toward bit 0) along the C flag. The bit shifted from the register is automatically moved to the C flag which is moved to the bit on the opposite side of the register.



Mnemonic, Operands		Description	Cycles	14-Bit Instruction Word				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Figure 14. Midrange Instruction Set

Bit-oriented Instructions

BCF and BSF instructions clear or set any bits in memory. Although it may seem like a simple process, it is not. The CPU first reads the entire byte, changes a bit, and rewrites the entire byte to the same location.

Typical fixed connection for any application.

In the circuits where the PIC16F84 is used, it is usual to use a supply voltage of + 5V and as an external clock circuit one of the XT type at a frequency of 4MHz. With this configuration, the fixed wiring for any application is shown in Figure 15. The pins that are not connected are those dedicated to transferring the information with the peripherals used by the application.

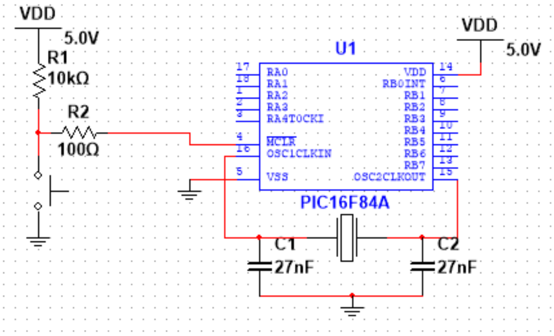


Figure 15. Typical Fixed Connection for any Application.

Actual designs use various peripherals that must be connected to the microcontroller pins that support the I / O lines.

Pushbuttons.

These devices allow a logic level to be entered at the moment they are activated, passing to the opposite level when they are stopped (they return to the rest position).

In the diagram on the left of Figure 16 the input line receives a high logic level when the pushbutton is idle and a low logic level when it is actuated. The button on the right works in reverse.

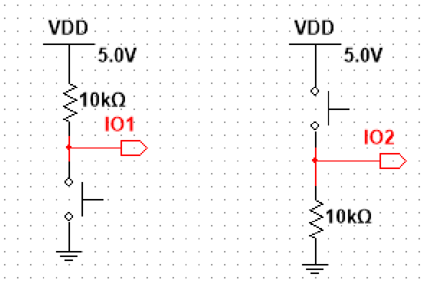


Figure 16. Two Possible Ways to Connect a Pushbutton.

There are a multitude of detectors, limit switches and digital sensors that work in the same way as push buttons.

LEDs.

The led diode is an element that is used as a light indicator. When the potential difference between its anode and its cathode exceeds a certain threshold value, the led diode will turn on. The PIC lines can supply enough current to turn on an LED diode, so they can be connected directly through a resistor as shown in Figure 17. If we use the connection on the left of the figure, the led diode will light when the microcontroller output is set to ‘1’, while with the connection on the right it will do so when the output is set to ‘0’.

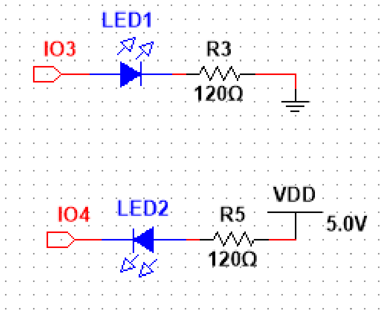


Figure 17. LED Connections

Diodes or other loads sometimes require more current than the PIC lines can deliver. In this case it is necessary to insert an amplifier stage.

Relays

Activating and deactivating a relay provides the opportunity to control much larger loads (more current) because they can be controlled by the relay contacts (Figure 18).

When the output line, OUT, applies a high level to the base of the Darlington transistor (amplifier stage) causing it to conduct and activate the relay. Closing the relay contacts controls a higher load. The value of the resistance depends on the type of relay and the transistor.

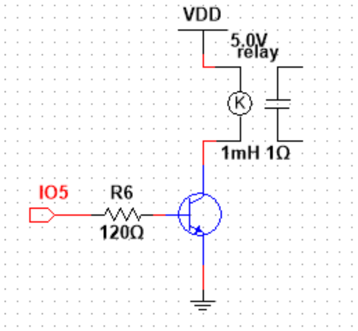


Figure 18. Scheme of the Control of a Relay.

General Structure of an Assembler Program.

In a program written in assembly language, in addition to the 35 instructions interpreted by the processor, directives are also placed, which are commands to perform certain operations with the program. The parts that generally exist in a program are discussed below:

1. Processor model and numbering system.

Programs start with the list directive that references the microcontroller model. The type of numbering to be used with the radix directive is also usually specified. The usual thing is to use the hexadecimal system, in which the values are expressed preceded by “0x”.

List p = 16F84; PIC16F84 microcontroller is used

2. Variables

Data memory locations are used to store operands and results, as well as to store special registers.

To make it easier for the programmer to make the program, instead of referring to the memory locations where the data to be used is located, a name is associated with each of these locations. The equ directive relates a name to the address that is assigned, so the programmer works with names and the compiler automatically translates these to the corresponding addresses. For example, the register that contains the status information is at address 0x03, input port A at 0x05, etc. If we want to use variable names for these memory addresses, we would write:

STATUS equ 0x03 ; The label “STATUS” is associated with the address 0x03

PORT equ 0x05 ; The label “PORT” is associated with the address 0x05

3. Origin of the program.

Before starting to write machine instructions, you must define the address of the program memory from which you want to start loading the program. The org directive is used for this. In PICs, the origin of the program is always set at address 0x00 because that is where the program starts executing after doing a reset. We will define the origin as follows:

org 0x00 ; Program start

When the program handles interrupts, it does not start to load the program from address 0x00, because if an interruption is generated, the program that attends it starts at address 0x04 (interrupt vector). In this case, what is usually done is to put at address 0x00 a jump to an address of the program memory after the reset vector, for example we would jump to a position labeled START which is at address 0x05.

```
        org 0x00                    ; The following instruction will be at the beginning
        of memory

        goto START                  ; Jumps to the address labeled HOME

        org 0x05                    ; The next instruction will be at address 0x05

START

        -------

        -------

        End
```

4. Body of the program and end.

After indicating the address where the program will begin to load, follow the body of the program composed of the machine instructions and their operands.

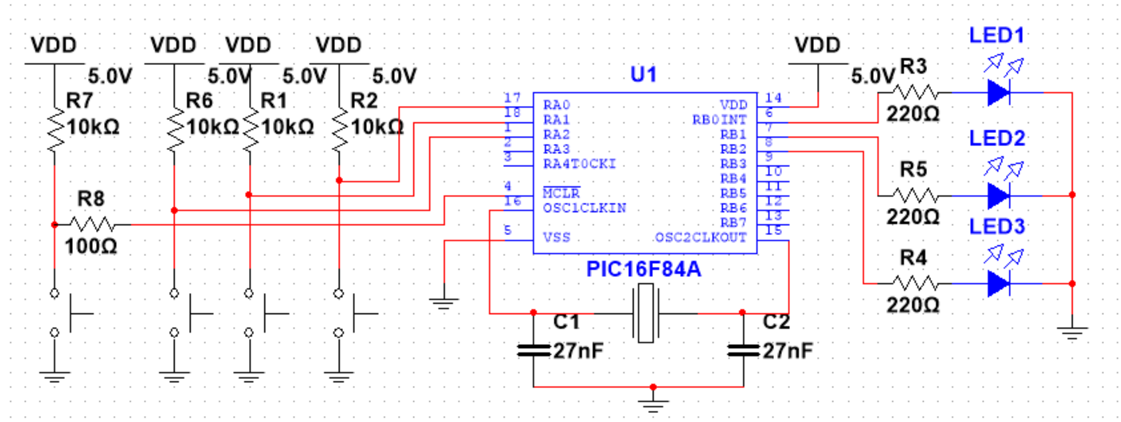


Figure 19. Sample PIC 16F84 Microcontroller

The PIC16F84 has two doors, PORT A (RA0-RA4) and PORT B (RB0-RB7). Each of the lines of these doors can act as an entrance or an exit. The value of the data entering or leaving each port is materialized in two positions in RAM, PORTA for port A and PORTB for port B (figure 6.21). There are two other registers in the RAM memory (TRISA and TRISB) that are used to configure the PORTA and PORTB lines as input or output, in such a way that when a '1' is recorded in one of the bits, the corresponding line of the affected gate acts as an input, whereas if a '0' is recorded, it acts as an output.

In positions 5 and 6 (Figure 6-22) of bank 0 the data registers of the doors are located and in the same positions of the memory, but in bank 1, there are the corresponding configuration registers. If you want to access bank 1 positions, set bit 5 (RP0) of a register called STATUS to ‘1’.

For example, if you wanted to configure all the lines of gate A as inputs and those of gate B as outputs, you would have to address bank 1 (bit 5 of STATUS = ‘1’), load the TRISA bits with ‘1’ and with ‘0’ all TRISB bits. Once the gates have been configured, bit 5 of the STATUS register will have to be set back to ‘0’ (return to bank 0) to be able to read the information entered by the lines configured as inputs or send through the lines configured as outputs.

```
List p = 16f84
STATUS equ 0x03
PORTA equu 0x05
PORTB equ 0x06
org 0x00
goto HOME
org 0x05                ; skip interrupt vector
BEGINNING
bsf STATUS, 5                   ; switch to bank1
movlw b’00000000 ’              ; W ¬‘ 00 ’
movwf PORTB                     ; TRISB¬W (PORTB outputs)
movlw b’00011111 ’              ; W ¬‘ 1F ’
movwf PORTA                     ; TRISA¬W (PORTA inputs)
bcf STATUS, 5                   ; switch to bank0
LOOP
movf PORTA, W                   ; W ¬ PORTA
movwf PORTB                     ; PORTB ¬ W
goto LOOP
end
```

References

Wilmshurst, T. (2006).*Designing embedded systems with PIC microcontrollers: principles and applications*. Elsevier.

Predko, M. (2002).*Programming and customizing PICmicro microcontrollers* (Vol. 2). Upper Saddle River, NJ: McGraw-Hill.

Bates, M. (2011). *PIC microcontrollers: an introduction to microelectronics*. Elsevier.

Peatman, J. B. (1998). *Design with PIC microcontrollers*. Pearson Education India.

Sanchez, J., & Canton, M. P. (2018). *Microcontroller programming: the microchip PIC*. CRC press.

Borowik, B. (2011).*Interfacing PIC microcontrollers to peripherial devices* (Vol. 49). Springer Science & Business Media.

Smith, D. W. (2006). *PIC in practice: a project-based approach*. Elsevier.

Ibrahim, D. (2011). *Pic Basic Projects: 30 Projects Using Pic Basic and Pic Basic Pro*. Elsevier.

Di Jasio, L. (2007). *Programming 16-bit PIC Microcontrollers in C: Learning to Fly the PIC 24*. Elsevier.

Verle, M. (2008). *PIC microcontrollers* (pp. 36-39). Mikroelektronika.

Iovine, J. (2002). *PIC microcontroller project book*. McGraw-Hill Professional.

Shea, J. J. (2005). Programming the PIC microcontroller with Mbasic [Book Review]. *IEEE Electrical Insulation Magazine*, *21*(6), 44-45.

Katzen, S. (2006). *The Quintessential PIC® Microcontroller*. Springer Science & Business Media.

Siegesmund, M. (2014). Embedded C programming: Techniques and applications of C and PIC McUs. Newnes.

Predko, M. (2002). *Programming and customizing PICmicro microcontrollers* (Vol. 2). Upper Saddle River, NJ: McGraw-Hill.

[URL1] www.mikroe.com

[URL2] https://www.microchip.com/

[URL3] https://www.statista.com/statistics/891620/microchip-technology-net-sales-by-product-line/

[URL4] https://microchipdeveloper.com/8bit:summary

[URL5] https://www.icinsights.com/news/bulletins/MCUs-Sales-To-Reach RecordHigh -Annual-Revenues-Through-2022/

[URL6] https://ww1.microchip.com/downloads/en/DeviceDoc/30009630M.pdf

About the Authors

Sabri KOCER, PhD, He graduated from the Electrical Engineering Department of Selcuk University. He completed his graduate and his doctorate in Gazi University. Currently, Necmettin Erbakan University, Faculty of Engineering, Computer Engineering is working. Electronics, Computer, Telecommunication, Signal Processing and Biomedical studies in the area.

Email: skocer@erbakan.edu.tr , Orcid: 0000-0002-4849-747X

Resul BUTUNER, is a Computer Teacher at Adil Karaagac Vocational and Technical Anatolian High School in Konya, Turkey. He has a master's degree in Computer Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, robotic coding, data mining and augmented reality applications. He is an instructor in the field of Robotic coding within TUBITAK. He continues to write a book in the field of robotic coding at the Ministry of National Education. He worked as a coordinator in budgeted projects related to student education.

E-mail: rbutuner@gmail. com, Orcid: 0000-0002-9778-2349

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 19%.

To Cite This Chapter:

Kocer, S. & Butuner, R. (2021). Pic Microcontroller. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 126 –148). ISRES Publishing.

ZigBee

Tarik UNLU
Adnan Akgul Special Education Vocational High School

Zeki KUCUKKARA
Konya Adil Karaagac Vocational and Technical Anatolian High School

Orhan OZYURT
Konya Adil Karaagac Vocational and Technical Anatolian High School

Introduction

ZigBee, known as low speed wireless personal local area network (LR-WPAN: Low Rate Wireless Personal Area Network) communication in wireless communication technologies, enables many different devices to establish a connection so that they can work simultaneously. Zigbee is an IEEE 802.15.4-based feature for a suite of high-level communication protocols used to build personal area networks with small, low-power digital radios such as home automation, medical device data acquisition and other low power consumption. It is designed for small-scale projects with low bandwidth needs and wireless connectivity.

Working on ZigBee-style networks has been started since 1998 due to the inadequacy of wireless communication technologies such as Wi-Fi and Bluetooth in some applications. In these studies, the main target was to produce products that are low cost, high reliability, energy efficient and suitable for establishing networks for imaging and management. The installation of ZigBee 1.0 was realized by IEEE in May 2003 with the completion of the 802.15.4 standard. On December 14, 2004, ZigBee features were approved. ZigBee was first released in June 2005. ZigBee 2007 was released on September 30, 2007 with new technical features. Finally, home automation, the first ZigBee application profile, was put into use on November 2, 2007 (Yuksel, 2010).

ZigBee Alliance; It is a worldwide unit responsible for the standards of ZigBee technology, founded by many companies to produce reliable, low cost and low power consumption products. This community; works on the standards of products for display and control purposes.

ZigBee Protocol Stack

The ZigBee protocol stack consists of a series of blocks called layers, based on the International Organization for Standardization (ISO) open system interconnection (OSI) reference model. These layers are defined by IEEE 802.15.4 and the Zigbee Alliance. The OSI model defines how applications running on devices with network connectivity will communicate with each other. Layers transmit data and command directly the upper and lower layers. It consists of seven layers in the OSI model. Zigbee model uses layers which are important for low power and low data rate wireless network communication. The Zigbee model consists of four layers: physical layer, media access layer, network layer and application layer. Physical layer and Media Access Control sublayer are defined by IEEE 802.15.4, while Network layer and Application layer are defined by Zigbee Alliance (Ali, 2020).

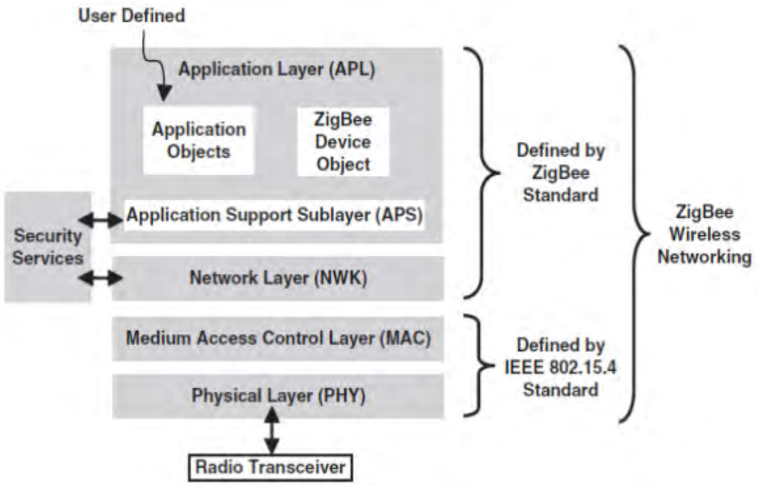


Figure 1. ZigBee and IEEE 802.15.4 Protocol Stack

Physical Layer

The physical layer provides two services, PHY data services and PHY management service. The PHY data service enables the PHY Protocol Data Unit to be transmitted and received over the physical radio channel. The PHY management service Physical Layer Management Asset service interfaces to the access point. There are three available frequency bands defined by IEEE 802.15.4, 868 MHz in Europe, 915 MHz in the USA and Australia, and 2.4 GHz worldwide. The 2.4 GHz band provides a data rate of 250 kbps, 915 MHz provides 40 kbps, and 868 MHz provides a 20 kbps data rate.

MAC Layer

The Mac layer is the layer where channel access, frame update, pointer management and association with the PAN coordinator functions. It provides a secure connection between two nodes. It performs processes such as encryption of data and access of data. The other task is to watch when the channel is empty for transmission (Sagır, 2016).

Network Layer

The network layer handles the setup of the network. Allows a new device to join or leave the network. Configures stack settings. It performs the process of assigning the addresses of the devices newly joined to the network. It determines the direction discovery of packets sent on the network. Cancels unreliable packages (Bascifci, 2011).

Application Layer

The application layer handles the management of the network. It provides a secure communication between network devices. It regulates the operations that the devices will perform with each other. It enables the use of layer services in line with the requirements of two or more devices and directs them accordingly (Bascifci, 2011).

ZigBee Network Architecture

ZigBee technology basically enables devices to communicate with each other using RF technique. ZigBee network constitutes the main component of WPAN. It consists of various devices classified as physical and logical. Devices in the physical category can be classified as a Fully Functional Device (FFD) and Reduced Function Device (RFD). FFD and RFD devices act as a sensor node and a control node. However, only the routing functions of a ZigBee network are performed by FFD devices. Logical devices are divided into three as ZigBee Coordinator (ZC), ZigBee Router (ZR) and ZigBee End Device (ZED). Among these logical devices, ZC is the most capable device to build the network. Each ZigBee network must have a ZC. ZR functions as intermediary devices that allow data to pass to other devices. ZED, on the other hand, has limited functionality for communicating with main nodes. ZigBee network architecture is shown in Figure 2 (Ali, 2020).

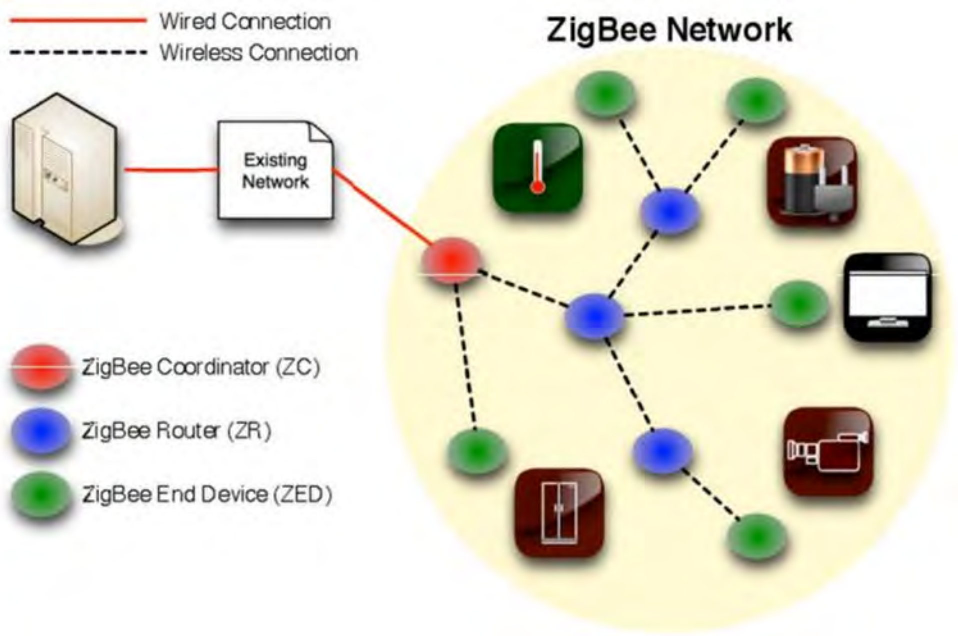


Figure 2. ZigBee Network Diagram

ZigBee Network Topology

ZigBee supports three different types of wireless network topologies: star, tree and mesh topologies.

Star Topology

Star topology is the simplest and most limited topology found in ZigBee. All devices in the star network topology are connected to a single coordinator node and all communication is carried out through this coordinator. The biggest disadvantage of this topology is that when the network coordinator becomes inoperable, the network also becomes inoperable. Because there is no alternative way from source to destination. Figure 3 shows the structure of the star topology.

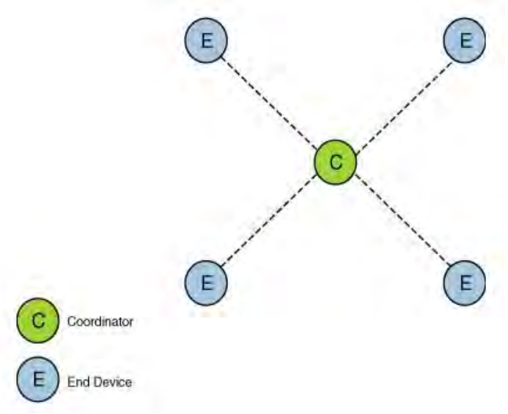


Figure 3. Star Topology Structure

Mesh Topology

It is one of the most flexible topologies offered by ZigBee. Unlike star topology, there can be many different paths on the network, and each device on the network can communicate with other devices on the same network. Since it has a route finding feature, if any device fails on the network, it finds the best route to a specific node. In mesh topology, it is easy to add and remove devices from the network. Figure 4 shows the structure of the mesh topology.

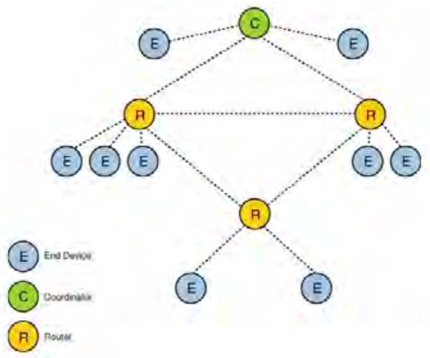


Figure 4. Mesh Topology Structure

Tree Topology

In tree topology, the coordinator and the router are called upper nodes, and the end devices are called sub nodes. Direct communication can only occur between a parent node and a child node. The tree topology is shown in Figure 5.

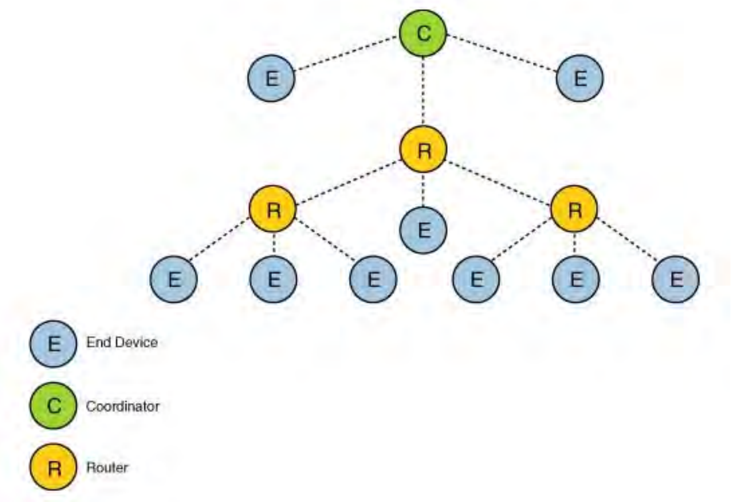


Figure 5. Tree Topology Structure

ZigBee Power Consumption

As seen in the graph below, ZigBee devices save energy by going into sleep mode depending on the data transmission times. In this way, the usage time of the battery is quite prolonged with the sleep cycles that last for hours (Gislason, 2008).

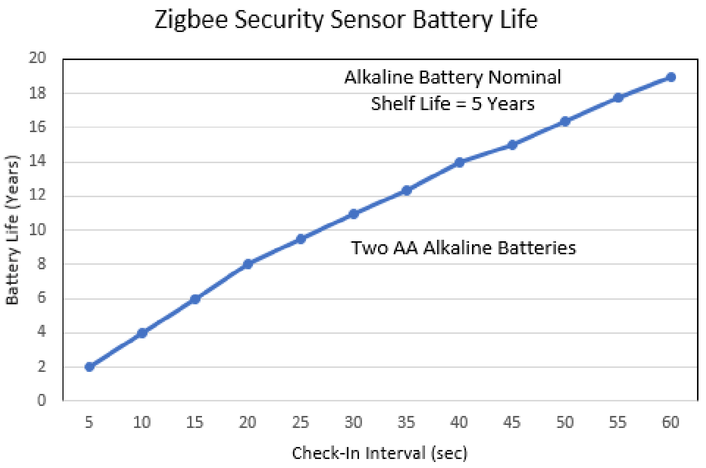


Figure 6. ZigBee Sensor Battery Life

Table 1. Comparative Power Consumption Test Results Of Bluetooth, Zigbee And Wifi

	Bluetooth	ZigBee	Wi-Fi
IEEE Spec	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.11b
Type of Module	HC-05	XBee Series 1	Arduino Yun*
Sleeping Mode	9 µA	12 µA	30 µA
Awake Mode	35 µA	50 µA	245 µA
Transmitting Mode	39 µA	52 µA	251 µA
Receiving Mode	37 µA	54 µA	248 µA
Power Supply	3.3 V	3.3 V	5 3.3 V
*The ESP8266 module which is also powered by 3.3 V could be used as the Wi-Fi modüle. Arduino Yun was used so as not to duplicate Wi-Fi modüle.			

The table above shows the power consumption of low energy wireless standards. (Kazeem et al., 2017)

Security

ZigBee uses AES-128 bit, a powerful encryption technique. This encryption technique can prevent attacks on the network. ZigBee systems provide evidence and check whether the message has arrived at the correct location. In addition, with the high security techniques used in the systems, the attacker is prevented from acting like another device. This evidence used can be used at the device level as well as at the network level. Network level authentication used in ZigBee systems is provided using a common network key. At the device level, authentication is achieved using only one connection key. Encryption in ZigBee systems can also be used as network level and device level evidence. The public network key is used for network level encryption. In this way, the attacker is prevented from infiltrating the system by using very little memory. The shared key is used for device level encryption (Yuksel, 2010).

Scope of Application

Distance and Accuracy

The communication distance between Zigbee devices can be up to 100 meters and this distance can be reduced according to the difference of the environment in which these devices are located. In addition, the size of the data packet to be sent affects the success of the transmission. 200 different data strings with a length of 64 bytes were sent by the transmitter and the successful transmission percentage values in Table 2 were obtained by comparing with the data received by the receiver (Somay, 2009).

Table 2. Zigbee's Transmission Distance (according to the medium)

	Distance	Interior	Outdoor
Unimpeded	2 meter	%100	%100
	4 meter	%100	%100
	10 meter	%100	%95,5
Unimpeded, Mobile	2 meter	%100	%100
	4 meter	%99	%100
	10 meter	%96	%95
Glass Barrier	2 meter	%100	----------
4 meter			
%100			
----------			
	10 meter	%99	----------
Glass + Plastic Barrier	2 meter	%90	----------
	4 meter	%86,2	----------
	10 meter	----------	----------

Usage Areas of ZigBee

The use of ZigBee networks is widespread. Usage areas and sample network models are discussed in many different categories in the literature.

- Commercial building automation (such as stock tracking, ambient temperature tracking, energy level control)
- Home automation (such as fire, indoor temperature and humidity control), home entertainment (such as smart lighting, film and music, sound and video systems adjustment)
- Livestock sector (tracking the productivity of dairy cows, detecting missing animals in the pen)
- Agriculture and agricultural crop protection (such as plant height, leaf size measurements)
- Mobile applications (like m-payment, m-monitoring, m-security and access control, m-healthcare and tele-assist)
- Other applications are automatic measurement reading, wireless telemetry, chemistry, paint, pharmaceutical industry (such as monitoring chemical processes, product quality control), water treatment, waste cleaning (in gigantic water treatment plants sensors are added to the section where each pump is located and real-time measurement data is sent to the control room)

In addition to all these, it is seen that ZigBee networks are preferred in a wide range of areas such as military applications, ecological monitoring studies and medical monitoring applications, which are one of the exit applications of this technology.

Comparison of ZigBee and Other Wireless Communication Technologies

As seen in Table 1 and Figure 1, ZigBee has an advantage over other wireless communication technologies in terms of long battery life and network expansion capacity, while it falls behind in terms of data transmission speed in Table 3. These features make ZigBee technology a suitable candidate for sensor networks and control systems, on the other hand, it makes it disadvantageous in applications requiring large-scale data transfer (Kizilirmak, 2012).

Table 3. Comparison of IoT Based Wireless Communication Protocols

	Bluetooth	Wi-Fi	ZigBee
Frequency	2.4 GHz	2.4 GHz 5 GHz	868 MHz 915 MHz 2.4 GHz
Modulation	FHSS	QPSK OFDM QAM	DSSS BPSK O-QPSK
Range	10 metres	100 metres	10-100 metres
Data Rate	1 Mbps	11 & 54 Mbps	250 Kbps
Topology	Star and Mesh	Star and Point-to-Point	Star and Mesh
Network size	7	Depends on the number of IP addresses	64000
Power consumption	Medium	High	Low
Number of Channels	19	13	11 and 16

ZigBee Modules and Features

While some modules running on the IEEE 802.15.4 protocol contain only a receiver / transmitter chip called transceiver, a microcontroller and transceiver chips are preferred in many other modules together. By using additional amplifier circuits in some modules, the wireless communication range has been increased due to the signal strength. Generally 900 MHz Xbee products licensed in North America and 868 Mhz products licensed in Europe.

CC2530 ZigBee Module

The CC2530 is a complete System-on-Chip (SoC) solution for IEEE 802.15.4 ZigBee applications. It allows to create durable network points with very low material costs. The CC2530 combines a high performance RF transceiver chip with a 8051 microcontroller, flash memory, 8-KB RAM memory, and many other features. CC2530 has four different flash memory options: 32/64/128/256 KB. CC2530 has different operating modes suitable for applications where low power consumption is required. The possibility of fast switching between operating modes brings low energy consumption. (Texas Instruments, 2011)

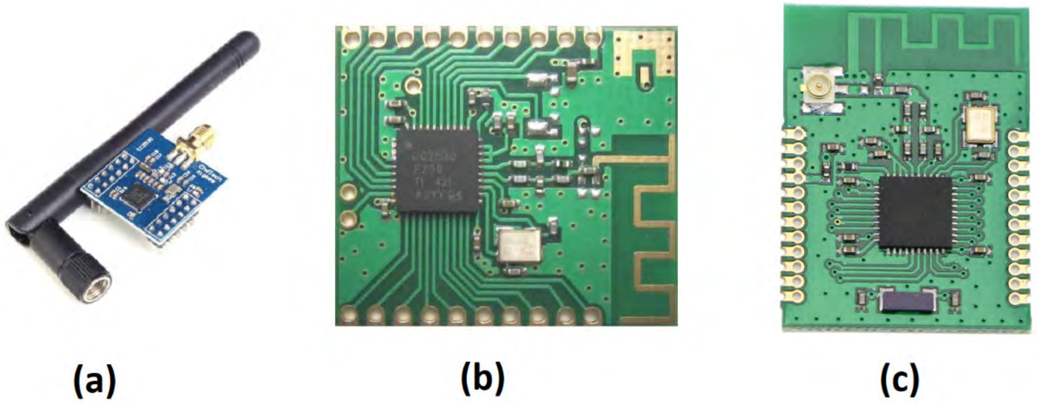


Figure 7. CC2530 Module Types, a: with SMA Antenna Connector, b: with PCB Antenna, c: with PCB Antenna and UFL Antenna Connector

There are different studies conducted indoors and outdoors using the CC2530 module. These include applications such as sound transmission systems to be used in emergency situations. (Mousavi et al., 2012)

XCore2530 ZigBee Module

XCore2530 is a CC2530 chip combined with the RXF2401 amplifier. In this way, it is used to communicate at greater distances. XCore2530 equipped with SMA antenna.

For example, in a study to detect forest fires, it was possible to use the XCore2530 module due to its long distance communication capability. (Hlaing & Aung, 2019)

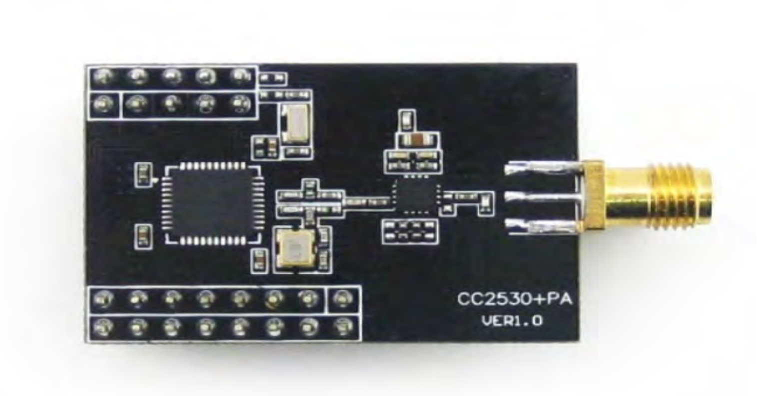


Figure 8. XCore2530 ZigBee Module

XBee

XBee modules allow the creation of complex ZigBee-based mesh networks. It provides a solid and simple communication between serial port data transfer and microcontrollers. XBee is ideal for general purpose communication applications thanks to its 2mW output power, 120 meters communication distance, internal antenna, 250 kbps data transmission and 8 digital IO pins (Ghazal & Khatib, 2015).

XBee series 1 modules can be used between two devices (P2P communications) without any configuration. In addition, series 1 modules can be used with wired serial communication without being configured with software such as XCTU. However, series 1 modules are not compatible with newer modules such as series 2, 2.5 or ZB.

Figure 9. xBee Series 1.

XBee 2

xBee Series 2 modules need configuration even for dual (P2P) usage. Series 2 modules have better power efficiency and longer communication distance than Series 1 XBee modules.

XBee Pro

xBee Pro modules are pin compatible with a regular XBee module, so they are easy to use together or interchangeably. Some XBee Pro modules take more space than regular XBee modules. XBee Pro modules consume more power and have a longer communication distance. XBee Pro modules can be used with chip-shaped antenna, SMA antenna, cable antenna or UFL antenna like other XBee modules.

Figure 10. xBee Pro with Wire (whip) Antenna.

ZigBee Module Connections with Microcontroller Cards

CC2530 and XBee family zigBee communication modules support SPI (Serial Peripheral Interface) and UART (Universal Synchronous Asynchronous Receive Transmit) protocols. UART or RS232 standard is slower than SPI protocol and only two devices can be connected to each other at the same time. SPI protocol has a master / slave structure and more than two devices can communicate with each other.

ZigBee Module Configuration Software

There are several configuration software that support different hardware. These software allow viewing and setting of Router, coordinator and endpoint devices in the coverage area of the ZigBee hardware connected to the computer. These software also allow the arrangement of the ZigBee network topology.

XCTU

XCTU software is compatible with XBee modules from Digi Corporation and it is free of charge. It allows ZigBee modules to be set up, configured and tested. It is possible to communicate with the modules using API and AT consoles.

Name: RT1-R2D2
Function: ZigBee Router API
Port: COM1 - 9600/8/N/1/H - API1
MAC: 0013A20040A198CD

Name: CO-YODA
Function: ZigBee Coordinator API
MAC: 0013A20040A9E788

Name: RT1-WINDU
Function: ZigBee Router API
MAC: 0013A20040A9527D

Name: RT2-OBI-WAN
Function: ZigBee Router API
MAC: 0013A20040A9E883

Name: RT4-LUKE
Function: ZigBee Router API
MAC: 0013A20040A9E756

Name: BAYMAX
Function: XBee DigiMesh 2.4
Port: COM6 - 9600/8/N/1/H - AT
MAC: 0013A20040D28039

Radio Configuration [RT1-R2D2 - 0013A20040A198CD]

Product family: XBP24BZ7
Function set: ZigBee Router API
Firmware version: 23A7

Networking
Change networking settings

ID PAN ID	C3B0	
SC Scan Channels	7FFF	Bitfield
SD Scan Duration	3	exponent
ZS ZigBee Stack Profile	0	
NJ Node Join Time	FF	x 1 sec
NW Network Watchdog Timeout	0	x 1 minute
JV Channel Verification	Enabled [1]	
JN Join Notification	Disabled [0]	
OP Operating PAN ID	C3B0	
OI Operating 16-bit PAN ID	5EL9	
CH Operating Channel	F	
NC Number of Re...ing Children	C	

Addressing
Change addressing settings

SH Serial Number High	13A200

Figure 11. Sample XCTU Configuration Layout.

References

Ali, A. I. (2020). *ZigBee and LoRa based communication for energy efficient smart buildings* (Publication No. 638989) [Master's thesis, Yıldız Technical University]. Council of higher education thesis center https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=_F5QEpayDXGqGZlp9XiFtNNzIY8K5xfp9k5VMuM51H4ZxTq9w6-zQCGjF0zqZ11_

Ali, A. I., Partal, S. Z., Kepke, S. & Partal, H. P. (2019). *ZigBee and LoRa based Wireless Sensors for Smart Environment and IoT Applications*, 2019 1st Global Power Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 2019, pp. 19- 23, doi: 10.1109/GPECOM.2019.8778505.

Arslan, H. (2021, May 1). *Zigbee Network Diagram*. https://argevetasarim.com/heterojen-kablosuz-aglarda-zigbee/zigbee-ag-semasi/

Bascifci, N. (2011). *Zigbee based mobile health monitoring system design and implementation* [Unpublished master's thesis]. Selcuk University Institute of Science and Technology.

Erten, A. (2019). *Research on zigbee technology and a sample applıcatıon on transmission of bıomedıcal signals* (Publication No. 564065) [Master's thesis, Dicle University]. Council of higher education thesis center https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=FgmkGchPKo23qQqBeqzVZqDpNkacvmXHe8El3_jSLLPSkFTEySTx9eayUjwo_10x

Gislason, D. (2008). *Zigbee Wireless Networking. Chapter 1 - Hello ZigBee*. Newnes. https://doi.org/10.1016/B978-0-7506-8597-9.X0001-8

Kazeem, O., Akintade, O., & Kehinde, L. (2017). Comparative Study of Communication Interfaces for Sensors and Actuators in the Cloud of Internet of Things. *International Journal of Internet of Things*. 6(1): 9-13. 10.5923/j.ijit.20170601.02.

Sagir, S. (2016, 10 January). *Zigbee Layers*. https://selimsagir.com.tr/2016/01/10/zigbee-katmanlari/

Sagir, S. (2015, 16 December). *Zigbee Topologies*. https://selimsagir61.wordpress.com/2015/12/16/28/

Somay, A. (2009). Implementation of a wireless measurement probe using ieee 802.15.4 (ZigBee) standard (Publication No. 244524) [Master's thesis, Karadeniz Technical University]. Council of higher education thesis center https://tez.yok.gov.tr/UlusalTezMerkezi/ TezGoster? key=

CwVIqqBuz1VkysVpueogAagQvZYtLKpZxAHkU2bynnNQKF5rjSSf_Y9OckcYMZ1y

Kizilirmak, E.Y. (2012). Industrial crane control using zigbee (Publication No. 321325). [Master’s thesis, Hacettepe University]. Council of higher education thesis center https://tez.yok.gov.tr/UlusalTezMerkezi/TezGoster?key=rcbWnuqW6HxCZ_98ARapgkBGmQv_Qw-hpnKigO39huUrBSs3bWT-M921fblb7-yV

Yuksel, E. (2010). Design of a Zigbee architecture for telemetering (Publication No. 282571). [Master’s thesis, Istanbul University]. Council of higher education thesis center https://tez.yok.gov.tr/UlusalTezMerkezi/TezGöster?key=zD1B0cW7zzD1B0cW7zVr3VcnZjitVXoyBQbbI39btMILVJXihsJLeEIvl4DhFqPRCOr2Sj_CY

Texas Instruments (2011). CC2530 Datasheet. A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications. https://www.ti.com/lit/ds/symlink/cc2530.pdf

Mousavi, S., Aksu, S., & Kurt, G. S. (2012). Voice Over Zigbee Wireless Sensor Network Applications, 20th Signal Processing and Communications Applications Conference (pp. 1-4). Proceedings.

Hlaing, S. & Aung, S. N. L. (2019). Monitoring of Forest Fire Detection System using ZigBee. International Journal of Trend in Scientific Research and Development. 3(4):1332-1335.

Ghazal, B. & Khatib, K. (2015). Smart Home Automation System for Elderly, and Handicapped People using XBee. International Journal of Smart Home, 9,203-210.

About the Authors

Tarik UNLU, is an Computer Teacher at Adnan Akgul Special Education Vocational High School in Konya, Turkey. I have a master's degree in Mechatronics Engineering from Karabuk University. His main areas of interest are artificial intelligence, robotic coding, web programmes and wireless network. I am an instructor in the field of Robotic coding within TUBITAK. E-mail: tarik1945@gmail.com, Orcid: 0000-0001-6543-2320

Zeki KUCUKKARA is a Computer Teacher at Adil Karaagac Vocational and Technical Anatolian High School in Konya, Turkey. He has a master's degree in Computer Engineering from Necmettin Erbakan University. He works on image processing. His

main areas of interest are artificial intelligence, image processing, desktop application and mobile application software.

E-mail: zekikucukkara@gmail.com, Orcid: 0000-0002-5204-0819

Orhan OZYURT, is an Electronics Teacher at Adil Karaagac Vocational and Technical Anatolian High School in Konya, Turkey. He has a master's degree in Electronics and Computer Education from Selcuk University. His main areas of interest are block chain, artificial intelligence, robotic and coding applications.

E-mail: orhanozyurt@gmail.com, Orcid: 0000-0003-2762-4481.

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 17%.

To Cite This Chapter:

Unlu, T. , Kucukkara, Z. & Ozyurt, O. (2021). ZigBee. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 149–162). ISRES Publishing.

The LilyPad Arduino

Hakan YUKSEL
Applied Science University of Isparta

Introduction

LilyPad is a wearable combination of Arduino that helps create interactive textiles or e-textiles (electronic textiles) (Buechley et al., 2008). The LilyPad project started in 2006 as an academic research project (Buechley, 2006). However, given on October 2007, SparkFun Electronics and Leah Buechley has become a commercial venture for sales (Buechley & Hill, 2010). LilyPad has found its place in the literature as the first kit designed for electronic textiles. Electronic textile design involves the use of several modules sewn with conductive thread to provide the necessary electrical connections to the fabric. LilyPad; it consists of a series of sewable electronic modules, including a spool of conductive thread RGB LED, accelerometer, temperature sensor and a sewable Arduino microcontroller. E-textiles are created by sewing these modules onto a fabric backing with conductive thread that provides both physical and electrical connections between parts. This structures, LilyPad microcontroller, it is determined by programming using the Arduino development environment (Buechley, 2009). The microcontroller can be programmed like regular Arduino boards using the usb connection.

LilyPad 's first version released in 2006 (Buechley, 2006). LilyPad the first version of the microcontroller structure and top/bottom view is shown in Figure 1. This structure to form conventional and electrically conductive fabric of a fabric PCB, a packed hole of a microcontroller ATTiny26 combination is connected. The patch consists 63.5 x 63.5 x 12 mm of (2.5 x 2.5 x 0.5 inches) and hard cover area 20 x 46 mm (0.825 x 1.825 inches) dimensions. There are 17 conductor transitions from the microcontroller to the pins. An electrical connection between the fabric and the yarn is established by means of a conductive yarn. Two of these conductors provide power and ground on the microcontroller. The remaining conductors serve as general-purpose input/output (I/O) channels that can be used to receive sensor input and control output devices such as light-emitting diodes (LEDs), loudspeakers, and vibrating motors (Buechley & Eisenberg, 2008).

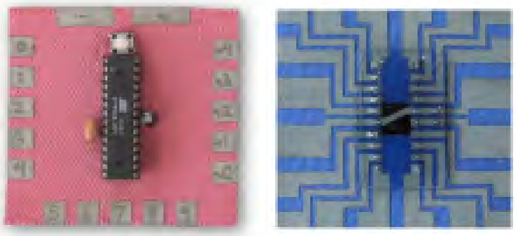


Figure 1. First Version of LilyPad

The circuit board layout was redesigned to make it both aesthetic and functional (Figure 2). First of all, it has made it easier to use circular format surface mount (SMD) components. Reproducing the thin right-angle lines of traditional circuit boards with precision conductive fabric at SMD scale was extremely challenging, while a circular layout allowed it to create solid triangular traces radiating from the center of the board (Buechley & Eisenberg, 2009).

Figure 2. LilyPad Enhanced Design Interface

Hardware

LilyPad ProtoSnap is a development board with twelve LilyPad components connected via conductive means to the LilyPad Arduino Simple microcontroller (Table 1). The connection component that performs the tracks in most is hidden, but by reference ProtoSnap (LilyPad and the combined circuit formed by hardware board) of each component on is attached LilyPad to the number of acid sewing tab white screen printed with a close tag is located (Sparkfun. (n.d. -a)).

LilyPad ProtoSnap, LilyPad simple Arduino and pre-wired LilyPad component classic with a set LilyPad Arduino protosnap development card is a kit (Figure 3).

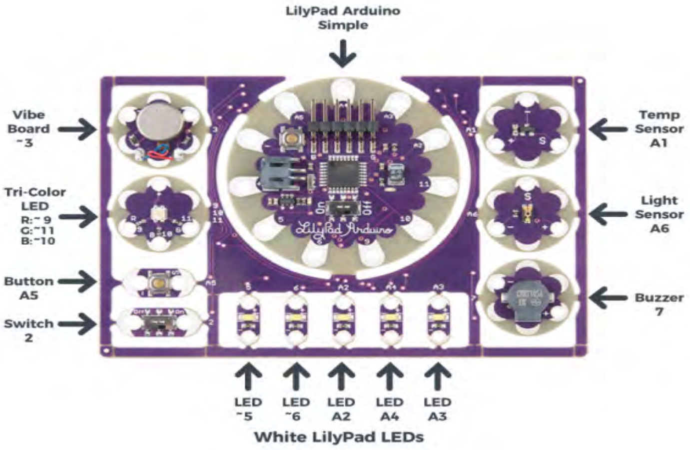


Figure 3. LilyPad Arduino ProtoSnap

LilyPad Arduino that has a special function of some sewing tab:

- The ‘A’ in front of the number denotes a tab that can act as an analog input. These strut tabs can read sensors that generate variable voltage, such as the light sensor built into the LilyPad Arduino Simple.

- The ‘~’ symbol in front of LilyPad Protosnap indicates that the tab supports PWM (Pulse Width Modulation) output. These tabs, LEDs etc. It can output an “analog” signal that can be used to change the brightness.

Table 1. LilyPad Components

LilyPad Component	Arduino Pin	Connected to LilyPad Arduino Simple Sew Tab	Description
LilyPad Vibe Board	~3		Haptic feedback is provided with the motor’s vibration and is controlled by the LilyPad Arduino Simple.
LilyPad Tri-Color (RGB) LED - Red	~9	✓	The tri-color LED’s red is controlled by the LilyPad Arduino Simple.
LilyPad Tri-Color (RGB) LED - Blue	~10	✓	The tri-color LED’s blue is controlled by the LilyPad Arduino Simple.
LilyPad Tri-Color (RGB) LED - Green	~11	✓	The tri-color LED’s green is controlled by the LilyPad Arduino Simple.
LilyPad Button	A5	✓	LilyPad Arduino Simple receives button presses as an input.
LilyPad Slide Switch	2		LilyPad Arduino Simple receives switch state (on/off) input.
LilyPad White LEDs	~5, ~6, A2, A3, A4	✓ (all 5 LEDs)	A set of white LEDs controlled by the LilyPad Arduino Simple.
LilyPad Buzzer (+)	7		A buzzer that create tones controlled by the LilyPad Arduino Simple.
LilyPad Buzzer (-)	12		A buzzer that create tones controlled by the LilyPad Arduino Simple. While normally connected to GND, it can be connected to an Arduino pin.
LilyPad Light Sensor	A6		LilyPad Arduino Simple receives ambient light level input from light sensor.
LilyPad Temperature Sensor	A1		LilyPad Arduino Simple receives temperature from a physical touch based on ambient conditions and body heat with the analog sensor.

Hardware: Components

LilyPad Vibe Board

LilyPad Vibe Board is a small vibration motor controlled by LilyPad Arduino that can be sewn into projects with conductive thread. The component can be used as a physical display on clothing and costumes for haptic feedback.

LilyPad Vibe Board the two upright tabs located: supply (+) and ground (-). A white label for reference next to each tab there. For power, an input voltage is connected to anywhere between 3.3V and 5V. The more voltage provides; the motor can vibrate faster. Due to the amount of current each I/O pin can source, it is recommended to connect the (+) tab to a MOSFET to power the motor when using it with an Arduino. To adjust the density, it would be convenient to use a PWM-capable stitch tab on the LilyPad Arduino (Sparkfun. (n.d. -b)).

Figure 4. LilyPad Vibe Board

LilyPad Tri-Color (RGB)

There are three small LEDs (red, green, blue) inside an RGB LED. Each of these LEDs connected to a sewing tab on all LED and a common anode (positive) connected to the pin.

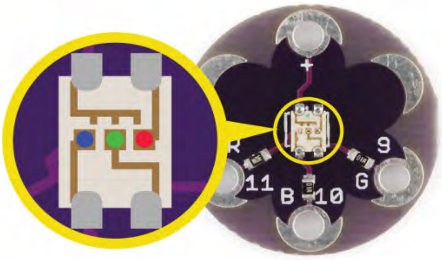


Figure 5. LilyPad Tri-Color (RGB)

The RGB LED on the tri-color LED has 4 connections. These are respectively; indicated as red, green, blue and common anode (Figure 5). R ‘on the LED (Red) tab temporarily 11A, G (Green) 9, B (blue) 10 are connected and used (Sparkfun. (n.d. -c)).

In the scope of analogWrite() from Arduino functions, 0 is 0% (off) and 255% is 100% (on). However, since the component used is the common anode, in the software part it is 100% (on) and 0 for 255, 0% (off). Mixing values for tricolor LED are shown in Figure 6.

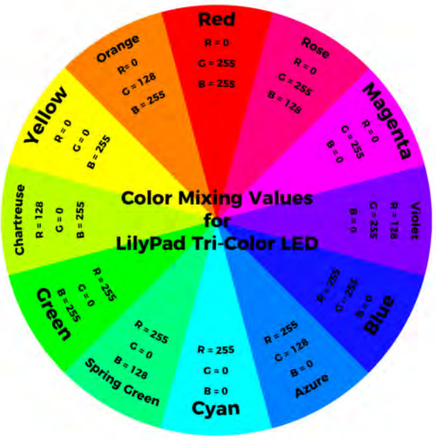


Figure 6. Mixing Values Tri-Color

LilyPad Buttons

Buttons and switches are electronic components that control the flow of current in a circuit. They can act as a simple way to light up an LED or as an input for a microcontroller. Buttons are generally considered to be a kind of switch with a momentary push action.

This compenent is also a type of switch (Figure 7). When you press the button in the middle of the board, it connects the two sewing tabs and allows the current to pass. When the button is released, the connection reopens and the button returns to its place. This button is an example of an instant toggle (Sparkfun. (n.d. -d)).

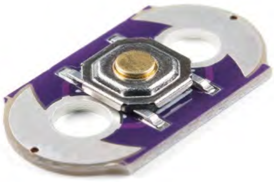


Figure 7. LilyPad Buttons

LilyPad Slide Switch

One condition, to change from one state to another, usually a flip, slide or push so on. physical action is required. This action is called operating the switch for short. Different types have different working methods. There are different types of switches in the LilyPad series by sliding, pushing or even using a magnet to trigger.

LilyPad Slide Switch is shown Figure 8; On/Off label has a small switch. When set to the “Off” position, the parts inside the switch diverge and open (disconnect) the circuit. No current will flow from the switch to components connected to the seam tabs. When the toggle switch is set to the “On” position, the two seam tabs on the switch connect, allowing current to flow through and close the circuit (Sparkfun. (n.d. -d)).

Each of the components of switches, the supply current of 300mA and voltage is 4 volts stated. However, given current with a reduction in the 5-volt work.

Figure 8. LilyPad Slide Switch

LilyPad White LEDs

LilyPad White LEDs (5 pcs) are attached to each other (Figure 9). LilyPad white LEDs are creatively designed and easily sewn into fabric. Component, various input, output, power and sensor together with the board comes. LilyPad White LEDs are washable. LilyPad White LEDs offer a slim 0.8mm build and a 5.5mm x 12.5mm PCB layout (Sparkfun. (n.d. -e)).

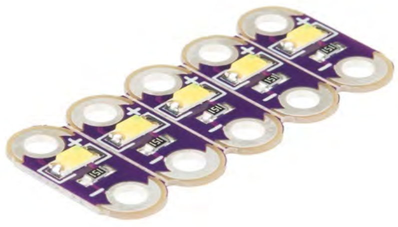


Figure 9. LilyPad White LEDs

LilyPad Buzzer

Lillpad Buzzer are the two I/O pins (Figure 10). With this I/O change, different melodies are formed depending on different frequencies. Volume enough is more, but annoying as it is not loud (Sparkfun. (n.d. -f)).

Figure 10. LilyPad Buzzer

LilyPad Light Sensor

The LilyPad Light Sensor, its stand-up component with a built-in and ready-to-use ALS-PT19 light sensor, is shown in Figure 11. Each sensor generates a voltage between 0 V and 3.3 V, depending on the level of ambient light shining on it. As more light is applied to the sensor, more current will flow from the board via the signal hop to the microcontroller to which the sensor is connected. If the sensor does not receive light, no current flows through it. In a typical indoor lighting situation, the sensor outputs around 1 to 2V. (Sparkfun. (n.d. -g)).

Figure 11. LilyPad Light Sensor

LilyPad Temperature Sensor

Temperature sensor, the set temperature in each 10mV for a Celsius degree to (° C) to give a certain voltage set. The current flowing through the signal tab can be read with an analog tab on the LilyPad Arduino board and converted to degrees Celsius or Fahrenheit via a formula.

The sensor generates an analog voltage that represents the temperature near it. The voltage output of the sensor is linearly proportional to the temperature in degrees Celsius. Once the output voltage of the sensor is known, these equations 1 and 2 can be used to calculate the temperature (Sparkfun. (n.d. -h)).

$$^{O}Celsius = (voltage - 0.5) * 100 \quad (1)$$

Equation 2 is used to convert to Fahrenheit.

$$^{O}Fahrenheit = (^{O}Celsius * 9.0/5.0) + 32.0 \quad (2)$$

LilyPad Arduino Simple

The LilyPad Arduino Simple is the brain of the ProtoSnap process. Microcontroller is divided into various modules in terms of digital pins, analog pins, programming interface and battery inputs as shown in Table 2 (Sparkfun. (n.d. -i); Nayyar, & Puri, 2016).

Table 2. LilyPad Models

Board	Microcontroller	Digital Pins	Analog Pins	Programming Interface	Attachment of the Battery
LilyPad Arduino Simple	ATMega328	9	4	FTDI	JST Connector
LilyPad Arduino USB	ATmega32U4	9	4	USB	JST Connector
LilyPad Arduino SimpleSnap	ATMega328	9	4	FTDI	Built in LiPo
LilyPad Arduino 328 Main Board	ATMega328	14	6	FTDI	Sew Tabs
LilyPad USB Plus	ATMega32U4	10	7	FTDI	Built in LiPo

LilyPad Arduino Simple has an ATmega328 microprocessor controller at its core. A microprocessor is a small computer that can be programmed to interact with LEDs, beeps, and many other peripherals. The ATmega328 microcontroller has many features (Table 3) besides the input and output set and works very well with Arduino software (Lakshmi Lasya, 2021). The features of LilyPad Arduino Simple are as follows:

- 5 I/O pin (Digital),
- 4 pin (Analog),
- ATMega328p microcontroller,
- LED on pin 13,
- On/Off switch,
- Onboard JST connector for charging circuit and 3.7V LiPo battery.

Table 3. LilyPad Arduino Simple Features

I/O Devices	Range Of Value
Microcontroller	ATMega328V
Operating Voltage	2.7 – 5.5.V
Input Voltage	2.7 – 5.5.V
Digital I/O pins	14
Analog Input pins	6
Current per I/O pin	40 mA
Flash Memory	16 KB
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	8 MHz

ATMega328p is a multifunctional microcontroller with command voice working in single cycle. The ATMega328p is generally an 8-bit RISC processor with low power consumption and high performance (Figure 12) (Sparkfun. (n.d. -i)).

It contains 131assembly instructions, known as single-cycle,32 8-bit general-purpose registers and has a completely static operation support. In addition, the ATMega328p microcontroller has an operating speed of up to 20 MHZ. ATMega328p system microcontroller also has 1 KB EEPROM and 32 KB Flash Memory system. For these reasons, it is a frequently used microcontroller system (Nayyar & Puri, 2016).

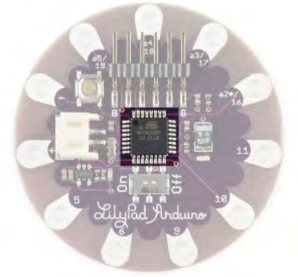


Figure 12. ATMega328p Microcontroller of LilyPad Arduino Simple

As shown in Figure 13, there is a small LED to the right of the ATmega328. This can be classified as the most important component in Arduino. The LED is connected to Arduino pin 13 and can be used for all your flashing needs. In the top left is a momentary button used to reset the ATmega328. This will restart the sketch the Arduino is running from the beginning.

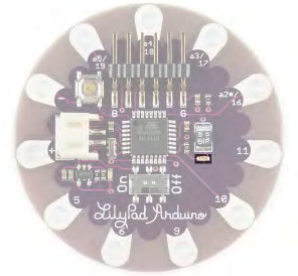


Figure 13. LED of LilyPad Arduino Simple

In Figure 14, there is an On/Off slide switch at the bottom of the ATmega328. There are several ways to power the ProtoSnap LilyPad Development Board, and in most cases the built-in FTDI board is used, possibly to power it using the computer’s USB. If the card is powered by the FTDI card, the On/Off switch has no function. The switch controls power to the Arduino only if powered through the small white connector to the left of the ATmega (Nayyar & Puri, 2016).

Figure 14. On/Off Slide Switch of LilyPad Arduino Simple

The white connector in Figure 15 is a fairly common power connector from the JST family of connectors. Mainly for connecting one of the Lithium Polymer batteries (Sparkfun. (n.d. -i)). LiPos are rechargeable batteries, a battery can be connected to the LilyPad Arduino Simple via the JST connector. P is installed, and a FTDI Basic Breakout is connected, the battery ignorant half of the charge can be investigation.

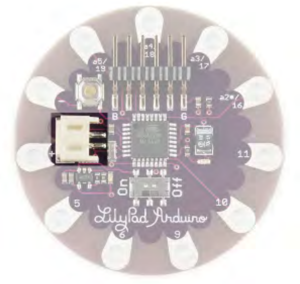


Figure 15. JST Connector of LilyPad Arduino Simple

Powering the LilyPad Development Board

LilyPad can be run in two ways (Sparkfun. (n.d. -j)):

- If you have an available USB power source (computer, 5V USB adapter, USB battery pack, etc.), A USB cable and a card from FTDI can run (Figure 16a).
- If the project is desired to be portable, a rechargeable Lithium-polymer battery can be connected to the JST connector end (Figure 16b).

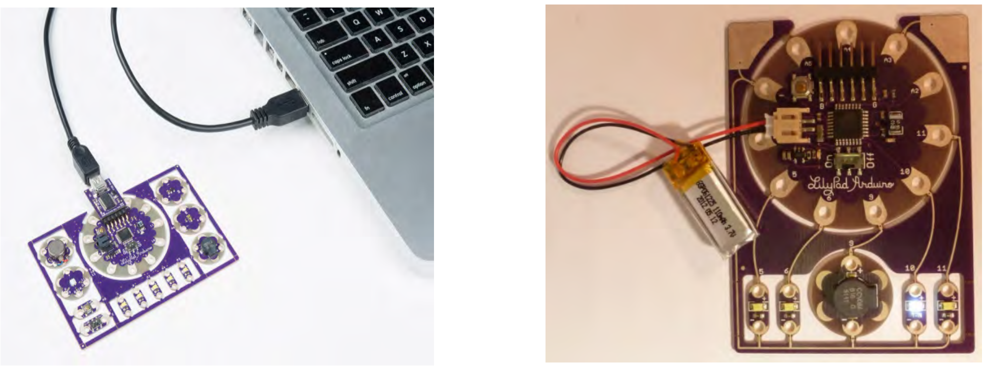


Figure 16. (a) USB Connection (b) LiPo Battery Connection

LilyPad Coding Editor: Arduino Sketch

LilyPad is for programming the Arduino’ s used languages, is a simplified version of the C ++ programming language. This simplified structure is provided by workpieces called functions. The complexity of the details of the definitions for microprocessors or microcontrollers is greatly simplified with Arduino. In addition, it enables the programmer to write code without tiring the programmer with small work pieces that will dominate many hardware (Sparkfun. (n.d. -k)).

The programming language consists of three main structures. These (Galadima, 2014);

I. *The part with definitions.* In this section, variables can be defined, libraries can be included in the program or operations such as naming pins can be done. For example; For a pin that is constantly used in the program, instead of using it as the 13th pin; 13p to the LED 13 connected pine "LED" name largely ensured by ease of use.

II. *Installation part.* This section in which the pins which input of the pins to be output, serial data communication related settings used in the program to be written like an Arduino make the necessary installations for properties. The installation part will run once when the program first starts and will not run again unless the Reset is triggered or the power is cut off.

III. *The Cyclic Part,* is the part where the actual program is executed. All functions of the project we will do are written in this part. This section runs the codes we have written continuously until the energy is cut off or reset. When the program starts from the beginning of this section and reaches the end, it returns to the beginning of this section and continues to run. This is called an infinite loop.

LilyPad Programming

When you want to program LilyPad, you will need to perform the following three steps.

1. LilyPad Development Board, FTDI card and your computer using a USB cable to connect.
2. Arduino the "Board" menu "LilyPad Arduino w/ ATmega328" choose.
3. Arduino's "Port" menu LilyPad the serial port that is connected to select.

1. LilyPad Development Board

After the LilyPad FTDI Board is connected to the headers on the LilyPad Arduino Simple, a mini USB-B cable is attached to the other side of the FTDI board. Plug the other end of the USB cable into any USB port on the computer. The connection process is completed by sliding the switch on the LilyPad Arduino Simple to the on position. If the switch is in the off position, the code is not loaded.

2. Select LilyPad Arduino w/ ATmega328

LilyPad Arduino Simple Tools to Tools > Boards menu LilyPad Arduino w/ ATmega328 is selected (Figure 17).

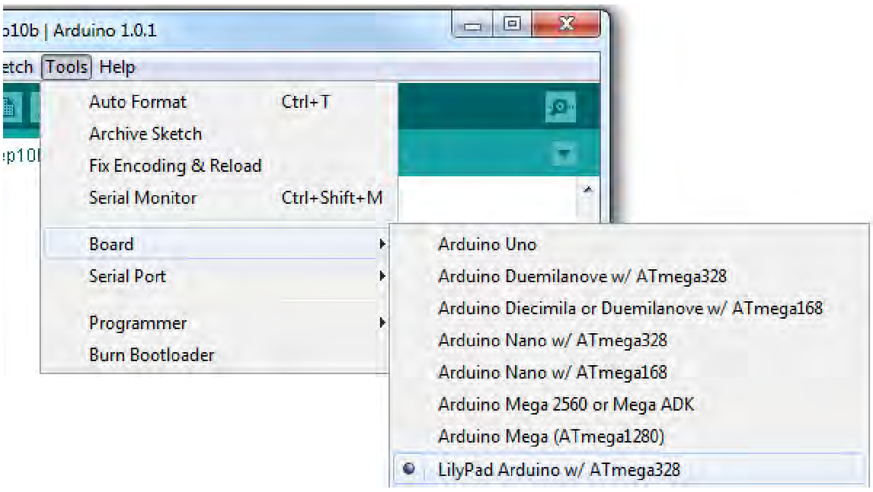


Figure 17. Board Menu LilyPad Arduino is Selected

3.Select Port

The Arduino needs to know which port the LilyPad Arduino is connected to in order to program it. When a USB device is plugged into the computer, the computer will assign a port number to that port. Tools > Serial Port menu, the port to which the LilyPad Arduino is connected is determined (Figure 18).

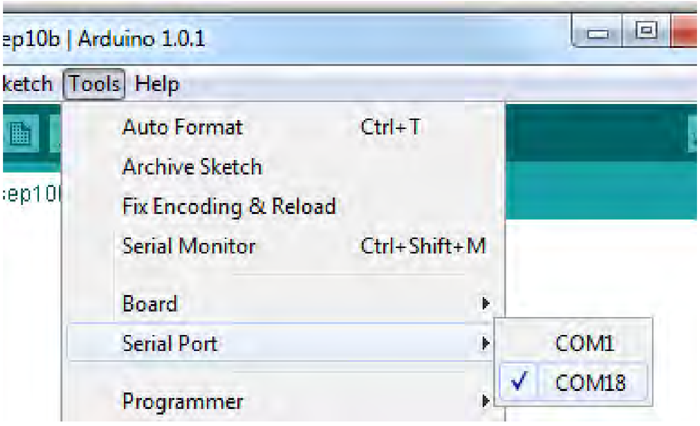


Figure 18. Determining the LilyPad Arduino Port from the Serial Port Menu

Lilypad Application

LilyPad offers a very useful structure for e-textiles and wearable technologies. In particular, the increase in the use of portable technology indicates that the control that consumers can exert over their environment will also increase (Baurley, 2004). Wearable technology can monitor health functions and vital signs, help improve the physical performance of athletes or rehabilitation patients, provide feedback from stimuli in one’s environment, and create new opportunities for aesthetic improvement and decoration in the clothing industry (Berzowska & Bromley, 2007). Mohsen et al. (2021), a medical practice can be worn for long-term health of the sensor system is designed (Mohsen et al., 2021). LilyPad Card was used as a wearable technology in the design of this system. Because of the various functions provided by wearable technology, this type of technology offers numerous opportunities for new product development (Ariyatum, 2005).

LilyPad Arduino is a sewable microcontroller used in textile-based wearable technology and runs on Arduino, an open source programming and prototyping platform. Originally designed as a pedagogical teaching tool, it taught a variety of user's basic computer and electronics skills through an embedded computer system (Buechley & Eisenberg, 2008). Users can sew the LilyPad into a textile-based project and connect the components with conductive fabric and thread (Buechley et al., 2008). It was largely created to expand on what its creators saw as an impressive extension of already existing tech themes as entertainment and automation, which in turn led to the creation of a new demographic (Buechley & Hill, 2010). Lasya (2021), in his study, aimed to help visually impaired individuals reach their goals without any difficulty by using shoes that are uniquely designed for the visually impaired and can be produced to fit easily inside.

The creators of LilyPad point out that human computer interaction and technology use is a third dimension that focuses on whether users can perform tasks or express themselves in ways that were previously thought impossible using a particular technology (Buechley et al., 2008). That is, people can easily purchase the device online and learn how to use it themselves through various open source tutorials. Buechley and Hill (2010) state that the mass consumerism that marked the 20th century will soon be replaced by a context where niche users will maximize their internet resources to create, find, share and consume content that suits their unique interests and needs (Buechley & Hill, 2010). This is in line with the stance of Smelik, Toussaint and Van Dongen (2016) that the relevance of wearable technology will only increase when users interact with design and discover new values and significance through it (Smelik et al., 2016). The Internet has facilitated the rapid spread of DIY concepts to the wider population; Open source tools such as Arduino allow collaborations and sharing to happen easily and quickly and are becoming more and more common in production architecture and design disciplines, creating a context where consumption is no longer passive. Rather, it is an opportunity to build one's creative experience and can lead to a change in attitudes and behaviors that characterize the broader consumer population (Lin, 2014).

Conclusion

Giving open source code application examples of microcontrollers brings development boards to the forefront today. One of the important advantages of development boards is that most development boards are inexpensive and do not require a separate power supply or can be operated with a normal power supply designed by the designer. The emergence of different development boards leaves the designer obliged to make different circuits and develops them in the direction of electronic circuit design. LilyPad offers a different alternative to other development boards as it offers a more flexible and softer content compared to other circuit boards in both wearable technology projects and e-textile projects.

Lilypad, by collecting data on the health system, analyzing and transmitting patient safety enhancing projects also clothes, watches and so on shoes. Considering that it is easy to place, it is quite easy to get values from different sensors.

References

Buechley, L., Eisenberg, M., Catchen, J., Crockett, A., (2008, April). The LilyPad Arduino: Using Computational Textiles to Investigate Engagement, Aesthetics, And Diversity in Computer Science Education. In Proceedings of the SIGCHI conference on Human factors in computing systems, Florence, Italy. https://doi.org/10.1145/1357054.1357123

Buechley, L., (2006, October). A Construction Kit for Electronic Textiles. In Proceedings of the IEEE International Symposium on Wearable Computers (ISWC), Montreux, Switzerland. http://doi.org/10.1109/ISWC.2006.286348

Buechley, L., Hill, B. M., (2010, August). Lilypad in The Wild: How Hardware's Long Tail Is Supporting New Engineering and Design Communities. In Proceedings of the 8th ACM Conference on Designing Interactive Systems, New York, United States. https://doi.org/10.1145/1858171.1858206

Buechley, L. (2009). *LilyPad Arduino: How an Open Source Hardware Kit Is Sparking New Engineering And Design Communities*. Cambridge, MA: MIT Media Lab, High-Low Tech Group.

Buechley, L., & Eisenberg, M. (2008). The LilyPad Arduino: Toward Wearable Engineering for Everyone. *IEEE Computer Society, Pervasive Computing*, 7(2), 12–15. https://doi.org/10.1109/MPRV.2008.38

Buechley, L., & Eisenberg, M. (2009). Fabric PCBs, electronic sequins, and socket buttons: Techniques for e-Textile Craft, *Pers Ubiquit Comput*, 13, 133–150. https://doi.org/10.1007/s00779-007-0181-0

Sparkfun. (n.d. -a) *LilyPad ProtoSnap Plus Hookup Guide.* Retrieved from https://learn.sparkfun.com/tutorials/lilypad-protosnap-plus-hookup-guide/all

Sparkfun. (n.d. -b) *LilyPad Vibe Board Hookup Guide*. Retrieved from https://learn.

sparkfun.com/tutorials/lilypad-vibe-board-hookup-guide/all

Sparkfun. (n.d. -c) *LilyPad Tri-Color LED Hookup Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-tri-color-led-hookup-guide/all

Sparkfun. (n.d. -d) *LilyPad Buttons and Switches*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-buttons-and-switches/all

Sparkfun. (n.d. -e) *Powering LilyPad LED Projects*. Retrieved from https://learn.sparkfun.com/tutorials/powering-lilypad-led-projects/all

Sparkfun. (n.d. -f) *LilyPad Buzzer Hookup Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-buzzer-hookup-guide/all

Sparkfun. (n.d. -g) *LilyPad Light Sensor Hookup Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-light-sensor-hookup-guide/all

Sparkfun. (n.d. -h) *LilyPad Temperature Sensor Hookup Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-temperature-sensor-hookup-guide/all

Sparkfun. (n.d. -i) *Choosing a LilyPad Arduino for Your Project*. Retrieved from https://learn.sparkfun.com/tutorials/choosing-a-lilypad-arduino-for-your-project/all

Nayyar, A., Puri, V. (2016, March). A Review of Arduino Board's, Lilypad's & Arduino Shields, 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India.

Lakshmi Lasya, V. S., (2021). *Multifunctional Navigation Assistance Device for Visually Impaired Using Arduino Lilypad*, International Journal of Innovative Science and Research Technology, 6 (4), 569-572.

Sparkfun, (n.d. -j). *LilyPad Development Board Hookup Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-development-board-hookup-guide/all

Sparkfun. (n.d. -k) *LilyPad Development Board Activity Guide*. Retrieved from https://learn.sparkfun.com/tutorials/lilypad-development-board-activity-guide/before-you-begin

Galadima, A. A., (2014, December). *Arduino as a Learning Tool*, 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria. https://doi.org/10.1109/ICECCO.2014.6997577

Baurley, S. (2004). *Interactive and Experiential Design in Smart Textile Products and Applications*. Personal and Ubiquitous Computing, 8 (3-4), 274-281.

Berzowska, J., Bromley, M. (2007, April). *Soft Computation Through Conductive*

Textiles. In Proceedings of the International Foundation of Fashion Technology Institutes Conference, Toronto, Canada.

Mohsen, S., Zekry, A., Youssef, K., Abouelatta, M., (2021). *A Self-Powered Wearable Wireless Sensor System Powered by a Hybrid Energy Harvester for Healthcare Applications*. Wireless Pers Commun, 116, 3143–3164. https://doi.org/10.1007/s11277-020-07840-y

Ariyatum, B., Holland, R., Harrison, D., Kazi, T. (2005). *The Future Design Direction of Smart Clothing Development*. Journal of the Textile Institute, 96 (4), 199-210. https://doi.org/10.1533/joti.2004.0071

Smelik, A., Toussaint, L., & Dongen, P. V., (2016). *Solar fashion: An Embodied Approach to Wearable Technology*. International Journal of Fashion Studies, 3 (2), 287-303. https://doi.org/10.1386/infs.3.2.287_1

Lin, Y. (2014). *Open Source and Consumption*. In J. M. Ryan & D. Cook (Eds.), Encyclopedia of Consumption and Consumer Studies. The Wiley-Blackwell.

About the Authors

Hakan YUKSEL, PhD, Isparta, is an Assistant Professor of Computer Technologies at Isparta University of Applied Sciences. His main interests are in artificial intelligence technologies, development boards, sensor and 3D modeling applications.

E-mail: hakanyuksel@isparta.edu.tr, Orcid: 0000-0003-2186-533X

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 11%.

To Cite This Chapter:

Yuksel, H. (2021). The LilyPad Arduino. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 162–177). ISRES Publishing.

Field Programmable Gate Array

Muhammed Fahri UNLERSEN
Necmettin Erbakan University

What is an FPGA?

FPGA is a term formed by combining the first letters of the word Field-Programmable Gate Array. The reason for using the term "field programming" is that the function of the FPGA integrated circuit (IC) is not programmed at factory output and is an IC that can be changed while in the field. The function mentioned here is a task created with the hardware architecture of IC. It has grown very rapidly since the FPGA term was introduced. While growing at a high rate in terms of capacity and performance, the decrease in cost per unit operation has made FPGAs remarkable (Unlersen, 2015). In Figure 1, an FPGA IC is presented.

Figure 1. An FPGA IC belongs Xilinx Company

Although Xilinx presented the first hardware that can be called FPGA in 1984, the term FPGA became popular in 1988 with the company Actel. The non-recurring engineering cost required for application-specific integrated circuit (ASIC) fabrication does not exist in FPGAs. But, this situation made FPGAs advantageous only in the use of a low number of units. In this process, ASICs were more popular because they were very low cost compared to FPGAs in high production. However, according to Moore's law, the prediction that the number of units that FPGA will be advantageous will increase in the future has prevented the interest in FPGAs from decreasing. Today, performance, I/O capacity, power consumption, time to market and other capabilities are more important than device cost in FPGA-ASIC comparison (*FPGA Designs with VHDL Documentation*, n.d.; Trimberger, 2015; *What Is an FPGA? Field Programmable Gate Array*, n.d.).

Some of the application areas of FPGAs can be listed as follows (Rajewski, 2017):

- Aerospace
- Defense
- Automotive
- High Performance Computing and Data Storage
- Data Center
- Industrial
- ASIC Prototyping
- Broadcast
- Video and Image Processing
- Wired and Wireless Communications
- Medical Imaging
- Security

In the design of an embedded system, the question of which platform should be designed first comes to mind. Because for the designer, there are many different hardware such as microcontrollers, ASIC, microcomputer, FPGA. Actually, FPGA is not a one-to-one alternative to other microprocessor-built platforms. On an FPGA, a hardware to perform the required operation can be designed. However, in systems created with a microprocessor, commands that will perform a desired operation are executed on a fixed hardware. Additionally, it is also possible to design a microprocessor with an FPGA.

The designer's choice of FPGA among these alternatives depends on the needs of the system to be designed rather than a matter of whim. For example, in a hardware where the algorithm to be used will change frequently and operations such as multiplication and division with complex numbers will be made frequently, using a DSP produced for this purpose may be more logical than using an FPGA. Because it will be very simple and flexible to make calculations on this DSP using a high-level language such as C. In a platform that should be cheap rather than high performance, choosing a microcontroller can be a fast, simple and satisfying solution. However, if the process requires high performance and speed, then FPGA will be more suitable for this type of applications

(Leong et al., 1998; Rajewski, 2017).

FPGA Structure

FPGAs are semi-ready silicon devices that can be electrically programmed to be part of a digital circuit or system. Its structure can be defined in three main parts: programmable logic blocks, input and output blocks surrounding this block array, and interconnections (Chu, 2008; Gunes & Ors, n.d.).

The basic FPGA structure consists of thousands of basic elements called Configurable Logic Blocks (CLB). These basic structures can be called Logic Blocks (LB), Logic Elements (LE) or Logic Cells (LC) according to the manufacturer (Gunes & Ors, n.d.). CLBs are formed by combining a set of logic elements such as a LookUp Table (LUT) and flip flops (FF). The hardware architecture of the FPGA consists of the data stored in these LUTs. The working principle of a 4-input LUT is illustrated in Figure 2.

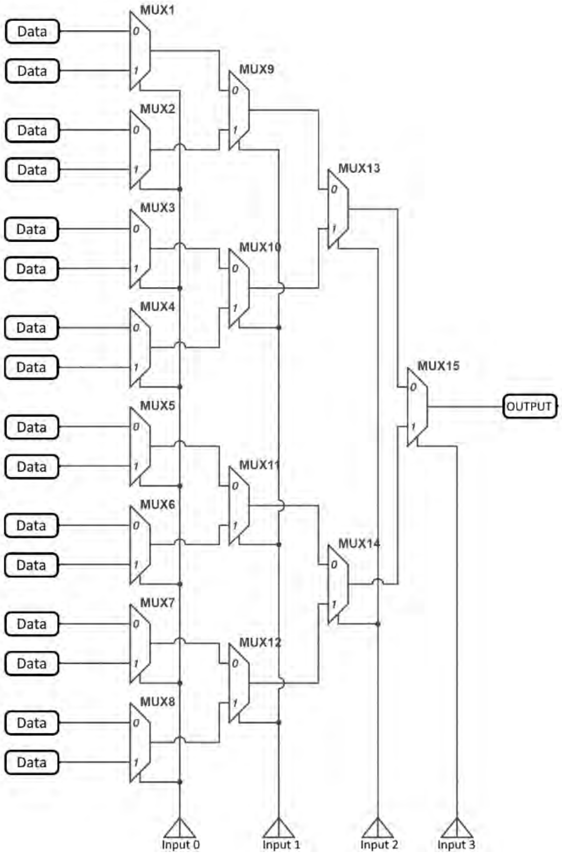


Figure 2. A 4-Input LUT Structure

The values specified here, as Data are the data loaded during FPGA programming. According to this loaded data, the value applied to the inputs is selected and transferred to the output. Thus, this LUT fulfills its special mission.

Although the LUT and the used logic elements that make up the CLB differ from company to company, an example CLB structure is presented in Figure 3.

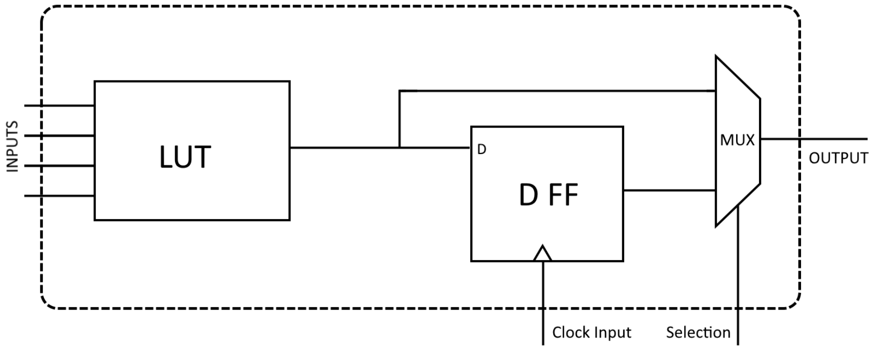


Figure 3. An example CLB Structure

In the CLB shown, a LUT output is selected with a multiplexer according to synchronous or asynchronous structure, while a D-type FF is activated in synchronous use.

The term programmability in FPGAs means that LUT information and other logic elements can be controlled according to the designed system after production. The programming mentioned here is not the commands operated by a hardware as in microcontrollers, but the codes that define the hardware.

There are interconnections around CLBs. Interconnects allow these blocks to be programmed and communicated with other blocks (Chu, 2008).

The communication of FPGAs with the outside world is provided by the input and output (I/O) blocks. These blocks can be configured in different directions as input or output. Today, the I/O blocks of FPGAs support 500MHz operating frequency. Also, some of the I/O blocks have the ability to read data on both falling and rising edges (*What Is an FPGA? Field Programmable Gate Array*, n.d.; *What Is an FPGA? Programming and FPGA Basics - INTEL*, n.d.).

Pins on FPGA IC are divided into 2 categories (Chu, 2008; Gunes & Ors, n.d.; Rajewski, 2017).

- Dedicated pins: Pins with special tasks on the FPGA are called dedicated pins. They are divided into three groups according to their functions.
    - Power pins: They provide the power needed for the IC to work.
    - Configuration pins: They are used to download the program created in the PC software to the IC.
    - Clock pins: These are the pins specially set to receive the simultaneous clock signals in IC.
- User pins: These are the standard pins that can be set as Input, Output or Input-

Output. Each pin has its own I/O cell. This cell determines in which mode a pin will work.

Synchronous design is an important area in logic circuit designs. Such designs are clock-based. This is also true for FPGA. However, the most important issue in synchronous design is that the clock signal reaches all logic elements simultaneously. Otherwise, timing problems and even electrical problems will arise. FPGA manufacturers have overcome such problems with a connection structure called Global Routing or Global Line for clock signals. Thanks to these connections, the clock signal reaches all CLBs simultaneously. For this reason, clock feeds must be made from the pins of the FPGA that are reserved as clock (Chu, 2008; Gunes & Ors, n.d.; Rajewski, 2017; Unlersen et al., 2018).

There are PLL blocks that can generate the high frequencies which are needed internally within the FPGA. These blocks can generate the necessary operating frequencies up to 500MHz, taking the frequencies from the clock input of the FPGA as reference, usually around 50MHz (Unlersen, 2015).

FPGAs contain RAM units in blocks. These memories are used for data storage processes that CLBs will need during their operations. These RAMs support both single access and multiple access. Multiple applications accessing the same RAM at the same time is called multiple access. While these block RAMs meet the large memory needs of CLBs, there are distributed RAMs interspersed around the CBLs for small memory needs (Chu, 2008; Gunes & Ors, n.d.; Rajewski, 2017).

Programming FPGAs

VHDL and Verilog are used in FPGA programming. The prepared program is loaded onto the FPGA IC with the Joint Test Action Group (JTAG) protocol. JTAG is an IEEE Standard 1149.1-1990 that was created in the 1980s to eliminate errors in the production of electronic cards (IEEE Standards Board. & IEEE Computer Society. Test Technology Technical Committee., 1993). Most FPGAs do not have an internal EEPROM. The loaded program is stored in SDRAM cells. In other words, the program loaded with JTAG is not permanent. Therefore, the program must be reloaded each time the FPGA is re-energized. For this reason, they are designed with an external EEPROM right next to the point where the FPGA ICs are located (*What Is an FPGA? Field Programmable Gate Array*, n.d.; *What Is an FPGA? Programming and FPGA Basics - INTEL*, n.d.).

VHDL - Very High-Speed Integrated Circuit Hardware Description Language

Hardware description languages are used in FPGA programming. These are VHDL and Verilog. Verilog uses a textual format to describe electronic systems. In the field of electronic design, Verilog can be used for verification through simulation for testability analysis, error grading, logic synthesis and timing analysis. Verilog has the IEEE 1364 standard. Design is performed with fewer commands. In terms of structure, it is a language that is often compared to C. However, Verilog is not as wordy as VHDL due to its nature. Therefore, VHDL is more capable of creating hierarchical structures (Gunes & Ors, n.d.; Nageswaran, 1997).

Details of VHDL will be given here. VHDL is a widely used hardware description language for designing and testing digital circuits. VHDL stands for Very high speed integrated circuit Hardware Description Language (Unlersen, 2015).

The most important feature of VHDL is that designs can be divided into components in a hierarchical way. Each design element should have a well-defined interface. There must be faultless behavior design in architecture. VHDL supports synchronous and asynchronous circuit design. The time behavior of functions can be observed by simulation. It thus allows the behavior of the underlying system to be verified and modeled before the design is translated into actual gates and cables ("IEEE Standard for VHDL Language Reference Manual," 2019; *VHDL Tutorial: Learn by Example*, n.d.; Nageswaran, 1997). In addition, programs prepared in VHDL are portable structures. A component prepared for a previous project can be integrated into subsequent projects (Baker, n.d.; Pak, n.d.).

There are many advantages of using VHDL (*VHDL Mini-Reference*, n.d.).

- Has independent design definitions
- Applicable to many manufacturer ICs
- The design can be updated when necessary
- Allows a standard documentation
- It shortens the design process
- Accelerates the commercialization of design
- Reduces research and development costs
- Increases final product quality
- Enables detailed control of its functions
- Re-use of previous designs as components

A program prepared with VHDL consists of three basic parts;

1. Entity
2. Architecture
3. Procedure

The input and output pins of the entity to be designed are defined in the Entity section. An example for “OR GATE” of Entity is given below.

```
entity OR_GATE is                -- Comment line
 generic(
           data_width: integer :=4;
           delay_time: time := 10ns
        );
 port   (
           D : in std_logic_vector( data_width-1 downto 0);
           Q : out std_logic
        );
 signal ln01 : std_logic;
end OR_GATE;
```

where OR_GATE is the project name. The name of the vhd file with all the codes must be the same as the project name. Expressions defined in the generic field are used to define presets that can be set without changing the code structure. Here, the width of inputs the OR gate will have is defined as generic. Here, two expressions are defined inside the generic(…) expression. There should be no markup at the end of the last statement. Generic values can be updated as needed during synthesis and simulation. But the hardware is not hot-swappable (Gunes & Ors, n.d.; “IEEE Standard for VHDL Language Reference Manual,” 2019; Nageswaran, 1997).

In the Port section, there are pin definitions that the designed hardware will use when communicating with the outside world. Here, there is a lot of information such as port names, port direction, data width, data type. All items must be separated with a semicolon. There is no sign at the end of the last item (Baker, n.d.; *VHDL Tutorial: Learn by Example*, n.d.).

In this section, after the port definition, signal definition can be made if needed. Detailed information about this will be given in the definitions section.

The “- -” (double dash) symbol in the first line is used to add a comment line. This sign is like // in C language, % in Matlab, or # in Python. Expressions after the double dash are ignored by the compiler.

It is the unit in which the type and direction of the input and output pins of the Entity design are defined in detail. However, it does not contain any information about the functions of the design. Input-output definitions and directions (in, out, inout, buffer) are made in this section (Baker, n.d.; "IEEE Standard for VHDL Language Reference Manual," 2019).

The descriptions of pin directions used here are as follows (Pak, n.d.);

- In: It is a read-only structure used for input pins.
- Out: It is the structure used for the output pins and can only be given a value but cannot be read.
- Buffer: It is the only driver accepting structure that can be read and written for bidirectional pins.
- Inout: It is a pin structure that can have more than one driver and can both read and write values.

A procedure is a construct used within the designed entity to avoid duplicating an operation that is often repeated. It is not an indispensable structure. A procedure that increments the value in a variable by one is presented below.

```
procedure one_incrementer (variable vr: inout int8) is
begin
    if (vr <= MAKSIMUM) then
        vr := vr + 1;
    end if;
end
```

Architecture is the area where the structure of the designed asset is determined. The architecture required for the OR_GATE design defined in the Entity section is shown below.

```
architecture behaviour of OR_GATE is
begin

    Q <= D(0) or D(1) or D(2) or D(3);

end behaviour;
```

The term "behaviour" as used here is an arbitrary term for the programmer. However, special terms of the programming language cannot be used here.

The function of the designed system is defined in Architecture. This design can be done in three different ways (Gunes & Ors, n.d.; Pak, n.d.). They are:

- Behavioral
- Data Flow
- Structural

Let’s examine these identification forms with a full adder design.

First, the entity required for the 1-bit full adder design must be defined. The entity required for this structure is given below.

```
entity FULL_ADDER is
port    (
         A : in std_logic;
         B : in std_logic;
         Carry_in : in std_logic;
         Q : out std_logic;
         Carry_out : out std_logic
         );
end FULL_ADDER;
```

Behavioral Style

Here, A, and B are the two numbers to add, while Carry_in is the carry input. The truth table for this full adder will be as in Table 1.

Table 1. Truth Table for Full Adder

A	B	Carry_in	Carry_out	Q
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

In line with this table, a code can be created for the behavioral style as follows.

```
architecture Behaviour of FULL_ADDER is
begin
    process (A, B, Carry_in)
    begin
        if (A='0' and B='0' and Carry_in='1') or
                (A='0' and B='1' and Carry_in='0') or
                (A='1' and B='0' and Carry_in='0') then
            Q <= '1';
            Carry_out <= '0';

        elsif (A='0' and B='0' and Carry_in='0') then
            Q <= '0';
            Carry_out <= '0';

        elsif (A='0' and B='1' and Carry_in='1') or
                (A='1' and B='0' and Carry_in='1') or
                (A='1' and B='1' and Carry_in='0') then
            Q <= '0';
            Carry_out <= '1';
        else
            Q <= '1';
            Carry_out <= '1';
        end if;
    end process;
end Behaviour;
```

The process structure and if, elsif, else structure used here will be mentioned in the following stages. Behavioral style is designed using processes. In this style, typing is performed in a program-like manner. However, it is not clear what kind of formulation the operations to be performed have. It’s just like constructing a truth table with conditions (Gunes & Ors, n.d.; Pak, n.d.).

Data Flow Style

In the data flow style, the operations are presented more clearly. Results are obtained by arithmetic and/or logic operations with control signals and data. In order for the full adder to be designed in a data flow architecture, logic operations should be introduced by passing from the truth table to the Karnaugh diagram. Karnaugh diagrams and logic formula of the truth table given above are given in Table 2.

Table 2. Karnaugh Diagrams for Q and Carry out in Full Adder Design

Q B C

		00	01	11	10
A	0	0	1	0	1
	1	1	0	1	0

Q = (A xor B) xor C

Carry_out B C

		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

Carry_out = (A and B) or (A and C) or (B and C)

The codes in the data flow style created according to the obtained formulas of the full adder circuit are presented below.

```
architecture BEH of FULL_ADDER is
    signal L : std_logic;
begin
  Carry_out<=(A and B) or (A and Carry_in) or (B and Carry_in);
  L <= A xor B;
  Q <= L xor Carry_in;
end BEH;
```

Structural Style

It is formed as a result of the organization of sub-modules working simultaneously in a structural style. For the full adder, this can be achieved by combining the half adders. For this reason, a half adder structure is needed first for this example. First, let's examine the half adder structure.

```
entity HALF_ADDER is
port   (
        A : in std_logic;
        B : in std_logic;
        Carry_out : out std_logic;
        Q : out std_logic
        );
end HALF_ADDER;

architecture DATAFLOW of HALF_ADDER is
begin
    Q <= A xor B;
    Carry_out <= A and B;
end;
```

As can be seen, A and B are defined as input, Carry_out and Q are defined as output. These values are simply obtained with an EXCLUSIVE-OR and an AND operation.

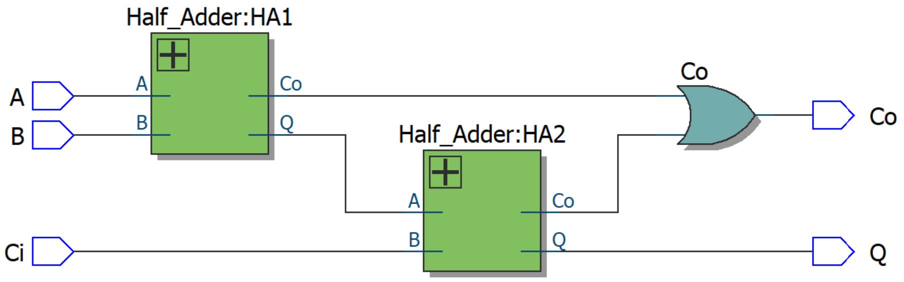


Figure 4. Full Adder Construction with Half Adders

Using half adders, it is possible to obtain a full adder with the scheme given in Figure 4. Defining this schema in architecture is what we call structural style. The architecture created with the definition of this schema is shown below.

```
architecture STRUCTURE of FULL_ADDER is

    component HALF_ADDER
        port (
            In1, In2 : in std_logic;
            Out1, Out2 : out std_logic
        );
    end component;

    signal line1, line2, line3 : std_logic;
begin

    HA1 : HALF_ADDER port map (A, B , line1, line2);
    HA2 : HALF_ADDER port map (line2, Carry_in, line3, Q);
    Carry_out <= line1 or line3;

end STRUCTURE;
```

Components are one of the most important elements of VHDL. Thanks to its component definition feature, VHDL offers the opportunity to include previous studies into new studies. A defined component can be used repeatedly within the architecture. The line1, line2, line3 specified here is a signal object. It can also be understood from the code that these provide the electrical connection between two points (Gunes & Ors, n.d.; Pak, n.d.).

Data Objects

The data objects used in VHDL are:

- Signal
- Constant
- Variable

Signals, the most important data object, provide connections within the design. Constant and Variable are less frequently used data objects.

There are a number of standard details to consider when specifying names for data objects.

- When creating names for data objects, characters, numbers, and the underscore are permitted for use.
- Characters not used in English cannot be used.

- Keywords used in VHDL cannot be specified as data object name.
- The first character of the data object must be a letter.
- The last character of the data object cannot be an underscore.

Assigning a Value to a Data Object

While assigning a single-bit value to the signal data object, it is written as 1 or 0 in single quotes. But for assigning a multiple bit value, it is written as a sequence of 1 and 0 in double quotes. For example, if beep is a 1 bit signal and dt is a 4 bit signal:

```
beep <= '0';
dt <= "1011";
```

The signal object is defined as shown below.

```
SIGNAL data_obj_name : type [:= default_value];

signal line1, line2, line3 : std_logic;
Signal Beep : Bit := '0';
Signal data : integer;
```

In this definition, it is not obligatory to include the expression given in square brackets. Sample signal definitions are given above.

Constants are expressions that are initially assigned a value and cannot be changed afterwards. The definition and examples are given below.

```
CONSTANT data_obj_name : type := value;

Constant Yes : Boolean := true;
Constant No : Boolean := false;
```

Variable stores temporary information. They are expressed in the process and subprogram, and the information they store is accessible within the respective process. The definition of a variable is given below. It is not mandatory to use the part in square brackets.

```
VARIABLE data_obj_name : type [:= default_value];

VARIABLE X,Y : std_logic_vector(7 downto 0);
variable a : integer := 0;
```

Data Types

Data objects are always defined using a data type. The data types that can be synthesized are as follows.

- BIT
- BIT_VECTOR
- STD_LOGIC
- STD_LOGIC_VECTOR
- SIGNED
- UNSIGNED
- INTEGER
- ENUMERATION
- BOOLEAN

BIT and BIT_VECTOR

BIT defines a single bit object. This object can take a value of ‘0’ or ‘1’. BIT_VECTOR is used to define the BIT type object with the specified width.

```
SIGNAL x1 : BIT;
SIGNAL C : BIT_VECTOR (1 to 4);
SIGNAL VR1 : BIT_VECTOR (7 downto 0);
SIGNAL VR2 : BIT_VECTOR (0 to 7);
```

As defined above, x1 is a single-bit object while C is a 4-bit object. VR1 and VR2 are both 8-bit objects. The difference between these two is in the order of the bits. In VR1, the most significant bit is in the 7 indexed bit and the least significant bit is in the 0 indexed bit. In VR2, on the other hand, the most significant bit is in the 0 indexed bit, while the least significant bit is in the 7 indexed bit. This can be seen in the data assignment and results below.

```
VR1 <= "10110110";                VR2 <= "10110110";
    -- VR1(0) -> 0                    -- VR2(0) -> 1
    -- VR1(1) -> 1                    -- VR2(1) -> 0
    -- VR1(2) -> 1                    -- VR2(2) -> 1
    -- VR1(3) -> 0                    -- VR2(3) -> 1
    -- VR1(4) -> 1                    -- VR2(4) -> 0
    -- VR1(5) -> 1                    -- VR2(5) -> 1
    -- VR1(6) -> 0                    -- VR2(6) -> 1
    -- VR1(7) -> 1                    -- VR2(7) -> 0
```

STD_LOGIC and STD_LOGIC_VECTOR

STD_LOGIC type, which has a more flexible structure than BIT type, can take Z, 0, 1, L, H, U, -, X and W values. The explanations of these values are shown in Table 3 ("IEEE Standard for VHDL Language Reference Manual," 2019; *VHDL Tutorial: Learn by Example*, n.d.; Nageswaran, 1997). Objects with this data type can perform AND, NAND, OR, NOR, XOR, XNOR, NOT logic operations. As with the BIT structure, adding _VECTOR defines the width of the object. To use it, it is necessary to install the std_logic_1164 package in the ieee library.

Table 3. Possible Values of STD_LOGIC type

Value	Explanation
0	Logic 0
1	Logic 1
Z	High Impedance
W	Weak signal (unable to say 0 or 1)
L	Weak 0
H	Weak 1
-	Don't Care
U	Uninitialized
X	Unknown

```
VARIABLE x : std_logic_vector (7 downto 0);

VARIABLE btn: std_logic;
```

Here, the variable x is 8 bits wide, while the btn variable is 1 bit wide.

Signed - Unsigned

It is a data type used for signed and unsigned data. In addition to the ability to access the value of each bit separately, such as std_logic_vector, it also supports a number of operations such as arithmetic (+, - *), comparison and shifting. To use it, the std_logic_arith package from the ieee library must be installed.

Objects of UNSIGNED data type always store positive values. The value is defined by all bits. For example, an 8-bit data object stores values between 0 and 255.

Objects of SIGNED data type can have positive or negative values. The sign of the data object is determined by the most significant bit. If the most significant bit is 0, it is positive. Contrarily, if the most significant bit is 1, it is negative. The absolute value (magnitude) of a negative value is found by inverting the remaining bits and adding 1.

```
SIGNAL in1: unsigned (3 downto 0);
SIGNAL in2: signed (7 downto 0);
```

Integer

Generally, it can take values in the range of -2^{31} to 2^{31} -1. Depending on the user, this range of values can be changed. If a value outside the definition range is given, an error will occur. It can store negative values. In addition to standard operations such as addition, subtraction, multiplication, division, and modulus, they can also be used in comparison operations such as greater than, greater than or equal, less than, less than or equal, equal, and unequal.

```
VARIABLE a : integer;
VARIABLE b : integer range -100 to 200;
```

The variable a, defined here is defined to be used in the natural range of the integer. B, on the other hand, is set to store values in the range of minimum -100 and maximum 200.

Boolean

This data type has two values as TRUE and FALSE.

```
Constant Yes : Boolean := true;
Constant No : Boolean := false;
```

Enumeration

It is a data type whose values can be defined by the programmer. In order to use this data type, the data type must be defined first. All values to be used in this type must be given during type definition. For example, let’s define a color variable.

```
TYPE color is (red, orange, yellow, blue, black, white);
SIGNAL pxl : color;
```

Here, the pxl variable cannot take a value other than the defined colors. While the `pxl <= blue;` operation is a valid assignment, the `pxl <= green;` operation is an invalid assignment. Because the color green is not included in the type definition.

Operators

The operators used in VHDL are grouped and presented in Table 4.

Table 4. Operators Used in VHDL

	Symbol of Operator	
Miscellaneous	MOD	Modulus
	ABS	Absolute value
	**	Power
Multiplication, Division	*	Multiplication
	/	Division
Addition, Subtraction	+	Addition
	-	Subtraction
Sign	+	Positive
	-	Negative
Logic	NOT	Not
	AND	And
	NAND	Not and
	OR	Or
	NOR	Not or
	XOR	Exclusive or
	XNOR	Not exclusive or

Sequential Operations

The simultaneous assignment statements in the architectural design have no priority or order. Changing their order does not affect the function of the final structure. However, VHDL also provides another type of statement, called sequential statements. These are statements such as if statement, case statement, loop statements. The order of these will affect its function. So the order is important. Separation of concurrent statements that do not change the result of their sequencing and sequential statements whose ordering is important is provided by the PROCESS structure (Gunes & Ors, n.d.; “IEEE Standard for VHDL Language Reference Manual,” 2019; *VHDL Mini-Reference*, n.d.).

The PROCESS statement is included in the architecture. The variable type is defined in PROCESS as shown. Data in a variable defined in a PROCESS can only be exported by transferring it to another signal type object.

```
[Name :] PROCESS (Sensitivity List)
 VARIABLE var_name : var_type [range . to .][:=default_value];
begin
                -- codes
    ....        -- codes
end process [Name];
```

The expressions in square brackets specified here are not mandatory. The expression indicated by Name is used to define PROCESS. It is not a mandatory statement, but naming is necessary to create a regular code. The sensitivity list is important. Here, the signals to be used in PROCESS are listed. Thus, it is defined that this PROCESS is affected by the change of signals in the specified list. If there is a variable to be used in the sequential code, its definition should be added to the presented location as VARIABLE. The scope of the defined variable is limited in PROCESS.

The most frequently used expression in the process is the IF structure. The IF structure is a structure used when the activation or deactivation of some codes depends on certain conditions. Here is how the IF structure is in VHDL.

```
If comparison then        -- mandatory line
    VHDL statements;
Elsif comparison then     -- not mandatory, in case of need
    VHDL statements;
Else                      -- not mandatory, in case of need
    VHDL statements;
End if;                   -- mandatory line
```

An example code is shown below. With this code, x1 or x2 is transferred to the f signal according to the state of the Sel signal. In addition, the structure of the process and the sensitivity list can be seen. As can be understood, a multiplexer structure is presented here.

```
PROCESS (Sel, x1, x2)
BEGIN
    IF Sel = '0' THEN
        f <= x1;
    ELSE
        f <= x2;
    END IF;
END PROCESS;
```

From the expression given here, it is not clear how many bits wide or type x1, x2, and f are. However, all 3 signals must be of the same type and width.

The case statement can be viewed as an alternative to the nested if (elsif) statement. The fact that VHDL is a fully defined language shows itself here. Here, all states for the signal (or variable) that is the input of the case structure should be given in options. Otherwise, synthesis will not be possible. In order to provide this situation easily, there is an option called **others** in VHDL. This is a term used to denote all alternatives other than those defined. The Case structure used in VHDL is given below.

```
case statement is
    when value_1 =>
        VHDL statement;
    when value_2 =>
        VHDL statement;
    when value_3 =>
        VHDL statement;
    when others =>
        VHDL statement;
end case;
```

Here, if the expression given between the “case” and “is” is a value other than value_1, value_2 and value_3, which is compared in case options, the “when others” tab will be active. The “when others” statement is not needed if all possible values are defined. Below is a multiplexer structure with 2 bit select inputs.

```
case Sel is
    when "00" =>
        f <= x0;
    when "01" =>
        f <= x1;
    when "10" =>
        f <= x2;
    when "11" =>
        f <= x3;
end case;
```

The “When others” are omitted as all possible states are mentioned here.

When we talk about loops in VHDL, we encounter for and while loops. A for loop is a loop used to execute a specific group of commands a specified times. The structure of the “for loop” is given below.

```
[Name :] for arbitrary name in start to stop loop
    VHDL statements;
end loop [Name];
```

For example, the design of a structure that finds the number of bits that are 1 in an A signal coming from outside is given below.

```
process(A)  -- Defination of Process
variable V1:integer range 0 to 7;
begin
    V1:=0;
    for i in 0 to A'length-1 loop
        if ( A(i) = '1' ) then
            V1 := V1 + 1;
        end if;
    end loop;
    X <= V1;
end process;
```

Another loop structure in VHDL is the while loop. The general structure of the while loop is given below.

```
[Name :] while Comparison Statement loop

    VHDL ifadeleri;

end loop [Name];
```

The design of a structure realized with a while loop that finds the number of bits that are 1 in an incoming A signal is given below.

```
process(A)
 variable V1 : integer range 0 to 7;
 variable i : integer := 0;
begin
    V1 := 0;
    while i < A'length loop
        if ( A(i) = '1' ) then
            V1 := V1 + 1;
        end if;
        i := i + 1;
    end loop;
    X <= V1;
end process;
```

The necessity of performing synchronized operations with the clock pulse in VHDL is a very common situation. In these cases, the detection of the pulse moment of the clock signal (as a rising edge or falling edge) becomes important. This process can be defined in VHDL with PROCESS and is controlled by the ‘EVENT statement. Below are two different examples that increase the value of the counter signal by one on the rising edge and by the other one the falling edge. The clk used here is a signal received from the global clock line, and the counter is an integer type signal that can be mathematically processed.

The counter design triggered on the rising edge is as follows.

```
PROCESS (clk)
Begin
    If (clk'EVENT and clk='1') then
        counter <= counter + 1;
    End if;
End process;
```

The counter design triggered on the falling edge is as follows.

```
PROCESS (clk)
Begin
    If (clk'EVENT and clk='0') then
        counter <= counter + 1;
    End if;
End process;
```

The only difference between the two processes is that the clk in the second comparison in the if statement is equal to 1 in one and 0 in the other. Because the ‘EVENT statement is triggered on both rising edge and falling edge. However, the value of the triggered signal determines which edge it is. If there has been a trigger and the signal value is 1, it means a rising edge change. If there has been a trigger and the signal value is 0, it means a falling edge change.

An Example of VHDL Application

Let’s make an example application using what we have mentioned so far. For example, let’s design one 32kHz and one 1MHz signal generator using the 50 MHz clock signal. Let’s do this on a physical card. The card we will use in this example is the Altera DE2-115 FPGA board. This board has slide switches, buttons, leds, 7 segment displays, 2-line LCD screen, SDRAM, SRAM, FLASH memory, 50MHz oscillator, DAC and VGA output ports and many ports and hardware. There is a Cyclone IV FPGA IC on this board. This is a development board and the datasheet details which pin of the FPGA IC is connected to which hardware. The picture of the development board is shown in Figure 5.

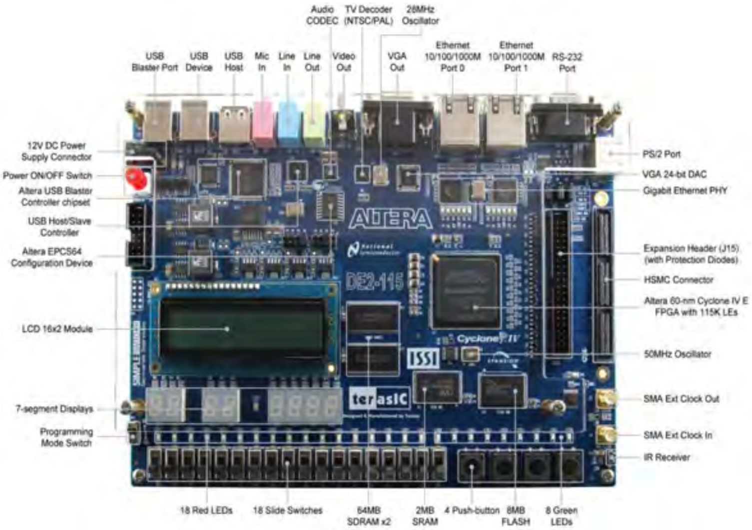


Figure 5. Altera Cyclone IV EP4CE115F29C7 DE2-115 Development Board

This board contains the JTAG programming structure on its own. It is programmed by connecting to a PC via USB with Quartus software web edition, which Altera company distributes free of charge. It is possible to write code in both VHDL and Verilog hardware description languages with Quartus software. Here, after installing the Quartus software, it will be explained step by step how to create a project, how to simulate it and how to download to the FPGA IC. The Quartus software, whose images are given here, is v13.1 version.

The screen shown in Figure 6 appears first. On this screen, there is the Project Navigator section in the upper left corner. Here is the information of the project being worked on. Under the Project Navigator section, items that we may want to see related to the project such as Hierarchy, files, design units etc. are listed in tabs. On this screen, creating a new project starts by clicking New Project Wizard or clicking New Project Wizard from the File menu.

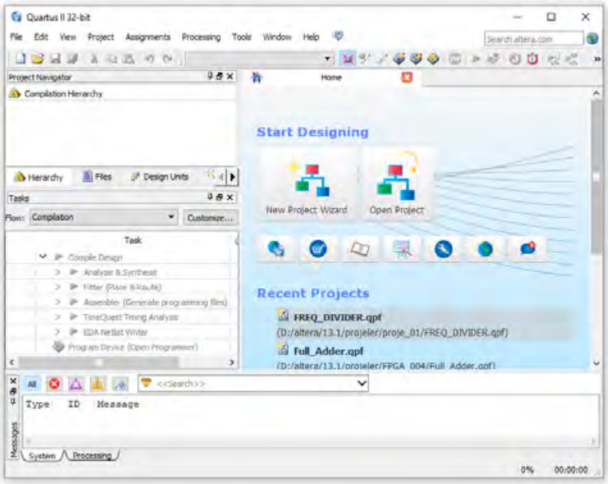
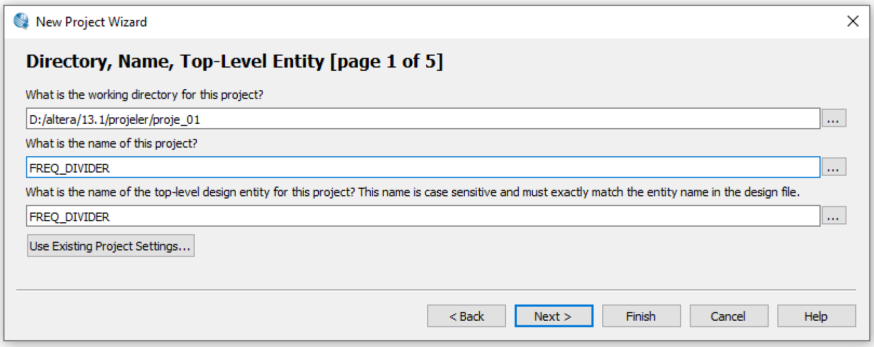


Figure 6. Quartus Software Welcome Interface

After clicking on the New Project Wizard, the Introduction page first appears. We proceed on this page with the Next button. On the next page, Directory, Name, Top-Level Entity information should be entered. This page view is given in Figure 7. Here, in the first line, the folder where our project will be saved should be selected. The second line contains the project name. The name we will give here is automatically transferred to the third line. Let's give FREQ_DIVIDER as the name of this project. Let's press the Next button to move to the next page.



Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?
D:/altera/13.1/projeler/proje_01

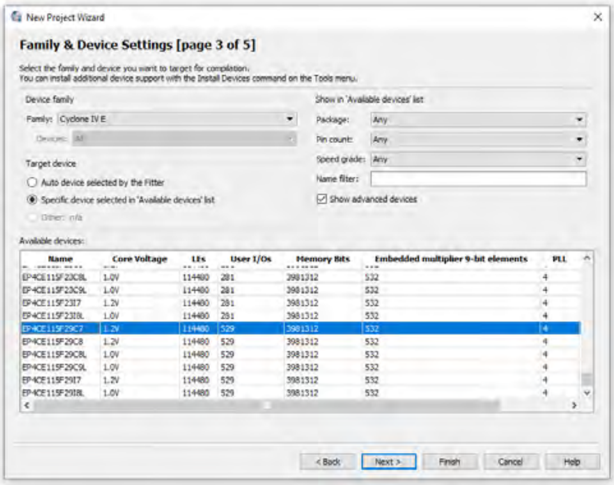
What is the name of this project?
FREQ_DIVIDER

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
FREQ_DIVIDER

Figure 7. New Project Creation Steps; Folder and Project Name Entry

On the next page, if you have VHD files that you have prepared before and need to use in the new project, a page will open for you to add them. Since there is no such need for this project, we can proceed with the Next button without taking any action.

Then, on the page shown in Figure 8, it will ask you to specify the FPGA IC you will use. There is Cyclone IV EP4CE115F29C7 IC on the development board we will use in this project. This device must be selected from the list.



Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family
Family: Cyclone IV E
Devices: All

Target device
○ Auto device selected by the Fitter
● Specific device selected in 'Available devices' list
○ Other: n/a

Show in 'Available devices' list
Package: Any
Pin count: Any
Speed grade: Any
Name filter:
☑ Show advanced devices

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL
EP4CE115F23C8L	1.0V	114480	281	3981312	532	4
EP4CE115F23C9L	1.0V	114480	281	3981312	532	4
EP4CE115F23I7	1.2V	114480	281	3981312	532	4
EP4CE115F23I8L	1.0V	114480	281	3981312	532	4
EP4CE115F29C7	1.2V	114480	529	3981312	532	4
EP4CE115F29C8	1.2V	114480	529	3981312	532	4
EP4CE115F29C8L	1.0V	114480	529	3981312	532	4
EP4CE115F29C9L	1.0V	114480	529	3981312	532	4
EP4CE115F29I7	1.2V	114480	529	3981312	532	4
EP4CE115F29I8L	1.0V	114480	529	3981312	532	4

Figure 8. Determining the Family and Name of the FPGA IC

Since we will not make any changes on the next pages, we can complete the process by clicking Finish after this point. On the page that opens, the name of the FPGA IC used in the project and the name of the project can now be seen in Project Navigator.

For the operations to be done in this project, we need to add a new VHDL file. For this, clicking New from the File menu will open a list containing many file types. Select VHDL File from this list and click OK.

In the editor page that opens, we can start writing the codes of the frequency divider. Here, we need to write the *library ieee* command to indicate that we will use the ieee library first. Later, *the std_logic_1164* and *std_logic_arith* packages that we will use in this project are added.

After this point, the Entity should be created. In this project, 1 input pin for 50MHz input, 1 input pin for reset and 2 output pins for 32kHz and 1MHz outputs are required.

```
Library ieee;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;

Entity FREQ_DIVIDER is
    Port(
            clk     :in std_logic;
            rst     :in std_logic;
            Q_32kHz :out std_logic;
            Q_1MHz  :out std_logic
    );
    signal cntr_32k : unsigned (9 downto 0) := (others=>'0');
    signal cntr_1M  : unsigned (4 downto 0) := (others=>'0');
    signal sclk_32k :std_logic :='0';
    signal sclk_1M  :std_logic :='0';
end FREQ_DIVIDER;
```

After this point, the architecture should be designed. The basic logic here would be: There is 20ns between two rising pulses of the 50MHz signal. Since 1MHz is 0 for 500ns and 1 for 500ns, 1MHz signal output should change its level for every 25 pulses of 50MHz signal. The 0 and 1 periods of the 32kHz signal are 15.63µs. Approximately 781 pulses (781x20ns =15.62µs and f=32.01kHz) are needed to obtain this period. A 5-bit unsigned counter is needed to count 25 pulses, and a 10-bit unsigned counter is needed to count 781 pulses.

The logic in this architecture is to create 2 different counters. When these counters reach the specified numbers, they should toggle their output and reset their own counter. The created entity and architecture are as follows.

```
Architecture Beh of FREQ_DIVIDER is
Begin
    process (clk,rst)
    Begin
        if (rst='1') then
            cntr_32k <= (others=>'0');
            cntr_1M <= (others=>'0');
            sclk_1M <='0';
            sclk_32k <='0';
        elsif ( clk'Event AND clk='1' ) then
            cntr_32k <= cntr_32k + 1;
            cntr_1M <= cntr_1M + 1;
            if (cntr_1M >= 25) then
                sclk_1M <= not sclk_1M;
                cntr_1M <= (others=>'0');
            end if;
            if (cntr_32k >= 781) then
                sclk_32k <= not sclk_32k;
                cntr_32k <= (others=>'0');
            end if;
        end if;
    end process;
    Q_32kHz<=sclk_32k;
    Q_1MHz<=sclk_1M;
end Beh;
```

In this architecture, the frequencies created such as sclk_32k and sclk_1M are first transferred to an object of signal type and from there to the output. This is because the final value is inverted after a certain number of clock pulses. In order to perform an inversion operation, it is necessary to read the value of the signal first. However, since it cannot be read from the pins defined as out, firstly, the data is transferred to the signal object and then to the output.

The created file can be saved with the CTRL+S key combination. The project name automatically appears as the file name during recording. Here it is necessary to save without modification.

After saving, the project should be compiled. For this, click Start Compilation from the Processing menu. Compilation stages can be followed in the Task menu under Project Navigator on the left. The entire build process should be completed in green. Red errors can be observed in the warning messages pane at the bottom. The errors specified in this menu should be corrected and recompiled.

After a successful compilation, the simulation can be started. To start the simulation, select Run Simulation Tool from the Tools menu and RTL Simulation from the drop-down menu when hovering over it. Then Quartus opens ModelSim software and adds a library called work. On the page that opens, the library is located on the far left. If the mentioned menus are not visible, it is possible to open them from the Window menu of ModelSim.

When the Work in the library is expanded, the project name freq_divider appears under it. Double-clicking on the Freq_divider displays all signals defined in the project in the "Object list". These signals (clk, rst, Q_32kz and Q_1MHz) are dragged and dropped into Wave windows, from which we will intervene and observe.

Right click clk in the Wave window and select Clock. Since the period is in picoseconds in the window that opens, enter 20000 and press the OK button.

Right click rst in the Wave window and click Force. The value of U is set to 0 in the window that opens.

After this point, we need to determine how long the simulation should be run. Since the longest period belonging to the 32kHz signal is 31.25µs, simulation duration can be 50µs . For this, 50us is written in the upper middle area of ModelSim that is 100ps on the start up. Instead of the symbol for micro µ, the u letter is used. Simulation is performed by clicking "Run 100 F9" in the Run tab from the Simulate menu. Right click on the simulation screen and select Zoom Full. The simulation result will appear as shown in Figure 9.

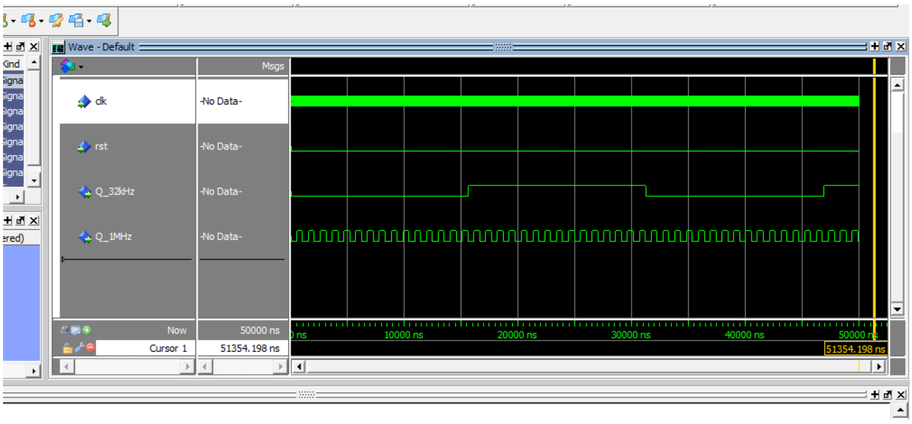
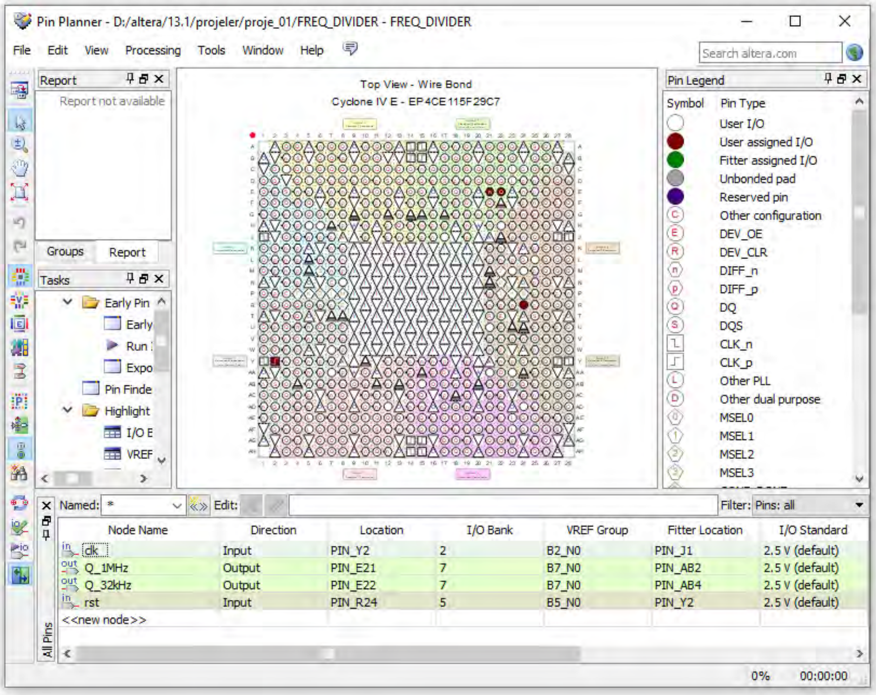


Figure 9. Wave Window Simulation Results of ModelSim Software

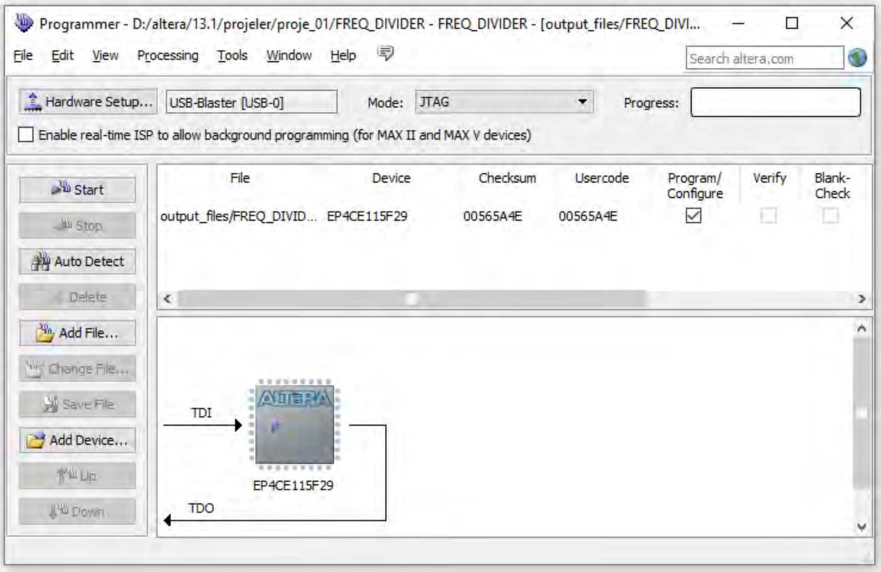
For the application, the pins of the FPGA IC must be associated with the pins defined in the entity. For this, click on Pin Planner in the Assignments menu. The Pin Planner interface shown in Figure 10 will open. There is an All Pins section at the bottom of this interface. Here are the pins defined in Entity. Here, the relevant pin numbers should be entered according to the datasheet on the Location tab. For example, Y2 pin is given for 50MHz. For reset operation, a push button connected to R24 can be selected. The outputs can be transferred to the outside world from the header ends or given to the LEDs for monitoring. Here, E21 and E22, which are the connections of the green LEDs, are used. Although their flashing is not obvious due to the high frequency, these LEDs on the board should be able to be observed at low brightness like a PWM with 50% duty.



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
clk	Input	PIN_Y2	2	B2_N0	PIN_J1	2.5 V (default)
Q_1MHz	Output	PIN_E21	7	B7_N0	PIN_AB2	2.5 V (default)
Q_32kHz	Output	PIN_E22	7	B7_N0	PIN_AB4	2.5 V (default)
rst	Input	PIN_R24	5	B5_N0	PIN_Y2	2.5 V (default)
<<new node>>						

Figure 10. Pin Planner Interface

After this process, Pin Planner is closed. The project is compiled again. As a result of error-free compilation, it can be passed to the programming phase. The DE2-115 development board is connected to the computer and its drivers are installed. For drivers, the Altera folder containing the installed version of the Quartus program should be searched. Then click Programmer in the Tools menu and the interface in Figure 11 will open.



USB-Blaster [USB-0] Mode: JTAG

☐ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check
output_files/FREQ_DIVID...	EP4CE115F29	00565A4E	00565A4E	☑	☐	☐

Figure 11. The Programmer Interface

The freq_divider of the prepared project should be seen on the opened page. If this file does not exist, the add file button is pressed. In the opened interface, the FREQ_DIVIDER.sof file is selected in the output_files folder in the folder where the project is located. When the start button is pressed, the program will be loaded into the FPGA.

FPGAs are integrated circuits with very powerful processing capacity. However, creating an embedded system with an FPGA is a very laborious and detailed work. According to the needed system, How much speed the needed how much processing load it has, how much hardware it needs to be solved with logic elements, the EEPROM capacity required for the program, etc. should be carefully determined, the FPGA to be used in line with these needs should be determined and the necessary PCB design should be made. All these issues should be carefully examined for determining the system structure to consist of an FPGA or microcontroller.

References

Baker, G. (n.d.). *VHDL references*. https://www2.cs.sfu.ca/~ggbaker/reference/

Chu, P. P. (2008). *FPGA prototyping by VHDL examples*. A John Wiley & Sons Inc. Publication.

FPGA designs with VHDL documentation. (n.d.). Retrieved July 11, 2021, from https://vhdlguide.readthedocs.io/en/latest/index.html

Gunes, E. O., & Ors, S. B. (n.d.). *VHDL notes*. Retrieved June 5, 2021, from https://web.itu.edu.tr/~ayhant/dersler/sstu/vhdl/VHDL_sunumI.pdf

IEEE standard for VHDL language reference manual. (2019). In *IEEE Std 1076-2019* (pp. 1–673). https://doi.org/10.1109/IEEESTD.2019.8938196

IEEE Standards Board., & IEEE Computer Society. Test Technology Technical Committee. (1993). *1149.1-1990 - IEEE standard test access port and boundary-scan architecture*.

Leong, P. H. W., Tsang, P. K., & Lee, T. K. (1998). A FPGA based forth microprocessor. *IEEE Symposium on FPGAs for Custom Computing Machines, 1998-April*, 1–2. https://doi.org/10.1109/FPGA.1998.707903

Nageswaran, J. M. (1997). *VHDL reference manual*. Synario Design Automation. www.synario.com

Pak, B. (n.d.). *VHDL lecture notes*.

Rajewski, J. (2017). *Learning FPGAs : digital design for beginners with Mojo and Lucid HDL*. O'Reilly Media, Inc.

Trimberger, S. M. (2015). Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology. *Proceedings of the IEEE, 103*(3), 318–331. https://doi.

org/10.1109/JPROC.2015.2392104

Unlersen, M. F., Yaldiz, E., & Imeci, S. T. (2018). FPGA based fast bartlett DoA estimator for ULA antenna using parallel computing. *Applied Computational Electromagnetics Society Journal*, *33*(4).

Unlersen, M. F. (2015). FPGA Kullanılarak Dizi Anten Performansının İyileştirilmesi - Improving of array antenna performance using FPGA. In *Institute of Science and Technology - Electrical and Electronics Engineering Department*. Selcuk University.

VHDL mini-reference. (n.d.). The University of California, Irvine. Retrieved June 5, 2021, from https://www.ics.uci.edu/~jmoorkan/vhdlref/vhdl.html

VHDL tutorial: Learn by example. (n.d.). Retrieved July 11, 2021, from http://esd.cs.ucr.edu/labs/tutorial/

What is an FPGA? Field Programmable Gate Array. (n.d.). Retrieved June 11, 2021, from https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html

What is an FPGA? Programming and FPGA basics - INTEL. (n.d.). Retrieved June 11, 2021, from https://www.intel.com/content/www/us/en/products/details/fpga/resources/

About the Authors

Muhammed Fahri UNLERSEN received B.Sc., MSEE and Ph.D. degrees in Electrical and Electronics Engineering from Selcuk University, Konya, Turkey in 2004, 2007 and 2015, respectively. He was a Lecturer in Doganhisar Vocational School at Selcuk University from 2005 to 2016. He has been working as an Assistant Professor in the Department of Electrical and Electronics Engineering at Necmettin Erbakan University since 2016. His current research areas are the array antennas, electromagnetic field theory, field programmable gate arrays and optimization techniques. He also has interests in control theory, inverse kinematics of 6 DoF Stewart Platform and experimental motion platforms capable of generating high g for simulators.

Email: mfunlersen@erbakan.edu.tr, Orcid: https://orcid.org/0000-0001-7850-6712

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 13%.

To Cite This Chapter:

Unlersen, M. F. (2021). Field Programmable Gate Array. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 179–207). ISRES Publishing.

Deneyap Card

Resul BUTUNER
Konya Adil Karaagac Vocational and Technical Anatolian High School

M. Hanefi CALP
Karadeniz Technical University

Introduction

Education systems are being developed by children who prepare applications with robotic coding at an early age (Goksoy & Yilmaz, 2018). 21. looking at the age of technology of the century, it has become very important to actively use the technological innovations required by the era in courses in order to ensure the permanence of Education. At the beginning of these technological innovations are the robotic and robotic coding concepts (Butuner, 2019). Currently, many microcontrollers use electronic cards to provide us with solutions to problems in everyday life. One of these cards is the Deneap card, which has just been produced by engineers working in Baykar Defense.

DENEYAP card has been developed using the original ESP32-WROVER-E module, with a versatile and powerful dual core Tensilica Xtensa LX6 microprocessor electronic structure. There are modules for internal wifi and bluetooth communication on the card. Thanks to these features, it helps to prepare internet of things and cloud-based applications. With the wifi feature, data can be exchanged between the installed system and the cloud-based application. A closed loop wireless communication network can be established by using the DENEYAP card as a server. There is a dual mode Bluetooth feature on the card. This feature allows data exchange with devices with both BLE and Bluetooth EDR infrastructure. There is a high-sensitivity LSM6DSM sensor on the DENEYAP card, which enables 3 axis acceleration and 3 axis rotation measurements and measures the temperature of the system. MP34DT05 microphone, built into the DENEYAP card, helps to prepare both the loudness of the surrounding sounds and the applications for voice and speech recognition depending on them. There are 24 pins on the card to be used for general purposes. Those that are used as digital inputs and outputs from these pins are protected by PTC fuses against overcurrents. It has a special lipo connection connector to help power the system safely. There is an FPC connector on the card that allows the camera to be connected directly. There are 2 buttons and 1 RGB LED internally on the card. DENEYAP card allows output with 3.3V and 5V voltages to be used in peripheral units with external connection continuously. DENEYAP card was developed based on USB 32 and designed in a way that is compatible with breadcard. The pins on the card are soldered to the breadcard for various applications and projects. DENEYAP Mini, on the other hand, is the reduced size of the DENEYAP card. In Figure 1, the drawing of the DENEYAP card and the explanations of its pins are given (Deneyap Card, 2021).

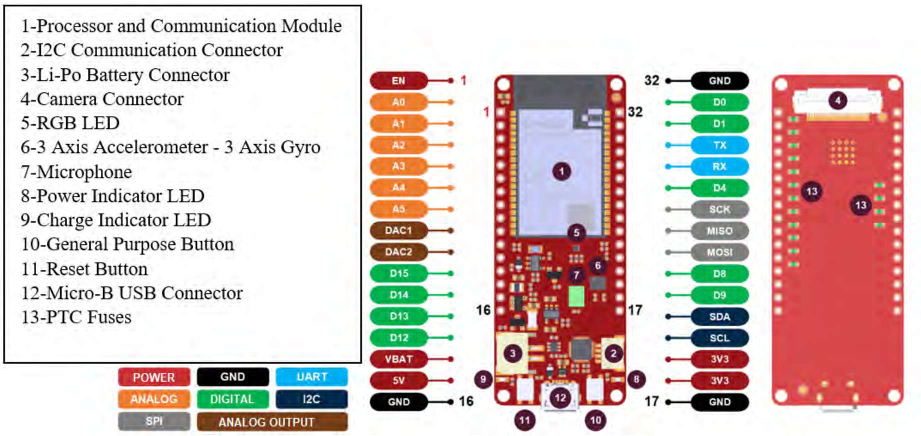


Figure 1. Pin Structure of DENEYAP Card(Deneyap Card, 2021)

Pin Properties

Power Pins

5V: This pin provides 5V voltage and 0.5A output. It can be used to feed peripherals to be connected externally to the card.

3V3: This pin provides 3V3 voltage and 1A output. Since some peripherals work with 3V3, voltage can be supplied from this pin.

VBAT: It gives the voltage of the externally connected Li-po battery as an output between 3.3V-4.2V. It can be used to feed external peripherals to be connected.

EN: For the ESP32 module to work, the pin value must be set to the logic “1” level. Logic “1” gives 3V3 voltage. The “RES” button is used to reset the system. When the button is pressed, the EN pin reaches the logical “0” level.

GND: It is the common ground to complete the circuit. Since it is designed as a card, it has more than one GND pin.

Input/Output Pins

D0-D15: These pins are digital input/output pins. It is used to give 3V3 or 0V according to the software in the card or to detect that these voltages are given from the outside. Short circuit protection is provided by PTC fuses.

Analog Pins

A0-A5: These pins are analog input pins. There are 6 pins on the card. It means that it is divided into 1024 parts with the applied voltage between 0-3V3 and that the voltage value can be read with this precision (3V3/1024). A0-A3 pins are only used as input pins, while A4 and A5 pins can be used as digital input/output.

DAC1-DAC2: These pins are used to provide analog output with voltage between 0-3V3. It also allows it to be used as a digital input/output pin.

Communication Pins

TX-RX: These are called serial communication pins. Uploading the code and communicating with the terminal is done through these pins. RX is the input pin and TX is the output pin. It is connected to the D2 and D3 pins. It is used as a digital input/ output pin.

I2C: For serial communication, it is an example of synchronous communication. Apart from the ground line, two lines, SCL and SDA, are needed to ensure communication. SDA is both input and output data line, and SCL is time synchronization pin as output. Besides I2C communication feature, it can also be used as a digital input/output pin and is connected to D10 and D11 pins.

SPI: Thanks to MOSI, MISO and SCK pins, any sensor or actuator can be controlled. It is the MOSI output and MISO input data line. SCK is the time synchronization pin as output. It can also be used as a digital input/output pin with SPI communication. SPI pins are connected to D5, D6 and D7 pins.

PWM Pins

PWM0 – PWM1: It is called square wave output pin. These pins are connected to the D0 and D1 pins. It can be used as a digital input/output pin. All digital pins D0-D15 can be used as PWM pins.

Camera Pins

CAMD2-CAMD9: These are the pins where image data is transferred and used as output. These are pins D0, D1, D4, D5, D6, DAC2, A2 and A3.

CAMSD-CAMSC: CAMSD bidirectional data transfer is used as serial communication pins, and CAMSC is used as time synchronization. These pins are connected to A5 and DAC1.

CAMXC: It is the basic timing signal source required for the operation of the camera sensor from the peripherals. It is used as an input and is connected to the A4 pin. It allows it to be used as a digital input-output.

CAMPC: It is the timing mark included in the image data derived from the camera sensor. It is used as an output and is connected to pin D7. It can also be used as a digital input/output.

CAMV: Output pin indicating that the image data is available. It is connected to the A0 pin and is used only as a digital input.

CAMH: It is the output pin that indicates the image data is available. It is also connected to the A1 pin and is used only as a digital input.

Internal Sensor Pins

6-Axis Inertial Measurement Unit: It contains IMUSD and IMUSC pins, and IMUSD is used as bidirectional data transfer and IMUSC as input for time synchronization. These pins are also connected to the common SDA and SCL lines.

Microphone: Contains MICD and MICC pins and MICD is used as bidirectional data transfer and MICC as input for time synchronization.

Capacitive Sensor Pins

T0-T5: It is the input pin used for capacitive sensing and is connected to A4, A5, D15, D14, D13 and D12 pins respectively. These pins can be used as digital inputs/outputs.

RGB Pins

LEDR: Internally, it is the pin that the red led is connected to and is connected to the RX line. It is also used as a status led during serial port data communication.

LEDG: Internally, it is the pin that the green led is connected to and is connected to the TX line. It is also used as a status led during serial port data communication.

LEDB: Internally, it is the pin to which the blue led is connected and it is connected to the SDA line. It is also used as a status led during I2C data communication.

Button Pin

GPKEY: It is used for general purpose. It is the pin to which the internal button is connected. It is also used to put the system into manual install mode. It is connected to the D8 pin and can also be used as a digital input and output.

Programming with Arduino IDE

Arduino IDE Editor Download and Installation

It is compatible with Arduino IDE in order to upload a program to the DENEYAP Card. The version suitable for your operating system must be downloaded from Arduino Ide’s own page (https://www.arduino.cc/en/software). The Arduino ide download screen is given in Figure 2. On the page that opens, your operating system is selected from the

Download Option tab, and then the installation file is downloaded by clicking the just download button.

Downloads

Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

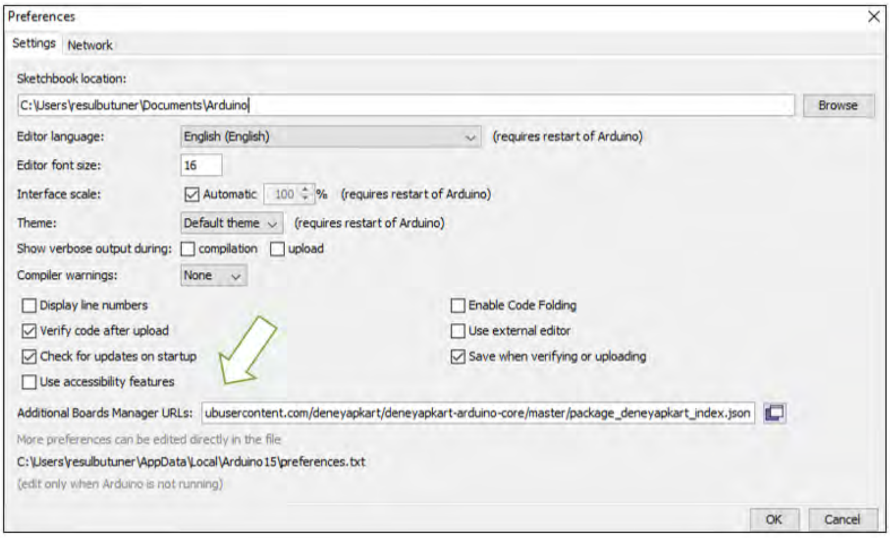
Mac OS X 10.10 or newer

Release Notes Checksums (sha512)

Figure 2. Arduino Ide Editor Download Screen

Adding DENEYAP Card to Arduino IDE

It must be added to the Arduino IDE before the program can be uploaded to the DENEYAP Card. For this, the window is opened by following the File-Preferences step in the Arduino IDE. In this window, the address of the JSON file should be copied to the Additional Circuit Cards Manager URLs section (https://raw.githubusercontent.com/deneyapkart/deneyapkart-arduino-core/master/package_deneyapkart_index.json) in the Settings tab. Figure 3 shows the preferences screen.



Preferences

Settings | Network

Sketchbook location:

C:\Users\resulbutuner\Documents\Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 16

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

☐ Use accessibility features

☐ Enable Code Folding

☐ Use external editor

☑ Save when verifying or uploading

Additional Boards Manager URLs: ubusercontent.com/deneyapkart/deneyapkart-arduino-core/master/package_deneyapkart_index.json

More preferences can be edited directly in the file

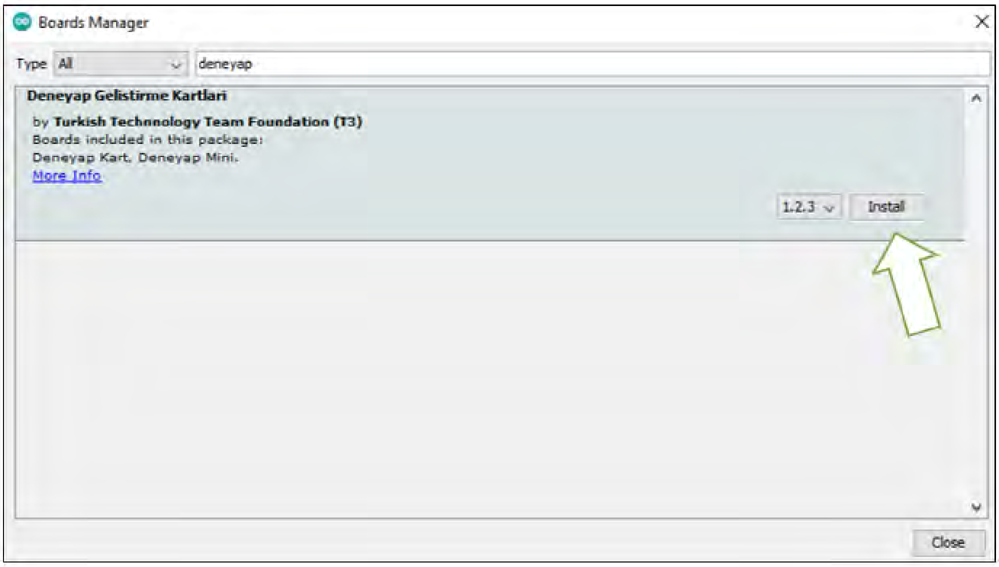
C:\Users\resulbutuner\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Figure 3. Arduino IDE Preferences Screen

Then, “DENEYAP Development Cards” is written in the search line on the screen that appears by following the steps of Tools-Card-Card Manager. The latest version is the default on the new screen. Clicking on the Install button will perform the installation.

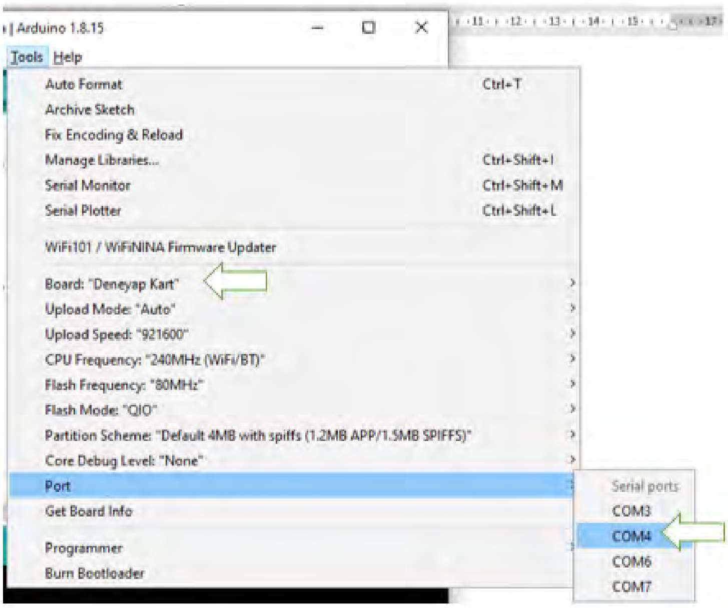
The Card Manager screen is given in Figure 4.



Deneyap Gelistirme Kartlari
by **Turkish Technology Team Foundation (T3)**
Boards included in this package:
Deneyap Kart, Deneyap Mini.

Figure 4. Arduino IDE Card Manager screen

Finally, the coding area is entered by selecting the Tools-Kart, then “DENEYAP Card” and the Port to which the card is connected. Figure 5 shows the screen for selecting DENEYAP Card and related port.



Board: "Deneyap Kart"
Upload Mode: "Auto"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"

Figure 5. Arduino IDE Card and Port setting

“File-Examples- Examples for DENEYAP Card” is selected to access the sample codes made with the DENEYAP Card. Here, related applications using various sensors can be examined. In Figure 6, the screen is given to access the examples made with the DENEYAP Card.

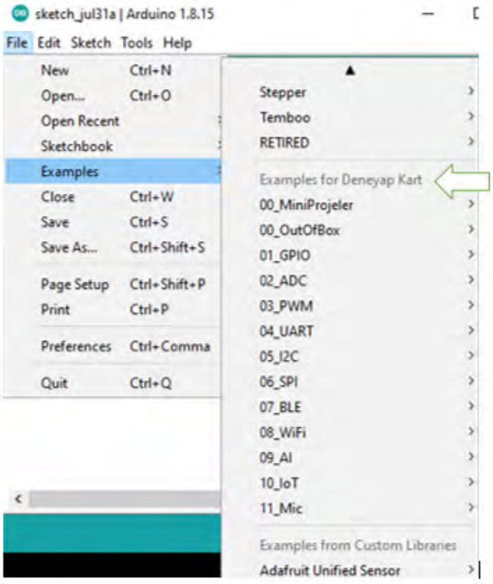
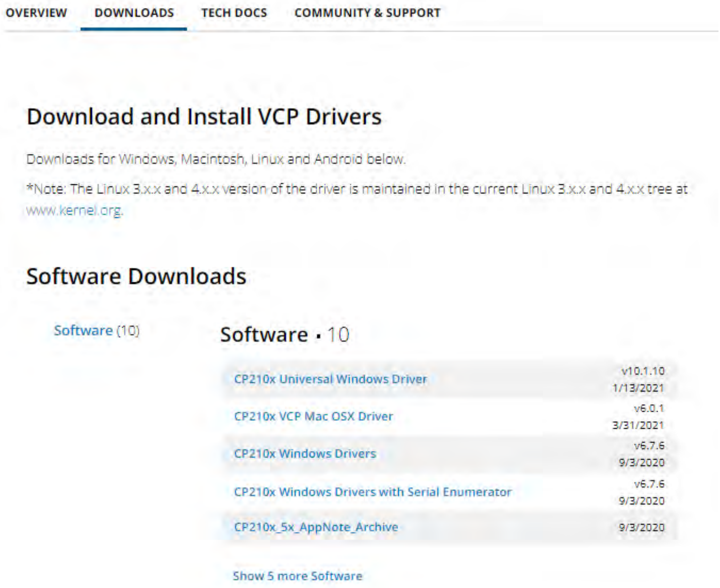


Figure 6. Examples of DENEYAP Card

USB Driver

If the drivers are not automatically installed when you insert the Test Card to your computer (https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers) the Downloads tab is selected from the address link and the driver suitable for the operating system is downloaded and installed. The download screen is given in Figure 7.



Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

Software Downloads

Software (10)

Software • 10

CP210x Universal Windows Driver	v10.1.10 1/13/2021
CP210x VCP Mac OSX Driver	v6.0.1 3/31/2021
CP210x Windows Drivers	v6.7.6 9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6 9/3/2020
CP210x_5x_AppNote_Archive	9/3/2020

Show 5 more Software

Figure 7. USB Driver File Download Screen

Programming with MicroPython

Download and Installation

DENEYAP Card works in harmony with MicroPython. First of all, Visual Studio Code (https://code.visualstudio.com/Download) must be downloaded and installed by clicking the appropriate tab for the operating system from the download link. Next, the Pymakr plugin needs to be installed via the Visual Studio Code editor. After the installation is complete, the Visual Studio Code should be opened, the "Pymakr" plugin should be searched and installed from the "Extensions" tab. In Figure 8, the installation order of the "Pymakr" plugin is given (Deneyap Card, 2021).

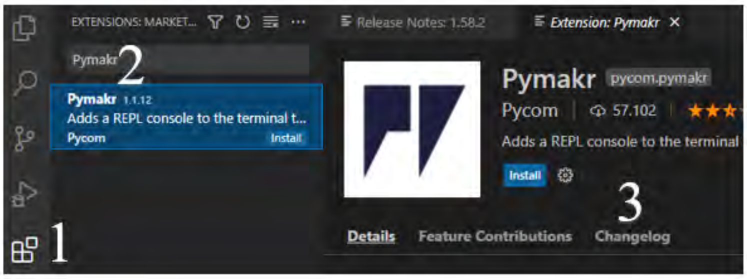


Figure 8. "Pymakr" Plugin Installation Sequence

Programming with DENEYAP Block

It offers the opportunity to program the DENEYAP Card on a block-based basis. For this, a program can be prepared with application-oriented blocks by clicking on the (https://deneyapkart.org/deneyapkart/deneyapblok/) address. DENEYAP Card Block screen is given in Figure 9. Block or text-based programs created on the block page are uploaded to the DENEYAP Card through another editor such as Arduino IDE.

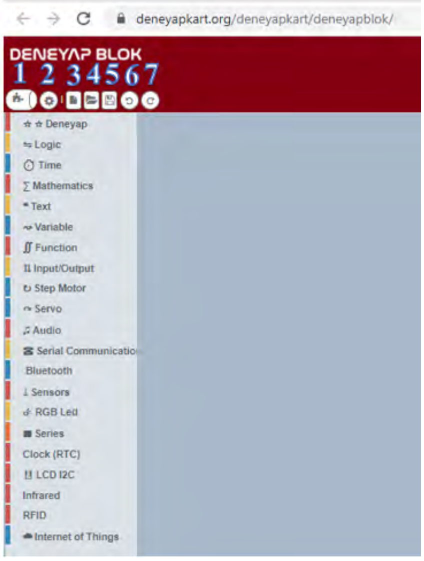


Figure 9. DENEYAP Block screen

Looking at the page, respectively;

1: It allows switching between block or text-based editors.

2: This section with settings on the block screen, which blocks to show or not, etc.

3: It is used for a new worksheet.

4: It is used to open the previously saved file.

5: It is used to save the file with the bloc extension.

6: It is used to undo the action.

7: It is used to redo the undo action.

DENEYAP Camera

DENEYAP Camera has an infrared filter lens, 2 Megapixels and OV2640 sensor. Compatible with DENEYAP Card and connected to the card with a 24-pin Flat Flexible Cable. Video shooting is done with 15 FPS image transfer rate and 1600x1200 pixel resolution. Likewise, it is possible to take photos with a resolution of 1600x1200 pixels. Using DENEYAP Camera, face detection, image classification, object recognition, motion detector, etc. artificial intelligence applications can be made. DENEYAP Camera image is shown in Figure 9.

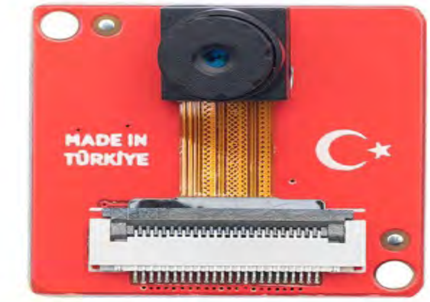


Figure 9. DENEYAP Camera

References

Butuner, R. (2019). Effect of Coding and Robotic Coding Training on Students. Journal of Information Systems and Management Research, 24-30

Deneyap Card. (2021). Access Adress: https://docs.deneyapkart.org/

Goksoy, S., & Yilmaz, I. (2018). The Opinions Of Information Relations Teacher And Their Students Wıth Regard To Lessons Of Robots And Coding. Duzce University Journal of Social Sciences Institute,8(1), 178-196.

About the Authors

Resul BUTUNER is a Computer Teacher at Adil Karaagac Vocational and Technical

Anatolian High School in Konya, Turkey. He has a master's degree in Computer Engineering from Necmettin Erbakan University. His main areas of interest are artificial intelligence, robotic coding, data mining and augmented reality applications. He is an instructor in the field of Robotic coding within TUBITAK. He continues to write a book in the field of robotic coding at the Ministry of National Education. He worked as a coordinator in budgeted projects related to student education.

E-mail: rbutuner@gmail. com, Orcid: 0000-0002-9778-2349

M. Hanefi CALP received Ph. D. degree in 2018 from the department of Management Information Systems at Gazi University, one of the most prestigious universities in Turkey. He works as Associate Professor in the department of Management Information Systems of the Faculty of Economics & Administrative Sciences of the Karadeniz Technical University. His research interest includes Management Information Systems, Artifical Neural Networks, Expert Systems, Fuzzy Logic, Risk Management, Risk Analysis, Human-Computer Interaction, Technology Management and Project Management.

E-mail: mhcalp@ktu.edu.tr, Orcid: 0000-0001-7991-438X.

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 2%.

To Cite This Chapter:

Butuner, R. & Calp, M. Hc (2021). Deneyap Card. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 208–217). ISRES Publishing.

Programmable Smart Microcontroller Cards is published from the selected papers invited by the editors.

This edition includes 13 sections from the electronic cards of used in today's technology. All submissions are reviewed by at least two international reviewers.

The purpose of the book is to provide the readers with the opportunity of a scholarly refereed publication in the field of techonology and engineerring.

Programmable Smart Microcontroller Cards is published by ISRES Publishing.

ISRES
Publishing