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.

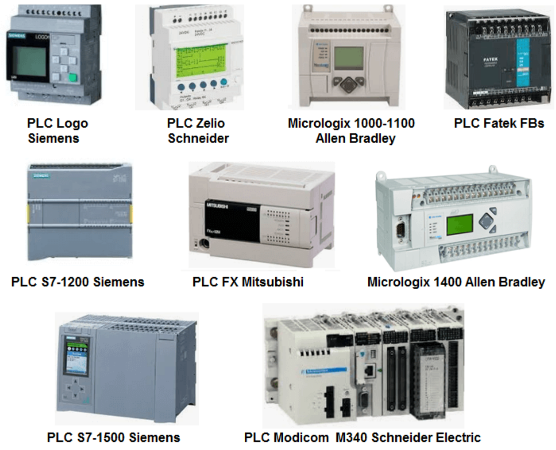

Figure1: PLC Types

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.

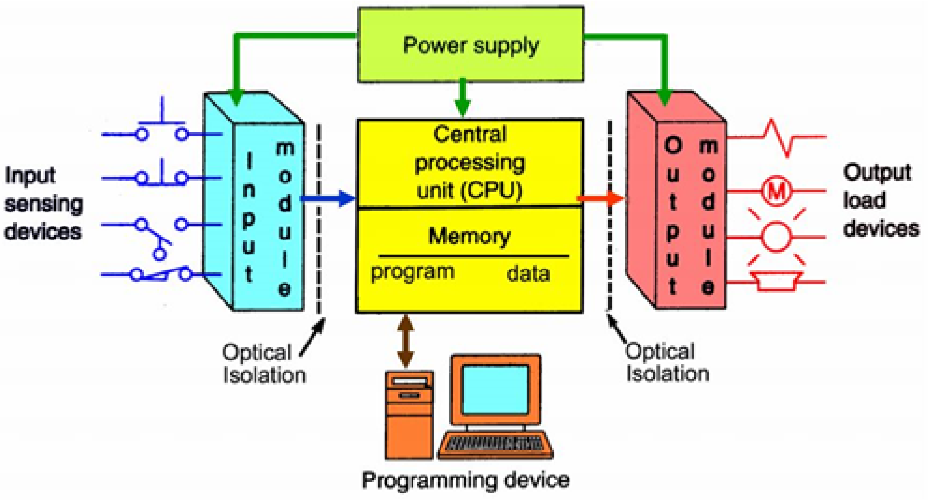

Figure 2: PLC System

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.

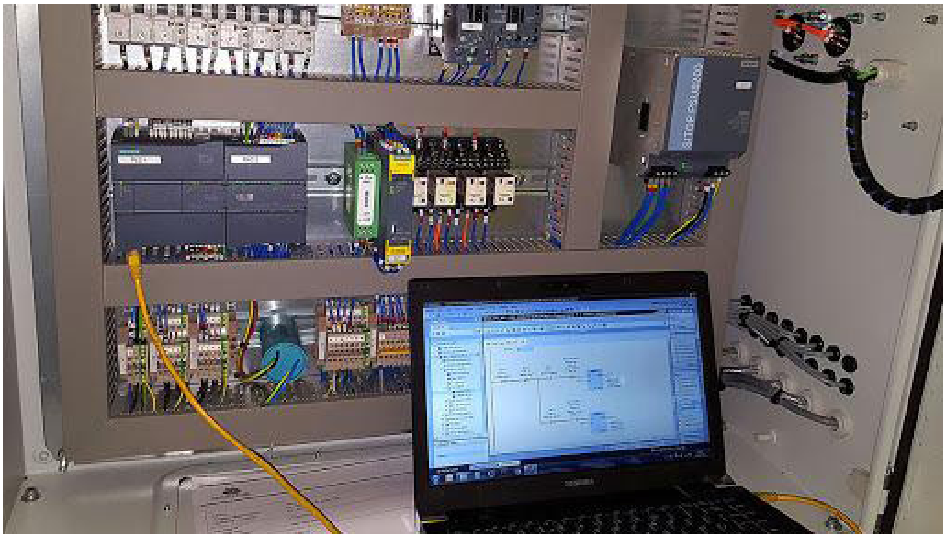

Figure 3: Programming the PLC

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.

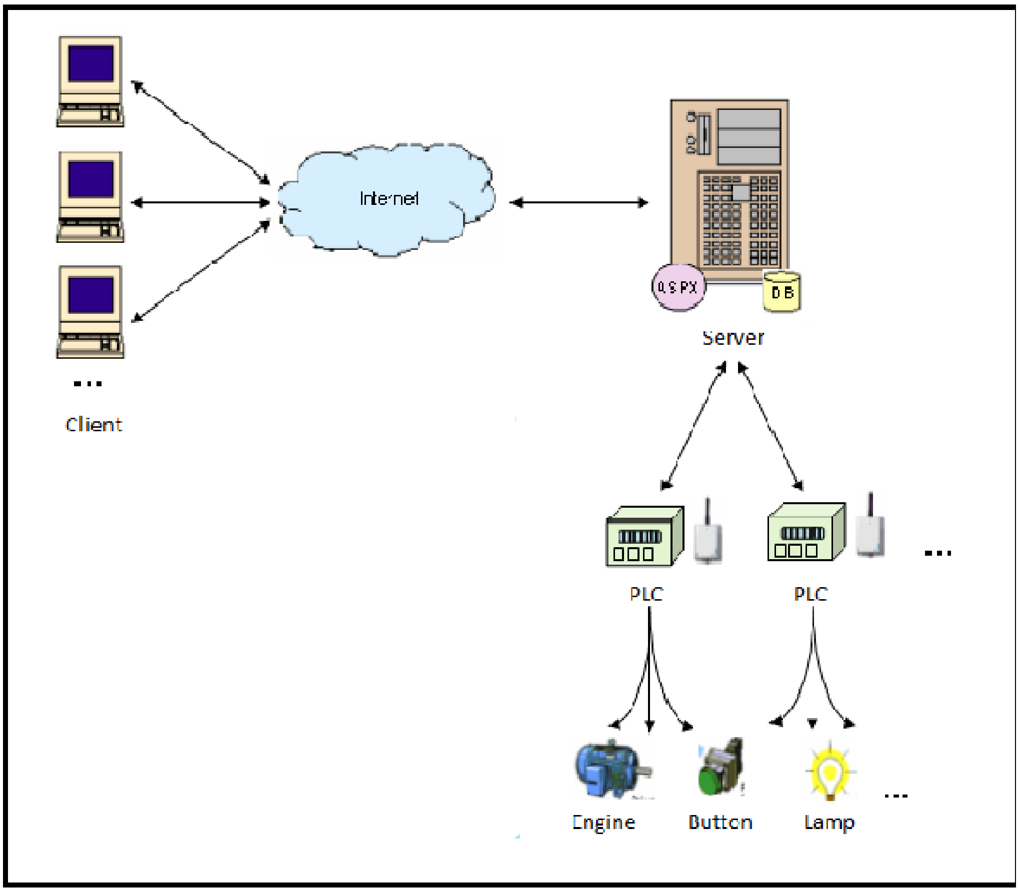

Figure 4: General Structure of the Remote Access System to the PLC

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).

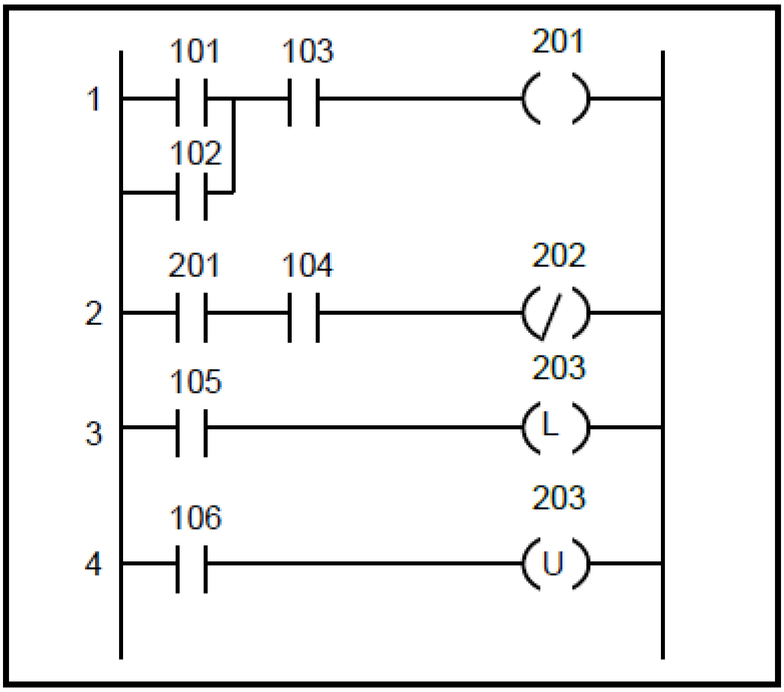

Figure 5: An Example Ladder Diagram Using Relay Type Commands

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).

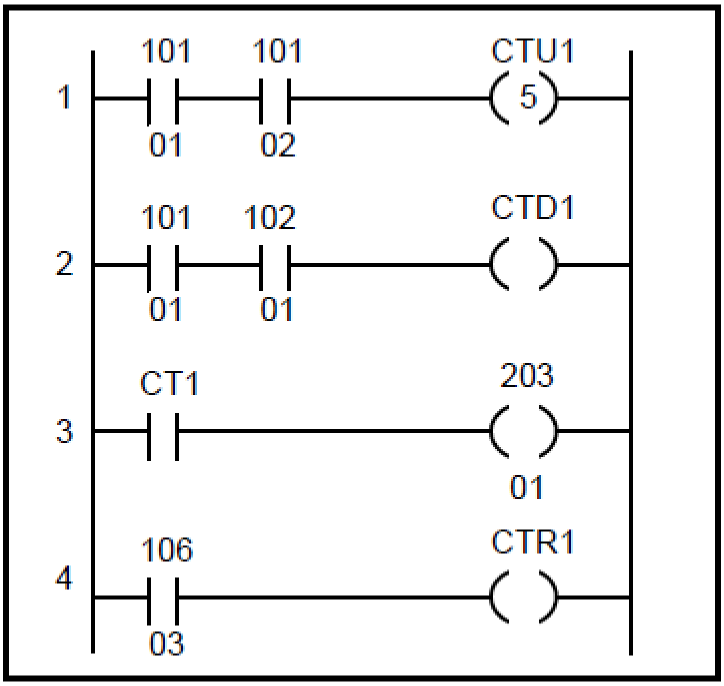

Figure 6: An example Ladder Diagram Using Counting type Commands

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.

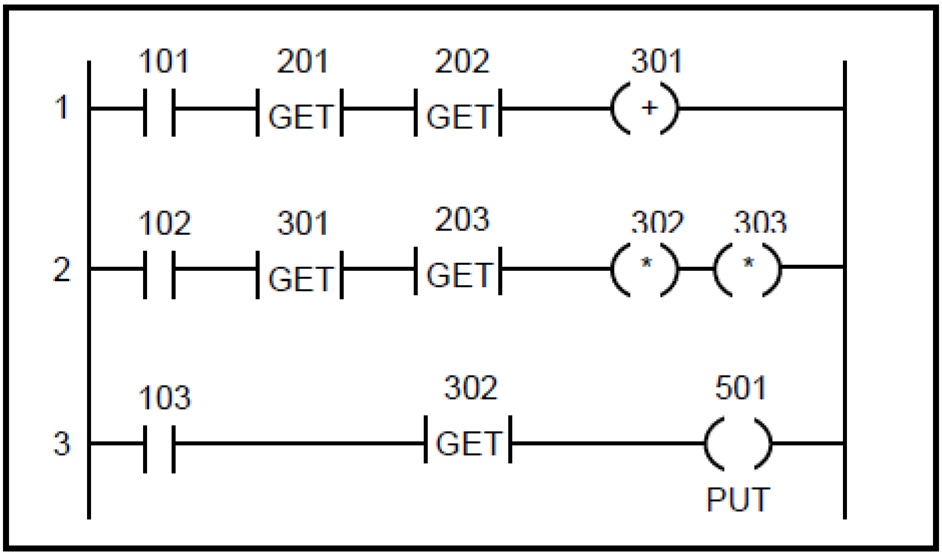

Figure 7: An Example Ladder Diagram About Data Transfer and Arithmetic Operation Commands

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.

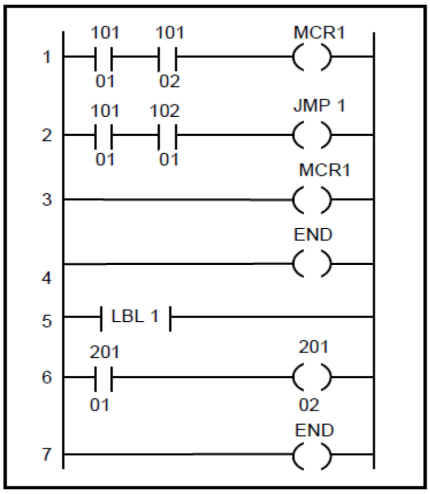

Figure 8: An Example Ladder Diagram Using Data Comparison Operation Instructions and Program Flow Control Instructions

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.