# Design and Implementation of a Programmable Logic Controller Using PIC18F4580

Ufuk Sanver[1], Erdem Yavuz[2], Can Eyupoglu[3]
Department of Computer Engineering
Istanbul Commerce University
Istanbul, Turkey
[1]usanver@ticaret.edu.tr, [2]eyavuz@ticaret.edu.tr,
[3]ceyupoglu@ticaret.edu.tr

Tuncay Uzun
Department of Electronics and Communication Engineering
Yildiz Technical University
Istanbul, Turkey
uzun@yildiz.edu.tr

***Abstract*—Use of electricity in industry opened a new age which is dominated by industrial automation. In the beginning, automation systems were based on electro-mechanical systems consisting of relays and contactors. Developments in semi-conductor technology inevitably transformed the existing structure and cause electronic-controlled Programmable Logic Controller (PLC) to prevail in the automation systems. PLCs consist of a microcontroller/microprocessor, relay or transistor for switching, isolating equipment such as opto-couplers and voltage converter/inverters. Industrial PLCs are quite expensive devices. In this study, design and implementation of a low-cost PLC using PIC18F4580 microcontroller are realized. Special software including a graphical user-interface is developed for programming the PLC and transferring code data to the device. Both ladder diagram and word instructions can be used for programming the PLC.**

***Keywords*—industrial automation; programmable logic controller; microcontroller**

## I. Introduction

If a system does the works or missions of people partly or completely, this system is called as automatized systems. First, it was tried to automatize the systems, needing only human power. After this process, it was tried to automatize intelligent systems. The main reasons to automatize systems are the restrictions of humans in power, data process capability, and working speed. Furthermore, products manufactured by humans may not always be of the same quality because the working concentration of workers varies over time. All the above-mentioned reasons show the significance of automation.

History of automation dates back to ancient times. Although industrial control systems are a part of modern age (after 18th century), there has been automation applications which were utilized by civilizations in Ancient Greece and Ancient Time Arabic Groups. Water clocks, oil lamps, and improved water depots are some example of these applications. The steam machine, which was improved by James Watt in 1788, opened a new age, the industrial age. There were a lot of studies and improvements related to steam machine in 19th century. Successful researches based on Maxwell equations have had an important role in improving of industrial automation. After these influential improvements, many technological products were invented especially in branch of transportation, manufacturing, and power plants [1, 2].

In the beginning, automation systems were based on electro-mechanical systems consisting of relays and contactors. After 1970s, a new product has been used with the help of developments in semi-conductor technology, called Programmable Logic Controller (PLC), which has enormous effect and wide spread use in industrial production. The author of the book “Introduction to PLCs”, Jay Hooper explains the reason of popularity of PLCs with an example. A car manufacturer was tired because of changing the electro-mechanical control units for every modification in production line. They decided to take help from a software company to control their system. It worked well, but the company understood that they were depended for software companies for every modification, because they did not know software well enough such as FORTRAN or Assembler. Thus, they decided to purchase a system, which has Ladder-Systematic that every electronics expert in industry knows and uses well. It is also the point behind of the success of the PLC: The Ladder Diagram is the standard of industrial automation. Dick Morley, who is also known as “the father of PLC” produced his first PLC in 1968. He named his company, Modicon, which means “Modular Digital Controller”. In 1988, Siemens produced the PLC S5, which had a wide spread use. The development and use of PLC technology continues and PLCs are still very popular products in industrial automation area [2, 3].

A typical PLC is a device, which consists of a CPU, input and output modules, memory, programming device, and user program [4]. In addition to these ones, most PLC devices also have connection components for communication methods such as Controller Area Network (CAN), Modbus, ProfiNet, and Ethernet [5-7], serial (RS-232, RS-485 etc.).

In genarlly PLCs, there are three standard programming languages in the software using: LAdder Diagram (LAD), STatament List (STL) and State Flow Chart (SFC). You can switch from one language to the other and choose the most suitable language in the PLC sofware programming.

The organization of the rest of the paper is as follows. In Section II, related works are given. Section III introduces the proposed method and implementation of the system. In Section IV, simulation of the system is demonstrated. Finally, conclusions and future works are summarized in Section V.

## II. RELATED WORKS

In literature, there are a lot of studies and books [8, 9] related to PLCs. Some academic studies about PLCs are listed and explained briefly below.

Several studies in literature are focused on design new tools for PLC devices. For example, Burhan et al. added new items to NAIS FPI·C24 PLC some DC equipment such as motor, relay, and solenoid piston cylinder, enabling this PLC to be a user friendly device [10].

Some studies focused on performance and conformance analysis of PLCs. Wagner et al. [11] studied the sag response of the PLCs used in factories for bad power quality conditions. Baniyonis et al. [12] focused their efforts on determining the source of errors in PLC software. They also proposed a new model to determine software errors better. Another example is the study of Darvas et al. [13] in which they developed the conformance of PLCs with different levels of permissibility.

Cyber security in PLC software has an important role, because PLCs produced in recent years come with Ethernet/IP connections. Wardak et al. [5] showed the vulnerabilities of recent PLCs from network attacks. In a study [6], researchers investigated the devices having the risk of vulnerability and the techniques to overcome it. Gavrilov [7] also explained the cybersecurity methods for PLC applications in a very recent study.

PLC education is of great importance. After their experiments on students, Gavali et al. [14] concluded that undergraduate students are more successful in learning PLC with the help of graphical laboratory experiments.

PLCs can be beneficially used in many different areas besides the industry. For instance, Howimanporn et al. [15] utilized PLC to develop a new speed control technique for conveyor belt using particle swarm optimization-based PID control. Iqbal et al. [16] developed a tank-level control implementation using PLC. Moreover, Akmansayar et al. [17] used PLCs for railway signalization system.

There are also studies about improving new programming methodologies for PLCs and the capacity of PLCs without changing hardware of PLCs. Petri-Nets have common usage from industrial applications to sportive game applications [18]. Andreu et al. [19] proposed a new programming approach for PLCs using Petri-Nets. Moallim et al. [20] improved wireless communication capacity of PLC for ZigBee communication. They asserted that their own PLC has more feasibility and flexibility in wireless communication.

Rida et al. [21] designed a new PLC device using ATxmega256A3U-AU Microcontroller, asserting that their system is an ideal solution for mini-board applications. Asif et al. [22] designed a PLC prototype using PIC18F452 microcontroller. Their major goal in this implementation is to teach internal architecture of PLCs to students. They used C language for programming the microcontroller and C# for user interface. Vasu et al. [23] designed and implemented a new PLC device based on multi-core PLCs because they believed that single-core PLCs have not generally enough speed for complex applications like speed control. Thus, they used python language for programming multicore device.

In this paper, a low-cost PLC has been designed and implemented using PIC18F4580 microcontroller. Graphical user-interface software has been designed for developing programs on the PLC and uploading code data to the device. In order to program the PLC, either ladder diagram or word instructions can be used.

## III. METHODS AND IMPLEMENTATION

The designed and implemented PLC consists of a microcontroller, input and output interfaces, programming device, and user program. The block diagram of our PLC design is shown in Fig. 1.

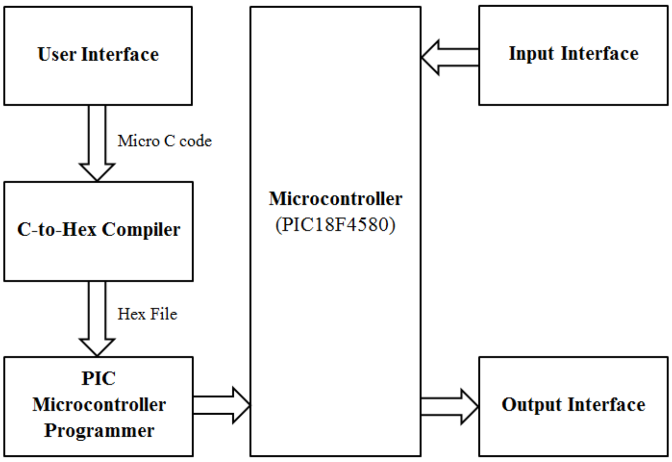


Fig. 1. Block diagram of our PLC design

Subsequent sections elaborate on hardware structure, user (design) interface, uploading software of PLC, and connection between PLC and user interface.

### *A. Hardware Structure*

Hardware of the implemented PLC has several components. The most important component is the PIC18F4580 microcontroller. The most prominent features of the PIC1854580 microcontroller for preferring it in this work are listed below:

- 40 pins in dual inline packages (DIPs),
- 32 KB flash memory,
- 1536 bytes SRAM and 256 bytes EEPROM data memory,
- 36 I/O pins,
- ECAN technology [24].

The designed PLC has both eight inputs and outputs. Four outputs are used for relay connections and other outputs are used for transistor connections. All the inputs and outputs connected to the microcontroller are isolated using opto-couplers. For uploading the code to the PLC, any device which is able to program the microcontroller such as Pickit, Biopic, and Mikroelektronika devices.

## *B. User Interface and Uploading Software of PLC*

The user interface has been developed in Visual Basic platform. Visual Basic platform is a commonly used platform in specific end-user applications [25]. The user interface of our designed PLC can be seen in Fig. 2.

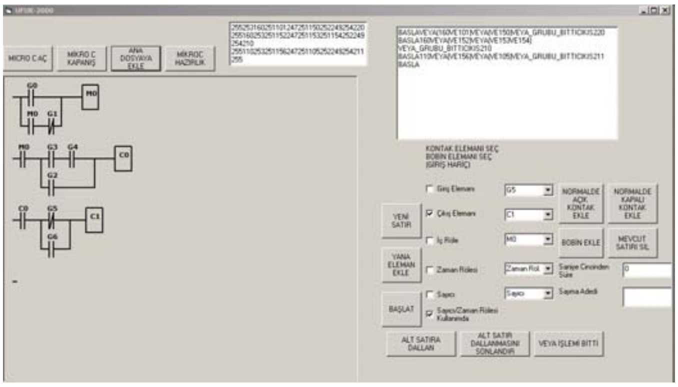

Fig. 2. User interface of the designed PLC

In this user-friendly environment, one can easily construct the ladder diagram using eight internal relays, eight counters, and eight timers and previously mentioned eight input and outputs. The number of these units can be increased when necessary. The main routine of the implemented PLC user interface is given in Fig. 3.

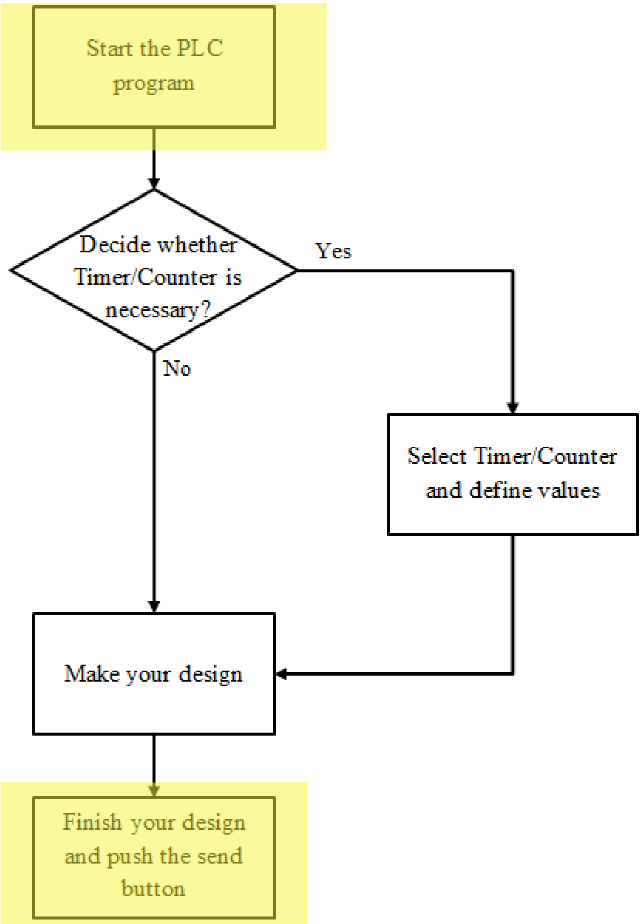


Fig. 3. Main routine of the implemented PLC user interface

The definitions for timers and counters are taken from user at the beginning. Main building blocks of PLC system design are normally-open, normally-close contacts and coils as in the systems based on relays and contactors. The circuits with these building blocks can also be mathematically expressed with bitwise operations. The PLC program creates a code for each building block and calculation. After the design of a coil, the system calculates the bitwise operation for this coil. Here, the serial connection of contacts is bitwise multiplication process and parallel connection of contacts is bitwise summing process. In the ladder diagram, each line consisting of some building blocks ends with a coil. In other words, operations of building blocks are equal to a coil at the end. A simple example of a ladder diagram is given in Fig. 4.

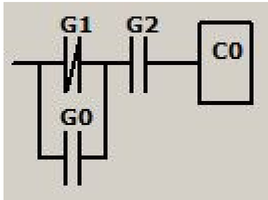


Fig. 4. A simple example of a ladder diagram

For the example in Fig. 4, “G0”, “G1”, and “G2” symbolize inputs contacts, where “G0” and “G2” are normally-open contacts and “G1” is normally-closed contact. “C0” symbolizes the coil of output. The program creates that code sequence for the ladder design in Fig. 4: “255-253-150-247-251-101-252-251-152-254-210”. After the end of the line (for the output coil, “C0”), the PLC program deciphers this code sequence. Each three digit has different meanings. The meanings of code bytes in this example are explained in Table I.

TABLE I. CODES BYTES AND MEANINGS FOR THE EXAMPLE LADDER

| Code bytes | Meaning |
|---|---|
| 255 | Start |
| 253 | Sub-branch |
| 150 | Normally-open contact of G0 |
| 247 | End of this sub-branch |
| 251 | Serial connected contact means, but here no usage |
| 101 | Normally-closed contact of G1 |
| 252 | End of OR operations |
| 251 | Serial connected contact, AND operation |
| 152 | Normally-open contact of G2 |
| 254 | End of the one side of equation |
| 210 | The coil output interface, C0 |

The program deciphers the code sequence explained in Table I and converts the sequence into bitwise operations: coil0=1*(nog0*1+ncg1*1)*nog2*1. The variables in this equation (of C code) are explained in Table II.

TABLE II. VARIABLES AND CORRESPONDING BUILDING BLOCKS IN THE LADDER

| Variable | Corresponding building blocks in the ladder |
|---|---|
| coil0 | The coil of C0 |
| nog0 | Normally-open contact of G0 |
| ncg1 | Normally-closed contact of G1 |
| nog2 | Normally-open contact of G2 |

While the user constructs his custom design step by step in the ladder diagram, the PLC program prepares the C code in the background. After completing the ladder diagram, the user should press the send button. Upon pressing the button, the PLC program invokes MikroC software automatically. There is no need to write any code or doing anything here in MikroC software.

### *C. Uploading the PLC Code to Device*

The MikroC software is utilized only for compiling the PLC code and producing hex file. If the user has a programming kit of Mikroelektronika for PIC, the user can embed the PLC code only using one button as seen in Fig. 5. Alternatively, other PIC programmers such as Pickit and Biopic can be used to embed hex files created by MikroC software.


Fig. 5. Embedding the code of PLC using Mikroelektronika device

## IV. SIMULATION

The electronic hardware of the implemented PLC is designed first in a popular simulation program, Proteus (ISIS) of Labcenter Electronics. All the hardware components for this PLC model are tested in the simulation program before assembling the prototype. However, here, minimum number of components is used in the simulation stage in order to track the steps of the user's custom program. For this purpose, only buttons for inputs and LEDs for outputs are used. The user should first embed the hex file in microcontroller in simulation program. After embedding the code, the user can start begin the simulation. The simulation screen-shot of the implemented PLC is given in Fig. 6.

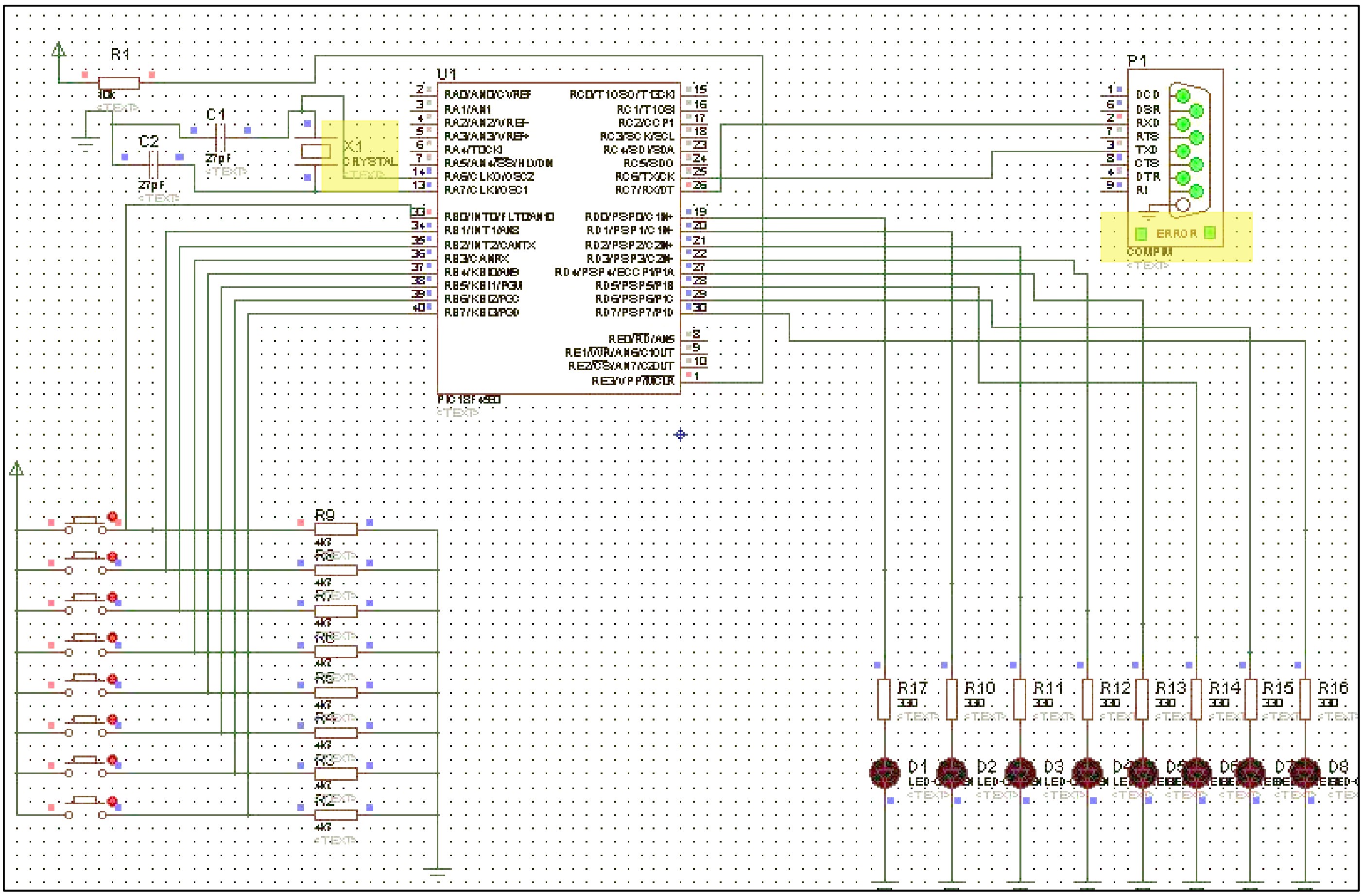


Fig. 6. Simulation screen-shot of the implemented PLC with minimum number of components

## V. Conclusion and Future Work

Commonly used expensive commercial products have much more capability that some of their features are used rarely. In this work, a low-cost PLC using PIC18F4580 microcontroller is designed and implemented. It is aimed to produce a basic PLC prototype which can easily be modified according to the needs of specific processes. Another reason to realize this project is to have an educational tool that helps students to understand the systematics of the PLC device and motivate them to make their own ones.

The choice of PIC18F4580 microcontroller is because it has a CAN communication capability. In this study, the RS-232 communication of the microcontroller is exploited. In the future, it is planned to incorporate CAN communication because it is a commonly used protocol in industry. In addition, it is aimed to transfer this code to Mplab software of Microchip and build the hex file directly from user interface.

## References


[1] U. Sanver, "Scoreboarding System for Table Tennis Game with Using Pic32," Master Thesis, Istanbul Technical University, 2015.

[2] E. Hayden, M. Assante, and T. Conway, "An Abbreviated History of Automation & Industrial Controls Systems and Cybersecurity," SANS Analyst Whitepaper, 2014.

[3] J. F. Hooper, Introduction to PLCs, Carolina Academic Press, 2005.

[4] A. S. Rahama1 and D. Mahmoud, "Control of Induction Motors by Using Programmable Logic Controllers (PLC)," International Journal of Science and Research (IJSR), vol. 5, pp. 1923-1926, 2016.

[5] H. Wardak, , S. Zhioua, and A. Almulhem, "PLC access control: a security analysis," IEEE 2016 World Congress on Industrial Control Systems Security (WCICSS), 2016.

[6] I. Ahmed, S. Obermeier, S. Sudhakaran, and V. Roussev, "Programmable Logic Controller Forensics," IEEE Security & Privacy, vol. 15, pp. 18-24, 2017.

[7] N. V. Gavrilov, "Appliance of WEB-technologies in automation of industrial facilitie," 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017.

[8] W. Bolton, Programmable Logic Controllers, Newnes, 2015.

[9] L. A. Bryan and E. A. Bryan, Programmable controllers: theory and implementation, Industrial Text Company, 1997.

[10] I. Burhan, A. A. Azman, and S. Talib, "Multiple input/outputs Programmable Logic Controller (PLC) module for educational applications," 2015 Innovation & Commercialization of Medical Electronic Technology Conference (ICMET), 2015.

[11] V. E. Wagner, A. A. Andreshak, and J. P. Staniak, "Power quality and factory automation," IEEE Transactions on Industry Applications, vol. 26, pp. 620-626, 1990.

[12] M. Baniyounis, A. Mesmar, "Diagnosability of Programmable Logic Controllers," 14th International Multi-Conference on Systems, Signals & Devices (SSD), 2017.

[13] D. Darvas, I. Majzik, and E. B. Vinuela, "Conformance checking for programmable logic controller programs and specifications," 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES), 2016.

[14] A. B. Gavali, S. A. Patil, and A. R. Koli, "Technology-Based Learning system in Programmable Logic Controller Education," 2016 IEEE Eighth International Conference on Technology for Education (T4E), 2016.

[15] S. Howimanporn, S. Thanok, S. Chookaew, and W. Sootkaneung, "Speed Control Technique for Conveyor Using PSO based PID with Programmable Logic Controller," 2016 IEEE/SICE International Symposium on System Integration, 2016.

[16] M. M. Iqbal, S. Gupta, and C. Gurudas, "Overflow Protection of Level Tanks using Programmable Logic Controller," International Conference on Electronics, Communication and Aerospace Technology(ICECA), 2017.

[17] B. Akmansayar, S. Kurtulan, S. B. Ors, "Design of core blocks and implementation on a programmable logic controller for a train signalization system," 2015 23th Signal Processing and Communications Applications Conference (SIU), 2015.

[18] E. Yavuz, U. Sanver, M. C. Kasapbasi, and R. Yazici, "An Implementation of Vibration-Based Automatic Score-Keeping System for Table Tennis Game," International Journal of Electronics; Mechanical And Mechatronics Engineering, vol. 4, pp. 725–732, 2014.

[19] D. Andreu, J.-C. Pascal, and R. Valette, "Fuzzy Petri Net-Based Programmable Logic Controller," IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, vol. 27, pp. 952-961, 1997.

[20] A. Moallim, J.-M. Lee, and D.-S. Kim, "Wireless Control and Monitoring using Programmable Logic Controller (PLC)," 2017 17th International Conference on Control, Automation and Systems (ICCAS), 2017.

[21] M. E. Rida, F. Liu, and Y. Jadi, "Design Mini-PLC based on ATxmega256A3U-AU Microcontroller," 2014 International Conference on. Information Science, Electronics and Electrical Engineering (ISEEE), 2014

[22] M. Asif, A. Raza, A. Sultan, and F. Malik, "Design of Mini PLC based on PIC18F452 Microcontroller using Concepts of Graceful Degradation," Technical Journal, University of Engineering and Technology Taxila, vol. 21, no. 1, pp. 51-57, 2016.

[23] P. Vasu; H. Chouhan, and N. Naik, "Design and implementation of optimal soft-programmable logic controller on multicore processor," 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), 2017.

[24] PIC18F2480/2580/4480/4580 Data Sheet, Microchip Company, 2007.

[25] U. Sanver, E. Yavuz, M. C. Kasapbasi, and R. Yazici, "Modelling and implementation of an automatic table-tennis scoreboarding system," 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), 2016.