British Journal of Mathematics & Computer Science
21(3): 1-19, 2017; Article no.BJMCS.32165
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Designing and Implementation of PIC Microcontroller Based Educational Kit

Mohamed B. El-Mashade[1*], Abdelmonem Fouda[2] and Ashraf M. Etman[2]

[1]Department of Electrical Engineering, Faculty of Engineering, Al Azhar University, Nasr City, Cairo, Egypt.
[2]Department of Computer Engineering, Modern Academy for Eng. and Tech., Cairo, Egypt.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJMCS/2017/32165
Editor(s):
(1) Kai-Long Hsiao, Taiwan Shoufu University, Taiwan.
Reviewers:
(1) Muhammad Mujtaba Asad, Universiti Tun Hussein Onn Malaysia, Malaysia.
(2) Fabio Leccese, Università degli Studi "Roma Tre", in Rome, Italy.
Complete Peer review History: http://www.sciencedomain.org/review-history/18354

Received: 11th February 2017
Accepted: 25th February 2017
Published: 27th March 2017

Original Research Article

Abstract

The microcontrollers are very common components in modern electronic systems. Their using is so widespread that it is almost impossible to work in electronics without coming across it. They are now providing us with a new way of designing circuits. As a result, less complex applications can be designed and built quickly and cheaply. These devices are low-cost and easy to be programmed. They have traditionally been programmed using assembly as well as high-level languages. In this regard, very complex control algorithms can be developed and implemented on the microcontrollers. The PIC is one of the most popular embedded real-time computers used in educational, commercial and industrial applications. Owing to its ease of programming and easy to interfacing with other peripherals, PIC became successful microcontroller.
This paper is devoted to practical hardware circuit design of educational PIC 16F877A microcontroller including: DC-motor (turn right and left), display a counter using seven-segment, and traffic light control system using LED's. Many other user applications can be developed and implemented to the Kit. Moreover, we develop a PIC simulator using Matlab software package which is capable to test, verify, and validate a user program loaded file. The input loaded file can be written either in high level language (micro-C) or in hexadecimal (H-file) file format. The different outputs of the implemented circuits are shown in virtual load of the developed simulator. The output window of this simulator contains the

**Corresponding author: E-mail: MohamedElMashade@gmail.com;*

address of the RAM memory location, data, assembly instruction along with its op-code, and description of each instruction in such a way that the user can understand, keep track and follow the program easily.

Keywords: Microcontroller; PIC; simulators; hex-file.

1 Introduction

The microprocessor based products can be classified into one of two categories: those which their construction is based on high-performance microprocessor and those in which some issues (such as cost, space, power, and rapid development) are considered to be more critical than raw processing power. The microprocessor of the last category is often called a microcontroller. A microcontroller differs from a microprocessor in many ways. The most important difference is its functionality. In this regard, the microprocessor is the integration of a number of useful functions into a single IC package. These functions include its ability to execute a stored set of instructions to carry out user defined tasks as well as its capability in accessing external memory chips to both read and write data from and to the memory. This means that the microprocessor is a general-purpose processor which contains no other components except the CPU on the chip itself. From this point of view, a general-purpose processor can be defined as a finite-state automaton that executes instructions held in a memory. Therefore, it needs the addition of other components (such as ROM, RAM, I/O ports, and timer) to make it functional. Although these external additions create bulkier and more expensive systems, the resulting systems have the advantage of versatility, enabling the designer to decide the amount of RAM, ROM, and I/O ports needed to fit a specified task at hand. This is not the case with microcontroller which is a CPU in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. Therefore, no other external components are needed for its application because all necessary peripherals are already built into it. Thus, we save the time and space needed to construct devices. In other words, the fixed amount of on-chip ROM, RAM, and number of I/O ports makes the microcontrollers ideal for many applications in which cost and space are critical. Additionally, the cost of microcontrollers is less compared to general-purpose computers, which makes them good candidates for embedded controller applications. Even though the microprocessors are considered to be powerful computing machines, their weak point is that they are not adjusted to be communicating with peripheral equipment. In this regard, for the microprocessor to communicate with peripheral environment, it must use specialized circuits added as external chips. On the other hand, the microcontroller is designed to be all of that in one. Hence, no other specialized external components are needed for its application and this in turn saves the time and space needed to design a device [1-5].

Like a microprocessor, a microcontroller needs to be able to compute, although not necessarily with big numbers, since it provides, in a simplified form, all the main elements of the microprocessor system on a single chip. In this regard, the microcontroller has features similar to those of the standard personal computer, where it is capable of storing and running a program. But it has other needs as well. Primarily, it must have excellent input/output capability. Additionally, because many embedded systems are both size and cost conscious, it must be small, self-contained and low cost. Moreover, the microcontroller may need to put up with the harsh environmental conditions, and be able to operate in extremes of temperature. As a result, less complex applications can be designed and built quickly and cheaply. A working system can consist of a microcontroller chip and just a few external components for feeding data and signals in and out. They tend to be used for control operations requiring limited amounts of memory but operating at high speed, with external hardware attached only as required by a specific application. Being inexpensive single chip computers, microcontrollers are easy to embed into larger electronic circuit designs. Their ability to store and run unique programs makes them extremely versatile [6].

Essentially, the microprocessor and microcontroller are similar, but with a little bit of difference. A CPU which is the heart of these devices needs a host of external devices to make it communicate with real-world. These devices which are independent circuits, work in harmony with the CPU, to make one system. In a typical PC, these devices are attached to the CPU, using hard-wired connections. This makes the system more flexible that means one can add more memory, change capacity of hard drives, add or remove external components. A microcontroller, on the other hand, is made up of most of these devices all packed within one integrated circuit. This facilitates the development process, as well as reduces the requirements of external components. However, this means that the number and type of integrated devices cannot be changed. The applications where a microcontroller will be used, are usually quite simple, and do not require as much processing power as a PC does. So, the microcontrollers with varying amounts of RAM, ROM, I/O lines and timers have been made available. Thus, a microcontroller is a complete, small scale computer with all the necessary devices on-board.

Peripheral interface controller (PIC) is a family of microcontrollers manufactured by the microchip technology Inc. In comparison with other microcontrollers, the beauty of these devices is their easy availability, low cost, simplicity in programming and handling. These characteristics along with its facility of interfacing, where one can connect analog devices directly without any extra circuitry, with other peripherals, make PIC as a successful microcontroller. Currently the PIC is one of the most popular embedded real-time computers employed in educational, commercial and industrial applications. It also enjoys some merits as: it is reliable where malfunctioning of PIC percentage is very low, its performance is very fast because of using RISC architecture, the power conception is also very less in comparison with other microcontrollers, and it's easier and quicker development. However, the drawback of PIC lies in the length of the program, which is big owing to the using of RISC and the program memory which is not accessible, and only one single accumulator is present. This category of embedded real-time computers has been classified as low range, mid range, high range, and digital signal processing microcontrollers [7-10].

Simulation is an important feature in engineering systems or any system that involves many processes. For example in electrical engineering, delay lines may be used to simulate propagation delay and phase shift caused by an actual transmission line. A microcontroller simulator is basically a computer program running on an independent hardware which simulates the CPU, the instruction set and the I/O of the target microcontroller. The CPU of PIC microcontroller uses Harvard architecture with separate program and variable data memory interface. This mapping of memory facilitates instruction fetching and the data accessing of variables. PIC16F877A is considered as one of the most commonly used microcontrollers owing to its low voltage, high performance CMOS microcomputer flash programmable and erasable read-only memory (PEROM) [1]. Here, we use it in designing an educational kit in which some useful applications are established. These applications include: DC-motor (turn right and left), display a counter with the aid of seven-segment, and traffic light control system using LED's. Our educational kit and developed simulator are very important for students, as users, to test, verify and validate applications and projects. They have the following features:

a) Firmware implementation using PICKIT3 to burn the H.L.L micro-C written program into the designed kit.
b) Capability of loading user H.L.L files written in micro-C into the developed "MODERN Simulator" and converting it into hex-file format.
c) Capability of loading hex-files to the developed "MODERN Simulator".
d) Displaying microcontroller RAM contents in an illustrating way such that the user keep track any error.
e) The developed simulator output window contains the RAM address, data and assembly instruction along with its op-code, and description of each instruction in such a way that the user can understand, keep track and follow the program easily
f) Displaying results using different virtual loads including; seven-segments display of counter, LED's representing traffic lights, and DC-motor orientation (clockwise and anticlockwise).
g) Friendly use of the hardware designed kit and developed simulator.
h) Providing manual user guide to deal with the kit and simulator.

The rest of the paper is organized as follows: section II introduces the internal structure of the PIC16F877A microcontroller with some of its basic features. Section III provides system model. Section IV illustrates the developed kit design steps and the developed software with different output forms and explanations. Finally, our concluded remarks are outlined in section V.

2 Specifications of PIC16F877A Microcontroller

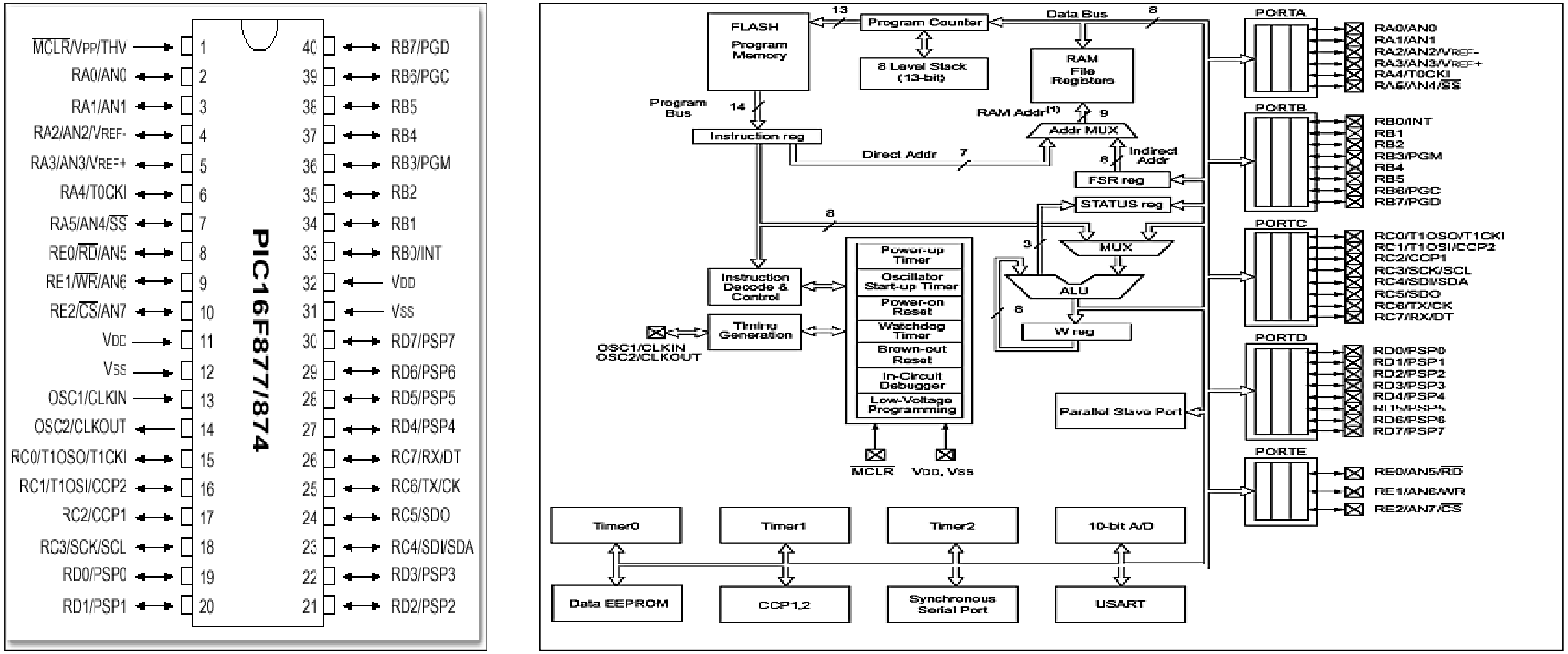
The PIC devices have been very successful in 8-bit microcontrollers. The main reason of this success is that microchip technology has continuously upgraded the device architecture and added the needed peripherals to the microcontroller to suit customer's requirements. The PIC microcontroller architecture is based on a modified Harvard RISC instruction set with dual-bus architecture, providing fast and flexible design with an easy migration path from only 6 pins to 80 pins, and from 384 bytes to 128 Kbytes of program memory. Although there are many models of PIC microcontrollers, they are upward compatible with each other and a program developed for one model can very easily be run on other models of the family. The basic assembler instruction set of PIC microcontrollers consists of only 33 instructions and most of the family members (except the newly developed devices) use the same instruction set [11-13].

Although there are several hundred models of PIC microcontrollers, choosing a specified one for an application is not a difficult task and requires taking into account the number of I/O pins required, the desired peripherals, the minimum size of program memory, the minimum size of RAM, whether or not EEPROM non-volatile data memory is wanted, the speed, the physical size, and the cost. Although there are many models which satisfy all of the above requirements, it is demanded to find the model which satisfies the minimum requirements as well as the one which does not offer more than the needed specifications. On the other hand, the PIC embedded real-time computers can be broken down into three main groups, which are 12-, 14-, and 16-bit instruction word. All the three groups share the same RISC architecture and the same instruction set, with a few additional instructions available for the 14-bit and many more instructions available for the 16-bit models. Instructions occupy only one word in memory, thus increasing the code efficiency and reducing the required program memory. Instructions and data are transferred on separate buses, thus the overall system performance is increased. PIC microcontroller is chosen to implement our kit due to its advantages like high speed, Harvard and RISC architecture, low cost and flexibility for programming. Some features are added to the kit to widen the application fields and consequently a multi-purpose programmable controller can be realized [5].

The pin configuration and the internal structure of a 40-pin 8-bit CMOS flash microcontroller PIC 16F877A is displayed in Fig. 1. The core architecture is high-performance RISC CPU with only 35 single word instructions. It has 3-operating speeds 4, 8, and 20 MHz clock input along with two types of internal memories: program memory and data memory. Program memory is provided by 8K words (or 8K*14 bits) of flash memory, and data memory which is separated into multiple banks that contain the general purpose registers (GPR) and special function registers (SFR). One type of data memory is a 368-byte RAM and the other is a 256-byte EEPROM.

3 System Model

The electric drive systems used in industrial applications are increasingly required to meet higher performance and reliability requirements. The direct current (DC) motor is an attractive piece of equipment in many industrial applications requiring variable speed and load characteristics due to its ease of controllability. Nowadays, the DC motor becomes an important device in a wide range of industrial applications that require high dynamics on position control such as numerically controlled machinery, robotics, automation and other mechanism where the starting and stopping functions are quickly and accurately. The ease of control and excellent performance of the DC motors will ensure that the number of applications using them will continue grow for the foreseeable future. Thus, automatic control of DC motor has played a vital role in the advance of electromechanical engineering. Microcontrollers provide a suitable means of meeting the required needs. This project is mainly concerned with DC motor speed control system through the use of microcontroller PIC16F877A.



a- Pin configuration **b- Internal structure**

Fig. 1. PIC 16F877A

Traffic congestion problem is a phenomenon which contributed huge impact to the transportation system in a large city. This causes many problems especially when there are emergency cases at traffic light intersections which are always busy with many vehicles. Nowadays, vehicular travel is increasing through the world and many countries are facing many problems at traffic light intersection which are caused many accidents between the emergency vehicle and other vehicle. Therefore, traffic light control at the intersection point is a matter of concern in large cities. However, the control of traffic at road junction, which was done purely by human effort, proves to be inefficient owing to the increasing rate of both motorists as well as the complexity of road networks. On the other hand, as the number of road users constantly increase and resources provided by current infrastructures are limited, modern control of traffic will become a very important issue in the future. Thus, a traffic light controller system is designed in order to solve the aforementioned problems. In this regard, traffic control establishes a set of rules and instructions that drivers, pilots, train engineers, and ship captains rely on to avoid collisions and other hazards. Motorists depend on traffic control devices to avoid collisions and travel safely to their destinations. Traffic control devices for highway travel include signs, signal lights, pavement markings, and a variety of devices placed on, over, near, or even under, the roadway. The signal light is probably the most easily recognized traffic control device. Thus, the need for the development of a microcontroller-based traffic light control system becomes indispensable. In other words, traffic light controlled by microcontroller is becoming a common place in many cities because these units can easily adjust for different timing sequence.

On several occasions, counting is required. The manual counting becomes time-consuming and inaccurate when objects to be connected are very large in number or they pass through in quick succession. In such situations, counter using electronic circuits are more reliable than manual counting. A counter with suitable connectivity can work as a stopwatch, or as frequency counter.

Our implementing kit is basically depends on the microcontroller PIC16F877A of 20MHz speed and powered by V_{dc} of 5volt. This kit is devoted to three applications: dc motor control, seven-segment display control, and traffic light control. To carry out these tasks, we use six input push buttons "main kit", as Fig. 2 illustrates, dc-motor, seven-segment and three LED's as the "output kit", as Fig. 3 depicts. The applications are designed and tested using Proteus 8 professional as Fig. 4 demonstrates.

4 Developed kit Design Steps

The recent trend in the development of the methods of implementation is to make all the devices as compact as possible without affecting the performance of the system. Size reduction is the major interest of the electronic designers. One means of achieving this is to come up with an optimal design layout for all the chips on a circuit board once all the connections have been finalized. That would require minimizing the total area of the circuit board and the total wirings while ensuring that certain physical constraints are satisfied.

Creating a schematic of an analog circuit, making simulations on it, and designing a printed circuit board for it are the basic steps that are taken into consideration in any system design. In this regard, printed circuit boards (PCBs) are ubiquitous [6]. PCB is a thin board made of fiberglass, composite epoxy, or other laminate material. Conductive pathways are etched or printed onto board, connecting different electronic components on the PCB. These tools are the backbones of almost every electronic device, and therefore, PCB design and manufacturing are extremely important components of many industrial production processes. Before a PCB can serve its task, it evolves through three main steps. The first one is the logic design which defines the components to be used and their interconnections. The second step is the physical layout of the PCB where the geometric positions of the components and their physical connections are decided. The final step is the industrial production of the PCB. Using Proteus IDE, we designed the PCB layout of the main kit microcontroller circuit as illustrated in Figs. 5-6. The layout of the loads "output kit" is depicted in Fig. 7. By applying the PCB layout procedure on the "output kit", we got the printed circuit board shown in Fig. 8. After that, we programmed the microcontroller using pickit3 connected to the microcontroller via a microchip debug (RJ-11) connector and connected the components to the board. Driving the DC motor left and right is achieved by pressing switch 1 and switch 2, respectively, as Fig. 9 clarifies.

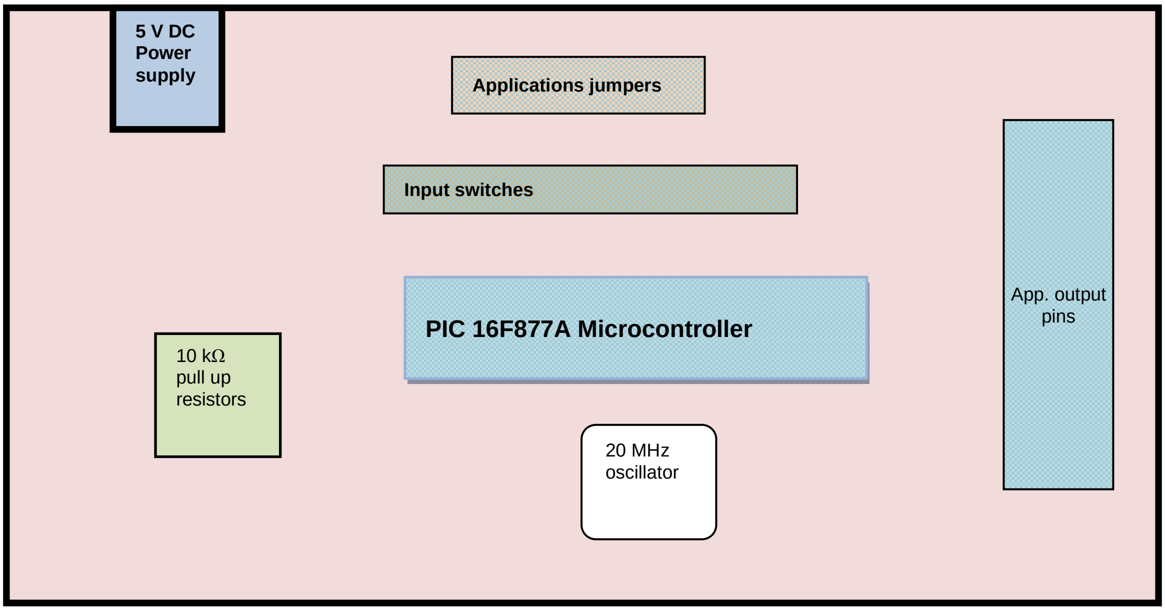


Fig. 2. Main kit PIC 16F877A and inputs

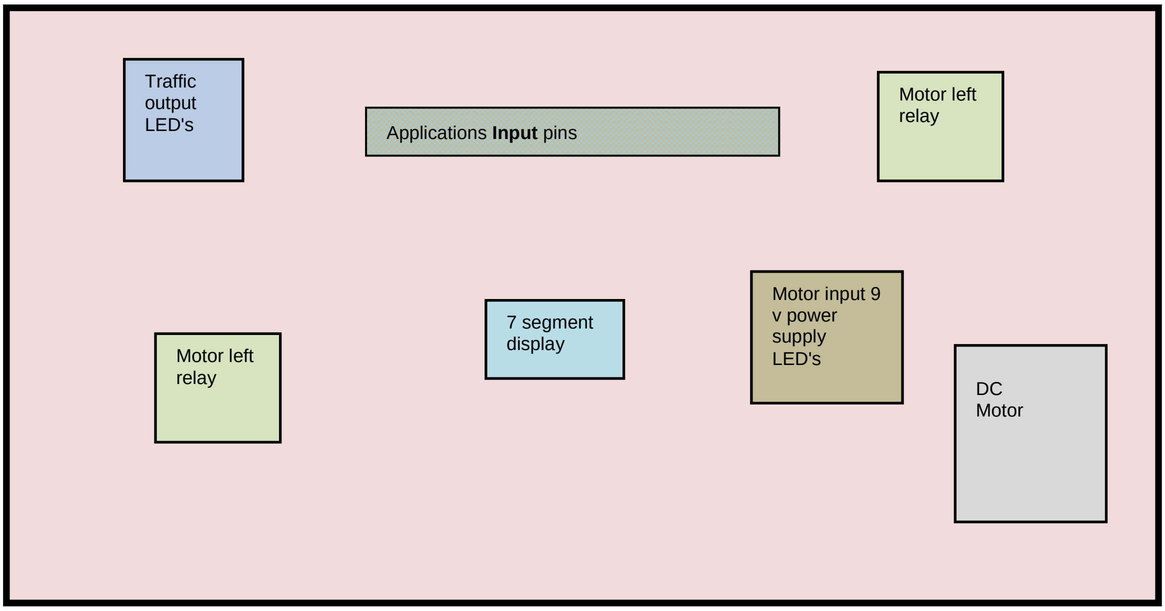


Fig. 3. Output applications

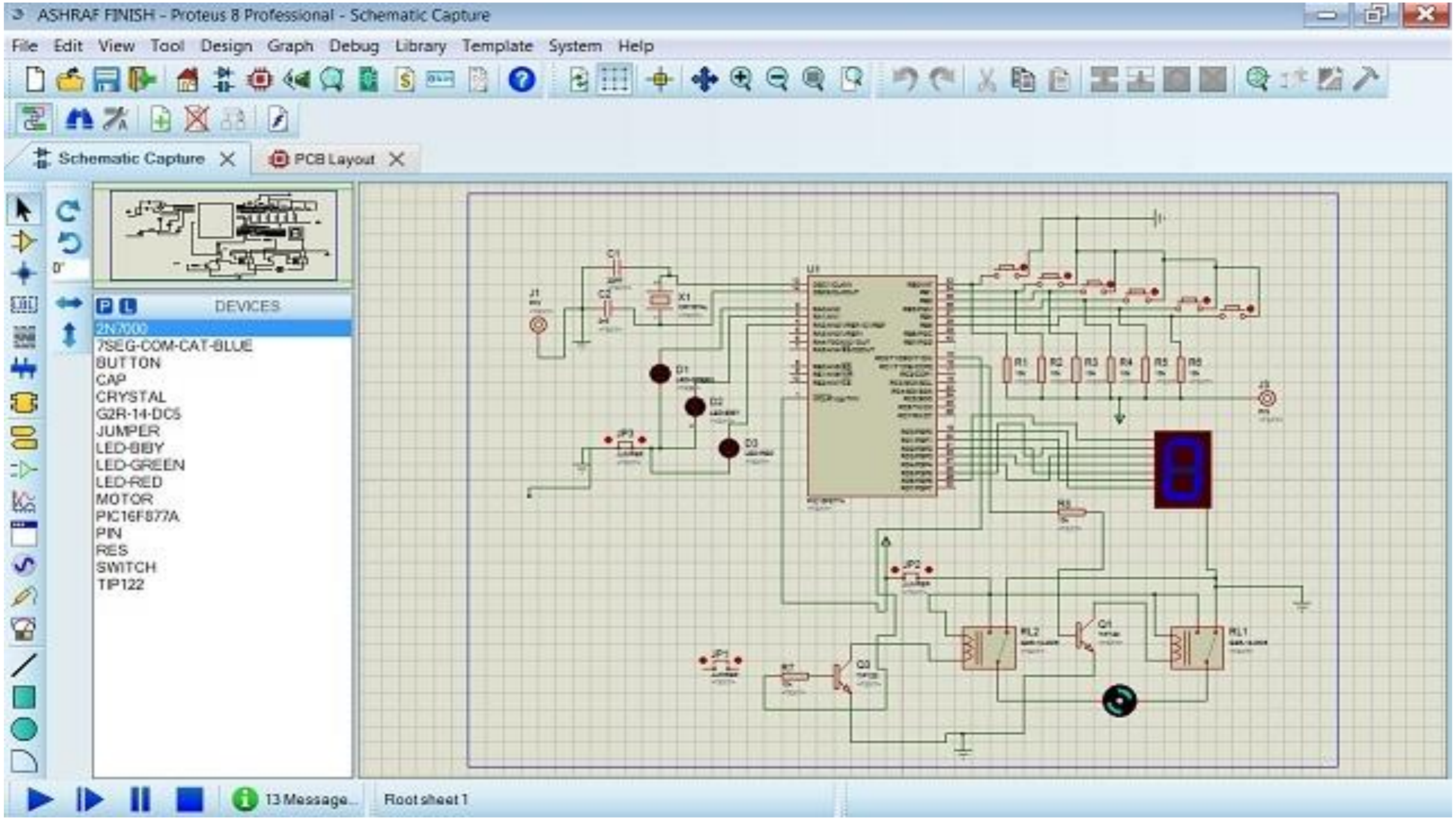


Fig. 4. Circuit design using proteus 8

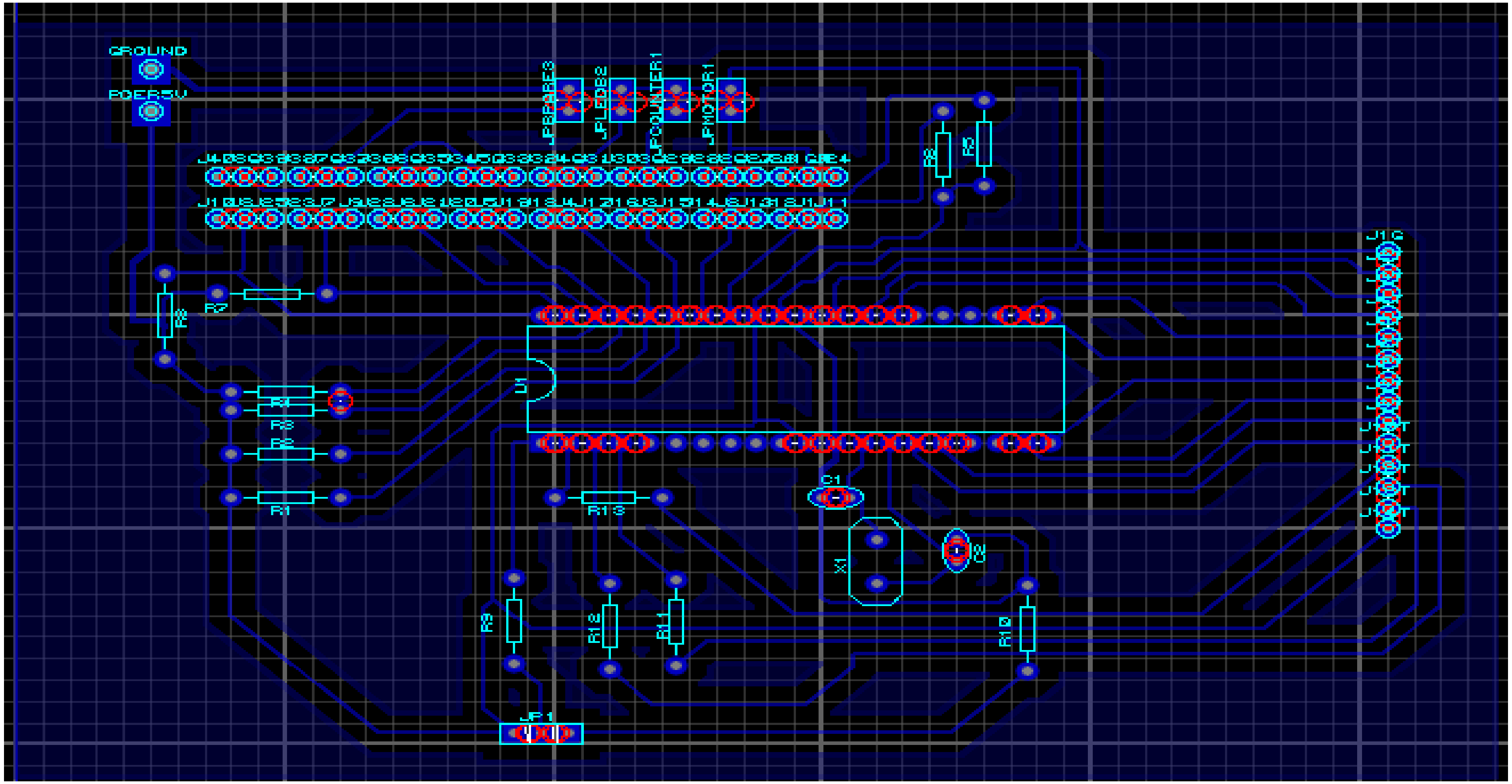


Fig. 5. PCB layout of the main kit

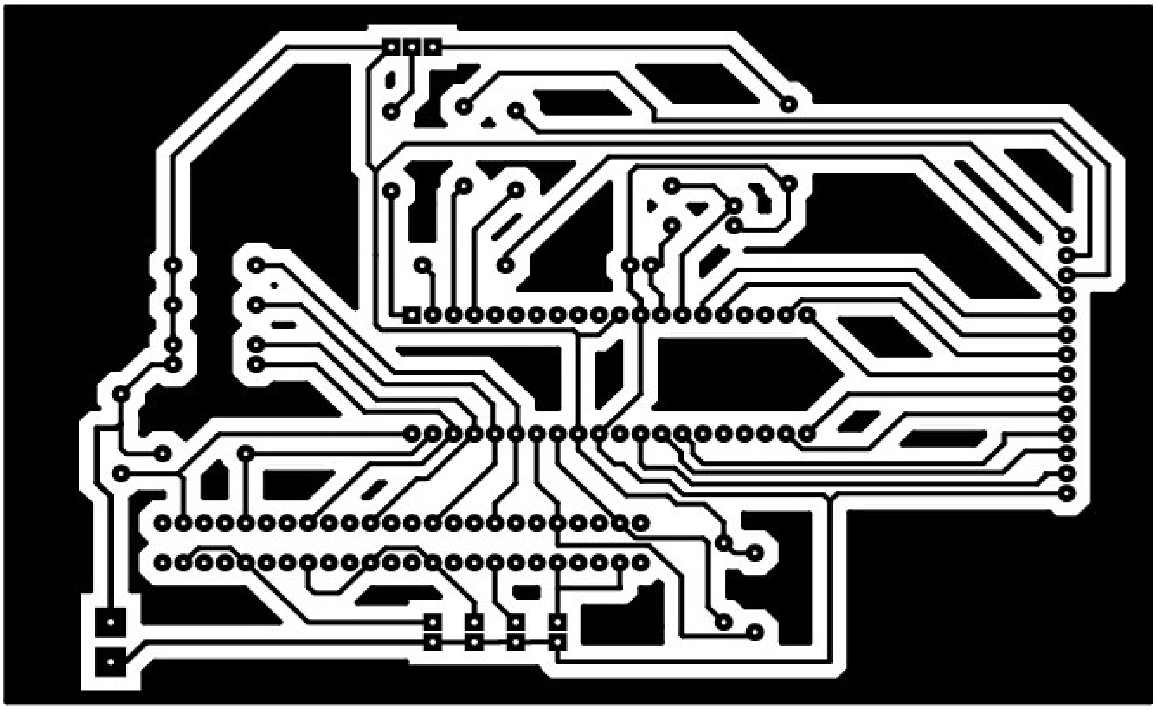


Fig. 6. Layout of the main kit

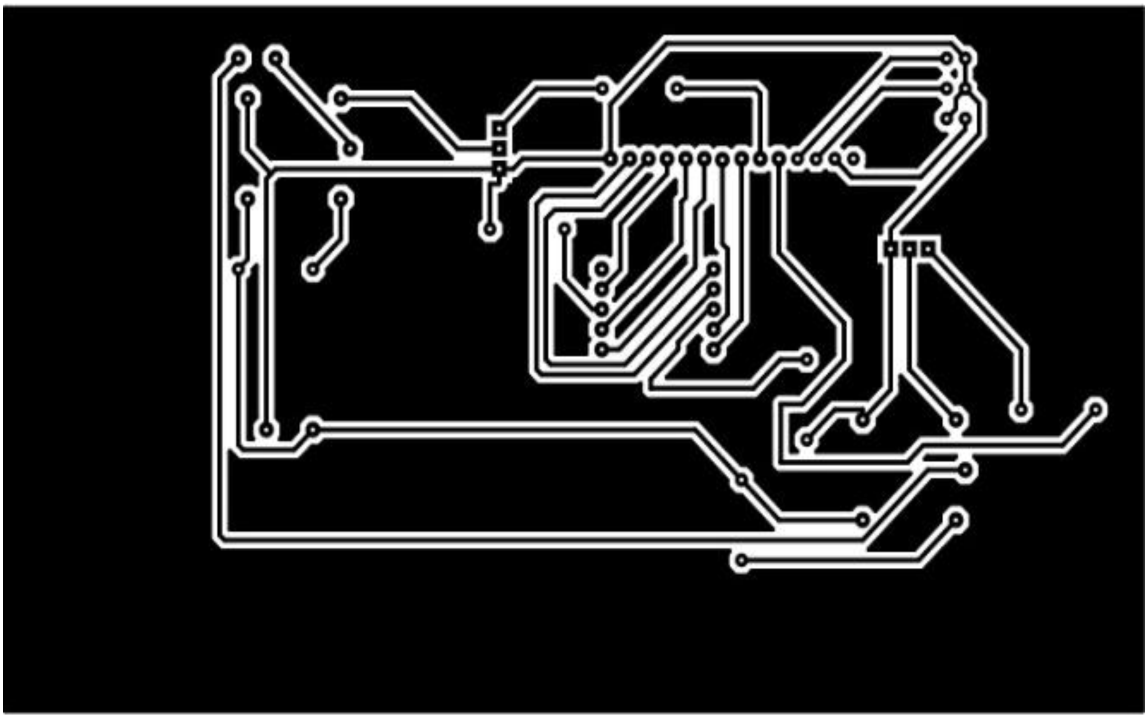


Fig. 7. Layout of output kit

Switch 1

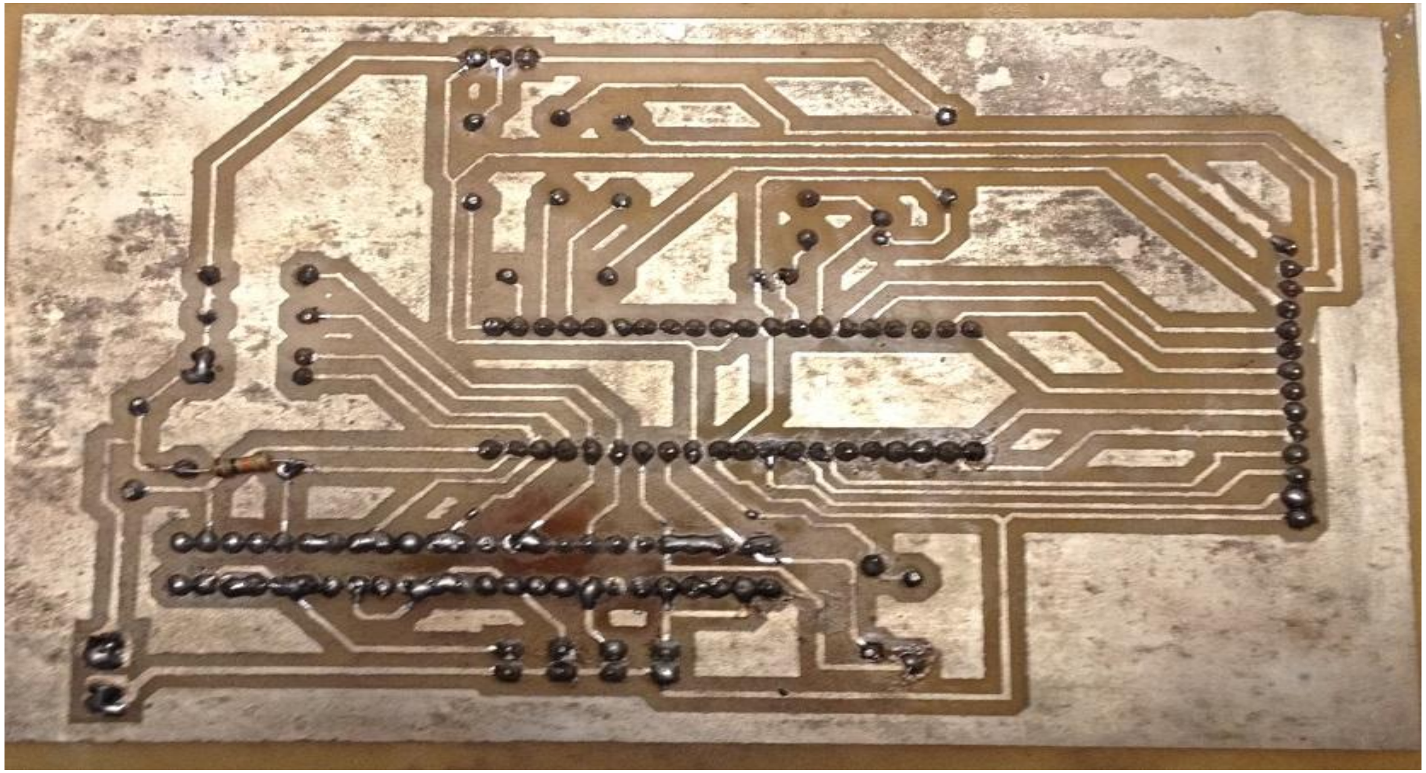


Fig. 8. Printed circuit board

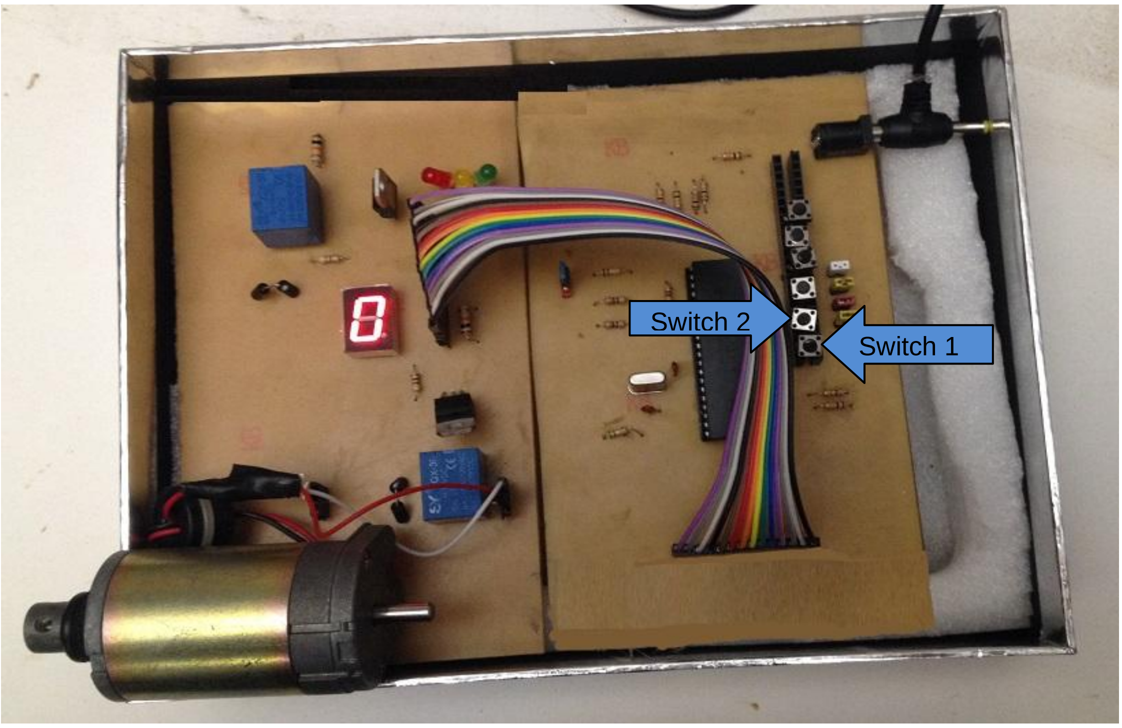


Fig. 9. Motor drive application

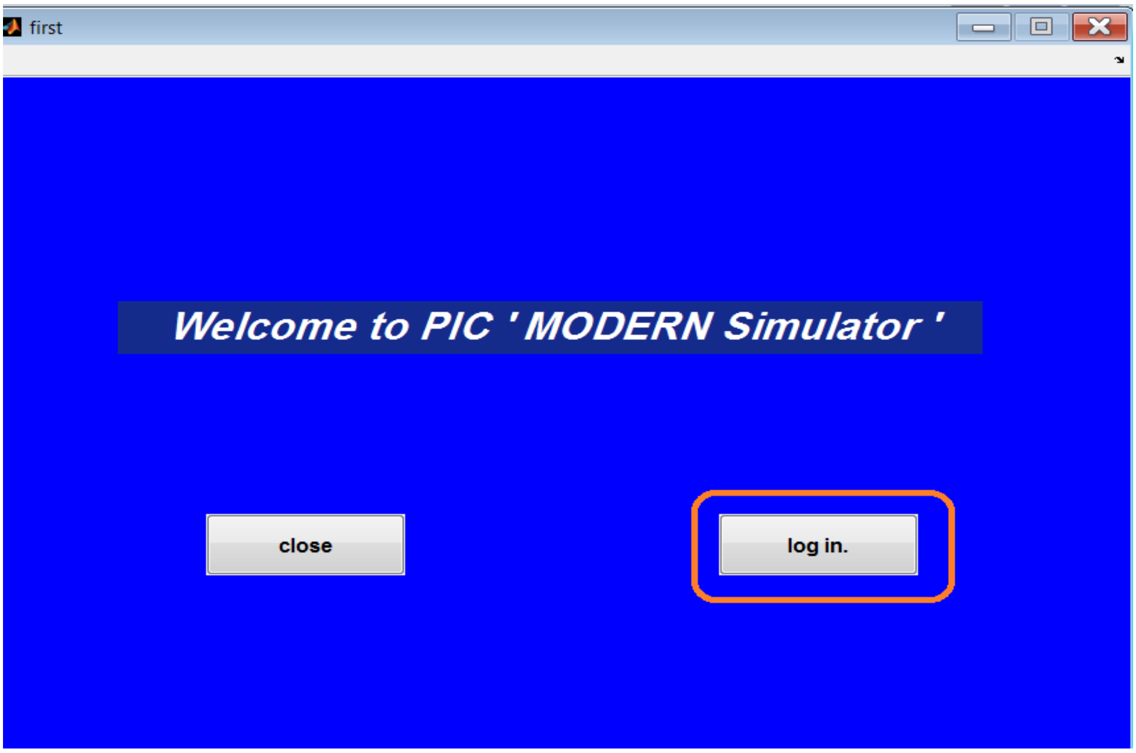


Fig. 10. Select "log in" button as input

In looking to understand the best solution for gaining the engineering knowledge practically, it is necessary to have knowledge of simulation [7]. The purpose is to use electronics simulation tool such as Proteus VSM for engineering education program. Proteus's VSM has microcontroller programming tool, environment, with its many software features and hardware options. It is already unique in its capability to run near real-time simulations of complete microcontroller systems; its real power comes from its ability to perform these simulations in single step mode. It is equipped with comprehensive diagnostic or trace messaging. This allows one to specify which components or processor peripherals are of interest at any given time and receive detailed textual reporting of all activity and system interaction. This is invaluable as a debugging aid, allowing one to locate and fix problems in both software and hardware much faster than one could when working on a physical prototype. Additionally, it provides extensive debugging facilities including breakpoints, single stepping, and variable display for both assembly code and high level language source. Therefore, The Proteus VSM simulation environment is preferred since it enables direct implementation of the compiled program file to the microcontroller both for simulation and hardware model, thus reducing hardware development time.

To simulate the applied program, we designed "MODERN Simulator" using Matlab IDE application and select log in button as exhibited in Fig. 10. Then, we can either click a high level language (micro-C) using "LOAD H.L.L File" button, as Fig. 11 displays, or click hex file using "Load HEX File". MPLAB program will be opened and micro-C program file, "project C", is loaded, as shown in Fig. 12. The resulting program is converted to hex file "project.hex" as Fig. 13 displays.

Simulator output window containing the address, data and assembly instruction is poped up to help the user to understand the flow of each instruction in the program as presented in Fig. 14. "Modern simulator" output virtual load representing the DC motor clockwise direction is demonstrated in Fig. 15.

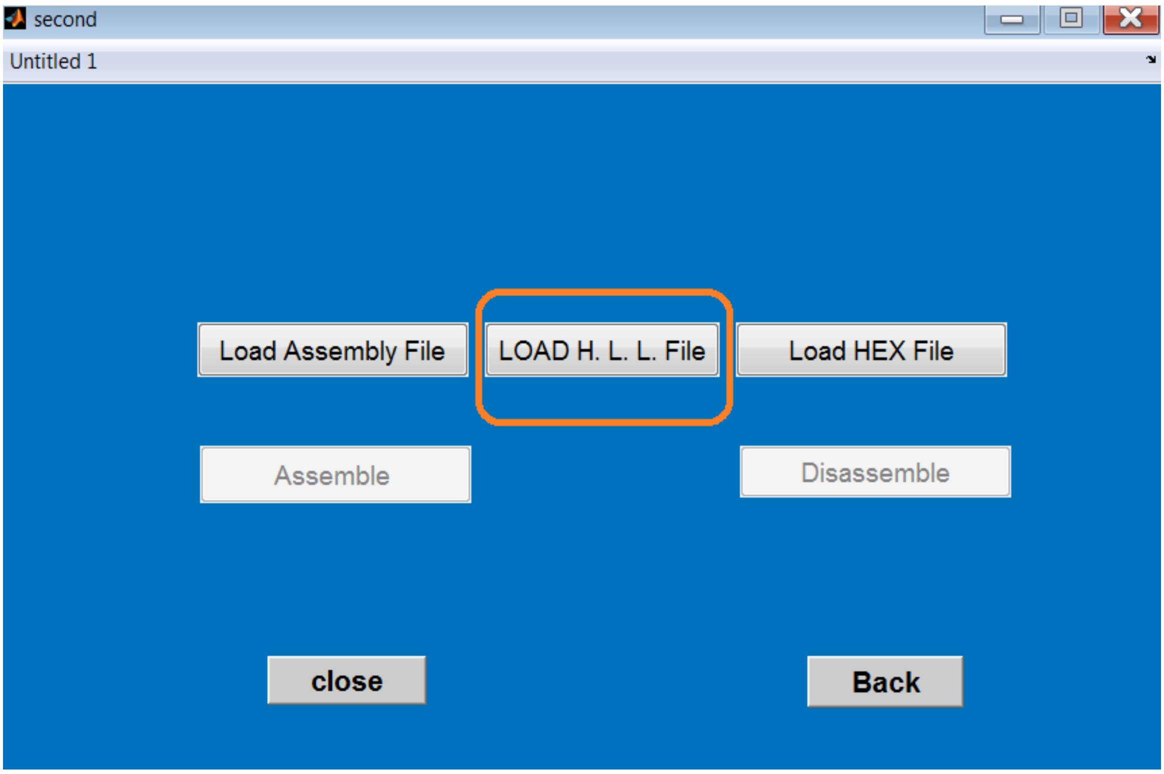


Fig. 11. Click using "LOAD H. L. L." file button

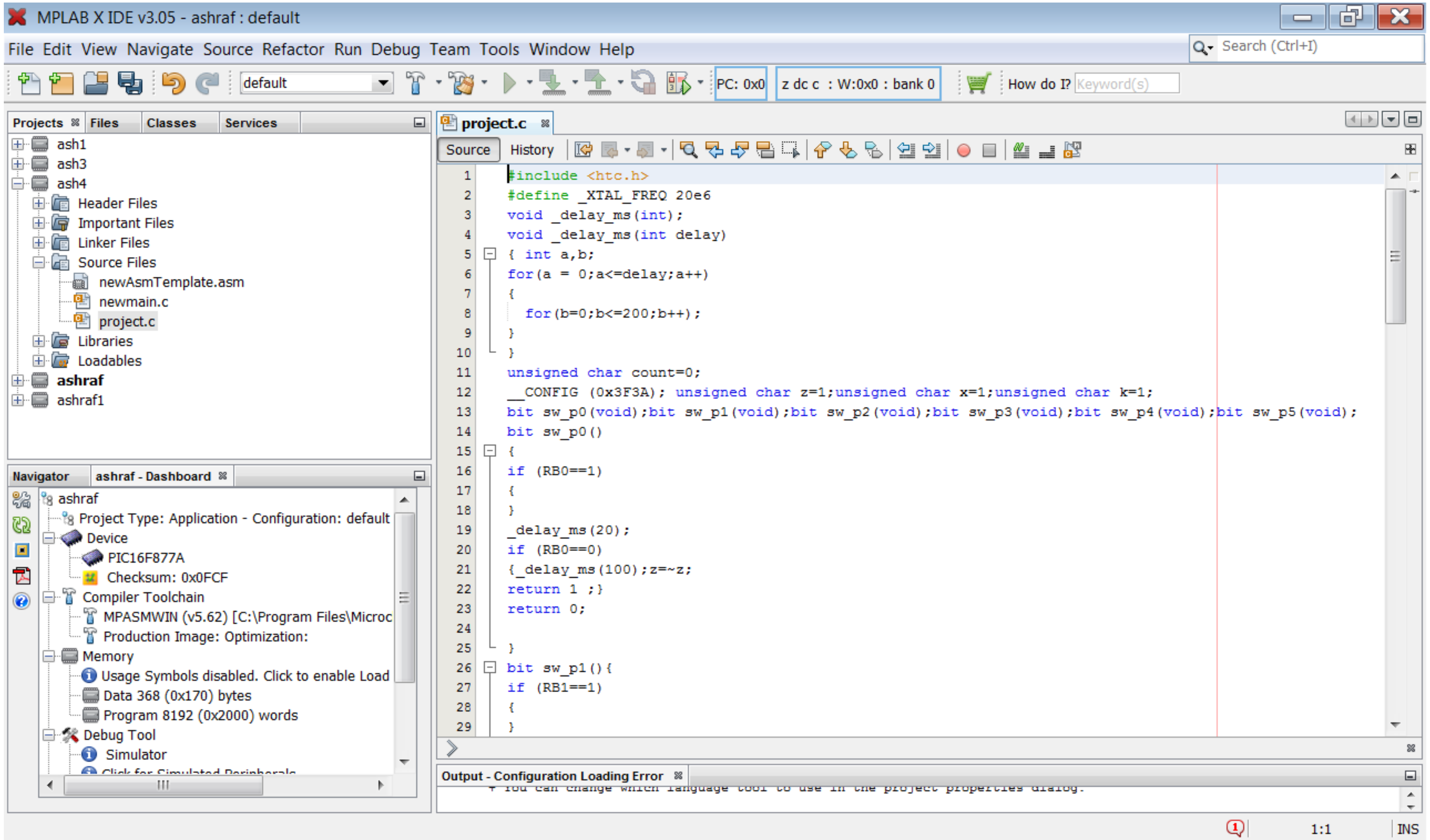


Fig. 12. File "Project. C" is loaded by MPLAP IDE program

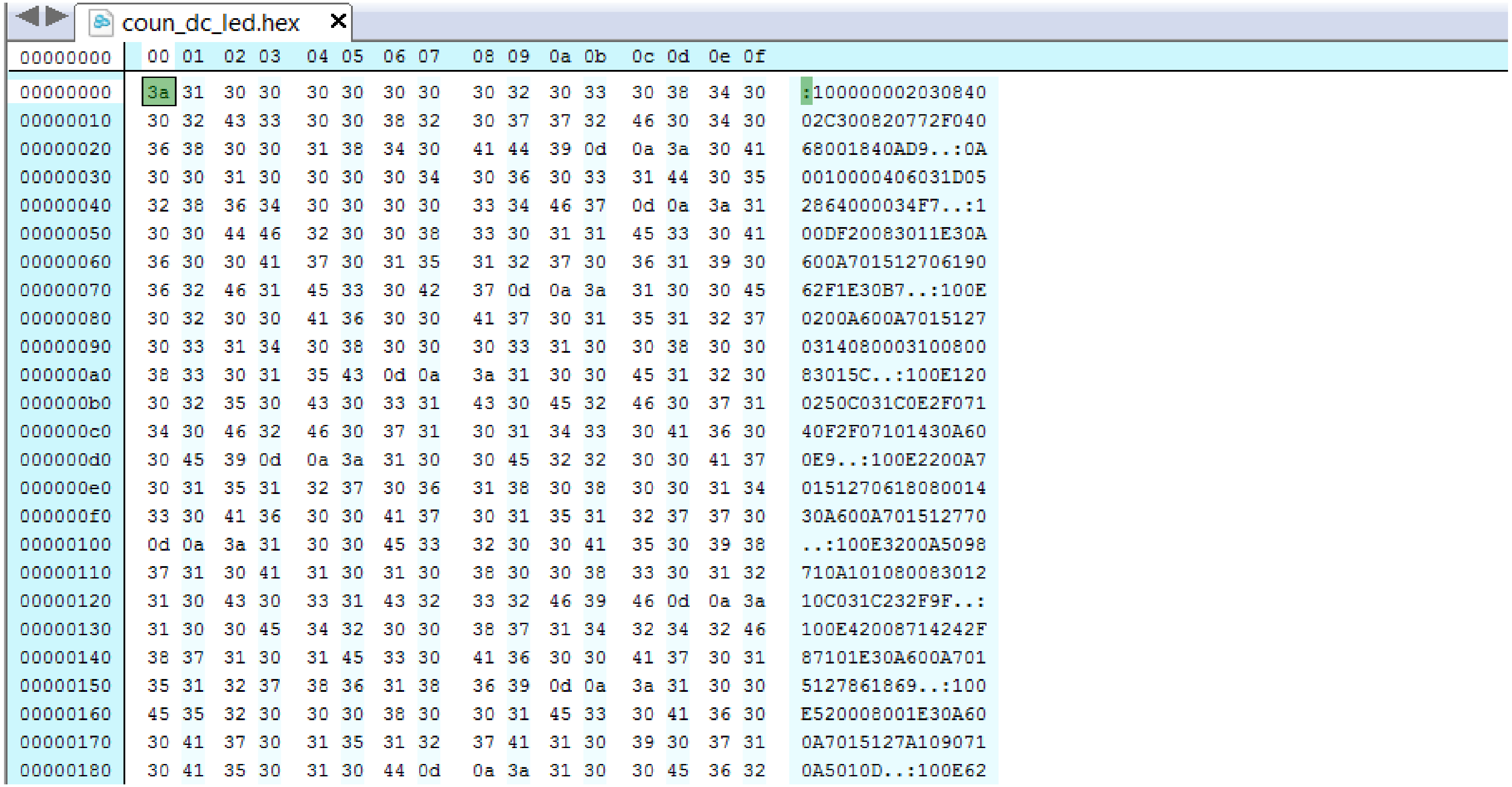
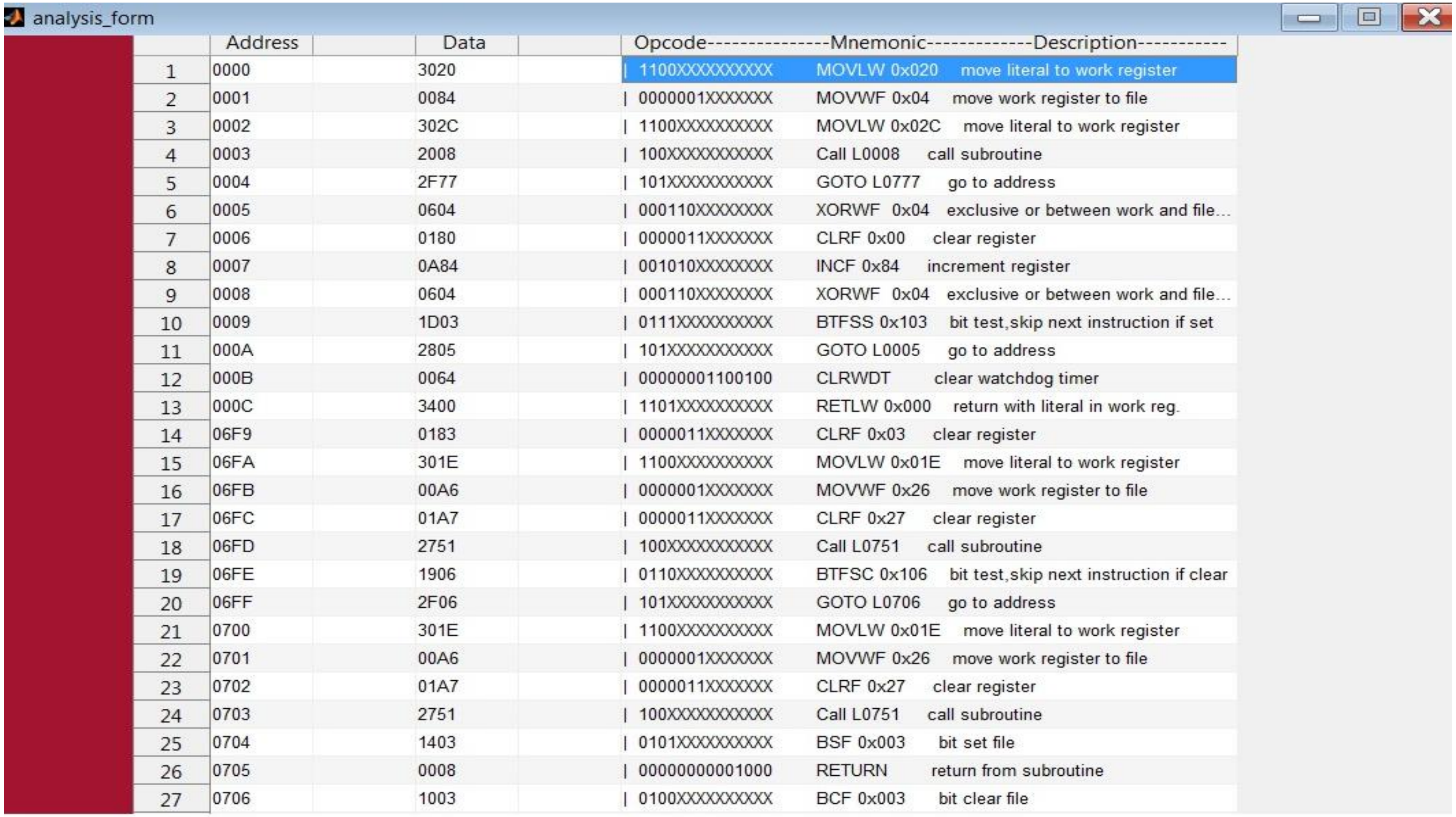


Fig. 13. Hexadecimal file



analysis_form

	Address	Data	Opcode	Mnemonic	Description
1	0000	3020	1100XXXXXXXXXX	MOVLW 0x020	move literal to work register
2	0001	0084	0000001XXXXXXX	MOVWF 0x04	move work register to file
3	0002	302C	1100XXXXXXXXXX	MOVLW 0x02C	move literal to work register
4	0003	2008	100XXXXXXXXXXX	Call L0008	call subroutine
5	0004	2F77	101XXXXXXXXXXX	GOTO L0777	go to address
6	0005	0604	000110XXXXXXXX	XORWF 0x04	exclusive or between work and file...
7	0006	0180	0000011XXXXXXX	CLRF 0x00	clear register
8	0007	0A84	001010XXXXXXXX	INCF 0x84	increment register
9	0008	0604	000110XXXXXXXX	XORWF 0x04	exclusive or between work and file...
10	0009	1D03	0111XXXXXXXXXX	BTFSS 0x103	bit test,skip next instruction if set
11	000A	2805	101XXXXXXXXXXX	GOTO L0005	go to address
12	000B	0064	00000001100100	CLRWDT	clear watchdog timer
13	000C	3400	1101XXXXXXXXXX	RETLW 0x000	return with literal in work reg.
14	06F9	0183	0000011XXXXXXX	CLRF 0x03	clear register
15	06FA	301E	1100XXXXXXXXXX	MOVLW 0x01E	move literal to work register
16	06FB	00A6	0000001XXXXXXX	MOVWF 0x26	move work register to file
17	06FC	01A7	0000011XXXXXXX	CLRF 0x27	clear register
18	06FD	2751	100XXXXXXXXXXX	Call L0751	call subroutine
19	06FE	1906	0110XXXXXXXXXX	BTFSC 0x106	bit test,skip next instruction if clear
20	06FF	2F06	101XXXXXXXXXXX	GOTO L0706	go to address
21	0700	301E	1100XXXXXXXXXX	MOVLW 0x01E	move literal to work register
22	0701	00A6	0000001XXXXXXX	MOVWF 0x26	move work register to file
23	0702	01A7	0000011XXXXXXX	CLRF 0x27	clear register
24	0703	2751	100XXXXXXXXXXX	Call L0751	call subroutine
25	0704	1403	0101XXXXXXXXXX	BSF 0x003	bit set file
26	0705	0008	00000000001000	RETURN	return from subroutine
27	0706	1003	0100XXXXXXXXXX	BCF 0x003	bit clear file

Fig. 14. Simulator output's window containing the address, data and assembly instruction

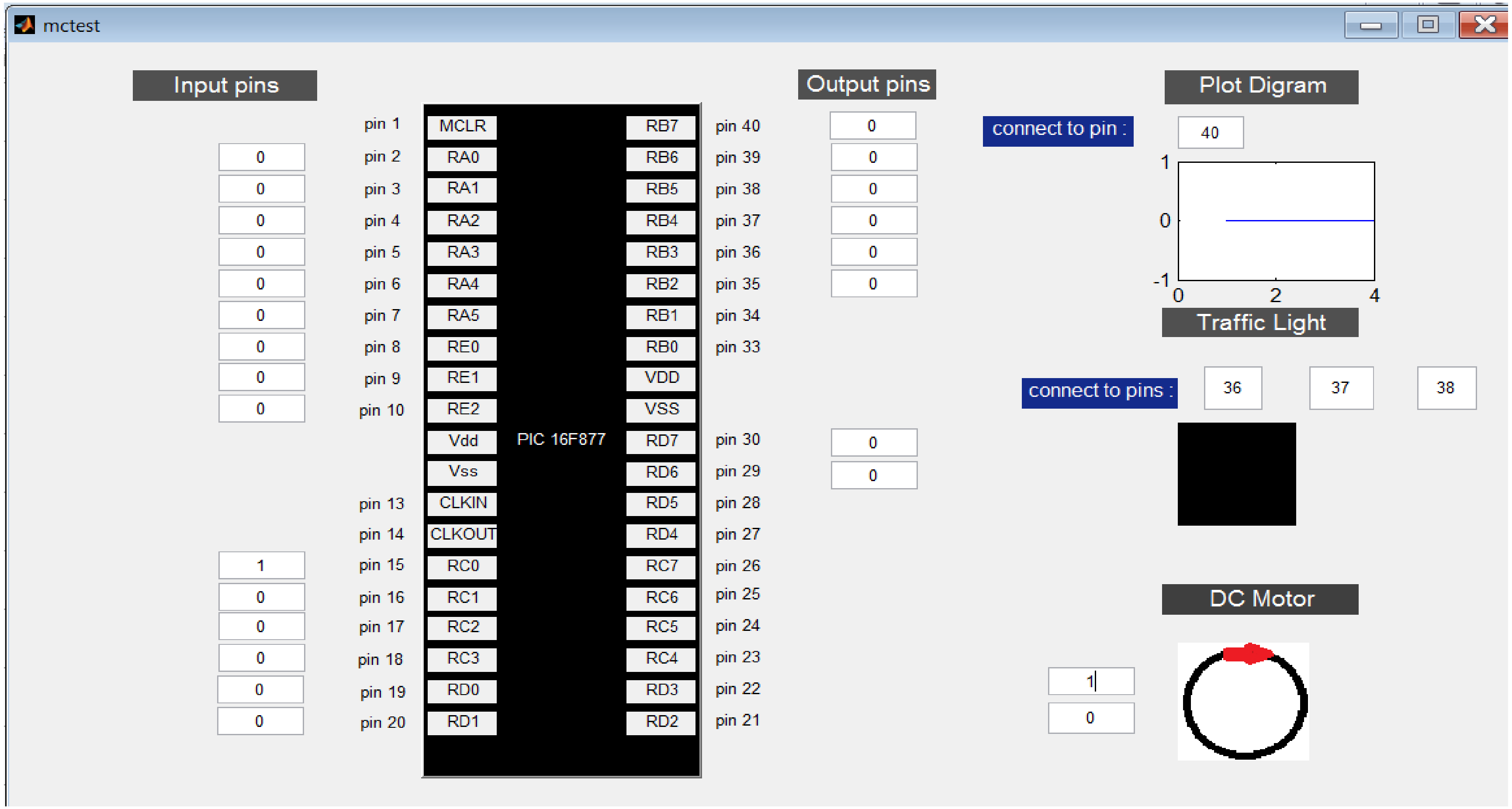


Fig. 15. Modern simulator's output virtual load representing DC-motor clockwise direction

5 Conclusions

The paper introduces an educational friendly used kit based on PIC16F877A microcontroller accompanied with a developed "Modern simulator". Some hardware applications are applied and tested to operate correctly, without error, with the aids of the designed kit and the developed "Modern Simulator" using Matlab program. The developed simulator is efficiently provided by H.L.L loading of user application files, and then converted it into H-file. The file had been correctly tested to prevent any logical error introduced via the so-called testing, validation and verification process of the PIC16F877A microcontroller before the programming (burning) process of the chip (firmware). Different output windows of the developed simulator can depict the RAM address, data and assembly instruction, its op-code, and description of each instruction in such a way that the user can understand, keep track and follow the program easily. Moreover, some of the practical results are displayed using different virtual load window including; seven-segment displaying the count of the counter, LED's representing the traffic lights, and DC motor orientation (clockwise and anticlockwise).

Although PIC16-series microcontrollers are excellent general purpose microcontrollers, they have certain limitations. For example, the program and data memory capacities are limited, the stack is small, and the interrupt structure is primitive, all interrupt sources sharing the same interrupt vector. These series of microcontrollers also do not provide direct support for advanced peripheral interfaces and interfacing with such devices is not easy. The instruction set for these microcontrollers is also limited. For example, there are no multiplication or division instructions, and branching is rather simple, being a combination of skip and goto instructions. Microchip has developed the PIC18 series of microcontrollers for use in high-pin count, high-density, and complex applications. The PIC18F microcontrollers offer cost efficient solutions for general purpose applications written in C that use a real-time operating system and require a complex communication protocol stack. PIC18F devices provide flash program memory in sizes from 8 to 128Kbytes and data memory from 256 to 4Kbytes, operating at a range of 2.0 to 5.0 volts, at speeds from DC to 40MHz.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] John Morton. The PIC microcontroller: Your personal introductory course. Third Edition, Newnes; 2005.

[2] Martin P. Bates. Interfacing PIC microcontrollers: Embedded design by interactive simulation. Newnes; 2013.

[3] Mohamed B. El_Mashade, Abdelmonem Fouda, Ashraf M. Etman. MATLAB toolbox establishment of PIC microcontroller simulator. International Journal of Science and Technology (IJST). 2016;5(10):497-506.

[4] Dogan Ibrahim. Advanced PIC microcontroller projects in C; 2008.

[5] Smith DW. PIC in practice. Newnes, Oxford, U.K.; 2006.

[6] MPASM-assembler user's guide. Microchip technology Inc.; 2005.

[7] Intel. Hexadecimal Object File Format Specification; 2004.

[8] Caterina De Vito, Vincenzo Ferrini, Silvano Mignardi, Marco Cagnetti, Fabio Leccese. Progress in carbon dioxide sequestration via carbonation of aqueous saline wastes. Periodico di Mineralogia. 2012;81(3):333-344.

[9] Fabio Leccese. A cheap Wi-fi instrumentation remote control platform for electric and electronic didactic laboratory. 17th Symposium IMEKO TC4, Sept. 8-10, Kosice, Slovakia. 2010;129-133.

[10] Fabio Leccese, Marco Cagnetti, Andrea Calogero, Daniele Trinca, Stefano di Pasquale, Sabino Giarnetti, Lorenzo Cozzella. A new acquisition and imaging system for environmental measurements: An experience on the Italian cultural heritage. Sensors. 2014;14:9290-9312.

[11] Fabio Leccese, Marco Cagnetti, Andrea Ferrone, Alessandro Pecora, Luca Maiolo. An infrared sensor tx/rx electronic card for aerospace applications. IEEE Metrology for Aerospace (MetroAeroSpace); 2014.

[12] Fabio Leccese. Remote control system of high efficiency and intelligent street lighting using a ZigBee Network of Devices and Sensors. IEEE Transactions. 2013;28(1):21–28.

[13] Fabio Leccese, Marco Cagnetti, Daniele Trinca. A smart city application: A fully controlled street lighting isle based on raspberry-Pi card, a ZigBee sensor Network and WiMAX. Sensors. 2014;14:24408-24424.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/18354