UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

PROJECT: MICROPROCESSOR CONTROLLED DIGITAL LOCK

PROJECT INDEX: 104

NAME: MBUGUA ERIC NGURE
REG. NO: F17/29174/2009

SUPERVISOR: PROF. E. MWANGI
EXAMINER: MR. S. A. AHMED

A project report submitted to the Department of Electrical and Information Engineering in partial fulfilment of the requirements of BSc. Electrical and Electronic Eng. of the University of Nairobi

Submitted on 28th April 2014

DECLARATION OF ORIGINALITY

FACULTY/ SCHOOL/ INSTITUTE: Engineering

DEPARTMENT: Electrical and Information Engineering

COURSE NAME: Bachelor of Science in Electrical & Electronic Engineering

TITLE OF NAME OF STUDENT: Mbugua Eric Ngure

REGISTRATION NUMBER: F17/29174/2009

COLLEGE: Architecture and Engineering

WORK: Digital lock

1. I understand what plagiarism is and I am aware of the university policy in this regard.
2. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.
3. I have not sought or used the services of any professional agencies to produce this work.
4. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
5. I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

Signature: ……………………………………………………………………………..…

Date: ………………………………………………………………………………………

DEDICATION

This project is dedicated to my parents for the moral and financial support and also to those who have guided me throughout my journey of education.

ACKNOWLEDGEMENTS

First and foremost, I thank the almighty God for guiding me throughout my studies till the accomplishment of this project.

I am grateful to my supervisor, Prof. E Mwangi for the useful guidance and suggestions throughout the project, it has been a great pleasure for me to get an opportunity to work under him.

A project of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in reference section. I also acknowledge all of them.

Lastly, my thanks goes to all of my lecturers, classmates and friends in the department of Electrical and Electronic Engineering at the University of Nairobi who extended all sorts of help to the accomplishment of this undertaking.

Table of Content

List of figures

List of Acronyms

ALU –	Arithmetic logic unit
RISC -	Reduced Instruction Set Computer
CISC -	Complex instruction set computer
CPU-	Control processing unit
PIC -	Peripheral Interface Controller
SFR-	special function register
ROM -	Read Only Memory
RAM -	Random Access Memory
LCD -	Liquid Crystal Display
LED -	Light emitting diode
MOS -	Metal oxide semiconductor
EMF –	Electromotive force
SIM –	Subscriber identification module
PCB –	Printed circuit board

ABSTRACT

The intent of this paper is to design an efficient, effective and low cost microcontroller security system that could be used to realize a digital locking mechanism with safety overrides. The user enters a code through a keypad within specified time duration and the system accepts or rejects the request. For more than a specified number of attempts exceeded, the system should lock permanently and send an alert for an attempted entry. A master unlock code is provided to the system administrator to restore the control mechanism in case of an auto-lock due to failed attempts.

Keywords (microcontroller security system, digital locking mechanism, control mechanism)

CHAPTER 1: INTRODUCTION

Security describes protection of life and property. In this modern age, security has becomes an important issue in the society. This sector is experiencing a huge diversification. Where the reliability of already existing systems has been looked into for the possibility of creating better systems that are smarter and more secure. Here is where an embedded microcontroller system comes into the picture, to realize a digital locking mechanism with an ability that allows users to input numerical password to open locks. When the user enters the right numerical password, a relay is turned on which in turn powers an actuator to open a door lock. The significance of this idea over the old mechanical key locking system is that it eliminates the need for a physical key.

Other digital locks have been implemented using different modes of inputs where they build more on the basis of this project such systems include the digital safe where one rotates a knob clockwise and anticlockwise while entering the code this is mainly used in safes, there is also the use of radio frequency tag which has been indulged in the modern day hotels also we have digital locks which can be assessed remotely i.e. via mobile phones or from a control room by use of computers [1]. Another interesting mode of input used in security systems is the use of biometric devices which authenticate user's access by a personal identifiers which may be fingerprint scan, voice scan, iris scan, retina scan, or facial scan. On the fingerprint scan biometric there has been an interesting development where the users' finger to be scanned should have blood flow in it, to avoid security hacks by use of chopped off users fingers or dummy fingers [2].

1.1 PROBLEM DEFINITION/STATEMENT

Design a microcontroller system that could be used to realize a digital locking mechanism with safety overrides. The user enters a code through a keypad within specified time duration and the system accepts or rejects the request. For more than a specified number of attempts which when exceeded system should lock permanently and send an alert for an attempted entry. The system designer then provides a master unlock code to restore the mechanism.

1.2 PROJECT OBJECTIVE

The objectives of this project can be divided into three parts

- To design and construct a microcontroller system that could be used to realise a digital locking mechanism.
- To enter a code through a keypad within specified time duration and the system accepts or rejects the request.
- To have a specified number of attempts which when exceeded system should lock permanently and send an alert for an attempted entry and The system designer then provides a master unlock code to restore the mechanism

1.3 PROJECT SCOPE

This project entails the following:

- Understanding microcontroller's structure and functionality.
- Understanding the PIC16F690 internal components specifically the EEPROM and timers and counters.
- Interfacing microcontrollers with input/output devices specifically the keypad, Liquid crystal display, buzzer and relay.

Developing a microcontroller program or code using assembly language.

CHAPTER 2: LITERATURE REVIEW

This chapter describes and discusses the research from various sources such as textbooks, articles and the internet. It consists of information which is vital in the development of this project.

2.1 MICROCONTROLLER

A microcontroller is a small computer furnished in a single integrated circuit that needs support chips. Its principal nature is self-sufficiency and low cost. It is not intended to be used as a computing device in the conventional sense; that is, a microcontroller is not designed to be a data processing machine, but rather an intelligent core for a specialized dedicated system.

Microcontrollers are embedded in many control, monitoring, and processing systems. Some are general-purpose devices but most microcontrollers are used in specialized systems such as washing machines, telephones, microwave ovens, automobiles, and weapons. A microcontroller usually includes a central processor, input and output ports, memory for program and data storage, an internal clock, and one or more peripheral devices such as timers, counters, analogue-to-digital converters, serial communication facilities, and watchdog circuits [3].

2.1.1 Types of microcontrollers

Microcontrollers can be classified according to the following:

- Internal bus width
- Instruction set
- Memory architecture
- Embedded and external memory microcontroller
- Integrated circuits (IC) chip or very large scale integration (VLSI) core, very high speed integrated circuits (VHSIC), hardware description language (VHDL or Verilog) file
- Family.

2.1.1.1 Internal Bus Width

They are classified to 8-bit, 16-bit and 32-bit microcontrollers.

In an 8-bit microcontroller it contains an 8-bit bus and the arithmetic logic unit (ALU) performs the arithmetic and logic operations on a byte at an instruction. Examples include PICx, Intel 8031/8051 and Motorola MC68HC11 families [4].

In 16-bit microcontroller, contains a 16-bit bus and the arithmetic logic unit (ALU) performs arithmetic and logic operations on the operand words of 16 bits at the instructions. These microcontrollers have a greater advantage than 8-bit microcontrollers in that they provide better performance and precision than their 8-bit counter parts. Typical examples include Intel 8096, PIC2x, Extended 8051XA, Atmega 16 and Motorola MC68HC12 families [4].

In 32-bit microcontrollers, the internal bus for the data transfer operations in this microcontroller is 32-bit. The ALU performs arithmetic and logic operations on operand words of 32 bits at the instructions. They give better performance and precision as compared to 16-bit microcontrollers but are more expensive. Some of the most popular 32-bit microcontrollers are PIC3x, Atmega 32, Intel/Atmel 251 family, Motorola M683xx and ARM 7, 9 or 11 processor based families. They find applications in embedded computing systems for applications-MPEG processing, mobile phones, MP3 audio systems, image processing based products and aerospace systems [4].

2.1.1.2 Instruction Set

This includes the following:

- ❖ Complex instruction set computer (CISC)
- ❖ Reduced instruction set computer (RISC)

A RISC machine contains few instructions not more than 35 and each instruction performs more elementary operations. Consequences of this are a smaller silicon area, faster execution, reduced program size with fewer accesses to main memory and less power dissipation. The PIC designers have followed the RISC route [3].

While on the other hand in a CISC (Complex Instruction Set Computer) microcontroller both data and instructions are 8-bits wide, with a minimum of 200 instructions. Data and code are on the same bus and cannot be fetched simultaneously [5].

2.1.1.3 Memory Architecture

This is how exchange of data between the control processing unit (CPU) and memory is carried out. .Memory architecture includes the following:

- ❖ Princeton (Von-Neumann) memory architecture microcontroller
- ❖ Harvard memory architecture microcontroller

Microcontrollers using the von architecture have only one memory block and one data bus. As all data are exchanged by use of one data bus, thus the data bus can be overloaded in applications requiring high computational power hence making it very slow and inefficient. The CPU can either read an instruction or read/write data from/to the memory. Both cannot occur at the same time since the instructions and data use the same bus system [6].

Harvard architecture is a computer design in which data and instruction use different buses and storage areas. In other words, data and instructions are not located in the same memory area but in separate ones. In this type of machine, the processor can read and write instructions and data to and from memory at the same time and this result in a faster, more complex machine [3].

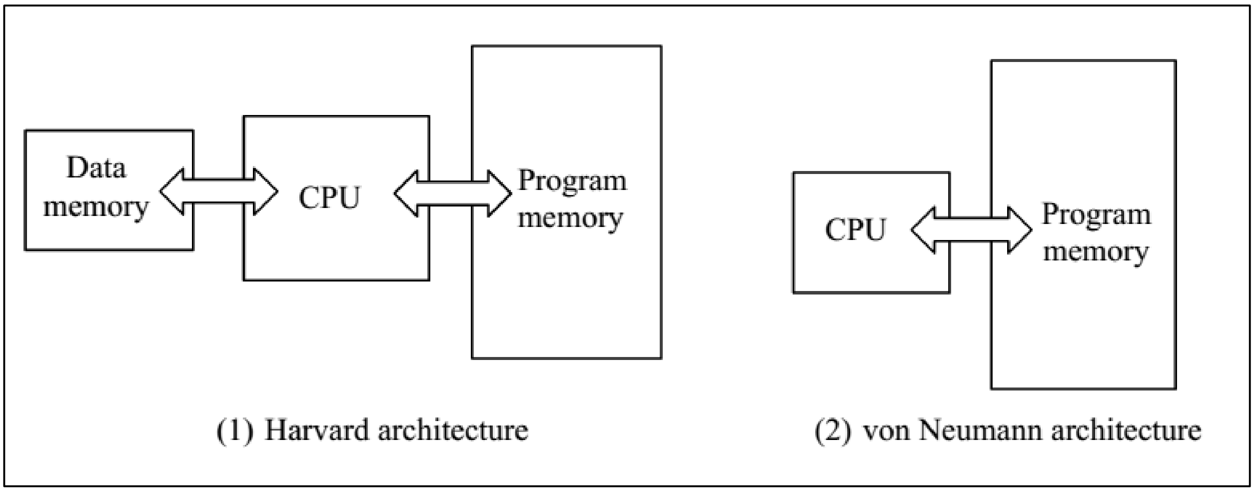


Figure 2.1: Memory architecture

2.1.1.4 Embedded and External Memory Microcontrollers

An embedded microcontroller has all the software and hardware units in a single integrated circuit while an external memory microcontroller does not have all the software and hardware units inside the integrated circuit. It may or may not have the entire memory unit externally interfaced. The interfacing circuit is known as a glue circuit [4].

2.1.2 The mid-range PIC microcontroller core features

PIC microcontrollers are designed by Microchip Technology, the real name for this microcontroller is PICMICRO (Peripheral Interface Controller), but it is better known as PIC. Its first ancestor was designed in 1975 by General Instruments. This chip PIC1650 was meant for a totally different purposes, where ten years after, by adding an EEPROM memory, this circuit was transformed into a real PIC microcontroller [6].

In a microcontroller system, the central processing unit (CPU) is the component that executes the program instructions and processes data. It provides the fundamental functionality of a digital system. In the PIC architecture, it fetches and executes the instructions contained in a program. The arithmetic-logic unit (ALU) is the CPU element that performs arithmetic, bitwise, and logical operations. It also controls the bits in the STATUS register as they are changed by the execution of the various program instructions [3].

PIC microcontrollers are unique in many ways. They have several general Characteristics which include: Harvard architecture, RISC processor design, single-word Instructions, machine and data memory configuration, and characteristic instruction formats [3].

2.1.2.1 Single-word Instructions

Since the device has separate buses for instructions and data, that is the Harvard architecture it is possible for instructions to be sized differently than data items. This ability to vary the number of bits in each instruction op-code makes it possible for the optimization program memory and use of single-word instructions that can be fetched in one bus cycle. The PIC architecture has a two-stage instruction pipeline; however, since the fetch of the current instruction and the execution of the previous one can overlap in time, one complete instruction is fetched and executed at every machine cycle due to pipelining. The one exception is when an instruction modifies the contents of the Program Counter. In this case, a new instruction must be fetched, requiring an additional machine cycle [3].

The PIC clocking system is designed so that an instruction is fetched, decoded, and executed every four clock cycles. In this manner, a PIC equipped with a 4MHz oscillator clock beats at a rate of 0.25 μs. Since each instruction executes at every four clock cycles, hence each instruction takes 1 μs [3].

2.1.2.2 Instruction format

The mid-range family of PICs including the PIC16F690 have 14-bit instructions and a set of 35 instructions. The format for the instructions follows three different pattern

- ❖ Byte-oriented operations
- ❖ Bit-oriented operations
- ❖ Literal and control operations

2.1.2.3 Input/output (I/O) Ports

Microcontroller require to be connected to additional electronics, i.e. peripherals hence, microcontrollers have one or more registers (called "port" in this case) connected to the microcontroller pins, where the Port pins can be configured either as input or output, that is, general ports are bidirectional using specific special function registers [6].

An important feature of I/O pins is the maximum current they can source/sink. For the most microcontrollers, current obtained from one pin is sufficient to activate low current devices (10-20 mA). If a microcontroller has many I/O pins, then the maximum current of one pin is lower [4]. Usually, each I/O port is under control by a special function register (SFR), this means that each bit of that register determines state of the corresponding pin. For example, by writing logic one (1) to one bit of that control register SFR, the appropriate port pin is automatically configured as input. It means that voltage brought to that pin can be read as logic 0 or 1. Otherwise, by writing zero to the SFR, the appropriate port pin is configured as output. Its voltage (0V or 5V) corresponds to the state of the appropriate bit of the port register.

Another important pin feature is to (not) have pull-up resistors. These resistors connect pin to positive power supply voltage and their effect is visible when the pin is configured as input connected to mechanical switches or push buttons. Some versions of microcontrollers have pull-up resistors connected to and disconnected from the pins by software [6].

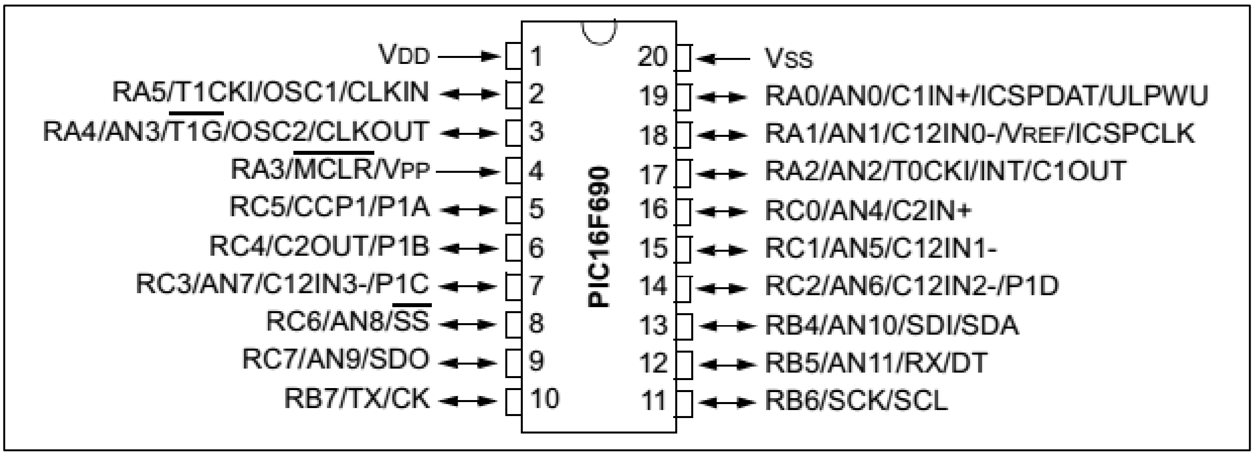


Figure 2.2: PIC16F690 [7]

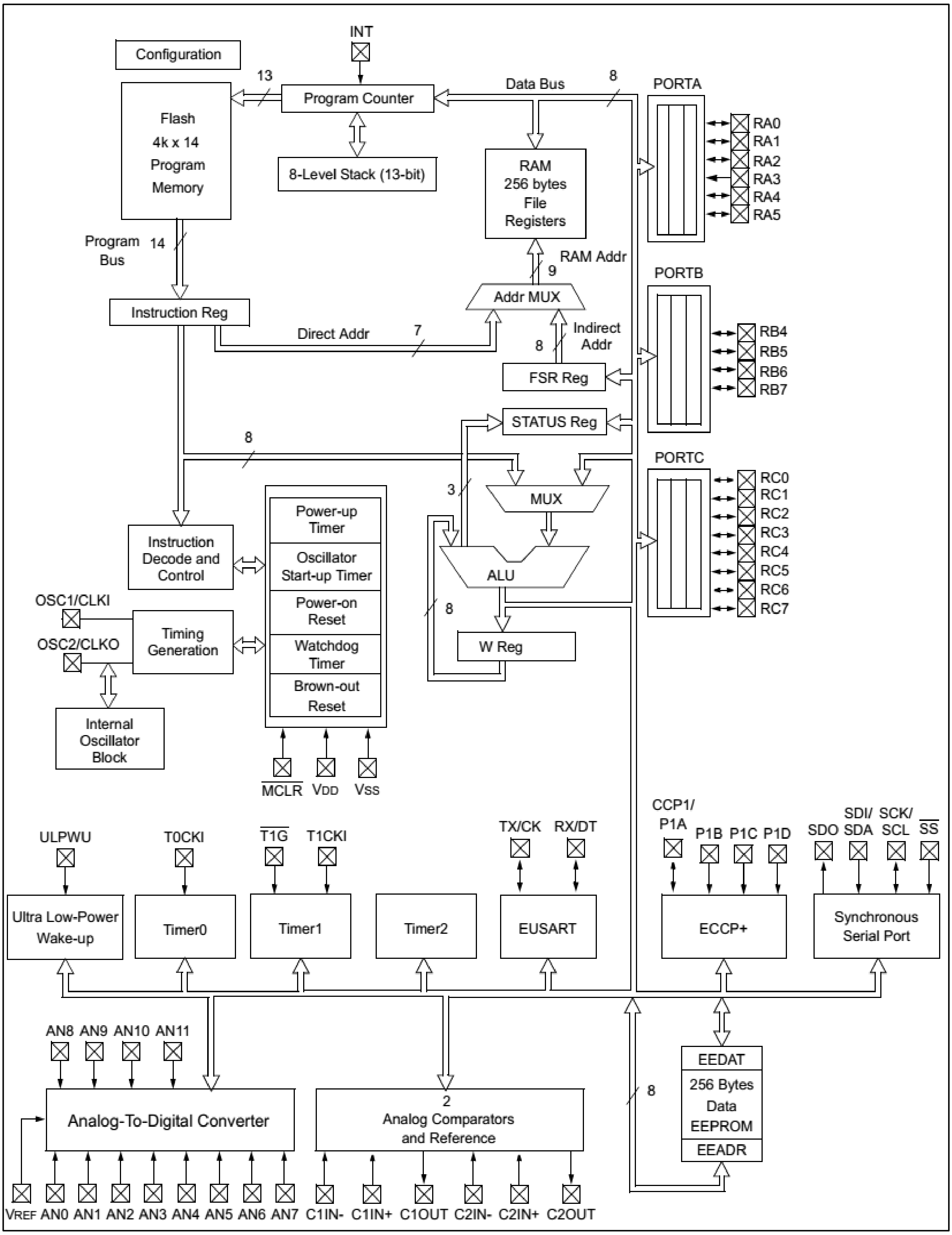


Figure 2.3 PIC16F690 Block diagram [7]

2.1.2.4 Memory unit

Memory is part of the microcontroller used for data storage. A memory address corresponds to one memory location. The content of any location becomes known by addressing. This means that, on one hand it is necessary to select the desired memory location, on the other hand it is necessary to wait for the contents of that location. In addition to read, memory also has to allow writing to these locations [6].

2.1.2.4.1 Read Only Memory (ROM)

This type of memory is not affected by power loss. It is used to permanently save program being executed [4]. The PIC16f690 microcontrollers uses 8-bit addressing, which means that one can be able to address up to 256 bytes memory, i.e. 256 locations [7].

Types of rom;-

- **Masked rom**. Microcontrollers containing this ROM are reserved for the great manufacturers. Program is loaded into the chip by the manufacturer. In case of large scale manufacture, the price is very low [6].
- **One Time Programmable ROM (OTP rom).** If the microcontroller contains this memory, one can download a program into the chip only once [6].
- **Ultra-Violet Erasable Programmable ROM (UV EPROM)** Both manufacturing process and characteristics of this memory are completely identical to OTP ROM. However, the package of this microcontroller has recognizable “window” on the upper side, which enables the surface of the silicon chip to be lit by an UV lamp, which completely clears the program in the memory and a new program download is enabled [6].
- **Flash memory**. This type of memory was invented in the 80s by the INTEL Company and was represented as a successor of the UV EPROM. Since the contents of this memory can be written and cleared practically for an unlimited number of time. Microcontrollers with Flash ROMs are ideal for learning, experimentation and small-scale manufacture. Because of its popularity, most microcontrollers are manufactured in flash version today [6].
- **Electrically Erasable Programmable ROM (EEPROM).** The contents of this memory may be changed during operation (similar to RAM), but remains permanently saved even upon power loss. Accordingly, EEPROM is often used to store values, created during operation, which must be permanently saved. For example, in designing an electronic

lock or an alarm, it would be great to enable the user to create and enter a password on his/her own, hence, the idea of having the user password even after power loss applies to this type of memory [6].

The PIC16F690 uses this technology, where it has 4K words of program EEPROM with an address range from 0h to 0FFFh [7].

2.1.2.4.2 Random Access Memory (RAM)

Once the power supply is off the contents of this memory are lost. In microcontrollers it is used for temporary storing data and intermediate results created and used during operation [6].

2.1.2.5 Oscillator

This is a devices that produces a periodic waveform that is used as a synchronizing signal. Ideally the waveform are depicted as a square wave but they need not to be perfectly symmetrical. The requirement of this waveforms is that they should be perfectly periodic. The basic timing interval for a digital circuit, which is equal to one full waveform period, is called the clock cycle [3].

A stable and uniform waveform reaches exactly the same voltage every time the clock is high, and by the same token, every time the clock signal goes low the voltage level must be the same. The clock signal must remain at the high and low levels for the same time and the time between each high and low cycle must be exactly the same, this is the frequency stability of the clock. In practice, the stability and uniformity of the clock signal are more important than the absolute value. Another characteristic of the clock signal is the time required for clock levels to change from high to low and vice versa. Ideally this transition could be represented by a vertical line. This would mean that the transition is instantaneous, which is not achievable in actual circuits. Practically, some time is required for the waveform to transition from low to high and vice versa.

Mid-range PICs use internal or external device to produce clock cycles which are required for their operation. The PIC executes an instruction every four clock cycles, so the oscillator speed determines the device performance. Mid-range PICs support up to eight different oscillator modes .The oscillator mode is selected at device programming time and cannot be changed at runtime. The configuration bits, which are non-volatile flags set during device programming, and this determine which oscillator mode is used by the program, among the following modes [3]:

- ❖ Low Frequency Crystal (LP)

- ❖ Crystal Resonator (XT)
- ❖ High Speed Crystal Resonator (HS)
- ❖ External Resistor/Capacitor (RC)
- ❖ External Resistor/Capacitor (EXTRC)
- ❖ External Resistor/Capacitor (EXTRC) with clock out (CLKOUT)
- ❖ Internal 4 MHz Resistor/Capacitor (INTRC)
- ❖ Internal 4 MHz Resistor/Capacitor with CLKOUT (INTRC)

2.1.2.6 Reset

The reset mechanism places the PIC in a known condition. The reset mechanism is used to gain control of a runaway or hung-up program, as a forced interrupt in program execution, or to make the device ready at program load time [3].

Practically this means that a microcontroller can behave rather inaccurately under certain undesirable conditions. In order to continue its proper functioning it has to be reset, meaning all registers would be placed in a starting position. Reset is not only used when microcontroller doesn't behave the way we want it to, but can also be used when trying out a device as an interrupt in program execution [8].

The mid-range PICs are capable of several reset actions:

- ❖ Reset during power on (POR).
- ❖ Master clear (MCLR) reset during normal operation.
- ❖ Reset during SLEEP mode.
- ❖ Watchdog timer reset (WDT).
- ❖ Brown-out reset (BOR).
- ❖ Parity error reset.

The first two reset sources in the above list are the most common. POR reset serves to bring all PIC registers to an initial state, including the program counter register. The second source of reset action takes place when the !MCLR line is intentionally brought down, usually by the action of a push-button reset switch (figure 2.2). This switch is useful during program development since it provides a way of forcefully restarting execution [3].

The PIC16F690 has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses. Voltages applied to the pin that exceed its specification can result in both MCLR Resets and excessive current beyond the device specification. For this reason, Microchip recommends that the MCLR pin should not be tied directly to VDD. The use of an RC network is suggested where the capacitor is optional [7].

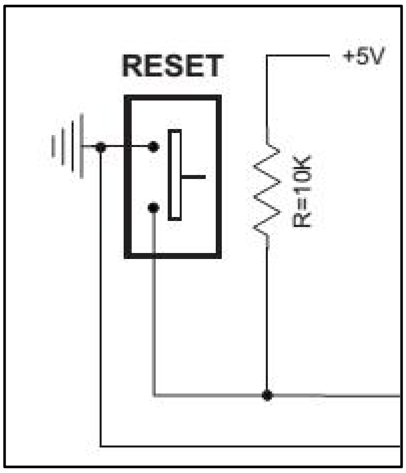


Figure 2.2: MCLR reset button [3]

During a reset, RAM memory locations are not being reset. They are unknown during a power up and are not changed at any reset. While, SFR registers are reset to a starting position initial state. One of the most important effects of a reset is setting a program counter (PC) to zero (0000h), which enables the program to start executing from the first written instruction [8].

Brown-out reset (BOR). Brown out is a potentially dangerous state which occurs at the moment the microcontroller is being turned off or in situations when power supply voltage drops to the limit due to powerful electric noises. As the microcontroller consists of several circuits which have different operating voltage levels, this state can cause the PIC out-of-control performance. In order to prevent it, the PIC microcontroller usually has built-in circuit for brown out reset. This circuit immediately resets the whole electronics when the voltage level drops below the set limit for the PIC16F690 it is 2VDC when operating on a 4 MHz clock frequency [6].

2.1.2.7 Interrupts

An interrupt is an asynchronous signal calling for processor attention, originating from hardware or software. The interrupt mechanism is a way to avoid wasting processor time, since without interrupts code has to poll hardware devices using closed loops. With interrupts, the processor can continue to do its work since the interrupt mechanism ensures that the CPU receives a signal whenever an event occurs that requires its attention. After responding to or ignoring the interrupt, the CPU resumes processing where it left off. Processors and microprocessors usually support hardware and software interrupts and mask-able and non-mask-able interrupts; interrupts originate in practically any device connected to the system [3].

The interrupt source usually originates in one of the hardware modules, although some sources generate more than one interrupt. The following are interrupt sources in the mid-range PIC16F690

- External Interrupt pin RA2/INT
- Timer0 (TMR0) Overflow Interrupt
- PORTA/PORTB Change Interrupts
- 2 Comparator Interrupts
- Analogue/digital (A/D) Interrupt
- Timer0 0verflow
- Timer1 Overflow Interrupt
- Timer2 Match Interrupt
- EEPROM Data Write Interrupt
- Fail-Safe Clock Monitor Interrupt
- Enhanced CCP Interrupt
- EUSART Receive and Transmit interrupts

Several SFRs are related to the interrupt systems. The INTCON register provides interrupt enabling and control and the PIE1, PIE2, PIR1, and PIR2 registers have specific device-related functions [7].

2.1.2.8 Timers

These are used to carry out timing operations or counting operations simultaneously with the program, to make the program faster and more efficient [9].

Timer0 uses an 8-bit register, TMR0, file register. Its output is an overflow flag, timer0 interrupt flag (T0IF), bit 2 in the Interrupt Control Register (INTCON). The timer register is incremented via a clock input which is derived either from the microcontroller oscillator (FOSC) or an external pulse train at RA4. The register counts from 0 to 255 in binary, and then rolls over to 00 again. When the register goes from 0xFF to 0x00, T0IF is set.

The timers are driven from the instruction clock, which can be monitored externally at the CLKOUT pin shown in figure 2.3, if the chip is operating with an RC clock.

Alternatively, a count of external pulses can be made, and read from the register when finished, or the read triggered by external signal. Thus, the timers can also be used as counters.

The pre-scaler is a divide by N register, where N= 2, 4, 8, 16, 32, 64,128 or 256, meaning that the output count rate is reduced by this factor. This extends the count period or total count by the same ratio, giving a greater range to the measurement [9]. Equation (2.1) can be used

$$T = \frac{C}{4PR} \text{ .... Equation (2.1)}$$

Where T is the number of clock beats per second, C is the system clock speed in Hz, P is the value stored in the prescaler, and R is the number of iterations counted in theTMR0 register [3].

Timer1 is a 16-bit counter, consisting of two 8bit registers TMR1H and TMR1L. When the low byte rolls over from FF to 00, the high byte is incremented. The maximum count is therefore 65535, which allows a higher count without sacrificing accuracy [9].

Timer2 is an 8-bit counter (TMR2) with a 4-bit pre-scaler, 4-bit post-scaler and a comparator. It can be used to generate Pulse Width Modulated (PWM) output which is useful for driving DC motors and servos. These timers can also be used in capture and compare modes, which allow external signals to be more easily measured [9].

2.1.2.9 Watchdog Timer

The watchdog timer is designed to automatically reset the microcontroller if the program malfunctions, by stopping or getting stuck in loop. This could be caused by an undetected bug in the program, an unplanned sequence of inputs or supply fault. A separate internal oscillator and counter automatically generates a reset about every 18ms, unless this is disabled in the configuration word [9].

If the watchdog timer is enabled, instructions which reset the watchdog timer are set on the appropriate program locations, besides commands which are regularly executed, then the operation of watchdog timer will not affect program execution. If for any reason (usually electrical noises in industry), the program counter "gets stuck" on some memory location from which there is no return, the watchdog will not be cleared and the register's value being constantly incremented will reach the maximum hence Reset occurs [6].

2.1.2.10 Sleep Mode

The PIC microcontroller offer a built-in sleep modes where executing this instruction puts the microcontroller into a mode where the internal oscillator is stopped and the power consumption is reduced to an extremely low level. The main reason of using the sleep mode is to conserve the battery power when the microcontroller is not doing anything useful. The microcontroller usually wakes up from the sleep mode by external reset and interrupts or by a watchdog time-out [5].

2.2 KEYPAD INPUT

This mode of input relies on switches or push buttons which give only two states that is a high or a low. The switch contacts are metal surfaces that are forced into contact by an actuator. Due to momentum and elasticity, the striking action of the contacts causes a rapidly pulsating electrical current instead of a clean transition from zero to full current. Switch bounce is taken care-off using two main methods

- timing-based schemes
- hysteresis-based schemes

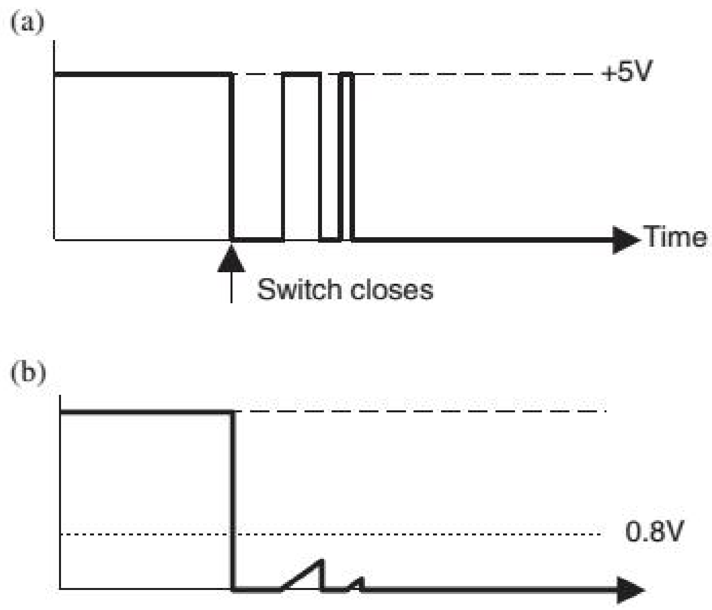
Timing-based schemes. This is where a sufficient delay is set so as to prevent the bounce from being detected. The main advantage of using timing to control bouncing is that it does not require any special switch design because it is software approach.

Hysteresis-based schemes. This technique uses hysteresis to separate the positions where the make and break actions are detected. There three types of hardware employed for de-bouncing which include: RS flip-flops, CMOS gate de-bouncers, and integrated RC circuit de-bouncers.

Switch de-bouncing can be implemented by means of a simple resistor-capacitor circuit. The circuit action is based on the rate at which the capacitor recharges once the ground connection is broken by the switch. As long as the capacitor voltage is below the threshold level of the logic zero value, the output signal continues to appear as logic zero [3].

$$t = \frac{1}{RC} \quad \text{Seconds} \qquad \text{…………..Equation 2.2}$$

With a 10KΩ pull-up resistor, a 10nF capacitor would give a time constant of 100mS, which should be more than adequate [9].



(a) No capacitor. (b) capacitor present
Figure 2.4: Ideal de-bouncing wave

2.3 LIQUID CRYSTAL DISPLAY (LCD)

This a pixilated flat output device capable of displaying ASCII characters and dot-based graphics which can be monochrome or colour by use of the light modulating properties of liquid crystals which do not emit light directly.

The LCD is very popular in the market due to its low cost and good contrast. It also has the distinct advantage over the light emitting diode (LED) in that it has a lower power consumption (typically of the order of microwatts as opposed to LEDs which is of order of mili-watts). It is due to low power consumption that there is compatibility with the metal oxide semiconductor (MOS) integrated circuit also LCDs have an incorporated controller for refreshing LCD, there by relieving the CPU of the task of refreshing the LCD and also the ease of programming for characters and graphics [6].

Nevertheless, the LCD suffers from a number of limitations including short operating life, poor visibility in low ambient lighting, requirement of an alternating current (AC) drive, limited temperature range (between 0 and 100 degree Celsius), slow speed and need of light source [3].

2.3.1 LCD Structure

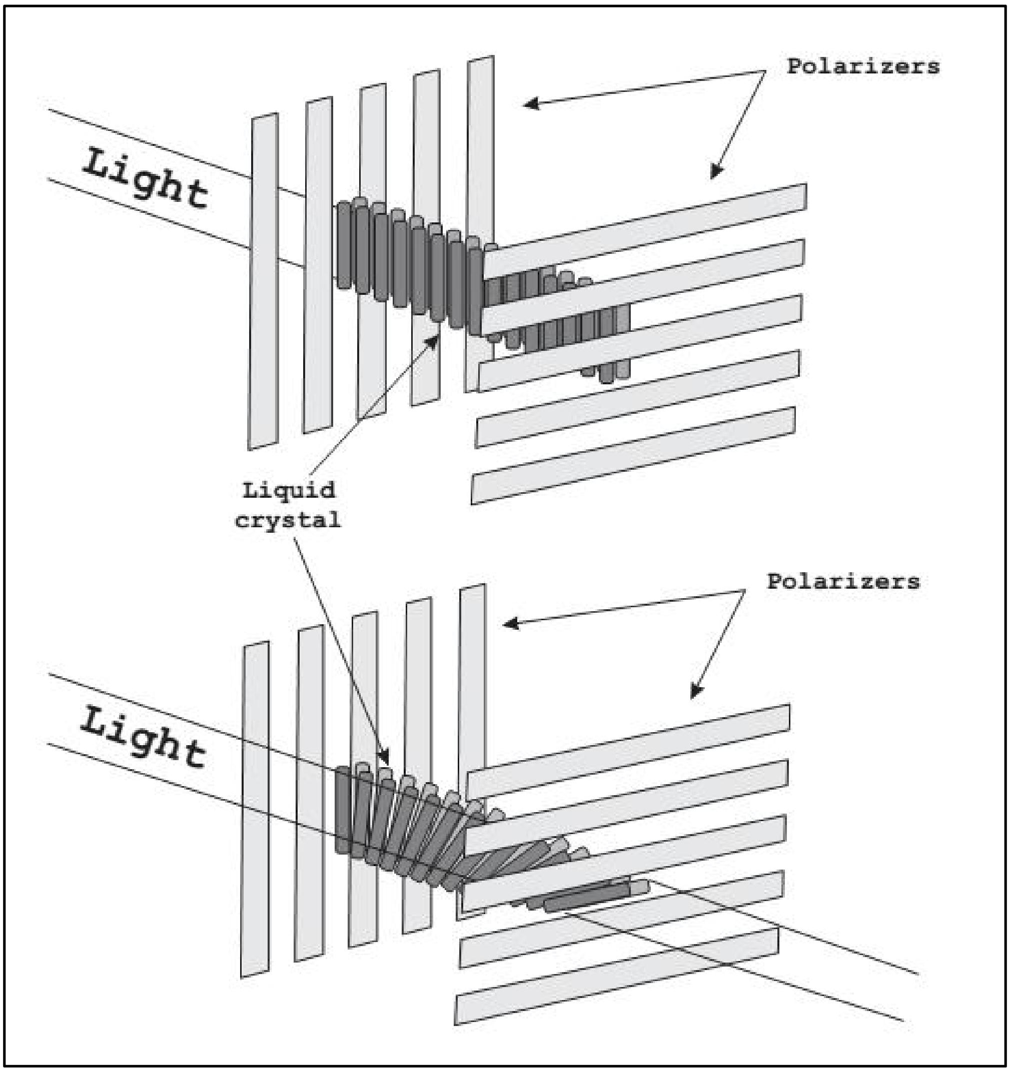


Figure 2.5: LCD operation [3]

In operation the liquid crystal display consists of two pieces of polarized glass with perpendicular axes of polarity. Sandwiched between the polarizers is a layer of Nematic crystals, as shown schematically in Figure 2.5

In the top image of Figure2.5 light cannot pass through the system since the liquid crystal layer preserves the original angle of vibration of the light which cannot pass through the polarizer. In the lower image the various molecular layers of the liquid crystal are twisted approximately 90 degrees. This twisting of the liquid crystal also changes the light's pane of vibration. So when light reaches the second polarized filter it vibrates at the same angle as the final molecule layer of the liquid crystal and can pass through the polarizer.

When the current supply is removed, the various molecular layers resume to their twisted form. By varying the amount of twist in the liquid crystals the amount of light that passes through can be controlled [3].

2.3.2 LCDs Classification

LCDs can be reflective type (when one glass sheet is transparent and the other has a reflective coating) or transmissive type (when both glass sheets are transparent). With regards to illumination, the LCD depends entirely on illumination falling on it from an external source for its visual effects. The transmissive LCD is illuminated from the back and viewed from the front. This type is common in applications that require high levels of illumination, as is the case with computer displays and television sets. Reflective LCDs, on the other hand, are illuminated by an external source. This type finds use in digital watches and calculators.

LCDs can be colour or monochrome. In colour systems each individual pixel consists of three cells, which are coloured red, green, and blue. These cells, sometimes called sub-pixels, are controlled independently to yield many possible colours for each pixel. Most LCDs used in microcontroller systems are monochrome [3].

In high-resolution and colour LCDs an active matrix display is used in the design and a grid of thin –film transistors is added to the polarizing and colour filter. Each row line and column line is addressed individually. This active matrix display are brighter and shaper and also have a quick response time hence the name thin filmed transistors (TFT) [3].

2.4 ELECTROMAGNETIC LOCK

A solenoids is wound wire magnetic coils with an open core to receive a sliding cylindrical plunger. When the coil is energized with an electrical current, a magnetic field is created in the hollow opening which pulls the plunger into it or pushes it out, based on the orientation of the solenoid and the poles of the plunger.

Most solenoid switches have only one switched pole due to the amount of current being passed through them. Some are only momentarily operated, such as is in the case of a starter solenoids on automobiles. Solenoid switches perform both electrical switching and coordinated mechanical motion. Solenoid switches are used to mechanically engaging or disengaging their shafts. This allows latching and opening mechanisms for windows, doors and hatches .Coils inside the solenoid are energized to create a magnetic field which attracts and pulls a plunger.

An electric strike can also be used for door access control. The electric strike can have to basic control configurations

- Fail-secure – this is where applying current to the strike will cause it to open. In this configuration the door would remain closed in case of power failure

- Fail-safe - this is where applying current to the strike will cause it to lock hence on power failure it does not provide security [10]

2.5 POWER SUPPLY UNIT

Standard logic circuits usually require a power source of +5 VDC. One possible source of +5 VDC is in one or more batteries. An alternative power source can be from the standard wall outlet which is 230VAC at a frequency of 50Hz.The circuitry that convert the 230VAC to the required DC voltage is known as the power supply. Components utilized for this are a step down transformer which scales the AC voltage down, bridge rectifier which converts the AC voltage to a DC voltage, capacitors for filtering the ripples from the bridge rectifier. In addition, most power supplies include a voltage regulator component that ensures that the circuit voltage is exactly the required amount [3].

2.6 PROGRAMMING

This is the art of communication with devices using an artificial language that can be used to define a sequence of instructions that can be processed and executed by such devices. There are two broad classifications of programming languages – the low-level and the high-level languages [3].

2.6.1 Low-level Language

This type of language is written specifically for a particular type of micro-controller or microprocessor. This means that it cannot be used by another microcontroller. The instructions in this language are in mnemonics. This is called Assembly language [11].

2.6.2 High Level Language

This type of language is formed from parts of natural language such as English. It is a high level of abstraction between what is asked by the computer and what the computer actually understands. It is easily understood by humans more easily than assembly languages. But, like the assembly language the computer cannot understand it. They therefore have to be translated into machine code the language the computer understands [11].

2.6.2 Machine Code

This is a sequence of carefully timed series of ON and OFF signals that can also be called high and low pulses or digital zeros and ones. The code usually represents numbers, data and instructions for manipulating those numbers and data [11].

CHAPTER 3: DESIGN AND IMPLEMENTATION

3.1 Introduction

This chapter seeks to explain in depth the process used in the implementation and operation of the digital lock. This lock design entailed the following:

- Hardware module
- Software module
- Printed Circuit Board
- Power supply unit

The flow chart shown in fig 3.2 depicts the sequence of events used in the implementation of the project.

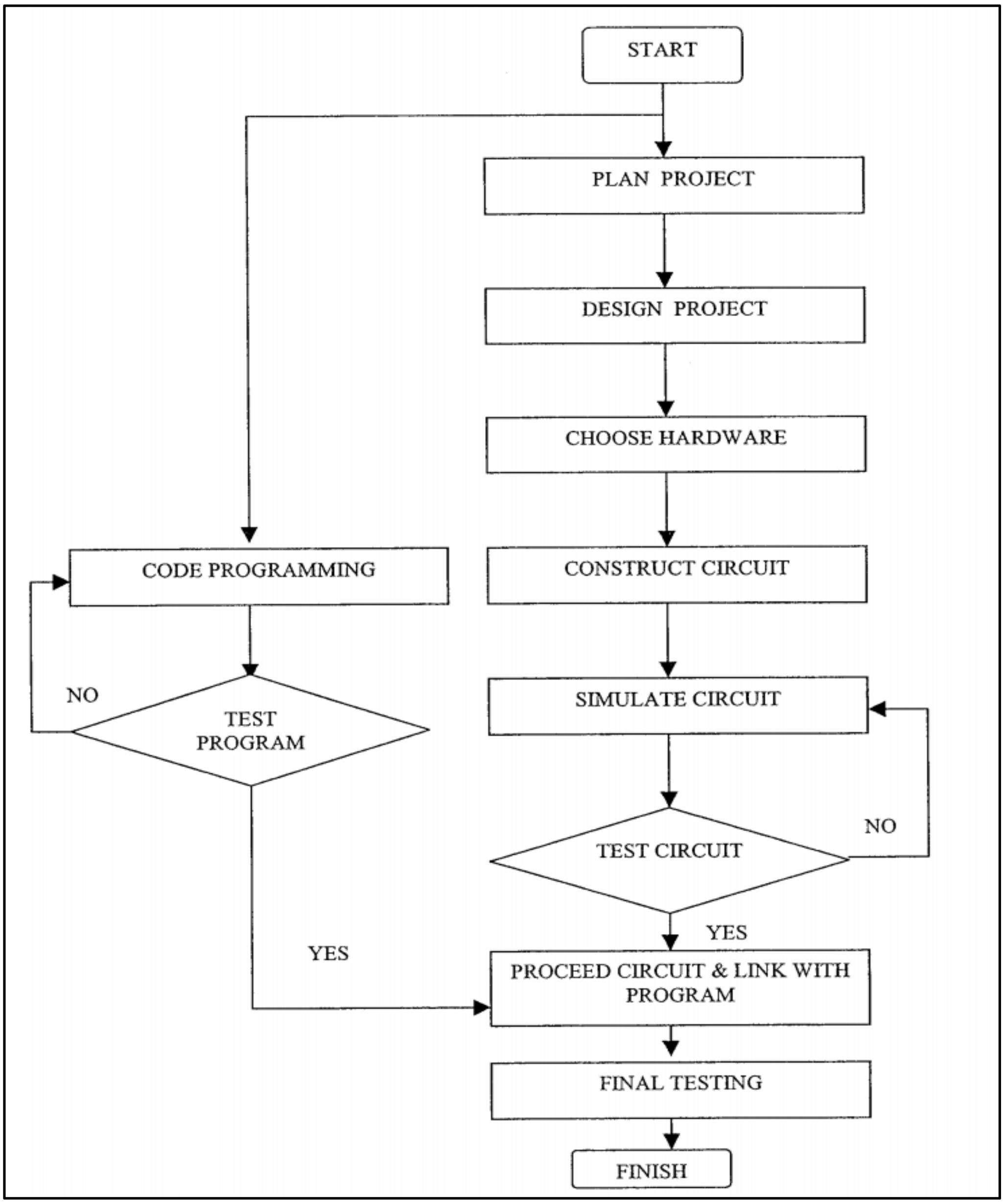


Figure 3.1: Work flow Chart

3.2 Hardware module

The project hardware can be summarized /simplified in form of a block diagram as shown in fig 3.1 below.

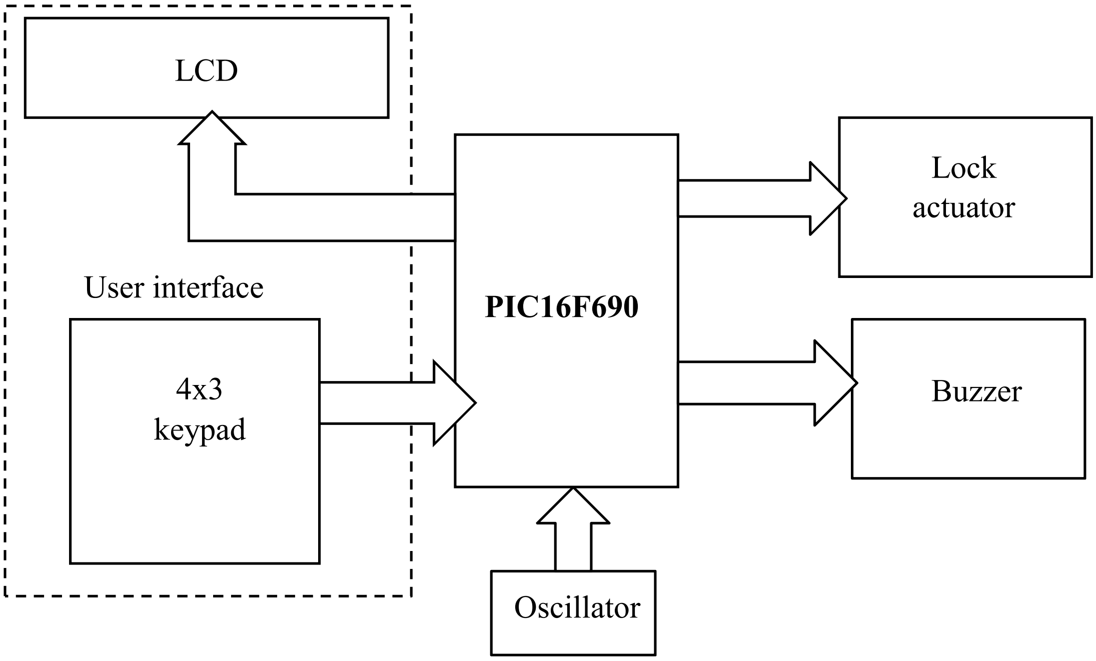


Figure 3.2: Block Diagram

3.2.1 LCD

LCDs are classified according to their interface into serial and parallel. Serial LCDs require less I/O resources but execute slower than their parallel counter parts. In addition, they are considerably more expensive [3], hence the reason for using the parallel interface LCD. The LCD displays alphabetical, numerical and symbolic characters from the standard ASCII character set, it can also display low-resolution graphics [6].

The display used is a standard LM016L which displays 2 lines of 16 characters (16x2). Each character is 5x8 pixels, making it 80x16 pixels overall. The display receives ASCII codes for each character at the data inputs (D0–D7) when operating in 8bit mode but for the project it’s operated in 4 bit mode hence the data inputs are D4 to D7. The data is presented to the display inputs by the PIC16F690, and latched in by pulsing the E (Enable) input. The RW (Read/Write) is tied low (write mode), as the LCD is receiving data only. The RS (Register Select) input allows commands to be sent to the display. RS=0 selects command mode, RS=1 data mode. The display itself contains a microcontroller; the standard chip in this type of display is the Hitachi HD44780. It must be initialised according to the data and display options required.

The LCD in the project is used in 4bit mode and hence requires only 6 port bits in the PIC16F690 instead of 10 port bits when used in 8bit mode, the LCD is used to display instructions, actions, options and data entered during operation of the digital lock. In the circuit diagram a variable resistor is present which is used to vary the brightness of the LCD. The LCD connection to the PIC16F690 is as shown in figure 3.8

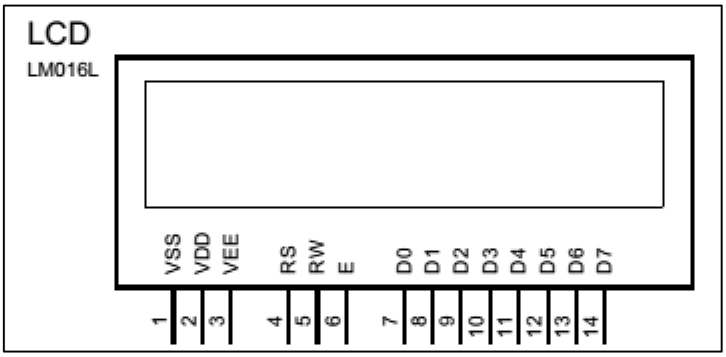


Figure 3.3: LCD module

3.2.2 KEYPAD

In this project 4x3 matrix keypad is used. A keypad is simply an array of push buttons connected in rows and columns, so that each can be tested for closure with the minimum number of connections. A total number of 12 keys which makes the probability of guessing the correct passcode very low that is

$$P = \frac{1}{12^n} \quad \text{.............. Equation 3.1}$$

Where P is the probability of a correct passcode while n is the length of the passcode.

An additional function of the keypad is the ‘H’ button which is used to wake the microcontroller from a sleep mode using an external interrupt pin RA2 (pin 17) connected to the ‘H’ button. The interrupt on pin 17 responds to any change in state of the pin once activated.

In connection 3 bits of a port are used for column indexing, which are set as input port, to check whether they have dropped to 0 voltages when they are attached with the selected row after pressing the key. On the other hand, 4 bits of a port are set to output (scan lines), and are used for row indexing. Since they are the output ports, they can be set to high (1) or low (0) .By setting each row to zero for every key scan, the system can check each column, to see if any one of them is set to 0. If so, this means that one key of that row has been pressed. This process is repeated for all rows to check all the keys [12]. The 3 pins for the keypad columns are kept high by software using the PIC16F690 internal pull ups. The keypad connection to the PIC16F690 is as shown in figure 3.8

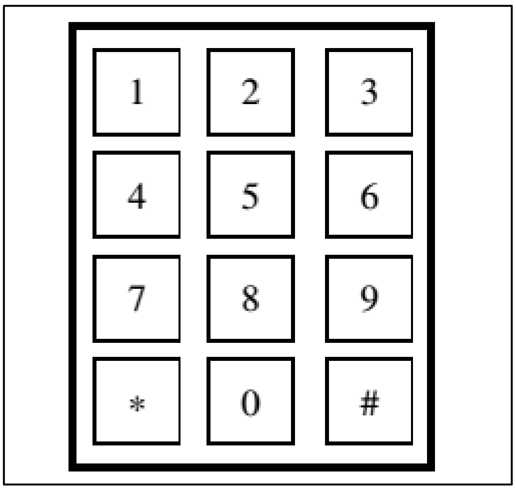


Figure 3.4: 4x3 Matrix Keypad

3.2.3 XT Oscillator mode

For accurate timing applications, the clock circuitry consists of a crystal, and two small ceramic capacitors. The commonly used crystal frequency is 4-MHz. The capacitors C3 and C4 for this crystal should be around 22pF to 33pF for stabilizing the frequency. Current consumption in this mode is medium.

3.2.4 Buzzer

A buzzer rated 3.5VDC – 24VDC which operates at a rated 20mA at 24VDC is used to produce a beep in case of a wrong code entry and a continuous loud beep for the case of an attempted entry.

3.2.5 Lock actuator

The solenoid lock actuator is designed in a normally closed mode this means when there is no supply it is mechanically closed hence a door cannot be opened when in the closed position. When a voltage is supplied the plunger is pulled inside for 6 second duration hence allowing pushing or pulling of the door to an open position. On return of the door to the closed position a sliding ramp on the plunger allows the plunger to slide back in to closed position figure 3.5 shows the positions. Operates at 9-12VDC draws a maximum of 500mA due to this the solenoid lock actuator is connected to the PIC16F690 via a relay and a transistor.

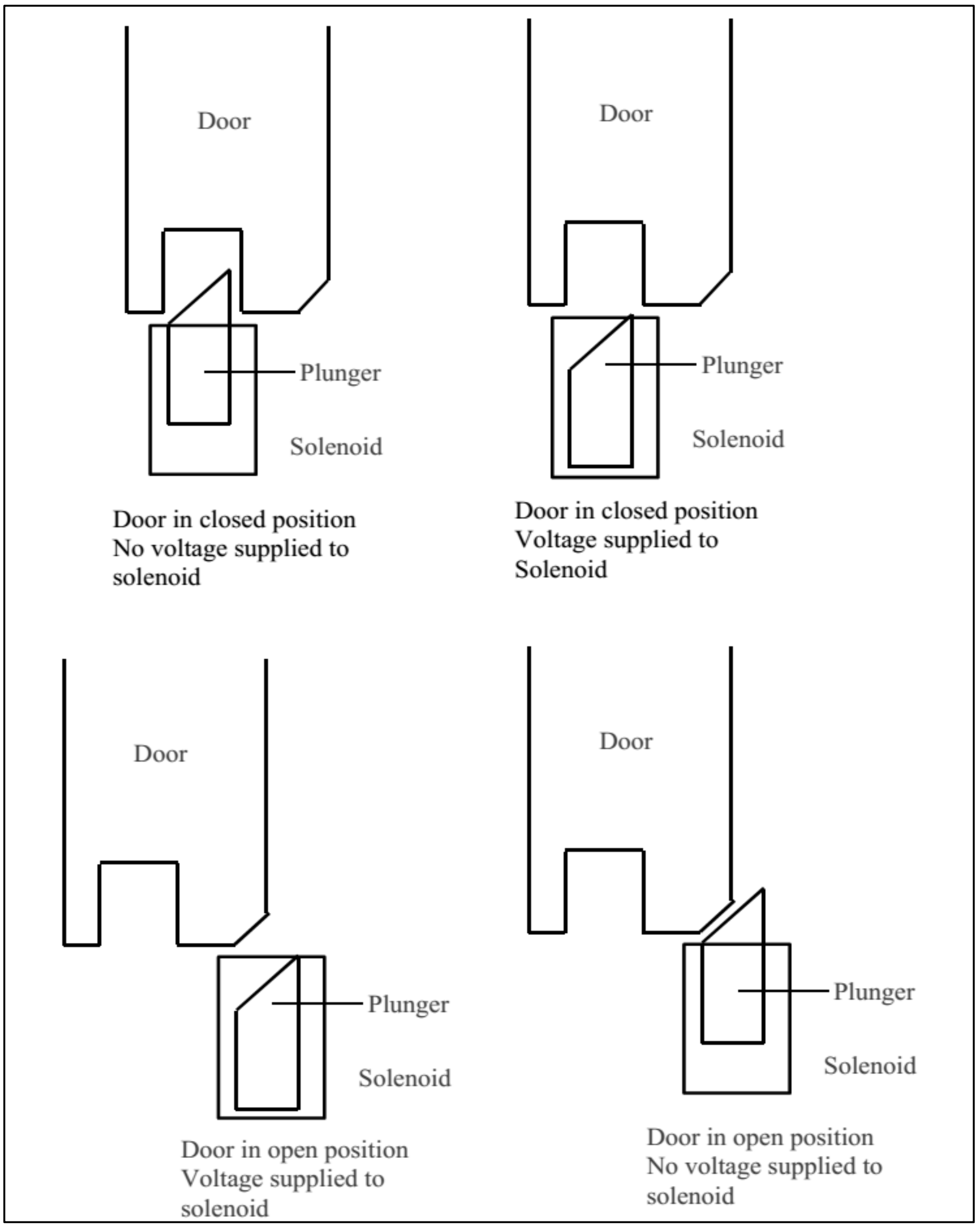


Figure 3.5: Door and Actuator positions in Plan view

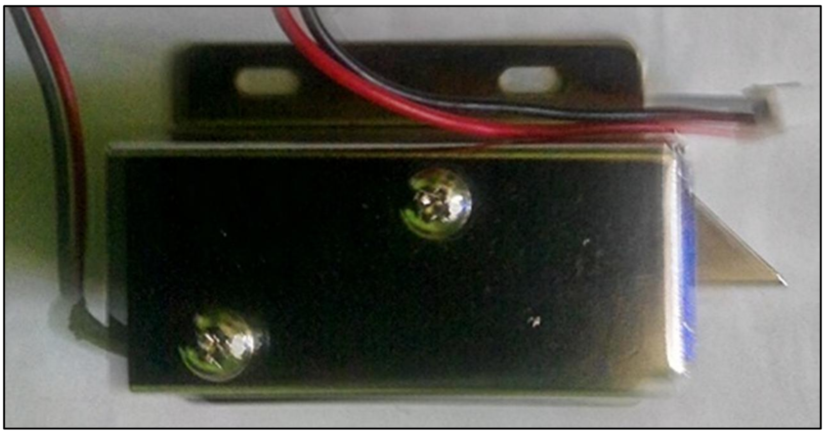


Figure 3.6: solenoid lock actuator

3.2.6 Relay

This is an electromechanical device which transforms an electrical signal into mechanical movement. It consists of a coil which when supply voltage is applied on it current flows and a magnetic field is produced that moves an armature to close the normally open contact and open the normally closed contact, when power is removed from the relay the coil produces a voltage in the opposite direction called the back EMF

The relay is used to operate the external solenoid forming part of a locking device, normally the relay remains off. As soon as the output pin of the microcontroller goes high, the transistor switches on hence the relay operates. A Free-wheeling diode 1N4001 protects the relay driver circuit from the reverse voltage developed in the relay coil by dissipating the back EMF without damaging the transistor. [5]

Relay calculations

Figure 3.7: HK4100F-DC5V-SHG

HK4100F- type, DC5V- coil type and voltage, S- Seal type H-coil power, where H is 0.2W and D is 0.15W from the relay data sheet

$$p = vi \text{...............} \quad \text{Equation 3.2}$$

Hence for the 5V coil requires a current of $= \frac{0.2w}{5v} =$ 40mA

3.2.7 Transistor drive

Since the microcontroller cannot provide sufficient supply for the relay coil where the PIC16F690 maximum sinking/sourcing current is 25mA while the coil require 40 mA. Hence the relay is connected to the Microchip PIC16F690 output pin via a driver NPN transistor (BC547) which is a medium power NPN transistor that is mostly used as a switch as shown in figure 3.8. The transistor needs to operate in saturation mode for switching operation and also to minimize the power dissipated across the collector and emitter. The purpose of the resistor at the base of the transistor is to ensure that the transistor operates at saturation and it also protects the microcontroller pin from damage

Calculating transistor drive base resistor

The NPN BC547 has DC current gain (hfe) of 150, while the relay coil requires a current of 40mA

$$\beta = \frac{I_C}{I_B} \quad \text{..........................Equation 3.3}$$

$$I_B = \frac{40mA}{150} = 0.267mA$$

Ideally 0.267mA should be the base current but practically the value is doubled hence the base current is 0.534mA

$$R_B = \frac{5V}{0.534mA} = 9.363K\Omega$$

Hence the approximate value for the base resistor should be 10KΩ

3.2.8 Safety override switch

A switch is present on the 12VDC side of the relay that connects the negative side of the relock actuator to ground when activated, this switch immediately completes the circuit of the lock actuator which in turn opens. This switch is for safety override in case of emergencies its connection is as shown on the figure 3.8

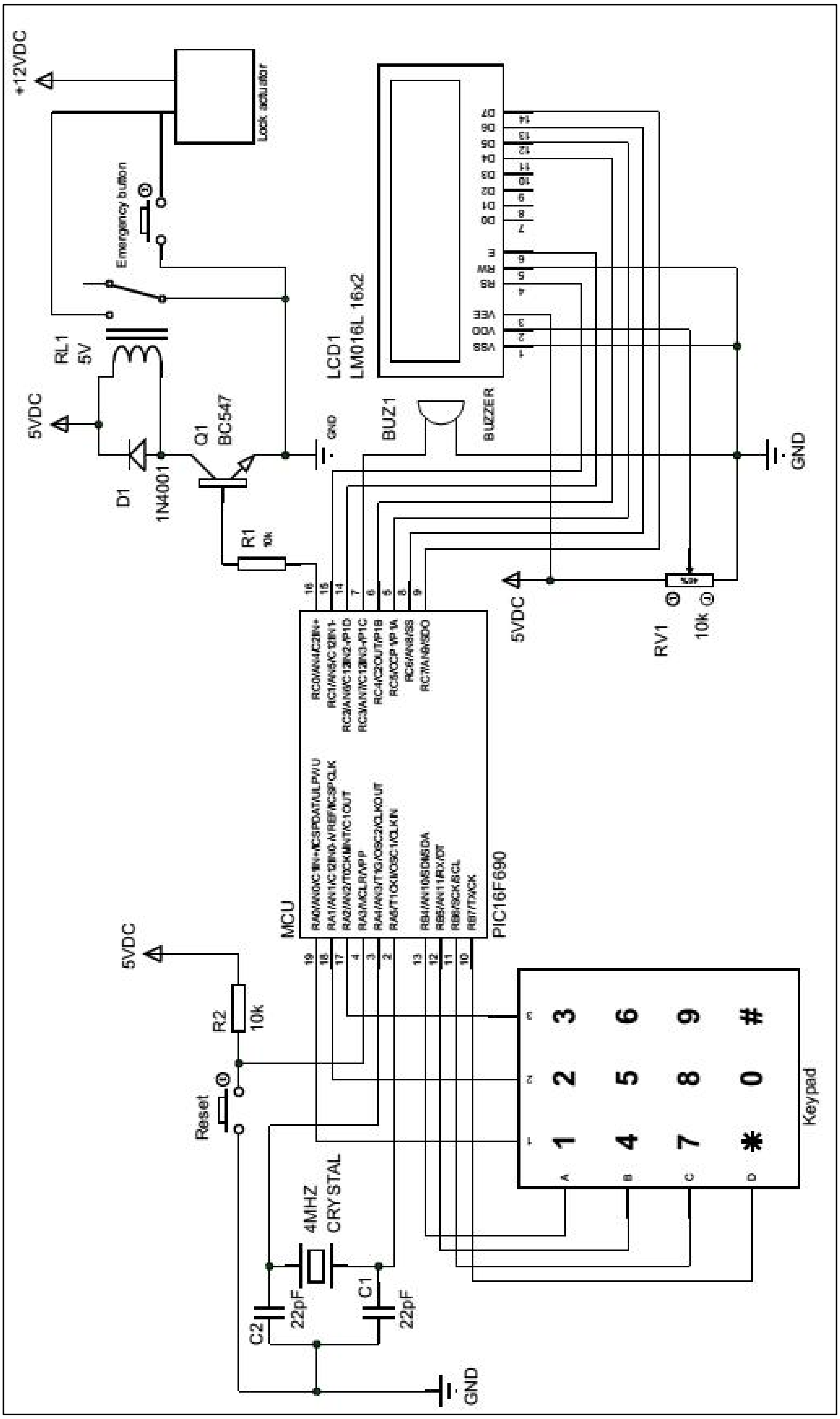


Figure 3.8: Circuit diagram

3.2 Software module

This is the programming part of the project design. The flow chart in figure 3.9 demonstrates the working of the code used in the digital lock.

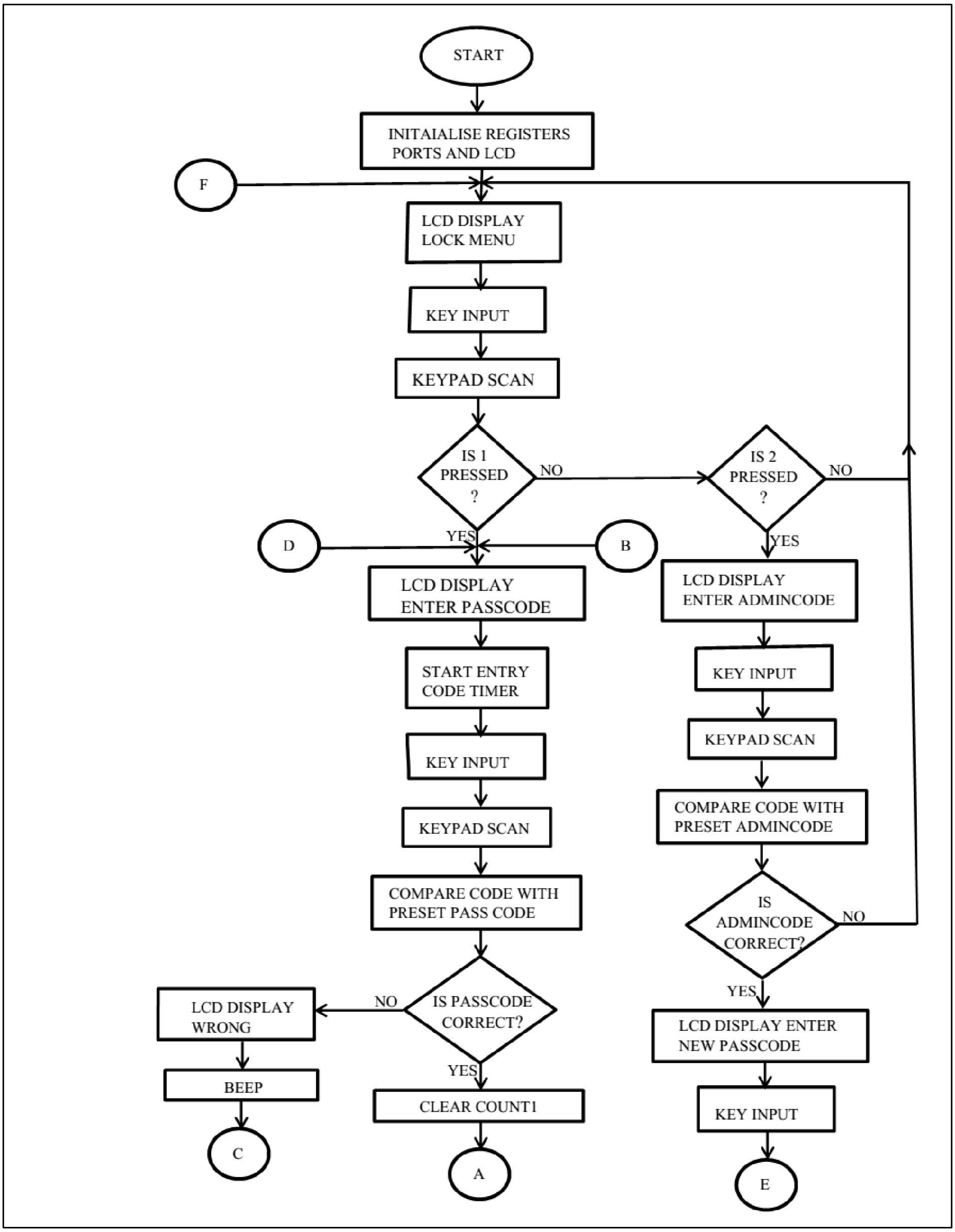
The probability of guessing a five digit passcode and getting it correct in one attempt was calculated to be

$$P = \frac{1}{12^5} = 4.019 \times 10^{-6}$$

That meant that the probability of guessing the wrong code was 0.999996 approximately one. Hence the passcode was set to be a 5 digit passcode.

45second code entry duration was chosen for code entry duration as it is a replica of the Safaricom Mpesa code entry time limit.

The number of attempts on code entry was chosen to be three as this is also a replica of mobile phone SIM cards code entry attempts


START
INITAIALISE REGISTERS PORTS AND LCD
F
LCD DISPLAY LOCK MENU
KEY INPUT
KEYPAD SCAN
IS 1 PRESSED ?
NO
IS 2 PRESSED ?
NO
YES
D
B
YES
LCD DISPLAY ENTER PASSCODE
LCD DISPLAY ENTER ADMINCODE
START ENTRY CODE TIMER
KEY INPUT
KEY INPUT
KEYPAD SCAN
KEYPAD SCAN
COMPARE CODE WITH PRESET ADMINCODE
COMPARE CODE WITH PRESET PASS CODE
IS ADMINCODE CORRECT?
NO
YES
LCD DISPLAY WRONG
NO
IS PASSCODE CORRECT?
LCD DISPLAY ENTER NEW PASSCODE
YES
BEEP
CLEAR COUNT1
KEY INPUT
C
A
E

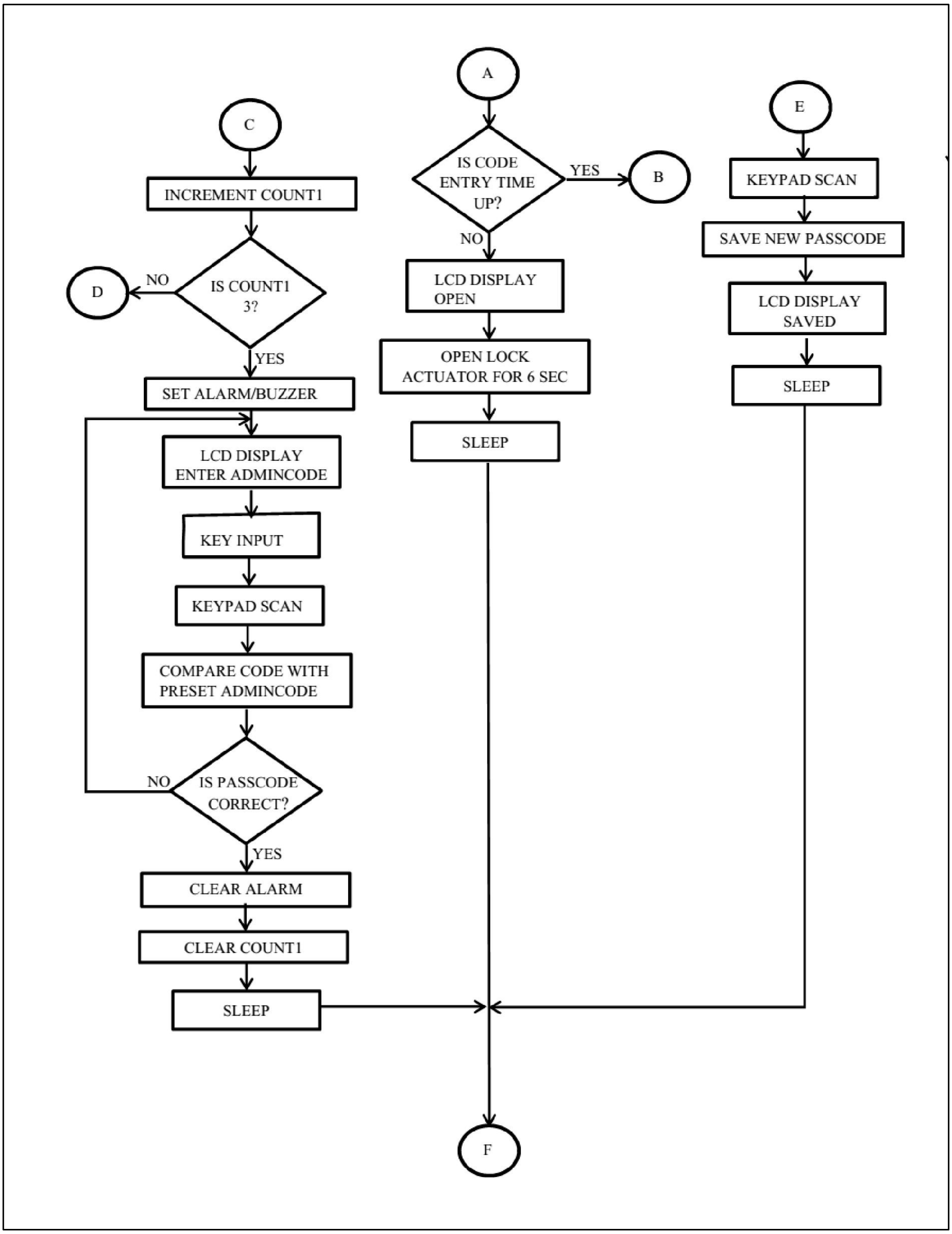


Figure 3.9: flow chart

From the flow chart an assembly language code was written and programed into the PIC16F690 using the PICKIT 2

3.3 Printed Circuit Board

This board was used to mechanically support and electrically connect electronic components using conducting tracks etched from a copper sheets laminated onto a non-conductive substrate in a process known as fabrication. Figure. 3.10 shows the PCB circuit that was used in the fabrication process.

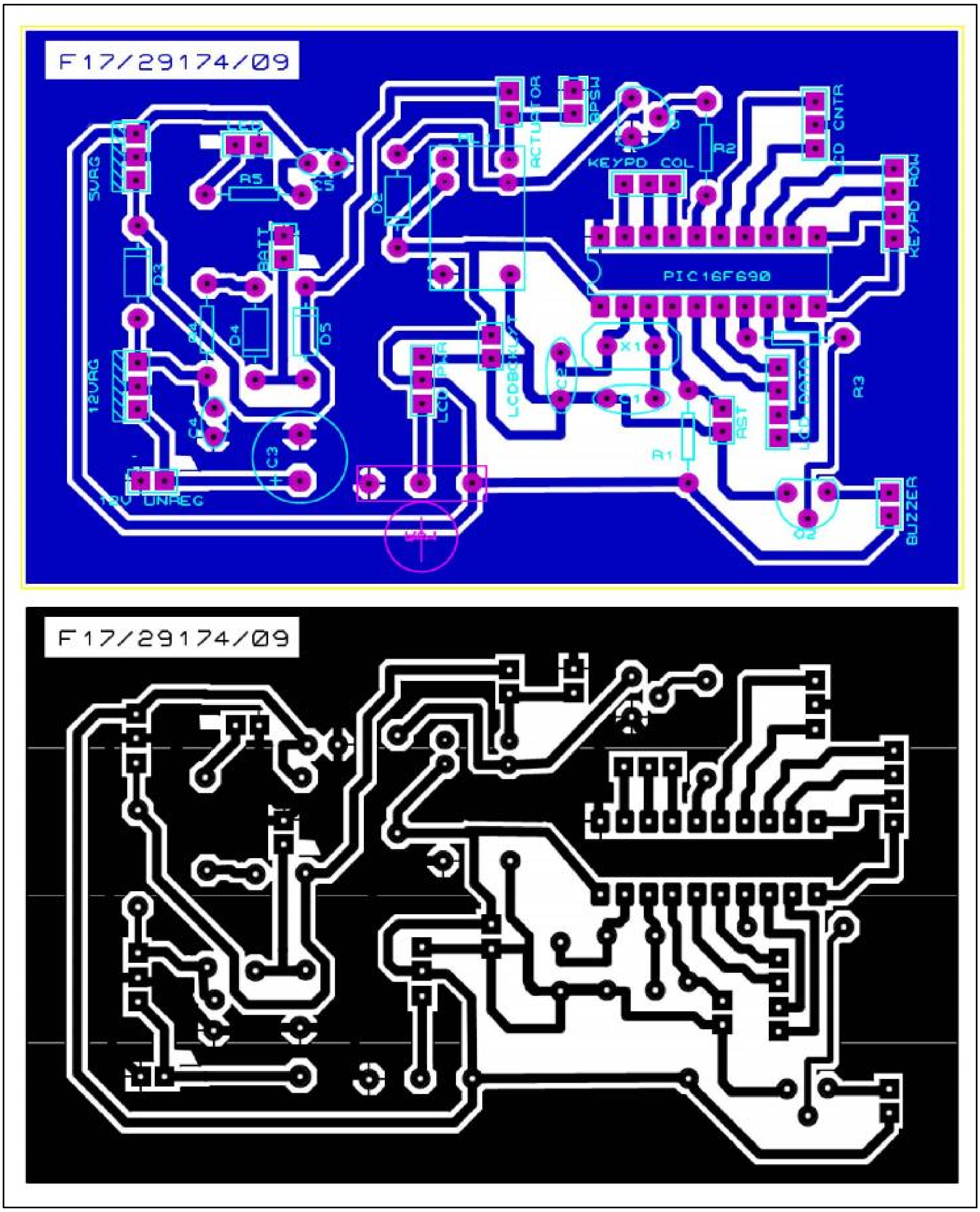


Figure 3.10: PCB image

Figure 3.11 shows the designed PCB board with components mounted on it.

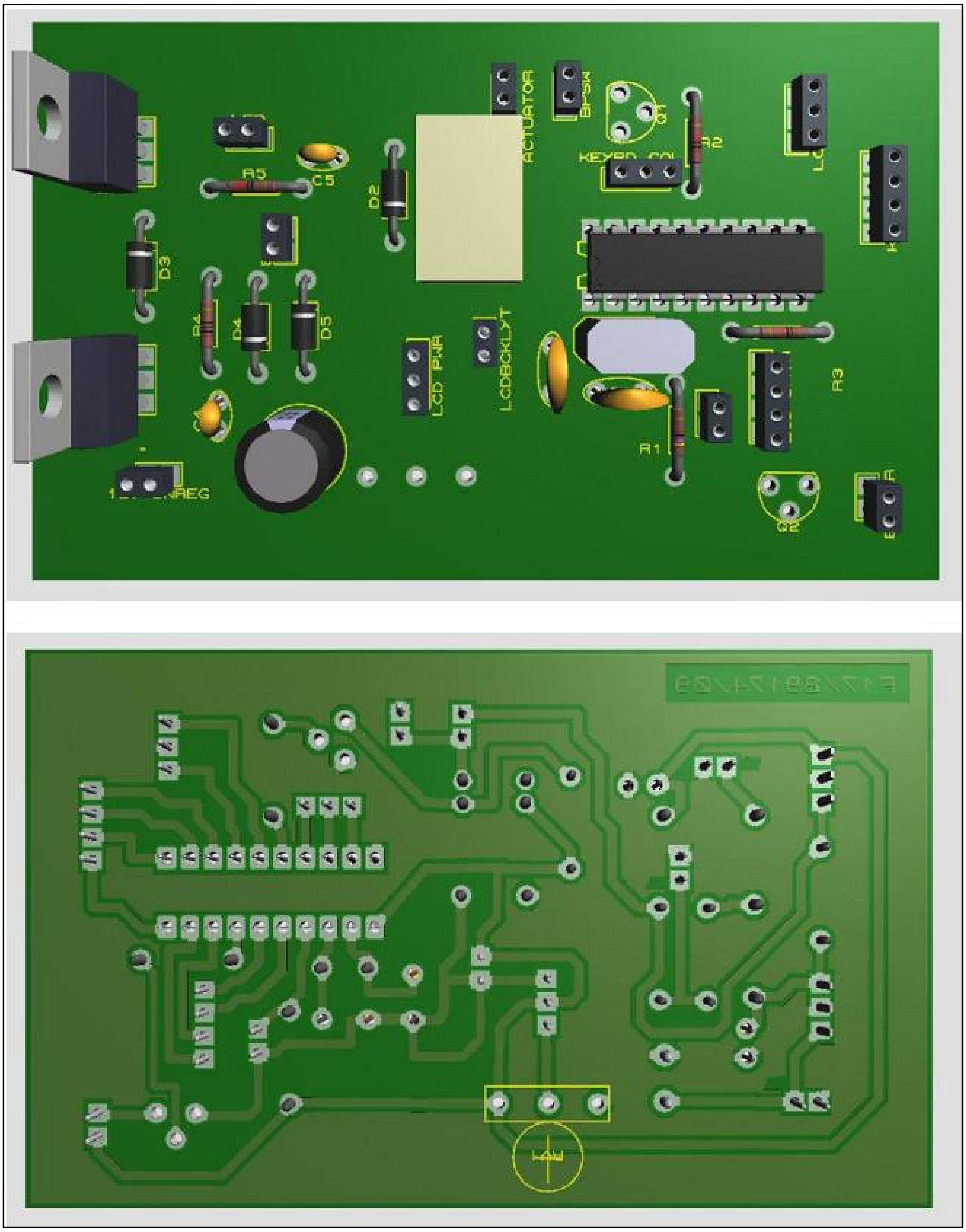


Figure 3.11: Software 3d view of the PCB board top and bottom surfaces

3.4 Power supply

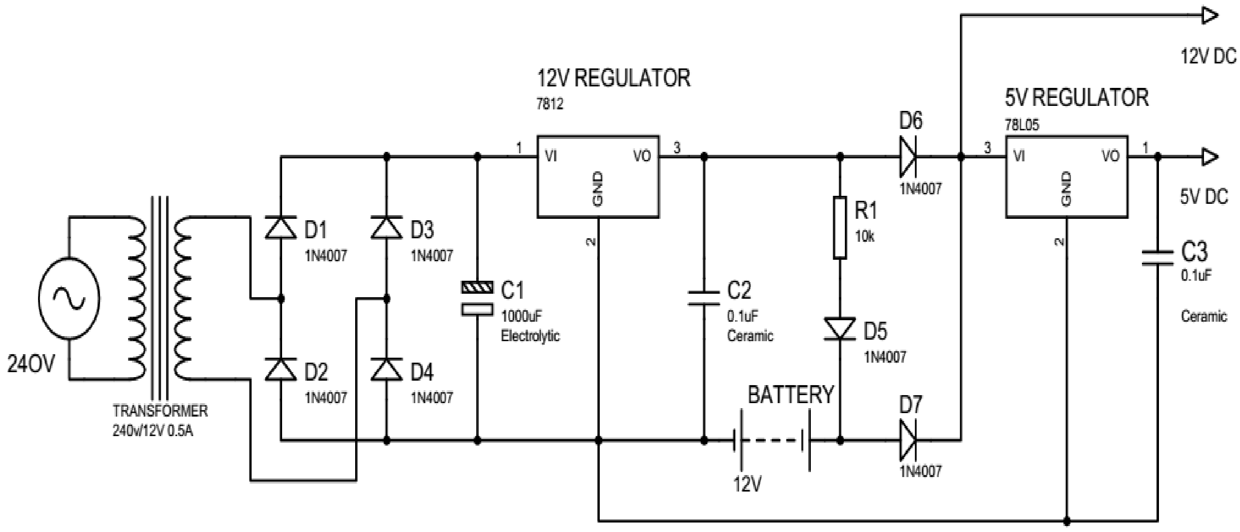


Figure 3.12: power supply circuit and backup battery circuit

The PIC16F690 operates in the range of 2.0VDC to 5.5VDC [7]. A voltage regulator circuit is usually used to obtain the required power supply voltage when the device is to be operated from a mains adaptor or batteries [5].

A Power supply circuit is required to convert the socket 240VAC to a stable 5VDC. The 240VAC mains supply is stepped down to 12VAC using a step-down transformer. The output from the secondary of the transformer is rectified by a bridge rectifier comprising of four diodes (1N4007) which have a root mean square (RMS) Reverse Voltage of 700V and an average rectified output Current 1 Amp and filtered by an electrolytic capacitor 1000uF which has an across voltage of 35V which smooth's out the dc from the rectifier hence producing a large dc with little ripples. The filtered output is regulated by 7812 IC to obtain the required 12VDC through a 0.1uF ceramic capacitor which filters the output of the 7812 IC [1]. The 12VDC obtained from the regulator is used to trickle charge a backup lead acid battery as shown in the circuit [13]. Lead acid batteries are inexpensive and simple to manufacture in terms of cost per watt hours, reliable, have Low self-discharge and also Low maintenance requirements. However they have their limitations which are not of great a magnitude to this application like they Cannot be stored in a discharged condition, poor weight-to-energy density limits use to stationary and wheeled applications and environmentally unfriendly.

D6 is forward biased when using the ac power and reverse biased when using the battery. R1 limits the charging current to the batteries where the current for charging is specified on the batteries data sheet. When ac power goes of the battery power kicks in where it supplies the circuit [13].

12VDC is used for the door actuator (linear solenoid actuator) while 5VDC obtained from a 7805 regulator is used to power the PIC16F690. The 7805 IC can receive an input of 8VDC to 35VDC to output a constant 5VDC.Often, the last two digits of the device number are the output voltage i.e. a 78L05 is a +5 VDC regulator [3].

CHAPTER 4: RESULTS AND ANALYSIS

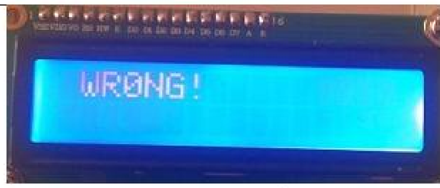
This chapter seeks to show the results obtained and the analysis done.

There were two types of results which included the following:

- Simulation results
- Practical results

Table 4.1: Results

EXPECTED RESULT	SIMULATION	PRACTICAL
Switching on the lock Result '1DOOR PASSCODE' '2CHNG PASSCODE'		
Entering the door passcode '12345' Result 'OPEN'	 	 
Entering the door passcode '54321' Result		

‘WRONG!’ Beep sound	 Beep sound	 Beep sound
Entering the wrong code two times Result ‘LAST ATTEMPT!’ Beep sound	 Beep sound	 Beep sound
Entering the wrong code three times Result ‘ADMIN CODE!’ Alarm sounds	 Alarm sounds	 Alarm sounds
Entering the correct code after 45 seconds Result ‘TIME OUT!’		
Saving new passcode ‘11111’ Result ‘ADMIN CODE!’ ‘99999’		
‘NEW PASSCODE’ ‘11111’ ‘SAVED’	 	 

From table 4.1.1 the practical and simulation results were as expected

Figure 4.1: PCB fabrication

Figure 4.2: project pictures

CHAPTER 5: CONCLUSSIONS AND RECOMMENDATIONS

5.1 Discussion and Conclusion

Although the main challenge was time which was brought about by unavailability of components, the project objectives were accomplished.

A system that could be used to realise a digital locking mechanism was designed and constructed using the PIC16F690 microcontroller. Where code input was via a keypad within 45second, And lastly only 3 attempts for code entry were allowed and the last one would trigger an alert for an attempted entry, a master unlock code was then provided by the system administrator to restore the control mechanism.

In addition to the project objectives an option of changing the user passcode was also implemented where the administrator had the rights to change the existing code by entering the correct admin code the system would then prompt him to enter the new passcode which would be then saved in the PIC16f690 EEPROM.

In conclusion the use of the PIC16F690 microcontroller reduced the size and cost of implementing the circuit, making the digital lock system a low cost control system which could be used in homes, offices and industrial premises.

5.3 RECOMMENDATION

The project at hand envisaged a user with the ability to press keys and also to see by use of the keypad input and LCD output. Other modes of identification and communication which incorporate the disabled could also be implemented where modification on the mode of input and output can be implemented as an enhancement i.e. combining the keypad with biometric identification or combining the LCD with an audio output.

Modification can also be made to the whole system where information showing time, date and the user who opened the lock could be sent and stored into a database. This would be used for monitoring personnel access to rooms within a building, where each personnel would have to input his/her user ID and a specific passcode for the ID. The system would then send the ID access information to a database system which would then store the information for analysis by management. Hence security and monitoring achieved

For the case of an attempted entry more information about the user attempting to enter could be corrected at the instance of the attempt. Modification to achieve this would involve taking of a pictures and time recording. Then information would then be sent to the administrator mobile phone via the PIC16F690 in build communication modules for faster security response.

APENDIX A: Assembly language code

APENDIX B Cost analysis

Components	Cost (Ksh)
PIC16F690	450
4x3 Keypad	650
16x2 LCD	800
relay	350
5- Capacitors	50
5- Resistors	25
Variable resistor	40
4-Diode	40
LED	10
Solenoid actuator	1000
2-Transistor	50
Crystal oscillator	50
Buzzer	250
2-Voltage regulator	100
PCB	1200
Heat sink	50

Bibliography

[1] A. BezBorAh, “efymag,” 28th June 2010. [Online]. Available: www.efymag.com. [Accessed 17th January 2014].

[2] N. Hoffman, “Patent Anti-fraud biometric scanner that accurately detects blood flow”. Patent us5737439, 7 april 1998.

[3] J. S. a. M. p. Canton, Microcontroller Programming the Microchip PIC, Taylor & Francis Group , 2007.

[4] A. K. Maini, Digital Electronics Principles, Devices and Applications, JohnWiley&Sons, Ltd, 2007.

[5] D. Ibrahim, PIC BASIC Projects 30 Projects Using PIC BASIC and PIC BASIC PRO, Great Britain: MPG Books Ltd, 2006.

[6] M. Verle, PIC Microcontrollers, Mihrolektronika 1st edition, 2010.

[7] *PIC16F631/677/685/687/689/690 Data Sheet 20-Pin Flash-Based, 8-Bit CMOS Microcontrollers with Nano Watt Technology.* [Performance]. Microchip Technology Inc, 2006.

[8] jakab, *pic 16f84 Tutorial,* california: LLH Technology , 2001.

[9] M. Bates, Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation, Martin Bates: Elsevier, 2006.

[10] “White papers: Magnetic locks,” 20th March 2012. [Online]. Available: http://www.sdcsecurity.com/docs/eblasts/whitepapers_emlocks.pdf. [Accessed 17th January 2014].

[11] T. D. Green, Embedded Systems Programming with the PIC16F877, 2008.

[12] K. S. Horng, “ Microchip PIC Controlled HD44780 LCD and 3x4 Matrix Keypad,” p. 12, 22 December 2008.

[13] T. Bert, “KB8MF project page,” 2 April 2001. [Online]. Available: htttp:// www.qsl.net/kb8mfv/projects.html. [Accessed 7 march 2014].