بسم الله الرحمن الرحيم

SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

College of graduates study

School of Electronics Engineering

Wireless Patient Digital Clock Time Monitoring and Alarm System by Using Microcontroller

المراقبة والإنذار اللاسلكي لزمن مراقبة المريض باستخدام المتحكم الدقيق

The submitted in partial fulfillment of the requirement of M.Sc Design in Computer Engineering

SUBMITTED BY:

Rania Galal Hassan El Nooor

SUPERVISOR BY :

Dr.*Abdelrasoul Jabar Alzubaidi*

November 2015

الآيــــة

قال تعالى:

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿ 1 ﴾ خَلَقَ الْإِنسَانَ مِنْ عَلَقٍ ﴿ 2 ﴾ اقْرَأْ وَرَبُّكَ الْأَكْرَمُ ﴿ 3 ﴾ الَّذِي عَلَّمَ بِالْقَلَمِ ﴿ 4 ﴾ عَلَّمَ الْإِنسَانَ مَا لَمْ يَعْلَمْ ﴿ 5 ﴾

سورة العلق الآيات (1 – 5)

الإهـــداء

كنتم ولا زلتم كالنخلة الشامخة تعطي بلا حدود

(أبي ..... أمي)

إلى من هم أقرب إلي من روحي.....

أسرتي الصغيرة (زوجي الفاضل .....أبنائي)

إلي من ظلوا يواصلون معي رحلة العطاء وبهم استمد عزتي وإصراري

(أخواني ..... أخواتي)

إلى من ساروا بي في بحر المعرفة إلى مرافئ الأمان

(أساتذتي الأجلاء)

إلي من قضيت معهم سنوات الدراسة والعمل

(زملائي .....زميلاتي)

إلي روح خالتي الغالية وأمي الثانية سعاد (رحمها الله)

إلى كل هؤلاء أهدي هذا الجهد المتواضع

مع المحبة والتقدير....

Acknowledgment

All favor due to Allah first and foremost, all our thanks to our lord. Thanks to the supervisor **D. Abd alrasoul ALzubaidi** for any things he gave me his experience in the knowledge and work, helped and motivated me, to reach this form. To my family and friends, those who pushed my goals till clarity, supported to the limit for to be, sometimes over pushed it just so we succeed.

ABSTRACT

This project for wireless patient digital clock time monitoring and alarm system using microcontroller , Health monitoring is repeatedly mentioned as one of the main application areas for Pervasive computing. patient monitor of time is important factor in the recovery and healing where it is difficult to follow the patient for medication at all times, especially in places such as hospitals where the doctor monitors large number of patients and often taking the medicine in the out of indefinite time.

So here is design a patients reminder system where he notifies a time of medication in his time schedule. The system supports edit and adjust the time and also the introduction of the medication time and all the medicine is placed in a specific box inside it led when match medication appointed time with real-time illuminated that the LED in the box in the turn on and alert buzzer.

The system use microcontroller to control the alarm and keys to adjust the time and the time is display on LCD screen for viewing and assistance. There is two side one of patient and other for doctor and is connect together by wireless.

Expected results of the system works efficiently is excellent as it is required, where the patient will be informed by a time of medicine at the specified time and in all sides the doctor and patient.

المستخلص

هذا المشروع بعنوان مراقبة المريض لاسلكيا باستخدام المتحكم الدقيق والمراقبة الصحية تعتبر واحده من المجالات الصحية الرئيسية والمهمة. حيث تلعب مراقبه المريض للزمن العامل المهم في التعافي والشفاء حيث انه من الصعب متابعه المريض للدواء في كل الأوقات وخصوصا في الأماكن مثل المستشفيات حيث أن الطبيب يقوم بمراقبه عدد كبير من المرضي وكثيرا ما يتم تناول الدواء في غير زمنه المحدد .

من هنا تم تصميم جهاز منبه للمرضي حيث انه يقوم بإعلامك بزمن الدواء في زمنه المحدد، النظام يدعم تعديل وضبط الزمن وأيضا إدخال زمن الدواء و كل دواء يوضع في صندوق معين بداخله أضاءه عند تطابق زمن الدواء المعين مع الزمن الحقيقي يضاء هذا الضوء الموجود في الصندوق وينطلق جهاز إنذار.

تم استخدام المتحكم الدقيق للتحكم في الإنذار والمراقبة ومفاتيح لضبط الزمن ووحده زمن وشاشه عرض للعرض والمساعدة وايضا يدعم الجهاز ان يكون جزءين واحد لدي الطبيب والثاني عند المريض والربط بينهم لاسكيا.
النتائج المتوقعه للنظام يعمل بكفاءه ممتازة كما هو مطلوب حيث يتم اعلام المريض بزمن الدواء في الزمن المحدد وفي كل الجانبين الطبيب والمريض.

Table of Contents

List of Tables

LIST OF FIGUERS

List of abbreviations

ATmega16	:	atmel 16 Mega
AVR	:	advanced virtual RISC.
EEPROM	:	electric erasable programmable read only memory.
EMG	:	electromyography sensor.
RISC	:	reduce instruction set computer.
RAM	:	Random access memory.
RF	:	radio frequency.
SPI	:	serial peripheral interface.
LCD	:	Liquid Crystal Display.
MISC	:	Minimum Instruction Set Computer.

CHAPTER

ONE

INTRODUCTION

1. Introduction

1.1 Overview

In general, most of the patients forget to take the appropriate prescribed medication at the required time. There are occasions when patients remember to take medicines at the stipulated time but forget which pill has to be taken at that particular time. This poses a big problem as it affects the dosage quantum required for the patient that results in not yielding the right recovery result. It is difficult for doctors/paramedics/attenders to monitor patients round the clock. In order to avoid these problems, we have implemented this patient medicine reminder system.

The aim of this project is to inform the patient about the time for his Medicine through Digital clock. For the medical professionals it becomes important to continuously monitor the patient of his Medicine. And In a large setup like a hospital or clinical center where a single doctor attends many patients, it becomes difficult to keep informed about the critical conditions developed in each of the patients and his Medicine. This project provides a device which will continuously monitor the time to be monitored for a patient and Alarm when time of patient Medicine.

When it's medicine time kicks off alarm to tell the patient or the doctor that the medicine time has come, and so we have organized a patient's medicine time. The device consists of three boxes three medications placed inside and each Medicine in the box is set time Alarm separately, and also device supports modification time and adjust.

1.2 Problem Statement

Patient in critical medical situation needs to be monitored the time for his medication and for all the time and forget to take the appropriate medication at the required time. This poses a big problem as it affects the dosage quantum required for the patient that results in not yielding the right recovery result. Doctors need to be around the patient for a long time to give the medication for the patient which considered time consuming and cost.

1.3 Proposed Solution

The proposed solution is to design of wireless Patient time Monitoring and alarm System using microcontroller. The system should be able to monitor the medication time for patient.

1.4 Objectives

- Design of wireless Patient Digital clock time Monitoring and alarm System using microcontroller .
- Simulation of the proposed circuit will be run using Proteus Professional.
- Performance evaluation of the design will be highlighted.

1.5 Methodology

This project aims to Design a wireless Patient Digital clock time Monitoring and alarm System using microcontroller in efficiency way and contain two different parts software and hardware.

The program is written using proteus isis as simulation part and bascom compiler as programming language. Digital clock IC are used to save and give time to be monitored by microcontroller

The system allows the user to enter the prescribed timings, at which the patient has to take the medication. This is done using a few switches . This data will be stored in the microcontroller. The microcontroller continuously reads the time from the RTC .When the timings read from the RTC equals the timings stored in the Microcontroller, the system alerts the buzzer and the box of alarm is turn and send that by wireless communication. Thus, the patient can listen to the audio indication and see on box green LED of the medicines and take them on time. LCD also used as monitoring the time and modify the time and alarm medicines time.

1.6 chapter layout

This project will contain five chapters as follows:

Chapter one Introduction which explain the problem statement along with the proposed solution and the objectives of the research

Chapter two Literature review highlight the main parts of the system and shows the previous systems

Chapter three System design explain the working principle of the system.

Chapter four Software & Program discuss in details the program and flow chart of the software program

Chapter five Simulation results and analysis highlight several simulation result and discuss the obtained result

Chapter six Conclusion, recommendation summarized the work done in the thesis and suggest several recommendation for future work.

CHAPTER

TWO

LITERATURE

REVIEW

2. Chapter two

2.1 overview

This chapter explain the main parts used in the system and also will highlight the previous work.

2.2 previous work

In [1], In general, most of the patients forget to take the appropriate prescribed medication at the required time. There are occasions when patients remember to take medicines at the stipulated time but forget which pill has to be taken at that particular time. This poses a big problem as it affects the dosage quantum required for the patient that results in not yielding the right recovery result. It is difficult for doctors/paramedics/attenders to monitor patients round the clock. In order to avoid these problems, we have implemented this patient medicine reminder system.

The system allows the user to enter the prescribed timings, at which the patient has to take the medication. This is done using a few switches and the list of medicines to be administered is entered through the PC. This data will be stored in the EEPROM by the microcontroller. The microcontroller continuously reads the time from the RTC .When the timings read from the RTC equals the timings stored in the EEPROM, the system alerts the buzzer and displays the list of medicines to be taken at that particular prescribed time on the LCD. Thus, the patient can listen to the audio indication and see the name of the medicines on the LCD and take them on time.

Patient monitoring and management in critical care environments such as the ICU's , SICU's and ANCU's involve estimating the status of the patient and reacting to events that may be life threatening. It is impossible to keep a tab on every patient throughout the day. New

solutions are needed in this field to help the doctors and the nursing staff to monitor the patients.

A critical element of this is the medicine administration and monitoring. This has been achieved by the patient medicine reminder system. This system consists of an 8-bit microcontroller with an in-built EEPROM and a real time circuit. This system is driven by an embedded program that inputs predefined parameters which is processed based on the input variables entered via a user interface device such as the PC. All the entries made by on the PC is concurrently and simultaneously displayed on the LCD panel of the device. The logic for the processing is built into the embedded program to initiate the alert through an audio alarm. Not only does it have an alarm system, but also an LCD display which displays which medicine is to be taken at the reminder time.

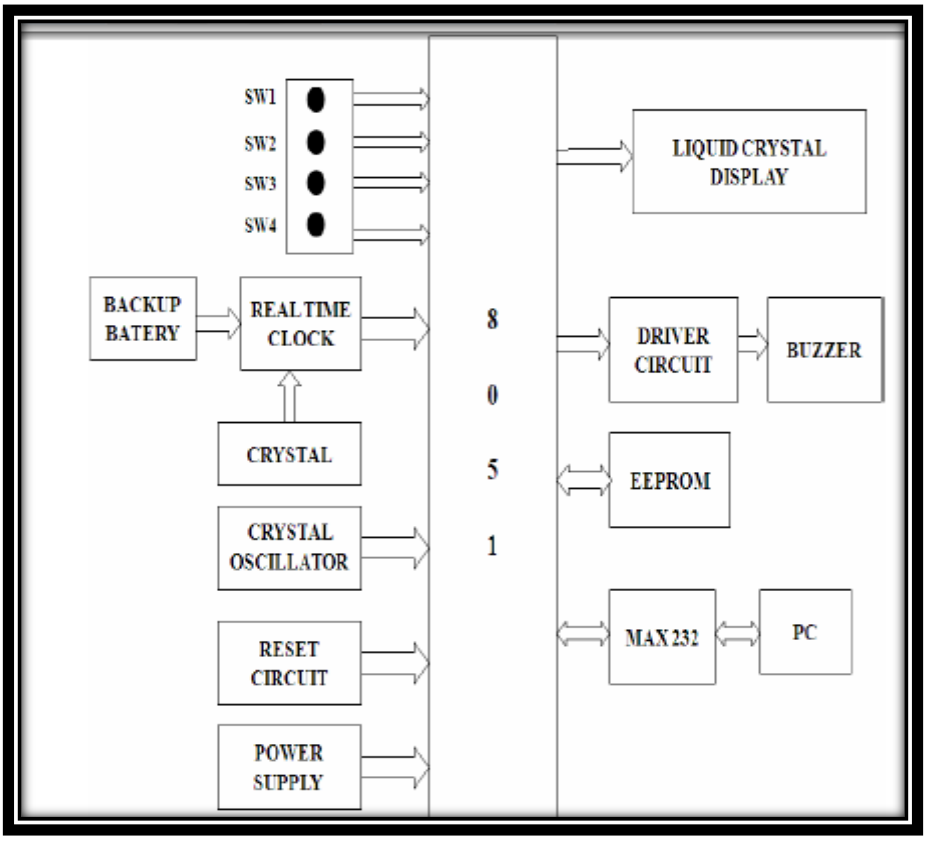

Figure 2.1 Block diagram of the Microcontroller based patient Medicine reminder system.

In [2], The main objective of this project is to design a programmable medication reminder system for patients to get the reminder of the medicine with a buzzing sound as well as to display the name of the medicine to be taken at that time.

In general, most of the patients forget to take the appropriate prescribed medication at the specified time. This creates a big problem as it affects the timely dosage required for the patient, which eventually prolongs the illness and the recovery process gets delayed. It is difficult for the doctors/paramedics/attenders to monitor patients round the clock. In order to avoid these problems, a programmable medication remainder system can be developed for reminding patients who forget to take their medicine on time.

The proposed system uses a microcontroller of the 8051 family and a battery for power supply. Here the keypad interfaced to the microcontroller for storing information. The real-time clock (RTC) is used for maintaining accurate time, and a buzzer for indication, which is connected to the microcontroller. The 16*2 LCD display is interfaced to the microcontroller for displaying the information.

The system allows a user to enter the prescribed timings (the timings at which the patient has to take the medicines) through the keypad. This data will be stored in the microcontroller. The microcontroller continuously reads the time from the RTC. Whenever the time matches with the stored time, the system gives an alert through a buzzing sound and also displays the medicine name.

This project in future can be enhanced by integrating it with GSM technology with which a patient receives a reminder about the medicine he has to take via SMS on his/her cell phone, the next figure3.2 show the block diagram.

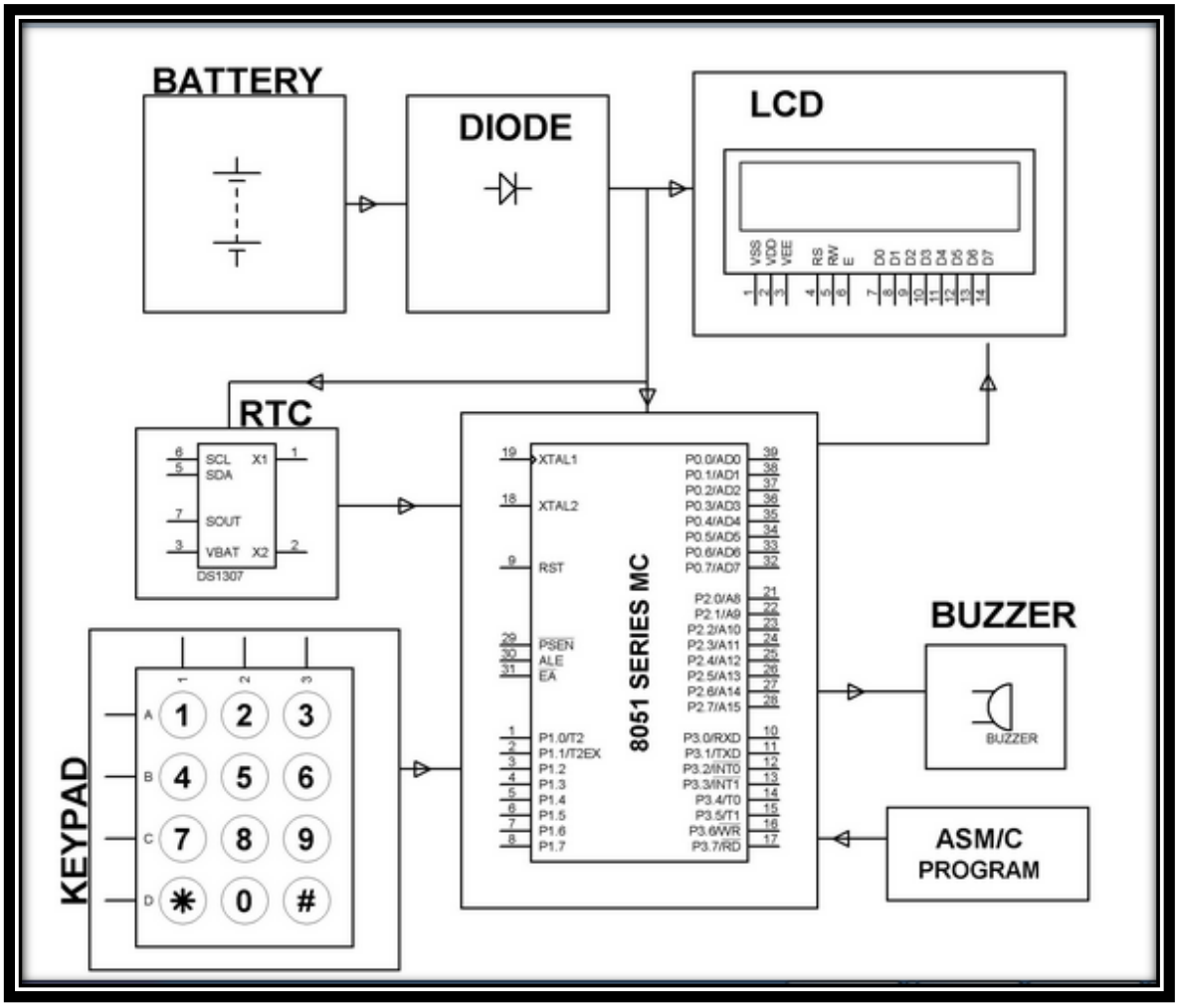

Figure 2.2 the block diagram

In [3], Sometimes patients forget to take the medicine at the required time of medicines. And sometimes patient also forgets which medicine He/She have to take at required time. And it is difficult for Doctor/Compounder to monitor patients around the clock. To avoid this problem, we have made this **medicine reminder system for patients using** Arduino. See next figure 2.3

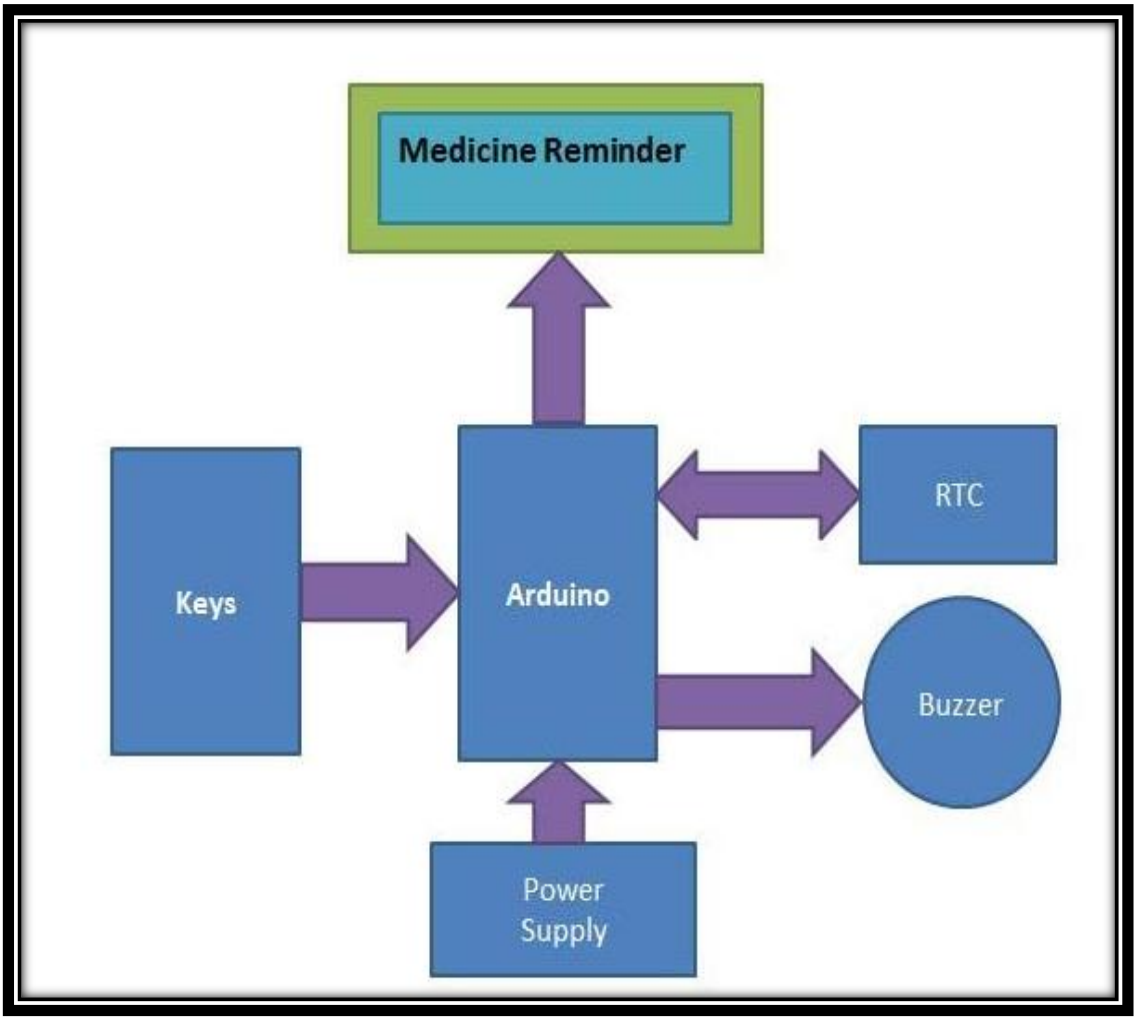

Figure 2.3 medicine reminder system for patients using Arduino.

2.3 microcontrollers

A microcontroller is a compact microcomputer designed to govern the operation of embedded in motor vehicles, robots, office machines, complex medical devices, mobile radio transceivers, vending machines, home appliances, and various other devices. A typical microcontroller includes a processor, memory, and peripherals.

The simplest microcontrollers facilitate the operation of the electromechanical systems found in everyday convenience items. Originally, such use was confined to large machines such as furnaces and automobile engines to optimize efficiency and performance. In recent years, microcontrollers have found their way into common items such as ovens, refrigerators, toasters, clock radios, and lawn watering systems. Microcomputers are also common in office machines such as photocopiers, scanners, fax machines, and printers.

The most sophisticated microcontrollers perform critical functions in aircraft, spacecraft, ocean-going vessels, life-support systems, and robots of all kinds. Medical technology offers especially promising future roles. For example, a microcontroller might regulate the operation of an artificial heart, artificial kidney, or other artificial body organ. Microcomputers can also function with prosthetic devices (artificial limbs). A few medical-science futurists have suggested that mute patients might someday be able, in effect, to speak out loud by thinking of the words they want to utter, while a microcontroller governs the production of audio signals to drive an amplifier and loudspeaker.

Microcomputers enjoy immense popularity among electronics hobbyists and experimenters. Perhaps the most widely known and used of these devices belong to the PIC family, manufactured by Microchip Technology, Inc. of Chandler, Arizona. All devices in the PIC family

come with a wide variety of development tools, are easy to find, remain relatively inexpensive, and have excellent documentation.

2.3.1 microcontroller and microprocessor

A microprocessor (abbreviated as μP or uP) is a computer electronic component made from miniaturized transistors and other circuit elements on a single semiconductor integrated circuit (IC) (microchip or just chip). The central processing unit (CPU) is the most well known microprocessor, but many other components in a computer have them, such as the Graphics Processing Unit (GPU) on a video card. In the world of personal computers, the terms microprocessor and CPU are used interchangeably. At the heart of all personal computers and most workstations sits a microprocessor. Microprocessors also control the logic of almost all digital devices, from clock radios to fuel-injection systems for automobiles.

Microcontroller is a computer-on-a-chip optimized to control electronic devices. It is designed specifically for specific tasks such as controlling a specific system. A microcontroller (sometimes abbreviated μC, uC or MCU) is basically a specialized form of microprocessor that is designed to be self-sufficient and cost-effective. Also, a microcontroller is part of an embedded system, which is essentially the whole circuit board. An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.

Examples of microcontrollers are Microchip's PIC, the 8051, Intel's 80196, and Motorola's 68HCxx series. Microcontrollers which are frequently found in automobiles, office machines, toys, and appliances

are devices which integrate a number of components of a microprocessor system onto a single microchip:

- ❖ The CPU core (microprocessor)
- ❖ Memory (both ROM and RAM)
- ❖ Some parallel digital I/O

The microcontroller sees the integration of a number of useful functions into a single IC package. These functions are:

- ❖ The ability to execute a stored set of instructions to carry out user defined tasks.
- ❖ The ability to be able to access external memory chips to both read and writes data from and to the memory.

The difference between the two is that a microcontroller incorporates features of microprocessor (CPU, ALU, Registers) along with the presence of added features like presence of RAM, ROM, I/O ports, counter, etc. Here a microcontroller controls the operation of a machine using fixed programs stored in ROM that doesn't change with lifetime.

From another view point, the main difference between a typical microprocessor and a micro controller leaving there architectural specifications is the application area of both the devices. Typical microprocessors like the Intel Core family or Pentium family processors or similar processors are in computers as a general purpose programmable device.

2.3.2 Atmega16

Microcontroller is used in many things such as in microwave cooker, telephone, refrigerator, washing machine and television.

2.3.2.1 Functions of atmega16

General types that uses in programming to do every type of tasks such as general microcontroller that can be used in devices as alarm or

control in light. Special function that can execute unique type with capable programming.

2.3.2.2 Pin Configurations

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC Architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.see next figure 2.4

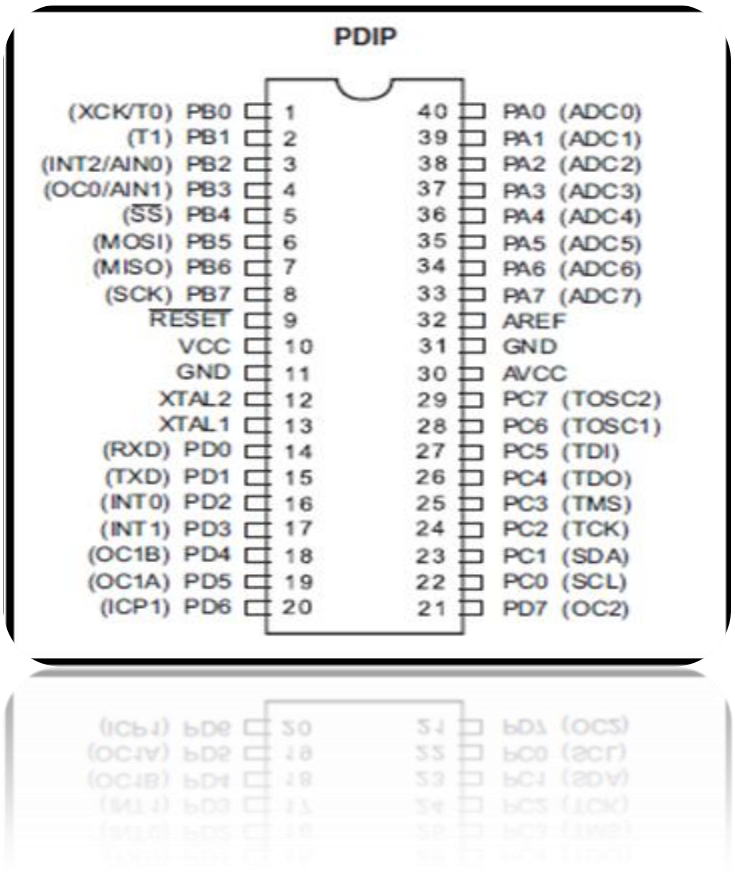

Figure 2.4 pins configuration of atmega 16

2.3.2.3 Atmega16 architecture

Atmega16 microcontroller has the following components:

- Program module
- Languages like C, C++, and Java
- Assembly language requires knowledge of the internals of the

CPU since we are operating at a lower level.

- Machine language is the native language of the CPU
- CPU and register (program counter and stack pointer)

2.3.2.4 Pin description.

- **VCC** Digital supply voltage
- GND Ground
- Port X(X=A,B,C,D)
- Port X (PX7..PX0)
- Port X serves as the analog inputs to the A/D Converter. Port X also serves as an 8-bit bi-directional I/O port if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.
- **RESET** Input. A low level on this pin for longer than the minimum pulse length will generate a reset.
- **XTAL1** Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
- **XTAL2**Output from the inverting Oscillator amplifier
- **AVCC** is the supply voltage pin for Port A and the A/D Converter
- **AREF** is the analog reference pin for the A/D Converter.

2.3.2.5 Timer

For AVR microcontrollers with Timer/Counter Oscillator pins (TOSC1 and TOSC2), the crystal is connected directly between the pins. No external capacitors are needed. The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock source to TOSC1 is not recommended. there are three types of timer timer one , timer two , timer three timer one has 8bit , timer two has 16bit . timer three has 16bit. We use this project timer one.

2.4 liquid-crystal display (LCD)

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. Figure 2.5 show the (lcd) .

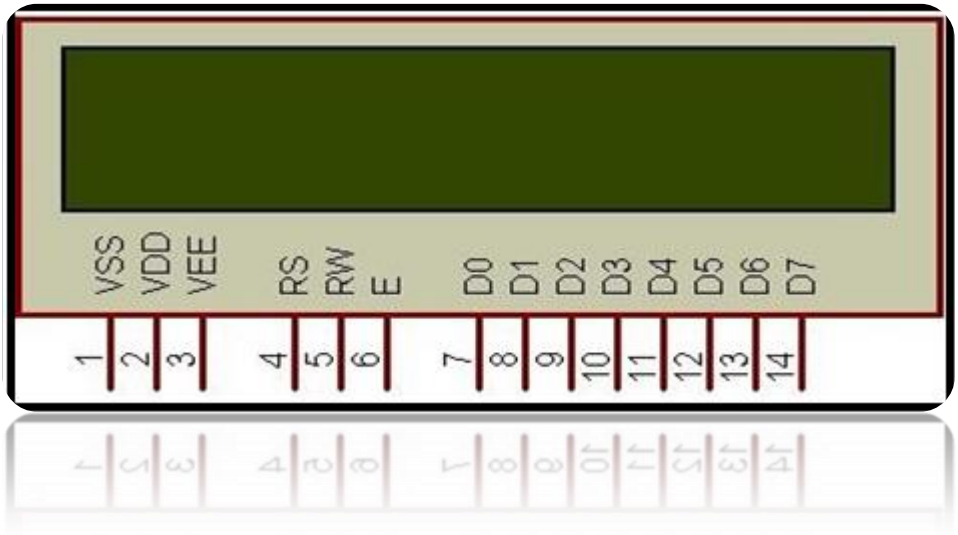

Figure (2.5) liquid-crystal display (LCD).

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on.

2.3.1 Types of LCDs – Passive Matrix

These LCDs use a simple grid to supply the charge to particular pixels on the display. Passive Matrix LCDs start with two glass layers called the substrates.

One substrate is given rows and the other is given the columns, made from a present, that is, only if the row and column are not on a low or high level at the same time. More precisely, the pixel is selected if the RMS voltage is above the threshold for reorientation

In a passive dual-scan modulator the number of pixel can be doubled without loss in optical contrast by cutting the row stripes in the center of the display and supplying two strobe signals at each half. In twisted nomadic displays the liquid crystal molecules lie parallel to the glass plates, and the glass is specially treated so that the crystal is forced to point a particular direction near one of the plates and perpendicular to that direction near the other plate. This forces the director to twist by 90° from the back to the front of the display, forming a helical structure similar to chiral nematic liquid crystals. In fact, some chiral nematic crystal is added to make sure all of the twists go the same direction.

The thin film of twisted nematic liquid crystal is circularly birefringent. When linearly polarized light passes through, the optical activity of the material causes the polarization of the Twisted nematic displays are simple in architecture, cheap and easy to manufacture.

Table 2.2 pin descriptions

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6

2.3.2 Operation of LCD

LCD, this does not mean that the liquid material where it's scientifically material between the liquid steel. We will not get in the scientific details. This is important in how to operate the screen.

1 - All computer monitors paint a picture in the form of tiny dots side by side made up the picture and called every point Pixel

2 - There are three primary colors from which all the colors you see are the eye and the red, green and blue are called RGB.

2.3.3 The concept of the LCD

This type consists of screens of some of the basic components:

1. The source of light polarization met
2. Liquid crystals can send and change the polarization of light
3. Transparent material and are connected to the electrical

When light falls on a glass slide, they are working on the initial polarization of the light, and then operate the molecules of liquid crystals in each layer to direct the light to the class the next level with the change of polarization of light. And when it reaches the last layer of light layers of liquid crystals, it would be polarized in the same direction molecules that layer and thus performs light them.

2.4 Definition of Protues

Protues is a virtual system modeling (VSM) that combines a circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller basic designs, this is the prefect tool for engineers to test their microcontroller designs before constructing a physical prototype in a real time.

This program allows users to interact with the design using on screen indicators and/or LED and LCD displays , Protues in VSM comes with extensive debugging features including break points , single steping and variable display for neat design period to hardware prototyping .

In summary protues is the program to use when you want to simulate the interaction between software running on microcontroller and any analogue or digital electronic device connected to it.

- **Advantages of protues :**
  - Real time simulation
  - Time and money saving

There are some important steps that take when you want to design a circuit in protues simulation softwere.

2.5 Definition of bascom

BASCOM-AVR is the original Windows BASIC COMPILER for the **AVR** family. It is designed to run on XP/**VISTA/WIN7 and WIN8** BASCOM AVR is a very powerful and easy-to-use compiler for the AVR series of micro controllers developed by Atmel. The program comes with a very user-friendly interface and a set of simple commands, and provides more flexibility than other programs in this category. Anybody with some basic knowledge of C or C++ can write a successful program using BASCOM, as most of its functions and its statement structure is similar to those in C.

To make a program takes just a few steps :

- Write the program in BASIC
- Compile it to fast machine binary code
- Test the result with the integrated simulator(with additional hardware you can simulate the hardware too).
- Program the chip with one of the integrated programmers. (hardware must be purchased separately) [4].

CHAPTER

THREE

SYSTEM DESIGN

3. System design

3.1 Over view

This design contains parts of the basic circuit, Switches for Edit Time Alarm Medicines, Real Time Circuit, Main Unit one and two Microcontroller Atmega 16, Display Unit one and two, transmission (TX, RX) and boxes and its leds. The figure 3.1 below shows the Block Diagram contains all the basic components.

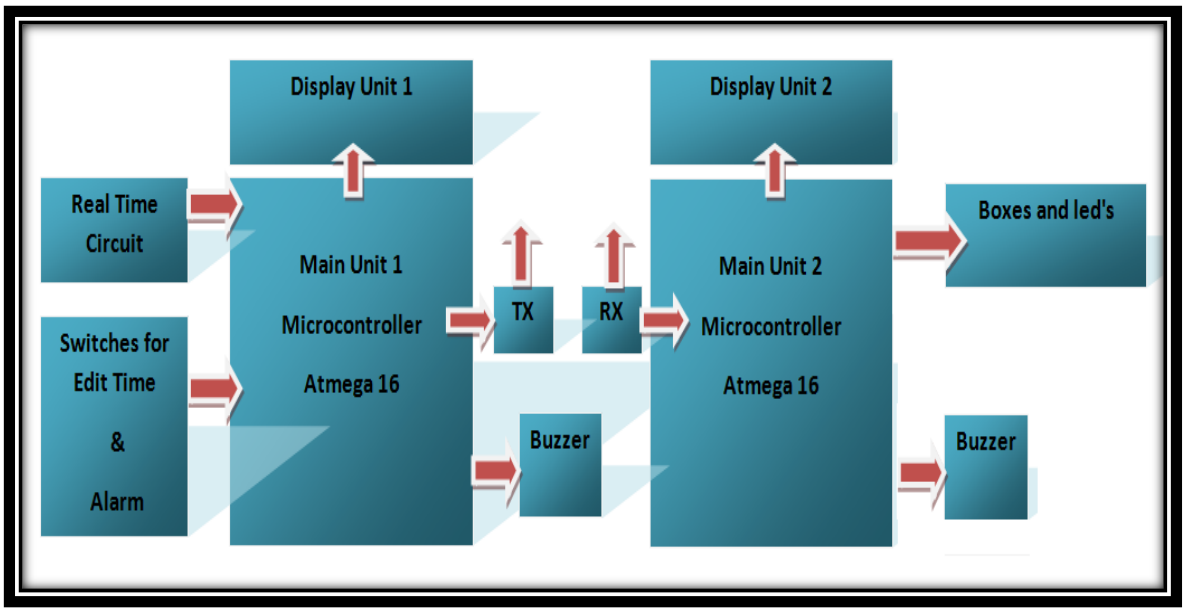

Figure 3.1 block diagram of system design.

- ❖ Real Time Circuit (RTC):

Real time circuit RTC its save the time in side and give it to the microcontroller for monitoring and alarm its connect by protocol I2C.

- ❖ Switches:

for Edit Time of RTC and the time for all patients.

- ❖ Main Unit one Microcontroller Atmega 16:

The data time came from the switches. This data time of patient will be stored in the microcontroller. The microcontroller continuously reads the

time from the RTC , When the timings read from the RTC equals the timings stored in the Microcontroller the alarm well turn on and inform the doctor for medicine time and send the alarm by wireless.

- ❖ Alarm Medicine for Doctor:

After the microcontroller is compare the time of RTC and the patient if it same time the Doctor buzzer is alert.

- ❖ Lcd in side of Transmitter:

Its display the patient alarm number (one , two, and three) and RTC time for edit and display the time.

- ❖ Transmitter And Receiver Unit:

Transmitter Send that Alarm patient to the side of receiver to turn on the box Medicine.

- ❖ Main Unit Two Microcontroller Atmega 16:

Its receive the data through the serial communication and give control signal of desired box and signal of patient Buzzer.

- ❖ Alarm Medicine for Patient:

After the microcontroller is receive the signal data its Alerts the Patient buzzer and turn on led box.

- ❖ Lcd in side of Receiver:

Its display the patient alarm.

3.2 DS1307-RTC

Areal-time clock (**RTC**) is a computer clock (most often in the form of an integrated circuit) that keeps track of the current time. Although the term often refers to the devices in personal computers, servers and embedded systems, RTCs are present in almost any electronic device which needs to keep accurate time, Although keeping time can be done without an RTC, using one has benefits:

- ❖ Low power consumption (important when running from alternate power)
- ❖ Frees the main system for time-critical tasks
- ❖ Sometimes more accurate than other methods

Most RTCs use a crystal oscillator, but some use the power line frequency. In many cases the oscillator's frequency is 32.768 kHz. This is the same frequency used in quartz clocks and watches, and for the same reasons, namely that the frequency is exactly 2^{15} cycles per second, which is a convenient rate to use with simple binary counter circuits, figure 3.2 show that.

Figure 3.2 DS1307-RTC

3.2.1 SOME FEATURES OF DS1307

Real time clock counts seconds, minutes ,hours , date of month ,day of week and year with leap year compensation valid up to 2100

- 56 byte nonvolatile RAM for general data storage .
- 2-wrire interface (I2C) .
- Automatic power fail detect .
- Consumes less than 500 nA for battery back-up. next figure 3.3 show ds1307 and connection.

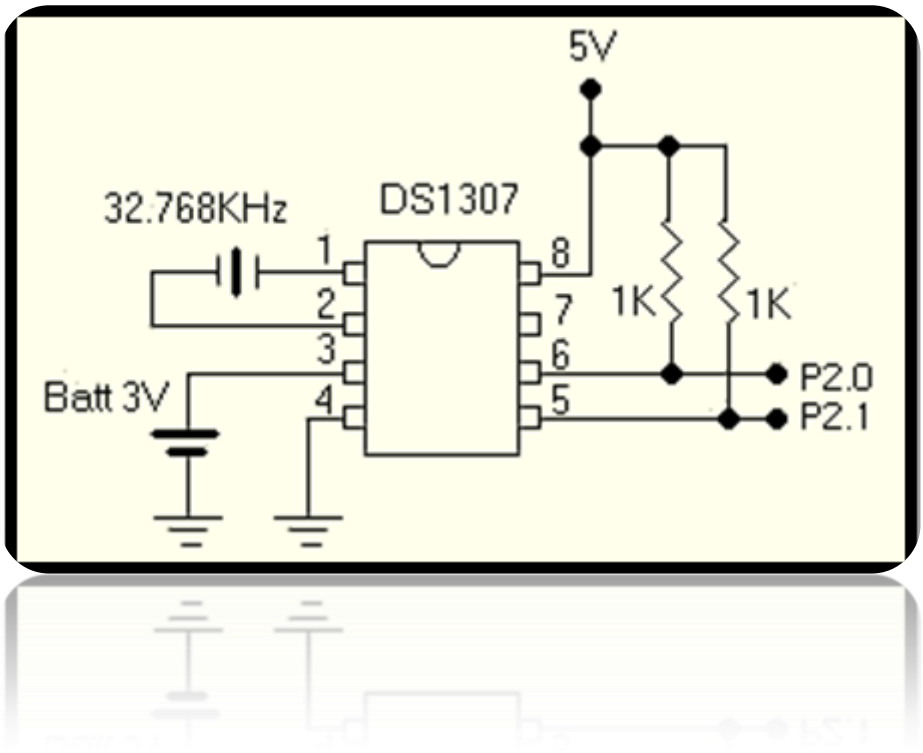

figure 3.3 show ds1307 and connection.

3.2.2 The I2C protocol (TWI)

- After the connection between the microcontroller and the RTC and detect the master and the slave and through the bits of (read ,1, or write ,0,) in address of bit(R/W).
- Every time the master send 8 bits to…
- The receiver whatever (master or slave) send back (ACK).
- When the transmission between the (master and slave) the master to send stop bit. The next figure 3.4 show that.

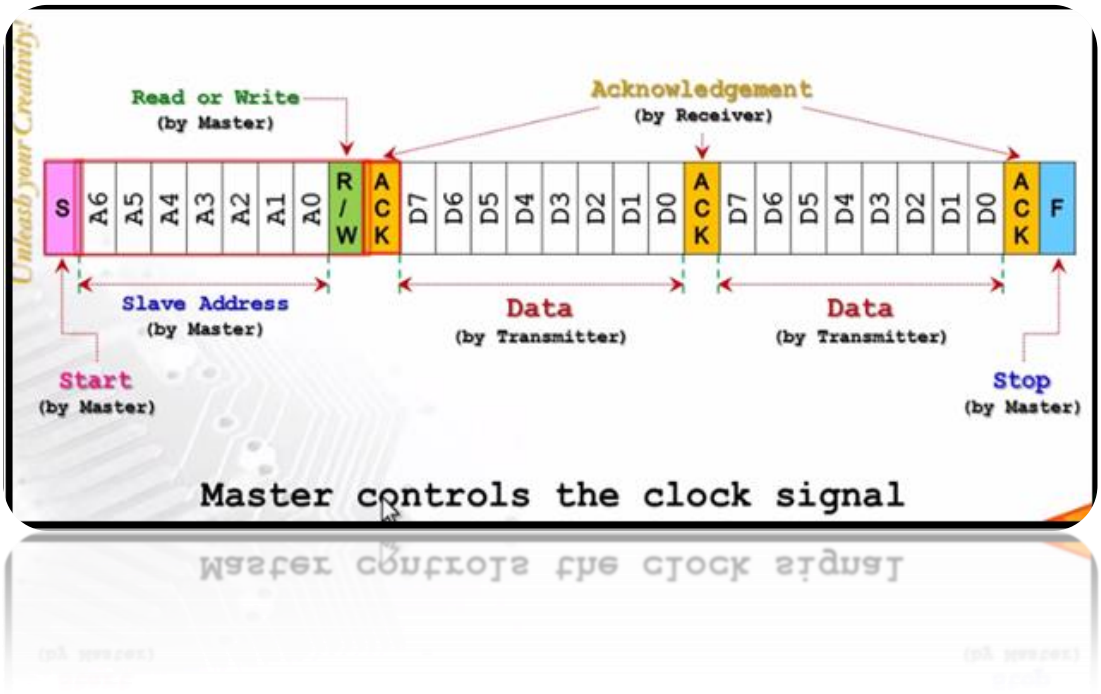

figure 3.4 the I2C protocol.

3.3 Explains The Block Diagram and hardware in the simulation:

3.3.1 Main Unit Microcontroller Atmega 16:

The data time came from the switches. This data time of patient will be stored in the microcontroller. The microcontroller continuously reads the time from the RTC .When the timings read from the RTC equals the timings stored in the Microcontrollerits alarm and turn on the led of box and send the information via wirless in the sid of the patient. See next figure 3.5

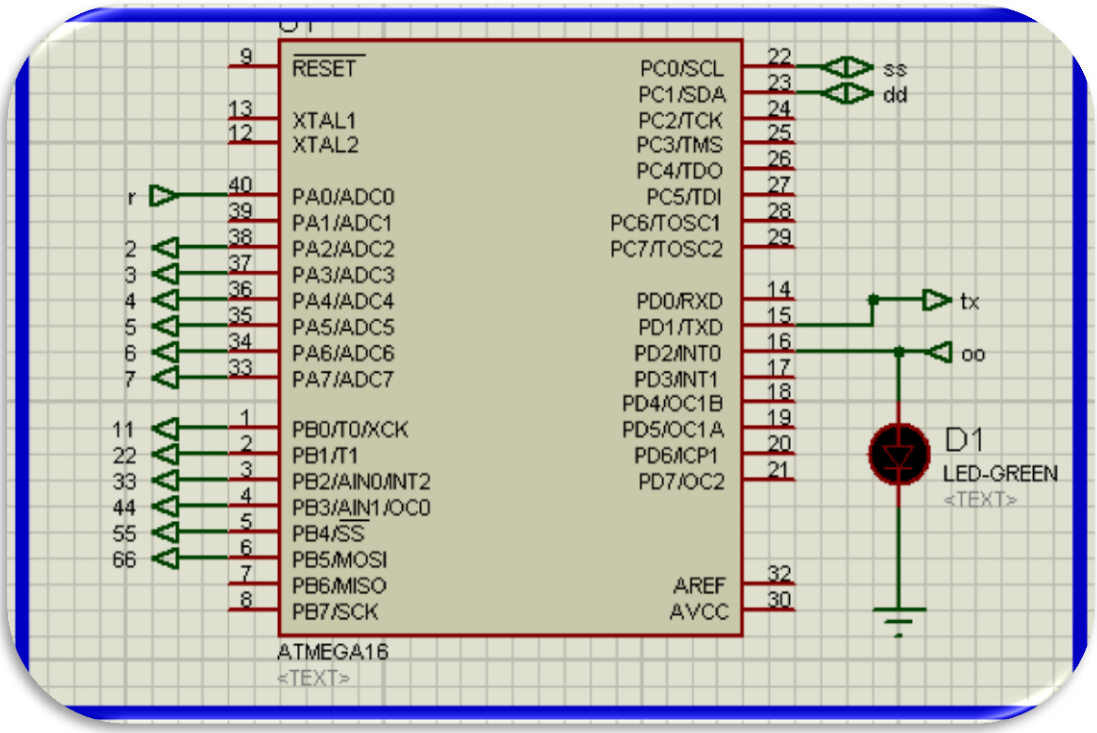

Figure 3.5 The microcontroller in side of doctor in protues.

3.3.2 Real Time Circuit (RTC):

Real time circuit RTC its save the time in side and give it to the microcontroller for monitoring and alarm its connect by protocol I2C.see next figure, The next table show the RTC connection with microcontroller.

Table 3.1 RTC connection with microcontroller

RTC pin	Pin microcontroller
Pin 6 scl	Pin 22 port c0
Pin 5 sda	Pin 23 port c1

3.3.2.1 Battery:

The Battery is very importance in the system because if the power turn of the battery give power to RTC to calculate the time and not lost. its connect with pin 3 in RTC. See the next figure 3.6 shows the RTC in the system.

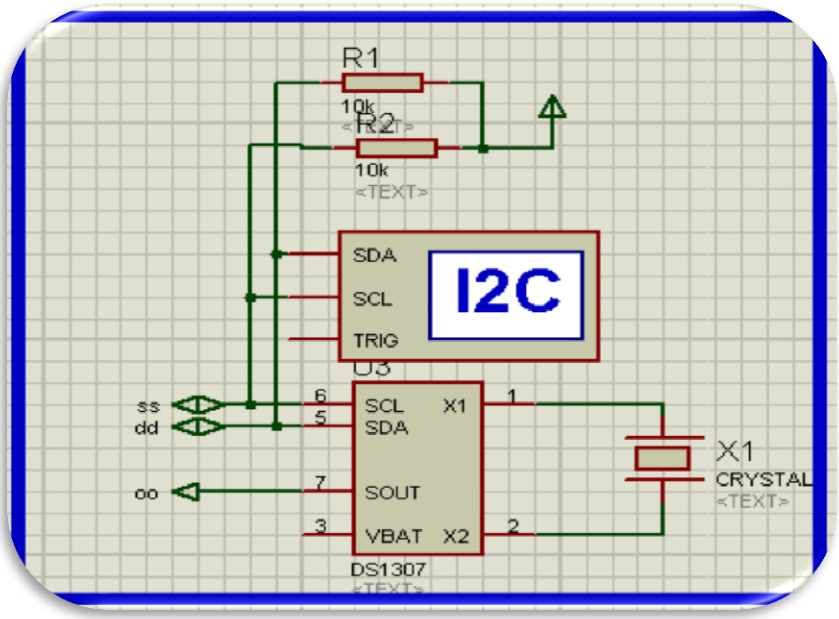

Figure 3.6 show the **RTC** in the system.

3.3.3 Display Unit (LCD) in side of the doctor:

its Display time RTC and edit time patients (time Alarm). see next figure 3.7, the next table show the connection of Lcd with the microcontroller.

Table 3.2 connection of Lcd with the microcontroller

Lcd Pin	Microcontroller pin
D7 pin 14	Pin 33 port a7
D6pin 13	Pin 34 port a6
D5 pin 12	Pin 35port a5
D4 pin 11	Pin 36 port a4
E pin 6	Pin 38 port a2
RS pin 4	Pin 37 porta3
VCC pin 2	Power
Vss vee RW (1, 3, 5)	GND

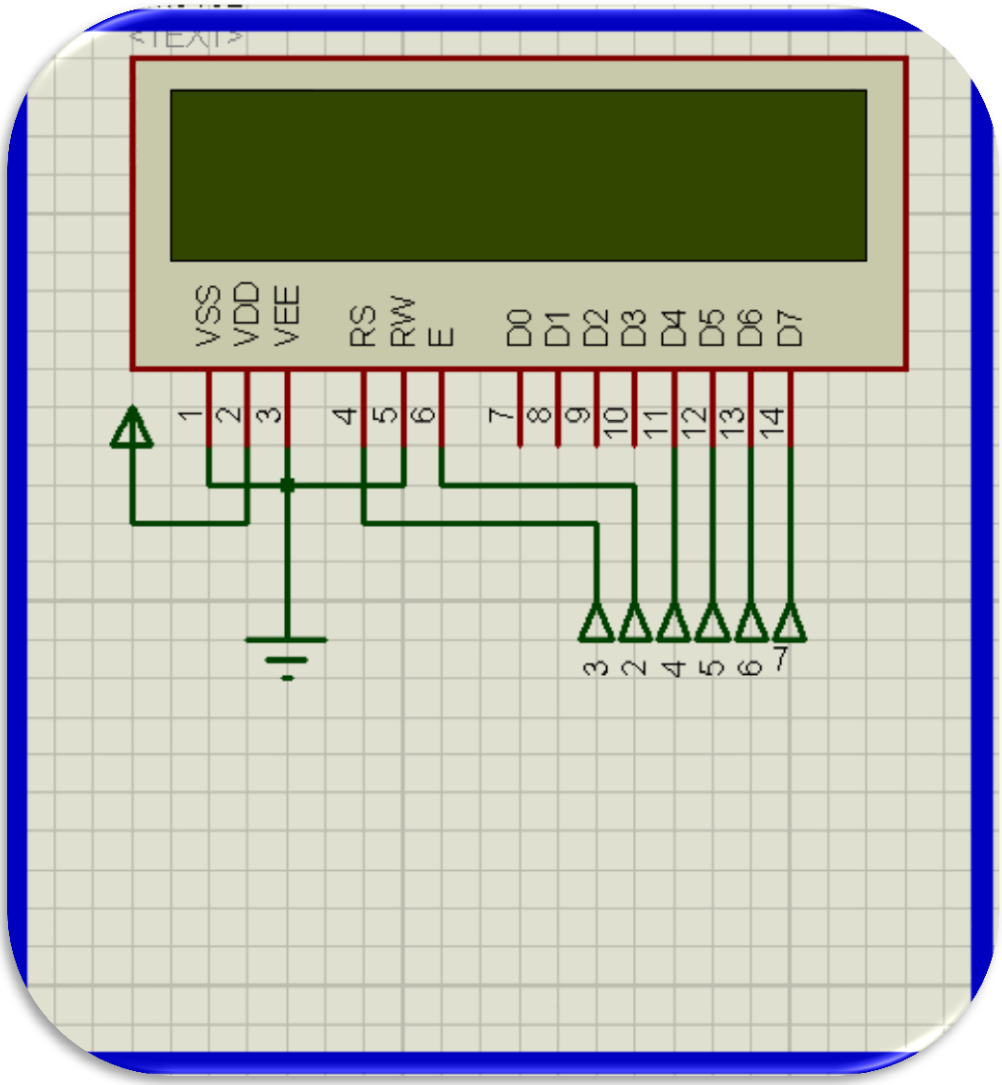

Figure 3.7 display unit LCD in side of the doctor.

3.3.4 Switches:

for Edit Time of RTC and the time for all container, see next figure 3.8 the next table show the connection:

Table 3.3 connection with switches:

Switch	Pin Port number
Sw1 (yy++)	Pin 1 Port b0
Sw2 (yy--)	Pin 2 Port b1
Sw3(xx++)	Pin 3 Port b2
sw4(xx--)	Pin 4 Port b3
Sw5 (increment value for position of xx)	Pin 5 Port b4
Sw6 (decrement for position of xx)	Pin 6 Port b5
Sw7 write the time in the RTV	Pin 40 Port a0

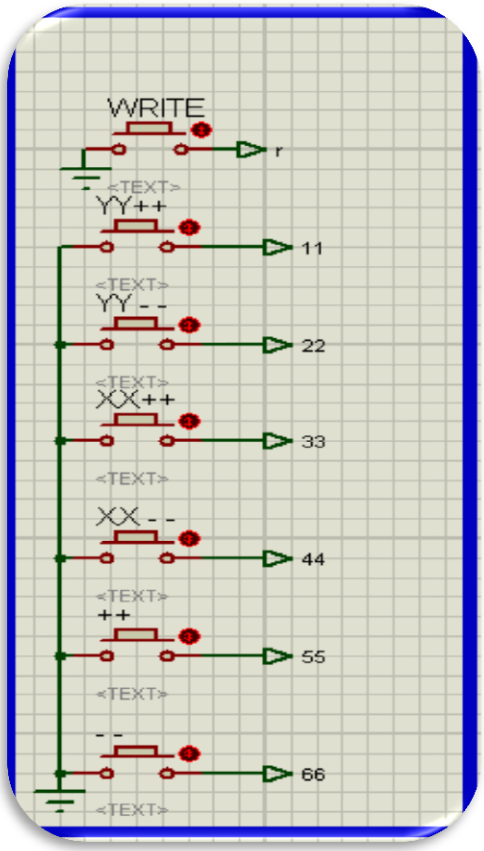

Figure 3.8 switches for edit time.

3.3.4.1 Condition of work and switch:

Two switch for increment and decrement yy.

Two switch for increment and decrement xx.

Two switch for increment and decrement the value number with position xx.

When yy=0 the lcd display the time of RTC.

When yy=1 edit the time to be interred in the RTC.

When yy=2 edit patient alarm one (hours and minute).

When yy=3 edit patient alarm two (hours and minute).

When yy=4 edit patient alarm three (hours and minute.)

Xx= its change in value form 0 to 5 to adjest year, month, day, hour,minite and secound.

3.3.5 Buzzer sound Alarm in side of the doctor:

After the microcontroller is compare the time of RTC and the patient if it same time, the alarm sound in side of doctor is turn on. see next figure 3.9.

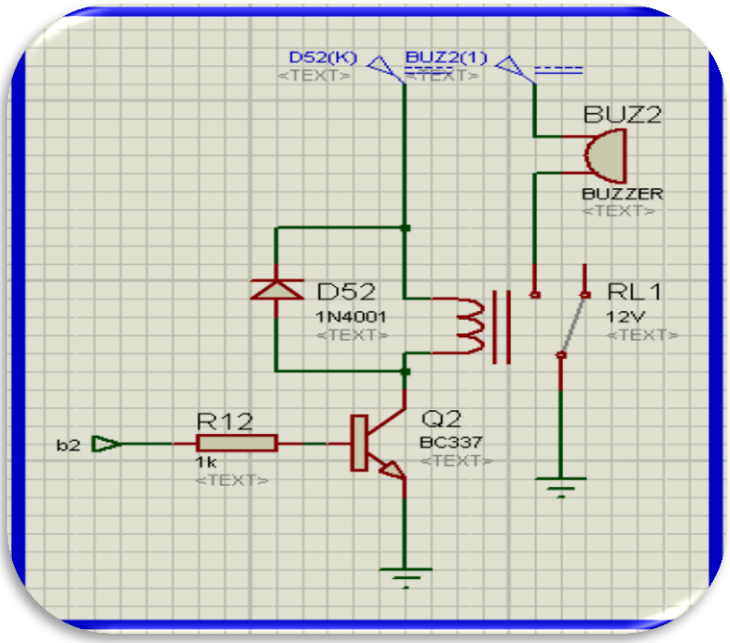

Figure 3.9 Buzzer sound Alarm in side of the doctor.

3.3.6 Boxes and led's:

There is three boxes in every box is medicine when time of medicine came the green led box well turn on to inform the right box. next figure its illesrate three led in side of the doctor and other three leds in side of the patient , The next table show the connection with microcontroller.

Table 3.4 connection with microcontroller

Led's	Pin microcontroller
Container led one in side of doctor	Pin 21 port d7
Container led two in side of doctor	Pin 20 port d6
Container led three in side of doctor	Pin 19 portd5
Container led one in side of patint	Pin 21 port a0
Container led two in side of patint	Pin 20 port a1
Container led three in side of patint	Pin 19 port a2

The next figure 3.10 show the leds in two side doctor and patient.

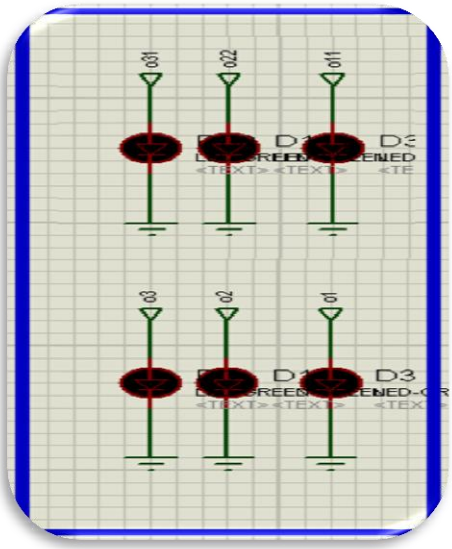

Figure 3.10 leds in two side doctor and patient.

The wirless system:

The wirless system consist of transmitter and recever RF by using modulation teqnice Amplitude shift key (ASK) to connect between the doctor and the patient in the hospital or in home. The next figure 3. 11 show the wireless transmitter and the receiver in simulation.

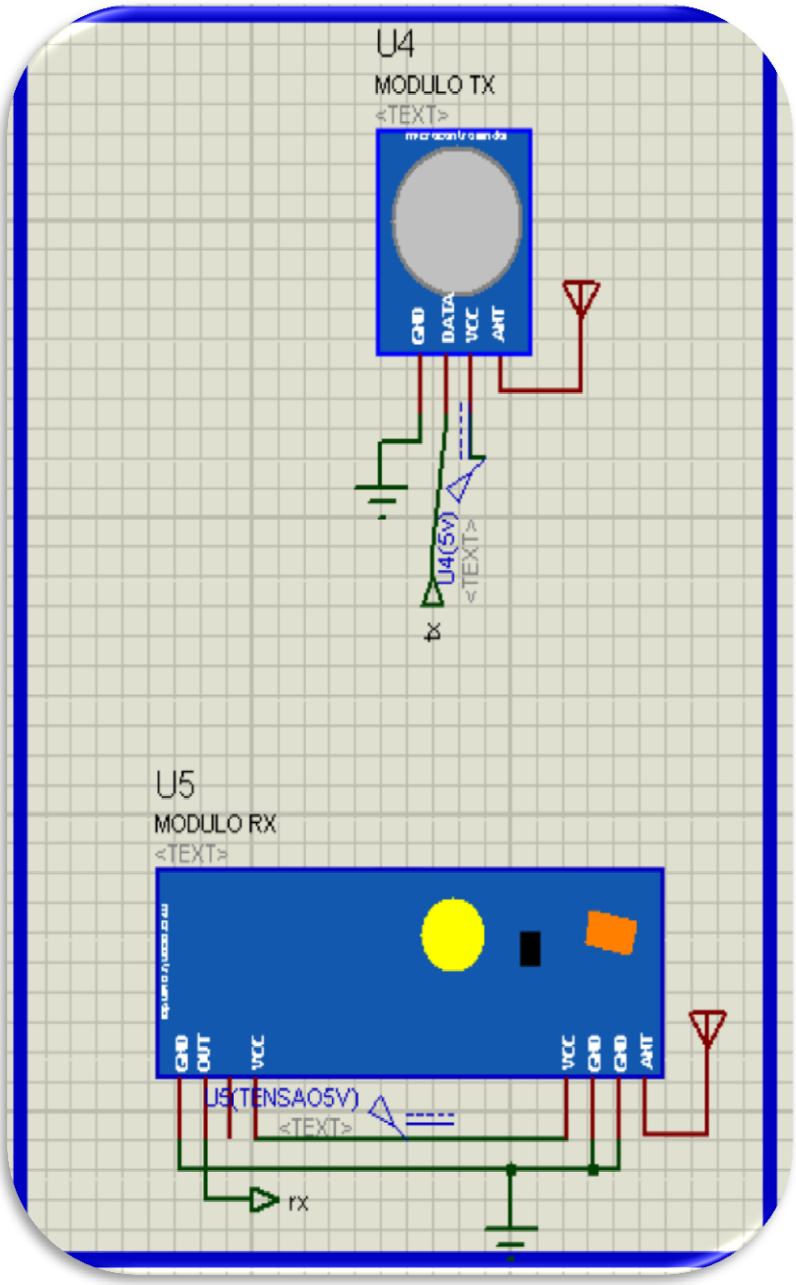

Figure 3.11 the wireless transmitter and the receiver in simulation.

3.3.7 The main unit two Microcontroller in side of the patient:

The microcontroller receive the data patient alarm from the side of the doctor by wireless communication via serial communication and alert the box of the medicine and buzzer sound alarm and display that in the LCD, the next figure 3.12 show the microcontroller in the side of the patient.

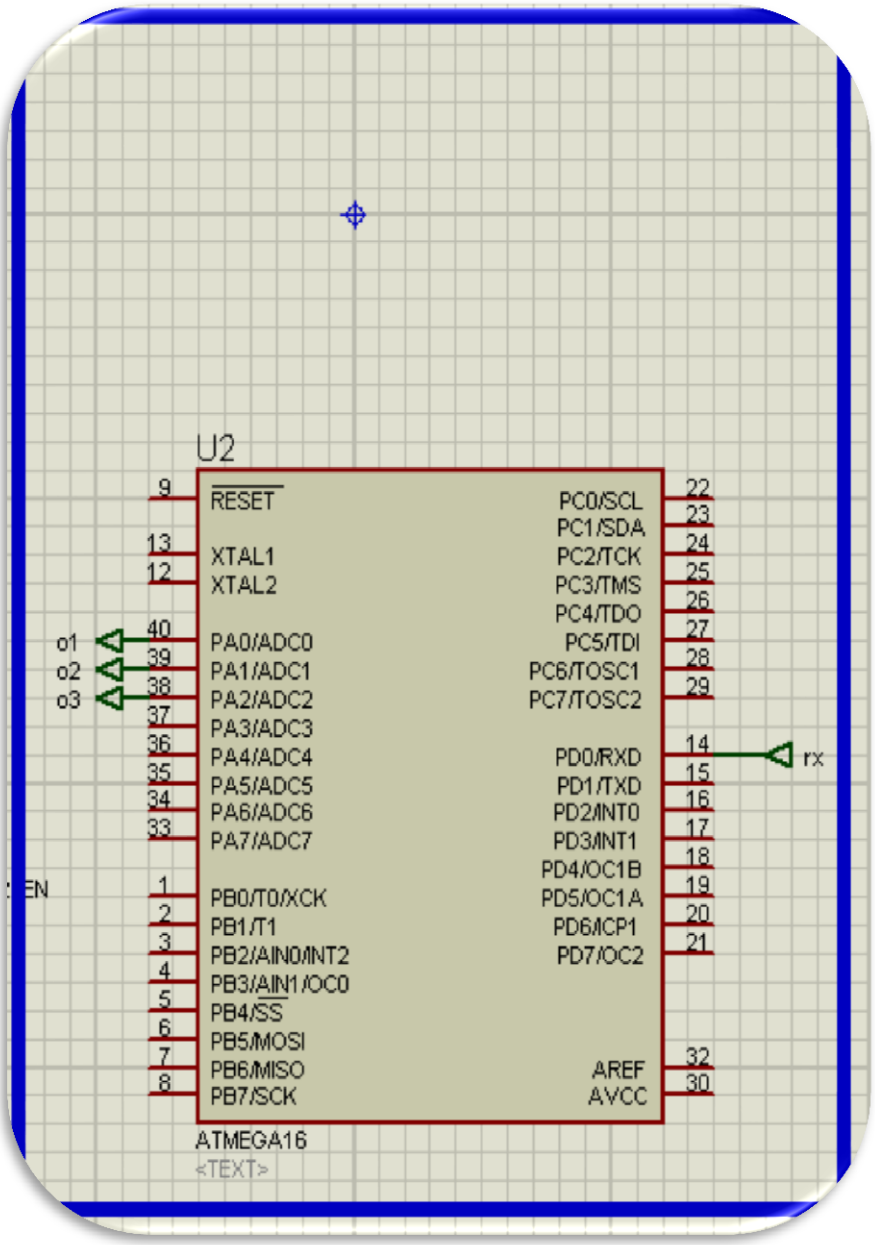

Figure 3.12 The microcontroller in side of the patient.

3.3.8 Buzzer sound Alarm in side of the patient:

After the microcontroller receive the patient alarm from side of the doctor the led of patient is turn on and output signal to the buzzer, see next figure 3.13 show the buzzer in side of the patient.

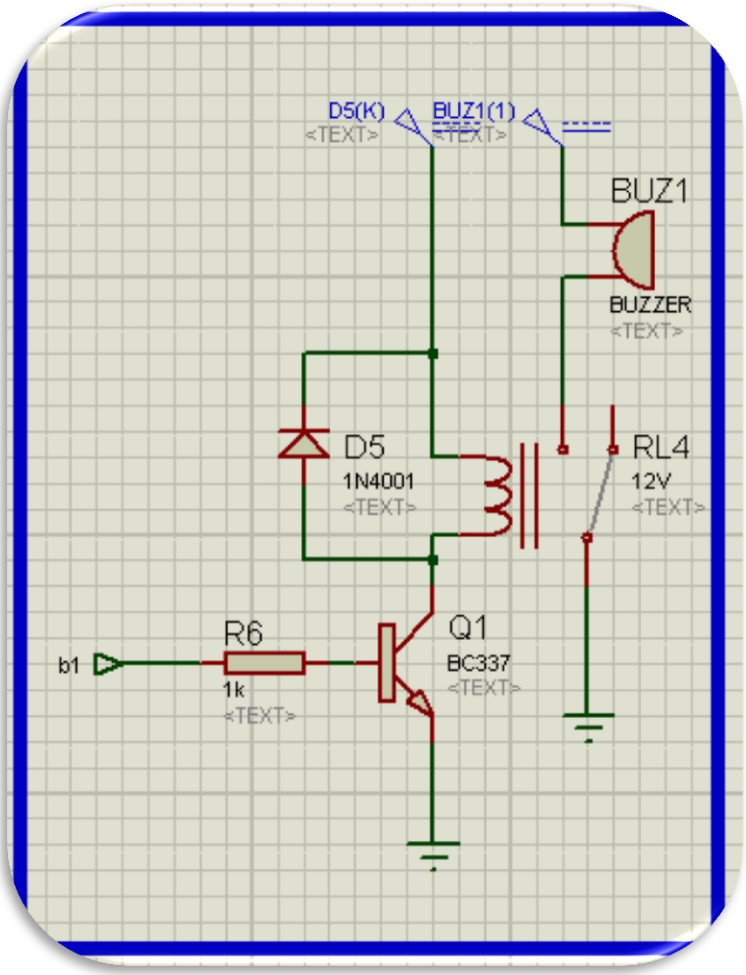

Figure 3.13 the buzzer alarm in side of the patient.

3.3.9 Display unit (LCD) in side of the patient:

Its display the patient alarm box number (one, two and three) to inform the patient the right box and if there is no alarm its show that no alarm see next figure 3.14.

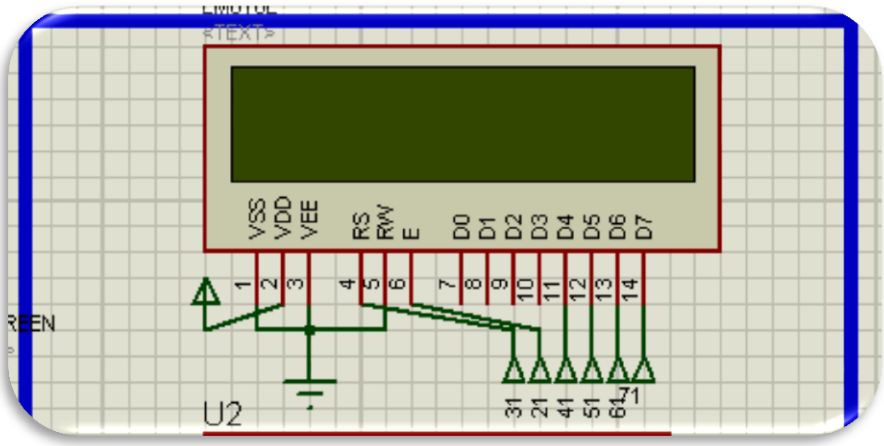

Figure 3.14 the LCD in side of the patient.

CHAPTER

FOUR

SOFTWARE AND PROGRAM

4.1 Overview

Those days, electronic circuits consist of numerous CMOS and TTL chips. Chip 89C2051 as an ideal replacement for a lot of CMOS/TTL chips. No one was able to design his own chips !

Initially wrote a complete tool for DOS. rewrote the tool, .after that , MCS started to sell BASCOM-LT, a BASIC compiler for Windows 3.1 . It was the first Windows application that offered a complete and affordable solution, editor, compiler, simulator and programmer .

BASCOM-LT was an 8051, using an LCD display was simple, just a configure tion line to define the used pins and voila, a working application in minutes. When you needed a different LCD display, you could simply change the CONFIG line .When a different processor was needed, you only had to change the name of the definition file, no need for a lot of .h files .

4.2 INTRODUCTION

Programming language is a set of commands and rules written in specific way to build a program. The program consist of a sequence of commands written in the programming language, which the circuit or mainly the microcontroller do one after other.

Compiler is a program used in computer to compile the program to the matching language (0, 1). The programmer is a device used to download the hex files from the computer to the flash memory in the microcontroller.

4.3 BASCOM CHARACTERS

Characters from the BASCOM character set are put together to form labels, keywords, variables and operators.

This is description of the format of BASCOM program lines:

- The specific characters in the character set and the special meanings of some characters.
- The format of a line in a BASCOM program.
- Line labels.
- Program line length.

4.4 Expressions and Operators

How expressions combine, modify, compare, or get information by using the operators available in BASCO following kind of operators:

- Arithmetic operators, used to perform calculations.
- Relational operators, used to compare numeric or string values.
- Logical operators, used to test conditions or manipulate individual bits.
- Functional operators, used to supplement simple operators.

Here is a fragment from the Microsoft knowledge base about FP:

1. Floating-point concepts.

It is very important to realize that any binary floating-point system can represent only a finite number of floating-point values in exact form. All other values must be approximated by the closest represent able value. BASCOM supports the standard and the method for rounding value according to the IEEE rules:

- Rounding

When a Long is assigned to a single, the number is rounded according to the rules of the IEEE committee

- Double

The double is essential the same as a single. Except the double consist of 8 bytes instead of 4. The exponent is 11 bits leaving 52 bits for the mantissa .

- Arrays

An array is a set of sequentially indexed elements having the same type. Each element of it has a unique index number that identifies it.

- Strings

A string is used to store text. A string must be dimensioned with the length specified.

DIMS as STRING5* Will create a string that can store a text with a maximum length of 5 bytes.

- Casting

In BASCOM-AVR when you perform operations on variables they all must be of the same data type.

Long = long1 * long2 ' for example

- Single Conversion

When you want to convert a SINGLE into a byte, word, integer or long the compiler will automatic convert the values when the source string is of the SINGLE data type.

Integer = single

You can also convert a byte, word, integer or long into a SINGLE by assigning this variable to a SINGLE.

Single = long

4.5 Writing and compiling the program:

The program used was BASCOM language. Any program in this language will save in a life with extension name –bas. After writing and save the program it must be compiled by press compiler icon. The compiling is completing in two steps:

- Convert (bas) extension file to assembly code.
- The compiler automatically convert the assembly code to executable HEX file can be downloaded to the microcontroller.

We cannot observe any delay between the two steps because the operation of converting the assembly code to HEX file is automatic and undeliverable operation.

If there is syntax Error in the program, the HEX file cannot be found.

BASCOM – AVR is a language used to programming Atmel family microcontrollers. The good compiling operation in BASCOM produces many files:

- xxx. Bas – program file
- xxx. HEX – Intel Hexadecimal file, which is needed by some programmers.

- xxx. BIN – Binary file which can be programmed into microprocessor.
- xxx. DBG – Debug file that is needed by simulator.
- Xxx .OPG – object file for simulating using AVR studio. Also needed by the internal simulator.
- xxx . ERR – Error file. Only created when errors are found.
- xxx . RPT – Report file.
- xxx . EEP – EEPROM image file.

4.6 Structures

The program is beginning with various based instructions. Because of the huge number of instructions in BASCOM, we can only discuss the instructions that we used in our programs:

$regfile

Action:-

Instruct the compiler to use the specified register file instead of the selected dat file

Syntax:

$ Regfile = “name”

Remarks:

Name ➔ the name of the register file .The register files are stored in the BASCOM – AVR application directory and they all have DAT extension.

The register file holds information about the chip such as the internal registers and interrupts addresses.

The register file info is derived from Atmel definition files .

$crystal

Action:

Instruct the compiler to override the crystal frequency option setting.

Syntax:

Crystal = var

Remarks”

Var = A numeric constant with the frequency of the crystal.

Config port

Action:

Sets the port or port pin to the right data direction.

Syntax:

Config port x = state

Config port x.y = state

Remarks:

State = A numeric constant that can be input or output. Input will set the data direction register to input for portx. Output will set the data direction to output for portx. You can also use a number for state. B0001111 will set the upper nibble to input and the lower nibble to output.

You can also set one port pin with the (config pin = state), statement. Again you can use input, output or a number in this case the number can be only (0-1).

Do – Loop:

Action:

Repeat a block of statements until condition is true.

Syntax:

Do statements

Loop [until expression]

Remarks:

You can exit Do – Loop with the exit statement, The Do – Loop is always performed at least once.

End

Action: Terminate program execution.

Syntax: End

Remarks: Stop can also be used to terminate program.

If – Then

Action: Allow condition execution or branching, based on the evaluation of Boolean expression.

Syntax: If expression Then

End If

Remarks:

Expression = any expression that evaluates to true or false.

Waitus

Action: Suspends program execution for a given time in microsecond.

Syntax: Waitus μs

Remarks: μs = The number of microseconds to wait (1 – 65535). This must be a constant. Not a variable.

Waitms

Action: Suspends program execution for a given time in ms

Syntax: Waitms ms

Remarks:

Ms = the number of milliseconds to wait (1 – 65535).

While – Wend

Action:

Executes a series of statements in a loop, as long as given condition is true.

Syntax:

While condition

Statements

Wend

Remarks:

If the condition is true the any intervening statements are executed until the wend statement is encountered. BASCOM then return to the while statement and check the condition. If it is still true, the process is repeated. If it is not true, execution resumes with the statement following the wend statement,

Dim

Action:

Dimension à variable

Syntaxe:

Dim var As [XRAM / IRAM] type [To Location]

Remarks:

Var = any valid variable name such as b1, i or long name. Var can also be an array: ar (10) for example.

Type = Bit, Byte, Word, Integer, Long, Single or string.

4.7 Definition of Protues

Protues is a virtual system modeling (VSM) that combines a circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller basic designs, this is the prefect tool for engineers to test their microcontroller designs before constructing a physical prototype in a real time.

This program allows users to interact with the design using on screen indicators and/or LED and LCD displays , Protues in VSM

comes with extensive debugging features including break points , single steping and variable display for neat design period to hardware prototyping .

In summary protues is the program to use when you want to simulate the interaction between software running on microcontroller and any analog or digital electronic device connected to it.

Advantages of protues :

- ❖ Real time simulation
- ❖ Time and money saving

There are some important steps that take when you want to design a circuit in protues simulation software.

4.8 Flow chart

4.8.1 The program of flow chart description:

First Adjust the RTC time, patient alarm one, two and three, and turn buzzers off. After that the microcontroller Read RTC time every second to be Compared the time RTC with the time patient alarm one, two, three or go back read and adjust the times. And after that Turn on the container led and buzzer of any matched time of doctor alarm one, two, three with RTC time and then send that wirelessly to the side of the patient and turn on the led of matched time and buzzer then Return to the start to repeat all that, The next figure 3.15 show the flow chart of the system.

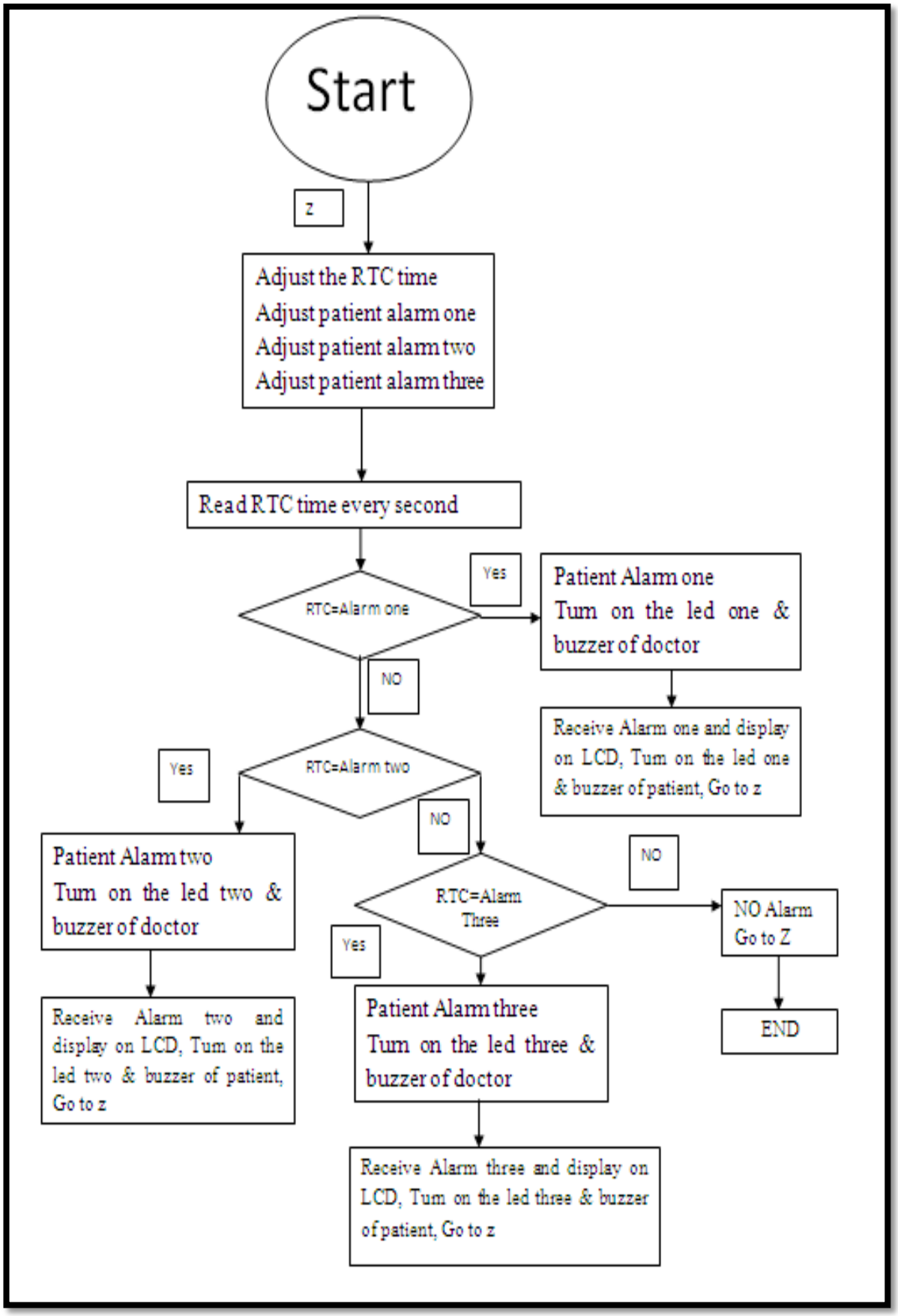

Figure 4.1 show the flow chart of the system.

CHAPTER

FIVE

SIMULATION AND RESULT

5. Simulation and Result

5.1 Over view

This chapter contains the obtained results of the circuit in all stages of the process and the details of each phase sensitive separately, where displays the following:

1. The results obtained from the RTC.

2-The results obtained from the switch to adjust the time and medicine patient.

3. The results obtained for the boxes and alarm in side of doctor and patient.

5.2 The results obtained from the RTC

The next figure 5.1 show the results obtained from the RTC witch its displayed the time read from the RTC IC in the LCD.

- The yy is equal to zero.

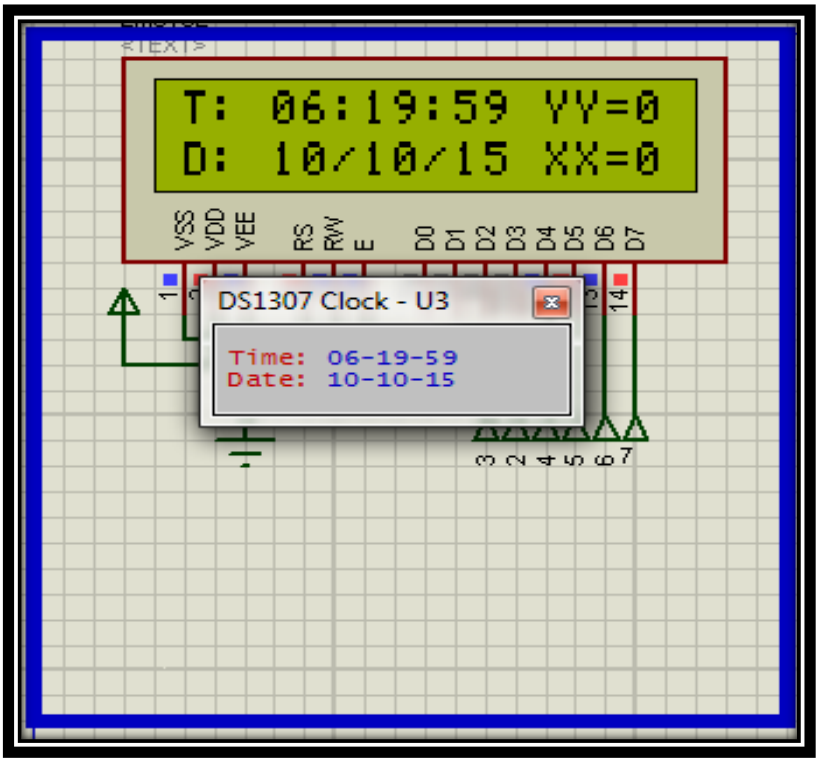

figure 5.1 the results obtained from the RTC.

5.3 The adjust of the time for the RTC

The next figure 5.2 show the results obtained for adjust the RTC witch its displayed the time to be write in the RTC IC in the LCD.

- The yy is equal to one.
- The value xx change from zero to 5 to adjust the time

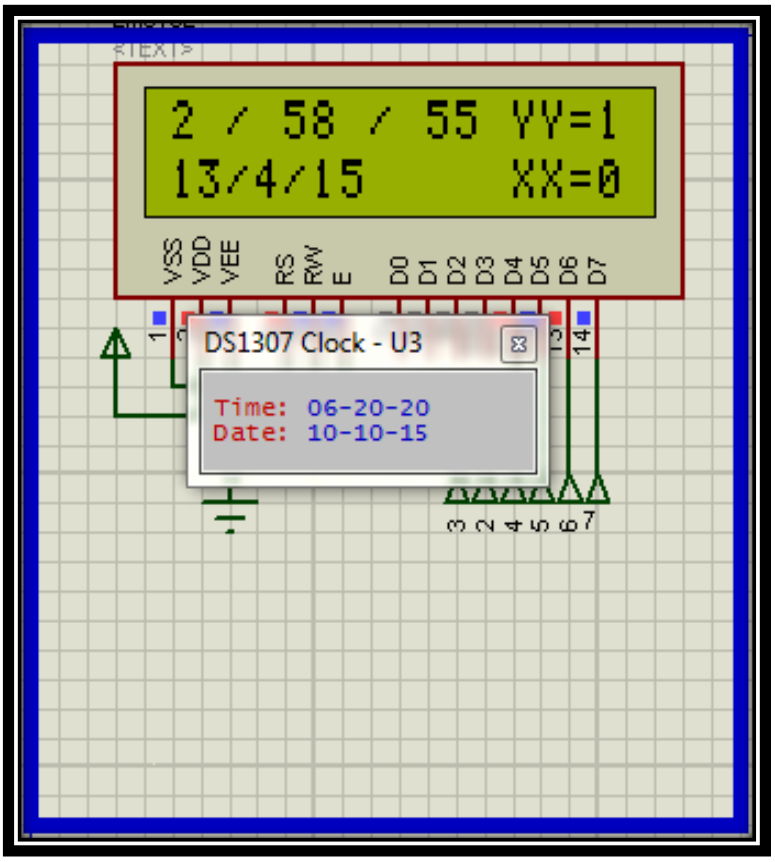

figure 5.2 show the results obtained for adjust the RTC

The next figure 5.3 show the results obtained for adjust the RTC witch its displayed the time witch has been written in the RTC (ds1307) IC.

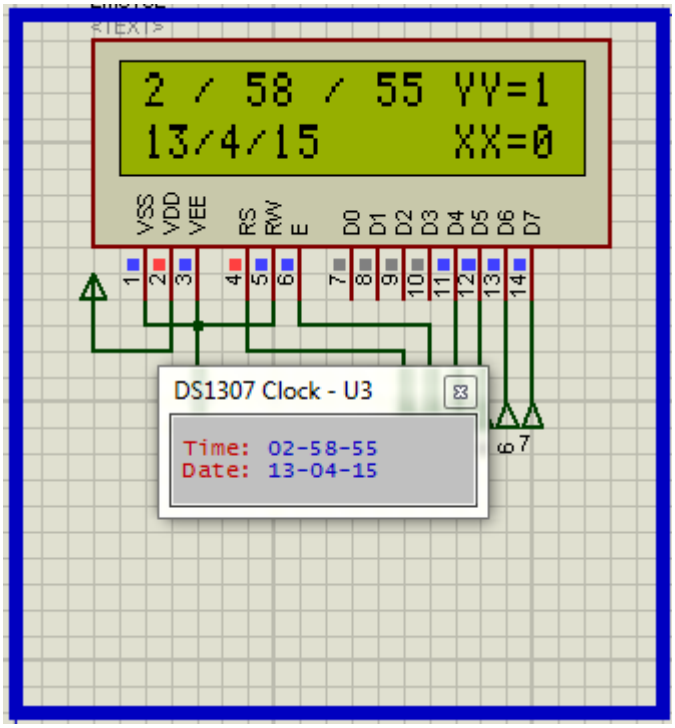

figure 5.3 the written time in the RTC (ds1307) IC.

5.4 The adjust of the patient time and first alarm for patient

The next figure 5.4 and figure 5.5 show the results obtained for adjust the time patient witch its displayed the houre_1 and minit_1 to the patient medicine to be compared with time in the RTC.

- The yy is equal to 2.
- The value xx change from zero to 2 to adjust the time patient alarm one.
- Hour_1 is equal to two and minit_1 is equal to 58
- The hour is equal to two and minit_1 is equal to 58 in the RTC IC.
- The alarm led of first patient is turn on in two side doctor and patient to inform the patient and doctor.
- Alarm sound well be run in two side doctor and patient.
- The LCD in the side of patient display alarm one to inform the patient, figure 5.4 show the adjust value in the lcd and figure 4.5 show the other detelse.

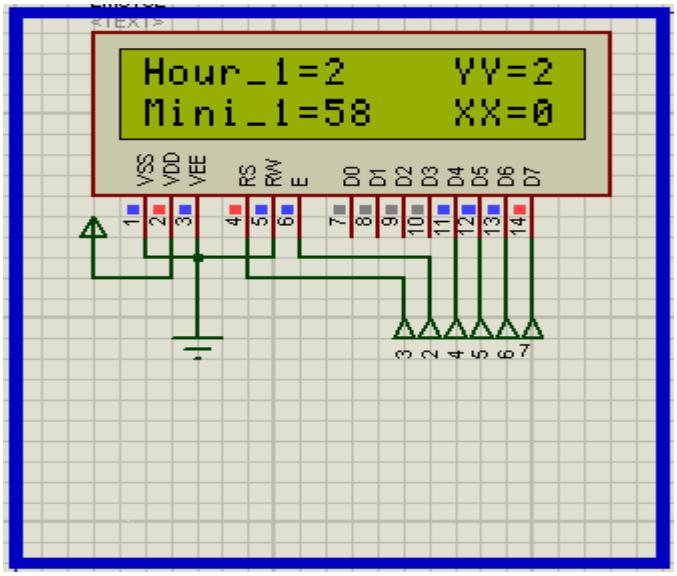

Figure 5.4 the adjust time for patient alarm one.

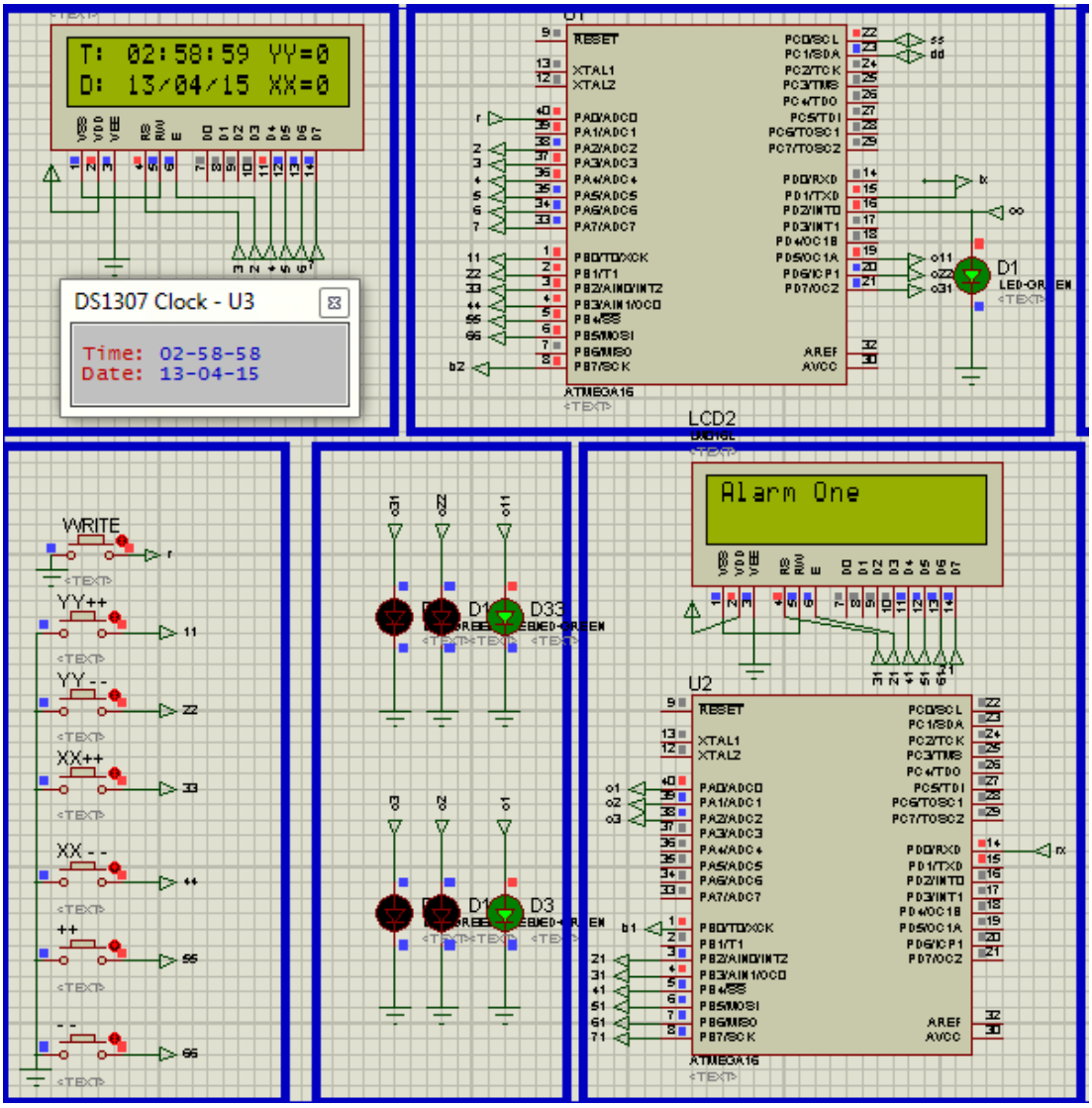

Figure 5.5 the adjust of the patient time and alarm for first patient.

5.5 The adjust of the patient time and second alarm for patient

The next figure 5.6 and figure 5.7 show the results obtained for adjust the time patient witch its displayed the houre_2 and minit_2 to the patient medicine to be compared with time in the RTC.

- The yy is equal to 3.
- The value xx change from zero to 2 to adjust the time patient two.
- Hour_2 is equal to two and minit_2 is equal to 59
- The hour is equal to two and minute is equal to 59 in the RTC IC.
- The alarm led of second patient is turn on in two side doctor and patient to inform the patient and doctor.
- Alarm sound well be run in two side doctor and patient.
- The LCD in the side of patient display alarm two to inform the patient, figure 5.6 show the adjust time for alarm two in the lcd and figure 5.7 show the other detelse.

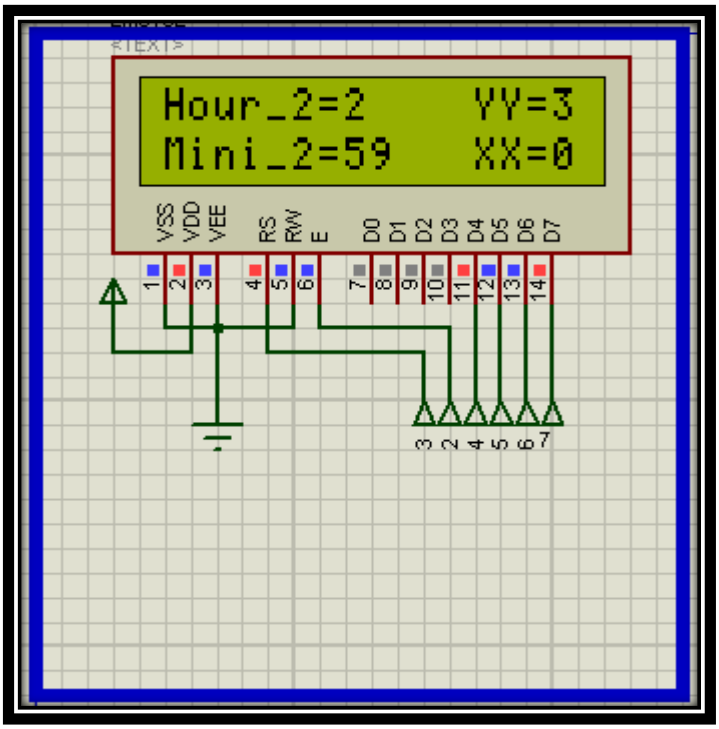

Figure 5.6 the adjust time for patient alarm two.

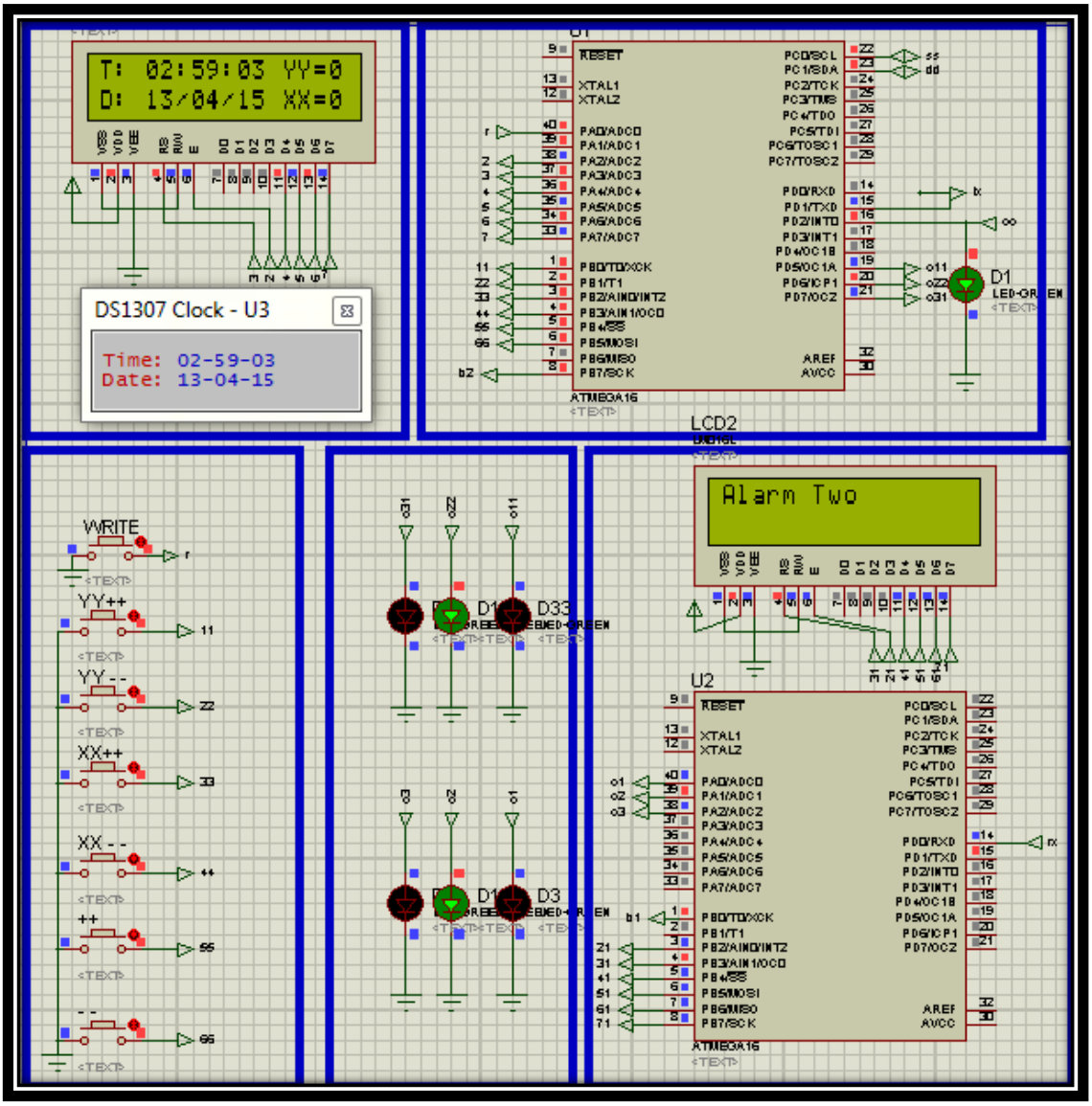

Figure 5.7 the adjust of the patient time and alarm for second patient

5.6 The adjust of the patient time and third alarm of patient

The next figure 5.8 and figure 5.9 shows the results obtained for adjust the time patient witch it's displayed the houre_3 and minit_3 to the patient medicine to be compared with time in the RTC.

- The yy is equal to 4.
- The value xx change from zero to 2 to adjust the time patient three.
- Hour_3 is equal to three and minit_3 is equal to zero.

- The hour is equal to three and minute is equal to zero in the RTC IC.
- The alarm led of third patient is turn on in two side doctor and patient to inform the patient and doctor.
- Alarm sound well be run in two side doctor and patient.
- The LCD in the side of patient display alarm three to inform the patient, figure 5.8 show the adjust time for alarm three in the lcd and figure 5.9 show the other detelse.

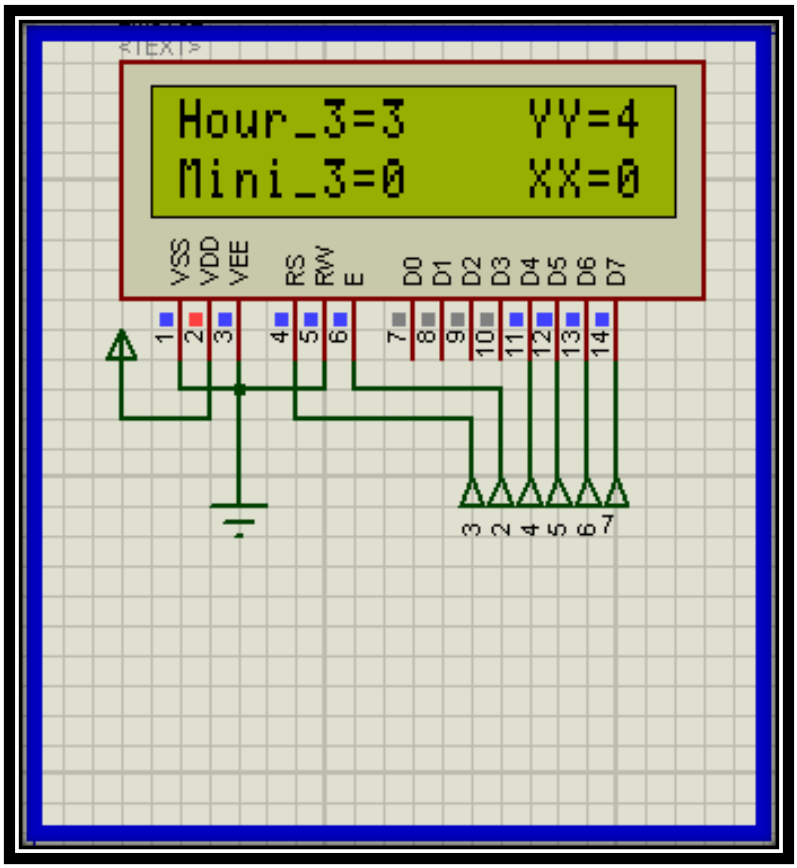

Figure 5.8 the adjust time for patient alarm three.

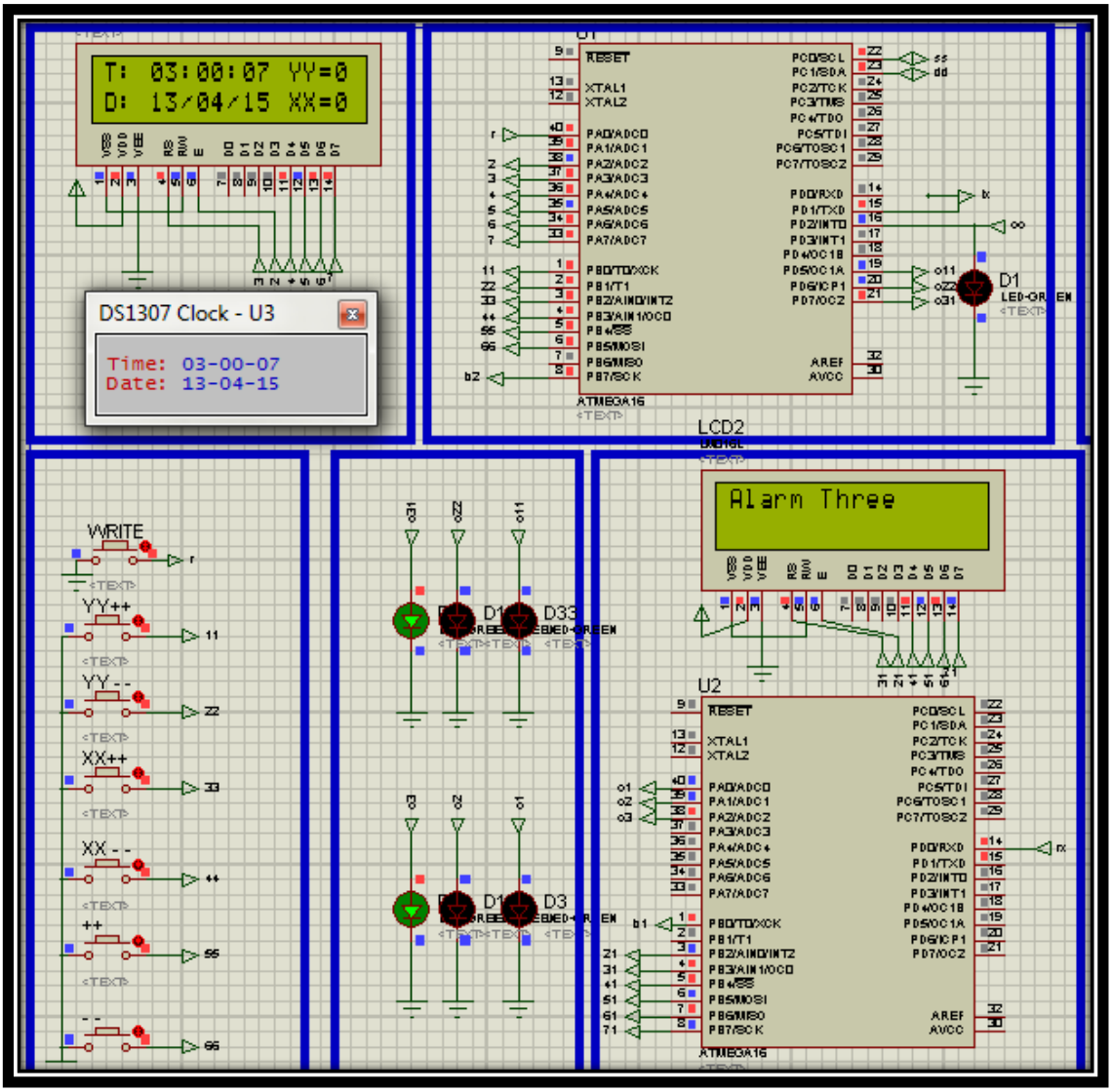

Figure 5.9 the adjust of the patient time and alarm for third patient.

5.7 The time RTC and patient for all alarms

The next figure 5.10 show the results obtained for time RTC and alarm for all patient medicines to be compared together and no alarm is turn on because no one patient have the same time with the RTC clock .

- ❖ No patient has same time with the RTC clock.
- ❖ No alarm turns on.

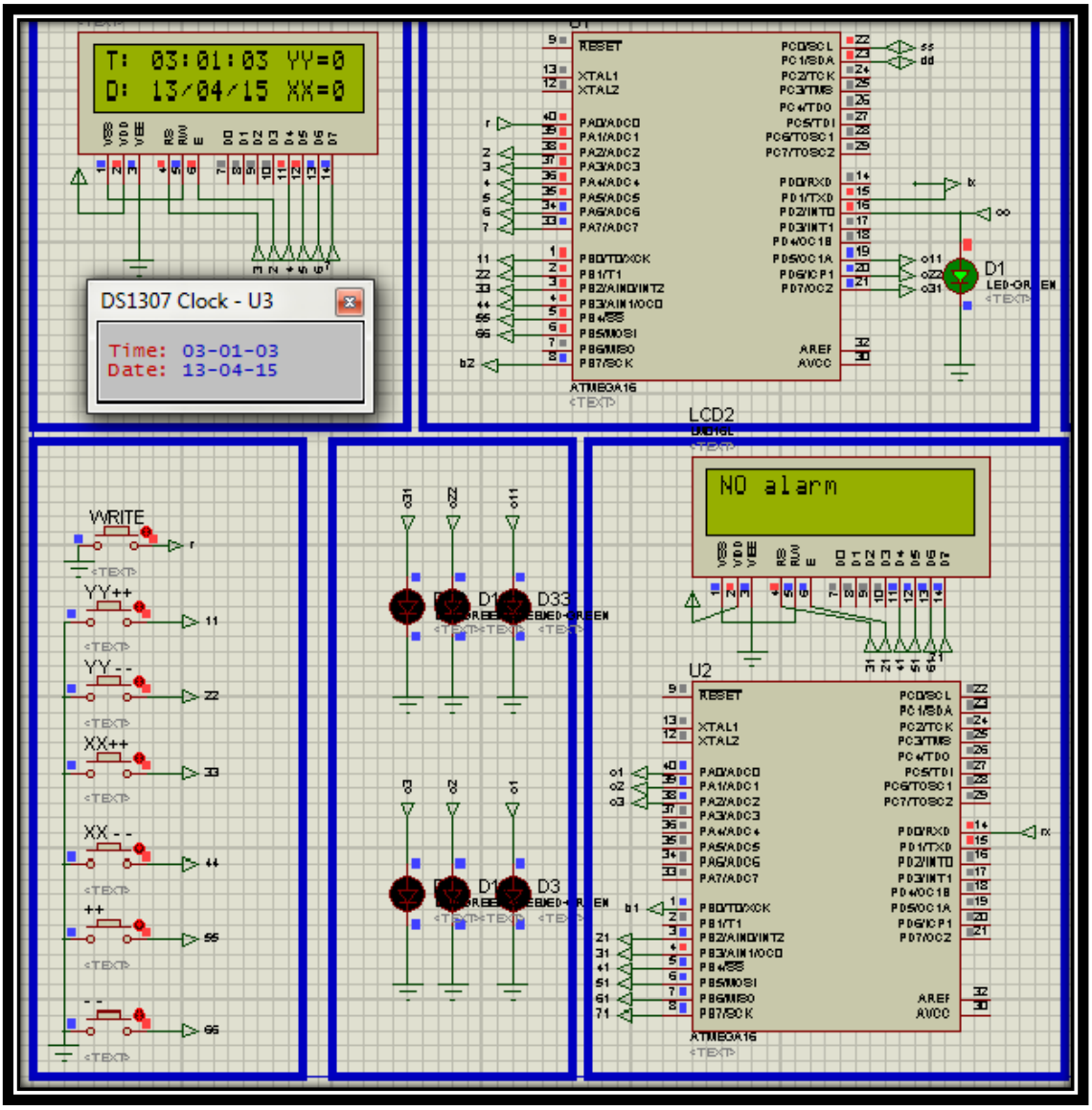

Figure 5.10 the time RTC and alarm for all patients.

CHAPTER

SIX

CONCLUSION AND RECOMMENDATION

Conclusion and Recommendation

6.1 Conclusions

This project describe the human health situation by monitoring time using real time clock (RTC) to give the time, The system allows the user to enter the prescribed timings, at which the patient has to take the medication. This is done using a few switches this data will be stored in the microcontroller. The microcontroller continuously reads the time from the RTC .When the timings read from the RTC equals the timings stored in the Microcontroller, the system alerts the buzzer in two side doctor and patient by wireless communication, and the box of alarm is turn in side of the patient. and display the result on LCD. The aim of this device is to design system should be able to monitor the medicine patient by simple way and low cost power size to provide the portable by using microcontroller and RTC. This system has good ability to give accurate and reliable values and alarm for time medicine. The device is portable and can be get with the patient every were.

6.2 Recommendation

1. Increasing for types of used boxes and the number of medicine.
2. For more patient health mentoring such as database.
3. Increasing system accuracy by using more advance microcontroller.
4. Enhance the system for wireless by using more advance technique with more features (GSM).
5. Designed this system as application in mobile for easy use and more features.

REFERENCES

[1]Microcontroller Based Patent Medicine Reminder System , Amritha B.shyam , Gujjula Kavya Reddy", Monica Maddukuri 2010.

[2]Senior Care and Safety Check, Garland County Sheriff's Departmen..

[3]Celler. Branko. Nigel Lovell, and Daniel Chan. "The Potential Impact of Home Telecare of Clinical Practice". The Medical Journal of Australia. 1999- 518,521.

[4]Datasheet Archive." Atmega16 <,www.atmel.com " 1-11,2014>.

[5]Datasheet.16,2-LCD module <www.alldatasheet.com 15-11,2014>.

[6]Datasheet Archive ,RTC ds1307 <www.alldatasheet.com 12-11,2014>.

[7]Datasheet Archive ,Switch's < www.alldatasheet.com 5-11,2014>.

[8]Datasheet Archive ,led's < www.alldatasheet.com 5-11,2014>.

Appendixes:

The code A:

```
$regfile = "m16def.dat"
$crystal = 1000000
$baud = 9600
$lib "i2c_twi.lbx"
Config Scl = Portc.0
Config Sda = Portc.1
Config Twi = 10000
I2cinit
'---------------------------- [clock configration]
Config Clock = User
Config Date = Dmy , Separator = /
'-----------------------------------[int0 interrupt]
Config Int0 = Rising : On Int0 Int0_isr : Config Pind.2 = Input : Portd.2
= 1
'------------------------[lcd configration]
Config Lcdpin = Pin , Db4 = Porta.4 , Db5 = Porta.5 , Db6 = Porta.6 ,
Db7 = Porta.7 , E = Porta.2 , Rs = Porta.3
Config Lcd = 16 * 2
Initlcd
Config Pina.0 = Input : Set_key Alias Pina.0 : Porta.0 = 1
Config Pina.1 = Input : W_key Alias Pina.1 : Porta.1 = 1
Config Portb = Input
Sw_1 Alias Pinb.0 : Portb.0 = 1
Sw_2 Alias Pinb.1 : Portb.1 = 1
Sw_3 Alias Pinb.2 : Portb.2 = 1
Sw_4 Alias Pinb.3 : Portb.3 = 1
```

```
Sw_5 Alias Pinb.4 : Portb.4 = 1
Sw_6 Alias Pinb.5 : Portb.5 = 1
Config Portd.5 = Output
Config Portd.6 = Output
Config Portd.7 = Output
Config Portb.7 = Output
Dim Dsp_flag As Bit , Weekday As Byte , F As Byte , Yy As Byte , Xx
As Byte
Const Ds1307w = &B11010000 : Const Ds1307r = &B11010001
Enable Int0 : Enable Interrupts : Cls : Cursor Off
Dim E As Word

Dim Mini_1 As Byte : Dim Hour_1 As Byte
Dim Mini_2 As Byte : Dim Hour_2 As Byte
Dim Mini_3 As Byte : Dim Hour_3 As Byte
Dim Mini_set As Byte : Dim Hour_set As Byte : Dim Sec_set As Byte
Dim Year_set As Byte : Dim Month_set As Byte : Dim Day_set As Byte
Hour_1 = 2
Hour_2 = 2
Hour_3 = 3
Mini_1 = 58
Mini_2 = 59
Mini_3 = 00
Yy = 0
Xx = 0
Year_set = 15
Month_set = 4
Day_set = 13
Hour_set = 2
```

```
 Mini_set = 58
Sec_set = 55


 Gosub Set_sqw
Do
Gosub Switchs
Gosub Yy_state
Gosub Chech
Loop
End

Switchs:
Debounce Sw_1 , 0 , Yy_increment , Sub
Debounce Sw_2 , 0 , Yy_decriment , Sub
Debounce Sw_3 , 0 , Xx_increment , Sub
Debounce Sw_4 , 0 , Xx_decriment , Sub
Debounce Set_key , 0 , Set_val , Sub
Locate 1 , 13
Lcd "YY="
Lcd Yy
Locate 2 , 13
Lcd "XX="
Lcd Xx

Return

Yy_increment:
Xx = 0
```

```
If Yy <> 4 Then
Cls
Yy = Yy + 1
Else
Yy = 0
End If
Return

Yy_decriment:
Xx = 0
If Yy <> 0 Then
Cls
Yy = Yy - 1
Else
Yy = 4
End If
Return

Xx_increment:
If Xx <> 5 Then
Cls
Xx = Xx + 1
Else
Xx = 0
End If
Return

Xx_decriment:
If Xx <> 0 Then
```

```
Cls
Xx = Xx - 1
Else
Xx = 5
End If
Return

Yy_state:
If Yy = 0 Then

If Dsp_flag = 1 Then
Reset Dsp_flag
Locate 1 , 1 : Lcd "T: " ; Time$
Locate 2 , 1 : Lcd "D: " ; Date$
End If

Elseif Yy = 1 Then
Locate 1 , 1
Lcd Hour_set : Lcd " / " : Lcd Mini_set : Lcd " / " : Lcd Sec_set
Locate 2 , 1
Lcd Day_set : Lcd "/" : Lcd Month_set : Lcd "/" : Lcd Year_set
If Xx = 0 Then
Debounce Sw_5 , 0 , Year_set_incr , Sub
Debounce Sw_6 , 0 , Year_set_decr , Sub
Elseif Xx = 1 Then
Debounce Sw_5 , 0 , Month_set_incr , Sub
Debounce Sw_6 , 0 , Month_set_decr , Sub
Elseif Xx = 2 Then
Debounce Sw_5 , 0 , Day_set_incr , Sub
```

```
Debounce Sw_6 , 0 , Day_set_decr , Sub
Elseif Xx = 3 Then
Debounce Sw_5 , 0 , Hour_set_incr , Sub
Debounce Sw_6 , 0 , Hour_set_decr , Sub
Elseif Xx = 4 Then
Debounce Sw_5 , 0 , Mini_set_incr , Sub
Debounce Sw_6 , 0 , Mini_set_decr , Sub
Elseif Xx = 5 Then
Debounce Sw_5 , 0 , Sec_set _incr , Sub
Debounce Sw_6 , 0 , Sec_set _decr , Sub
End If

Elseif Yy = 2 Then
Locate 1 , 1
Lcd "Hour_1=" : Lcd Hour_1                    ': Lcd "   "
Locate 2 , 1
Lcd "Mini_1=" : Lcd Mini_1                    ': Lcd "   "
If Xx = 0 Then
Debounce Sw_5 , 0 , Hour_1_set_incr , Sub
Debounce Sw_6 , 0 , Hour_1_set_decr , Sub
Elseif Xx = 1 Then
Debounce Sw_5 , 0 , Mini_1_set_incr , Sub
Debounce Sw_6 , 0 , Mini_1_set_decr , Sub
End If

Elseif Yy = 3 Then
Locate 1 , 1
Lcd "Hour_2=" : Lcd Hour_2                    ': Lcd "     "
Locate 2 , 1
```

```
Lcd "Mini_2=" : Lcd Mini_2                    ': Lcd "     "
If Xx = 0 Then
Debounce Sw_5 , 0 , Hour_2_set_incr , Sub
Debounce Sw_6 , 0 , Hour_2_set_decr , Sub
Elseif Xx = 1 Then
Debounce Sw_5 , 0 , Mini_2_set_incr , Sub
Debounce Sw_6 , 0 , Mini_2_set_decr , Sub
End If

Elseif Yy = 4 Then
Locate 1 , 1
Lcd "Hour_3=" : Lcd Hour_3                     ': Lcd "     "
Locate 2 , 1
Lcd "Mini_3=" : Lcd Mini_3                    ' : Lcd "     "
If Xx = 0 Then
Debounce Sw_5 , 0 , Hour_3_set_incr , Sub
Debounce Sw_6 , 0 , Hour_3_set_decr , Sub
Elseif Xx = 1 Then
Debounce Sw_5 , 0 , Mini_3_set_incr , Sub
Debounce Sw_6 , 0 , Mini_3_set_decr , Sub
End If


End If
Return

Year_set_incr:
Year_set = Year_set + 1
Return
```

```
Year_set_decr:
Year_set = Year_set - 1
Return
Month_set_incr:
Month_set = Month_set + 1
Return
Month_set_decr:
Month_set = Month_set - 1
Return
Day_set_incr:
Day_set = Day_set + 1
Return
Day_set_decr:
Day_set = Day_set - 1
Return
Hour_set_incr:
Hour_set = Hour_set + 1
Return
Hour_set_decr:
Hour_set = Hour_set - 1
Return
Mini_set_incr:
Mini_set = Mini_set + 1
Return
Mini_set_decr:
Mini_set = Mini_set - 1
Return
Sec_set_incr:
Sec_set = Sec_set + 1
```

```
Return
Sec_set_decr:
Sec_set = Sec_set - 1
Return
Hour_1_set_incr:
Hour_1 = Hour_1 + 1
Return
Hour_1_set_decr:
Hour_1 = Hour_1 - 1
Return
Mini_1_set_incr:
Mini_1 = Mini_1 + 1
Return
Mini_1_set_decr:
Mini_1 = Mini_1 - 1
Return
Hour_2_set_incr:
Hour_2 = Hour_2 + 1
Return
Hour_2_set_decr:
Hour_2 = Hour_2 - 1
Return
Mini_2_set_incr:
Mini_2 = Mini_2 + 1
Return
Mini_2_set_decr:
Mini_2 = Mini_2 - 1
Return
Hour_3_set_incr:
```

```
Hour_3 = Hour_3 + 1
Return
Hour_3_set_decr:
Hour_3 = Hour_3 - 1
Return
Mini_3_set_incr:
Mini_3 = Mini_3 + 1
Return
Mini_3_set_decr:
Mini_3 = Mini_3 - 1
Return

Int0_isr:
Set Dsp_flag
Gosub Getdatetime
Return


Set_sqw:
I2cstart
I2cwbyte Ds1307w
I2cwbyte &H07
I2cwbyte &B00010000
I2cstop
Return


Getdatetime:
I2cstart
```

```
I2cwbyte Ds1307w
I2cwbyte 0

I2cstart
I2cwbyte Ds1307r
I2crbyte _sec , Ack
I2crbyte _min , Ack
I2crbyte _hour , Ack
I2crbyte Weekday , Ack
I2crbyte _day , Ack
I2crbyte _month , Ack
I2crbyte _year , Nack
I2cstop
Gosub Bcd_dec_date : Gosub Bcd_dec_time
Return

Bcd_dec_date:
_day = Makedec(_day) : _month = Makedec(_month) : _year = Makedec(_year)
Return
Bcd_dec_time:
_sec = Makedec(_sec) : _min = Makedec(_min) : _hour = Makedec(_hour)
Return

Dec_bcd_date:
_day = Makebcd(_day) : _month = Makebcd(_month) : _year = Makebcd(_year)
Return
```

```
Dec_bcd_time:
_sec = Makebcd(_sec) : _min = Makebcd(_min) : _hour = Makebcd(_hour)
Return


Setdate:
I2cstart
I2cwbyte Ds1307w
I2cwbyte 4
I2cwbyte _day
I2cwbyte _month
I2cwbyte _year
I2cstop

Return
Settime:
I2cstart
I2cwbyte Ds1307w
I2cwbyte 0
I2cwbyte _sec
I2cwbyte _min
I2cwbyte _hour
I2cstop
Return

Set_val:
If Yy = 1 Then
_year = Year_set
```

```
_month = Month_set
_day = Day_set
_hour = Hour_set
_min = Mini_set
_sec = Sec_set
Gosub Dec_bcd_date : Gosub Setdate
Gosub Dec_bcd_time : Gosub Settime
Gosub Set_sqw

Elseif Yy = 3 Then
Elseif Yy = 4 Then
Elseif Yy = 5 Then
End If
Return


Chech:
If Mini_1 = _min And Hour_1 = _hour Then
Printbin 1
Portd.5 = 1
Portb.7 = 1
Else
Printbin 2
Portd.5 = 0
End If
If Mini_2 = _min And Hour_2 = _hour Then
Printbin 3
Portd.6 = 1
Portb.7 = 1
```

```
Else
Portd.6 = 0
Printbin 4
End If
If Mini_3 = _min And Hour_3 = _hour Then
Printbin 5
Portd.7 = 1
Portb.7 = 1
Else
Portd.7 = 0
Printbin 6
End If
If Pind.5 = 0 And Pind.6 = 0 And Pind.7 = 0 Then
Portb.7 = 0
End If
Return
```

The code B:

```
$regfile = "m16def.dat"
$crystal = 1000000
$baud = 9600
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 ,
Db7 = Portb.7 , E = Portb.2 , Rs = Portb.3
Config Lcd = 16 * 2
Initlcd
Cursor Off
Config Porta.0 = Output
Config Porta.1 = Output
Config Porta.2 = Output
Config Portb.0 = Output
```

```
Dim X As Byte
Do
Inputbin X
If X = 1 Then
Porta.0 = 1
Portb.0 = 1
Locate 1 , 1 : Lcd "Alarm One   "
Elseif X = 2 Then
Porta.0 = 0
Elseif X = 3 Then
Porta.1 = 1
Portb.0 = 1
Locate 1 , 1 : Lcd "Alarm Two   "
Elseif X = 4 Then
Porta.1 = 0
Elseif X = 5 Then
Porta.2 = 1
Portb.0 = 1
Locate 1 , 1 : Lcd "Alarm Three"
Elseif X = 6 Then
Porta.2 = 0
End If


If Pina.1 = 0 And Pina.0 = 0 And Pina.2 = 0 Then
Locate 1 , 1 : Lcd "NO alarm        " : Portb.0 = 0
End If

loop _month = Month_set
```

```
_day = Day_set
_hour = Hour_set
_min = Mini_set
_sec = Sec_set
Gosub Dec_bcd_date : Gosub Setdate
Gosub Dec_bcd_time : Gosub Settime
Gosub Set_sqw

Elseif Yy = 3 Then
Elseif Yy = 4 Then
Elseif Yy = 5 Then
End If
Return


Chech:
If Mini_1 = _min And Hour_1 = _hour Then
Portd.7 = 1
Else
Portd.7 = 0
End If
If Mini_2 = _min And Hour_2 = _hour Then
Portd.6 = 1
Else
Portd.6 = 0
End If
If Mini_3 = _min And Hour_3 = _hour Then
Portd.5 = 1
Else
```

```
Portd.5 = 0
End If
Return
```