JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

COMPUTER STREAM

TCBTP PHASE IV

Title: - Smart Medicine Reminder and Dispenser

By:

Group Members:	**Id.No.**
1. Hailemicael Lulseged	RU1608/10
2. Hunduma Atomsa	RU1685/10
3. Eden Degefu	RU1373/10

Advisor: Mr. Eyob Sisay

_____

Sign

Submission date: May 24, 2021

Jimma, Ethiopia.

ACKNOWLEDGMENT

First of all, we would like to thank Jimma University to design this community based training program education. Next, we heart-fully gratitude our thanks to electrical and computer engineering department for giving us this chance to do this project. At the last we would like to thanks our advisor Mr. Eyob Sisay, for his advice and help us constructive comments and ideas to do this project with great motivation and confidence.

ABSTRACT

Many health care patients to enroll in assisted living facilities because they are not able to manage their complex medication regimens without the active assistance of the care given. Management of medication is very important for acute illness and long-term conditions. Many aged people need to take several drugs due to physical and mental function decline before this seriously affected health. Forgetting to take prescribed medicine is a problem. In such problem, managing medication is required. There are several android apps and other software products designed. Nonetheless, many of the elderly people can't make use of smartphones because in countries like Ethiopia some of them are not familiar with these devices or cannot afford the price. Hence, a system proposed in this project is a simple medicine reminder and disperser that can help with medication reminding for the elderly people as well as for the people who are suffering from Alzheimer's disease. This simple medicine reminder and dispenser is simple, common and can be used by patients at home, doctors at hospitals, and any other places to take medicine on time. We have designed and simulated this system using Arduino Uno microcontroller. The time slot can be selected using a keypad. Based on the real-time clock interfaced to the microcontroller, the programmed time for the medicine is displayed on the LCD and with the buzzer alerts the patient to take the appropriate medicine indicated by LED and also the servo motor dispenses the medicine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

WHO	**W**orld **H**ealth **O**rganization
LED	**L**ight **E**mitting **D**iode
LCD	**L**iquid **C**rystal **D**isplay
RTC	**R**eal **T**ime **C**lock
MRS	**M**edicine **R**eminder **S**ystem
SMRD	**S**mart **M**edicine **R**eminder and **D**ispenser
EPROM	**E**rasable **P**rogrammable **Read-O**nly **Memory**
IR	**I**nfra-**R**ed
IoT	**I**nternet **o**f **T**hings

CHAPTER 1: INTRODUCTION

1.1 Background of the Project

In today's world, everything is dependent on technology. The radical growth of technology makes the way of life, reduces problems, and substitutes human being's labor with machines and computer works. Following this, many different electronic devices are developed for different purposes such as health care, transportation, marketing, education, etc. From all the above the most and very important area has been got consideration is healthcare-related one. Today almost all over the world, solving the problems related to human life is being the most issue. Significantly one of these will be done by improving the way of health care of patients. Some of them are the interrelationship between patients and medicine on the sack of reminding and dispensing of medicine. Thus, to improve problems related to the way of taking medicine, developing a system is needed.

Nowadays in our world, the greater issue next to corona is the high percentage of the world is dying due to diseases which are called" silent killer" such as hypertension, diabetics and so on. And those patients who suffer from these diseases need to take medicine regularly. Not only these diseases and also TB is another disease which patients suffer from it need to take medicine many times a day. In the current situation, it is listed under the related case to the corona.

Some diseases are temporary while many are permanent life-threatening diseases. Life-threatening diseases get mixes with the human body in such a way that they can't leave the body ever and they increase in rapid time. The lifespan of humans became less because of such diseases and to overcome or to live a better life some people need to take medicines regularly.

This project attempts to solve this important issue by developing a smart medication monitoring and notification system that can enable patients to follow prescribed medication schedules with minimal effort. This should enhance the lives of patients, and enable their independent living.

We need to be on the advice of a Doctor who tells us to take desired pills in the desired way so that patients face problems like forgetting pills to take at right time and also when the Doctor changes the prescription of medicine patients have to remember the new schedule of medicine.

The problem of forgetting to take pills at right time, taking wrong medicines, and accidentally taking expired medicine causes health issues for the patient and leads to suffering from unhealthy life. This project is to make a Smart medicine box that uses a Real-time clock and drug dispenser. The patient must take pills from the box at the right time the system continues to make a large sound.

1.2 Statements of the Problem

This project is done depending on the problems in the healthcare system. The problem is that the inability of patients and healthcare providers to perfectly match medicine and schedule with a patient's requirements? Because of this, many patients are not taking their medicines as directed, leading to unnecessary disease progression, complications, lower quality of life, and mortality. Many patients would prefer to live independently. This would allow them to have more freedom, and relieve stress on the healthcare system. However, many elderly patients are not able to do this, simply because they are not capable of following their medication protocol. This is because it may be too complex, or the patient may be forgetful. Currently, the patient's only option is admission to an assisted living facility, such as a hospital or nursing home, or requiring the attentive care of a volunteer caregiver.

1.3 Objectives

1.3.1 General Objective

The general objective of this project is to design, develop and simulate a system that provides medicine for the patient to take appropriately.

1.3.2 Specific Objectives

- To design and simulate the overall circuit that would control the remainder using Arduino Uno with the interface of a real-time clock.
- To design and simulate the system that makes alert at the set time to take the medicine.
- To design and simulate the system that dispenses a medicine.

1.4 Significance of the Project

The significance of this project is that patients and any sick family member will have it at their home and can easily remind to take their medicines. This means the SMRD would be within their reach and can be reminded and dispensed automatically. The alarm reminds the patient to take and the servo motor dispenses the medicine correctly. However, people who are suffered from deaf and blind problems can also use this system to take the medicine appropriately. This can will be done in two ways. When the alarm gives the sound, the blind patient can remind to take their medicine and a deaf patient can be reminded via the light of LED.

On the other hand nurses, doctors, and others who attend to the sick and disabled people can use this system. In place of giving attention to the patients, they can focus more on other things since the giving of medicines will be done by the automated system appropriately and reserved medicine will be dispensed at the right time.

Since this project is done using Arduino Uno it is faster when compared with any device implemented using PIC. Generally, it has a good performance.

1.5 Scope and Limitation of the Project

The scope of this project (smart medicine reminder and dispenser) in its most basic form on that it alerts the patient to take the medicine by dispensing it at the appropriate time set and also it shows the medicine which the patient should take. The device can be used at different places such as home, hospital, and other clinical health care centers. The device (i.e. SMRD) simple and not complex to use. Any person can be familiar with and can use it.

The limitation of this project is that it doesn't give any sign whether the patient takes the medicine or not. And also, the system can be adjusted to give the medicine dosage using an IR sensor; but due to the lack of time, we can’t do it in such a manner. Additionally, the device doesn’t detect whether the patient picks the medicine or another person picks it.

1.6 Outlines of the Project

This project is outlined with an introduction containing the background of the project, a statement of the problem, objectives, significance of the project, scope, and limitation of the project in chapter one, whereas chapter two deals with the literature review. Subsequently, chapter three contains methodology, block diagram, and tools. Then chapter four contains the design, modeling of the overall circuit, and working principle. Following this, chapter five deals with results and discussion. Finally, chapter six allocates the conclusion and recommendations.

CHAPTER 2: LITERATURE REVIEW

As stated by World Health Organization (WHO) poor adherence can lead to serious health risks. For instance, a recent study found out that the risk of hospitalized patients, having diabetes congestive heart failure, Mellitus, hypertension, or hypercholesterolemia who was no adherent to prescribed remedies was more in comparison with the general population. Non-adherence rates can vary widely, even in the rigid controlled and monitored environment of a clinic test [1].

2.1 Existing Systems

There are several patents published for various types of medication reminding and dispensing devices. Some products proposed by each patient include the following protocols: -

Timed medicine dispenser: this product is a timed-based dispenser that will give only notification to the user about the medication. The dispenser unit pills need to be pre-separated first to get the correct dosage.

Medicine reminder and dispenser: this project, has a slide-able plate that supports for dispensing of medicine which is arranged in the forms of rows and columns.

Med center medication reminder: this device, consists of the idea of color coding and verbal reminders, for helping people which have complicated medication of routines.

In general, there are some existing projects done like ours. Some of these are:

The autonomous pill dispenser Mechanizing the delivery of tablet medication: This project is an android app that sends a signal to the device via Bluetooth. The patient needs to flip the unit so one pill gets trapped in the tip of the cone and using a vibrator it gets dispensed [2]**.**

Pro: for the developed country it is most reliable due to its android app and monetarization

Con: on this system elderly people, find this difficult to flip the device and less likely to use the android phone mostly developing country like (Ethiopia). Even also, technology is not grown when compared to other developed countries. Another great issue is the economy to buy that smartphone and lack of knowledge how to use these smartphones.

Automatic Pill Dispenser: makes use of the concept of rotating components in circular step-wise motion using a stepper motor which is used to store the pills. Notification on a smartphone is provided. And also here in this system, there may the limitation that smartphones are less likely used by elderly people [3].

Pill dispenser with alarm via smartphone notification: These systems used the available technology to send notifications on the smartphone using the instep push application after receiving the notification user needs to press the dispenser button which is located at the pill dispenser unit. This system also has its limitation that the same as with automatic pill dispensers; the smartphone is rarely used by elderly people or patients. And also this system is difficult to use by blind people. As it need press the button on the dispensing unit, it is not applicable for blind patients or peoples [4].

Timed medicine dispenser: it gives notification about medication. It has an alarm built-in alarm system. The limitation of this system is that in the dispenser unit the pills need to be pre-separated first to get the correct dosage. The other limitation of this system is that deaf patients or people cannot use it.

Almost in all the above existing projects, in the context of our country, 80 % of the country's economy is dependent on agriculture. So on the follow up of technology, most of people life is on the ways of agriculture. Due to this, the assessment of the smartphone is limited to a small number of people. Even our system further app manual, it looks like luxury regarding simply give our system [5].

2.2 Notice Notification System

The effectiveness of therapy or treatment directly depends upon a patient's ability and willingness to follow a prescribed regimen. The patient's ability for reading and understanding the instruction for medication is a key factor. Patients who face difficulties in understanding the instruction in a prescription ultimately results in decreased adherence and poor medication management and consumption. The elderly people are suffered from using smartphone and understanding it. Issues of low literacy must be recognized and strategies designed with this limitation in consideration.

We design and include notice notification depends on the standard of our country. The system we have developed is that it has an alarming system on the same perspective dispensing feature that the servo motor which used to dispense the medicine that the patient will take at the set time (alarm) for the user. We also use LED which has the best property to show the patient who has a deaf problem take the medicine with the proper dose and time and also the user which has a blind problem, the buzzer used as language or to indicate to take the medicine for those patients. we tried to solve the limitation of the existing system developed before.

CHAPTER 3: METHODOLOGY AND TOOLS

3.1 Methodology

The interface of Arduino-based medicine reminder was well developed and coordinated for efficient performance. There were several stages that we had gone through to complete this project successfully. First, we studied the literature review of different medical reminders and dispensers that is related to our projects. In this stage, we try to complete the project with good comfortable for users especially in our country, and also avoid some drawbacks from it. Then we choose a compatible method and function for the project. Subsequently, we studied how these components integrated. Then we constructed a block diagram for our medical reminder and dispenser with a time set based on the integration of the components and explained each of the components. Afterward, we designed and made the interface component of the overall system. Then, we design our system depending on the designation and interface component made. Then using simulating software called Proteus software and Arduino IDE to compile the modeled overall system and settled the output we gained. Accordingly, we discussed each result we gained. Flowingly, we concluded the project based on the results and discussion. Finally, we gave recommendation.

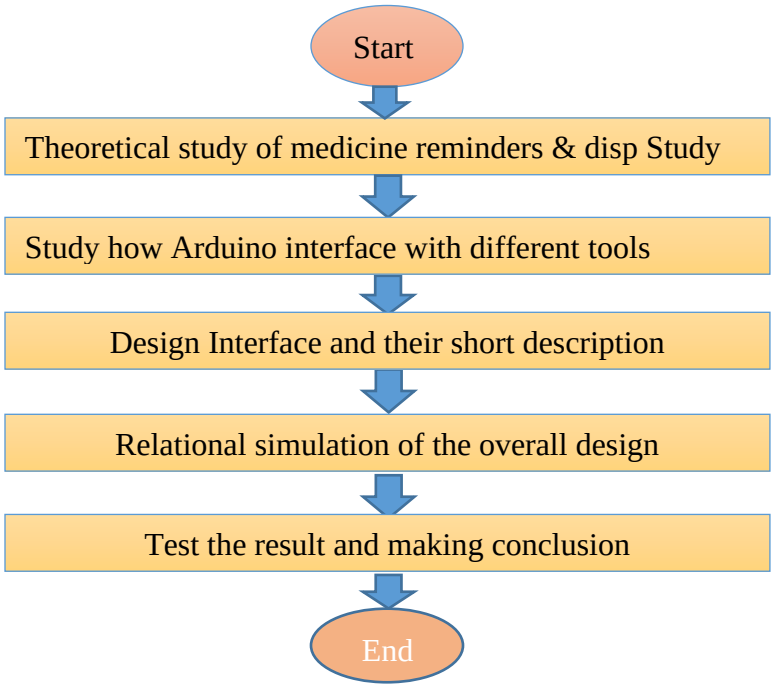

Figure 3-1: Follow chart of the system

3.2 Block Diagram

The first Arduino is activated by the power supply. Then, based on the programs loaded on it, it permits both keypad and LCD. The LCD shows different options. Then the time for medicine adjusts the time base this system's real-time clock runs the time on a 16×2 LCD. then After pressing the keypad, LCD shows different options**.** Following this, we can select how many times we want to take per day and the time we want to set for the start. And real-time is continuously checked our set time. If any match occurs. LCD shows a sign for medication and the buzzer starts beeping continuously and parallels the servo motor dispense the medicine. the component is also properly connected with the Arduino.

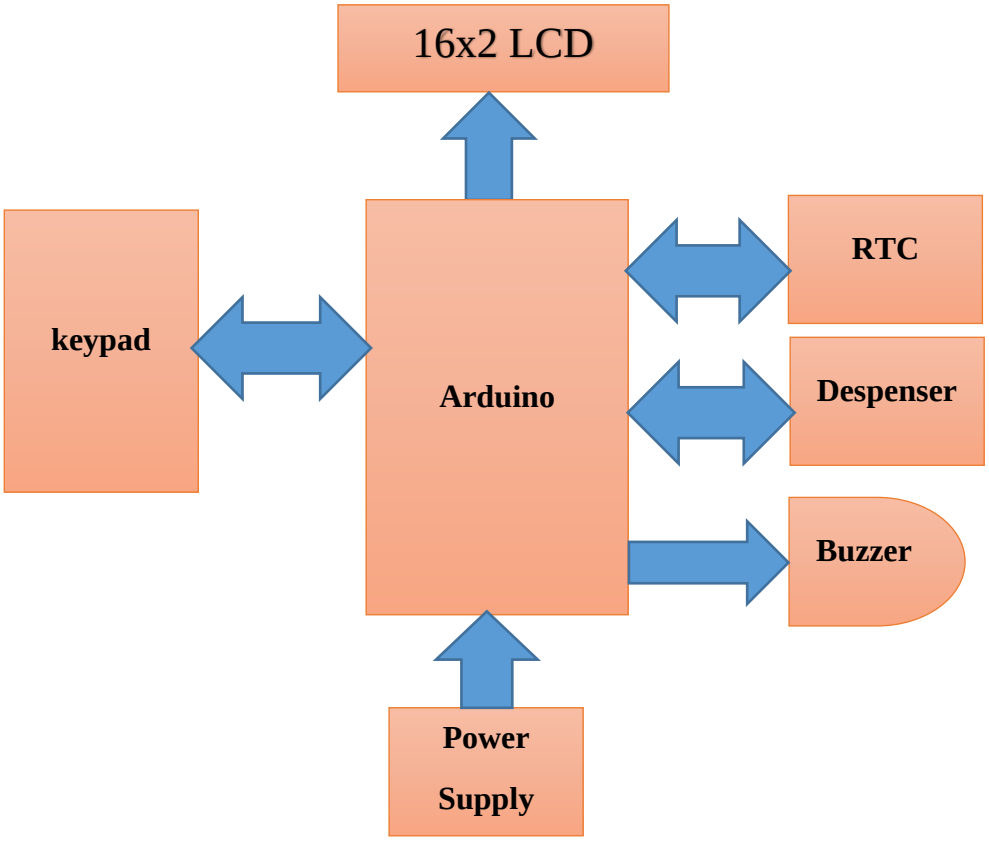

Figure 3-2: Block diagram of the system

3.3 Tools

While we are doing this project, we use different types of tools such as Arduino UNO which is the main tool for our project, LCD, LED, RTC3232, buzzer, Keypad, Servo-motor, resistors, etc.

1. **Arduino –Uno**

Arduino Uno is a microcontroller board based on an 8-bit ATmega328P microcontroller. Along with ATmega328P, it consists of other components such as crystal oscillator, serial communication, voltage regulator, etc. to support the microcontroller. Arduino Uno has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog input pins, a USB connection, A Power barrel jack, an ICSP header, and a reset button[1].

Way of using Arduino Uno

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.

External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM Pins 3, 5, 6, 9, and 11: These pins provide an 8-bit PWM output by using the analogWrite() function.

SPI Pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK): These pins are used for SPI communication.

[1] Arduino Uno, Arduino –Uno - Bing, Accessed: 5/04/2021 6:11 AM

In-built LED Pin 13: This pin is connected with a built-in LED, when pin 13 is HIGH – the LED is on and when pin 13 is LOW, it's off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provides 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using the AREF pin with analog Reference() function.

Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using the Wire library.

Arduino Uno has a couple of other pins as explained below:

AREF: Used to provide a reference voltage for analog inputs with analogReference() function.

Reset Pin: Making this pin LOW, resets the microcontroller.

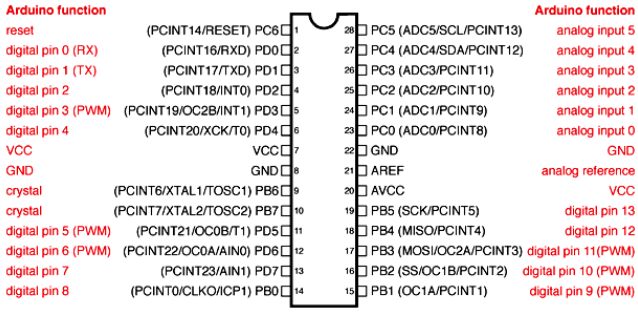

Figure 3-3: Pin configuration of Arduino

2. DS3232 RTC

RTC means **Real-Time Clock**. RTC modules are simply TIME and DATE remembering systems that have battery setup which in the absence of external power keeps the module running. This keeps the TIME and DATE up to date. So we can have accurate TIME and DATE from the RTC module whenever we want.

DS3232 RTC Pin Configuration

DS3232 RTC is a six-terminal device, out of the two pins is not compulsory to use. So we have mainly four pins. These four pins are given out on the other side of the module sharing the same name.

Table 3-1: Pins of DS3232 RTC and their descriptions

Pin Name	Description
VCC	Connected to positive of power source.
GND	Connected to ground.
SDA	Serial Data pin (I2C interface)
ACL	Serial Clock pin (I2C interface)
SQW	Square Wave output pin
32K	32K oscillator output

The only way to communicate with this RTC module is through the I2C interface. The data is sent to the module or received from the module through the I2C interface. So we have to get the information of DATE and TIME through this interface. The default circuit diagram of the DS3232 RTC module interface is shown below[2].

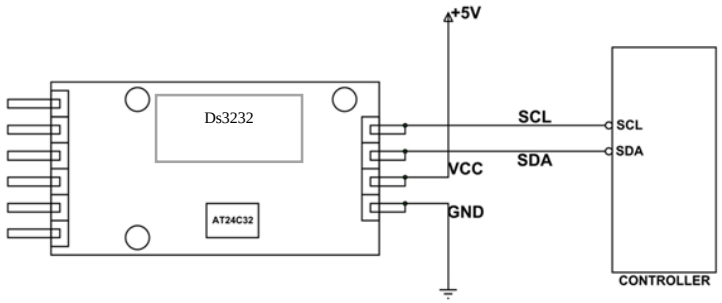

Figure 3-4: Design of DS3232 RTC module

3. 16x2 LCD (Liquid Crystal Display)

It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are

[2] RTC module, wikipedia - Bing, Accessed: 5/03/2021 11:05 PM

mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.[3]

LCD 16×2 PIN Configuration

The 16×2 LCD pinout is shown below.

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin, and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute the Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to the microcontroller unit like 0 to 7.

[3] LCD 16x2, Introduction to 16x2 LCD Module - The Engineering Projects, Accessed: 5/04/2021 3:20 AM

- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

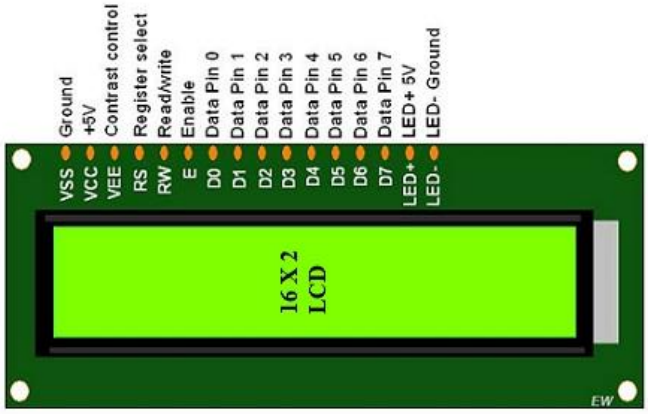

Figure 3-5: 16X2 LCD

4. LED

An LED lamp or LED light bulb is an electric light that produces light using light-emitting diodes (LEDs). In this project, we use it to indicate the medicine which the patient takes.

Figure 3-6: Red LED

5. Buzzer

A **buzzer** or **beeper** is an audio signaling device,[1] which may be mechanical, electromechanical, or piezoelectric (*piezo* for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke. In this project, we use it to remind the patient of the scheduled time at which he/she takes his/her medicine[4].

Figure 3-7: Buzzer

6. Keypad

A keypad is a set of buttons arranged in a block or pad which bears digits, symbols, or alphabetical letters. Pads mostly containing numbers and used with computers are numeric keypads. Keypads are found on devices that require mainly numeric input such as calculators, television remotes, push-button telephones, vending machines, ATMs, Point of Sale devices, combination locks, and digital door locks. Many devices follow the E.161 standard for their arrangement[5].

[4] Buzzer, Buzzer - Wikipedia, Accessed: 5/02/2021 1:42 AM
[5] Keypad, Keypad - Wikipedia, Accessed: 5/01/2021 7:54 AM

Figure 3-8: Keypad

7. Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor that runs through a servo mechanism[6].

Figure 3-9: Servo-motor

[6] Servo-Motor, Servo motor - Wikipedia, Accessed: 5/01/2021 6:02 AM

CHAPTER 4: SYSTEM DESIGN AND SIMULATION

4.1 System Design

The main goal of this project in this chapter was to create a smart medicine remainder and dispenser device. In this project, the goal is to demonstrate a part of the system and investigate how it works with these frameworks. Under this one there is some target that considered when the system is designed. These are:

Design hardware layout

write a code to interface components with Arduino

4.2 The Layout of the Hardware Part

LCD with Arduino interfacing

Table 4-1: Connection between Arduino and LCD

Arduino Pins	**LCD Pins**
2	D7 of 16x2 LCD Display
3	D6 of 16x2 LCD Display
4	D5 of 16x2 LCD Display
5	D4 of 16x2 LCD Display
11	E of 16x2 LCD Display
12	RS of 16x2 LCD Display

The other pins of LCD VSS and RW connected to the ground and VDD connected

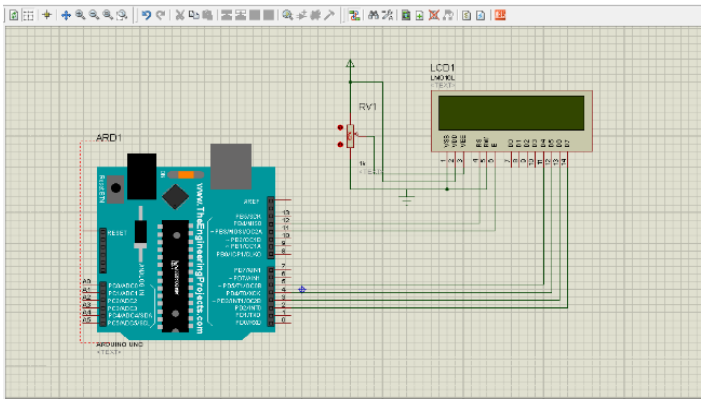

Figure 4-1: Interface LCD

DS3232 with Arduino interfacing

Table 4-2: Connection of LCD pins with Arduino

Arduino Pins	**LCD Pins**
A4	SDA of DS3232
A5	SCL of DS3232
3.3V	Vcc of DS3232
Ground	Ground

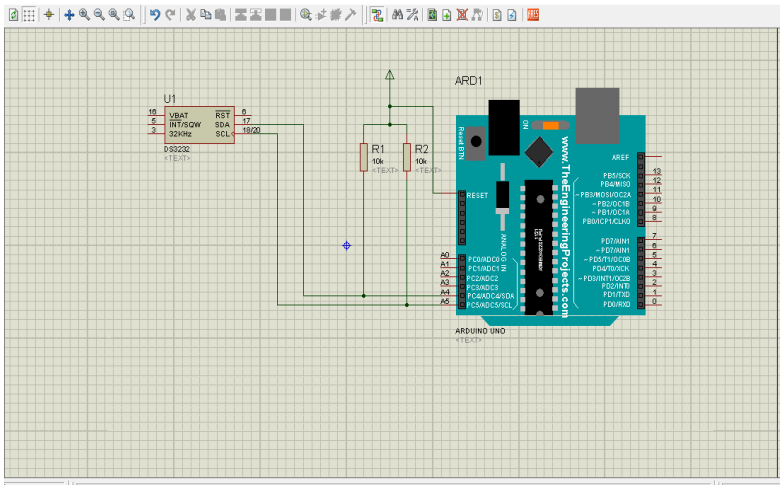

Figure 4-2: Interface of DS3232

Keypad interface with Arduino

Table 4-3: Connection of keypad pins with Arduino

Arduino	Keypad
Pin 7	first row(A)
Pin 8	second row(B)
Pin 9	third row(C)
Pin 10	fourth row(D)
Pin 6	first colomun(1)
Pin 1	second colomun(2)
Pin 0	third colomun(3)

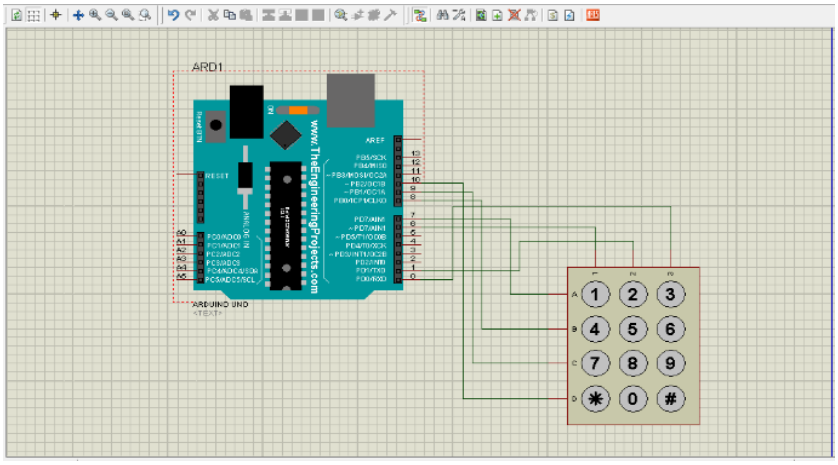

Figure 4-3: Interface of keypad

LED, Servo-motor, Buzzer with Arduino

Table 4-4: Connection of Buzzer, servo motor and Led with Arduino

Arduino	**Peripheral**
Pin 13	LED
Pin 13	Buzzer
Pin A3	Servo-motor

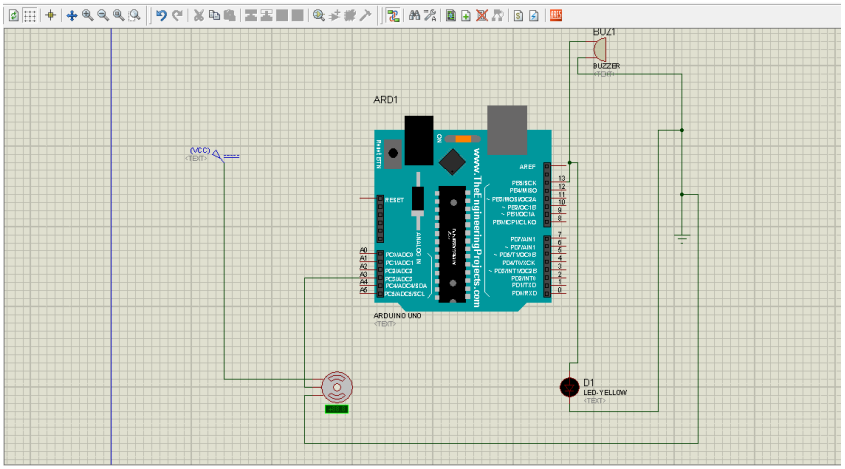

Figure 4-4: Interface of LED, servo motor, and buzzer

4.3 Software/Coding/ to Interface Components with Arduino

Since, we have used other peripherals like 16x2 LCD Display, RTC DS3232, so we first have to include libraries for that. Library requirements (*See Appendix A).*

The real-time is generated from the RTC 3232 using this library" #include <DS3232RTC.h>" we can generate to the Arduino (*See Appendix C).*

The Arduino library to interface with LCDs using #include <LiquidCrystal.h>, using this we can generate the text on the LCD (*See Appendix B).*

This switch statement is used to read the user input and saved state of the keypad and resuming the state for appropriate and accurate reminder time *(See Appendix B).*

The other hardware like Buzzer, servomotor, and LED is used to alarm, dispense and indicate when the set time is reached to remained and take medicine in the required manner we will see in detail on the working principle (*See Appendix B).*

4.4 Working Principle of Smart Medicine Reminder and Dispenser

The smart Medicine Reminder and dispenser is powered using a 5V supply. When it first boots up, it shows a welcome message as “medicine remainder” and followed by “repetition/day?”. At this time the user enters the repetition per day for taking the medicine. Then the LCD screen displays “(user input) +” times a day? *cancel #confirm” if the user enters missed data when entering ‘*’ they system back and request the previous question “repetition/day?”. and also if the user enters “#” the LCD (“Set Begin Hour:(user input)#OK *Clear”) on this time the user can enter the beginning time for its prescription after that if the user enters ‘*’ the system clear the entered value and also give a chance for other request and on the other hand if the user enters “#” the system process the user request and they will work on the proper manner on the alarm time is addressed the buzzer is buzzing parallel the servo motor dispenses the medicine and also the LED which indicate the time is reached for medicine. The LCD show (‘take a medicine’) continuously for one minute. When the user selects desired slots by pressing Keypad, the user input is recorded and the time is taken from RTC. When time is matched with the selected time slot then the buzzer starts buzzing.

Generally, can be shown as the following block diagram.

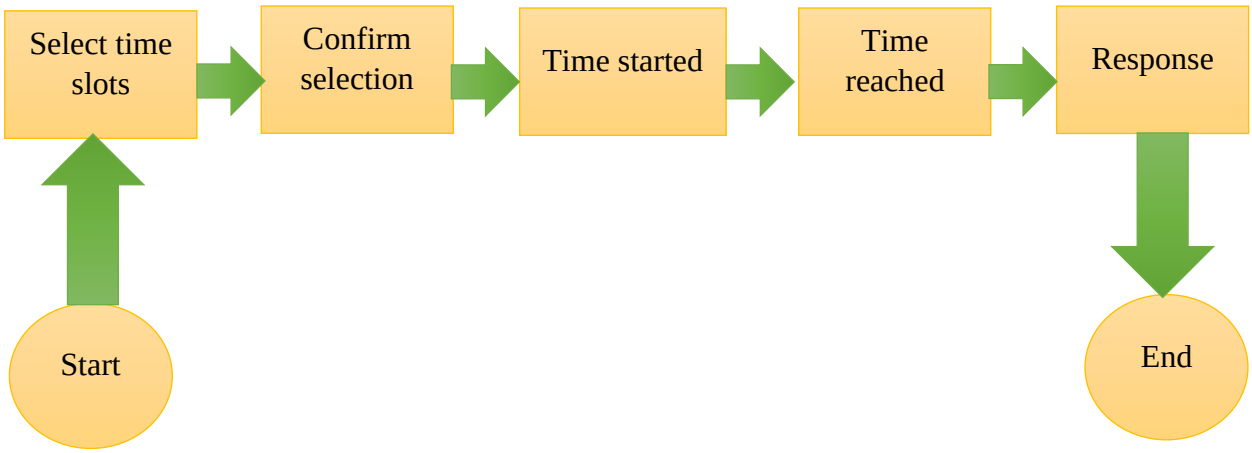

Figure 4-5: Block diagram of working principle

CHAPTER 5: RESULT AND DISCUSSION

5.1 Results

The followings are the results we have observed from the simulation of this project

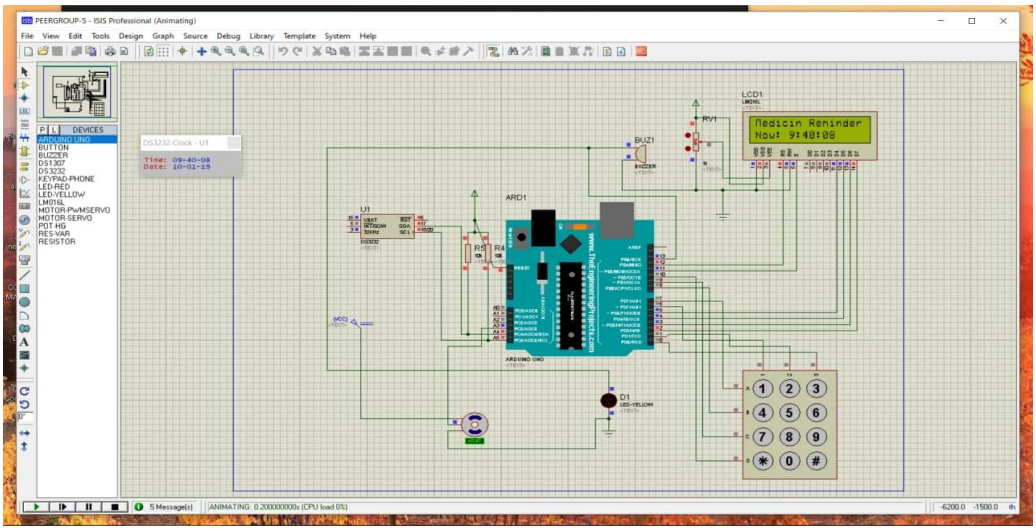

Figure 5-1: When the system is turned on

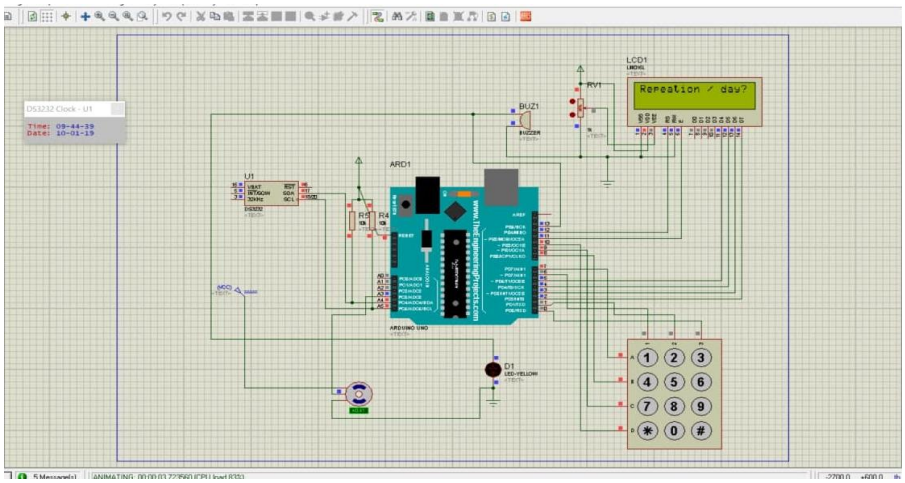

Figure 5-2: Choice for medicine intake per day

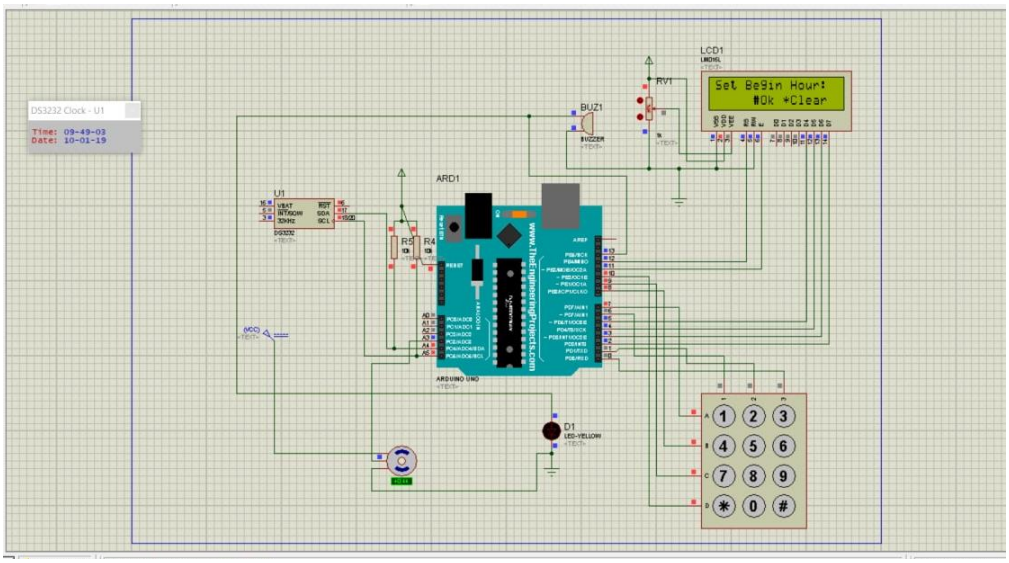

Figure 5-3: Setting the time for starting

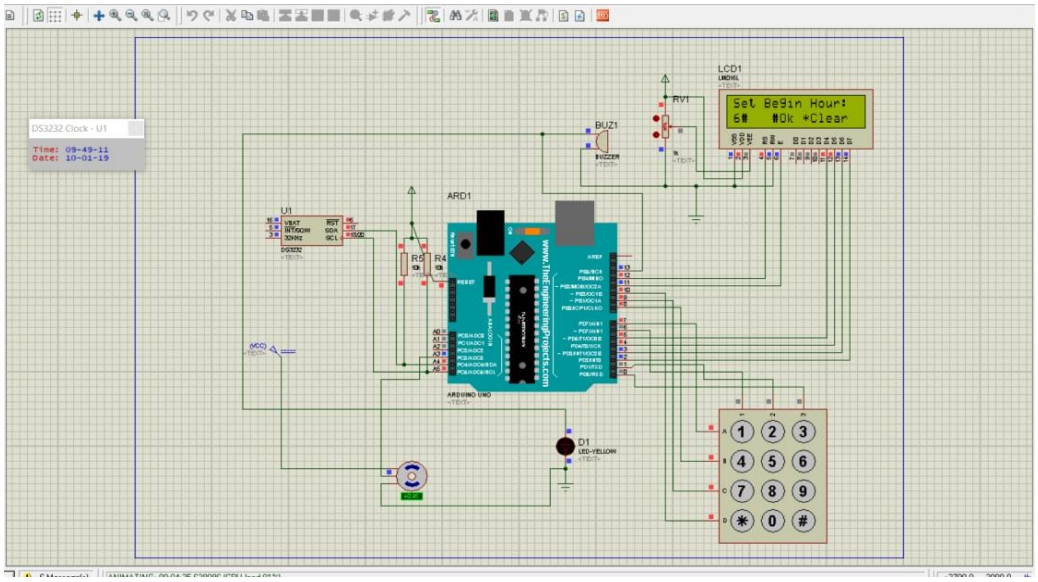

Figure 5-4: The time already set and the system starts counting

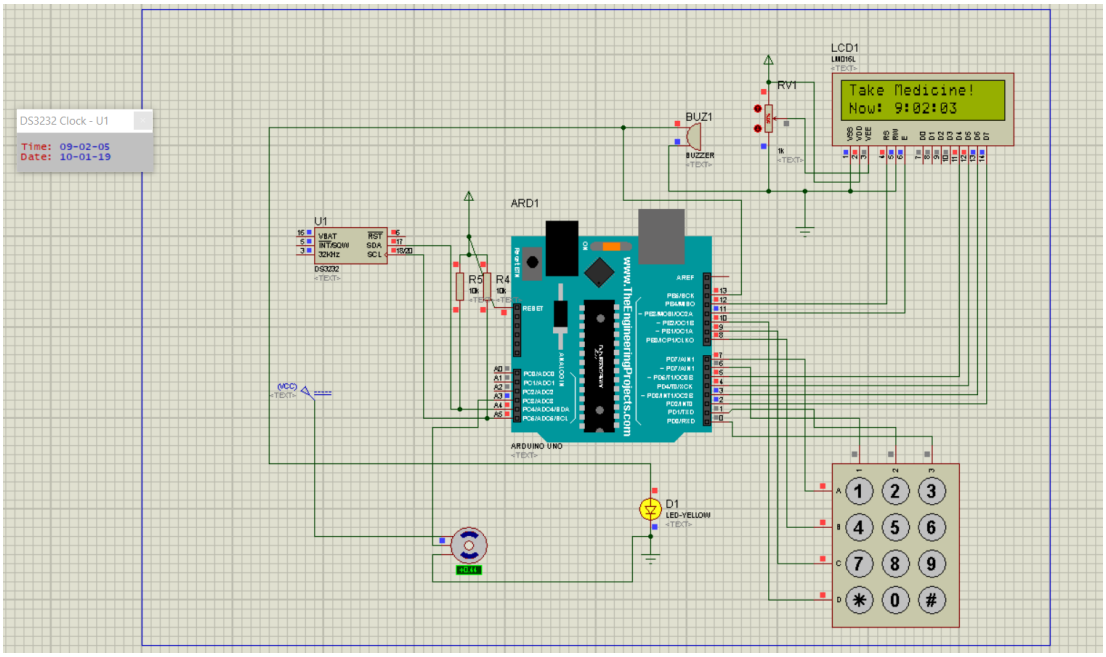

Figure 5-5: When the time of taking medicine reached

5.2 Discussion

The project has been already designed and simulated successfully. When we run the simulation, it displayed the expected results. The integrated keypad works correctly and able to enter for us any value we want to enter to adjust or set the time at which the patient takes medicine. After we run the simulation, we able to adjust the time of the patient as required and the clock displays as the system counting to reach the time set. Then when the time set has been reached, the buzzer showed us the sign as it was giving the sound for an alarming case. And also, the servo motor rotates once to dispenses the medicine that the patient should take. Another thing we have seen was that at the same time the buzzer and servo motor gave a response, the LED also gave light to show as the time of taking medicine is reached, to blind patient.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

There are many MRS developed on different platforms. These systems require either complex hardware or software applications to remind and dispense the patients about the medicine intake timings. Many of them are costly and more time-consuming. Moreover, these systems need adequate prior knowledge to operate those complex systems. So, in this work, an attempt has been made to implement a system that is economical, easily accessible, and improves medication adherence. The system can be updated by adding some additional features like notification for the next heath check-up, automatic consultation to the doctor, etc. Overall, this type of system is most important for all levels of people of different statuses like doctor's nurse aged peoples, and more patients.

6.2 Recommendation

Some further improvements can be made to this project to increase its reliability and effectiveness. The parts of improvement are on the hardware and software improvement. It is more advisable for future work and another thing waiting time increment using EPROM for snoozing and duration of long dilation on its buzzing state. Additionally, we recommend for future work by using different sensors the system can be developed to be more advanced like sensing whether the medicine is taken or not. And also, it is advisable if the system is developed to use face recognition by using sensors for detecting the specific patient to take the medicine. On the software part, we recommend that if the system further advanced on IOT system the doctors and the patient relatives can control the patient to be at a different place away from the patient.

REFERENCES

[1] J.Jose and B.Jimmy, "Patient Medication Adherence: Measures in Daily Practice," *Oman medical journal*, vol. 26(3), p. 155–159, 2011 May.

[2] S. Chawla, "The autonomous pill dispenser: Mechanizing the delivery of tablet medication," *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 1-4, 2016.

[3] C. Chariah, N. Kumia, J. R. Zubarriain, S. He, H. Rahemi and H. Jack, "Automated Pill Dispenser," 2016.

[4] P. Jayesh, "The Autonomous Pill Dispenser with Alarm and Mobile Notifications," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, p. 2502, 2019.

[5] O. Betgilu, Nguse, Tiblets, M. F. Farkas and Z. Zoltan, "Economic Growth, Investment, Population Growth and Unemployment in Ethiopia," *SHS Web of Conferences*, vol. 90, p. 1013, 2021/01/01.

[6] M. D. H, "Automatic Pill Dispenser".

APPENDICES

Appendix A: Labraries

```
#include <DS3232RTC.h>

#include <LiquidCrystal.h> // includes the LiquidCrystal Library

#include <Keypad.h>

#include <Servo.h>
```

Appendix B: Code To Interface LCD, Keypad, and Others

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Creates an LC object. Parameters: (rs, enable, d4, d5, d6, d7)

const byte ROWS = 4; //four rows
const byte COLS = 3; //three columns
char keys[ROWS][COLS] = {
  { '1', '2', '3' },
  { '4', '5', '6' },
  { '7', '8', '9' },
  { '*', '0', '#' }
};
byte rowPins[ROWS] = {7,8,9,10};
//connect to the row pinouts of the keypad
byte colPins[COLS] = {6,1,0};
 //connect to the column pinouts of the keypad
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
int buzzer = 13;
int storedHour;
int storedMinute;
Servo myservo;

int pos = 0;
void setup() {
  //Serial.begin(9600);
  myservo.attach(A3);
  setSyncProvider(RTC.get);
  lcd.begin(16, 2);
  lcd.clear();

  pinMode(buzzer, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer, LOW);
}
void loop() {
  char key;
  int gapTime;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Medicin Reminder");
  lcd.setCursor(0, 1);
  lcd.print("Now:");
  lcd.setCursor(5, 1);
  digitalClockDisplay();
  delay(2000);
  while(1){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Repeation / day?");
    lcd.setCursor(0, 1);
    while(1){
      key = keypad.getKey();
      if (key != '\0') break;
    }
    lcd.print(key);
    switch(key){
     case '1':   gapTime = 24;
                 lcd.clear();
                 lcd.setCursor(0, 0);
                 lcd.print("1 times a day?");
                 lcd.setCursor(0, 1);
                 lcd.print("*Cancel #Confirm");
                 while(1){
                   key = keypad.getKey();
                   if (key == '#') break;
```

```
              if (key == '*') { key='\0'; break;
}
            }
            break;
    case '2':   gapTime = 12;
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("2 times a day?");
            lcd.setCursor(0, 1);
            lcd.print("*Cancel #Confirm");
            while(1){
              key = keypad.getKey();
              if (key == '#') break;
              if (key == '*') { key='\0'; break;
}
            }
            break;
    case '3':   gapTime = 8;
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("3 times a day?");
            lcd.setCursor(0, 1);
            lcd.print("*Cancel #Confirm");
            while(1){
              key = keypad.getKey();
              if (key == '#') break;
              if (key == '*') { key='\0'; break;
}
            }
            break;
    case '4':   gapTime = 6;
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("4 times a day?");
            lcd.setCursor(0, 1);
            lcd.print("*Cancel #Confirm");

            while(1){

              key = keypad.getKey();
              if (key == '#') break;
              if (key == '*') { key='\0'; break;
}
            }
            break;
          default:  lcd.print("Meet
Doctor!");
    }
    if (key == '#') { key='\0'; break; }
    if (key == '*') key='\0';
  }
  int d=0;
  String beginHour;
  lcd.clear();
  while(key != '#'){
    lcd.setCursor(0, 0);
    lcd.print("Set Begin Hour:");
    lcd.setCursor(4, 1);
    lcd.print(" #Ok *Clear");
    lcd.setCursor(0, 1);
    while(1){
      key = keypad.getKey();
      if (key != '\0') break;
    }
    beginHour += key;
    if(beginHour != '\0')
    lcd.print(beginHour);
    if (key == '*') { key='\0'; beginHour='\0';
d=0;
        lcd.clear();
    }
  }
```

Appendix C: Code To Generate Alarm and Others

```
  int mediMinute = minute();
  int mediHour[4];
  for (int i=0; i<24; i=i+gapTime)
    mediHour[i/gapTime] =
(beginHour.toInt()+i);
  delay(3000);
  int clk=0;
  while (1) {
    lcd.clear();
```

```
    storedHour = mediHour[clk/gapTime];

    storedMinute = mediMinute;

    int realHour = hour();
    int realMinute = minute();
    if ((realHour == storedHour) &&
(realMinute == storedMinute)) {
      soundAlarm();
      soundAlarm2();
    }
    lcd.setCursor(0, 0);
    lcd.print("Medication Time:");
    lcd.setCursor(0, 1);
    if(mediHour[clk/gapTime] < 10){
        lcd.print('0');
        lcd.setCursor(1, 1);
        lcd.print(mediHour[clk/gapTime]);
    } else lcd.print(mediHour[clk/gapTime]);
    lcd.setCursor(2, 1);
    lcd.print(":");
    lcd.setCursor(4, 1);
    lcd.print(mediMinute);
    delay(2000);
    clk = clk + gapTime;
    if (clk == 24) clk = 0;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Medicin Reminder");
    lcd.setCursor(0, 1);
    lcd.print("Now:");
    lcd.setCursor(5, 1);
    digitalClockDisplay();
    delay(2000);
  }
}
void digitalClockDisplay()
{
    // digital clock display of the time
    lcd.print(hour());
    printDigits(minute());
    printDigits(second());
}

void printDigits(int digits)
{
    // utility function for digital clock display:
prints preceding colon and leading 0
    lcd.print(':');
    if(digits < 10)
        lcd.print('0');
    lcd.print(digits);
}
/* makes alarming beep sounds. */
void soundAlarm() {
  digitalWrite(buzzer, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
  digitalWrite(buzzer, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Take Medicine!");
  lcd.setCursor(0, 1);
  lcd.print("Now:");
  lcd.setCursor(5, 1);
  digitalClockDisplay();
  delay(2000);
  lcd.clear();
  digitalWrite(buzzer, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
  digitalWrite(buzzer, LOW);
  delay(100);
  digitalWrite(buzzer, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
  digitalWrite(buzzer, LOW);
  delay(500);
// waits 15ms for the servo to reach the
position
  }
```

```
 void soundAlarm2() {
  pos = 0;
  pos <= 180;
  pos += 1; // goes from 0 degrees to 180
degrees
    // in steps of 1 degree
   myservo.write(pos);

// tell servo to go to position in variable 'pos'

    delay(150)
```