ISOEVA-2022

6th INTERNATIONAL SYMPOSIUM OF EDUCATION AND VALUES

(6. ULUSLARARASI EĞİTİM VE DEĞERLER SEMPOZYUMU)

27-30 Ekim/October 2022

Kemer/ANTALYA

TAM METİN KİTABI

(FULL TEXT BOOK)

ISBN: 978-605-73901-6-5

KORINT
YAYINCILIK

AUTOMATIC MEDICINE REMINDER USING ARDUINO: RESEARCH NEEDED

MISS SITI NURSUHAILA BINTI MOHD JAFRI
Politeknik Sultan Salahuddin Abdul Aziz Shah, Shah Alam, Malaysia
ssiti2035@gmail.com

MISS NOR KHARUL AINA BINTI MAT DIN
Politeknik Sultan Salahuddin Abdul Aziz Shah, Shah Alam, Malaysia,
Nkaina82@gmail.com

ABSRTACT

Most of times patients may forget to take the medicines at proper time as per the specified in the prescription which may cause in late recovery from the disease/illness. It is necessary to take proper medicines in proper quantity at proper time. In this paper will introduce an Automatic Medicine Reminder using Arduino application for the patients. This application will remind their user to take proper medicines in proper quantity at proper time by automatically setting the reminders in the mobile. These reminders will be automatically set by the application as per the prescription. This reminder will remind their user patient that now it's time to take the medicine.

INTRODUCTION

This medicine reminder is focused on patients who frequently take medications or vitamin supplements, or attendants who deal with the more seasoned or patients. Our Medicine reminder is programmable that enables medical caretakers or clients to remind to take pills, and the service times for every day. At the point when the pills time has been set, the medicine reminder will remind clients or patients to take pills utilizing sound, light and vibration. The warning of pills should be taken will be shown by an android application which is held by the patient. Contrasted and the conventional medicine reminder that requires clients or attendants to stack the crate each day or consistently. This shrewd medicine reminder would essentially discharge medical attendants or clients' weight on much of the time preloading pills for patients or clients and overlook the measurements which must be taken. The remarkable problem is that patients forget to take the proper medicines in proper proportion and in proper time.

Medication adherence, which refers to the degree or extent to which a patient takes the right medication at the right time according to a doctor's prescription, has recently emerged as a serious issue because many studies have reported that non-adherence may critically affect the patient, thereby raising medical costs. Medication non adherence is a common, complex, and costly problem that contributes to poor treatment outcomes and consumes health care resources. There can

be a lot of individuals out there who need constant help – may it be our elderly people, family members, the ones who have special needs. Elders are more affected by the timing of taking a certain drug than others, in order to prevent any dysfunction or illness timing is a must. But as with aging comes poor eye sight and poor memory, what if the patient has a dementia like Alzheimer.

METHODOLOGY

The project configuration is using a RTC DS3231 that is interfaced through I2C protocol with Arduino Uno. You can also use RTC IC DS1307 for reading the time with Arduino. RTC DS3231 also has inbuilt 32k memory which can be used to store additional data. RTC module is powered through the 3.3V pin of Arduino Uno. A 16x2 LCD display is interfaced using SPI. We have divided time slots into three modes. Mode 1 selects to take medicine once/day at 8am when user presses 1st push button. Mode 2 selects to take medicine twice/day at 8am and 8pm when user presses 2nd push button. Mode 3 selects to take medicine thrice/day at 8am, 2pm and 8pm if user presses 3rd push button. This will make this device portable because this device will be easy to use anywhere and anytime. In addition, this device is also friendly user where the device is simple and easy to use this is because the operations of this device is easy to learn by adults. After that, "Automatic Medicine reminder using Arduino" is convenient to use because the design of this device is simple and convince for other people to use it.

This project is using buzzer to remind the patients of their dosage timings through buzzer ringing system so that they can stay fit and healthy. This automatic medicine reminder using Arduino Uno for intended for semi-permanent installation in objects or exhibitions. When we set up the date and time on the device, this application will remind their user to take proper medicines in proper quantity at proper time by automatically setting the reminders in the mobile. These reminders will be automatically set by the application as per the prescription. Furthermore, it also have option to select three time slots (once/twice/thrice per day) and when time will reach it start alerting the patient by buzzing the buzzer.

BLOCK DIAGRAM OF THE PROJECT

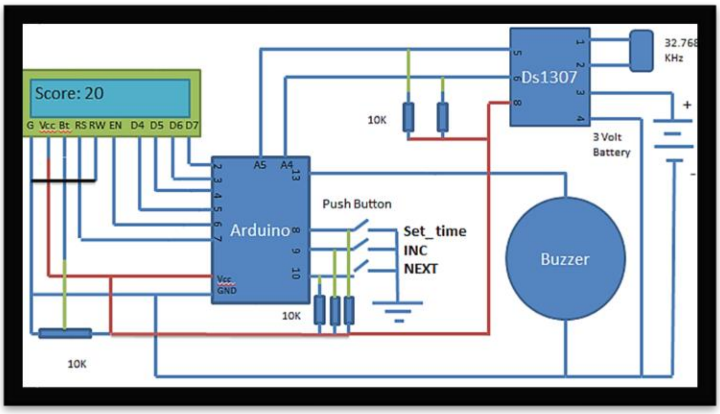


Figure 3.1 : Shows the block diagram of Automatic Medicine Reminder Using Arduino.

We start this system real time clock runs the time on 16×2 LCD. And if we want to set alarm time for medication we have to press set_mad buttons which is connected with pin number 8 of arduino. After pressing this button LCD shows Set Time 1. And then we can selects the time as we want to set for medication by using INC and Next button which is connected to pin 9 and 10 respectively of arduino. After set time 1, LCD shows set Time 2. Now using previous process set the time again. And after second time set, LCD shows again set time 3. And set this time like previous. In this system “Group medicine” indication (take group 1 medicine, take group 2 medicine and take group 3 medicine) is used instead of medicine name. When any alarm occurs LCD indicates Group medicine 1, Group medicine 2, Group medicine 3.

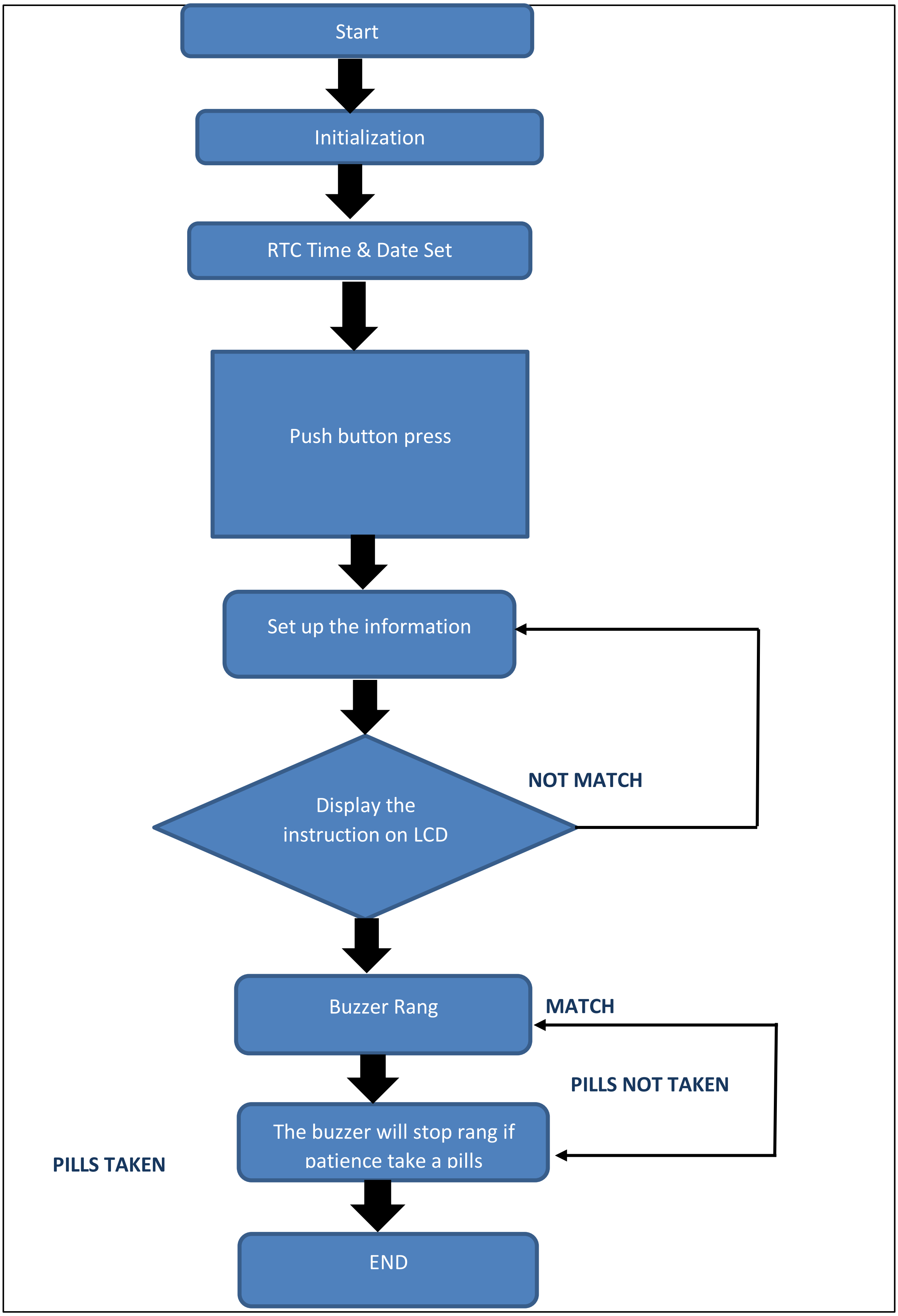


Figure 3.2 : As shown in flowchart when time & date are set through push buttons, devise will continuously compare the real time & set time.

If the time is matched, LED will blink & buzzer will ring. It then senses the box is opened by the user or not. If box is opened, LED & buzzer stops and if it is not opened, LED will continuously blinks & buzzer will continuously ring

FINDINGS

At hospitals, there are many patients and it is difficult to remind every patient to take medicine on time. The elderly when they come to senile they were difficult on remember times they should take the medication. Most of sick people need something just like alarm to them so that they have encourage to be more alert on something. When it comes to our loved ones, we always want to stay them healthy and fit. But what will happen if they get ill and forget to take medicine on time.

CONCLUSION AND DISCUSSION

Increasing medication adherence through a reminder system is one of the most common types of behavioural intervention: it targets and is helpful for patients who forget to take their medication unintentionally. Based on the above analysis, mobile phones, in-home electronic devices and portable devices used to communicate reminder messages have been shown to be useful in improving medication adherence and achieve a high user satisfaction, as summarised. Based on the three different types of reminder systems identified, we can see that electronic reminder technology has evolved in several parallel streams over the past 10 years. Simple text messages are now moving towards interactivity through interactive voice response messages. Mobile phone apps are also becoming increasingly popular as an effective and convenient way of dose reminding.

RECOMMENDATIONS

From the time the pills were introduced in the medication industry the requirement of a device that can come as a substitute to the human being for delivery of medicine. No pill dispenser to today could replace could determine the person who to give a pill to, when to give a pill and the no. of pills to be given. There are many other problems that plague the home pill user, such as someone stealing pills, forgetting to take them or having the pills available to take too many. AMDR mechanism such that it works on the principle of MECHATRONICS that is it combines the mechanics as well electronics to dispense medicine. Therefore, chances of error are very less. Following are some of the sector that can be made better by AMDR: -

i. Hospitals: –The automatic pill dispenser can ease up the pill dispensing operation of the large as well as small health care center. With the help AMDR, the medical staff, as well as a doctor, can reduce the amount of time as well frequent checkup of every patient. This will help the doctor to give time to those patients which are in critical condition or are in greater need for attention from the medical staff

ii. Home: – Vitamins and another essential dietary supplement that is required everyday before and after a meal can become a great area of interest for the use of an automatic pill dispenser. An individual keychain key can initiate the pill dispensing for the user’s personal diet regime and a buzzer can remind them when the supplements need to be taken.

iii. Elderly: –The automatic pill dispenser is an excellent way for the elderly to be reminded when to take their pills and the current access key can be worn around the neck or on a bracelet

ETHICAL TEXT

“In this article, the journal writing rules, publication principles, research and publication ethics, and journal ethical rules were followed. The responsibility belongs to the author (s) for any violations that may arise regarding the article. **REFERENCES**

[1] Park, KeeHyun & Lim, SeungHyeon, (2012) “Construction of a Medication Reminder Synchronization System based on Data Synchronization”, International Journal of Bio-Science and Bio-Technology, Vol.4, No. 4, pp1-10.

[2] AAbhimanyu Pandit Published (January 22, 2019), “Automatic Medicine Reminder Using Arduino And Component Required.” https://circuitdigest.com/microcontroller-projects/arduino-medicine-reminder

[3] Slagle, J.M., Gordon, J.S., Harris, C.E., Davison, C.L., Culpepper, D.K., Scott P. and Johnson, K.B., (2011) “MyMediHealth – Designing a next generation system for child-centered medication

management”, Journal of Biomedical Informatics, Vol. 43No. 5, pp. 27-31.

[4] Wise Techon on November 27, 2014. “The Pill Reminder using Arduino” What Exactly Does the Pill Reminder Do? https://duino4projects.com/the-pill-reminder-using-arduino/

[5]. A. Kliem, M. Hovestadt, and O. Kao.Security and Communication Architecture for Networked Medical Devices in Mobility-Aware eHealth Environments,” 2012 IEEE First Int. Conf. Mob. Serv., pp. 112–114;2012.

[6.] IJSTE - International Journal of Science Technology & Engineering | Volume 3 | Issue 10 | April 2017 ISSN (online): 2349-784X . “Smart Medicine Reminder Box”

http://www.ijste.org/articles/IJSTEV3I10093.pdf

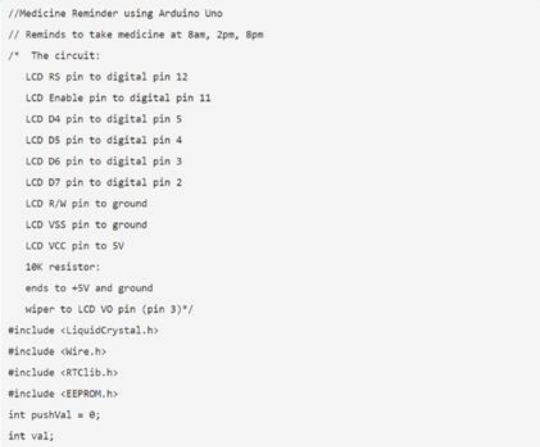
[7] Salman Khan on June ,2018 “Medicine Reminder using Arduino”, Block Diagram of Arduino based Medicine Reminder System

https://www.engineersgarage.com/medicine-reminder-using-arduino/

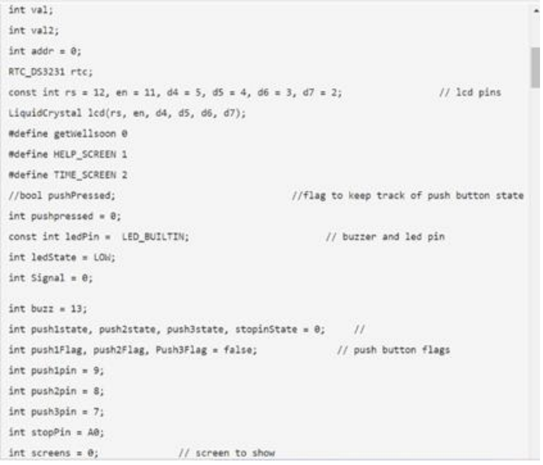
[8]. S. Huang, H. Chang, Y. Jhu, and G. Chen.The Intelligent Pill Box - Design and Implementation. pp. 235–236; 2014.

APPENDIX

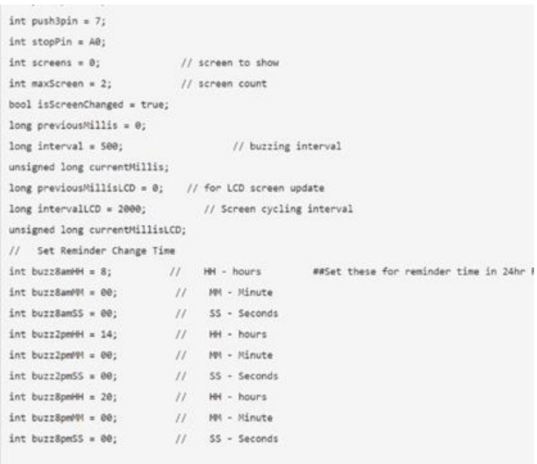
Appendix 1: Program Coding



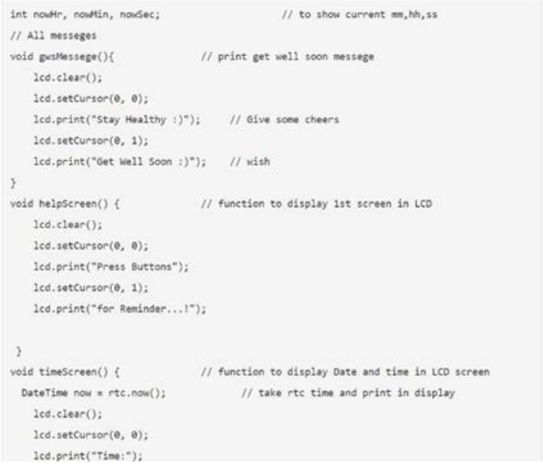
```
//Medicine Reminder using Arduino Uno
// Reminds to take medicine at 8am, 2pm, 8pm
/*  The circuit:
   LCD RS pin to digital pin 12
   LCD Enable pin to digital pin 11
   LCD D4 pin to digital pin 5
   LCD D5 pin to digital pin 4
   LCD D6 pin to digital pin 3
   LCD D7 pin to digital pin 2
   LCD R/W pin to ground
   LCD VSS pin to ground
   LCD VCC pin to 5V
   10K resistor:
   ends to +5V and ground
   wiper to LCD VO pin (pin 3)*/
#include <LiquidCrystal.h>
#include <Wire.h>
#include <RTClib.h>
#include <EEPROM.h>
int pushVal = 0;
int val;
```



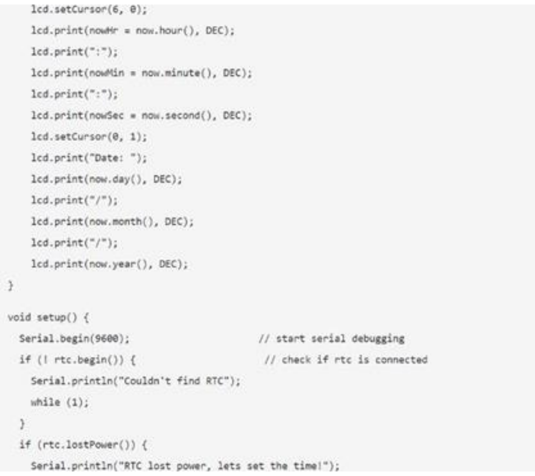
```
int val;
int val2;
int addr = 0;
RTC_DS3231 rtc;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;               // lcd pins
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define getWellsoon 0
#define HELP_SCREEN 1
#define TIME_SCREEN 2
//bool pushPressed;                               //flag to keep track of push button state
int pushpressed = 0;
const int ledPin =  LED_BUILTIN;                        // buzzer and led pin
int ledState = LOW;
int Signal = 0;

int buzz = 13;
int push1state, push2state, push3state, stopinState = 0;     //
int push1Flag, push2Flag, Push3Flag = false;             // push button flags
int push1pin = 9;
int push2pin = 8;
int push3pin = 7;
int stopPin = A0;
int screens = 0;              // screen to show
```



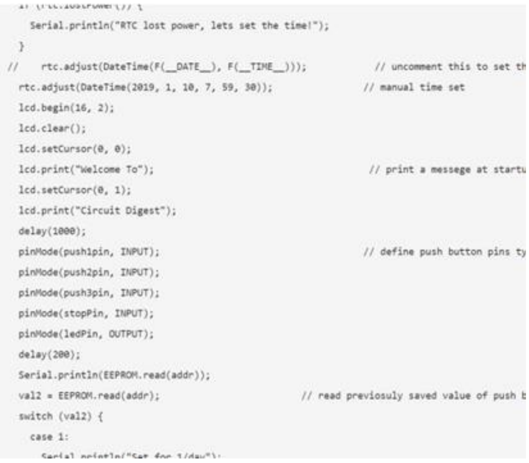
```
int push3pin = 7;
int stopPin = A0;
int screens = 0;              // screen to show
int maxScreen = 2;            // screen count
bool isScreenChanged = true;
long previousMillis = 0;
long interval = 500;                   // buzzing interval
unsigned long currentMillis;
long previousMillisLCD = 0;    // for LCD screen update
long intervalLCD = 2000;          // Screen cycling interval
unsigned long currentMillisLCD;
//   Set Reminder Change Time
int buzz8amHH = 8;          //    HH - hours         ##Set these for reminder time in 24hr
int buzz8amMM = 00;          //    MM - Minute
int buzz8amSS = 00;          //    SS - Seconds
int buzz2pmHH = 14;          //    HH - hours
int buzz2pmMM = 00;          //    MM - Minute
int buzz2pmSS = 00;          //    SS - Seconds
int buzz8pmHH = 20;          //    HH - hours
int buzz8pmMM = 00;          //    MM - Minute
int buzz8pmSS = 00;          //    SS - Seconds
```



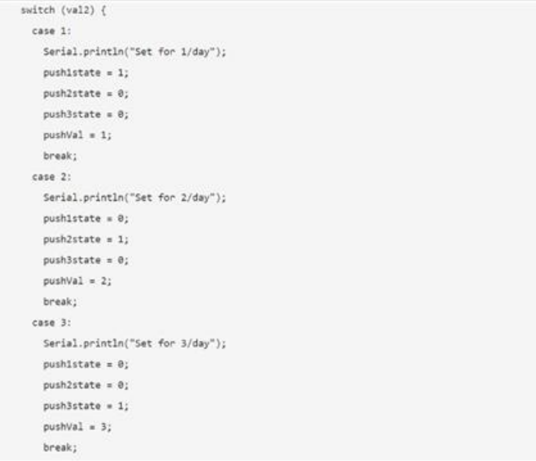
```
int nowHr, nowMin, nowSec;                     // to show current mm,hh,ss
// All messeges
void gwsMessege(){              // print get well soon messege
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Stay Healthy :)");     // Give some cheers
    lcd.setCursor(0, 1);
    lcd.print("Get Well Soon :)");    // wish
}
void helpScreen() {             // function to display 1st screen in LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Press Buttons");
    lcd.setCursor(0, 1);
    lcd.print("for Reminder...!");

 }
void timeScreen() {             // function to display Date and time in LCD screen
  DateTime now = rtc.now();             // take rtc time and print in display
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Time:");
```



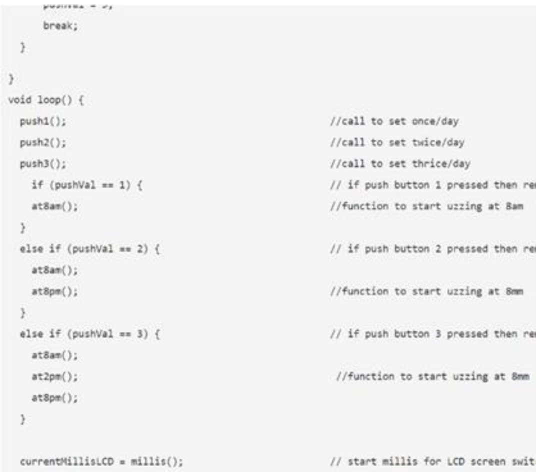
```
    lcd.setCursor(6, 0);
    lcd.print(nowHr = now.hour(), DEC);
    lcd.print(":");
    lcd.print(nowMin = now.minute(), DEC);
    lcd.print(":");
    lcd.print(nowSec = now.second(), DEC);
    lcd.setCursor(0, 1);
    lcd.print("Date: ");
    lcd.print(now.day(), DEC);
    lcd.print("/");
    lcd.print(now.month(), DEC);
    lcd.print("/");
    lcd.print(now.year(), DEC);
}

void setup() {
  Serial.begin(9600);                      // start serial debugging
  if (! rtc.begin()) {                      // check if rtc is connected
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (rtc.lostPower()) {
    Serial.println("RTC lost power, lets set the time!");
```



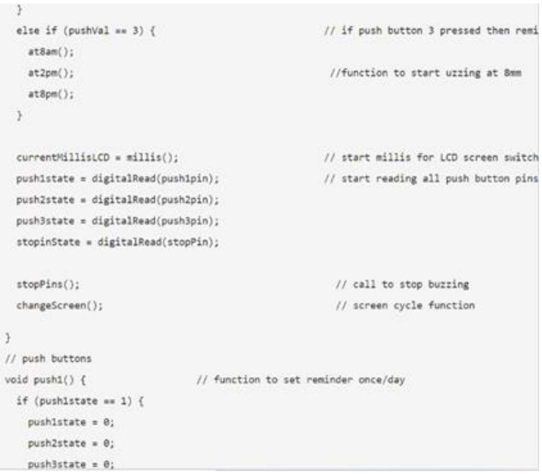
```
  if (rtc.lostPower()) {
    Serial.println("RTC lost power, lets set the time!");
  }
//    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));           // uncomment this to set th
  rtc.adjust(DateTime(2019, 1, 10, 7, 59, 30));               // manual time set
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Welcome To");                                     // print a messege at startu
  lcd.setCursor(0, 1);
  lcd.print("Circuit Digest");
  delay(1000);
  pinMode(push1pin, INPUT);                                   // define push button pins ty
  pinMode(push2pin, INPUT);
  pinMode(push3pin, INPUT);
  pinMode(stopPin, INPUT);
  pinMode(ledPin, OUTPUT);
  delay(200);
  Serial.println(EEPROM.read(addr));
  val2 = EEPROM.read(addr);                        // read previosuly saved value of push b
  switch (val2) {
    case 1:
      Serial.println("Set for 1/day");
```



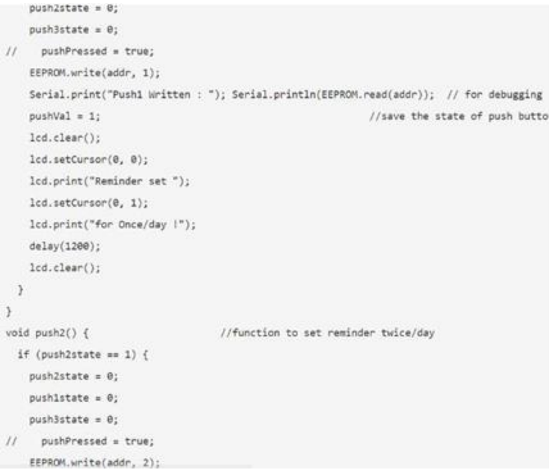
```
  switch (val2) {
    case 1:
      Serial.println("Set for 1/day");
      push1state = 1;
      push2state = 0;
      push3state = 0;
      pushVal = 1;
      break;
    case 2:
      Serial.println("Set for 2/day");
      push1state = 0;
      push2state = 1;
      push3state = 0;
      pushVal = 2;
      break;
    case 3:
      Serial.println("Set for 3/day");
      push1state = 0;
      push2state = 0;
      push3state = 1;
      pushVal = 3;
      break;
```



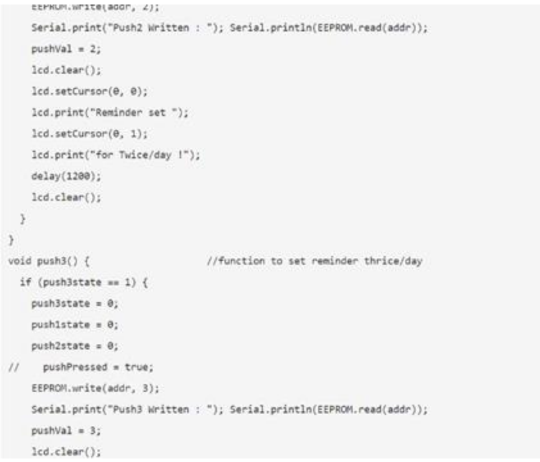
```
      pushVal = 3;
      break;
  }

}
void loop() {
  push1();                                    //call to set once/day
  push2();                                    //call to set twice/day
  push3();                                    //call to set thrice/day
    if (pushVal == 1) {                       // if push button 1 pressed then re
    at8am();                                  //function to start uzzing at 8am
  }
  else if (pushVal == 2) {                    // if push button 2 pressed then re
    at8am();
    at8pm();                                  //function to start uzzing at 8mm
  }
  else if (pushVal == 3) {                    // if push button 3 pressed then re
    at8am();
    at2pm();                                   //function to start uzzing at 8mm
    at8pm();
  }

  currentMillisLCD = millis();                // start millis for LCD screen swit
```



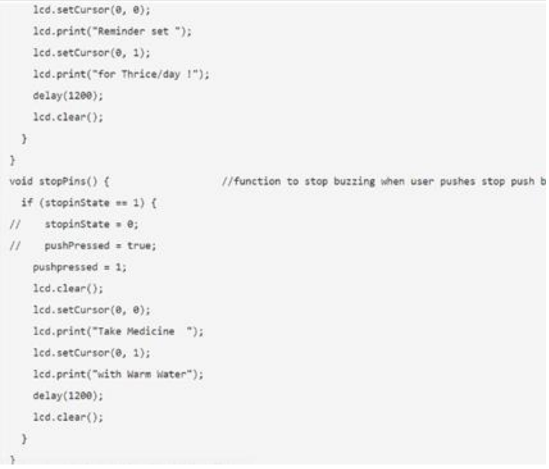
```
  }
  else if (pushVal == 3) {                    // if push button 3 pressed then remi
    at8am();
    at2pm();                                   //function to start uzzing at 8mm
    at8pm();
  }

  currentMillisLCD = millis();                // start millis for LCD screen switch
  push1state = digitalRead(push1pin);         // start reading all push button pins
  push2state = digitalRead(push2pin);
  push3state = digitalRead(push3pin);
  stopinState = digitalRead(stopPin);

  stopPins();                                  // call to stop buzzing
  changeScreen();                              // screen cycle function

}
// push buttons
void push1() {                    // function to set reminder once/day
  if (push1state == 1) {
    push1state = 0;
    push2state = 0;
    push3state = 0;
```



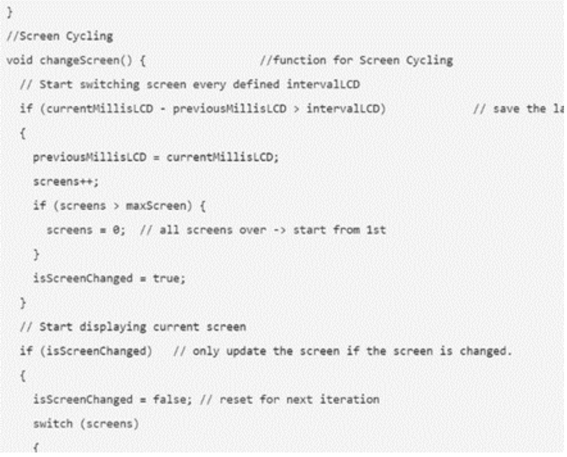
```
    push2state = 0;
    push3state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 1);
    Serial.print("Push1 Written : "); Serial.println(EEPROM.read(addr));  // for debugging
    pushVal = 1;                                             //save the state of push butto
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Once/day !");
    delay(1200);
    lcd.clear();
  }
}
void push2() {                      //function to set reminder twice/day
  if (push2state == 1) {
    push2state = 0;
    push1state = 0;
    push3state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 2);
```



```
    EEPROM.write(addr, 2);
    Serial.print("Push2 Written : "); Serial.println(EEPROM.read(addr));
    pushVal = 2;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Twice/day !");
    delay(1200);
    lcd.clear();
  }
}
void push3() {                    //function to set reminder thrice/day
  if (push3state == 1) {
    push3state = 0;
    push1state = 0;
    push2state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 3);
    Serial.print("Push3 Written : "); Serial.println(EEPROM.read(addr));
    pushVal = 3;
    lcd.clear();
```



```
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Thrice/day !");
    delay(1200);
    lcd.clear();
  }
}
void stopPins() {                   //function to stop buzzing when user pushes stop push b
  if (stopinState == 1) {
//    stopinState = 0;
//    pushPressed = true;
    pushpressed = 1;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Take Medicine  ");
    lcd.setCursor(0, 1);
    lcd.print("with Warm Water");
    delay(1200);
    lcd.clear();
  }
}
```



```
}
//Screen Cycling
void changeScreen() {                 //function for Screen Cycling
  // Start switching screen every defined intervalLCD
  if (currentMillisLCD - previousMillisLCD > intervalLCD)             // save the la
  {
    previousMillisLCD = currentMillisLCD;
    screens++;
    if (screens > maxScreen) {
      screens = 0;  // all screens over -> start from 1st
    }
    isScreenChanged = true;
  }
  // Start displaying current screen
  if (isScreenChanged)   // only update the screen if the screen is changed.
  {
    isScreenChanged = false; // reset for next iteration
    switch (screens)
    {
```