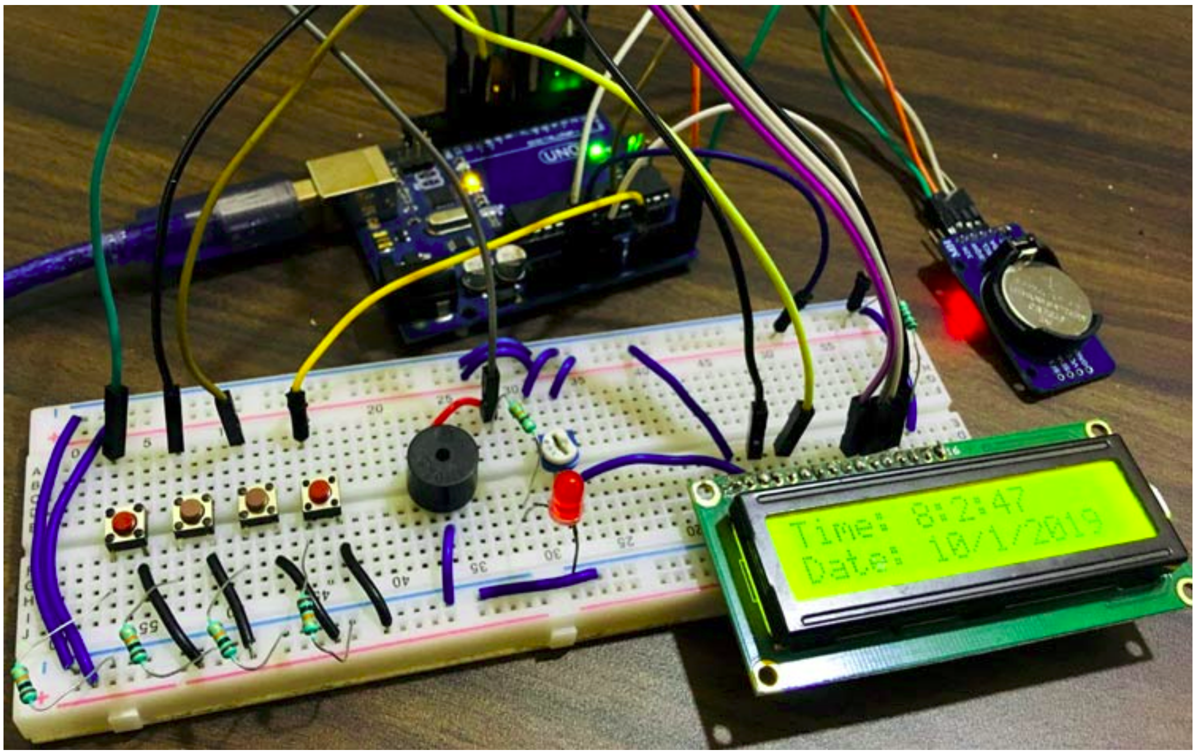
Automatic Medicine Reminder Using Arduino (/microcontroller-projects/arduino-medicine-reminder)



Medicine Reminder Using Arduino

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. We would be worried, right? At hospitals, there are many patients and it is difficult to remind every patient to take medicine on time. The traditional ways require human efforts to remind to take medicines on time. The digital era doesn't follow that and we can use machines to do that. The application of **Smart Medicine Reminder** is very wide and can be used by patients at home, doctors at hospitals and at many other places. When it comes to reminding, there can be many ways to remind it:

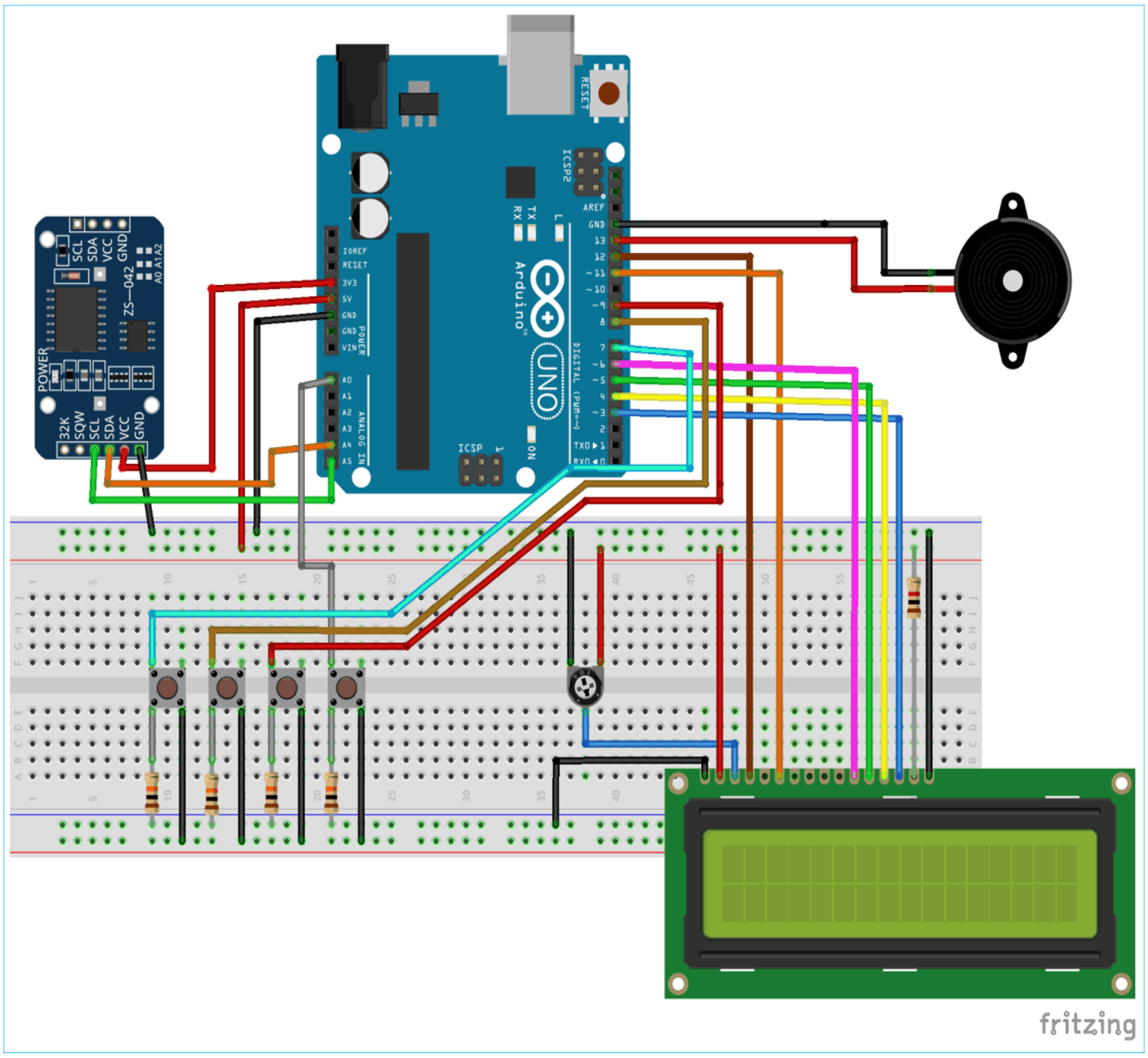
1. Show it on a display
2. Send notification on email or Phone
3. Using mobile apps
4. Buzz alarm
5. Using Bluetooth/ Wi-Fi
6. Get a call
7. Remind for next medicine time while reminding current time

We can combine ways depending upon the need. To keep things simple here we made a **simple Medicine Reminder Using Arduino** which reminds to take medicines 1 or 2 or 3 times a day. The time slot can be selected using push buttons. Also, it shows current Date and Time. We will further extend it to a IoT project in coming articles where a email or SMS notification will be sent to user. This medication reminder can also be integrated with Patient Monitoring System (https://circuitdigest.com/microcontroller-projects/iot-based-patient-monitoring-system-using-esp8266-and-arduino).

Components Required

1. Arduino Uno (We can use other Arduino boards also, like Promini, Nano)
2. RTC DS3231 module
3. 16x2 LCD Display
4. Buzzer
5. Led(any color)
6. Breadboard
7. Push Buttons
8. 10K Potentiometer
9. 10K,1K Resistors
10. Jumper Wires

Arduino Medicine Reminder Circuit Diagram and Connections

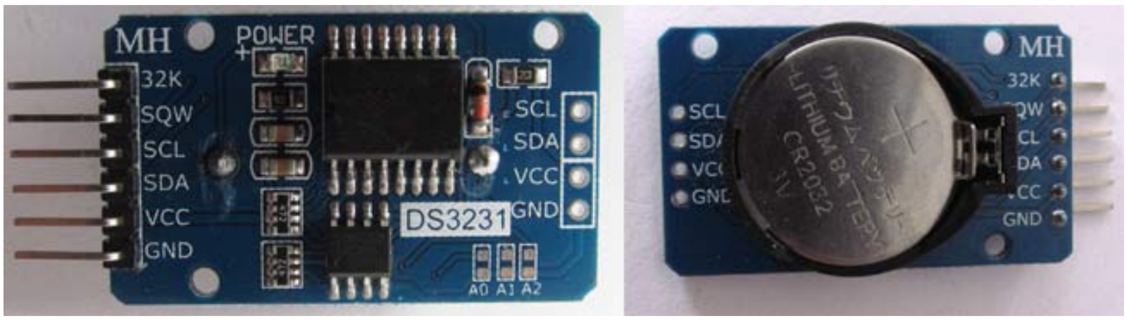


(/fullimage?i=circuitdiagram_mic/Circuit-Diagram-Medicine-Reminder-Using-Arduino.png)

Below are the **pin connections** of Arduino with different peripherals

Arduino Pins	**Peripheral 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
7	----------------------------> 3rd push button
8	----------------------------> 2nd push button
9	----------------------------> 1st push button
11	----------------------------> EN pin of 16x2 LCD Display
12	----------------------------> RS pin of 16x2 LCD Display
13	----------------------------> +Ve Pin of Buzzer and Led
A0	----------------------------> Stop Push Button
A4	----------------------------> SDA of DS3231
A5	----------------------------> SCL of DS3231

- 3.3V ----------------------------> Vcc of DS3231
- Gnd ----------------------------> Gnd

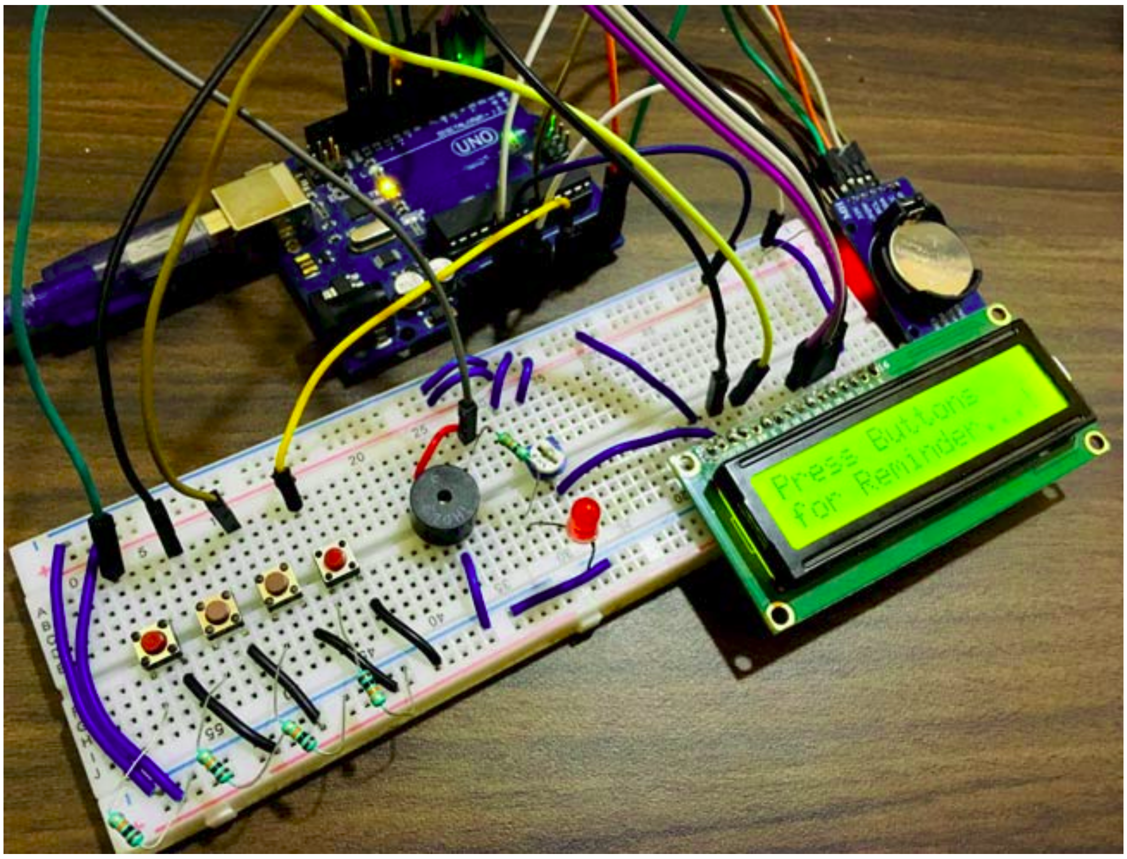


In this **Medicine Reminder Project**, RTC DS3231 is interfaced through I2C protocol with Arduino Uno (https://circuitdigest.com/microcontroller-projects/arduino-data-logger-project). You can also use RTC IC DS1307 for reading the time with Arduino (https://circuitdigest.com/microcontroller-projects/arduino-alarm-clock). 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 (https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet) is interfaced using SPI. A **buzze**r is used to alert and remind that it's time to take medicine. **Four push buttons** are used where each has distinct select feature. The first push button is used for reminding to take medicine once per day. The second push button is used to remind twice per day and the third push button is used to remind thrice per day. The fourth push button is used to stop the buzzer when user has heard the alert.

Working of Automatic Medicine Reminder System

The **Pill Reminder Alarm** is powered using 5V supply. When it first boots up, it shows a welcome massage as "*Welcome to Circuit Digest*". The LCD screen is set to cycle in three screens. The 1st screen shows massage as "*Stay Healthy, Get Well Soon*". The second screen is a help screen which tells to press select push button to select any one time-slot to remind (once/twice/thrice in a day). The time slot is changeable in program and can be configured accordingly. Right now we have fixed this into three durations i.e. 8am, 2pm, and 8pm.

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.



We can also add a feature to snooze the buzzer for 10 minutes (not included in this project). When user selects desired slots by pressing push buttons, the user input is recorded and the time is taken from RTC. When time is matched with selected time slot then the buzzer starts buzzing. User can stop the buzzer by pressing STOP button. The same process continues for the next slot reminder. Complete process is shown in the **Video** given at the end of this aricle.

Programming Arduino UNO for Medicine Reminder

It is very easy to write program once you have thought of the ways to remind taking the pills. Here it will show the reminder on display, buzz a buzzer and indicate it using LED. 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. Then the whole system will look like following:

User gets help instructions on display > User selects time slots (once/day, twice/day, thrice/day) > Print confirmation message on display > Time keeping started > Buzzer and LED starts when time matches with user selected slot > User stops by pushing a stop push button > End

We can change the program and hardware if we want to add more features. To understand in much simpler way, we have broken down program into small functions. The functions are easy to understand and implement. The complete program is given at the end of this project. Let's start with the program.

Since, we have used other peripherals like 16x2 LCD Display, RTC DS3231, so we first have to **include libraries f**or that. Library required are as following:

```
<LiquidCrystal.h>
<RTClib.h>   (https://github.com/adafruit/RTClib)
<EEPROM.h>
<Wire.h>
```

The EEPROM library is used to keep the track of user select input if Arduino is not turned on. And when user power on the Arduino it gets the previous state of push buttons using EEPROM library. The Wire.h library is used since the RTC DS3231 module is communicated using I2C.

Always **check if the RTC is properly wired or it is not damaged**, since RTC will play an important role in time keeping of the whole reminder system.

```
  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!");
  }
```

The time adjustment can be done in two ways, either automatically using system compile time or by entering it manually. Once we have set the time, comment the below lines unless you want to change the RTC time again.

```
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
//rtc.adjust(DateTime(2019, 1, 10, 7, 59, 52));
```

This **switch statement is used to read the previously saved state of the push button** and resuming the state for appropriate and accurate reminder time.

```
val2 = EEPROM.read(addr);                         // read previosuly saved value of pus
  switch (val2) {
    case 1:
      Serial.println("Set for 1/day");
      push1state = 1;
      push2state = 0;
      push3state = 0;
      pushVal = 01;
      break;

    case 2:
      Serial.println("Set for 2/day");
      push1state = 0;
      push2state = 1;
      push3state = 0;
      pushVal = 10;
      break;

    case 3:
      Serial.println("Set for 3/day");
      push1state = 0;
      push2state = 0;
      push3state = 1;
      pushVal = 11;
      break;
  }
```

This statement is used to get the millis to use for timing and control of the defined interval screen cycling.

```
currentMillisLCD = millis(); // start millis for LCD screen switching at defined interv
```

Start reading the digital pins connected to push buttons.

```
push1state = digitalRead(push1pin);
push2state = digitalRead(push2pin);
push3state = digitalRead(push3pin);
stopinState = digitalRead(stopPin);
```

Below function is used to **read the push button state and write it to EEPROM**. Whenever the push button gets pressed the state is written to EEPROM. Also it prints the message on LCD display of the selected user input choice. Similarly the functions *push2()* and *push3()* is used.

```
void push1() {                   // function to set reminder once/day
  if (push1state == 1) {
    push1state = 0;
    push2state = 0;
    push3state = 0;
//    pushPressed = true;
    EEPROM.write(addr, 1);
    Serial.print("Push1 Written : "); Serial.println(EEPROM.read(addr));  // for debugg
    pushVal = 1;                                          //save the state of push b
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reminder set ");
    lcd.setCursor(0, 1);
    lcd.print("for Once/day !");
    delay(1200);
    lcd.clear();
  }
}
```

Below function is used to stop the buzzer and led. It is always good to give suggestions. Print a suggestion message on display “Take medicine with warm water”.

```
void stopPins() {                   //function to stop buzzing when user pushes stop pu

  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();
  }
}
```

The below function is independent of the time keeping but always cycles in three screens which helps user. As we are keeping a care to patients lets print a greeting message as we know that emotional support is very helpful in healing patients in more quick time. You can choose your own creative message. Let’s **print a message as “Stay healthy, Get well soon”.**

The second screen is for **giving instruction to patients as “Press buttons for reminder..”.** The Third screen is used to simply **show current date and time**.

```
void changeScreen() {                 //function for Screen Cycling

  // Start switching screen every defined intervalLCD
  if (currentMillisLCD - previousMillisLCD > intervalLCD)             // save the last
  {
    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)
    {
      case getWellsoon:
        gwsMessege();                // get well soon message
        break;
      case HELP_SCREEN:
        helpScreen();               // instruction screen
        break;
      case TIME_SCREEN:
        timeScreen();                  // to print date and time
        break;
      default:
        //NOT SET.
        break;
    }
  }
}
```

This function is used to start buzzing and blinking the LED simultaneously if the selected time has reached.

```
void startBuzz() {                    // function to start buzzing when time reaches to
//  if (pushPressed == false) {

if (pushpressed == 0) {
    Serial.println("pushpressed is false in blink");
    unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;         // save the last time you blinked the LED
      Serial.println("Start Buzzing");

if (ledState == LOW) {                  // if the LED is off turn it on and vice-versa:
        ledState = HIGH;
      }  else {
        ledState = LOW;
      }
      digitalWrite(ledPin, ledState);
    }
  }

else if (pushpressed == 1) {
    Serial.println("pushpressed is true");
    ledState = LOW;
    digitalWrite(ledPin, ledState);
  }
}
```

This function is used to compare the user selected time slot at 8am and starts buzzing the buzzer and blinking the LED until user presses the stop push button. Similarly the functions ***void at2pm()*** and ***void at8pm*** is used to start buzzer and led at 2pm and 8pm.

```
void at8am() {                      // function to start buzzing at 8am
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz8amHH) {
    if (int(now.minute()) >= buzz8amMM) {
      if (int(now.second()) > buzz8amSS) {
        /////////////////////////////////////////////////////
        startBuzz();
        /////////////////////////////////////////////////////
      }
    }
  }
}
```

This is how you can simply make your own Automatic Medicine Reminder using Arduino. You can also use ESP8266 with Arduino (https://circuitdigest.com/arduino-esp8266-projects) to make it an IoT project which will be able to send email alert to the user.

Complete Code and demonstration Video is given below.

RECOMMENDED TI WHITEPAPERS

Code

```
//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 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 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 Format
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!");
 }
//    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));          // uncomment this to set the current time and then
comment in next upload when u set the time
 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 startup
 lcd.setCursor(0, 1);
 lcd.print("Circuit Digest");
 delay(1000);
 pinMode(push1pin, INPUT);                    // define push button pins type
 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 button to start from where it was left
previously
 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;
 }
}
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 remind at 8am
```

```
    at8am();                                    //function to start uzzing at 8am
  }
  else if (pushVal == 2) {                      // if push button 2 pressed then remind at 8am and 8pm
    at8am();
    at8pm();                                    //function to start uzzing at 8mm
  }
  else if (pushVal == 3) {                      // if push button 3 pressed then remind at 8am and 8pm
    at8am();
    at2pm();                                     //function to start uzzing at 8mm
    at8pm();
  }

  currentMillisLCD = millis();                  // start millis for LCD screen switching at defined interval of time
  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;
//    pushPressed = true;
    EEPROM.write(addr, 1);
    Serial.print("Push1 Written : "); Serial.println(EEPROM.read(addr));  // for debugging
    pushVal = 1;                                //save the state of push button-1
    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);
   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 button
 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();
  }
}
void startBuzz() {                    // function to start buzzing when time reaches to defined interval
//  if (pushPressed == false) {
 if (pushpressed == 0) {
    Serial.println("pushpressed is false in blink");
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;         // save the last time you blinked the LED
      Serial.println("Start Buzzing");
      if (ledState == LOW) {                  // if the LED is off turn it on and vice-versa:
        ledState = HIGH;
      }  else {
        ledState = LOW;
      }
      digitalWrite(ledPin, ledState);
    }
  }
  else if (pushpressed == 1) {
    Serial.println("pushpressed is true");
    ledState = LOW;
    digitalWrite(ledPin, ledState);
  }
}
void at8am() {                      // function to start buzzing at 8am
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz8amHH) {
    if (int(now.minute()) >= buzz8amMM) {
      if (int(now.second()) > buzz8amSS) {
        /////////////////////////////////////////////////////
        startBuzz();
        /////////////////////////////////////////////////////
      }
    }
  }
```

```
}
void at2pm() {                     // function to start buzzing at 2pm
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz2pmHH) {
    if (int(now.minute()) >= buzz2pmMM) {
      if (int(now.second()) > buzz2pmSS) {

        /////////////////////////////////////////////////////
        startBuzz();
        /////////////////////////////////////////////////////
      }
    }
  }
}
void at8pm() {                      // function to start buzzing at 8pm
  DateTime now = rtc.now();
  if (int(now.hour()) >= buzz8pmHH) {
    if (int(now.minute()) >= buzz8pmMM) {
      if (int(now.second()) > buzz8pmSS) {

        /////////////////////////////////////////////////////
        startBuzz();
        /////////////////////////////////////////////////////
      }
    }
  }
}
//Screen Cycling
void changeScreen() {             //function for Screen Cycling
  // Start switching screen every defined intervalLCD
  if (currentMillisLCD - previousMillisLCD > intervalLCD)           // save the last time you changed the display
  {
    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)
  {
    case getWellsoon:
      gwsMessege();              // get well soon message
      break;
    case HELP_SCREEN:
      helpScreen();             // instruction screen
      break;
    case TIME_SCREEN:
      timeScreen();                // to print date and time
      break;
    default:
      //NOT SET.
      break;
  }
 }
}
```