```
//ARDUINO CODE FOR PILL REMINDER DEVICE
//Developed by Sourav Paul

#include <Wire.h>
#include "RTClib.h"
#include <avr/sleep.h>
#include <avr/wdt.h>
#include <avr/power.h>


const int clockPower = 14; //RTC module VCC power fed from Atmega328P IC Pin
const int boxOpenPin = 15; //device top lid open / close detection pin
const int buzzerPin = 16; // buzzer for Alarm
const int ledPin = 17; // led for missed pill indicator


RTC_DS1307 rtc;  //Real-Time Clock module declaration
DateTime now;
boolean boxOpened = false;
boolean ledOn = false;
// watchdog interrupt
ISR (WDT_vect)
{
  wdt_disable();  // disable watchdog
}  // end of WDT_vect

void setup() {
 //Begin RTC and adjust initial time
  toggleRTCPower(true);
  //Set pins as user input
  for (int digitalPin=2; digitalPin<=17; digitalPin++)
  {
    pinMode(digitalPin, INPUT);
  }

}

void loop() {
  sleep(); // 2 secs sleep to save power thus increasing battery life

  ledAlertOff(); // turn off led missed pill indicator alert if it is already on

  toggleRTCPower(false);

  //check for user scheduled alarm time through 12 pins DIP switch
  checkTime(2,23,30,0,30);    //12 AM

  //loop for check time from @2 AM to @10 PM at 2 hours interval
  for(int pinCounter=3, timeCounter=0; pinCounter<=13; pinCounter++)
     {
       checkTime(pinCounter,++timeCounter, 30,++timeCounter,30 );
     }
}

//function to turn LED missed pill indiactor alarm off
void ledAlertOff()
{
   if(ledOn == true && digitalRead(boxOpenPin) == HIGH)
  {
      digitalWrite(ledPin, LOW);
      pinMode(ledPin, INPUT);
      ledOn = false;
  }
}

//method to check for current time through RTC and turn on buzzer alert if device top lid is not opened
void checkTime(int pin, int minHr, int minMin, int maxHr, int maxMin)
```

```
{
  if (digitalRead(pin) == HIGH)
  {
    if ((now.hour() == minHr && now.minute() > minMin) || (now.hour() == maxHr && now.minute() < maxMin)) //check the boxOpenPin status wit
    {
      if (digitalRead(boxOpenPin) == HIGH) // check whether boxOpenPin is HIGH through device top lid open/close system
      {
        boxOpened = true;
      }
    }
    if (now.hour() == maxHr && now.minute() >= maxMin && now.minute() < (maxMin + 5) &&  boxOpened == false )
    {
      alarm();
    }
  }
  if (now.hour() == maxHr && now.minute() >= (maxMin + 5))
  {
    boxOpened = false;
  }
}

//function to give alarm sound through the buzzer
void alarm() // currently duration set for 5 mins or 300 secs
{
  for (int timeLoop=1; timeLoop<=150; timeLoop++)
  {
    pinMode(buzzerPin, OUTPUT);
    digitalWrite(buzzerPin, HIGH);
    delay(2000);
    //stop alarm
    if (digitalRead(boxOpenPin) == HIGH)
    {
      digitalWrite(buzzerPin, LOW);
      for (int remainingLoop=1; remainingLoop<=(150-timeLoop); remainingLoop++)
      {
       ledAlertOff();
       sleep(); // sleeps for 2 secs interval
      }
      break;
    }
 // Turn on LED missed pill indicator
    if(timeLoop == 150)
    {
      pinMode(ledPin, OUTPUT);
      digitalWrite(ledPin, HIGH);
      ledOn = true;
    }
  }
  pinMode(buzzerPin, INPUT);
}

//sleep code
void sleep()
{
  // disable ADC
  ADCSRA = 0;
  // clear various "reset" flags
  MCUSR = 0;
  // allow changes, disable reset
  WDTCSR = bit (WDCE) | bit (WDE);
  // set interrupt mode and an interval
  WDTCSR = bit (WDIE) | bit (WDP2) | bit (WDP1) | bit (WDP0);    // set WDIE, and 2 seconds delay
  wdt_reset();  // pat the dog
  set_sleep_mode (SLEEP_MODE_PWR_DOWN);
  noInterrupts ();          // timed sequence follows
  sleep_enable();
  // turn off brown-out enable in software
```

```
  MCUCR = bit (BODS) | bit (BODSE);
  MCUCR = bit (BODS);
  interrupts ();             // guarantees next instruction executed
  sleep_cpu ();
  // cancel sleep as a precaution
  sleep_disable();
}

//Toggle power to the RTC VCC pin through Atmega328P IC pin
void toggleRTCPower(boolean resetRTC)
{
// power up RTC clock chip
 pinMode (clockPower, OUTPUT);
 digitalWrite (clockPower, HIGH);

 // activate I2C
  Wire.begin();

  if(resetRTC == true)
  {
  rtc.begin(); // Start the RTC library code
  // June 20, 2016, 10PM is set as initial date and time (24 hours format used)
  rtc.adjust(DateTime(2016, 6, 20, 22, 0, 0));
  }
 // find the time
 now = rtc.now();

 // finished with clock
 digitalWrite (clockPower, LOW);
 pinMode (clockPower, INPUT);

 // turn off I2C
  TWCR &= ~(bit(TWEN) | bit(TWIE) | bit(TWEA));

  // turn off I2C pull-ups
  digitalWrite (A4, LOW);
  digitalWrite (A5, LOW);


}
```