PILL REMINDER DEVICE

FIELD OF THE INVENTION

This present invention is for new user-friendly electronic pill reminder device that alerts if user forgets to take his/her prescribed medicines at a scheduled time.

BACKGROUND OF THE INVENTION

1. The Problem

It is important to take medicines at prescribed intervals. The medicines can be ineffective or even dangerous if is not taken at proper time . Medicines like high blood pressure pills, birth control pills etc. should not be skipped. What is needed is a simple device to remind patients when to take their medications. Such devices are especially important for the elderly patient with weak memory. Many such devices do exist in the market but such devices do not seem to have become widely available and it is believed that the major shortcomings of such prior devices are complexity and cost of manufacture. There are also smartphone apps available but they are unable to detect whether patient has taken pill or not. So there are needs for a user-friendly, simple, low-cost pill reminding device.

2. The Solution

The present invention is directed to provide such a solution with an electronic reminder device.

SUMMARY OF THE INVENTION

The present invention contains an Arduino based reminder device to remind user to take pill regularly. This device is rectangular box shaped with pill holding compartment inside it. A mechanical contact detection system is developed using spring which is attached to the outside of the device. When user opens the box top lid, mechanical contact happens between two wires in the mechanical contact detection system which in turns provides signal to Arduino indicating the medicine box has been opened i.e. patient has taken pill. The before mentioned logic is used to detect pill intake because it is not possible for someone to take pills without opening the medicine box.

For the first time, user have to turn on the device at a predefined time like 8 PM or 10 PM to set time in device. Because no user interface of time adjust is used in this device. Some fixed time like 8 PM or 10 PM will be hardcoded in RTC timer program. When user turn the device on, the program will set the external RTC chip time to 8 PM or 10 PM as hardcoded in the Arduino code. User can also schedule up to twelve alarms within 24 hours using the DIP switch on the top lid of the device. E.g. User can set alarm at 6 AM, 12 PM and 6 PM at a time. These alarms are flexible up to 30 minutes. So the first alarm will trigger at 6:30 AM if user has not taken medicines within the time period of 5:30 AM – 6:30 AM. The second alarm will trigger at 12:30 PM and the third will at 6:30 PM. The alarm will last for around 5 minutes and will be turned off as soon as the top lid is opened.

There is an additional missed pill indicator LED on the top lid of the device. It will light up after the alarm goes off after full 5 minutes of time duration. Suppose user is not present near the device and comes back later. Then he will be able to know if he has missed any pill by seeing the status of the LED. The LED will go off if user opens the box lid.

This device uses high precision RTC(Real Time clock) DS3231 timer chip to keep track of the time externally since Arduino's inbuilt watchdog timer is inaccurate which is used in Arduino power savings.

This device consumes very low power (in order of micro watt) because of its using Atmega328P standalone chip instead of any complete Arduino board as well as incorporation of Arduino power savings technique like sleep. So it will last for over one year by using 4xAA sized pencil batteries.

A simple battery power indicator circuit is integrated with this device which tells the user whether battery is low or in good condition.

Here is the summary of the uniqueness of present invention compared to the prior art reminder devices :

1. The present invention device can be placed within any rectangular sized box and the box lid will be used as user action input. So it requires no major modification or new design of conventional pill bottle containers. This would save a lot of time and money for engineering and manufacturing costs of a new bottle container.
2. The present invention reminder device is a user-friendly device. Nothing for user to learn, no push-buttons to push to stop the alarm.
3. This device uses missed pill alert concept. So even after user is not able to hear buzzer alarm, he will surely be alerted through the LED indicator that his last pill is missed.
4. The present invention has kept flexible time so that user need not to take medicine at the exact scheduled time. He/she can take pills within 30 minutes before or after the scheduled time. Moreover user can schedule up to 12 alarms at a time.
5. Arduino standalone concept is used here so that it will take very low current compared to complete Arduino boards.
6. Arduino power management code is used so that the total system takes power in micro watt order for the most of the time.
7. Simple battery power indicator circuit is incorporated (design in progress) to notify the low battery power.
8. Very cheap, lightweight and portable compared to other pill reminder devices available currently in market.

CIRCUIT BLOCK DIAGRAM OF PILL REMINDER DEVICE

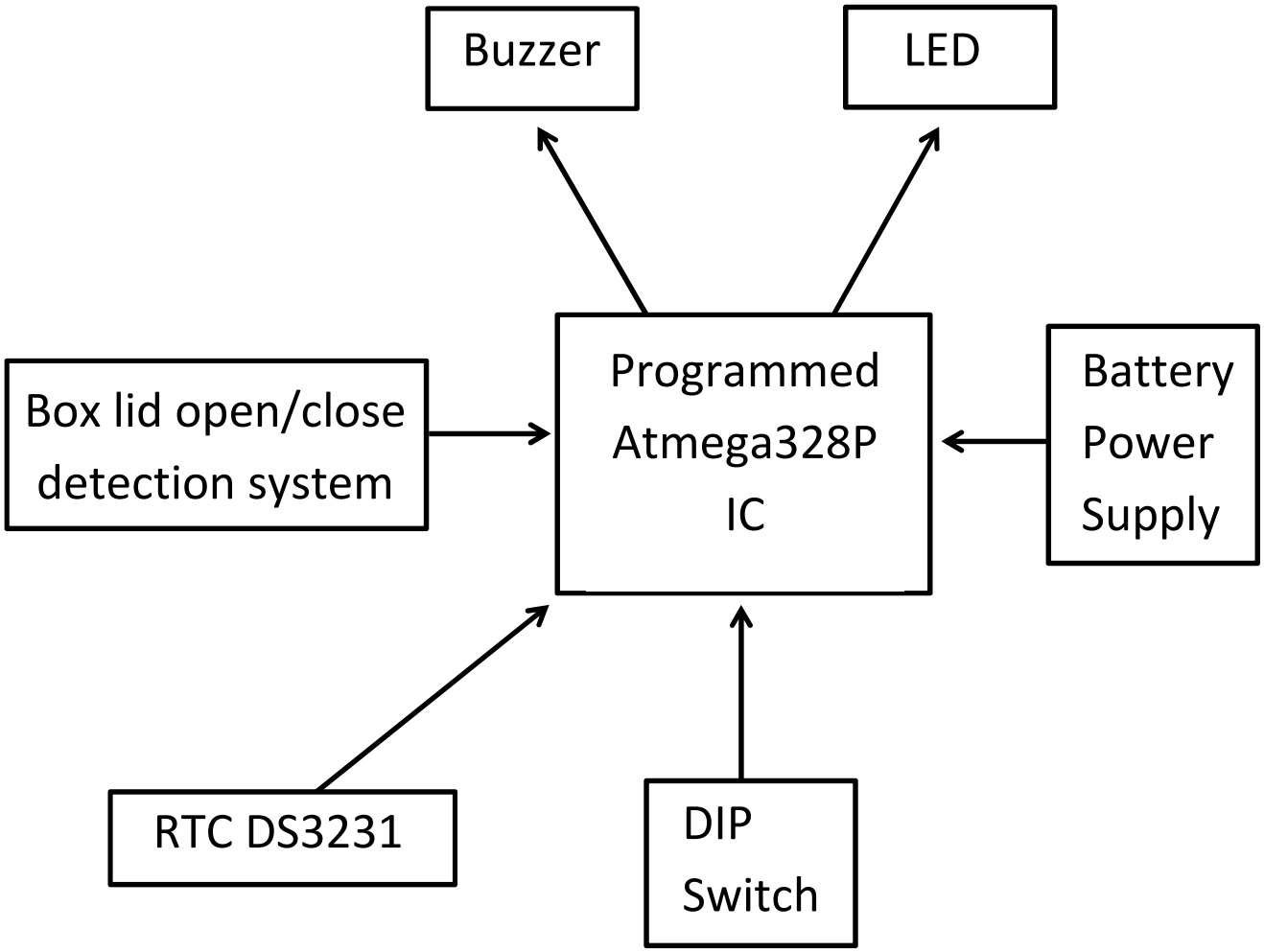


Fig. 1

DEVICE TOP LID OPENED/CLOSED DETECTION SYSTEM

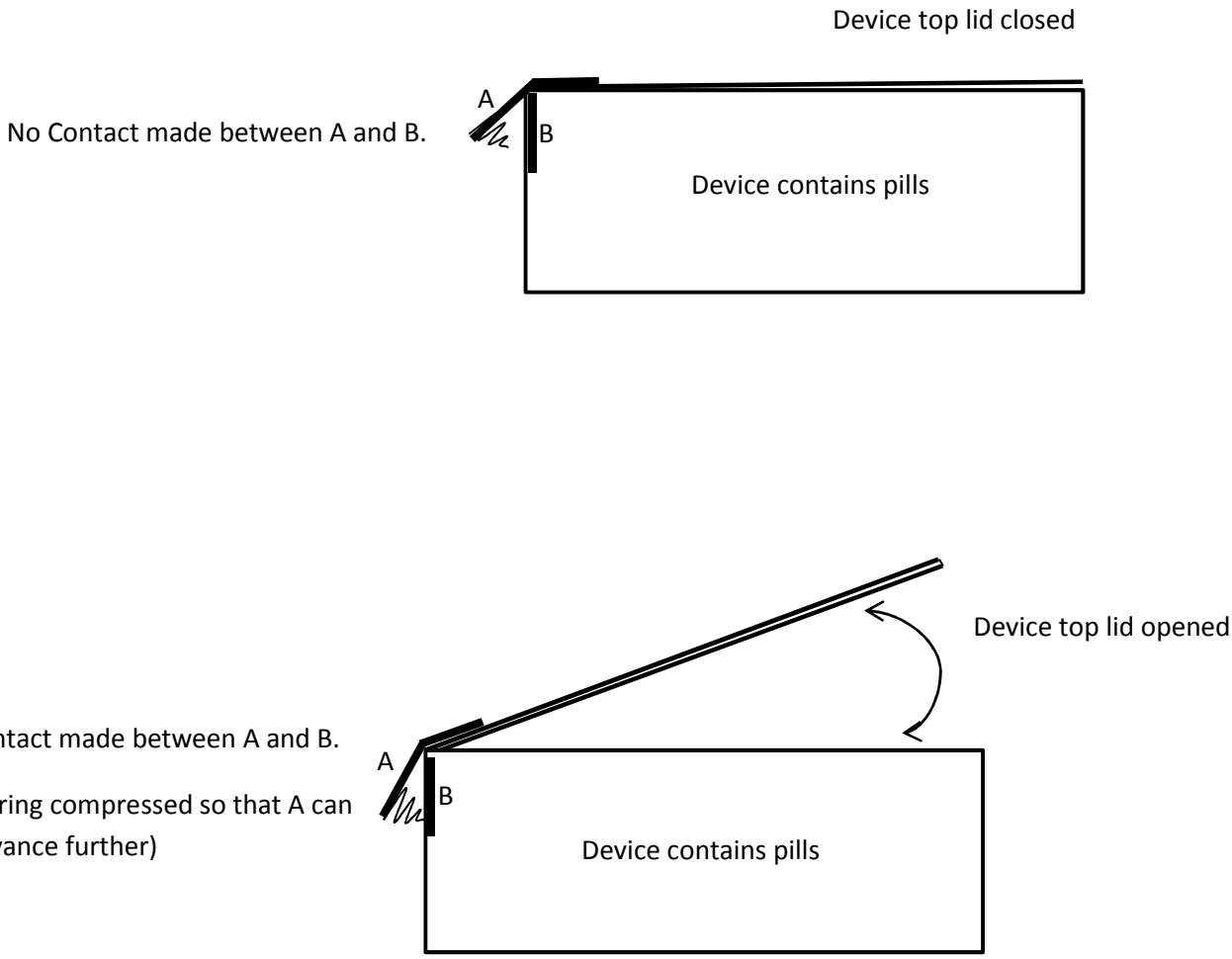


Fig. 2

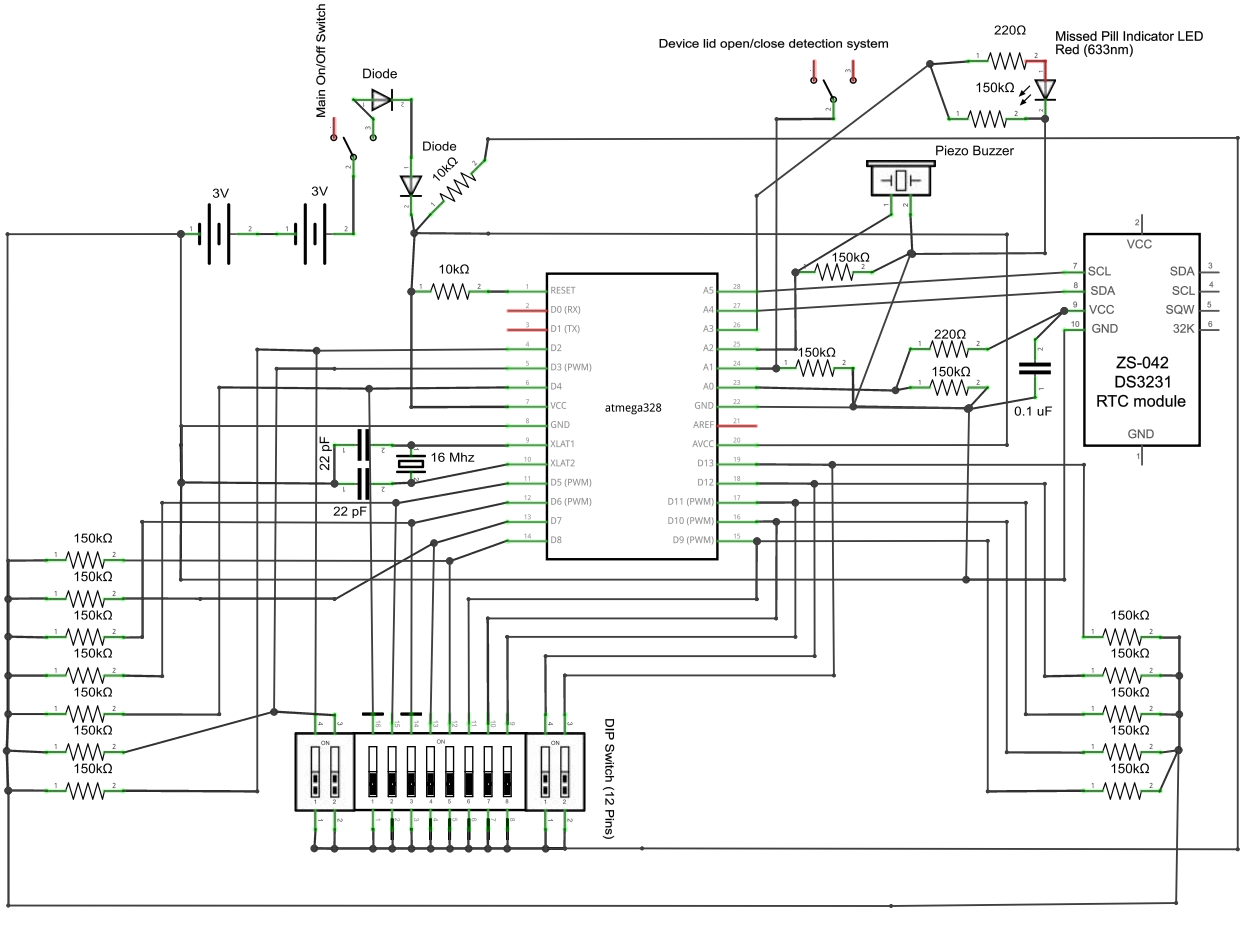
DETAILED DESCRIPTION

The present invention consists of electronic components working as a timer/controller device inside the rectangular pill box.

Fig.1 is the circuit block diagram of the reminder device:

1. ATmega328P IC :
   At first the program code is uploaded to Arduino board through Arduino IDE software and then Atmega328P IC from Arduino board (Uno/ Duemilanove) is detached to make it standalone. The Arduino standalone circuit is also included in the Final schematic Diagram.
2. Battery Power Supply :
   Since this device uses buzzer and missed pill indicator concept so it needs much current to draw. So 4xAA sized battery is recommended which will last for over one year. Two diodes are connected in series with the battery to reduce voltage supplied to the Arduino since IC pins are rated as 5.5 volts max. Moreover, since the battery is placed inside the pill box, non-toxic batteries with proper enclosure must be used to prevent problems, such as battery leakage.
3. DIP Switch :
   12 pins DIP switch or 2 x 6 pins DIP switch is used so user will be able to schedule alarm up to maximum twelve different times in one day (24 hours) with two hours interval. One side of each individual DIP switch will be powered to battery through resistor. And the other pins are connected to Atmega328P IC as input pins. When any switch is turned on, the corresponding IC pin will go HIGH i.e. program will only check for user action (device lid open/close) for the time defined for that pin. At any other time the user actions will be ignored so user will be able to open/close the pill device lid many times to check how much medicine left, inserting new medicine etc. activities.
4. RTC (Real-Time Clock) Module :
   RTC DS3231 is used to keep track of times accurately. I am saving power by turning off clock chips when they aren't needed. To save current, clock is powered from digital pin 14(A0).
5. Device top lid open/close system (Fig. 2): A and B are two metallic strips. A is connected to Arduino digital pin 15(A1) which is defined as “boxOpenPin” in program code and B is connected to battery. When the lid is opened A and B will be touched and A will be HIGH thus indicating user has opened the lid i.e. taken medicines. All other times the lid will be closed and A will be LOW.
6. Buzzer :
   A small DC 3V Piezo buzzer is used to generate Audible Alerts for the user.
7. LED :
   LED is used to visually indicate that user has not taken pill at the scheduled time limit even after the buzzer alarm beeping for 5 minutes. It will be turned on/blink until the box lid is opened by user.
8. Resistors (150K) :
   The final schematic circuit uses several 150K ohm resistors for pull-down operations.

FINAL SCHEMATIC
(Developed by Sourav Paul)



```
//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);


}
```