

```
/*
Example 10.2
On/off timer

tronixstuff.com/tutorials > Chapter 10
based on code by Maurice Ribble
17-4-2008 - http://www.glacialwanderer.com/hobbyrobotics
and John Boxall - http://tronixstuff.com
CC by-sa 2.5 Australia
*/

int timeron = 0; // 0 for off, 1 for on
int alarmh = 12; // timer on hour
int alarmm = 00; // time on minute
int alarmhoff = 0; // timer off hour
int alarmmoff = 0; // timer off minute

#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68
#include <LiquidCrystal.h> // we need this library for the LCD commands
LiquidCrystal lcd(12,11,5,4,3,2);

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val%10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val%16) );
}

// 1) Sets the date and time on the ds1307
// 2) Starts the clock
// 3) Sets hour mode to 24 hour clock

// Assumes you're passing in valid numbers

void setDateDs1307(byte second,          // 0-59
byte minute,          // 0-59
byte hour,            // 1-23
byte dayOfWeek,      // 1-7
byte dayOfMonth,     // 1-28/29/30/31
byte month,           // 1-12
byte year)            // 0-99
{
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.send(0);
  Wire.send(decToBcd(second));    // 0 to bit 7 starts the clock
  Wire.send(decToBcd(minute));
  Wire.send(decToBcd(hour));
  Wire.send(decToBcd(dayOfWeek));
  Wire.send(decToBcd(dayOfMonth));
  Wire.send(decToBcd(month));
  Wire.send(decToBcd(year));
  Wire.send(0x10); // sends 0x10 (hex) 00010000 (binary) to control register - turns on square wave
  Wire.endTransmission();
}

// Gets the date and time from the ds1307
void getDateDs1307(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
```

```

Wire.send(0);
Wire.endTransmission();

Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

// A few of these need masks because certain bits are control bits
*second    = bcdToDec(Wire.receive() & 0x7f);
*minute    = bcdToDec(Wire.receive());
*hour      = bcdToDec(Wire.receive() & 0x3f); // Need to change this if 12 hour am/pm
*dayOfWeek = bcdToDec(Wire.receive());
*dayOfMonth = bcdToDec(Wire.receive());
*month     = bcdToDec(Wire.receive());
*year      = bcdToDec(Wire.receive());
}

int readdial(int rangemax, int dialpin)
// rangemax is the number of values in your range, e.g. if you want 0~9, set rangemax to be '10'
// dialpin is the analog pin number connected to the potentiometer to read
{
  int kv=0;
  int kr=0;
  int kb=0;
  float a=0;
  float rd=0;
  rd=1023/rangemax;
  kb=analogRead(dialpin); // read potentiometer connected to analog pin 1
  a=kb/rd;
  kr=int(a);
  if (kr>rangemax)
  {
    kr=rangemax;
  }
  return kr;
}

void displaymenu()
// display the menu options, selectable by using the knob
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Turn knob slowly");
  lcd.setCursor(0,1);
  lcd.print("to select option");
  while (digitalRead(8)==LOW)
  {
    if (readdial(11,1)==0) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Set hours  ");
    }
    else if (readdial(11,1)==1) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Set minutes ");
    }
    else if (readdial(11,1)==2) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Timer on/off ");
    }
    else if (readdial(11,1)==3) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Set on hour  ");
    }
    else if (readdial(11,1)==4) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Set on minute ");
    }
    else if (readdial(11,1)==5) {

```

```
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Set off hour ");
}
else if (readdial(11,1)==6) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Set off minute ");
}
else if (readdial(11,1)==7) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" exit menu ");
}
delay(100); // stop screen flicker
}

switch(readdial(11,1))
{
case 0:
    msethours();
    break;
case 1:
    msetminutes();
    break;
case 2:
    timeronoff();
    break;
case 3:
    msetalarmh();
    break;
case 4:
    msetalarmm();
    break;
case 5:
    msetalarmhoff();
    break;
case 6:
    msetalarmmmoff();
    break;
}
// if the knob is 7, that is for return to clock display. function will end and return to clock by
default
}

void timeronoff()
// allows user to turn timer on or off
{
    delay(300); // for debounce
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Turn knob slowly");
    lcd.setCursor(0,1);
    lcd.print("to select option");
    while (digitalRead(8)==LOW)
    {
        if (readdial(2,1)==0) {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(" Timer: Off ");
        }
        else if (readdial(2,1)==1) {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(" Timer: On ");
        }
        delay(100); // stop screen flicker
    }
    switch(readdial(2,1))
    {
    case 0:
        timeron=0;
```

```
        break;
    case 1:
        timeron=1;
        break;
    }
}

void msethours()
// allows user to select hour value
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    delay(300); // for debounce
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Select hour:0~23");
    while (digitalRead(8)==LOW)
    {
        lcd.setCursor(8,1);
        lcd.print("    ");
        lcd.setCursor(8,1);
        lcd.print(readdial(24,1));
        delay(100);
    }
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    hour=readdial(24,1);
    setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void msetminutes()
// allows user to select minute value
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    delay(300); // for debounce
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Set minute:0~59");
    while (digitalRead(8)==LOW)
    {
        lcd.setCursor(8,1);
        lcd.print("    ");
        lcd.setCursor(8,1);
        lcd.print(readdial(60,1));
        delay(100);
    }
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    minute=readdial(60,1);
    setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void msetalarmh()
// allows user to select timer on hour value
{
    delay(300); // for debounce
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tmr On hour:0~23");
    while (digitalRead(8)==LOW)
    {
        lcd.setCursor(8,1);
        lcd.print("    ");
        lcd.setCursor(8,1);
        lcd.print(readdial(24,1));
        delay(100);
    }
    alarmh=readdial(24,1);
}

void msetalarmm()
// allows user to select timer on minute value
{
    delay(300); // for debounce
    lcd.clear();
```

```

lcd.setCursor(0,0);
lcd.print("Tmr On mins:0~59");
while (digitalRead(8)==LOW)
{
  lcd.setCursor(8,1);
  lcd.print(" ");
  lcd.setCursor(8,1);
  lcd.print(readdial(60,1));
  delay(100);
}
alarmm=readdial(60,1);
delay(300);
timeronoff(); // force user to turn alarm on or off
}

void msetalarmhoff()
// allows user to select timer off hour value
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TmrOff hour:0~23");
  while (digitalRead(8)==LOW)
  {
    lcd.setCursor(8,1);
    lcd.print(" ");
    lcd.setCursor(8,1);
    lcd.print(readdial(24,1));
    delay(100);
  }
  alarmhoff=readdial(24,1);
}

void msetalarmmoff()
// allows user to select timer off minute value
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TmrOff mins:0~59");
  while (digitalRead(8)==LOW)
  {
    lcd.setCursor(8,1);
    lcd.print(" ");
    lcd.setCursor(8,1);
    lcd.print(readdial(60,1));
    delay(100);
  }
  alarmmoff=readdial(60,1);
  delay(300);
  timeronoff(); // force user to turn alarm on or off
}

void switchon()
// checks to see if it is time to turn on, if so - turns on relay
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  if (alarmh==hour && alarmm==minute)
    // Time to turn on!
  {
    digitalWrite(9, HIGH); // turn output to HIGH
    lcd.clear(); // clear LCD screen
    lcd.setCursor(0,0);
    lcd.print(" * Activated! * ");
    lcd.setCursor(0,1);
    lcd.print("It is now: ");
    lcd.print(alarmh);
    lcd.print(":");
    if (alarmm<10)
    {
      lcd.print("0");
    }
  }
}

```

```
    }
    lcd.print(alarmm);
    lcd.print("h");
    delay(59000); // wait a minute, otherwise this function will loop may times while the real time
minute = alarmm on minute
}
}

void switchoff()
// checks to see if it is time to turn off, if so - turns off relay
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    if (alarmhoff==hour && alarmmoff==minute)
        // Time to turn off :(
        {
            digitalWrite(9, LOW); // turn output to LOW
            lcd.clear(); // clear LCD screen
            lcd.setCursor(0,0);
            lcd.print(" * Deactivated! * ");
            lcd.setCursor(0,1);
            lcd.print("See you again...");
            delay(59000); // wait a minute, otherwise this function will loop may times while the real time
minute = alarmm on minute
        }
}

void setup()
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    Wire.begin();
    second = 0;
    minute = 23;
    hour = 23;
    dayOfWeek = 4;
    dayOfMonth = 19;
    month = 5;
    year = 10;
    // setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year); // only do this once on
first use
    lcd.begin(16, 2); // tells Arduino the LCD dimensions
    lcd.setCursor(0,0);
    lcd.print("tronixstuff.com"); // print text and move cursor to start of next line
    lcd.setCursor(0,1);
    lcd.print("* example 10.2 *");
    delay(5000);
    lcd.clear(); // clear LCD screen
    lcd.setCursor(0,0);
    lcd.print("* Please check *"); // print text and move cursor to start of next line
    lcd.setCursor(0,1);
    lcd.print(" * timer data * ");
    delay(5000);
    lcd.clear(); // clear LCD screen
    pinMode(8, INPUT); // for button
    pinMode(9, OUTPUT); // to relay circuit
}

void loop()
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    lcd.clear(); // clear LCD screen
    lcd.setCursor(0,0);
    lcd.print("0n");
    if (alarmh<10)
    {
        lcd.print("0");
    }
    lcd.print(alarmh);
    if (alarmm<10)
    {
        lcd.print("0");
    }
}
```

```
}
lcd.print(alarmm);
lcd.print("h");
lcd.print(" Now:");
if (hour<10)
{
  lcd.print("0");
}
lcd.print(hour, DEC);
if (minute<10)
{
  lcd.print("0");
}
lcd.print(minute,DEC);
lcd.print("h");
lcd.setCursor(0,1);
lcd.print("Off");
if (alarmhoff<10)
{
  lcd.print("0");
}
lcd.print(alarmhoff);
if (alarmmoff<10)
{
  lcd.print("0");
}
lcd.print(alarmmoff);
lcd.print("h Tmr:");
if (timeron==1)
{
  lcd.print("On");
}
else
  if (timeron==0)
  {
    lcd.print("Off");
  }
if (digitalRead(8)==HIGH)
  // has the user pressed the button? If so, display the menu
{
  delay(200); // for debounce
  displaymenu();
}
if (timeron==1)
{
  switchon(); // should I turn on?
  switchoff(); // or should I turn off?
}
delay(200); // to stop screen flicker
}
```