```
/*
 Example 9.3
 alarm clock!

 tronixstuff.com/tutorials  > Chapter 9
 based on code by Maurice Ribble
 17-4-2008 - http://www.glacialwanderer.com/hobbyrobotics
 */

int t1224 = 0; // set later on in function mset1224()
int alarmon = 0; // 0 for off, 1 for on
int alarmh = 12; // alarm hour
int alarmm = 00; // alarm 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(" 12 or 24h time?");
    }
    else if (readdial(11,1)==3) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Set day of month");
    }
    else if (readdial(11,1)==4) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("   Set month    ");
    }
    else if (readdial(11,1)==5) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("   Set year     ");
```

```
    }
    else if (readdial(11,1)==6) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(" Set Mon~Sun    ");
    }
    else if (readdial(11,1)==7) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  Alarm on/off  ");
    }
    else if (readdial(11,1)==8) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(" Set alarm hour");
    }
    else if (readdial(11,1)==9) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(" Set alarm min. ");
    }
    else if (readdial(11,1)==10) {
      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:
    mset1224();
    break;
  case 3:
    msetday();
    break;
  case 4:
    msetmonth();
    break;
  case 5:
    msetyear();
    break;
  case 6:
    msetdow();
    break;
  case 7:
    alarmonoff();
    break;
  case 8:
    msetalarmh();
    break;
  case 9:
    msetalarmm();
    break;
  }
  // if the knob is 10, that is for return to clock display. function will end and return to clock by
default
}

void mset1224()
//  allows user to select between displaying 12 and 24 hour time
{
  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("  12-hour time  ");
    }
    else if (readdial(2,1)==1) {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("  24-hour time  ");
    }
    delay(100); // stop screen flicker
  }
  switch(readdial(2,1))
  {
  case 0:
    t1224=0;
    break;
  case 1:
    t1224=1;
    break;
  }
}

void alarmonoff()
//  allows user to turn alarm on or off
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Current: ");
  lcd.print(alarmh);
  lcd.print(":");
  if (alarmm<10)
  {
    lcd.print("0");
  }
  lcd.print(alarmm);
  lcd.print("h");

  while (digitalRead(8)==LOW)
  {
    if      (readdial(2,1)==0) {
      lcd.setCursor(0,1);
      lcd.print("   Alarm OFF    ");
    }
    else if (readdial(2,1)==1) {
      lcd.setCursor(0,1);
      lcd.print("   Alarm ON     ");
    }
    delay(100); // stop screen flicker
  }
  switch(readdial(2,1))
  {
  case 0:
    alarmon=0;
    break;
  case 1:
    alarmon=1;
    break;
  }
  delay(300);
}

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 msetalarmh()
//  allows user to select alarm hour value
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Alarm 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 alarm minute value
{
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Alarm 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);
  alarmonoff(); // force user to turn alarm on or off
}

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 msetday()
//  allows user to select day of the month
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("  Set day: 1~31 ");
  while (digitalRead(8)==LOW)
  {
    lcd.setCursor(8,1);
    lcd.print("     ");
    lcd.setCursor(8,1);
    lcd.print(readdial(31,1));
    delay(100);
  }
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  dayOfMonth=readdial(31,1);
  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void msetmonth()
//  allows user to select day of the month
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Set month: 1~12");
  while (digitalRead(8)==LOW)
  {
    lcd.setCursor(8,1);
    lcd.print("     ");
    lcd.setCursor(8,1);
    lcd.print(readdial(12,1));
    delay(100);
  }
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  month=readdial(12,1);
  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void msetyear()
//  allows user to select year
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  delay(300); // for debounce
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Set year:00~99 ");
  while (digitalRead(8)==LOW)
  {
    lcd.setCursor(8,1);
    lcd.print("     ");
    lcd.setCursor(8,1);
    lcd.print(readdial(100,1));
    delay(100);
  }
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  year=readdial(100,1);
  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void msetdow()
//  allows user to select day of week (Monday~Sunday)
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
```

```
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(" Choose the day ");
  delay(300); // for debounce
  while (digitalRead(8)==LOW)
  {
    if      (readdial(7,1)==0) {
      lcd.setCursor(0,1);
      lcd.print("   Sunday       ");
    }
    else if (readdial(7,1)==1) {
      lcd.setCursor(0,1);
      lcd.print("  Monday        ");
    }
    else if (readdial(7,1)==2) {
      lcd.setCursor(0,1);
      lcd.print(" Tuesday        ");
    }
    else if (readdial(7,1)==3) {
      lcd.setCursor(0,1);
      lcd.print("  Wednesday     ");
    }
    else if (readdial(7,1)==4) {
      lcd.setCursor(0,1);
      lcd.print("   Thursday     ");
    }
    else if (readdial(7,1)==5) {
      lcd.setCursor(0,1);
      lcd.print("   Friday       ");
    }
    else if (readdial(7,1)==6) {
      lcd.setCursor(0,1);
      lcd.print(" Saturday       ");
    }
    delay(100); // stop screen flicker
  }
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  dayOfWeek=readdial(7,1)+1;
  setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void checkalarm()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  if (alarmh==hour && alarmm==minute)
    // alarm time!
  {
    while (digitalRead(8)==LOW) // do noisy or blinky things until user presses the button
    {
      lcd.clear(); // clear LCD screen
      lcd.setCursor(0,0);
      lcd.print(" * ALARM TIME * ");
      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(300);
      // insert functio to turn on/off piezo buzzer, etc
    }
    delay(500);
    alarmonoff();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Careful: don't");
    lcd.setCursor(0,1);
```

```
    lcd.print("go back to sleep");
    delay(60000); // wait a 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);

  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("* exercise 9.3 *");
  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(" * alarm data * ");
  delay(5000);
  lcd.clear(); // clear LCD screen
  pinMode(8, INPUT);
}

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("    ");
  if (t1224==0)
  {
    if (hour==0)
    {
      lcd.print("12");
    }
    else
      if (hour>0 && hour<13)
      {
        lcd.print(hour, DEC);
      }
      else
        if (hour>=13)
      {
        lcd.print((hour-12), DEC);
      }
  }
  else if (t1224==1)
  {
    lcd.print(hour, DEC);
  }
  lcd.print(":");
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  lcd.print(":");
  if (second<10)
  {
    lcd.print("0");
```

```
  }
  lcd.print(second, DEC);
  if (t1224==1)
  {
    lcd.print("h");
  }
  else if (t1224==0)
  {
    if ((23-hour)>11)
    {
      lcd.print(" am");
    }
    if ((23-hour)<12)
    {
      lcd.print(" pm");
    }
  }
  if (alarmon==1)
  {
    lcd.print(" *");// if alarm on, display asterix
  }

  lcd.setCursor(0,1);
  lcd.print(" ");
  switch(dayOfWeek){
  case 1:
    lcd.print("Sun");
    break;
  case 2:
    lcd.print("Mon");
    break;
  case 3:
    lcd.print("Tue");
    break;
  case 4:
    lcd.print("Wed");
    break;
  case 5:
    lcd.print("Thu");
    break;
  case 6:
    lcd.print("Fri");
    break;
  case 7:
    lcd.print("Sat");
    break;
  }
  lcd.print("  ");
  lcd.print(dayOfMonth, DEC);
  lcd.print("/");
  lcd.print(month, DEC);
  lcd.print("/20");
  lcd.print(year, DEC);

  if (digitalRead(8)==HIGH)
    // has the user pressed the button? If so, display the menu
  {
    delay(200); // for debounce
    displaymenu();
  }
  if (alarmon==1)
  {
    checkalarm(); // should the alarm activate?
  }
  delay(200); // to stop screen flicker
}
```