

```
/*
Example 12.1
Create a digital LED display clock
http://tronixstuff.com/tutorials > Chapter Twelve
based on code by Maurice Ribble
17-4-2008 - http://www.glacialwanderer.com/hobbyrobotics
Uses code from Sparkfun website, no author or licensing information
from: http://bit.ly/a90H2x
*/

#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68

int dpInEncoderA=2;
int dpInEncoderB=3;
int dpInEncoderPress=4;
int latchpin = 9; // connect to pin 12 on the '595
int clockpin = 7; // connect to pin 11 on the '595
int datapin = 10; // connect to pin 14 on the '595
int lhd = 0;
int mhd = 0;
int mhd2 = 0;
int rhd = 0;
float a = 0;
int b = 0;
int c = 0;
float d = 0;
int leadingzero = 1; // 0 for no leading zeroes, 1 for leading zeroes
int rnum = 0;
int zzz=0;
byte esh = 12;
byte esm = 30;
int sett=0;

int segdisp[10] = {
  123, 72,103,79,92,31,63,74,127,95}; // base 10 equivalents for digits 0~9
int posdisp[4] = {
  1,2,4,8}; // base 10 equivalents to close cathodes on display 0~3 on module

static void _ResetPins()
{
  // Rotary encoder input lines
  // Configure as input, turn on pullup resistors
  pinMode(dpInEncoderA, INPUT);
  digitalWrite(dpInEncoderA, HIGH);
  pinMode(dpInEncoderB, INPUT);
  digitalWrite(dpInEncoderB, HIGH);
  pinMode(dpInEncoderPress, INPUT);
  digitalWrite(dpInEncoderPress, HIGH);
}

void _lowlevel_ReadEncoder(int &rotate, int& press)
{
  rotate = (digitalRead(dpInEncoderB) * 2) + digitalRead(dpInEncoderA);
  press = digitalRead(dpInEncoderPress);
}

void operate()
{
  int Position, Press;
  int Position2, Press2;
  _ResetPins();
  _lowlevel_ReadEncoder(Position, Press);
  while (!Serial.available())
  {
    do
      // show the clock until user presses the button
    {
      _lowlevel_ReadEncoder(Position2, Press2);
      showtime();
    }
  }
}
```

```

while (Press2 == Press);

if (Press2 != Press) // oh! user pressed button...
{
    ReadEncoder2();
}
Press = Press2;
}
}

void ReadEncoder2()
// function to allow user to set the time - rotate encoder one way to adjust minutes, the other to
adjust hours - press button to loc
in new time and return to normal clock display
{
    int Position, Press, Position2, Press2;
    _ResetPins();
    _lowlevel_ReadEncoder(Position, Press);
    sett=0;
    do
    {
        do
        {
            _lowlevel_ReadEncoder(Position2, Press2);
            zzz = (esh*100)+esm;
            displaynumber(zzz,50);
        }
        while ((Position2 == Position) && (Press2 == Press));
        if (Position2 != Position)
        {
            int isFwd = ((Position == 0) && (Position2 == 1)) ||
                ((Position == 1) && (Position2 == 3)) ||
                ((Position == 3) && (Position2 == 2)) ||
                ((Position == 2) && (Position2 == 0));
            if (isFwd == 0)
            {
                esh++;
                if (esh>23) {
                    esh=0;
                }
            }
            if (isFwd == 1)
            {
                esm++;
                if (esm>59) {
                    esm=0;
                }
            }
        }
    }
    if (Press2 != Press)
    {
        // user pressed button, lock in esh and esm as current time
        setDateDs1307(1, esm, esh, 1, 1, 1, 00);
        sett=1;
    }
    Position = Position2;
    Press = Press2;
    if (sett==1) {
        break;
    }
}
while (!Serial.available());
}

// 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 dsl307
// 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();
}

void cleardisplay()
// turns off all segments of all digits
{
  for (int aa=0; aa<4; aa++)
  {
    digitalWrite(latchpin, LOW);
    shiftOut(datapin, clockpin, MSBFIRST, posdisp[aa]); // sets the digit to address
    shiftOut(datapin, clockpin, MSBFIRST, 0); // clears the digit
    digitalWrite(latchpin, HIGH);
  }
}

void onedigitnumber(int subject)
// displays a one-digit number on the display module with leading zeroes
{
  cleardisplay();
  if (leadingzero==1)
  {
    digitdisplay(0,0);
    digitdisplay(0,1);
    digitdisplay(0,2);
  }
  digitdisplay(subject,3);
}

void twodigitnumber(int subject)
// displays a two-digit number on the display module with leading zeroes
{
  cleardisplay();
  rhd = subject % 10;
  a = subject/10;
  lhd = int(a);
  if (leadingzero==1)
  {
    digitdisplay(0,0);
    digitdisplay(0,1);
  }
  digitdisplay(lhd,2);
  digitdisplay(rhd,3);
}
```

```

void threedigitnumber(int subject)
// displays a three-digit number on the display module with leading zeroes
{
  cleardisplay();
  a = subject/100;
  lhd = int(a);
  a = subject/10;
  b = int(a);
  mhd = b % 10;
  b=subject%100;
  rhd=b%10;
  if (leadingzero==1)
  {
    digitdisplay(0,0);
  }
  digitdisplay(lhd,1);
  digitdisplay(mhd,2);
  digitdisplay(rhd,3);
}

void displaynumber(int rawnumber, int cycles)
// takes an integer and displays it on our 4-digit LED display module
{
  for (int q=1; q<=cycles; q++)
  {
    if (rawnumber>=0 && rawnumber<10)
    {
      onedigitnumber(rawnumber);
    }
    else if (rawnumber>=10 && rawnumber<100)
    {
      twodigitnumber(rawnumber);
    }
    else if (rawnumber>=100 && rawnumber<1000)
    {
      threedigitnumber(rawnumber);
    }
    else if (rawnumber>=1000)
    {
      fourdigitnumber(rawnumber);
    }
  }
}

void fourdigitnumber(int subject)
// displays a four-digit number on the display module with leading zeros
{
  cleardisplay();
  a = subject/1000;
  lhd = int(a);
  b=lhd*1000;
  c=subject-b;
  a = c/100;
  mhd = int(a);
  a = c/10;
  b = int(a);
  mhd2 = b % 10;
  b=subject%1000;
  c=b%100;
  rhd=c%10;
  digitdisplay(lhd,0);
  digitdisplay(mhd,1);
  digitdisplay(mhd2,2);
  digitdisplay(rhd,3);
}

void digitdisplay(int digit, int location)
// displays "digit" on display "location" 0~3
{
  digitalWrite(latchpin, LOW);
  shiftOut(datapin, clockpin, MSBFIRST, posdisp[location]); // sets the digit to address
}

```

```
    shiftOut(datapin, clockpin, MSBFIRST, segdisp[digit]); // clears the digit
    digitalWrite(latchpin, HIGH);
}

// 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());
}

void showtime()
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    if (bcdToDec(hour)<1)
    {
        zzz=minute;
    }
    else if (bcdToDec(hour)>=1)
    {
        zzz=hour*100;
        zzz=zzz+minute;
    }
    displaynumber(zzz,100);
}

void setup()
{
    // configure the pins
    _ResetPins();
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
    pinMode(latchpin, OUTPUT);
    pinMode(clockpin, OUTPUT);
    pinMode(datapin, OUTPUT);
    Wire.begin();
    second = 0;
    minute = 45;
    hour = 1;
    dayOfWeek = 5;
    dayOfMonth = 20;
    month = 5;
    year = 10;
    //setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void loop()
{
    operate();
}
```

