

```

/*
Example 12.2
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

*/

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

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;
int segdisp[10] = {
  125,9,103,79,27,94,126,13,127,95}; // base 10 equivalents for digits 0-9
int posdisp[4] = {
  1,2,4,8}; // base 10 equivalents to close anodes on display 0-3 on module

// 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();
}

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 setup()
{
    byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

```

```
pinMode(latchpin, OUTPUT);
pinMode(clockpin, OUTPUT);
pinMode(datapin, OUTPUT);
Wire.begin();
Serial.begin(9600);

// Change these values to what you want to set your clock to.
// You probably only want to set your clock once and then remove
// the setDateDs1307 call.

second = 0;
minute = 37;
hour = 0;
dayOfWeek = 5;
dayOfMonth = 20;
month = 5;
year = 10;
// setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);
}

void loop()
{

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,1000);
}
```