

```

/*
Exercise 5.1
Using 2 7-segment displays with the 74HC595 shift registers
CC by-sa-nc 3.0
http://tronixstuff.wordpress.com
*/

int latchpin = 8; // connect to pin 12 on the '595
int clockpin = 12; // connect to pin 11 on the '595
int datapin = 11; // connect to pin 14 on the '595
float b = 0;
int c = 0;
float d = 0;
int e = 0;
int speed = 150; // used to control speed of counting

int segdisp[10] = {
    63,6,91,79,102,109,125,7,127,111 };

void setup()
{
    pinMode(latchpin, OUTPUT);
    pinMode(clockpin, OUTPUT);
    pinMode(datapin, OUTPUT);
}

void loop()
{
    // Count up
    for (int z=0; z<100; z++)
    {
        digitalWrite(latchpin, LOW);
        shiftOut(datapin, clockpin, MSBFIRST, 0); // clears the right display
        shiftOut(datapin, clockpin, MSBFIRST, 0); // clears the left display
        digitalWrite(latchpin, HIGH);

        if (z<10)
        {
            digitalWrite(latchpin, LOW);
            shiftOut(datapin, clockpin, MSBFIRST, segdisp[z]); // sends the digit down the serial path
            shiftOut(datapin, clockpin, MSBFIRST, 0); // sends a blank down the serial path to push the
digit to the right
            digitalWrite(latchpin, HIGH);
        }
        else if (z>=10)
        {
            d=z%10; // find the remainder of dividing z by 10, this will be the right-hand digit
            c=int(d); // make it an integer, c is the right hand digit
            b=z/10; // divide z by 10 - the whole number value will be the left-hand digit
            e = int(b); // e is the left hand digit
            digitalWrite(latchpin, LOW); // send the digits down to the shift registers!
            shiftOut(datapin, clockpin, MSBFIRST, segdisp[c]);
            shiftOut(datapin, clockpin, MSBFIRST, segdisp[e]);
            digitalWrite(latchpin, HIGH);
        }
        delay(speed);
    }
    delay(2000);

    // Count down
    for (int z=99; z>=0; z--)
    {
        digitalWrite(latchpin, LOW);
        shiftOut(datapin, clockpin, MSBFIRST, 0); // clears the right display
        shiftOut(datapin, clockpin, MSBFIRST, 0); // clears the left display
        digitalWrite(latchpin, HIGH);

        if (z<10)
        {
            digitalWrite(latchpin, LOW);
            shiftOut(datapin, clockpin, MSBFIRST, segdisp[z]); // sends the digit down the serial path
            shiftOut(datapin, clockpin, MSBFIRST, 0); // sends a blank down the serial path to push the
digit to the right

```

```
    digitalWrite(latchpin, HIGH);
}
else if (z>=10)
{
    d=z%10; // find the remainder of dividing z by 10, this will be the right-hand digit
    c=int(d); // make it an integer, c is the right hand digit
    b=z/10; // divide z by 10 - the whole number value will be the left-hand digit
    e = int(b); // e is the left hand digit
    digitalWrite(latchpin, LOW); // send the digits down to the shift registers!
    shiftOut(datapin, clockpin, MSBFIRST, segdisp[c]);
    shiftOut(datapin, clockpin, MSBFIRST, segdisp[e]);
    digitalWrite(latchpin, HIGH);
}
delay(speed);
}
delay(2000);
}
```