

```
/*
```

```
See:  
http://www.bristolwatch.com/ele2/arduino\_MM5451.htm
```

```
by Lewis Loflin  
lewis@bvu.net
```

```
# for driving common anode displays  
# brightness control eliminates need for resistors  
# Vcc 4.75 - 11V  
  
# 7-segment common anode display connected bits 1-8  
# 8 LEDs bits 9-16 common anodes
```

```
    msb  
      PGFEDCBA  
0b00111111 - 0  
0b00000110 - 1  
0b01011011 - 2  
0b01001111 - 3  
0b01100110 - 4  
0b01101101 - 5  
0b01111101 - 6  
0b00000111 - 7  
0b01111111 - 8  
0b01100111 - 9
```

```
*/  
  
#define CLK 8 // to MM5451 pin 21  
#define dataBit 9 // to MM5451 pin 22
```

```
// 7-segment display code array  
byte segCode[] = {  
    0b00111111, 0b00000110, 0b01011011,  
    0b01001111, 0b01100110, 0b01101101,  
    0b01111101, 0b00000111, 0b01111111,  
    0b01100111 };
```

```
void setup() {  
    pinMode(CLK, OUTPUT);  
    pinMode(dataBit, OUTPUT);  
    digitalWrite(CLK, 0);  
    digitalWrite(dataBit, 0);  
}
```

```
// the loop routine runs over and over again forever:  
void loop() {
```

```
    for (int myCount = 0; myCount <= 255; myCount++)    {  
        // start bit  
        digitalWrite(dataBit, 1);  
        pulseCLK();  
        // the "+ 0x80" turns on DP  
        ssrWriteLSB(segCode[myCount % 10]); // MOD myCount = 0-9  
        ssrWriteMSB(myCount); // eight LEDs binary count 0-255  
        zeroWrite(21); // pad remaining bits with 0s  
        delay(500);  
    }
```

```
} // end loop
```

```
void pulseCLK()    {  
    digitalWrite(CLK, 1);  
    digitalWrite(CLK, 0);  
}
```

```

// LSB out first!
void ssrWriteLSB(byte value)  {
    for(int x =0; x < 8; x++)  {
        byte temp = value & 0x01;
        if (temp == 0x01)    digitalWrite(dataBit, 1); // data bit HIGH
        else digitalWrite(dataBit, 0); // data bit LOW
        pulseCLK();
        value = value >> 0x01; // shift left
    }
}

// LSB out first!
void ssrWriteMSB(byte value)  {
    for(int x =0; x < 8; x++)  {
        byte temp = value & 0x80;
        if (temp == 0x80)    digitalWrite(dataBit, 1); // data bit HIGH
        else digitalWrite(dataBit, 0); // data bit LOW
        pulseCLK();
        value = value << 0x01; // shift left
    }
}

void zeroWrite(byte num1)  {
    digitalWrite(dataBit, 0); // data bit LOW
    for(int x = 0; x < num1; x++) pulseCLK();
}

```