

```

#include <multiCameraIrControl.h>
#include <digitalWriteFast.h>
Canon D5(12);
//Nikon D5000(12);

unsigned long start_time = 0;
const int status_led_pin = 12;
const int remote_trigger_pin = 13;
const int pushbutton_pin = 10;

int prev_digits[3] = {
    -1,-1,-1};

//Digit 1 is RHS, digit 3 is LHS
const int digit_pins[3][8] = {
    {
        A1,A0,5,3,2,A2,A3,6
    },
    {
        A5,A4,9,14,15,A6,A7,8
    },
    {
        A11,A10,17,19,20,A12,A13,16
    }
};

const int digit_commonpins[] = {
    4,7,18}; //Common Anode pins.

/*
const int digit2_pins[] = {
    A5,A4,9,14,15,A6,A7,8}; //A,B,C,D,E,F,G,DP
const int digit3_pins[] = {
    A11,A10,17,19,20,A12,A13,16}; //A,B,C,D,E,F,G,DP
*/

//Borrowed some 7 segment code from here:
//http://www.hacktronics.com/Tutorials/arduino-and-7-segment-
//led.html
const byte seven_seg_digits[10][7] = {
    {
        1,1,1,1,1,1,0
    }, // = 0
    {
        0,1,1,0,0,0,0
    }, // = 1
    {
        1,1,0,1,1,0,1
    }, // = 2
    {
        1,1,1,1,0,0,1
    }, // = 3
    {
        0,1,1,0,0,1,1
    }
};

```

```

, // = 4
{
    1,0,1,1,0,1,1
, // = 5
{
    1,0,1,1,1,1,1
, // = 6
{
    1,1,1,0,0,0,0
, // = 7
{
    1,1,1,1,1,1,1
, // = 8
{
    1,1,1,0,0,1,1
} // = 9
};

```

```

void setup(){
    pinMode(status_led_pin,OUTPUT);
    digitalWrite(status_led_pin,HIGH);
    pinMode(pushbutton_pin,INPUT);
    digitalWrite(pushbutton_pin,HIGH); //Pullups.
    pinMode(remote_trigger_pin,OUTPUT);
    digitalWrite(remote_trigger_pin,LOW);
    //Set up commons:
    for (int i = 0; i< 3; i++){
        pinMode(digit_commonpins[i],OUTPUT);
        digitalWrite(digit_commonpins[i],LOW);
    }

    //Set up digits:
    for (int j = 0; j<3; j++){
        for (int i = 0; i< 8; i++){
            pinMode(digit_pins[j][i],OUTPUT);
            digitalWrite(digit_pins[j][i],HIGH);
        }
    }
}

```

```

void loop(){
    //test_pins();
    //test_digits();
    digitalWriteFast(remote_trigger_pin,LOW);
    set_digit(0,0);
    set_digit(1,0);
    set_digit(2,0);
    while (digitalRead(pushbutton_pin)==HIGH){
        //Do nothing.
    }
    for (int i = 3; i>0 ; i--){
        set_digit(0,i);
        set_digit(1,i);
        set_digit(2,i);
        delay(1000);
    }
}

```

```

}

D5.shutterNow();
//D5000.shutterNow();

digitalWriteFast(remote_trigger_pin,HIGH);//Trigger pin.
start_time = millis();

while((millis()-start_time)<1000){
    //Loop forever:
    count_ms();
}
digitalWriteFast(remote_trigger_pin,LOW);//Trigger pin.
digits_off();
}

boolean led_status = false;
const int offset = 0;
inline void count_ms(){
    //Take the current time and display it on the screen.
    int current_time = ((millis()-start_time) % 1000)+offset;
    set_digit(0,current_time % 10);
    set_digit(1,(current_time % 100)/10);
    set_digit(2,(current_time % 1000)/100);
    led_status = not led_status;
    digitalWriteFast(status_led_pin,led_status);
}

void test_pins(){
    int delay_time = 10;

    //Test all the LEDs work, one at a time. :
    for (int j= 0; j<3; j++){
        digitalWrite(digit_commonpins[j],HIGH);
        for (int i = 0; i< 8; i++){
            digitalWrite(digit_pins[j][i],LOW); //Segment on.
            delay(delay_time);
            digitalWrite(digit_pins[j][i],HIGH);//Segment off.
        }
    }
}

void test_digits(){
    for (int i = 0; i<10; i++){
        for (int j = 0; j<3; j++){
            set_digit(j,i);
        }
        delay(1000);
    }
}

```

```

inline void set_digit(int digit_select, byte digit_val) {
    //Change the digit. Blank first to avoid corruption in photo.
    if (prev_digits[digit_select]==digit_val){
        return; //currently updated, don't bother changing.
    }
    prev_digits[digit_select]=digit_val;

    boolean segment_a = not seven_seg_digits[digit_val][0];
    boolean segment_b = not seven_seg_digits[digit_val][1];
    boolean segment_c = not seven_seg_digits[digit_val][2];
    boolean segment_d = not seven_seg_digits[digit_val][3];
    boolean segment_e = not seven_seg_digits[digit_val][4];
    boolean segment_f = not seven_seg_digits[digit_val][5];
    boolean segment_g = not seven_seg_digits[digit_val][6];

    //Update the digit.
    switch (digit_select){
    case 0:
        //A1,A0,5,3,2,A2,A3,6
        digitalWriteFast(digit_commonpins[0],LOW); //Digit off.
        digitalWriteFast(A1, segment_a);
        digitalWriteFast(A0, segment_b);
        digitalWriteFast(5, segment_c);
        digitalWriteFast(3, segment_d);
        digitalWriteFast(2, segment_e);
        digitalWriteFast(A2, segment_f);
        digitalWriteFast(A3, segment_g);
        digitalWriteFast(digit_commonpins[0],HIGH); //Digit on.
        break;
    case 1:
        //A5,A4,9,14,15,A6,A7,8
        digitalWriteFast(digit_commonpins[1],LOW); //Digit off.
        digitalWriteFast(A5, segment_a);
        digitalWriteFast(A4, segment_b);
        digitalWriteFast(9, segment_c);
        digitalWriteFast(14, segment_d);
        digitalWriteFast(15, segment_e);
        digitalWriteFast(A6, segment_f);
        digitalWriteFast(A7, segment_g);
        digitalWriteFast(digit_commonpins[1],HIGH); //Digit on.
        break;
    case 2:
        //A11,A10,17,19,20,A12,A13,16
        digitalWriteFast(digit_commonpins[2],LOW); //Digit off.
        digitalWriteFast(A11, segment_a);
        digitalWriteFast(A10, segment_b);
        digitalWriteFast(17, segment_c);
        digitalWriteFast(19, segment_d);
        digitalWriteFast(20, segment_e);
        digitalWriteFast(A12, segment_f);
        digitalWriteFast(A13, segment_g);
        digitalWriteFast(digit_commonpins[2],HIGH); //Digit on.
        break;
    }
}

```

```
    }  
}
```

```
inline void digits_off(){  
    digitalWriteFast(digit_commonpins[0],HIGH);  
    digitalWriteFast(digit_commonpins[1],HIGH);  
    digitalWriteFast(digit_commonpins[2],HIGH);  
}
```