

```
//Firework control panel using Arduino MEGA 2560
```

```
//Created by Jonathan Bush
```

```
//Last Updated: 6/23/2015
```

```
#include <Adafruit_NeoPixel.h>
```

```
#define NUM_PIXELS 34
```

```
#define NEO_DATA_PIN 22//Data out to Neopixels
```

```
//Define hardware launch buttons
```

```
#define LAUNCH_BTN1 30
```

```
#define LAUNCH_BTN2 28
```

```
#define LAUNCH_BTN3 26
```

```
#define LAUNCH_BTN4 24
```

```
#define LAUNCH_BTN5 36
```

```
#define LAUNCH_BTN6 38
```

```
#define LAUNCH_BTN7 34
```

```
#define LAUNCH_BTN8 32
```

```
//Define remote control module inputs
```

```
#define REMOTE_A 12
```

```
#define REMOTE_B 10
```

```
#define REMOTE_C 13
```

```
#define REMOTE_D 11
```

```
//Define LEDs for hardware launch buttons
```

```
#define LAUNCH_BTN_LED1 9
```

```
#define LAUNCH_BTN_LED2 8
```

```
#define LAUNCH_BTN_LED3 7
```

```
#define LAUNCH_BTN_LED4 6
```

```
#define LAUNCH_BTN_LED5 5
```

```
#define LAUNCH_BTN_LED6 4
```

```
#define LAUNCH_BTN_LED7 3
```

```
#define LAUNCH_BTN_LED8 2
```

```
//Define firing relays
```

```
#define RELAY1 37
```

```
#define RELAY2 35
```

```
#define RELAY3 33
```

```
#define RELAY4 31
```

```
#define RELAY5 29
```

```
#define RELAY6 27
```

```
#define RELAY7 25
```

```
#define RELAY8 23
//Define Node sensing inputs
#define SENSE_NODE8 53
#define SENSE_NODE7 51
#define SENSE_NODE6 49
#define SENSE_NODE5 47
#define SENSE_NODE4 45
#define SENSE_NODE3 43
#define SENSE_NODE2 41
#define SENSE_NODE1 39
//Define Key selection inputs
#define KEY_SELECT_DISARM 46
#define KEY_SELECT_TEST 48
#define KEY_SELECT_FIRE 50
#define KEY_SELECT_FIREALL 52
//Define pin that potentiometer is attached to
#define POTPIN 0
//Define Launch Nodes
int Node1 =HIGH;
int Node2 =HIGH;
int Node3 =HIGH;
int Node4 =HIGH;
int Node5 =HIGH;
int Node6 =HIGH;
int Node7 =HIGH;
int Node8 =HIGH;
//Launch status integers
int Launch1 =LOW;
int Launch2 =LOW;
int Launch3 =LOW;
int Launch4 =LOW;
int Launch5 =LOW;
int Launch6 =LOW;
int Launch7 =LOW;
int Launch8 =LOW;
//Remote status integers
int Remote1 =LOW;
int Remote2 =LOW;
```

```
int Remote3 =LOW;
int Remote4 =LOW;
//other integers
int ARM_Mode = 0;
int BRIGHTNESS = 50;
int Relay_delay = 300;
int NO_LAUNCH = 0;
int potval = 0;
int MAXBRIGHT = 150;
int MINBRIGHT = 20;
```

```
Adafruit_NeoPixel grid = Adafruit_NeoPixel(NUM_PIXELS, NEO_DATA_PIN,
```

```
//Colors we may want to use
```

```
uint32_t colorWhite = grid.Color(255, 255, 255);
uint32_t colorBlack = grid.Color(0, 0, 0);
uint32_t colorRed = grid.Color(255, 0, 0);
uint32_t colorGreen = grid.Color(0, 255, 0);
uint32_t colorBlue = grid.Color(0, 0, 255);
```

```
//Define Neopixel segments to turn on
```

```
int arrONE[] = {6,7,28,29,-1};
int arrTWO[] = {0,1,2,3,4,5,28,29,30,31,-1};
int arrTHREE[] = {0,1,6,7,4,5,28,29,30,31,-1};
int arrFOUR[] = {6,7,28,29,4,5,33,32,31,30,-1};
int arrFIVE[] = {30,31,32,33,4,5,6,7,1,0,-1};
int arrSIX[] = {0,1,2,3,4,5,6,7,30,31,32,33,-1};
int arrSEVEN[] = {6,7,28,29,30,31,-1};
int arrEIGHT[] = {0,1,6,7,4,5,28,29,30,31,32,33,2,3,-1};
int arrNINE[] = {6,7,4,5,28,29,30,31,32,33,-1};
int arrZERO[] = {0,1,6,7,28,29,30,31,32,33,2,3,-1};
int arrARMED[] = {22,23,24,25,-1};
int arrTEST[] = {16,17,18,-1};
int arrDISARMED[] = {22,23,24,25,26,27,-1};
int arrARMEDALL[] = {22,23,24,25,19,20,21,-1};
int arrNODE1[] = {15,-1};
int arrNODE2[] = {14,-1};
int arrNODE3[] = {13,-1};
```

```
int arrNODE4[] = {12,-1};
int arrNODE5[] = {8,-1};
int arrNODE6[] = {9,-1};
int arrNODE7[] = {10,-1};
int arrNODE8[] = {11,-1};
```

```
void setup(){
    //define output pins (set1) relay firing pins
    for(int pins4 = 23; pins4 < 38; pins4 = pins4 + 2){
        pinMode(pins4, OUTPUT);
    }
    all_relays_off();
    //define input pins (set1) sensing nodes and key selector
    for(int pins1 = 39; pins1 < 54; pins1++){
        pinMode(pins1, INPUT);
    }
    //define input pins (set2) remote inputs
    for(int pins2 = 10; pins2 < 14; pins2++){
        pinMode(pins2, INPUT);
    }
    //define input pins (set3) launch buttons input
    for(int pins3 = 24; pins3 < 39; pins3 = pins3 + 2){
        pinMode(pins3, INPUT);
    }

    //define output pins (set2) NeoPixel Pin
    pinMode(NEO_DATA_PIN, OUTPUT);
    //testing
    /*pinMode(46, INPUT);
    pinMode(48, INPUT);
    pinMode(50, INPUT);
    pinMode(52, INPUT);
    */
    //no need to set button led pins as they are analogWrites via PWM

    grid.begin();
    grid.show();// Initialize all pixels to 'off'
```

```
}
```

```
void loop(){  
    potval =analogRead(POTPIN);  
    BRIGHTNESS =map(potval,0,1023,MINBRIGHT,MAXBRIGHT); //adjust brig  
    grid.setBrightness(BRIGHTNESS);  
    keystate();//check armed mode  
    if(ARM_Mode == 1){  
        disarmed_setup();  
    }else if(ARM_Mode == 2){  
        test_setup();  
    }else if(ARM_Mode == 3){  
        fire_setup();  
    }else if(ARM_Mode == 4){  
        fire_all_setup();  
    }else{  
        delay(10);  
    }  
    delay(300); //to help eliminate any bouncing effect of key swit  
}
```

```
void fire_all_setup(){  
    colorWipe(colorBlack,0);  
    paintWord(arrARMEDALL,colorRed);  
    nodecheck();  
    node_display();  
    launch_button_status();  
    remote_button_status();  
    analogWrite(LAUNCH_BTN_LED1,BRIGHTNESS);  
    if(Launch1 ==HIGH || Remote1 ==HIGH){  
        if(Node1 ==LOW){  
            digitalWrite(RELAY1,LOW); //Pull low to activate relay  
            paintWord(arrNODE1,colorRed);  
            countdown();  
            digitalWrite(RELAY1,HIGH);  
            paintWord(arrNODE1,colorGreen);  
        }  
        if(Node2 ==LOW){
```

```
digitalWrite(RELAY2,LOW); //Pull low to activate relay
paintWord(arrNODE2,colorRed);
countdown();
digitalWrite(RELAY2,HIGH);
paintWord(arrNODE2,colorGreen);
}
if(Node3 ==LOW){
digitalWrite(RELAY3,LOW); //Pull low to activate relay
paintWord(arrNODE3,colorRed);
countdown();
digitalWrite(RELAY3,HIGH);
paintWord(arrNODE3,colorGreen);
}
if(Node4 ==LOW){
digitalWrite(RELAY4,LOW); //Pull low to activate relay
paintWord(arrNODE4,colorRed);
countdown();
digitalWrite(RELAY4,HIGH);
paintWord(arrNODE4,colorGreen);
}
if(Node5 ==LOW){
digitalWrite(RELAY5,LOW); //Pull low to activate relay
paintWord(arrNODE5,colorRed);
countdown();
digitalWrite(RELAY5,HIGH);
paintWord(arrNODE5,colorGreen);
}
if(Node6 ==LOW){
digitalWrite(RELAY6,LOW); //Pull low to activate relay
paintWord(arrNODE6,colorRed);
countdown();
digitalWrite(RELAY6,HIGH);
paintWord(arrNODE6,colorGreen);
}
if(Node7 ==LOW){
digitalWrite(RELAY7,LOW); //Pull low to activate relay
paintWord(arrNODE7,colorRed);
countdown();
```

```

    digitalWrite(RELAY7,HIGH);
    paintWord(arrNODE7,colorGreen);
}
if(Node8 ==LOW){
    digitalWrite(RELAY8,LOW); //Pull low to activate relay
    paintWord(arrNODE8,colorRed);
    countdown();
    digitalWrite(RELAY8,HIGH);
    paintWord(arrNODE8,colorGreen);
}
}else{delay(10);}
}

```

```

void fire_setup(){
    NO_LAUNCH = 0;//reset the no_launch function abort value
    colorWipe(colorBlack, 0);
    paintWord(arrARMED,colorRed);
    nodecheck();
    buttonlights();
    node_display();
    launch_button_status();
    remote_button_status();
    relay_fire();

    if(NO_LAUNCH < 1){

        //begin countdown, relay is active this whole time
        countdown();
    }

    all_relays_off();//turn off all relays
}

```

```

void countdown(){
    paintWord(arrEIGHT,colorBlack);
    paintWord(arrTHREE,colorRed);
    delay(1000);
    paintWord(arrEIGHT,colorBlack);
}

```

```
    paintWord(arrTWO,colorRed);
    delay(1000);
    paintWord(arrEIGHT,colorBlack);
    paintWord(arrONE,colorRed);
    delay(1000);
    paintWord(arrZERO,colorRed);
    delay(1000);
}
```

```
void all_relays_off(){
    digitalWrite(RELAY1,HIGH);
    digitalWrite(RELAY2,HIGH);
    digitalWrite(RELAY3,HIGH);
    digitalWrite(RELAY4,HIGH);
    digitalWrite(RELAY5,HIGH);
    digitalWrite(RELAY6,HIGH);
    digitalWrite(RELAY7,HIGH);
    digitalWrite(RELAY8,HIGH);
}
```

```
void relay_fire(){
    if(Launch1 ==HIGH || Remote1 ==HIGH){
        digitalWrite(RELAY1,LOW); //Pull low to activate relay
        paintWord(arrNODE1,colorRed);
    }else if(Launch2 ==HIGH || Remote2 ==HIGH){
        digitalWrite(RELAY2,LOW);
        paintWord(arrNODE2,colorRed);
    }else if(Launch3 ==HIGH || Remote3 ==HIGH){
        digitalWrite(RELAY3,LOW);
        paintWord(arrNODE3,colorRed);
    }else if(Launch4 ==HIGH || Remote4 ==HIGH){
        digitalWrite(RELAY4,LOW);
        paintWord(arrNODE4,colorRed);
    }else if(Launch5 ==HIGH){
        digitalWrite(RELAY5,LOW);
        paintWord(arrNODE5,colorRed);
    }else if(Launch6 ==HIGH){
        digitalWrite(RELAY6,LOW);
    }
}
```



```

    paintWord(arrNODE6,colorRed);
}else if(Launch7 ==HIGH){
    digitalWrite(RELAY7,LOW);
    paintWord(arrNODE7,colorRed);
}else if(Launch8 ==HIGH){
    digitalWrite(RELAY8,LOW);
    paintWord(arrNODE8,colorRed);
}else{
NO_LAUNCH = 1;
}
}

```

```

void remote_button_status(){
    Remote1 =digitalRead(REMOTE_A);
    Remote2 =digitalRead(REMOTE_B);
    Remote3 =digitalRead(REMOTE_C);
    Remote4 =digitalRead(REMOTE_D);
}

```

```

void launch_button_status(){
    Launch1 =digitalRead(LAUNCH_BTN1);
    Launch2 =digitalRead(LAUNCH_BTN2);
    Launch3 =digitalRead(LAUNCH_BTN3);
    Launch4 =digitalRead(LAUNCH_BTN4);
    Launch5 =digitalRead(LAUNCH_BTN5);
    Launch6 =digitalRead(LAUNCH_BTN6);
    Launch7 =digitalRead(LAUNCH_BTN7);
    Launch8 =digitalRead(LAUNCH_BTN8);
}

```

```

void test_setup(){
    colorWipe(colorBlack, 0);
    buttonlightsoff();
    paintWord(arrTEST, colorGreen);
    nodecheck();
    node_display();
}

```

```

void node_display(){
  if(Node1 ==LOW){
    paintWord(arrNODE1,colorGreen);
  }else{paintWord(arrNODE1,colorBlack);}
  if(Node2 ==LOW){
    paintWord(arrNODE2,colorGreen);
  }else{paintWord(arrNODE2,colorBlack);}
  if(Node3 ==LOW){
    paintWord(arrNODE3,colorGreen);
  }else{paintWord(arrNODE3,colorBlack);}
  if(Node4 ==LOW){
    paintWord(arrNODE4,colorGreen);
  }else{paintWord(arrNODE4,colorBlack);}
  if(Node5 ==LOW){
    paintWord(arrNODE5,colorGreen);
  }else{paintWord(arrNODE5,colorBlack);}
  if(Node6 ==LOW){
    paintWord(arrNODE6,colorGreen);
  }else{paintWord(arrNODE6,colorBlack);}
  if(Node7 ==LOW){
    paintWord(arrNODE7,colorGreen);
  }else{paintWord(arrNODE7,colorBlack);}
  if(Node8 ==LOW){
    paintWord(arrNODE8,colorGreen);
  }else{paintWord(arrNODE8,colorBlack);}
}

```

```

void disarmed_setup(){//define what the system does when in disarm
  colorWipe(colorBlack, 0);
  buttonlightsoff();
  paintWord(arrDISARMED, colorGreen);
}

```

```

void buttonlightsoff(){
  analogWrite(LAUNCH_BTN_LED1,0);
  analogWrite(LAUNCH_BTN_LED2,0);
  analogWrite(LAUNCH_BTN_LED3,0);
  analogWrite(LAUNCH_BTN_LED4,0);
}

```

```
    analogWrite(LAUNCH_BTN_LED5,0);
    analogWrite(LAUNCH_BTN_LED6,0);
    analogWrite(LAUNCH_BTN_LED7,0);
    analogWrite(LAUNCH_BTN_LED8,0);
}
```

```
void buttonlights(){
    if(Node1 ==LOW){
        analogWrite(LAUNCH_BTN_LED1,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED1,0);}
    if(Node2 ==LOW){
        analogWrite(LAUNCH_BTN_LED2,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED2,0);}
    if(Node3 ==LOW){
        analogWrite(LAUNCH_BTN_LED3,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED3,0);}
    if(Node4 ==LOW){
        analogWrite(LAUNCH_BTN_LED4,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED4,0);}
    if(Node5 ==LOW){
        analogWrite(LAUNCH_BTN_LED5,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED5,0);}
    if(Node6 ==LOW){
        analogWrite(LAUNCH_BTN_LED6,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED6,0);}
    if(Node7 ==LOW){
        analogWrite(LAUNCH_BTN_LED7,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED7,0);}
    if(Node8 ==LOW){
        analogWrite(LAUNCH_BTN_LED8,BRIGHTNESS);
    }else{analogWrite(LAUNCH_BTN_LED8,0);}
}
```

```
void nodecheck(){
    Node1 =digitalRead(SENSE_NODE1);
    Node2 =digitalRead(SENSE_NODE2);
    Node3 =digitalRead(SENSE_NODE3);
    Node4 =digitalRead(SENSE_NODE4);
```

```

Node5 =digitalRead(SENSE_NODE5);
Node6 =digitalRead(SENSE_NODE6);
Node7 =digitalRead(SENSE_NODE7);
Node8 =digitalRead(SENSE_NODE8);
}

```

```

void keystate(){
    if(digitalRead(KEY_SELECT_DISARM) ==HIGH){
        ARM_Mode = 1;
    }else if(digitalRead(KEY_SELECT_TEST) ==HIGH){
        ARM_Mode = 2;
    }else if(digitalRead(KEY_SELECT_FIRE) ==HIGH){
        ARM_Mode = 3;
    }else if(digitalRead(KEY_SELECT_FIREALL) ==HIGH){
        ARM_Mode = 4;
    }else{
        ARM_Mode = 0;//in case key selector is not on anything
    }
}

```

```

void paintWord(int arrWord[], uint32_t intColor){
    for(int i = 0; i < grid.numPixels() + 1; i++){
        if(arrWord[i] == -1){
            grid.show();
            break;
        }else{
            grid.setPixelColor(arrWord[i],intColor);
        }
    }
}

```

```

void colorWipe(uint32_t color, uint8_t wait) {
    for(uint16_t i=0; i<grid.numPixels(); i++) {
        grid.setPixelColor(i, color);
    }
    grid.show();
    delay(wait);
}

```

