// Vixen v3.4.2

// Dragon Fireworks

// Arduino MEGA 2560 R3

// Solid State Relay Boards

// by Stephen Peek 2017

// www.stephenpeek.co.uk/peek\_electronics.htm

// Define Arduino MEGA Pins EVEN 22-52 (16 Channels)

#define CHAN01 22

#define CHAN02 24

#define CHAN03 26

#define CHAN04 28

#define CHAN05 30

#define CHAN06 32

#define CHAN07 34

#define CHAN08 36

#define CHAN09 38

#define CHAN10 40

#define CHAN11 42

#define CHAN12 44

#define CHAN13 46

#define CHAN14 48

#define CHAN15 50

#define CHAN16 52

// Define Arduino MEGA Pins Odd 23-53 (16 Channels)

#define CHAN17 23

#define CHAN18 25

#define CHAN19 27

#define CHAN20 29

#define CHAN21 31

#define CHAN22 33

#define CHAN23 35

#define CHAN24 37

#define CHAN25 39

#define CHAN26 41

#define CHAN27 43

#define CHAN28 45

#define CHAN29 47

#define CHAN30 49

#define CHAN31 51

#define CHAN32 53

int channels[] = {CHAN01,CHAN02,CHAN03,CHAN04,CHAN05,CHAN06,CHAN07,CHAN08,CHAN09,

CHAN10,CHAN11,CHAN12,CHAN13,CHAN14,CHAN15,CHAN16,CHAN17,CHAN18,CHAN19,CHAN20,

CHAN21,CHAN22,CHAN23,CHAN24,CHAN25,CHAN26,CHAN27,CHAN28,CHAN29,CHAN30,CHAN31,CHAN32};

// Total Channel Count

#define CHANNEL\_COUNT 32

// Serial Speed

#define PC\_COM\_SPEED 57600

int i = 0;

int incomingByte[CHANNEL\_COUNT];

void setup()

{

 Serial.begin(PC\_COM\_SPEED);

 // Set the channel pins to output mode

 for(int channelIndex=0;channelIndex<CHANNEL\_COUNT;channelIndex++){

 pinMode(channels[channelIndex],OUTPUT);

 }

 SwitchOffAllRelays();

}

void loop()

{

 if (Serial.available() >= CHANNEL\_COUNT)

 {

 for (int i=0; i<CHANNEL\_COUNT; i++)

 {

 incomingByte[i] = 255-Serial.read();

 }

 digitalWrite(CHAN01, incomingByte[0]);

 digitalWrite(CHAN02, incomingByte[1]);

 digitalWrite(CHAN03, incomingByte[2]);

 digitalWrite(CHAN04, incomingByte[3]);

 digitalWrite(CHAN05, incomingByte[4]);

 digitalWrite(CHAN06, incomingByte[5]);

 digitalWrite(CHAN07, incomingByte[6]);

 digitalWrite(CHAN08, incomingByte[7]);

 digitalWrite(CHAN09, incomingByte[8]);

 digitalWrite(CHAN10, incomingByte[9]);

 digitalWrite(CHAN11, incomingByte[10]);

 digitalWrite(CHAN12, incomingByte[11]);

 digitalWrite(CHAN13, incomingByte[12]);

 digitalWrite(CHAN14, incomingByte[13]);

 digitalWrite(CHAN15, incomingByte[14]);

 digitalWrite(CHAN16, incomingByte[15]);

 digitalWrite(CHAN17, incomingByte[16]);

 digitalWrite(CHAN18, incomingByte[17]);

 digitalWrite(CHAN19, incomingByte[18]);

 digitalWrite(CHAN20, incomingByte[19]);

 digitalWrite(CHAN21, incomingByte[20]);

 digitalWrite(CHAN22, incomingByte[21]);

 digitalWrite(CHAN23, incomingByte[22]);

 digitalWrite(CHAN24, incomingByte[23]);

 digitalWrite(CHAN25, incomingByte[24]);

 digitalWrite(CHAN26, incomingByte[25]);

 digitalWrite(CHAN27, incomingByte[26]);

 digitalWrite(CHAN28, incomingByte[27]);

 digitalWrite(CHAN29, incomingByte[28]);

 digitalWrite(CHAN30, incomingByte[29]);

 digitalWrite(CHAN31, incomingByte[30]);

 digitalWrite(CHAN32, incomingByte[31]);

 }

}

void SwitchOffAllRelays()

{

 for(int channelIndex=0;channelIndex<CHANNEL\_COUNT;channelIndex++){

 analogWrite(channels[channelIndex], 255);

 }

}