Using an Arduino to Control Switch Servo Machines

by Charlie Long,

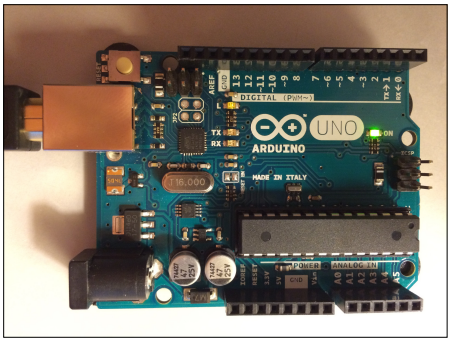
Several years ago I substituted low cost servo motors for the high maintenance twin coil switch machines that I had installed almost 50 years ago on my layout. I tried utilizing the Tam Valley Depot Octopus III controller. The scheme I used was a single push button equipped with an LED indicator for each track switch. Each push will toggle between straight and divergent and the LED indicator would light when the switch was set for the divergent move. I have 14 powered switches on the layout and I have seven fully duplicated control panels. Any switch can be controlled from any panel. The push buttons control 5V relays so the wiring to the Octopus III's can be kept short and I utilized Tam Valley's remote relays to light the LED's so the LED's on each of the seven panels can be paralleled.

Unfortunately the Octopus III seemed to be affected by random electronic noise and as I ran a car the switches would change randomly. I tried filter capacitors at the power input but with no improvement. Also, while each servo output on the Octopus III can be adjusted for throw and delay, it required pushing buttons in sequence on the remote aligner along with shorting jumper changes that did not seem, at least to me, to be well documented or easy to accomplish.

Looking through the INTERNET I found the following: http://thenscaler.com/?page_id=174

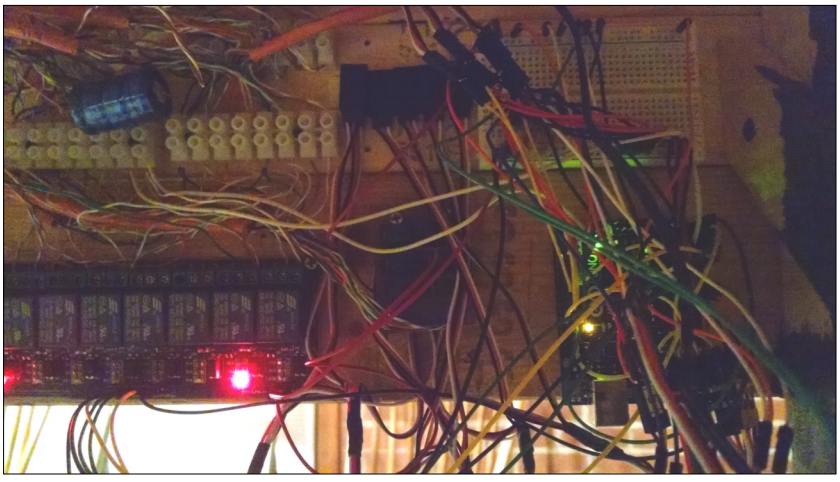
It looked like what I wanted to do so I bought a couple of Arduinos from All Electronics and started experimenting. I now have six track switches running from one Arduino and my experiment appears to be a success.

Listed below is the Arduino program (they are called sketches) I developed to operate up to six track switches: (Please note two slashes are used (//) to cause the compiler to ignore everything to the right of them on that line. Below the second line is used to describe what is happening in lines 3 to 14.)



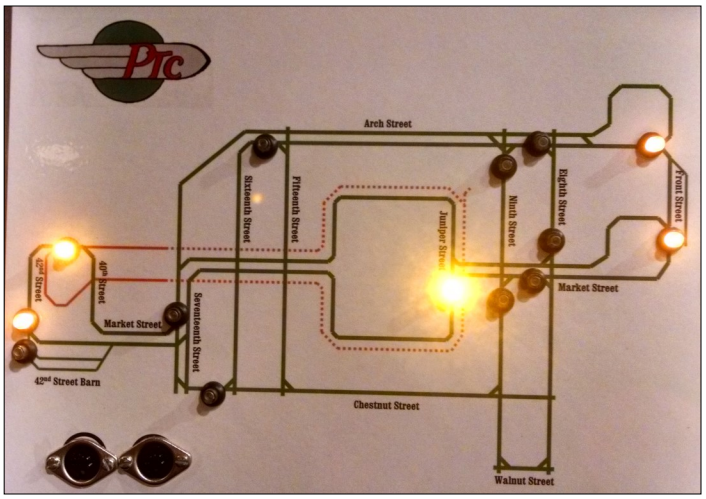
Arduino Uno with just the USB cable connected.

```
#include <Servo.h>
// constant variables used to set servo angles, in degrees
const int straight1 = 115;
const int straight2 = 80;
const int straight3 = 115;
const int straight4 = 70;
const int straight5 = 150;
const int straight6 = 115;
const int divergent1 = 75;
const int divergent2 = 115;
const int divergent3 = 75;
const int divergent4 = 110;
const int divergent5 = 100;
const int divergent6 = 75;
// constant variables holding the ids of the pins we are using
const int led1 = 14;
const int led2 = 15;
const int led3 = 16;
const int led4 = 17;
const int led5 = 18;
const int led6 = 19;
const int buttonpin1 = 2;
const int buttonpin2 = 3;
const int buttonpin3 = 4;
const int buttonpin4 = 5;
const int buttonpin5 = 6;
const int buttonpin6 = 7;
const int servopin1 = 8;
const int servopin2 = 9;
const int servopin3 = 10;
const int servopin4 = 11;
const int servopin5 = 12;
const int servopin6 = 13;
// servo movement step delay, in milliseconds
const int step_delay = 5;
// create a servo object
Servo servo1;
Servo servo2;
Servo servo3;
```



The Arduino is now installed and hiding behind some wires to the right. To the bottom left is the 8 relay bank, one of the component attached to the Arduino.

```
Servo servo4;
Servo servo5;
Servo servo6;
// global variables to store servo position
// current
int pos1 = straight1;
int pos2 = straight2;
int pos3 = straight3;
int pos4 = straight4;
int pos5 = divergent5;
int pos6 = straight6;
// previous
int old_pos1 = pos1;
int old_pos2 = pos2;
int old_pos3 = pos3;
int old_pos4 = pos4;
int old_pos5 = pos5;
int old_pos6 = pos6;
void setup()
{
  // set the mode for the digital pins in use
  pinMode(buttonpin1, INPUT);
  pinMode(buttonpin2, INPUT);
  pinMode(buttonpin3, INPUT);
  pinMode(buttonpin4, INPUT);
  pinMode(buttonpin5, INPUT);
  pinMode(buttonpin6, INPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  pinMode(led6, OUTPUT);
  // setup the servo
  servo1.attach(servopin1); // attach to the servo on pin 8
  servo2.attach(servopin2); // attach to the servo on pin 9
  servo3.attach(servopin3); // attach to the servo on pin 10
  servo4.attach(servopin4); // attach to the servo on pin 11
  servo5.attach(servopin5); // attach to the servo on pin 12
  servo6.attach(servopin6); // attach to the servo on pin 13
  servo1.write(pos1); // set the initial servo 1 position
  servo2.write(pos2); // set the initial servo 2 position
  servo3.write(pos3); // set the initial servo 3 position
  servo4.write(pos4); // set the initial servo 4 position
  servo5.write(pos5); // set the initial servo 5 position
  servo6.write(pos6); // set the initial servo 6 position
  // set initial led states
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);
  digitalWrite(led4, HIGH);
  digitalWrite(led5, LOW);
  digitalWrite(led6, HIGH);
}
void loop()
{
  // start each iteration of the loop by reading the button
  // if the button is pressed (reads HIGH), move the servo
  int button_state1 = digitalRead(buttonpin1);
  if (button_state1 == HIGH)
  {
    old_pos1 = pos1;
    // save the current positions
```



One of the control panels on the layout. Duplicates of it exist in several location around the layout so you can route your trolley from while standing at any DCC jack location.

Two of the servos that are used to move the swich points (one per switch) and are controlled by the push buttons via the Arduino.

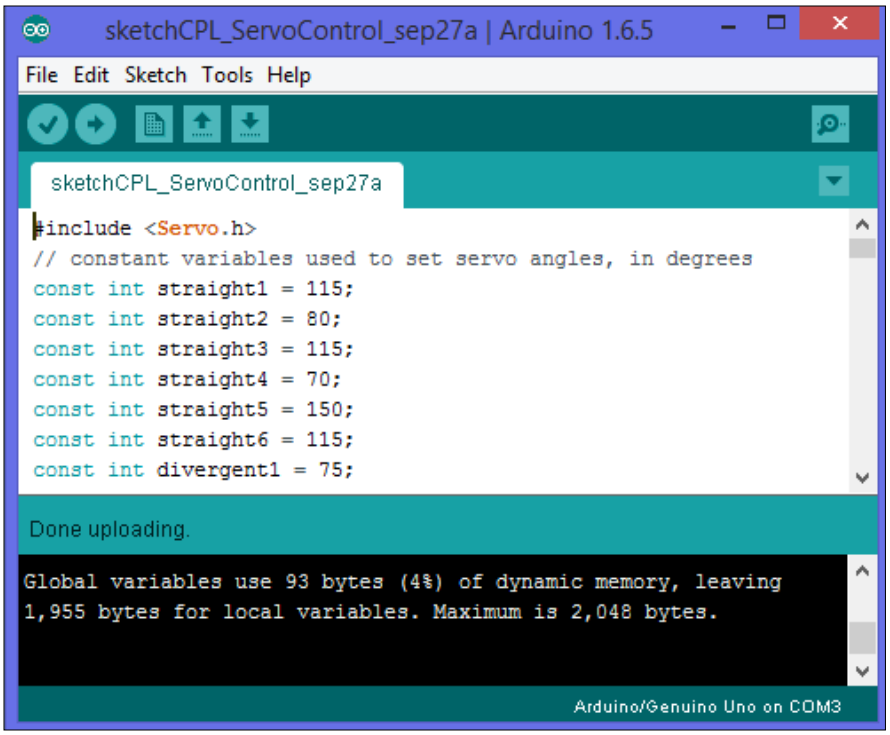
```
  // Toggle the position to the opposite value
  pos1 = pos1 == straight1 ? divergent1 : straight1;
  // Move the servo to its new position
  if (old_pos1 < pos1)
  {
    // if the new angle is higher
    //increment the servo position from old_pos to pos
    for (int i1 = old_pos1 + 1; i1 <= pos1; i1++) {
      servo1.write(i1);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  else
  {
    // otherwise the new angle is equal or lower
    // decrement the servo position from oldpos to pos
    for (int i1 = old_pos1 - 1; i1 >= pos1; i1--)
    {
      servo1.write(i1);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  // turn on the appropriate LED.
  if (pos1 == straight1)
  {
    digitalWrite(led1, HIGH);
  }
  else
  {
    digitalWrite(led1, LOW);
  }
}
// start each iteration of the loop by reading the button
// if the button is pressed (reads HIGH), move the servo
int button_state2 = digitalRead(buttonpin2);
if (button_state2 == HIGH)
{
  old_pos2 = pos2;
  // save the current positions
  // Toggle the position to the opposite value
  pos2 = pos2 == straight2 ? divergent2 : straight2;
  // Move the servo to its new position
  if (old_pos2 < pos2) { // if the new angle is higher
    //increment the servo position from old_pos to pos
    for (int i2 = old_pos2 + 1; i2 <= pos2; i2++)
    {
      servo2.write(i2);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  else
  {
    // otherwise the new angle is equal or lower
    // decrement the servo position from old_pos to pos
    for (int i2 = old_pos2 - 1; i2 >= pos2; i2--)
    {
      servo2.write(i2);
```

```
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  // turn on the appropriate LED.
  if (pos2 == straight2)
  {
    digitalWrite(led2, HIGH);
  }
  else
  {
    digitalWrite(led2, LOW);
  }
}
// start each iteration of the loop by reading the button
// if the button is pressed (reads HIGH), move the servo
int button_state3 = digitalRead(buttonpin3);
if (button_state3 == HIGH)
{
  old_pos3 = pos3;
  // save the current positions
  // Toggle the position to the opposite value
  pos3 = pos3 == straight3 ? divergent3 : straight3;
  // Move the servo to its new position
  if (old_pos3 < pos3)
  {
    // if the new angle is higher
    //increment the servo position from old_pos to pos
    for (int i3 = old_pos3 + 1; i3 <= pos3; i3++)
    {
      servo3.write(i3);
      // write the next positionto the servo
      delay(step_delay);
      // wait
    }
  }
  else
  {
    // otherwise the new angle is equal or lower
    // decrement the servo position from oldpos to pos
    for (int i3 = old_pos3 - 1; i3 >= pos3; i3--)
    {
      servo3.write(i3);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  // turn on the appropriate LED.
  if (pos3 == straight3)
  {
    digitalWrite(led3, HIGH);
  }
  else
  {
    digitalWrite(led3, LOW);
  }
}
// start each iteration of the loop by reading the button
// if the button is pressed (reads HIGH), move the servo
int button_state4 = digitalRead(buttonpin4);
if (button_state4 == HIGH) {
```

```
  old_pos4 = pos4;
  // save the current positions
  // Toggle the position to the opposite value
  pos4 = pos4 == straight4 ? divergent4 : straight4;
  // Move the servo to its new position
  if (old_pos4 < pos4) {
    // if the new angle is higher
    //increment the servo position from old_pos to pos
    for (int i4 = old_pos4 + 1; i4 <= pos4; i4++)
    {
      servo4.write(i4);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  else
  {
    // otherwise the new angle is equal or lower
    // decrement the servo position from oldpos to pos
    for (int i4 = old_pos4 - 1; i4 >= pos4; i4--)
    {
      servo4.write(i4);
      // write the next position to the servo
      delay(step_delay);
      // wait
    }
  }
  // turn on the appropriate LED.
  if (pos4 == straight4)
  {
    digitalWrite(led4, HIGH);
  }
  else
  {
    digitalWrite(led4, LOW);
  }
}
// start each iteration of the loop by reading the button
// if the button is pressed (reads HIGH), move the servo
int button_state5 = digitalRead(buttonpin5);
if (button_state5 == HIGH)
{
  old_pos5 = pos5;
  // save the current positions
  // Toggle the position to the opposite value
  pos5 = pos5 == straight5 ? divergent5 : straight5;
  // Move the servo to its new position
  if (old_pos5 < pos5)
  {
    // if the new angle is higher
    //increment the servo position from old_pos to pos
    for (int i5 = old_pos5 + 1; i5 <= pos5; i5++)
    {
      servo5.write(i5);
      // write the next position to the servo
      delay(step_delay); // wait
    }
  }
  else
  {
    // otherwise the new angle is equal or lower
    // decrement the servo position from oldpos to pos
```

```
      for (int i5 = old_pos5 - 1; i5 >= pos5; i5--)
      {
        servo5.write(i5); // write the next position to the servo
        delay(step_delay); // wait
      }
    }
    // turn on the appropriate LED.
    if (pos5 == straight5)
    {
      digitalWrite(led5, HIGH);
    }
    else
    {
      digitalWrite(led5, LOW);
    }
  }
  // start each iteration of the loop by reading the button
  // if the button is pressed (reads HIGH), move the servo
  int button_state6 = digitalRead(buttonpin6);
  if (button_state6 == HIGH)
  {
    old_pos6 = pos6;
    // save the current positions
    // Toggle the position to the opposite value
    pos6 = pos6 == straight6 ? divergent6 : straight6;
    // Move the servo to its new position
    if (old_pos6 < pos6) { // if the new angle is higher
      //increment the servo position from old_pos to pos
      for (int i6 = old_pos6 + 1; i6 <= pos6; i6++)
      {
        servo6.write(i6);
        // write the next position to the servo
        delay(step_delay);
        // wait
      }
    }
    else
    {
      // otherwise the new angle is equal or lower
      // decrement the servo position from old_pos to pos
      for (int i6 = old_pos6 - 1; i6 >= pos6; i6--)
      {
        servo6.write(i6);
        // write the next position to the servo
        delay(step_delay);
        // wait
      }
    }
    // turn on the appropriate LED.
    if (pos6 == straight6)
    {
      digitalWrite(led6, HIGH);
    } else {
      digitalWrite(led6, LOW);
    }
  }
}
// end of loop
```

I am not sure how efficient this program is but it seems to do what I want. The big advantage of the Arduino is that different programs can be developed on a laptop and easily uploaded for testing.



To the left is a screen shot of the Arduino application after uploading the program above onto the Arduino UNO. The programing is case sensitive, so if you get an error from the compiler and it is the beginning of the line of code which happens to be capitalized while the rest remain in lower case, it might be something as simple as changing the letter to lower case. You can go to the Arduino website and download the application for free and play around with the program prior to purchasing the board. While line number do make it easier to read the lines below, they are not included so you may copy them and paste right to the application/compiler. One other item to note: Under "Tools" the correct board must be selected and the "Port" set to COM3 to successfully upload onto the Arduino. (You can probably guess what didn't happen on the first attempt). *David Gallagher screen shot and comment*

Relay Board and LED Test Program

The following tests the relay board I bought from Amazon to light the panel LED's. It continuously activates each LED output in sequence:

sketch_LED_Test

```
#include <Servo.h>
// constant variables used to set servo angles, in degrees
const int straight1 = 115;
const int straight2 = 115;
const int straight3 = 115;
const int straight4 = 115;
const int straight5 = 115;
const int straight6 = 115;
const int divergent1 = 75;
const int divergent2 = 75;
const int divergent3 = 75;
const int divergent4 = 75;
const int divergent5 = 75;
const int divergent6 = 75;
// constant variables holding the ids of the pins we are using
const int led1 = 14;
const int led2 = 15;
const int led3 = 16;
const int led4 = 17;
const int led5 = 18;
const int led6 = 19;
const int buttonpin1 = 2;
const int buttonpin2 = 3;
const int buttonpin3 = 4;
const int buttonpin4 = 5;
const int buttonpin5 = 6;
const int buttonpin6 = 7;
const int servopin1 = 8;
const int servopin2 = 9;
const int servopin3 = 10;
```

```
const int servopin4 = 11;
const int servopin5 = 12;
const int servopin6 = 13;
  // servo movement step delay, in milliseconds
const int step_delay = 200;
  // create a servo object
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
// global variables to store servo position
// current
int pos1 = straight1;
int pos2 = straight2;
int pos3 = straight3;
int pos4 = straight4;
int pos5 = straight5;
int pos6 = straight6;
// previous int old_pos1 = pos1;
int old_pos2 = pos2;
int old_pos3 = pos3;
int old_pos4 = pos4;
int old_pos5 = pos5;
int old_pos6 = pos6;
void setup()
{
// set the mode for the digital pins in use
pinMode(buttonpin1, INPUT);
pinMode(buttonpin2, INPUT);
pinMode(buttonpin3, INPUT);
pinMode(buttonpin4, INPUT);
pinMode(buttonpin5, INPUT);
pinMode(buttonpin6, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
pinMode(led5, OUTPUT);
pinMode(led6, OUTPUT);
// setup the servo
servo1.attach(servopin1);
// attach to the servo on pin 8
servo2.attach(servopin2);
// attach to the servo on pin 9
servo3.attach(servopin3);
// attach to the servo on pin 10
servo4.attach(servopin4);
// attach to the servo on pin 11
servo5.attach(servopin5);
// attach to the servo on pin 12
servo6.attach(servopin6);
// attach to the servo on pin 13
servo1.write(pos1);
// set the initial servo 1 position
servo2.write(pos2);
// set the initial servo 2 position
servo3.write(pos3);
// set the initial servo 3 position
servo4.write(pos4);
// set the initial servo 4 position
servo5.write(pos5); //
```

```
set the initial servo 5 position
servo6.write(pos6);
// set the initial servo 6 position
// set initial led states
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
}
void loop()
{
// turn on  LED 1
digitalWrite(led1, LOW);
delay(step_delay);
//wait
// turn off LED 1
digitalWrite(led1, HIGH);
delay(step_delay);
//wait
// turn on LED 2
digitalWrite(led2, LOW);
delay(step_delay);
//wait
// turn off LED 2
digitalWrite(led2, HIGH);
delay(step_delay);
//wait
// turn on  LED 3
digitalWrite(led3, LOW);
delay(step_delay);
//wait
// turn off LED 3
digitalWrite(led3, HIGH);
delay(step_delay);
//wait
// turn on  LED 4
digitalWrite(led4, LOW);
delay(step_delay);
//wait
// turn off LED 4
digitalWrite(led4, HIGH);
delay(step_delay);
//wait
// turn on  LED 5
digitalWrite(led5, LOW);
delay(step_delay);
//wait
// turn off LED 5
digitalWrite(led5, HIGH);
delay(step_delay);
//wait
// turn on  LED 6
digitalWrite(led6, LOW);
delay(step_delay);
//wait
// turn off LED 6
digitalWrite(led6, HIGH);
delay(step_delay);
//wait
}
// end of loop
```

Test for Push Buttons and Associated LED

The following tests the push buttons by lighting the associated LED When the button is pushed:

sketch_BUTTON_Test

```
#include <Servo.h>
//constant variables used to set servo angles, in degrees
const int straight1 = 115;
const int straight2 = 115;
const int straight3 = 115;
const int straight4 = 115;
const int straight5 = 115;
const int straight6 = 115;
const int divergent1 = 75;
const int divergent2 = 75;
const int divergent3 = 75;
const int divergent4 = 75;
const int divergent5 = 75;
const int divergent6 = 75;
// constant variables holding the ids of the pins we are using
const int led1 = 14;
const int led2 = 15;
const int led3 = 16;
const int led4 = 17;
const int led5 = 18;
const int led6 = 19;
const int buttonpin1 = 2;
const int buttonpin2 = 3;
const int buttonpin3 = 4;
const int buttonpin4 = 5;
const int buttonpin5 = 6;
const int buttonpin6 = 7;
const int servopin1 = 8;
const int servopin2 = 9;
const int servopin3 = 10;
const int servopin4 = 11;
const int servopin5 = 12;
const int servopin6 = 13;
  // servo movement step delay, in milliseconds
const int step_delay = 200;
  // create a servo object
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;
// global variables to store servo position
// current
int pos1 = straight1;
int pos2 = straight2;
int pos3 = straight3;
int pos4 = straight4;
int pos5 = straight5;
int pos6 = straight6;
// previous
int old_pos1 = pos1;
int old_pos2 = pos2;
int old_pos3 = pos3;
```

```
int old_pos4 = pos4;
int old_pos5 = pos5;
int old_pos6 = pos6;
void setup()
{
// set the mode for the digital pins in use
pinMode(buttonpin1, INPUT);
pinMode(buttonpin2, INPUT);
pinMode(buttonpin3, INPUT);
pinMode(buttonpin4, INPUT);
pinMode(buttonpin5, INPUT);
pinMode(buttonpin6, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);
pinMode(led4, OUTPUT);
pinMode(led5, OUTPUT);
pinMode(led6, OUTPUT);
  // setup the servo
servo1.attach(servopin1); // attach to the servo on pin 8
servo2.attach(servopin2); // attach to the servo on pin 9
servo3.attach(servopin3); // attach to the servo on pin 10
servo4.attach(servopin4); // attach to the servo on pin 11
servo5.attach(servopin5); // attach to the servo on pin 12
servo6.attach(servopin6); // attach to the servo on pin 13
servo1.write(pos1); // set the initial servo 1 position
servo2.write(pos2); // set the initial servo 2 position
servo3.write(pos3); // set the initial servo 3 position
servo4.write(pos4); // set the initial servo 4 position
servo5.write(pos5); // set the initial servo 5 position
servo6.write(pos6); // set the initial servo 6 position
// set initial led states
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
}
void loop()
{
// turn on  LED 1 with button
Int button_state1=digitalRead(buttonpin1);
if(button_state1==HIGH)
{
digitalWrite(led1, LOW);
}
Else
{
// turn off LED 1
digitalWrite(led1, HIGH);
}
// turn on  LED 2 with button
Int button_state2=digitalRead(buttonpin2);
if(button_state2==HIGH)
{
digitalWrite(led2, LOW);
}
Else
{
// turn off LED 2
digitalWrite(led2, HIGH);
}
```

```
// turn on  LED 3 with button
Int button_state3=digitalRead(buttonpin3);
if(button_state3==HIGH)
{
digitalWrite(led3, LOW);
}
Else
{
// turn off LED 3
digitalWrite(led3, HIGH);
}
// turn on  LED 4 with button
Int button_state4=digitalRead(buttonpin4);
if(button_state4==HIGH)
{
digitalWrite(led4, LOW);
}
Else
{
// turn off LED 4
digitalWrite(led4, HIGH);
}
// turn on  LED 5 with button
int button_state5=digitalRead(buttonpin5);
if(button_state5==HIGH)
{
digitalWrite(led5, LOW);
}
Else
{
// turn off LED 5
digitalWrite(led5, HIGH);}
// turn on  LED 6 with button
int button_state6=digitalRead(buttonpin6);
if(button_state6==HIGH)
{
digitalWrite(led6, LOW);
}
Else
{
// turn off LED 6
digitalWrite(led6, HIGH);
}

}
// end of loop
```

Depending on how much shipping is involved, the cost is actually less for the Arduino.
To do the 14 track switches the following Tam Valley equipment was required:

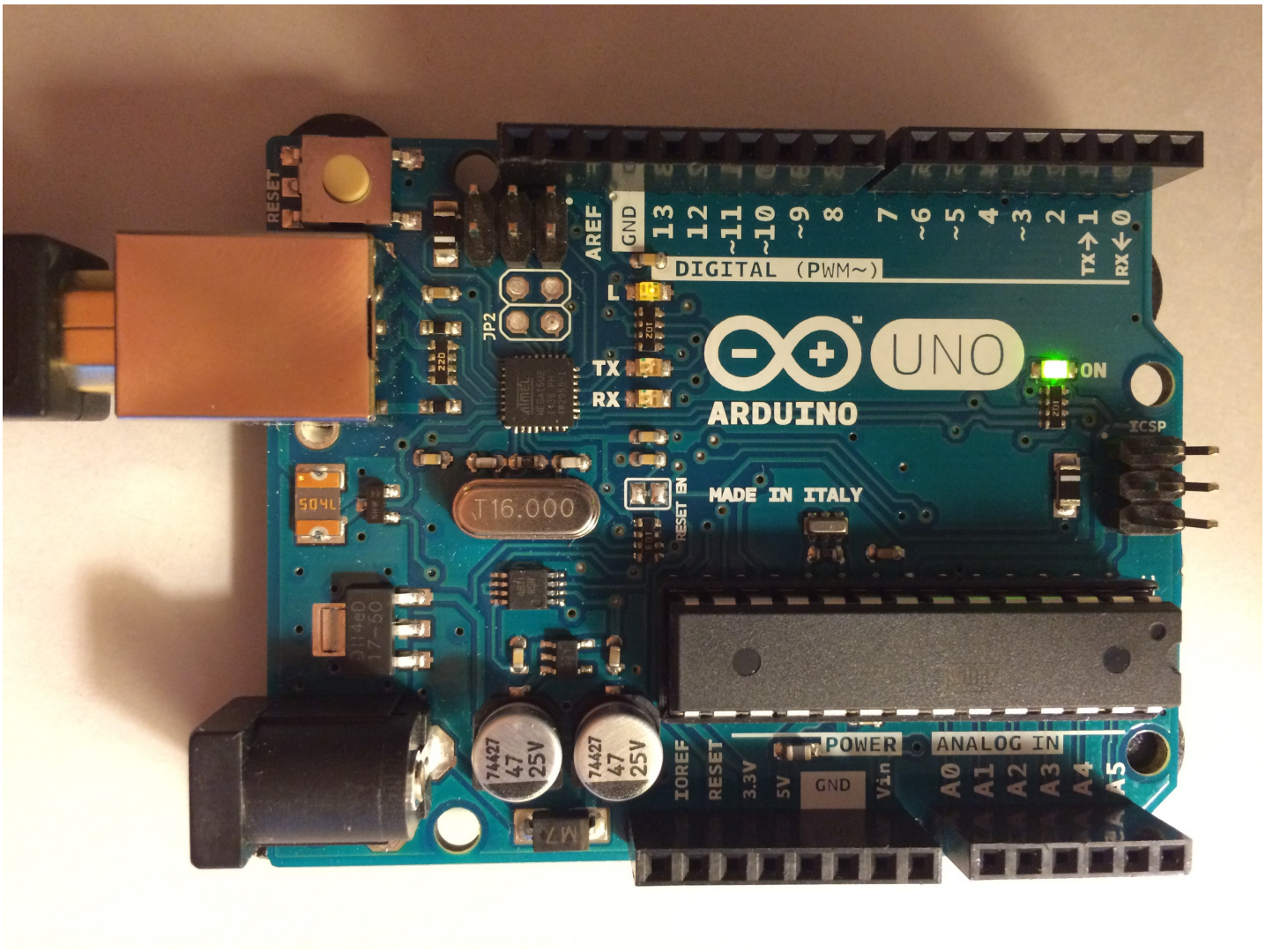
2 - Octopus III boards @ 39.95 each =	$79.90
1 - Remote aligner =	$12.00
1 - 12 Volt Power Supply =	$16.95
14 - DPDT remote relays =	$64.00

Total = $172.85

With Arduino I required the following:

3 – Arduino boards @ 24.20 each =	$72.60
2 - 9 volt power supplies @ 9.75 =	$19.50
2 - 5 volt power supplies @8.95 =	$17.90
1 - coax "Y" power adapter =	$ 2.25
2 - 8 channel relay boards @8.99 =	$17.98
1 - Set of breadboard jumpers =	$6.59

2 - Sets of breadboard jumpers =	$7.50
2 - Breadboards @ 4.00 each =	$8.00
14- Resistors @0.08 each =	$1.12
Total =	$153.44



Above is an enlarged photo of an Arduino Uno with just the USB cable attached. The digital pins are pictured above while the analog pins are below to the right.