

```
//Arduino Code By Sietze van de Star for TARSCI project 2016
```

```
//Define the states of the machine
```

```
#define LED_OFF 0
```

```
#define Servo2 1
```

```
#define LED_ON 2
```

```
#include "mpr121.h"
```

```
#include <Wire.h>
```

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

```
#include "WS2812_Definitions.h"
```

```
int irqpin = 2; // Digital 2
```

```
boolean touchStates[12]; //to keep track of the previous touch states
```

```
#define PIN 7
```

```
#define LED_COUNT 30
```

```
Adafruit_NeoPixel leds = Adafruit_NeoPixel(LED_COUNT, PIN, NEO_GRB + NEO_KHZ800);
```

```
//int pos = 90; // variable to store the servo position
```

```
//int pos1 = 0;
```

```
int Position = 90;
```

```
int Position1 = 0;
```

```
#include <Servo.h>
```

```
Servo myservo;
```

```
Servo myservo1;
```

```
Servo myservo2;
```

```
Servo myservo3;// create servo object to control a servo
```

```
Servo myservo4;
```

```
//This is the memory element that contains the state variable
```

```
uint8_t fsm_state = LED_OFF;
```

```
//This holds the number of milliseconds that have been counted
```

```
uint16_t msCounts = 0;
```

```
void setup() {
```

```
myservo.attach(5);
```

```
myservo1.attach(6);
```

```
myservo2.attach(9);
```

```
myservo3.attach(10);// attaches the servo on pin 9 to the servo object
```

```
myservo4.attach(11);
```

```
pinMode(irqpin, INPUT);
```

```
digitalWrite(irqpin, HIGH); //enable pullup resistor
```

```

Serial.begin(9600);
Wire.begin();

mpr121_setup();
//
=====
//LEDS

leds.begin(); // Call this to start up the LED strip.
clearLEDS(); // This function, defined below, turns all LEDs off...
leds.show(); // ...but the LEDs don't actually update until you call this.

}

void loop()
{
  readTouchInputs();

  showLedsHuman();

  if (touchStates[0] == true) {
    myservo.write(Position);
    myservo1.write(Position);
    myservo2.detach();
    myservo3.detach();
    myservo4.detach();

  } else {
    myservo.write(Position1);
    myservo1.write(Position1);
    myservo2.attach(9);
    myservo3.attach(10); // attaches the servo on pin 9 to the servo object
    myservo4.attach(11);

  }

  if (touchStates[2] == true) {

    myservo.detach();
    myservo1.detach();
    myservo2.write(Position);
    myservo3.write(Position); // tell servo to go to position in variable 'pos'
    myservo4.write(Position);

  } else {

    myservo.attach(5);
    myservo1.attach(6);
    myservo2.write(Position1);
    myservo3.write(Position1); // tell servo to go to position in variable 'pos'
    myservo4.write(Position1);

```

```

}

if (touchStates[4] == true) {

    myservo.detach();
    myservo1.detach();
    myservo2.detach();
    myservo3.detach();
    myservo4.write(Position);

} else {
    myservo.attach(5);
    myservo1.attach(6);
    myservo2.attach(9);
    myservo3.attach(10); // attaches the servo on pin 9 to the servo object
    myservo4.write(Position1);
}

}

//
=====
=====

void readTouchInputs() {
    if (!checkInterrupt()) {

        //read the touch state from the MPR121
        Wire.requestFrom(0x5A, 2);

        byte LSB = Wire.read();
        byte MSB = Wire.read();

        uint16_t touched = ((MSB << 8) | LSB); //16bits that make up the touch states

        for (int i = 0; i < 12; i++) { // Check what electrodes were pressed
            if (touched & (1 << i)) {

                if (touchStates[i] == 0) {
                    //pin i was just touched
                    Serial.print("pin ");
                    Serial.print(i);
                    Serial.println(" was just touched");
                } else if (touchStates[i] == 1) {
                    //pin i is still being touched
                }

                touchStates[i] = 1;
            } else {
                if (touchStates[i] == 1) {

```

```

    Serial.print("pin ");
    Serial.print(i);
    Serial.println(" is no longer being touched");

    //pin i is no longer being touched
}

touchStates[i] = 0;
}

}

}
}

```

```

boolean checkInterrupt(void) {
    return digitalRead(irqpin);
}

```

```

void set_register(int address, unsigned char r, unsigned char v) {
    Wire.beginTransaction(address);
    Wire.write(r);
    Wire.write(v);
    Wire.endTransmission();
}

```

```
//
```

```

=====
=====

```

```

void clearLEDS()
{
    for (int i = 0; i < LED_COUNT; i++)
    {
        leds.setPixelColor(i, 0, 0, 0);
        leds.show();//
    }
}

```

```

void showLedsHuman() {
    for (int i = 0; i < LED_COUNT; i++) {
        leds.setPixelColor(i, 255, 255, 255);
        leds.show();// Initialize all pixels to 'off'
    }
}

```

```

void showLedsSystem() {
    for (int i = 0; i < LED_COUNT; i++) {
        leds.setPixelColor(i, 0, 0, 255);
        leds.show();// Initialize all pixels to 'off'
    }
}

```