

# MAKER MANUAL

**Group 5: Dance Your Cares Away**

Audrina Aziz  
Célia Cannappah  
Nicole Wan

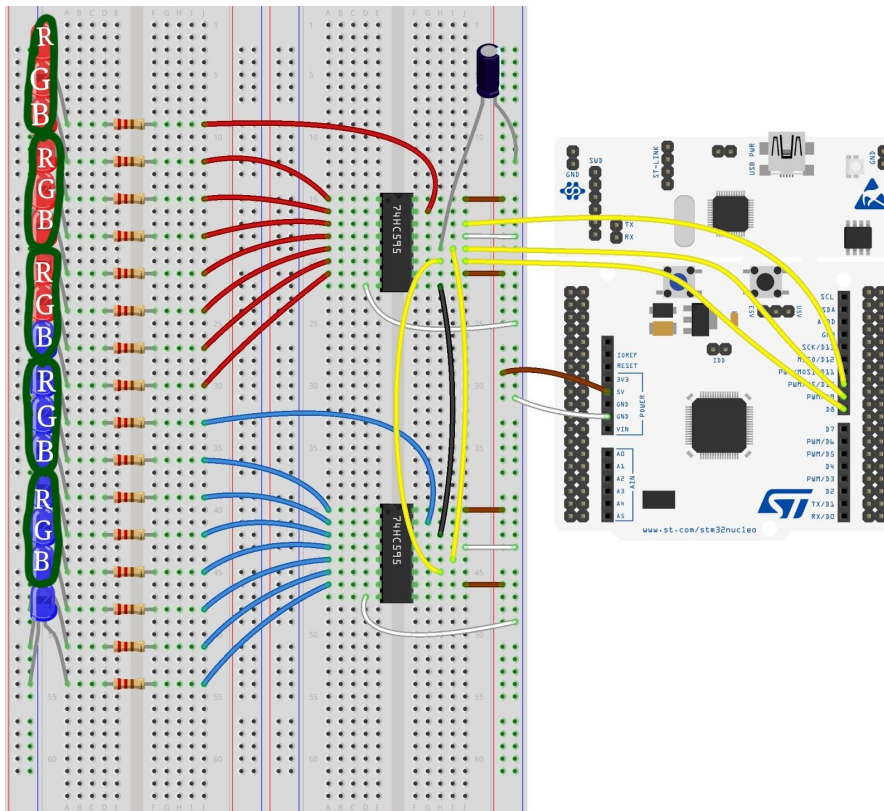
## About the Project:

Our final product was a prototype that used a potentiometer, in place of a flex sensor, as an input and RGB LEDs as the output. The LEDs were scattered on the bra and randomly assigned to a group. The potentiometer was used as a replacement as a user can still give the RGB LEDs analog input. The flex sensor was not used in the final prototype as it was unreliable. The original concept was using conductive knit fabric as a flex sensor along the bodysuit on the abdomen area. The abdomen area is where there is most movement on the particular type of dance we were targeting - belly dancing. Depending on the dancer's movement, it will stretch and compress the fabric, changing the resistance of the fabric itself. The change in resistance, will change which LED group on the bra will light up.

# Materials List

- 5mm RGB LEDs
- Conductive Thread
- Single core wire (black, red, green, blue)
- Shift registers
- Bra
- 560kΩ Resistors
- Adafruit Flora V2

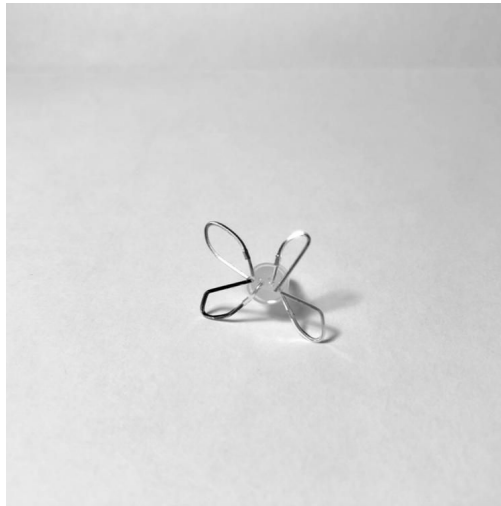
# Breadboard Layout



# Building Process

## Prepping the Components

- Mark the negative leg of the RGB LED with a black marker
- Using pliers and wire cutters, form the legs of the RGB LEDs like so:



*The bottom of an RGB LED with its legs cut and bent into loops*

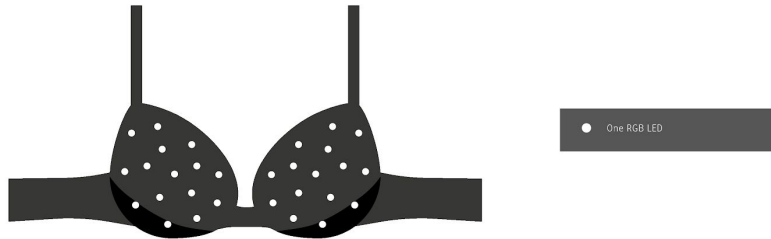
- Do the same for the resistors:



*Legs of resistor bent into a loop*

## Sewing the Components

- Sew (with normal thread) 15 LEDs on each bra cup. Scatter them randomly like so:

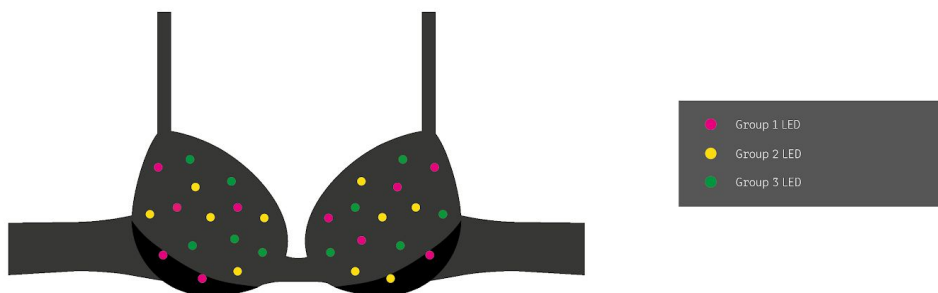


- Link one resistor to the negative leg of each LED. Then sew each resistor to the bra. Sew conductive thread tightly between the resistor and the LED leg to make sure there is consistent contact.



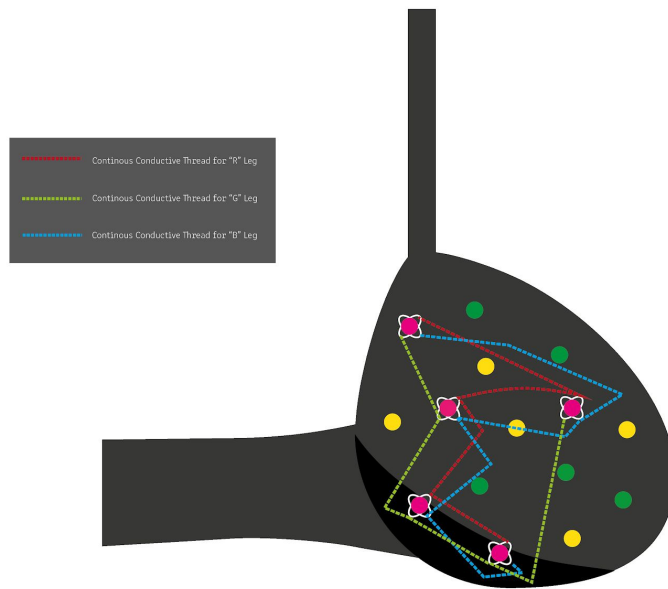
*Resistor linked and sewn*

- Divide the LEDs into 3 different groups. Mark them using markers.



*Example of marking the LEDs into three separate groups*

- For each bra cup, use conductive thread to connect each LED in a group. For example, for all LEDs in group 1 on the left bra cup, connect the R legs with one continuous conductive thread. Then use another continuous conductive thread to connect the G legs of all the LEDs in group 1. So on and so forth. ⚠ **Note:** Take care to not cross the conductive thread. They act like wires and when crossed, will mess up connections. **Tip:** make use of different levels within the padded material of the bra.



## Wiring

- Lay the breadboard out as shown in the breadboard layout above
- Each R shown in the breadboard layout represents a group of R legs. Connect one R leg to the shift register. Doing so will connect all the R legs.
- Repeat with G and B legs.
- Connect a single 560kΩ resistor to each GROUND legs of each LED.
- Finally, connect all of the above resistors to ground.



*One wire to connect all resistor ground in a series*

- For each bra cup, choose one LED from each LED group to wire up. On each chosen LED, attach a red, green, blue wire on the corresponding leg. This is so that it is clear which leg is which if any problems occur.
- Wire up the coloured wires to the shift register, as shown in the breadboard layout.

### Extra LEDs

The code included in this manual, allows the use of 6 sets of 5 RGB LEDs (3 sets on each side of the bra) and 20 single LEDs. By connecting the serial registers in parallel, just as the second serial register is connected to the first serial register. The single LEDs can be connected after the 18th output from the serial registers. This is because the first 18 outputs will be reserved for the RGB LEDs. The ordinary LEDs should include the complementary resistor to limit the flow of current to the LED. The LEDs must all be connected to ground from the negative cathode. They must also be connected in order of how they appear, for the wave effect.

# Code

```
//declare pinsU
int serialIn = 0;
int registerClk = 1;
int serialClk = 12;
int serialClr = 9;

int scrollVal = 0;
int turnPin = A10;

//boob groups
int groupBrightness = 10; //blinking tempo for fading effect
int groupOn = 0; //the group that will be on

int leftLine[10] = {19, 20, 21, 22, 23, 24, 25, 26, 27, 28}; //holds the position of each LED
int rightLine[10] = {29, 30, 31, 32, 33, 34, 35, 36, 37, 38}; //as above

//loop and tempo related declarations
int current = 295; //number count down
int rounds = 1;
int tempo; //delay after each round

//colour related variables
int c = 0; //variable for storing at which point in the colours array to show.
int colours[] = {1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1}; //array to store the
pattern for the colours
//stores rgb colours
//1 for on
//0 for off

void setup() {
  //setting the mode of the pins
  pinMode(serialIn, OUTPUT);
  pinMode(registerClk, OUTPUT);
  pinMode(serialClk, OUTPUT);
  pinMode(serialClr, OUTPUT);

  //setting the initial pin values
  digitalWrite(serialIn, LOW);
  digitalWrite(registerClk, LOW);
  digitalWrite(serialClk, LOW);
  digitalWrite(serialClr, LOW);

  //clearing out the registers
  digitalWrite(registerClk, HIGH);
  digitalWrite(registerClk, LOW);
  digitalWrite(serialClr, HIGH);
  Serial.begin(9600);
}

//contains everything that happens in a single moment, determined by the delay
void loop() {
  digitalWrite(registerClk, LOW); //resets the clock
```



```

scrollVal = analogRead(turnPin);
Serial.println(scrollVal);
int tempo = map(scrollVal, 0, 1023, 10, 500);

//fade in/out moment, dependant on group of LEDs
int remainder = current % groupBrightness;
char fade = 0; //0 is fade in , 1 is fade out

Serial.println(current);
Serial.println(remainder);

if (remainder == 0) {
  callmLED(group0n);
  Serial.println(group0n);
  Serial.println("done here");
  if (fade == 0) {
    rounds++; //adding rounds
    if (rounds == 5) {
      //below swaps fade to fade out
      groupBrightness --;
      if (groupBrightness == 0) {
        fade = 1;
      }
    }
  } else if (fade == 1) {
    rounds--; //subtracting rounds
    if (rounds == 0) {
      //below swaps fade to fade in/out
      groupBrightness ++;
      if (groupBrightness == 10) {
        fade = 0;
        Serial.println(fade);
        //group0n ++;
      }
    }
  }
} else {
  groupOFF();
}

callSingleLED(rounds);

digitalWrite(registerClk, HIGH); //sends the signals down.
if ( current == 0) {
  current = 295;
  //group0n = group0n + 1;
  Serial.println(group0n);
  if (group0n == 3) {
    group0n = 0;
  }
}
if ((current % 10) == 0) {
  changeColour;
}

delay(tempo);
current --;

```

```

    if (current == 0) {
        current = 255;
    }
}

void changeColour() {
    c = c + 3;
    //below resets the c variable back to zero so that the cycle continues
    if ( c == 21) {
        c = 0;
    }
}
//1. calling group led types. multicolour
//total 30 LEDs, 3 groups

void groupOFF() {
    for (int i = 0; i < 6; i++) {
        alloFF;
    }
}

//calls the coloured LED
void callmLED(int i) {
    if (i == 0) {
        allON;
        alloFF;
        alloFF;
        allON;
        alloFF;
        alloFF;
    }
    if (i == 1) {
        alloFF;
        allON;
        alloFF;
        alloFF;
        allON;
        alloFF;
    }
    if (i == 2) {
        alloFF;
        alloFF;
        allON;
        alloFF;
        alloFF;
        allON;
    }
}

//calls the led "strips"
void callSingleLED(int ledNumber) {
    int leftonLED = leftLine[ledNumber];
    int rightonLED = rightLine[ledNumber];
}

```

```

//left line turn all off aside from main
for (int offLed = leftLine[0]; offLed < (leftonLED - 1); offLed++) {
    digitalWrite(serialIn, LOW);
    digitalWrite(serialClk, HIGH);
    digitalWrite(serialClk, LOW);
}
for (int onLed = 0; onLed < 1; onLed++) {
    digitalWrite(serialIn, HIGH);
    digitalWrite(serialClk, HIGH);
    digitalWrite(serialClk, LOW);
}
for (int offLed = leftonLED; offLed < leftLine[10]; offLed++) {
    digitalWrite(serialIn, LOW);
    digitalWrite(serialClk, HIGH);
    digitalWrite(serialClk, LOW);

    //below repeated for the right line.
    for (int offLed = rightLine[0]; offLed < (rightonLED - 1); offLed++) {
        digitalWrite(serialIn, LOW);
        digitalWrite(serialClk, HIGH);
        digitalWrite(serialClk, LOW);
    }
    for (int onLed = 0; onLed < 1; onLed++) {
        digitalWrite(serialIn, HIGH);
        digitalWrite(serialClk, HIGH);
        digitalWrite(serialClk, LOW);
    }
    for (int offLed = rightonLED; offLed < rightLine[10]; offLed++) {
        digitalWrite(serialIn, LOW);
        digitalWrite(serialClk, HIGH);
        digitalWrite(serialClk, LOW);
    }
}
}

//turns all the multicoloured LEDs OFF
void alloFF() {
    for (int i = 0; i < 3; i++) {
        digitalWrite(serialIn, LOW);
        digitalWrite(serialClk, HIGH);
        digitalWrite(serialClk, LOW);
    }
}

//turn numbers on, for whichever numbers on RGB colourings
void allON() {
    for (int i = 0; i < 3; i++) {
        int colour = colours[i + c];
        if (colour == 1) {
            digitalWrite(serialIn, HIGH);
            digitalWrite(serialClk, HIGH);
            digitalWrite(serialClk, LOW);
        } else {
            digitalWrite(serialIn, LOW);
            digitalWrite(serialClk, HIGH);
        }
    }
}

```

```
        digitalWrite(serialClk, LOW);  
    }  
}  
}
```