



# SMSsenger Bag

Created by Becky Stern



<https://learn.adafruit.com/smssenger-bag>

Last updated on 2021-11-15 06:34:30 PM EST

# Table of Contents

<a href="#">Overview</a>	3
<a href="#">Circuit Diagram</a>	4
<a href="#">Assemble Circuit</a>	4
<a href="#">SMS Scroller Code</a>	6
<a href="#">Sew a Fabric Pouch</a>	9
<a href="#">Wear it!</a>	10

---

# Overview

Scroll text messages on your bag, jacket, or any other surface, using a Flex NeoPixel Matrix and Flora + Fona!

This guide will show you how to build and test your circuit, upload our sample code to get your project up and running, and then house your project in a fabric pouch that you can sew to a fabric substrate of your choice.

This is a fairly straightforward soldering project, but an intermediate project when you take into account the complexity of the code. If you're an ultra-beginner, this might not be the easiest "first project," but could easily be a great second or third.

Before you begin this project, take a look at the following prerequisite guides:

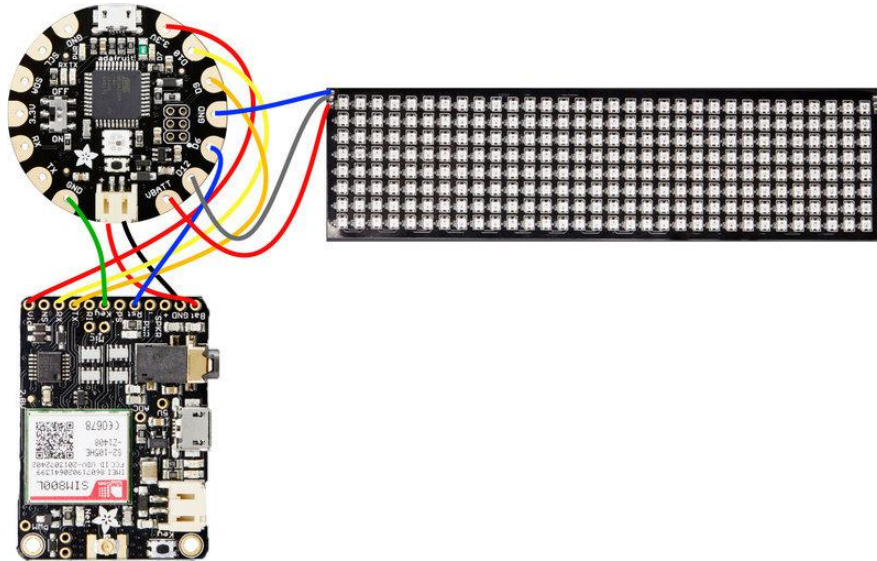
- [Getting Started with FLORA \(https://adafru.it/dwi\)](https://adafru.it/dwi)
- [Adafruit Fona \(https://adafru.it/iDO\)](https://adafru.it/iDO)
- [FLORA + FONA \(https://adafru.it/jAU\)](https://adafru.it/jAU)
- [NeoPixel Uberguide \(https://adafru.it/dhw\)](https://adafru.it/dhw)
- [Adafruit Guide to Excellent Soldering \(https://adafru.it/drl\)](https://adafru.it/drl)

To build this project, you will need the following parts and tools:

- [Flora microcontroller \(https://adafru.it/dgx\)](https://adafru.it/dgx)
- [Fona cellular module \(https://adafru.it/jAV\)](https://adafru.it/jAV) ([Fona 808 \(https://adafru.it/2542\)](https://adafru.it/2542) and [3G \(https://adafru.it/jAW\)](https://adafru.it/jAW) will also work) with [SIM card \(https://adafru.it/fbW\)](https://adafru.it/fbW)
- [GSM antenna \(https://adafru.it/fbL\)](https://adafru.it/fbL)
- [Flexible 8x32 NeoPixel Matrix \(https://adafru.it/jAX\)](https://adafru.it/jAX) ([dotstar matrices \(http://adafru.it/2736\)](http://adafru.it/2736) will work too with slight code and wiring adjustments)
- [2500mAh lipoly battery \(https://adafru.it/dVq\)](https://adafru.it/dVq)
- [26 gauge silicone-coated stranded wire in various colors \(https://adafru.it/egK\)](https://adafru.it/egK)
- [soldering iron and tools \(https://adafru.it/drl\)](https://adafru.it/drl)
- E6000 adhesive
- Velcro tape
- Sewing machine (optional but convenient)

---

## Circuit Diagram

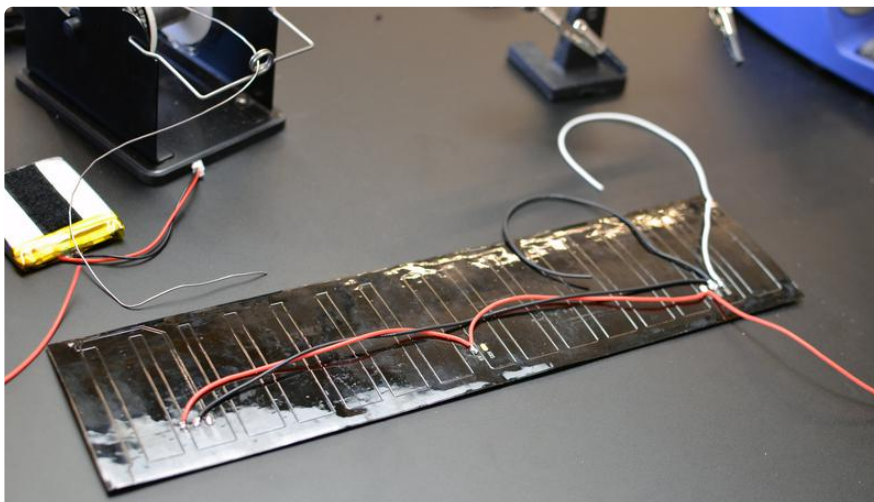


Wiring connections are as follows:

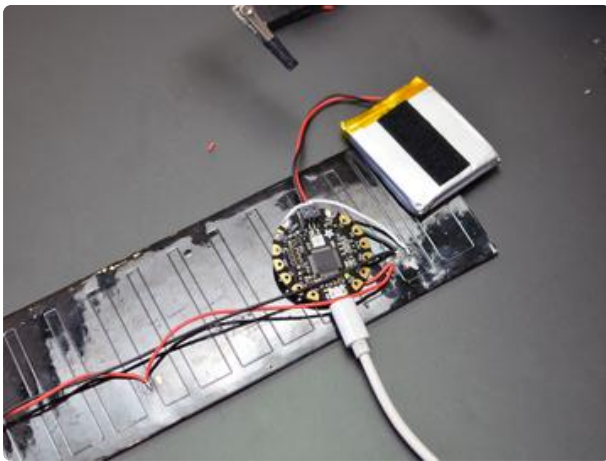
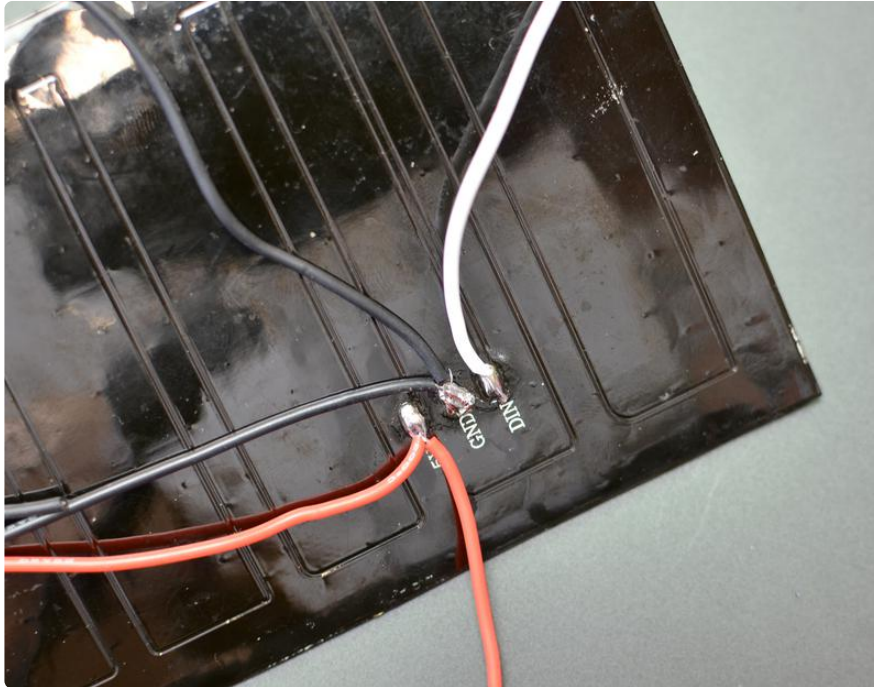
- Fona Vio -> Flora 3.3v
- Fona RX -> Flora D10 (configurable)
- Fona TX -> Flora D9 (configurable)
- Fona Key -> Flora GND
- Fona RST -> Flora D6 (configurable)
- NeoPixel matrix DIN -> Flora D12 (configurable)
- NeoPixel matrix 5V -> Flora VBATT
- NeoPixel matrix GND -> Flora GND

---

## Assemble Circuit

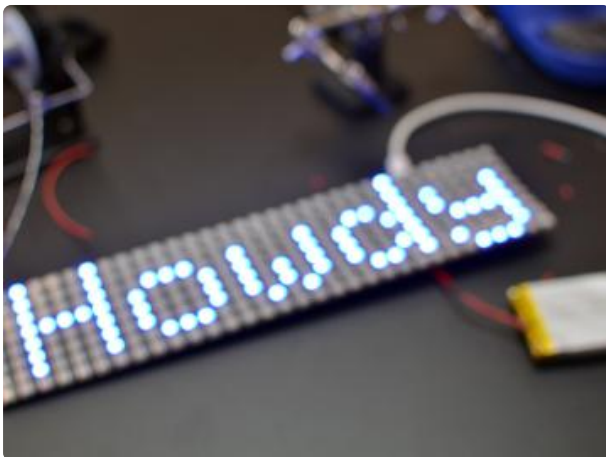


First up, solder sturdy wires to the NeoPixel Matrix. This thing draws a lot of current, so solid connections are an absolute must! To evenly distribute the power, connect up all the GNDs to each other, and likewise with the 5v pads. You can twist two pieces of wire together, tin the twist and the pad, then reheat both together to join. The input side will look like this:



Solder your free 5v connection to VBATT on Flora, and your spare GND connection to GND on Flora. The data input pin should be soldered to pin D12.

Load up the NeoPixel Matrix demo sketch, change PIN to 12, connect the battery, and make sure you see a message scrolling across the display before proceeding! It's much easier to troubleshoot problems incrementally than it is to wire everything up and try to identify a wiring error later.

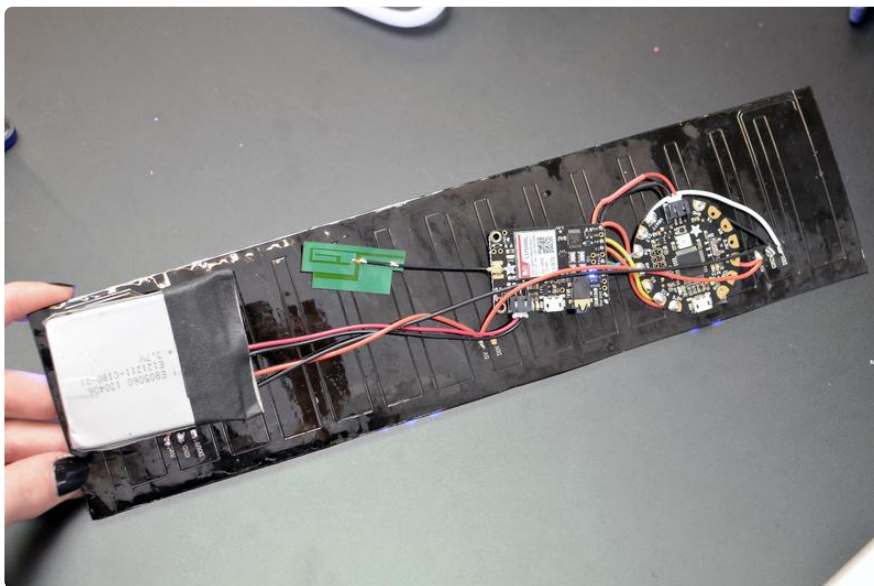


If you get errors when trying to upload to Flora, be sure you have the NeoMatrix, NeoPixel, Adafruit BusIO and Adafruit GFX libraries for Arduino installed, and that the right board and serial port are selected. You can install libraries directly from the Arduino app under Sketch -> Include Library -> Manage Libraries (Arduino 1.6.4 and above).

When your matrix is up and running, you can disconnect the battery and USB cord and protect the wire connections on the matrix with E6000 adhesive. While you're at it, squeeze some adhesive into the top of the battery, where the wires connect, for some strain relief. Allow glue to dry for 24 hours.

Next, solder up the connection to Fona according to the circuit diagram. Attach the GSM antenna to Fona, and the battery to Fona's JST port.

Add some Velcro tape to secure the battery. Your finished circuit should look something like this:



---

## SMS Scroller Code

Next its time to load up the SMS-receiving-and-displaying Arduino sketch, included below. Make sure you have the Fona library installed!

```
// Fona + Flora SMSsenger bag
// receives and displays text messages on a 8x32 flex NeoPixel matrix
// tutorial: https://learn.adafruit.com/flora-plus-fona

#include "Wire.h"
#include <Adafruit_GFX.h>;
#include <Adafruit_NeoMatrix.h>;
#include <Adafruit_NeoPixel.h>;
#include <SoftwareSerial.h>;
```

```

#include "Adafruit_FONA.h"

#define FONA_RX 10
#define FONA_TX 9
#define FONA_RST 6

#define PIN 12

// MATRIX DECLARATION:
// Parameter 1 = width of NeoPixel matrix
// Parameter 2 = height of matrix
// Parameter 3 = pin number (most are valid)
// Parameter 4 = matrix layout flags, add together as needed:
//   NEO_MATRIX_TOP, NEO_MATRIX_BOTTOM, NEO_MATRIX_LEFT, NEO_MATRIX_RIGHT:
//   Position of the FIRST LED in the matrix; pick two, e.g.
//   NEO_MATRIX_TOP + NEO_MATRIX_LEFT for the top-left corner.
//   NEO_MATRIX_ROWS, NEO_MATRIX_COLUMNS: LEDs are arranged in horizontal
//   rows or in vertical columns, respectively; pick one or the other.
//   NEO_MATRIX_PROGRESSIVE, NEO_MATRIX_ZIGZAG: all rows/columns proceed
//   in the same order, or alternate lines reverse direction; pick one.
//   See example below for these values in action.
// Parameter 5 = pixel type flags, add together as needed:
//   NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB    Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB    Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

// Example for NeoPixel Shield. In this application we'd like to use it
// as a 5x8 tall matrix, with the USB port positioned at the top of the
// Arduino. When held that way, the first pixel is at the top right, and
// lines are arranged in columns, progressive order. The shield uses
// 800 KHz (v2) pixels that expect GRB color data.
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(32, 8, PIN,
  NEO_MATRIX_BOTTOM + NEO_MATRIX_RIGHT +
  NEO_MATRIX_COLUMNS + NEO_MATRIX_ZIGZAG,
  NEO_GRB + NEO_KHZ800);
const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };

// this is a large buffer for replies
char replybuffer[255];
uint8_t msgLen = 0; // Empty message
int x = matrix.width();
int pass = 0; // variable we'll use to keep track of a cycling of colors
bool newMessageCheck = 1; // variable to toggle checking of SMSs after each message
scroll
uint8_t n; // variable for the number of SMSs

SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);

uint8_t readline(char *buff, uint8_t maxbuff, uint16_t timeout = 0);

unsigned long prevFrameTime = 0L; // For animation timing
#define FPS 20 // Scrolling speed

void setup() {
  Serial.begin(115200);
  Serial.println(F("FONA SMS LCD test"));
  Serial.println(F("Initializing...(May take 3 seconds)"));

  matrix.begin();
  matrix.setTextWrap(false);
  matrix.setBrightness(40);
  matrix.setTextColor(colors[0]);

  Serial.println(F("Checking FONA"));

```

```

// See if the FONA is responding
fonaSerial-&gt;begin(4800);
if (! fona.begin(*fonaSerial)) { // make it slow so its easy to read!
  //lcd.print("Not found! :(");
  Serial.println(F("Couldn't find FONA"));
  while (1);
}
Serial.println(F("FONA is OK"));

delay(1000);
}

void loop() {

  unsigned long t = millis(); // Current elapsed time, milliseconds.
  // millis() comparisons are used rather than delay() so that animation
  // speed is consistent regardless of message length & other factors.
  if (newMessageCheck == 1){
    while (fona.getNetworkStatus() != 1) {
      Serial.println(F("Waiting 4 signal"));
      Serial.print(".");
      delay(400);
    }
    Serial.println(F("Waiting for SMS"));
    int8_t smsnum = fona.getNumSMS();
    if (smsnum &lt; 0) {
      Serial.println(F("Could not read # SMS"));
      return;
    } else {
      Serial.print(smsnum);
      Serial.println(F(" SMS's on SIM card!"));
    }

    if (smsnum == 0) return;
    else n = smsnum; // there's an SMS! Set index to most recent message

    Serial.print(F("\n\rReading SMS #")); Serial.println(n);
    uint16_t len;
    if(!fona.readSMS(n, replybuffer, 250, &len)) {
      // Uh oh failed to read SMS, show error perhaps?
      Serial.println("failed to read SMS");
      //break;
    }
    if (len == 0) {
      Serial.println(F("[empty slot]"));
      n++;
      //continue;
    }
    msgLen = len;

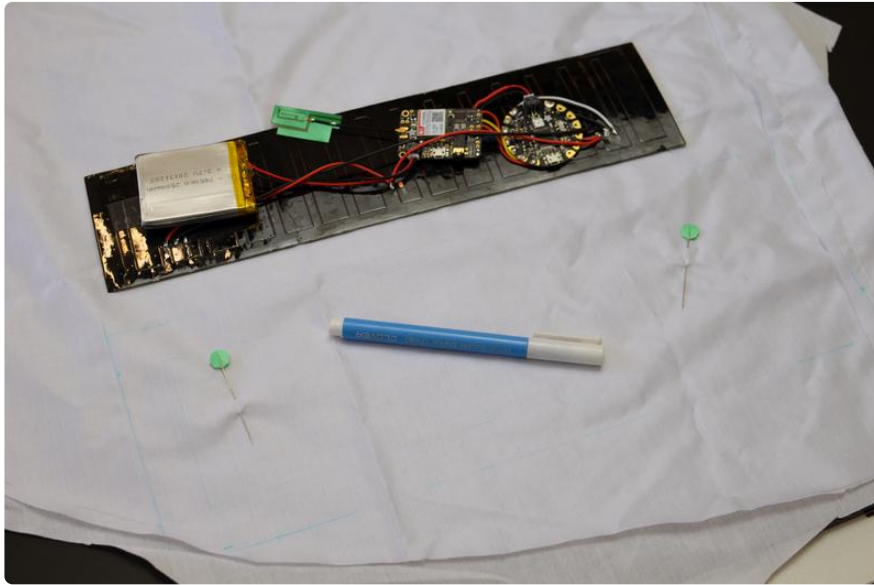
    Serial.print(F("***** SMS #")); Serial.print(n);
    Serial.print(" ("); Serial.print(len); Serial.println(F(") bytes *****"));
    Serial.println(replybuffer);
    Serial.println(F("*****"));
    newMessageCheck = 0;
  }
  if((t - prevFrameTime) &gt;= (1000L / FPS)) { // Handle scrolling
    matrix.fillScreen(0);
    matrix.setCursor(x, 0);
    matrix.print(replybuffer);
    if(--x &lt; (msgLen * -6)) {
      newMessageCheck = 1;
      x = matrix.width(); // We must repeat!
      if(++pass &gt;= 3) pass = 0;
      matrix.setTextColor(colors[pass]);
    }
  }
}

```

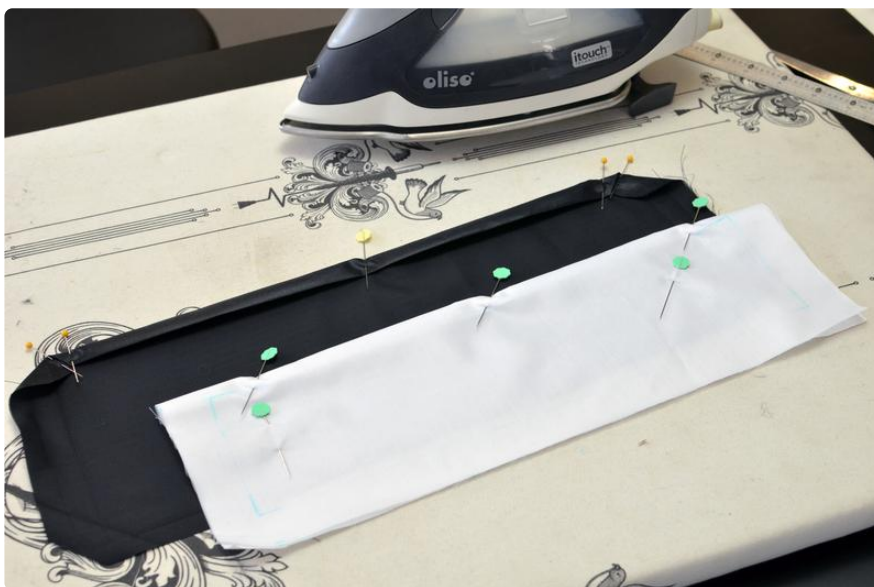


```
matrix.show();  
prevFrameTime = t;  
}  
  
}
```

## Sew a Fabric Pouch



Choose some fabric for the front diffusing part of the pouch-- it doesn't have to be white! Black material can also work well as long as it's translucent. Experiment, even bring your circuit to the fabric store! Trace the perimeter of your matrix onto the fabric with a disappearing marking pen, then draw a bigger rectangle that's ~1" larger in each direction than the first, for seam allowance.

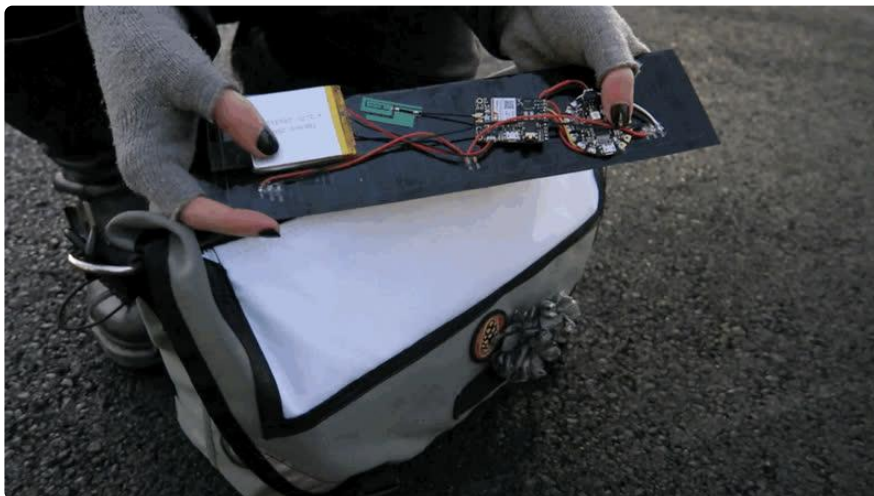


Cut out the front panel, and make a back panel that is slightly larger than the front. Fold over and iron the edges of the back panel around the raw edge of the front panel, and topstitch around three sides. Add velcro tape to the top edges and stitch the finished pouch to your bag, jacket, curtain, or any other fabric substrate you wish to adorn.



---

## Wear it!



When you power it up with Flora's onboard switch, the matrix will display the last SMS it received. Eventually your SIM will fill up, though, as this code does not delete any messages it receives. To clean up your SMSs, load up the FONAtest code example and use the serial interface to delete them.

These pixel matrices are nice and relatively flexible, but are not designed for repeated or extreme bending. If your application puts excessive bend force on the matrix, consider adding a cardboard or other stabilizing material to reduce the amount of bend.

Unless you sewed your pouch with waterproof fabric and sealed the seams, this project is not weatherproof! If you get caught in the rain, power down your circuit and stow it inside your bag.

