PixeLight

Bibiana Bauer

Programming & Electronics

Concept Explorations

Long-Exposure LED Light Painting Display
series of LED lights that blink on and off to reveal a picture or message with long-exposure photo

Visual Meditation Breathing Display
responsive display which shows the steadying and slowing of one's breathing during meditation

Tap-Dancing Keyboard (w/feet)
responsive surface which translates patterns of foot taps into letters or words

similar to Doug Engelbert's original keyboard design

3-D Painting w/holograms
drawing program which allows user to draw 3D shapes inside a box and then project them as 3-D holograms

Programmable Head Massager
device which scratches/massages the user's head based on chosen patterns and length of time

Motion-operated morse code transcriber
program which translates motion into morse code and then into written word

Live-feed Hologram Projection
3D projection of video from Skype or Facetime call

3D B-day Present Printer
a program that automatically prints a small gift for a friend w/upcoming b-day on FB

Project Proposal

Summary:

- An interactive lightly display which allows users to create and send messages using blinking LEDs and long-exposure photography.

Intentions:

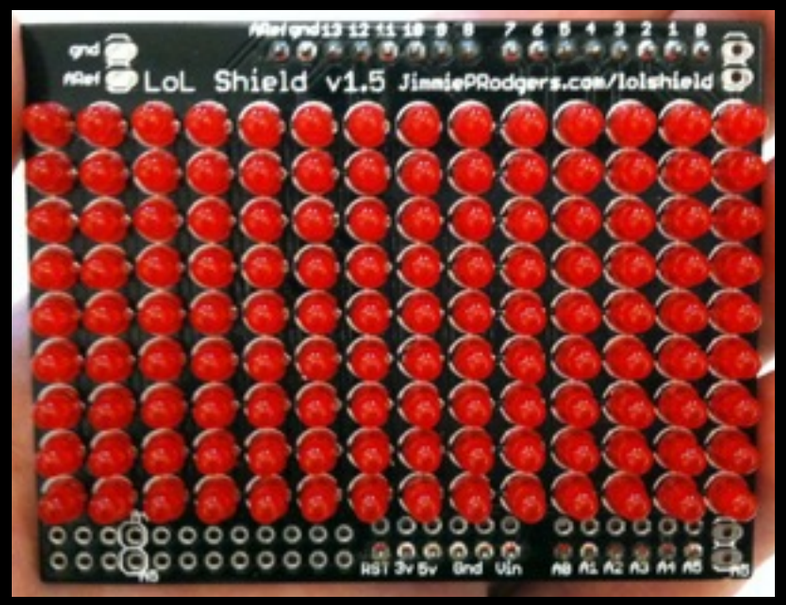
- To explore the subtleties of hidden communication: how the medium of a message can affect its content

- To promote an intimate experience/engagement within the creation and reception of a message

Inspiration - The PixelStick

http://www.thepixelstick.com/

Parts & Software



LoL Shield

Arduino Uno

Processing

Arduino

p5

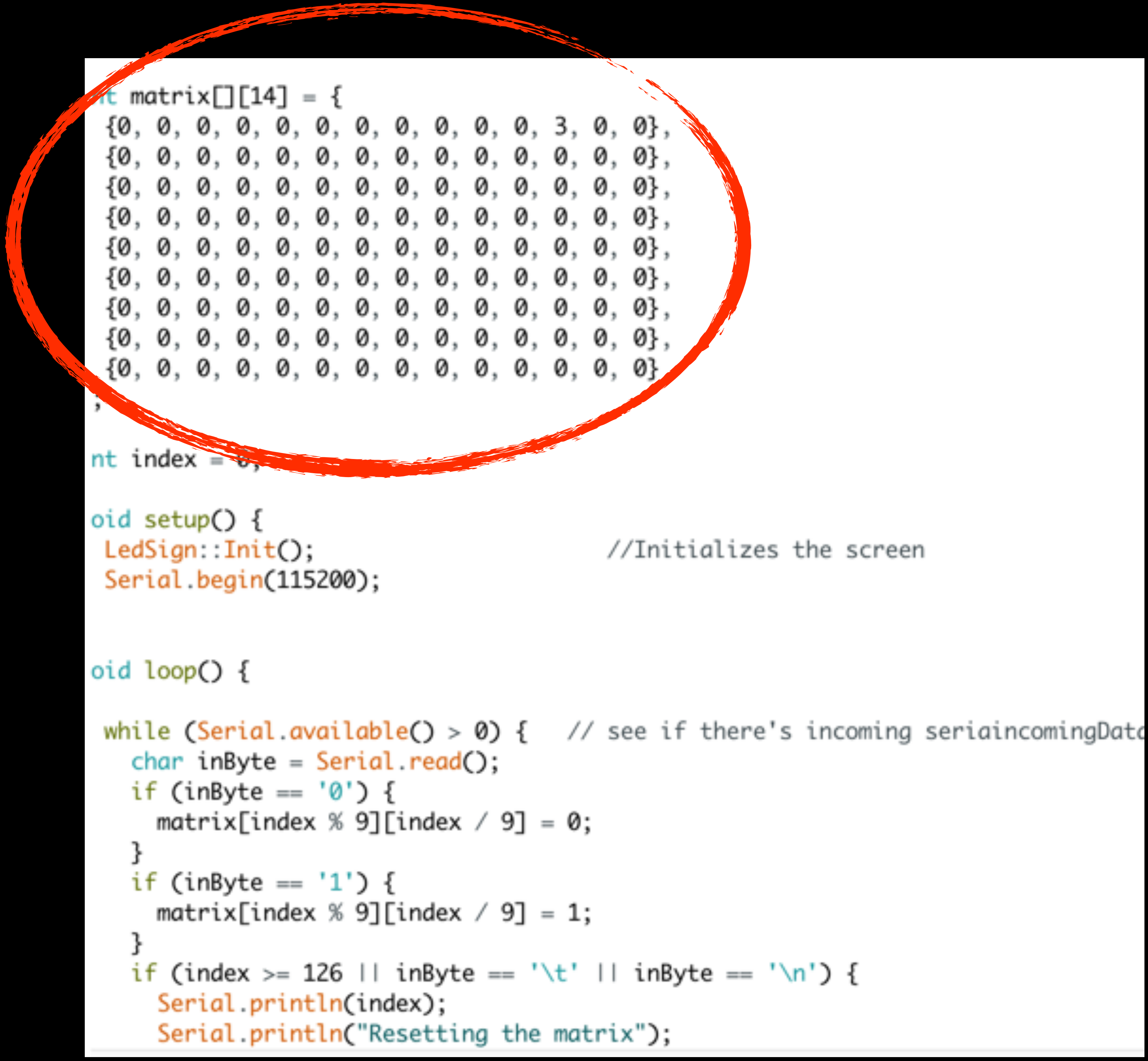
Original Arduino Code

- Tedious
- Inefficient

```
// ------------------------ LETTER "B"--------------------------

// left downstem of B
      {0, 1, 0, 0, 0, 0, 0, 0, 0},
      {0, 1, 0, 0, 0, 0, 0, 0, 0},
      {0, 0, 1, 0, 0, 0, 0, 0, 0},
      {0, 0, 0, 1, 0, 0, 0, 0, 0},
      {0, 0, 0, 0, 1, 0, 0, 0, 0},
      {0, 0, 0, 0, 0, 1, 0, 0, 0},
      {0, 0, 0, 0, 0, 0, 1, 0, 0},
      {0, 0, 0, 0, 0, 0, 0, 1, 0},

// top crosccbridge of B
      //{0, 1, 0, 0, 0, 0, 0, 0, 0},
      {0, 2, 0, 0, 0, 0, 0, 0, 0},
      {0, 4, 0, 0, 0, 0, 0, 0, 0},
      //{0, 8, 0, 0, 0, 0, 0, 0, 0},

// middle crossbridge of B
      //{0, 0, 0, 0, 1, 0, 0, 0, 0},
      {0, 0, 0, 0, 2, 0, 0, 0, 0},
      {0, 0, 0, 0, 4, 0, 0, 0, 0},
      //{0, 0, 0, 0, 8, 0, 0, 0, 0},

// bottom crossbridge of B
      //{0, 0, 0, 0, 0, 0, 0, 1, 0},
      {0, 0, 0, 0, 0, 0, 0, 2, 0},
      {0, 0, 0, 0, 0, 0, 0, 4, 0},
      //{0, 0, 0, 0, 0, 0, 0, 8, 0},

//right downstem of B
      {0, 8, 0, 0, 0, 0, 0, 0, 0},
      {0. 0. 8. 0. 0. 0. 0. 0. 0}.
```

Final Arduino Code

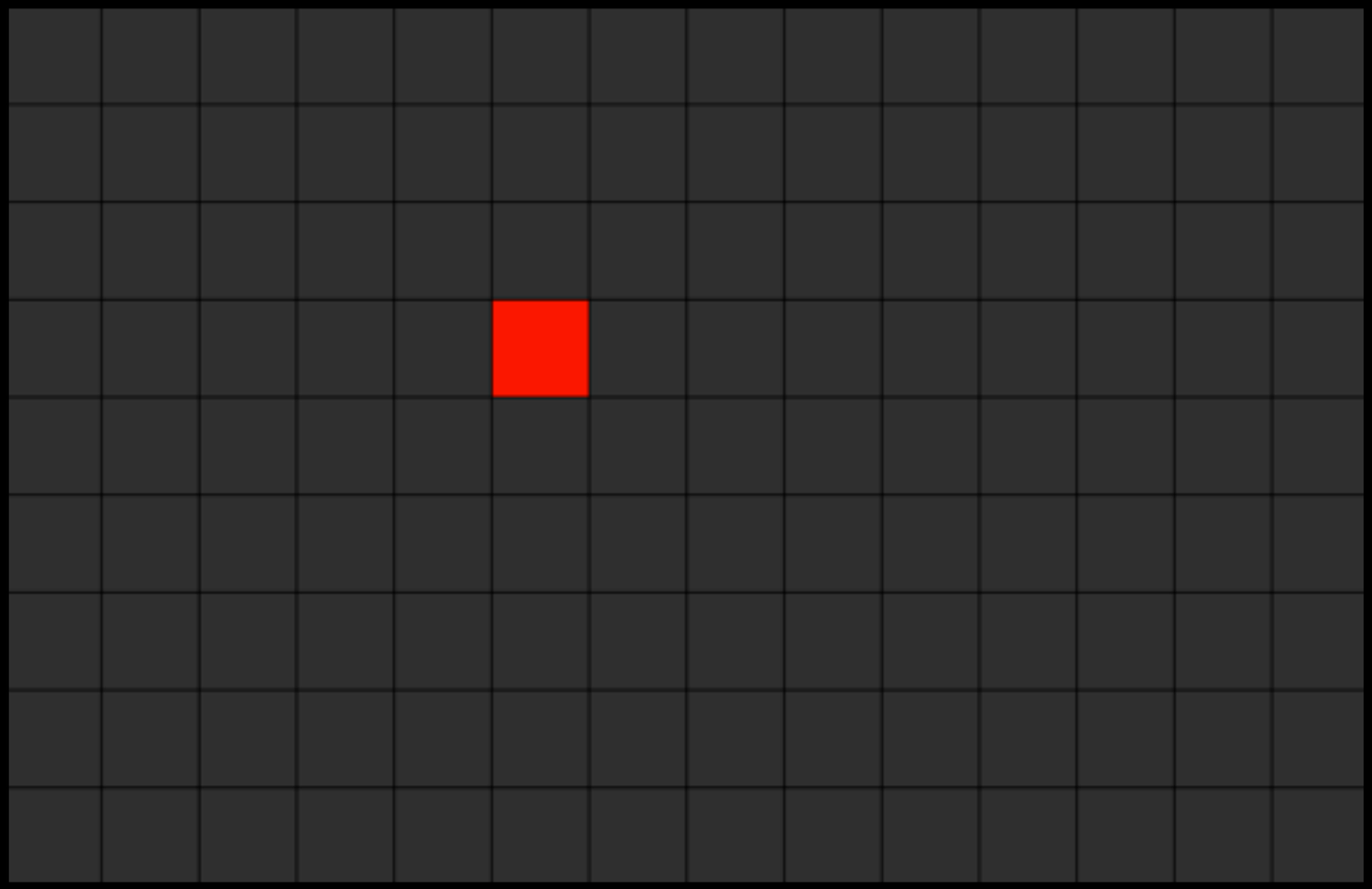
- Simpler
- Directly understandable



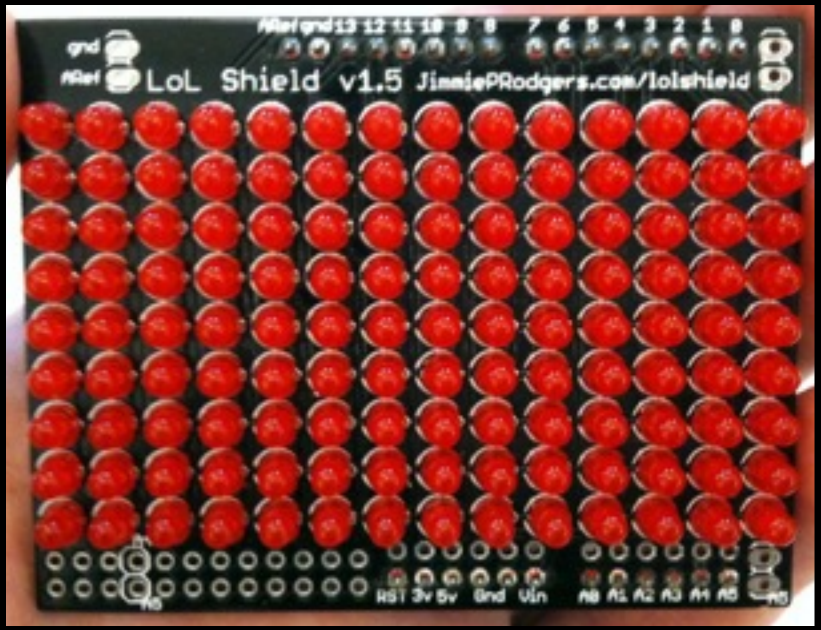
```
nt matrix[][14] = {
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
;

nt index = 0;

oid setup() {
 LedSign::Init();                        //Initializes the screen
 Serial.begin(115200);


oid loop() {

 while (Serial.available() > 0) {   // see if there's incoming seriaincomingDat
   char inByte = Serial.read();
   if (inByte == '0') {
     matrix[index % 9][index / 9] = 0;
   }
   if (inByte == '1') {
     matrix[index % 9][index / 9] = 1;
   }
   if (index >= 126 || inByte == '\t' || inByte == '\n') {
     Serial.println(index);
     Serial.println("Resetting the matrix");
```

P5 Interface

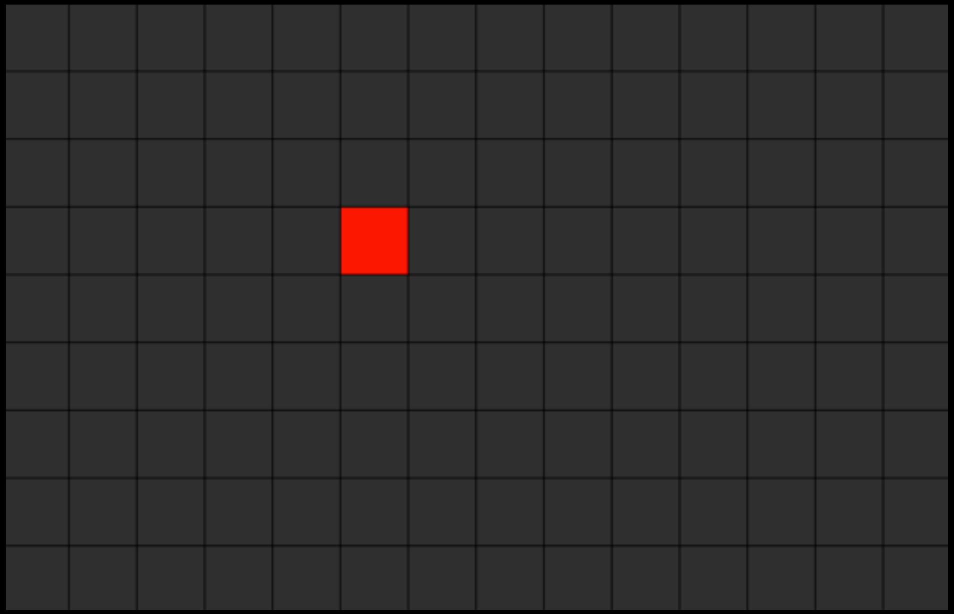
- Simple
- Easy-to-code



Analogous Systems



```
t matrix[][14] = {
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
;
```



What I Learned:

- How to use value statements for turning things on and off in Javascript
- How to connect a Javascript sketch to Arduino
- If you don’t have to build it yourself– DON’T

5 Most Difficult Problems

#1 Finding The Parts

- 150+ LEDS
- Corresponding resistors
- Female-to-female wires

Solution:

Use a pre-made LoL shield from the Hybrid Lab

#2 Decoding Charlieplexing

- The LoL shield came with its own Charlieplexing library, but it had very little corresponding explanation for how it actually worked or how to modify it without breaking it

Solution:

I played around with the sample code for long enough to figure out that each individual LED was controlled by numbers occurring in multiples of 2

#3 Responsive Buttons

- After coding the first prototype of the P5 Sketch, I could get each pixel to change color with a mouse hover, but I couldn’t get them to stay that color after a mouse click

Solution:

I integrated an array of pixelStatus values linked to the pixel color that would switch between True/False whenever a pixel was clicked

#4 Arduino to P5 Connection

- In order to get output values from the P5 sketch to be translated over as input values for the arduino code, I had to connect P5 to Arduino via a serial port. But it wouldn’t connect.

Solution:

Instead of using P5, I switched my Javascript code over into to Processing and connected it to the serial port that way instead.

#5 Rogue LEDs

- After testing my prototype quite thoroughly, I noticed that certain LEDs and matrix paths were either acting strangely or not functioning at all.

Solution:

I have yet to find a solution for this. It is possible that some of these malfunctions are an issue with the LoL Shield hardware, but others may be rooted in the charlieplexing library (which I don't yet have the expertise to manipulate at the current time)