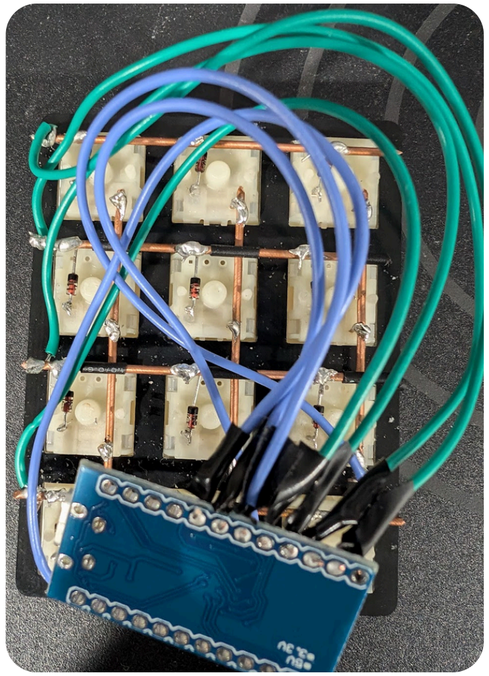
Handwired Keyboard

By: Annah Singh

EE459 Microprocessors

Introduction

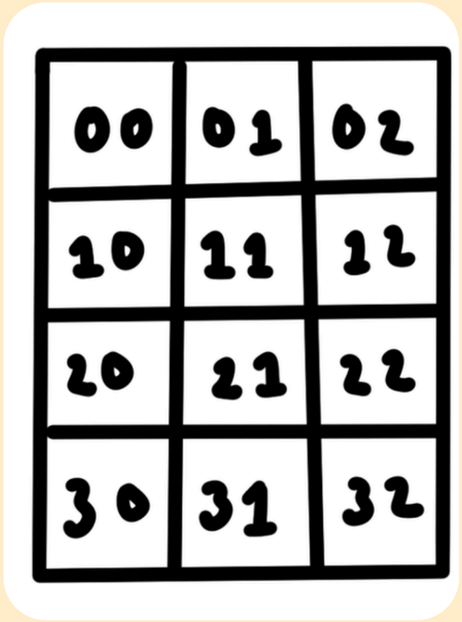
A handwired keyboard is a keyboard in which connections are made manually through soldering, and it is made functional through the use of a microcontroller.



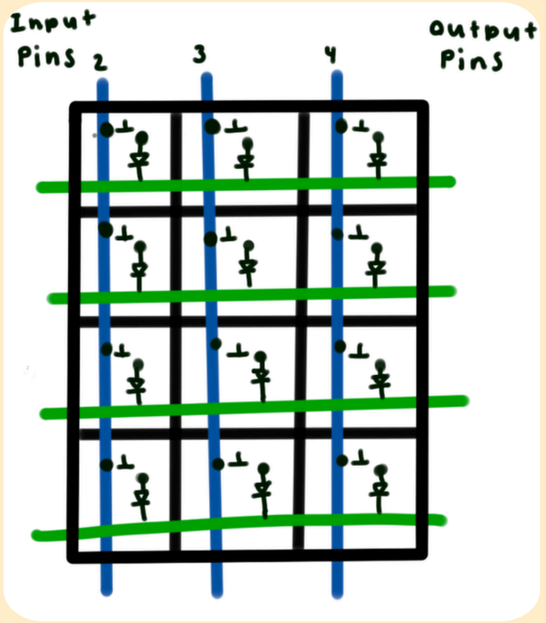
How it Works

- Each key follows its own address based on its row and column
- A key switch essentially connects two leads when pressed
- A diode is used to ensure that current doesn't flow into the second terminal of the switch, so each keypress is registered individually

Gateron KS-3X1



Keyboard Matrix



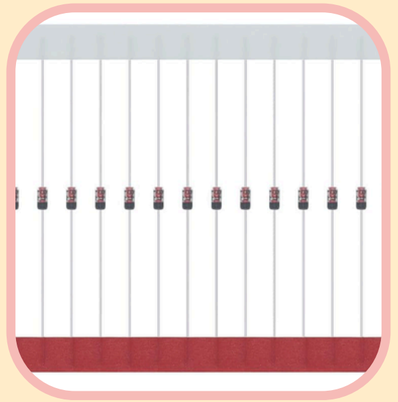
Schematic

Parts & Pricing

Material	Cost
Third Party Arduino Pro Micro (ATMega32U4)	$7.30
16 Gauge Copper Wire	$10
1N4148 Diodes	$6
Adafruit MacroPad Enclosure	$4.95
Total	$28.25

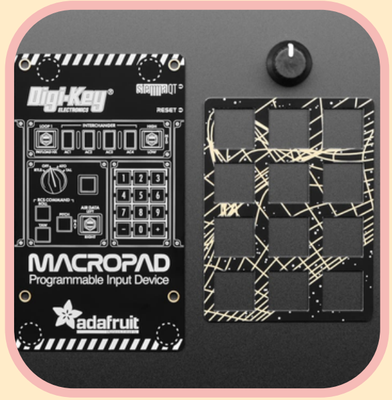
Materials Already Available:

- Switches, Keycaps, Heat Shrink, Electrical Tape

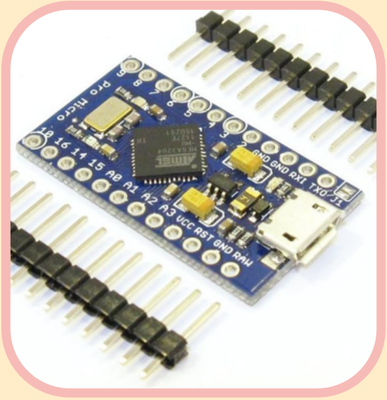


1N4148 Diodes

Copper Wire



Adafruit MacroPad



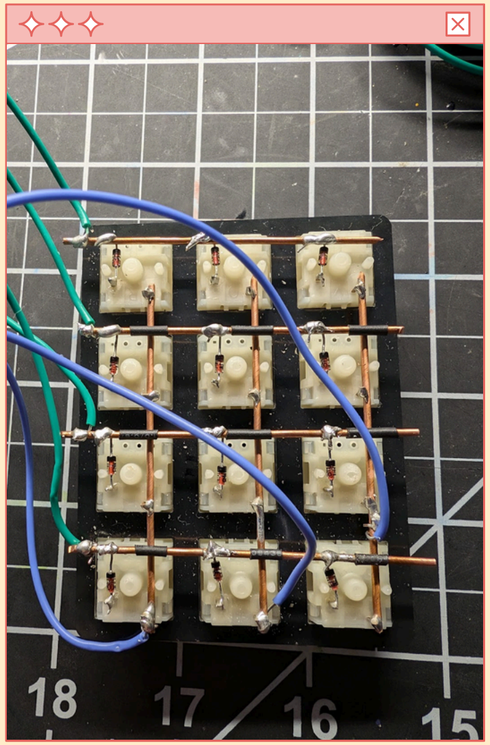
Third Party Arduino Pro Micro

Video Viewer

1080px(W) 1920px(H)

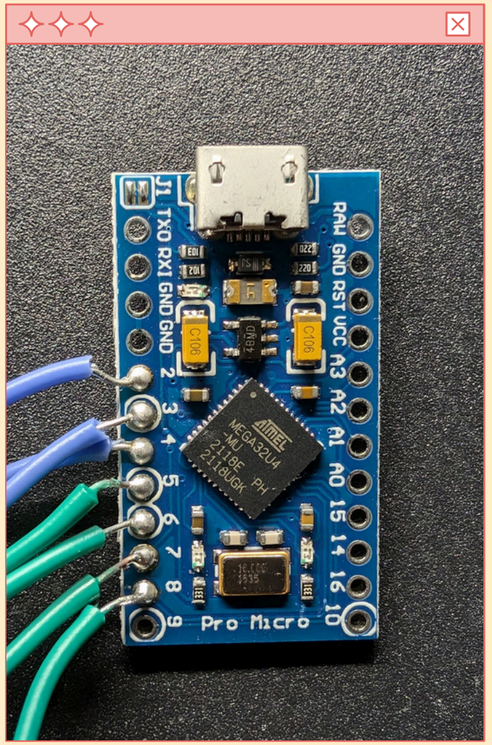
Building Stage

watch_this.mp4

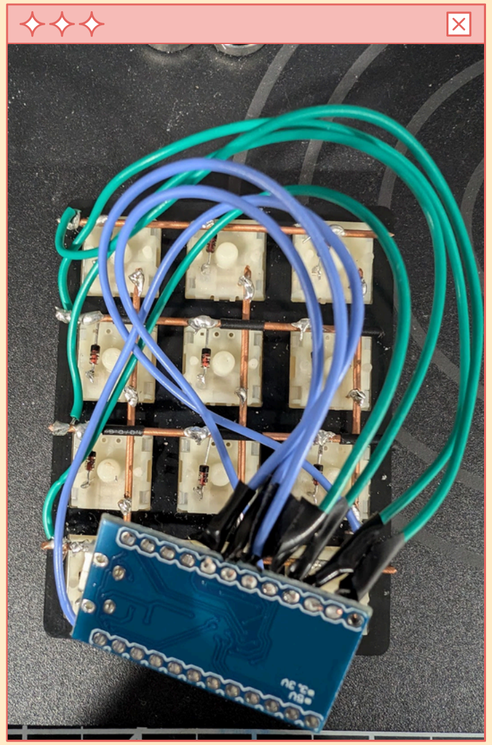


Solder Rows and Columns

Be sure to insulate where the columns and rows touch



Solder on Microcontroller



Complete

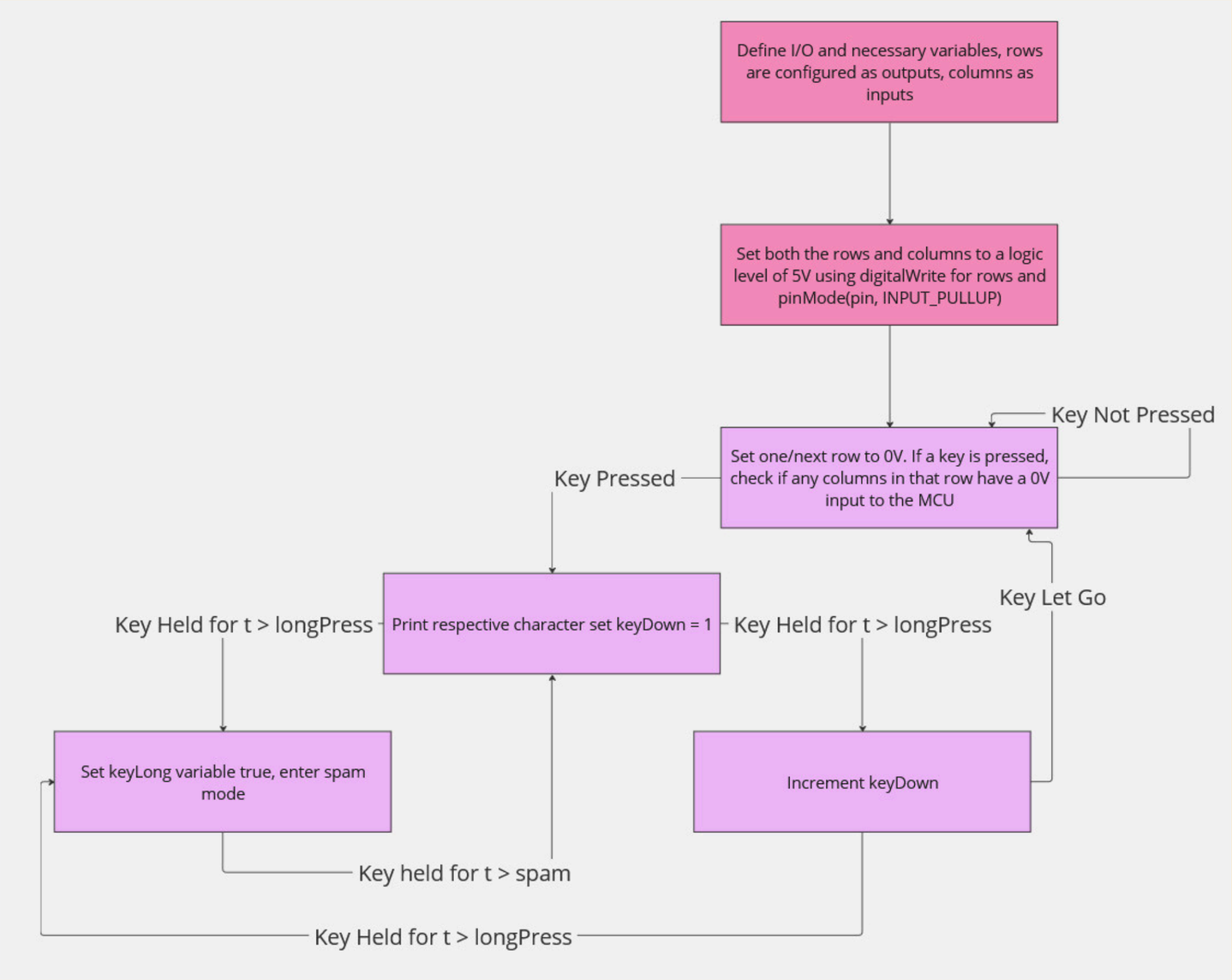
Software

```ino
#include <Keyboard.h>

byte inputs[] = {2,3,4}; //declaring inputs
byte outputs[] = {5,6,7,8};
int spam = 400; //spam speed
int longPress = 700; //delay
const int incount = 3;
const int outcount = 4;

char layout[4][3] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7','8','9'},
  {'0', 0xB0, 0xB2},
};
int keyDown[4][3];
int keyLong[4][3];

void setup() {
 for(int i = 0; i < outcount; i++)
 {
  pinMode(outputs[i], OUTPUT);
  digitalWrite(outputs[i], HIGH);
 }
 for(int i = 0; i < incount; i++)
 {
  pinMode(inputs[i], INPUT_PULLUP);
 }
 Serial.begin(9600);
 Serial.println("Connected");
 Keyboard.begin();

}
```

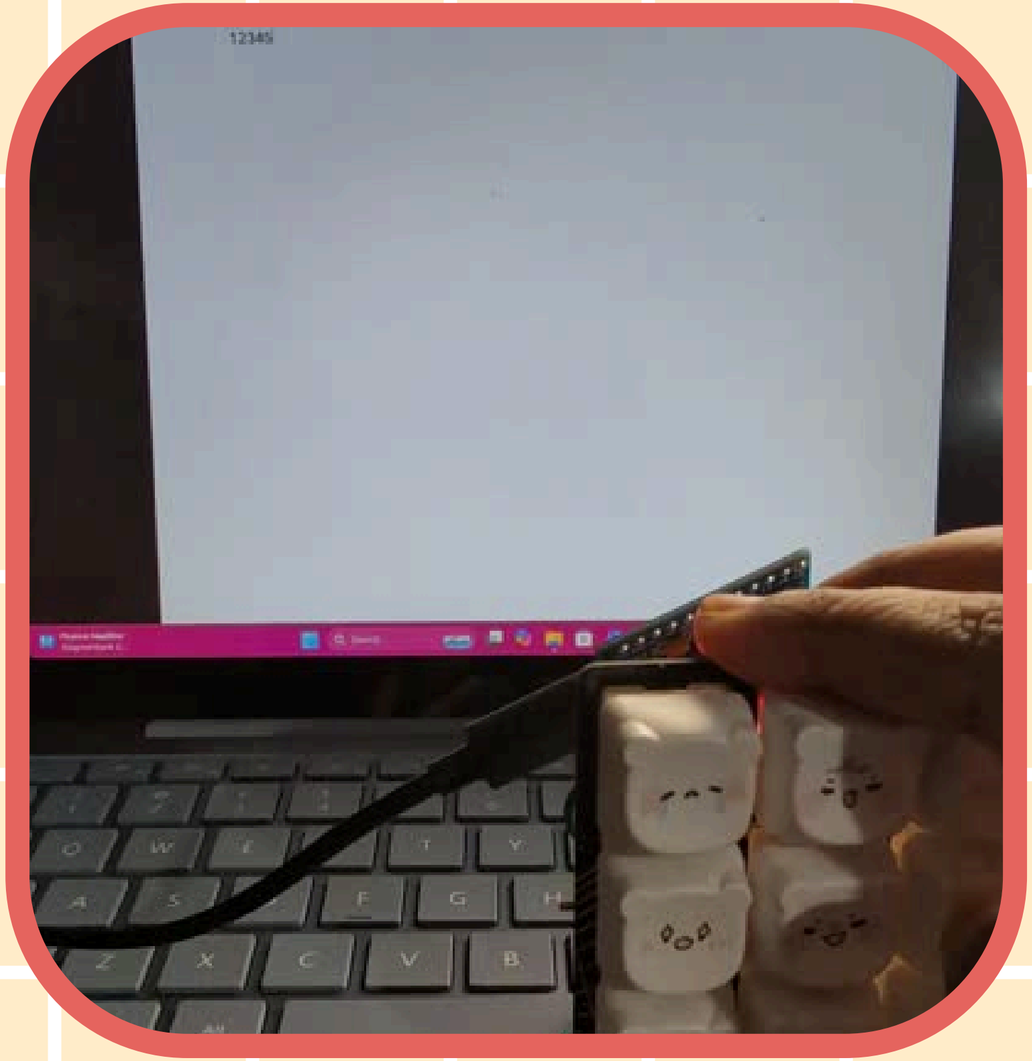
```ino
void loop() {
  // put your main code here, to run repe
  for (int i = 0; i < 4; i++)
  {
    digitalWrite(outputs[i], LOW);
    delayMicroseconds(5);

    for (int j = 0; j < 3; j++)
      {
      if(digitalRead(inputs[j]) == LOW)
      {
        keyPressed(i,j);
      }
      else if(keyDown[i][j] != 0)
      {
        resetKey(i,j);
      }
      }
    digitalWrite(outputs[i], HIGH);
    delayMicroseconds(500);
  }

}
```

```ino
oid keyPressed(int row, int col)

 Serial.print("Output: ");
 Serial.print(row);
 Serial.print("Input: ");
 Serial.print(col);
 Serial.println(layout[row][col]);

 if(keyDown[row][col] == 0)
 {
   Keyboard.write(layout[row][col]);
 }
 else if(keyLong[row][col] && keyDown[row][col] > spam)
 {
   Keyboard.write(layout[row][col]);
   keyDown[row][col] = 1;
 }
 else if(keyDown[row][col] > longPress)
 {
   keyLong[row][col] = true;
 }
 keyDown[row][col]++;


oid resetKey(int row, int col)

 keyDown[row][col] = 0;
 keyLong[row][col] = false;
```

Block Diagram



Demonstration



Difficulties Encountered

- Resoldering required
- Microcontroller must support USB usage as a Human Interface Device (HID)
  - An Arduino Uno will not work because of its USB capabilities

Improvements

- Casing
- Sound quality
- Design a Bluetooth Keyboard

Sources

https://www.youtube.com/watch?v=hjml-K-pV4E&t=1023s

https://www.youtube.com/watch?v=lq3oY91x9Vk&t=973s

https://www.amazon.com/gp/product/B01MTU9GOB/ref=ppx_yo_dt_b_search_asin_image?ie=UTF8&psc=1

https://www.amazon.com/gp/product/B0CJQ42CRK/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

https://www.amazon.com/gp/product/B0CKRMK45V/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1

https://www.amazon.com/gp/product/B0C2BW6GQX/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1