**Flex**

void setup() {

Serial.begin(9600);

pinMode(6, OUTPUT);}

void loop() {

int flex = analogRead(A0);

Serial.println(flex);

int adjust = constrain( map(flex, 15, 350, 0, 255), 0, 255);

analogWrite(6, adjust);}

**Switch Case**

int mode = 0;

int button = 2;

void setup() {

Serial.begin(9600);

pinMode(button, INPUT);}

void loop() { checkButton();

switch (mode) {

case 0: splash(); break;

case 1: mainMenu(); break;

case 2: about(); break;}}

void checkButton(){

int raw = digitalRead(button);

if (raw == 1) {

mode = (mode + 1) % 3; // % is "mod" and its the number of cases you have, makes it loop around

delay(250); //keeps it from reading button a bunch }}

void splash() {

Serial.println("Splash");}

void mainMenu() {

Serial.println("Main Menu");}

void about() {

Serial.println("About");}

**Screen**

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define OLED\_RESET 4

Adafruit\_SSD1306 display(OLED\_RESET);

void setup() {

Serial.begin(9600);

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);

display.clearDisplay();

display.display(); //takes things you told it to draw and actually draw them}

void loop() {

display.drawPixel(10, 10, WHITE);

display.drawCircle(60, 30, 20, WHITE);

display.setCursor(10, 45);

display.setTextColor(WHITE);

display.setTextSize(3);

display.print("Hello World");

display.drawBitmap(90, 0, umbrella, 32, 32, WHITE); // location, object, size (actual), color

display.display();}

**Temp**

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 2

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

void setup(){

Serial.begin(9600);

sensors.begin(); }

void loop() {

sensors.requestTemperatures();

float c = sensors.getTempCByIndex(0);

float f = (c \* (9.0/5.0)) + 32;

Serial.println(f);

delay(1000); }

330 ohm ( OOB)

4.7k (PYR > temp)

10k (OBB > sensors)

Button, Flex, Photo > V Divider

Screen dim: 128 x 64

Image2cpp: size has to be divisible by 8, black background, invert, scale to fit, keeping proportions, output: Adafruit GFXbitmap Font

**NeoPixel**

#include <Adafruit\_NeoPixel.h>

#ifdef \_\_AVR\_\_

#include <avr/power.h>

#endif

#define PIN 6

#define NUMPIXELS 3

Adafruit\_NeoPixel pixels = Adafruit\_NeoPixel(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

int delayval = 500;

void setup() {

#if defined (\_\_AVR\_ATtiny85\_\_)

if (F\_CPU == 16000000) clock\_prescale\_set(clock\_div\_1);

#endif

pixels.begin();}

void loop() {

pixels.setPixelColor(0, pixels.Color(0, 10, 0));

pixels.setPixelColor(1, pixels.Color(10, 0, 0));

pixels.setPixelColor(2, pixels.Color(0, 10, 0));

pixels.show();

delay(delayval);}

**GFX Library**

drawLine(x0, y0, x1, y1, color);

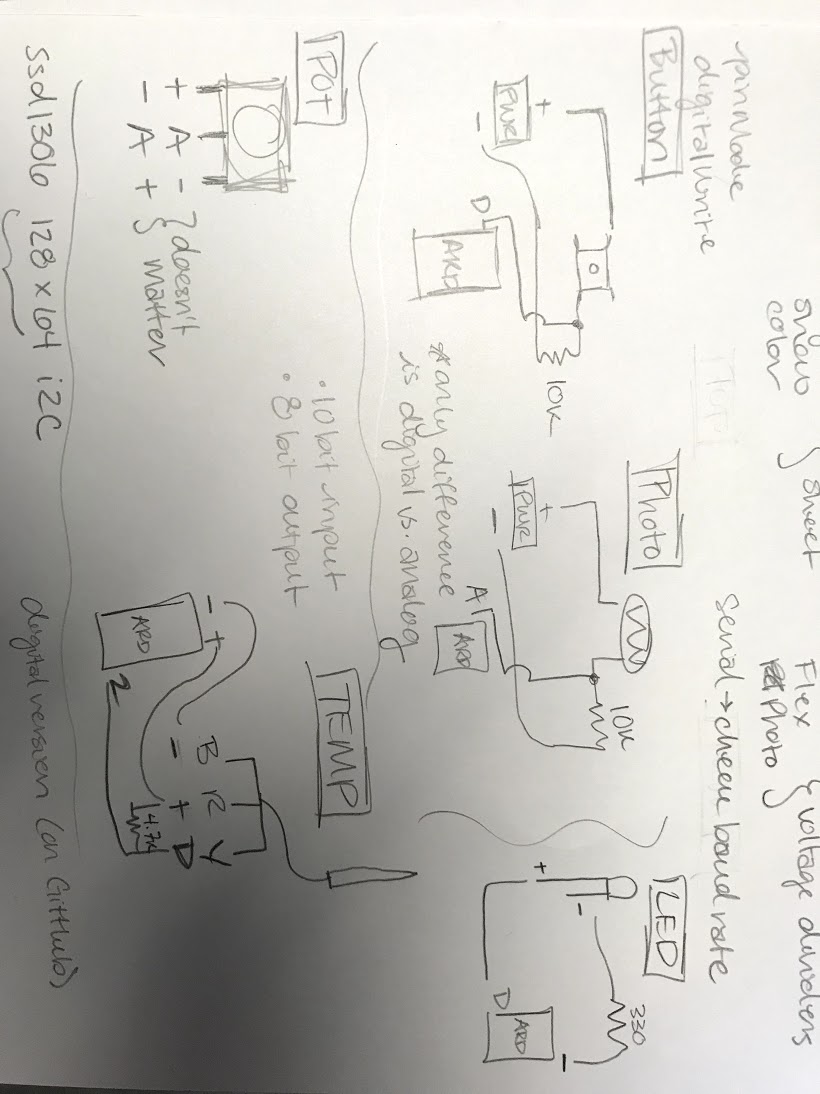
drawRect(x0, y0, w, h, color);

fillRect(x0, y0, w, h, color);

fillScreen(color);

drawTriangle(x0, y0, x1, y1, x2, y2, color);

fillTriangle(x0, y0, x1, y1 x2, y2, color);



**Tone**

const int buzzer = 9; //buzzer to arduino pin 9

void setup(){

pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output}

void loop(){

tone(buzzer, 1000); // Send 1KHz sound delay(1000); // ...for 1 sec

noTone(buzzer); // Stop sound...

delay(1000); // ...for 1sec}