Advanced Prototyping Final Study Sheet

Switch Cases

```
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 creexbutton()
int raw = digitalRead(button);
if(raw == 1){
mode = (mode + 1) % 3;
 delay(250);
void splash() {
   Serial.println("Splash");
void mainMenu() {
 Serial.println("Main Menu");
void about() {
 Serial.println("About");
```

Screen Elements

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit GFX.h>
 #include <Adafruit_SSD1306.h>
#define OLFD_RESET 4
Adafruit_SSD1306
display(OLED_RESET);
const unsigned char umbrella []
PROGMEM = {
    // 'umbrella, 32x32px
    0x00, 0x01, 0x80, 0x00, 0x00,
    0x01, 0x80, 0x00, 0x00,
    0x01, 0x80, 0x00, 0x01,
0x80, 0x00, 0x00, 0x01, 0x80,
0x00, 0x1f, 0xf0, 0x00, 0x00,
0xff, 0xfe, 0x00, 0x01, 0xff, 0xff,
0x80, 0x07, 0xff, 0xff, 0xc0,
0x0f, 0xff, 0xff, 0xe0, 0x1f, 0xff,
0xff, 0xf0, 0x3f, 0xff, 0xff, 0xf8,
0x3f, 0xff, 0xff, 0xfc,
0x7f, 0xff, 0xff, 0xfc,
0x7f, 0xff, 0xff, 0xfc, 0x7f, 0xff,
0xff, 0xfe, 0x7f, 0xff, 0xff, 0xfe,
0xff, 0xff, 0xff, 0xfe,
0xe0, 0xf8, 0x1e, 0x06, 0x80,
0x31, 0x88, 0x00, 0x00, 0x01,
0x80, 0x00, 0x00, 0x01, 0x80,
0x00, 0x01, 0x80, 0x00, 0x00, 0x00,
0x01. 0x80. 0x00. 0x00. 0x01.
0x80, 0x00, 0x00, 0x01, 0x80,
  0x00, 0x01, 0x80, 0x00, 0x00,
0x81, 0x80, 0x00, 0x01, 0xc1, 0x80, 0x00, 0x01, 0xc3, 0x80,
  UNUU UNCZ UNSU UNUU UNUU
0xff, 0x00, 0x00, 0x00, 0x7e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00,
0x00,
```

void setup() {
 Serial.begin(9600);

display.clearDisplay(): display.display();

void loop() { display.drawBitmap(0,0, umbrella, 32, 32, WHITE); // display.drawPixel(10, 10, WHITE);

WHITE);

BLACK):

display.begin(SSD1306 SWITCH CAPVCC, 0x3C); // Clear the

// display.drawCircle(80, 20, 10, WHITE);
// display.fillCircle(80, 20, 5,

// display.setCursor(25, 50); // display.setTextColor(WHITE,

// display.setTextSize(1); // display.print("Hello World");

display.display();

Map and Constrain

```
int led = 5:
void setup() {
// put your setup code here, to
run once:
 Serial.begin(9600):
 pinMode(led, OUTPUT);
void loop() {
    // put your main code here, to run repeatedly:
 int flex = analogRead(A1);
Serial.println(flex);
 flexled = map(flex, 50, 500, 0,
255);
flexled = constrain(flexled, 0,
255);
 analogWrite(led, flexled):
 delay(10);
```

Sensor Reader

```
int led = 5;
int flexled
 void setup() {
  // put your setup code here, to
  Serial.begin(9600);
pinMode(led, OUTPUT);
 void loop() {
 // put your main code here, to
run repeatedly:
int flex = analogRead(A1);
  Serial.println(flex):
```

flexled = map(flex, 100, 400, 0, 255); analogWrite(led, flexled):

Sensor Reader (temp)

#include <OneWire.h> #include <DallasTemperature.h> #define ONE_WIRE_BUS 2 OneWire oneWire(ONF WIRE BUS): DallasTemperature sensors(&oneWire);

void setup(){ Serial.begin(9600); sensors.begin();

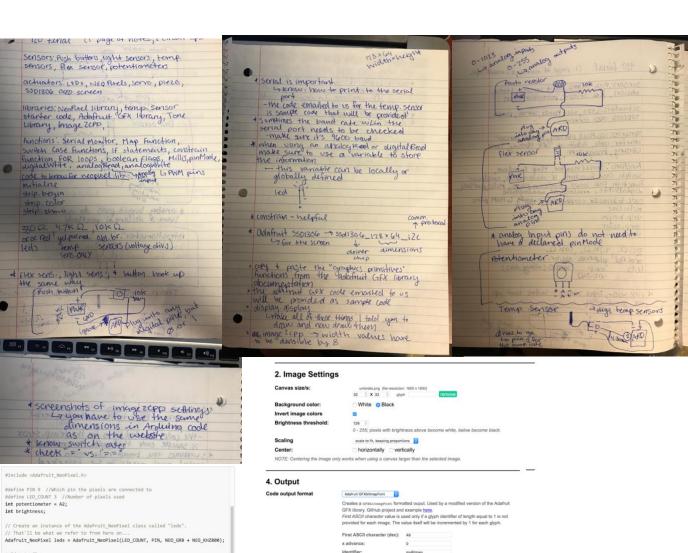
void loop() { sensors.requestTemperatures(); float c = sensors.getTempCByIndex(0); float f = (c * (9.0/5.0)) + 32; Serial.println(f); delay(1000);

Image to Cpp link

https://javl.github.io/image2cpp

NeoPixel Code

```
#include <Adafruit_NeoPixel.h>
#define PIN 6
#define LED_COUNT 3
Adafruit_NeoPixel leds = Adafruit_NeoPixel(LED_COUNT, PIN, NEO_GRB + NEO_KHZ800);
leds.begin();
leds.show();
Void loop()
leds.setPixelColor(0.255.0.0):
leds.setPixelColor(1,0,255,0);
leds.setPixelColor(2,0,0,255);
leds.show();
\\ or
Void setup()
leds.begin():
leds.show();
for(int x=0: x<LED_COUNT: x++)
leds.setPixelColor(x,255,255,0);
leds.show();
```



Refine PIN 9 //Which pin the pixels are connected to #### Bine LUD_COUNT 3 //Mamber of pixels used int potentioneter = A2;
int brightness;
// Create an instance of the Adafruit_NeoPixel class called "leds".
// That'll be what we refer to from here on...
Adafruit_NeoPixel leds = Adafruit_NeoPixel(LED_COUNT, PIN, NEO_GRB + NEO_NHISBOB)
void setup()
{
// Set the potentioneter as an IMPUT: pinMode(potentioneter, IMPUT);
// Serial communication so we can view readings from the potentioneter
Serial.begin(9680);
// Serial communication so we can view readings from the potentioneter
Serial.begin(9680);
// Since the color of the LEDs
// we can use a for loop in setup to set the color once
for(int xe0; xxLED_COUNT; x++)
{
 leds.setPixelColor(x, 255, 0, 0); //Set the pixel to RED
}
}
void loop();
// LEDs don't actually update until you call this.
}

void loop()
// Here we'll use the reading from the potentioneter to set the brightness
brightness = map(analogRead(potentioneter), 0, 1023, 0, 255);
leds.setOrightness(brightness); // Set the brightness of the LEDS
leds.setOrightness(brightness); // Set the brightness of the LEDS
leds.show(); // Display the LEDs

OLED: void draw?rest.viantic t. w. wintic t. eolor): //to draw i px

OLED: woid draw?ixel;wistl6 t x, unstl6 t y, uistl6 t color); //to draw 1 px
Exr display,drawPixel(5,5, WRITE);
display,drawFatVLise (6,9,9,x,1); color);
display,drawFatVLise (6,9,0, 1,eepth, color);
display,drawFatVLise (6,0,0, 1,eepth, color);
display,drawFatVLise (6,0,0, 1,eepth, color);
display,drawGatCle(64, 35, 25, MRITE); //format to draw object; x, y, r, color
display,drawGatGlec t ca also fill circle
display,drawGatGlec
display,drawGatGlec
display,drawGatGlec
display,drawGatGlec
totalColor totalColor totalColor (6,0,0,0,x1,y1,x2,y2,color);
display,drawGatGlec
display,drawGatGlec
totalColor totalColor totalColor (6,0,0,0,x1,y1,x2,y2,color);

Resistors: 330 (blue) [LEDs], 4.7k yellow purple red[temp sensor], 10k brown black orange [buttons]

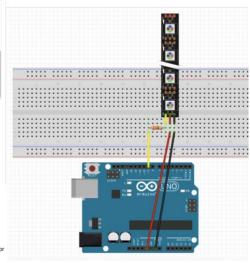
[don't use pin 0 or 1 for digital, it is for cereal]

Neopixel (digital): setup: strip.begin(); // initialize the strip strip.show(); // make sure it is visible strip.clear(); // initialize all pixels to 'off initialize, begin, show color, print Loop: strip.setPixelColor(I, r, g, b); / i = neopix to be called, 0-#heopix, or can use i++ func for all

strip.show(); DON'T FORGET Serial.begin(9600); in void setup to be able to print serially

NeoPixel programming circuit

P ×



Horizontal Vertical