330 LED   4.7k temp   10k Sensors

Only digital pins can be output and input
Analog pins can be only inputs - no need for pinMode
Digital is 0-255   Analog is 0-1023

Serial.begin(9600); // to check for errors

Photo flex button

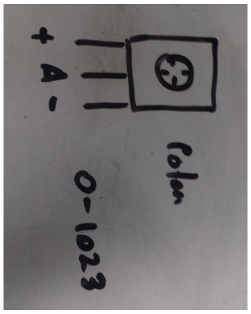
- Button - Any digital pin except 0, 1
- Flex and photo - analog pin

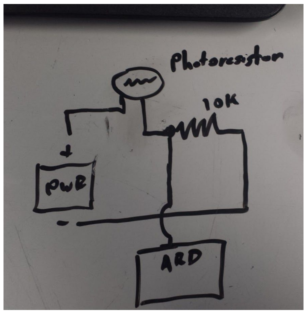
Pot temp

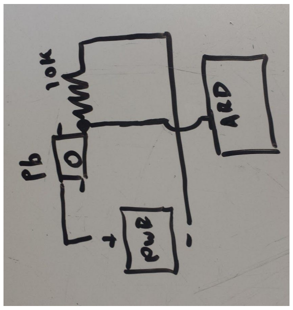
- Pot - - and + changes the rightmost or leftmost
    - No resistor because it already has one
- Temp - 4.7k
    - Digital pin 2 // Yellow is data

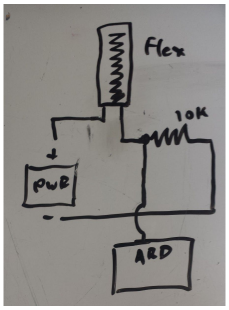
Mapping function will continue scaling

```
tone(pin, frequency, duration); //no library needed
noTone(duration);
```



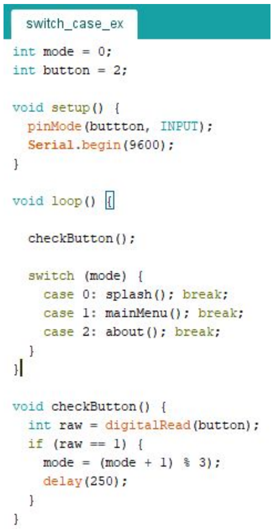




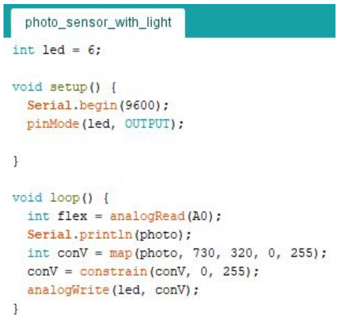


SCREEN:
Ground (-)   bcc (+)   scl   sda

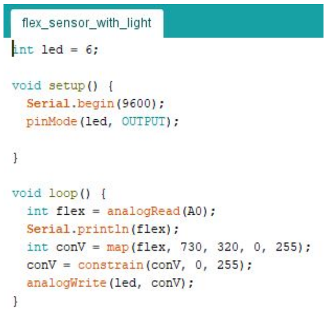
```
void drawPixel(uint16_t x, uint16_t y, uint16_t color);
void drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color);
void drawRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);
void fillRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t color);
void drawCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color);
void fillCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color);
void drawRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color);
void fillRoundRect(uint16_t x0, uint16_t y0, uint16_t w, uint16_t h, uint16_t radius, uint16_t color);
void drawTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color);
void fillTriangle(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t x2, uint16_t y2, uint16_t color);
void drawChar(uint16_t x, uint16_t y, char c, uint16_t color, uint16_t bg, uint8_t size);
void setCursor(uint16_t x0, uint16_t y0);
void setTextColor(uint16_t color);
void setTextColor(uint16_t color, uint16_t backgroundcolor);
void setTextSize(uint8_t size);
void setTextWrap(boolean w);
void drawBitmap(int16_t x, int16_t y, uint8_t *bitmap, int16_t w, int16_t h, uint16_t color);
void fillScreen(uint16_t color);
display.display();
```



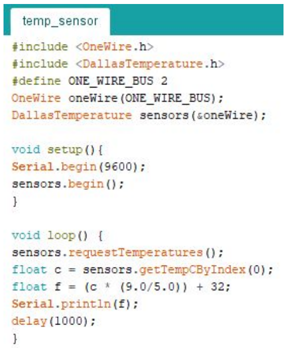
```
int mode = 0;
int button = 2;

void setup() {
  pinMode(buttton, INPUT);
  Serial.begin(9600);
}

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);
    delay(250);
  }
}
```



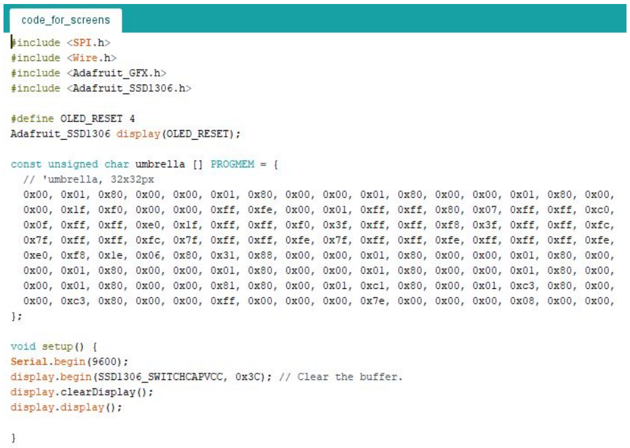
```
int led = 6;

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);

}

void loop() {
  int flex = analogRead(A0);
  Serial.println(photo);
  int conV = map(photo, 730, 320, 0, 255);
  conV = constrain(conV, 0, 255);
  analogWrite(led, conV);
}
```



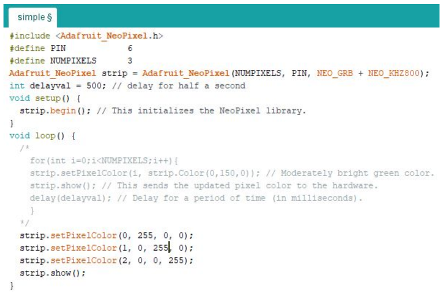
```
int led = 6;

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);

}

void loop() {
  int flex = analogRead(A0);
  Serial.println(flex);
  int conV = map(flex, 730, 320, 0, 255);
  conV = constrain(conV, 0, 255);
  analogWrite(led, conV);
}
```



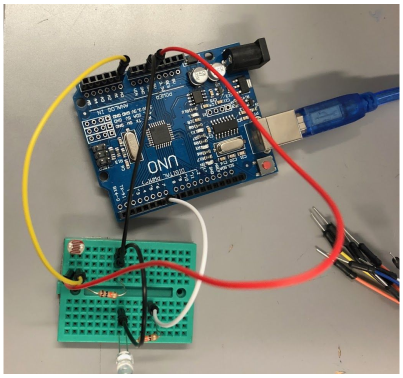
```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

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);
}
```

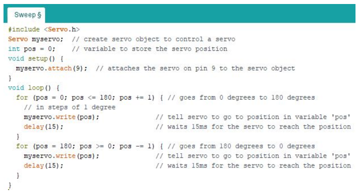


```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_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, 0x00, 0x01, 0x80, 0x00,
  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, 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,
  0x00, 0x01, 0x80, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00, 0x01, 0x80, 0x00,
  0x00, 0x01, 0x80, 0x00, 0x00, 0x81, 0x80, 0x00, 0x01, 0xc1, 0x80, 0x00, 0x01, 0xc3, 0x80, 0x00,
  0x00, 0xc3, 0x80, 0x00, 0x00, 0xff, 0x00, 0x00, 0x00, 0x7e, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00,
};

void setup() {
Serial.begin(9600);
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Clear the buffer.
display.clearDisplay();
display.display();

}
```

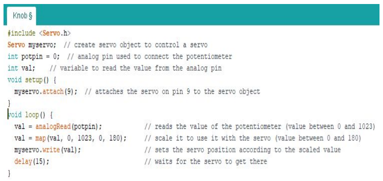


```
#include <Adafruit_NeoPixel.h>
#define PIN            6
#define NUMPIXELS      3
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
int delayval = 500; // delay for half a second
void setup() {
  strip.begin(); // This initializes the NeoPixel library.
}
void loop() {
  /*
    for(int i=0;i<NUMPIXELS;i++){
    strip.setPixelColor(i, strip.Color(0,150,0)); // Moderately bright green color.
    strip.show(); // This sends the updated pixel color to the hardware.
    delay(delayval); // Delay for a period of time (in milliseconds).
    }
  */
  strip.setPixelColor(0, 255, 0, 0);
  strip.setPixelColor(1, 0, 255, 0);
  strip.setPixelColor(2, 0, 0, 255);
  strip.show();
}
```





```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int pos = 0;    // variable to store the servo position
void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```



```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin
void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
void loop() {
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
```