

<https://goo.gl/cw56rD>

420 Final notes - 1 page, front back of notes. Include snippets of code to utilize during practical

Can upload notes to github so we can copy/paste code

Part 1: choose between two options of components input/output

Part 2: programming with a screen OLED-- get pic of a screen, we need to replicate it with graphics and text

Don't forget necessary **libraries**

When in doubt, try and use example code & **serial print** to find where it went wrong

Be sure to differentiate between local and global **variables**

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-#neopix, or can use `i++` func for all

`strip.show();`

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

Temp sensor: digital input

`analogWrite(5, 128);` --- cites pin5, ~½ power (0-255)

Circuitry

Photosensor (power -> ps -> analog, resistor & ground)

Flexsensor (power -> flexsen -> analog, resistor & ground)

For analog inputs, don't need to declare pinmode

OLED: `void drawPixel(uint16_t x, uint16_t y, uint16_t color);` //to draw 1 px

Ex: `display.drawPixel(5,5, WHITE);`

`display.drawLine (x0,y0,x1,y1, color);`

`display.drawFastVLine (x0, y0, length, color);`

`display.drawFastHLine (x0, y0, length, color);`

`display.drawCircle(64, 38, 20, WHITE);` //format to draw object: x, y, r, color

`display.fillRoundRect` ^can also fill circle

`display.drawRoundRect`

`Display. draw/fill triangle (x0, y0, x1, y1, x2, y2, color);`

`display.setRotation(uint8_t rotation);` (either 0-3 or 0, 90 etc

Secure | <https://javl.github.io/image2cpp/>

Apps Viking Web Student Work 123Movies Whose Line Berry Printing Track Records Te

2. Image Settings

Canvas size/s: umbrella.jpg (file resolution: 344 x 344)
 32 X 32 glyph *need width / 8 remove

Background color: ☐ White ☒ Black

Invert image colors ☒


Brightness threshold: 128
 0 - 255; pixels with brightness above become white, below become black

Scaling scale to fit, keeping proportions

Center: ☐ horizontally ☐ vertically

NOTE: Centering the image only works when using a canvas larger than the selected image.

3. Preview



4. Output

Code output format Adafruit GFXbitmapFont

--width needs to be divisible by 8--

display.drawBitmap(0,0, umbrella, 32, 32, WHITE);

^32, 32 need to be inputs from image2CPP

