

Understanding Sensors and Data Analysis Using the Arduino Nano 33 BLE Sense

DAY 2 : Setting Up an Arduino Nano 33 BLE Sense Development Environment

Sponsored by

DigiKey

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

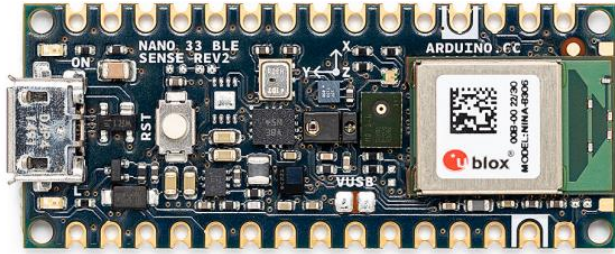


Dr. Don Wilcher

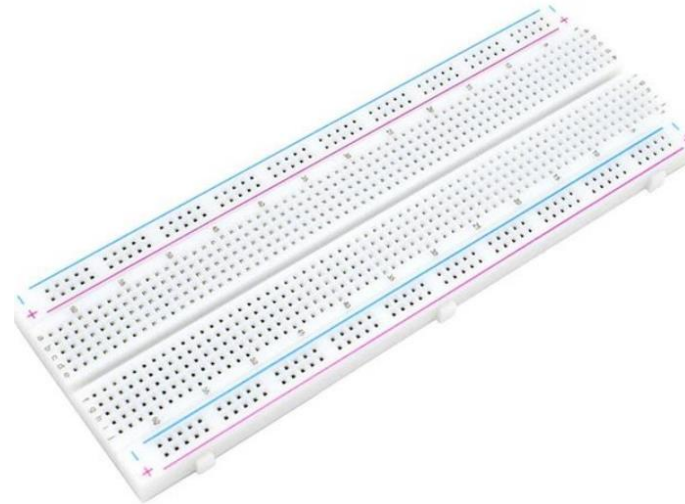
Visit 'Lecturer Profile' in your console for more details.

Course Kit and Materials

Arduino Nano 33 BLE Sense Board



Solderless Breadboard



Adafruit Parts Pal Kit

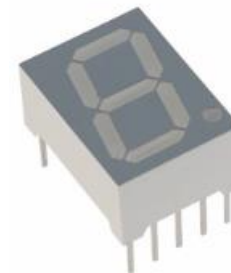


I2C OLED Display



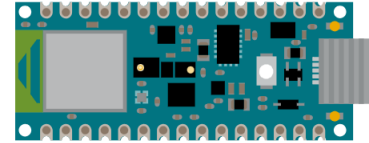
https://www.amazon.com/HiLetgo-Serial-128X64-Display-Color/dp/B06XRBYJR8/ref=sr_1_6?crid=1VC2UTZ2P8NWF&keywords=i2c%2Boled&qid=1700192985&srefix=I2C%2B%2Caps%2C108&sr=8-6&th=1

7 Segment LED Display, Common Cathode

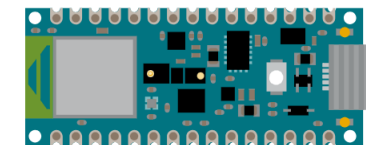


Agenda:

- Setting Up The Libraries
 - a) Adafruit GFX
 - b) Adafruit SSD1306
- Mini Labs: Hands-On Activities
 - a) Blinking LED
 - b) Pushbutton-Alternating Blinking LED
 - c) Arduino IDE Serial Plotter
- Lab: Analog Sensor-I2C OLED Demonstrator (Bricolage)

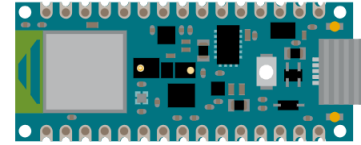


Research Perspective



“The Internet of Things, which has been quietly building and evolving over the past decade, now impacts many aspects of society.” (Chua & Storey, 2023).

Setting Up The Libraries



- Libraries are files written in C or C++ (.c, :cpp), providing extra functionality for your code. Examples include the following abilities.
 - a) controlling LEDs
 - b) servo motors
 - c) encoders
- To use existing libraries in the Arduino IDE, the Library Manager can be initiated.
- The Library Manager allows obtaining and installing libraries with ease.
- The **#include** statement is used to gain access to the public functions and constants defined by the library.

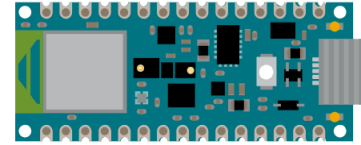
Question 1

What statement is used to gain access to the public functions and constants defined by the library?

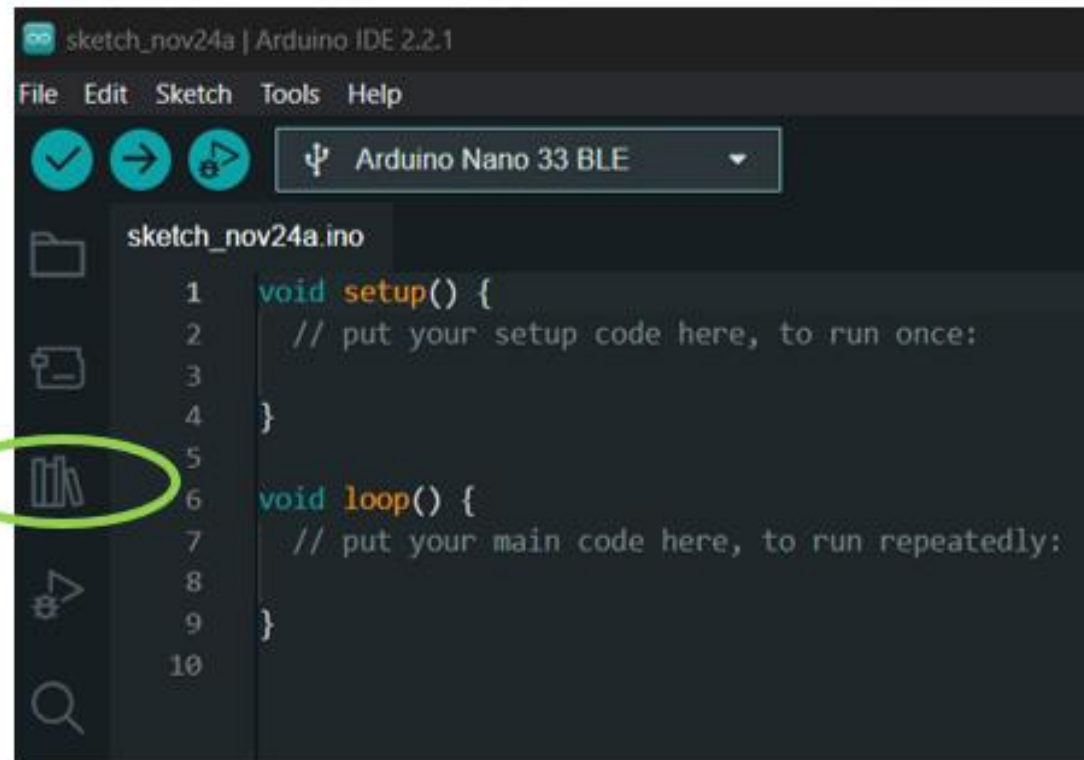
- a) **##include**
- b) **#Include**
- c) **#included**
- d) **#include**



Setting Up The Libraries...

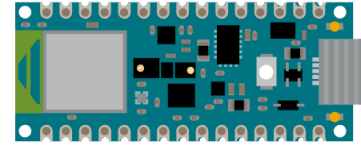


Accessing the
Library Manager



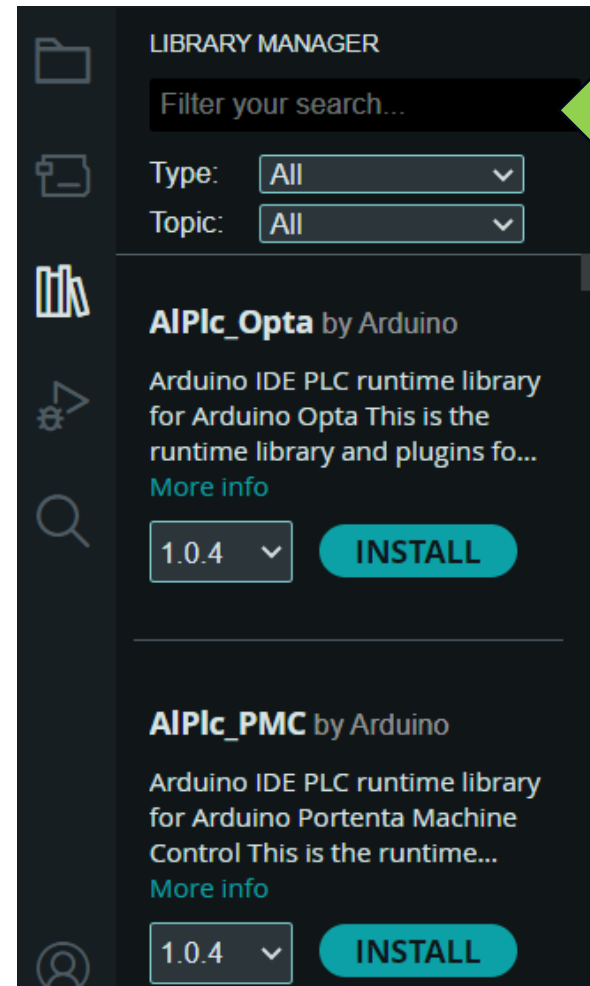
```
sketch_nov24a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
[Icons] Arduino Nano 33 BLE
sketch_nov24a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

Setting Up The Libraries...



The libraries that will need to be installed are listed below.

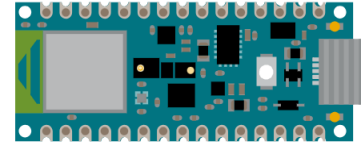
- a) Adafruit GFX
- b) Adafruit SSD1306



Type your Library searches here

Setting Up The Libraries. . .

Reference > Libraries > Adafruit gfx library



Adafruit GFX Library

Display

Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.

Author: Adafruit

Maintainer: Adafruit

[Read the documentation](#)

Compatibility

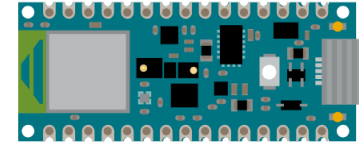
This library is compatible with **all** architectures so you should be able to use it on all the Arduino boards.

Releases

To use this library, open the [Library Manager](#) in the Arduino IDE and install it from there.

<https://www.arduino.cc/reference/en/libraries/adafruit-gfx-library/>

Setting Up The Libraries...



Github provides additional documentation for the library.

<https://github.com/adafruit/Adafruit-GFX-Library>

★ Adafuit-GFX-Library Public

Watch 157 Fork 1.5k Star 2.2k

master 12 branches 92 tags

Go to file Add file Code

About

Adafuit GFX graphics core Arduino library, this is the 'core' class that all our other graphics libraries derive from

learn.adafruit.com/adafruit-gfx-graphics-...

arduino-library

Readme

View license

Activity

2.2k stars

157 watching

1.5k forks

Report repository

Releases 92

1.11.9 - Remove volatile variable ... Latest on Oct 10

+ 91 releases

Packages

No packages published

Contributors 60

+ 49 contributors

File/Folder	Description	Time Ago
.github	Update CI action versions	6 months ago
Fonts	Fix missing imports in font files for PlatformIO (#423)	5 months ago
examples	fixed fast line drawing in GFXcanvas* classes	3 years ago
fontconvert	Merge pull request #268 from BlueAndi/master	2 years ago
.gitignore	doxy gitignore	5 years ago
Adafuit_GFX.cpp	Fix bug w/large GFXcanvas1's (use 32-bit byte counts)	10 months ago
Adafuit_GFX.h	Doxy class members moved from private to protected	10 months ago
Adafuit_GrayOLED.cpp	Gray oled support (#317)	3 years ago
Adafuit_GrayOLED.h	Gray oled support (#317)	3 years ago
Adafuit_SPITFT.cpp	clang	last month
Adafuit_SPITFT.h	Add dmaBusy()	last year
Adafuit_SPITFT_Macros.h	WIP do not use yet!	4 years ago
CMakeLists.txt	Enable Adafuit-GFX-Library as ESP-IDF component support (#376)	last year
README.md	Add more roadmap notes to README	2 years ago
component.mk	Enable Adafuit-GFX-Library as ESP-IDF component support (#376)	last year
gfxfont.h	Update gfxfont.h (#286)	3 years ago
glcdfont.c	Actionsci2 (#264)	4 years ago
library.properties	Update library.properties	last month
license.txt	Fixed constructor() kludge - all subclass libs updated too	10 years ago

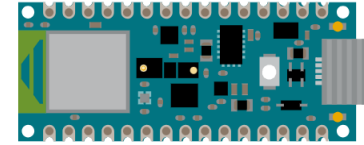
Question 2

In reviewing the Adafruit GFX library on slide 11, the Read the document link provides additional information using what platform?

- a) Blog**
- b) word document**
- c) Github**
- d) Wiki**



Setting Up The Libraries. . .



Reference > Libraries > Adafruit ssd1306

Adafruit SSD1306

Display

SSD1306 oled driver library for monochrome 128x64 and 128x32 displays

Author: Adafruit

Maintainer: Adafruit

[Read the documentation](#)

Compatibility

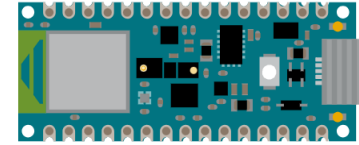
This library is compatible with **all** architectures so you should be able to use it on all the Arduino boards.

Releases

To use this library, open the [Library Manager](#) in the Arduino IDE and install it from there.

<https://www.arduino.cc/reference/en/libraries/adafruit-ssd1306/>

Setting Up The Libraries...



Github provides additional documentation for the library.

https://github.com/adafruit/Adafruit_SSD1306

Adafruit_SSD1306 Public

Watch 130 Fork 944 Star 1.6k

master 4 branches 42 tags

Go to file Add file Code

tyeth Merge pull request #270 from tyeth/bump-version-2023-11-14-20-33-27 7a4d33e last week 252 commits

- .github Update CI action versions 6 months ago
- examples no default reset for i2c last year
- scripts Remove trailing whitespace. 2 years ago
- .gitignore Add .gitignore 5 years ago
- Adafruit_SSD1306.cpp Fix up -Wshadow "error: declaration of 'spi' shadows a member of 'Ada... last year
- Adafruit_SSD1306.h add no splash def last year
- CMakeLists.txt Update CMakeLists.txt last year
- README.md Add splash opt-out. 2 years ago
- component.mk Add files via upload last year
- library.properties Bump version number to 2.5.9 2 weeks ago
- license.txt read me! 11 years ago
- splash.h Regenerate splash.h 2 years ago

README.md

Adafruit_SSD1306

Arduino Library CI passing Docs Doxygen

This is a library for our Monochrome OLEDs based on SSD1306 drivers

Pick one up today in the adafruit shop! -----> http://www.adafruit.com/category/63_98

These displays use I2C or SPI to communicate, 2 to 5 pins are required to interface.

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

About

Arduino library for SSD1306 monochrome 128x64 and 128x32 OLEDs

www.adafruit.com/category/63_98

arduino-library

Readme

View license

Activity

1.6k stars

130 watching

944 forks

Report repository

Releases 42

2.5.9 Bump version in library.pr... Latest last week

+ 41 releases

Packages

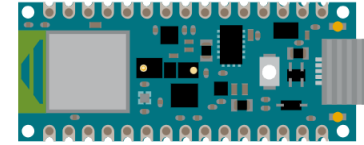
No packages published

Contributors 36

+ 25 contributors

Mini Labs: Hands-On Activities

Introduction:

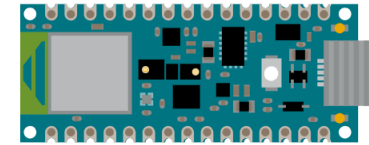


- The following Mini Labs will introduce some basic digital and analog input/output (I/O) capabilities of the Arduino Nano 33 BLE Sense.
- The Mini Labs will allow exploring the use of Bard AI to assist in generating the code for the mini labs.
- An approach to capturing data for analysis will be presented.
- A basic Arduino example will be enhanced with the assistance of Bard AI.
- The Mini Labs will consist of the following demonstrators.
 - a) Blinking LED
 - b) Pushbutton – Alternating Blinking LED
 - c) Arduino IDE – Serial Plotter
- Functional Overview Pinout will assist in the wiring of the Mini Lab circuits on a solderless breadboard.

Mini Labs: Hands-On Activities

Introduction:

Functional Overview



Pinout

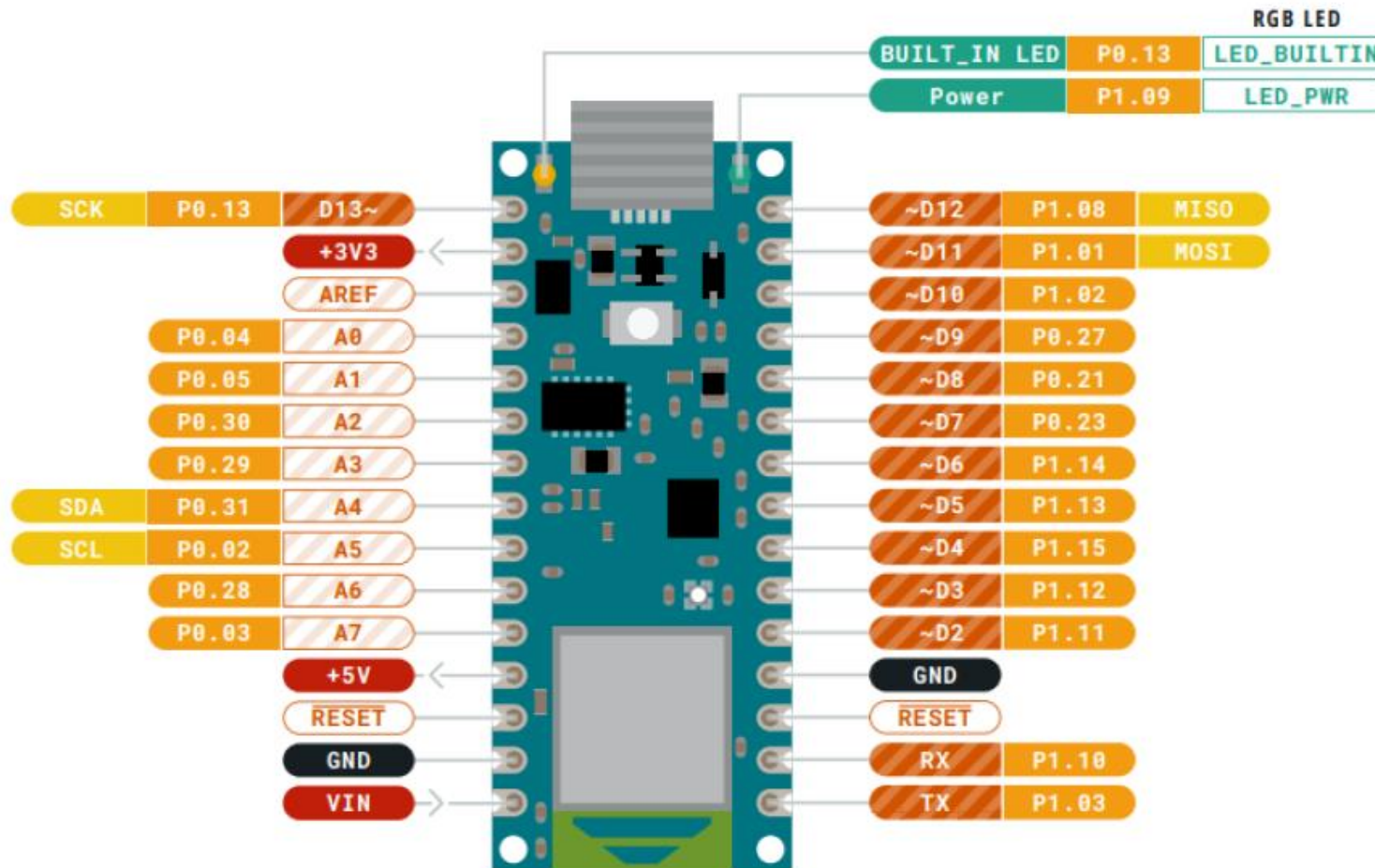
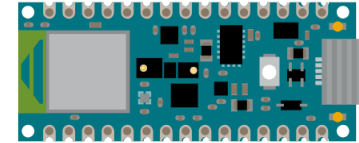


Diagram courtesy of Arduino.cc

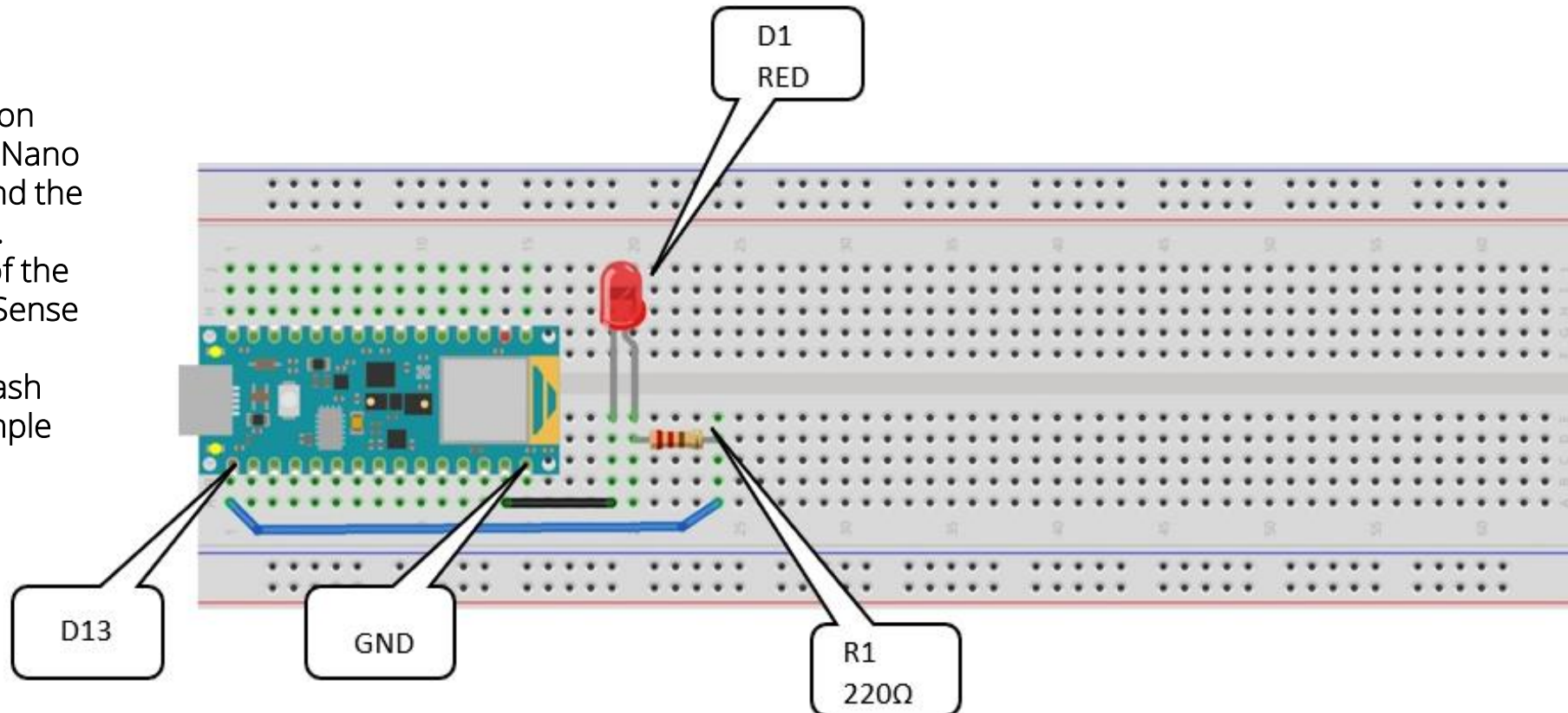
Mini Labs: Hands-On Activities... Blinking LED



Electrical Wiring Diagram: Solderless Breadboard view

Lab Objectives:

- Test USB communication between The Arduino Nano 33 BLE Sense board and the development machine.
- Test the Digital pin13 of the Arduino Nano 33 BLE Sense board.
- Experiment with the flash (Blink) rate of the Example Code.



Question 3

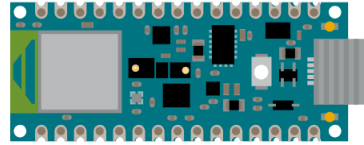
On slide 18, the external LED circuit is wired to what pin of the Arduino Nano 33 BLE Sense board?

- a) D12**
- b) D13**
- c) D15**
- d) D11**



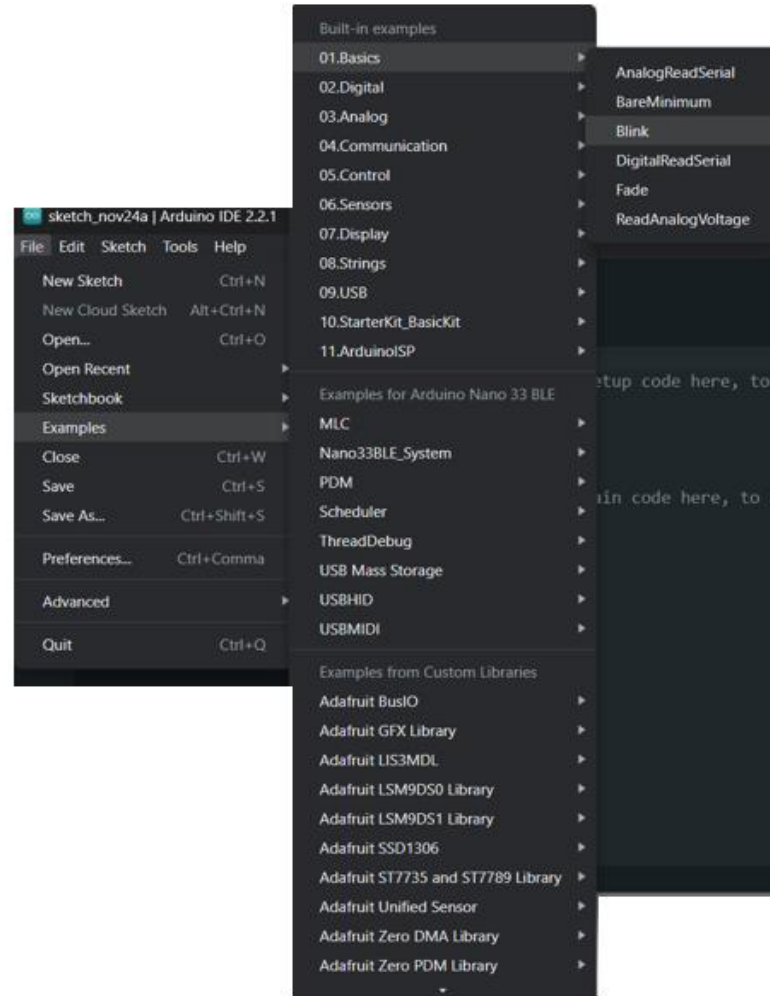
Mini Labs: Hands-On Activities

Blinking LED



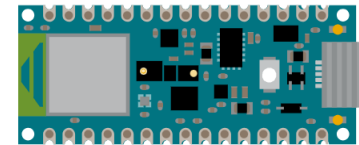
Accessing the Blink Code (Sketch)

Example>Basics>Blink



Mini Labs: Hands-On Activities

Blinking LED



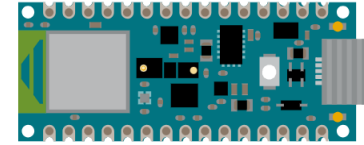
Blink Code (Sketch)

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Mini Labs: Hands-On Activities

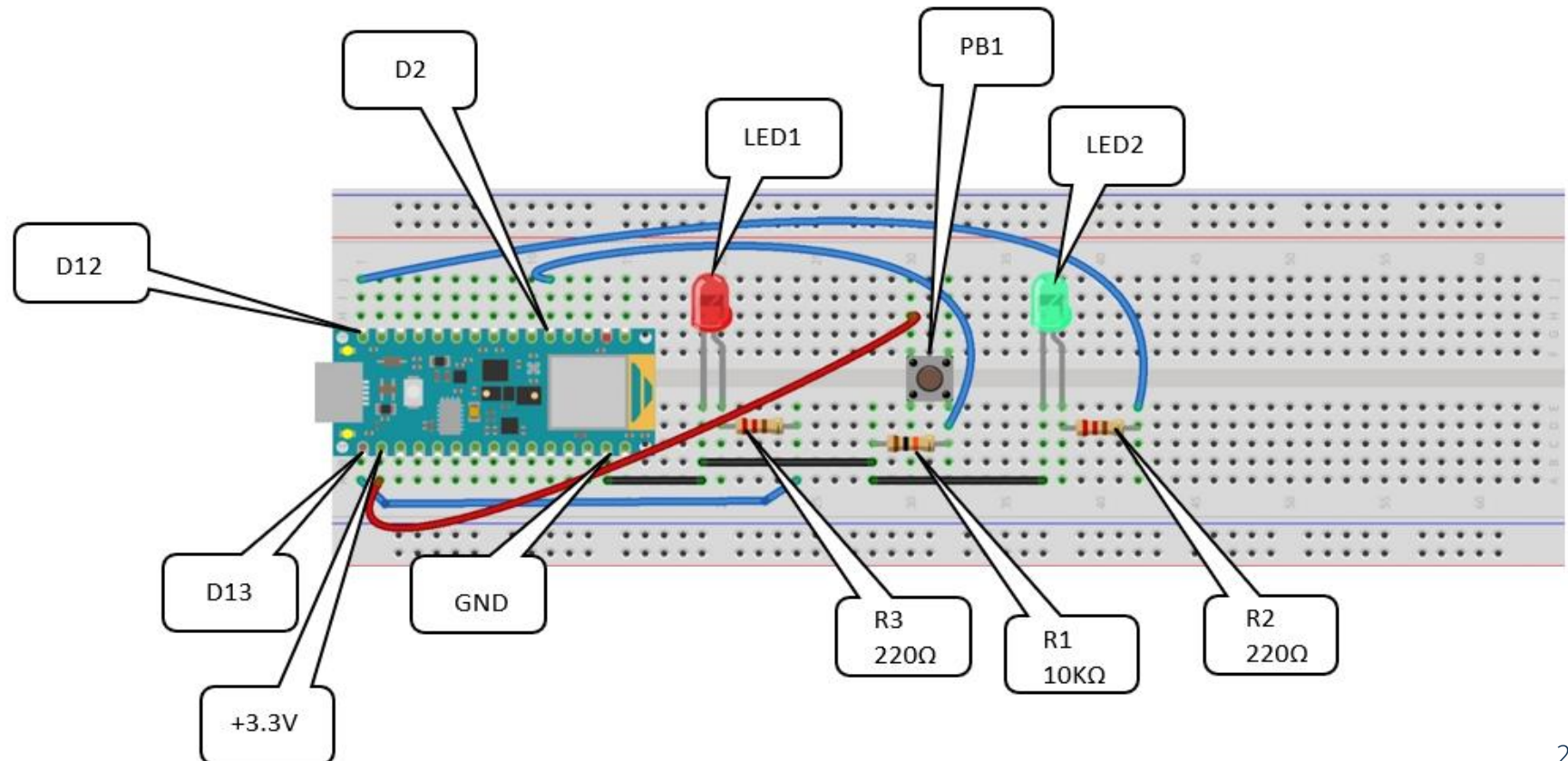
Pushbutton - Alternating Blinking LEDs



Electrical Wiring Diagram: Solderless Breadboard view

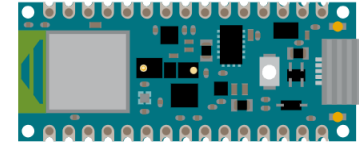
Lab Objectives:

- Test USB communication between The Arduino Nano 33 BLE Sense board and the development machine.
- Test Digital pins 2, 12, and 13 of the Arduino Nano 33 BLE Sense board.
- Modify the Blink code using Bard AI.
- Experiment with the flash (Blink) rate of the Example Code.



Mini Labs: Hands-On Activities

Pushbutton – Alternating Blinking LEDs



Pushbutton Alternating Blinking LEDs Bard AI-Generated Code (Partial)



Write code where pressing a pushbutton switch (PB1) wired at digital pin 2 (D2) will blink an LED wired at digital pin 13 (D13).



Sure, here is the code where pressing a pushbutton switch (PB1) wired at digital pin 2 (D2) will blink an LED wired at digital pin 13 (D13):

```
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13;   // the number of the LED pin
```

1

```
// variable for reading the pushbutton status
int buttonState = 0;
```

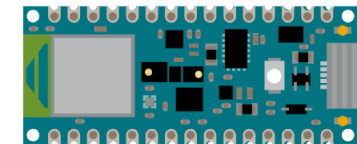
2

```
void setup() {
  // initialize the pushbutton pin as an input
  pinMode(buttonPin, INPUT);

  // initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
}
```

Mini Labs: Hands-On Activities

Pushbutton – Alternating Blinking LEDs



Pushbutton Alternating Blinking LEDs Bard AI-Generated Code Several Zero-Shot Learning revision

Pushbutton_Alternate_Blink.ino

```
1  const int buttonPin = 2; // the number of the pushbutton pin
2  const int ledPin1 = 13; // the number of the LED pin at D13
3  const int ledPin2 = 12; // the number of the LED pin at D12
4
5  int buttonState = 0; // variable for reading the pushbutton status
6
7  void setup() {
8    // initialize the pushbutton pin as an input
9    pinMode(buttonPin, INPUT);
10
11    // initialize the LED pins as outputs
12    pinMode(ledPin1, OUTPUT);
13    pinMode(ledPin2, OUTPUT);
14
15    // initially turn off both LEDs
16    digitalWrite(ledPin1, LOW);
17    digitalWrite(ledPin2, LOW);
18  }
19
```

```
20 void loop() {
21   // read the state of the pushbutton
22   buttonState = digitalRead(buttonPin);
23
24   // control the LEDs based on the button state
25   if (buttonState == HIGH) {
26     // turn on D12, then turn off after half the blink interval
27     digitalWrite(ledPin2, HIGH);
28     delay(500);
29     digitalWrite(ledPin2, LOW);
30
31     // turn on D13, then turn off after half the blink interval
32     digitalWrite(ledPin1, HIGH);
33     delay(500);
34     digitalWrite(ledPin1, LOW);
35   }
36 }
```


Question 4

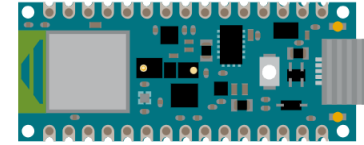
In analyzing the Pushbutton – Alternating Blinking LEDs code, which line number reads the state of the pushbutton?

- a) 9**
- b) 16**
- c) 25**
- d) 22**



Mini Labs: Hands-On Activities

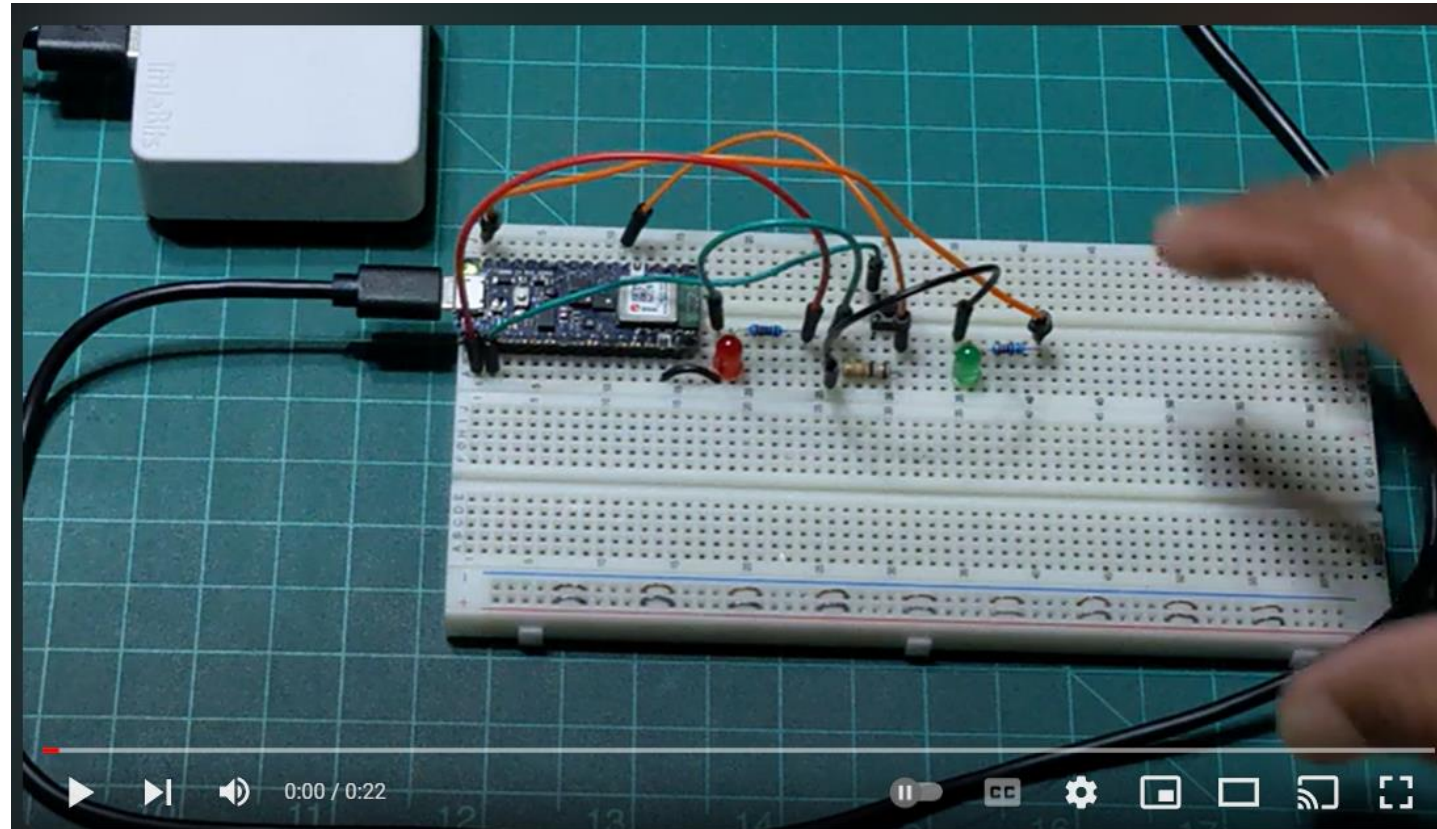
Pushbutton - Alternating Blinking LEDs



Pushbutton Alternating Blinking LEDs Controller Video Clip

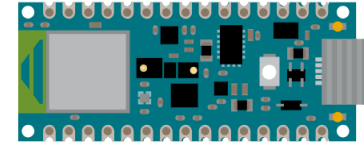
Click on the link to watch the
Pushbutton-Alternating Blinking
LED Controller Demo

<https://youtu.be/-CKXcvOhJDo>



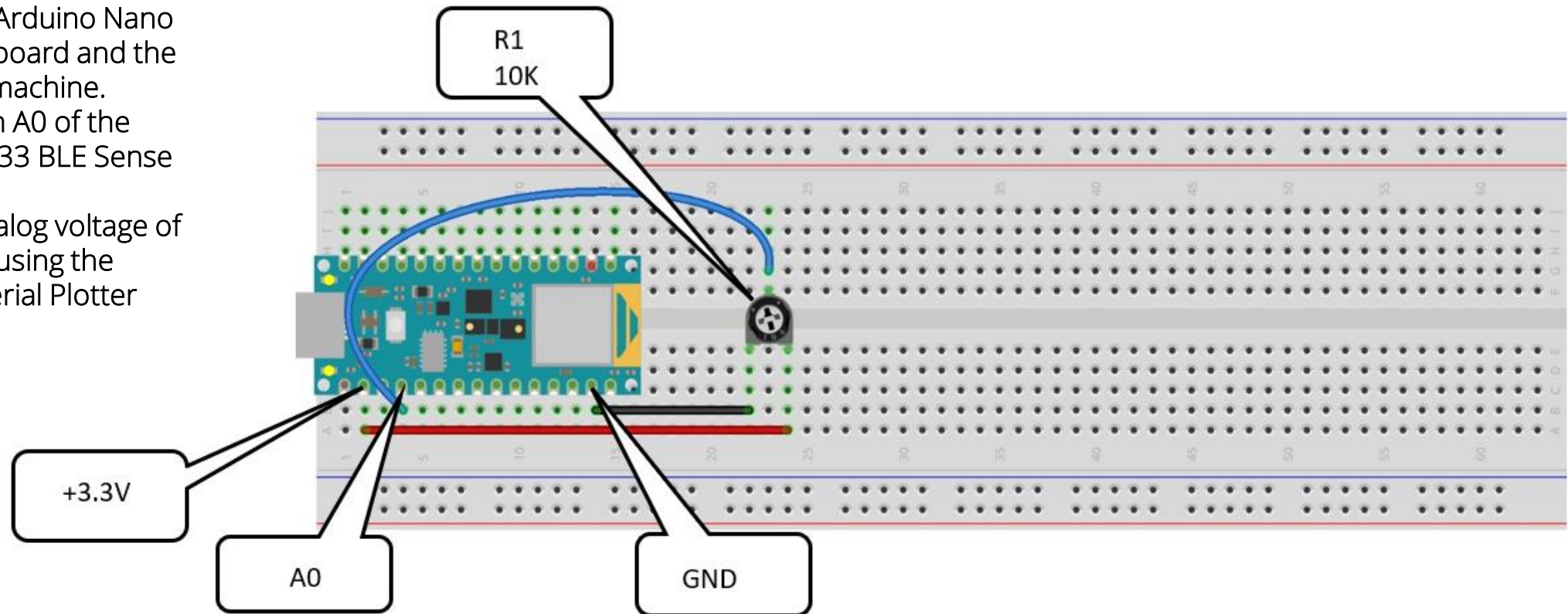
Mini Labs: Hands-On Activities

Arduino IDE Serial Plotter



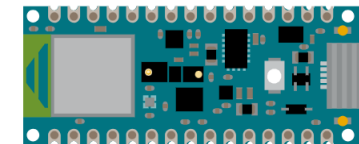
Lab Objectives:

- Test USB communication between The Arduino Nano 33 BLE Sense board and the development machine.
- Test Analog pin A0 of the Arduino Nano 33 BLE Sense board.
- Display the Analog voltage of Analog pin A0 using the Arduino IDE Serial Plotter



Mini Labs: Hands-On Activities

Arduino IDE Serial Plotter Code



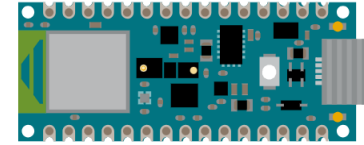
Analog_Sensor_Serial_Plotter1.ino

```
1  const int potPin = A0;
2
3  void setup() {
4      Serial.begin(9600);
5  }
6
7  void loop() {
8      int sensorValue = analogRead(potPin);
9
10     // Send the sensor value to the Serial Plotter
11     Serial.println(sensorValue);
12
13     delay(100); // Adjust the delay as needed
14 }
```

Mini Labs: Hands-On Activities

Analog Sensor Serial Plot: Arduino IDE

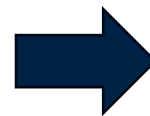
Serial Plot



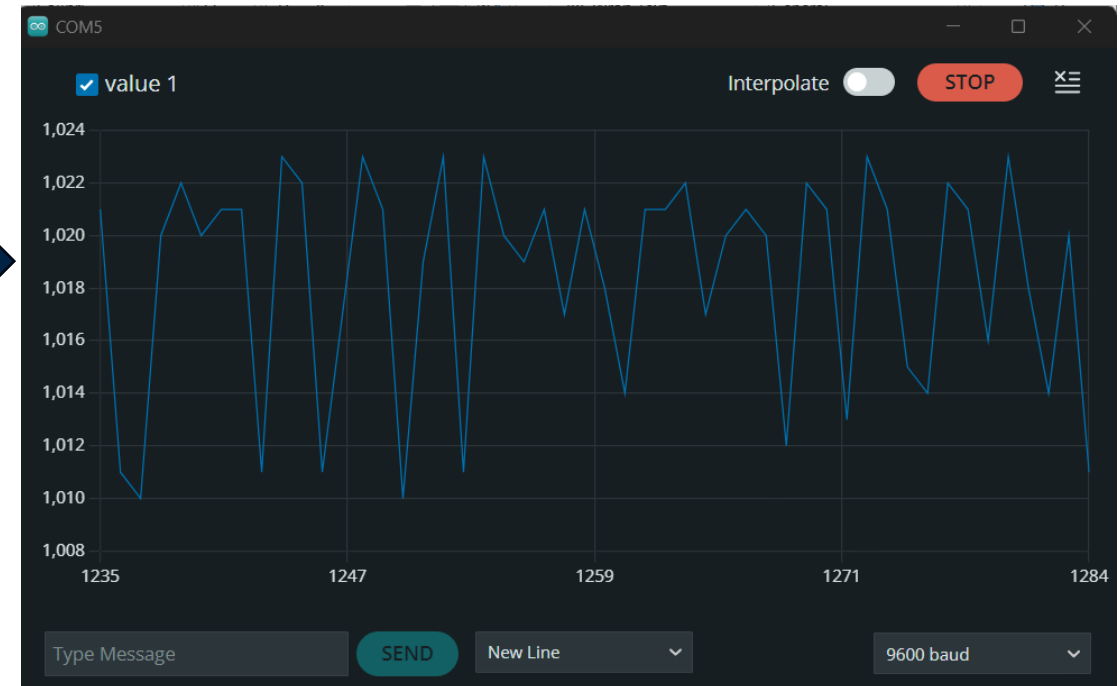
Analog Sensor Serial Plot Code

```
Arduino Nano 33 BLE
Analog_Sensor_Serial_Plotter1.ino
1  const int potPin = A0;
2
3  void setup() {
4    Serial.begin(9600);
5  }
6
7  void loop() {
8    int sensorValue = analogRead(potPin);
9
10   // Send the sensor value to the Serial Plotter
11   Serial.println(sensorValue);
12
13   delay(100); // Adjust the delay as needed
14 }
```

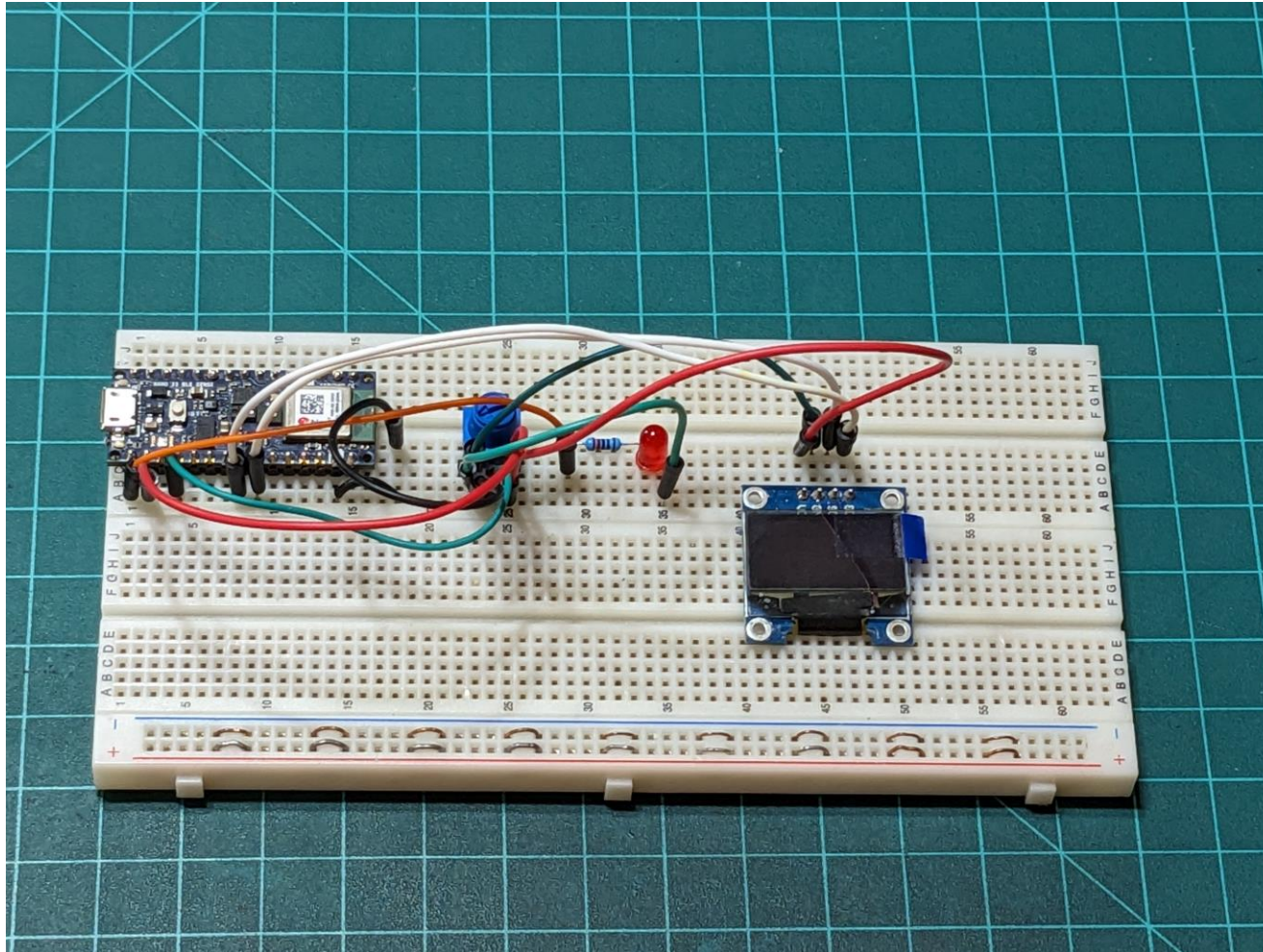
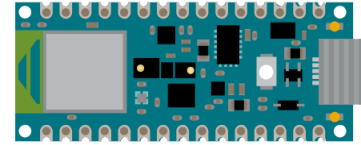
Accessing the
Serial Plotter



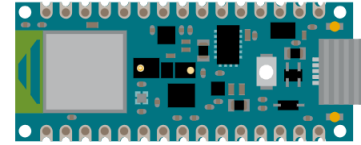
Analog Plot



Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)



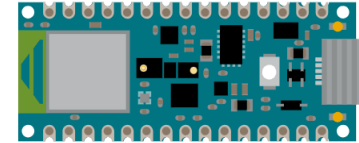
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



Lab Objectives:

- Participants will learn to wire the Analog Sensor I2C OLED demonstrator.
- Participants will learn to install the Adafruit GFX and the Adafruit SSD1306 OLED libraries.
- Participants will learn to program the Arduino Nano 33 BLE Sense board.
- Participants will learn how to run the Analog Sensor I2C OLED demonstrator.

Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



What is Bricolage?

Bard AI-Generated Explanation.



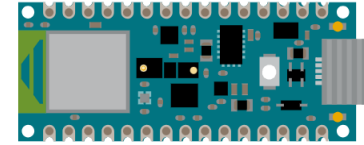
Please explain Bricolage.



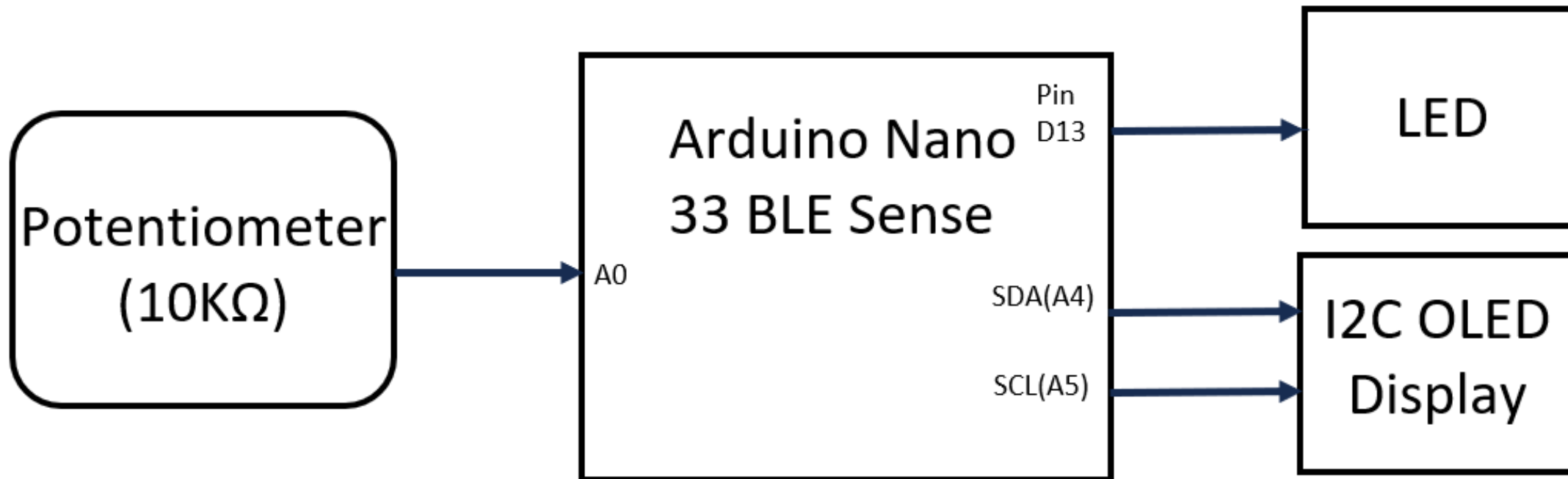
The term "bricolage" is derived from the French word "bricoler," which means "to tinker" or "to make do with what one has." It was first introduced into the English language by anthropologist Claude Lévi-Strauss in his 1966 book *The Savage Mind*. Lévi-Strauss used the term to describe the way in which people in traditional societies use whatever materials are available to them to solve problems and create new things.



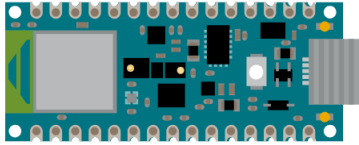
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



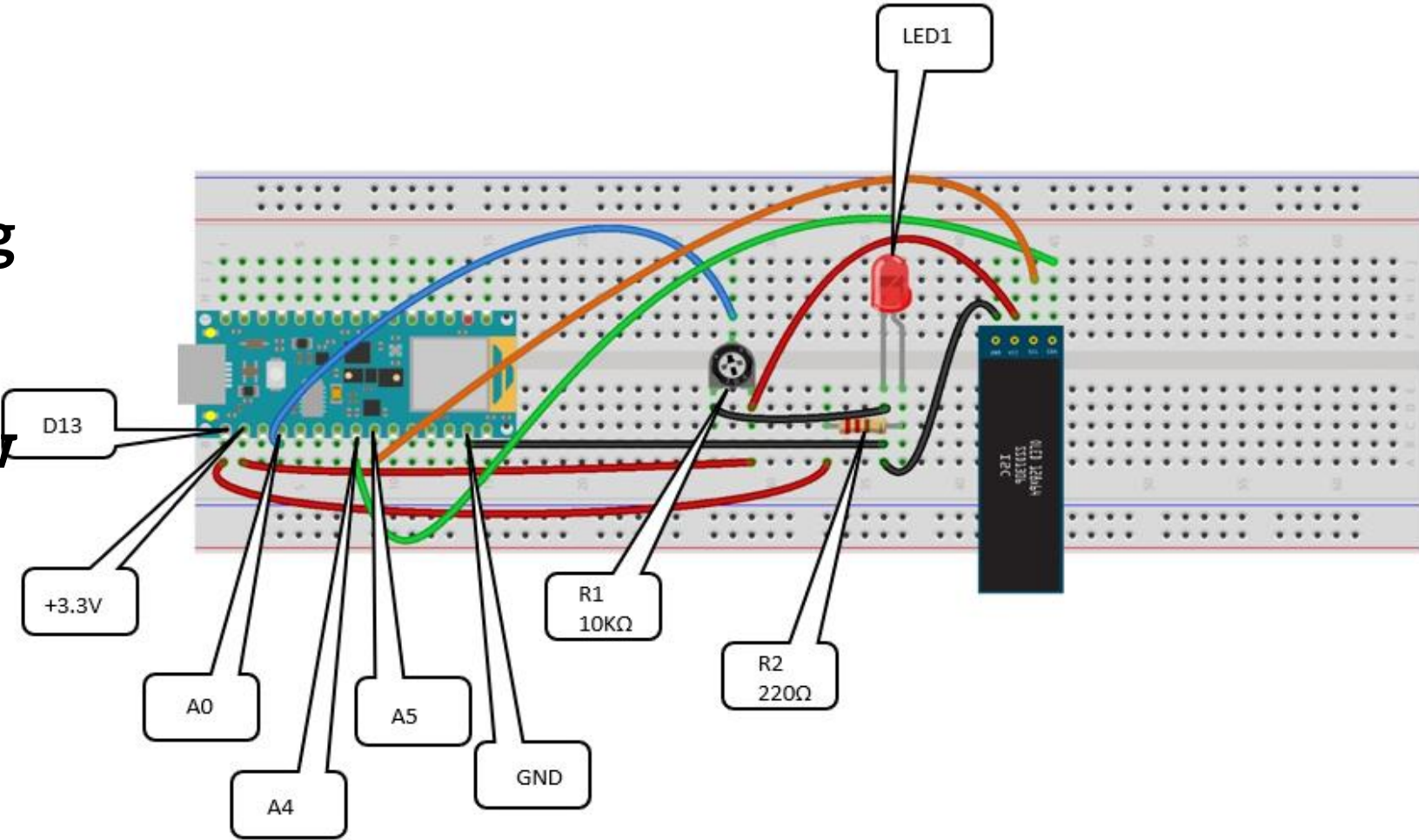
Concept System Block Diagram



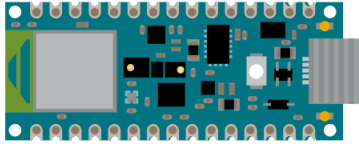
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



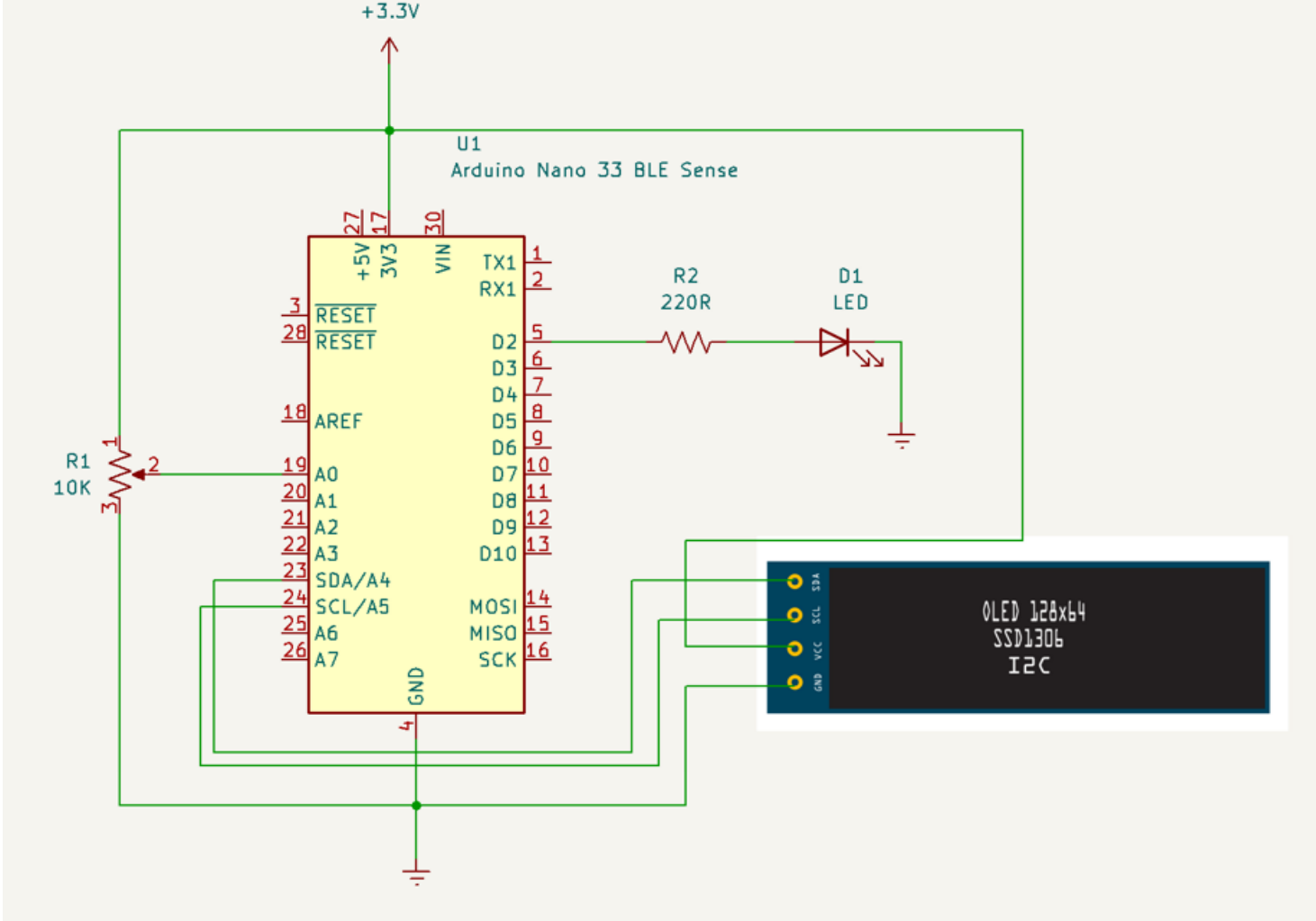
**Electrical Wiring Diagram:
Solderless Breadboard view**



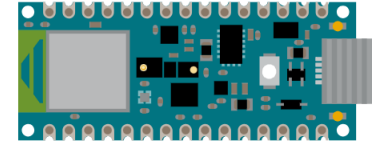
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



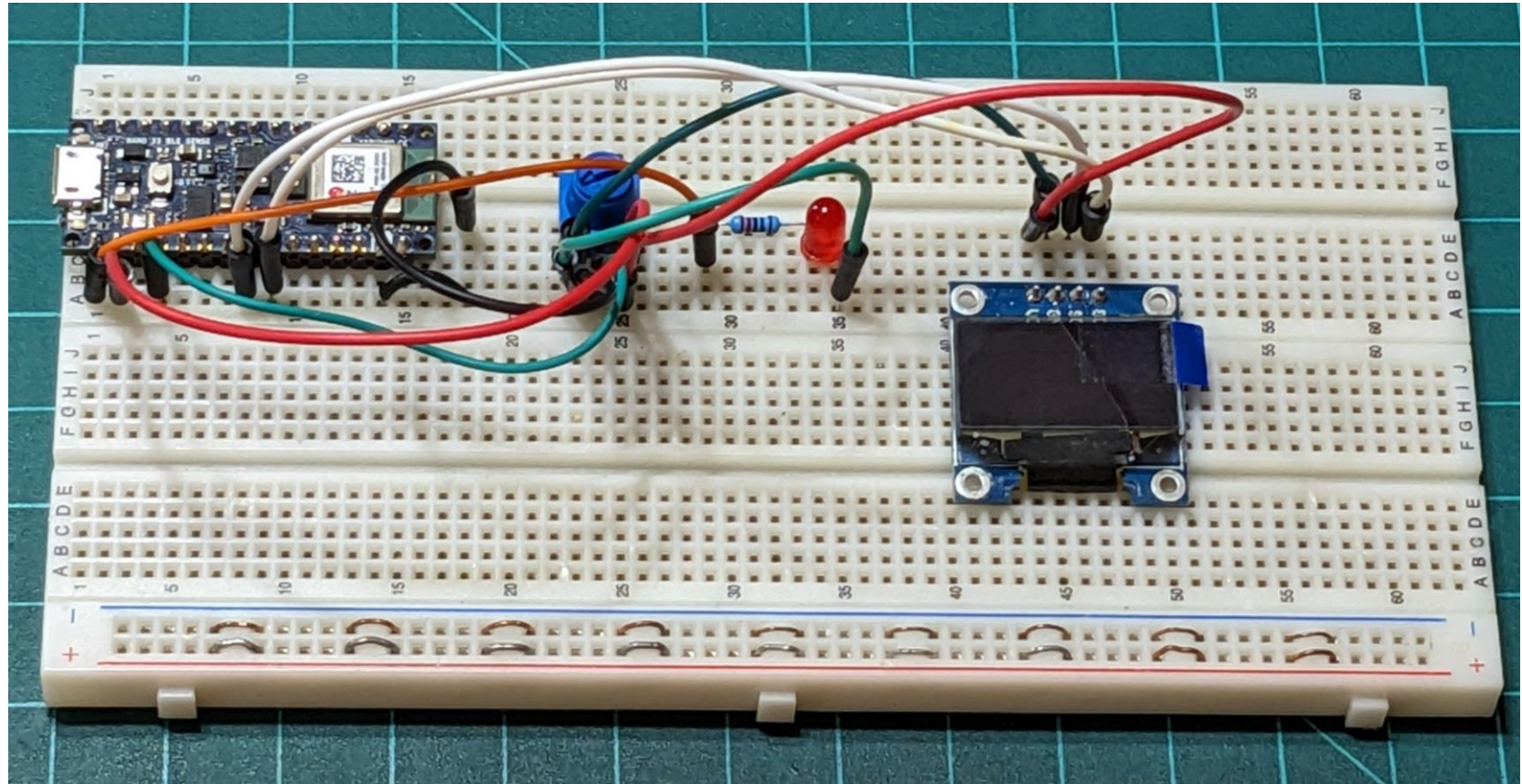
Electronic Circuit Schematic Diagram



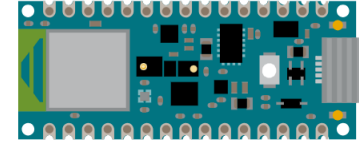
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



Solderless Breadboard Build



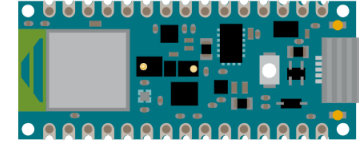
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



Arduino Nano 33 BLE Sense Code: Part 1

```
1  #include <Wire.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_SSD1306.h>
4
5  //Adafruit_SSD1306 display(128, 64);
6
7  #define SCREEN_WIDTH 128
8  #define SCREEN_HEIGHT 64
9
10 #define OLED_RESET 4
11 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
12
13 const int potPin = A0; // Define the analog pin connected to the potentiometer
14
15 void setup() {
16     Serial.begin(9600);
17     display.setTextColor(SSD1306_WHITE); // Set the color for the text
18
19     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // The I2C address for this particular OLED display is provided.
20         Serial.println(F("SSD1306 allocation failed"));
21         for(;;);
22     }
23
24
25     // Set the background color to black
26     display.fillRect(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT, SSD1306_BLACK);
27     display.display();
28     delay(2000);
29     display.clearDisplay(); // Clear the display after setting the background color
30 }
```

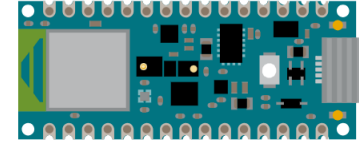
Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



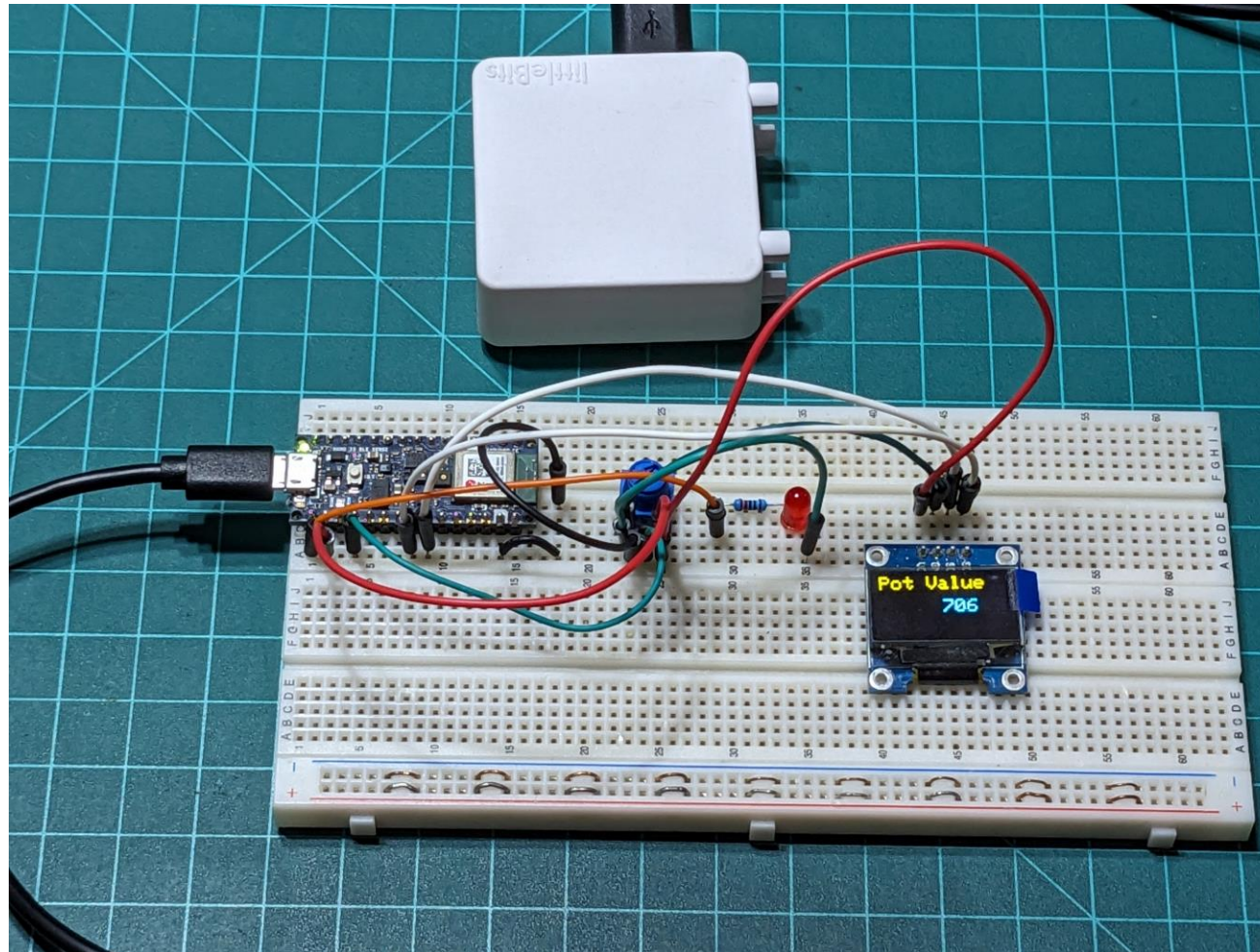
Arduino Nano 33 BLE Sense Code: Part 2

```
25 // Set the background color to black
26 display.fillRect(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT, SSD1306_BLACK);
27 display.display();
28 delay(2000);
29 display.clearDisplay(); // Clear the display after setting the background color
30 }
31
32
33 void loop() {
34   int potValue = analogRead(potPin); // Read the potentiometer value
35   //int potPercentage = map(potValue, 0, 1023, 0, 100); // Convert the potentiometer value to a percentage (0-100)
36
37   // Display the potentiometer value and percentage on the OLED display
38   display.clearDisplay();
39   display.setTextSize(2);
40   display.setCursor(0, 0);
41   display.print("Pot Value");
42   display.setTextSize(2);
43   display.setCursor(64, 24);
44   display.print(potValue);
45   Serial.println(potValue);
46   delay(100);
47
48   // If the sensor value is greater than 800, turn on the onboard LED
49   if (potValue > 800) {
50     digitalWrite(LED_BUILTIN, HIGH);
51   } else {
52     digitalWrite(LED_BUILTIN, LOW);
53   }
54
55
56   display.display(); // Update the display
57 }
```

Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...



**Solderless
Breadboard Build:
Demonstrator
Powered ON and
Functional**

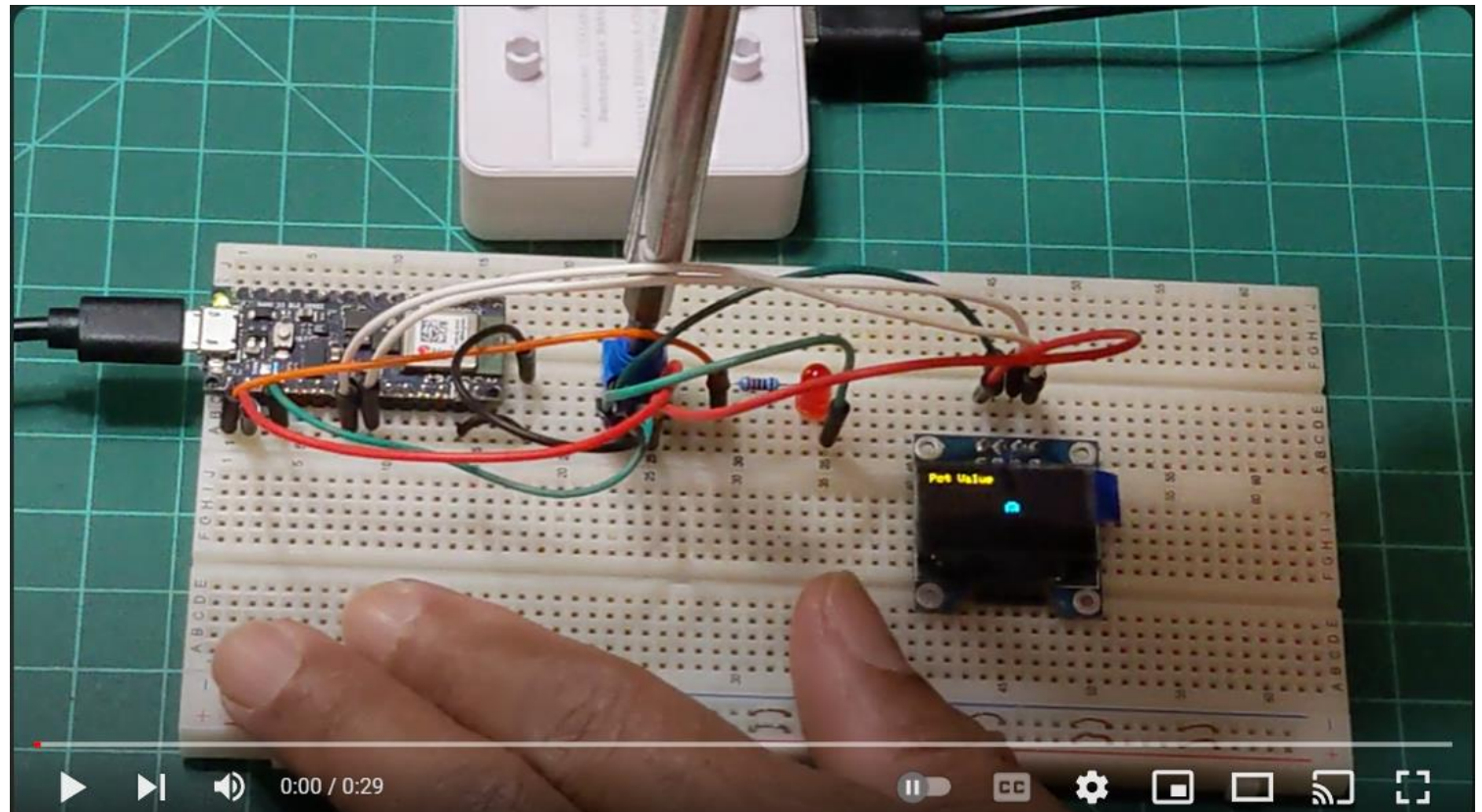


Lab: Analog Sensor I2C OLED Demonstrator (Bricolage)...

Running Analog Sensor I2C OLED Demonstrator on the Solderless Breadboard

Click on the link to watch the Analog Sensor I2C OLED Demonstrator

<https://youtu.be/7NmV3JOaii8>



Question 5

In analyzing the Analog Sensor I2C OLED Demonstrator code on slide 38, `if(potvalue < 800)` what is the output response of the LED indicator?

- a) on**
- b) off**
- c) tri-state**
- d) None of the above**



Thank you for attending

Please consider the resources below:

Chua, C.E., & Storey, V.C. (2022). A tutorial on prototyping internet of things device and systems: A gentle introduction to technology that shapes our lives. *Communications of the Association for Information Systems*, 51(34), 327-364.

https://www.researchgate.net/publication/360263045_A_Tutorial_on_Prototyping_Internet_of_Things_Devices_and_Systems_A_Gentle_Introduction_to_Technology_that_Shapes_Our_Lives

Course_Lab_project_code.zip folder: Github Repository: https://github.com/DWilcher/HCI_Electronics

Kurniawan, A. (2021). *Iot projects with arduino nano 33 ble sense*. Apress.
https://link.springer.com/chapter/10.1007/978-1-4842-6458-4_3



DesignNews

Thank You

Sponsored by

DigiKey

