

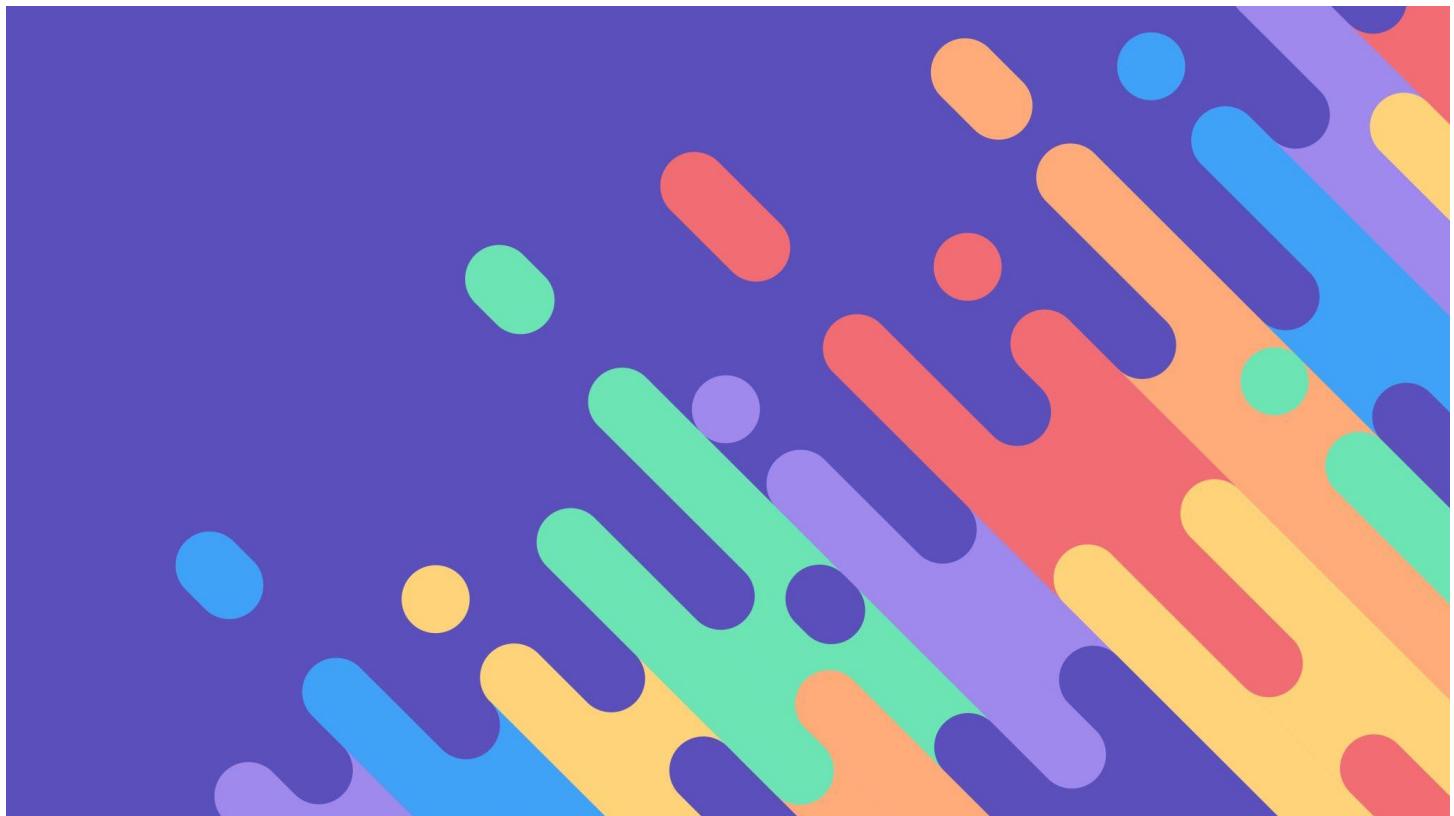
Arduino Introduction to Basic Electronics Circuit & Projects

DR SHAHINO MAH ABDULLAH

Arduino

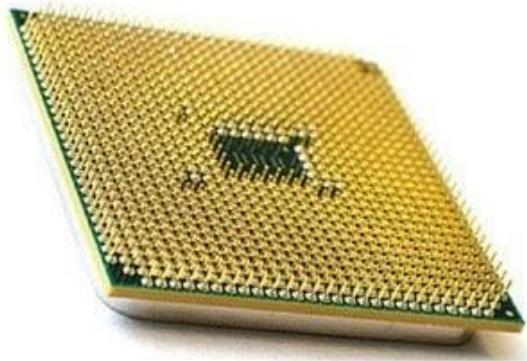
Introduction to Basic Electronics Circuit & Projects

DR SHAHINO MAH
ABDULLAH
shahinomah@gmail.com
0197883473





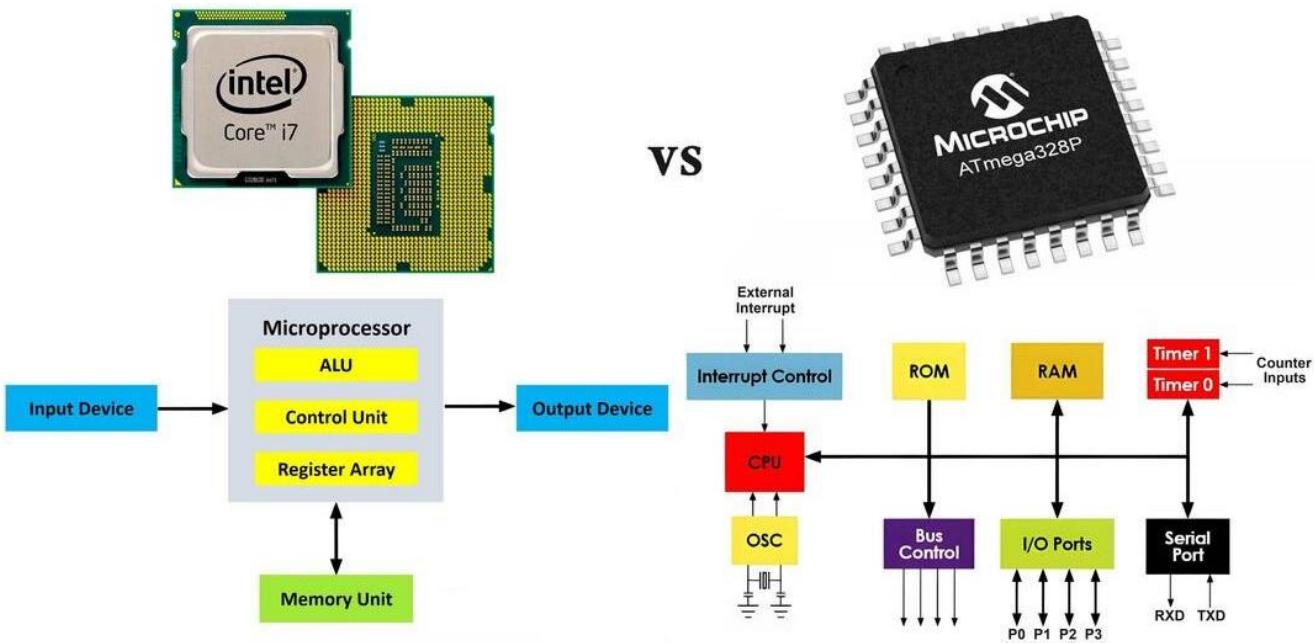
Difference Basis	Microprocessor	Microcontroller
Core	It has only CPU Embedded into it	It has CPU, RAM, FLASH and other peripherals embedded into it
Application	Widely used in the computer systems	Widely used in the embedded system
Scale	Large scale processing	Small scale processing
Usage	PC, Mobile Phone, Raspberry pi etc.	Arduino, STM32 and MSP432 etc



VS



Microprocessor



Microcontroller

Microprocessor vs Microcontroller

Memory	Needs external memory & storage, flexible	On-chip memory (RAM, ROM, EEPROM)
Peripheral	Need additional part via external bus, ideal for complex, powerful applications.	Integrated peripheral SPI, UART, RTI, ADC for direct HW control
Clock Speed	Fast GHz for heavy duty tasks	KHz - MHz, efficient for specific tasks.
Power Consumption	High power consumption, no low power option.	Low power consumption with power saving options
Connectivity	Handles high-speed transfer	supports low to moderate speed communication I2C, SPI, UART
Operating System	Need full-fledge OS (Linux, Windows)	OS is optional (no-OS or light weight RTOS)
UseCase	For generic computing, high complexity capacity system	For compact systems, battery powered, or logic processing devices
Cost	Higher cost — suited for performance-driven applications	Lower cost — economical for mass production.

Microcontroller Applications

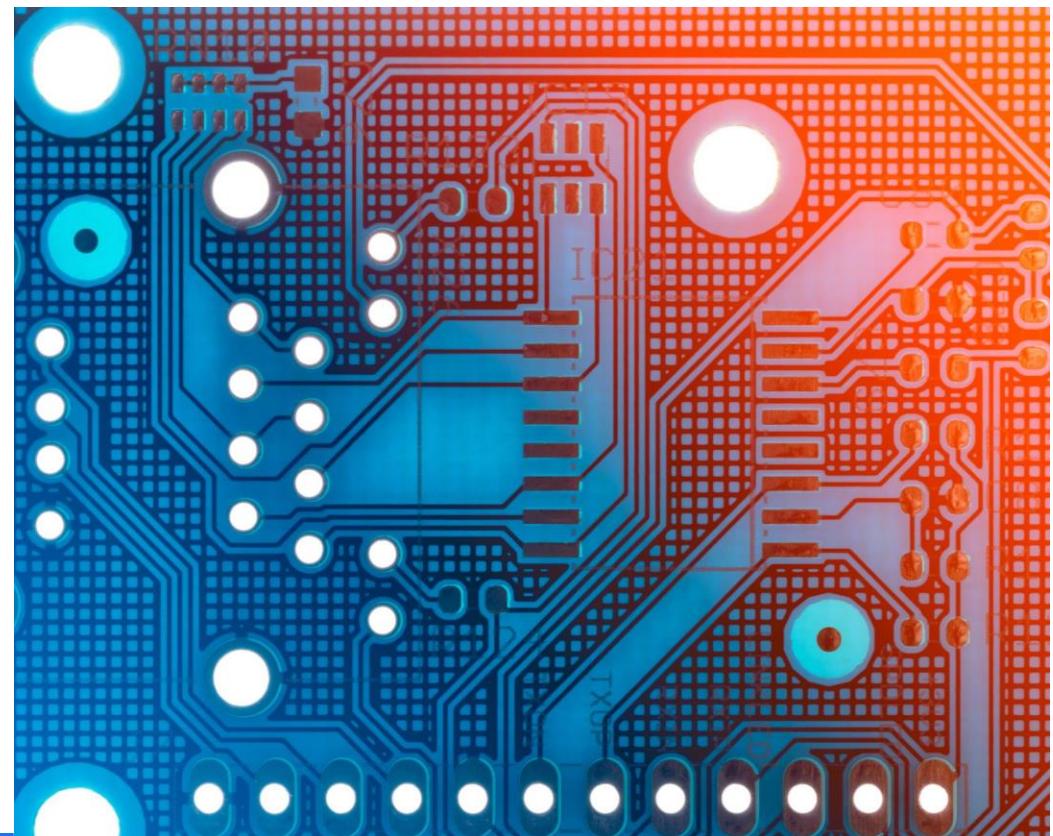


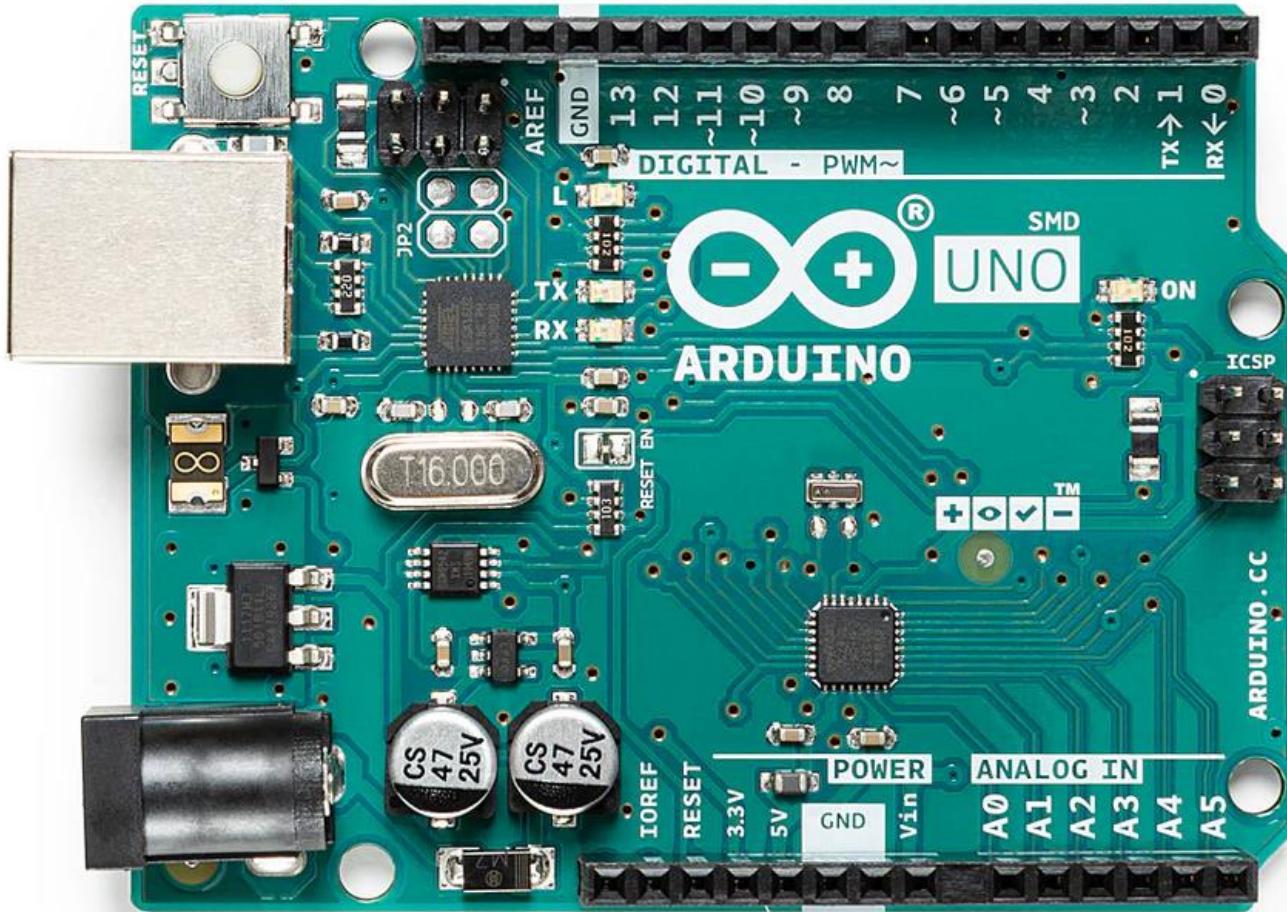
What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, turn it into an output - activating a motor, turning on an LED, publishing something online.

You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino is a programmable device that enables interaction with external hardware devices using software programs. Hardware devices that can be connected to Arduino include lights, sensors, actuators, screens, speakers, and other electronic devices.





Arduino Uno

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments.

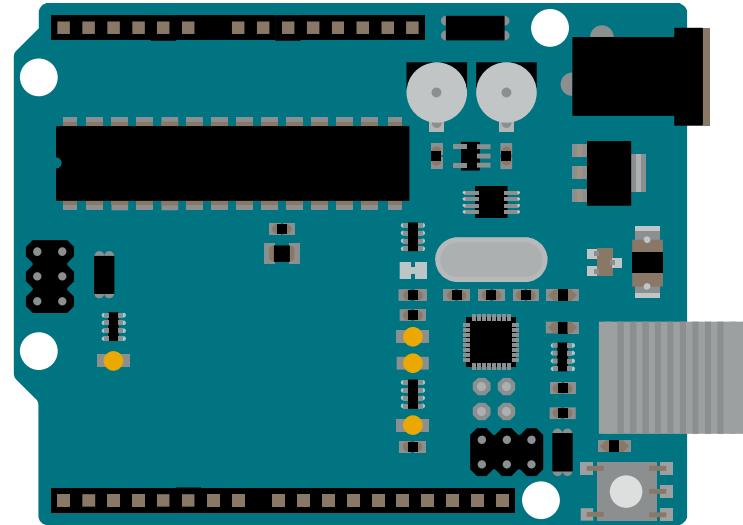
A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.



Board Type	Microcontroller	Clock Frequency
Arduino UNO	Atmega 328P	16M Hz
Arduino Nano	Atmega328	20MHz
Arduino Mega	ATmega2560	16 MHz
Arduino Micro	ATmega32U4	16 MHz
Arduino Due	Atmel SAM3X8E	84 MHz
Arduino Pro Mini	ATmega328	16MHz
Arduino Lilly Pad	ATmega168	8 MHz
Arduino Protenta H7	STM32H747	480 MHz
Arduino Leonardo	ATmega32u4	16 MHz
Arduino Zero	ATSAMD21G18	48 MHz

ARDUINO UNO R3

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.



Arduino UNO is a microcontroller board based on the [ATmega328P](#).

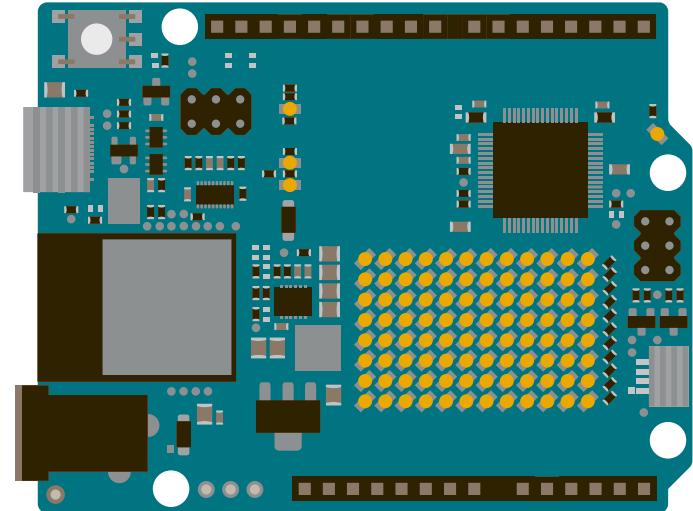
It has:

- 14 digital input/output pins (of which 6 can be used as PWM outputs)
- 6 analog inputs
- a 16 MHz ceramic resonator
- a USB connection
- a power jack
- an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

ARDUINO UNO R4 WiFi

The Arduino UNO R4 WiFi is designed around the 32-bit microcontroller [RA4M1 from Renesas](#) while also featuring a [ESP32 module for Wi-Fi®](#) and [Bluetooth® connectivity](#). Its distinctive [12x8 LED matrix](#) makes it possible to prototype visuals directly on the board, and with a Qwiic connector, you can create projects plug-and-play style.



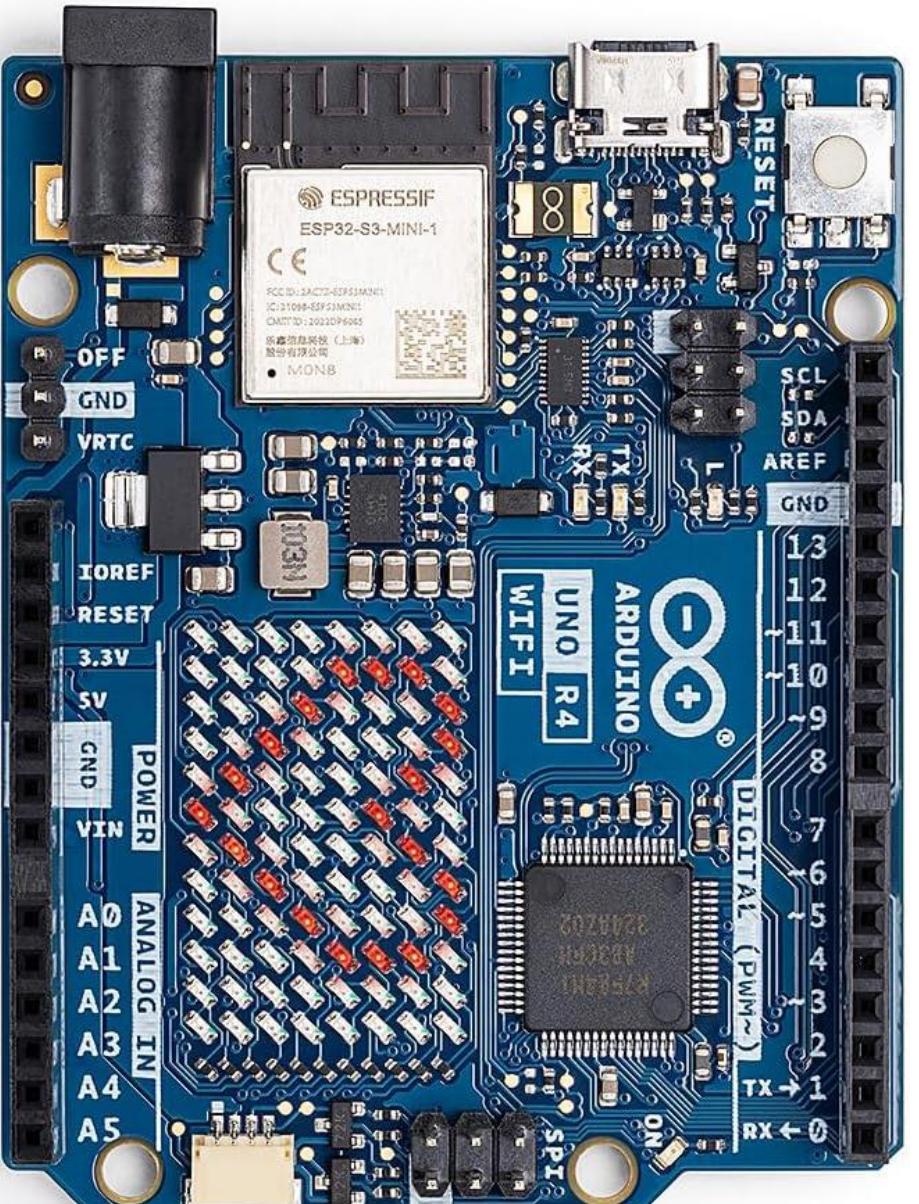
The MCU on the board is the high performance Renesas RA4M1 (Arm® Cortex®-M4) with a 48 MHz clock speed, 32 kB SRAM and 256 kB flash memory. This MCU features an RTC, a DAC and a CAN bus and has support for HID via USB.

The UNO R4 WiFi also features an ESP32-S3 for Wi-Fi®/Bluetooth® connectivity, which can also be separately programmed via a specific header

Digital-to-Analog Converter (DAC)

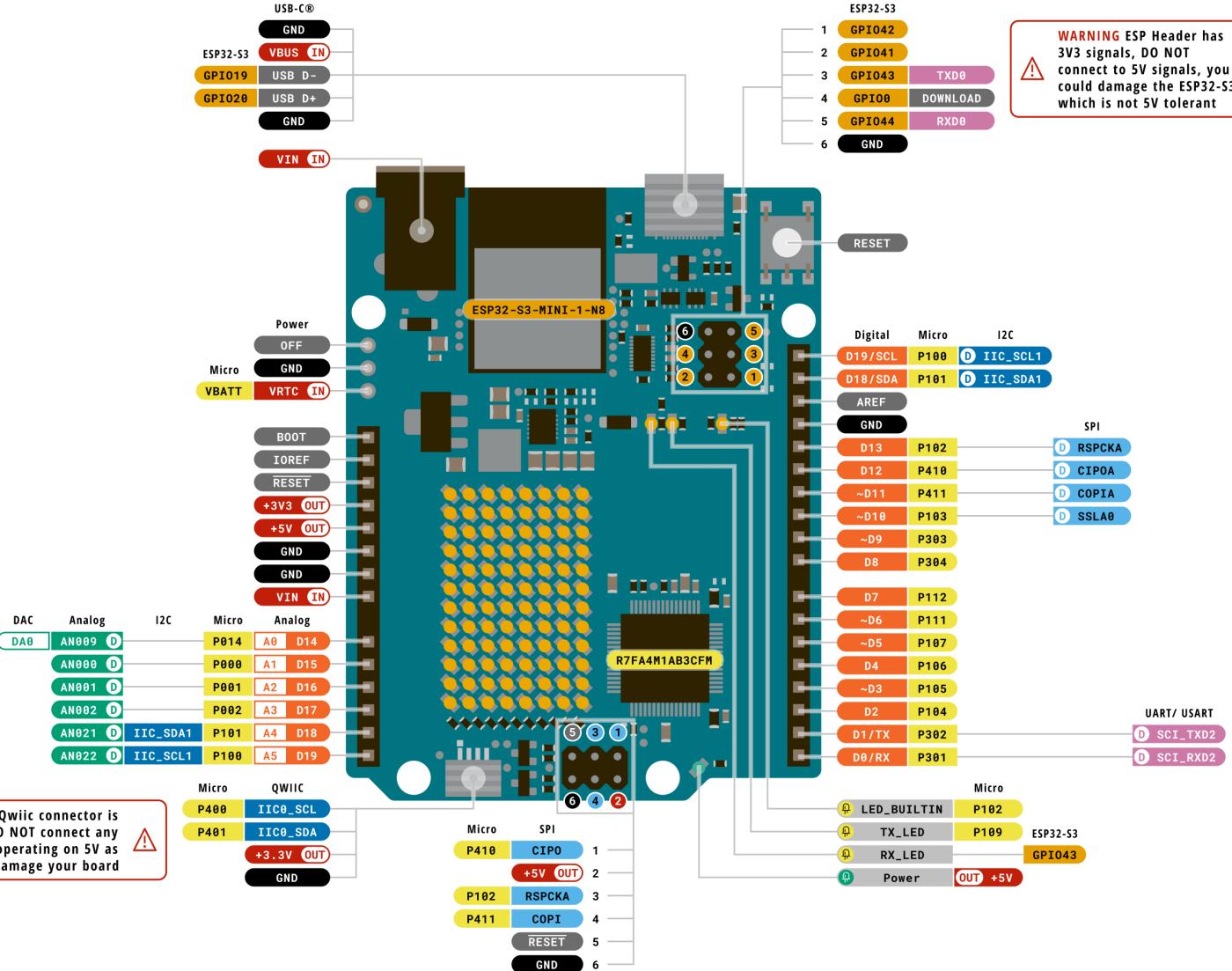
Use the onboard 12-bit DAC to build sophisticated audio projects.

ARDUINO UNO R4 WiFi



WARNING The Qwiic connector is 3V3 only. DO NOT connect any components operating on 5V as this may damage your board

Legend:	■ Digital	■ I2C	■ Other SERIAL
■ Power	□ Analog	■ SPI	■ Analog
■ Ground	■ Main Part	■ UART/USART	■ PWM/Timer



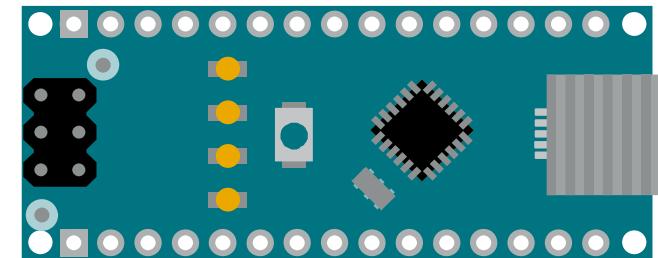
WARNING ESP Header has 3V3 signals, DO NOT connect to 5V signals, you could damage the ESP32-S3 which is not 5V tolerant

Pin	Function	Type	Description	UNO (LEFT)
1	NC	NC	Not connected	
2	IOREF	IOREF	Reference for digital logic V - connected to 5V	
3	Reset	Reset	Reset	
4	+3V3	Power	+3V3 Power Rail	
5	+5V	Power	+5V Power Rail	
6	GND	Power	Ground	
7	GND	Power	Ground	
8	VIN	Power	Voltage Input	
9	A0	Analog/GPIO	Analog input 0 /GPIO	
10	A1	Analog/GPIO	Analog input 1 /GPIO	
11	A2	Analog/GPIO	Analog input 2 /GPIO	
12	A3	Analog/GPIO	Analog input 3 /GPIO	
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line	
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line	

Pin	Function	Type	Description	UNO (RIGHT)
1	D0	Digital/GPIO	Digital pin 0/GPIO	
2	D1	Digital/GPIO	Digital pin 1/GPIO	
3	D2	Digital/GPIO	Digital pin 2/GPIO	
4	D3	Digital/GPIO	Digital pin 3/GPIO	
5	D4	Digital/GPIO	Digital pin 4/GPIO	
6	D5	Digital/GPIO	Digital pin 5/GPIO	
7	D6	Digital/GPIO	Digital pin 6/GPIO	
8	D7	Digital/GPIO	Digital pin 7/GPIO	
9	D8	Digital/GPIO	Digital pin 8/GPIO	
10	D9	Digital/GPIO	Digital pin 9/GPIO	
11	SS	Digital	SPI Chip Select	
12	MOSI	Digital	SPI1 Main Out Secondary In	
13	MISO	Digital	SPI Main In Secondary Out	
14	SCK	Digital	SPI serial clock output	
15	GND	Power	Ground	
16	AREF	Digital	Analog reference voltage	
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)	
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)	Dr Shahino Mah Abdullah

ARDUINO NANO

The Arduino Nano is Arduino's classic breadboard friendly designed board with the **smallest dimensions**. The Arduino Nano comes with pin headers that allow for an easy attachment onto a breadboard and features a Mini-B USB connector (can be found with type-C connector)



Microcontroller

Built-in LED	ATmega328
Digital I/O	Pin 13
Analog input pins	Pins 14
PWM pins	8
	6

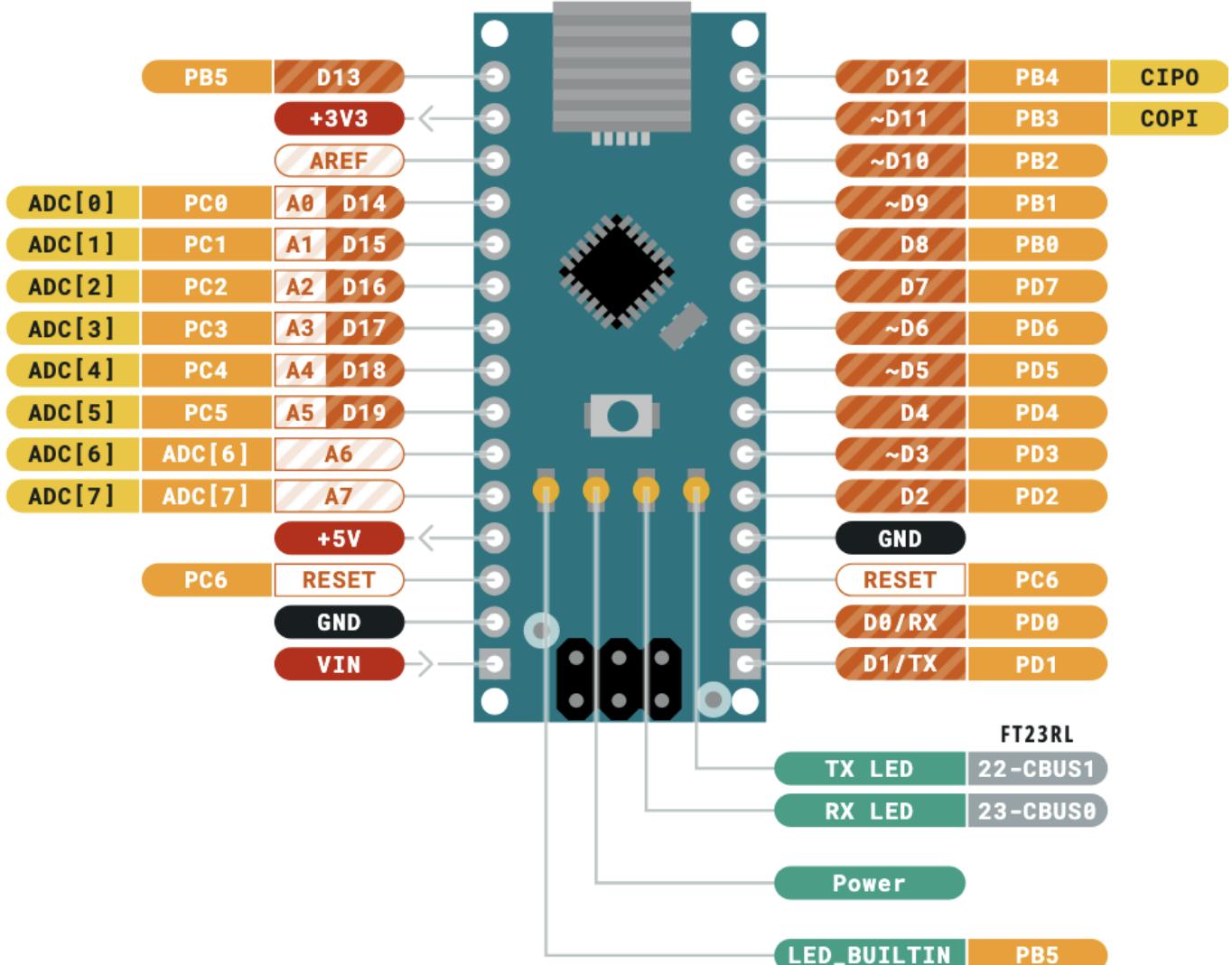
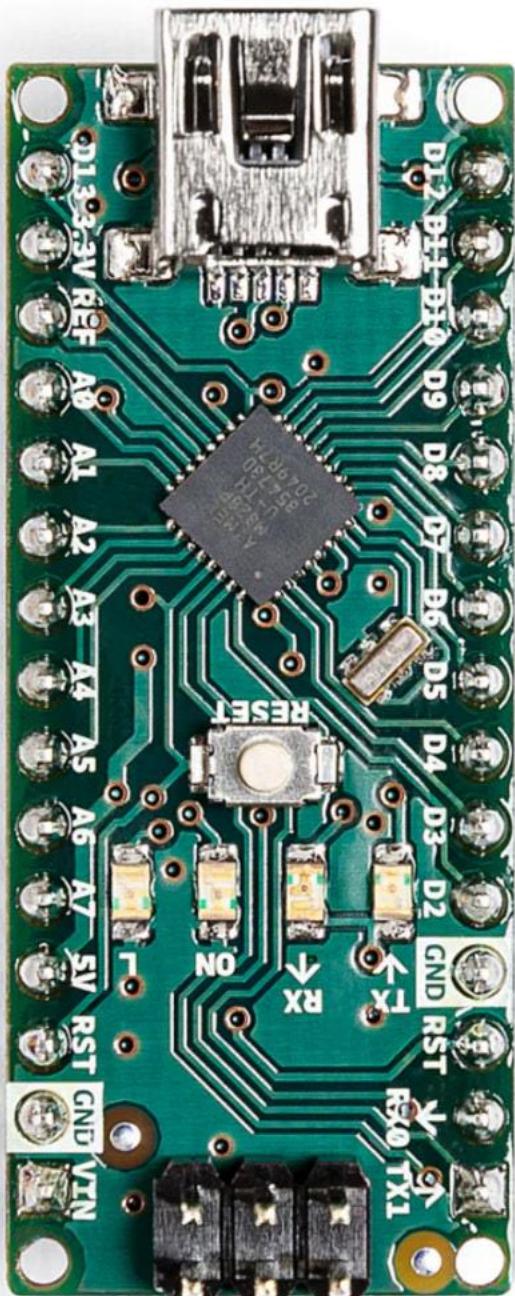
Communication

UART	RX/TX
I2C	A4 (SDA), A5 (SCL)
SPI	D11 (COPI), D12 (CIPO), D13 (SCK). Use any GPIO for Chip Select (CS).

Power

I/O Voltage	5V
Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA

ARDUINO NANO



█ Ground
█ Power
█ LED
█ Internal Pin
█ SWD Pin

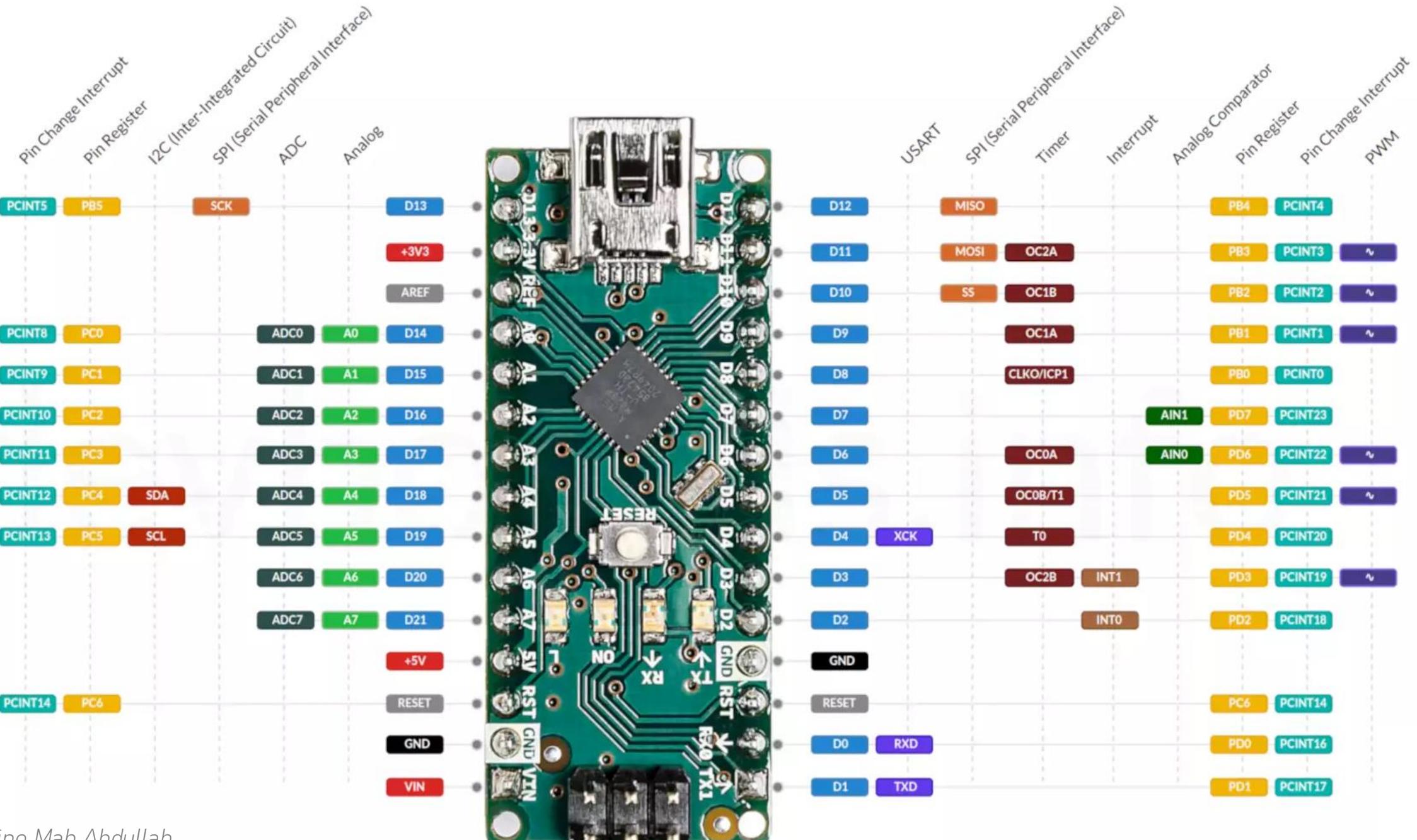
█ Digital Pin
█ Analog Pin
█ Other Pin
█ Microcontroller's Port
█ Default

⚠ MAXIMUM current per I/O pin is 20mA
⚠ MAXIMUM current per +3.3V pin is 50mA

NOTE: CIPO/COP1 have previously been referred to as MISO/MOSI

VIN 7-12 V input to the board.

ARDUINO NANO



Pin	Function	Type	Description	NANO (LEFT)
1	+3V3	Power	5V USB Power	
2	A0	Analog	Analog input 0 /GPIO	
3	A1	Analog	Analog input 1 /GPIO	
4	A2	Analog	Analog input 2 /GPIO	
5	A3	Analog	Analog input 3 /GPIO	
6	A4	Analog	Analog input 4 /GPIO	
7	A5	Analog	Analog input 5 /GPIO	
8	A6	Analog	Analog input 6 /GPIO	
9	A7	Analog	Analog input 7 /GPIO	
10	+5V	Power	+5V Power Rail	
11	Reset	Reset	Reset	
12	GND	Power	Ground	
12	VIN	Power	Voltage Input	

Pin	Function	Type	Description	NANO (RIGHT)
1	D1/TX1	Digital	Digital Input 1 /GPIO	
2	D0/RX0	Digital	Digital Input 0 /GPIO	
3	D2	Digital	Digital Input 2 /GPIO	
4	D3	Digital	Digital Input 3 /GPIO	
5	D4	Digital	Digital Input 4 /GPIO	
6	D5	Digital	Digital Input 5 /GPIO	
7	D6	Digital	Digital Input 6 /GPIO	
8	D7	Digital	Digital Input 7 /GPIO	
9	D8	Digital	Digital Input 8 /GPIO	
10	D9	Digital	Digital Input 9 /GPIO	
11	D10	Digital	Digital Input 10 /GPIO	
12	D11	Digital	Digital Input 11 /GPIO	
13	D12	Digital	Digital Input 12 /GPIO	
14	D13	Digital	Digital Input 13 /GPIO	
15	Reset	Reset	Reset	
16	GND	Power	Ground	

LIST OF COMPONENTS

Arduino basic learning kit

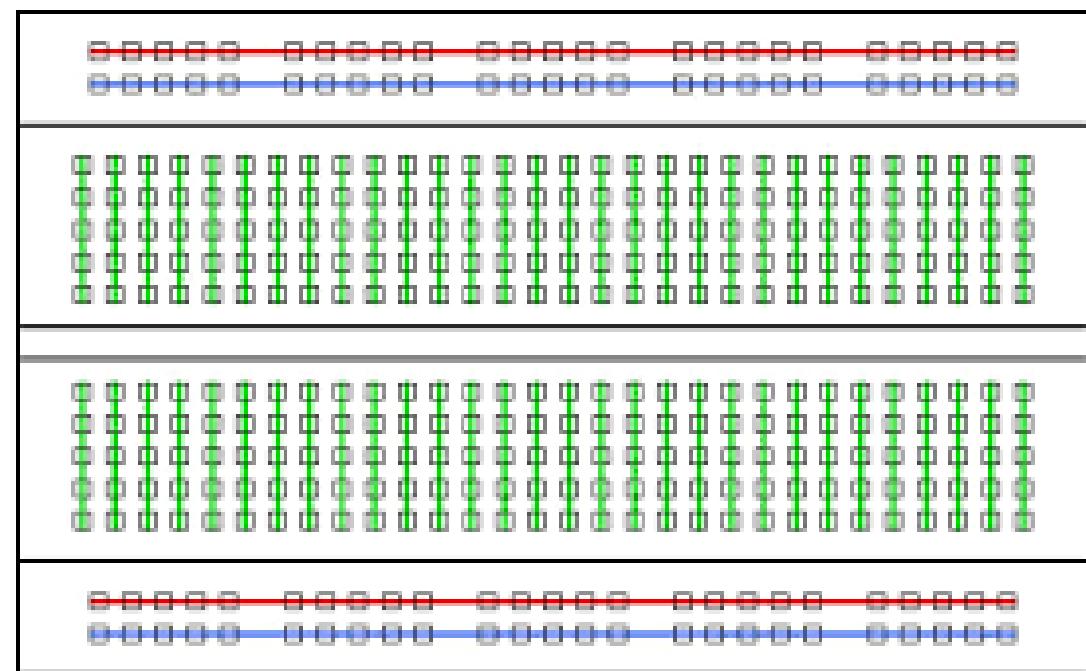
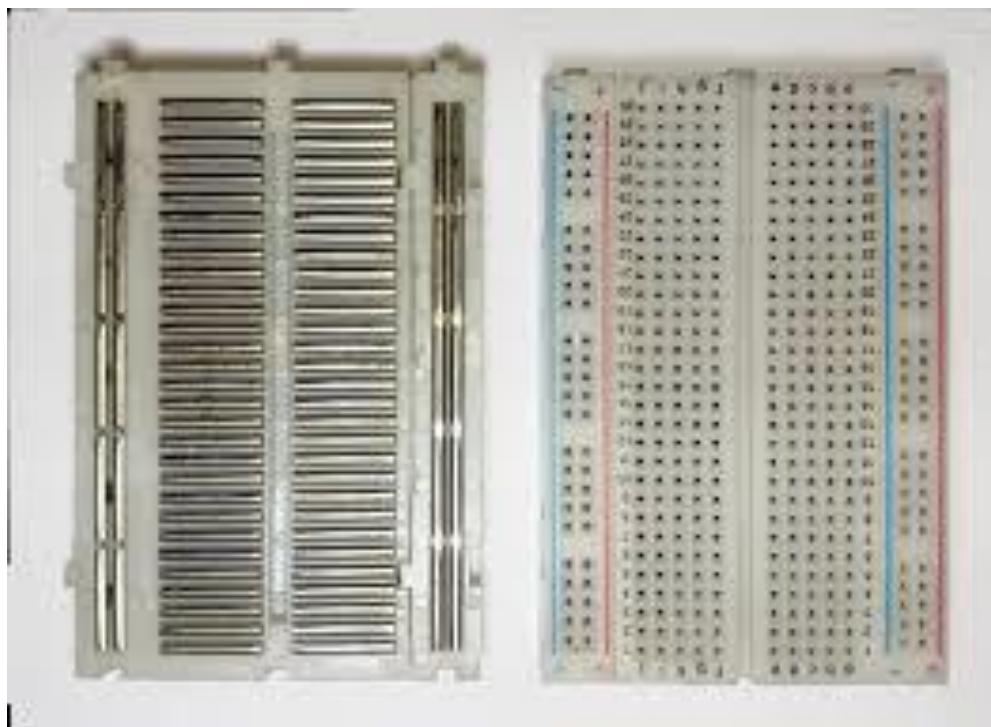
1. Arduino Nano (Type-C connector) + Type-C data cable
2. Breadboard, full size (830 points)
3. Jumper wires (Male-to-male)
4. Light Emitting Diode, LED (Red, Yellow, Green)
5. Potentiometer
6. Light Dependent Resistor (LDR) photoresistor
7. AHT10 temperature and humidity sensor
8. Buzzer
9. Ultrasonic HC-SR04
10. OLED display

Please use your own PC or laptop to download and install Arduino IDE software

LIST OF COMPONENTS

Breadboard

Electrical connection: Red line (VCC / +), Blue line (GND / -)

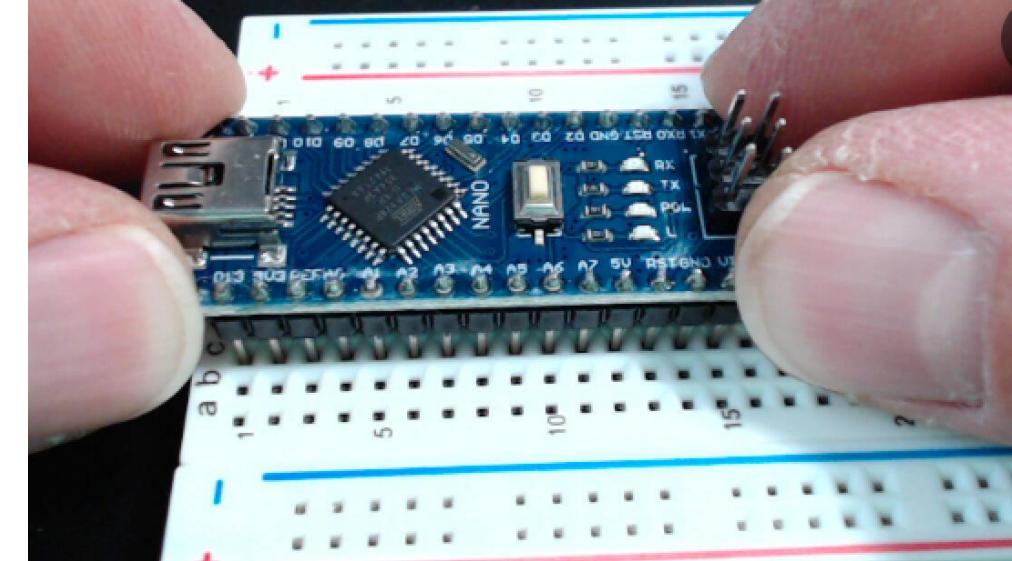
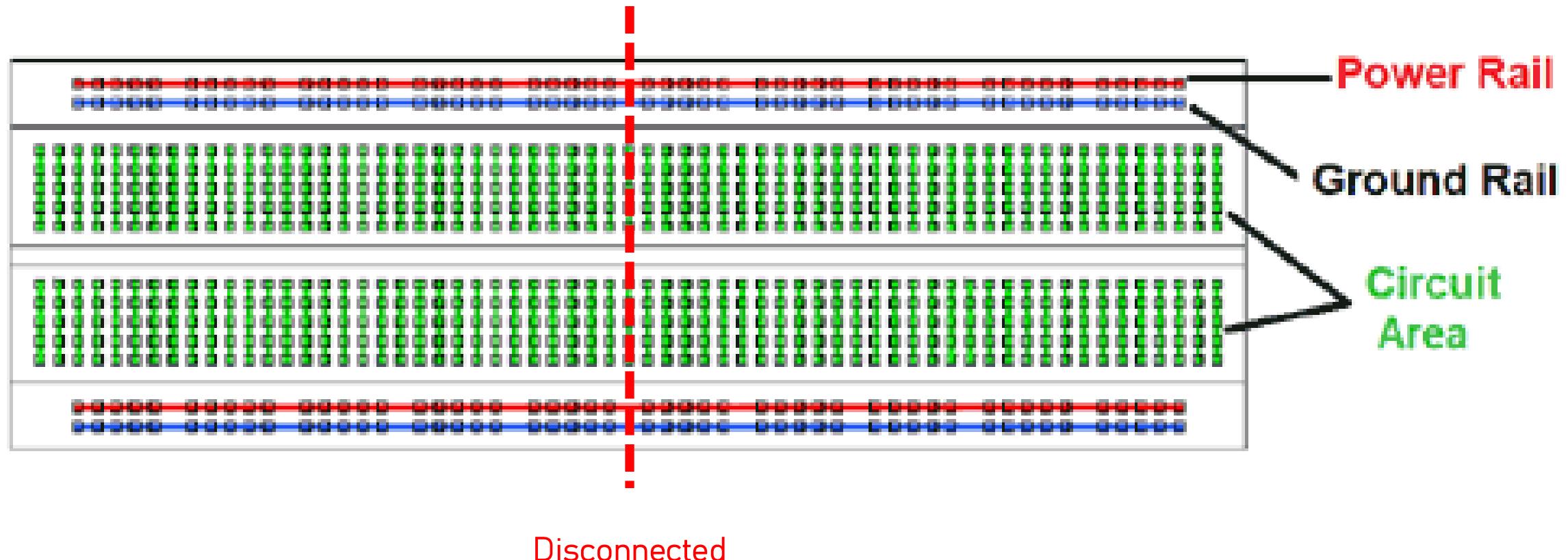


— Circuit Connections — Distribution Strips

LIST OF COMPONENTS

Breadboard

Electrical connection: Red line (VCC / +), Blue line (GND / -)

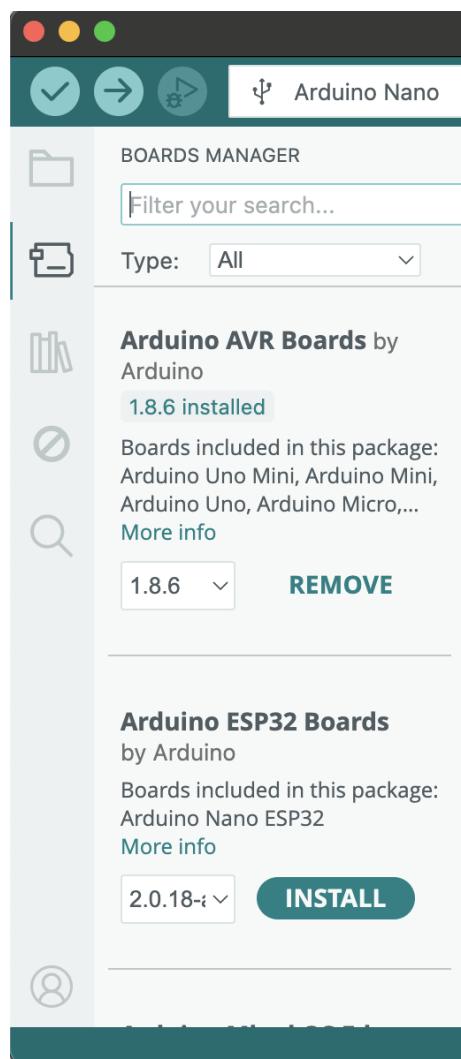


Arduino IDE (integrated development environment) software

Download link: <https://wwwarduino.cc/en/software/>

Arduino IDE 2.3.6

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.



The screenshot shows the Arduino IDE 2.3.6 interface. The title bar reads "sketch_jun10a | Arduino IDE 2.3.5". The main area displays a sketch named "sketch_jun10a.ino" with the following code:

```
1 void setup() {  
2     // put your setup code here, to run once:  
3 }  
4  
5 void loop() {  
6     // put your main code here, to run repeatedly:  
7 }  
8  
9 }  
10
```

To the left, the "BOARDS MANAGER" sidebar is open, showing two packages:

- Arduino AVR Boards** by Arduino, version 1.8.6 installed. It includes boards like Arduino Uno Mini, Arduino Mini, Arduino Uno, and Arduino Micro. A "REMOVE" button is present.
- Arduino ESP32 Boards** by Arduino, version 2.0.18-rc. It includes the Arduino Nano ESP32. An "INSTALL" button is present.

A large blue text block on the right side of the screen provides a summary of the Arduino language:

Arduino programming primarily uses a simplified version of C++, but it's often referred to as the "Arduino language" because it includes Arduino-specific functions and libraries. This language is a subset of C++, and the Arduino IDE essentially converts your code into standard C/C++ code before compiling it for the microcontroller. Therefore, while you're writing code that looks like C/C++, it's tailored to the Arduino environment

Ln 1, Col 1 Arduino Nano on /dev/cu.usbserial-0001 [not connected]

Arduino IDE (integrated development environment) software

Start with Example

Select other boards and ports

V

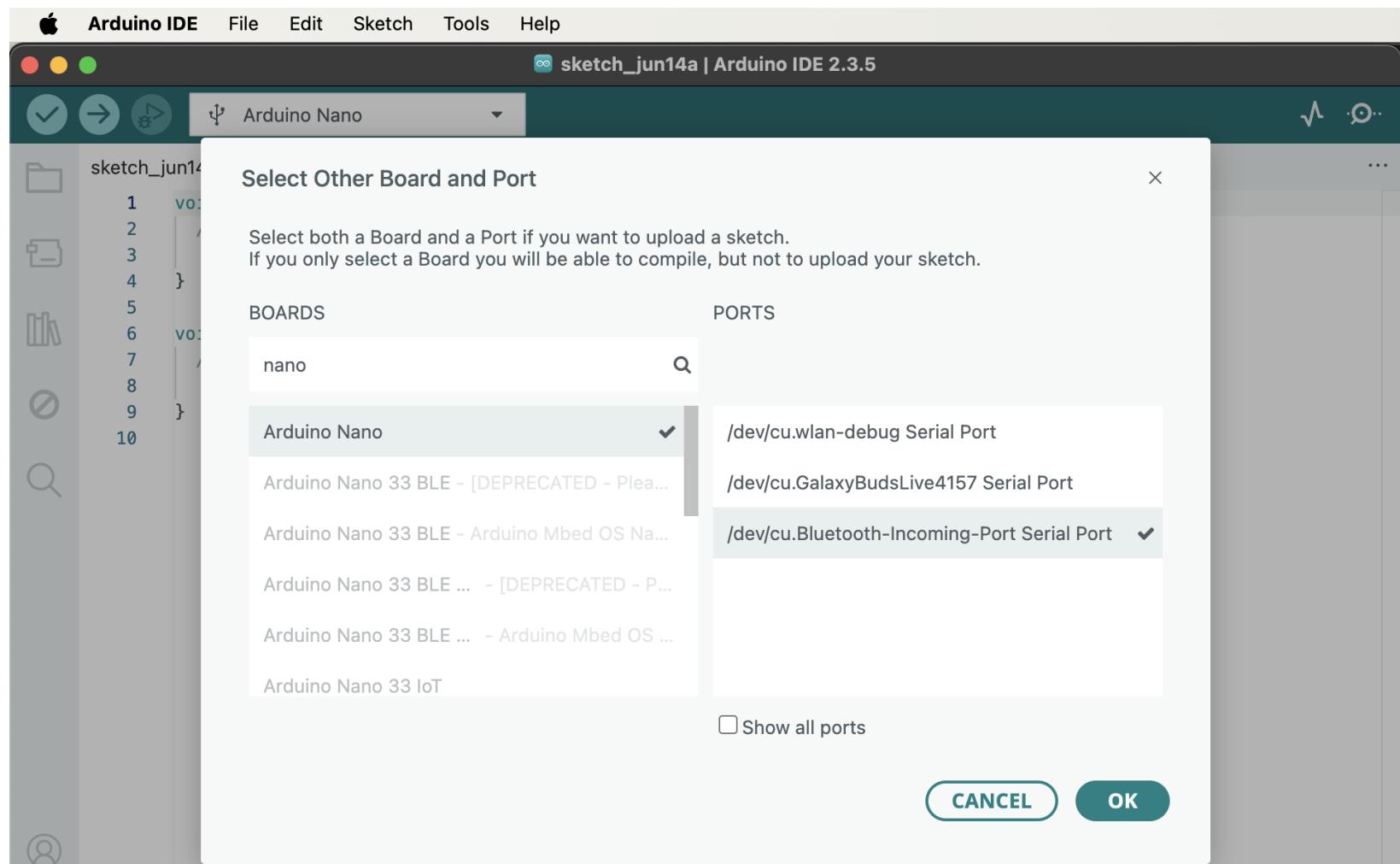
BOARDS = Arduino Nano

V

PORTS = (Serial Port)

V

OK

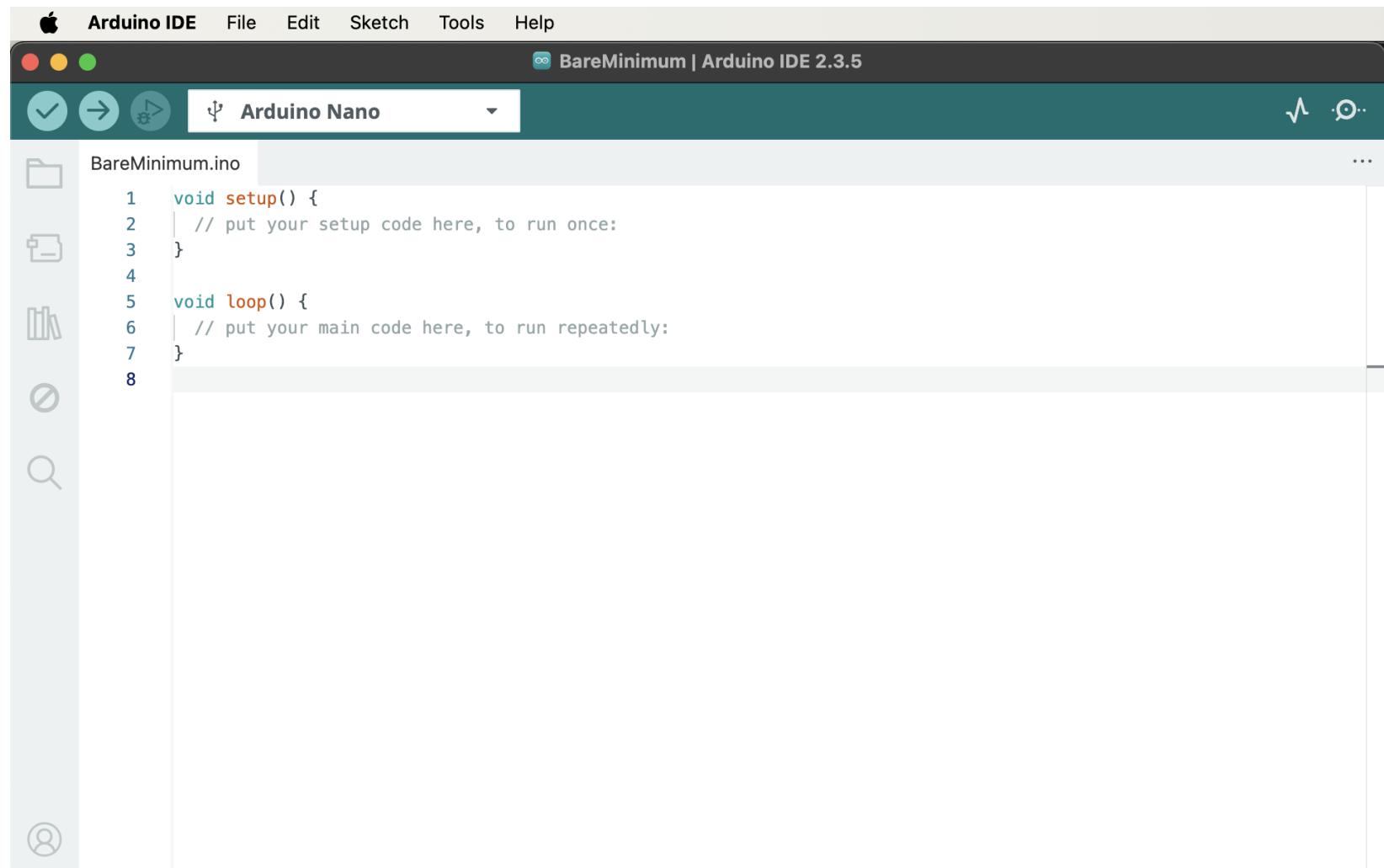


Arduino IDE (integrated development environment) software

Bare Minimum

The "Bare Minimum" Arduino sketch contains only the basic structure required for the Arduino IDE to compile and run a program.

It includes the void `setup()` and void `loop()` functions. The `setup()` function runs once at the start of the program, while the `loop()` function executes repeatedly



The screenshot shows the Arduino IDE interface with the title bar "BareMinimum | Arduino IDE 2.3.5". The central code editor window displays the "BareMinimum.ino" sketch. The code is as follows:

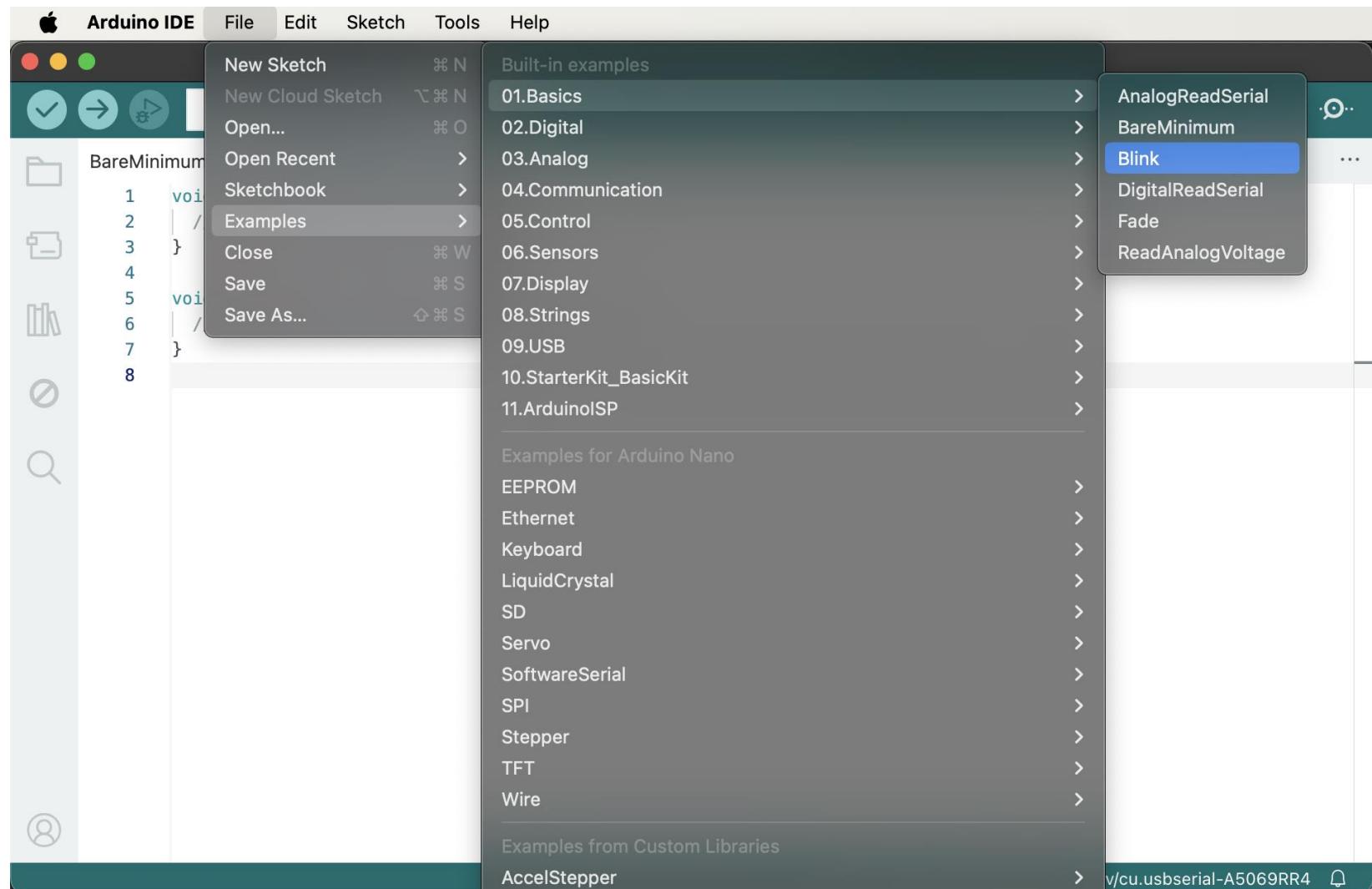
```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino IDE (integrated development environment) software

Start with Example

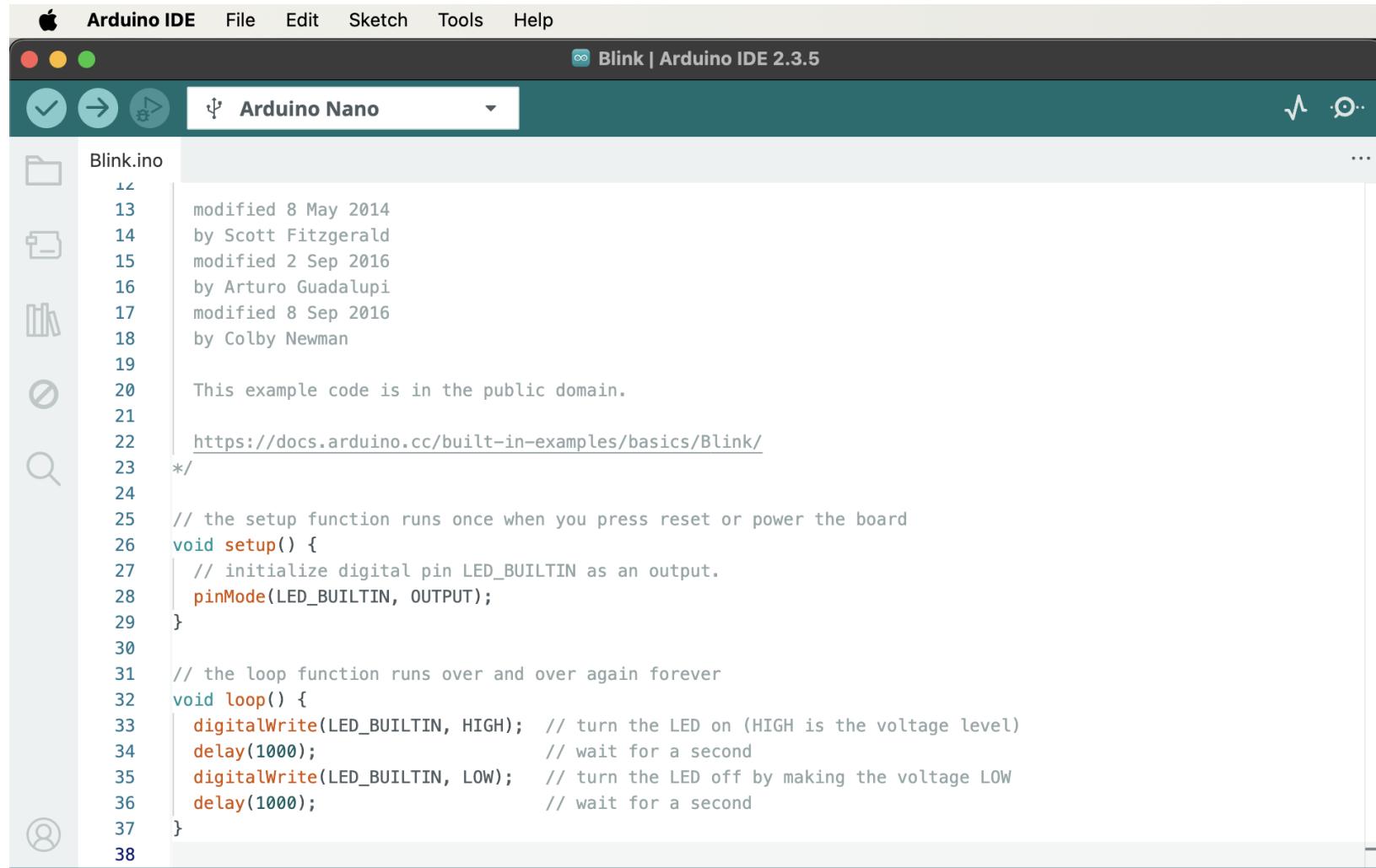
Arduino IDE
V
File
V
Example
V
Basics
V
Blink



Arduino IDE (integrated development environment) software

Start with Example

Arduino IDE
V
File
V
Example
V
Basics
V
Blink



The screenshot shows the Arduino IDE interface with the title bar "Arduino IDE" and version "2.3.5". The central area displays the "Blink" example sketch. The code is as follows:

```
Arduino Nano

Blink.ino
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

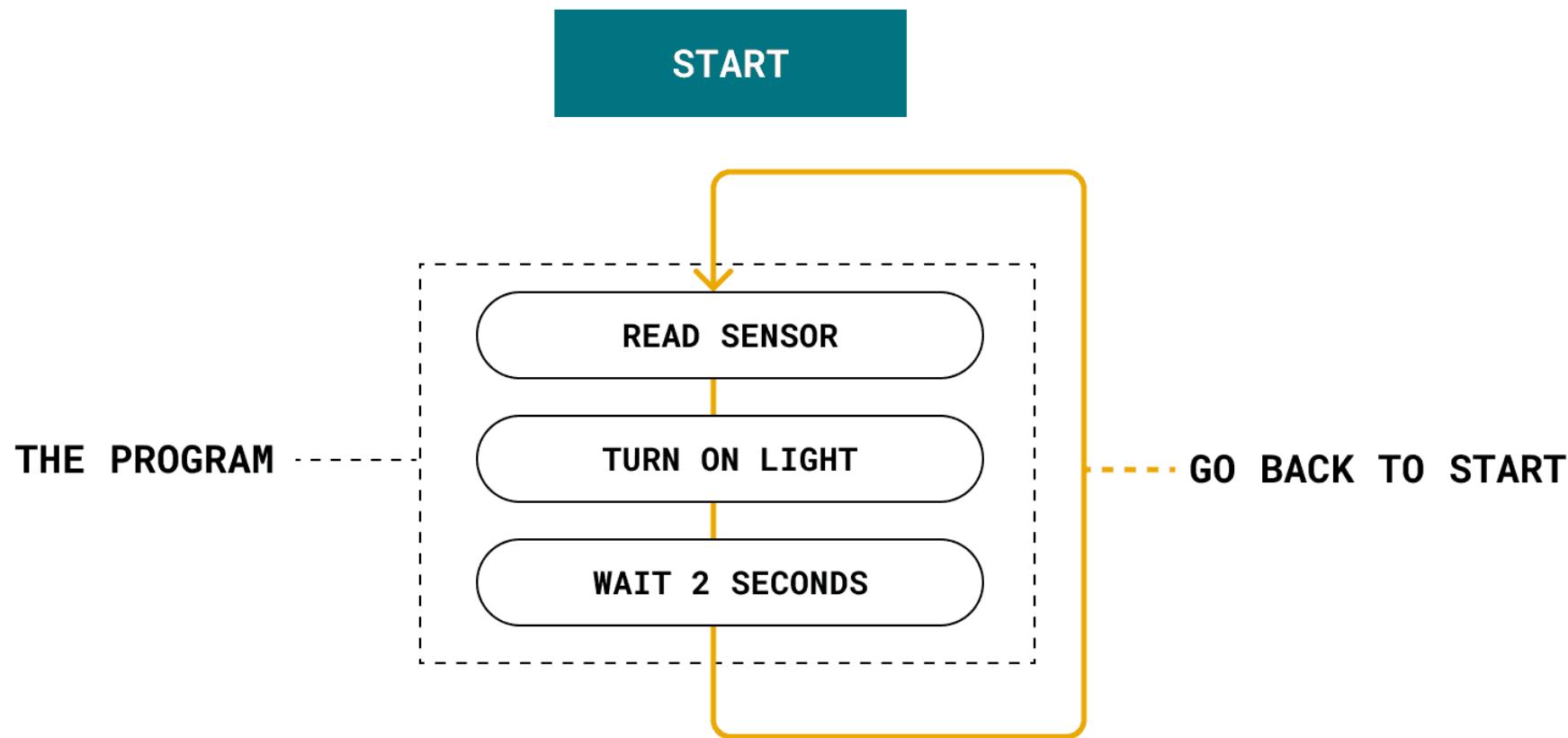
https://docs.arduino.cc/built-in-examples/basics/Blink/
 */

// 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
}
```

How Arduino works

Simple example with LED (light On and OFF for specific period of time)



Project: Blinking internal LED & external LED

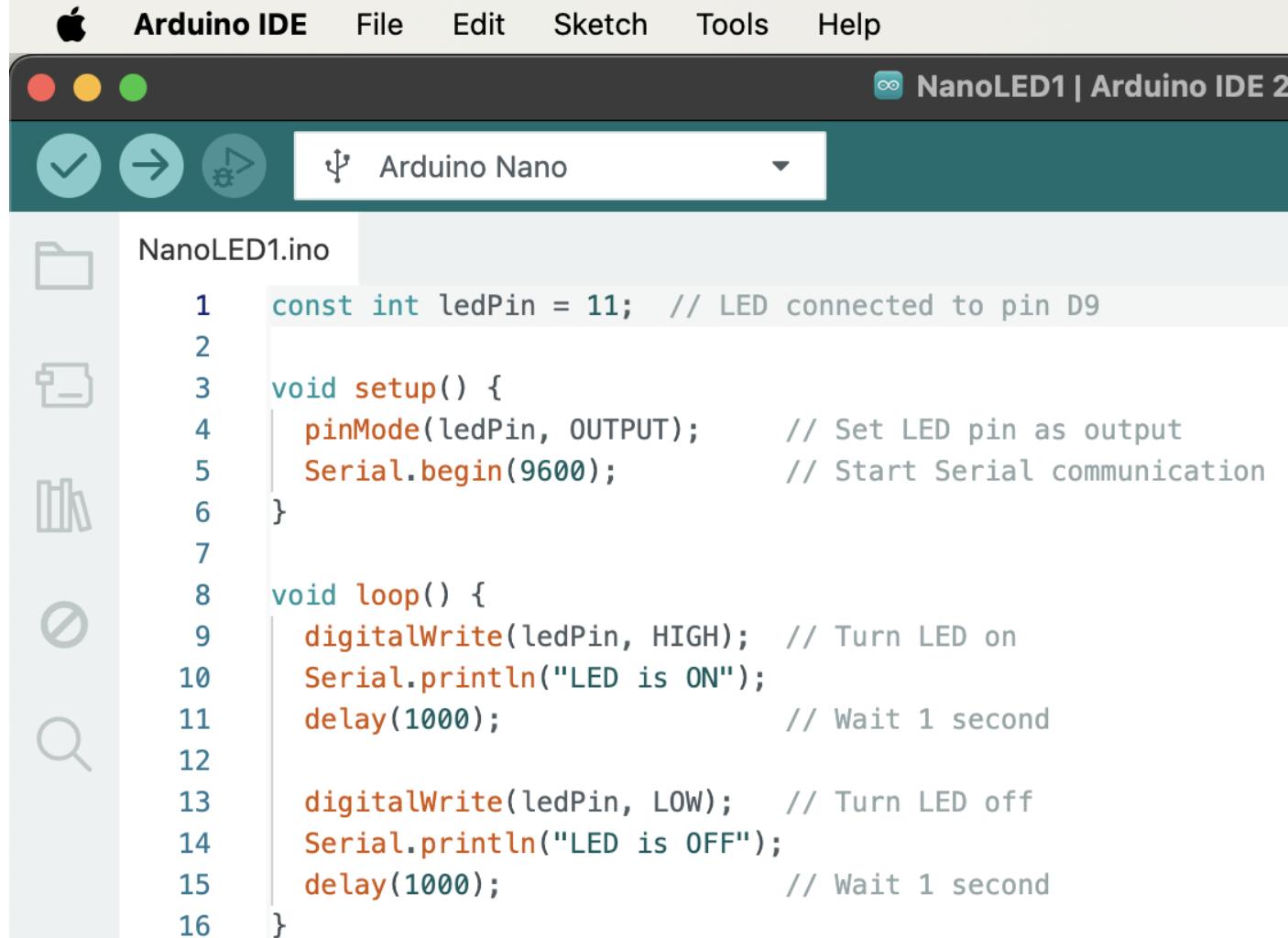
Start with Arduino Nano + LED: <https://github.com/shahinomah/arduinobasic/blob/main/NanoLED1.ino>

Activities:

Please change the delay time from 1000 ms to 500 ms

Try 5000 ms too

What is the outcome?



The screenshot shows the Arduino IDE interface on a Mac OS X system. The title bar reads "Arduino IDE" and "NanoLED1 | Arduino IDE 2". The main window displays the code for "NanoLED1.ino". The code is as follows:

```
const int ledPin = 11; // LED connected to pin D9

void setup() {
  pinMode(ledPin, OUTPUT); // Set LED pin as output
  Serial.begin(9600); // Start Serial communication
}

void loop() {
  digitalWrite(ledPin, HIGH); // Turn LED on
  Serial.println("LED is ON");
  delay(1000); // Wait 1 second

  digitalWrite(ledPin, LOW); // Turn LED off
  Serial.println("LED is OFF");
  delay(1000); // Wait 1 second
}
```

Download Code on Github

Link: <https://github.com/shahinomah/arduinobasic/>

The screenshot shows a GitHub repository page for the user 'shahinomah' named 'arduinobasic'. The repository is public and contains 11 commits. The code section lists several Arduino sketch files (INO) with their descriptions and commit times. The 'About' section includes a brief description 'Learn Basic Arduino', statistics (0 stars, 0 forks, 0 watching), and links to README, Activity, and package publishing. The 'Releases' and 'Languages' sections are currently empty.

Code

shahinomah / arduinobasic

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

arduinobasic Public

main 1 Branch 0 Tags

Go to file Add file Code

About

Learn Basic Arduino

Readme

Activity

0 stars

0 watching

0 forks

No releases published

Create a new release

No packages published

Publish your first package

Languages

File	Description	Commit
NanoBuzz1.ino	Arduino Nano + Buzzer	yesterday
NanoLED1.ino	Arduino Nano + LED (pin D11)	2 days ago
NanoLED1_POT1.ino	Arduino Nano + LED + Potentiometer	yesterday
NanoLED_LDR0.ino	Arduino Nano + LED + LDR	yesterday
NanoSonic.ino	Arduino Nano + Ultrasonic	yesterday
NanoSonicOLED-Buzz.ino	Arduino Nano + Ultrasonic + OLED Display + Buzzer	yesterday
NanoSonicOLED-BuzzLED.ino	Arduino Nano + Ultrasonic + OLED + Buzzer + LED	yesterday
NanoSonicOLED.ino	Arduino Nano + Ultrasonic + OLED Display	yesterday
NanoTraffic.ino	Arduino Nano + LED (RED + YELLOW + GREEN)	2 days ago
NanoTraffic_OLED.ino	Arduino Nano + LED 3 COLORS + OLED DISPLAY	2 days ago

Project: DRL

Start with Arduino Nano + LED: <https://github.com/shahinomah/arduinobasic/blob/main/NanoLED1.ino>

Activities:

Please add LED by combining your Arduino Nano with LDR and LED to make Daytime running lights (DRL)



Daytime running lights (DRL) are lights on the front of a vehicle that automatically switch on when the engine is started, designed to increase visibility to other road users during daylight hours. They are not intended to illuminate the road for the driver, but rather to make the vehicle more noticeable.

Project: Traffic Light

Start with Arduino Nano + LED: <https://github.com/shahinomah/arduinobasic/blob/main/NanoTraffic.ino>

Activities:

Please use LEDs with 3 different colours and built a basic traffic light



Project: Traffic Light

Start with Arduino Nano + LED + OLED display:

https://github.com/shahinomah/arduinobasic/blob/main/NanoTraffic_OLED.ino

Activities:

Then, please add OLED display and add light status GO, Slow Down, and STOP



Project: Traffic Light

Start with Arduino Nano + LED + OLED display:

https://github.com/shahinomah/arduinobasic/blob/main/NanoTraffic_OLED.ino

Activities:

Then, please change the delay time of the LED so that it will imitate real traffic light

Green : 5 seconds, blink for 5 x

Yellow : 2 seconds

Red : 2 seconds



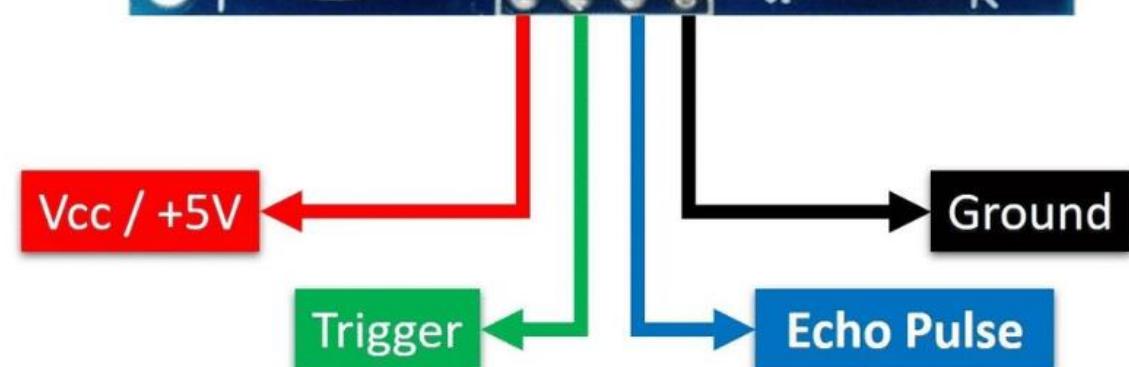
Ultrasonic / Proximity Sensor to measure distance

Start with Arduino Nano + Ultrasonic:

<https://github.com/shahinomah/arduinobasic/blob/main/NanoSonic.ino>

Activities:

Please use Ultrasonic
sensor to measure
distance / proximity

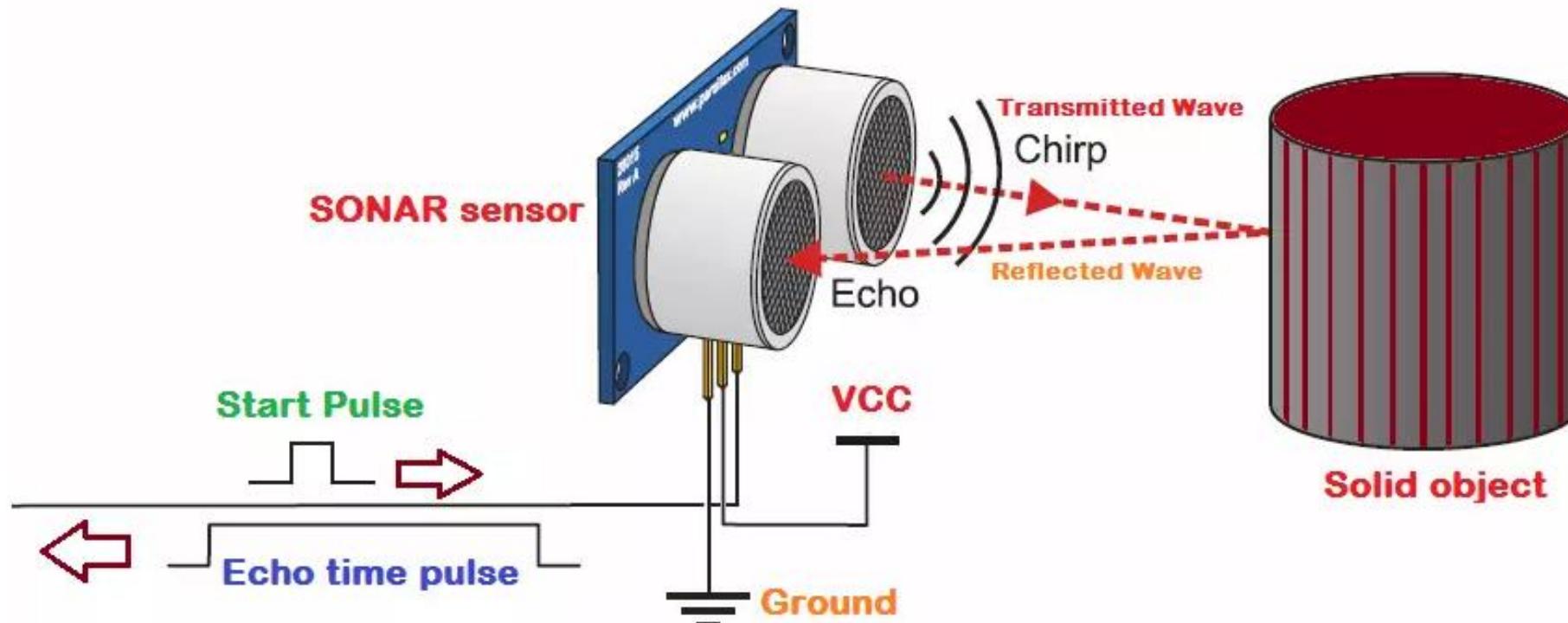


Ultrasonic / Proximity Sensor to measure distance

Start with Arduino Nano + Ultrasonic:

<https://github.com/shahinomah/arduinobasic/blob/main/NanoSonicOLED.ino>

Ultrasonic Sensor Working Principle



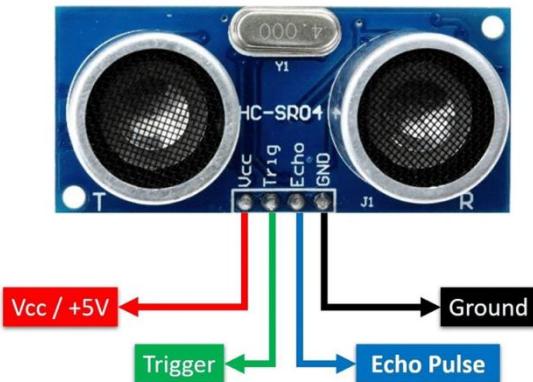
Project: Blind Spot Sensor

Start with Arduino Nano + Ultrasonic + LED:

<https://github.com/shahinomah/arduino-basic/blob/main/NanoSonicOLED-BuzzLED.ino>

Please add LED

Application:
Blind zone alert
(on car side mirror)



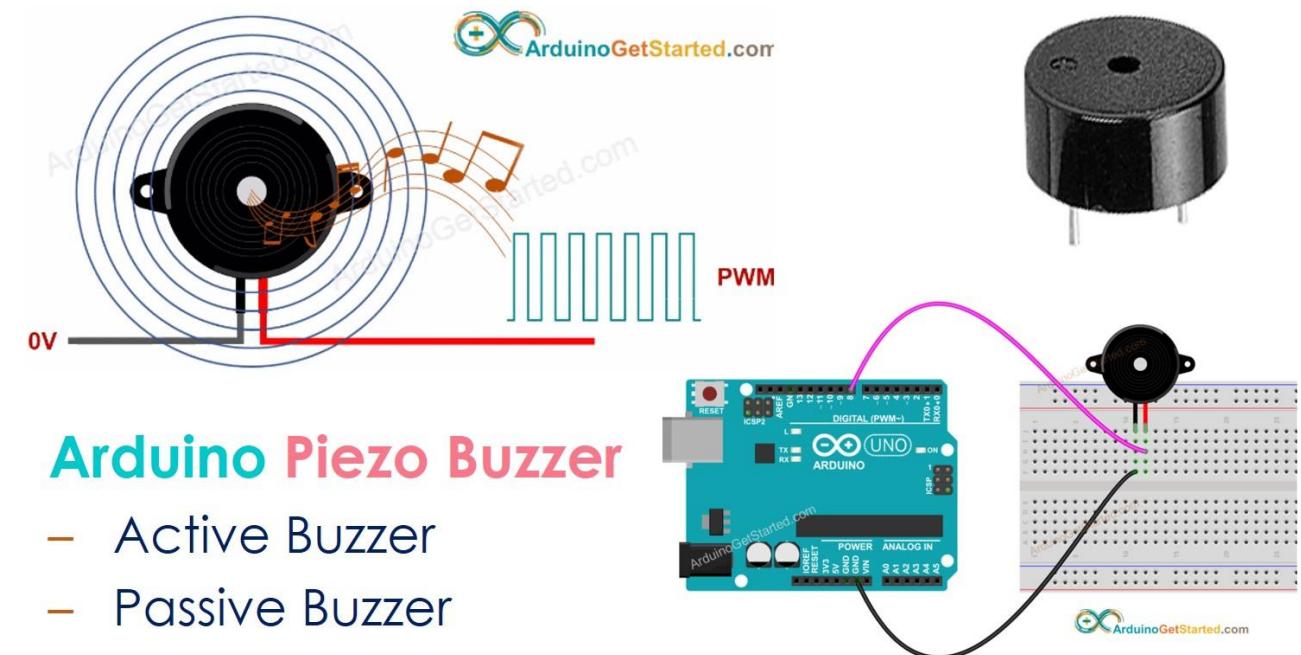
Project: Proximity with buzzer

Start with Arduino Nano + Ultrasonic + LED + Buzzer:

<https://github.com/shahinomah/arduinobasic/blob/main/NanoSonicOLED-BuzzLED.ino>

Activities:

Finally, add buzzer
Please make it beeping
for each second, and
continuous beep when
the distances is lower
than 15 cm



Project: Car Reverse Sensor

Start with Arduino Nano + Ultrasonic + LED + Buzzer:

<https://github.com/shahinomah/arduino-basic/blob/main/NanoSonicOLED-BuzzLED.ino>

Activities:

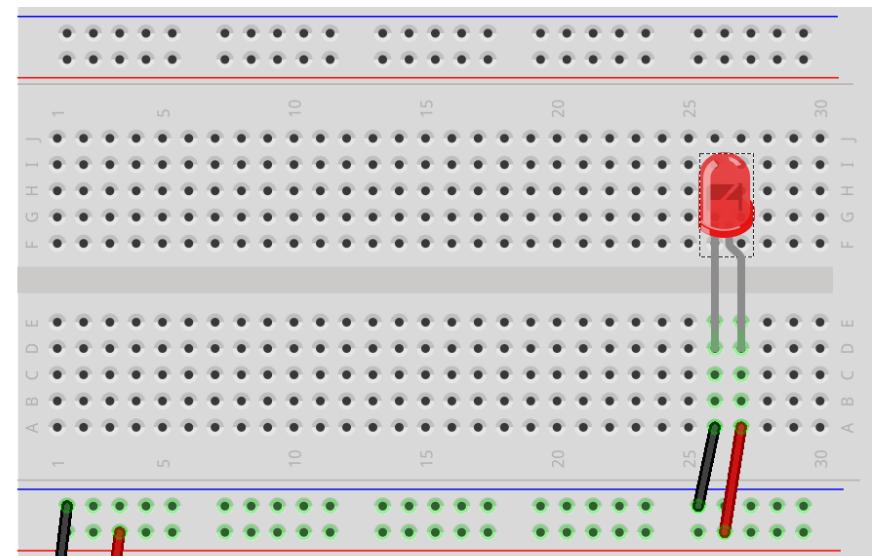
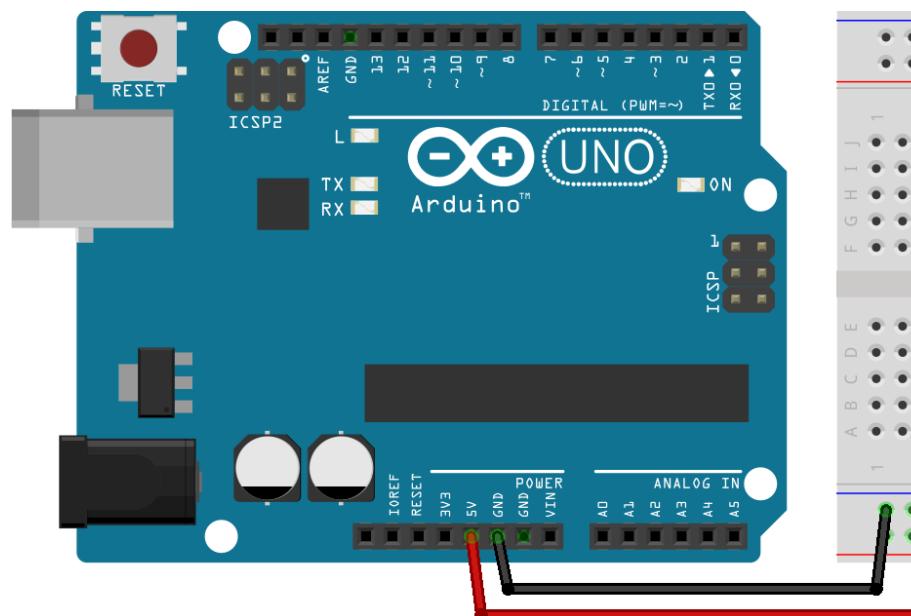
Car reverse
sensor with
Arduino Nano +
Ultrasonic +
OLED display+
LED + Buzzer



Fritzing software

Fritzing is a printed circuit software that permits you to do beautiful circuit for Arduino. Furthermore, you can export this circuit to an image.

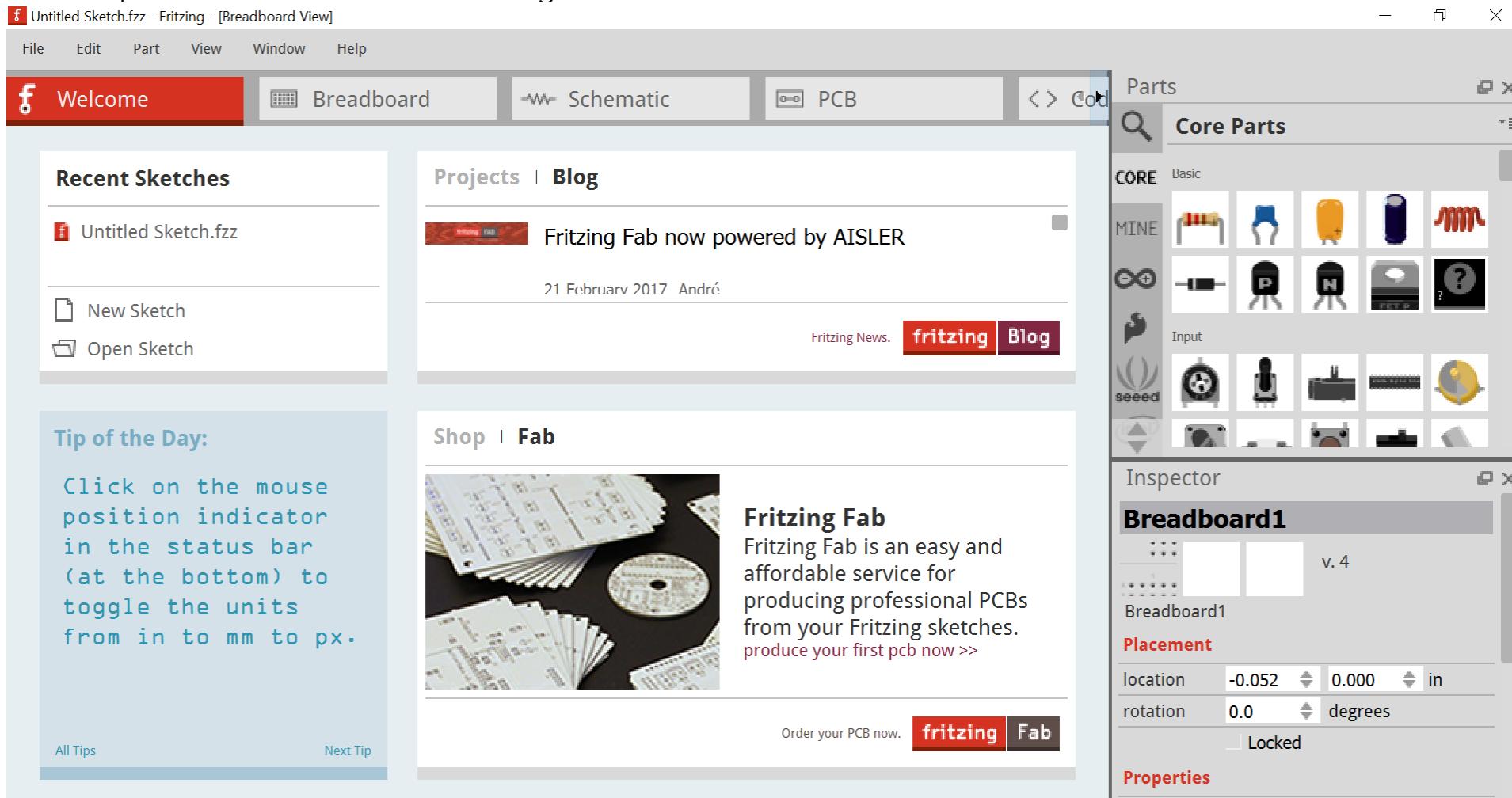
<https://arduinofactory.fr/en/download-fritzing-for-free/>



fritzing

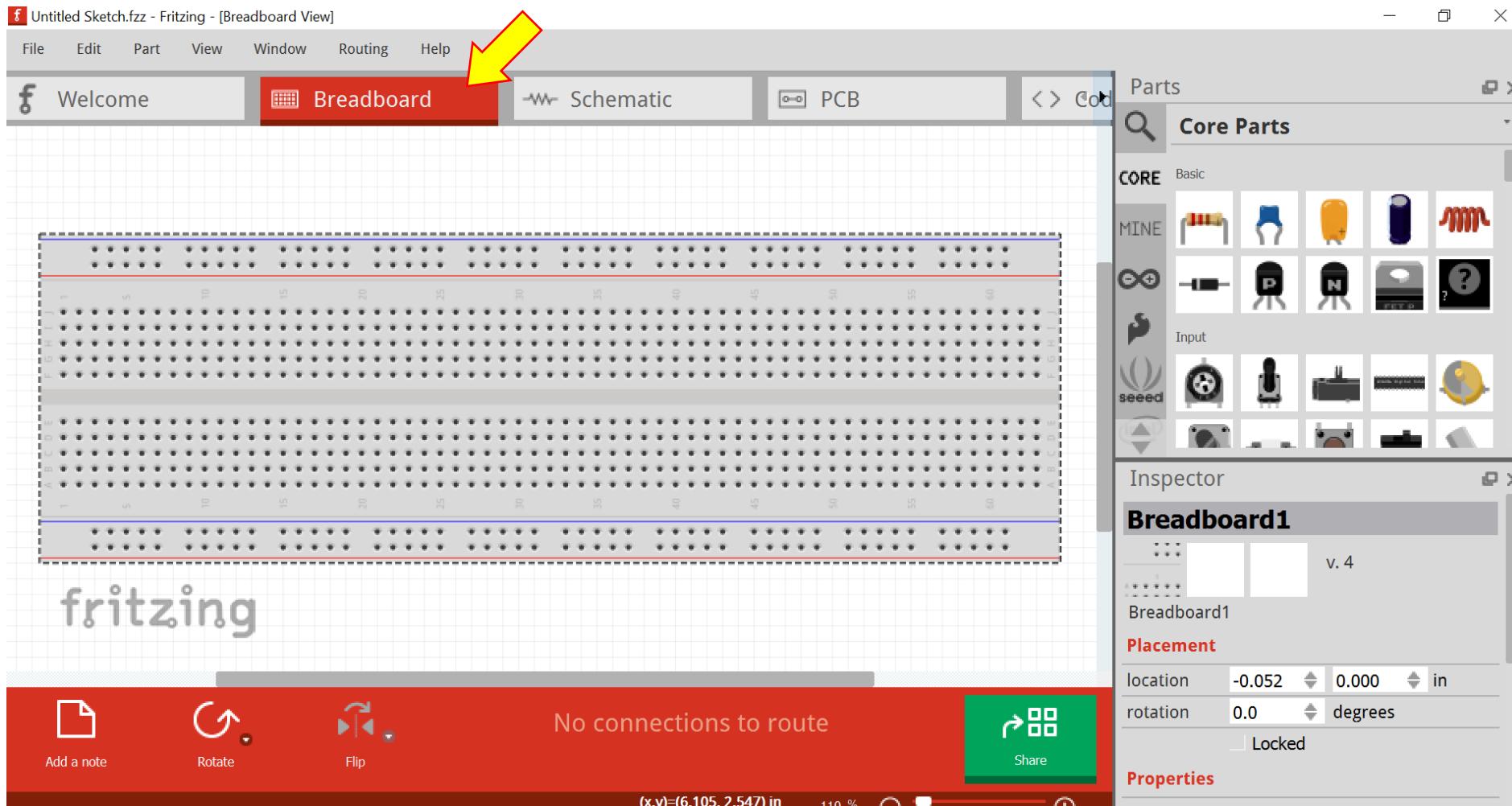
Fritzing software

Fritzing is a printed circuit software that permits you to do beautiful circuit for Arduino. Furthermore, you can export this circuit to an image.



Fritzing software

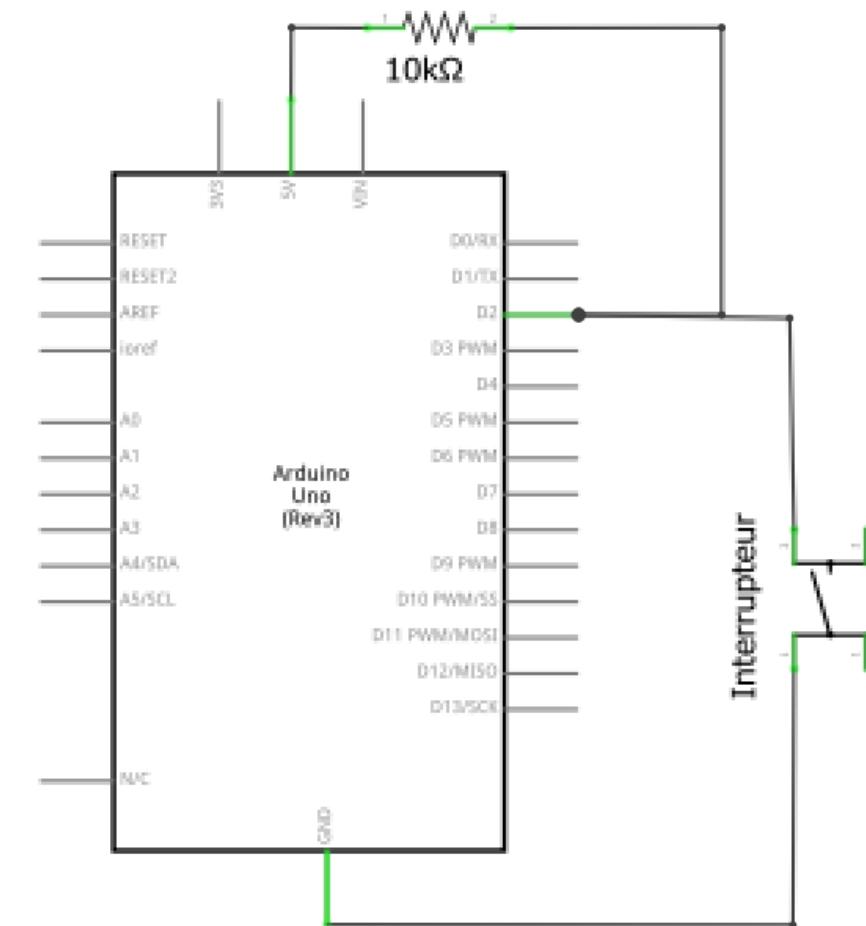
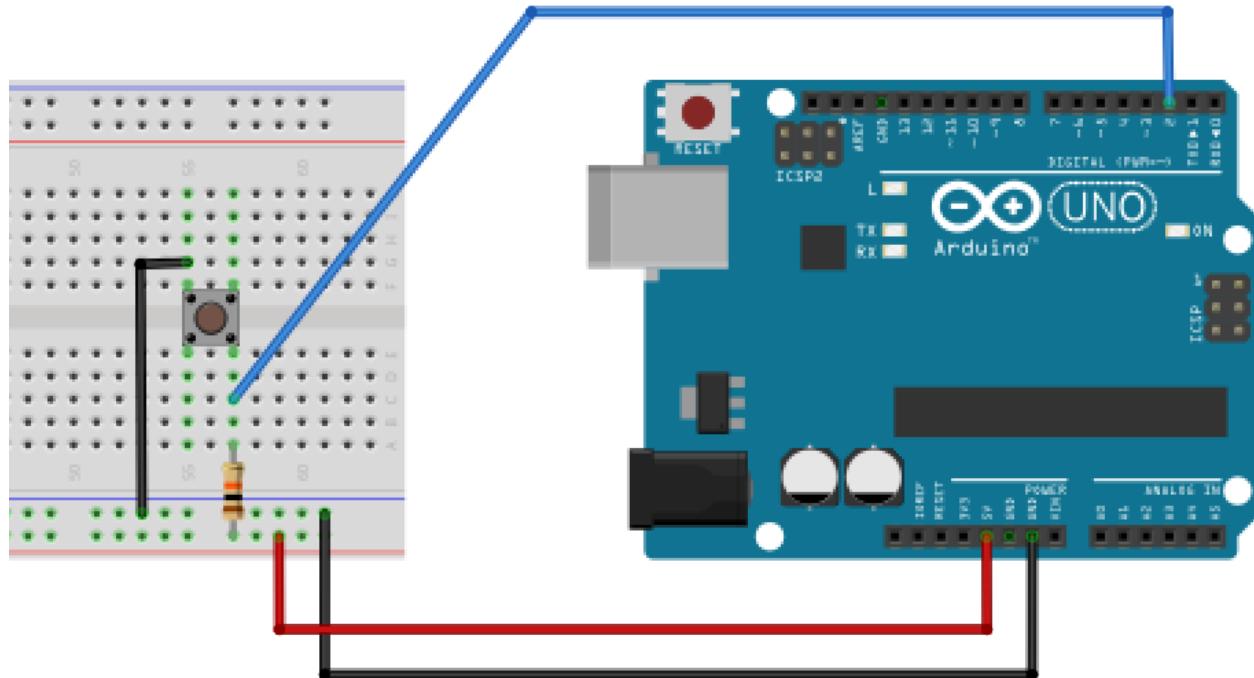
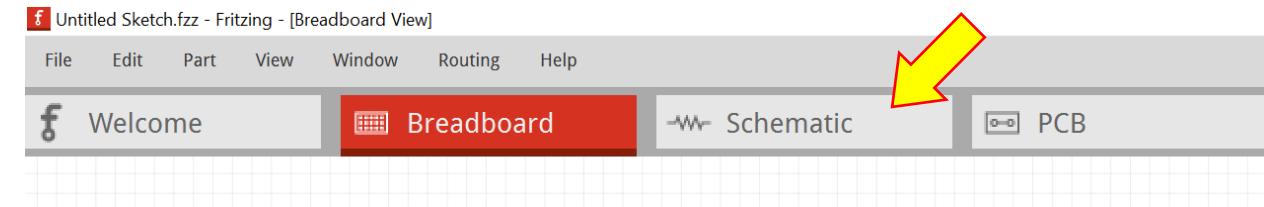
Fritzing is a printed circuit software that permits you to do beautiful circuit for Arduino. Furthermore, you can export this circuit to an image.



Drag and drop
From Parts into
Breadboard space

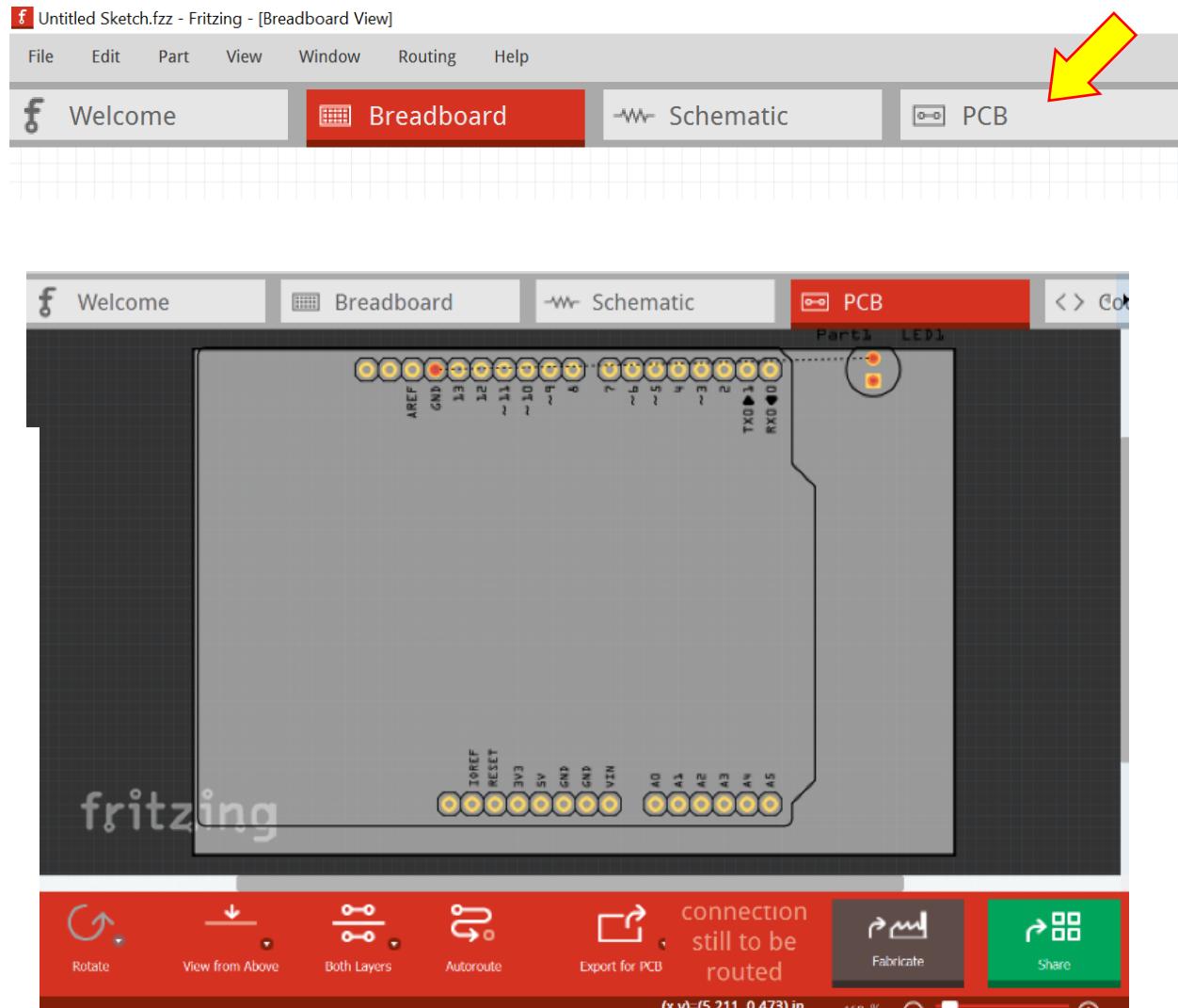
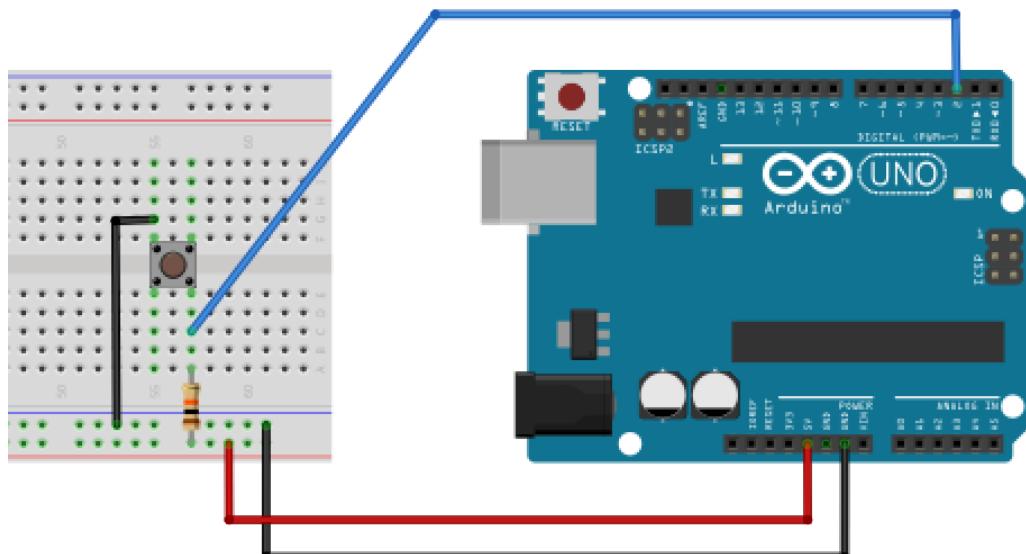
Fritzing software

Panel Tab → Schematic (Diagram)



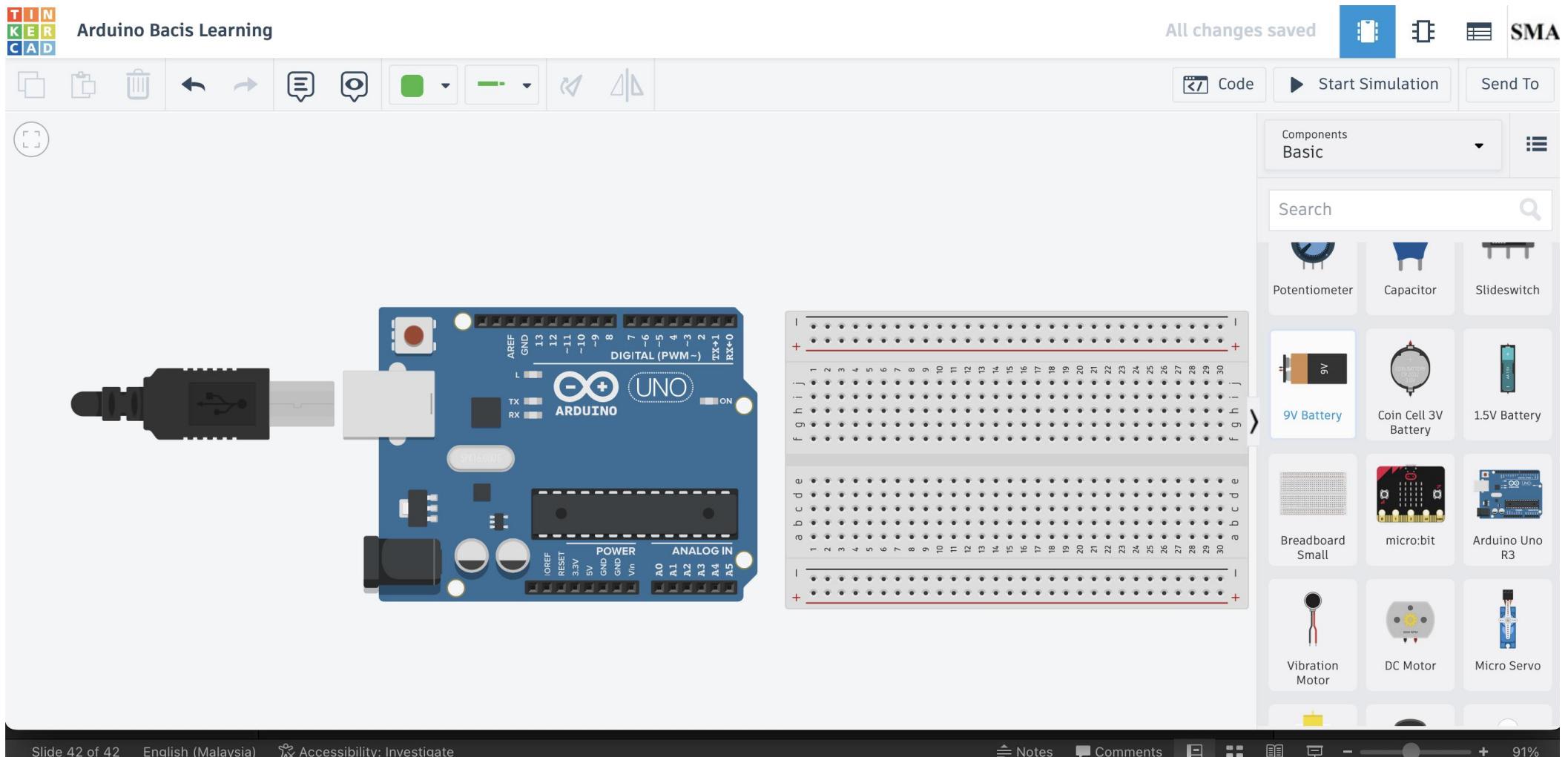
Fritzing software

Panel Tab → PCB
(to be used, submitted, and produced
by the manufacturer)



TinkerCAD - Circuit

Try to make and run Arduino Simulation using tinkercad circuit online application



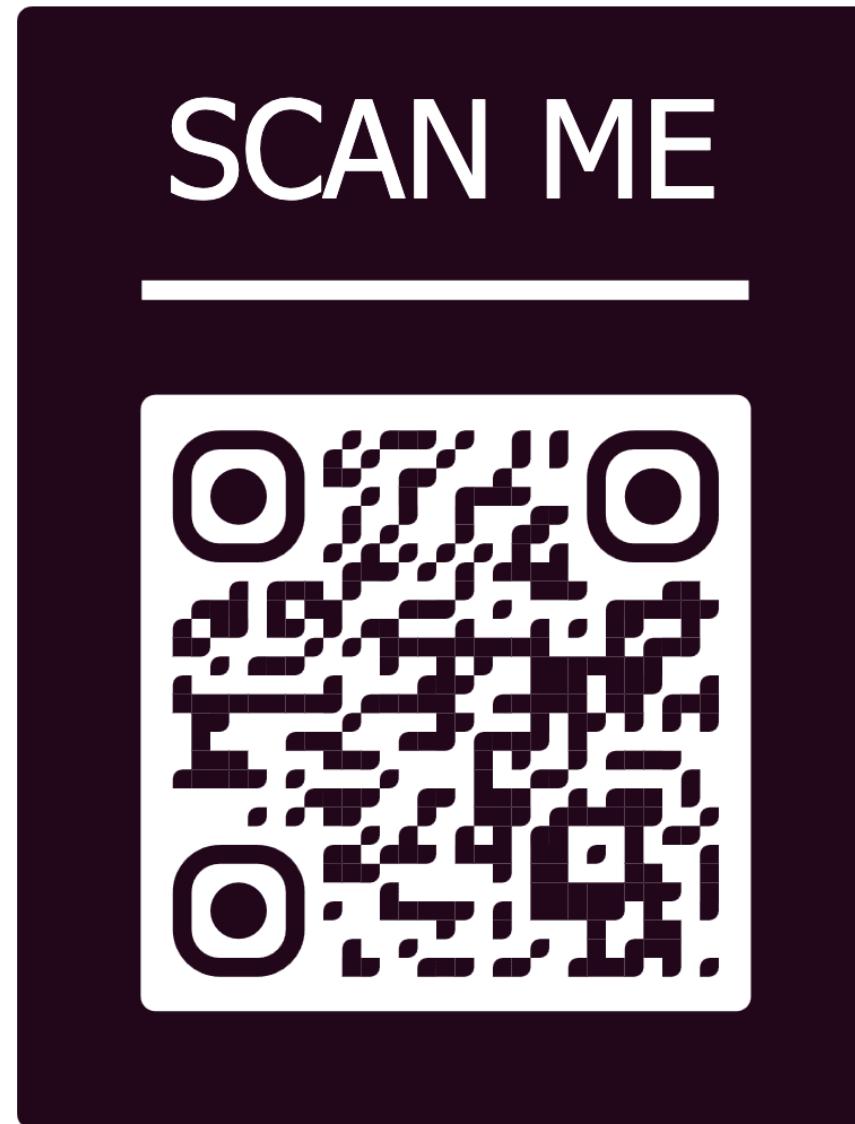
Thank you

Please fill in Feedback Form

Digital Lab Practicum: Borang Maklum Balas Bengkel Arduino (Dr Shahino Mah)

<https://forms.gle/V4APkL3Vik3vrcFXA>

Digital Lab Practicum: Borang Maklum Balas Bengkel Arduino (Dr Shahino Mah)



<https://forms.gle/V4APkL3Vik3vrcFXA>

Gallery



Gallery



shahinomah@gmail.com

0197883473

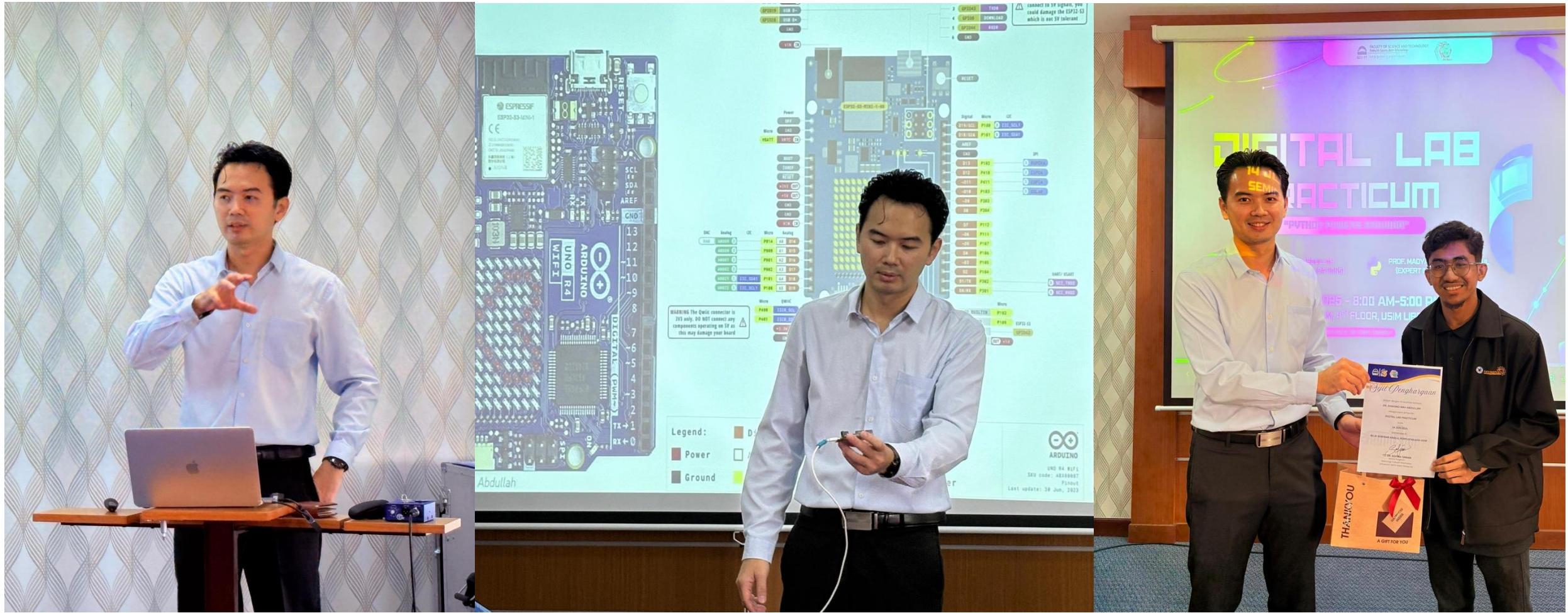
Dr Shahino Mah Abdullah

Thank you!

rafahusim



Gallery



Contact:

shahinomah@gmail.com

0197883473

Dr Shahino Mah Abdullah

Thank you!

rafahusim

