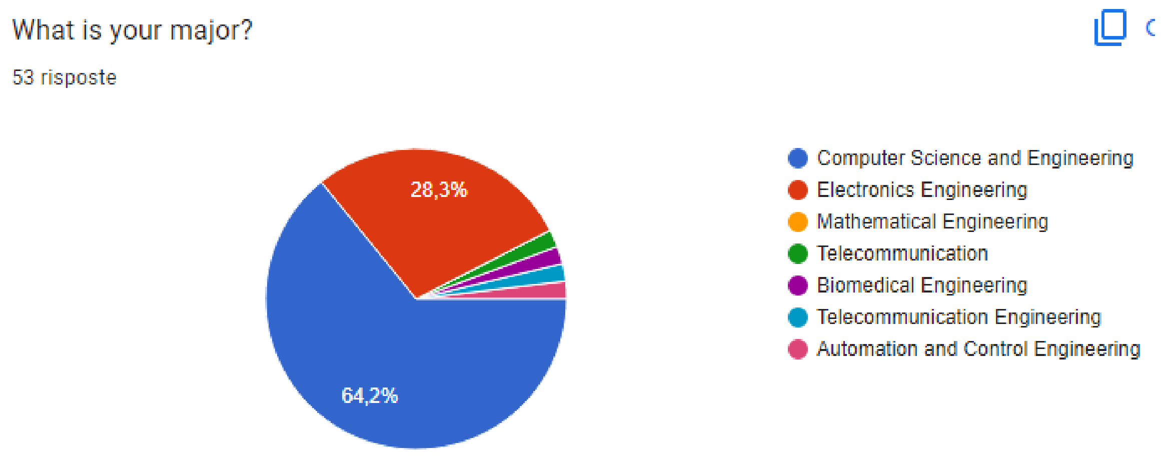
POLITECNICO
MILANO 1863

Hardware Architectures for Embedded and Edge AI

Prof Manuel Roveri – manuel.roveri@polimi.it
Massimo Pavan – massimo.pavan@polimi.it

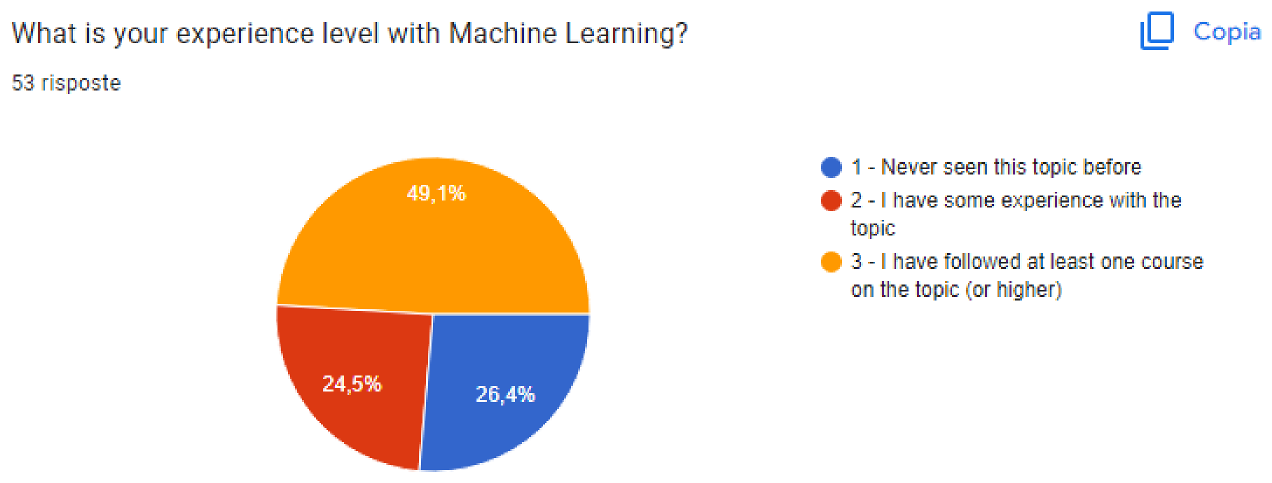
Exercise session 1 – Platform and firmware

Poll results and a disclaimer



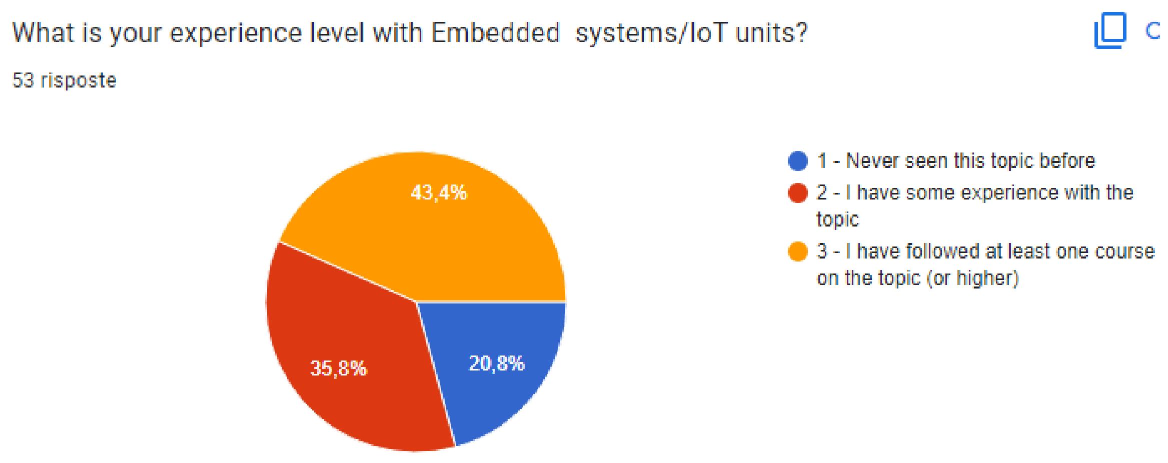
What is your major?

53 risposte



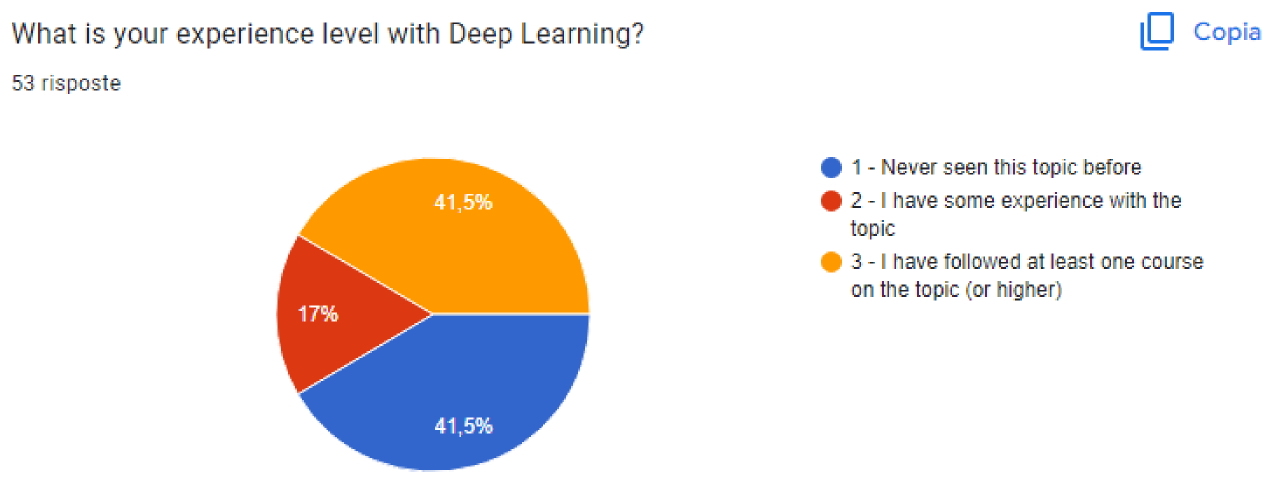
What is your experience level with Machine Learning?

53 risposte



What is your experience level with Embedded systems/IoT units?

53 risposte



What is your experience level with Deep Learning?

53 risposte

Pre-requisites

- Intermediate/basic knowledge of python coding
- Basic knowledge of C coding

Nice to have, but will be revised togheter:

- Basic knowledge of Embedded systems
- Basic knowledge of Machine Learning or Deep Learning

Exercise sessions

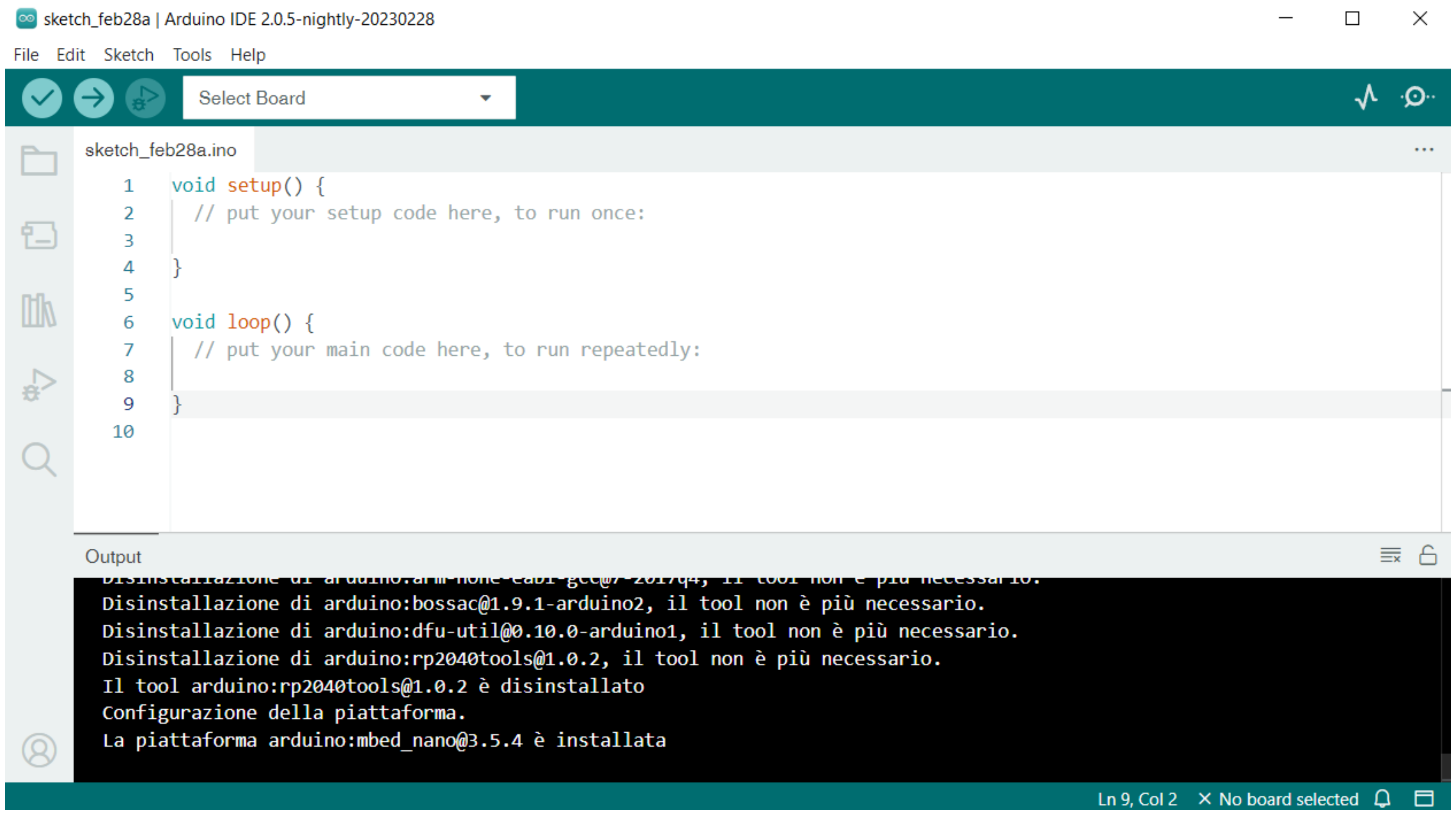
1. Intro/review on Embedded Systems
2. Intro/review on Deep Learning with Tensorflow
3. TF lite optimizations: quantization and pruning
4. Training Keyword Spotting – Microphone
5. Deploying Keyword Spotting – Microphone
6. Training and Deploying Visual Wake Word – Camera
7. Training and Deploying Anomaly detection – Accelerometer
8. Data Collection and Engineering with Edge Impulse – Camera
9. On-device learning of a Speaker Verification algorithm - Microphone

Resources

- Books
- Online course
- For python and C coding: any online tutorial/crash course should do it, as long as you have taken at least one coding course

The Arduino IDE

https://www.arduino.cc/en/software



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

}

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

}

```

```
Disinstallazione di arduino:bossac@1.9.1-arduino2, il tool non è più necessario.
Disinstallazione di arduino:dfu-util@0.10.0-arduino1, il tool non è più necessario.
Disinstallazione di arduino:rp2040tools@1.0.2, il tool non è più necessario.
Il tool arduino:rp2040tools@1.0.2 è disinstallato
Configurazione della piattaforma.
La piattaforma arduino:mbed_nano@3.5.4 è installata
```

Embedded systems in a nutshell

An embedded system is a combination of hardware and software designed for a specific function.

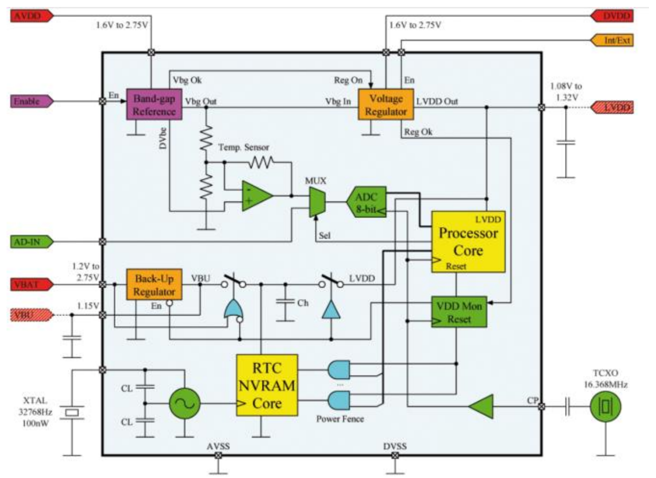
The systems can be programmable or have fixed functionalities.

There are three distinct approaches to develop an Embedded system:

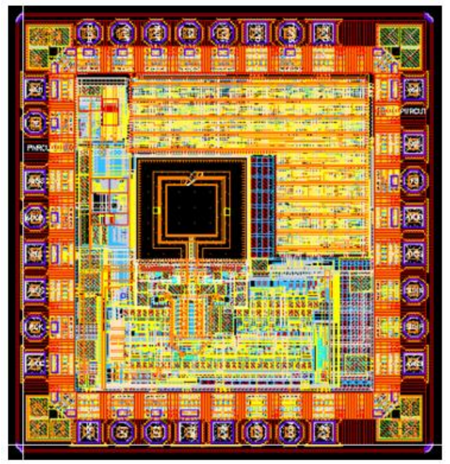
- ASIC (Application Specific Integrated Circuit)
- FPGA (Field-Programmable Gate Array)
- uC (Microcontroller)

ASIC – Application Specific Integrated Circuit

- designed for a particular application
- provide the best performance
- Extremely expensive to design and manufacture (this restrict them to very large volume applications)



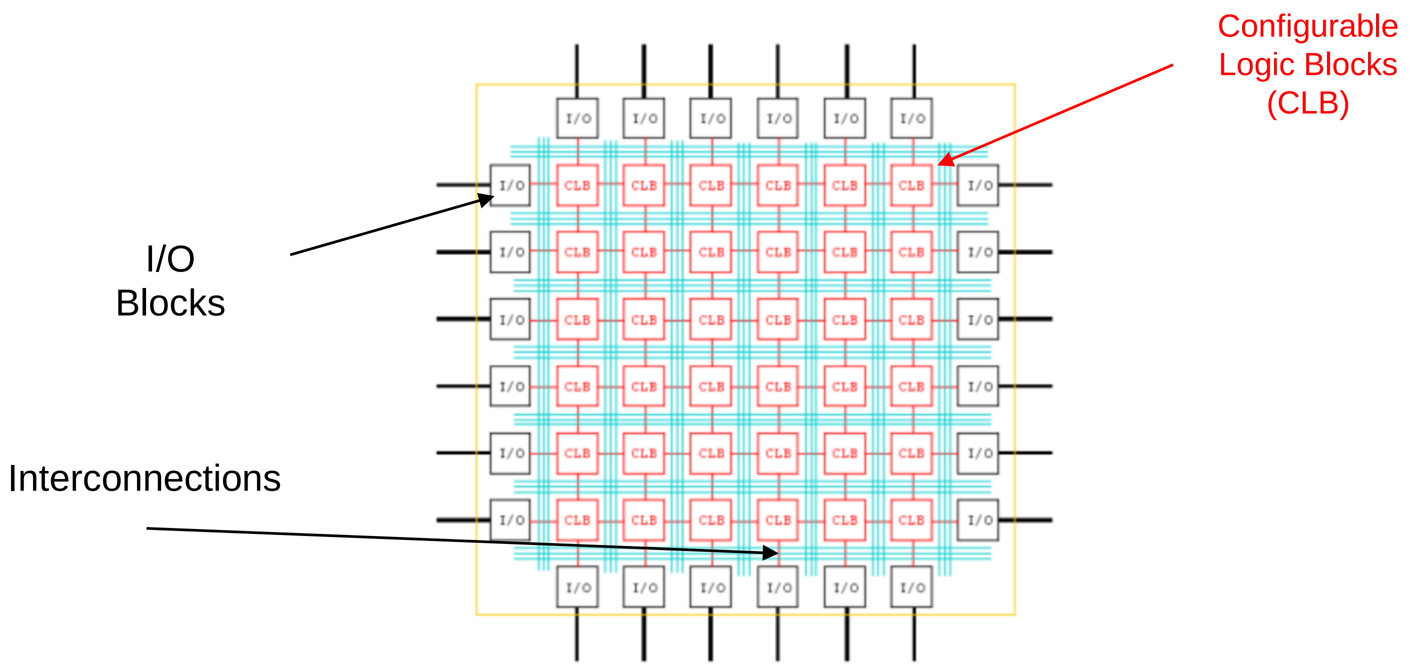
Schematic



Layout

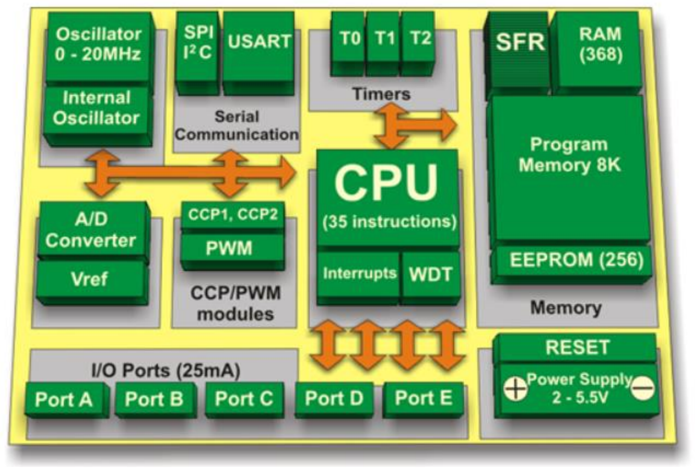
FPGA – Field Programmable Gate Array

- A package integrated circuit with an array of gates and flip-flops,
- They can be connected by "programming" the device to produce the desired function
- This is specified using a hardware description language such as VHDL or Verilog



MCU - Microcontroller Units

- Have a fixed hardware built around a central processing unit (CPU)
- The CPU controls a range of peripherals, both digital and analog functions
- Small devices usually include both volatile and non-volatile memory on the chip
- Larger processors may need separate memory
- Usually programmed using a machine-language like Assembly or high-level language such as C

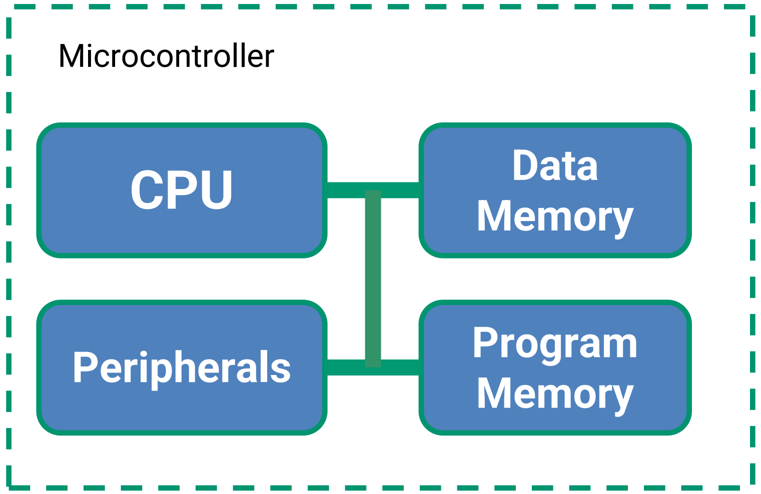


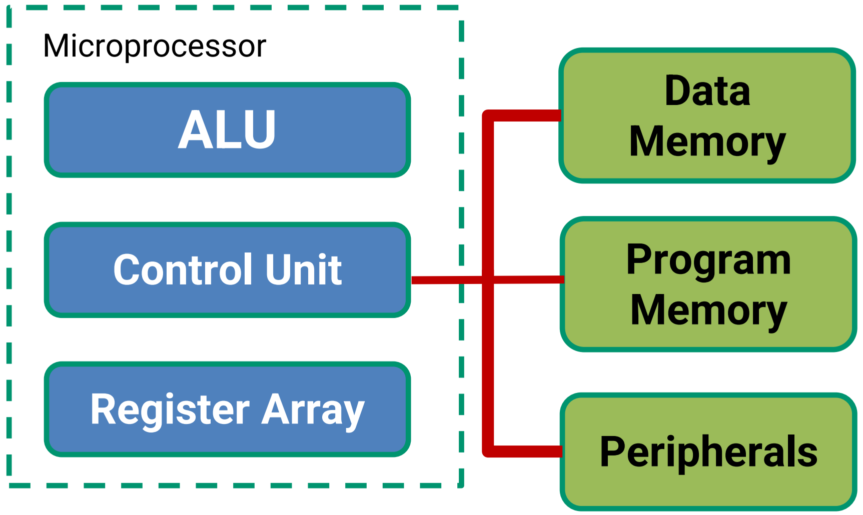
Microcontrollers are very interesting candidates for embedded systems, because their on-chip peripherals and memory enable the embedded system designers to save circuit space and overall dimensions, by not having to add these things on the board.

Microcontrollers vs Microprocessors

- A **microcontroller** contains:
  - A microprocessor (uP)
  - Program Memory
  - Data Memory
  - A number of internal peripherals devices (e.g., timers, serial ports, GPIO pins, ADC converters, etc.

  all inside a single silicon chip.
- A **microprocessor** (uP) is a silicon chip containing an advanced Central Processing Unit (CPU) that fetches program instructions from an external memory and executes them.

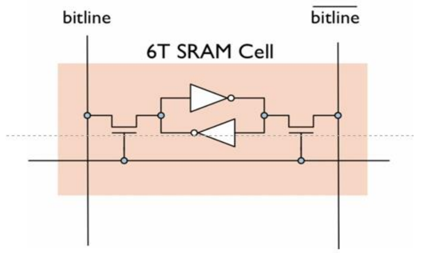


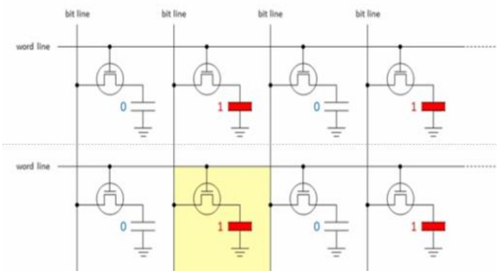


Memories

Volatile memories : loses their content when power is removed

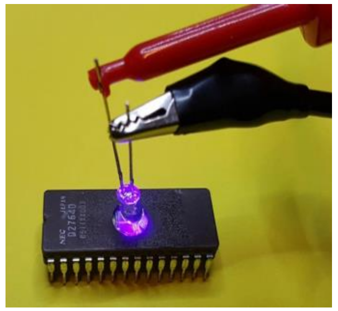
- SRAM (Static Random Access Memory)
  - Fast but takes a lot of space (6 transistors)
- DRAM (Dynamic Random-Access Memory)
  - little space (1 transistor and 1 capacitor) but slow, requires periodic refresh





Non-volatile memories:

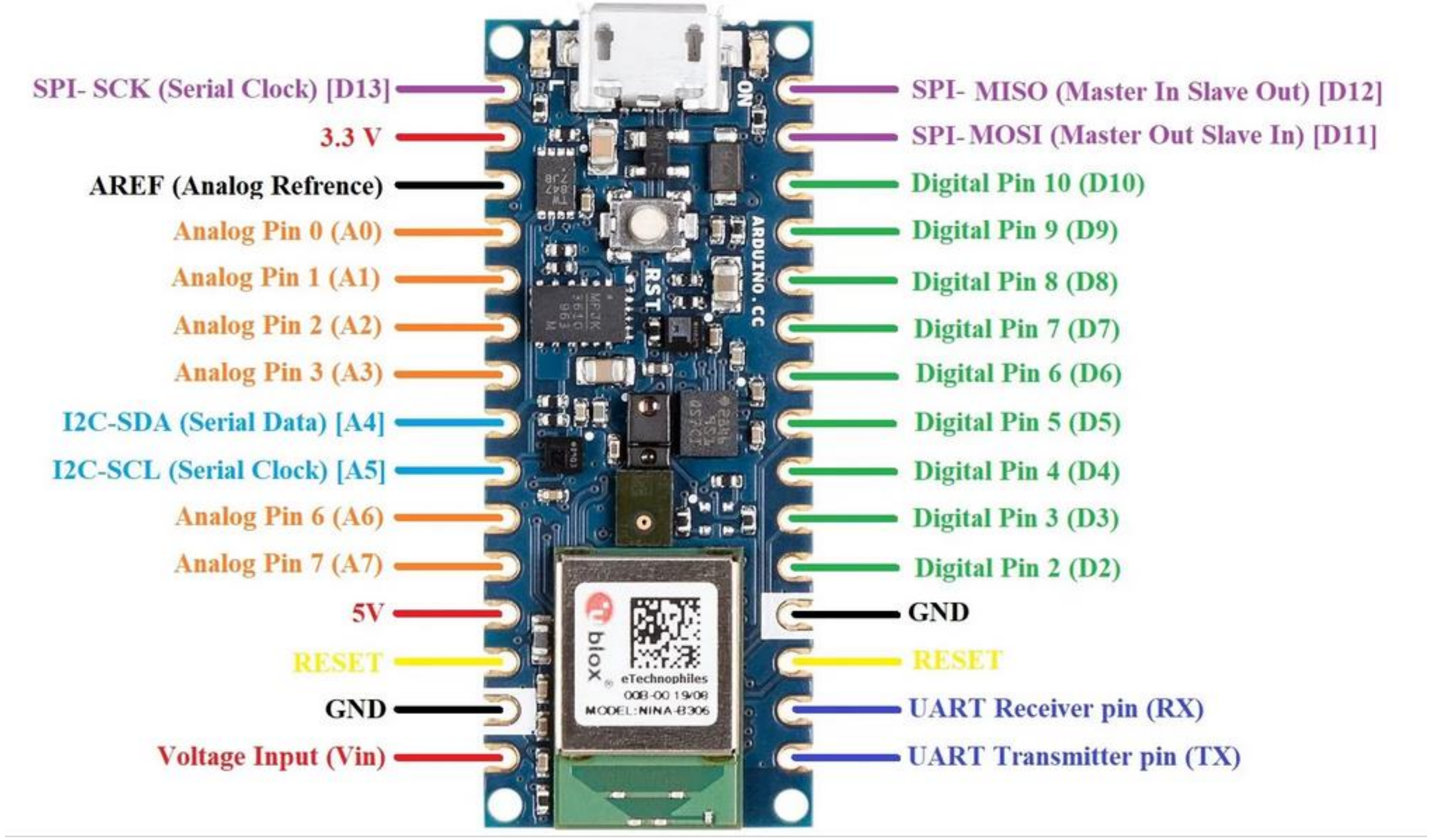
- Mask ROM (Read Only Memory)
  - contents are programmed during the IC manufacturing.
  - any change to the data requires a new mask to be produced at great expense.
- EPROM (Electrically Programmable ROM)
  - It can be programmed electrically, but cannot erase them. To clear the content, must be exposed to UV
- Flash Memory
  - Similar to the EPROM but flash can be electrically erased in blocks
  - 2 types: NAND (smallest, sequential access) and NOR (larger, sequential or random access)
  - Many microcontrollers use NOR flash, which is slower to write but permits random access



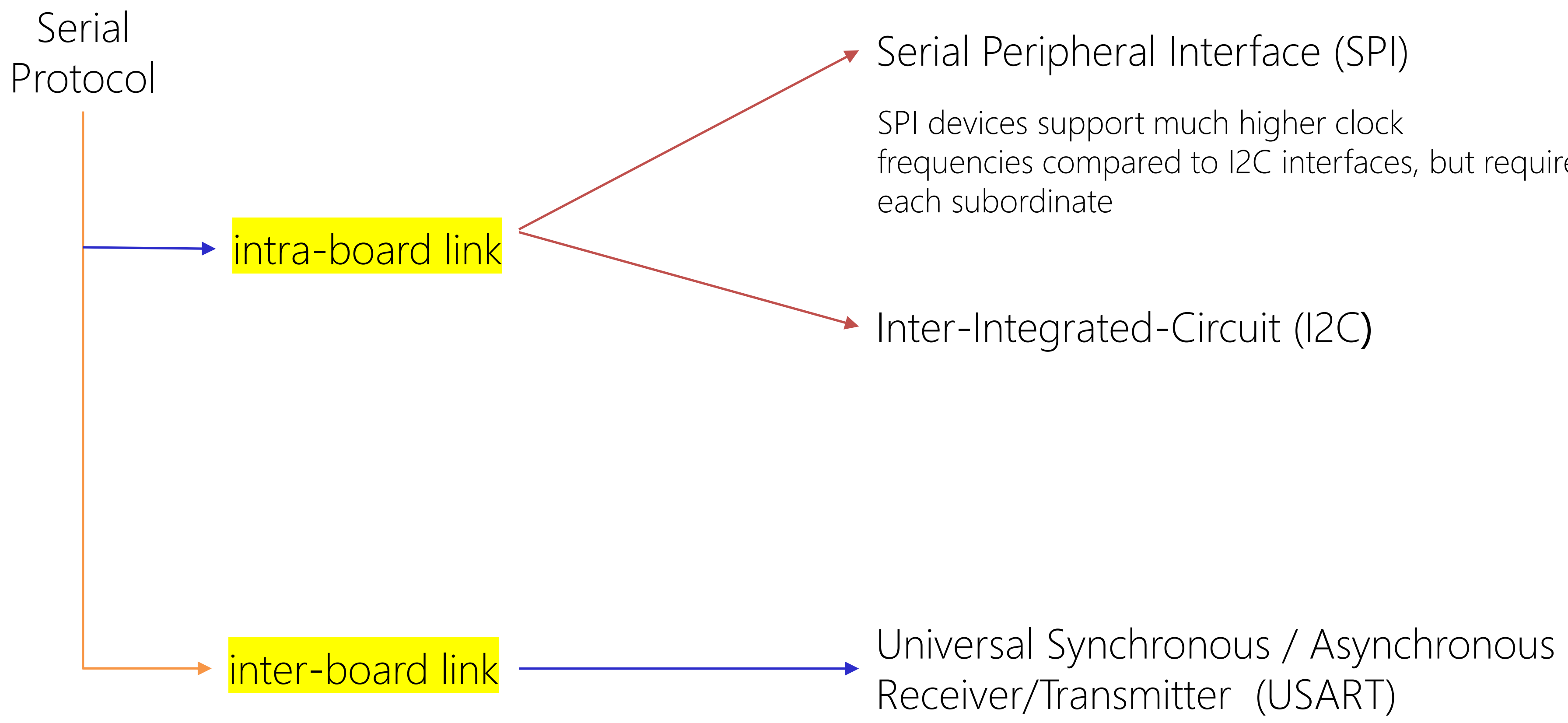
I/O Ports

- Input and Output ports are the simplest example of interface toward the outside the world.
- The single pin I/O can be an input(I) or an output (O) of the microcontroller depending on how the firmware sets the port.
- The single IO can be configured and used in different ways :
  - Digital Input : can be a normal CMOS input for example
  - Interrupt : Similar to the digital input, but it can also be enabled to generate an IRQ to the uC core
  - Analog Input : allow to acquire analog signals and send them to the on-chip Analog to Digital Converter (ADC) inside the uC
  - Digital Output: usually a CMOS levels, sometime open-drain

I/O ports on the Arduino nano BLE 33 sense



Serial Communication devices


Serial Protocol
intra-board link
inter-board link
Serial Peripheral Interface (SPI)
Inter-Integrated-Circuit (I2C)
Universal Synchronous / Asynchronous Receiver/Transmitter (USART)

SPI devices support much higher clock frequencies compared to I2C interfaces, but require one CS for each subordinate

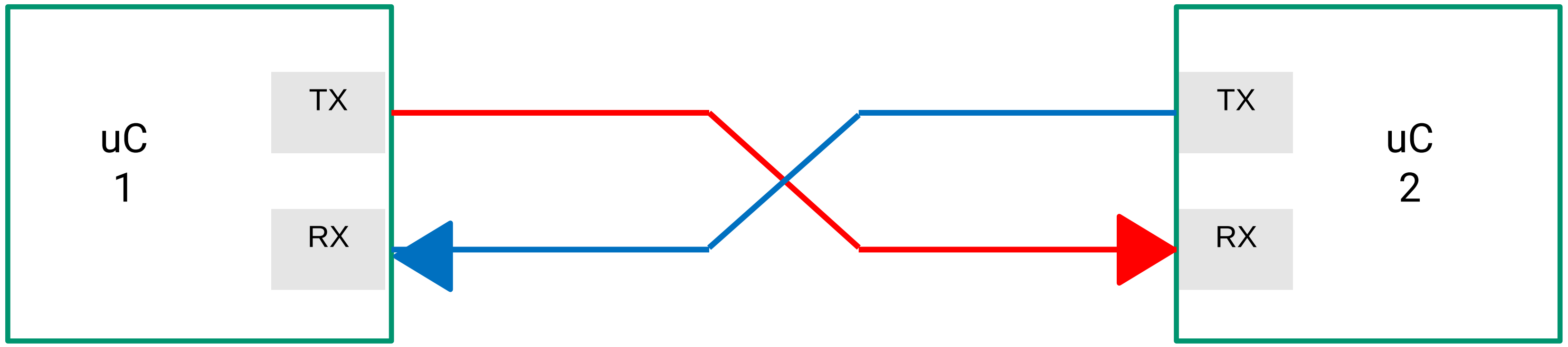
UART (Universal Asynchronous Receiver/Transmitter)

UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed.

Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end.

The UART use signals:

- TX Transmitter
- RX Receiver



UART: Data transmission

Start Bit	Data Frame	Parity Bit	Stop
(1 bit)	(5 to 8 Data bits)	(0 to 1 bit)	(1 to 2 bit)

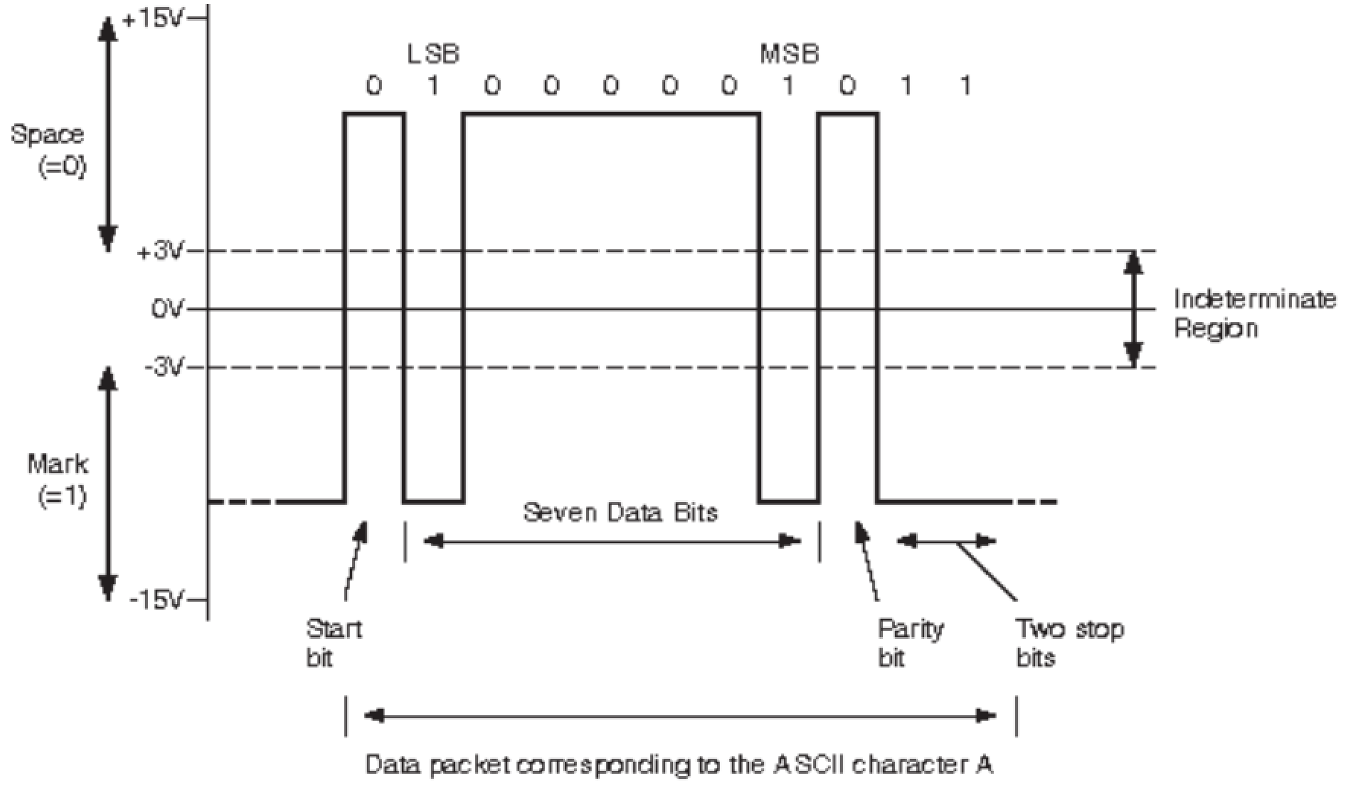
- Start Bit
  - To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one (1) clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate
- Data Frame
  - The data frame contains the actual data being transferred. It can be five (5) bits up to eight (8) bits long if a parity bit is used, even 9 otherwise
- Parity Bit
  - Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during transmission. Bits can be changed by electromagnetic radiation, mismatched baud rates, or long- distance data transfers.
- Stop Bit
  - To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for one (1) to two (2) bit(s) duration.

RS-232 standard

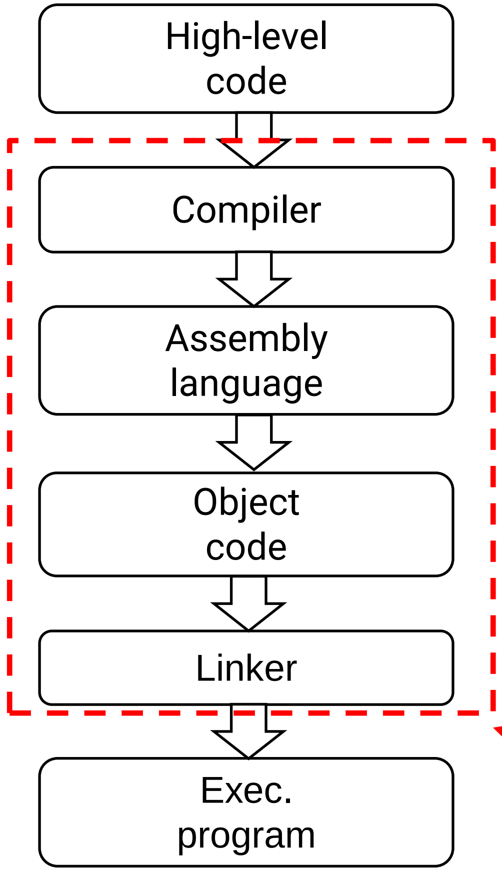
- Map the digital value with two voltages level

“0” with +12V (all levels between +3V to +15V)

“1” with -12V (all levels between -3V to -15V)



Programming the board

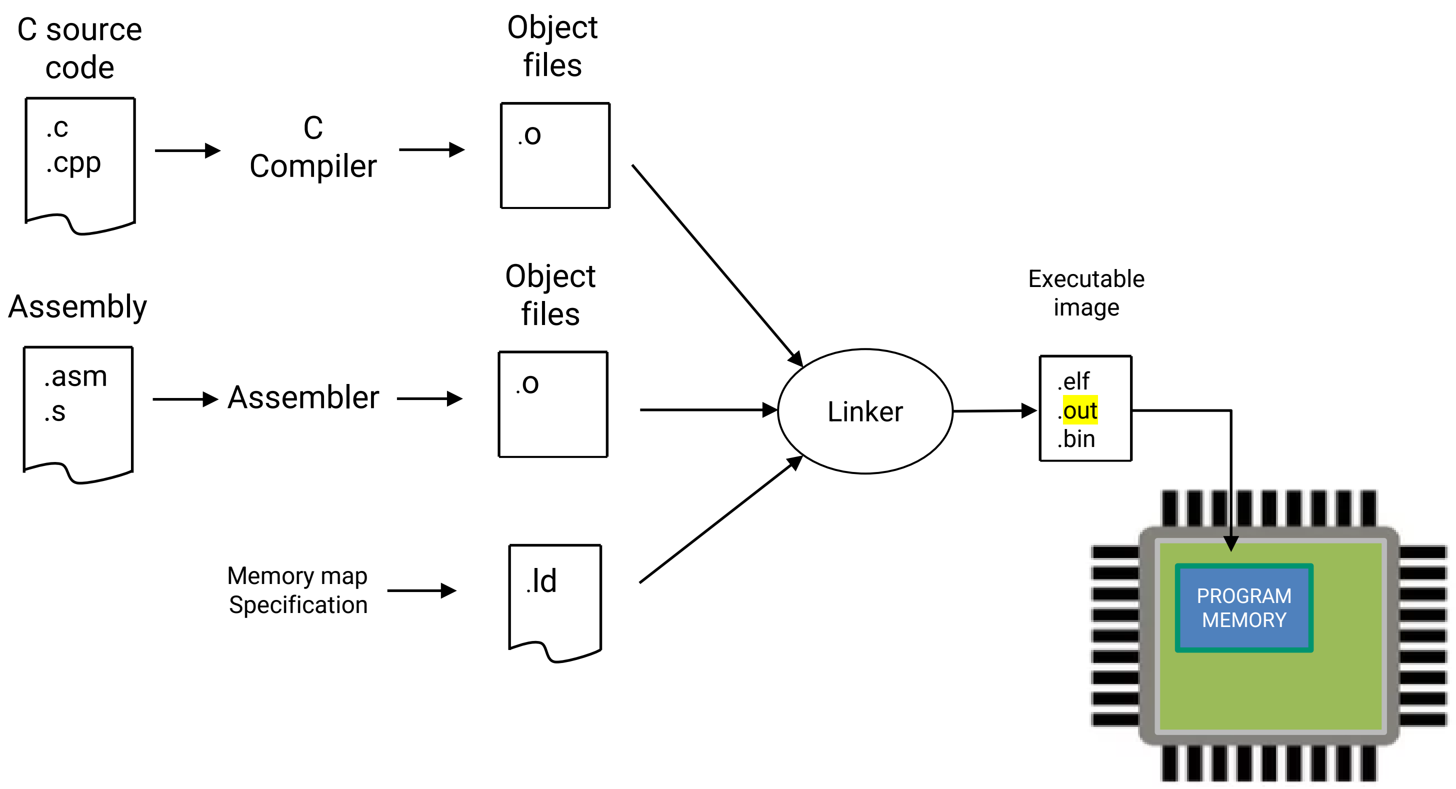


Edit the C or C++ by using a text editor and save the file with ".c" or ".cpp" file extension

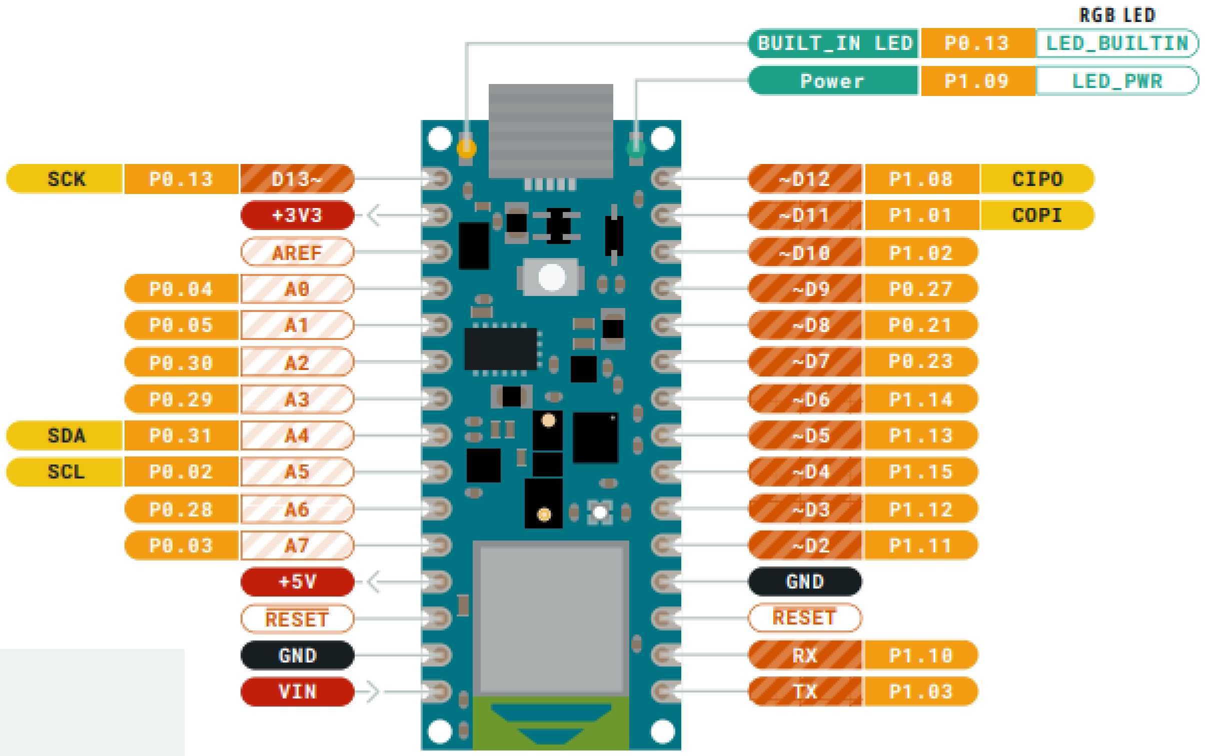
When using the C tools to create an executable file, there are actually two main steps involved : the first is called compiling and the second is called linking.

Most IDE take care of running the compiler and linker at the correct time, in just one step

Programming compiler and linker

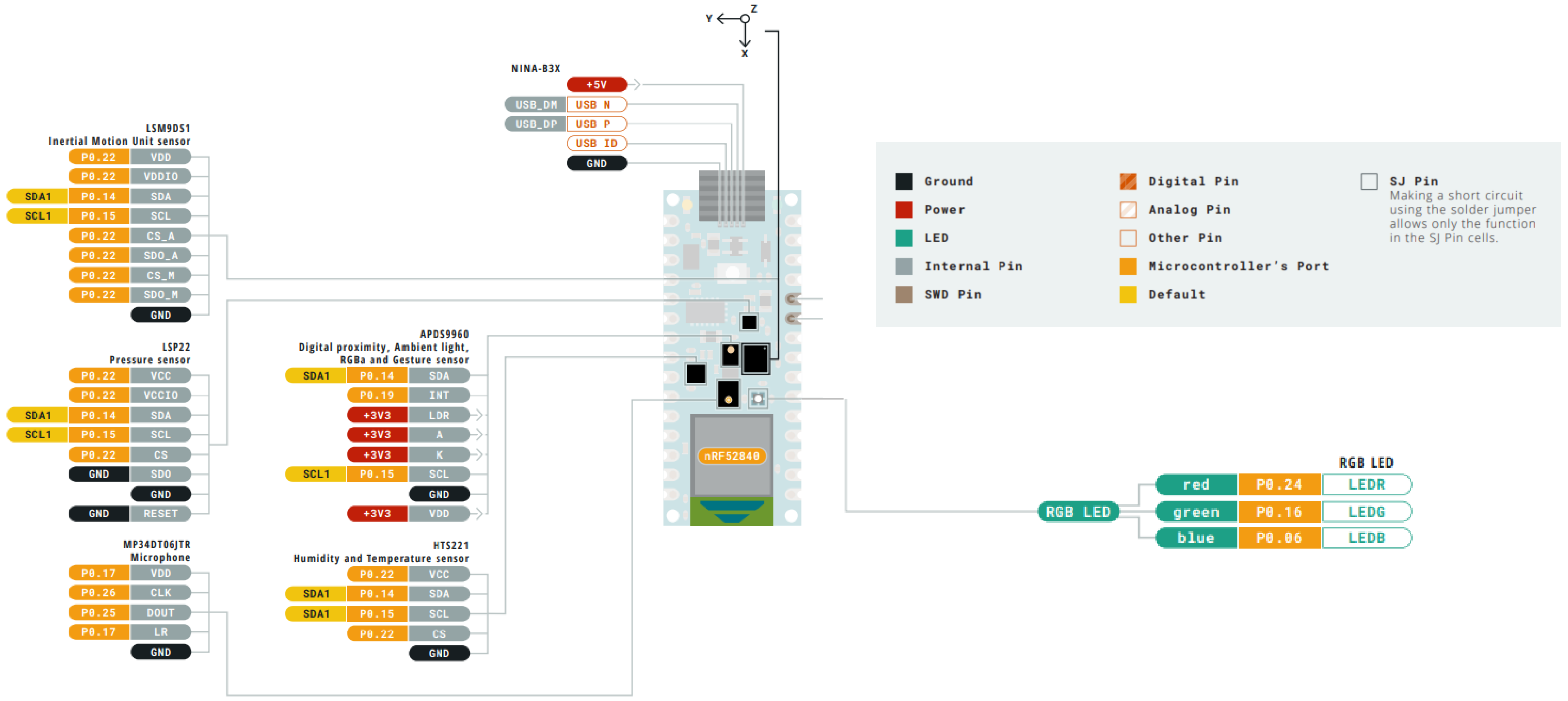


The Arduino Nano 33 BLE sense



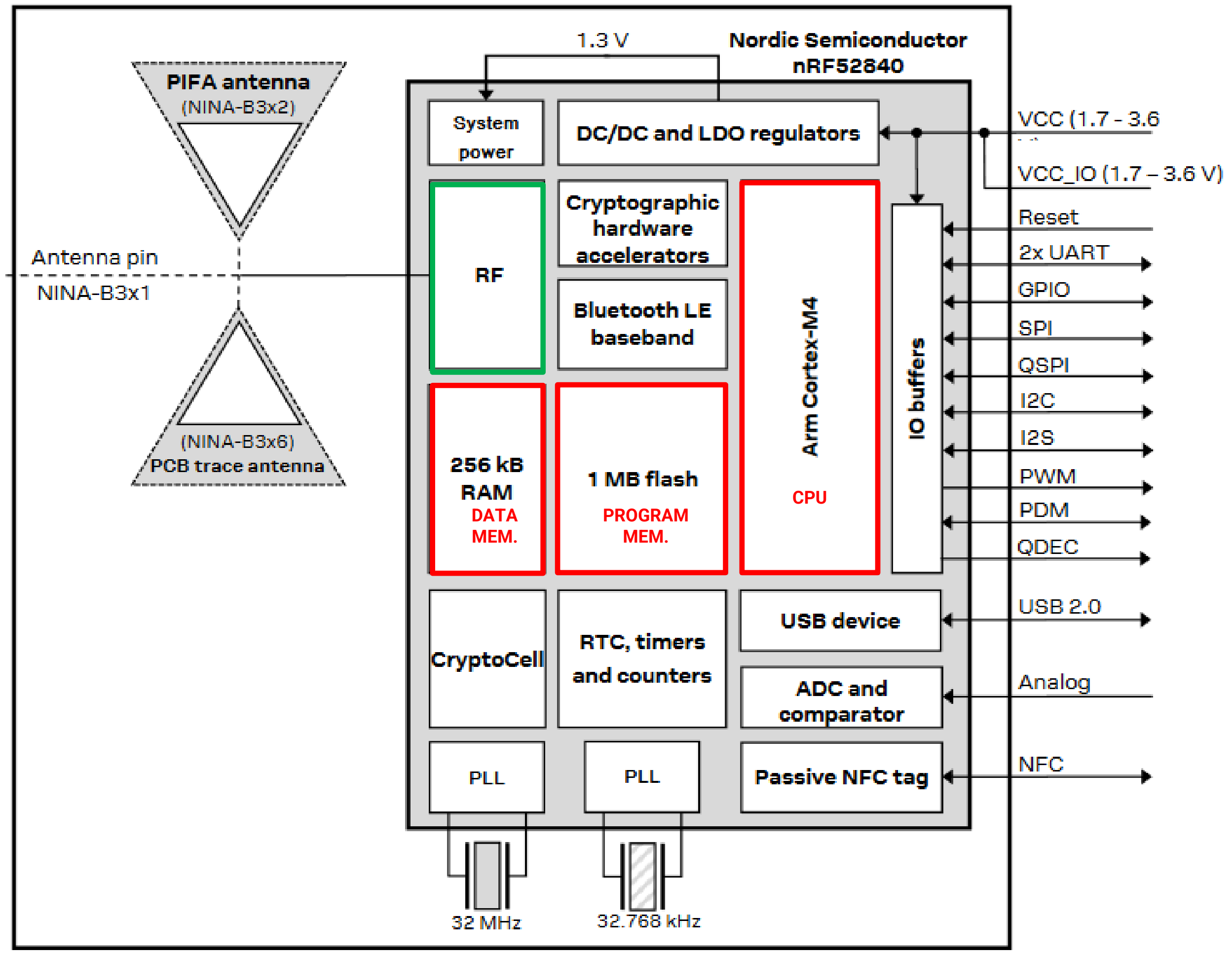
⬛	Ground	▨	Digital Pin
🟥	Power	▨	Analog Pin
🟩	LED	⬜	Other Pin
⬜	Internal Pin	🟧	Microcontroller's Port
🟫	SWD Pin	🟨	Default

Sensors, Actuators, Microcontroller and communication scheme



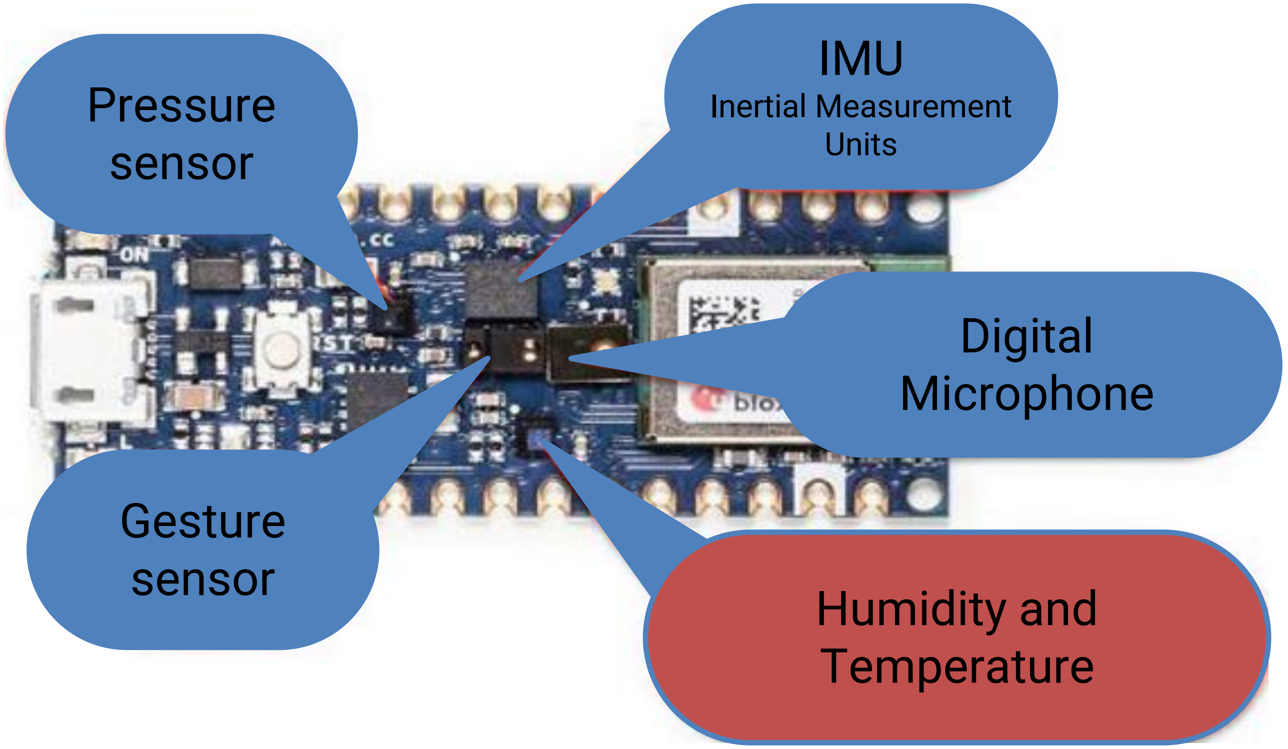
Nordic nRF52840 block diagram

- CPU: ARM Cortex M4
- 64 MHz
- 1 MB flash
- 256 KB RAM
- Bluetooth v5.0



Sensors on the board

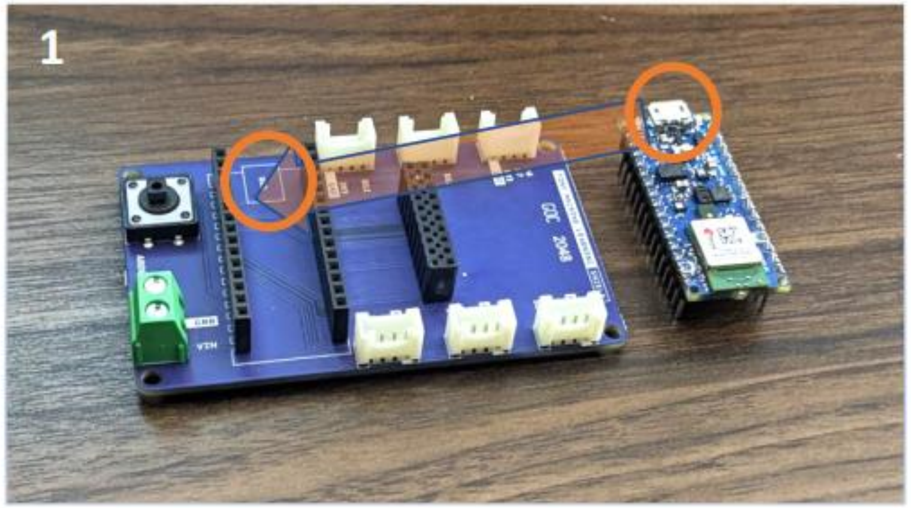
- The Arduino Nano 33 BLE Sense Board has some internal sensors such as IMU, a pressure sensor, a digital microphone, a humidity and temperature sensor, and a gesture sensor
- Most sensors chip are attached on I2C bus

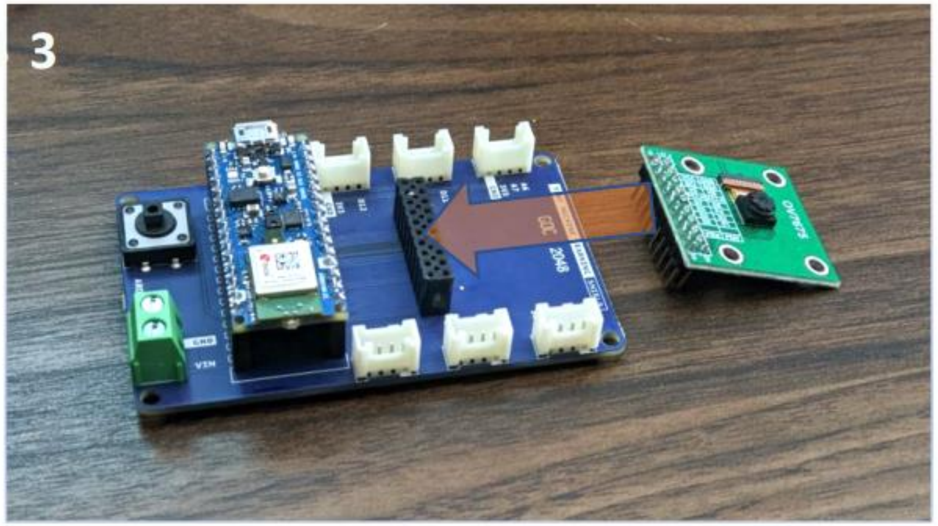


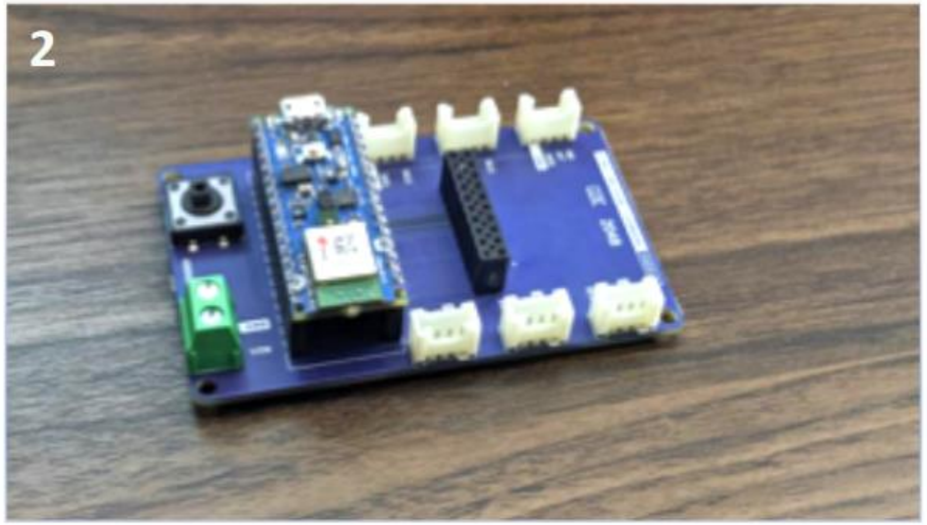
TinyML kit

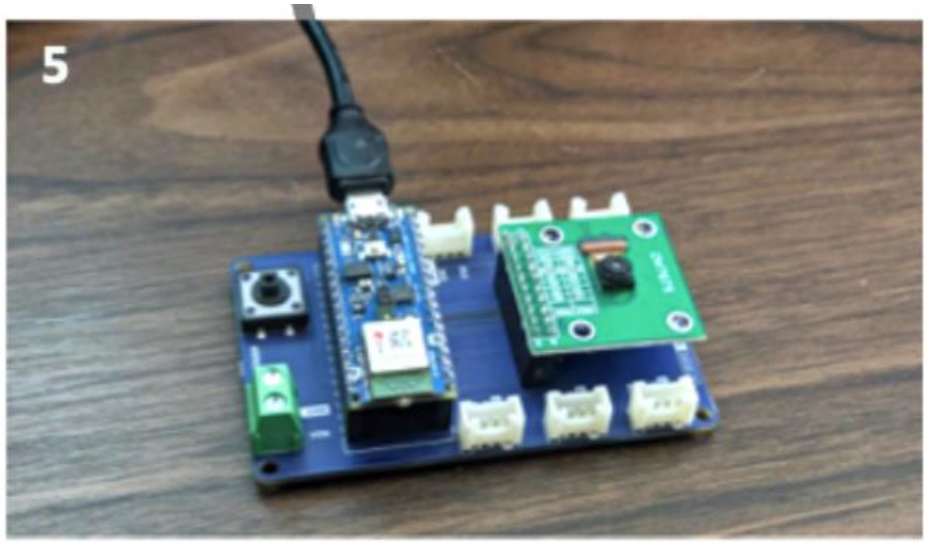
ARDUINO
TINY MACHINE
LEARNING KIT
in partnership with
edX
TensorFlow

Building up the kit









The arduino Environment

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

- Setup() is a function meant to be run only once, at the beginning

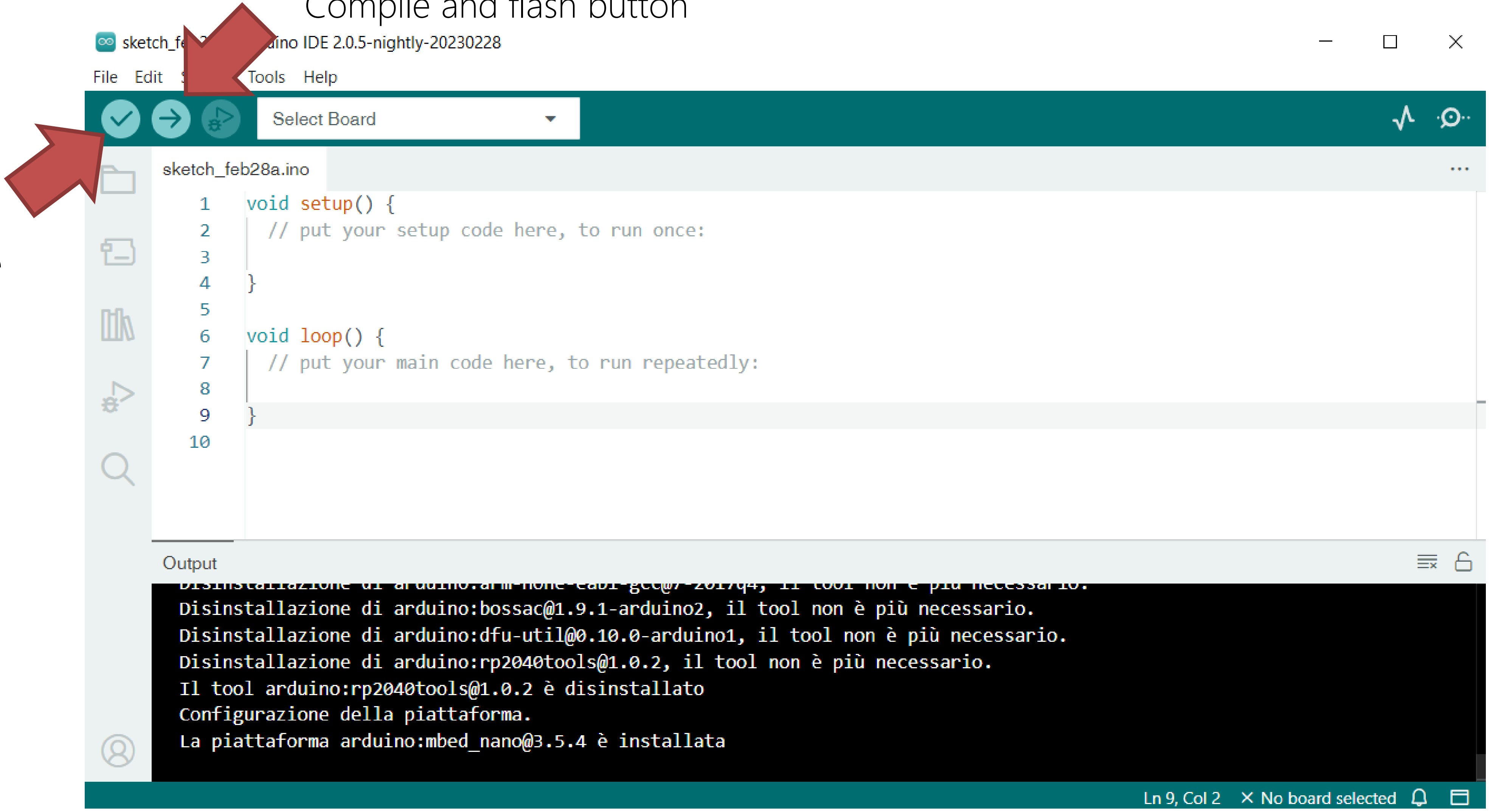
```
void loop() {
  // put your main code here, to run repeatedly:

}
```

- Loop() is a function that is executed continously and indefinetely by the board. Similar to a while(True) loop
- They are put inside of the «main» c file that constitute the «starting point» of the firmware

Compile and flash the firmware

Compile and flash button

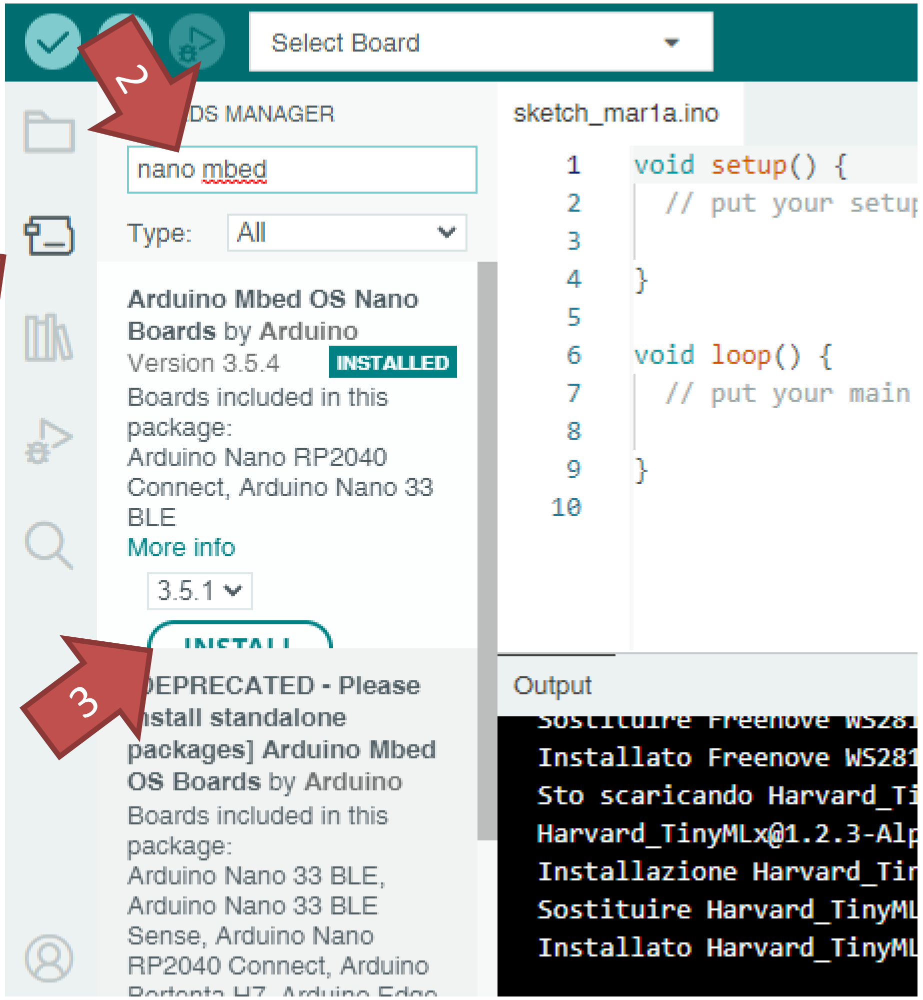
Compile button


sketch_feb28a | Arduino IDE 2.0.5-nightly-20230228
File Edit Sketch Tools Help
Select Board
sketch_feb28a.ino
Output
Ln 9, Col 2 ✕ No board selected

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

}

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

}

```

```
Disinstallazione di arduino:arm-none-eabi-gcc@7-2017q4, il tool non è più necessario.
Disinstallazione di arduino:bossac@1.9.1-arduino2, il tool non è più necessario.
Disinstallazione di arduino:dfu-util@0.10.0-arduino1, il tool non è più necessario.
Disinstallazione di arduino:rp2040tools@1.0.2, il tool non è più necessario.
Il tool arduino:rp2040tools@1.0.2 è disinstallato
Configurazione della piattaforma.
La piattaforma arduino:mbed_nano@3.5.4 è installata
```

Add the board from the board manager and enable it in tools



nano mbed

Type: All

Arduino Mbed OS Nano Boards by **Arduino**
Version 3.5.4 INSTALLED
Boards included in this package:
Arduino Nano RP2040 Connect, Arduino Nano 33 BLE
More info

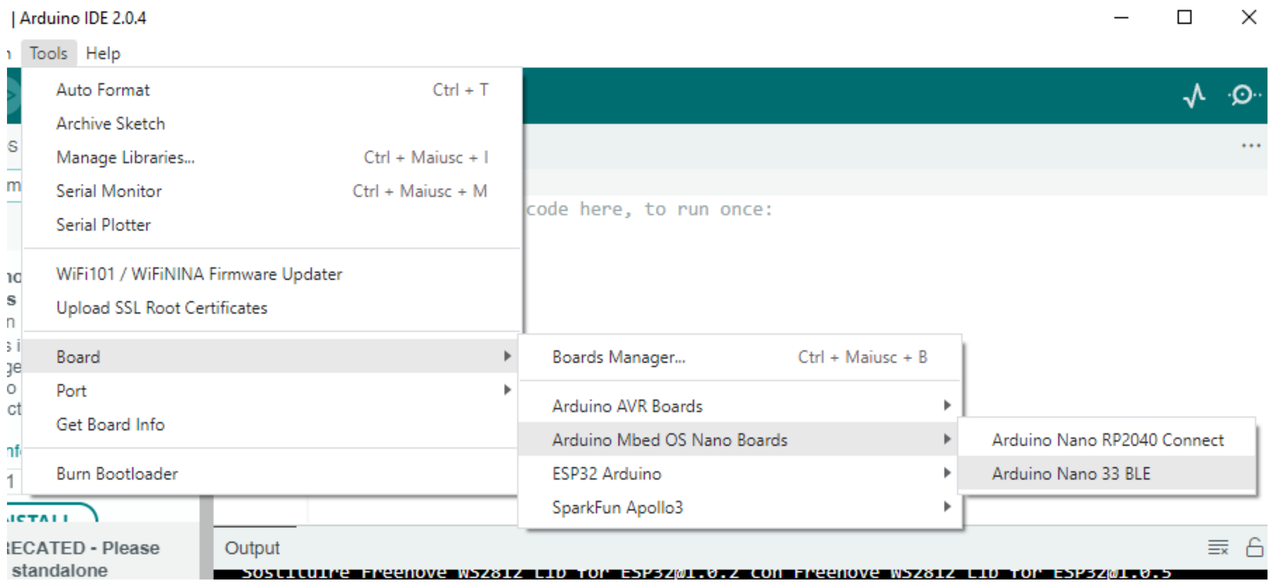
3.5.1

INSTALL

EPRECATED - Please nstall standalone packages] Arduino Mbed OS Boards by **Arduino**
Boards included in this package:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect, Arduino Portenta H7, Arduino Edge

```
void setup() {
  // put your setup
}

void loop() {
  // put your main
}
```

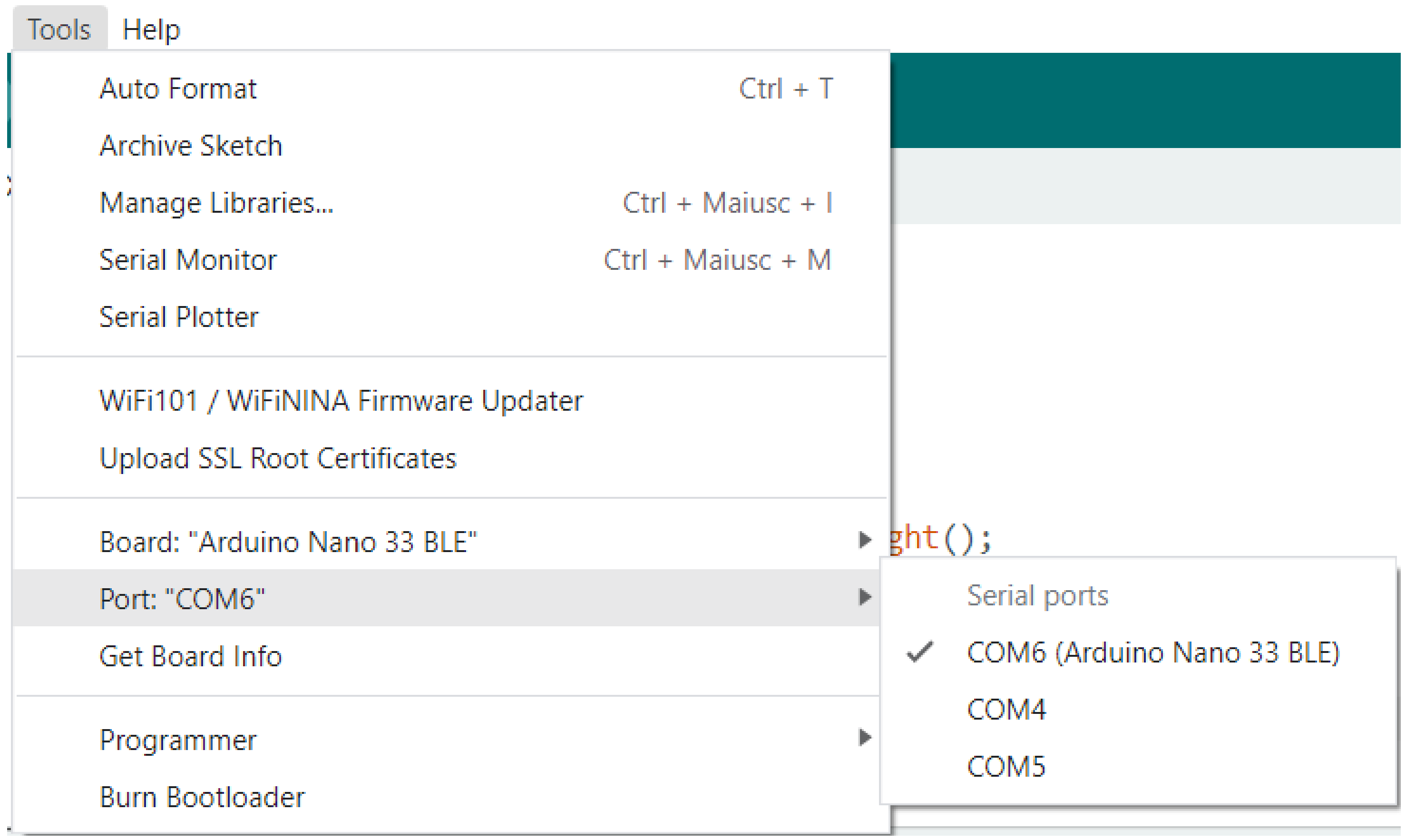
```
Sostituire Freenove WS281
Installato Freenove WS281
Sto scaricando Harvard_Ti
Harvard_TinyMLx@1.2.3-Alp
Installazione Harvard_Tir
Sostituire Harvard_TinyML
Installato Harvard_TinyML
```


| Arduino IDE 2.0.4
Tools
Help
Auto Format Ctrl + T
Archive Sketch
Manage Libraries... Ctrl + Maiusc + I
Serial Monitor Ctrl + Maiusc + M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board
Port
Get Board Info
Burn Bootloader
Boards Manager... Ctrl + Maiusc + B
Arduino AVR Boards
Arduino Mbed OS Nano Boards
ESP32 Arduino
SparkFun Apollo3
Arduino Nano RP2040 Connect
Arduino Nano 33 BLE
ECATED - Please standalone
Output

```
code here, to run once:
```

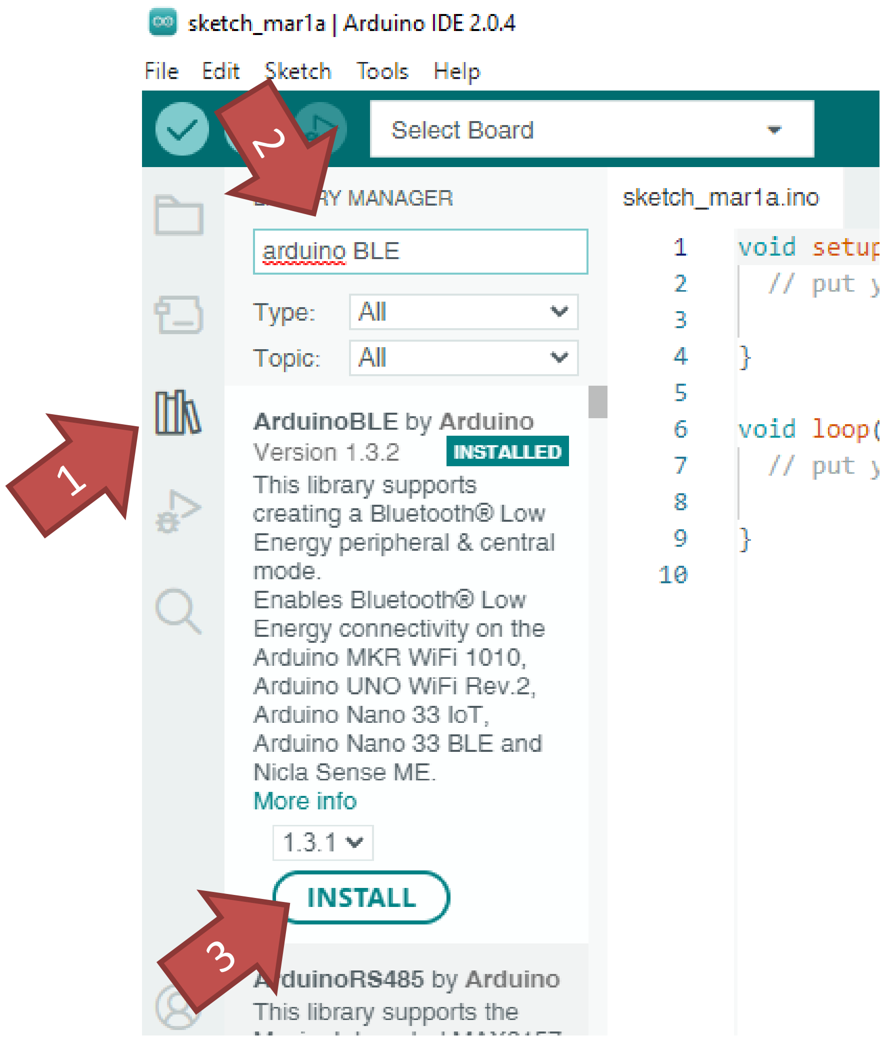
*It can happen that the process of selecting the board from tools needs to be repeated sometime.

Select the port


Tools
Help
Auto Format Ctrl + T
Archive Sketch
Manage Libraries... Ctrl + Maiusc + I
Serial Monitor Ctrl + Maiusc + M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "Arduino Nano 33 BLE"
Port: "COM6"
Get Board Info
Programmer
Burn Bootloader
Serial ports
✓ COM6 (Arduino Nano 33 BLE)
COM4
COM5

*It can happen that the process of selecting the port needs to be repeated sometime.

Install a new library



arduino BLE

Type: All

Topic: All

ArduinoBLE by **Arduino**
Version 1.3.2 INSTALLED
This library supports creating a Bluetooth® Low Energy peripheral & central mode.
Enables Bluetooth® Low Energy connectivity on the Arduino MKR WiFi 1010, Arduino UNO WiFi Rev.2, Arduino Nano 33 IoT, Arduino Nano 33 BLE and Nicla Sense ME.
More info

1.3.1

INSTALL

ArduinoRS485 by **Arduino**
This library supports the

```
void setup
  // put y

}

void loop(
  // put y

}
```

Then add it directly from the code with `include` …

Examples

- Example 0 – Blink
- Example 1 – UART communication
- Example 2 – Humidity and Temperature Test
- Example 3 – IMU, accelerometer, gyroscope, magnetometer Tests
- Example 4 – Microphone Test
- Example 5 – Camera Test

POLITECNICO
MILANO 1863

Appendix

Links

https://colab.research.google.com/drive/1oX2STKatr6y0jtgDipQmjh1j932z4fuq?usp=sharing

https://tinyml.seas.harvard.edu/assets/other/4D/22.03.11_Marcelo_Rovai_Handout.pdf

Libraries needed for the example:

- Built-in examples
- HTS221 (Arduino_HTS221)
- TinyMLx (Harvard_TinyMLx)

References

- "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers", Daniel Situnayake, Pete Warden, O'Reilly Media, Inc.
- Online course:
  - https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning
- Tutorials with the board:
  - https://docs.arduino.cc/hardware/nano-33-ble-sense
- Deeper materials on the Hardware:
  - Course slide for "Embedded and Edge AI" Phd course @ Politecnico di Milano, material available on webeep of this course
- Example of python and c programming crash course (I take no responsibility on the completeness and correctness of this material, find what's best for you):
  - Python: https://www.youtube.com/watch?v=JJmcL1N2KQs&t=634s
  - C/C++: https://www.youtube.com/watch?v=1v_4dL8l8pQ
- A lot more material on TinyML:
  - http://tinyml.seas.harvard.edu/

LSM9DS1 (9 axis IMU) description

Accelerometer:

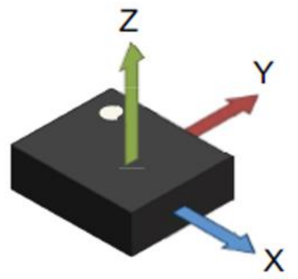
- used to measure the acceleration in m/s2 (meters per second per second) or g's (gravities [about 9.8 m/s2])
- its scale can be set to either ± 2, ± 4, ± 8, or ± 16 g

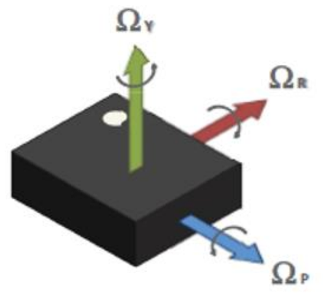
Gyroscope:

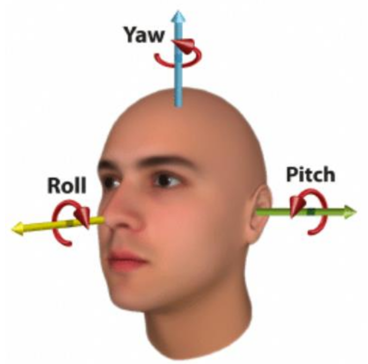
- used to measure the angular velocity in degrees per second (usually abbreviated to DPS or °/s).
- its scale can be set to either to ± 245 ± 500 or ± 2000 DPS

Magnetometer:

- measures the power and direction of magnetic fields in units of gauss (Gs)
- its measurement scale to either ± 4, ± 8, ± 12, or ± 16 Gs.




Ω_Y
Ω_R
Ω_P


Yaw
Roll
Pitch