

The Arduino life

Workshop on Arduino Embedded Development Board

Egin beharrekoak

Arduino mikrokontrolagailuaren programazioa.

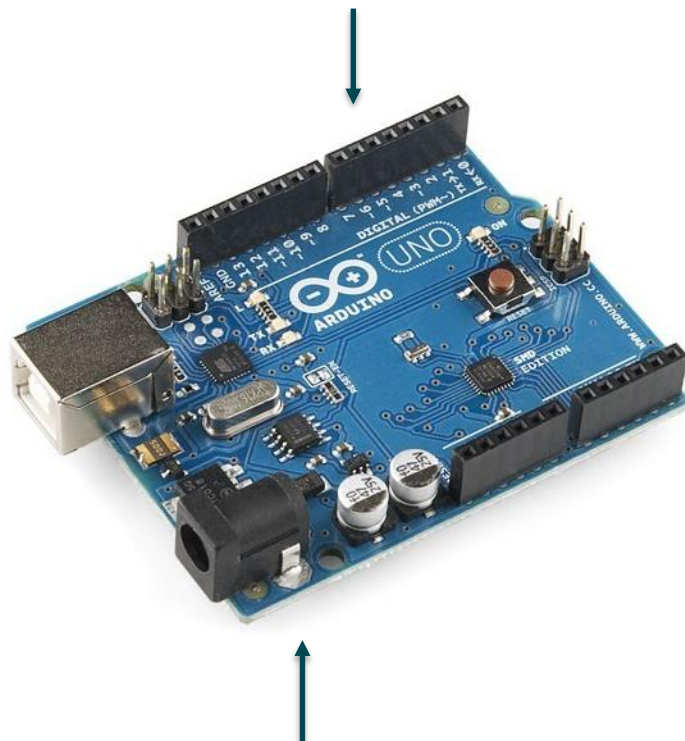
- Honekin jauziak detektatzen dituen azelerometroa kontrolatuko da.
- GPS batetik irakurriko da pertsonaren posizioa.
- Jauzi bat dagoenean abisua eman beharko du:
 1. Led baten bitartez (Arduinoren Led-a piztu behar da).
 2. Dei baten bitartez.
 3. SMS baten bitartez (posizioarekin batera).

Egin beharrekoak



Egin beharrekoak

Azelerometroa



GPS

Led-a piztu



Introduction

- Arduino Basics
- Arduino Architecture
- Arduino board layout.
- Arduino IDE
- General purpose functions
- Serial Port

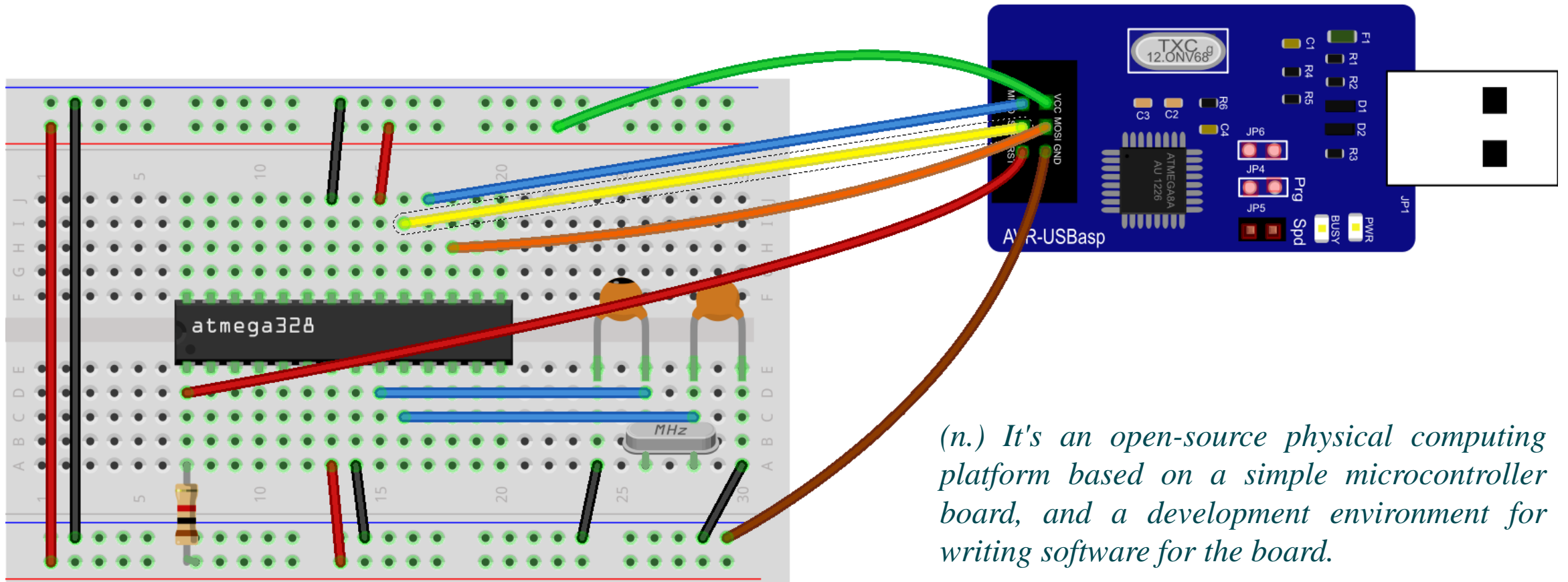
Learn by Doing – Sense, Control & Actuate

- Project 1: LED Blinking
- Project 2: Push Button
- Project 3: Serial Port

Additional information

- HC-05 Bluetooth module
- MMA8451 Accelerometer
- Making emergency calls
- NEO 6M GPS
- Send SMS with arduino

Arduino..?

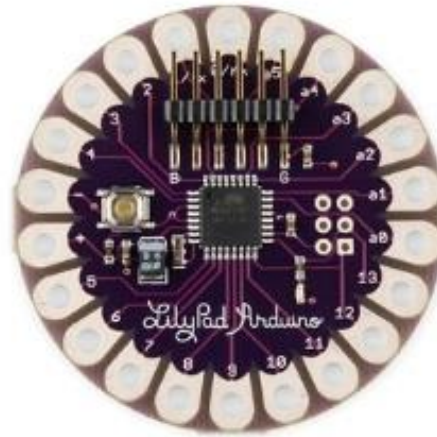


(n.) It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

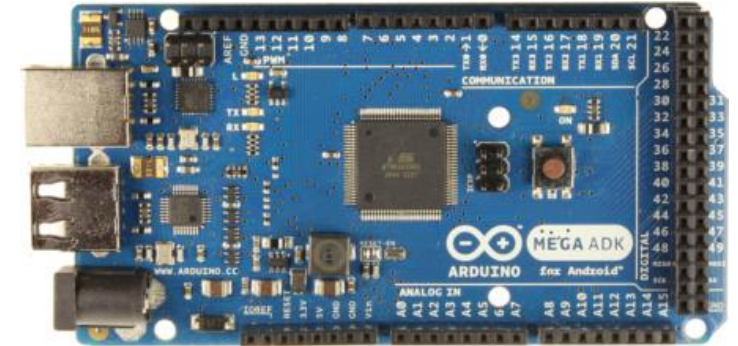
Types of Arduino boards



Arduino Nano



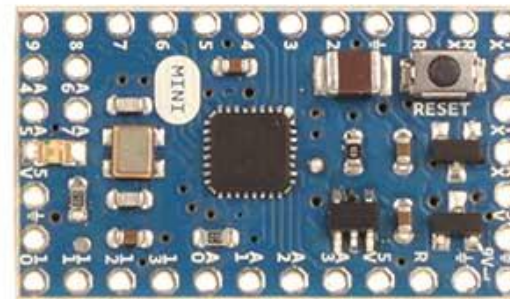
Arduino LilyPad



Arduino Mega



Arduino Uno



Arduino Mini



Arduino Leonardo

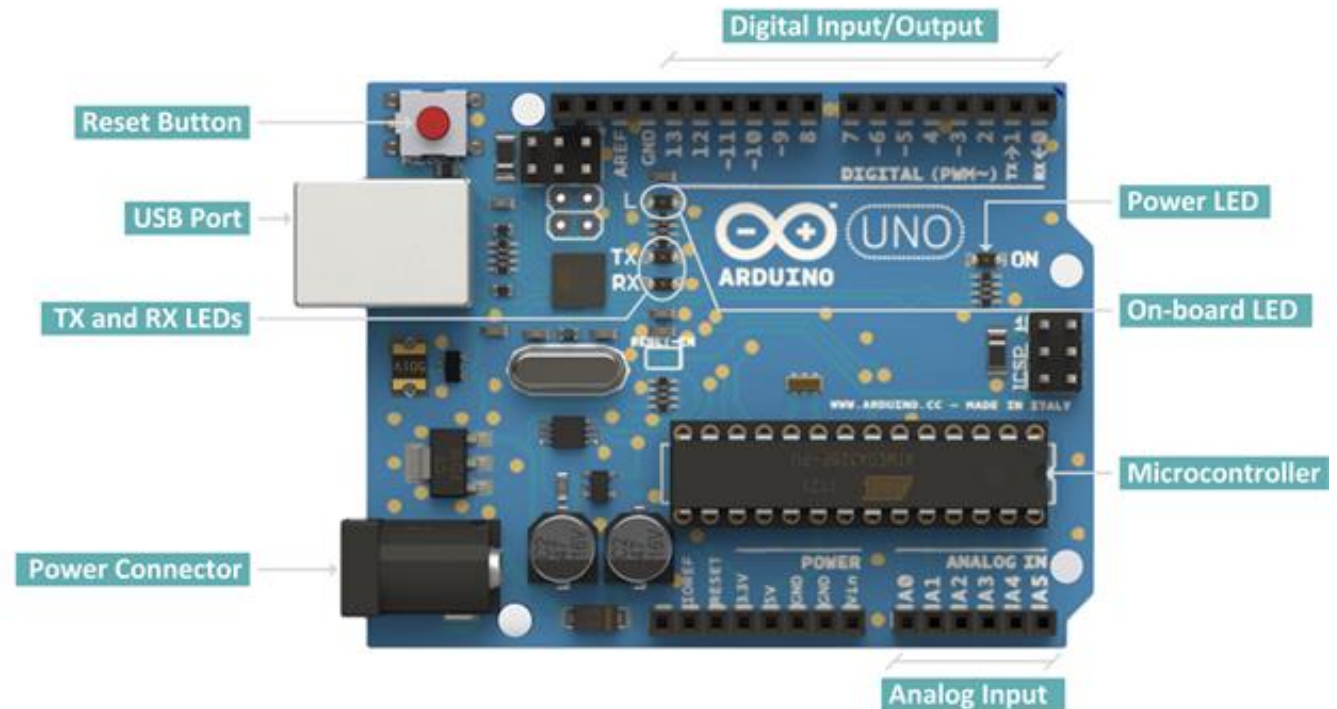
Arduino Uno pins

Digital pins:

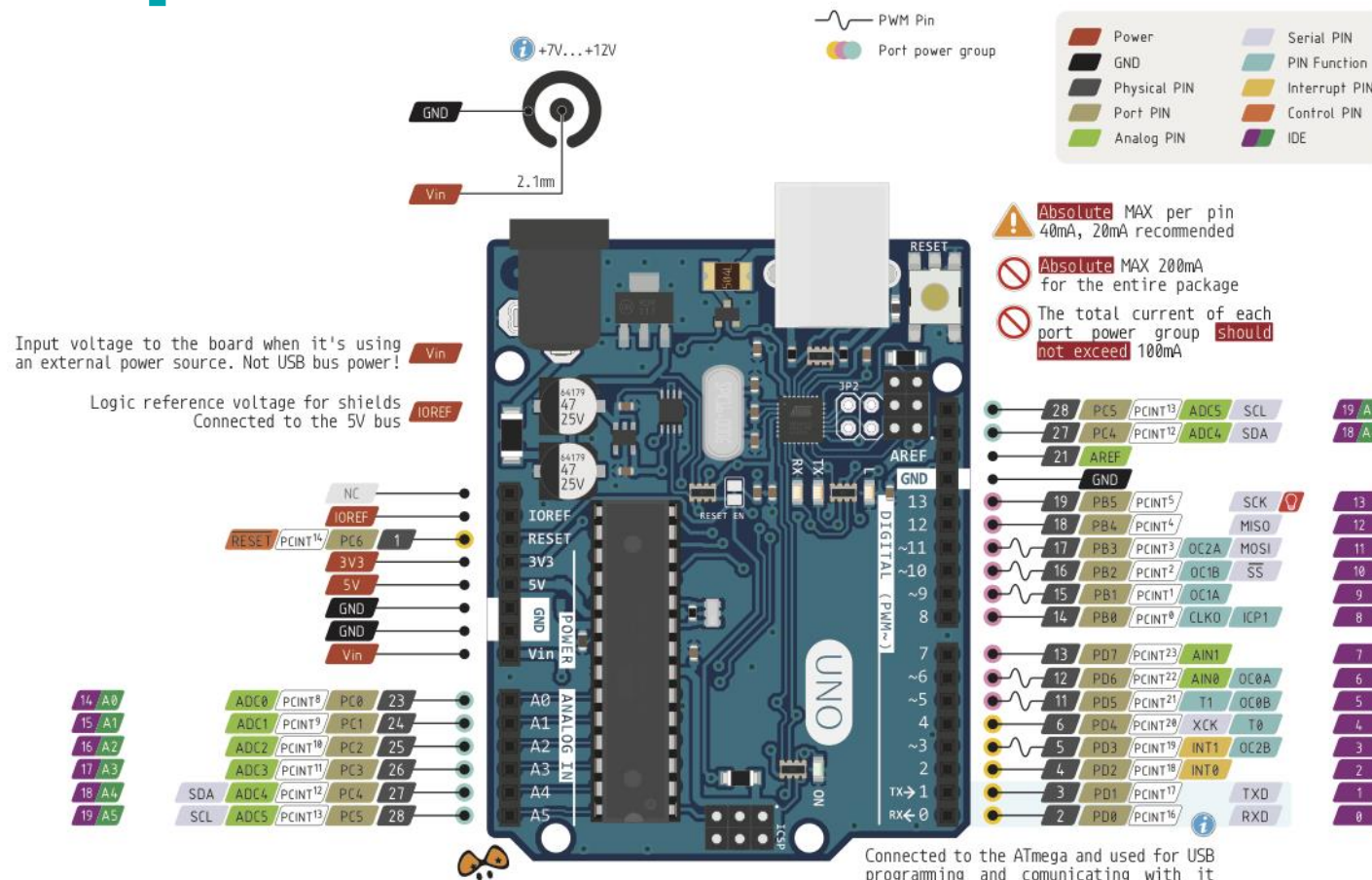
- 14 digital IO pins
- 6 are PWM pins (3, 5, 6, 9, 10, and 11).

Analog pins:

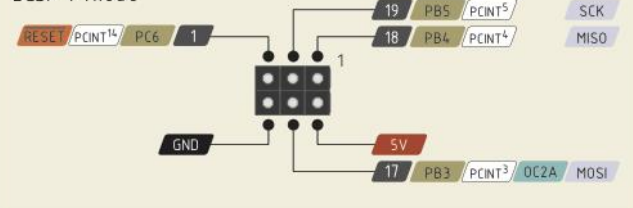
- 6 analog pins(A0, A1, A2, A3, A4, and A5)
- Takes analog values as an input



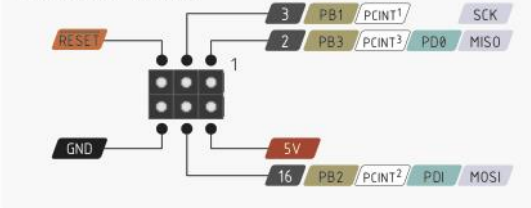
Arduino Uno pinouts



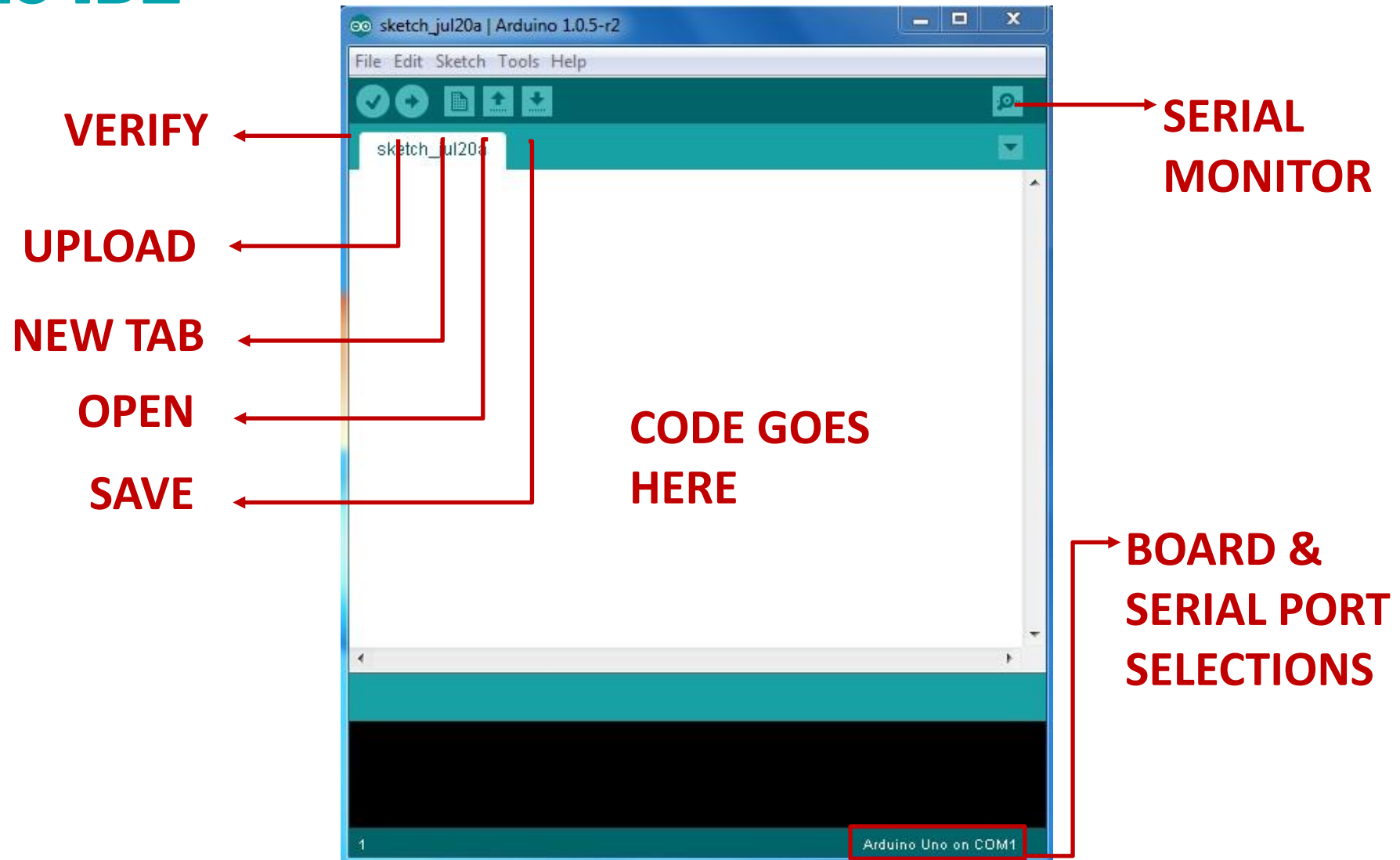
ICSP Pinout



16U2 ICSP Pinout

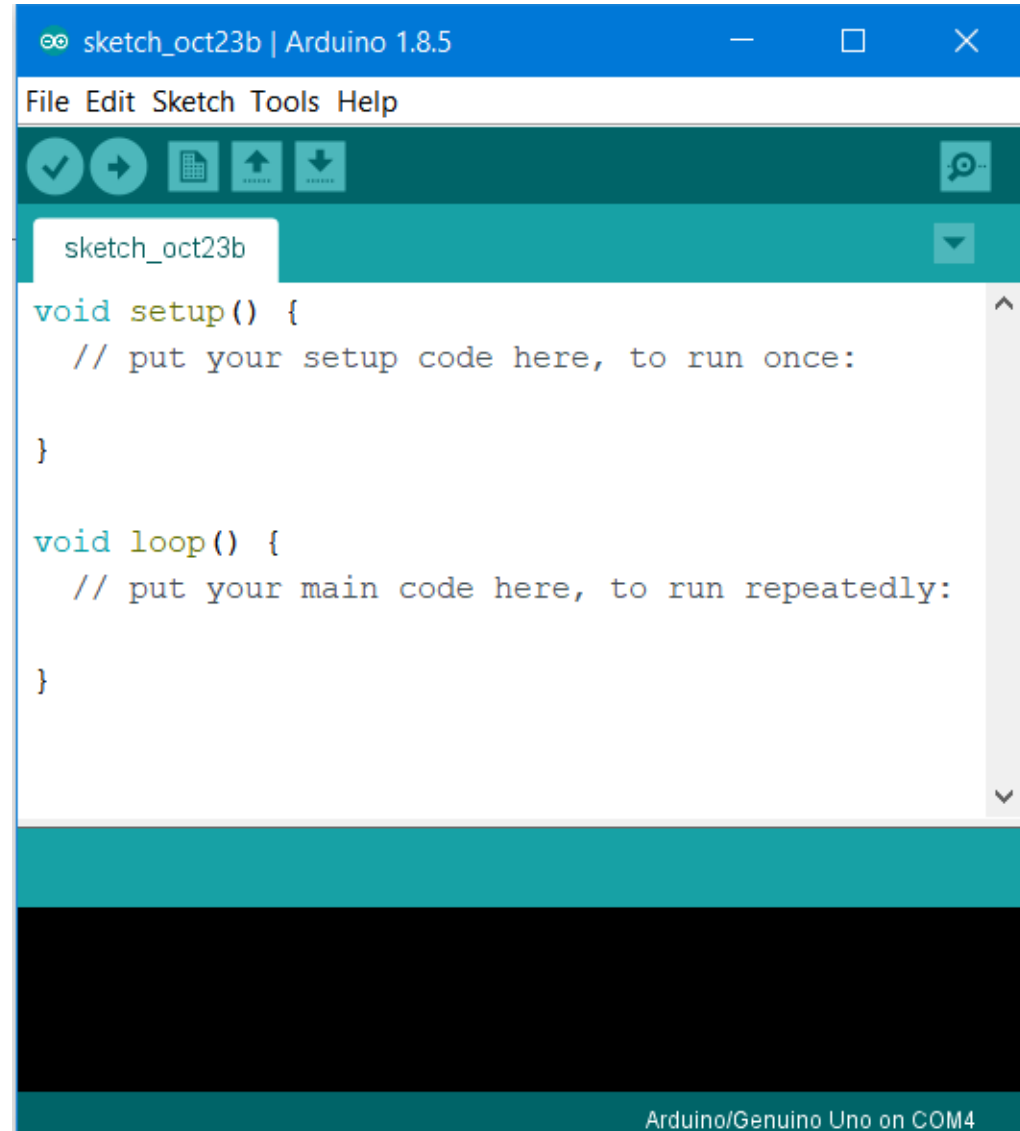
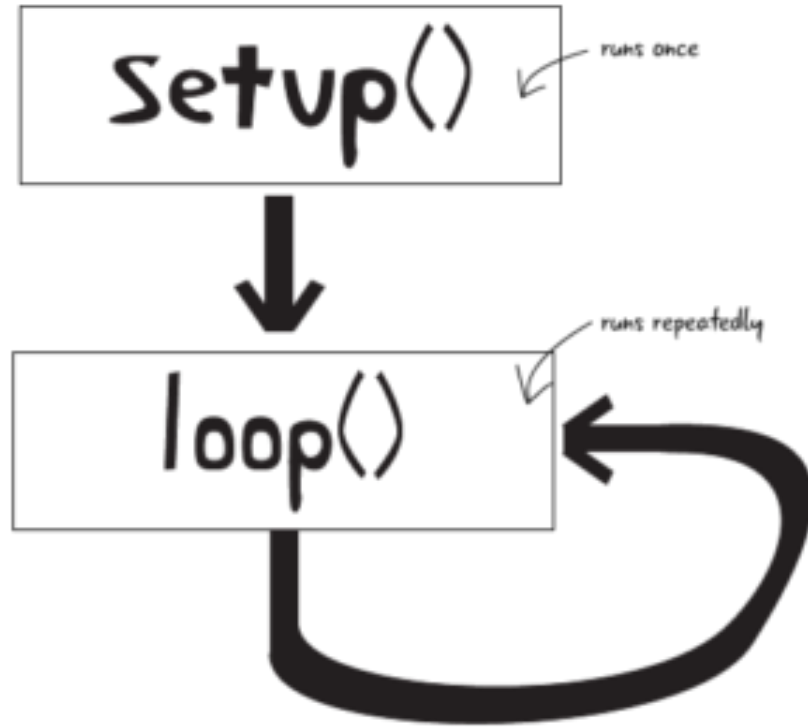


Arduino IDE



IDE Download: <https://www.arduino.cc/en/software>

Basic Coding structure



```
sketch_oct23b | Arduino 1.8.5
File Edit Sketch Tools Help
[Icons: Checkmark, Run, Upload, Download, Serial Monitor]
sketch_oct23b
void setup() {
  // put your setup code here, to run once:
}

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

The screenshot shows the Arduino IDE interface. The title bar indicates the file is "sketch_oct23b" and the version is "Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, running, uploading, downloading, and opening the serial monitor. The main text area shows the basic coding structure: `void setup() {` followed by a comment `// put your setup code here, to run once:` and a closing brace `}`. Below this is `void loop() {` followed by a comment `// put your main code here, to run repeatedly:` and a closing brace `}`. The status bar at the bottom indicates the board is "Arduino/Genuino Uno on COM4".

Basic Coding structure

■ *setup()* function

- Called when a sketch starts.
- To initialize variables, pin modes, start using libraries, etc.
- Will only run once, after each power-up or reset of the Arduino board.

■ *loop()* function

- Loops consecutively.
- Code in the loop() section of the sketch is used to actively control the Arduino board.

■ *Commenting*

- Any line that starts with two slashes (//) will not be read by the compiler, so you can write anything you want after it.

General purpose functions

■ *pinMode()*

- Instruction used to set the mode (INPUT or OUTPUT) in which we are going to use a pin.
- Eg: `pinMode (13, OUTPUT);`
- ie. setting pin13 as output.

■ *digitalWrite()*

- Write a HIGH or a LOW value to a digital pin.
- Eg: `digitalWrite (11, HIGH);`
- ie. setting pin 11 to high.

General purpose functions

■ *digitalRead()*

- Reads the value from a specified digital pin, either HIGH or LOW
- Eg: `int inPin=7;`
`val = digitalRead(inPin);`
- ie. reads the value from inPin and assigns it to val.

■ *delay()*

- Pauses the program for the amount of time (in milliseconds) specified as parameter.
- Eg: `delay(1000);`
- ie. waits for a second (1000 ms = 1 s)

Serial Port

Arduinos use serial ports for communicating with computers and other devices. The USB port of an Arduino is used for serial communication with a computer, with the added advantage that USB can also be used to power the device. USB also has the advantage of auto-configuring most of the parameters.

Some Arduinos have other hardware serial ports, enabling communication with other devices. The USB communication is sent to Arduino pins 0 and 1, meaning that those pins are reserved if your device must communicate with a computer.

Serial Port

To **Start a Serial Connection** you must first do some basic configuration. To do this, you use the **begin** function of the Serial object. Typically, 9,600 is an appropriate speed for communicating. You are free to use any speed you want as long as both devices are operating at the same speed.

```
void setup()
{
  Serial.begin(9600); // Opens the serial port
}
```

Serial configuration is normally done in setup() because devices tend to not change the speed at which they communicate over time.

Serial Port

Sending Text

To send data to the serial device , use the function `print()`. The data to be printed can be in any format.

```
Serial.print("Hello, world"); // Output an entire string  
Serial.print('!'); // Output a single character
```

You need the following components:

- Arduino Uno
- USB cable

Serial Port Writing example

```
void setup( )  
{  
  pinMode(13, OUTPUT);  
  Serial.begin(9600); // sets data rate to 9600 bps  
}  
void loop()  
{  
  digitalWrite(13, HIGH);  
  Serial.println("LED is On");  
  delay(1000);  
  digitalWrite(13, LOW);  
  Serial.println("LED is Off");  
  delay(1000);  
}
```

Serial Port Reading

Arduinos can also receive data. Receiving data can be used for many projects; computers can send data, for example, to control the brightness of an LED. To read data from the serial device , use the function **Serial.read();**

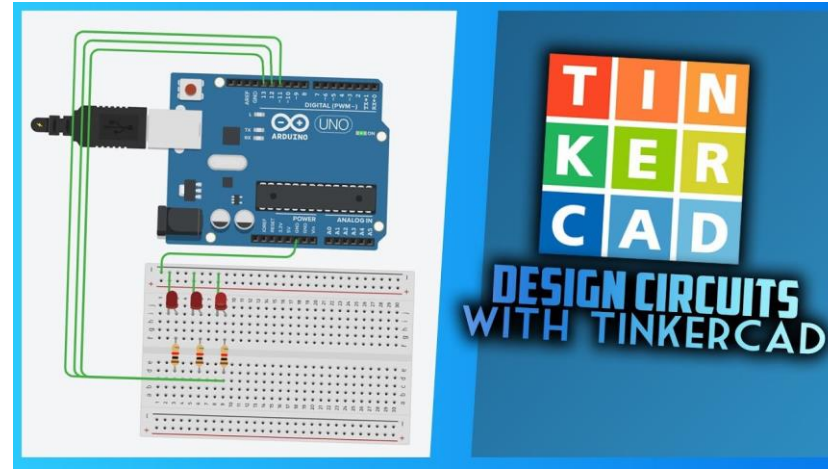
```
int ledPin=13;  
int value;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledPin,OUTPUT);  
}
```

Serial Port Reading example

```
void loop ()  
{value = Serial.read();  
  if (value == '1')  
  {  
    digitalWrite(ledPin,HIGH);  
  }  
  else if (value == '0')  
  {  
    digitalWrite(ledPin,LOW);  
  }  
}
```

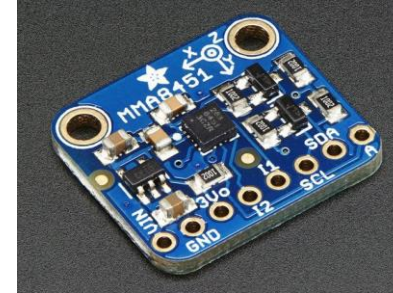
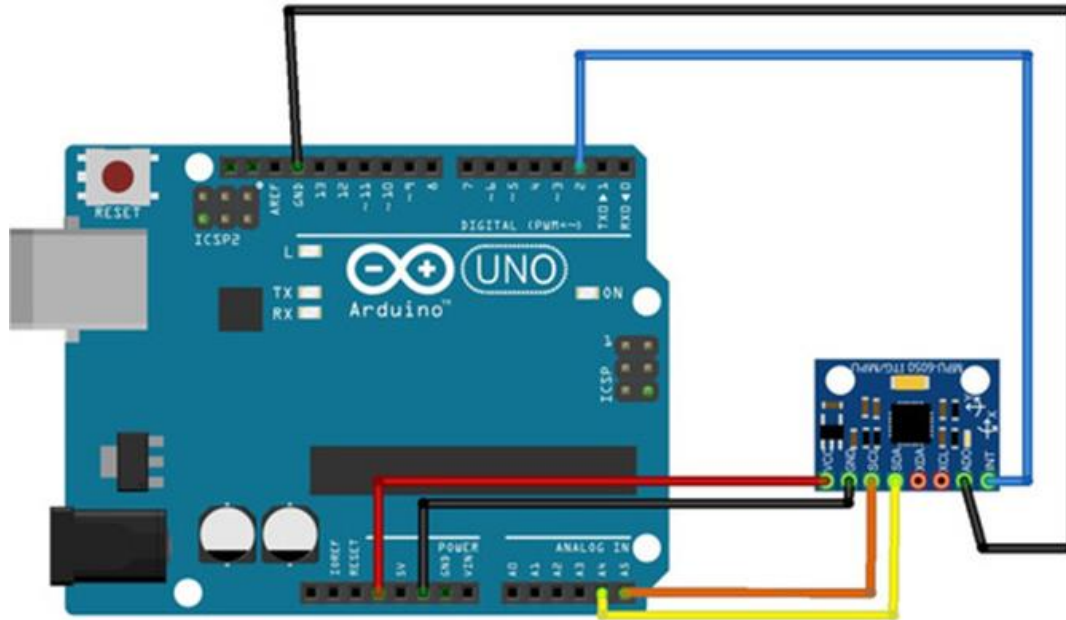

Arduino program examples

WebPage: www.tinkercad.com



- Project 1: LED Blinking (<https://www.tinkercad.com/things/bDHquWI5k2q-lesson-1-blinking-led>)
- Project 2: Push Button (<https://www.instructables.com/Digital-Input-With-a-Pushbutton-With-Arduino-in-Ti/>) Codea Step 4 atalean dago.

MMA8451 Accelerometer

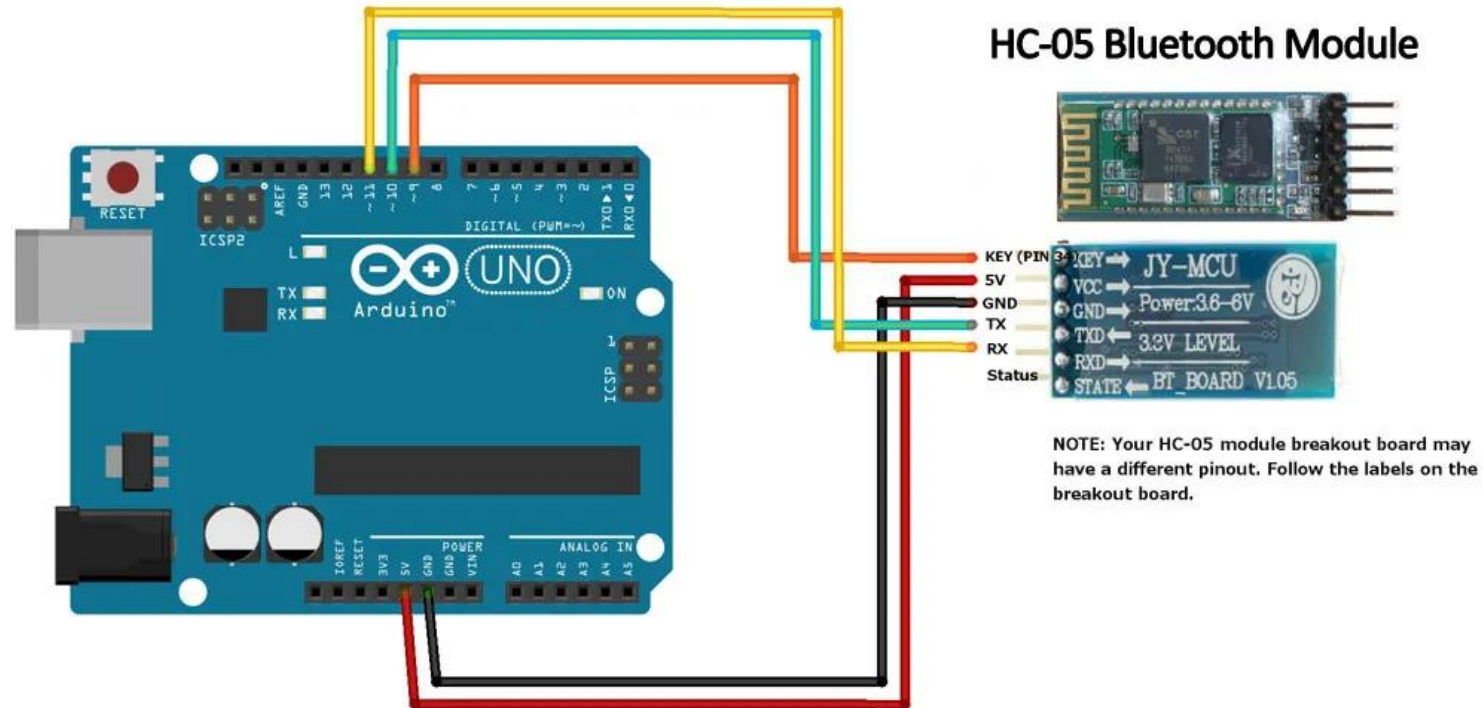


Proposed website:

<https://learn.adafruit.com/adafruit-mma8451-accelerometer-breakout/overview>

(Azelerometroa Arduinora nola konektatzen den azaltzen du. Baita Azelerometrotik dautak nola irakurtzen dira azaltzen da)

HC-05 Bluetooth module



Proposed website:

https://naylampmechatronics.com/blog/24_configuracion-del-modulo-bluetooth-hc-05-usando-comandos-at.html

(Bluetooth modulua Arduinora nola konektatzen den azaltzen du. Bluetooth nola irakurri eta bildali azaltzen da)

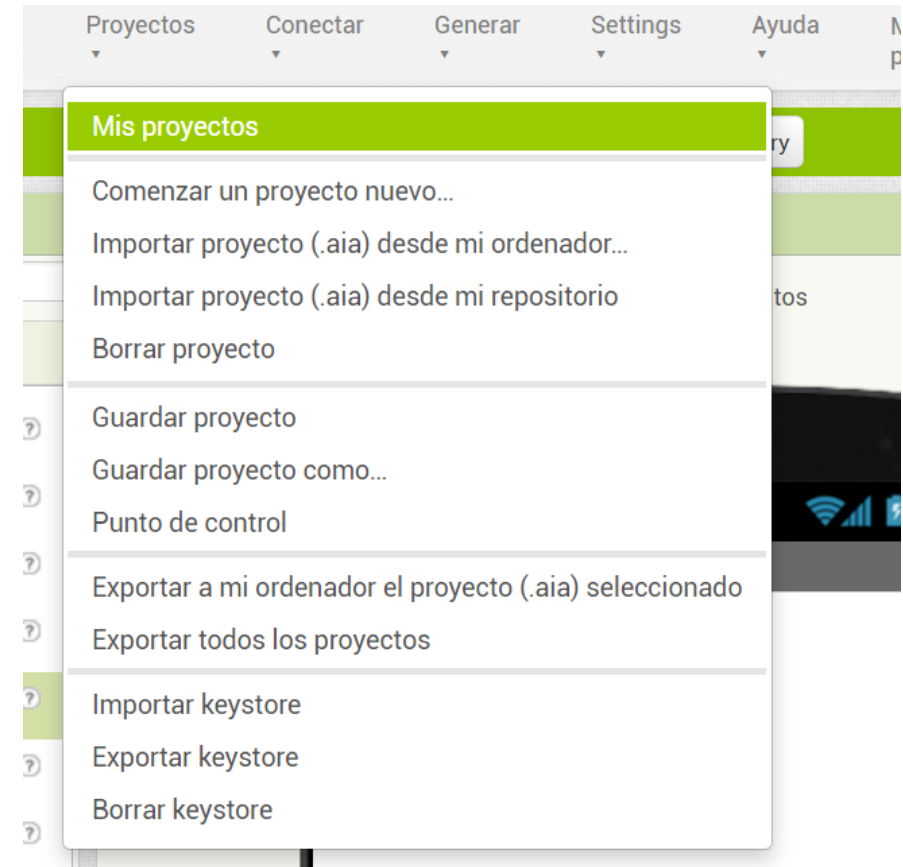
MIT App Inventor

- *Android aplikazioak garatzeko era grafiko batean.*
- 1. *Android emuladorea instalatu (telefono bat simulatzeko):*
<https://appinventor.mit.edu/explore/ai2/windows>
- 2. *Aplikazioak garatu: <https://appinventor.mit.edu/> -> Create Apps!*



MIT App Inventor - Adibidea

- *Adibide bat duzue Mudelen PBL atalean: botoi baten click kopurua kontatzen duen aplikazioa.*
- *Proiektuaren fitxategiak .aia formatua du eta importatu ahal da.*
- *Exekutatzeko emuladorean:
Conectar -> Emulador*



MIT App Inventor - Adibidea

- *Egiten dituzen aplikazioak mobilean exekutatzeke:*

<https://appinventor.mit.edu/explore/ai2/setup-device-wifi.html>

Making emergency calls



Proposed videos:

<https://www.instructables.com/Android-receiving-data-from-arduino-via-bluetooth-/> (Android aplikazioan datuak bluetooth-etik nola irakurtzen diren azaltzen du)

<https://www.youtube.com/watch?v=ubTqh3c9v54> (Android aplikazio batean deiak nola egiten diren azaltzen du).

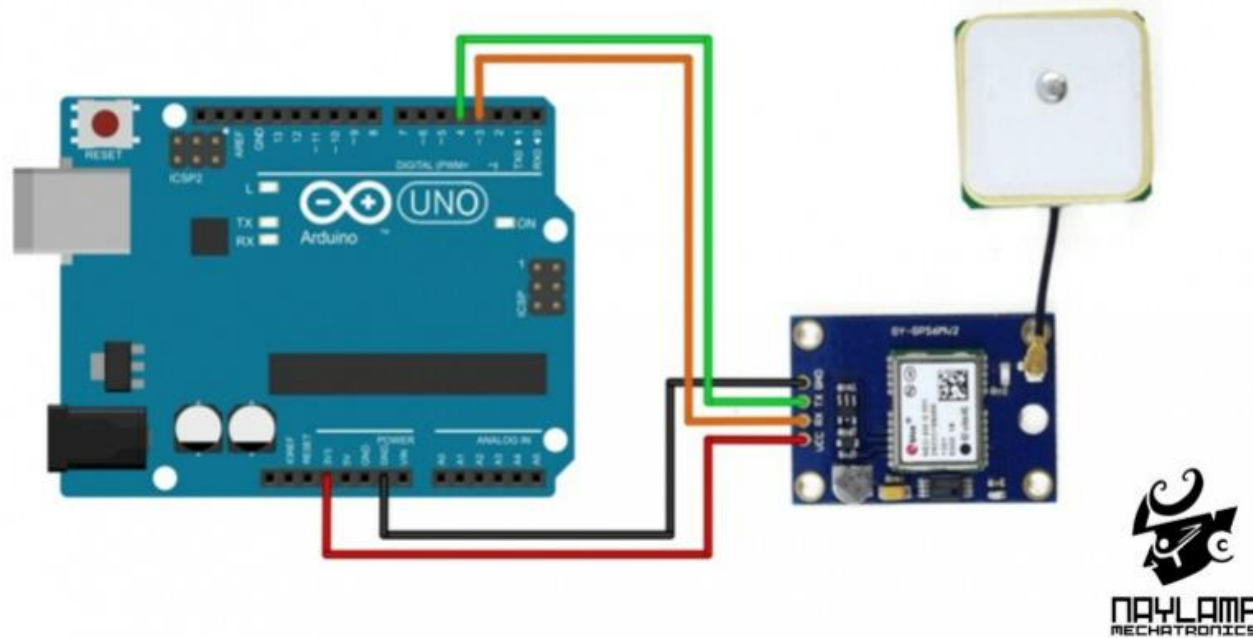
Send SMS

- *Bidali behar da SMS bat posizioarekin (latitudea eta longitudea).*
- *Bluetooth bitartez bidaliko dugu informazioa mugikorrera.*

Proposed Web:

<https://www.robotique.tech/robotics/creating-sms-application-with-app-inventor/>
(Android aplikaziotik nola bidali SMS bat azaltzen du).

NEO 6M GPS



Proposed website:

https://naylampmechatronics.com/blog/18_tutorial-modulo-gps-con-arduino.html

(GPS-a Arduniora nola konektatzen den eta programaren bitartez nola irakurtzen den posizioa azaltzen du)

*“How much more Arduino can do is
only left to your imagination”*

Thank you

Gaizka Bellido
gbellido@mondragon.edu