
Controlando un SumoBot Jr desde telegram

— Sergio Sinuco @sergiosinuco —
Pedro Caicedo @caicedo1089



Agenda

1. 09:00 Inicio y llegada de asistentes.
2. 09:15 Bienvenida al evento y presentación de patrocinadores.
3. 09:30 Intro a Arduino + Johnny Five.
4. 11:00 Cómo armar el Sumobot Jr.
5. 12:30 Almuerzo y networking.
6. 13:30 Creación y programación de un bot en Telegram.
7. 15:00 Piñata, torta, show de magia y más networking.
8. 16:00 Cierre

Sumobot Jr

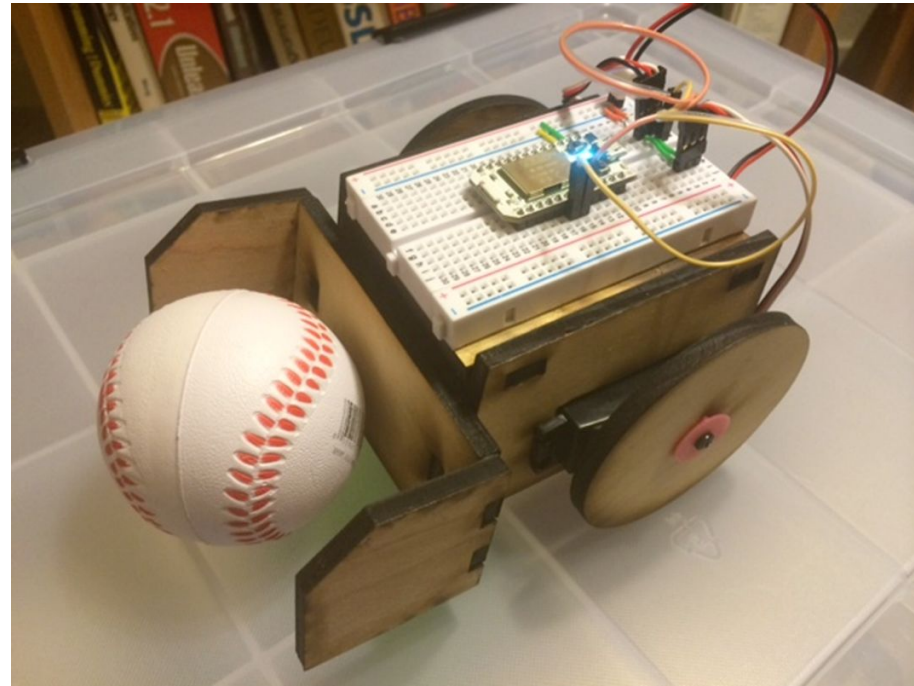
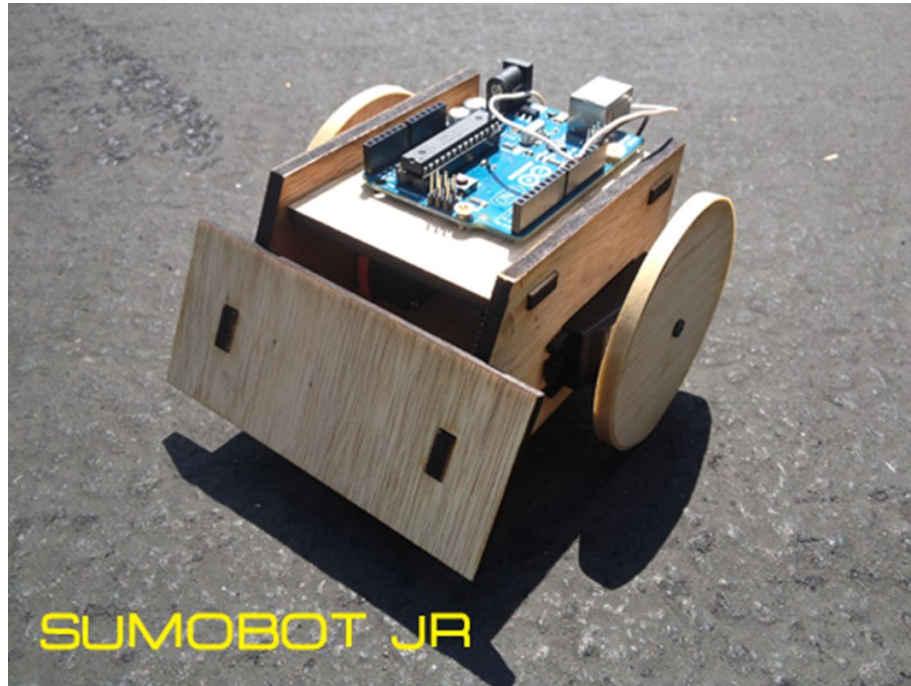
<https://github.com/makenai/sumobot-jr>

Por Pawel Szymczykowski,

Sumobot Jr. es un diseño de un sumobot para el Nodebots Day.

La mayoría de bots cuestan entre \$90USD y \$160USD, éste está diseñado para costar aproximadamente \$50USD.

Sumobot Jr



Las pa

Madera cortac

Impresión 3D

1 Esfera (o can

2 Servomotore

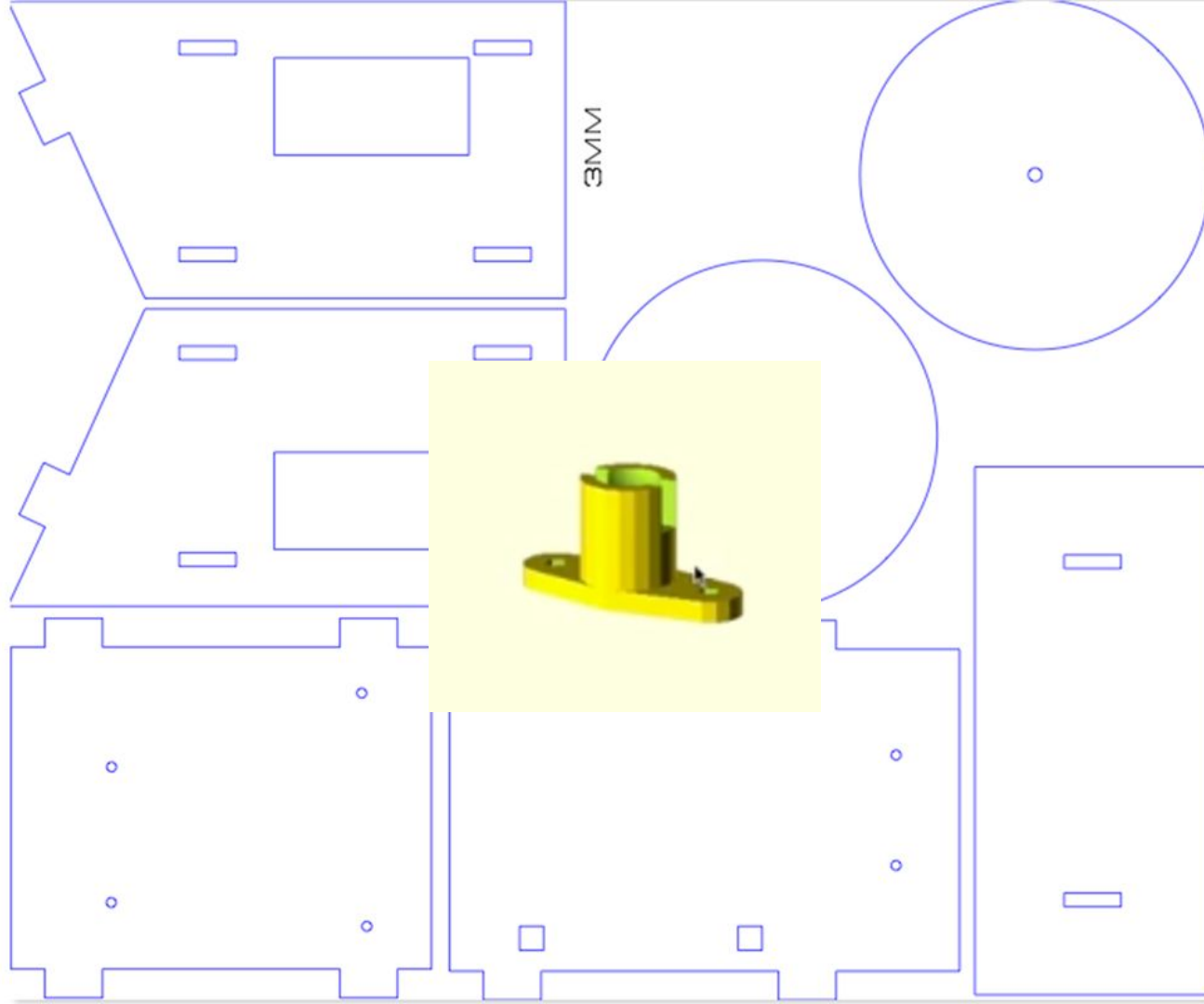
2 cauchos plan

2 amarres

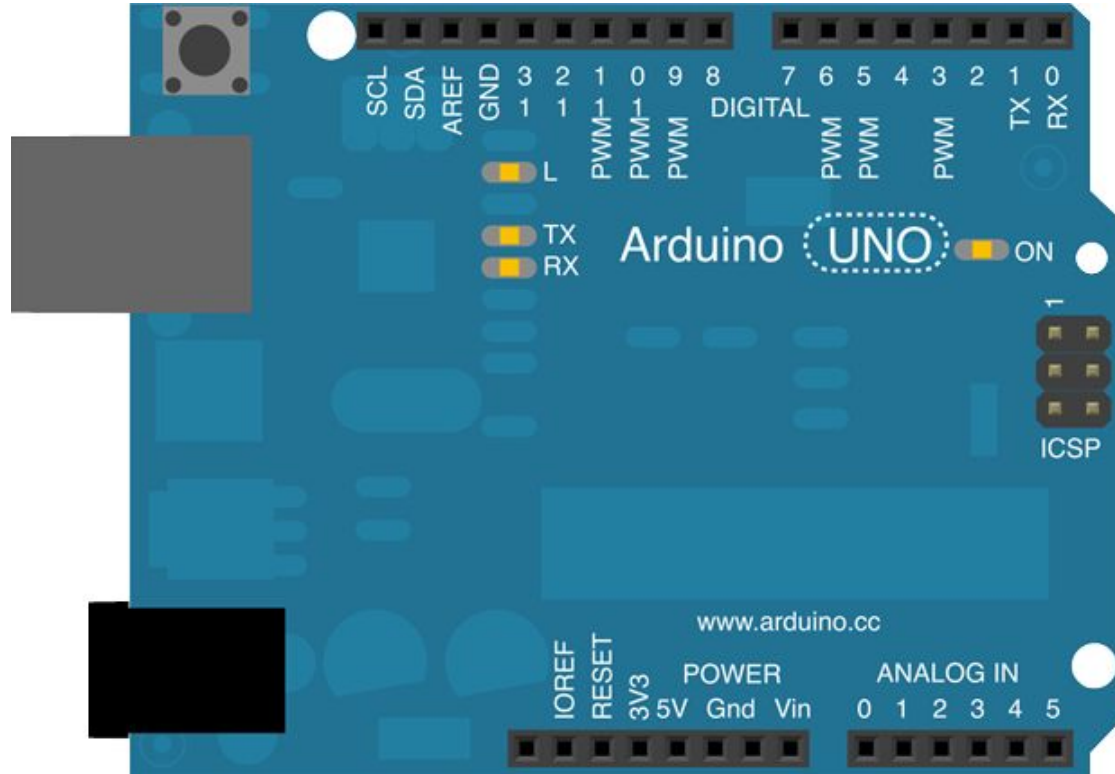
4 tornillos

1 Arduino

6 cables de pro



Arduino



Arduino

Open-source physical computing platform (?)(hardware + arduino sw)

Es de origen italiano. El colombiano Hernando Barragán es el Director de Estrategia de Diseño.

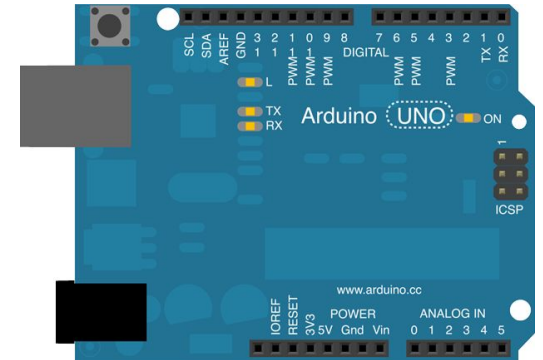
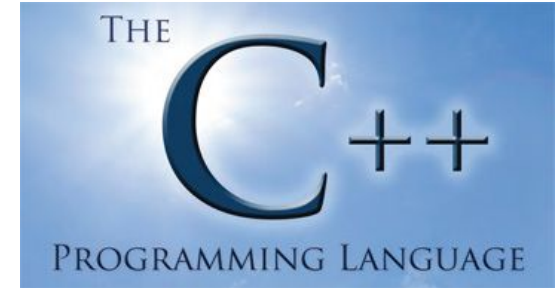
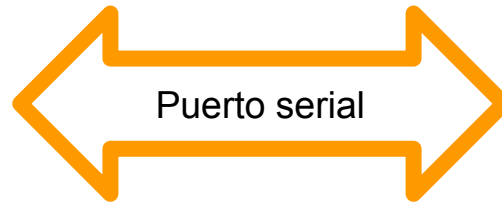
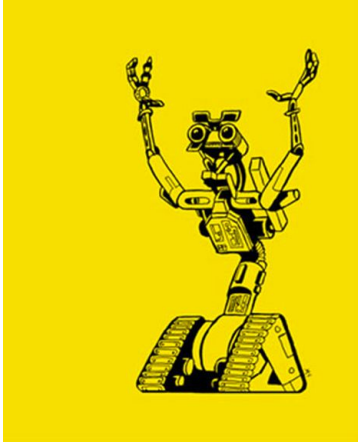
Arduino esta patentado (*duino)

El modelo principal es el Arduino Uno

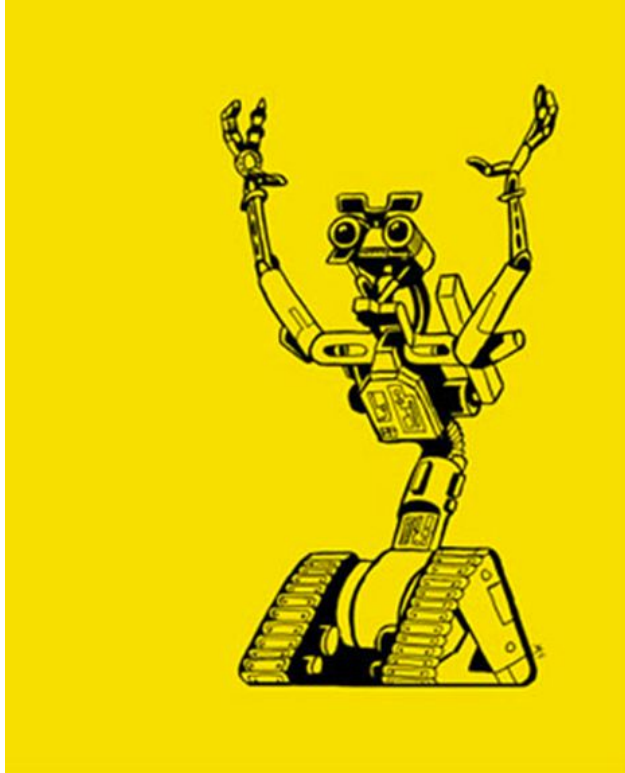
Se programa en C/C++

16Hz (Atmel AVR 8bit + Oscillator), 32K-128K Memory, 8K Ram, 6 Analog Inputs, 13 Digital Connections

Arquitectura del Sumobot Jr.



JohnnyFive



JohnnyFive

<http://johnny-five.io/>

Usa node-serialport de Chris Williams, para manejar comunicaciones seriales usando JavaScript

JohnnyFive de Rick Waldron, facilita el control de hardware (motores, servos, leds, etc) por medio de serialport.

Ejercicios con el arduino y JonnyFive

Encender y apagar un LED

Encender un RGB LED

Leer un sensor de temperatura

Descargar e instalar Arduino IDE

Download the Arduino IDE




ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

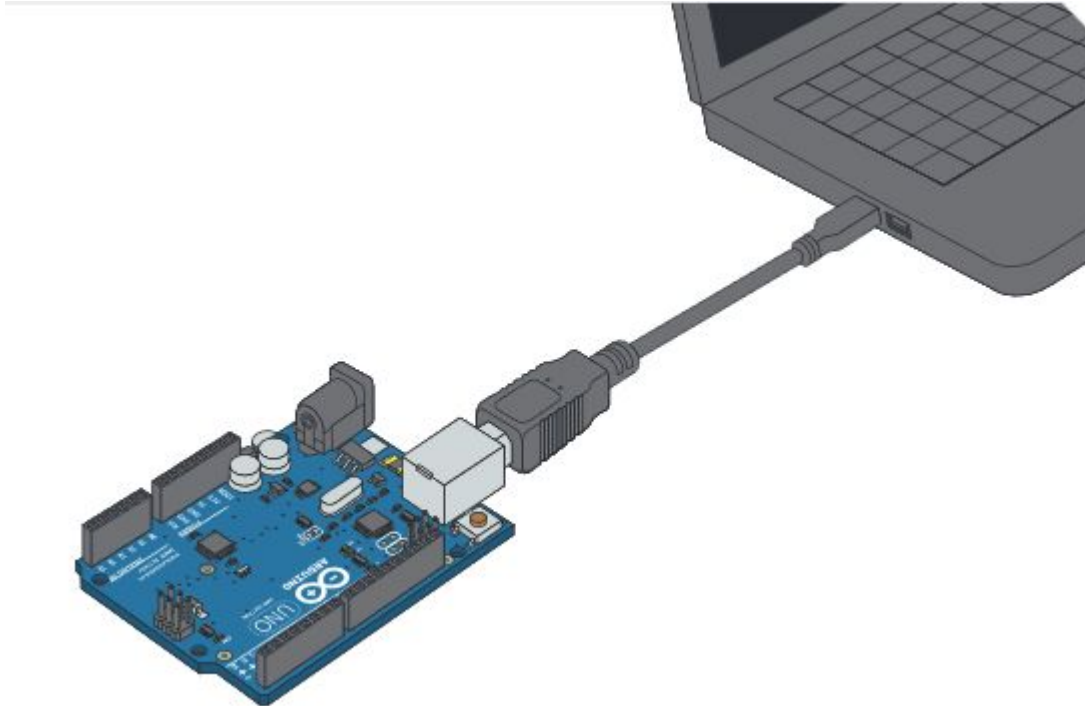
Descargar el código de ejemplo

git clone <https://github.com/ssinuco/ArduinoCvsJs.git>

cd ArduinoCvsJS

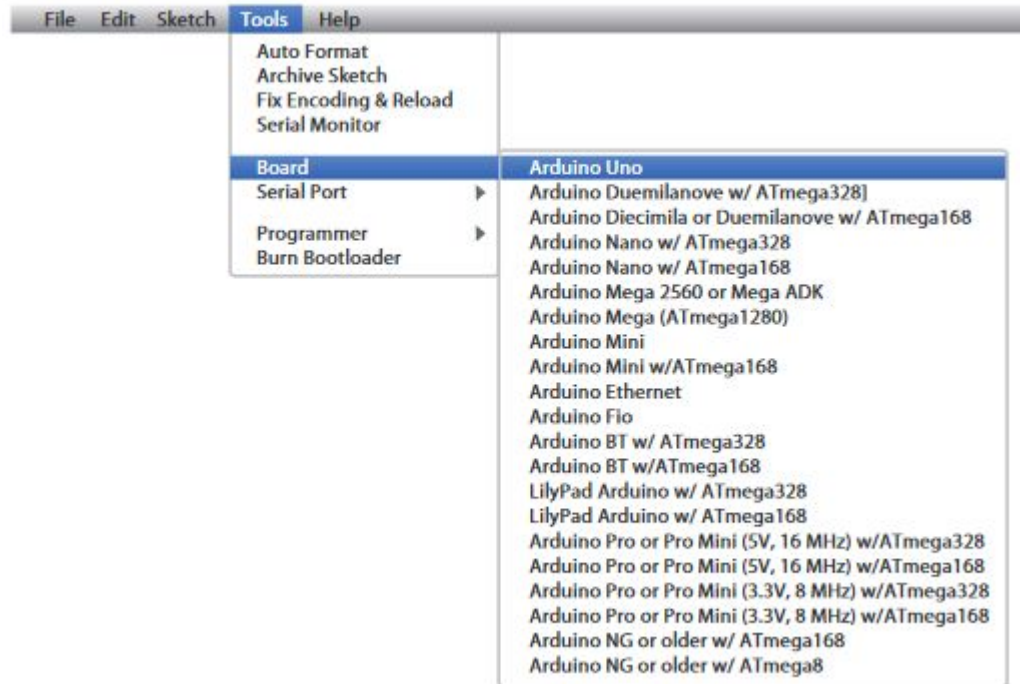
npm i

Conectar arduino al PC



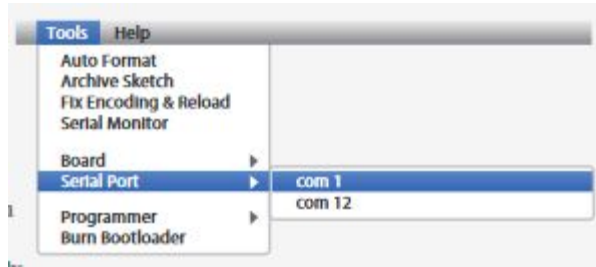
Tomado de : Vilros Ultimate Starter Kit Guide

Seleccionar la versión de arduino

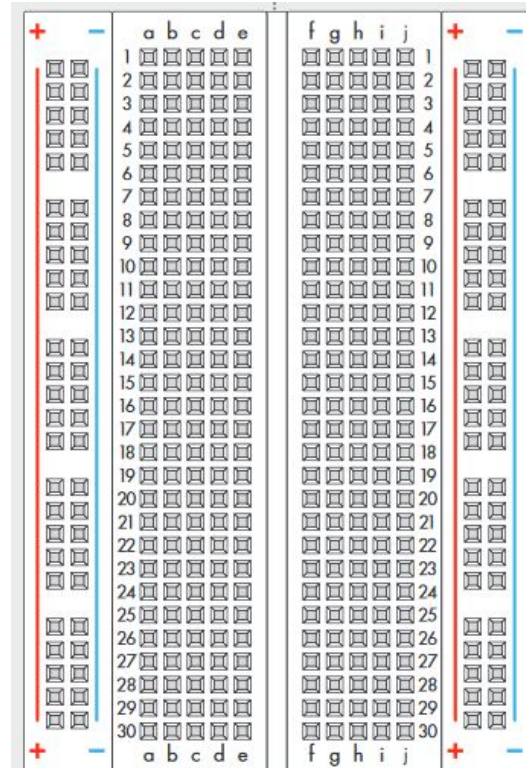


Tomado de : Vilros Ultimate Starter Kit Guide

Seleccionar el puerto



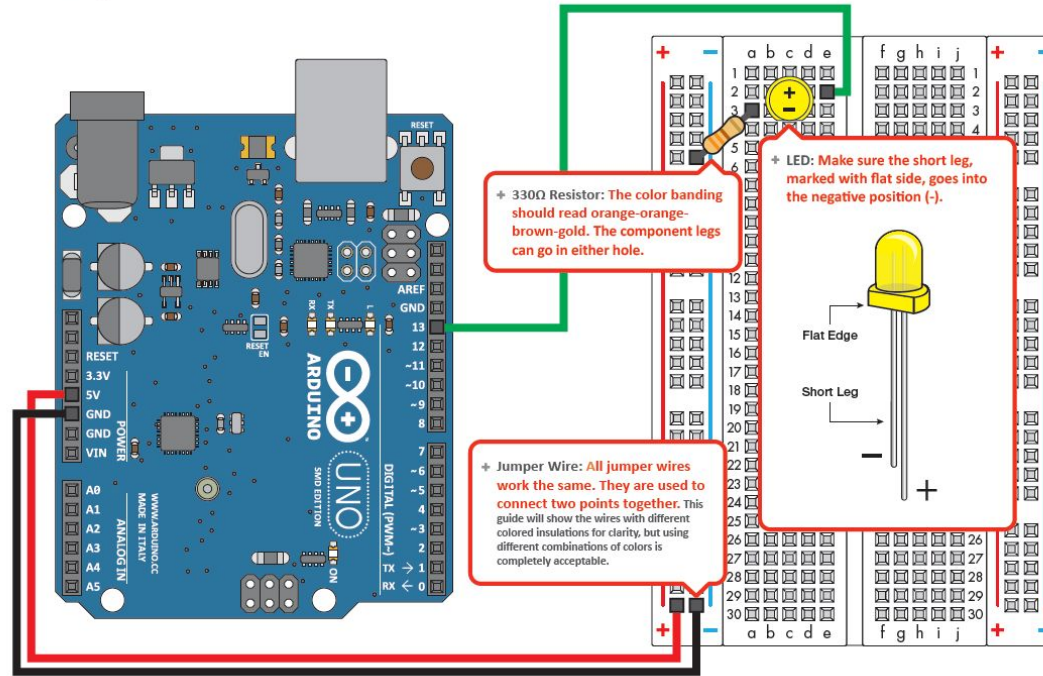
Protoboard o Breadboard



Tomado de : Vilros Ultimate Starter Kit Guide

Encender y apagar un LED - Circuito

Circuit 1: Blinking a LED



Encender y apagar un LED en C++

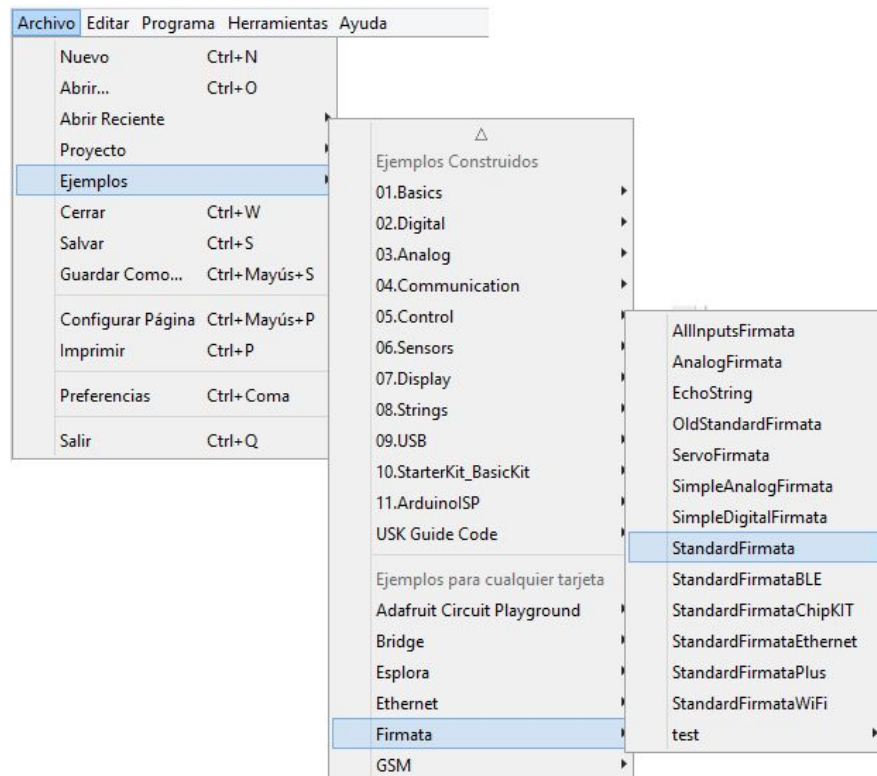
```
// the setup function runs once when you press reset or power the board
void setup()
{
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop()
{
    digitalWrite(13, HIGH);    // Turn on the LED

    delay(1000);               // Wait for one second

    digitalWrite(13, LOW);     // Turn off the LED

    delay(1000);               // Wait for one second
}
```

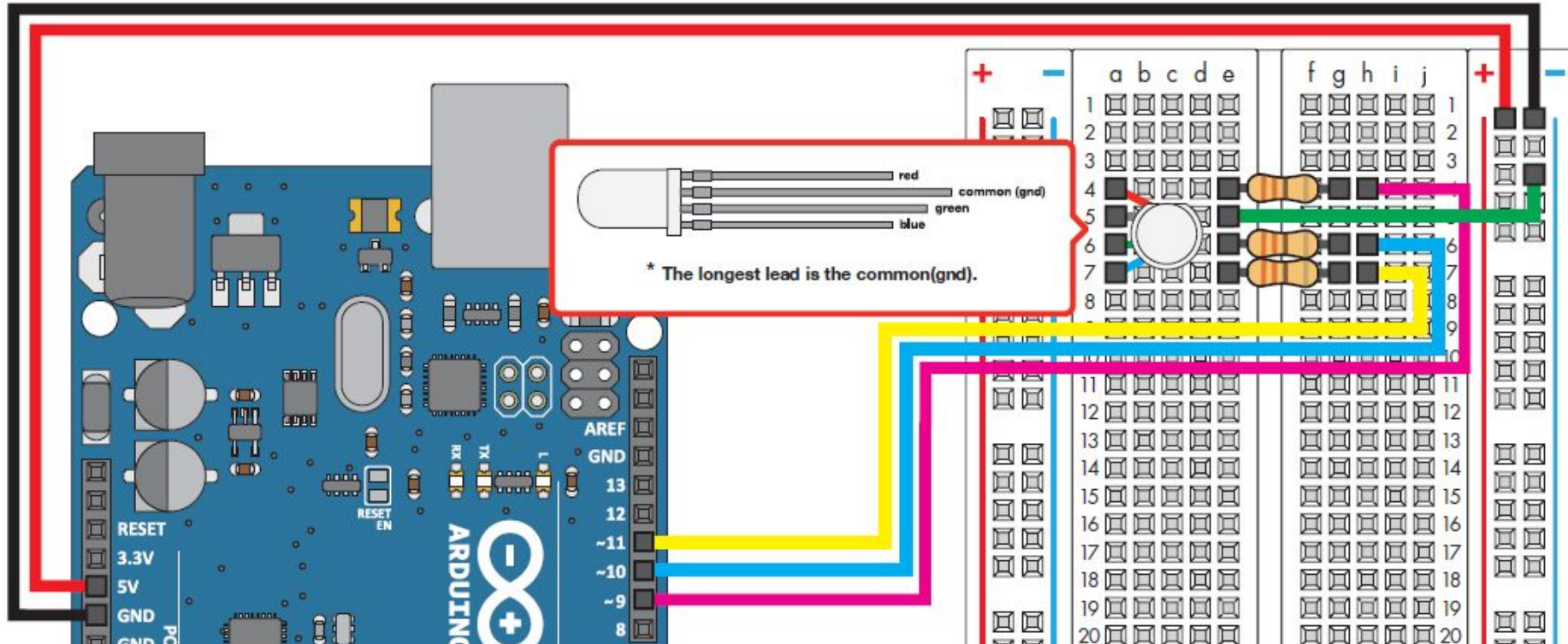
Encender y apagar un LED con JohnnyFive



Encender y apagar un LED con JohnnyFive

```
var five = require("johnny-five");  
var board = new five.Board();  
  
board.on("ready", function() {  
  var led = new five.Led(13);  
  led.blink(1000);  
});
```

Encender un RGB LED - Circuito

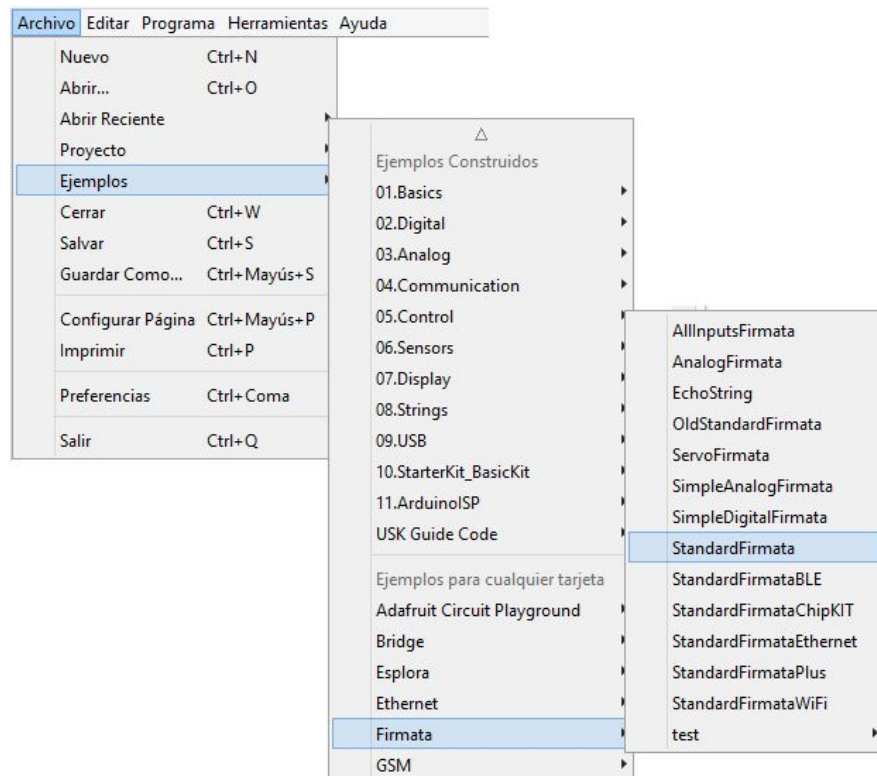


Tomado de : Vilros Ultimate Starter Kit Guide

Encender un RGB LED en C++

```
// the setup function runs once when you press reset or power the board
void setup()
{
    // Here we'll configure the Arduino pins we're using to drive the LED to be outputs:
    pinMode(RED_PIN, OUTPUT);
    pinMode(GREEN_PIN, OUTPUT);
    pinMode(BLUE_PIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop()
{
    // Yellow (turn red and green on):
    digitalWrite(RED_PIN, HIGH);
    digitalWrite(GREEN_PIN, HIGH);
    digitalWrite(BLUE_PIN, LOW);
}
```

Encender un RGB LED con JohnnyFive



Encender un RGB LED con JohnnyFive

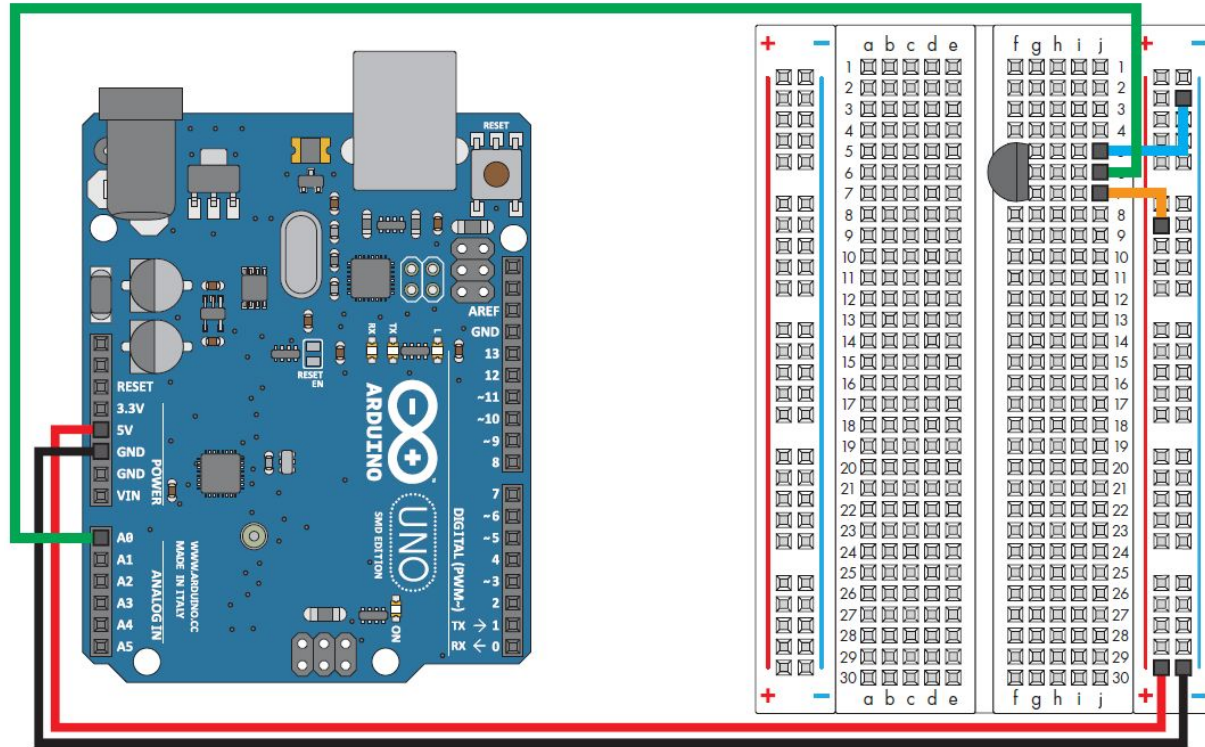
```
var five = require("johnny-five");  
var board = new five.Board();
```

```
board.on("ready", function() {  
  // Initialize the RGB LED  
  var led = new five.Led.RGB({  
    pins: {  
      red: 9,  
      green: 10,  
      blue: 11  
    }  
  });  
});
```

Encender un RGB LED con JohnnyFive

```
var colombia_flag = ["FFFF00", "0000FF", "FF0000"];  
var index = 0;  
this.loop(1000, function() {  
  led.color(colombia_flag[index++]);  
  if (index === colombia_flag.length) {  
    index = 0;  
  }  
});  
});
```

Leer un sensor de temperatura - Circuito



Tomado de : Vilros Ultimate Starter Kit Guide

Leer un sensor de temperatura en C++

```
// We'll use analog input 0 to measure the temperature sensor's signal pin.  
const int temperaturePin = 0;
```

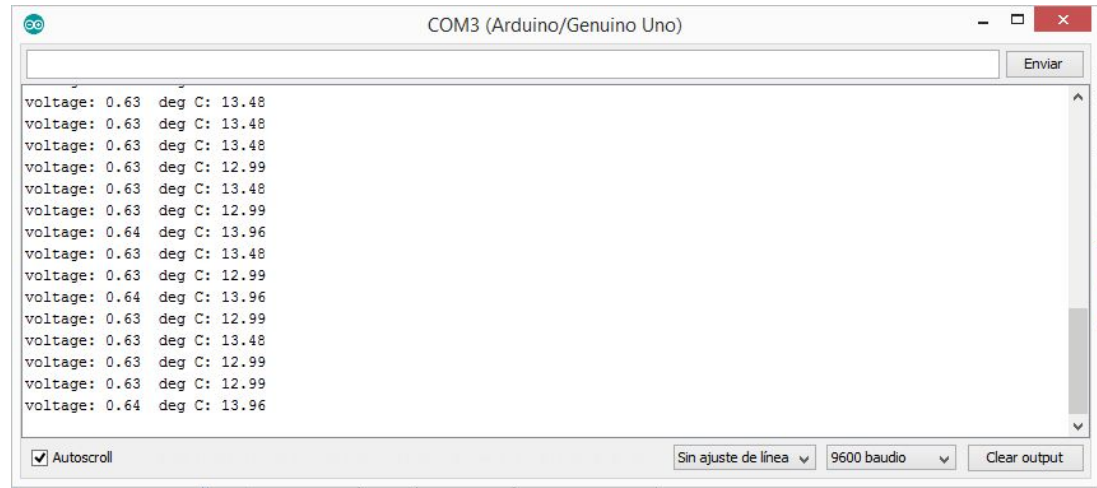
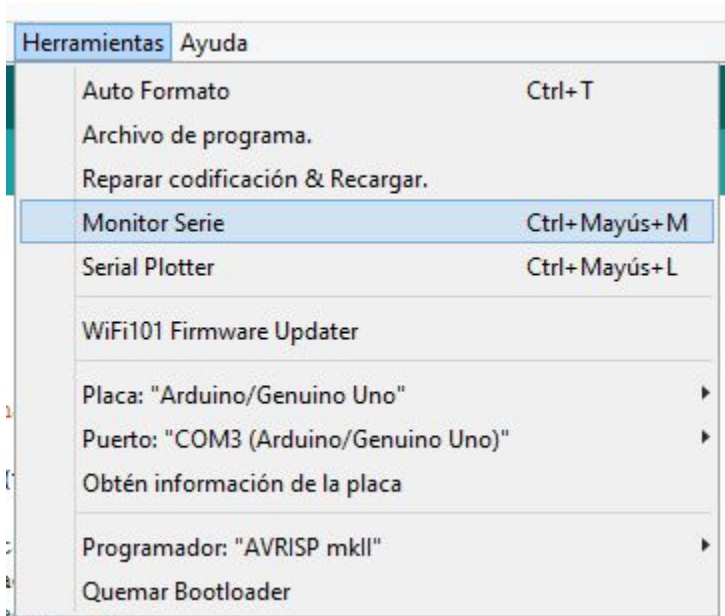
```
void setup()  
{  
  //Set Arduino's serial port speed  
  Serial.begin(9600);  
}
```

Leer un sensor de temperatura en C++

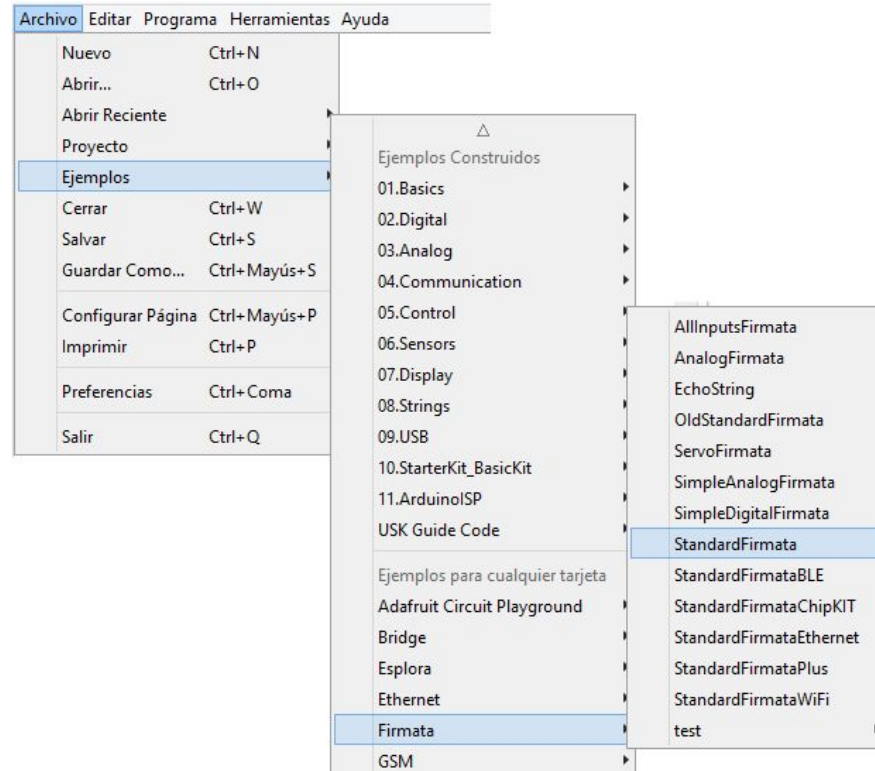
```
void loop()
{
    float voltage = analogRead(temperaturePin) * 0.004882814;
    float degreesC = (voltage - 0.5) * 100.0;

    Serial.print("voltage: ");
    Serial.print(voltage);
    Serial.print(" deg C: ");
    Serial.print(degreesC);
    delay(1000); // repeat once per second
}
```

Leer un sensor de temperatura en C++



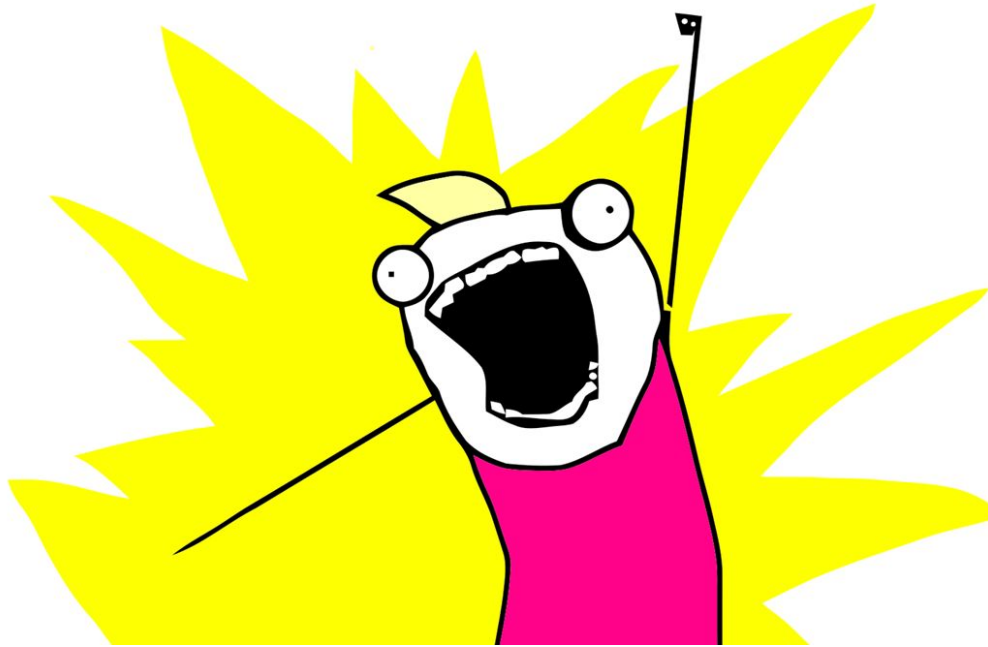
Leer un sensor de temperatura con JohnnyFive



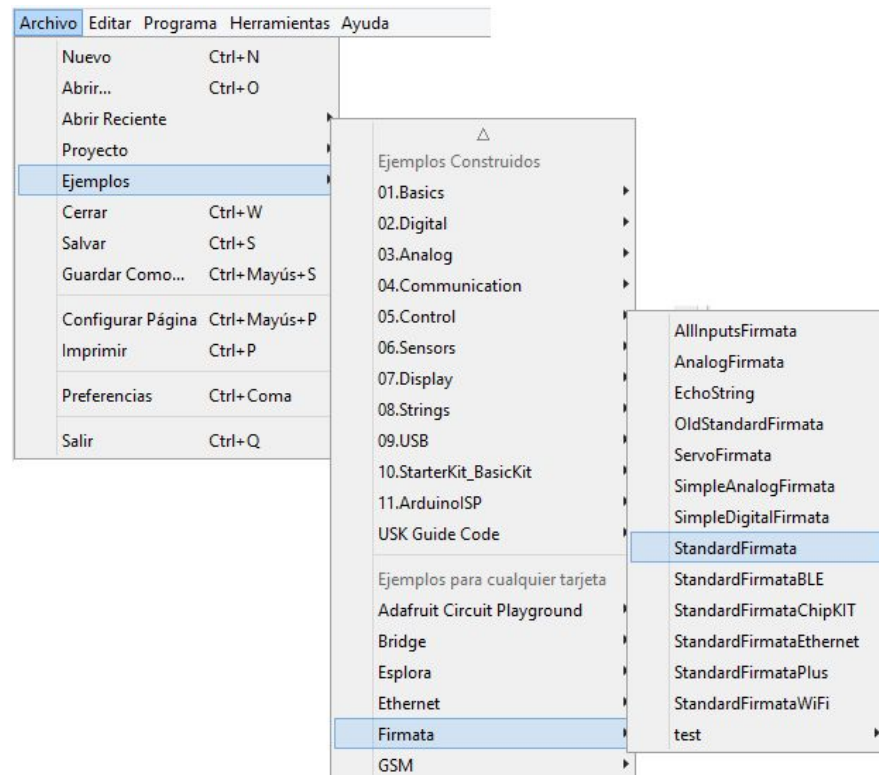
Leer un sensor de temperatura con JohnnyFive

```
var five = require("johnny-five");  
var board = new five.Board();  
board.on("ready", function() {  
  this.samplingInterval(1000);  
  this.pinMode(0, five.Pin.ANALOG);  
  
  this.analogRead(0, function(ADCValue) {  
    var voltage = ADCValue * 0.004882814;  
    var degreesC = (voltage - 0.5) * 100.0;  
    console.log("voltage: "+voltage+" deg C: "+degreesC);  
  });  
});
```

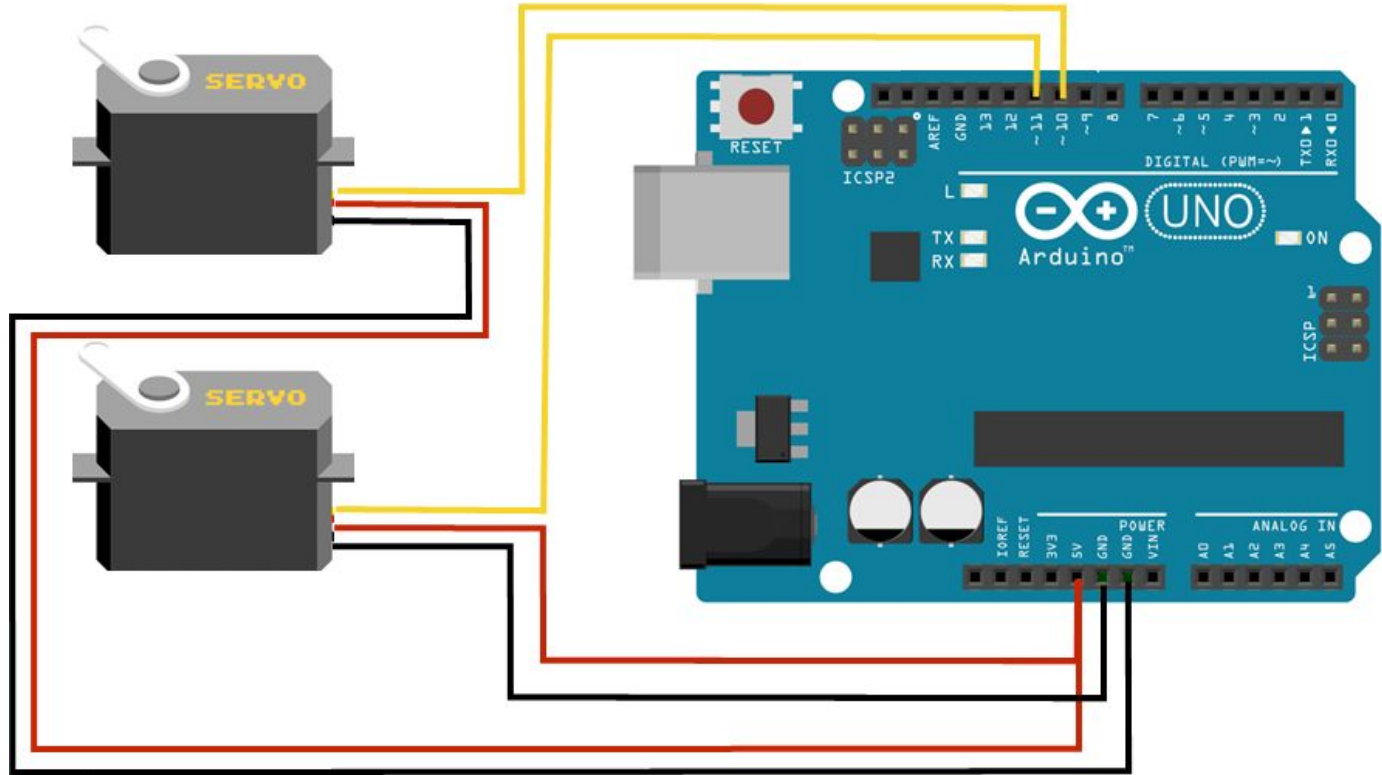

Construyamos el Sumobot Jr.



StandardFirmata para Sumobot Jr.



Sumobot Jr. Circuito



Descargar el código de ejemplo Sumobot Jr.

```
git clone https://github.com/makenai/sumobot-jr.git
```

```
cd code_example
```

```
npm i
```

```
node sumobot.js
```

Controla el Sumobot Jr. utilizando Telegram



Agenda

- ¿Qué es Telegram?
- ¿Qué son los bots?
- ¿Qué son los Chat Bots en Telegram?
- Creamos nuestro primer bot de Telegram
- ¿Cómo controlamos el sumobot jr utilizando un bot de Telegram?

¿Qué es Telegram?

- Es un servicio de mensajería por Internet desarrollado por hermanos Nikolai y Pavel Durov desde el 2013.
- Telegram utiliza su propio protocolo llamado **MTPROTO - Mobile Transport Protocol**
- Funcionalidades destacables:
 - Chat bots
 - Chats Secretos
 - MultiPlataforma
 - Soporte a Juegos
 - Chat consigo mismo, entre otras...

¿Qué son los bots?

- Un bot es un programa informático que simula el comportamiento humano
- ¿Qué pueden hacer los bots?
- ¿Dónde han utilizado bots?
 - Tal vez han jugado contra ellos

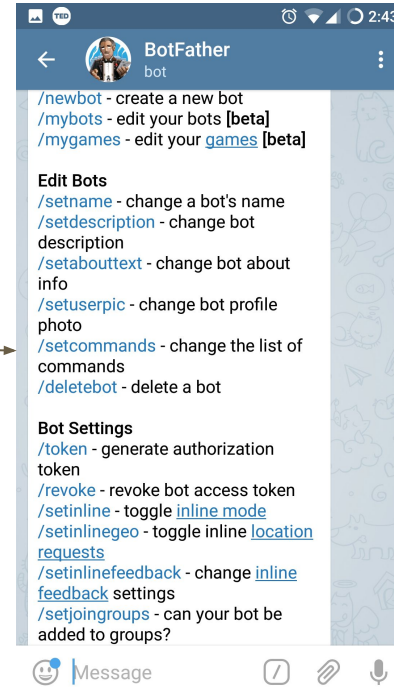
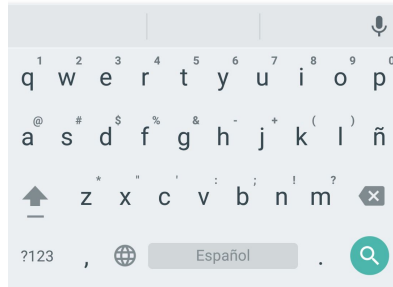
¿Chat bots en telegram?

- Es simplemente un bot que utiliza el API de Telegram, donde se simula una conversación (chat) y su principal interacción es el texto, pero se soporta fotos, video y sonido
- Los bots no inician conversaciones (no spam)
- Algunos Chat bots, populares:
 - @botFather - Creador de bots
 - @PollBot - Hacer encuestas
 - @AlertBot - Alertas programadas
 - @PokerBot - Jugar poker online

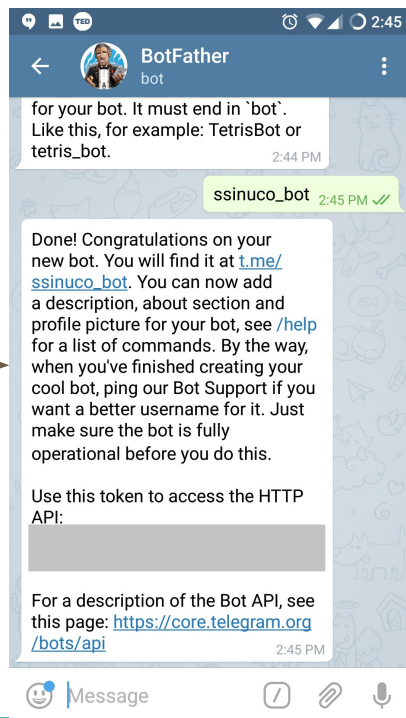
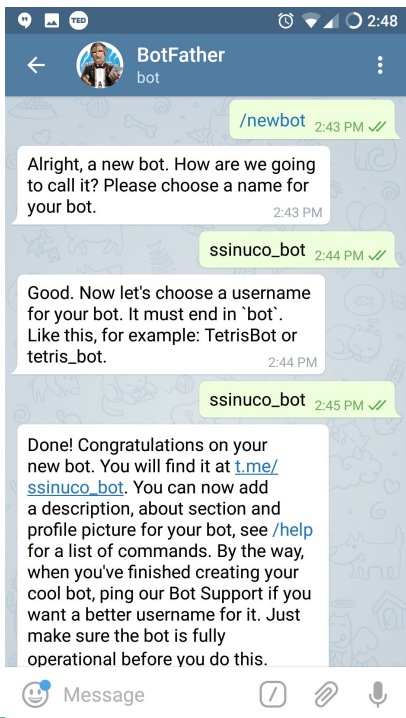
Para ver más chat bots revisar <https://storebot.me/>

Creamos nuestro primer bot de telegram 1/3

Le hablamos al BotFather y ejecutamos **/newbot**



Creamos nuestro primer bot de telegram 2/3




- Le asignamos un nombre y luego un username.
- El username debe terminar en bot.
- Y obtendremos una clave el cual usaremos para consumir el API de Telegram.

Creamos nuestro primer bot de telegram 3/3

Podemos acceder al api de telegram. https://api.telegram.org/bot<token>/METHOD_NAME

Ejemplo:

<https://api.telegram.org/bot123456:ABC-DEF1234ghIk1-zyx57W2v1u123ew11/getMe>



```
▼ {ok: true, result: {id: 474532882, is_bot: true, first_name: "PCaicedoBot", id: 474532882, is_bot: true, username: "PCaicedoBot"}}
```

Para ver el api completo -> <https://core.telegram.org/bots/api>

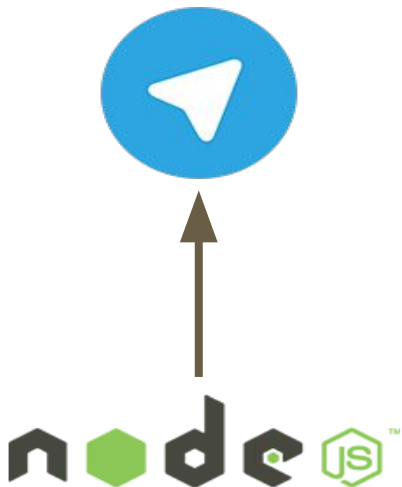
El API se puede consumir de dos modos:

- Consumiendo el método getUpdates (pull - consumo las actualizaciones)
- Configurando un webhook (push - me envían las actualizaciones a un punto de acceso web)

<https://core.telegram.org/bots/webhooks>

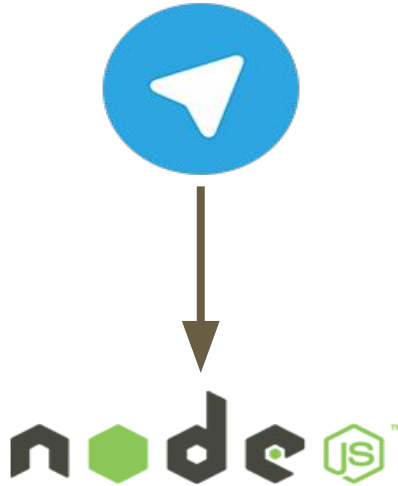
API Telegram getUpdates

- El cliente inicia la conversación para consumir el API (pull)



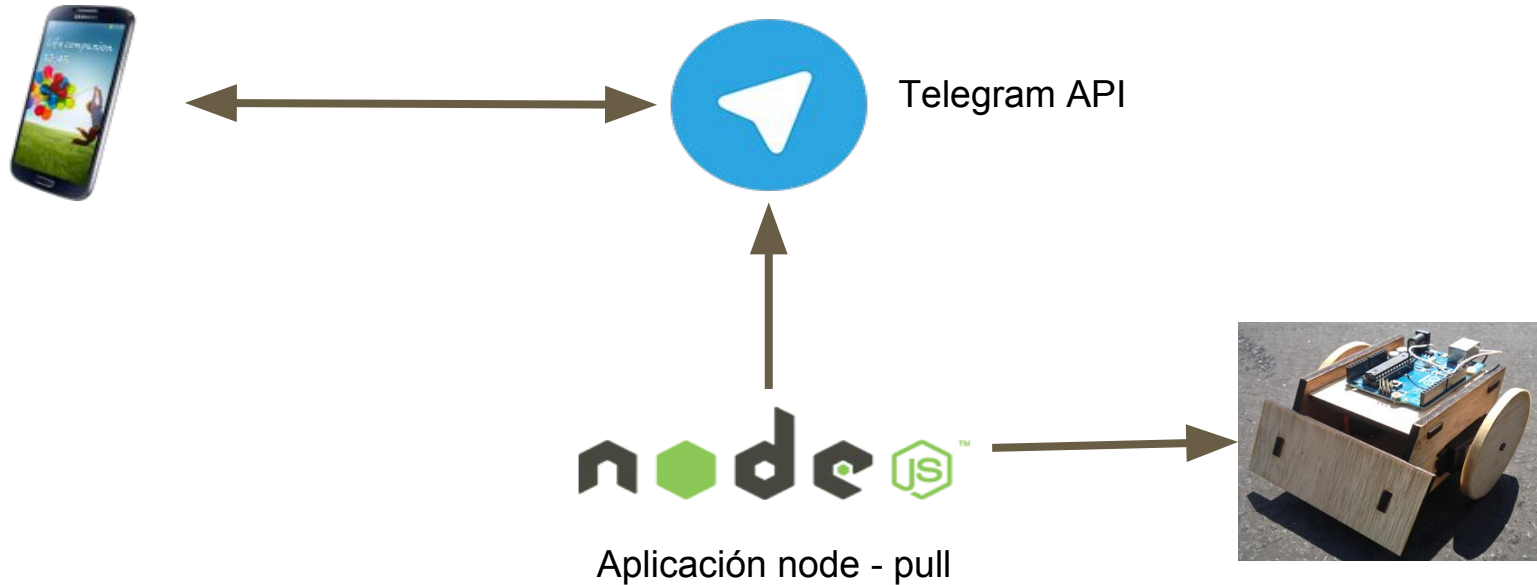
API Telegram webhook

- Telegram redirige los mensajes a un endpoint (push)



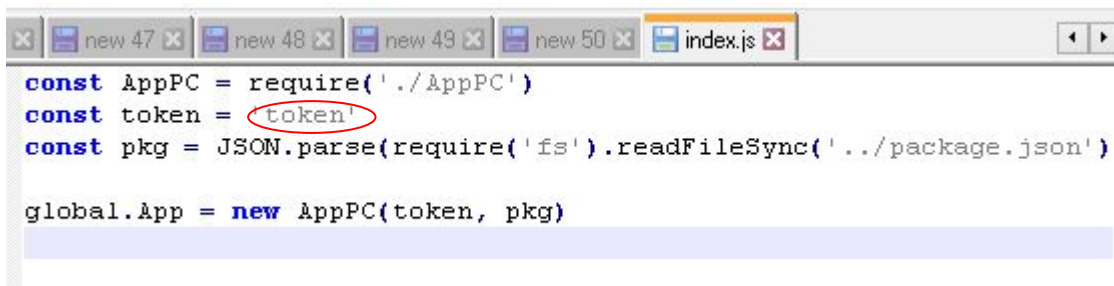
¿Cómo controlamos el sumobot jr utilizando un bot de telegram?

- Configuraremos un bot de Telegram, donde enviaremos los comandos, los capturaremos y ejecutaremos acciones al sumobot jr, según el comando enviado. El esquema gráfico es:



¿Cómo controlamos el sumobot jr utilizando un bot de telegram?

- Descargar el proyecto <https://github.com/caicedo1089/telegram-sumobotjr>
- Entrar a la carpeta ejecutar **\$> npm install**
- Luego abrir el archivo **index.js** en la carpeta **src/** y sustituir el **token** por el nuestro



```
const AppPC = require('./AppPC')
const token = 'token'
const pkg = JSON.parse(require('fs').readFileSync('../package.json'))

global.App = new AppPC(token, pkg)
```

- Para finalizar en la carpeta **src/** ejecutar el comando **\$> node index.js**

¿Cómo controlamos el sumobot jr utilizando un bot de telegram?

- Y listo debe quedar aplicación node escuchando, esperando comandos para el sumobot. Recuerden que deben tener conectado el sumobot jr al puerto USB antes de ejecutar la aplicación de node

```
pcaicedo@PC-Satellite-L45-B ~/Escritorio/bots/telegram-sumobotjr-master $ npm install  
[ ] / fetchMetadata: sill mapToRegistry uri https://registry.npmjs.org/browser-serialport
```

```
aicedo@PC-Satellite-L45-B ~/Escritorio/bots/telegram-sumobotjr-master $ cd src/  
aicedo@PC-Satellite-L45-B ~/Escritorio/bots/telegram-sumobotjr-master/src $ sudo node index.js  
comando "start" fue cargado.  
mando config  
comando "config" fue cargado.  
10863553471 Available /dev/ttyACM0  
10863553479 Connected /dev/ttyACM0  
10863557286 Repl Initialized  
█
```

¿Cómo controlamos el sumobot jr utilizando un bot de telegram?

- En el caso del bot en telegram debemos escribir /start (comando global al igual que /help y settings <https://core.telegram.org/bots#commands>) para iniciar el bot



¿Cómo controlamos el sumobot jr utilizando un bot de telegram?

- Felicidades ya tienen al sumobot js controlado desde un bot de Telegram



- **EL RETO:** agregarle más comandos al bot para que el sumobot jr tenga más opciones de movimiento. Ejemplo:
 - Moverse a la izquierda
 - Moverse a la derecha
 - O todas las combinaciones que desees.

Muchas gracias y feliz cumpleaños para Bogota JS

Pedro Caicedo @caicedo1089

Sergio Sinuco @sergiosinuco

