

Indice

Procesos iniciales para Escornabot en Ubuntu 16.04.....	2
Enlaces.....	2
Descarga e instalación del IDE de Arduino.....	2
Configurar IDE para tarjeta ESP8266.....	4
Control WiFi.....	6
Mando universal.....	6
Programador para el módulo ESP01.....	8
Programador con Arduino nano.....	8
Programador con Arduino UNO.....	9
Programación del ESP01 desde el IDE de Arduino.....	9
Comprobación del funcionamiento del ESP01.....	11
Control vía Bluetooth desde App para Android.....	11
Conexión Bluetooth.....	12
Procesos con la App.....	12
Comprobación del funcionamiento con Bluetooth.....	13
Reprogramación de Escornabot.....	13
Carga de nueva librería.....	14

Procesos iniciales para Escornabot en Ubuntu 16.04

En este documento vamos a ir describiendo como actuar desde cero para programar nuestro **Escornabot Brivoi** y controlarlo via WiFi para finalmente reprogramarlo con una librería externa.

Lo primero de todo y fundamental es tener nuestro Escornabot montado y en funcionamiento, cosa que resulta fácil de hacer dado que estos vienen ya con un firmware previo que permite testear el funcionamiento. Aunque ya se están desarrollando librerías para el mismo que veremos más adelante.

Vamos a meternos directamente en realizar el control remoto via WiFi programando desde el IDE de Arduino.

Enlaces

Toda la información oficial relativa a escornabot está en:


- Web: <https://escornabot.com/web/es>
- Repositorio: <https://github.com/escornabot>
- [Librería](#) desarrollada por Prudencio Luna y Pedro Ruiz para el [Club de Tecnología, programación y robótica de Granada](#)
- Programación ESP01: <https://aindustriosa.org/2017/09/escornabot-muwi-mucho-mas-que-wifi/>

Descarga e instalación del IDE de Arduino

Desde <https://www.arduino.cc/en/Main/Software> descargamos la última versión del IDE para nuestro sistema operativo. En este caso estos son los datos:

Si ya tenemos instalado el IDE pasamos directamente al apartado: Configurar IDE para tarjeta ESP8266.

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

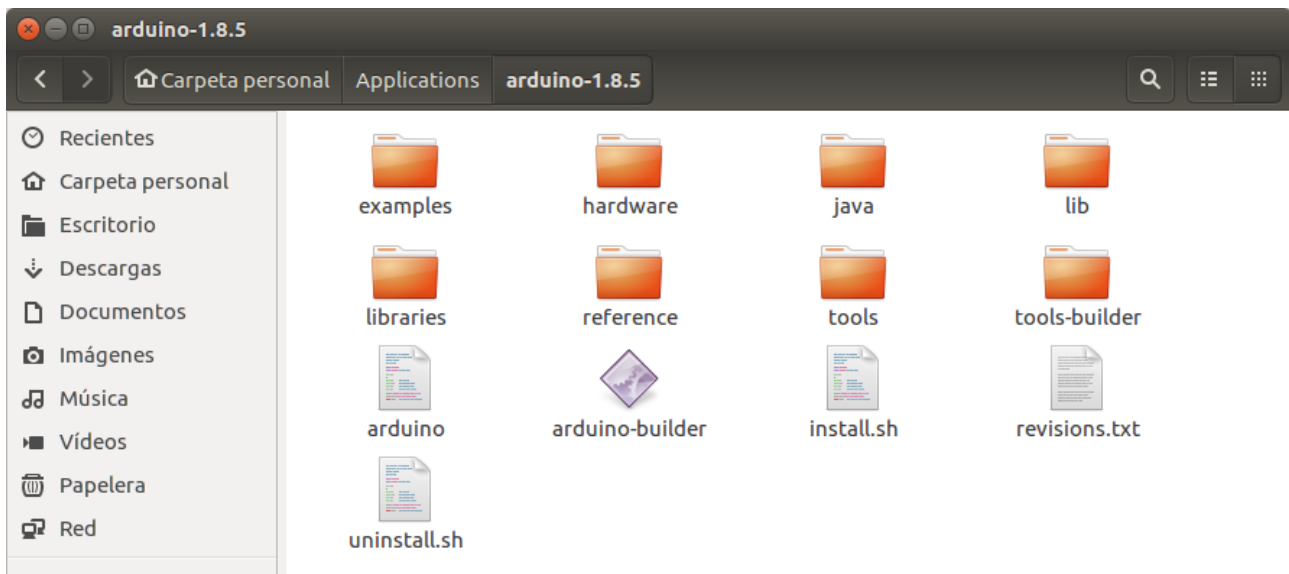
Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

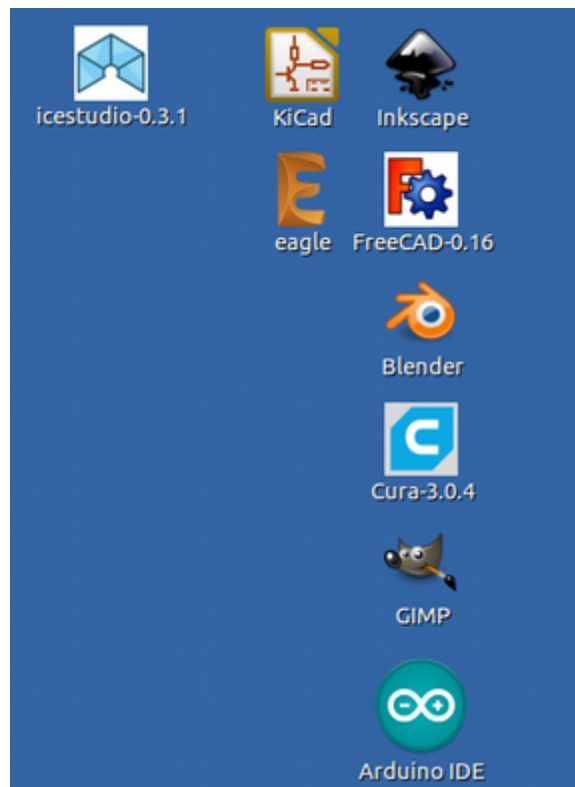
Una vez descargado descomprimos y tendremos algo similar a:



Ahora debemos ejecutar el escript de instalación, para lo que abrimos una terminal en la carpeta donde está ubicado el install.sh y lo hacemos.

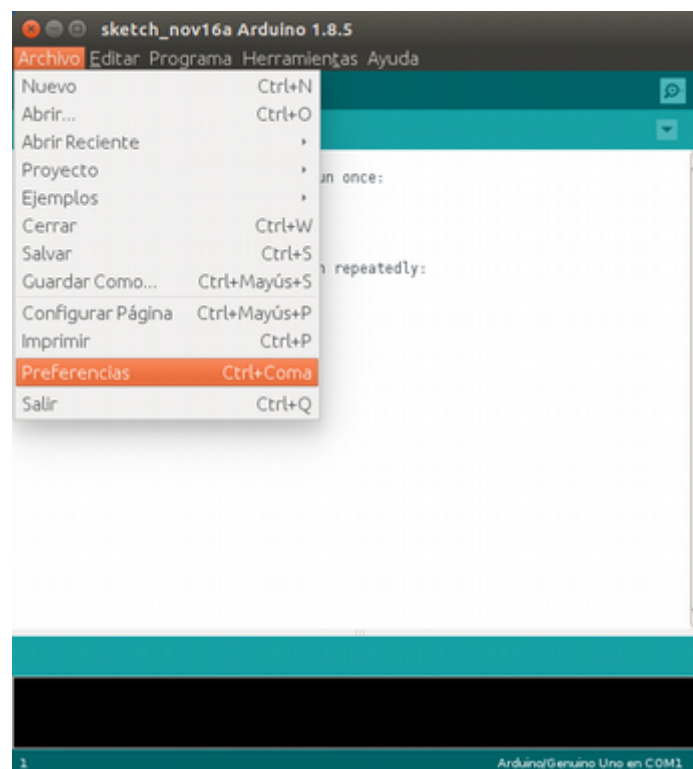
```
fede@fedo-KATANA: ~/Applications/arduino-1.8.5
fede@fedo-KATANA:~/Applications/arduino-1.8.5$ ./install.sh
Adding desktop shortcut, menu item and file associations for Arduino IDE...mv: ¿
reemplazar '/home/fede/.local/share/applications/mimeapps.list', sustituyendo el
modo 0644? (rw-r--r--) (s/n) s
mv: ¿reemplazar '/home/fede/.config/mimeapps.list', sustituyendo el modo 0644? (
rw-r--r--) (s/n) s
done!
fede@fedo-KATANA:~/Applications/arduino-1.8.5$
```

Se nos creará un acceso directo en el escritorio con el que iniciamos el IDE de Arduino.

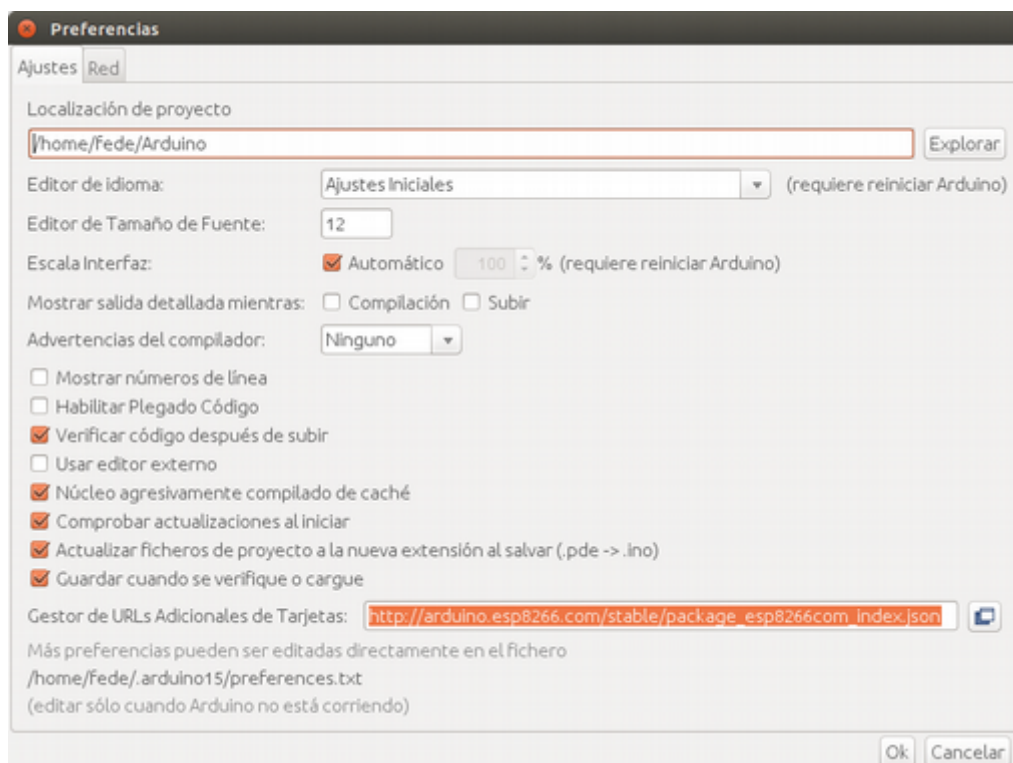


Configurar IDE para tarjeta ESP8266

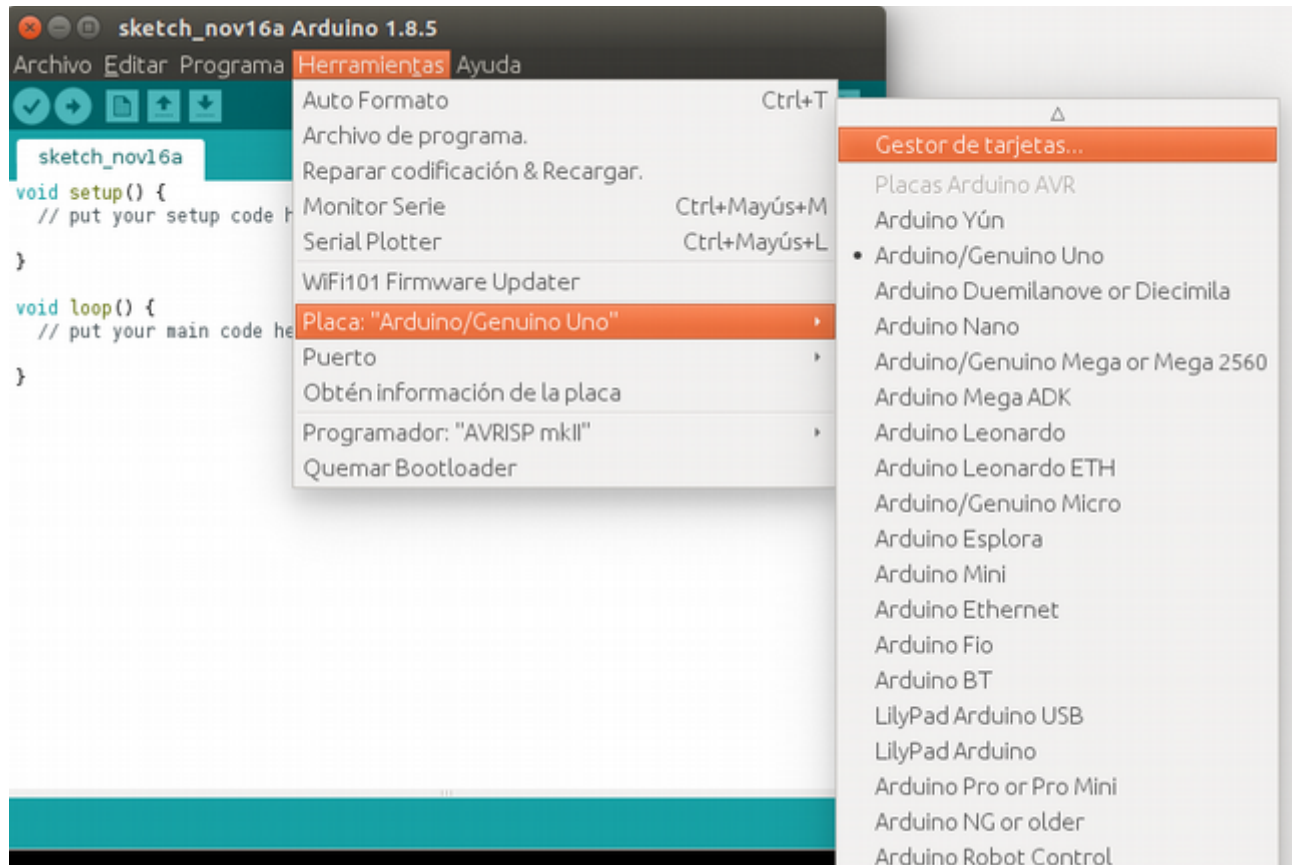
Una vez arrancado el IDE nos vamos a:



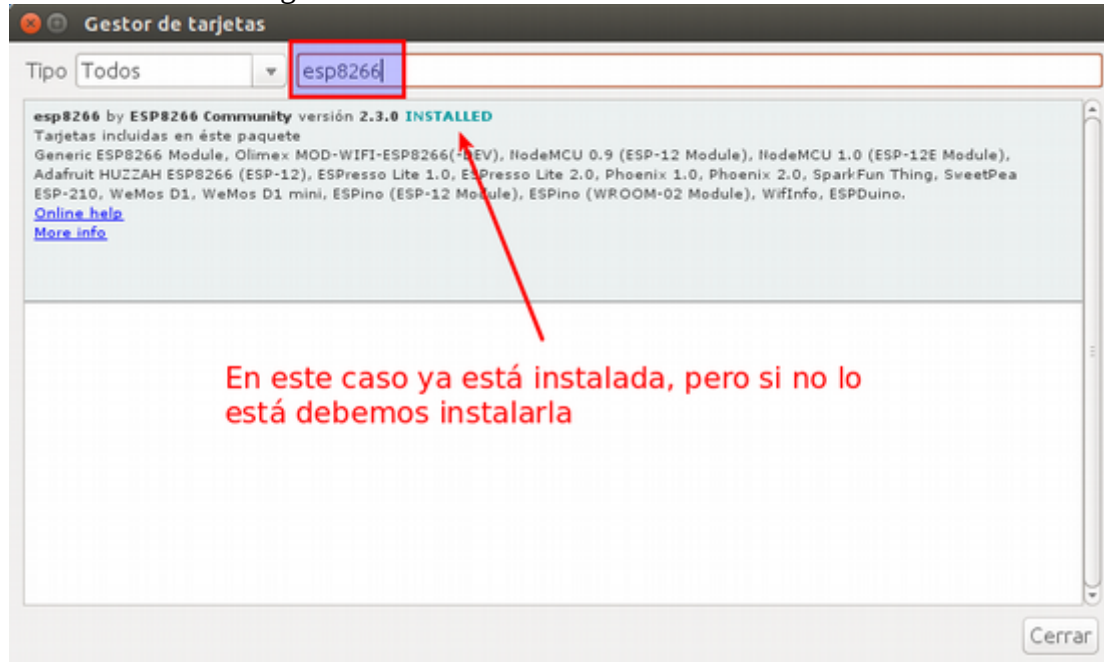
Y escribimos esta dirección http://arduino.esp8266.com/stable/package_esp8266com_index.json en el gestor de URLs.



Ahora vamos a agregar la placa.



Procedemos como en la imagen



Control WiFi

En Escornabot han desarrollado un mando universal para el control via WiFi mediante un módulo ESP01 que han llamado ESP-MUWi (Mando Universal WiFi) al que accedemos desde uno de los repositorios.

En el blog de <https://aindustriosa.org> encontramos una entrada donde nos describen todos los procesos para programar el módulo WiFi y el acceso al mando universal, el enlace es: <https://aindustriosa.org/2017/09/escornabot-muwi-mucho-mas-que-wifi/>

A partir de la versión 2.12 de Escornabot es posible llevar a cabo este proceso sin ningún problema, tal y como hemos hecho nosotros.

Mando universal

Tal y como se describe en el enlace citado lo que se pretende es reprogramar el ESP01 para que tenga el siguiente comportamiento:

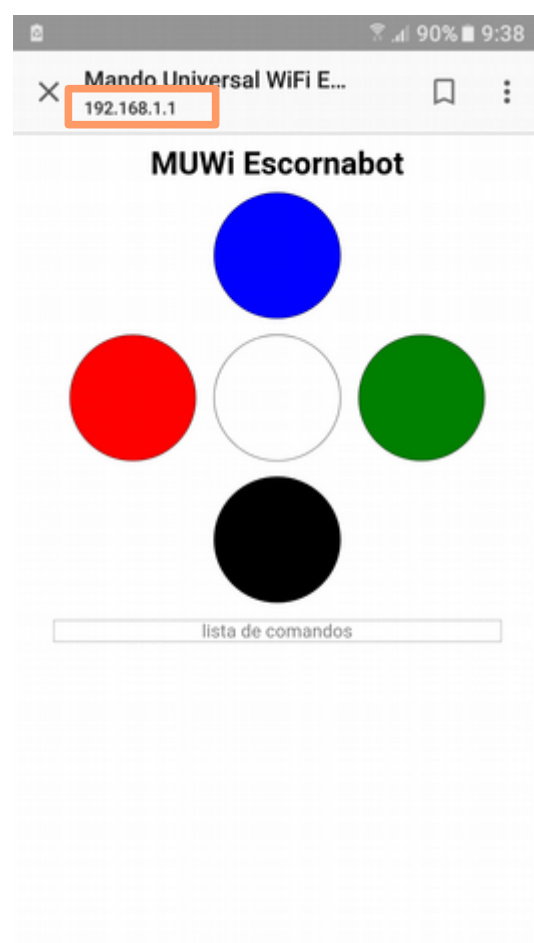
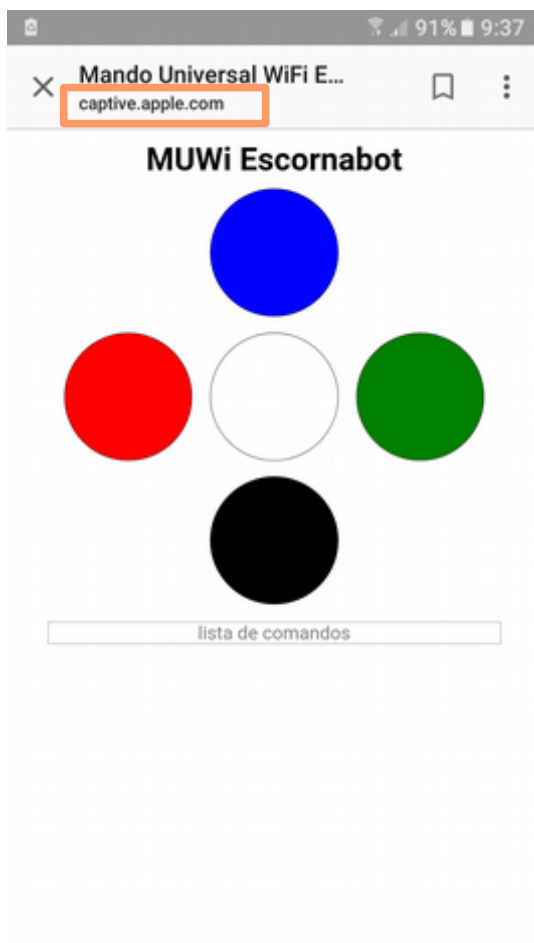
1. Levantar el punto de acceso WiFi ESCORNABOT en abierto (sin clave).
1. Crear un portal cautivo en el que todas las peticiones DNS serán atendidas por el propio ESP-01 y redirigidas a sí mismo.
2. Ejecutar un servidor web que atienda todas las peticiones recibidas y que nos sirva la página web con el mando de control remoto.
3. Transmitir al Escornabot la información que se reciba de dicho mando.

Con el Escornabot encendido nos podremos conectar a él y acceder directamente al mando de control remoto.

Conectamos nuestro dispositivo con la nueva WiFi.



Ya podemos acceder a la web que nos sirve el mando universal por cualquiera de las vías que vemos a continuación.



Programador para el módulo ESP01

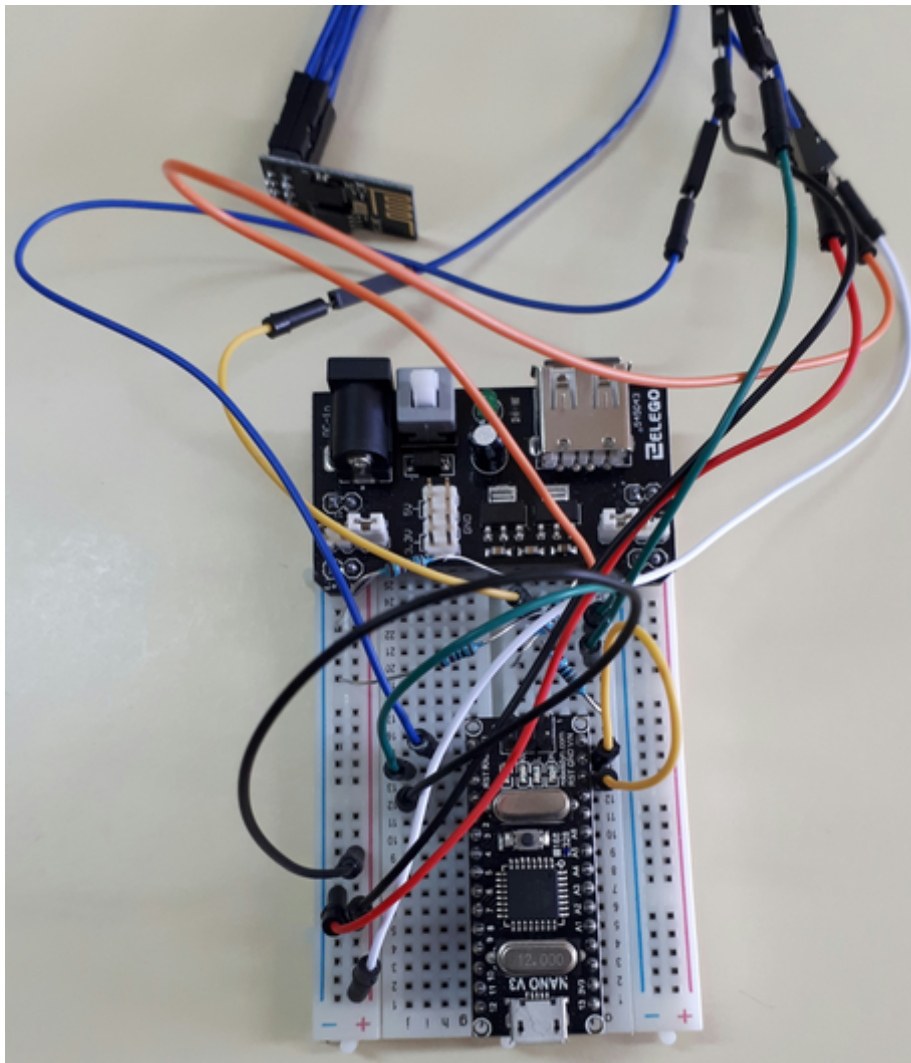
Aunque existen muchas formas de programar el ESP01 en la web explican muy claramente como programarlo utilizando un Arduino nano y siguiendo el procedimiento recomendado no debe plantearnos ningún tipo de problema. Es posible programarlo usando un Arduino UNO u otro modelo y en este caso, como dispone de un pin de alimentación de 3.3 V ya no es necesaria la fuente externa.

Programador con Arduino nano

Necesitamos los siguientes materiales:

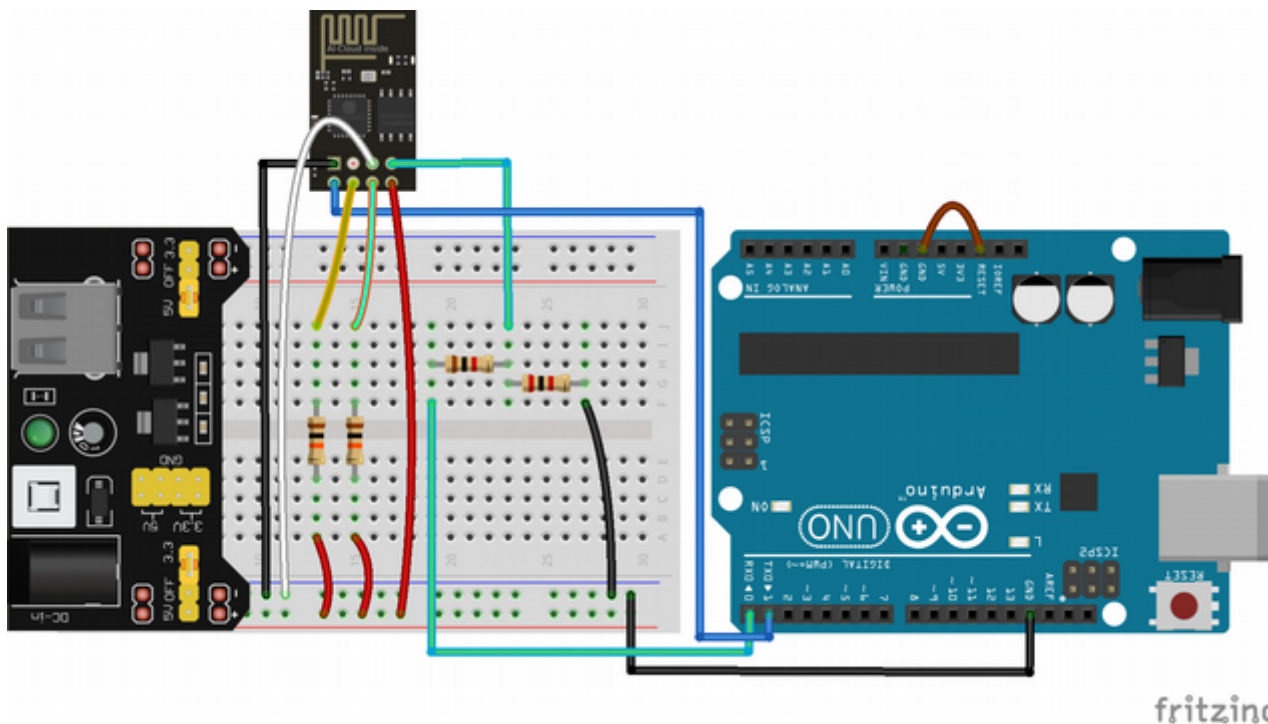
- 1 - Arduino Nano
- 1 - Protoboard
- 2 - Resistencia 10K Ω
- 1 - Resistencia 1K Ω
- 1 - Resistencia 2K Ω
- 1 - Fuente de alimentación 3.3 V y 250+ mA
- Cables de conexión

Realizamos el montaje que podemos ver en el enlace <https://aindustriosa.org/2017/09/escornabot-muwi-mucho-mas-que-wifi/>



Programador con Arduino UNO

Utilizando los mismos componentes realizamos el mismo montaje teniendo en cuenta que los pines que usamos para hacer nuestro programador son: TX, RX, GND y Reset.

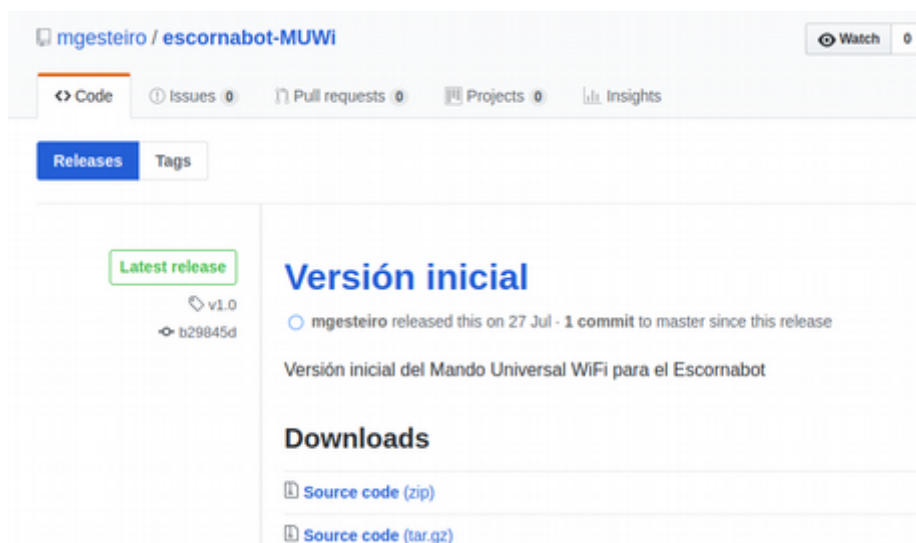


Aunque las placas Arduino disponen de pines de 3.3V podríamos pensar en usar este pin como fuente para nuestros programadores, sobre todo si no tenemos una fuente de alimentación externa de 3.3 V, pero no es aconsejable porque suele hacer la programación del módulo ESP01 muy inestable.

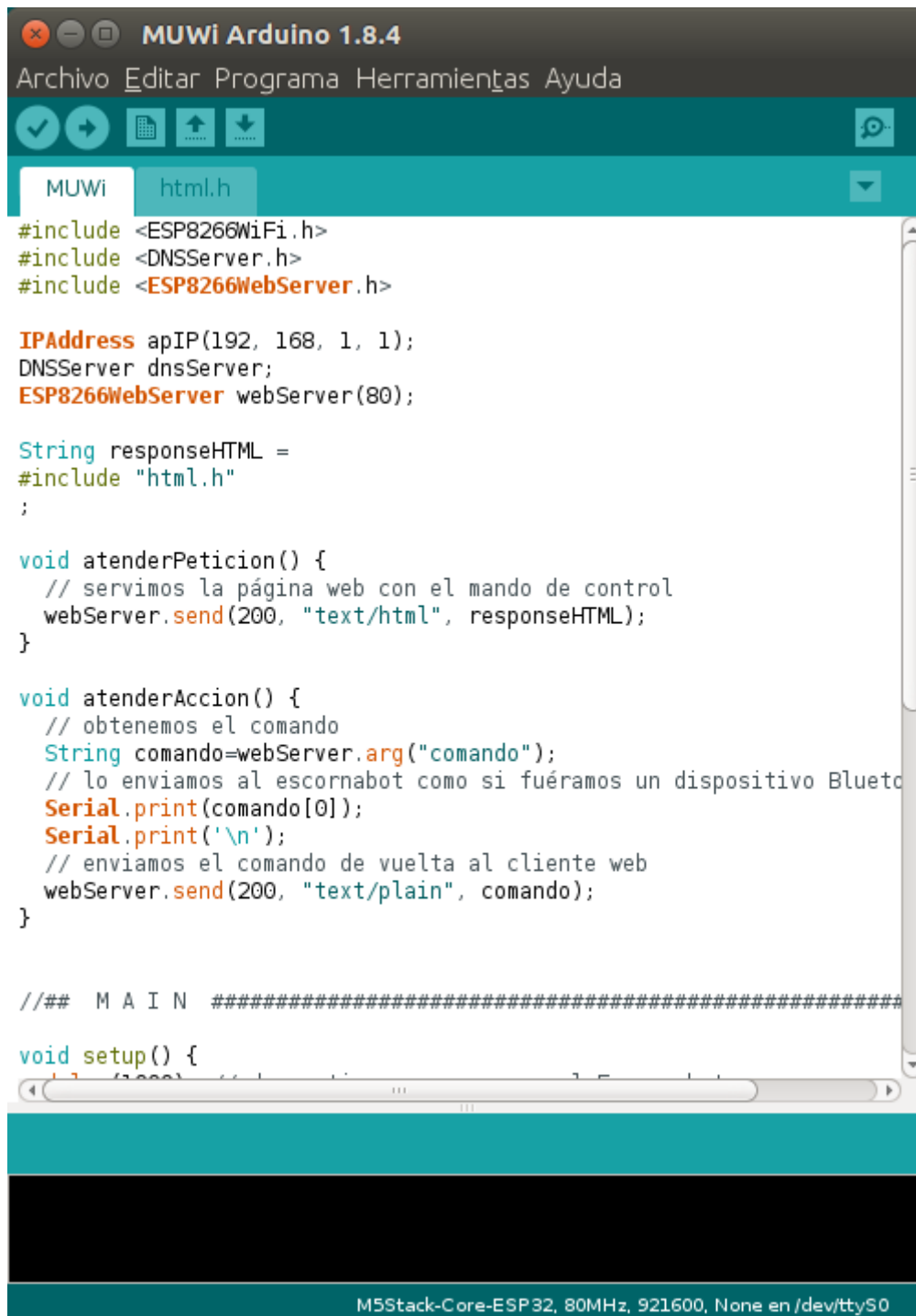
Programación del ESP01 desde el IDE de Arduino

Nos descargamos el código del firmware para el módulo ESP01 desde el enlace:

<https://github.com/mgesteiro/escornabot-MUWi/releases/latest>



Descomprimos el archivo descargado y abrimos el fichero MUWi.ino con el IDE de Arduino y ya podemos subirlo al ESP-01.



```

MUWi  html.h
#include <ESP8266WiFi.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>

IPAddress apIP(192, 168, 1, 1);
DNSServer dnsServer;
ESP8266WebServer webServer(80);

String responseHTML =
#include "html.h"
;

void atenderPeticion() {
  // servimos la página web con el mando de control
  webServer.send(200, "text/html", responseHTML);
}

void atenderAccion() {
  // obtenemos el comando
  String comando=webServer.arg("comando");
  // lo enviamos al escornabot como si fuéramos un dispositivo Bluetooth
  Serial.print(comando[0]);
  Serial.print('\n');
  // enviamos el comando de vuelta al cliente web
  webServer.send(200, "text/plain", comando);
}

//##  M A I N  #####

void setup() {
  // (3000)

```

M5Stack-Core-ESP32, 80MHz, 921600, None en /dev/ttyS0

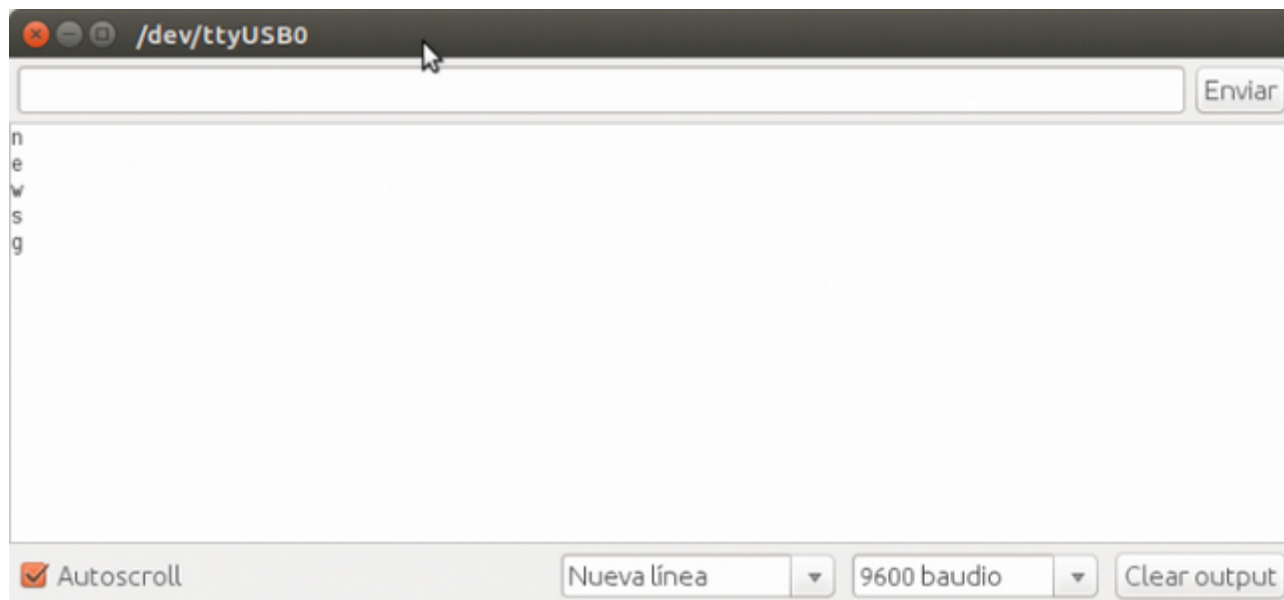
Puede ser necesario, para poder usar el puerto USB desde el IDE de Arduino en Ubuntu, añadir tu usuario al grupo dialout. En una terminal vamos a ejecutar el siguiente comando:

- `~$ sudo usermod -a -G dialout $USER` -> añadimos el usuario *USER* al grupo *dialout*.

El comando 'usermod' lo usamos para cambiar los atributos de un usuario ya existente. La opción -G nos permite indicar grupos suplementarios y la opción -a, añadir al usuario al grupo suplementario indicado en G sin eliminarlo de otros grupos. Es necesario para que el usuario pueda acceder a los dispositivos serie USB.

Comprobación del funcionamiento del ESP01

Si el proceso de programación ha finalizado correctamente, el ESP-01 ejecutará, justo a continuación, el programa recién cargado, con lo que podrás comprobar que todo funciona: para ello abre el Monitor Serie del IDE de Arduino (menú Herramientas -> Monitor Serie), configúralo a 9600 baudios, conéctate con tu móvil (o cualquier otro dispositivo) al punto de acceso del ESP-01 (= Escornabot), espera a que te aparezca el mando en la pantalla (tal y como se ve en la tercera imagen de este artículo) y pulsa sus teclas. Si todo es correcto, verás cómo salen en la ventana del Monitor Serie tus pulsaciones en forma de letra (n = North, s = South, e = East, w = West y g = Go):



En los siguientes videos vemos en acción el control remoto de EscornaBot y el funcionamiento autónomo mediante la botonera.

- Botonera onboard:

<https://youtu.be/Mb06r2Gx4bg>

- MUWi:

<https://www.youtube.com/8PU0huBNPnA>

Control vía Bluetooth desde App para Android

Jorge Lobo tiene preparada una sencilla aplicación para dispositivos Android con la herramienta MIT App Inventor. Se trata de una App simplemente funcional que podemos descargar desde cualquiera de los enlaces:

- Desde Google Drive:

<https://drive.google.com/file/d/0BwtFTf5rXVKGYmNjaXhzUk9ieGc/view?usp=sharing>

- Desde el enlace a este post:

<http://ceipmiskatonic.blogspot.com.es/2015/04/controlar-el-escornabot-por-bluetooth.html>

Toda la información sobre la App está disponible en:

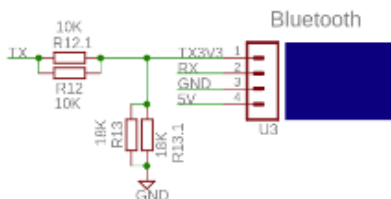
<https://github.com/escornabot/android>

El firmware para el control por Bluetooth con la clase BluetoothInterface, se puede descargar en:

<https://github.com/escornabot/arduino/releases>

Conexión Bluetooth

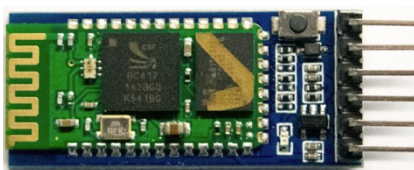
La conexión con el módulo Bluetooth con Arduino es TX con RX y RX con TX (cruzados). Las otras 2 necesarias serán GND y VCC.



Esquemático



PCB: Top y Boton



Bluetooth Top

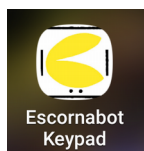


Bluetooth Boton



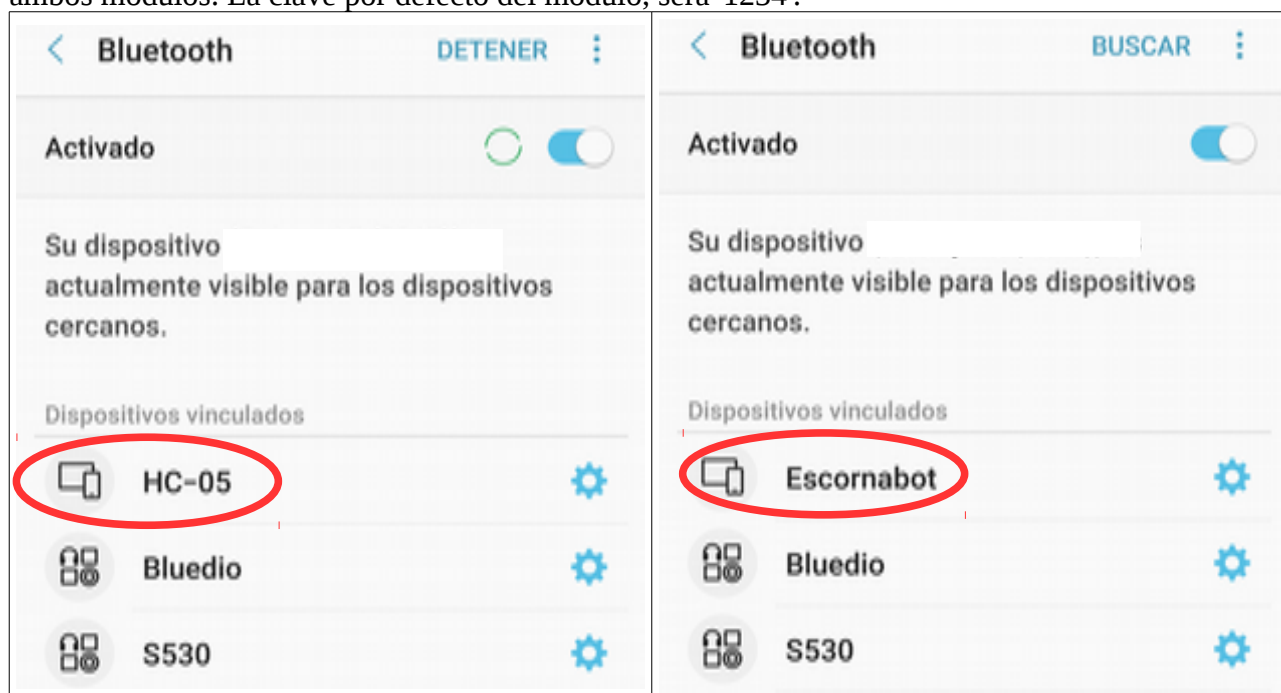
Montaje

Procesos con la App



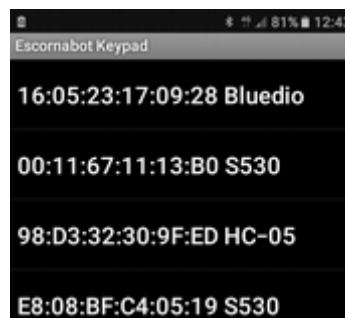
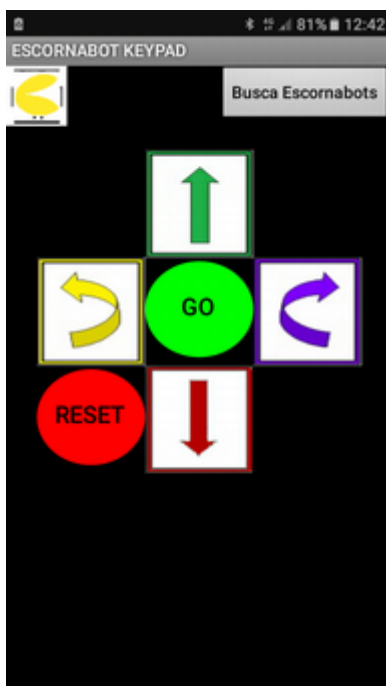
Una vez descargada e instalada en nuestro dispositivo tendremos un acceso directo a la misma, pero antes de iniciarla debemos emparejar nuestros dispositivos Bluetooth.

Con el Bluetooth de nuestro dispositivo activado y Escornabot encendido procedemos a vincular ambos módulos. La clave por defecto del módulo, será '1234'.



Si el Escornabot está conectado y emparejado con nuestro dispositivo, el LED presente en el adaptador Bluetooth debe estar iluminado y no parpadear.

Una vez hecho esto, buscaremos el dispositivo bluetooth del escornabot con el botón 'Buscar escornabots' de la app, y lo seleccionaremos. Ya tendremos todo listo para programar el escornabot del mismo modo que haríamos con la botonera.



Comprobación del funcionamiento con Bluetooth

Enlace al video demostrativo:

<https://youtu.be/33xc6AU-5a0>

Reprogramación de Escornabot

En esta sección vamos a trabajar con la librería desarrollada por Prudencio Luna y Pedro Ruiz del [Club de Robótica Granada](#) que podemos encontrar en su [repositorio](#).

Para ello es necesario reprogramar el Arduino de nuestro Brivois y antes de nada debemos asegurarnos de tener a mano los firmwares originales para posteriormente poderlo devolver a su estado original.

En el repositorio que alberga a este documento están disponibles los correspondientes a la versión de los mismos existente en el momento en el que se realizan las pruebas aunque, en los sitios oficiales de Escornabot podremos encontrar versiones más actuales.

- Firmware para Arduino:

<https://github.com/escornabot/arduino>

- Firmware MUWi:

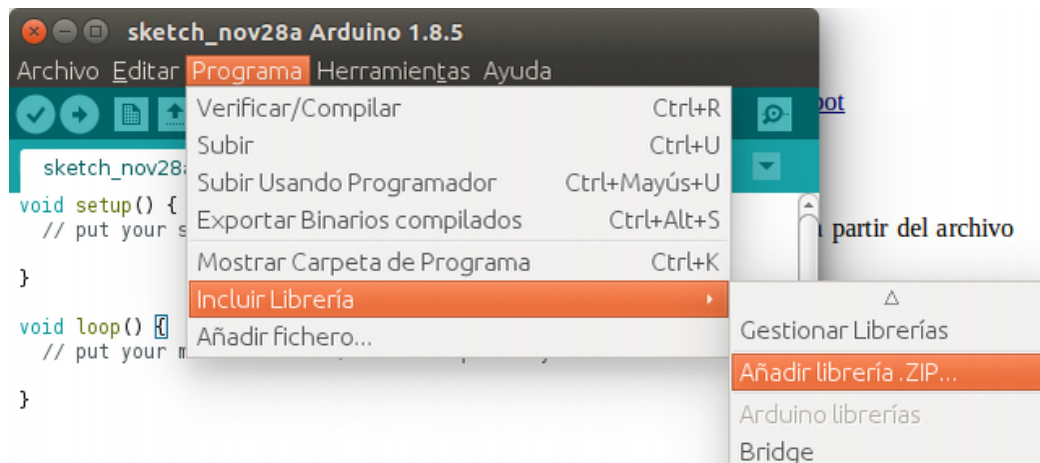
<https://github.com/escornabot/esp-muwi>

- Librería Club Robótica Granada

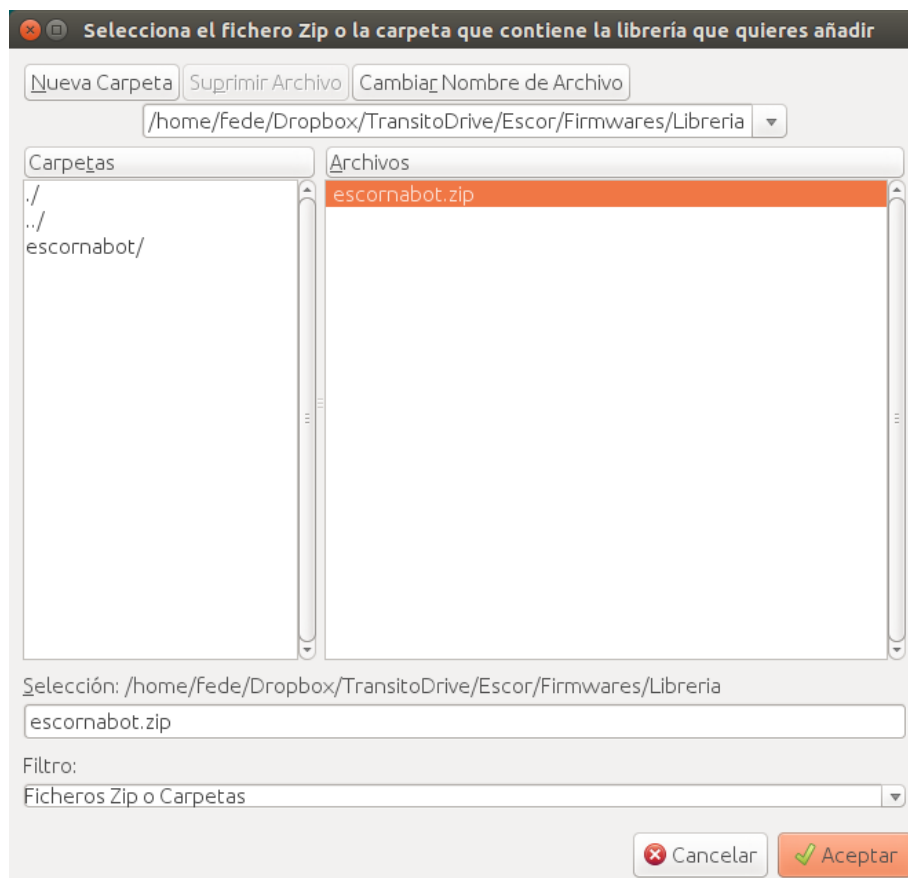
<https://github.com/clubroboticagranada/libreria-arduino-escornabot>

Carga de nueva librería

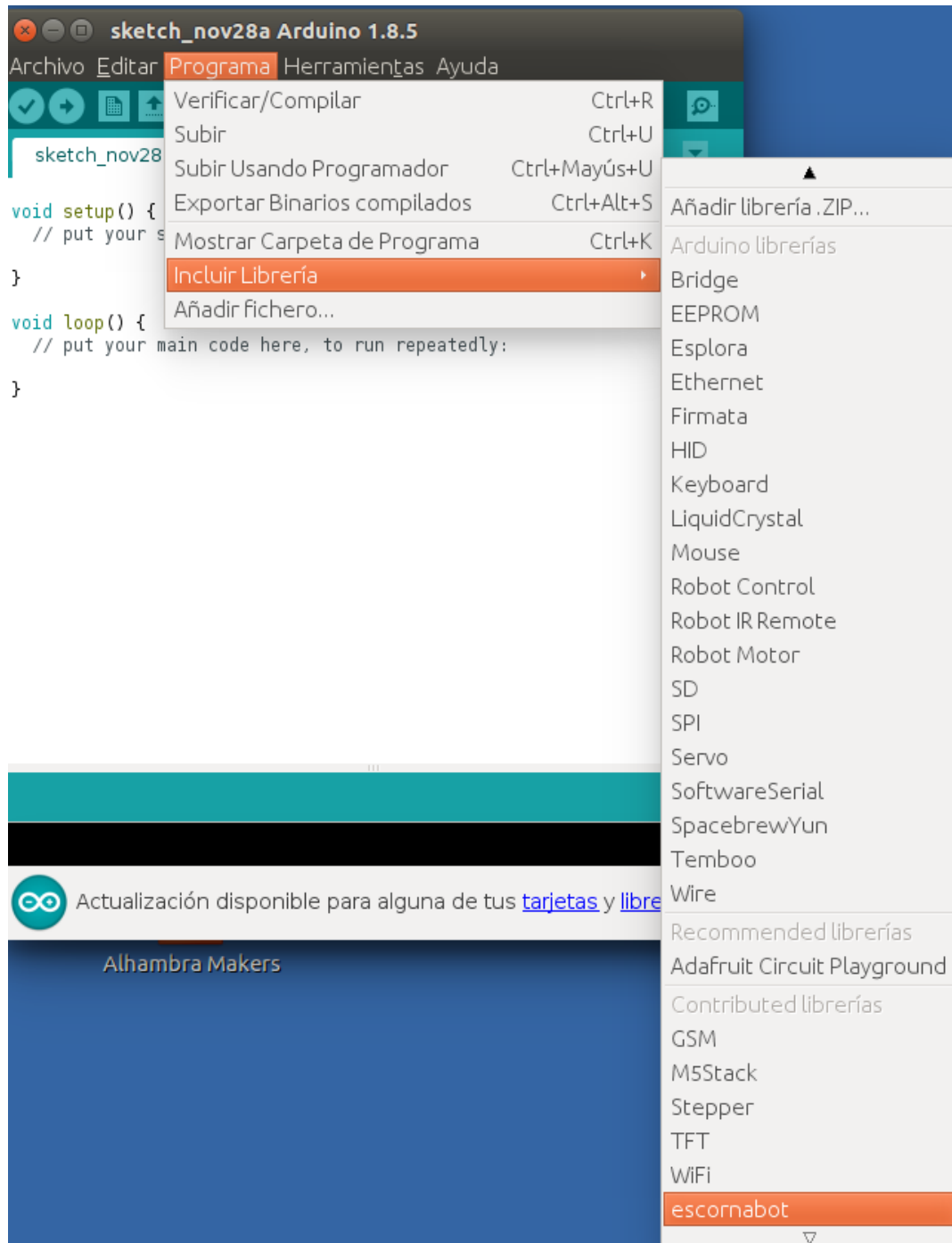
Para cargar la librería en el IDE de Arduino recurrimos al método de incluirla a partir del archivo comprimido zip que previamente hemos descargado.



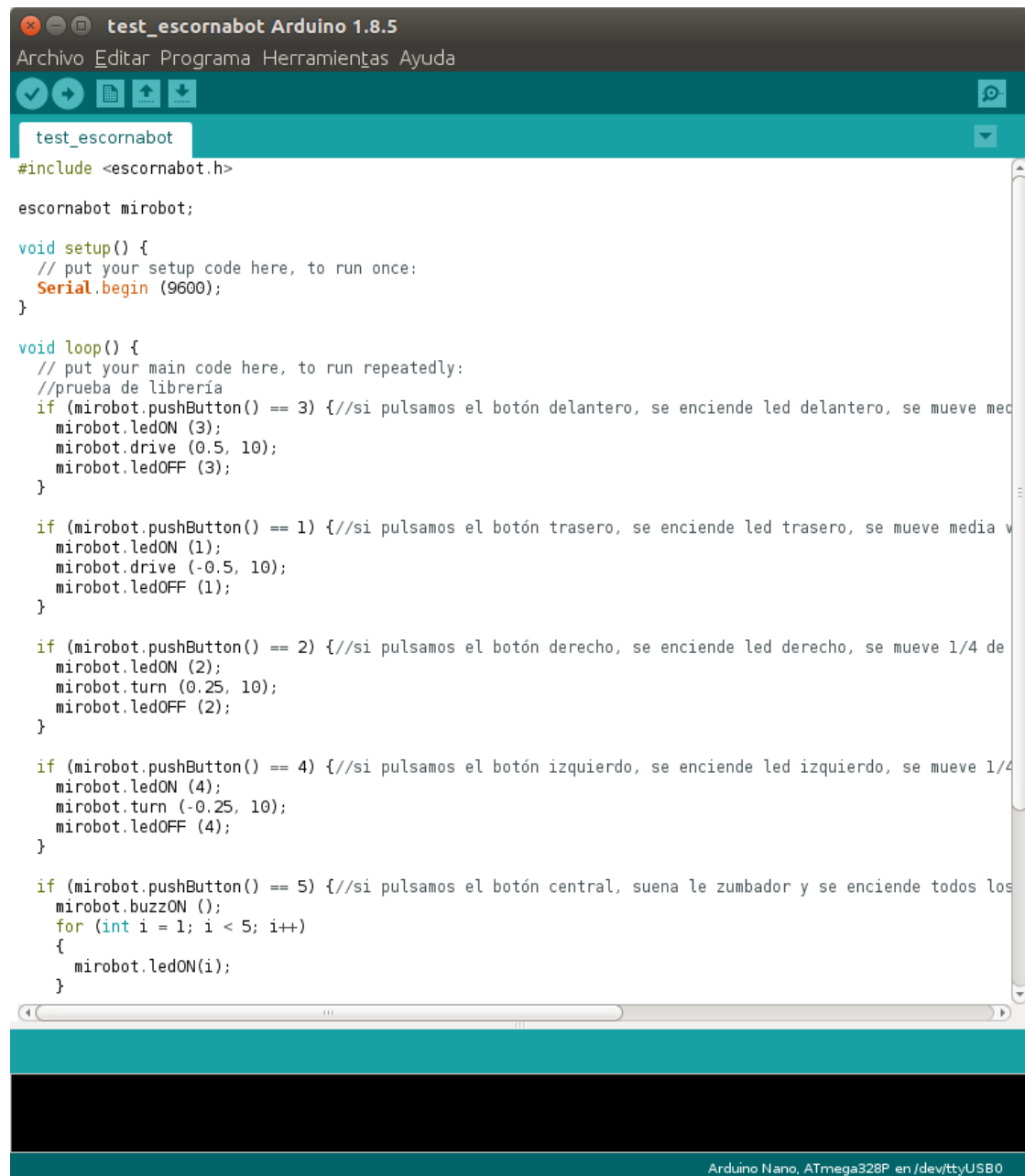
Una vez localizado el archivo clicamos en aceptar.



Ya podemos ver que está disponible a partir de la entrada Incluir Librería del menú Programa.



Para probar el funcionamiento de la librería vamos a usar el ejemplo de test desarrollado por los autores de la librería.



```
test_escornabot Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

test_escornabot
#include <escornabot.h>

escornabot mirobot;

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  //prueba de librería
  if (mirobot.pushButton() == 3) { //si pulsamos el botón delantero, se enciende led delantero, se mueve medio
    mirobot.ledON (3);
    mirobot.drive (0.5, 10);
    mirobot.ledOFF (3);
  }

  if (mirobot.pushButton() == 1) { //si pulsamos el botón trasero, se enciende led trasero, se mueve media v
    mirobot.ledON (1);
    mirobot.drive (-0.5, 10);
    mirobot.ledOFF (1);
  }

  if (mirobot.pushButton() == 2) { //si pulsamos el botón derecho, se enciende led derecho, se mueve 1/4 de
    mirobot.ledON (2);
    mirobot.turn (0.25, 10);
    mirobot.ledOFF (2);
  }

  if (mirobot.pushButton() == 4) { //si pulsamos el botón izquierdo, se enciende led izquierdo, se mueve 1/4
    mirobot.ledON (4);
    mirobot.turn (-0.25, 10);
    mirobot.ledOFF (4);
  }

  if (mirobot.pushButton() == 5) { //si pulsamos el botón central, suena le zumbador y se enciende todos los
    mirobot.buzzON ();
    for (int i = 1; i < 5; i++)
    {
      mirobot.ledON(i);
    }
  }
}
```

Arduino Nano, ATmega328P en /dev/ttyUSB0

Básicamente la librería es una clase con su constructor y con los procedimientos definidos siguientes:

drive (vueltas, velocidad): Sirve para avanzar o retroceder. Se mueve el número de vueltas indicado, si son negativas va en el sentido contrario. La velocidad se da rpm

turn (vueltas, velocidad): Sirve para girar. Se indica como antes el número de vueltas o fracción a girar, si son positivas gira en un sentido y negativas en el contrario. La velocidad se da en rpm.

stop (): detiene los dos motores.

ledON (número de led): sirve para encender los leds de escornabot. Los leds son: 3 (ambar, posición delantera), 1 (azul, posición trasera), 2 (rojo, posición derecha), y 4 (verde, posición izquierda).

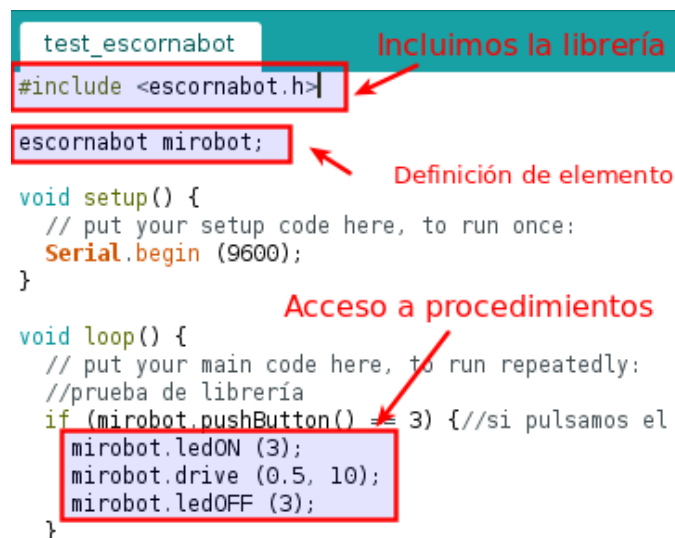
ledOFF (número de led): sirve para apagar los leds de escornabot.

buzzON (): enciende el zumbador.

buzzOFF (): apaga el zumbador.

pushButton(): devuelve el valor del botón pulsado. 3 delantero, 1 trasero, 2 derecha, 4 izquierda, 5 central.

El funcionamiento se muy simple, basta con definir un elemento de la clase y ya tenemos accesibles los procedimientos definidos.



The image shows a screenshot of an Arduino IDE window titled "test_escornabot". The code is as follows:

```
#include <escornabot.h>

escornabot mirobot;

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  //prueba de librería
  if (mirobot.pushButton() == 3) { //si pulsamos el
    mirobot.ledON (3);
    mirobot.drive (0.5, 10);
    mirobot.ledOFF (3);
  }
```

Annotations with red arrows point to specific parts of the code:

- "Incluimos la librería" points to the `#include <escornabot.h>` line.
- "Definición de elemento" points to the `escornabot mirobot;` line.
- "Acceso a procedimientos" points to the `mirobot.ledON (3);`, `mirobot.drive (0.5, 10);`, and `mirobot.ledOFF (3);` lines inside the `loop()` function.

En el mismo ejemplo está descrito su funcionamiento y en el enlace podemos verlo:

<https://youtu.be/58rhLhGofAU>

Y a jugar... con nuestro Escornabot!!