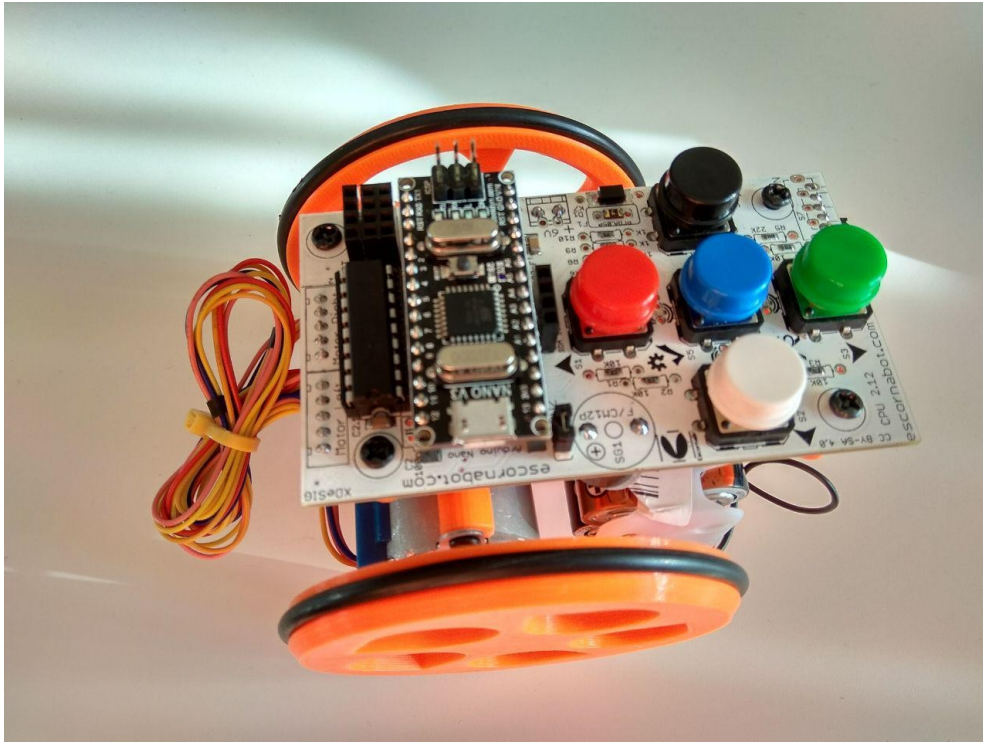


# MANUAL DE USO DE LIBRERÍA DE ARDUINO PARA ESCORNABOT



Documento creado por: *Pedro Ruiz Fernández*

Licencia: Creative Commons by-nc-sa 3.0

# ÍNDICE

- 1 [¿Por qué la librería?](#)
- 2 [¿Cómo incorporar la librería en IDE de Arduino?](#)
- 3 [¿Cómo mover escornabot?](#)
- 4 Luces y sonido.
- 5 Control mediante botonera.
- 6 Control mediante Bluetooth.

## 1. ¿Por qué la librería?

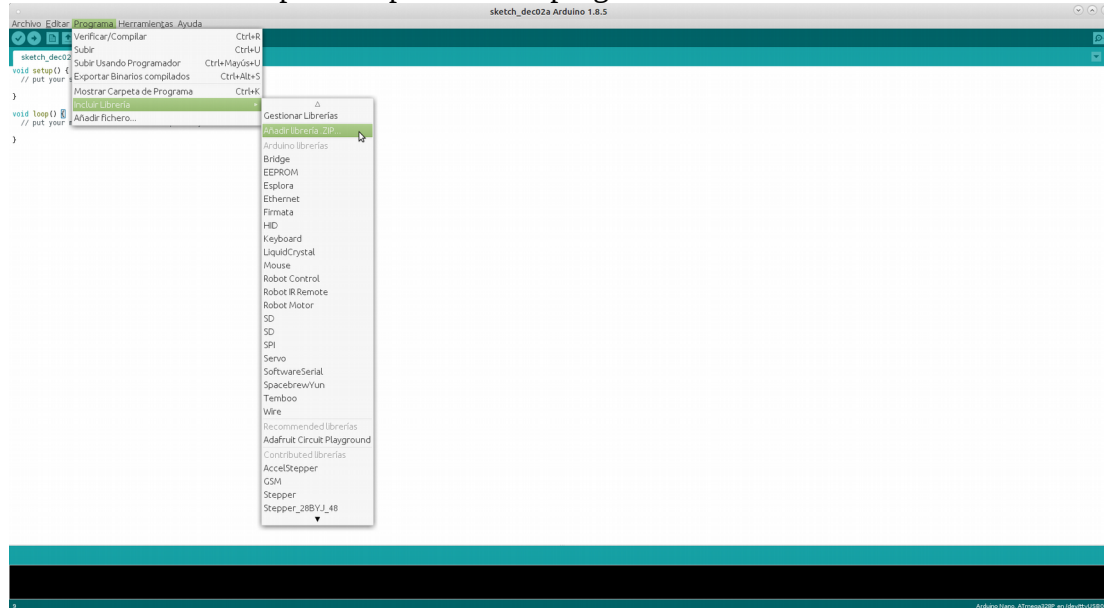
La librería nace en el seno del [Club de Tecnología, programación y robótica de Granada](#), con objeto de cubrir la necesidad de programar escornabot en un entorno de programación textual (IDE Arduino). Escornabot en un principio ha sido diseñado para programarlo exclusivamente con la botonera pensando en el alumnado de primaria, por tanto para alumnado más mayor y con más capacidad de abstracción (tercer ciclo de primaria y ESO) se hacía necesario adaptarle una serie de instrucciones para poder manejar escornabot con programación textual.

La librería es desarrollada por Prudencio Luna (@plunax) y Pedro Ruiz (@pedroruizf).

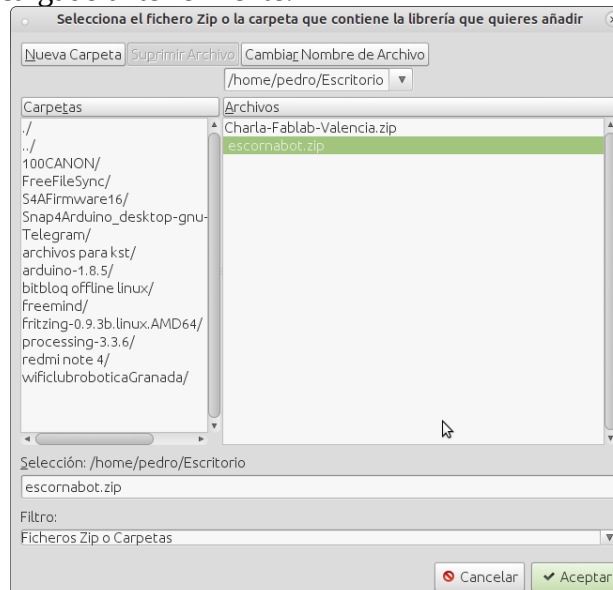
## 2. ¿Cómo incorporar la librería en IDE de Arduino?

Para ello descargamos el archivo “[escornabot.zip](#)” de la librería, y lo incorporamos como librería a nuestro entorno arduino según la siguiente secuencia:

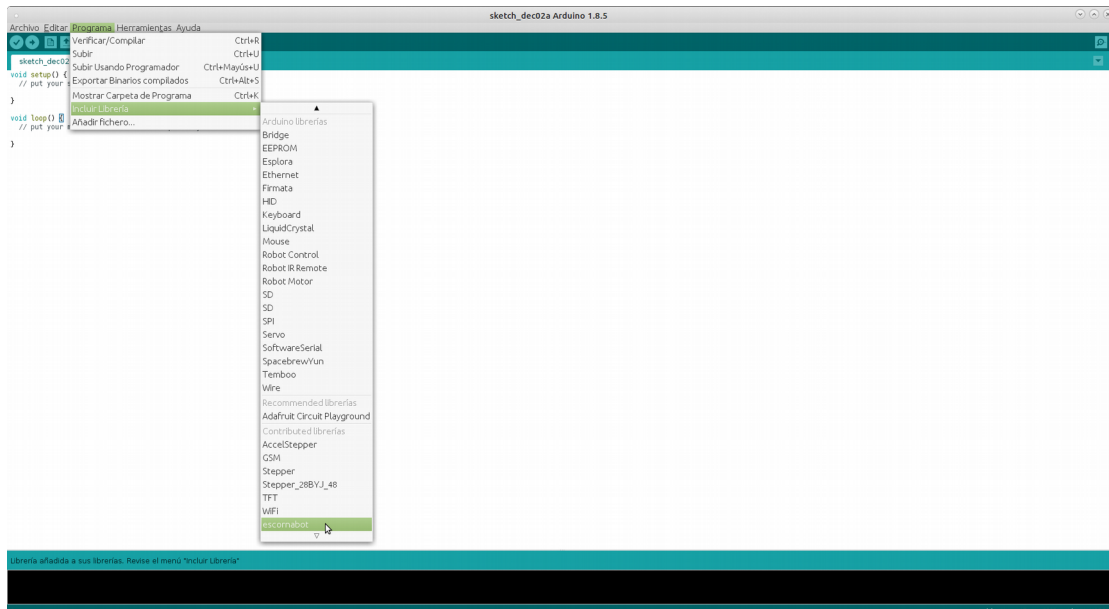
1. Añadimos la librería .zip en la opción Menu programa > Incluir librería > Añadir librería .zip



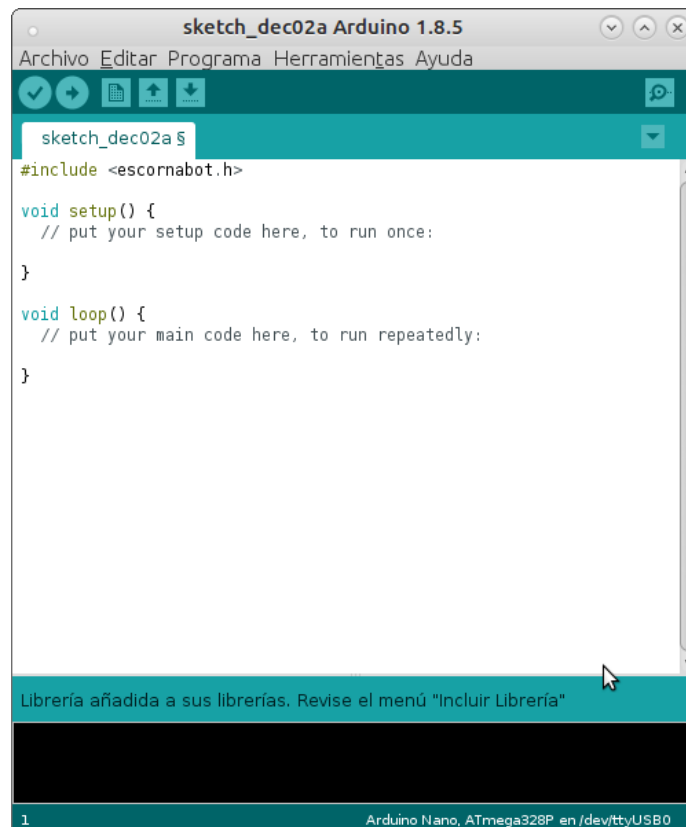
2. Ahora nos aparece un menú de elección del fichero zip, debemos elegir nuestro fichero “escornabot.zip” descargado anteriormente.



3. Una vez incluido ya tenemos la librería disponible en nuestro listado de librerías.



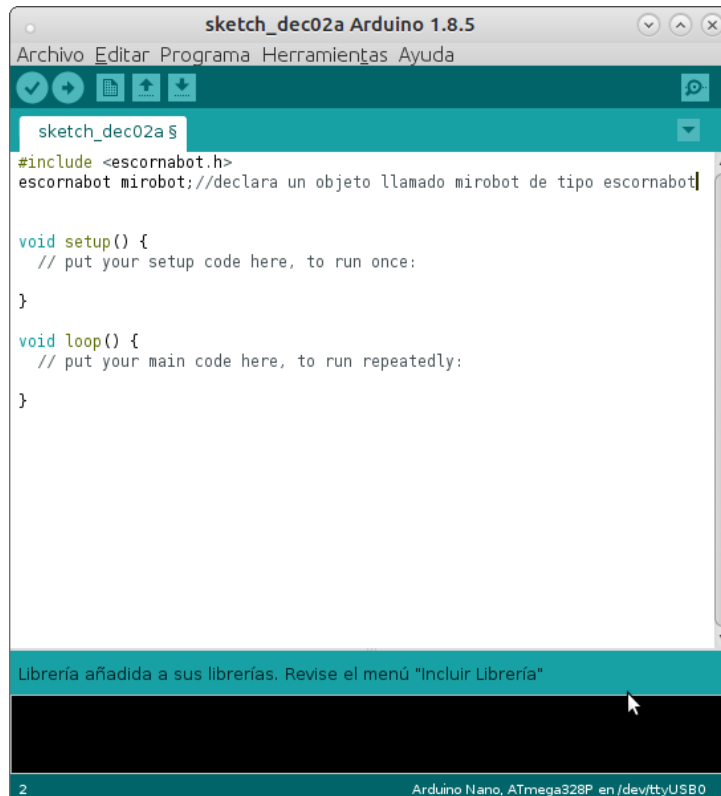
4. Seleccionándola la incluimos en nuestro programa “#include <escornabot.h>”, a partir de ahora ya puedo utilizar las funciones de la misma.



### 3. ¿Cómo mover escornabot?

Para mover escornabot o realizar cualquier otra función con el mismo, debemos en nuestro programa declarar un objeto tipo escornabot, para que el mismo pueda realizar todas las funciones (procedimientos) de un escornabot.

Para ello en nuestro programa en la parte de declaración de variables vamos a incluir dicho objeto, en mi caso se llama “mirobot”.



Ahora a este objeto “mirobot” podemos decirle que haga varias acciones de movimiento. Las posibilidades de movimiento del mismo son:

- **objetoEscornabot.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
- **objetoEscornabot.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.
- **objetoEscornabot.stop ():** detiene los dos motores.

En el siguiente ejemplo (ejemplo 01) hemos programado en el “*setup()*”, para que sólo se ejecute una vez, el siguiente código:

```
/*mueve el robot 1/4 de vuelta hacia delante a una velocidad de 10 rpm*/
mirobot.drive (0.25, 10);
/*mueve el robot 1/4 de vuelta hacia detrás a una velocidad de 10 rpm*/
mirobot.drive (-0.25, 10);
/*gira el robot 1/8 de vuelta hacia delante a una velocidad de 10 rpm*/
mirobot.turn (0.125, 10);
/*gira el robot 1/8 de vuelta hacia delante a una velocidad de 10 rpm*/
mirobot.turn (-0.125, 10);
```



The image shows the Arduino IDE interface with a sketch named 'sketch\_dec02a'. The code defines a 'mirobot' object and includes functions for setup and loop. The setup function moves the robot forward and backward, and turns it. The loop function is currently empty. A status bar at the bottom indicates the board is 'Arduino Nano, ATmega328P' and the port is '/dev/ttyUSB0'.

```
sketch_dec02a $
#include <escornabot.h>
escornabot mirobot;//declara un objeto llamado mirobot de tipo escornabot

void setup() {
  // put your setup code here, to run once:
  /*mueve el robot 1/4 de vuelta hacia delante a una velocidad de 10 rpm*/
  mirobot.drive (0.25, 10);
  /*mueve el robot 1/4 de vuelta hacia detrás a una velocidad de 10 rpm*/
  mirobot.drive (-0.25, 10);
  /*gira el robot 1/8 de vuelta hacia delante a una velocidad de 10 rpm*/
  mirobot.turn (0.125, 10);
  /*gira el robot 1/8 de vuelta hacia delante a una velocidad de 10 rpm*/
  mirobot.turn (-0.125, 10);}

}

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

}
```

Auto Formato terminado.

14 Arduino Nano, ATmega328P en /dev/ttyUSB0