

# Primeras pruebas de programación

Antes de comenzar, ¿**tu ordenador es Mac**? En ese caso puede que experimentes algunos problemas con el driver CH340G de la Arduino Nano, por lo que te sugiero que sigas este [tutorial para adaptar el CH340G a MAC OS, de Alex @TCRobotics de BricoLabs](#).

---

## Librerías

---

Para facilitar la instalación de las librerías, éstas se incluyen comprimidas en la carpeta fw/libraries.

La única librería imprescindible para que el robot funcione es la librería **SoftWire**. Es necesaria para comunicarse por bus I2C al expansor I2C, ya que debido a la cantidad de entradas/salidas necesarias no se ha podido implementar en el bus I2C que Arduino trae por defecto, sino en dos pines digitales cualquiera. Precisamente por la gran cantidad de entradas/salidas necesarias ha sido por lo que se ha decidido incluir un expansor I2C para poder conectar más dispositivos.

La librería SoftWire puede encontrarse en el siguiente enlace:

<http://playground.arduino.cc/Main/SoftwareI2CLibrary>

---

## Consideraciones para la programación

---

Debido a la existencia de algunas diferencias entre los kits de Cyclops-project, cabe mencionarlas para evitar conflictos con la programación:

- Por la falta de pines de entrada/salida disponibles, el Bluetooth comparte puerto serie con el USB de programación de la Arduino, por lo que **con la primera versión de la PCB principal es necesario desconectar el Bluetooth para poder cargar los programas en la Arduino**. De hecho, el Bluetooth está pensado para hacer pruebas en movimiento, corriendo sobre algún circuito, para enviar telemetría y recibir órdenes sin el cable USB.

Si la PCB principal lleva por detrás la inscripción “Main\_board\_V2”, se tratará de la segunda versión, en la cual se ha solucionado ese problema añadiendo un diodo en el diseño. En este caso no hará falta desconectar el Bluetooth para la carga de programas.

- Se aconseja que el **baudrate** declarado para el puerto serie sea siempre el mismo y que coincida con el que se ha programado en el Bluetooth siguiendo su tutorial de configuración. Así se podrá cambiar de USB a Bluetooth y viceversa sin necesidad de cambiar el baudrate.

- Existen dos tipos de expansor I2C que sirven y se han empleado en Cyclops-Project hasta ahora: PCF8574P y PCF8574AP. Funcionan exactamente igual a nivel de hardware, pero para poder comunicarse con ellos, hay que hacer uso de direcciones I2C diferentes:
  - **PCF8574P**: utiliza la dirección hexadecimal **0x20**, teniendo sus tres pines físicos de dirección conectados a masa.
  - **PCF8574AP**: utiliza la dirección hexadecimal **0x38**, teniendo sus tres pines físicos de dirección conectados a masa.

El valor de dirección se puede modificar en la constante **expAddress** que puede verse declarada en el programa de test. Y para saber cuál es el código de tu expansor, basta con mirar la inscripción en el integrado, que estará justo debajo de la Arduino.

- **Sentido de los motores**: es probable que los motores se hayan soldado con los pines cambiados, ya que no hay pautas fijas obligatorias para hacerlo. El resultado es que el motor gira al contrario. En este caso lo que habría que hacer es intercambiar, en el programa de test, los valores de los “defines” DIR\_IZQ\_1 y DIR\_IZQ\_2, para el motor izquierdo, o DIR\_DER\_1 y DIR\_DER\_2, para el motor derecho.
- **Los encoders magnéticos**, al igual que los sensores CNY70 o los pulsadores, actúan como transistores o interruptores que abren y cierran un circuito, pero para que a su salida experimenten una variación de tensión de 0 a 5V **necesitan una resistencia de pull up**. Algunos de los encoders empleados en los kits pueden no llevar esas resistencias. La configuración por defecto en la función de setup es la siguiente:

```
pinMode(ENC_IZQ_A, INPUT);
pinMode(ENC_IZQ_B, INPUT);
pinMode(ENC_DER_A, INPUT);
pinMode(ENC_DER_B, INPUT);
```

Pero si no se aprecia variación en Monitor Serial al cargar el programa de test de los encoders y girar las ruedas, es muy probable que sea porque los encoders no tienen resistencias de pull up y aparezcan todos los valores en cero. Bastaría cambiar el código anterior por este otro para utilizar las resistencias internas de la Arduino:

```
pinMode(ENC_IZQ_A, INPUT_PULLUP);
pinMode(ENC_IZQ_B, INPUT_PULLUP);
pinMode(ENC_DER_A, INPUT_PULLUP);
pinMode(ENC_DER_B, INPUT_PULLUP);
```

---

*Comprobar que todo funciona correctamente*

---

Se ha diseñado un programa de test para comprobar que todas las partes del robot funcionan correctamente. Este programa se encuentra dentro de la carpeta de firmware del proyecto.

Para probar cada uno de los dispositivos, hay que descomentar la línea correspondiente en la cabecera del programa, cargarlo en la Arduino, y abrir Monitor Serial o Putty con el baudrate que se haya especificado en el firmware, para USB o para Bluetooth:

```
#define TEST_LEDS
//#define TEST_PULSADORES
//#define TEST_BATERIA
//#define TEST_SENSORES_LINEA
//#define TEST_SENSORES_DISTANCIA
//#define TEST_MOTORES
//#define TEST_ENCODERS
//#define TEST_CAMARA
```