

SBC-MOTODRIVER3

Placa controladora de motores para motores de corriente continua (DC) y motores paso a paso

Atención! Este manual ha sido traducido automáticamente, en caso de duda, consulte el manual en inglés o póngase en contacto con nuestro servicio de atención a clientes.

Todas las solicitudes de soporte deben ser en alemán o en inglés.

ÍNDICE

1. Información general
2. Cambio de la dirección I²C
3. Asignación de pines
4. Uso con Raspberry Pi
 - 4.1 Conexión
 - 4.2 Instalación
 - 4.3 Uso de motores paso a paso
5. Uso con Raspberry Pi Pico
 - 5.1 Conexión
 - 5.2 Instalación
 - 5.3 Ejemplo de código
 - 5.4 Uso de motores paso a paso
6. Uso con ARD-ONE-C (Compatible con Arduino)
 - 6.1 Conexión
 - 6.2 Instalación
 - 6.3 Ejemplo de código
 - 6.4 Uso de motores paso a paso
7. Uso con micro:bit
 - 7.1 Conexión
 - 7.2 Instalación
 - 7.3 Ejemplo de código
 - 7.4 Uso de motores paso a paso
 - 7.5 Emparejando el micro:bit
8. Información adicional
9. Ayuda

1. INFORMACIÓN GENERAL

Estimado cliente, Gracias por elegir nuestro producto. A continuación, le mostraremos en qué debe prestar atención durante la puesta en marcha y el uso.

Si encuentra algún problema inesperado durante el uso, no dude en ponerse en contacto con nosotros.

Con el MotoDriver3 tienes la opción de controlar y alimentar hasta cuatro motores de corriente continua o dos motores paso a paso. Los motores conectados se controlan a través del chip PCA9634, que proporciona una interfaz I2C.

2. CAMBIO DE LA DIRECCIÓN I2C

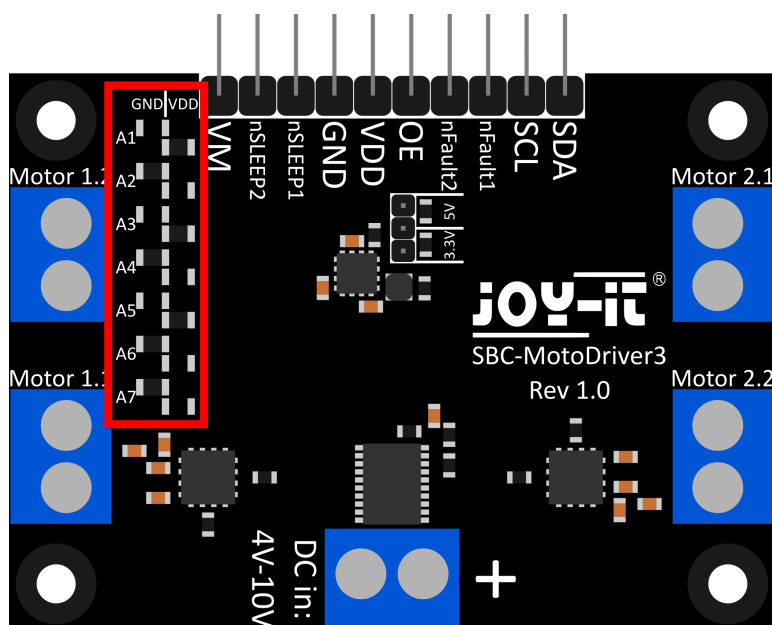
Esta placa controladora de motores se puede configurar con 128 direcciones I2C posibles. Estas pueden ser elegidas libremente, pero en el transcurso posterior se utiliza la dirección predeterminada 0x15.

En la siguiente figura puedes ver qué resistencias se pueden cambiar para cambiar la dirección I2C de la placa del controlador del motor. Puedes volver a soldar estas resistencias de A1 a A7 entre GND y VDD para obtener diferentes direcciones I2C.

Aquí, las posiciones de las siete resistencias se pueden traducir a un valor binario de 7 bits. Aquí tomamos la dirección estándar de la placa del controlador del motor **0x15** que se puede traducir a **001 0101b**.

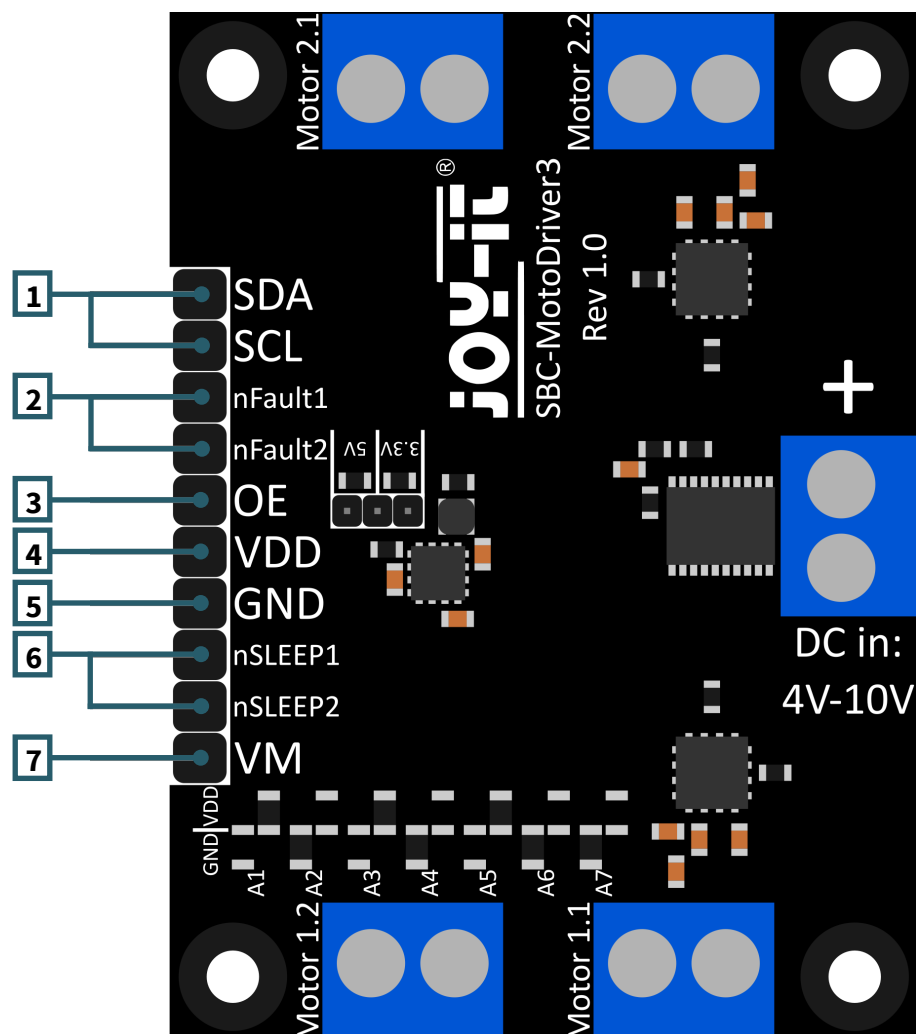
A1 = 1 / A2 = 0 / A3 = 1 / A4 = 0 / A5 = 1 / A6 = 0 / A7 = 0

Cabe señalar que nunca se debe cambiar la dirección I2C durante el funcionamiento, ya que esto puede provocar problemas impredecibles.



3. ASIGNACIÓN DE PINES

En esta sección, explicaremos brevemente las funciones de cada pin:

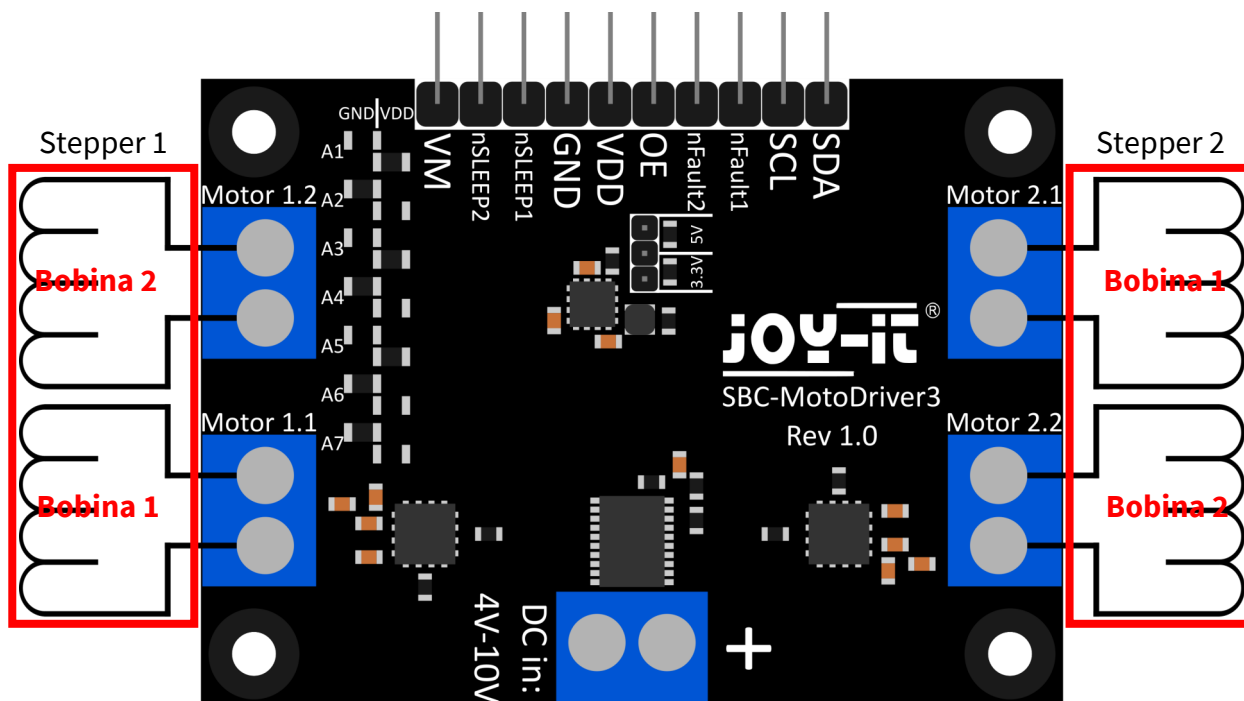


1. **SDA y SDA: Interfaz I2C.** Pines para la comunicación con el controlador IC.
2. **nFault1 y nFault2: Pines de estado de los controladores de motor.** Generan una señal lógica BAJA si los controladores de motor se sobrecalientan o se cargan con demasiada corriente, por ejemplo.
3. **VDD: Salida de voltaje del convertidor de voltaje.** Este pin proporciona la salida de voltaje del convertidor de voltaje integrado en la placa. Aquí puedes elegir entre 2 voltajes mediante un puente (jumper) en la placa. Aquí se pueden seleccionar 5V y 3.3V.
4. **GND: Conexión a tierra.**
5. **nSLEEP1 y nSLEEP2: Habilitan la salida de los controladores de motor.** Cuando estos pines se llevan a una señal lógica ALTA, las salidas de los controladores de motor se habilitan, y cuando los pines se llevan a una señal lógica BAJA, los controladores de motor entran en modo de bajo consumo (sleep mode).
6. **VM: Salida del voltaje del motor.** Este pin proporciona el voltaje desde la entrada de corriente continua (DC in), el cual se utiliza para alimentar los motores, como un voltaje utilizable.

Esta placa de conductor de motor tiene 5 terminales de tornillo.

Cuatro de ellos tienen la serigrafía "Motor 1.1 | Motor 1.2 | Motor 2.1 y Motor 2.2". Estos son los terminales para los motores. Aquí se pueden conectar 2 motores paso a paso o 4 motores de corriente continua. El último terminal es la entrada de tensión para los drivers de los motores y el transformador de tensión.

Es importante que no supere la tensión máxima permitida de la tarjeta de controladores de motor y seleccionar los motores en consecuencia, de lo contrario la tarjeta de controladores de motor podría resultar dañada.



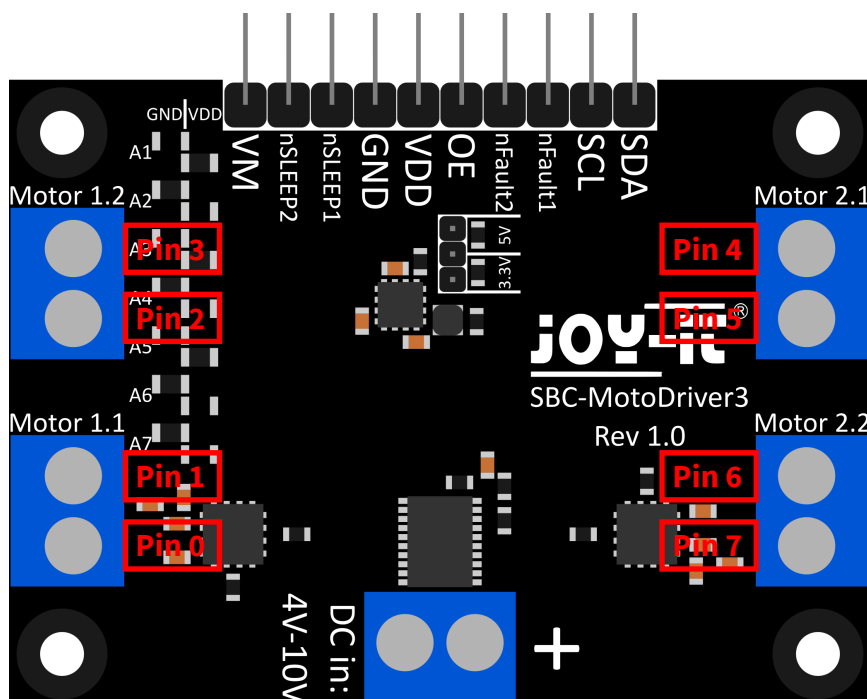
Conecte los cables de las bobinas de sus motores paso a paso a la placa del controlador del motor como se muestra arriba para poder utilizarlos sin problemas.

El terminal "Motor 1.1" tendrá posteriormente los pines 0 y 1 en el software.

El terminal "Motor 1.2" tendrá posteriormente los pines 2 y 3 en el software.

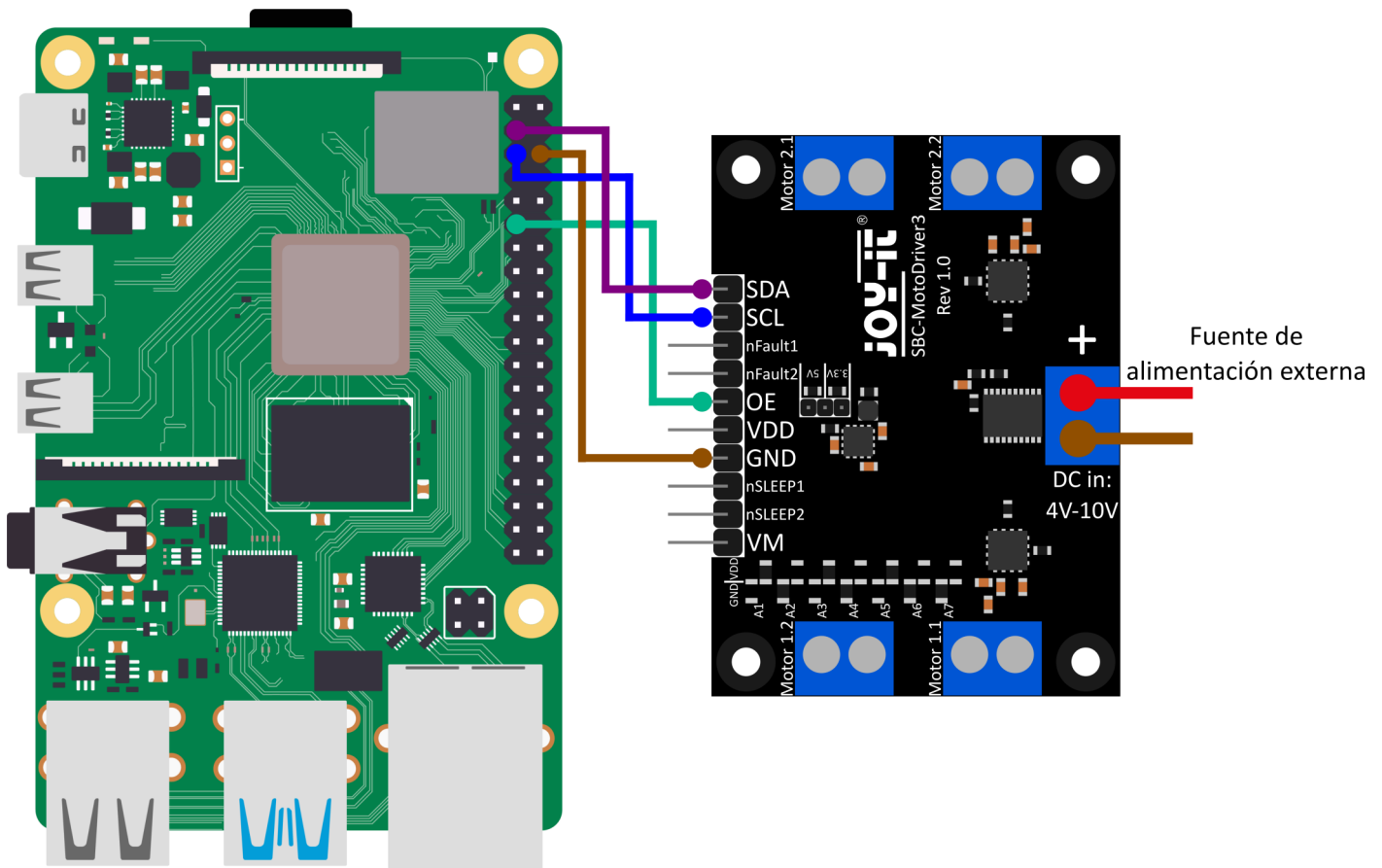
El terminal "Motor 2.1" tendrá posteriormente los pines 4 y 5 en el software.

El terminal "Motor 2.2" tendrá posteriormente los pines 6 y 7 en el software.



4. USO CON RASPBERRY PI

4.1. Conexión



| Raspberry Pi | SBC-MotoDriver3 | Externo |
|------------------|-----------------|---|
| SDA (Pin 3) | SDA | - |
| SCL (Pin 5) | SCL | - |
| - | nFault1 | - |
| - | nFault2 | - |
| GPIO 17 (Pin 11) | OE | - |
| - | VDD | - |
| GND (Pin 6) | GND | - |
| - | nSLEEP1 | - |
| - | nSLEEP2 | - |
| - | VM/DC in | La fuente de voltaje externa debe ajustarse de acuerdo a las especificaciones de los motores. |

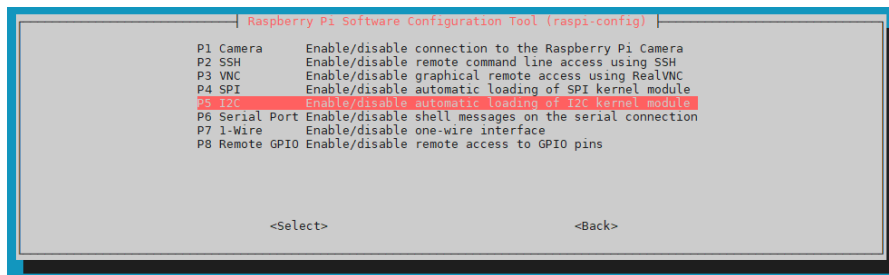
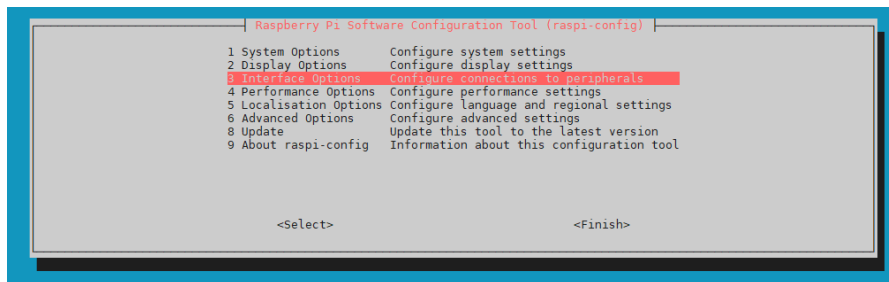
Ten en cuenta que no debes conectar VDD a tu Raspberry Pi, ya que el convertidor de voltaje en la placa solo puede suministrar 1 amperio y la Raspberry Pi puede requerir más de 1 amperio bajo carga. Para evitar dañar los ICs del convertidor de voltaje y, por lo tanto, toda la placa, te recomendamos que no conectes VDD a tu Raspberry Pi.

4.2. Instalación

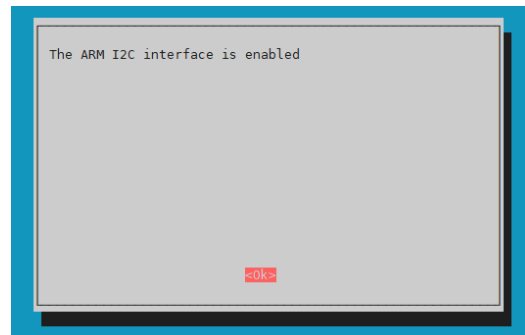
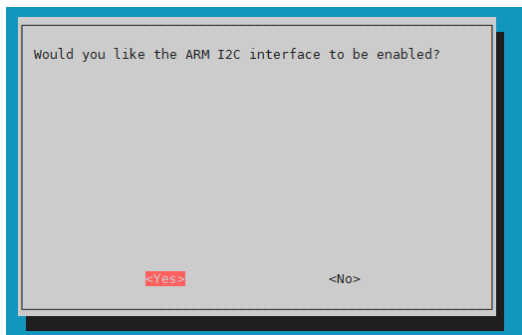
Primero, debes habilitar I2C en tu Raspberry Pi. Para abrir la configuración, ingresa el siguiente comando:

```
sudo raspi-config
```

Allí selecciona **3 Interface Options** → **P5 I2C**.



Allí se te hará la pregunta: **Would you like the ARM I2C Interface to be enabled?** Confirma esto seleccionando **Yes**.



Ahora has habilitado exitosamente I2C. La placa controladora de motores ahora es accesible bajo la dirección I2C **0x15**, que es la configuración predeterminada para esta placa. La dirección I2C será diferente si ya la has configurado antes de configurar tu Raspberry Pi.

Ahora, para encontrar tu placa controladora de motores, debes descargar e instalar `i2c-tools` en tu Raspberry Pi. Puedes hacerlo con el siguiente comando.

```
sudo apt-get install i2c-tools
```

Después de descargar e instalar `i2c-tools`, ahora puedes iniciar el programa a través de...

```
i2cdetect -y 1
```

Para encontrar tu placa controladora de motores, si haces esto, deberías ver las direcciones I2C **0x15** y **0x70**.

Tenga en cuenta que 0x70 es una dirección I2C interna del PCA9634 y no debe utilizarse si no es necesario. Porque esta dirección actúa como reset de hardware del PCA9634.

A continuación, necesitas descargar las bibliotecas adicionales necesarias. Para ello, debes ejecutar los siguientes comandos.

```
sudo apt install -y python3-dev python3-pip
pip3 install smbus2
sudo apt install -y git
```

A continuación necesitas reiniciar tu Raspberry Pi. Para reiniciar tu Raspberry Pi, simplemente ejecuta el siguiente comando.

```
sudo reboot now
```

Para nuestro ejemplo de código, utilizamos la biblioteca [SBC-MotoDriver3-RaspberryPi](#) de [Joy-IT](#), la cual ha sido publicada bajo la [licencia GNU](#) y ya contiene un ejemplo de código. Utiliza el siguiente comando para descargar esta biblioteca desde GitHub. Alternativamente, puedes descargar solo el ejemplo de código [aquí](#).

```
git clone https://github.com/joy-it/SBC-MotoDriver3-RaspberryPi.git
```

Ahora que has descargado la biblioteca, debes ejecutar el siguiente comando para utilizar el ejemplo de la biblioteca del controlador del motor.

```
cd SBC-MotoDriver3-RaspberryPi
python3 SBC-MotoDriver3-Example.py
```

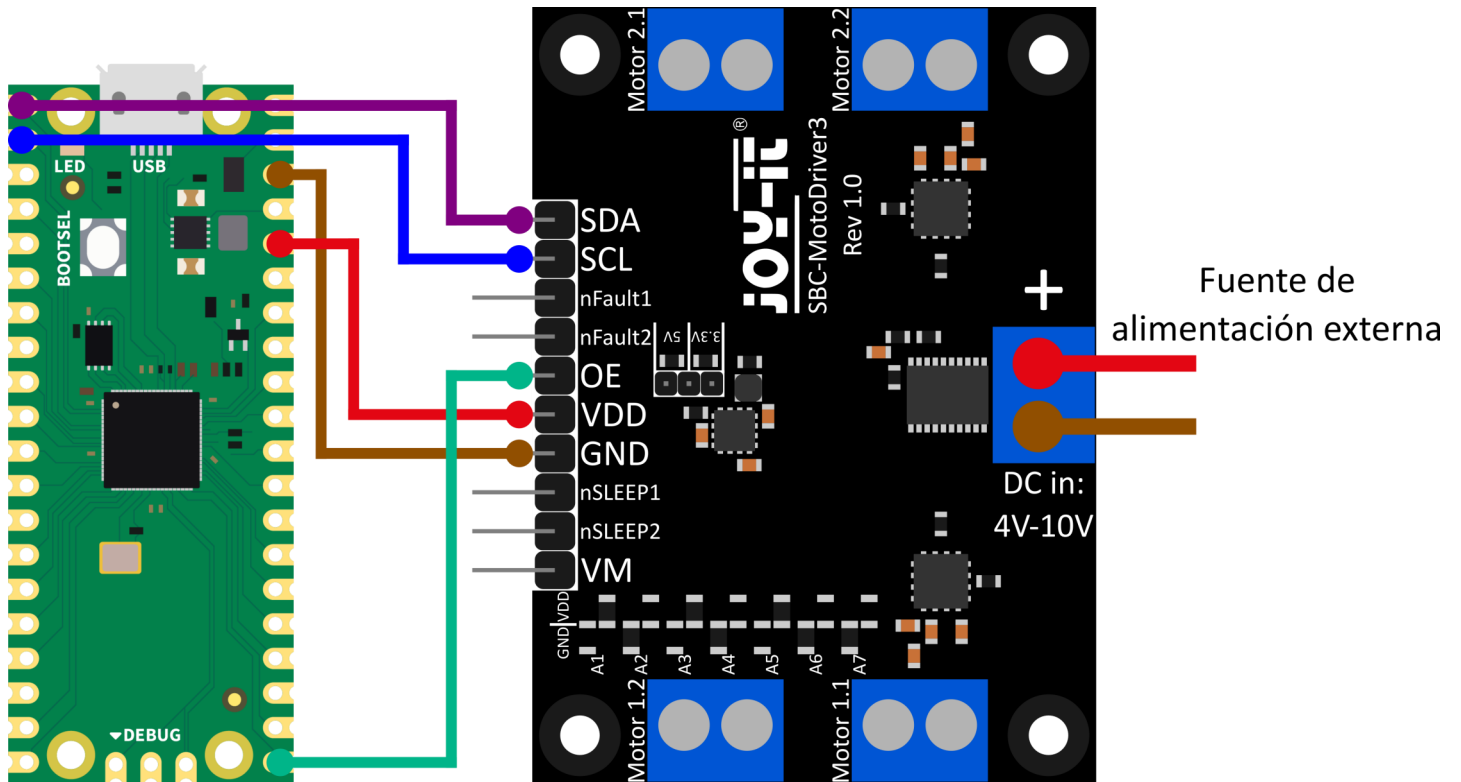
4.3 Utilización de motores paso a paso

Tenga en cuenta que sólo se puede conectar un motor paso a paso por lado. En cuanto al hardware, es importante utilizar los motores paso a paso adecuados para el rango de tensión permitido. Además, independientemente de la plataforma en la que vaya a utilizar el producto, asegúrese de no conectar nunca un motor paso a paso y un motor de corriente continua al mismo lado al mismo tiempo. Esto puede causar problemas o incluso dañar el producto.

En cuanto al software, asegúrese de no enviar la función de motor paso a paso al mismo controlador de motor con las funciones de motor de CC, independientemente de la plataforma en la que tenga previsto utilizar el producto. Mantenga siempre estas funciones separadas entre sí.

5. USO CON LA RASPBERRY PI PICO

5.1. Conexión



| Raspberry Pi Pico | SBC-MotoDriver3 | Externo |
|-------------------|-----------------|---|
| Pin 0 | SDA | - |
| Pin 1 | SCL | - |
| - | nFault1 | - |
| - | nFault2 | - |
| Pin 16 | OE | - |
| 3V3 | VDD | - |
| GND | GND | - |
| - | nSLEEP1 | - |
| - | nSLEEP2 | - |
| - | VM/DC in | La fuente de voltaje externa debe ajustarse de acuerdo a las especificaciones de los motores. |

Ten en cuenta que cuando conectas VDD a tu Raspberry Pi Pico, debes configurar la tensión de salida del convertidor de voltaje en la placa a 3.3V utilizando el puente (jumper) para poder alimentar tu Pico a través de la placa controladora de motores.

5.2. Instalación

Para utilizar esta biblioteca en su Raspberry Pi Pico, le recomendamos que descargue el [IDE Thonny Python](#).

Una vez que haya descargado el IDE Thonny, ahora puede descargar la última versión de [Micropython](#) y flashearla en su Pico.

Una vez descargada la última versión de Micropython, puedes flashearla en tu Pico manteniendo pulsado el botón BOOTSEL cuando conectes tu Pico al PC. Después puedes soltar el botón BOOTSEL. El Pico debería abrirse en el Explorador de tu monitor. Simplemente arrastra el archivo descargado y tu Pico estará flasheado con la última versión de Micropython.

5.3. Ejemplo de código.

A continuación, puedes descargar la biblioteca desde GitHub. Utilizamos la biblioteca [SBC-MotoDriver3-RaspberryPiPico](#) de [Joy-IT](#) para nuestro ejemplo de código, la cual está publicada bajo la [licencia GNU](#).

Después de descargar el archivo *.zip, debes descomprimirlo y seguir estos pasos para utilizar la biblioteca sin problemas:

1. Abre el IDE de Thonny, ve a **"View"** (Vista) y activa **"Files"** (Archivos) allí.
2. Crea una nueva carpeta en tu Pico y dale el nombre **"lib"**.
3. En la ventana de **"Files"** que ahora está abierta, abre la carpeta en la que se encuentra la biblioteca descargada desde GitHub.
4. Ve a la carpeta **"lib"** en tu Pico y haz clic derecho en el archivo **"SBC_MotoDriver3_Lib.py"** en la ventana de **"Files"** y descárgalo en tu carpeta **"lib"** en el Pico.
5. Deja la carpeta **"lib"** en tu Pico y ahora descarga el archivo **"SBC-MotoDriver3-Example.py"** en tu ventana de **"Files"** haciendo clic derecho en tu Pico.

Si has seguido estos pasos, ya puedes utilizar el ejemplo de código.

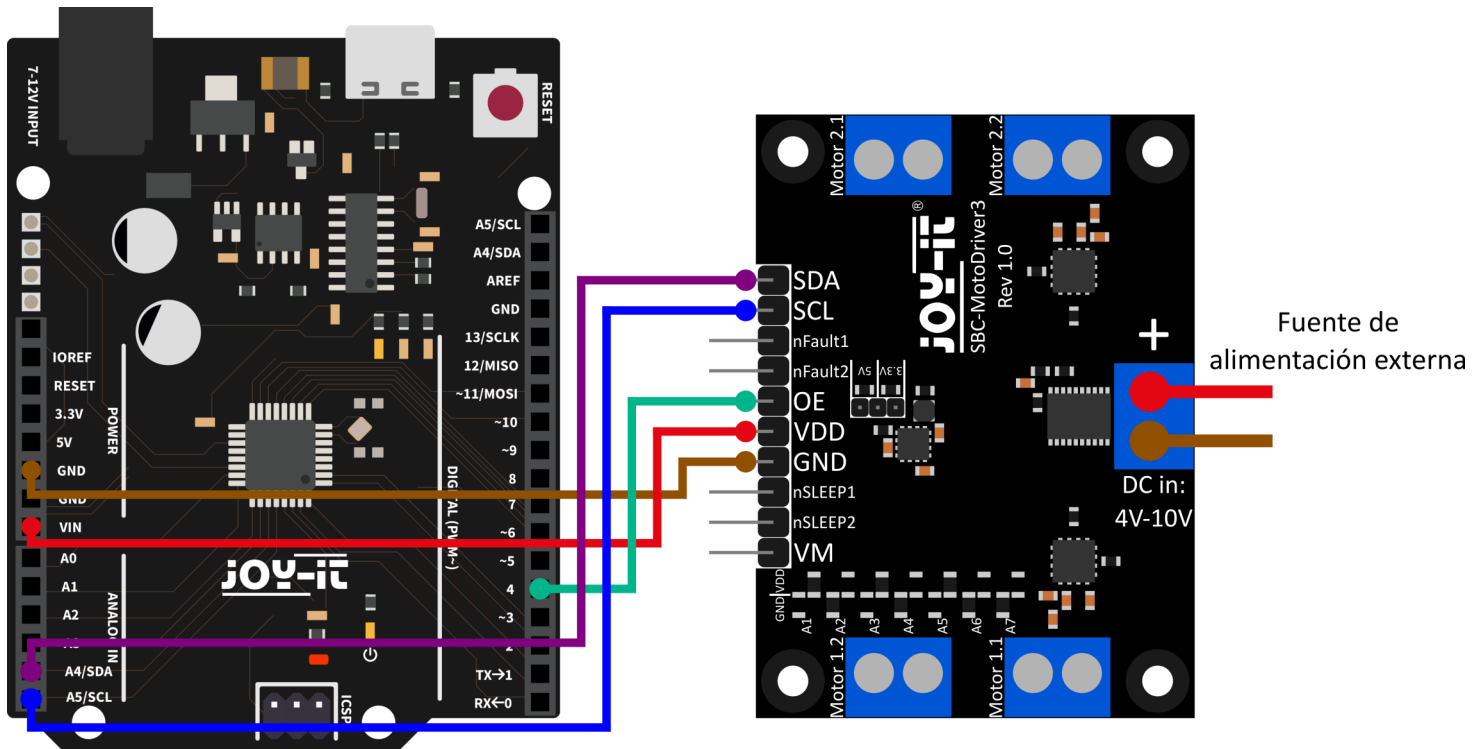
5.4 Utilización de motores paso a paso

Tenga en cuenta que sólo se puede conectar un motor paso a paso por lado. En cuanto al hardware, es importante utilizar los motores paso a paso adecuados para el rango de tensión permitido. Además, independientemente de la plataforma en la que vaya a utilizar el producto, asegúrese de no conectar nunca un motor paso a paso y un motor de corriente continua al mismo lado al mismo tiempo. Esto puede causar problemas o incluso dañar el producto.

En cuanto al software, asegúrese de no enviar la función de motor paso a paso al mismo controlador de motor con las funciones de motor de CC, independientemente de la plataforma en la que tenga previsto utilizar el producto. Mantenga siempre estas funciones separadas entre sí.

6. USO CON ARD-ONE-C (COMPATIBLE CON ARDUINO)

6.1. Conexión



| ARD-ONE-C | SBC-MotoDriver3 | Externo |
|-----------|-----------------|---|
| SDA (A4) | SDA | - |
| SCL (A5) | SCL | - |
| - | nFault1 | - |
| - | nFault2 | - |
| D4 | OE | - |
| VIN | VDD | - |
| GND | GND | - |
| - | nSLEEP1 | - |
| - | nSLEEP2 | - |
| - | VM/DC in | La fuente de voltaje externa debe ajustarse de acuerdo a las especificaciones de los motores. |

Ten en cuenta que si conectas VDD a tu ARD-ONE-C, debes establecer la tensión de salida del convertidor de voltaje en la placa con el jumper en 5V para poder suministrar voltaje a tu ARD-ONE-C a través de la placa controladora de motores.

6.2. Instalación

Para nuestro ejemplo de código, utilizamos la biblioteca [SBC-MotoDriver3-Arduino](#) de [Joy-IT](#), que está publicada bajo la [licencia GNU](#). Para utilizar esta biblioteca, debes descargarla desde GitHub. Puedes instalar esta biblioteca yendo a **Sketch (Boceto) → Include Library (Incluir biblioteca) → Add .zip Library (Agregar biblioteca .zip)**... Busca la biblioteca que acabas de descargar y selecciónala.

6.3 Ejemplo de código

Para probar tu placa controladora de motores con un ejemplo de código, simplemente ve a **Archivo (File) → Ejemplos (Examples) → SBC-MotoDriver3-Arduino** en tu entorno de desarrollo Arduino IDE y luego selecciona **SBC-MotoDriver3-Example...** Haz clic en "Subir" (Upload) para cargar el ejemplo de código en tu placa Arduino. También puedes descargar el ejemplo de código [aquí](#).

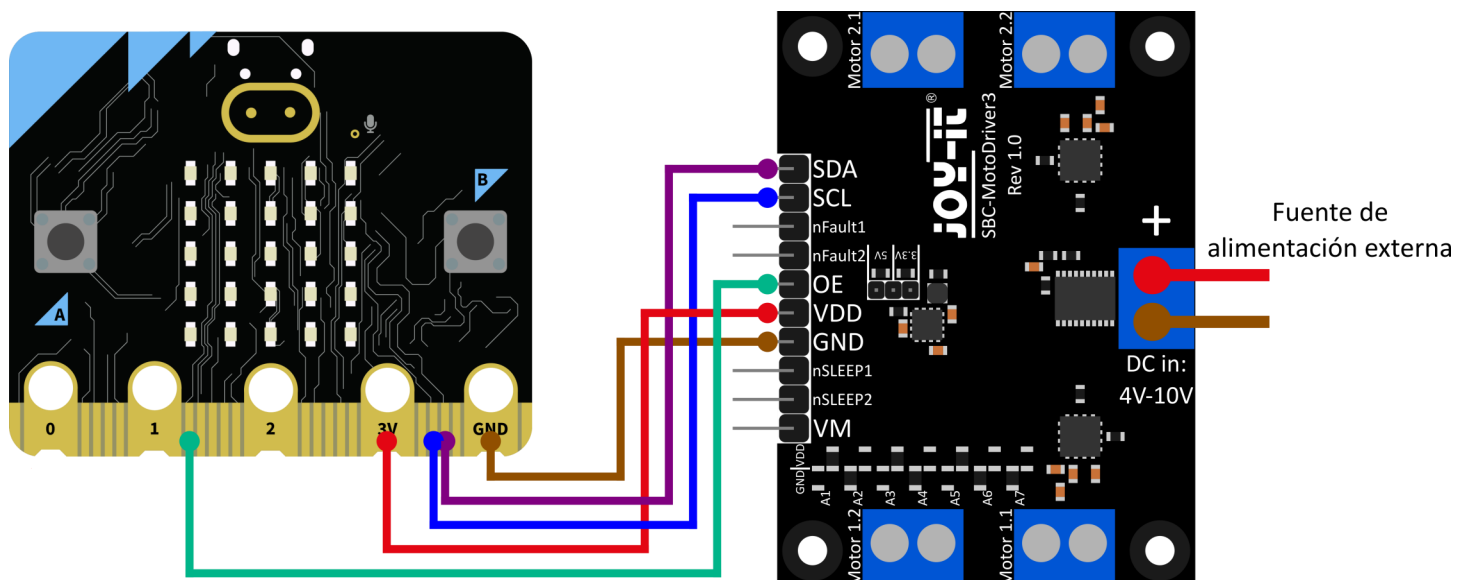
6.4 Utilización de motores paso a paso

Tenga en cuenta que sólo se puede conectar un motor paso a paso por lado. En cuanto al hardware, es importante utilizar los motores paso a paso adecuados para el rango de tensión permitido. Además, independientemente de la plataforma en la que vaya a utilizar el producto, asegúrese de no conectar nunca un motor paso a paso y un motor de corriente continua al mismo lado al mismo tiempo. Esto puede causar problemas o incluso dañar el producto.

En cuanto al software, asegúrese de no enviar la función de motor paso a paso al mismo controlador de motor con las funciones de motor de CC, independientemente de la plataforma en la que tenga previsto utilizar el producto. Mantenga siempre estas funciones separadas entre sí.

7. USO CON EL MICRO:BIT

7.1. CONEXIÓN



| micro:bit | SBC-MotoDriver3 | Extern0 |
|--------------|-----------------|--|
| SDA (Pin 20) | SDA | - |
| SCL (Pin 19) | SCL | - |
| - | nFault1 | - |
| - | nFault2 | - |
| Pin 8 | OE | - |
| 3V | VDD | - |
| GND | GND | - |
| - | nSLEEP1 | - |
| - | nSLEEP2 | - |
| - | VM/DC in | La fuente de voltaje externa debe ser ajustada de acuerdo a las especificaciones de los motores. |

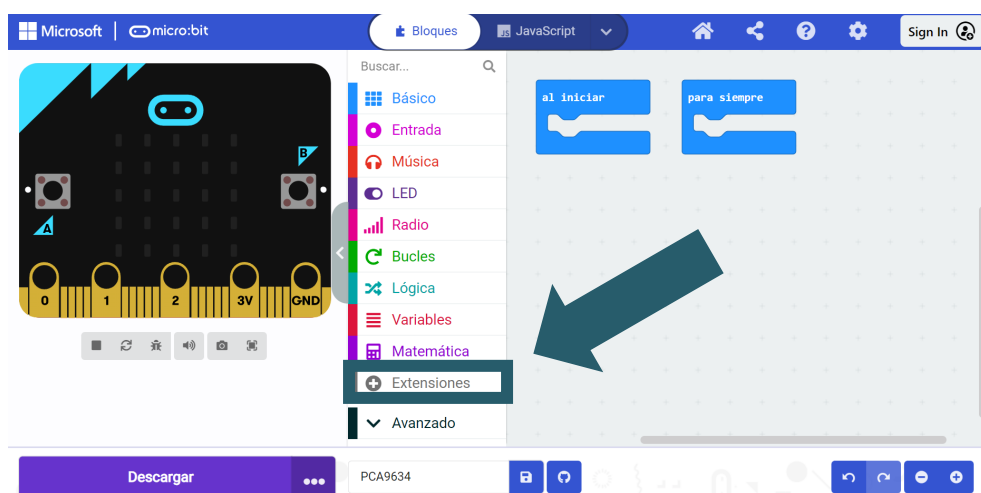
Ten en cuenta que si conectas VDD a tu micro:bit, debes ajustar la tensión de salida del convertidor de voltaje en la placa a 3.3V con el jumper para poder suministrar voltaje a tu micro:bit a través de la placa controladora de motores.

Es posible que necesites utilizar una placa de expansión (breakout board) para conectar cables a algunos pines de tu Micro:Bit. En este ejemplo se utiliza una placa de expansión. Recomendamos [la placa de expansión para micro:bit de Joy-IT](#).

7.2. Instalación

Para el control, recomendamos el uso de la biblioteca [pxt-SBC-MotoDriver3](#), publicada por [Joy-IT](#) bajo la [licencia GNU](#).

Puedes agregar la biblioteca haciendo clic en "Extensions" en la [página de Makecode](#)



y luego ingresar https://github.com/joy-it/pxt-SBC_MotoDriver3 en la barra de búsqueda. Después de hacer eso, simplemente debes hacer clic en la extensión para agregarla automáticamente a tu proyecto actual.

7.3. Ejemplo de código

Para probar tu placa controladora de motores, puedes descargar un ejemplo de código de nuestra parte [aquí](#) o simplemente reconstruir el código tú mismo como se muestra a continuación.

Si has decidido descargar nuestro ejemplo de código, **después de descomprimir el archivo**, simplemente puedes arrastrar y soltar el contenido en la página web de Makecode y soltarlo para importar el archivo y abrirlo automáticamente.

Ejemplo de código Micro:bit Figura:

```
on start
  Initialize the SBC_MotoDriver3 with Address 21
  Set the enablepin to Pin 8 and On
  Turn off all channels

forever
  serial write line "Normal usage"
  Turn on all Forward channels
  pause (ms) 1000
  Turn off all channels
  pause (ms) 1000
  Turn on all Backward channels
  pause (ms) 1000
  Turn off all channels
  pause (ms) 1000
  Turn on channel 0
  pause (ms) 2000
  Turn off channel 0
  pause (ms) 500
  Fade in channel 0 over 20000 milliseconds to 250
  Fade out channel 0 over 20000 milliseconds to 0
  pause (ms) 1000
  Set channel 0 to pwm value 199
  serial write line Read the status of channel 0
  serial write line Read the pwm status of channel 0
  pause (ms) 2000
  Turn off all channels
  pause (ms) 1000
```

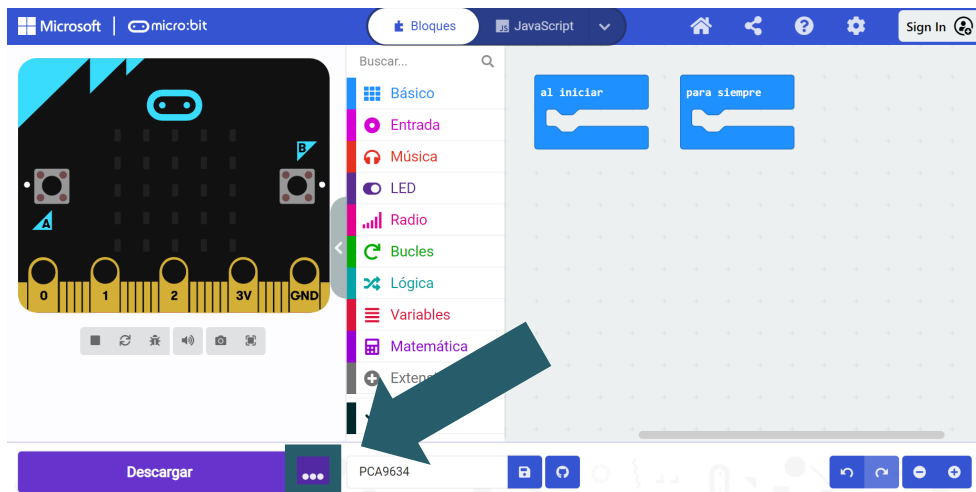
7.4 Utilización de motores paso a paso

Tenga en cuenta que sólo se puede conectar un motor paso a paso por lado. En cuanto al hardware, es importante utilizar los motores paso a paso adecuados para el rango de tensión permitido. Además, independientemente de la plataforma en la que vaya a utilizar el producto, asegúrese de no conectar nunca un motor paso a paso y un motor de corriente continua al mismo lado al mismo tiempo. Esto puede causar problemas o incluso dañar el producto.

En cuanto al software, asegúrese de no enviar la función de motor paso a paso al mismo controlador de motor con las funciones de motor de CC, independientemente de la plataforma en la que tenga previsto utilizar el producto. Mantenga siempre estas funciones separadas entre sí.

7.4. Emparejando el Micro:Bit

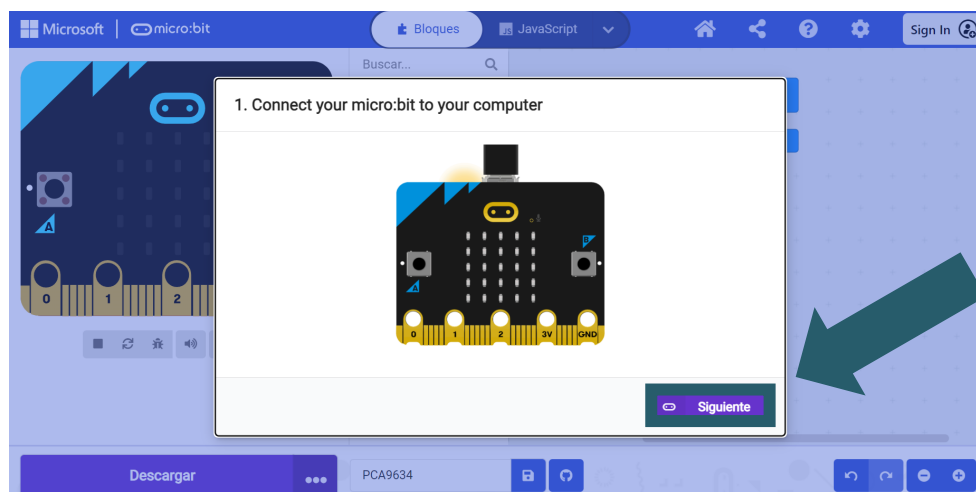
Haz clic en el botón ubicado a la derecha de "Descargar".



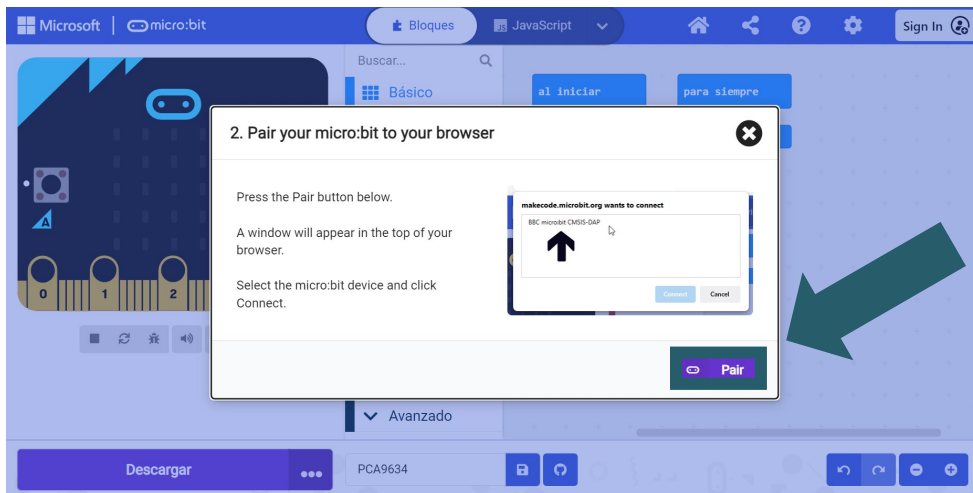
Haz clic en **Connect Device**.



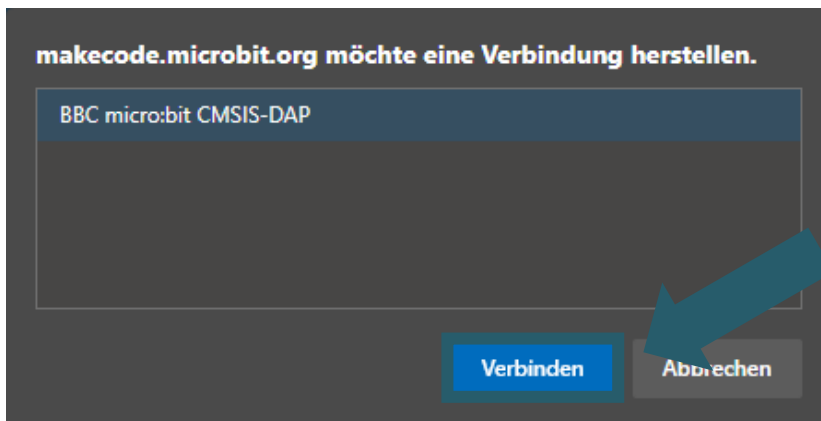
Haz clic en **Siguiente**.



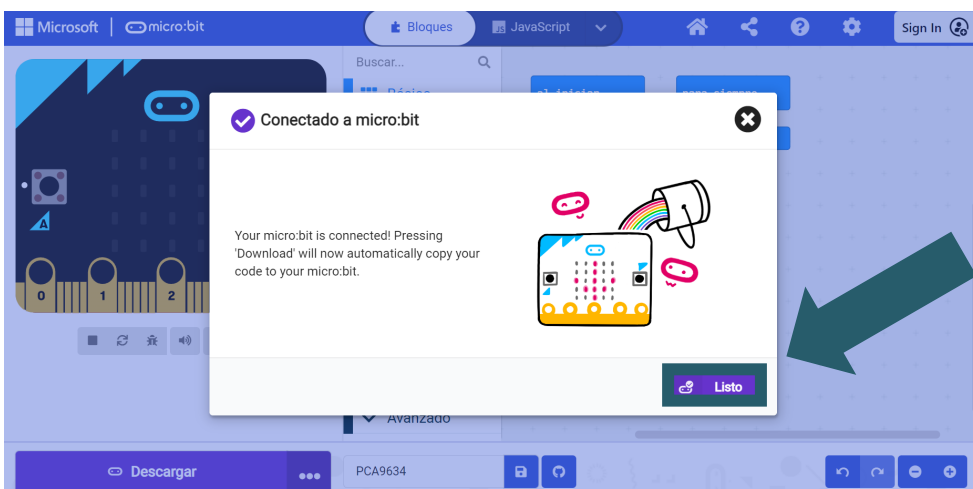
Haz clic en **Pair**.



Selecciona tu Micro:Bit en la ventana abierta y haz clic en "Conectar" para poder conectarlo y utilizarlo.



Lo último que debes hacer es hacer clic en **Listo** para poder cargar tu código. Puedes cargar tu código haciendo clic en "Descargar" (Download).



8. INFORMACIÓN ADICIONAL

Nuestras obligaciones de información y devolución de acuerdo con la Ley de Equipos Eléctricos y Electrónicos (ElektroG)

Símbolo en equipos eléctricos y electrónicos:



Este símbolo de un contenedor de basura tachado significa que los aparatos eléctricos y electrónicos no deben desecharse en la basura doméstica. Debes devolver los aparatos viejos a un punto de recogida designado.

Antes de entregar pilas y acumuladores usados que no estén integrados en un equipo de desecho, deben ser separados de él.

Opciones de devolución:

Como usuario final, puedes devolver tu antiguo dispositivo (que cumple básicamente la misma función que el nuevo dispositivo adquirido con nosotros) de forma gratuita para su disposición cuando compres un nuevo dispositivo.

Los pequeños electrodomésticos con dimensiones externas no superiores a 25 cm se pueden desechar en cantidades normales en el hogar, de forma independiente a la compra de un nuevo electrodoméstico.

Posibilidad de devolución en nuestra ubicación de la empresa durante el horario de atención:

SIMAC Electronics GmbH, Pascalstr. 8, D-47506 Neukirchen-Vluyn, Alemania.

Posibilidad de devolución en su área:

Le enviaremos un sello de paquete con el cual puede devolvernos el dispositivo sin costo alguno. Por favor, contáctenos por correo electrónico a Service@joy-it.net o por teléfono.

Información sobre el embalaje:

Si no tiene material de embalaje adecuado o no desea utilizar el suyo propio, por favor contáctenos y le enviaremos un embalaje adecuado.

9. SUPPORT

Si aún hay problemas pendientes o surgen problemas después de su compra, le brindaremos soporte por correo electrónico, teléfono y a través de nuestro sistema de soporte de tickets.

Email: service@joy-it.net

Sistema de tickets : <http://support.joy-it.net>

Teléfono: +49 (0)2845 9360-50 (Lun - Dom: 10:00 - 17:00 hrs,
viernes: 10:00 - 14:30)

Para obtener más información, visite nuestro sitio web :

www.joy-it.net