ANALOG TO DIGITAL CONVERTER

COM-KY053ADC

1. GENERAL INFORMATION

Dear Customer,
thank you for choosing our product. In the following, we will show you what to pay attention to during commissioning and use.

If you encounter any unexpected problems during use, please feel free to contact us.

2. CHANGE OF THE I2C ADDRESS

This analog to digital converter (or in short ADC) can be configured to 4 different I2C addresses. These can be chosen freely, but in the further course the standard address 0x48 is used.

In the following table you can see all possible addresses and also how to configure them. The main thing to note here is that the ADDR pin of the ADC is responsible for changing the address.

Von	Zu	Adresse
ADDR Pin	GND Pin	0x48
ADDR Pin	VDD Pin	0x49
ADDR Pin	SDA Pin	0x4A
ADDR Pin	SCL Pin	0x4B

The state of the ADDR pin is continuously sampled. Be sure to connect the ADDR pin to the other pins while ensuring the possibility of further connection.

3. PIN ASSIGNMENT

In this section we briefly explain the functions of each pin.

VDD & GND: Power supply

SCL & SDA: I2C interface

ADDR: Used to select the I2C slave address.

ALRT: Used to send a signal when the conversion is ready or when the comparator output is ready.

A0 to A4: Analog inputs.
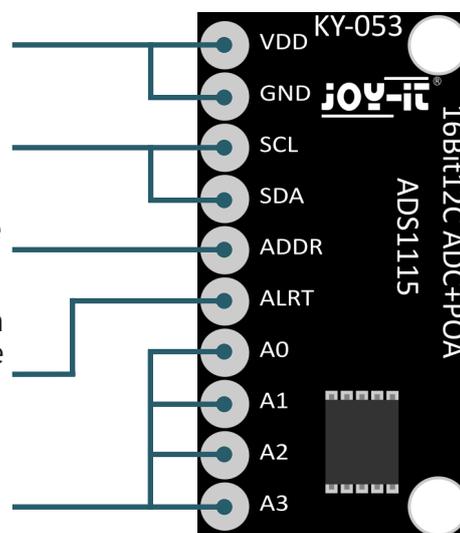
i

This guide was written under Raspberry Pi OS Bookworm for the Raspberry Pi 4 and 5. It has not been checked with newer operating systems or hardware.

4.1 Connection
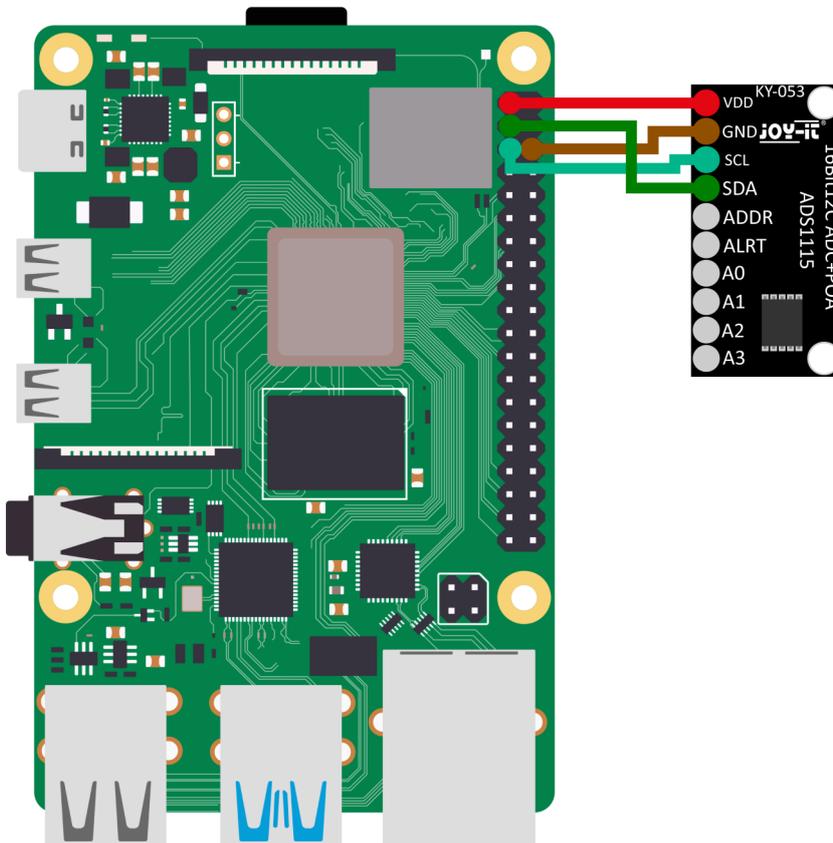
Raspberry Pi	KY-053	External
3,3V	VDD	-
GND	GND	-
SCL (Pin 5)	SCL	-
SDA (Pin 3)	SDA	-
-	ADDR	-
-	ALRT	-
-	A0	Analog signal source to be measured
-	A1	Analog signal source to be measured
-	A2	Analog signal source to be measured
-	A3	Analog signal source to be measured

Here the analog-digital converter is connected to 3.3 V. This means that the converter can only measure 3.3 V on the analog inputs without suffering permanent damage.

The voltage on the analog inputs must not be higher than the supply voltage.

If you want to operate the analog-digital converter with 5 V, you need a **voltage converter** for the data lines, because the Raspberry Pi can be damaged at a logic level of 5 V.

4.2 Installation

First you need to enable I2C on your Raspberry Pi. To open the configuration, enter the following command:

```
sudo raspi-config
```

There select *3 Interface Options → I4 I2C* and enable the I2C interface.

You have now successfully activated I2C. The analog-to-digital converter is now accessible under the I2C address 0x48, which is set by default for this sensor. The I2C address will be different if you have already configured it before configuring your Raspberry Pi.

Now install pip3 with the following command:

```
sudo apt-get install python3-pip
```

The next step is to set up the virtual environment. To do this, enter the following commands:

```
mkdir dein_projekt
```

```
cd dein_projekt
```

```
python -m venv --system-site-packages env
```

```
source env/bin/activate
```

We use the **Adafruit_CircuitPython_ADS1x15** library for our code example, which is released under the **MIT license**. Use the following commands to download and install this library.

```
pip3 install adafruit-circuitpython-ads1x15
```

4.3. code example

Then, after you have downloaded the library, all you need to do is to execute the following command

```
nano COM-KY053ADC-RPi.py
```

to create a new file on your Raspberry Pi and then copy the following code into the file you just created or download it [here](#).

```python
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn

# Create the I2C bus
i2c = busio.I2C(board.SCL, board.SDA)

# Creating the ADC object via the I2C bus
ads = ADS.ADS1115(i2c)

# Create single-ended input on channels
chan0 = AnalogIn(ads, ADS.P0)
chan1 = AnalogIn(ads, ADS.P1)
chan2 = AnalogIn(ads, ADS.P2)
chan3 = AnalogIn(ads, ADS.P3)

while True:
    print("channel 0: ","{:>5}\t{:>5.3f}".format
(chan0.value, chan0.voltage))
    print("channel 1: ","{:>5}\t{:>5.3f}".format
(chan1.value, chan1.voltage))
    print("channel 2: ","{:>5}\t{:>5.3f}".format
(chan2.value, chan2.voltage))
    print("channel 3: ","{:>5}\t{:>5.3f}".format
(chan3.value, chan3.voltage))
    print("-------------------------------------------
-------")
    time.sleep(1)
```

In this code example, the digital value with the corresponding voltage of the individual inputs is displayed. The measurements of an input to which no voltage is applied are faulty and are therefore not representative.

You can then save the code sample with CTRL+O and close it with CTRL+X.

Afterwards you only need to execute the code sample with the following command.

```
python3 COM-KY053ADC-RPi.py
```

5.1 Connection
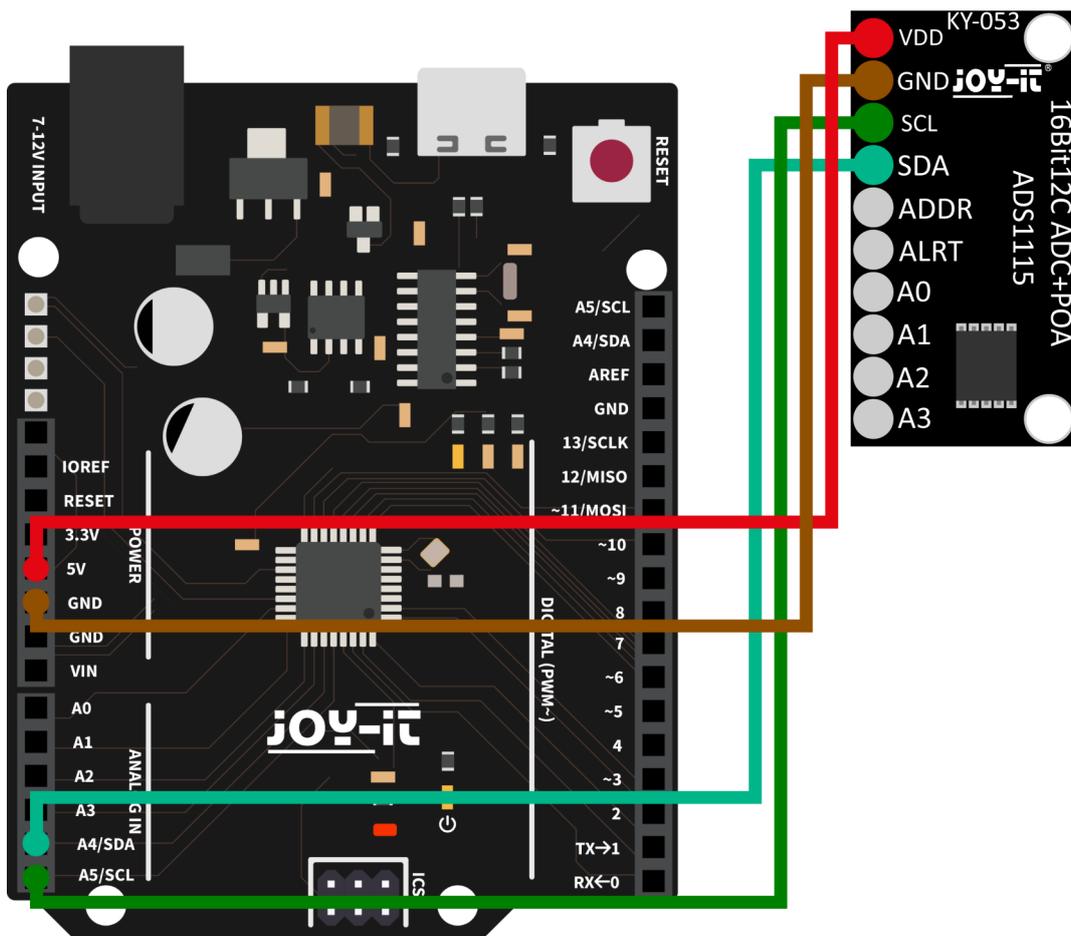
Arduino	KY-053	External
3,3V	VDD	-
GND	GND	-
SCL (A5)	SCL	-
SDA (A4)	SDA	-
-	ADDR	-
-	ALRT	-
-	A0	Analog signal source to be measured
-	A1	Analog signal source to be measured
-	A2	Analog signal source to be measured
-	A3	Analog signal source to be measured

Here the analog-digital converter is connected to 5 V. This means that the converter can only measure 5 V on the analog inputs without suffering permanent damage.

The voltage on the analog inputs must not be higher than the supply voltage.

5.2 Installation

For our code example we use the library **ADS1X15** by **RobTillaart**, which is released under the **MIT license**. You can install this library by searching for **ADS1X15** under **Sketch → Include Library → Manage Libraries...**. Now you just have to click on the **Install** button.

5.3 Code example

To test your ADC with a code example, simply go to **File → Examples → ADS1X15** in your Arduino IDE and then **ADS_read...**. Click **Upload** to upload the code example to your Arduino. Then, to get the output of the sample code displayed, open the **serial monitor** with a **baud rate of 115200**. You can also download the code sample **here**.

In this code example, the digital value with the corresponding voltage of the individual inputs is displayed. The measurements of an input to which no voltage is applied are faulty and are therefore not representative.

6. USE WITH THE MICRO:BIT

6.1 Connection
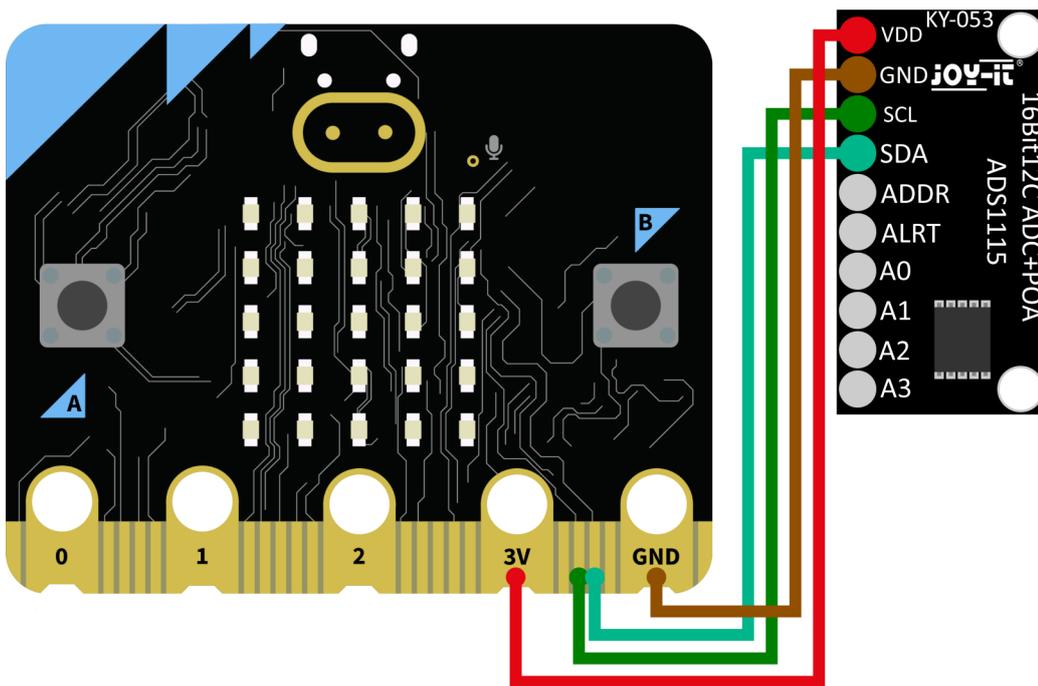
Micro:Bit	KY-053	External
3,3V	VDD	-
GND	GND	-
Pin 19	SCL	-
Pin 20	SDA	-
-	ADDR	-
-	ALRT	-
-	A0	Analog signal source to be measured

		A1	Analog signal source to be measured
-		A2	Analog signal source to be measured
-		A3	Analog signal source to be measured

You may need to use a breakout board to connect cables to some pins of your Micro:Bit. In this example a breakout board is used. We recommend the **breakout board for Micro:Bit from Joy-IT**.

6.2 Installation

For control we recommend the use of the **pxt-ads1115 library**, which was published by **us** under the **MIT license**.

You can add the library by clicking **Extensions** on the **Makecode page** and then typing **ADS1115** in the search bar. After you have done that, you just need to click on the extension to add it automatically for your current project.

6.3 Code example

To test your ADC you can download a code example from us **here** or simply rebuild the code yourself as shown below.

If you have decided to download our code sample, then **after unzipping** the file you can simply drag and drop the contents onto the Makecode web page and release to import the file and open it automatically.

6.4 Pairing the Micro:Bit
Click on the button located to the right of Upload/Download.
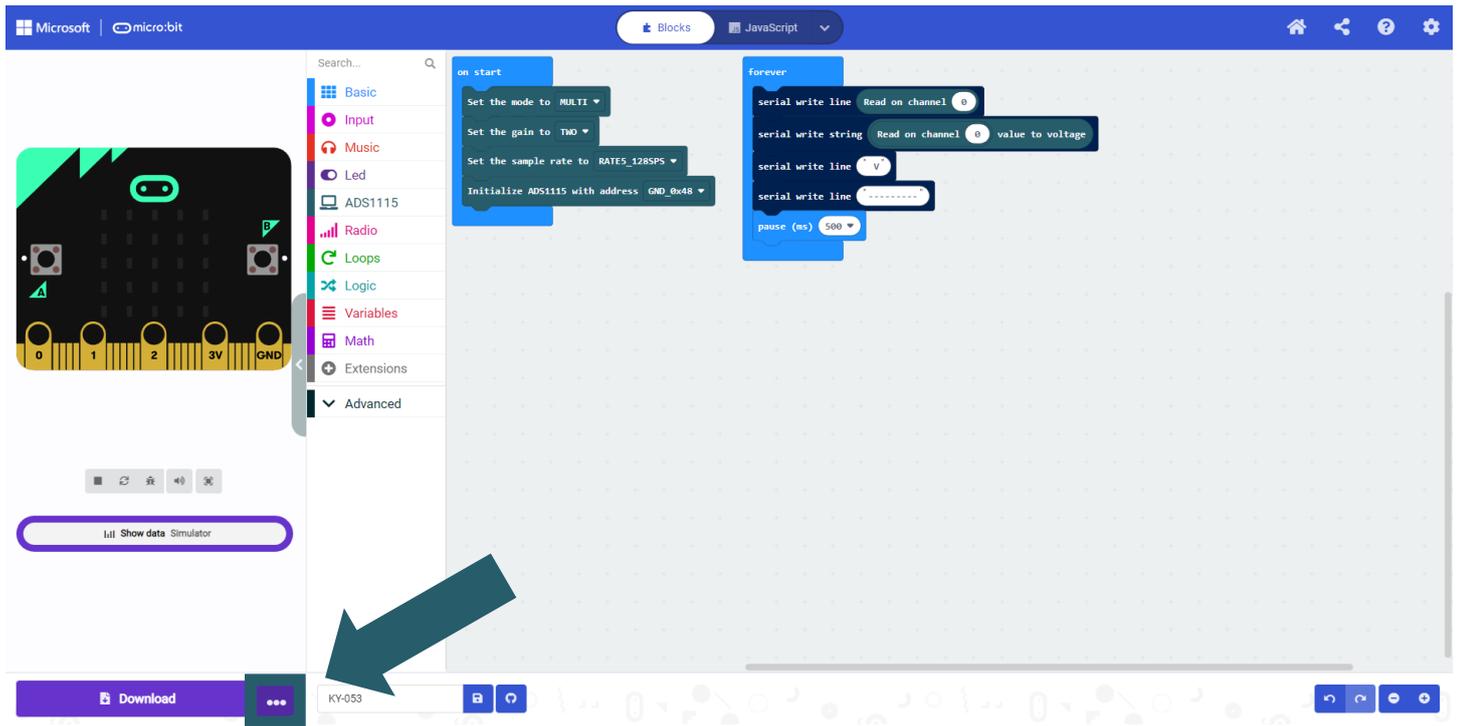
Click **Connect Device**.
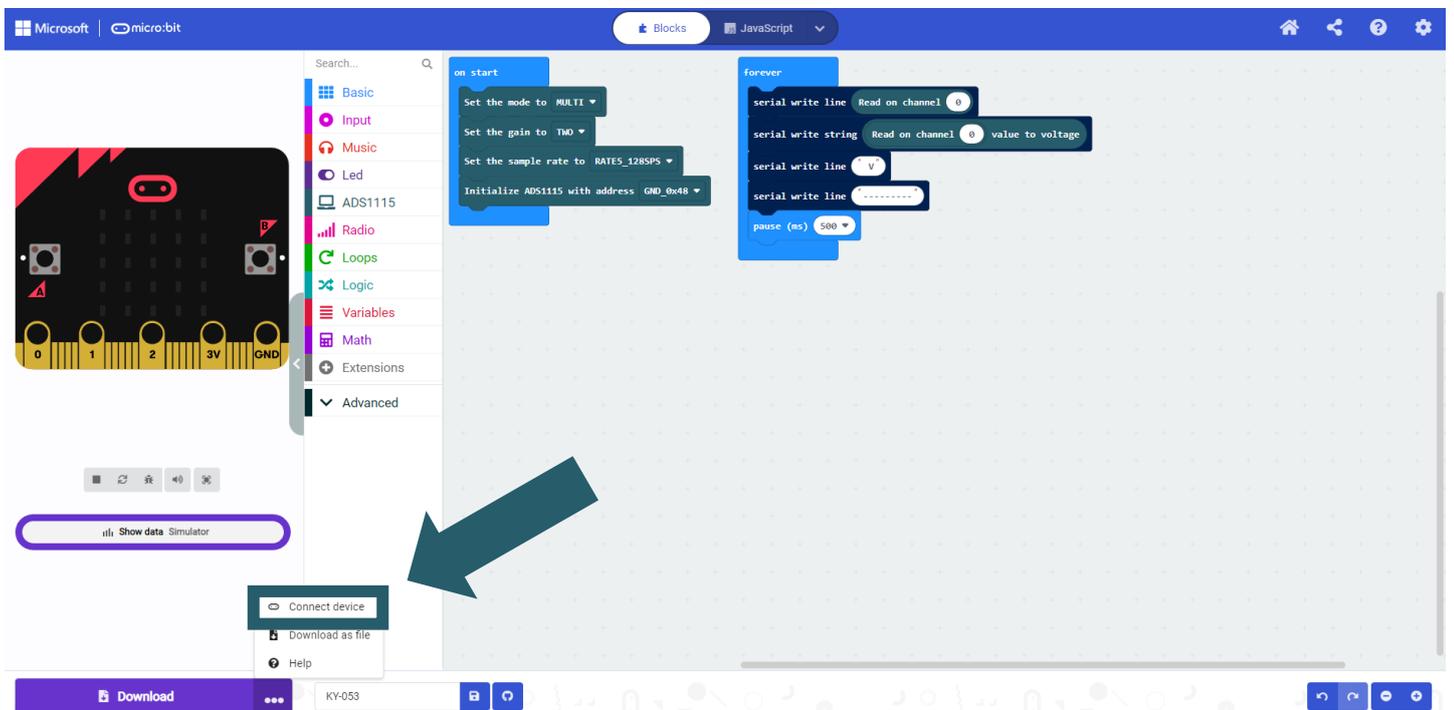
Click **Next.**
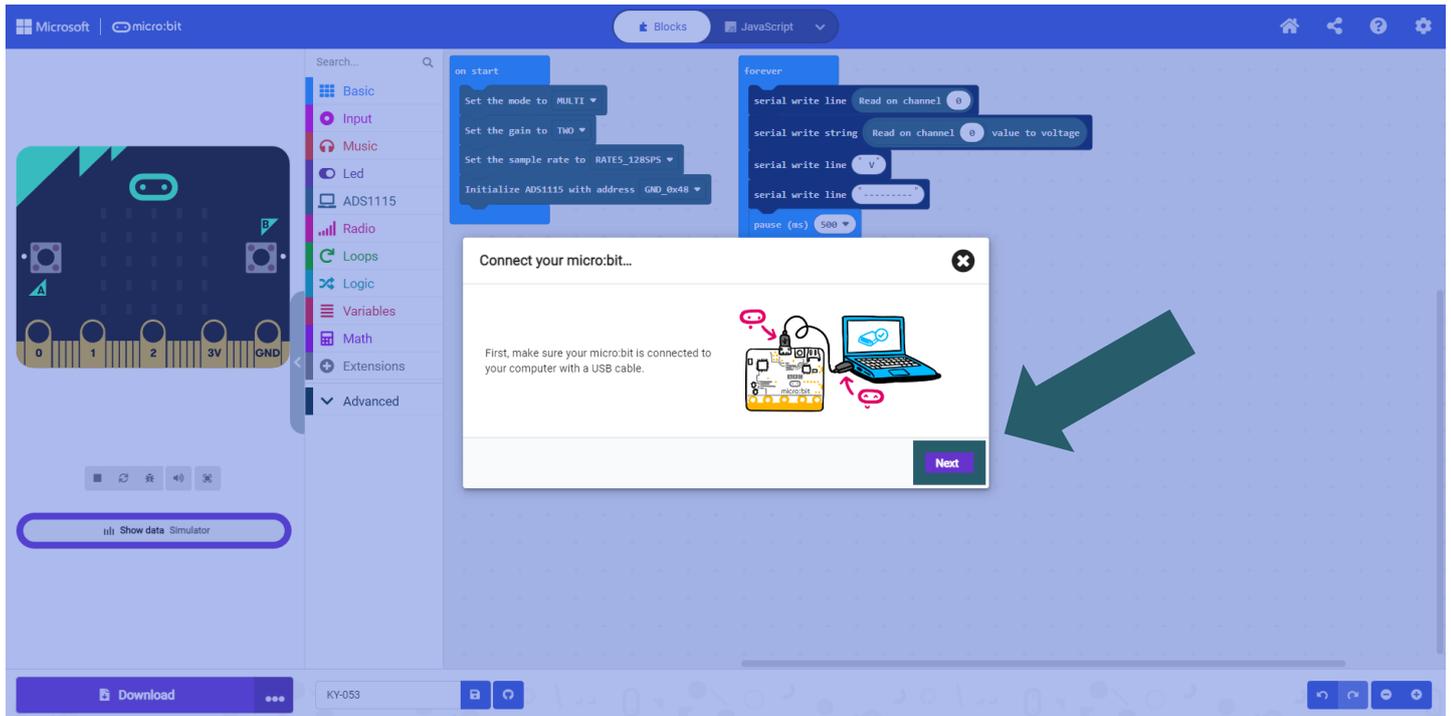
Click **Next.**
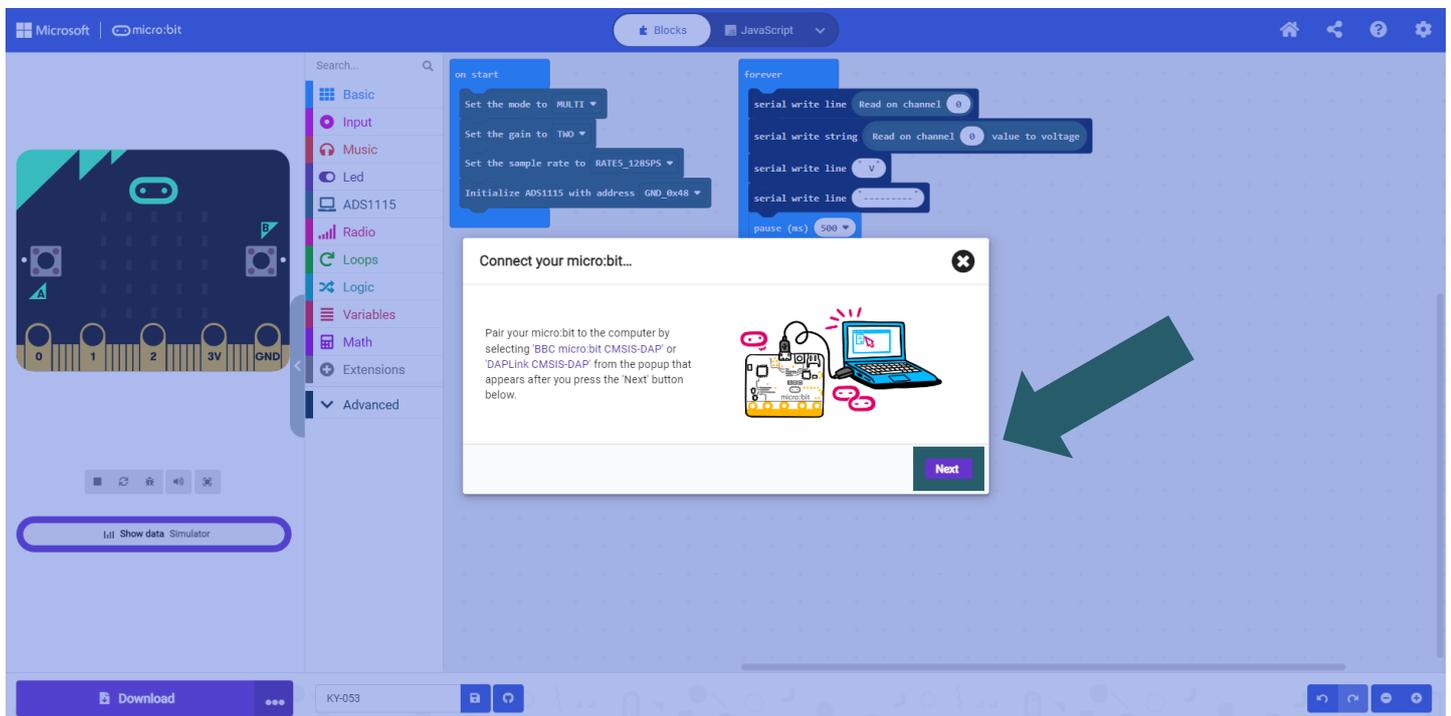
Select your Micro:Bit in the opened window and click **Connect** to connect and use it.

The last thing you need to do is click on **Done** to be able to upload your code. You can upload the code by clicking on **Download**.
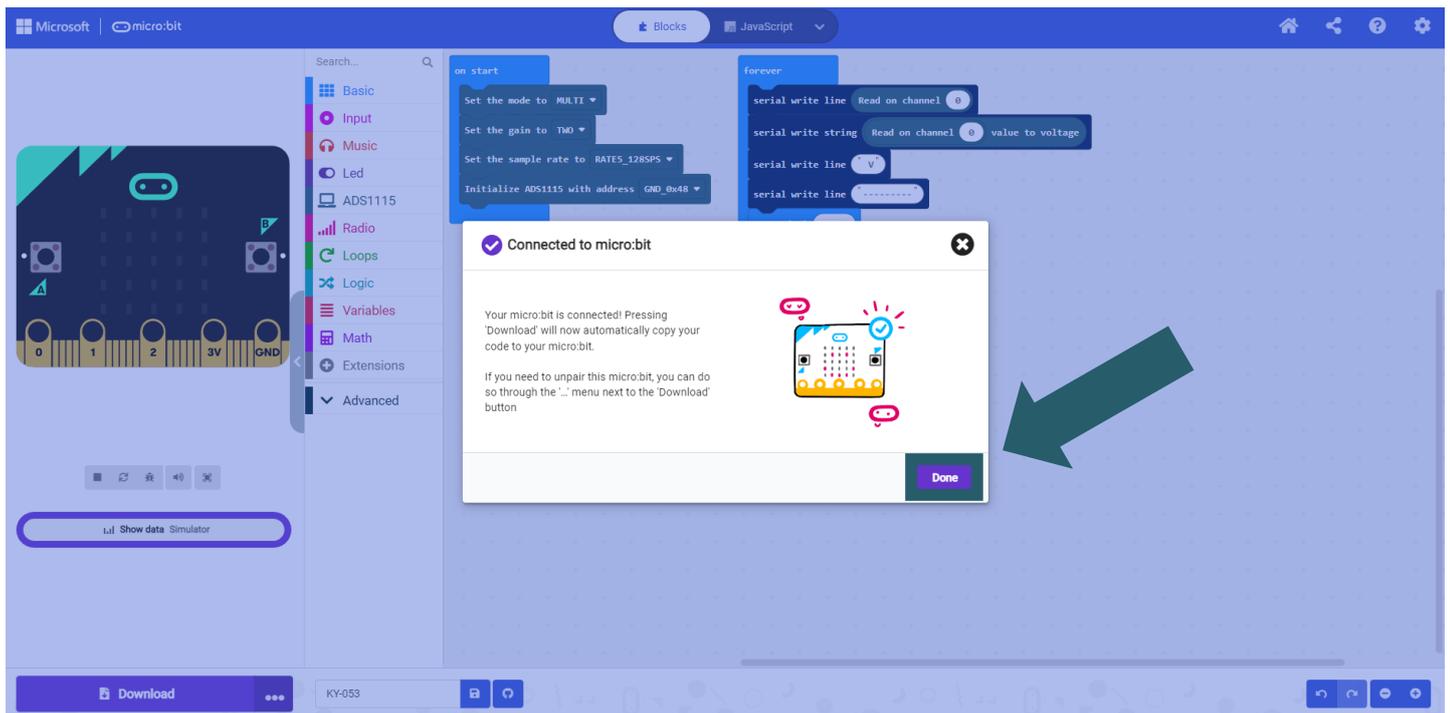
Now that you have paired your Micro:Bit, you can open the serial output and view your results.
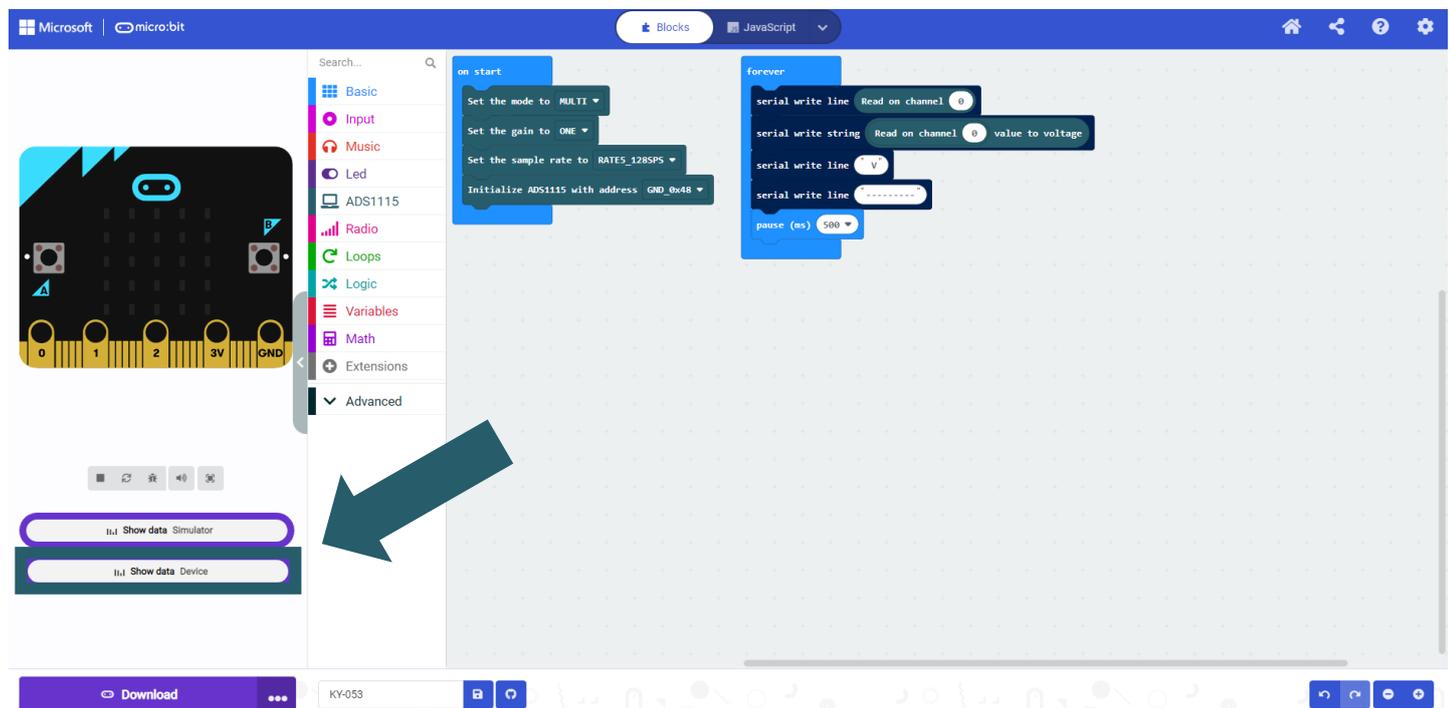
7. OTHER INFORMATION

Our information and take-back obligations according to the Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrical and electronic equipment:

This crossed-out dustbin means that electrical and electronic appliances do not belong in the household waste. You must return the old appliances to a collection point.

Before handing over waste batteries and accumulators that are not enclosed by waste equipment must be separated from it.

Return options:

As an end user, you can return your old device (which essentially fulfills the same function as the new device purchased from us) free of charge for disposal when you purchase a new device.

Small appliances with no external dimensions greater than 25 cm can be disposed of in normal household quantities independently of the purchase of a new appliance.

Possibility of return at our company location during opening hours:

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

Possibility of return in your area:

We will send you a parcel stamp with which you can return the device to us free of charge. Please contact us by email at **Service@joy-it.net** or by telephone.

Information on packaging:

If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging.

7. SUPPORT

If there are still any issues pending or problems arising after your purchase, we will support you by e-mail, telephone and with our ticket support system.

Email: **service@joy-it.net**
Ticket system: **http://support.joy-it.net**
Telephone: +49 (0)2845 9360-50 (Mon - Thur: 09:00 - 17:00 o'clock,
Fri: 09:00 - 14:30 o'clock)

For further information please visit our website:
www.joy-it.net