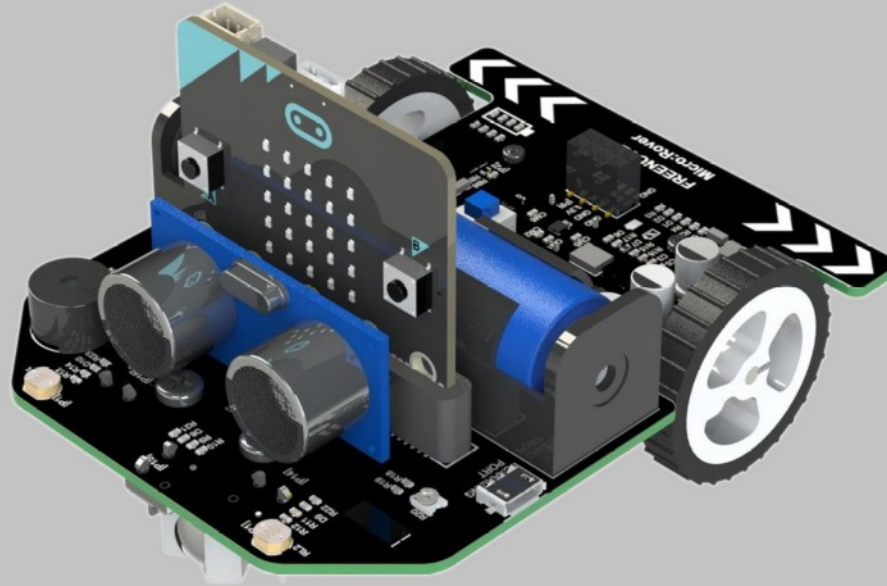
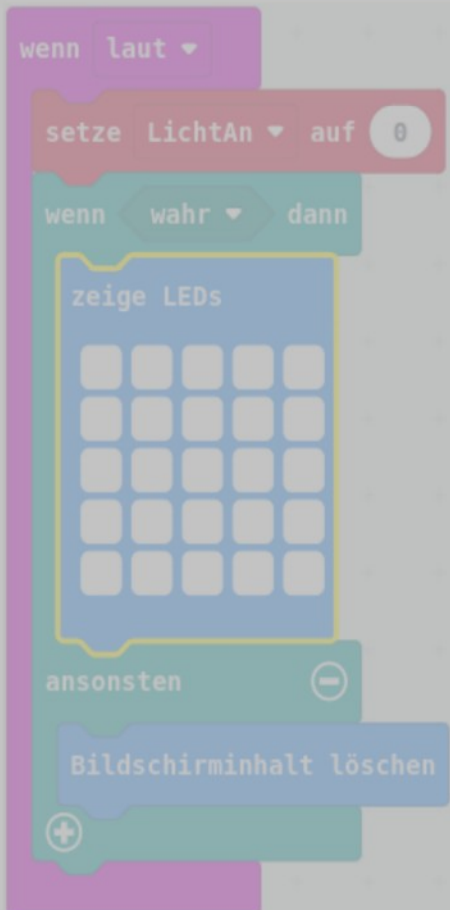


Ein Führerprogramm im Microbit

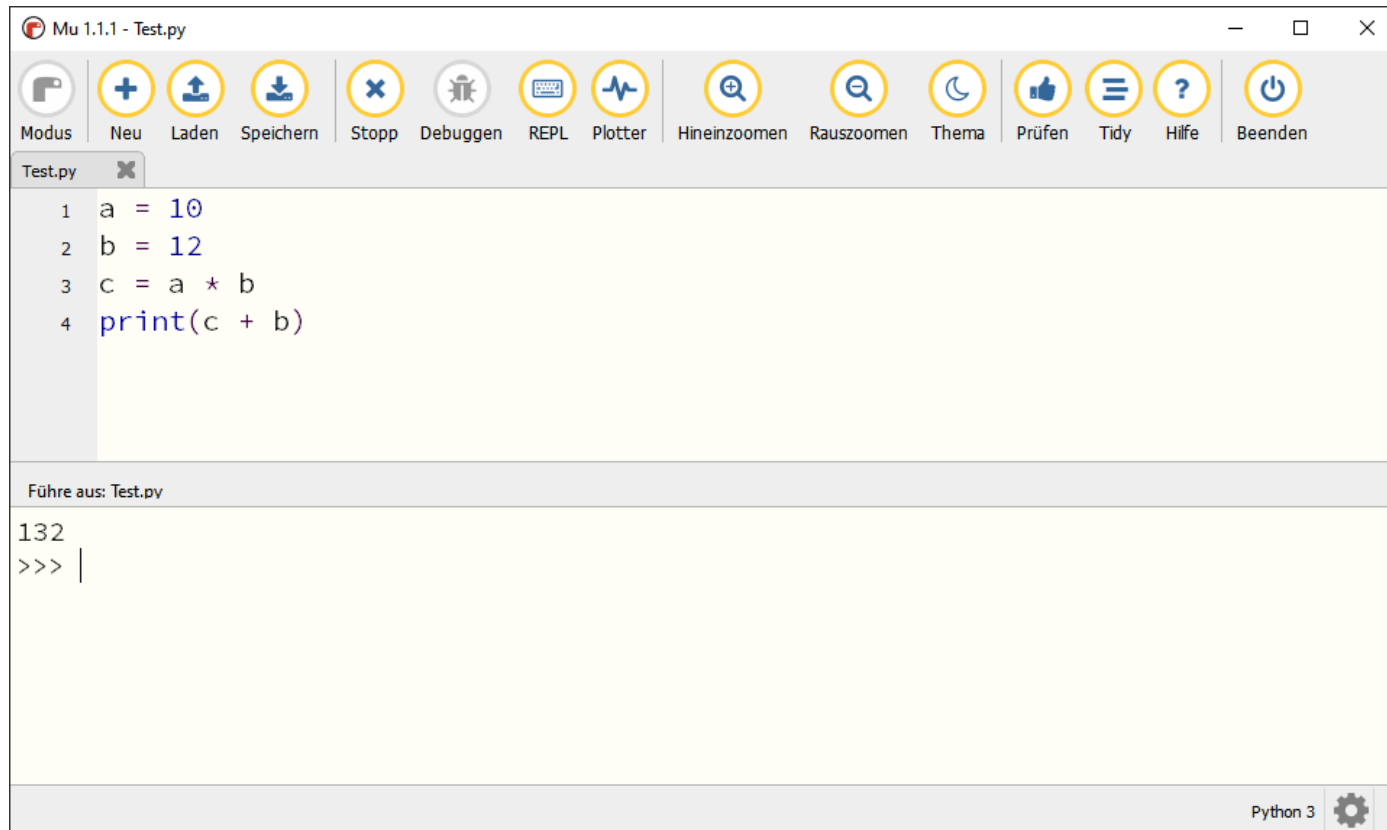


```
1 from microbit import *
2 lightsOn = False
3
4 while True:
5     if microphone.was_event(SoundEvent.LOUD):
6         lightsOn = not lightsOn
7         if lightsOn:
8             display.show(Image('99999:'
9                                 '99999:'
10                                '99999:'
11                                '99999:'
12                                '99999'))
13         else:
14             display.clear()
15             sleep(100)
```

microbit.digital

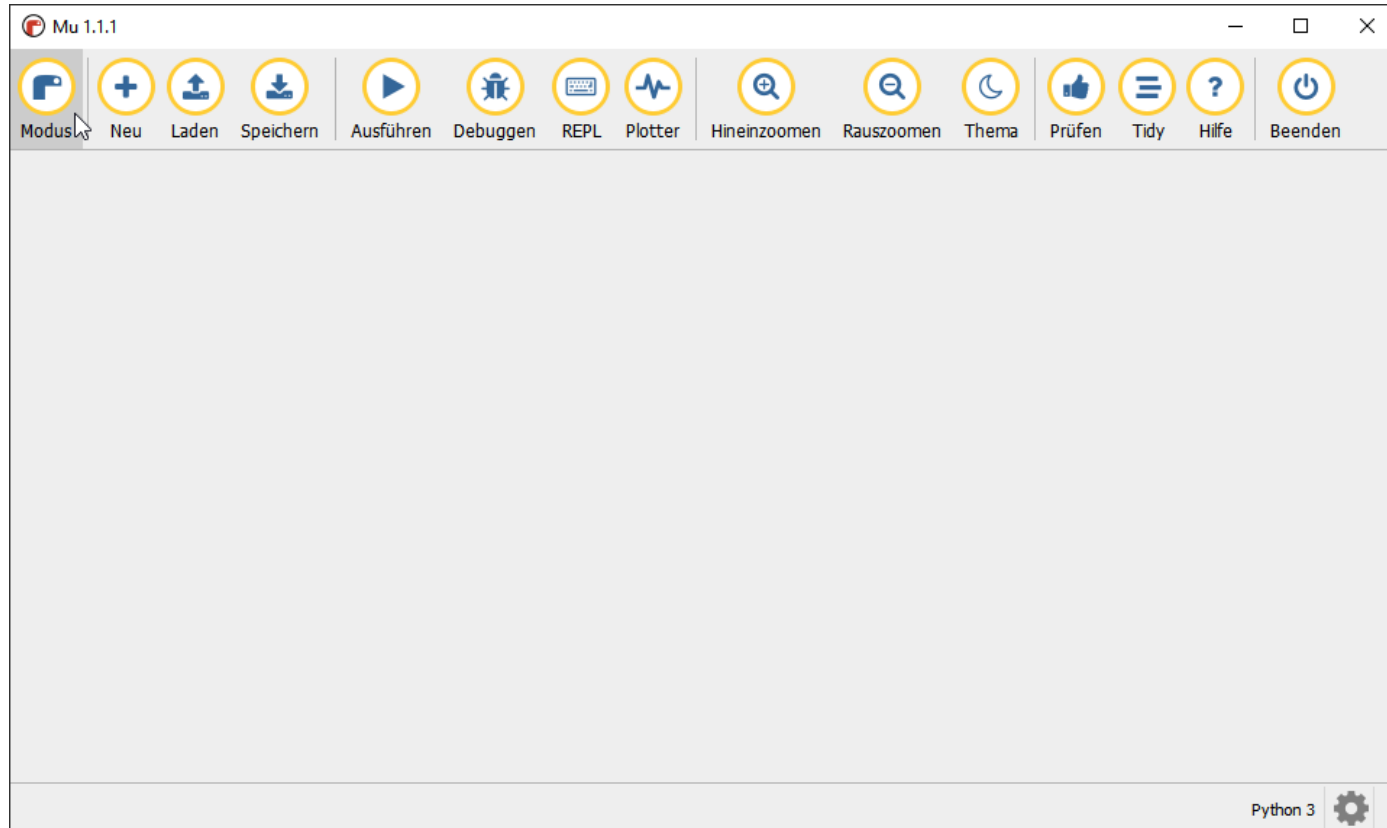
The logo for miXmas digital, featuring the word "miXmas" in a stylized font where the letters are composed of small dots, followed by the word "digital" in a clean, sans-serif font.

Der Mu-Editor



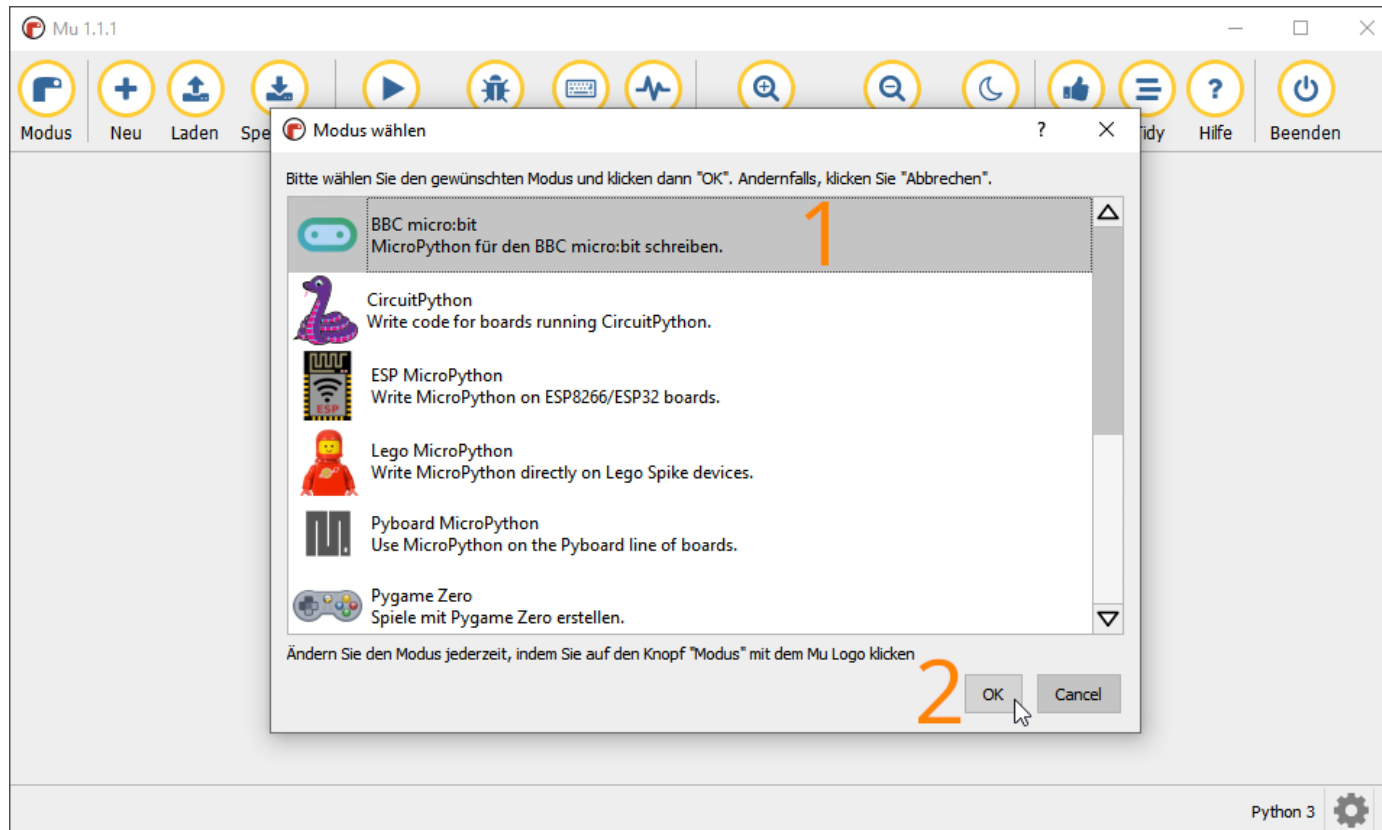
mi.knuss.digital

Der Mu-Editor

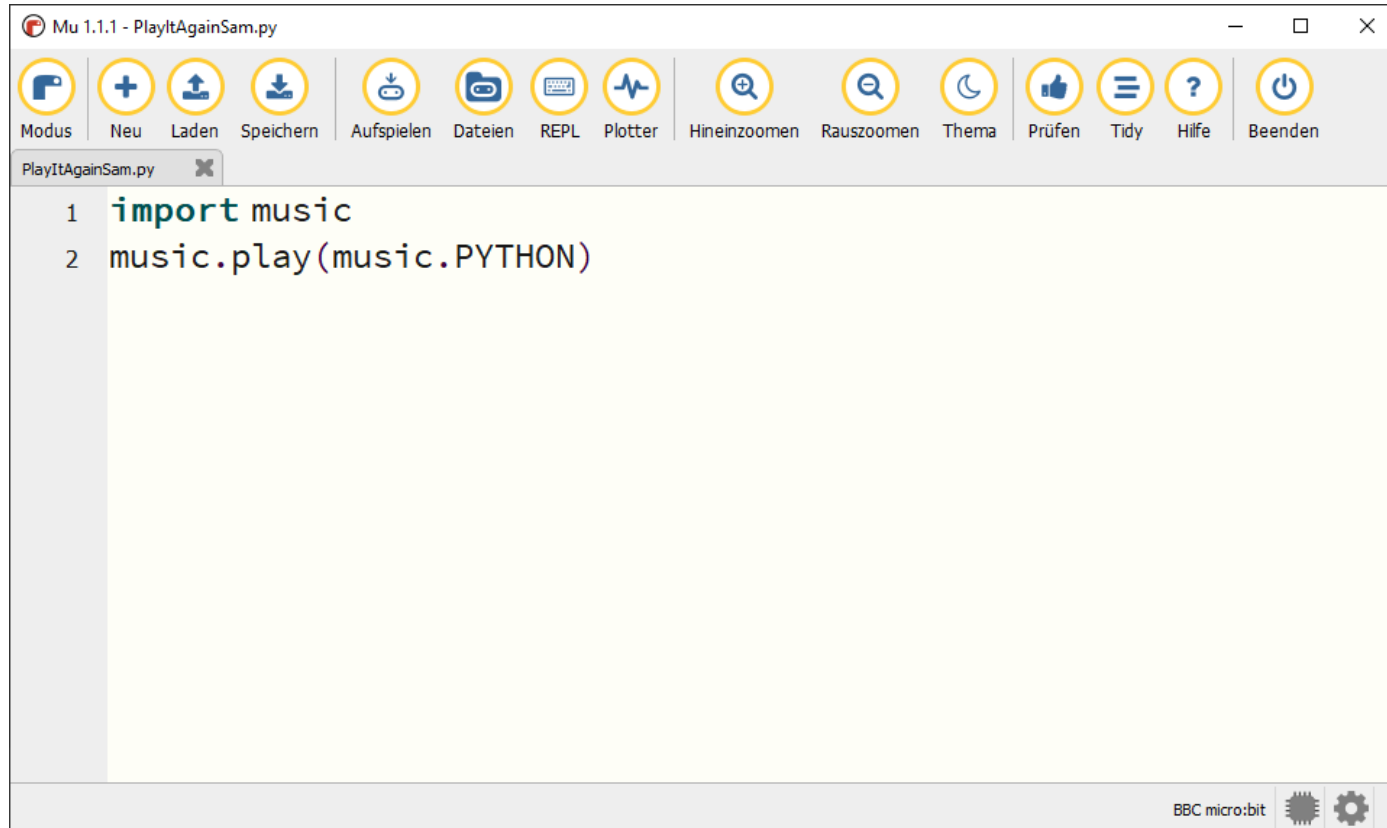


mi-kraus.s.s.digital

Der Mu-Editor

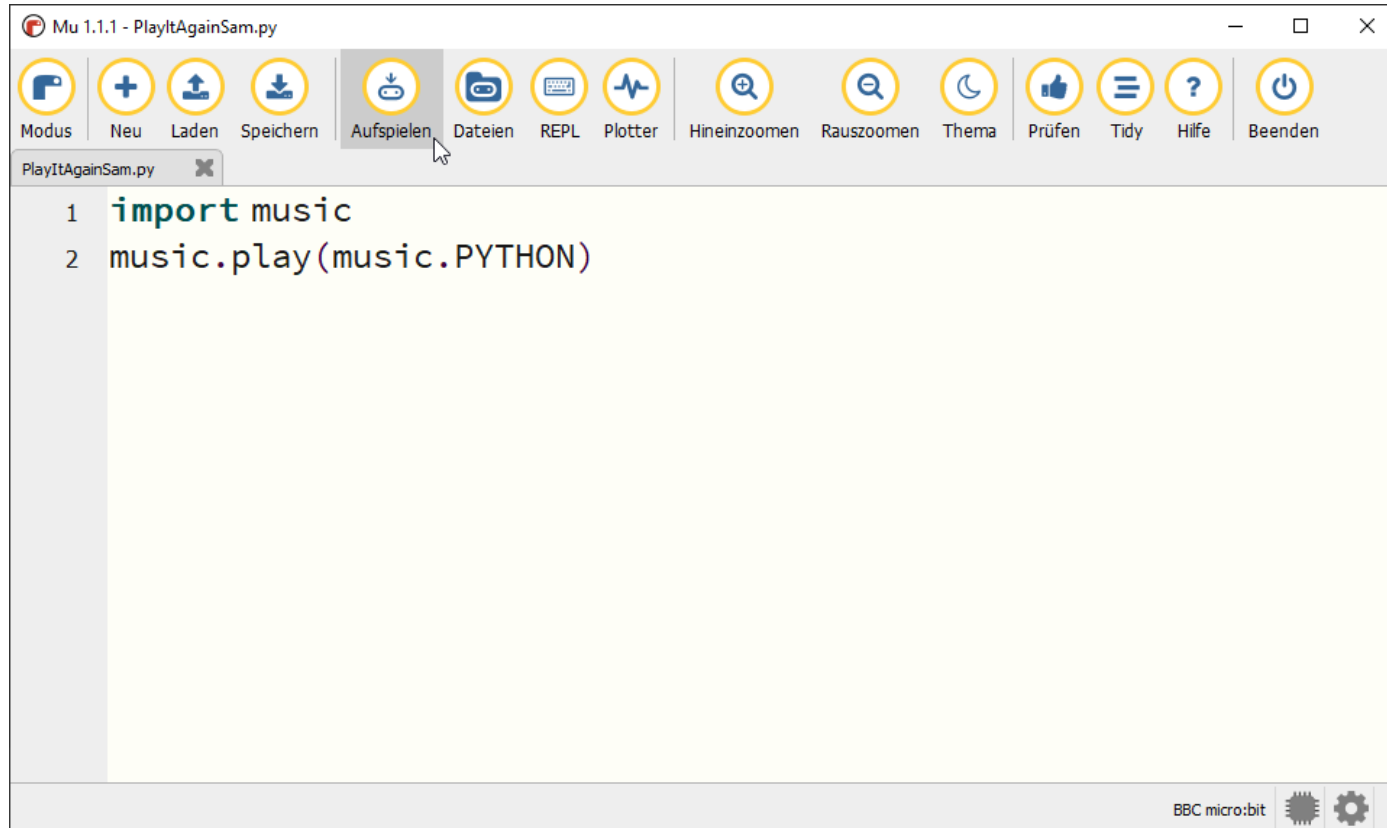


Der Mu-Editor

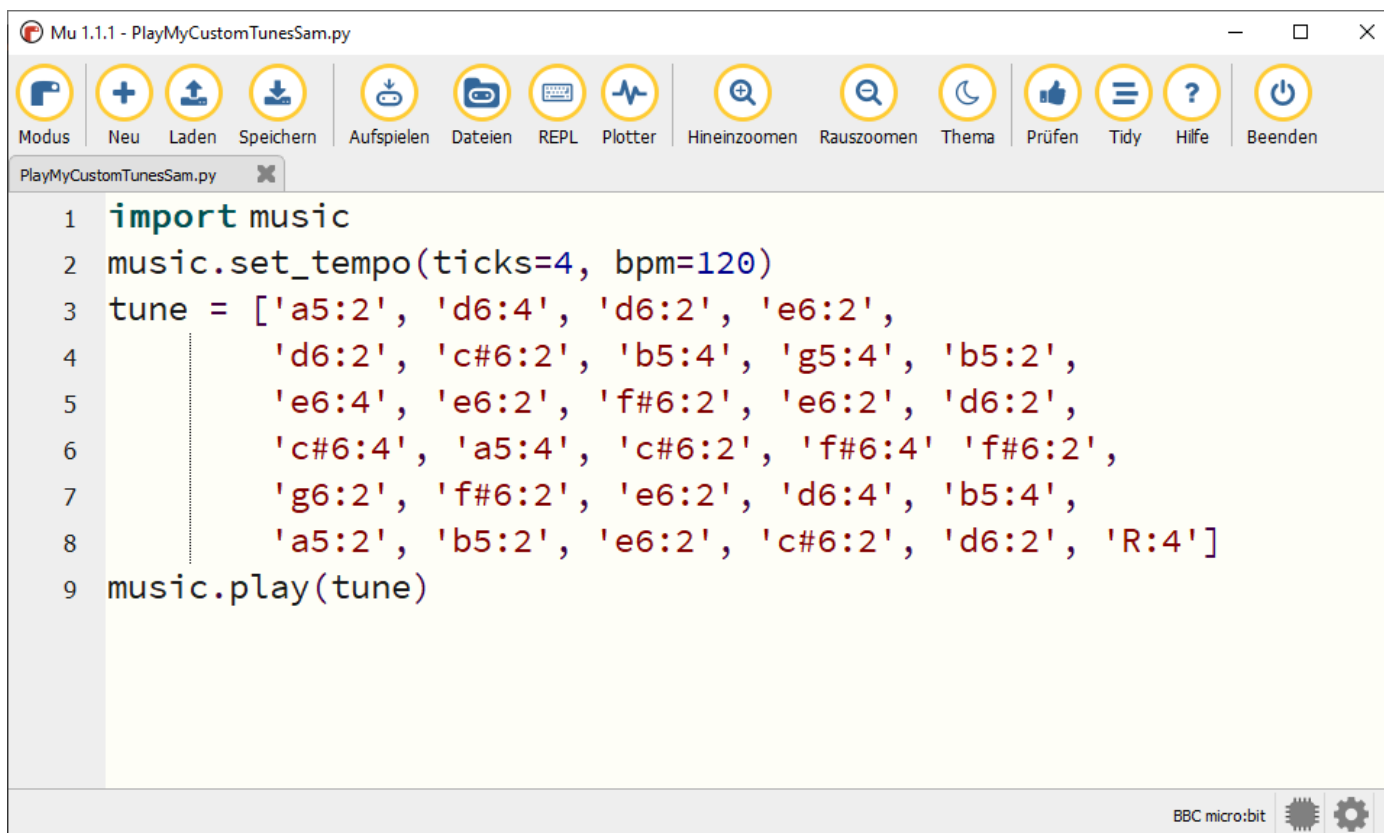


miXmas.digital

Der Mu-Editor



Eigene Melodien spielen



The screenshot shows the Mu Python IDE window titled "Mu 1.1.1 - PlayMyCustomTunesSam.py". The interface includes a toolbar with icons for various functions: Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. Below the toolbar is a tab labeled "PlayMyCustomTunesSam.py". The main editor area contains the following Python code:

```
1 import music
2 music.set_tempo(ticks=4, bpm=120)
3 tune = ['a5:2', 'd6:4', 'd6:2', 'e6:2',
4         'd6:2', 'c#6:2', 'b5:4', 'g5:4', 'b5:2',
5         'e6:4', 'e6:2', 'f#6:2', 'e6:2', 'd6:2',
6         'c#6:4', 'a5:4', 'c#6:2', 'f#6:4', 'f#6:2',
7         'g6:2', 'f#6:2', 'e6:2', 'd6:4', 'b5:4',
8         'a5:2', 'b5:2', 'e6:2', 'c#6:2', 'd6:2', 'R:4']
9 music.play(tune)
```

At the bottom right of the window, there is a "BBC micro:bit" logo and two icons: a sun and a gear.

4 Ticks pro Beat, 120 Beats pro Minute

Buchstabe: Note

Kleiner Buchstabe: Flache Note

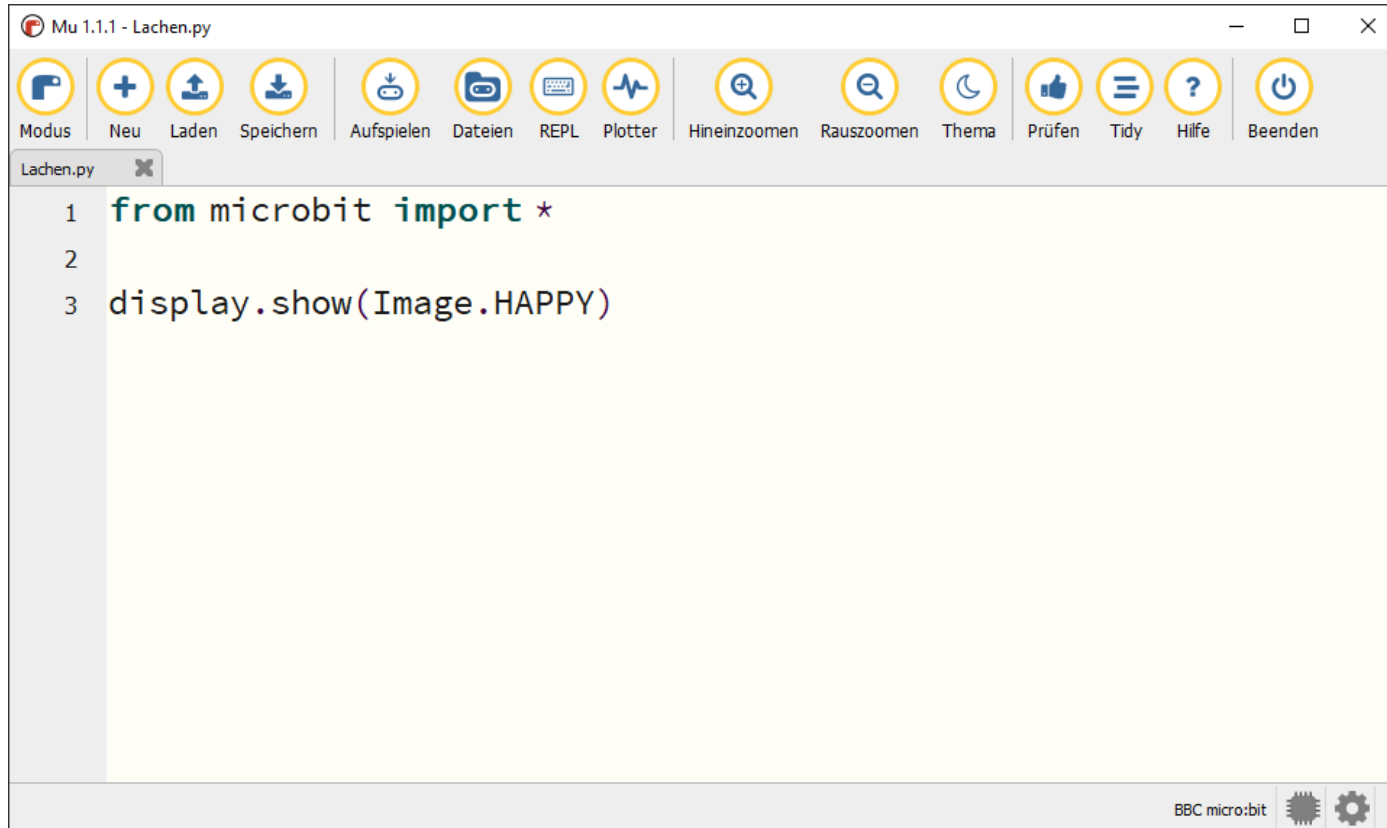
hinter Buchstabe: Scharfe Note

Zahl vor dem : Oktave

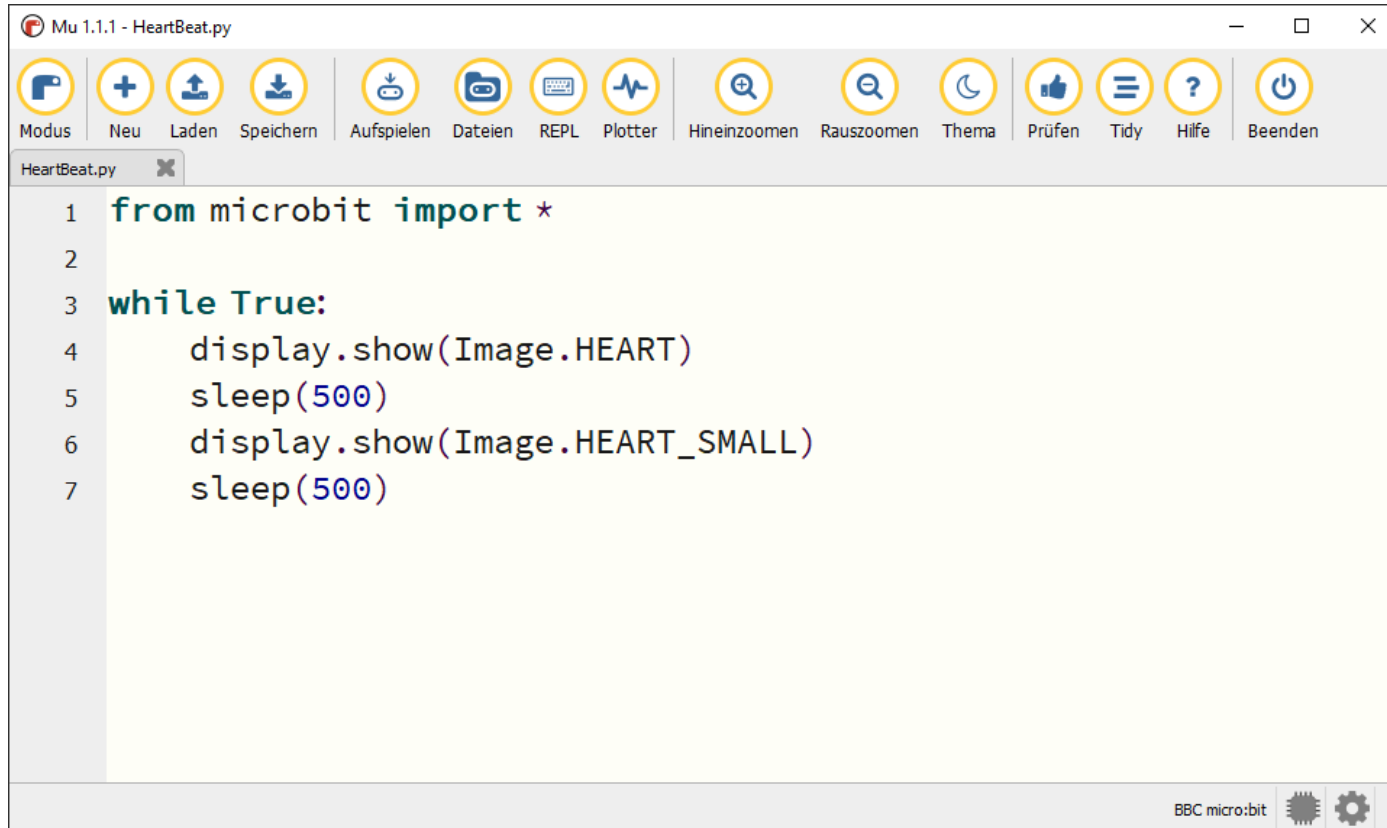
Zahl hinter dem : Länge (in Ticks)

Spieler Melodie ab

Ein Bild anzeigen



Zwei Bilder anzeigen

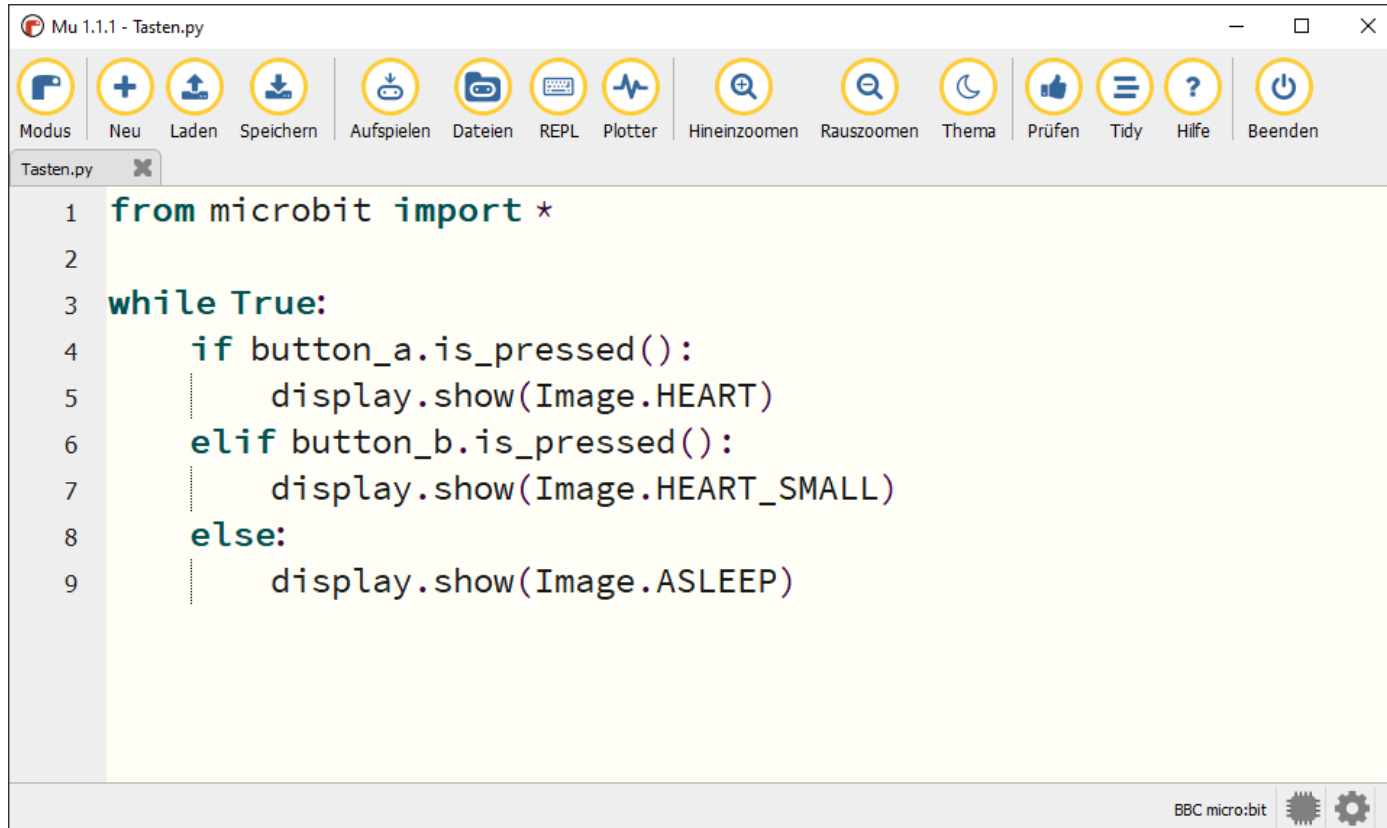


The screenshot shows the Mu Python IDE window titled "Mu 1.1.1 - HeartBeat.py". The interface includes a toolbar with icons for various functions like Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. Below the toolbar is a tab labeled "HeartBeat.py". The main code editor contains the following Python code:

```
1 from microbit import *
2
3 while True:
4     display.show(Image.HEART)
5     sleep(500)
6     display.show(Image.HEART_SMALL)
7     sleep(500)
```

The bottom status bar indicates "BBC micro:bit" with a micro:bit logo and a settings gear icon.

Auf Tasten reagieren



The screenshot shows the Mu Python IDE window titled "Mu 1.1.1 - Tasten.py". The interface includes a toolbar with icons for various functions: Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. Below the toolbar is a tab labeled "Tasten.py". The main code area contains the following Python code:

```
1 from microbit import *
2
3 while True:
4     if button_a.is_pressed():
5         display.show(Image.HEART)
6     elif button_b.is_pressed():
7         display.show(Image.HEART_SMALL)
8     else:
9         display.show(Image.ASLEEP)
```

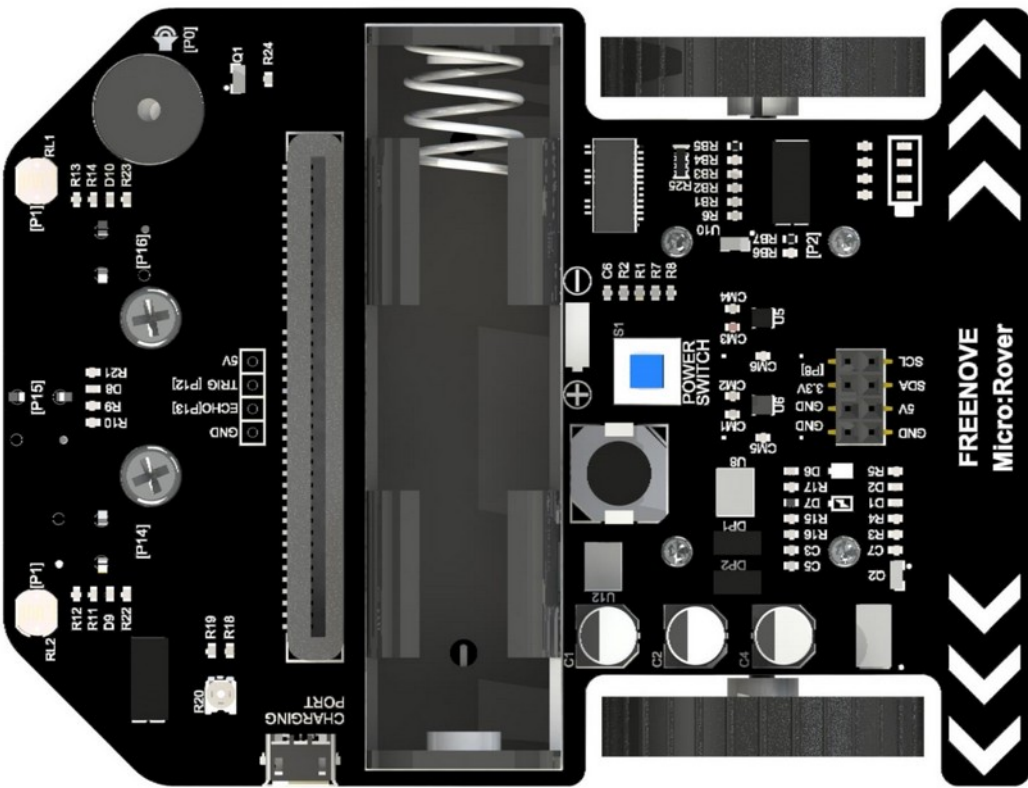
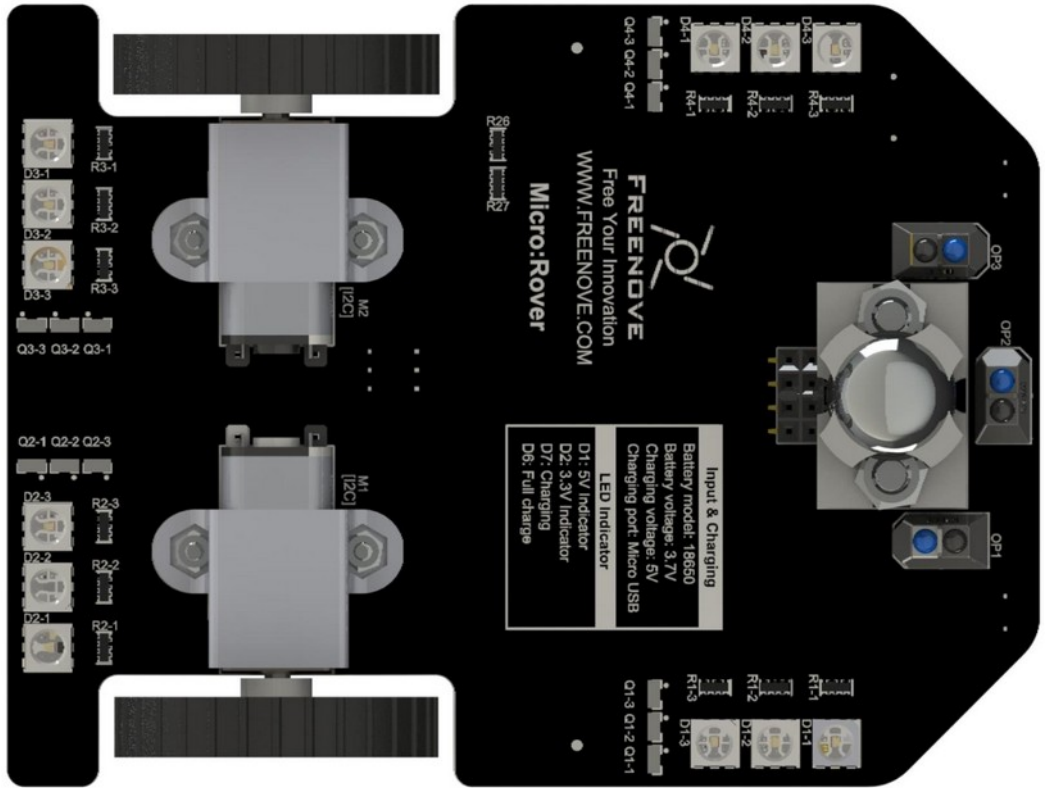
At the bottom right of the window, there is a status bar with the text "BBC micro:bit" and two icons: a micro:bit logo and a settings gear.

Kurze Pause

niXmas.digital

analog.digital

Free Your Innovation

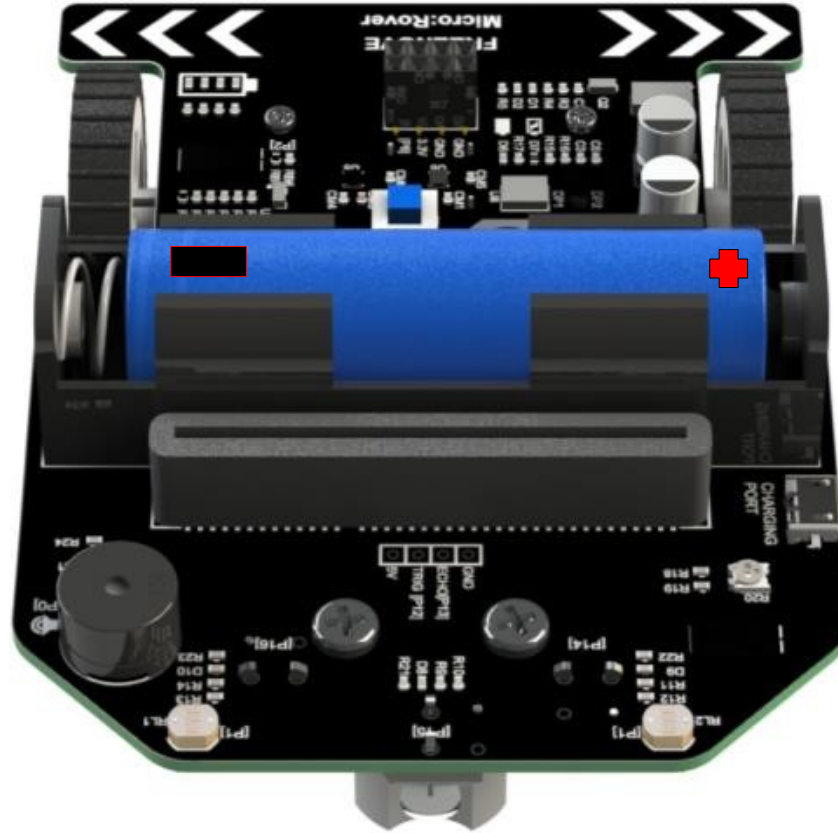


how to assemble a rover



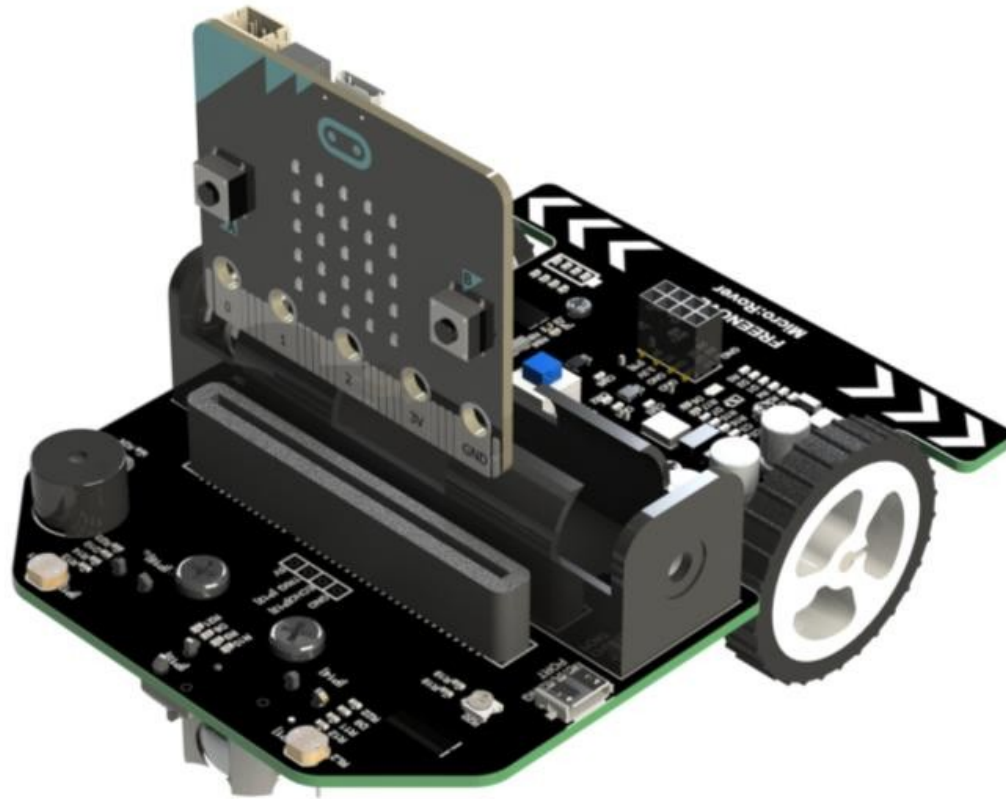
www.digital

university of the south



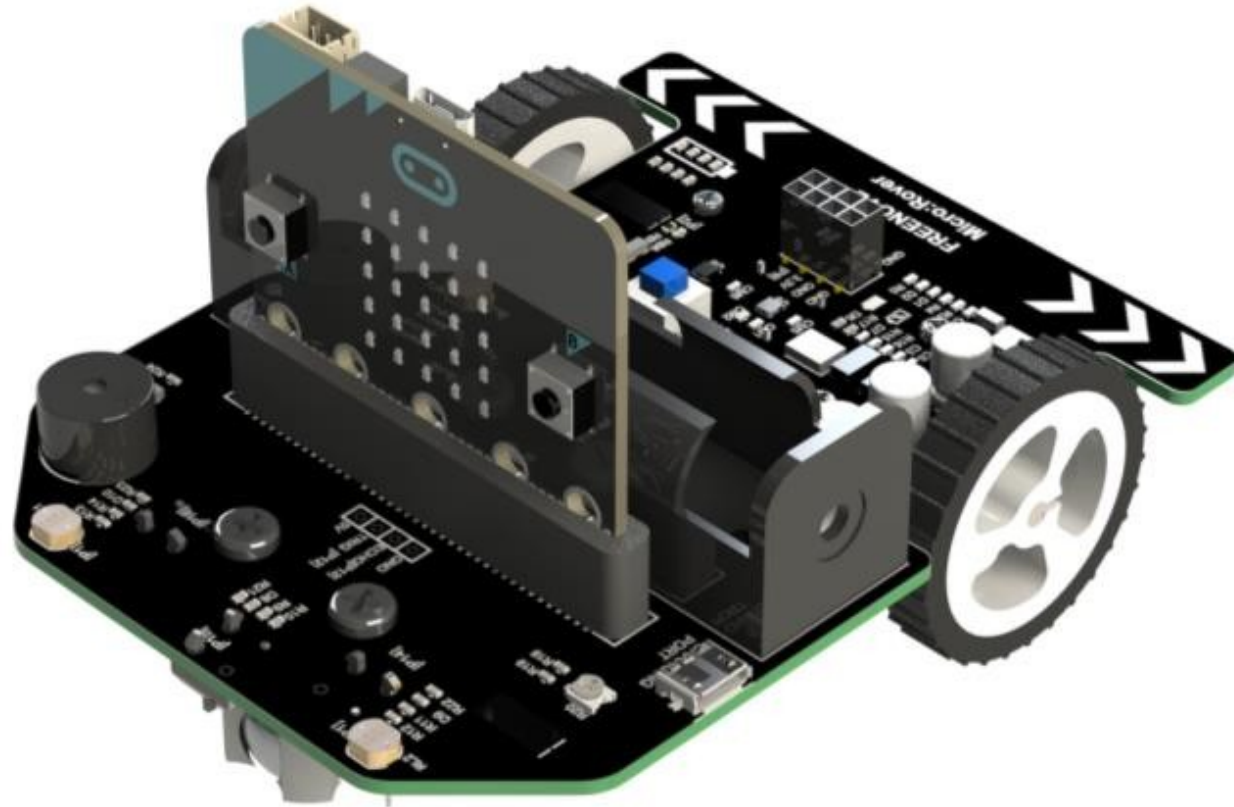
digital

Power to a simple machine



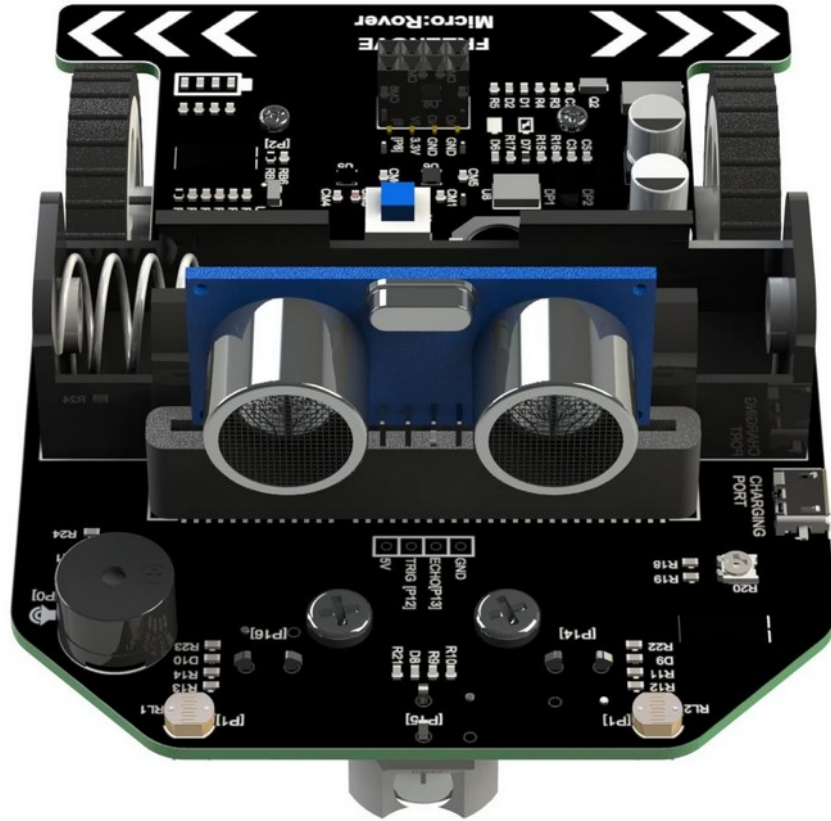
mikrus.digital

How to assemble a robot



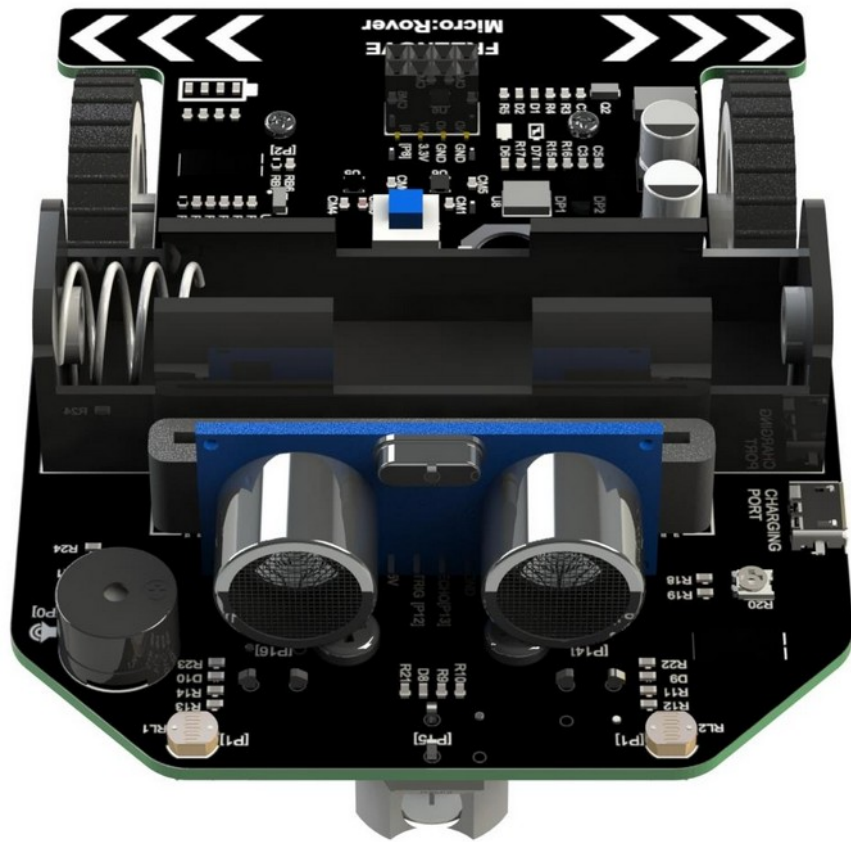
mixmus.digital

ultra-compact digital



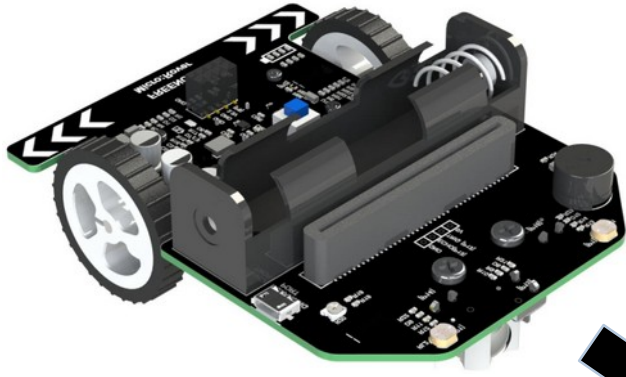
mus.digital

Micro:Rover



digital

How to Program



Mu 1.1.1 - Rover.py

Modus Neu Laden Speichern Aufspielen Dateien REPL Plotter Hineinzoomen Rauszoomen Thema Prüfen Tidy Hilfe Beenden

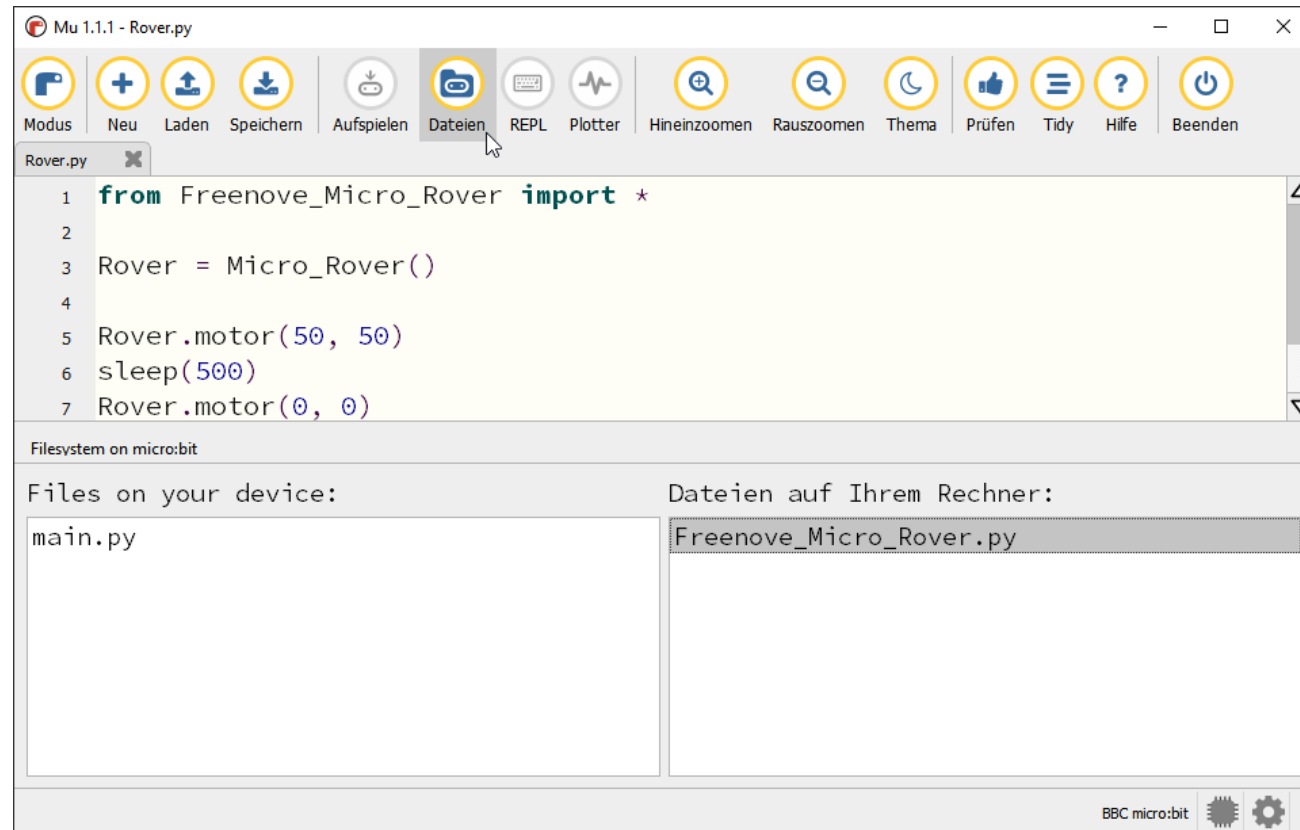
Rover.py

```
1 from Freenove_Micro_Rover import *
2
3 Rover = Micro_Rover()
4
5 Rover.motor(50, 50)
6 sleep(500)
7 Rover.motor(0, 0)
8 Rover.all_led_show(255, 255, 255, 0)
```

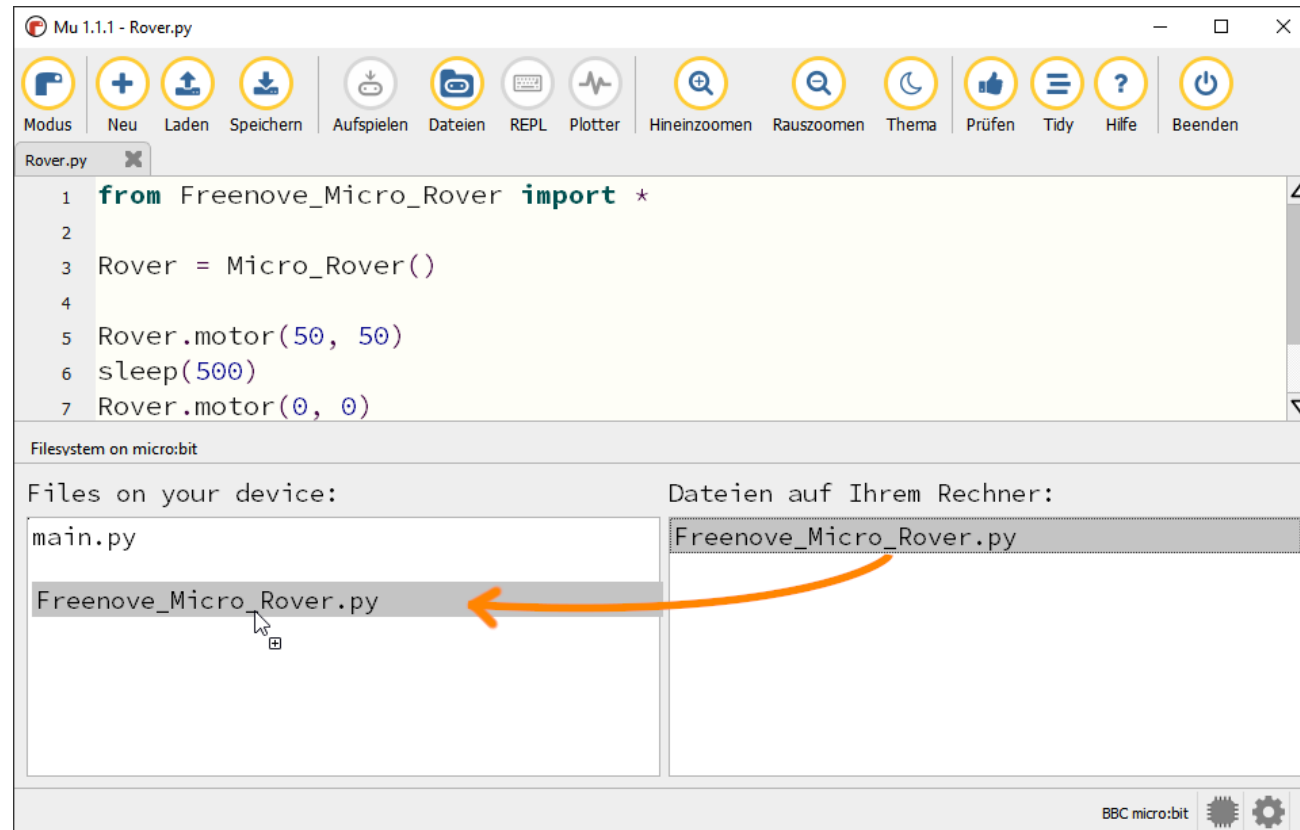
BBC micro:bit

miXmus.digital

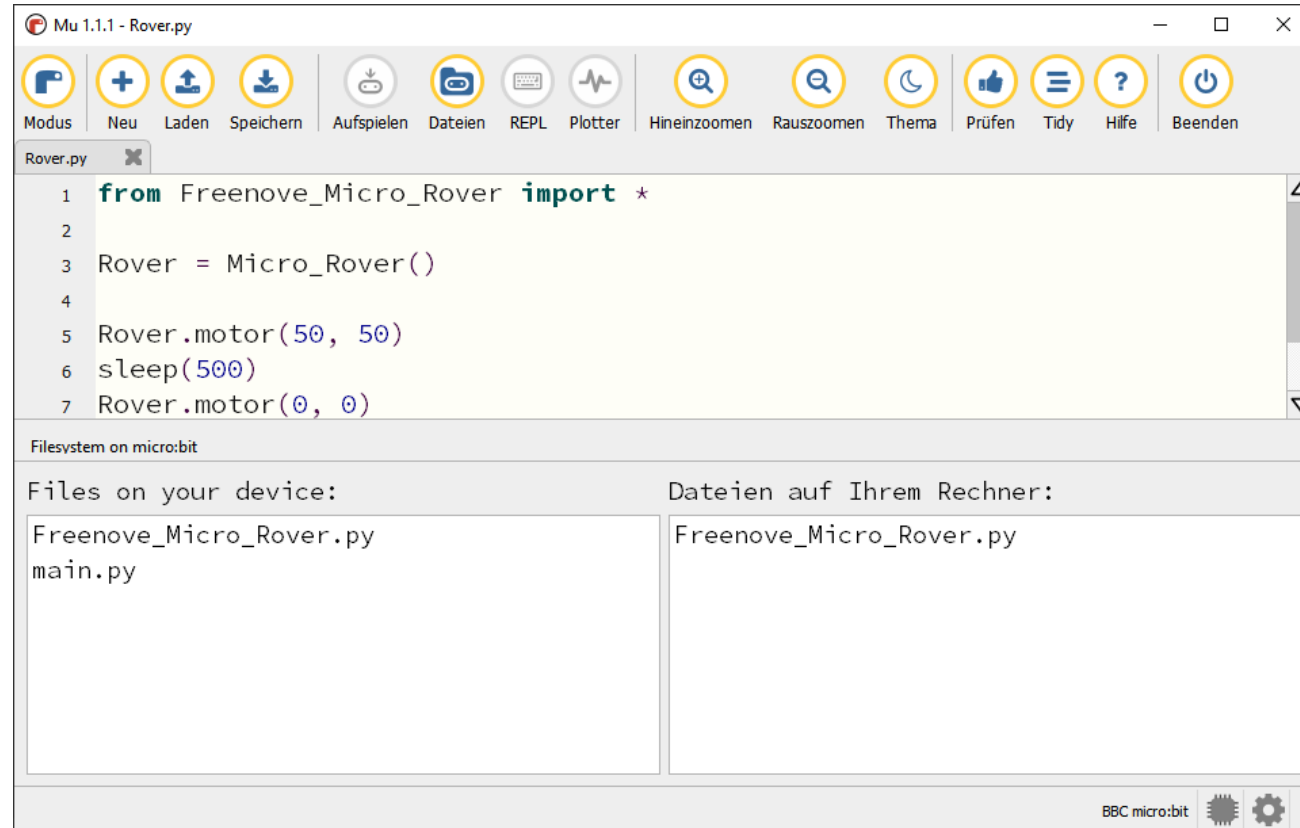
Bibliotheken importieren



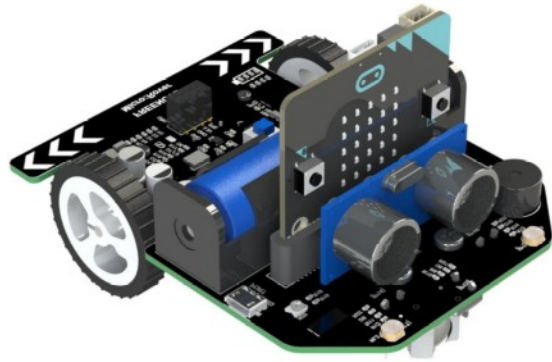
Bibliotheken importieren



Bibliotheken importieren

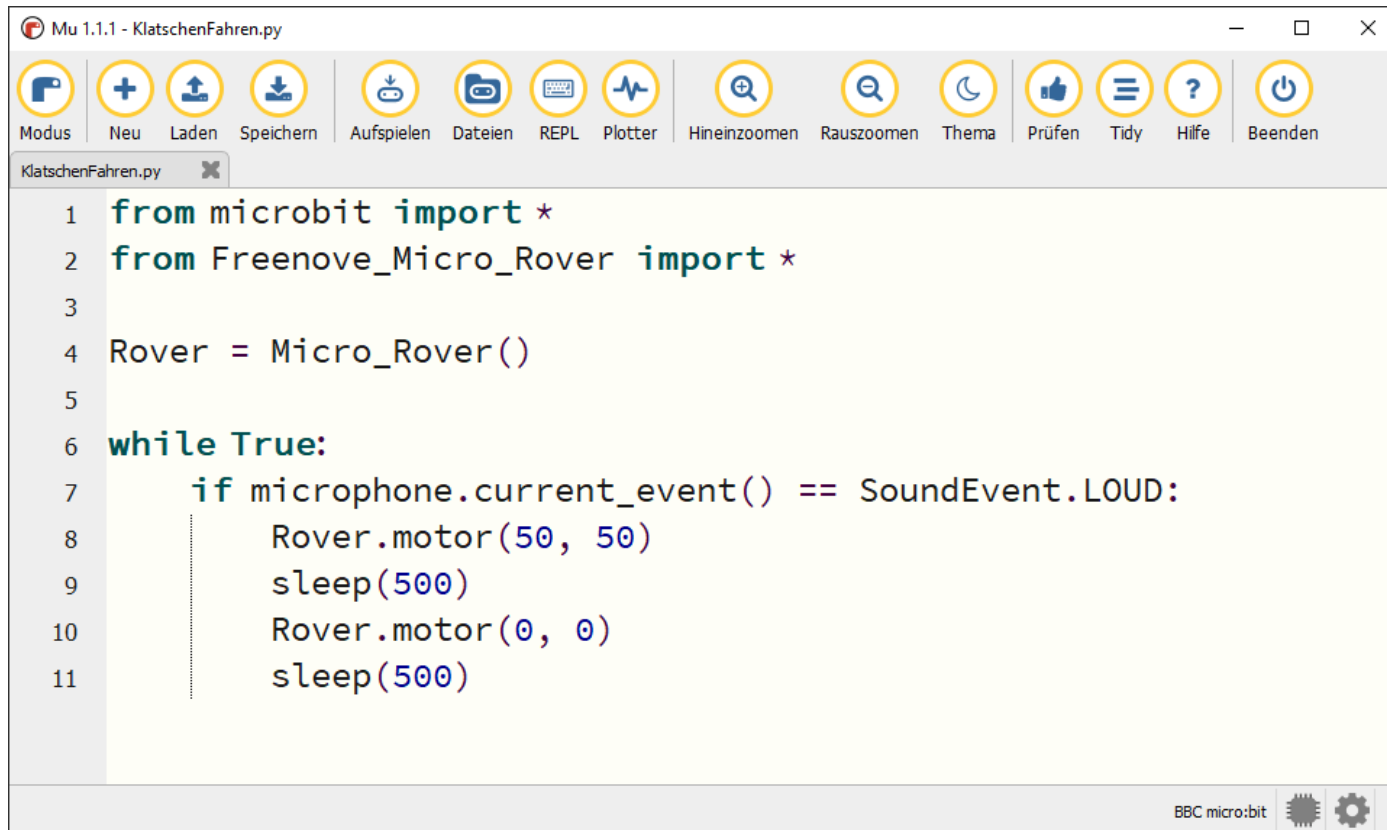


Router und Laptop verbinden



mixmus.digital

Beim Klatschen fahren



Mu 1.1.1 - KlatschenFahren.py

Modus Neu Laden Speichern Aufspielen Dateien REPL Plotter Hineinzoomen Rauszoomen Thema Prüfen Tidy Hilfe Beenden

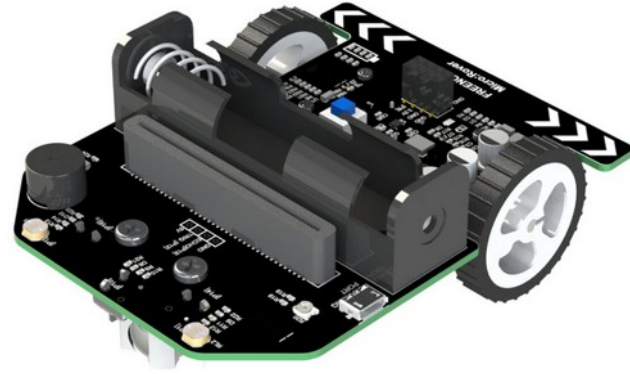
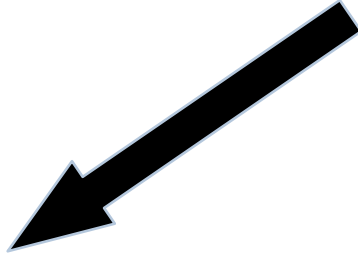
KlatschenFahren.py

```
1 from microbit import *
2 from Freenove_Micro_Rover import *
3
4 Rover = Micro_Rover()
5
6 while True:
7     if microphone.current_event() == SoundEvent.LOUD:
8         Rover.motor(50, 50)
9         sleep(500)
10        Rover.motor(0, 0)
11        sleep(500)
```

BBC micro:bit

Aufgabe 1

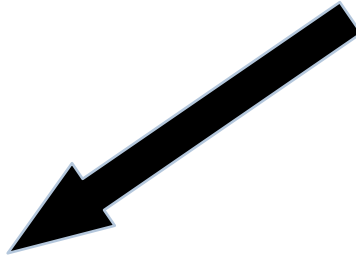
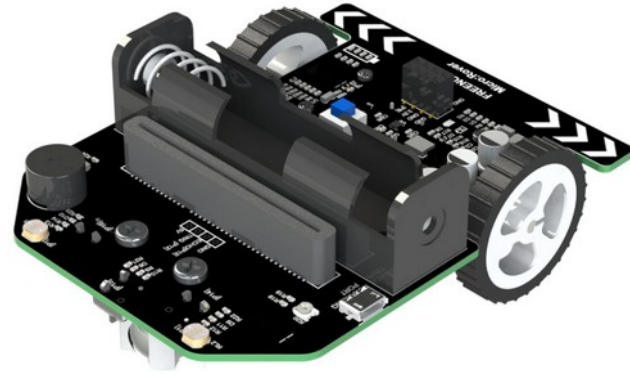
Rover dreht sich durch Klatschen



mikrus.digital

Aufgabe 2

Quadrat zeichnen

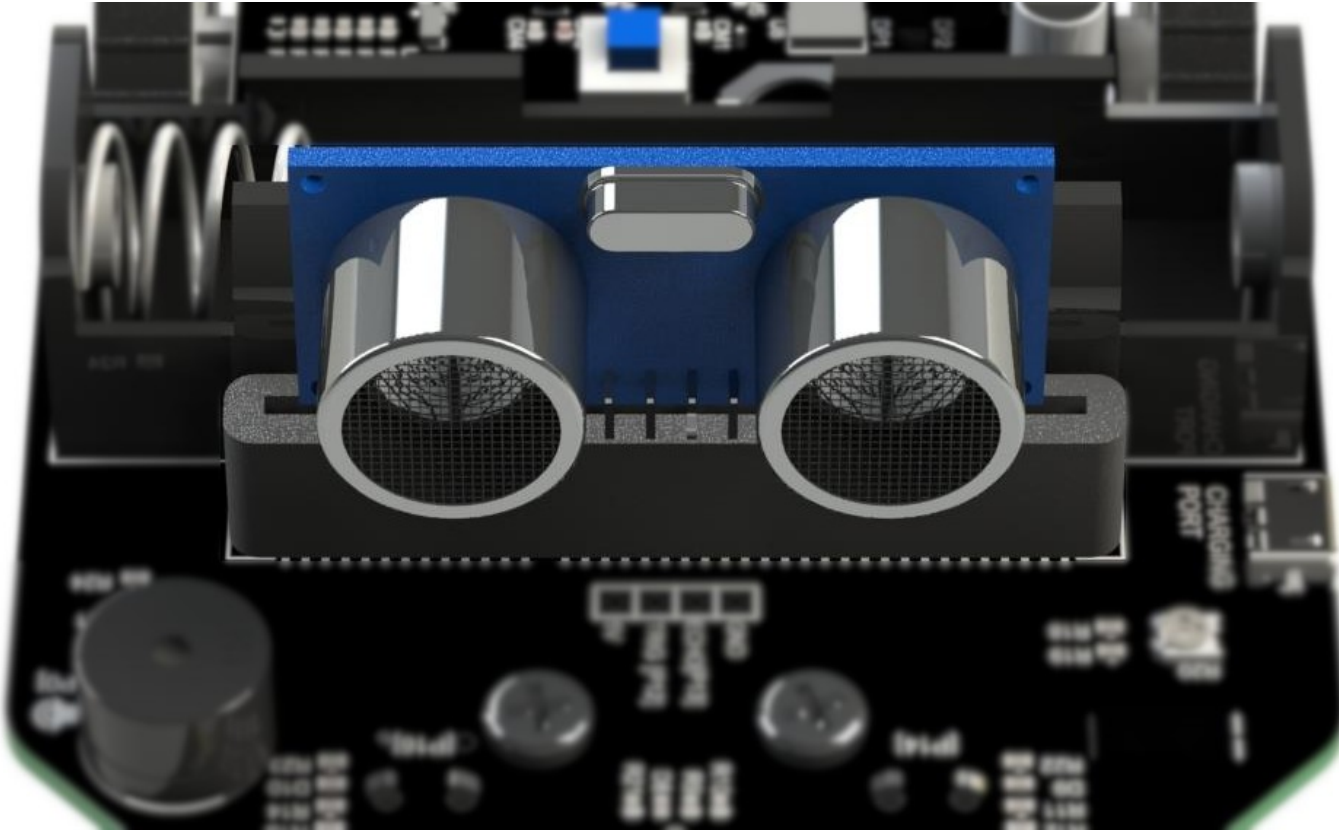


mixmus.digital

Abstand Hitte!

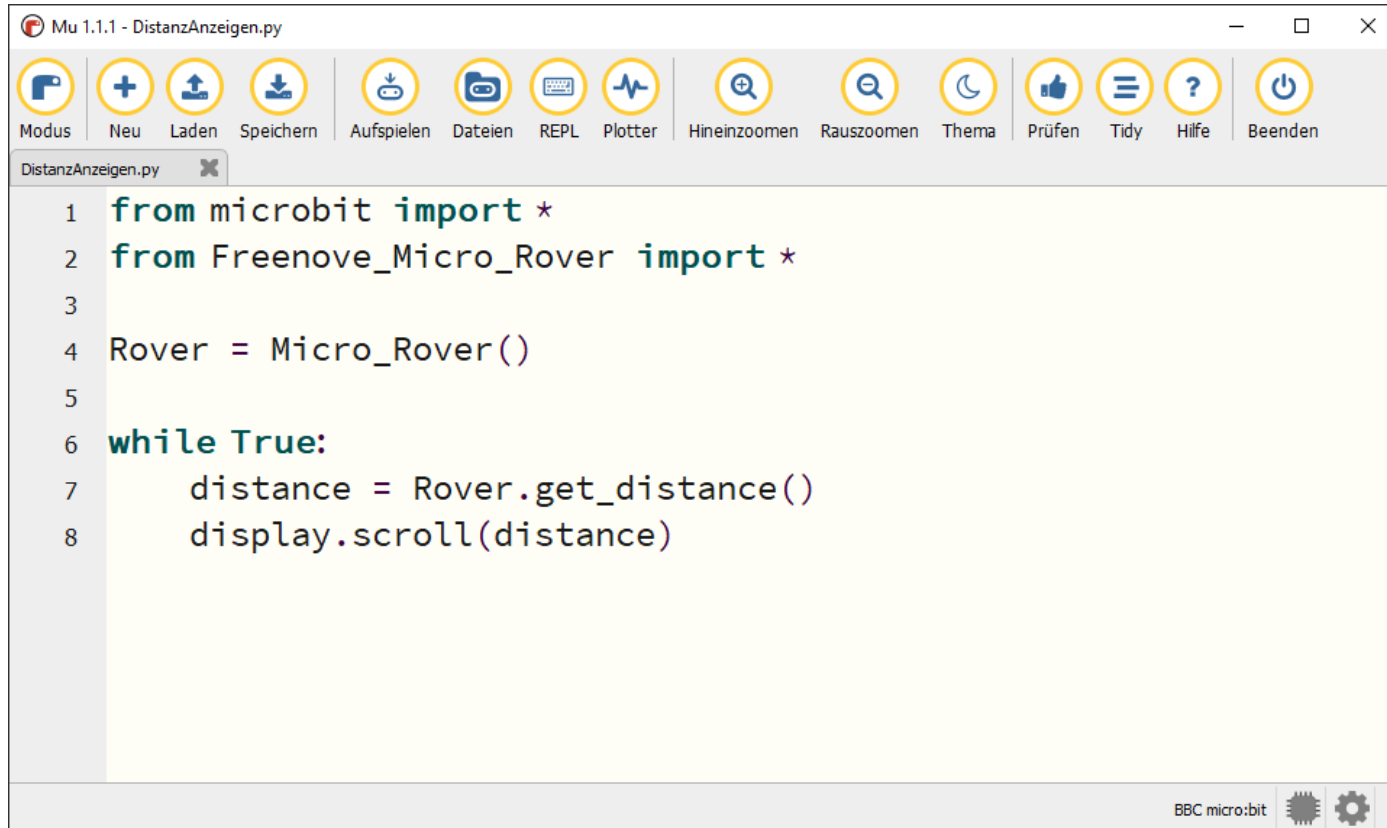
- Ultraschallsensor
- Gibt Abstand in Zentimeter aus

`Rover.get_distance()`



`miXmuss.digital`

Abstand anzeigen



The screenshot shows the Mu Python IDE interface. The title bar reads "Mu 1.1.1 - DistanzAnzeigen.py". The toolbar contains icons for various functions: Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. The main editor area displays the following Python code:

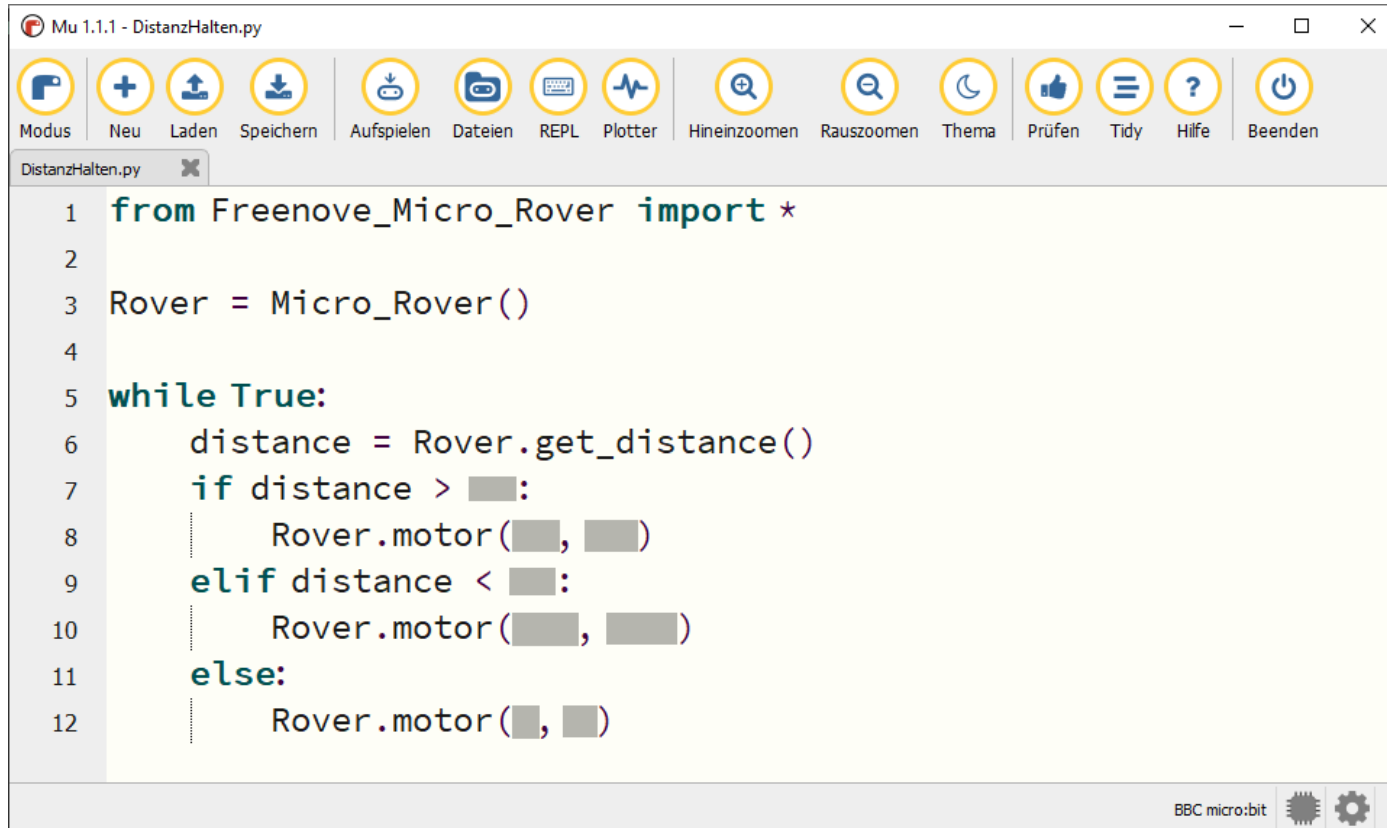
```
1 from microbit import *
2 from Freenove_Micro_Rover import *
3
4 Rover = Micro_Rover()
5
6 while True:
7     distance = Rover.get_distance()
8     display.scroll(distance)
```

The status bar at the bottom right indicates "BBC micro:bit" with a micro:bit logo and a settings gear icon.

Abstand halten

- Vorwärtsfahren solange Hindernis weiter als 10 cm entfernt ist
- Wird ein Hindernis im Abstand von 10 cm erkannt, soll angehalten werden
- Kommt das Hindernis näher an den Roboter (weniger als 10 cm) soll rückwärts gefahren werden

Abstand halten

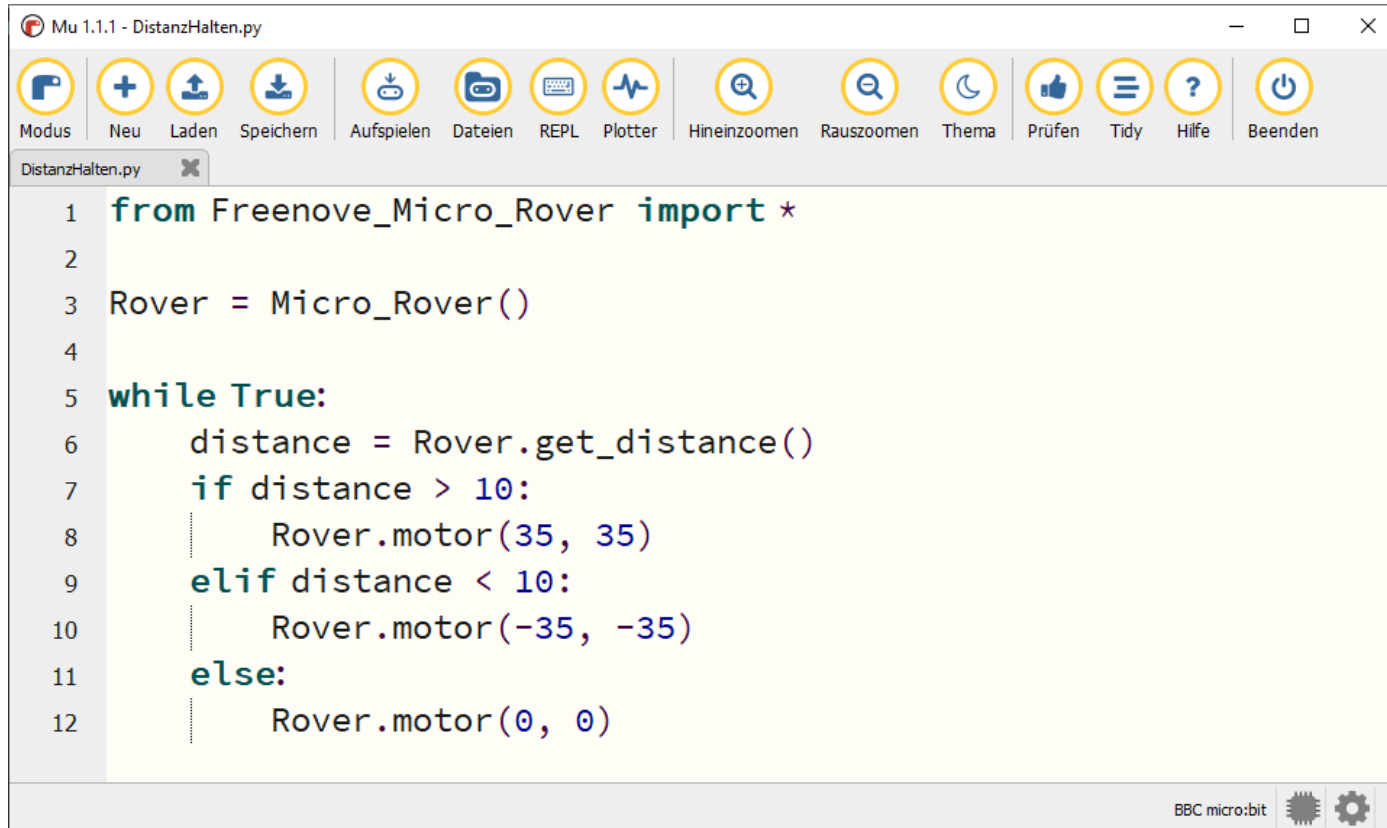


The screenshot shows the Mu Python IDE interface. The title bar reads "Mu 1.1.1 - DistanzHalten.py". The toolbar contains icons for various functions: Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. Below the toolbar, a tab labeled "DistanzHalten.py" is active. The code editor displays the following Python code:

```
1 from Freenove_Micro_Rover import *
2
3 Rover = Micro_Rover()
4
5 while True:
6     distance = Rover.get_distance()
7     if distance > █:
8         Rover.motor(█, █)
9     elif distance < █:
10        Rover.motor(█, █)
11    else:
12        Rover.motor(█, █)
```

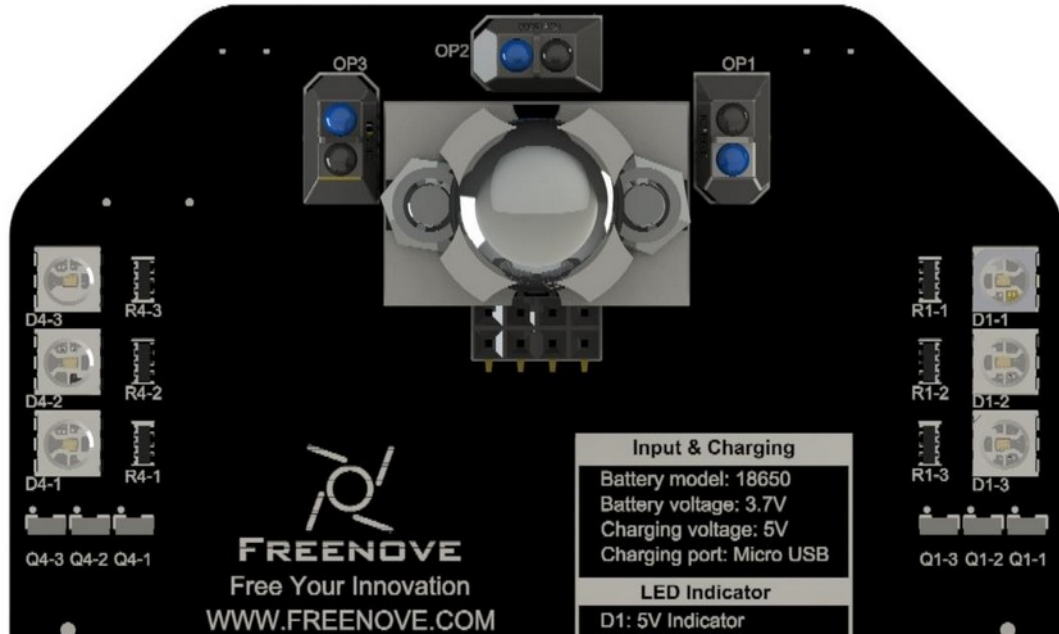
The bottom status bar indicates "BBC micro:bit" with icons for the micro:bit and settings.

Abstand halten



```
1 from Freenove_Micro_Rover import *
2
3 Rover = Micro_Rover()
4
5 while True:
6     distance = Rover.get_distance()
7     if distance > 10:
8         Rover.motor(35, 35)
9     elif distance < 10:
10        Rover.motor(-35, -35)
11    else:
12        Rover.motor(0, 0)
```


Untergrund erkennen



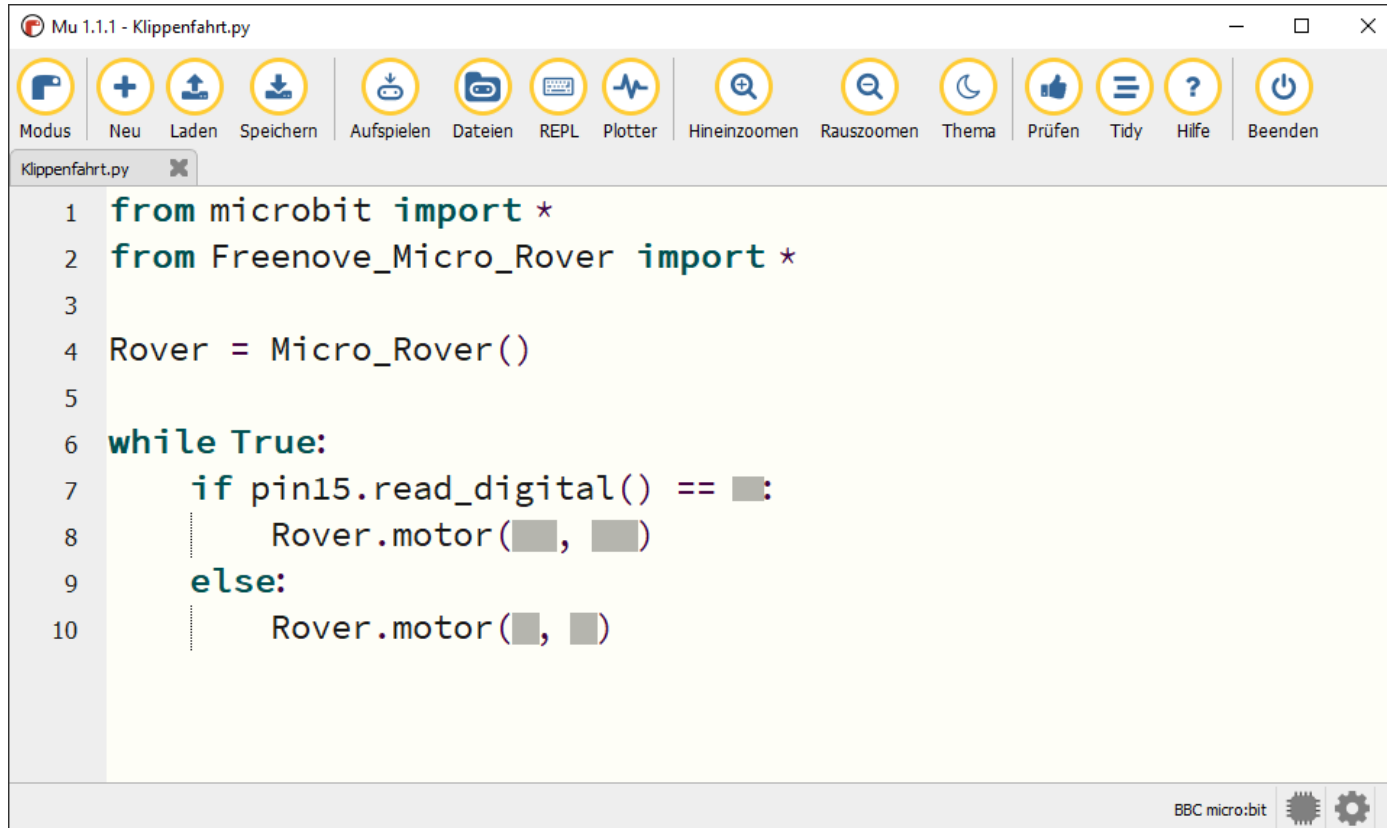
- Drei Infrarot-Sensoren
- Gibt an, ob der Untergrund hell 0 oder dunkel 1 ist
- Vorne: pin15
Links: pin14
Rechts: pin16
- `pin15.read_digital()`

`pin15.read_digital()`

An Klippe anhalten

- Unser Roboter mag keine Abgründe, denn er hat Höhenangst
- Damit er trotzdem auf den Tischen fahren kann, soll unser Roboter anhalten, wenn der vordere Sensor die Tischplatte nicht mehr erkennt
- Unsere Tische sind sehr hell, für den Roboter schon fast weiß

Am Klippe anhalten



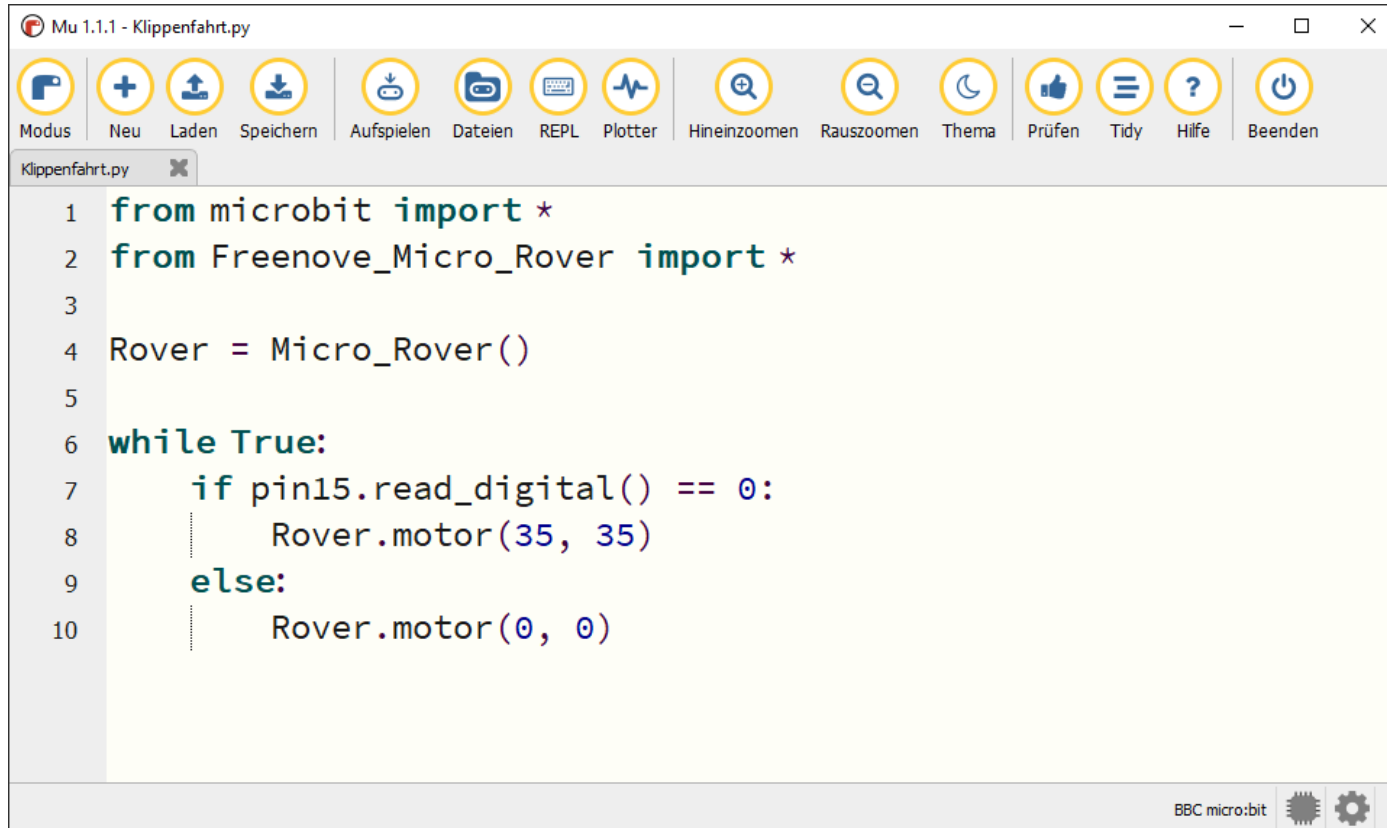
The screenshot shows the Mu Python IDE interface. The title bar reads "Mu 1.1.1 - Klippenfahrt.py". The toolbar contains icons for various functions: Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. The main editor area displays the following Python code:

```
1 from microbit import *
2 from Freenove_Micro_Rover import *
3
4 Rover = Micro_Rover()
5
6 while True:
7     if pin15.read_digital() == 1:
8         Rover.motor(1, 1)
9     else:
10        Rover.motor(1, 1)
```

The bottom status bar indicates "BBC micro:bit" with icons for the micro:bit and settings.

pin15.read_digital

Am Klippe anhalten



The screenshot shows the Mu Python IDE window titled "Mu 1.1.1 - Klippenfahrt.py". The interface includes a toolbar with icons for various functions like Modus, Neu, Laden, Speichern, Aufspielen, Dateien, REPL, Plotter, Hineinzoomen, Rauszoomen, Thema, Prüfen, Tidy, Hilfe, and Beenden. Below the toolbar is a tab labeled "Klippenfahrt.py". The main area contains the following Python code:

```
1 from microbit import *
2 from Freenove_Micro_Rover import *
3
4 Rover = Micro_Rover()
5
6 while True:
7     if pin15.read_digital() == 0:
8         Rover.motor(35, 35)
9     else:
10        Rover.motor(0, 0)
```

At the bottom right of the window, there is a status bar with the text "BBC micro:bit" and two icons: a micro:bit logo and a settings gear.

pin15.read_digital

Auf der Linie bleiben

- Unser Roboter fährt gerne Rennen
- Eine Rennstrecke befindet sich im Karton des Roboters



mikrus.digital

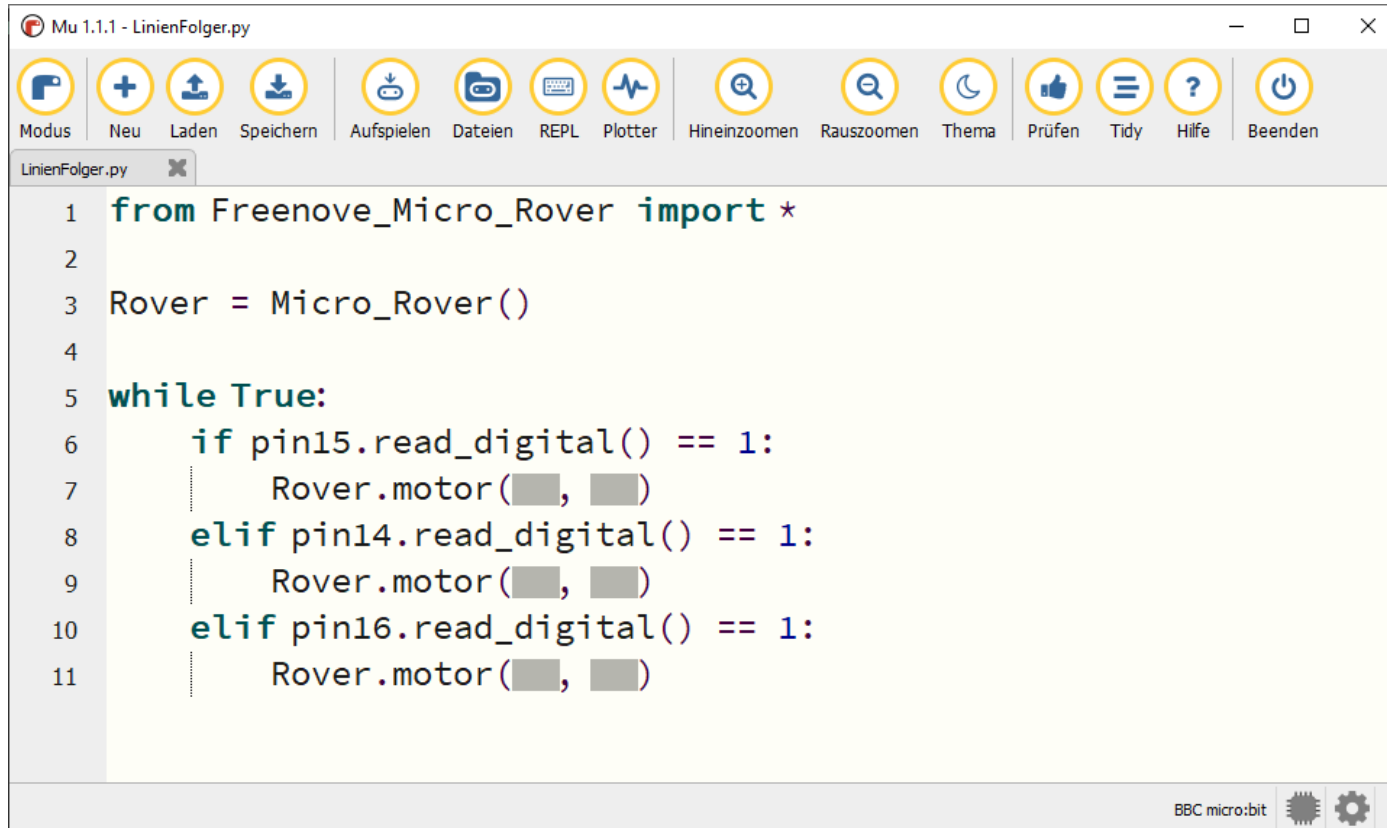
Auf der Linie bleiben

- Die Untergrund-Sensoren des Roboters können die schwarze Linie vom weißen Papier unterscheiden
- Auf den Geraden ist der Roboter mittig auf der Linie, dann darf er vorwärts fahren
- An den Ecken der Strecke muss der Roboter dem Streckenverlauf aber folgen



mikrus.digital

Auf der Linie bleiben



Mu 1.1.1 - LinienFolger.py

Modus Neu Laden Speichern Aufspielen Dateien REPL Plotter Hineinzoomen Rauszoomen Thema Prüfen Tidy Hilfe Beenden

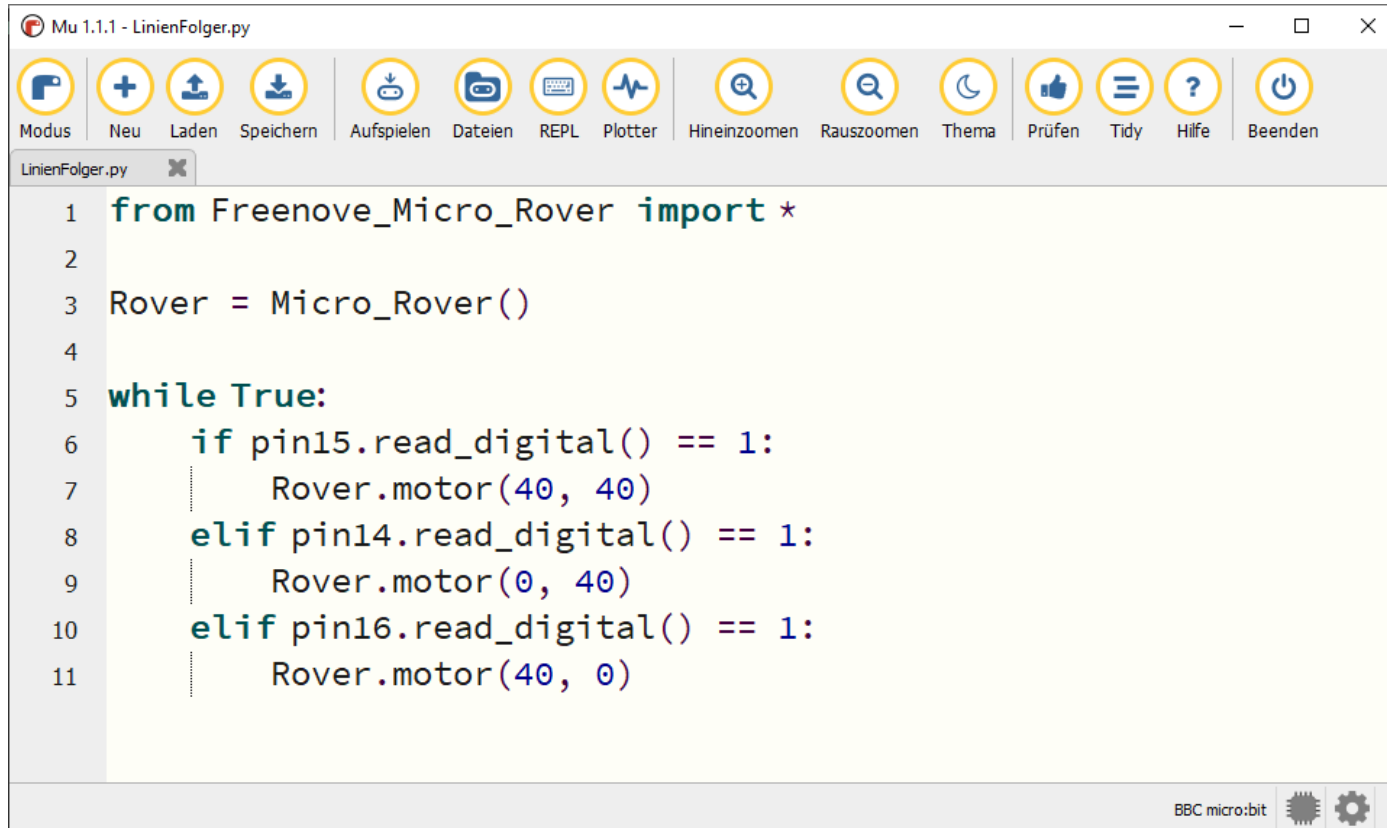
LinienFolger.py

```
1 from Freenove_Micro_Rover import *
2
3 Rover = Micro_Rover()
4
5 while True:
6     if pin15.read_digital() == 1:
7         Rover.motor(■, ■)
8     elif pin14.read_digital() == 1:
9         Rover.motor(■, ■)
10    elif pin16.read_digital() == 1:
11        Rover.motor(■, ■)
```

BBC micro:bit

pin15.read_digital

Auf der Linie bleiben



Mu 1.1.1 - LinienFolger.py

Modus Neu Laden Speichern Aufspielen Dateien REPL Plotter Hineinzoomen Rauszoomen Thema Prüfen Tidy Hilfe Beenden

LinienFolger.py

```
1 from Freenove_Micro_Rover import *
2
3 Rover = Micro_Rover()
4
5 while True:
6     if pin15.read_digital() == 1:
7         Rover.motor(40, 40)
8     elif pin14.read_digital() == 1:
9         Rover.motor(0, 40)
10    elif pin16.read_digital() == 1:
11        Rover.motor(40, 0)
```

BBC micro:bit

pin15.read_digital

Vielen Dank

- Infos zur Projektseite und Kontakte etc.
- .digital

.digital