



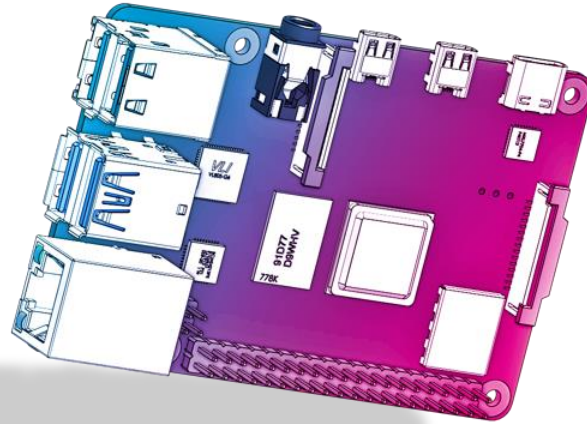
How to build your robot

www.pib.rocks/build

instructions for:

Start Up

pib#4s



PRINT

BUILD

DEVELOP

YOUR OWN ROBOT

Build it better: our suggestion for assembling pib



We recommend **tools** for each step. These are a suggestion, you can of course also use other tools.



1-5

We have categorized each step according to its **difficulty** - from **1-5** (1 being the easiest, 5 the hardest)

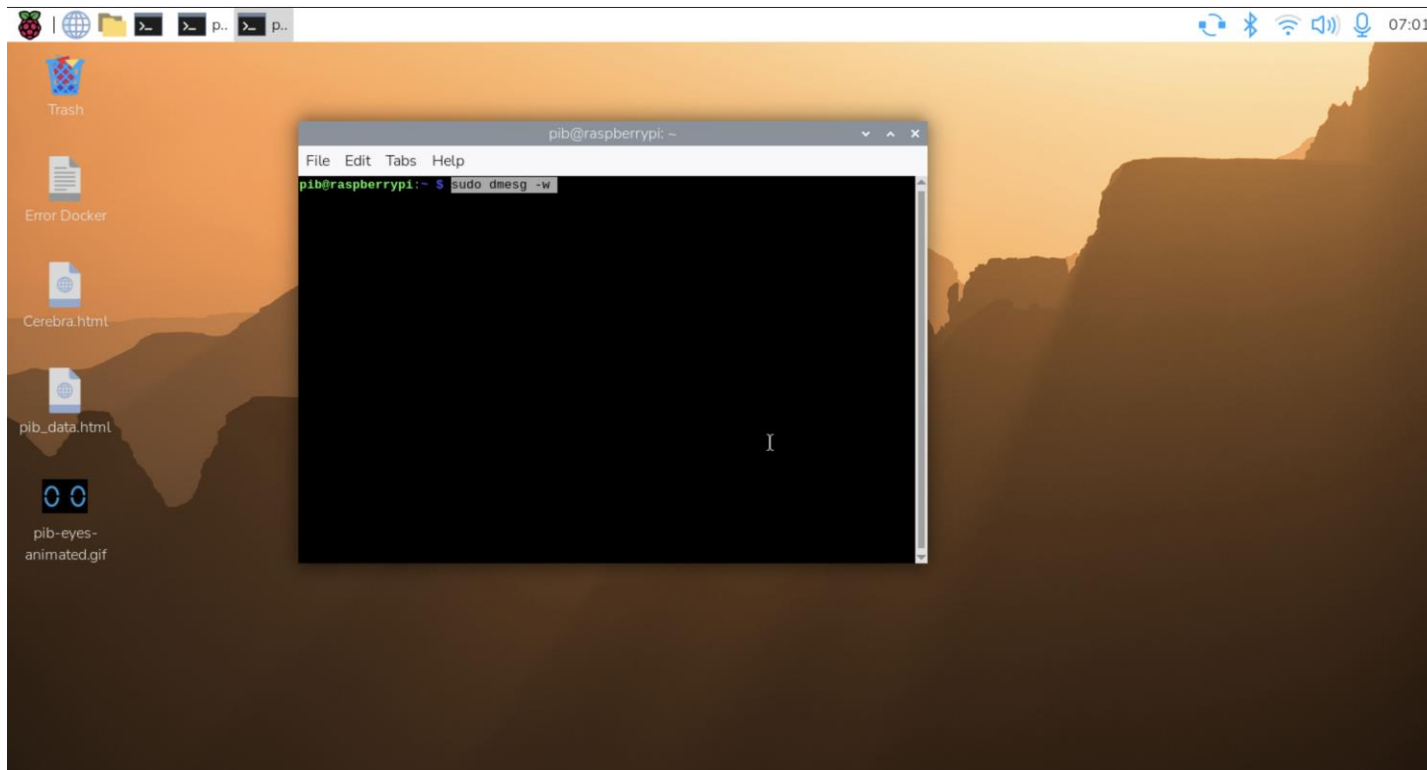


We also show you which **non-printable parts** you need for each step

Step 1



Open a terminal window and type in this command "sudo dmesg -w"



Step 2



Plug in E17 bricklets with their numbered order. 0 first then 1 then 2 then 3. And monitor the serial number showed on the terminal. If you insert the bricklets with the correct order then 1 should correspond to ttyACM0, 2 to ttyACM1, 3 to ttyACM2 and 4 to ttyACM3. Write down serial number for each one on a piece of paper case sensitive

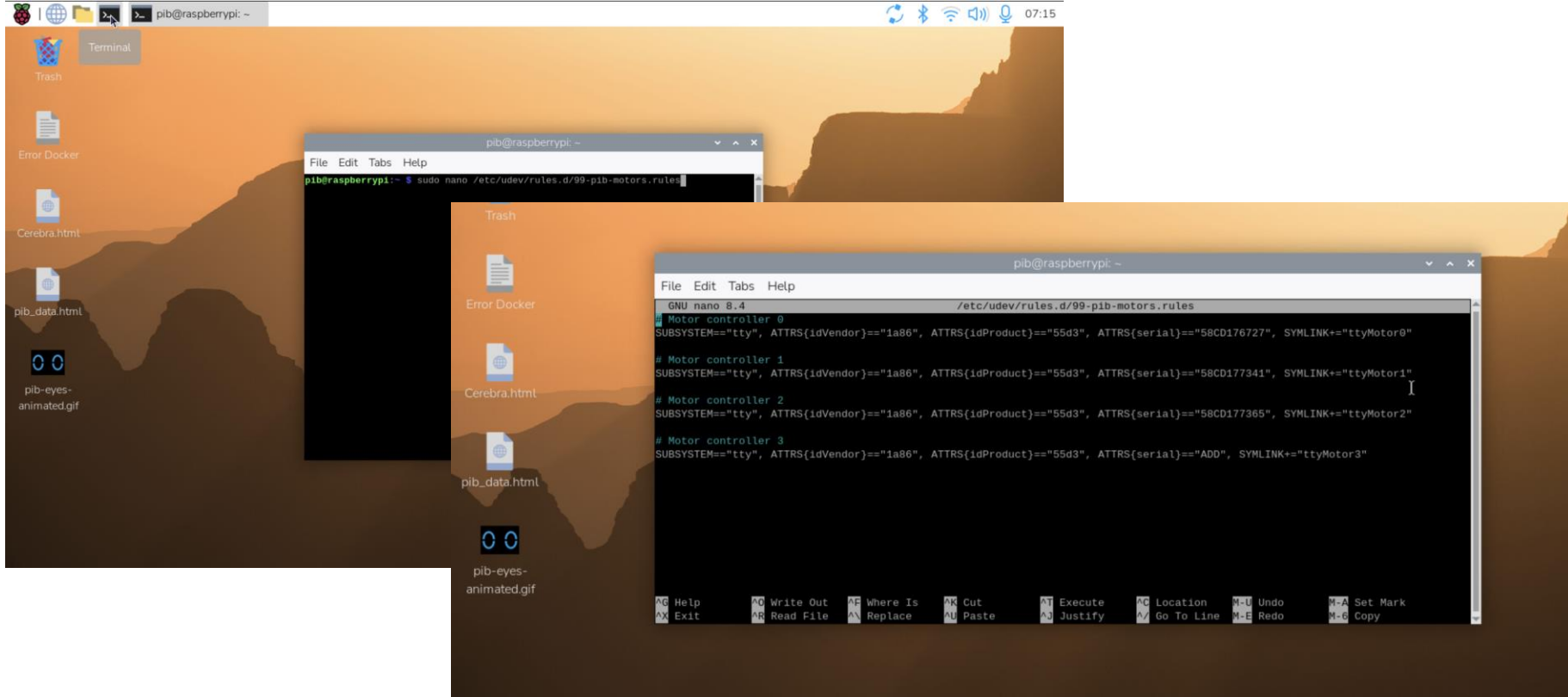
```

pi@raspberrypi: ~
File Edit Tabs Help
000.pc1e/1f00300000.usb/xhci-hcd.1/usb3/3-2/3-2.1/3-2.1.0/0003:0460:C063.0004:
input/input11
[ 422.661180] hid-generic 0003:0460:C063.0004: Input,hidraw2: USB HID v1.10 Mou
se [DELL DELL USB Laser Mouse] on usb-xhci-hcd.1-2.1/input0
[ 466.249867] usb 3-2.3: USB disconnect, device number 9
[ 500.042213] usb 3-2.1: USB disconnect, device number 11
[ 501.066277] usb 3-2.2: USB disconnect, device number 8
[ 527.659445] usb 3-2.1: new full-speed USB device number 12 using xhci-hcd
[ 527.784029] usb 3-2.1: New USB device found, idVendor=1a86, idProduct=55d3, b
cdDevice= 4.45
[ 527.784034] usb 3-2.1: New USB device strings: Mfr=0, Product=2, SerialNumber
=3
[ 527.784037] usb 3-2.1: Product: USB Single Serial
[ 527.784039] usb 3-2.1: SerialNumber: 58CD176727
[ 527.819717] cdc_acm 3-2.1:1.0: ttyACM0: USB ACM device
[ 530.731392] usb 3-2.3: new full-speed USB device number 13 using xhci-hcd
[ 530.852064] usb 3-2.3: New USB device found, idVendor=1a86, idProduct=55d3, b
cdDevice= 4.45
[ 530.852070] usb 3-2.3: New USB device strings: Mfr=0, Product=2, SerialNumber
=3
[ 530.852073] usb 3-2.3: Product: USB Single Serial
[ 530.852074] usb 3-2.3: SerialNumber: 5860177341
[ 530.891912] cdc_acm 3-2.3:1.0: ttyACM2: USB ACM device
    
```

Step 3



Create custom rules by typing this command "sudo nano /etc/udev/rules.d/99-pib-motors.rules" in terminal and add the shown lines exactly but with the serial numbers discovered in previous step. ttyMotor0 should correspond to bricklet 1, ttyMotor1 to 2 and so on



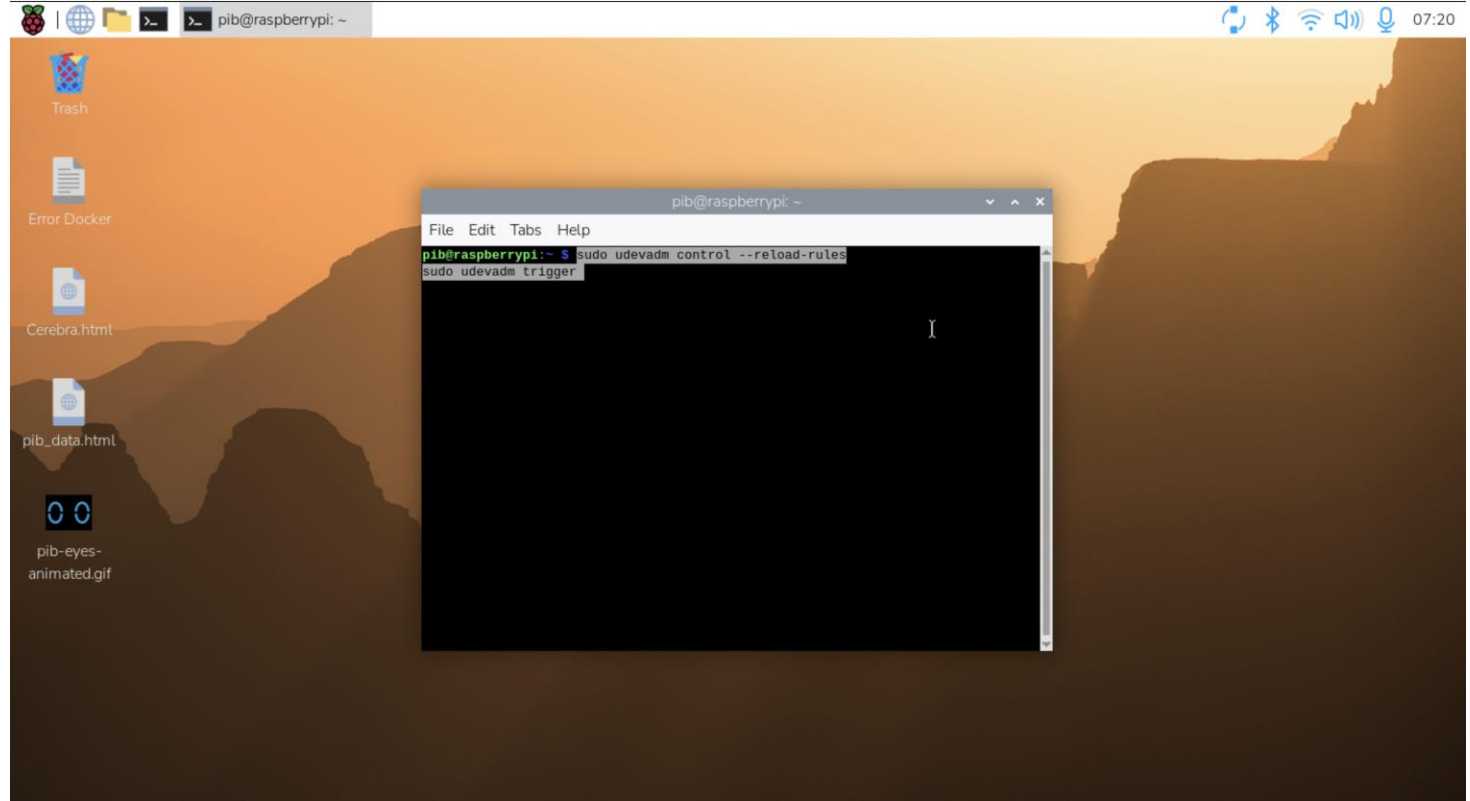
Step 4



Type the following commands in a terminal, if it doesn't work instantously then unplug usbs, replug them and restart pi**b**.

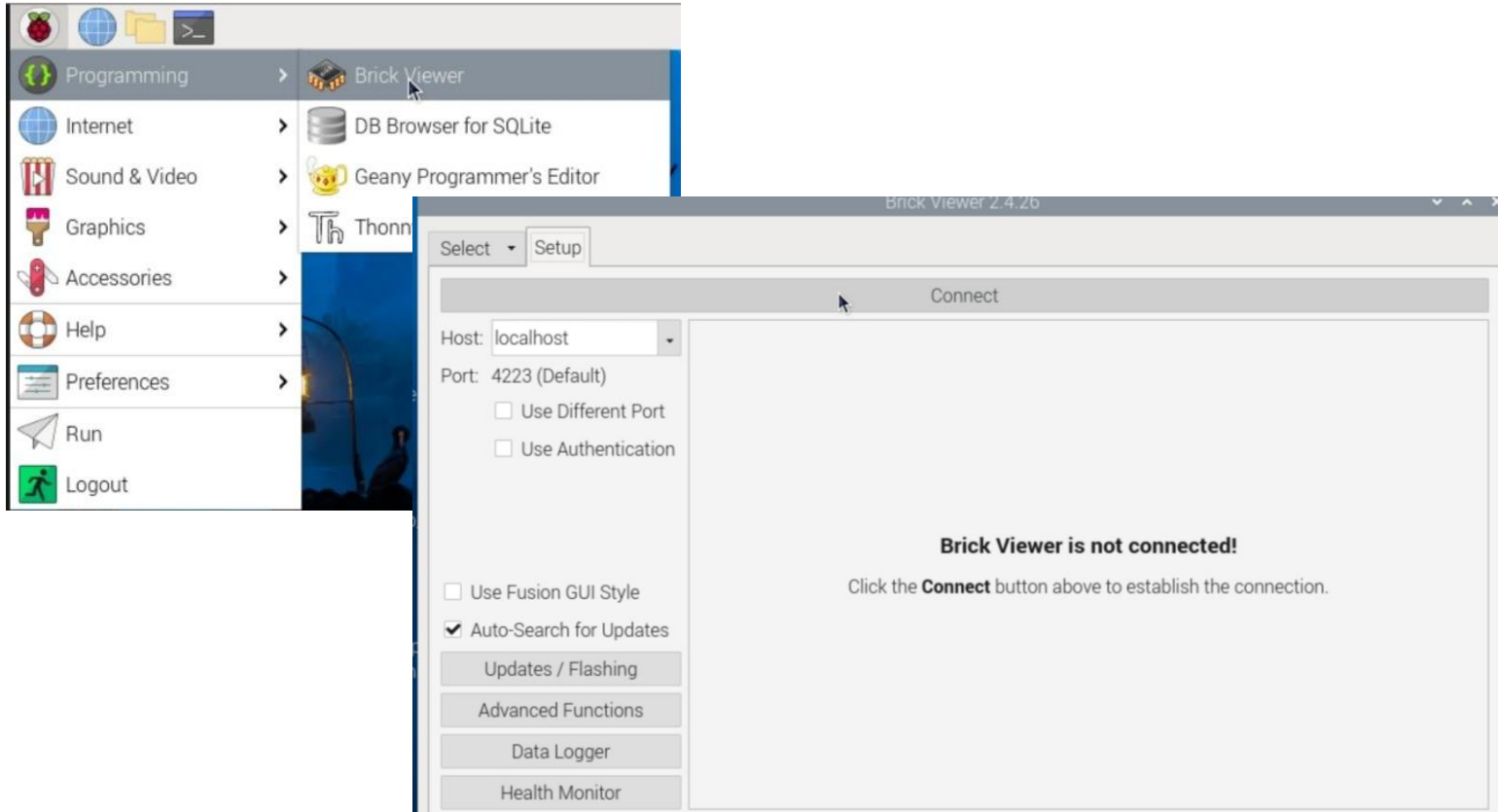
```
sudo udevadm control --reload-rules
```

```
sudo udevadm trigger
```



Step 5

Open brickviewer and click on connect

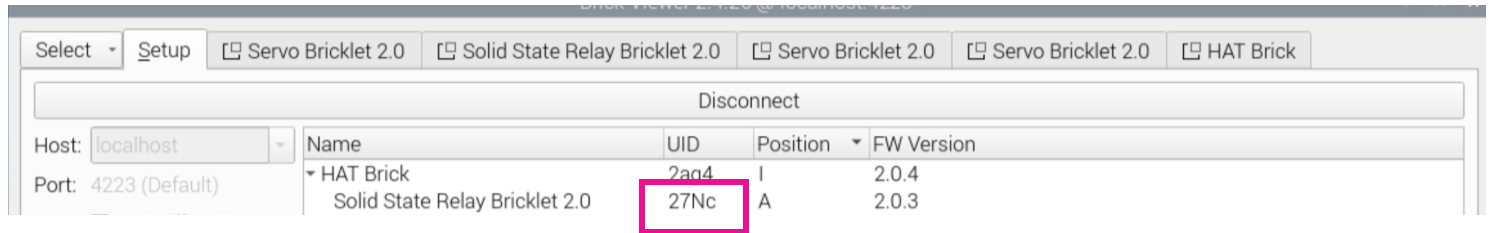


Step 6



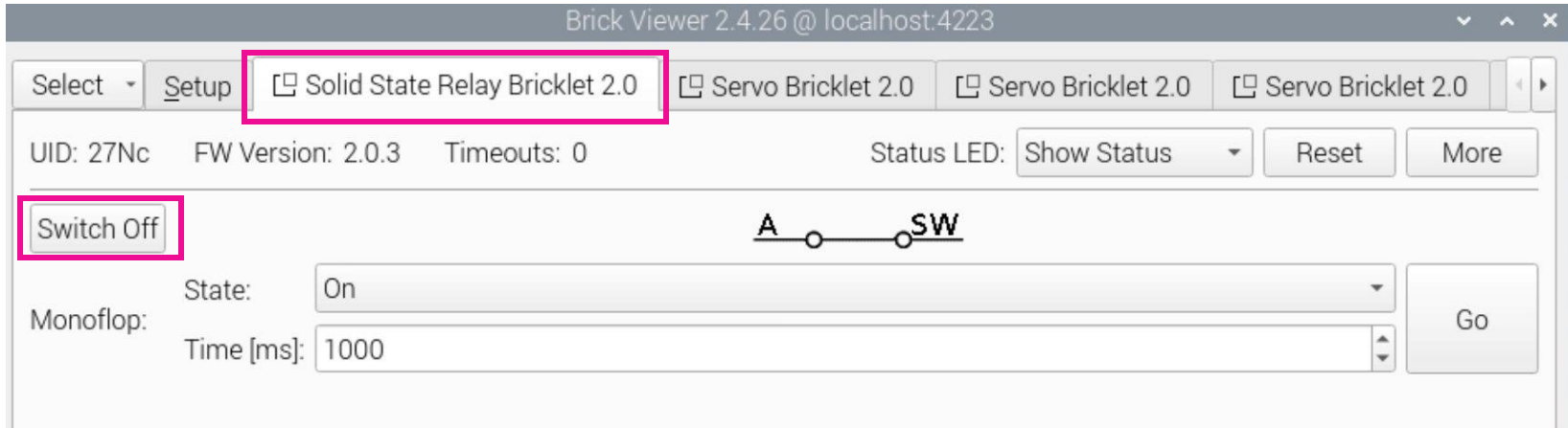
1

Write the UID beside solid state relay in a piece of paper (case sensitive)



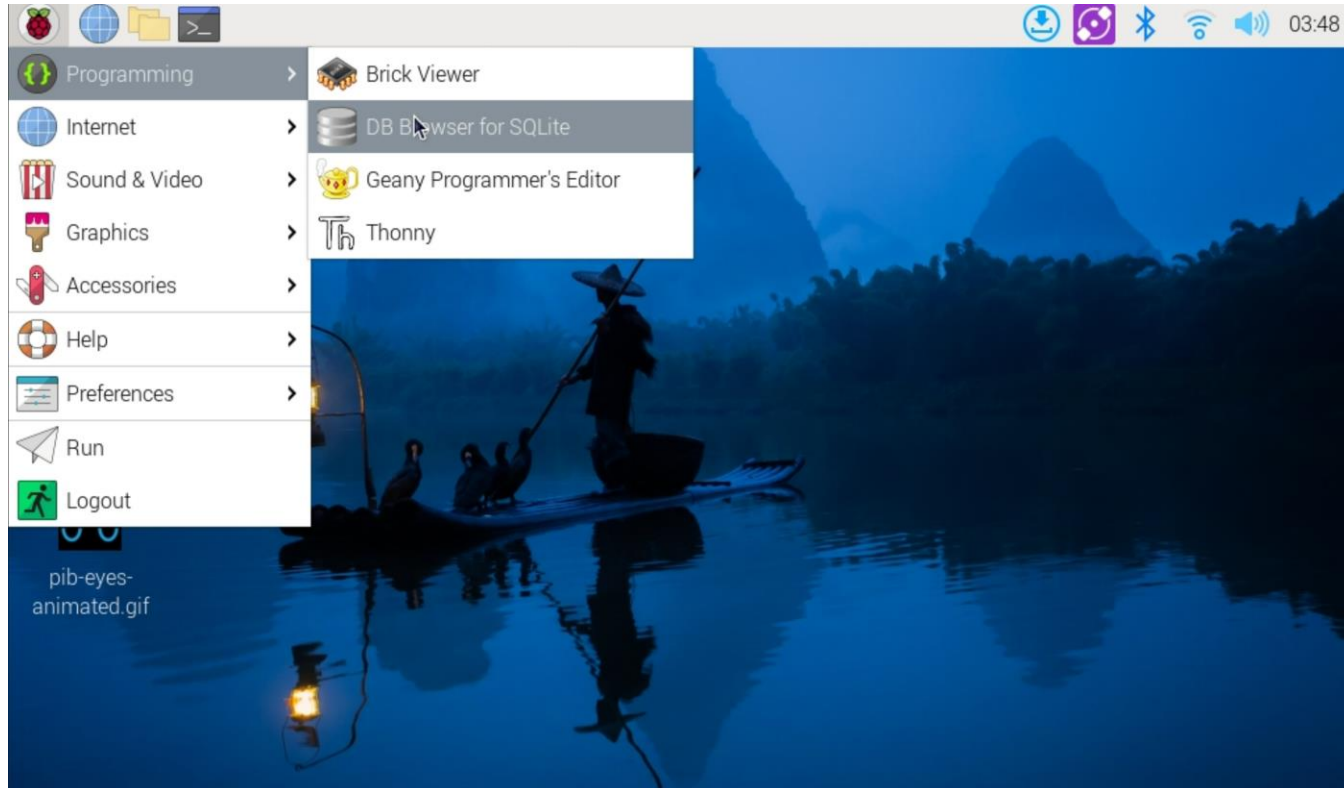
Note

To turn on and off the power for motors without Cerebra click on the shown button in solid state relay bricklet tab



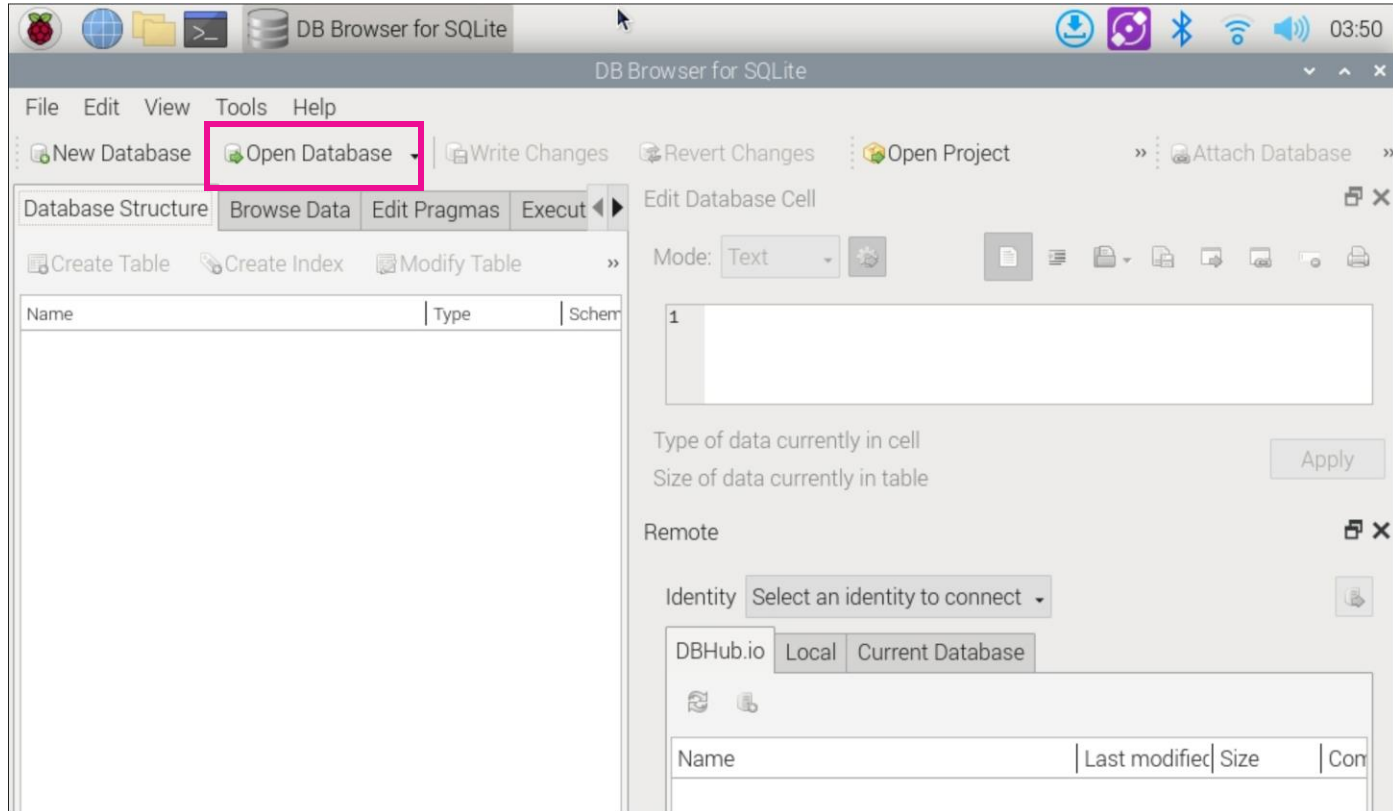
Step 7

Click on the top left Raspberry icon, then point at programming and select DB browser for SQLite



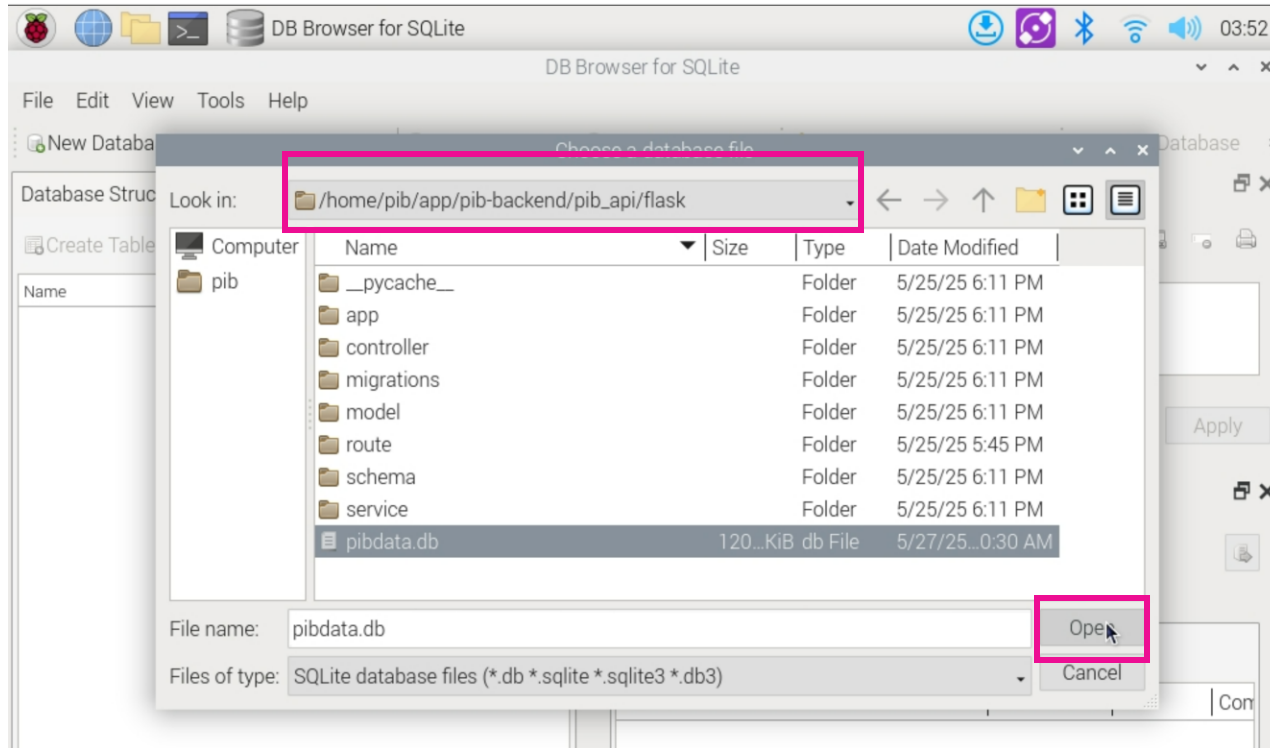
Step 8

Click on open database



Step 9

Navigate to `app/pi-backend/pi-backend/pi_api/flask` and select `pi_data.db`



Step 10

Click on browse data



The screenshot shows the DB Browser for SQLite application. The 'Browse Data' button is highlighted with a red box. The interface includes a menu bar (File, Edit, View, Tools, Help), a toolbar with options like 'New Database', 'Open Database', 'Write Changes', 'Revert Changes', 'Open Project', and 'Attach Database'. Below the toolbar is a 'Database Structure' panel with buttons for 'Create Table', 'Create Index', and 'Modify Table'. A table lists 12 tables with columns for Name, Type, and Schema. The 'Edit Database Cell' window is open, showing a text input field with the number '1'. Below the input field are labels for 'Type of data currently in cell' and 'Size of data currently in table', and an 'Apply' button. A 'Remote' section is also visible, showing an 'Identity' dropdown menu and a table with columns for Name, Last modified, Size, and Con.

Name	Type	Schema
alembic_version	CREATE	
assistant_model	CREATE	
bricklet	CREATE	
brickletPin	CREATE	
cameraSettings	CREATE	
chat	CREATE	
chatMessage	CREATE	
motor	CREATE	
motor_position	CREATE	
personality	CREATE	
pose	CREATE	
program	CREATE	
Indices (0)		
Views (0)		
Triggers (0)		

Step 11

Go to bricklet and change uid for relay to the one discovered in brickviewer

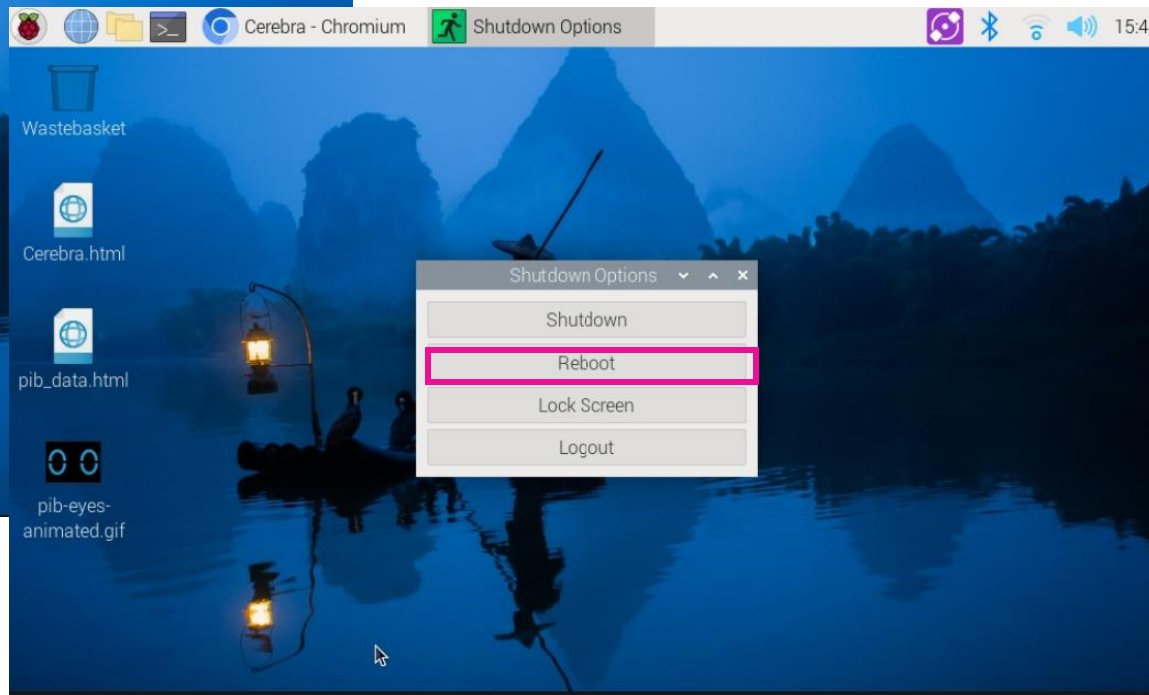
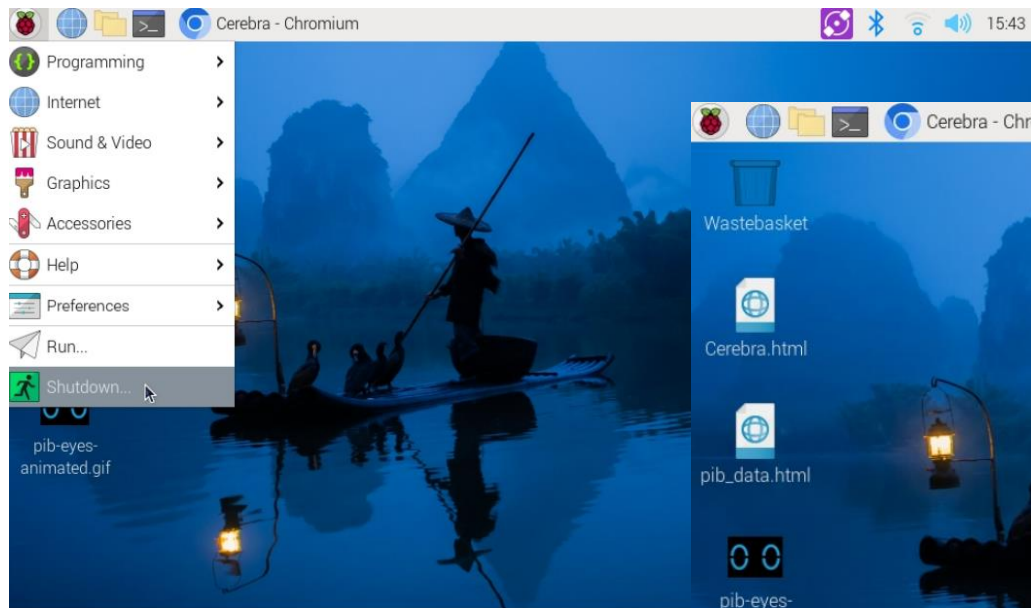
The screenshot shows the DB Browser for SQLite interface. The 'Table:' dropdown menu is set to 'bricklet'. The table data is as follows:

id	uid	bricklet_number	type
1	/dev/ttyMotor0	1	Servo Bricklet
2	/dev/ttyMotor1	2	Servo Bricklet
3	/dev/ttyMotor2	3	Servo Bricklet
4	/dev/ttyMotor3	4	Servo Bricklet
5	27Mf	5	Solid State Relay Bricklet

The 'uid' cell for row 5 is selected in the 'Edit Database Cell' window, and its value '27Mf' is being edited. The status bar indicates 'Editing row=5, column=2'.

Step 12

Restart pib



Feature 1

Click on joint control and select any joint you want to move and just move the slider



The screenshot shows a web browser window with the following details:

- Browser: Cerebra - Chromium
- Address Bar: localhost/joint-control/head/motor/turn-head
- Page Title: Cerebra
- Page Content:
 - Left Sidebar: JOINT CONTROL (selected), POSES, CAMERA, VOICE ASSISTANT, PROGRAM.
 - Main Area: RIGHT ARM, HARDWARE-IDS, Select a touch point to adjust.
 - Table:

Settings	Active	Current
	<input checked="" type="checkbox"/>	0 mA
	<input type="checkbox"/>	0 mA
 - Head Rotation slider: -90, 0, 90.
 - Bottom: Smart connect button.

Note



If a finger moves in the opposite direction to others, click on the finger, click on settings icon, click on extended values and click invert

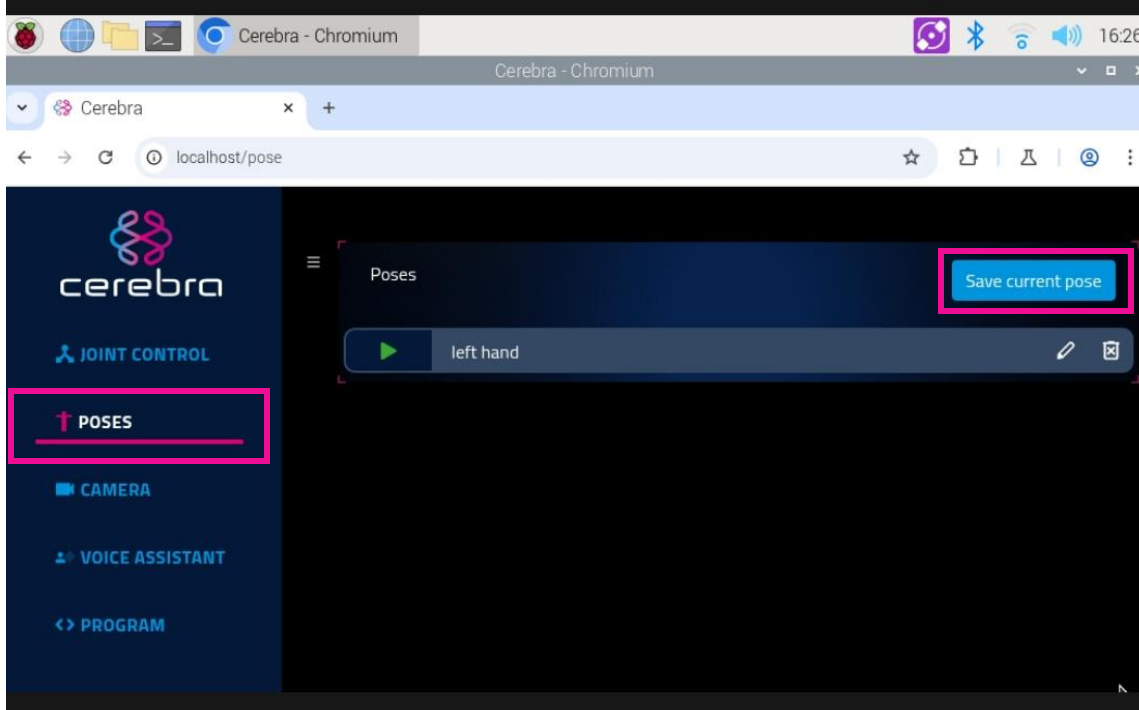
The screenshot shows the Cerebra web interface in a Chromium browser. The main page is titled 'Cerebra - Chromium' and shows the 'RIGHT HAND' settings. A gear icon for settings is highlighted in pink. An inset window titled 'MOTOR SETTINGS' is open, showing the 'EXTENDED VALUES' tab selected and highlighted in pink. The inset window displays 'Pulse width (µs)' at 750 and 'Degree (°/100)' at -90.

This close-up screenshot shows the 'MOTOR SETTINGS' dialog with the 'Degree (°/100)' slider set to -90. The 'Invert motor' checkbox is highlighted in pink.

Feature 2



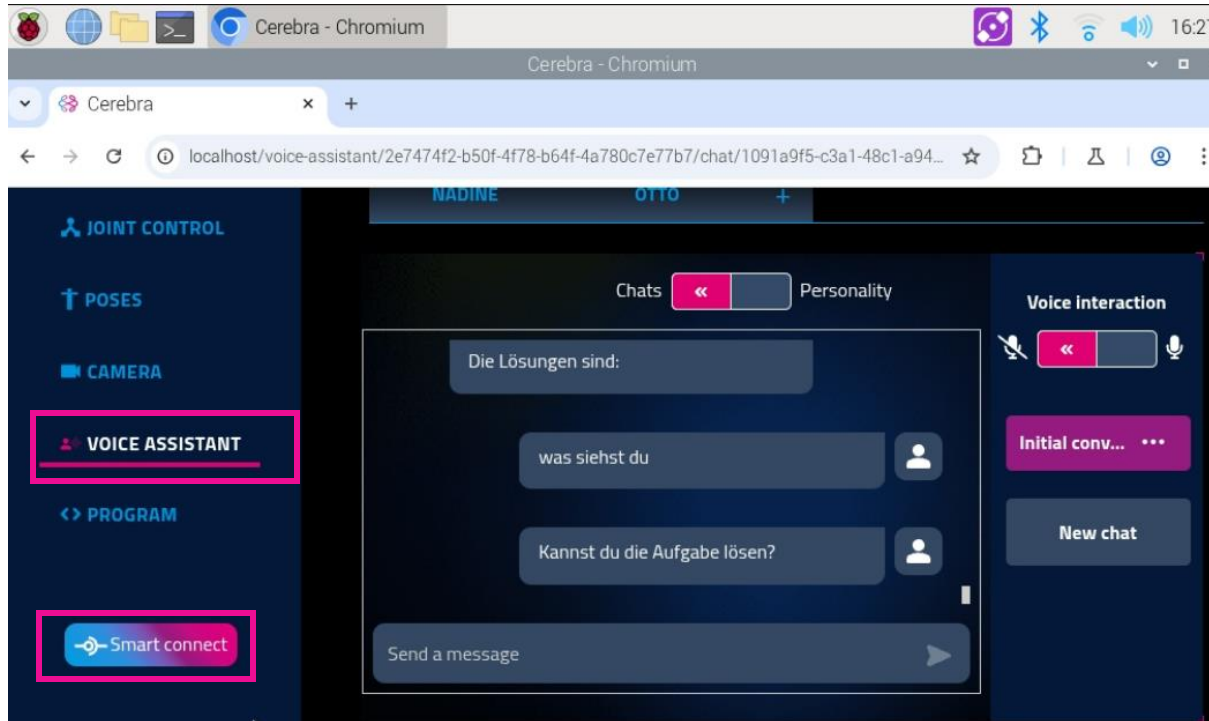
Move the joints with joint control to a configuration you want to save, then navigat to poses and create new pose. Now every time you click on this pose, pi**b** will go to the saved configuration



Feature 3



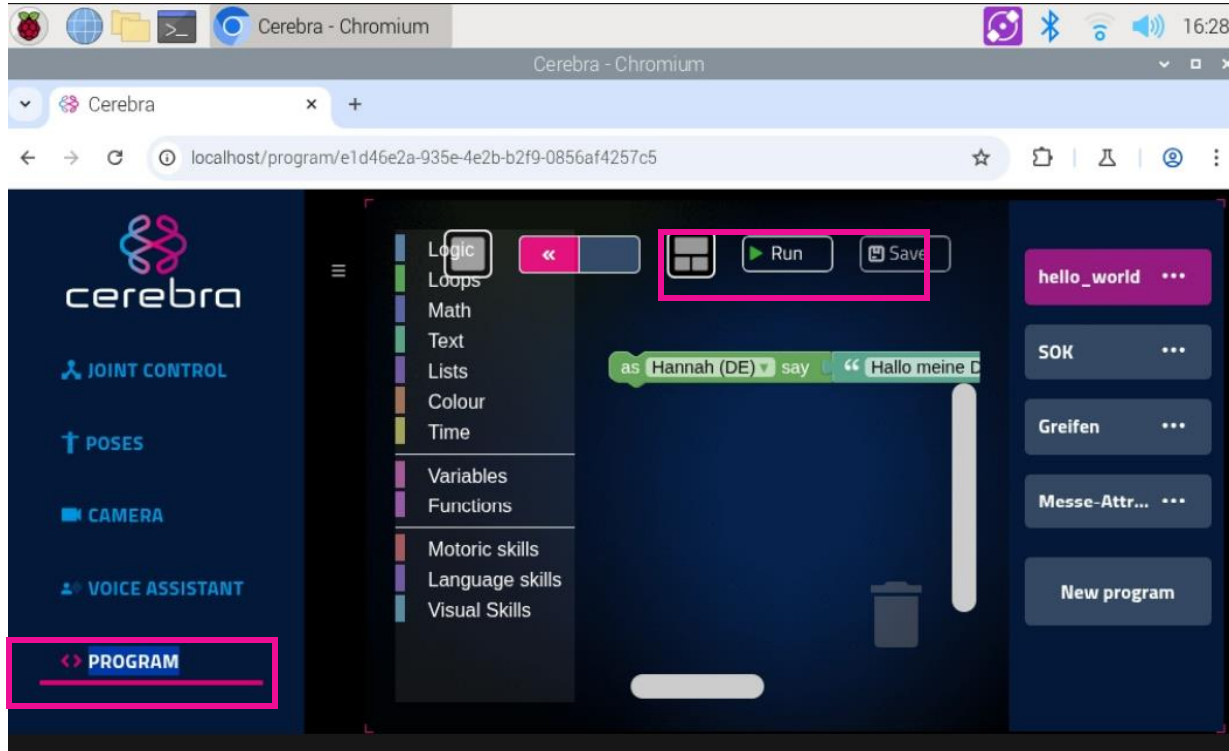
Click on smart connect API, get in touch with pib team at team@pib.rocks to provide you with a token, enter this token, a new password and start talking with pib



Feature 4

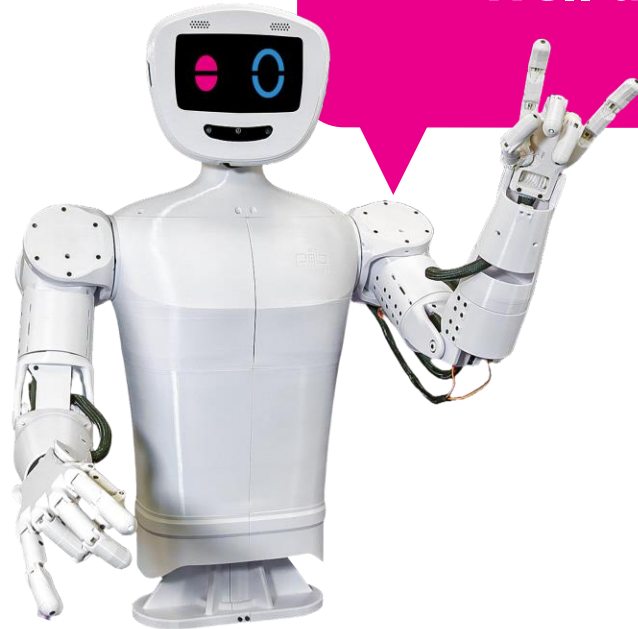


Click on block programming to create a script with basic blocks, then click run to start it



Congratulations

You did a great job, you have started pi**b**!



Well done!

Do you need support?

Or do you need our pib.Box with all non-printable parts?

Or maybe you have some new ideas and improvements?

Please contact us.



team@pib.rocks
Send us an email.



discord.com/invite/GRdpyeDu7P
Join us on Discord.



shop.pib.rocks
Order non-printable parts for pib.