Smart Violin

INSTRUCTABLES

DE MEYER HANNE

Smart violin

Music is an important part of my life. I have been playing the violin for 10 years, but there is one problem. I never know how long I practiced for. That's why I made this project. In my project I will keep track of the temperature and humidity in the violin case and how long I practised for.. It's a stand alone project, but I also build a website that shows the temperature, humidity and practice time. In this instructables I will show you how I made the violin case and how to code it.

What electronics do you need

I listed everything that you need to make this project yourself. I paid a total of €130,55 for the electronics. Maybe you still have old electronics where you can get components from, or maybe you have some parts from previous projects.

On the instructable you can download a pdf where you will find a link to order these electronics.

Basics:

- Raspberry Pi 4
- Raspberry Pi USB-C 3A
- Micro SD-card (+/- 16GB)
- Breadboard (x2)
- Breadboard power supply 9V
- T-cobbler
- 40 pin extension board adapter

Sensors:

- DHT11
- Push button (x3)

Actuator:

- Electromagnet ZYE1-P20/15

Other:

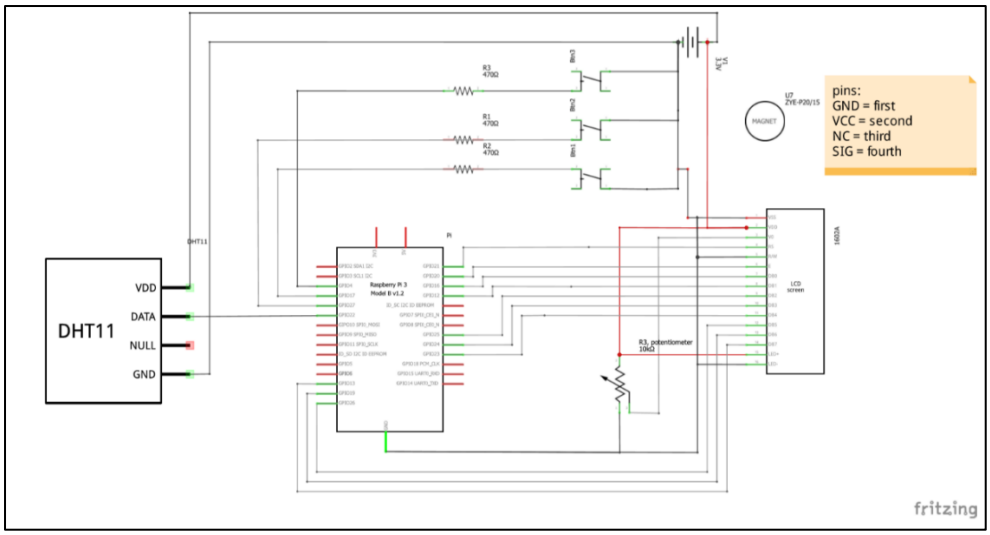
- LCD display 1602A
- Resistor 220 Ohm (x3)
- Male-to-male cables
- Male-to-female cables

Frizting scheme

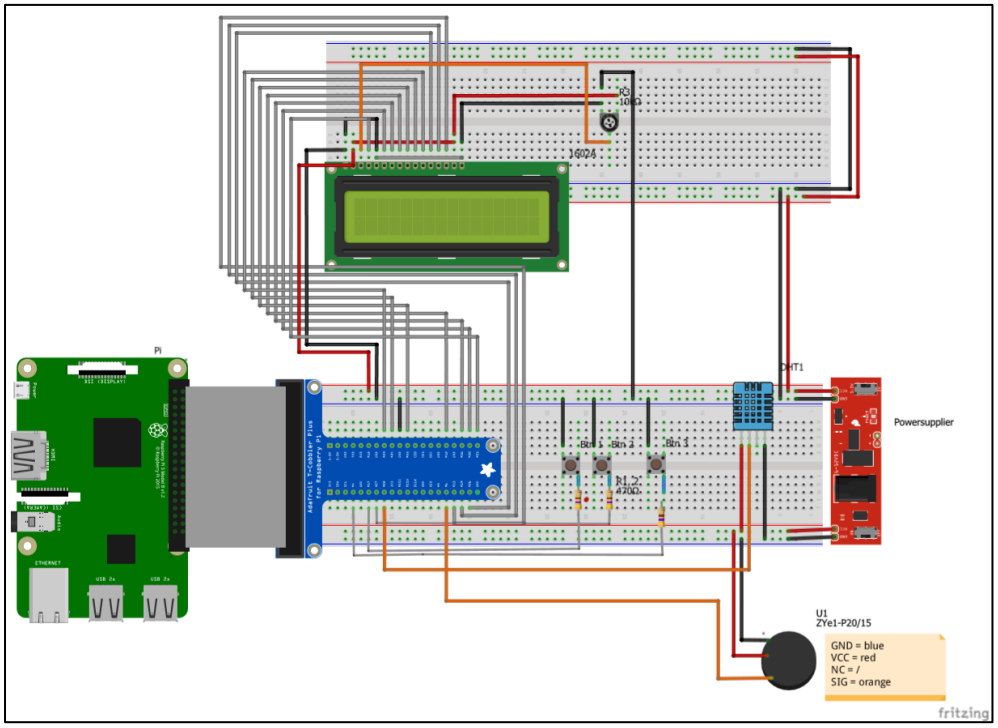
I made two frizting schemes. The first scheme is how I'm going to switch it and the second one is the layout on the breadboard.

I soldered all the buttons. Do not forget to place a 220 Ohm resistor with each button. This is for safety reasons if you switch it wrong. I attached to LCD display to the breadboard with maile-to-female cables. The potentiometer is switched on the breadboard.

1. Elektronics



2. Breadboard

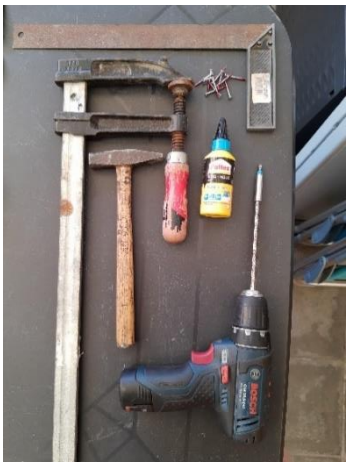


What tools do you need

You need different tools to build the case. I have listed them all for you.

Tools:

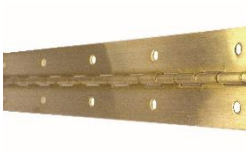
- Screwdriver
- Drill
- Milling machine
- Wood glue
- Square tool
- Hammer
- Screw clamp
- Cyllinder drill
- Jigsaw



Of course you can't use tools if you have nothing to attach your case with. That is why I have listed all components.

Tools:

- Screws
- Ducktape
- Tape
- Piano hinge (100 cm)
- Wood (see step 4)
- Gas spring 50N/5kg 250mm
- Fastener toggle latch (x2)

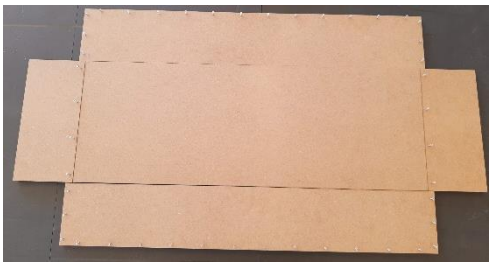


Product development

You need a couple of different sizes of wood. The thickness of the wood is 1,8 cm.

Outside of the case:

- Back face = 98,6 x 16,0 x 1, 8 cm
- Front face = 98,6 x 16, 0 x 1,5 cm
- Down face = 95,0 x 34,0 x 1,8
- Up face = 98,6 x 37,8 x 1,8
- Left Face = 16,0 x 34,0 x 1,8 cm
- Right face = 16,0 x 34,0 x 1,8 cm



Inside of the case:

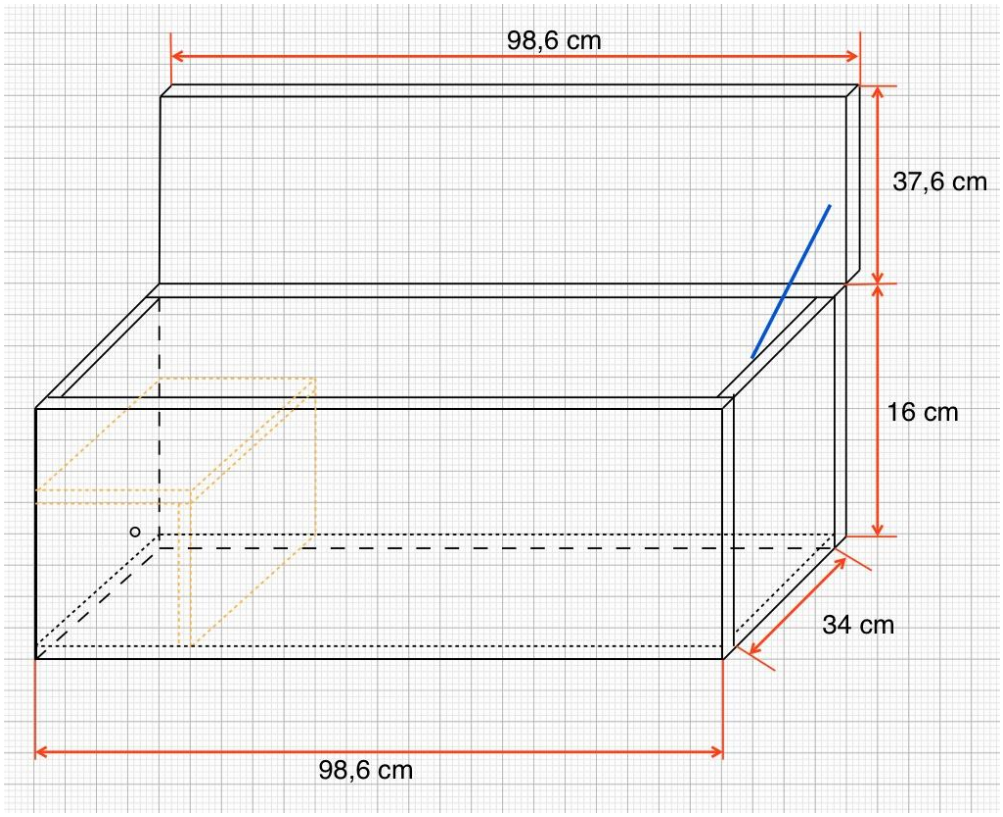
- Up face = 20,0 x 34,0 cm
- Left face = 11,0 x 34,0 cm
- Grid = 34,0 x 2,5 cm
- Support blocks = 8,0 x 34x0 cm
- Magnet wood = 8,0 x 4,0 cm

I also cut out the shape of my violin from isomo so that it stays in place in the box.

- Isomo = 71,5 x 34,5 cm
- Wood = 71,5 x 34,5 cm

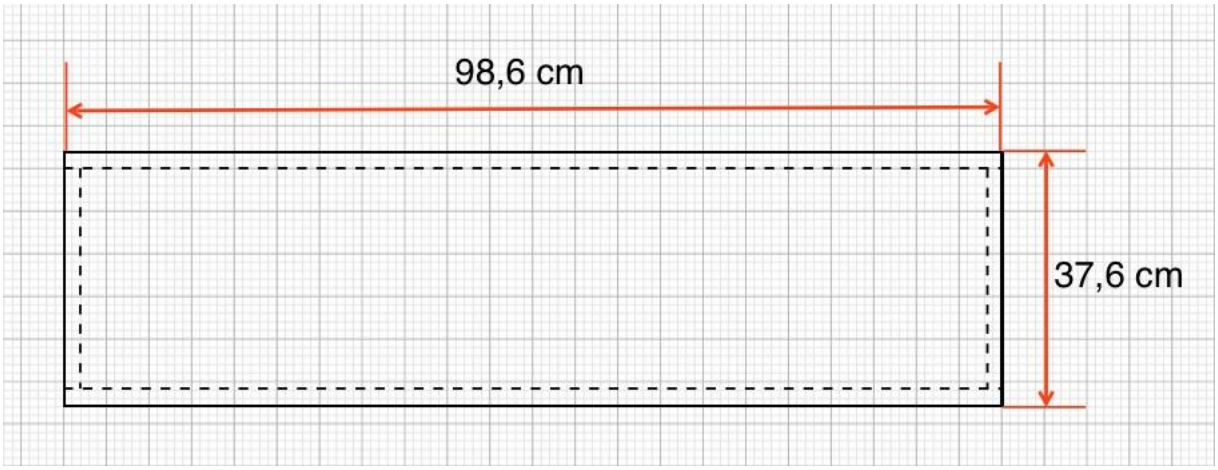
Front face

The blue lines are the gas spring.The orange lines form the box for the elektronics.



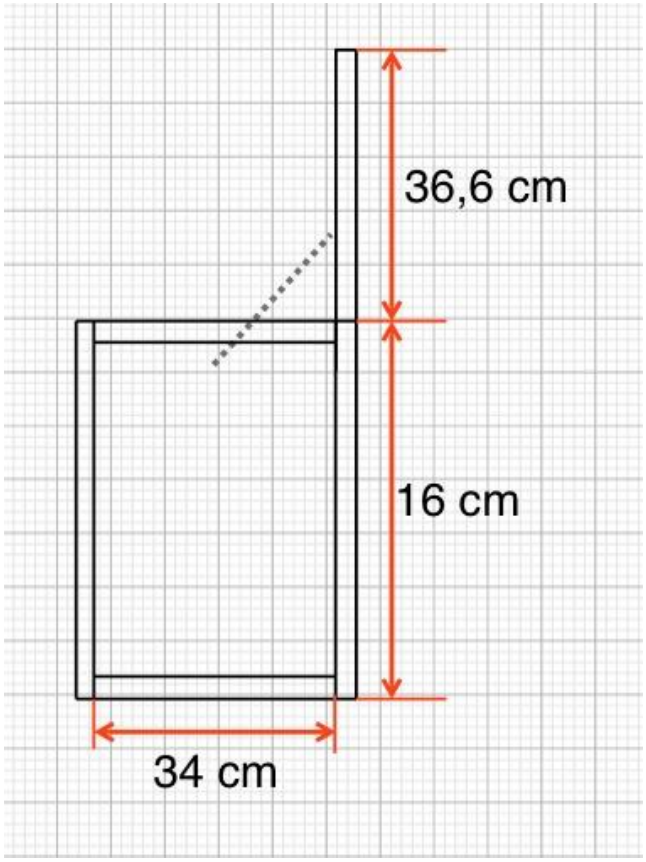
Up face

the dotted lines show how the wooden plates should be underneath



Right face

Watch out for the lid. When you hang the gas springs, the lid can no longer be 90°.

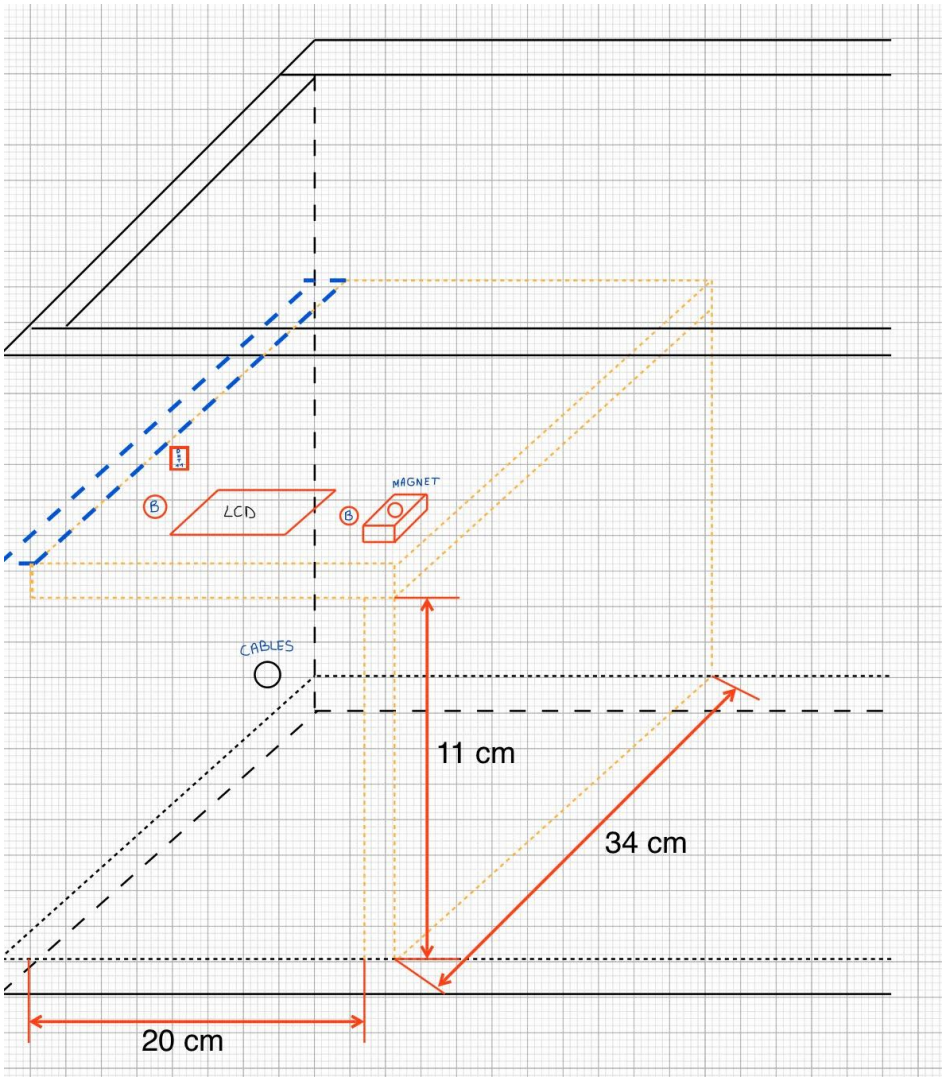


Close up elektronica compartment

I made some sketches of how I want to place my sensors and actuator. You can choose the layout of the elektronic components such as the lcd, DHT11… I chose to put them at the bottom. you can attach it with screws, but I still had to reach my pi easily. That's why I attached extra support blocks so that I could easily take it out. I put them just where the up face of the elektronic compartment ended.s

I had some trouble with the magnet, but after trying some things it worked out. I cut an extra piece of wood and I milled it out. The dimensions are 8,0 x 4,0 cm. On the lid I attached two washers with a screw in the same place as the magnet that is attached to the shelf with all electronics components.

I have attached a grille (blue part) on the left side for ventilation. You can also drill several small holes on the right side where the cable hole is to have ventilation to the outside.



I will explain step by step how I put the case together.

Step 1

Connect the front and back face with the down face. I used 13 screws and wood glue. Then I attached the sides with 4 screws and wood glue. Drill a hole for wiring the electronics in the left side.

Step 2

The back is 98.6 cm long. I also cutthe piano hinge that long; so I can attach it full length at the back inside. First I attached it to top of the back face. To attach it to the lid, it's best to ask someone for help. You proceed in the same way as the previous one.

Step 3

Attach the fastener toggle latches. Choose where you place them. I placed them 20 cm from the edge. Both left and right. They should be easy to attach. Then the outside of the suitcase is finished.

Step 4

I had to be able to put the electronics somewhere. I signed off everything on the top plank. Then I milled everything out. It's a precise job, but it gives the best result. The magnet was a problem, but I solved it by placing an extra wooden block where the magnet should be. With a cylinder drill of 20 you can make a hole in the wooden block. The magnet will fit exactly.

Attach the side and top by screwing them together and don't forget to put wood glue in between. The component plank will not fit because the gas spring is in the way. Somehow I needed ventilation. I attached a grid of 34,0 x 2,5 cm on the left to the top plank.I still had to reach my RPi for another project, so I didn't screw it down to the outside of the case. I made two support blocks on which the inner shelves can lie.

Hang the two support blocks where the font face starts. So the side closest to the bottom. At the top is a photo to make it clearer. Now you can put the elektronics and the compartment in there.

Step 5

Now the violin case is almost finished. A place has yet to be made for the violin. Place your violin on the wooden plank and draw it over. I did not copy the piece for the neck. This way the violin gets some support. Once you have drawn it, you can now cut out the shape with a jigsaw.

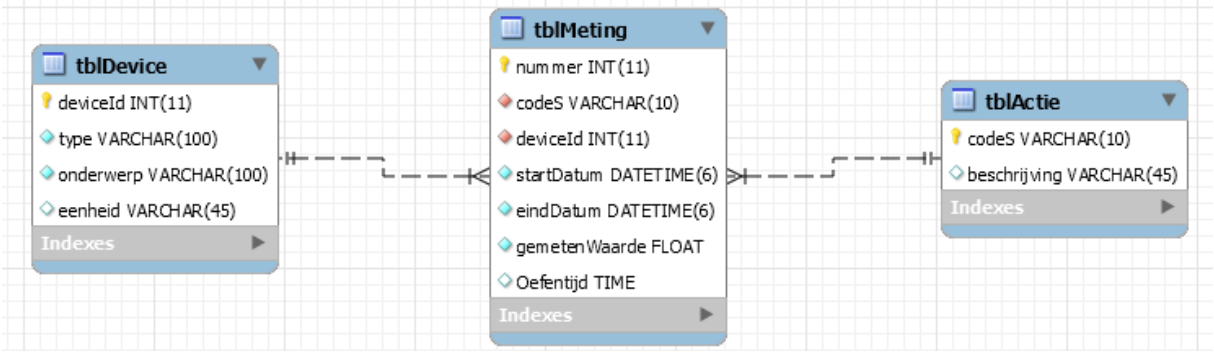
After I cut it out, Iplaced the wood on the isomo and trace the shape. THen I cut out the violin shape with a utility knife. If you want, you can paint the wooden plank. I worked with black graffiti. Be careful not to use this on the isomo, it will absorb the paint! When its dry, you can put everything in de case.

Finished product

This is the finished product. You can paint the case, but do that before you attach everything together. That gives the best result.

Normalized database structure

I made my 3NF database in MySQL Workbench. It's a very easy program to work with.



I made my database in Dutch, but I'm going to translate it to English so everyone can understsand it.

tblDevice	tblmeting	tblActie
deviceId (PK)	number (PK)	code (PK)
topic	code (FK)	description
unit	deviceId (FK)	
type	startDate	
	endDate	
	measuredValue	
	practiseTime	

Setting up the Rapsberry Pi

Necessities

Hardware

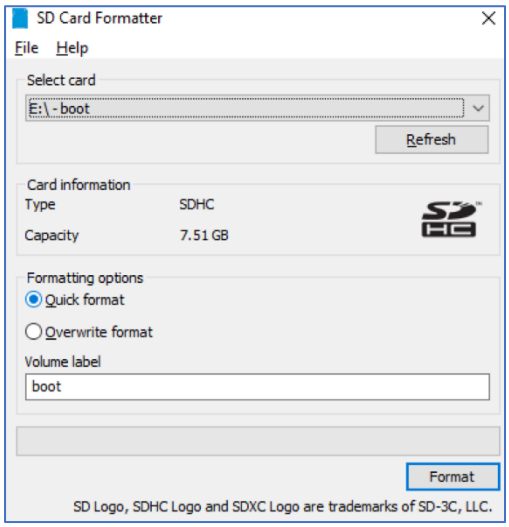
- Raspberry Pi 3/4
- Micro-SD card (8GB or more)
- SD card reader
- Network cable

Software

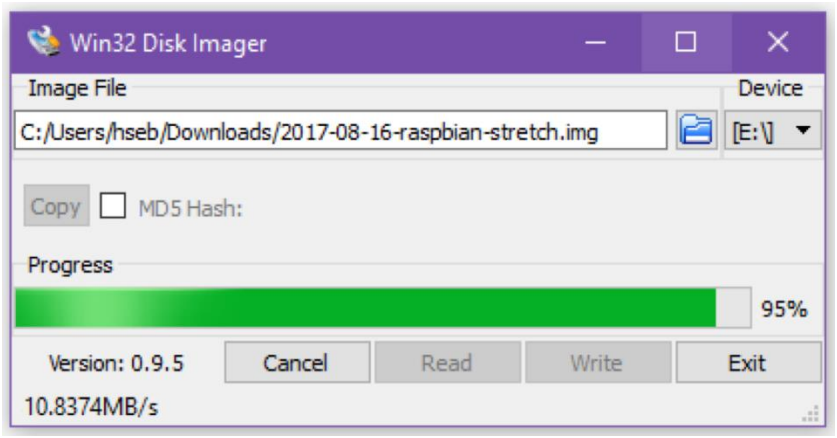
- SD-card formatter (https://www.sdcard.org/downloads/formatter/eula_windows/index.html)
- Raspbian OS image (with Desktop GUI) (https://www.raspberrypi.org/downloads/raspbian/)
- Win32-DiskImager (http://sourceforge.net/projects/win32diskimager) or Etcher (https://etcher.io/)
- puTTY (https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)
- WinSCP (https://winscp.net/eng/download.php) or FileZilla (https://filezilla-project.org)

Installation

If you use an SD card that is not new, you can reformat it with SD Card Formatter. Choose quick format and then click on Format.



raspbian is provided as a disk image. We need to write the content directly to the SD card. This is very easy via Win32DiskImager. Find the correct folder and click write. This program is also useful for backing up your SD card from time to time.



I use an APIPA address on my RPi. We create a fallback address just to be sure. but before you can do that, you have to format the partition. If you get a warning, just click on format disk.

When viewing your boot partition, you should see the cmdline.txt and config.txt. These are two important files. I work with an APIPA address and I assign it in the file cmdline.txt.

1. open the cmdline.txt file
2. add: ip = 169.254.10.1, **but make sure everything is on 1 line**
3. save it

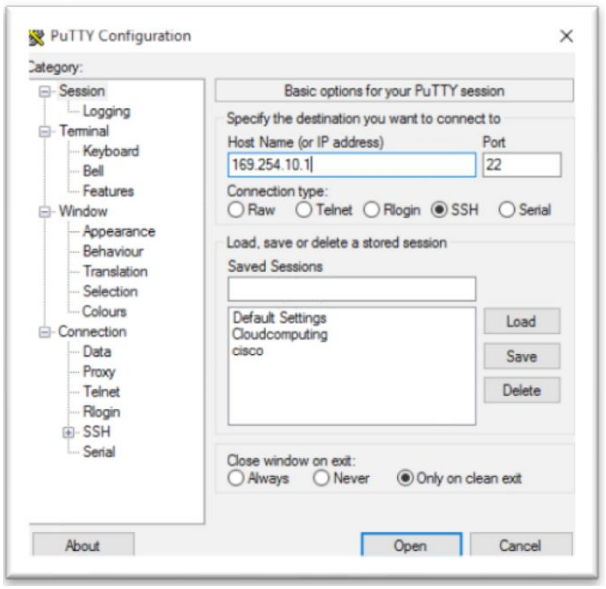
SSH

Go to the boot partition and create a file called ssh. **Note**: in lower case and no extension!

Safely remove your SD card from the PC and insert the micro SD card into the RPi.

connect your computer to the RPi via a network cable and then plug in the RPi power supply.

Open putty and enter the APIPA address as host name.



Setting up home network

Now you can add your home network. Execute the following command to see all available SSIDs

```
sudo iw dev wlan0 scan | grep SSID
```

Find the SSID of your home network and run the following command.

```
sudo wpa_passphrase “NameOfYourNetwork”
```

If you have run it, you will have to enter your password. copy the code from 'Network = {' to '}'

Run the following command. Now paste the code at the bottom of the file. Do ctrl x y enter.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Make sure you reboot your RPi with the following command.

```
sudo reboot
```

To make sure your network has been added you can run this command after rebooting. You should see the WLAN0 with an inet, netmask and broadcast address.

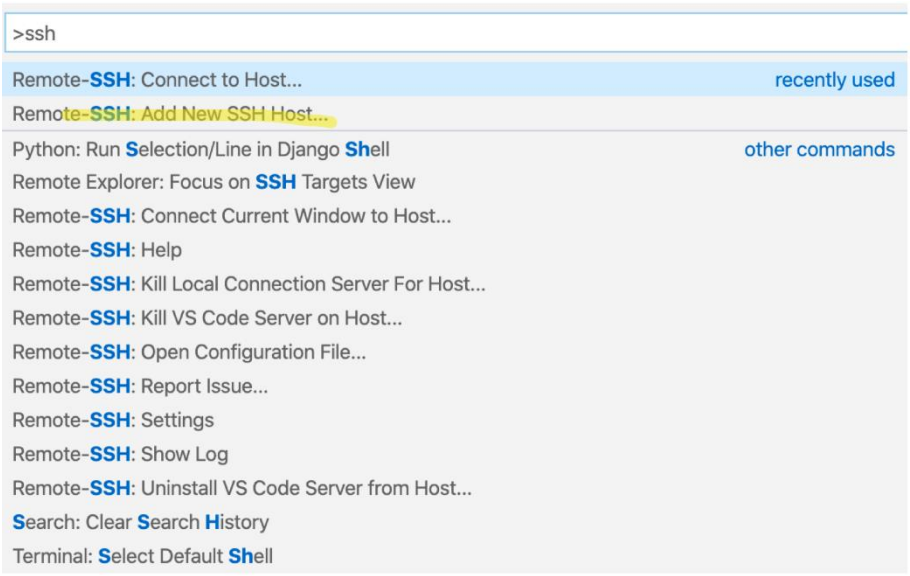
```
sudo ifconfig
```

Setting up Visual Studio Code

I use VSC to program. It is a very easy programming program. Now it has to be set correctly.

configure extensions

In VSC, press F1 and tap SSH. Choose Remote-SSH: Add New SSH Host.



Enter SSH Connection Command:

- Enter the following details: SSH PiUser@169.254.10.1

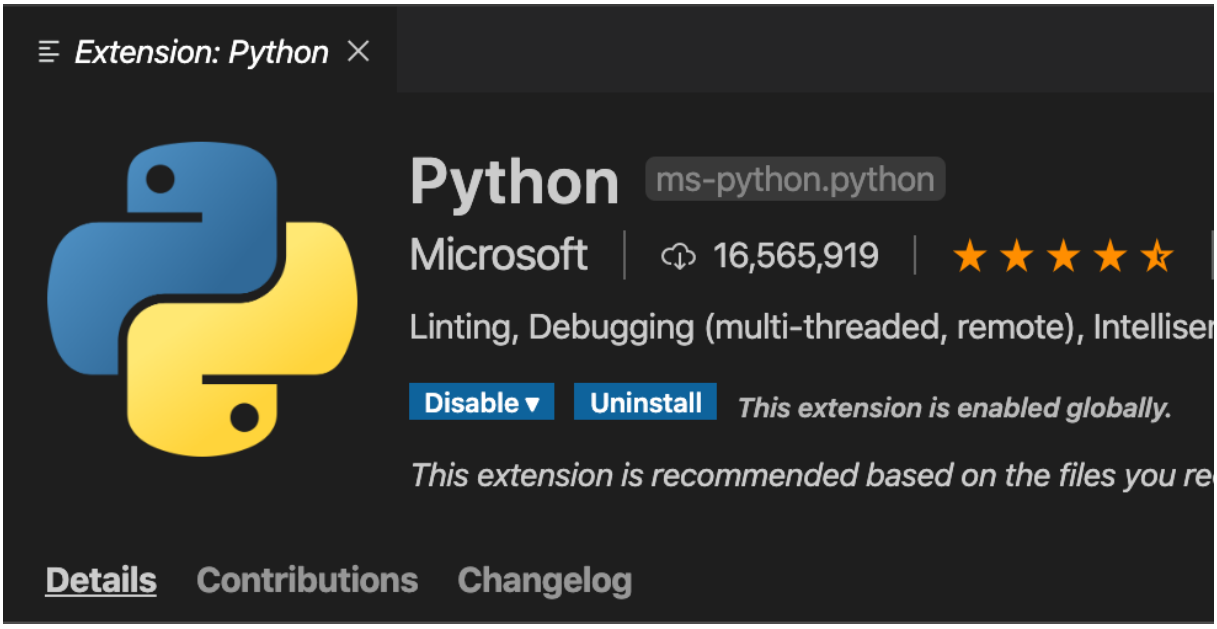
Select SSH configuration file to update:

- /Users/user/.ssh/config

Now everything is set up from the SSH. Now you can easily connect via SSH to your RPi. You can start an SSH connection by pressing F1, typing SSH and choosing Remote-SSH: Connect to Host. Choose your host and a new screen will open. Enter your password and you have made a connection.

Python extension

Install the python extension on the remote machine.



Permissions, files and folders

Open a new terminal and create a new folder on your remote system. I gave it the name project1, but of course you can choose what you call it.

```
mkdir project1
```

Open the folder you created. All documents for the backend end up there. We will also create a folder where we will insert the frontend. As a frontend we use Apache. My folder is under / var / www / html. Add the folder to the workspace by right clicking in the explorer of VSC and choosing Add folderto Workspace.

Get permissions by running the following commands in puTTY.

```
cd /var/www/html
```

```
sudo rm index.php
```

```
sudo chmod 777 /var/www/html
```

Backend

Create a file in the folder for the backend and run it. In the terminal we execute some commands that get packages.

```
pip3 intall mysql-connector-python
```

```
pip3 install-flask-socketio
```

```
pip3 install flask-cors
```

```
pip3 install gevent
```

```
pip3 install gevent-websocket
```

You can now run the program, but it will crash after a few seconds. Create a database in MySQL. Create a config.py file containing the following information. Then the error should be resolved.

config.py

pi > project1 > config.py > ...

```
[connector_python]
user = xxx
host = 127.0.0.1
port = 3306
password = xxx
database = vioolkofferdb

[application_config]
driver = 'SQL Server'
```

Code on Github

In the previous step we created a workspace with a folder for the frontend and a folder for the backend. They come in handy now. I write my backend code with python and my frontend consists of html, CSS and Javascript.

Backend

config.py

I start with the config.py file. This contains the information of your database. The hostname and password are the same as your login on MySQL. At database you put the name of your database.

projectDataRepository

In projectDataRepository I can perform all CRUD actions. CRUD stands for Create, Read, Update and Delete. I retrieve data from the database, I can update it or add new

data. If necessary I can also do a delete, but I don't use that in this project. I request data for the graphs and the last measurement. I also have 3 inserts to add the data coming from the sensors.

app.py

In this file I put the code to get data to my database, my electronics code and also the code to get things to my frontend or to receive things from the frontend.

At the bottom of the code I do everything. If you get errors from GPIO, take a look at the pins you used for your buttons, LCD ...

Frontend

app.js

The graphs for the site are created here. Connections are made to the backend, but things also come from the backend to the frontend.

index.html

This file contains all information for the website. Also links to the Javascript file.

screen.css

This is where the layout of the website is done.