Weatherstation Documentation

Setting up a real-time clock

- Open the Terminal
- Clone the Github repo Weather Station

```
cd ~ && git clone https://github.com/raspberrypi/weather-station
```

RTC setup

- Check if you have the latest updates for your Raspberry pi

```
sudo apt-get update && sudo apt-get upgrade
```

- You have to make some changes in the config file to allow Raspberry Pi to use the real-time clock

```
sudo nano /boot/config.txt
```

- Add the following lines at the bottom of the file

```
dtoverlay=w1-gpio
dtoverlay=pcf8523-rtc
```

- Press Ctrl & O to save and Ctrl & X to quit nano
- Now set the required modules to load automatically on boot

```
sudo nano /etc/modules
```

- Add the following lines to the bottom of the file

```
i2c-dev
w1-therm
```

- Press Ctrl & O to save and Ctrl & X to quit nano
- For the next steps, you have to connect the Weather Station HAT to the Raspberry Pi

```
sudo halt
```

- Reboot for the changes to take effect

```
sudo reboot
```

- Check that the real-time clock (RTC) appears in /dev

```
ls /dev/rtc*
```

  You should see something like /dev/rtc0

Initialise the RTC with the correct time

- Use the date command to see if the current system time is correct
  - when it is correct then set the RTC from the system clock:

    ```
    sudo hwclock -w
    ```
- If not then set the time manual

  ```
  sudo hwclock --set --date="yyyy-mm-dd hh:mm:ss" --utc
  ```
- Then set the system clock from the RTC time:

  ```
  sudo hwclock -s
  ```
- Now you have to enable setting the system time automatically
  - First edit the rule in /lib/udev

    ```
    sudo nano /lib/udev/hwclock-set
    ```
- Find the lines at the bottom that read:

  ```
  if [ yes = "$BADYEAR" ] ; then
          /sbin/hwclock --rtc=$dev --systz --badyear
  else
           /sbin/hwclock --rtc=$dev --systz
  fi
  ```
- Change the --systz options to --hctosys

Remove the fake hardware clock package

- Use this commands
  1. sudo update-rc.d fake-hwclock remove
  2. sudo apt-get remove fake-hwclock -y

Testing the sensors

- Install the necessary software packages
- Type the following commands:

```
sudo apt-get install i2c-tools python-smbus telnet -y
```

- Test that the I2C devices are online working

```
sudo i2cdetect -y 1
```

You should see an Output like this:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- UU 69 6a -- -- -- -- --
70: -- -- -- -- -- -- -- 77
```

- 40 = HTU21D, the humidity and temperature sensor.
- 77 = BMP180, the barometric pressure sensor.
- 68 = PCF8523, the real-time clock. It will show as UU because it's reserved by the driver.
- 69 = MCP3427, the analogue-to-digital converter on the main board.
- 6a = MCP3427, the analogue-to-digital converter on the snap-off AIR board.

Note: 40, 77 and 6a will only show if you have connected the **AIR** board to the main board.
Now that the sensors are working, we need a database to store the data from them.

Database setup

Now you have to setup the database, that the measurements get automatically updated and saved on your Raspberry PIs SD Card.

- Type the following commands in the command line:
  1.) sudo apt-get update
  2.) sudo apt-get install apache2 mysql-server python-mysqldb php5 libapache2-mod-php5 php5-mysql -y

  This will take some time

Create a local mysql database

- Now enter the following line:

    -
      ```
      mysql -u root -p
      ```
      Now enter the password you chose during installation
- Now you should be in the prompt “mysql>”
    - Now Create the Database:
      ```
      CREATE DATABASE weather;
      ```
      You should see “Query OK, 1 row affected (0.00 sec)”
- Now switch the database:
  ```
  USE weather;
  ```
  You should see “Database changed”

Create a Table

- Type the following code:
  ```
  CREATE TABLE WEATHER_MEASUREMENT(
     ID BIGINT NOT NULL AUTO_INCREMENT,
     REMOTE_ID BIGINT,
     AMBIENT_TEMPERATURE DECIMAL(6,2) NOT NULL,
     GROUND_TEMPERATURE DECIMAL(6,2) NOT NULL,
     AIR_QUALITY DECIMAL(6,2) NOT NULL,
     AIR_PRESSURE DECIMAL(6,2) NOT NULL,
     HUMIDITY DECIMAL(6,2) NOT NULL,
     WIND_DIRECTION DECIMAL(6,2) NULL,
     WIND_SPEED DECIMAL(6,2) NOT NULL,
     WIND_GUST_SPEED DECIMAL(6,2) NOT NULL,
     RAINFALL DECIMAL (6,2) NOT NULL,
     CREATED TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
     PRIMARY KEY ( ID )
   );
  ```
- Now you should see “Query OK,..”
- Now press Ctrl & D to exit

Set up sensors

- Begin with downloading the data logging code with the following commands

```
cd ~
git clone https://github.com/raspberrypi/weather-station.git
```

- This creates a new folder in your home directory

Start the daemon

- To start the daemon type in the following command

```
sudo ~/weather-station/interrupt_daemon.py start
```

Now you should see something like “PID: 2243”

Test it

- You can use telnet to test it
- Type in the following command

```
telnet localhost 49501
```

- You should see something like this

```
Trying 127.0...
Connected to localhost.
Escape character is '^]'.
OK
```

- Now you can use this commands
  - RAIN
  - WIND
  - GUST
  - RESET
  - BYE
- After testing use BYE to quit

Make the daemon start automatically during boot

- Type the following command to do the daemon in the autostart

```
sudo nano /etc/rc.local
```

- Insert the following lines at the bottom before “exit 0”

```
echo "Starting Weather Station daemon..."
/home/pi/weather-station/interrupt_daemon.py start
```

- Now press Ctrl & O then Enter and Ctrl & X to quit

Update MYSQL credentials

- When you are not in the “weather-station” type

```
cd ~/weather-station
```

- Then write the following command (You have to use your MYSQL root password)

```
nano credentials.mysql
```

- Then change the password field to your MYSQL root password
- Now press Ctrl & O then Enter and Ctrl & X to quit

Automatic updating of your database

- You should enable cron to automatically take measurements (data logging mode)
  ```
  crontab < crontab.save
  ```
- The weatherstation is now live and records data in timed intervals

View the data in the database

- Enter the command:
  ```
  mysql -u root -p
  ```
- Enter your MYSQL root password
- Enter the following command to change to the database weather:
  ```
  USE weather;
  ```
- To show the measurements type the following query:
  ```
  SELECT * FROM WEATHER_MESURMENT;
  ```
- Write “exit” to quit MYSQL

Start the REST-Service

- First clone the REST-Service from GitHub:
  ```
  cd ~ && git clone https://github.com/Krauck1/Weatherstation.git
  ```
- Change to the directory Python:
  ```
  cd Python
  ```
- Then type the following command:
  ```
  chmod +x libraries.sh
  ```
- Then install the libraries by running the following script:
  ```
  ./rest.py
  ```

Set up the Homepage

- Get the ip address with the following command:
  ```
  ifconfig
  ```
- Now change in the directory WebPage in the file line.js in the function getJson() to the ip address you got from your Raspberry
- Then execute the html file: weatherstation.html