Temperature Monitoring with Raspberry Pi

Fix Me! add grafana

Fix Me! add overview of all important files and services

Fix Me! Write down account names and passwords somewhere

Fix Me! add overview of the general structure of the setup

Fix Me! add “Important: it gets automatically new slaves but doesn't support removing slaves while running.”

Fix Me! add console prompt to code segments

The monitoring setup was developed by Etiénne Wodey. It is explained in

this talk

by him. The main aspects on how to set it up are documented in

this email

.

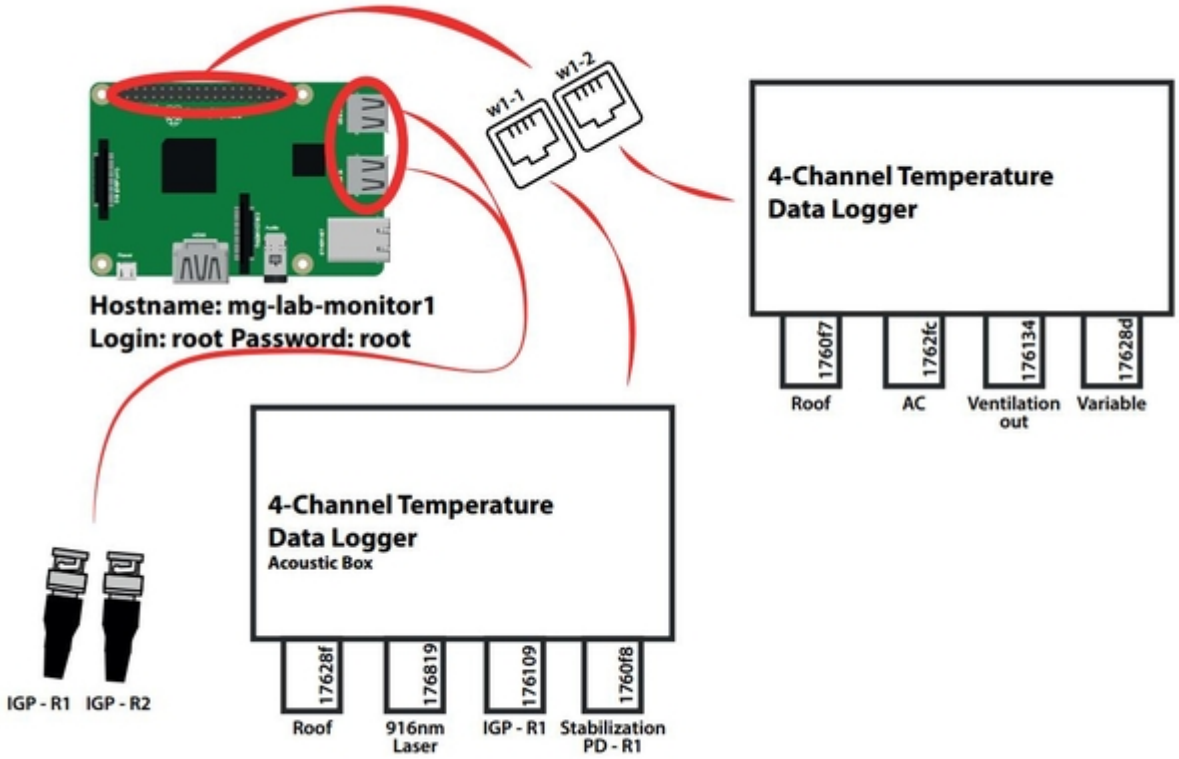
Short version: the raspberry pi reads out temperature sensors connected via one-wire bus, sends the data via LAN to an InfluxDB database located on the Magnesium Server. There the temperatures are stored and can be visualized in the browser using a Grafana, also running on the magnesium server. It can be accessed here

Database and Visualisation

- thingol (server name)
  - debian 9
  - influxdb
    - time series database
    - used for storing temperature values
    - set up as decribed in getting started section
  - Grafana
    - used for visualisation of the values stored in influxdb

Raspbery Pi

Basic principle



rasptemp1.pdf

Hardware setup

Fix Me! pictures and schematics

Installing Raspbian

Enabling One-Wire-Bus

Edit `/boot/config.txt`:

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

and add the line

```
dtoverlay=w1-gpio
```

Edit `/etc/modules`:

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

and add the lines

```
w1-gpio
w1-therm
```

Reboot the System:

```
sudo reboot
```

There should be folders `w1_bus_master<N>` in `/sys/devices`:

```
ls -l /sys/devices/
```

Check if there are devices connected:

```
ls -l /sys/devices/w1_bus_master1/
```

Replace `w1_bus_master_1` with any of the folders listed previously.

There should be folders named something like `3b-0000001760f8/`. These are the devices connected. Make sure there is a file named `w1_slave` for every device connected. This file is used to read from and write to the connected devices.

In case the folders or the files don't exist, make sure your devices are connected correctly, the steps above were done and/or reboot the system.

collectd

https://collectd.org/index.shtml

`collectd` is run on the raspberry pi to read out the temperature sensors and send the messages to the influxdb database located on another server.

install collectd:

```
sudo apt-get update
sudo apt-get install collectd
```

check the status of the daemon:

```
sudo systemctl status collectd
```

if the status is running, stop the daemon:

```
sudo systemctl stop collectd
```

to view all systemlog messages by collectd:

```
journalctl -u collectd -b
```

The collectd configuration file is located at /etc/collectd/collectd.conf. Edit it:

```
sudo nano /etc/collectd/collectd.conf
```

Copy the contents of

this file

into there.

Make sure to adjust the name of the server in the network plugin section to your needs.

Python Plugin

The temperature sensors are read out by a python module using the python plugin for collectd. Place the script

w1_therm_monitor.py

in the folder `/home/pi/` or change the ModulePath option for the python plugin in `/etc/collectd/collectd.conf` if you want to place it in another folder.

The collectd module reads `w1_master_slaves`, takes all the serial numbers starting with 28 (DS18B20 family) or 3b (MAX31850 family) and reads the corresponding slave files (i.e. the temperatures).

Starting collectd

Start the collectd service using:

```
sudo systemctl start collectd
```

Check if it runs without errors by reading

```
sudo systemctl status collectd
```

or checking

```
journalctl -u collectd -b
```

The system automatically gets new slaves but doesn't support removing slaves while running

influxdb

https://www.influxdata.com/time-series-platform/influxdb/

Setup

set up as decribed in getting started section

Creating the database

Create a database in which you later wish to store the temperatures measured, for now we call it `collectd_test`.

In influxDB execute:

```
CREATE DATABASE collectd_test
```

Check if it was created properly using:

```
USE collectd_test
```

Exit using

```
EXIT
```

Configuring for collectd

Stop the influxdb server by running:

```
sudo systemctl stop influxd
```

The configuration file is `/etc/influxdb/influxdb.conf`. The documentation can be found here: https://docs.influxdata.com/influxdb/v1.6/administration/config/

The config file we use is

this

.

The interesting part is:

```
[[collectd]]
  enabled = true
  bind-address = ":25826"
  database = "collectd_test"
  security-level = "none"
```

This enables the collectd plugin for influx and writes to the database we specify without any privileges required.

adjust the configuration file to be the same as above.

types.db

The collectd plugin for influx requires a types.db document. Copy this from `/usr/share/collectd/` on the raspberry pi to `/usr/local/share/collectd/` on the server running influxDB.

Restarting the server

Start the server by typing

```
influxd
```

and leave the terminal open. If there are any errors, these should be displayed now.

To check if there are values being written into the database, use another terminal or ssh to log into influx:

```
influx
```

and check for values in the database:

```
USE collectd_test
SHOW SERIES
```

If something is displayed, try listing those values by using

```
SELECT * FROM <name of the time series>
```