**Getting things ready**

Download the RaspberryPI OS image

<https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit>

Unzip the image and put it into a folder.

Put the SDcard into your laptop.

It should appear as another drive in your drive overview.

Download <https://www.balena.io/etcher>

* Select the image
* Select the destination drive
* Hit flash

**Flash the SDcard**

After Flash you will see two drives.

Boot D: with the OS

Put a file named “ssh” into the D directory.

This is to activate ssh to allow remote login.

Put the SDcard into the Pi hardware and plug it in.

Figure out the IP address via your DHCP router and connect via Putty (ssh client) to the Pi server.

Alternative to figure out the IP address if you are logged into the PI, with an attached screen and keyboard.

hostname -I

Make the IP address of your Pi static

/etc/dhcpcd.conf

interface eth0

static ip\_address=192.168.0.99/24

static routers=192.168.0.1

static domain\_name\_servers=192.168.0.1 8.8.8.8

If you use the screen with Pi.

* Setup the password for pi user
* Setup location and timezone
* Setup wifi
* Let it update itself (this will take a while)

Set the password for root

Sudo passwd

Enable SSH if it hadn’t started already.

sudo systemctl enable ssh

sudo systemctl start ssh

Reboot (with root)

SSH should now start automatically.

Check on your router, what ipaddress has been added.

The name of the host should be rasberrypie.

Ssh to the pie, using putty.

**Set the timezone**

The NTP client is set automatically to connect to a timeserver, provided the client can access the internet. You can check this with:

# timedatectl status

Adjust what you need with the following command.

timedatectl set-timezone Europe/Amsterdam

**Set hostname**

raspi-config

1 -> S4

Set hostname to WaterMon

Reboot

Finally update Pi to latest packages

sudo apt update

sudo apt full-upgrade

sudo shutdown -r now

Cleanup the installed packages to create some space

df -h

sudo apt clean

**Now you are set for installation.**

**How does it work?**

1. **The sensor script calls the sensor every 15min via a cronjob**
2. **It stores the value in to the database**
3. **An HTTP server is setup, so you can surf to a webpage**
4. **The webpage calls the values out of the database and displays them in a histogram for easy overview**

So you need to setup:

* HTTP server (apache)
* Mysql (MariaDB)
* PHP
* phpMyAdmin (for easy admin of the Mysql DB

**Install Apache2**

[**https://pimylifeup.com/raspberry-pi-apache**](https://pimylifeup.com/raspberry-pi-apache)

**sudo apt install apache2 -y**

**Where is the apache important files installed?**

/sbin/apached

/etc/apache2/apache2.conf

The apache deamon will be started automatically.

sudo /usr/sbin/apache2ctl -k stop

sudo /usr/sbin/apache2ctl -k start

**Logfiles**

/var/log/apache2/access.log and the error.log

**Home page**

/var/www/html/index.html

Check if the HTTP server is setup correctly

By going to the http port with a browser.

Set the right permissions

Apache2 runs as user www-data

Add the pi user to this group.

sudo usermod -a -G www-data pi

sudo chown -R -f www-data:www-data /var/www/html

**PHP for apache2 – latest version is 7.3**

sudo apt install php libapache2-mod-php -y

The install creates a default php page

/var/www/html/index.php

<?php phpinfo(); ?>

Check if PHP works by surfing to the http server, call the index.php.

**MYSQL aka MariaDB**

<https://pimylifeup.com/raspberry-pi-mysql/>

sudo apt install mariadb-server

sudo mysql\_secure\_installation

set admin access with password littlebeast

Login to check if DB has been installed correctly.

sudo mysql -u root -p

**Install PHP for mysql**

sudo apt install php-mysql

**which version of php is this?**

pi@WaterMon:/etc $ php -v

PHP 7.3.27-1~deb10u1 (cli) (built: Feb 13 2021 16:31:40) ( NTS )

Copyright (c) 1997-2018 The PHP Group

Zend Engine v3.3.27, Copyright (c) 1998-2018 Zend Technologies

with Zend OPcache v7.3.27-1~deb10u1, Copyright (c) 1999-2018, by Zend Technologies

**Install phpMyAdmin tool to manage the Mysql db via a web interface**

<https://www.liquidweb.com/kb/install-phpmyadmin-ubuntu-18-04/>

Follow the instruction

Make sure you are **root or use sudo**

apt install phpmyadmin php-mbstring php-gettext

Password set for phpmyadmin

Make sure you add the conf file to the last line of the httpd startup conf file

vi /etc/apache2/apache2.conf

Include /etc/phpmyadmin/apache.conf

I’ve setup phpMyAdmin to listen to port 99, as 80 is already used by pi-hole.

Note: pihole runs on httpdlight, not on Apache2 (!!!)

/usr/sbin/lighttpd -D -f /etc/lighttpd/lighttpd.conf

Restart the httpd

/usr/sbin/apache2ctl -k stop

/usr/sbin/apache2ctl -k start

Surf to the MyAdmin page

<http://192.168.0.99:99/phpmyadmin>

Login with **phpmyadmin / xxx**

If you can’t login, try to login with root.

If that doesn’t work, you may need to reset the root password.

sudo service mysql stop

sudo mysqld\_safe --skip-grant-tables

You now run a safe version of mysql. Don’t close this screen.

Open a second putty session

mysql -u root

use mysql;

mysql> update user set authentication\_string=password('password') where user='root';

Now the password is reset to “password”.

Now do a cntr-z on the first screen.

Reboot the Pi

sudo shutdown -r

**Check if you can login via de command line.**

sudo mysql -u root -p

Check what users are configured in the mysqlDB

MariaDB [(none)]> select user from mysql.user;

+------------+

| user |

+------------+

| peter |

| phpmyadmin |

| root |

+------------+

3 rows in set (0.001 sec)

FLUSH PRIVILEGES;

**How to check your version of phpMyAdmin?**

pi@WaterMon:/etc $ dpkg -l phpmyadmin

Desired=Unknown/Install/Remove/Purge/Hold

| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend

|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)

||/ Name Version Architecture Description

+++-==============-============-============-=================================

ii phpmyadmin 4:4.6.6-5 all MySQL web administration tool

**I saw error on the PHPmyadmin page.**

Solving annoying version difference between PHP and PHPmyadmin.

<https://stackoverflow.com/questions/48001569/phpmyadmin-count-parameter-must-be-an-array-or-an-object-that-implements-co>

/usr/share/phpmyadmin/libraries/sql.lib.php

Replace line 613

|| (count($analyzed\_sql\_results['select\_expr'] == 1)

with

|| ((count($analyzed\_sql\_results['select\_expr']) == 1)

No need to restart apache. Just refresh your browser page while using PHPmyadmin.

**I get an messages “no privileges” if I want to modify the tables. This is a security feature. By default you cannot setup databases when logged in from a machine with a different IP address as the DB machine.**

sudo mysql -u root -p

create database water\_level;

CREATE USER 'peter'@'%' identified by xxx;

grant all on \*.\* to 'peter'@'%';

flush privileges;

The % allows you to remotely access the mysql db from another machine.

OK.

Now you are ready to setup the code to run the watermonitor.

What code do you need?

1. Database configuration file
2. Script that calls for a sensor value and store it in the DB
3. Crontab to make sure that 2. Runs every 15min
4. HTML page pulling the values out of the DB and putting them into a histogram
5. Some test scripts to verify that above scripts are working correctly.

**Setup the database.**

mysql -u root -p

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

Use water\_level;

CREATE TABLE IF NOT EXISTS `level\_log` (

`id` int(10) NOT NULL,

`level` dec(10,1) NOT NULL,

`timestamp` int(12) NOT NULL,

`date\_time` datetime NOT NULL ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 AUTO\_INCREMENT=1 ;

ALTER TABLE `level\_log`

ADD PRIMARY KEY (`id`);

ALTER TABLE `level\_log`

MODIFY `id` int(10) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=1;

Enter one line to use for some tests.

INSERT INTO `level\_log`(`id`, `level`, `timestamp`, `date\_time`) VALUES ('1','10','1622498400','2021-06-01 00:00:00');

Then run as root user in the db.

grant all on \*.\* to 'peter'@'%';

flush privileges;

**Test script to check if the sensor works**

[**https://www.dfrobot.com/forum/topic/312660**](https://www.dfrobot.com/forum/topic/312660)

**Code to pull a value from the sensor and insert it into the database**

# Project: Rain Water Tank

# Using Raspberry PI UART interface

# Using waterproof A02YYUW distance sensor

#To test this script

#Run from the command line with python3 small.py

#this script is executed every 15 minutes with CRON settings of Raspberry Pi

#0,15,30,45 \* \* \* \* python3 /home/pi/small.py

# Settings

import time,os

import datetime

# Load sensor class

from DFRobot\_RaspberryPi\_A02YYUW import DFRobot\_A02\_Distance as Board

board = Board()

# Take 5 samples of the sensor

def get\_distance():

dist\_add = 0

for x in range(5):

try:

distance = board.getDistance()

# distance is measured in mm, converting to cm

distance = distance/10

# Entry to print the sample output

print (x, "distance: ", distance)

dist\_add = dist\_add + distance

time.sleep(.3) # 300ms interval between readings

except Exception as e:

pass

# show a list of the samples

print ("x: ", x+1, "samples")

# Take the average of the samples, rounded at 1 digits after the decimal point

avg\_dist=dist\_add/(x+1)

dist=round(avg\_dist,1)

# show the average result

print ("distance: ", dist)

return dist

# end of function

# Function to send the sensor measured distance to the remote database

def sendData\_to\_remoteServer(url,dist):

url\_full=url+str(dist)

urlopen(url\_full)

print("sent to url: ",url\_full)

from urllib.request import urlopen

passcode="xxx"

url\_remote="http://192.168.0.99/db\_insert.php?level="

# End of function

# Main logic

distance=get\_distance()

sendData\_to\_remoteServer(url\_remote,distance)

# End of program

**Crontab setup**

<https://phoenixnap.com/kb/how-to-list-display-view-all-cron-jobs-linux>

sudo crontab -e

0,15,30,45 python3 /home/pi/small.py

HTML scripts

pi@pi-hole:/var/www/html $ ls -l

total 36

-rw-r--r-- 1 pi pi 6766 Aug 14 16:28 barchart.php

-rw-r--r-- 1 pi pi 986 Jun 25 2021 db\_check.php

-rw-r--r-- 1 pi pi 1051 Jun 25 2021 db\_insert.php

**Test script DB connectivity**

**db\_check.php**

Script to check if the connectivity to the Database is setup.

<?php

#$servername = "192.168.1.102";

$servername = "localhost";

$username = "peter";

$password = "xxx";

// Create connection

$conn = mysqli\_connect($servername, $username, $password);

// Check connection

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

echo "Connected successfully";

?>

You can run this script when calling from a browser.

If this script fails for the hardcoded IPaddress, but works as localhost, then you have a permission issue. You can fix this. <https://linuxize.com/post/mysql-remote-access/>

[root@WaterMon:/etc/mysql/mariadb.conf.d](mailto:root@WaterMon:/etc/mysql/mariadb.conf.d)

edit the server file: 50-server.cnf

bind-address = 0.0.0.0

Then restart the mysql DBserver

sudo systemctl restart mysql

**Test script DB insert setup**

**db\_insert.php**

<?php

// Project Smart Water Tank

// Coded by PeterK

// To test this script

// call the script as an url via http

// Check MySQL connection

$link = mysqli\_connect("localhost", "peter", "xxx", "water\_level");

if($link === false){

die("ERROR: Could not connect. " . mysqli\_connect\_error());

}

// Setting variables

date\_default\_timezone\_set("Europe/Amsterdam");

// $level="10"; // hardcoded entry to call by commandline - replace by next line to call from http

$level=$\_GET["level"]; // variable set by remote http call

$t=time();

$date\_time= date('Y-m-d H:i:s', $t);

// Check variables

echo "value = $level<br>";

echo "time = $t<br>";

echo "date\_time = $date\_time<br>";

// Attempt insert query execution

$sql="INSERT INTO `level\_log`(`level`, `timestamp`, `date\_time`)

VALUES ('$level','$t', '$date\_time')";

if(mysqli\_query($link, $sql)){

echo "Records inserted successfully.";

} else{

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

// Close connection

mysqli\_close($link);

?>

**Creating a graph**

<https://developers.google.com/chart/interactive/docs/gallery/columnchart>

**apache2 www directory: /var/www/html/** **barchart.php**

Note that the barchart.html file has the formula to calculate the water height.

The code for this script is quite extensive.

The code is available on my github page.

<https://github.com/pkassies/Rainwater-calculator>

As my waterbox is 42cm high, I need to substract the measured sensor value from it to obtain the waterlevel.

echo "['". $date3 ."',**42** - '".$row\_val['level']."'],";

**Integrating A02YYUW and RaspberryPI**

Setup

* Have an running RaspberryPI 4 with either a DSI or SSH for access available
* Have a A02YYUW sensor

<https://wiki.dfrobot.com/_A02YYUW_Waterproof_Ultrasonic_Sensor_SKU_SEN0311>

The example here uses a Raspberry Pi IO Expansion Board. However, you can connect the sensor directly to the right pins on the RaspberryPI and it will work too.

**Configuring RaspberryPI**

The sensor needs to be attached to the UART bus. If you don’t know what UART is, this is a good link with a primer on how the UART works.

[**https://www.electronicwings.com/raspberry-pi/raspberry-pi-uart-communication-using-python-and-c**](https://www.electronicwings.com/raspberry-pi/raspberry-pi-uart-communication-using-python-and-c)

By default, the primary UART is assigned to the Linux console. If you wish to use the primary UART for other purposes, you must reconfigure Raspberry Pi OS. You can check this:

ls -l /dev | more

lrwxrwxrwx 1 root root 5 May 8 18:45 serial0 -> ttyS0

lrwxrwxrwx 1 root root 7 May 8 18:45 serial1 -> ttyAMA0

cd /boot

Add the following two lines to config.txt

enable\_uart=1

dtoverlay=disable-bt

Reboot the RaspberryPI.

Check if the UART is setup correctly.

ls -l /dev | more

You should see two lines passing by

lrwxrwxrwx 1 root root 7 aug 7 22:17 serial0 -> ttyAMA0

lrwxrwxrwx 1 root root 5 aug 7 22:17 serial1 -> ttyS0

A picture containing text

Description automatically generated

This means that the output of serial0 is forwarded to ttyAMA0.

You may also need to install the serial package for Raspberry PI.

sudo apt-get install python3-serial

**Connecting RaspberryPI and A02YYUW sensor**

The sensor has 4 wires: power, ground, transmission and receive

I’ve used the following setup:

* Sensor red wire = power => RasberryPI pin 4
* Sensor blue wire = TX => RaspberryPI pin 8 = GPIO14 = UART\_TXD0
* Sensor green wire = RX => RaspberryPI pin 10 = GPIO15 = UART\_RXD0
* Sensor black wire = ground = Raspberry pin 14

Diagram

Description automatically generated

A picture containing text, electronics

Description automatically generated

A picture containing text, electronics, circuit

Description automatically generated

Download the **A02YYUW** sensor library from

<https://github.com/DFRobot/DFRobot_RaspberryPi_A02YYUW>

/home/pi

sudo git clone https://github.com/DFRobot/DFRobot\_RaspberryPi\_A02YYUW

You should see a new directory

pi@raspberrypi:~ $ ls -l

totaal 40

drwxr-xr-x 2 pi pi 4096 mrt 4 2021 Bookshelf

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Desktop

**drwxr-xr-x 6 root root 4096 aug 7 22:44 DFRobot\_RaspberryPi\_A02YYUW**

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Documents

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Downloads

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Music

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Pictures

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Public

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Templates

drwxr-xr-x 2 pi pi 4096 mrt 5 2021 Videos

cd into the DFRobot directory

The directory consist of:

pi@raspberrypi:~/DFRobot\_RaspberryPi\_A02YYUW $ ls -l

totaal 36

-rw-r--r-- 1 pi pi 1287 aug 7 22:04 demo.py

-rw-r--r-- 1 root root 2465 aug 7 21:08 DFRobot\_RaspberryPi\_A02YYUW.py

drwxr-xr-x 2 root root 4096 aug 7 21:08 examples

-rw-r--r-- 1 root root 285 aug 7 21:08 library.properties

-rw-r--r-- 1 root root 1053 aug 7 21:08 LICENSE

-rw-r--r-- 1 root root 2269 aug 7 21:08 README\_CN.md

-rw-r--r-- 1 root root 2227 aug 7 21:08 readme.md

drwxr-xr-x 3 root root 4096 aug 7 21:08 resources

To check if the setup works

cd examples

python3 demo\_get\_distance.py

If you have connected the wires correctly you should see the following output:

pi@raspberrypi:~/DFRobot\_RaspberryPi\_A02YYUW/examples $ python3 demo\_get\_distance.py

Distance 1641 mm

Distance 1645 mm

Distance 1645 mm

Distance 1641 mm

Distance 1645 mm

Distance 1653 mm

Note: that if you move the sensor, the distance doesn’t change.

However, if you restart the script, the values are correct.

You can benchmark the distance using a ruler.

A picture containing wall, indoor

Description automatically generated

Text

Description automatically generated