Nackademin -22 Internship

IBM

[IBM/Nackademin]

[PUMP DOCUMENTATION]

[Software set up: 3](#_Toc1265672255)

[Installation: 3](#_Toc1354070497)

[Raspberry pi: 3](#_Toc1622539223)

[Raspbian 64bit: 3](#_Toc1760886531)

[Node red: 3](#_Toc1276621803)

[TinyDB: 3](#_Toc751967925)

[Installation: 3](#_Toc2100864111)

[Mosquitto: 4](#_Toc1565062412)

[Installation: 4](#_Toc1186930165)

[Arduino ide: 4](#_Toc1741455996)

[React: 4](#_Toc234342823)

[HARDWARE 6](#_Toc34691373)

[DPSD lighted pushbutton switch: 6](#_Toc1726775358)

[Wiring: 6](#_Toc799150063)

[Relay module for arduino: 7](#_Toc1362320268)

[Datasheet: 7](#_Toc311680051)

[Potentiometer: 7](#_Toc712002995)

[Datasheet: 7](#_Toc1099991735)

[CODE 8](#_Toc1014869708)

[Arduino: 8](#_Toc360369493)

[Functions: 8](#_Toc88454200)

[buttonAIsTriggered(); 8](#_Toc494840659)

[debounce(); 8](#_Toc396803258)

[pumpcontroller(); 8](#_Toc1471336944)

[buttonAIsTriggered(); 9](#_Toc1651388589)

[commandHandler(char comdata); 9](#_Toc1028851237)

Documentation Water pump Ibm

# Software set up:

## Installation:

### Raspberry pi:

#### Raspbian 64bit:

https://www.raspberrypi.com/software/

#### Node red:

**Install:**

bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)

**Uninstall:**

sudo npm -g remove node-red

**Auto start on boot**: sudo systemctl enable nodered.service

*Install palette*:

Node red Dashboard.

*Control commands*:

* **node-red-start** - this starts the Node-RED service and displays its log output. Pressing Ctrl-C or closing the window does *not* stop the service; it keeps running in the background
* **node-red-stop** - this stops the Node-RED service
* **node-red-restart** - this stops and restarts the Node-RED service
* **node-red-log** - this displays the log output of the service

### TinyDB:

#### Installation:

**sudo pip3 install tinydb**

Example to create a “collection” in TinyDB.

pi@raspberrypi:~ $ python3

Python 3.9.2 (default, Feb 28 2021, 17:03:44)

[GCC 10.2.1 20210110] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> from tinydb import TinyDB, Query

>>> db = TinyDB('/home/pi/pumpstate.json')

>>> Device = Query()

>>> db.insert({'City': 'Stockholm' , 'DeviceName' : 'Pump1' , 'PumpState' : 0})

2

>>> db.search(Device.DeviceName == 'Pump1')

[{'City': 'Stockholm', 'DeviceName': 'Pump1', 'PumpState': 0}]

>>> Component = Query()

>>> db.update({'PumpState': 1}, Component.DeviceName == 'Pump1')

[2]

>>> db.all()

[{'PumpState': 0}, {'City': 'Stockholm', 'DeviceName': 'Pump1', 'PumpState': 1}]

From: <https://tinydb.readthedocs.io/en/latest/> (Examples for database usage)

### Mosquitto:

#### Installation:

sudo apt install mosquitto mosquitto-clients

**Auto start on boot**: sudo systemctl enable mosquitto.service

**Verify installation**:

sudo systemctl status mosquitto

### Arduino ide:

<https://www.arduino.cc/en/Main/Software>

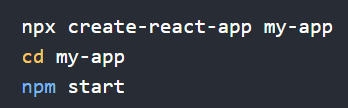
32 bit, Linux arm.

install.sh file and select Execute.

### React:

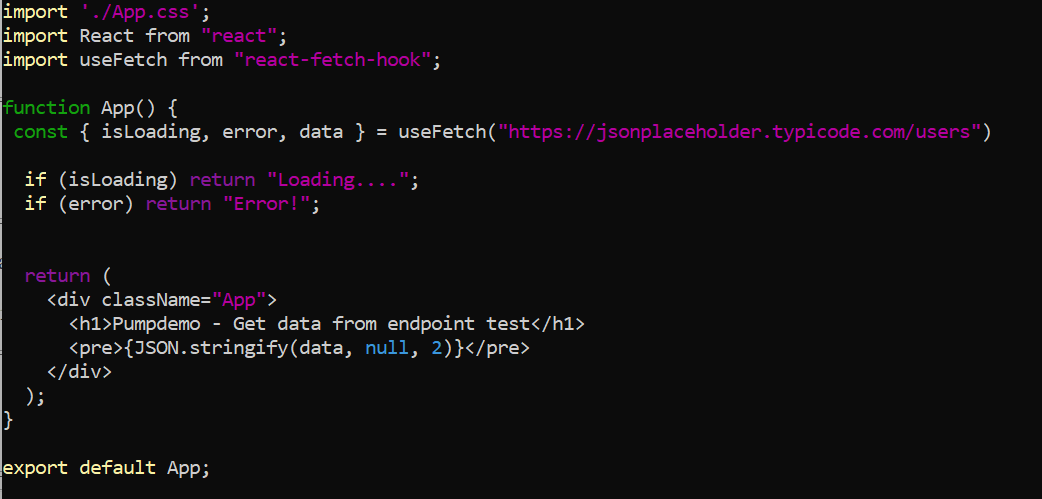
Using website:[**https://reactjs.org/docs/create-a-new-react-app.html**](https://reactjs.org/docs/create-a-new-react-app.html)

Create template app first with the following command (change my-app name)

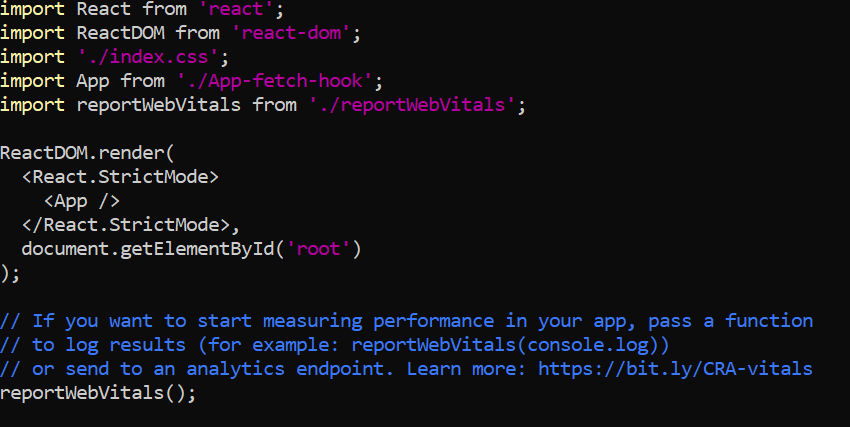


Used “react-fetch-hook" to access end point by installing npm i react-fetch-hook.

Actual code example can be found on the Pi in **/home/pi/React/pumpdemo/src/App-fetch-hook.js.**



*You must also change the link to this function in index.js as shown on line 4. - import App from ‘./App-fetch-hook';*



# HARDWARE

## DPSD lighted pushbutton switch:

Type: ModMyToys (19mm?) Illuminated Momentary Switch

**Characteristics:**

Diameter: 19mm?

Switching capacity: 3A / 250VAC

Form: Flathead  
LED type: ring lighting  
Colors: Red, Blue, Green, Orange  
Voltage: 12V  
Terminal: 6-Pin Terminal  
Temperature: -40 to 75 ° C

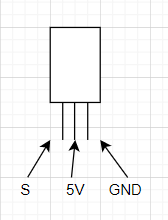
### Wiring:

Diagram

Description automatically generated

## Relay module for arduino:

### Datasheet:



## Potentiometer:

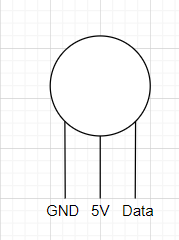
Characteristics:

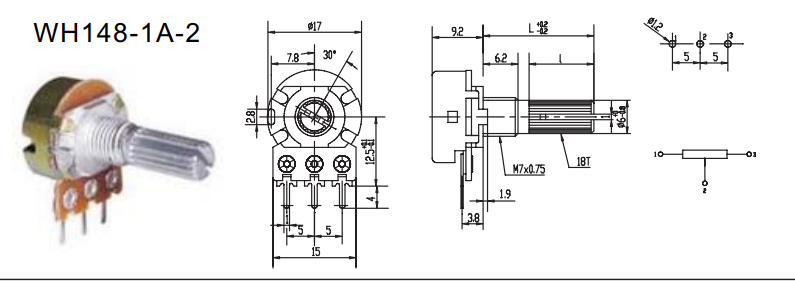
Modell: WH148

Type: Linear

Resistance: 100 kΩ

### Datasheet:





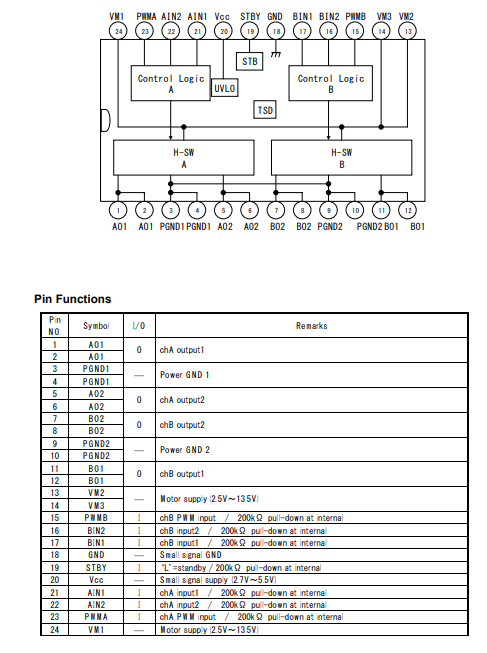
H-Bridge:

Characteristics:

Name: Motor Driver 1A Dual TB6612FBG

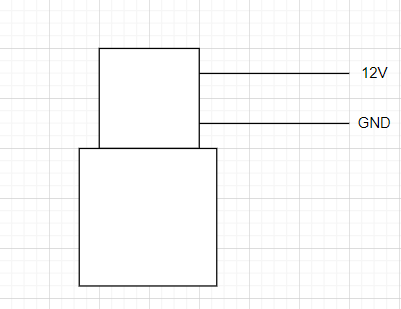
Dimensions: 0.8×0.8

### Datasheet:



## Valve:

### Datasheet:



## Flow Sensor:

**YF-S201 Hall-Effect Water Flow Sensor**Diagram

Description automatically generated

#### Formula:

Here we have determined flow rate by change in velocity of water. The pipe’s cross-sectional area is known and remains constant, the average velocity is an indication of the flow rate. The basis relationship for determining the liquid’s flow rate in such cases is **Q=VxA**, where **Q** is flow rate/total flow of water through the pipe, **V** is average velocity of the flow and **A** is the cross-sectional area of the pipe (viscosity, density and the friction of the liquid in contact with the pipe also influence the flow rate of water).

* Pulse frequency (Hz) = 7.5Q, Q is flow rate in Litres/minute
* Flow Rate (Litres/Min) = (Pulse frequency) / 7.5Q

------------------------------------------------------------------------------------------

* Sensor Frequency (Hz) = 7.5 \* Q (Liters/min)
* Litres = Q \* time elapsed (seconds) / 60 (seconds/minute)
* Litres = (Frequency (Pulses/second) / 7.5) \* time elapsed (seconds) / 60
* Litres = Pulses / 7.5

*\*Error range +-10*

# CODE

## Arduino:

### Functions:

#### buttonAIsTriggered();

when a button is pushed, it sets the triggered state to true,

as an extra check before the debounce() function.

#### debounce();

check if the button is actually pressed with the buttonAisTriggered() function, den it checks if it was x millis long ago. If so, it calls the function.

#### pumpcontroller();

checks if the pump is on or off, and sets it to the opposite.

#### pumpSerial();

checks if theres incommin serial data, if so it sends data into commandHandler();

#### commandHandler(char comdata);

The incomming char saved in the variable comdata runs through a switch case, to trigger the right function.

Eg.

#### 

void commandHandler(char comdata) {

switch (comdata) {

case 'A':

pumpcontroller(); // A Starts or turns off pump

break;

}

}