**Stepwise Installation Instructions for OpenHAB and the basic telemetry system**

**Note- The easiest path to get your system running is to follow the Installation Instructions and use the included shell script. The instructions written below take you through a stepwise installation using the same material as in the shell script.**

**These instructions have been tested and verified on a Raspberry Pi 4. They will not work on a Raspberry Pi 5, as its GPIO interface is not compatible with pigpiod. It is possible that pigpiod will be rewritten to be compatible with a Raspberry Pi 5. In the meantime, be sure to procure Raspberry Pi 4 hardware (or earlier) to work with this software.**

Setting up OpenHAB on Raspberry Pi 4:

1. **Install JAVA version 17 with:**

sudo apt install openjdk-17-jdk -y

1. **Type the following commands to set up the JAVA environment:**

sudo export JAVA\_HOME=/usr/lib/jvm/java-17-openjdk-arm64

sudo export PATH=$PATH:$JAVA\_HOME/bin

**For more information, see** [**http://thelinuxcode.com/install-java-17-raspberry-pi/**](http://thelinuxcode.com/install-java-17-raspberry-pi/)

1. **Follow the installation instructions at the OpenHAB site for apt-based Linux solutions (reproduced below):**

**First, add the OpenHAB repository key to your package manager. Type:**

curl -fsSL "https://openhab.jfrog.io/artifactory/api/gpg/key/public" | gpg --dearmor > openhab.gpg

sudo mkdir /usr/share/keyrings

sudo mv openhab.gpg /usr/share/keyrings

sudo chmod u=rw,g=r,o=r /usr/share/keyrings/openhab.gpg

**Then add the OpenHAB Stable Repository to your system’s apt sources list (the text below is all one line):**

echo 'deb [signed-by=/usr/share/keyrings/openhab.gpg] https://openhab.jfrog.io/artifactory/openhab-linuxpkg stable main' | sudo tee /etc/apt/sources.list.d/openhab.list

**Then synchronize the package index:**

sudo apt-get update

**and install OpenHAB**

sudo apt-get install openhab

**Prevent automatic upgrades of the OpenHAB package, as these can unexpectedly break your system:**

sudo apt-mark hold openhab

sudo apt-mark hold openhab-addons

**Finally, start OpenHAB and make it a service that will run on every bootup:**

sudo systemctl start openhab.service

sudo systemctl status openhab.service

sudo systemctl daemon-reload

sudo systemctl enable openhab.service

**Note that the first start may take up to 15 minutes.**

1. **Copy the following files to the directories listed below:**

repeater.things to the /etc/openhab/things folder.

repeater.items to the /etc/openhab/items folder.

repeater.rules to the /etc/openhab/rules folder.

repeater.sitemap to the /etc/openhab/sitemap folder.

addons.cfg to the /etc/openhab/services folder.

ntp.cfg to the /etc.openhab/services folder.

rrd4j.cfg to the /etc.openhab/persistence folder.

blinkatest.py to your home directory ~

sensorscript.py to your home directory ~

MCNsensor.service to /usr/lib/system/system

**An example of how to do this:** Go to the directory where you downloaded repeater.things. Type:

sudo mv repeater.things /etc/openhab/things

1. **Set up the script to read the GPIO ports by doing the following:**
2. **Install Paho mqtt client** This needs to be done as sudo to ensure it will run as an executable service by root.

python3 -m venve ~/venv

source ~/venv/bin/activate

pip3 install paho-mqtt

See: [How to Connect Raspberry Pi with MQTT for IoT Communication (iotdesignpro.com)](https://iotdesignpro.com/projects/how-to-connect-raspberry-pi-with-mqtt#:~:text=For%20this%20first%20install%20Paho%20MQTT%20library%20using,code%20and%20paste%20it%20to%20the%20subscriber%20file.)

1. Some DHT sensors do not work using the other details found at the website above. **In that case, you need to use the pigpio DHT library found at**: <https://pypi.org/project/pigpio-dht/>

pip install pigpio-dht

sudo apt-get install pigpiod

sudo systemctl enable pigpiod

sudo systemctl start pigpiod

1. **Install the GPIO binding and configure** as noted on <https://www.openhab.org/addons/bindings/gpio/>
2. **Install the ADS1115 ADC according to the instructions at** [**https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115**](https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115)**. We will be using the SCL and SDA pins to connect the module.**
   1. **Need to install circuitpython instead. Go to:** [Python & CircuitPython | Adafruit 4-Channel ADC Breakouts | Adafruit Learning System](https://learn.adafruit.com/adafruit-4-channel-adc-breakouts/python-circuitpython)

**In summary, type the following (as shown on the website above):**

cd ~

pip3 install --upgrade adafruit-python-shell

wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/raspi-blinka.py

sudo -E env PATH=$PATH python3 raspi-blinka.py

**Then check I2C and SPI:**

ls /dev/i2c\* /dev/spi\*

**You should see the response**

/dev/i2c-1 /dev/spidev0.0 /dev/spidev0.1

1. **Copy blinkatest.py to your home directory. Run it by typing:**

python blinatest.py

It should indicate that Digital I/O, I2C and SPI are OK.

1. **Type:** pip3 install adafruit-circuitpython-ads1x15

**Now you are ready to use the board.**

1. **The program sensorscript.py contains all of the details needed to run the two DHT sensors and the four ADC input ports.** 
   1. **The line** mqttc = mqtt.Client(mqtt.CallbackAPIVersion.VERSION1) **is needed with the new version of paho mqtt. The old version required** mqttc = mqtt.Client()

**Note that this is a temporary deprecation, and you can find other workaround details at** [Migrations — Eclipse paho-mqtt documentation](https://eclipse.dev/paho/files/paho.mqtt.python/html/migrations.html)

* 1. **Make this python script executable by typing**

cd ~

sudo chmod +x sensorscript.py

* 1. **Copy the file MCNsensor.service to the following directory:**

/usr/lib/systemd/system/

**Type the following commands:**

sudo chmod 644 usr/lib/systemd/system/MCNsensor.service

sudo systemctl daemon-reload

sudo systemctl enable MCNsensor.service

sudo systemctl start MCNsensor.service

**Note – Both DHT sensors must be connected to the board – even if you only configure the system to report on only one sensor.** This requirement is due to the fact that MCNsensor.service polls two DHT11 sensors, and if it sense one is missing, it will hang up.