

Smogomierz PCKZiU

Smogomierz jest projektem którego celem jest gromadzenie i wyświetlanie danych pogodowych takich jak: temperatura, wilgotność, ciśnienie, PM1.0, PM2.5, PM10.0.

Legenda

1. Wstęp
2. Serwer
 1. Instalacja Systemu
 2. Instalacja Oprogramowania
 3. Konfiguracja oprogramowania
3. Klient
 1. Lista części
 2. Program do ESP8266

Wstęp

System smogomierza składa się z dwóch głównych części z **serwera** (raspberry pi) oraz **klienta** (esp8266). Zadaniem serwera jest :

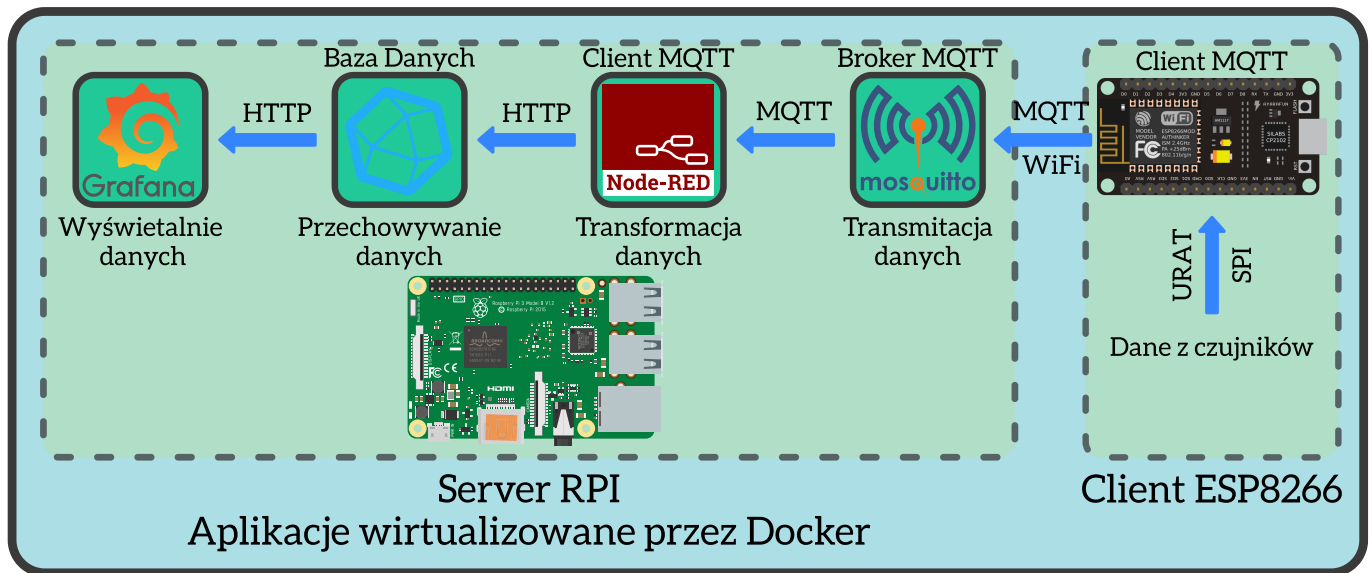
- Udostępnianie możliwości wysyłania danych (Mosquitto broker)
- Przetwarzanie tych danych (Node-RED)
- Gromadzenie danych (InfluxDB)
- Wyświetlanie danych (Grafana)

Zadania te realizowane przez aplikacje wirtualizowane na serwerze. Wirtualizacją zajmują się aplikacja **Docker** i skrypt pomocniczy **Docker-compose**.

Do zadań klienta należą :

- Odczyt danych z czujnika
- Wysyłanie danych do serwera

Schemat komunikacji i działania smogomierza znajduje się w zdjęciu poniżej.

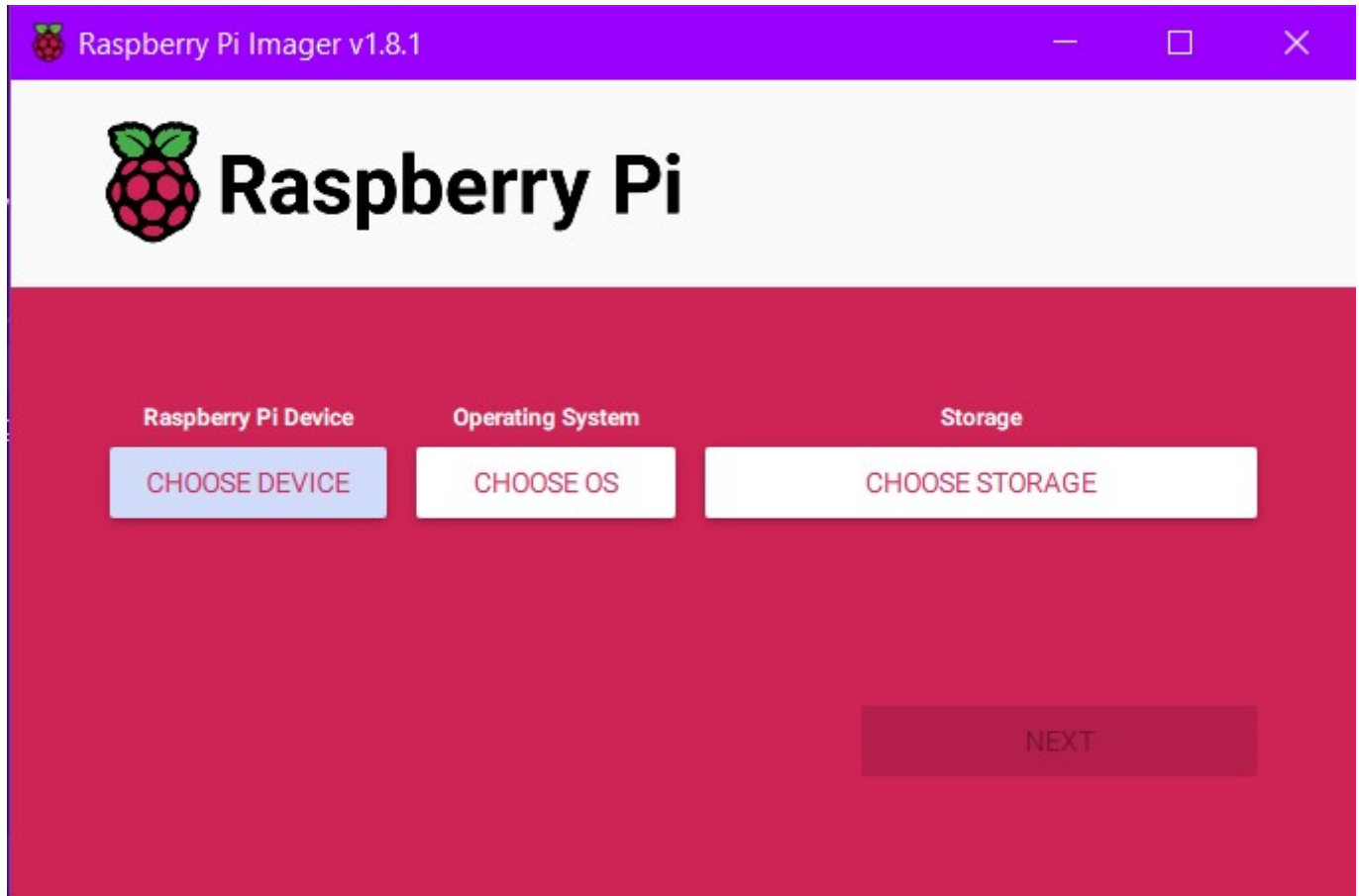


Serwer

Poniżej znajduje się proces konfiguracji serwera.

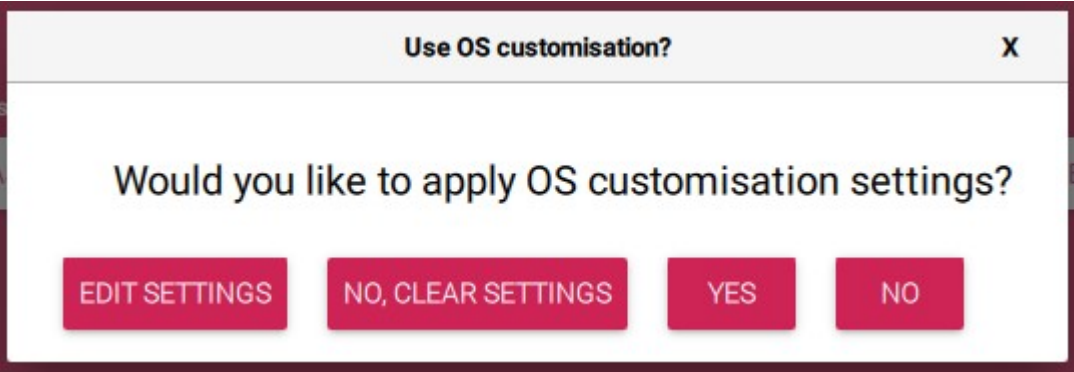
1 - Instalacja Systemu

Aby zainstalować system na Raspberry pi należy pobrać i zainstalować **Raspberry Pi Imager** ze strony [Raspberry Pi](#). Po zainstalowaniu i uruchomieniu programu pojawi się taki ekran aplikacji :

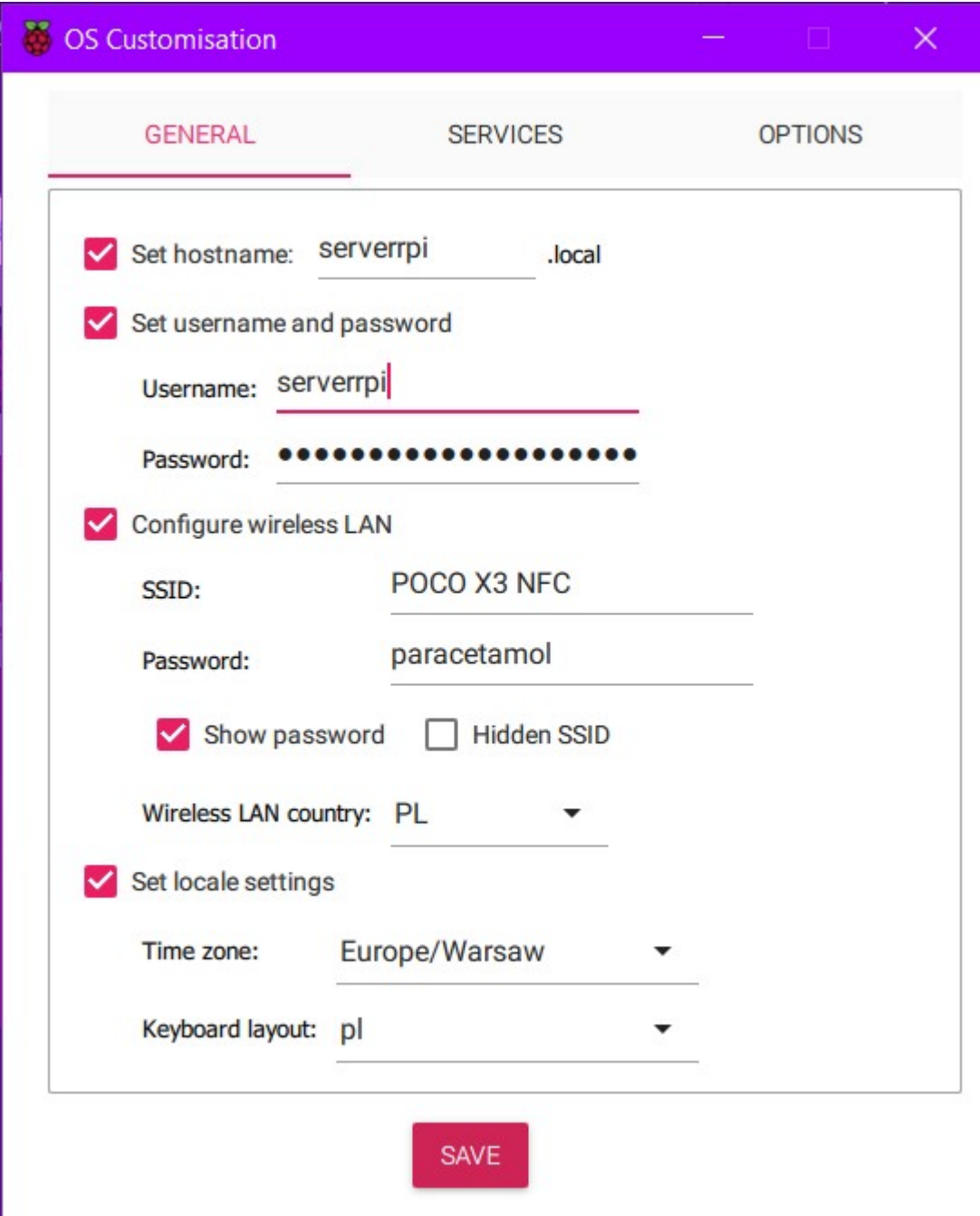


Należy wtedy wybrać odpowiednie urządzenie z którego korzystamy. Z zakładki system operacyjny wybieramy **Raspberry Pi OS Legacy**, a następnie naszą kartę SD i klikamy dalej. Aplikacja powinna nas zapytać o to czy

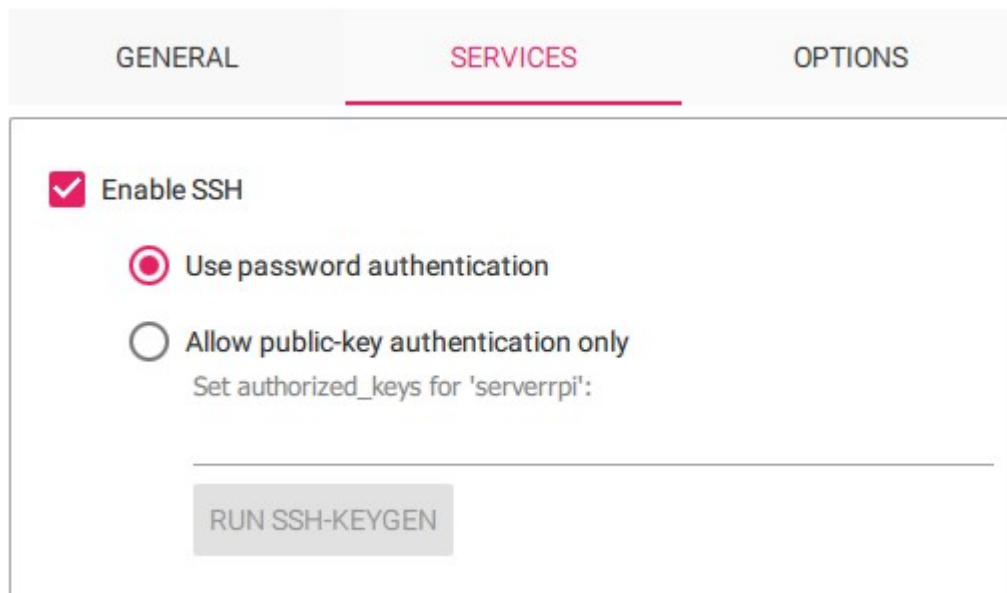
chcemy użyć niestandardowych ustawień systemu.



Klikamy edytuj ustawienia i uzupełniamy jak na poniższym zdjęciu. Hasło poniżej to **3Dprinter**.



Następnie przechodzimy do zakładki **SERVICES** i załączmy SSH.



The screenshot shows the 'SERVICES' tab of the Raspberry Pi configuration tool. Under the 'SSH' section, the 'Enable SSH' checkbox is checked. Below it, there are two radio button options: 'Use password authentication' (which is selected) and 'Allow public-key authentication only'. Under the second option, there is a text field labeled 'Set authorized_keys for 'serverrpi':'. At the bottom of the SSH section, there is a button labeled 'RUN SSH-KEYGEN'.

Zapisujemy ustawienia i Rozpoczynamy proces wygrywania systemu na kartę.

2 - Instalacja Oprogramowania

2.1 Łączenie z serwerem

Aby zainstalować oprogramowanie należy wsadzić kartę do raspberry pi podpiąć zasilanie i podłączyć do internetu kablem lub przez WiFi. W komputerze podpiętym do tego samego internetu należy uruchomić terminal połączyć się z serwerem przez ssh wpisując komendę :

```
ssh serverrpi@serverrpi
```

Podczas pierwszego łączenia terminal zapyta się czy na pewno chcemy się połączyć zatwierdzamy tą akcję wpisując **yes** do terminala, następnie wpisujemy hasło i zatwierdzamy enterem.

W razie problemów z łączeniem można też spróbować zamiast nazwy serwera użyć jego ipv4 np. :

```
ssh serverrpi@192.168.0.107
```

Może wystąpić też problem o istnieniu już takiego urządzenia w liście kluczy należy wtedy edytować plik z kluczami i usunąć urządzenie.

2.2 Aktualizacja

Przed rozpoczęciem jakiegokolwiek instalacji oprogramowania należy zainstalować aktualizacje poprzez wpisanie komendy :

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

Akcję tą trzeba potwierdzić wpisując **Y** do konsoli. Aktualizacja może zająć trochę czasu w zależności od połączenia internetowego. Może pojawić się też dodatkowa interakcja podczas instalacji należy wtedy postępować według instrukcji. Poniżej znajduje się przykładowe zdjęcie z tej operacji.

```

serverpi@serverpi: ~ $ sudo apt-get update && sudo apt-get upgrade
Get:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease [15.0 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye InRelease [23.6 kB]
Get:3 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf Packages [13.2 MB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf Packages [313 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian bullseye/non-free armhf Packages [106 kB]
Fetched 13.7 MB in 10s (1,386 kB/s)
Reading package lists... Done
N: Repository 'http://raspbian.raspberrypi.org/raspbian bullseye InRelease' changed its 'Suite' value from 'stable' to 'oldstable'
N: Repository 'http://archive.raspberrypi.org/debian bullseye InRelease' changed its 'Suite' value from 'stable' to 'oldstable'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  adduser base-files bind9-host bind9-libs bluez-firmware chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra cpio cups cups-browsed
  cups-client cups-common cups-core-drivers cups-daemon cups-filters cups-filters-core-drivers cups-ipp-utils cups-ppdc cups-server-common curl dbus
  dbus-user-session dbus-x11 distro-info-data ffmpeg file firmware-atheros firmware-brcm80211 firmware-libertas firmware-misc-nonfree firmware-realtek
  ghostscript gstreamer1.0-alsa gstreamer1.0-plugins-bad gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-x libaom0 libavcodec58
  libavdevice58 libavfilter7 libavformat58 libavresample4 libavutil56 libblas3 libbsd0 libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dbg
  libc6-dev libcamera-apps libcamera-tools libcamera0 libcups2 libcupsfilters1 libcupsimage2 libcurl3-gnutls libcurl4 libdbus-1-3 libflac8 libfontembed1
  libgs9 libgs9-common libgssapi-krb5-2 libgstreamer-gli.0-0 libgstreamer-plugins-bad1.0-0 libgstreamer-plugins-base1.0-0 libjavascriptcoregtk-4.0-18
  libjson-c5 libk5crypto3 libkrb5-3 libkrb5support0 liblapack3 libmagic-mgc libmagic1 libncurses6 libncursesw6 libpam-systemd libpostproc55
  libprotobuf-lite23 librsvg2-2 librsvg2-common libssh-gcrypt-4 libssl1.1 libswresample3 libswscale5 libsynctex2 libsystemd0 libtinfo5 libtinfo6 libudev1
  libvlc-bin libvlc5 libvlccore9 libvpx6 libwebkit2gtk-4.0-37 libwebp6 libwebpdemux2 libwebpmux3 libx11-6 libx11-data libx11-xcb1 libxpm4 locales
  logrotate lxplug-bluetooth lxplug-cputemp lxplug-menu lxplug-network lxplug-ptbatt lxplug-updater ncurses-base ncurses-bin ncurses-term openssh-client
  openssh-server openssh-sftp-server openssl python3-flask python3-libcamera python3-picamera2 python3-sense-hat python3-werkzeug raspberrypi-sys-mods
  raspi-config realvnc-vnc-server rp-prefapps rpi-eeprom rpi-imager ssh systemd systemd-sysv systemd-timesyncd udev vlc vlc-bin vlc-data vlc-l10n
  vlc-plugin-access-extra vlc-plugin-base vlc-plugin-notify vlc-plugin-qt vlc-plugin-samba vlc-plugin-skins2 vlc-plugin-video-output
  vlc-plugin-video-splitter vlc-plugin-visualization xserver-common xserver-xorg-core xwayland
154 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 320 MB of archives.
After this operation, 127 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Po ukończeniu instalacji należy uruchomić ponownie urządzenie i połączyć się ponownie. Uruchomić ponownie można poprzez wpisanie komendy :

```
sudo reboot
```

2.3 Instalacja Docker'a

Następnym krokiem jest pobranie skryptu instalacyjnego do Docker'a.

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

Po pobraniu skryptu należy go uruchomić komendą

```
sudo sh get-docker.sh
```

Poniżej zdjęcie z terminalu procesu części instalacji :

```
serverpi@serverpi: ~  
Server: Docker Engine - Community  
Engine:  
  Version:      24.0.7  
  API version:  1.43 (minimum version 1.12)  
  Go version:   go1.20.10  
  Git commit:   311b9ff  
  Built:        Thu Oct 26 09:08:26 2023  
  OS/Arch:      linux/arm  
  Experimental: false  
containerd:  
  Version:      1.6.25  
  GitCommit:    d8f198a4ed8892c764191ef7b3b06d8a2eeb5c7f  
runc:  
  Version:      1.1.10  
  GitCommit:    v1.1.10-0-g18a0cb0  
docker-init:  
  Version:      0.19.0  
  GitCommit:    de40ad0  
  
=====
```

To run Docker as a non-privileged user, consider setting up the Docker daemon in rootless mode for your user:

```
dockerd-rootless-setuptool.sh install
```

Visit <https://docs.docker.com/go/rootless/> to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root users access, refer to <https://docs.docker.com/go/daemon-access/>

WARNING: Access to the remote API on a privileged Docker daemon is equivalent to root access on the host. Refer to the 'Docker daemon attack surface' documentation for details: <https://docs.docker.com/go/attack-surface/>

```
=====
```

serverpi@serverpi: ~ \$

Po zainstalowaniu Docker'a należy dodać użytkownika do grupy użytkowników Docker

```
sudo usermod -aG docker ${USER}
```

Aby aplikacje uruchamiały się wraz z systemem należy dodać Docker'a do serwisów uruchamianych przy starcie :

```
sudo systemctl enable docker
```

2.4 Instalacja Docker-compose

Aby zainstalować Docker-compose potrzebny jest moduł pip3:

```
sudo apt-get install libffi-dev libssl-dev  
sudo apt install python3-dev  
sudo apt-get install -y python3 python3-pip
```

Następnie należy zainstalować Docker-compose wpisując komendę :

```
sudo pip3 install docker-compose
```

2.5 Instalacja aplikacji

Aby zainstalować aplikacje należy przenieść się od głównego folderu wpisując :

```
cd
```

Następnie należy utworzyć folder containers w którym będą przetrzymywane pliki aplikacji :

```
mkdir containers
```

Utworzyć plik docker-compose.yml :

```
touch containers/docker-compose.yml
```

Następnie należy wkleić zawartość pliku **

, znajdującego się w folderze "smog\config filse for server**", do pliku na serwerze. Plik na serwerze można edytować komendą :

```
nano containers/docker-compose.yml
```

Aby zapisać plik na serwerze należy kliknąć ctrl+s i ctrl+x.

Instalowanie aplikacji rozpoczyna się wpisując komendę :

```
sudo docker-compose -f containers/docker-compose.yml up -d
```

3 - Konfiguracja oprogramowania

3.1 Konfiguracja Mosquitto

Do skonfigurowania Mosquitto trzeba utworzyć plik konfiguracyjny

```
sudo touch containers/config/mosquitto.conf
```

Następnie należy wkleić zawartość pliku **mosquitto.conf**, znajdującego się w folderze "**smog\config filse for server**", do pliku na serwerze. Plik na serwerze można edytować komendą :

```
sudo nano containers/config/mosquitto.conf
```

Następnym plikiem który musimy utworzyć jest plik z hasłami :

```
sudo touch containers/config/pwfile
```

Musimy jeszcze zmienić właściwości dostępu do pliku. Robimy to wpisując komendy :

```
sudo chmod 0700 containers/config/pwfile  
sudo chown root containers/config/pwfile  
sudo chgrp root containers/config/pwfile
```

Aby dodać użytkownika oraz zapisać zanim wprowadzone wcześniej musimy znać ID kontenera. Takich informacji dostarczy nam komenda :

```
sudo docker ps
```

Na początku musimy uruchomić ponownie nasz kontener :

```
sudo docker restart <container-id>
```

Podczas podawania ID nie trzeba przepisywać całego ID wystarczy ciąg dzięki któremu można go odróżnić od innych kontenerów (może to być nawet jeden znak, w tym przypadku mogło by to być tylko "a"). Poniżej zdjęcie ze znalezieniem ID oraz restartem kontenera :

```

serverrpi@serverrpi:~$ sudo docker ps
CONTAINER ID   IMAGE                                NAMES      COMMAND                  CREATED        STATUS        PORTS
77ab699b466d   nodered/node-red:latest            node-red   "/entrypoint.sh"        About an hour ago   Up About an hour   0.0.0.0:1880->1880/tcp, :::1880->1880/tcp
80/tcp
bdf522a58672   grafana/grafana:10.1.2             grafana    "/run.sh"               About an hour ago   Up About an hour   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
00/tcp
f7487c682322   portainer/portainer-ce:latest      portainer  "/portainer"            About an hour ago   Up About an hour   8000/tcp, 9000/tcp, 0.0.0.0:9443->9443/tcp
443/tcp, :::9443->9443/tcp
a979d5a9a3b4   eclipse-mosquitto                  mosquitto  "/docker-entrypoint..." About an hour ago   Up 8 minutes      0.0.0.0:1883->1883/tcp, :::1883->1883/tcp
83/tcp
1522e111c94c   influxdb:1.8.10                   influxdb   "/entrypoint.sh infl..." About an hour ago   Up About an hour   0.0.0.0:8086->8086/tcp, :::8086->8086/tcp
86/tcp
serverrpi@serverrpi:~$ sudo docker restart a97
a97
serverrpi@serverrpi:~$

```

Żeby dodać użytkownika musimy połączyć się ze wewnętrznym terminalem aplikacji :

```
sudo docker exec -it <container-id> sh
```

W konsoli aplikacji wpisujemy :

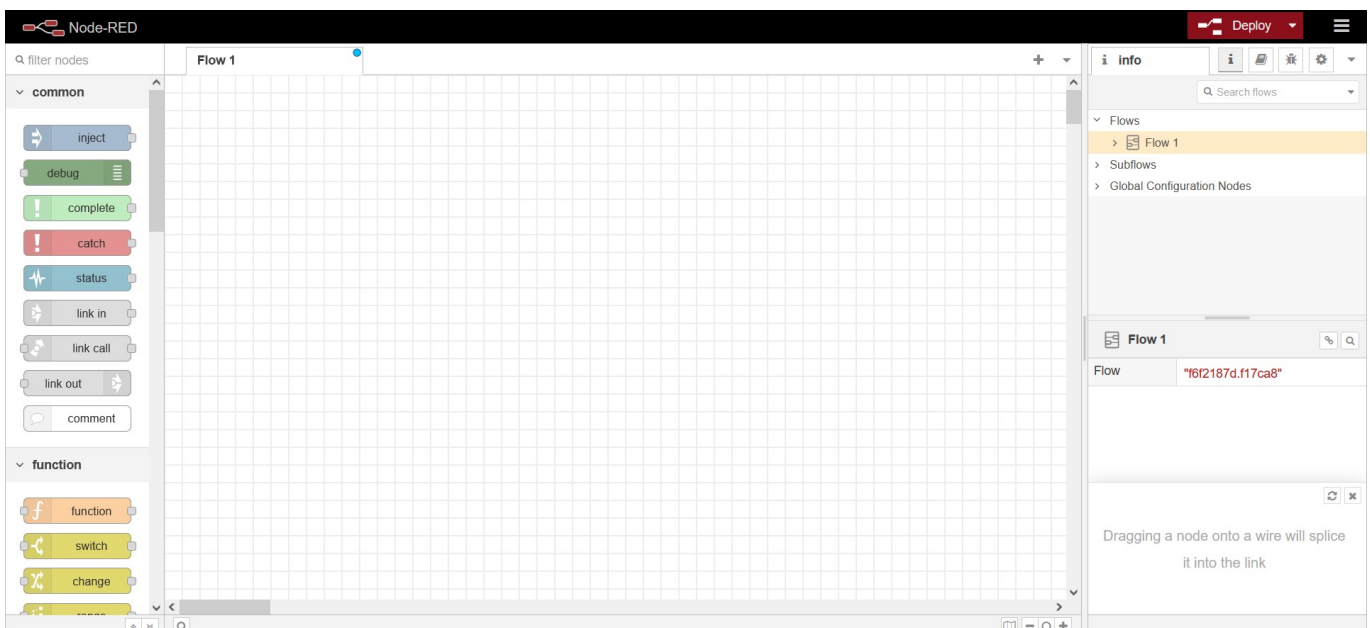
```
mosquitto_passwd -c /mosquitto/config/pwfile user
```

Następnie podajemy hasło **3Dprinter** i wychodzimy z terminala aplikacji wpisując **exit** i uruchamiamy ponownie kontener :

```
sudo docker restart <container-id>
```

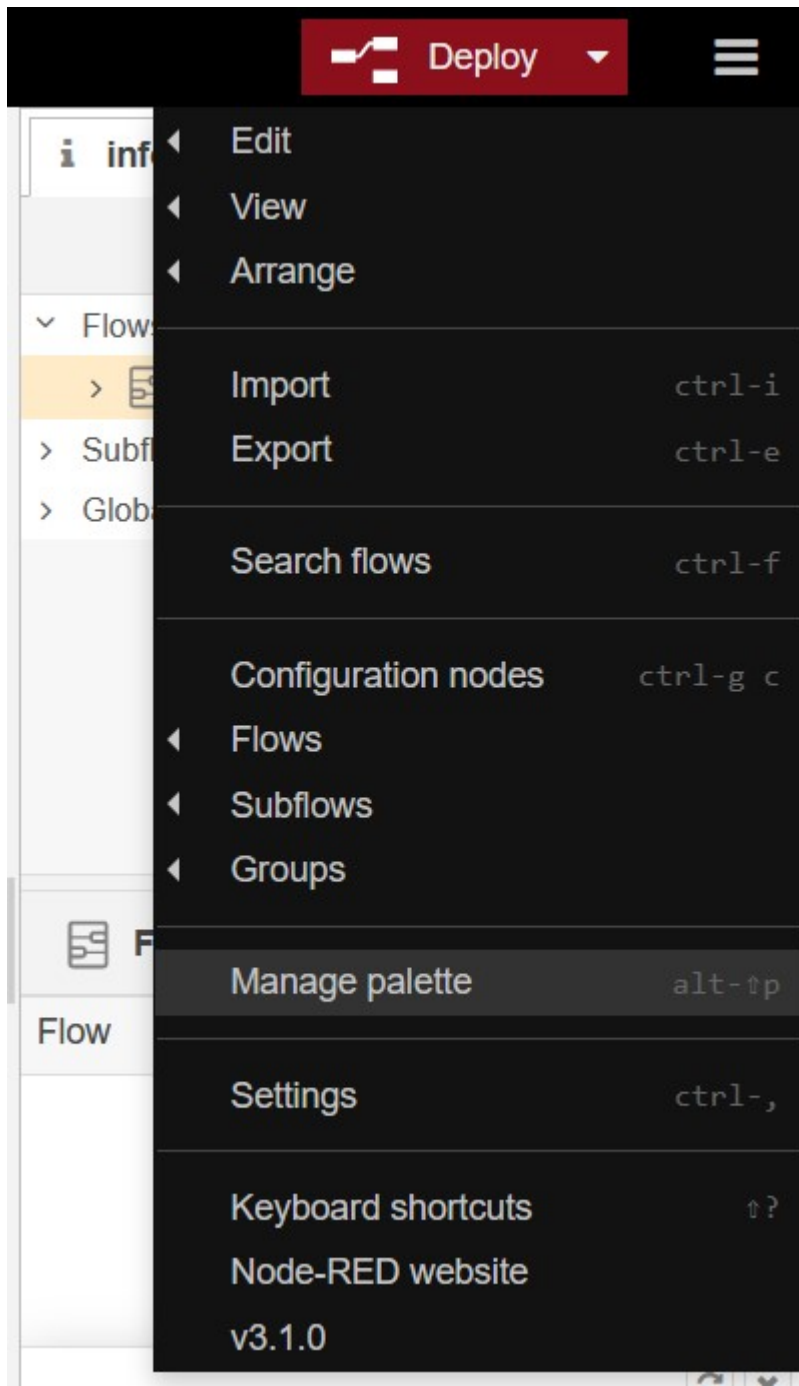
3.2 Konfiguracja Node-Red

Uruchom dowolną przeglądarkę i połącz się z witryną **serverrpi:1880**. Powinien ci się ukazać interface Node-Red.



W razie problemów z połączeniem możesz spróbować uruchomić ponownie komputer i serwer lub spróbować połączyć się po ip serwera.

W prawy górnym rogu należy kliknąć na ikonkę rozwijania menu i wybrać **Manage palette**.



W oknie którym wyskoczyło należy wybrać zakładkę **Install** i zainstalować dwa rozszerzenia:

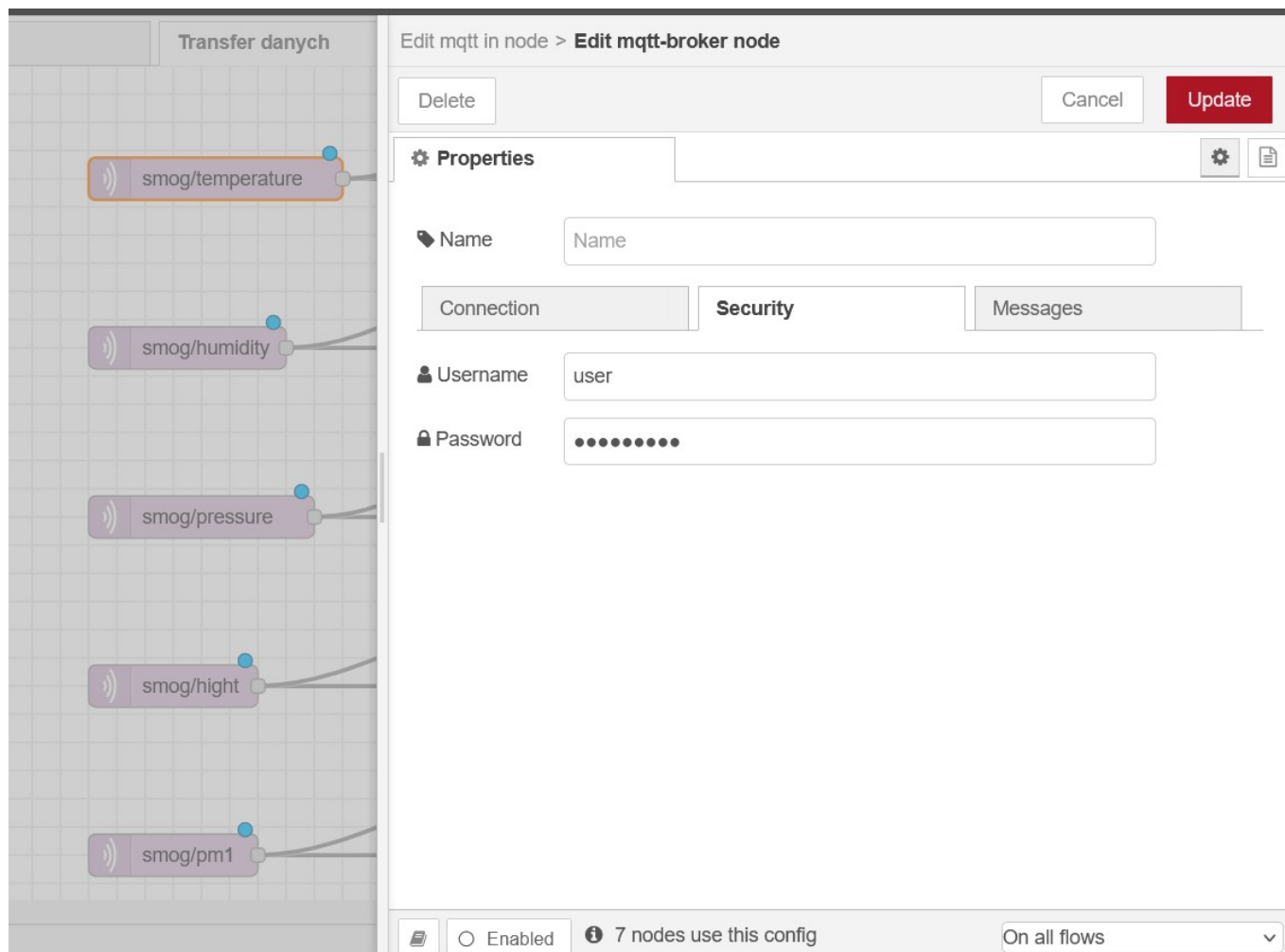
- node-red-contrib-influxdb
- node-red-dashboard

Po zainstalowaniu rozszerzeń należy zamknąć okno z rozszerzeniami i z menu wybrać zakładkę import i załadować plik **flows.json** znajdujący się w folderze "**smog\config filse for server**"

Następnie wejść do nowej zakładki **Transfer danych** i dwa razy kliknąć lewym przyciskiem myszy na różowy węzeł i kliknąć w ikonę edycji **Servera**. Klikamy w zakładkę **Security** i wpisujemy wcześniej skonfigurowane ustawienia serwera mosquitto czyli :

Username: User

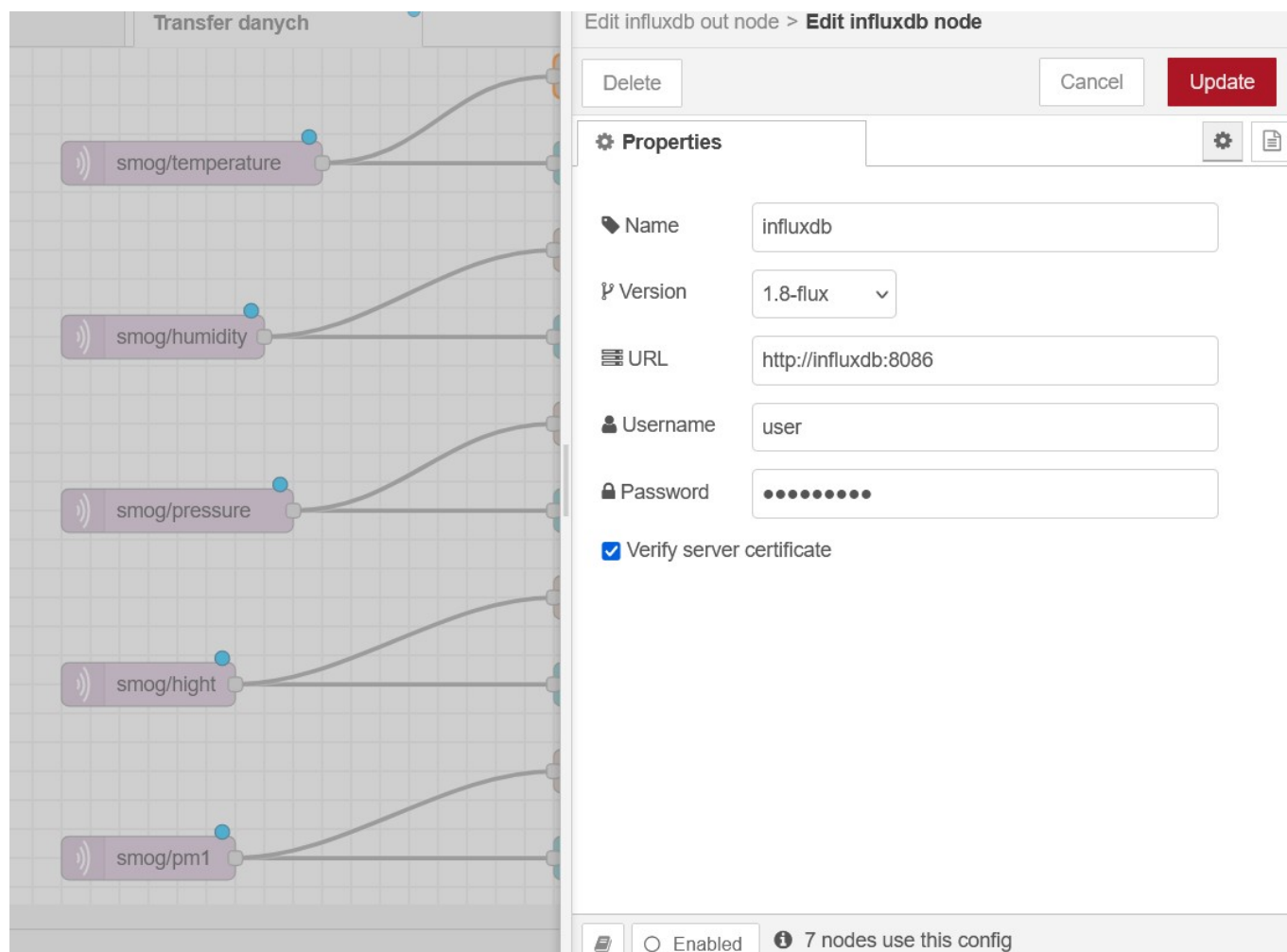
Password: 3Dprinter



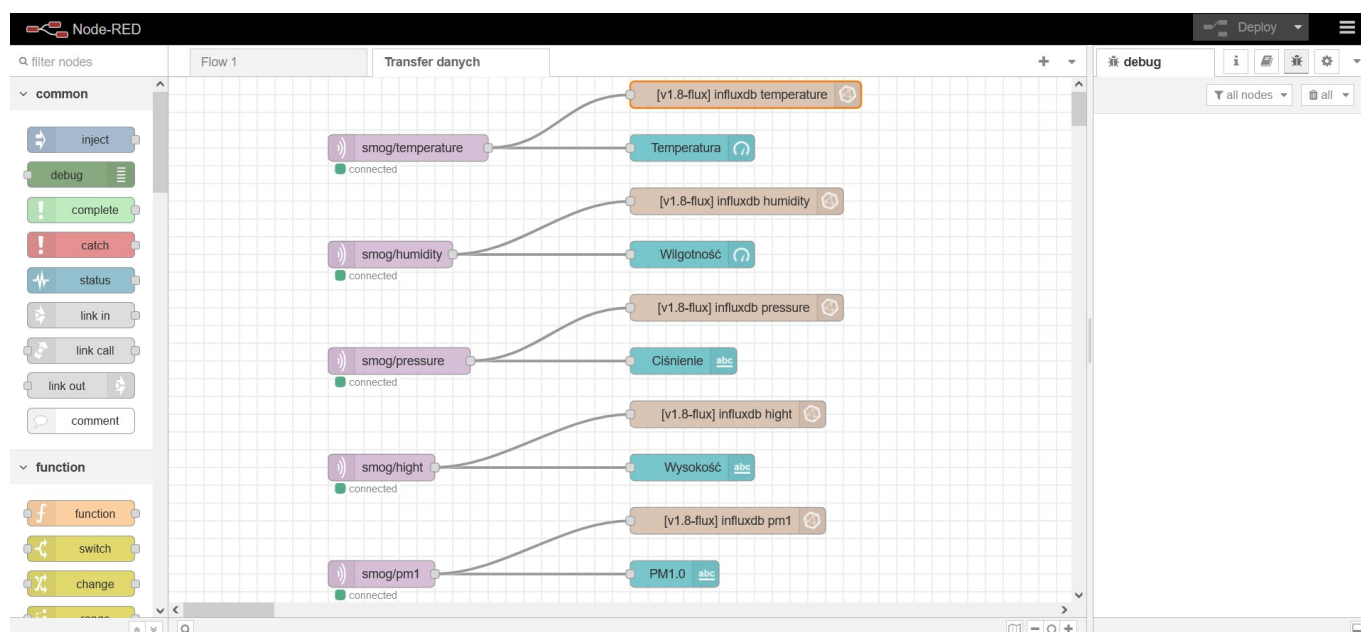
Następnie aktualizujemy zmiany i klikamy dwa razy w brązowy węzeł. Tam klikamy w ikonkę edycji **Servera**. Tam tak ja w przypadku poprzedniego węzła wpisujemy dane :

Username: User

Password: 3Dprinter



Tak jak poprzednio aktualizujemy zmiany i na sam koniec klikamy czerwony przycisk **Deploy** w prawym górnym rogu. Po poprawnym skonfigurowaniu pod różowymi węzłami powinien pojawić się zielony kwadrat z napisem connected i konsoli debugowania nie powinien pojawić się żaden błąd (konsola debugowania znajduje się po prawej stronie interface'u i jest oznaczona ikonką robaka). Zdjęcie poprawnie skonfigurowanego Node-Red :



3.3 Konfiguracja Grafana

TODO

Klient

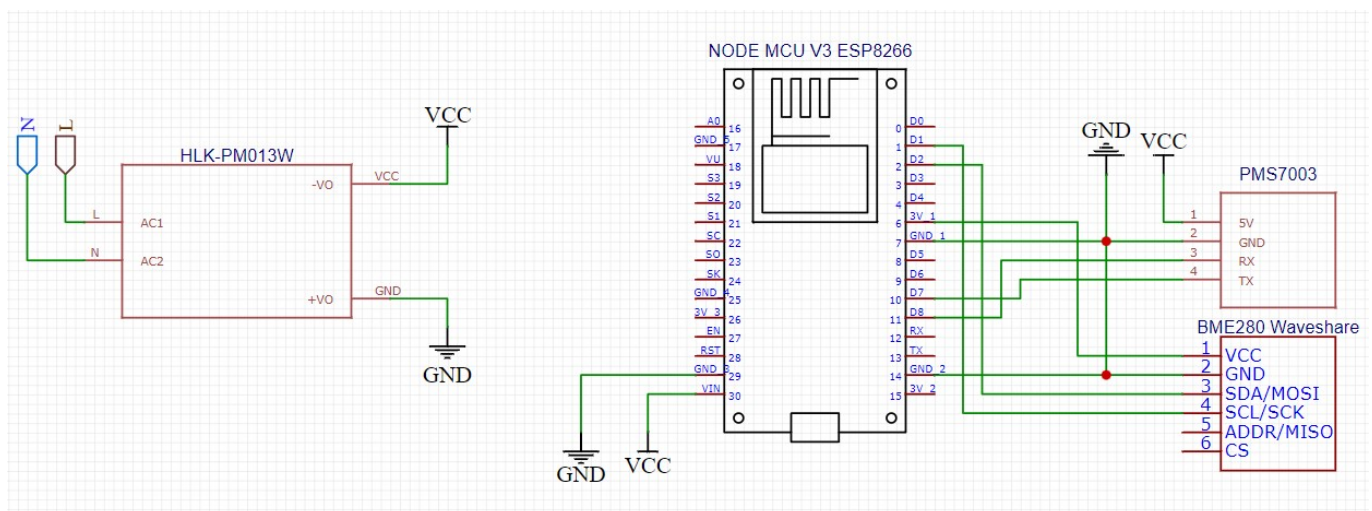
Poniżej znajdują się informacje na konfiguracji esp8266.

1 - Lista części

Klient smogomierza składa się z :

- Płytki mikrokontrolera (ESP8266)
- Zasilacza (HLK-5M05)
- Czujnika pyłu (PMS7003)
- Czujnika wilgotności, temperatury oraz ciśnienia (BME280)

Poniżej schemat połączenia części :



Zasilacza (HLK-5M05) przekształca napięcie przemienne do stałego i zasila **wszystkie czujniki**. **Czujnik pyłu (PMS7003)** jest połączony z **mikrokontrolerem (ESP8266)** przez interfejs **URAT**, natomiast **czujnik warunków atmosferycznych (BME280)** jest połączony poprzez interfejs **I2C**.

2 - Program do ESP8266

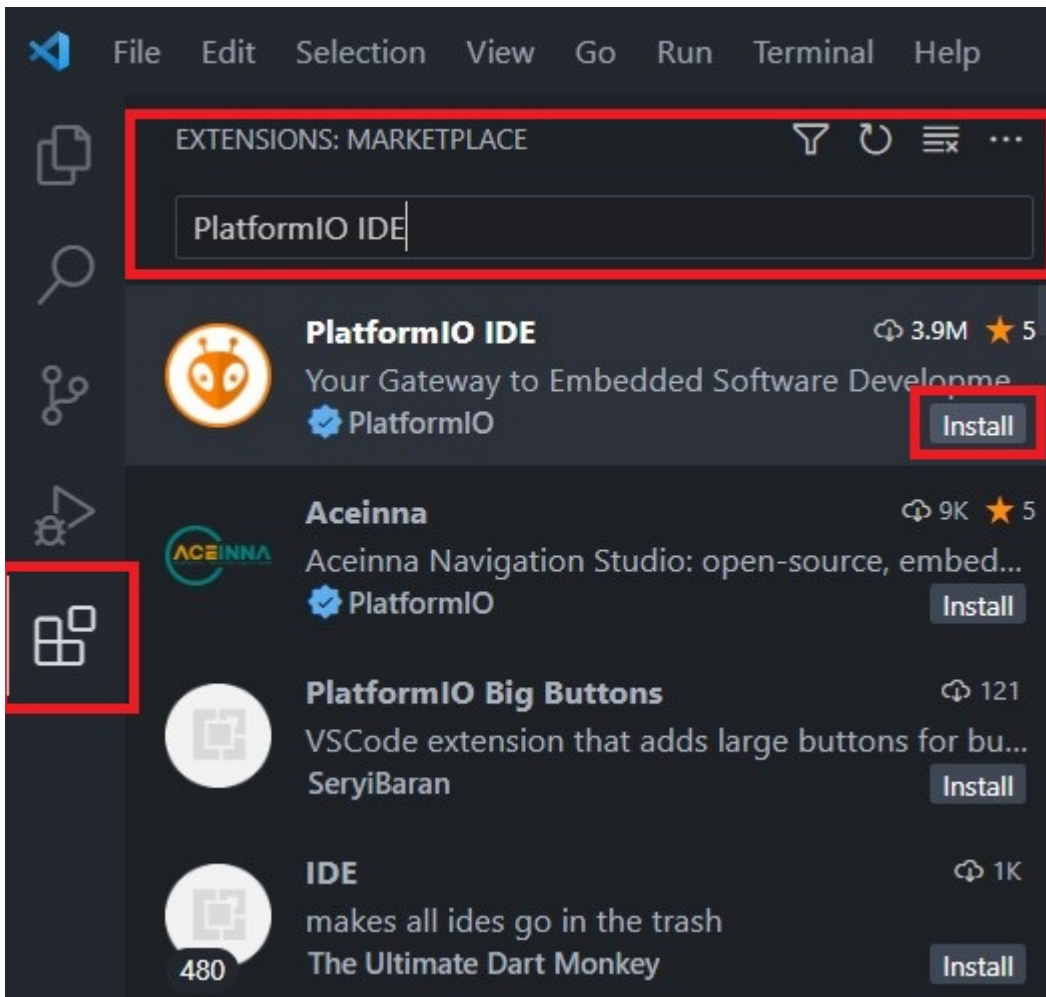
2.1 Instalacja oprogramowania

Do wgrania i edytowania kodu jest potrzebny program **Visual Studio Code** który można poprać [tutaj](#).

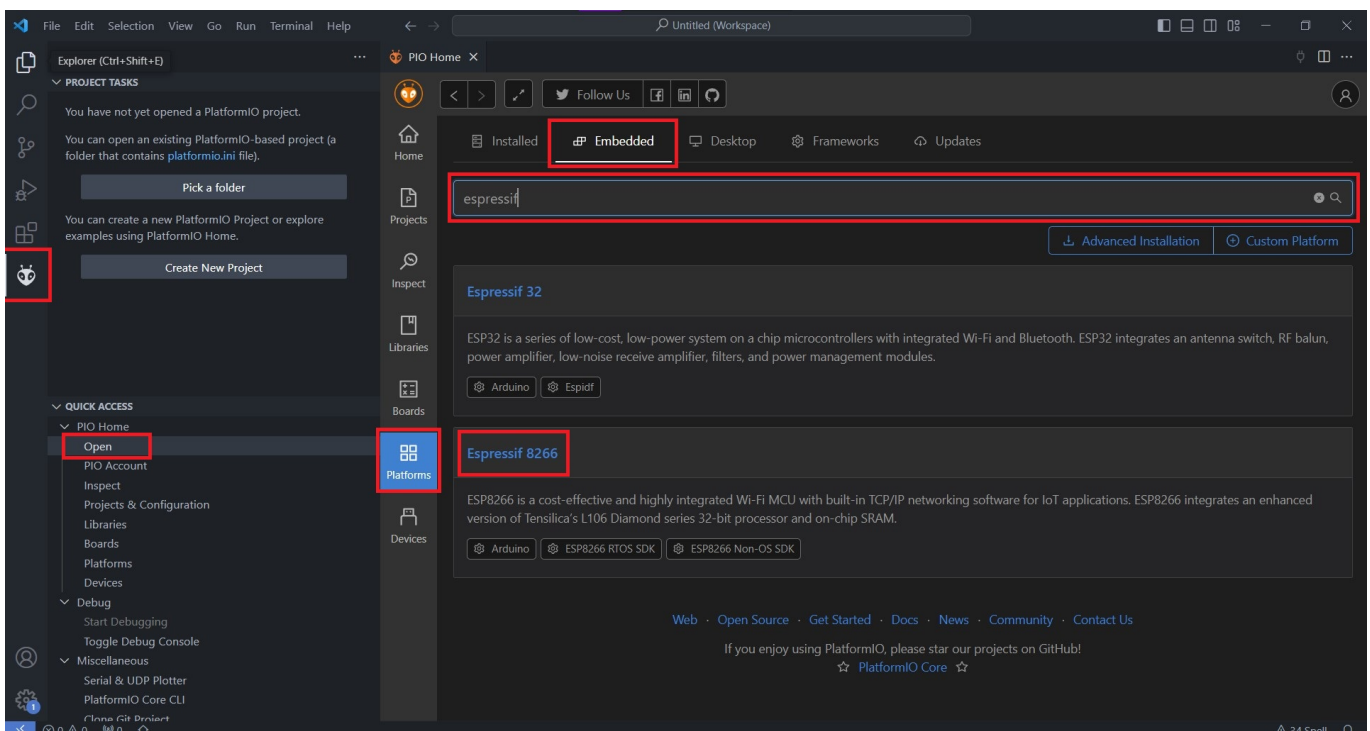
Po pobraniu instalatora należy go uruchomić i podążać za jego krokami.

2.2 Konfiguracja środowiska

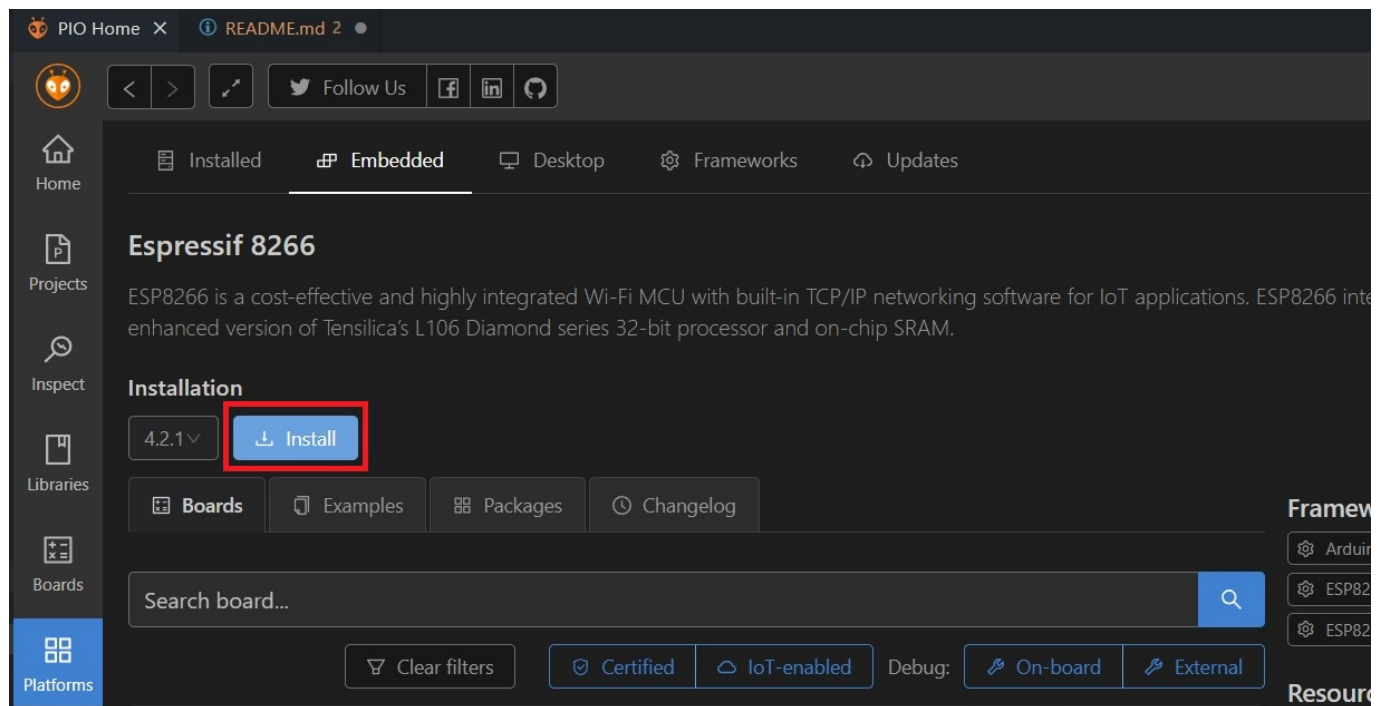
Aby kompilować kod będzie nam potrzebne **środowisko**, aby je zainstalować należy kliknąć w przycisk **Extensions** i wyszukać w pasku wyszukiwania **PlatformIO IDE**, a następnie kliknąć **Install**, jak pokazano na zdjęciu poniżej :



Następnym krokiem jest dodanie **płytki ESP8266** do środowiska aby to osiągnąć należy na pasku po lewej stronie kliknąć w nowo dodaną ikonkę **PlatformIO**, a następnie w przycisk **Open** aby otworzyć **interfejs Środowiska PlatformIO IDE**. Następnie klikamy w **Platforms**, zakładkę **Embedded**, wpisujemy w wyszukiwarkę **espressif** i klikamy w **Espressif 8266**. Poniżej zdjęcie procesu :

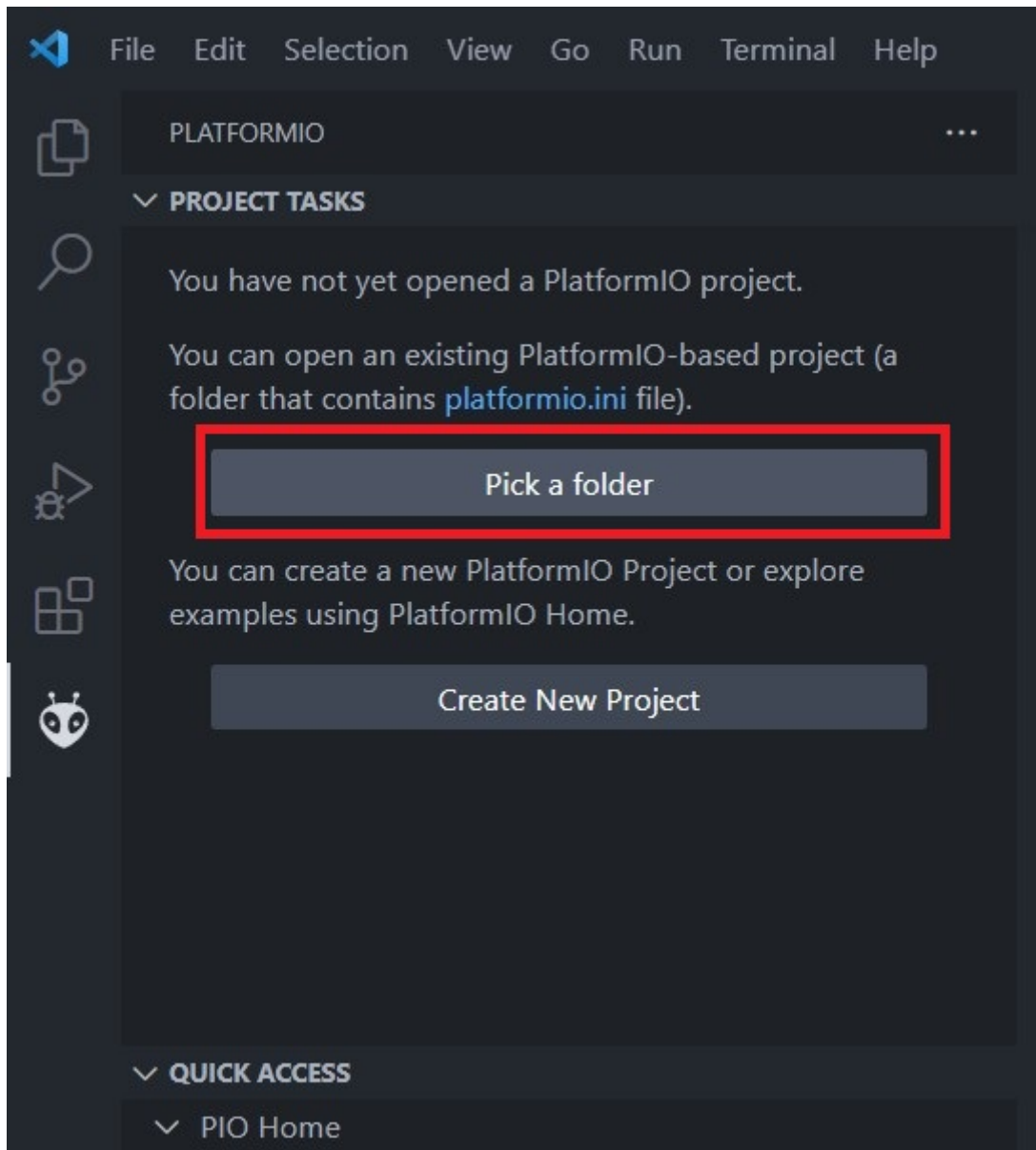


Następnie klikamy install **Install**. Po poprawnym zainstalowaniu powinien wyskoczyć komunikat.



2.3 Konfiguracja, kompilacja oraz wgrywanie projektu

Na początku musimy otworzyć folder obszaru roboczego robimy to poprzez naciśnięcie przycisku **Pick a folder** i wybieramy folder o nazwie **smog_esp8266**



Następnie w **main.cpp** w **linijce 14** należy zmienić **SSID sieci** na **nazwę sieci** do której jest połączony server, w **linijce 15** **hasło** należy zamienić na **hasło sieci** i w **linijce 18** **podane IP** należy zmienić na **IP servera**.

Aby wgrać program do **ESP8266** należy w dolnym pasku kliknąć w ikonkę **Upload**. Poniżej zdjęcie z umiejscowieniem ikony:

