

[WiFi Halow] Configuración

- Preparación del entorno
 - Instalación del Sistema Operativo
 - Preparación del Software
 - Publicador
 - Suscriptor
- Conexión por Wi-Fi HaLow
 - AP
 - STA
- Ejecución y visualización
 - Publicador
 - Suscriptor

Preparación del entorno

Instalación del Sistema Operativo

- Descargamos la [imagen](#) del sistema operativo y la flasheamos a las respectivas tarjetas MicroSD, usando un software como puede ser [Balena Etcher](#).
- Podremos acceder a las Raspberry Pi por SSH con los ajustes por defecto:
 - Hostname: *raspberrypi*
 - Usuario: *pi*
 - Contraseña: *raspberrypi*

Preparación del Software

Para preparar todo el software, necesitaremos que las Raspberry Pi tengan conexión a internet. Realizaremos el siguiente proceso para cada placa:

- Conectamos la Raspberry Pi a un ordenador Windows (lo llamaremos host) mediante un cable Ethernet. El host deberá tener conexión a internet, ya sea por Wifi o por otro puerto Ethernet.
- En el host, presionamos "Win + R" y escribimos el comando "*ncpa.cpl*".
- Hacemos click derecho sobre la conexión de red en la que tenemos acceso a internet, y hacemos click en "Propiedades".
- Accedemos a la pestaña "Uso compartido", y habilitamos la casilla "Permitir que los usuarios de otras redes se conecten a través de la conexión a Internet de este equipo".
- Accedemos a la Raspberry Pi por SSH, y abrimos el fichero "*/etc/dhcpd.conf*".

- Lo editaremos de forma que nos queden las siguientes líneas. Es posible que ya existan, o que estén comentadas:

```
interface eth0
static ip_address=<RASPBERRY_IP>/24
static routers=<HOST_IP>
static domain_name_servers=8.8.8.8
```

- Para saber qué IPs debemos usar, abriremos una terminal en el host, y pondremos el comando "*ipconfig*". Nos fijaremos en la sección correspondiente al adaptador Ethernet.
- La <HOST_IP> se corresponde con la "*IPv4 Address*", y la <RASPBERRY_IP> se corresponde con cualquier IP dentro del mismo rango. Por ejemplo, si la <HOST_IP> es "*192.168.137.1*", podremos usar como <RASPBERRY_IP> la "*192.168.137.2*".



Si no funciona, desde los ajustes de "*Red e internet*" en el host, establecemos manualmente la configuración IPv4 para el puerto Ethernet.

- Reiniciamos la placa para aplicar los cambios.

```
sudo reboot
```

Una vez hecho esto, ya tendremos conexión a internet. A partir de ahora, deberemos diferenciar las placas. Una será la dedicada a simular el barco (la llamaremos publicador) y la otra será el servidor de tierra (la llamaremos suscriptor), en la que también configuraremos el broker MQTT.

Publicador

- Instalamos el módulo "*paho-mqtt*" de Python.

```
pip3 install paho-mqtt
```

- Instalamos el módulo "*opencv-python*" de Python.

```
pip3 install opencv-python
```

- Compilamos las librerías necesarias.

```
wget https://github.com/xianyi/OpenBLAS/archive/refs/tags/v0.3.23.tar.gz
tar -xvzf v0.3.23.tar.gz
cd OpenBLAS-0.3.23
make TARGET=ARMV7 NO_LAPACK=1 DEBUG=0 -j$(nproc)
sudo make install
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libcblas.so.3
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libf77blas.so.3
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libatlas.so.3
```



El proceso de compilación tarda aproximadamente 15 minutos.

- Abrimos el fichero "*/etc/environment*" y añadimos la siguiente línea:

```
LD_LIBRARY_PATH="/opt/OpenBLAS/lib"
```

- Creamos un fichero "*pub_sensores.py*" con [este](#) código.
- Creamos un fichero "*pub_camara.py*" con [este](#) código.

Suscriptor

- Instalamos el módulo "*paho-mqtt*" de Python.

```
pip3 install paho-mqtt
```

- Instalamos el módulo "*opencv-python*" de Python.

```
pip3 install opencv-python
```

- Compilamos las librerías necesarias.

```
wget https://github.com/xianyi/OpenBLAS/archive/refs/tags/v0.3.23.tar.gz
tar -xvzf v0.3.23.tar.gz
cd OpenBLAS-0.3.23
make TARGET=ARMV7 NO_LAPACK=1 DEBUG=0 -j$(nproc)
sudo make install
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libcblas.so.3
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libf77blas.so.3
sudo ln -s /opt/OpenBLAS/lib/libopenblas.so /opt/OpenBLAS/lib/libatlas.so.3
```



El proceso de compilación tarda aproximadamente 15 minutos.

- Abrimos el fichero "*/etc/environment*" y añadimos la siguiente línea:

```
LD_LIBRARY_PATH="/opt/OpenBLAS/lib"
```

- Creamos un fichero "*sub.py*" con [este](#) código.



Este código, además de suscribirse a los tópicos MQTT necesarios, crea un servidor web para la visualización de los datos. Debido al bajo rendimiento de placas, y a la espera activa en la sección de código que lee los datos de los tópicos, la visualización de los datos en la interfaz web es casi imposible. Sería necesaria una modificación para hacerlo más eficiente.

- Creamos un directorio "*templates*", y dentro creamos un fichero "*dashboard_secciones.html*" con [este](#) código.
- Instalamos y configuramos el broker MQTT:
 - Añadimos la clave GPG de Docker.

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

- Abrimos el fichero "*/etc/apt/sources.list*" y añadimos la siguiente línea:

```
deb [arch=armhf] https://download.docker.com/linux/debian stretch stable
```

- Descargamos e instalamos los paquetes "*pigz*" y "*cgroupfs-mount*".

```
wget http://ftp.debian.org/debian/pool/main/p/pigz/pigz_2.4-1_armhf.deb
wget http://ftp.debian.org/debian/pool/main/c/cgroupfs-mount/cgroupfs-mount_1.4_all.deb
sudo dpkg -i pigz_2.4-1_armhf.deb
sudo dpkg -i cgroupfs-mount_1.4_all.deb
```

- Instalamos Docker.

```
sudo apt update
sudo apt install docker-ce -y
```

- Instalamos el contenedor con el broker MQTT.

```
sudo docker run -d --name rabbitmq --privileged --restart=always -p 1883:1883 -p 15672:15672
rabbitmq:management
sudo docker exec rabbitmq rabbitmq-plugins enable rabbitmq_mqtt
```

- Reiniciamos la placa para aplicar los cambios.

```
sudo reboot
```

Conexión por Wi-Fi HaLow

Para la conexión mediante Wi-Fi HaLow, una de las placas será el punto de acceso (la llamaremos AP) y otra la estación (la llamaremos STA). En nuestro caso, el suscriptor será el AP, y el publicador el STA.

Para ambas placas, conectaremos el módulo de Wi-Fi HaLow y su antena, y colocamos los switches del módulo según la siguiente imagen:



1	2	3	4	5	6
OFF	ON	OFF	OFF	OFF	ON

AP

- Abrimos el fichero `/etc/dhcpd.conf`, y lo editamos de forma que queden las siguientes líneas:

```
interface eth0
static ip_address=192.168.0.10/24
```

- Creamos un fichero `ap.sh` con [este](#) código.
- Le damos permisos de ejecución.

```
sudo chmod +x ap.sh
```

- Abrimos el fichero `/etc/rc.local` y añadimos la ruta al fichero `ap.sh`, por ejemplo de la siguiente forma:

```
/home/pi/RSFM/ap.sh

exit 0
```

- Reiniciamos la placa para aplicar los cambios.

```
sudo reboot
```

STA

- Abrimos el fichero `/etc/dhcpd.conf`, y lo editamos de forma que queden las siguientes líneas:

```
interface eth0
static ip_address=192.168.0.20/24
```

- Creamos un fichero `sta.sh` con [este](#) código.
- Le damos permisos de ejecución.

```
sudo chmod +x sta.sh
```

- Abrimos el fichero `/etc/rc.local` y añadimos la ruta al fichero `sta.sh`, por ejemplo de la siguiente forma:

```
/home/pi/RSFM/sta.sh

exit 0
```

- Reiniciamos la placa para aplicar los cambios.

```
sudo reboot
```

Ejecución y visualización

Si se han seguido los pasos correctamente, ambas placas estarán conectadas por WiFi HaLow a través de su interfaz `wlan0`. Para comprobarlo, podemos hacer ping a sus respectivas IPs, por ejemplo, desde el publicador:

```
ping 192.168.200.1
```

Publicador

- Conectamos los sensores y la cámara a la Raspberry Pi mediante sus puertos USB.
- Ejecutamos el programa `pub_sensores.py` para comenzar a leer los datos de los sensores, y publicarlos contra el broker MQTT.

```
python3 pub_sensores.py
```

- Ejecutamos el programa "*pub_camara.py*" para comenzar a leer los datos de la cámara, y publicarlos contra el broker MQTT.

```
python3 pub_camara.py
```

Suscriptor

- Ejecutamos el programa "*sub.py*" para conectarnos al broker MQTT y leer los datos recibidos.

```
python3 sub.py
```

- Abrimos una ventana del navegador y accedemos a <http://localhost:8080> para visualizar la transmisión de vídeo y una gráfica con los datos recogidos de los sensores.



Para visualizar correctamente la gráfica con los datos de los sensores, debemos disponer de conexión a internet desde el suscriptor. Esto se debe a que el HTML de la interfaz web emplea un script descargado de internet para el renderizado de las gráficas.

Haz click [aquí](#) para más información sobre cómo conseguir conexión a internet.