

## Protocole de streaming via wifi:

### 1. Téléchargement des modules de depthai

Dans un premier temps, télécharger les modules depthai sur la page de téléchargement du site officiel : <https://docs.luxonis.com/projects/api/en/latest/install/#install-from-pypi>

Suivre les instructions correspondants à votre système d'exploitation.

### 2. Téléchargement ESP-IDF (module Wifi-Bluetooth)

Dans un deuxième temps, il faut télécharger ESP-IDF sur le site : <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/windows-setup.html>

Je vous conseille de télécharger la version : ESP-IDF v 4.1.2 car c'est avec cette version que le test a été fait.

Téléchargez ce fichier dans votre C : \Users\nom\_d'utilisateur , sous le nom de ESP

### 3. Programmation dans Window Powershell

Nous pouvons maintenant commencer la programmation.

Lorsque vous téléchargez pour la première fois ESP-IDF v 4.1.2, celui-ci ouvre automatiquement deux pages : une page Windows PowerShell et une page commande.

Pour la suite nous allons utiliser seulement la page Windows PowerShell.

Dans un premier temps, il vous faut taper les commandes suivantes :

```
cmd.exe
```

```
export.bat
```

Une fois que cela est fait, il vous faut télécharger le GitHub correspondant aux recherches déjà effectuées via la commande suivante :

```
git clone https://github.com/luxonis/esp32-spi-message-demo.git
```

Cela va vous télécharger les fichiers dans votre dossier de travail actuel.

Afin d'éviter un problème de longueur de chemin d'accès lors des futures commandes, je vous conseille de renommer « esp-spi-message-demo » par « esp32 ».

Le fichier qui nous intéresse se nomme « mjpeg\_streaming\_wifi », je vous conseille de même que précédemment de le renommer « wifi » .

La prochaine étape consiste à télécharger les modules qui pourraient manquer. Pour cela il vous faut tout d'abord entrer dans le fichier « esp32 » :

```
cd esp32
```

Puis de faire :

```
git submodule update --init --recursive
```

Maintenant, nous allons rentrer dans le fichier « wifi » :

```
cd wifi
```

**IMPORTANT** : une erreur de chemin existe dans les dossiers. Pour y remédier il vous faut, via votre explorateur de fichier, ouvrir le dossier « main » puis ouvrir « CMakeLists.txt » A l'intérieur il vous faut modifier \$ENV{IDF\_PATH} :

```
# Create library
set(SOURCES
    main.cpp
    file_server.cpp
)

set(INCLUDE_DIRS
    .
    $ENV{IDF_PATH}/examples/common_components/protocol_examples_common/include
)

set(REQUIRES
    esp_http_server
    depthai-spi-api
    protocol_examples_common
)

idf_component_register(SRCS ${SOURCES} INCLUDE_DIRS ${INCLUDE_DIRS} REQUIRES ${REQUIRES} PRIV_REQUIRES ${PRIV_REQUIRES} EMBED_FILES "favicon")
```

Par le chemin d'accès de votre esp-idf-v4.1.2 :

```
# Create library
set(SOURCES
    main.cpp
    file_server.cpp
)

set(INCLUDE_DIRS
    C:/Users/floma/Espressif2/frameworks/esp-idf-v4.1.2/examples/common_components/protocol_examples_common/include
)

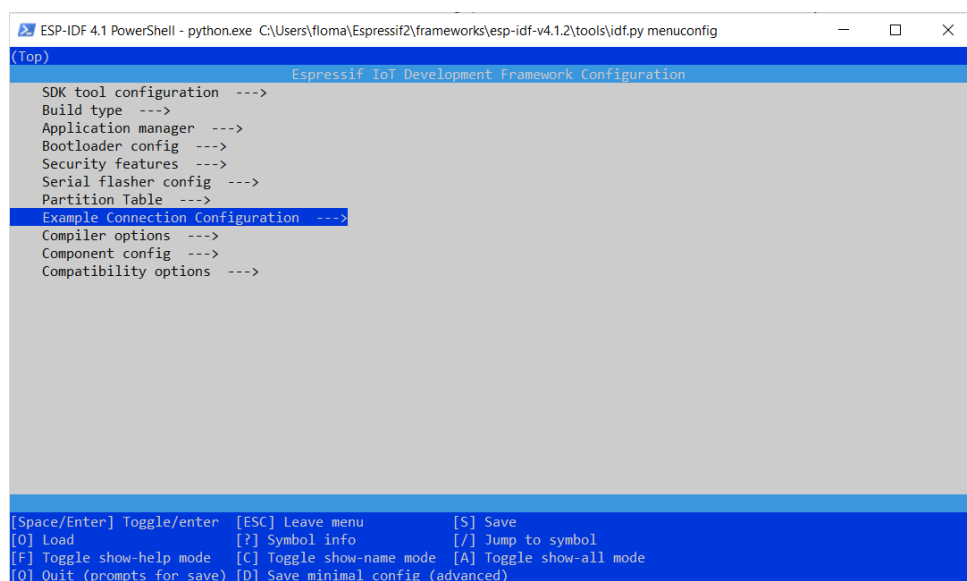
set(REQUIRES
    esp_http_server
    depthai-spi-api
    protocol_examples_common
)

idf_component_register(SRCS ${SOURCES} INCLUDE_DIRS ${INCLUDE_DIRS} REQUIRES ${REQUIRES} PRIV_REQUIRES ${PRIV_REQUIRES} EMBED_FILES "favicon")
```

Une fois que cette modification est effectuée, vous pouvez effectuer la commande suivante :

**idf.py menuconfig**

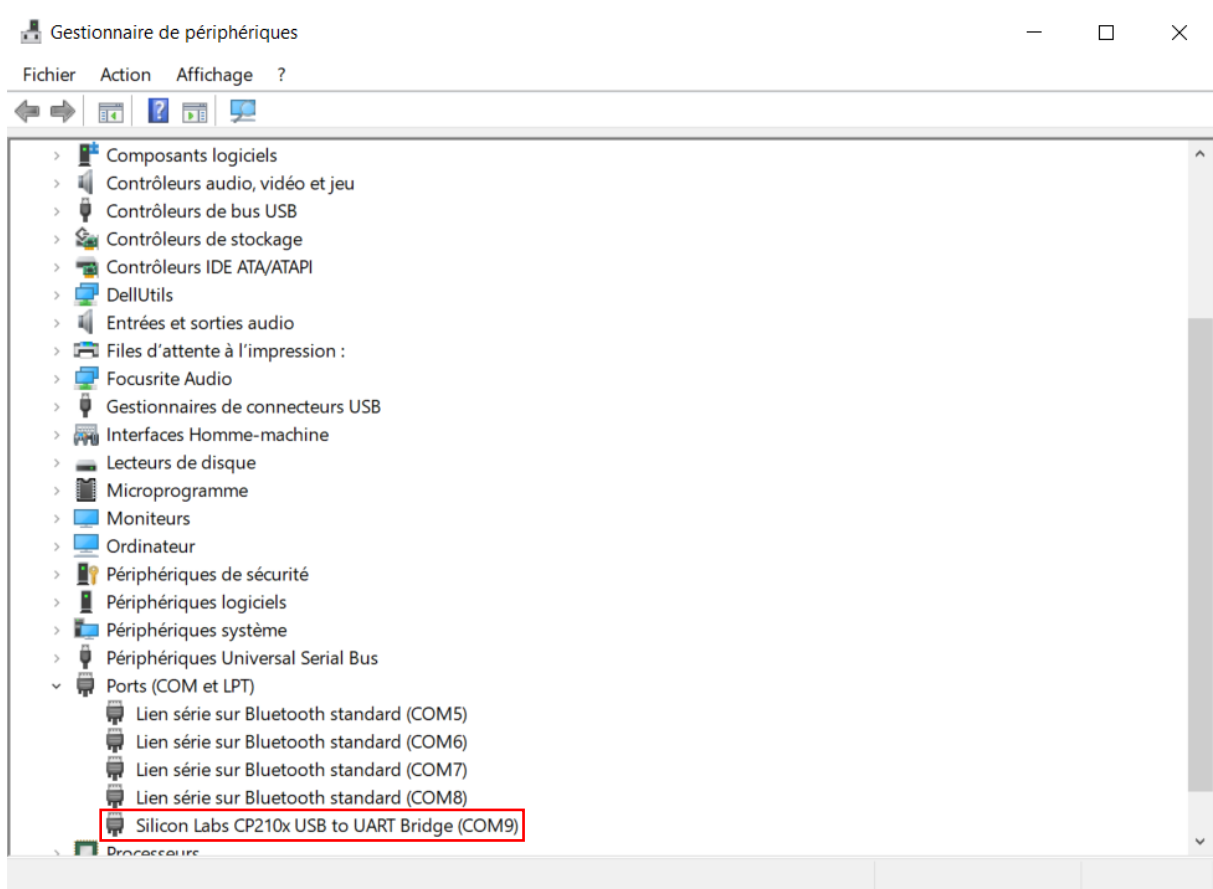
Elle permet d'ouvrir une page comme celle-ci :



Dirigez-vous dans l'onglet « Example Connection Configuration » puis remplacez « Wifi SSID » par le nom du wifi sur lequel est connecté votre ordinateur et « Wifi Password » par son mot de passe. Il

faut impérativement que ces données correspondent au réseau internet sur lequel est connecté l'ordinateur.

Ensuite, il vous faut ouvrir le « Gestionnaire de périphérique » sur votre ordinateur et noté le port COM sur lequel est connecté votre ESP32 (USB B) :



Ensuite il vous faut « flash » votre carte avec la commande suivante :

```
idf.py -p PORT flash
```

Remplacez PORT par:

- **COM9** si vous êtes sur Windows
- **/dev/ttyUSB9** si vous êtes sur linux
- **/dev/cu.USB9** si vous êtes sur macOS

Il vous faut ensuite noter l'adresse IP correspondant à votre système en utilisant :

```
idf.py -p PORT monitor
```

```
I (21728) example_connect: Connected to AndroidAP
I (21738) example_connect: IPv4 address: 192.168.43.103
I (21738) example_connect: IPv6 address: fe80:0000:0000:0000:26d7:ebff:feab:56a0
I (21748) MJPEG Server: Starting HTTP Server
I (21758) gpio: GPIO[2]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 1| Pulldown: 0| Intr:2
```

Une fois que cela est fait, vous pouvez faire CTRL+C afin de sortir de la manipulation et il ne vous reste plus qu'à lancer le programme main.py ou main\_lightweight.py :

```
python main_lightweight.py
```

Ouvrez ensuite votre navigateur et allez à l'adresse correspondant à votre IP dans notre cas :

<http://192.168.43.103/>

Vous pouvez maintenant regarder votre stream en wifi.