

Instalación

La máquina virtual entregada se encuentra sin configurar para poder "experimentar" el proceso desde 0. Por ello, antes de la instalación de la extensión de ESP-IDF para VScode, es recomendable instalar los requisitos que nos indican desde ESP-IDF en la web:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-macos-setup.html>

Concretamente, para Ubuntu recomiendan:

- `sudo apt-get update`
- `sudo apt-get install git wget flex bison gperf python3 python3-venv cmake ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0`

Puedes consultar los paquetes instalados con

- `apt list --installed`

Tras instalar dichos paquetes, se debería proceder a la instalación de la extensión siguiendo los pasos de:

<https://github.com/espressif/vscode-esp-idf-extension/blob/master/docs/tutorial/install.md>

Si, al comenzar la instalación, aún tuviéramos algún problema con el módulo *pip* de Python3, entonces procederíamos a instalar:

- `sudo apt install python3-pip`

Tras finalizar la instalación, podríamos probar a ejecutar alguno de los ejemplos siguiendo los pasos de:

https://github.com/espressif/vscode-esp-idf-extension/blob/master/docs/tutorial/basic_use.md

(nota: el paso 6 de esas insrtucciones puede no ser necesario. El contenido por defecto de `c_cpp_propoerties.json` es válido)

Sobre la conexión serie

Como mencioné en clase, nuestra comunicación con el dispositivo será a través del puerto serie (UART). Pero el conector que usaremos es del de USB. La placa que usamos tiene un chip que hace la función de *Bridge USB-to-UART*. Para poder comunicar desde nuestro PC/portátil con ese chip (y así, comunicarnos a su vez con el [ESP32](#)), debemos tener los drivers correspondientes instalados en nuestro equipo.

En <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/establish-serial-connection.html> se indica que Espressif suele usar dos tipos de chips: de FTDI y el CP210x (de Silicon Labs). En el enlace anterior encontraréis información para encontrar los drivers y cómo comprobar su correcta instalación en Windows.

Asimismo, recordad que si usáis la máquina virtual, tras conectar el dispositivo a vuestro PC/portátil, **deberéis reclamar el dispositivo desde la propia máquina virtual** (a través de Virtual Box) para que esté disponible en el sistema operativo *guest*.

Grupo *dialout* en Linux

Por defecto, cuando conectamos nuestra placa a un puerto USB en nuestro PC/portátil Linux, se creará un fichero de dispositivo con un nombre similar a `/dev/ttyUSB0`. Dicho fichero pertenecerá al usuario root y al grupo `dialout`. Si nuestro usuario no está en dicho grupo, deberemos añadirlo.

Podemos comprobar los grupos en los que está nuestro usuario con el comando:

- `groups`

Si no aparece el grupo `dialout`, entonces deberemos incluir nuestro usuario en el grupo. Para hacerlo, podemos usar la orden:

- `sudo usermod -a -G dialout $USER`

Suele ser necesario reiniciar la máquina para que el cambio surta efecto.



Última modificación: miércoles, 28 de septiembre de 2022, 19:31

