SETTING UP VSCODE FOR ESP-IDF

Follow this tutorial to install ESP-IDF for VSCode: <https://github.com/espressif/vscode-esp-idf-extension/blob/master/docs/tutorial/install.md>. If ESP-IDF is already installed, select *Use existing setup*. Otherwise, select *Express*, which will install ESP-IDF and its tools.

Upon installation, Espressif logo will appear on the left side, from where you can access the extension.

Graphical user interface, application

Description automatically generated

You can configure your ESP-IDF extension, access examples, and create a new project from template by clicking **View > Command Palette**.

Graphical user interface, text, application

Description automatically generated

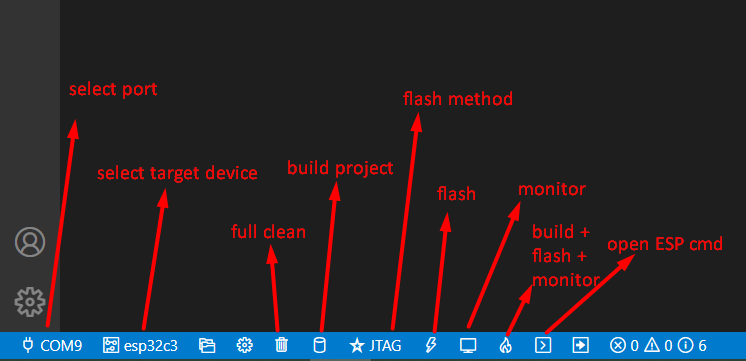
Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

After creating a project, the bottom bar will look something like this:



**Before building your project, you need to configure PORT and DEVICE TARGET.**

**Before flashing, you need to configure FLASH METHOD.**

You need to setup the port (of ESP-PROG) and Espressif device target by clicking on the bar. Available ports, devices and flash methods will appear on top.

After selecting the device target, the following will appear:

Graphical user interface, text, application, email

Description automatically generated

Where you select the first option if you're using ESP-PROG.

For **flash method**, if you are debugging using JTAG and ESP-PROG, select **JTAG.** Otherwise, if you've simply plugged your ESP32 into the PC with a USB cabel, select **UART.**

**Note**: if you haven't flashed this project yet, option „JTAG“ will not appear until you've clicked on „Flash device“ .

If you use the „Build + flash + monitor“, you will get an output, but there's no debug option.

For configuring the debugger, you have to add a new debug configuration, and select „ESP-IDF Debug: Launch“.

Graphical user interface, text, application

Description automatically generated Text

Description automatically generated

And that's it! The debugger is now ready to use!