Last updated 27 May 2024

1. **Source an ESP-PROG JTAG debugger board**
   1. Low-cost clones of this Espressif debugger board are widely available on eBay and AliExpress for around £10.
   2. Download and install the FTDI drivers for the ESP-Prog from here: <https://ftdichip.com/drivers/vcp-drivers/>, the actual download link is: <https://ftdichip.com/wp-content/uploads/2021/08/CDM212364_Setup.zip>
   3. Install Zadig USB driver management tool from: <https://zadig.akeo.ie/>
   4. Plug the ESP-PROG into a reliable USB socket i.e. mainboard integrated.
   5. Open Windows Device Manager to see two new COM ports in the ports section.
   6. Start Zadig and click OPTIONS > LIST ALL DEVICES.
   7. Look for "Dual RS232-HS (Interface 0)" and replace the FTDI driver with WinUSB (v6.1.7600.16385), or similar.
   8. Check in Device Manager that one of the two new COM ports has disappeared and been replaced by Dual RS232-HS in the Universal Serial Bus Devices section, at the bottom of the Device Manager list. Dual RS232-HS is the interface that PlatformIO uses to communicate with the target ESP32 hardware via the JTAG bus.
2. **Download and install Visual Studio Code (VSCode)**
   1. Download from: <https://code.visualstudio.com>
   2. For installation video see: <https://www.youtube.com/watch?v=cu_ykIfBprI>
3. **Install PlatformIO extension**
   1. Open VSCode Extension Manager
   2. Search for official PlatformIO IDE extension
   3. Install PlatformIO IDE
4. **Create a new C++ main.cpp file**
   1. Click the PlatformIO extension button in the activity bar at the left
   2. Under quick access click PIO Home > Open
   3. Then in the HOME tab click NEW PROJECT
      1. Give the project a name
      2. Choose a board by searching on a partial name e.g. ESP32 Dev…
      3. Choose a FRAMEWORK e.g. Arduino
   4. An example main.cpp
      * #include <Arduino.h> //part of PlatformIO Arduino framework
      * #include "Arduino\_GFX\_Library.h" //in lib folder
      * #include "RunningAverage.h" //in lib folder
      * // put function declarations here:
      * int count = 0;
      * static unsigned long counter = 5;
      * void setup() {
      * // put your setup code here, to run once:
      * pinMode(2,OUTPUT);
      * Serial.begin(115200); // send and receive at 9600 baud
      * }
      * void loop() {
      * // put your main code here, to run repeatedly:
      * // turn the LED off (HIGH is the voltage level)
      * digitalWrite(LED\_BUILTIN, HIGH);
      * // wait for a second
      * delay(500);
      * // turn the LED on by making the voltage LOW
      * digitalWrite(LED\_BUILTIN, LOW);
      * // wait for a second
      * delay(500);
      * count++;
      * counter++;
      * printf("The value of count is: %d \n", count);
      * }
5. **Including Arduino libraries**
   1. Copy library folder from e.g. GFX\_Library\_for\_Arduino from:
      1. E:\Users\Steven\Documents\GitHub\Arduino\_IDE2\Arduino\_IDE2\_Sketches\libraries\
   2. Paste into e.g.
      1. E:\Users\Steven\Documents\GitHub\PlatformIO\Blink\_ESP32\_and\_debug\lib
   3. The added libraries will appear in the “lib” section of the project in the explorer activity Ctrl + Shift + E
6. **PlatformIO.ini file**
   1. An example configuration is shown below:
      1. [env:esp32doit-devkit-v1] ;environment specifier
      2. platform = espressif32 ;device platform
      3. board = esp32doit-devkit-v1 ;board type
      4. framework = arduino ;programming framework
      5. build\_type = debug ;build in debug mode, not release mode
      6. debug tool = esp-prog ;debugging board name
      7. debug\_speed = 5000 ;JTAG protocol clock speed in KHz, between 1000 to 5000
      8. debug\_build\_flags = -O0 -g -ggdb ;leave everything as is, e.g. do not optimise variables out of existence
      9. debug\_init\_break = tbreak setup ;creates hidden breakpoint at first line of setup() function
      10. upload\_protocol = esp-prog ;use for upload as well as debug
      11. monitor\_speed = 115200 ;serial monitor COM port speed (defaults to 115200 anyway)
      12. monitor\_port = COM3 ;COM port used for serial operations like printf
7. **Build**
   1. Click the tick ✔ on the STATUS bar at the bottom, or the tab bar at the top in the main panel or pressing Ctrl + Alt + S to build an ESP32 image.
8. **Upload**
   1. Click the right arrow ⇾ on the STATUS bar at the bottom, or the tab bar at the top in the main panel or pressing Ctrl + Alt + U to upload an ESP32 image to the target hardware. The target hardware will immediately run the executable.
9. **Debugging**
   1. See: <https://docs.platformio.org/en/latest/plus/debugging.html>
   2. Debug flags: <https://gcc.gnu.org/onlinedocs/gcc/Debugging-Options.html>
   3. On the menu bar click Run > Start Debugging, or press F5.
   4. Code will be built in debug mode, if not already done, uploaded and the debugger started which will stop at the first line of setup().
   5. See: <https://www.youtube.com/watch?v=ENLhW0MQuu0&t=33s>
   6. Stop debugging from menu Run > Stop Debugging or press Shift + F5, or press the red square button in the debugger control bar.
   7. Conditional Breakpoints are very useful but get the syntax correct e.g. count == 100 will break when the variable count reaches 100, but count = 100 just sets the value of the variable to 100!
10. **Clean**
    1. Sometimes problems can be eliminated using the CLEAN button (Rubbish bin icon), which removes the build folder located at:
       1. E:\Users\Steven\Documents\GitHub\PlatformIO\Blink\_for\_ESP32\.pio\build
11. **View output using Serial Monitor**
    1. Open a Serial Monitor using the AND gate symbol on the STATUS bar at the bottom, or the tab bar at the top in the main panel or pressing Ctrl + Alt + B.
12. **How to:**
    1. Change the default location for new PIO projects:
       1. Highlight the TERMINAL tab in the bottom panel, make visible with Ctrl + J
          1. Look on the right pane of the bottom panel where different terminals are listed, new ones can be added using the +↓ e.g. Bash shell and PowerShell et cetera. Look for and select PlatformIO CLI (command line interface), where PIO commands can be executed. For more info see: <https://docs.platformio.org/en/latest/core/userguide/cmd_settings.html>
          2. View the existing default location by entering:
             1. pio settings get or pio settings get projects\_dir
          3. Set a new default path using e.g.
             1. pio settings set projects\_dir E:\Users\Steven\Documents\GitHub\PlatformIO
          4. Delete the old projects folder (copy any projects into the new projects folder before doing this).
          5. Close and re-open VSCode & PIO
          6. Verify default project path has been changed by opening PIO Home, clicking the New Project button and then hovering the mouse cursor over the ? next to the ***Use default location*** tick box – the default location will pop-up.
          7. Note that the PIO Workspace is just a place to which project folders can be added and removed, it does not have to be moved when moving the default projects path.
          8. After building a project the PlatformIO: Build terminal will be active and will appear in the list of terminal types in the right pane of the bottom panel.
          9. Show multiple tabs in main panel, if closed:
             1. View > Appearance > Tab Bar > Multiple Tabs
13. **Problems**
    1. Any unusual errors during upload or debug that have no obvious solution:
       1. Delete the folder with the target device name in the build folder at this type of path:
          1. E:\Users\Steven\Documents\GitHub\PlatformIO\Blink\_ESP32\_and\_debug\.pio\build\esp32doit-devkit-v1
14. **Useful web resources**
    1. <https://docs.platformio.org/en/latest/plus/debugging.html#piodebug>
    2. <https://freelance-drupal.com/en/blog/debug-arduino-code>
15. **Support Forum**
    1. https://community.platformio.org/