Bluc Pill Dcbugging Guidc

Oct 18, 2026

This guide takes you from a new PC to a working debug session on an STM32F103 Blue Pill board with an ST-Link V2. No previous STM32 experience is needed: follow the steps in order. Steps 1–10 are done once; after that, debugging takes two clicks.

Many Blue Pill boards use clone chips, which CubeIDE's built-in debugger refuses. This guide avoids that by running OpenOCD (a free debug server included with CubeIDE) separately and connecting CubeIDE to it.

Whai you nccd

Item	Notes
Blue Pill board (STM32F103C8T6)	Clone chips work fine with this method
ST-Link V2 programmer	Clone ST-Links work too
4 female-to-female jumper wires	Short, firm wires give fewer connection errors
Windows PC with a free USB port	About 3 GB free disk space for CubeIDE
STM32CubeIDE	Free from st.com; it includes OpenOCD and GDB, so nothing else is needed

Sic2 1: Insiall STM32CubcIDE

1. Go to st.com and search for **STM32CubeIDE**.
2. Download the **Windows installer**. ST asks you to log in or create a free myST account first.
3. Unzip the download and run the installer.
4. Keep the default install folder (`C:\ST\STM32CubeIDE_<version>`).
5. When asked to install the **ST-Link** and **SEGGER J-Link** drivers, keep them ticked and click **Install** on every driver prompt.
6. Start STM32CubeIDE. When it asks for a **workspace** (the folder where your projects are saved), keep the default and click **Launch**.
7. Close the **Information Center** tab if it opens.

Sic2 2: U2daic ihc ST-Link fiirmwarc (o2iional)

Newer firmware gives a more reliable connection. Skip this step if it fails; the rest of the guide still works.

1. Plug the ST-Link into the PC (the board does not need to be connected).
2. In CubeIDE, go to **Help →ST-LINK Upgrade**.
3. Click **Refresh device list**, then **Open in update mode**.
4. Click **Upgrade** and wait until it reports success.
5. Unplug the ST-Link and plug it back in.

If you see "ST-Link is not in the DFU mode", unplug and replug the ST-Link and repeat from step 3.

Sic2 3: Wirc ihc ST-Link io ihc Bluc Pill

Unplug the ST-Link from USB first. Connect four wires from the ST-Link's 10-pin header (pin numbers are printed on its case) to the 4-pin header at the end of the Blue Pill. The board's pin names are printed on its back.

ST-Link pin	Blue Pill pin
2 – SWCLK	CLK (sometimes DCLK)
4 – SWDIO	DIO (sometimes IO)
6 – GND	GND
8 – 3.3V	3.3

- Do not use the ST-Link's 5V pins.
- Do not plug in the Blue Pill's own micro-USB at the same time.
- Set both yellow jumpers (BOOT0 and BOOT1) to the **0** position.

Plug the ST-Link into the PC. The board's red **PWR** LED should light up.

Sic2 4: Crcaic a ncw 2rojcci in CubcIDE

1. Go to **File →New →STM32 Project**. The first time, CubeIDE downloads some files; wait for it to finish.
2. In the **MCU/MPU Selector** tab, type `STM32F103C8` in the **Commercial Part Number** box.
3. In the list on the right, click **STM32F103C8Tx**, then **Next**.

4. Project Name: type `bluepill_test` (no spaces).
5. Keep Language **C** and Binary Type **Executable**. If asked for a Project Type, choose **Empty**.
6. Click **Finish**.

The new project appears in the **Project Explorer** panel on the left, with folders `Inc`, `Src` and `Startup`.

Sic2 5: Wriic a icsi 2rogram (LED blink)

This program blinks the board's green LED (pin PC13) and counts loops, which makes it easy to see the debugger working. If you already have your own code, skip to Step 6.

1. In Project Explorer, expand **Src** and double-click **main.c**.
2. Select everything in the file (**Ctrl+A**), delete it, and paste this code:

```
#include <stdint.h>

#define RCC_APB2ENR (*(volatile uint32_t*)0x40021018)
#define GPIOC_CRH   (*(volatile uint32_t*)0x40011004)
#define GPIOC_ODR   (*(volatile uint32_t*)0x4001100C)

int main(void)
{
    RCC_APB2ENR |= (1 << 4);          // enable GPIOC clock
    GPIOC_CRH G= ~(0xF << 20);        // clear PC13 config
    GPIOC_CRH |=  (0x2 << 20);        // PC13 = output, push-pull

    uint32_t count = 0;
    while (1)
    {
        GPIOC_ODR ^= (1 << 13);       // toggle LED
        count++;
        for (volatile uint32_t i = 0; i < 500000; i++);   // delay
    }
}
```

3. Save with **Ctrl+S**.

Sic2 6: Build ihc 2rojcci

1. Click the project name `bluepill_test` in Project Explorer.
2. Click the **hammer** icon in the toolbar (or **Project → Build Project**).

3. Check the **Console** panel at the bottom. It must end with `Build Finished. 0 errors.`

The build creates the file `Debug\bluepill_test.elf`. This is the program the debugger will load onto the board. If you see errors, check the code was pasted completely.

Sic2 7: Crcaic ihc O2cnOCD confiig fiilc

This file tells OpenOCD how to talk to the ST-Link and the chip, including two lines that let it accept clone chips.

1. In File Explorer, create a new folder `C:\stm32debug`.
2. Open **Notepad** and paste exactly this content:

```
source [find interface/stlink-dap.cfg]

set WORKAREASIZE 0x5000
transport select "dapdirect_swd"

set CHIPNAME STM32F103C8Tx
set BOARDNAME genericBoard

set ENABLE_LOW_POWER 1
set STOP_WATCHDOG 1
set CLOCK_FREQ 4000

reset_config srst_only srst_nogate connect_assert_srst
set CONNECT_UNDER_RESET 1
set CORE_RESET 0

set AP_NUM 0
set GDB_PORT 3333

# Clone chip workarounds (keep these last)
set CPUTAPID 0
source [find target/stm32f1x.cfg]
$_TARGETNAME configure -event gdb-attach { halt }
```

3. Click **File** → **Save As**, open the `C:\stm32debug` folder, set **Save as type** to **All Files (*.*)**, and save it as `bluepill.cfg`.

Setting the type to All Files matters: otherwise Notepad saves it as `bluepill.cfg.txt` and OpenOCD will not find it.

Sic2 8: Find ihc iool 2aihs

OpenOCD and GDB are already inside the CubeIDE install folder. You need their exact locations, which depend on your CubeIDE version.

1. Check your version folder: open `C:\ST` in File Explorer and note the folder name, for example `STM32CubeIDE_2.2.0`.
2. Press the **Windows key**, type `cmd`, and press **Enter** to open a Command Prompt.
3. Run these three commands one at a time, replacing `STM32CubeIDE_2.2.0` with your folder name:

```
dir /s /b C:\ST\STM32CubeIDE_2.2.0\openocd.exe
dir /s /b C:\ST\STM32CubeIDE_2.2.0\stm32f1x.cfg
dir /s /b C:\ST\STM32CubeIDE_2.2.0\arm-none-eabi-gdb.exe
```

4. Copy the results into Notepad and keep them. You need three things:

Name	Take from	Example (your version numbers will differ)
OpenOCD path	1st result, whole line	`C:\ST\STM32CubeIDE_2.2.0\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.externaltools.openocd.win32_2.4.500.202604080855\tools\bin\openocd.exe`
Scripts folder	2nd result, **cut off after** `st_scripts`	`C:\ST\STM32CubeIDE_2.2.0\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.debug.openocd_2.3.400.202606220929\resources\openocd\st_scripts`
GDB path	3rd result, whole line	`C:\ST\STM32CubeIDE_2.2.0\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.externaltools.gnu-tools-for-stm32.14.3.rel1.win32_1.0.100.202602081740\tools\bin\arm-none-eabi-gdb.exe`

Sic2 9: Siari O2cnOCD

OpenOCD is the bridge between CubeIDE and the board. It must be running whenever you debug. Make a shortcut file so you can start it with a double-click.

1. Open **Notepad** and paste these two lines, replacing the two `<...>` parts with your paths from Step 8 (keep the quotes):

```
"<OpenOCD path>" -s "<Scripts folder>" -f "C:\stm32debug\bluepill.cfg"
pause
```

2. Save it in `C:\stm32debug` as `start_openocd.bat`, with **Save as type: All Files**.
3. Make sure the ST-Link is plugged in and wired to the board.
4. Double-click `start_openocd.bat`.

A black window opens. It works when the last line reads:

```
Info : Listening on port 3333 for gdb connections
```

Leave this window open for the whole debug session. Closing it disconnects the debugger.

Sic2 10: Crcaic ihc dcbug confiiguraiion in CubcIDE

This tells CubeIDE to connect to OpenOCD instead of using its own debugger. You create it once.

1. In CubeIDE, go to **Run →Debug Configurations…**
2. In the list on the left, right-click **GDB Hardware Debugging** and choose **New Configuration**.
3. At the top, change **Name** to `Blue Pill OpenOCD`.
4. **Main tab:**
   - **Project:** click **Browse** and choose `bluepill_test`.
   - **C/C++ Application:** click **Search Project** and choose `bluepill_test.elf`.
5. **Debugger tab:**
   - **GDB Command:** delete the text in the box and paste your **GDB path** from Step 8.
   - Tick **Use remote target**.
   - If there is a **JTAG Device** list, choose **Generic TCP/IP**.
   - **Host name or IP address:** `localhost`
   - **Port number:** `3333`
6. **Startup tab:**
   - Untick **Reset and Delay (seconds)**.
   - Untick **Halt**.
   - In the **Initialization Commands** box, type: `monitor reset halt`
   - Tick **Load image** and select **Use project binary**.
   - Tick **Load symbols** and select **Use project binary**.

- Tick **Set breakpoint at:** and type `main` in the box.
- Tick **Resume**.

7. Click **Apply**, then **Close**.

Do not use CubeIDE's default **STM32 C/C++ Application** debug type on clone boards. It stops with the error "Could not verify ST device".

Sic2 11: Your fiirsi dcbug scssion

1. Check that the OpenOCD window from Step 9 is open and shows `Listening on port 3333`.
2. In CubeIDE, click the small **arrow next to the bug icon** in the toolbar and choose **Blue Pill OpenOCD**. (If it is not listed, use **Run → Debug Configurations…**, select it, and click **Debug**.)
3. When CubeIDE asks to switch to the **Debug perspective**, tick **Remember my decision** and click **Switch**.
4. Wait a few seconds while the program is loaded onto the board. The editor opens `main.c` with the first line of `main()` highlighted in green. The program is now paused there.

Now try the debugger:

1. Press **F6** a few times. The green line moves down one line each time.
2. Look at the **Variables** panel on the right. After you pass the line `uint32_t count = 0;`, `count` shows **0**. (Before that it shows a random number; this is normal.)
3. Double-click in the left margin next to the line `count++;`. A blue dot appears: this is a **breakpoint**.
4. Press **F8**. The program runs until the breakpoint and stops. The green LED on the board changes state.
5. Press **F8** a few more times. Each time `count` goes up by 1 and the LED toggles.
6. Double-click the blue dot to remove the breakpoint, then press **F8**. The LED now blinks continuously.
7. Click the **Suspend** button (two vertical bars) to pause the program wherever it is.
8. Click the red **Terminate** button (square) to end the session.

If all of this works, your setup is complete.

Dcbugging conirols

Control	What it does
F6	Step over: run one line
F5	Step into a function
F7	Step out of the current function
F8	Resume: run until the next breakpoint
Suspend (two vertical bars)	Pause the running program
Double-click the left margin	Add or remove a breakpoint on that line
Terminate (red square)	End the debug session
Variables panel	Live values of local variables
Expressions panel	Watch any variable or expression you type in
Window →Show View →Registers	CPU registers
Window →Show View →SFRs	Peripheral registers (GPIO, timers, UART…)

Evcry iimc afiicr ihai

Once Steps 1–10 are done, each debug session is just:

1. Plug in the ST-Link (wired to the board).
2. Double-click `C:\stm32debug\start_openocd.bat` and leave the window open.
3. Edit your code, save, and click the **hammer** to rebuild.
4. Click the arrow next to the **bug icon** and choose **Blue Pill OpenOCD**.

Always rebuild before debugging. The debugger loads the last built `.elf`, so unbuilt changes do not reach the board.

For a different project, open **Run →Debug Configurations…**, right-click **Blue Pill OpenOCD**, choose **Duplicate**, and change only the Project and C/C++ Application on the Main tab.

Troublcshooiing

Problem	Fix
"Could not verify ST device"	You used the STM32 C/C++ Application debug type. Use **Blue Pill OpenOCD** (Step 10).
CubeIDE: "could not connect" to localhost:3333	OpenOCD is not running. Start `start_openocd.bat` and check it shows `Listening on port 3333`.
The OpenOCD window closes immediately or shows "The system cannot find the path"	A path in `start_openocd.bat` is wrong. Recheck Step 8 and keep the quotes.
OpenOCD: "unable to select a session transport" or "couldn't open bluepill.cfg"	The config file is wrong or saved as `.txt`. Redo Step 7 with Save as type: All Files.
OpenOCD cannot find the ST-Link or the chip	Check the 4 wires, try another USB port, and close any other program using the ST-Link (another CubeIDE debug session, STM32CubeProgrammer).
Board stopped connecting after flashing other code	Hold the board's RESET button, start OpenOCD, and release RESET after about 1 second.
ST-Link Upgrade: "not in the DFU mode"	Unplug and replug the ST-Link, click **Refresh device list**, then **Open in update mode**.
Build errors	Check the code was pasted completely and saved, then rebuild.
Code changes do not show on the board	Rebuild with the hammer before starting the debug session.