ESP32 + J-link debugger does not start

PIO Plus
PIO Unified Debugger

Yourigh 2019-05-08 13:39:25 UTC #1

Hello,
I am aware there have been posts on this topic and went through them. Unfortunately I did not find the fix for my issue.
I was trying many thing on different computer (including some of custom definitions of debug_tool) on different computer unsuccessfully.

Could you please help me to resolve this? I would love to give a unified debugger a chance (just bought a month plan).

On this computer I did these steps:

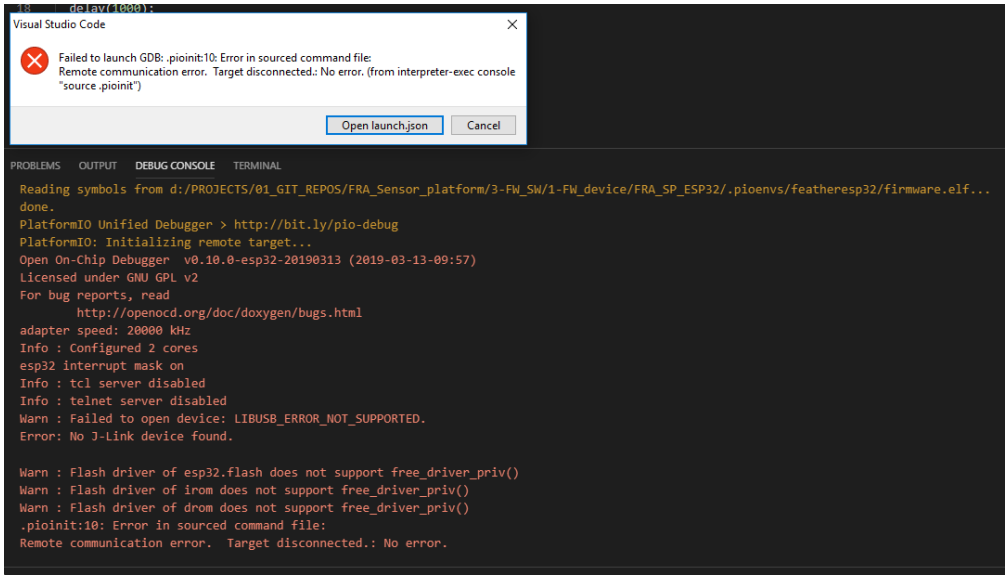
1. J-link driver intstalled
2. latest SEGGER J-link software installed (V6.44h)
3. ESP32 Feather connected to Jlink by this table:

JLINK pin	JLINK name	ESP32 pin	ESP32 signal	Feather huzzah32
1	Vcc			3V
4	GND	1	GND	GND
5	TDI	14	IO12	12
7	TMS	13	IO14	14
9	TCK	16	IO13	13
13	TDO	23	IO15	15
15	nRST	3	EN	RST

EDIT: ESP32 EN connected to pin 3(nTRST) of J-link

4. connected USB of the ESP32 feather board to PC
5. connected Jlink to PC
6. Upload simple program to board (pin toogle) - successfully running
7. Start debugging

8. **this error occurred:**



I have the latest PlatformIO on Visual Studio Code Home 2.0.2·Core 3.6.7

The project configuration plaformio.ini:

```
[env:featheresp32]
platform = espressif32
board = featheresp32
framework = arduino
debug_tool = jlink
```

launch.json:

```
{
"version": "0.2.0",
"configurations": [
    {
        "type": "platformio-debug",
        "request": "launch",
        "name": "PIO Debug",
        "executable": "d:/PROJECTS/01_GIT_REPOS/FRA_Sensor_platform/3-FW_SW/1-FW_device
        "toolchainBinDir": "D:/Cadence/SPB_Data/.platformio/packages/toolchain-xtensa32
        "preLaunchTask": {
            "type": "PlatformIO",
            "task": "Pre-Debug"
        },
        "internalConsoleOptions": "openOnSessionStart"
    },
    {
        "type": "platformio-debug",
        "request": "launch",
```

```
        "name": "PIO Debug (skip Pre-Debug)",
        "executable": "d:/PROJECTS/01_GIT_REPOS/FRA_Sensor_platform/3-FW_SW/1-FW_device
        "toolchainBinDir": "D:/Cadence/SPB_Data/.platformio/packages/toolchain-xtensa32
        "internalConsoleOptions": "openOnSessionStart"
    }
```

maxgerhardt 2019-05-08 15:49:22 UTC #2

Can you also not program the device via the J-Link? (use `upload_protocol = jlink` in the `platformio.ini`)

Yourigh 2019-05-08 18:33:05 UTC #3

Thank you for a reply. Yes, uploading also fails through j-link, I can send a log shortly.

I verified the connections in a great detail couple times.

Yourigh 2019-05-08 18:51:59 UTC #4

When uploading via j-link:

> esptool.py v2.6
> Configuring upload protocol…
> AVAILABLE: esp-prog, esptool, iot-bus-jtag, jlink, minimodule, olimex-arm-usb-ocd, olimex-arm-usb-ocd-h, olimex-arm-usb-tiny-h, olimex-jtag-tiny, tumpa
> CURRENT: upload_protocol = jlink
> Uploading .pioenvs\featheresp32\firmware.bin
> Open On-Chip Debugger v0.10.0-esp32-20190313 (2019-03-13-09:57)
> Licensed under GNU GPL v2
> For bug reports, read
> **http://openocd.org/doc/doxygen/bugs.html**
> adapter speed: 20000 kHz
> Info : Configured 2 cores
> esp32 interrupt mask on
> Warn : Failed to open device: LIBUSB_ERROR_NOT_SUPPORTED.
> Error: No J-Link device found.
> ** OpenOCD init failed **
> shutdown command invoked
> Warn : Flash driver of esp32.flash does not support free_driver_priv()
> Warn : Flash driver of irom does not support free_driver_priv()
> Warn : Flash driver of drom does not support free_driver_priv()
> *** [upload] Error 1`

maxgerhardt 2019-05-08 18:53:46 UTC #5

Hm did you try solutions like **https://github.com/sandeepmistry/arduino-nRF5/issues/228#issuecomment-358783576**? Can you download **https://zadig.akeo.ie/** and replace the J-Link driver with libusb?

Yourigh 2019-05-08 19:20:27 UTC #6

JLINK detected! by changing the driver to WinUSB via Zadig, moving forward to this message:

```
Uploading .pioenvs\featheresp32\firmware.bin
Open On-Chip Debugger  v0.10.0-esp32-20190313 (2019-03-13-09:57)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
adapter speed: 20000 kHz
Info : Configured 2 cores
esp32 interrupt mask on
Info : J-Link V10 compiled Mar 21 2019 15:43:57
Info : Hardware version: 10.10
Info : VTarget = 3.315 V
Info : Reduced speed from 20000 kHz to 15000 kHz (maximum).
Info : Reduced speed from 20000 kHz to 15000 kHz (maximum).
Info : clock speed 20000 kHz
Info : JTAG tap: esp32.cpu0 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : JTAG tap: esp32.cpu1 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : esp32: Debug controller 0 was reset (pwrstat=0x5F, after clear 0x0F).
Info : esp32: Core 0 was reset (pwrstat=0x5F, after clear 0x0F).
Info : esp32: Debug controller 1 was reset (pwrstat=0x5F, after clear 0x0F).
Info : esp32: Core 1 was reset (pwrstat=0x5F, after clear 0x0F).
Info : Listening on port 3333 for gdb connections
Info : JTAG tap: esp32.cpu0 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : JTAG tap: esp32.cpu1 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Error: cpu0: esp32_fetch_all_regs (line 268): DSR (00000088) indicates DIR instruction generated an overrun!
Error: cpu1: esp32_fetch_all_regs (line 268): DSR (00000088) indicates DIR instruction generated an overrun!
Info : Target halted. PRO_CPU: PC=0x400EB38E (active)    APP_CPU: PC=0x00000000
Error: cpu0: xtensa_write_memory (line 802): DSR (8000CC13) indicates DIR instruction generated an exception!
Warn : esp32: Failed writing 4 bytes at address 0x3ff5f064, data - a1, 3a, d8, 50, b8, 8d, 3a, 02
Error: Failed to write ESP32_TIMG0WDT_PROTECT (-4)!
Info : Detected debug stubs @ 3ffc0fb8 on core0 of target 'esp32'
Error: cpu0: xtensa_read_memory (line 695): DSR (00000088) indicates DIR instruction generated an overrun!
Warn : cpu0: Failed reading 76 bytes at address 0x50000000
Error: esp32_soc_reset 528 err=-4
in procedure 'program_esp32'

embedded:startup.tcl:480: Error: ** Unable to reset target **
in procedure 'program_esp32'
in procedure 'program_error' called at file "D:/Cadence/SPB_Data/.platformio/packages/tool-openocd-esp32/share/openocd/scripts/target/esp32.cfg", line 119
at file "embedded:startup.tcl", line 480
Info : Restore debug stubs @ 3ffc0fb8 on core0 of target 'esp32'
Warn : Flash driver of esp32.flash does not support free_driver_priv()
Warn : Flash driver of irom does not support free_driver_priv()
Warn : Flash driver of drom does not support free_driver_priv()
*** [upload] Error 1
```

Do you have any ideas about it? (i tried to connect CHIP_PU (RST) to pin 3 or 15 on JTAG, but it stays the same)

---

maxgerhardt 2019-05-08 19:23:43 UTC #7

First result is **https://github.com/espressif/esp-idf/blob/master/docs/en/api-guides/jtag-debugging/tips-and-quirks.rst#do-not-use-jtag-pins-for-something-else**… do you use GPIO15 in your sketch? Is it connected to something else?

Yourigh 2019-05-08 19:31:00 UTC #8

Nope, GPIO15 is unused. Same as others for JTAG. I do not use on-board LED either. Could the connected LED interfere?

Interestingly, the platformio does not detect Jlink when used for debug but does when used for upload. On debug I get the same as before.

---

maxgerhardt 2019-05-08 19:39:50 UTC #9

The onboard LED is on GPIO13 which is MTCK. We can give it a try if it changes something. Can you change the pin to any other but not 12,13,14 or 15?

Yourigh 2019-05-08 19:52:55 UTC #10

I was not using GPIO13 in code. But the was LED was physically connected to a GPIO13.
I de-soldered it and it did not make any difference.

---

maxgerhardt 2019-05-08 20:00:20 UTC #11

Since I don't have that hardware it's hard for me to reproduce. I'd suggest opening an issue according to **https://github.com/espressif/esp-idf/blob/master/docs/en/api-guides/jtag-debugging/tips-and-quirks.rst#reporting-issues-with-openocd--gdb** or maybe **@ivankravets** can help.

Yourigh 2019-05-08 20:05:27 UTC #12

successful upload after dropping the frequency in share/openocd/scripts/board/esp-wroom-32.cfg

```
# On DevKit-J, this can go as high as 20MHz if CPU frequency is 80MHz, or 26MHz>     #
#adapter_khz 20000
adapter_khz 5000
```

Debugger also running.
Thank you a lot!

---

maxgerhardt 2019-05-08 20:11:04 UTC #13

Oh great, I actually read that and forgot about it >_>

The maximum possible adapter speed can also be influenced by wire lengths (or bad wire quality), so maybe try another shorter, better cable with the higher speed of 20MHz. Or stay on the working 5MHz for now 🙂

Yourigh 2019-05-08 20:14:07 UTC #14

10MHz works fine, I will increase it when on the manufactured board with better signal integrity.

Release Notes

Continuous Integration

Cloud & Standalone IDEs

Community

- **Forums**
- **GitHub**
- **Bintray**
- **Blog**
- **Twitter**
- **Facebook**
- **Hackaday**