

mbed Tutorial 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

Nội dung trình bày:

- Cài đặt
- Hướng dẫn nạp/gỡ lỗi cho VBLUno trên GNU ARM Eclipse.

1. Cài đặt

a. Cài đặt pyOCD

+ pyOCD là một chương trình mã nguồn mở cho phép nạp chương trình và gỡ lỗi cho các vi xử lý ARM Cortex-M sử dụng CMSIS-DAP (DAPLink) interfaces.

+ pyOCD được xây dựng bằng ngôn ngữ lập trình Python, và dĩ nhiên nó hỗ trợ trên nền tảng: Windows, Linux và OSX.

+ Tham khảo cách cài đặt tại: <https://github.com/mbedmicro/pyOCD>

+ Sau khi cài thành công, bạn thử gõ "pyocd-flashtool" tại Terminal, nếu bạn thấy hành động như dưới đây tức là đã cài đặt thành công, bạn cần bổ sung đường dẫn của pyOCD vào biến môi trường PATH.

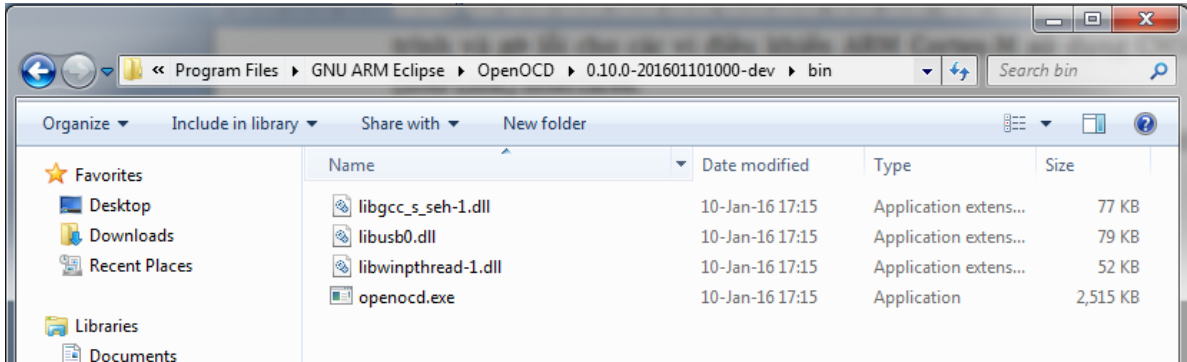
```
C:\Users\MrNam>pyocd-flashtool
No connected boards
Error: There is no board connected.
```

b. Cài đặt OpenOCD

+ Trong khi pyOCD của mbed team thì tập trung riêng cho CMSIS-DAP (DAPLink) interface thì OpenOCD là một phần mềm có chức năng tổng quát hơn, nó có thể hỗ trợ việc nạp/gỡ lỗi cho CMSIS-DAP và rất nhiều công cụ khác như: Jlink, ST-Link, Ulink, redbee-usb,....

TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

+ Nếu bạn đã cài đặt GNU ARM Eclipse plugin cho Eclipse như hướng dẫn trong Tutorial 2, thì OpenOCD đã được cài đặt sẵn vào máy tính của bạn. Nếu bạn sử dụng Windows OS, có thể OpenOCD sẽ được cài vào máy tính như hình sau

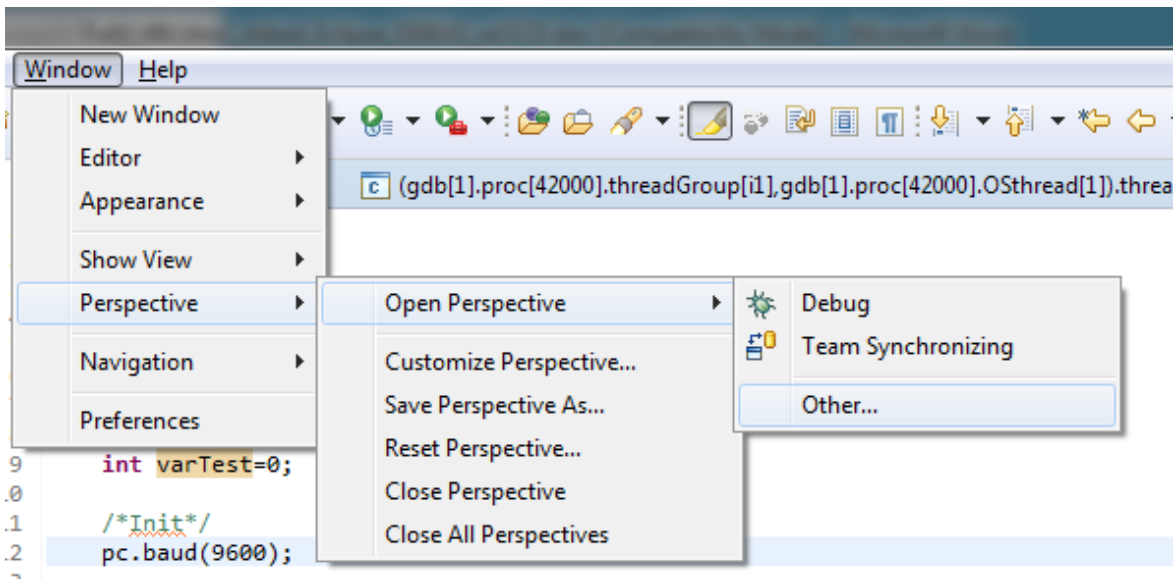


+ Bạn cũng có thể cài đặt OpenOCD theo hướng dẫn sau

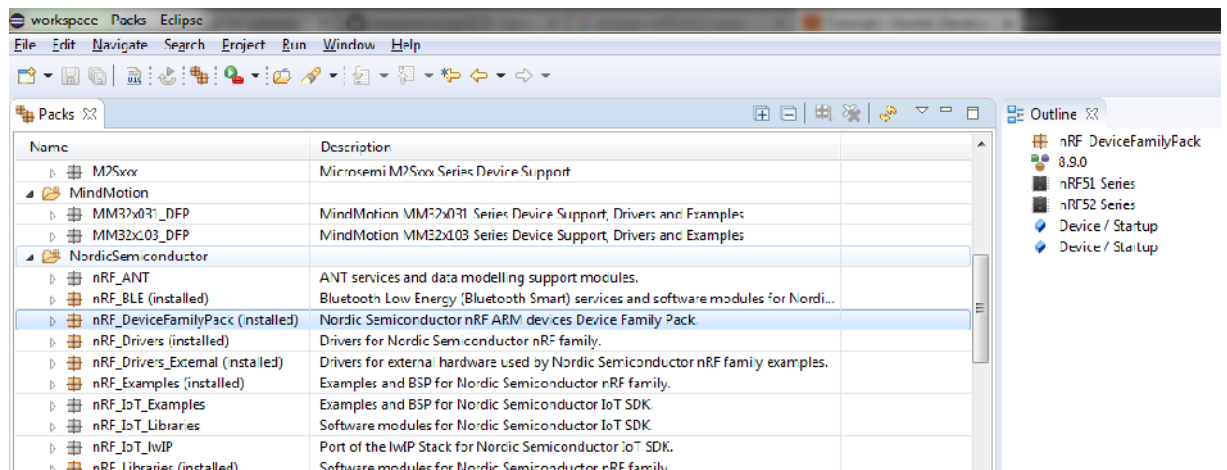
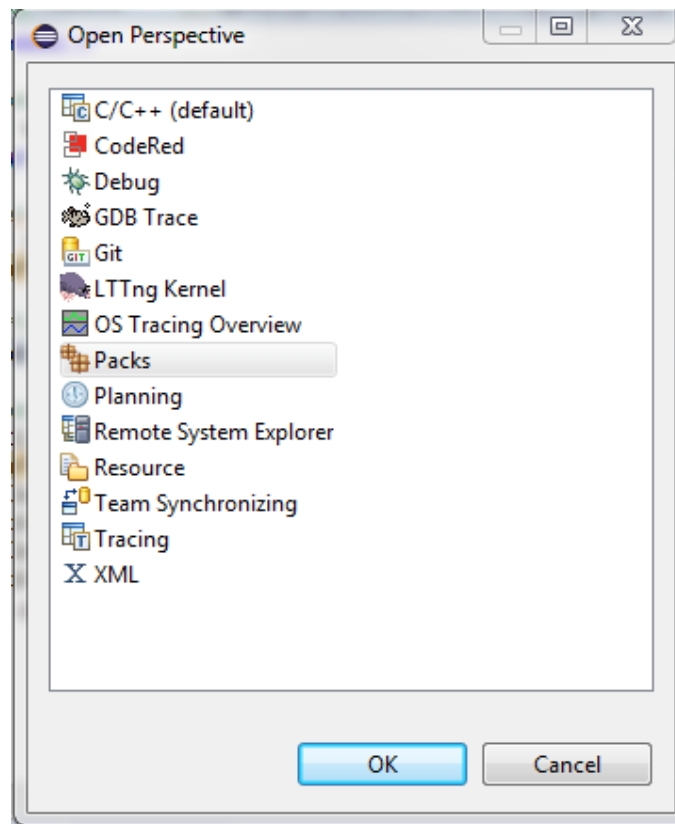
<https://sourceforge.net/projects/openocd/>

c. Cài đặt thêm các Device Packs

Device packs của nhà sản xuất chip giúp GNU ARM Eclipse plugin có thể hiển thị các thông tin về ngôn ngữ lập trình, thanh ghi, biến, ... của chip trong khi chúng ta tiến hành debug. Bạn cài đặt theo các hình sau

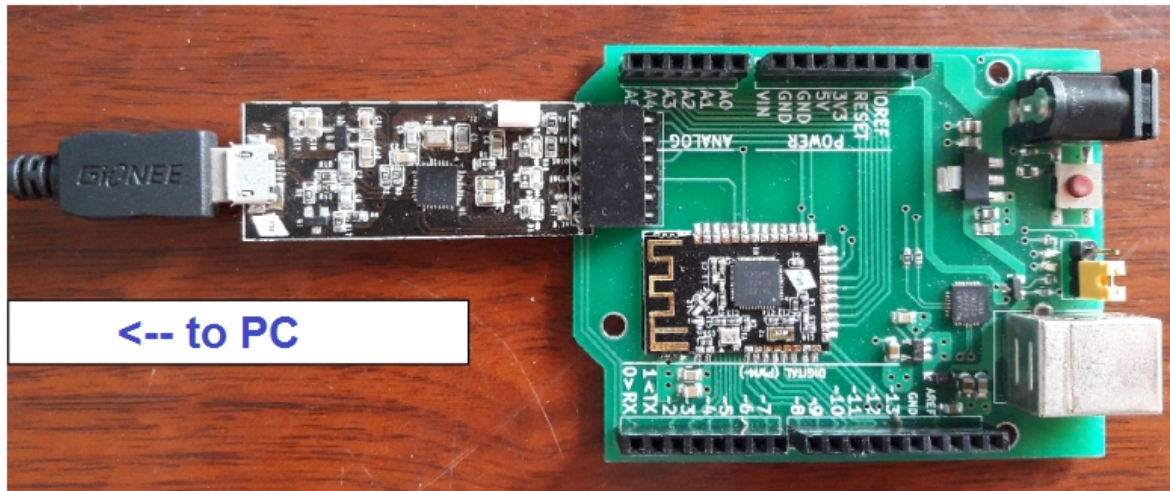


TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse



2. Hướng dẫn nạp/gỡ lỗi cho VBLUno trên GNU ARM Eclipse.

S k t n i:



a. Nạp chương trình

+ S a Makefile:

B sung thêm t dòng 128 n 134

```
127
128 flash_pyocd:
129     @echo -----Flashing by pyOCD: combined.hex-----
130     pyocd-flashtool combined.hex
131
132 flash_openocd:
133     @echo -----Flashing by OpenOCD: combined.hex-----
134     "C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201601101000-dev/bin/openocd.exe"
135 endif
136
```

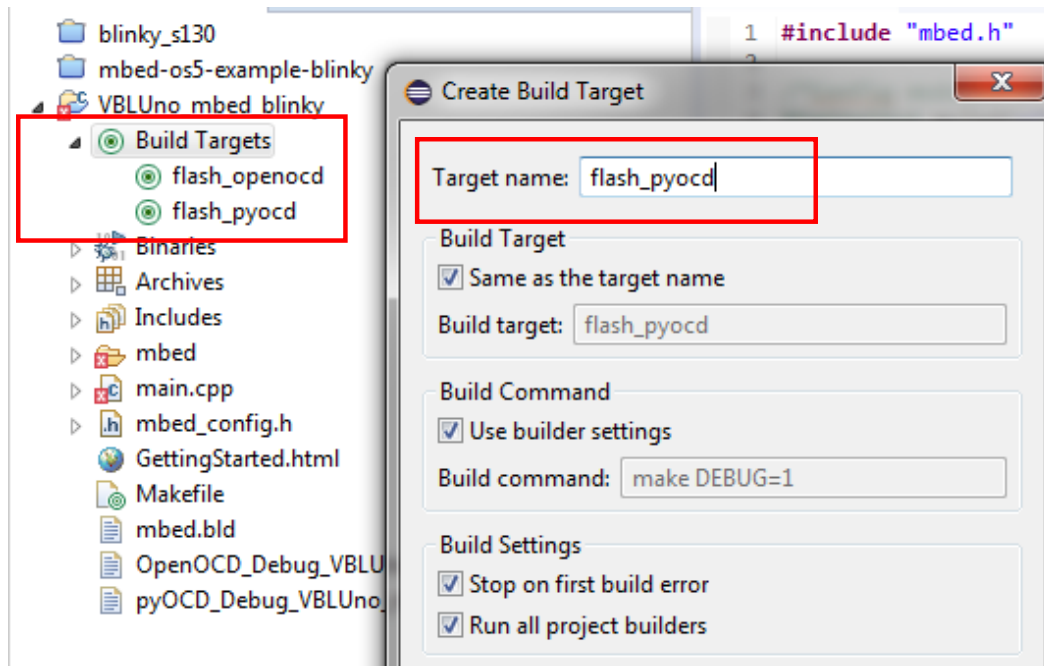
Trong ó dòng 134 có n i dung là:

```
"C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201601101000-dev/bin/openocd.exe" -f "C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201601101000-dev/scripts/interface/cmsis-dap.cfg" -f "C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201601101000-dev/scripts/target/nrf51.cfg" -c "program combined.hex verify reset exit 0x0"
```

+ T o target:

T o 2 target v i tên là *flash_pyocd* và *flash_openocd*:

TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

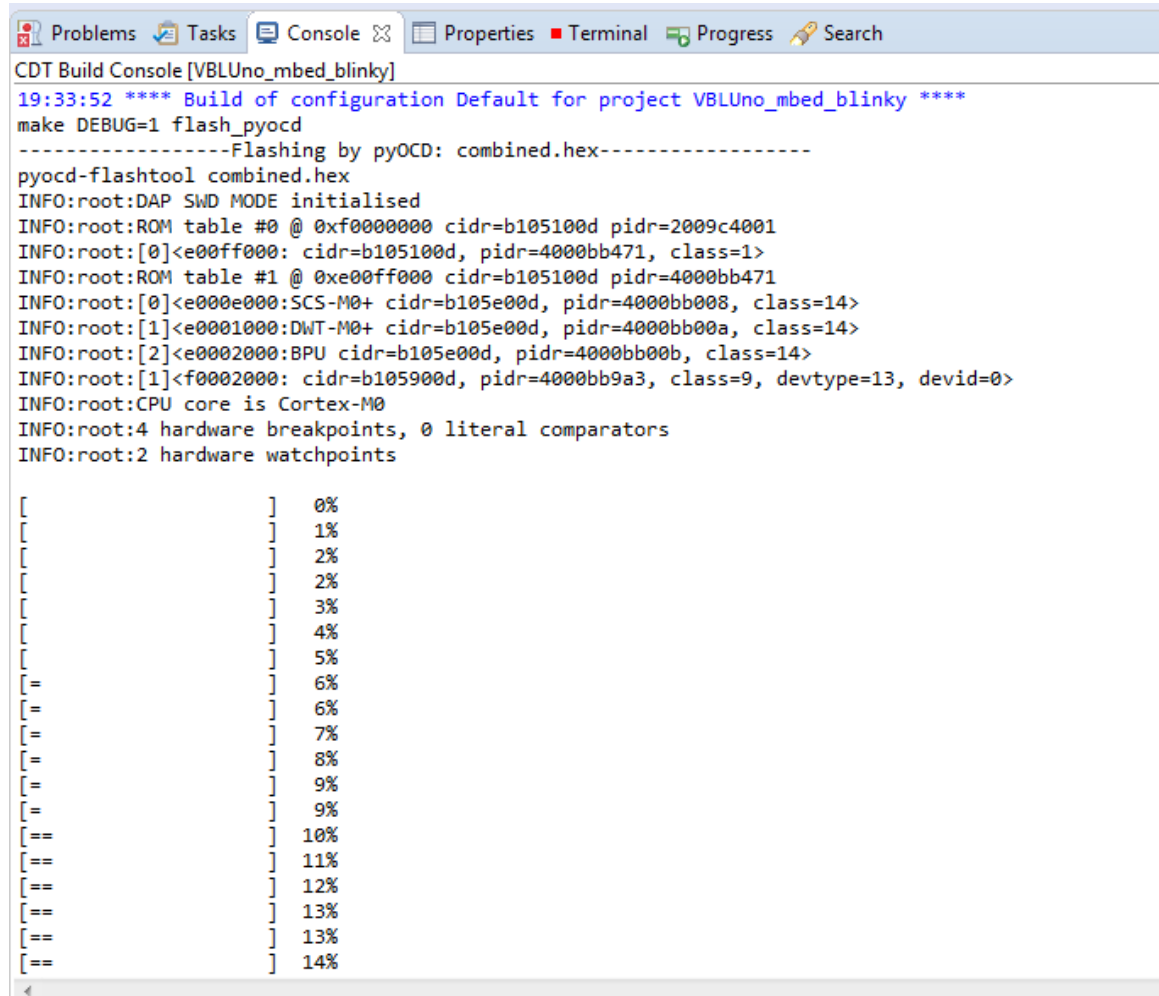


+ Flash chương trình xuống mạch VBLUno

Sau khi build project, nhấp chọn chương trình cho mạch board pyOCD, nhấp vào target flash_pyocd. Tiếp theo cho OpenOCD.

Hình dưới đây xuất hiện khi nhấp board pyOCD

TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

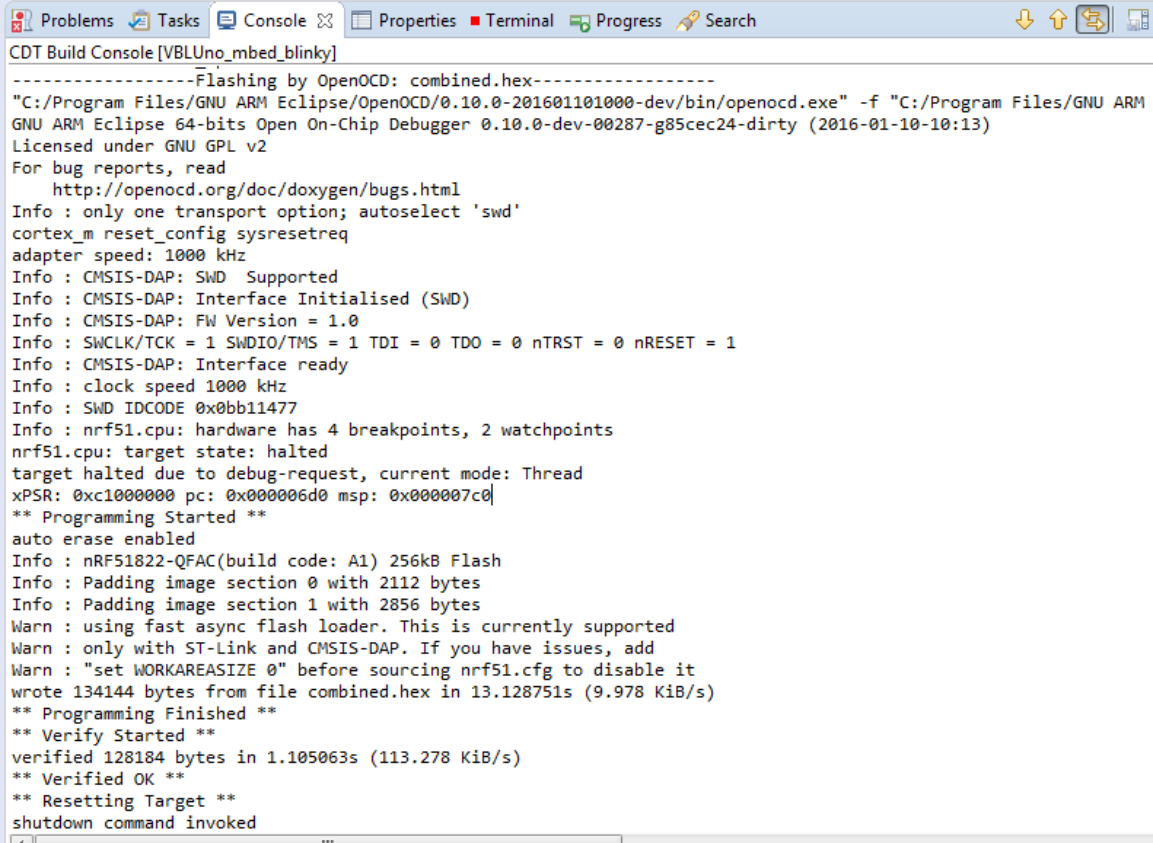


```
Problems Tasks Console Properties Terminal Progress Search
CDT Build Console [VBLUno_mbed_blinky]
19:33:52 **** Build of configuration Default for project VBLUno_mbed_blinky ****
make DEBUG=1 flash_pyocd
-----Flashing by pyOCD: combined.hex-----
pyocd-flashtool combined.hex
INFO:root:DAP SWD MODE initialised
INFO:root:ROM table #0 @ 0xf0000000 cidr=b105100d pidr=2009c4001
INFO:root:[0]<e00ff000: cidr=b105100d, pidr=4000bb471, class=1>
INFO:root:ROM table #1 @ 0xe00ff000 cidr=b105100d pidr=4000bb471
INFO:root:[0]<e000e000:SCS-M0+ cidr=b105e00d, pidr=4000bb008, class=14>
INFO:root:[1]<e0001000:DWT-M0+ cidr=b105e00d, pidr=4000bb00a, class=14>
INFO:root:[2]<e0002000:BPU cidr=b105e00d, pidr=4000bb00b, class=14>
INFO:root:[1]<f0002000: cidr=b105900d, pidr=4000bb9a3, class=9, devtype=13, devid=0>
INFO:root:CPU core is Cortex-M0
INFO:root:4 hardware breakpoints, 0 literal comparators
INFO:root:2 hardware watchpoints

[ ] 0%
[ ] 1%
[ ] 2%
[ ] 2%
[ ] 3%
[ ] 4%
[ ] 5%
[ ] 6%
[ ] 6%
[ ] 7%
[ ] 8%
[ ] 9%
[ ] 9%
[ ] 10%
[ ] 11%
[ ] 12%
[ ] 13%
[ ] 13%
[ ] 14%
```

Hình d i này xu t hi n khi n p b ng OpenOCD

TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse



```
CDT Build Console [VBLUno_mbed_blinky]
-----Flashing by OpenOCD: combined.hex-----
"C:/Program Files/GNU ARM Eclipse/OpenOCD/0.10.0-201601101000-dev/bin/openocd.exe" -f "C:/Program Files/GNU ARM
GNU ARM Eclipse 64-bits Open On-Chip Debugger 0.10.0-dev-00287-g85cec24-dirty (2016-01-10-10:13)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.org/doc/doxygen/bugs.html
Info : only one transport option; autoselect 'swd'
cortex_m reset_config sysresetreq
adapter speed: 1000 kHz
Info : CMSIS-DAP: SWD Supported
Info : CMSIS-DAP: Interface Initialised (SWD)
Info : CMSIS-DAP: FW Version = 1.0
Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 0 nTRST = 0 nRESET = 1
Info : CMSIS-DAP: Interface ready
Info : clock speed 1000 kHz
Info : SWD IDCODE 0x0bb11477
Info : nrf51.cpu: hardware has 4 breakpoints, 2 watchpoints
nrf51.cpu: target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0xc1000000 pc: 0x000006d0 msp: 0x000007c0
** Programming Started **
auto erase enabled
Info : nRF51822-QFAC(build code: A1) 256kB Flash
Info : Padding image section 0 with 2112 bytes
Info : Padding image section 1 with 2856 bytes
Warn : using fast async flash loader. This is currently supported
Warn : only with ST-Link and CMSIS-DAP. If you have issues, add
Warn : "set WORKAREASIZE 0" before sourcing nrf51.cfg to disable it
wrote 134144 bytes from file combined.hex in 13.128751s (9.978 KiB/s)
** Programming Finished **
** Verify Started **
verified 128184 bytes in 1.105063s (113.278 KiB/s)
** Verified OK **
** Resetting Target **
shutdown command invoked
```

Sau khi nạp xong thì nhấn nút reset và chạy chương trình bắt đầu chạy.

b. Debug với pyOCD

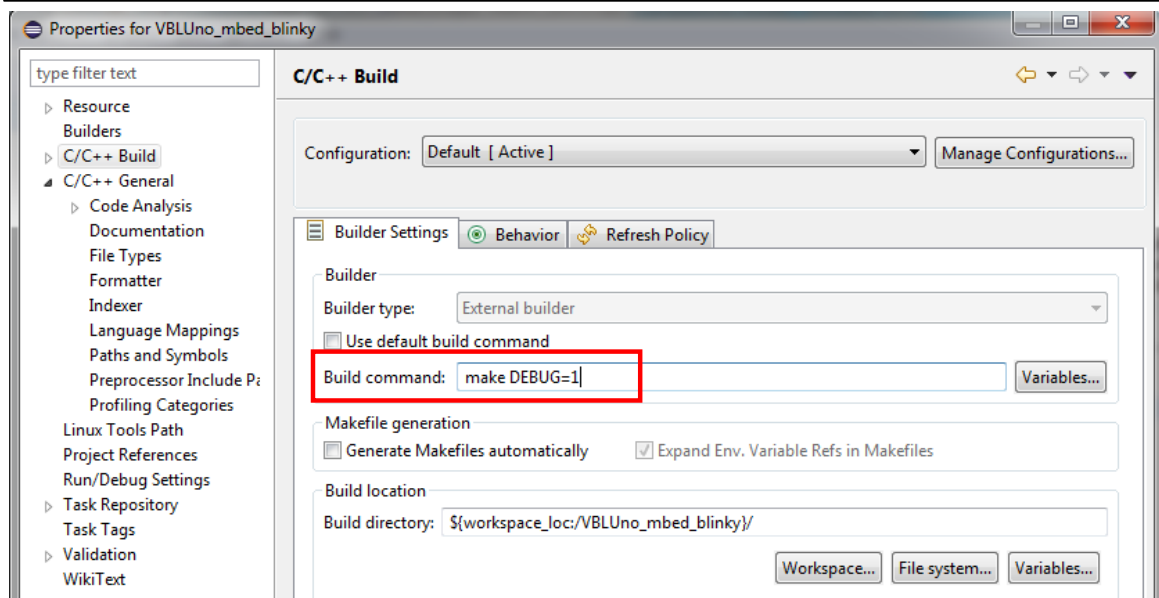
+ Sửa Makefile

Chú ý dòng 68, 69 trong hình sau

```
66
67 ifeq ($(DEBUG), 1)
68   CC_FLAGS += -DDEBUG -O0 -g3
69   CXX_FLAGS += -DDEBUG -O0 -g3
70 else
71   CC_FLAGS += -DNDEBUG -Os
72   CXX_FLAGS += -DNDEBUG -Os
```

+ Sửa Build command

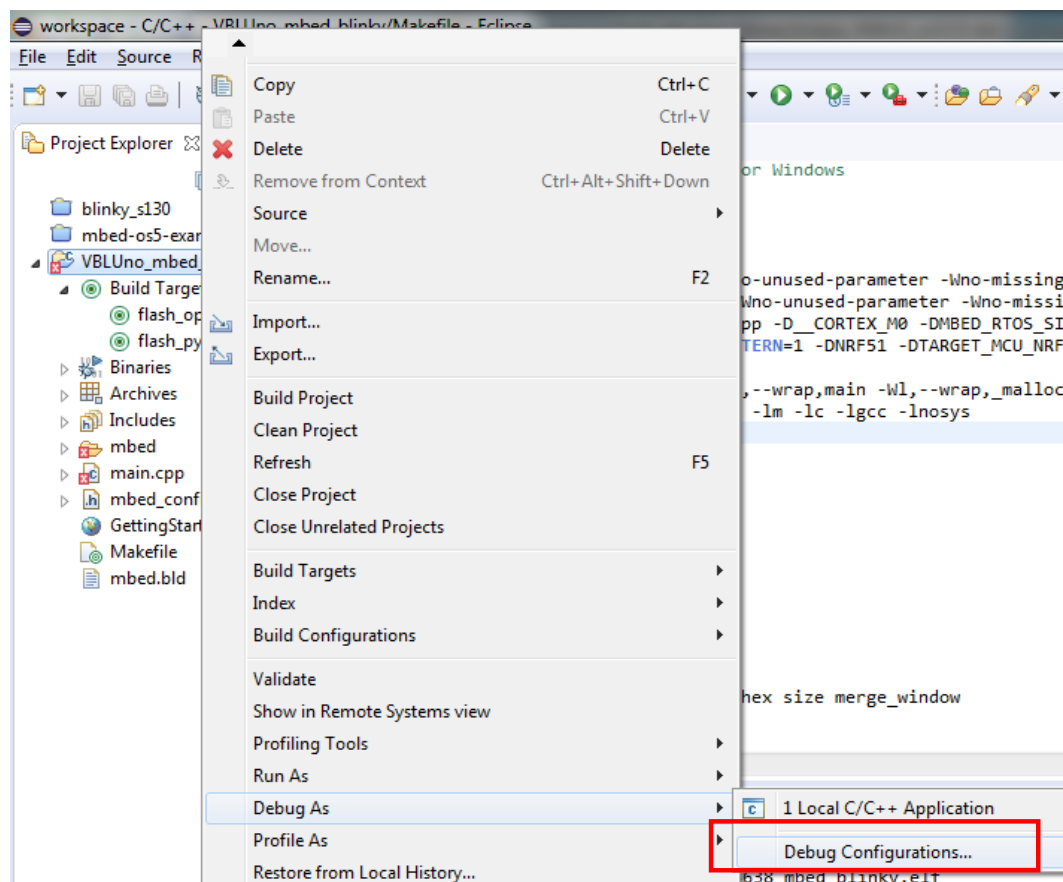
TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse



+ Build lại project

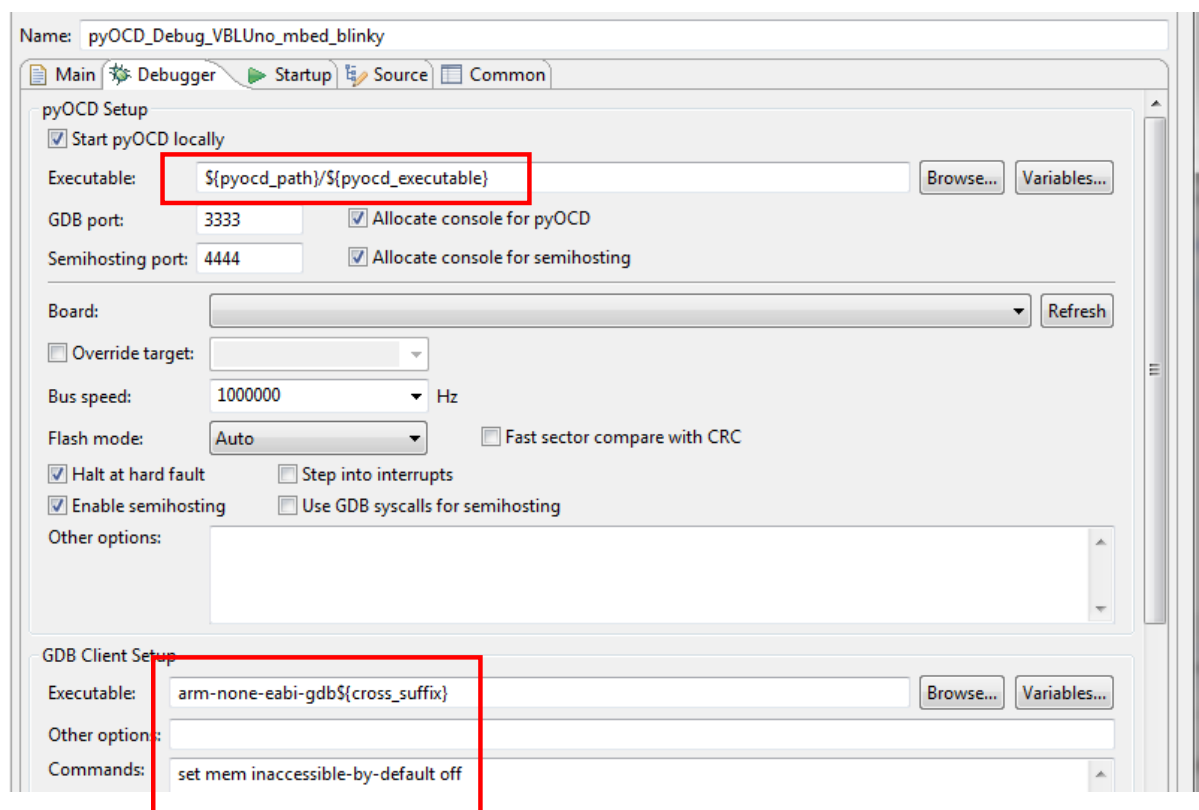
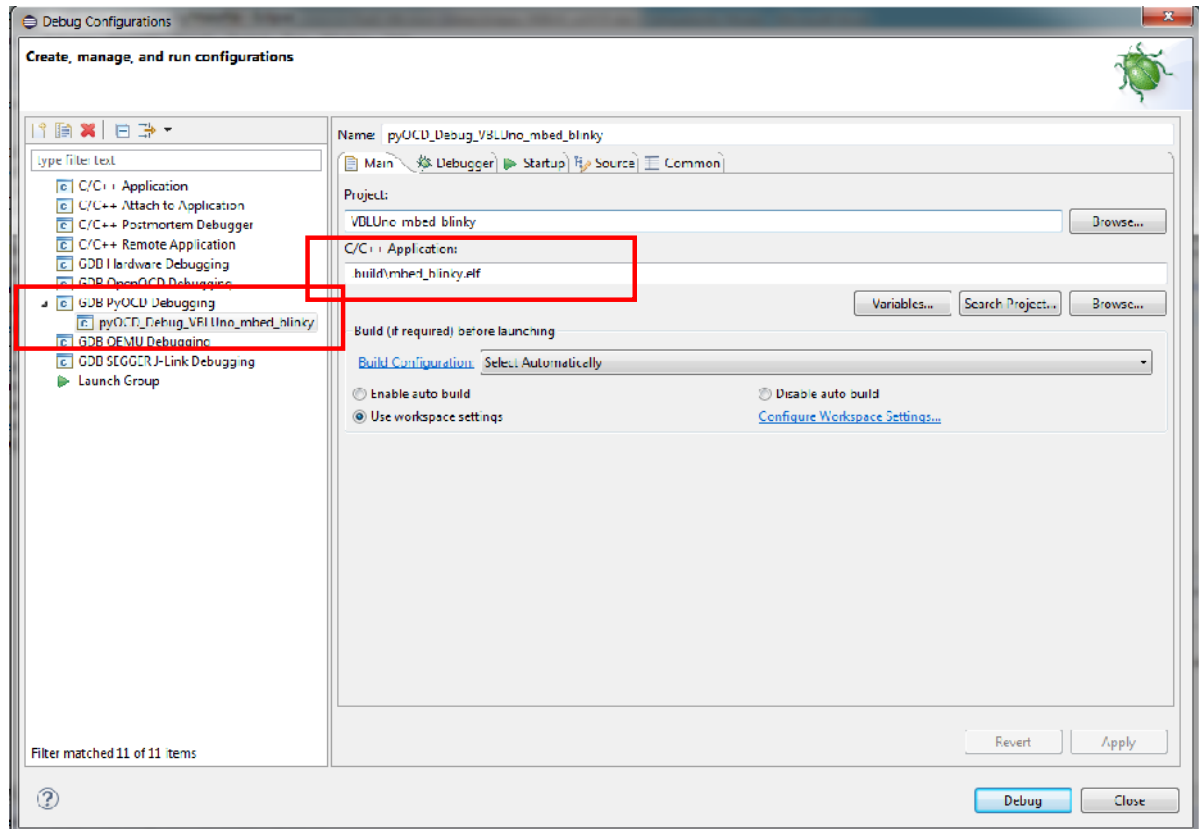
+ Cấu hình debug

Chuột phải vào project -> Chọn Debug Configurations

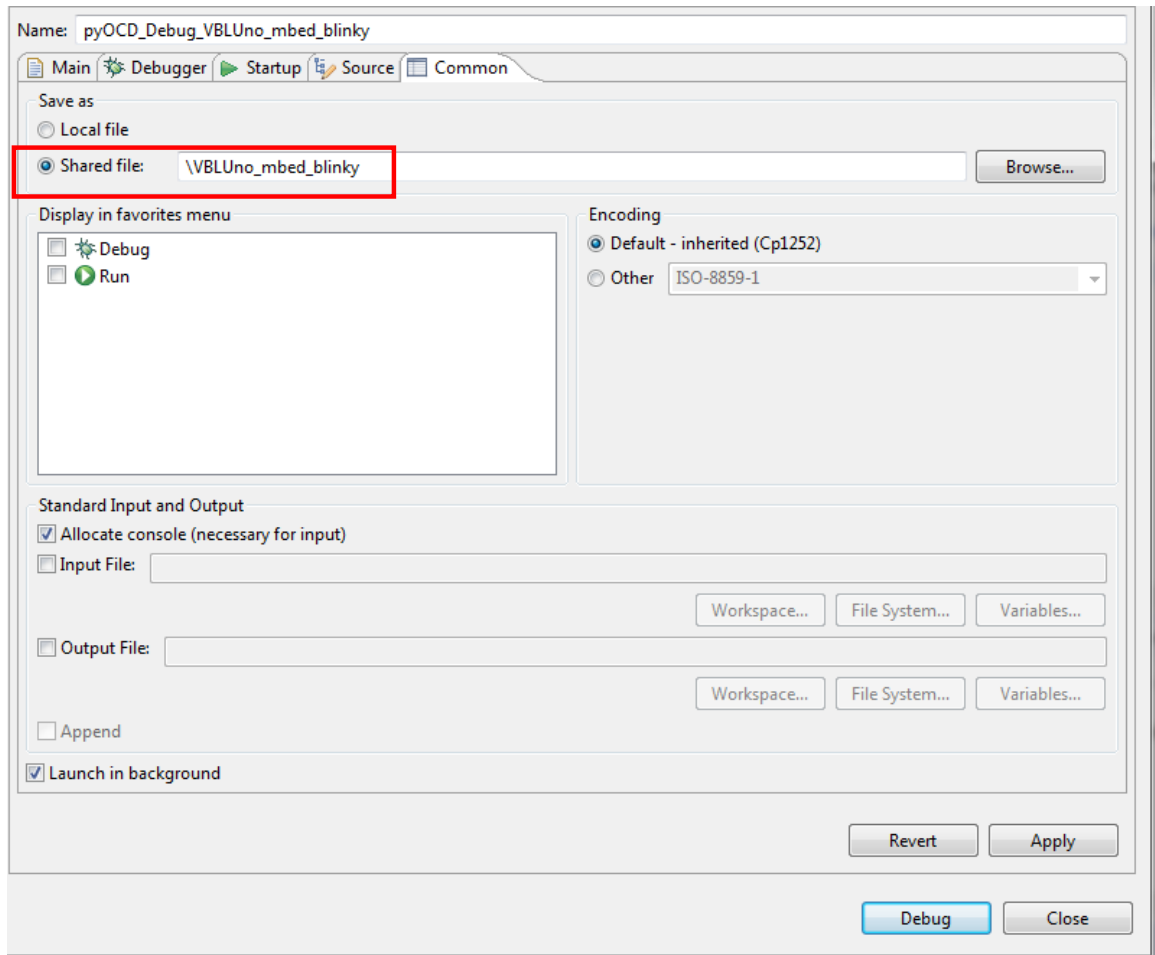


TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

Nhập vào GDB pyOCD Debugging để có hình debug như sau:

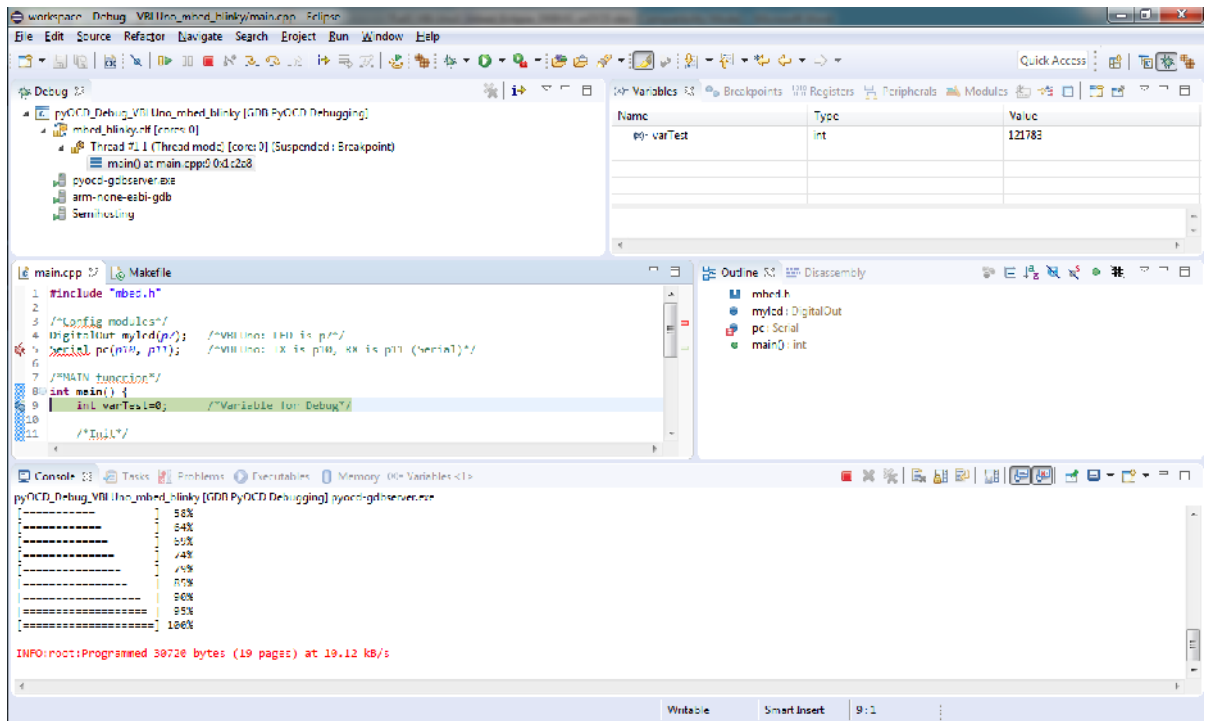


TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse



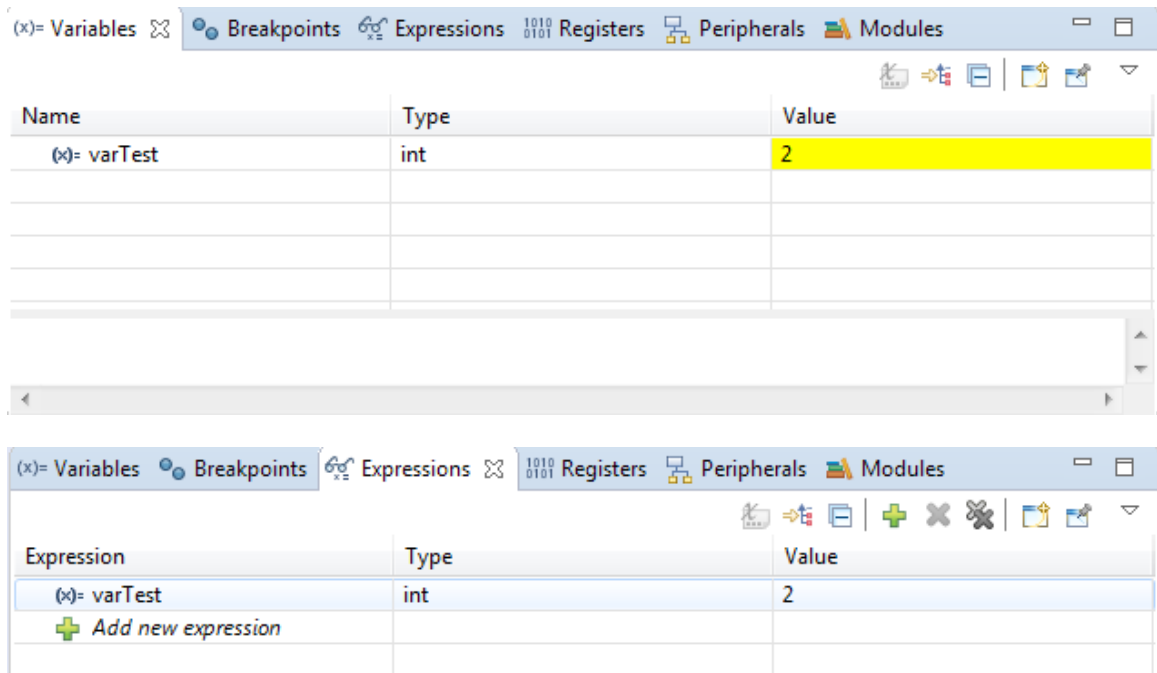
+ Nhấn Debug bắt đầu quá trình debug

TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

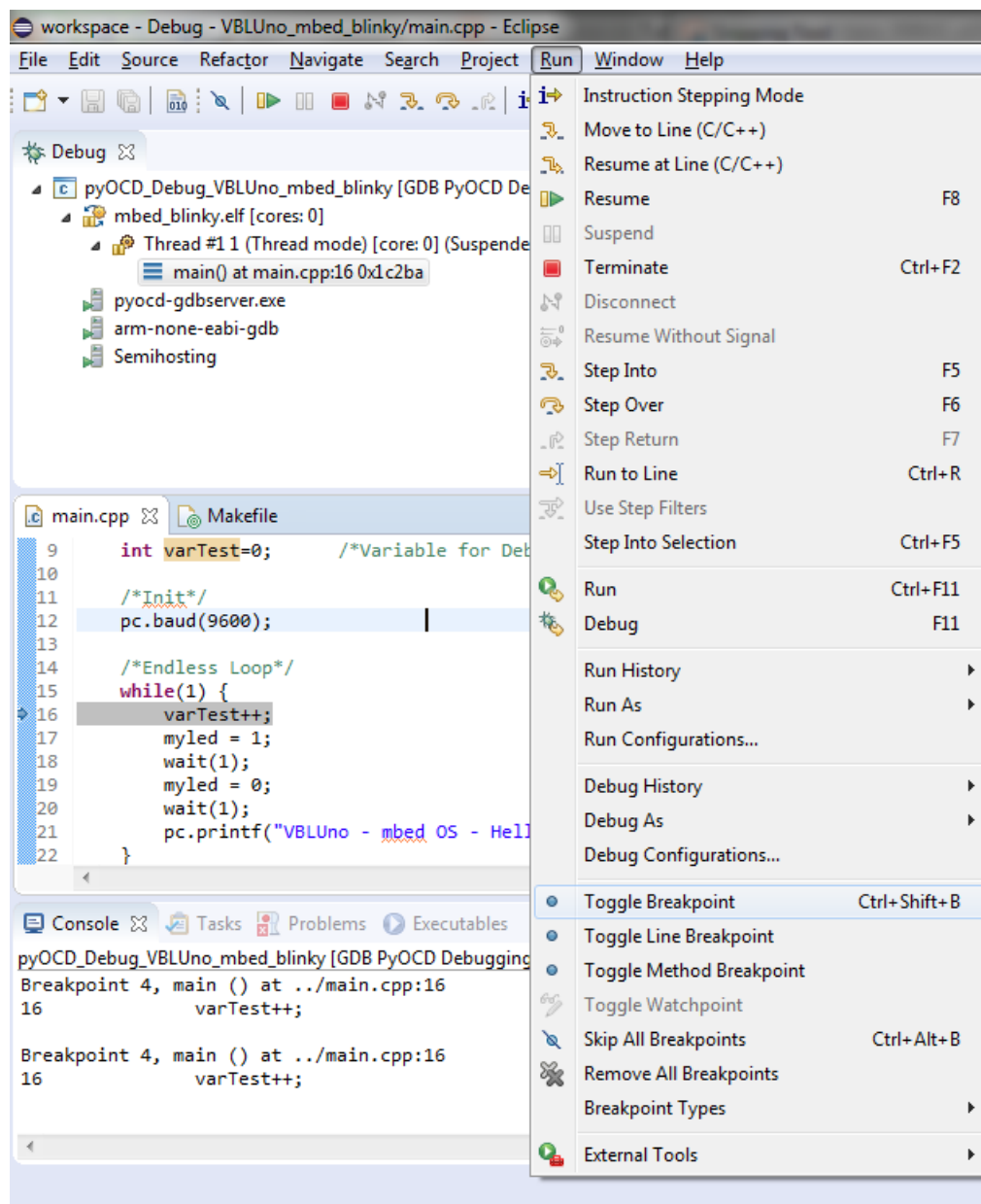
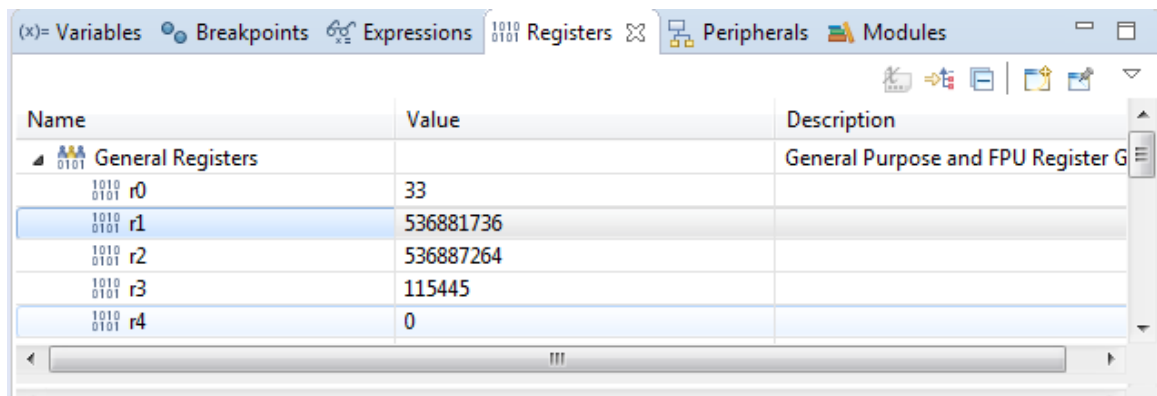


pyOCD hỗ trợ các tính năng debug thông thường như : Run, Pause, Stop, Step Into, Step Over, Run to Line, Breakpoint, Watch biến và biến thể, ... Ngoài ra còn hỗ trợ xem giá trị Registers, Memory.

Một số hình ảnh minh họa:

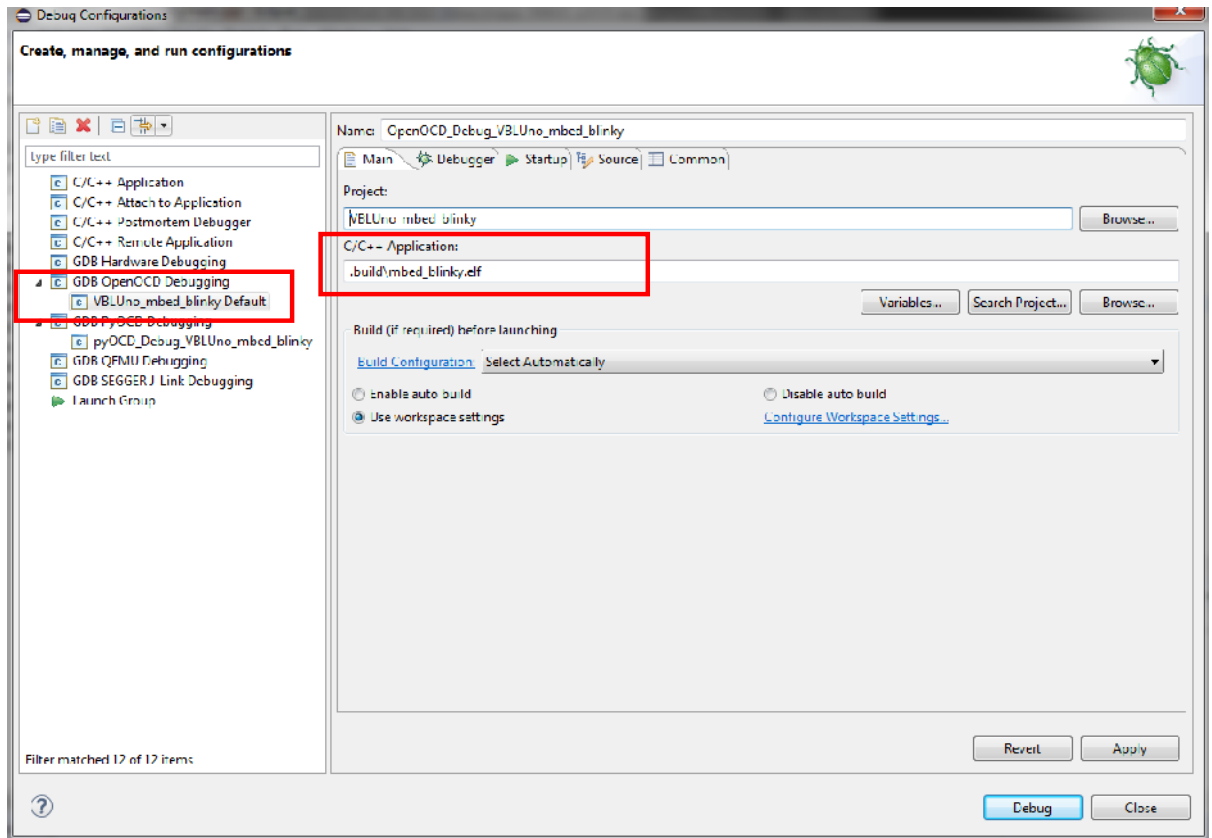


TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

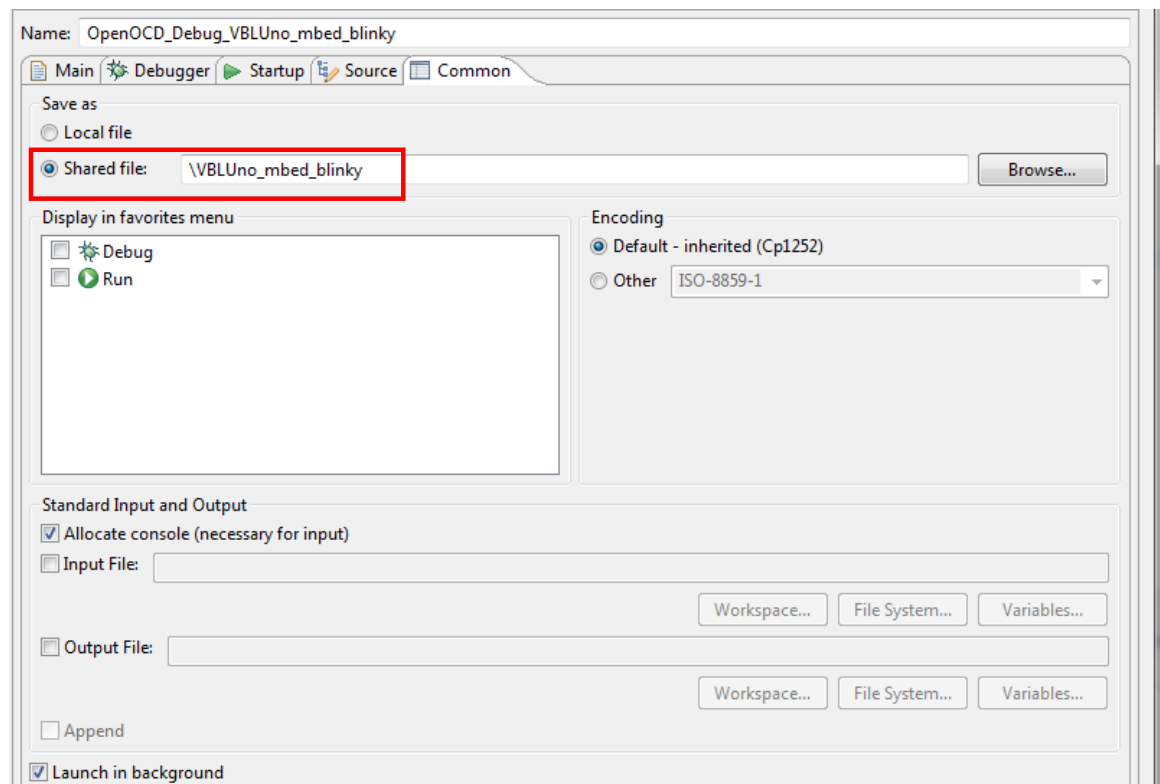
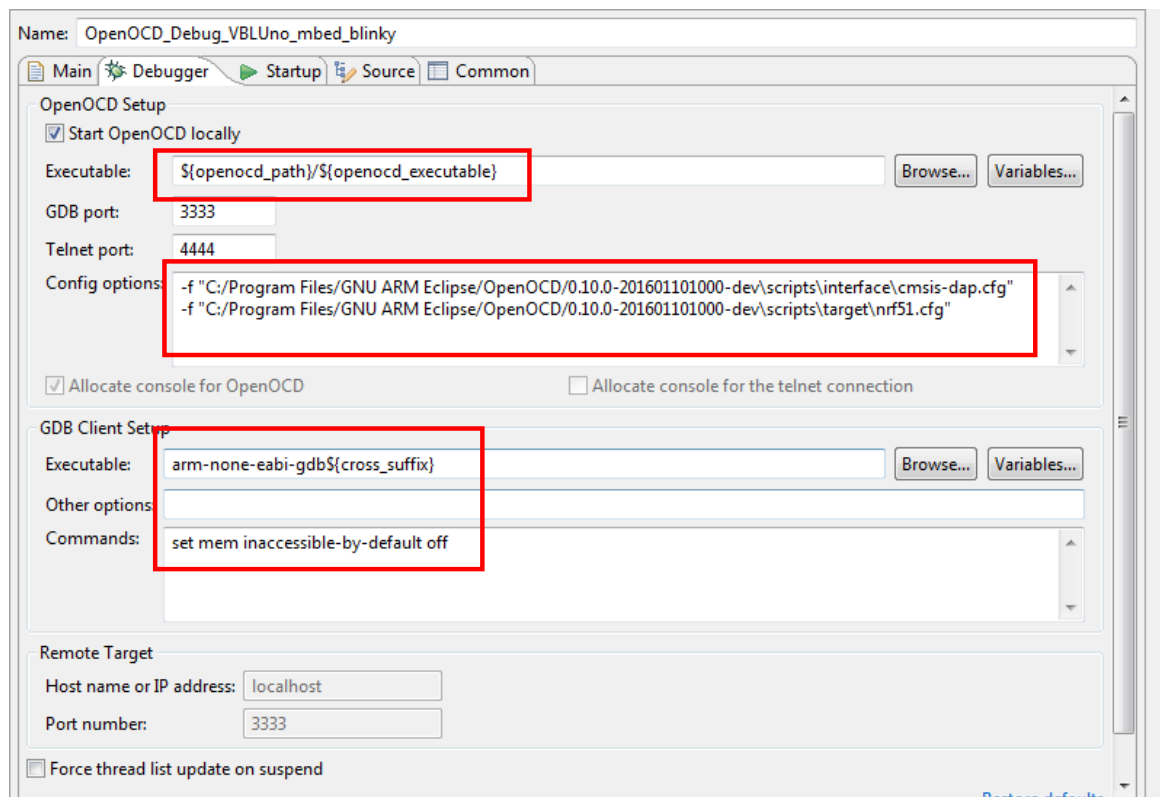


c. Debug với OpenOCD

Thì chỉ cần tạo một dự án mới với pyOCD, chú ý cấu hình theo các hình sau:

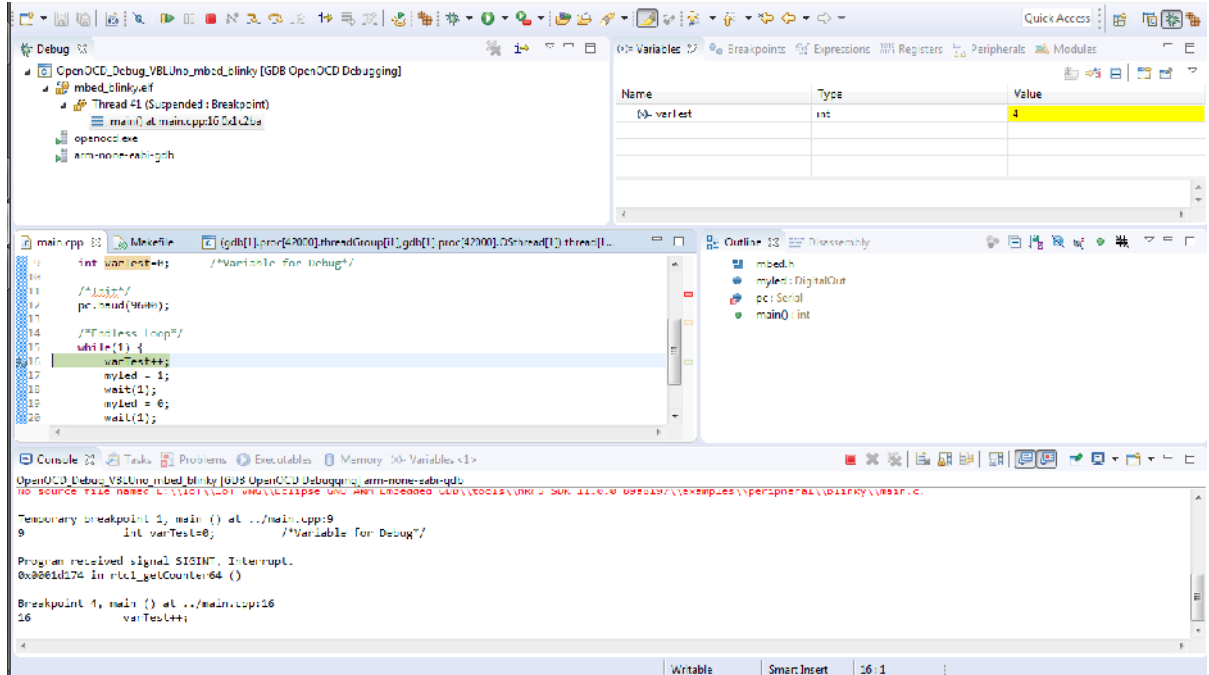


TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse



TUTORIAL 3: Sử dụng công cụ nạp/gỡ lỗi pyOCD và OpenOCD trên môi trường GNU ARM Eclipse

OpenOCD cung cấp các tính năng debug như pyOCD



Có thể thấy, Eclipse + GNU ARM Eclipse plugin + GNU ARM Embedded toolchain + pyOCD, OpenOCD hỗ trợ developers phát triển ứng dụng cho board VBLUno một cách dễ dàng. Một môi trường lập trình hỗ trợ GUI, cho phép nạp và gỡ lỗi chương trình một cách thuận tiện sẽ giúp nâng cao hiệu suất làm việc của các lập trình viên.

Như vậy, trong ba tutorial đầu tiên của mbed Tutorial series cho board VBLUno, chúng tôi đã giới thiệu với các bạn các công cụ, môi trường để phát triển ứng dụng IoT với hệ điều hành nhúng mbed-OS.

Trong những bài hướng dẫn tiếp theo, chúng tôi sẽ hướng dẫn các bạn phát triển các ứng dụng Bluetooth Low Energy với board VBLUno của VNGIoTLab. Với VBLUno các bạn có thể dễ dàng tiếp cận vào làm chủ công nghệ BLE, giúp xây dựng các ứng dụng IoT hiệu quả.

Mọi thắc mắc hoặc ý kiến đóng góp bạn có thể trao đổi tại đây:

<https://www.facebook.com/bleviet/messages/>

https://github.com/VNGIoTLab/mbed_for_VBLUno-CMSIS-DAP/issues