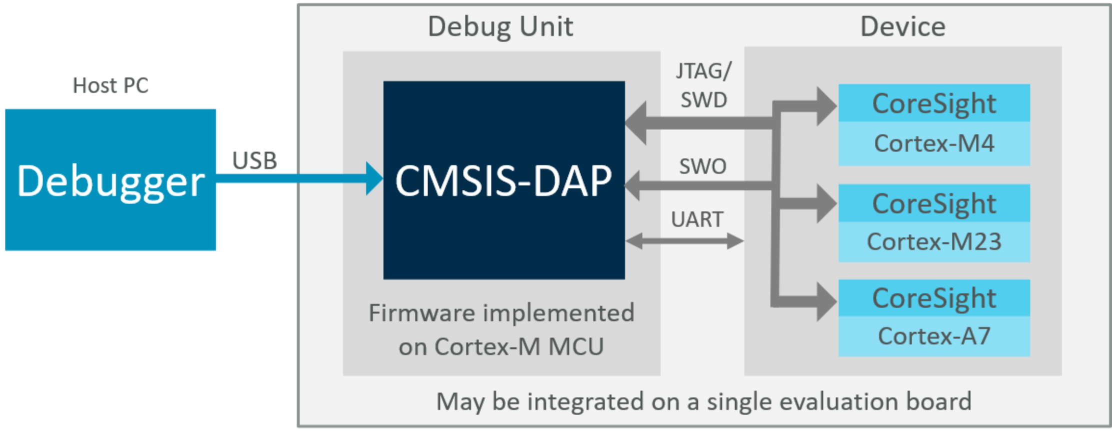
CMSIS-DAP FOR RA

WHAT IS CMSIS-DAP

- ❑ Protocol defined by Arm for a debug probe to communicate with Debug Access Port of a Cortex-M device over SWD
  - Part of Arm’s “Cortex Microcontroller Software Interface Standard”
- ❑ Arm provide source code, under open source license, including some example implementations.
  - DAP-Link (which came out of Arm’s mbed project) is one particular implementation of CMSIS-DAP, terms sometimes get confused
- ❑ Originally aimed mainly at low cost, low performance on-board probes on eval boards
  - Many – but not all - implementations are still in this category
  - But various standalone probes are also available (e.g. NXP’s LPC-Link2 and MCU-Link, RaspberryPI)
- ❑ Arm initially tried not to “upset” probe vendors like SEGGER by keeping performance and features limited, now changing :
  - Higher performance
    - Can now use USB bulk endpoints instead of HID in CMSIS-DAP v2.0.
    - Seeing better MCUs used in probe implementations (clock speeds and USB high speed vs Full speed, packet sizes)
  - More functionality : SWO trace

CMSIS-DAP AND RA



- ❑ Two sides to the “Do we support CMSIS-DAP for RA” question….

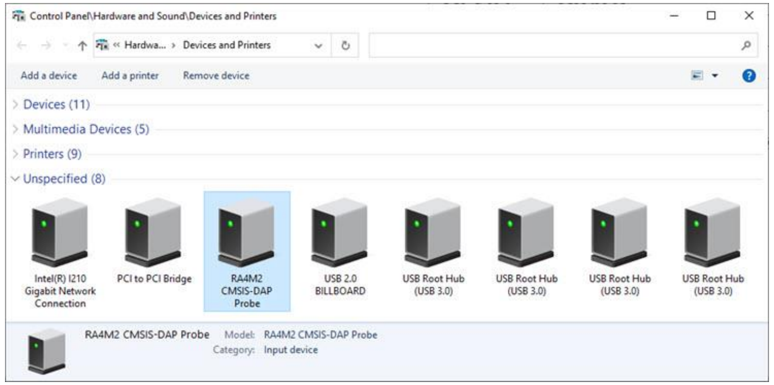
1) Probe firmware running on an RA MCU

2) Use of a CMSIS-DAP probe (potentially, but not necessarily, running on an RA MCU) to debug an RA MCU target from tools running on a host PC

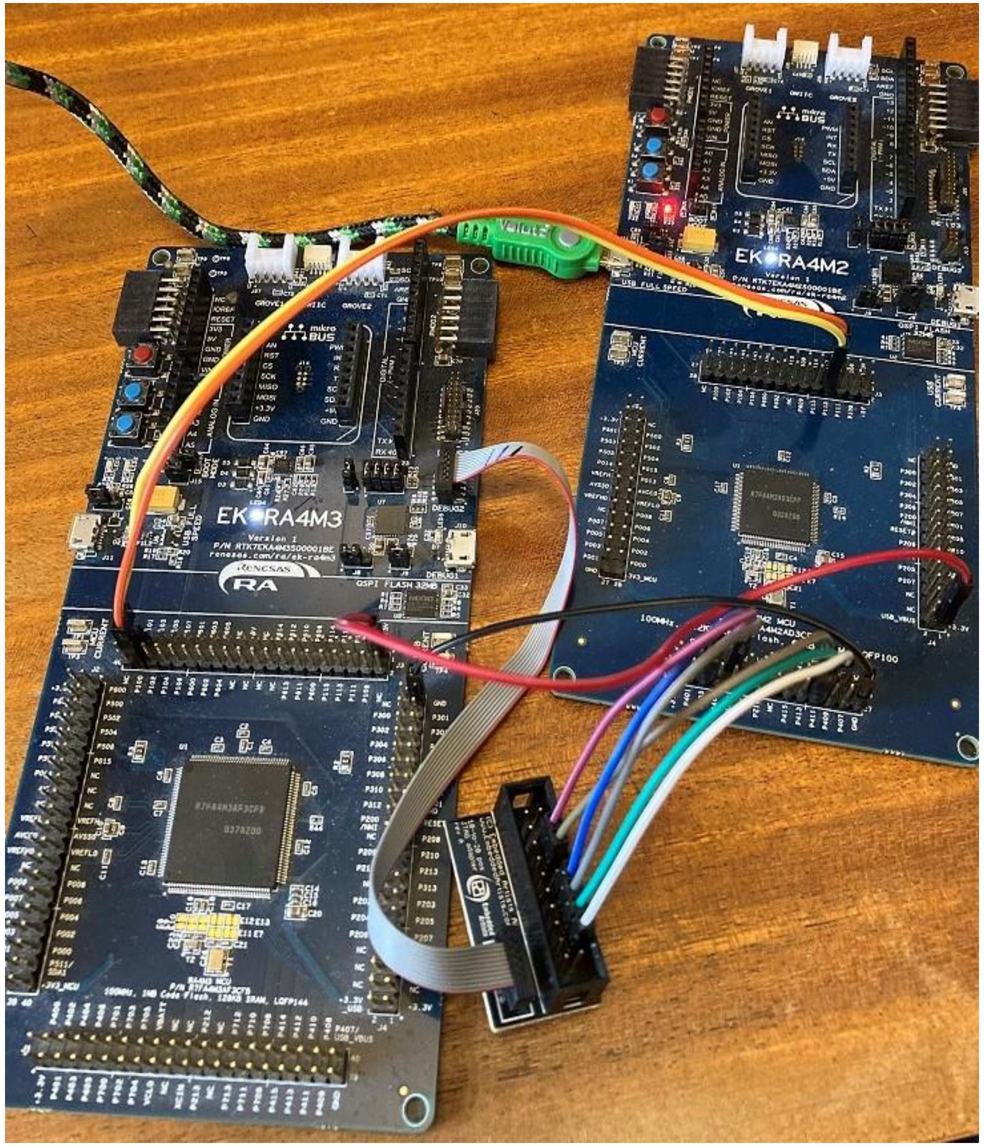
- ❑ Remember that for RA “Cortex-M33 with DLM” MCUs, boot mode access is not possible via CMSIS-DAP
  - Currently have to use RFP (or similar) via a separate USB/SCI connection to board (and remember to manually pull MD pin low to put MCU into boot mode)

PORT OF CMSIS-DAP TO RA MCU

- ❑ Done under contract by a third party (Ashling), not SDD
  - Targets RA4M2 / EK-RA4M2, built using FSP 4.3 / e^2 studio 2023-01
- ❑ Port (source project and prebuilt image) available from:
  - https://github.com/renesas/ra-cmsis-dap-port
  - Provides debug and VCOM interfaces
- ❑ Performance
  - Depends on host tools, but much slower than J-Link
  - Aiming to improve by porting to FSP 4.4 - should allow use of USB bulk endpoints instead of HID, due to FSP USB stack enhancement



RA4M2 CMSIS-DAP Probe
Model: RA4M2 CMSIS-DAP Probe
Category: Input device



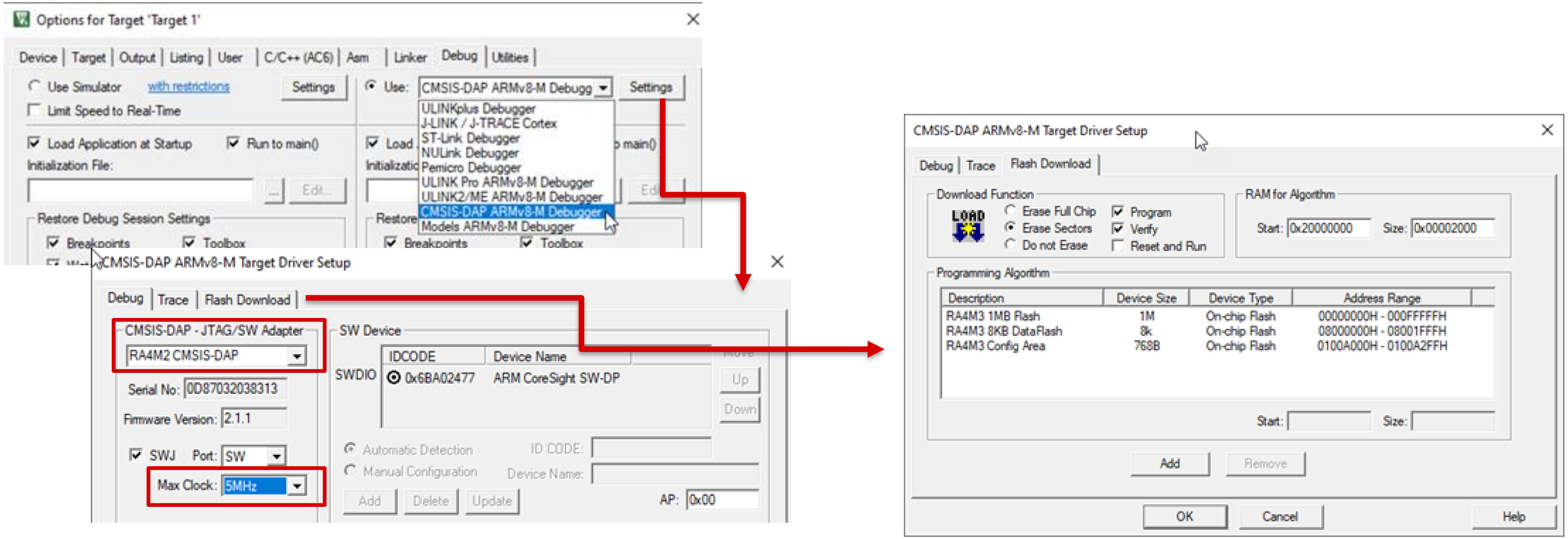
Target

Probe

CMSIS-DAP CONFIGURATION

- ❑ Use of CMSIS-DAP probes is supported directly by both Keil MDK and IAR EWARM, just by changing the debug settings as appropriate. Remember to also change the flash loader settings!

Keil MDK



Options for Target 'Target 1'

Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities

○ Use Simulator with restrictions [Settings]

☐ Limit Speed to Real-Time

☑ Load Application at Startup ☑ Run to main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

● Use: CMSIS-DAP ARMv8-M Debugg [Settings]

ULINKplus Debugger
J-LINK / J-TRACE Cortex
ST-Link Debugger
NULink Debugger
Pemicro Debugger
ULINK Pro ARMv8-M Debugger
ULINK2/ME ARMv8-M Debugger
CMSIS-DAP ARMv8-M Debugger
Models ARMv8-M Debugger

CMSIS-DAP ARMv8-M Target Driver Setup

Debug | Trace | Flash Download

CMSIS-DAP - JTAG/SW Adapter: RA4M2 CMSIS-DAP

Serial No: 0D87032038313

Firmware Version: 2.1.1

☑ SWJ Port: SW

Max Clock: 5MHz

SW Device

	IDCODE	Device Name
SWDIO	⊙ 0x6BA02477	ARM CoreSight SW-DP

● Automatic Detection ID CODE:

○ Manual Configuration Device Name:

AP: 0x00

CMSIS-DAP ARMv8-M Target Driver Setup

Debug | Trace | Flash Download

Download Function

○ Erase Full Chip ☑ Program
● Erase Sectors ☑ Verify
○ Do not Erase ☐ Reset and Run

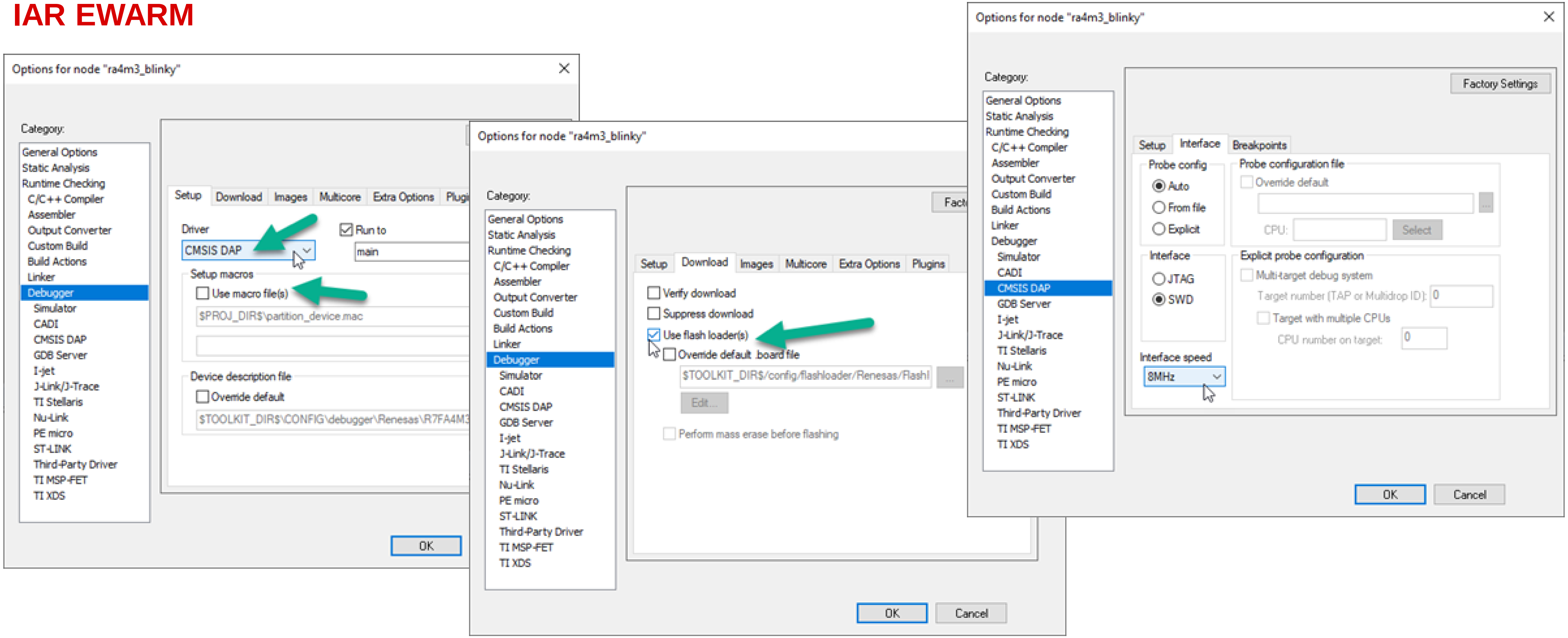
RAM for Algorithm: Start: 0x20000000 Size: 0x00002000

Programming Algorithm

Description	Device Size	Device Type	Address Range
RA4M3 1MB Flash	1M	On-chip Flash	00000000H - 000FFFFFH
RA4M3 8KB DataFlash	8k	On-chip Flash	08000000H - 08001FFFH
RA4M3 Config Area	768B	On-chip Flash	0100A000H - 0100A2FFH

CMSIS-DAP CONFIGURATION (2)

IAR EWARM



Options for node "ra4m3_blinky"

Category: Debugger

Setup

Driver: CMSIS DAP

☑ Run to: main

Setup macros

□ Use macro file(s)

$PROJ_DIR$\partition_device.mac

Device description file

□ Override default

$TOOLKIT_DIR$\CONFIG\debugger\Renesas\R7FA4M3

Options for node "ra4m3_blinky"

Category: Debugger

Download

□ Verify download

□ Suppress download

☑ Use flash loader(s)

□ Override default .board file

$TOOLKIT_DIR$/config/flashloader/Renesas/Flashl

□ Perform mass erase before flashing

Options for node "ra4m3_blinky"

Category: CMSIS DAP

Interface

Probe config: ● Auto ○ From file ○ Explicit

Probe configuration file: □ Override default

CPU:

Interface: ○ JTAG ● SWD

Explicit probe configuration

□ Multi-target debug system

Target number (TAP or Multidrop ID): 0

□ Target with multiple CPUs

CPU number on target: 0

Interface speed: 8MHz

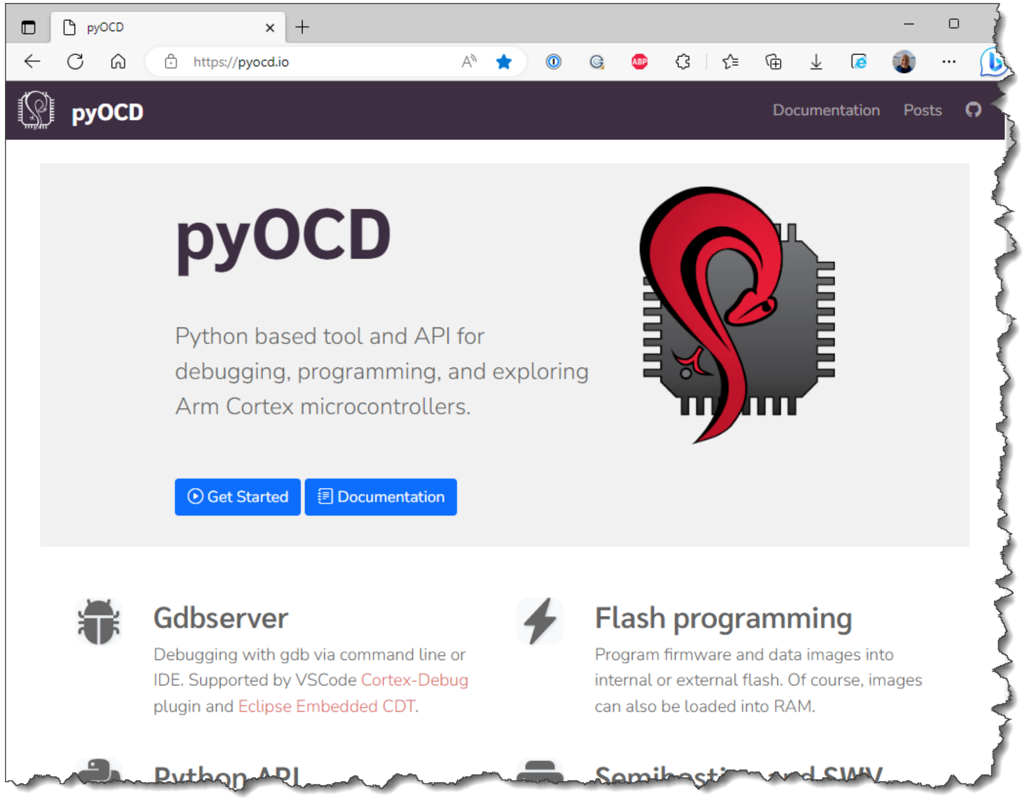
PYOCD

❑ Pyocd is an open source project that provides ability to use a CMSIS-DAP probe from command line

- Also includes a gdb server that can used from Eclipse (by installing an additional Eclipse Embedded CDT plugin) and hence provide CMSIS-DAP support within e^2 studio
  - ○ Note: Functionality inside e^2 studio that relies upon additional capabilities provided by the e^2 studio gdbserver will not work.

❑ pyOCD information and docs can be found at:

- https://pyocd.io/
- https://github.com/pyocd/pyOCD



pyOCD

Python based tool and API for debugging, programming, and exploring Arm Cortex microcontrollers.

Gdbserver

Debugging with gdb via command line or IDE. Supported by VSCode Cortex-Debug plugin and Eclipse Embedded CDT.

Flash programming

Program firmware and data images into internal or external flash. Of course, images can also be loaded into RAM.

Python API

Semihosti and SWV

INSTALLING PYOCD

- ❑ To install from DOS command line (assuming you have python3 on your path):

```
python -m pip install -U pyocd
```

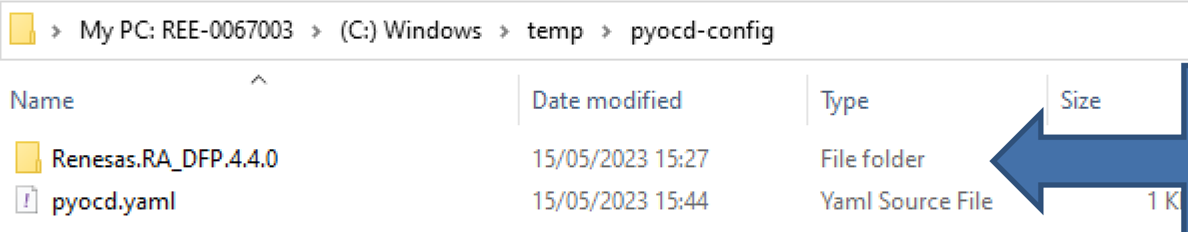
- ❑ Note: check where pyocd executables have been installed (for use later in e^2 studio). Will be something like …

```
c:\users\a5131878\appdata\local\programs\python\python310\scripts\pyocd-gdbserver.exe
c:\users\a5131878\appdata\local\programs\python\python310\scripts\pyocd.exe
```

- ❑ Also requires our MDK Device Family Pack to provide RA device knowledge, in particular the flash loader
  - Obtain this from latest FSP release :
    - https://github.com/renesas/fsp/releases

CREATE PYOCD CONFIGURATION

- ❑ Create a location to store common pyOCD configuration, including your Device Family Pack (DFP)
  - Note that there are other ways to install DFPs – but I’ve experienced issues trying to make them work

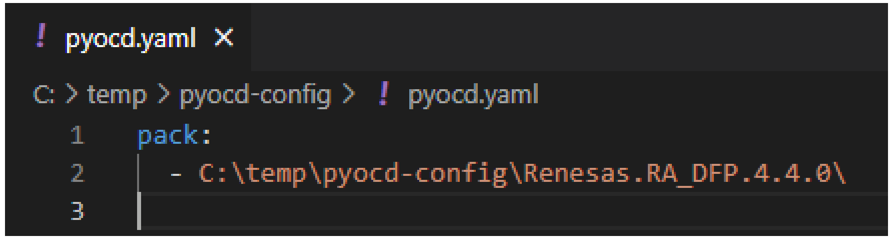


My PC: REE-0067003 › (C:) Windows › temp › pyocd-config

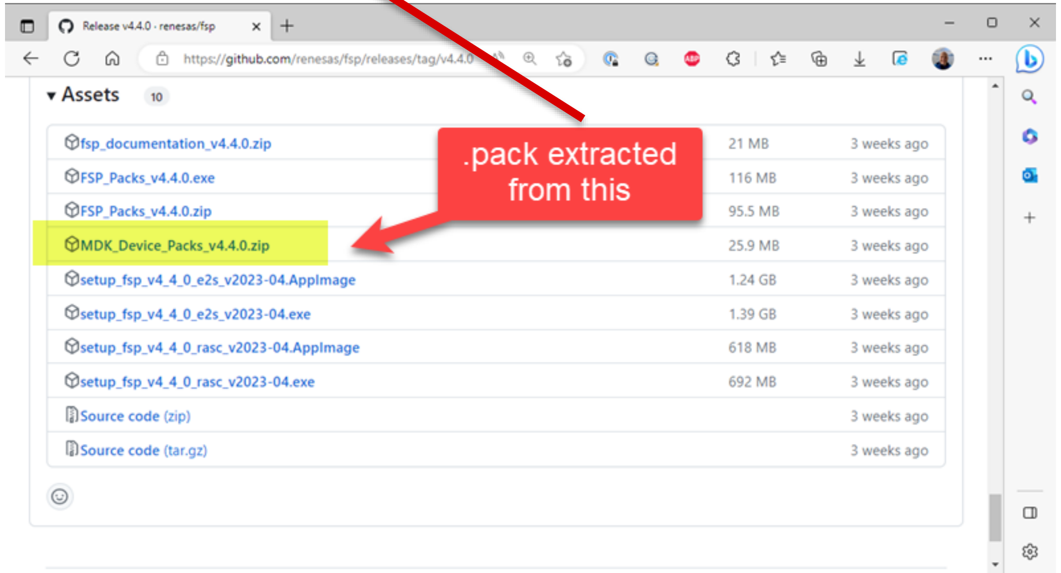
Name	Date modified	Type	Size
Renesas.RA_DFP.4.4.0	15/05/2023 15:27	File folder	
pyocd.yaml	15/05/2023 15:44	Yaml Source File	1 K

Unzipped DFP .pack file

Unzipping .pack not essential, but should improve performance slightly



```
pack:
  - C:\temp\pyocd-config\Renesas.RA_DFP.4.4.0\
```



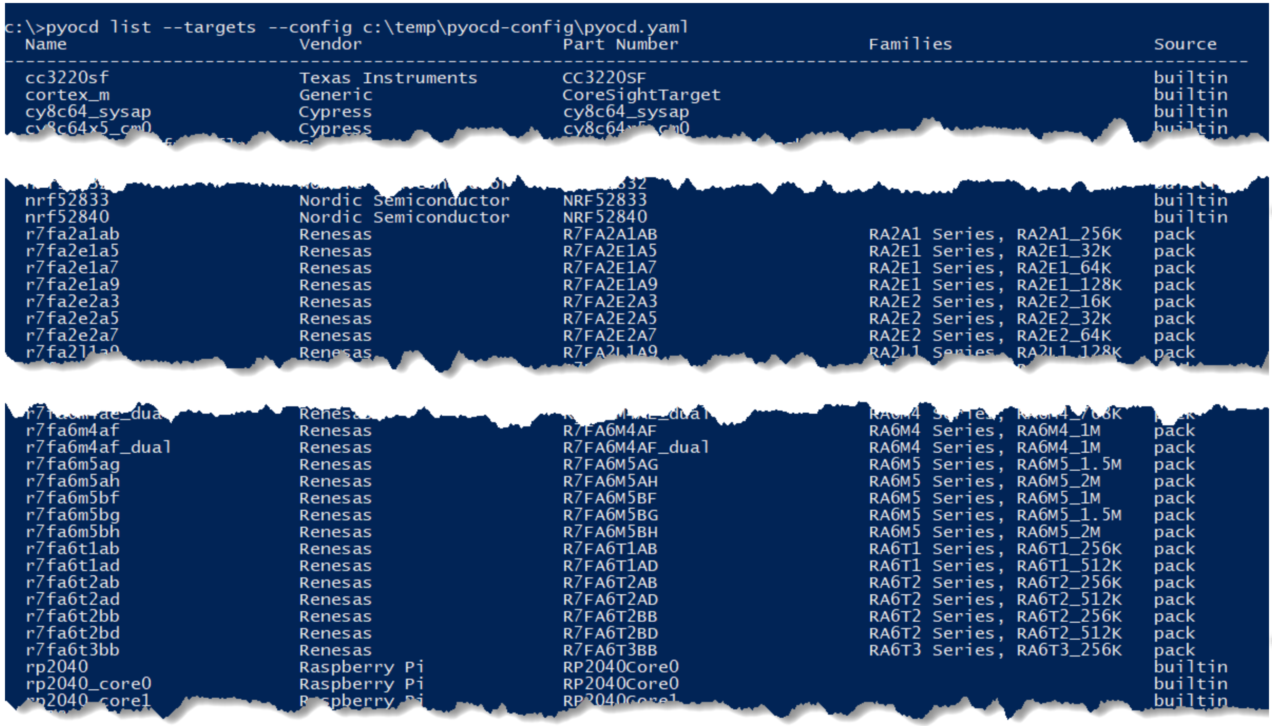
Release v4.4.0 · renesas/fsp
https://github.com/renesas/fsp/releases/tag/v4.4.0
Assets 10

fsp_documentation_v4.4.0.zip	21 MB	3 weeks ago
FSP_Packs_v4.4.0.exe	116 MB	3 weeks ago
FSP_Packs_v4.4.0.zip	95.5 MB	3 weeks ago
MDK_Device_Packs_v4.4.0.zip	25.9 MB	3 weeks ago
setup_fsp_v4_4_0_e2s_v2023-04.AppImage	1.24 GB	3 weeks ago
setup_fsp_v4_4_0_e2s_v2023-04.exe	1.39 GB	3 weeks ago
setup_fsp_v4_4_0_rasc_v2023-04.AppImage	618 MB	3 weeks ago
setup_fsp_v4_4_0_rasc_v2023-04.exe	692 MB	3 weeks ago
Source code (zip)		3 weeks ago
Source code (tar.gz)		3 weeks ago

.pack extracted from this

CHECKING PYOCD INSTALLATION FROM COMMAND LINE

- ❑ `pyocd list --targets --config c:\temp\pyocd-config\pyocd.yaml`



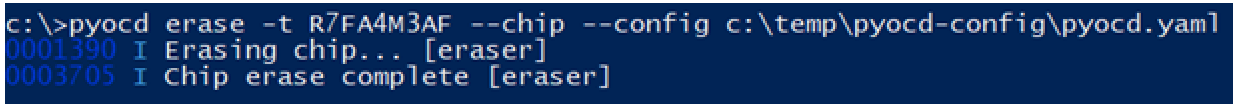
```
c:\>pyocd list --targets --config c:\temp\pyocd-config\pyocd.yaml
  Name                         Vendor                     Part Number                    Families                     Source
----------------------------------------------------------------------------------------------------------------------------
  cc3220sf                     Texas Instruments          CC3220SF                                                    builtin
  cortex_m                     Generic                    CoreSightTarget                                             builtin
  cy8c64_sysap                 Cypress                    cy8c64_sysap                                                builtin
  cy8c64x5_cm0                 Cypress                    cy8c64x5_cm0                                                builtin

  nrf52832                     Nordic Semiconductor       NRF52832                                                    builtin
  nrf52833                     Nordic Semiconductor       NRF52833                                                    builtin
  nrf52840                     Nordic Semiconductor       NRF52840                                                    builtin
  r7fa2a1ab                    Renesas                    R7FA2A1AB                      RA2A1 Series, RA2A1_256K     pack
  r7fa2e1a5                    Renesas                    R7FA2E1A5                      RA2E1 Series, RA2E1_32K      pack
  r7fa2e1a7                    Renesas                    R7FA2E1A7                      RA2E1 Series, RA2E1_64K      pack
  r7fa2e1a9                    Renesas                    R7FA2E1A9                      RA2E1 Series, RA2E1_128K     pack
  r7fa2e2a3                    Renesas                    R7FA2E2A3                      RA2E2 Series, RA2E2_16K      pack
  r7fa2e2a5                    Renesas                    R7FA2E2A5                      RA2E2 Series, RA2E2_32K      pack
  r7fa2e2a7                    Renesas                    R7FA2E2A7                      RA2E2 Series, RA2E2_64K      pack
  r7fa2l1a9                    Renesas                    R7FA2L1A9                      RA2L1 Series, RA2L1_128K     pack

  r7fa6m4ae_dual               Renesas                    R7FA6M4AE_dual                 RA6M4 Series, RA6M4_768K     pack
  r7fa6m4af                    Renesas                    R7FA6M4AF                      RA6M4 Series, RA6M4_1M       pack
  r7fa6m4af_dual               Renesas                    R7FA6M4AF_dual                 RA6M4 Series, RA6M4_1M       pack
  r7fa6m5ag                    Renesas                    R7FA6M5AG                      RA6M5 Series, RA6M5_1.5M     pack
  r7fa6m5ah                    Renesas                    R7FA6M5AH                      RA6M5 Series, RA6M5_2M       pack
  r7fa6m5bf                    Renesas                    R7FA6M5BF                      RA6M5 Series, RA6M5_1M       pack
  r7fa6m5bg                    Renesas                    R7FA6M5BG                      RA6M5 Series, RA6M5_1.5M     pack
  r7fa6m5bh                    Renesas                    R7FA6M5BH                      RA6M5 Series, RA6M5_2M       pack
  r7fa6t1ab                    Renesas                    R7FA6T1AB                      RA6T1 Series, RA6T1_256K     pack
  r7fa6t1ad                    Renesas                    R7FA6T1AD                      RA6T1 Series, RA6T1_512K     pack
  r7fa6t2ab                    Renesas                    R7FA6T2AB                      RA6T2 Series, RA6T2_256K     pack
  r7fa6t2ad                    Renesas                    R7FA6T2AD                      RA6T2 Series, RA6T2_512K     pack
  r7fa6t2bb                    Renesas                    R7FA6T2BB                      RA6T2 Series, RA6T2_256K     pack
  r7fa6t2bd                    Renesas                    R7FA6T2BD                      RA6T2 Series, RA6T2_512K     pack
  r7fa6t3bb                    Renesas                    R7FA6T3BB                      RA6T3 Series, RA6T3_256K     pack
  rp2040                       Raspberry Pi               RP2040Core0                                                 builtin
  rp2040_core0                 Raspberry Pi               RP2040Core0                                                 builtin
  rp2040_core1                 Raspberry Pi               RP2040Core1                                                 builtin
```

RA MCUs being picked up from our DFP

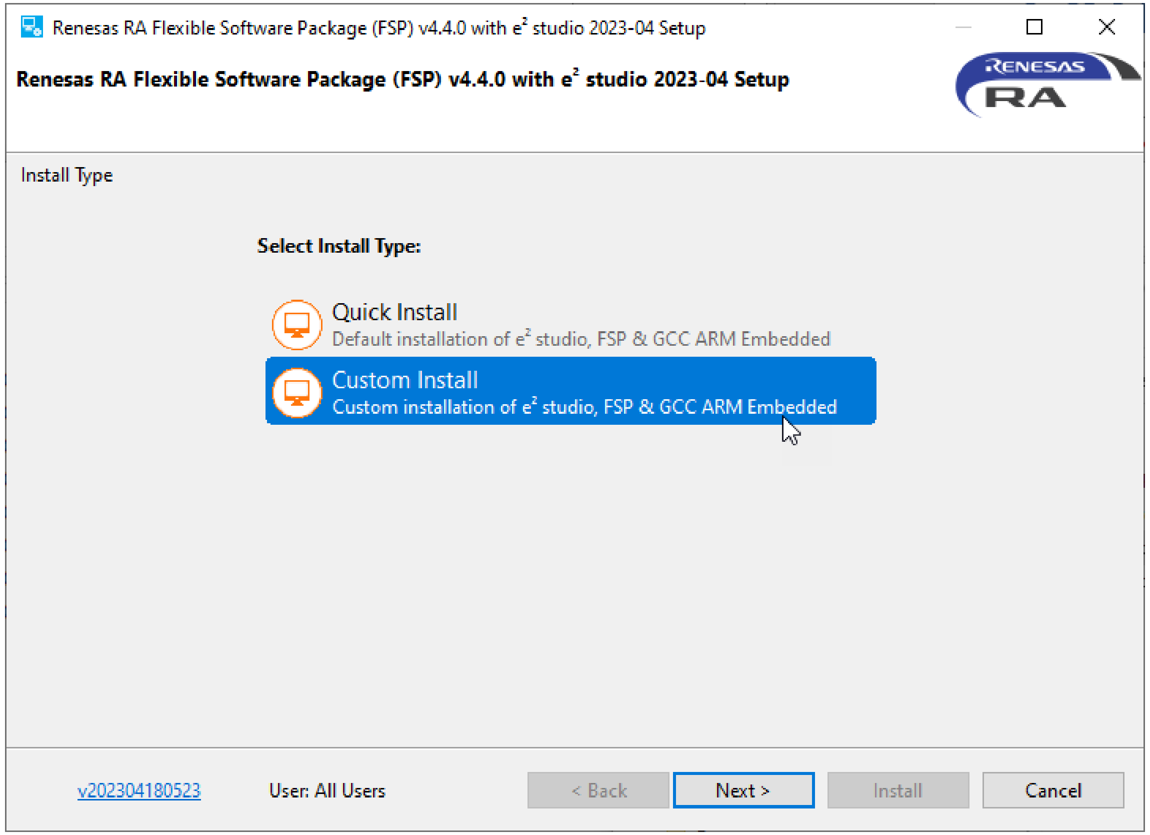
- ❑ Many other operations possible:



```
c:\>pyocd erase -t R7FA4M3AF --chip --config c:\temp\pyocd-config\pyocd.yaml
0001390 I Erasing chip... [eraser]
0003705 I Chip erase complete [eraser]
```

INSTALLING PYOCD SUPPORT PLUGIN INTO E^2 STUDIO

❑ Available as an optional feature in 2023-04 custom install ….



Renesas RA Flexible Software Package (FSP) v4.4.0 with e^2 studio 2023-04 Setup

Renesas RA Flexible Software Package (FSP) v4.4.0 with e^2 studio 2023-04 Setup

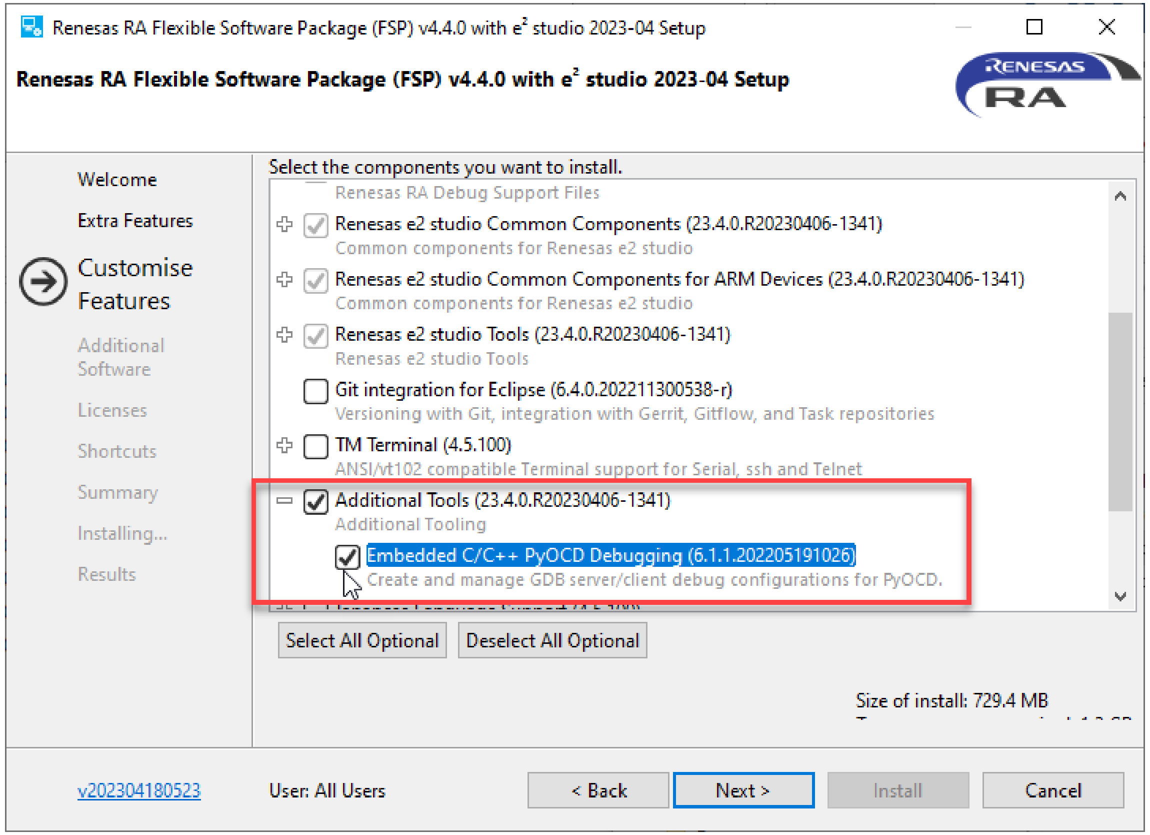
Install Type

Select Install Type:

○ Quick Install
Default installation of e^2 studio, FSP & GCC ARM Embedded

● Custom Install
Custom installation of e^2 studio, FSP & GCC ARM Embedded

v202304180523 User: All Users



Renesas RA Flexible Software Package (FSP) v4.4.0 with e^2 studio 2023-04 Setup

Renesas RA Flexible Software Package (FSP) v4.4.0 with e^2 studio 2023-04 Setup

Select the components you want to install.

Renesas RA Debug Support Files

☑ Renesas e2 studio Common Components (23.4.0.R20230406-1341)
Common components for Renesas e2 studio

☑ Renesas e2 studio Common Components for ARM Devices (23.4.0.R20230406-1341)
Common components for Renesas e2 studio

☑ Renesas e2 studio Tools (23.4.0.R20230406-1341)
Renesas e2 studio Tools

☐ Git integration for Eclipse (6.4.0.202211300538-r)
Versioning with Git, integration with Gerrit, Gitflow, and Task repositories

☐ TM Terminal (4.5.100)
ANSI/vt102 compatible Terminal support for Serial, ssh and Telnet

☑ Additional Tools (23.4.0.R20230406-1341)
Additional Tooling

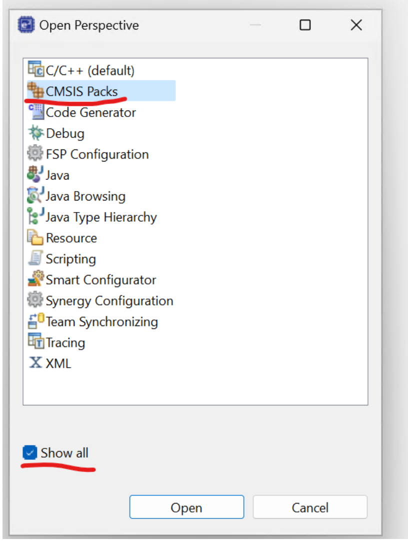
☑ Embedded C/C++ PyOCD Debugging (6.1.1.202205191026)
Create and manage GDB server/client debug configurations for PyOCD.

Size of install: 729.4 MB

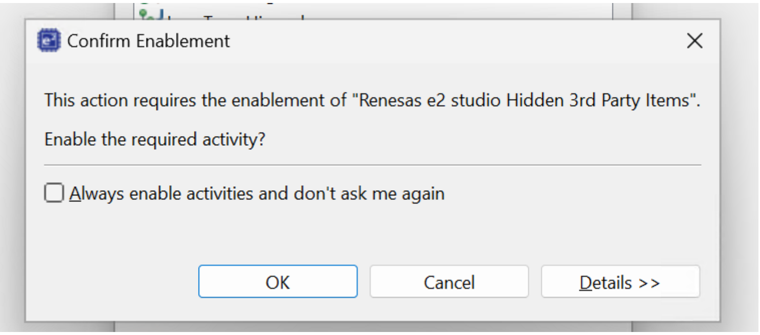
v202304180523 User: All Users

TO MAKE PYOCD DEBUG CONFIGURATIONS ACCESSIBLE…

- Recent “UI simplifications” mean that further configuration required to make pyOCD accessible after installing it ☹ And for each new workspace ☹ ☹
- Go to : *Window -> Perspectives -> Open Perspective -> Other…*
  - Select “*CMSIS Packs*” and click OK to confirm activity enablement
- This will switch e^2 studio into the CMSIS-Pack perspective, but this can just be closed.
- Can also remove “CMSIS-Packs” button from C/C++ Perspective Toolbar from *Window -> Perspective -> Customize Perspective*
- Talking to Engineering team to improve this for future releases



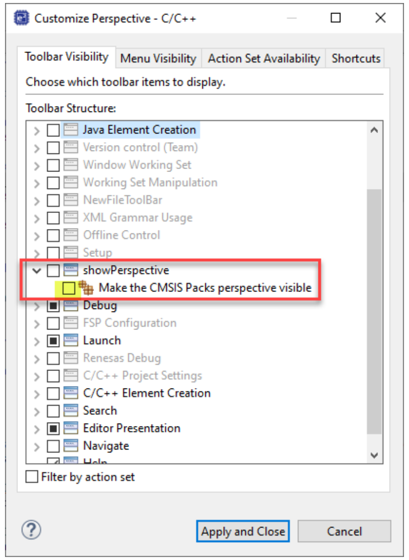
☑ Show all



This action requires the enablement of "Renesas e2 studio Hidden 3rd Party Items".

Enable the required activity?

☐ Always enable activities and don't ask me again



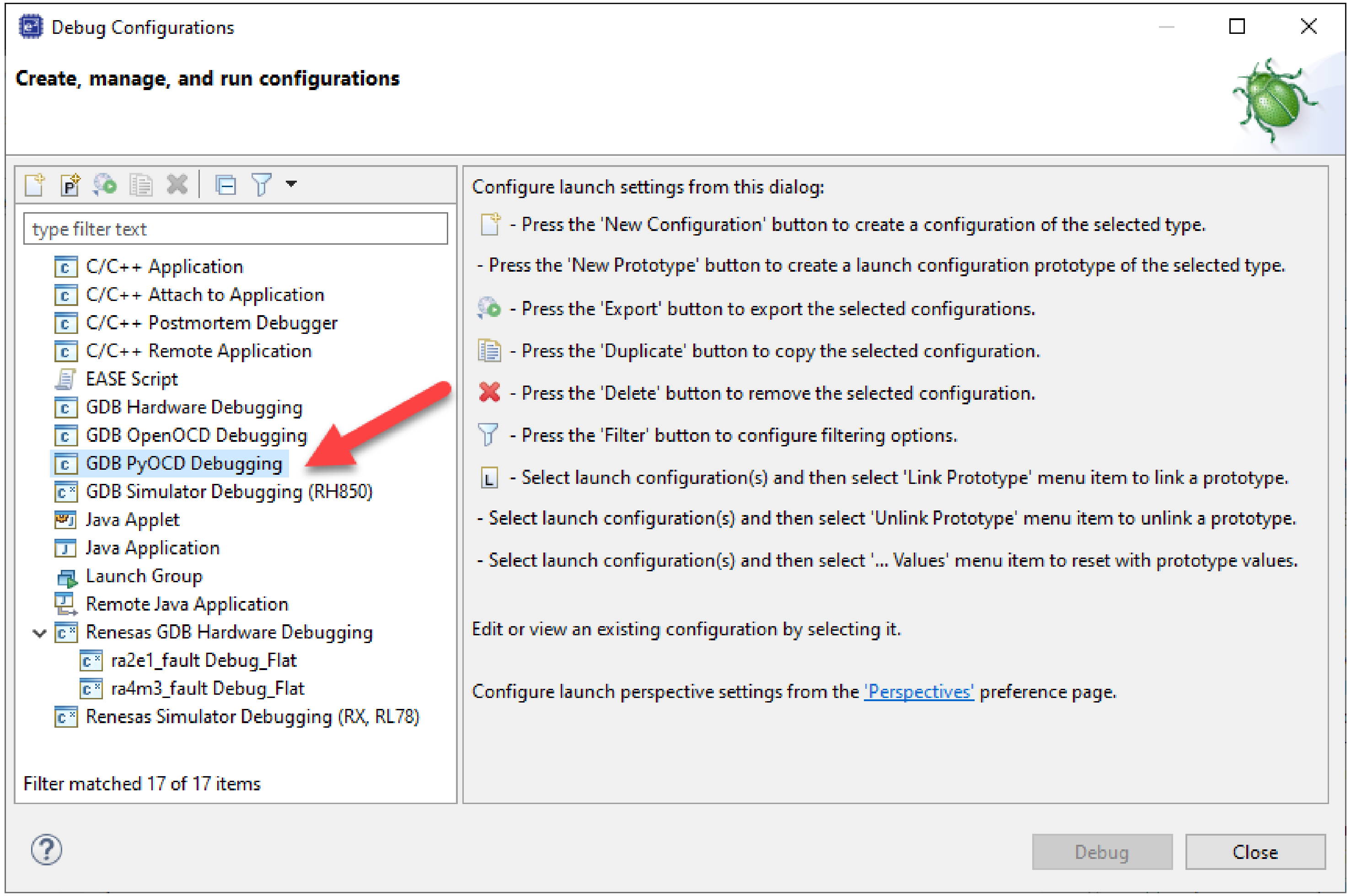
Choose which toolbar items to display.

Toolbar Structure:

☐ Java Element Creation
☐ Version control (Team)
☐ Window Working Set
☐ Working Set Manipulation
☐ NewFileToolBar
☐ XML Grammar Usage
☐ Offline Control
☐ Setup
☐ showPerspective
  ☐ Make the CMSIS Packs perspective visible
■ Debug
☐ FSP Configuration
■ Launch
☐ Renesas Debug
☐ C/C++ Project Settings
☐ C/C++ Element Creation
☐ Search
■ Editor Presentation
☐ Navigate

☐ Filter by action set

PYOCD DEBUG CONFIGURATION NOW ACCESSIBLE



Create, manage, and run configurations

type filter text

- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- EASE Script
- GDB Hardware Debugging
- GDB OpenOCD Debugging
- GDB PyOCD Debugging
- GDB Simulator Debugging (RH850)
- Java Applet
- Java Application
- Launch Group
- Remote Java Application
- Renesas GDB Hardware Debugging
  - ra2e1_fault Debug_Flat
  - ra4m3_fault Debug_Flat
- Renesas Simulator Debugging (RX, RL78)

Filter matched 17 of 17 items

Configure launch settings from this dialog:

- Press the 'New Configuration' button to create a configuration of the selected type.

- Press the 'New Prototype' button to create a launch configuration prototype of the selected type.

- Press the 'Export' button to export the selected configurations.

- Press the 'Duplicate' button to copy the selected configuration.

- Press the 'Delete' button to remove the selected configuration.

- Press the 'Filter' button to configure filtering options.

- Select launch configuration(s) and then select 'Link Prototype' menu item to link a prototype.

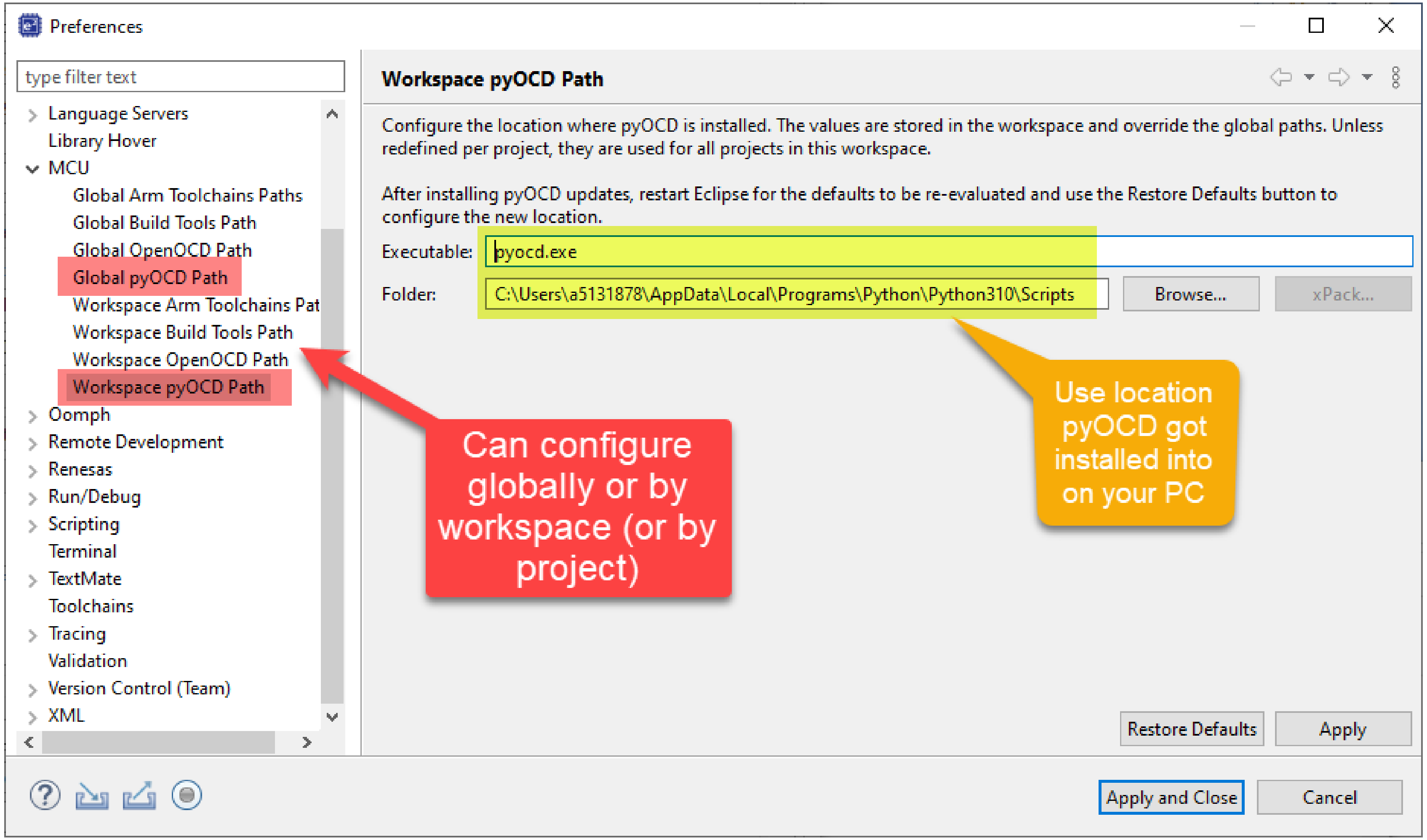
- Select launch configuration(s) and then select 'Unlink Prototype' menu item to unlink a prototype.

- Select launch configuration(s) and then select '... Values' menu item to reset with prototype values.

Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

CONFIGURE E^2 STUDIO TO FIND PYOCD



Workspace pyOCD Path

Configure the location where pyOCD is installed. The values are stored in the workspace and override the global paths. Unless redefined per project, they are used for all projects in this workspace.

After installing pyOCD updates, restart Eclipse for the defaults to be re-evaluated and use the Restore Defaults button to configure the new location.

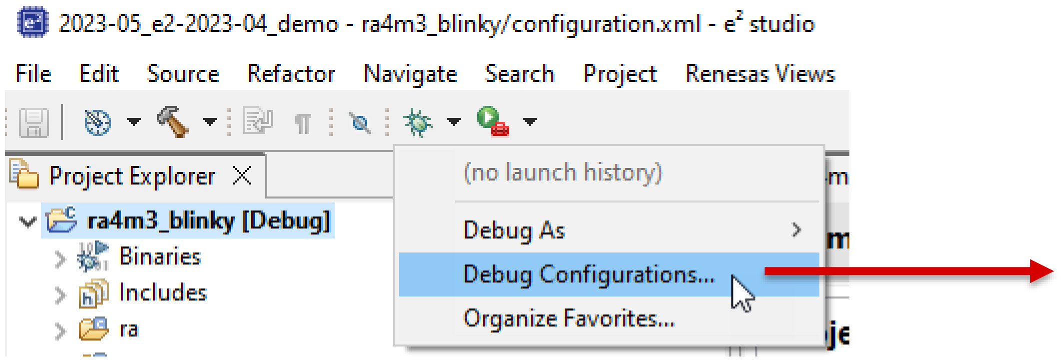
Executable: pyocd.exe

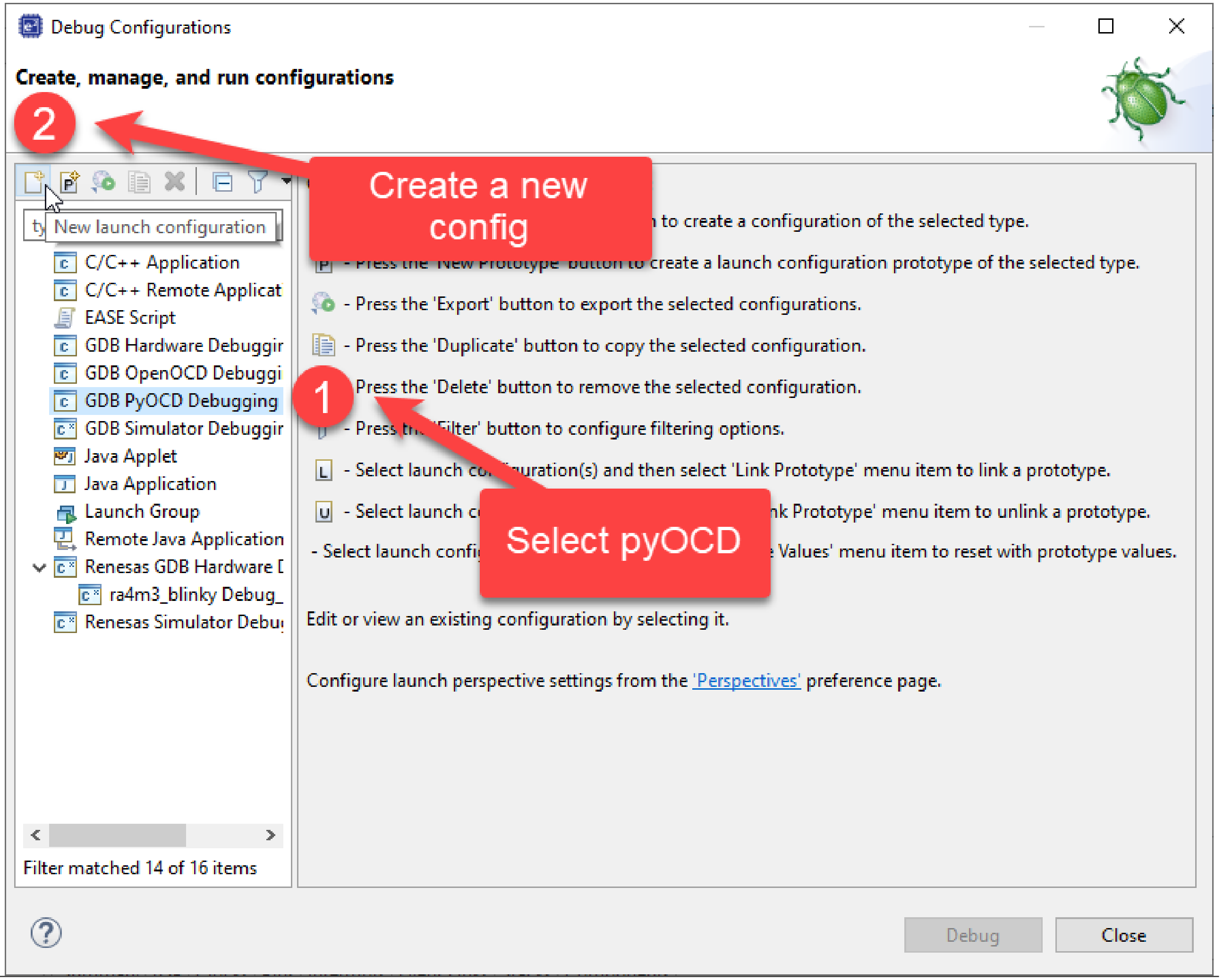
Folder: C:\Users\a5131878\AppData\Local\Programs\Python\Python310\Scripts

Can configure globally or by workspace (or by project)

Use location pyOCD got installed into on your PC

CREATING A DEBUG CONFIG FOR PYOCD (1)





n to create a configuration of the selected type.

- Press the 'New Prototype' button to create a launch configuration prototype of the selected type.

- Press the 'Export' button to export the selected configurations.

- Press the 'Duplicate' button to copy the selected configuration.

Press the 'Delete' button to remove the selected configuration.

- Press th 'Filter' button to configure filtering options.

- Select launch co uration(s) and then select 'Link Prototype' menu item to link a prototype.

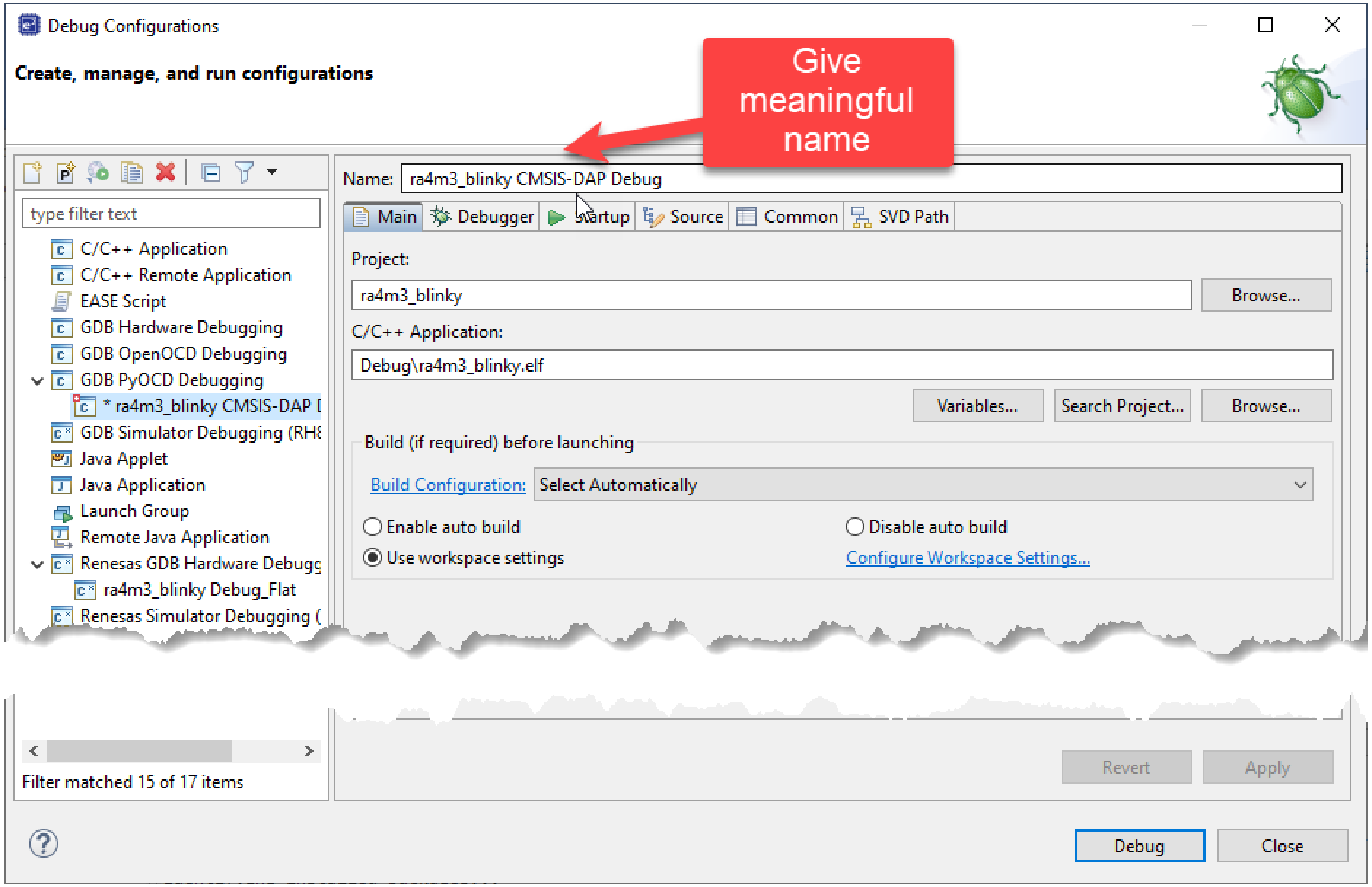
- Select launch c nk Prototype' menu item to unlink a prototype.

- Select launch confi Values' menu item to reset with prototype values.

Edit or view an existing configuration by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

CREATING A DEBUG CONFIG FOR PYOCD (2)



Debug Configurations

Create, manage, and run configurations

Give meaningful name

type filter text

- C/C++ Application
- C/C++ Remote Application
- EASE Script
- GDB Hardware Debugging
- GDB OpenOCD Debugging
- GDB PyOCD Debugging
  - * ra4m3_blinky CMSIS-DAP [
- GDB Simulator Debugging (RH8
- Java Applet
- Java Application
- Launch Group
- Remote Java Application
- Renesas GDB Hardware Debugg
  - ra4m3_blinky Debug_Flat
- Renesas Simulator Debugging (

Filter matched 15 of 17 items

Name: ra4m3_blinky CMSIS-DAP Debug

Main | Debugger | Startup | Source | Common | SVD Path

Project:
ra4m3_blinky

C/C++ Application:
Debug\ra4m3_blinky.elf

Build (if required) before launching

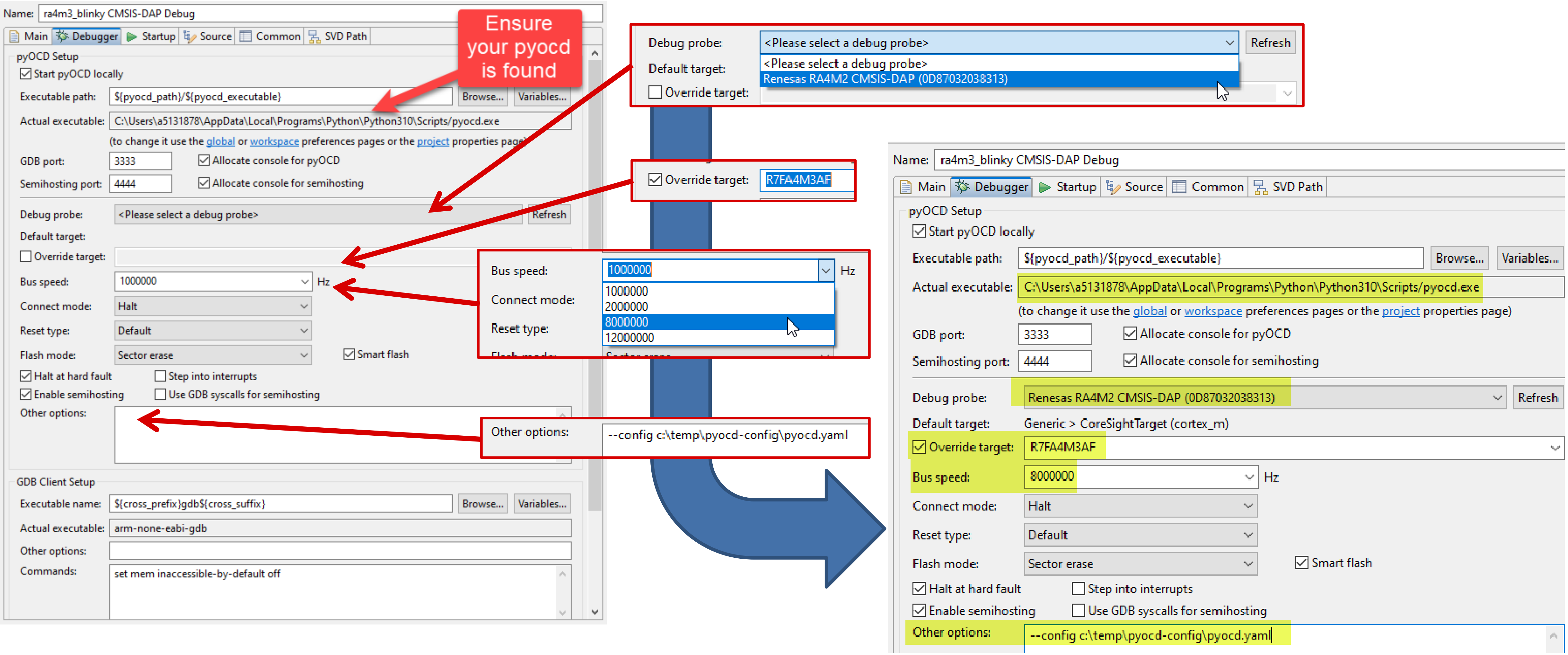
Build Configuration: Select Automatically

○ Enable auto build ○ Disable auto build

● Use workspace settings

Configure Workspace Settings...

CREATING A DEBUG CONFIG FOR PYOCD (3)



Name: ra4m3_blinky CMSIS-DAP Debug

Ensure your pyocd is found

pyOCD Setup

☑ Start pyOCD locally

Executable path: ${pyocd_path}/${pyocd_executable}

Actual executable: C:\Users\a5131878\AppData\Local\Programs\Python\Python310\Scripts/pyocd.exe

(to change it use the global or workspace preferences pages or the project properties page)

GDB port: 3333 ☑ Allocate console for pyOCD

Semihosting port: 4444 ☑ Allocate console for semihosting

Debug probe: <Please select a debug probe>

Default target:

☐ Override target:

Bus speed: 1000000 Hz

Connect mode: Halt

Reset type: Default

Flash mode: Sector erase ☑ Smart flash

☑ Halt at hard fault ☐ Step into interrupts

☑ Enable semihosting ☐ Use GDB syscalls for semihosting

Other options:

GDB Client Setup

Executable name: ${cross_prefix}gdb${cross_suffix}

Actual executable: arm-none-eabi-gdb

Other options:

Commands: set mem inaccessible-by-default off

Debug probe: <Please select a debug probe>

<Please select a debug probe>
Renesas RA4M2 CMSIS-DAP (0D87032038313)

Default target:

☐ Override target:

☑ Override target: R7FA4M3AF

Bus speed: 1000000 Hz

1000000
2000000
8000000
12000000

Connect mode:

Reset type:

Other options: --config c:\temp\pyocd-config\pyocd.yaml

Name: ra4m3_blinky CMSIS-DAP Debug

pyOCD Setup

☑ Start pyOCD locally

Executable path: ${pyocd_path}/${pyocd_executable}

Actual executable: C:\Users\a5131878\AppData\Local\Programs\Python\Python310\Scripts/pyocd.exe

(to change it use the global or workspace preferences pages or the project properties page)

GDB port: 3333 ☑ Allocate console for pyOCD

Semihosting port: 4444 ☑ Allocate console for semihosting

Debug probe: Renesas RA4M2 CMSIS-DAP (0D87032038313)

Default target: Generic > CoreSightTarget (cortex_m)

☑ Override target: R7FA4M3AF

Bus speed: 8000000 Hz

Connect mode: Halt

Reset type: Default

Flash mode: Sector erase ☑ Smart flash

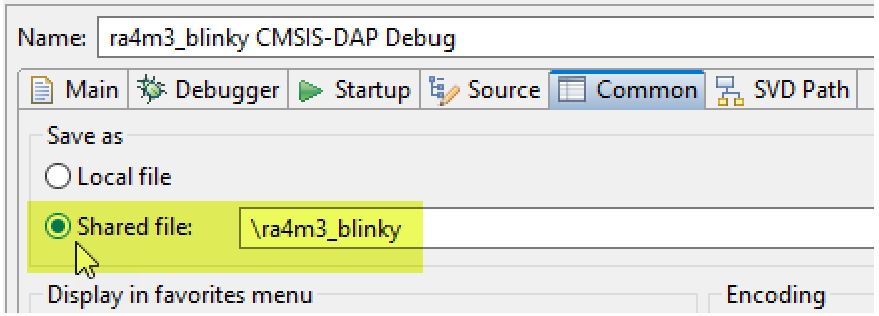
☑ Halt at hard fault ☐ Step into interrupts

☑ Enable semihosting ☐ Use GDB syscalls for semihosting

Other options: --config c:\temp\pyocd-config\pyocd.yaml

CREATING A DEBUG CONFIG FOR PYOCD (4)

- ❑ Make launch configuration part of project (as per our normal launch configurations)



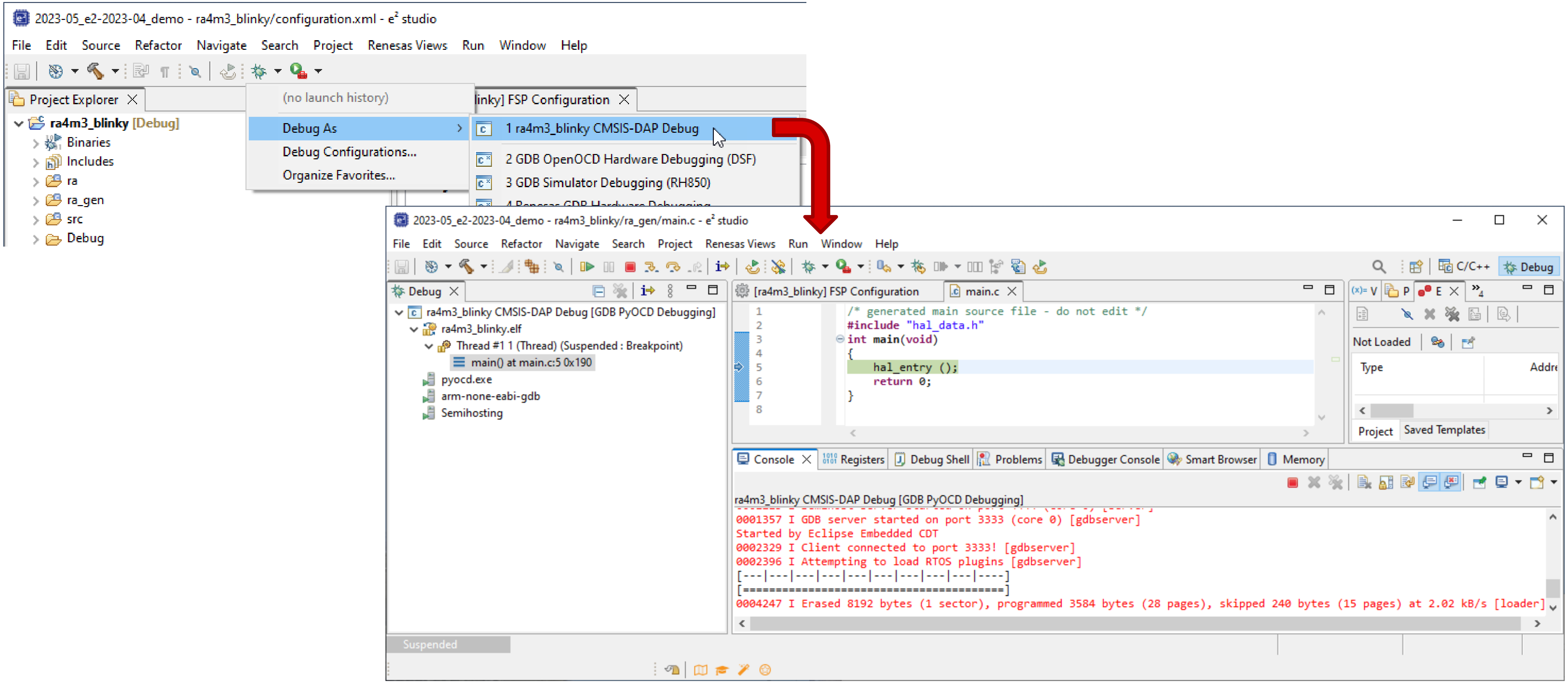
Name: ra4m3_blinky CMSIS-DAP Debug

Save as

○ Local file

● Shared file: \ra4m3_blinky

LAUNCH DEBUG SESSION VIA PYOCD & CMSIS-DAP



2023-05_e2-2023-04_demo - ra4m3_blinky/configuration.xml - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
Project Explorer
ra4m3_blinky [Debug]
Binaries
Includes
ra
ra_gen
src
Debug
(no launch history)
Debug As
Debug Configurations...
Organize Favorites...
1 ra4m3_blinky CMSIS-DAP Debug
2 GDB OpenOCD Hardware Debugging (DSF)
3 GDB Simulator Debugging (RH850)
4 Renesas GDB Hardware Debugging
[ra4m3_blinky] FSP Configuration
2023-05_e2-2023-04_demo - ra4m3_blinky/ra_gen/main.c - e² studio
File Edit Source Refactor Navigate Search Project Renesas Views Run Window Help
C/C++
Debug
Debug
ra4m3_blinky CMSIS-DAP Debug [GDB PyOCD Debugging]
ra4m3_blinky.elf
Thread #1 1 (Thread) (Suspended : Breakpoint)
main() at main.c:5 0x190
pyocd.exe
arm-none-eabi-gdb
Semihosting
[ra4m3_blinky] FSP Configuration
main.c
Not Loaded
Type
Addre
Project
Saved Templates
Console
Registers
Debug Shell
Problems
Debugger Console
Smart Browser
Memory
Suspended

```
/* generated main source file - do not edit */
#include "hal_data.h"
int main(void)
{
    hal_entry ();
    return 0;
}
```

```
ra4m3_blinky CMSIS-DAP Debug [GDB PyOCD Debugging]
0001357 I GDB server started on port 3333 (core 0) [gdbserver]
Started by Eclipse Embedded CDT
0002329 I Client connected to port 3333! [gdbserver]
0002396 I Attempting to load RTOS plugins [gdbserver]
[---|---|---|---|---|---|---|---|---|----]
[========================================]
0004247 I Erased 8192 bytes (1 sector), programmed 3584 bytes (28 pages), skipped 240 bytes (15 pages) at 2.02 kB/s [loader]
```