

Getting Started with OpenOCD

Contents

1.	Introduction.....	2
1.1.	OpenOCD Feature Support for the WBZ451, WBZ451HPE, WBZ450 and PIC32WM Curiosity Board.....	2
2.	System Requirements.....	2
3.	CMSIS-DAP Switcher.....	3
3.1.	Software and Package Dependencies	3
3.2.	Installation Procedure	4
3.3.	Supported Debuggers.....	4
3.3.1.	Switching PKOB4 Firmware from MPLAB to CMSIS-DAP	4
3.3.2.	Switching PKOB4 Firmware from CMSIS-DAP to MPLAB	5
4.	OpenOCD Integration with Visual Studio Code.....	5
5.	Steps to Flashing/Programming	10
5.1.	WBZ451 Device.....	10
5.2.	WBZ653 Device.....	11
6.	Steps to Debugging the WBZ451, WBZ451HPE, WBZ450 & WBZ653 Device	12
7.	References / Resources:	25
8.	Abbreviations:	25
9.	Document History:.....	25

1. Introduction

OpenOCD (Open On-Chip Debugger) is an open-source tool that provides debugging, in-system programming, and boundary-scan testing for embedded target devices. It is widely used in embedded systems development to interface with a wide range of hardware devices through JTAG, SWD, or other debug interfaces.

1.1. OpenOCD Feature Support for the WBZ451, WBZ451HPE, WBZ450 and PIC32WM Curiosity Board

Feature	Description	Supported (Yes/No)
Flash Programming	Supports firmware flashing to the target device	Yes
Debug Support	Supports debugging through the SWD (Serial Wire Debug) interface	Yes
Breakpoint Support	Supports up to four hardware breakpoints	Yes
Reset Control	Supports device reset via OpenOCD	Yes
SWD Interface	Supports Serial Wire Debug interface	Yes
GDB Integration	Compatible with GDB debugger	Yes

2. System Requirements

- WBZ451 Curiosity Board 1 No
- WBZ450 Curiosity Board 1 No
- WBZ451HPE Curiosity Board 1 No
- WBZ653 Curiosity Board 1 No
- MicroUSB cable 1 No
- Packages
 - “Zephyr-sdk-0.17.0”
 - “openocd_support_wbz_pic32wm”
 - Link to download the openOCD prebuilt binaries
https://github.com/MicrochipTech/openOCD-wireless/tree/main/prebuilt_binaries/windows/openocd_support_wbz_pic32wm
 - “Zephyrproject” (Repository)

- Pkob4-cmsis_dap switcher
 - Link to download the pkob4-cmsis_dap switcher(https://github.com/MicrochipTech/openOCD-wireless/tree/main/pkob4-cmsis_dap-switcher)

Note: Please ensure the following **folders** are available under the “C:\developers\” directory:

- “Zephyr-sdk-0.17.0”
 - “openocd_support_wbz_pic32wm”
 - “Zephyrproject”
 - “pkob4-cmsis_dap-switcher”
-
- Visual Studio Code v1.101.2 or above.
 - Python version 3.10 or newer
 - PICKit Basic (Optional)
 - Type C USB cable 1 No

3. CMSIS-DAP Switcher

pycmsisdapswitcher is a command line Python utility designed to switch the firmware on a PICkit™ Basic tool or on an evaluation board/kit between the Microchip® implementation and the Arm® CMSIS-DAP v2 implementation. It allows users to easily switch the firmware of supported Microchip tools, enabling compatibility with different development environments.

Important:

Download the pkob4-cmsis_dap-switcher folder from (https://github.com/MicrochipTech/openOCD-wireless/tree/main/pkob4-cmsis_dap-switcher) in to your local machine “C:\developers\”

3.1. Software and Package Dependencies

Requires Python version 3.10 or newer

3.2. Installation Procedure

Step-1: - Install “**pycmsisdapswitcher**” via “PyPI” using pip:

Step-2: - pip install pycmsisdapswitcher

3.3. Supported Debuggers

The **pycmsisdapswitcher** utility requires a target parameter, which specifies either the tool name or the kit name to be used.

Supported debuggers

- evalboard (PKOB4)

3.3.1. Switching PKOB4 Firmware from MPLAB to CMSIS-DAP

To switch an evaluation board equipped with **PKOB4** firmware from the **MPLAB®** implementation to the latest **CMSIS-DAP v2** firmware using the Microchip server, follow these steps:

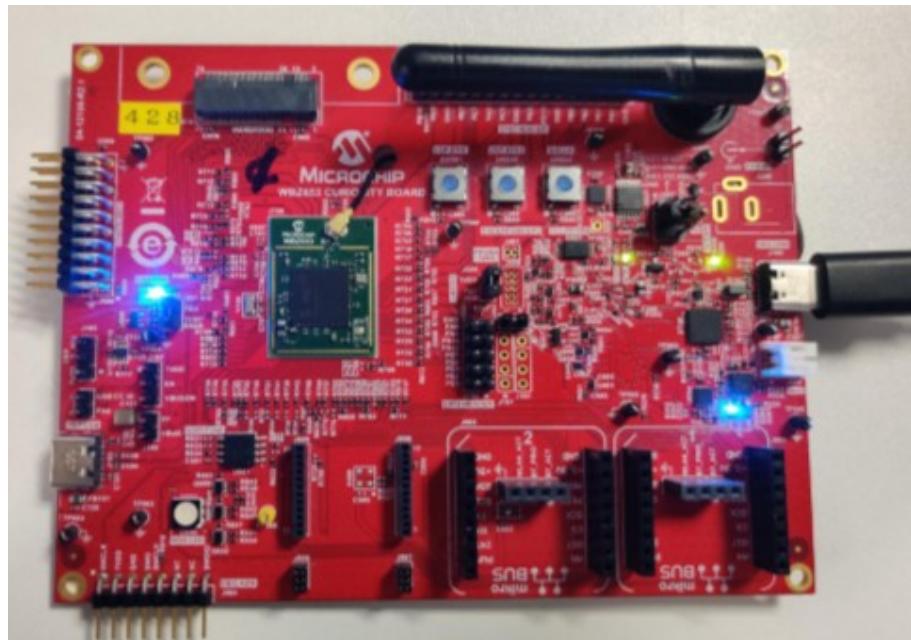
- Connect the Curiosity board to your computer via USB.
- Navigate to the “C:\developers\pkob4-cmsis_dap-switcher\” folder and run the below command from **command prompt**.

Note: Make sure you have copied the entire **pkob4-cmsis_dap-switcher** folder along with all its files into the location “**C:\developers**” from (https://github.com/MicrochipTech/openOCD-wireless/tree/main/pkob4-cmsis_dap-switcher)

Command: python -m pycmsisdapswitcher --action switch --target=evalboard --source=C:\developers\pkob4-cmsis_dap-switcher\pkob4_app_cmsis-dap.hex --fwtype=cmsis

```
C:\developers\pkob4-cmsis_dap-switcher>py -m pycmsisdapswitcher --action switch --target=evalboard --source=C:\developers\pkob4-cmsis_dap-switcher\pkob4_app_cmsis-dap.hex --fwtype=cmsis
16:15:16 - application switch started
16:15:19 - found evaluation board, S/N RYN233502666 running mplab firmware
16:15:19 - using application file C:\developers\pkob4-cmsis_dap-switcher\pkob4_app_cmsis-dap.hex
16:15:22 - application being written and verified
16:15:23 - application write and verify completed, waiting for evaluation board ready ..
16:15:29 - application switch successful
```





3.3.2. Switching PKOB4 Firmware from CMSIS-DAP to MPLAB

To revert an evaluation board's **PKOB4** firmware from the CMSIS-DAP implementation back to the **MPLAB (PKOB4)** firmware using a local HEX file, follow these steps:

- Connect the Curiosity board to your computer via USB.
- Navigate to the “C:\developers\pkob4-cmsis_dap-switcher\” folder and run the below command from command prompt.

Note: Make sure you have copied the entire **pkob4-cmsis_dap-switcher** folder along with all its files into the location “**C:\developers**” from (https://github.com/MicrochipTech/openOCD-wireless/tree/main/pkob4-cmsis_dap-switcher)

Command: python -m pycmsisdapswitcher --action switch --target=evalboard --source=C:\developers\pkob4-cmsis_dap-switcher\pkob4_app.hex --fwtype=mlab

```
C:\developers\pkob4-cmsis_dap-switcher>py -m pycmsisdapswitcher --action switch --target=evalboard --source=C:\developers\pkob4-cmsis_dap-switcher\pkob4_app.hex --fwtype=mlab
16:19:21 - application switch started
16:19:23 - found evaluation board, S/N RYN233502666 running cmsisv2 firmware
16:19:23 - using application file C:\developers\pkob4-cmsis_dap-switcher\pkob4_app.hex
16:19:27 - application being written and verified
16:19:28 - application write and verify completed, waiting for evaluation board ready ...
16:19:35 - application switch successful
```

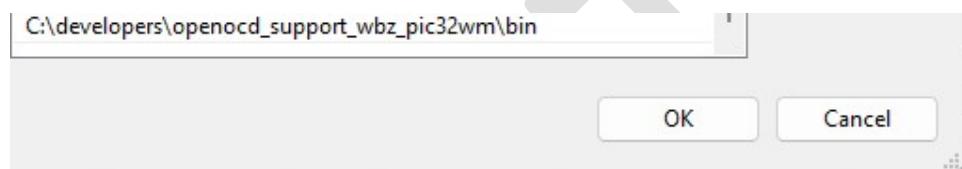
4. OpenOCD Integration with Visual Studio Code

To integrate **OpenOCD** with Visual Studio Code (VSCode) for WBZ451, WBZ451HPE, WBZ450 and PIC32WM development, follow the steps below:

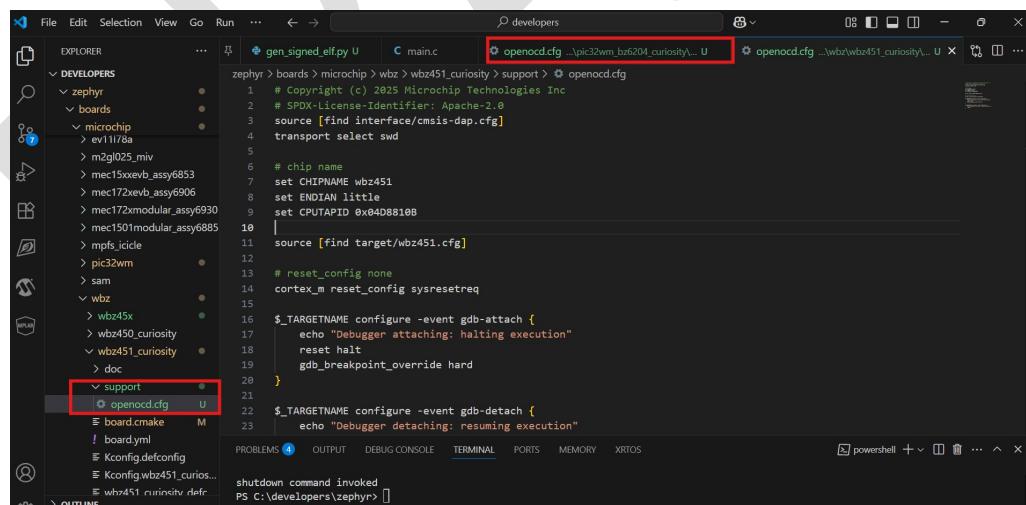
Step-1: - Extract the “openocd_support_wbz_pic32wm” package downloaded from (https://github.com/MicrochipTech/openOCD-wireless/tree/main/prebuilt_binaries/windows/openocd_support_wbz_pic32wm) and copy the openocd_support_wbz_pic32wm folder to “C:/developers/”

Step-2: - Add path “C:/developers/openocd_support_wbz_pic32wm/bin” to your system's Environment Variables

- Open System Environment Variables → Click Environment Variables → Under System variables, find and edit the Path variable.
- Add the copied path to the list and click OK.



Step-3: - In VSCode, create a new folder named support, and inside it, add a new file named openocd.cfg as shown below. This file will contain the OpenOCD configuration settings required for debugging.



Step-4: - Copy and paste the following content into the openocd.cfg file—you can use the same configuration as shown below:

```
# Copyright (c) 2025 Microchip Technologies Inc
```

```
# SPDX-License-Identifier: Apache-2.0
source [find interface/cmsis-dap.cfg]
transport select swd

# chip name
set CHIPNAME wbz451
set ENDIAN little
set CPUTAPID 0x04D8810B

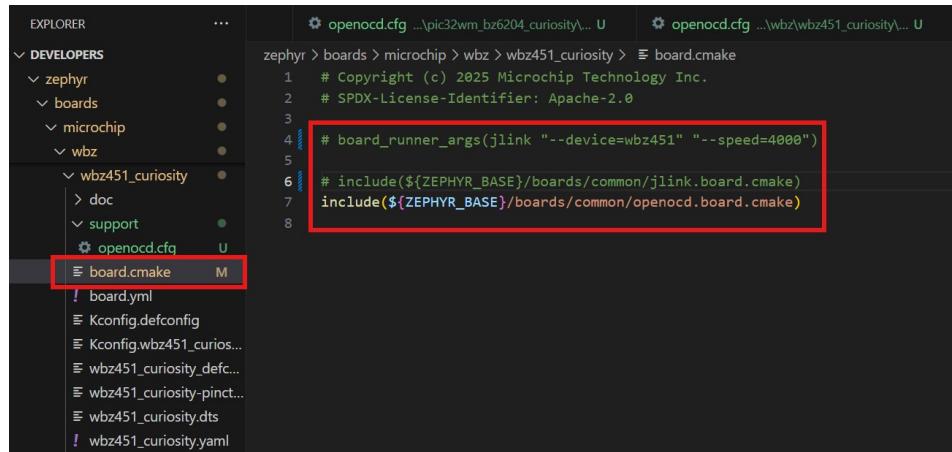
source [find target/wbz451.cfg]

cortex_m reset_config sysresetreq

$_TARGETNAME configure -event gdb-attach {
    echo "Debugger attaching: halting execution"
    reset halt
    gdb_breakpoint_override hard
}

$_TARGETNAME configure -event gdb-detach {
    echo "Debugger detaching: resuming execution"
    resume
}
```

Step-5: - Ensure the below content is present under the “board.cmake” file



```
zephyr > boards > microchip > wbz > wbz451_curiosity > board.cmake
1 # Copyright (c) 2025 Microchip Technology Inc.
2 # SPDX-License-Identifier: Apache-2.0
3
4 # board_runner_args(jlink "--device=wbz451" "--speed=4000")
5
6 # include(${ZEPHYR_BASE}/boards/common/jlink.board.cmake)
7 include(${ZEPHYR_BASE}/boards/common/openocd.board.cmake)
8
```

- Note: The changes is applicable for wbz451hpe_curiosity, wbz450_curiosity and pic32wm_bz6204_curiosity board.cmake as well.

Copy and paste the following content into the “board.cmake” file.

```
# Copyright (c) 2025 Microchip Technology Inc.  
# SPDX-License-Identifier: Apache-2.0  
  
# board_runner_args(jlink "--device=wbz451" "--speed=4000")  
  
# include(${ZEPHYR_BASE}/boards/common/jlink.board.cmake)  
include(${ZEPHYR_BASE}/boards/common/openocd.board.cmake)
```

Step-6: - Ensure the following **folders** are present under the “C:\developers\” directory:

- “Zephyr-sdk-0.17.0”
- “openocd_support_wbz_pic32wm”
- “Zephyrproject”
- “pkob4-cmsis_dap-switcher”

5. Steps to Flashing/Programming

5.1. WBZ451 Device

- Change directory to “**cd zephyr**” and build the application as shown below
 - Build Command: “**West build -p always -b wbz451_curiosity .\samples\basic\blinky**”

```
PRO Focus folder in explorer (ctrl + click) TERMINAL PORTS MEMORY ELF RIPPER
PS C:\developers\zephyrproject\zephyr> west build -p always -b wbz451_curiosity .\samples\basic\blinky\
```

- **Note:** Same build steps applicable for **WBZ450** and **WBZ451HPE** as below
 - “**West build -p always -b wbz450_curiosity .\samples\basic\blinky**”
 - “**West build -p always -b wbz451hpe_curiosity .\samples\basic\blinky**”
- After a successful build, connect the WBZ451 device and run the “**west flash**” command to program the device, as shown below:

```
-- Zephyr version: 4.0.0 (C:/developers/zephyrproject/zephyr), build: c1f783adc7cb
[117/117] Linking C executable zephyr\zephyr.elf
Memory region      Used Size Region Size %age Used
ROMSTART_REGION:    224 B     1 KB    21.88%
FLASH:           14616 B    1 MB    1.39%
RAM:             4224 B   128 KB    3.22%
IDT_LIST:          0 GB    32 KB    0.00%
Generating files from C:/developers/zephyrproject/zephyr/build/zephyr/zephyr.elf for board: wbz451_curiosity
PS C:\developers\zephyrproject\zephyr> west flash
```

- Ensure the flash process is completed successfully, as shown below:

TargetName	Type	Endian	TapName	State
0* wbz451.cpu	cortex_m	little	wbz451.cpu	unknown

```
[wbz451.cpu] halted due to breakpoint, current mode: Thread
xPSR: 0x01000000 pc: 0x01000730 msp: 0x20001040
Info : WBZ451: Write completed
Info : Padding image section 1 at 0x00045ea0 with 28 bytes
Info : Padding image section 2 at 0x00045ec0 with 28 bytes
Info : Padding image section 3 at 0x00045ee0 with 168 bytes
Info : Padding image section 4 at 0x00045fa0 with 28 bytes
Info : Padding image section 5 at 0x00045fc0 with 28 bytes
Warn : Adding extra erase range, 0x00045c00 .. 0x00045e87
Info : WBZ451: Write completed
Info : WBZ451: Write completed
auto erase enabled
wrote 16760 bytes from file C:/developers/zephyrproject/zephyr/build/zephyr/zephyr.hex in 1.391416s (11.763 KiB/s)
shutdown command invoked
PS C:\developers\zephyrproject\zephyr>
```

5.2. WBZ653 Device

- Change directory to “`cd zephyr`” and build the application as shown below
 - Build Command: “`West build -p always -b pic32wm_bz6204_curiosty .\samples\basic\blinky\`”

```
-- Build files have been written to: C:/developers/zephyr/build
-- west build: building application
[1/119] Generating include/generated/zephyr/version.h
-- Zephyr version: 4.0.0 (C:/developers/zephyr), build: 54fd5047f6f0
[117/119] Linking C executable zephyr\zephyr.elf
Memory region      Used Size Region Size %age Used
    FLASH:        16072 B   2096640 B   0.77%
      RAM:         4288 B     512 KB   0.82%
  IDT_LIST:          0 GB      32 KB   0.00%
Generating files from C:/developers/zephyr/build/zephyr/zephyr.elf for board: pic32wm_bz6204_curiosty
[119/119] Converting zephyr.hex to zephyr_signed.hex
PS C:\developers\zephyr> west build -p always -b pic32wm bz6204 curosity .\samples\basic\blinky
```

- After a successful build, connect the WBZ653 device and run the “`west flash --hex-file build/zephyr/zephyr_signed.hex`” command to program the device, as shown below:

```
Generating files from C:/developers/zephyr/build/zephyr/zephyr.elf for board: pic32wm_bz6204_curiosty
[119/119] Converting zephyr.hex to zephyr_signed.hex
PS C:\developers\zephyr> west flash --hex-file build/zephyr/zephyr_signed.hex
```

- Ensure the flash process is completed successfully, as shown below:

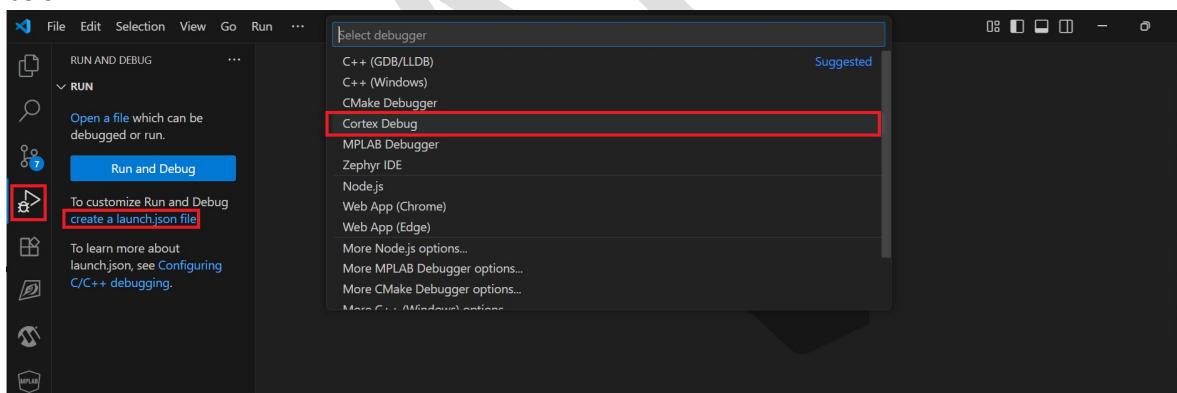
```
Info : Writing row to RAM buffer at 0x20000000
Info : Programming row at 0x01003c00
Info : Reading row at 0x01004000 for modify
Info : Writing row to RAM buffer at 0x20000000
Info : Programming row at 0x01004000
Info : WBZ653: Write complete
auto erase enabled
wrote 17784 bytes from file build/zephyr/zephyr_signed.hex in 1.814170s (9.573 KiB/s)
Info : SWD DPIDR 0x2ba01477
Error: Failed to write memory at 0xe000ed10
shutdown command invoked
PS C:\developers\zephyr>
```

6. Steps to Debugging the WBZ451, WBZ451HPE, WBZ450 & WBZ653 Device

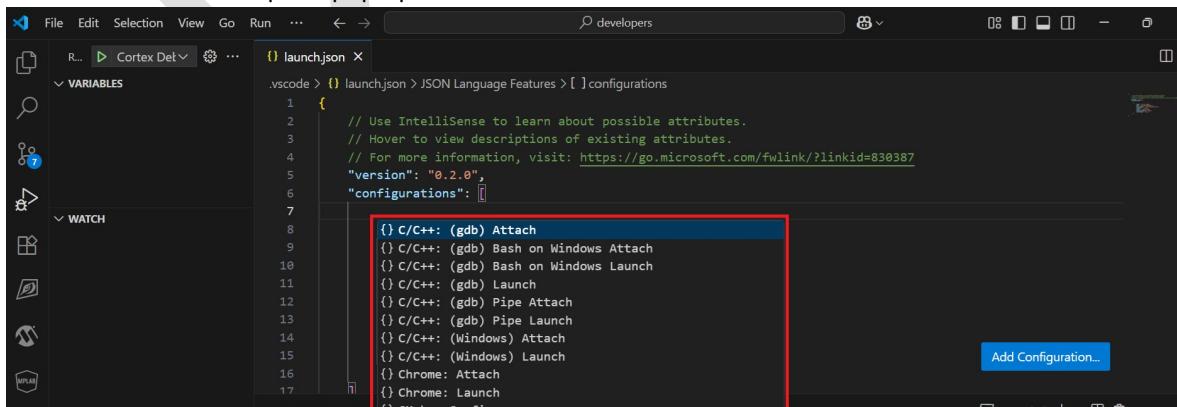
- Make sure you have built the application either for WBZ451 or WBZ451HPE or WBZ450 or WBZ653 before starting the debugging process.
- Make sure cortex-debug extension is installed in your local machine



- Click on the Run and Debug icon on the left. If the launch.json and tasks.json files are already present, VS Code will automatically start the debug session. Ensure that the contents of both files match the configurations provided in the steps below. If these files are not present, VS Code will prompt you to create a new launch.json file—select the "Cortex Debug" debugger option when prompted, as shown below.



- Press **Esc** to close or skip the pop-up window.



- Replace the content inside the launch.json file with the configuration shown below:

```
{  
    "version": "2.0.0",  
  
    "configurations": [  
        {  
            "name": "Debug PIC32WM",  
            "type": "cortex-debug",  
            "request": "attach",  
            "serverType": "openocd",  
            "cwd": "C:\\developers\\\\zephyr\\\\",  
            "executable": "C:/developers/zephyr/build/zephyr/zephyr.elf",  
            "device": "PIC32WM",  
            "configFiles": [  
                "interface/cmsis-dap.cfg",  
                "target/pic32wm.cfg"  
            ],  
            "gdbPath": "C:/developers/zephyr-sdk-0.17.0/arm-zephyr-eabi/bin/arm-zephyr-eabi-gdb.exe",  
            "preLaunchTask": "flash_pic32wm_signed_hex",  
            "postRestartCommands": [  
                "symbol-file C:/developers/zephyr/build/zephyr/zephyr.elf",  
                "monitor reset halt",  
                "break main"  
            ],  
            "showDevDebugOutput": "none"  
        }  
    ]  
}
```

```
},
{
  "name": "Debug WBZ451",
  "type": "cortex-debug",
  "request": "attach",
  "servertype": "openocd",
  "cwd": "C:\\developers\\zephyr\\",
  "executable": "C:/developers/zephyr/build/zephyr/zephyr.elf",
  "device": "WBZ451",
  "configFiles": [
    "interface/cmsis-dap.cfg",
    "target/wbz451.cfg"
  ],
  "gdbPath": "C:/developers/zephyr-sdk-0.17.0/arm-zephyr-eabi/bin/arm-zephyr-eabi-gdb.exe",
  "preLaunchTask": "flash_wbz451_hex",
  "postRestartCommands": [
    "symbol-file C:/developers/zephyr/build/zephyr/zephyr.elf",
    "monitor reset halt",
    "break main"
  ],
  "showDevDebugOutput": "none"
},
{
  "name": "Debug WBZ451HPE",
}
```

```
"type": "cortex-debug",

"request": "attach",

"servertype": "openocd",

"cwd": "C:\\developers\\zephyr\\",

"executable": "C:/developers/zephyr/build/zephyr/zephyr.elf",

"device": "WBZ451HPE",

"configFiles": [

    "interface/cmsis-dap.cfg",

    "target/wbz451h.cfg"

],



"gdbPath": "C:/developers/zephyr-sdk-0.17.0/arm-zephyr-eabi/bin/arm-zephyr-eabi-gdb.exe",

"preLaunchTask": "flash_wbz451hpe_hex",

"postRestartCommands": [

    "symbol-file C:/developers/zephyr/build/zephyr/zephyr.elf",

    "monitor reset halt",

    "break main"

],



"showDevDebugOutput": "none"

},



{

    "name": "Debug WBZ450",

    "type": "cortex-debug",

    "request": "attach",

    "servertype": "openocd",
```

```
"cwd": "C:\\developers\\zephyr\\",  
  
"executable": "C:/developers/zephyr/build/zephyr/zephyr.elf",  
  
"device": "WBZ450",  
  
"configFiles": [  
  
    "interface/cmsis-dap.cfg",  
  
    "target/wbz450.cfg"  
,  
  
    "gdbPath": "C:/developers/zephyr-sdk-0.17.0/arm-zephyr-eabi/bin/arm-zephyr-eabi-gdb.exe",  
  
    "preLaunchTask": "flash_wbz450_hex",  
  
    "postRestartCommands": [  
  
        "symbol-file C:/developers/zephyr/build/zephyr/zephyr.elf",  
  
        "monitor reset halt",  
  
        "break main"  
,  
  
        "showDevDebugOutput": "none"  
,  
  
    ]  
,  
  
}
```

```

0 launch.json x C main.c board.cmake ..\pic32wm_to6204_curiosity board.cmake ..\wbz\wbz451_curiosity board.cmake ..\wbz450_curiosity
vscode > 0 launch.json > Launch Targets > {} Debug WBZ450
1 {
2   "version": "2.0.0",
3   "configurations": [
4     {
5       "name": "Debug PIC32WM",
6       "type": "cortex-debug",
7       "request": "attach",
8       "servertype": "openocd",
9       "cwd": "C:\Developers\zephyr\\",
10      "executable": "C:/Developers/zephyr/build/zephyr/zephyr.elf",
11      "device": "PIC32MM"
12      "configFiles": [
13        "interface/cmsis-dap.cfg",
14        "target/pic32m.cfg"
15      ],
16      "gdbPath": "C:/Developers/zephyr-sdk-0.17.0/arm-zephyr-cabi/bin/arm-zephyr-cabi-gdb.exe",
17      "preLaunchTask": "flash_pic32m_signed_hex",
18      "postRestartCommands": [
19        "symbol-file C:/Developers/zephyr/build/zephyr/zephyr.elf",
20        "monitor reset halt",
21        "break main"
22      ],
23      "showDevDebugOutput": "none"
24    },
25    {
26      "name": "Debug WBZ451",
27      "type": "cortex-debug",
28      "request": "attach",
29      "servertype": "openocd",
30      "cwd": "C:\Developers\zephyr\\",
31      "executable": "C:/Developers/zephyr/build/zephyr/zephyr.elf",
32      "device": "WBZ451",
33      "configFiles": [
34        "interface/cmsis-dap.cfg",
35        "target/wbz451.cfg"
36      ],
37      "gdbPath": "C:/Developers/zephyr-sdk-0.17.0/arm-zephyr-cabi/bin/arm-zephyr-cabi-gdb.exe",
38      "preLaunchTask": "flash_wbz451_hex",
39      "postRestartCommands": [
40        "symbol-file C:/Developers/zephyr/build/zephyr/zephyr.elf",
41        "monitor reset halt",
42        "break main"
43      ],
44      "showDevDebugOutput": "none"
45    },
46    {
47      "name": "Debug WBZ451HPE",
48      "type": "cortex-debug",
49      "request": "attach",
50      "servertype": "openocd",
51      "cwd": "C:\Developers\zephyr\\",
52      "executable": "C:/Developers/zephyr/build/zephyr/zephyr.elf",
53      "device": "WBZ451HPE"
54      "configFiles": [
55        "interface/cmsis-dap.cfg",
56        "target/wbz451h.cfg"
57      ],
58      "gdbPath": "C:/Developers/zephyr-sdk-0.17.0/arm-zephyr-cabi/bin/arm-zephyr-cabi-gdb.exe",
59      "preLaunchTask": "flash_wbz451h_hex",
60      "postRestartCommands": [
61        "symbol-file C:/Developers/zephyr/build/zephyr/zephyr.elf",
62        "monitor reset halt",
63        "break main"
64      ],
65      "showDevDebugOutput": "none"
66    },
67    {
68      "name": "Debug WBZ458",
69      "type": "cortex-debug",
70      "request": "attach",
71      "servertype": "openocd",
72      "cwd": "C:\Developers\zephyr\\",
73      "executable": "C:/Developers/zephyr/build/zephyr/zephyr.elf",
74      "device": "WBZ458",
75      "configFiles": [
76        "interface/cmsis-dap.cfg",
77        "target/wbz458.cfg"
78      ],
79      "gdbPath": "C:/Developers/zephyr-sdk-0.17.0/arm-zephyr-cabi/bin/arm-zephyr-cabi-gdb.exe",
80      "preLaunchTask": "flash_wbz458_hex",
81      "postRestartCommands": [
82        "symbol-file C:/Developers/zephyr/build/zephyr/zephyr.elf",
83        "monitor reset halt",
84        "break main"
85      ],
86      "showDevDebugOutput": "none"
87    }
  ]
}

```

- Create a new **tasks.json** file inside the **.vscode** folder, and copy the following content into it:

```
{  
  "version": "2.0.0",  
  "tasks": [  
    {  
      "label": "flash_pic32wm_signed_hex",  
      "type": "shell",  
      "command": "openocd",  
      "args": [  
        "-f", "interface/cmsis-dap.cfg",  
        "-f", "target/pic32wm.cfg",  
        "-c", "init",  
        "-c", "reset halt",  
        "-c", "program C:/developers/zephyr/build/zephyr/zephyr_signed.hex reset exit"  
      ],  
      "problemMatcher": [],  
      "group": {  
        "kind": "build",  
        "isDefault": true  
      },  
    },  
    {  
      "label": "flash_wbz451_hex",  
    }  
  ]  
}
```

```
"type": "shell",
"command": "openocd",
"args": [
    "-f", "interface/cmsis-dap.cfg",
    "-f", "target/wbz451.cfg",
    "-c", "init",
    "-c", "reset halt",
    "-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
],
"problemMatcher": [],
"group": {
    "kind": "build",
    "isDefault": true
},
},
{
"label": "flash_wbz451hpe_hex",
"type": "shell",
"command": "openocd",
"args": [
    "-f", "interface/cmsis-dap.cfg",
    "-f", "target/wbz451h.cfg",
    "-c", "init",
    "-c", "reset halt",
]
```

```
"-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
],
"problemMatcher": [],
"group": {
  "kind": "build",
  "isDefault": true
},
{
  "label": "flash_wbz450_hex",
  "type": "shell",
  "command": "openocd",
  "args": [
    "-f", "interface/cmsis-dap.cfg",
    "-f", "target/wbz450.cfg",
    "-c", "init",
    "-c", "reset halt",
    "-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
  ],
  "problemMatcher": [],
  "group": {
    "kind": "build",
    "isDefault": true
  }
}
```

```
    },  
]  
}
```

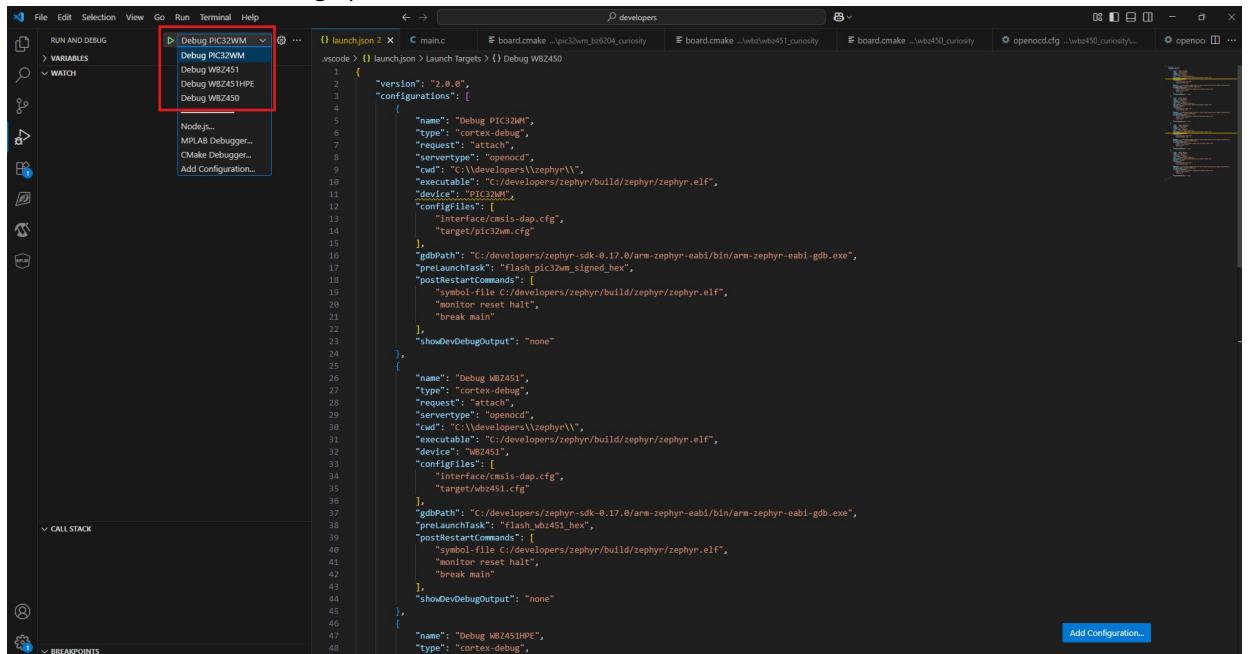
DRAFT

```

tasks.json | board.yml ..\wbz450.curiosity | board.cmake ..\wbz45x\.. | openocd.cfg ..\wbz45x\.. | board.cmake
.vscode > tasks.json > [ ]tasks > {} 0 > [ ]args
1  {
2    "version": "2.0.0",
3    "tasks": [
4      {
5        "label": "flash_pic32wm_signed_hex",
6        "type": "shell",
7        "command": "openocd",
8        "args": [
9          "-f", "interface/cmsis-dap.cfg",
10         "-f", "target/pic32wm.cfg",
11         "-c", "init",
12         "-c", "reset halt",
13         "-c", "program C:/developers/zephyr/build/zephyr/zephyr_signed.hex reset exit"
14       ],
15       "problemMatcher": [],
16       "group": {
17         "kind": "build",
18         "isDefault": true
19       },
20     },
21     {
22       "label": "flash_wbz451_hex",
23       "type": "shell",
24       "command": "openocd",
25       "args": [
26         "-f", "interface/cmsis-dap.cfg",
27         "-f", "target/wbz451.cfg",
28         "-c", "init",
29         "-c", "reset halt",
30         "-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
31       ],
32       "problemMatcher": [],
33       "group": {
34         "kind": "build",
35         "isDefault": true
36       },
37     },
38     {
39       "label": "flash_wbz451hpe_hex",
40       "type": "shell",
41       "command": "openocd",
42       "args": [
43         "-f", "interface/cmsis-dap.cfg",
44         "-f", "target/wbz451h.cfg",
45         "-c", "init",
46         "-c", "reset halt",
47         "-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
48       ],
49       "problemMatcher": [],
50       "group": {
51         "kind": "build",
52         "isDefault": true
53       },
54     },
55     {
56       "label": "flash_wbz450_hex",
57       "type": "shell",
58       "command": "openocd",
59       "args": [
60         "-f", "interface/cmsis-dap.cfg",
61         "-f", "target/wbz450.cfg",
62         "-c", "init",
63         "-c", "reset halt",
64         "-c", "program C:/developers/zephyr/build/zephyr/zephyr.hex reset exit"
65       ],
66       "problemMatcher": [],
67       "group": {
68         "kind": "build",
69         "isDefault": true
70       },
71     }
  ],
  "problemMatcher": []
}

```

- Connect the Required Curiosity board, then click the Run and Debug icon to start the debugging process. You will see a drop-down menu where you can choose between WBZ451, WBZ451HPE, WBZ450 and WBZ653 debug options.



- Make sure the debugger hits the breakpoint as shown below, then press the **Continue** button to proceed to the main.c application.

The screenshot shows the VS Code interface with the Zephyr device.h file open. A red box highlights the 'Breakpoints' icon in the top bar. The code editor shows a conditional breakpoint at line 57: `#endif compiler_barrier();`. The 'Breakpoints' panel indicates a hit count of 2. The 'CALL STACK' panel shows the application is paused on a breakpoint at address 0x010002d4. The bottom status bar shows the application is running on a WBZ451 Zephyr Application (developers).

- After pressing the **Continue** button, the application will load, allowing you to set breakpoints in the code and verify them as shown below.

The screenshot shows the VS Code interface with the main.c file open. A red box highlights the 'Breakpoints' icon in the top bar. The code editor shows a conditional breakpoint at line 39: `if (ret < 0) { | return 0; }`. The 'Breakpoints' panel indicates a hit count of 2. The 'CALL STACK' panel shows the application is paused on a breakpoint at address 0x0100030a. A terminal window titled 'COM101 - Tera Term VT' is open, displaying the LED state toggling between ON and OFF, which corresponds to the code execution.

7. References / Resources:

- TBD

8. Abbreviations:

Abbreviations	Full Form
OpenOCD	Open On-Chip Debugger
SWD	Serial Wire Debug
PKOB4	PICKit On-Board v4
CMSIS-DAP	Cortex Microcontroller Software Interface Standard - Debug Access Port
VSCode	Visual Studio Code
JSON	JavaScript Object Notation (used for configuration files)
PATH	Environment Variable for Executable File Locations

9. Document History:

Version	Date	Author	Description of Changes
v0.2	23 – OCct - 2025	Dinesh Arasu	Draft Version