Traveo II Entry Family Starter Kit – Getting Started

About this document

Scope and purpose

This document explains how to setup and use Traveo™ II Body Entry Family Starter Kit. The document also explains how to debug with single core and dual core environment in IAR Embedded Workbench for ARM (EWARM). The document uses the Traveo™ II Body Entry starter kit board with the Sample Driver Library (SDL).

Intended audience

This document is intended for software and hardware engineers integrating Traveo II Body Entry device into their application.

Table of contents

1 Getting started

This section explains the hardware set up. **Table 1** lists the pre-requisites for the set up.

Table 1 Pre-requisites

Quantity	Description	Remarks
1	CYTVII-B-E-1M-SK	Traveo II Body Entry Series Starter Kit
1	Micro USB cable	For power and communication
1	PC	With USB port
1	IAR EWARM 8.42.1	Downloaded from the web
1	Sample Driver Library v7.0.0	Downloaded from the web
-	Firmware	Pre-installed

1.1 Connection setup

Connect the USB cable from the PC to the Starter kit. The Starter kit is powered by the PC via the USB cable (5 V). Check if the mode LED (LED3) and the power LED (LED2) are turned ON.

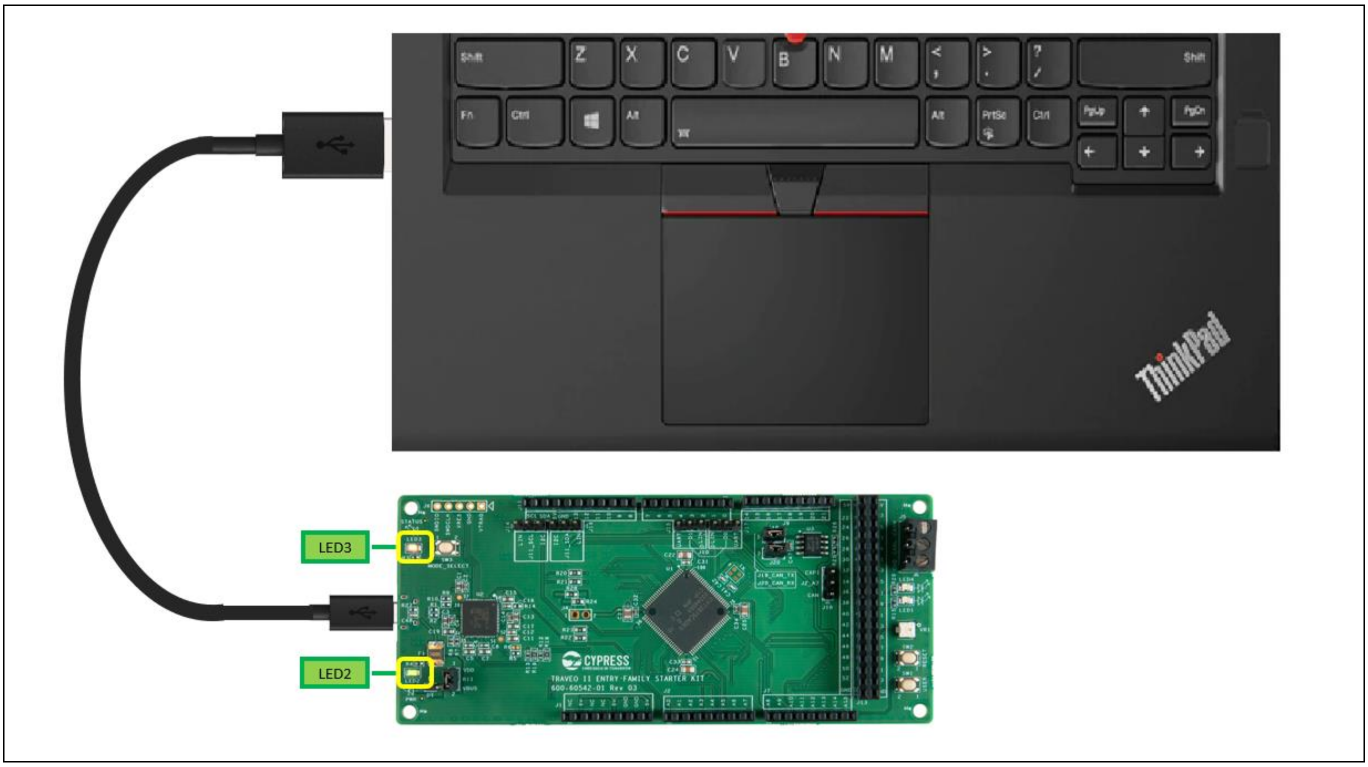


Figure 1 Connection between PC and starter kit

1.2 Power up

When powered ON, the device will start executing the pre-installed firmware, which is indicated by the blinking user LEDs (LED1 is controlled by core Arm® Cortex®-M0+ (CM0+) and LED4 is controlled by core Cortex-M4 (CM4)).

Note: To indicate that the starter kit is powered ON and the USB controller is starting in the correct mode, the amber colored status LED3 must be permanently ON. But, LED3 does not indicate a successful USB driver installation.

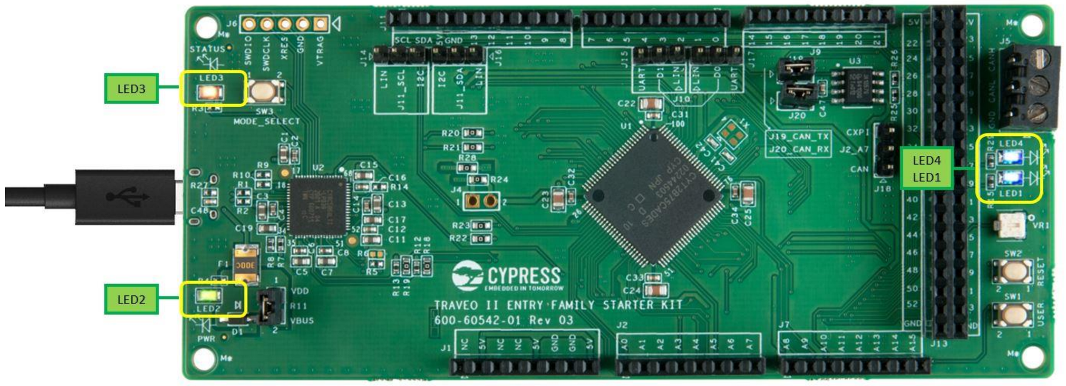
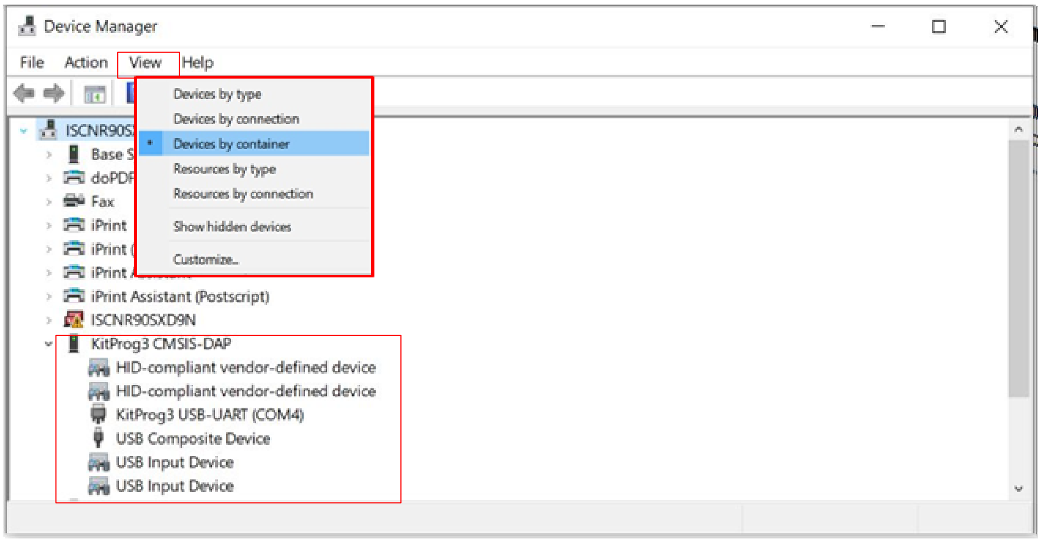


Figure 2 LED indicating power ON

1.3 Installing kit driver

To work with the Starter kit, the KitProg3 USB-UART driver must be installed on the system. For more details, see the Updating KitProg3 chapter in the KitProg3 user guide [1].

Confirm that the Starter kit is recognized as a KitProg3 device on Windows (open Device Manager, follow the menu path **View** > **Devices by container**) as shown in Figure 3. This completes the hardware setup.



Device Manager

File Action View Help

- Devices by type
- Devices by connection
- • Devices by container
- Resources by type
- Resources by connection
- Show hidden devices
- Customize...

- ISCNR90S
  - Base S
  - doPDF
  - Fax
  - iPrint
  - iPrint (
  - iPrint A
  - iPrint Assistant (Postscript)
  - ISCNR90SXD9N
  - KitProg3 CMSIS-DAP
    - HID-compliant vendor-defined device
    - HID-compliant vendor-defined device
    - KitProg3 USB-UART (COM4)
    - USB Composite Device
    - USB Input Device
    - USB Input Device

Figure 3 Viewing KitProg3 driver in Device Manager

Note: Do not press SW3. Pressing SW3 changes the mode of the USB controller. This is necessary only for upgrading the USB driver firmware and other activities. For more details, see the KitProg3 user guide [1].

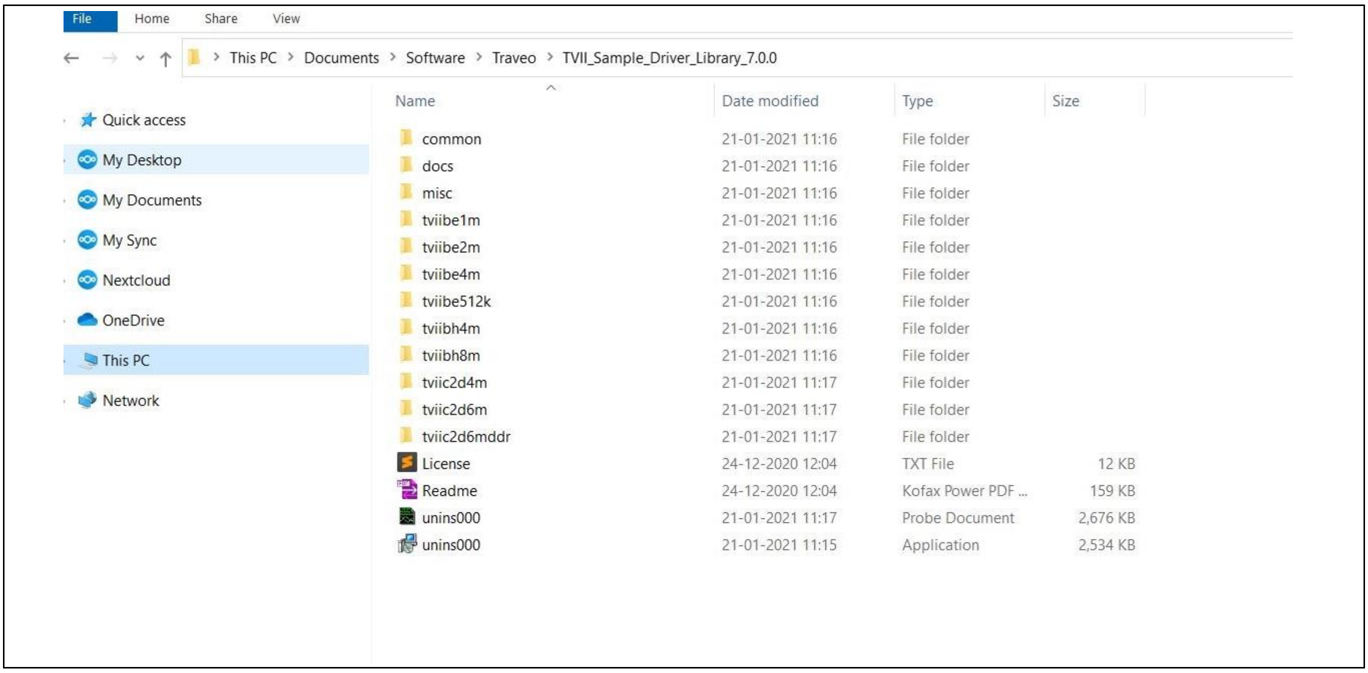
For more information related to the Starter kit, see the Traveo II Starter Kit User Guide [2].

2 SDL and IAR EWARM setup

This section explains how to run an example from the SDL on the Traveo II Starter Kit using the IAR C-SPY debugger.

2.1 SDL environment setup

Download the latest SDL for the Traveo II Starter Kit on the target system. Install the SDL outside the default Program Files to allow the IDE to access and create temporary files. **Figure 4** shows the sample path.



This PC > Documents > Software > Traveo > TVII_Sample_Driver_Library_7.0.0

Name	Date modified	Type	Size
common	21-01-2021 11:16	File folder	
docs	21-01-2021 11:16	File folder	
misc	21-01-2021 11:16	File folder	
tviibe1m	21-01-2021 11:16	File folder	
tviibe2m	21-01-2021 11:16	File folder	
tviibe4m	21-01-2021 11:16	File folder	
tviibe512k	21-01-2021 11:16	File folder	
tviibh4m	21-01-2021 11:16	File folder	
tviibh8m	21-01-2021 11:16	File folder	
tviic2d4m	21-01-2021 11:17	File folder	
tviic2d6m	21-01-2021 11:17	File folder	
tviic2d6mddr	21-01-2021 11:17	File folder	
License	24-12-2020 12:04	TXT File	12 KB
Readme	24-12-2020 12:04	Kofax Power PDF ...	159 KB
unins000	21-01-2021 11:17	Probe Document	2,676 KB
unins000	21-01-2021 11:15	Application	2,534 KB

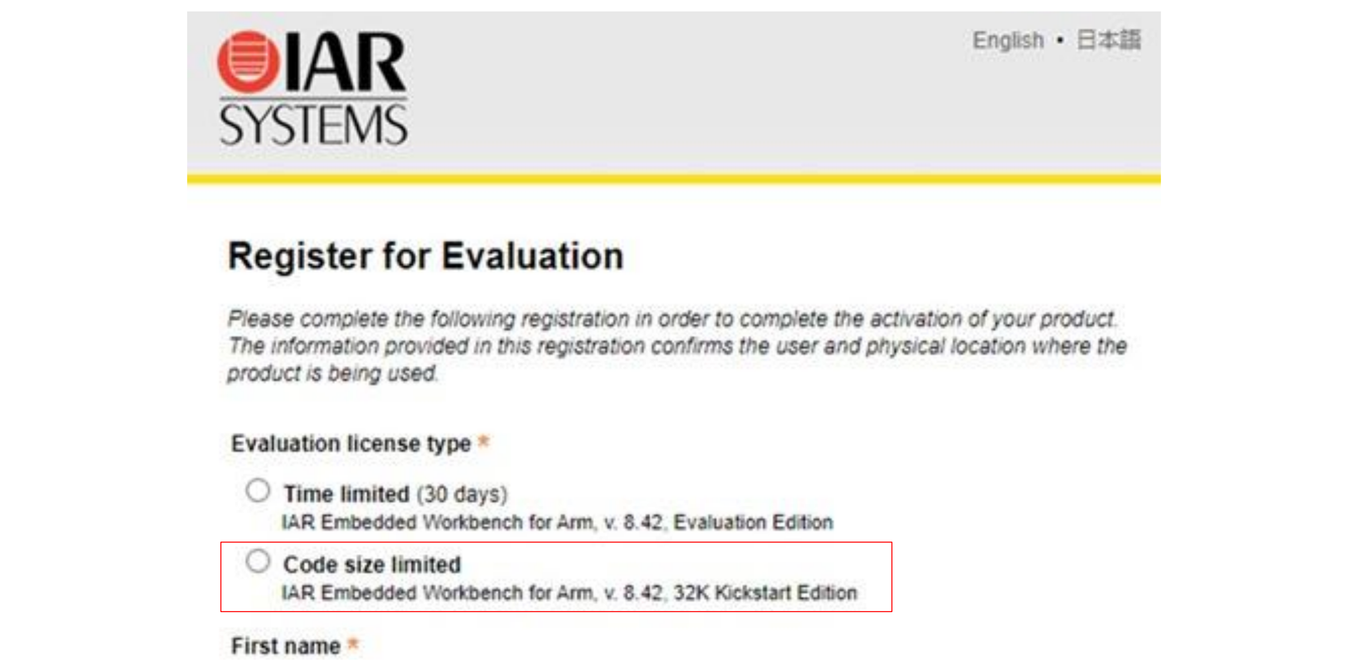
Figure 4 SDL 7.0.0 top-level directory

2.2 IAR EWARM setup

Open the *Readme.pdf* from the root folder of the SDL and use the same link to download the version of IAR EWARM supported by the SDL.

Download the software and run the installation (note that the installation might take some time). When you open the IAR EWARM for the first time, select the license in the License wizard. If you do not have the license, it is strongly recommended to register for **Code size limited** license type. See **Figure 5**.

Note: *IAR EWARM v8.42.1 is only for reference; always use the IAR EWARM version supported by the specific SDL release.*



English • 日本語

Register for Evaluation

Please complete the following registration in order to complete the activation of your product. The information provided in this registration confirms the user and physical location where the product is being used.

Evaluation license type *

○ Time limited (30 days)
IAR Embedded Workbench for Arm, v. 8.42, Evaluation Edition

○ Code size limited
IAR Embedded Workbench for Arm, v. 8.42, 32K Kickstart Edition

First name *

Figure 5 Registering IAR for evaluation (based on v. 8.42)

2.3 Connection setup

There are two debugging methods with IAR EWARM:

1. Debugging with code downloaded on to RAM
2. Debugging with code downloaded on to the FLASH memory (described in this document)

SDL supports three types of workspaces under FLASH memory debugging as listed in **Table 2**.

Table 2 IAR Flash workspaces and functions

Workspace	Application core	Number of cores supported	Details
Single core CM0+ workspace “tviibe1m_flash_cm0plus_template”	No	Single	Single core download and debug for CM0+ core.
Single core CM4 workspace “tviibe1m_flash_cm4_mc_template”	Yes	Single	Single core download and debug for CM4 core; works as client/slave workspace in multicore configuration. *Note:* *It is possible to debug only with CM4, when CM0+ executes some code, thus enabling CM4.*
Dual core CM0+ and CM4 workspace “tviibe1m_flash_cm0plus_cm4_template”	Yes	Dual	Dual core download and debug for CM0+ and CM4 cores; works

Workspace	Application core	Number of cores supported	Details
			as master workspace in multicore configuration.

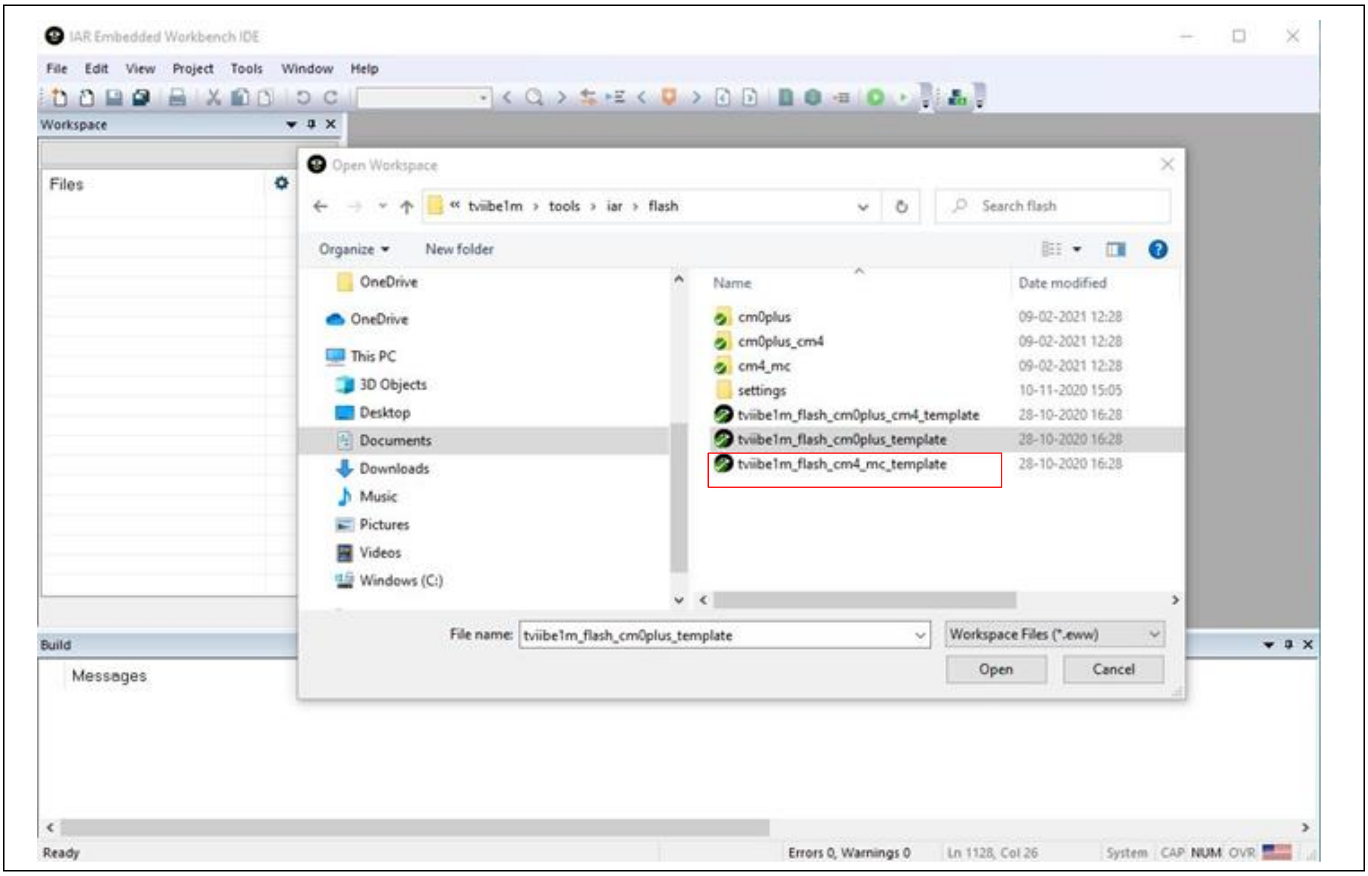
Note: Both CM0+ and CM4 can do normal code execution, but from an architectural point only CM4 is considered as the application core. After a reset, the default core is always the CM0+ core. To enable the CM4 core, CM0+ must call `Cy_SysEnableApplCore ()`. In the SDL, this is usually done within main_cm0plus.c.

Before downloading and debugging with the multicore master project, you must build the multicore slave project for the application core (which is CM4 here).

Open the master workspace (dual core CM0+ and CM4 workspaces) for multicore debugging. The slave workspace (single core CM4 workspace) will automatically open from the master workspace when you click **Download and Debug**.

2.3.1 Debugging with single core CM0+ workspace

1. Start IAR EWARM and open the SDL template workspace file: *tviibe1m\tools\iar\flash\tviibe1m_flash_cm0plus_template*



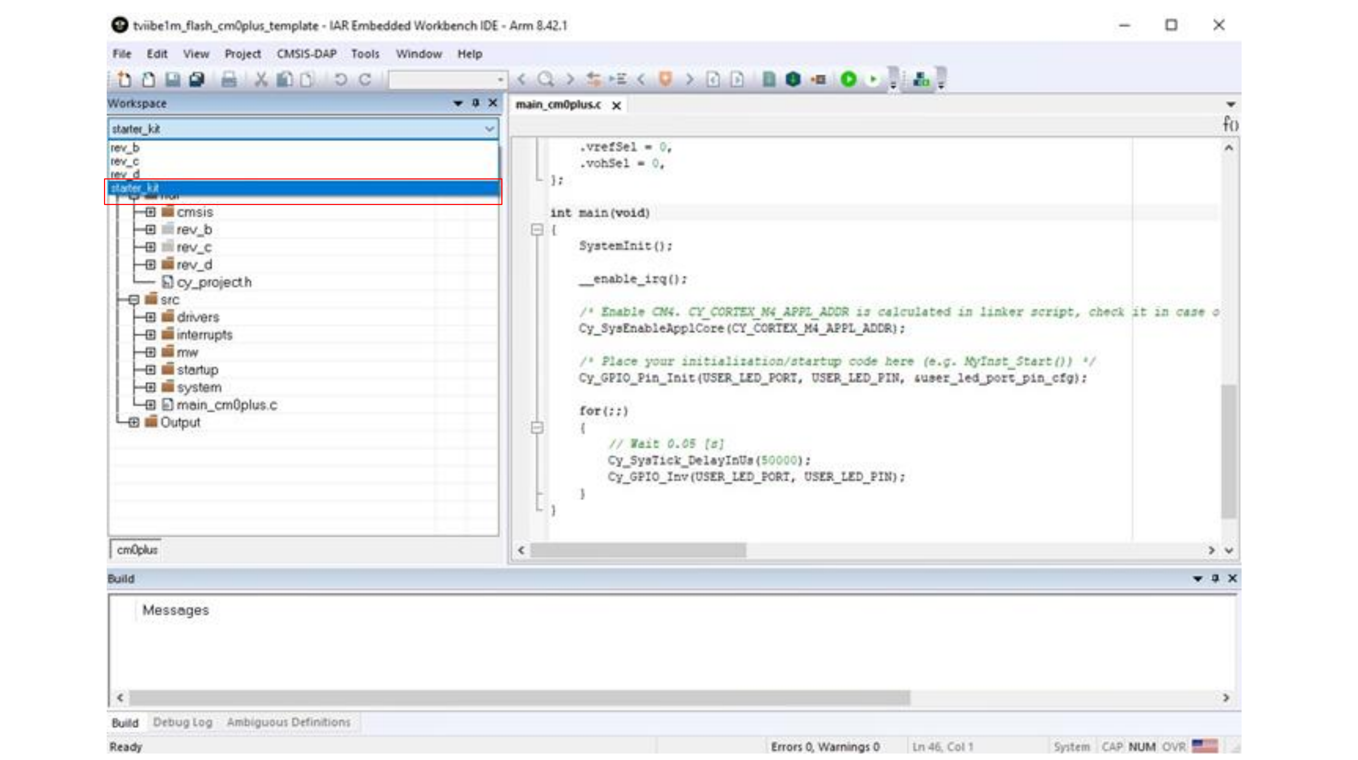
Name	Date modified
cm0plus	09-02-2021 12:28
cm0plus_cm4	09-02-2021 12:28
cm4_mc	09-02-2021 12:28
settings	10-11-2020 15:05
tviibe1m_flash_cm0plus_cm4_template	28-10-2020 16:28
tviibe1m_flash_cm0plus_template	28-10-2020 16:28
tviibe1m_flash_cm4_mc_template	28-10-2020 16:28

File name: tviibe1m_flash_cm0plus_template

Figure 6 Selecting CM0+ single core template

2. Select the workspace revision **starter_kit** from drop-down list under **Workspace**, as shown in **Figure 7**.

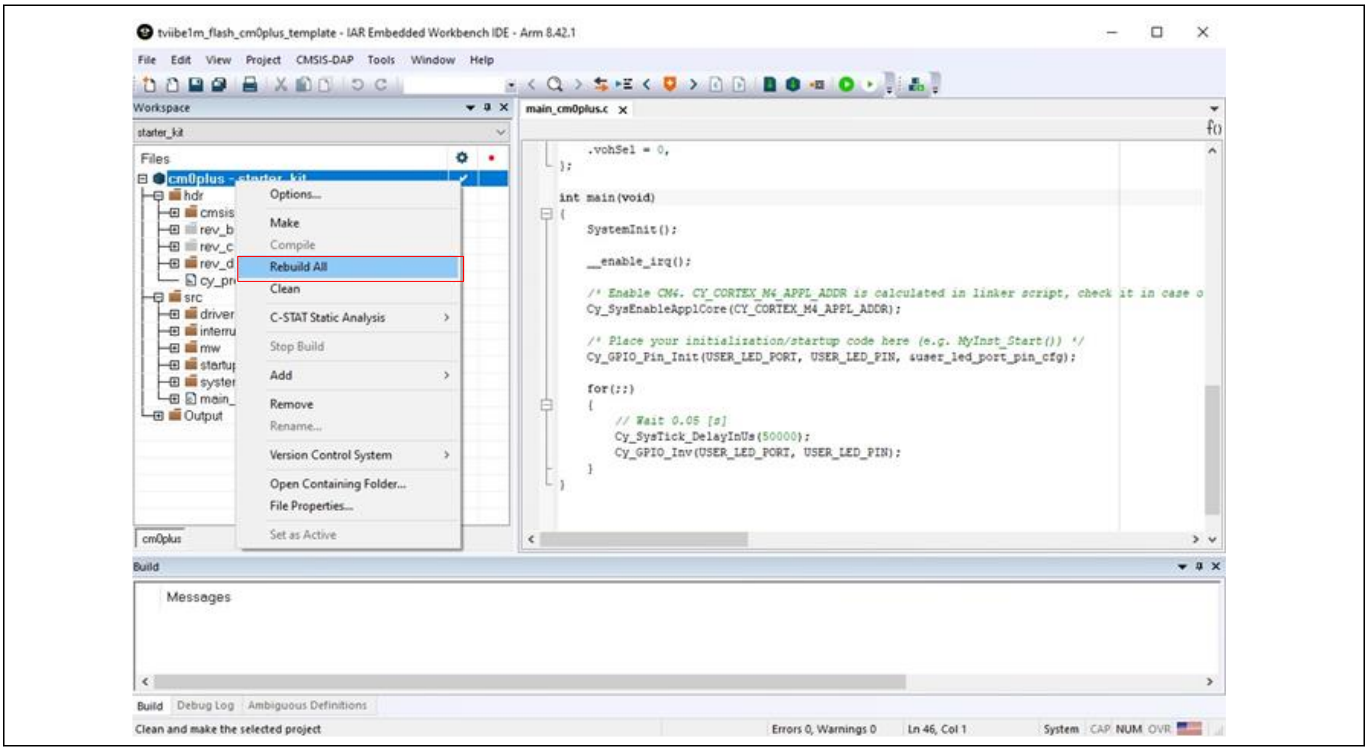
Note: *Ignore other workspace revisions for this Starter kit. The other revisions constitute the MCU assembled on other evaluation boards (CPU board).*



```
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case o
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Figure 7 Selecting workspace revision

3. For the build, right-click the cm0plus – starter_kit and select **Rebuild All**.

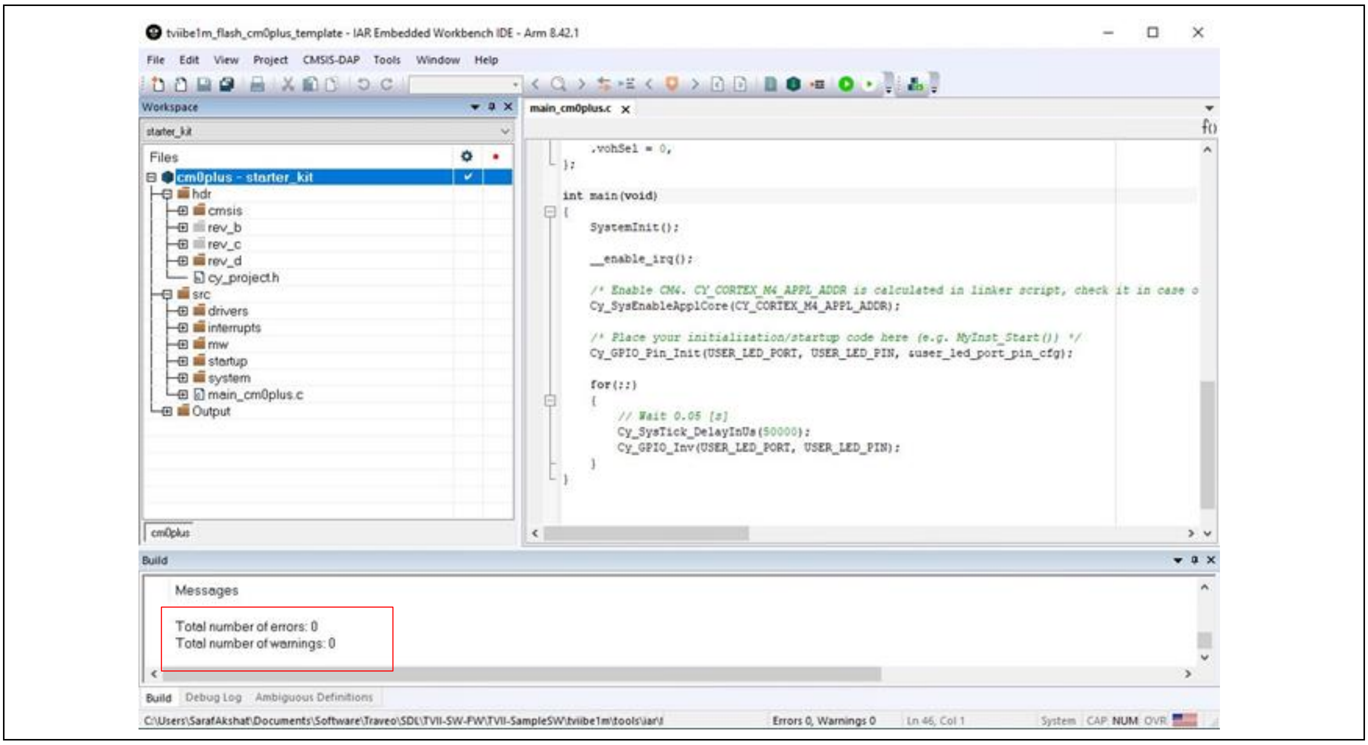


```
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case o
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Clean and make the selected project

Figure 8 Rebuilding workspace revision

4. The rebuild process starts. Check for errors and warnings in the **Build log**.


tviibe1m_flash_cm0plus_template - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
startup
system
main_cm0plus.c
Output
cm0plus
main_cm0plus.c
Build
Messages
Build Debug Log Ambiguous Definitions
Errors 0, Warnings 0 Ln 46, Col 1 System CAP NUM OVR

```
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case o
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

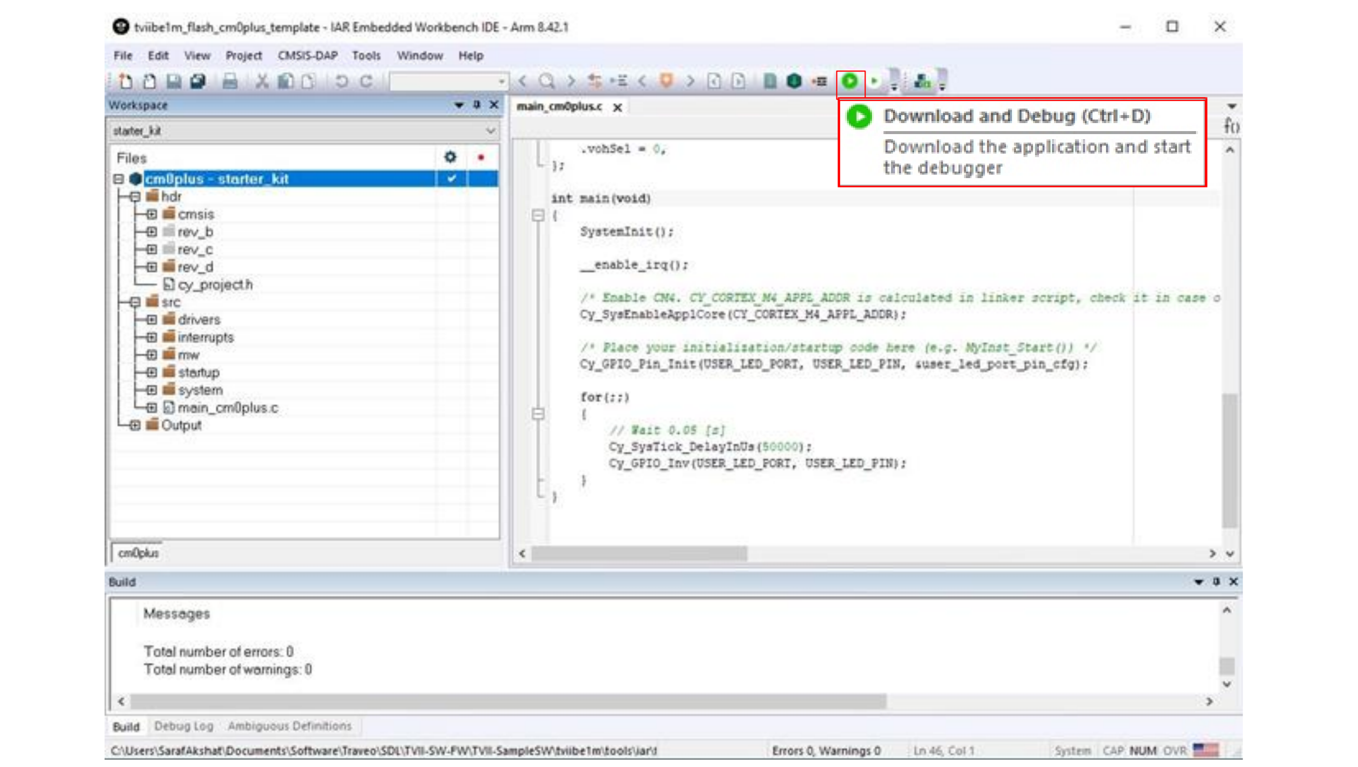
```
Total number of errors: 0
Total number of warnings: 0
```

C:\Users\SarafAkshat\Documents\Software\Traveo\SDL\TVII-SW-FW\TVII-SampleSW\tviibe1m\tools\iar\

Figure 9 Checking build status in build log

5. To load the program to the FLASH region of CM0+ core, click the green **Download and Debug** icon.

SDL and IAR EWARM setup


tviibe1m_flash_cm0plus_template - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
startup
system
main_cm0plus.c
Output
cm0plus
main_cm0plus.c
Download and Debug (Ctrl+D)
Download the application and start the debugger
Build
Build Debug Log Ambiguous Definitions

```
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case o
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

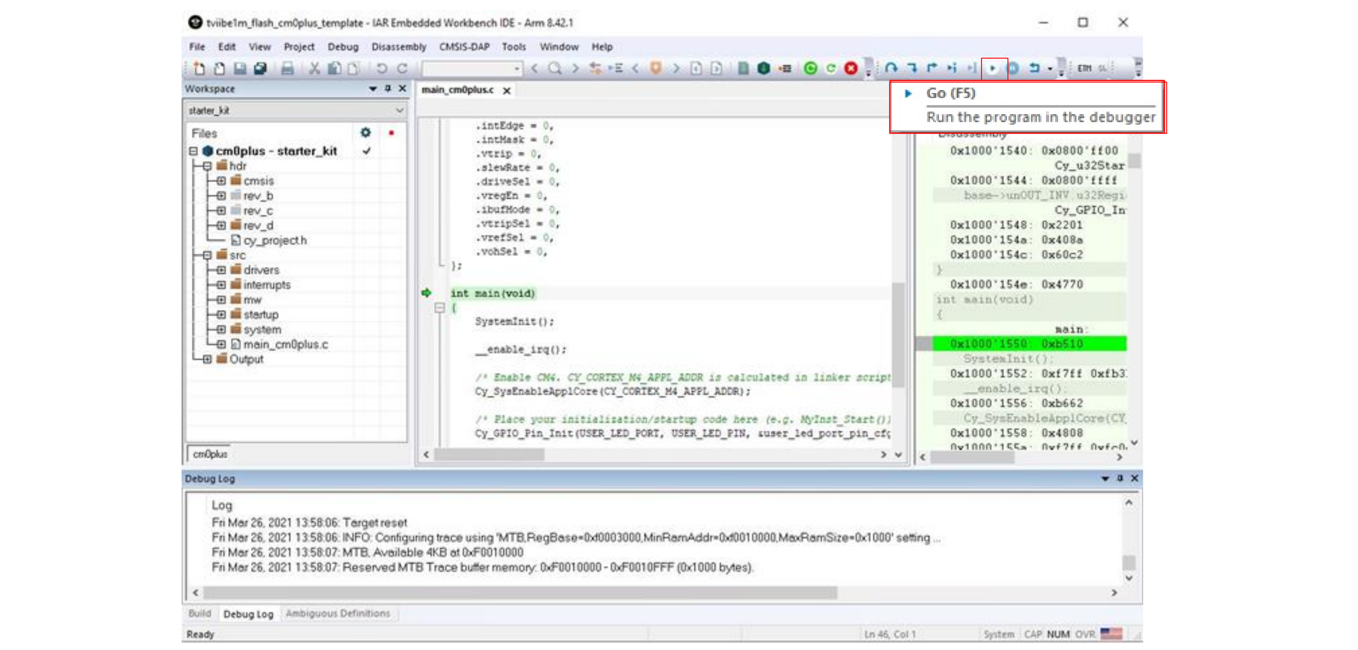
```
Messages

Total number of errors: 0
Total number of warnings: 0
```

```
C:\Users\SarafAkshat\Documents\Software\Traveo\SDL\TVII-SW-FW\TVII-SampleSW\tviibe1m\tools\iar\d    Errors 0, Warnings 0    Ln 46, Col 1    System  CAP NUM OVR
```

Figure 10 Downloading and debugging

6. Now, click the **Go** icon to start execution.

Note: You can also use the function keys in the Debug window: Go (F5), F10 (Step Over), F11 (Step into), Ctrl+D (Download and Debug).


tviibe1m_flash_cm0plus_template - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project Debug Disassembly CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
startup
system
main_cm0plus.c
Output
cm0plus
main_cm0plus.c
Go (F5)
Run the program in the debugger
Debug Log
Build Debug Log Ambiguous Definitions

```
    .intEdge = 0,
    .intMask = 0,
    .vtrip = 0,
    .slewRate = 0,
    .driveSel = 0,
    .vregEn = 0,
    .ibufMode = 0,
    .vtripSel = 0,
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start())
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cf
```

```
  0x1000'1540: 0x0800'ff00
                 Cy_u32Star
  0x1000'1544: 0x0800'ffff
    base->unOUT_INV.u32Regi
                 Cy_GPIO_In
  0x1000'1548: 0x2201
  0x1000'154a: 0x408a
  0x1000'154c: 0x60c2
}
  0x1000'154e: 0x4770
int main(void)
{
                       main:
  0x1000'1550: 0xb510
    SystemInit();
  0x1000'1552: 0xf7ff 0xfb3
    __enable_irq();
  0x1000'1556: 0xb662
    Cy_SysEnableApplCore(CY
  0x1000'1558: 0x4808
  0x1000'155a: 0xf7ff 0xfc0
```

```
Log
Fri Mar 26, 2021 13:58:06: Target reset
Fri Mar 26, 2021 13:58:06: INFO: Configuring trace using 'MTB,RegBase=0xf0003000,MinRamAddr=0xf0010000,MaxRamSize=0x1000' setting ...
Fri Mar 26, 2021 13:58:07: MTB, Available 4KB at 0xF0010000
Fri Mar 26, 2021 13:58:07: Reserved MTB Trace buffer memory: 0xF0010000 - 0xF0010FFF (0x1000 bytes).
```

```
Ready    Ln 46, Col 1    System  CAP NUM OVR
```

Figure 11 Debugging with CM0+ core

LED1 should start blinking.

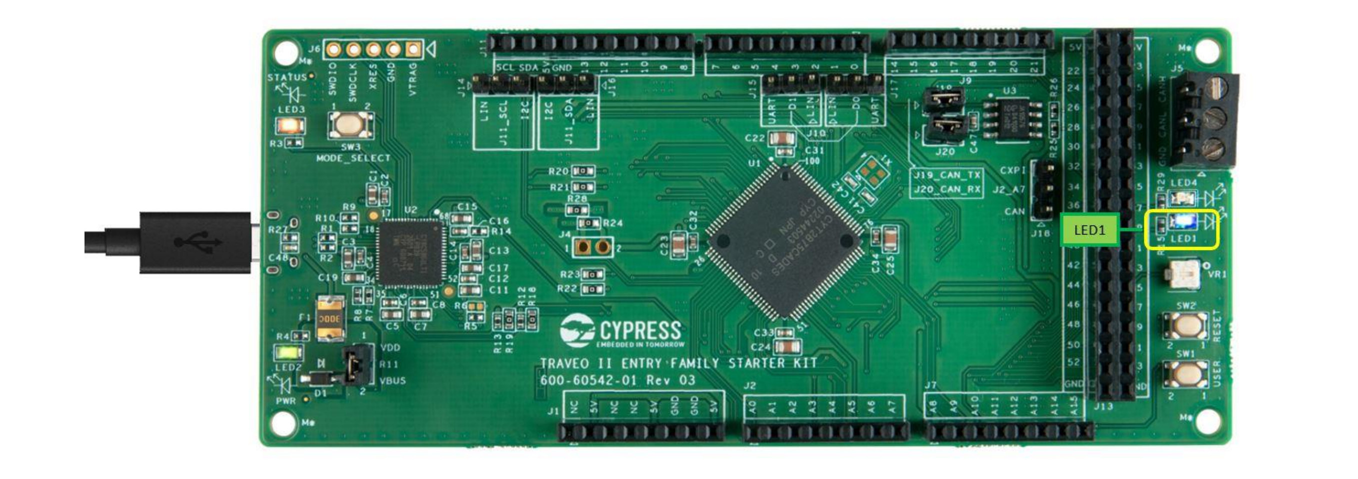
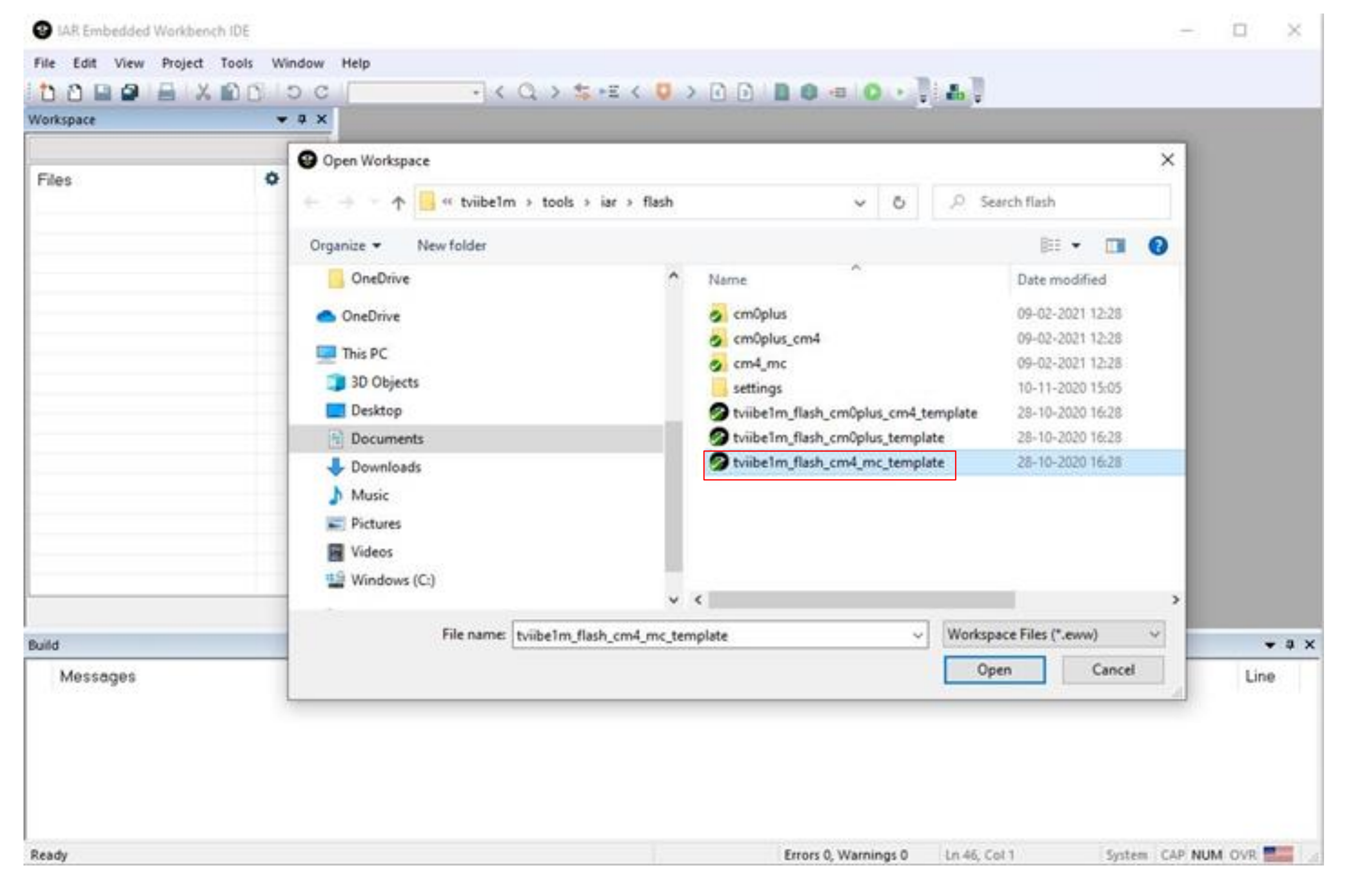


Figure 12 Blinking LED1

2.3.2 Debugging with single core CM4 workspace

1. Start IAR EWARM and open the SDL template workspace file:
   tviibe1m\tools\iar\flash\tviibe1m_flash_cm4_mc_template



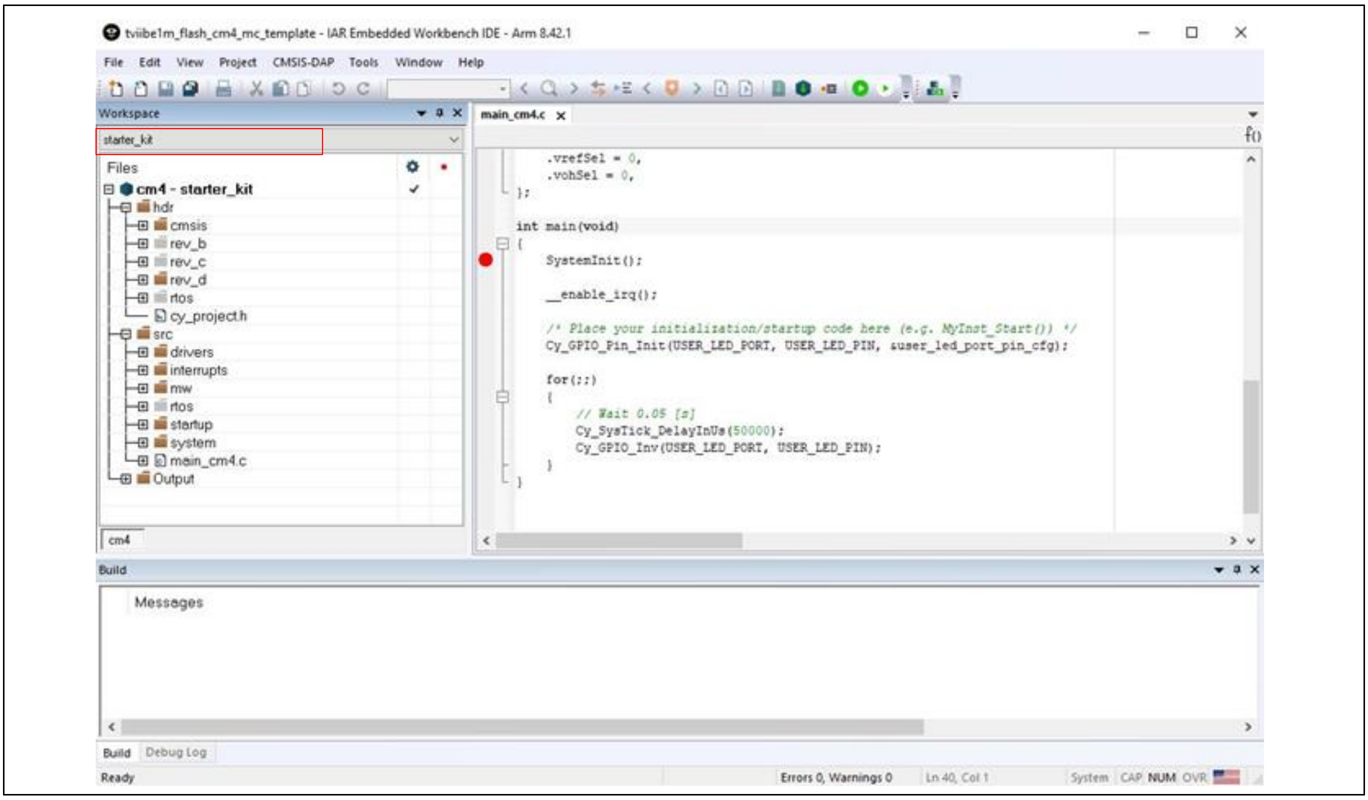
Name	Date modified
cm0plus	09-02-2021 12:28
cm0plus_cm4	09-02-2021 12:28
cm4_mc	09-02-2021 12:28
settings	10-11-2020 15:05
tviibe1m_flash_cm0plus_cm4_template	28-10-2020 16:28
tviibe1m_flash_cm0plus_template	28-10-2020 16:28
tviibe1m_flash_cm4_mc_template	28-10-2020 16:28

File name: tviibe1m_flash_cm4_mc_template

Workspace Files (*.eww)

Figure 13 Selecting CM4 single core template

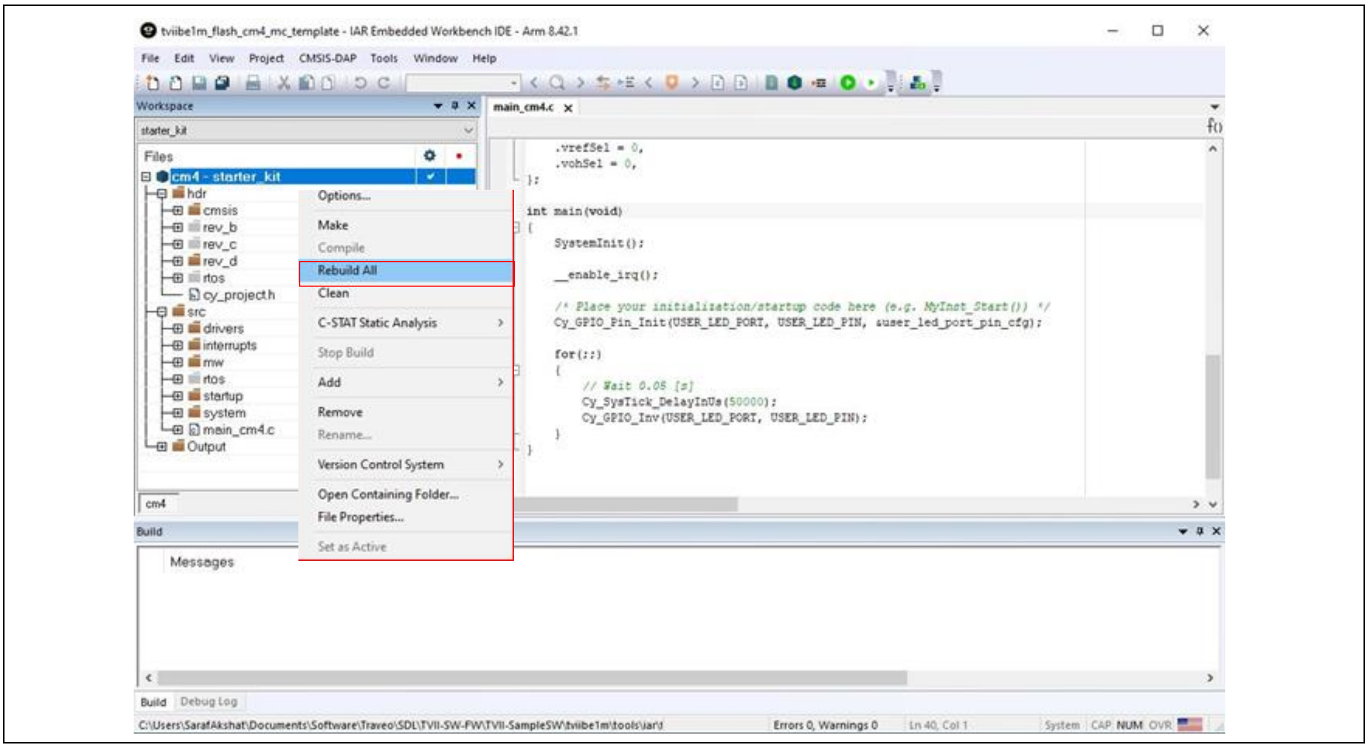
2. Select the workspace revision **starter_kit** from drop-down list under **Workspace**, as shown in **Figure 14**.



```
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Figure 14 Selecting workspace revision

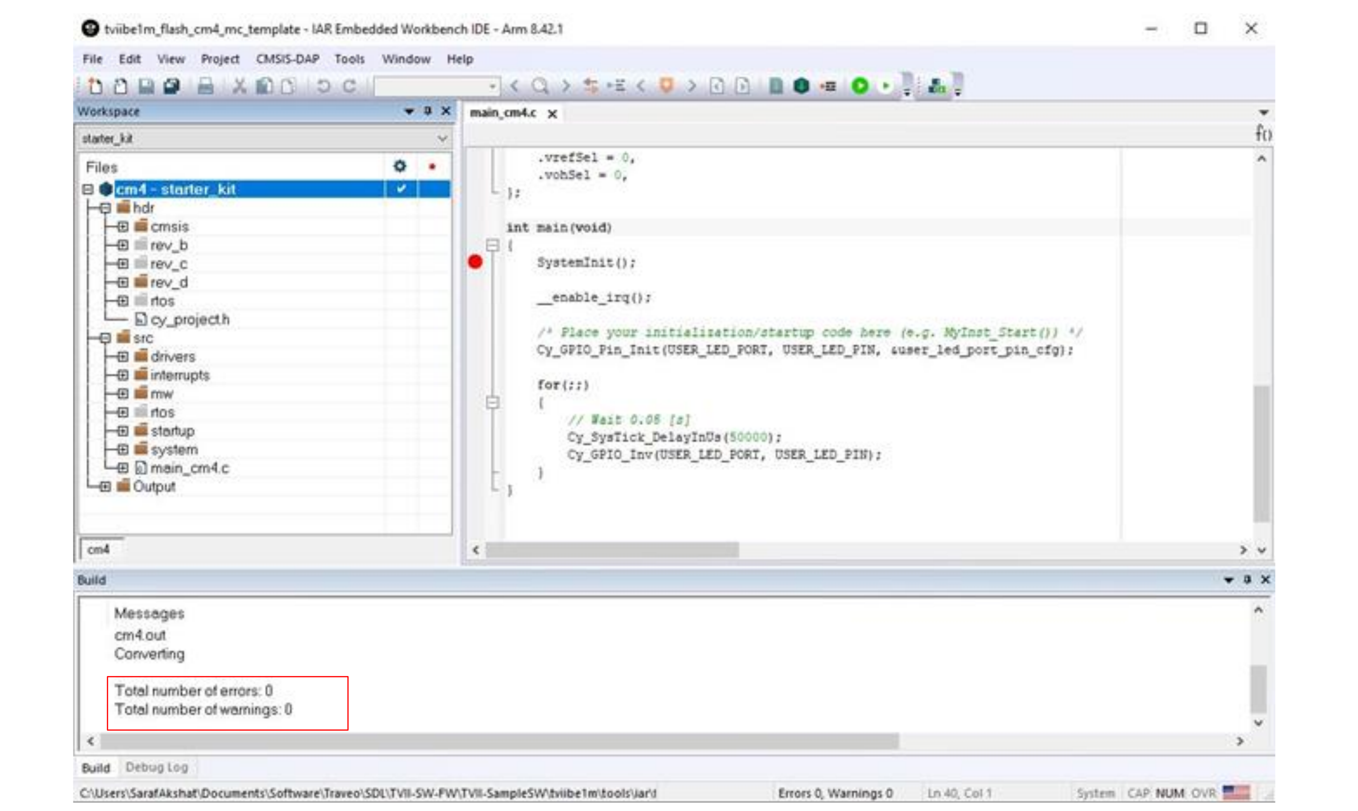
3. For the build, right-click the cm4 – starter_kit and select **Rebuild All**.



```
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Figure 15 Rebuilding workspace revision

4. The rebuild process starts. Check for errors and warnings in the **Build log**.

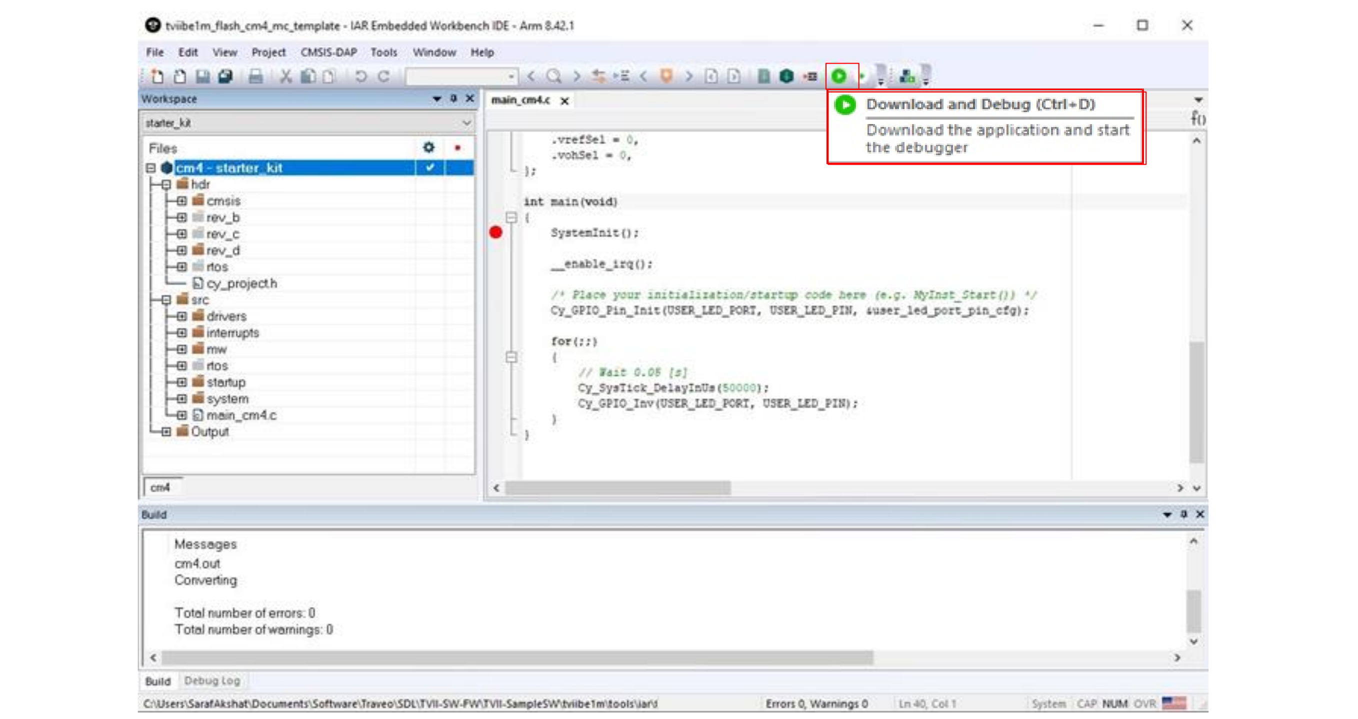


```
        .vrefSel = 0,
        .vohSel = 0,
    };

    int main(void)
    {
        SystemInit();

        __enable_irq();

        /* Place your initialization/startup code here (e.g. MyInst_Start()) */
        Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

        for(;;)
        {
            // Wait 0.05 [s]
            Cy_SysTick_DelayInUs(50000);
            Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
        }
    }
```

```
Messages
cm4.out
Converting

Total number of errors: 0
Total number of warnings: 0
```

Figure 16 Checking build status in build log

5. To load the program to the FLASH region of CM0+ core, click the green **Download and Debug** icon.

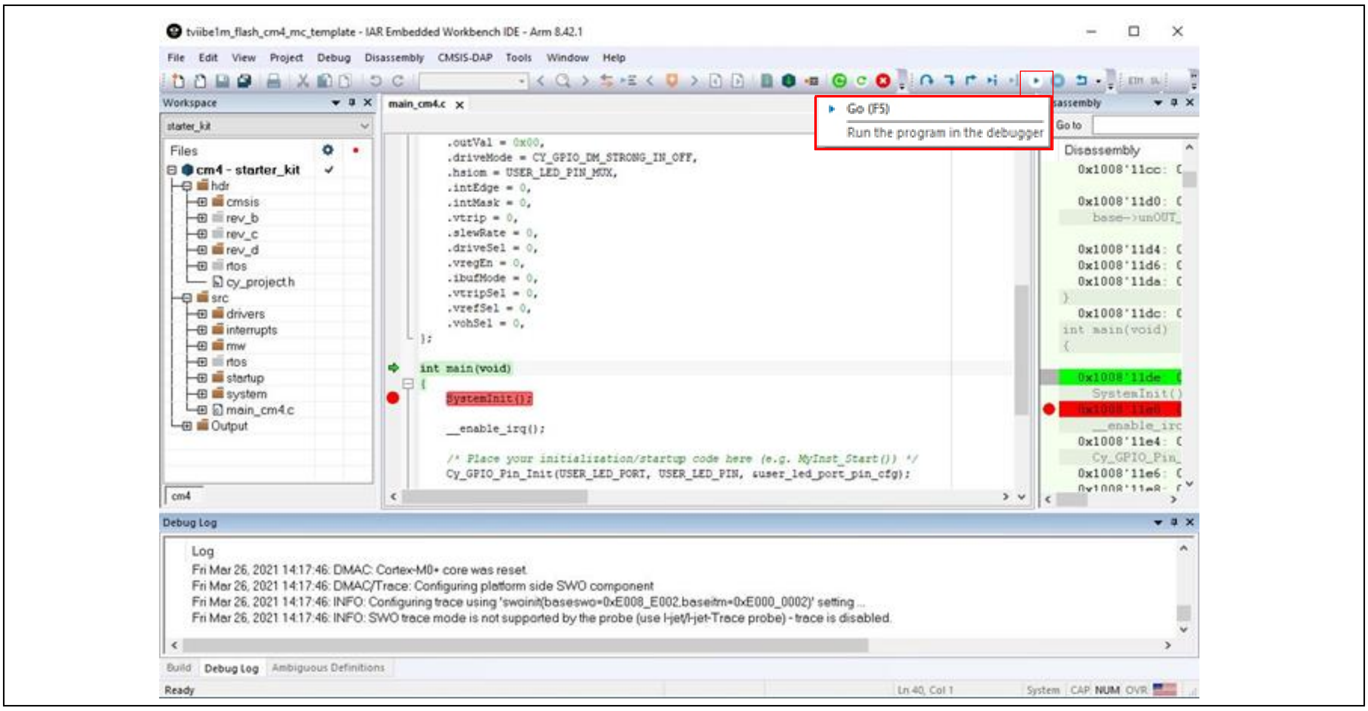

tviibe1m_flash_cm4_mc_template - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project CMSIS-DAP Tools Window Help
Download and Debug (Ctrl+D)
Download the application and start the debugger
Workspace
starter_kit
Files
cm4 - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
rtos
cy_project.h
src
drivers
interrupts
mw
rtos
startup
system
main_cm4.c
Output
cm4
main_cm4.c
Build
Build Debug Log
C:\Users\SarafAkshat\Documents\Software\Traveo\SDL\TVII-SW-FW\TVII-SampleSW\tviibe1m\tools\iar\
Errors 0, Warnings 0
Ln 40, Col 1
System CAP NUM OVR

```
        .vrefSel = 0,
        .vohSel = 0,
    };

    int main(void)
    {
        SystemInit();

        __enable_irq();

        /* Place your initialization/startup code here (e.g. MyInst_Start()) */
        Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

        for(;;)
        {
            // Wait 0.05 [s]
            Cy_SysTick_DelayInUs(50000);
            Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
        }
    }
```

```
Messages
cm4.out
Converting

Total number of errors: 0
Total number of warnings: 0
```

Figure 17 Downloading and debugging

6. Now, click the **Go** icon to start execution.



```
        .outVal = 0x00,
        .driveMode = CY_GPIO_DM_STRONG_IN_OFF,
        .hsiom = USER_LED_PIN_MUX,
        .intEdge = 0,
        .intMask = 0,
        .vtrip = 0,
        .slewRate = 0,
        .driveSel = 0,
        .vregEn = 0,
        .ibufMode = 0,
        .vtripSel = 0,
        .vrefSel = 0,
        .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);
```

```
Log
Fri Mar 26, 2021 14:17:46: DMAC: Cortex-M0+ core was reset.
Fri Mar 26, 2021 14:17:46: DMAC/Trace: Configuring platform side SWO component
Fri Mar 26, 2021 14:17:46: INFO: Configuring trace using 'swoinit(baseswo=0xE008_E002,baseitm=0xE000_0002)' setting ...
Fri Mar 26, 2021 14:17:46: INFO: SWO trace mode is not supported by the probe (use I-jet/I-jet-Trace probe) - trace is disabled.
```

Figure 18 Debugging with CM4 core

LED4 should start blinking.

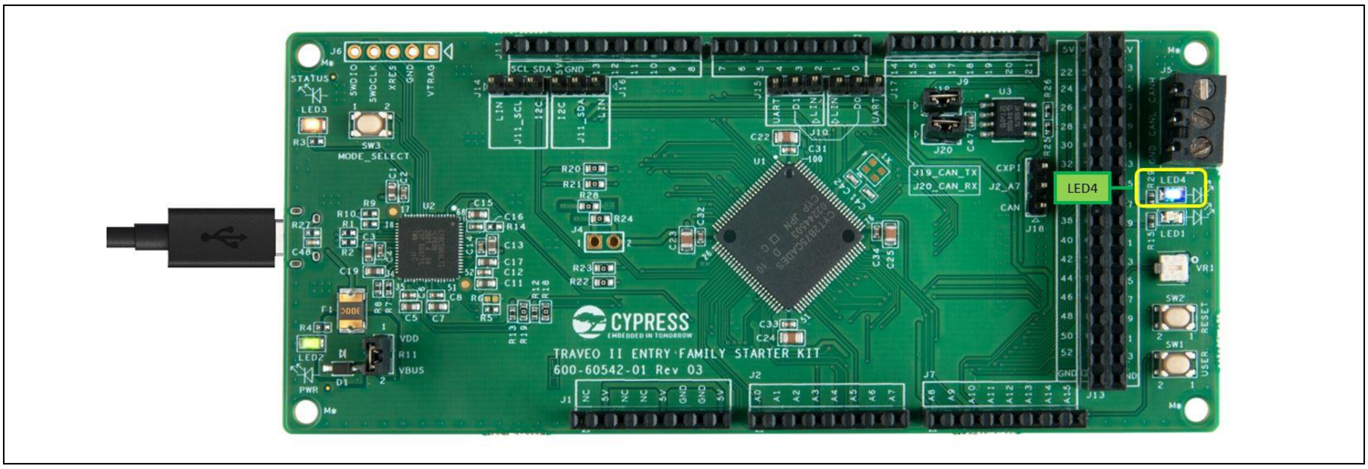


Figure 19 Blinking LED4

2.3.3 Debugging with dual core CM0+ and CM4 workspaces

1. Start IAR EWARM and open the SDL template workspace file: *tviibe1m\tools\iar\flash\tviibe1m_flash_cm0plus_cm4 _template*

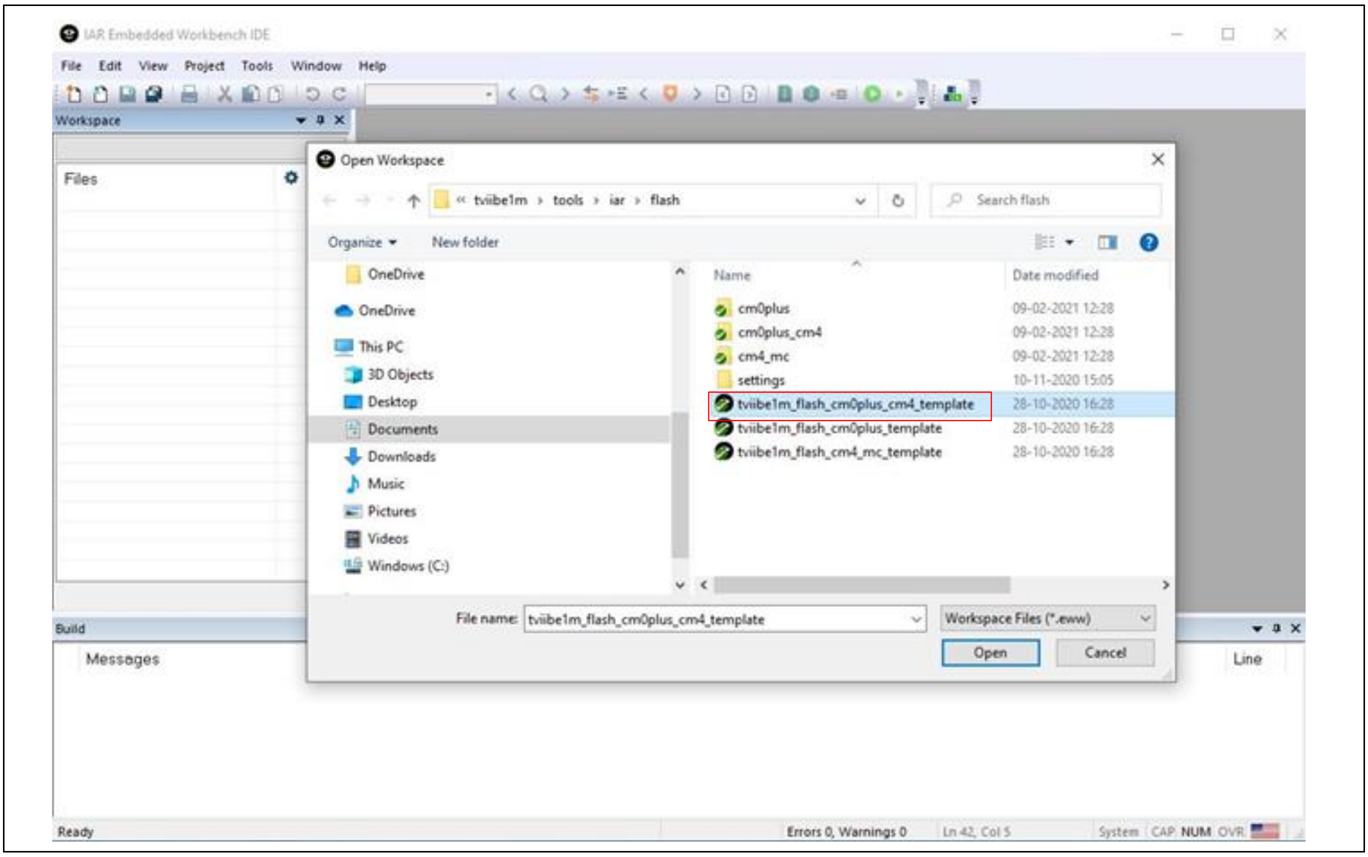
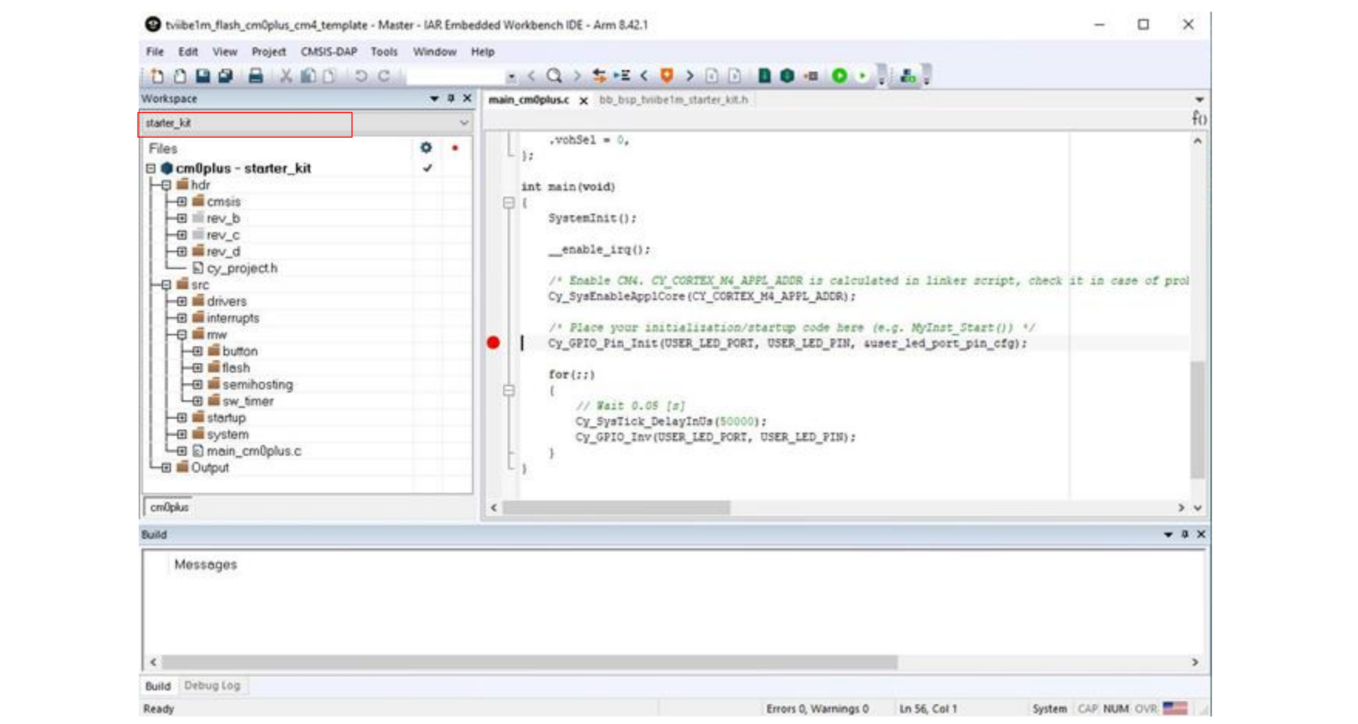

IAR Embedded Workbench IDE
File Edit View Project Tools Window Help
Workspace
Files
Open Workspace
« tviibe1m › tools › iar › flash
Search flash
Organize
New folder
OneDrive
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Name
Date modified
cm0plus 09-02-2021 12:28
cm0plus_cm4 09-02-2021 12:28
cm4_mc 09-02-2021 12:28
settings 10-11-2020 15:05
tviibe1m_flash_cm0plus_cm4_template 28-10-2020 16:28
tviibe1m_flash_cm0plus_template 28-10-2020 16:28
tviibe1m_flash_cm4_mc_template 28-10-2020 16:28
File name: tviibe1m_flash_cm0plus_cm4_template
Workspace Files (*.eww)
Open
Cancel
Build
Messages
Line
Ready
Errors 0, Warnings 0
Ln 42, Col 5
System CAP NUM OVR

Figure 20 Selecting CM0+ and CM4 dual core template

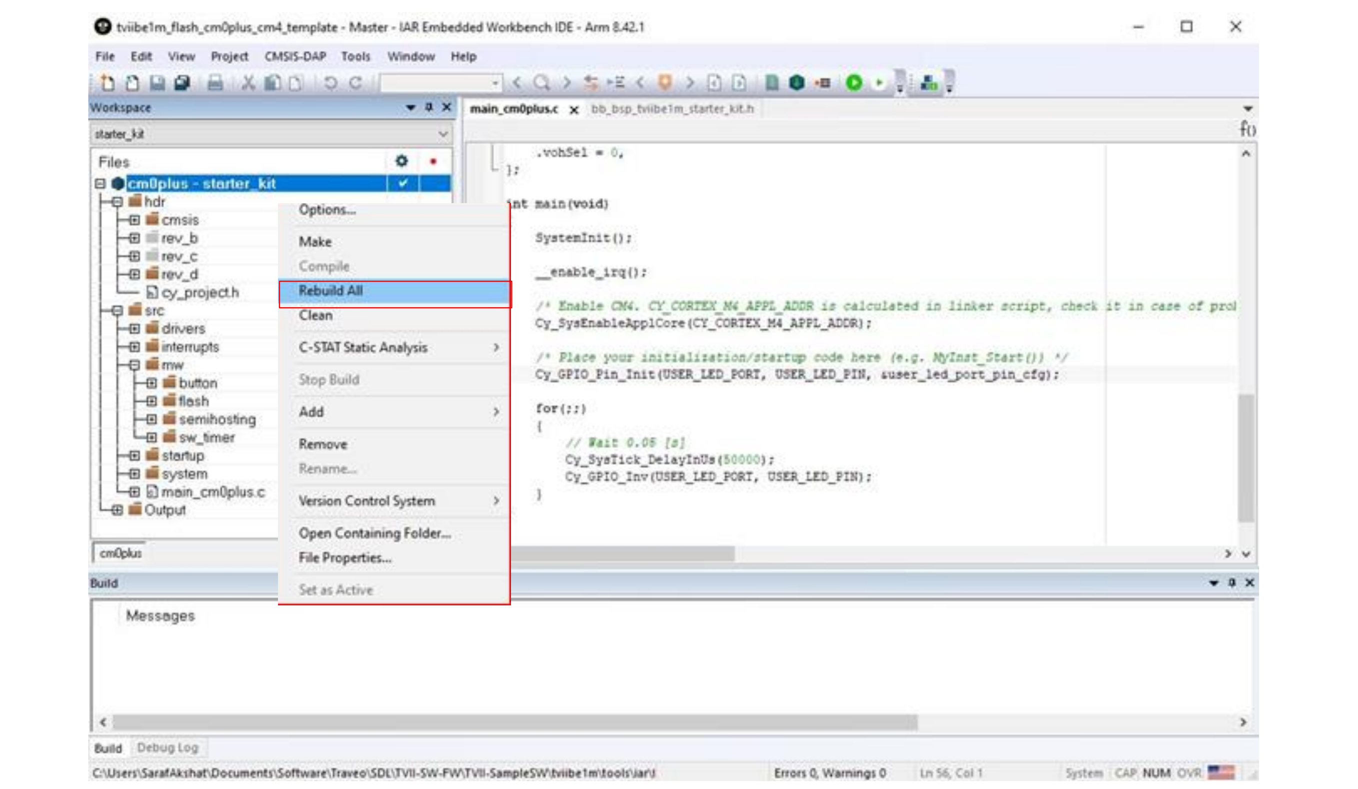
2. Select the workspace revision **starter_kit** from drop-down list under **Workspace**, as shown in **Figure 7**.



```
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of prol
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Figure 21 Selecting workspace revision

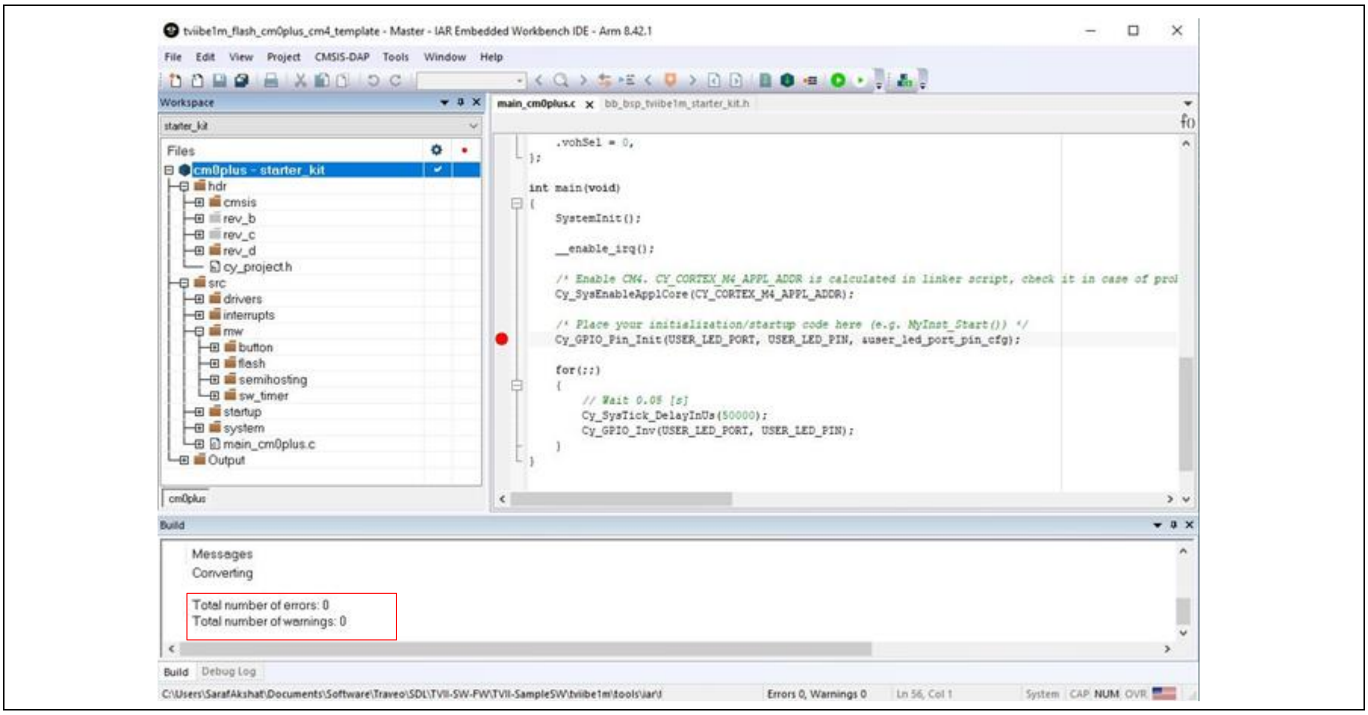
3. For the build, right-click the cm0plus – starter_kit and select **Rebuild All**.



```
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of prol
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

Figure 22 Rebuilding workspace revision

4. The rebuild process starts. Check for errors and warnings in the **Build log**.

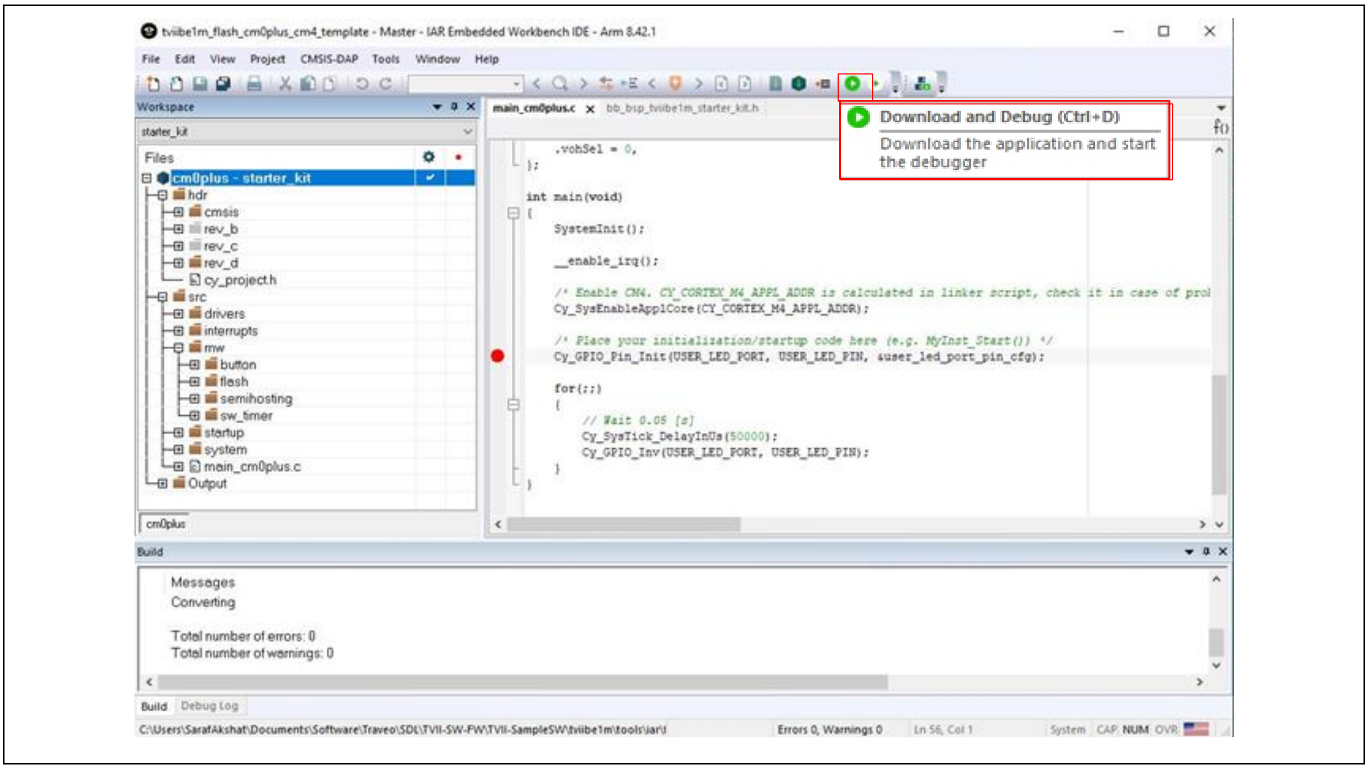

tviibe1m_flash_cm0plus_cm4_template - Master - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
button
flash
semihosting
sw_timer
startup
system
main_cm0plus.c
Output
cm0plus
main_cm0plus.c
bb_bsp_tviibe1m_starter_kit.h
Build
Build Debug Log
C:\Users\SarafAkshat\Documents\Software\Traveo\SDL\TVII-SW-FW\TVII-SampleSW\tviibe1m\tools\iar\
Errors 0, Warnings 0
Ln 56, Col 1
System CAP NUM OVR

```
        .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of prol
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

```
Messages
Converting

Total number of errors: 0
Total number of warnings: 0
```

Figure 23 Checking build status in build log

5. To load the program to the FLASH region of CM0+ and CM4 cores, click the green **Download and Debug** icon. This will automatically open the CM4 multicore client workspace, and the corresponding code will be downloaded into the FLASH region of respective cores.


tviibe1m_flash_cm0plus_cm4_template - Master - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter_kit
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
button
flash
semihosting
sw_timer
startup
system
main_cm0plus.c
Output
cm0plus
main_cm0plus.c
bb_bsp_tviibe1m_starter_kit.h
Build
Build Debug Log
C:\Users\SarafAkshat\Documents\Software\Traveo\SDL\TVII-SW-FW\TVII-SampleSW\tviibe1m\tools\iar\
Errors 0, Warnings 0
Ln 56, Col 1
System CAP NUM OVR

Download and Debug (Ctrl+D)

Download the application and start the debugger

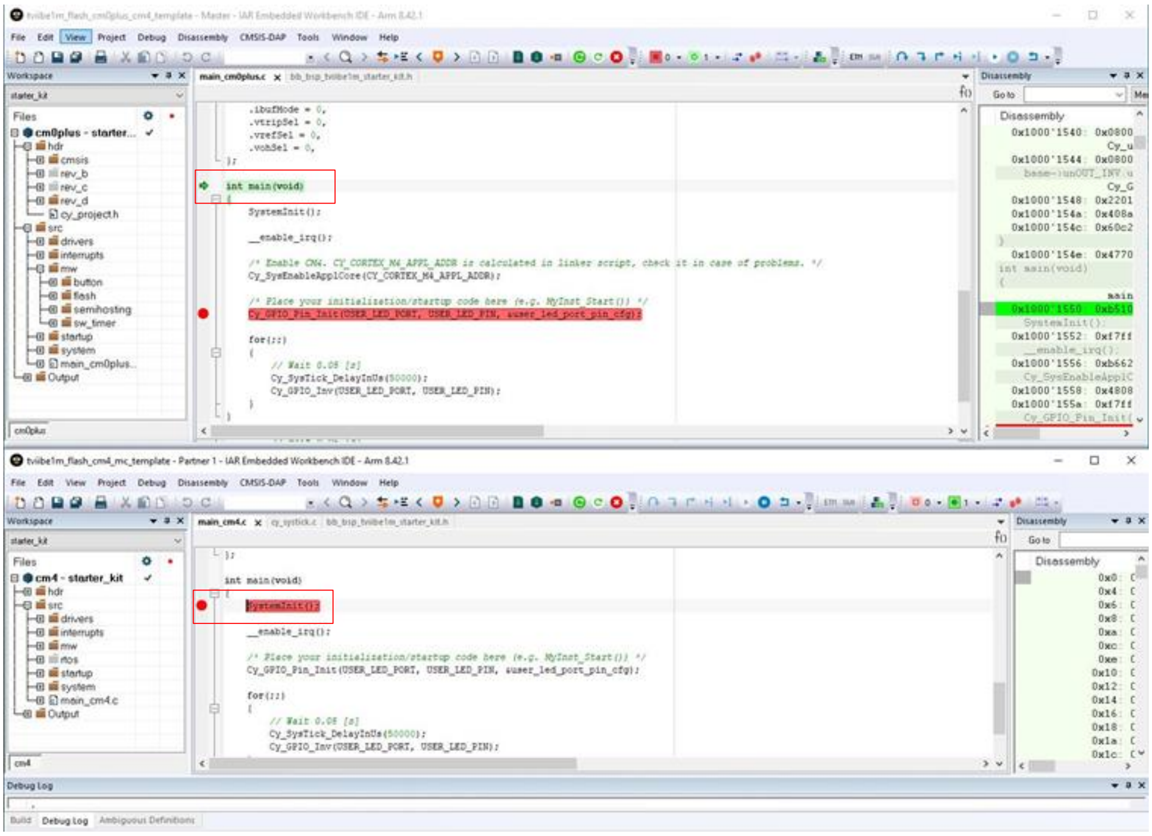
```
        .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of prol
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

```
Messages
Converting

Total number of errors: 0
Total number of warnings: 0
```

Figure 24 Downloading and debugging

6. Place a break point in the `SystemInit()` API in the CM4 core workspace. CM4 core will start executing after being enabled by CM0+ core. You can debug both cores simultaneously.

Note: To place a breakpoint at the target instruction, click the white space between the editor window (left pane) and the workspace window (center pane).

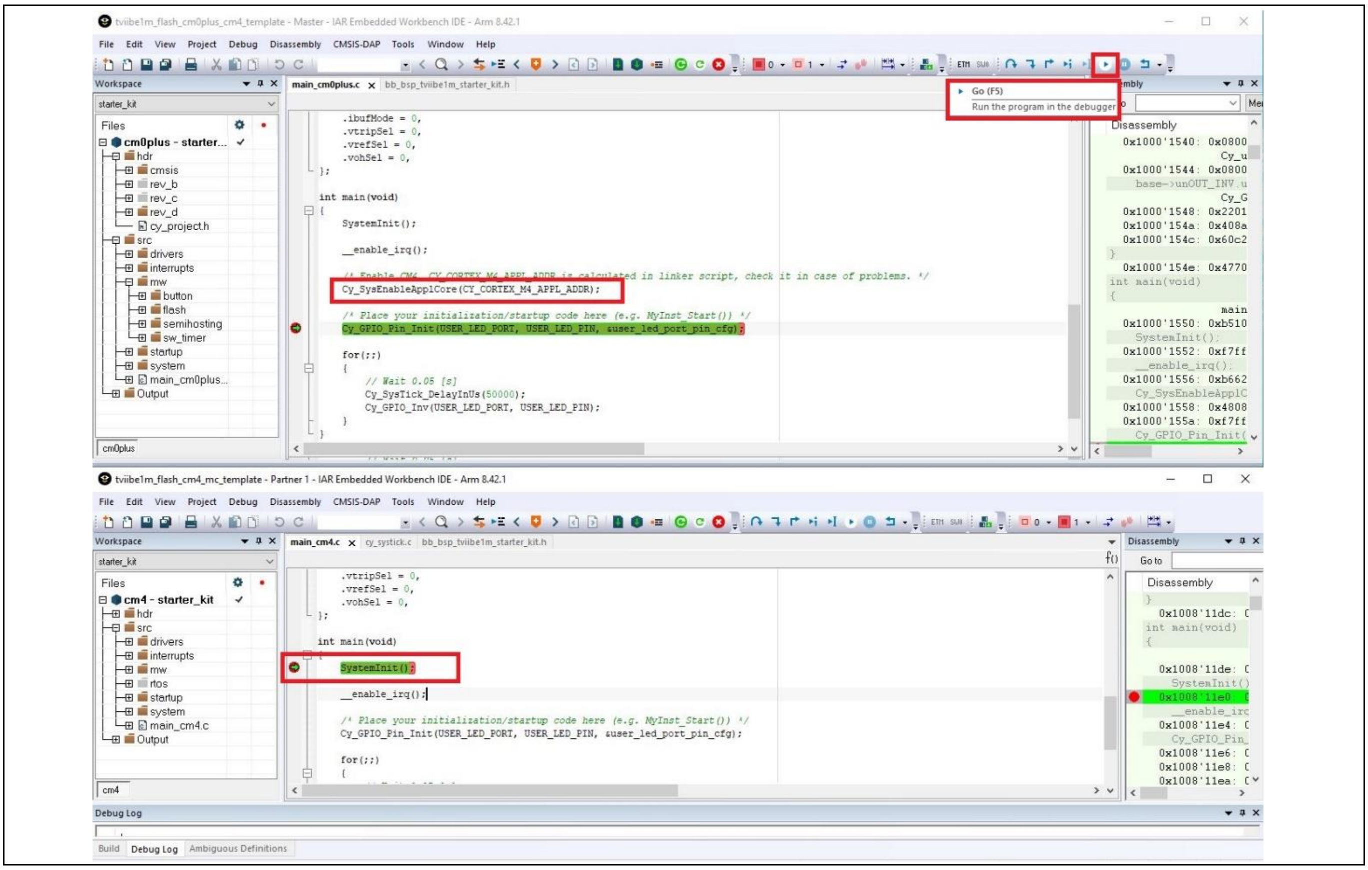


```
    .ibufMode = 0,
    .vtripSel = 0,
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of problems. */
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

```
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
```

Figure 25 Dual core Debug windows

7. Now, click the **Go** icon in the CM0+ workspace to start execution. After executing `Cy_SysEnableApplCore()`, CM4 core will be enabled and the execution will reach the breakpoint in the CM4 workspace. You can now continue to debug the code from CM4 core.


tviibe1m_flash_cm0plus_cm4_template - Master - IAR Embedded Workbench IDE - Arm 8.42.1
File Edit View Project Debug Disassembly CMSIS-DAP Tools Window Help
Workspace
starter_kit
Files
cm0plus - starter...
hdr
cmsis
rev_b
rev_c
rev_d
cy_project.h
src
drivers
interrupts
mw
button
flash
semihosting
sw_timer
startup
system
main_cm0plus...
Output
cm0plus
main_cm0plus.c
bb_bsp_tviibe1m_starter_kit.h
Go (F5)
Run the program in the debugger
Disassembly

```
    .ibufMode = 0,
    .vtripSel = 0,
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Enable CM4. CY_CORTEX_M4_APPL_ADDR is calculated in linker script, check it in case of problems. */
    Cy_SysEnableApplCore(CY_CORTEX_M4_APPL_ADDR);

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
        // Wait 0.05 [s]
        Cy_SysTick_DelayInUs(50000);
        Cy_GPIO_Inv(USER_LED_PORT, USER_LED_PIN);
    }
}
```

```
    .vtripSel = 0,
    .vrefSel = 0,
    .vohSel = 0,
};

int main(void)
{
    SystemInit();

    __enable_irq();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    Cy_GPIO_Pin_Init(USER_LED_PORT, USER_LED_PIN, &user_led_port_pin_cfg);

    for(;;)
    {
```

Figure 26 **Debugging with CM0+ and CM4 cores**

LED1 and LED4 should start blinking once both the cores are running.

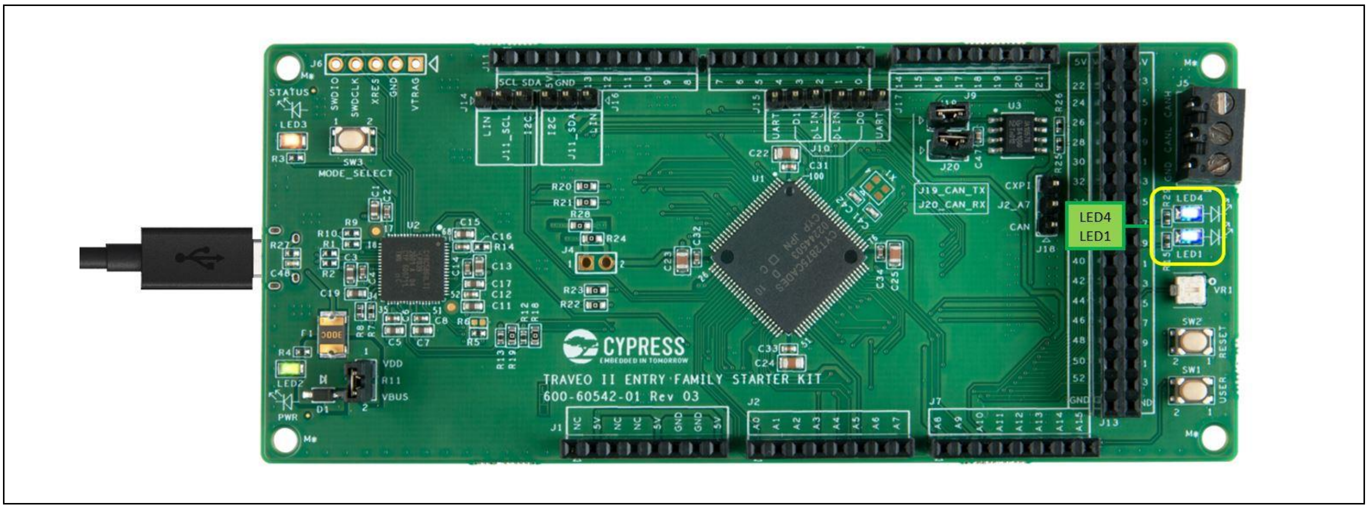


Figure 27 **Blinking LED1 and LED4**

3 Troubleshooting

This section explains possible issues and the workarounds.

3.1 Connection troubleshooting

Error: Starter kit is not detected on the target system.

- Connect the USB cable that comes with the Starter kit. Other USB cables may not connect data lines.
- Make sure LED3 is ON (CMSIS-DAP mode). If not, press SW3 to change the KitProg3 device mode.
- Make sure jumper R11 is closed.

3.2 Driver troubleshooting

Error: Driver is not detected on the target system or "KitProg3" is not visible

For more information on the supported driver, see the KitProg3 user guide **[1]**.

3.3 Debugger troubleshooting

Error: While programming Traveo II device, CMSIS-DAP device is not found.

Check the USB cable connection and the state of LED3 (LED should be ON for CMSIS-DAP mode).

3.4 Key points

- CM0+ core should be running while debugging with CM4 in single core mode. CM0+ core enables the CM4 core by calling the `Cy_SysEnableApplCore()` API.
- CM4 core workspace (tviibe1m_flash_cm4_mc_template) should be built before building the dual core workspace (tviibe1m_flash_cm0plus_cm4_template).

Glossary

Terms	Description
CM0+	Arm Cortex-M0 plus
CM4	Arm Cortex-M4
SDL	Sample Driver Library
SK	Starter Kit
USB	Universal Serial Bus
EWARM	Embedded Workbench for ARM
C-SPY	High Level Debugger Language for Embedded System

References

[1] **KitProg3 User Guide**

[2] **Traveo II Starter Kit User Guide**

Revision history

Document version	Date of release	Description of changes
**	2021-04-08	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

1.

Edition 2021-04-08
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?

Go to: www.cypress.com/support

Document reference
002-32910 Rev.**

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.