

Microchip MPLAB Debugger

Abstract

This document describes the usage of the old Microchip MPLAB debugger and the new Microchip MPLAB X debugger as target system. The minimum required version of MPLAB is 7.62. The minimum required version of MPLAB X is 1.90. TESSY supports the PIC16, PIC18, PIC24, PIC32, and the dsPIC controller family.

Table of Contents

Abstract	1
1 Microchip MPLAB Debugger	2
1.1 Setup of the MPLAB Environment.....	2
1.2 TESSY Environment Settings	4
1.2.1 Changing the Debug Target.....	5
1.3 Troubleshooting.....	6
1.3.1 Error Dialogs during Test Execution.....	6
1.3.2 Error Messages within the TESSY Message Window.....	7
2 Microchip MPLAB X Debugger	8
2.1 Target Configuration.....	8
2.2 Specific Tool Id.....	9
3 Setting PIC Configuration Bits & Fuses	10
3.1 Init Definitions.....	10
3.2 Use Board Files.....	11
4 SAM Device Configuration	16
5 Interactive Debugging.....	18
6 Known Issues	22
6.1 PIC16 Memory	22
6.2 XC32 Compiler	22
6.3 The target device is not ready for debugging	22
6.4 tthd: Java processes died unexpectedly, exiting... ..	22
6.5 MPLAB X Version > 2.15: Subsequent Test Runs Fail	22
6.6 Problem with interactive debugging.....	23

Please note: The MPLAB debugger does not support interactive debugging features when executing tests with TESSY. The normal debugger GUI is not accessible during the TESSY test execution. (See 5 to learn how to debug interactively having your test data statically built into the target binary.)

1 Microchip MPLAB Debugger

The communication between TESSY and MPLAB is based on the Microchip DBT (debug tool) interface DLL, which allows controlling the MPLAB target emulator or simulator. This DLL does not allow simultaneous access to the target hardware both from MPLAB and TESSY. This restricts the usage of the MPLAB frontend GUI: It is not possible to step into debugging or use the GUI when executing tests with TESSY.

The DBT API requires the connection to the target to be setup correctly before running tests with TESSY. The section below describes how to do the initial connection settings.

It is also necessary to configure certain configuration bits or registers for proper operation of the target controller. This depends on the actual controller in use and may differ from the default settings provided with TESSY. Please adjust all the controller specific settings accordingly.

Important note: The target connection and controller specific settings are essential for proper operation of the DBT interface. Especially correct watchdog, power on reset and oscillator settings are required to get the controller running. See 3 for further details.

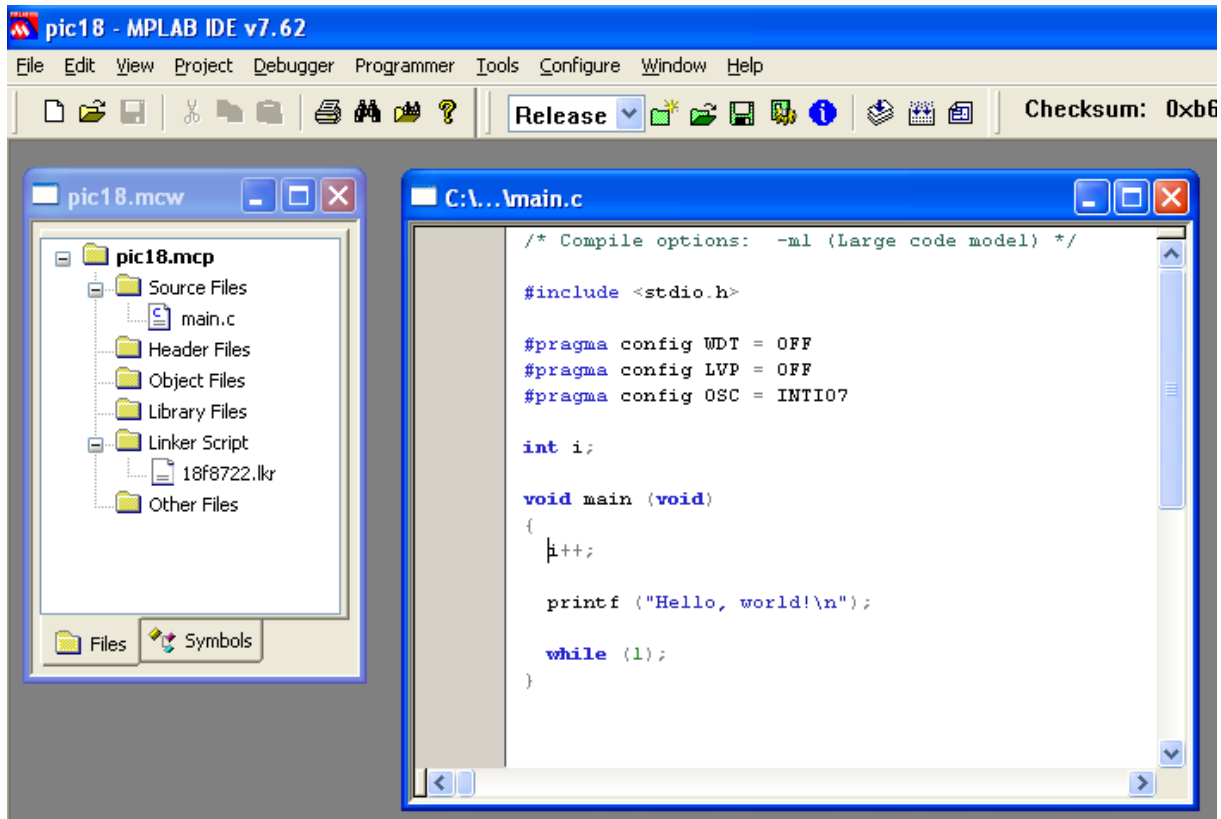
1.1 Setup of the MPLAB Environment

Before running a test from TESSY, please make sure that the debug hardware (e.g. ICD2, ICD3, ICD4, ICE 2000, ICE 4000, PICKit 3/4, Real ICE) and the target controller on the target board are working correctly when operated from within MPLAB. You should create a sample project or use your current development project and download and run this project on the debug hardware.

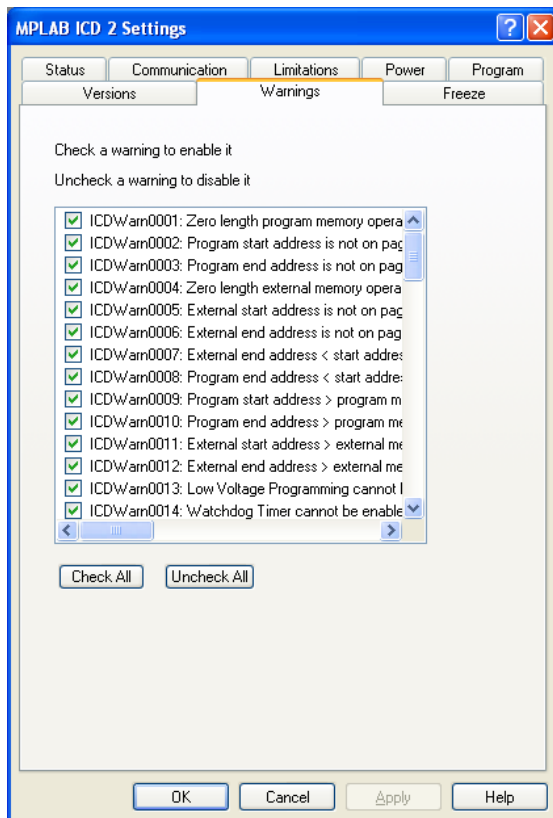
Please follow the steps below to run this sample project:

- Create a sample project using the MPLAB **Project wizard**. Select the appropriate CPU, add a C source code file containing an (empty) main program and add the linker script file.
- Select the appropriate debug tool using the **Debugger->Select tool** menu (e.g. ICD3).
- Build your application
- Download your application to the target
- Execute your application and step through the code.

After these steps, your project should look like this:



Make sure to switch on all warning messages within MPLAB before executing the sample project as shown in the screenshot below (to open this dialog choose **Debugger**→**Settings...** from the MPLAB menu bar):



A sample source code for use with the PIC18F8722 would be the following:

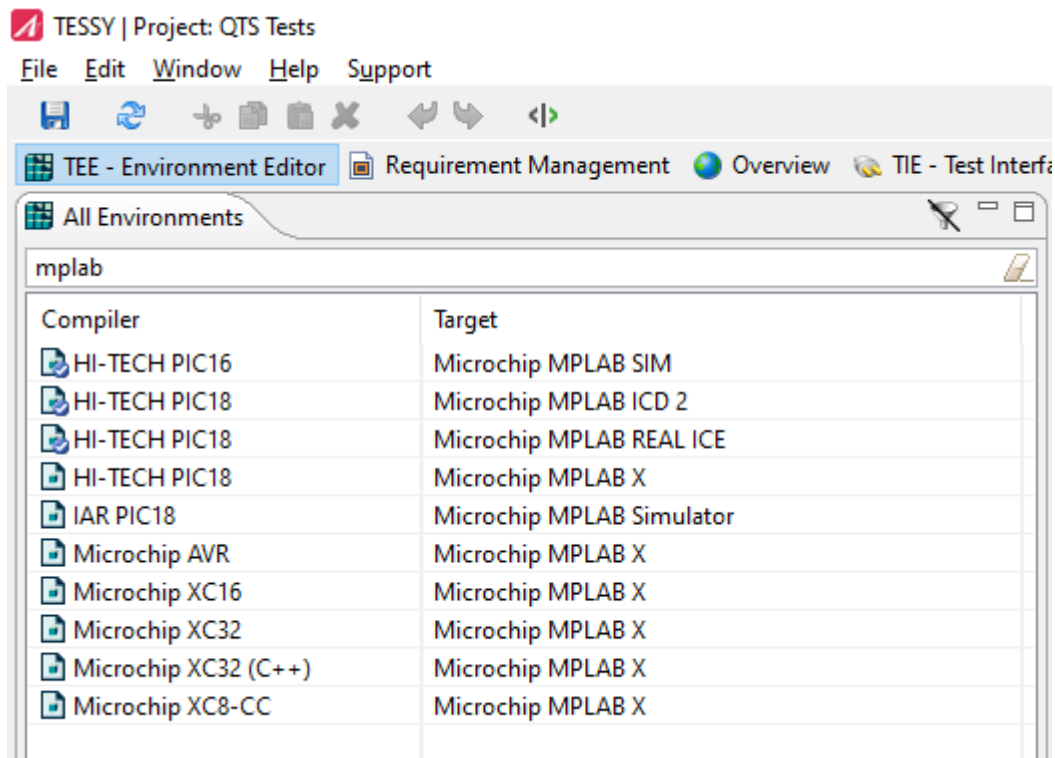
```
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config OSC = INTIO7

void main(void)
{
    while (1);
}
```

This application code disables the watchdog, disables the low voltage programming, and sets the oscillator to internal.

1.2 TESSY Environment Settings

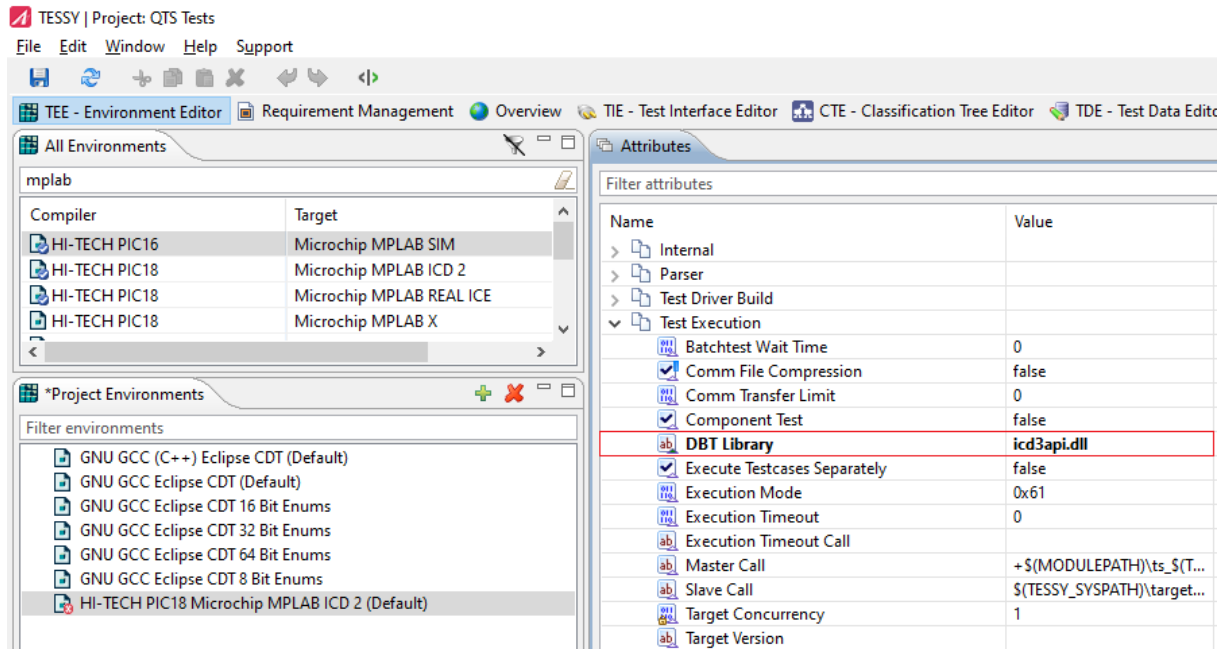
The TESSY environment contains different entries for MPLAB as target debugger within the **Configurations** section:



Activate your desired debug hardware (e.g. ICD3) and review the settings for **CPU**, **Linker File** and **Init Definitions** as shown below. The difference between the three MPLAB configurations is the value of the **DBT Library** attribute which contains the respective debug target DLL.

1.2.1 Changing the Debug Target

If you want to change the debug target platform, you need to adapt the **DBT Library** attribute. This attribute contains the name of the interface DLL used for communication with the MPLAB debug target.



The table below shows the available DLL files and the respective debug targets. You find these files within the TESSY installation directory:

C:\Program Files\Razorcat\TESSY_4.3\sys\targets\mplab

DLL file name	Debug target	Remarks
ICD2API.DLL	ICD2	
ICD3API.DLL	ICD3	
ICE2KAPI.DLL	ICE 2000	Not tested with TESSY
ICE4KAPI.DLL	ICE 4000	Not tested with TESSY
MPSIMAPI.DLL	MPLAB Simulator	
REALICEAPI.DLL	Real ICE	

The ICD2, ICD3, Real ICE and MPLAB simulator API have been tested with TESSY.

1.3 Troubleshooting

The test execution on the target may fail, if one or more of the following settings are invalid or missing:

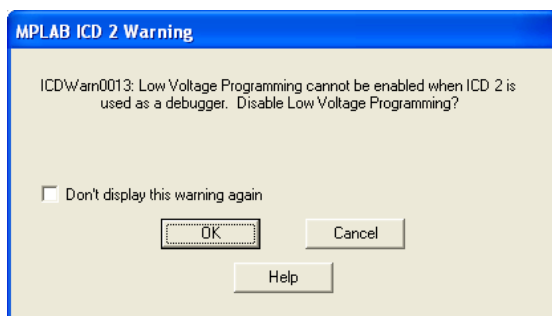
- watchdog (WDT) disabled
- low voltage programming (LVP) switched off
- oscillator setting (OSC)
- power on reset (POR) disabled
- incorrect operating system on debug hardware (e.g. ICD2) for the current controller

Refer to the sections above for details on how to correct the settings.

1.3.1 Error Dialogs during Test Execution

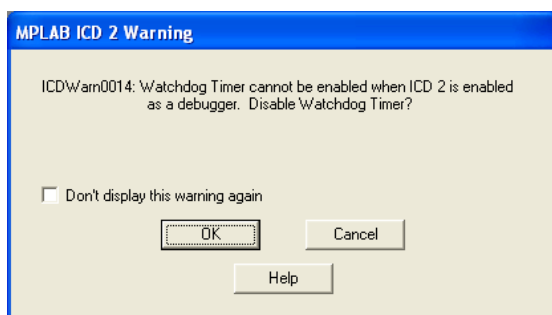
If one of the following dialogs show up during test execution, you need to check the settings of the configuration bits of the controller (e.g. for PIC18F8722).

Please note: The proposed settings are examples for the PIC18 controller only. For other controllers (e.g. dsPIC) or derivatives of the PIC18 different syntax or configuration values may apply.



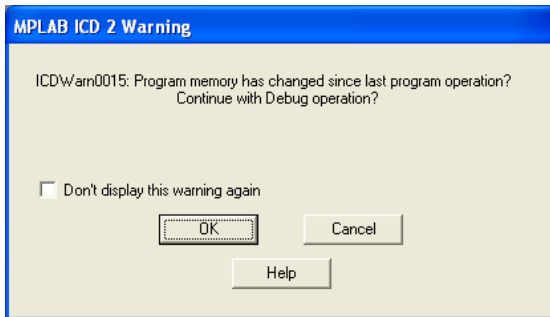
The warning above indicates that the low voltage programming is enabled. This requires the following config setting (e.g. for PIC18F8722):

```
#pragma config LVP = OFF
```



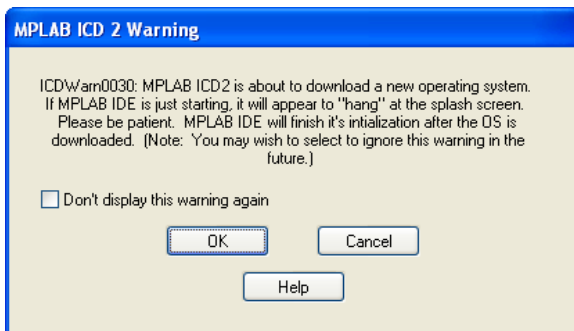
The warning above indicates that the watchdog is not disabled. This requires the following config setting (e.g. for PIC18F8722):

```
#pragma config WDT = OFF
```



The warning above indicates that the controller is not running and the communication with the debug hardware fails. This may be due to wrong oscillator settings. For PIC18F8722 the following settings may be necessary:

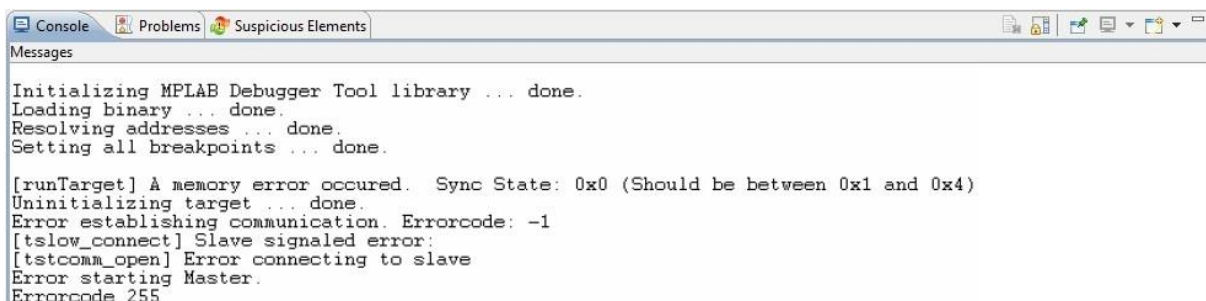
```
#pragma config OSC = INTIO7
```



The warning above indicates that the debug hardware (e.g. ICD2) has the wrong operating system loaded for the current target controller. Please start MPLAB and download the correct operating system for the debug hardware.

1.3.2 Error Messages within the TESSY Message Window

The following error messages indicate that the target controller is not running correctly. Please refer to the configuration settings described above for watchdog, power on reset and oscillator settings.



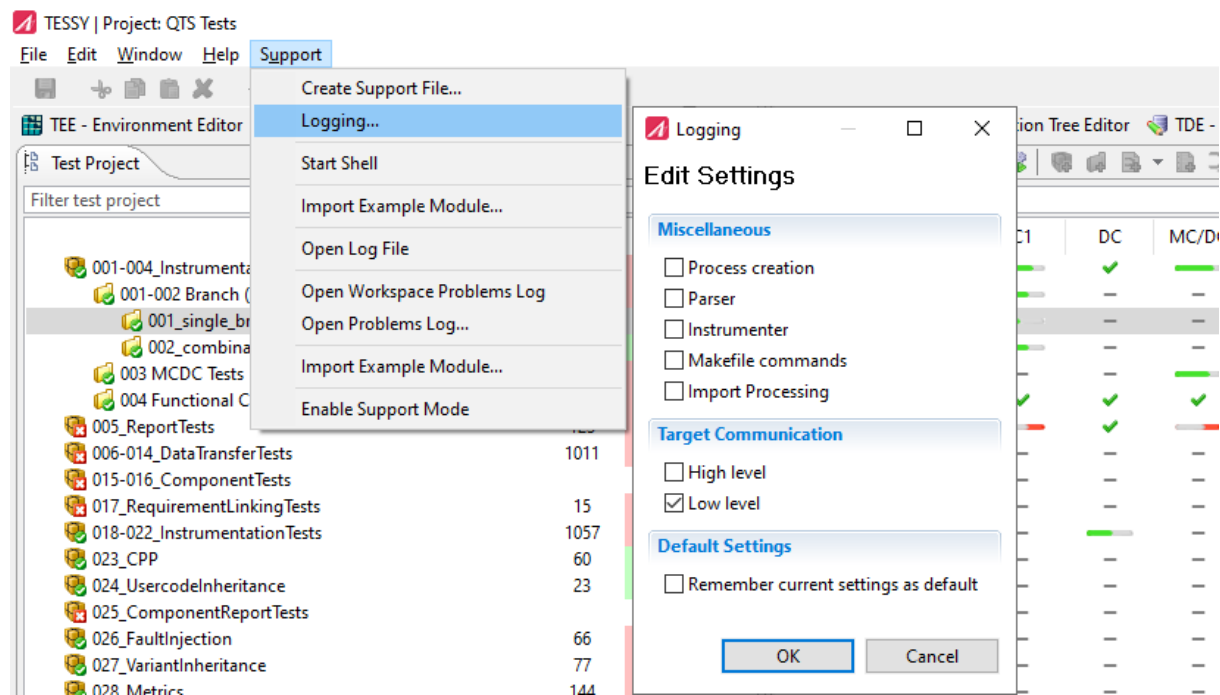
2 Microchip MPLAB X Debugger

The communication between TESSY and MPLAB X is based on the Microchip's MPLAB X SDK. The device to be used is chosen from the value of the TEE attribute **Target Configuration**. The default value is **Simulator**.

Important note: The target configuration settings are essential for proper operation of the MPLAB X SDK interface. Especially correct watchdog, power on reset and oscillator settings are required to get the controller running. See 3 for further details.

2.1 Target Configuration

To get a list of all available devices, i.e. **Tool Metas** and **Tools**, turn on “Low Level” logging from TESSY’s Support menu



and run a test. TESSY will list detailed information about the communication among which the available **Tool Metas** and **Tools** are printed.



```

Console | Problems | Progress
Messages
Tool Metas:
Name = ICD 3
Name = PICKit2
Name = PICKit3
Name = Licensed Debugger
Name = Starter Kits (PKOB)
Name = PM3
Name = Real ICE
Name = Simulator
Name = MCHV
Name = MICROSTICK
Name = SKDE 33 AUDIO
Name = SKDE Memory
Name = SKDE PIC18FJ
Name = SKDE PIC24F 1
Name = SKDE PIC24H SENSOR
Name = SKDE PIC32

Tools:
0 = :=MPLABCommUSB:=04D8:=8107:=0002:=Microchip Technology Incorporated:=PIC32MZ EC Family:=BUR135298081:=x:=h:=en

```

Please choose the one that fits your needs, e.g. Real ICE, and set the **Target Configuration** attribute in the TEE accordingly.

Attributes	
Name	Value
 Target Configuration	Real ICE
 Target Install Path	\$(ProgramFiles)\Microchip\MPLABX
<input checked="" type="checkbox"/> Generate Builtin Data	false
<input checked="" type="checkbox"/> Use Board Files	false

2.2 Specific Tool Id

For some targets TESSY cannot determine the **Tool Id** from the **Tool Meta** name. If you encounter an error message which looks similar to the following **Console** view example,

```

Console | Problems | Progress
Messages
ERROR: Couldn't find tool Id for Starter Kits (PKOB)

```

you will have to add the **Tool's** index to the **Target Configuration** attribute as well. The **Tool Id** index number can be determined as shown below.

```

Console | Problems | Progress
Messages
Tool Metas:
Name = ICD 3
Name = PICKit2
Name = PICKit3
Name = Licensed Debugger
Name = Starter Kits (PKOB) ← Tool Meta Name
Name = PM3
Name = Real ICE
Name = Simulator
Name = MCHV
Name = MICROSTICK
Name = SKDE 33 AUDIO
Name = SKDE Memory
Name = SKDE PIC18FJ
Name = SKDE PIC24F 1
Name = SKDE PIC24H SENSOR
Name = SKDE PIC32

Tools: Tool Index
0 ← :=MPLABCommUSB:=04D8:=8107:=0002:=Microchip Technology Incorporated:=PIC32MZ EC Family:=BUR135298081:=x:=h:=en (used)

```

For example, if our **Tool Meta** name is **Starter Kits (PKOB)** and our **Tool** index is **0** the entry for the **Target Configuration** attribute is

Attributes	
Name	Value
 Target Configuration	Starter Kits (PKOB):=0

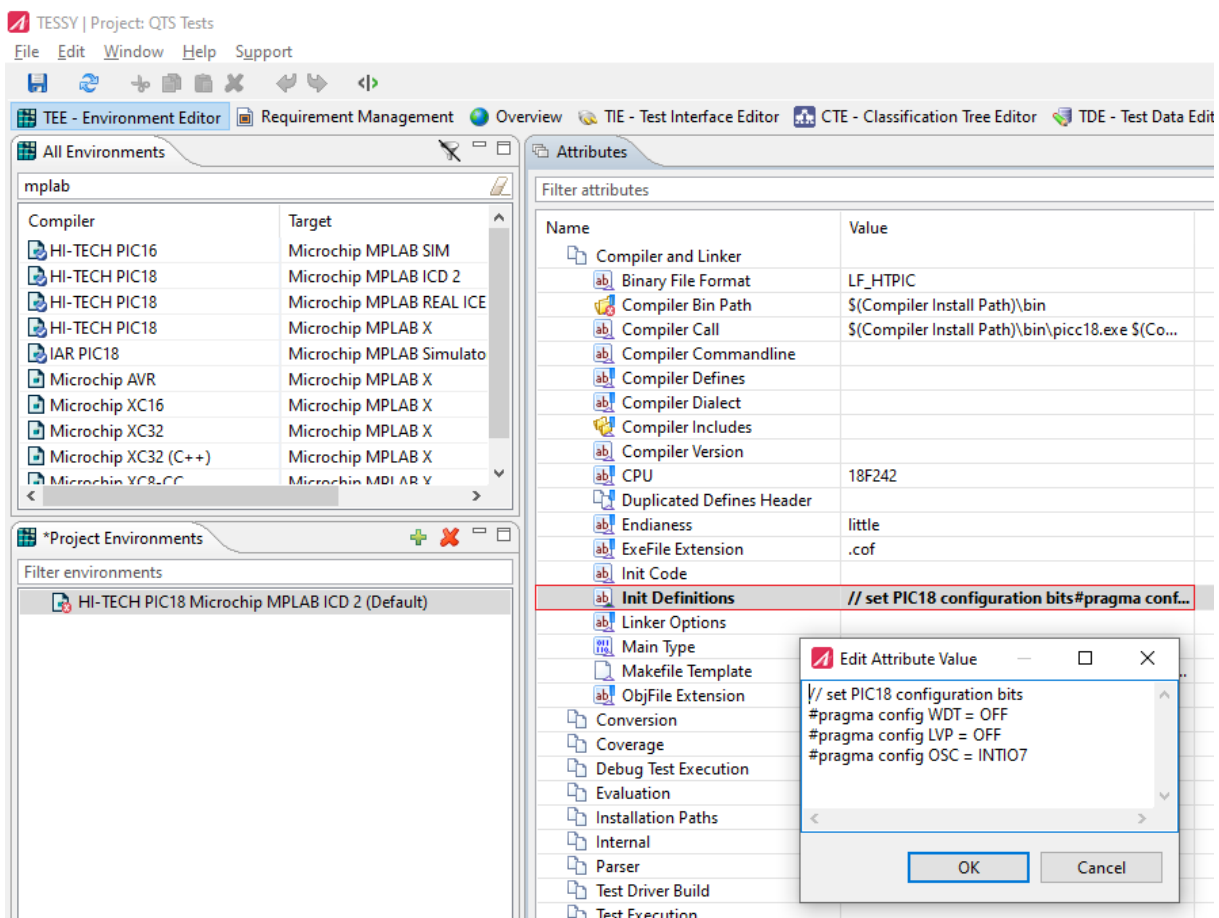
Please note the **:=** between the **Tool Meta** name and the **Tool** index.

3 Setting PIC Configuration Bits & Fuses

There are two ways to specify the configuration bits with TESSY. If only a few bits have to be set at the beginning of function main() you may use the TEE attribute **Init Definitions**. It is however recommended to create and compile your own configuration_bits.c file using MPLAB X and link this object to the test binary (see 3.2).

3.1 Init Definitions

The **Init Definitions** attribute is one way to specify definitions that control the configuration bits (also referred to as fuses) of the controller. The sample below shows the necessary configuration settings for the PIC18F8722 controller. This code requires adjustments for different microcontrollers. Refer to the compiler help for details about how to configure your specific microcontroller.



The screenshot shows the TESSY IDE interface. The 'Attributes' window is open, displaying a list of attributes. The 'Init Definitions' attribute is highlighted in red. An 'Edit Attribute Value' dialog box is open over it, showing the following configuration code:

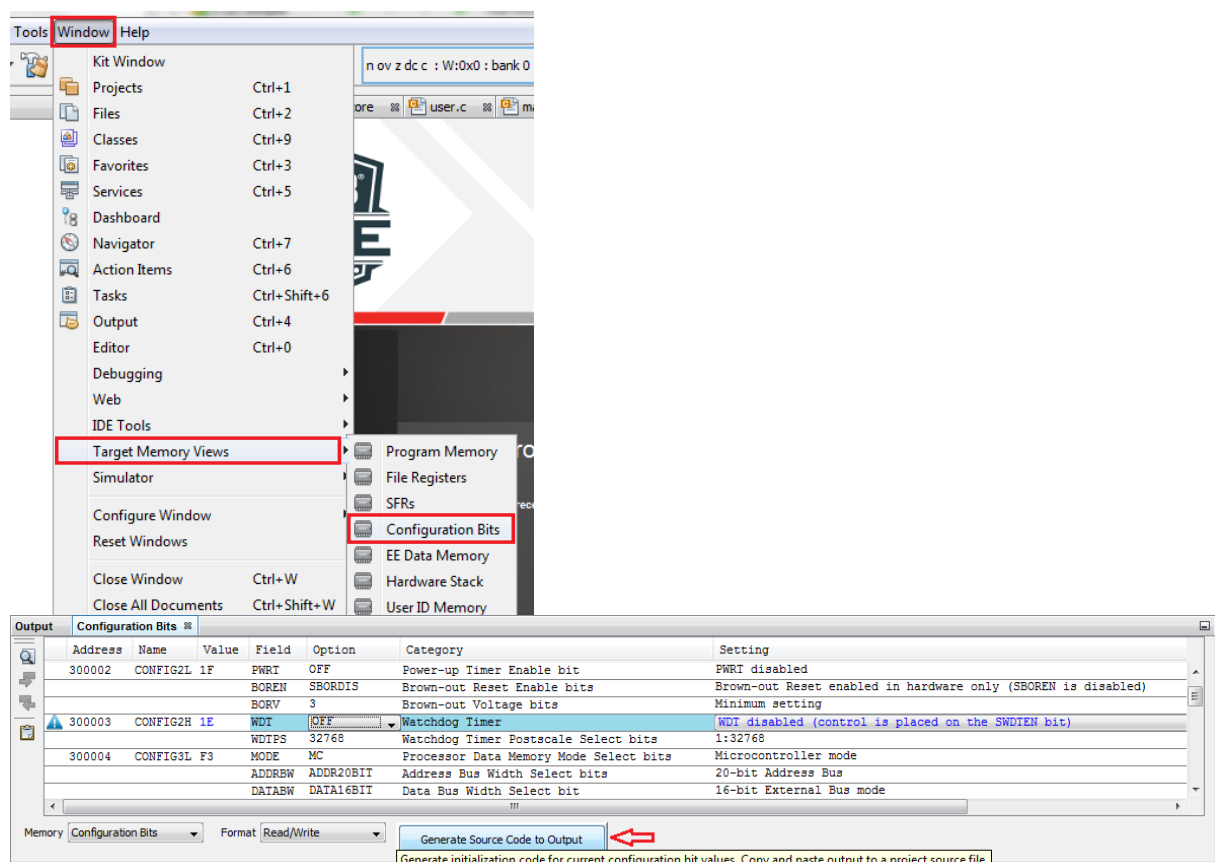
```
// set PIC18 configuration bits
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config OSC = INTIO7
```

If the **Init Definitions** are incorrect for your selected controller, you will see error messages while compiling the test driver in TESSY's **Console** view.

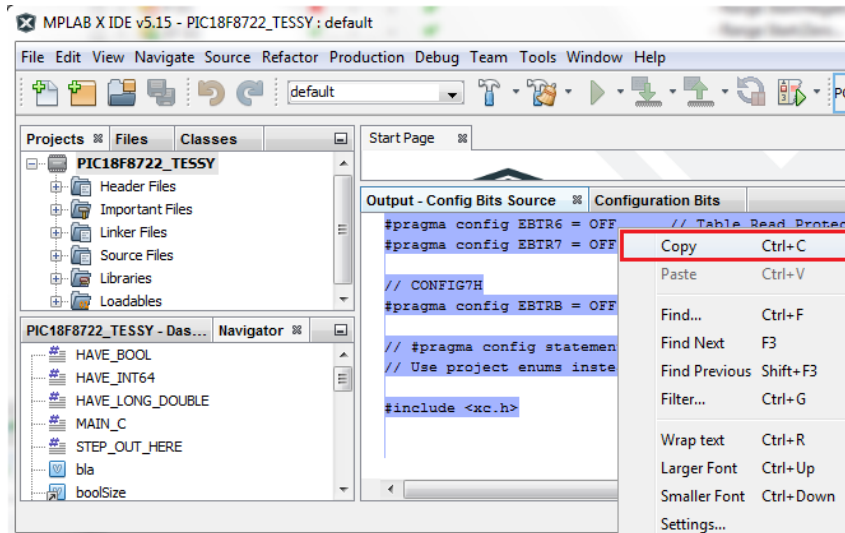
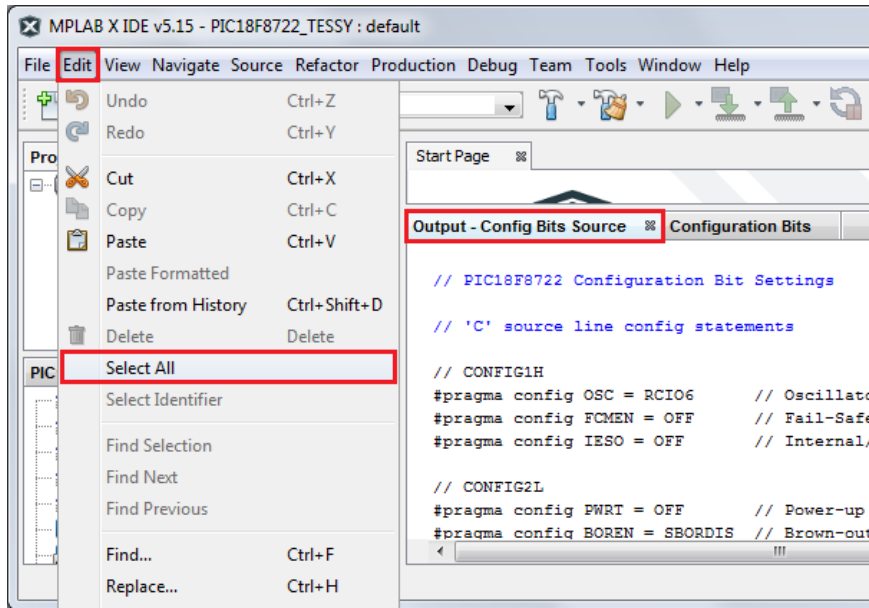
3.2 Use Board Files

It is recommended to create and compile a configuration bits file using MPLAB X and link the object file to the test binary. Please follow these instructions to do this.

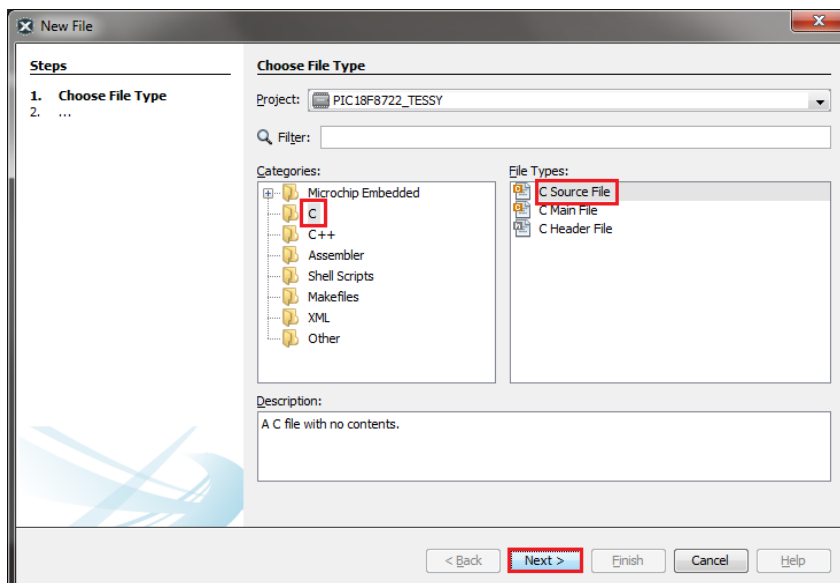
Open the **Configuration Bits** view, turn the watchdog timer off, and select **Generate Source Code to Output**.



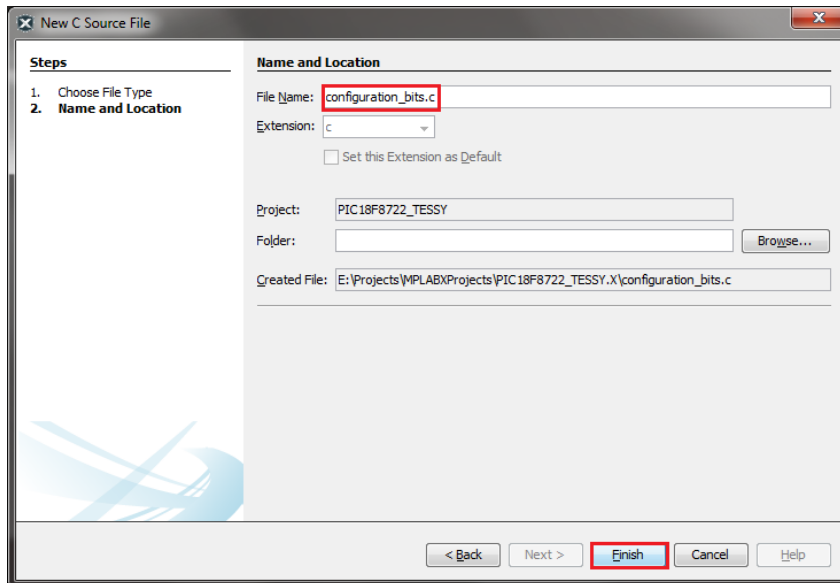
Select the **Output** view, select all of its content, and copy the content.



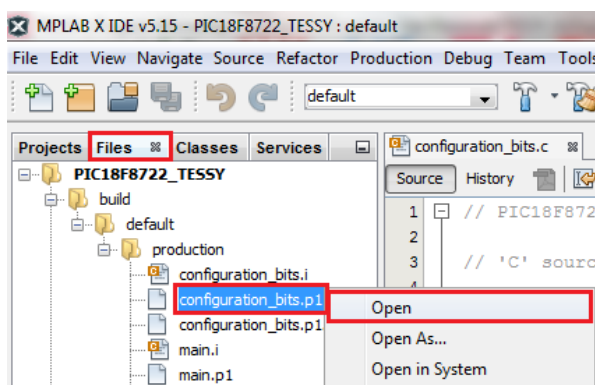
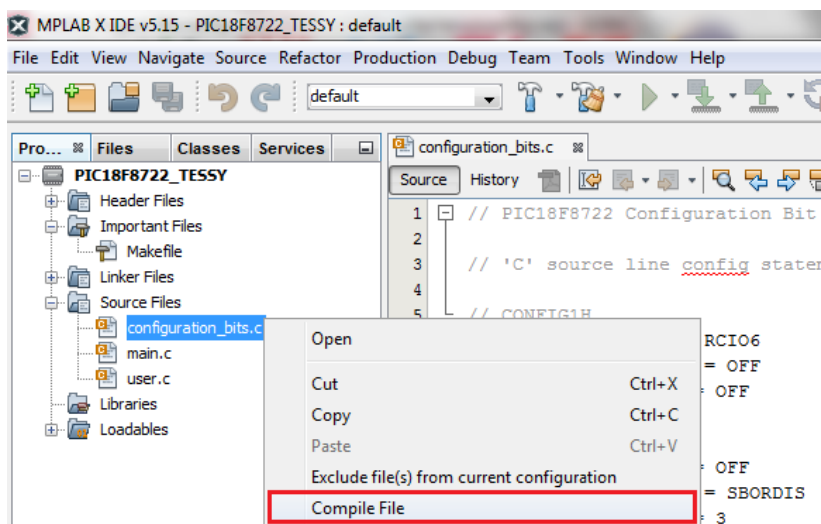
From the application's menu select **New File...** and choose **C Source File** as file type. Click **Next**, choose *configuration_bits.c* as file name and click **Finish**.



TESSY Application Notes

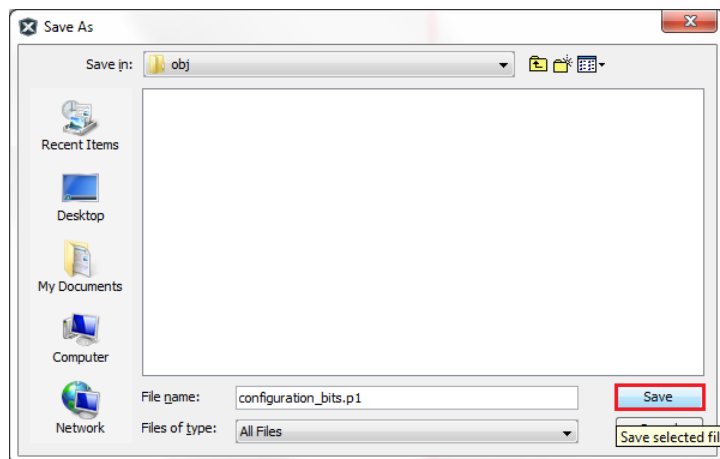
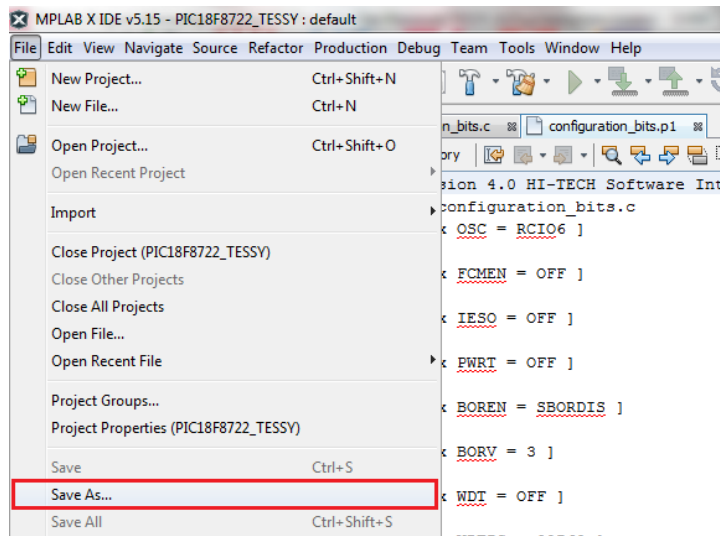


Compile the file, select the **Files** tab, and open the generated object file.

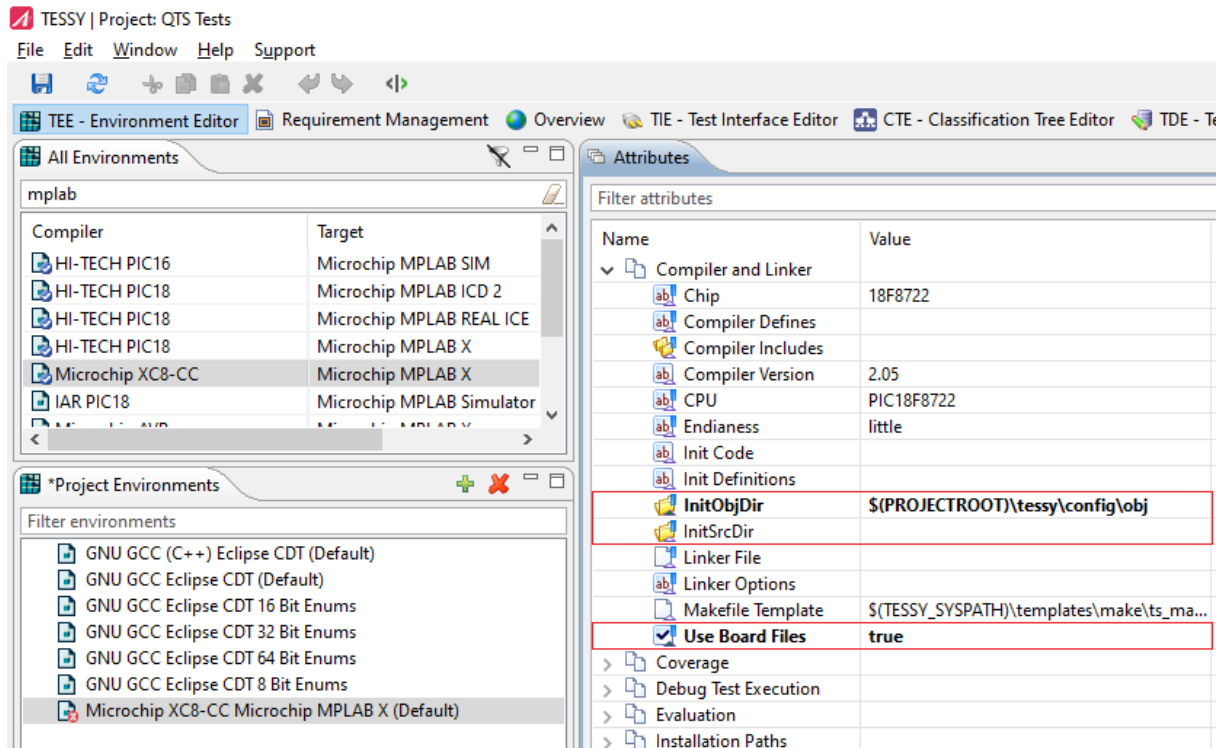


TESSY Application Notes

Create an object directory within your TESSY project's **config** directory and save `configuration_bits.p1` into this directory.



Launch the TEE and let attribute **InitObjDir** point to this directory and switch attribute **Use Board Files** to **true**. Save your settings.



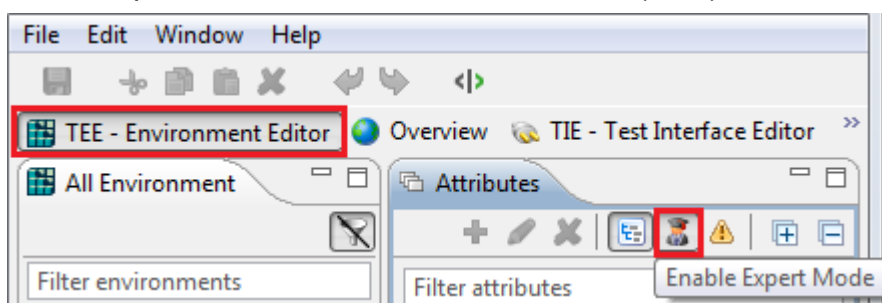
As long as TEE attribute **Use Board Files** is set to **true** all of your test objects for this configuration combination will be linked with the objects found in the directory pointed to by **InitObjDir**.

4 SAM Device Configuration

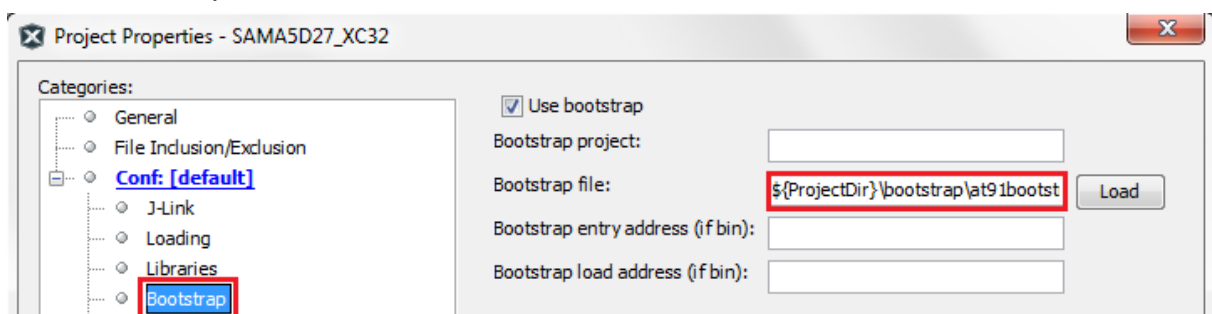
Please download the Microchip/SAMA5D27 TESSY sample project from our website at <https://www.razorcat.com/en/teffy-sample-projects.html>.

The target device requires the **SAM-BA Command Line Tool** to reset the MPU after each unit test. You can download the tool from <https://www.microchip.com/en-us/development-tool/SAM-BA-In-system-Programmer>. If your board is not supported by this tool, you will have to find another solution to trigger a hardware reset for the MPU. Furthermore, the SAM device requires a bootstrap program to be loaded before the test binary generated by TESSY is loaded. Also, board files are required, which you will find within your MPLAB X project folder.

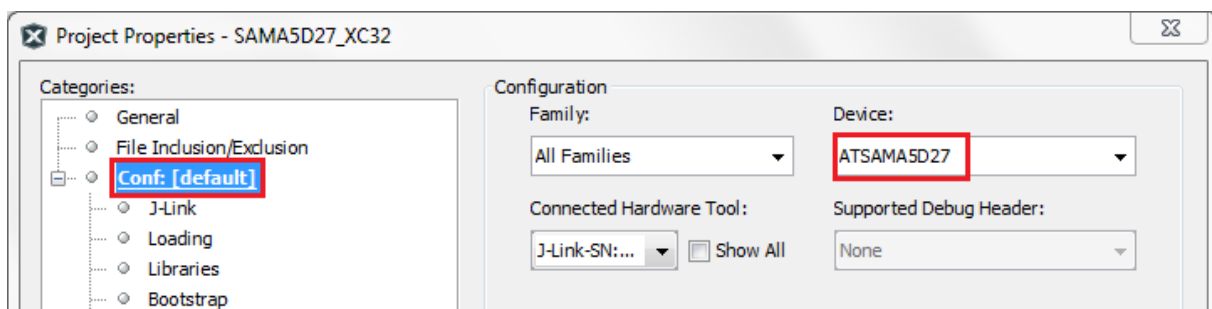
At first, open the TESSY environment editor (TEE) and turn on Expert Mode.



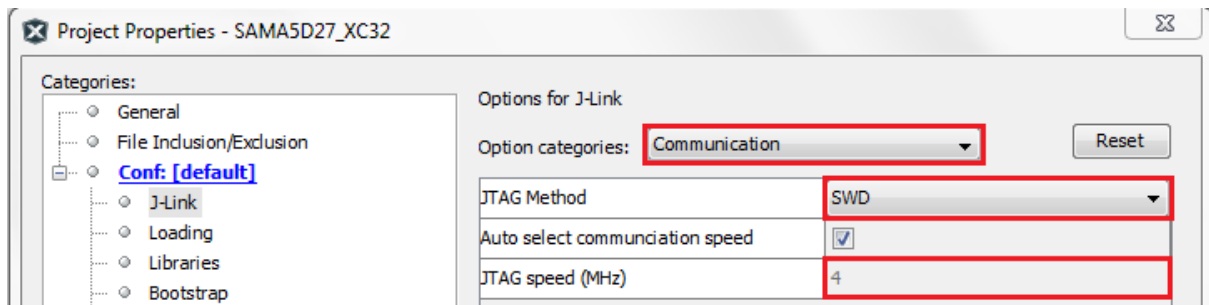
Our sample project was configured for the **SAMA5D27 SOM1 Evaluation Kit 1** which is connected via a J-Link interface. If you are using the same board and interface, all you have to do is to adjust the **Compiler Install Path**, the **Target Install Path**, and the **SAM-BA Command Line Tool** path. If you use another SAM board or interface, you will probably have to adjust the TEE attributes **Boot Path**, which points to the bootstrap file,



Chip, which means the device,



Communication Interface and Communication Speed,



as well as the **SAM-BA Board** and **SAM-BA Port**, which will be given by using the **SAM-BA Command Line Tool** as shown below.

```
C:\>E:\Microchip\sam-ba_3.5\sam-ba.exe -p help
```

```
Known ports: serial, j-link, secure
```

```
C:\>E:\Microchip\sam-ba_3.5\sam-ba.exe -p j-link -b help
```

```
Known boards: sam9x60-ek, sam9xx5-ek, sama5d2-icp, sama5d2-ptc-ek, sama5d2-xplained, sama5d27-som1, sama5d27-som1-ek, sama5d27-wlsom1-ek, sama5d3-ek, sama5d3-xplained, sama5d4-ek, sama5d4-xplained, sama7g5-ek
```

So, in our TESSY sample project **SAM-BA Board** is set to **sama5d27-som1-ek**, while **SAM-BA Port** is set to **j-link**.

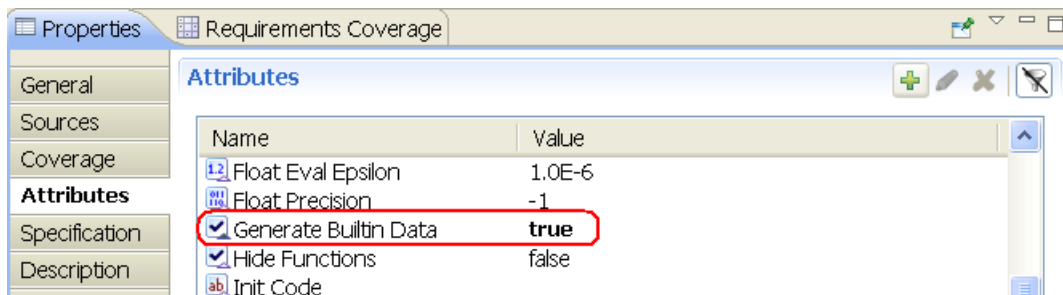
Furthermore, check your path to the **Device Family Pack**, of which version **1.6.69** or higher is required.

Using a different board also requires different board files, which you will find most likely within your MPLAB X project folder.

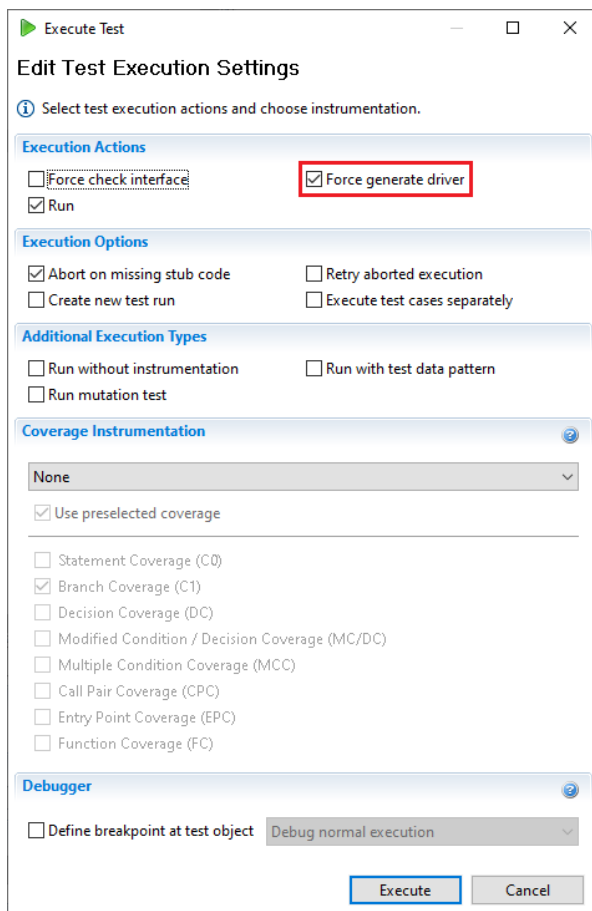
Important note: Before you start adjusting the TESSY SAM configuration, please make sure, that your MPLAB X project is working properly, i.e. that you can compile, link, and debug/run your built program successfully within the MPLAB X debugger!

5 Interactive Debugging

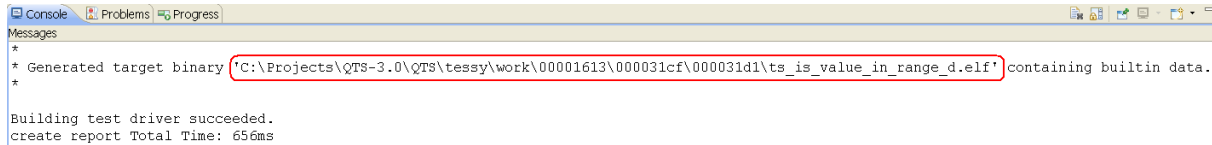
The TESSY MPLAB and TESSY MPLAB X adaptations do not support interactive debugging directly. In order to debug your target interactively TESSY provides the **Generate Builtin Data** attribute. The attribute is of type Boolean and if set to **true** TESSY will rebuilt your target binary on the next test run having the selected test data built-in, i.e. TESSY will not actually perform the test run but instead create the target binary with test data built-in. To disable this feature, you have to set the attribute to **false**.



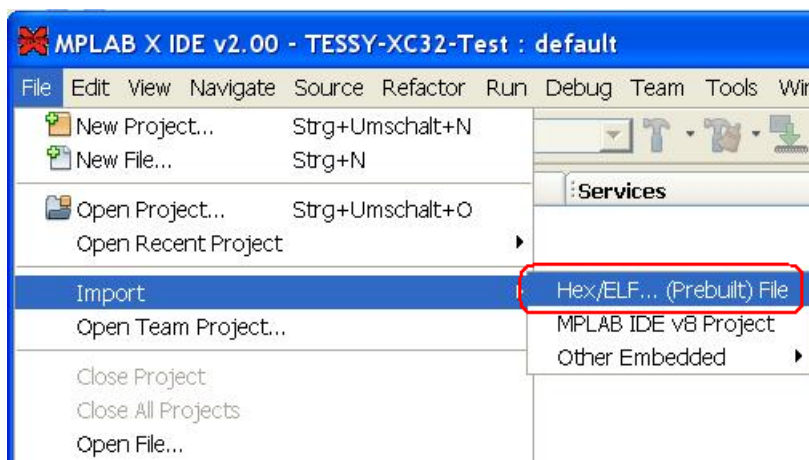
Open the **Execute Test** dialog and make sure **Force Generate Driver** is selected.



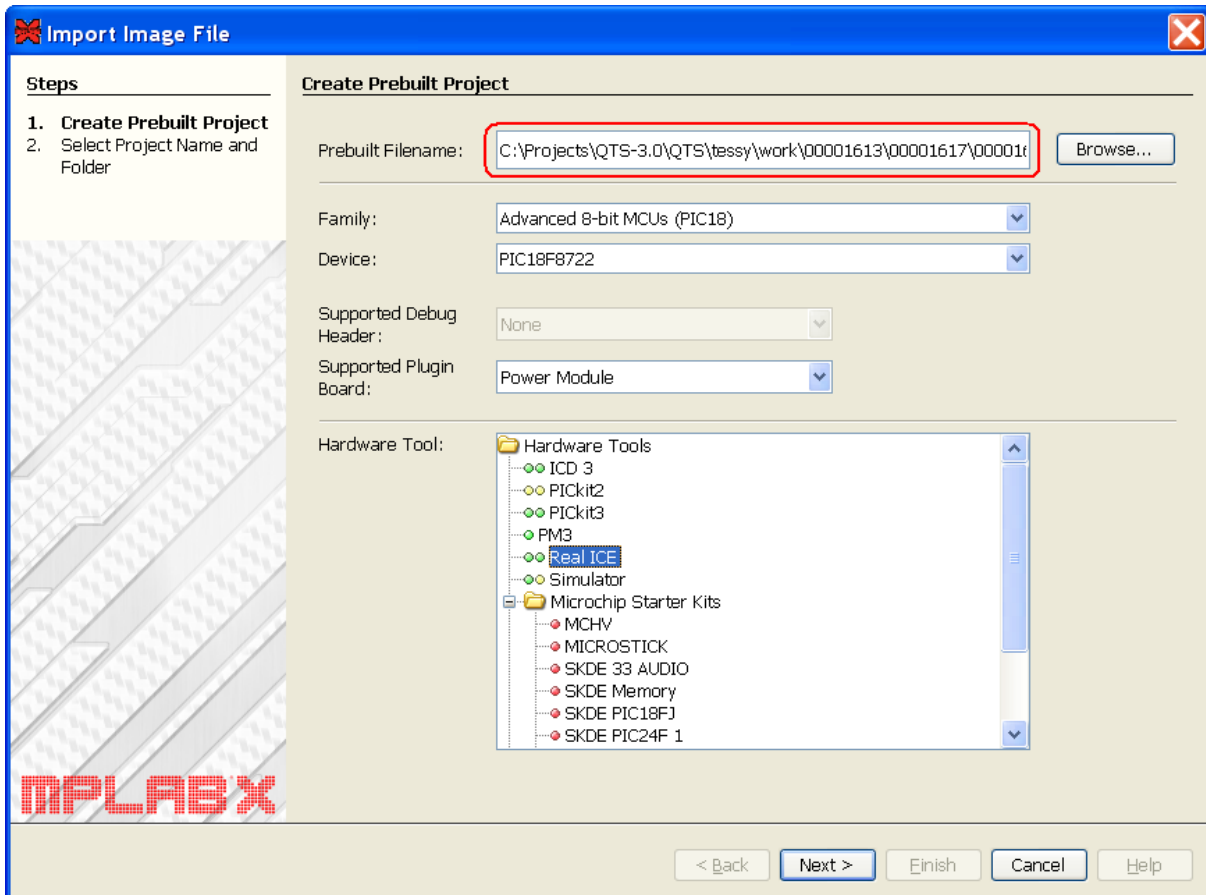
Now execute the test by pressing the **Execute** button. TESSY displays the path to the generated built-in target binary in the **Console** view.



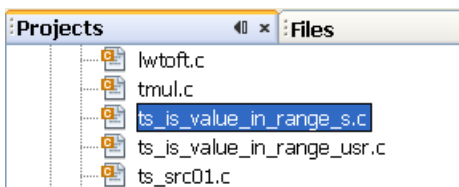
You can copy the path to the generated target binary from TESSY's **Console** view, open the **Import Image File** dialog of MPLAB X,



and paste it into the **Prebuilt Filename** box. Select your processor's family, device, and hardware tool and click next and finish.



From the **Projects** view of MPLAB X open your imported project and double click to the file containing `main()`. The file name is `ts_<test_object_name>_s.c`. That is, if your test object is called `is_value_in_range` the name of the `main()` function file is `ts_is_value_in_range_s.c`.



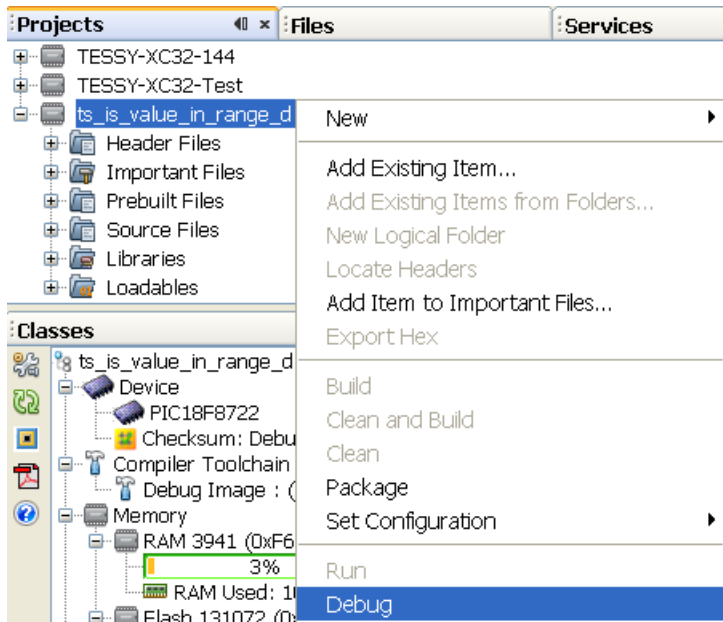
Scroll the editor to the end of the file and add a breakpoint as shown below.

TESSY Application Notes

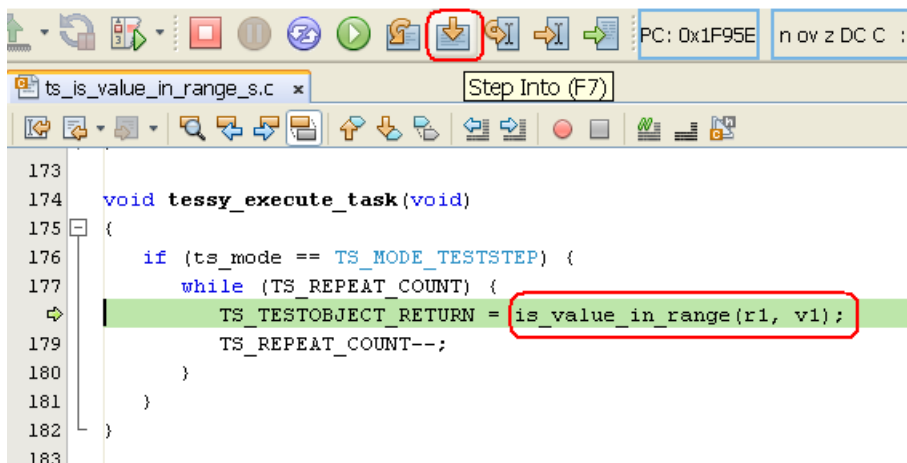
```

ts_is_value_in_range_s.c x
215 #if !defined TS_OTHER_MAIN
216 void main(void)
217 {
218     tussy_init_task();
219
220     while (1) {
221         tussy_start_task();
222         tussy_execute_task();
223         tussy_end_task();
224     }
225 }
226 }
227 #endif
  
```

Start the debugger as shown below.



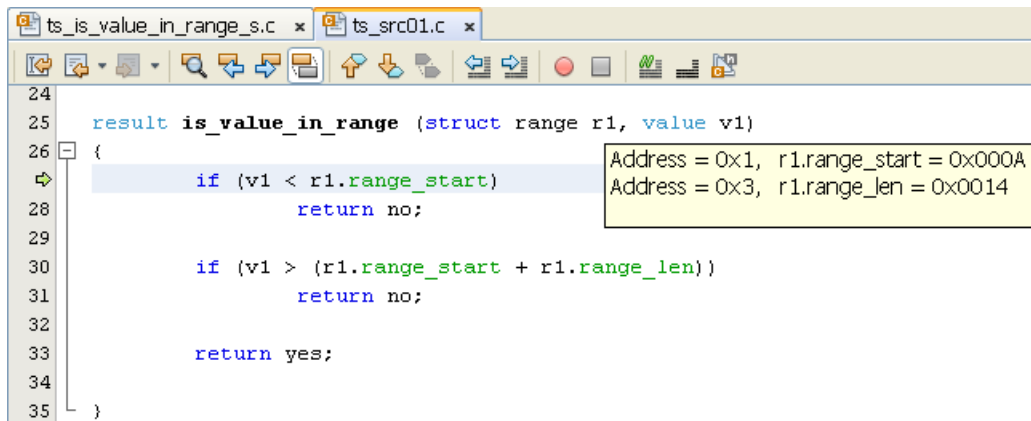
and step into function `tussy_execute_task()`.



```

173
174 void tussy_execute_task(void)
175 {
176     if (ts_mode == TS_MODE_TESTSTEP) {
177         while (TS_REPEAT_COUNT) {
178             TS_TESTOBJECT_RETURN = is_value_in_range(r1, v1);
179             TS_REPEAT_COUNT--;
180         }
181     }
182 }
183
  
```

Finally step into your test object function.



```
ts_is_value_in_range_s.c x ts_src01.c x
24
25 result is_value_in_range (struct range r1, value v1)
26 {
27     if (v1 < r1.range_start)
28         return no;
29
30     if (v1 > (r1.range_start + r1.range_len))
31         return no;
32
33     return yes;
34
35 }
```

Debugger window:
Address = 0x1, r1.range_start = 0x000A
Address = 0x3, r1.range_len = 0x0014

6 Known Issues

6.1 PIC16 Memory

The PIC16 controller only provides very small RAM resources. The number and size of variables is restricted to the available memory banks minus the memory required for the test driver program. Thus, the linking of the test driver program could fail if instrumentation is enabled. If so, try to adjust your linker file and/or try to exclude unused code using defines.

6.2 XC32 Compiler

Currently long double type variables may cause a test run to suspend and/or may produce wrong results.

6.3 The target device is not ready for debugging

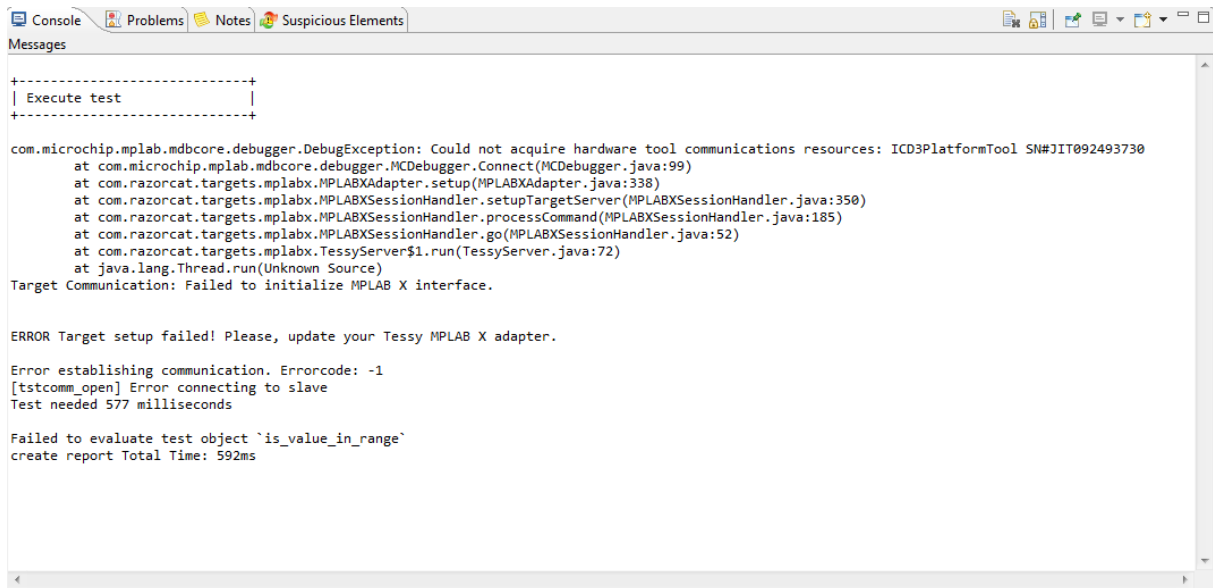
Please use an external power supply for your target hardware.

6.4 tthd: Java processes died unexpectedly, exiting...

The java class paths of the MPLAB X SDK may change from one MPLAB X version to another. Thus, TESSY provides different versions of the target handler script file as well, which you can find in the TESSY installation folder `sys\targets\mplab`. The files are called `target_handler-X.XX.sh` where X.XX denotes the MPLAB X version number nearest to your installed version. Adjust your TEE attribute **TESSYTHCmd** accordingly. The latest script version should correspond to the latest MPLAB X version. Nevertheless, if you do not find a matching script file, please, contact support@razorcat.com. Send us your MPLAB X version number, so that we can prepare the correct script file for your environment.

6.5 MPLAB X Version > 2.15: Subsequent Test Runs Fail

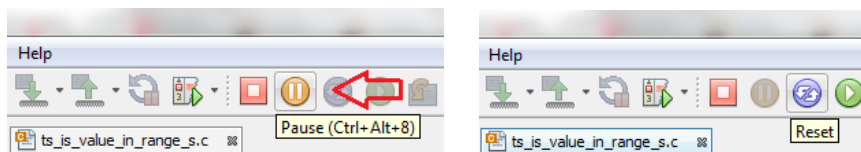
If you encounter the following error message after executing the first test you may create the Boolean type TEE attribute **Quit Target Handler** and set it to `true`.



This will quit TESSY’s MPLAB X target handler after each test run and thus it will be automatically restarted before the next test run.

6.6 Problem with interactive debugging

It may happen that a breakpoint cannot be set and the debugger does not stop at main. Click **Pause** and **Reset**. Now step manually until `tessy_execute_task()` is reached. Step into the function and step into `TESSY_TestobjectCall()` which contains your test object call.



```

195 void TESSY_TestobjectCall(void) {
196     TS_TESTOBJECT_RETURN = is_value_in_range(TESSY_TESTOBJECT_PARM_01, TESSY_TESTOBJECT_PARM_02);
197 }
    
```