­­­

|  |  |
| --- | --- |
| CEC1702 MPLABX efuse programming Guide | |
| User manual | |
| Rev 0.2 | 23rd March 2018 |

MICROCHIP CONFIDENTIAL

|  |
| --- |
| Copyright © 2018 Microchip or its subsidiaries. All rights reserved.  The information contained herein is confidential and proprietary to Microchip, shall be used solely in accordance with the agreement pursuant to which it is provided, and shall not be reproduced or disclosed to others without the prior written consent of Microchip. Although the information is believed to be accurate, no responsibility is assumed for inaccuracies. Microchip reserves the right to make changes to this document and to specifications and product descriptions at any time without notice. Neither the provision of this information nor the sale of the described semiconductor devices conveys any licenses under any patent rights or other intellectual property rights of Microchip or others. The product may contain design defects or errors known as anomalies, including but not necessarily limited to any which may be identified in this document, which may cause the product to deviate from published specifications. Microchip products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an officer of Microchip will be fully at the risk of the customer.    MICROCHIP DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE. IN NO EVENT SHALL MICROCHIP BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF MICROCHIP OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. |

Table of Contents

[1 Introduction 3](#_Toc509573766)

[1.1 Purpose 3](#_Toc509573767)

[1.2 Scope 3](#_Toc509573768)

[1.3 References 3](#_Toc509573769)

[1.4 Glossary of Terms and Acronyms 3](#_Toc509573770)

[1.5 Assumptions 3](#_Toc509573771)

[2 Tools & HW 4](#_Toc509573772)

[3 Efuse programming procedure 5](#_Toc509573773)

# Introduction

## Purpose

This document provides step by step user instructions for programming efuse memory using MPLABX IDE and ICD4 debugger

## Scope

## References

1. Efuse generator tool documentation
2. [CEC1x02 DevBoard](http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=dm990013)

## Glossary of Terms and Acronyms

## Assumptions

It is assumed that user is familiar with efuse generator tool or has access to the efuse generator tool and corresponding documentation

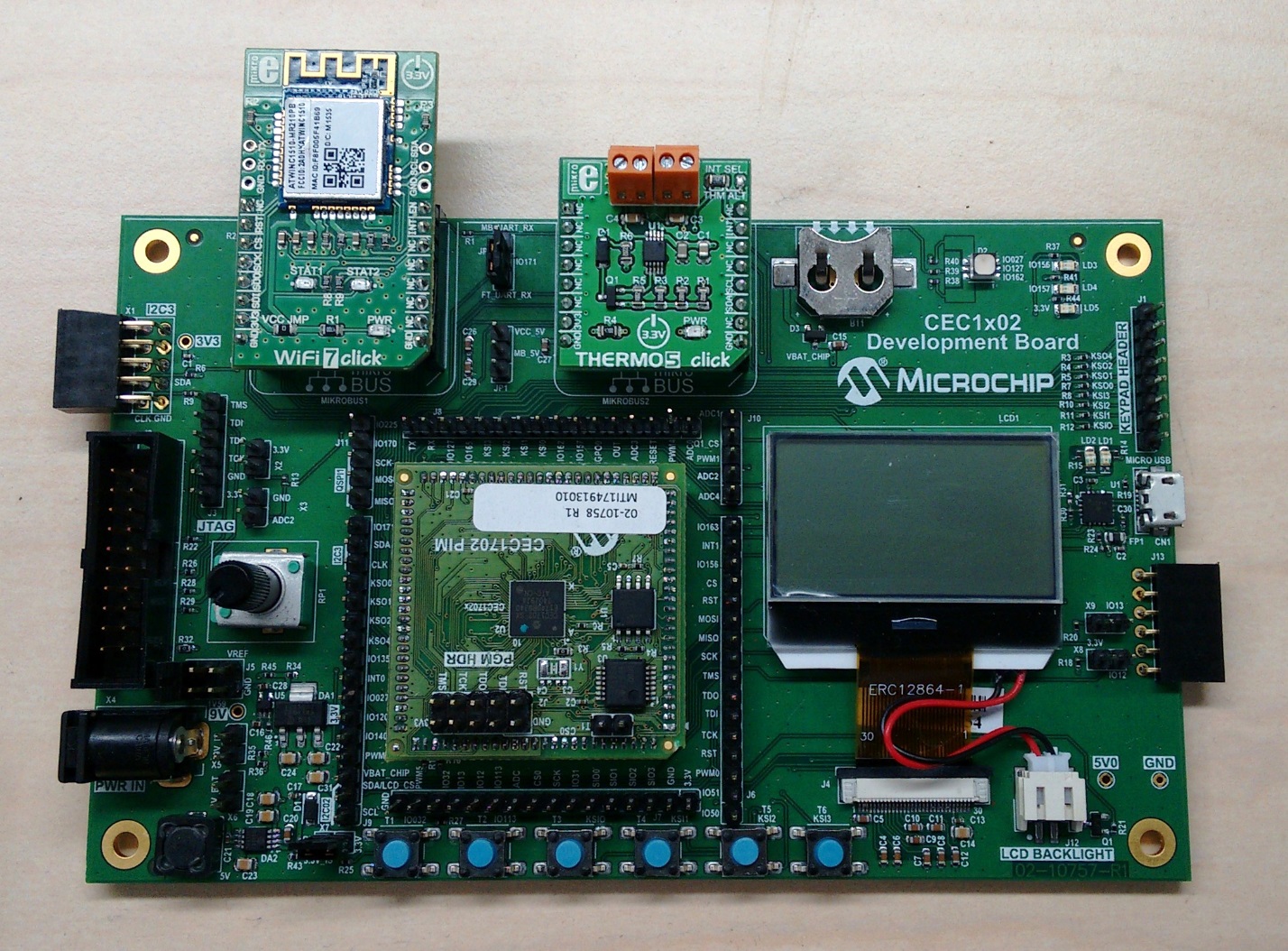
# Tools & HW

Following tools and utilities should be installed and available for accomplishing efuse programming

* Efuse\_generator utility v8.01
* MPLABX IDE
* ICD4 debugger with JTAG cable
* CEC1x02 Development board with USB cable

# Efuse programming procedure

Ensure the CEC1x02 development board is powered and ICD4 is connected to board using JTAG interface



J5

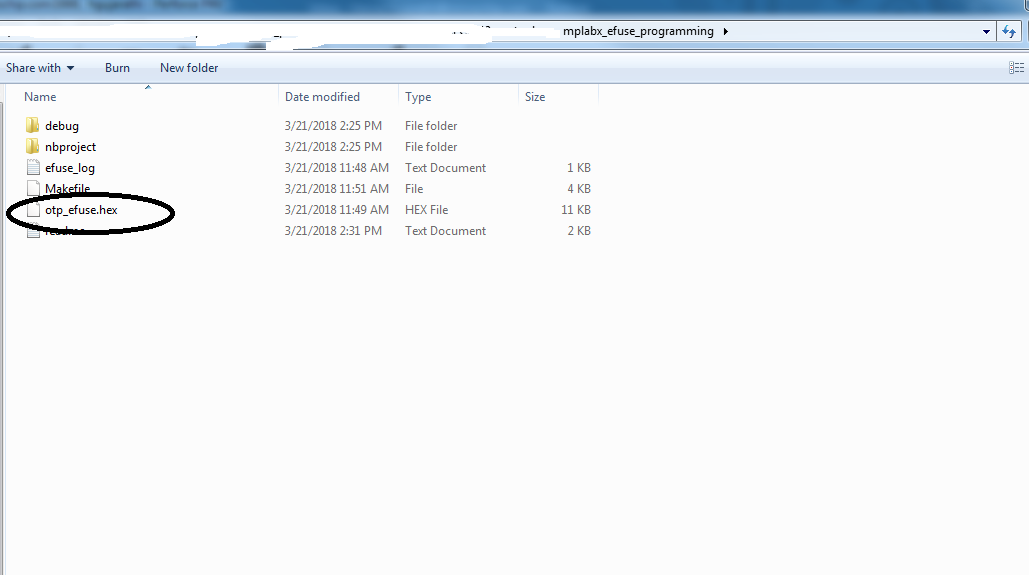
JTAG

Short pin 3 & 4 on jumper J5 to provide reference voltage for efuse programming. Please refer board schematic for more details

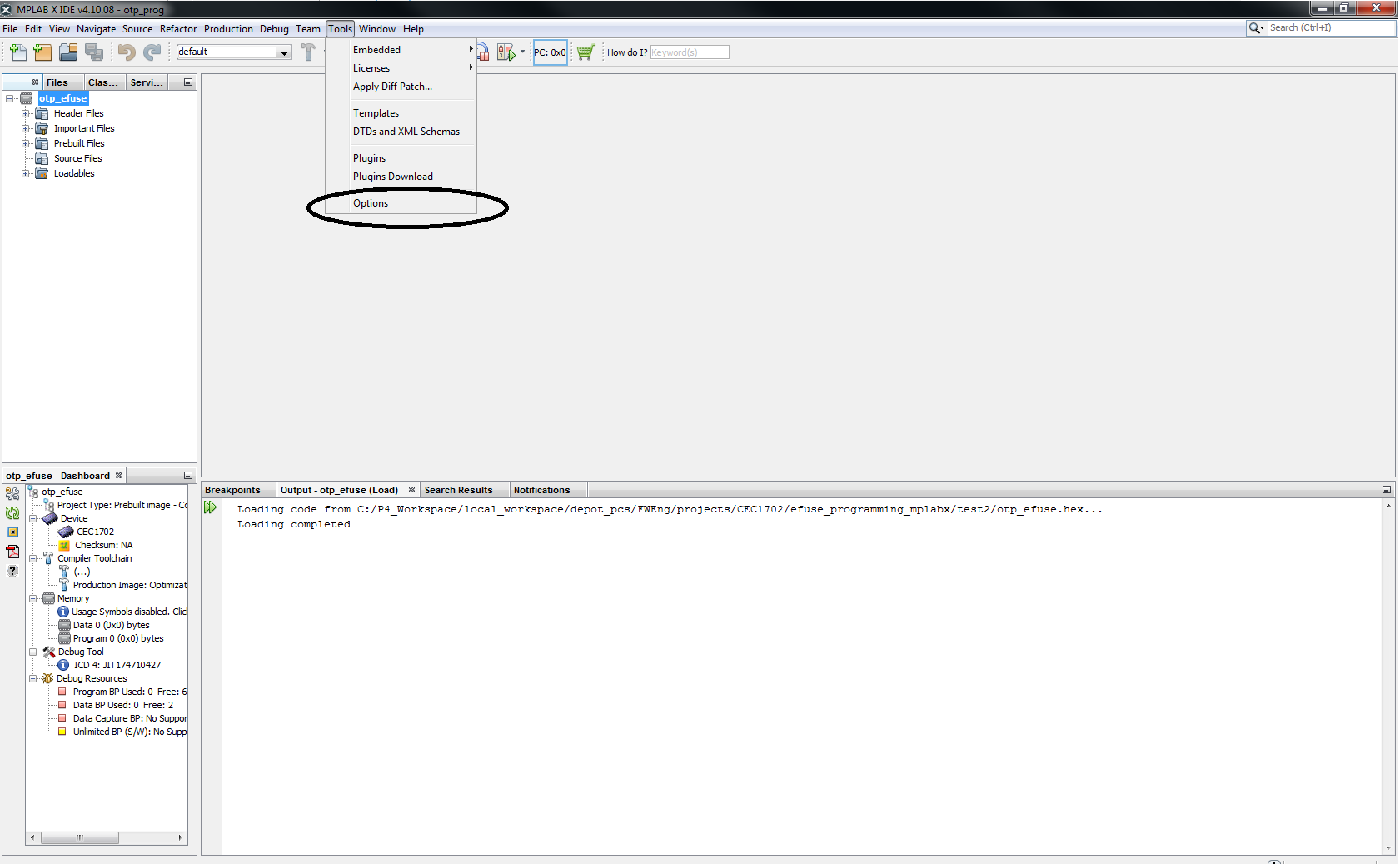


Please refer to efuse generator user guide for steps to generate efuse memory map output hex file (otp\_prog.hex)

1. Replace the otp\_efuse.hex file in mplabx\_efuse\_programming folder with new otp\_efuse.hex generated as per your requirement



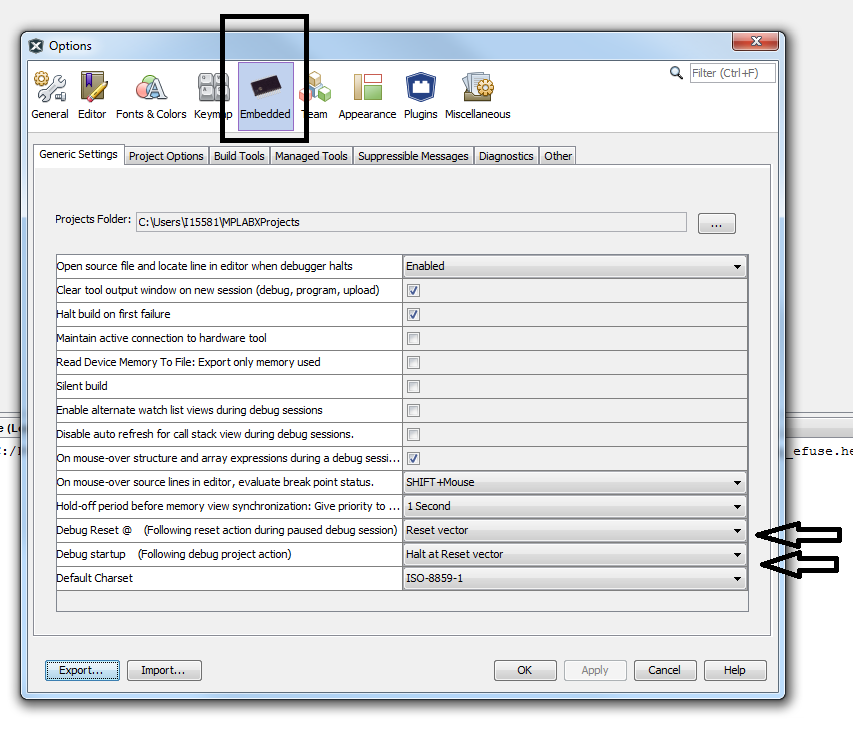
1. Start MPLABX IDE and select Tools -> Options



1. In the options tab select “Embedded”. Under *General Settings* tab

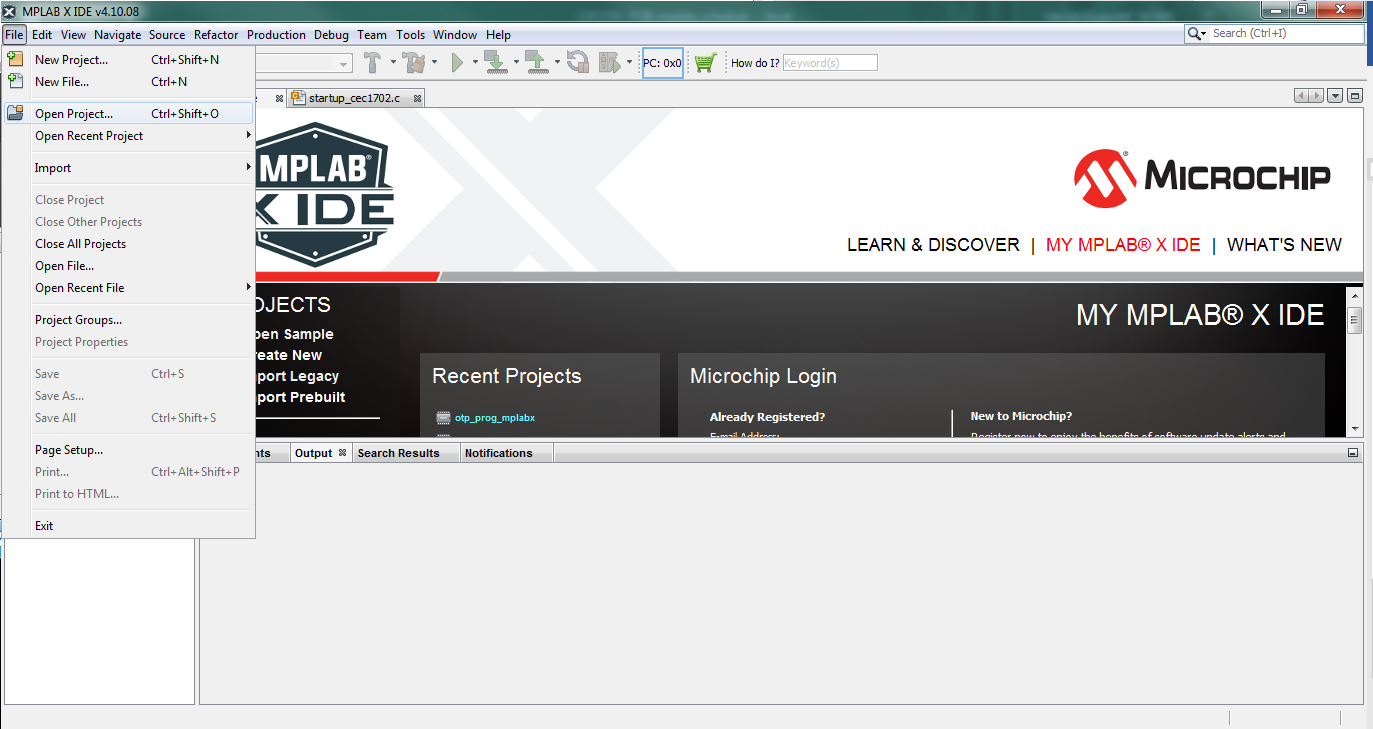
Make sure “Debug Reset” option is set to “Reset Vector”

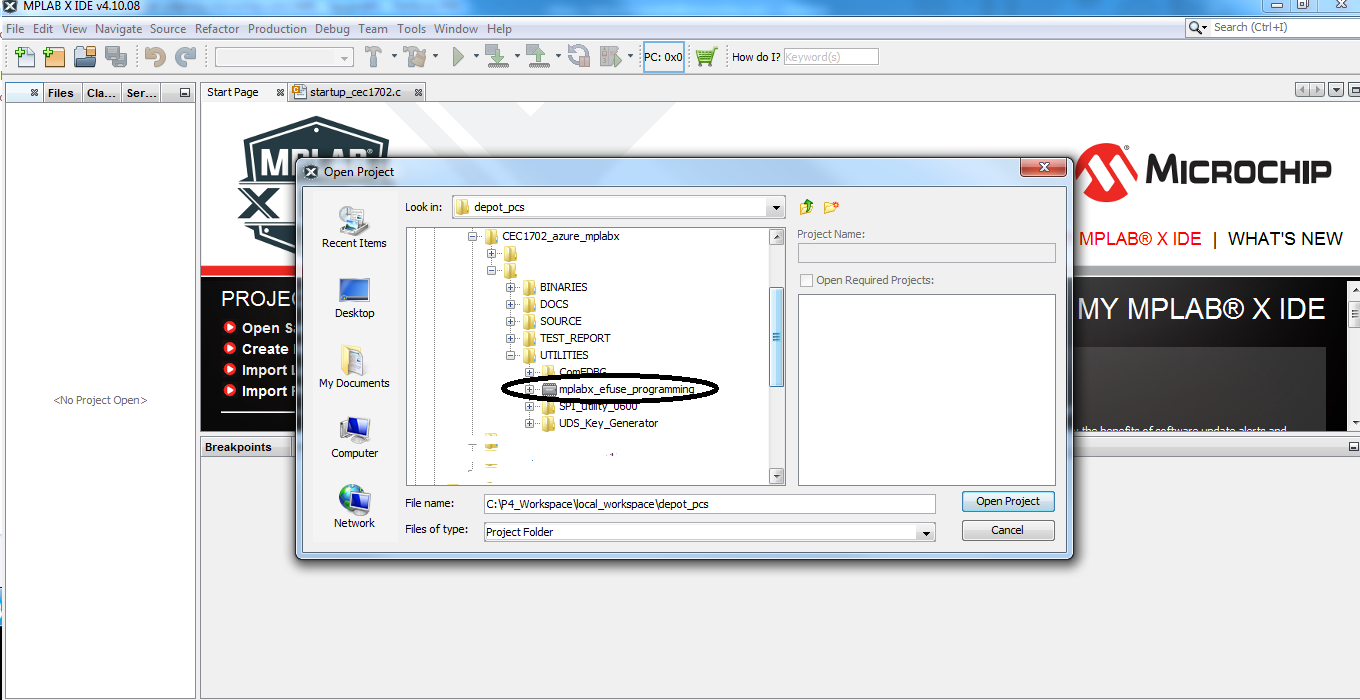
Make sure “Debug Startup” option is set to “Halt at Reset Vector”



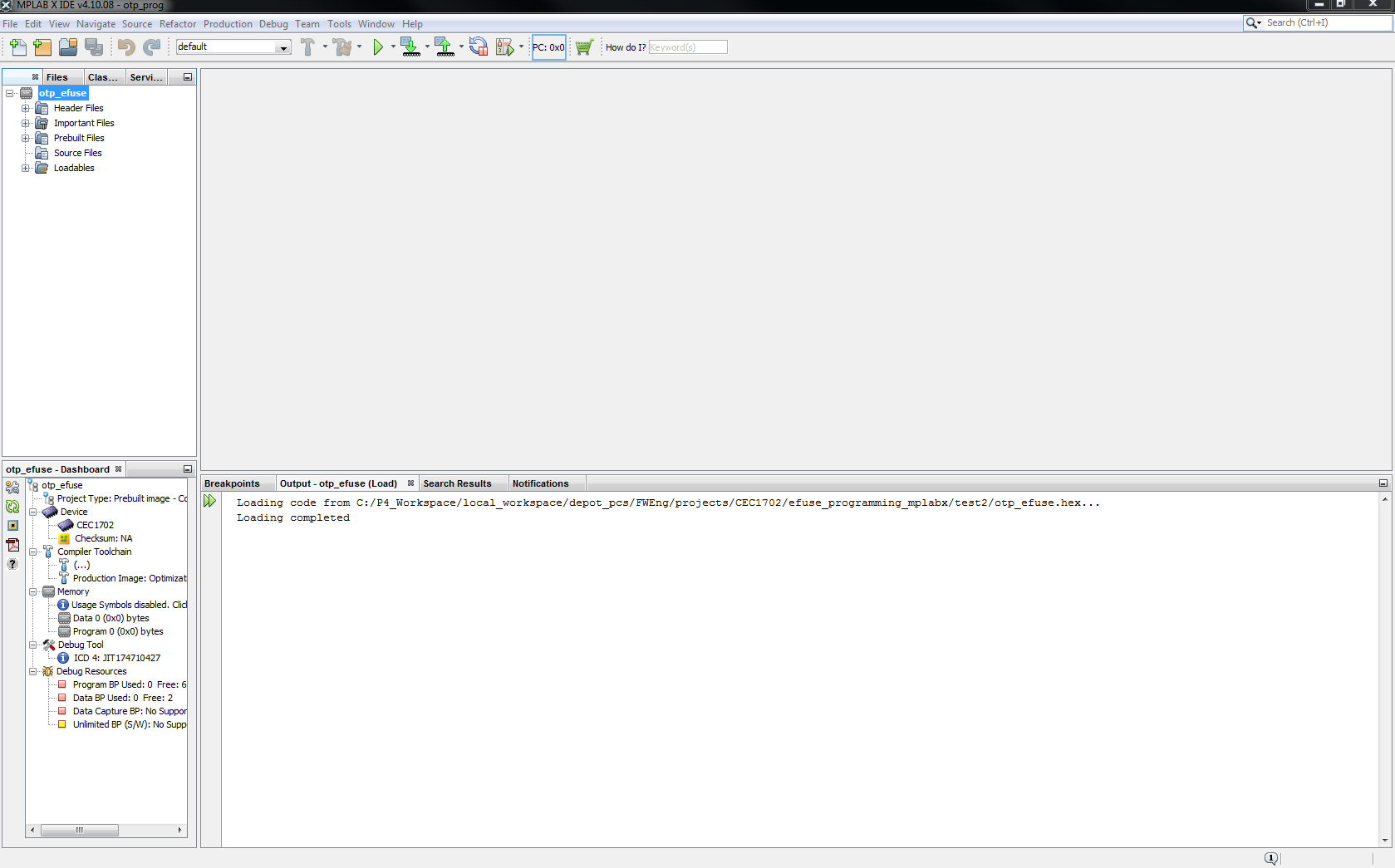
Click Ok and option window will be closed

1. Select File -> Open project and Open the mplabx\_efuse\_programming project from UTILITIES folder

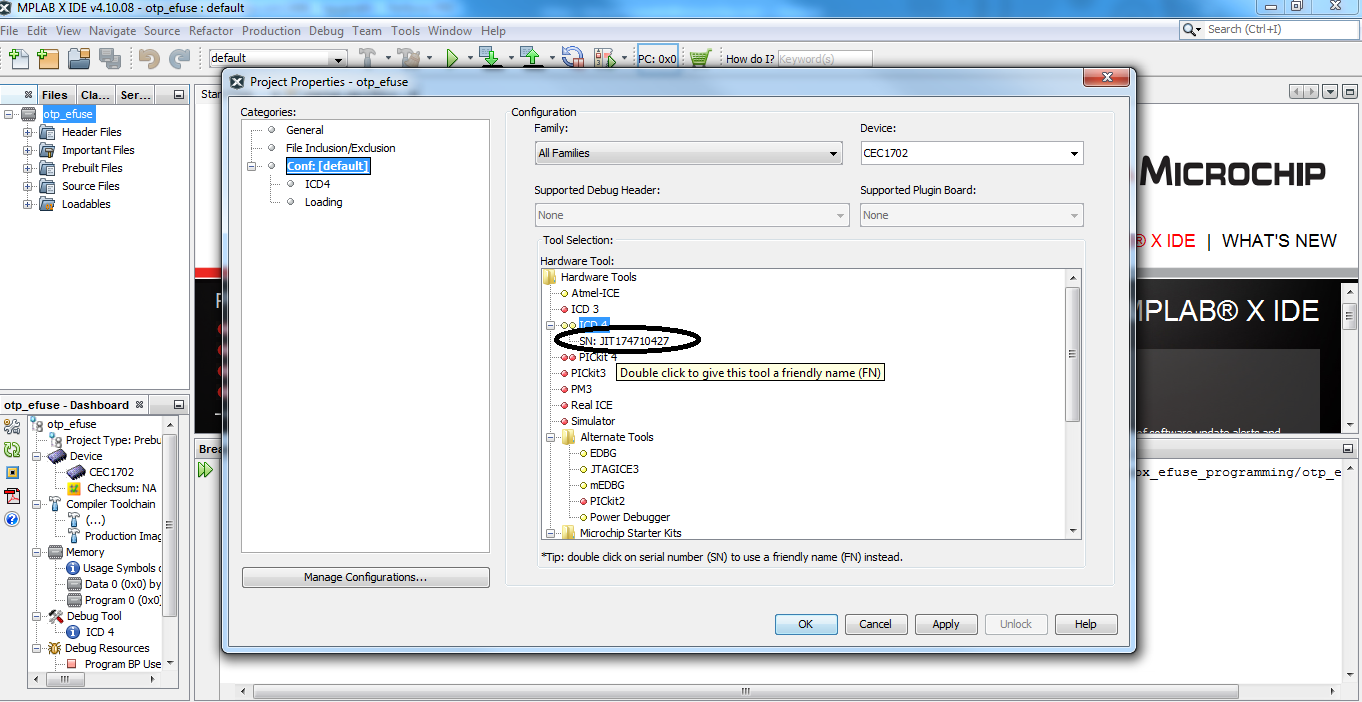




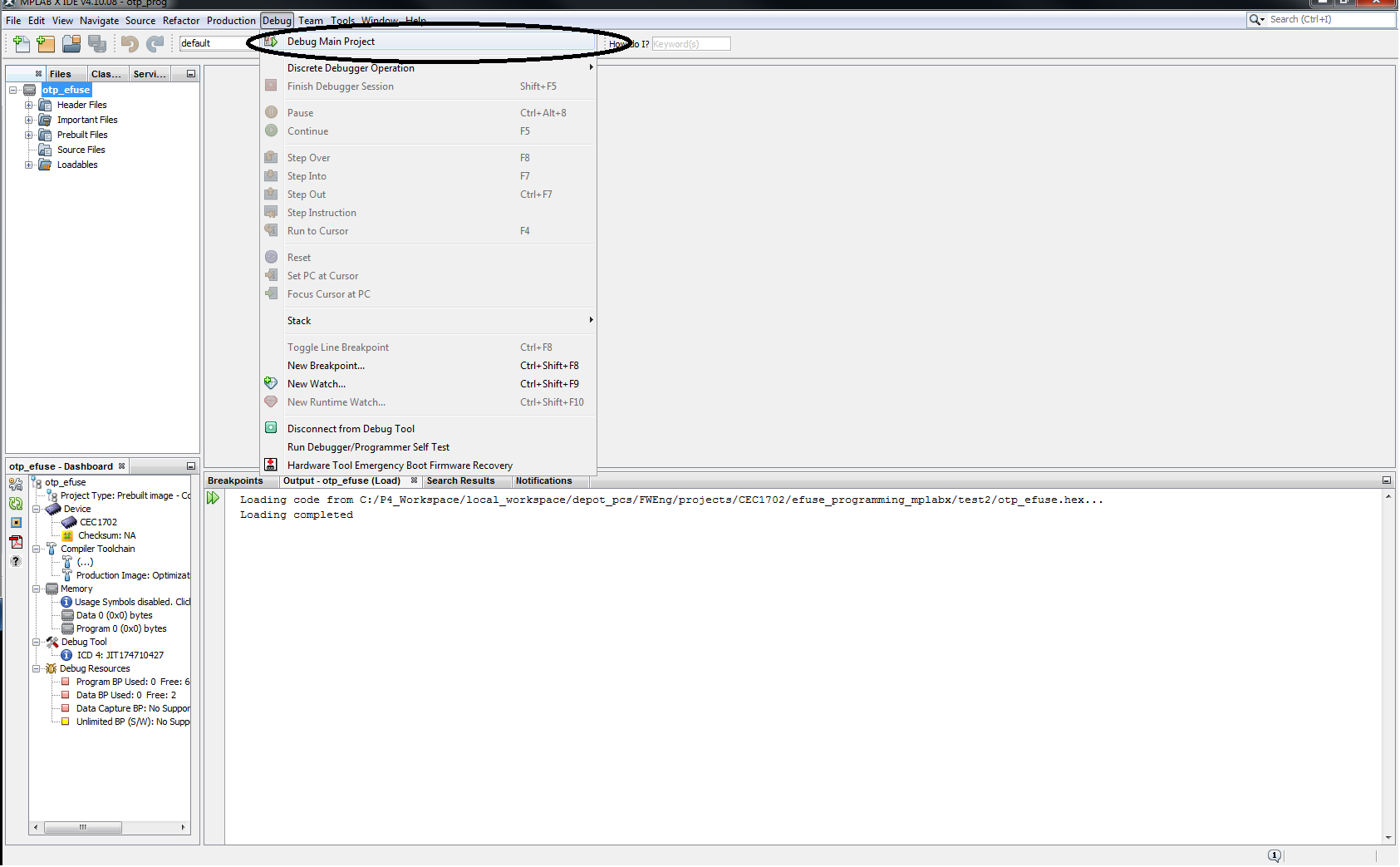
Once loaded you should see following



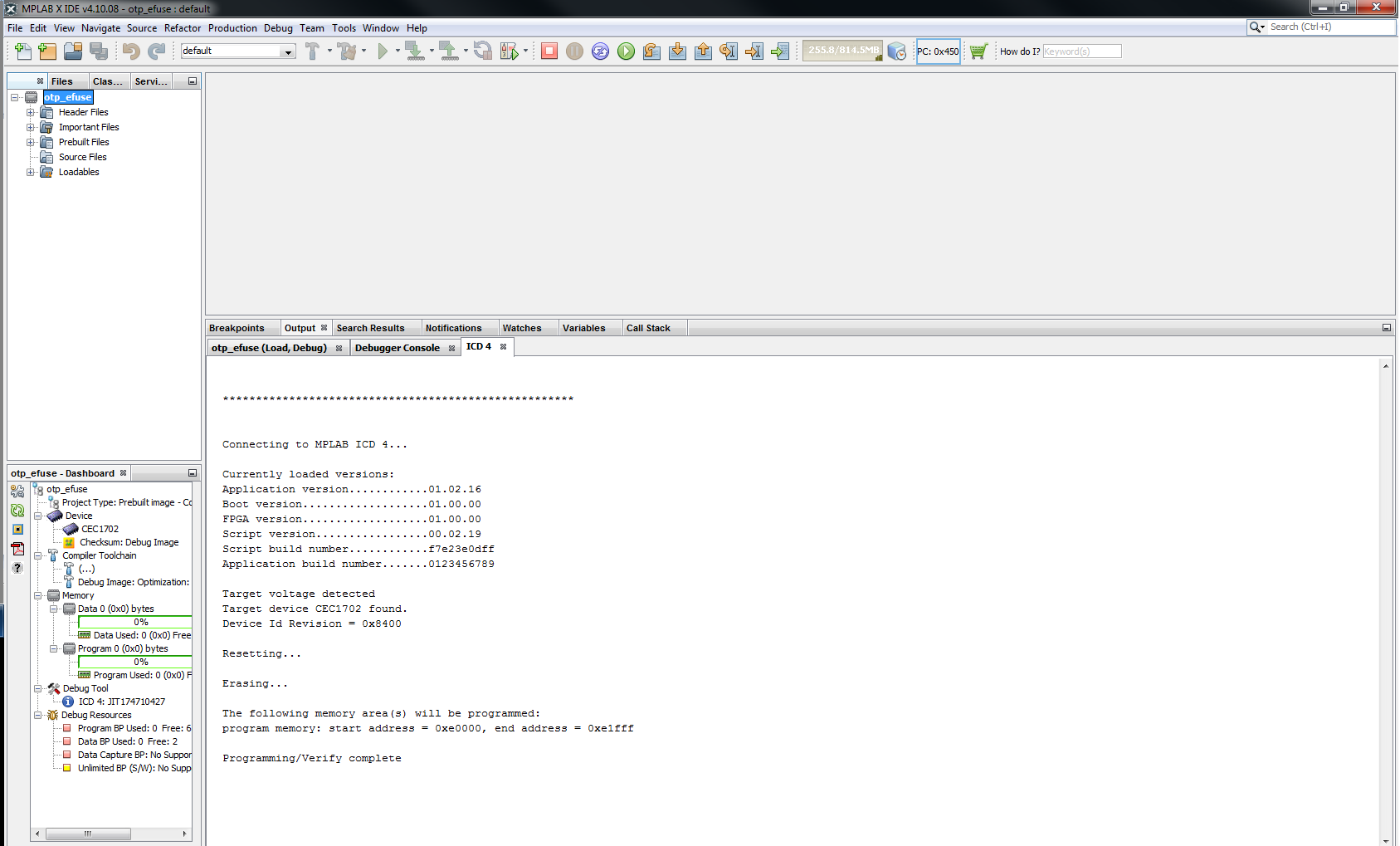
1. Ensure ICD4 is connected and detected. Go to project properties as Right Click otp\_efuse -> Properties



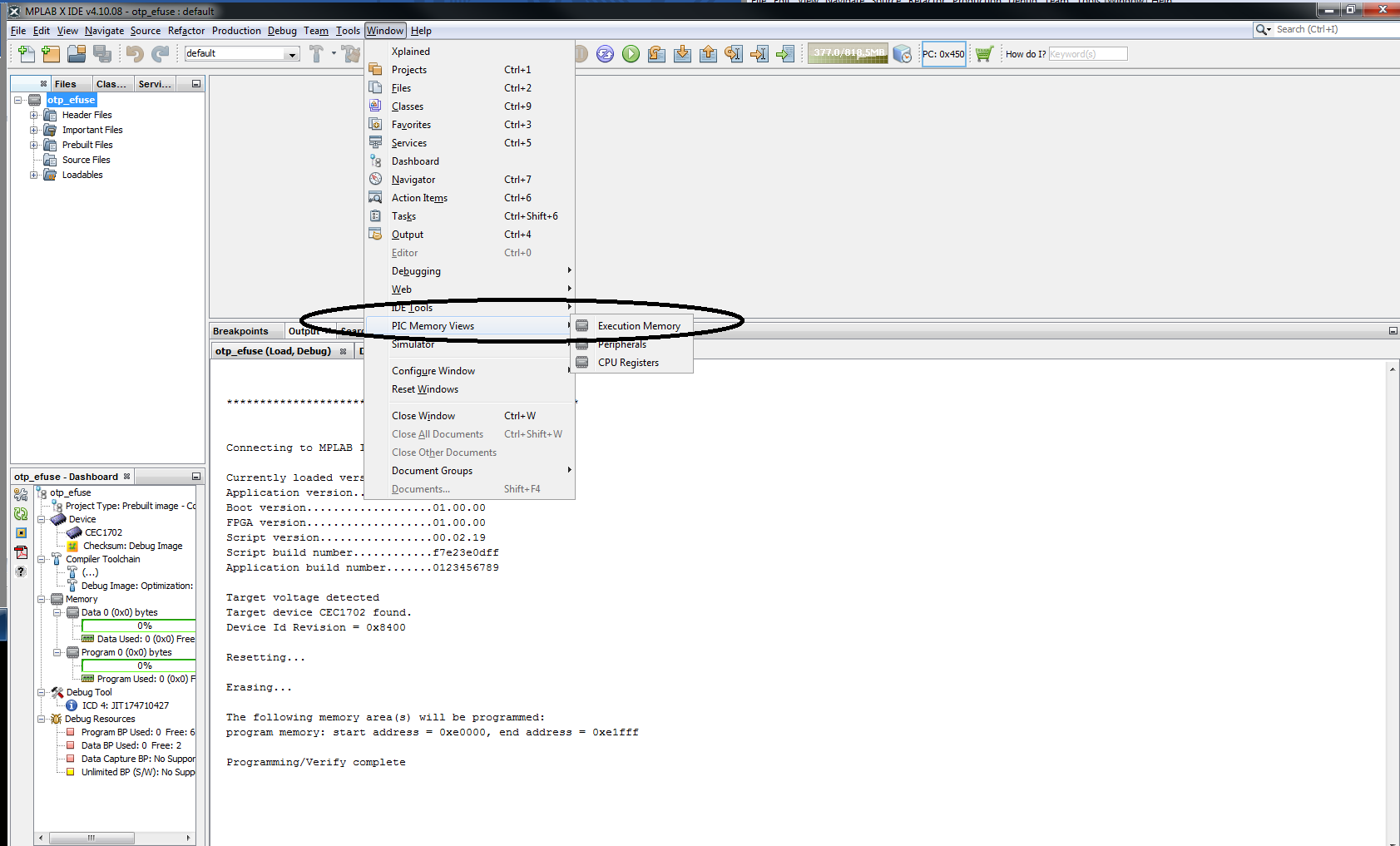
1. Debug and select Debug Main project.



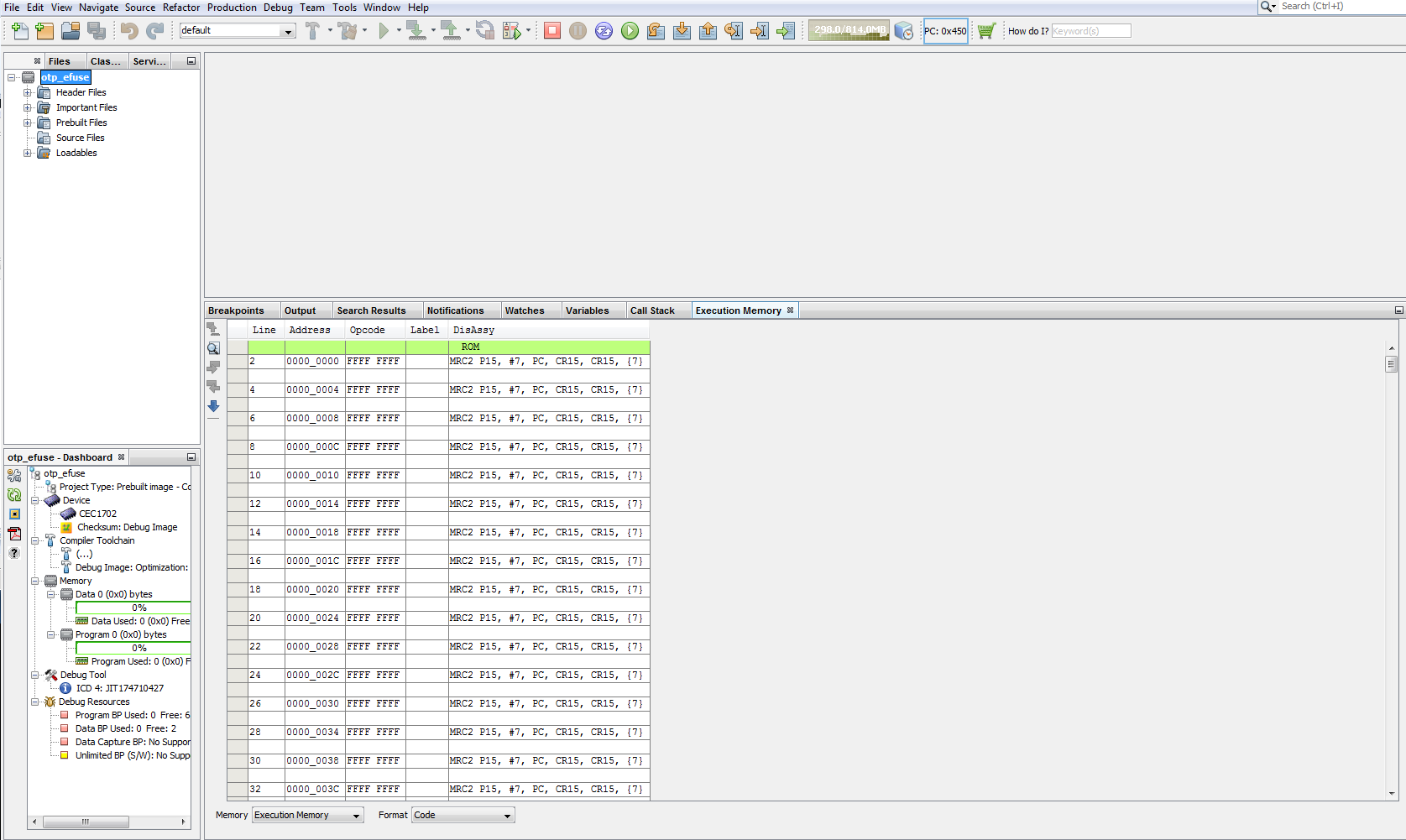
The device will be programmed and following message will be displayed



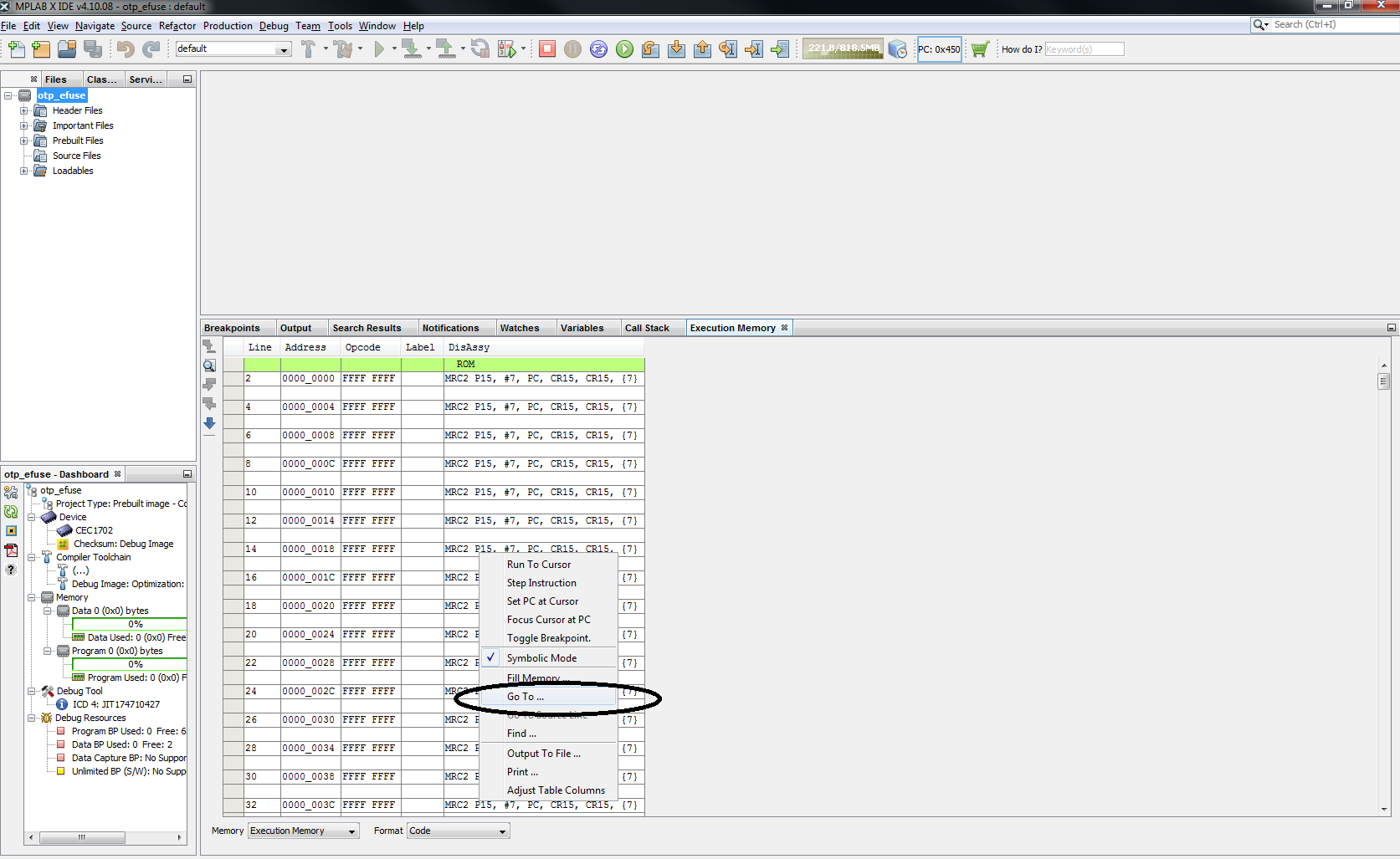
1. Next open the Execution memory window from Window->PIC Memory Views -> Execution Memory



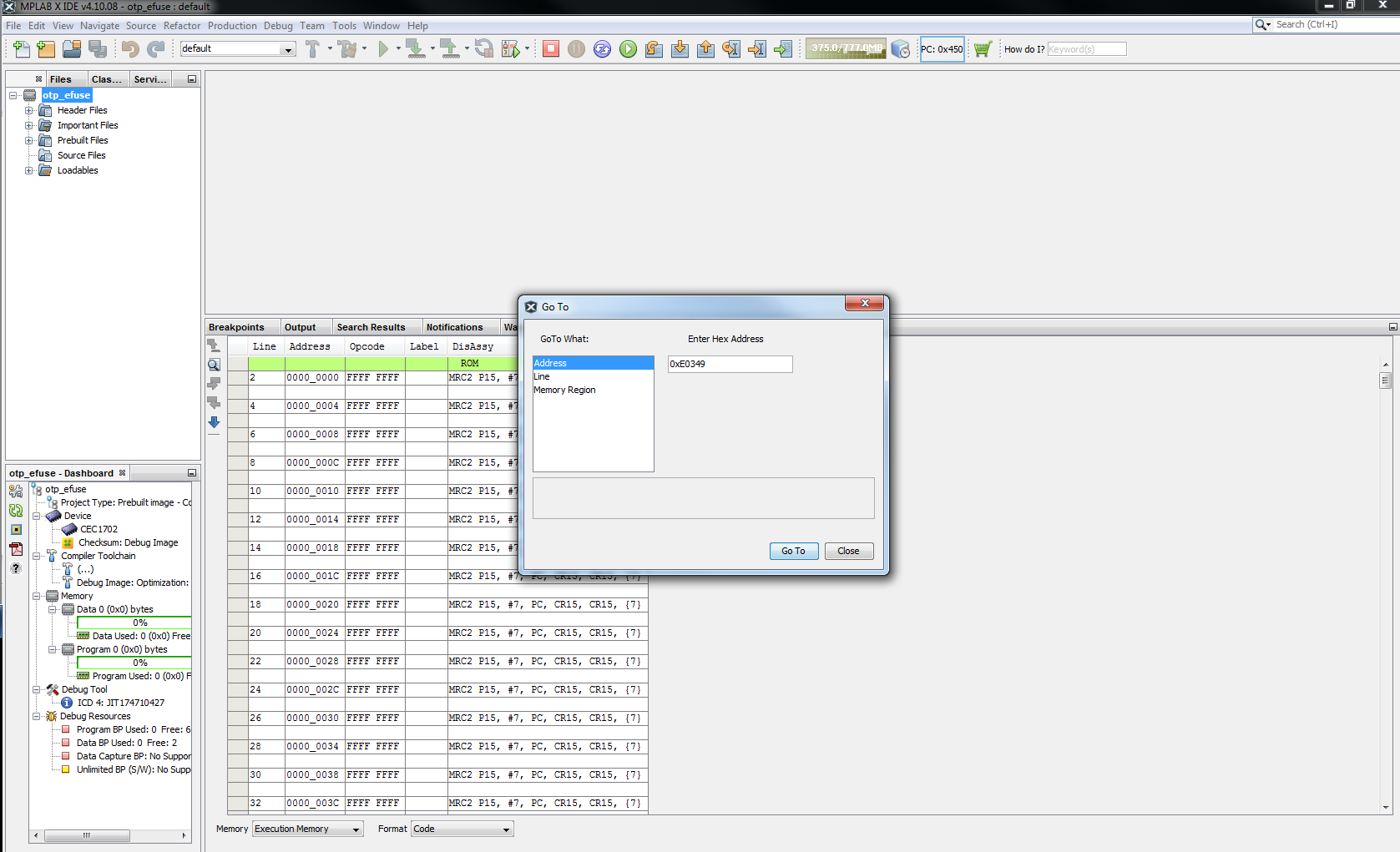
This shall display the Execution memory as below



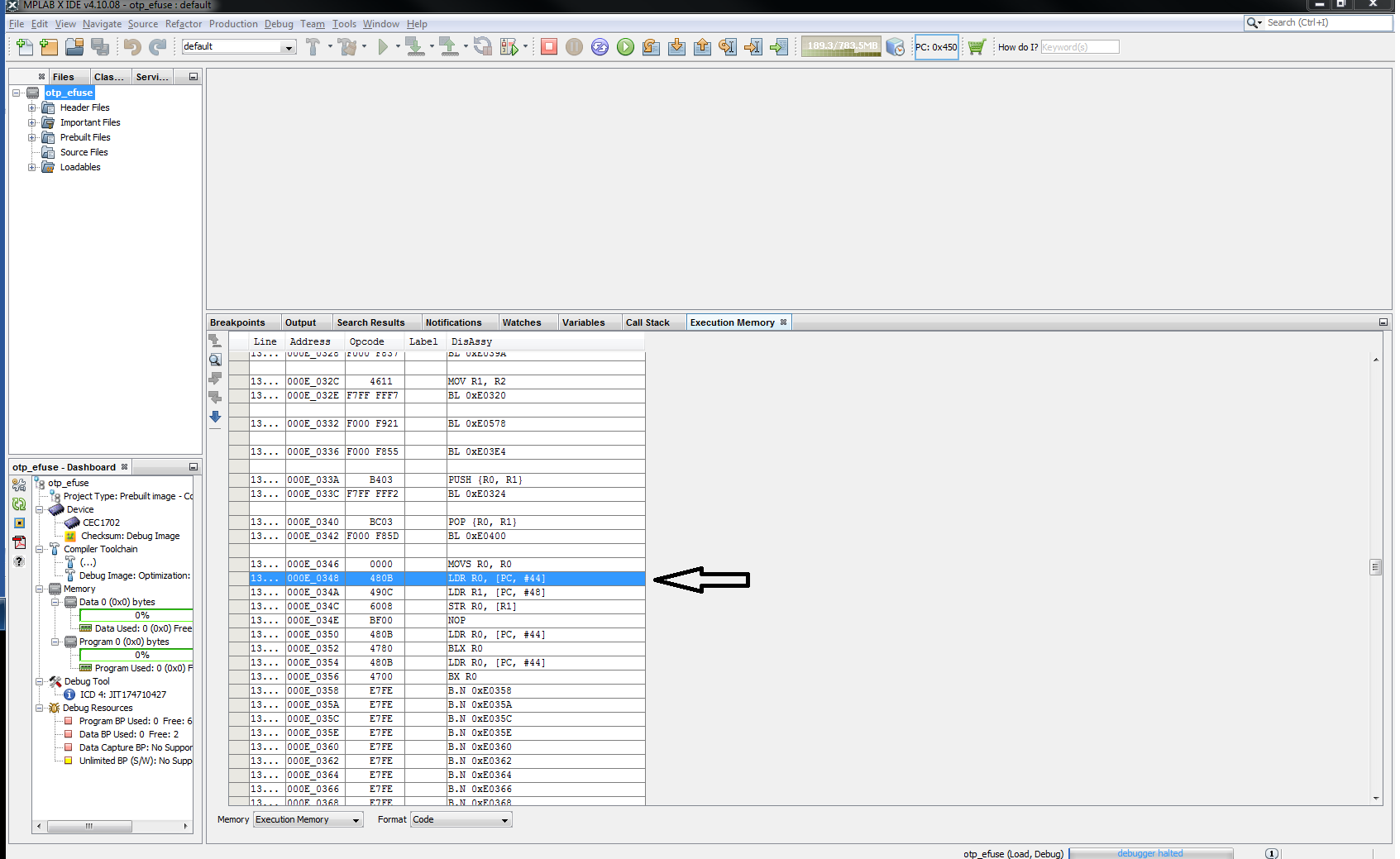
1. Right click on the memory window and select “Go to” option



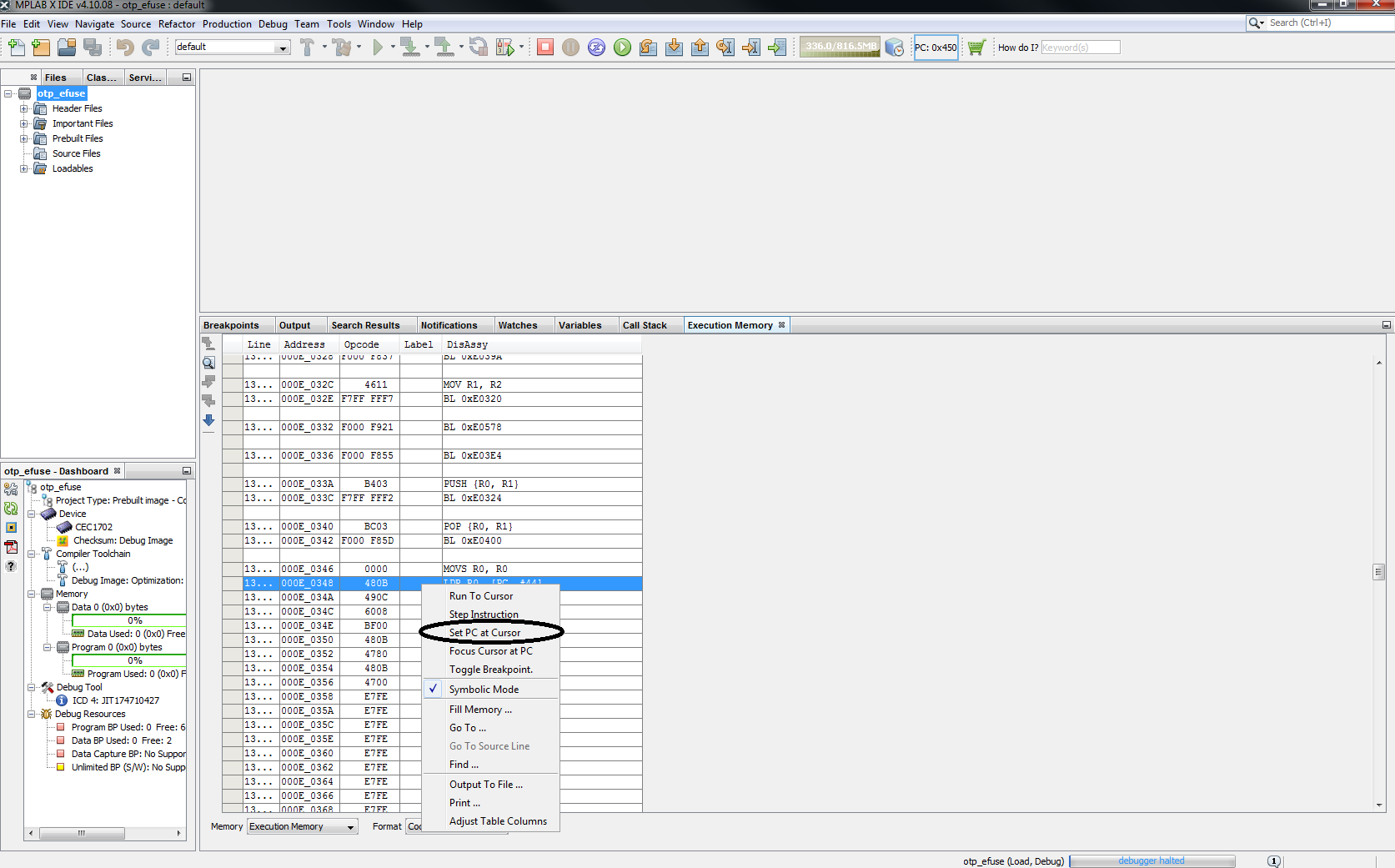
1. Select “Address” field and enter following hex address **0xE0349**



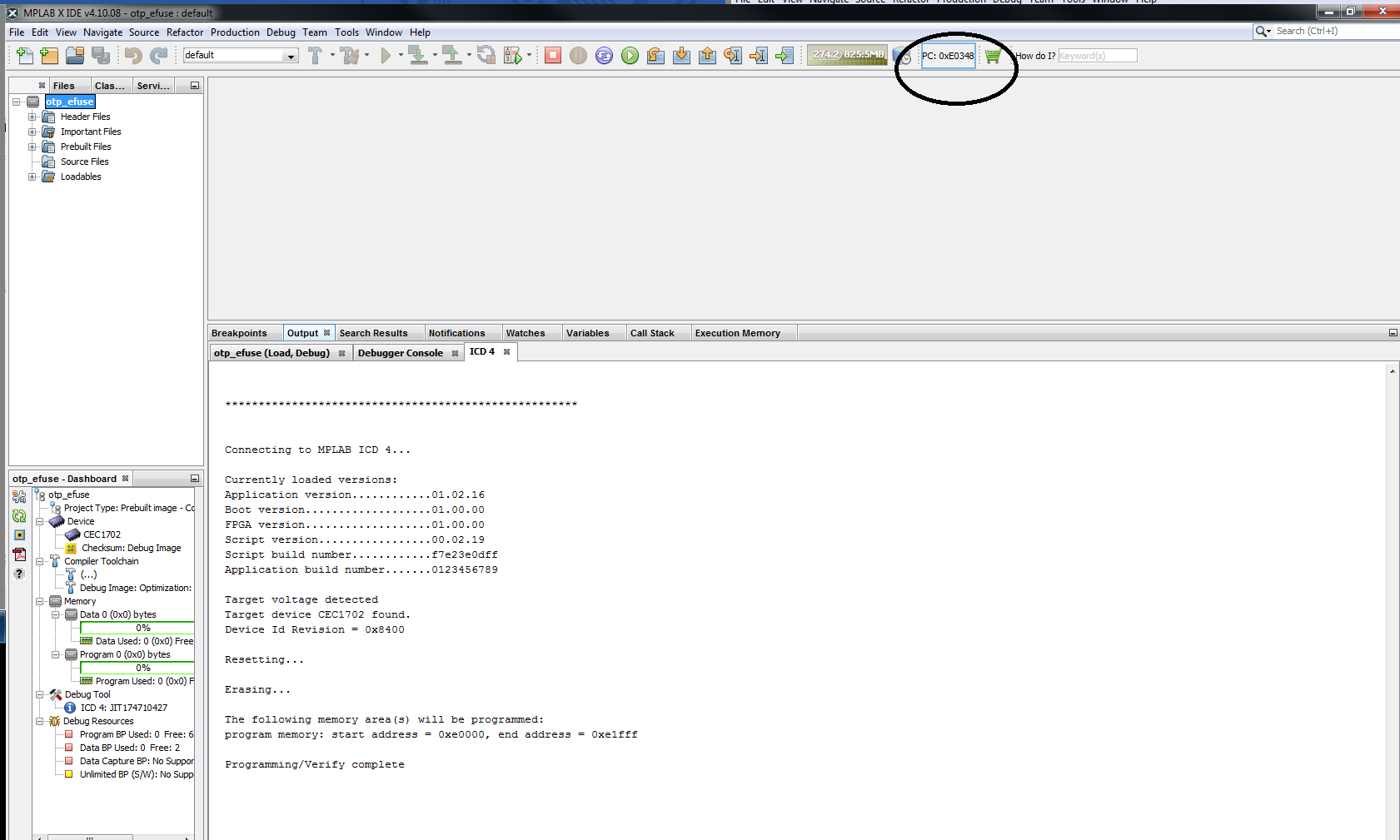
1. Click Go To and then click close. This shall update the Execution memory to display the code at address 000E\_0348



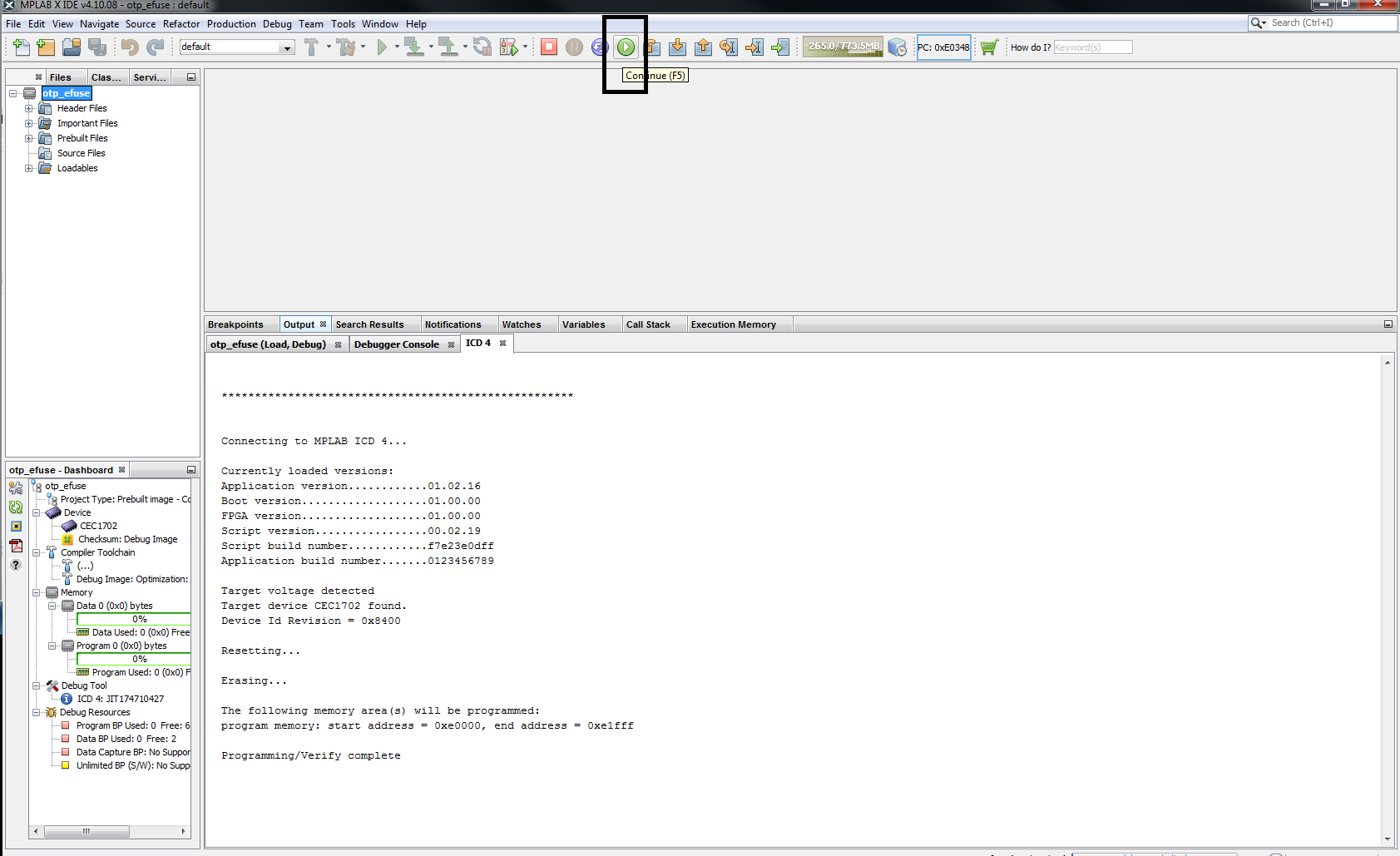
1. Select the highlighted line (with address 000E\_0348) and right click. Select option “Set PC at cursor”



1. This shall update the PC value to 0xE0348 as shown below



1. Now click “Continue” to run the code and program the efuse



Wait for ~10 seconds for efuse programming to complete. If the programming is successful LD3 will be OFF. If there is any failure during efuse programming LD3 and LD4 LED’s on the development board will blink alternatively.

You can remove the jumper on pin 3-4 on J5 and power cycle board