

AP16102

XC164CM

XC164CM "Cookery-Book" for a "Hello world" application

Microcontrollers



Never stop thinking

Edition 2008-07-16

**Published by
Infineon Technologies AG
81726 München, Germany**

**© Infineon Technologies AG 2008.
All Rights Reserved.**

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. 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) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

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

Warnings

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

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Note: Table of Contents [see page 6.](#)

Introduction:

This “Appnote” is an Infineon Hands-On-Training.

It will help inexperienced users to get an XC164CM Evaluation-Board / Starter-Kit-Board up and running.

With this Hands-On-Training / Cookery-Book / step-by-step-book you should be able to get your first useful program in less than 2 hours.

The purpose of this document is to gain know-how of the microcontroller and the tool-chain.

Additionally, the "hello-world-example" can easily be expanded to your needs.

You can connect either a part of - or your entire application to the Starter-Kit-Board.

You are also able to benchmark any of your algorithms to find out if the selected microcontroller fulfils all the required functions within the time frame needed.

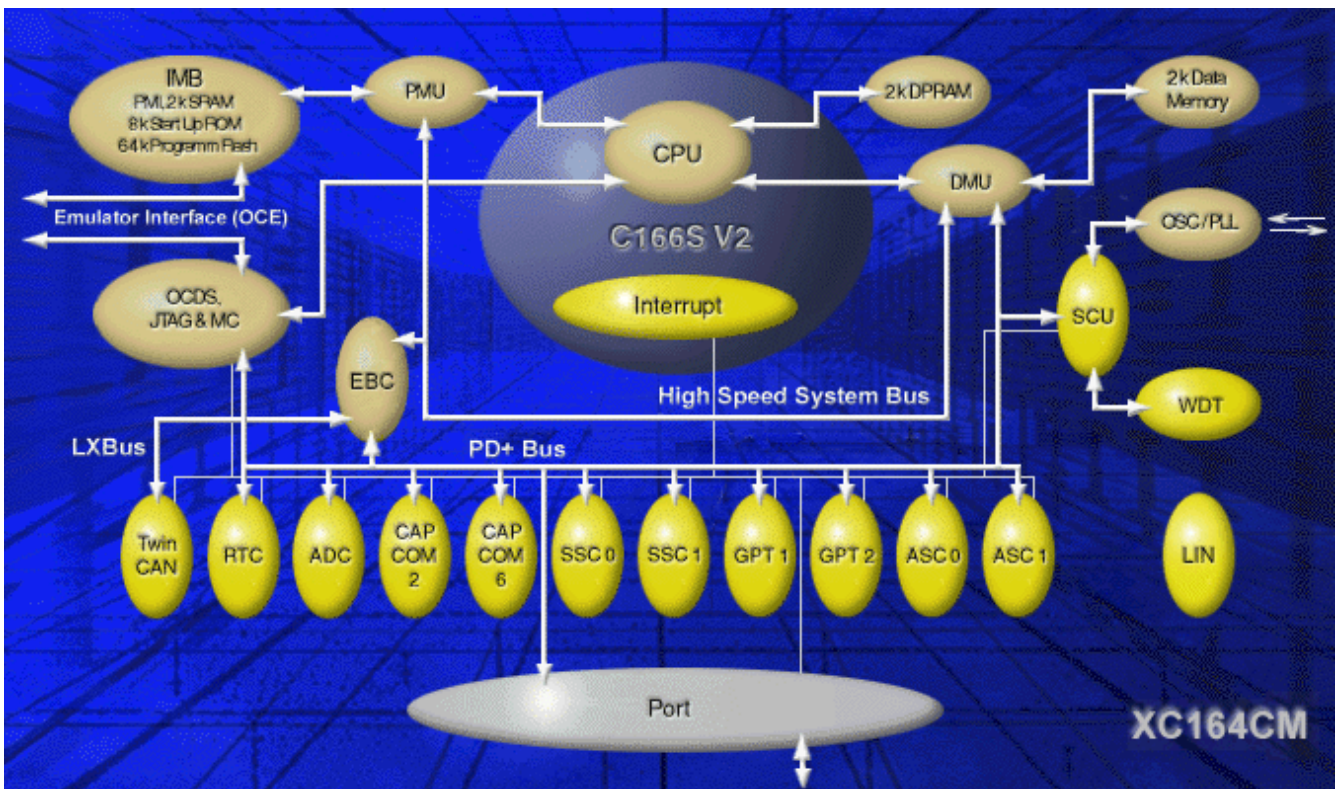
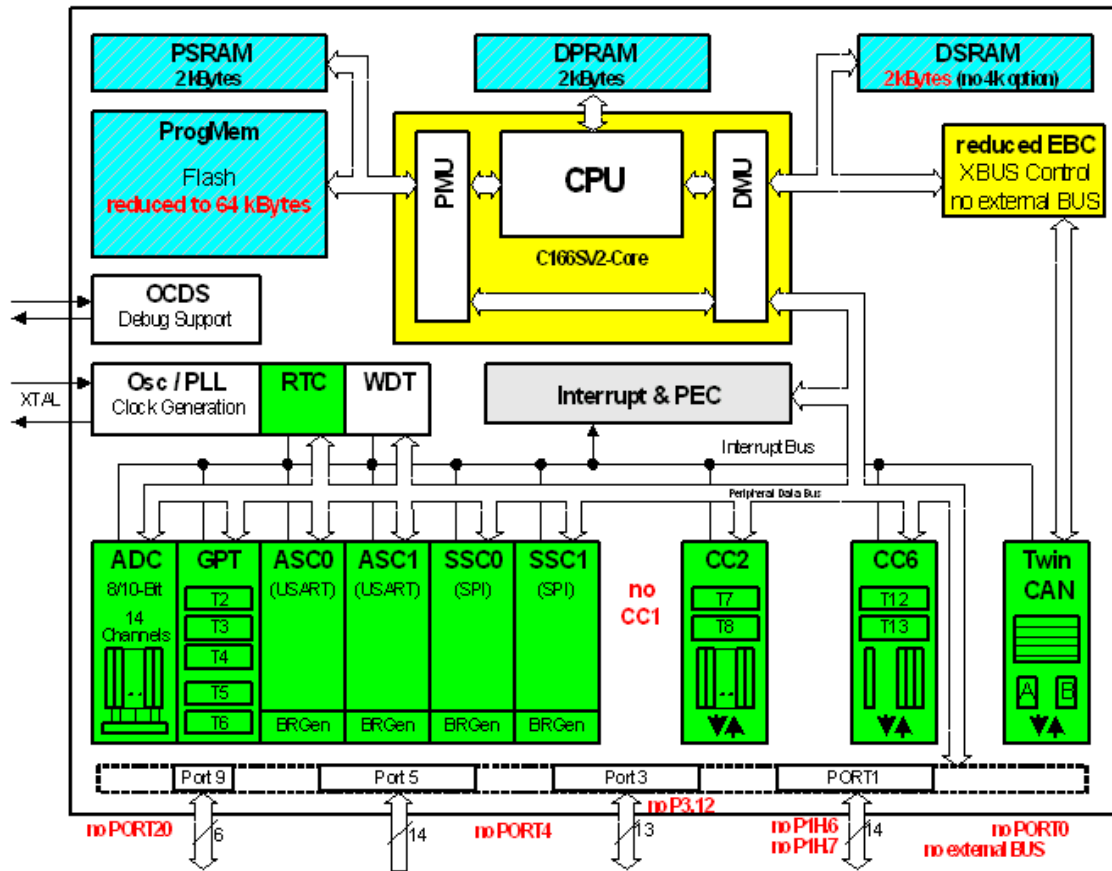
Note:

The style used in this document focuses on working through this material as fast and easily as possible. That means there are full screenshots instead of dialog-window-screenshots; extensive use of colours and page breaks; and listed source-code is not formatted to ease copy & paste.

Have fun and enjoy the XC164CM!

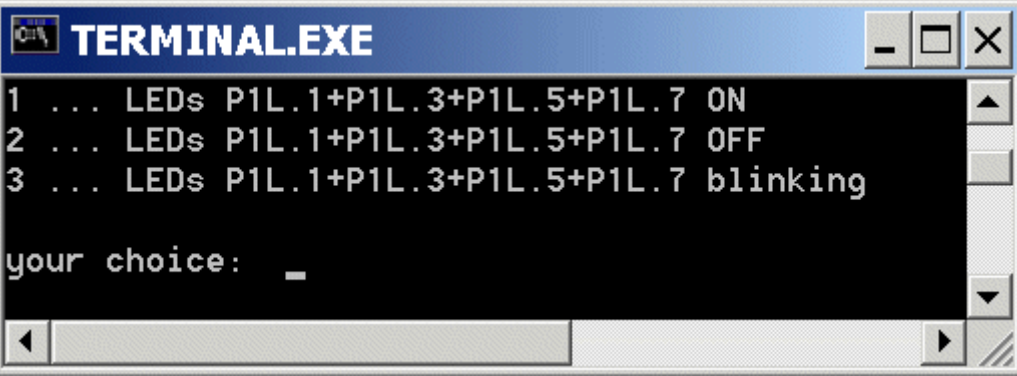


XC164CM Block Diagram

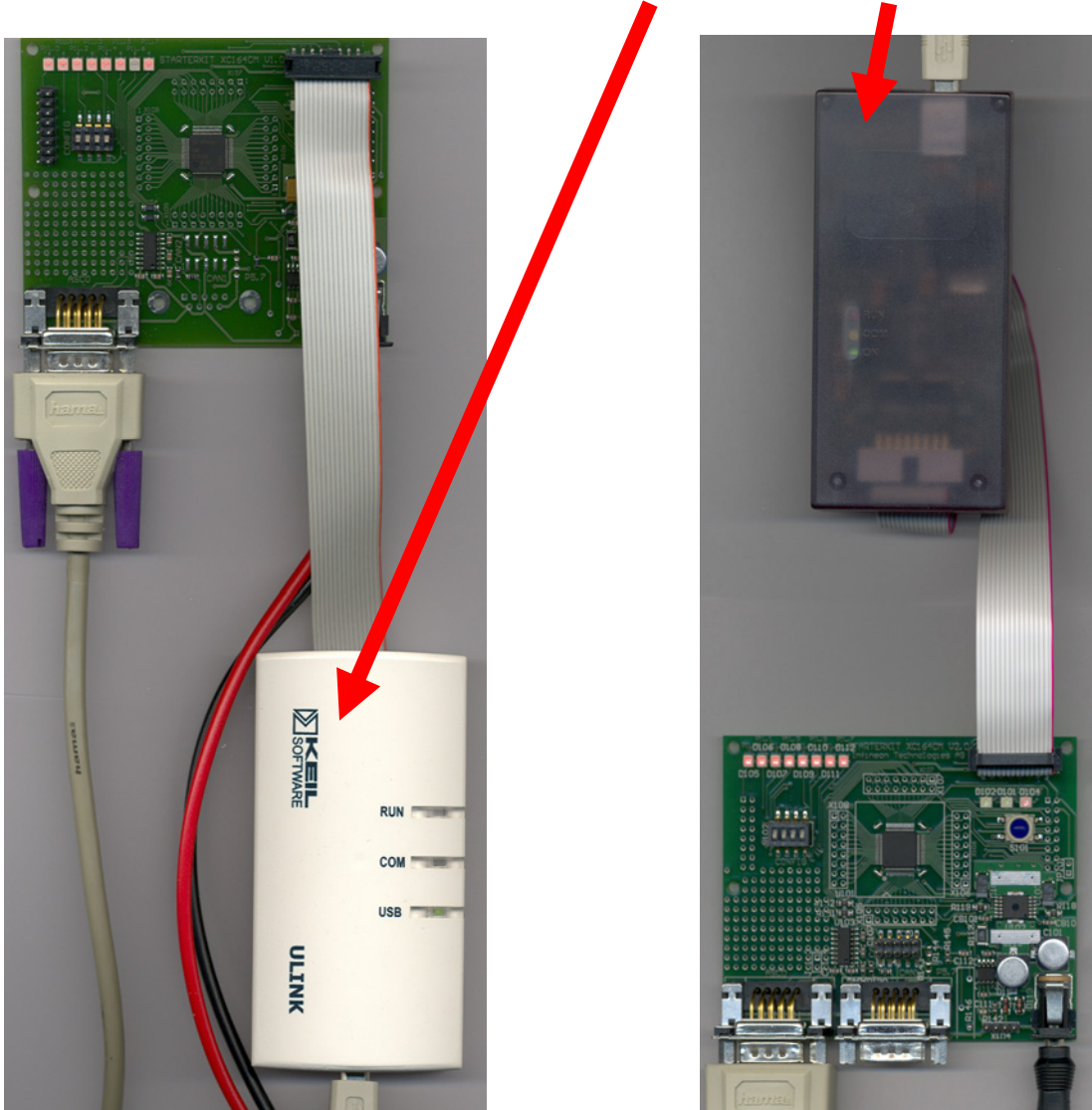


“Cookery-book“

For your first programming example for the XC164CM Starter-Kit-Board:

<p>Your program:</p>	
<p>Chapter/ Step:</p>	<p style="text-align: center;">*** Recipes ***</p>
<p>1.)</p>	<p>XC164CM Starter Kit Board Power Supply, Jumper Setting, Serial cable to the notebook</p>
<p>2.)</p>	<p>DAvE program generator DAvE installation (mothersystem) + DAVe Update (XC164CM.DIP) for XC164CM</p>
<p>3.)</p>	<p>Using DAVe Microcontroller initialization for your programming example</p>
<p>4.)</p>	<p>Using the KEIL Development Tools (C/EC++ Compiler) Programming of your application (XC164CM) with KEIL tool chain (µVision3) - Compiler V6.04a + first steps with the Simulator</p>

Additional exercises

5.)	OnChipFlash-Programming with Bootstrap Loader Mode (ASC0) + MEMTOOL
6.)	OnChipFlash-Programming and OCDS via Keil-ULINK or USB-JTAG-Wiggler-Box 
7.)	Merging DAvE with μ Vision3

k_40	ADC
k_50	CAPCOM6: PWM generation (asymmetrical/edge-aligned, single-shot, modulated) – playing music

Feedback

8.)	Feedback
-----	--------------------------

1.) XC164CM Easy Kit:



Ordering information:

Starter Kit - Type	Title	Ordering No.	Comment
SK-XC164CM	XC164CM Easy Kit	B158-H8647-X-0-7600	

Distribution Worldwide:

<http://www.infineon.com/cgi-bin/ifx/portal/ep/channelView.do?channelId=-66982&pageTypeId=17224>

You need a **Power Supply**:

The XC164CM Board requires an external power supply.

A regulated DC power supply with 9 Volts, 400 mA can be connected to the power connector.

You need a **RS-232 Serial Cable (1:1; 9-pin Sub-D plug – 9-pin Sub-D connector)**

[For further information, please refer to the XC164CM Board Manual, V0.3, April 2004 .](#)

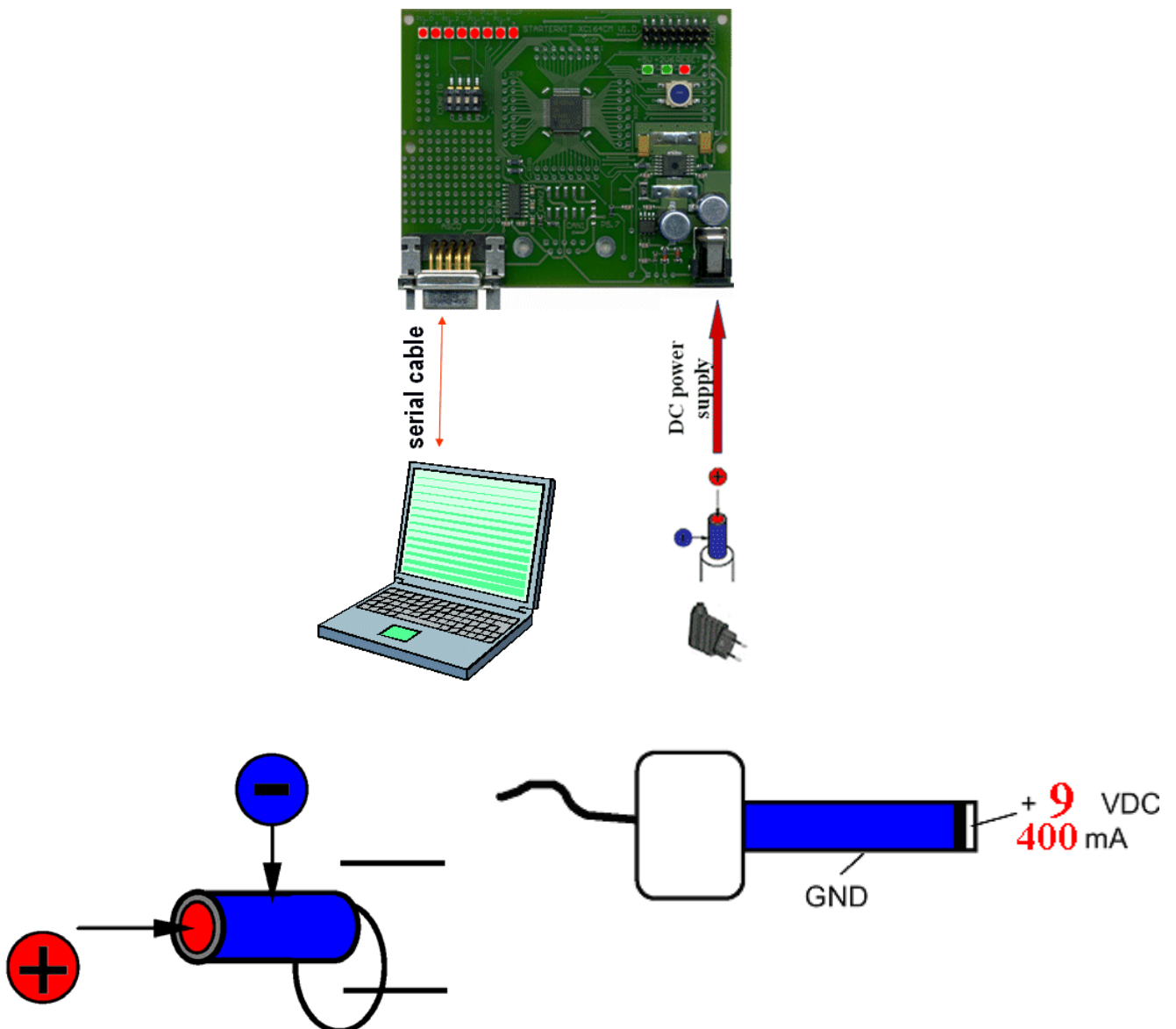
[For further information, please refer to the XC164CM Board Manual, V0.9, Aug. 2005 .](#)

[For further information, please refer to the XC164CM Board Manual, V1.0, Jan. 2006 .](#)

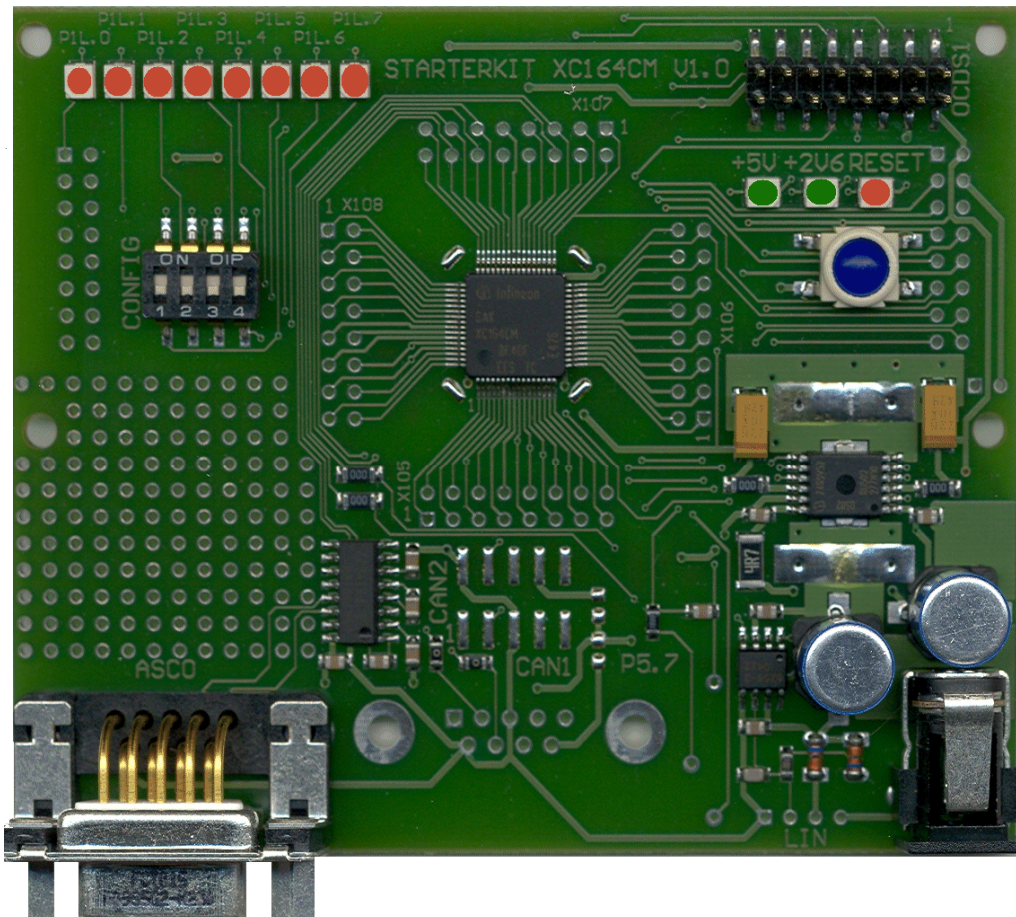
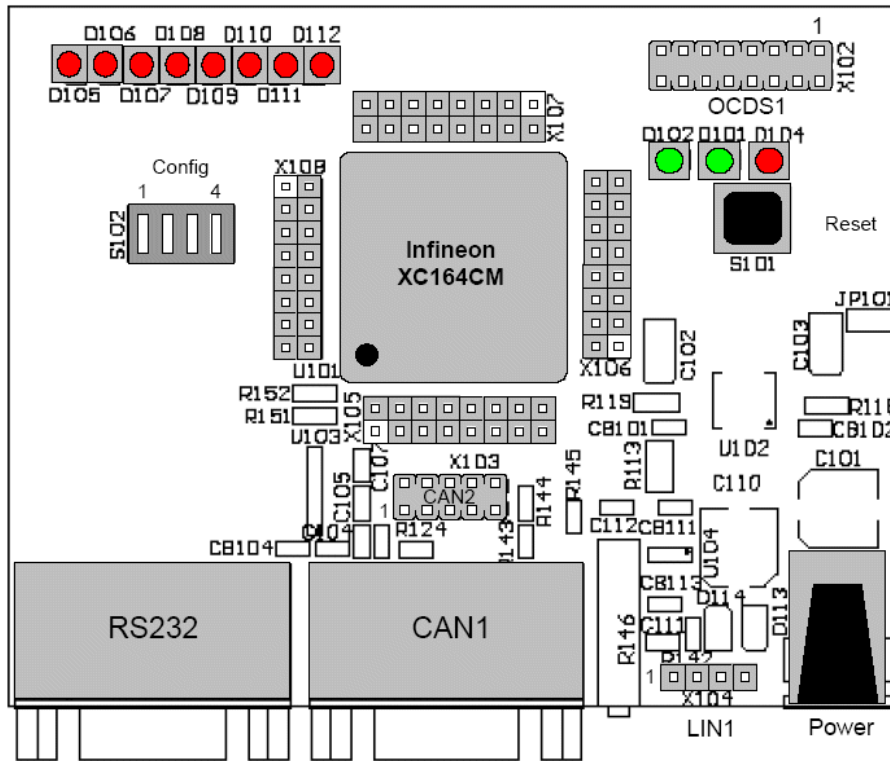
[For further information, please refer to the XC164CM Board Manual, V2.0, Jan. 2007 .](#)

Note:

Since Board Manual V2.0 the Easy Kit combines the powerful Debug Support, a serial communication and Power Supply over one USB cable.

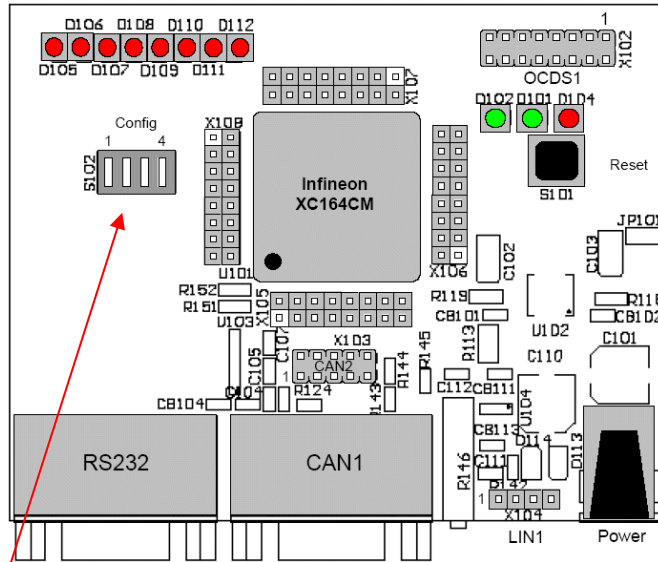


Layout Overview:



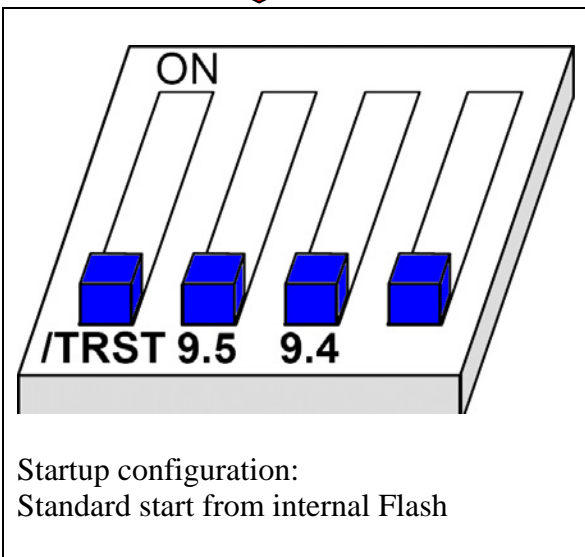
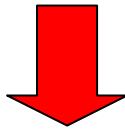
DIP switch 102 Jumper Settings:

Standard Internal Start



Standard Internal Start

Switch S102:



2.) DAvE – Installation for XC164CM microcontrollers:



Install DAvE (mothersystem):

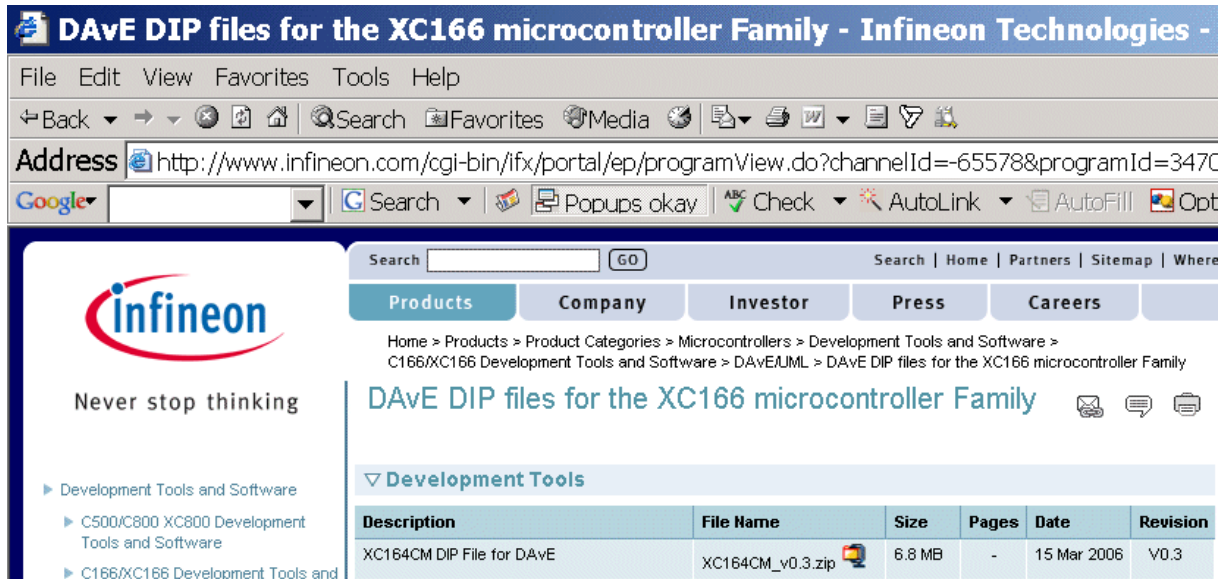
Download @ <http://www.infineon.com/DAvE> the DAvE-mothersystem **setup.exe**

Description	File Name	Size	Date	Revision
DAvE - Mothersystem without derivatives	setup.exe 	15.2 MB	7 Jul 2005	

and execute **setup.exe** to install DAvE .

Install the XC164CM microcontroller Update:

1.)
Download @ <http://www.infineon.com/DAvE> the DAvE-update-file (.DIP) for the required microcontroller



DAvE DIP files for the XC166 microcontroller Family - Infineon Technologies -

File Edit View Favorites Tools Help

⏪ Back ⏩ Forward 🏠 Home 🔍 Search 📁 Favorites 🌐 Media 📄 Print 📄 Save 📄 Open 📄 Close

Address <http://www.infineon.com/cgi-bin/ifx/portal/ep/programView.do?channelId=-655788&programId=3470>

Google Search Popups okay Check AutoLink AutoFill Opt

Search GO Search | Home | Partners | Sitemap | Where

Products Company Investor Press Careers

Home > Products > Product Categories > Microcontrollers > Development Tools and Software > C166/XC166 Development Tools and Software > DAvE/UML > DAvE DIP files for the XC166 microcontroller Family


DAvE DIP files for the XC166 microcontroller Family

Development Tools

Description	File Name	Size	Pages	Date	Revision
XC164CM DIP File for DAvE	XC164CM_v0.3.zip	6.8 MB	-	15 Mar 2006	V0.3

Unzip the zip-file “XC164CM_v0.3.zip” and save “XC164CM.DIP”
@ e.g. D:\DAvE\XC164CM-2006-07-20.

2.)

Start DAVe - ([click](#) )

3.)

View

Setup Wizard

Default: • [Installation](#)

Forward>

Select: • [I want to install products from the DAVe's web site](#)

Forward>

Select: [D:\DAVe\XC164CM-2006-07-20](#)

Forward>

Select: Available Products

[click](#) ✓ XC164CM

Forward>

Install

End

4.)

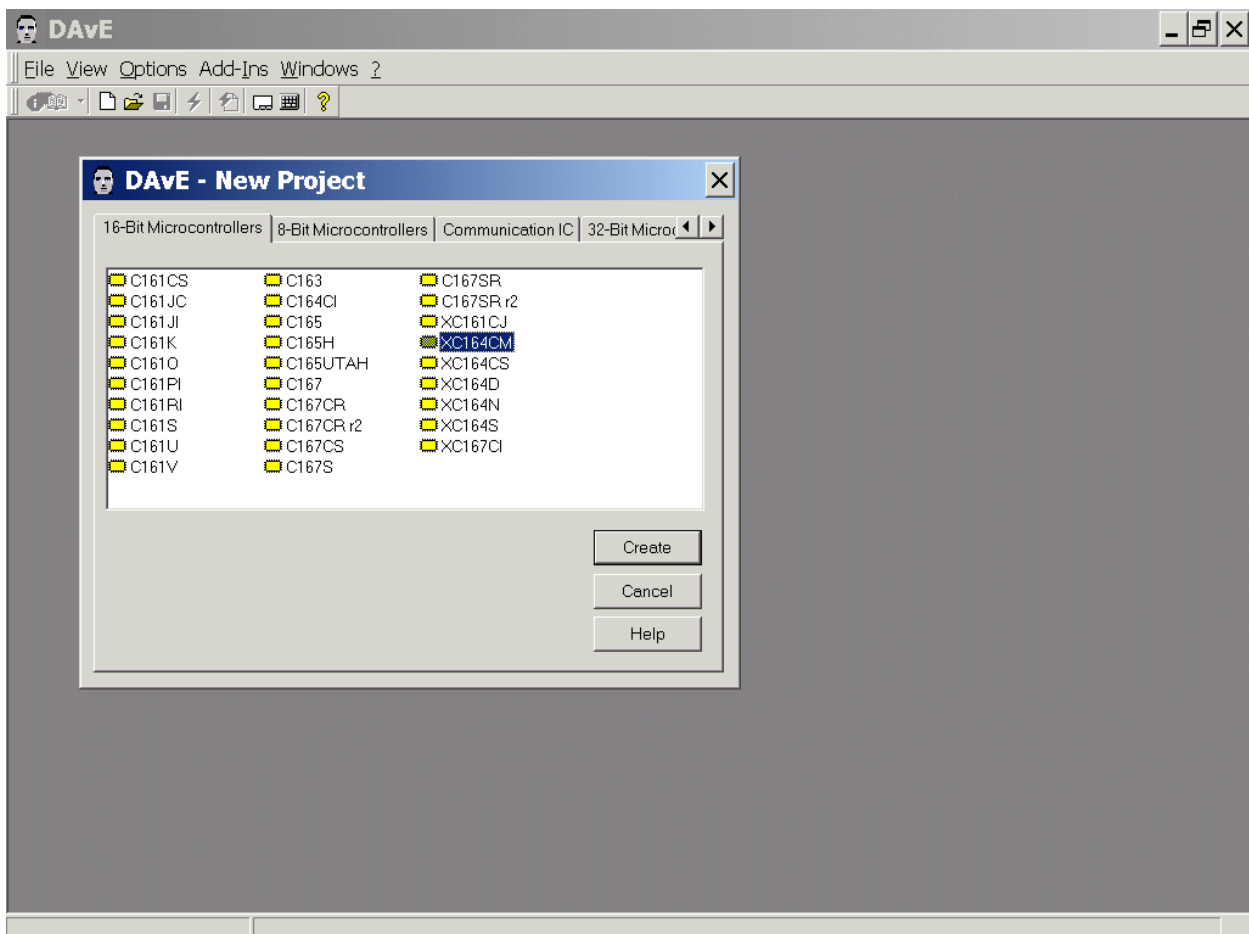
DAVe is now ready to generate code for the XC164CM microcontroller.

3.) DAVe - Microcontroller Initialization after Power-On:



Start the program generator DAVe and select the XC164CM microcontroller:

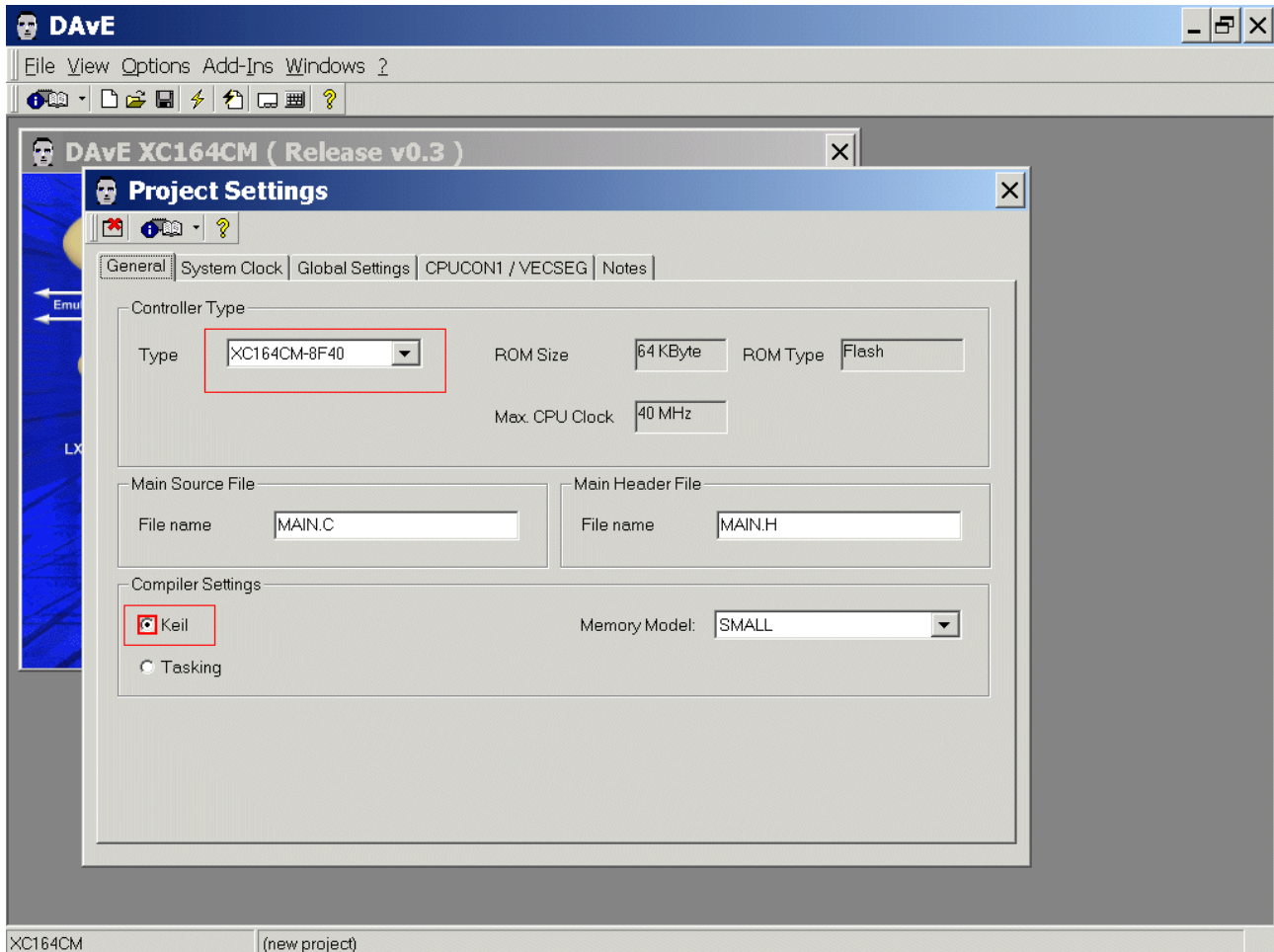
File;
New;
16-Bit Microcontrollers;
XC164CM;
Create:



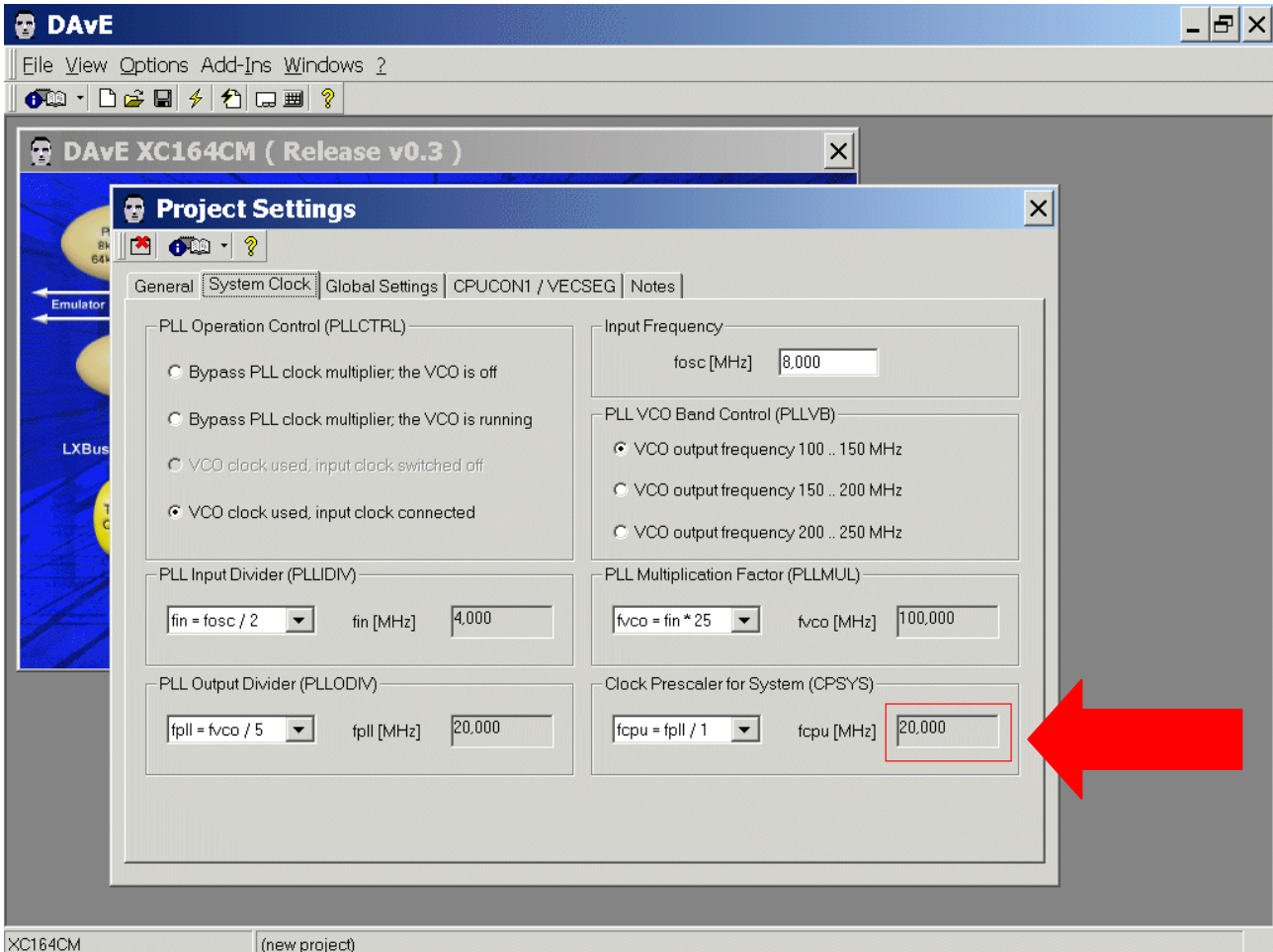
Choose the Project Settings as you can see in the Screenshots:

General: Controller Type: Type: select XC164CM-8F40

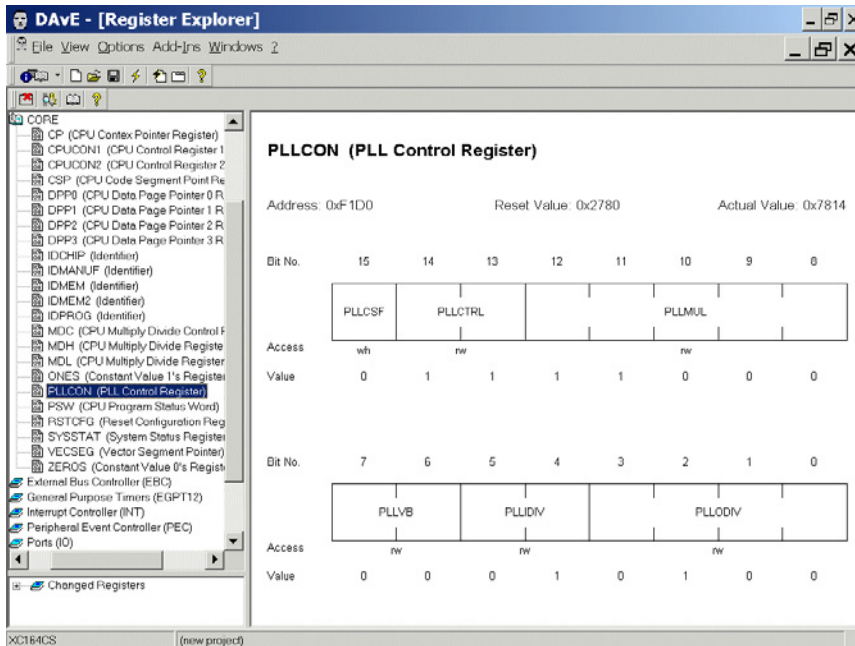
General: For the KEIL Compiler choose Keil in the Compiler Settings:



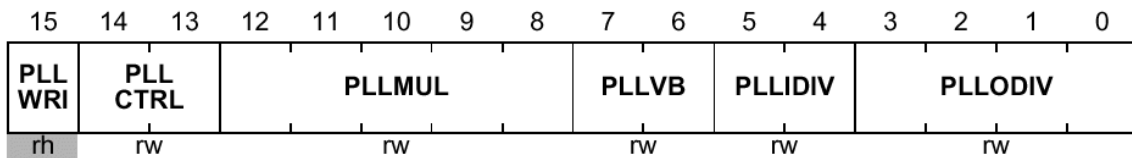
System Clock: CPU Clock will be 20 MHz – (do nothing)



See also Core-Register PLLCON: [View – Register Explorer](#):



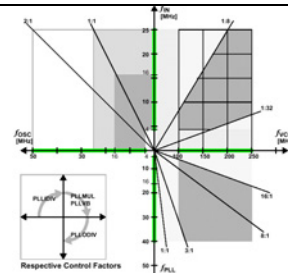
PLLCON
PLL Control Register **ESFR (F1D0H/E8H)**



PLLCON=0x7814

$$f_{PLL} = f_{OSC} * (PLLMUL+1) / ((PLLDIV+1) * (PLLODIV+1))$$

$$f_{PLL} = 8 * (24+1) / ((1 + 1) * (4 + 1)) = 20 \text{ MHz}$$

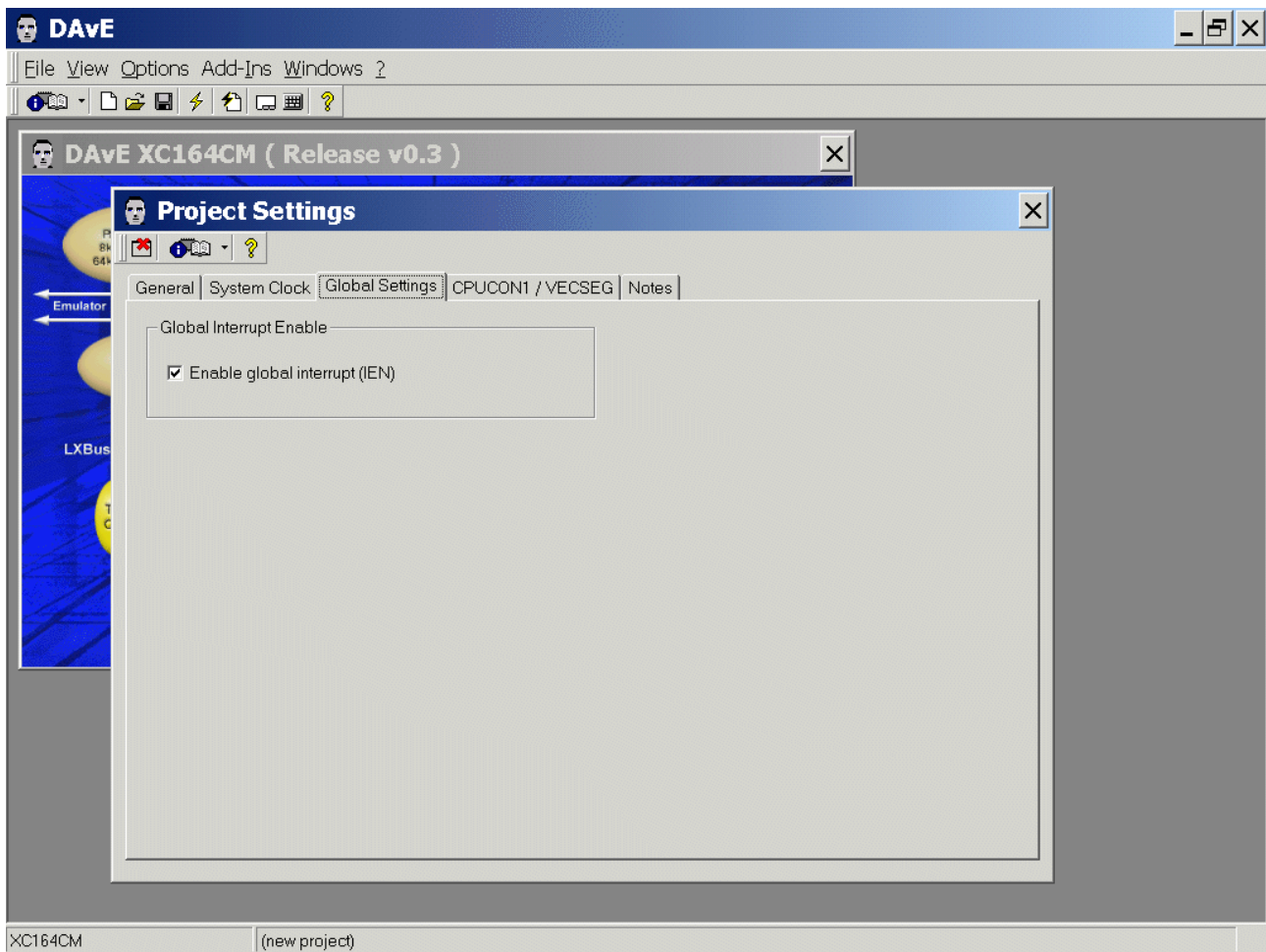


PLLCON=0x7814

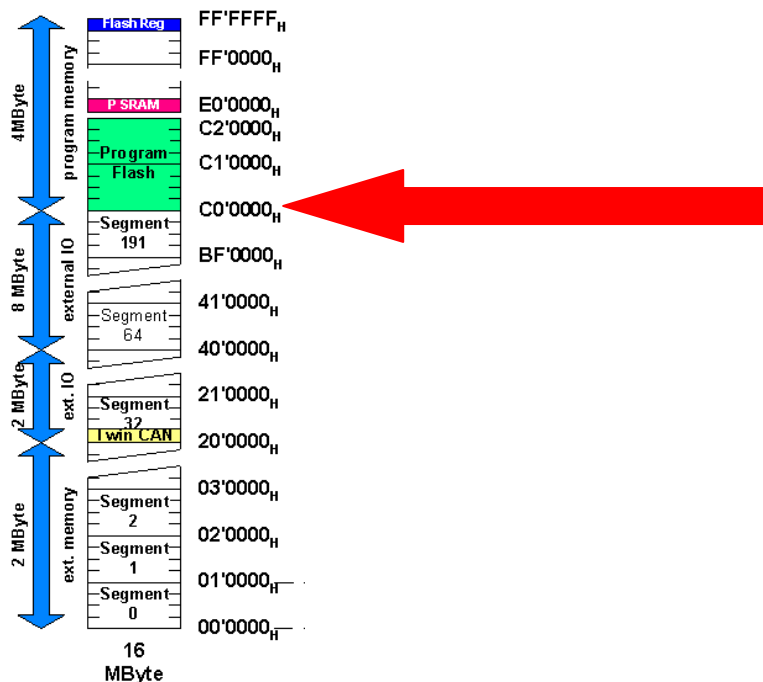
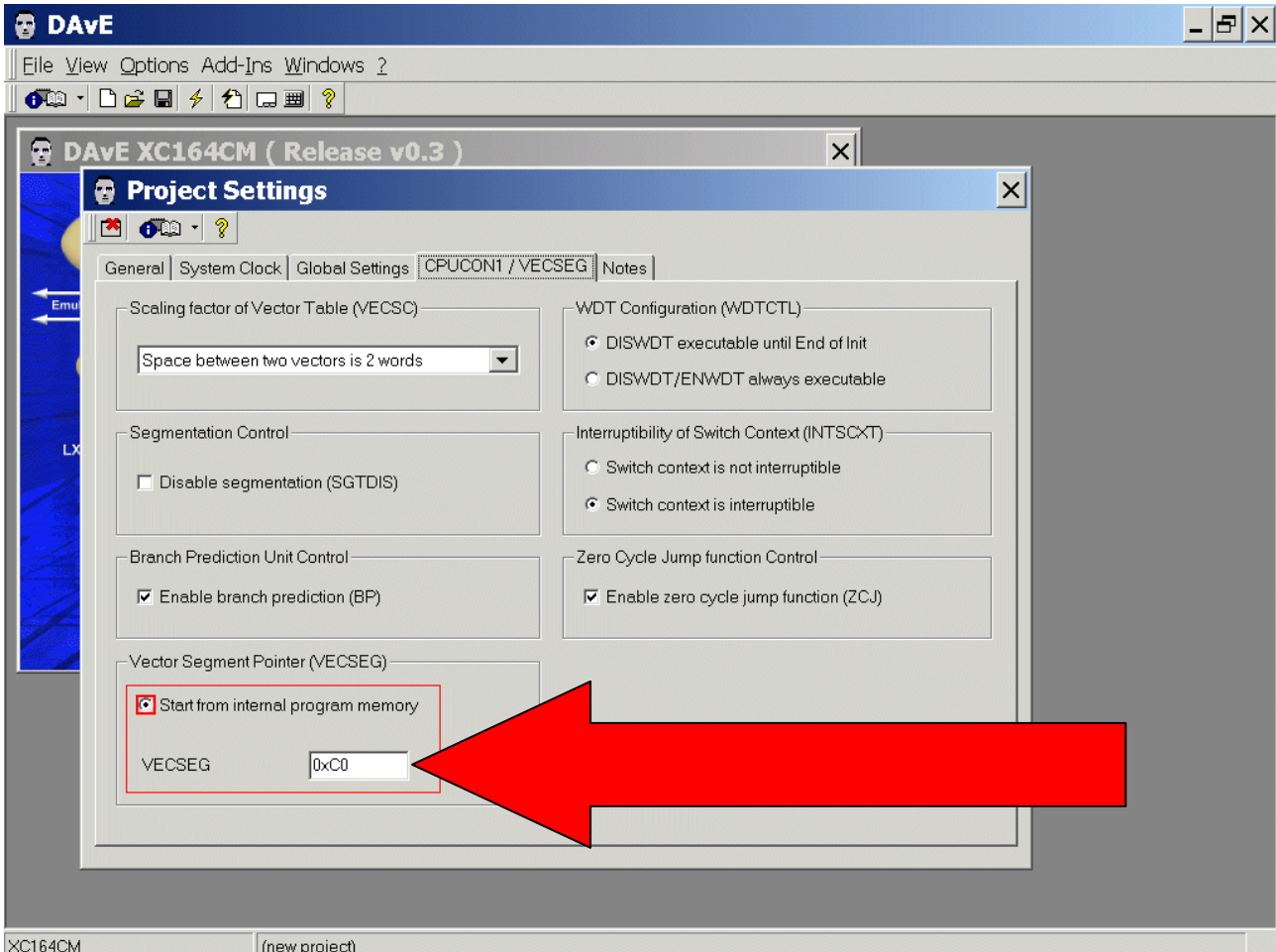
Clock Generation Control

PLLCON: <input type="text" value="0x7814"/>	PLLCTRL: <input type="text" value="PLL (Input On)"/>	PLLDIV: <input type="text" value="1"/>
	PLLVB: <input type="text" value="100..150 MHz"/>	PLLMUL: <input type="text" value="24"/>
XTAL (OSC): <input type="text" value="8.000000"/> MHz		PLLODIV: <input type="text" value="4"/>
CLOCK (CPU): <input type="text" value="20.000000"/> MHz		<input type="checkbox"/> PLLWRI

Global Settings: do not change configuration

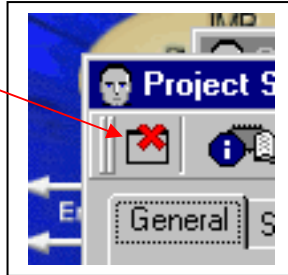


CPUCON1/VECSEG: Vector Segment Pointer: **click/check** ☉ Start from internal program memory



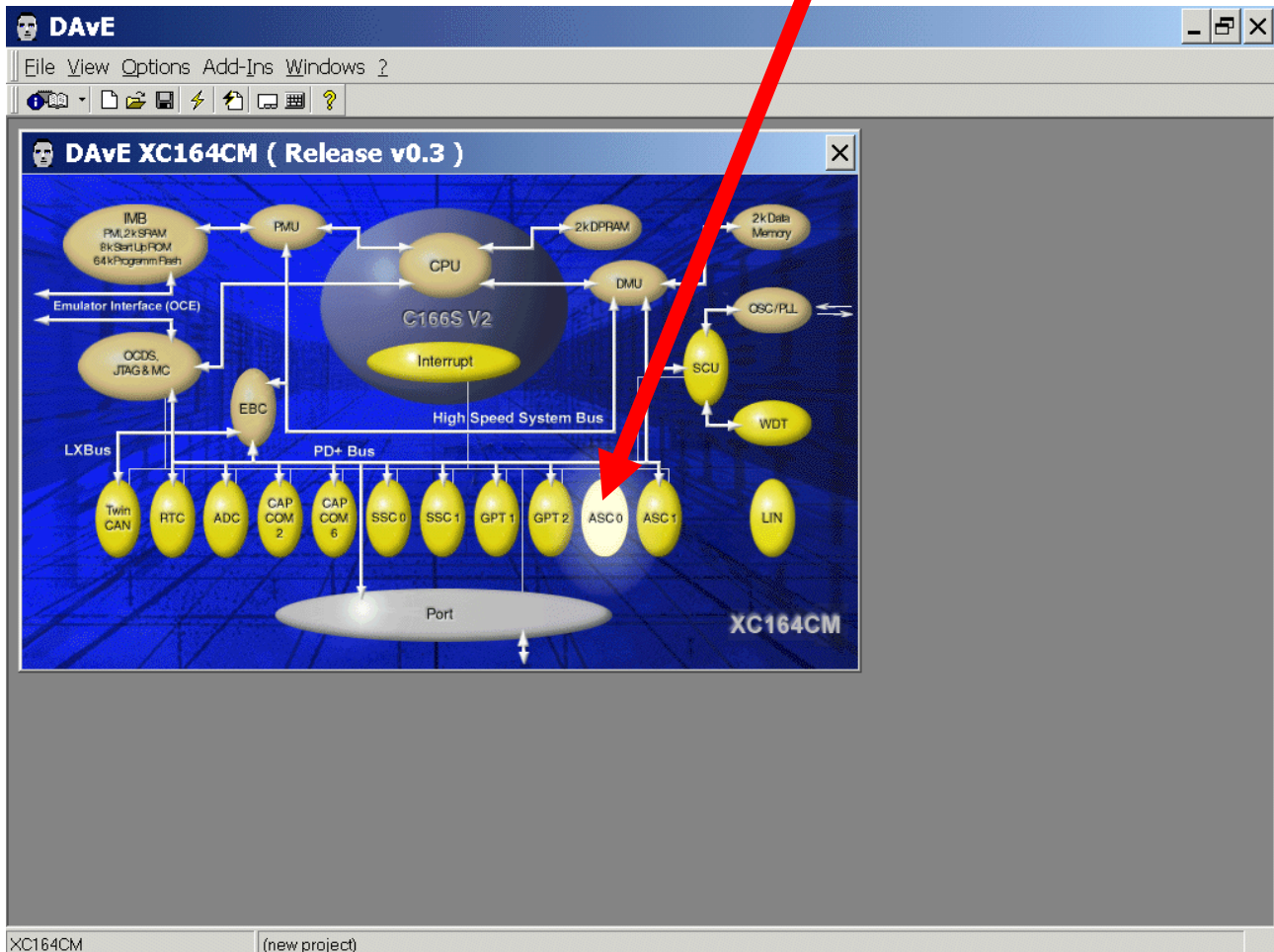
Notes: If you wish, you can insert your comments here.

Exit this dialog now by clicking  the close button:

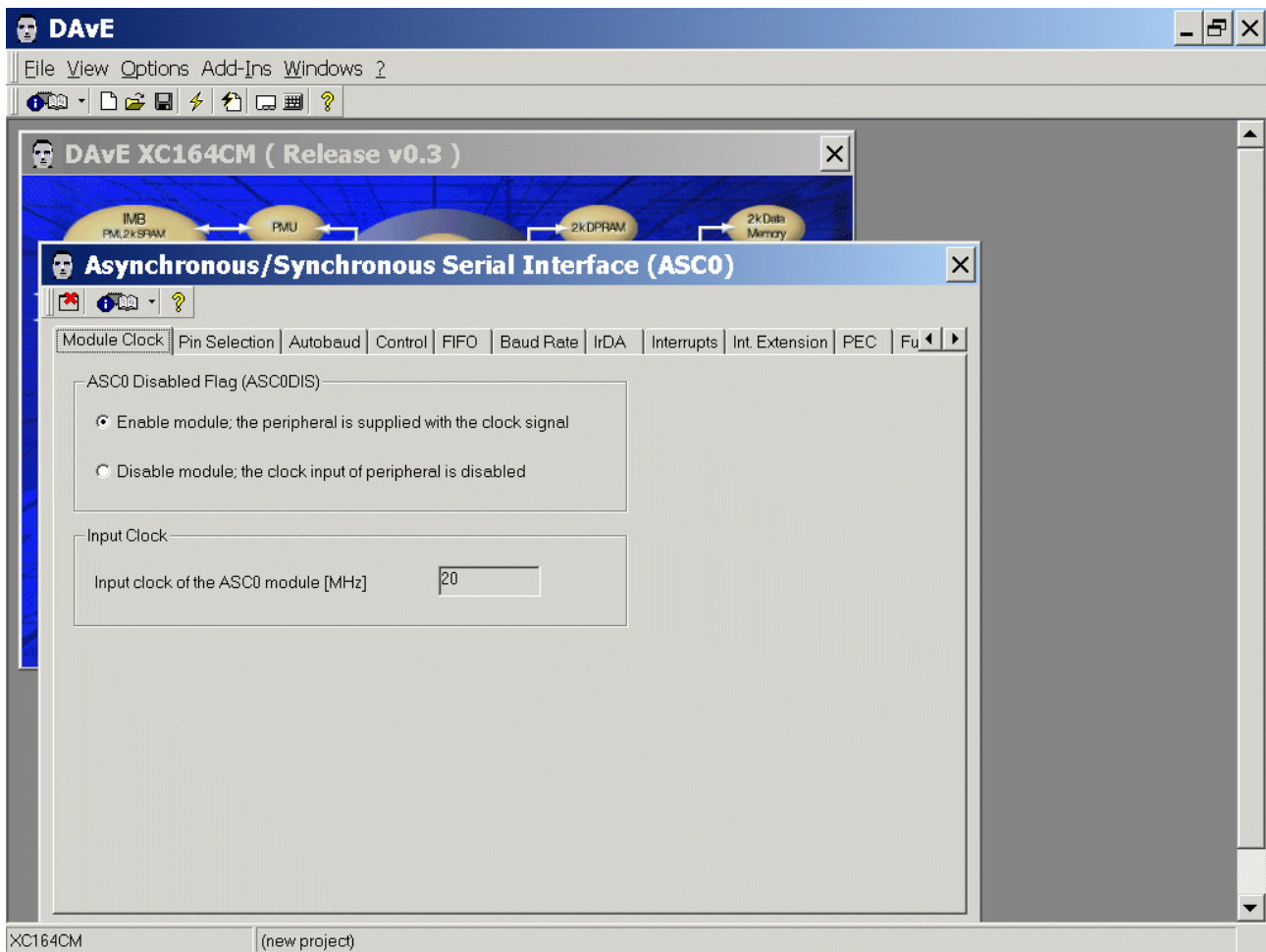


Configuration of the ASC0:

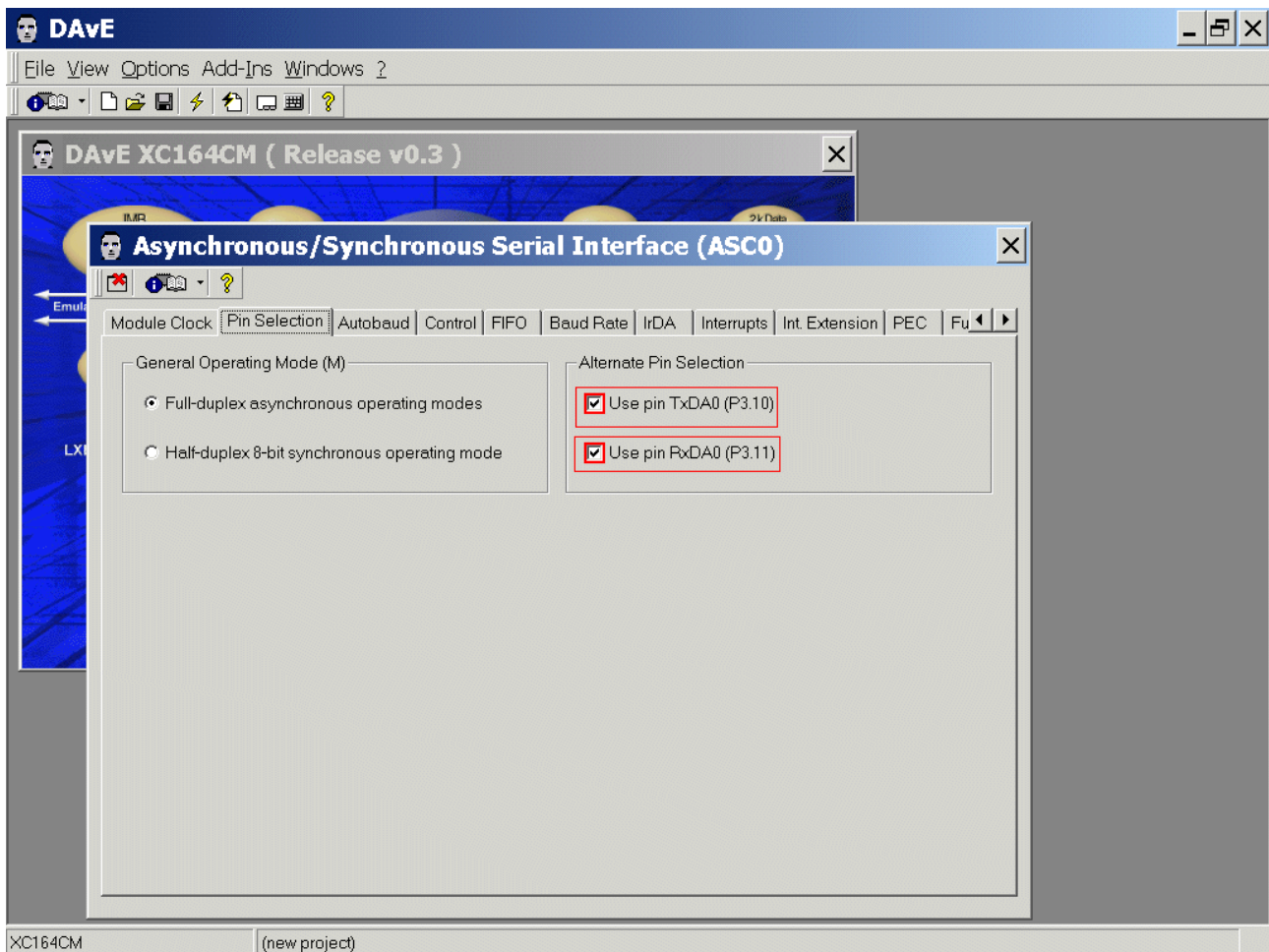
The configuration window can be opened by clicking the [specific block/module](#).



Module Clock: (do nothing)

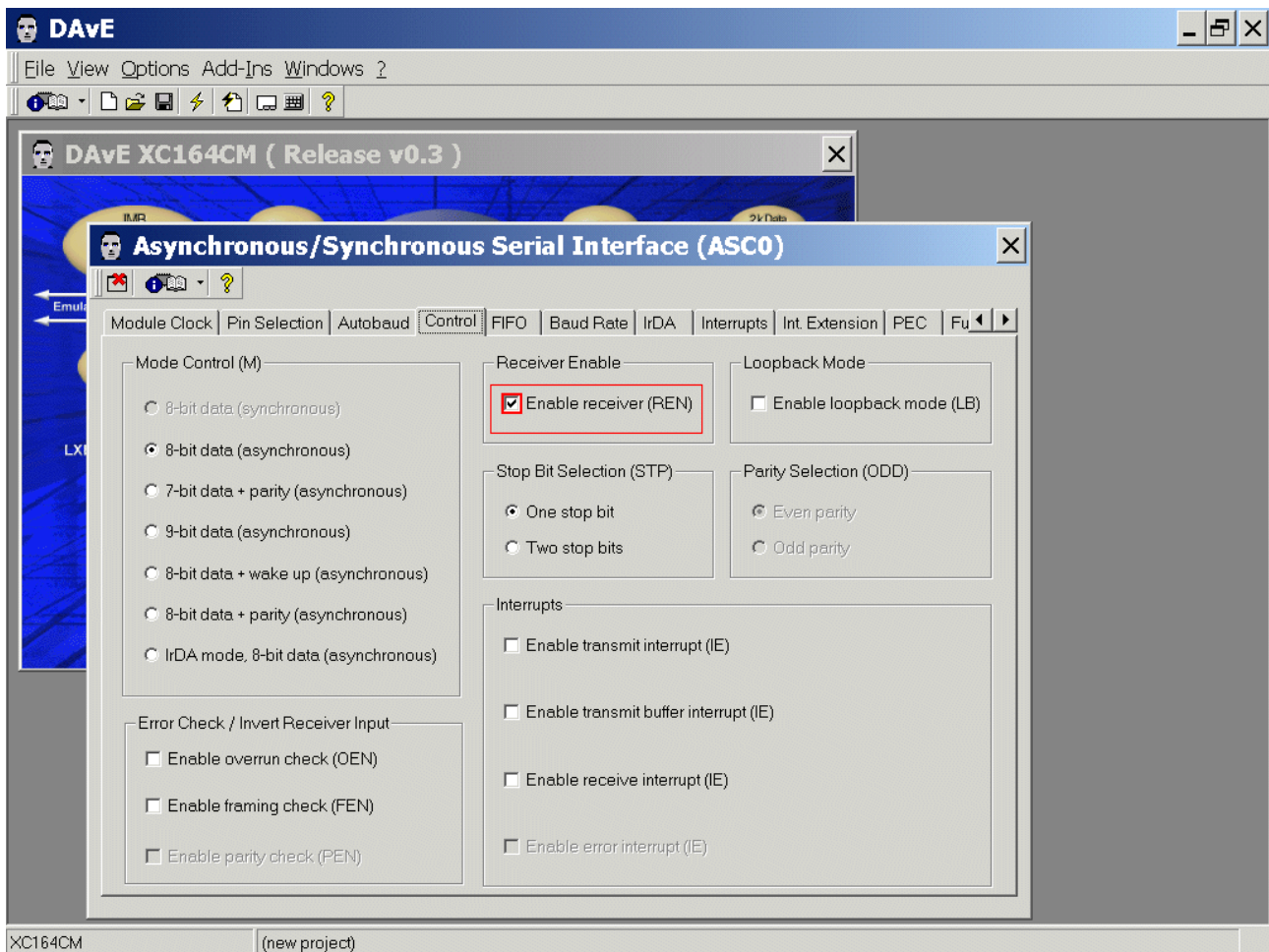


Pin Selection: Alternate Pin Selection: **click** ✓ Use pin Tx, **click** ✓ Use pin Rx



Autobaud: (do nothing)

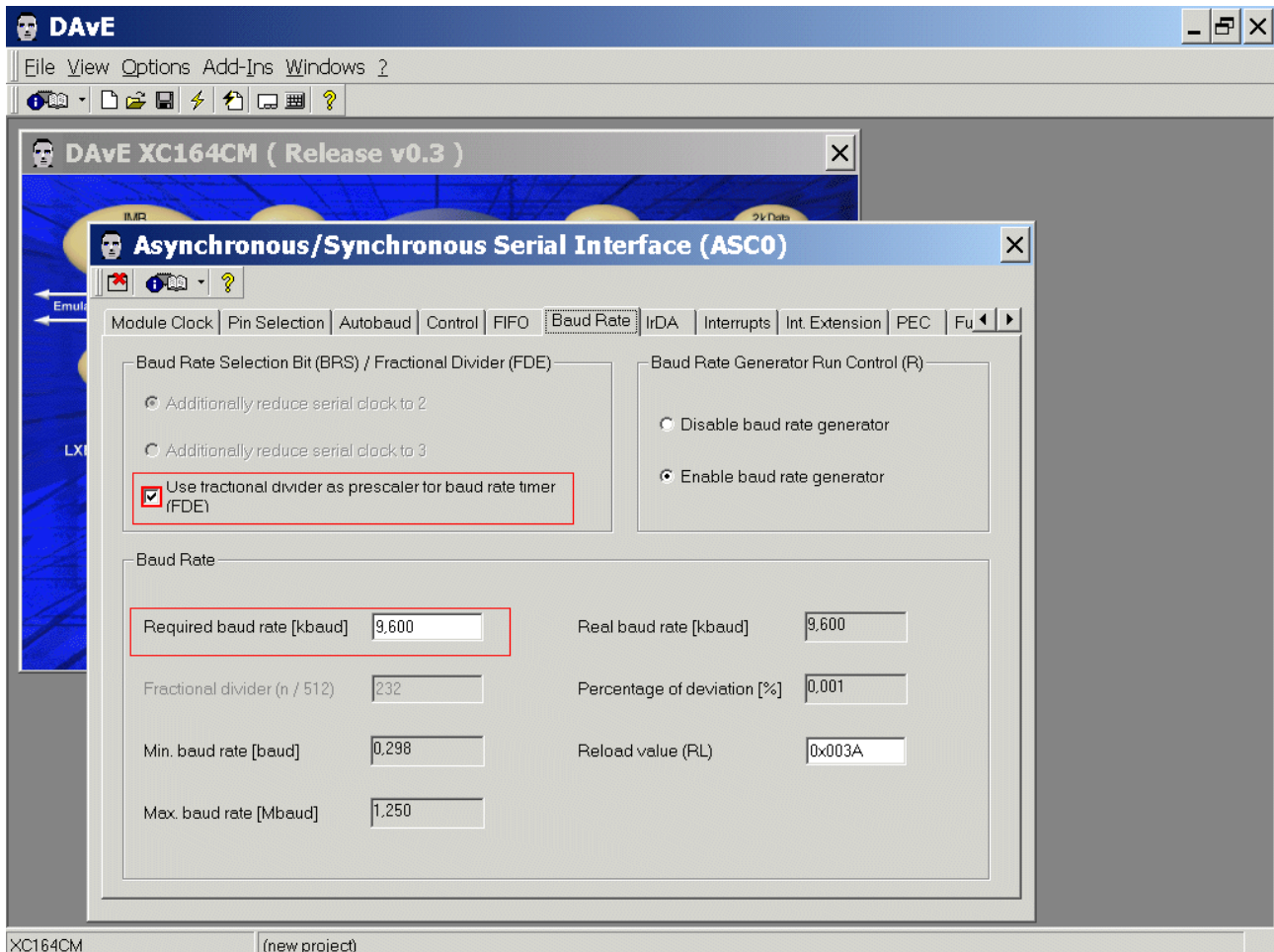
Control: Receiver Enable: **click** ✓ Enable receiver



FIFO: (do nothing)

Baud Rate: Baud Rate: Required baud rate [kbaud] **input** 9,600 <ENTER>

Baud Rate: Baud Rate Selection Bit: **additionally – if you want:** **click** ✓ Use fractional divider as prescaler ...



Note:

Validate each alpha numeric entry by pressing **ENTER**.

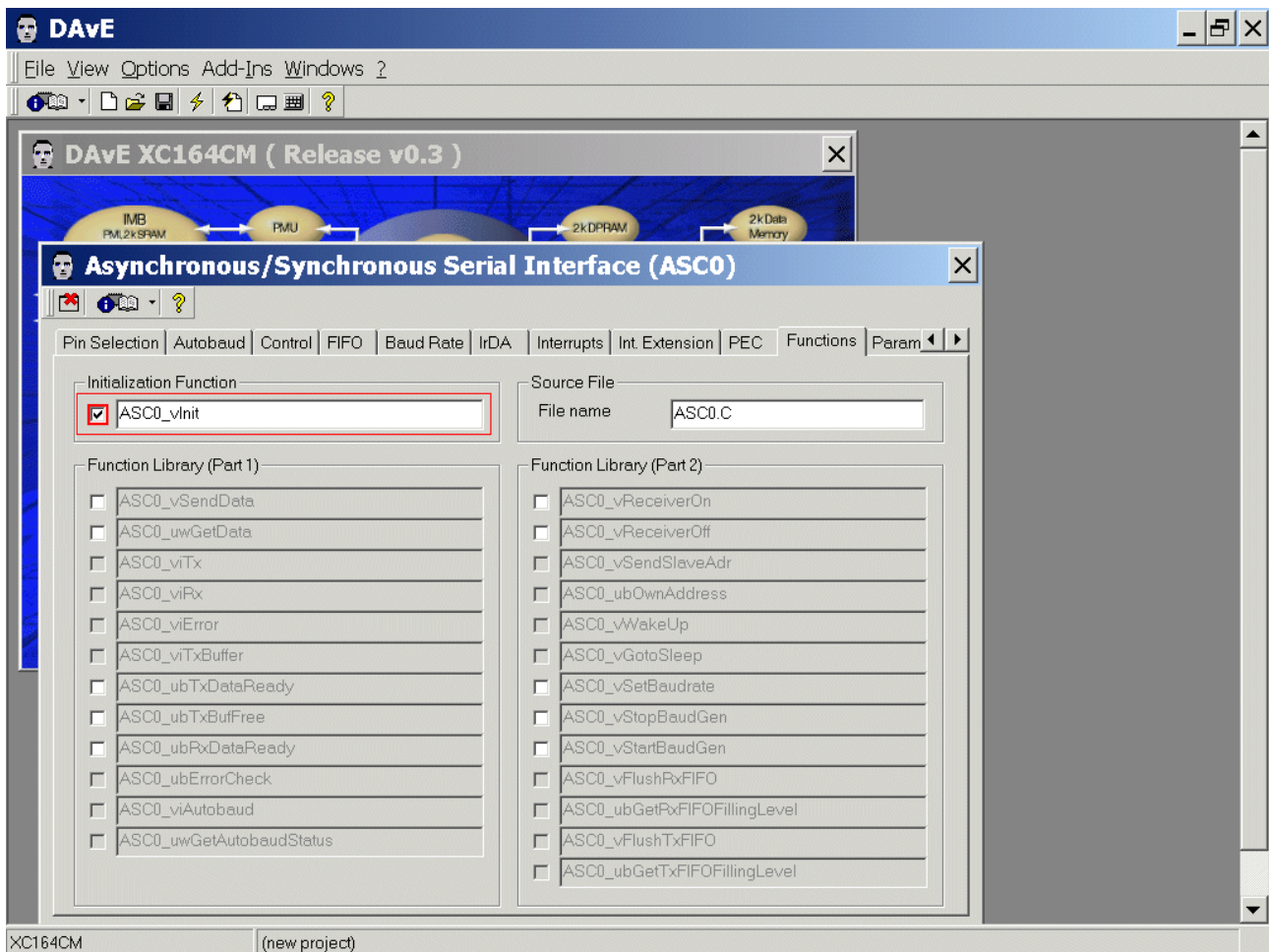
IrDA: (do nothing)

Interrupts: (do nothing)

Int. Extension: (do nothing)

PEC: (do nothing)

Functions: Initialization Function: **click** ✓ ASC0_vInit



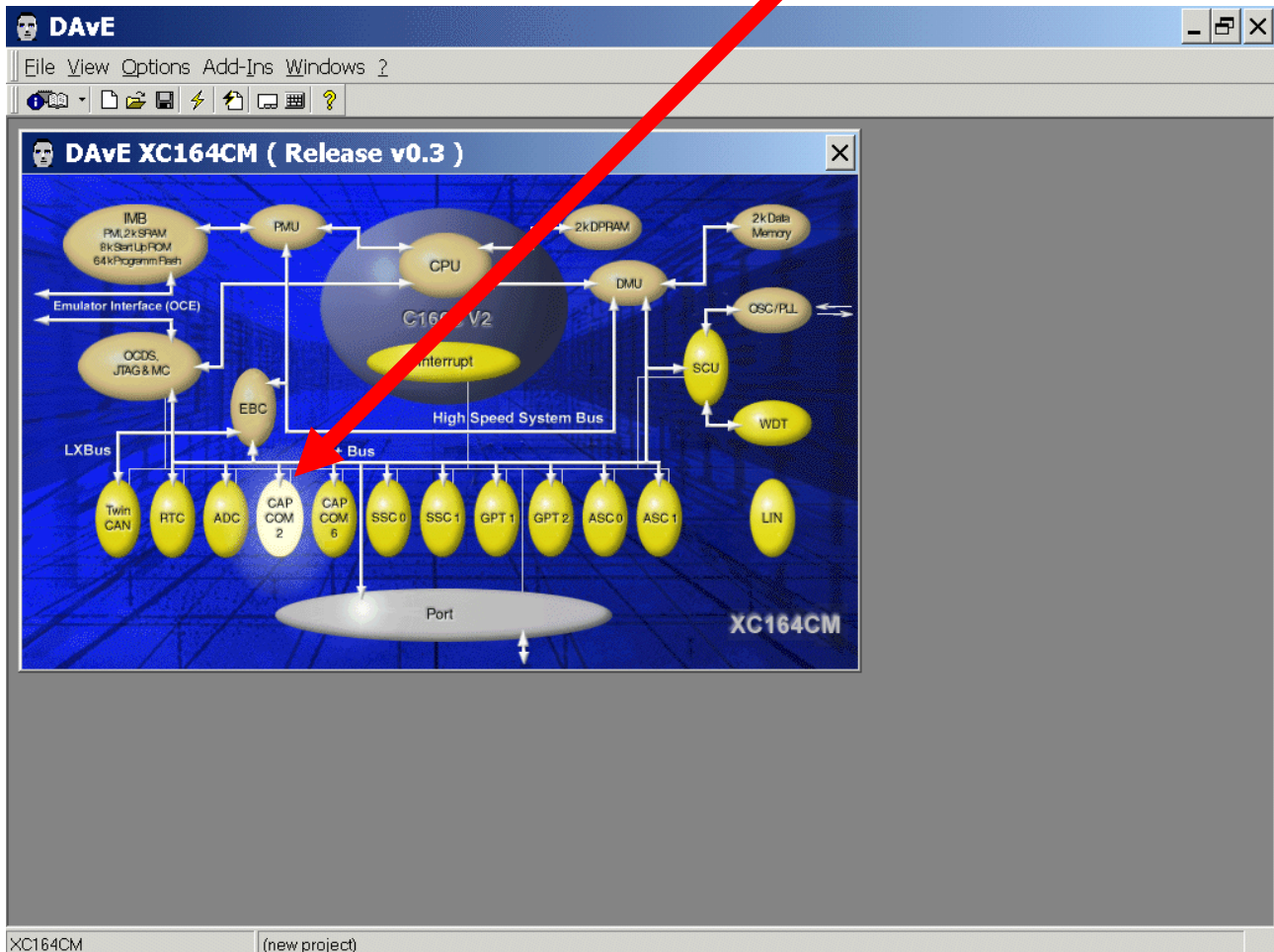
Parameters: (do nothing)

Notes: If you wish, you can insert your comments here.

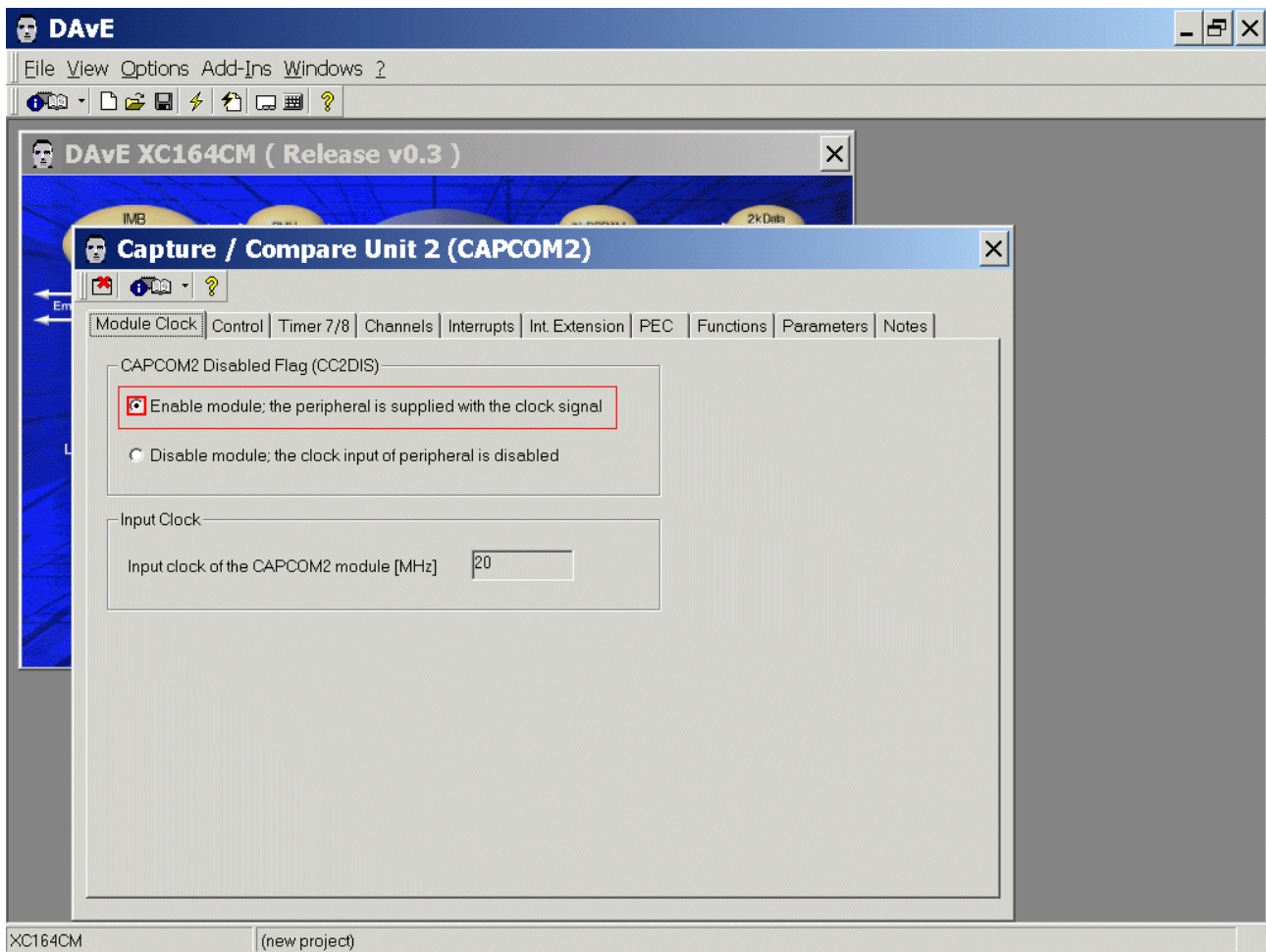
Exit this dialog now by clicking  the close button:

Configure Timer T7 in CAPCOM 2:

The configuration window can be opened by clicking the [specific block/module](#).

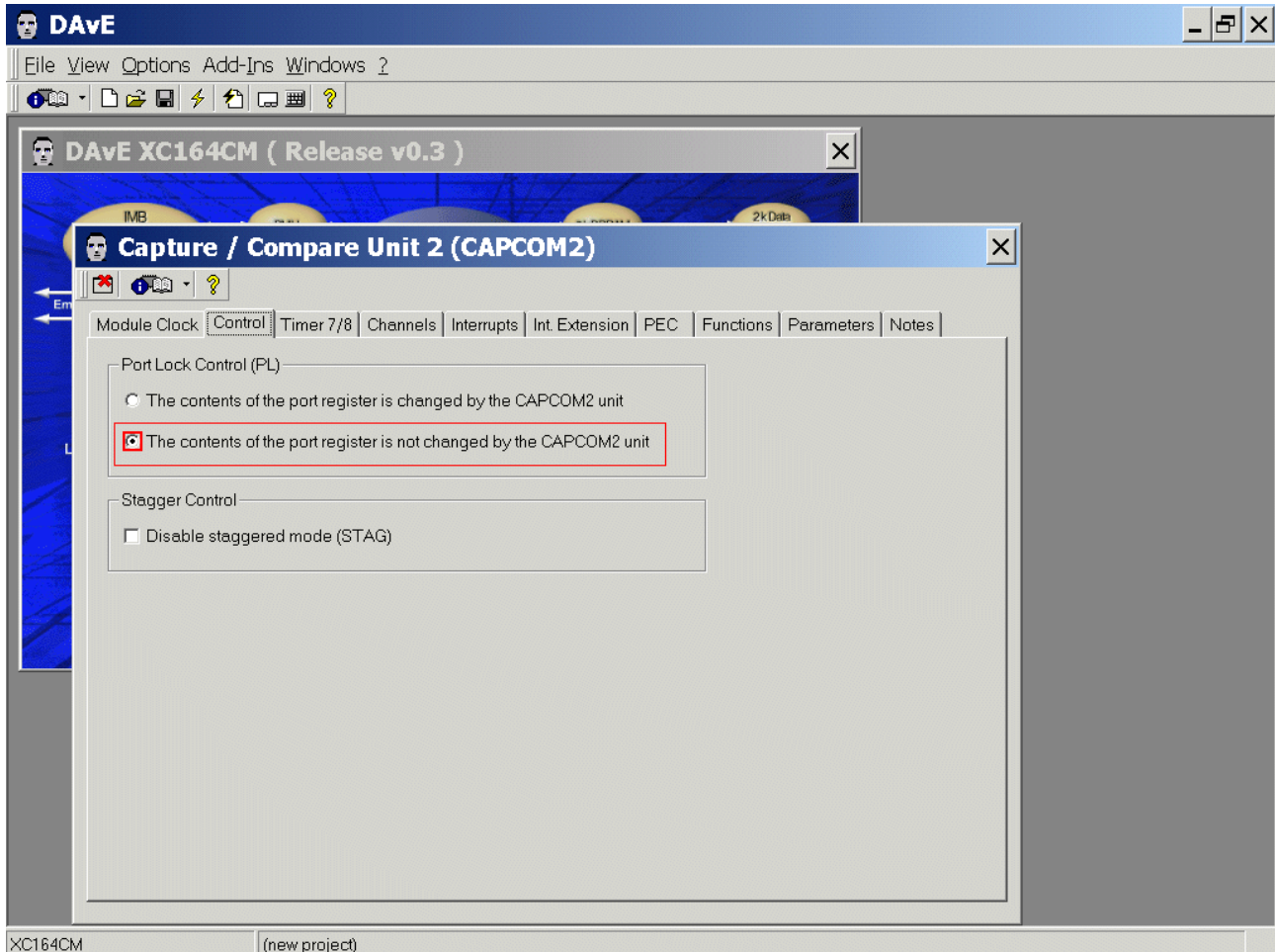


Module Clock: CAPCOM2 Disable Flag: **click** Enable module



Control: Port Lock Control:

click The contents of the port register is not changed by the CAPCOM2 unit

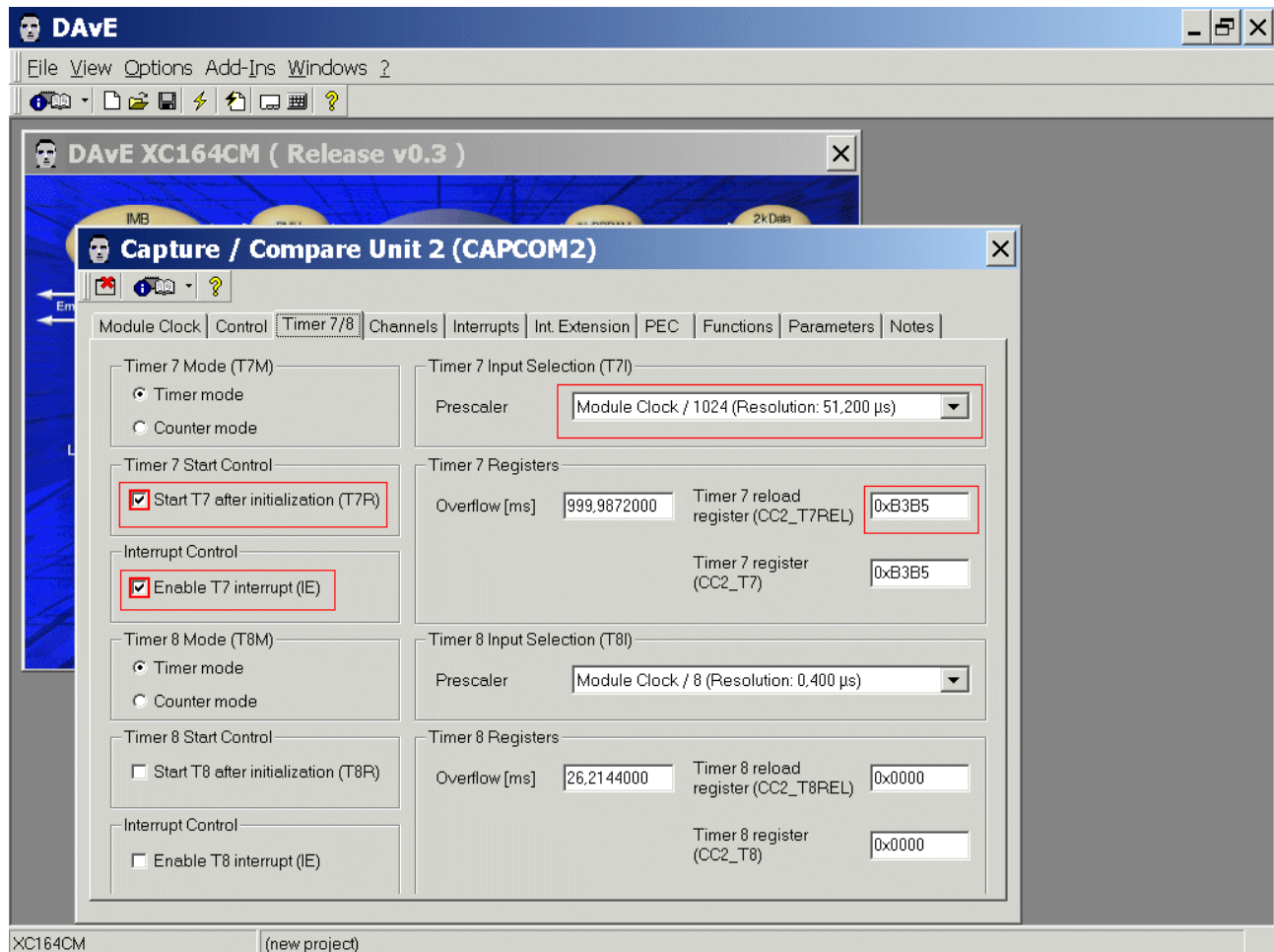


Timer 7/8: Timer 7 Start Control: **click** ✓ Start T7 after initialization (T7R)

Timer 7/8: Interrupt Control **click** ✓ Enable T7 interrupt (IE)

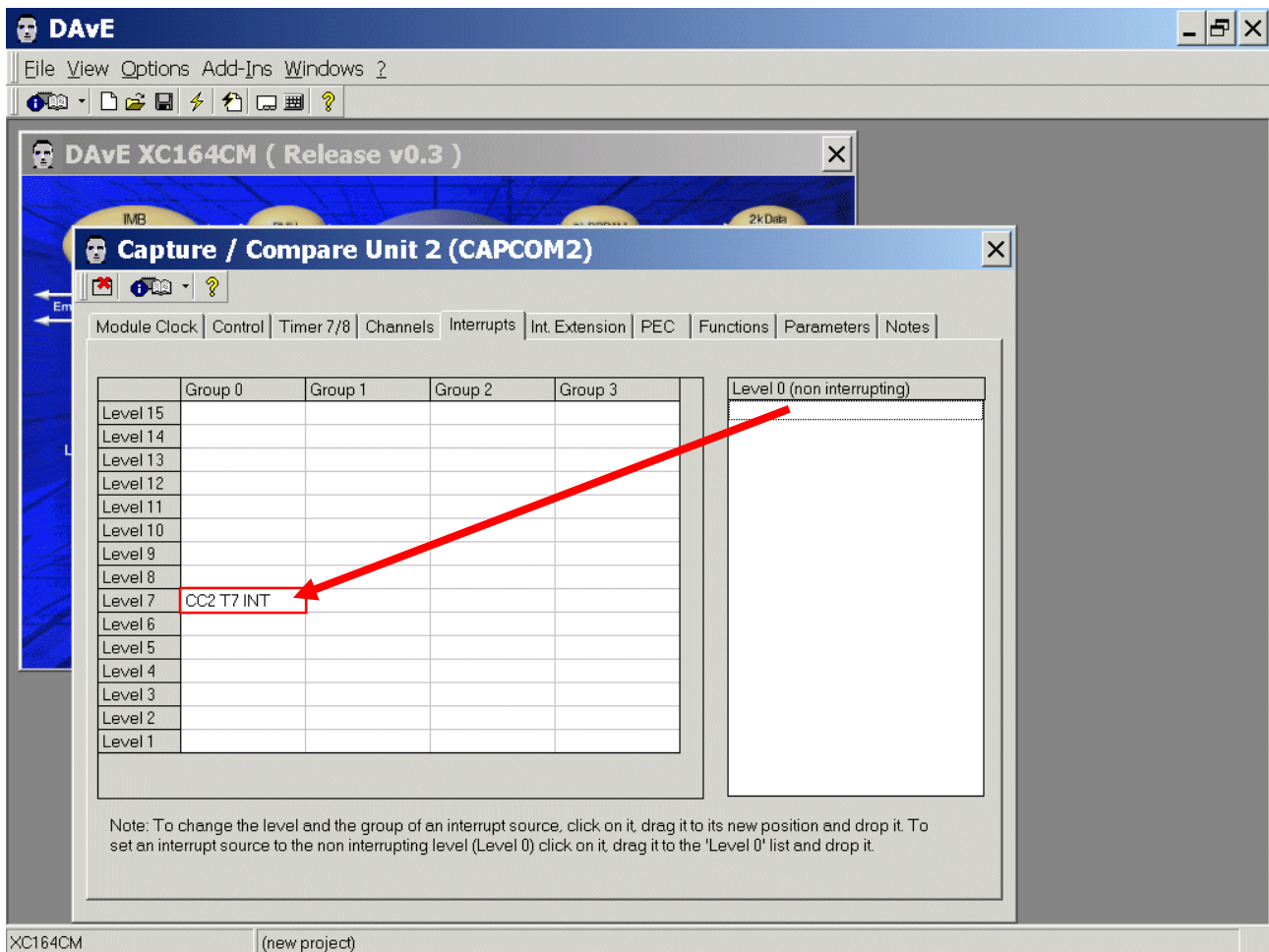
Timer 7/8: Timer 7 Input Selection (T7I): Prescaler: **choose** Module Clock/1024

Timer 7/8: Timer 7 Registers: Timer 7 reload register: **input** 0xB3B5 <ENTER> for 1 second



Channels: (do nothing)

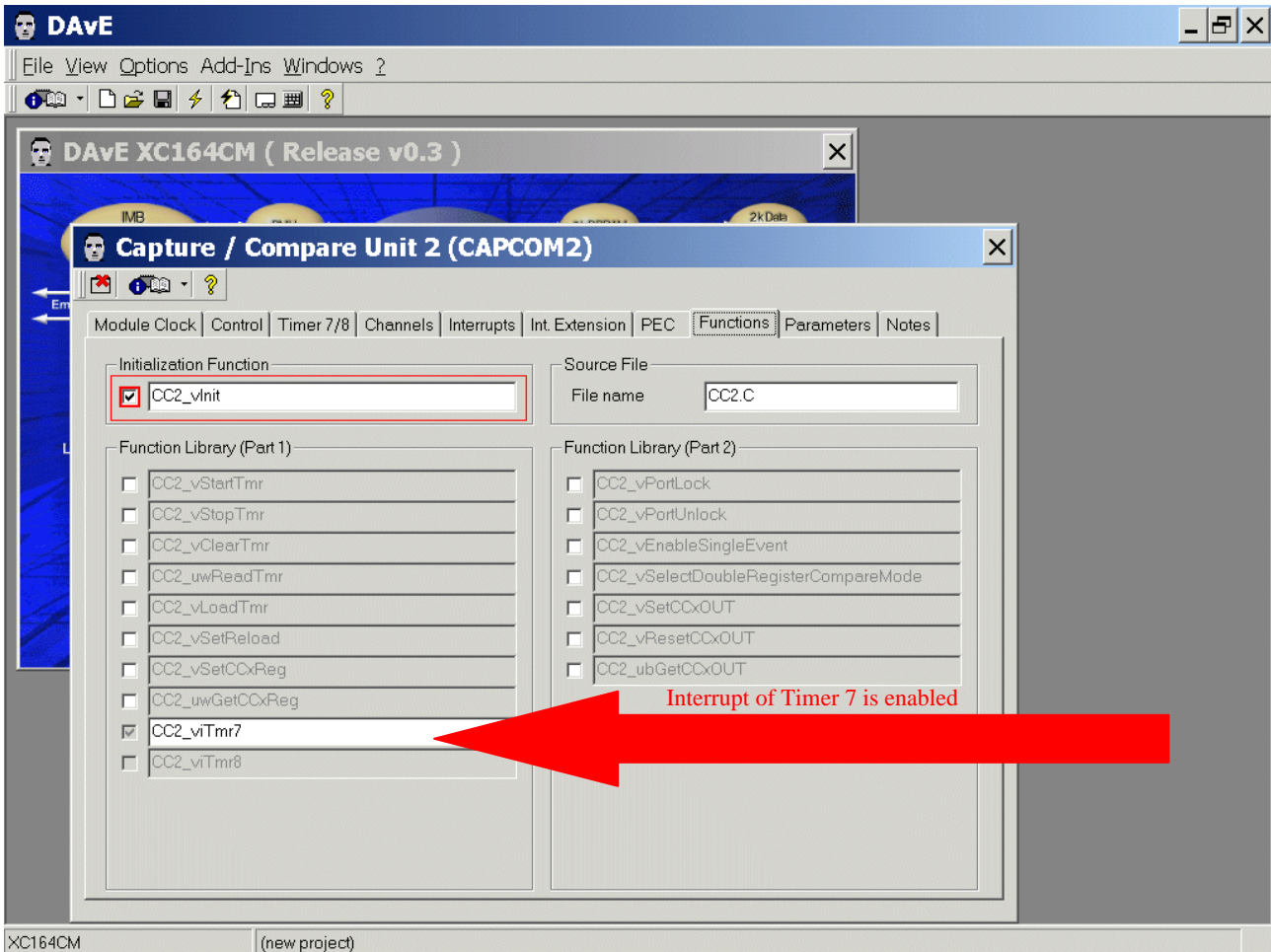
Interrupts: drag and drop the CC2 Timer 7 INT to Interrupt Level 7, Group 0



Int. Extension: (do nothing)

PEC: (do nothing)

Functions: Initialization Function: **click** ✓ CC2_vInit



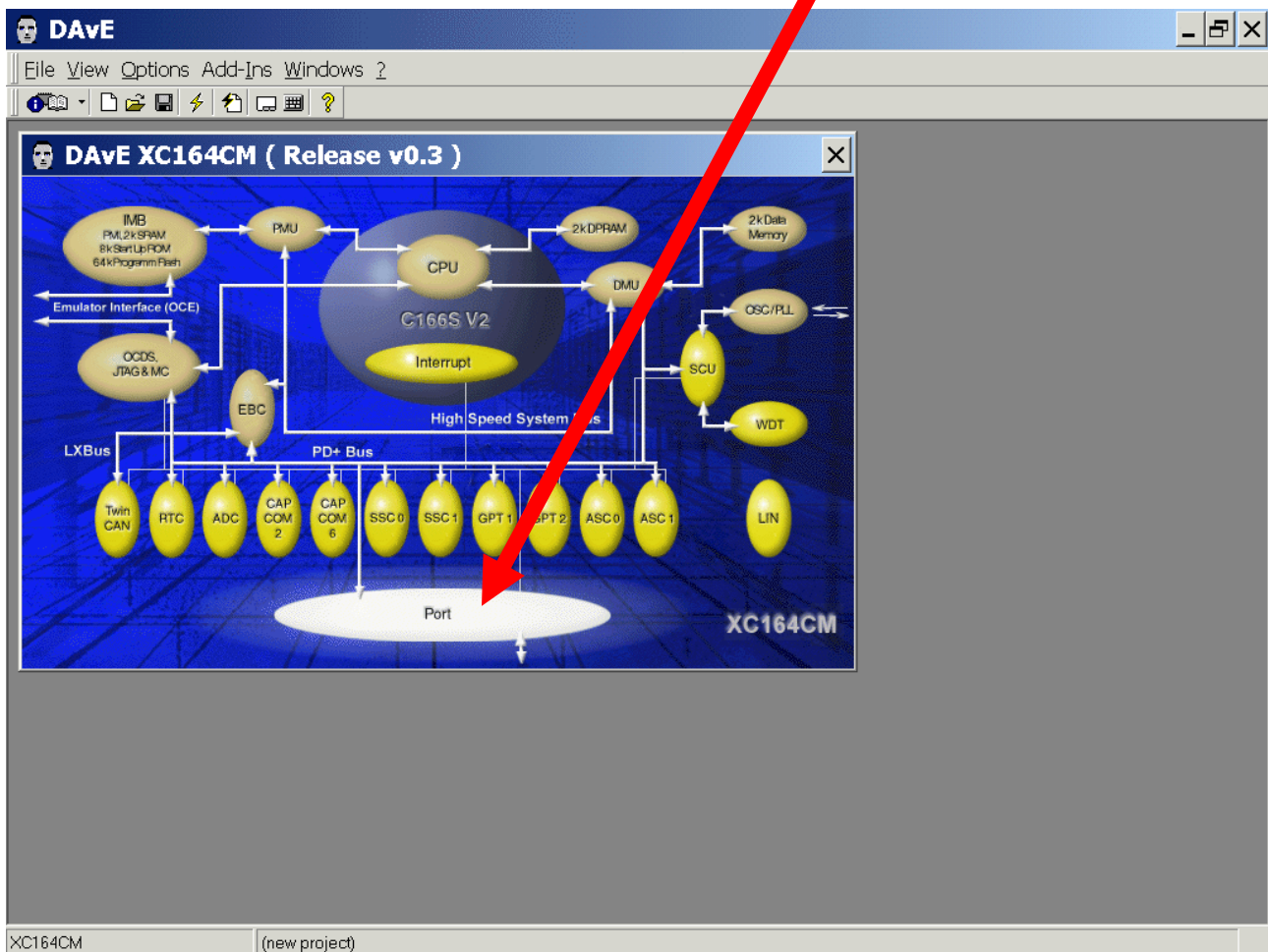
Parameters: (do nothing)

Notes: If you wish, you can insert your comments here.

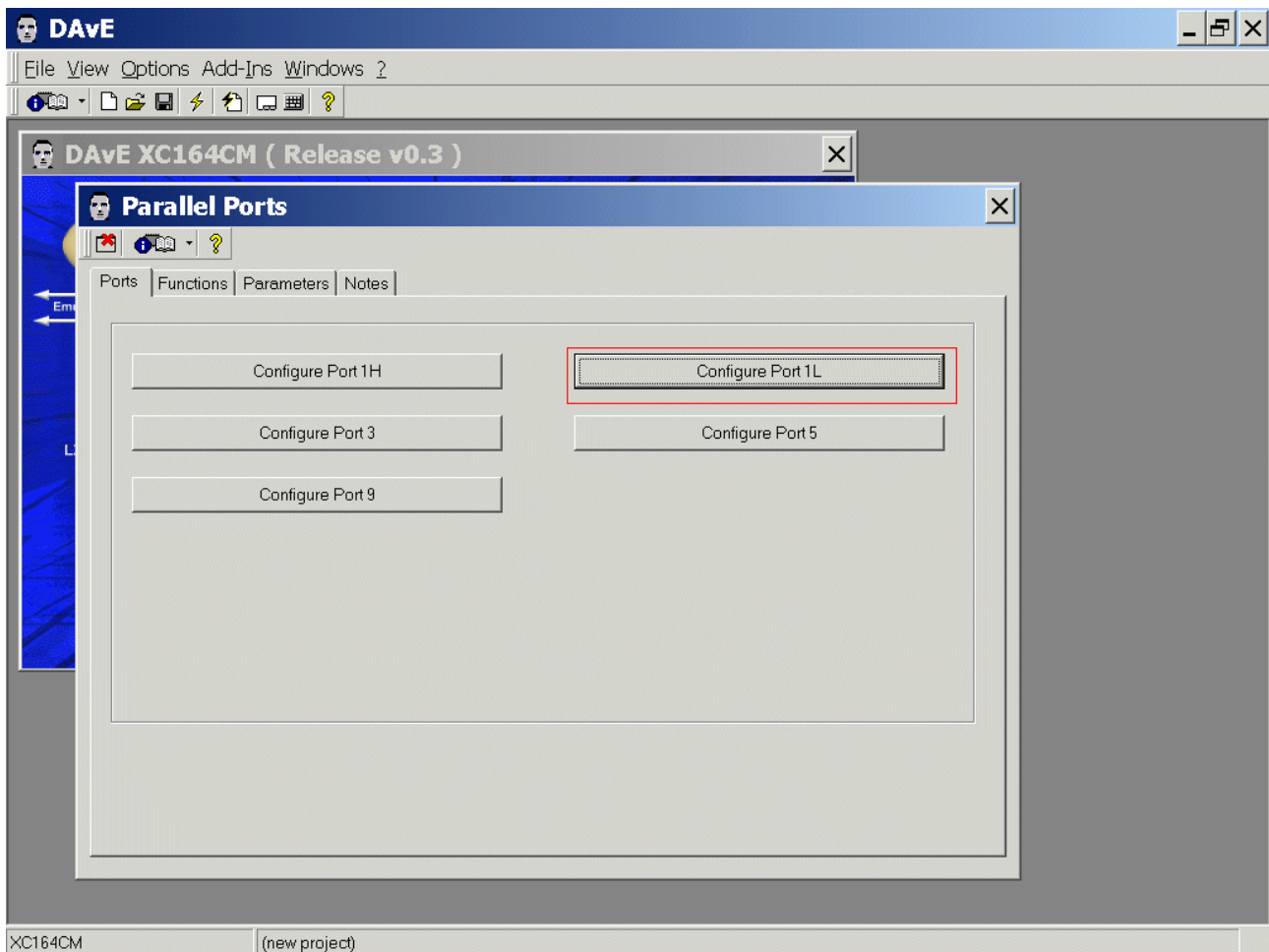
Exit this dialog now by clicking  the close button:

Configure Port 1L Pin 1,3,5,7 to Output :

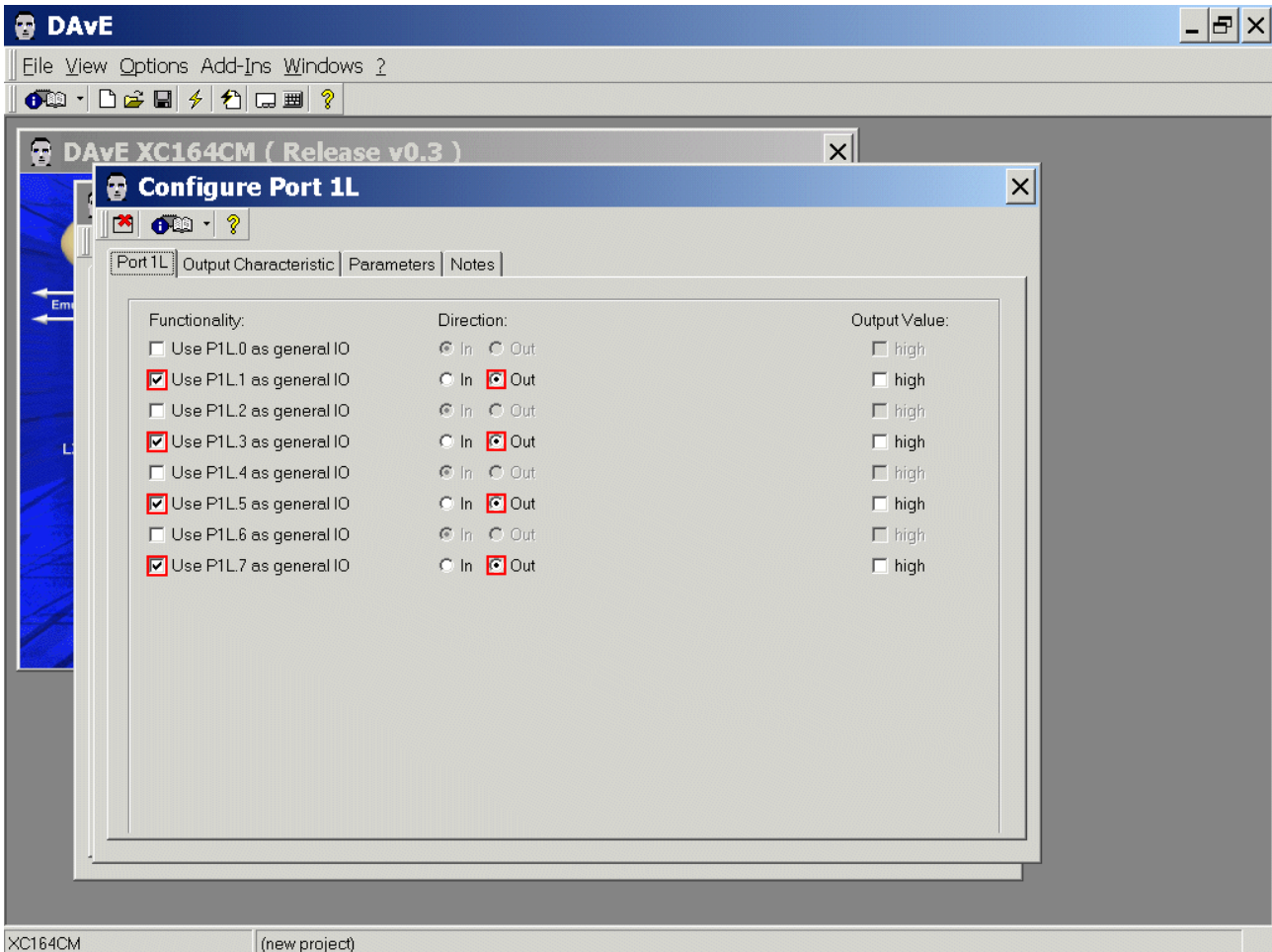
The configuration window can be opened by clicking the specific block/module.



Ports: **click** Configure Port 1L



- Port 1L: Functionality: **click** ✓ Use P1L.1 as general IO - Direction: **click** ⊙ Out
 Port 1L: Functionality: **click** ✓ Use P1L.3 as general IO - Direction: **click** ⊙ Out
 Port 1L: Functionality: **click** ✓ Use P1L.5 as general IO - Direction: **click** ⊙ Out
 Port 1L: Functionality: **click** ✓ Use P1L.7 as general IO - Direction: **click** ⊙ Out



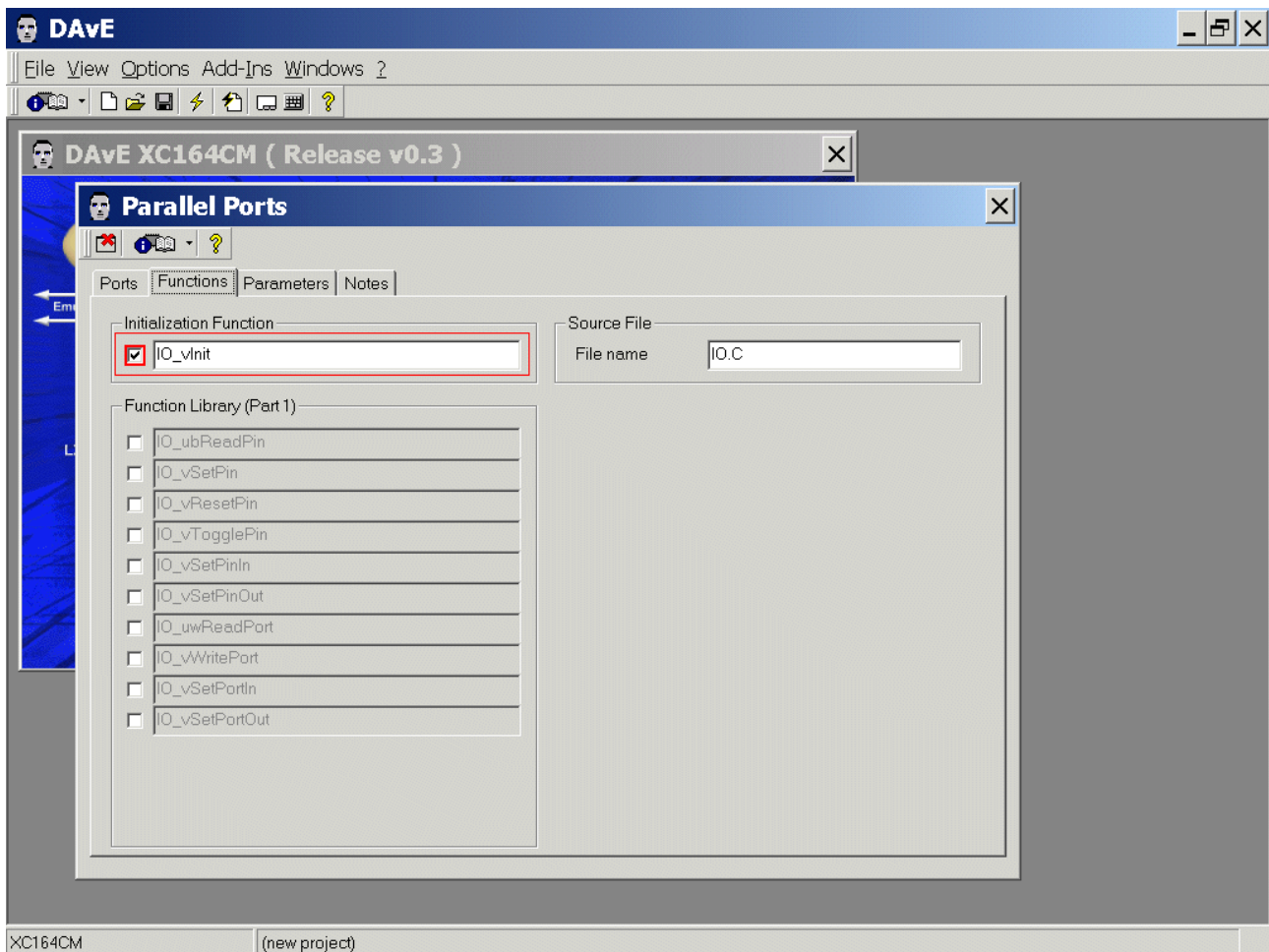
Output Characteristic: (do nothing)

Parameters: (do nothing)

Notes: If you wish, you can insert your comments here.

Exit this dialog now by clicking  the close button:

Functions: Initialization Functions: **click** ✓ IO_vInit



Parameters: (do nothing)

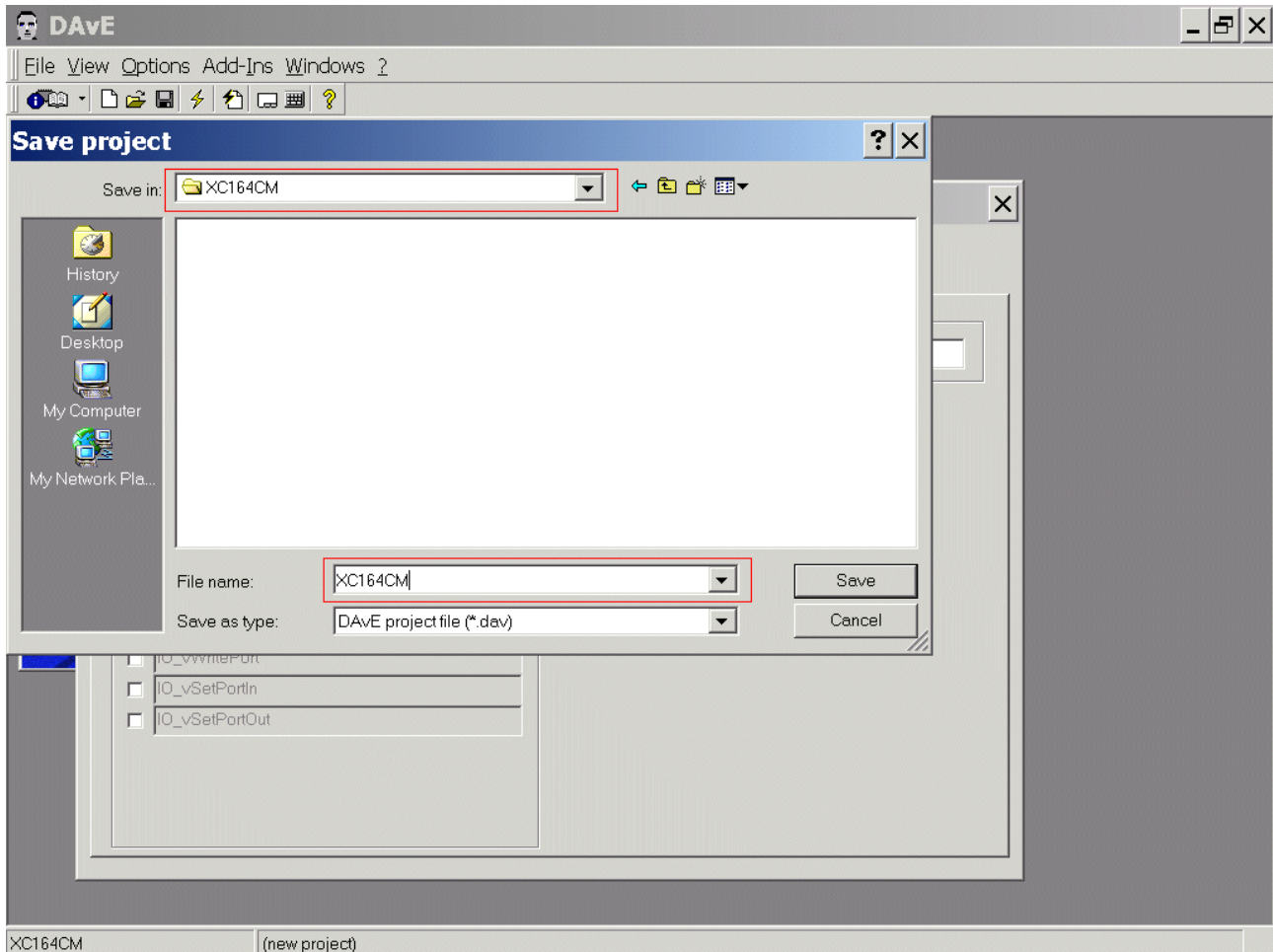
Notes: If you wish, you can insert your comments here.

Exit this dialog now by clicking  the close button:

Save the project:


File
Save

Save project: Save in: C:\XC164CM (create directory)
File name: XC164CM



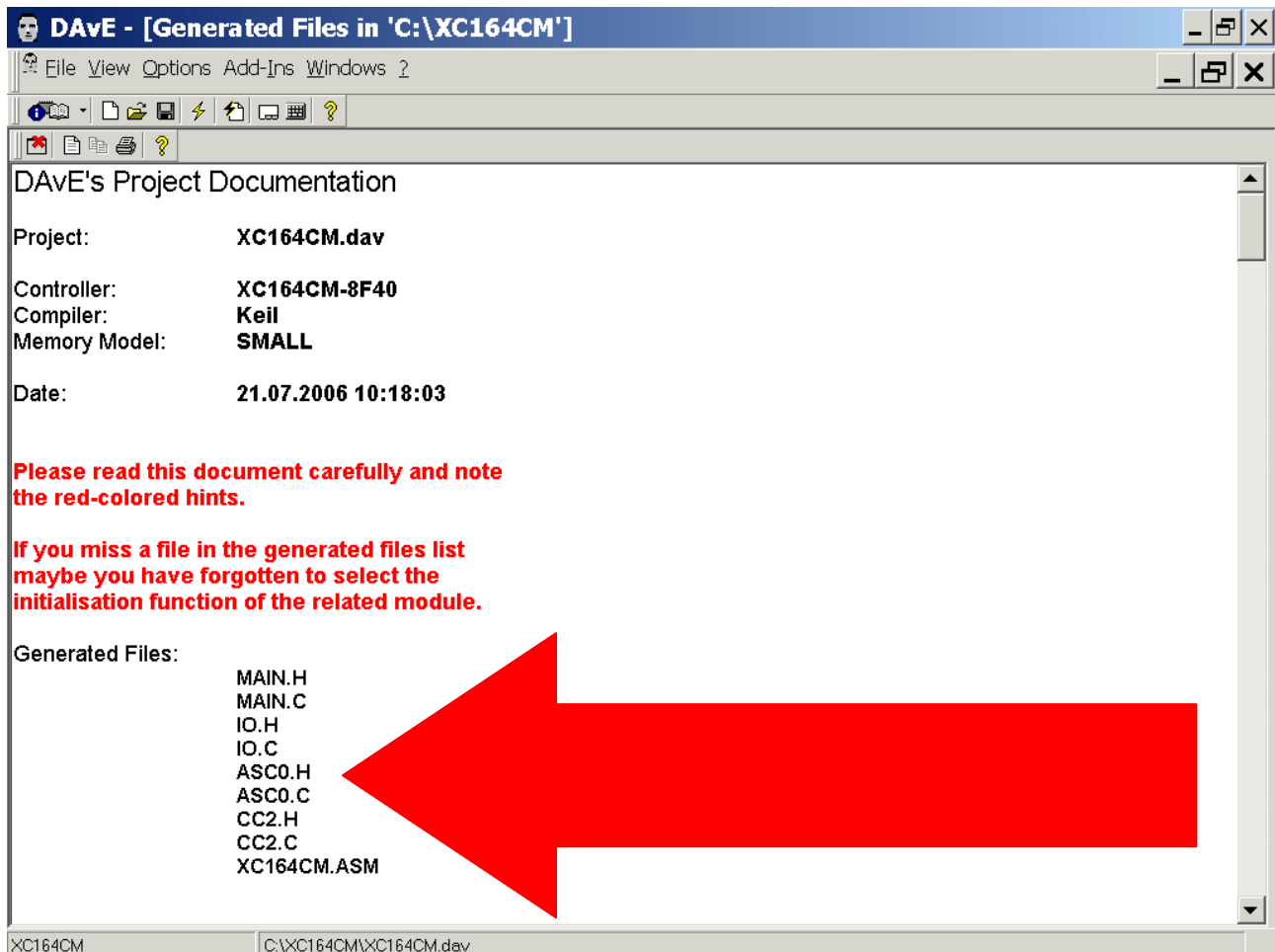
Save

Generate Code:

<p>File Generate Code</p>	<p>or click </p>
---	---



DAvE will show you all the files he has generated (File Viewer opens automatically).



File
Exit

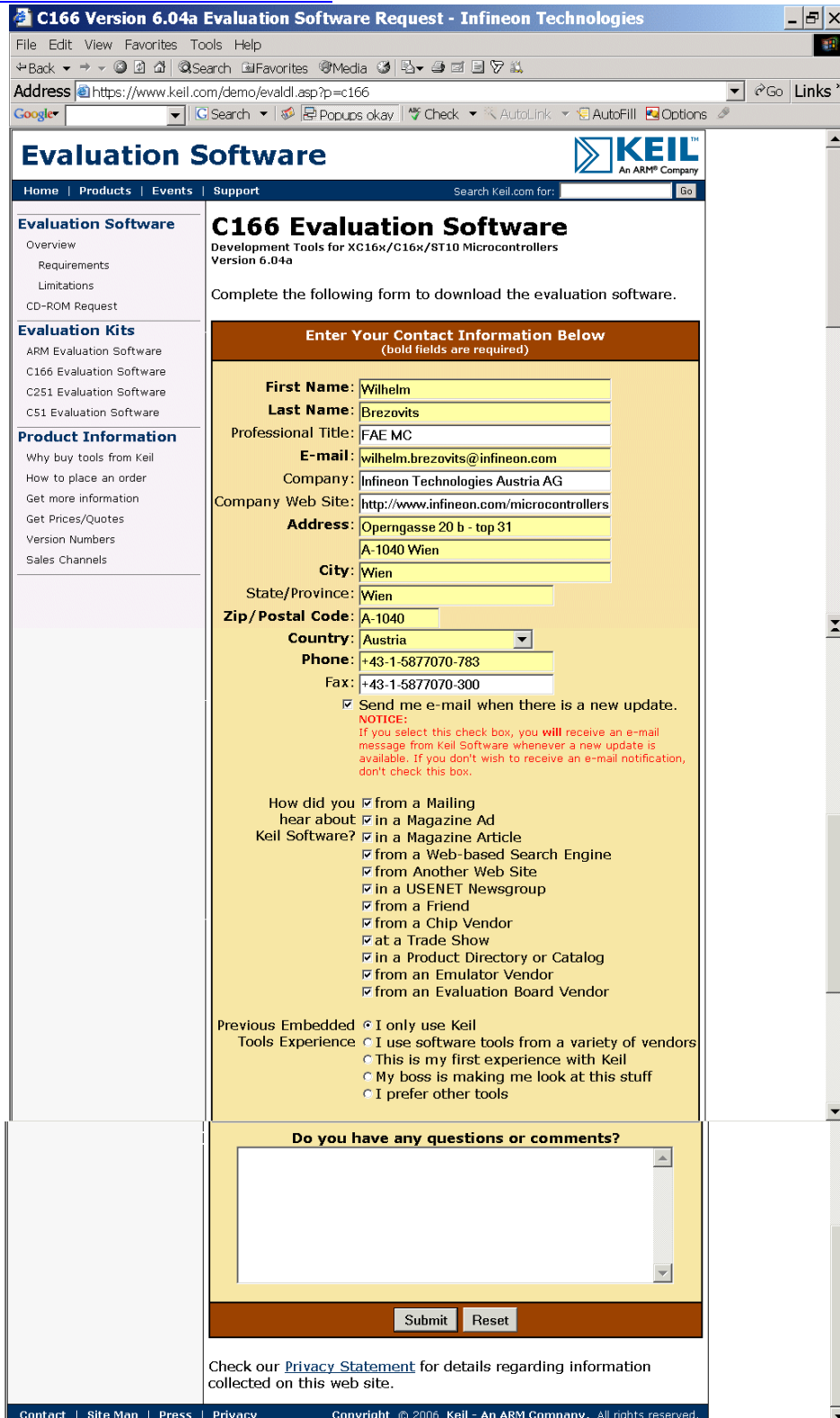
Save changes?

Click Yes

4.) Using of the KEIL - μ Vision 3 Development Tools:

Install the Tool chain - here you can download the Keil Development Tools:

<http://www.keil.com/demo/eval/c166.htm>



C166 Version 6.04a Evaluation Software Request - Infineon Technologies

File Edit View Favorites Tools Help

Address: <https://www.keil.com/demo/evaldl.asp?p=c166>

Evaluation Software

Home | Products | Events | Support

Search Keil.com for: Go

Evaluation Software

Overview
Requirements
Limitations
CD-ROM Request

Evaluation Kits

ARM Evaluation Software
C166 Evaluation Software
C251 Evaluation Software
C51 Evaluation Software

Product Information

Why buy tools from Keil
How to place an order
Get more information
Get Prices/Quotes
Version Numbers
Sales Channels

C166 Evaluation Software

Development Tools for XC16x/C16x/ST10 Microcontrollers
Version 6.04a

Complete the following form to download the evaluation software.

Enter Your Contact Information Below
(bold fields are required)

First Name: Wilhelm
Last Name: Brezovits
Professional Title: FAE MC
E-mail: wilhelm.brezovits@infineon.com
Company: Infineon Technologies Austria AG
Company Web Site: <http://www.infineon.com/microcontrollers>
Address: Opengasse 20 b - top 31
A-1040 Wien
City: Wien
State/Province: Wien
Zip/Postal Code: A-1040
Country: Austria
Phone: +43-1-5877070-783
Fax: +43-1-5877070-300

Send me e-mail when there is a new update.
NOTICE:
If you select this check box, you will receive an e-mail message from Keil Software whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.

How did you hear about Keil Software?
 from a Mailing
 in a Magazine Ad
 in a Magazine Article
 from a Web-based Search Engine
 from Another Web Site
 in a USENET Newsgroup
 from a Friend
 from a Chip Vendor
 at a Trade Show
 in a Product Directory or Catalog
 from an Emulator Vendor
 from an Evaluation Board Vendor

Previous Embedded Tools Experience
 I only use Keil
 I use software tools from a variety of vendors
 This is my first experience with Keil
 My boss is making me look at this stuff
 I prefer other tools

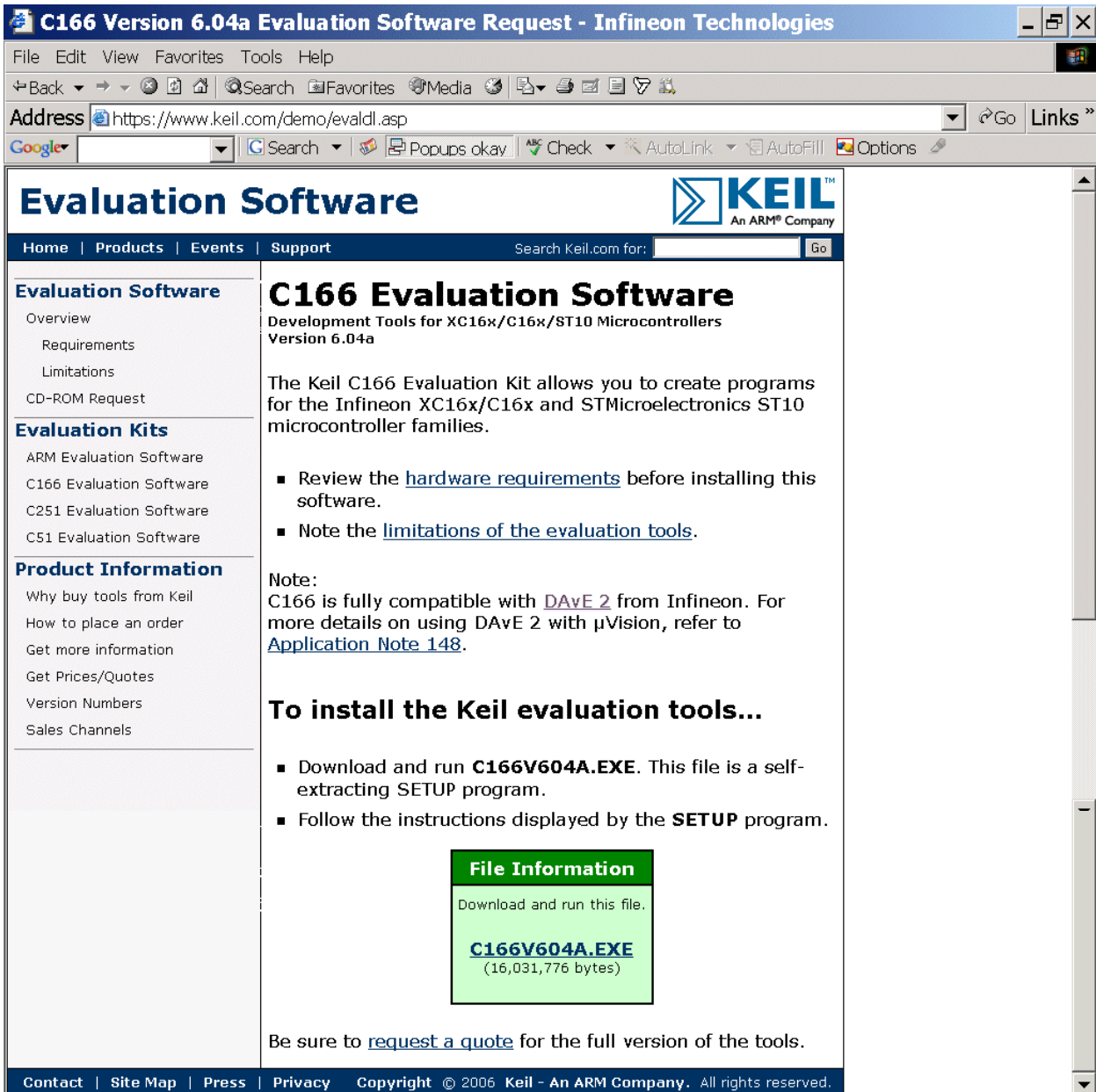
Do you have any questions or comments?

Submit Reset

Check our [Privacy Statement](#) for details regarding information collected on this web site.

Contact | Site Map | Press | Privacy Copyright © 2006 Keil - An ARM Company. All rights reserved.

Click Submit



C166 Version 6.04a Evaluation Software Request - Infineon Technologies

File Edit View Favorites Tools Help

Address <https://www.keil.com/demo/evaldl.asp>

Evaluation Software

Home | Products | Events | Support Search Keil.com for: Go

Evaluation Software

- Overview
- Requirements
- Limitations
- CD-ROM Request

Evaluation Kits

- ARM Evaluation Software
- C166 Evaluation Software
- C251 Evaluation Software
- C51 Evaluation Software

Product Information

- Why buy tools from Keil
- How to place an order
- Get more information
- Get Prices/Quotes
- Version Numbers
- Sales Channels

C166 Evaluation Software

Development Tools for XC16x/C16x/ST10 Microcontrollers
Version 6.04a

The Keil C166 Evaluation Kit allows you to create programs for the Infineon XC16x/C16x and STMicroelectronics ST10 microcontroller families.

- Review the [hardware requirements](#) before installing this software.
- Note the [limitations of the evaluation tools](#).

Note:
C166 is fully compatible with [DAvE 2](#) from Infineon. For more details on using DAVe 2 with μ Vision, refer to [Application Note 148](#).

To install the Keil evaluation tools...

- Download and run **C166V604A.EXE**. This file is a self-extracting SETUP program.
- Follow the instructions displayed by the **SETUP** program.

File Information

Download and run this file.

[C166V604A.EXE](#)
(16,031,776 bytes)

Be sure to [request a quote](#) for the full version of the tools.

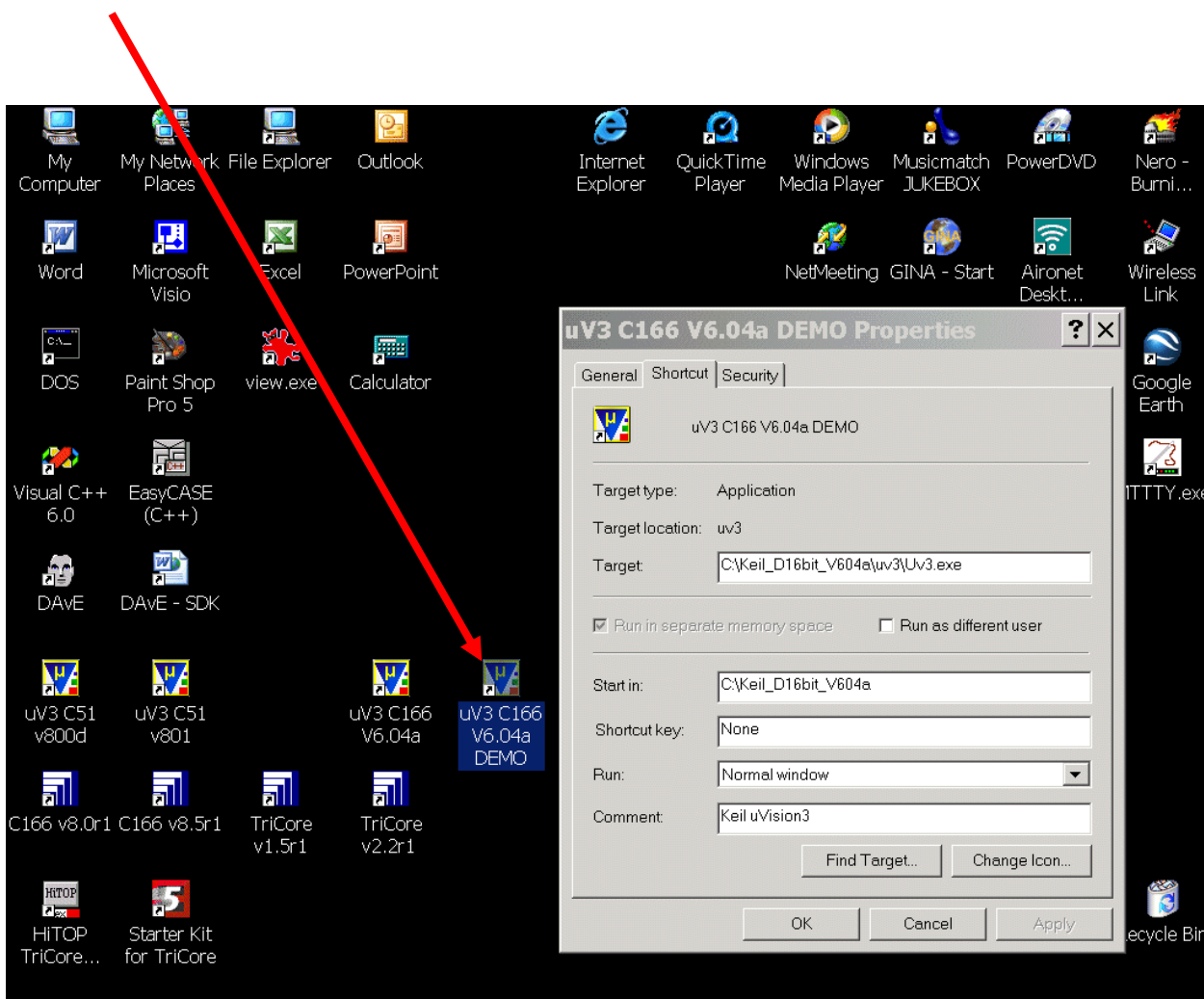
Contact | Site Map | Press | Privacy | Copyright © 2006 Keil - An ARM Company. All rights reserved.

Execute **C166V604a.EXE** (- or higher)
and choose e.g. **C:\Keil_D16bit_V604a** for the installation path



Start Keil μ Vision3 and open the DAVE Project:

Double-click



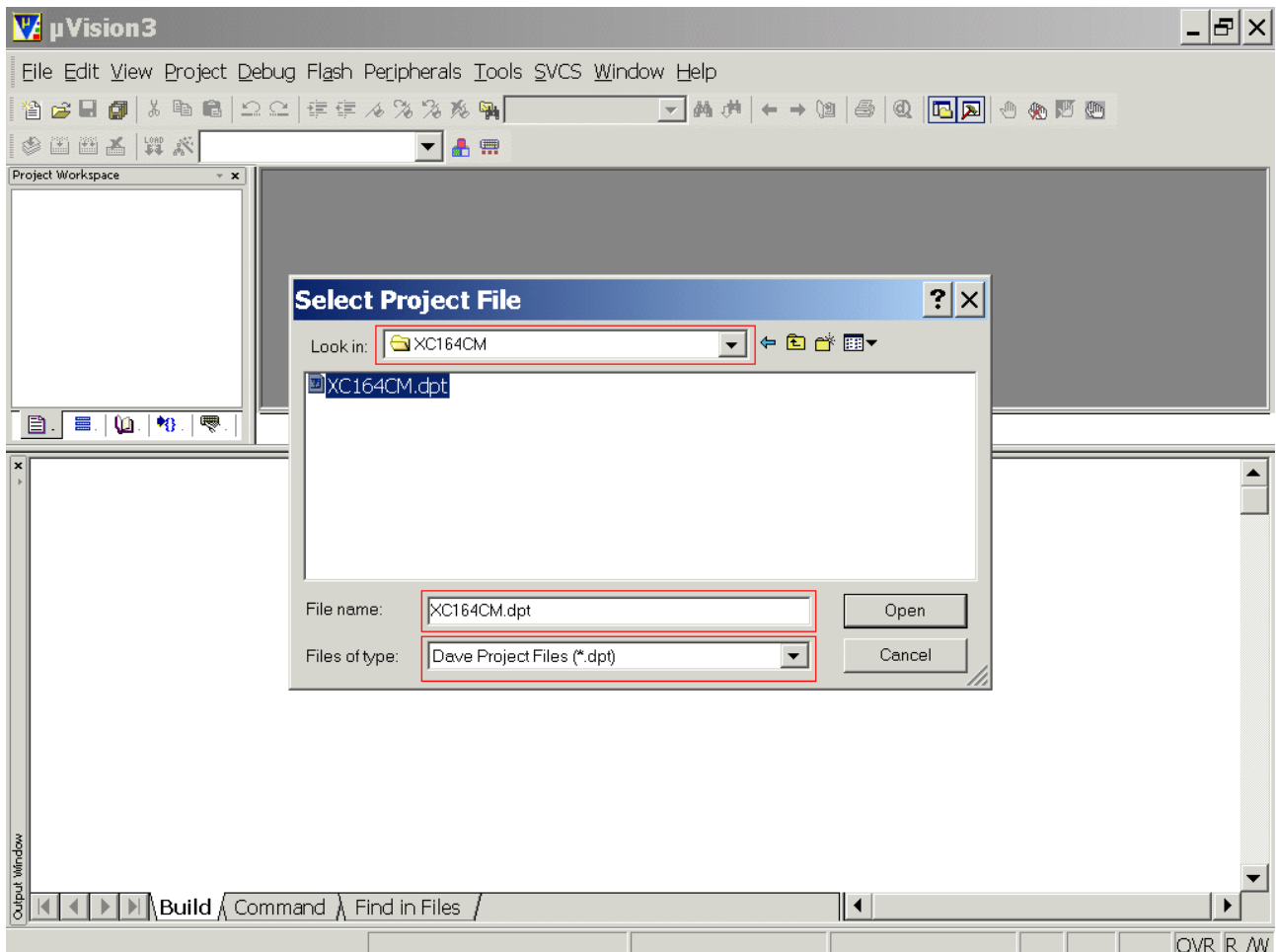
If you see an open project – close it: **Project - Close Project**

Project - Open Project

Look in: **C:\XC164CM**

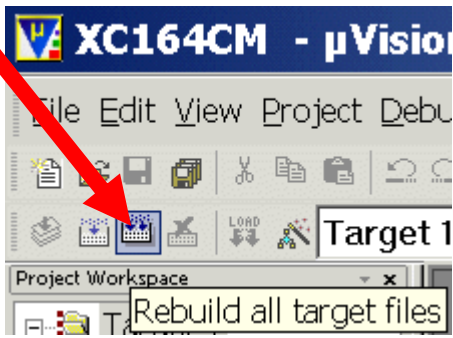
choose: File of type: **Dave Project Files**

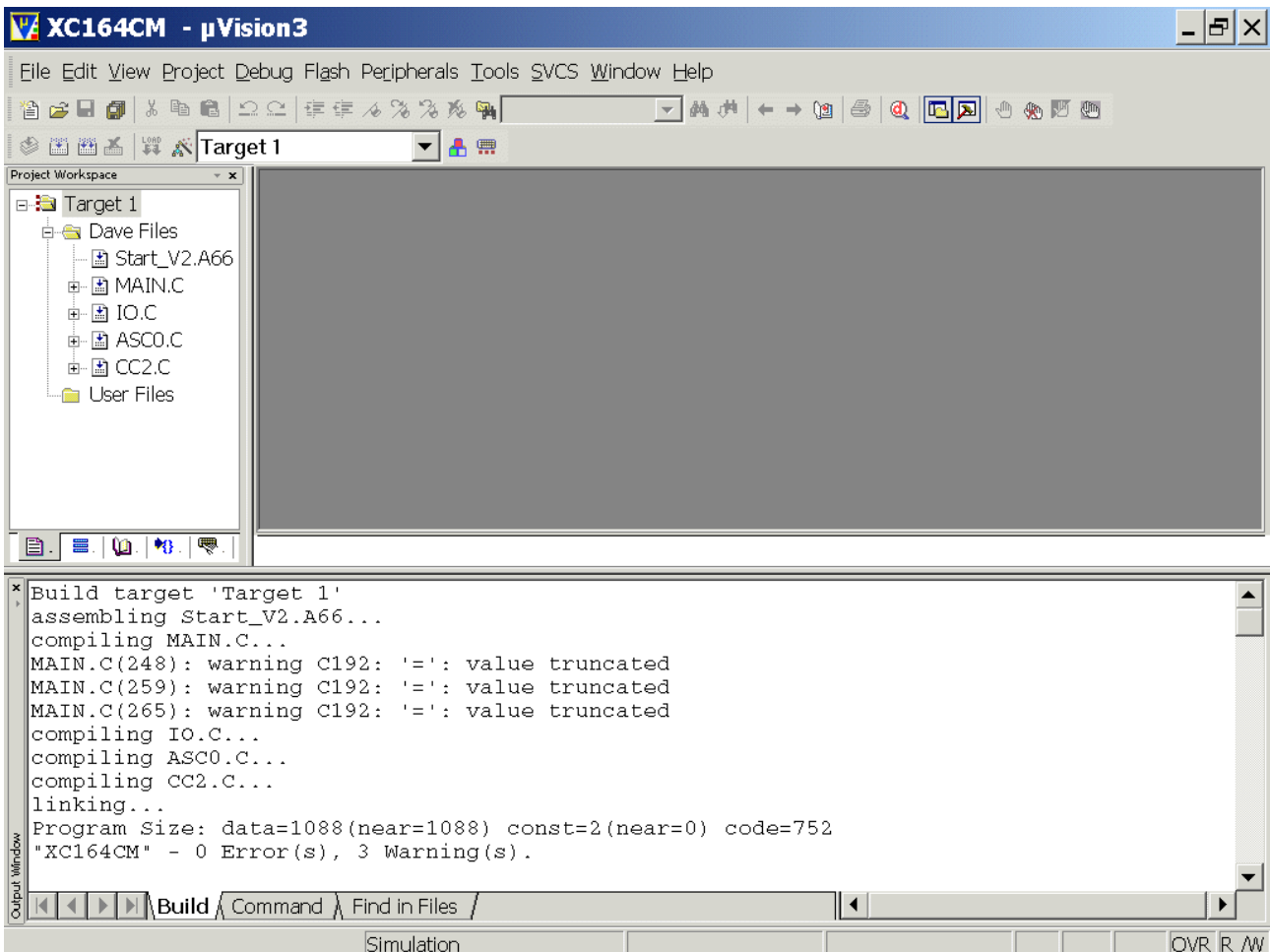
File name: **XC164CM.dpt**




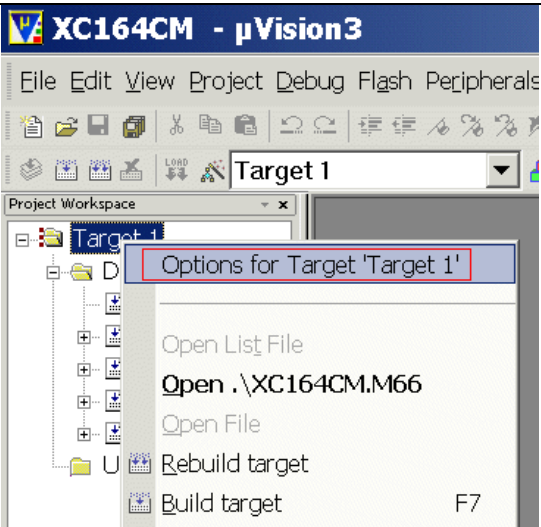
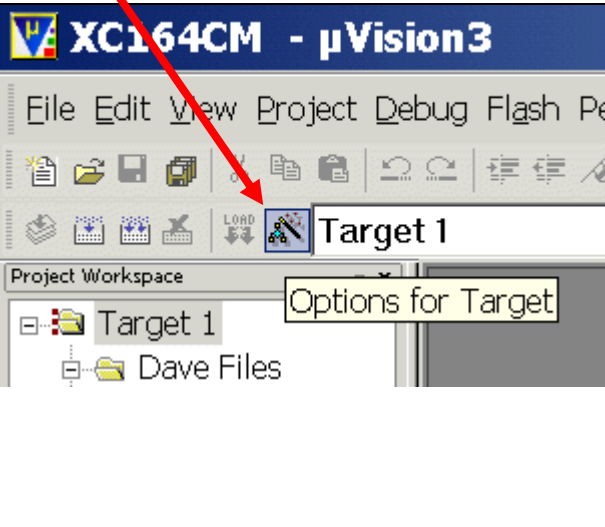
Open

Generate „make“-File:

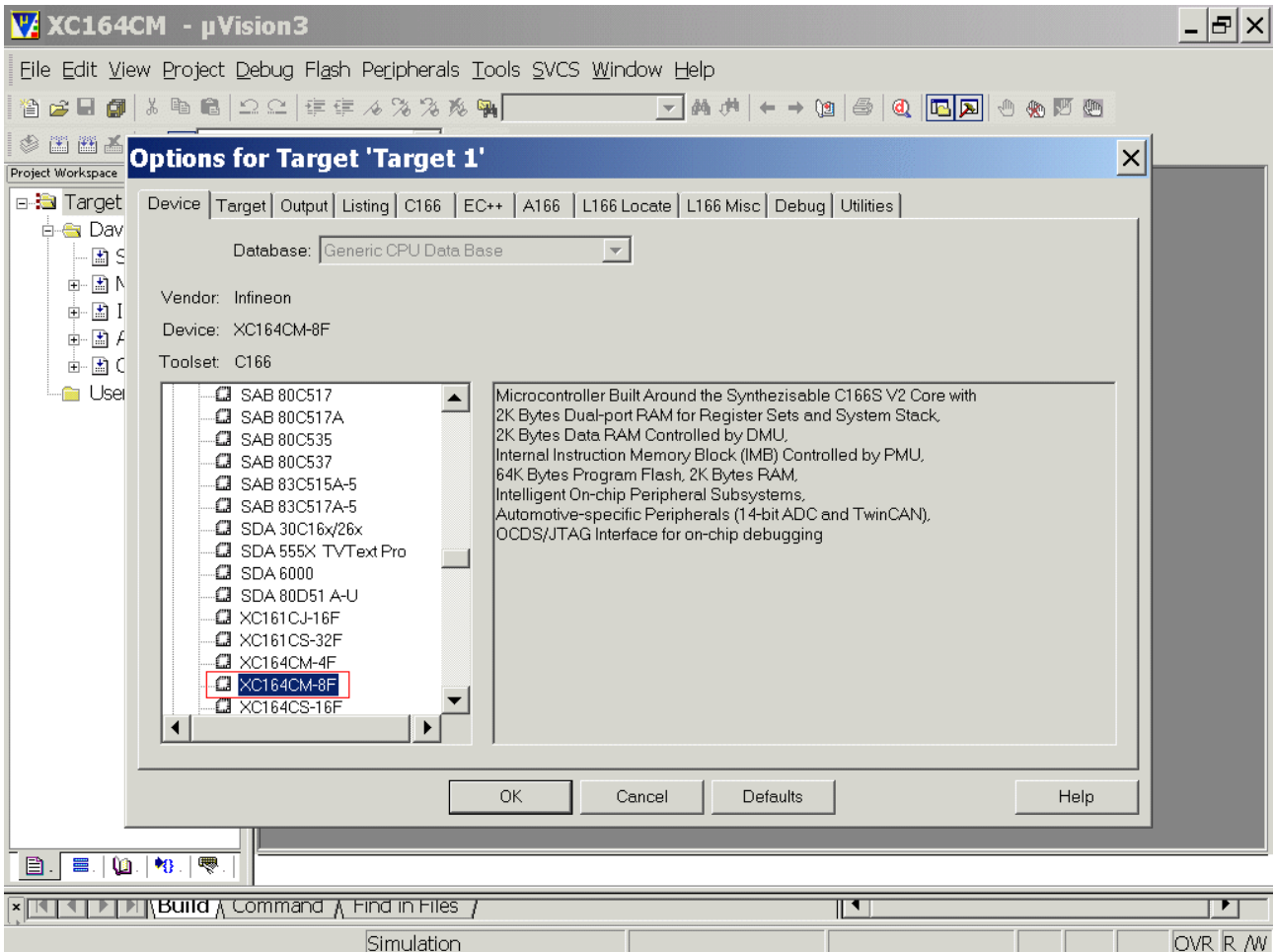
<p>Project – Rebuild all target files</p>	<p>or</p>	<p>click</p> 
---	-----------	---



Configure Compiler, Assembler, Linker, Locater, Hex-Converter and Build – Control:

<p>mouse position: (Project Workspace, Files): Target1 click right mouse button Options for Target 'Target1'</p>	<p>or</p>	<p>click</p> 
		

Device: **check** XC164CM-8F



- Target: **insert/check** Clock (MHz): 8
- Target: **click/check** ✓ Use On-chip ROM (64KB)
- Target: **click/check** ✓ Use On-chip XRAM

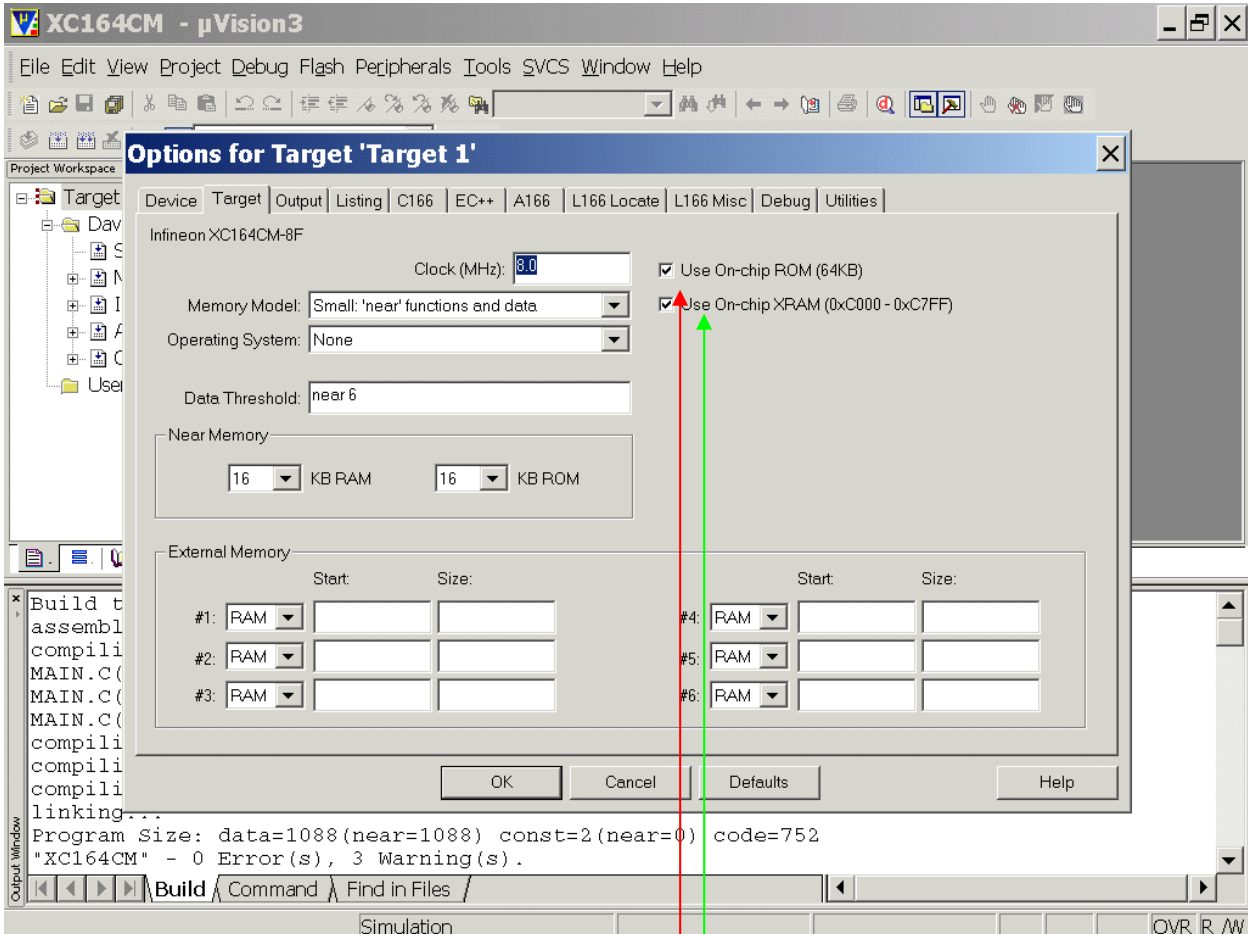
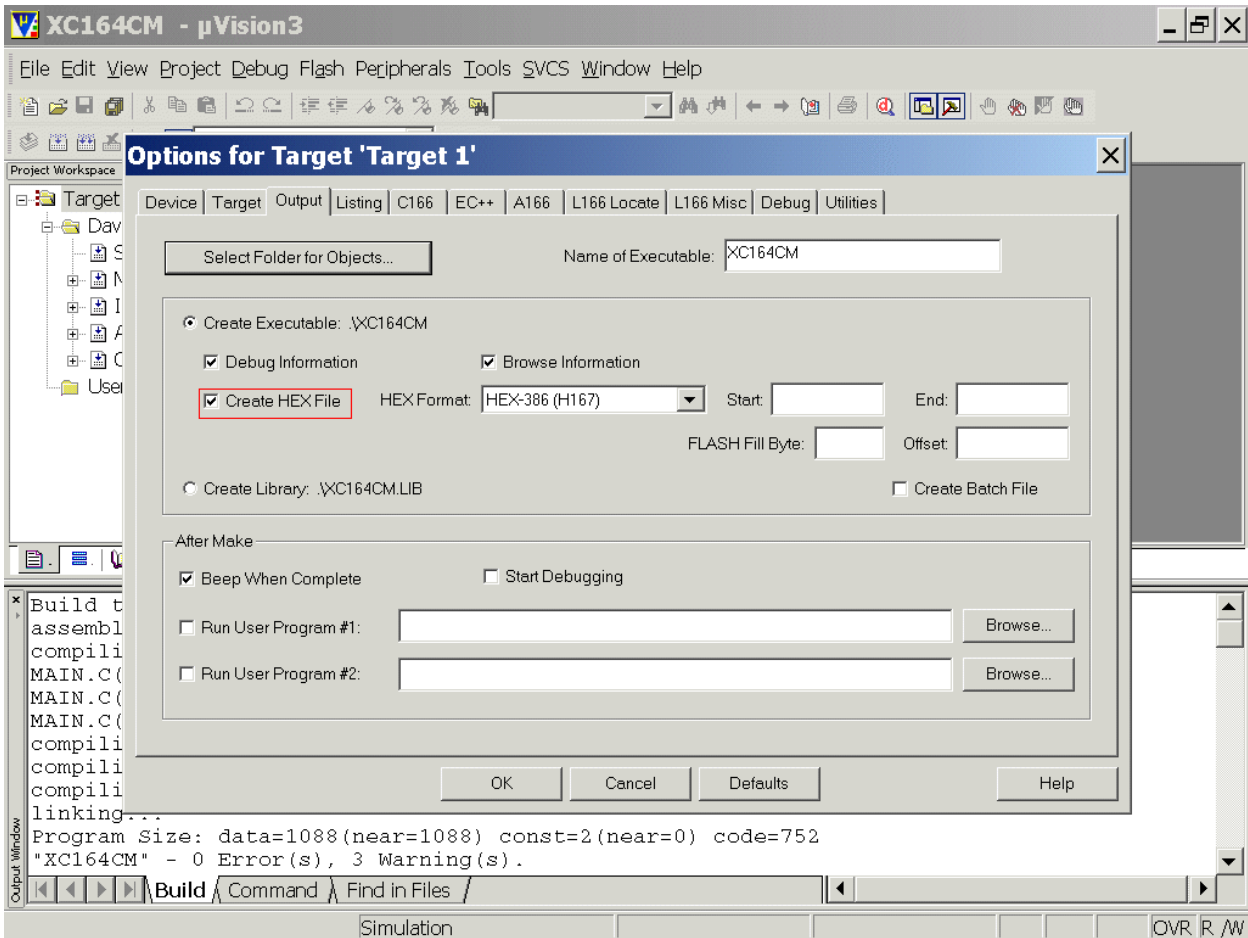


Table 3-1 XC164CM Memory Map

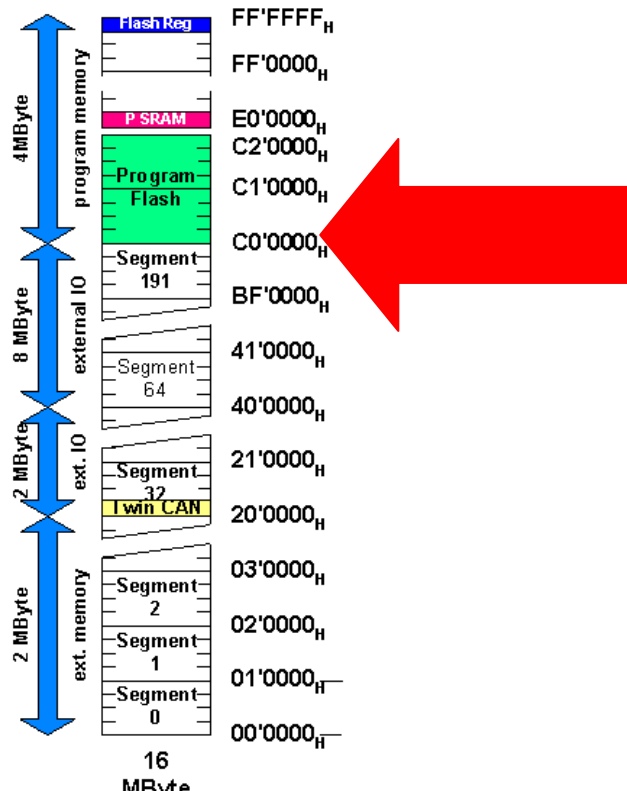
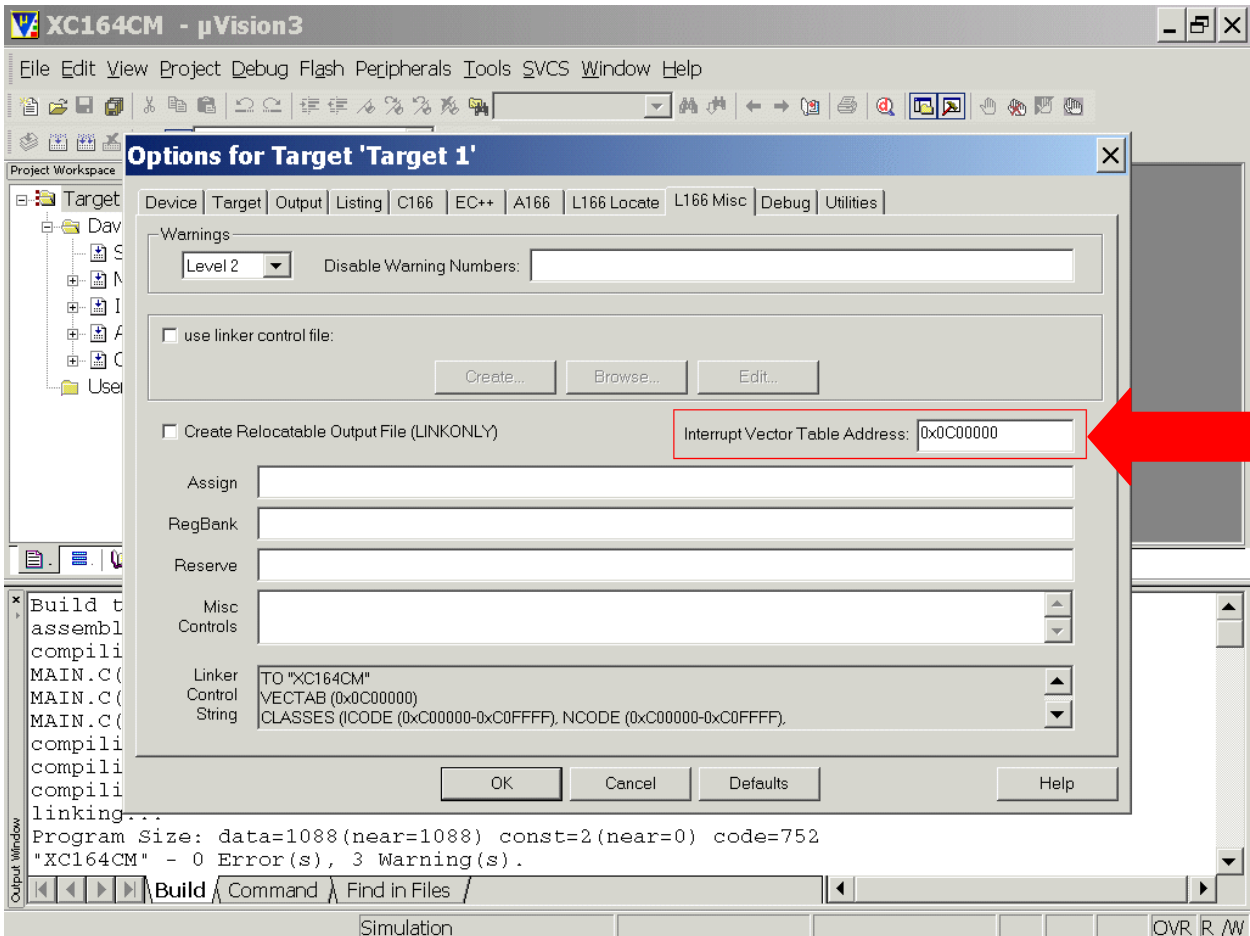
Address Area	Start Loc.	End Loc.	Area Size ¹⁾	Notes
Flash register space	FF'F000 _H	FF'FFFF _H	4 Kbytes	²⁾
Reserved (Acc. trap)	F8'0000 _H	FF'FFFF _H	508 Kbytes	
Reserved for PSRAM	E0'0800 _H	F7'FFFF _H	< 1.5 Mbytes	Minus PSRAM
Program SRAM	E0'0000 _H	E0'07FF _H	2 Kbytes	
Reserved for pr. mem.	C1'0000 _H	DF'FFFF _H	< 2 Mbytes	Minus Flash
Program Flash	C0'0000 _H	C0'FFFF _H	64 Kbytes	–
Reserved	40'0000 _H	BF'FFFF _H	8 Mbytes	–
Reserved	20'0800 _H	3F'FFFF _H	< 2 Mbytes	Minus TwinCAN
TwinCAN registers	20'0000 _H	20'07FF _H	2 Kbytes	Accessed via EBC
Reserved	01'0000 _H	1F'FFFF _H	< 2 Mbytes	Minus segment 0
SFR area	00'FE00 _H	00'FFFF _H	0.5 Kbyte	–
Dual-Port RAM	00'F600 _H	00'FDFF _H	2 Kbytes	–
Reserved for DPRAM	00'F200 _H	00'F5FF _H	1 Kbyte	–
ESFR area	00'F000 _H	00'F1FF _H	0.5 Kbyte	–
XSFR area	00'E000 _H	00'EFFF _H	4 Kbytes	–
Reserved	00'C800 _H	00'DFFF _H	6 Kbytes	–
Data SRAM	00'C000 _H	00'C7FF _H	2 Kbytes	XC164CM-8F only
Reserved for DSRAM	00'8000 _H	00'BFFF _H	16 Kbytes	–
Reserved	00'0000 _H	00'7FFF _H	32 Kbytes	–

Output: **click** ✓ Create HEX File



- Listing (do nothing)
- C166 (do nothing)
- EC++ (do nothing)
- A166 (do nothing)
- L166 Locate (do nothing)

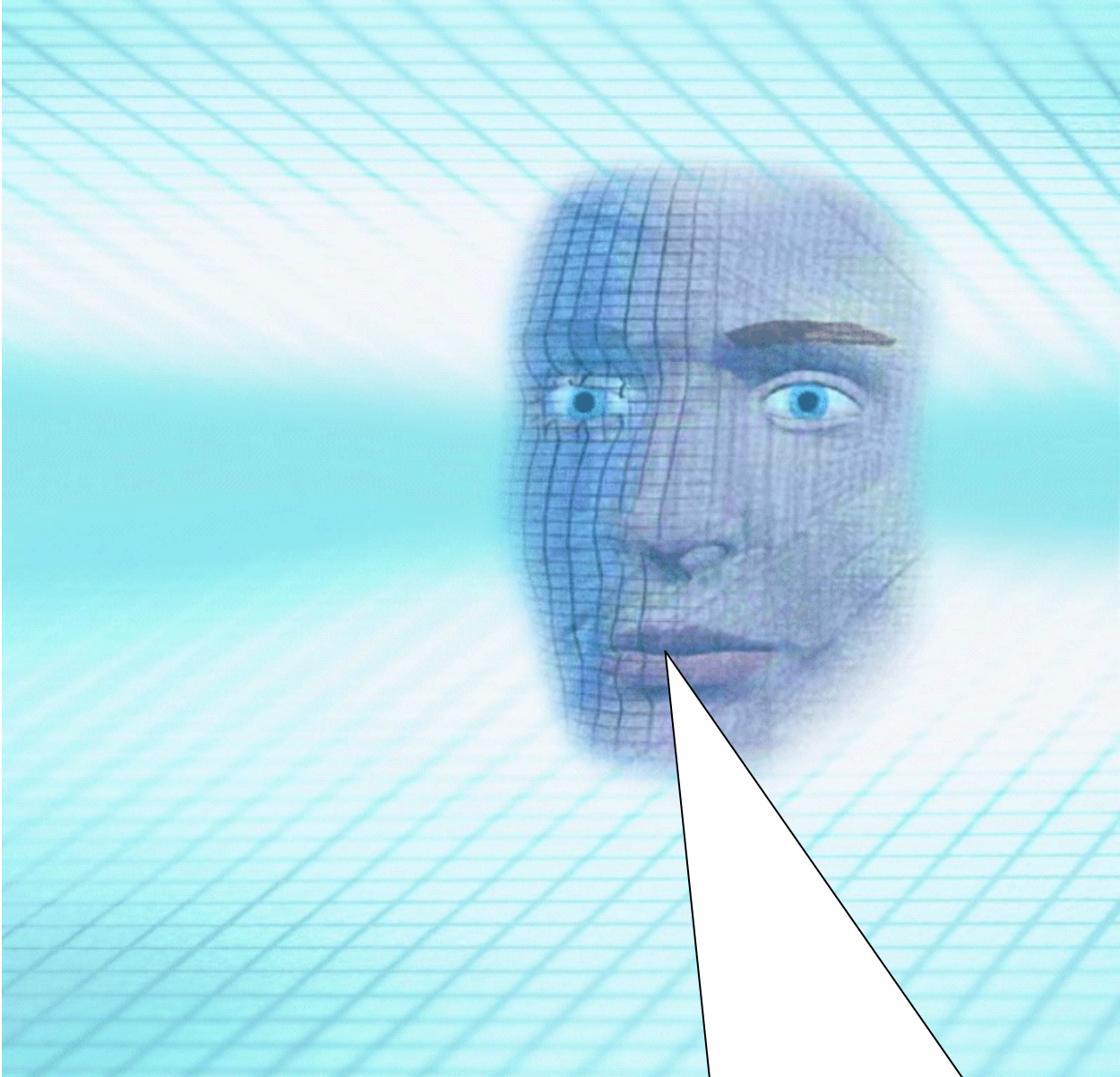
L166 Misc: Interrupt Vector Table Address: insert 0x0C00000



Debug (do nothing)
Utilities (do nothing)

OK

Insert your application specific program:



Note:

DAvE doesn't change code which is inserted between '`// USER CODE BEGIN`' and '`// USER CODE END`'. Therefore, whenever adding code to DAvE's generated code, write it between '`// USER CODE BEGIN`' and '`// USER CODE END`'.

If you wish to change DAvE's generated code or add code outside these 'USER CODE' sections you will have to insert/modify your changes each time after letting DAvE regenerate code!

Double click **MAIN.C** and insert Global Variables:

```
char menu[] =
"\r\n\r\n\r\n"
"1 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 ON\r\n"
"2 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 OFF\r\n"
"3 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 blinking\r\n"
"  \r\n";

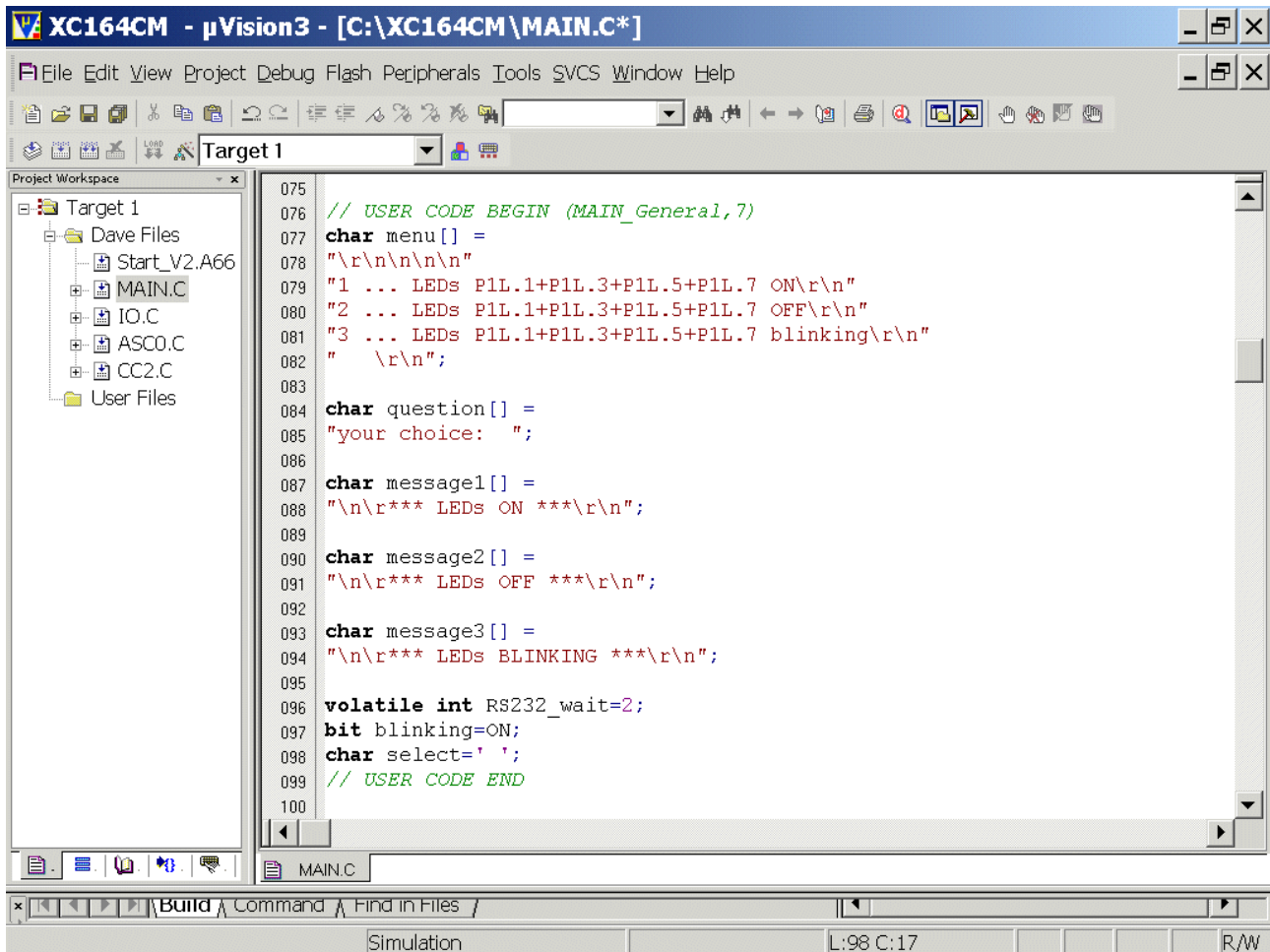
char question[] =
"your choice: ";

char message1[] =
"\n\r*** LEDs ON ***\r\n";

char message2[] =
"\n\r*** LEDs OFF ***\r\n";

char message3[] =
"\n\r*** LEDs BLINKING ***\r\n";

volatile int RS232_wait=2;
bit blinking=ON;
char select=' ';
```



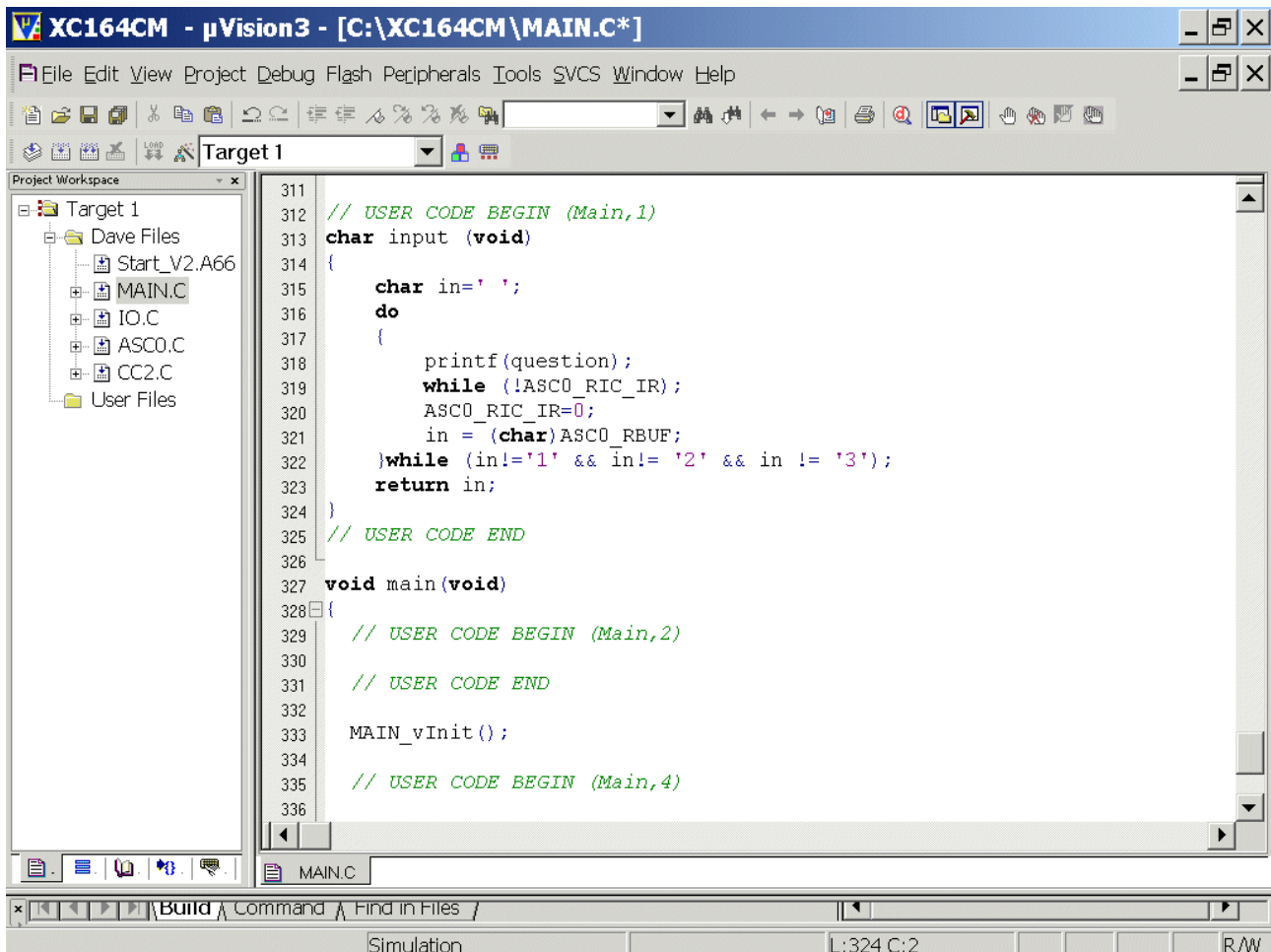
```

075
076 // USER CODE BEGIN (MAIN_General,7)
077 char menu[] =
078 "\r\n\r\n\r\n"
079 "1 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 ON\r\n"
080 "2 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 OFF\r\n"
081 "3 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 blinking\r\n"
082 "  \r\n";
083
084 char question[] =
085 "your choice: ";
086
087 char message1[] =
088 "\n\r*** LEDs ON ***\r\n";
089
090 char message2[] =
091 "\n\r*** LEDs OFF ***\r\n";
092
093 char message3[] =
094 "\n\r*** LEDs BLINKING ***\r\n";
095
096 volatile int RS232_wait=2;
097 bit blinking=ON;
098 char select=' ';
099 // USER CODE END
100

```

Double click **MAIN.C** and insert the function **input()**

```
char input (void)
{
    char in=' ';
    do
    {
        printf(question);
        while (!ASC0_RIC_IR);
        ASC0_RIC_IR=0;
        in = (char)ASC0_RBUF;
    }while (in!='1' && in!= '2' && in != '3');
    return in;
}
```



The screenshot shows the µVision3 IDE interface for the XC164CM project. The Project Workspace on the left shows a tree view with 'Target 1' containing 'Dave Files' (Start_V2.A66, MAIN.C, IO.C, ASC0.C, CC2.C) and 'User Files'. The main editor window displays the code for MAIN.C, with the 'input()' function inserted between lines 311 and 324. The code is as follows:

```
311 // USER CODE BEGIN (Main,1)
312 char input (void)
313 {
314     char in=' ';
315     do
316     {
317         printf(question);
318         while (!ASC0_RIC_IR);
319         ASC0_RIC_IR=0;
320         in = (char)ASC0_RBUF;
321     }while (in!='1' && in!= '2' && in != '3');
322     return in;
323 }
324 // USER CODE END
325
326 void main(void)
327 {
328     // USER CODE BEGIN (Main,2)
329     // USER CODE END
330
331     MAIN_vInit();
332
333     // USER CODE BEGIN (Main,4)
334
335
336
```

The status bar at the bottom indicates 'Simulation' mode, 'L:324 C:2', and 'R/W' permissions.

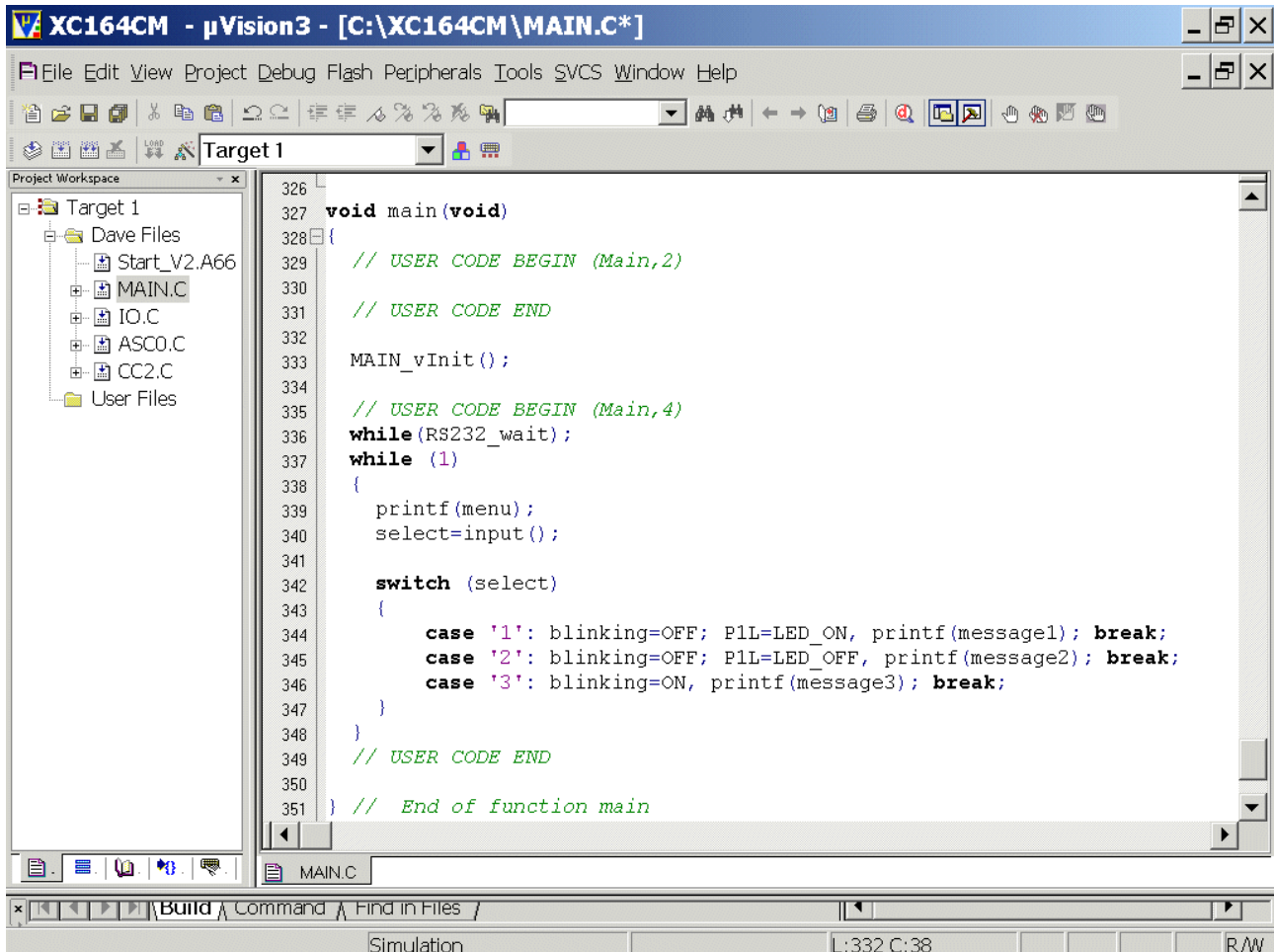
Double click **MAIN.C** and **insert** the following code into the **main** function:

```

while(RS232_wait);
while (1)
{
    printf(menu);
    select=input();

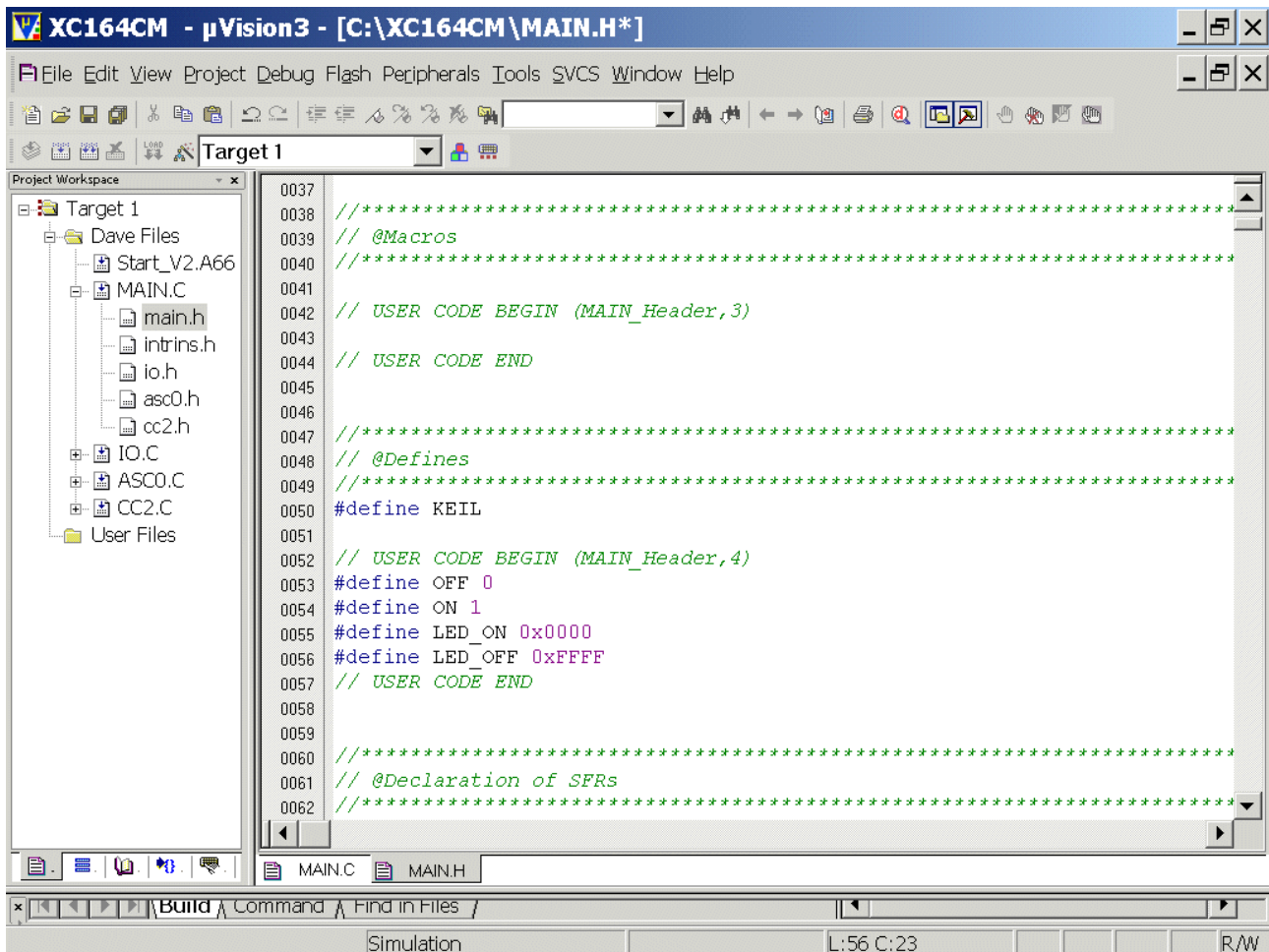
    switch (select)
    {
        case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
        case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
        case '3': blinking=ON, printf(message3); break;
    }
}

```



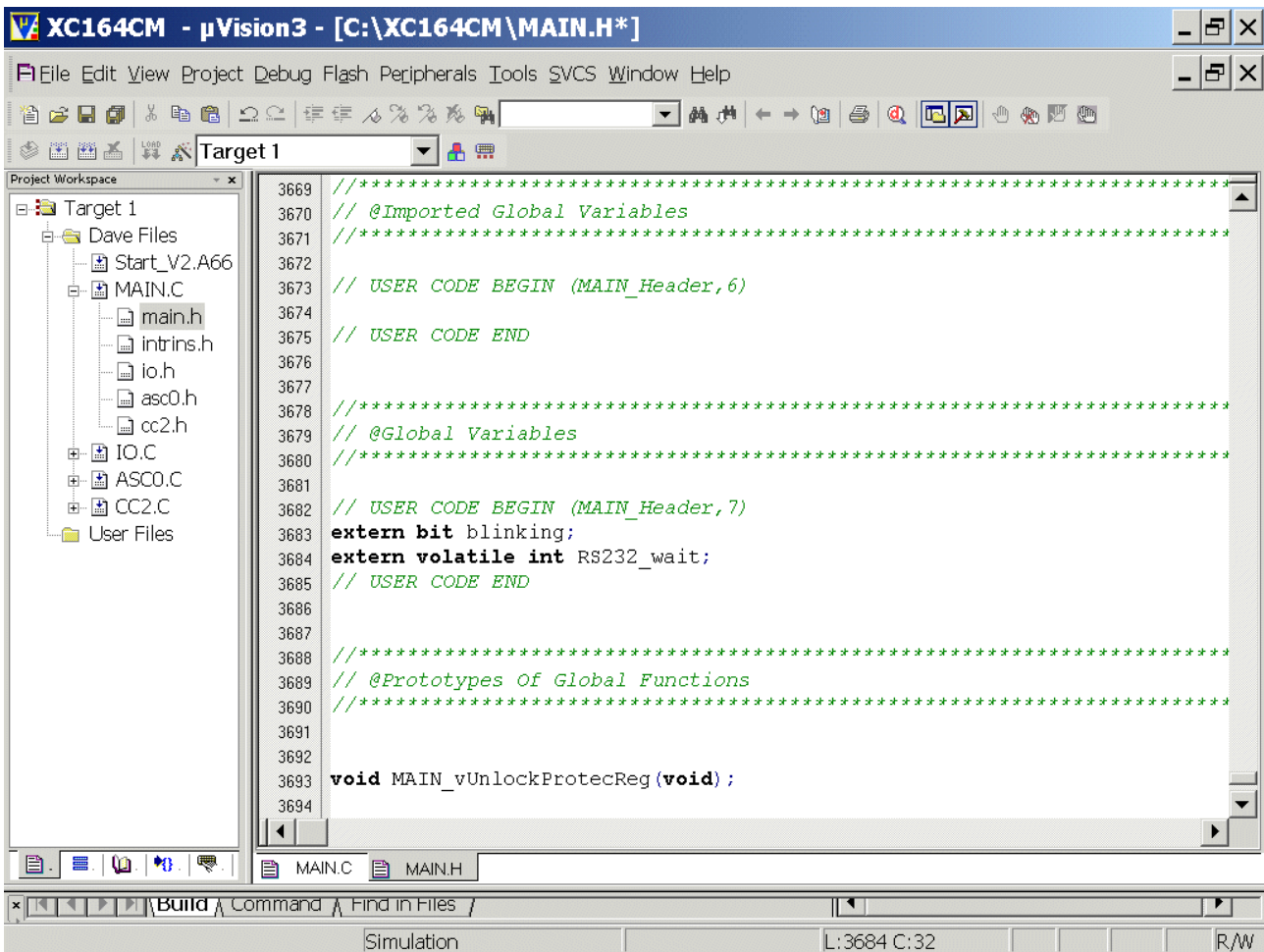
Double click **Main.h** and **insert** the following Defines:

```
#define OFF 0
#define ON 1
#define LED_ON 0x0000
#define LED_OFF 0xFFFF
```



Double click **Main.h** and insert extern-declaration "Global Variables"

```
extern bit blinking;
extern volatile int RS232_wait;
```



The screenshot shows the µVision3 IDE interface for the XC164CM project. The Project Workspace on the left shows the file structure with 'main.h' selected under 'MAIN.C'. The main editor window displays the content of 'MAIN.H' with the following code:

```

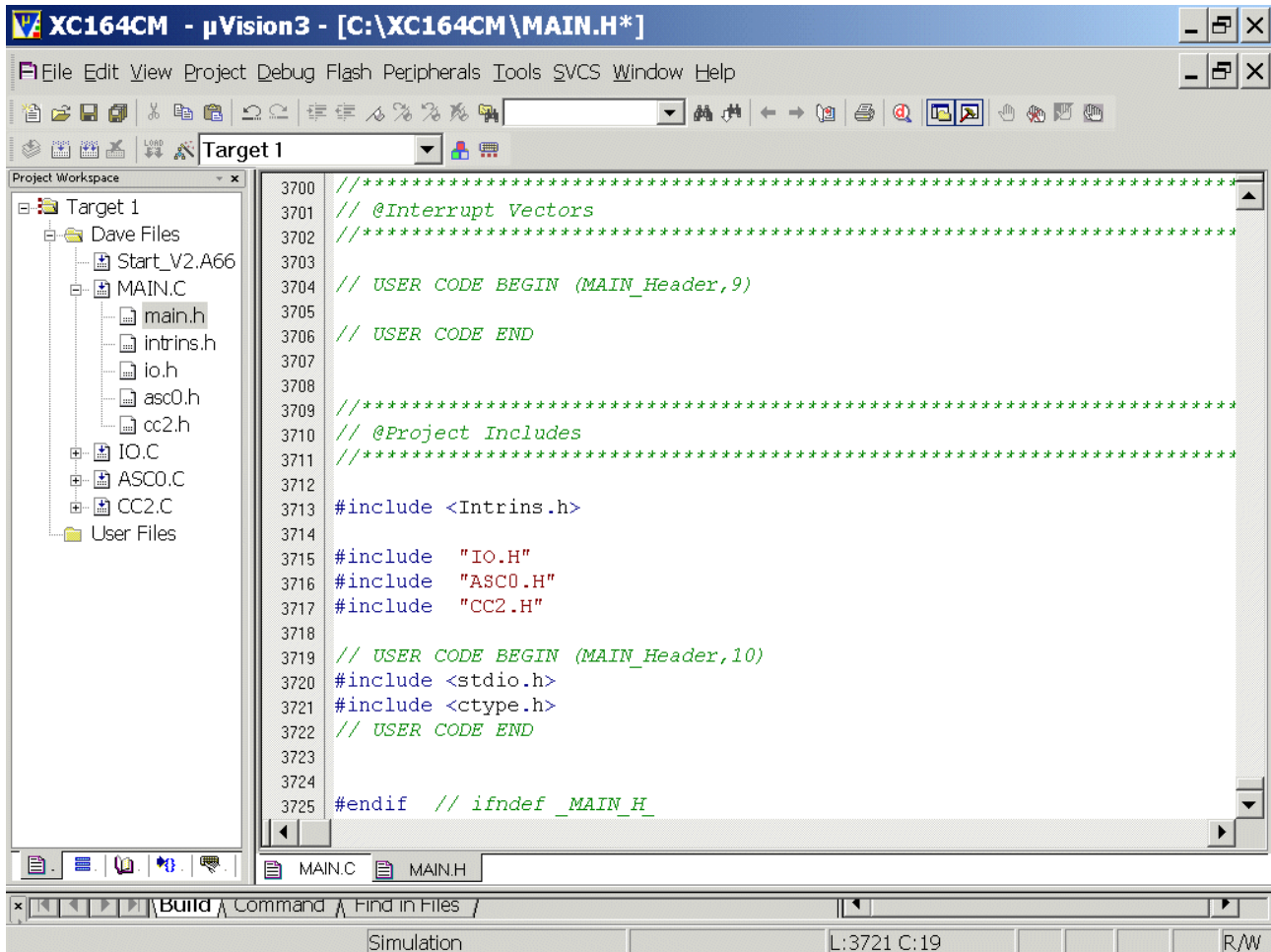
3669 //*****
3670 // @Imported Global Variables
3671 //*****
3672 // USER CODE BEGIN (MAIN_Header,6)
3673
3674 // USER CODE END
3675
3676
3677
3678 //*****
3679 // @Global Variables
3680 //*****
3681
3682 // USER CODE BEGIN (MAIN_Header,7)
3683 extern bit blinking;
3684 extern volatile int RS232_wait;
3685 // USER CODE END
3686
3687
3688 //*****
3689 // @Prototypes Of Global Functions
3690 //*****
3691
3692
3693 void MAIN_vUnlock ProtecReg (void) ;
3694

```

The status bar at the bottom indicates 'Simulation' mode and the cursor is at line 3684, column 32.

Double click **Main.h** and insert Includes (for printf()):

```
#include <stdio.h>
#include <ctype.h>
```



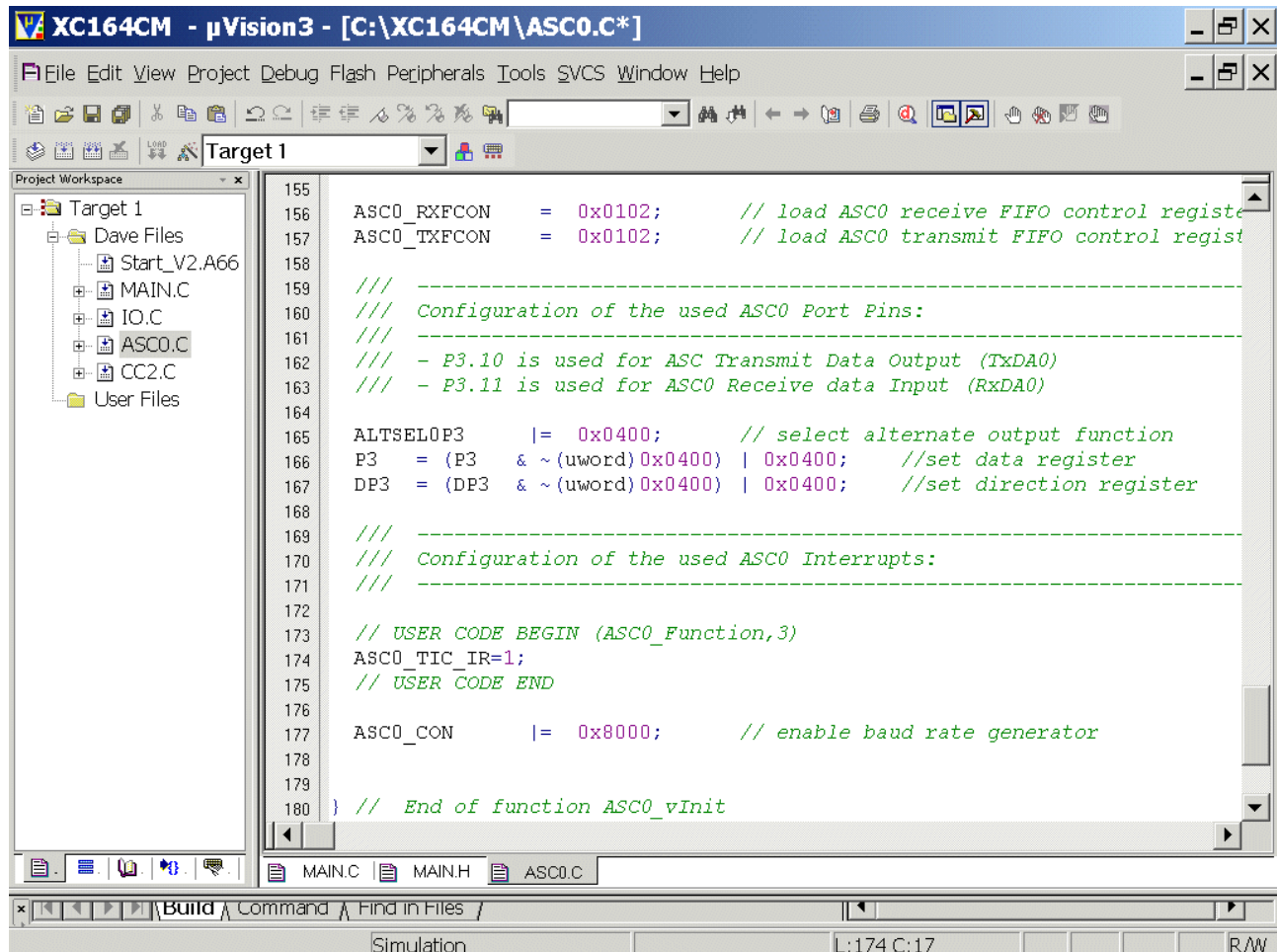
The screenshot shows the µVision3 IDE interface. The title bar reads "XC164CM - µVision3 - [C:\XC164CM\MAIN.H*]". The menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and development. The Project Workspace on the left shows a tree view with "Target 1" containing "Dave Files", "Start_V2.A66", "MAIN.C", "main.h", "intrins.h", "io.h", "asc0.h", "cc2.h", "IO.C", "ASC0.C", "CC2.C", and "User Files". The main editor window displays the content of MAIN.H with line numbers 3700 to 3725. The code includes comments for interrupt vectors and user code boundaries, and includes the following headers:

```
3700 //*****
3701 // @Interrupt Vectors
3702 //*****
3703
3704 // USER CODE BEGIN (MAIN_Header,9)
3705
3706 // USER CODE END
3707
3708
3709 //*****
3710 // @Project Includes
3711 //*****
3712
3713 #include <Intrins.h>
3714
3715 #include "IO.H"
3716 #include "ASC0.H"
3717 #include "CC2.H"
3718
3719 // USER CODE BEGIN (MAIN_Header,10)
3720 #include <stdio.h>
3721 #include <ctype.h>
3722 // USER CODE END
3723
3724
3725 #endif // ifndef _MAIN_H
```

The status bar at the bottom shows "Simulation", "Command", "Find in Files", "L:3721 C:19", and "R/W".

Double click `ASC0.C` and insert code into the `ASC0_vInit()` function: (to start `printf()`):

```
ASC0_TIC_IR=1;
```



```

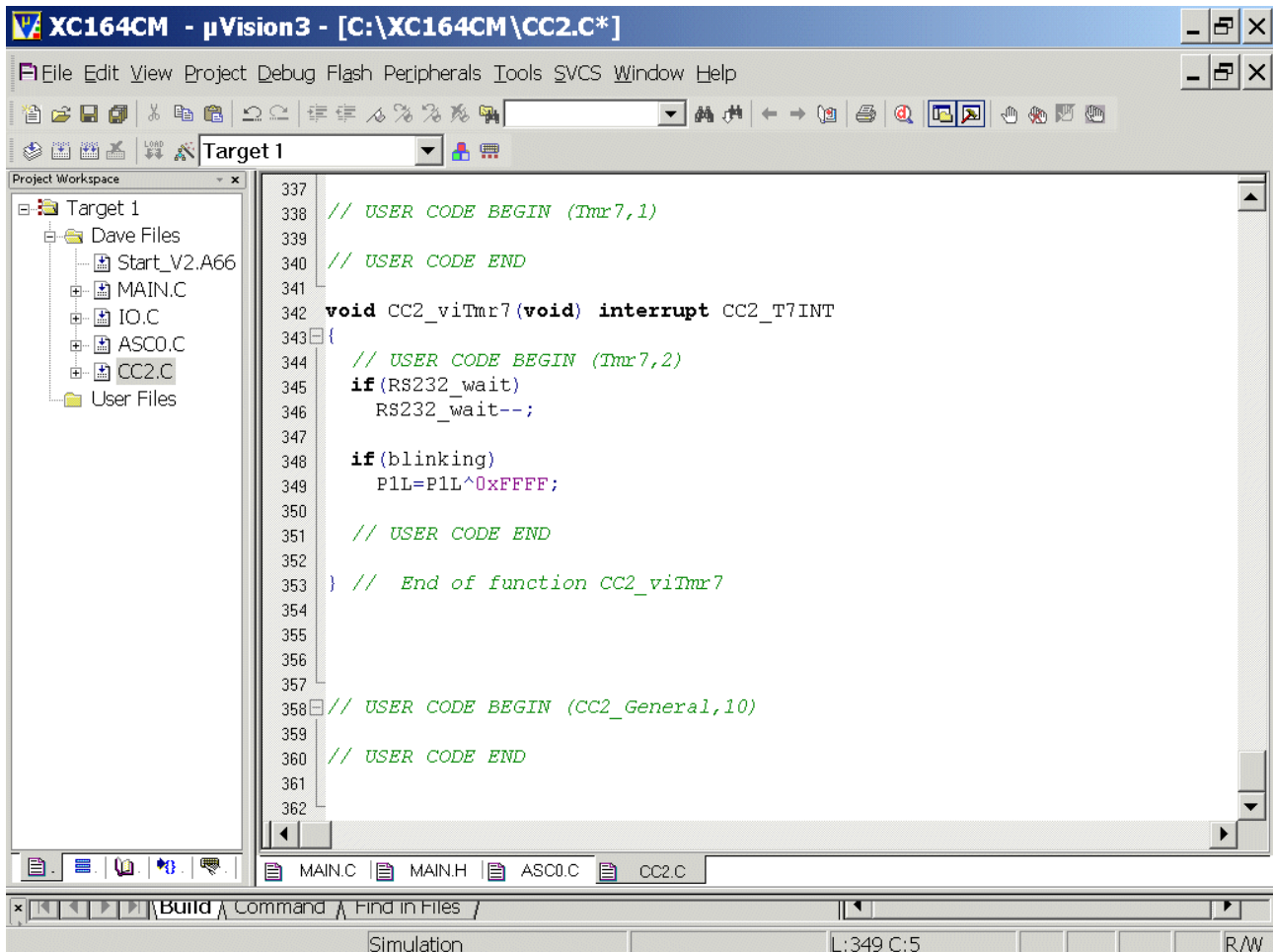
155
156 ASC0_RXFCON = 0x0102; // load ASC0 receive FIFO control register
157 ASC0_TXFCON = 0x0102; // load ASC0 transmit FIFO control register
158
159 /// -----
160 /// Configuration of the used ASC0 Port Pins:
161 /// -----
162 /// - P3.10 is used for ASC Transmit Data Output (TxDA0)
163 /// - P3.11 is used for ASC0 Receive data Input (RxDA0)
164
165 ALTSELOP3 |= 0x0400; // select alternate output function
166 P3 = (P3 & ~(uword)0x0400) | 0x0400; //set data register
167 DP3 = (DP3 & ~(uword)0x0400) | 0x0400; //set direction register
168
169 /// -----
170 /// Configuration of the used ASC0 Interrupts:
171 /// -----
172
173 // USER CODE BEGIN (ASC0_Function,3)
174 ASC0_TIC_IR=1;
175 // USER CODE END
176
177 ASC0_CON |= 0x8000; // enable baud rate generator
178
179
180 } // End of function ASC0_vInit

```

Double click CC2.C and insert code for T7 interrupt service routine:

```
if(RS232_wait)
    RS232_wait--;

if(blinking)
    P1L=P1L^0xFFFF;
```

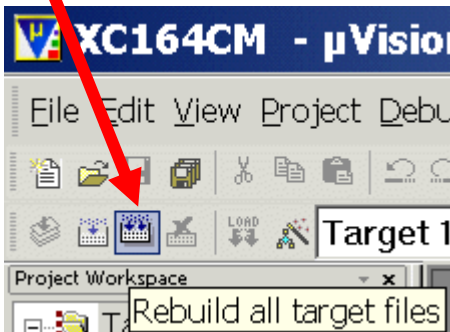


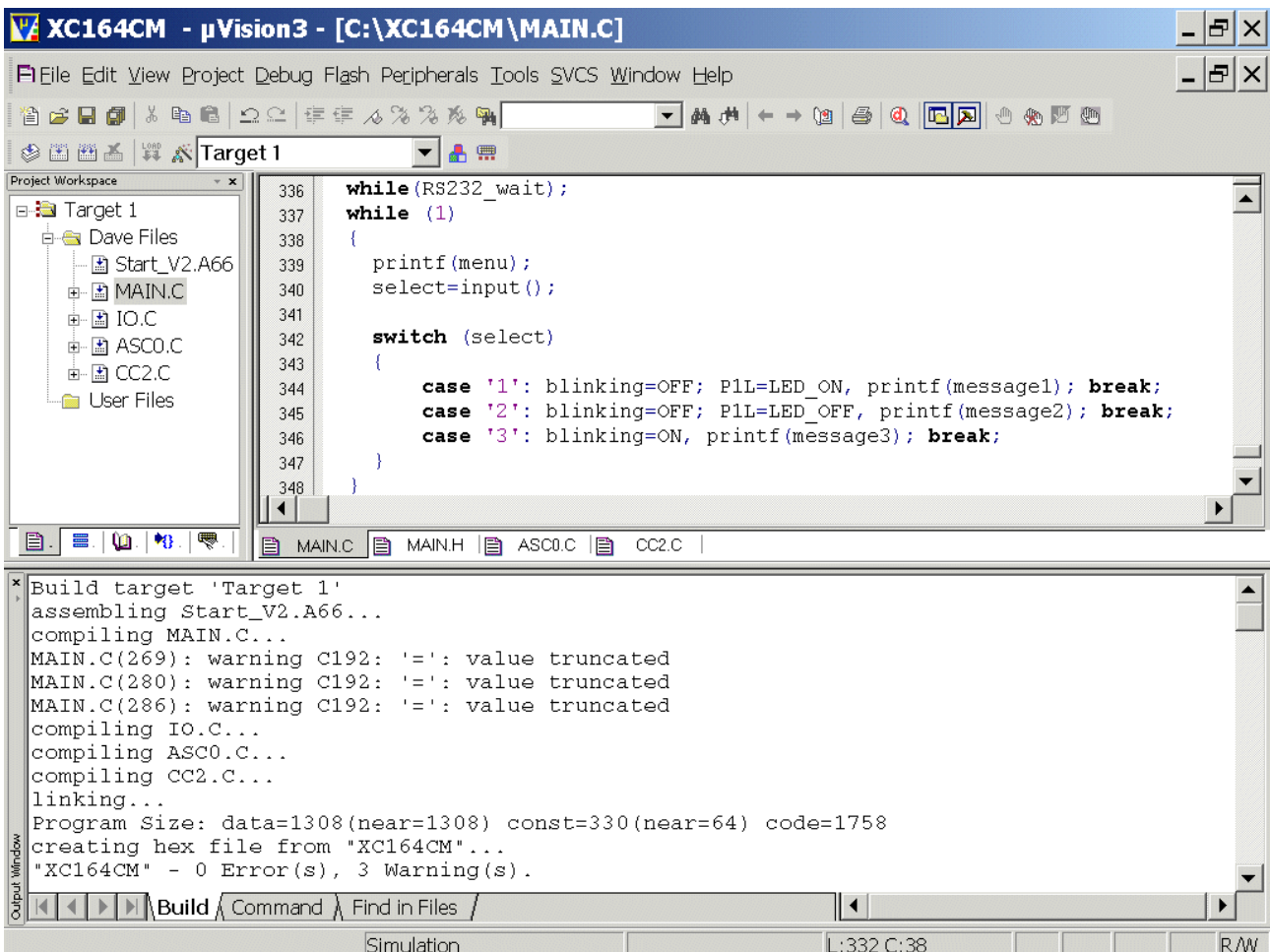
The screenshot shows the µVision3 IDE interface. The title bar reads "XC164CM - µVision3 - [C:\XC164CM\CC2.C*]". The menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and development. The Project Workspace on the left shows a tree view with "Target 1" expanded, containing "Dave Files" (Start_V2.A66, MAIN.C, IO.C, ASC0.C, CC2.C) and "User Files". The main editor window displays the following code:

```
337 // USER CODE BEGIN (Tmr7,1)
338
339 // USER CODE END
340
341 void CC2_viTmr7(void) interrupt CC2_T7INT
342 {
343     // USER CODE BEGIN (Tmr7,2)
344     if(RS232_wait)
345         RS232_wait--;
346
347     if(blinking)
348         P1L=P1L^0xFFFF;
349
350     // USER CODE END
351 } // End of function CC2_viTmr7
352
353 // USER CODE BEGIN (CC2_General,10)
354
355 // USER CODE END
356
357
358
359
360
361
362
```

The status bar at the bottom shows "Simulation" and "L:349 C:5".

Generate your application program:

<p>Project – Rebuild all target files</p>	<p>or click</p> 
---	---



The screenshot shows the IDE interface for XC164CM - μVision3. The main window displays the source code for MAIN.C:

```

336 while (RS232_wait);
337 while (1)
338 {
339     printf(menu);
340     select=input();
341
342     switch (select)
343     {
344         case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
345         case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
346         case '3': blinking=ON, printf(message3); break;
347     }
348 }
    
```

The Output Window at the bottom shows the build process:

```

Build target 'Target 1'
assembling Start_V2.A66...
compiling MAIN.C...
MAIN.C(269): warning C192: '=': value truncated
MAIN.C(280): warning C192: '=': value truncated
MAIN.C(286): warning C192: '=': value truncated
compiling IO.C...
compiling ASC0.C...
compiling CC2.C...
linking...
Program Size: data=1308(near=1308) const=330(near=64) code=1758
creating hex file from "XC164CM"...
"XC164CM" - 0 Error(s), 3 Warning(s).
    
```

Now you can close your project and μVision 3:

Project
Close Project

File - Exit

Note:

From now on you can simply open your μ Vision-Project with:

Start Keil μ Vision and open the Keil Project

If you see an open project – close it: **Project - Close Project**

Project - Open Project

choose: C:\XC164CM

choose: File type: **Project Files (*.uv2)**

choose: XC164CM.Uv2

Open

Note:

You could program now the hex output file **XC164CM.H86** with

Memtool (<http://www.pls-mc.com/>) or

Minimon (www.perschl.at) or

XCFLASHER (Open-Source by www.perschl.at) or

FLASHIT (<http://www.hse-electronic.com>) or

Flash-on-the-Fly

into the OnChipFlash (Program Memory) of the XC164CM microcontroller.

Additionally, you can use the μ Vision3 for OnChipFlash-Programming (using the JTAG interface) - of course - you can also use a Debugger for this task.

Using the simulator (first we will test our program inside the Keil Simulator without hardware):

The uVision3 simulator has a couple of VTREG's that allow you the configuration of the reset behaviour. These pins of the Infineon XC16x are reflected by the following VTREGs:

EA is the EA pin (0:= code fetch from external ROM ; 1:= code fetch from on-chip ROM)
 ALE is the ALE pin
 RD is the RD pin
 WR is the WR pin
 RSTCFGP are the levels on the configuration PINs
 NMIPIN is the NMI pin

By default, the value of the EA pin is set to 0 to simulate the off-chip ROM. When this VTREG is 1 at CPU reset, the uVision3 simulates on-chip ROM and the RESET vector and interrupt vectors are at address 0xC00000.

Additionally, see the XC16x Derivatives System Units User's Manual:

Basic Startup Configuration via External Circuitry				
Latched config.	EA = '0'		EA = '1'	
	RD = '0'	RD = '1'	RD = '0'	RD = '1'
ALE = '0'	Standard start external, PLL/OVD off ¹⁾	Standard start external, PLL/OVD ON	Standard Boot	Standard start internal
ALE = '1'	Reserved	Reserved	Alternate Boot	Alternate start internal
WR = '0'	RORMV = '0'		RORMV = '1' ²⁾	
WR = '1'	RORMV = '0'		RORMV = '0'	

¹⁾ Only effective in bypass mode.
²⁾ P20.12 enabled, this indicates a single-chip reset without external bus system.

Note: The pullups/pulldowns on the configuration pins are activated at the beginning of a hardware reset.

It is badly needed to start the µVision simulator with program start in on-chip ROM (Flash)!

Start the μ Vision simulator with program start in on-chip ROM (Flash):

You can enable the on-chip-ROM-behavior (Standard Internal Start) as described below:

<p>mouse position: (Project Workspace, Files): Target1 click right mouse button Options for Target 'Target1'</p>	<p>or click</p>

Debug click Settings

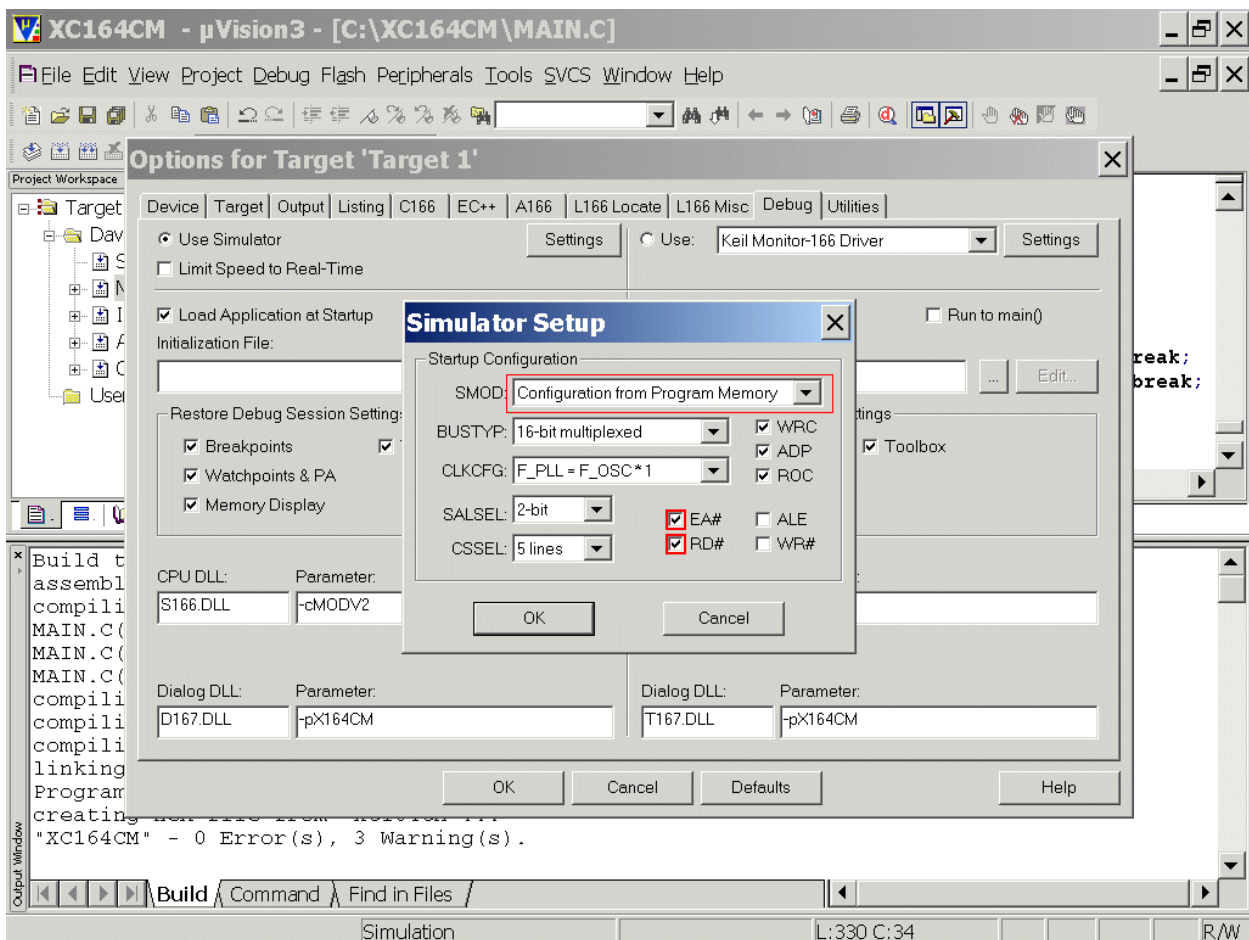
Simulator Setup Startup Configuration **SMOD** choose Configuration from Program Memory

click EA# = 1

click RD# = 1

click ALE# = 0

click WR# = 0



OK

OK

<p>Debug – Start/Stop Debug Session</p>	<p>or click</p>
---	------------------------



OK

XC164CM - μVision3 - [C:\XC164CM\MAIN.C]

File Edit View Project Debug Flash Peripherals Tools SVCS Window Help

Project Workspace

Register	Value
Word	
r0	0x8200
r1	0x00c4
r2	0x0000
r3	0xd000
r4	0x0001
r5	0x0000
r6	0x0000
r7	0x0000
r8	0x01f6
r9	0x00c0
r10	0x0000
r11	0x0000
r12	0x0000
r13	0x0000
r14	0x0000
r15	0x0000
Byte	
Mac	
mac	0x0
meh	0x0000
mal	0x0000

```

327 void main(void)
328 {
329     // USER CODE BEGIN (Main,2)
330
331     // USER CODE END
332
333     MAIN_vInit();
334
335     // USER CODE BEGIN (Main,4)
336     while(RS232_wait);
337     while (1)
338     {
339         printf(menu);
340         select=input();
341
342         switch (select)
343         {
344             case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
345             case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
346             case '3': blinking=ON, printf(message3); break;
347         }
348     }
349 }
    
```

Output Window

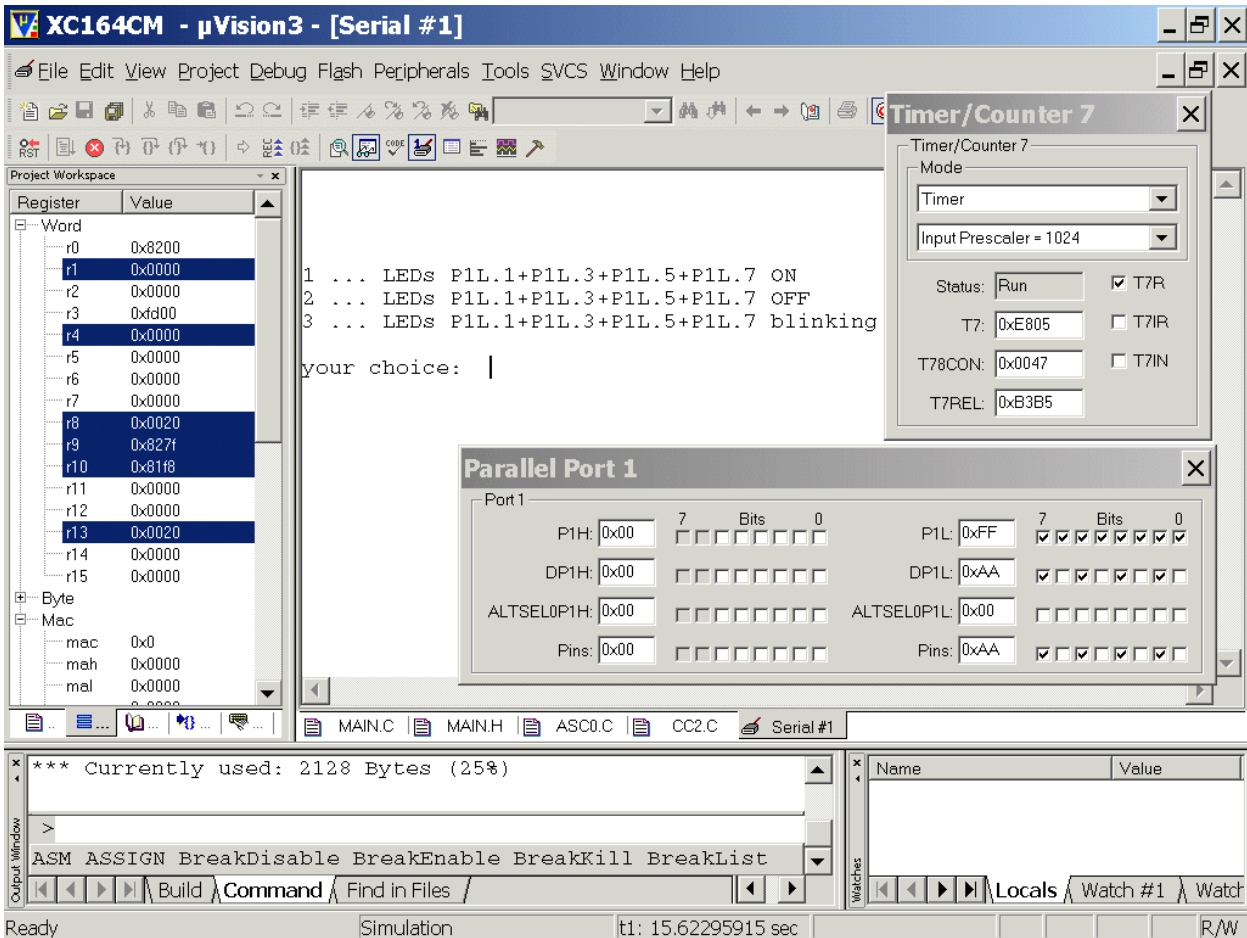
Load "C:\\XC164CM\\XC164CM"

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList

Build Command Find in Files

Ready Simulation t1: 0.00015300 sec L:333 C:1 R/W

- Debug – Run
- View - Serial Window #1
- Peripherals - I/O-Ports – Port1
- Peripherals – Capcom2 (T7,T8) – T7



Note:

By activating (click) the “**Serial Window #1**” you can type 1, 2 or 3 and you can watch the result in the “**Parallel Port 1**” window.

Now (if you want) you can close your simulator session:

- Debug
- Stop Running
- Debug
- Start/Stop Debug Session

Now (if you want) you can close your project and µVision 3:

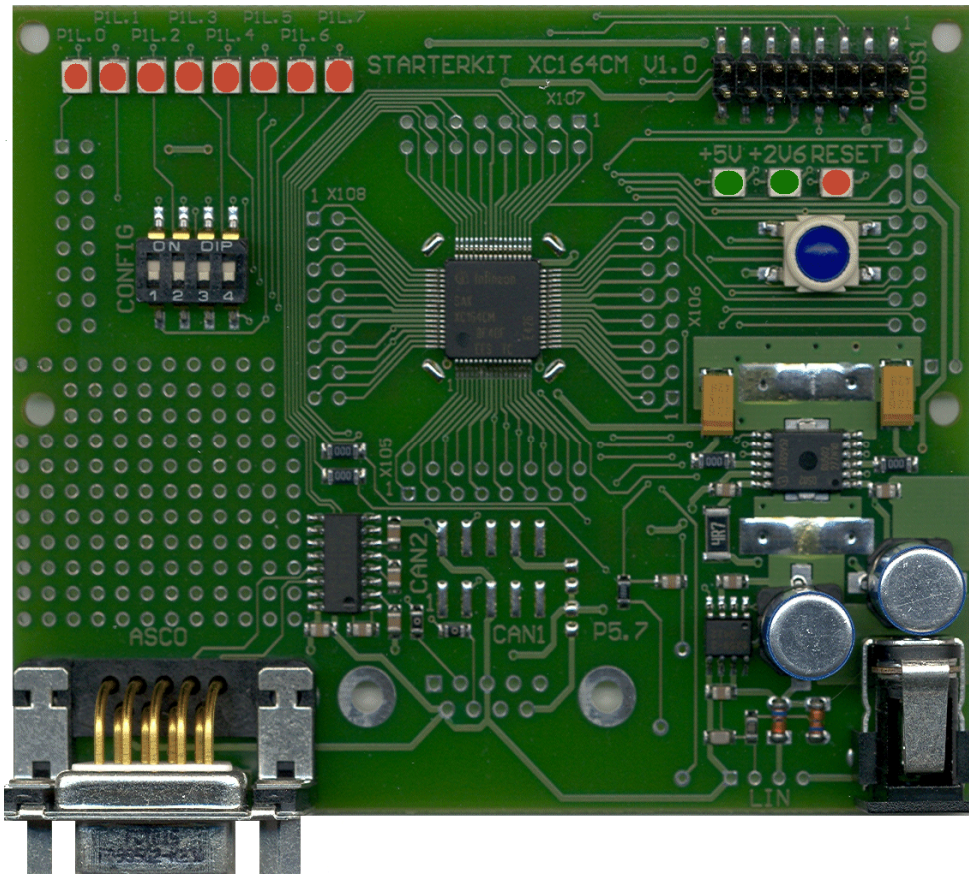
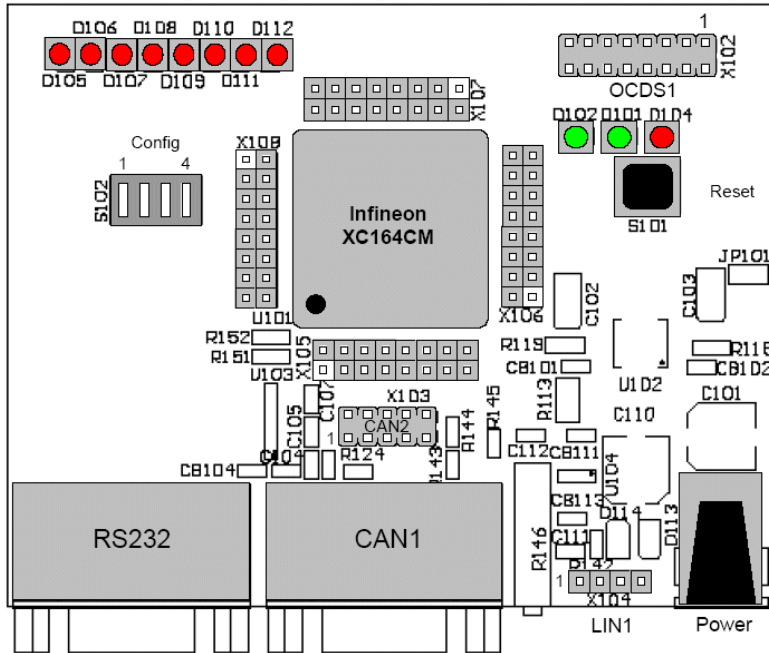
- Project
- Close Project

File - Exit

Using real hardware:

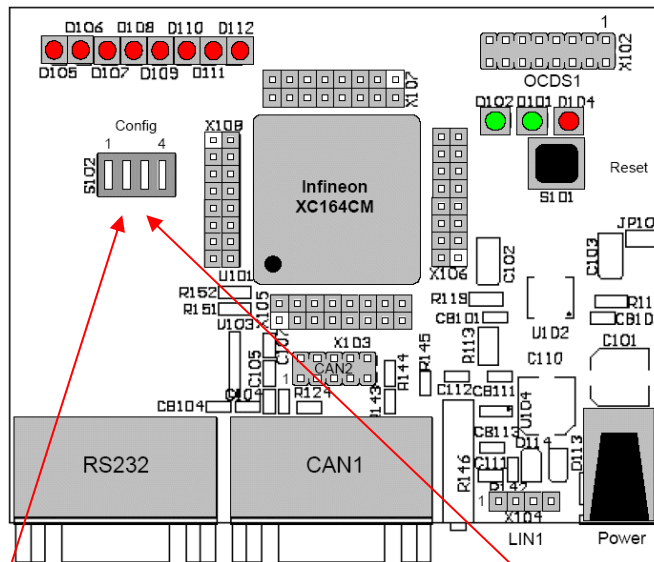
5.) Using Bootstrap loader (ASC0) + MEMTOOL:

Layout Overview:



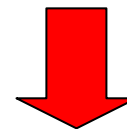
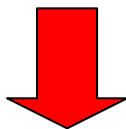
DIP switch 102 Jumper Settings:

Bootstrap Loader ASC	↔	Standard Internal Start
----------------------	---	-------------------------



Bootstrap Loader ASC	↔	Standard Internal Start
----------------------	---	-------------------------

Switch S102:



	↔	
Startup configuration: Bootstrap loader ASC		Startup configuration: Standard start from internal Flash

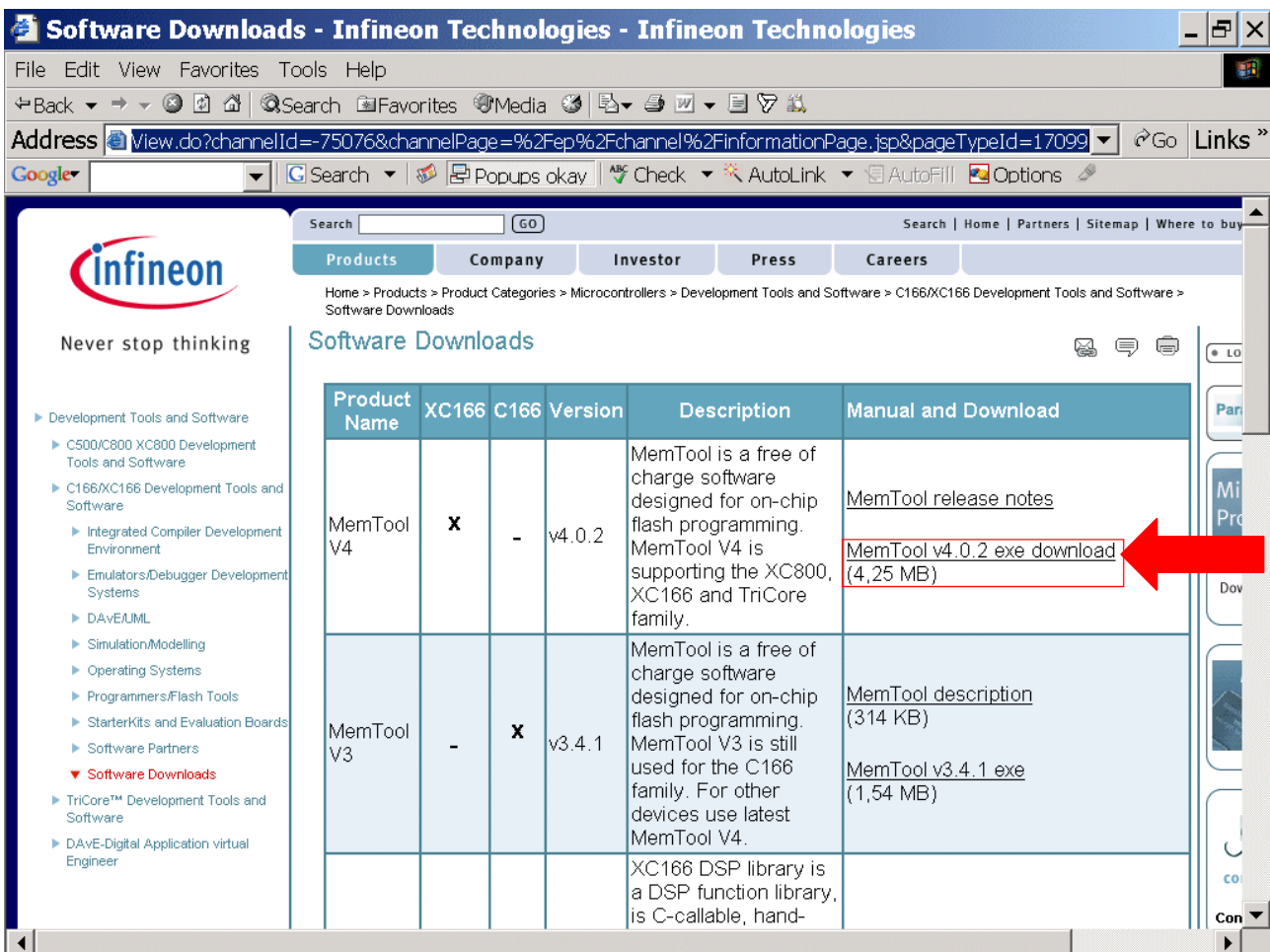
MEMTOOL Installation:



The following procedure will install/prepare Memtool for the programming of the XC164CM:

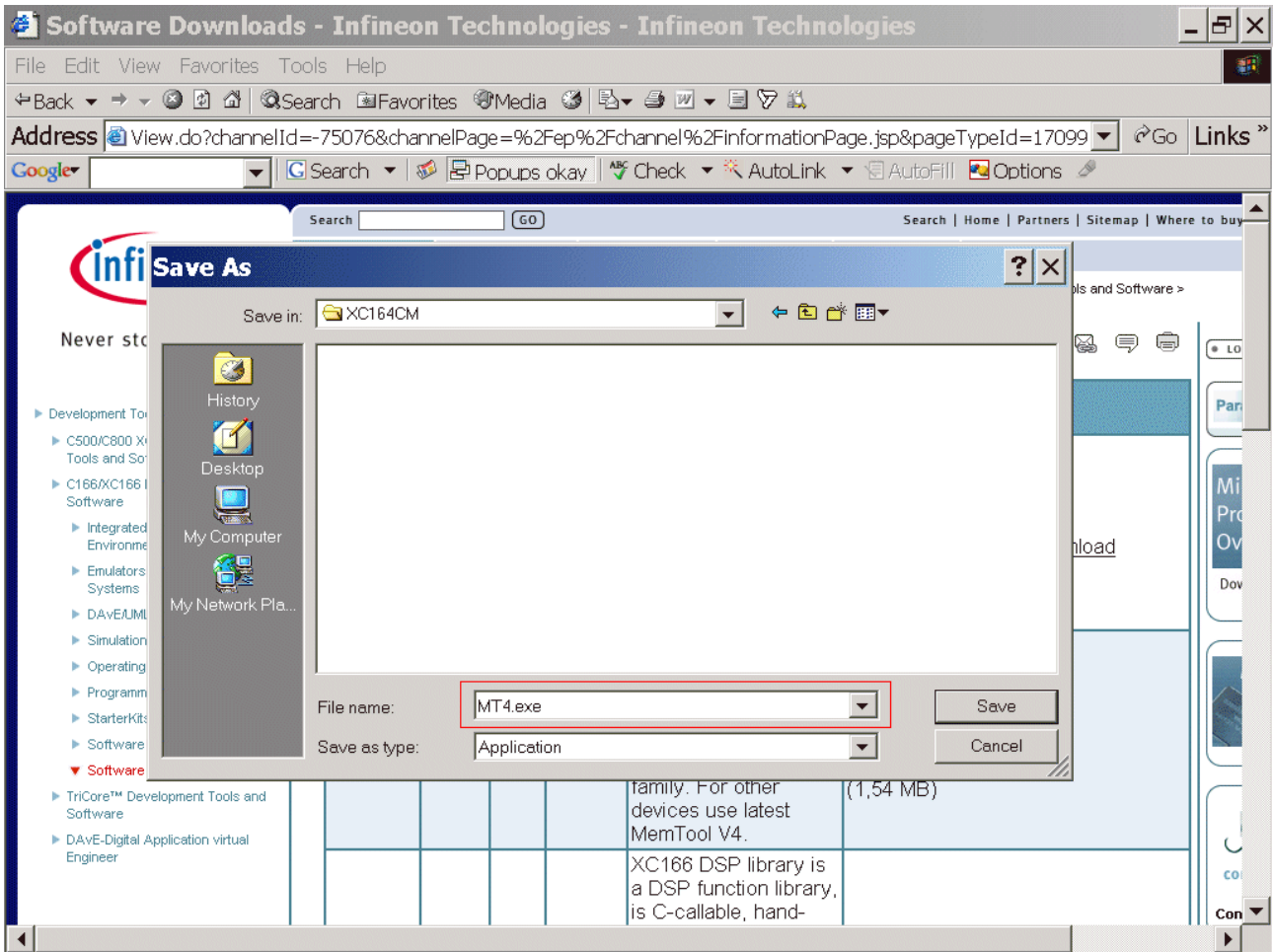
Download **MemToolv4.0.2.exe** (or higher version) @

<http://www.infineon.com/cgi-bin/ifx/portal/ep/channelView.do?channelId=-75076&channelPage=%2Fep%2Fchannel%2FinformationPage.jsp&pageTypeId=17099>

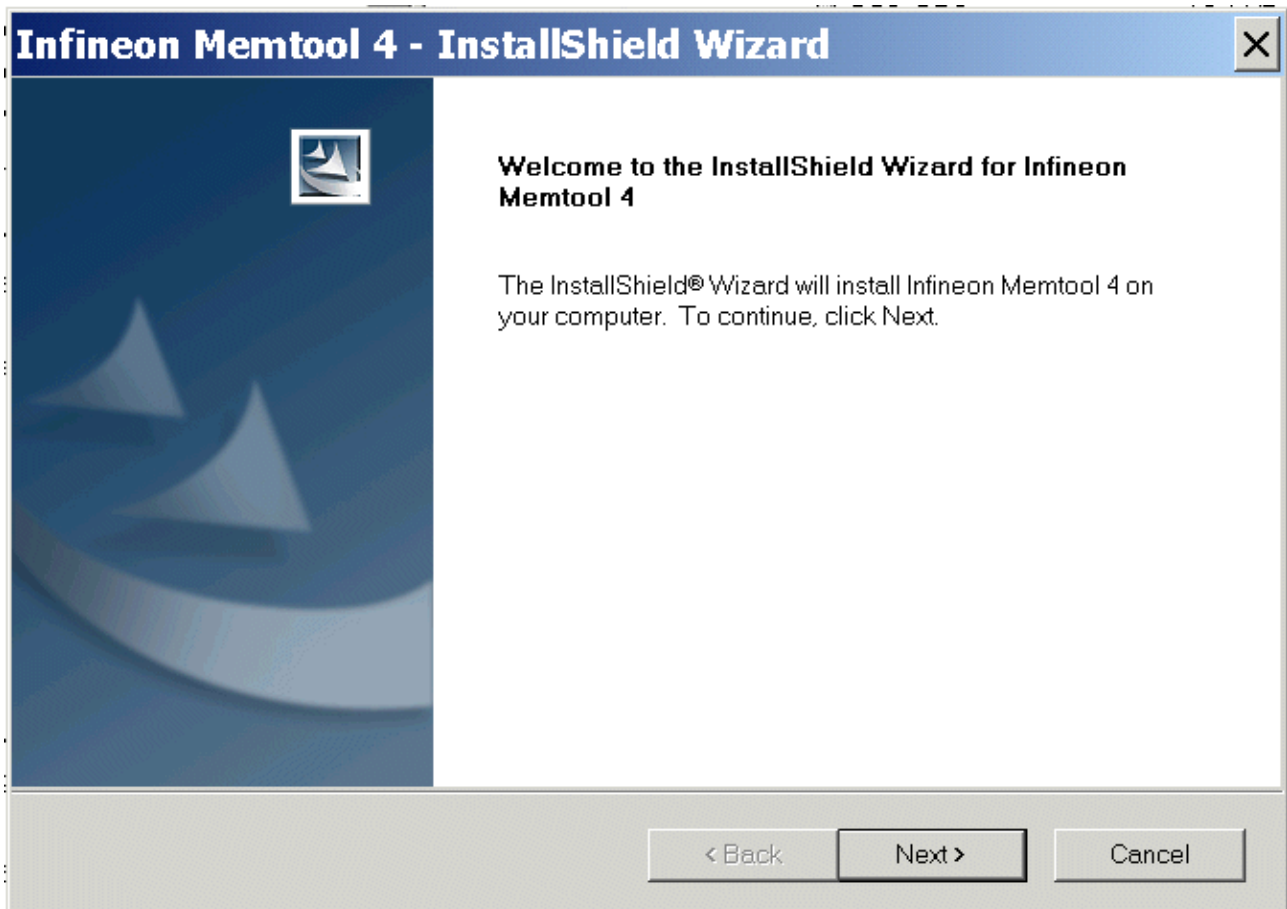


The screenshot shows the Infineon website's 'Software Downloads' section. A table lists MemTool versions for different microcontroller families. The 'MemTool v4.0.2 exe download (4,25 MB)' link is highlighted with a red arrow.

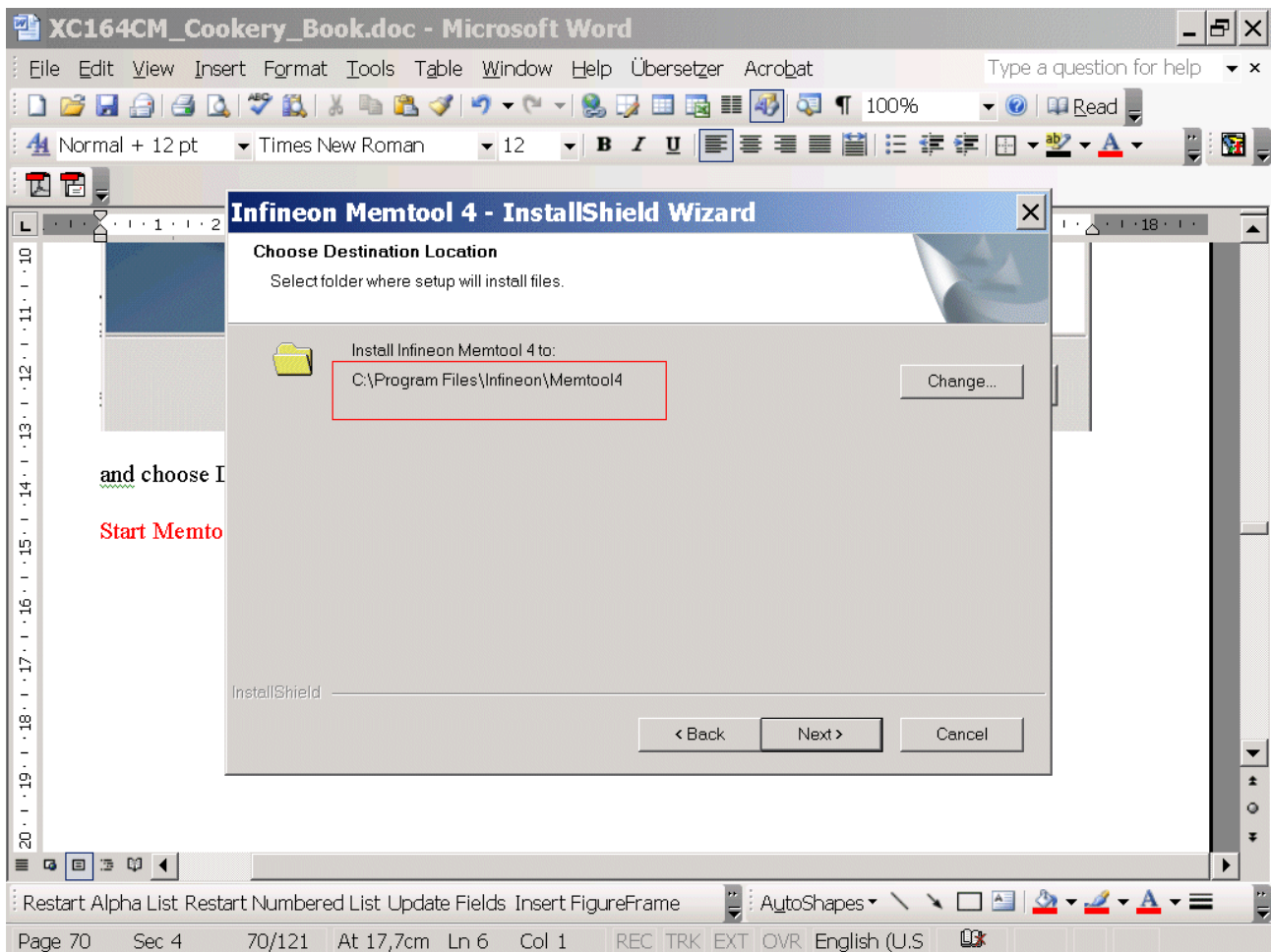
Product Name	XC166	C166	Version	Description	Manual and Download
MemTool V4	X	-	v4.0.2	MemTool is a free of charge software designed for on-chip flash programming. MemTool V4 is supporting the XC800, XC166 and TriCore family.	MemTool release notes MemTool v4.0.2 exe download (4,25 MB)
MemTool V3	-	X	v3.4.1	MemTool is a free of charge software designed for on-chip flash programming. MemTool V3 is still used for the C166 family. For other devices use latest MemTool V4.	MemTool description (314 KB) MemTool v3.4.1 exe (1,54 MB)
				XC166 DSP library is a DSP function library, is C-callable, hand-	

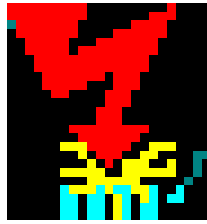


Execute file **MT4.exe** (- or higher version)

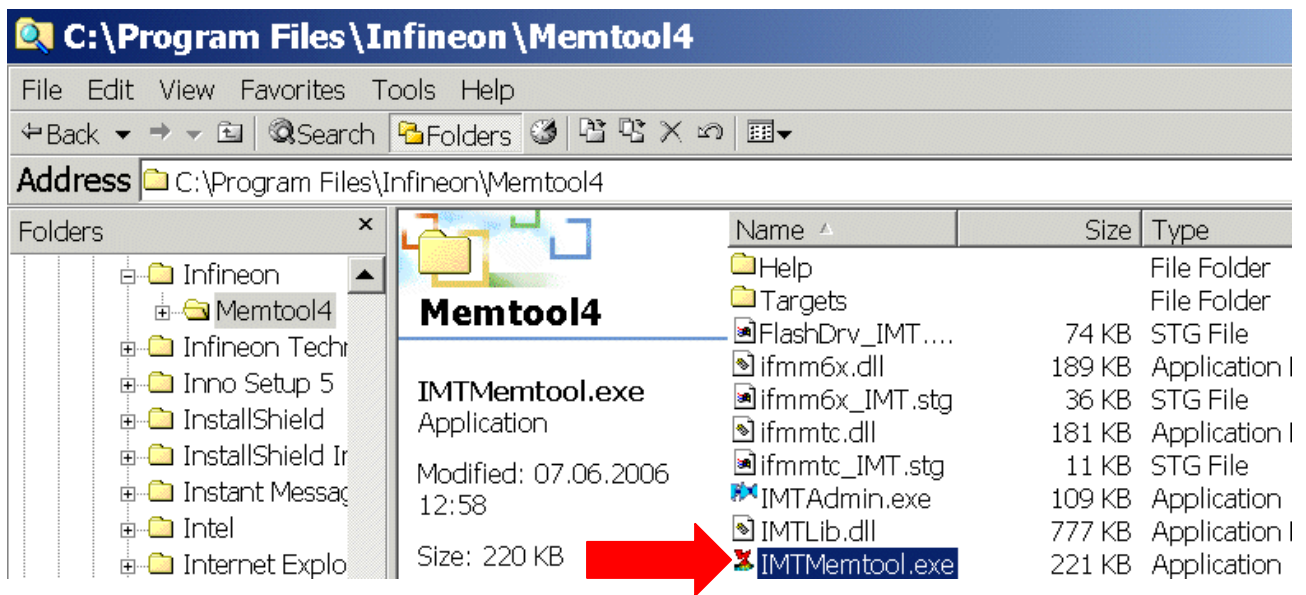


And choose Destination Location (default directory) `C:\Program Files\Infineon\Memtool4`:

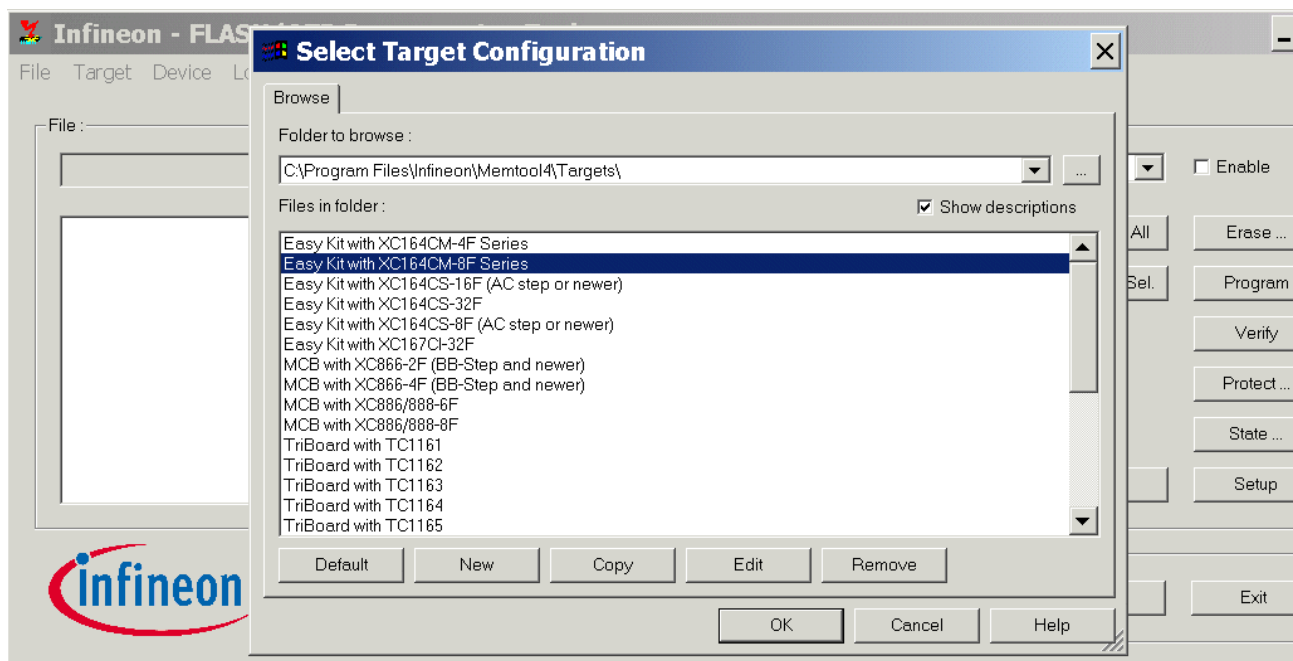




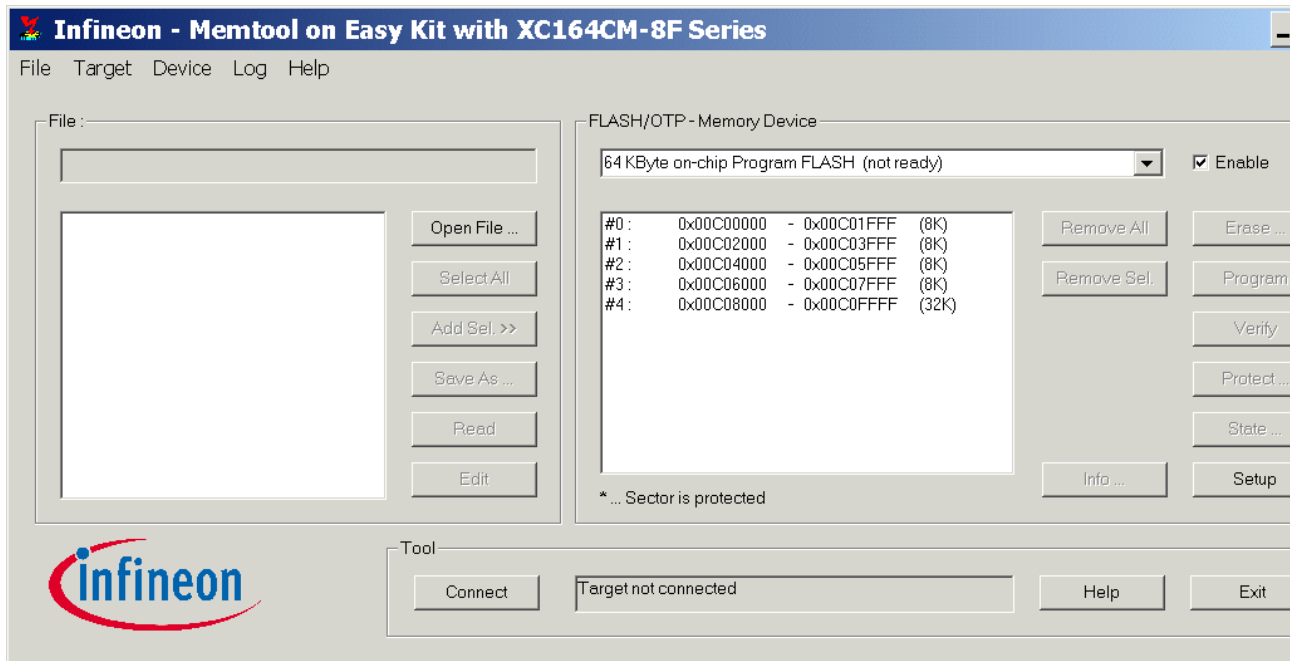
Start Memtool:



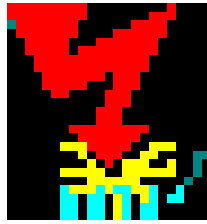
Browse: Select Easy Kit with XC164CM-8F Series:



OK



[MEMTOOL is now ready to program the OnChipFlash of the XC164CM microcontroller.](#)

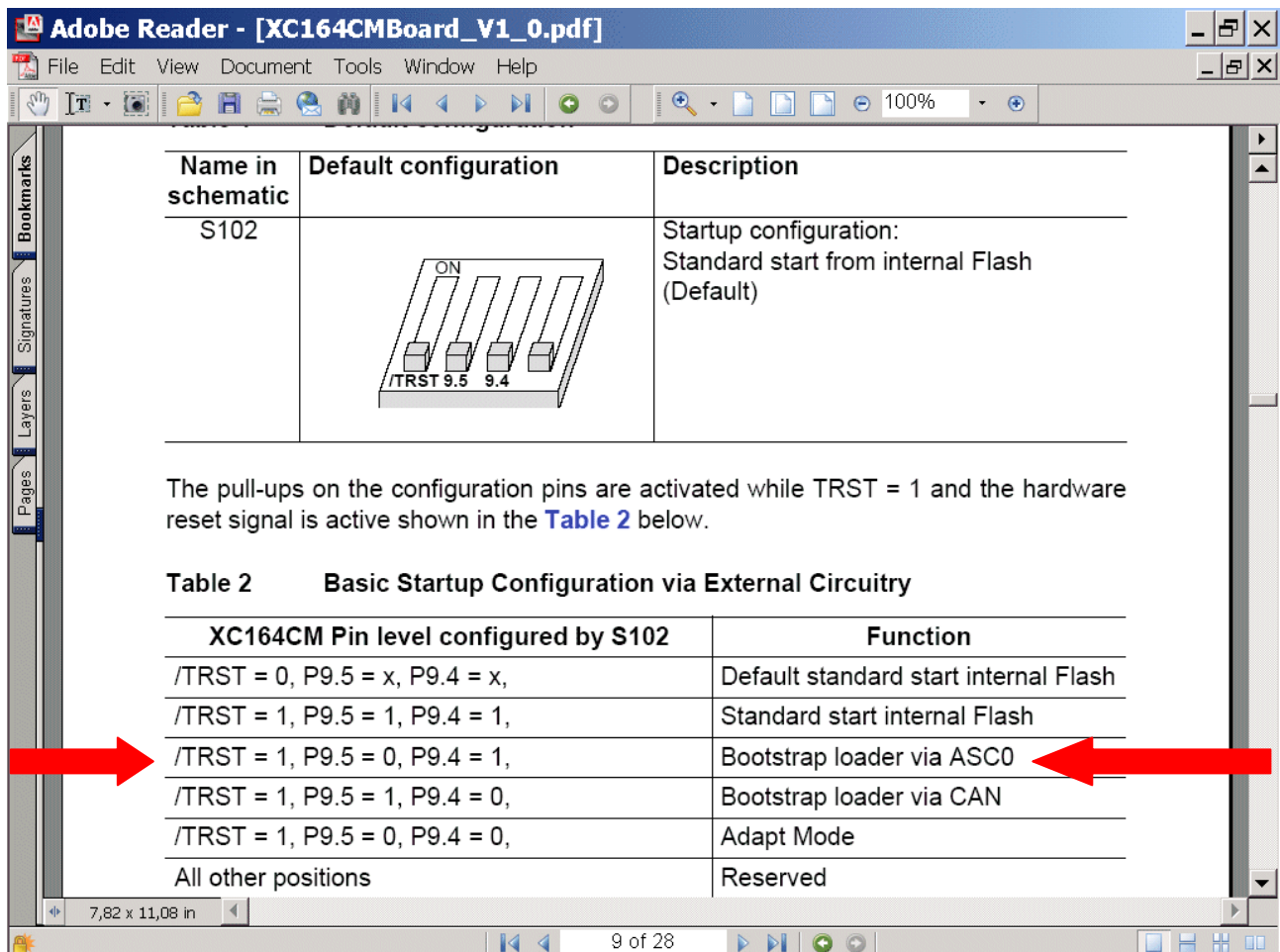


Using MEMTOOL:

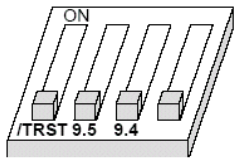
Programming **XC164CM.h86** into the OnChipFlash in the XC164CM:

Assign Power to your target board.

Set Dip Switch S102 to Bootstrap Loader ASC:



The screenshot shows a PDF document titled "XC164CMBoard_V1_0.pdf" in Adobe Reader. The document contains a table with columns for "Name in schematic", "Default configuration", and "Description". The entry for "S102" includes a diagram of a 4-pin dip switch labeled "ON" with pins "TRST 9.5" and "9.4". Below the table, there is a paragraph explaining that pull-ups on configuration pins are activated while TRST = 1 and the hardware reset signal is active. This is followed by "Table 2 Basic Startup Configuration via External Circuitry", which is a table with two columns: "XC164CM Pin level configured by S102" and "Function". The fourth row of this table, corresponding to "/TRST = 1, P9.5 = 0, P9.4 = 1", is highlighted with red arrows pointing to it from both the left and right sides of the page.

Name in schematic	Default configuration	Description
S102		Startup configuration: Standard start from internal Flash (Default)

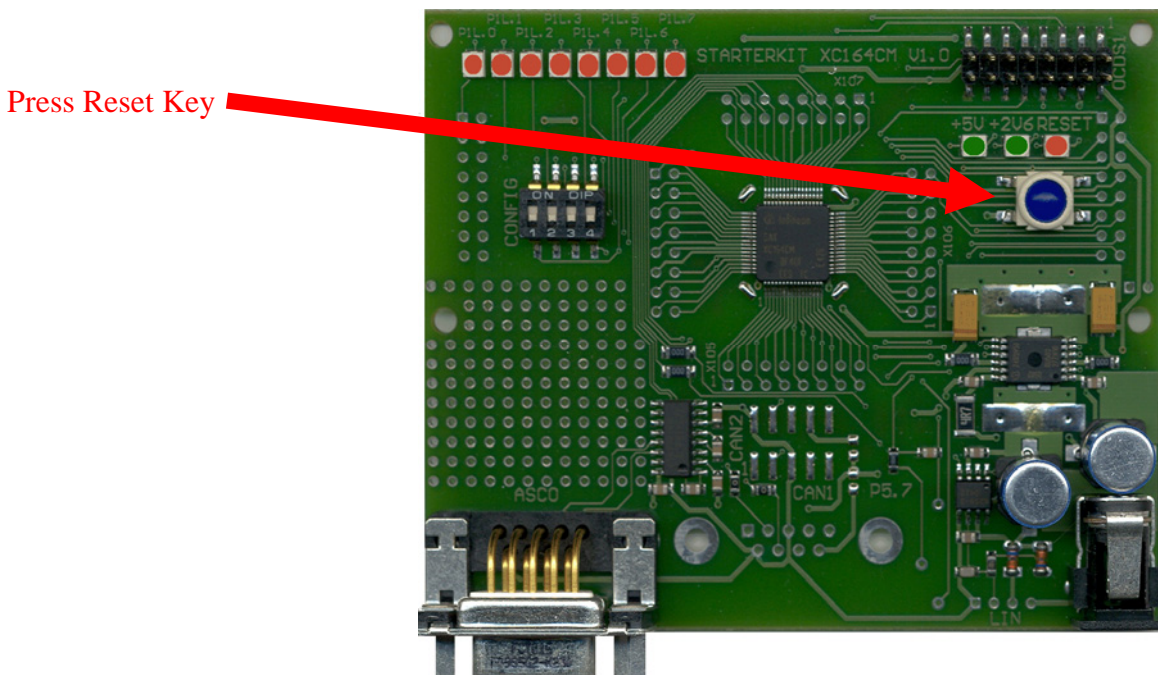
The pull-ups on the configuration pins are activated while TRST = 1 and the hardware reset signal is active shown in the [Table 2](#) below.

Table 2 Basic Startup Configuration via External Circuitry

XC164CM Pin level configured by S102	Function
/TRST = 0, P9.5 = x, P9.4 = x,	Default standard start internal Flash
/TRST = 1, P9.5 = 1, P9.4 = 1,	Standard start internal Flash
/TRST = 1, P9.5 = 0, P9.4 = 1,	Bootstrap loader via ASC0
/TRST = 1, P9.5 = 1, P9.4 = 0,	Bootstrap loader via CAN
/TRST = 1, P9.5 = 0, P9.4 = 0,	Adapt Mode
All other positions	Reserved

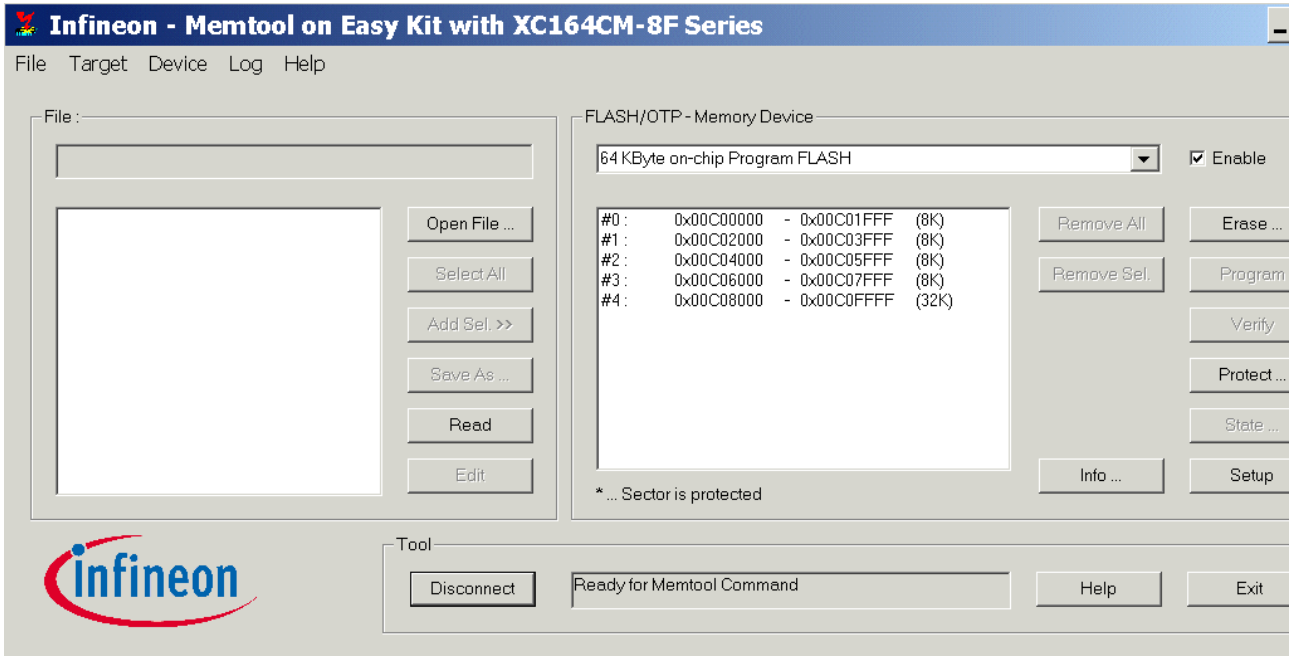
Adobe Reader - [XC164CMBoard_V1_0.pdf]

Name in schematic	Default configuration	Description
S102		Startup configuration: Standard start from internal Flash OFF-OFF-OFF-OFF
S102		Startup configuration: Bootstrap loader ASC ON-ON-OFF-OFF
S102		Startup configuration: Bootstrap loader CAN ON-OFF-ON-OFF
S102		Startup configuration Adapt Mode ON-ON-ON-OFF
S102		Startup configuration All other positions are reserved

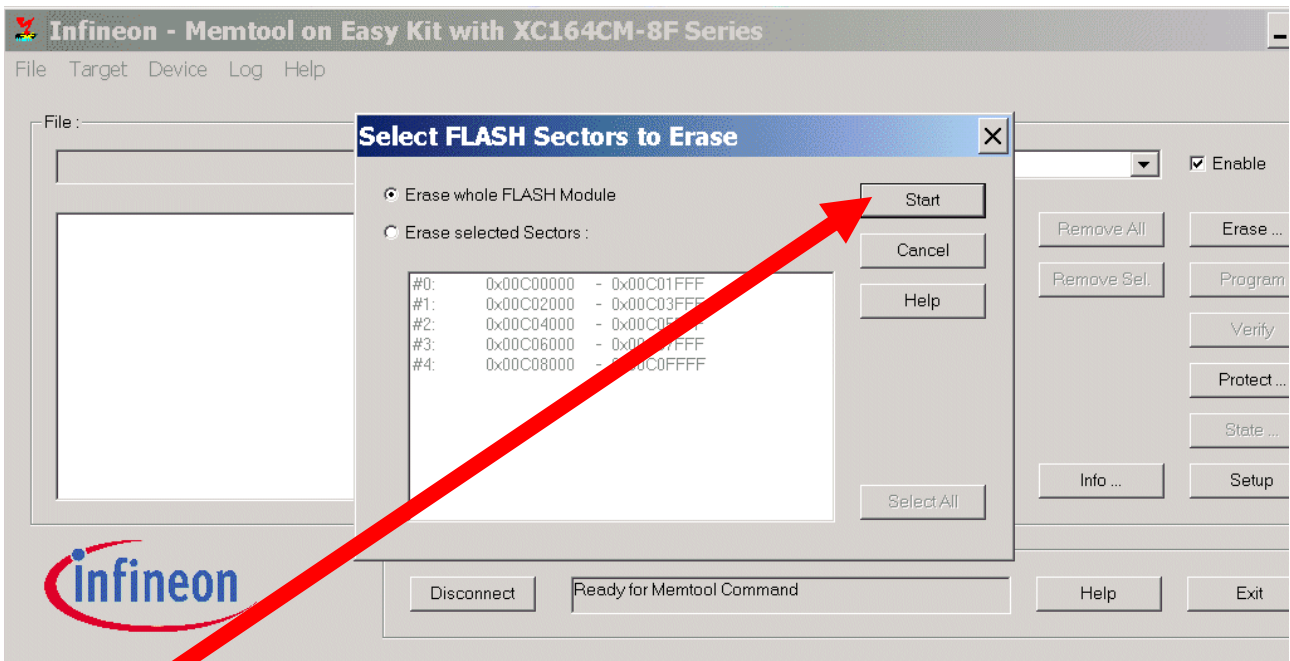


Erase the ONChipFlash:

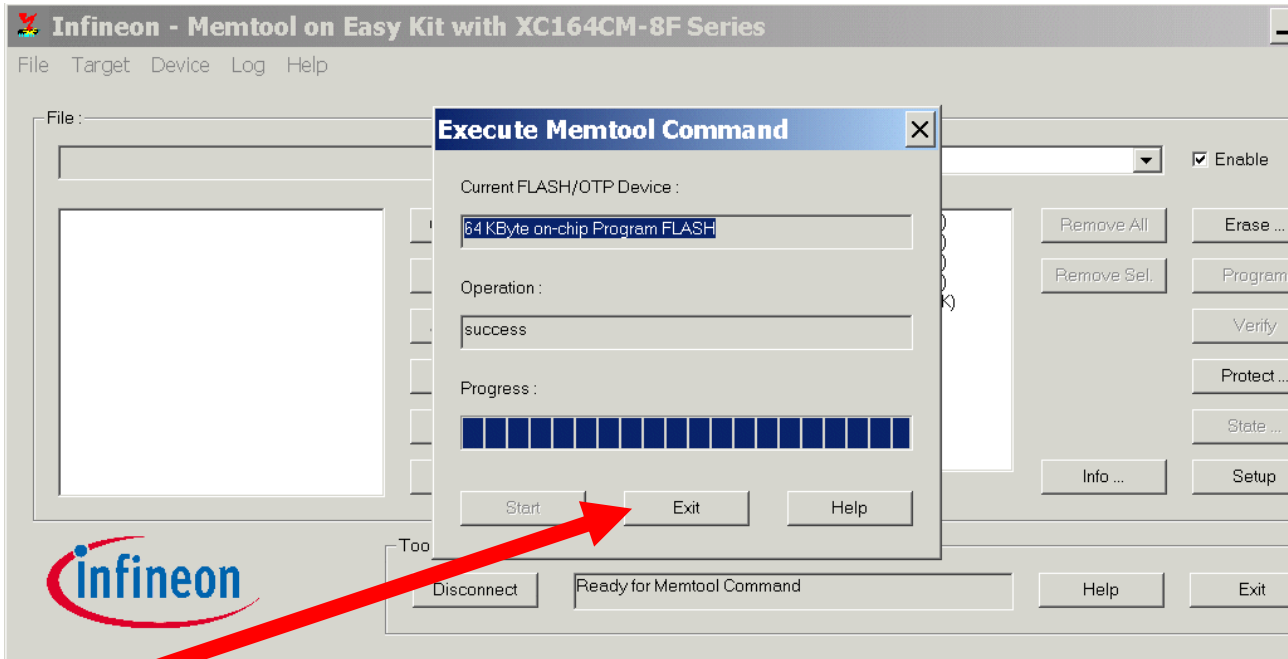
Connect



Erase



Start



Exit

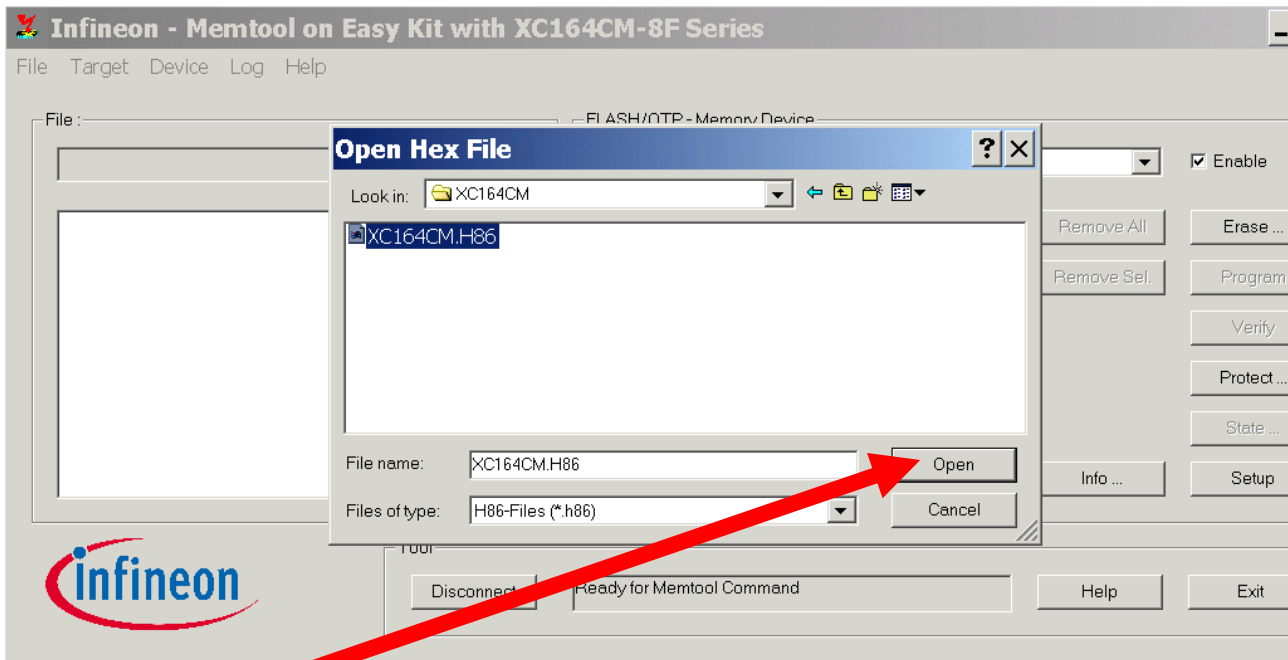
Load HEX File:

Open File

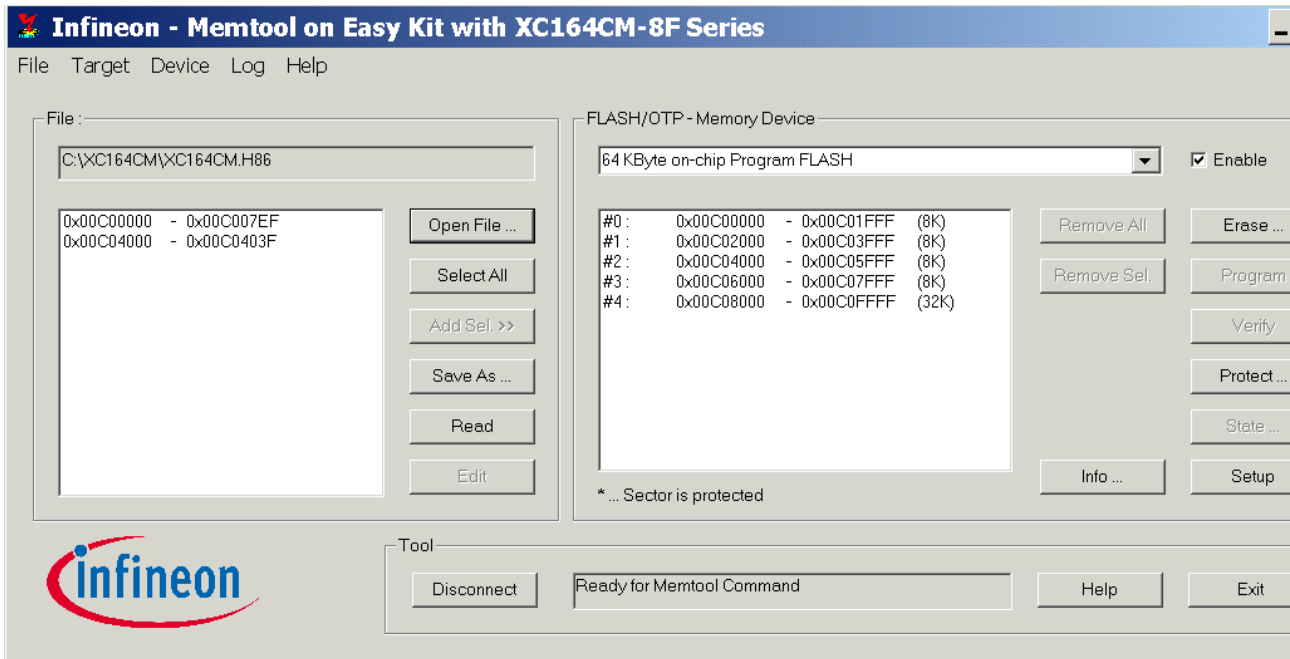
Dateityp (Files of type): select H86-Files (*.h86)

Look in: C:\XC164CM

Select XC164CM.H86

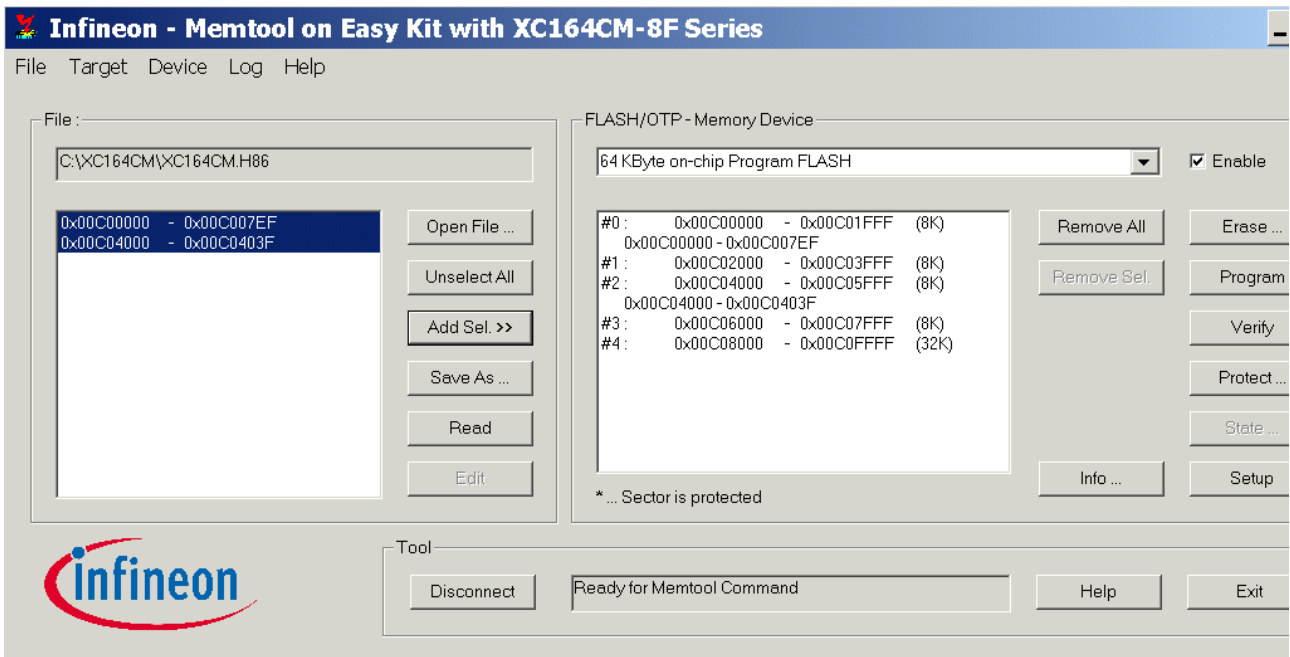


Open

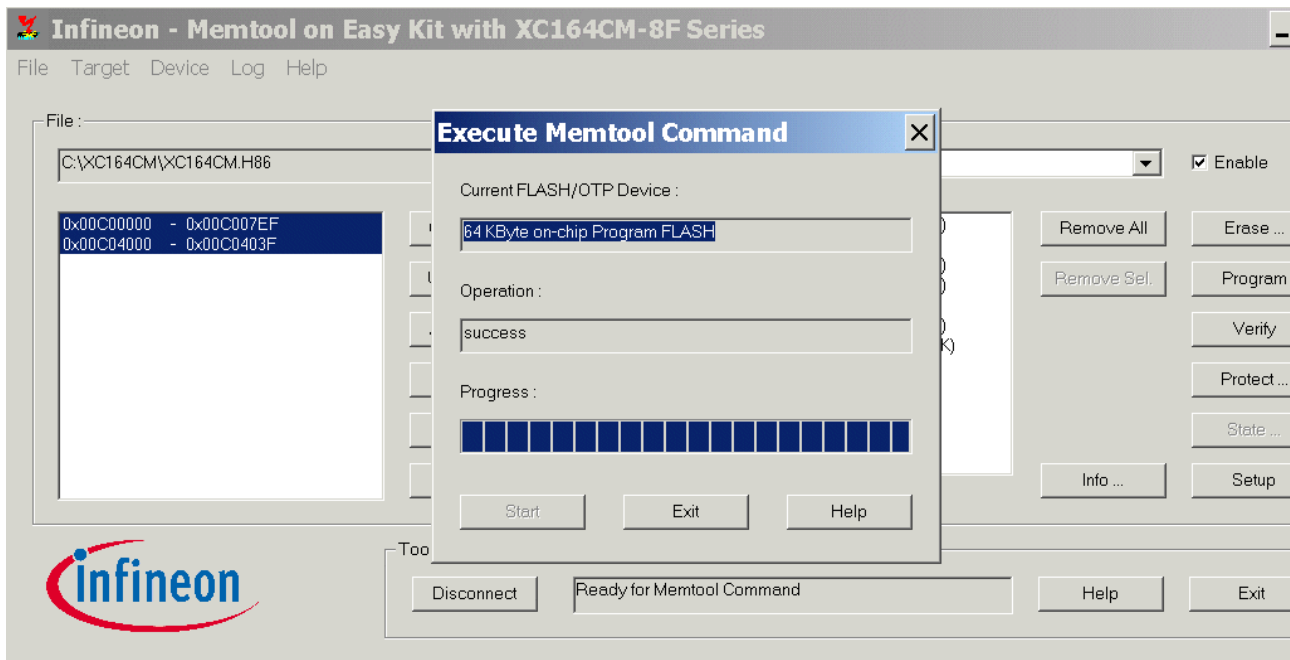


Program selected hex-file into the OnChipFlash:

Select All
Add Sel >>



Program



Exit

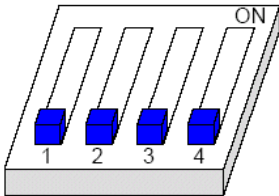
Exit MEMTOOL:

Disconnect

Exit

Start your program:

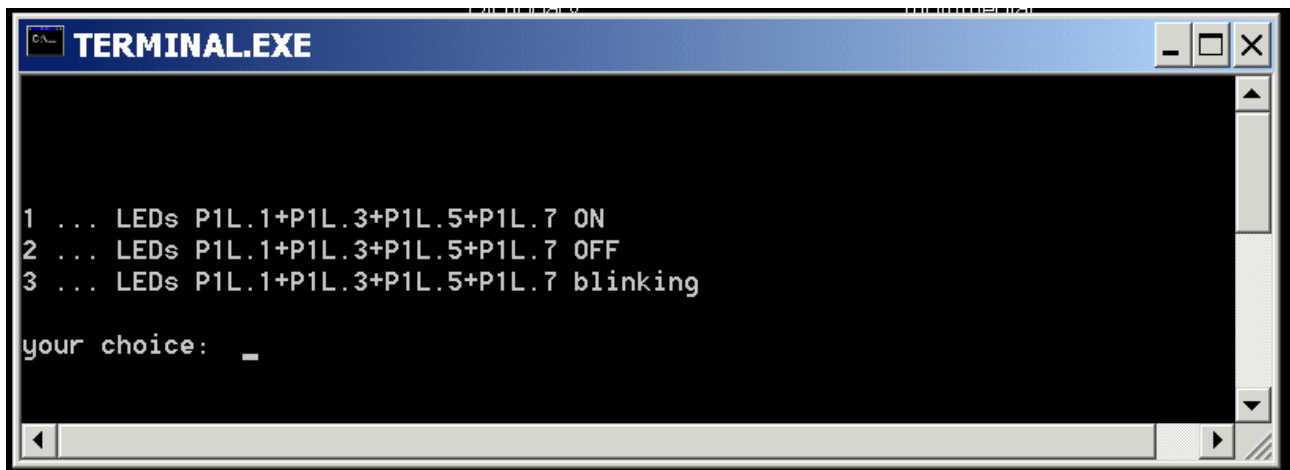
Set Dip Switch 102 to Standard internal start:



Startup configuration:
Standard start from internal Flash
OFF-OFF-OFF-OFF

Press Reset Key

Your application is running!



```
TERMINAL.EXE
1 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 ON
2 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 OFF
3 ... LEDs P1L.1+P1L.3+P1L.5+P1L.7 blinking
your choice: _
```

6.) OnChipFlash-Programming, OCDS-Debugging

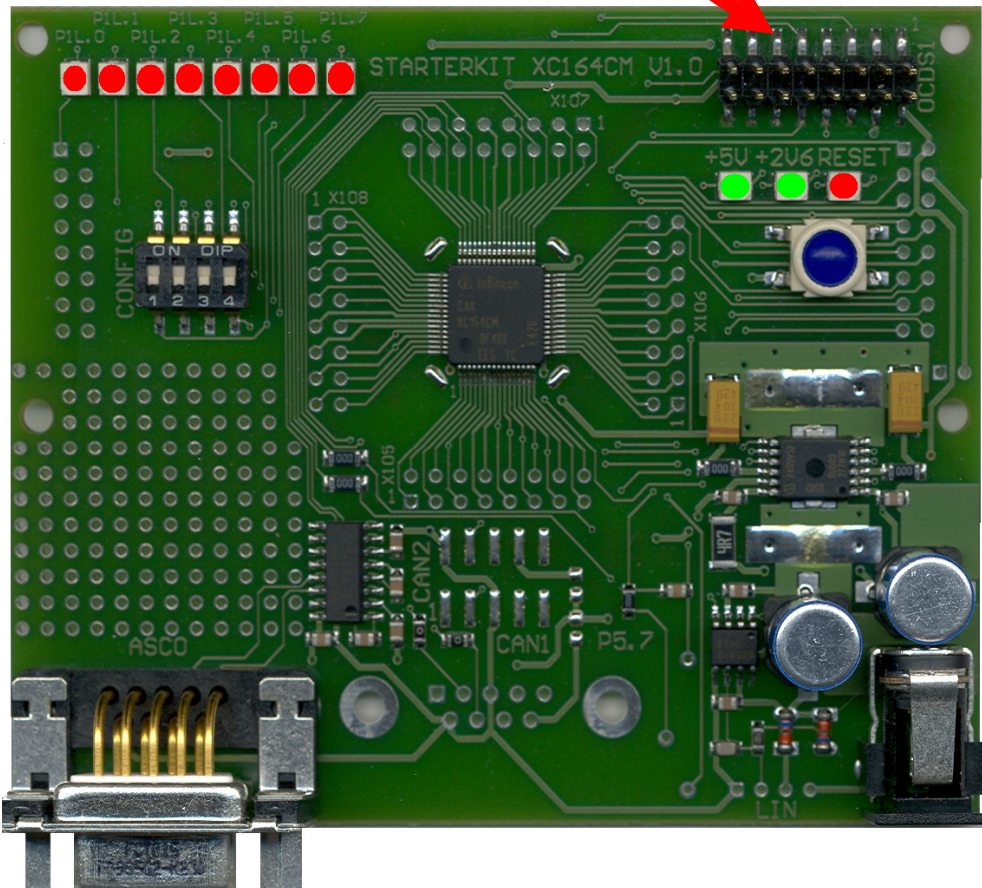
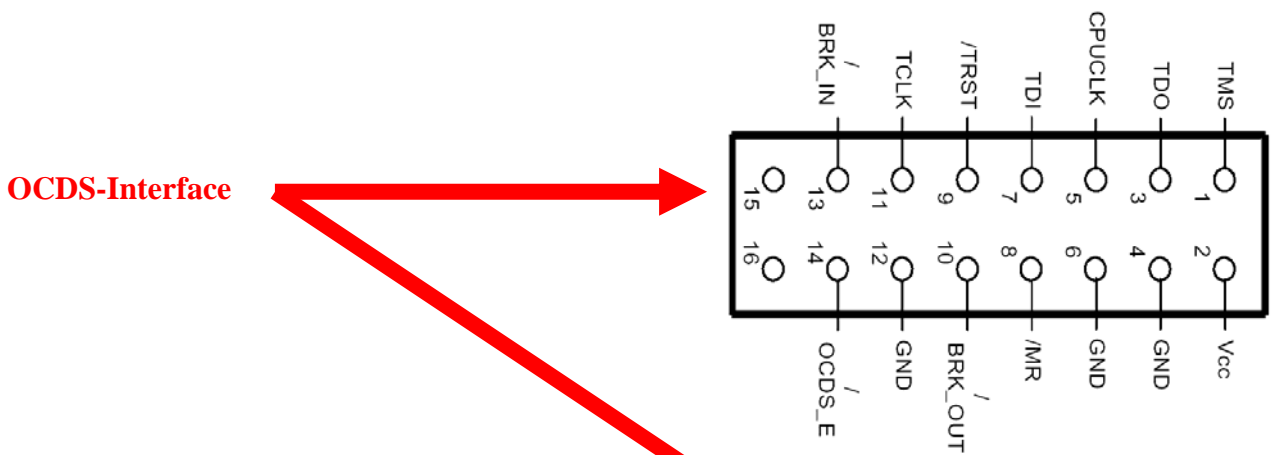
via [Keil-ULINK-JTAG-Interface](#) or [USB-JTAG-Wiggler-Box](#)

Board Configuration

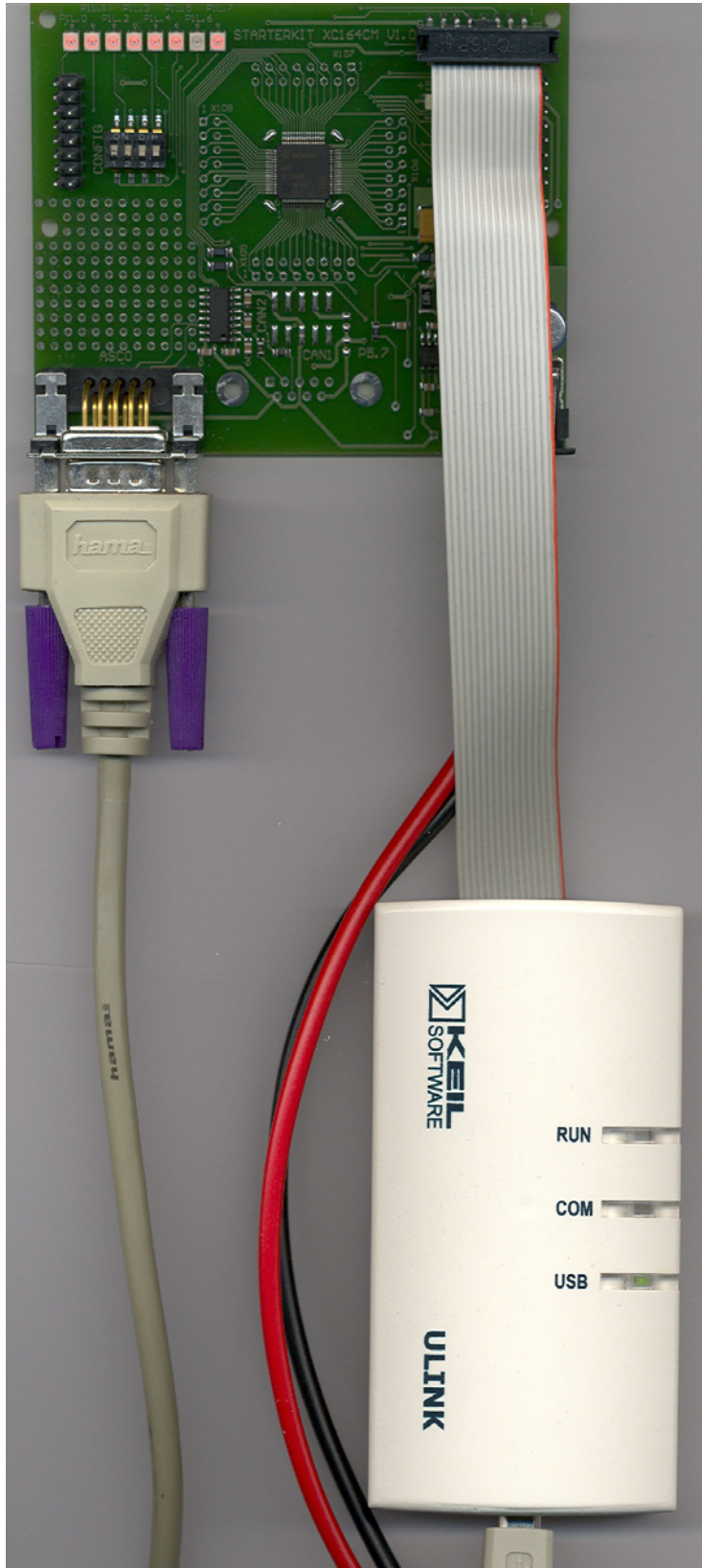
[For further information, please refer to the XC164CM Board Manual, V0.3, April 2004 .](#)

[For further information, please refer to the XC164CM Board Manual, V0.9, Aug. 2005 .](#)

[For further information, please refer to the XC164CM Board Manual, V1.0, Jan. 2006 .](#)



Keil-ULINK-JTAG-Interface



OnChipFlash-Programming via KEIL-ULINK

Start Keil μ Vision 3 and open the KEIL Project **XC164CM.Uv2**

If you see an open project – close it: **Project - Close Project**

Project - Open Project

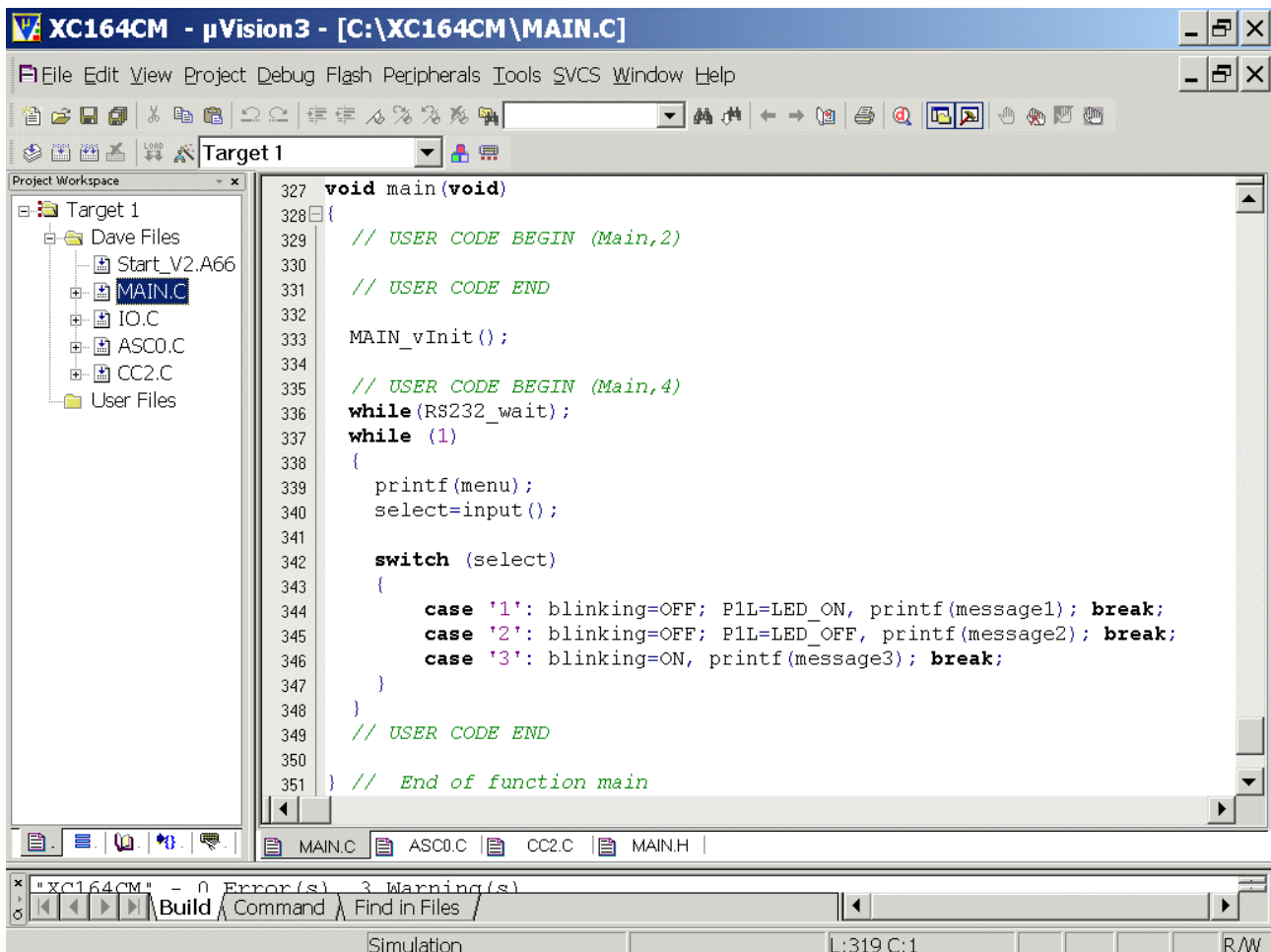
choose: **C:\XC164CM**

choose: File type: **Project Files (*.uv2)**

choose: **XC164CM.uv2**

Öffnen/Open

(Do you want to migrate the changes? **click JA/Yes**)



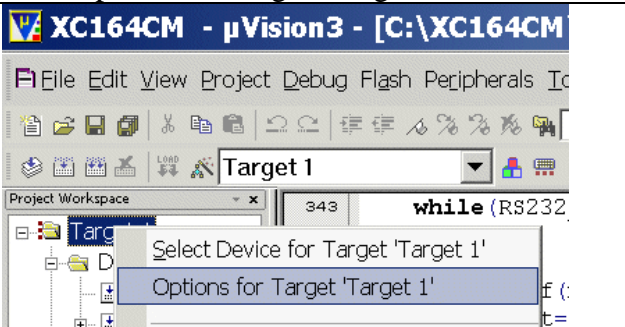
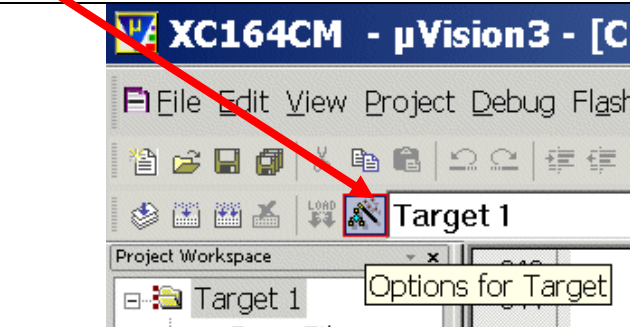
The screenshot shows the Keil μ Vision3 IDE interface. The title bar reads "XC164CM - μ Vision3 - [C:\XC164CM\MAIN.C]". The menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and development. The "Project Workspace" on the left shows a tree view with "Target 1" containing "Dave Files", "Start_V2.A66", "MAIN.C", "IO.C", "ASC0.C", "CC2.C", and "User Files". The main editor window displays the following C code:

```

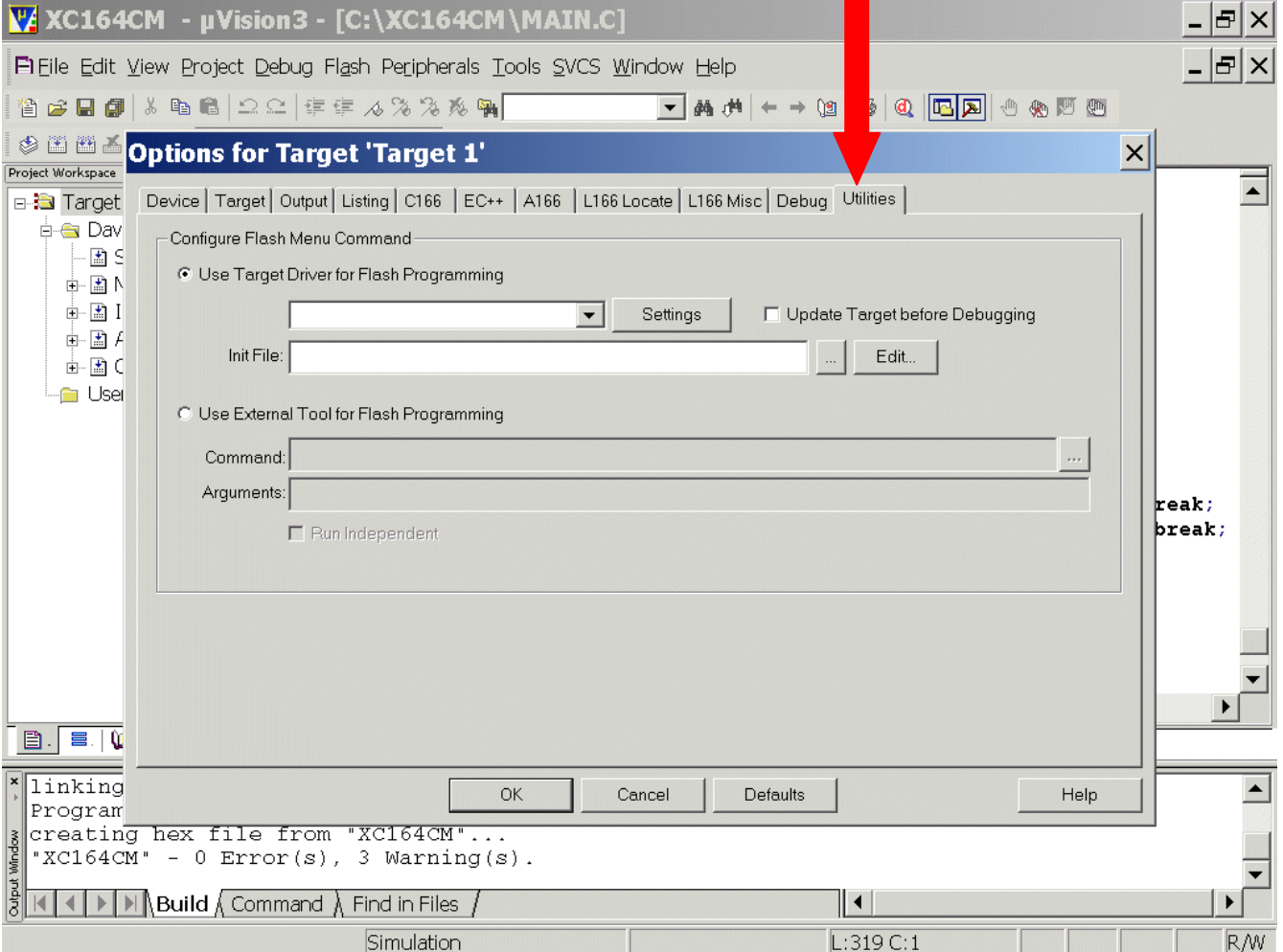
327 void main(void)
328 {
329     // USER CODE BEGIN (Main,2)
330
331     // USER CODE END
332
333     MAIN_vInit();
334
335     // USER CODE BEGIN (Main,4)
336     while(RS232_wait);
337     while (1)
338     {
339         printf(menu);
340         select=input();
341
342         switch (select)
343         {
344             case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
345             case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
346             case '3': blinking=ON, printf(message3); break;
347         }
348     }
349     // USER CODE END
350
351 } // End of function main

```

The status bar at the bottom shows "Simulation" and "L:319 C:1".

<p>mouse position: (Project Window, Files): Target 1: click right mouse button click Options for Target 'Target 1'</p>	<p>or: click Options for Target</p>
	

Options for Target 'Target 1': Utilities

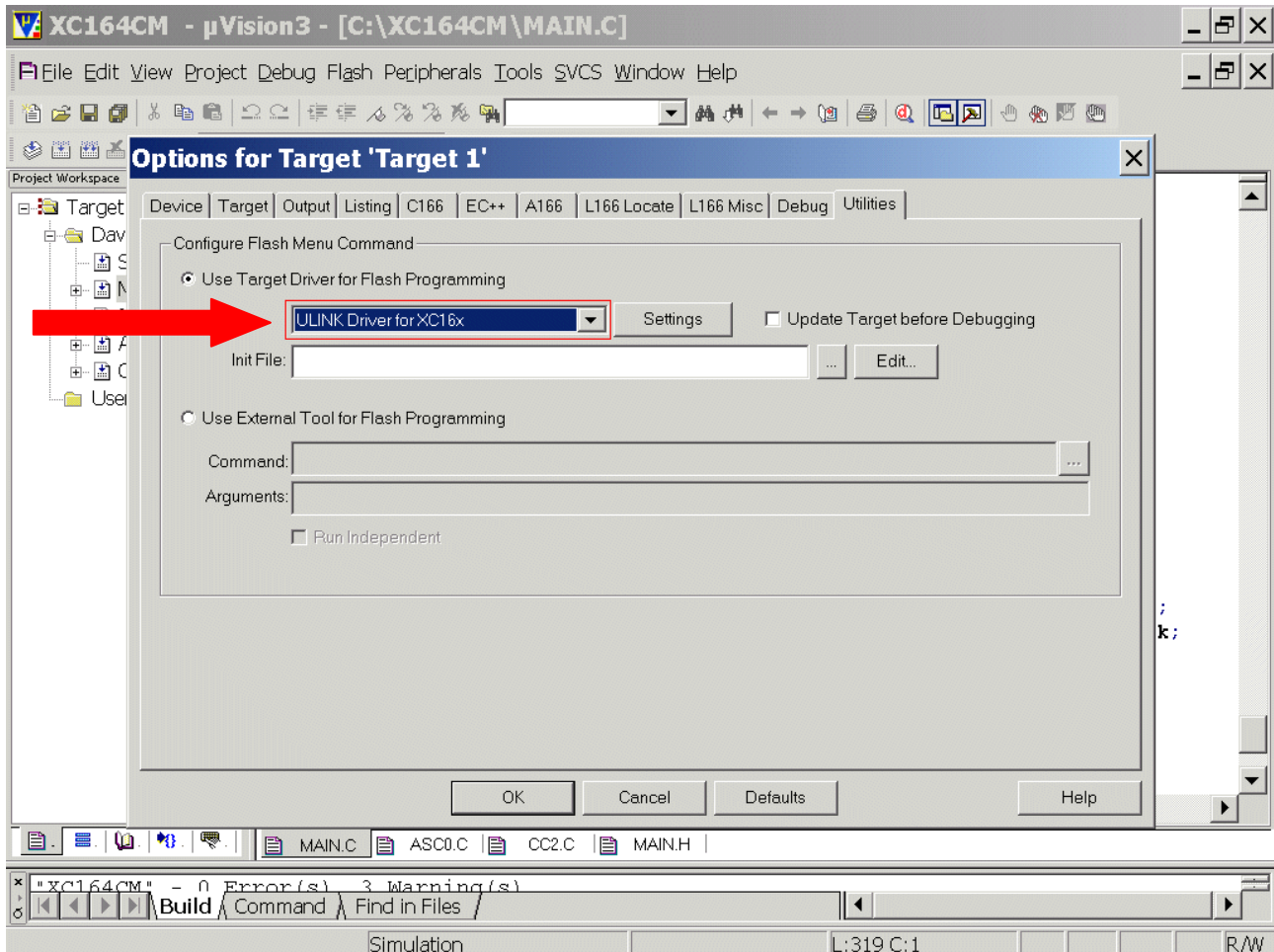


The screenshot shows the 'Options for Target' dialog box with the 'Utilities' tab selected. The dialog is titled 'Options for Target 'Target 1'' and has tabs for Device, Target, Output, Listing, C166, EC++, A166, L166 Locate, L166 Misc, Debug, and Utilities. The 'Utilities' tab is active, showing the 'Configure Flash Menu Command' section. Under this section, there are two radio button options: 'Use Target Driver for Flash Programming' (which is selected) and 'Use External Tool for Flash Programming'. The 'Use Target Driver' option includes a dropdown menu, a 'Settings' button, and a checkbox for 'Update Target before Debugging'. Below this, there are fields for 'Init File:' with '...' and 'Edit...' buttons. The 'Use External Tool' option includes fields for 'Command:' and 'Arguments:', both with '...' buttons, and a checkbox for 'Run Independent'. At the bottom of the dialog are 'OK', 'Cancel', 'Defaults', and 'Help' buttons. In the background, the IDE's output window shows the text: 'linking Program', 'creating hex file from "XC164CM"...', and '"XC164CM" - 0 Error(s), 3 Warning(s)'. The status bar at the bottom indicates 'Simulation' and 'L:319 C:1'.

Options for Target 'Target 1': Utilities

Configure Flash Menu Command: Use Target Driver for Flash Programming:

Select ULINK Driver for XC16x

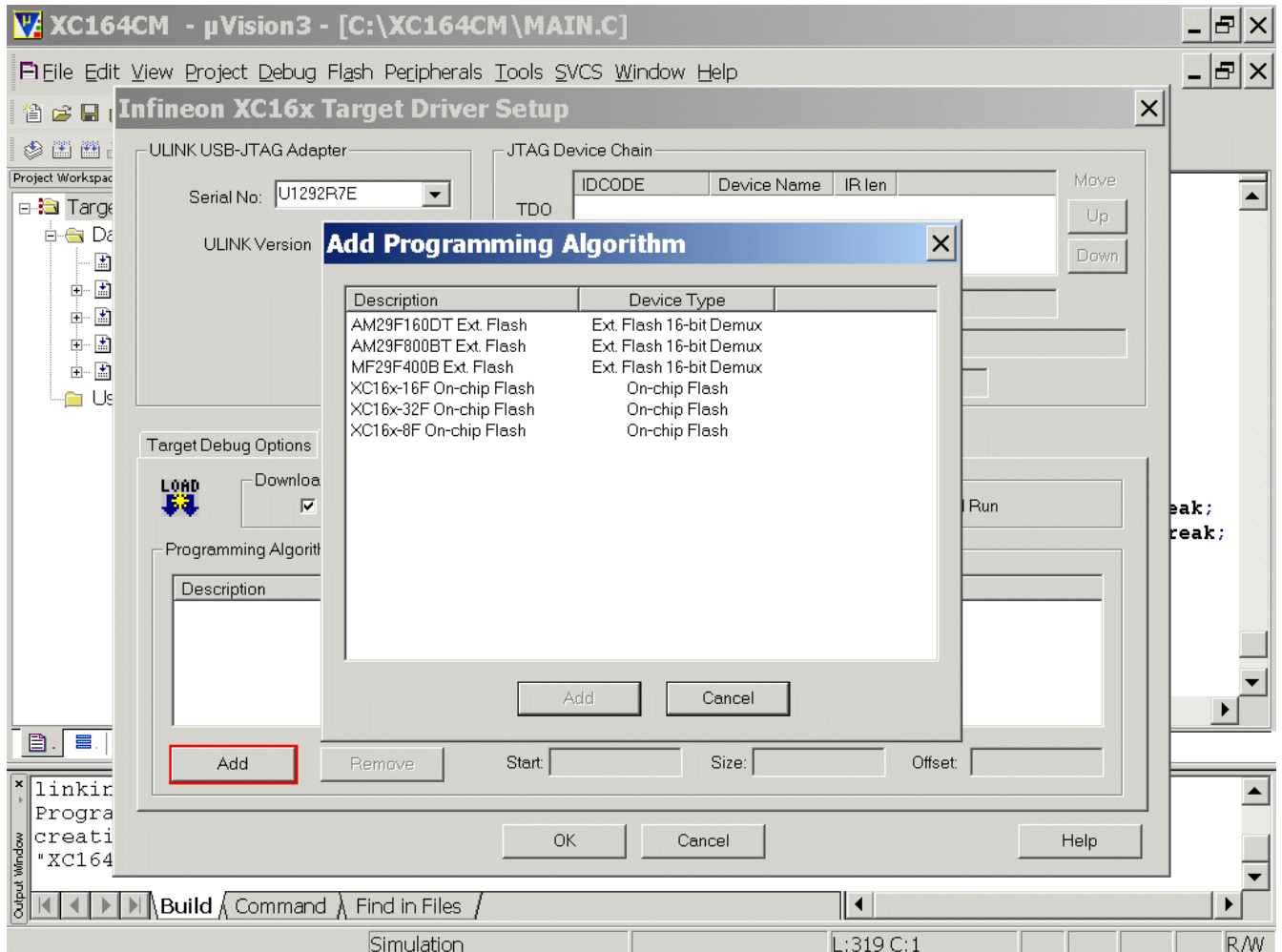


Options for Target 'Target 1': Utilities

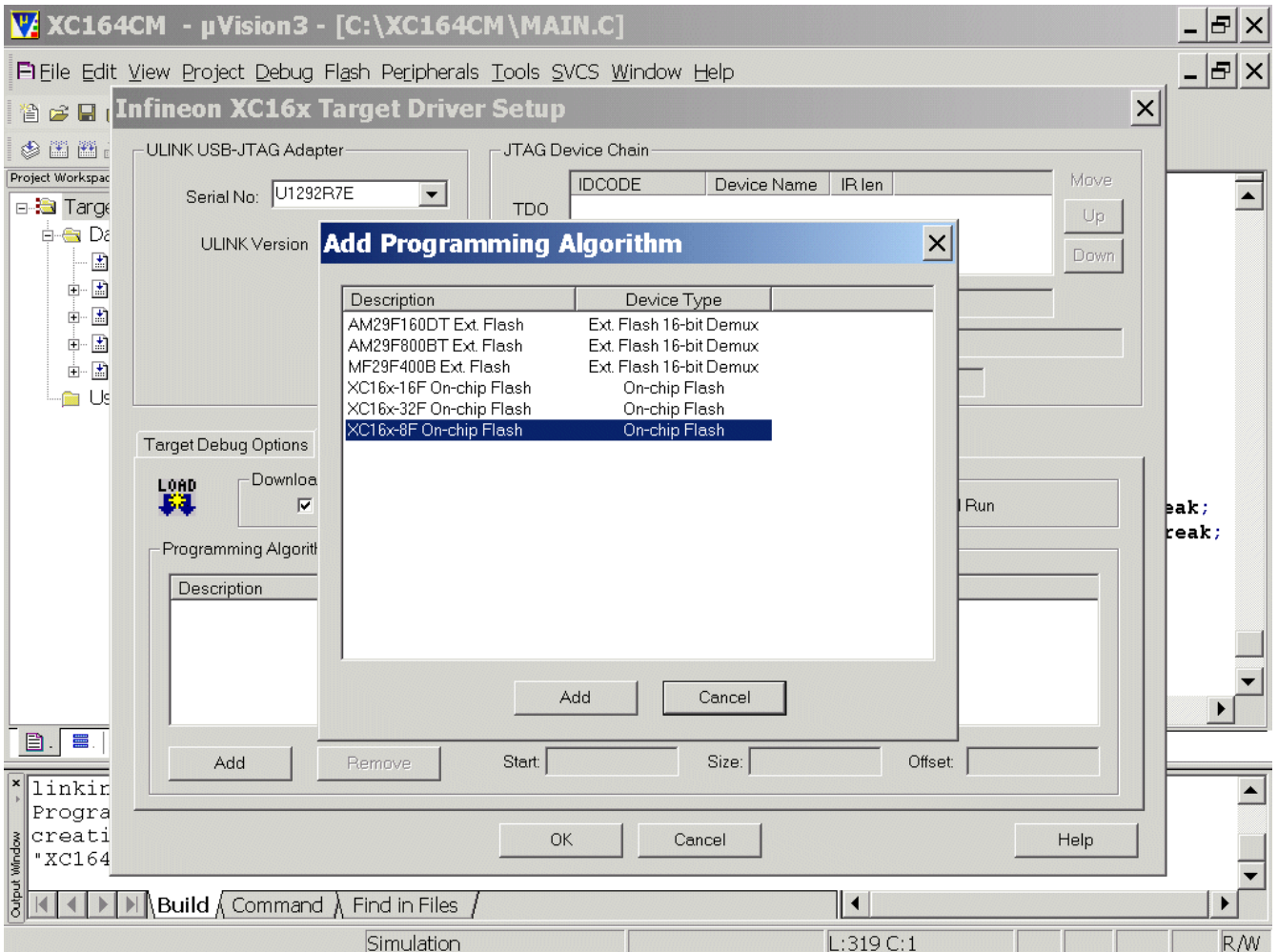
Configure Flash Menu Command: Use Target Driver for Flash Programming:

Click Settings

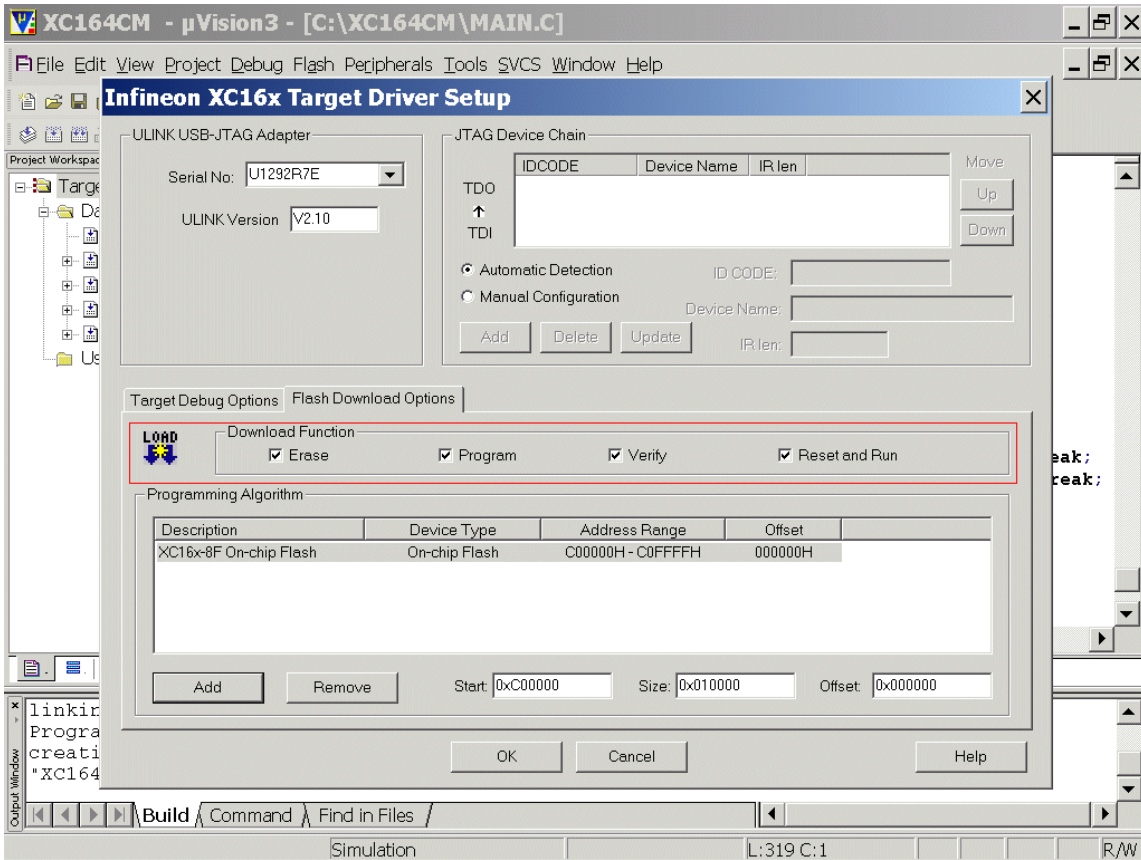
Click Add



Add Programming Algorithm: select XC16x-8F on-chip Flash

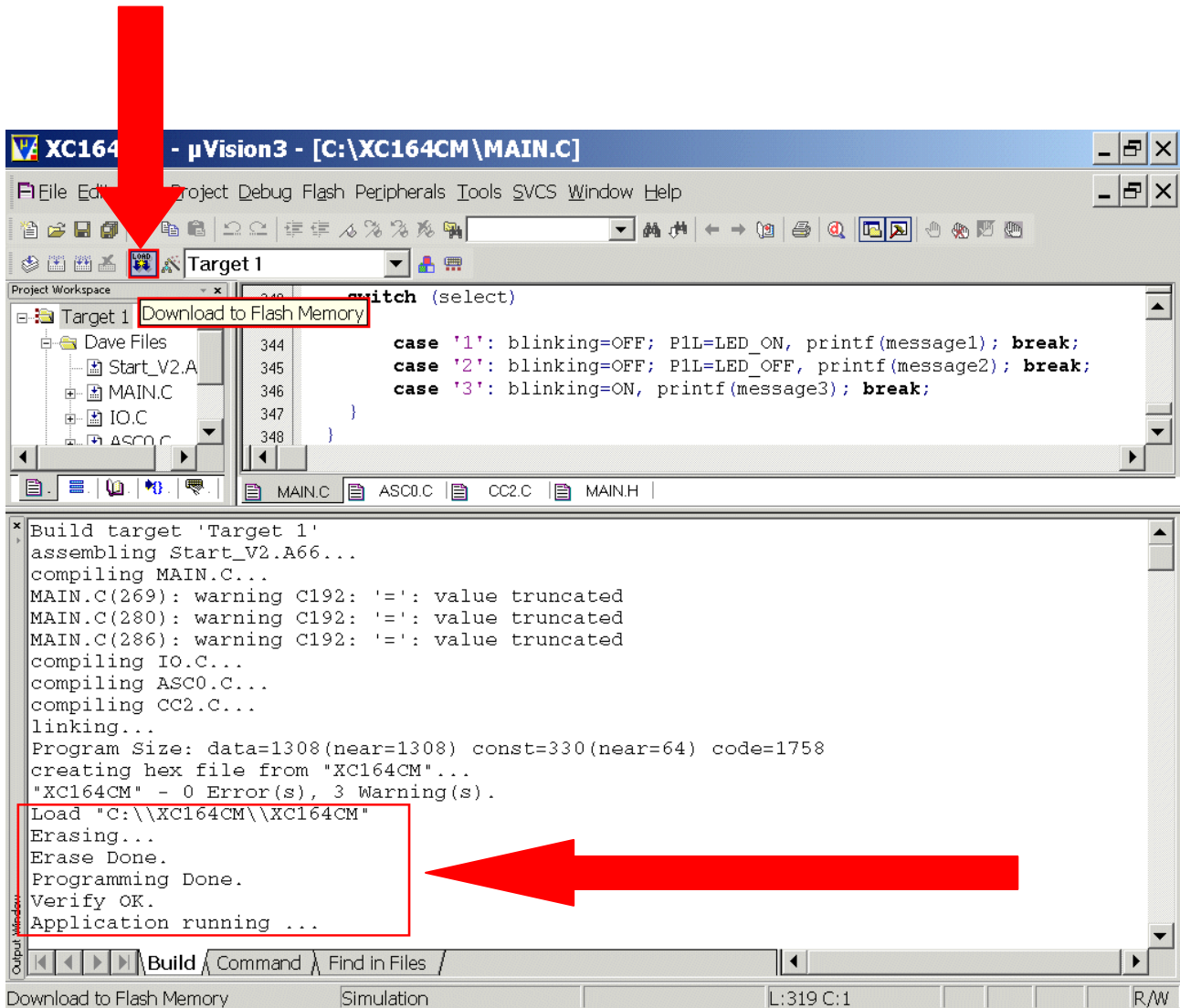


click Add



OK
OK

And see the result:

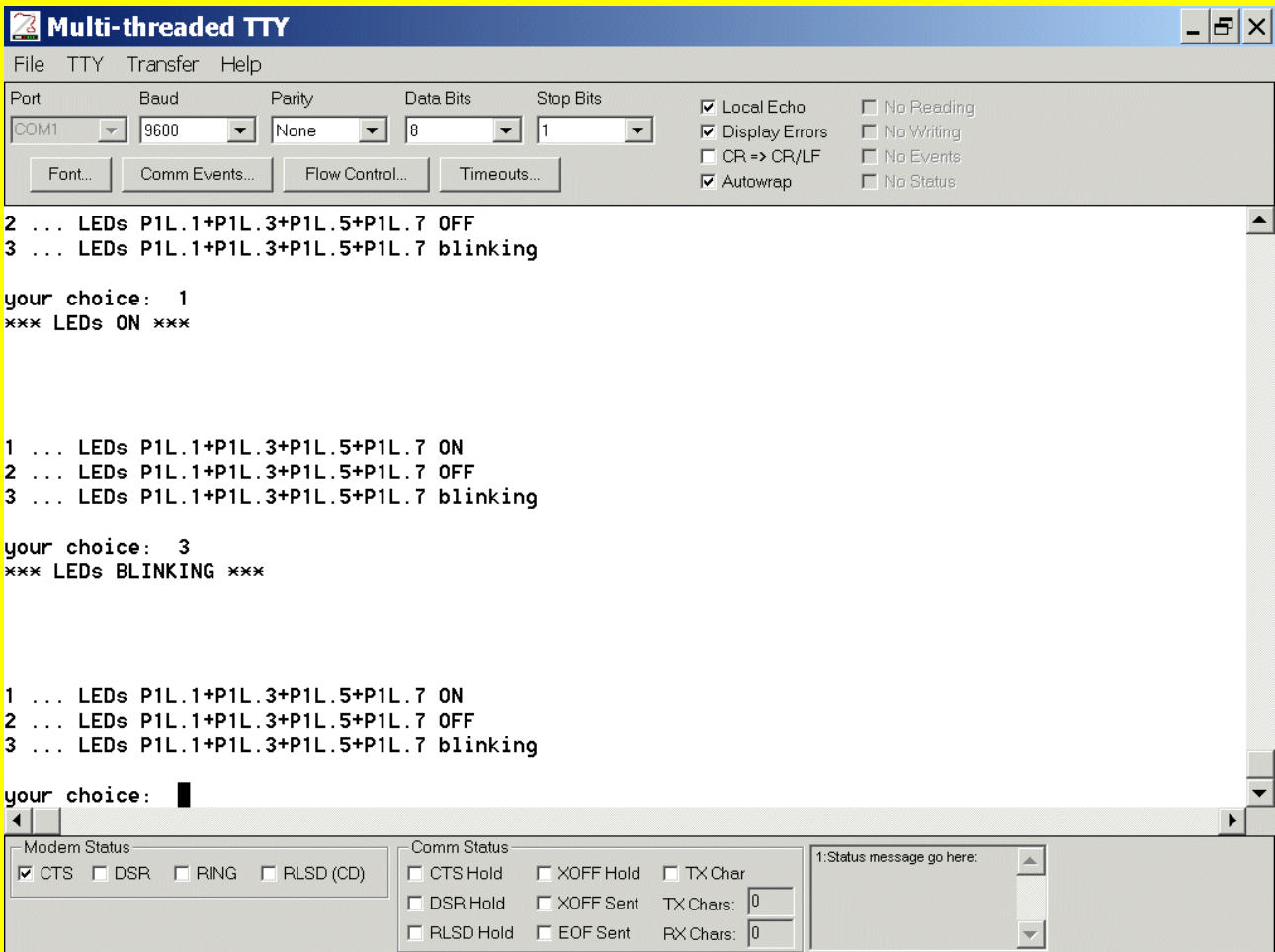


The screenshot shows the µVision3 IDE interface. The toolbar at the top contains a button labeled "Download to Flash Memory" which is highlighted with a red box and a red arrow pointing to it from the text above. The main editor window displays a C code snippet with a switch statement. The output window at the bottom shows the build process, including compilation of MAIN.C, IO.C, and ASC0.C, and the final step of loading the program to flash memory. A red box highlights the following output text, with a red arrow pointing from it back to the text above:

```

Load "C:\\XC164CM\\XC164CM"
Erasing...
Erase Done.
Programming Done.
Verify OK.
Application running ...
  
```

The status bar at the bottom of the IDE shows "Download to Flash Memory" as the active command, "Simulation" as the mode, and "L:319 C:1" as the current location.



OCDS-Debugging via KEIL-ULINK-JTAG-Interface

<p>mouse position: (Project Window, Files): Target 1: click right mouse button click Options for Target 'Target 1'</p>	<p>or: click Options for Target</p>

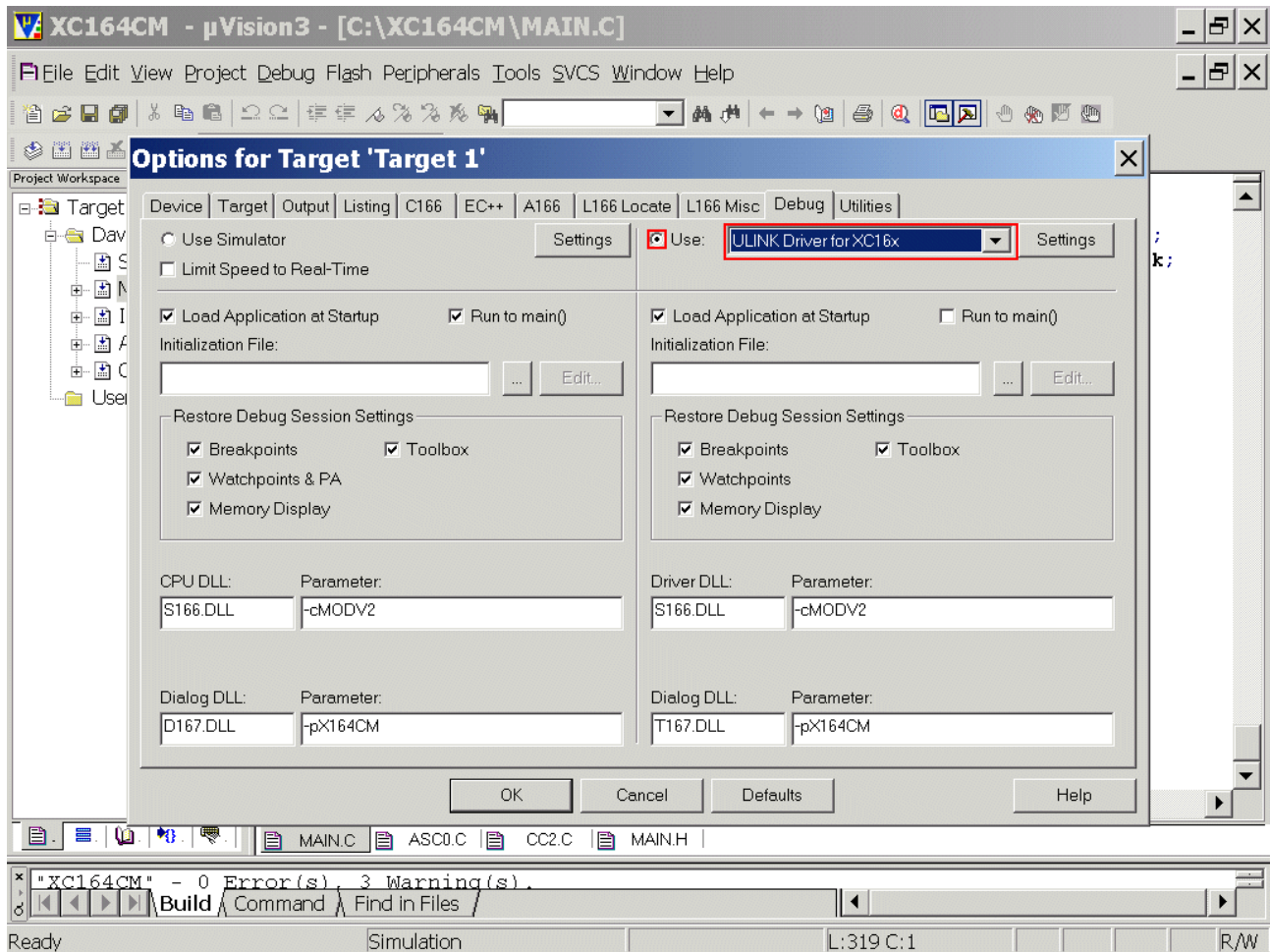
Options for Target 'Target 1': Debug

The 'Options for Target' dialog box is shown with the 'Debug' tab active. The 'Use Simulator' radio button is selected. The 'Keil Monitor-166 Driver' is chosen in the 'Use:' dropdown. The 'Load Application at Startup' checkbox is checked, and 'Run to main()' is also checked. The 'Restore Debug Session Settings' section has 'Breakpoints', 'Watchpoints & PA', 'Memory Display', and 'Toolbox' checked. The 'CPU DLL' is 'S166.DLL' with parameter '-cMODV2'. The 'Driver DLL' is 'S166.DLL' with parameter '-cMODV2'. The 'Dialog DLL' is 'D167.DLL' with parameter '-pXC164CM'. The 'Output Window' at the bottom shows the message: 'linking Program', 'creating hex file from "XC164CM"...', and '"XC164CM" - 0 Error(s), 3 Warning(s)'. The status bar at the bottom indicates 'Ready', 'Keil Monitor-166 Driver', 'L:319 C:1', and 'R/W'.

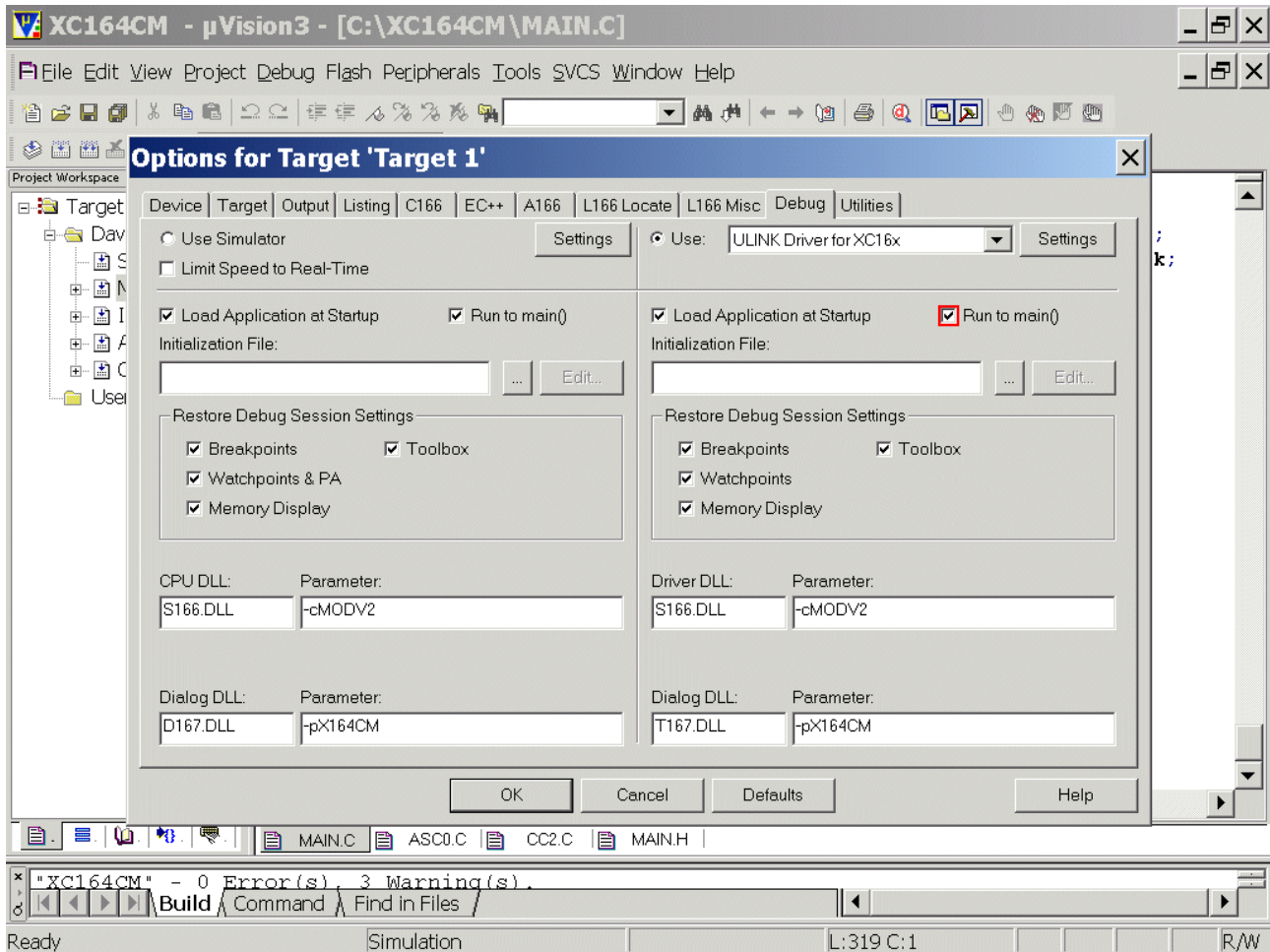
Options for Target 'Target 1': Debug

Click  Use :

Select ULINK Driver for XC16x



click ✓ Run to main():



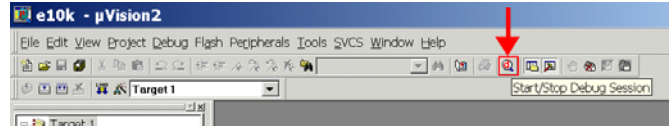
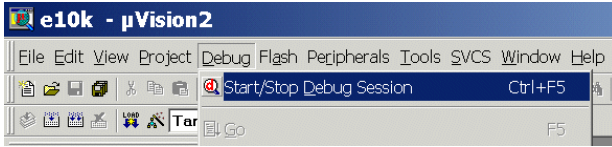
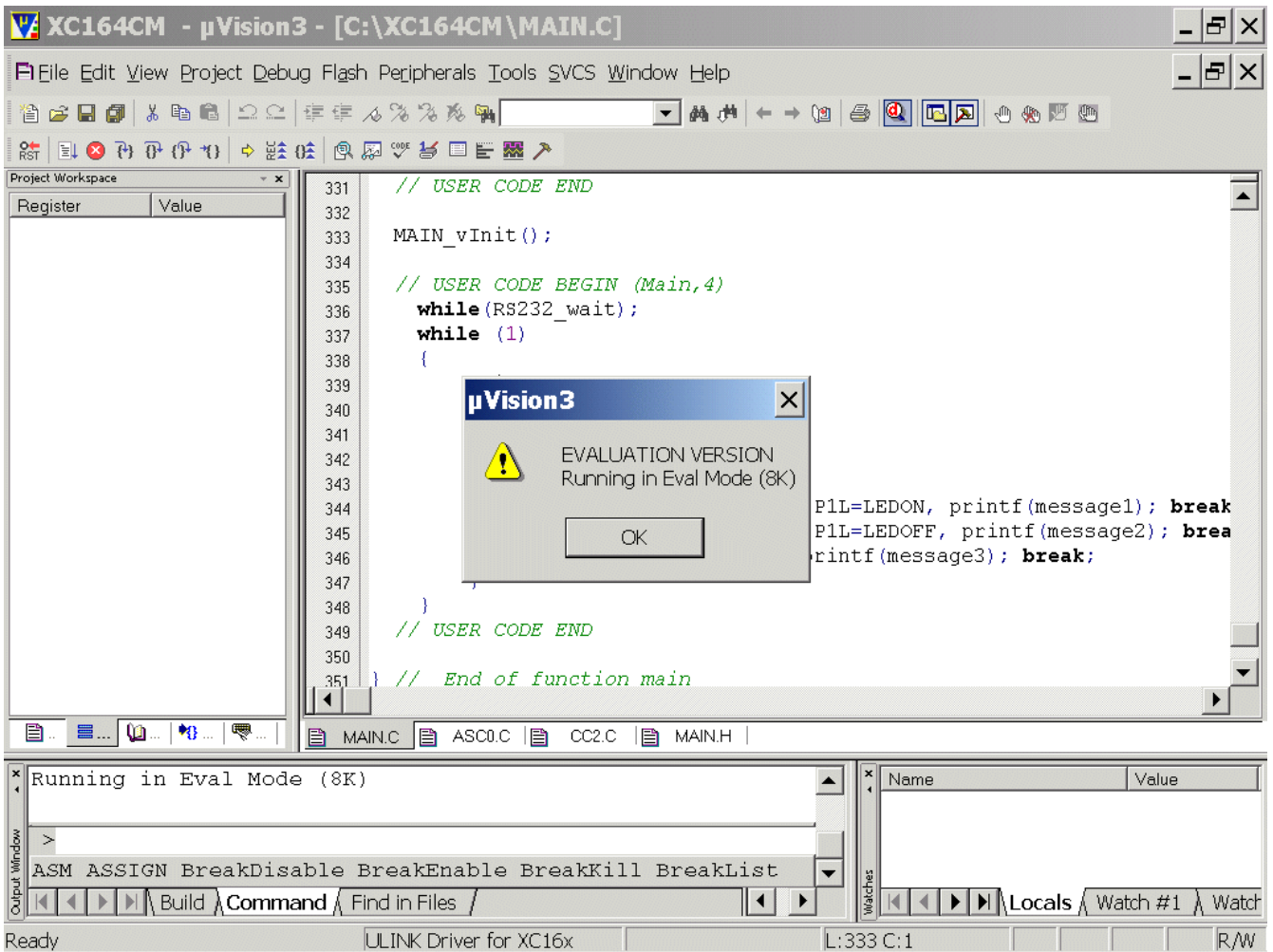
OK

Now, you can debug with OCDS via KEIL-ULINK-JTAG-Interface:

Debug – Start/Stop Debug Session

or

click 

The screenshot shows the Keil uVision3 IDE interface. The main window displays a C source file with the following code:

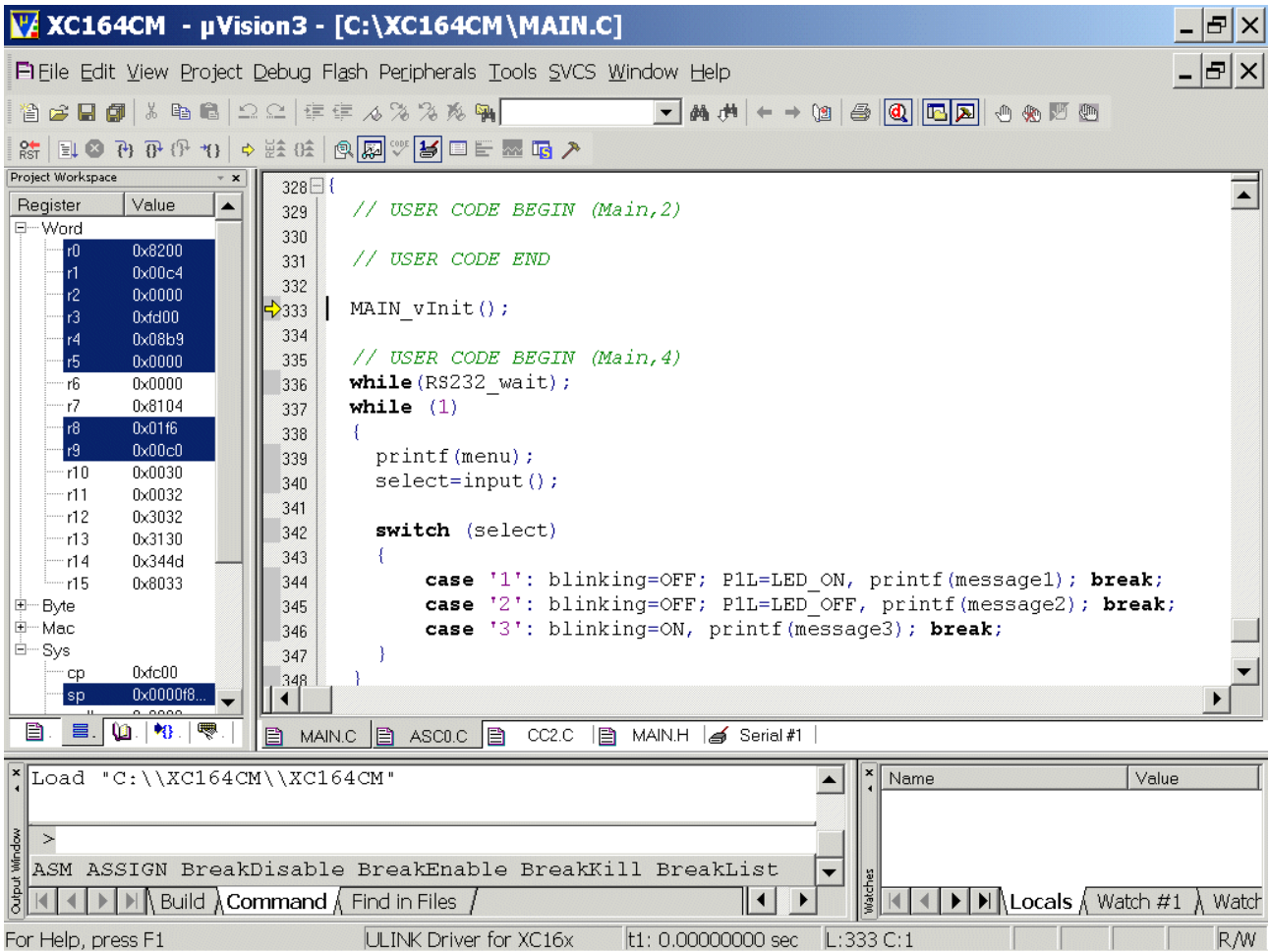
```

331 // USER CODE END
332
333 MAIN_vInit();
334
335 // USER CODE BEGIN (Main,4)
336 while (RS232_wait);
337 while (1)
338 {
339
340
341
342
343 P1L=LEDON, printf(message1); break
344 P1L=LEDOFF, printf(message2); brea
345 rintf(message3); break;
346
347
348 }
349 // USER CODE END
350
351 // End of function main

```

A dialog box titled "uVision3" is overlaid on the code, containing a warning icon and the text "EVALUATION VERSION Running in Eval Mode (8K)". The status bar at the bottom of the IDE shows "Running in Eval Mode (8K)".

OK



The screenshot shows the µVision3 IDE interface for the XC164CM project. The main window displays the source code for MAIN.C, which includes user code for initialization and a menu loop. The Project Workspace on the left shows the register values for the device.

Project Workspace - Register Values:

Register	Value
r0	0x8200
r1	0x00c4
r2	0x0000
r3	0x1d00
r4	0x08b9
r5	0x0000
r6	0x0000
r7	0x8104
r8	0x01f6
r9	0x00c0
r10	0x0030
r11	0x0032
r12	0x3032
r13	0x3130
r14	0x344d
r15	0x8033

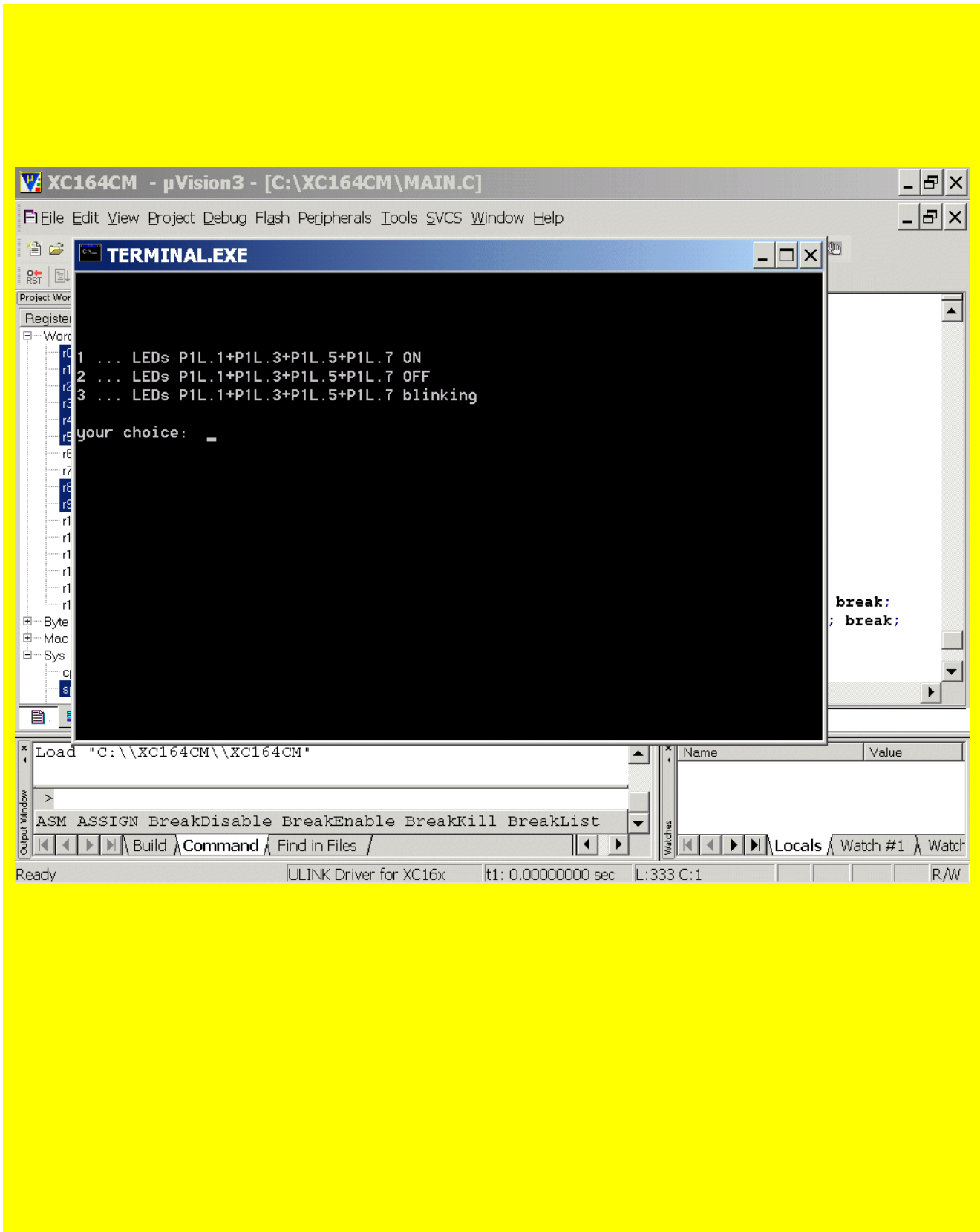
Source Code (MAIN.C):

```

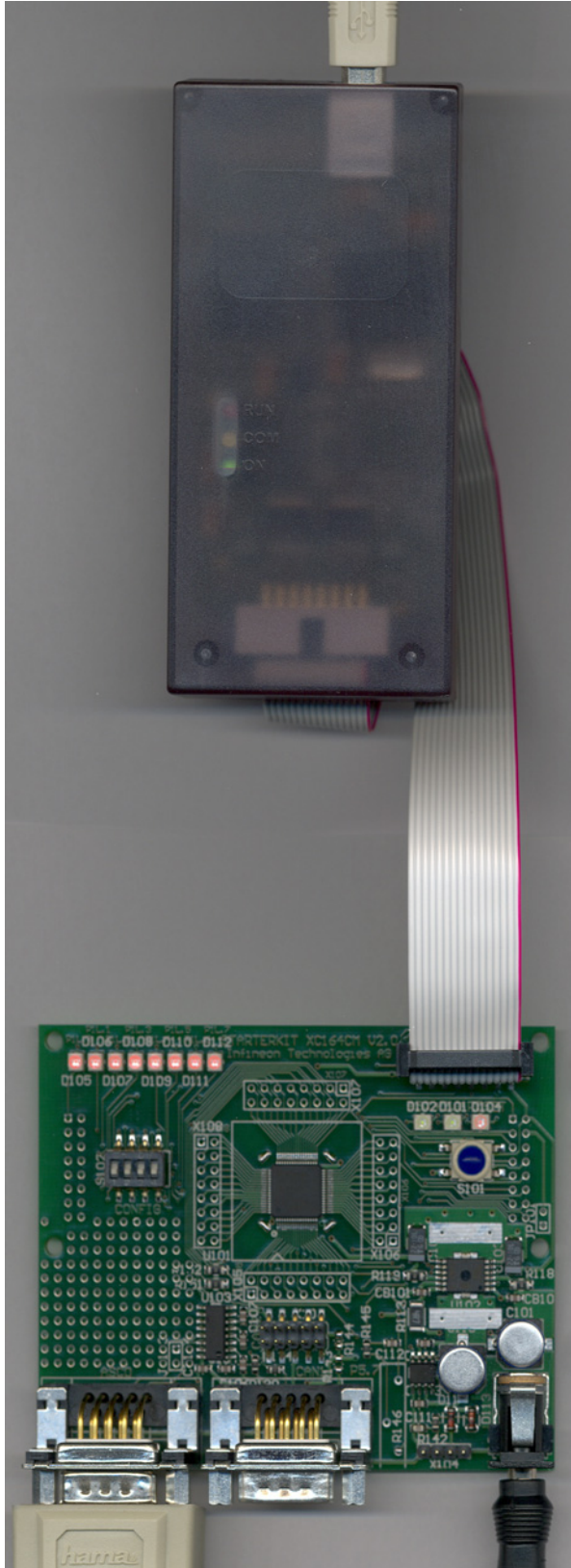
328 {
329 // USER CODE BEGIN (Main,2)
330
331 // USER CODE END
332
333 MAIN_vInit();
334
335 // USER CODE BEGIN (Main,4)
336 while(RS232_wait);
337 while (1)
338 {
339     printf(menu);
340     select=input();
341
342     switch (select)
343     {
344         case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
345         case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
346         case '3': blinking=ON, printf(message3); break;
347     }
348 }
    
```

The Output Window shows the command "Load 'C:\\XC164CM\\XC164CM'" and the status bar indicates the linker driver is ULINK Driver for XC16x, with a time of 0.00000000 sec and location L:333 C:1.

Debug – Run

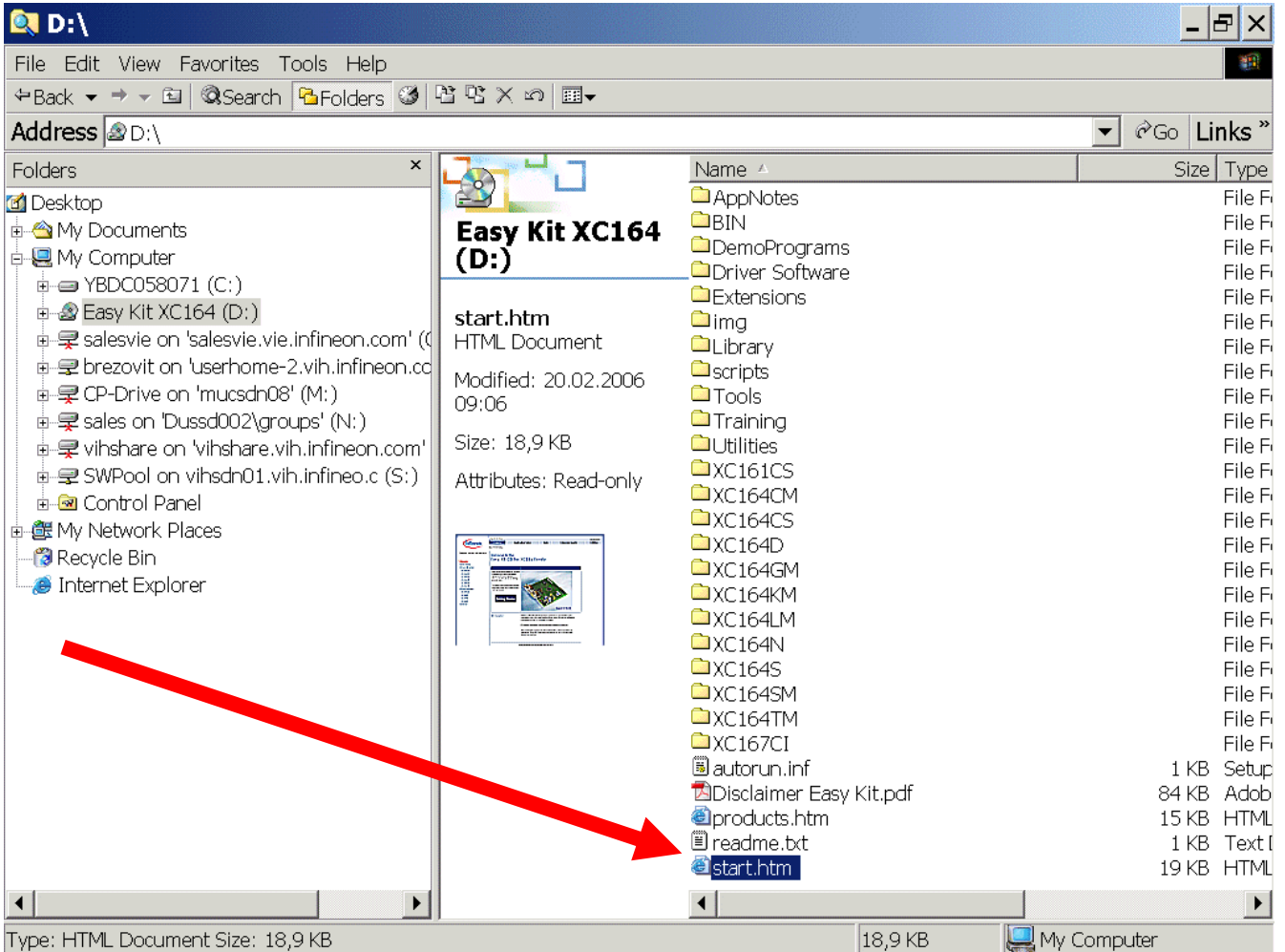


USB-JTAG-Wiggler-Box



OnChipFlash-Programming via USB-JTAG-Wiggler-Box

Easy Kit XC164 CD [click start.htm](#)



Click Getting Started

The screenshot shows a web browser window titled "Easy Kit CD for XC16x family - Infineon Technologies". The address bar contains "D:\start.htm". The website layout includes a top navigation bar with "Products", "Application Notes", "Tools", "Extension Boards", and "Utilities". A sidebar on the left lists product models: "XC161CJCS", "XC164CM series" (with sub-links for XC164CM, XC164GM, XC164KM, XC164LM, XC164SM, XC164TM), "XC164CS series" (with sub-links for XC164CS, XC164D, XC164N, XC164S), and "XC167CI". The main content area has a heading "Welcome to the Easy Kit CD for XC16x family" and a "Getting Started" button highlighted with a red arrow. Below the button is a photograph of the "Easy Kit XC164CM" board. A navigation instruction at the bottom reads: "Navigation: Select on the left side the product on which you would like to get information like User's Manual and Data Sheet, as well as hardware information on the XC16x Easy Kit board."

Getting Started - Infineon Technologies

File Edit View Favorites Tools Help

Address D:\Training\GettingStarted\GettingStarted.htm

Easy Kit CD Home Infineon Home

Products Application Notes Tools Extension Boards Utilities

Easy Kit CD Home > Getting Started

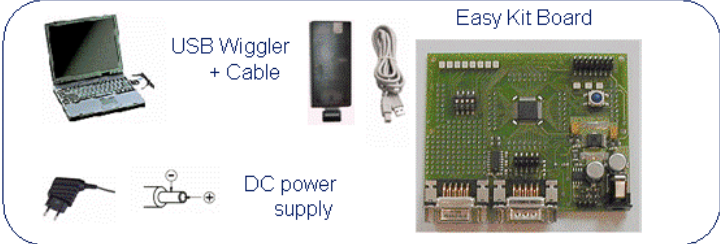
Never stop thinking.

Welcome
[XC161C/JCS](#)
XC164CM series
[XC164CM](#)
[XC164GM](#)
[XC164KM](#)
[XC164LM](#)
[XC164SM](#)
[XC164TM](#)
XC164CS series
[XC164CS](#)
[XC164D](#)
[XC164N](#)
[XC164S](#)
[XC167CI](#)

Getting Started -> Step 1 -> Hardware Setup

Step 1 Step 2 Step 3

To Setup the Hardware:



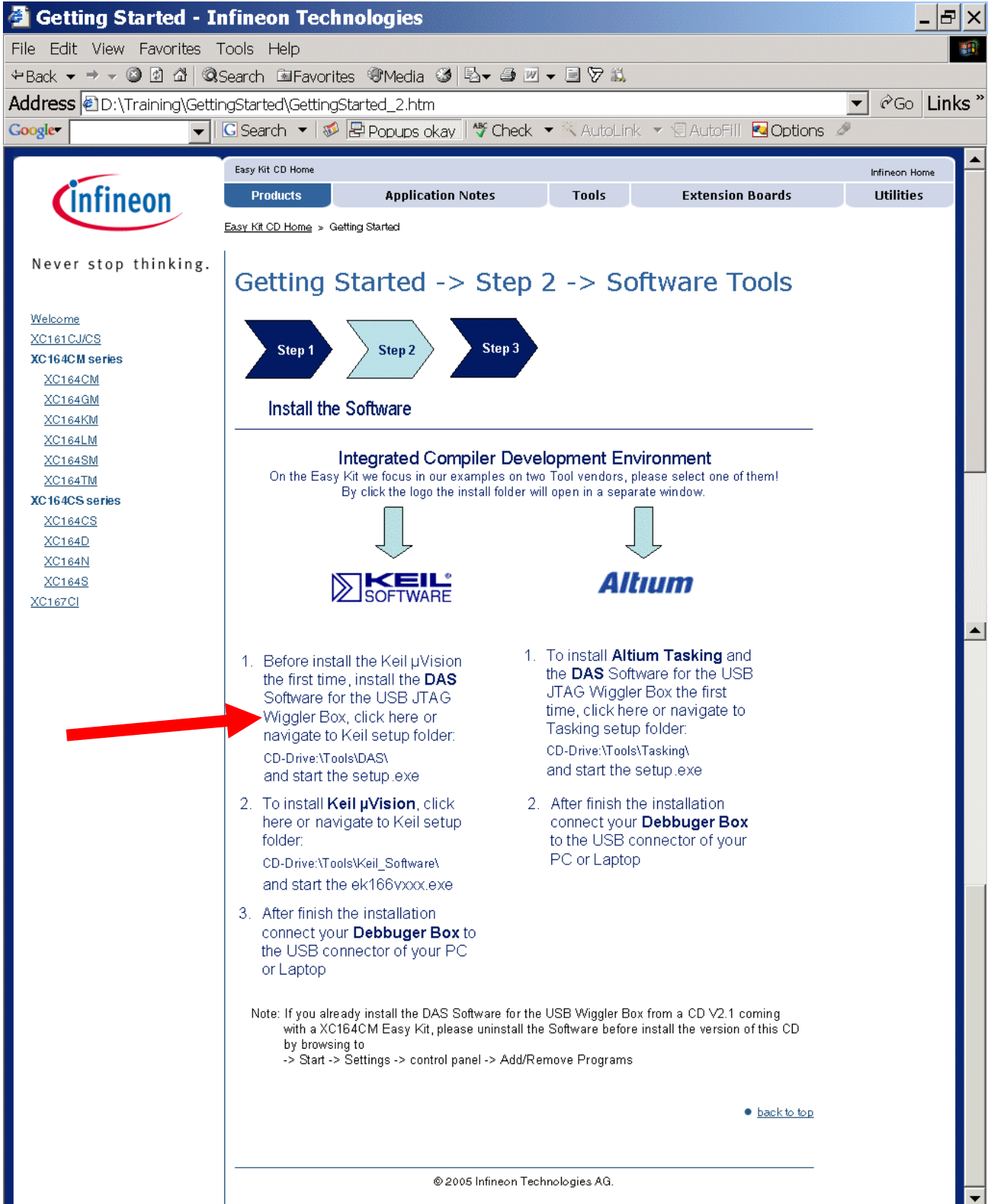
USB Wiggler + Cable

DC power supply

Easy Kit Board

- 1. Connect your DC power** supply with to the Easy Kit Board (ensure correct polarity) → the two green LED for 5V and 2,5V will switch on.
Note: The Power Supply (7-12 Volts, min 400mA) is not included in the Easy Kit!
- 2. The USB Wiggler** has to be connected to the PC after Step 2 (Software Installation)

Install (click) DAS-Software



Getting Started - Infineon Technologies

File Edit View Favorites Tools Help

Address D:\Training\GettingStarted\GettingStarted_2.htm

Easy Kit CD Home | [Products](#) | [Application Notes](#) | [Tools](#) | [Extension Boards](#) | [Utilities](#) | Infineon Home

Easy Kit CD Home > Getting Started

Never stop thinking.

Welcome

[XC161CJCS](#)

XC164CM series

- [XC164CM](#)
- [XC164GM](#)
- [XC164KM](#)
- [XC164LM](#)
- [XC164SM](#)
- [XC164TM](#)

XC164CS series

- [XC164CS](#)
- [XC164D](#)
- [XC164N](#)
- [XC164S](#)
- [XC167CI](#)


Getting Started -> Step 2 -> Software Tools


Step 1 → Step 2 → Step 3

Install the Software

Integrated Compiler Development Environment

On the Easy Kit we focus in our examples on two Tool vendors, please select one of them!
By click the logo the install folder will open in a separate window.





- Before install the Keil μ Vision the first time, install the **DAS Software for the USB JTAG Wiggler Box**, click here or navigate to Keil setup folder:
CD-Drive:\Tools\DAS\
and start the setup.exe
- To install **Keil μ Vision**, click here or navigate to Keil setup folder:
CD-Drive:\Tools\Keil_Software\
and start the ek166vxxx.exe
- After finish the installation connect your **Debugger Box** to the USB connector of your PC or Laptop

- To install **Altium Tasking** and the **DAS Software for the USB JTAG Wiggler Box** the first time, click here or navigate to Tasking setup folder:
CD-Drive:\Tools\Tasking\
and start the setup.exe
- After finish the installation connect your **Debugger Box** to the USB connector of your PC or Laptop

Note: If you already install the DAS Software for the USB Wiggler Box from a CD V2.1 coming with a XC164CM Easy Kit, please uninstall the Software before install the version of this CD by browsing to
-> Start -> Settings -> control panel -> Add/Remove Programs

[back to top](#)

© 2005 Infineon Technologies AG.

Start Keil μ Vision 3 and open the KEIL Project **XC164CM.Uv2**

If you see an open project – close it: **Project - Close Project**

Project - Open Project

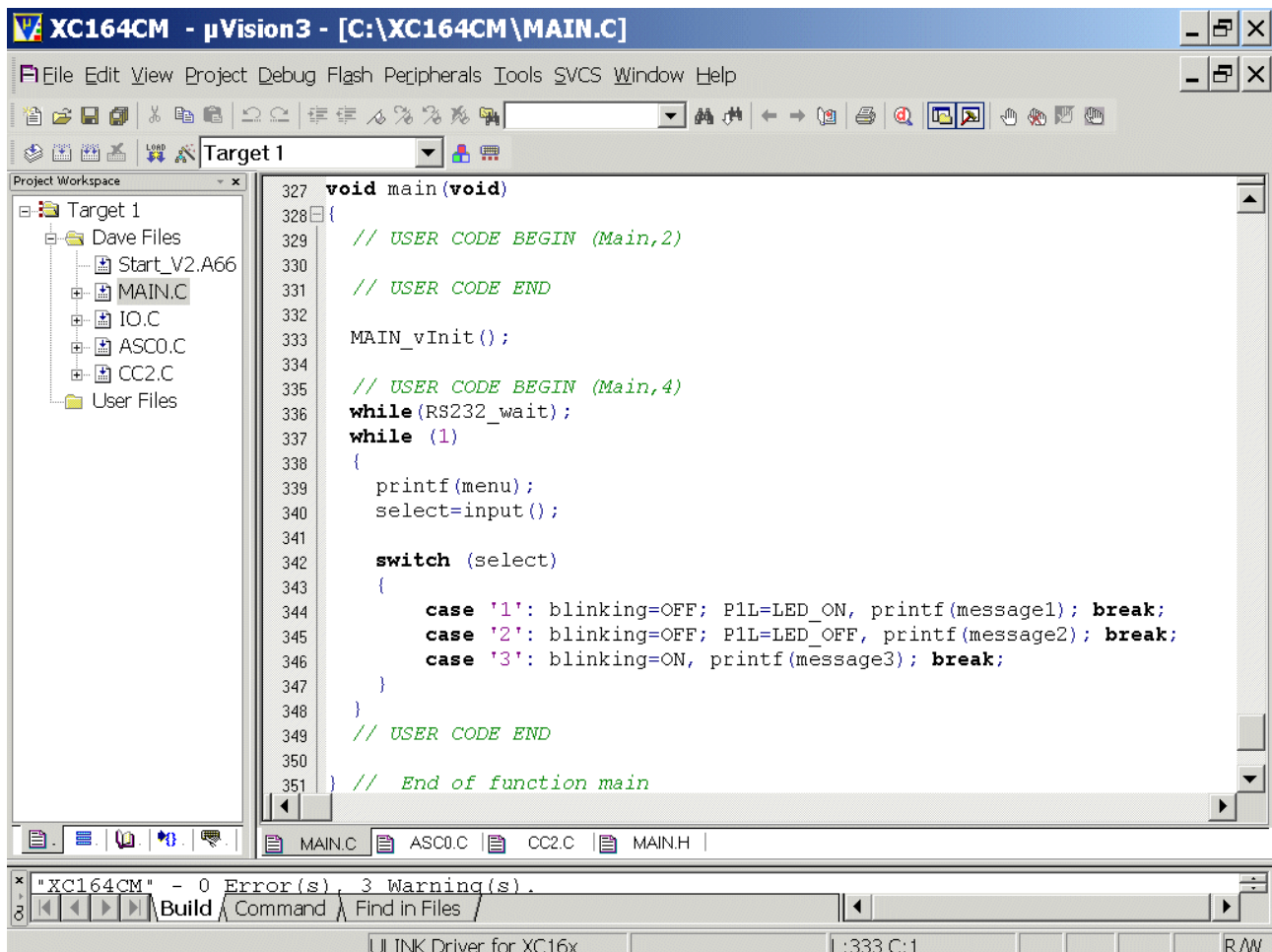
choose: **C:\XC164CM**

choose: File type: **Project Files (*.uv2)**

choose: **XC164CM.uv2**

Öffnen/Open

(Do you want to migrate the changes? **Click JA/Yes**)



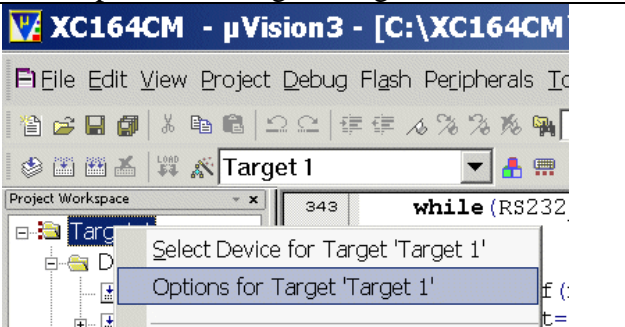
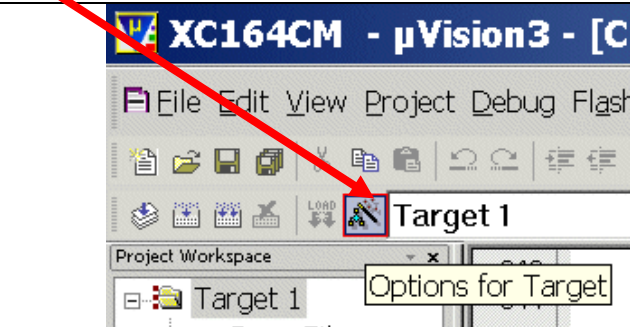
The screenshot shows the Keil μ Vision 3 IDE interface. The title bar reads "XC164CM - μ Vision3 - [C:\XC164CM\MAIN.C]". The menu bar includes File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, and Help. The toolbar contains various icons for file operations and development. The "Project Workspace" pane on the left shows a tree view with "Target 1" expanded, containing "Dave Files" (Start_V2.A66, MAIN.C, IO.C, ASC0.C, CC2.C) and "User Files". The main editor window displays the following C code:

```

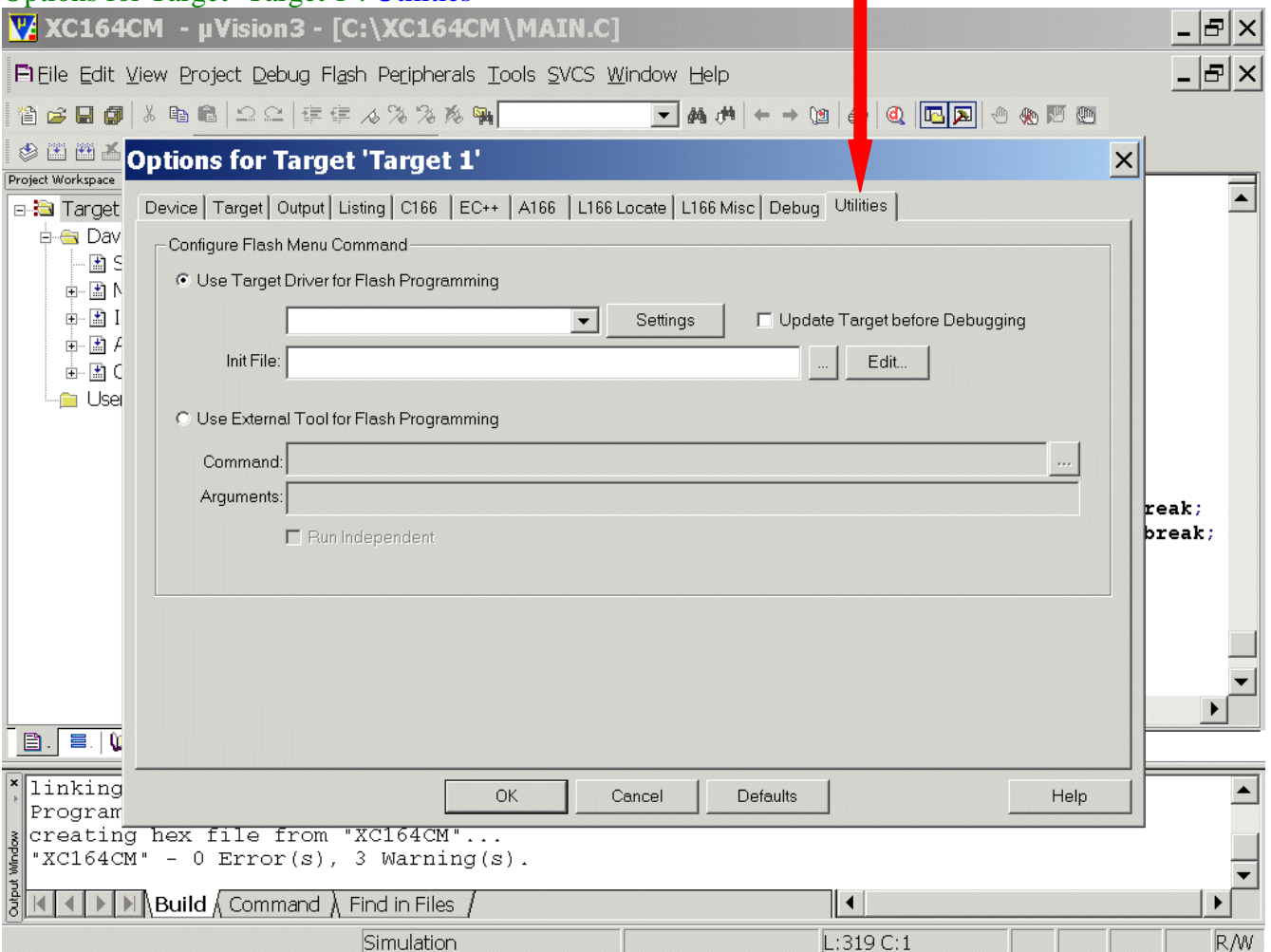
327 void main(void)
328 {
329     // USER CODE BEGIN (Main,2)
330
331     // USER CODE END
332
333     MAIN_vInit();
334
335     // USER CODE BEGIN (Main,4)
336     while(RS232_wait);
337     while (1)
338     {
339         printf(menu);
340         select=input();
341
342         switch (select)
343         {
344             case '1': blinking=OFF; P1L=LED_ON, printf(message1); break;
345             case '2': blinking=OFF; P1L=LED_OFF, printf(message2); break;
346             case '3': blinking=ON, printf(message3); break;
347         }
348     }
349     // USER CODE END
350
351 } // End of function main

```

The status bar at the bottom indicates "ULINK Driver for XC16x" and "L:333 C:1". The bottom-most status bar shows "R/W".

<p>mouse position: (Project Window, Files): Target 1: click right mouse button click Options for Target 'Target 1'</p>	<p>or: click Options for Target</p>
	

Options for Target 'Target 1': Utilities

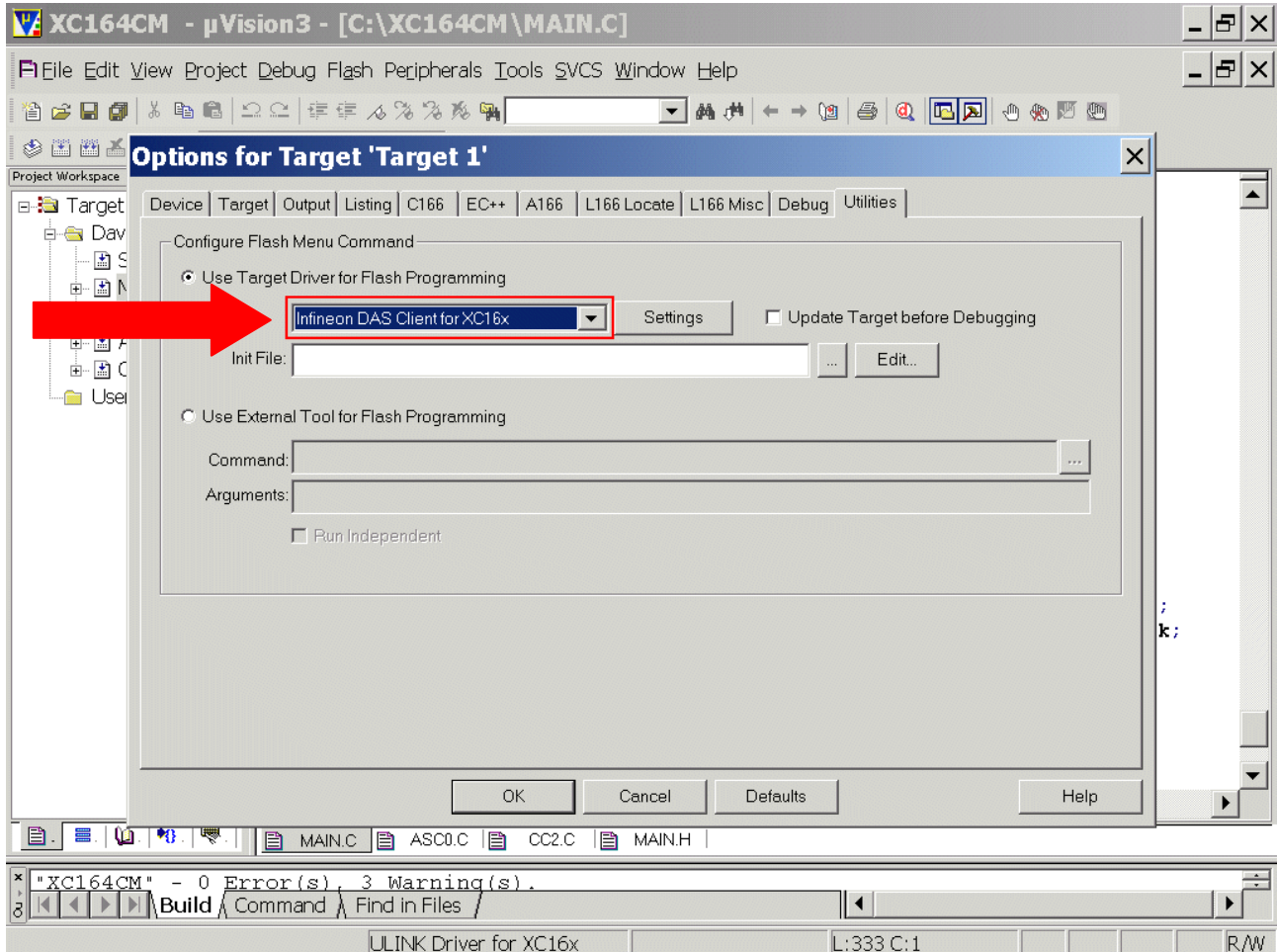


The screenshot shows the 'Options for Target' dialog box with the 'Utilities' tab selected. The dialog is titled 'Options for Target 'Target 1'' and has tabs for Device, Target, Output, Listing, C166, EC++, A166, L166 Locate, L166 Misc, Debug, and Utilities. The 'Utilities' tab is active, showing the 'Configure Flash Menu Command' section. Under this section, there are two radio button options: 'Use Target Driver for Flash Programming' (which is selected) and 'Use External Tool for Flash Programming'. The 'Use Target Driver' option includes a dropdown menu, a 'Settings' button, and a checkbox for 'Update Target before Debugging'. Below this, there are fields for 'Init File:' with '...' and 'Edit...' buttons. The 'Use External Tool' option includes fields for 'Command:' and 'Arguments:', both with '...' buttons, and a checkbox for 'Run Independent'. At the bottom of the dialog are 'OK', 'Cancel', 'Defaults', and 'Help' buttons. In the background, the IDE's output window shows the text: 'linking Program', 'creating hex file from "XC164CM"...', and '"XC164CM" - 0 Error(s), 3 Warning(s)'. The status bar at the bottom indicates 'Simulation' and 'L:319 C:1'.

Options for Target 'Target 1': Utilities

Configure Flash Menu Command: Use Target Driver for Flash Programming:

Select Infineon DAS Client for XC16x

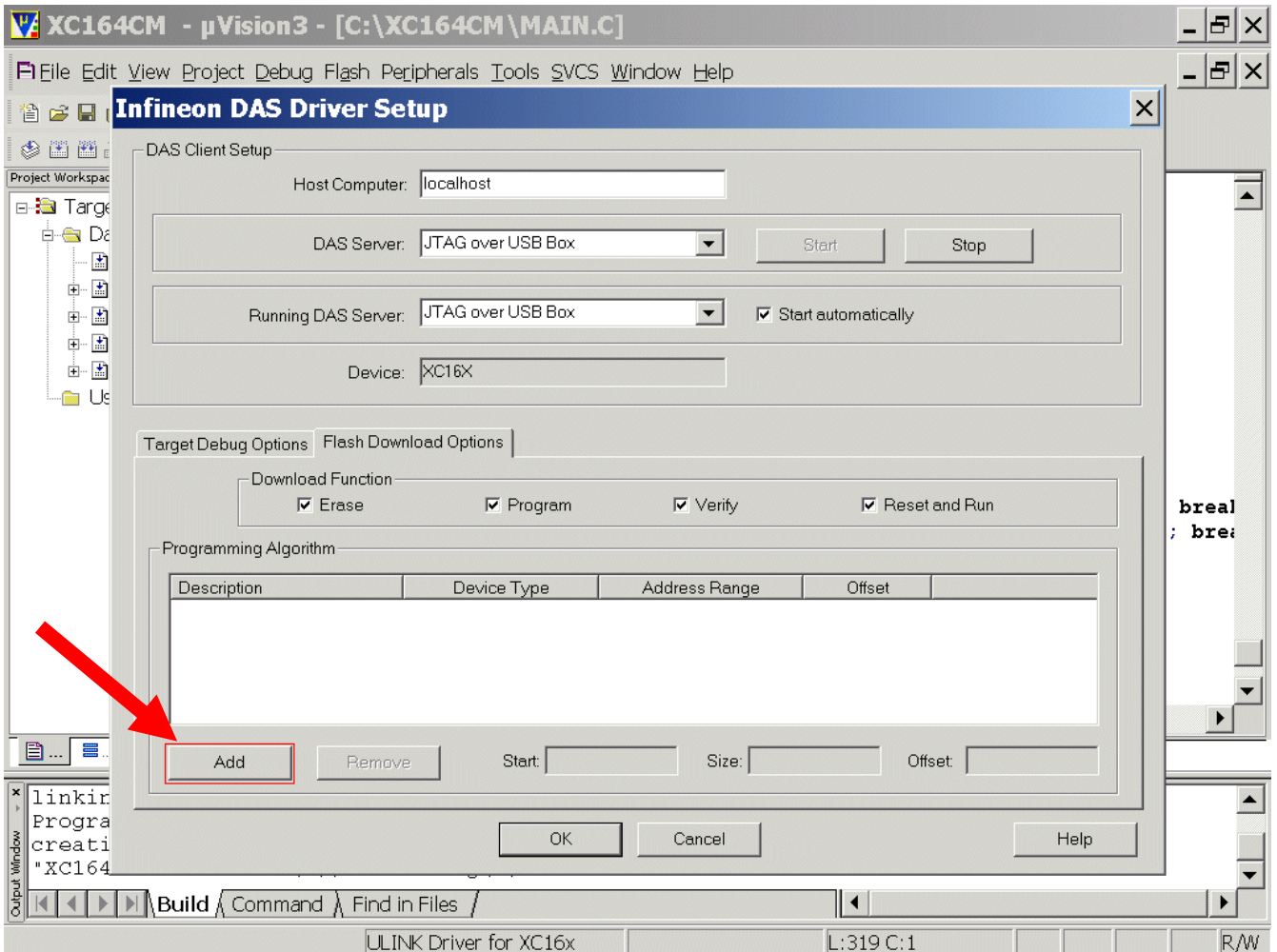


Options for Target 'Target 1': Utilities

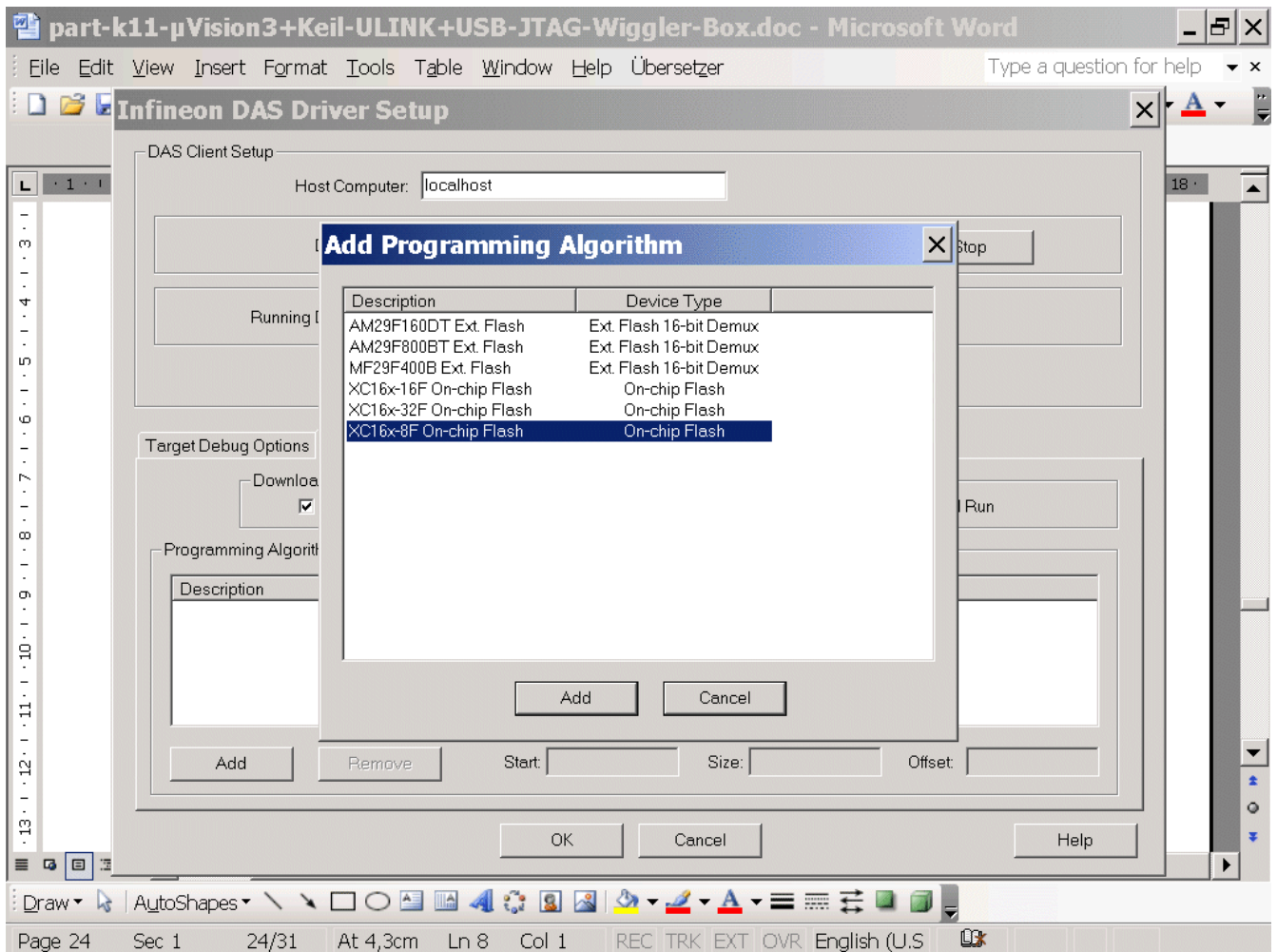
Configure Flash Menu Command: Use Target Driver for Flash Programming:

Click Settings

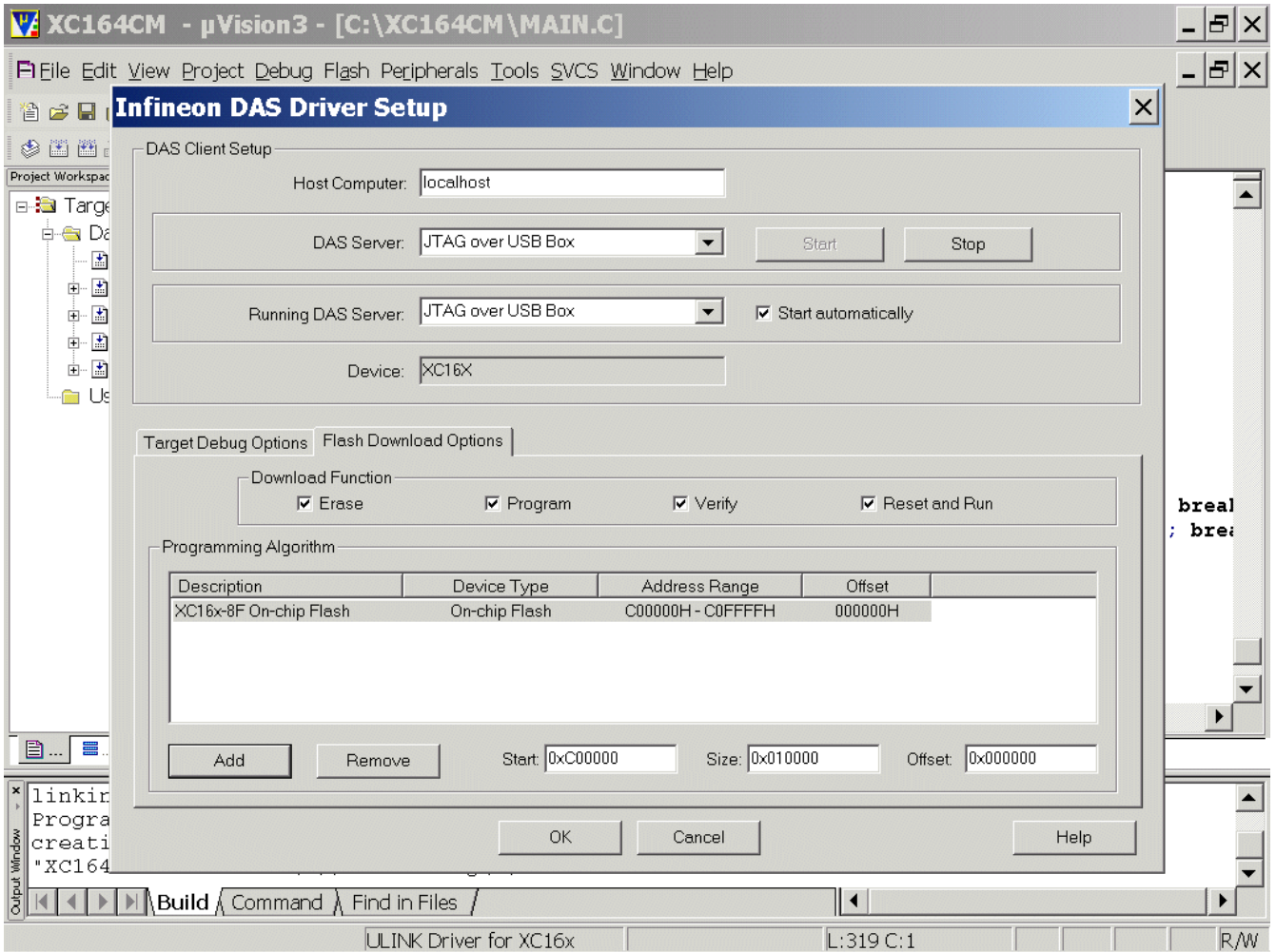
Click Add



Add Programming Algorithm: **select** XC16x-8F On-chip Flash

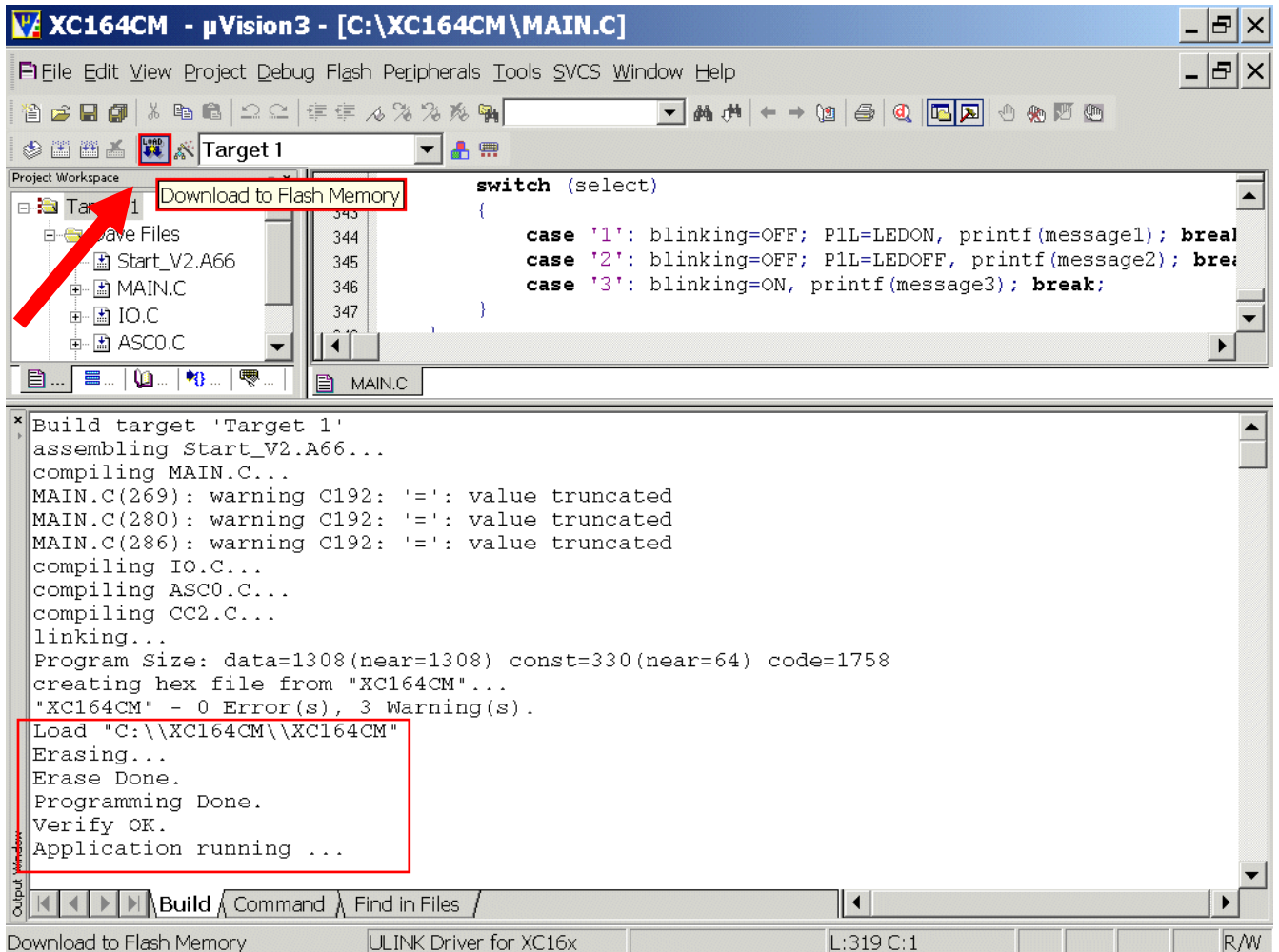


click Add



OK
OK

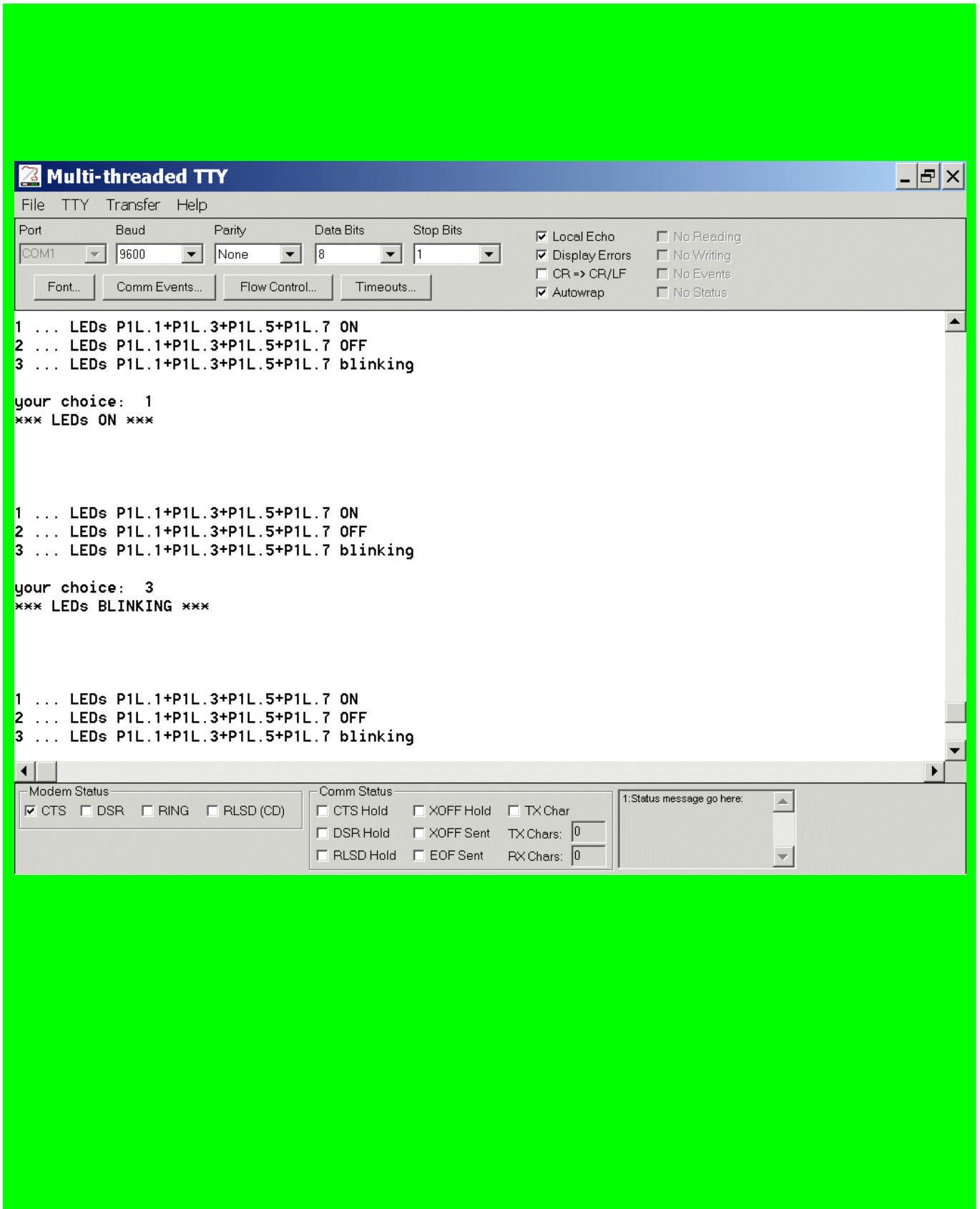
And see the result:



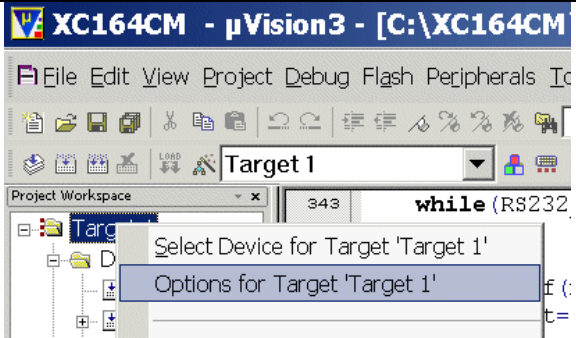
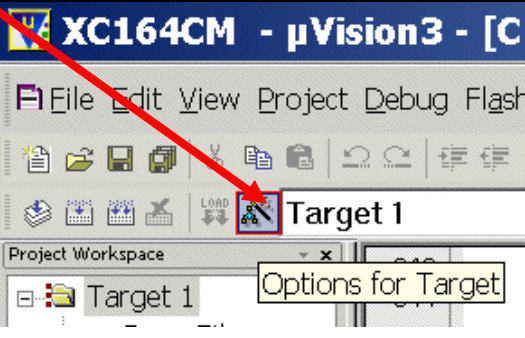
The screenshot shows the XC164CM development environment. The top window displays the project workspace and the source code for MAIN.C. A red arrow points to the 'Download to Flash Memory' button in the toolbar. The output window below shows the build process, including compilation and linking, and the successful download to flash memory.

```

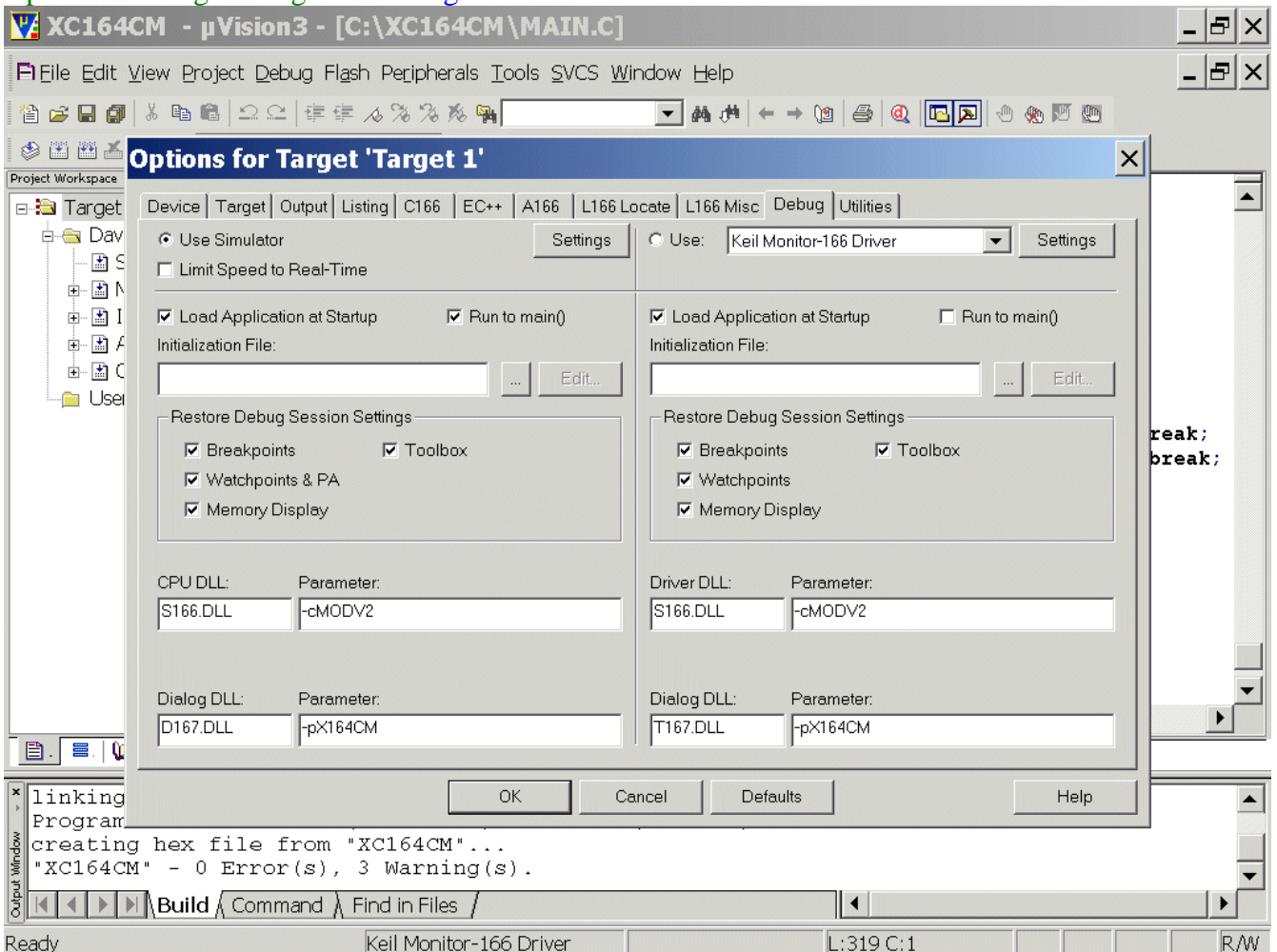
XC164CM - µVision3 - [C:\XC164CM\MAIN.C]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Target 1
Project Workspace
Download to Flash Memory
switch (select)
{
  case '1': blinking=OFF; P1L=LEDON, printf(message1); break;
  case '2': blinking=OFF; P1L=LEDOFF, printf(message2); break;
  case '3': blinking=ON, printf(message3); break;
}
Build target 'Target 1'
assembling Start_V2.A66...
compiling MAIN.C...
MAIN.C(269): warning C192: '=': value truncated
MAIN.C(280): warning C192: '=': value truncated
MAIN.C(286): warning C192: '=': value truncated
compiling IO.C...
compiling ASC0.C...
compiling CC2.C...
linking...
Program Size: data=1308(near=1308) const=330(near=64) code=1758
creating hex file from "XC164CM"...
"XC164CM" - 0 Error(s), 3 Warning(s).
Load "C:\XC164CM\XC164CM"
Erasing...
Erase Done.
Programming Done.
Verify OK.
Application running ...
Output Window
Build Command Find in Files
Download to Flash Memory ULINK Driver for XC16x L:319 C:1 R/W
  
```



OCDS-Debugging via USB-JTAG-Wiggler-Box

<p>mouse position: (Project Window, Files): Target 1: click right mouse button click Options for Target 'Target 1'</p>	<p>or: click Options for Target</p>
	

Options for Target 'Target 1': Debug

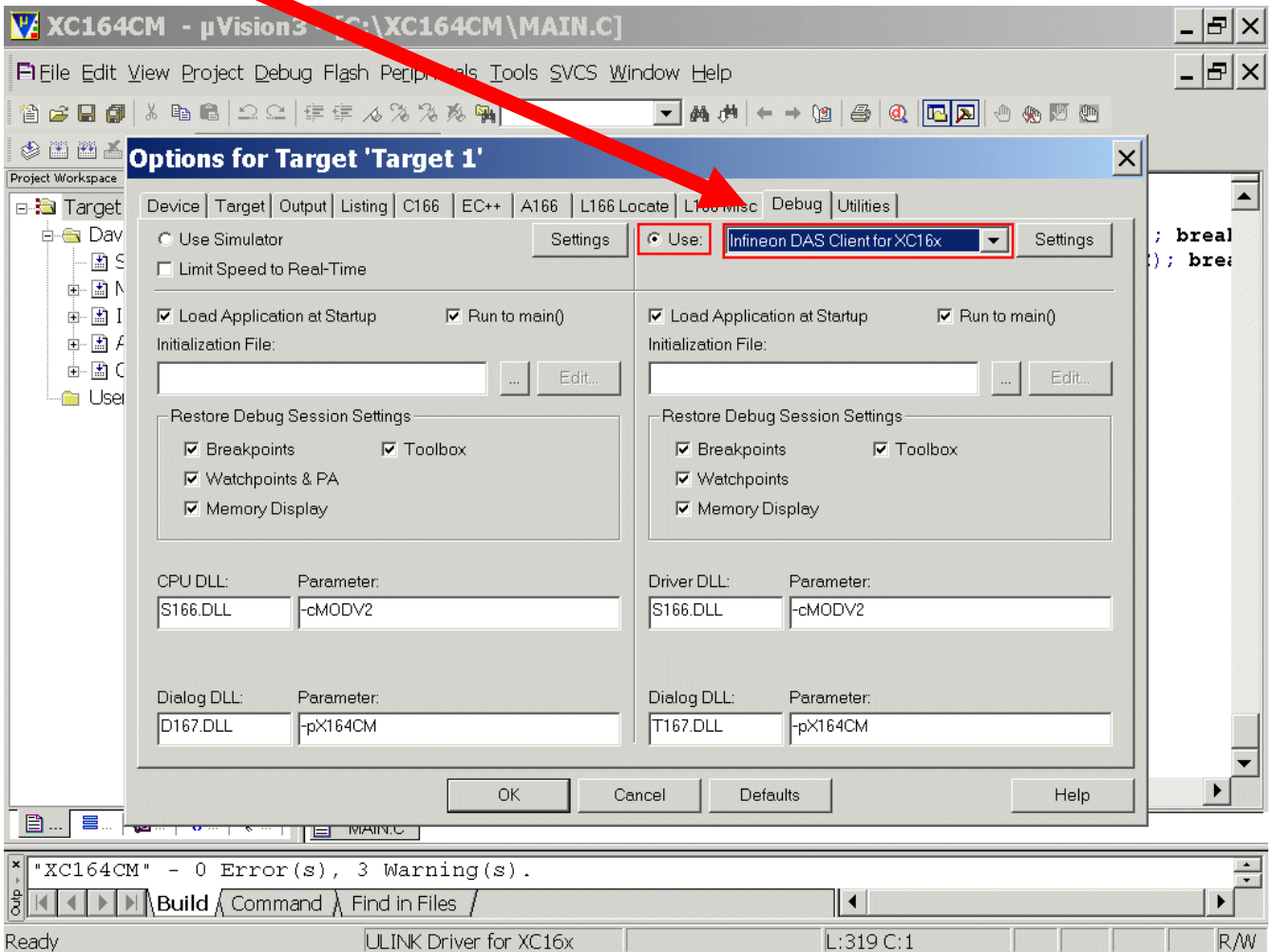


The dialog box 'Options for Target 'Target 1'' is shown with the 'Debug' tab selected. It contains two columns of settings:

- Left Column:**
 - Use Simulator (Settings)
 - Limit Speed to Real-Time
 - Load Application at Startup
 - Run to main()
 - Initialization File: [] [Edit...]
 - Restore Debug Session Settings:
 - Breakpoints
 - Watchpoints & PA
 - Memory Display
 - Toolbox
 - CPU DLL: S166.DLL | Parameter: -cMODV2
 - Dialog DLL: D167.DLL | Parameter: -pX164CM
- Right Column:**
 - Use: Keil Monitor-166 Driver (Settings)
 - Load Application at Startup
 - Run to main()
 - Initialization File: [] [Edit...]
 - Restore Debug Session Settings:
 - Breakpoints
 - Watchpoints
 - Memory Display
 - Toolbox
 - Driver DLL: S166.DLL | Parameter: -cMODV2
 - Dialog DLL: T167.DLL | Parameter: -pX164CM

Buttons at the bottom: OK, Cancel, Defaults, Help.

Options for Target 'Target 1': Debug
 click Use :
 select Infineon DAS Client for XC16x



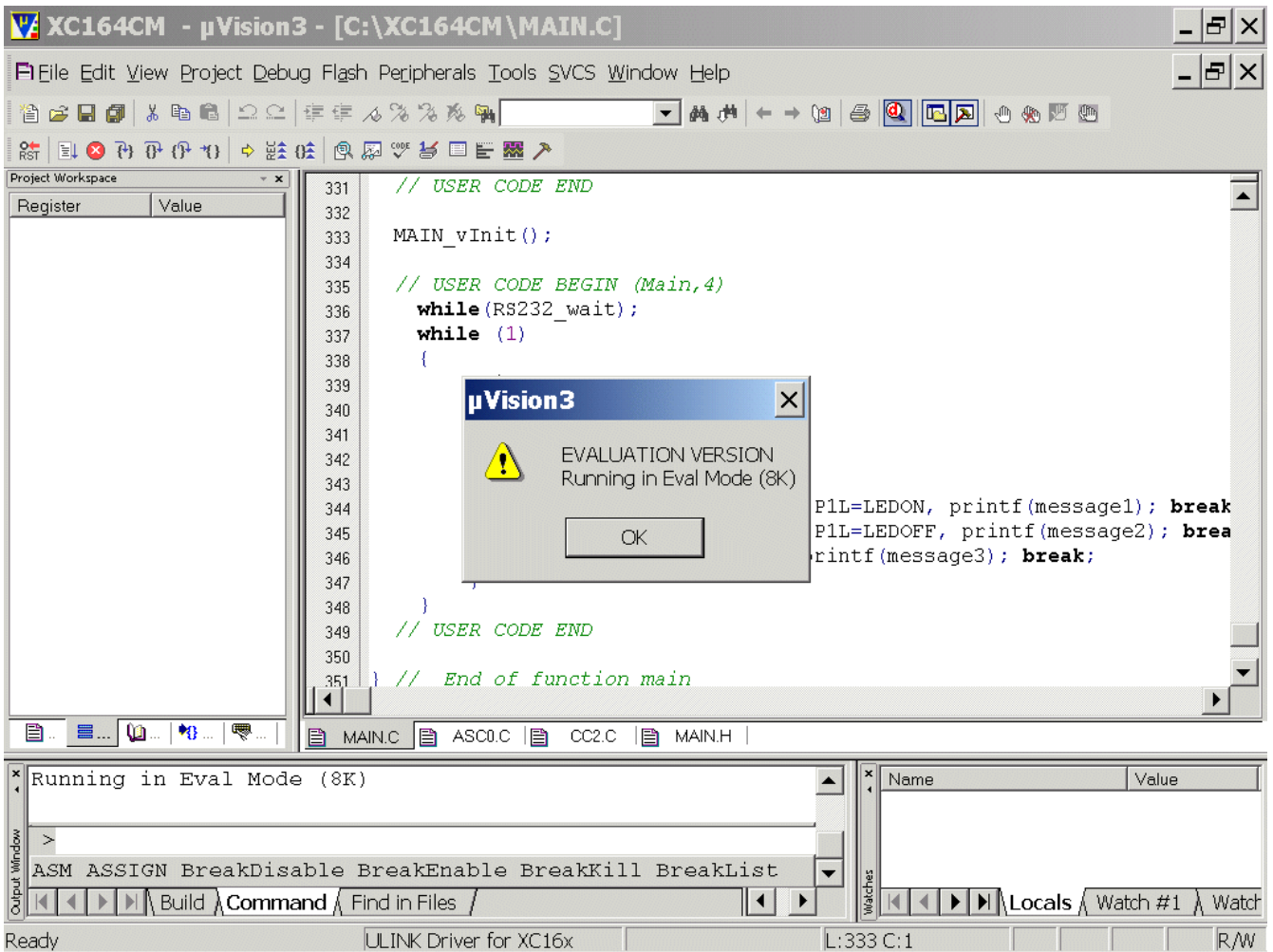
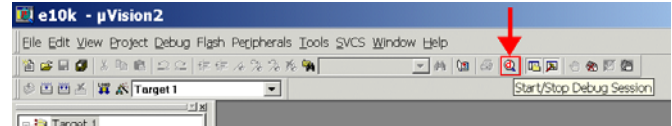
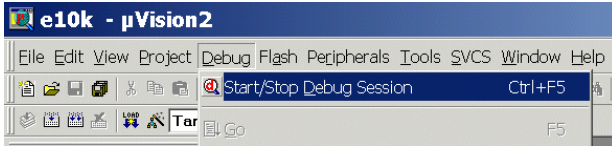
OK

Now, you can debug with OCDS via USB-JTAG-Wiggler-Box:

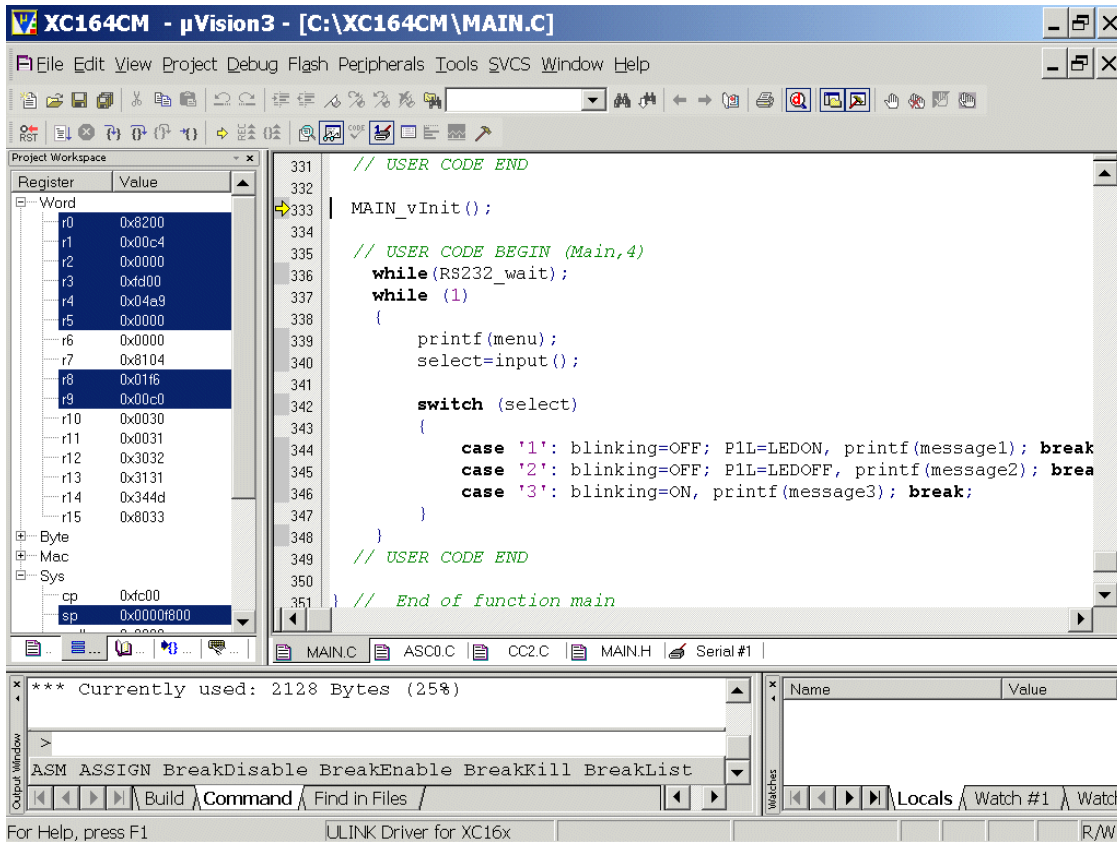
Debug – Start/Stop Debug Session

or

click 



OK



The screenshot shows the XC164CM IDE interface. The main window displays the source code for MAIN.C, which includes a `MAIN_vInit()` function. The code is as follows:

```

331 // USER CODE END
332
333 MAIN_vInit();
334
335 // USER CODE BEGIN (Main,4)
336 while (RS232_wait);
337 while (1)
338 {
339     printf(menu);
340     select=input();
341
342     switch (select)
343     {
344         case '1': blinking=OFF; P1L=LEDON, printf(message1); break
345         case '2': blinking=OFF; P1L=LEDOFF, printf(message2); brea
346         case '3': blinking=ON, printf(message3); break;
347     }
348 }
349 // USER CODE END
350
351 // End of function main

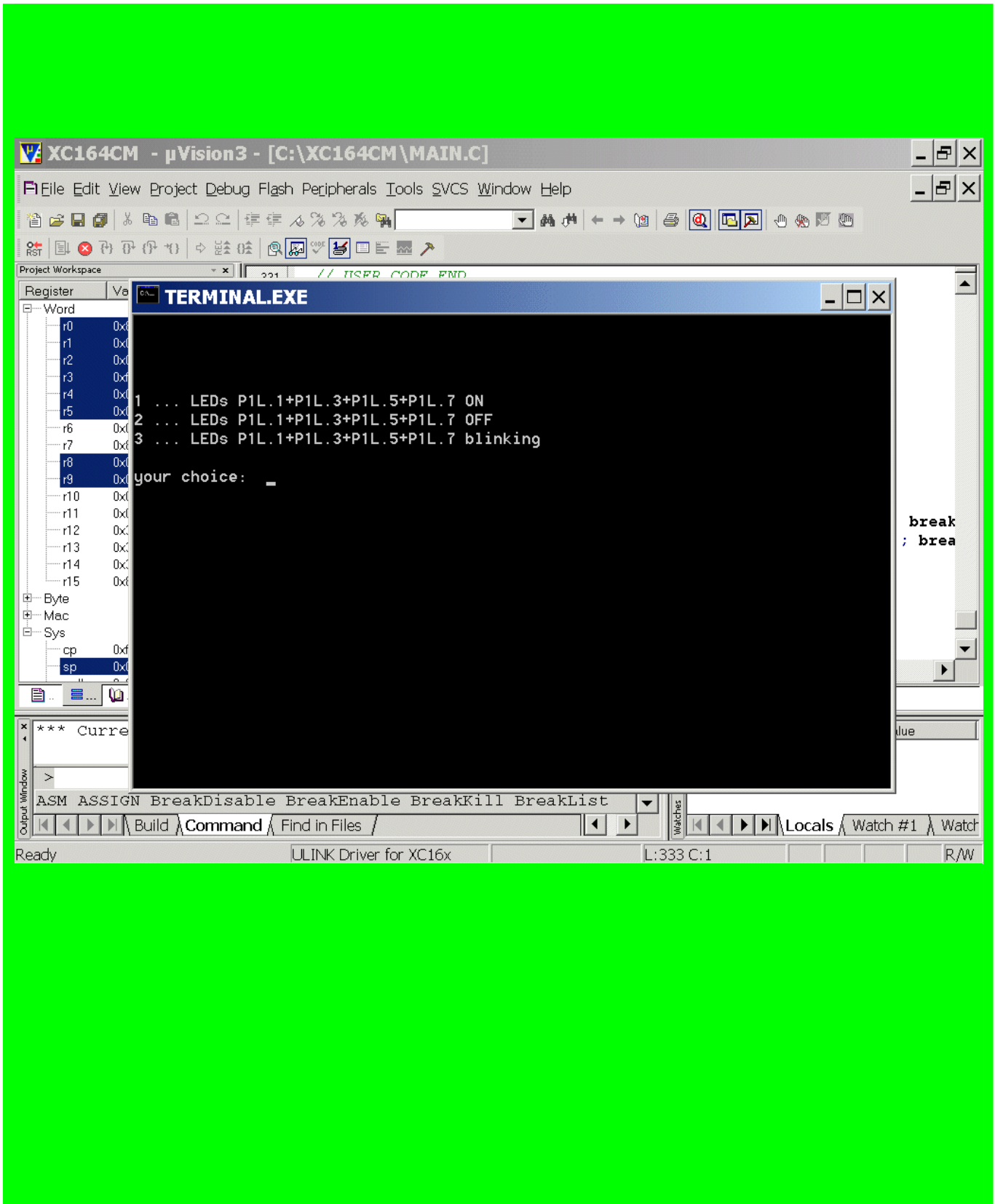
```

On the left side, the Register window shows the following values:


Register	Value
r0	0x8200
r1	0x00c4
r2	0x0000
r3	0xd00
r4	0x04e9
r5	0x0000
r6	0x0000
r7	0x8104
r8	0x0116
r9	0x00c0
r10	0x0030
r11	0x0031
r12	0x3032
r13	0x3131
r14	0x344d
r15	0x8033

At the bottom, the Output Window shows the message: "*** Currently used: 2128 Bytes (25%)". The Watch Window is currently empty.

Debug – Run



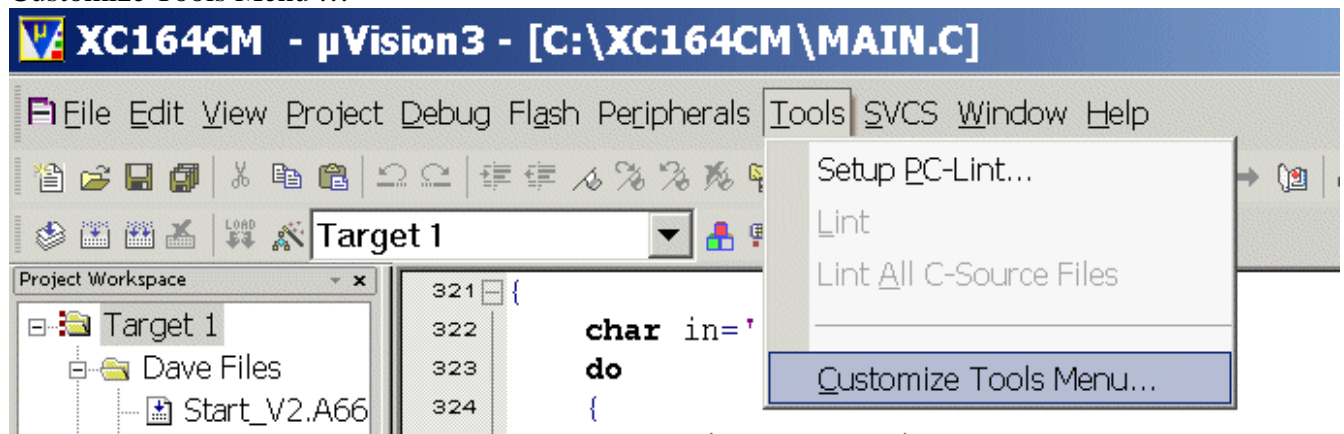
7.) Merging DAVe + μ Vision3

	Merging tools: DAvE + μVision3	http://www.infineon.com/DAvE
---	--	---

[open your current DAVe project:](#)

Tools

Customize Tools Menu ...



Menu Content:click **New (Insert)**

insert

Start DAVe with current Project '@P'

Command:

insert:

C:\Program Files\DAvE\dave.exe

Arguments:

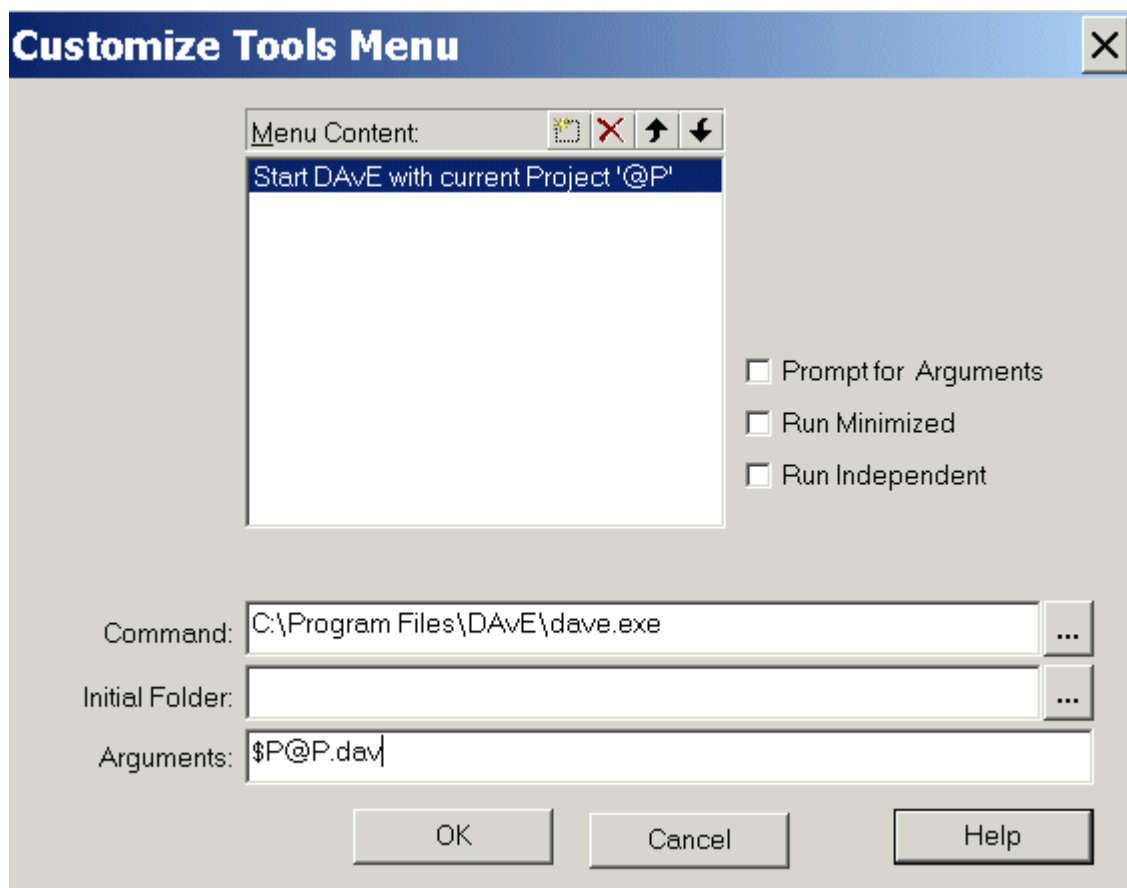
insert:

\$P@P.dav

Note:

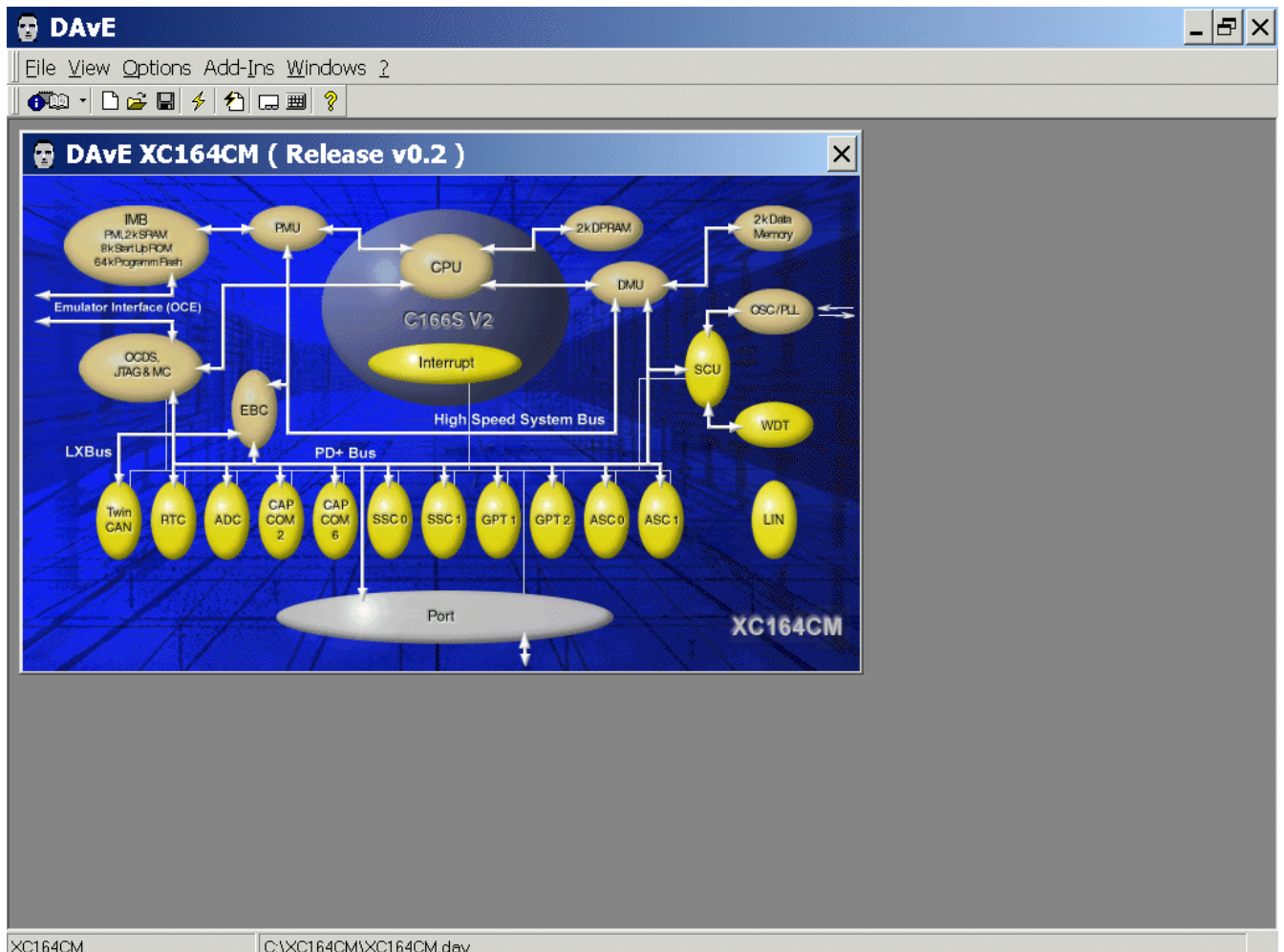
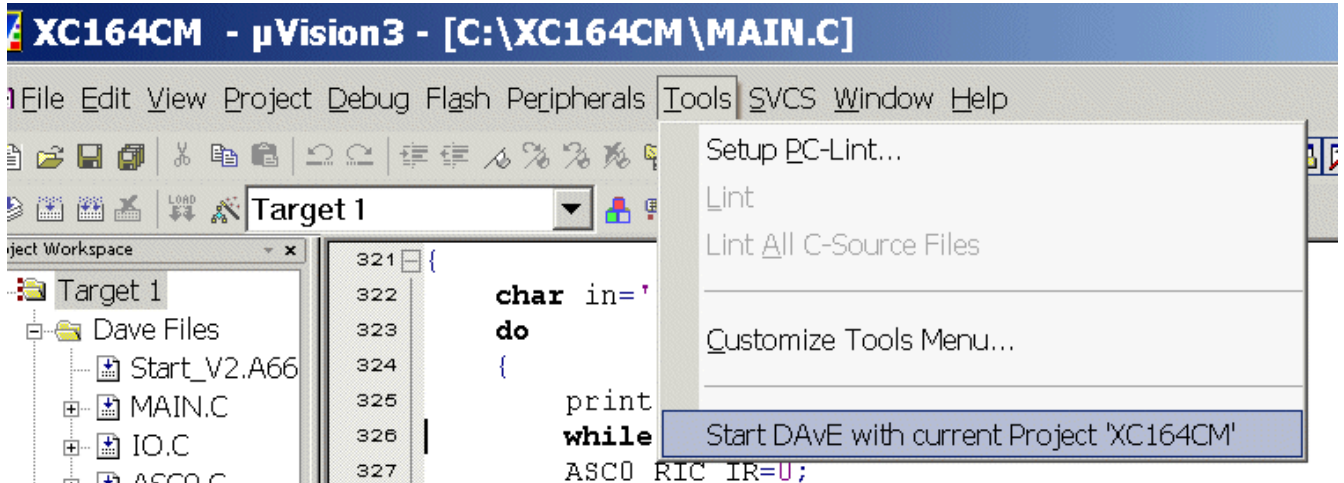
&P ... current directory (path)

@P ... project name without Extension and without path



OK

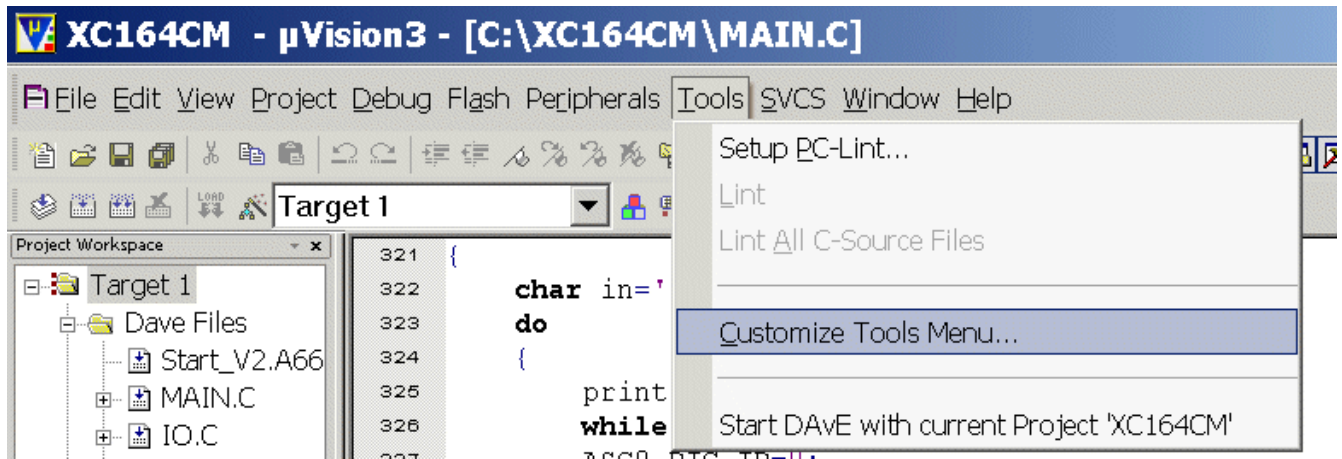
And see the result:



k10.2.) open DAvE without project:

Tools

Customize Tools Menu ...



Menu Content:

click New (Insert)

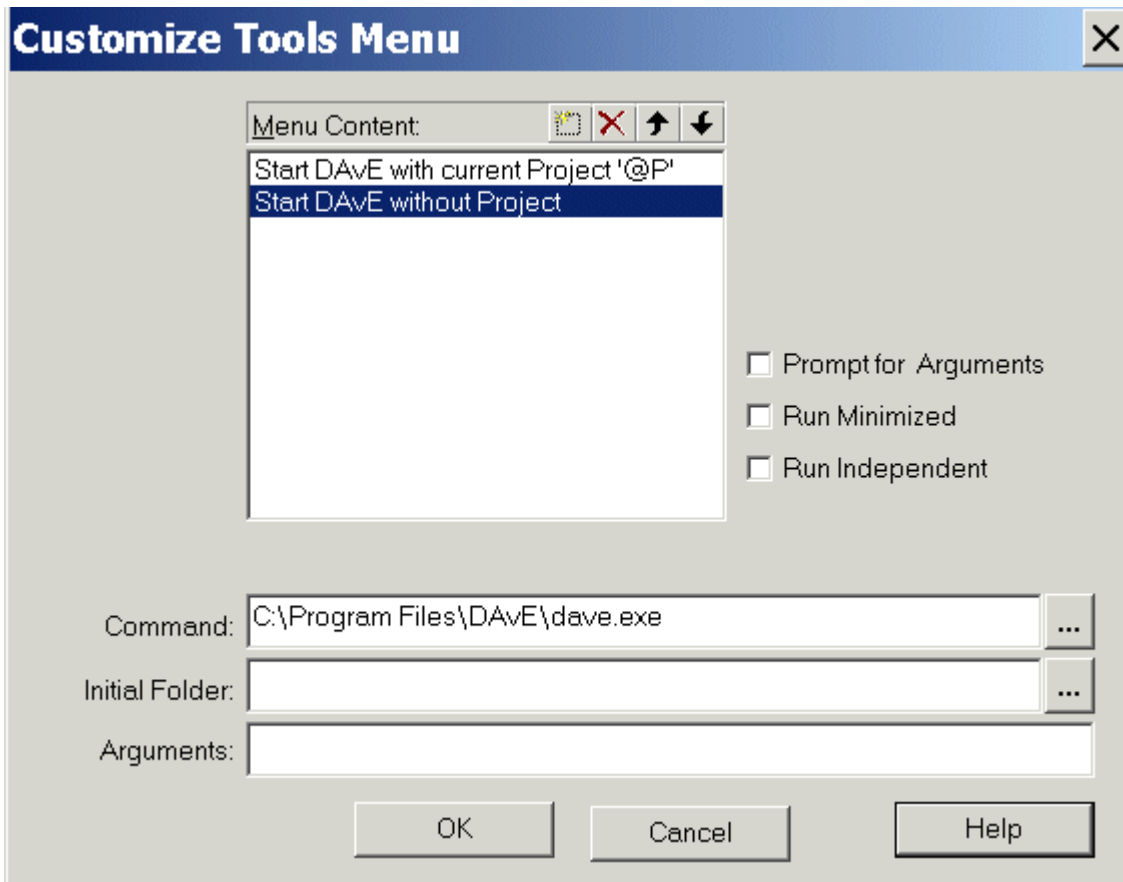
insert

Start DAVe without Project

Command:

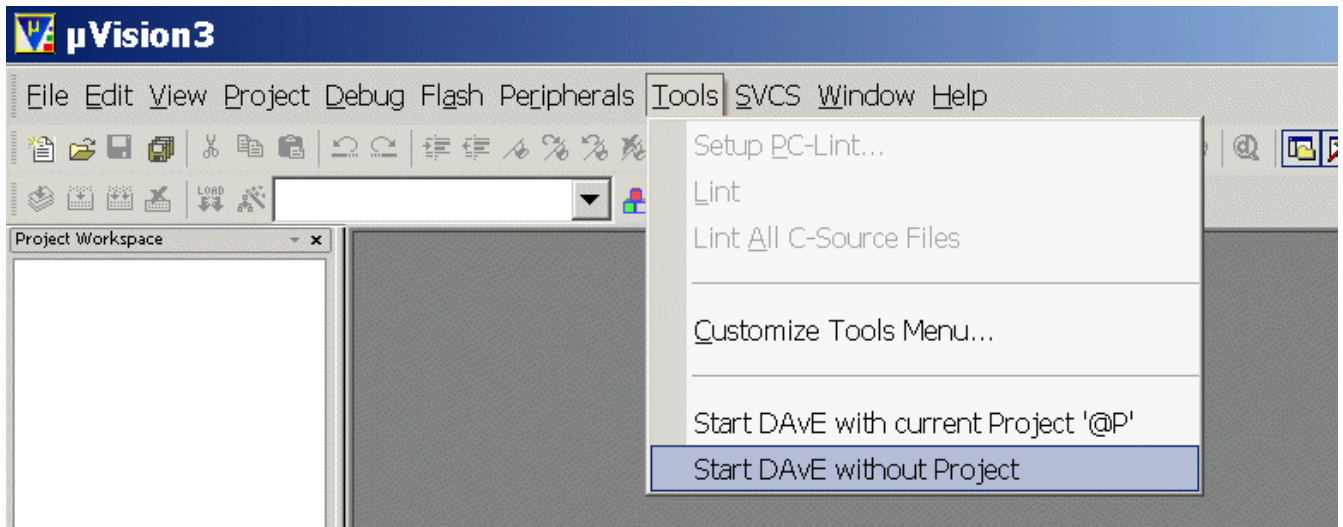
insert:

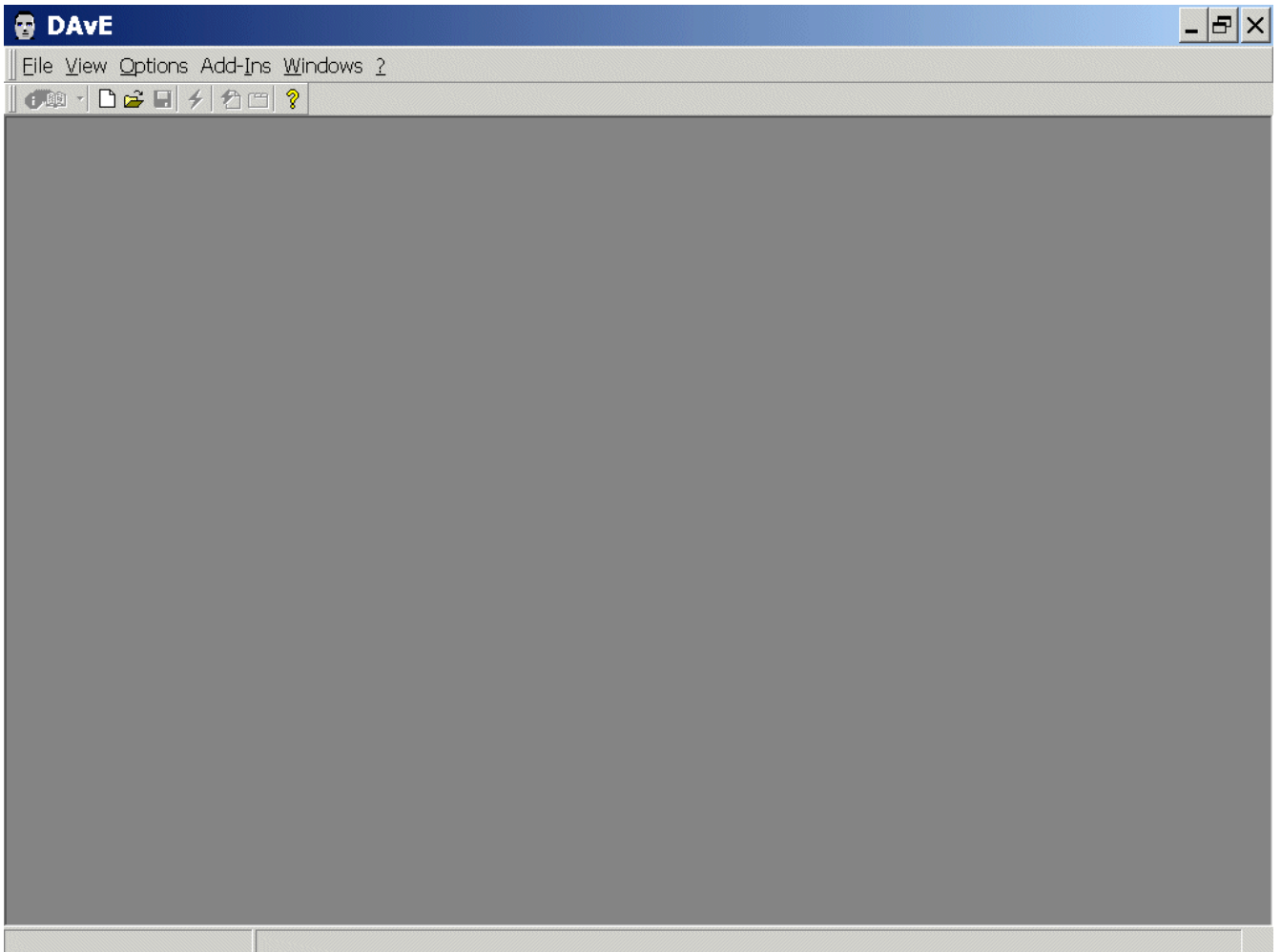
C:\Program Files\DAvE\dave.exe



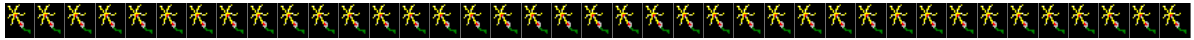
OK

And see the result:





8.) Feedback (XC164CM): Your opinion, suggestions and/or criticisms



Contact Details (this section may remain empty should you wish to offer feedback anonymously):

If you have any suggestions please send this sheet back to:

email: mcdocu.comments@infineon.com

FAX: +43 (0) 4242 3020 5783



Your suggestions:

<http://www.infineon.com>