AP16174

XE164

UConnect-CAN XE164 "Cookery Book" for a hello world application

Using DAvE (Code Generator)

Using the KEIL tool chain µVision 4 (IDE, Compiler, Utility Tools)

Microcontrollers

Never stop thinking

Edition 2010-04-22

Published by
Infineon Technologies AG
81726 München, Germany

© Infineon Technologies AG 2010.
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.

AP08048		
Revision History:	2010-04	V2.0
Previous Version:	none	

Page	Subjects (major changes since last revision)

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Table of Contents **Page**

Note: Table of Contents see page 9.

Introduction:

This “Appnote” is a Hands-On-Training / Cookery Book / step-by-step book.
It will help inexperienced users to get an UConnect-CAN XE164 up and running.

With this 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 suit your needs.
You can connect either a part of - or your entire application to the UConnect-CAN XE164.
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.

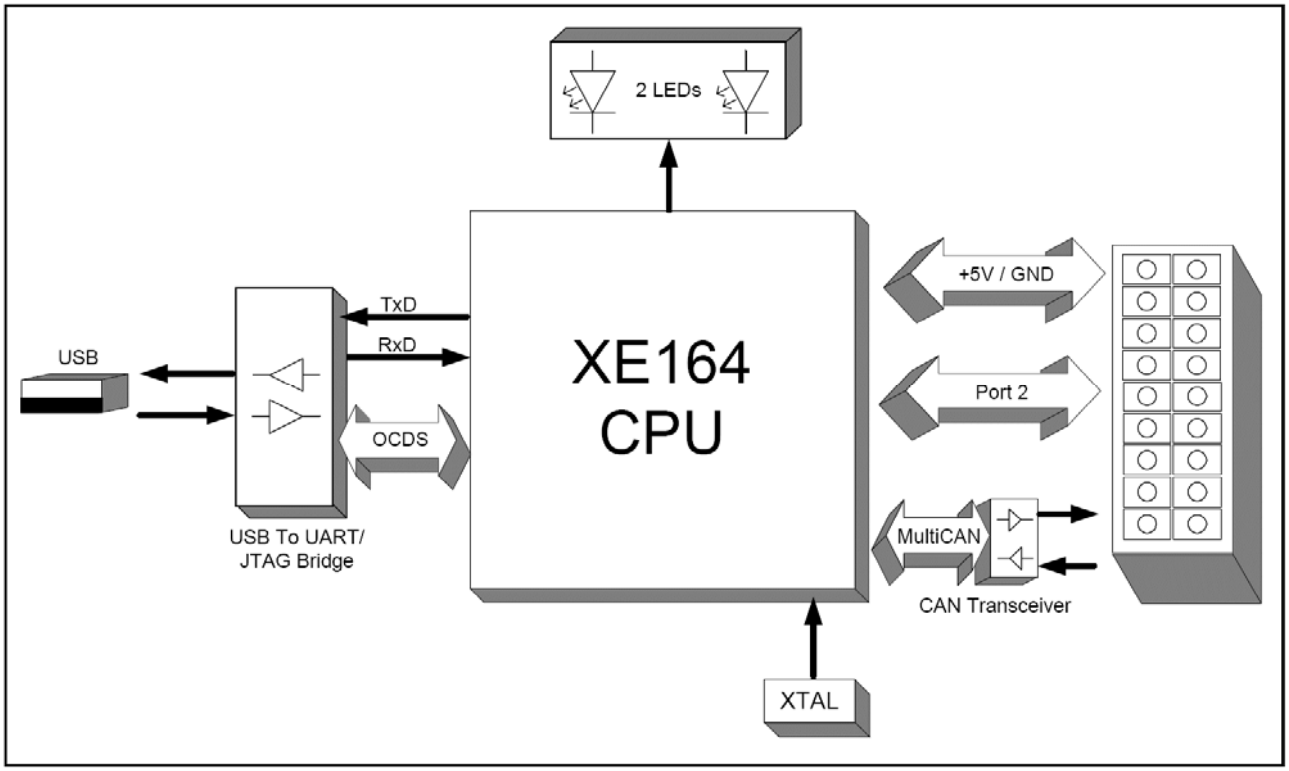
Note:
The style used in this document focuses on working through this material as fast and easily as possible. That means there are full screenshots instead of dialog-window-screenshots; extensive use of colours and page breaks; and listed source-code is not formatted to ease copy & paste.

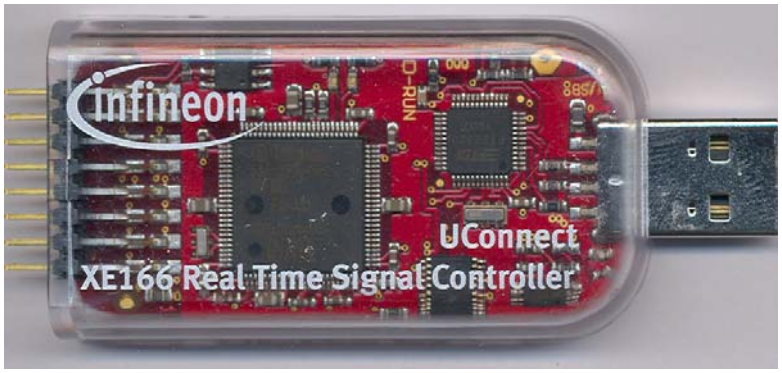
Have fun and enjoy the UConnect-CAN XE164!

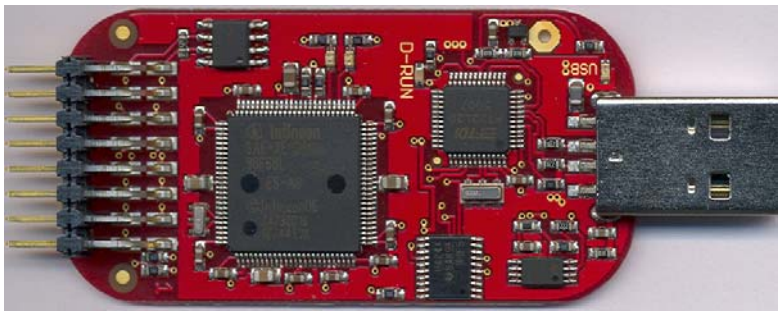
Programming Example

UConnect-CAN XE164

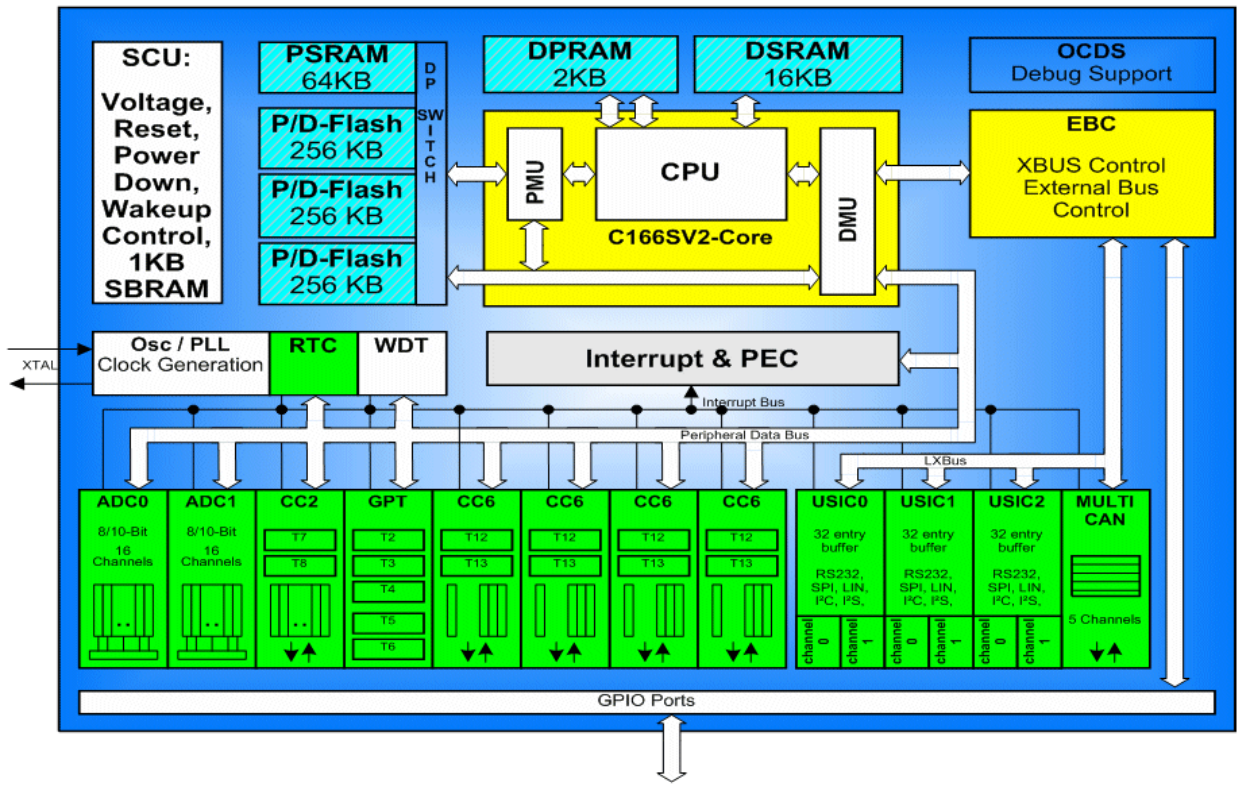
Block Diagram (Source: XE164 UConnect Manual)



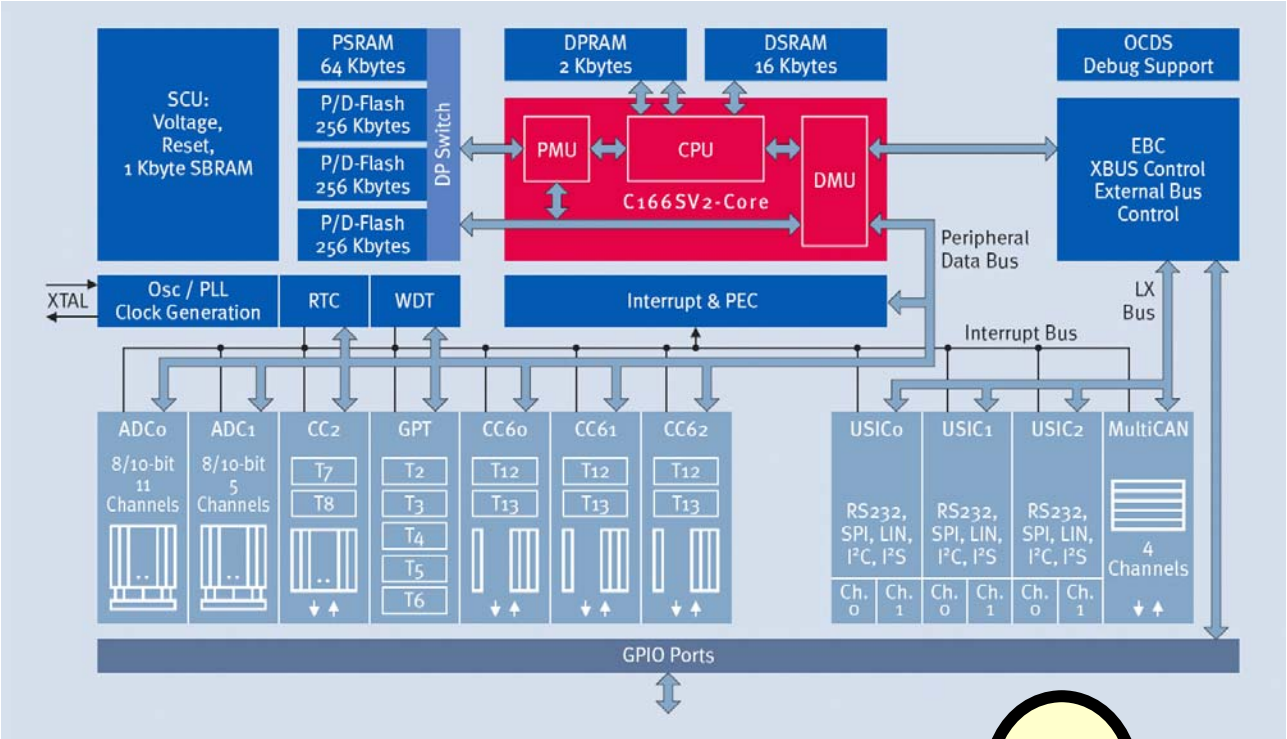




SAF-XE167F-96F66L Block Diagram (Source: Product Marketing)

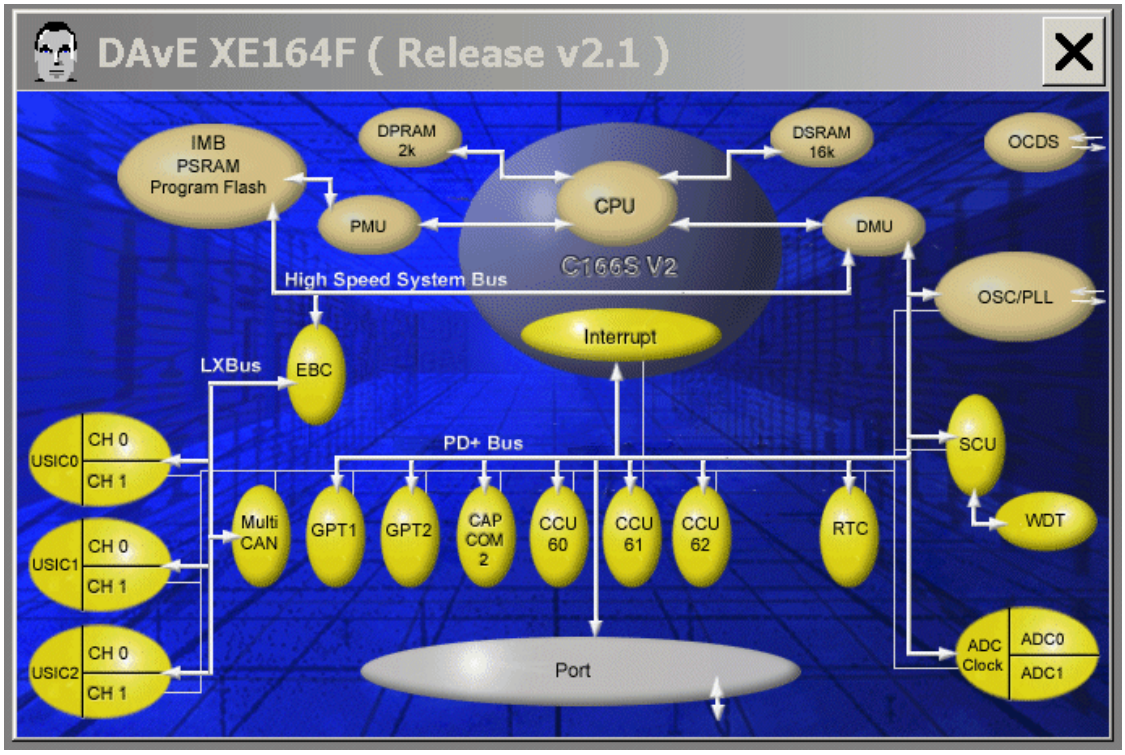


SAF-XE164F-96F66L Block Diagram (Source: Product Brief)

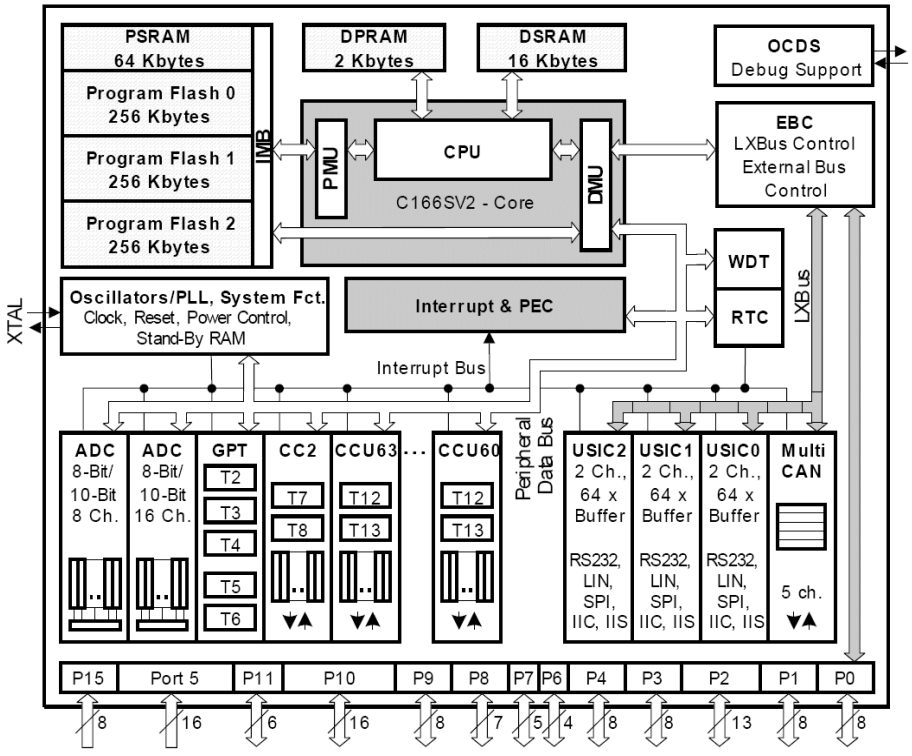


Note:
The XE164 microcontroller is a derivative of the XE167 microcontroller!

SAF-XE164F-96F66L Block Diagram (Source: DAvE)

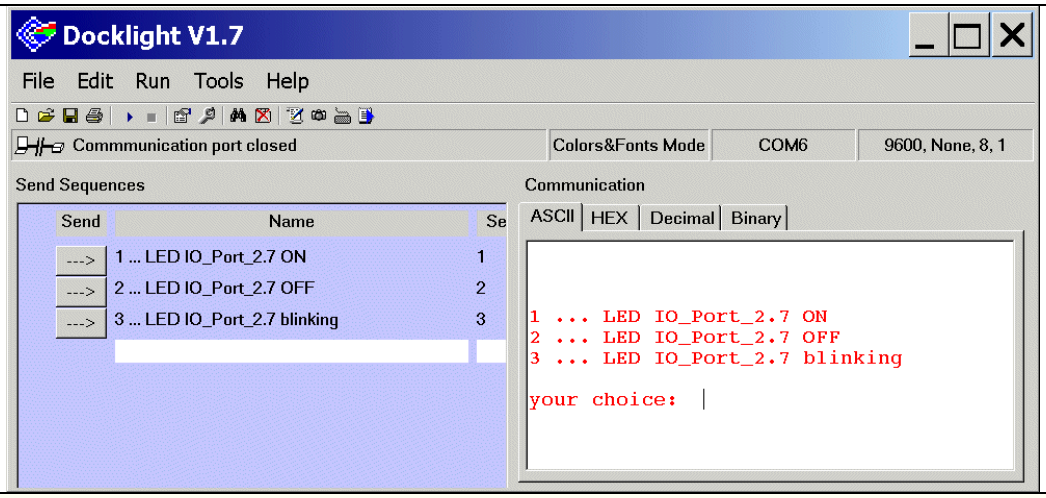


XE16x Block Diagram (Source: User's Manual)



Note:
Just by comparing the different sources of block diagrams, you should be able to get a complete picture of the microcontroller and to answer some of your initial questions.

“Cookery book“
For your first programming example for the UConnect-CAN XE164:

Your program:	
Chapter/ Step	***** Recipes *****
1.)	**DAS Installation + Connecting the UConnect-CAN XE164**
2.)	**DAvE (program generator)** **DAvE Installation (mothersystem) + DAvE Update Installation (XE16xx_Series.dip) for XE164**
3.)	**Using DAvE** **Microcontroller initialization for your programming example**
4.)	**Using the KEIL Development Tools (µVision4; C166 Version 7.00)** **Programming of your application (hello world) with the KEIL tool chain (µVision4)**
5.)	**Running your first programming example**

Feedback

6.)	**Feedback**

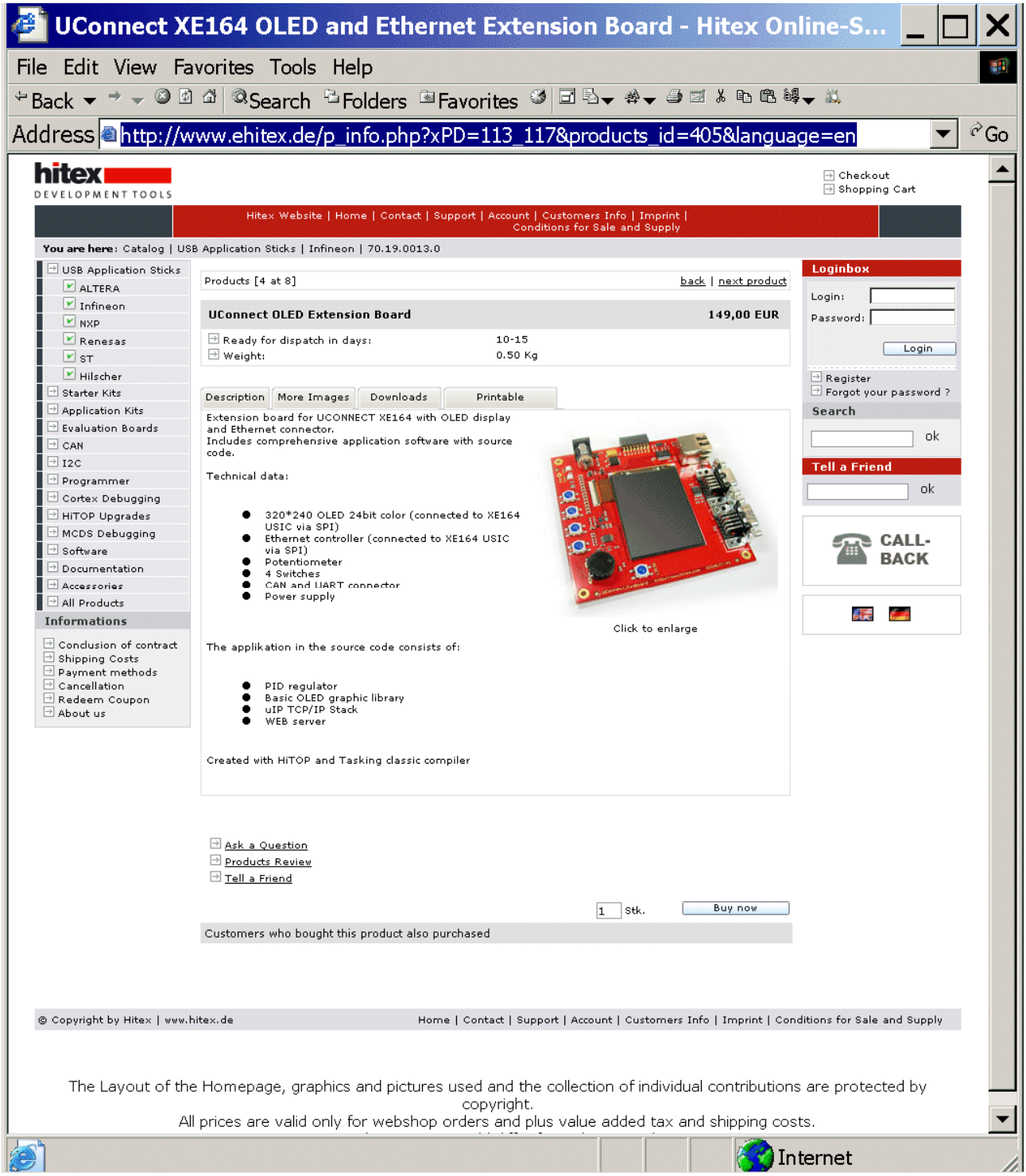
1.) DAS Installation + Connecting the UConnect-CAN XE164:

Screenshot of the UConnect-CAN XE164 Homepage:

http://www.infineon.com/cms/en/product/channel.html?channel=db3a3043183a955501183c3446e1006f

Note:
For further information, please refer to the XE164 UConnect Manual, V.1.0 .
For further information, please refer to the XE164 UConnect Manual, V.1.1 .

Note:
For the UConnect-CAN XE164 there is an additional extension board UConnect OLED Extension with
320*240 Organic Light Emitting Diode (OLED) 24bit color (Basic OLED graphic library),
and Ethernet controller (uIP TCP/IP Stack and WEB server) available at
http://www.ehitex.de/p_info.php?products_id=405
http://www.ehitex.de/p_info.php?xPD=113_117&products_id=405&language=en.

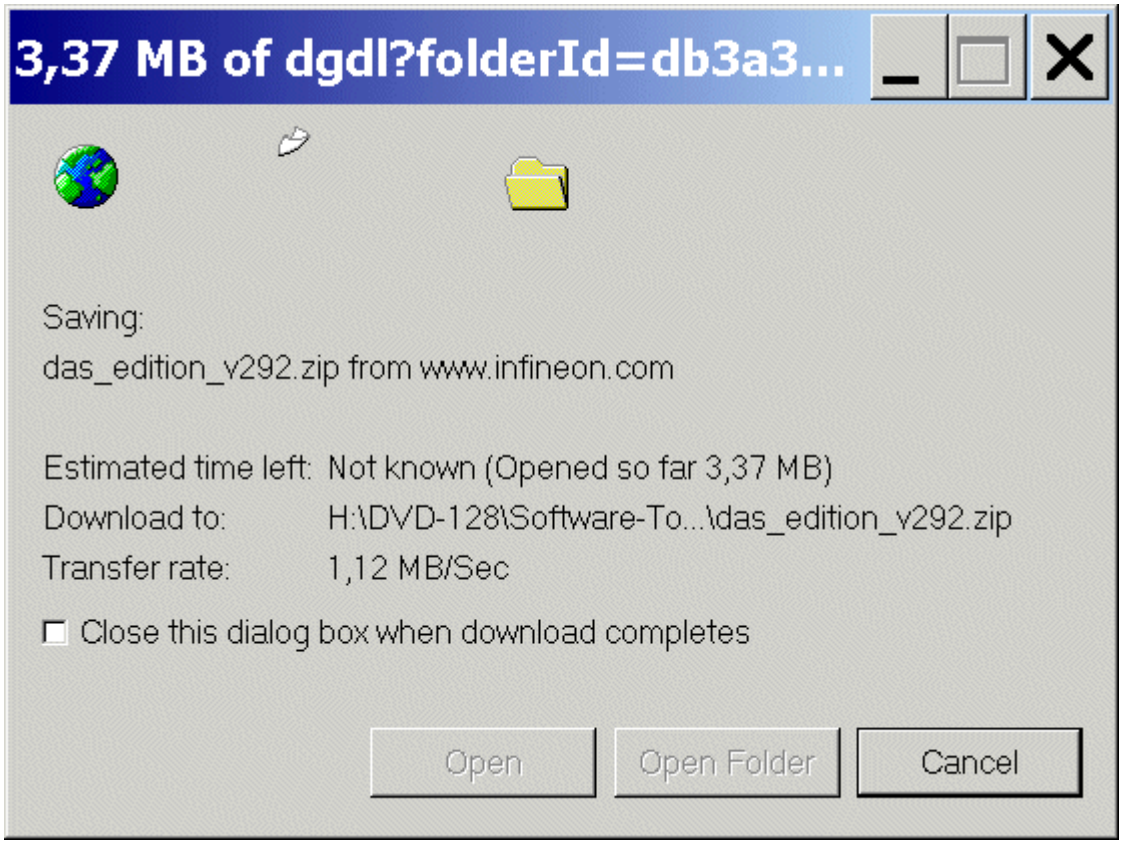


Install the Infineon DAS (D e v i c e A c c e s s S e r v e r) Server:

Go to www.infineon.com/DAS:

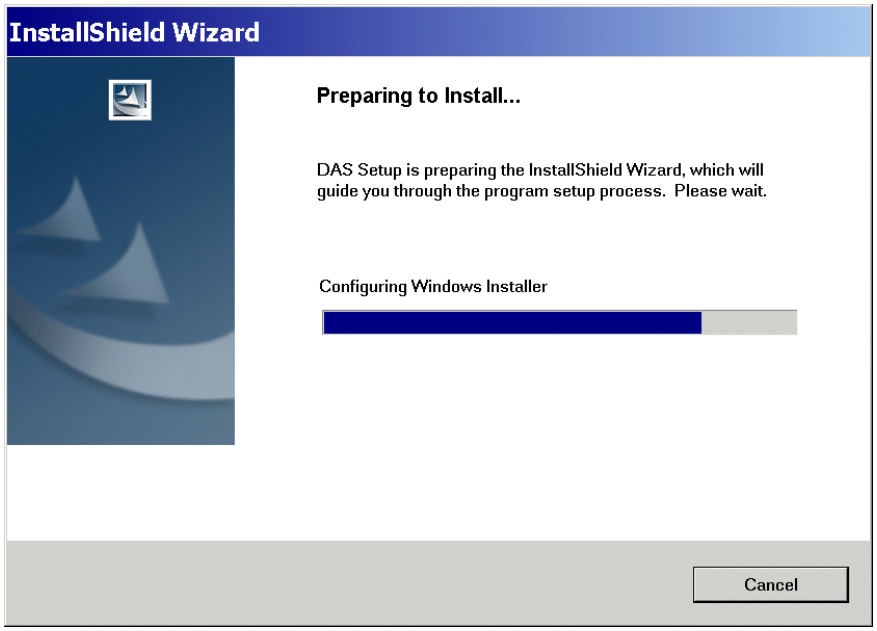
Note:
The DAS Server must be installed on your host computer!
The goal of the DAS software is to provide one single interface for all types of tools (e.g. Programmers/Flash Tools, Emulators/Debugger Development Systems).
The USB Device driver communicates with the UConnect-CAN XE164 when connected to the host computer.
The USB Device driver for the UConnect-CAN XE164 USB interface is included in the DAS software.
A virtual COM port driver is also included.

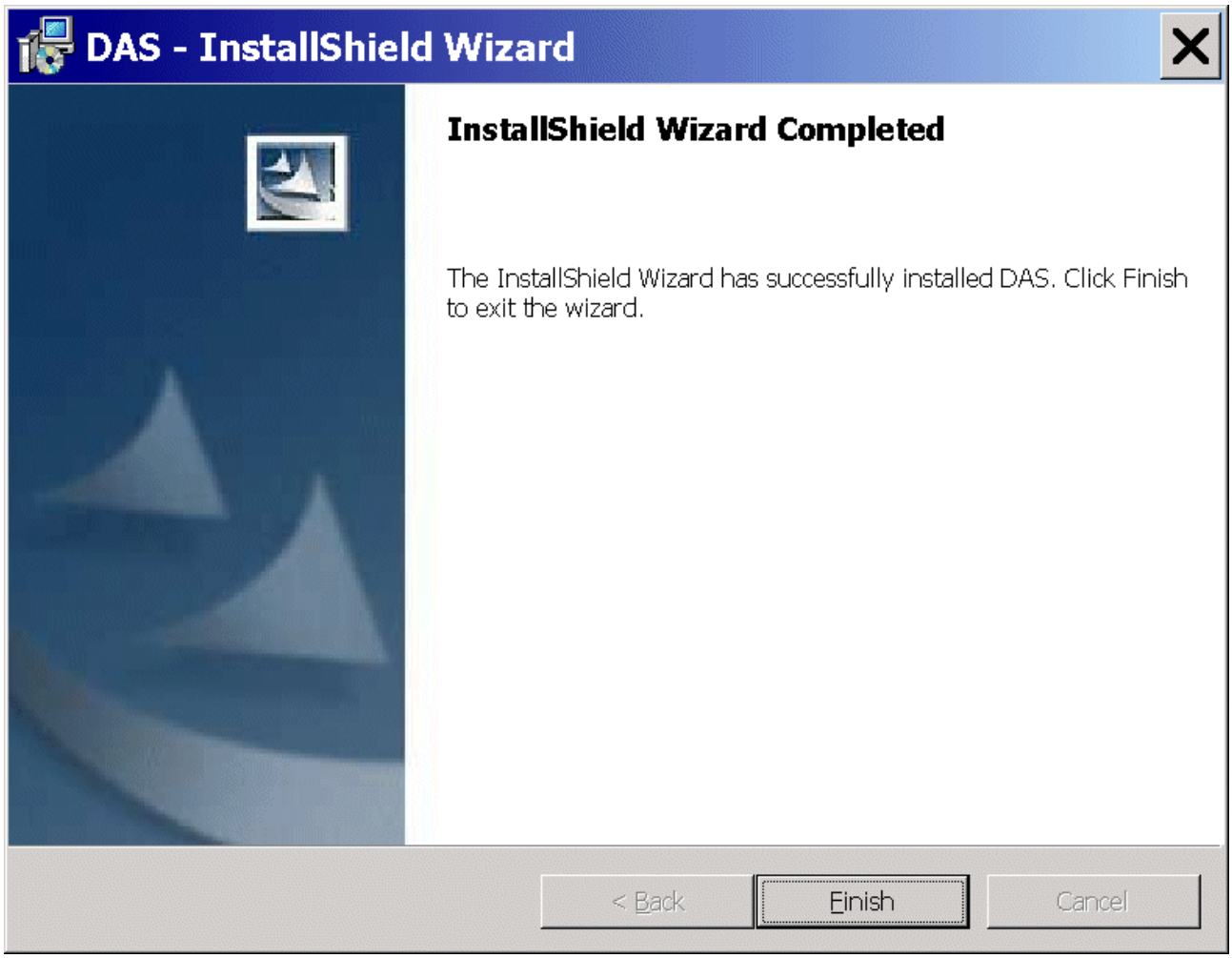
Download “The latest release version for download: DAS Edition v2.9.2”
(- or any higher version !!!):



Unzip das_edition_v292.zip and

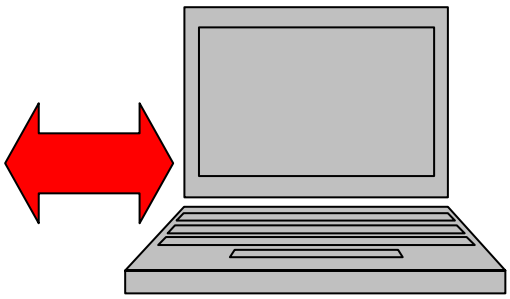
execute “DAS_v292_setup.exe” to install the DAS Server.

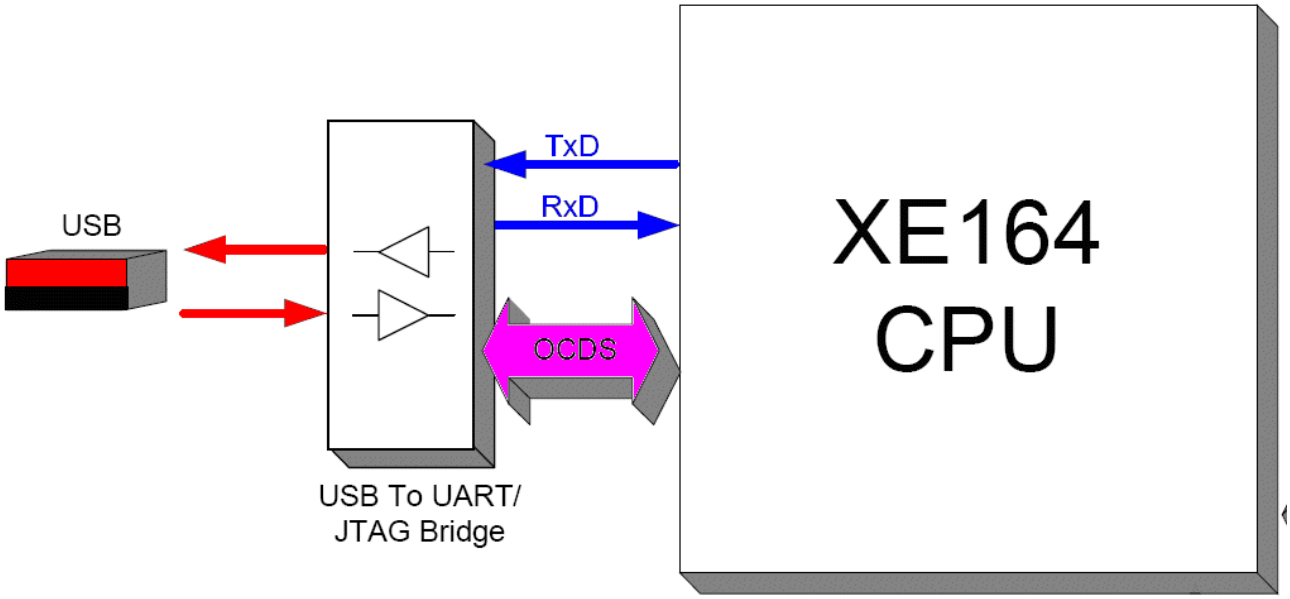




Click Finish

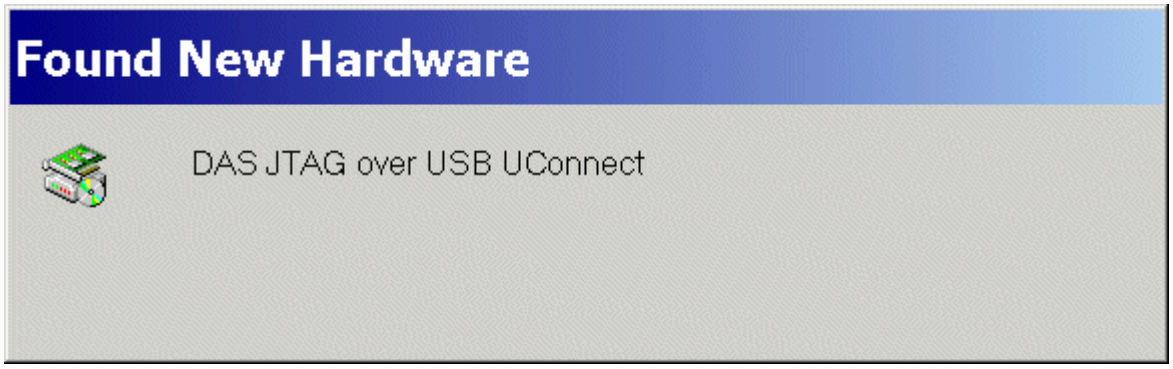
Connect the UConnect-CAN XE164 to the host computer:





USB Connection:

.) used for: UART communication (the USIC0_CH0/UART/RS232/serial interface is available via USB as a virtual COM port of the second USB channel of the FTDI FT2232 Dual USB to UART/JTAG interface).
.) used for: On-Chip-Flash-Programming and Debugging (first USB channel of the FTDI FT2232 Dual USB to UART/JTAG interface).
.) the USB connection works also as the power supply.

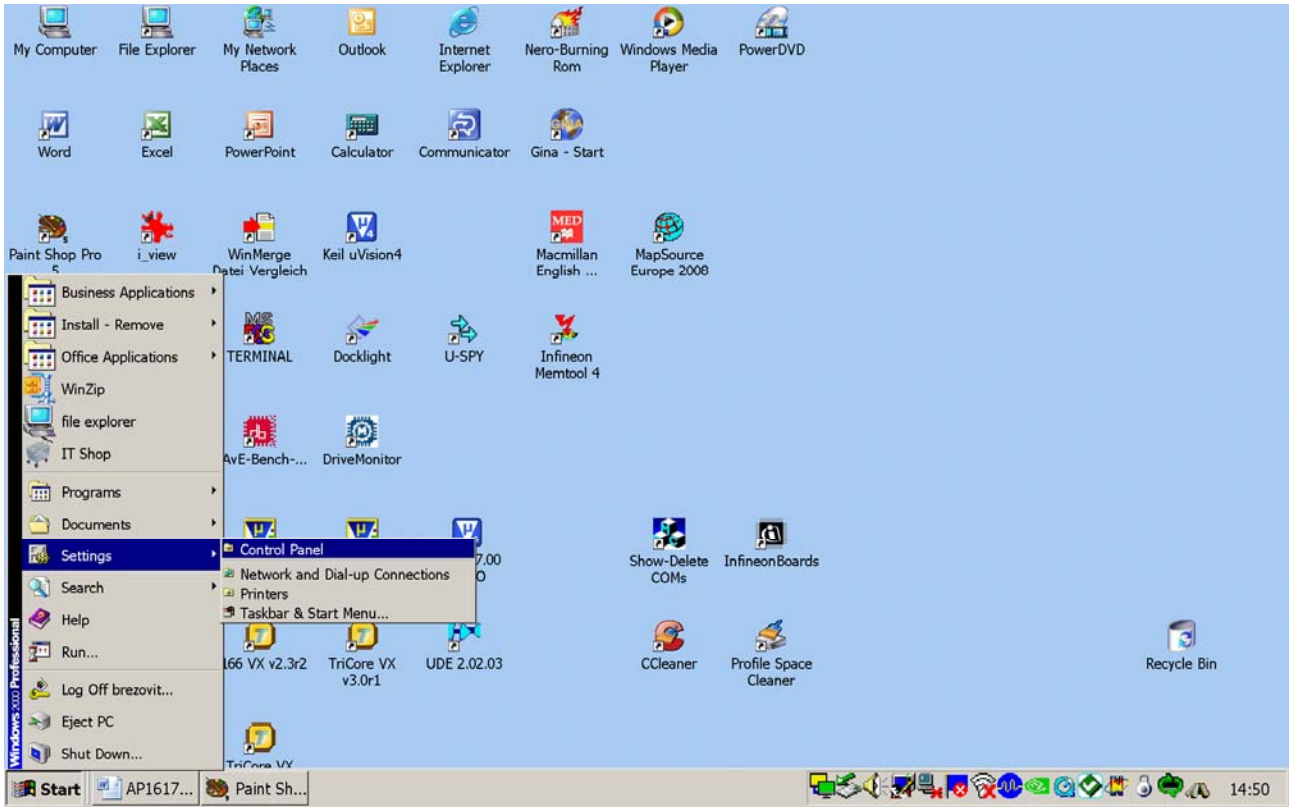


Note:
A USB driver is installed the first time while connecting the UConnect-CAN XE164 via USB to your host computer.

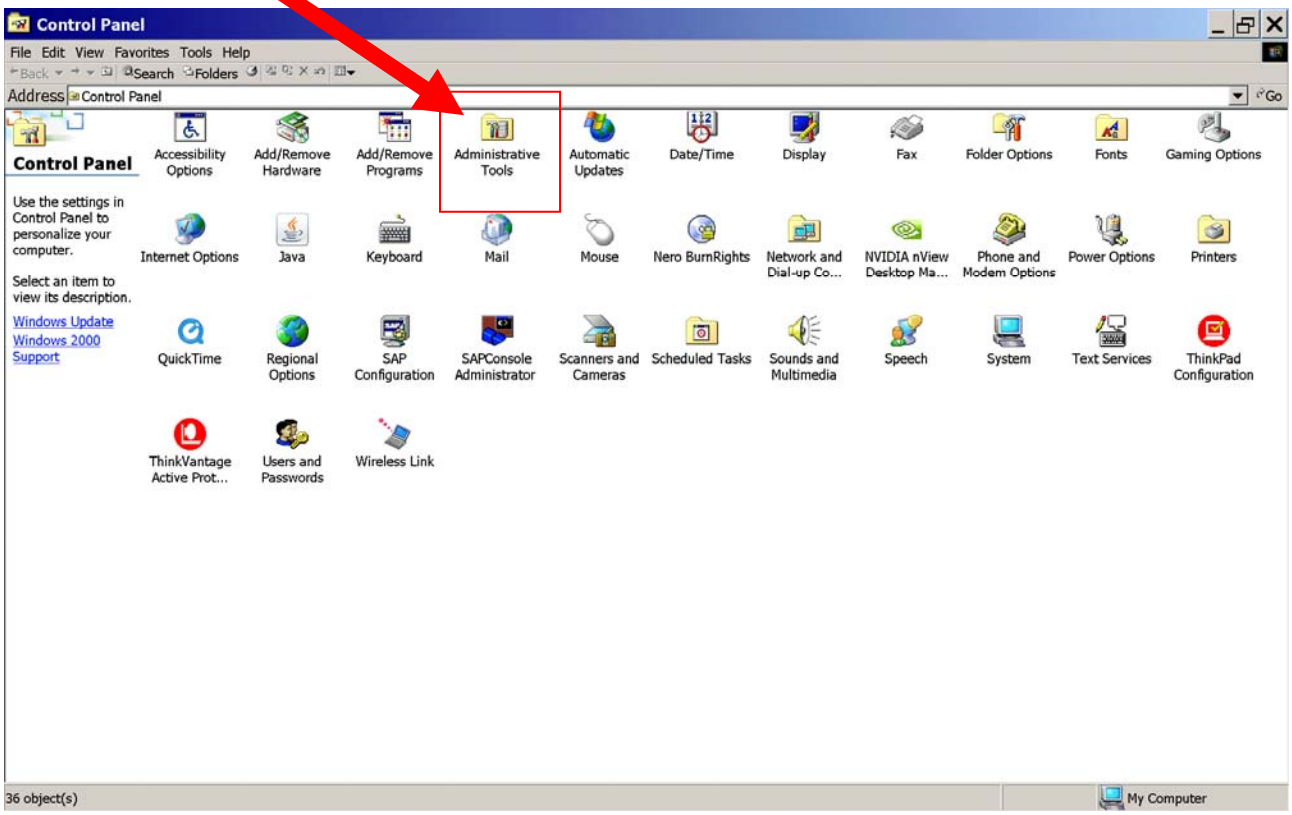
Note:
A default virtual COM Port is generated.

Using a Windows 2000 operating system, we are now going to search for the virtual COM Port which was generated after connecting our UConnect-CAN XE164:

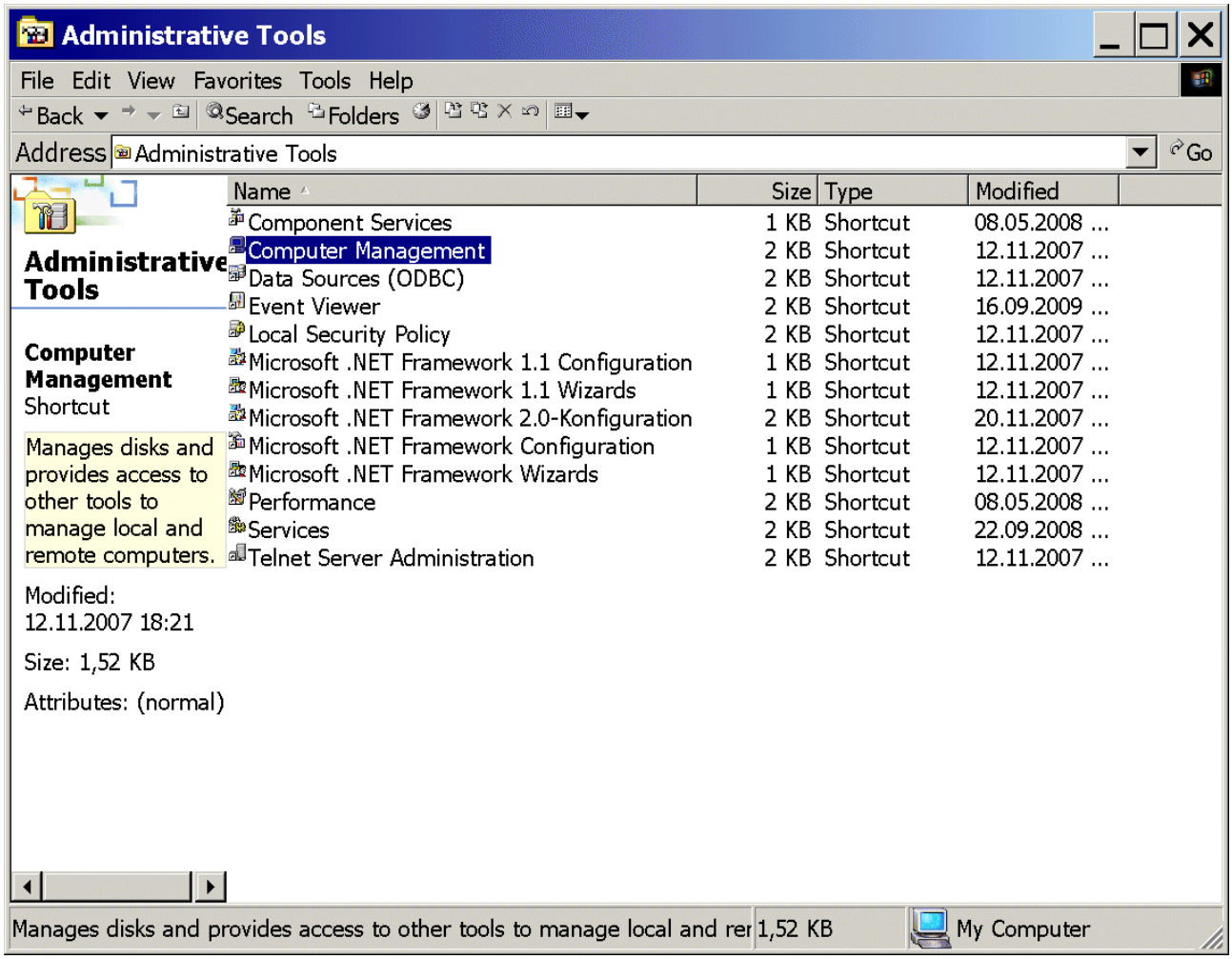
Start – Settings – Control Panel



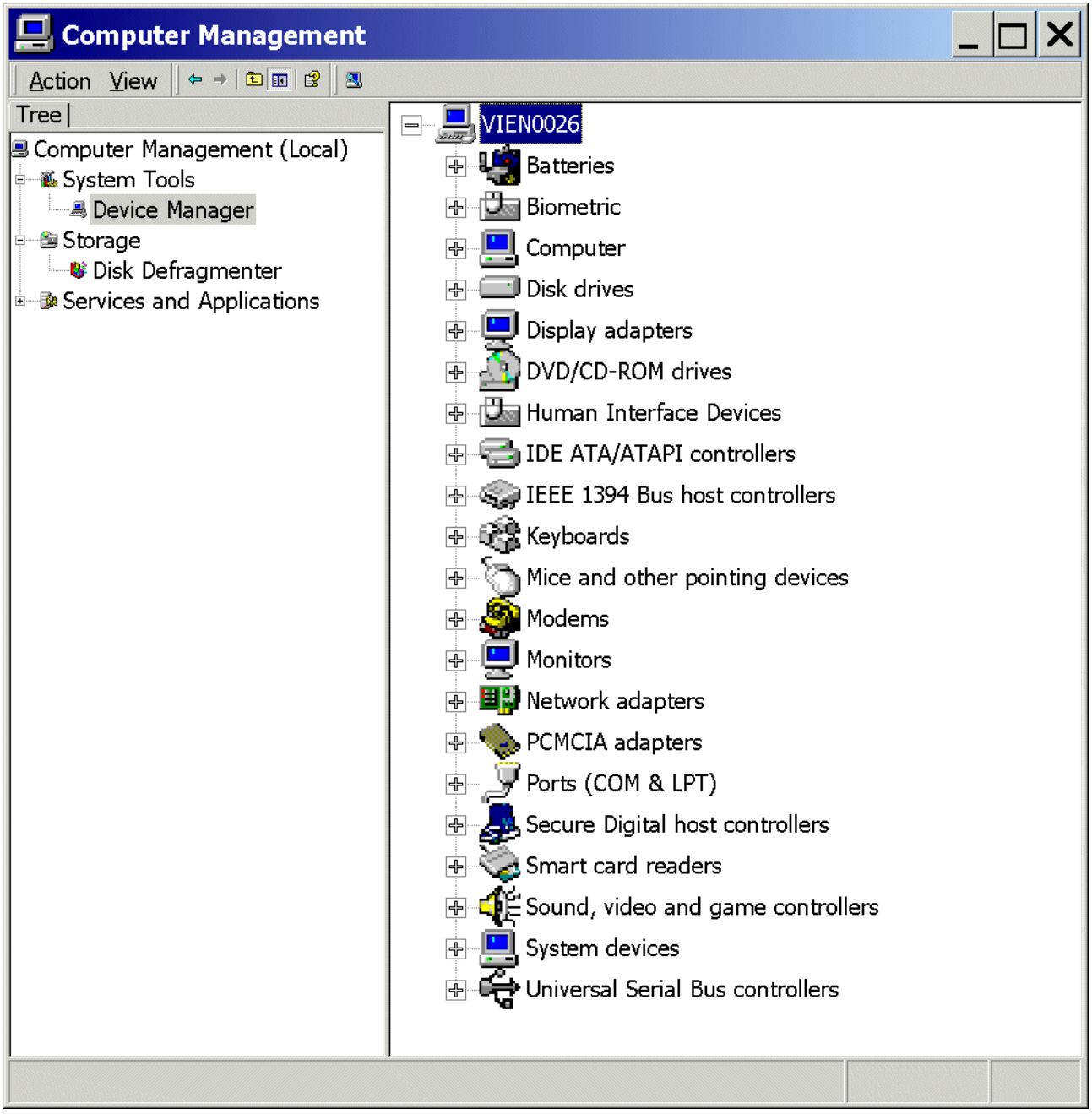
Double click: Administrative Tools



Double click: Computer Management

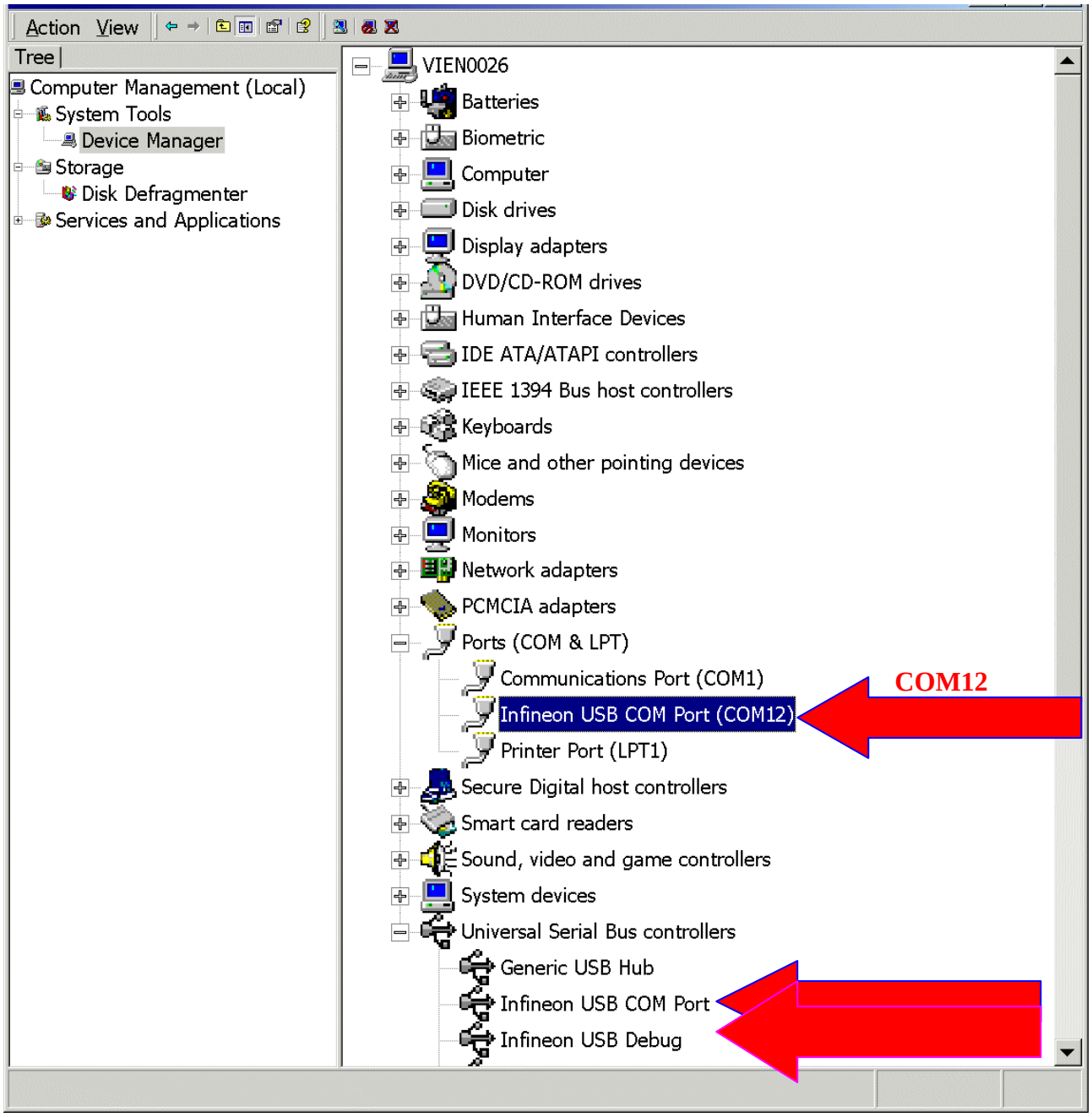


Click: Device Manager



Expand: Ports (COM & LPT):
Expand: Universal Serial Bus controllers:

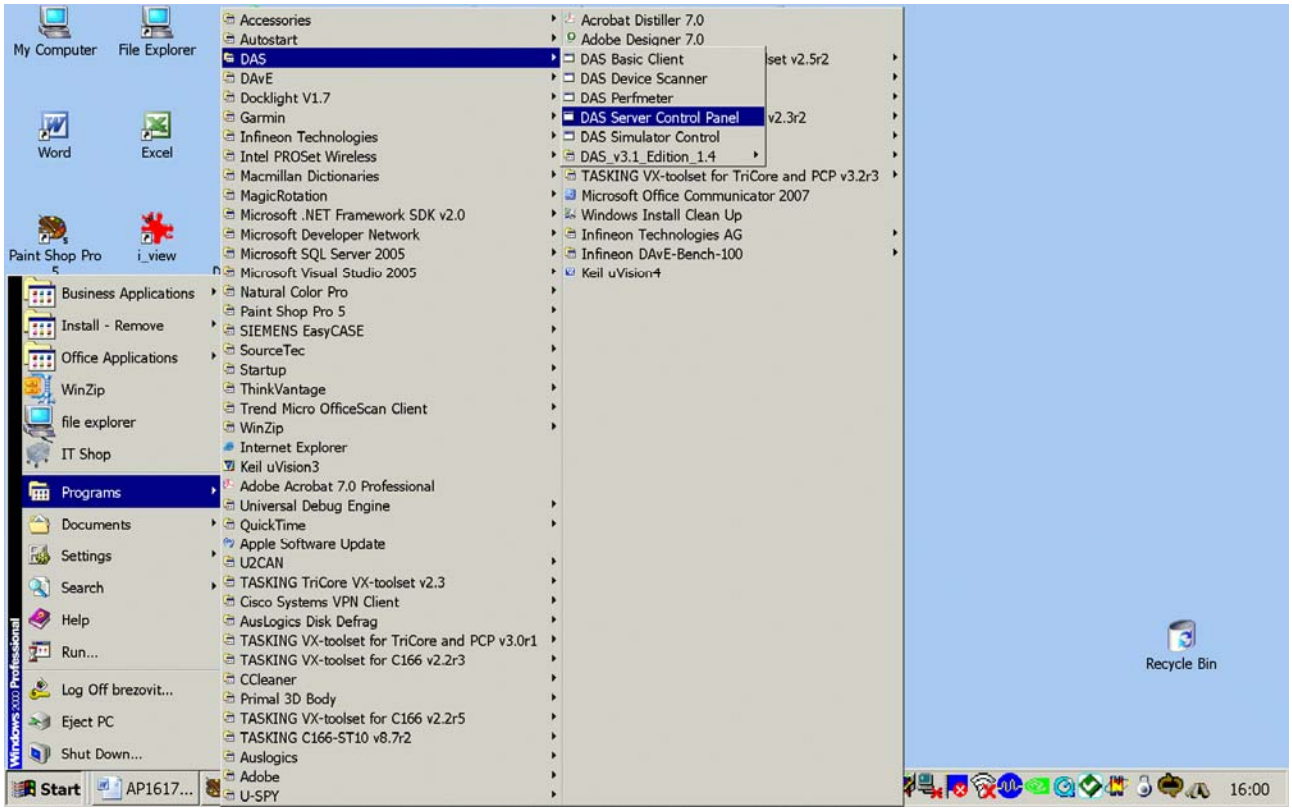
Note:
As we can see:
our virtual COM Port for UART/RS232 communication with the UConnect-CAN XE164 via USB is COM12!

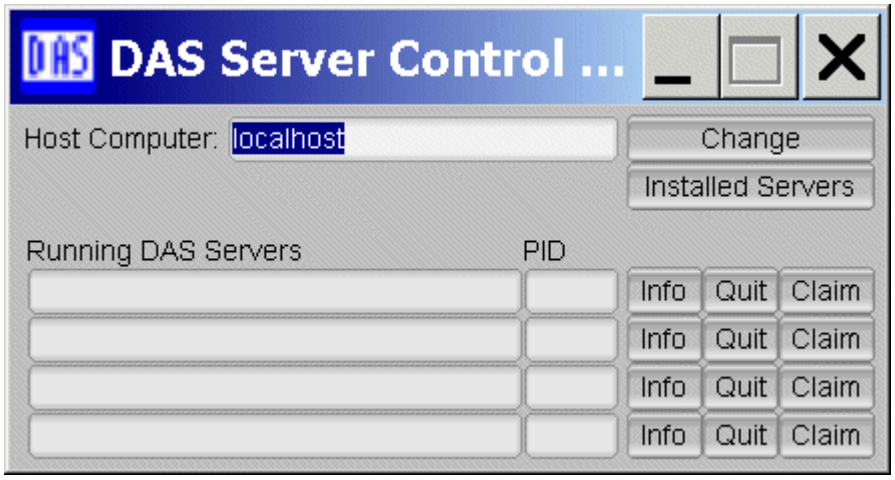


(U)DAS (Universal D e v i c e A c c e s s S e r v e r) Server hint 1:

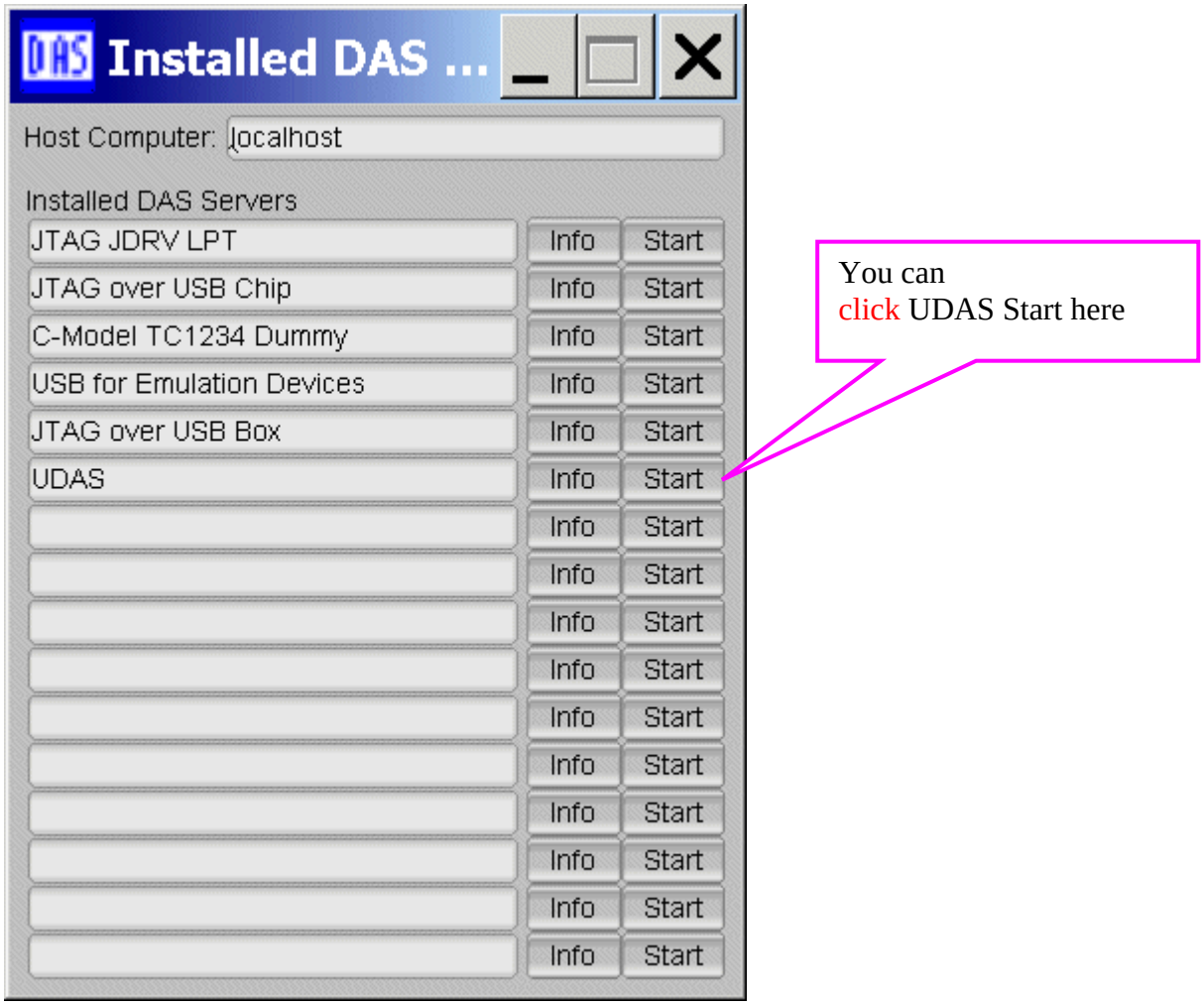
If (U)DAS is not running – even though the board is connected – you may start UDAS with the following dialog:

Start – Programs – DAS – DAS Server Control Panel





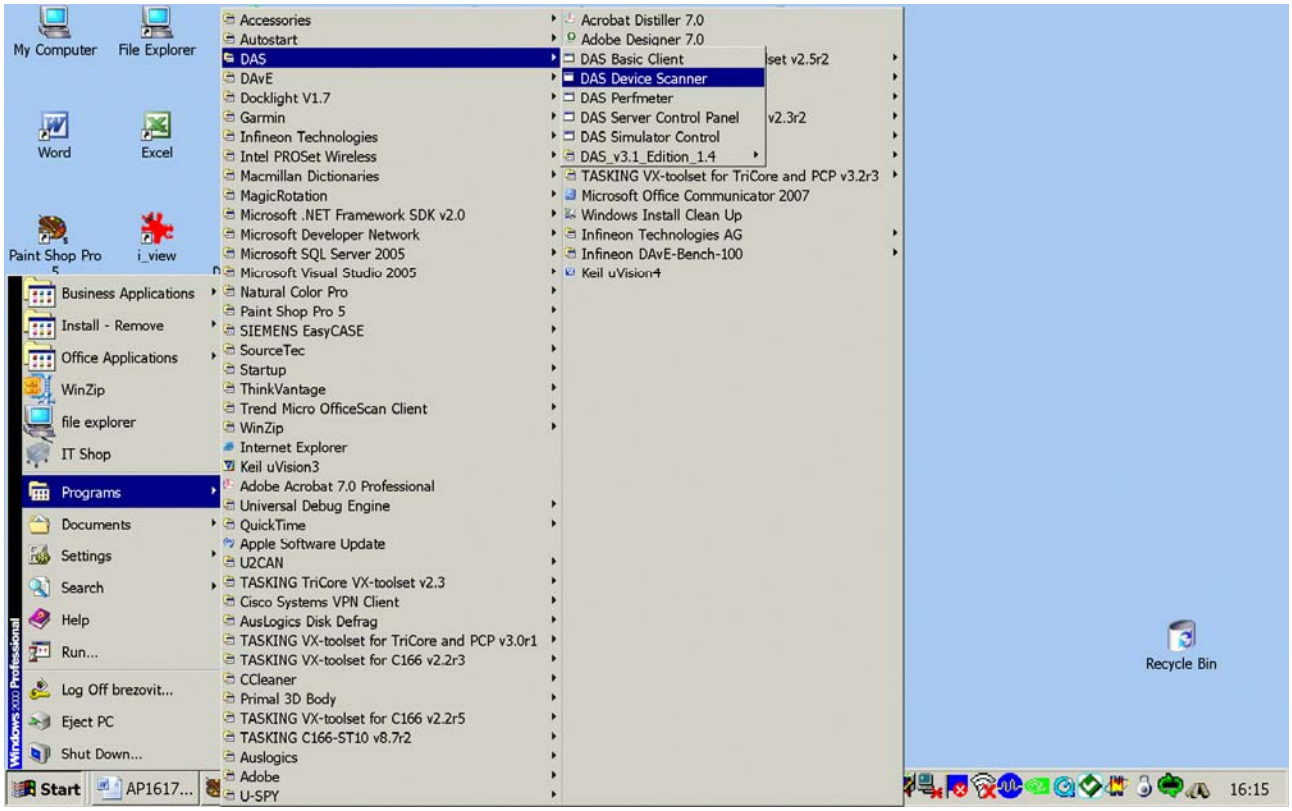
Click Installed DAS Servers:

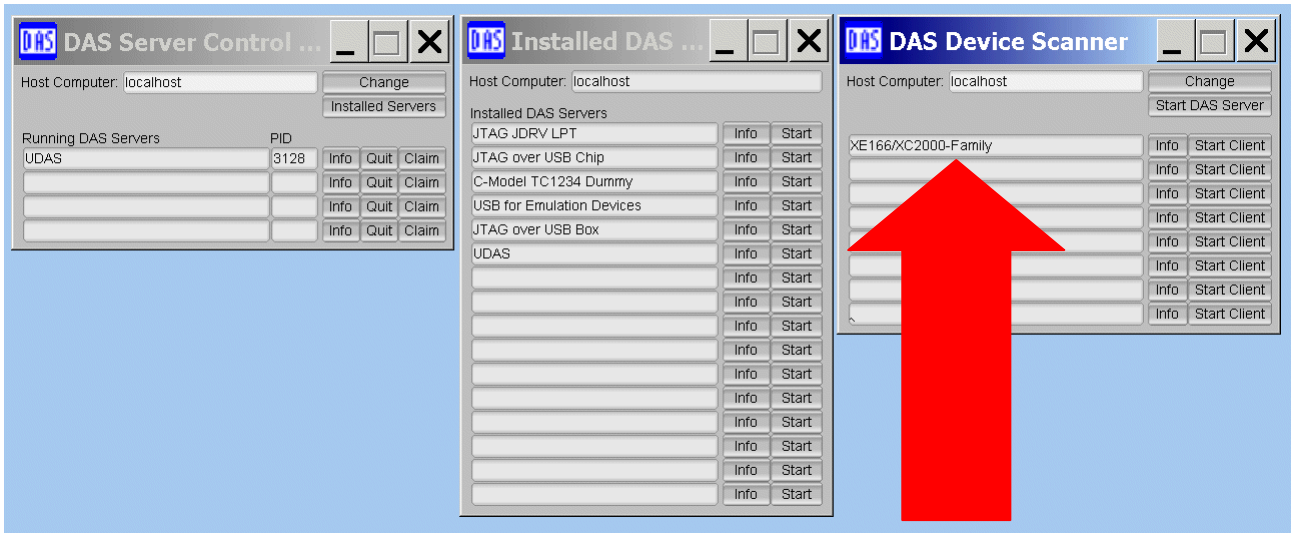


(U)DAS (Universal D e v i c e A c c e s s S e r v e r) Server hint 2:

With the DAS Device Scanner, you can see if the board has been recognized – to do this - you can use the following dialog:

Start – Programs – DAS – DAS Device Scanner

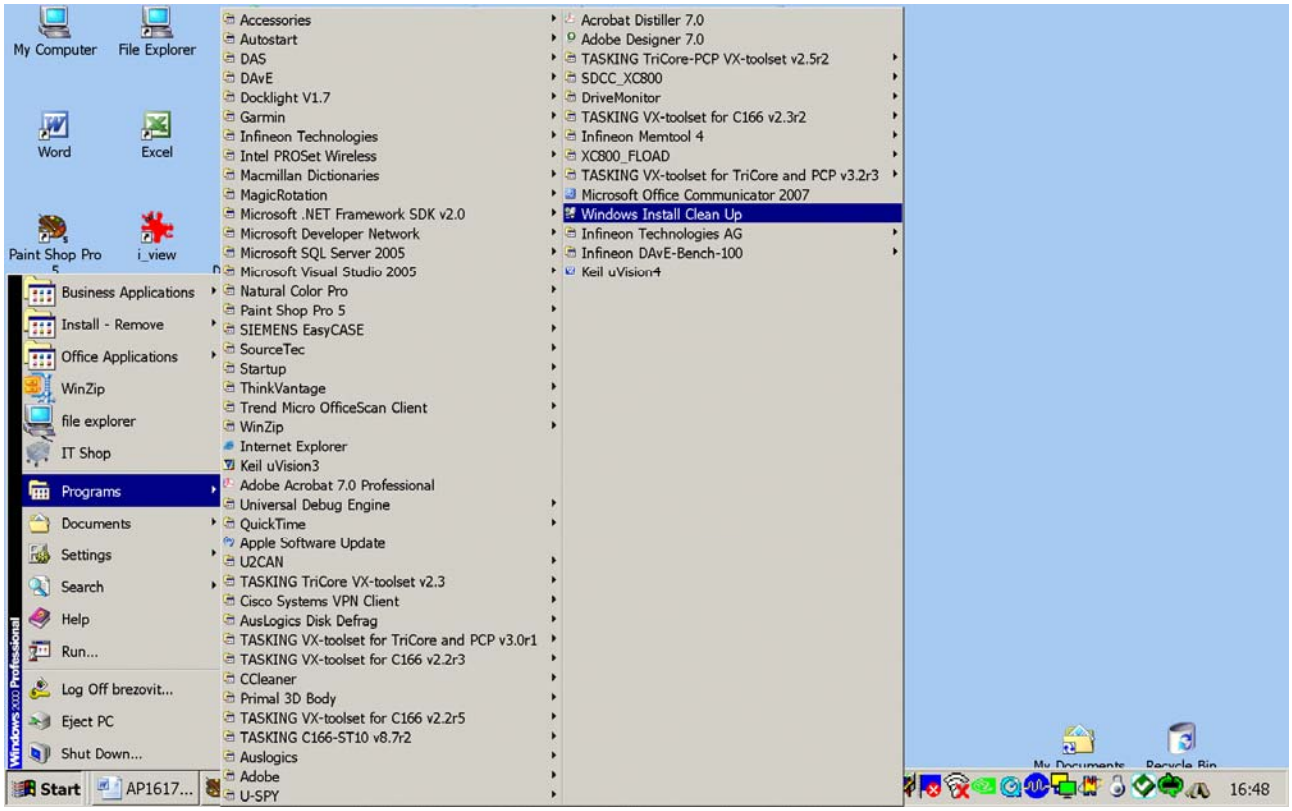


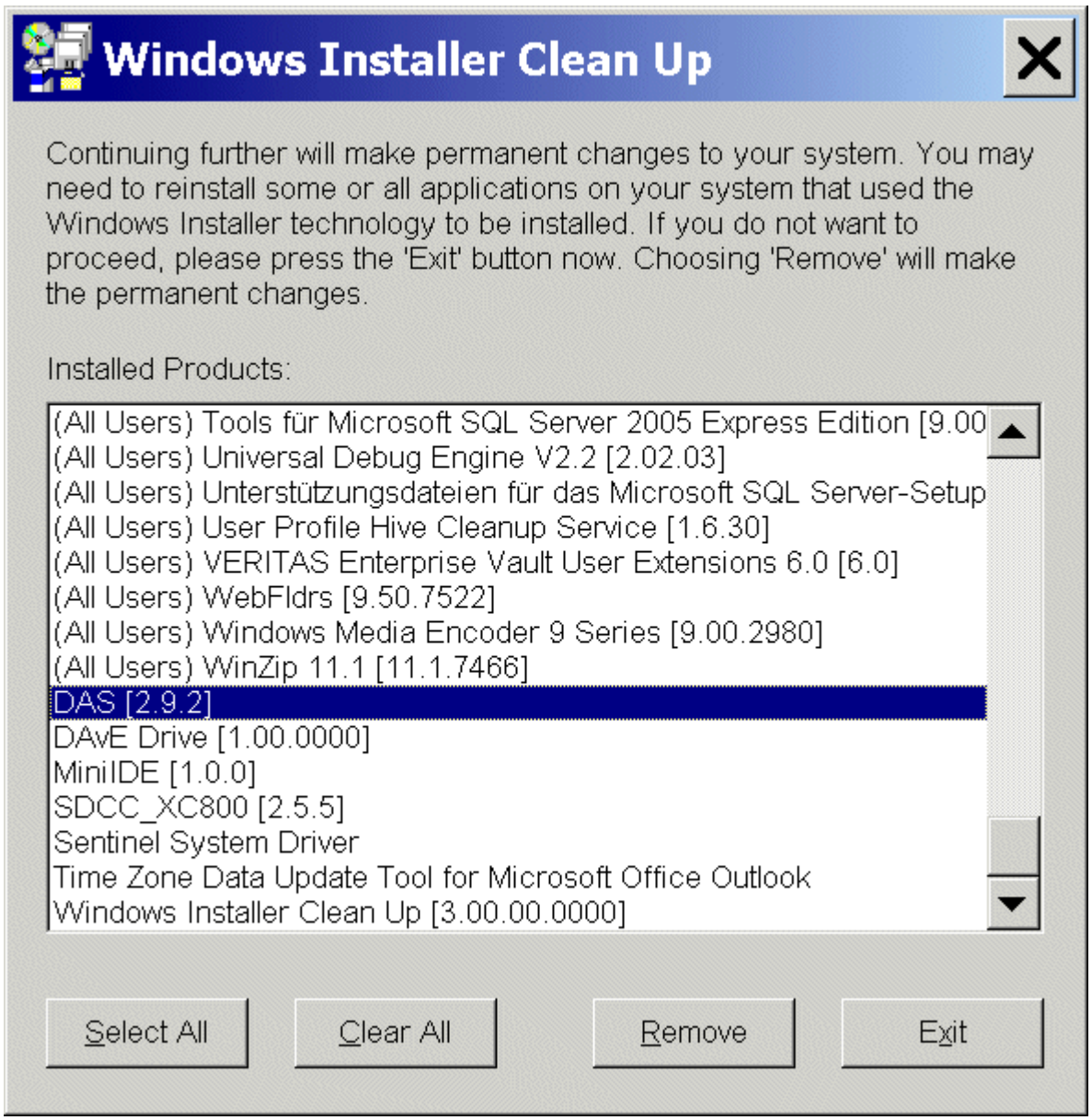


(U)DAS (Universal D e v i c e A c c e s s S e r v e r) Server hint 3:

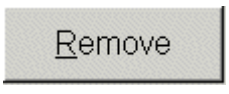
If you want to remove DAS you can use the following dialog:

Execute msicuu2.exe (to install Windows Install Clean Up)
Start – Programs – Windows Install Clean Up





Select DAS[ ]

Click 

Note:
Be careful with this tool!
With only 2 mouse clicks you can remove all software on your computer!

2.) DAvE – Installation for XE16x microcontrollers:

Install DAvE (mothersystem):

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

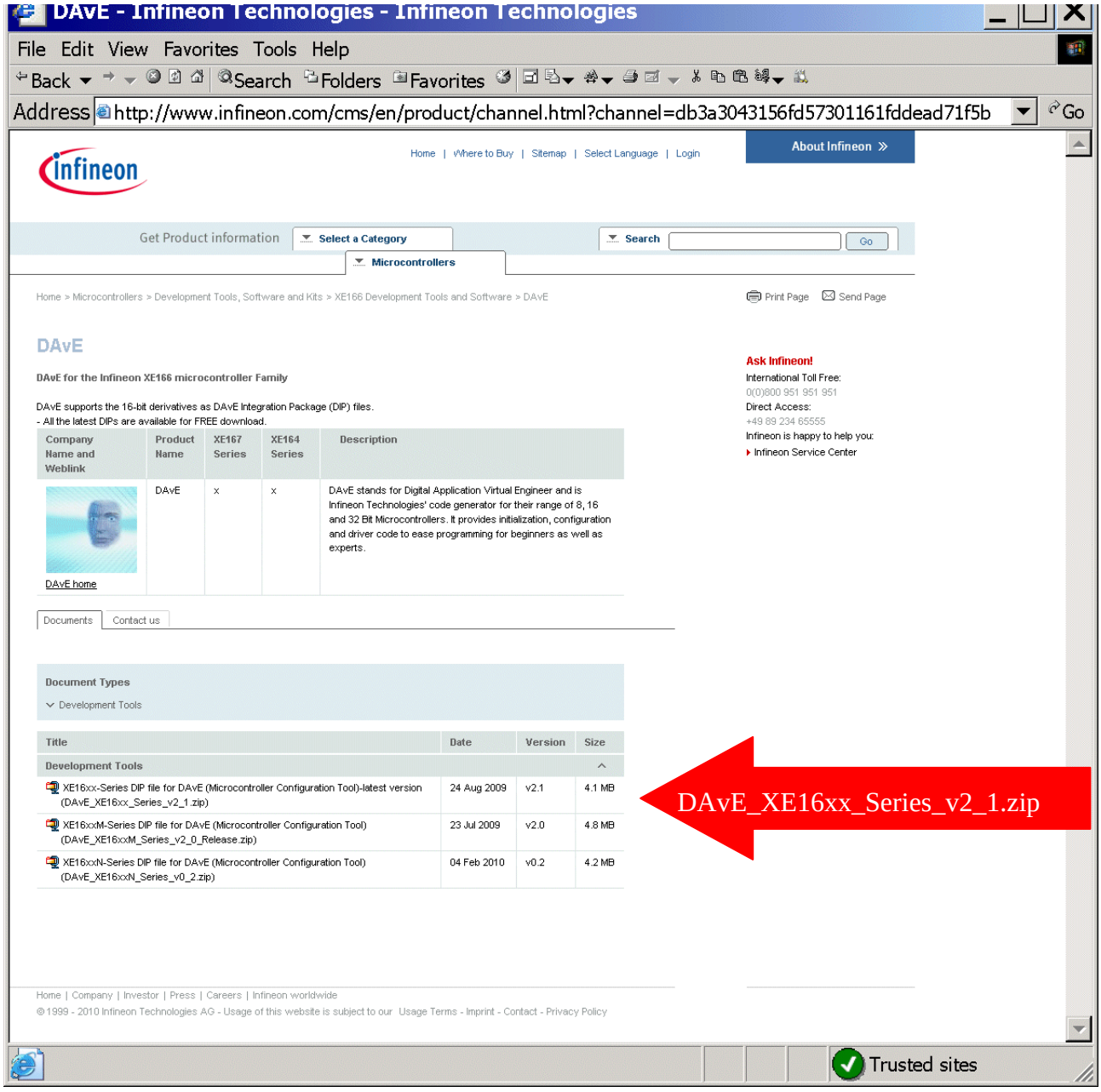
Title	Date	Version	Size
Tool Package			
DAvE - Mothersystem (DAvE_Mothersystem_v2_2r1.zip)	14 Dec 2009	V2.2	8.8 MB
DAvE - Mothersystem (setup.exe)	14 Dec 2009	V2.2	8.9 MB

and execute setup.exe to install DAvE .

Note:
Abort the installation of Acrobat Reader.

Install the XE164 microcontroller support/update (XE16xx_Series.dip):

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



Unzip the zip-file "DAvE_XE16xx_Series_v2_1.zip" and save "XE16xx_Series_v2.1.dip"
@ e.g. C:\DAvE\ XE16xx_Series_v2.1.dip.

2.)

Start DAvE - (click

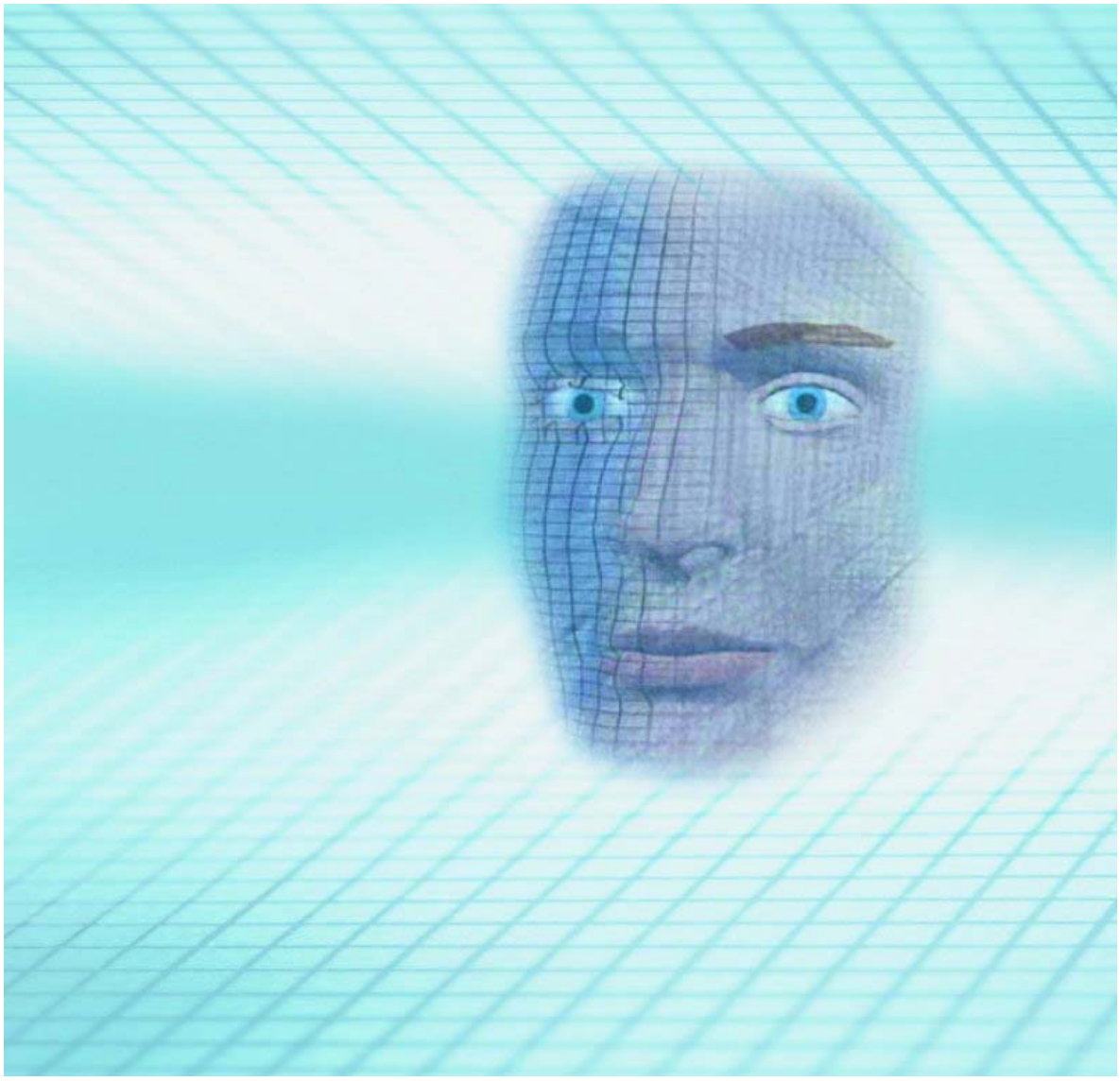
)

3.)

View
Setup Wizard
Default: • Installation
Forward>
Select: • I want to install products from the DAvE's web site
Forward>
Select: C:\DAvE
Forward>
Select: Available Products
click ✔ XE16xx_Series
Forward>
Install
End

4.) DAvE is now ready to generate code for the XE16x microcontrollers.

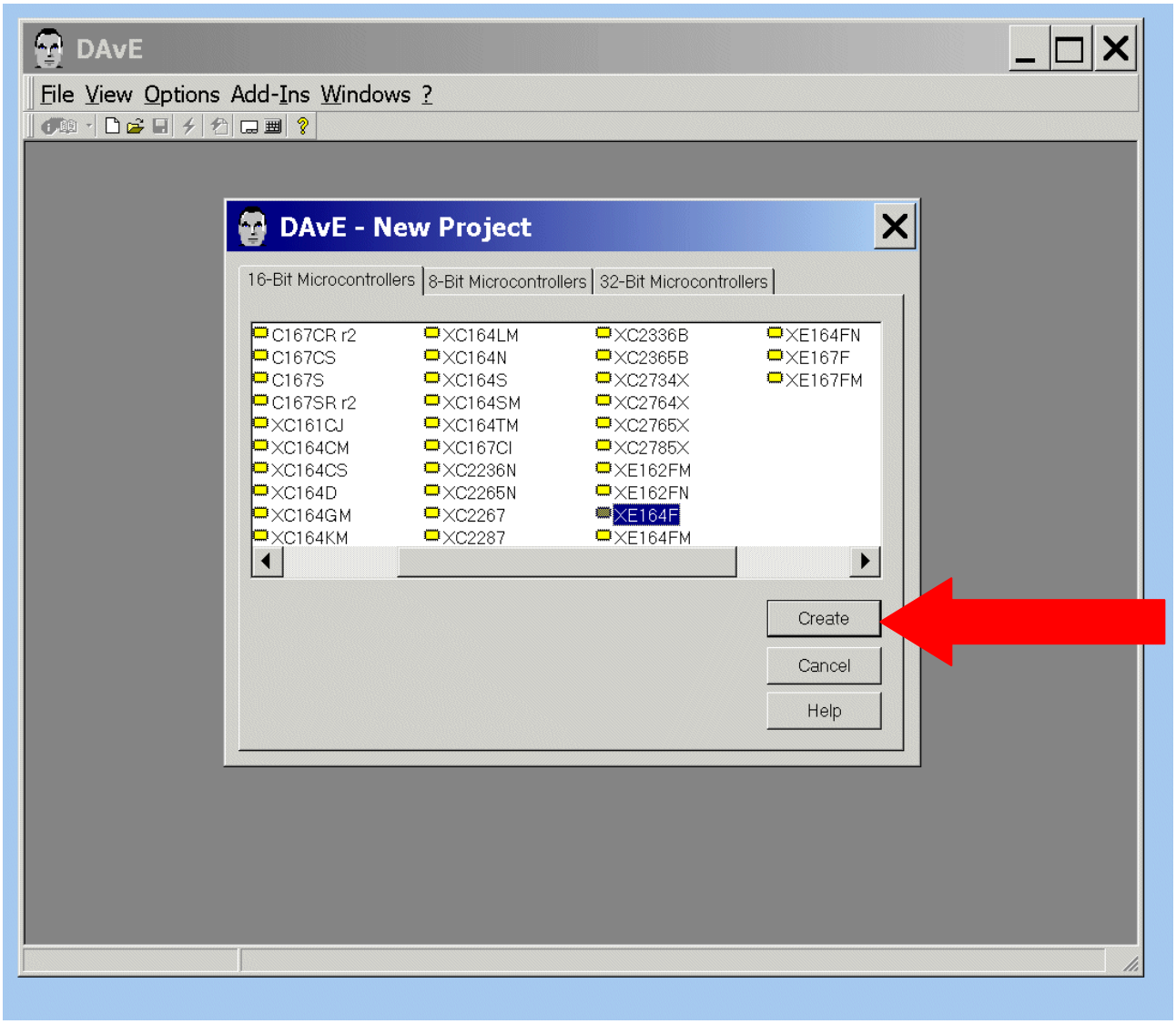
3.) DAvE - Microcontroller Initialization after Power-On:



Start the program generator DAvE and select the XE164 microcontroller:

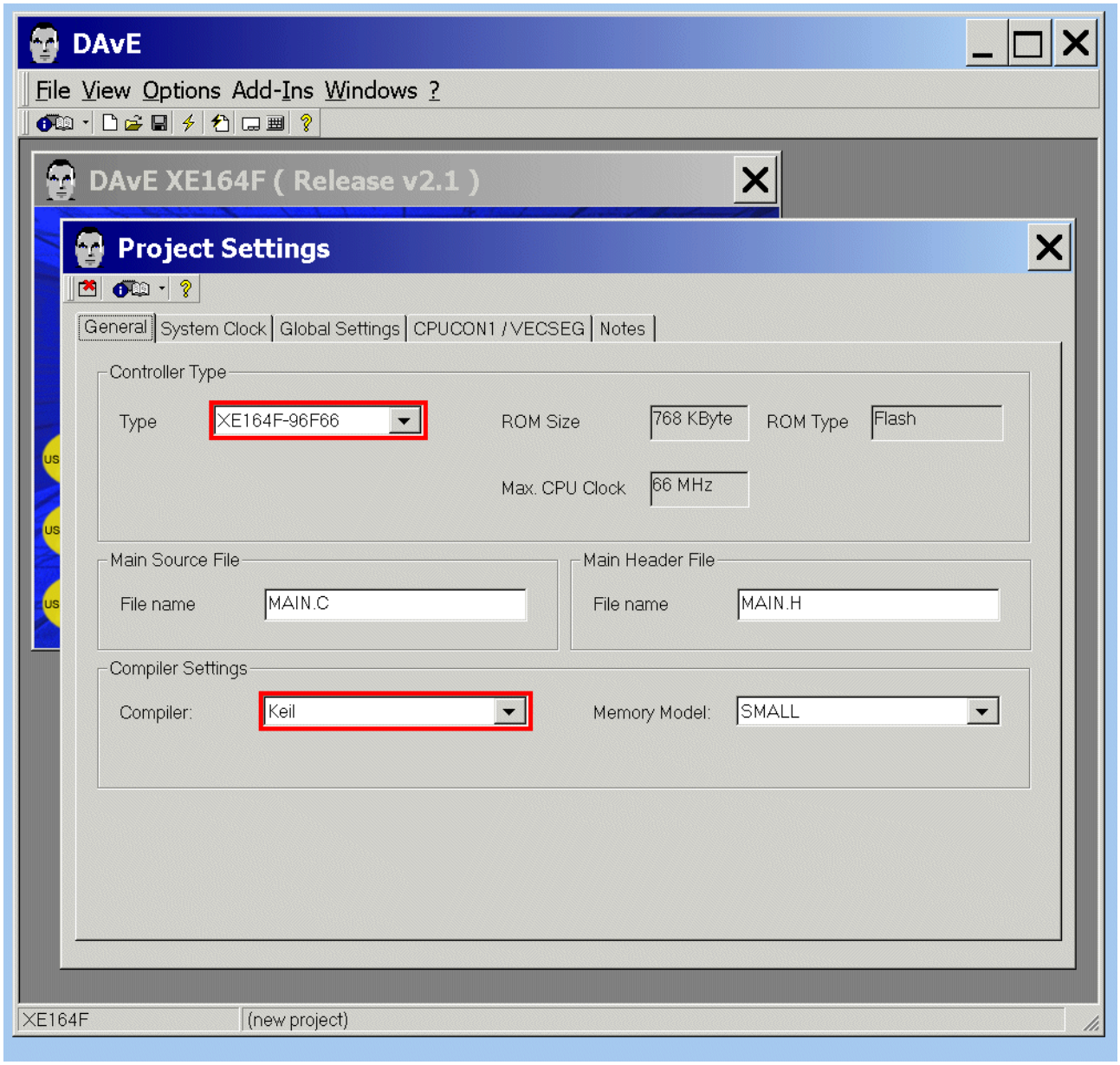
View - Project Window (Closes the Project Window)
View - Command Window (Closes the Command Window)

File - New
DAvE – New Project: click 16-Bit Microcontrollers
Select XE164F
Create



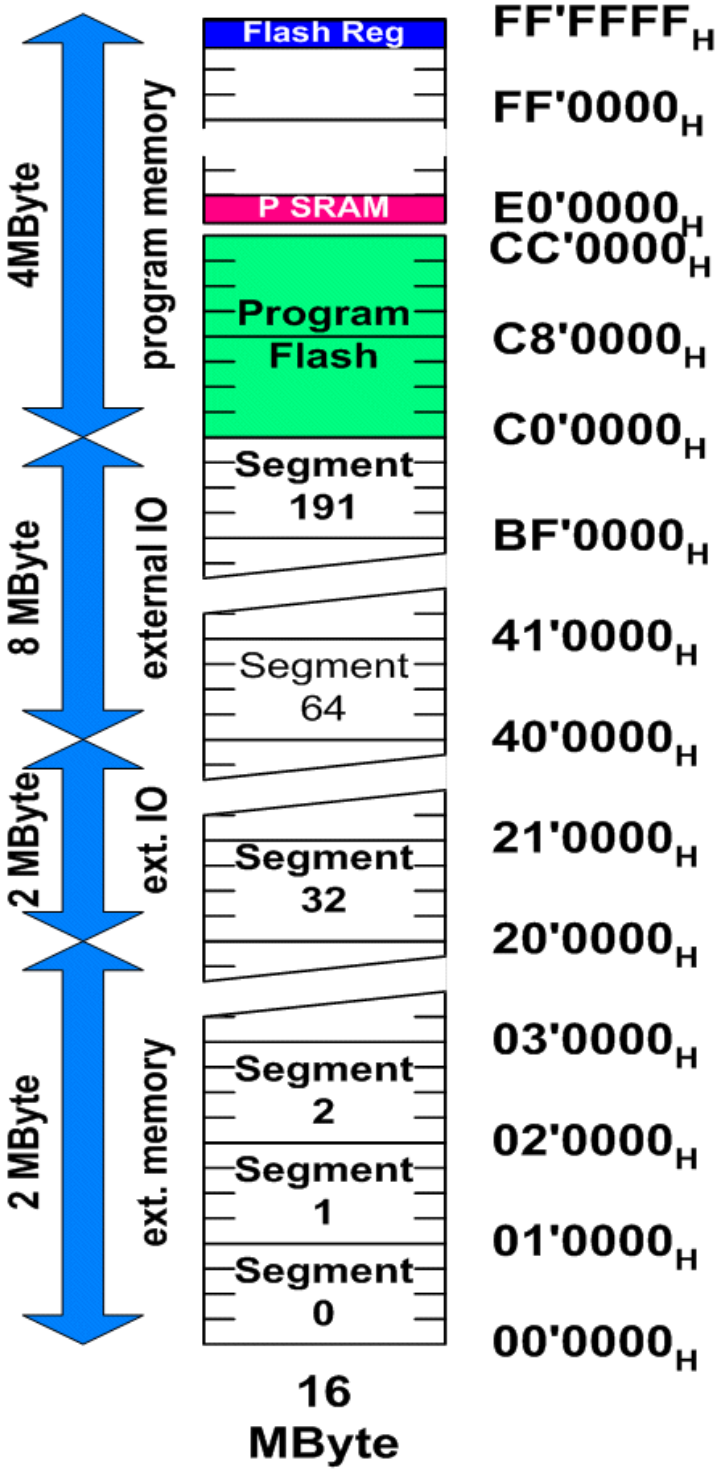
Choose the Project Settings as you can see in the following screenshots:

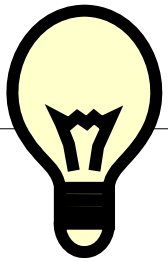
Project Settings: General: Controller Type: Type: check/select XE164F-96F66
Project Settings: General: Compiler Settings: Compiler: check/choose Keil



Note:
You can change file names (e.g. MAIN.C, MAIN.H) anytime.

Additional information: Address Space and Memory Model:





Additional information: Address Space and Memory Model:

Memory models:

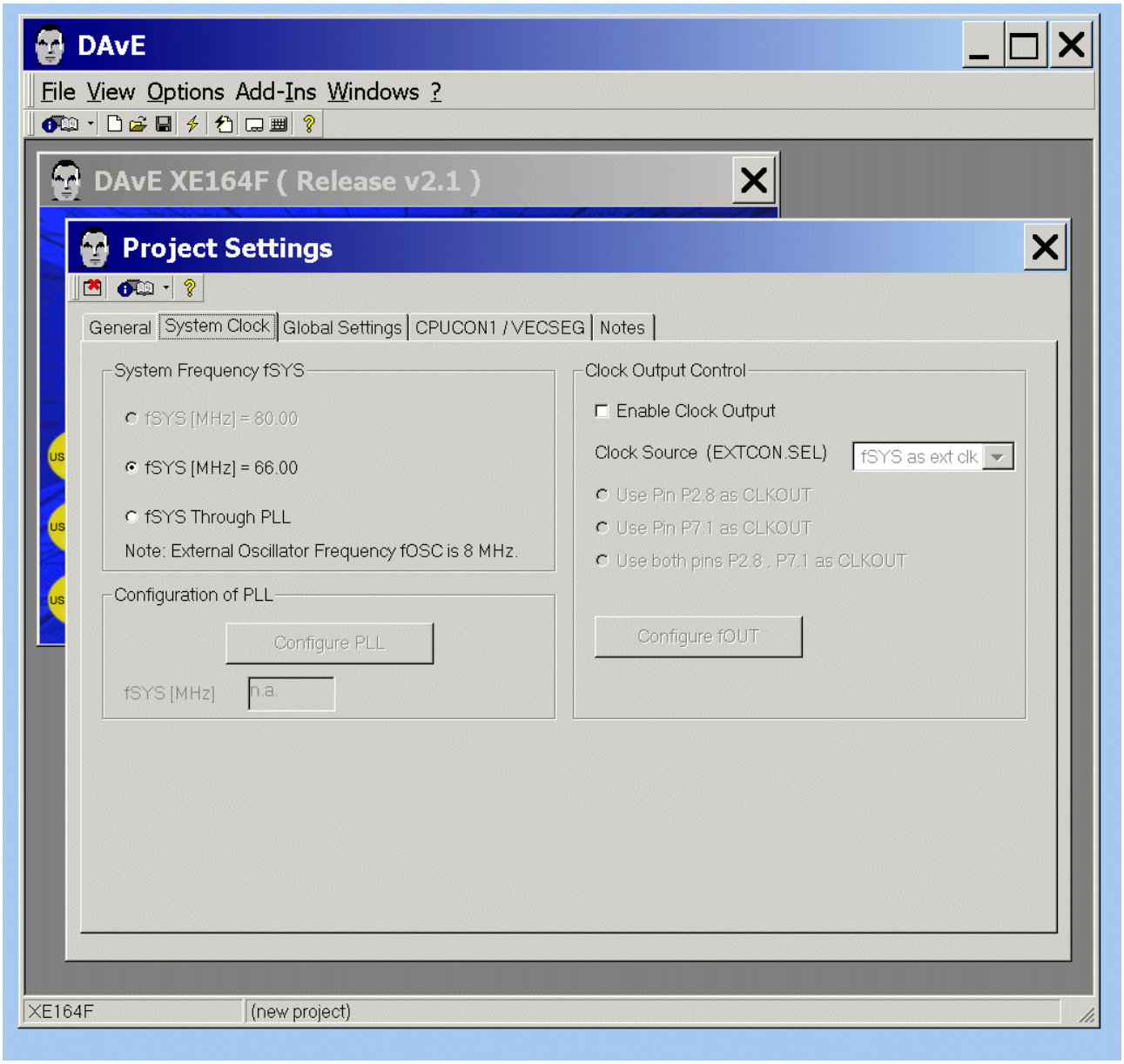
TASKING memory model	Default function memory qualifier[1]
TINY	_near
SMALL	_huge
MEDIUM	_near
LARGE	_huge
HUGE	_huge

Keil	Data space	Data object	Code space		TASKING	Data space	Data object	Code space
TINY	<64K	<16K	<64K	⇨	TINY	<64K	<64K	<64K
SMALL[2]	<64K	<16K	<64K	⇨	SMALL[2] I,II or III	<64K	<64K	<16M
MEDIUM[2]	<64K	<16K	<16M	⇨	SMALL[2] I,II or III	<64K	<64K	<16M
COMPACT	<16M	<16K	<64K	⇨	MEDIUM	<16M	<16K	<64K
LARGE	<16M	<16K	<16M	⇨	LARGE[3]	<16M	<16K	<16M
HCOMPACT	<16M	<64K	<64K	⇨	MEDIUM	<16M	<16K	<64K
HLARGE	<16M	<64K	<16M	⇨	LARGE[3]	<16M	<16K	<16M
XLARGE	<16M	<16M	<16M	⇨	HUGE	<16M	<16M	<16M

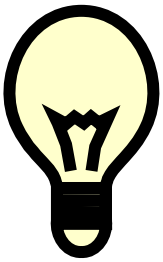
Memory qualifiers for variables:

Keil	TASKING[1]	Description
bdata	_bita	bit-addressable RAM
ebdata	_bita	extended bit-addressable memory
idata	_iram	internal RAM
sdata	_system	system page
near	_near	Keil: always restricted to 16K TASKING: 16K or 64K, depends on memory
far	_far	one object is limited to 16K
huge	_shuge	one object is limited to 64K
xhuge	_huge	objects are limited to available memory space

Project Settings: System Clock: (do nothing)

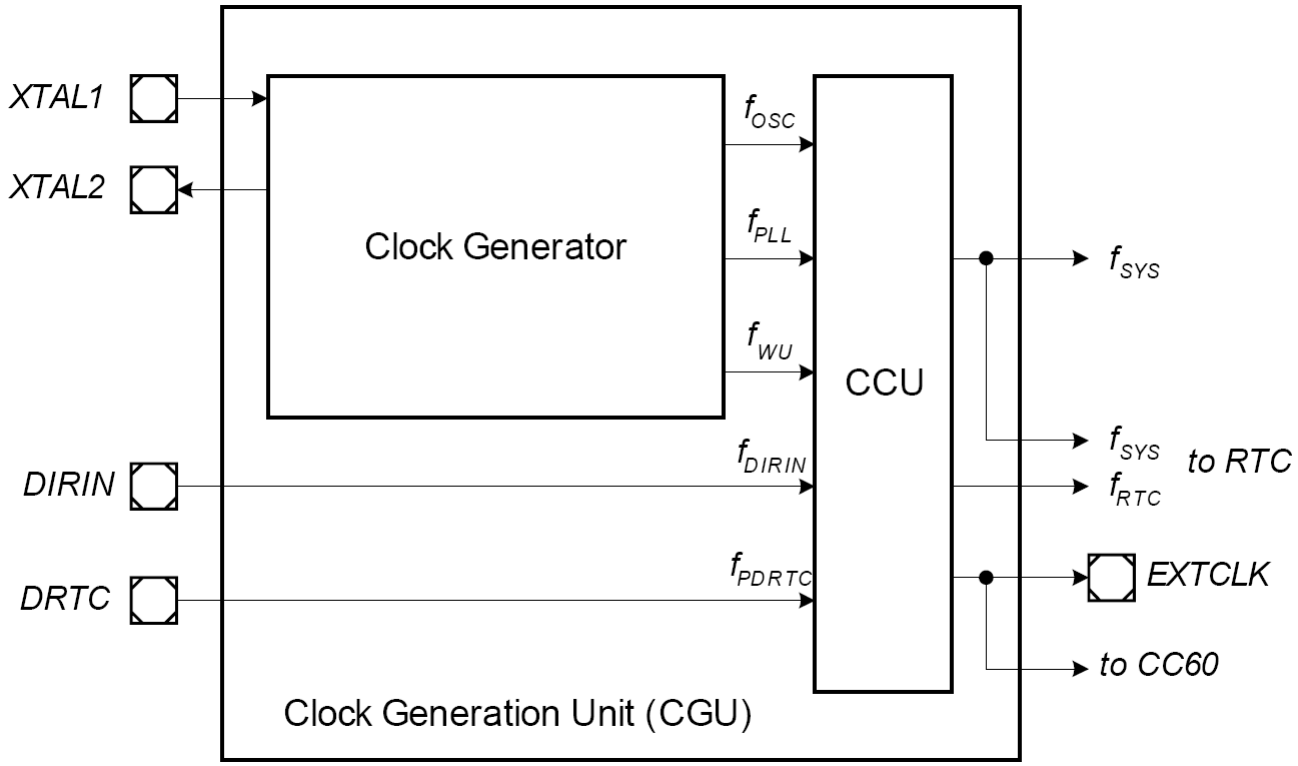


Note (Source: DAvE):
Configuration of the System Clock:
- VCO clock used, input clock is connected
- input frequency is 8,00 MHz (XTAL1)
- configured system frequency is 66,00 MHz
- system clock is 66.00 MHz



Additional information: Clock System (Source: User's Manual):

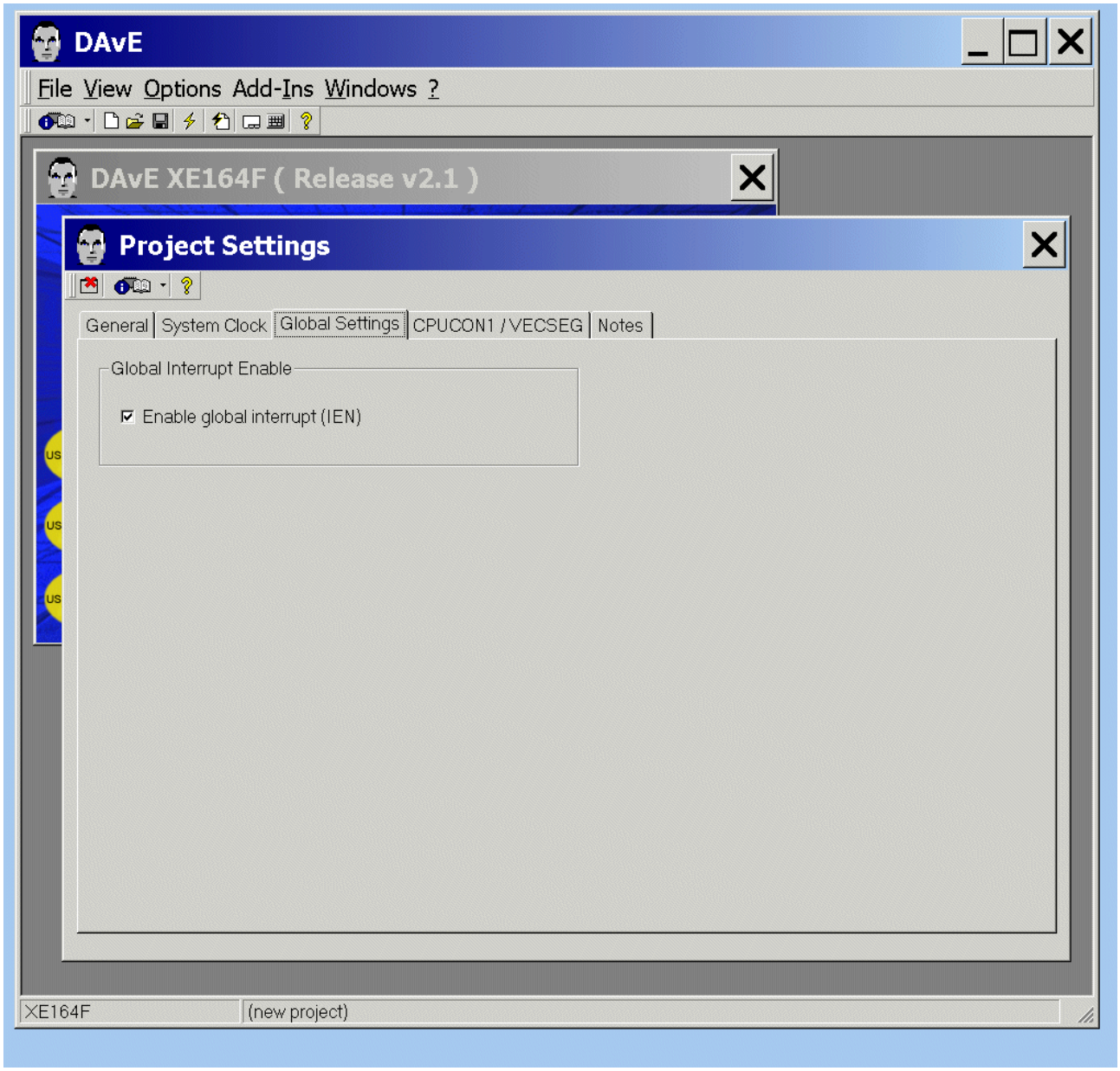
Clock Generation Unit (CGU) Block Diagram:



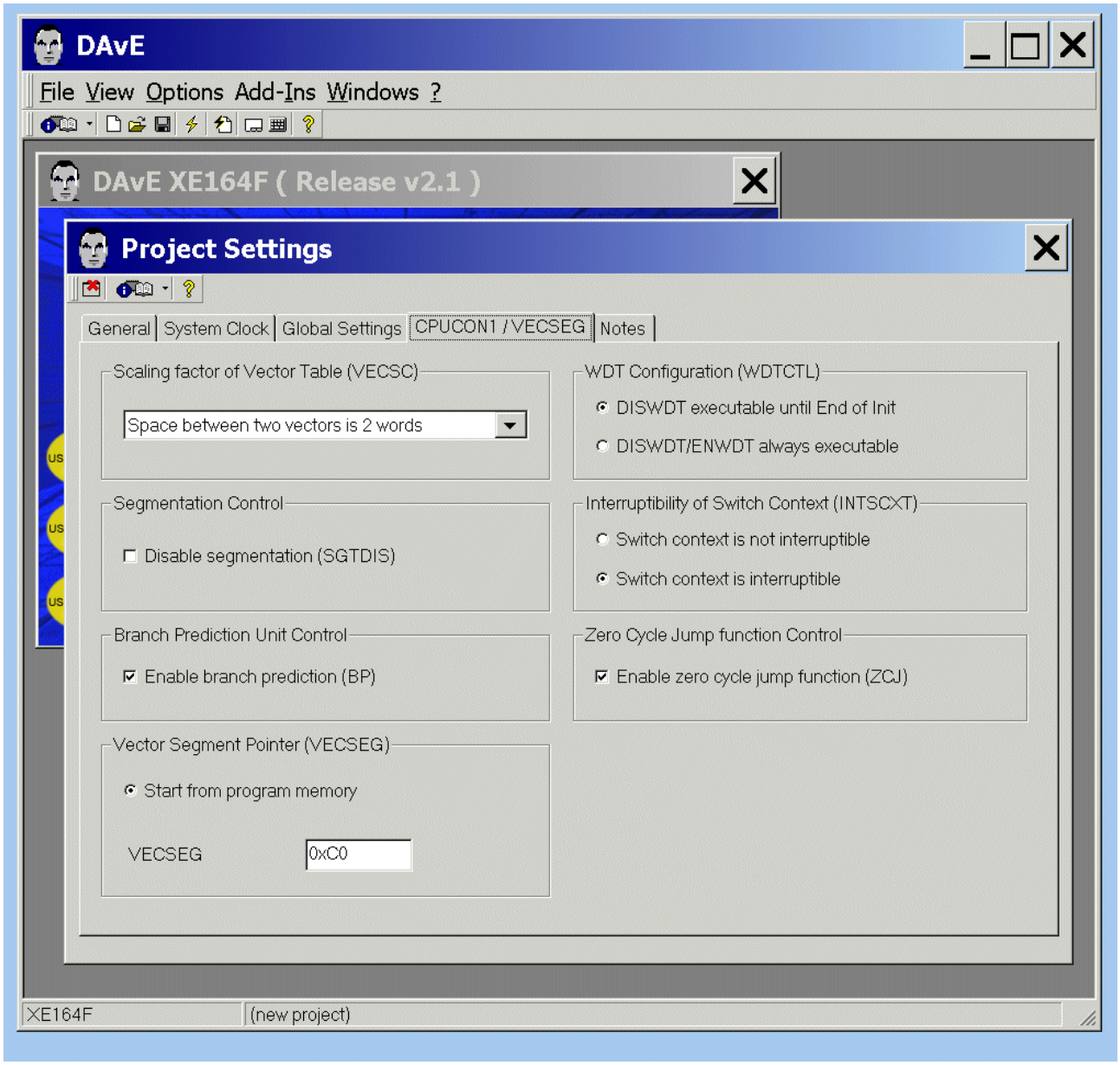
Note:
The CGU can convert a low-frequency external clock to a high-speed internal clock, or can create a high-speed internal clock without external input.

The system clock f_{SYS} is generated out of four selectable clocks:
• PLL clock f_{PLL}
• Wake-Up clock f_{WU}
• The Direct Clock f_{OSC}, from pin XTAL1
• Input DIRIN as Direct Clock Input f_{DIR}

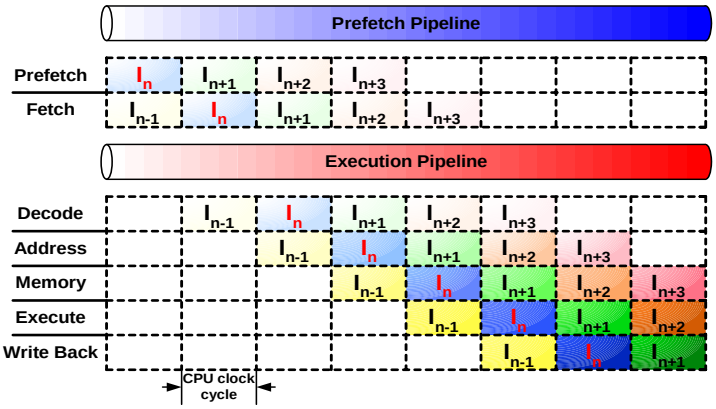
Project Settings: Global Settings: (do nothing. Do not change configuration)

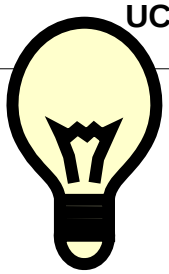


Project Settings: CPUCON1/VECSEG: (do nothing)

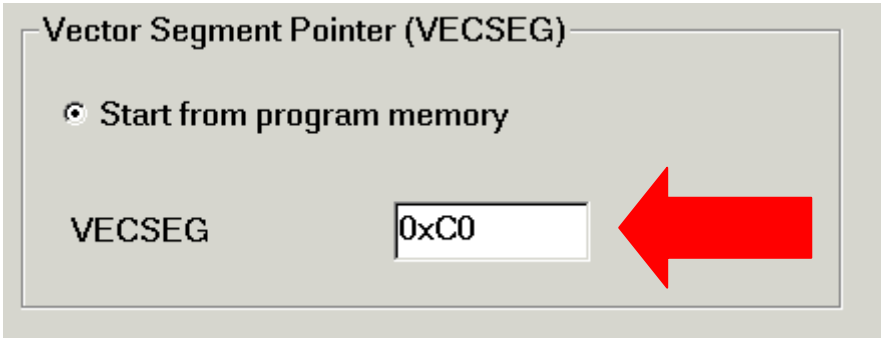


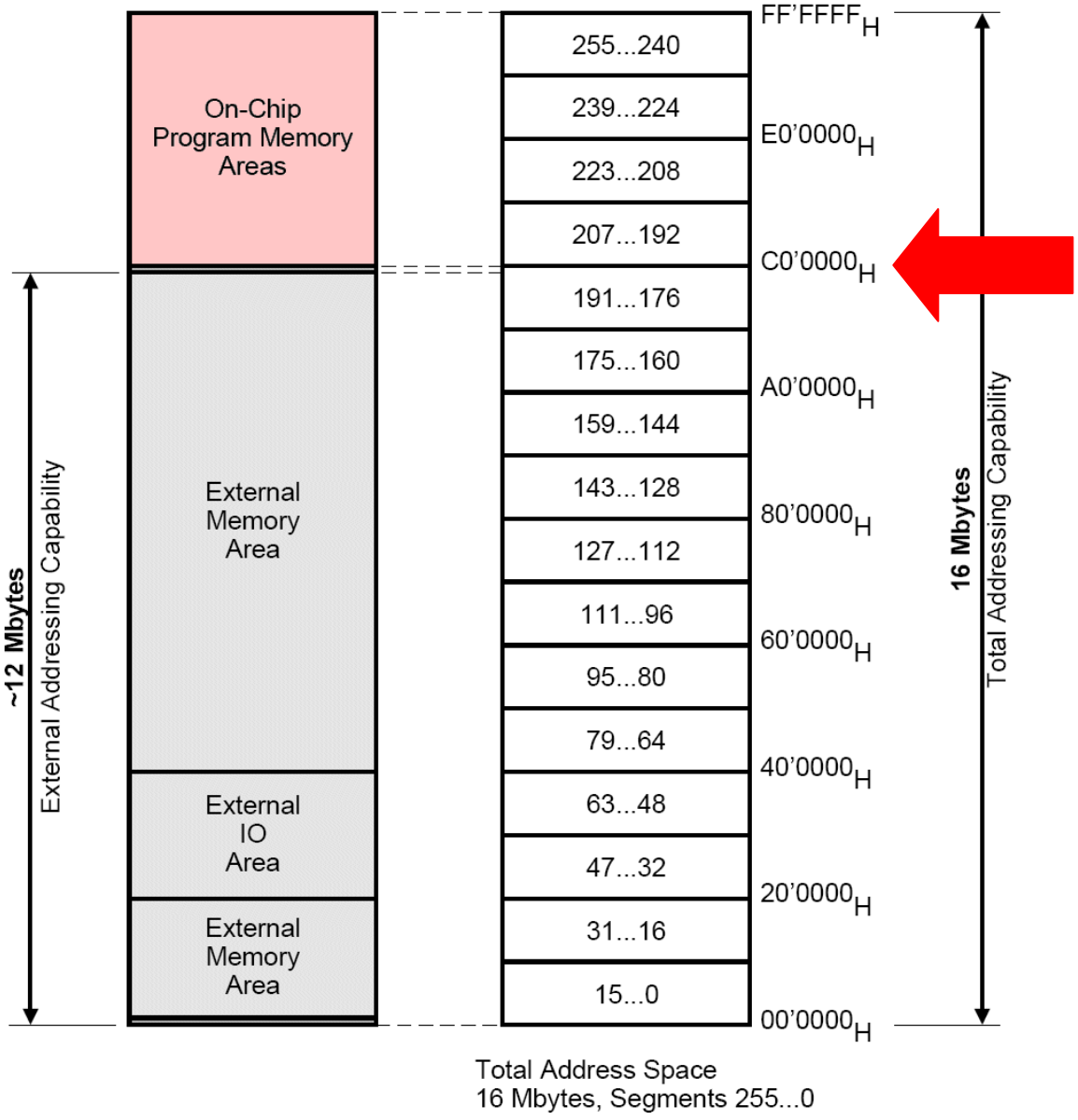
Note:
We should not change the pipeline behaviour.



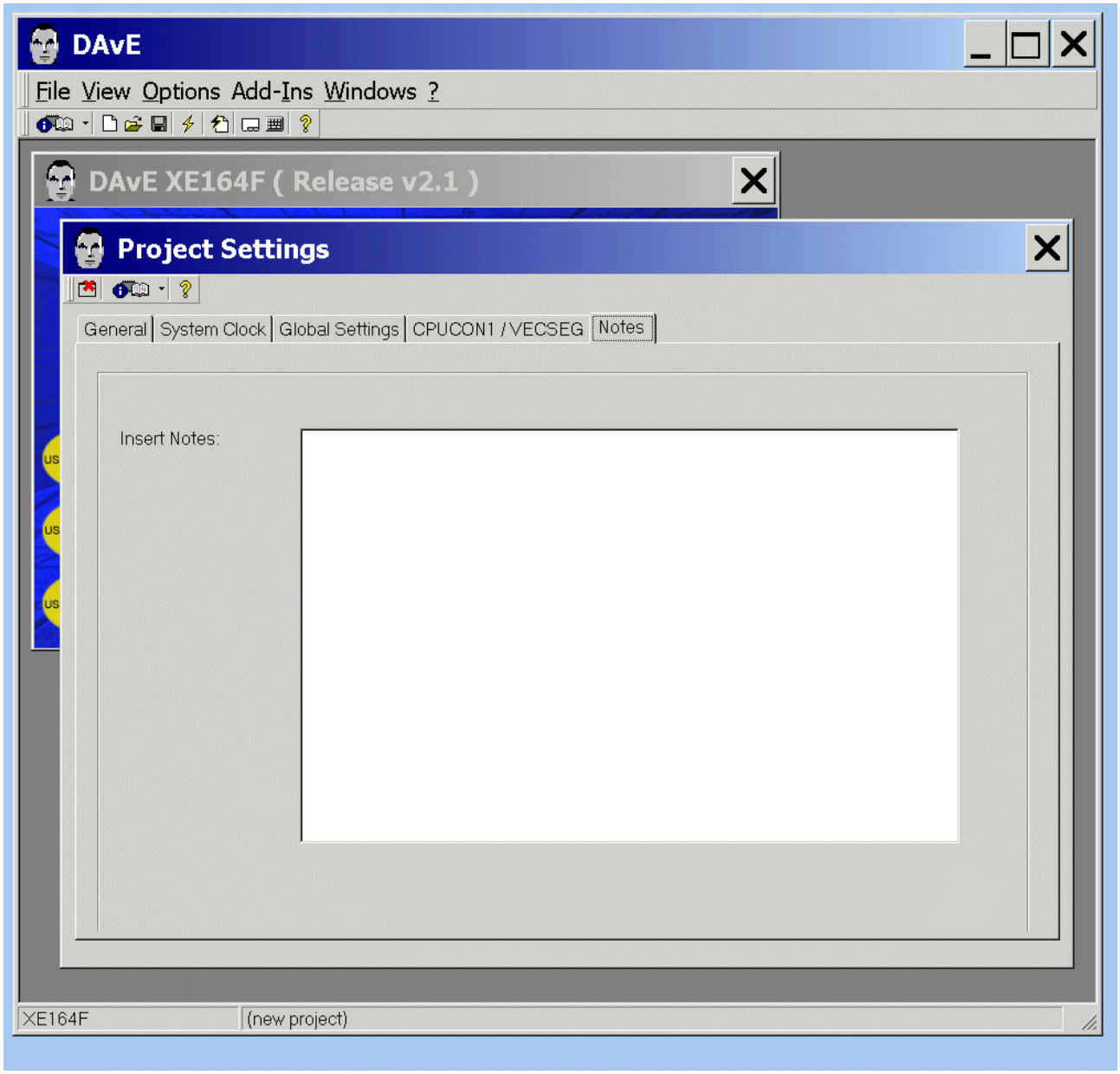


Additional information: Start from program memory (Source: User's Manual):

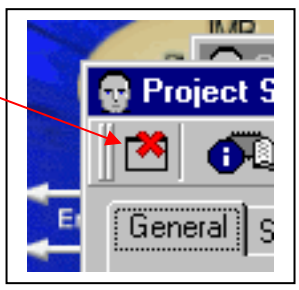




Project Settings: Notes: Insert Notes: If you wish, you can insert your comments here.

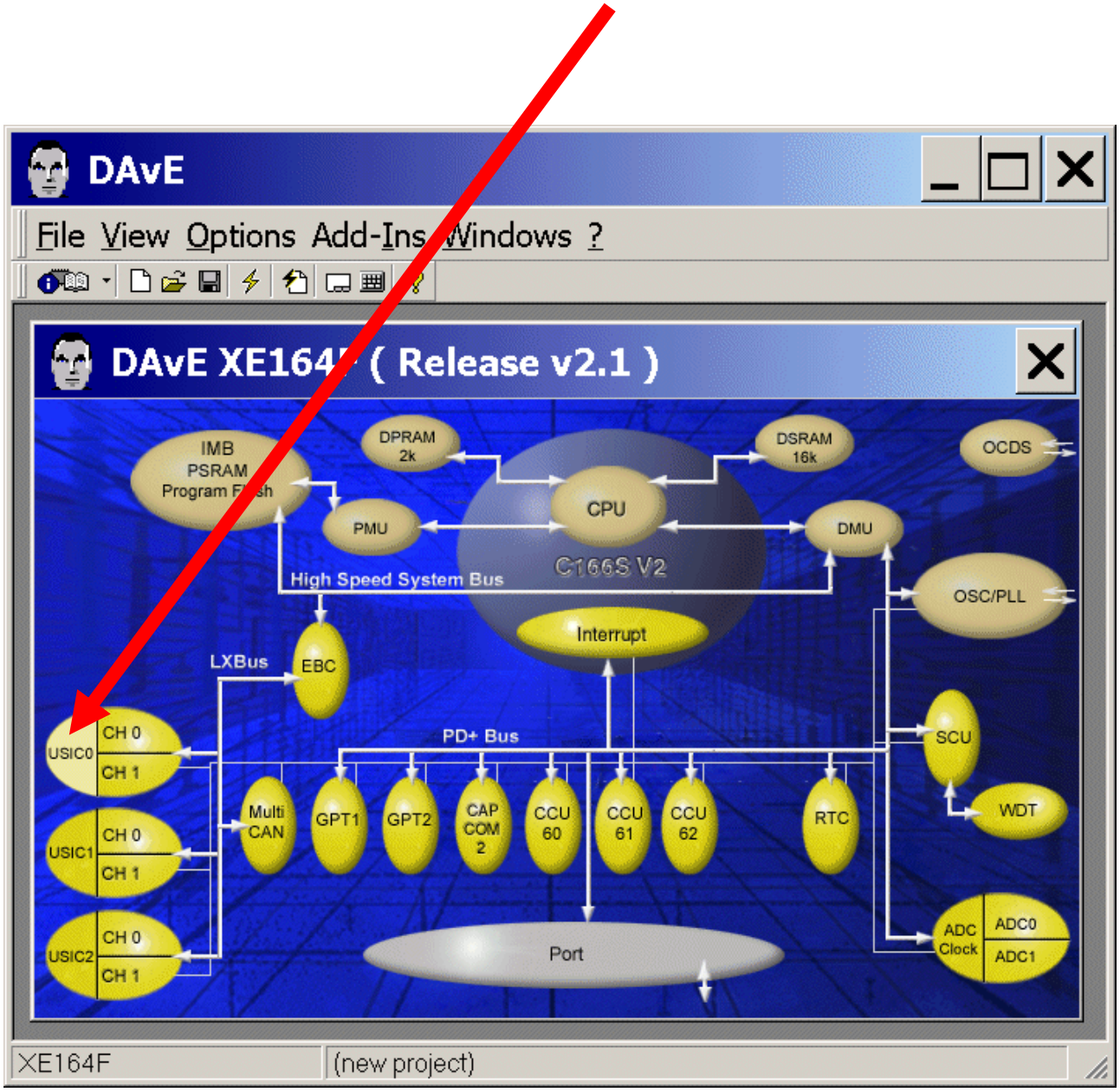


Exit and Save this dialog now by clicking the close button:



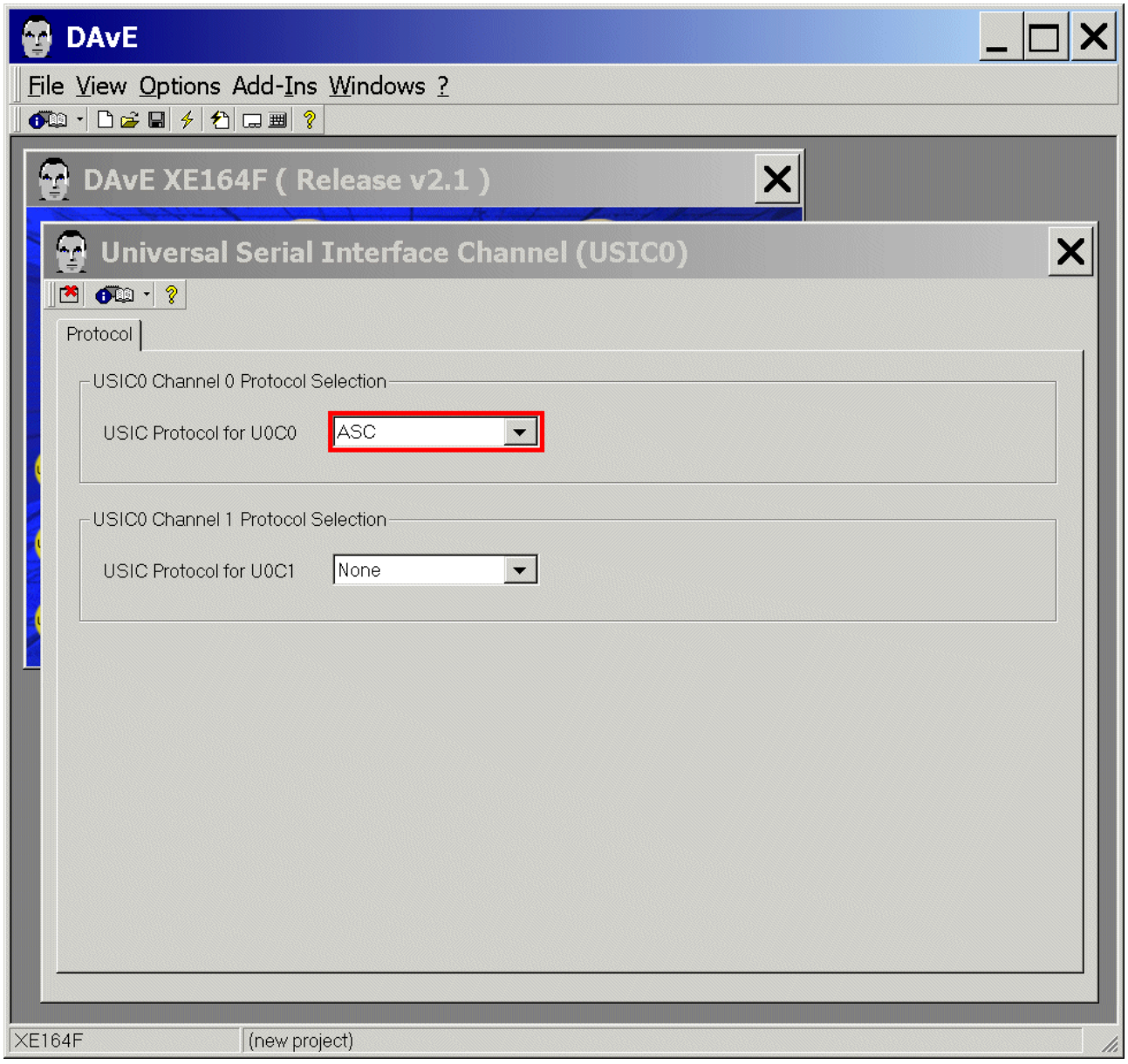
Configuration of the serial interface (Universal Serial Interface Channel module)
"ASC0" / UART / USIC0_CH0 / U0C0:

The configuration window/dialog can be opened by clicking the specific block/module (USIC0).



Note:
For serial communication with a terminal program running on your host computer U0C0 is used.

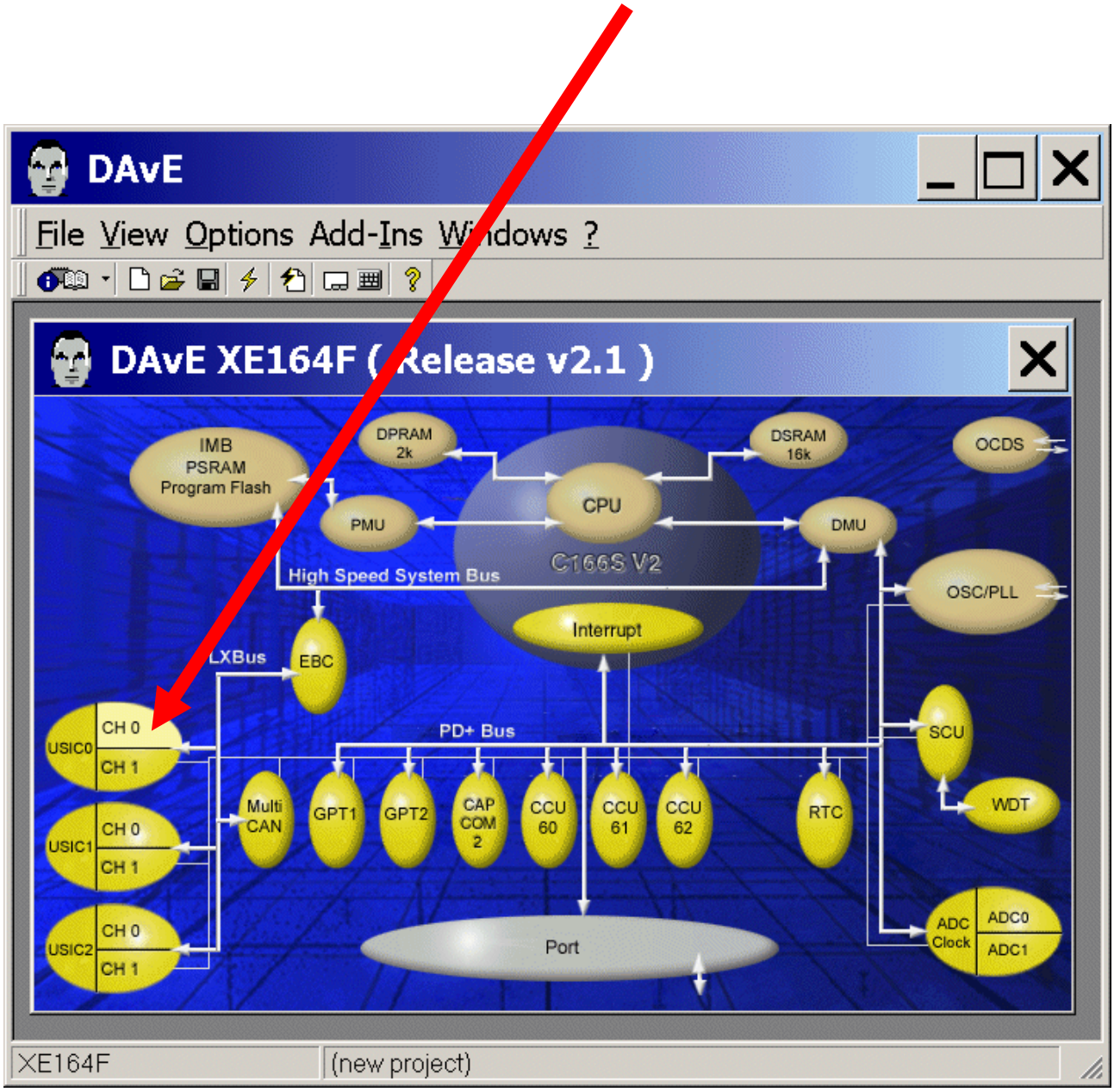
Protocol: USIC0 Channel 0 Protocol Selection: USIC Protocol for U0C0: select ASC

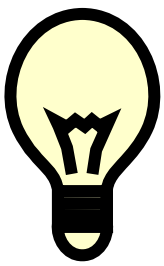


Exit and Save this dialog now by clicking the close button.

Configuration of the serial interface USIC0_CH0 / U0C0
(Universal Serial Interface Channel module 0, Channel 0):

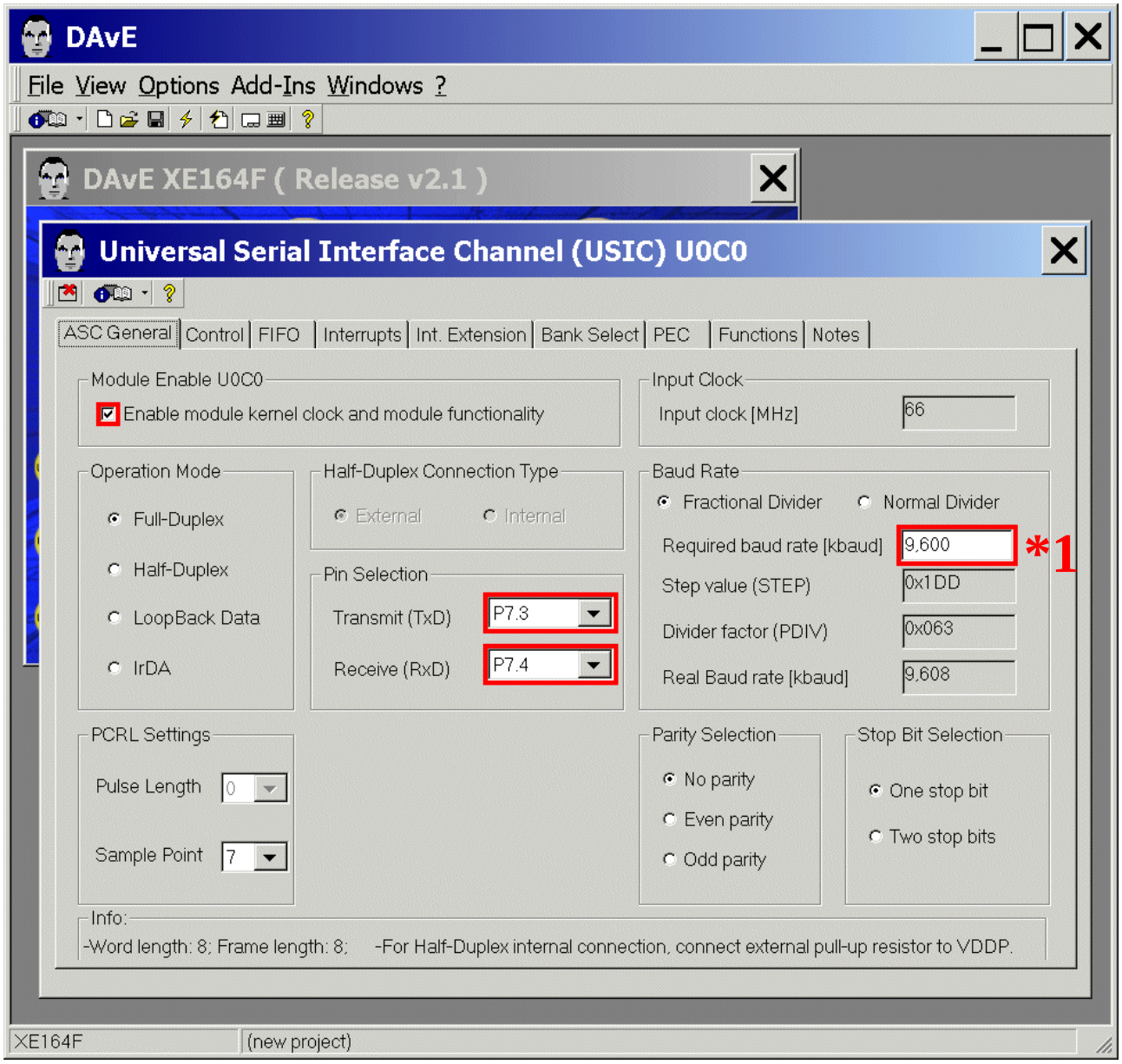
The configuration window/dialog can be opened by clicking the specific block/module (CH 0).





Note:
For serial communication with a terminal program running on your host computer U0C0 is used.

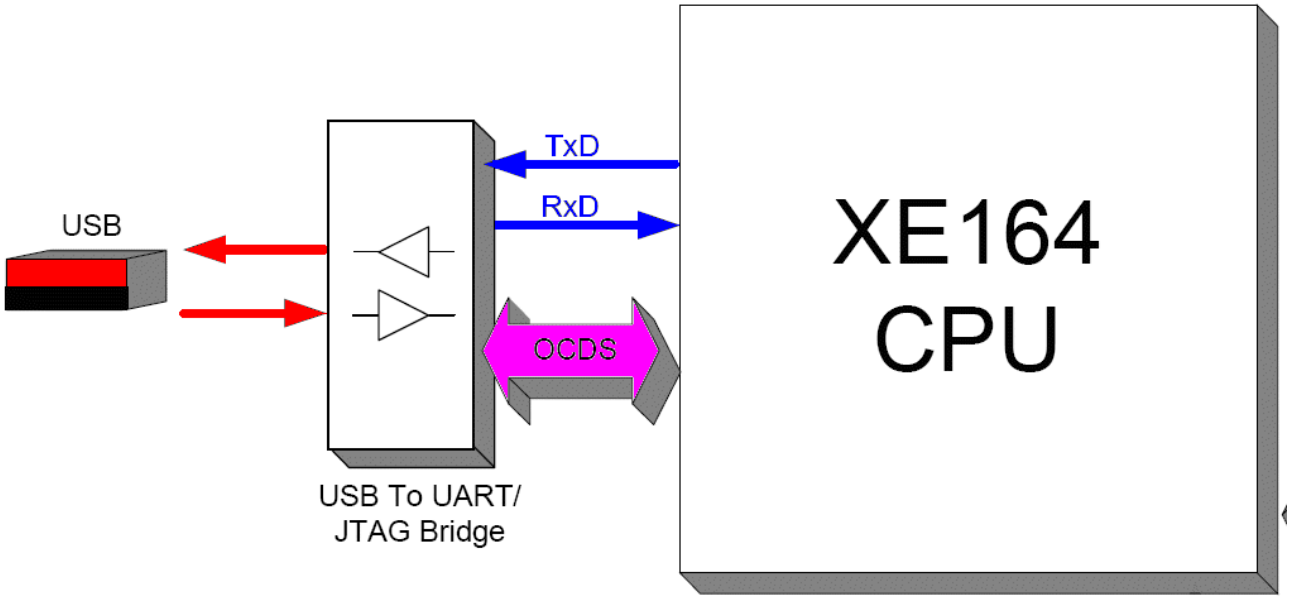
ASC General: Module Enable U0C0: tick ☑ Enable module kernel clock and module functionality
ASC General: Pin Selection: Transmit (TxD): select P7.3
ASC General: Pin Selection: Receive (RxD): select P7.4
ASC General: Baud Rate: Required baud rate [kbaud]: insert 9,600 <ENTER>

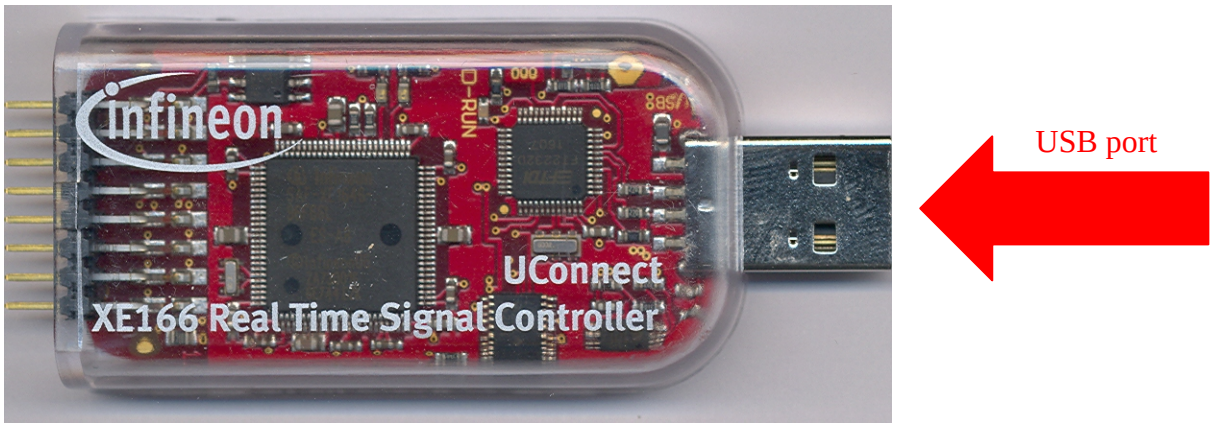


Note (*1):
Validate each alphanumeric entry by pressing <ENTER>.

Additional information: RS232 serial interface:

Note:
The RS232 serial interface (USIC_0_Channel_0 pins P7.3 and P7.4) is available via the USB port which converts the TTL-UART-signals to USB-signals (using a virtual COM port of the second USB channel of the FTDI FT2232 Dual USB to UART/JTAG interface).





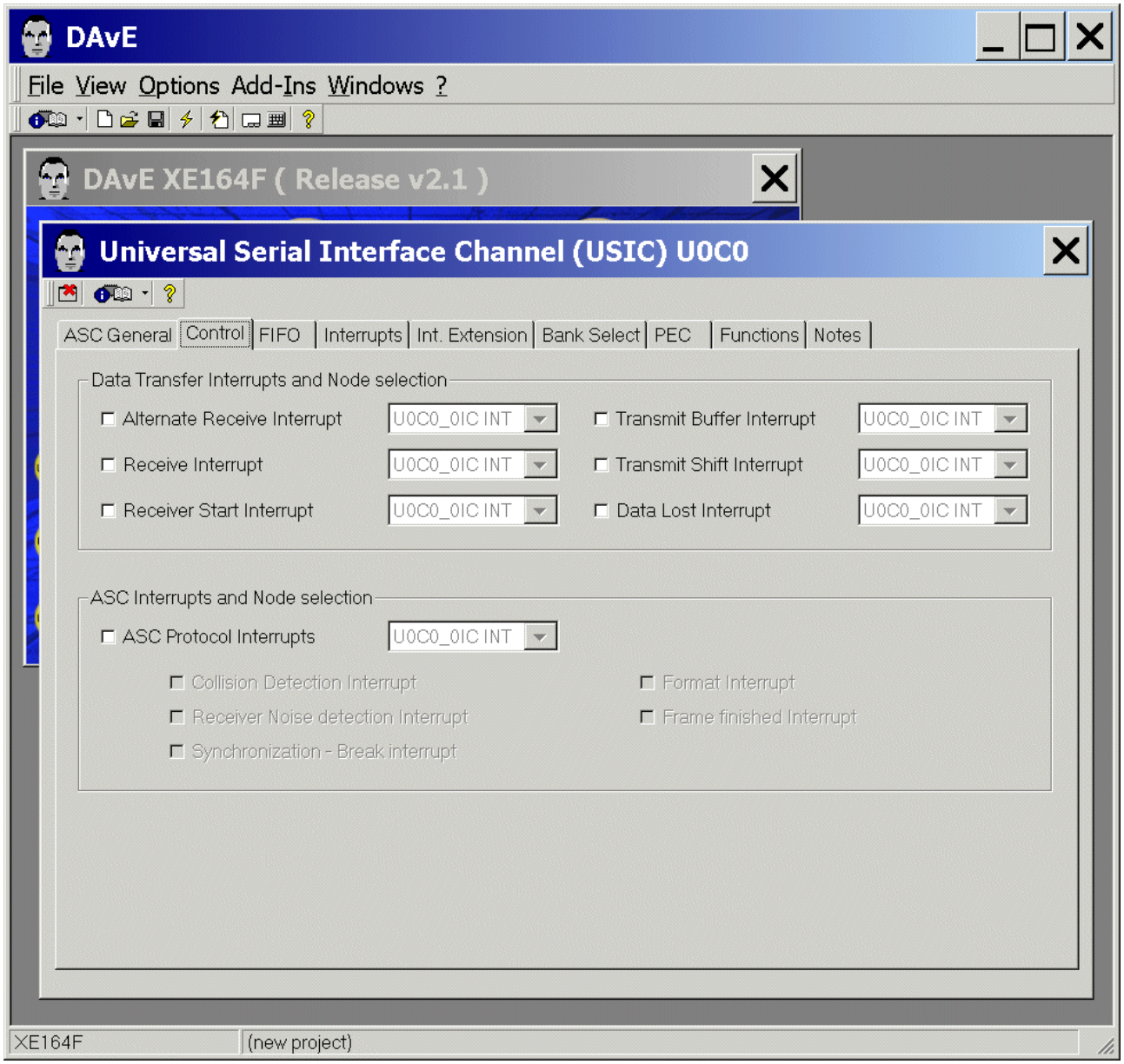
Additional information: Standard UART / USIC_0_Channel_0 Pins (Source: User's Manual):

Table 10-10 Configuration Data for Bootstrap Loader Modes

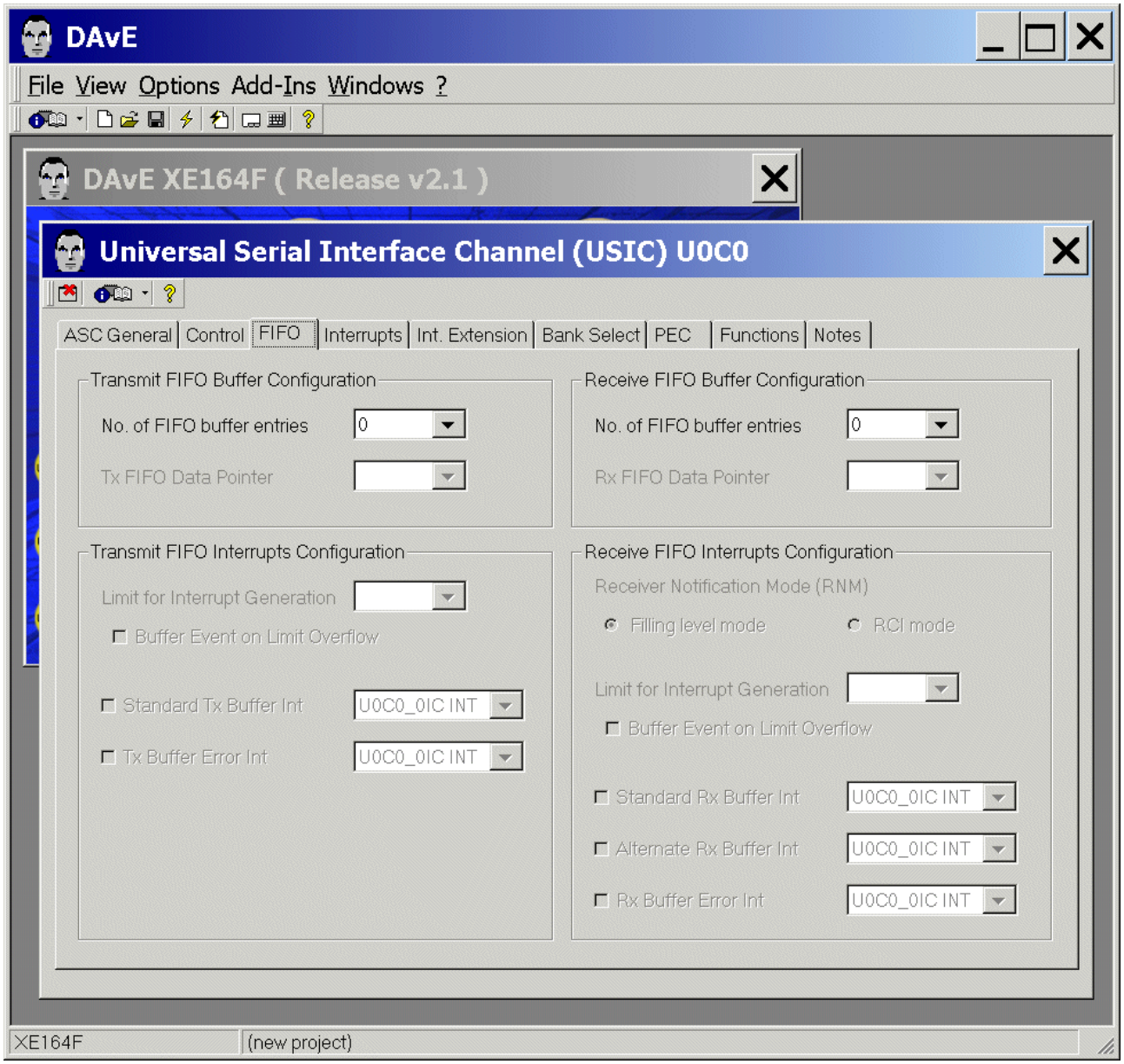
Bootstrap Loader Mode	Configuration on P10.3-0[1)]	Receive Line from Host	Transmit Line to Host	Transferred Data
Standard UART	$x110_B$	RxD = P7.4	TxD = P7.3	32 Bytes
Sync. Serial	1001_B	MRST = P2.4	MTSR = P2.3 SCLK = P2.5 SLS = P2.6	n Bytes; 1 … 65,280
MultiCAN	$x101_B$	RxDC0 = P2.6	TxDC0 = P2.5	8 × n Bytes

1) x means that the level on the corresponding pin is irrelevant.

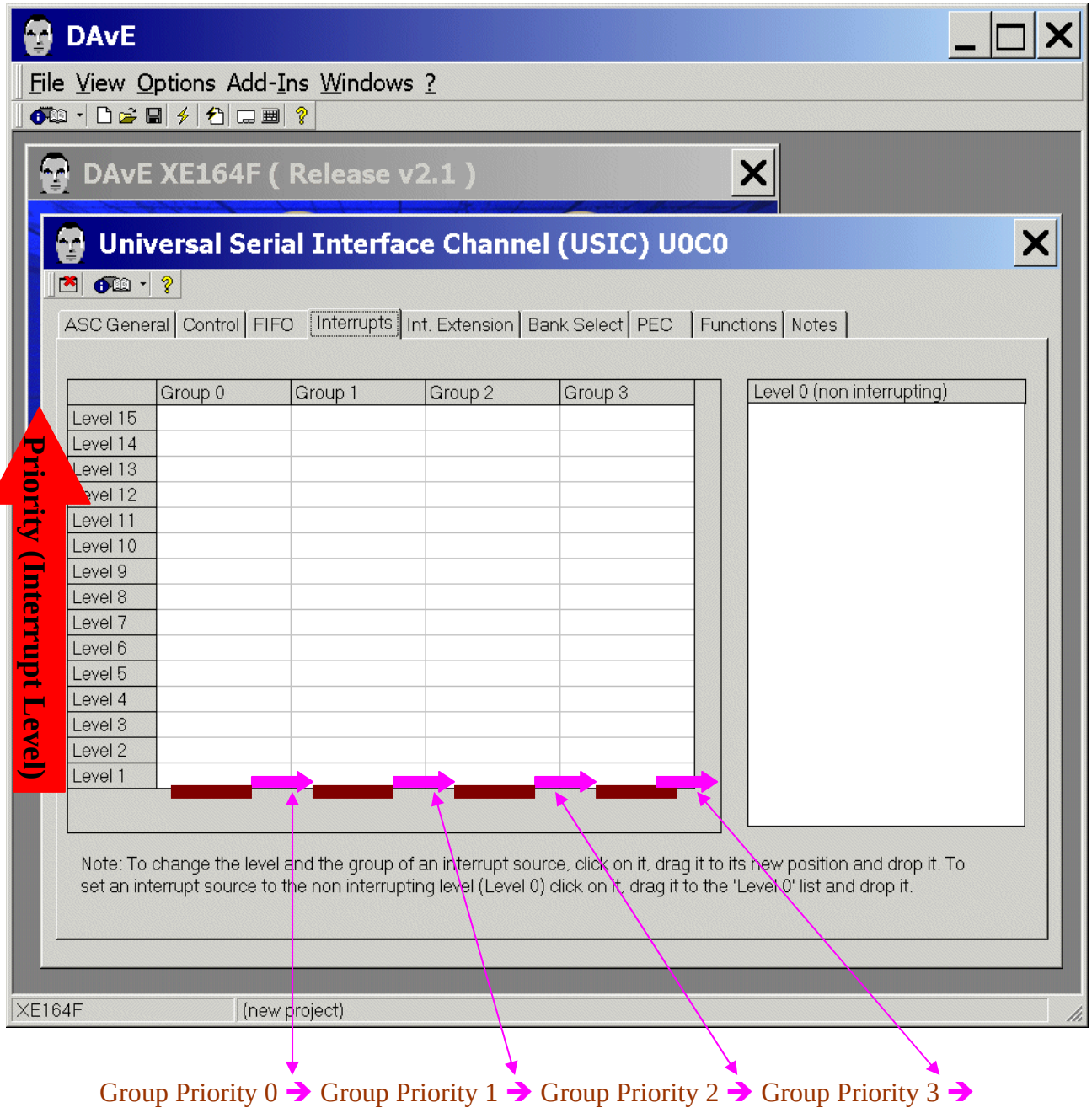
Control: (do nothing)



FIFO: (do nothing)



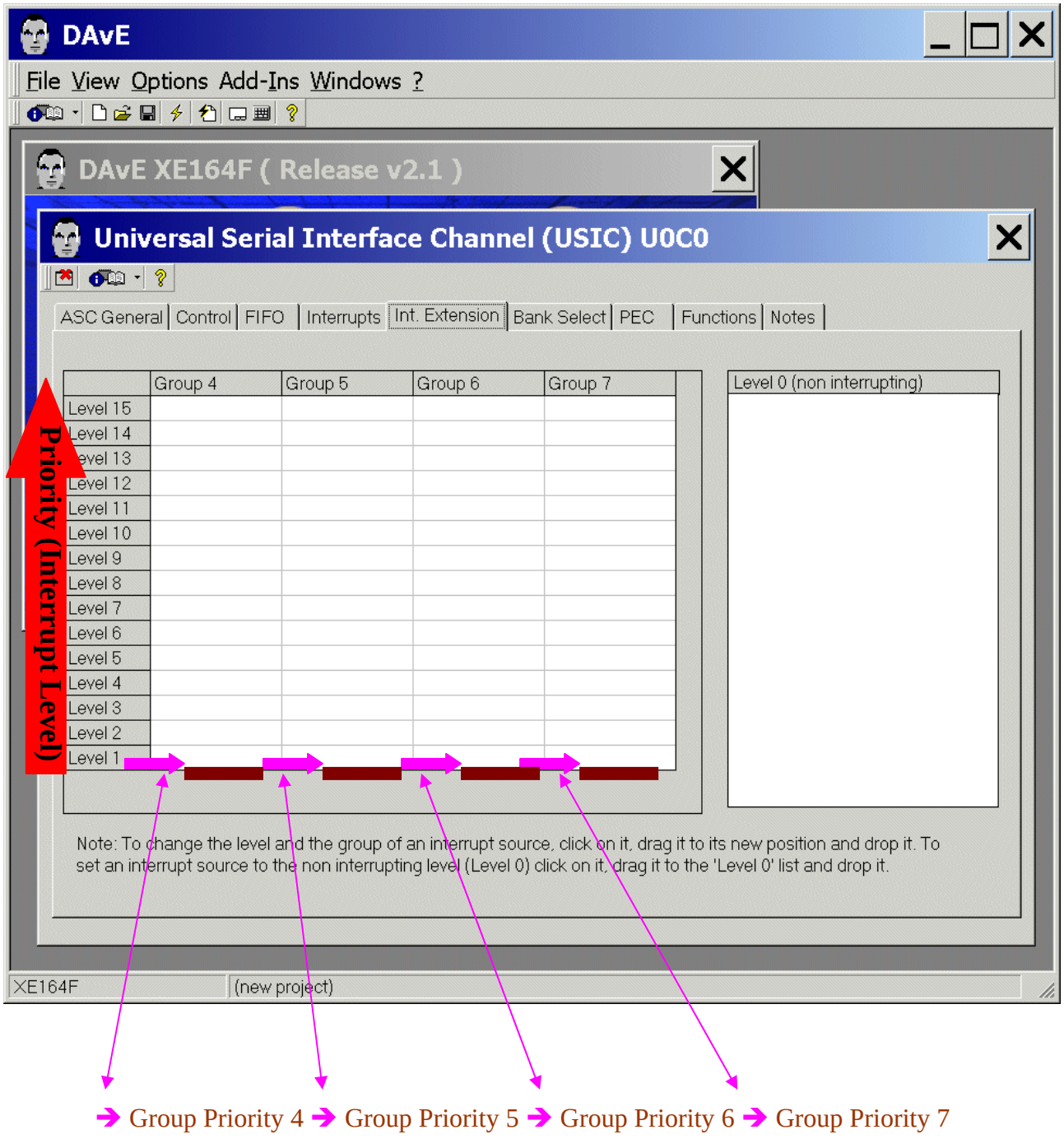
Interrupts: (do nothing)



Group Priority 0 → Group Priority 1 → Group Priority 2 → Group Priority 3 →

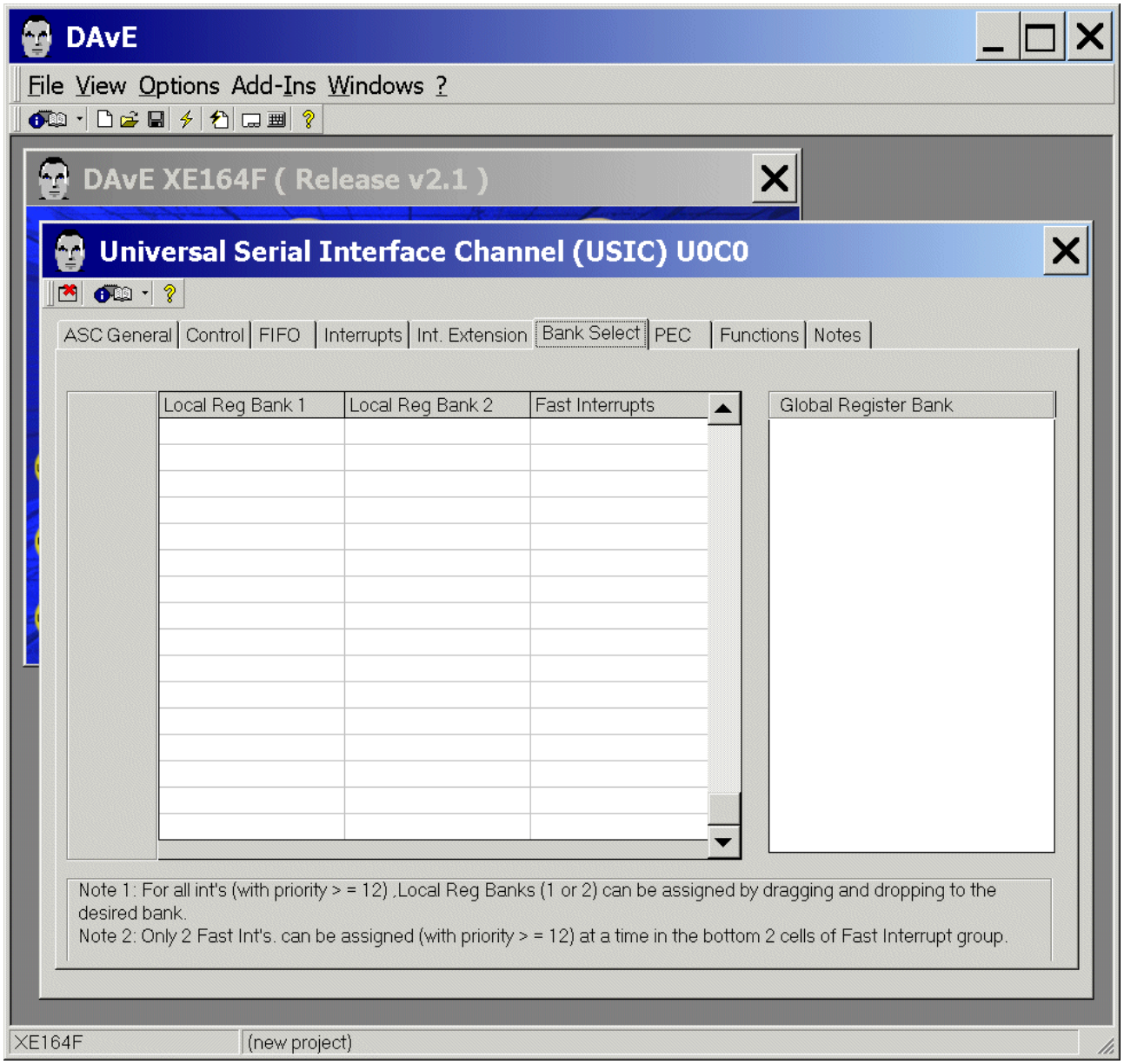
Note:
For the serial communication with a terminal program (e.g. Docklight, www.docklight.de) running on your host computer the myprintf function is used. The myprintf function uses Software-Polling-Mode therefore we do not need to configure any interrupts.

Int. Extension: (do nothing)

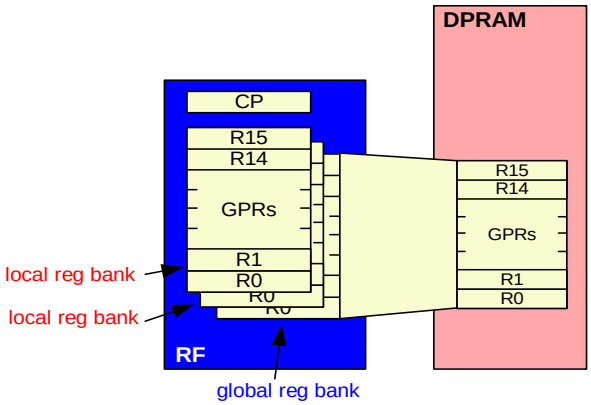


➔ Group Priority 4 ➔ Group Priority 5 ➔ Group Priority 6 ➔ Group Priority 7

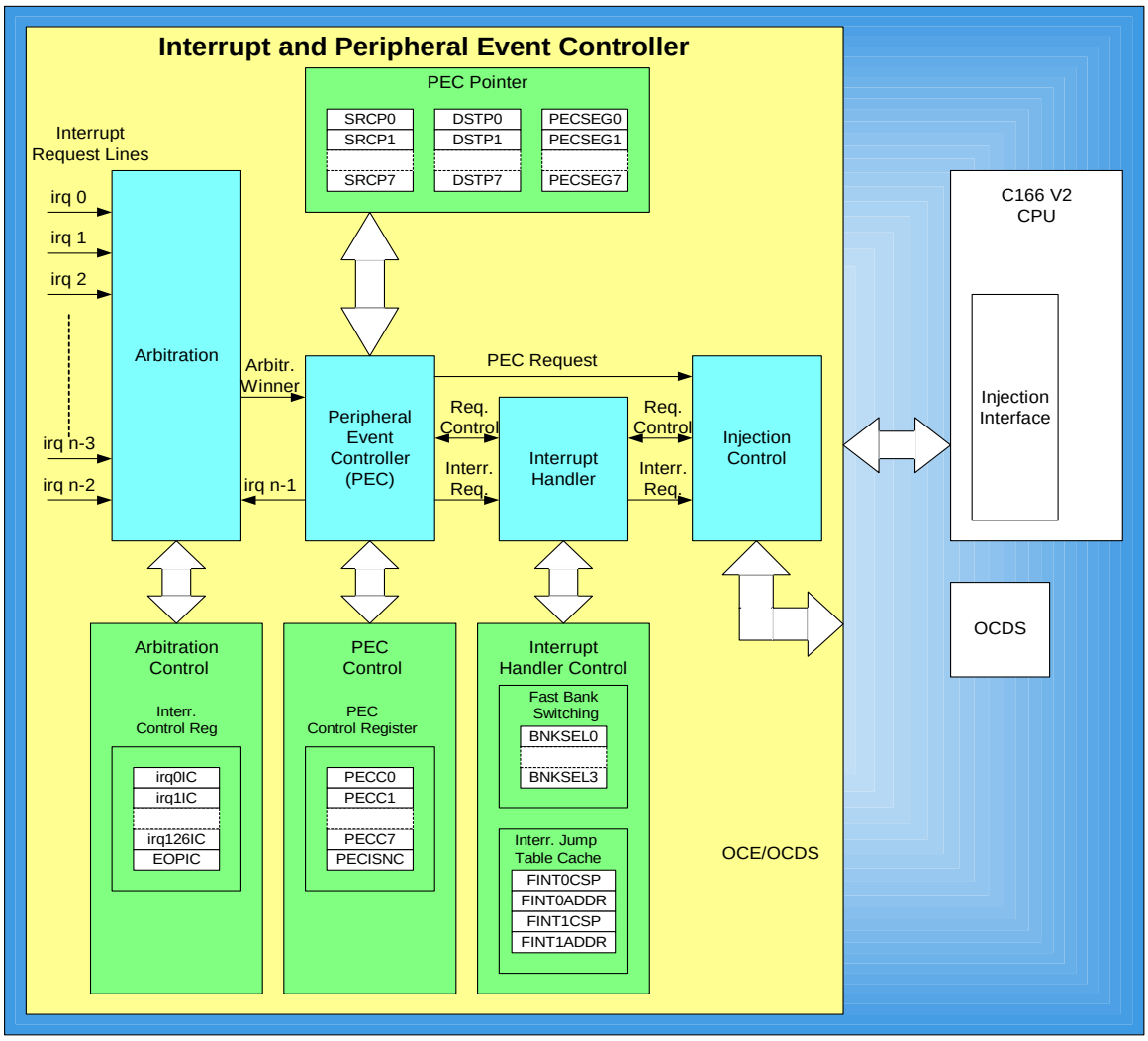
Bank Select: (do nothing)



Note:
For our hello world program the 2 local register banks are not needed.



Additional information: local register banks and fast interrupts (page 1/2):



Additional information: local register banks and fast interrupts (page 1/2):

void CC1_viCC1 (void) interrupt CC1_CC1INT { }

void CC1_viCC1 (void) interrupt CC1_CC1INT using REGBANK4711 { }

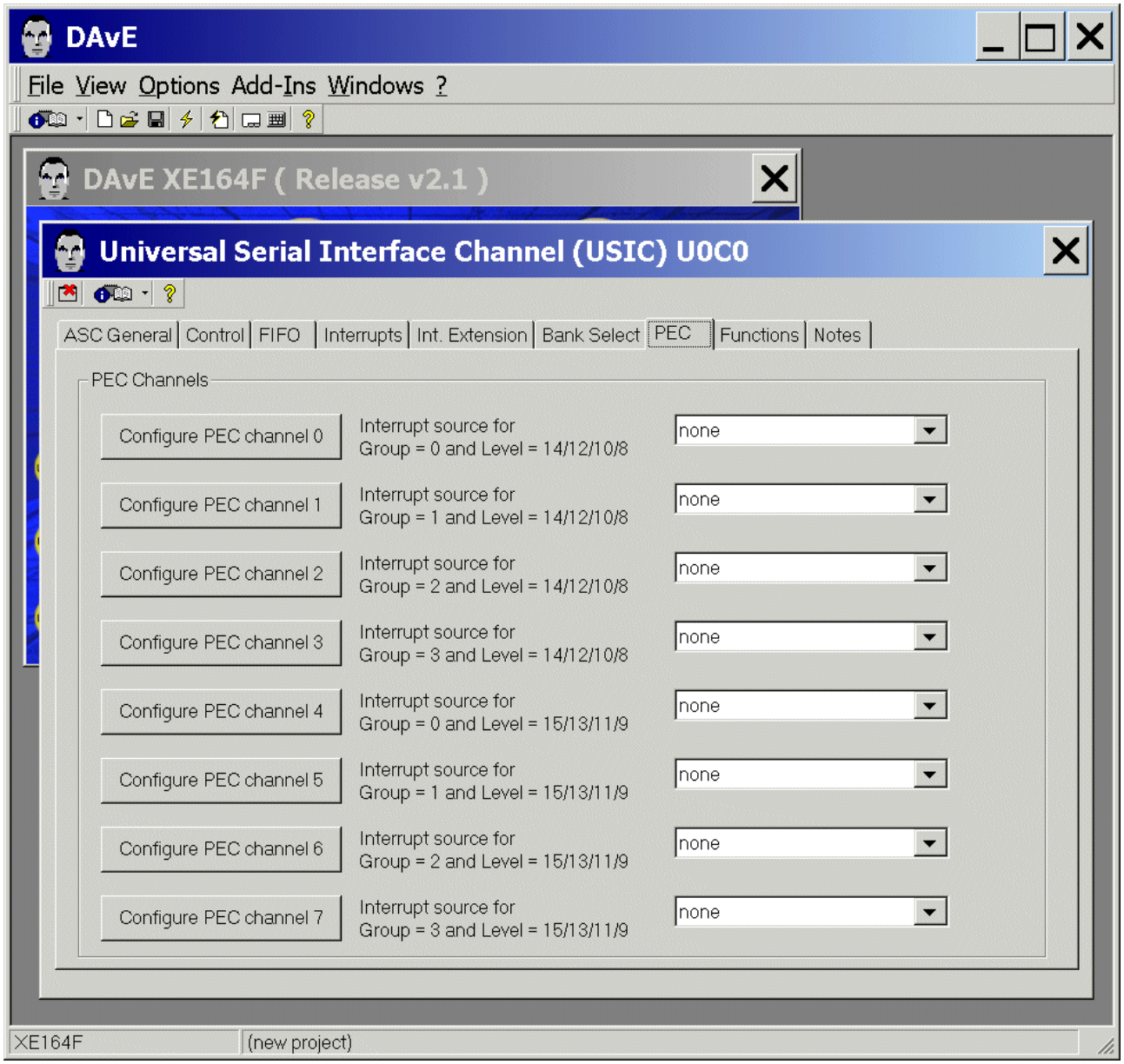
void CC1_viCC1 (void) interrupt CC1_CC1INT using _FAST_ABANK1_ { }

void CC1_viCC1 (void) interrupt MickeyMouse=CACHED { }

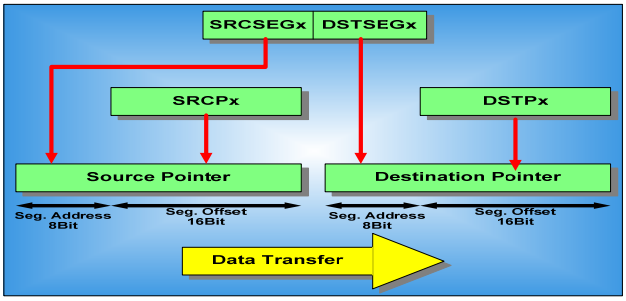
void CC1_viCC1 (void) interrupt MickeyMouse=CACHED using REGBANK4711 { }

void CC1_viCC1 (void) interrupt MickeyMouse=CACHED using _FAST_ABANK1_ { }

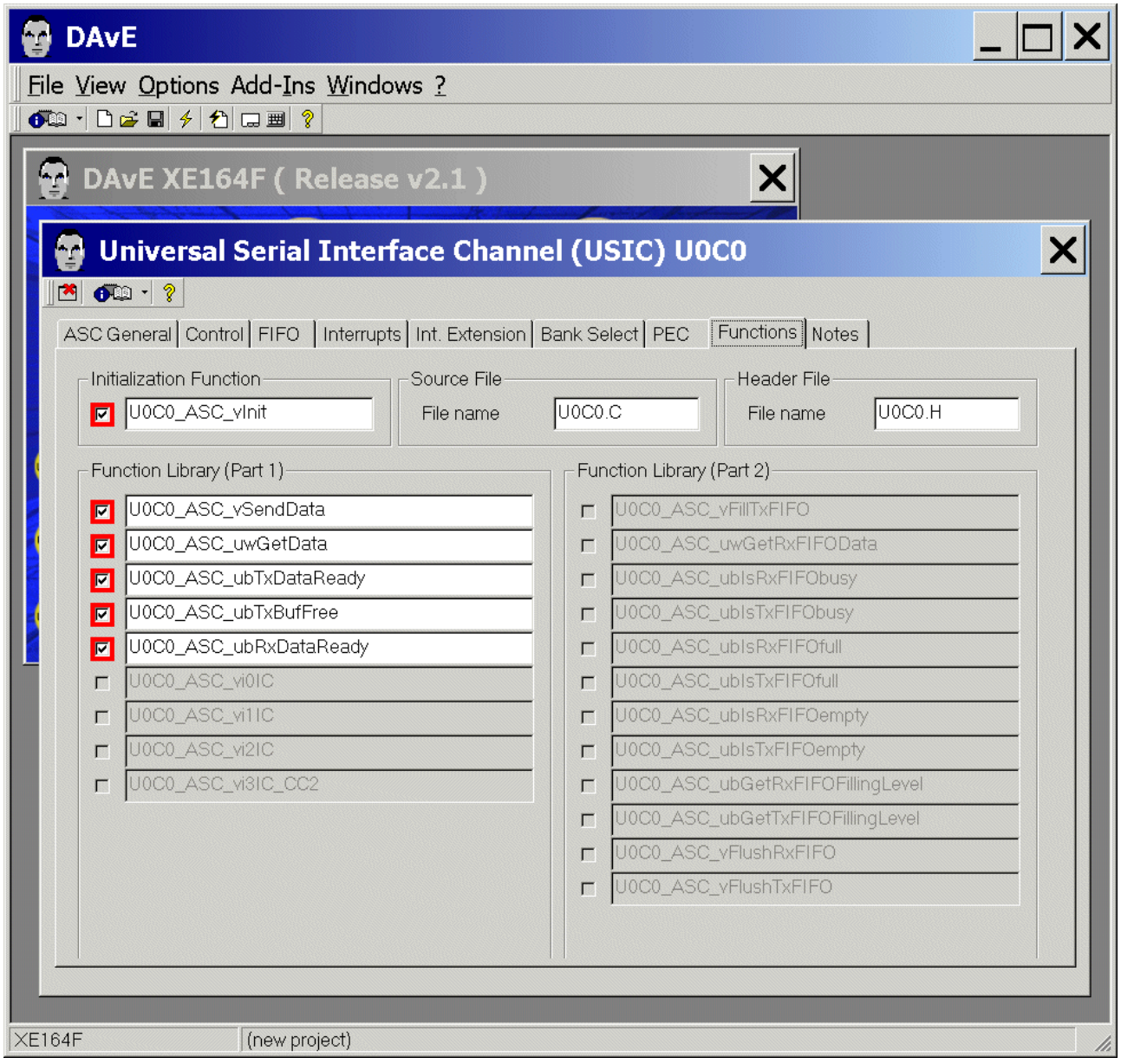
PEC: (do nothing)



Note:
For our hello world program the 8 PEC Channels are not needed.

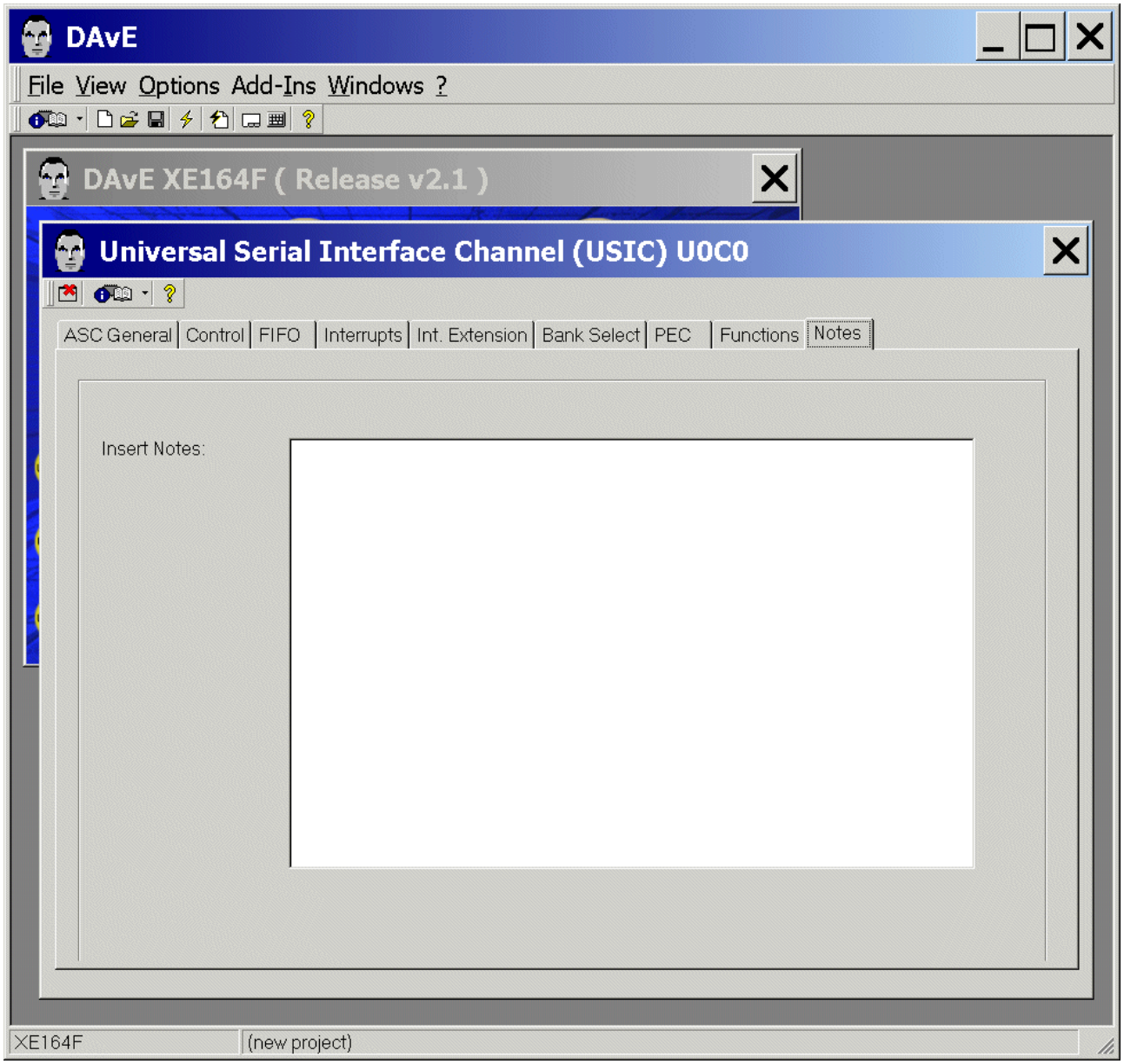


Functions: Initialization Function: tick/check ☑ U0C0_ASC_vInit
Functions: Function Library (Part 1): tick ☑ U0C0_ASC_vSendData
Functions: Function Library (Part 1): tick ☑ U0C0_ASC_uwGetData
Functions: Function Library (Part 1): tick ☑ U0C0_ASC_ubTxDataReady
Functions: Function Library (Part 1): tick ☑ U0C0_ASC_ubTxBufFree
Functions: Function Library (Part 1): tick ☑ U0C0_ASC_ubRxDataReady



Note:
You can change function names (e.g. U0C0_ASC_vInit) and file names (e.g. U0C0.C, U0C0.H) anytime.

Notes: (do nothing)

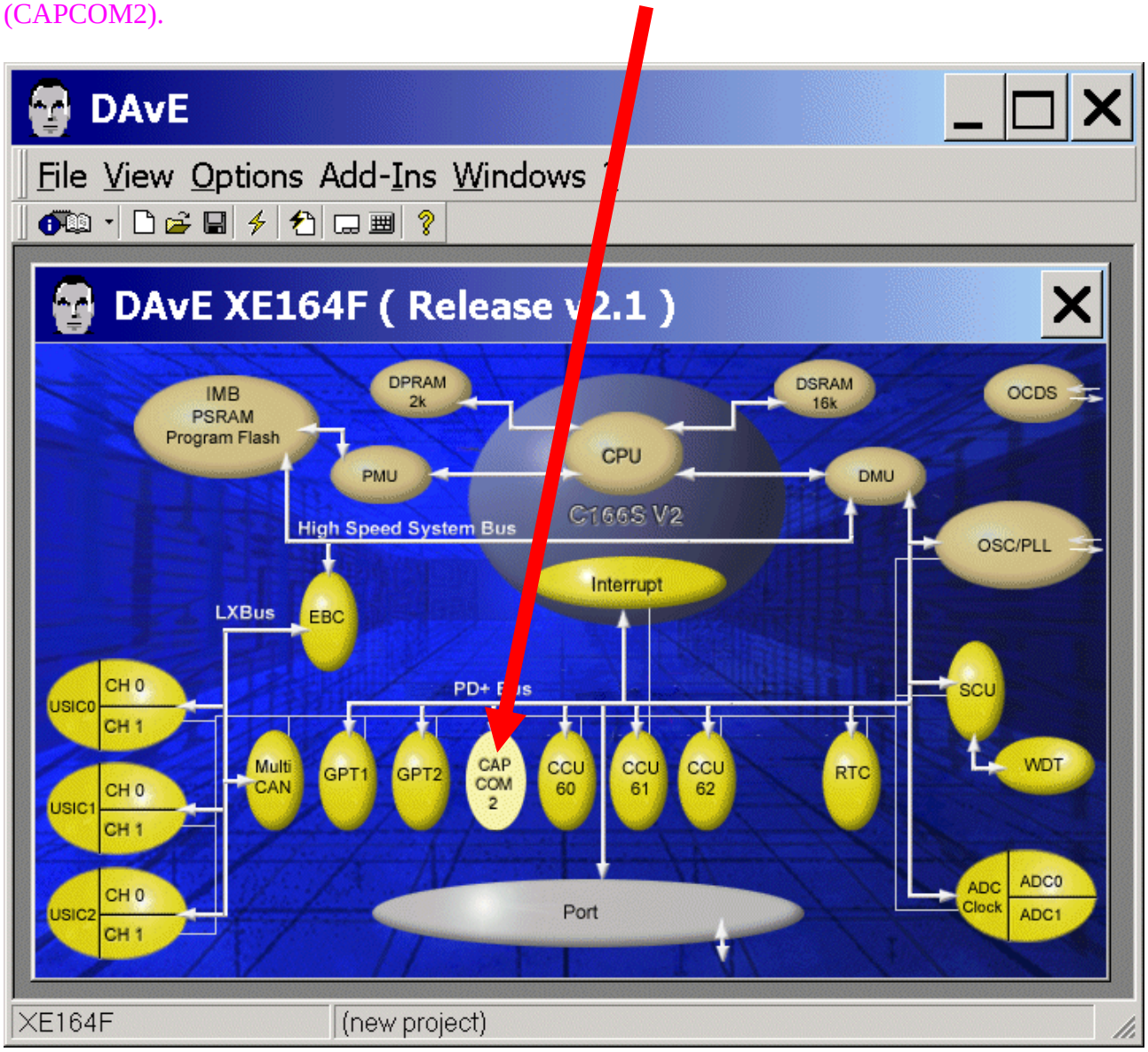


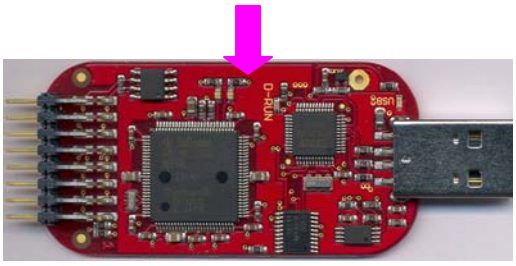
Note:
Notes: Insert Notes: If you wish, you can insert your comments here.

Exit and Save this dialog now by clicking the close button.

Configure Timer T7 in the CAPCOM 2 module:

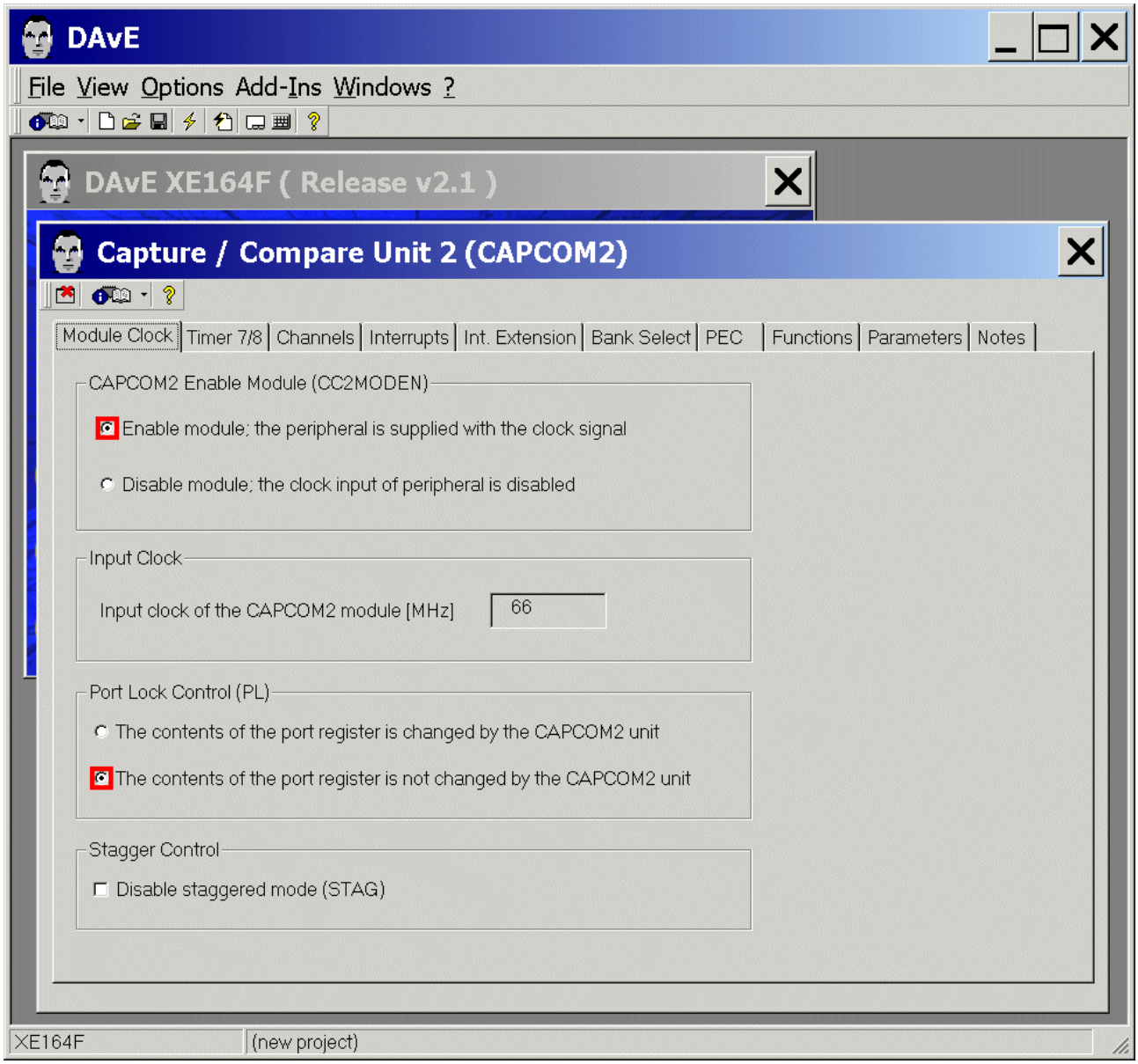
The configuration window/dialog can be opened by clicking the specific block/module (CAPCOM2).



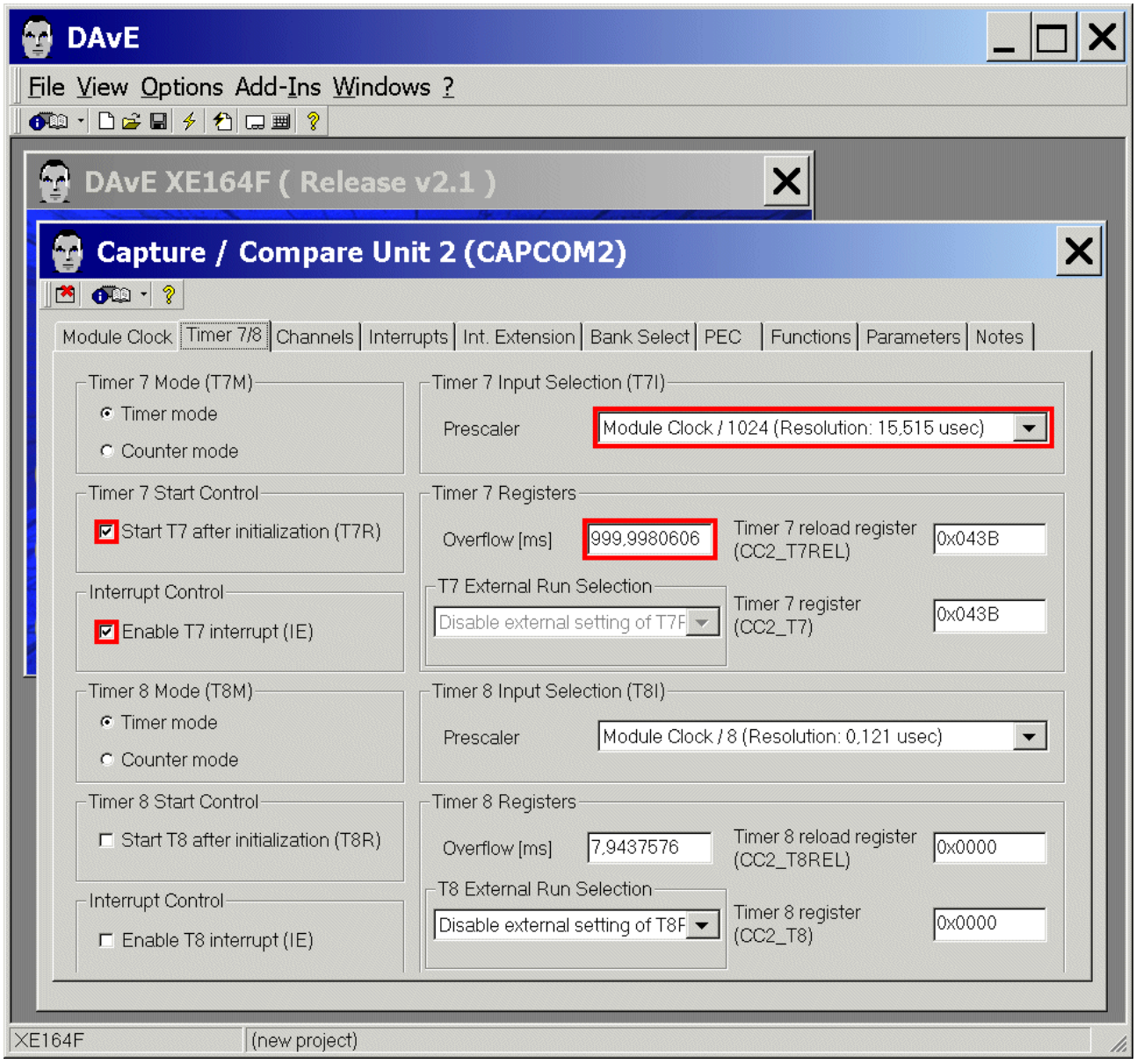


Note:
The LED on IO_Port_2.7 will be blinking (after Power-On or if selected in the main menu) with a frequency of about 1 second (done in the Timer_7-Interrupt-Service-Routine). Therefore we have to configure Timer_7.

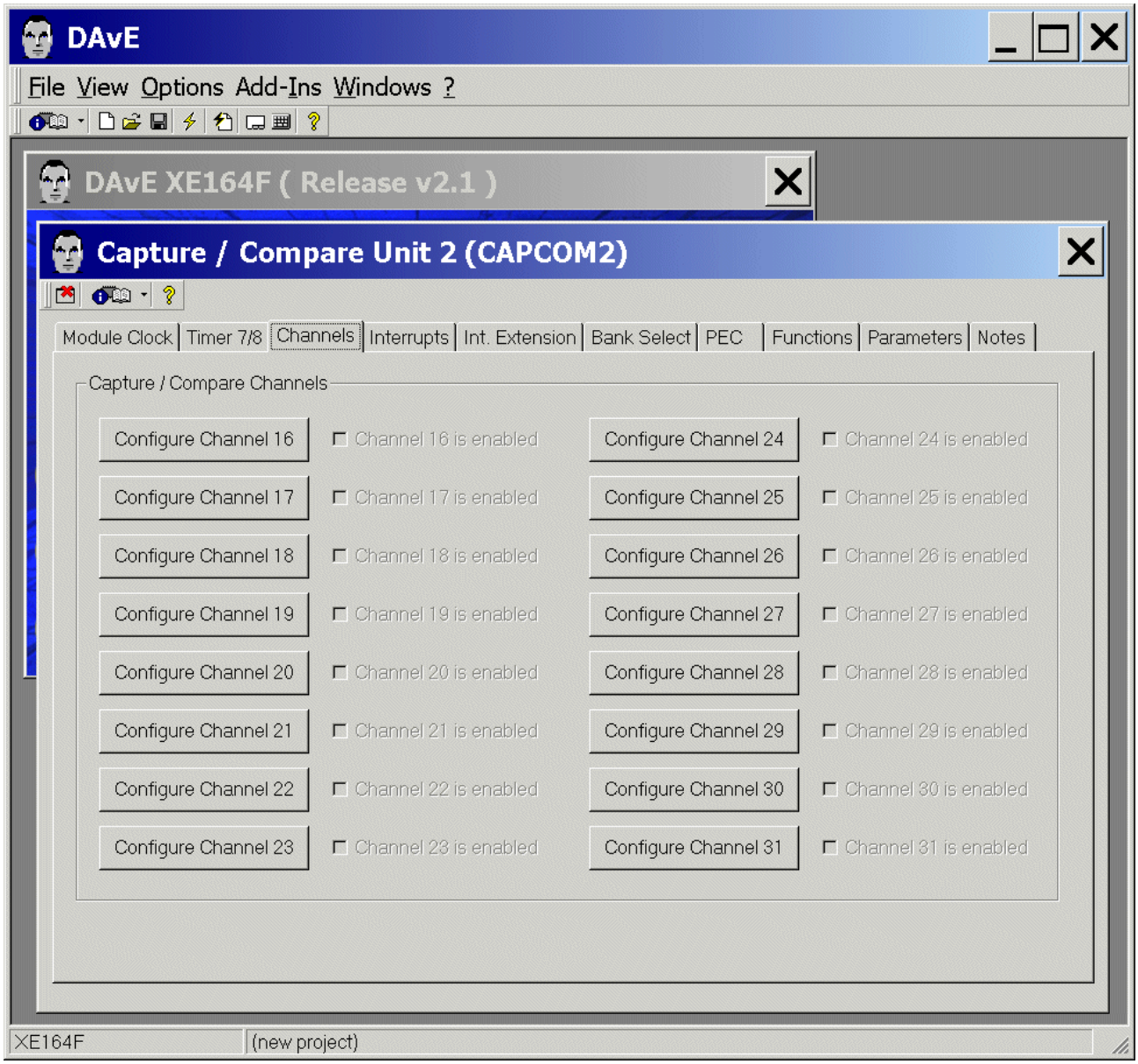
Module Clock: CAPCOM2 Enable Module: click ⊙ Enable module
Module Clock: Port Lock Control:
click ⊙ The contents of the port register is not changed by the CAPCOM2 unit



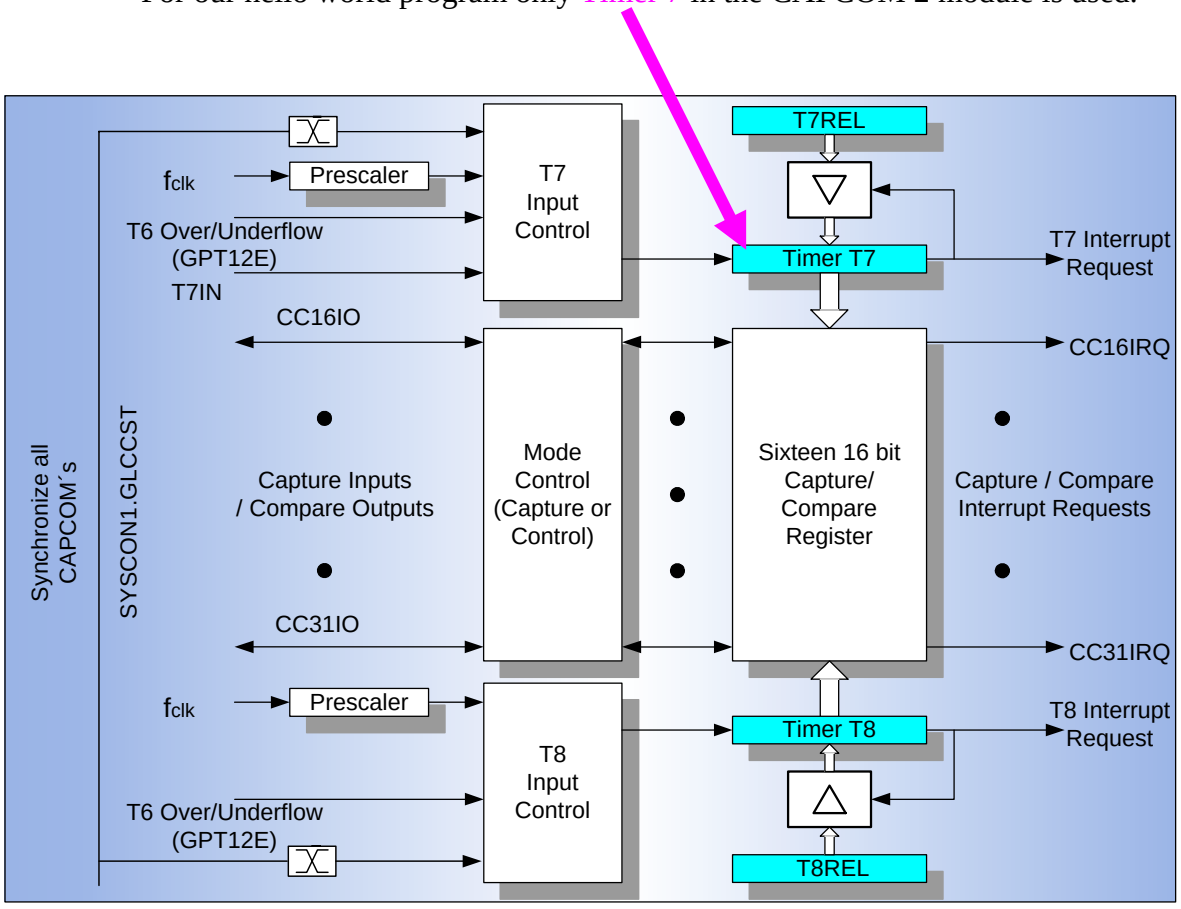
Timer 7/8: Timer 7 Start Control: tick ✓ Start T7 after initialization (T7R)
Timer 7/8: Interrupt Control: tick ✓ Enable T7 interrupt (IE)
Timer 7/8: Timer 7 Input Selection (T7I): Prescaler: choose Module Clock/1024
Timer 7/8: Timer 7 Registers: Overflow [s]: insert 1 <ENTER>



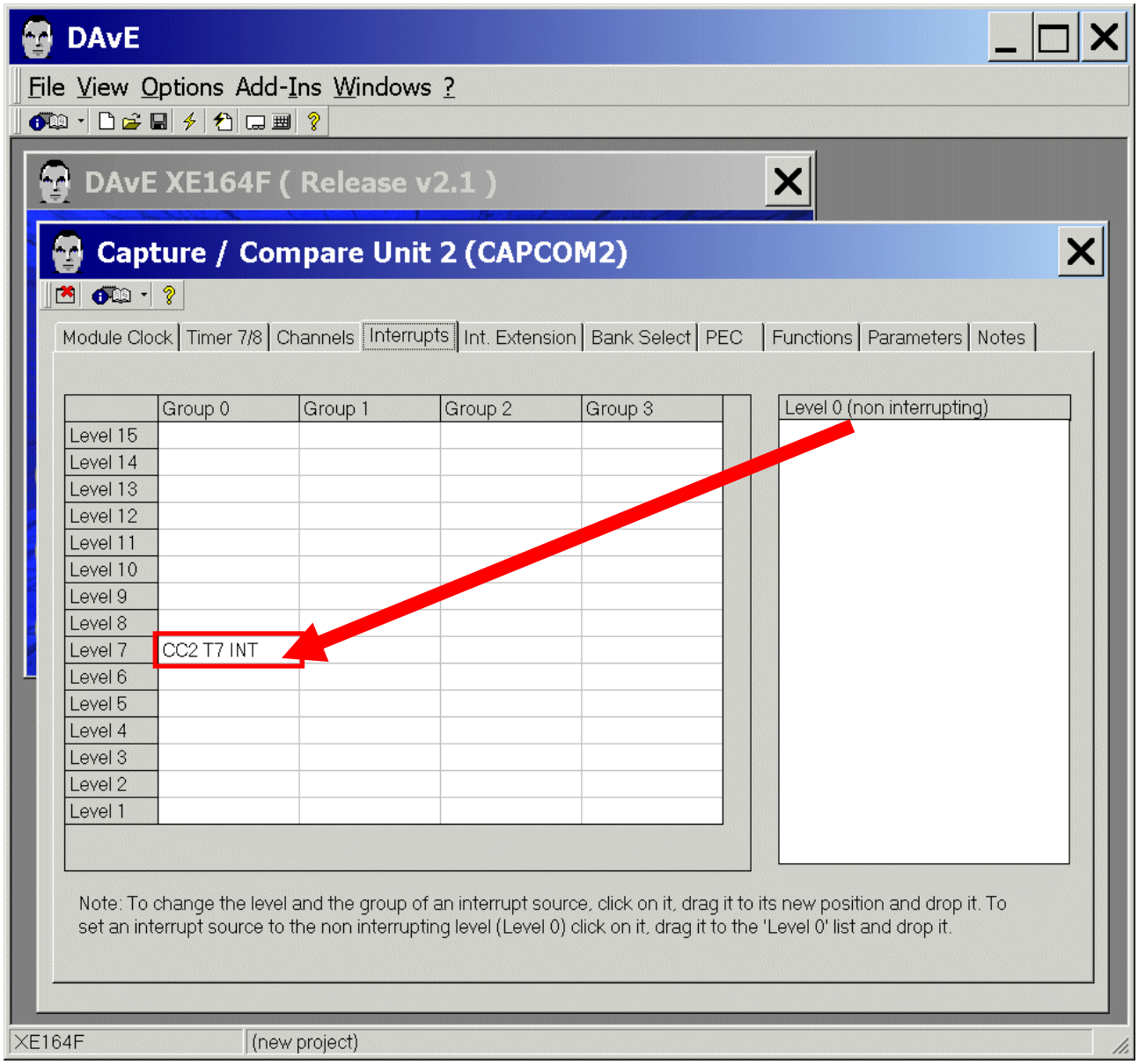
Channels: (do nothing)



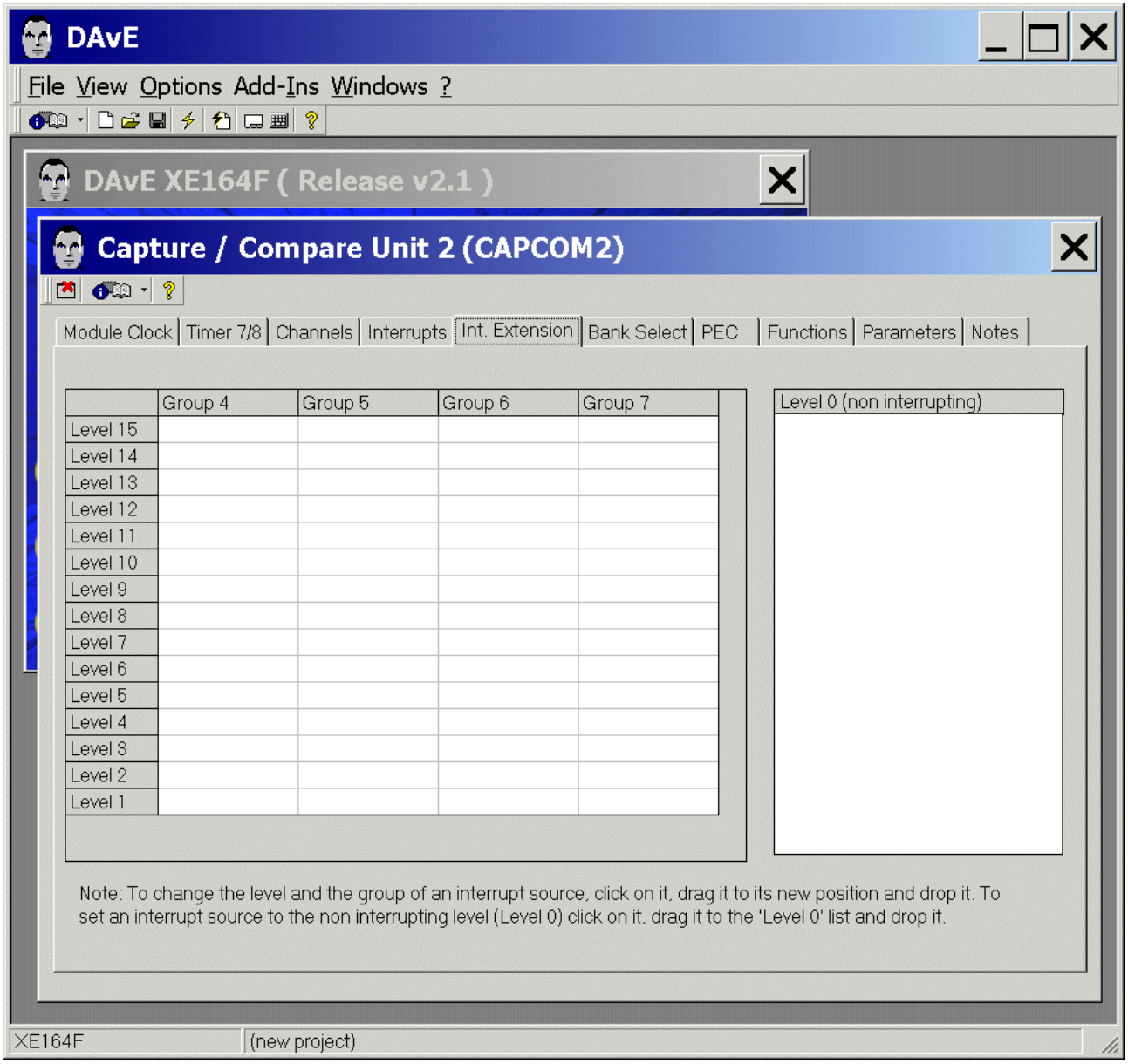
Note:
For our hello world program only Timer 7 in the CAPCOM 2 module is used.



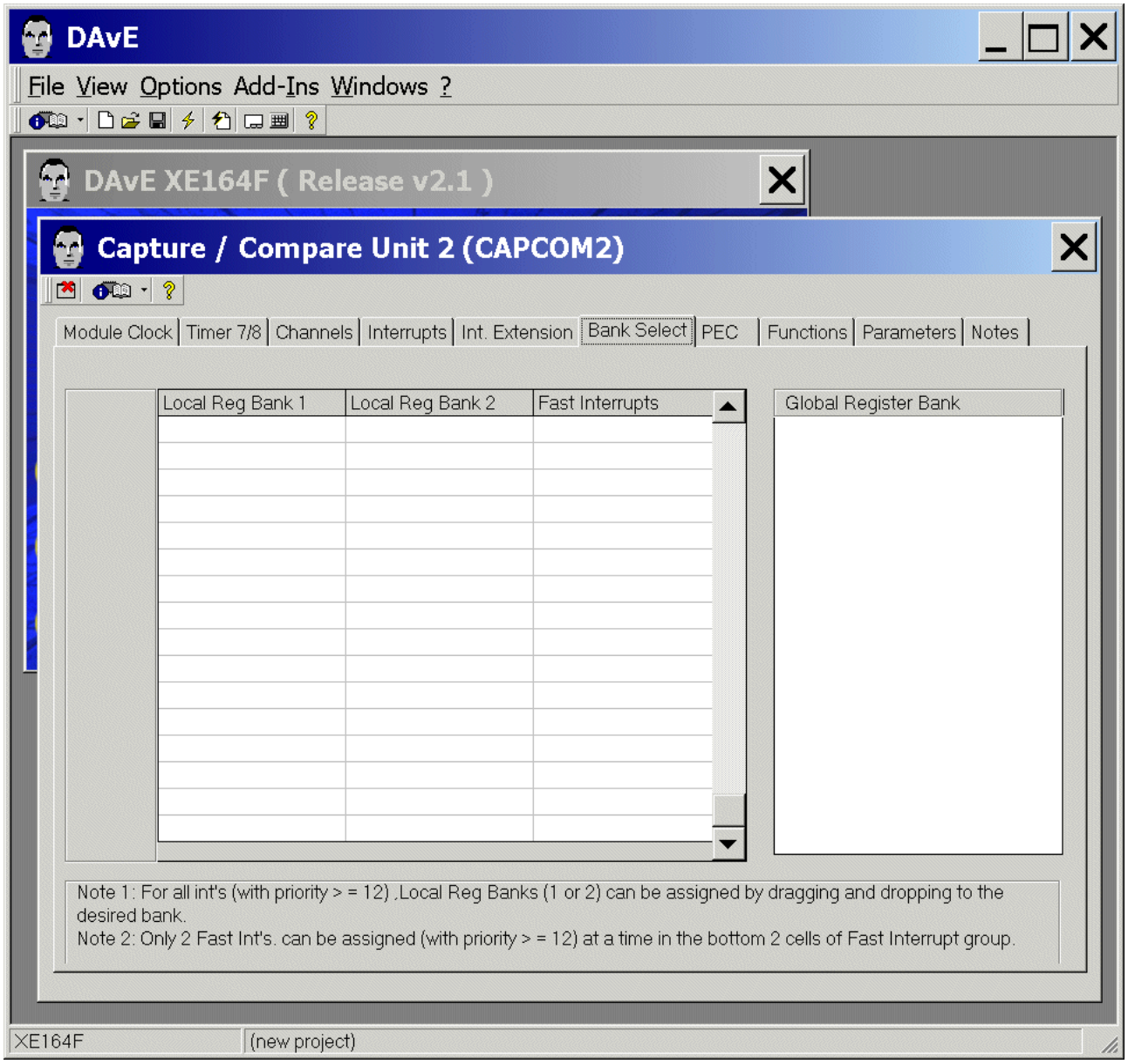
Interrupts: drag and drop the CC2 T7 INT to Interrupt Level 7, Group 0



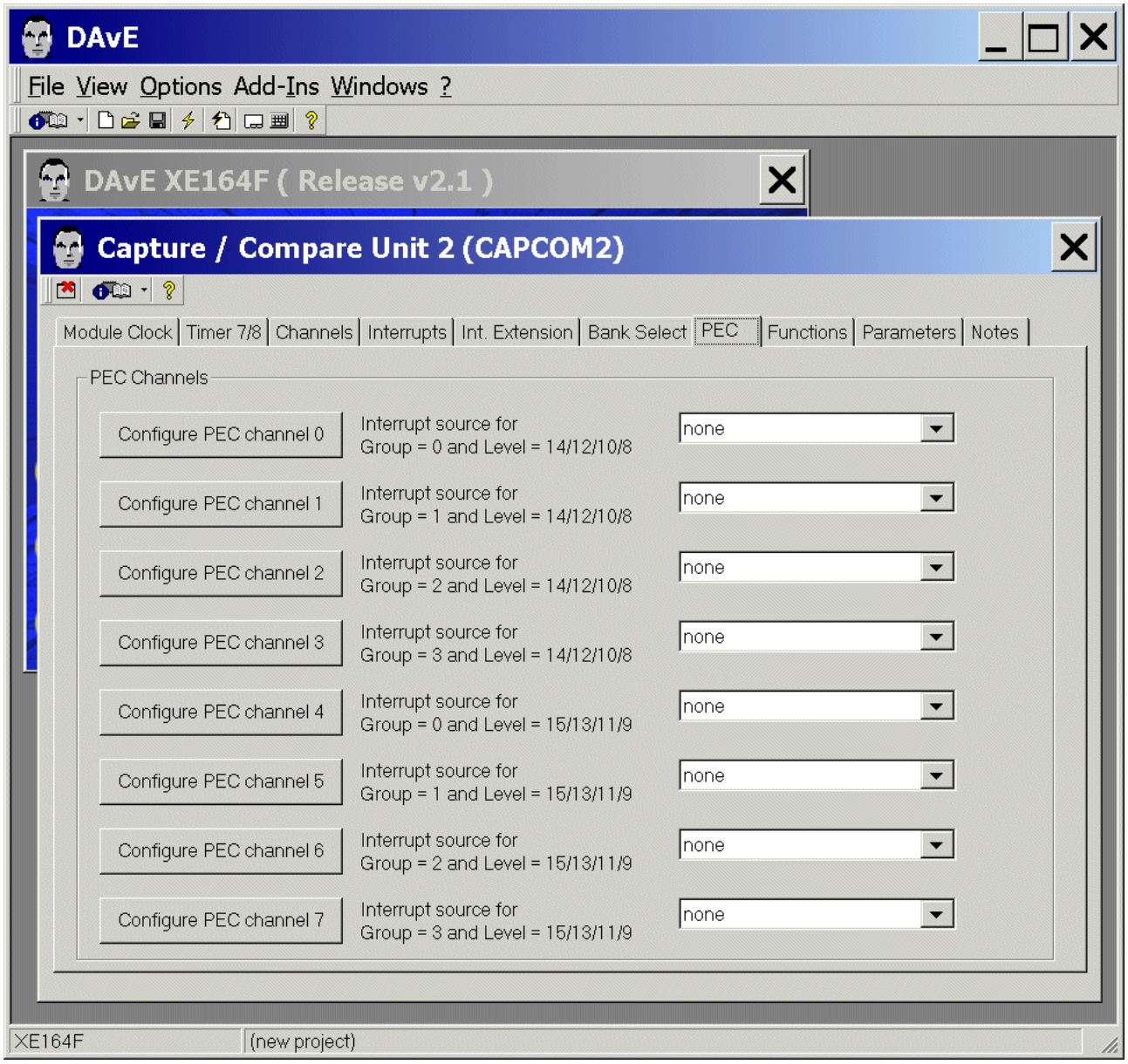
Int. Extension: (do nothing)



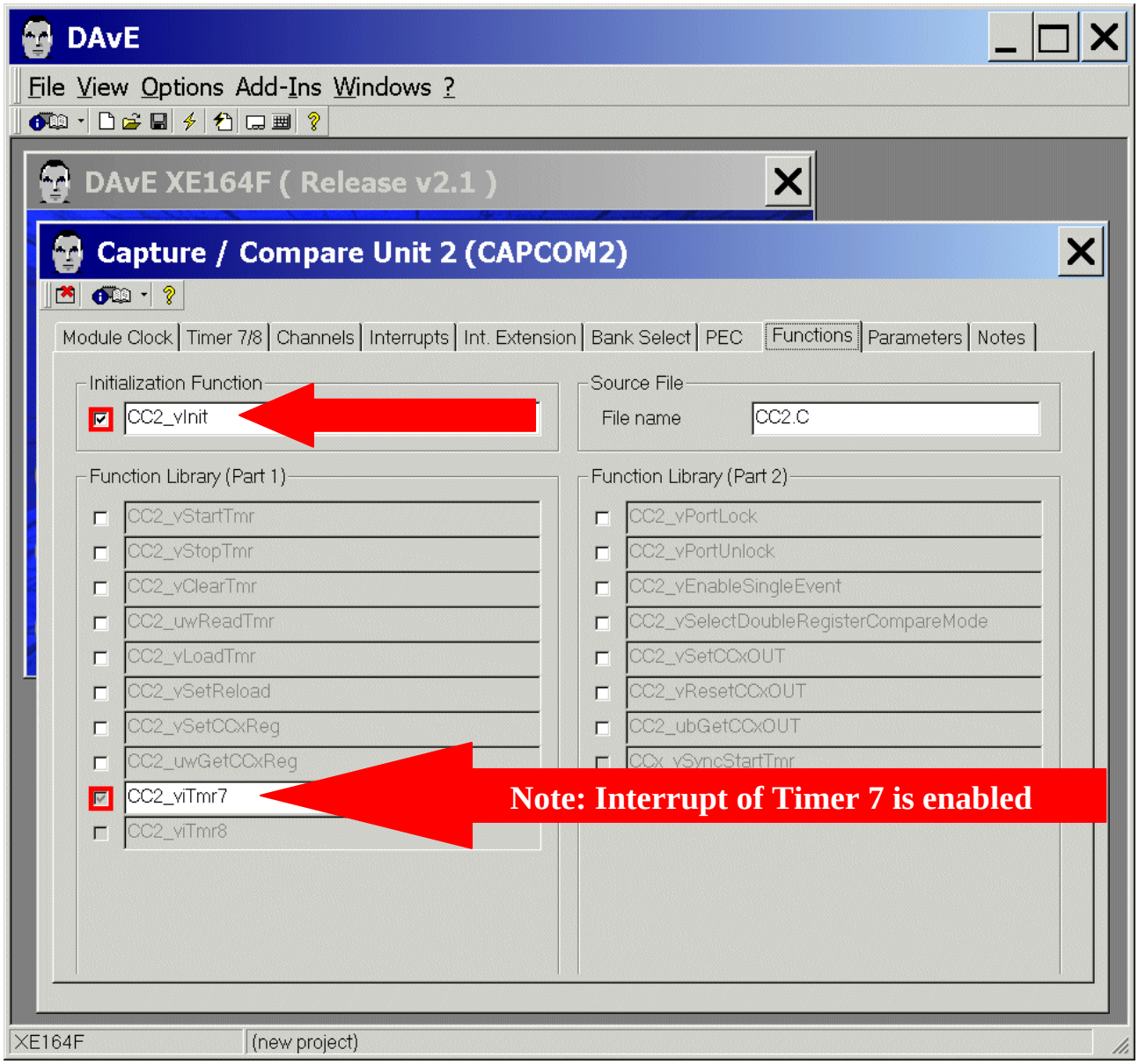
Bank Select: (do nothing)



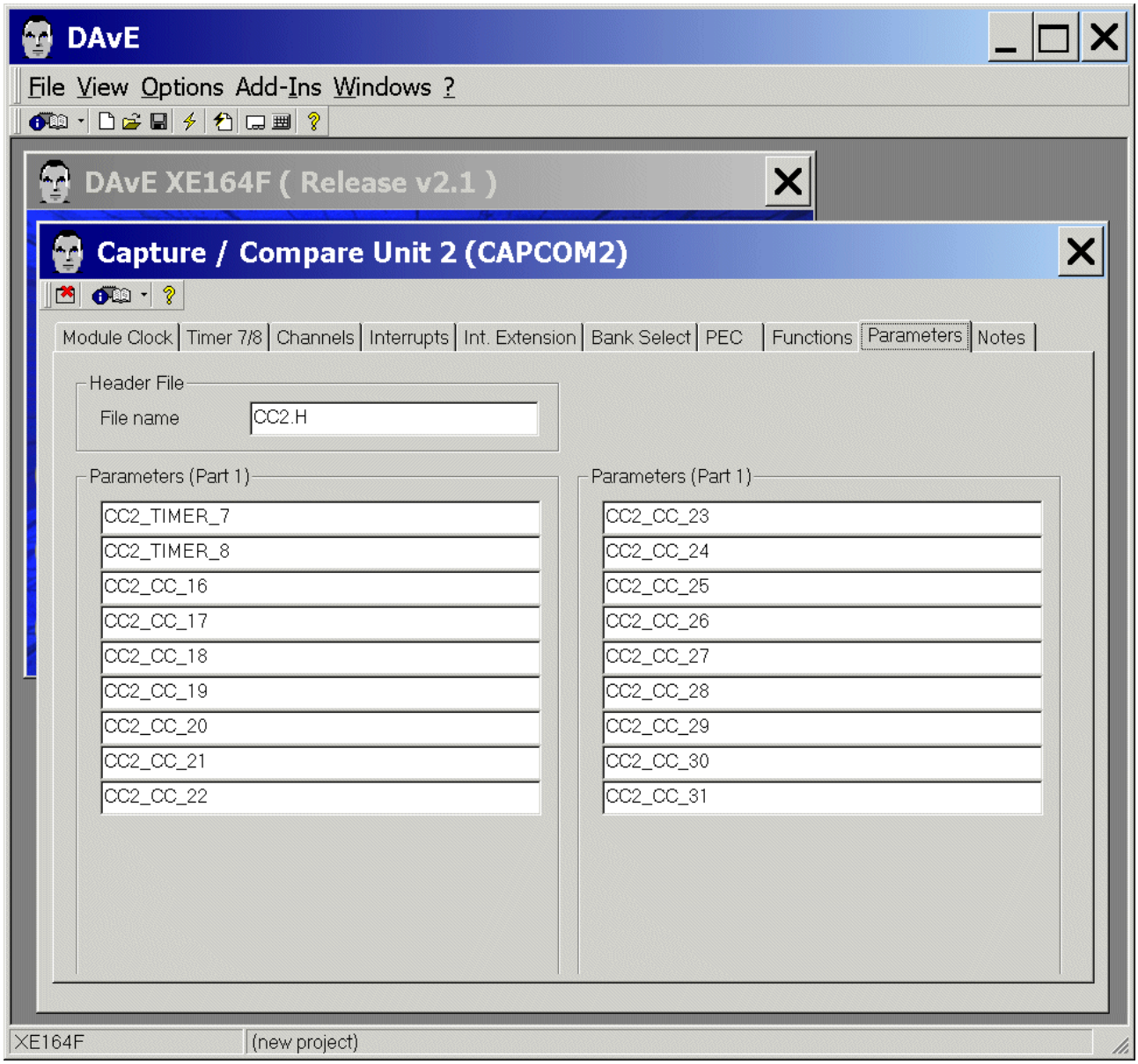
PEC: (do nothing)



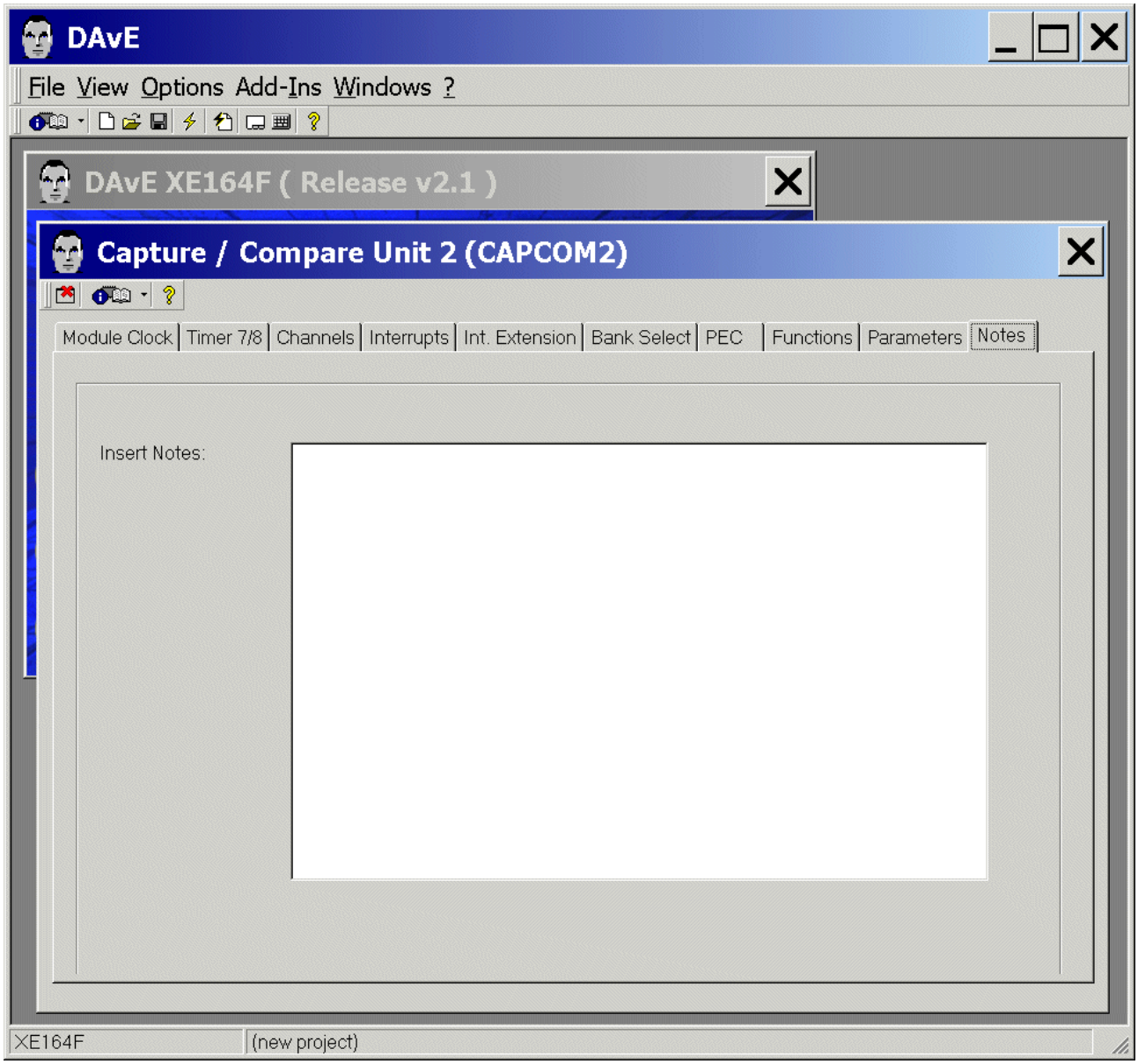
Functions: Initialization Function: tick/check ☑ CC2_vInit



Parameters: (do nothing)



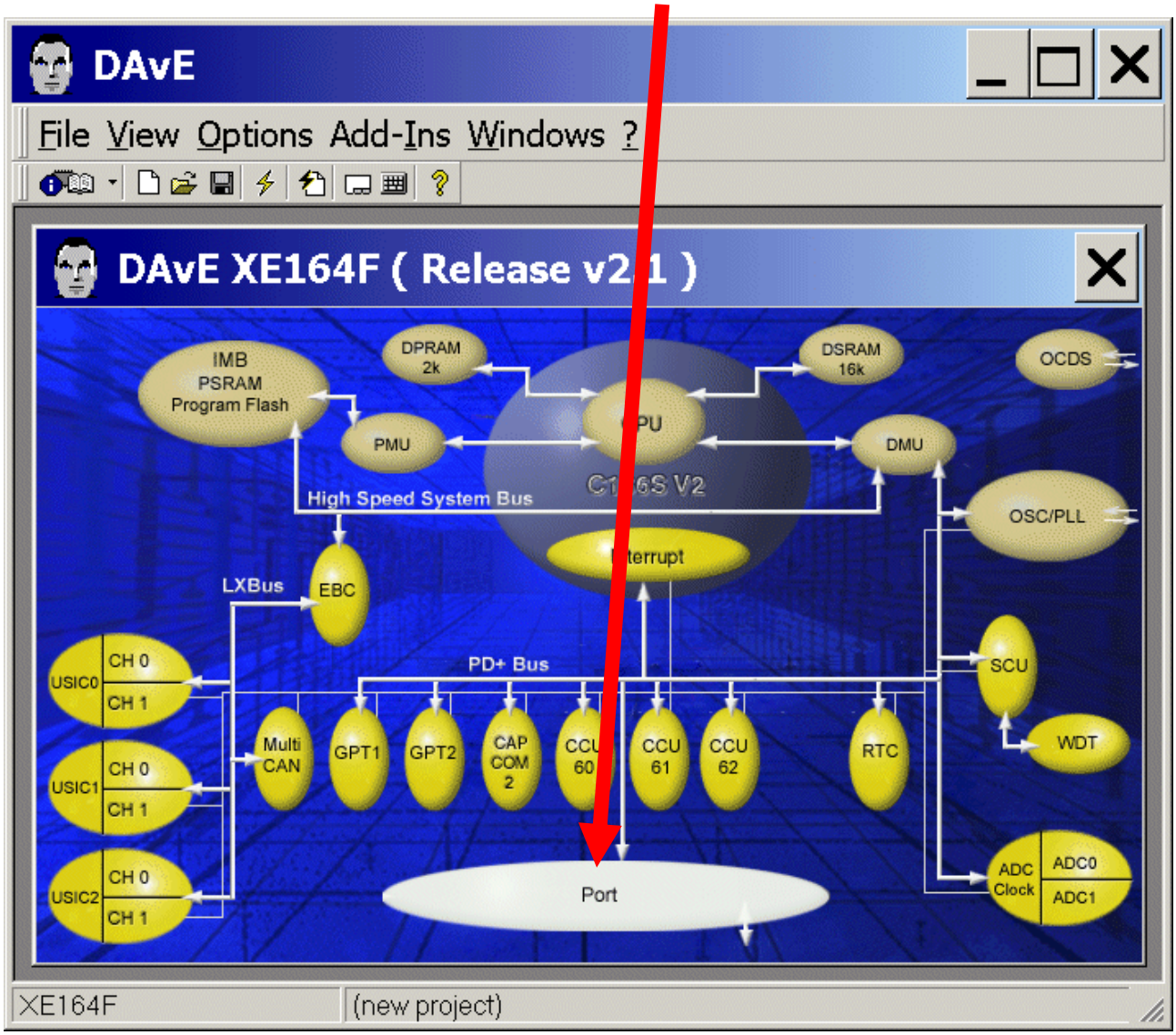
Notes: Insert Notes: If you wish, you can insert your comments here.



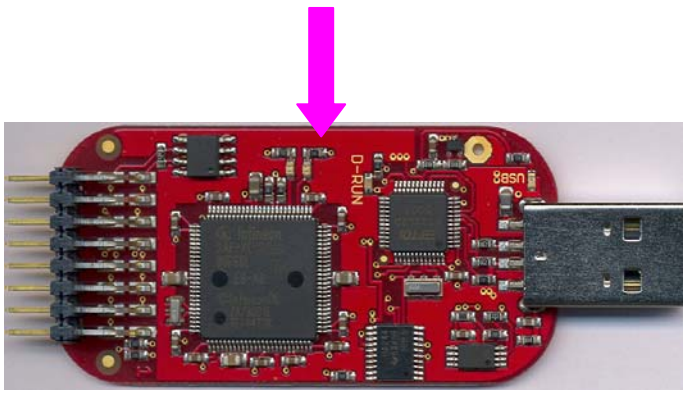
Exit and Save this dialog now by clicking the close button.

Configure Port 2 Pin 7 to Output :

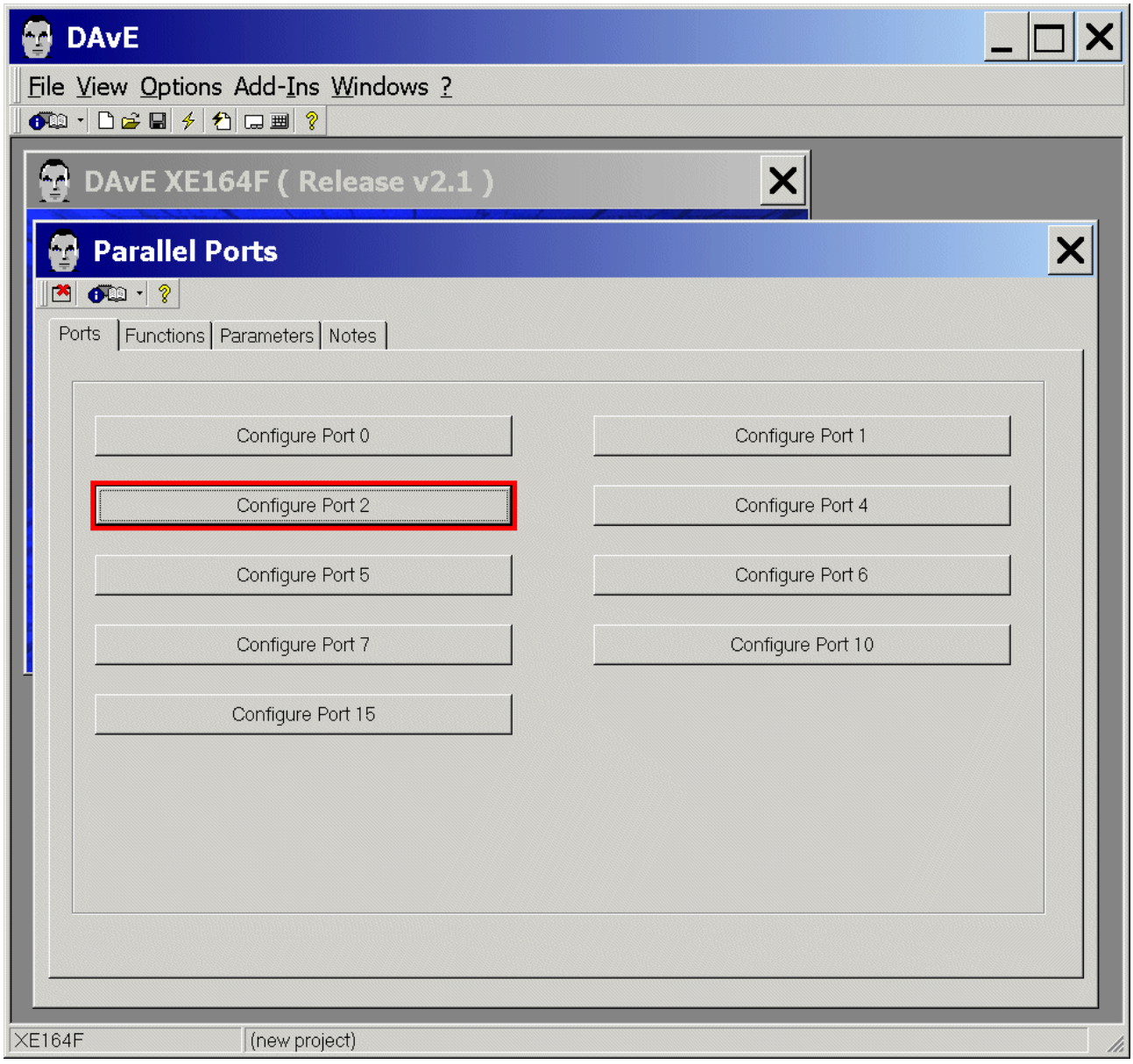
The configuration window/dialog can be opened by clicking the specific block/module (Port).



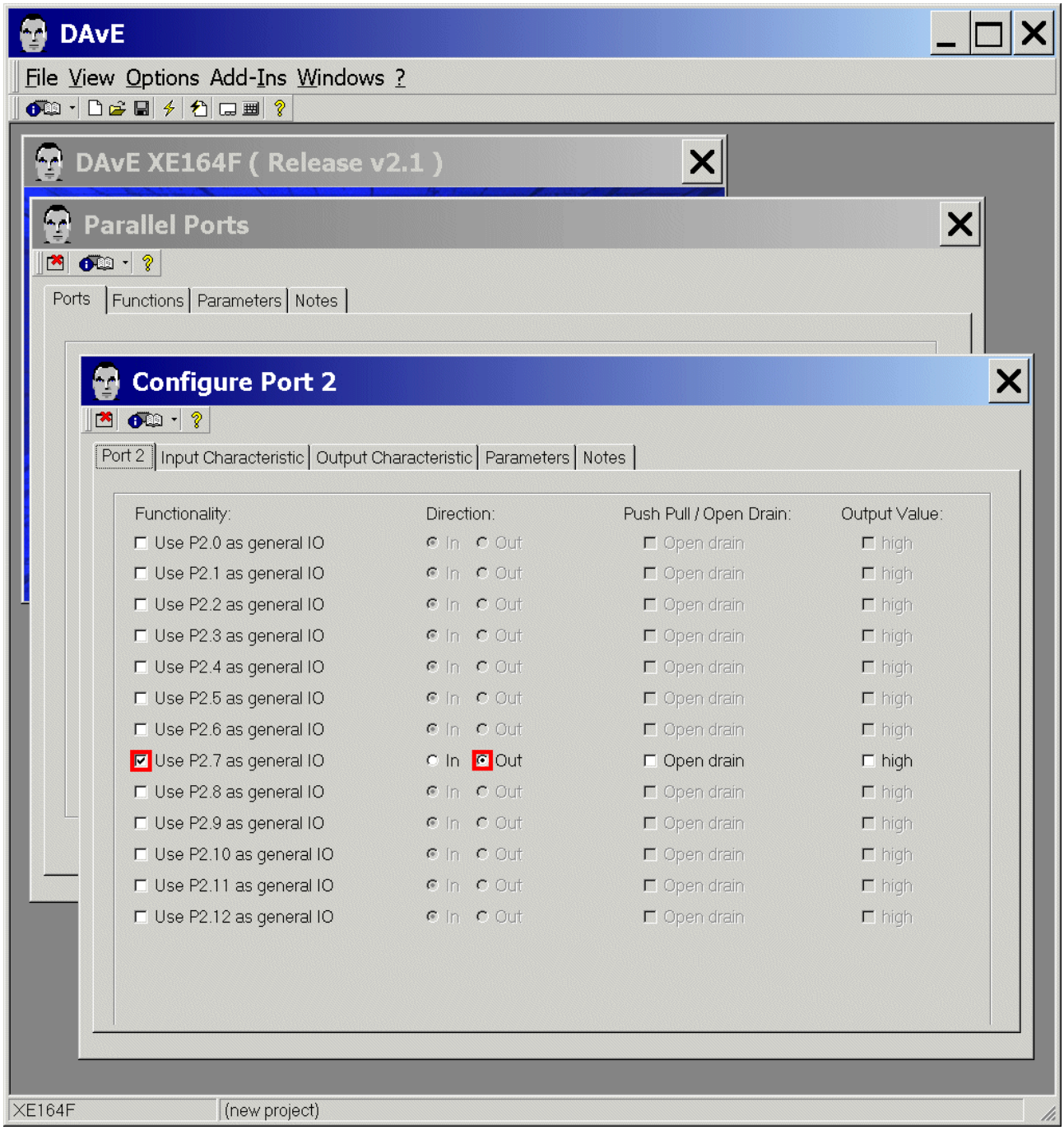
Note:
The LED is connected to IO_Port_2.7



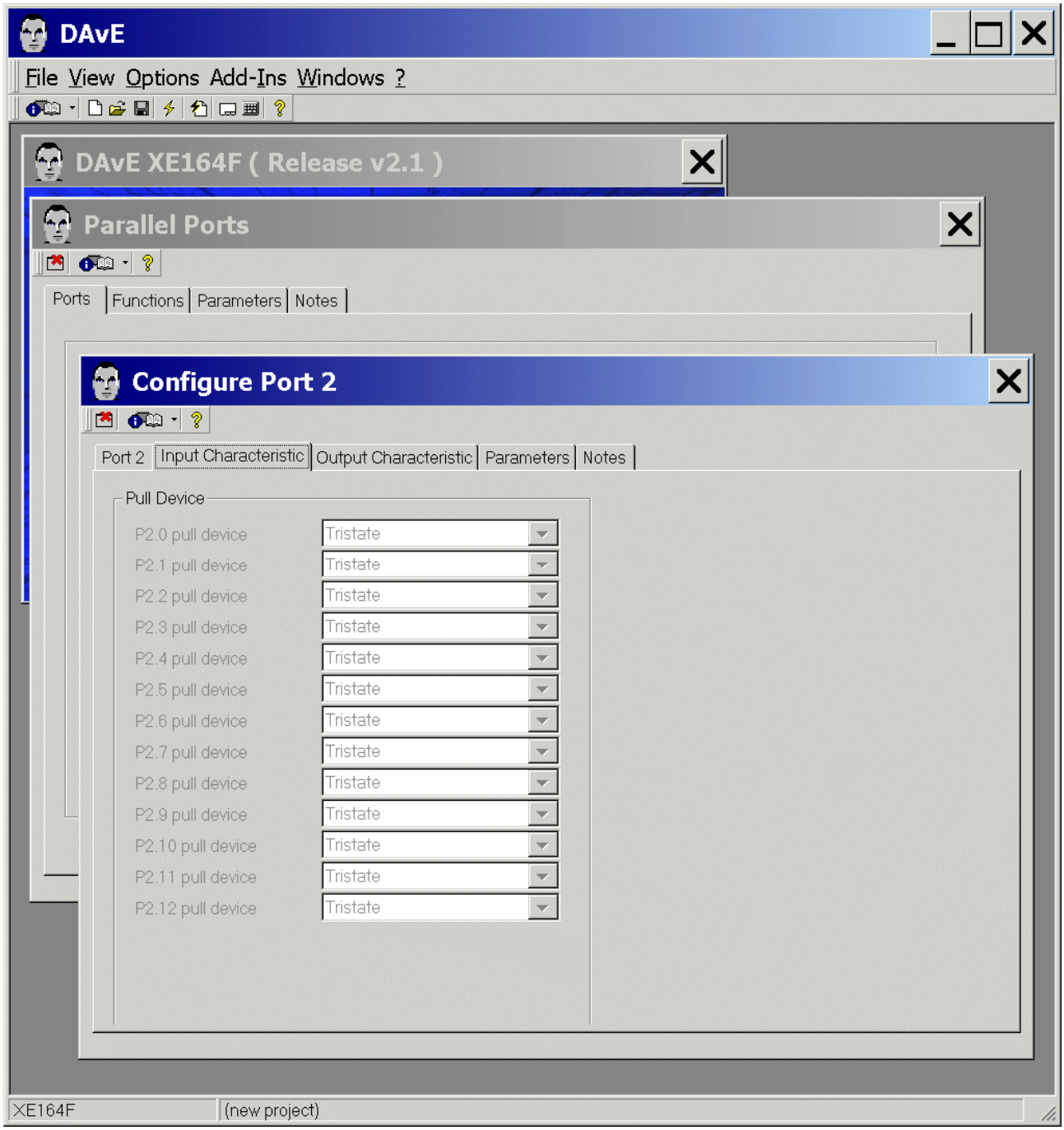
Ports: click “Configure Port 2”



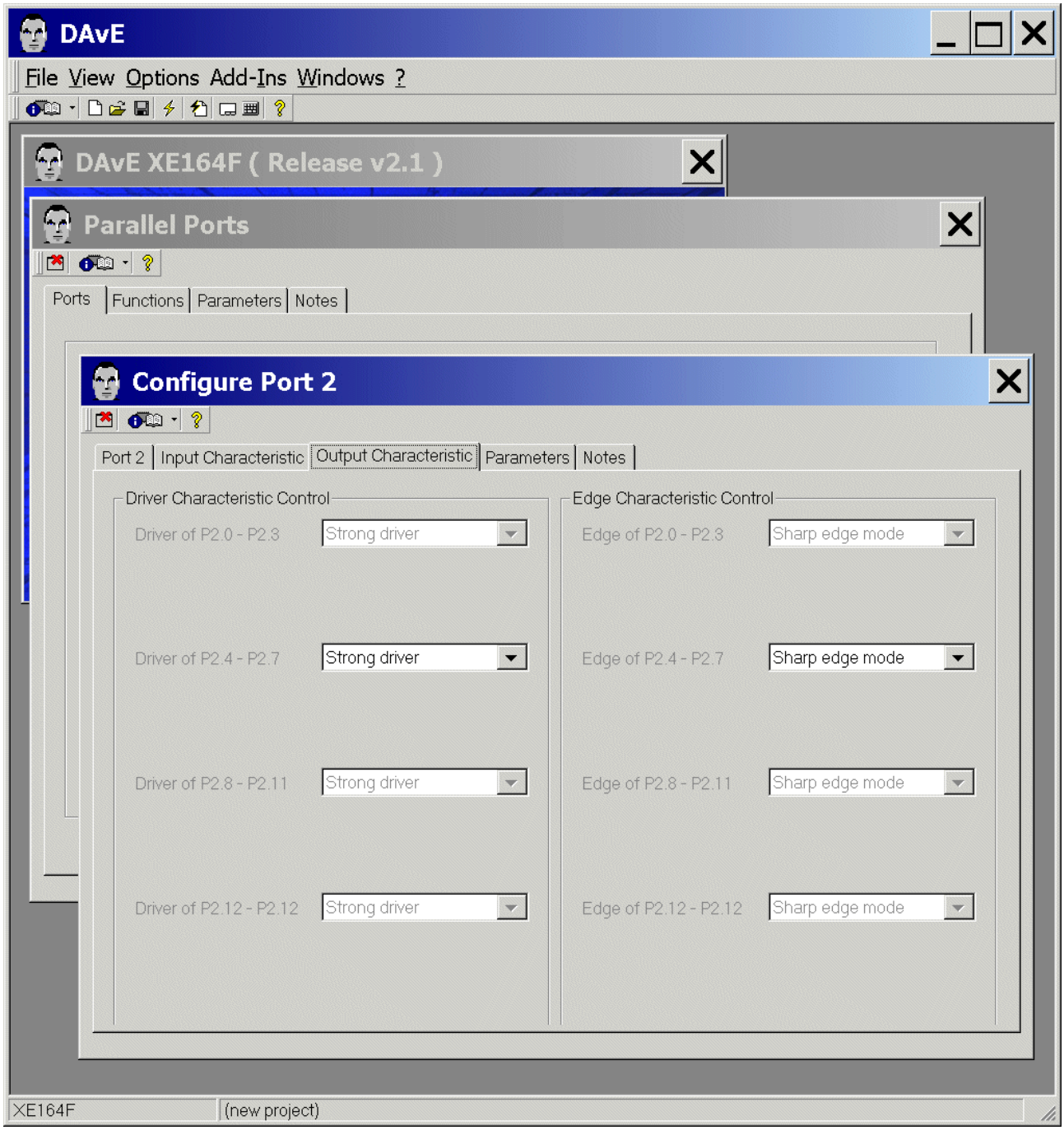
Port 2: Functionality: tick ☑ Use P2.7 as general IO - Direction: click ⊙ Out



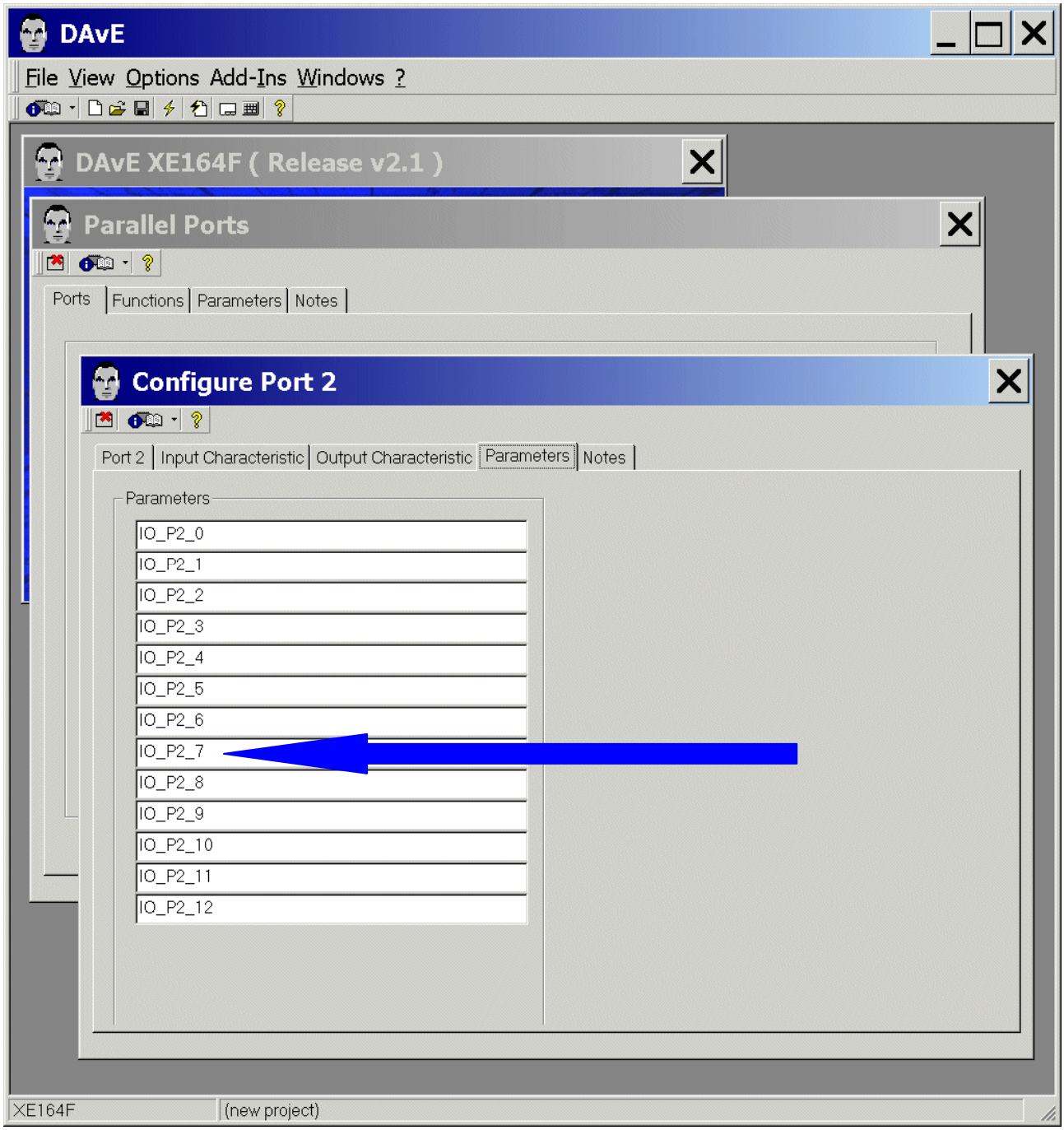
Input Characteristic: (do nothing)



Output Characteristic: (do nothing)

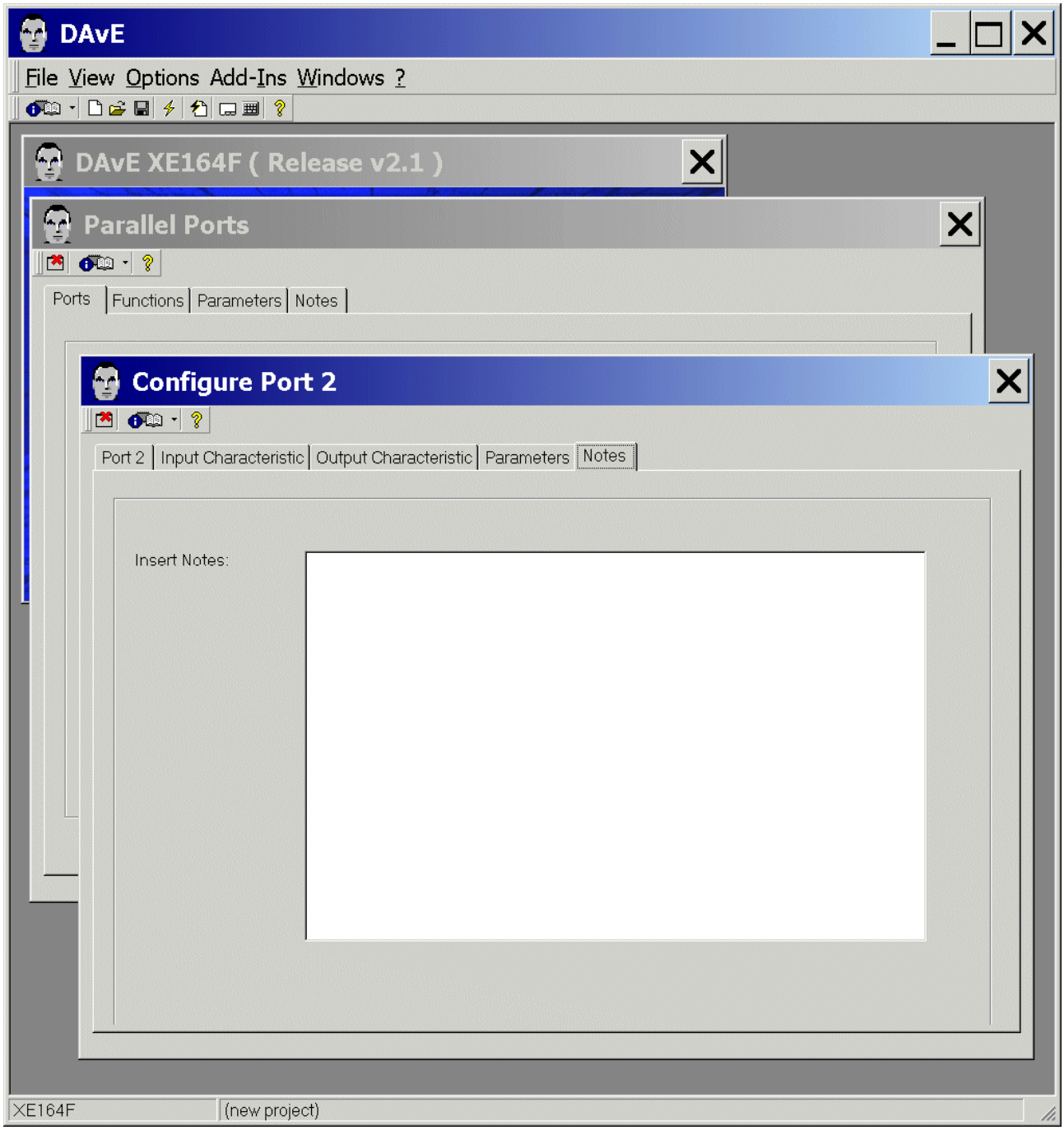


Parameters: (do nothing)



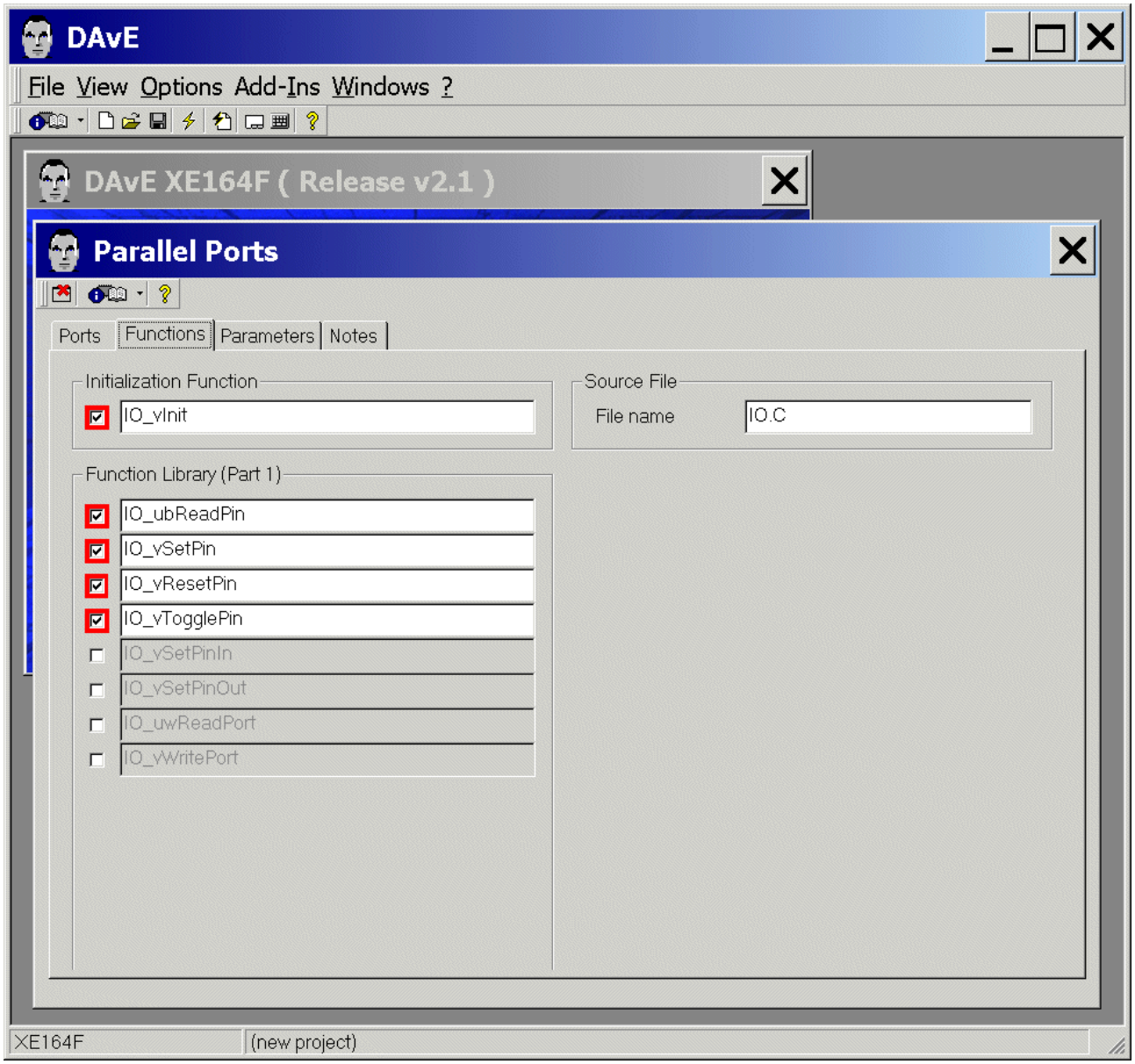
Note:
We will use the name IO_P2_7 in application programming.

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

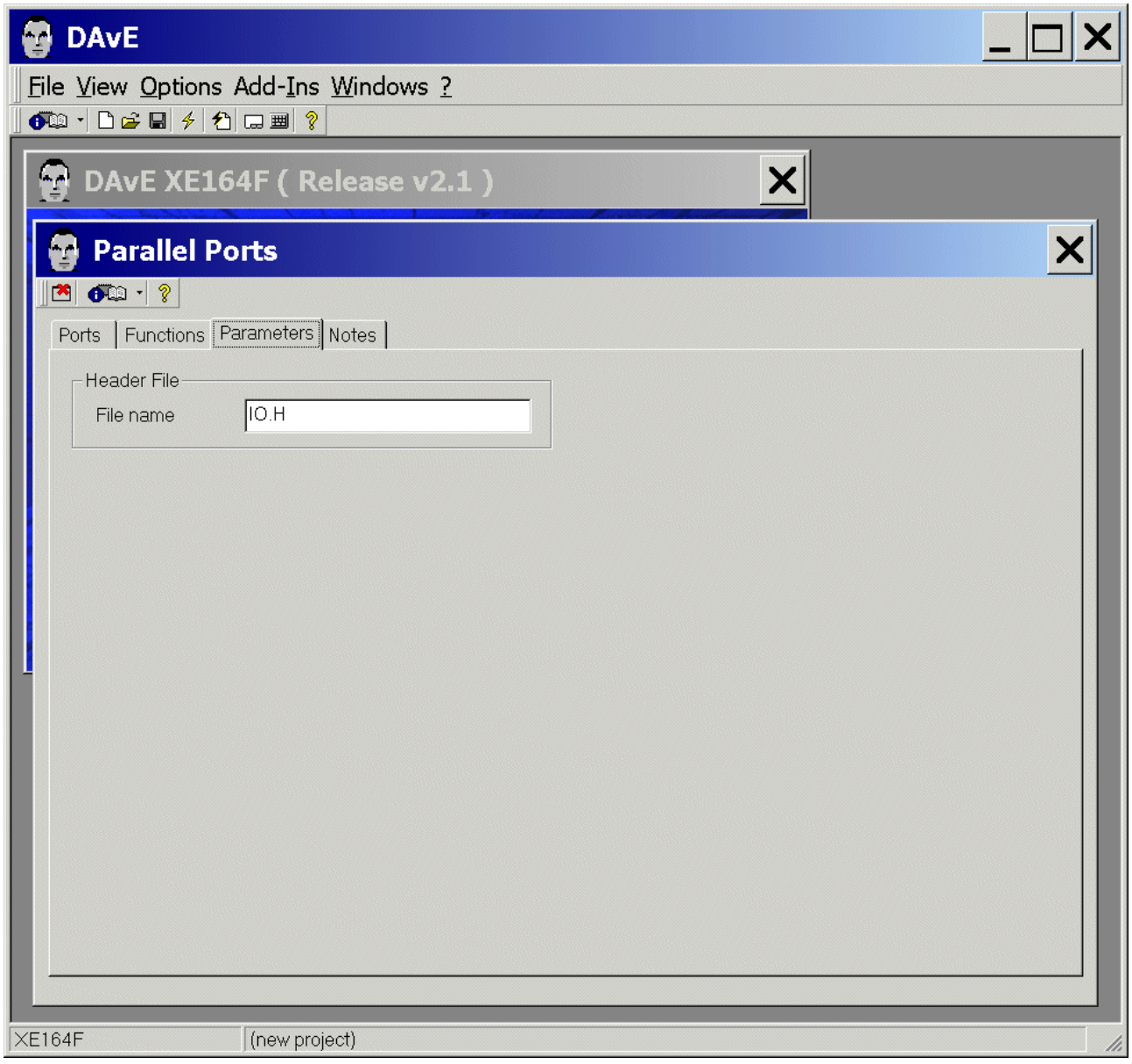


Exit and Save this dialog now by clicking the close button:

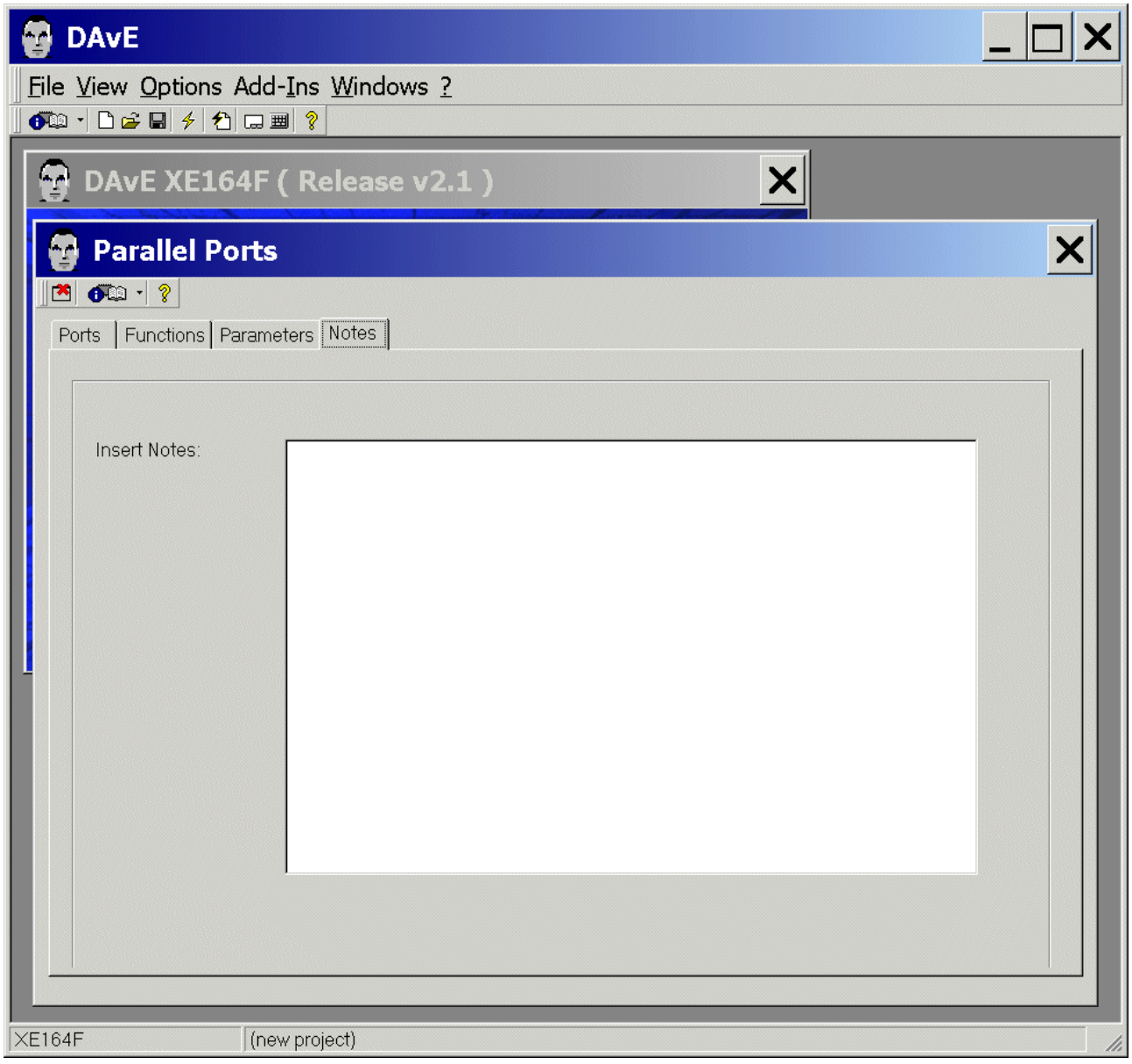
Functions: Initialization Functions: tick/check ☑ IO_vInit
Functions: Function Library (Part 1): tick ☑ IO_ubReadPin
Functions: Function Library (Part 1): tick ☑ IO_vSetPin
Functions: Function Library (Part 1): tick ☑ IO_vResetPin
Functions: Function Library (Part 1): tick ☑ IO_vTogglePin



Parameters: (do nothing)



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



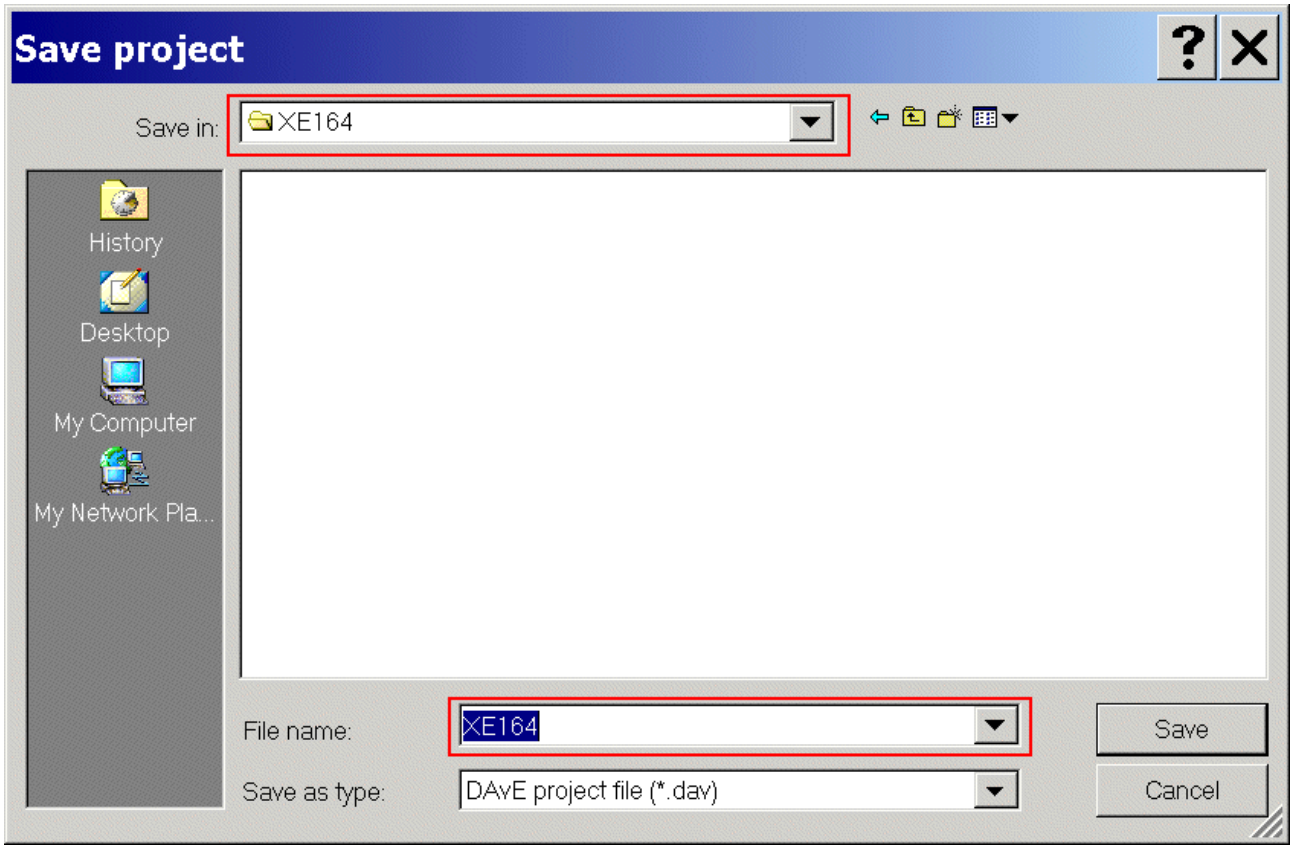
Exit and Save this dialog now by clicking the close button.

Save the project:

File
Save

Save project: Save in C:\XE164 (create new directory

!!!)
File name: XE164

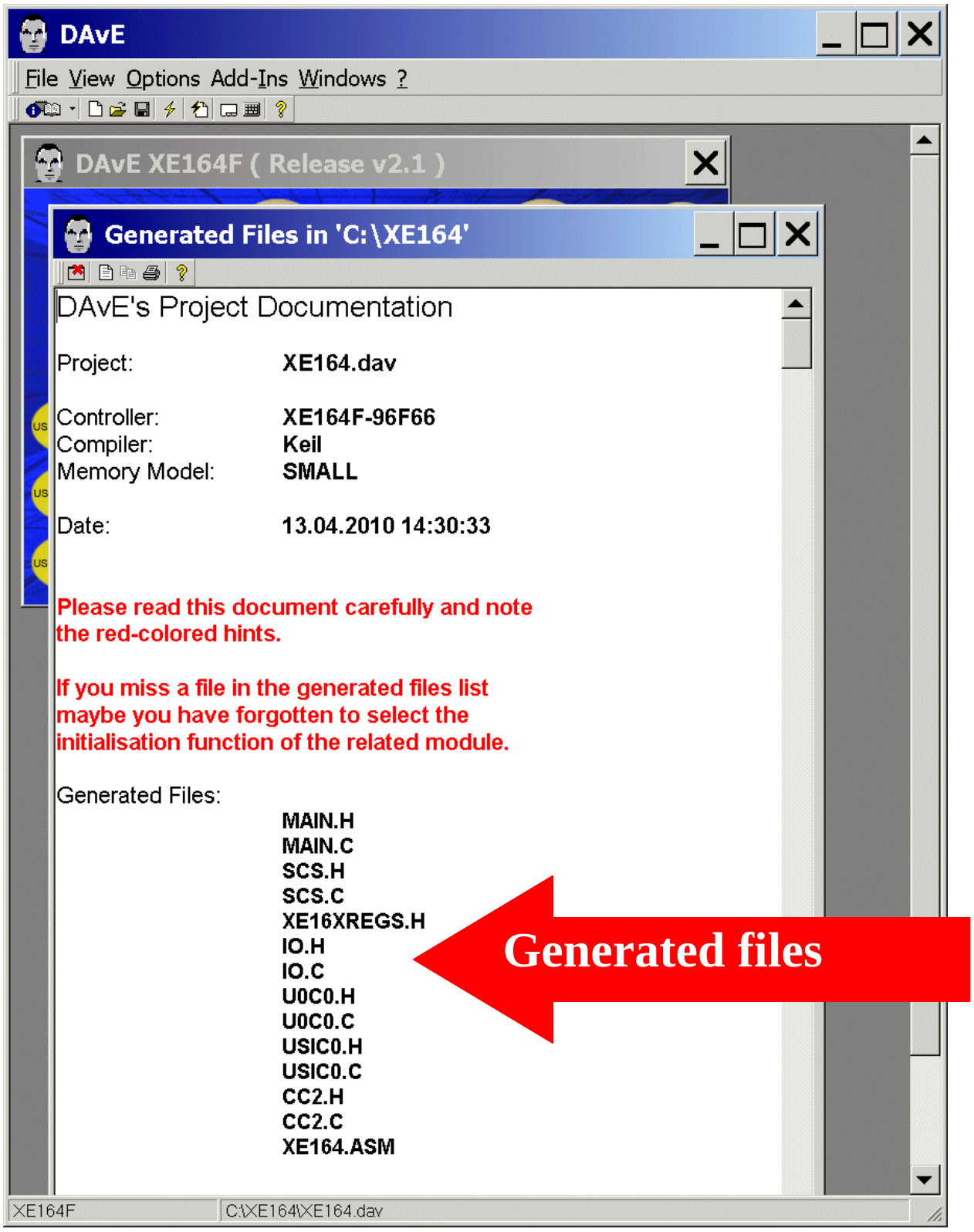


Save

Generate Code:

File - Generate Code	or	click

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

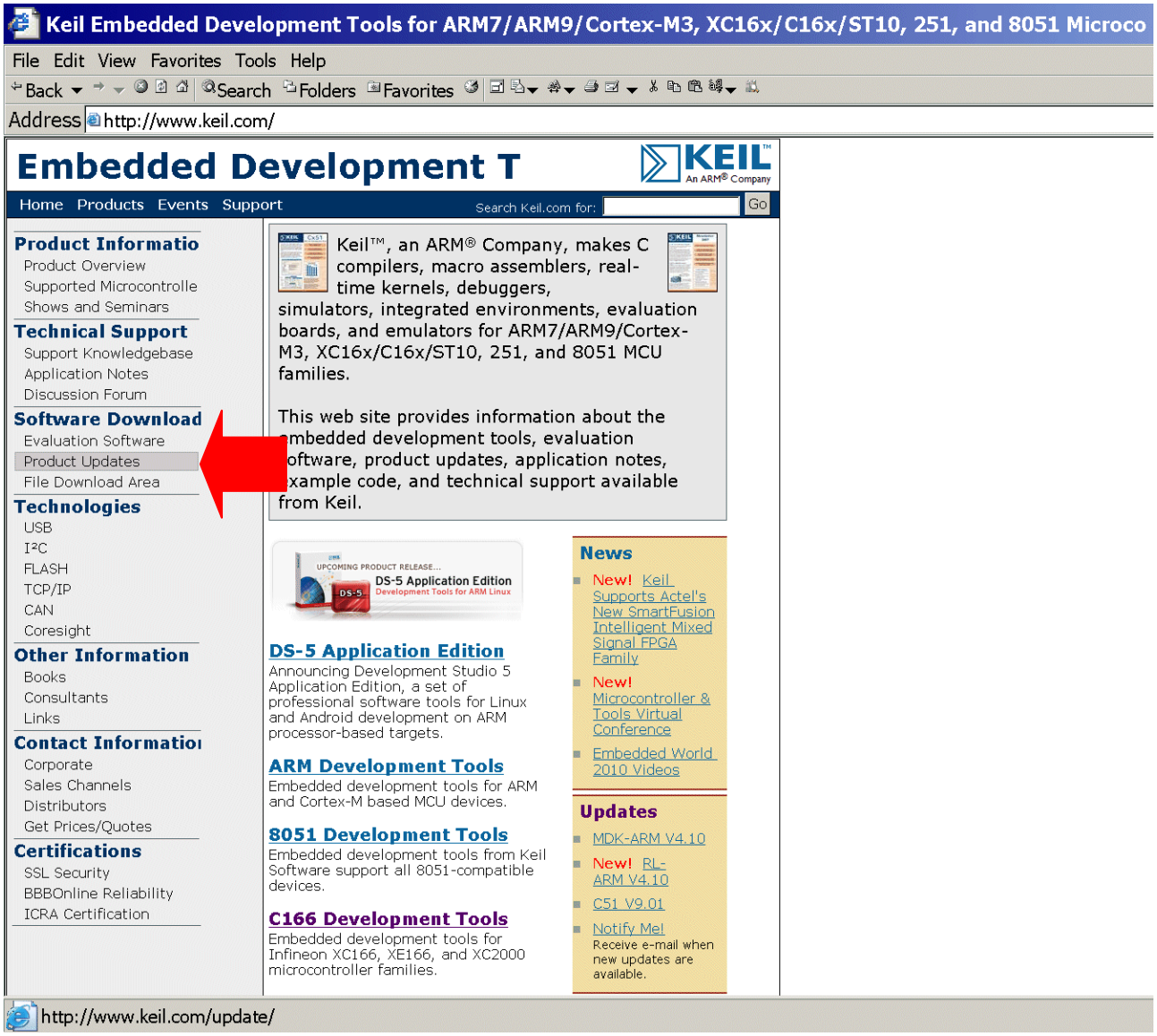


File - Exit
Save changes?

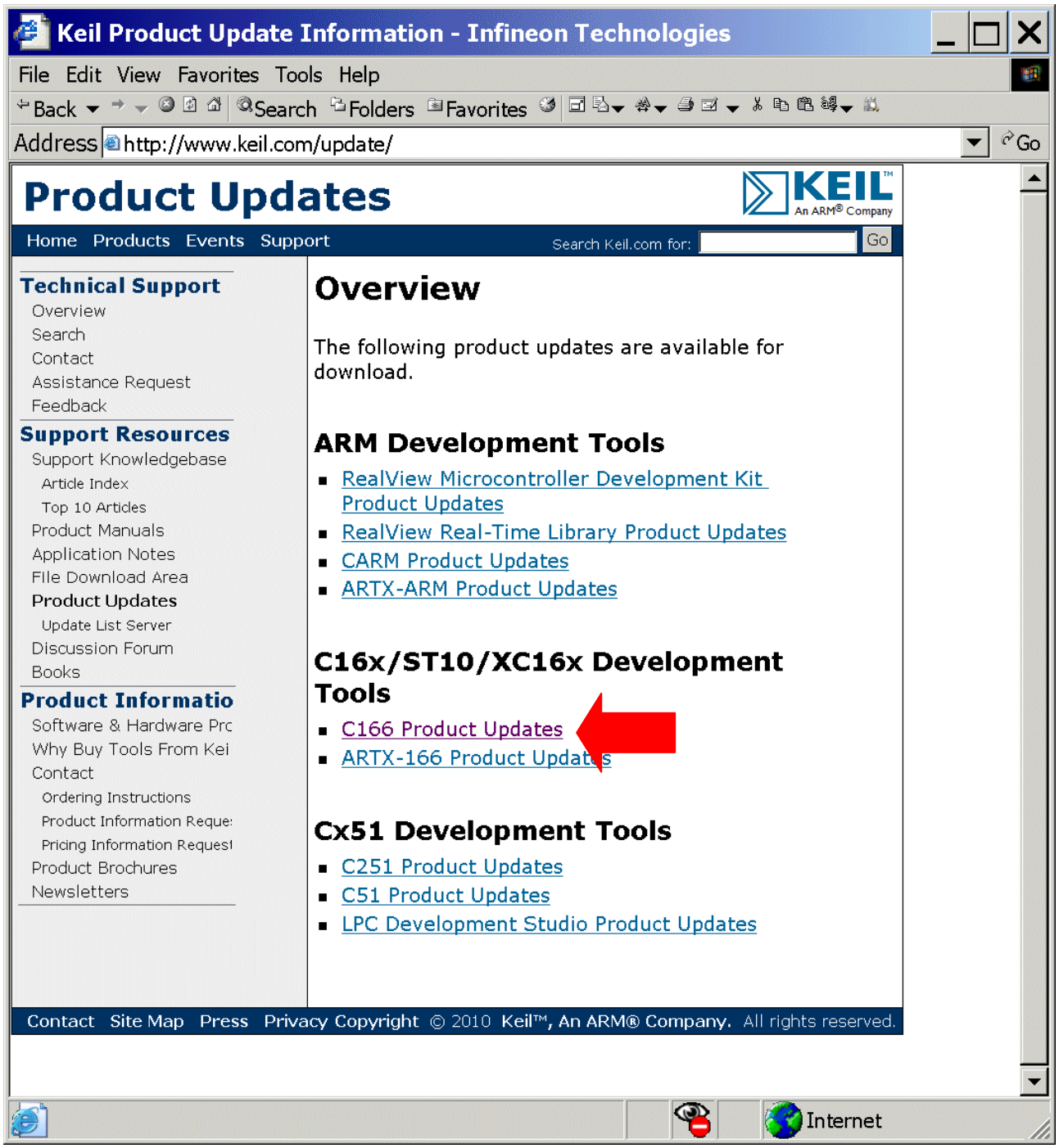
Click: Yes

4.) Using the KEIL - µVision 4 Development Tools:

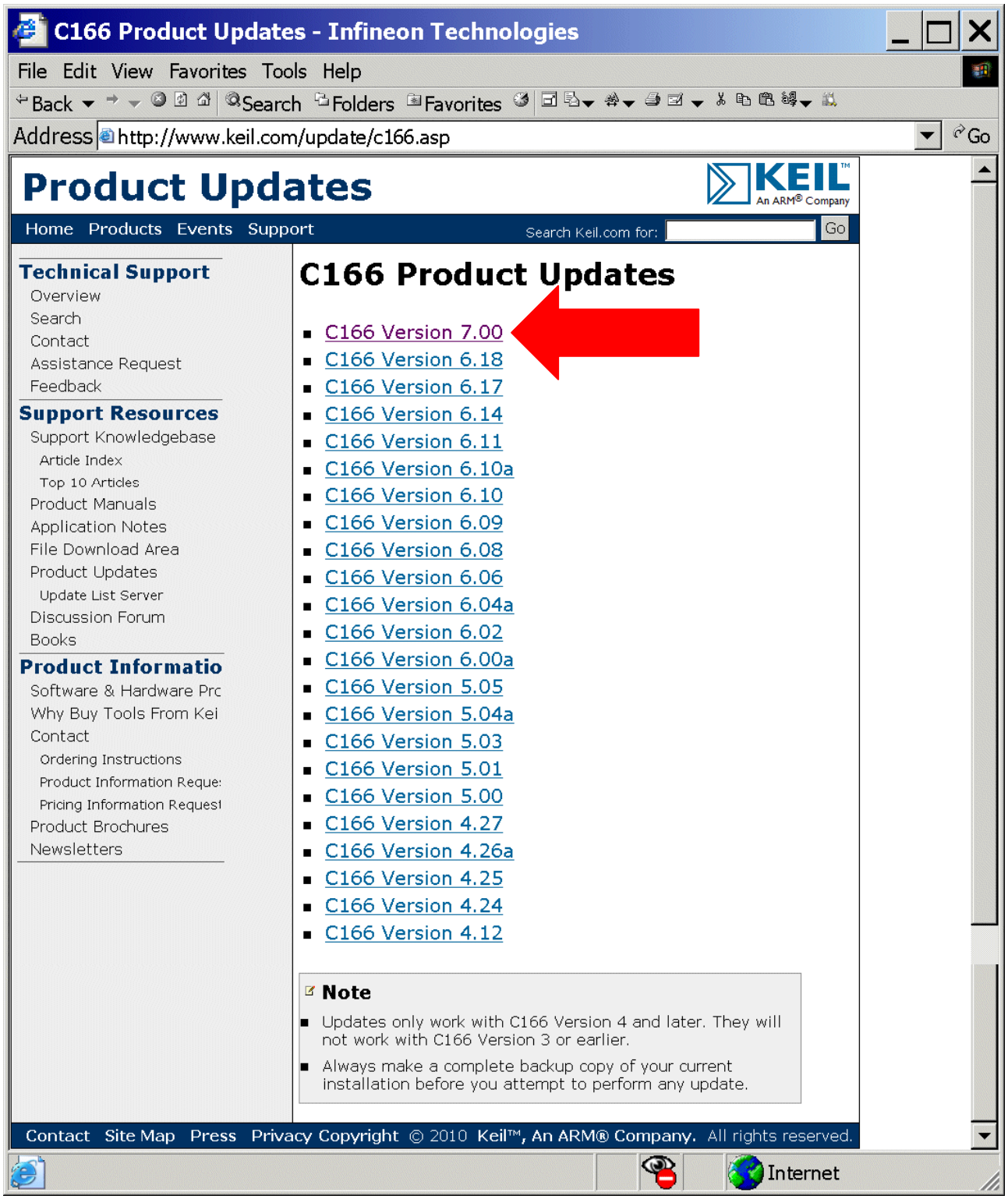
Download the tool chain: You can download the Keil Development Tools @ http://www.keil.com :



Click Product Updates

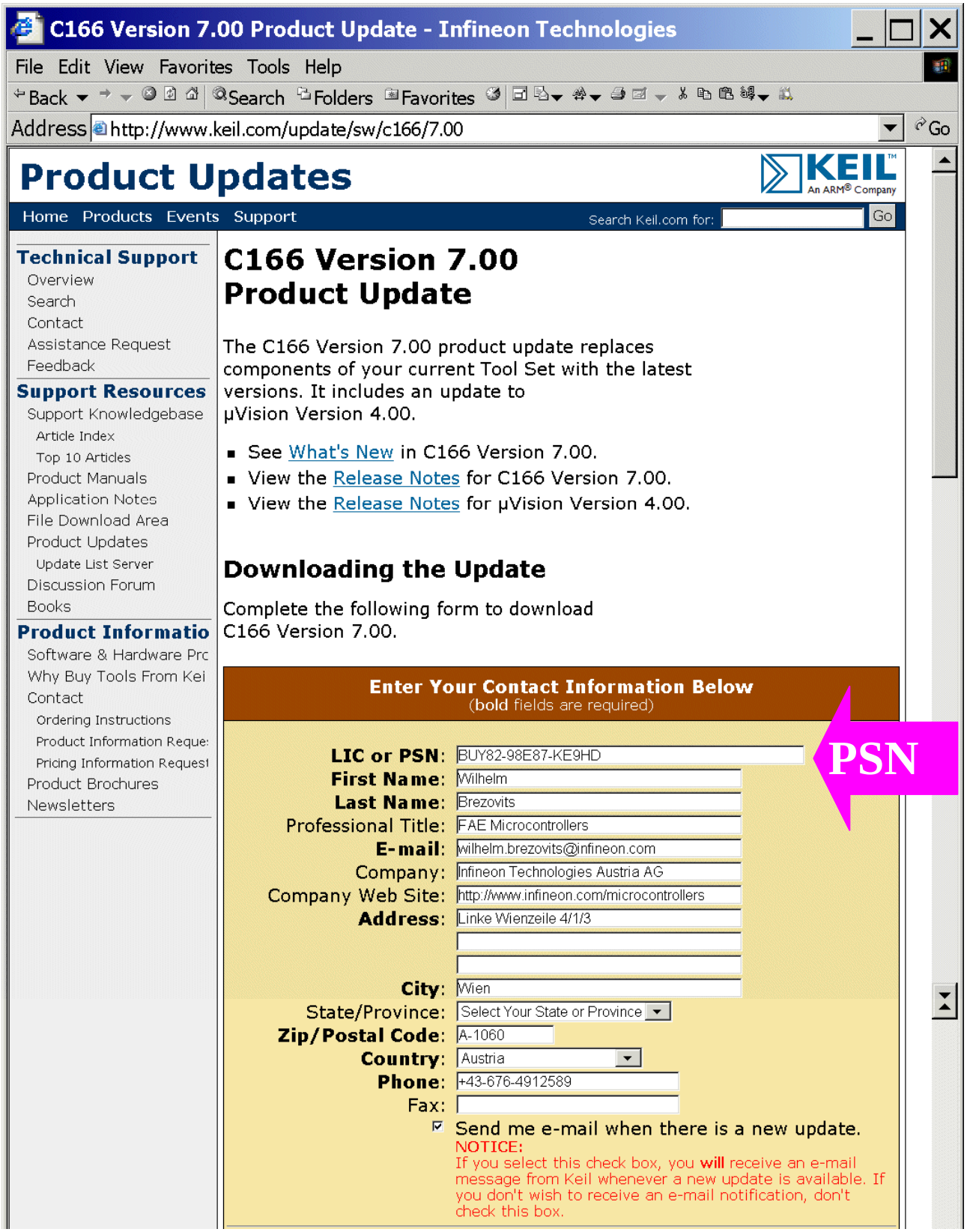


Click C166 Product Updates

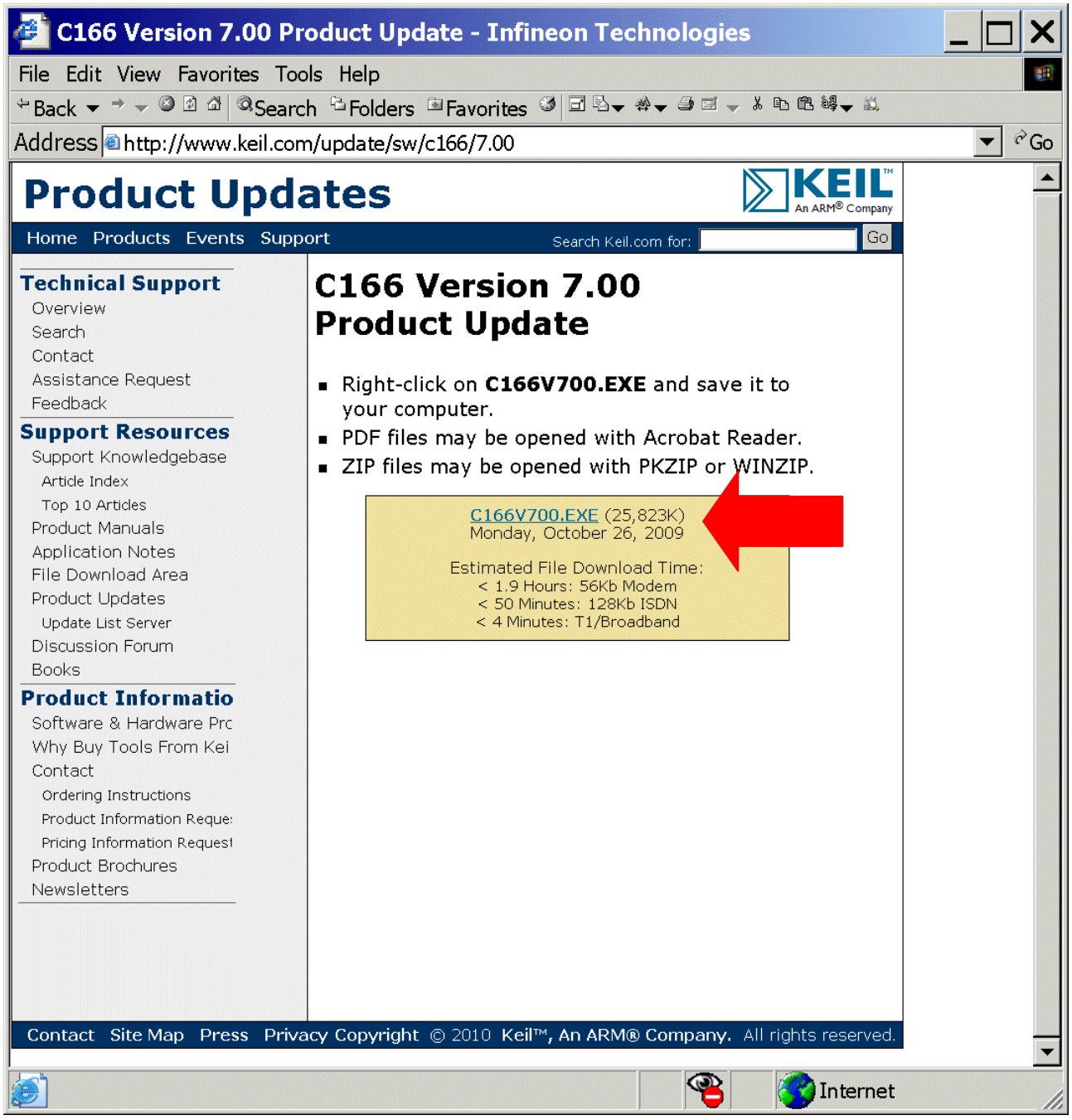


Click C166 Version 7.00

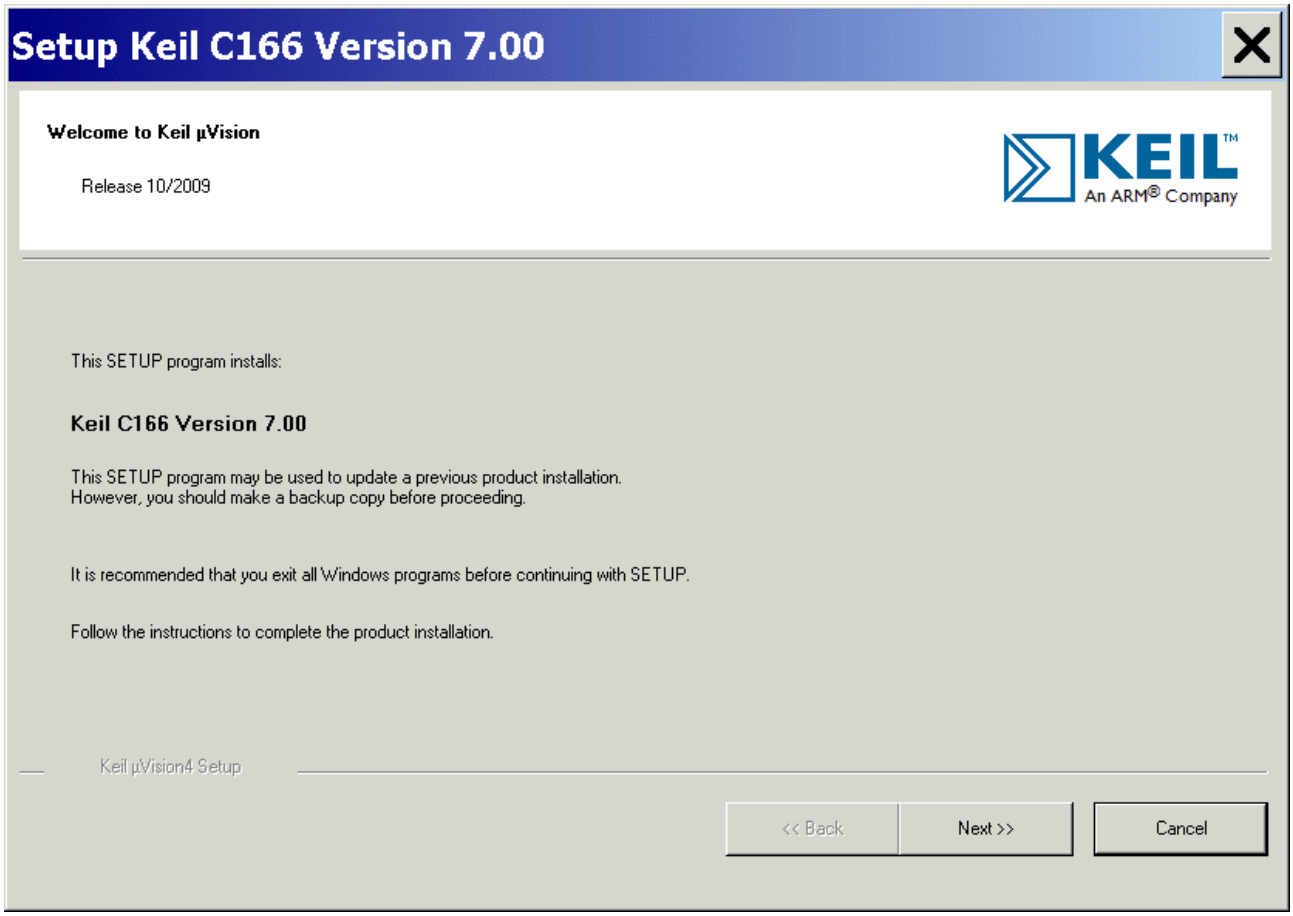
Insert your PRODUCT SERIAL NUMBER (PSN):

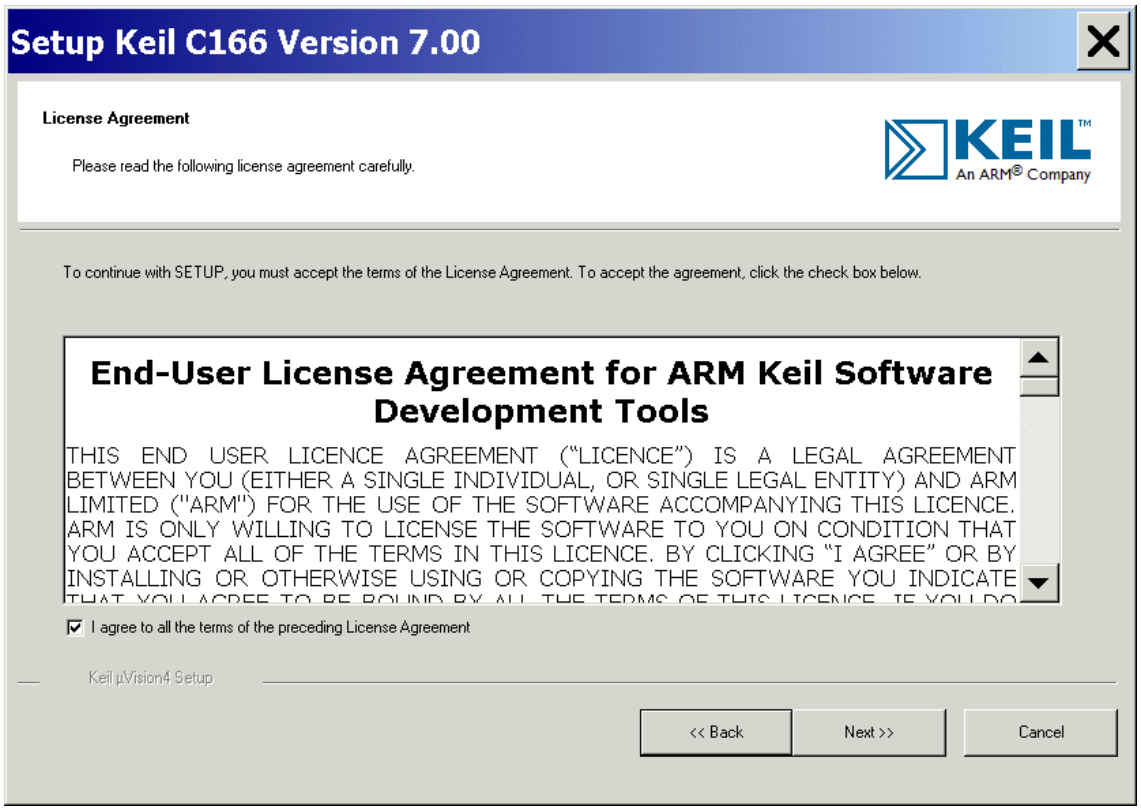


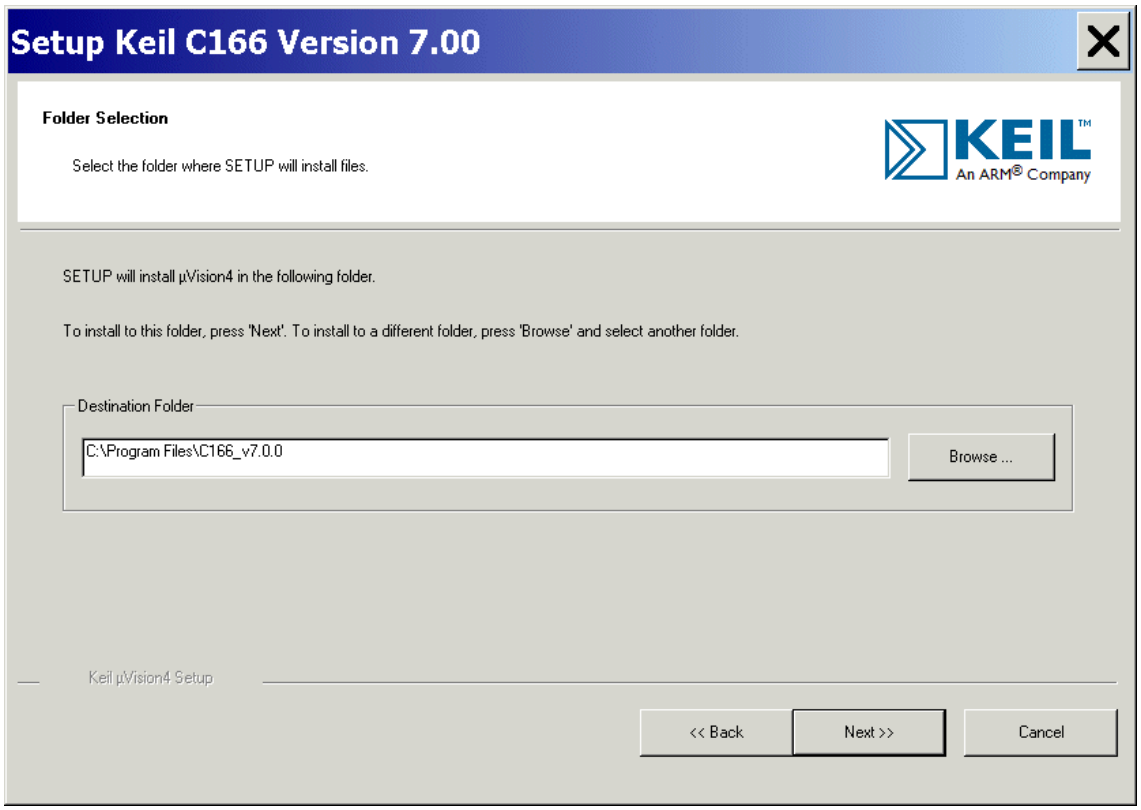
Click Submit

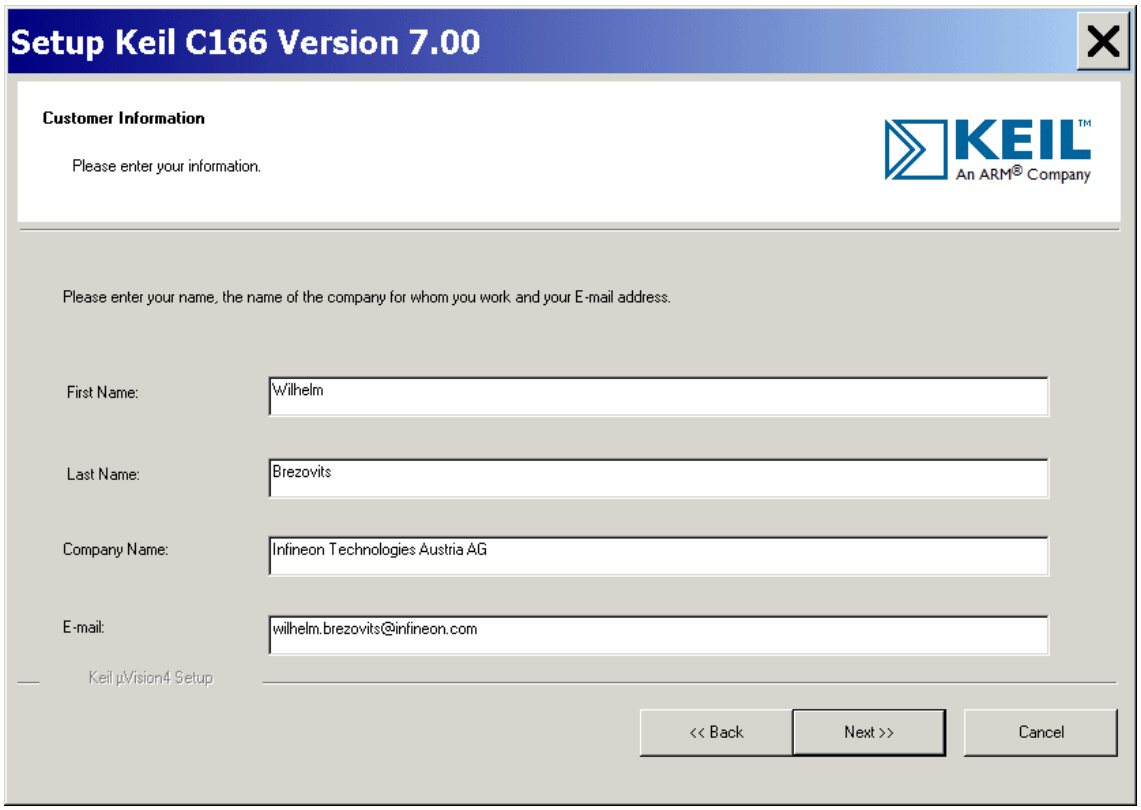

C166 Version 7.00 Product Update - Infineon Technologies
File Edit View Favorites Tools Help
Back Search Folders Favorites
Address http://www.keil.com/update/sw/c166/7.00
Go
Product Updates
KEIL An ARM® Company
Home Products Events Support
Search Keil.com for:
Go
Technical Support
Overview
Search
Contact
Assistance Request
Feedback
Support Resources
Support Knowledgebase
Article Index
Top 10 Articles
Product Manuals
Application Notes
File Download Area
Product Updates
Update List Server
Discussion Forum
Books
Product Informatio
Software & Hardware Prc
Why Buy Tools From Kei
Contact
Ordering Instructions
Product Information Reque
Pricing Information Request
Product Brochures
Newsletters
C166 Version 7.00
Product Update
Right-click on C166V700.EXE and save it to your computer.
PDF files may be opened with Acrobat Reader.
ZIP files may be opened with PKZIP or WINZIP.
C166V700.EXE (25,823K)
Monday, October 26, 2009
Estimated File Download Time:
< 1.9 Hours: 56Kb Modem
< 50 Minutes: 128Kb ISDN
< 4 Minutes: T1/Broadband
Contact Site Map Press Privacy Copyright © 2010 Keil™, An ARM® Company. All rights reserved.
Internet

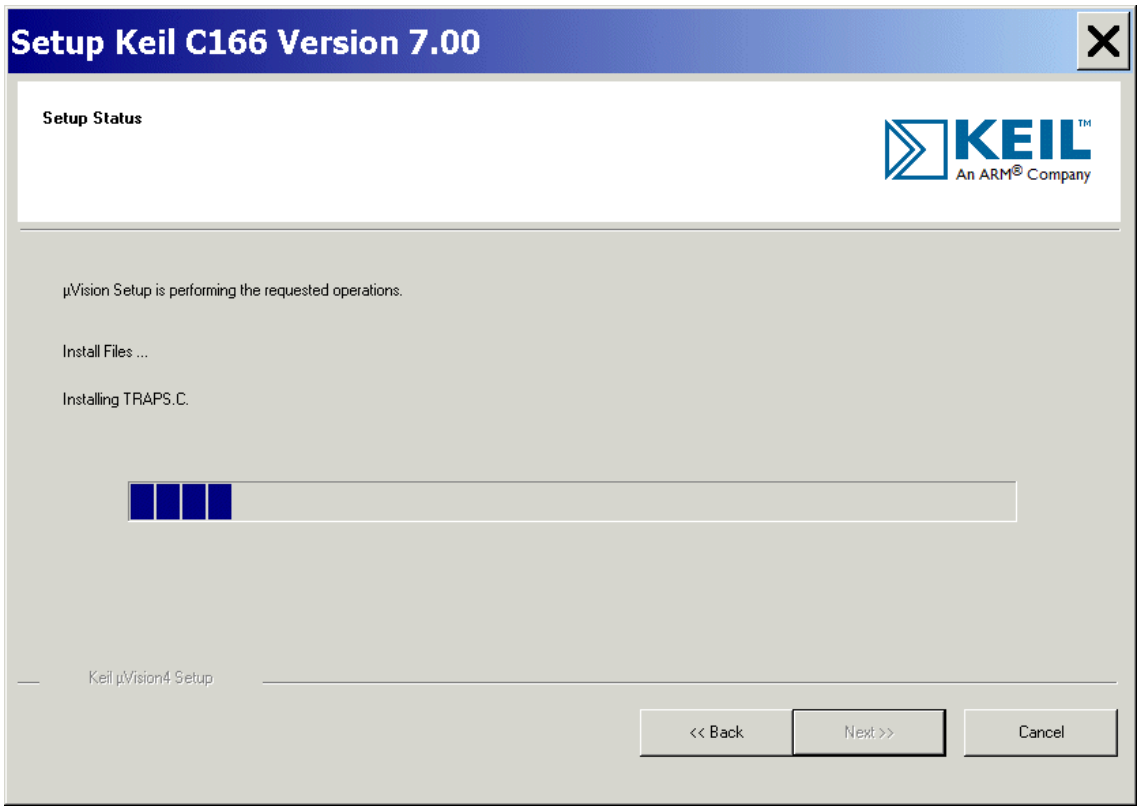
Download and Execute C166V700.EXE (- or any higher version)
and install the Keil tool chain.

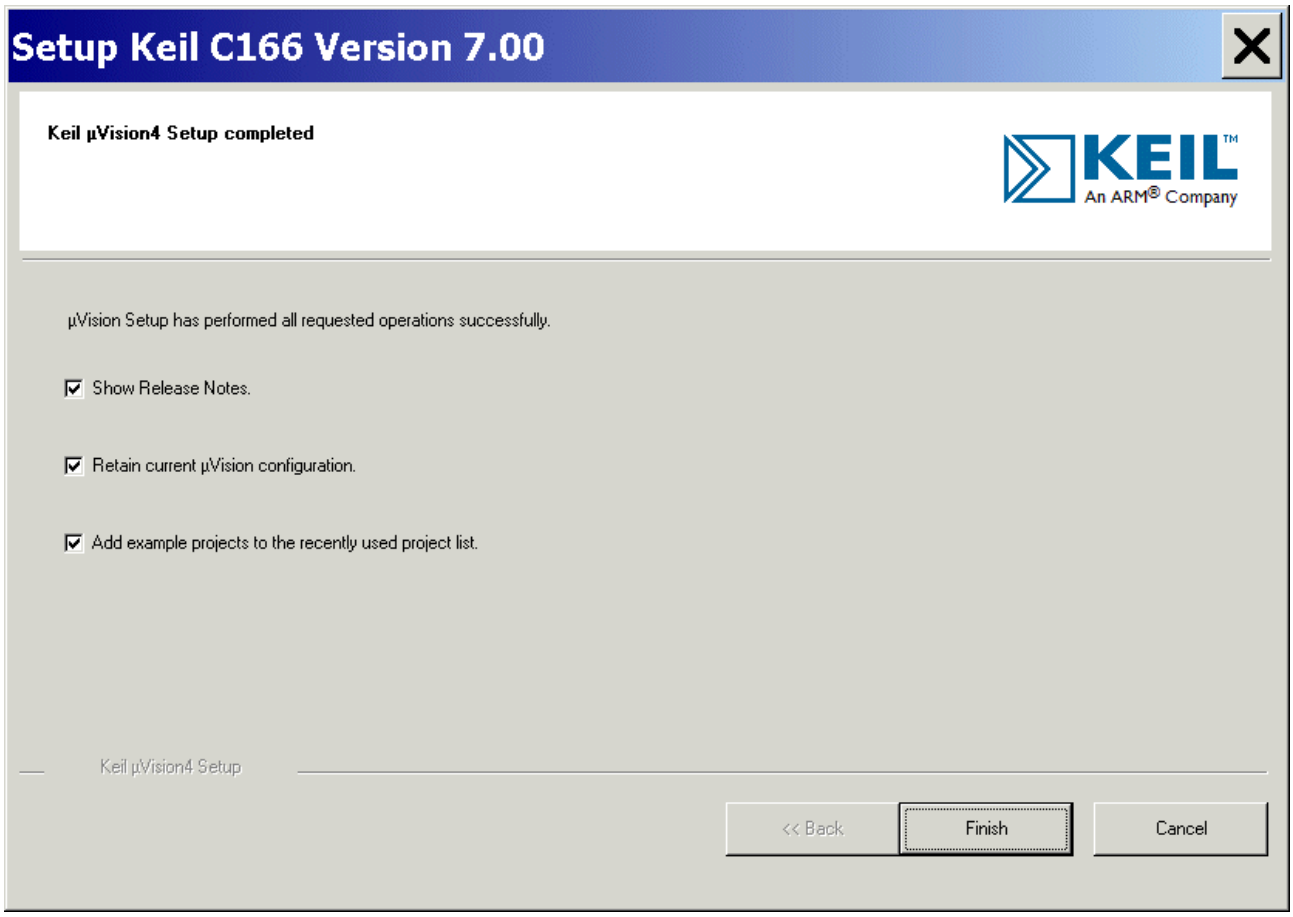



Setup Keil C166 Version 7.00
License Agreement
Please read the following license agreement carefully.
KEIL™
An ARM® Company
To continue with SETUP, you must accept the terms of the License Agreement. To accept the agreement, click the check box below.
End-User License Agreement for ARM Keil Software Development Tools
THIS END USER LICENCE AGREEMENT ("LICENCE") IS A LEGAL AGREEMENT BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR SINGLE LEGAL ENTITY) AND ARM LIMITED ("ARM") FOR THE USE OF THE SOFTWARE ACCOMPANYING THIS LICENCE. ARM IS ONLY WILLING TO LICENSE THE SOFTWARE TO YOU ON CONDITION THAT YOU ACCEPT ALL OF THE TERMS IN THIS LICENCE. BY CLICKING "I AGREE" OR BY INSTALLING OR OTHERWISE USING OR COPYING THE SOFTWARE YOU INDICATE
I agree to all the terms of the preceding License Agreement
Keil µVision4 Setup
<< Back
Next >>
Cancel


Setup Keil C166 Version 7.00
Folder Selection
Select the folder where SETUP will install files.
KEIL™
An ARM® Company
SETUP will install µVision4 in the following folder.
To install to this folder, press 'Next'. To install to a different folder, press 'Browse' and select another folder.
Destination Folder
C:\Program Files\C166_v7.0.0
Browse ...
Keil µVision4 Setup
<< Back
Next >>
Cancel


Setup Keil C166 Version 7.00
Customer Information
Please enter your information.
KEIL™
An ARM® Company
Please enter your name, the name of the company for whom you work and your E-mail address.
First Name:
Wilhelm
Last Name:
Brezovits
Company Name:
Infineon Technologies Austria AG
E-mail:
wilhelm.brezovits@infineon.com
Keil µVision4 Setup
<< Back
Next >>
Cancel

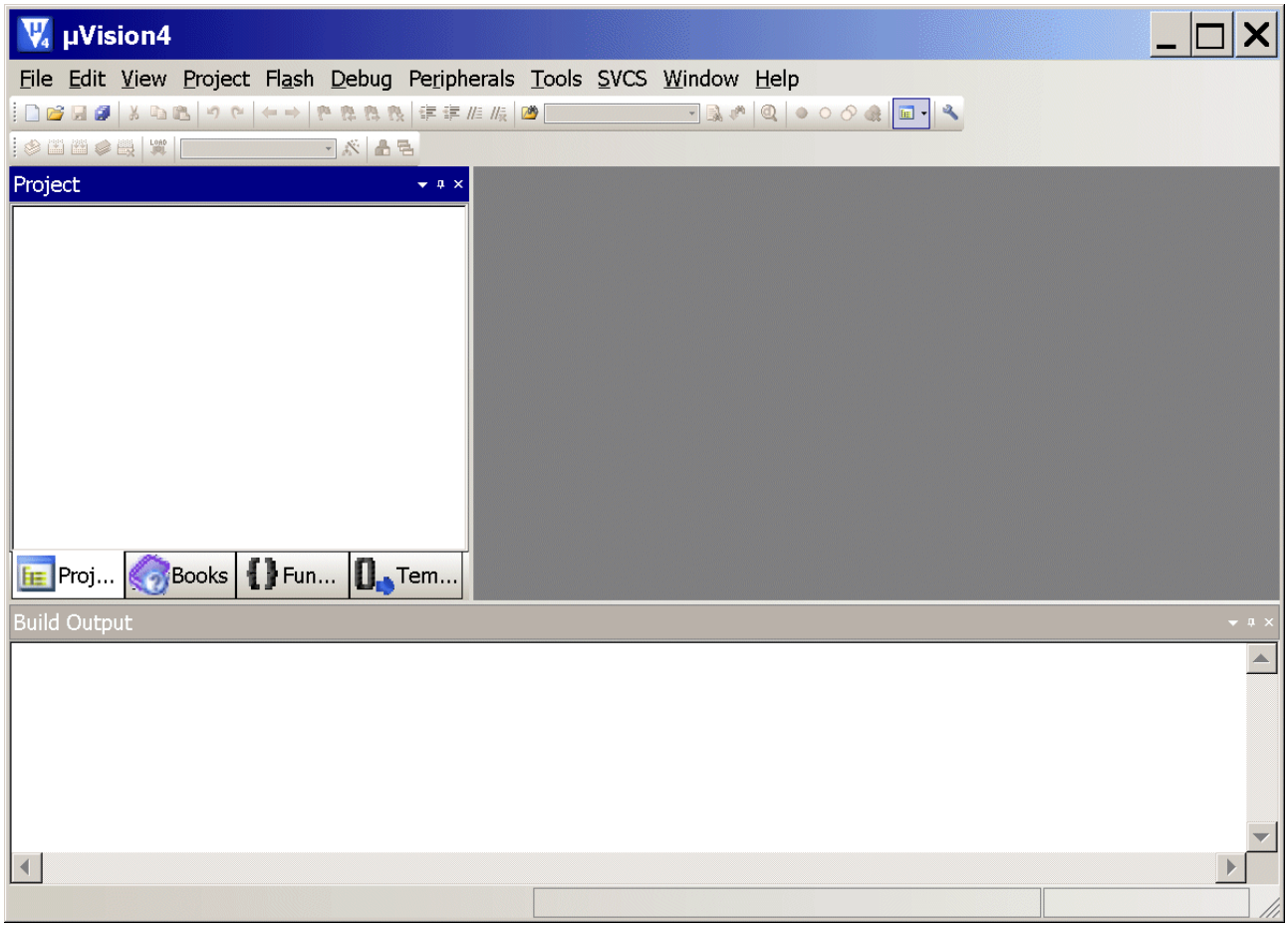

Setup Keil C166 Version 7.00
Setup Status
KEIL™
An ARM® Company
µVision Setup is performing the requested operations.
Install Files ...
Installing TRAPS.C.
Keil µVision4 Setup
<< Back
Next >>
Cancel


Setup Keil C166 Version 7.00
Keil µVision4 Setup completed
KEIL
An ARM® Company
µVision Setup has performed all requested operations successfully.
Show Release Notes.
Retain current µVision configuration.
Add example projects to the recently used project list.
Keil µVision4 Setup
<< Back
Finish
Cancel

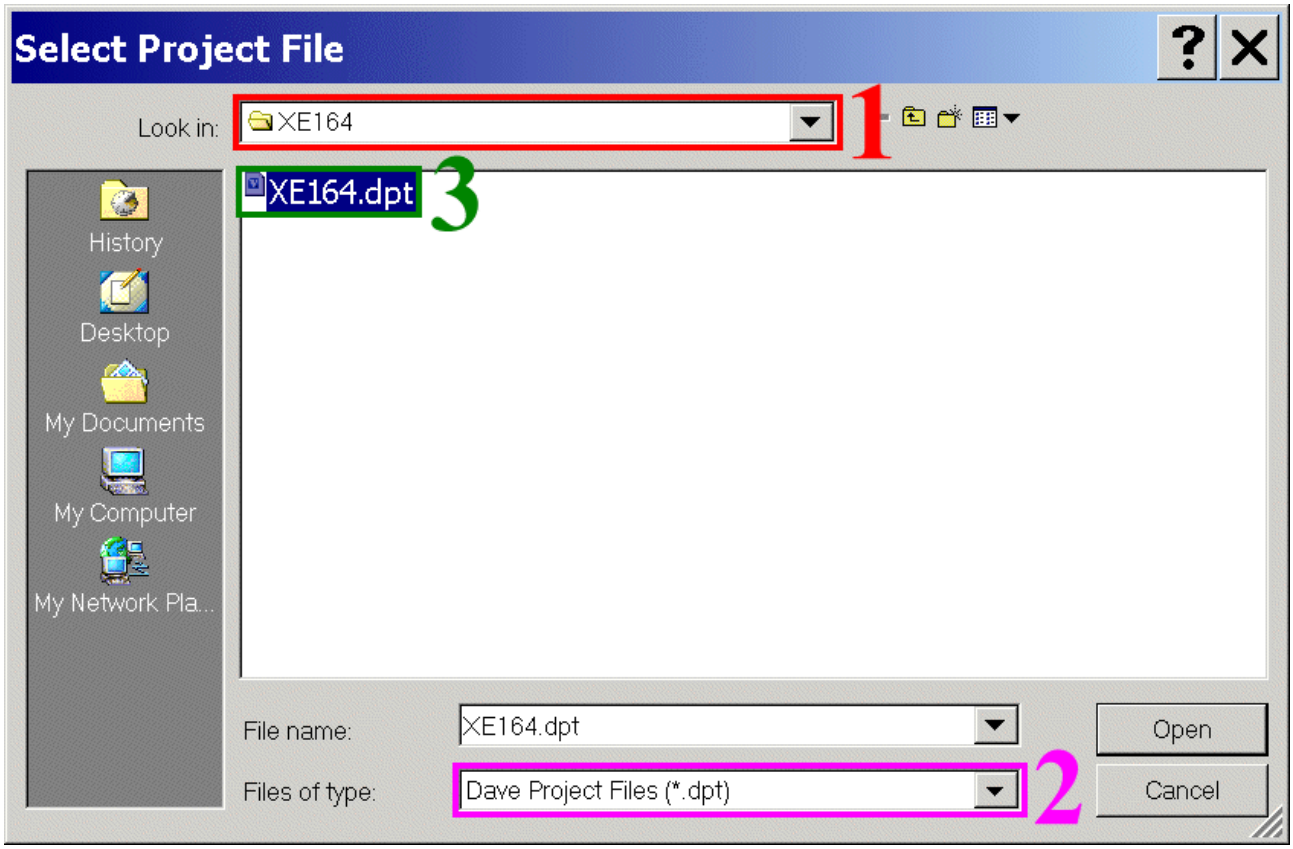
Click
Finish

Start Keil µVision4 and open the DAvE Project:

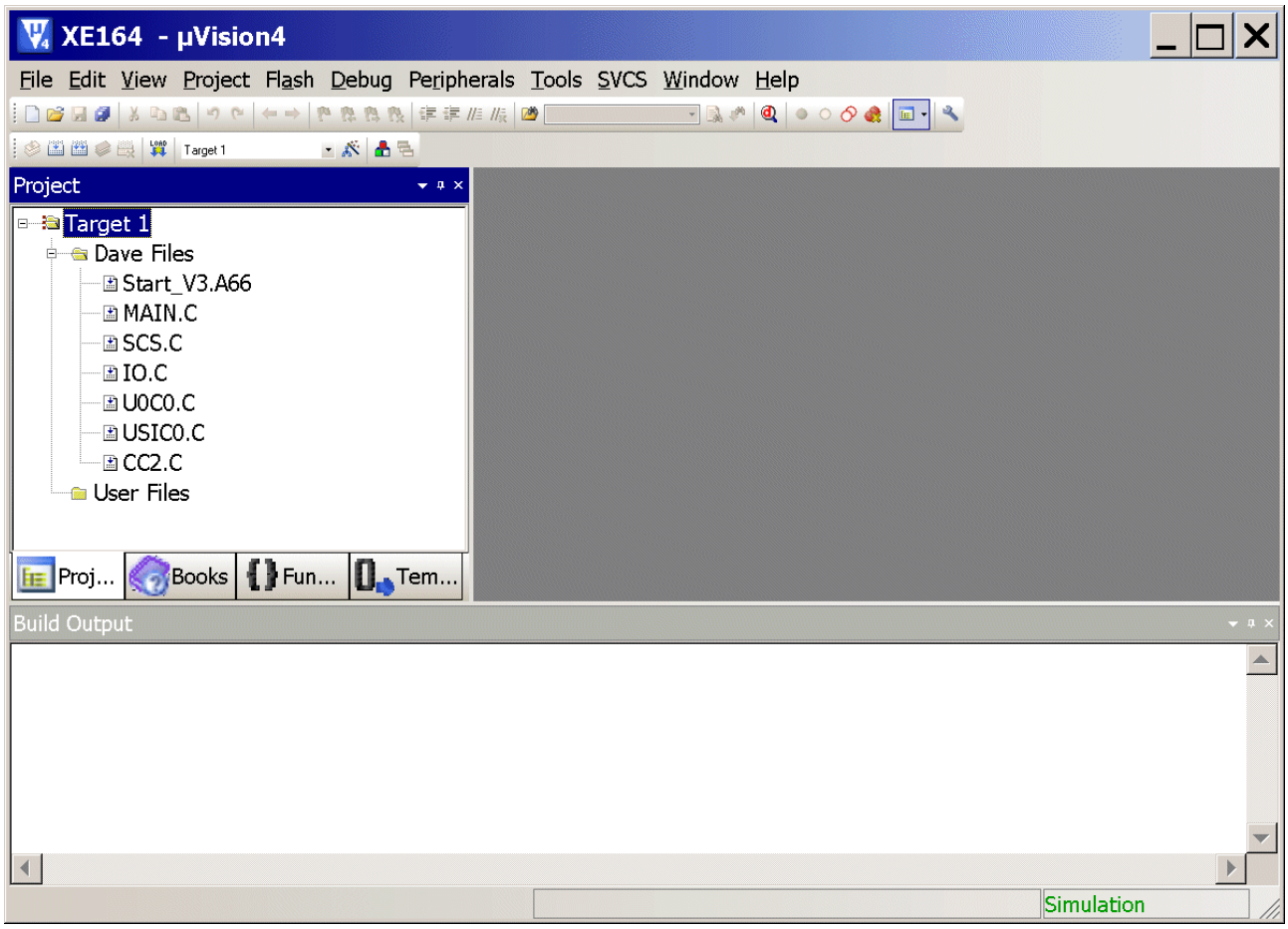
If you see an open project – close it: Project - Close Project

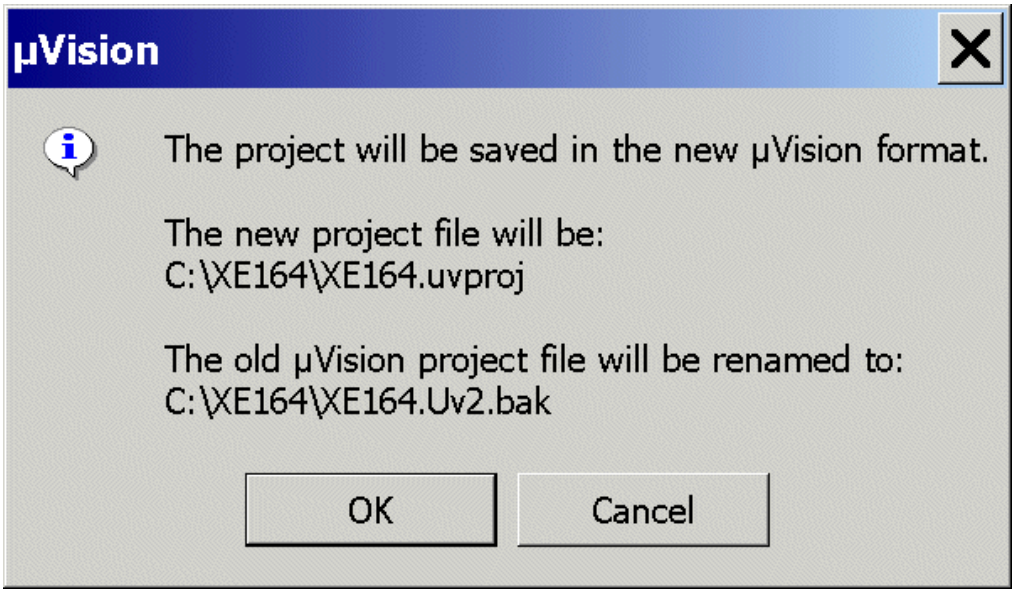


Project - Open Project
Select Project File: Look in: choose C:\XE164 **(1)**
Select Project File: Files of type: select Dave Project Files **(2)**
Click XE164.dpt **(3)**



Click

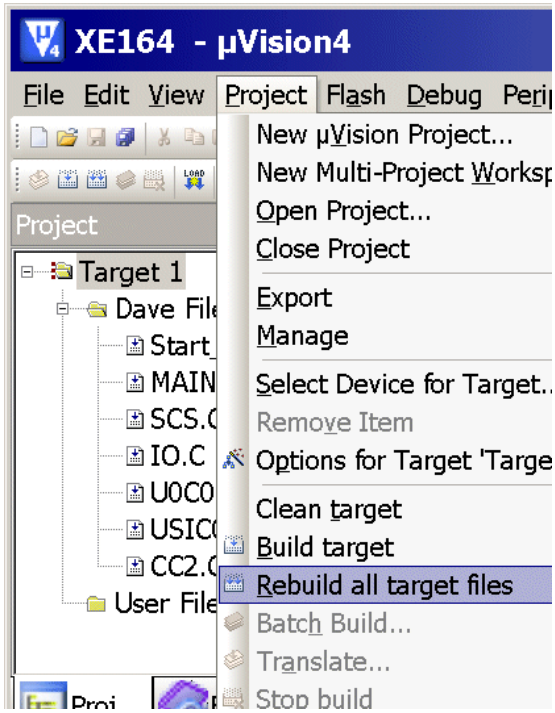

XE164 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Dave Files
Start_V3.A66
MAIN.C
SCS.C
IO.C
U0C0.C
USIC0.C
CC2.C
User Files
Proj...
Books
Fun...
Tem...
Build Output
Simulation


µVision
The project will be saved in the new µVision format.
The new project file will be:
C:\XE164\XE164.uvproj
The old µVision project file will be renamed to:
C:\XE164\XE164.Uv2.bak
OK
Cancel

Click OK

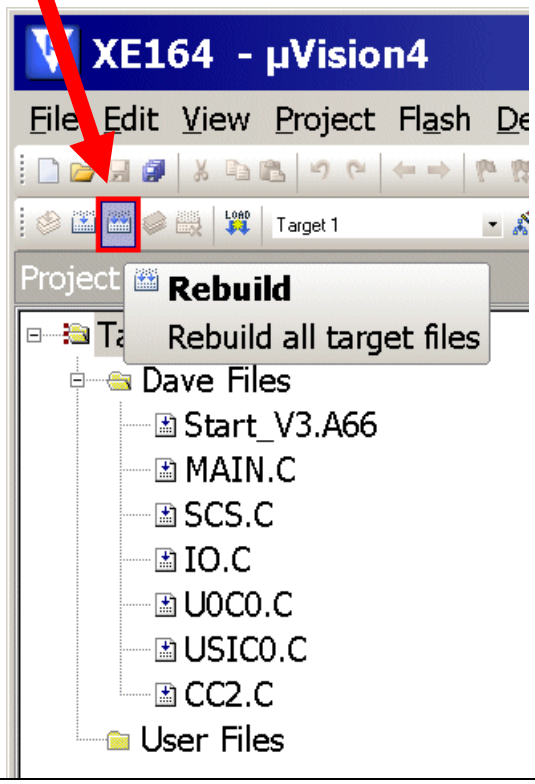
Generate „make“- file:

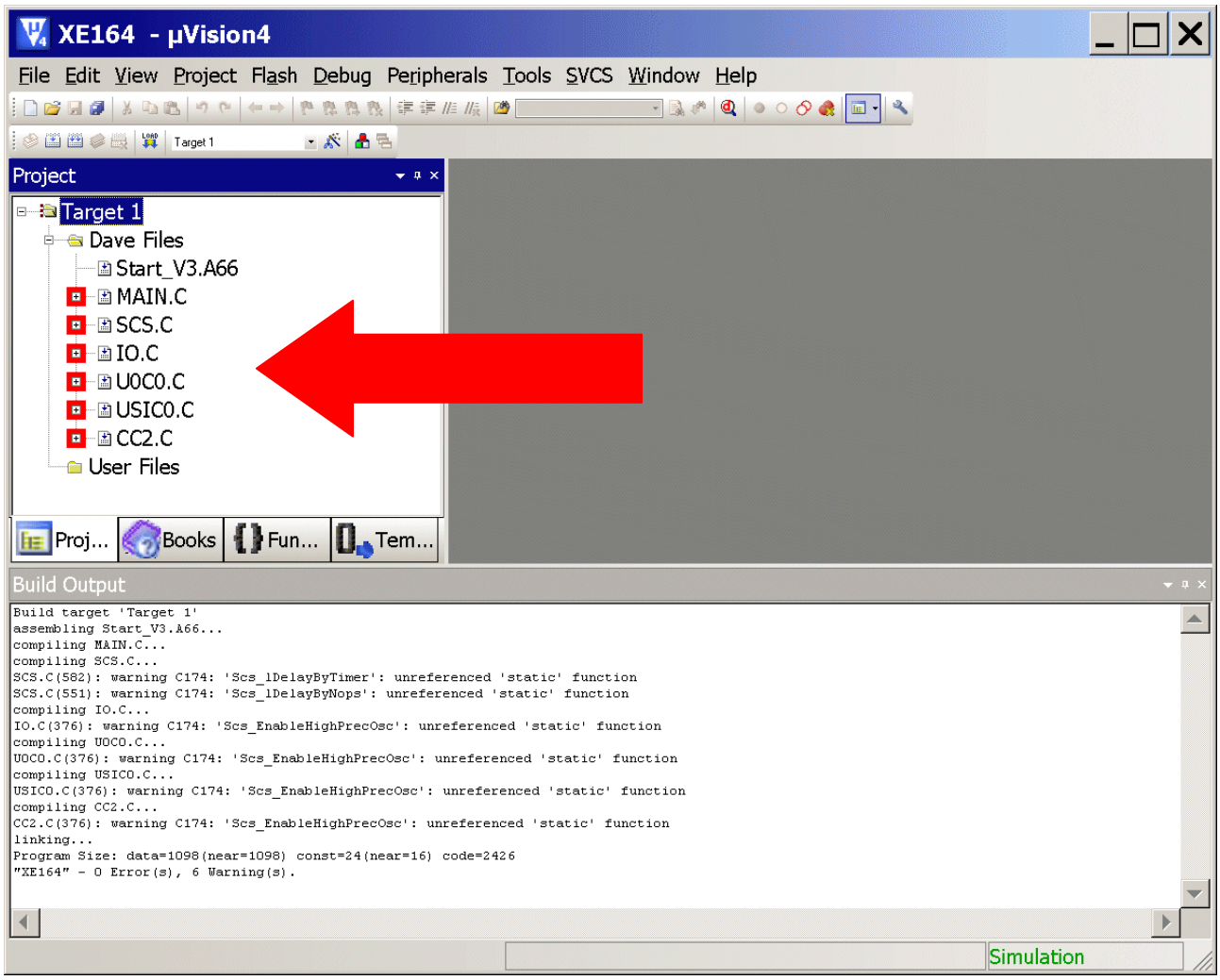
Project – Rebuild all target files



or

click



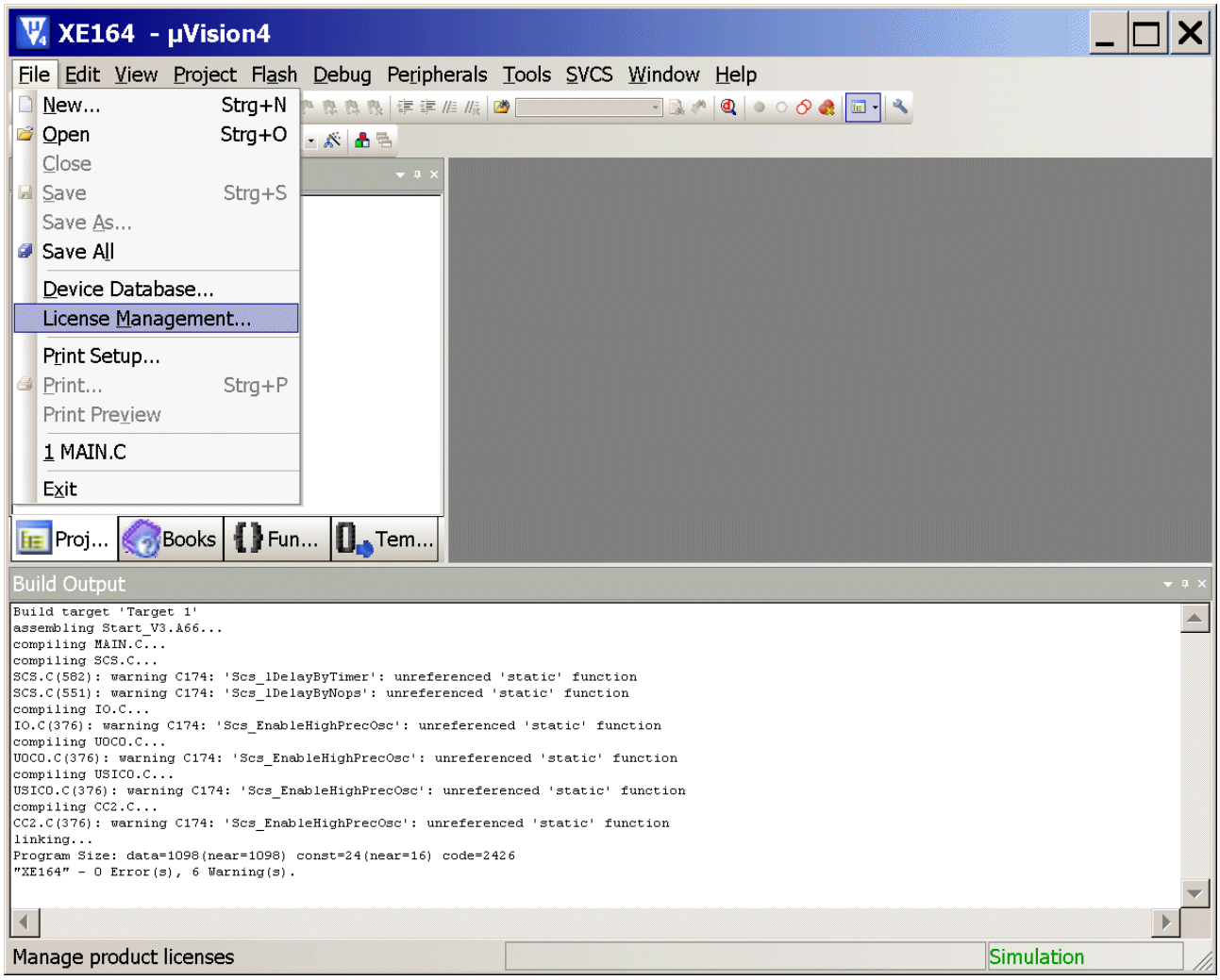


Note:
This step generates a makefile and shows the include files.

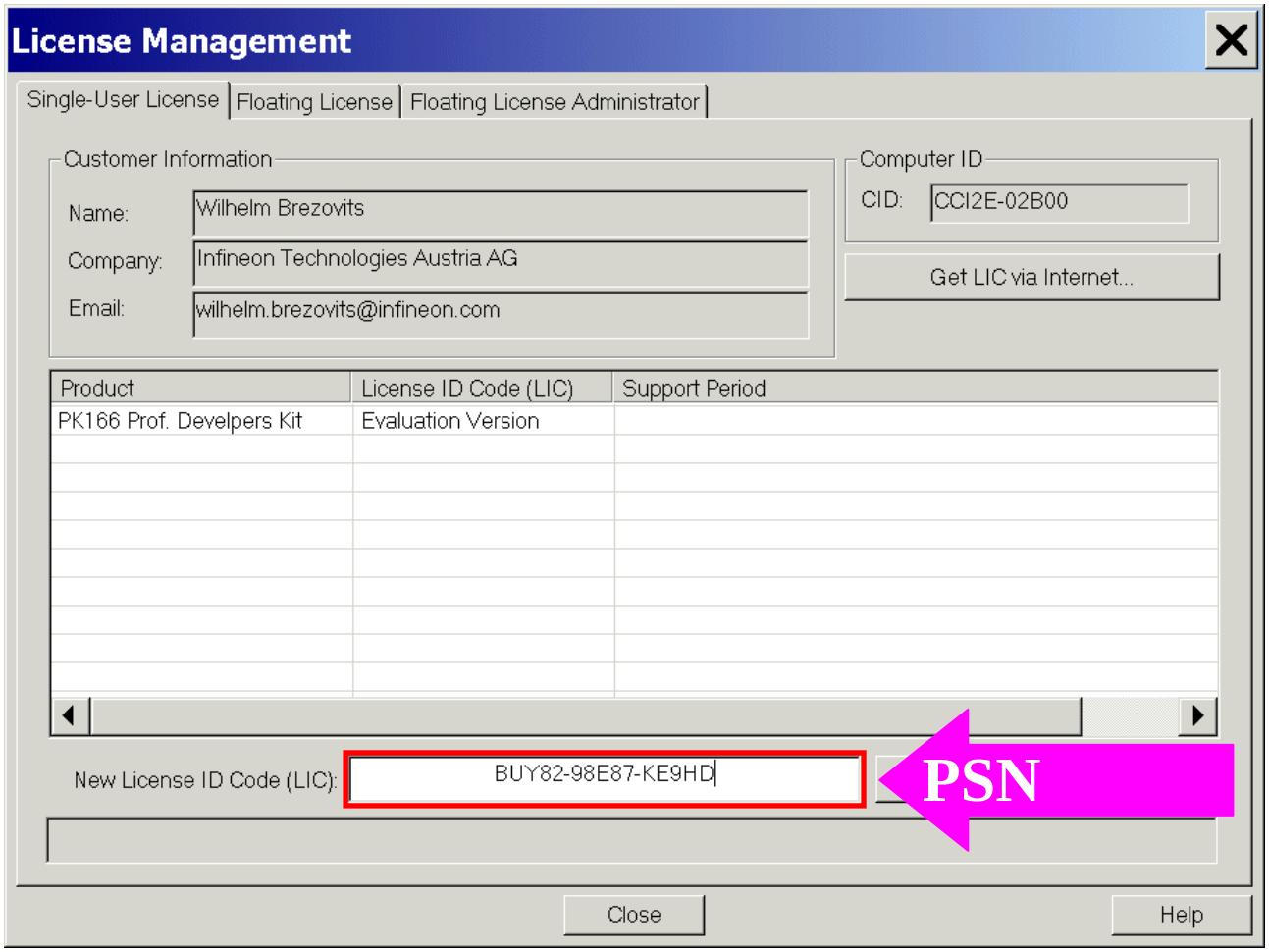
uVision4 License Management

Note:
To activate your Keil product, copy and paste your PRODUCT SERIAL NUMBER (PSN) into the New License ID Code (LIC) input field in the uVision4 License Manager Dialog (available from the File menu).

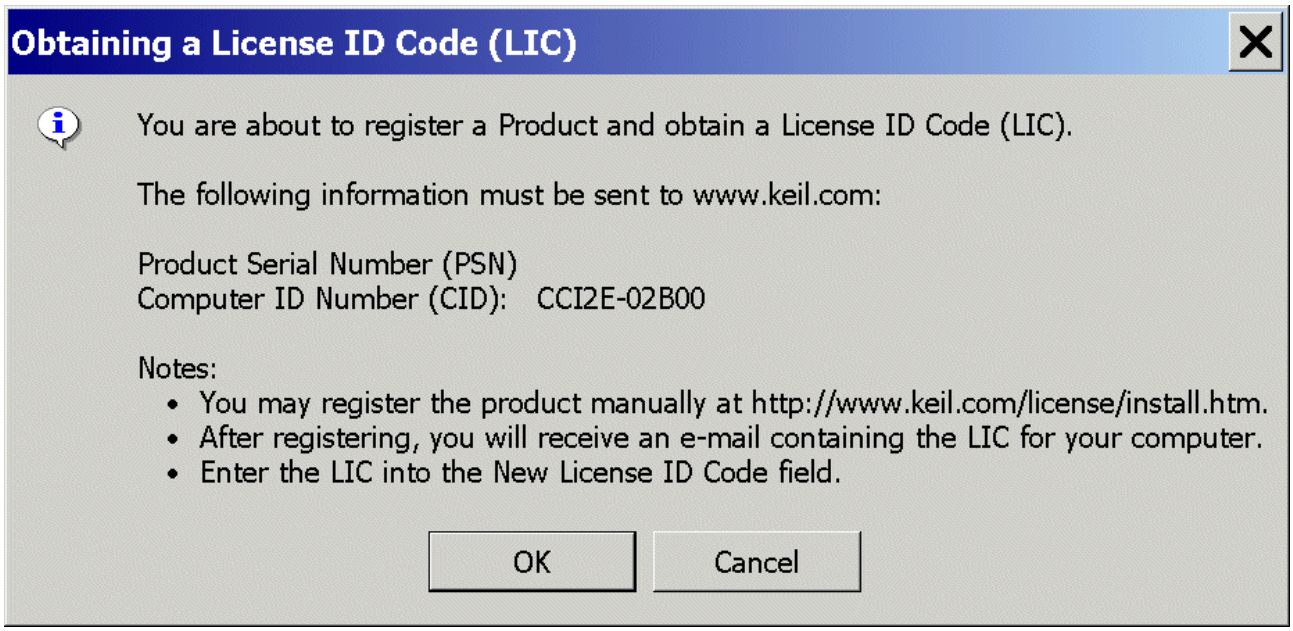
File - License Management…



License Management: Single-User License: New License ID Code (LIC):
Copy and paste your PRODUCT SERIAL NUMBER (PSN) into the New License ID Code (LIC) input field in the uVision4 License Manager Dialog.

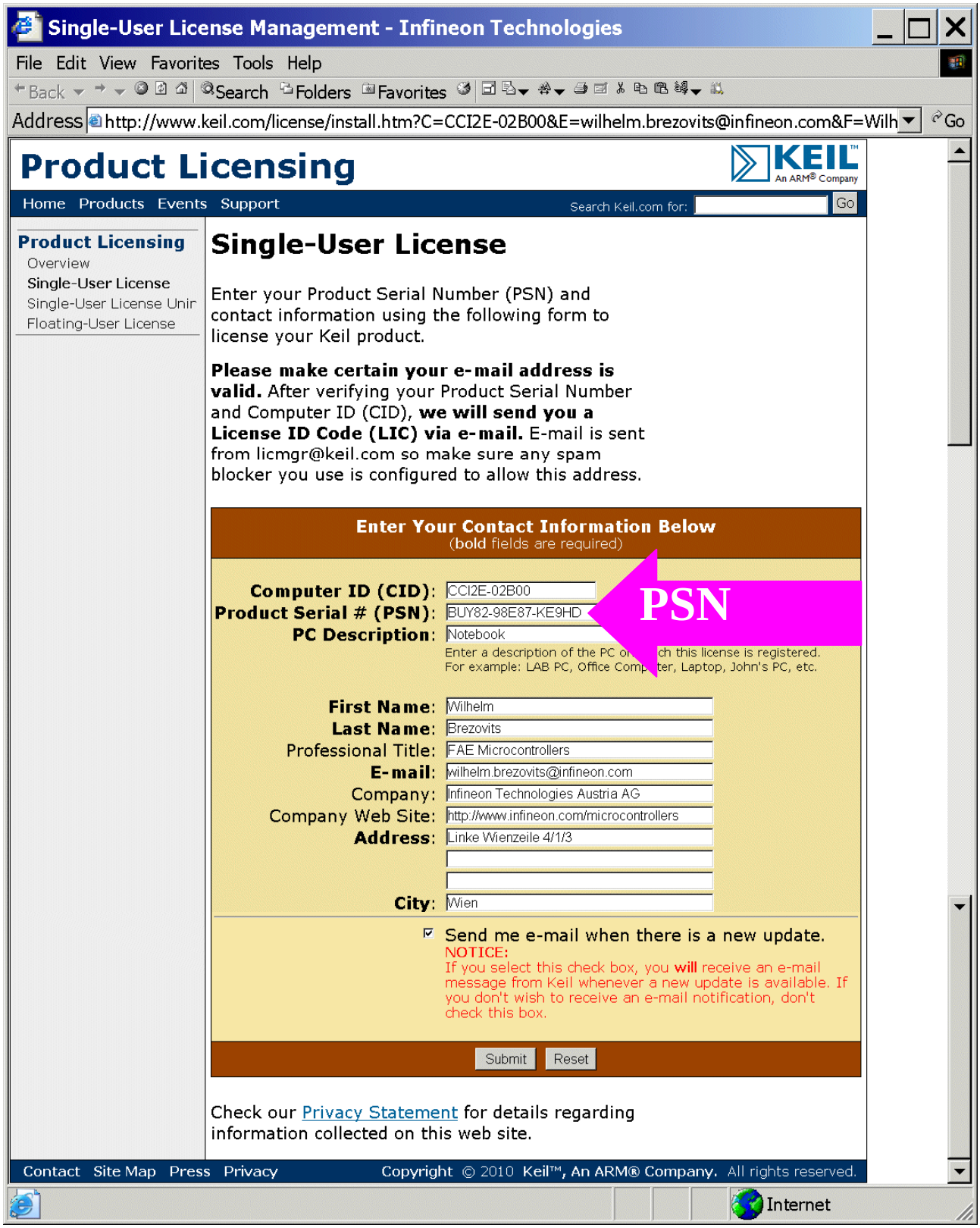


Click


Obtaining a License ID Code (LIC)
You are about to register a Product and obtain a License ID Code (LIC).
The following information must be sent to www.keil.com:
Product Serial Number (PSN)
Computer ID Number (CID): CCI2E-02B00
Notes:
• You may register the product manually at http://www.keil.com/license/install.htm.
• After registering, you will receive an e-mail containing the LIC for your computer.
• Enter the LIC into the New License ID Code field.
OK
Cancel

Click
OK

Insert your PRODUCT SERIAL NUMBER (PSN) and contact information:



Click Submit

Single-User License Management - Infineon Technologies
File Edit View Favorites Tools Help
Back Search Folders Favorites
Address http://www.keil.com/license/install.htm
Go
Product Licensing
KEIL
An ARM® Company
Home Products Events Support
Search Keil.com for:
Go
Product Licensing
Overview
Single-User License
Single-User License Unir
Floating-User License
Thanks for Licensing Your Product
We have sent your product registration information including the License ID Code (LIC) via e-mail to wilhelm.brezovits@infineon.com.
When you receive this e-mail, copy the License ID Code (LIC) and paste it into the New License ID Code input field in the µVision License Manager Dialog — Single-User License Tab (available from the File Menu).
If you have multiple Keil products you may Register Another Product at this time.
Quick Links
Licensing User's Guide
Knowledgebase
RE-LICENSING ISSUES
UNABLE TO REGISTER PRODUCT SERIAL NUMBER (PSN)
LICENSING A PC NOT CONNECTED TO THE INTERNET
Top 3 License Issues
Contact Site Map Press Privacy Copyright © 2010 Keil™, An ARM® Company. All rights reserved.
Error on page.
Internet

You've got mail!

Check your Email:

Keil Product License - Message (Plain Text)

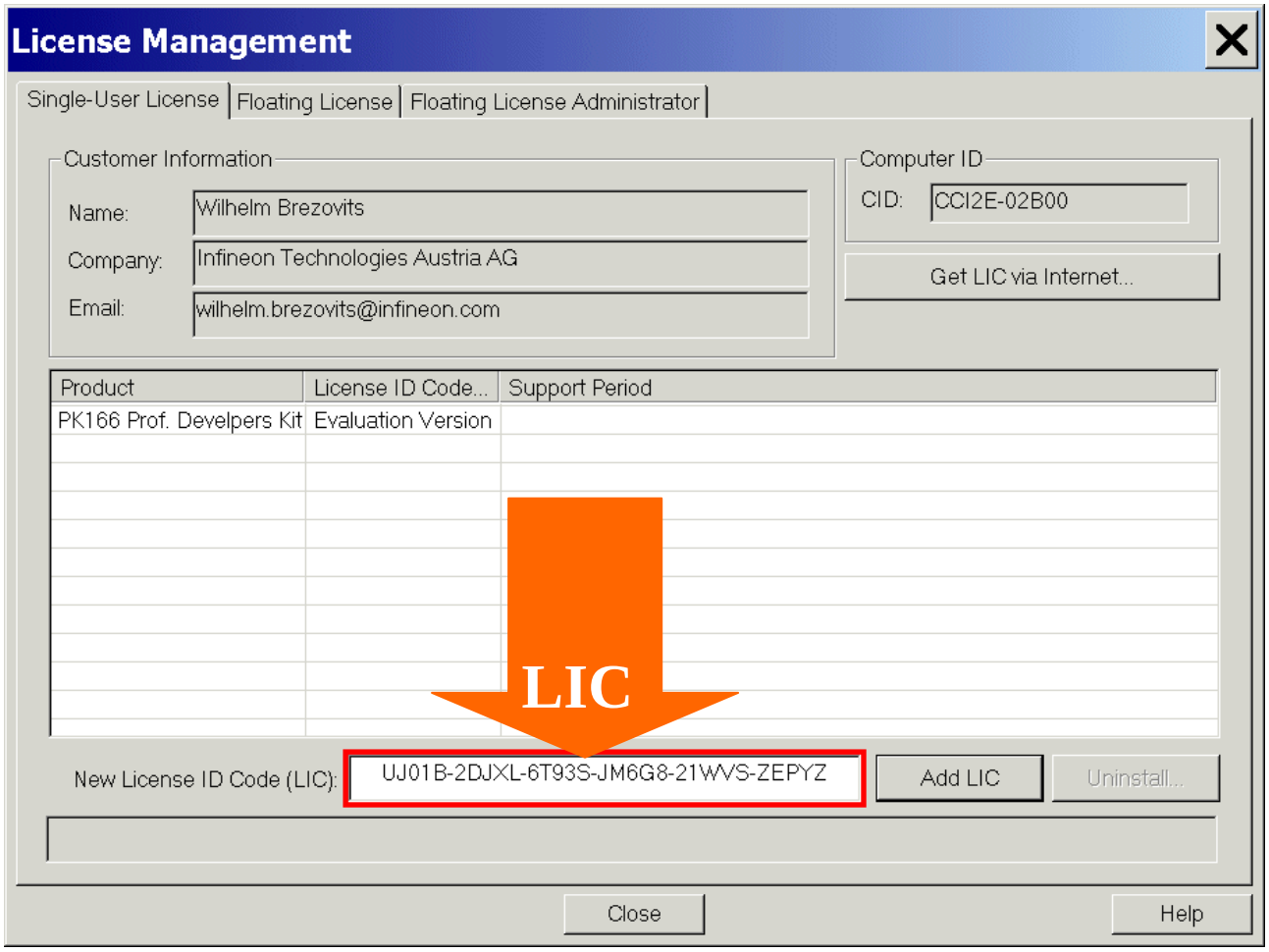
File Edit View Insert Format Tools Actions SECUDE secure mail Help

Reply | Reply to All | Forward

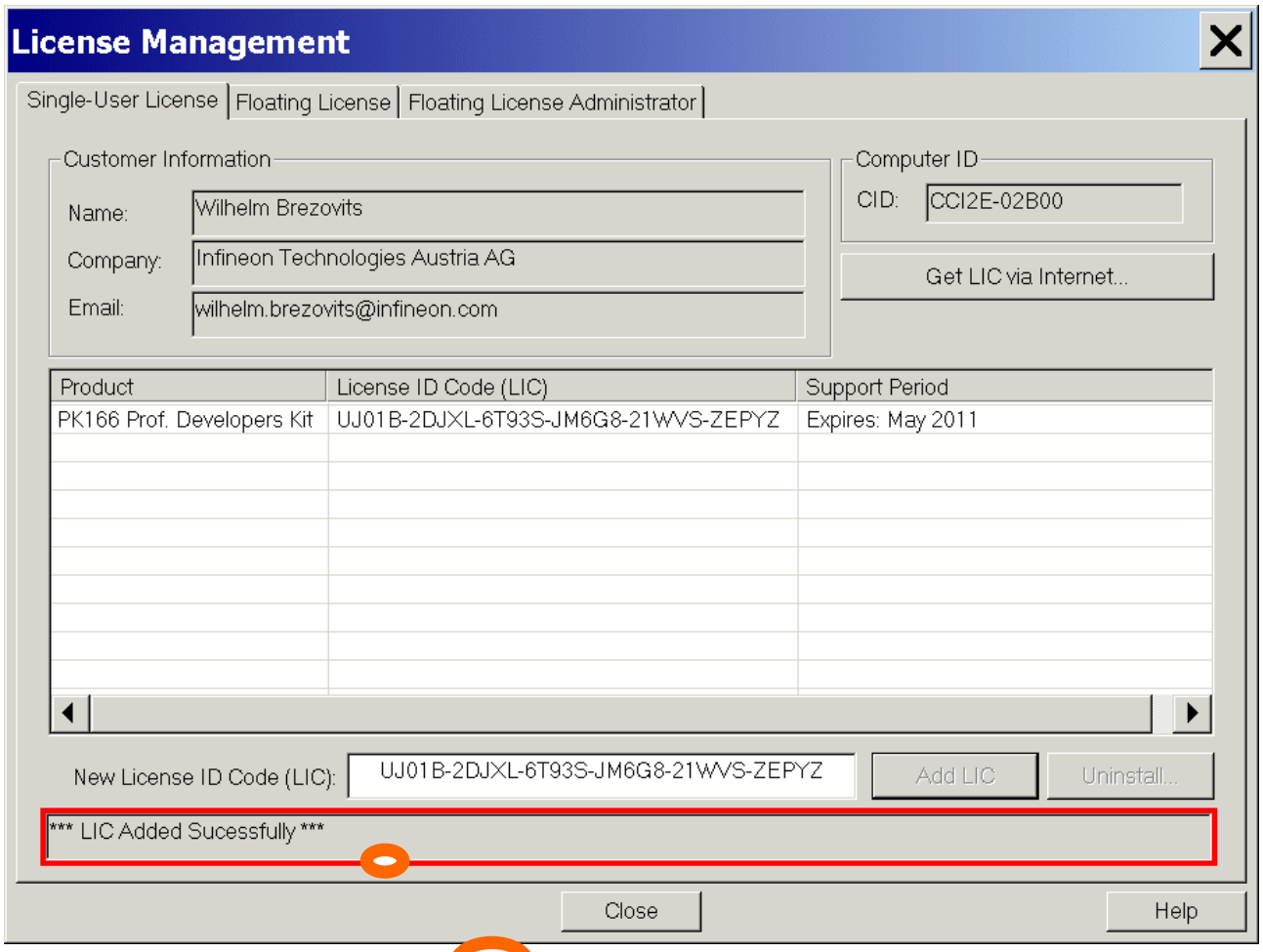
From: licmgr@keil.com
Sent: Mi 14.04.2010 14:16
To: Brezovits Wilhelm (IFAT SMD IMM FAE)
Cc:
Subject: Keil Product License

```
Thank you for licensing your Keil product.  Your License ID Code (LIC) is
printed below.  Print a copy of this e-mail to save with your installation CD.

PK166 Professional Developer's Kit
Support Ends 31 May 2011

PC Description       : NOTEBOOK
Computer ID     (CID): CCI2E-02B00
Serial Number   (PSN): BUY82-98E87-KE9HD

License ID Code (LIC): UJ01B-2DJXL-6T93S-JM6G8-21WVS-ZEPYZ


To activate your Keil product, copy the License ID Code (LIC) and paste it
into the New License ID Code input field on the Single-User License Tab in the
uVision3 License Manager Dialog (available from the File menu).

*** DO NOT REPLY TO THIS EMAIL:  For licensing problems or questions, please
contact Keil Technical Support.

Thank You,
Technical Support


Keil, An Arm Company
1501 10th Street, Suite 110
Plano, Texas 75074  USA
Phone: +1 972-312-1107
FAX: +1 972-312-1159
http://www.keil.com/

IMPORTANT NOTICE: The contents of this email and any attachments are
confidential and may also be privileged. If you are not the named recipient,
please notify the sender immediately and do not disclose the contents to any
other person, use it for any purpose, or store or copy the information in any
medium.
```

Insert your License ID Code (LIC):



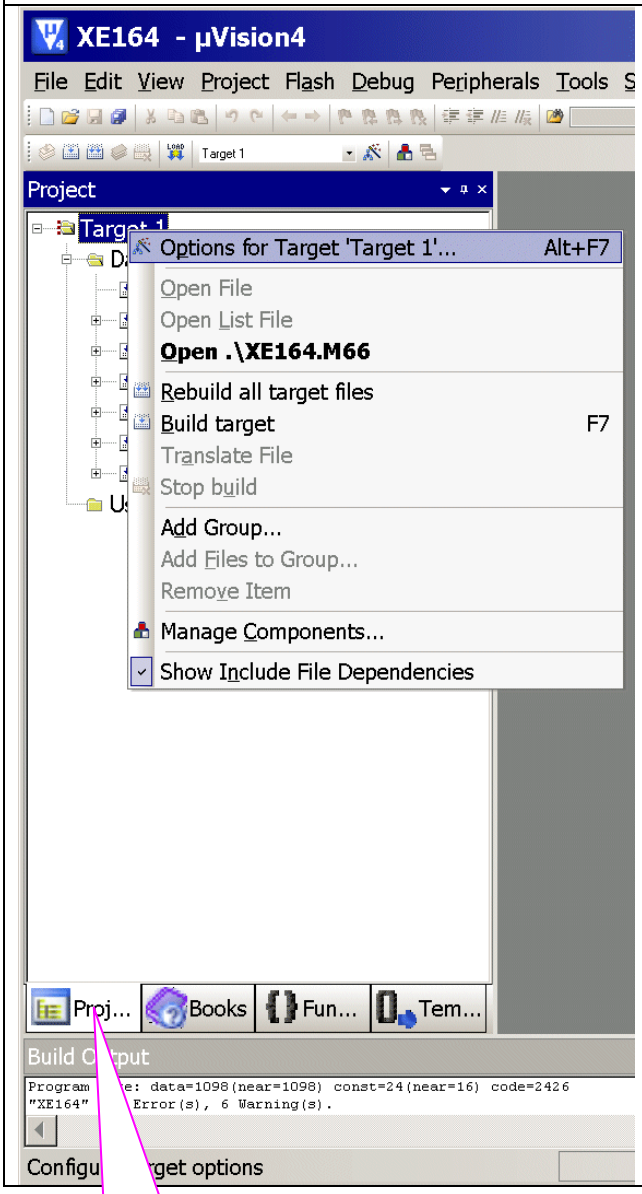
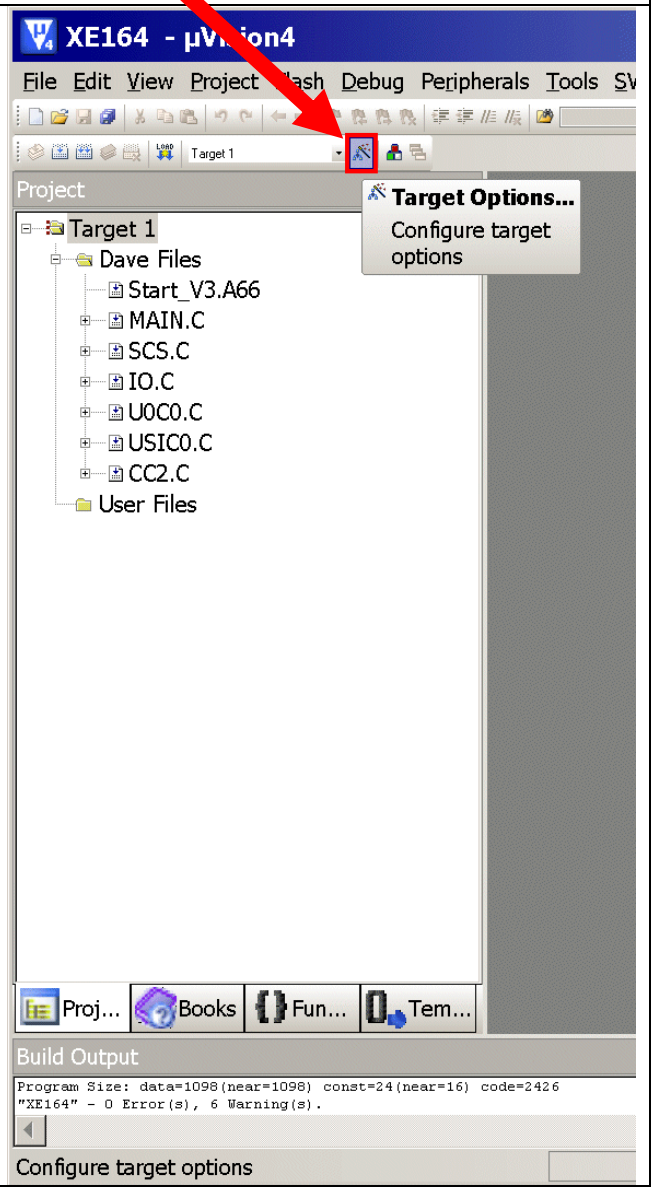
Click



Click

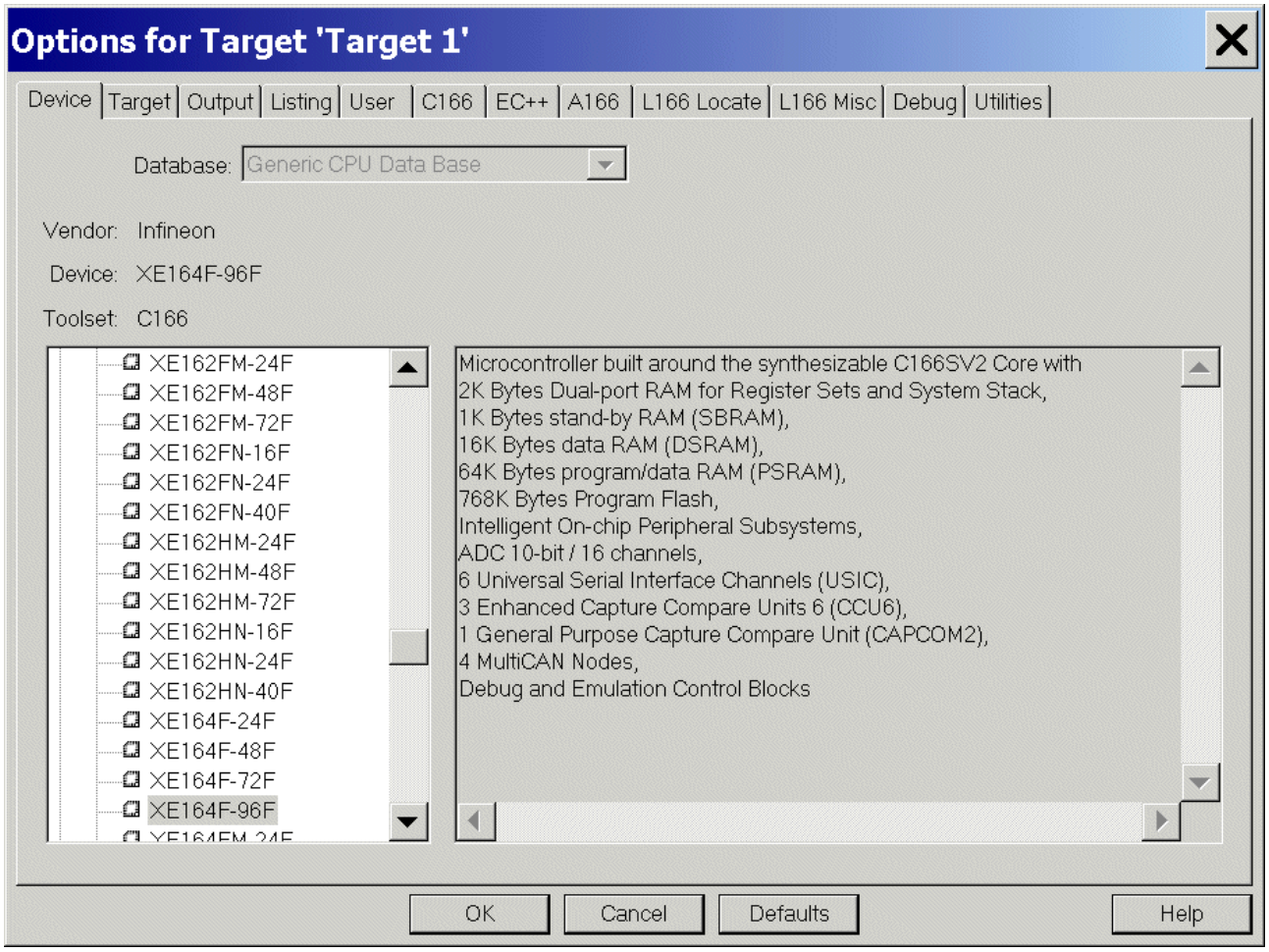
Note:
Now you can write software for Infineon's 16 bit microcontrollers without the limitations (4 KBytes) of the Evaluation Compiler. That means your program may be as large as there is program memory available.

Configure:
Compiler, Assembler, Linker, Locater, Hex-Converter, Build – Control, (Simulator, Debugger,)
Listings and Utilities (e.g. OnChip Flash Programming):

mouse position: (Project Workspace, Files): Target1 click right mouse button click Options for Target ´Target1‘	or	click
		

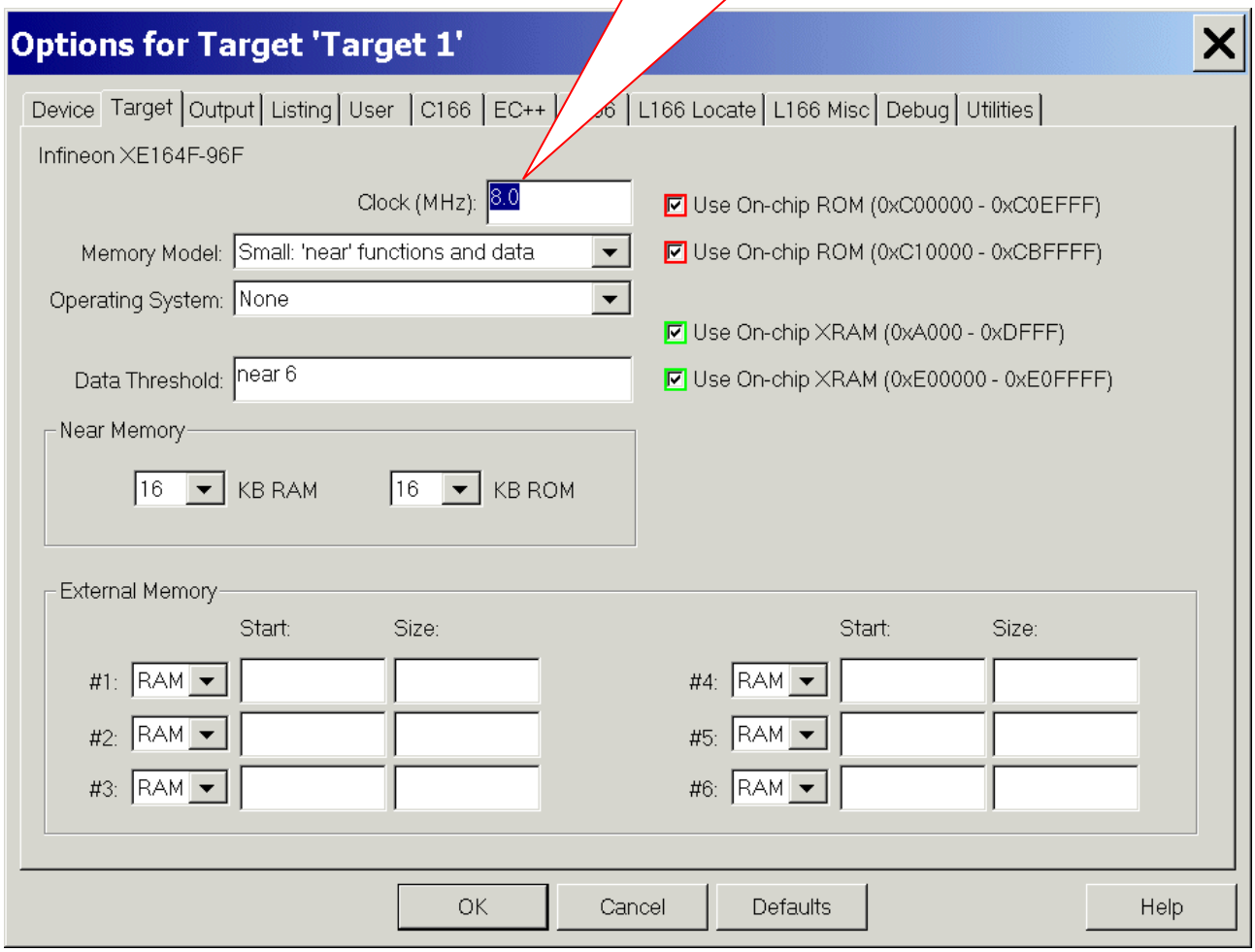
Project Workspace, Files

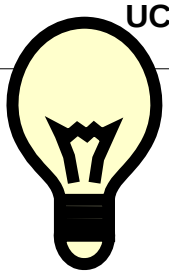
Device: check XE164F-96F



Target: Clock(MHz): check 8.0
Target: tick/check ☑ Use On-chip ROM
Target: tick/check ☑ Use On-chip ROM
Target: tick/check ☑ Use On-chip XRAM
Target: tick/check ☑ Use On-chip XRAM

Note (Source: DAvE):
Configuration of the System Clock:
- VCO clock used, input clock is connected
- input frequency is 8,00 MHz
- configured system frequency is 66,00 MHz
- system clock is 66.00 MHz





Additional information: Memory Map (Source: User's Manual):

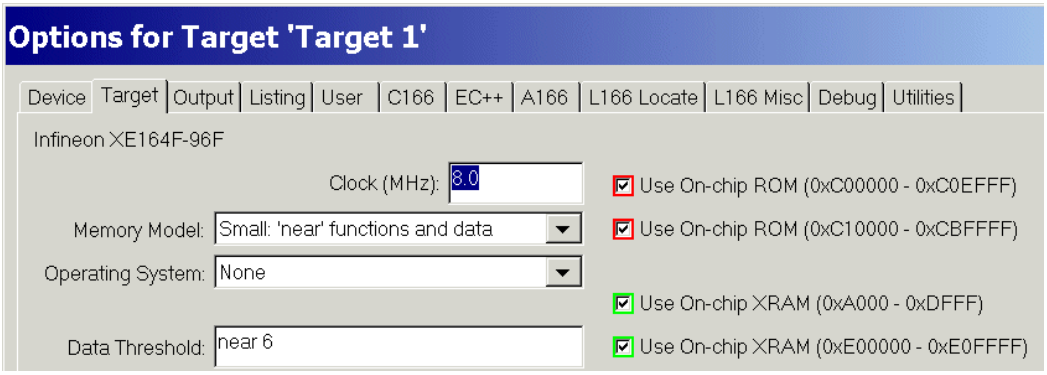
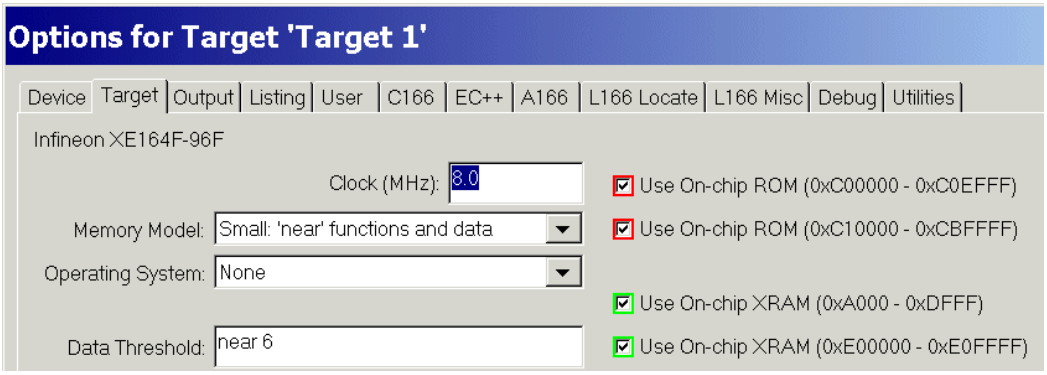
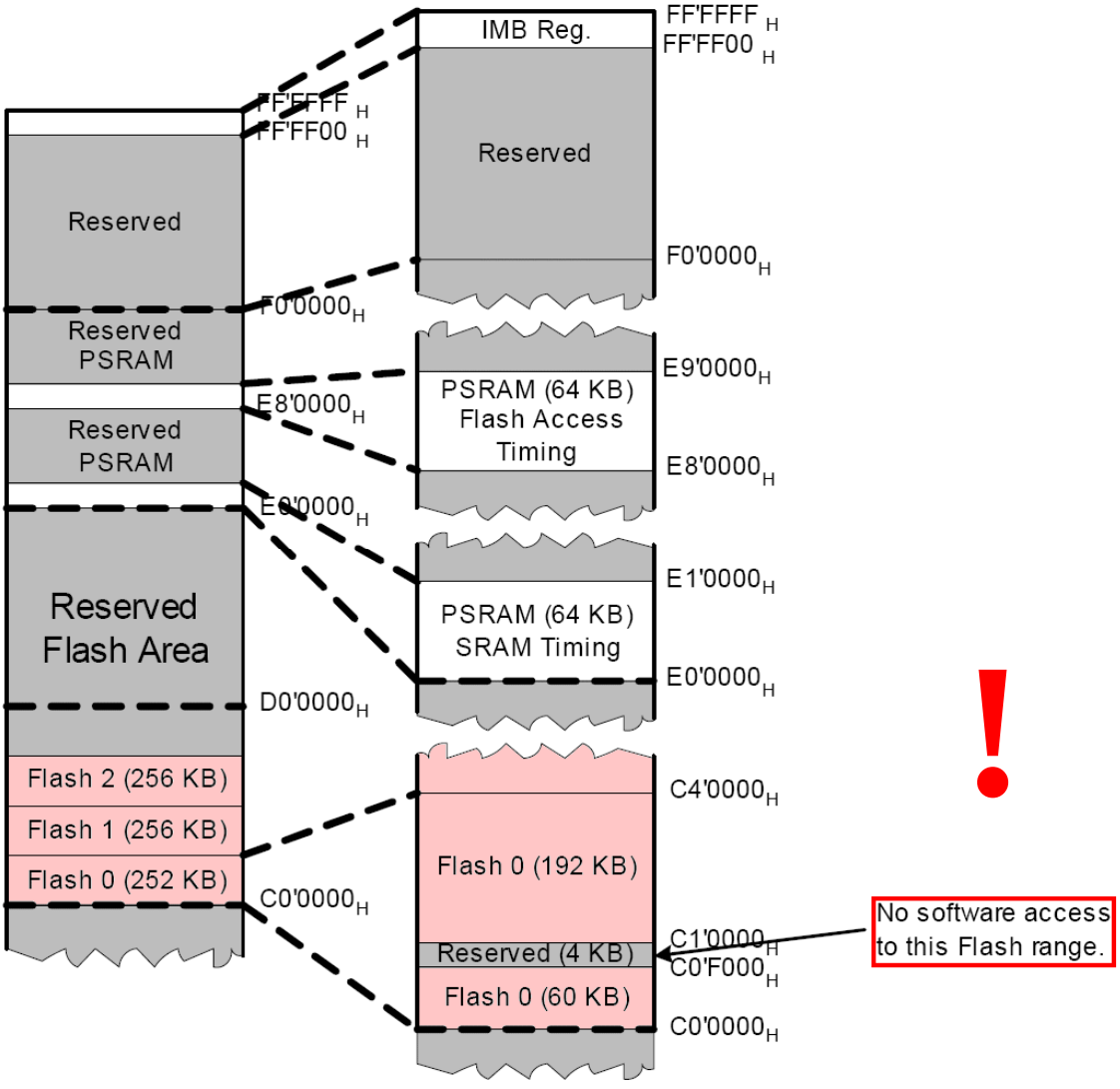


Table 3-1 XE16x Memory Map [1)]

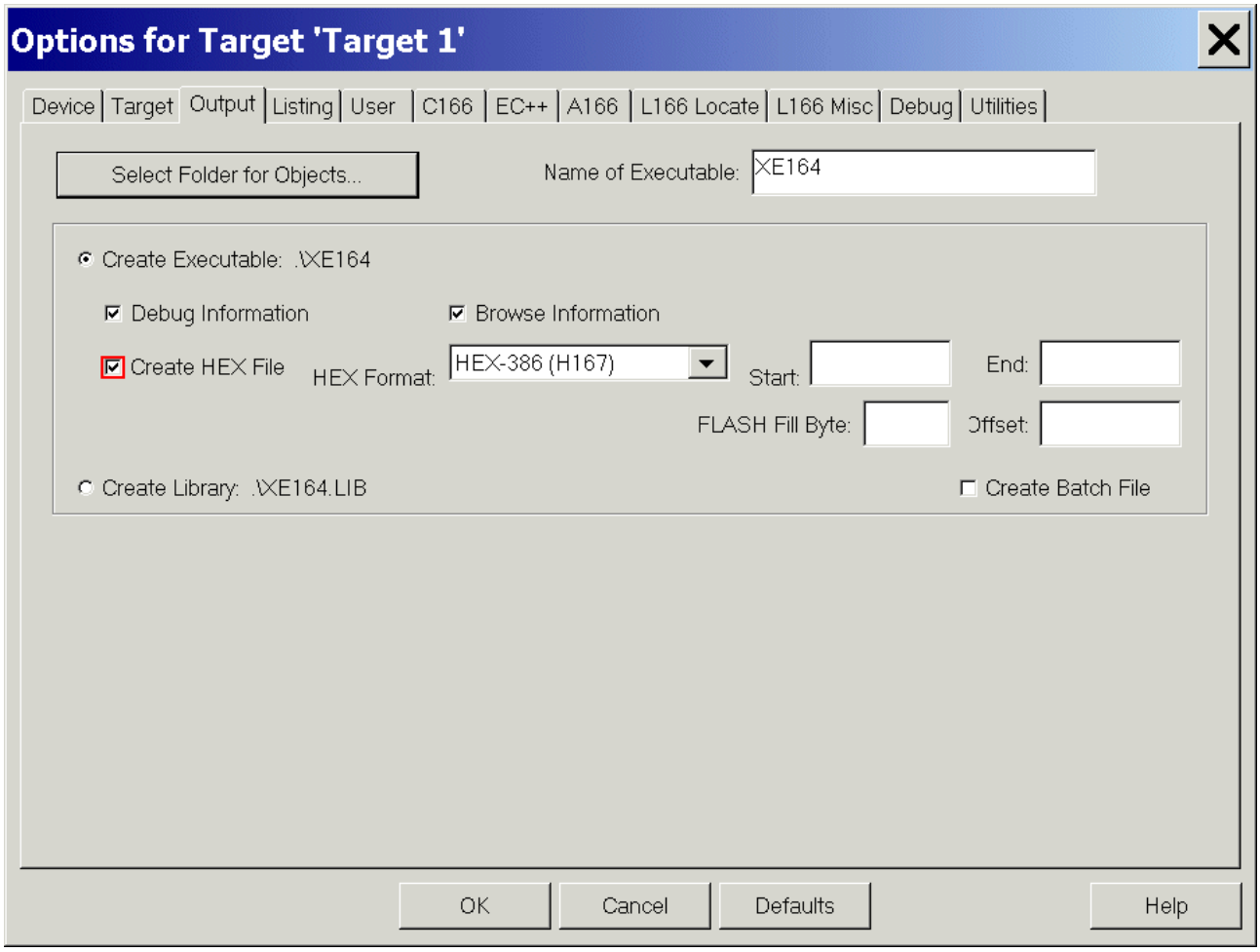
Address Area	Start Loc.	End Loc.	Area Size[2)]	Notes
IMB register space	$FF'FF00_H$	$FF'FFFF_H$	256 Bytes	
Reserved (access trap)	$F0'0000_H$	$FF'FEFF_H$	< 1 MByte	Minus IMB registers.
Reserved for EPSRAM	$E9'0000_H$	$EF'FFFF_H$	448 KBytes	
EPSRAM	$E8'0000_H$	$E8'FFFF_H$	64 KBytes	PSRAM with Flash timing.
Reserved for PSRAM	$E1'0000_H$	$E7'FFFF_H$	448 KBytes	
PSRAM	$E0'0000_H$	$E0'FFFF_H$	64 KBytes	Program SRAM.
Reserved for Flash	$CC'0000_H$	$DF'FFFF_H$	<1.25 MBytes	
Flash 2	$C8'0000_H$	$CB'FFFF_H$	256 KBytes	
Flash 1	$C4'0000_H$	$C7'FFFF_H$	256 KBytes	
Flash 0	$C0'0000_H$	$C3'FFFF_H$	252 KBytes[3)]	Minus res. seg.
External memory area	$40'0000_H$	$BF'FFFF_H$	8 MBytes	
External IO area[4)]	$20'5800_H$	$3F'FFFF_H$	< 2 MBytes	Minus CAN/USIC
USIC registers	$20'4000_H$	$20'57FF_H$	6 KBytes	Accessed via EBC
MultiCAN registers	$20'0000_H$	$20'3FFF_H$	16 KBytes	Accessed via EBC
External memory area	$01'0000_H$	$1F'FFFF_H$	< 2 MBytes	Minus segment 0
SFR area	$00'FE00_H$	$00'FFFF_H$	0.5 KBytes	
Dual-port RAM (DPRAM)	$00'F600_H$	$00'FDFF_H$	2 KBytes	
Reserved for DPRAM	$00'F200_H$	$00'F5FF_H$	1 KBytes	
ESFR area	$00'F000_H$	$00'F1FF_H$	0.5 KBytes	
XSFR area	$00'E000_H$	$00'EFFF_H$	4 KBytes	
Data SRAM (DSRAM)	$00'A000_H$	$00'DFFF_H$	16 KBytes	
Reserved for DSRAM	$00'8000_H$	$00'9FFF_H$	8 KBytes	
External memory area	$00'0000_H$	$00'7FFF_H$	32 KBytes	

Additional information: Memory Map (Source: User's Manual):



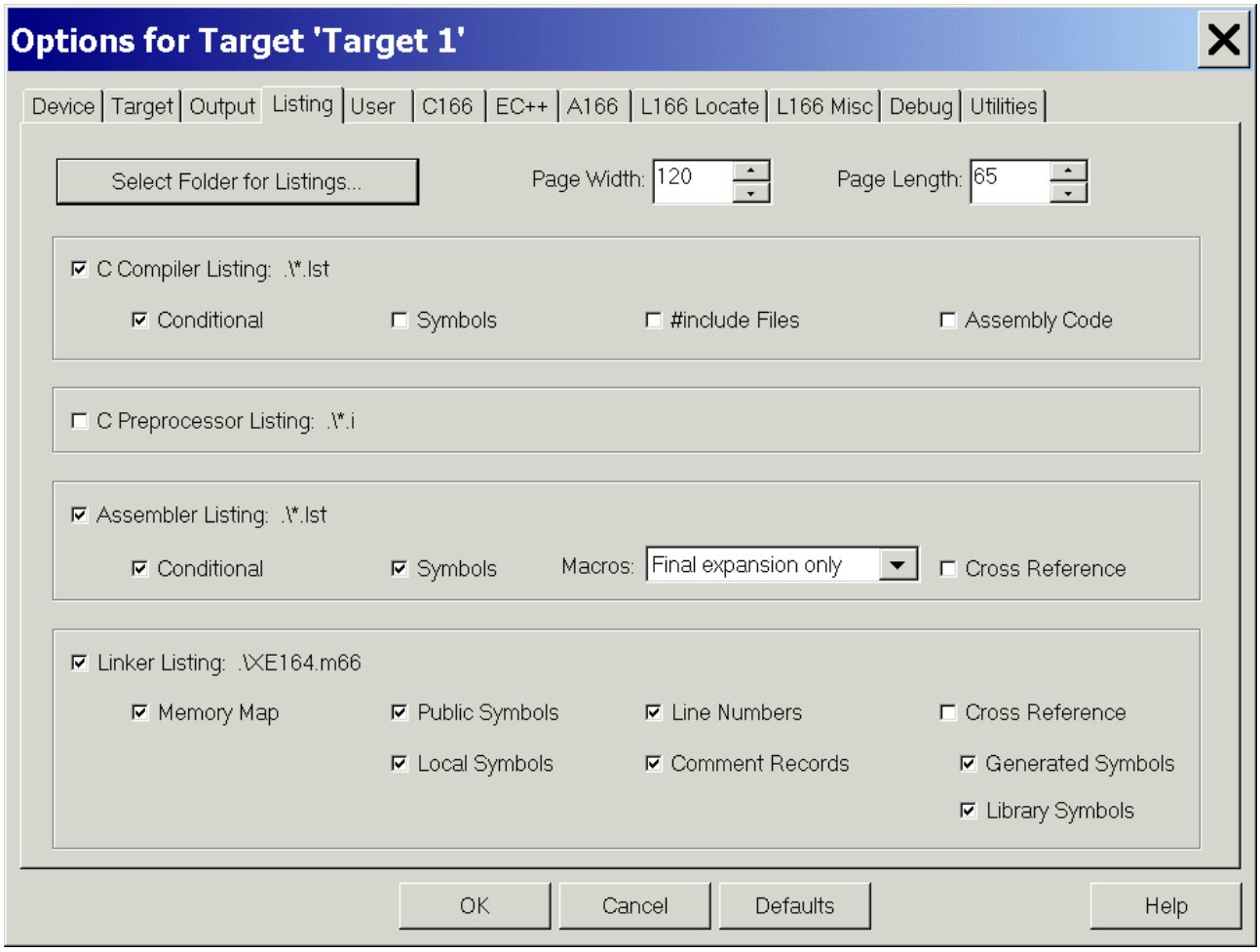


Output: tick ☑ Create HEX File

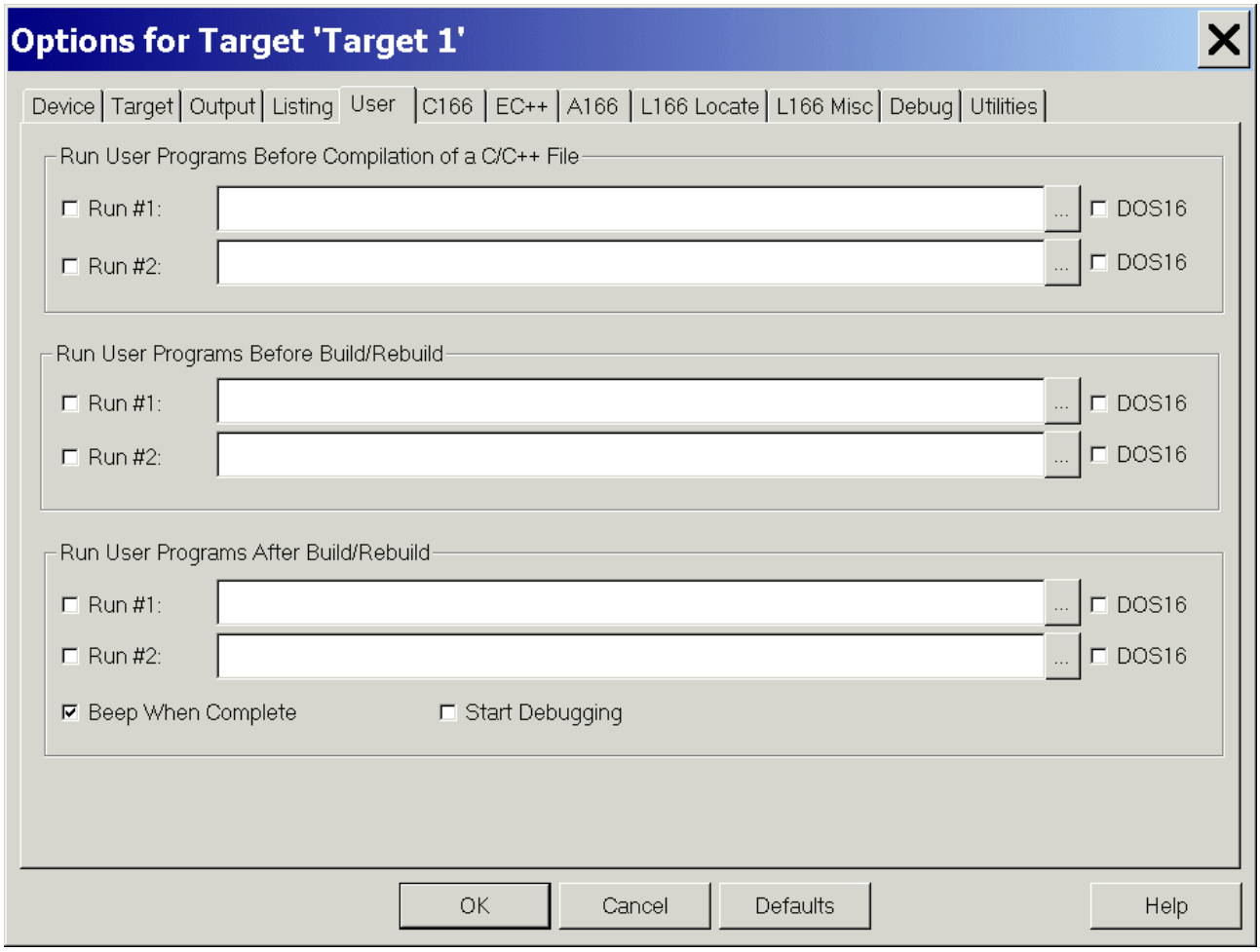


Note:
The HEX File could be used while working with the program
MEMTOOL for OnChip-Flash-Programming
via RS232-interface [Bootstrap Loader (BSL) Mode via UART/USIC0_CH0].

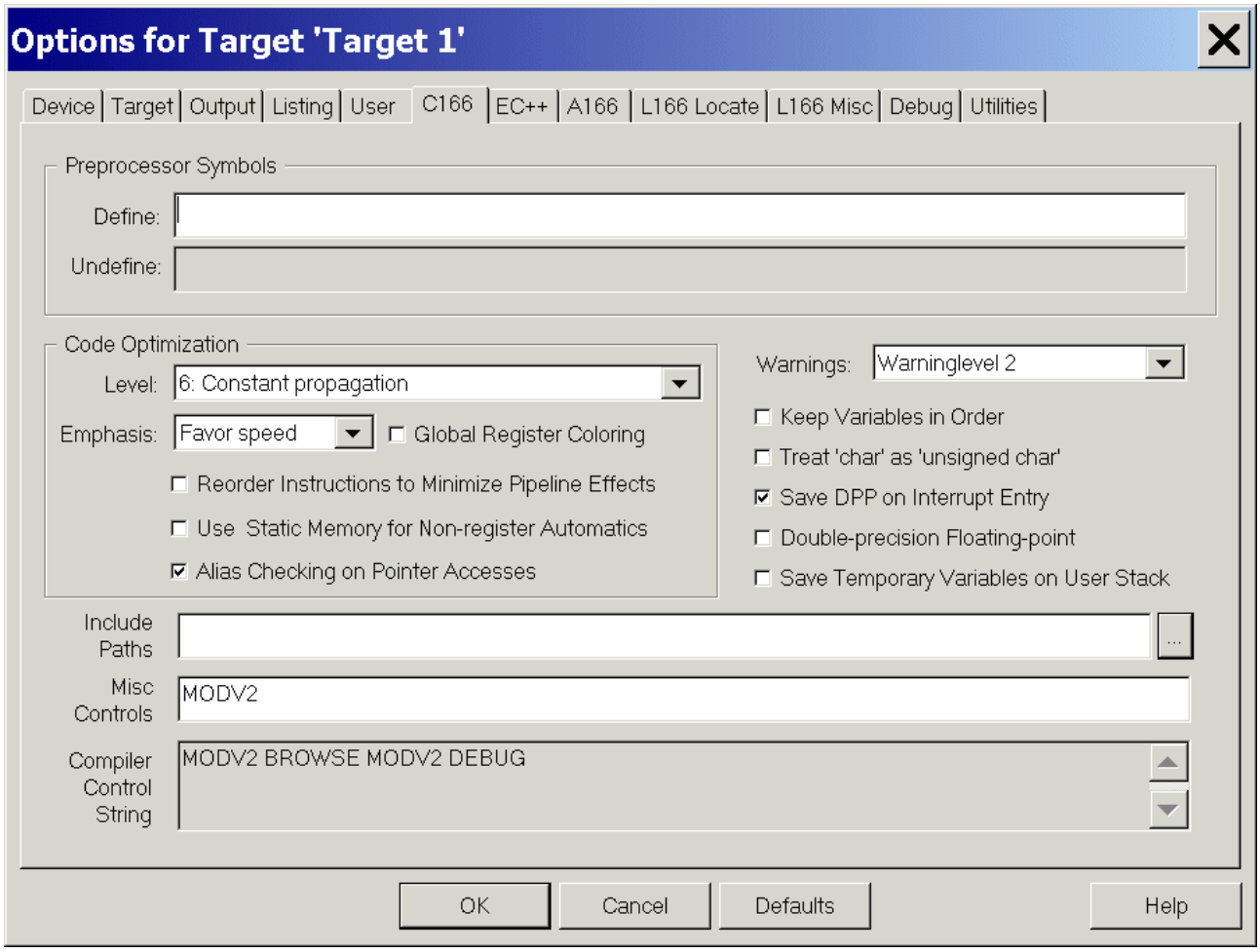
Listing: (do nothing)



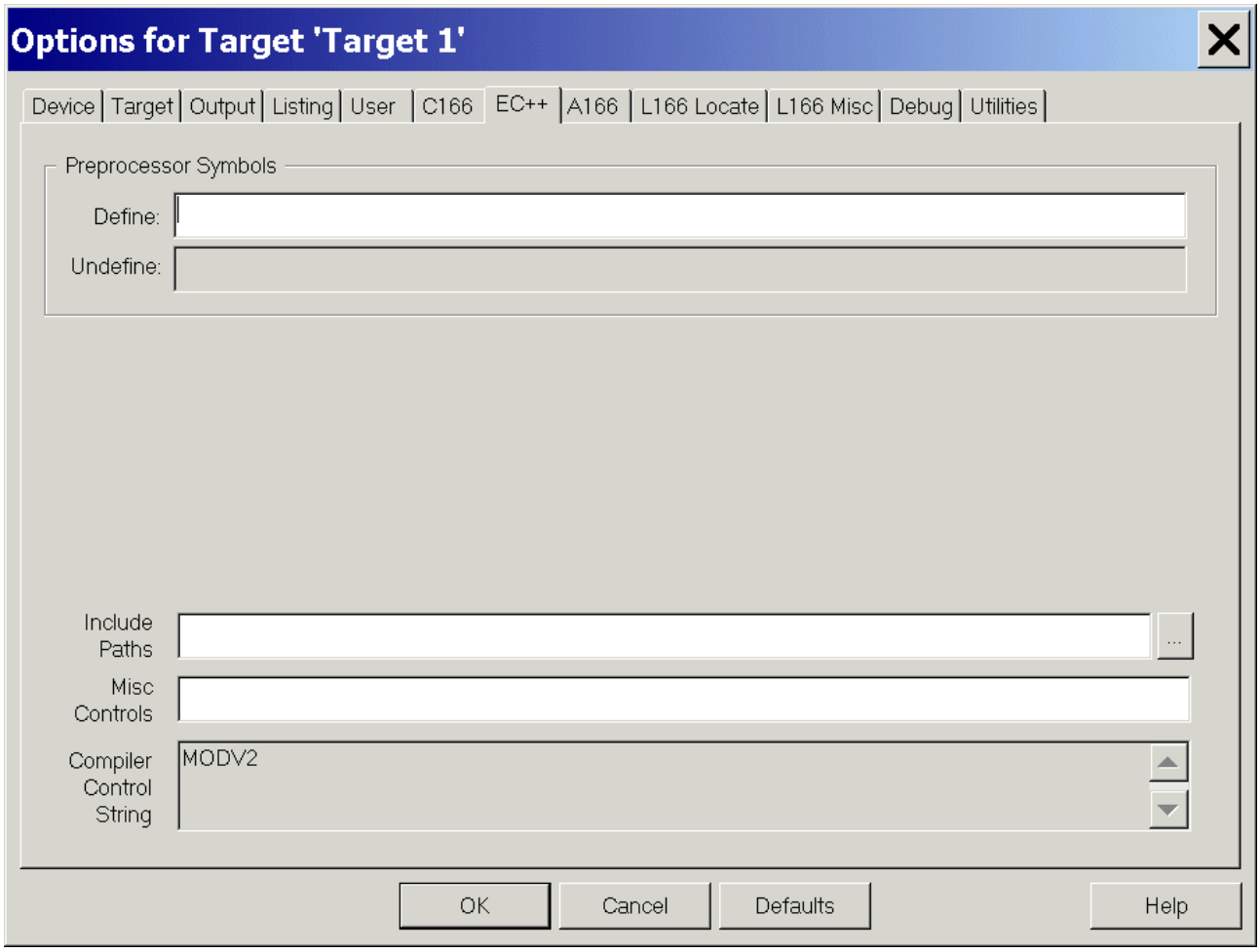
User: (do nothing)



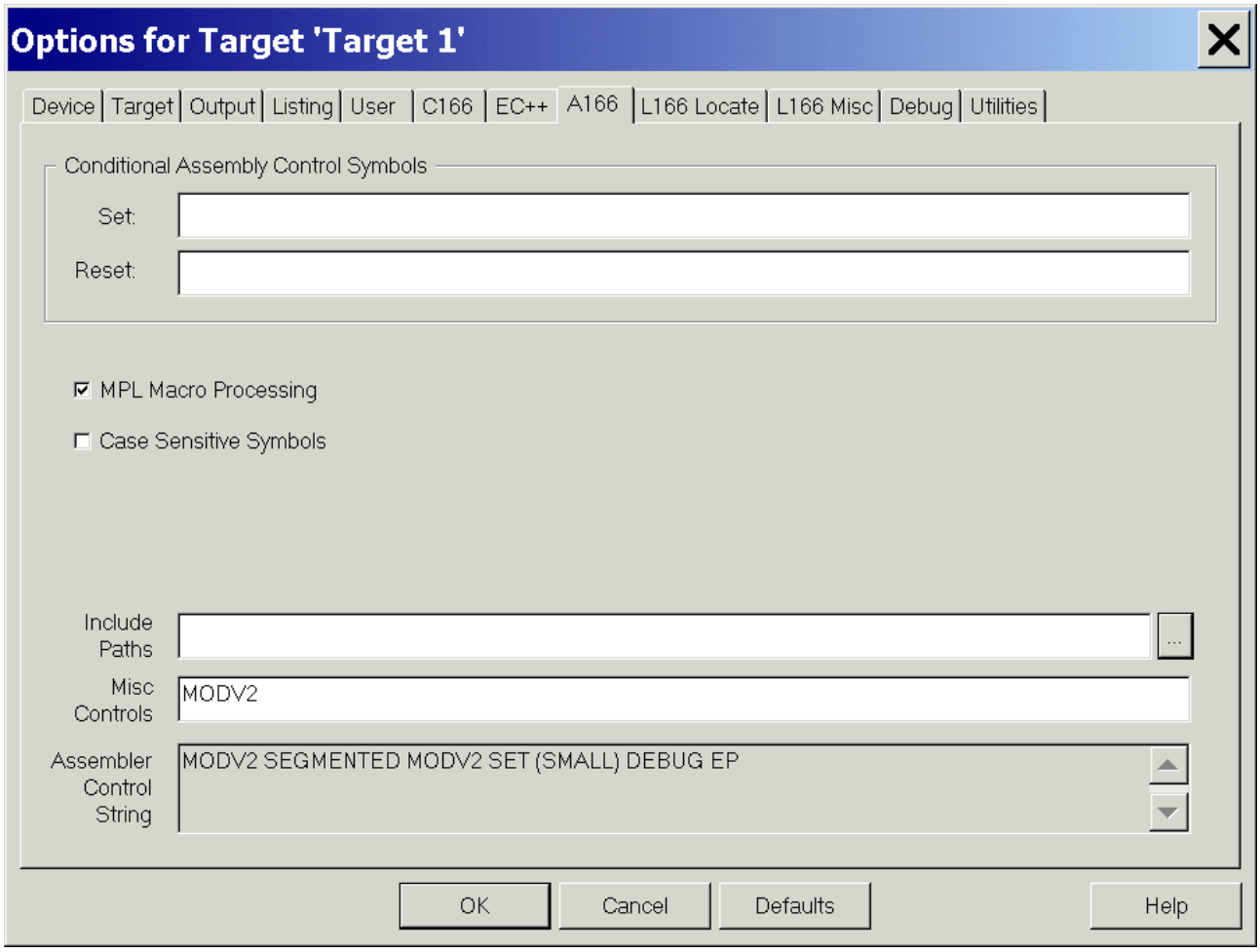
C166: (do nothing)



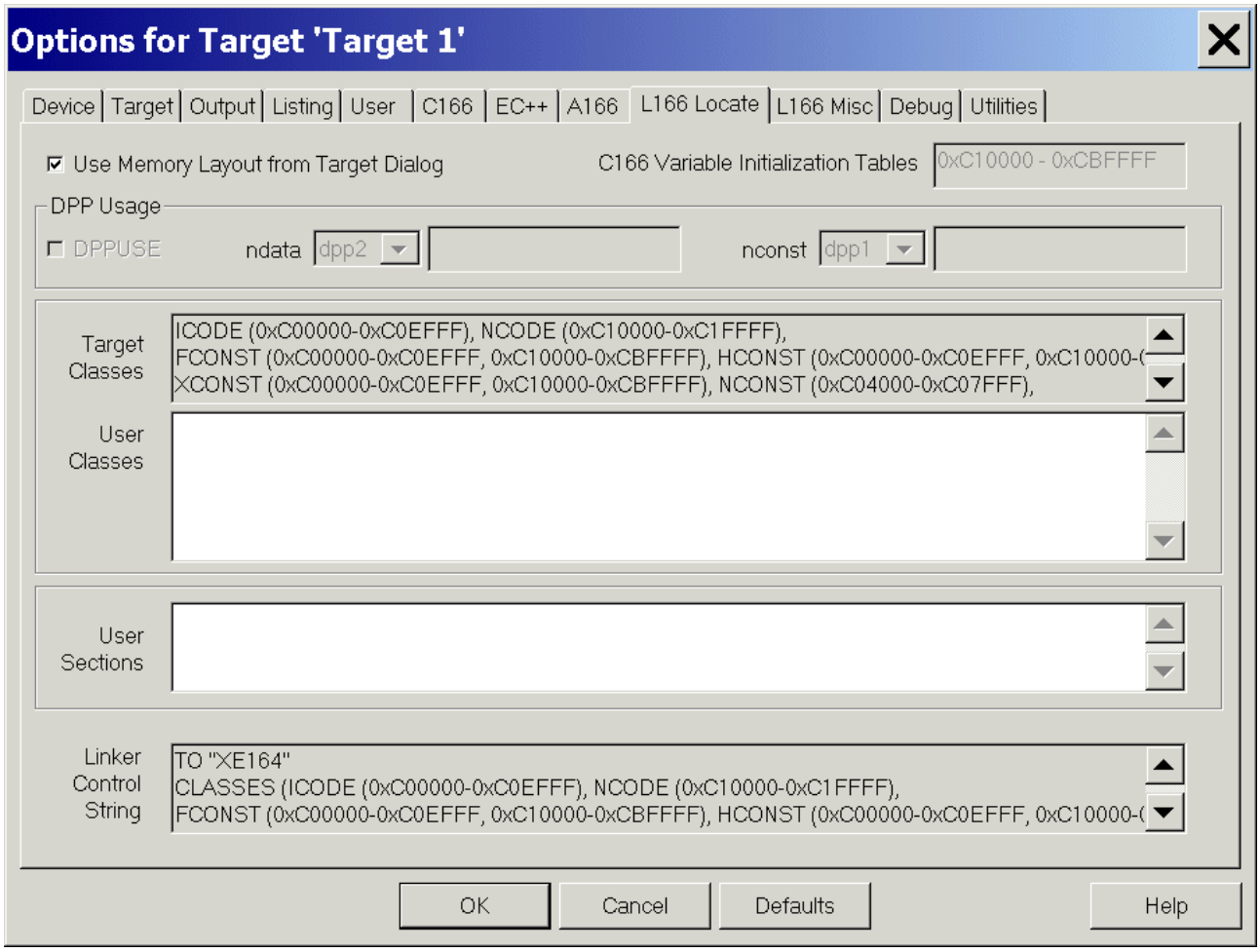
EC++: (do nothing)



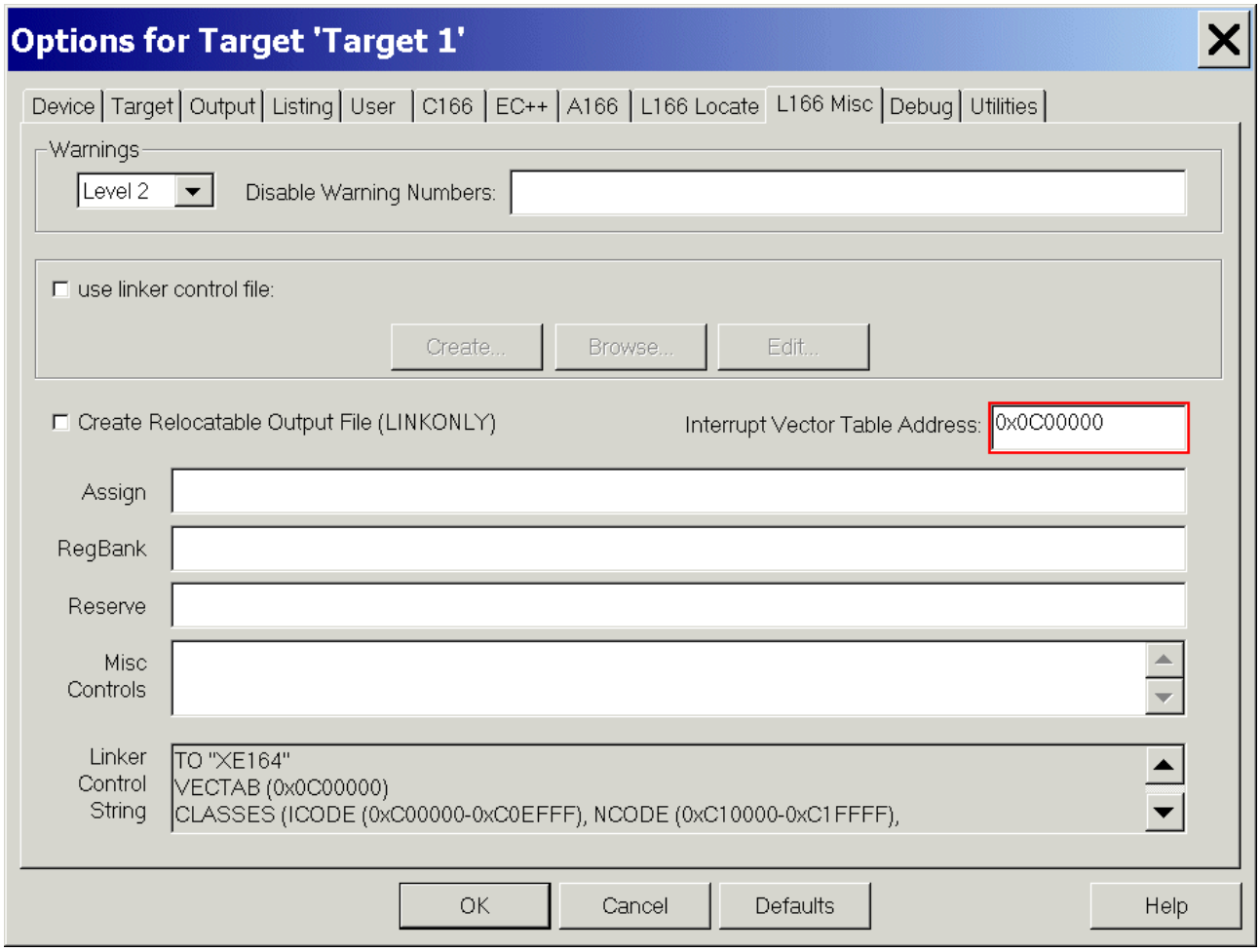
A166: (do nothing)



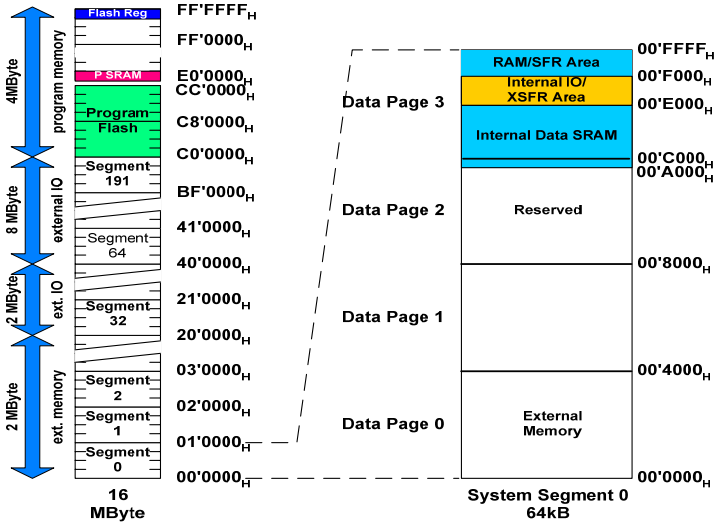
L166 Locate: (do nothing)



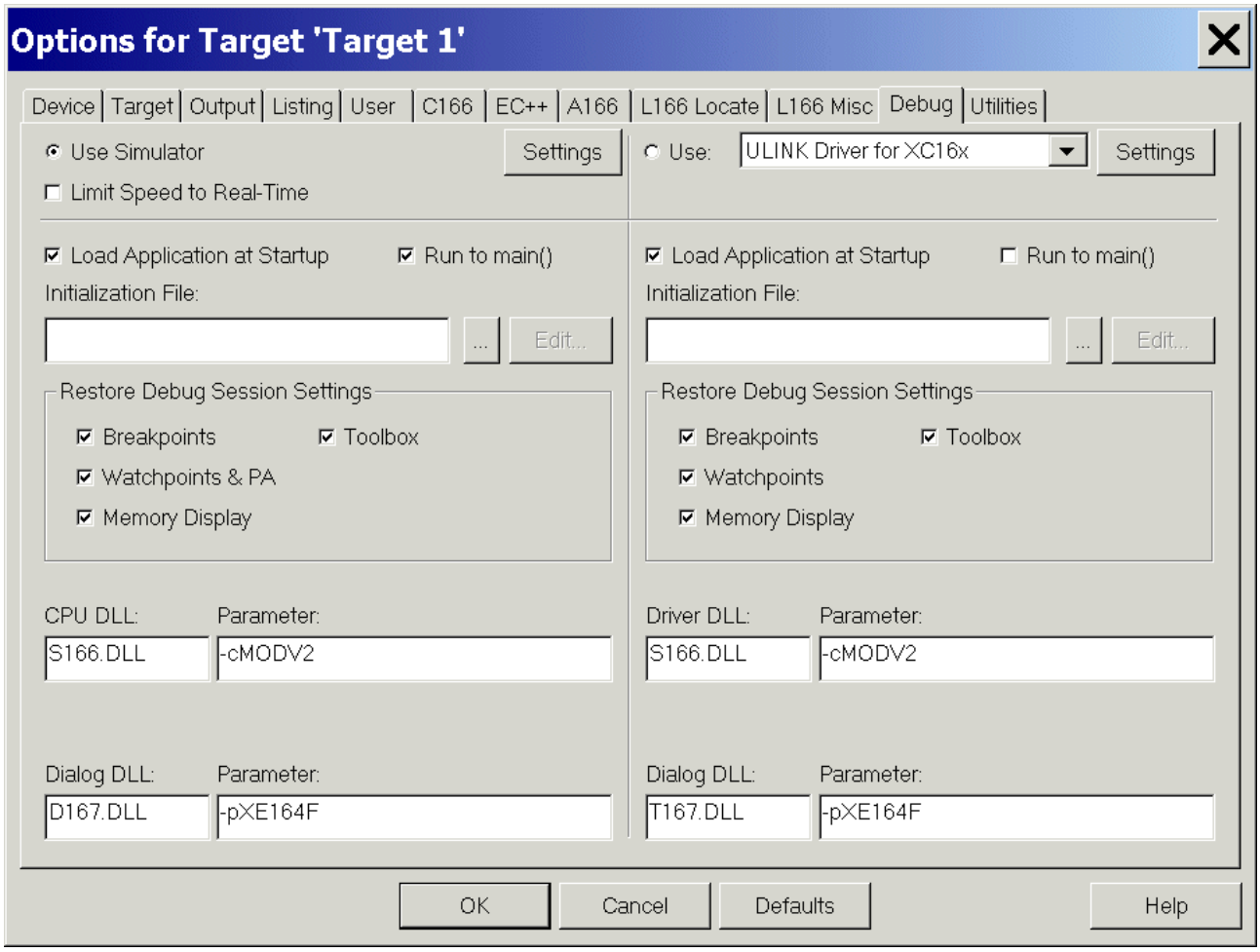
L166 Misc: Interrupt Vector Table Address: insert 0x0C00000



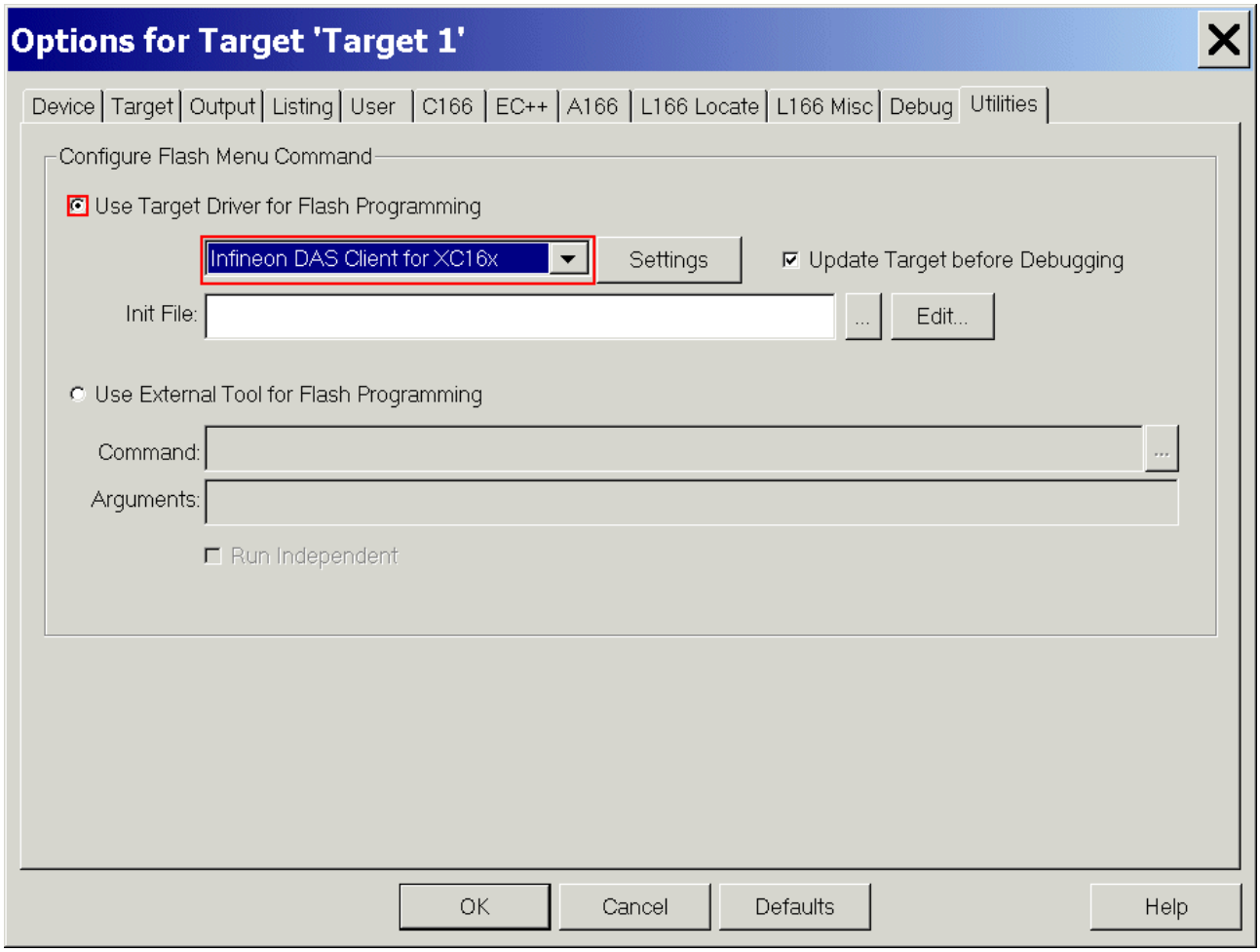
Note:
The On Chip Flash starts here.



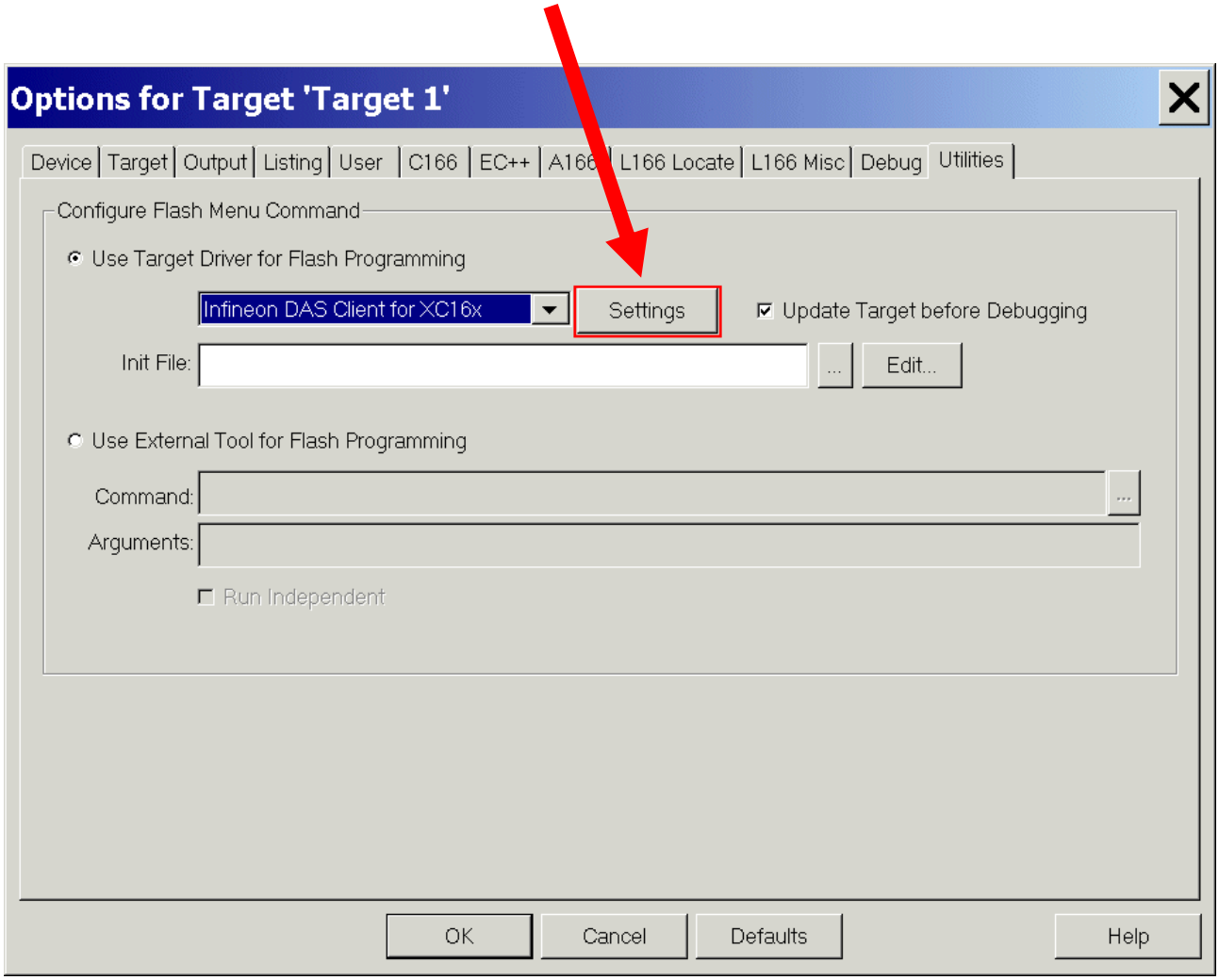
Debug: (do nothing)



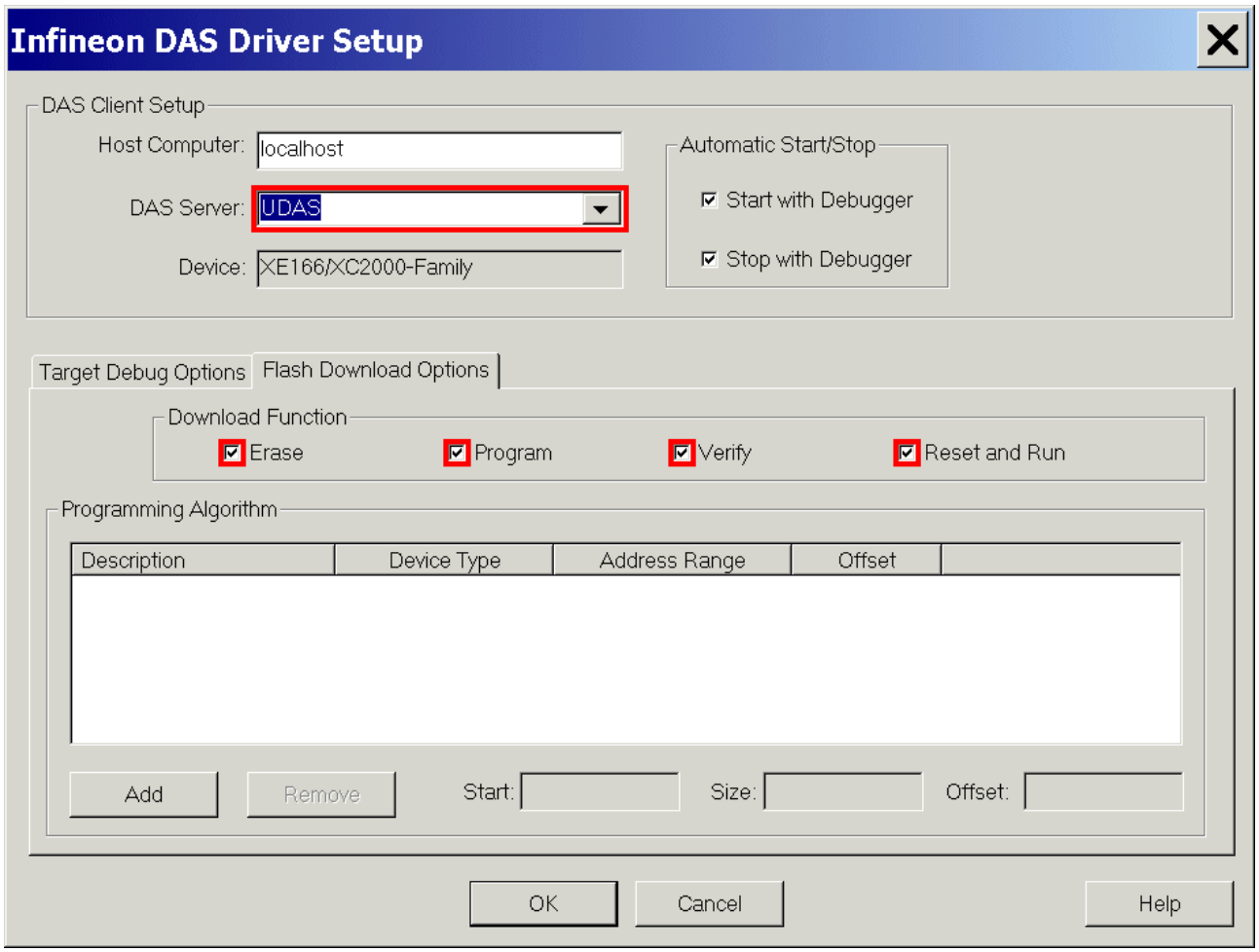
Utilities: Configure Flash Menu Command: check ⊙ Use Target Driver for Flash Programming
Utilities: Configure Flash Menu Command: select Infineon DAS Client for XC16x



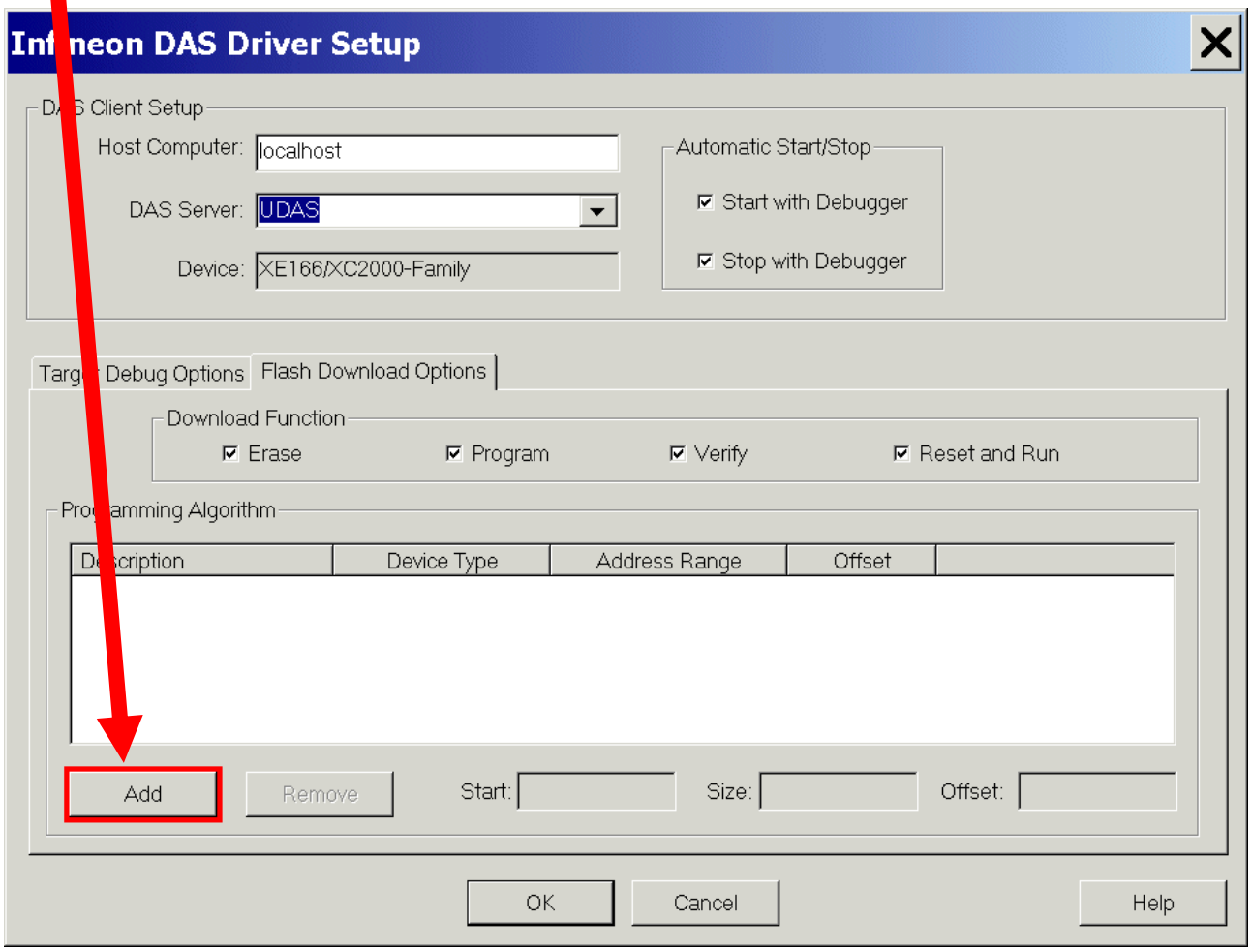
Utilities: Configure Flash Menu Command: click Settings



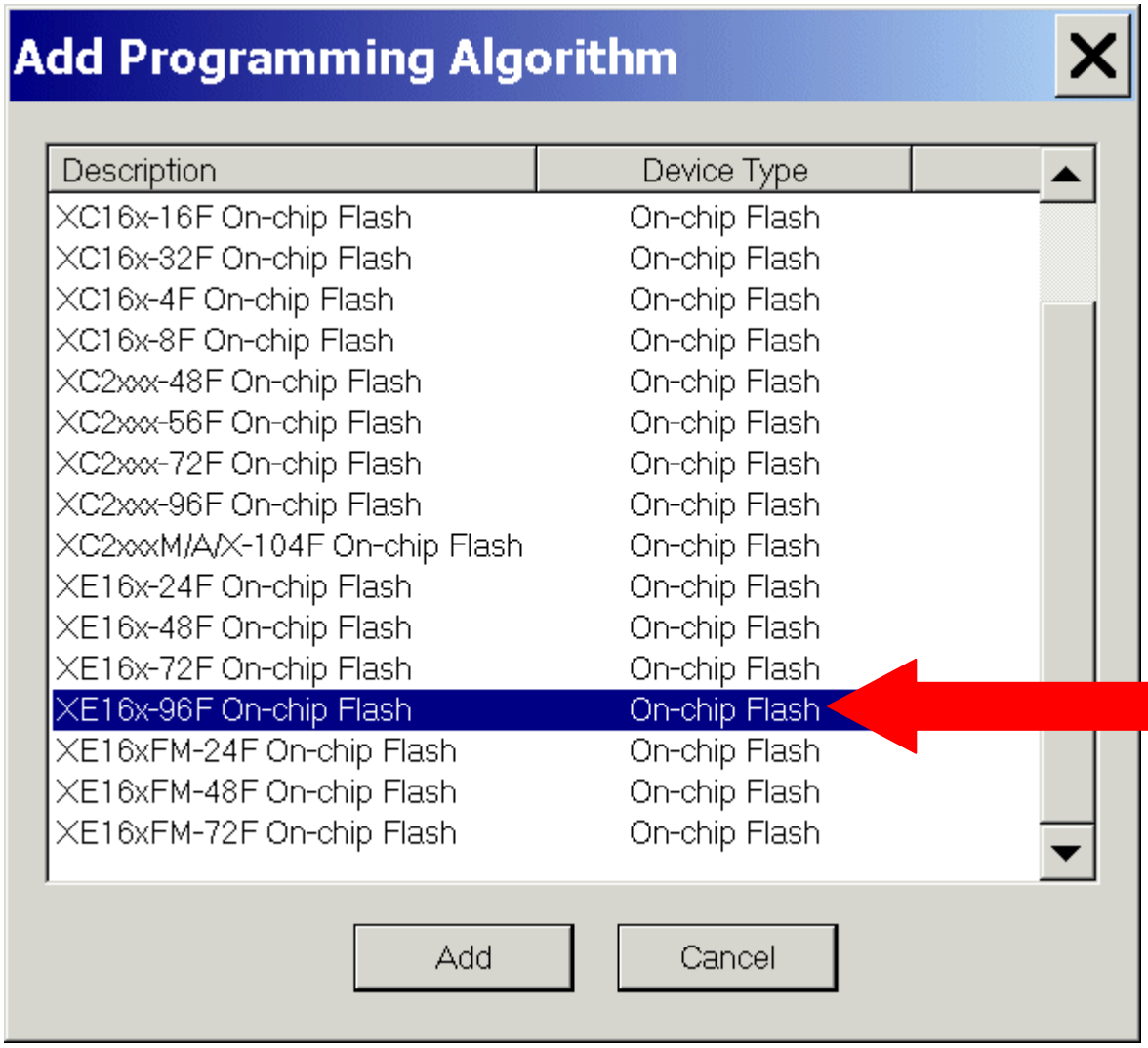
DAS Client Setup: DAS Server: select: UDAS (Universal Device Access Server)
Flash Download Options: Download Function: tick/check: ☑ Erase
Flash Download Options: Download Function: tick/check: ☑ Program
Flash Download Options: Download Function: tick/check: ☑ Verify
Flash Download Options: Download Function: tick/check: ☑ Reset and Run



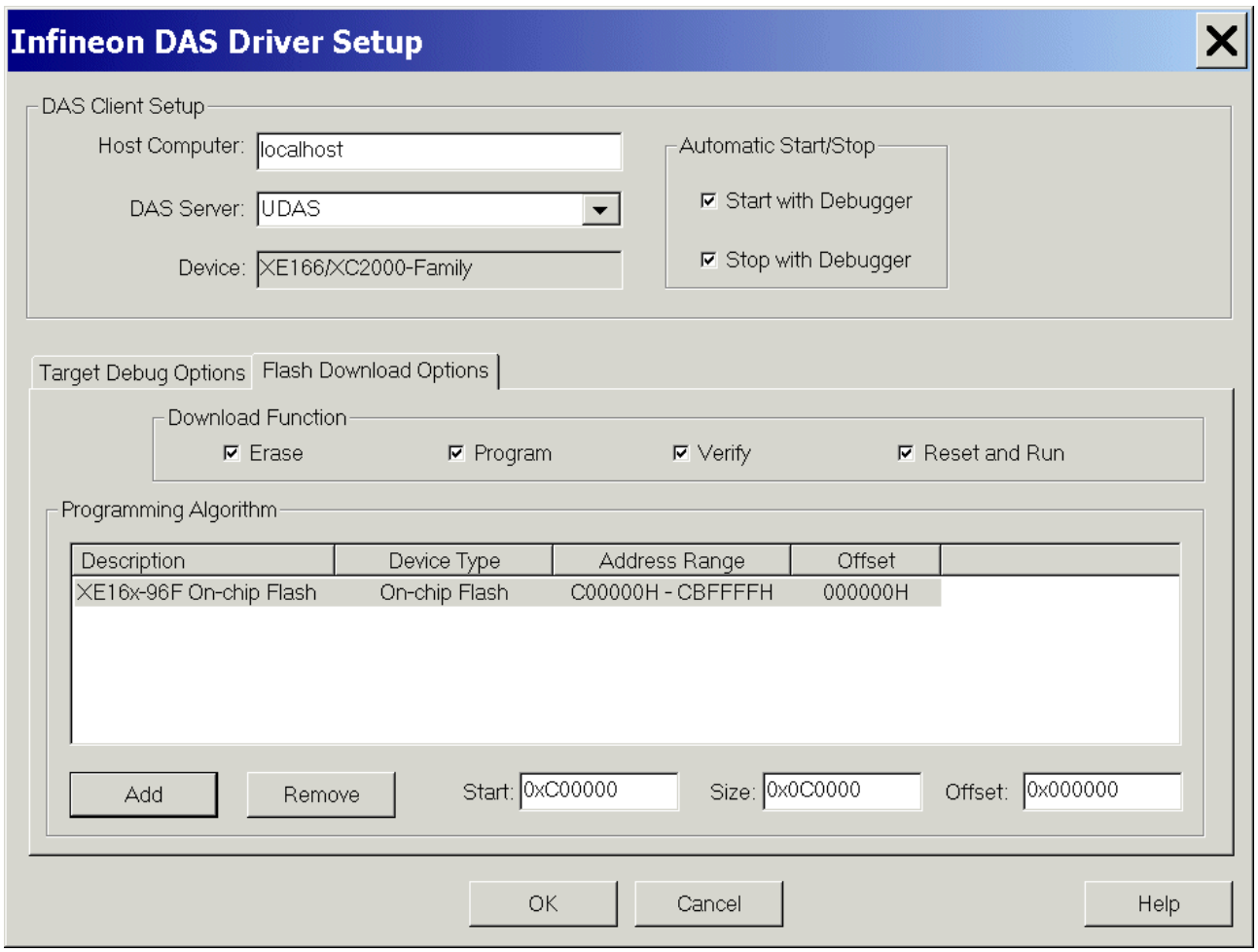
Click Add


Infineon DAS Driver Setup
DAS Client Setup
Host Computer:
localhost
DAS Server:
UDAS
Device:
XE166/XC2000-Family
Automatic Start/Stop
Start with Debugger
Stop with Debugger
Target Debug Options
Flash Download Options
Download Function
Erase
Program
Verify
Reset and Run
Programming Algorithm
Description
Device Type
Address Range
Offset
Add
Remove
Start:
Size:
Offset:
OK
Cancel
Help

Select: XE16x-96F On-chip Flash

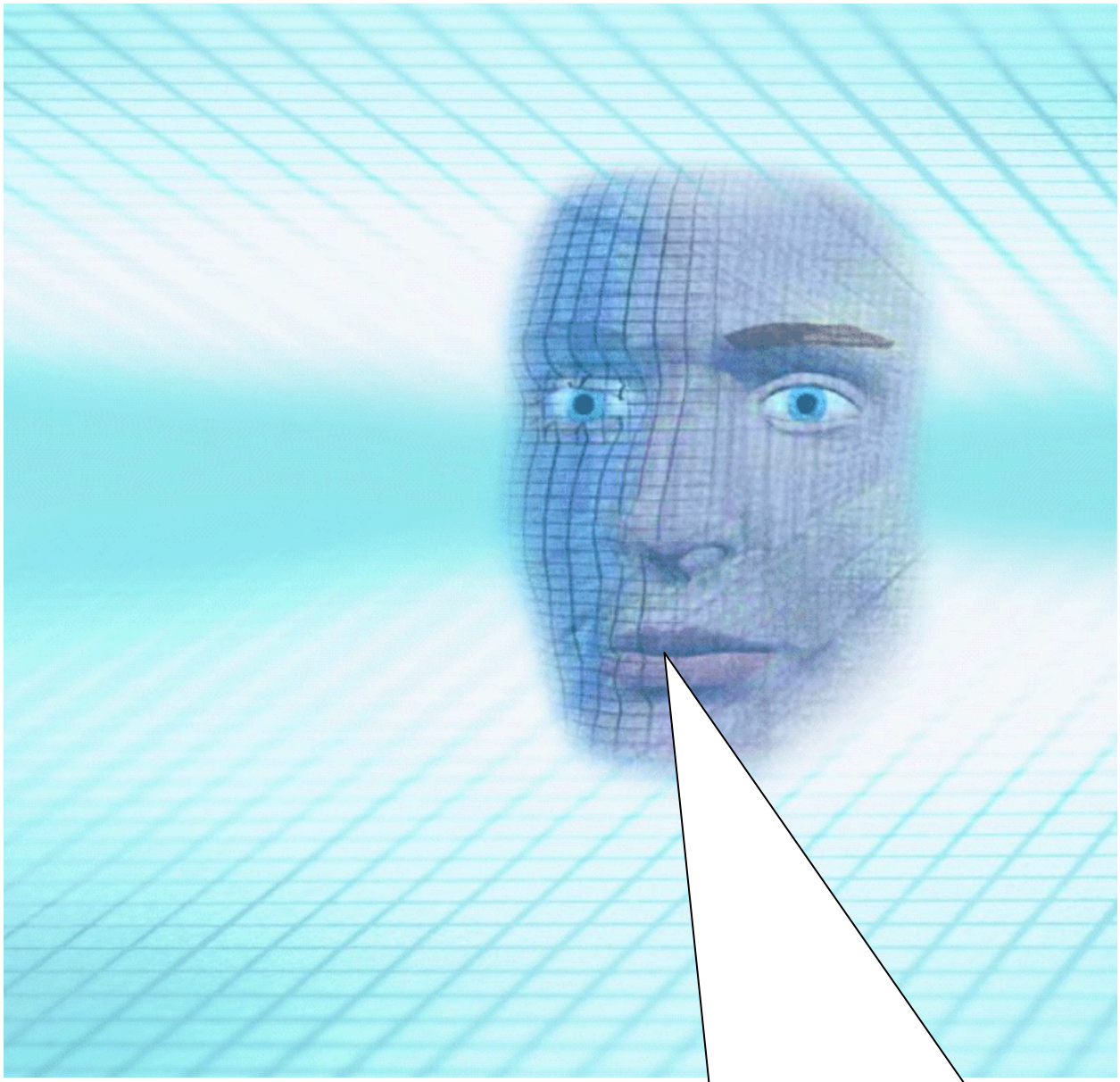


Click


Infineon DAS Driver Setup
DAS Client Setup
Host Computer:
localhost
DAS Server:
UDAS
Device:
XE166/XC2000-Family
Automatic Start/Stop
Start with Debugger
Stop with Debugger
Target Debug Options
Flash Download Options
Download Function
Erase
Program
Verify
Reset and Run
Programming Algorithm
Description
Device Type
Address Range
Offset
XE16x-96F On-chip Flash
On-chip Flash
C00000H - CBFFFFH
000000H
Add
Remove
Start:
0xC00000
Size:
0x0C0000
Offset:
0x000000
OK
Cancel
Help

OK
OK

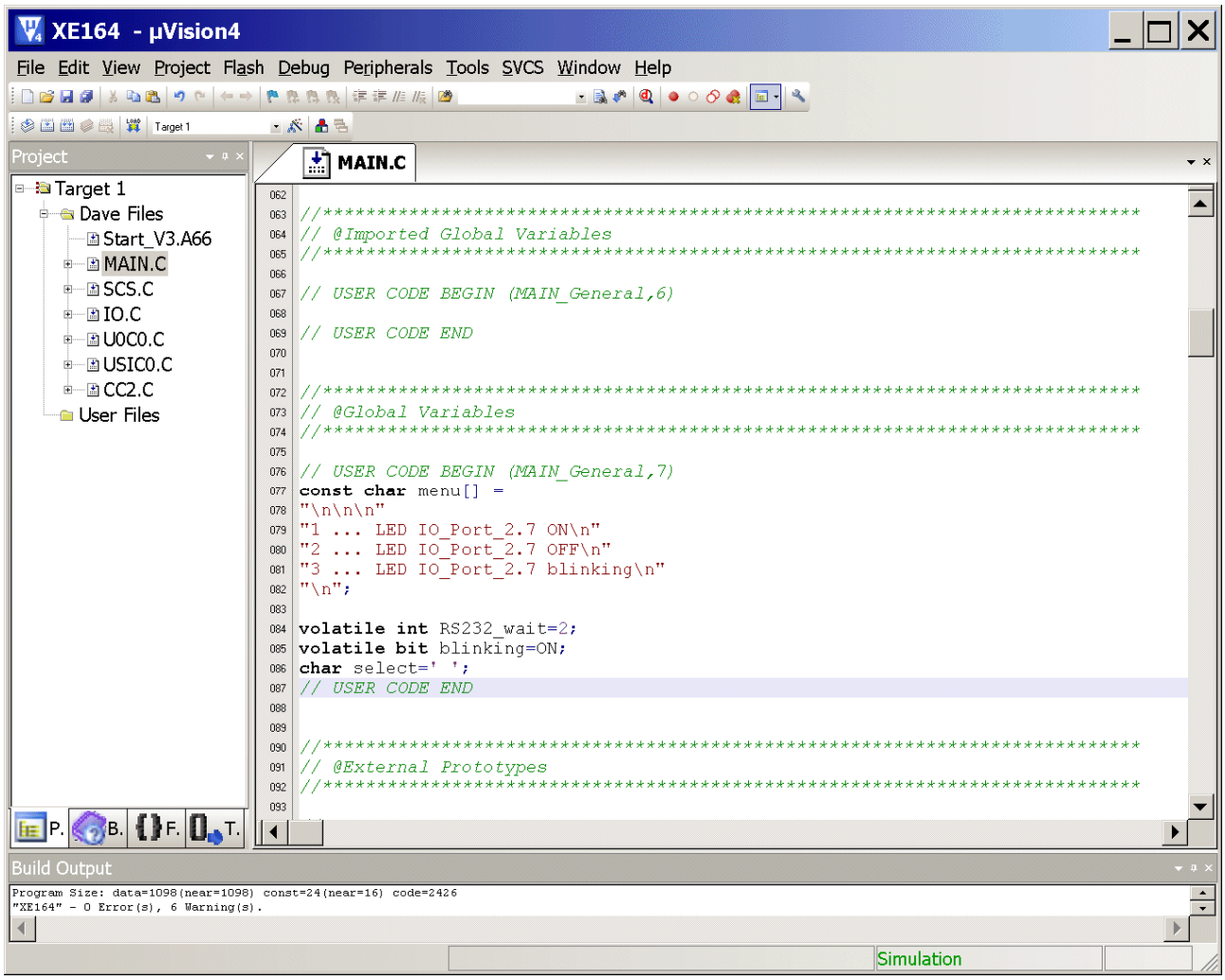
Insert your application specific program:



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!

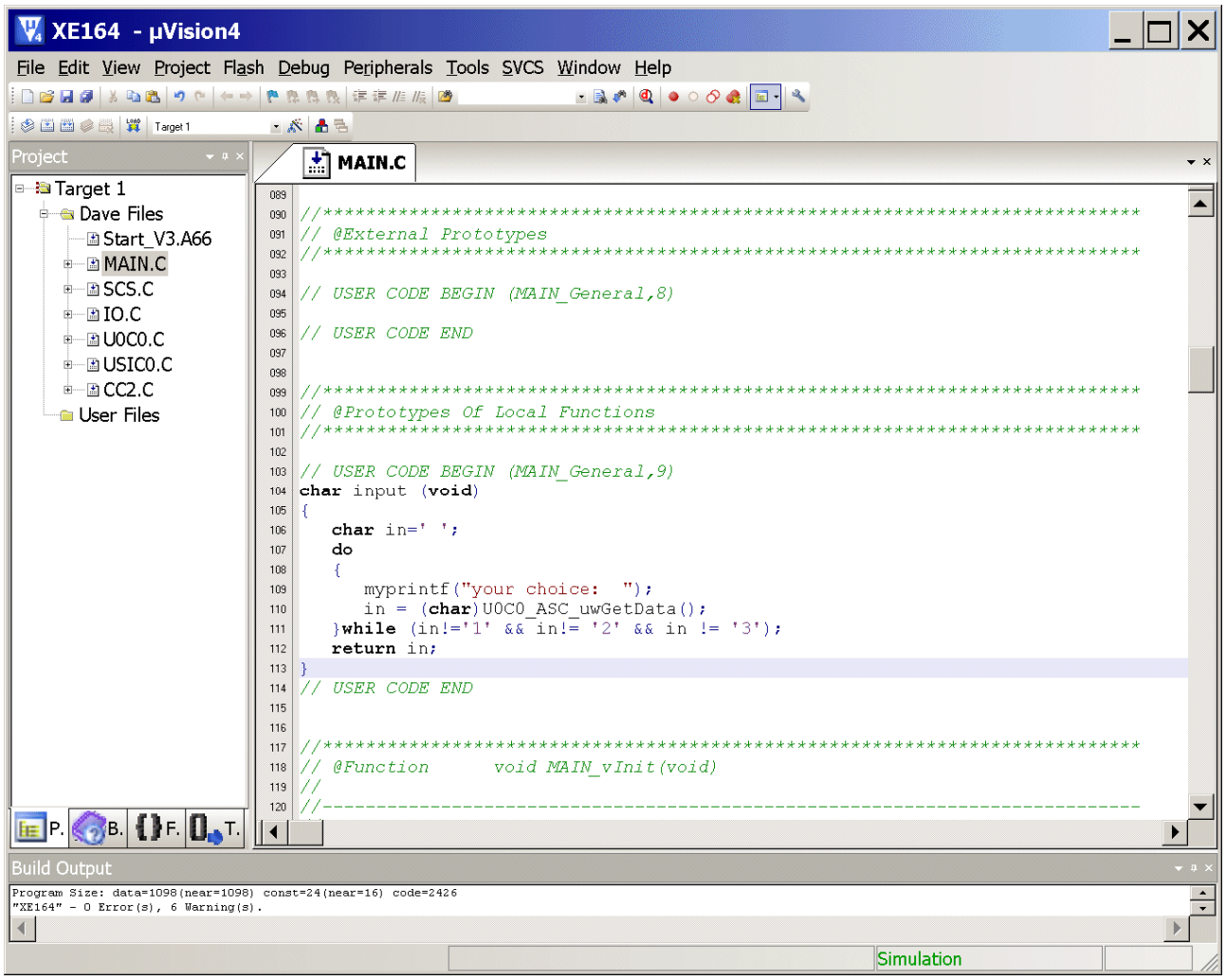
Double click MAIN.C and insert Global Variables:

```
const char menu[] =
"\n\n\n"
"1 ... LED IO_Port_2.7 ON\n"
"2 ... LED IO_Port_2.7 OFF\n"
"3 ... LED IO_Port_2.7 blinking\n"
"\n";

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



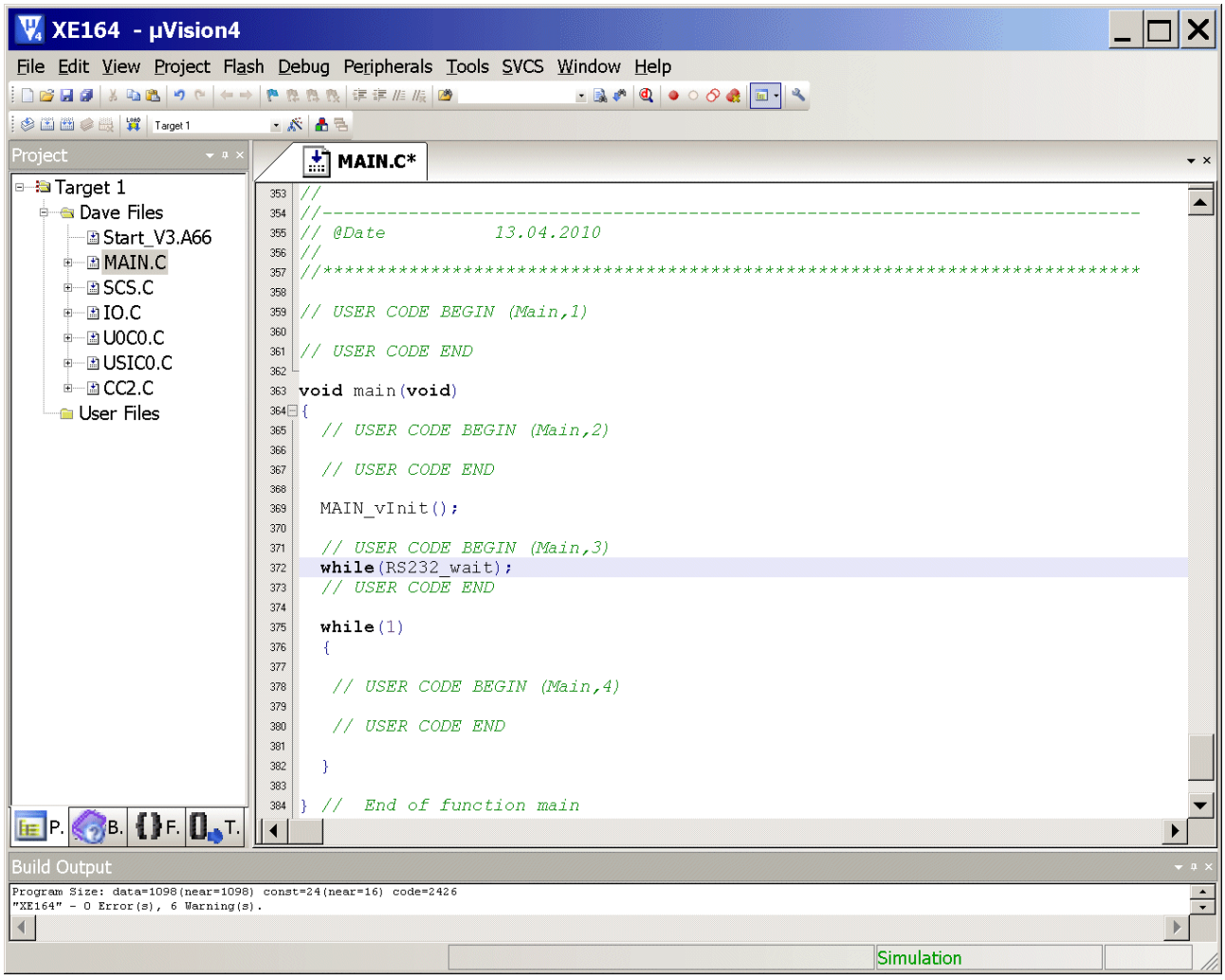
Double click MAIN.C and insert the function input():

```
char input (void)
{
  char in=' ';
  do
  {
    myprintf("your choice:  ");
    in = (char)U0C0_ASC_uwGetData();
  }while (in!='1' && in!= '2' && in != '3');
  return in;
}
```



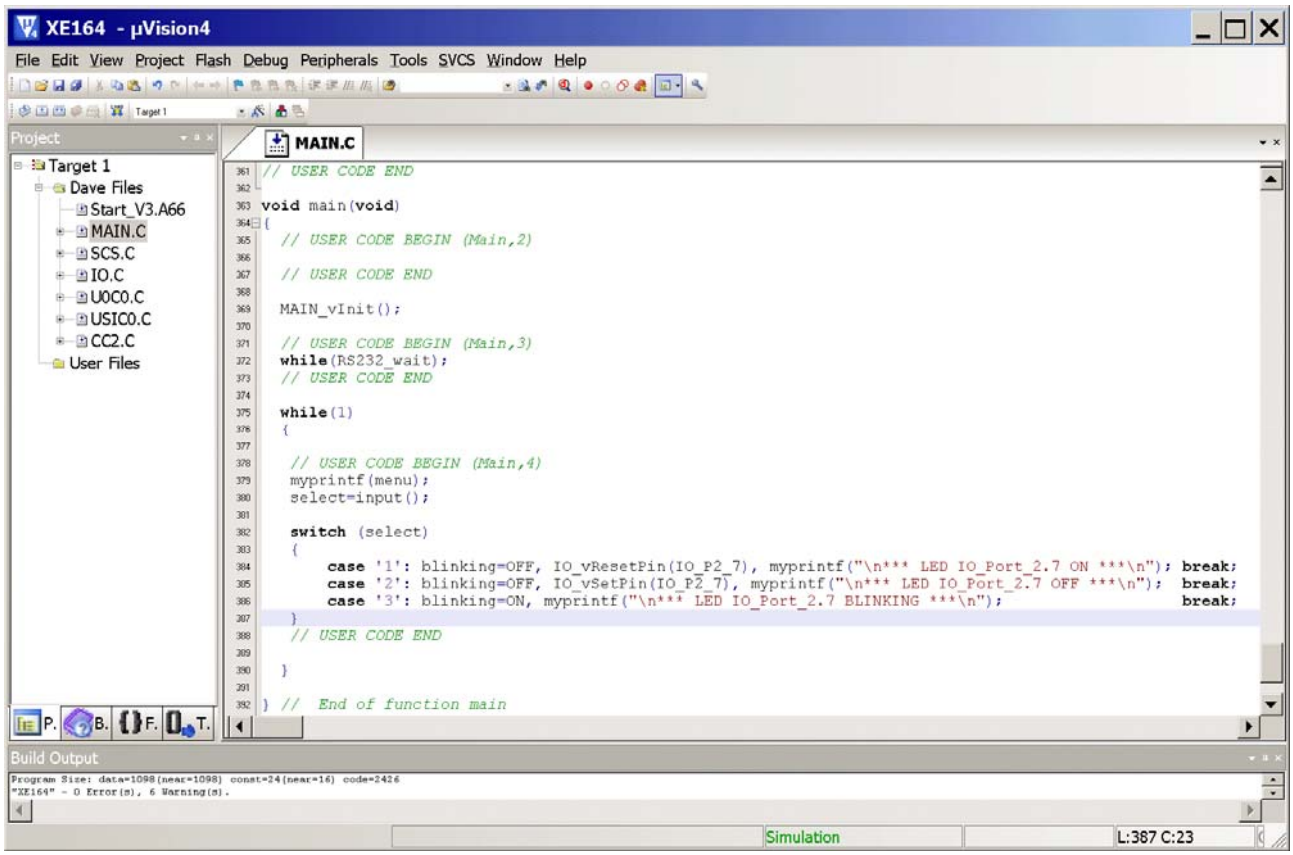
Double click MAIN.C and insert the following code in the main function:

```
while(RS232_wait);
```



Double click MAIN.C and insert the following code in the main function into the while(1) loop:

```
    myprintf(menu);
    select=input();

    switch (select)
    {
      case '1': blinking=OFF, IO_vResetPin(IO_P2_7), myprintf("\n*** LED IO_Port_2.7 ON
***\n");  break;
      case '2': blinking=OFF, IO_vSetPin(IO_P2_7), myprintf("\n*** LED IO_Port_2.7 OFF
***\n"); break;
      case '3': blinking=ON, myprintf("\n*** LED IO_Port_2.7 BLINKING ***\n");
break;
    }
```



Additional information: Port Output Modification Register (Source: User's Manual):

Pn_OMRL (n=6-11)
Port n Output Modification Register LowXSFR ($E9C0_H$+4*n) Reset Value: $XXXX_H$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC 7	PC 6	PC 5	PC 4	PC 3	PC 2	PC 1	PC 0	PS 7	PS 6	PS 5	PS 4	PS 3	PS 2	PS 1	PS 0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
PSx (x = 0-7)	x	w	**Port Set Bit x** Setting this bit sets or toggles the corresponding bit in the port output register Pn_OUT (see **Table 7-4**). On a read access, this bit returns 0.
PCx (x = 0-7)	x + 8	w	**Port Clear Bit x** Setting this bit clears or toggles the corresponding bit in the port output register Pn_OUT. (see **Table 7-4**). On a read access, this bit returns 0.

Function of the PCx and PSx bit fields

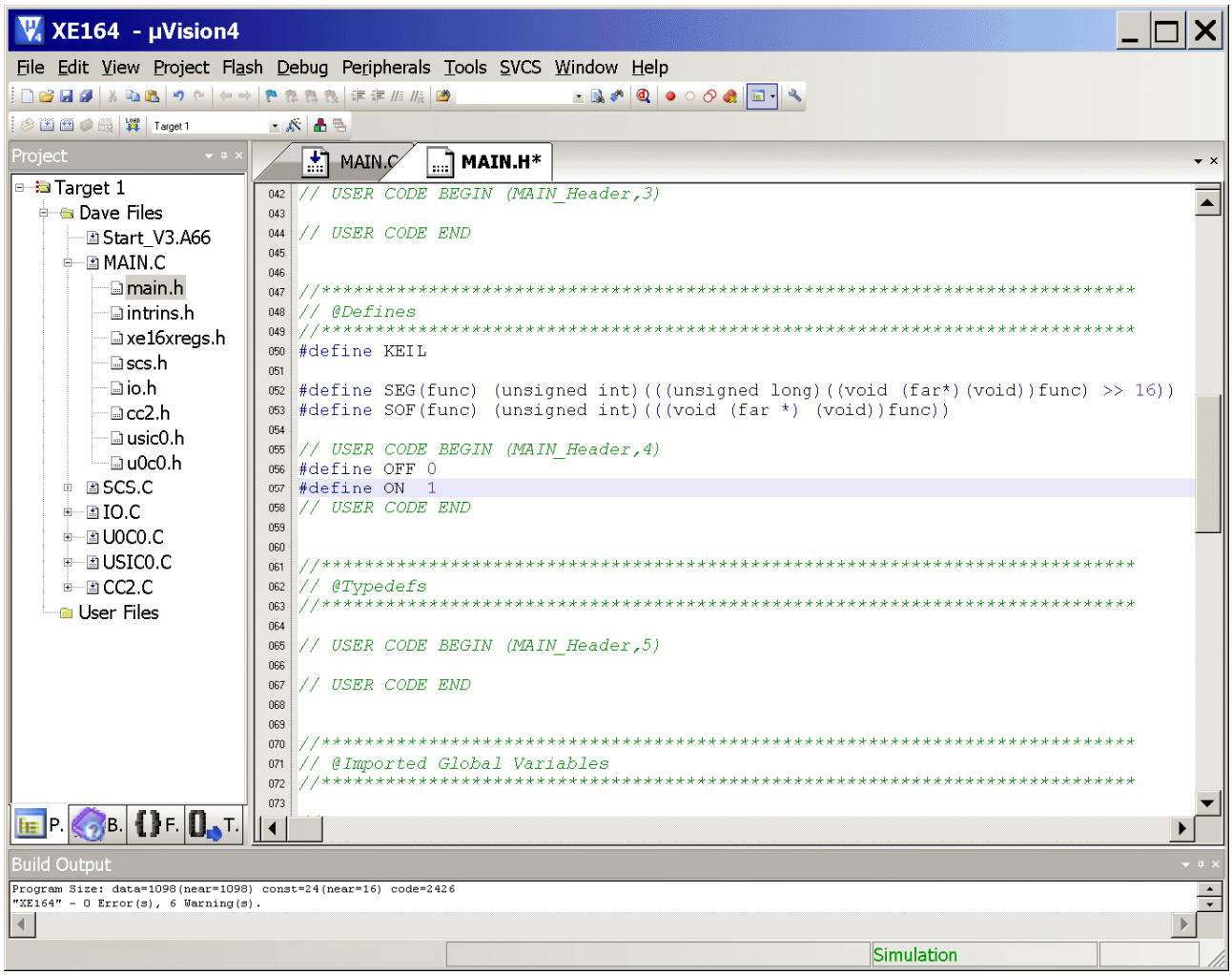
Table 7-4 Function of the Bits PCx and PSx

PCx	PSx	Function
0 or no write access	0 or no write access	Bit Pn_OUT.Px is not changed.
0 or no write access	1	Bit Pn_OUT.Px is set.
1	0 or no write access	Bit Pn_OUT.Px is cleared.
1	1	Bit Pn_OUT.Px is toggled.

Note: If a bit position is not written (one out of two bytes not targeted by a byte write), the corresponding value is considered as 0. Toggling a bit requires one 16-bit write.

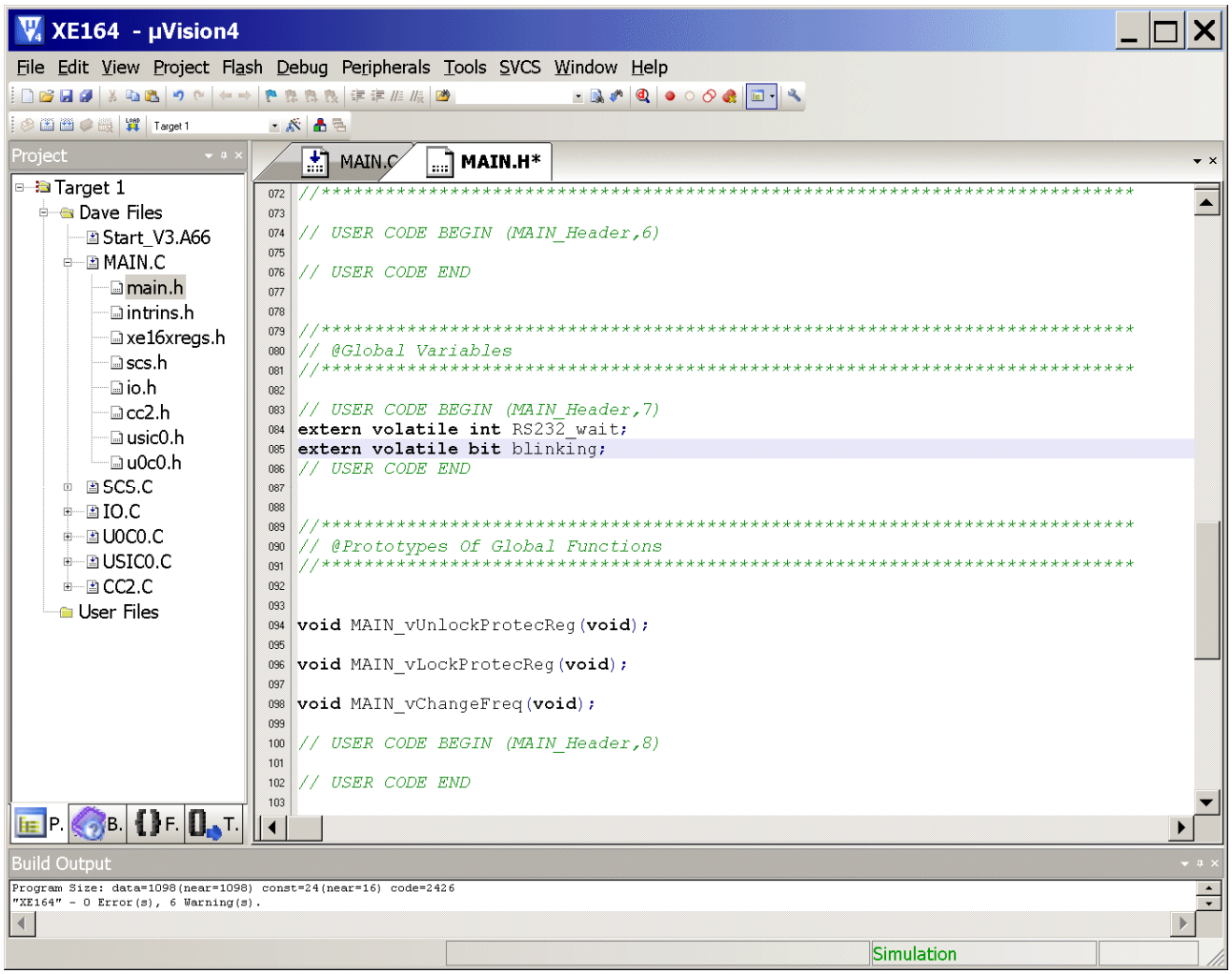
Double click Main.h and insert the following Defines:

```
#define OFF 0
#define ON  1
```



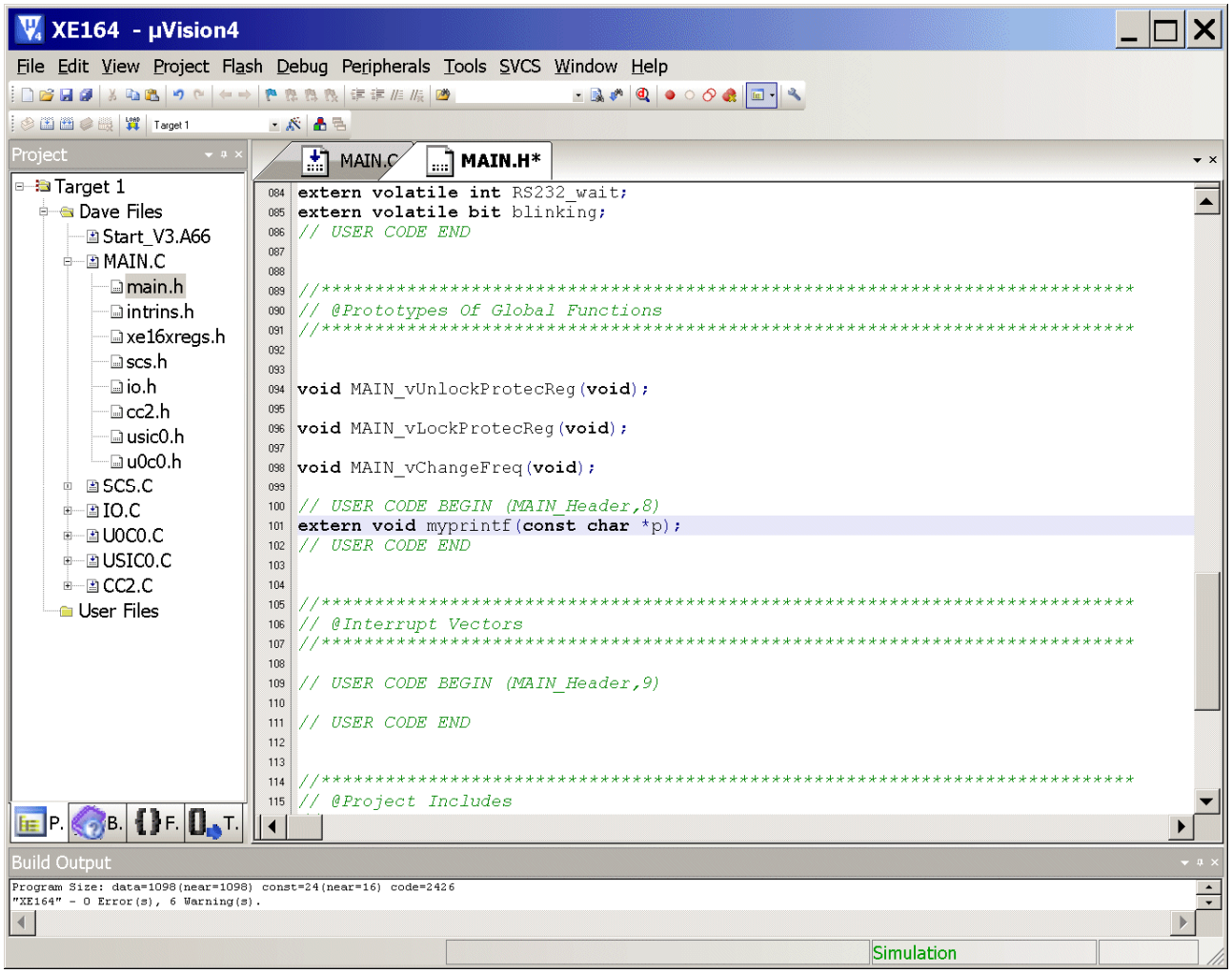
Double click Main.h and insert extern declarations “Global Variables”:

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



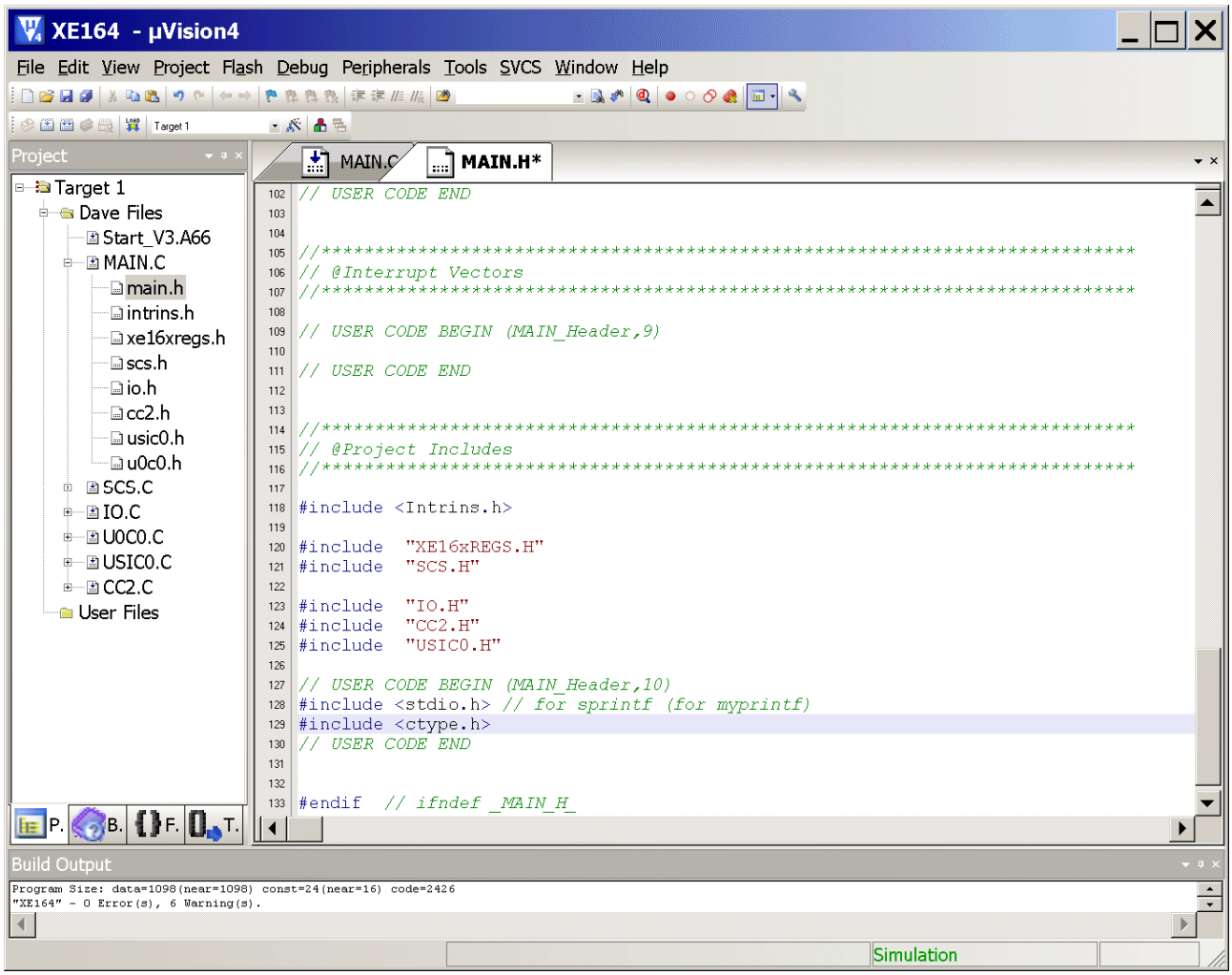
Double click Main.h and insert extern declarations “Global Functions”:

```
extern void myprintf(const char *p);
```

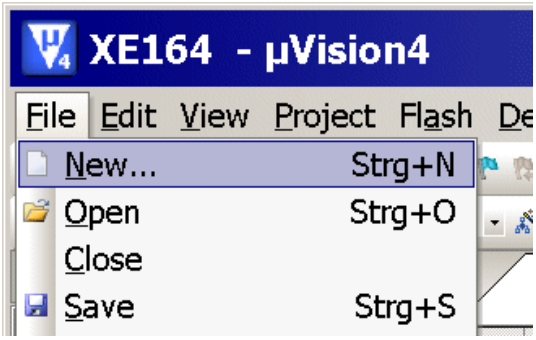


Double click Main.h and insert include files:

```
#include <stdio.h> // for sprintf (for myprintf)
#include <ctype.h>
```



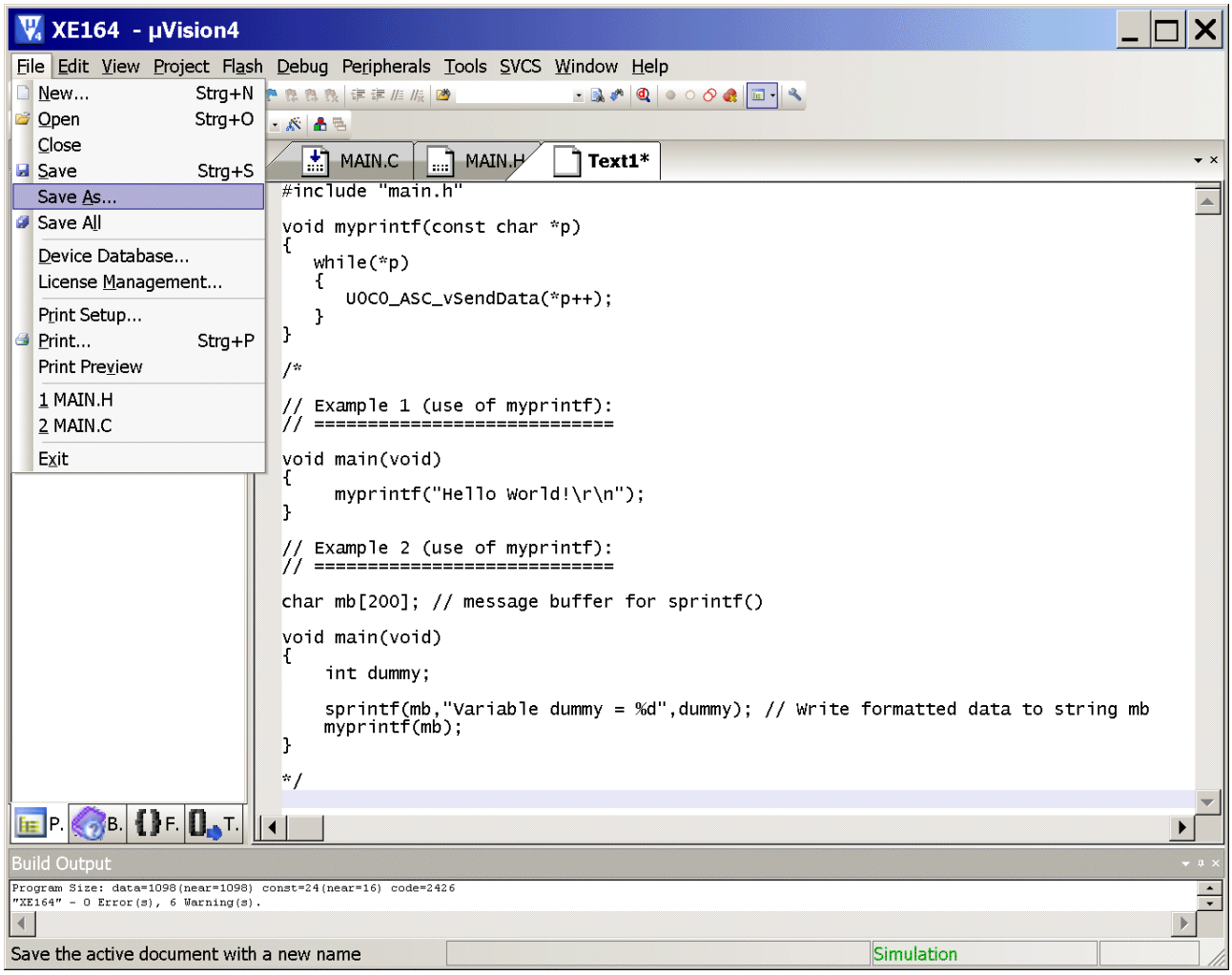
File – New



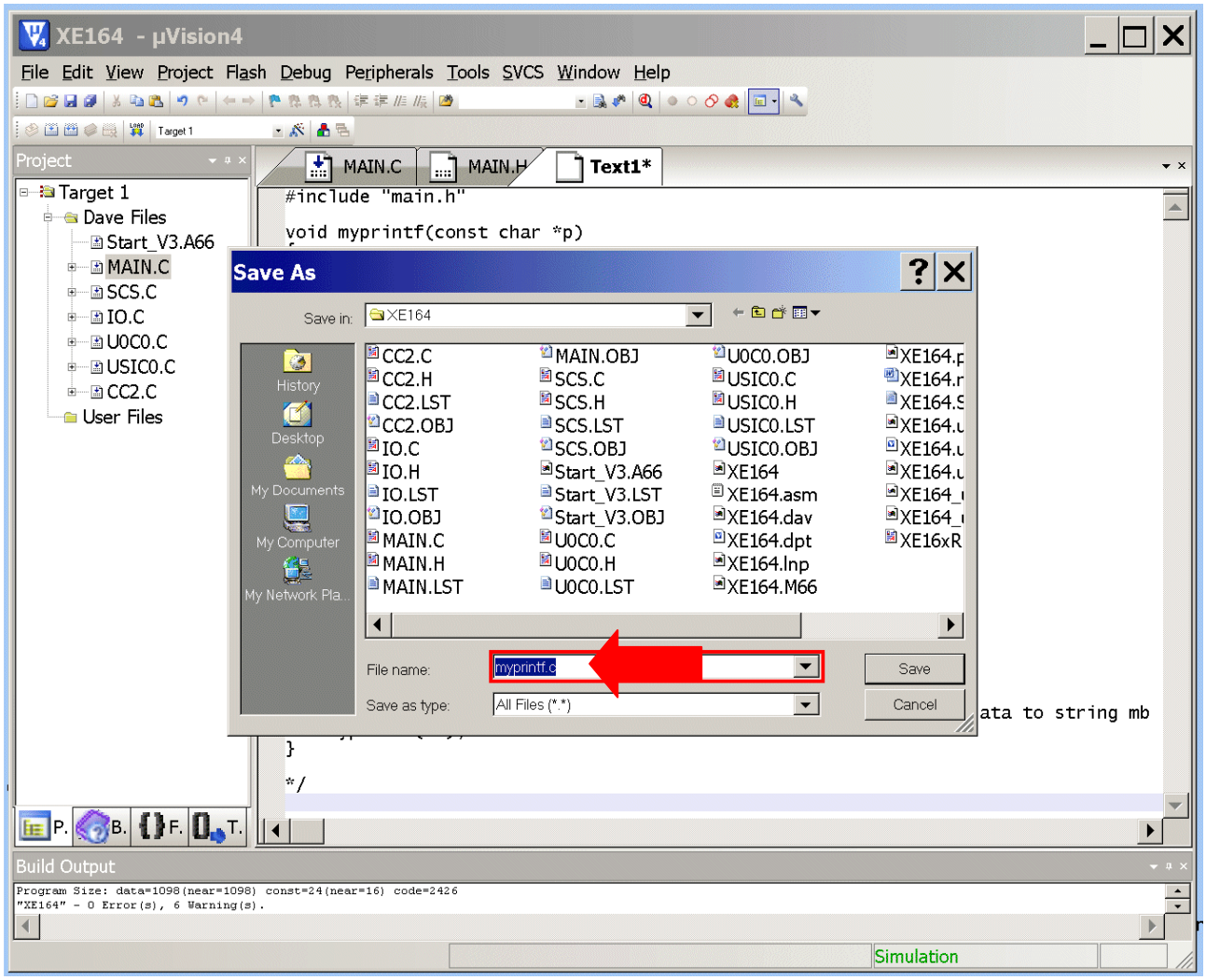
Insert:

```
#include "main.h"

void myprintf(const char *p)
{
  while(*p)
  {
    U0C0_ASC_vSendData(*p++);
  }
}

/*

// Example 1 (use of myprintf):
// ============================

void main(void)
{
   myprintf("Hello World!\r\n");
}

// Example 2 (use of myprintf):
// ============================

char mb[200]; // message buffer for sprintf()

void main(void)
{
  int dummy;

  sprintf(mb,"Variable dummy = %d",dummy); // Write formatted data to string mb
  myprintf(mb);
}

*/
```

File – Save As…

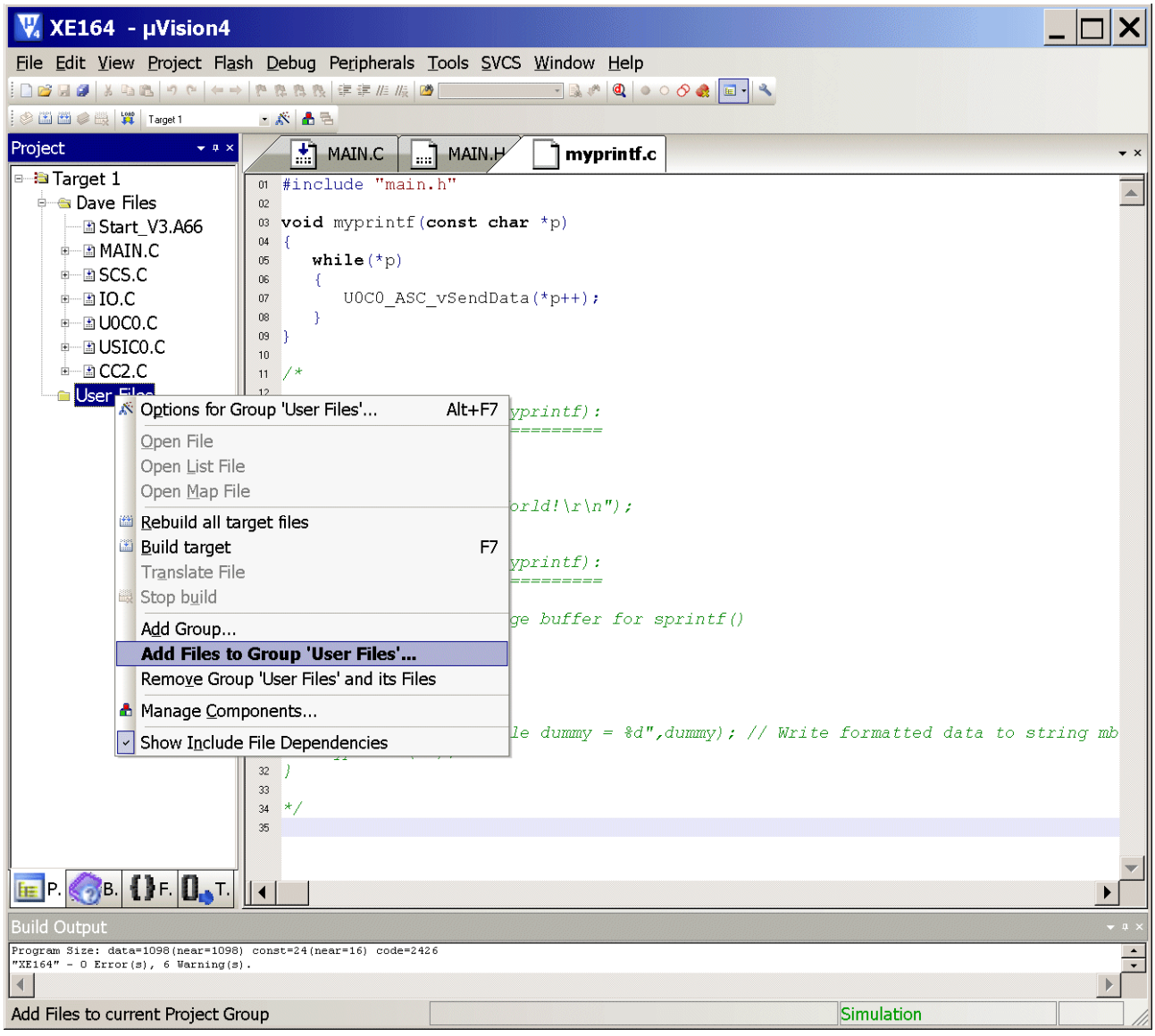

XE164 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
New... Strg+N
Open Strg+O
Close
Save Strg+S
Save As...
Save All
Device Database...
License Management...
Print Setup...
Print... Strg+P
Print Preview
1 MAIN.H
2 MAIN.C
Exit
MAIN.C
MAIN.H
Text1*
#include "main.h"
void myprintf(const char *p)
{
while(*p)
{
U0C0_ASC_vSendData(*p++);
}
}
/*
// Example 1 (use of myprintf):
// ============================
void main(void)
{
myprintf("Hello World!\r\n");
}
// Example 2 (use of myprintf):
// ============================
char mb[200]; // message buffer for sprintf()
void main(void)
{
int dummy;
sprintf(mb,"Variable dummy = %d",dummy); // Write formatted data to string mb
myprintf(mb);
}
*/
P. B. F. T.
Build Output
Program Size: data=1098(near=1098) const=24(near=16) code=2426
"XE164" - 0 Error(s), 6 Warning(s).
Save the active document with a new name
Simulation

Insert: myprintf.c

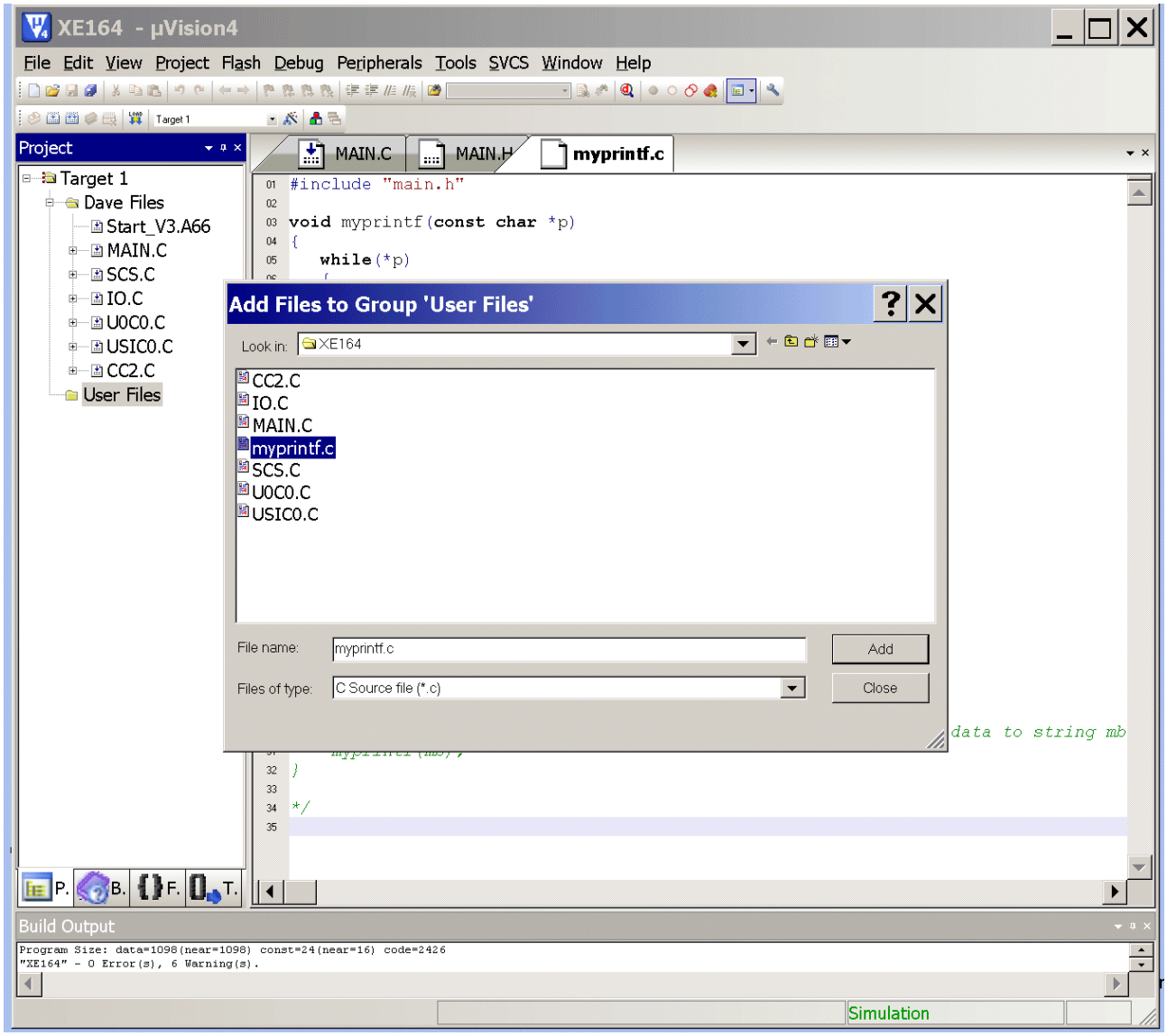


Save

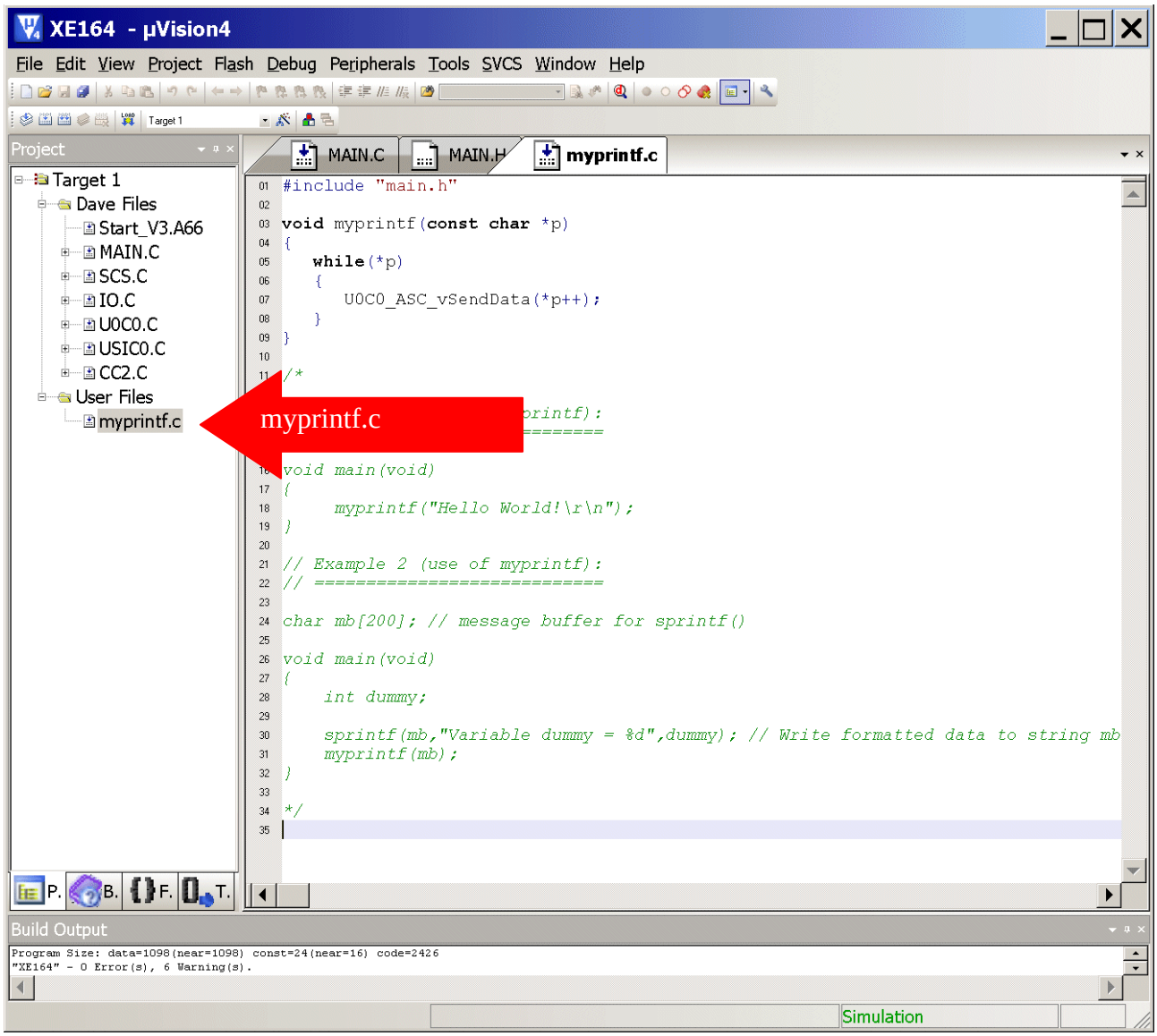
Mouse position: Project Window, User Files: click right mouse button
click Add Files to Group ‘User Files’



Click/select myprintf.c

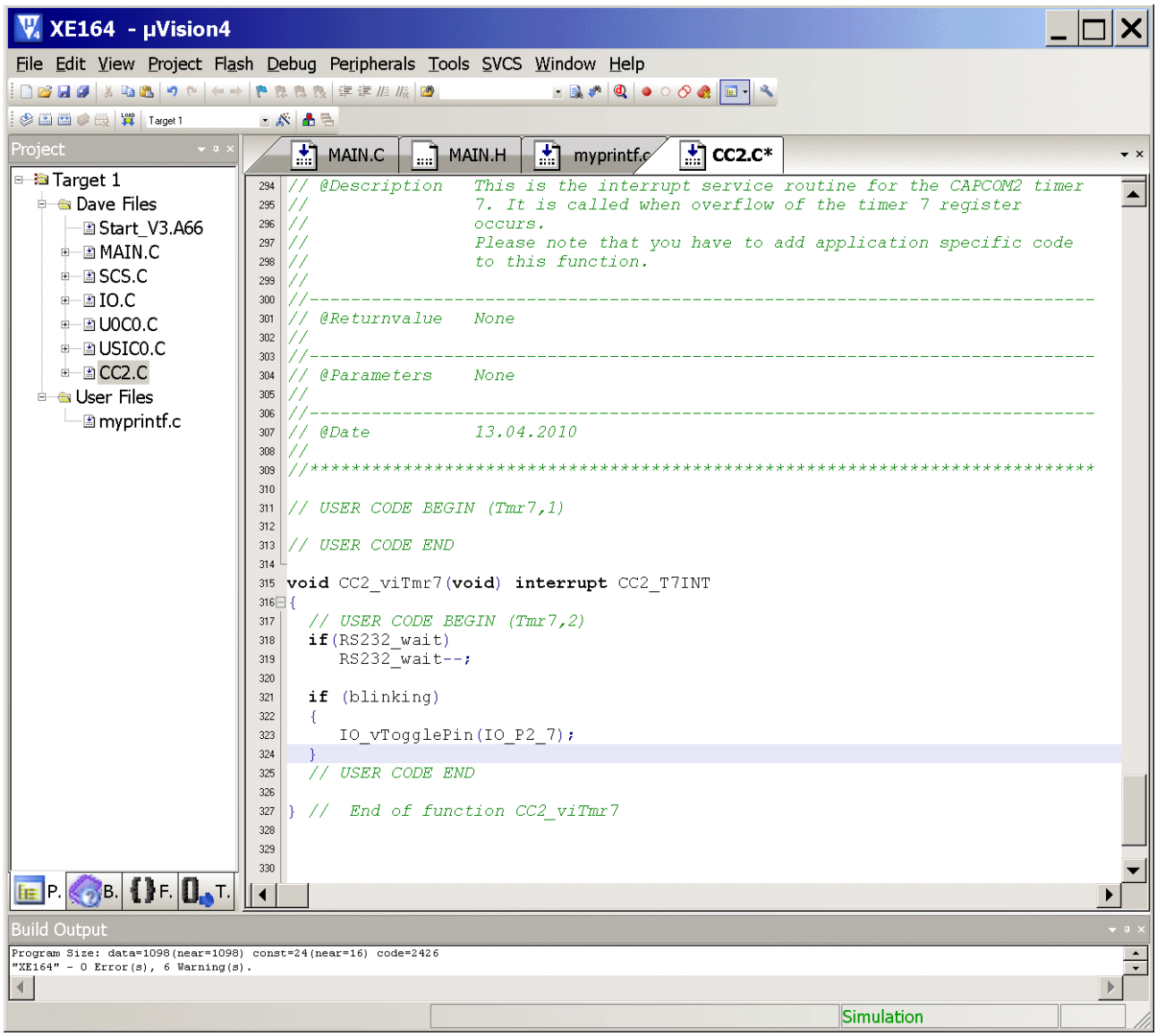


Add
Close

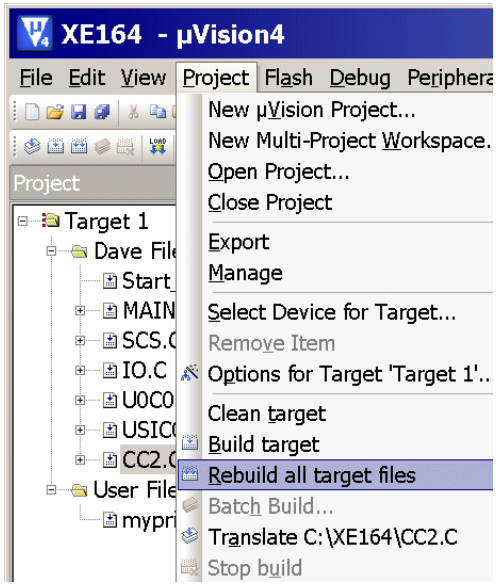
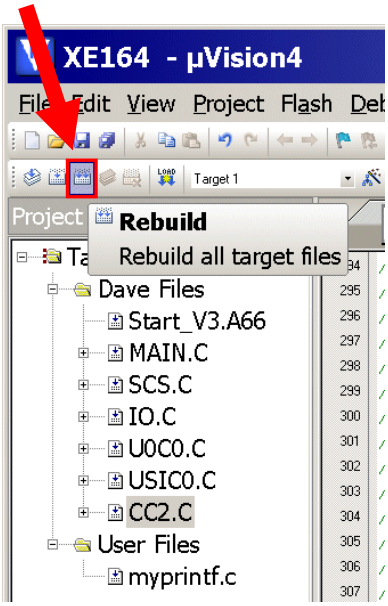

XE164 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Target 1
Dave Files
Start_V3.A66
MAIN.C
SCS.C
IO.C
U0C0.C
USIC0.C
CC2.C
User Files
myprintf.c
MAIN.C
MAIN.H
myprintf.c
#include "main.h"
void myprintf(const char *p)
{
while(*p)
{
U0C0_ASC_vSendData(*p++);
}
}
/*
void main(void)
{
myprintf("Hello World!\r\n");
}
// Example 2 (use of myprintf):
// ============================
char mb[200]; // message buffer for sprintf()
void main(void)
{
int dummy;
sprintf(mb,"Variable dummy = %d",dummy); // Write formatted data to string mb
myprintf(mb);
}
*/
Build Output
Program Size: data=1098(near=1098) const=24(near=16) code=2426
"XE164" - 0 Error(s), 6 Warning(s).
Simulation

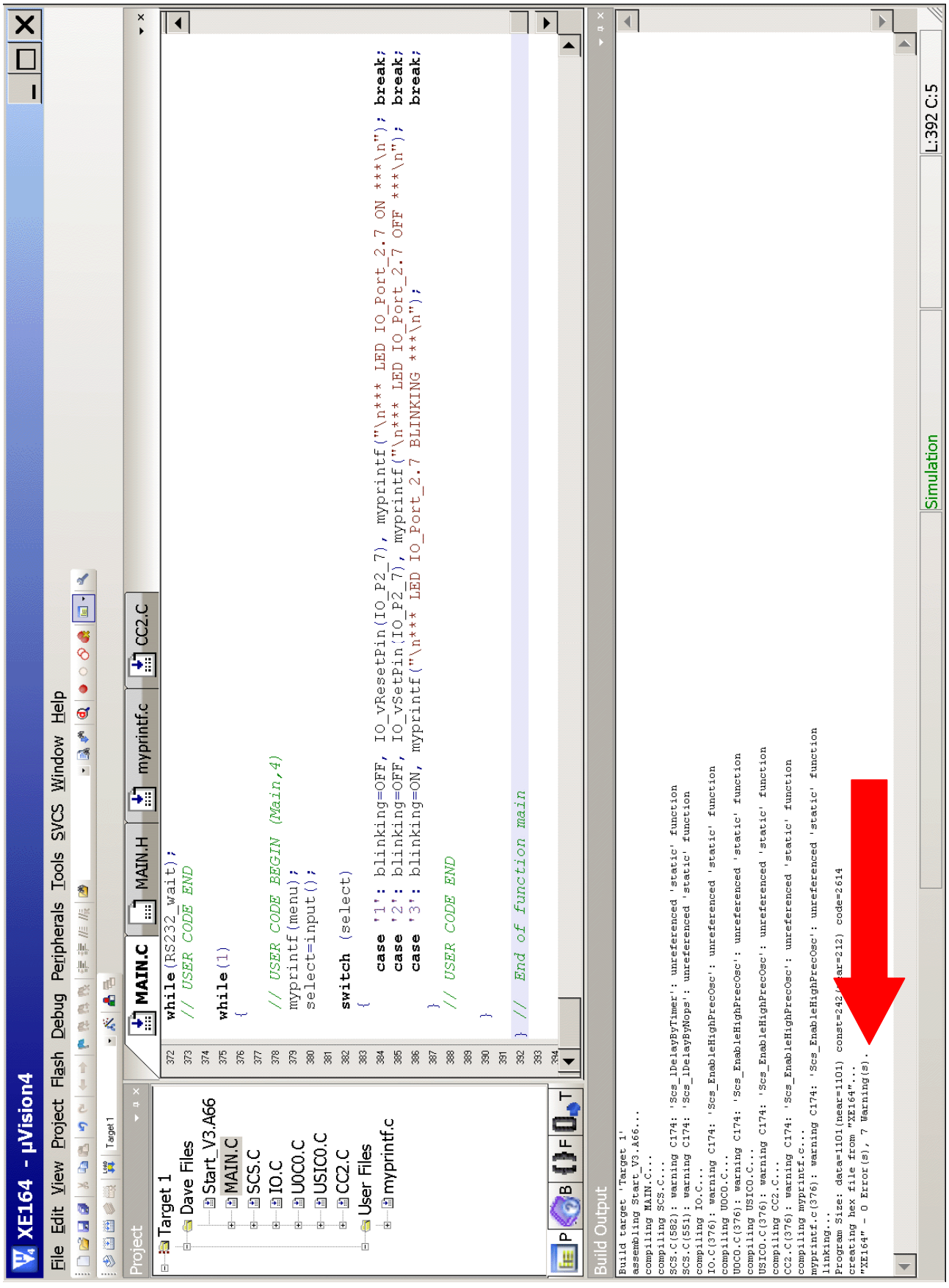
Double click CC2.C insert Code (CAPCOM 2 Timer 7 Interrupt Service Routine):

```
if(RS232_wait)
  RS232_wait--;

if (blinking)
{
  IO_vTogglePin(IO_P2_7);
}
```



Generate your application program:

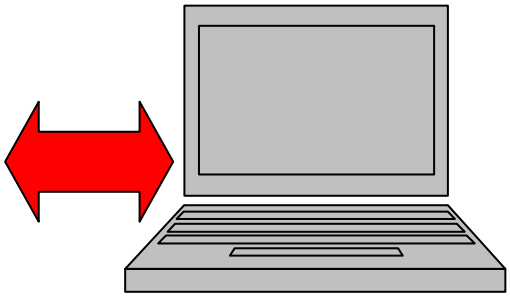
Project – Rebuild all target files	or	click
		

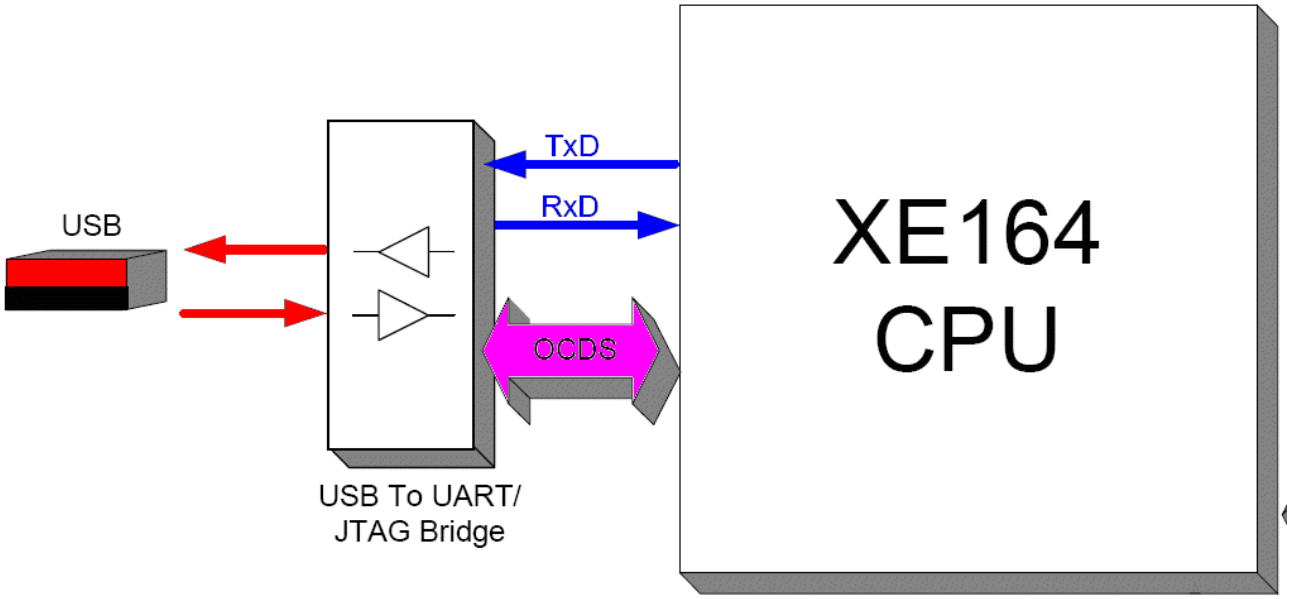

XE164 - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Target 1
Dave Files
Start_V3.A66
MAIN.C
SCS.C
IO.C
U0C0.C
USIC0.C
CC2.C
User Files
myprintf.c
MAIN.C
MAIN.H
myprintf.c
CC2.C
while(RS232_wait);
// USER CODE END
while(1)
{
// USER CODE BEGIN (Main,4)
myprintf(menu);
select=input();
switch (select)
{
case '1': blinking=OFF, IO_vResetPin(IO_P2_7), myprintf("\n*** LED IO_Port_2.7 ON ***\n"); break;
case '2': blinking=OFF, IO_vSetPin(IO_P2_7), myprintf("\n*** LED IO_Port_2.7 OFF ***\n"); break;
case '3': blinking=ON, myprintf("\n*** LED IO_Port_2.7 BLINKING ***\n"); break;
}
// USER CODE END
}
// End of function main
Build Output
Build target 'Target 1'
assembling Start_V3.A66...
compiling MAIN.C...
compiling SCS.C...
SCS.C(582): warning C174: 'Scs_lDelayByTimer': unreferenced 'static' function
SCS.C(551): warning C174: 'Scs_lDelayByNops': unreferenced 'static' function
compiling IO.C...
IO.C(376): warning C174: 'Scs_EnableHighPrecOsc': unreferenced 'static' function
compiling U0C0.C...
U0C0.C(376): warning C174: 'Scs_EnableHighPrecOsc': unreferenced 'static' function
compiling USIC0.C...
USIC0.C(376): warning C174: 'Scs_EnableHighPrecOsc': unreferenced 'static' function
compiling CC2.C...
CC2.C(376): warning C174: 'Scs_EnableHighPrecOsc': unreferenced 'static' function
compiling myprintf.c...
myprintf.c(376): warning C174: 'Scs_EnableHighPrecOsc': unreferenced 'static' function
linking...
Program Size: data=1101(near=1101) const=242(far=212) code=2614
creating hex file from "XE164"...
"XE164" - 0 Error(s), 7 Warning(s).
Simulation
L:392 C:5

Note:
Programming is now complete.
Unfortunately it is not possible to test your program with the Keil Simulator because this feature is currently not supported.
Therefore we are going to load (On Chip Flash Programming) and run your program on the UConnect-CAN XE164 in the next chapter.

5.) Running your first programming example:

Make sure that the UConnect-CAN XE164 is still connected to the host computer:

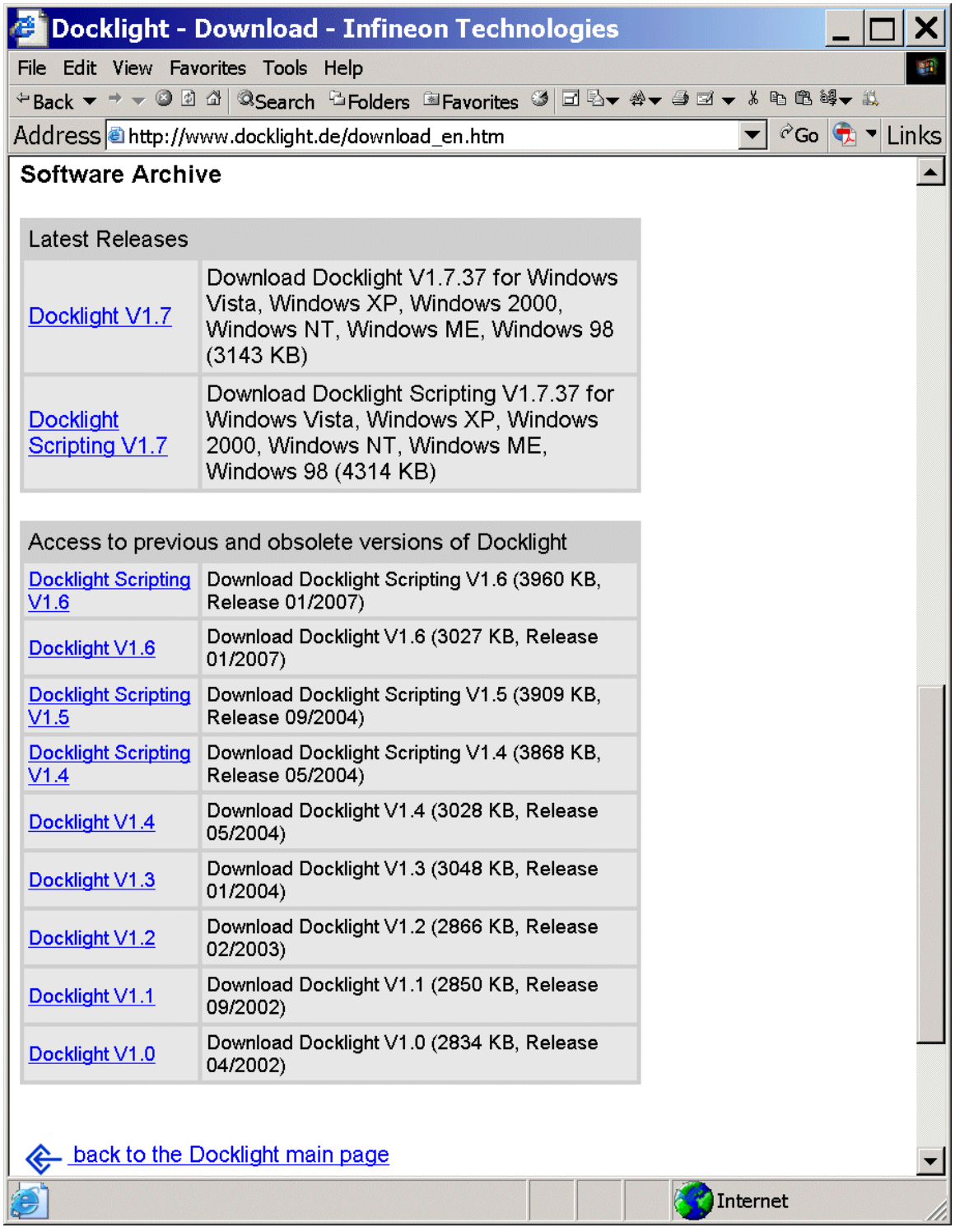




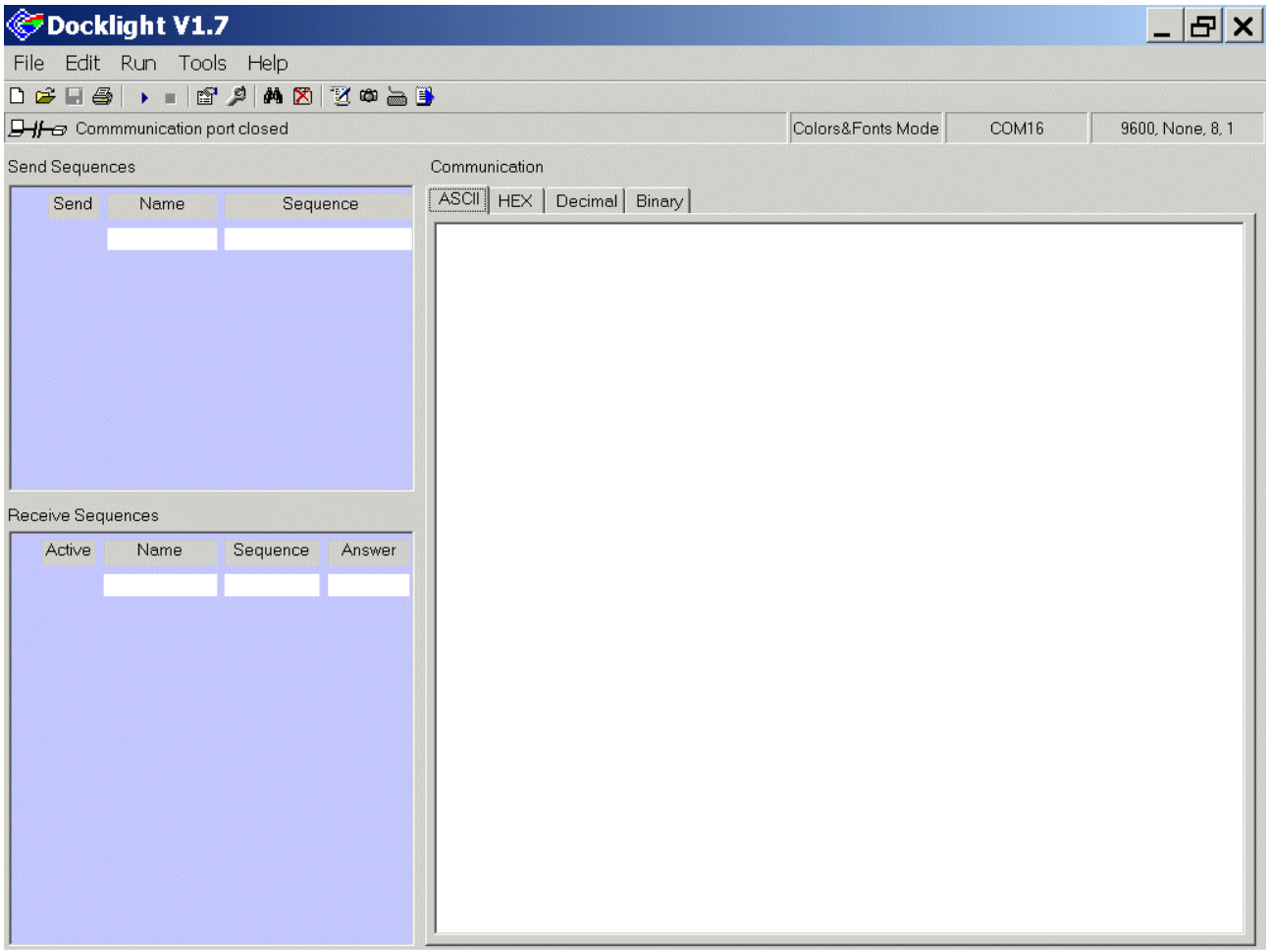
USB Connection:

.) used for: UART communication (the USIC0_CH0/UART/RS232/serial interface is available via USB as a virtual COM port of the second USB channel of the FTDI FT2232 Dual USB to UART/JTAG interface).
.) used for: On-Chip-Flash-Programming and Debugging (first USB channel of the FTDI FT2232 Dual USB to UART/JTAG interface).
.) the USB connection works also as the power supply.

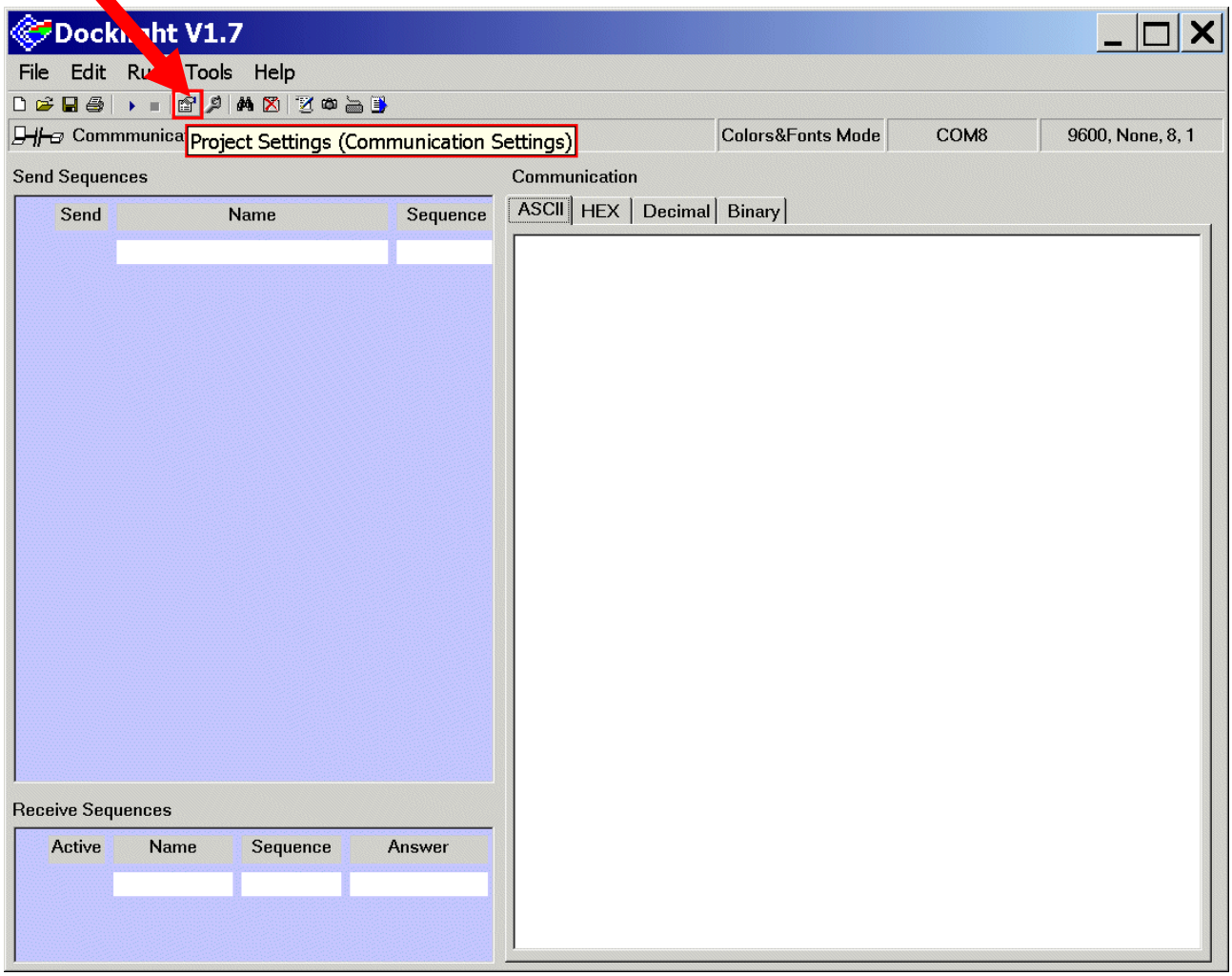
Note:
Now we need a terminal program which is able to handle a virtual COM port (COM12)!
As an example of “any terminal program” we are going to use Docklight.
Docklight can be downloaded @ http://www.docklight.de :



Now, start Docklight:



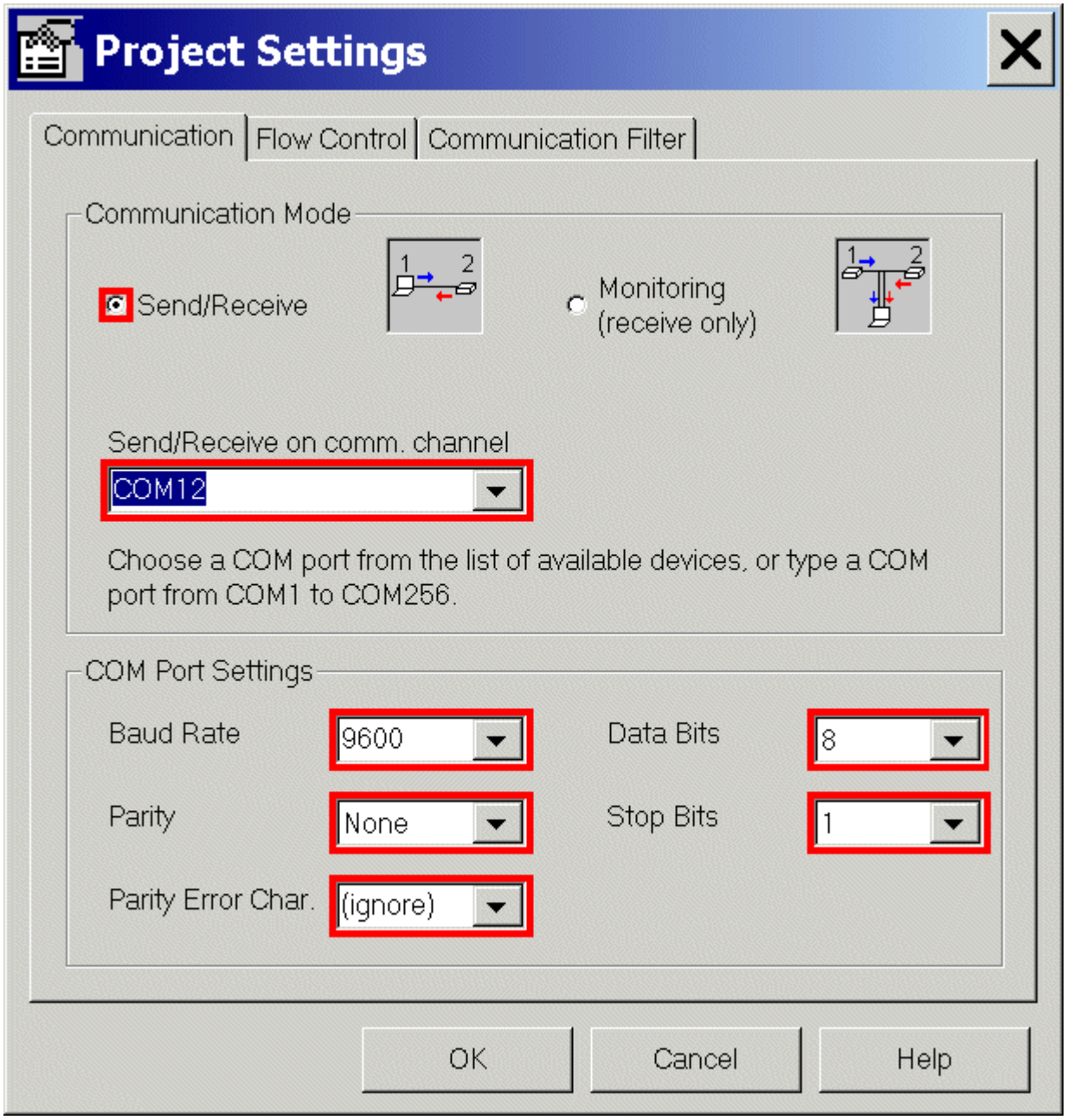
Click: Project Settings



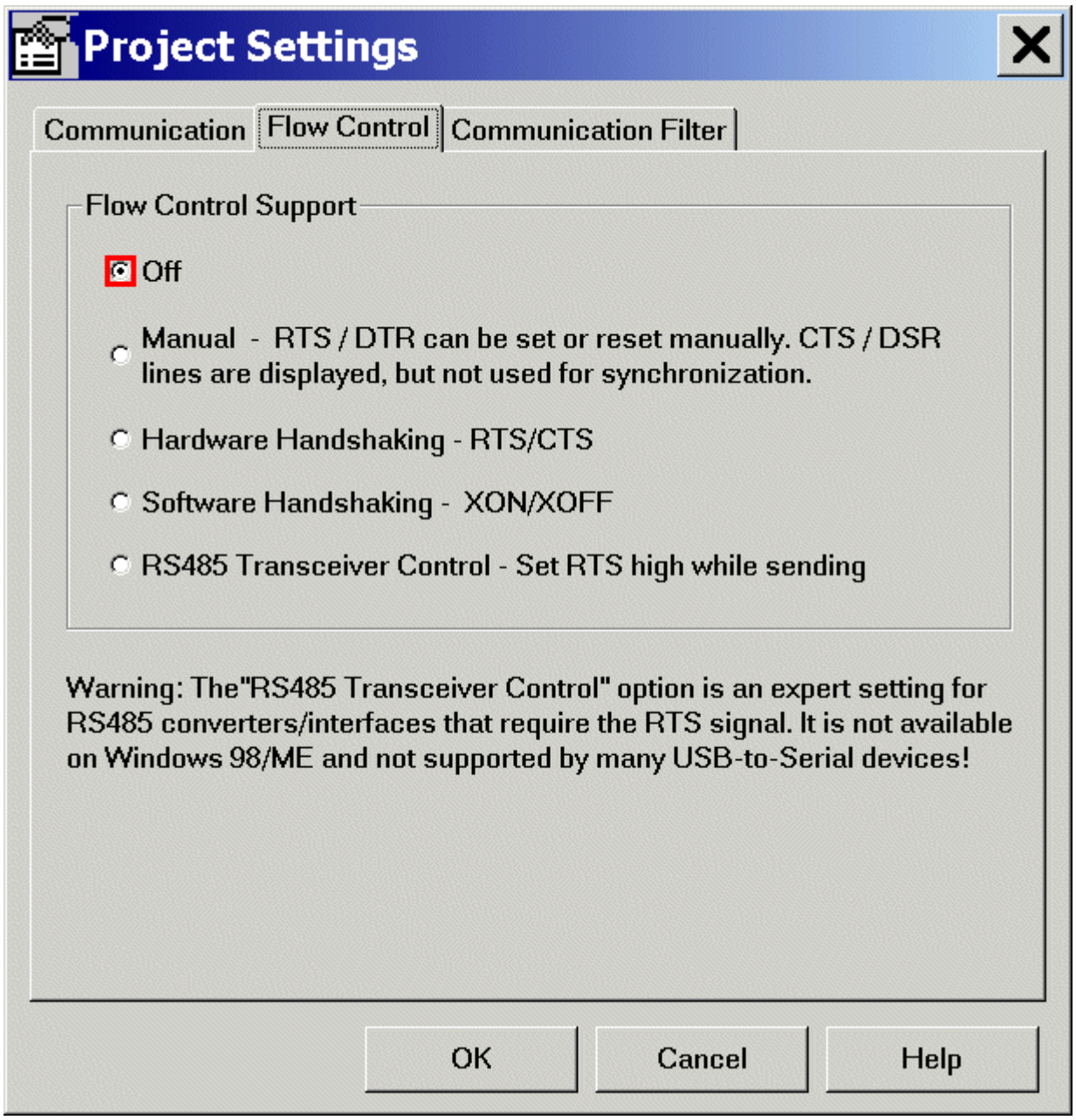
Project Settings:
Communication: Communication Mode: click/check ⊙ Send/Receive

Project Settings:
Communication: Communication Mode: Send/Receive on comm. channel: select COM12

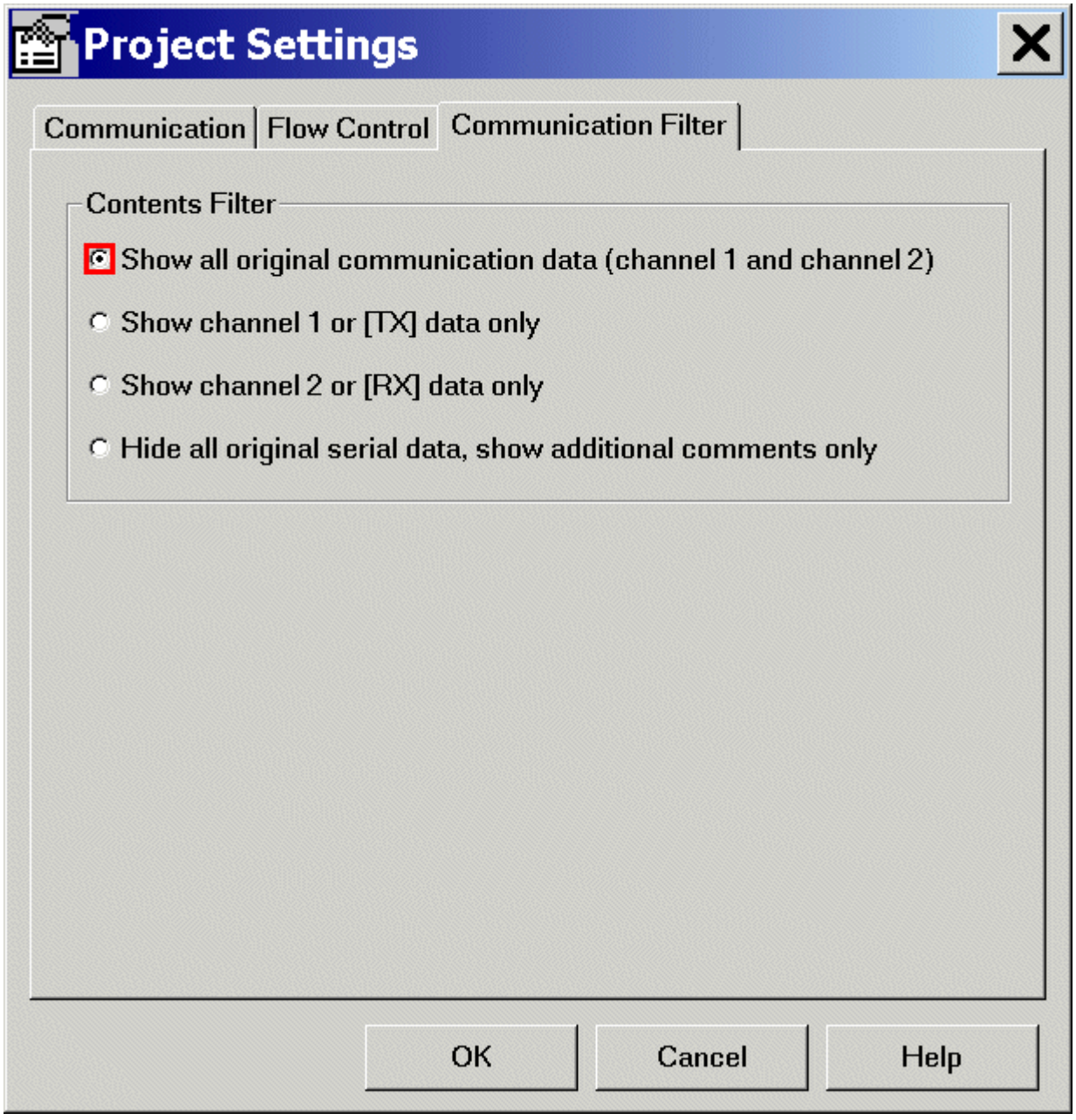
Project Settings: Communication: COM Port Settings: Baud Rate: select 9600
Project Settings: Communication: COM Port Settings: Parity: select None
Project Settings: Communication: COM Port Settings: Parity Error Char.: select (ignore)
Project Settings: Communication: COM Port Settings: Data Bits: select 8
Project Settings: Communication: COM Port Settings: Stop Bits: select 1



Project Settings: Flow Control: Flow Control Support: click ⊙ Off

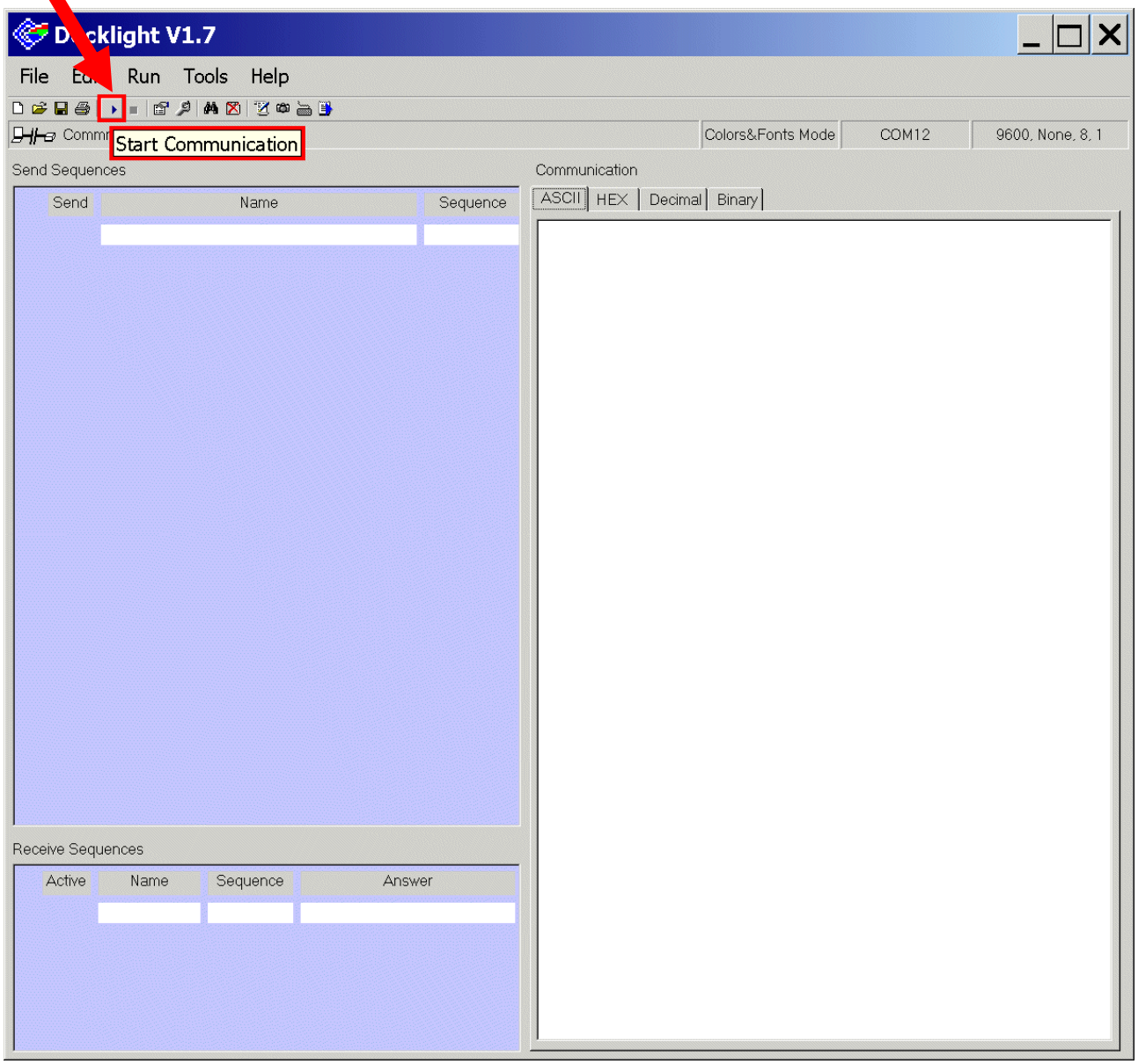


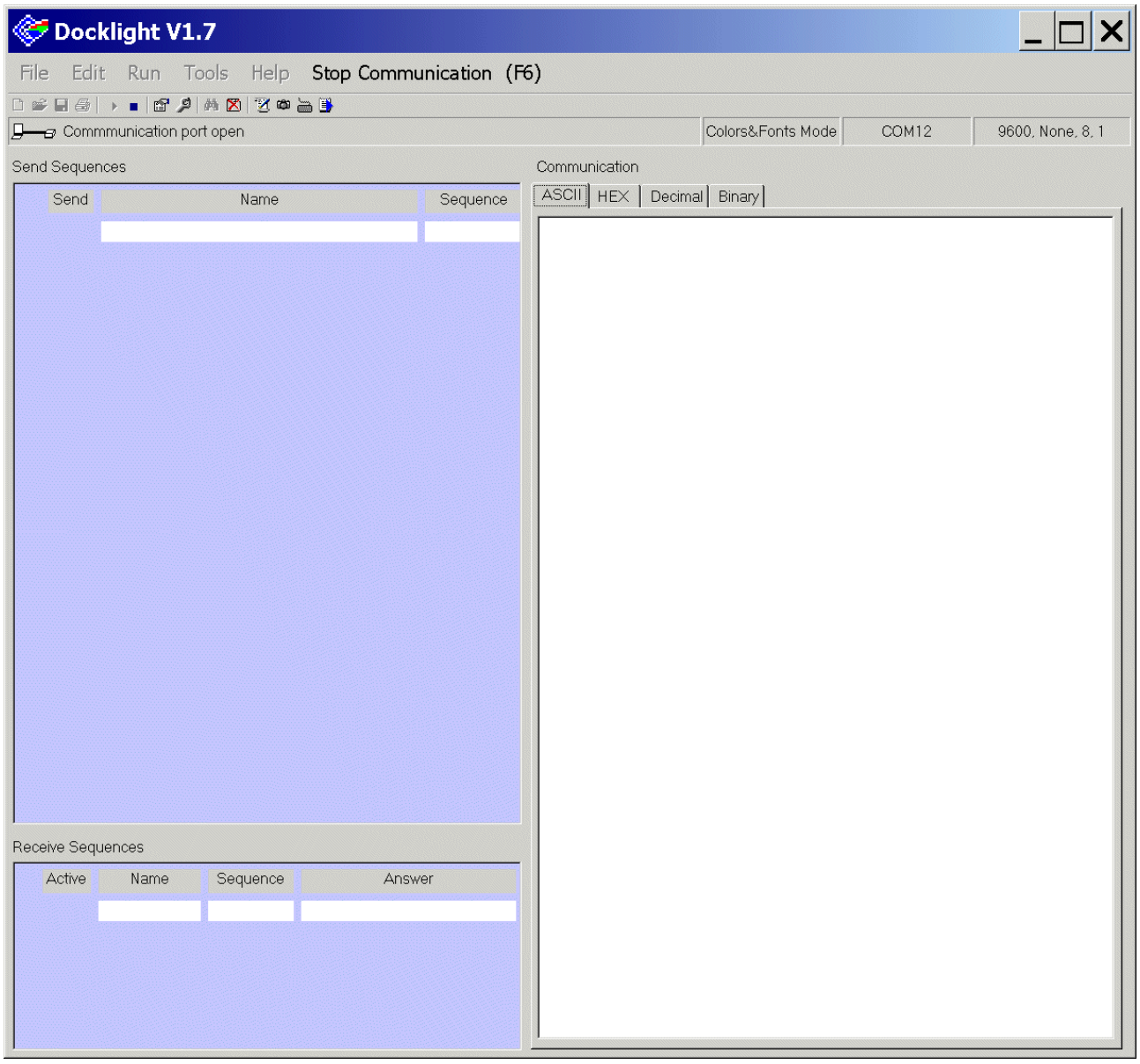
Project Settings:
Communication Filter: Contents Filter: click ⊙ Show all original communication data



OK

Click:

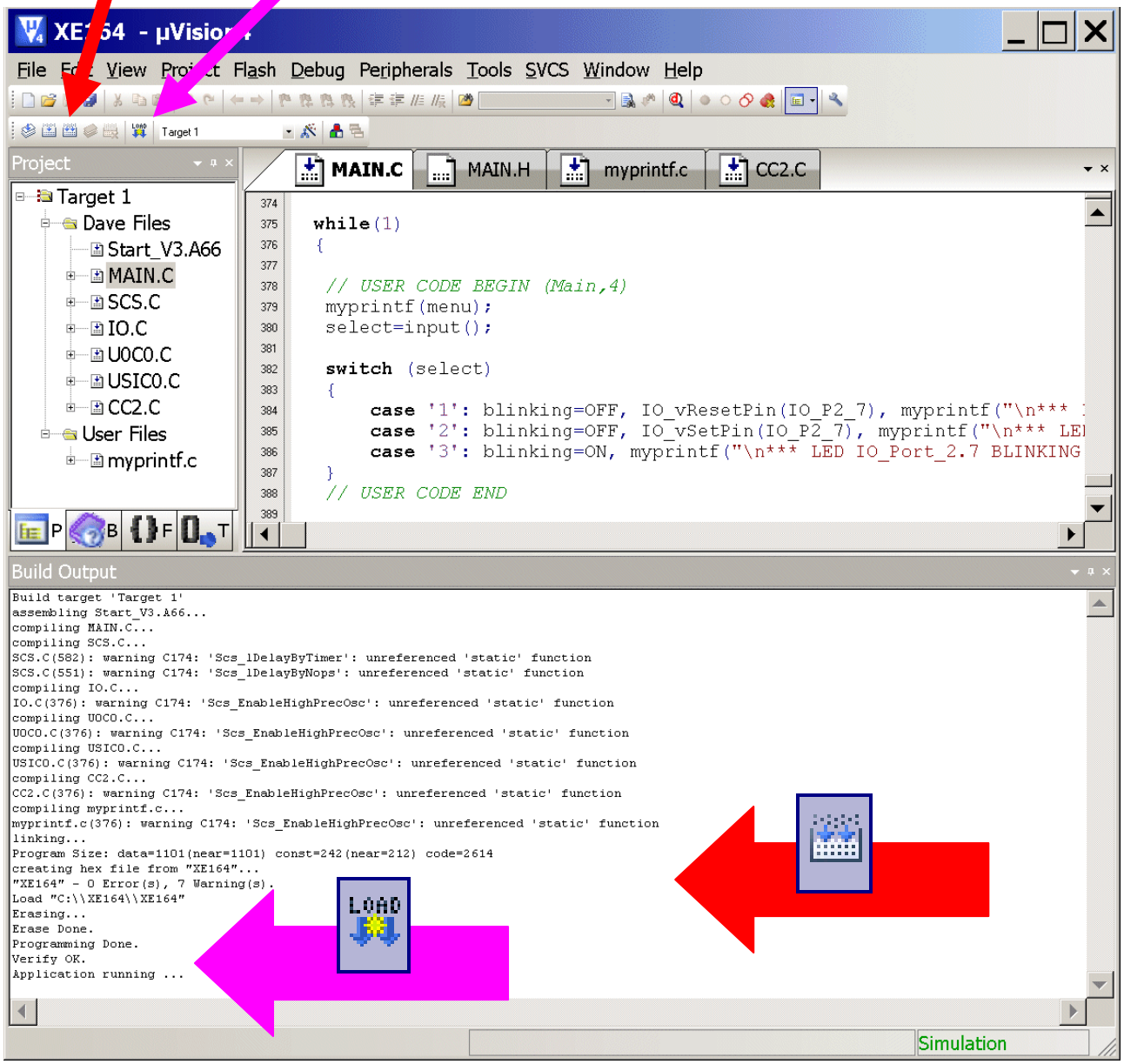

Start Communication
File
Run
Tools
Help
Colors&Fonts Mode
COM12
9600, None, 8, 1
Send Sequences
Send
Name
Sequence
Communication
ASCII
HEX
Decimal
Binary
Receive Sequences
Active
Name
Sequence
Answer



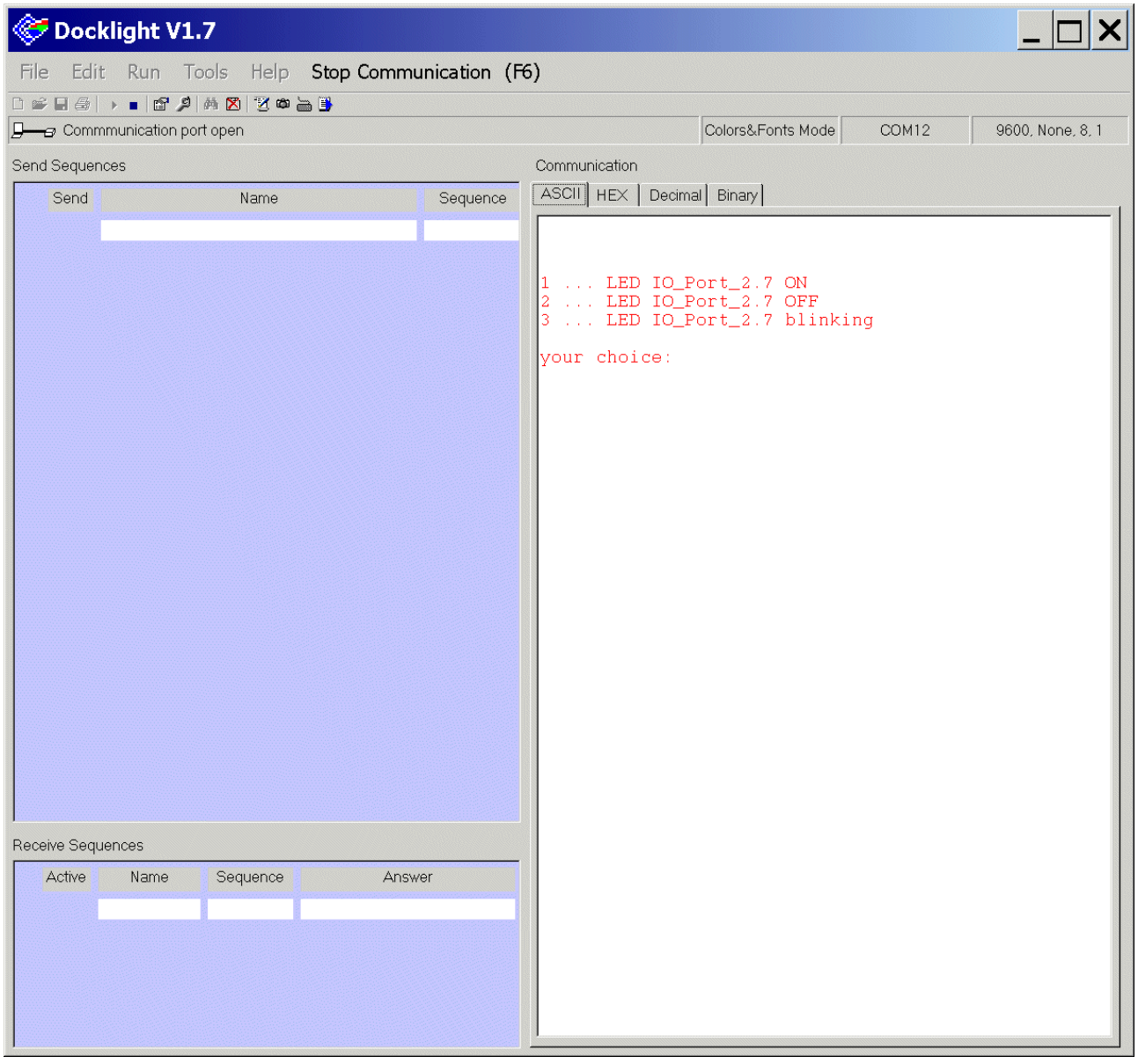
Note:
Docklight is now ready for serial communication!

Go to µVision4:

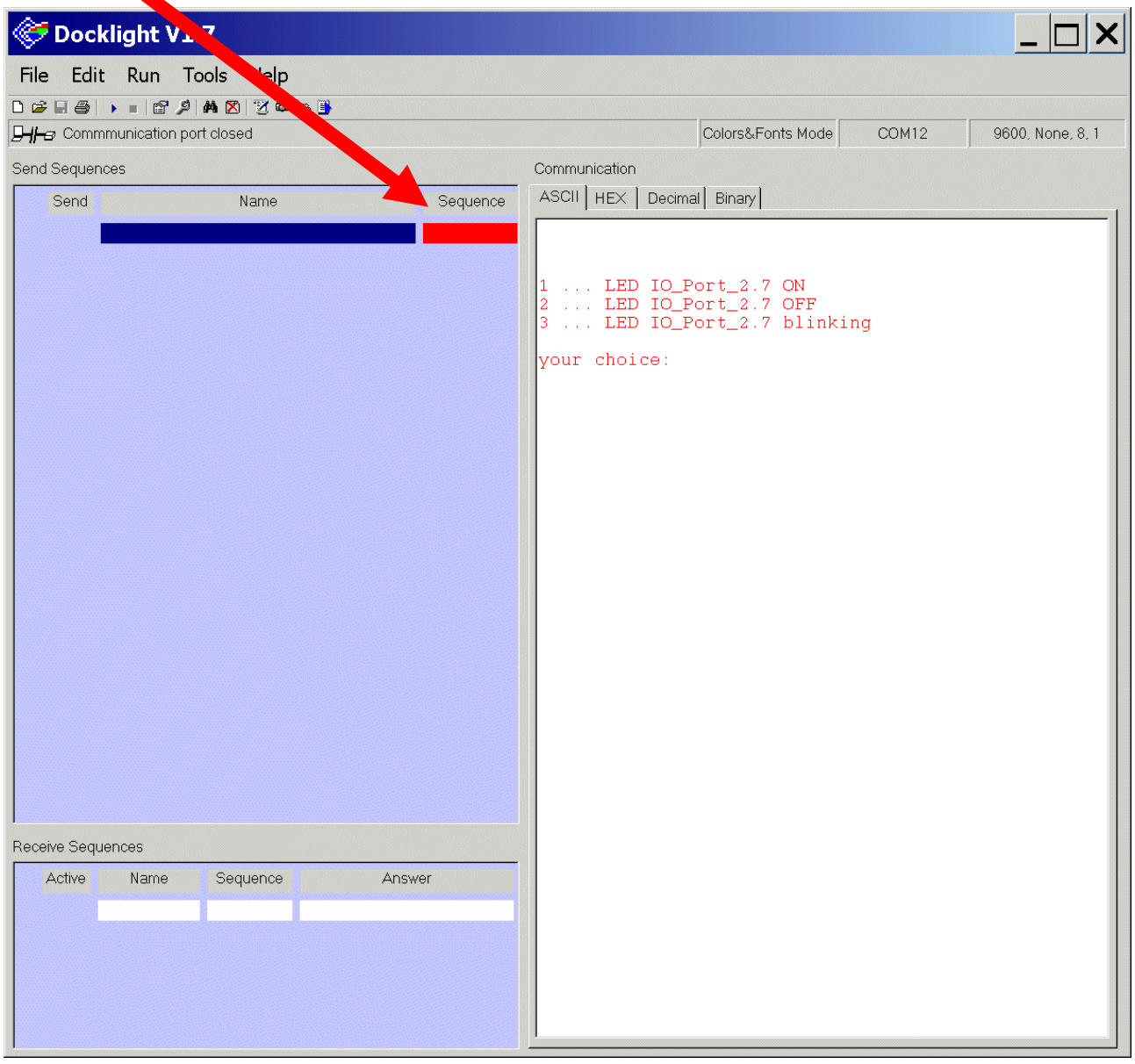
1.) click: 2.) click:



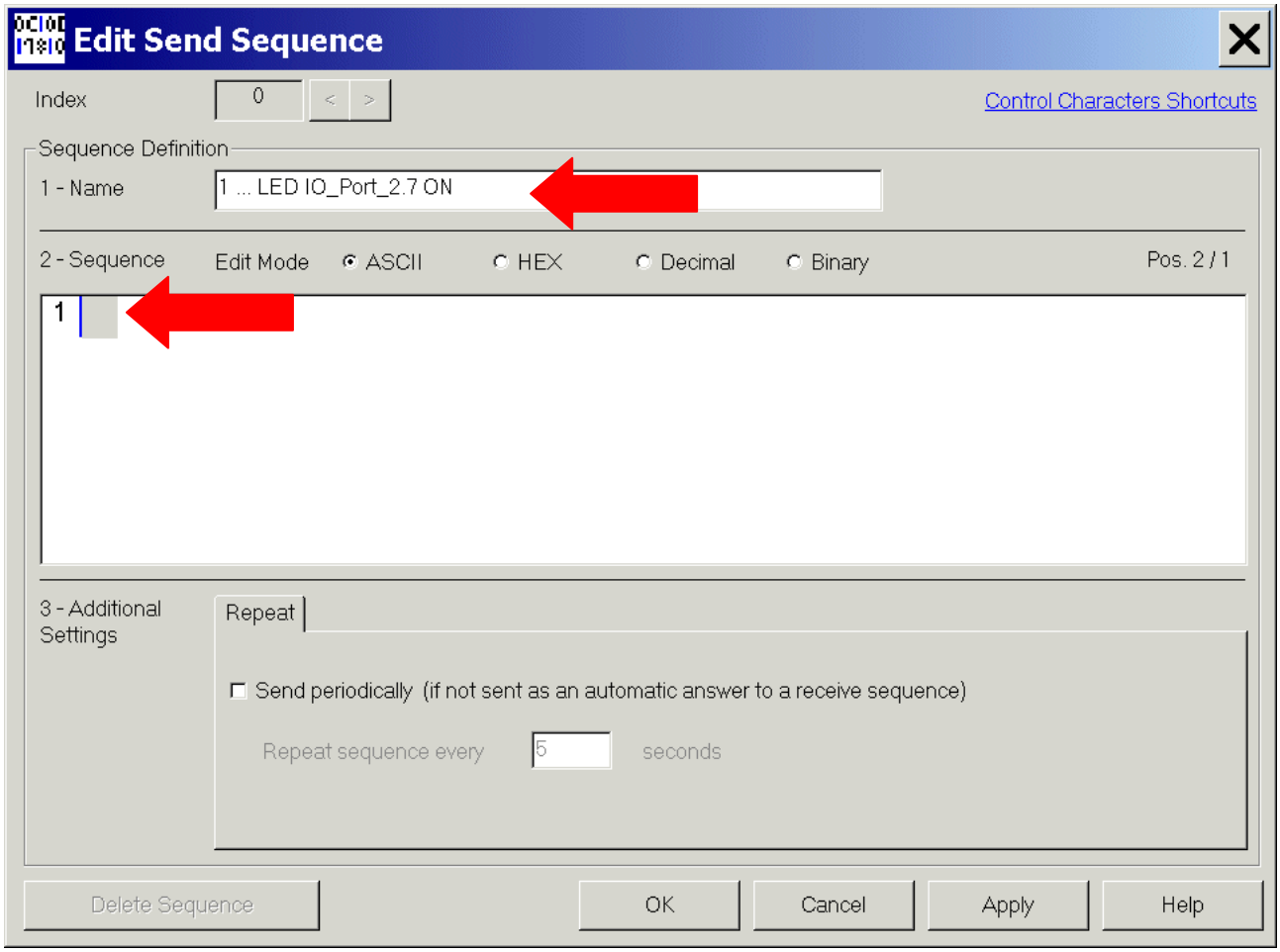
Go to Docklight and see the result:



Double click inside the red box:

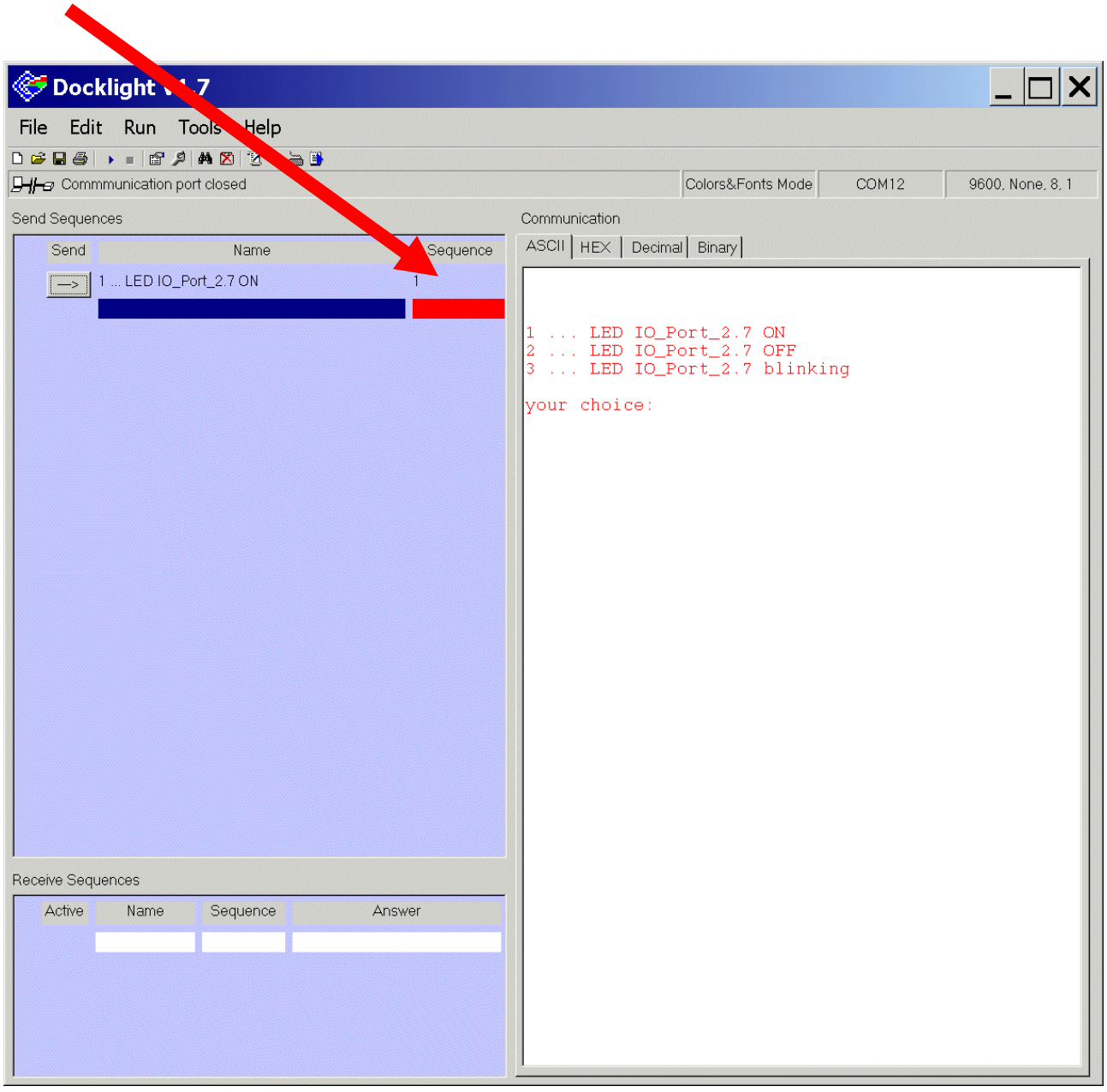


Edit Send Sequence: Sequence Definition: 1- Name: insert: 1 ... LED IO_Port_2.7 ON
Edit Send Sequence: Sequence Definition: 2- Sequence: insert: 1

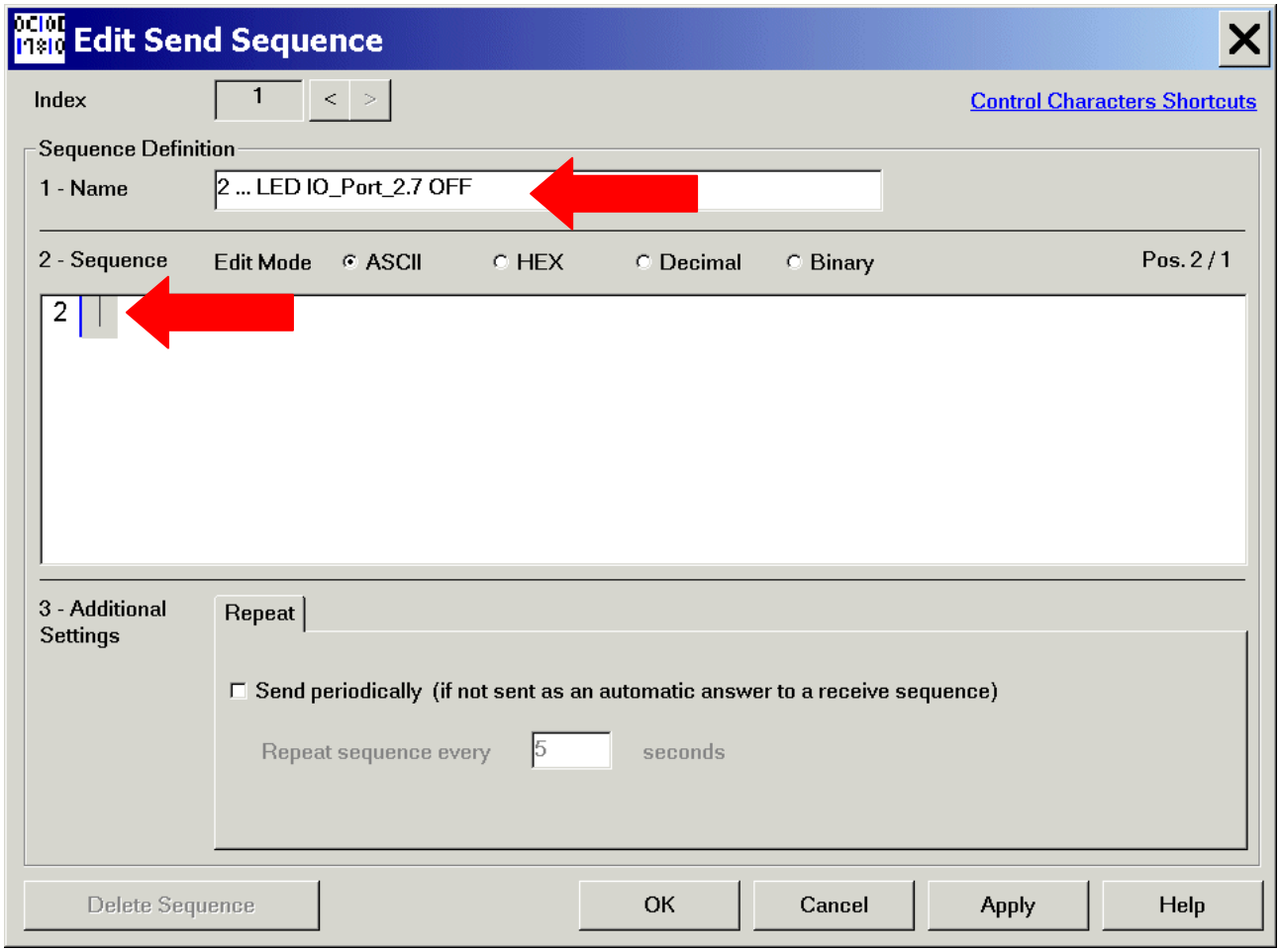


OK

Double click inside the red box:

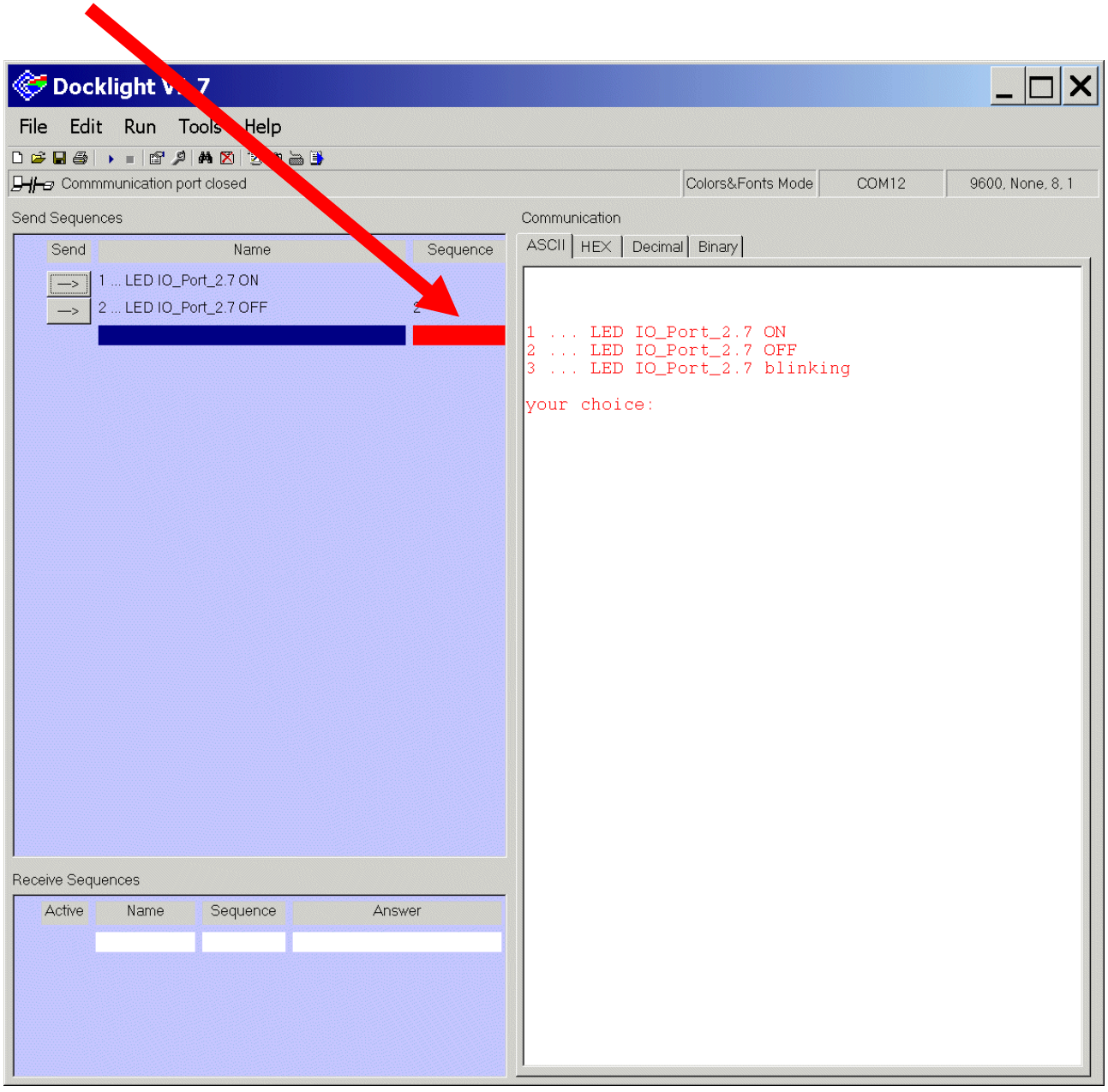


Edit Send Sequence: Sequence Definition: 1- Name: insert: 2 ... LED IO_Port_2.7 OFF
Edit Send Sequence: Sequence Definition: 2- Sequence: insert: 2

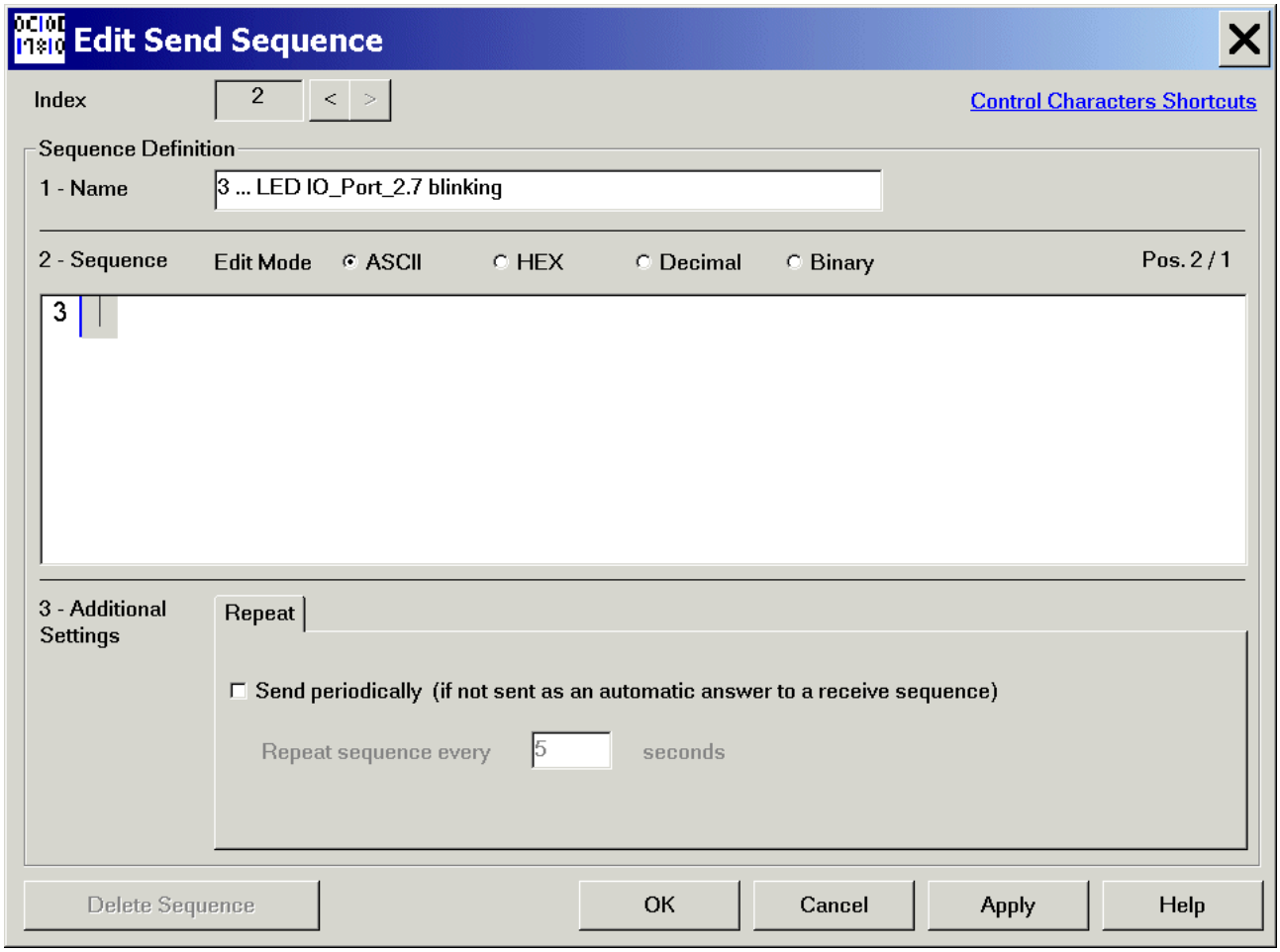


OK

Double click inside the red box:



Edit Send Sequence: Sequence Definition: 1- Name: insert: 3 ... LED IO_Port_2.7 blinking
Edit Send Sequence: Sequence Definition: 2- Sequence: insert: 3

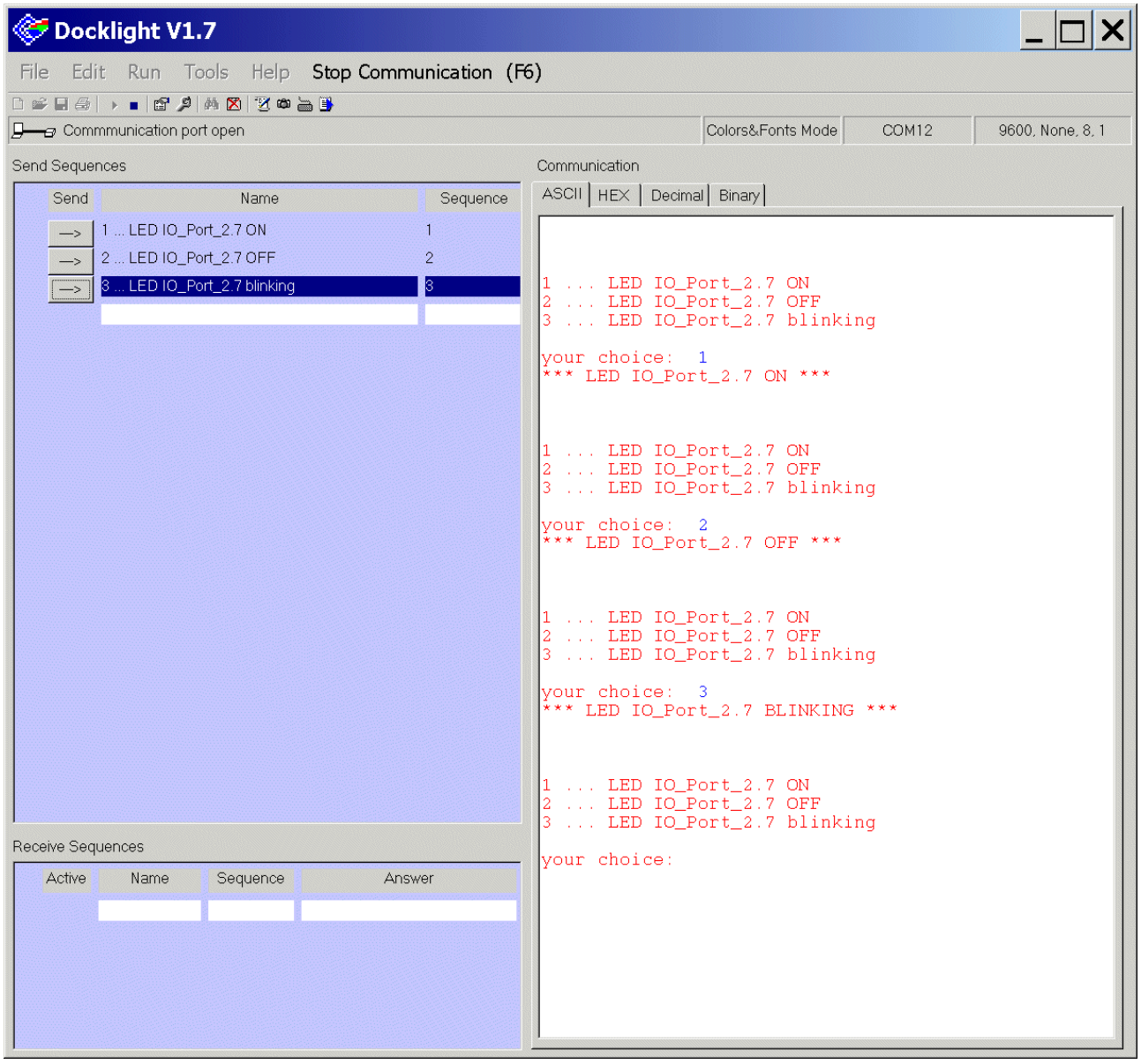


OK

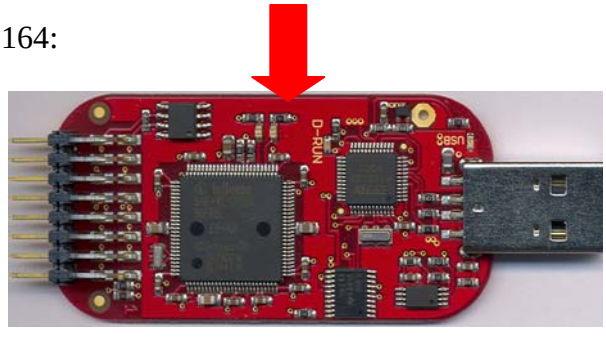
Click 1 ... LED IO_Port_2.7 ON or

Click 2 ... LED IO_Port_2.7 OFF or

Click 3 ... LED IO_Port_2.7 blinking



and check the results on your UConnect-CAN XE164:



Now we close our project and µVision 4:

Project - Close Project

File
Exit

Conclusion:

In this step-by-step book you have learned how to use the UConnect-CAN XE164 together with the Keil tool chain.

Now you can easily expand your ”hello world” program to suit your needs!

You can connect either a part of - or your entire application to the UConnect-CAN XE164.

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.

Have fun and enjoy working with XE16x microcontrollers!

Note:

There are step-by-step books for 8 bit microcontrollers (e.g. XC866, XC888 and XC878), 16 bit microcontrollers (e.g. C16x, XC16x and XE16x/XC2xxx) and 32 bit microcontrollers (e.g. TC1796 and TC1130).

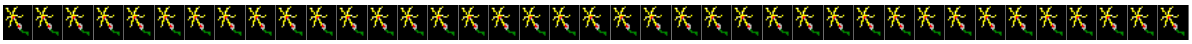
All these step-by-step books use the same microcontroller resources and the same example code.

This means: configuration steps, function names and variable names are identical.

This should give you a good opportunity to get in touch with another Infineon microcontroller family or tool chain!

There are even more programming examples using the same style available [e.g. ADC-examples, CAPCOM6-examples (e.g. BLDC-Motor, playing music), Simulator-examples, C++ examples] based on these step-by-step books.

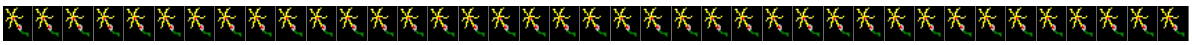
6.) Feedback (UConnect-CAN XE164, Keil tools, µVision4): Your opinion, suggestions and/or criticisms



Contact Details (this section may remain blank 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

Published by Infineon Technologies AG