

AP08048

XC866-4FR

XC866-4FR Starter-Kit "Cookery-Book" for a
"Hello world" application

Microcontrollers



Never stop thinking

Edition 2008-07-11

**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 7.](#)

Introduction:

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

It will help inexperienced users to get an XC866-4FR 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 XC866-4FR-Starter-Kit!

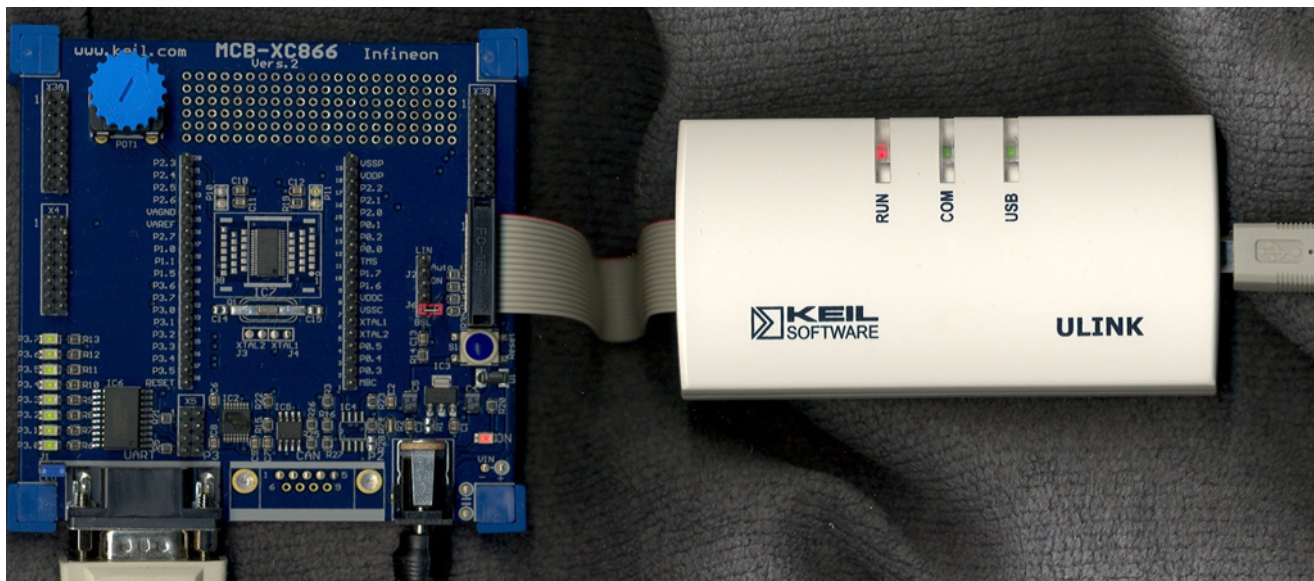
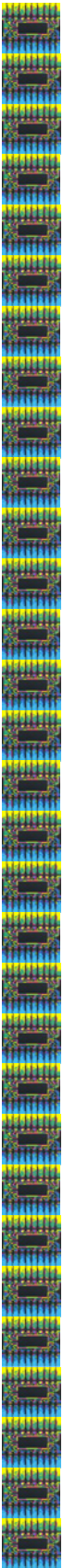


The New
XC800-Family
Designed to Make the Difference

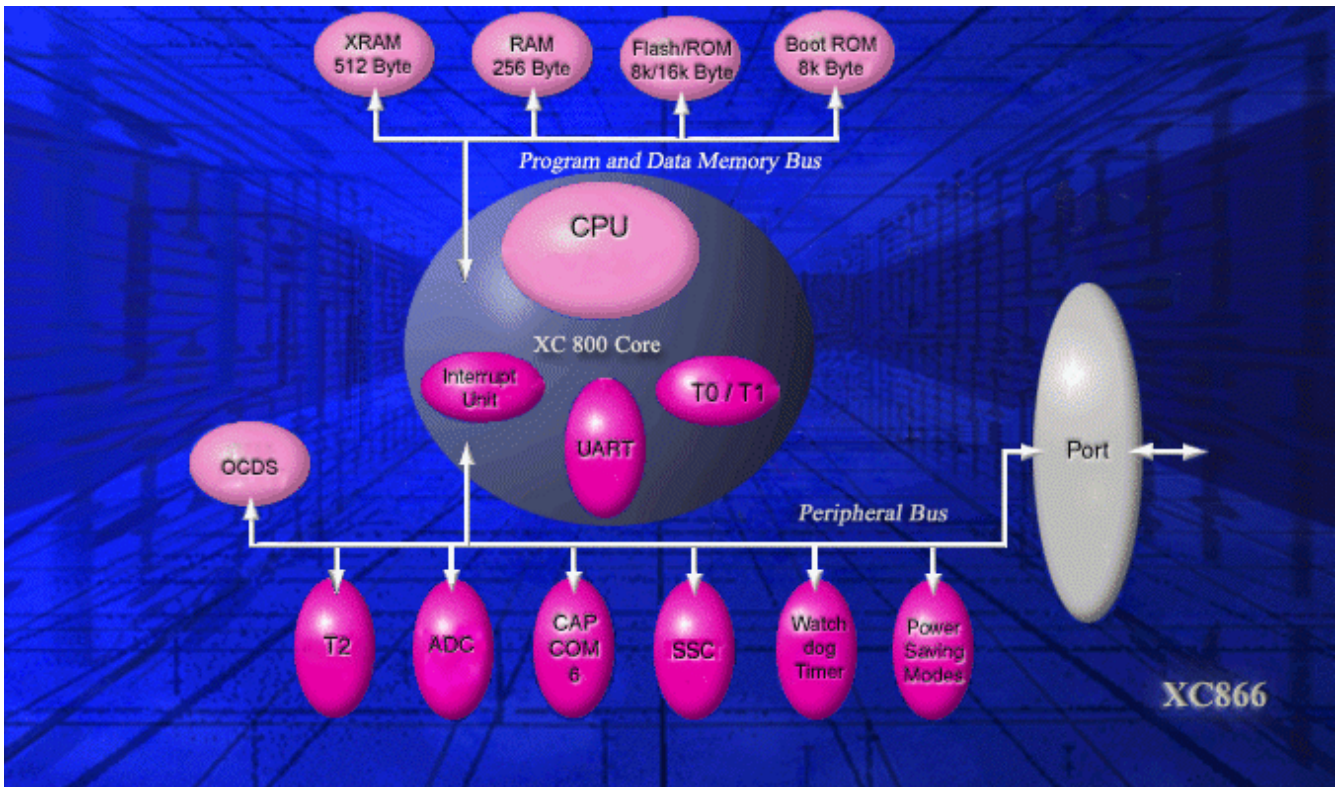
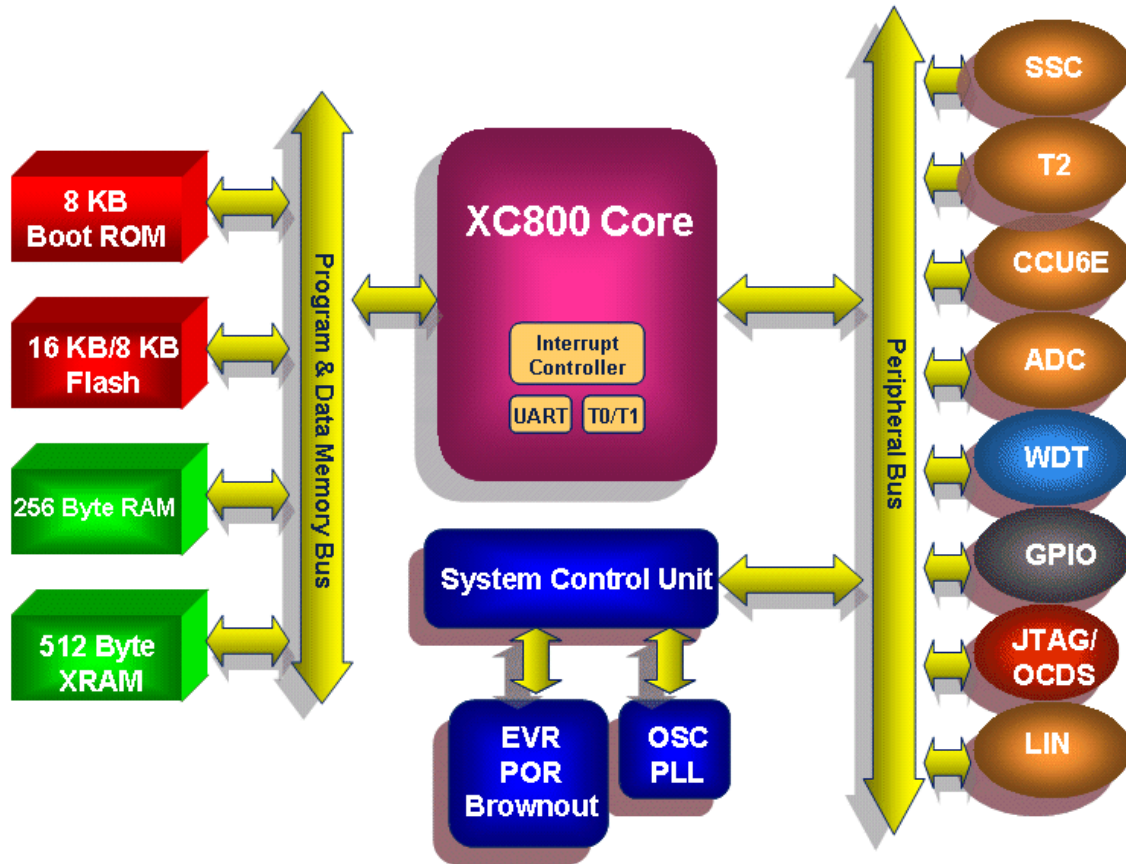
Programming Example

XC866-4FR

Starter Kit

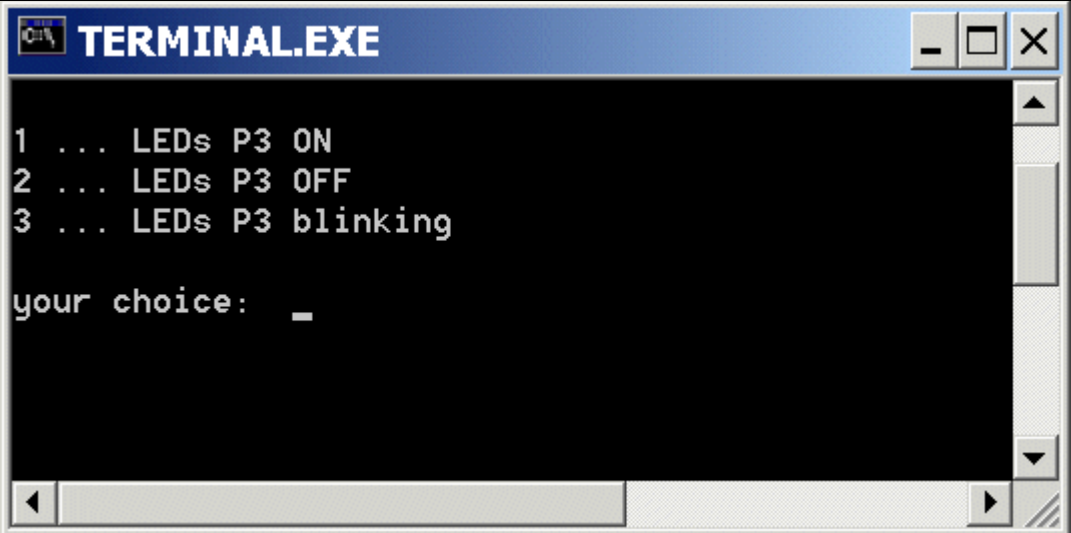


XC866-4FR Block Diagram



“Cookery-book“

For your first programming example for the XC866-4FR Starter-Kit-Board:

Your program:	
Chapter/ Step	*** Recipes ***
1.)	XC866-4FR Starter Kit Board Power Supply, Jumper Setting, Serial Cable to the notebook
2.)	DAvE (program generator) DAvE Installation (mothersystem) + DAvE Update Installation (XC866-4FR.DIP) for XC866-4FR
3.)	Using DAvE Microcontroller initialization for your programming example
4.)	Using the KEIL Development Tools (C-Compiler) Programming of your application (XC866-4FR) with KEIL tool chain (µVision3) Compiler V8.04 + first steps with the Simulator

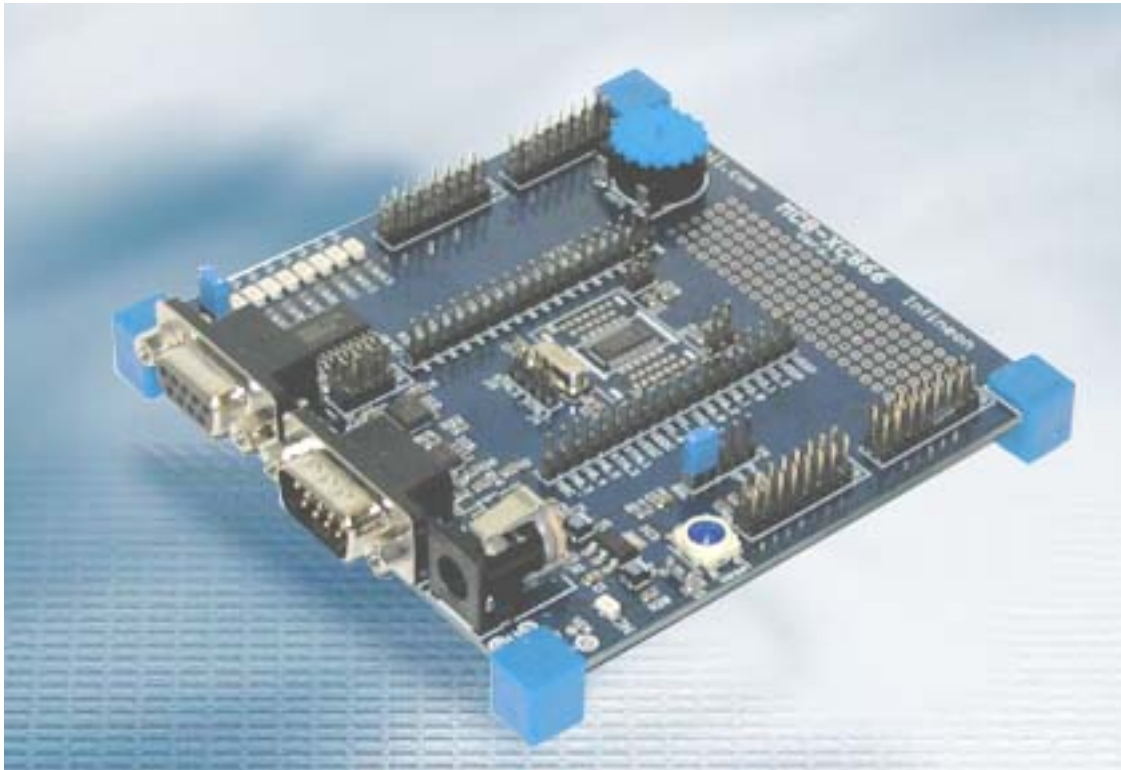
Additional exercises

part- µVision3	µVision3 - Open Linker COD File
k_50	CAPCOM6: PWM generation (asymmetrical/edge-aligned, single-shot, modulated) – playing music
part_ADC1	ADC1

Feedback

5.)	Feedback
-----	--------------------------

1.) XC866-4FR Starter Kit Board:



Ordering information:

Starter Kits - Type	Title	μ C	Ordering No.
SK-XC866	XC866 Starter Kit, X-Board	XC866-4FR	B158-H8548-X-0-7600
SK-XC866 Easykit	XC866 Easy Kit, Mini- X-Board	XC866-4FR	B158-H8576-X-0-7600

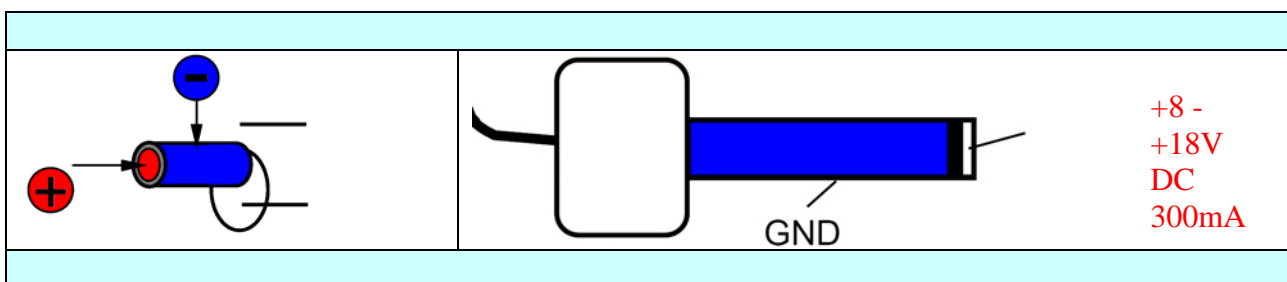
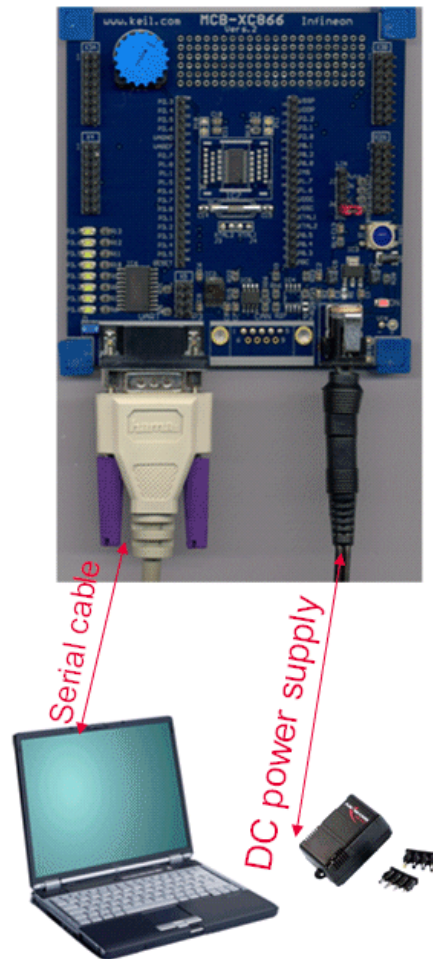
Distribution Worldwide:

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

You need a **Power Supply**:

The XC866-4FR Board requires an external 8-18V DC, 300mA power supply.

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 [XC866-4FR Board Manual, V0.1, Jan 2005](#) .

For further information, please refer to <http://www.keil.com/support/docs/3132.htm>

For further information, please refer to

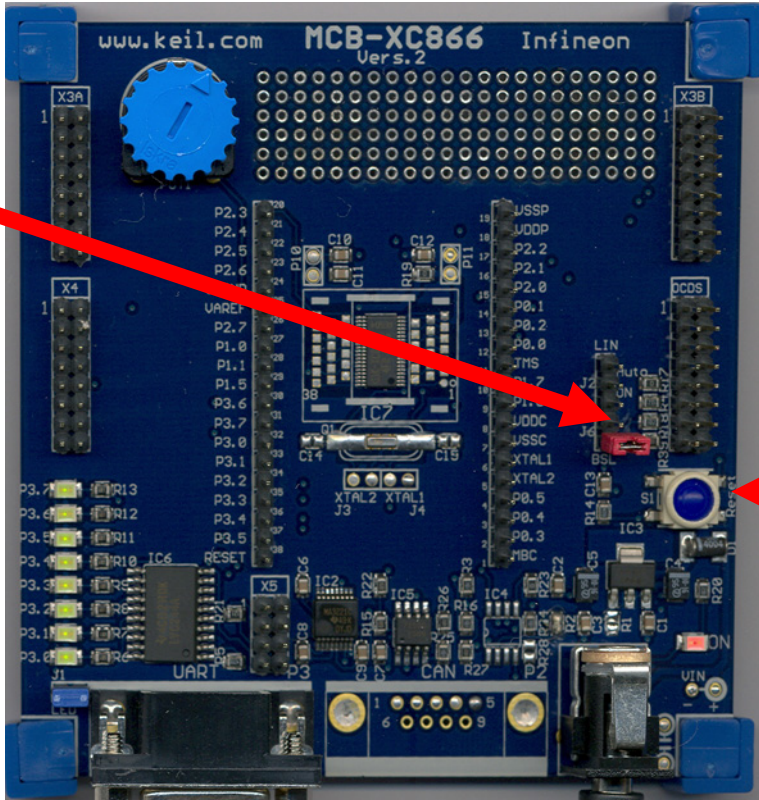
[MCBXC866 JUMPER2 DOES NOT WORK AS DESCRIBED.htm](#)

For further information, please refer to the [XC866-4FR Board Manual, V1.0, Jan 2006](#) .

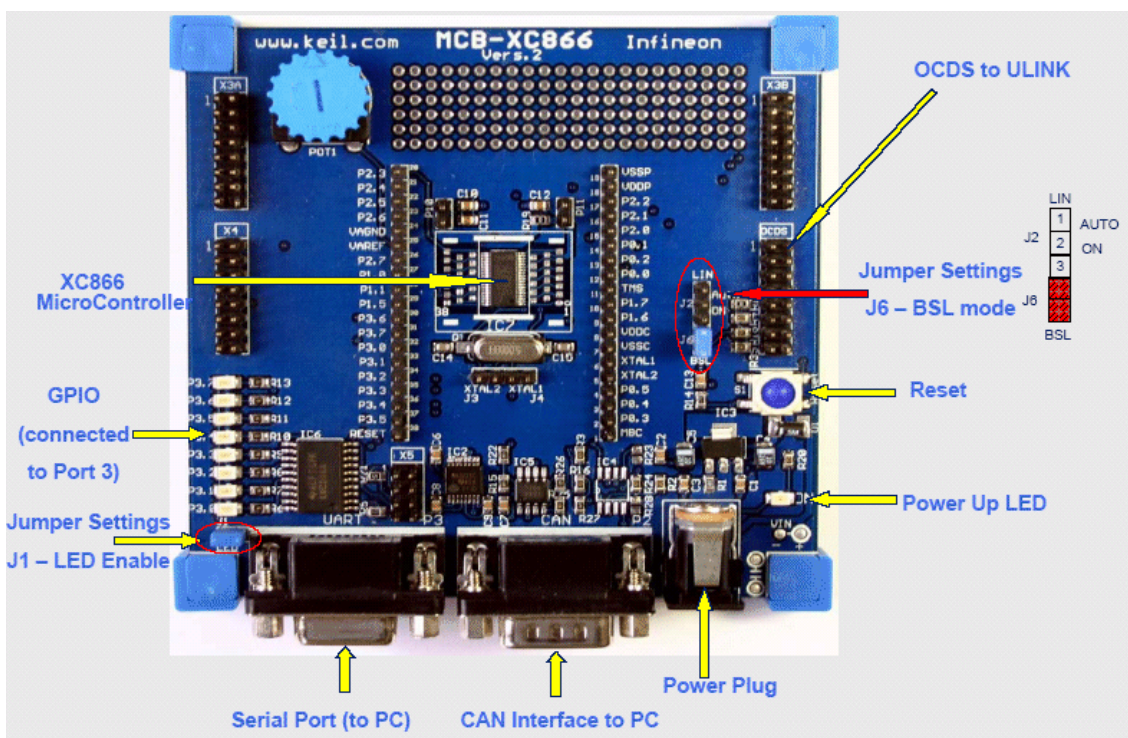
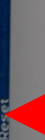
Bootstrap-Loader-ASC / Standard-Internal-Start:

1.)

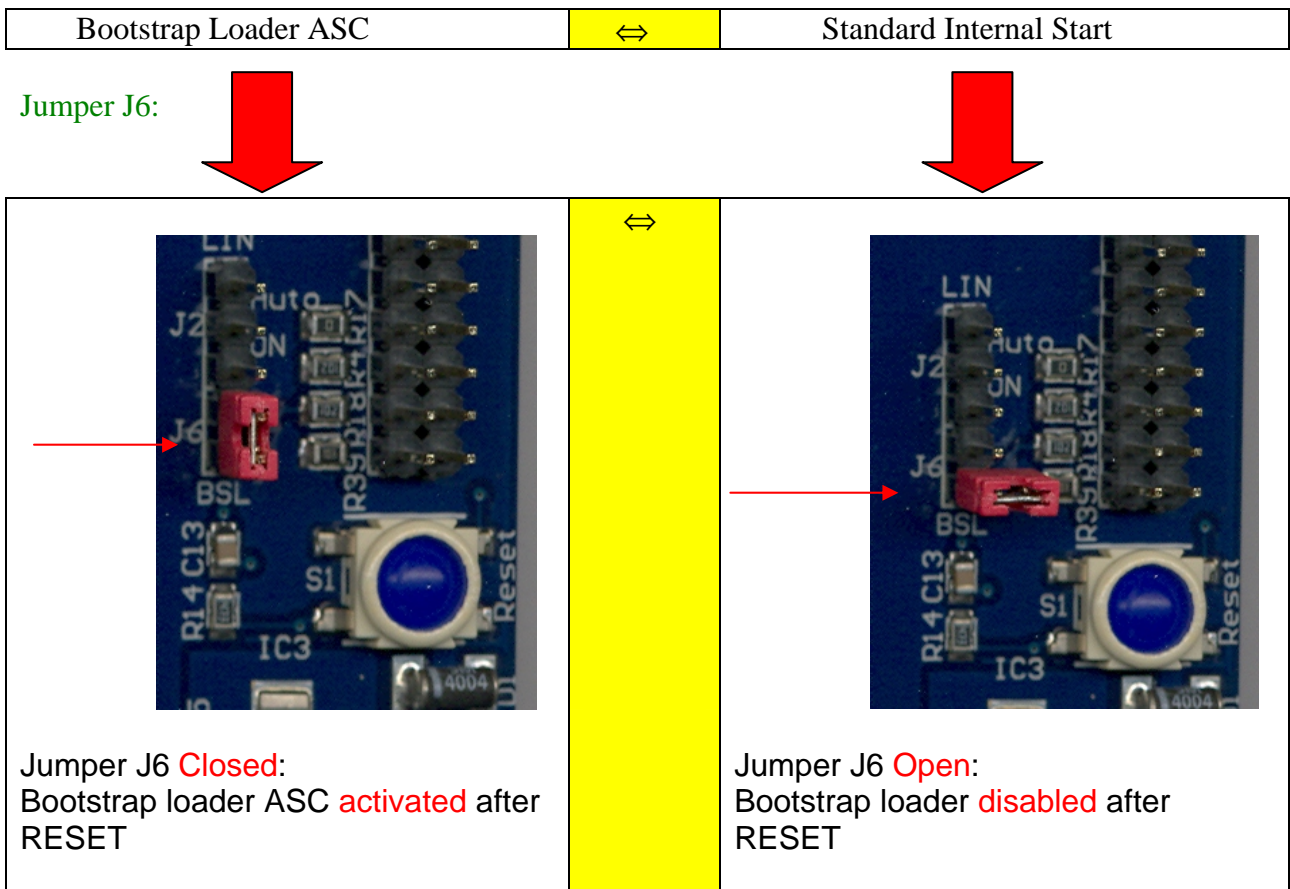
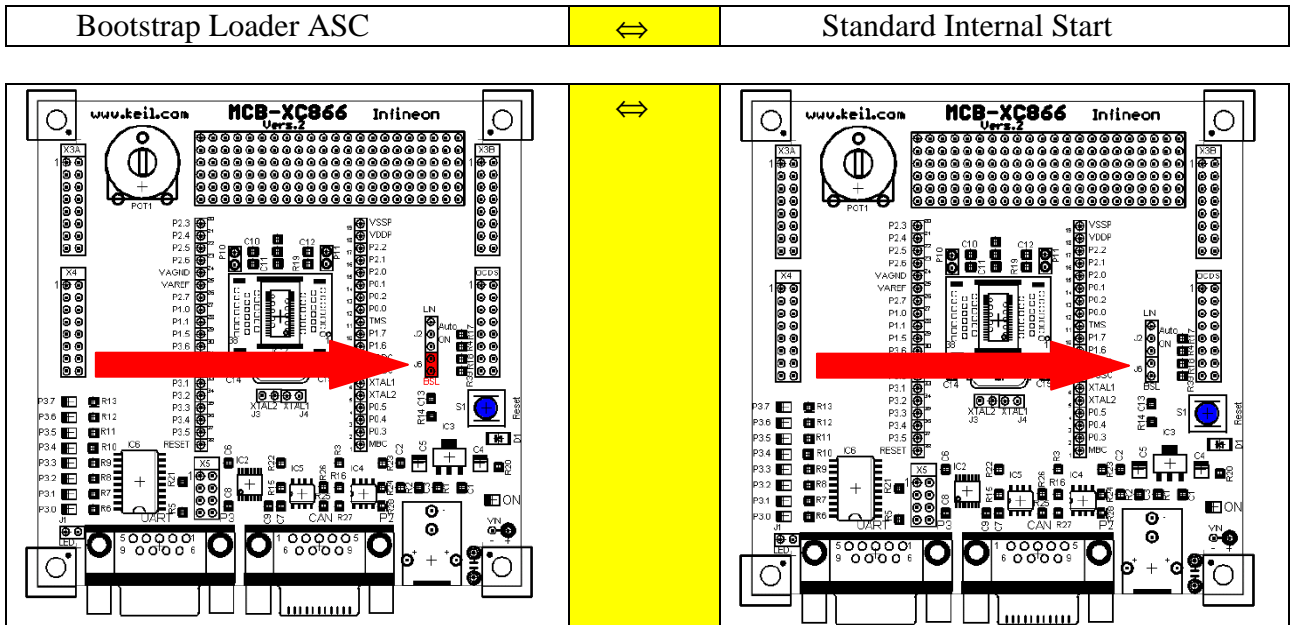
Select



2.) RESET



Jumper J6 - Jumper Setting:




2.) DAvE – Installation for XC866 microcontrollers:



Install DAvE (mothersystem):

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

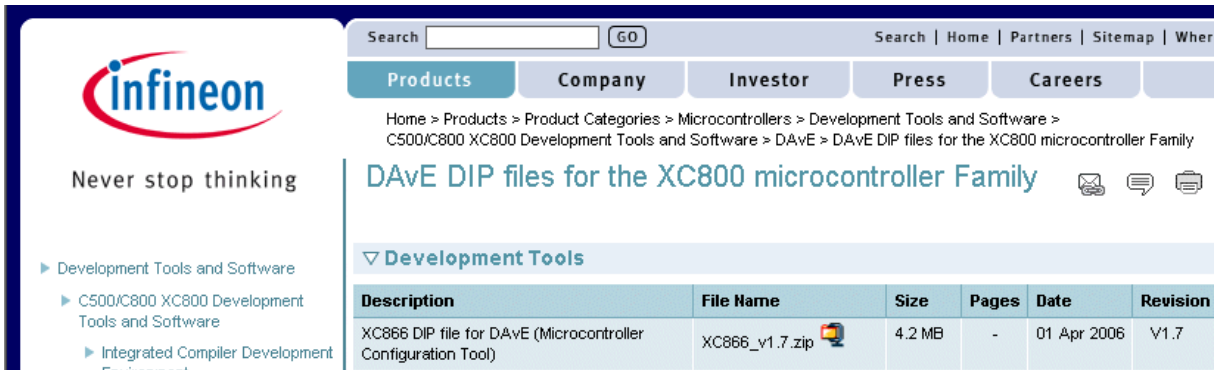
Description	File Name	Size	Date	Revision
DAvE - Mothersystem without derivatives	setup.exe 	15.2 MB	7 Jul 2005	

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

Install the XC866 microcontroller Update:

1.)

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



Home > Products > Product Categories > Microcontrollers > Development Tools and Software > C500/C800 XC800 Development Tools and Software > DAVe > DAVe DIP files for the XC800 microcontroller Family


DAvE DIP files for the XC800 microcontroller Family

Development Tools

Description	File Name	Size	Pages	Date	Revision
XC866 DIP file for DAVe (Microcontroller Configuration Tool)	XC866_v1.7.zip	4.2 MB	-	01 Apr 2006	V1.7

Unzip the zip-file “XC866_v1.7.zip” and save “XC866.DIP”
@ e.g. D:\DAvE\XC866-2006-07-12-ver.17 .

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\XC866-2006-07-12-ver.17](#)
Forward>
Select: Available Products
click ✓ [XC866](#)
Forward>
Install
End

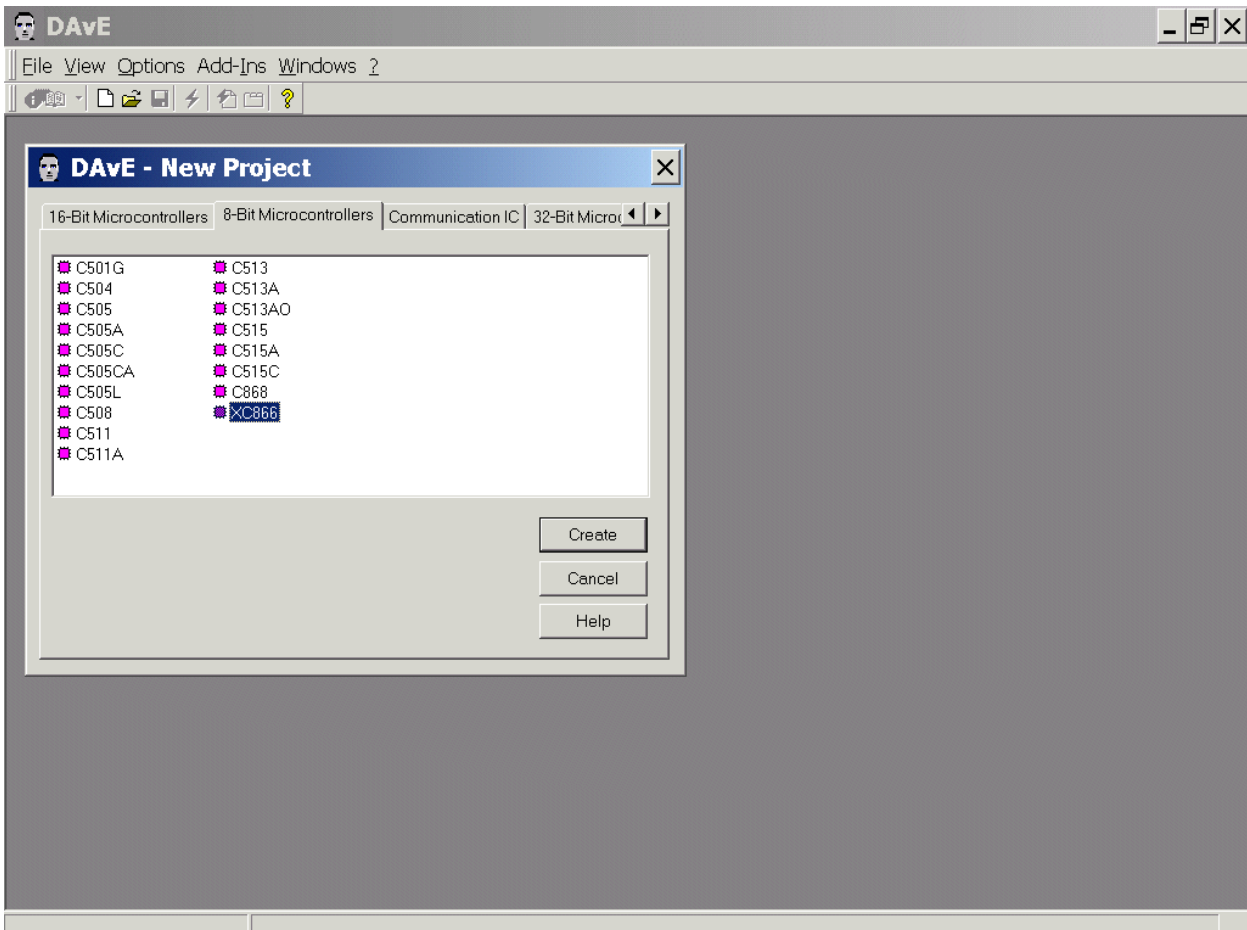
4.) DAVe is now ready to generate code for the XC866 microcontroller.

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



Start the program generator DAVe and select the XC866-4FR microcontroller:

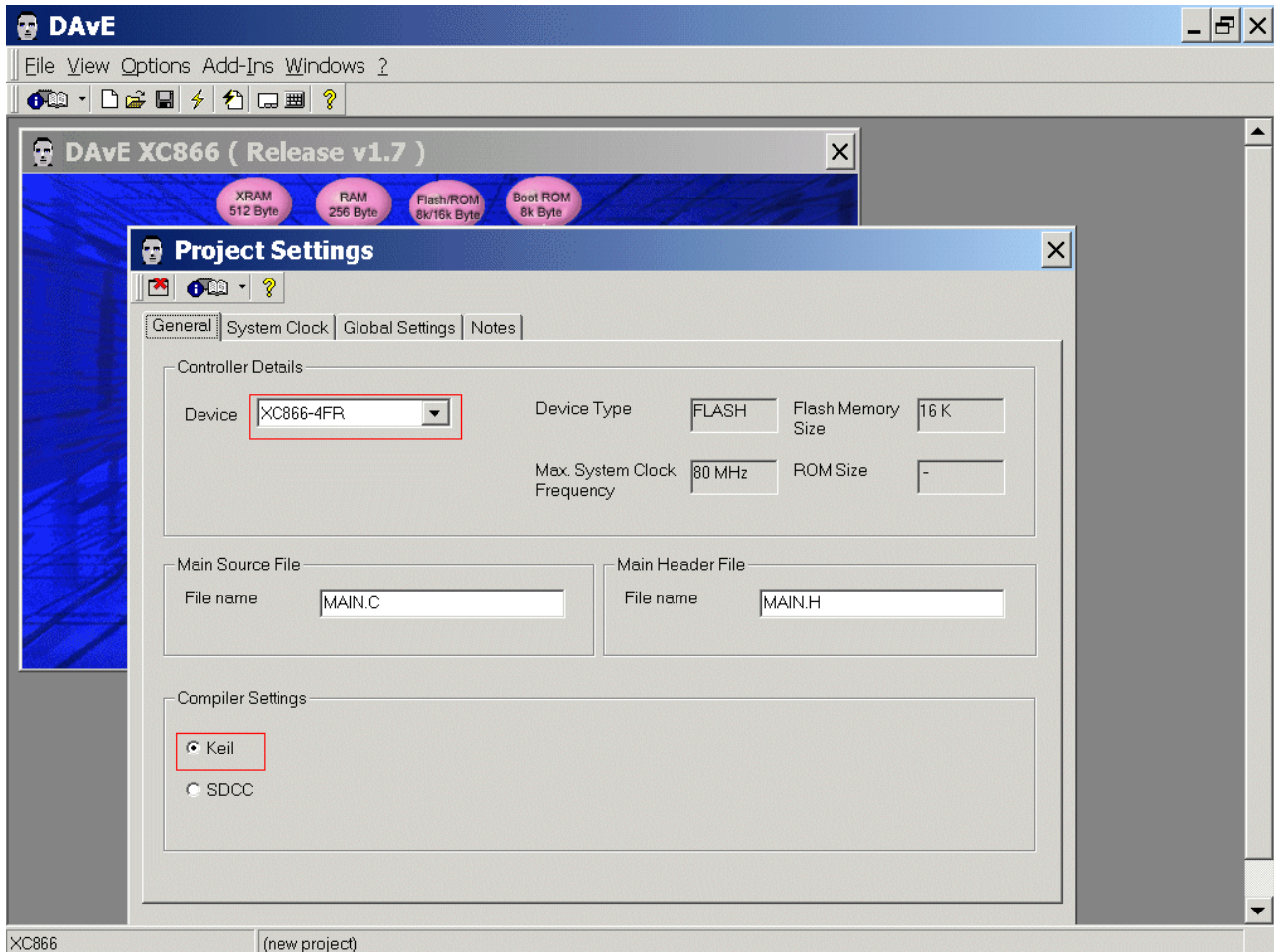
File
New
8-Bit Microcontrollers
select **XC866**
Create



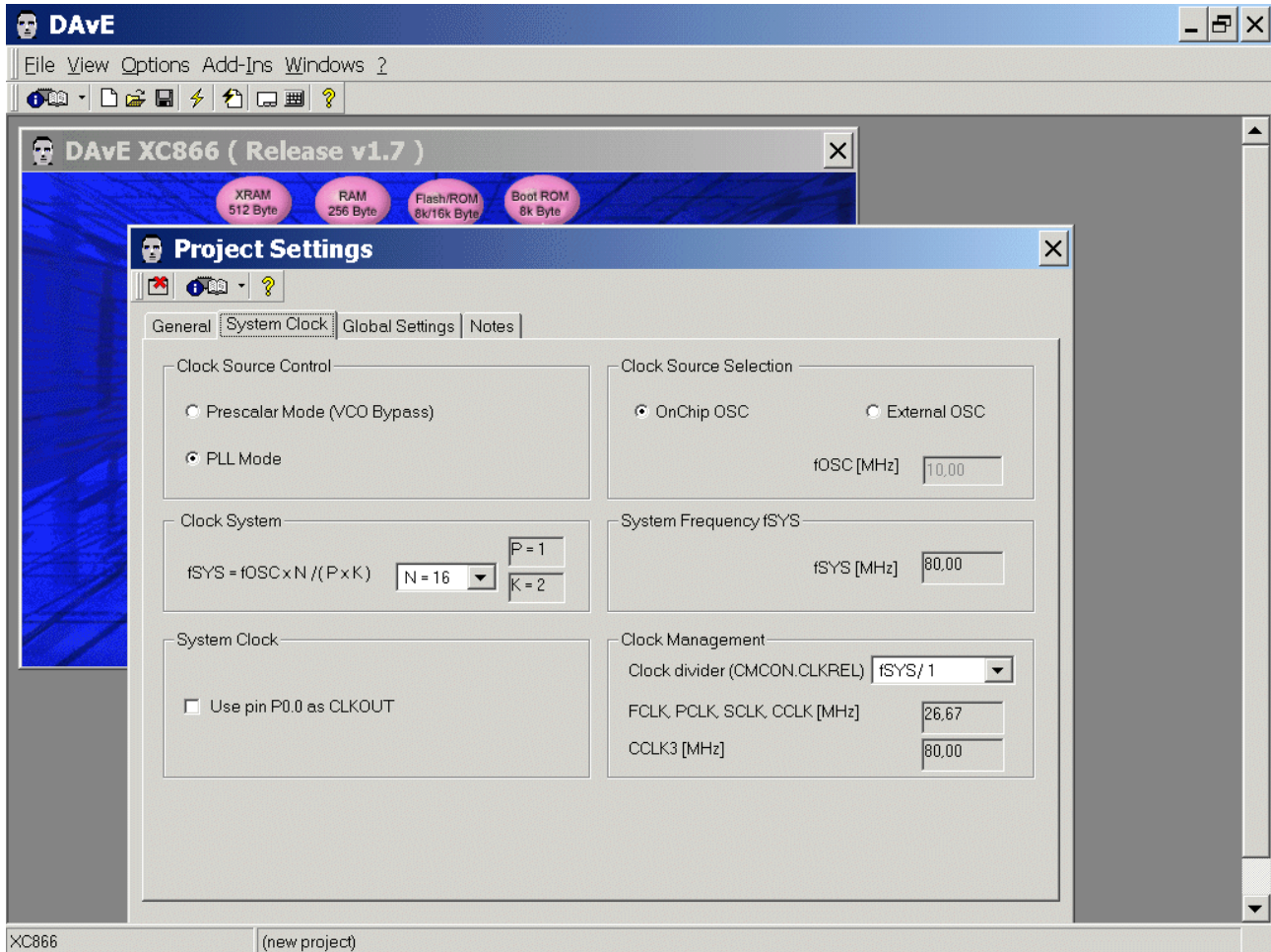
Choose the Project Settings as you can see in the Screenshots:

General: Controller Details: Device: **check/select** XC866-4FR

General: For the **KEIL** Compiler **check/choose** Keil in the Compiler Settings:



System Clock: (do nothing)

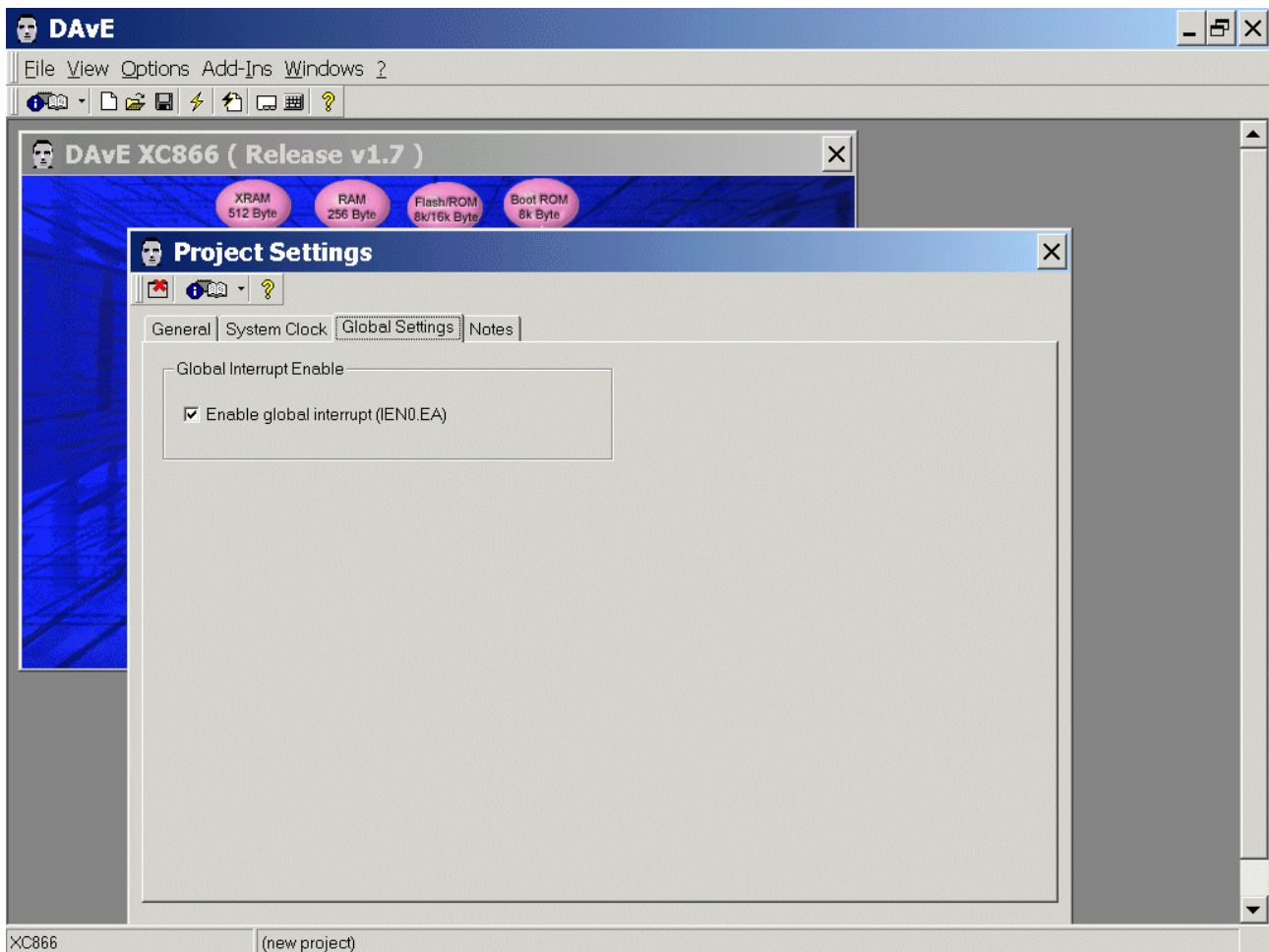


The screenshot shows the DAVE software interface with the 'Project Settings' dialog box open. The 'System Clock' tab is selected, and the following settings are visible:

- Clock Source Control:**
 - Prescalar Mode (VCO Bypass)
 - PLL Mode
- Clock Source Selection:**
 - OnChip OSC
 - External OSC
 - fOSC [MHz]: 10.00
- Clock System:**
 - fSYS = fOSC x N / (P x K)
 - N = 16 (dropdown)
 - P = 1 (dropdown)
 - K = 2 (dropdown)
- System Frequency fSYS:**
 - fSYS [MHz]: 80.00
- System Clock:**
 - Use pin P0.0 as CLKOUT
- Clock Management:**
 - Clock divider (CMCON.CLKREL): fSYS / 1 (dropdown)
 - FCLK, PCLK, SCLK, CCLK [MHz]: 26.67
 - CCLK3 [MHz]: 80.00

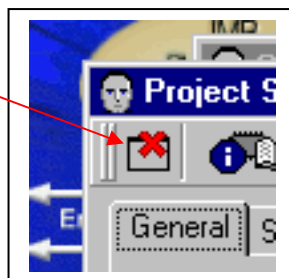
The status bar at the bottom of the dialog shows 'XC866' and '(new project)'.

Global Settings: (do not change configuration)



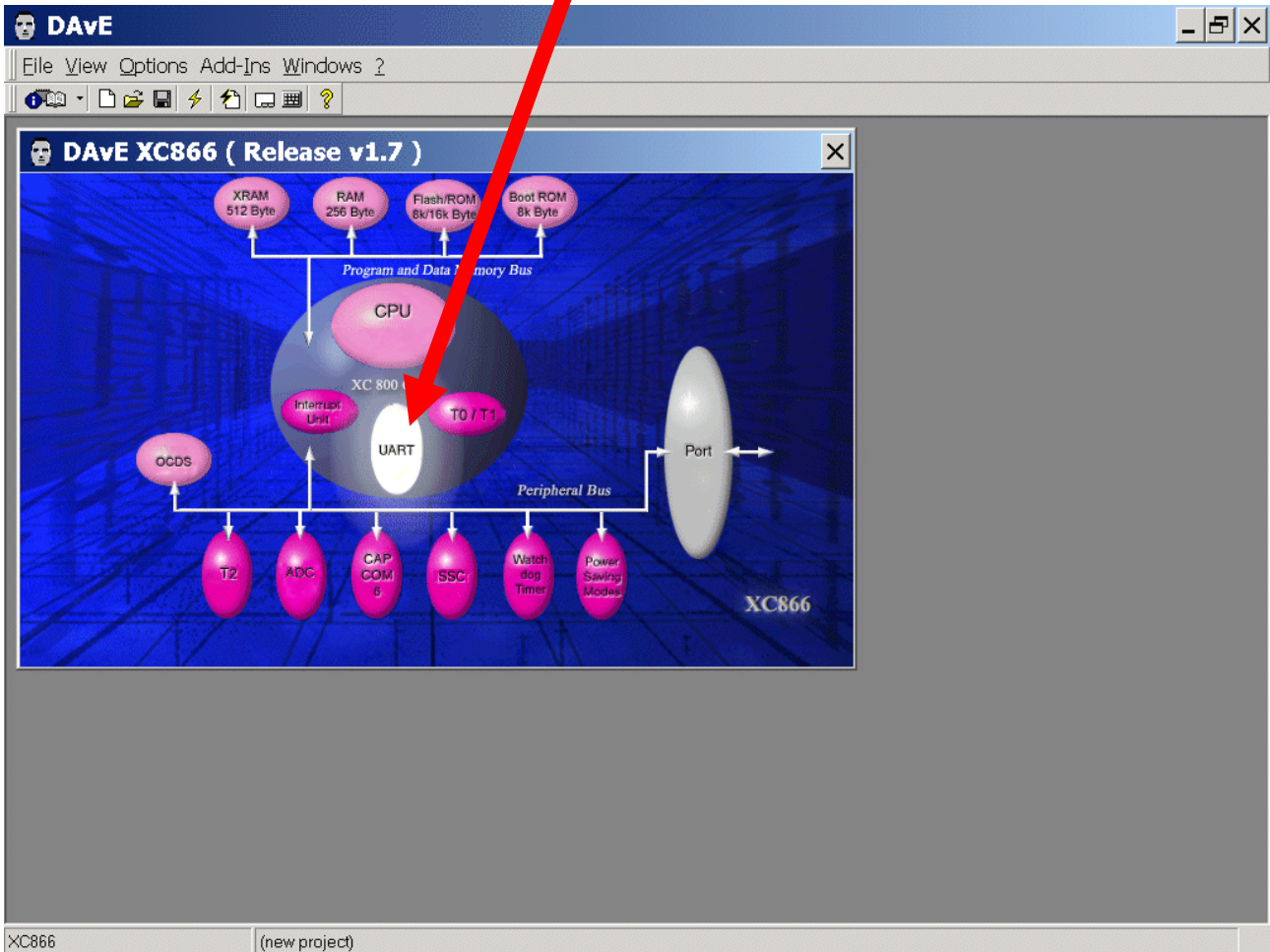
Notes: If you wish, you can insert your comments here.

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.

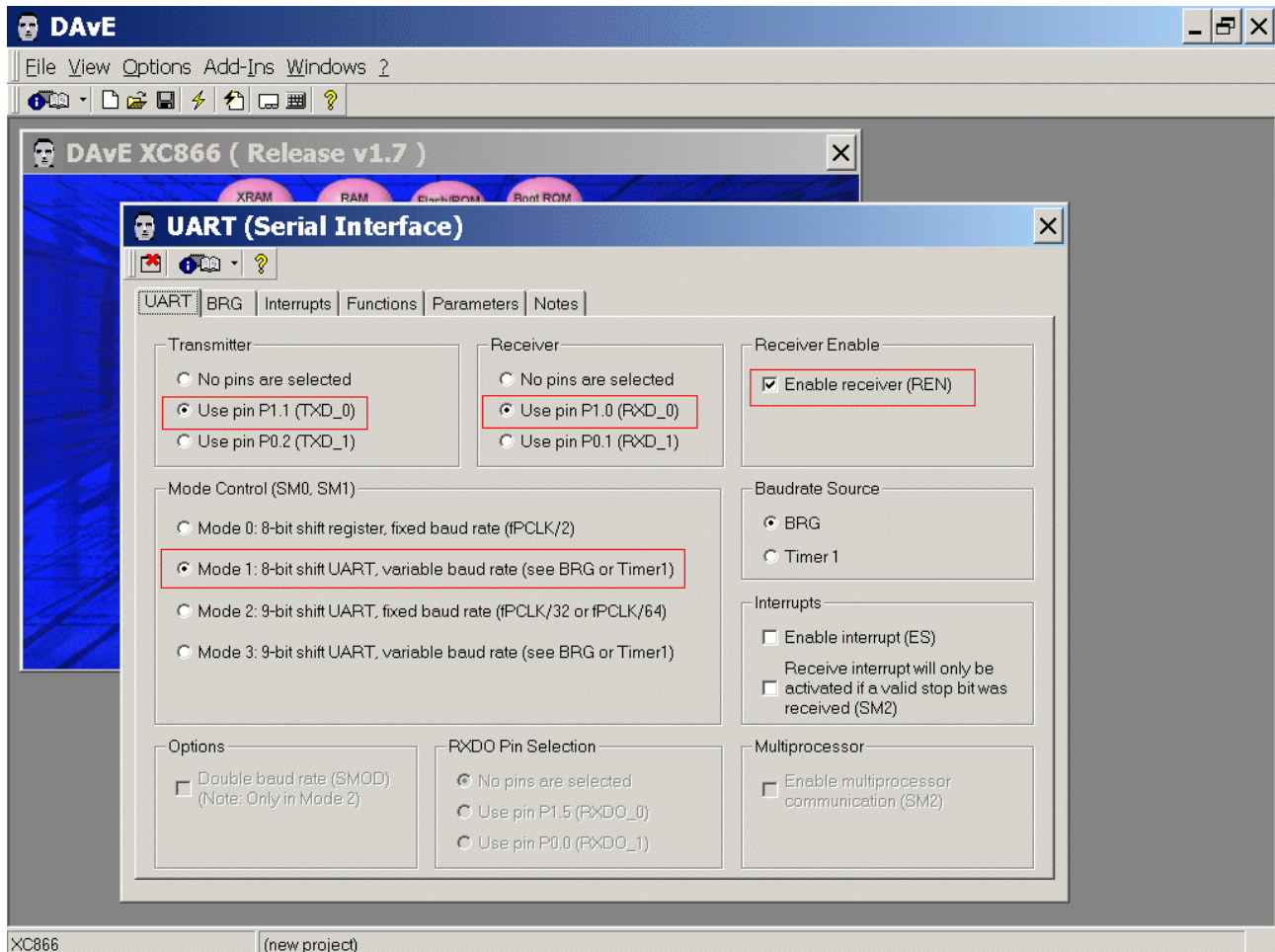


UART: Transmitter: **click** Use pin P1.1 (TXD_0)

UART: Receiver: **click** Use pin P1.0 (RXD_0)

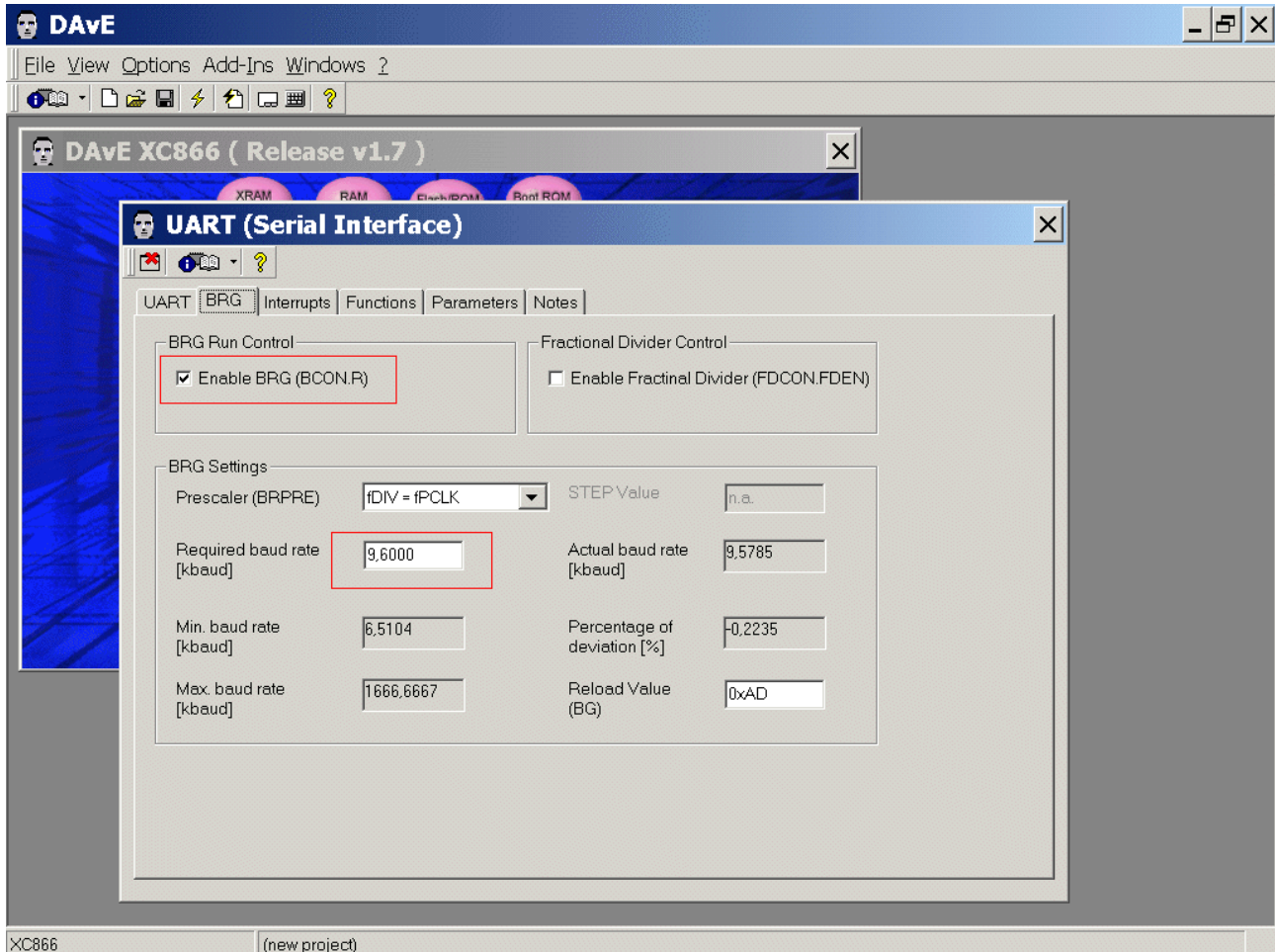
UART: Receiver Enable: **click** Enable receiver (REN)

UART: Mode Control: **click** Mode 1: 8-bit shift UART, variable baud rate (see BRG or Timer1)



BRG: BRG Run Control: **click/check** ✓ Enable BRG

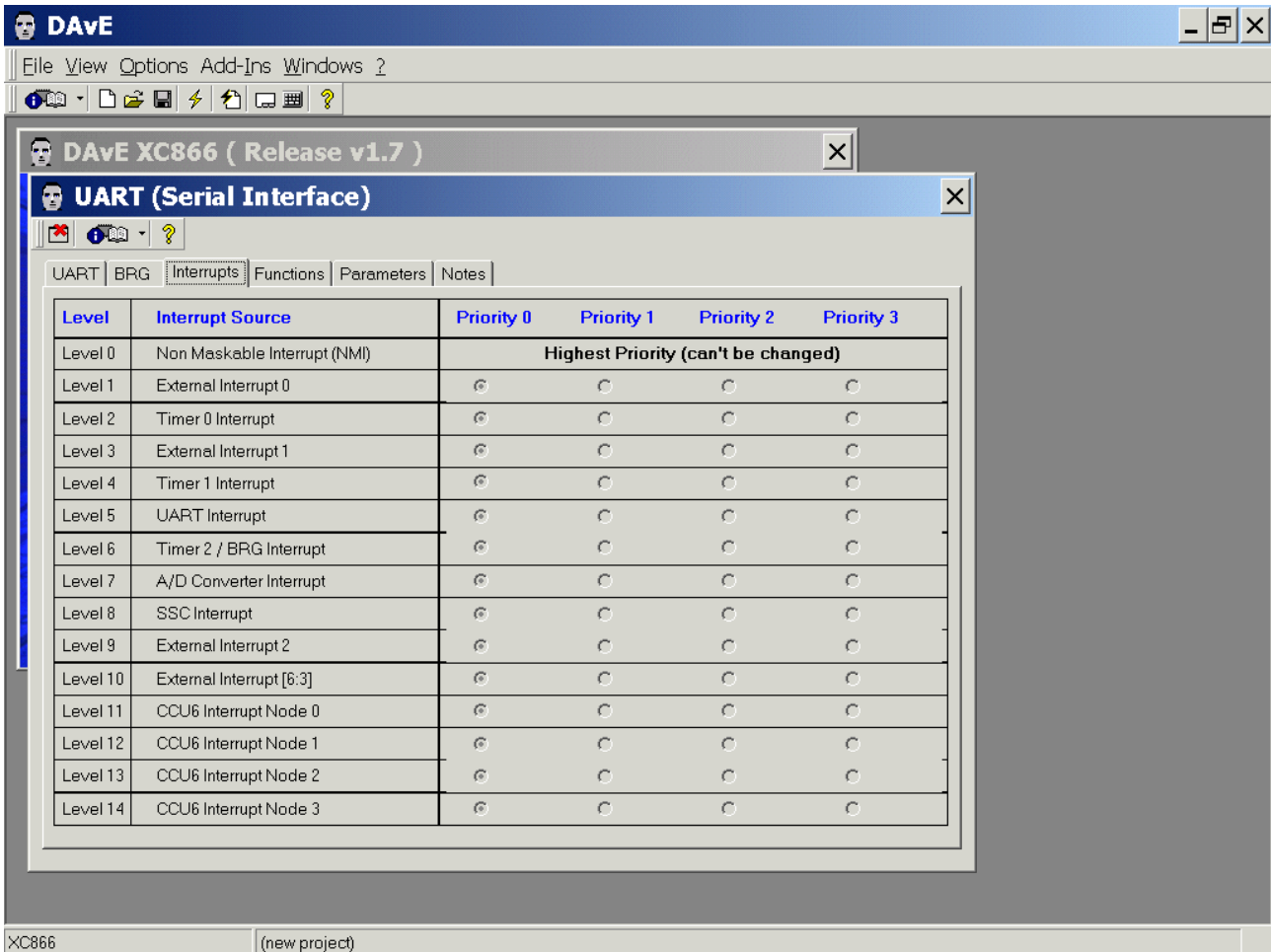
BRG: BRG Settings: Required baud rate [kbaud] **insert** 9,600



Note:

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

Interrupts: (do nothing)



DAvE XC866 (Release v1.7)

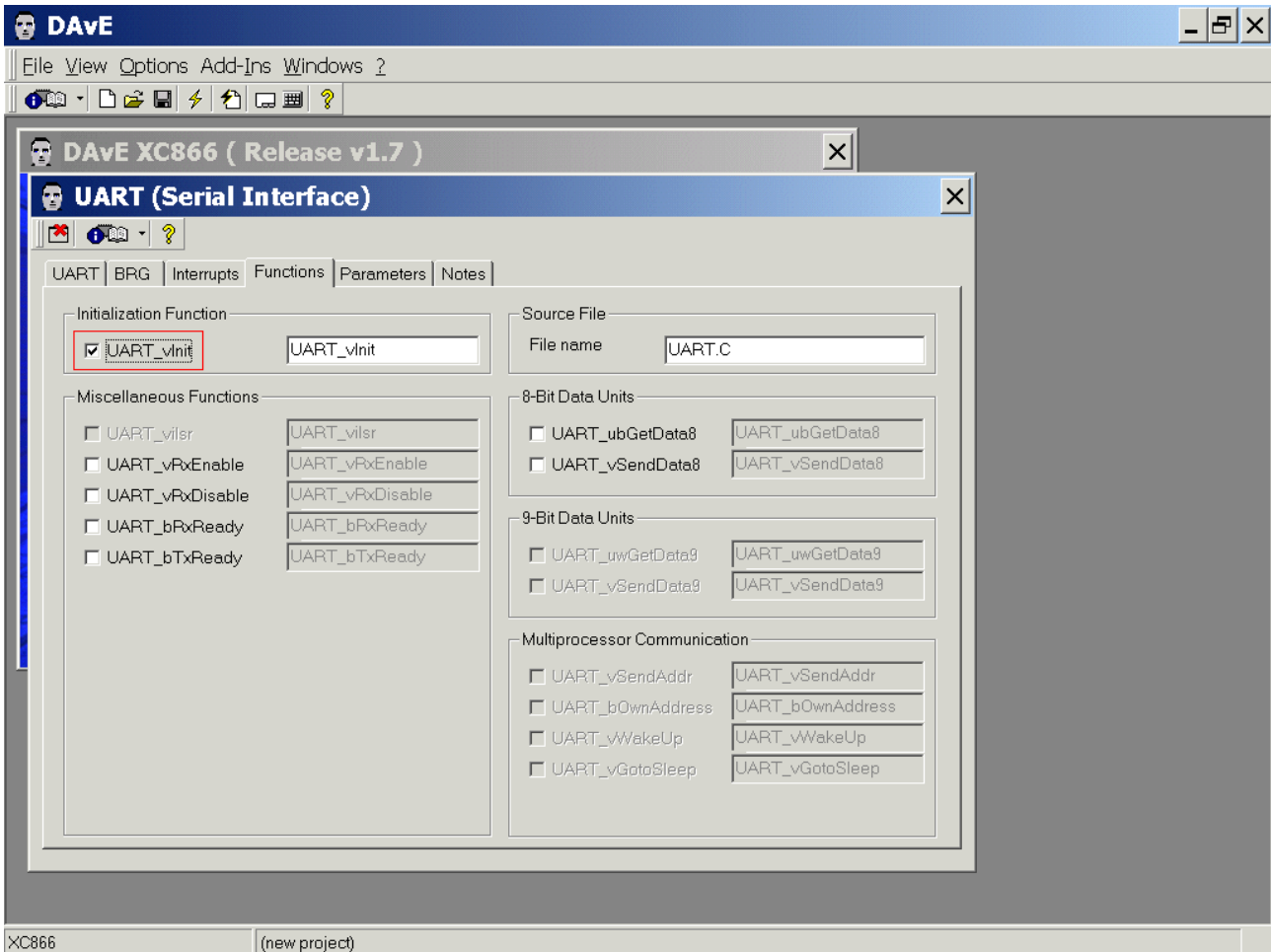
UART (Serial Interface)

UART | BRG | **Interrupts** | Functions | Parameters | Notes

Level	Interrupt Source	Priority 0	Priority 1	Priority 2	Priority 3
Level 0	Non Maskable Interrupt (NMI)	Highest Priority (can't be changed)			
Level 1	External Interrupt 0	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 2	Timer 0 Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 3	External Interrupt 1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 4	Timer 1 Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 5	UART Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 6	Timer 2 / BRG Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 7	A/D Converter Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 8	SSC Interrupt	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 9	External Interrupt 2	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 10	External Interrupt [6:3]	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 11	CCU6 Interrupt Node 0	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 12	CCU6 Interrupt Node 1	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 13	CCU6 Interrupt Node 2	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 14	CCU6 Interrupt Node 3	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

XC866 (new project)

Functions: Initialization Function: **click** UART_vInit



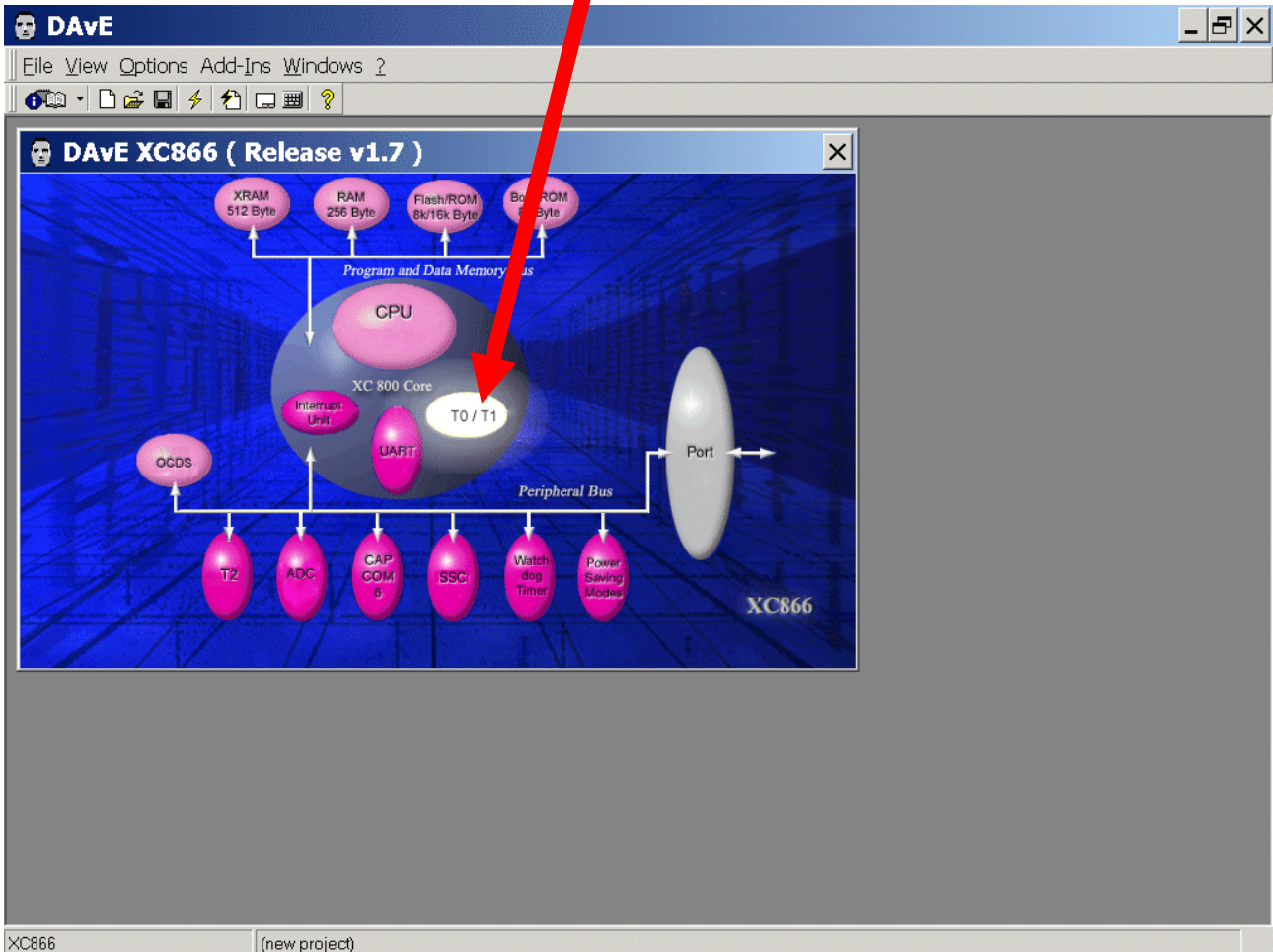
Parameters: (do nothing)

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

Exit this dialog now by clicking  the close button.

Configure Timer T0:

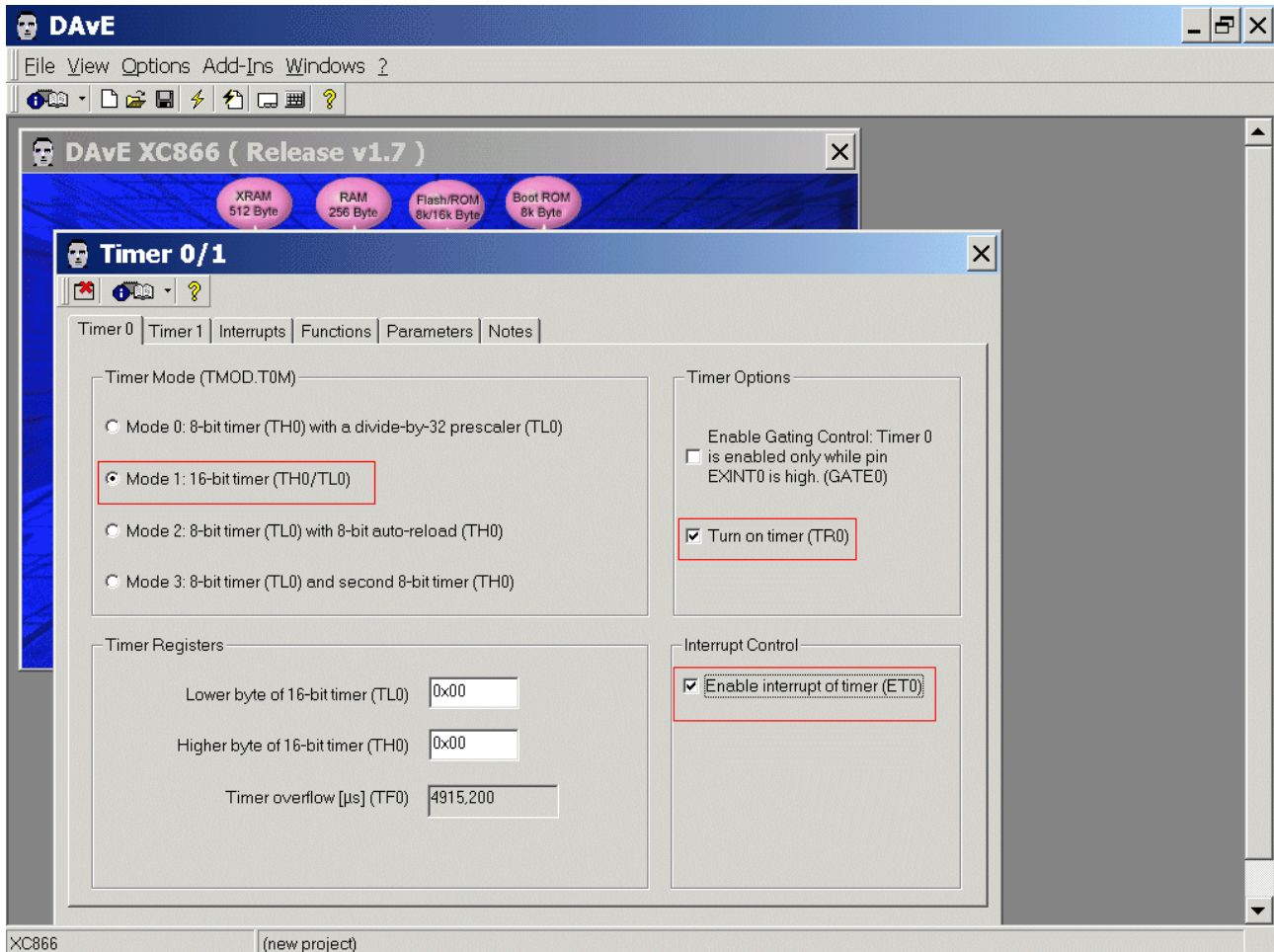
The configuration window can be opened by clicking the specific block/module.



Timer0: Timer Mode: **click** Mode 1: 16-bit timer

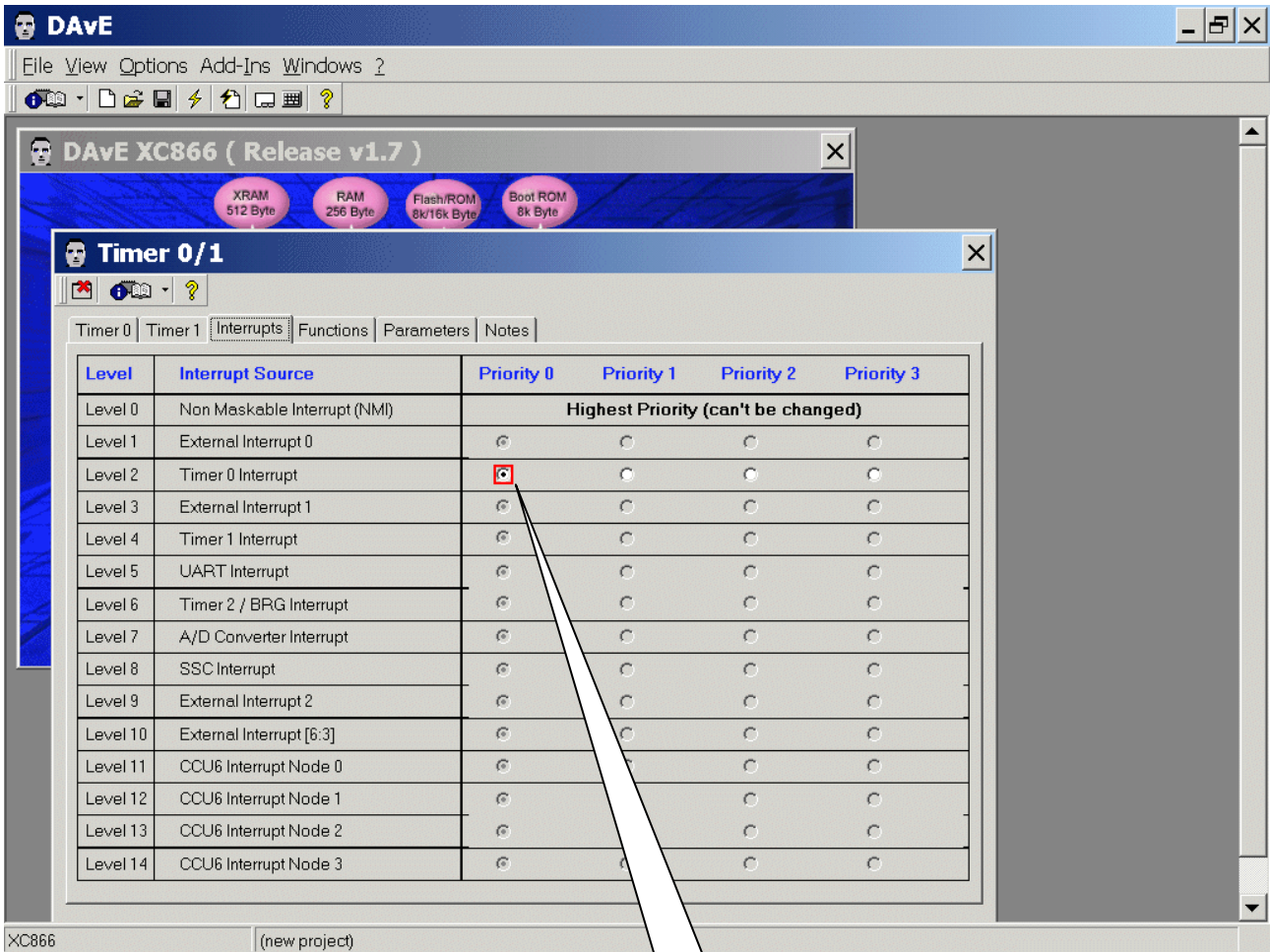
Timer0: Timer Options: **click** Turn on timer (TR0)

Timer0: Interrupt Control: **click** Enable interrupt of timer



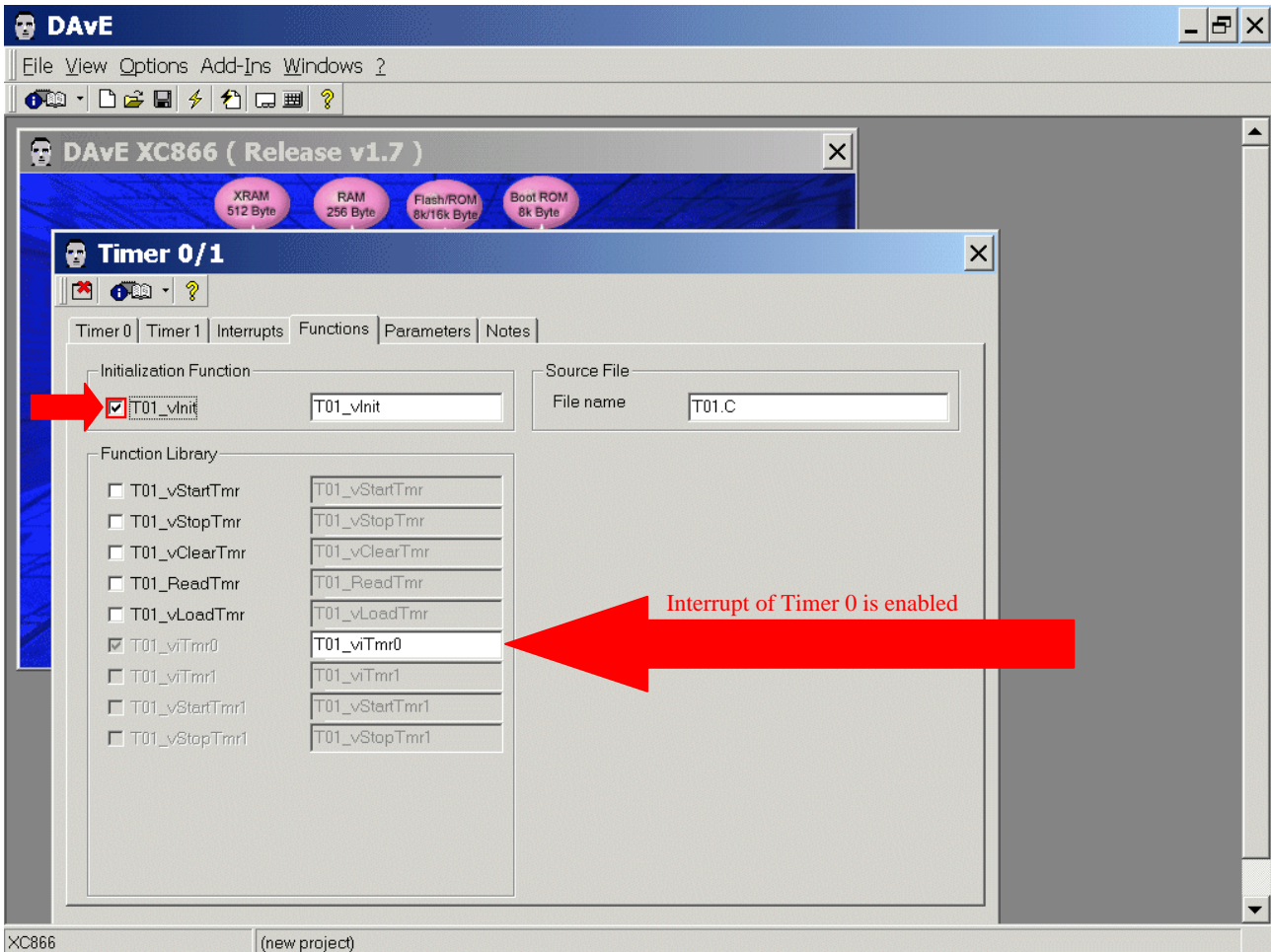
Timer1: do nothing

Interrupts: (do nothing)



Interrupt of Timer 0 is enabled

Functions: Initialization Function: **click** T01_vInit



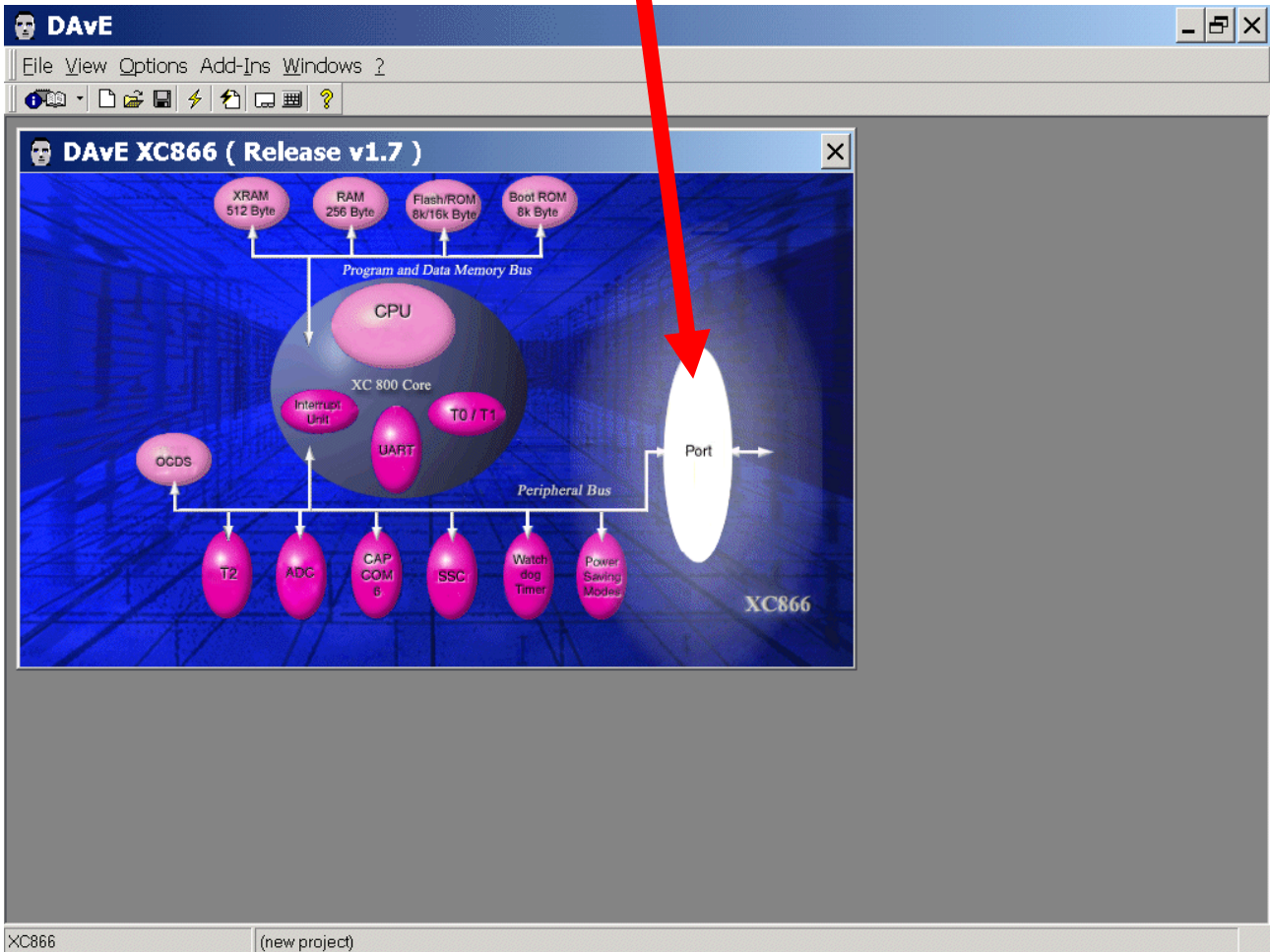
Parameters: (do nothing)

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

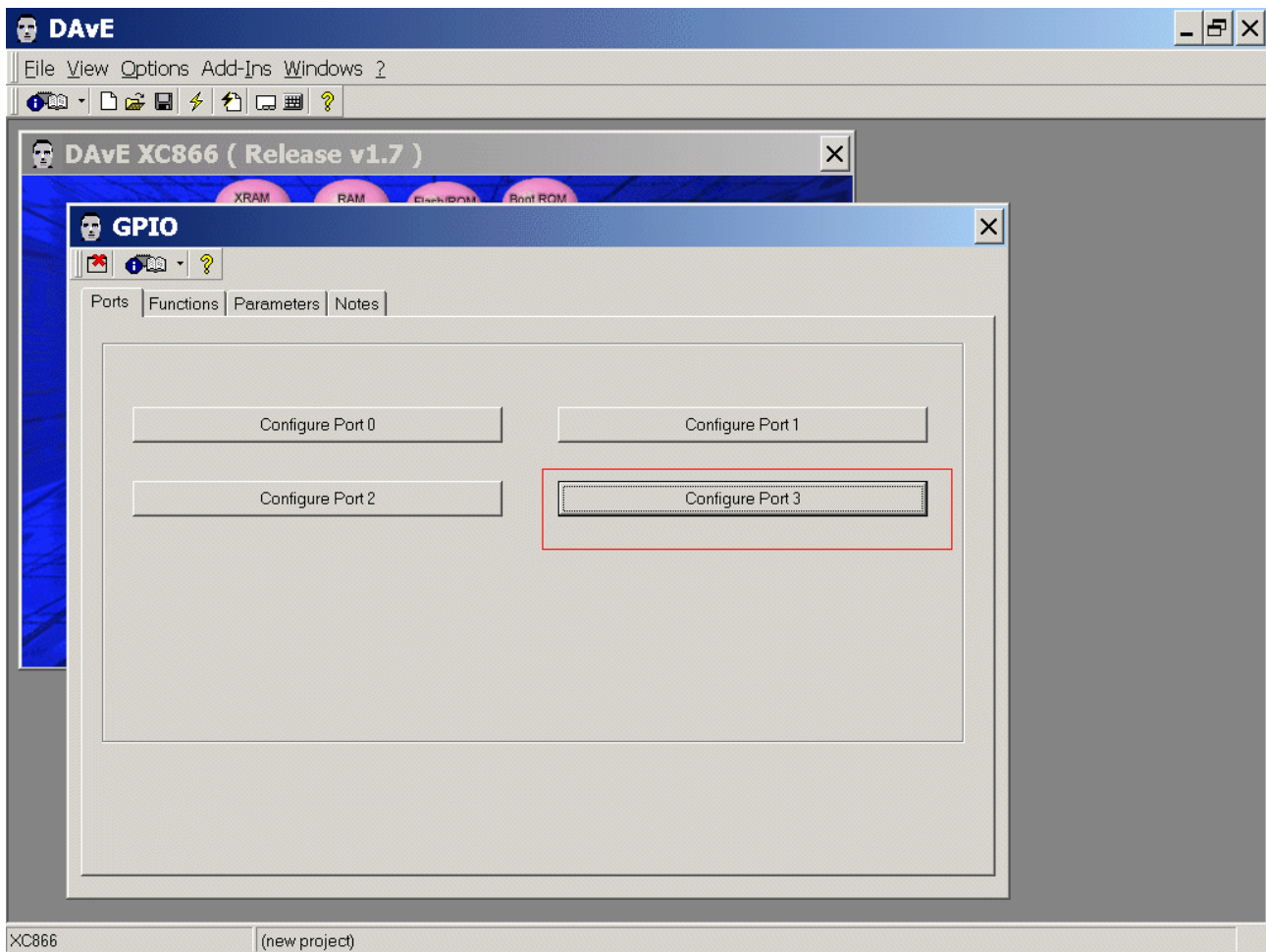
Exit this dialog now by clicking  the close button.

Configure Port 3 to Output:

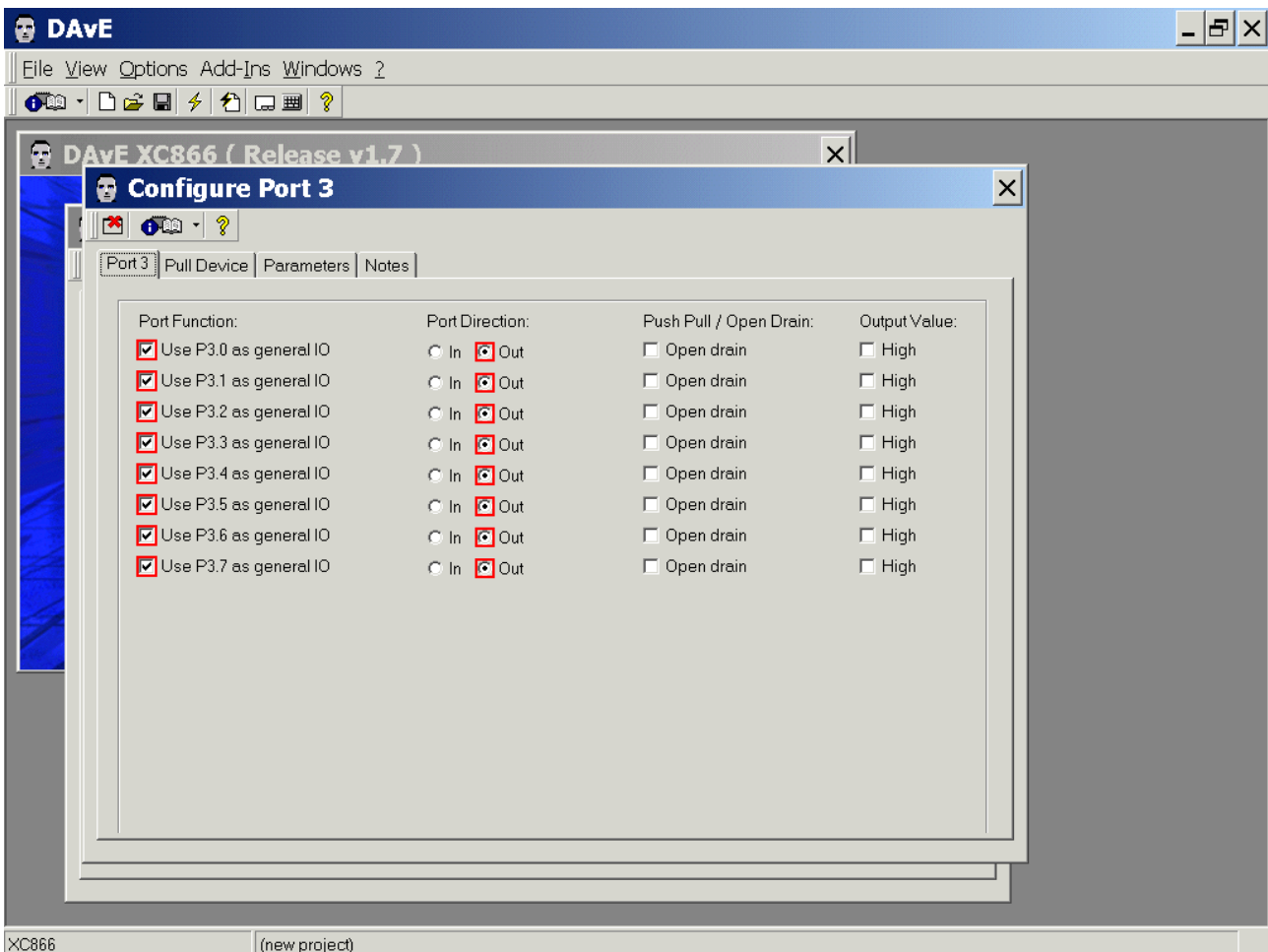
The configuration window can be opened by clicking the specific block/module.



Ports: click "Configure Port 3"



Port 3: Port Function: **click** ✓ Use P3.0 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.1 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.2 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.3 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.4 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.5 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.6 as general IO - Port Direction: **click** ⊙ Out
 Port 3: Port Function: **click** ✓ Use P3.7 as general IO - Port Direction: **click** ⊙ Out



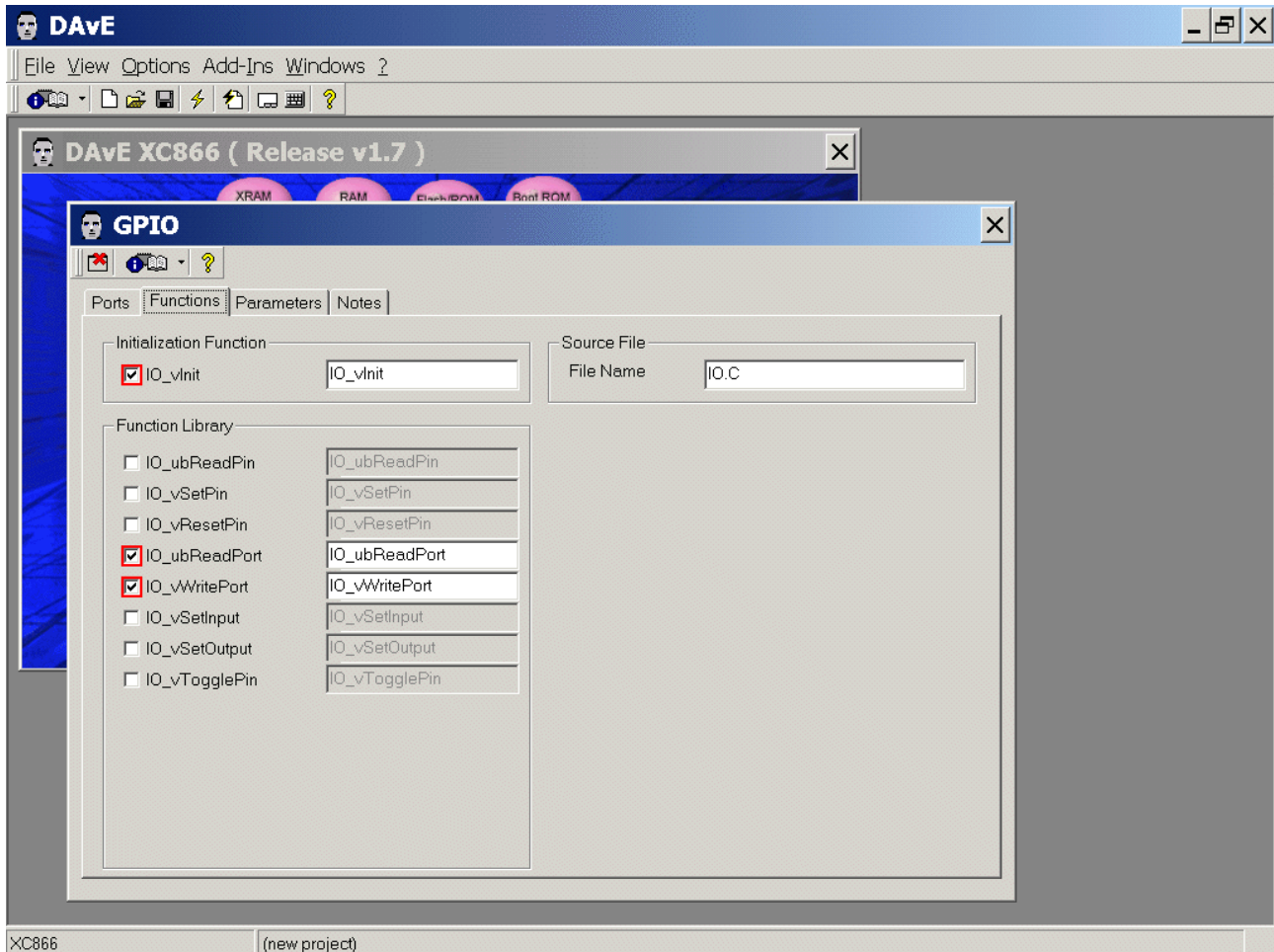
Pull Device: (do nothing)


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
 Functions: Function Library: **click** ✓ IO_ubReadPort
 Functions: Function Library: **click** ✓ IO_vWritePort

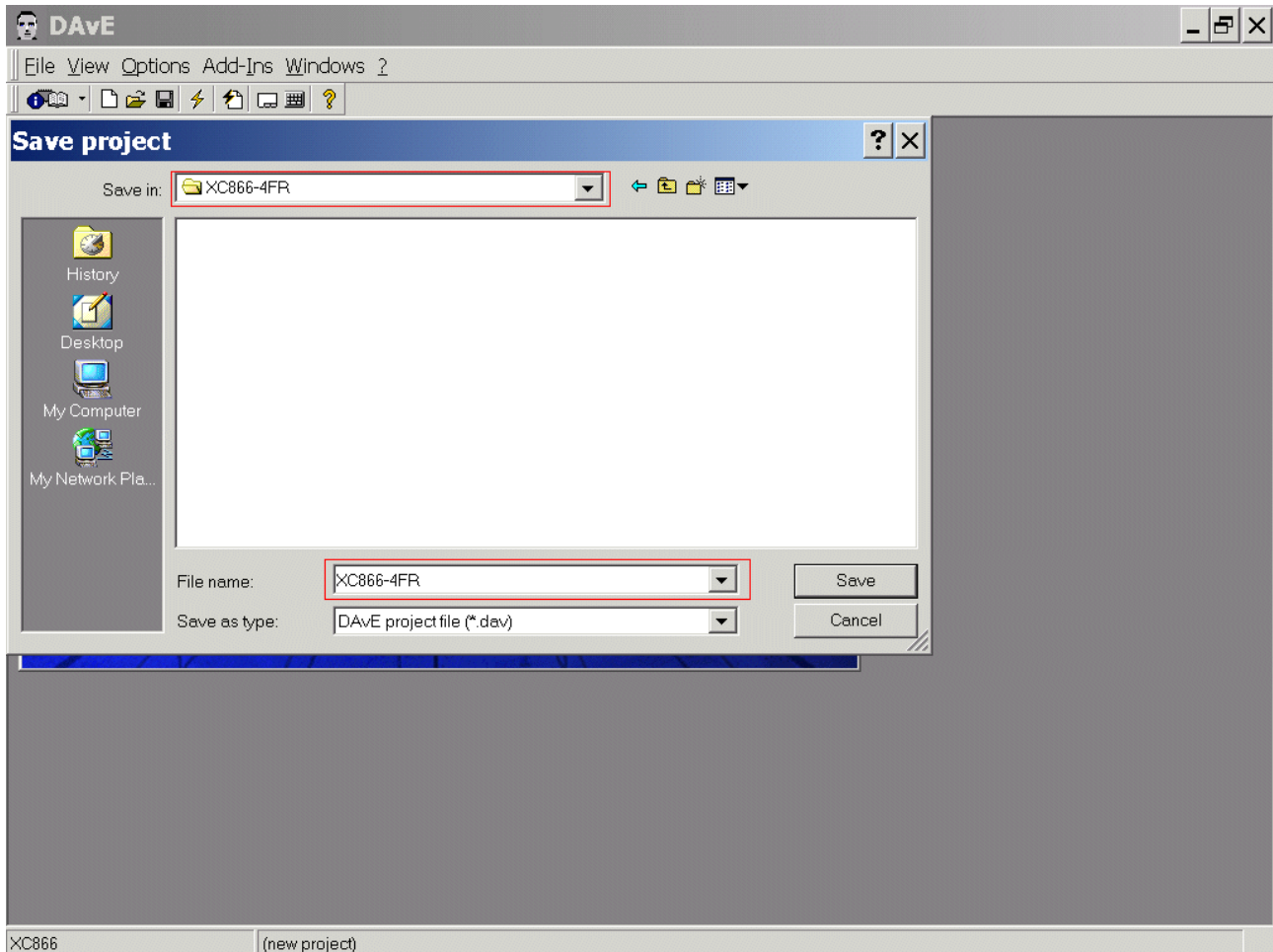


Parameters: (do nothing)
 Notes: If you wish, you can insert your comments here.
 Exit this dialog now by clicking  the close button.

Save the project:


File
Save

Save project: Save in C:\XC866-4FR (create directory)
File name: XC866-4FR



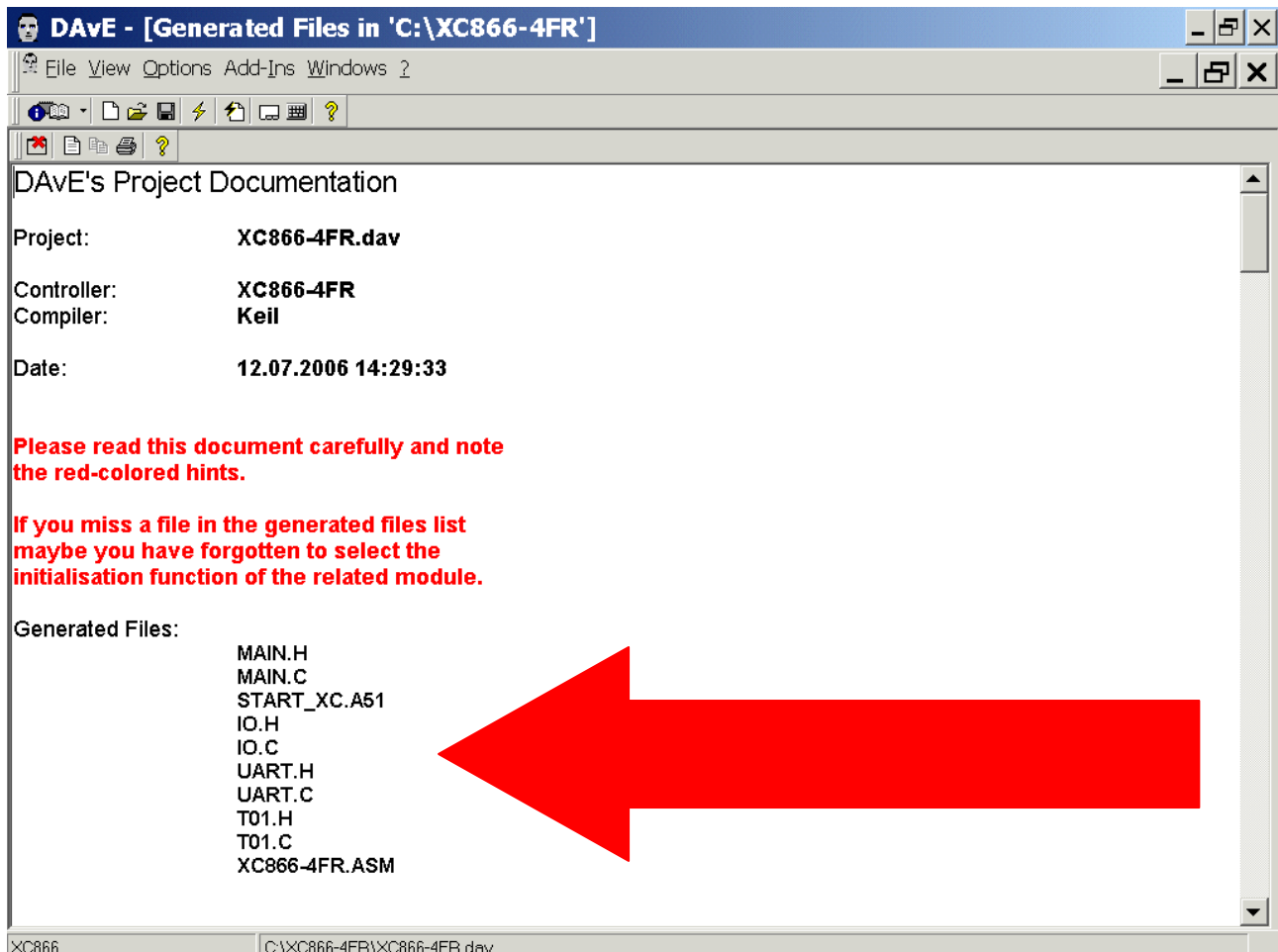
Save

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

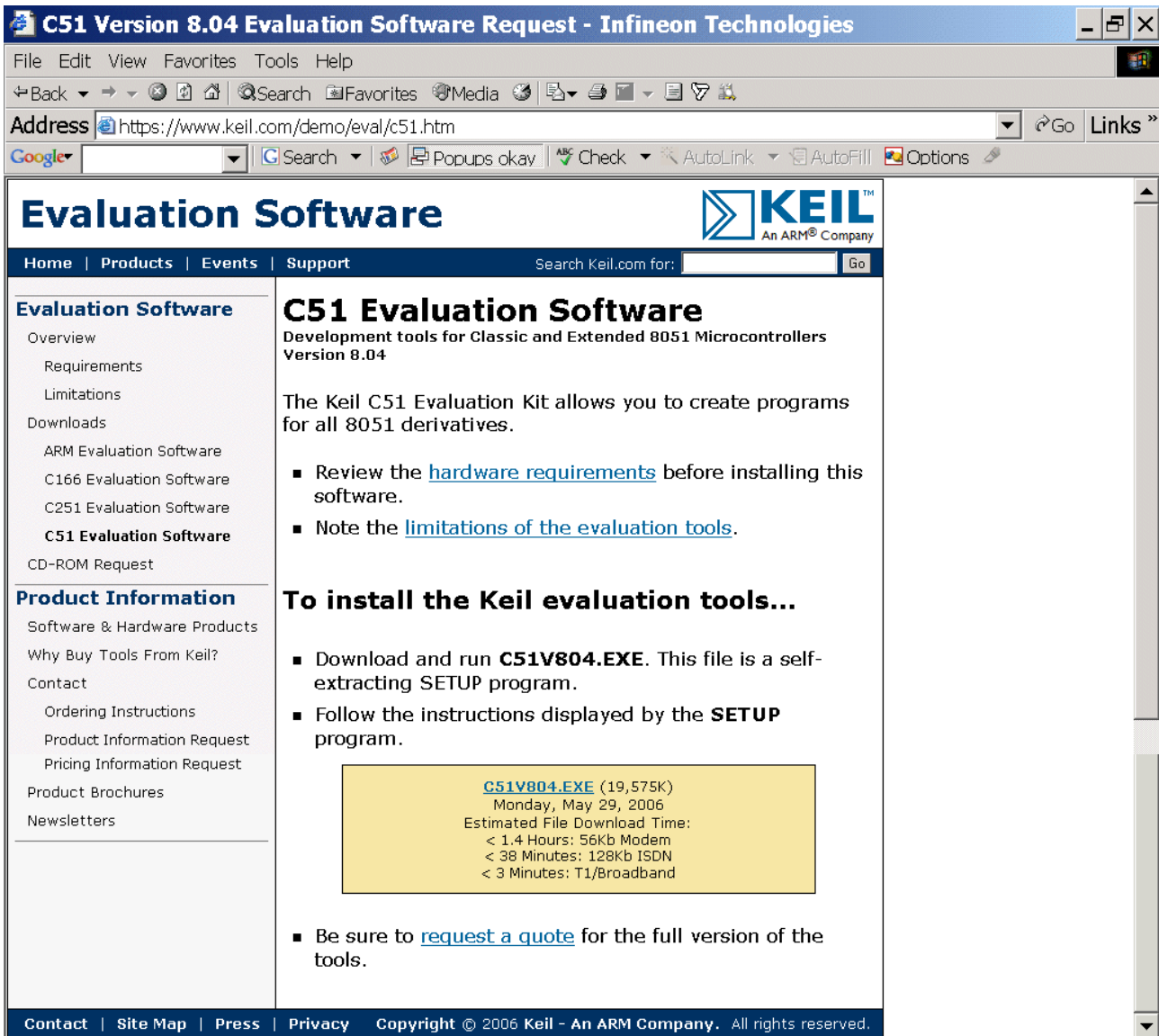
Save changes?

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/c51.htm>



C51 Version 8.04 Evaluation Software Request - Infineon Technologies

File Edit View Favorites Tools Help

Address <https://www.keil.com/demo/eval/c51.htm>

Evaluation Software  An ARM[®] Company

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

Evaluation Software

- Overview
- Requirements
- Limitations
- Downloads
 - ARM Evaluation Software
 - C166 Evaluation Software
 - C251 Evaluation Software
 - C51 Evaluation Software**
 - CD-ROM Request

Product Information

- Software & Hardware Products
- Why Buy Tools From Keil?
- Contact
 - Ordering Instructions
 - Product Information Request
 - Pricing Information Request
- Product Brochures
- Newsletters

C51 Evaluation Software
Development tools for Classic and Extended 8051 Microcontrollers
Version 8.04

The Keil C51 Evaluation Kit allows you to create programs for all 8051 derivatives.

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

To install the Keil evaluation tools...

- Download and run **C51V804.EXE**. This file is a self-extracting SETUP program.
- Follow the instructions displayed by the **SETUP** program.

[C51V804.EXE](#) (19,575K)
Monday, May 29, 2006
Estimated File Download Time:
< 1.4 Hours: 56Kb Modem
< 38 Minutes: 128Kb ISDN
< 3 Minutes: T1/Broadband

- 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 **C51V804.EXE** (- or higher version)



Start Keil μ Vision3 and open the DAVe Project:

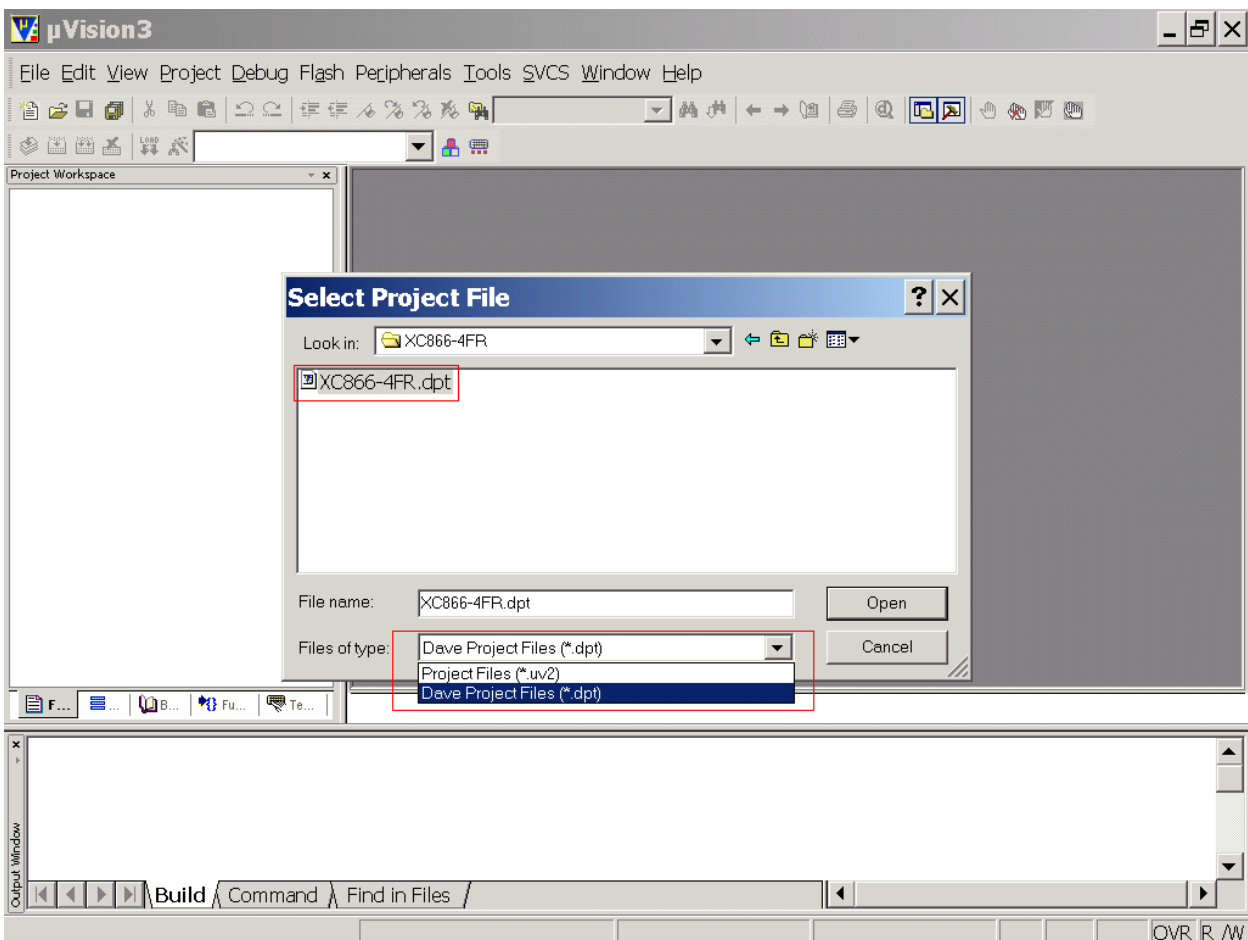
If you see an open project – close it: **Project - Close Project**

Project - Open Project

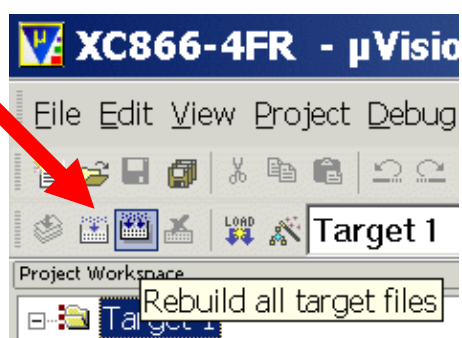
Select Project File: **Look in:** choose C:\XC866-4FR

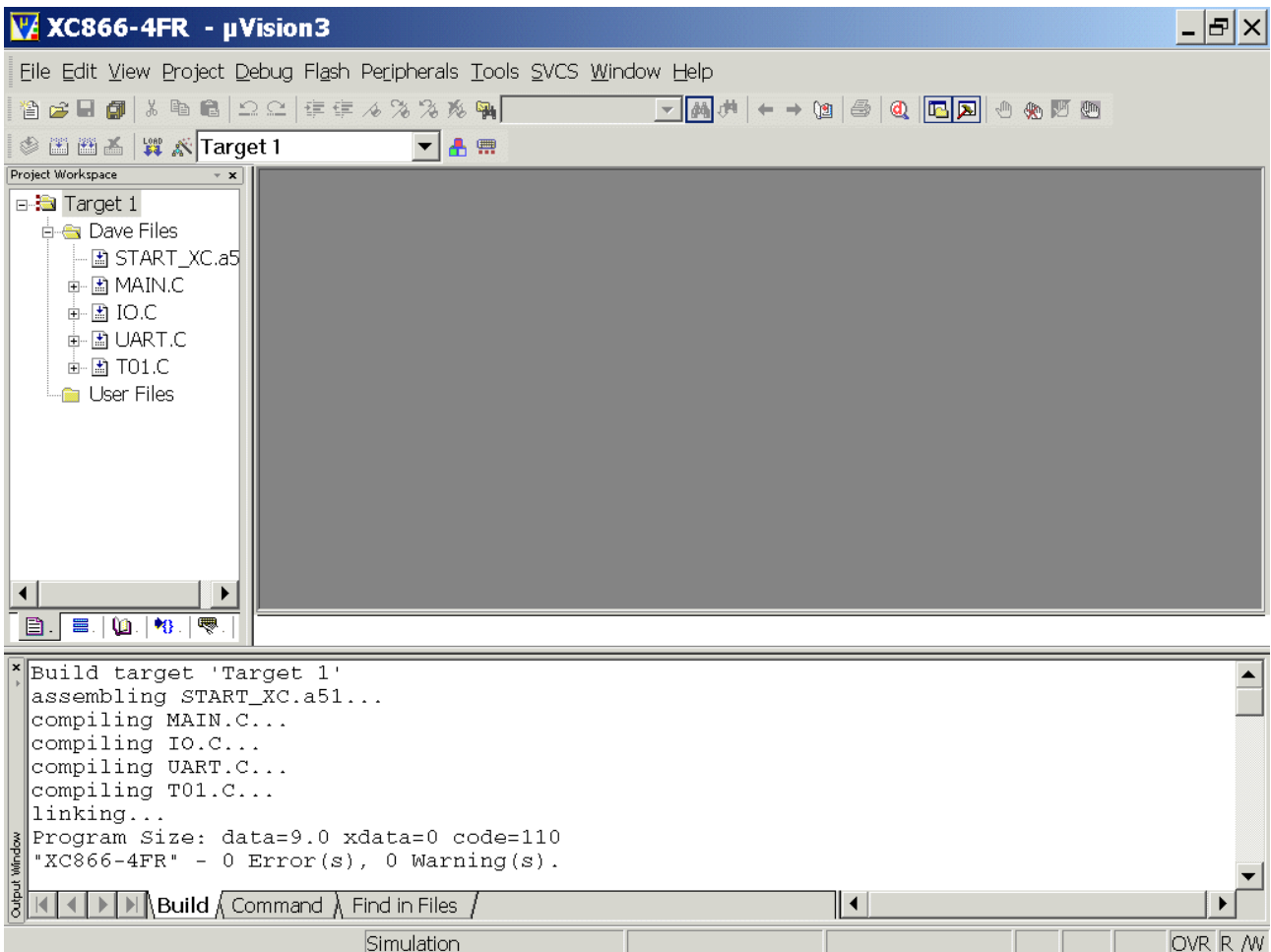
Select Project File: **Files of type:** choose Dave Project Files

Choose XC866-4FR.dpt

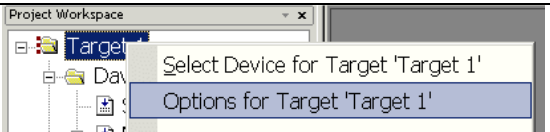
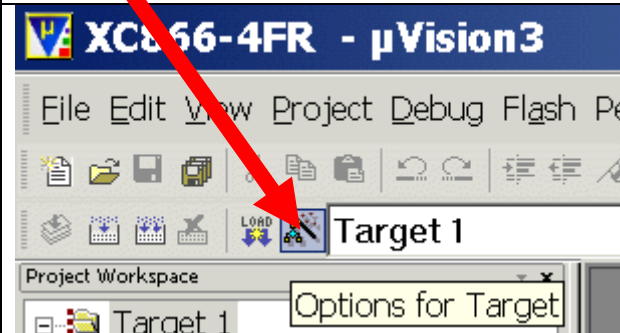


Open

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



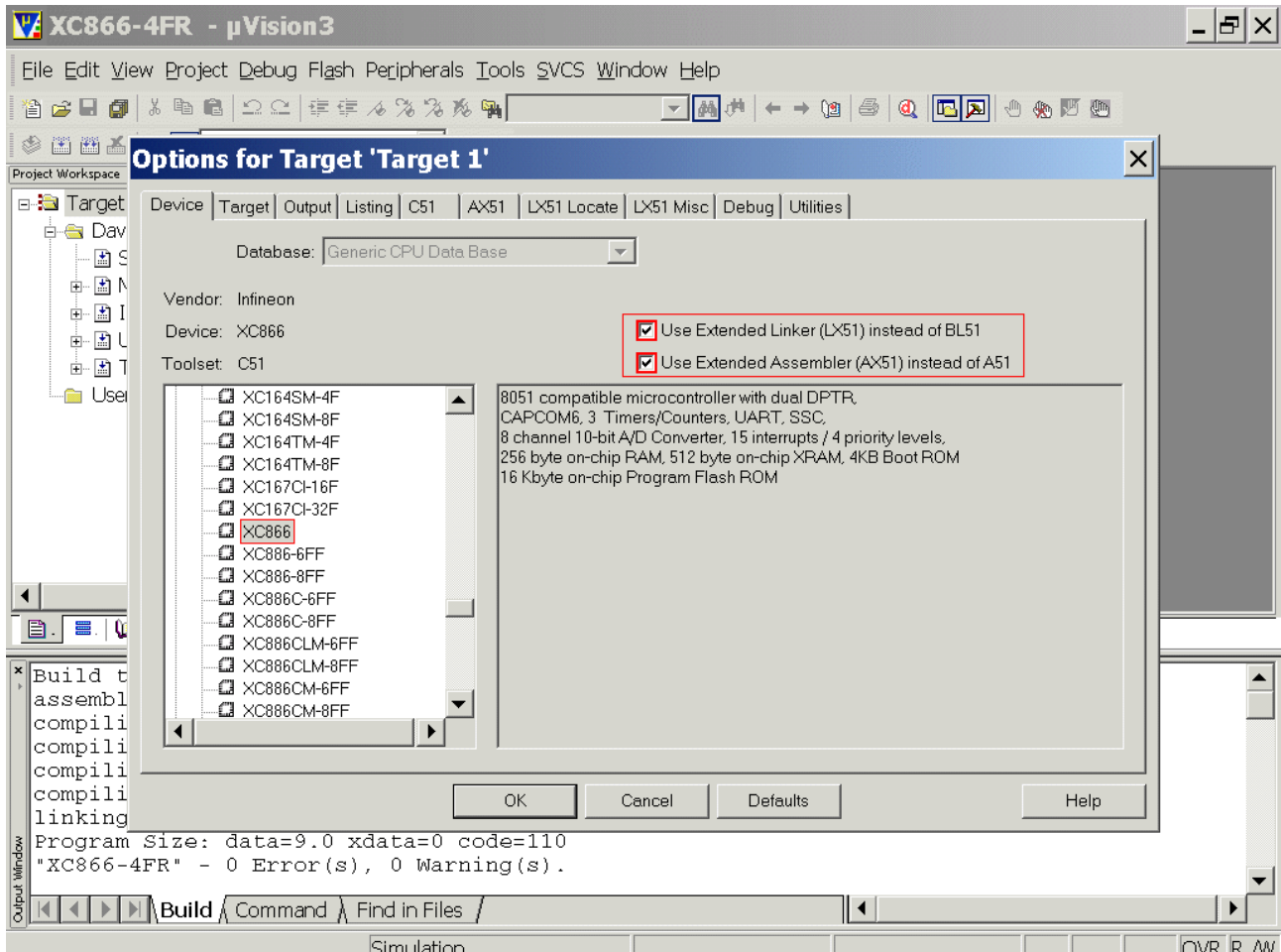
Configure Compiler, Assembler, Linker, Locator, Hex-Converter and Build – Control:

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

Options for Target 'Target 1': Device: check XC866

Options for Target 'Target 1': Device: click ✓ Use Extended Linker (LX51)

Options for Target 'Target 1': Device: click ✓ Use Extended Assembler (AX51)



Options for Target 'Target 1'

Database: Generic CPU Data Base

Vendor: Infineon

Device: XC866

Toolset: C51

Use Extended Linker (LX51) instead of BL51

Use Extended Assembler (AX51) instead of A51

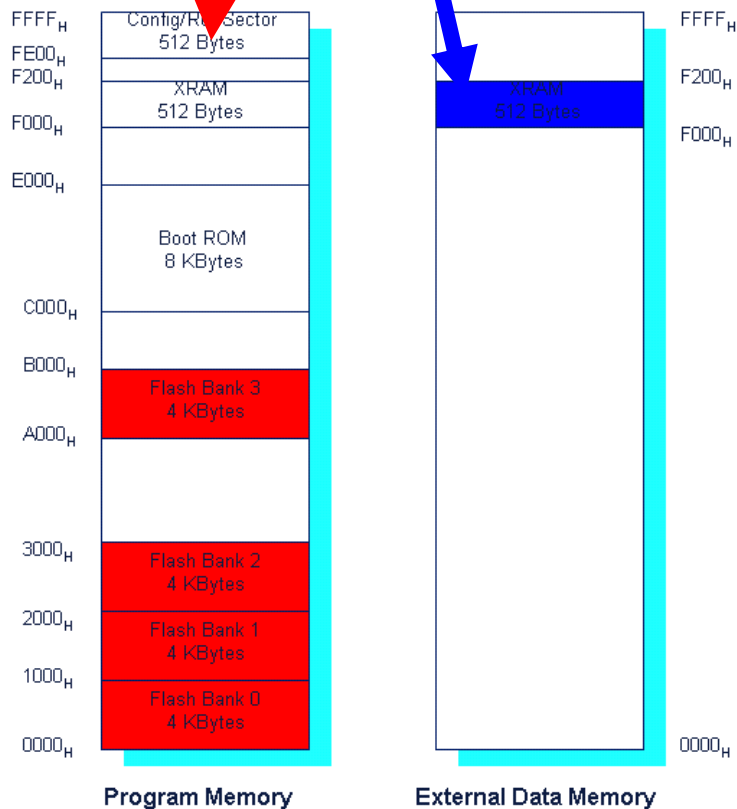
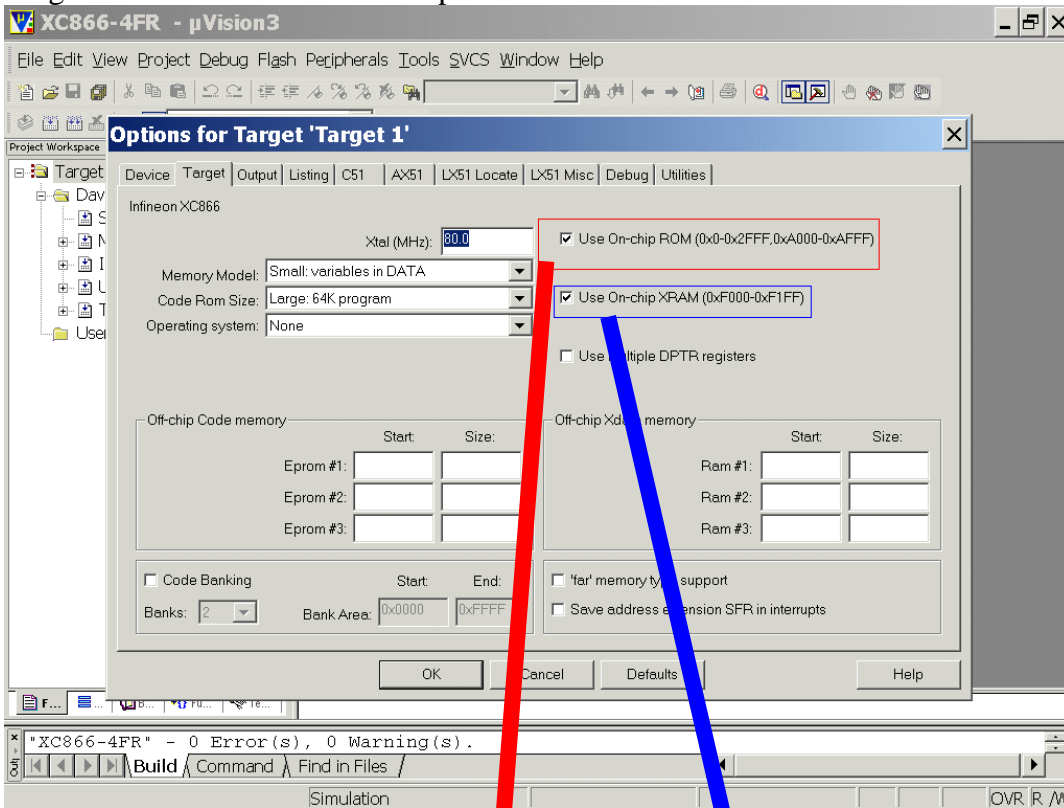
XC866

8051 compatible microcontroller with dual DPTR, CAPCOM6, 3 Timers/Counters, UART, SSC, 8 channel 10-bit A/D Converter, 15 interrupts / 4 priority levels, 256 byte on-chip RAM, 512 byte on-chip XRAM, 4KB Boot ROM, 16 Kbyte on-chip Program Flash ROM

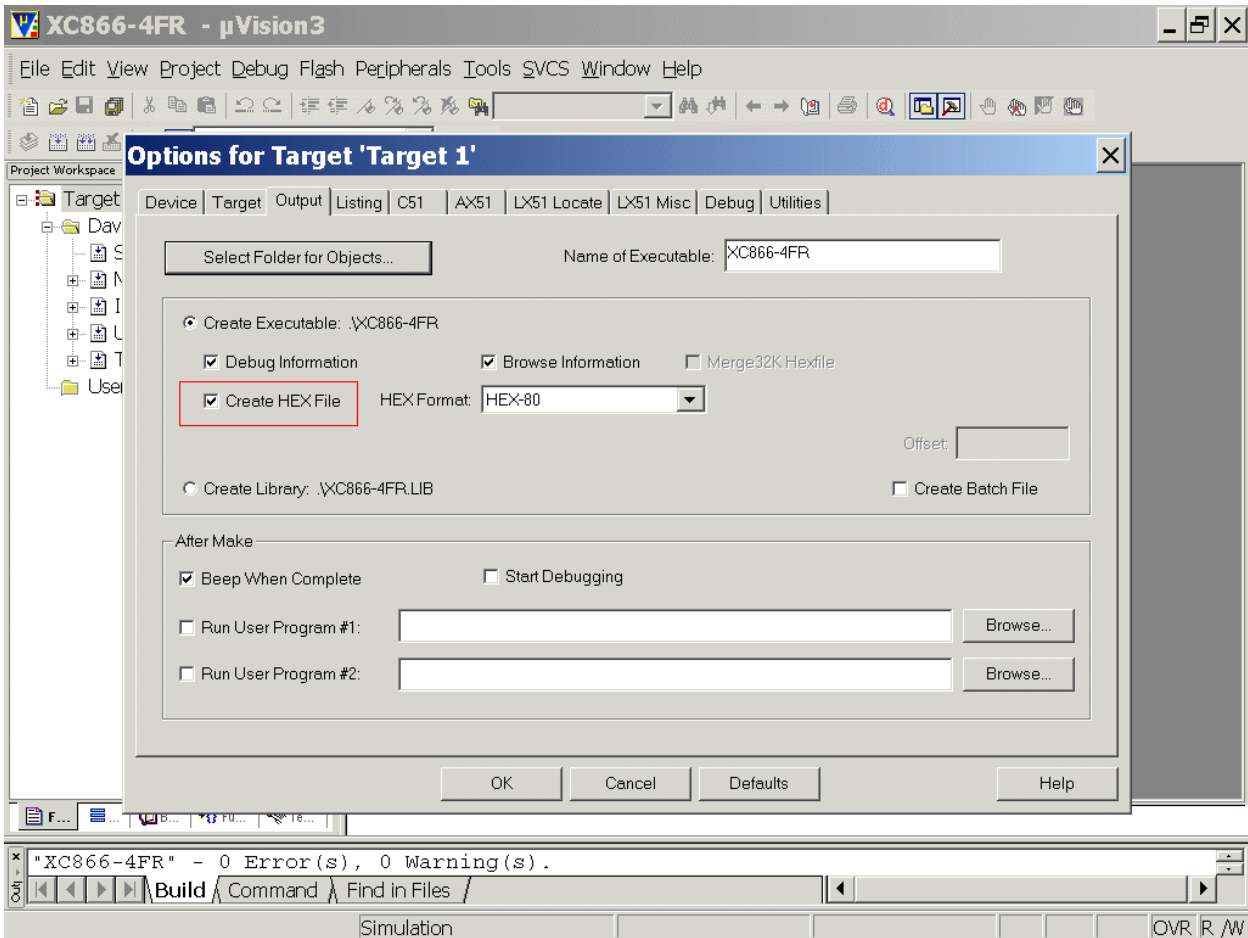
OK Cancel Defaults Help

Program Size: data=9.0 xdata=0 code=110
"XC866-4FR" - 0 Error(s), 0 Warning(s).

Target: **click/check** ✓ Use On-chip ROM
 Target: **click/check** ✓ Use On-chip XRAM



Output: **click** ✓ Create HEX File



Listing: (do nothing)

C51: (do nothing)

AX51: (do nothing)

LX51 Locate: (do nothing)

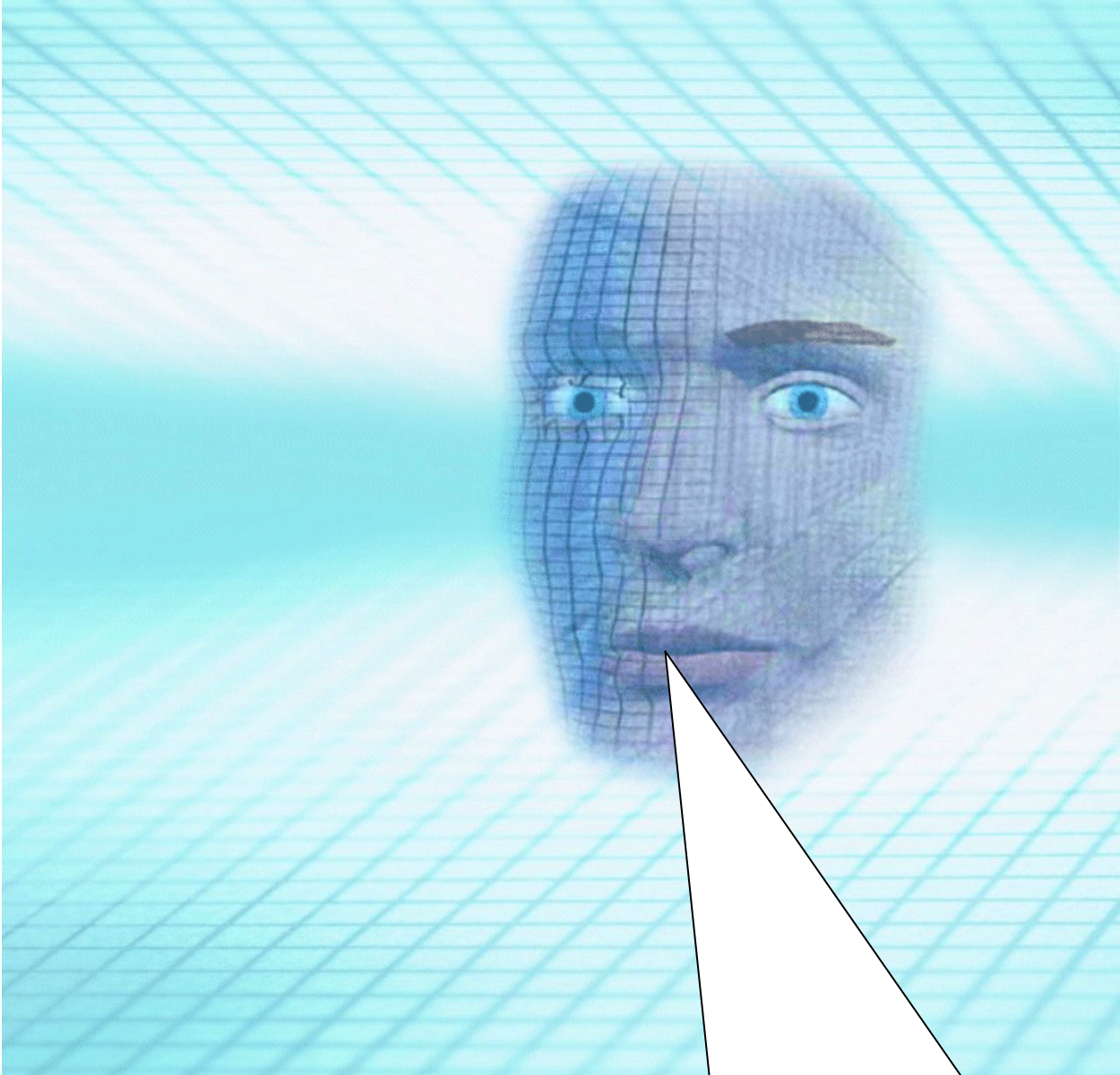
LX51 Misc: (do nothing)

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:

```
code char menu[] =
"\r\n\r\n\r\n"
"1 ... LEDs P3 ON\r\n"
"2 ... LEDs P3 OFF\r\n"
"3 ... LEDs P3 blinking\r\n"
" \r\n";

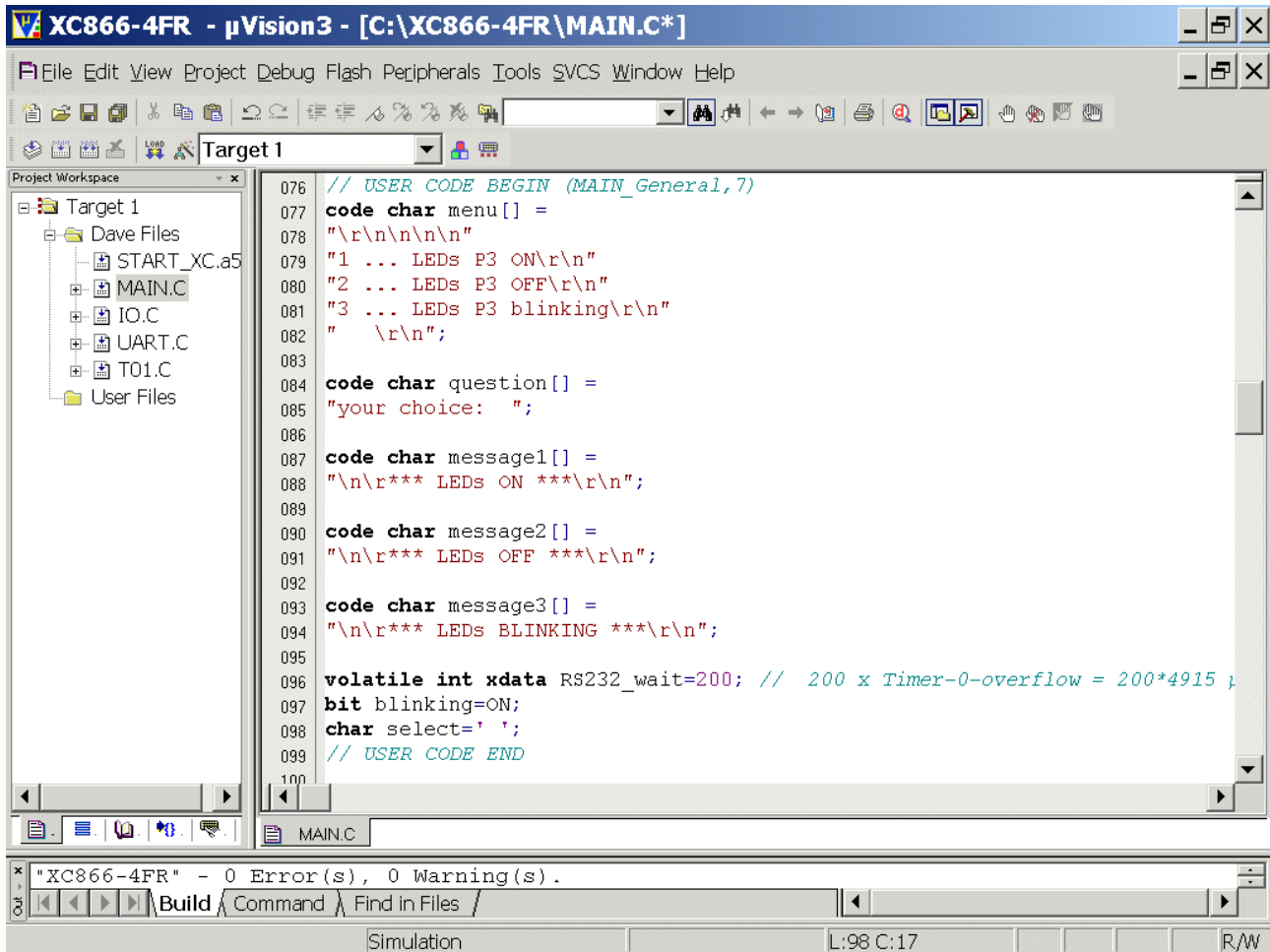
code char question[] =
"your choice: ";

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

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

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

volatile int xdata RS232_wait=200; // 200 x Timer-0-overflow = 200*4915 µs = 0,98 s
bit blinking=ON;
char select=' ';
```



The screenshot shows the Infineon IDE interface for the XC866-4FR project. The main window displays the source code for MAIN.C, which includes menu definitions, messages, and hardware control logic. The status bar at the bottom indicates 0 errors and 0 warnings, and the simulation is running at L:98 C:17.

```

076 // USER CODE BEGIN (MAIN_General,7)
077 code char menu[] =
078 "\r\n\r\n\r\n"
079 "1 ... LEDs P3 ON\r\n"
080 "2 ... LEDs P3 OFF\r\n"
081 "3 ... LEDs P3 blinking\r\n"
082 " \r\n";
083
084 code char question[] =
085 "your choice: ";
086
087 code char message1[] =
088 "\n\r*** LEDs ON ***\r\n";
089
090 code char message2[] =
091 "\n\r*** LEDs OFF ***\r\n";
092
093 code char message3[] =
094 "\n\r*** LEDs BLINKING ***\r\n";
095
096 volatile int xdata RS232_wait=200; // 200 x Timer-0-overflow = 200*4915 ;
097 bit blinking=ON;
098 char select=' ';
099 // USER CODE END
100

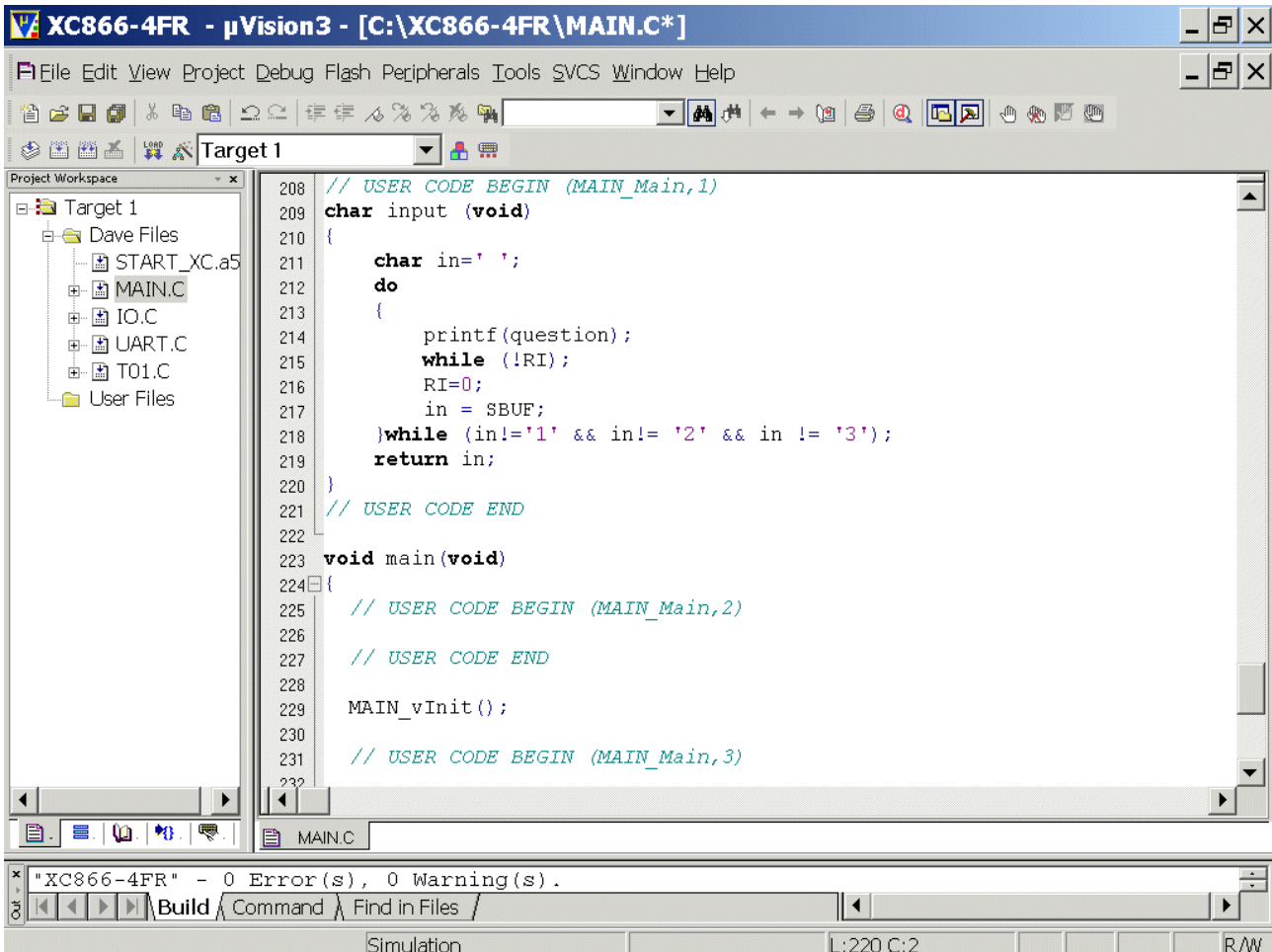
```

Project Workspace: Target 1
 Dave Files
 START_XC.a5
 MAIN.C
 IO.C
 UART.C
 T01.C
 User Files

Status: "XC866-4FR" - 0 Error(s), 0 Warning(s).
 Build Command: Find in Files /
 Simulation | L:98 C:17 | R/W

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

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



The screenshot shows the µVision3 IDE interface. The title bar reads "XC866-4FR - µVision3 - [C:\XC866-4FR\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_XC.a5", "MAIN.C", "IO.C", "UART.C", "T01.C", and "User Files". The main editor window displays the following code:

```
208 // USER CODE BEGIN (MAIN_Main,1)
209 char input (void)
210 {
211     char in=' ';
212     do
213     {
214         printf(question);
215         while (!RI);
216         RI=0;
217         in = SBUF;
218     }while (in!='1' && in!= '2' && in != '3');
219     return in;
220 }
221 // USER CODE END
222
223 void main(void)
224 {
225     // USER CODE BEGIN (MAIN_Main,2)
226
227     // USER CODE END
228
229     MAIN_vInit();
230
231     // USER CODE BEGIN (MAIN_Main,3)
232
233 }
```

The status bar at the bottom indicates "Simulation" and "L:220 C:2".

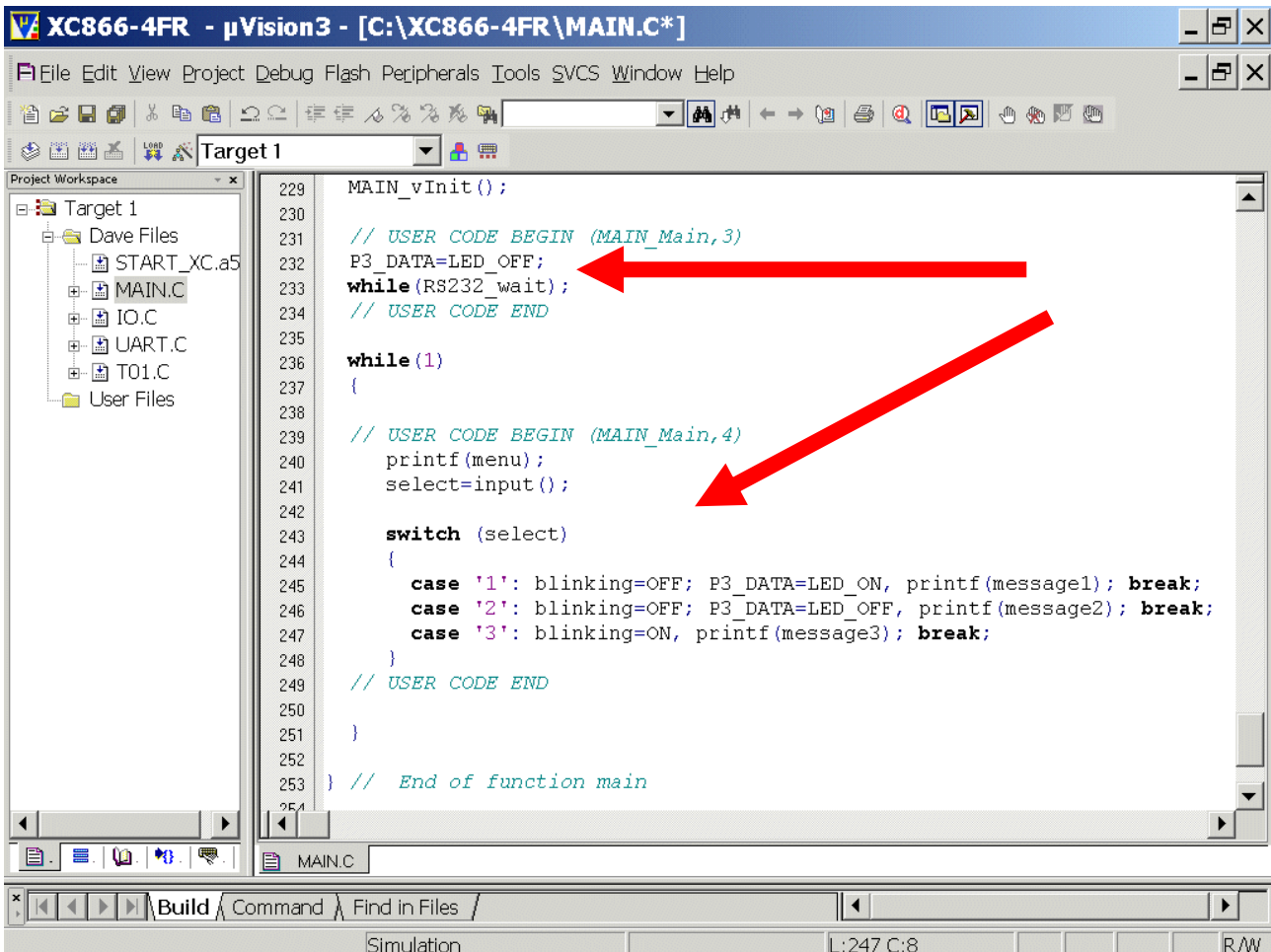
Double click **MAIN.C** and **insert** the following code in the **main** function:

```
P3_DATA=LED_OFF;
while(RS232_wait);
```

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

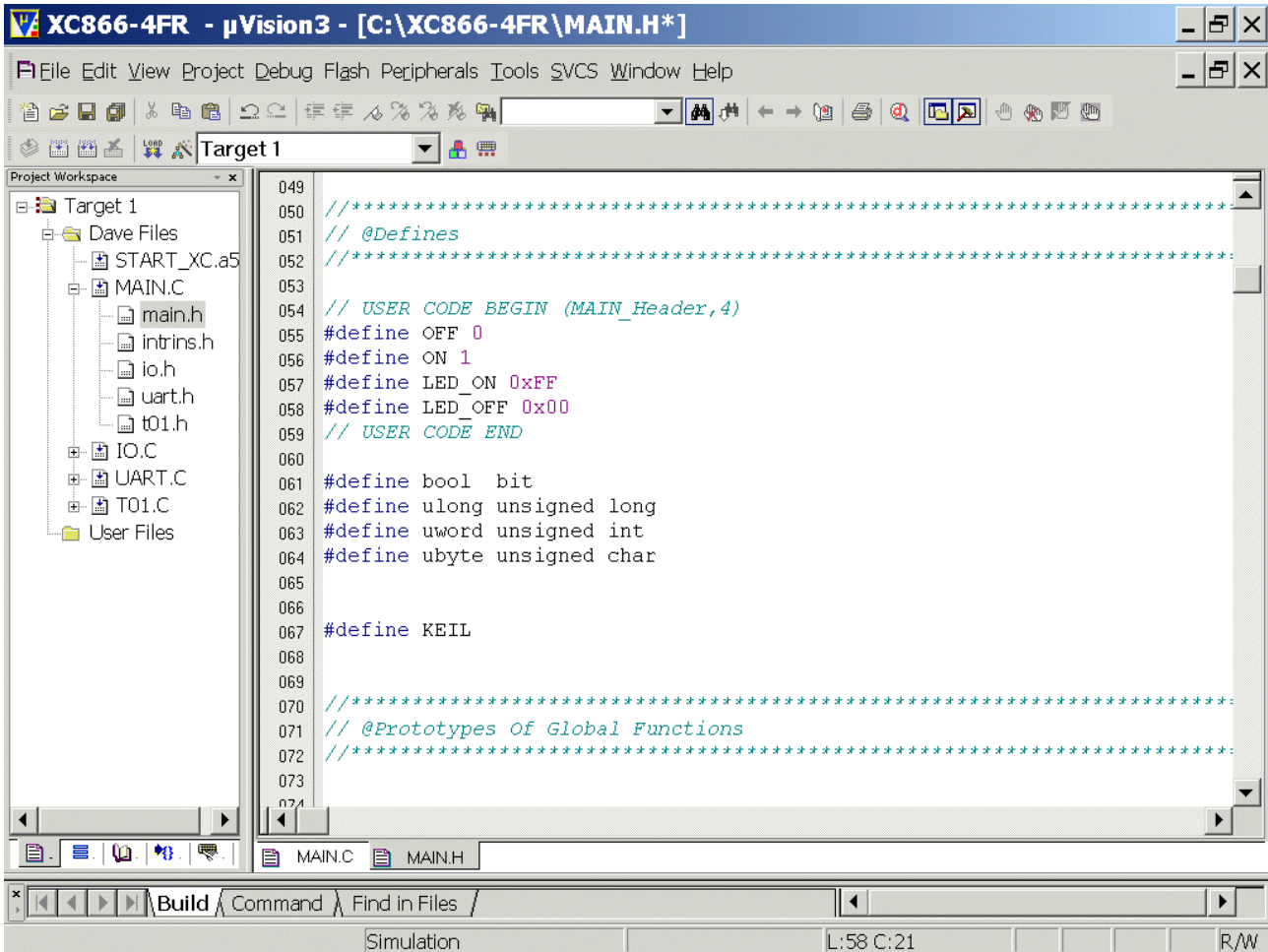
```
printf(menu);
select=input();

switch (select)
{
    case '1': blinking=OFF; P3_DATA=LED_ON, printf(message1); break;
    case '2': blinking=OFF; P3_DATA=LED_OFF, printf(message2); break;
    case '3': blinking=ON, printf(message3); break;
}
```



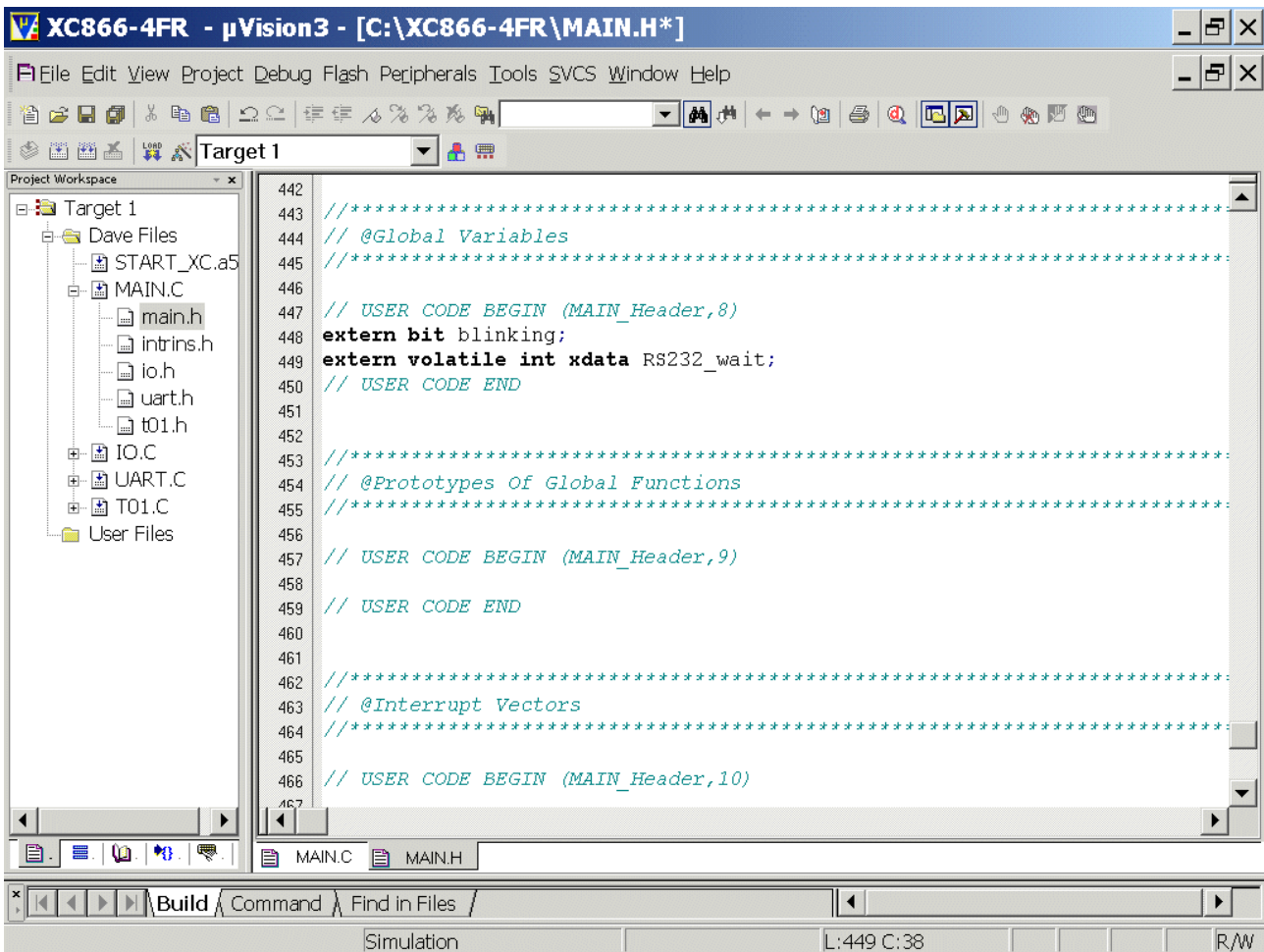
Double click **Main.h** and insert the following Defines:

```
#define OFF 0
#define ON 1
#define LED_ON 0xFF
#define LED_OFF 0x00
```



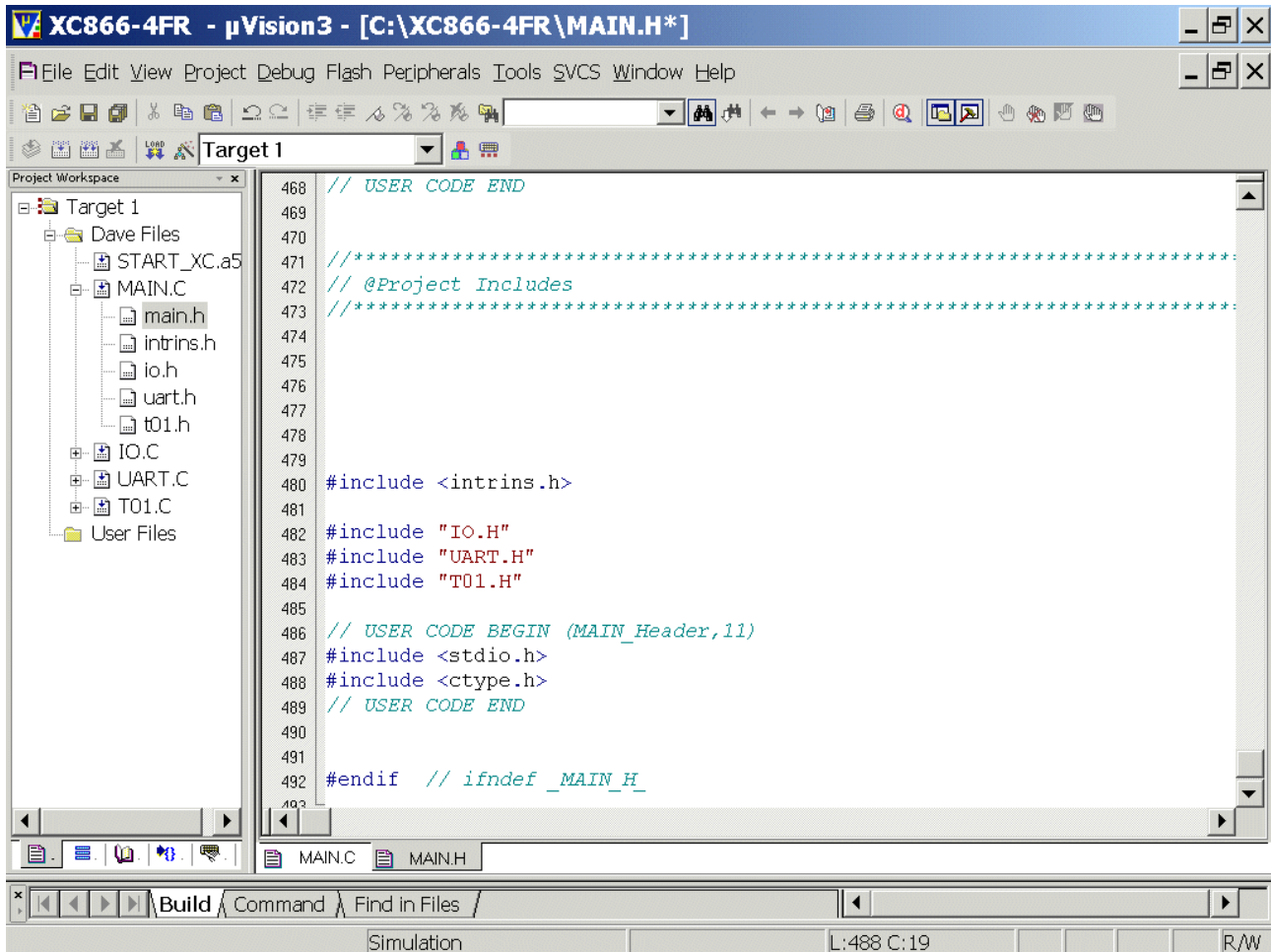
Double click **Main.h** and insert extern-declaration "Global Variables":

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



Double click **Main.h** and insert include files:

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



The screenshot shows the µVision3 IDE interface. The title bar reads "XC866-4FR - µVision3 - [C:\XC866-4FR\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" expanded, containing "Dave Files", "START_XC.a5", "MAIN.C", "main.h", "intrins.h", "io.h", "uart.h", "t01.h", "IO.C", "UART.C", "T01.C", and "User Files". The main editor window displays the content of MAIN.H, with line numbers 468 through 493. The code includes project-specific headers and the standard C headers requested in the text above.

```

468 // USER CODE END
469
470
471 //*****
472 // @Project Includes
473 //*****
474
475
476
477
478
479
480 #include <intrins.h>
481
482 #include "IO.H"
483 #include "UART.H"
484 #include "T01.H"
485
486 // USER CODE BEGIN (MAIN_Header,11)
487 #include <stdio.h>
488 #include <ctype.h>
489 // USER CODE END
490
491
492 #endif // ifndef _MAIN_H
493

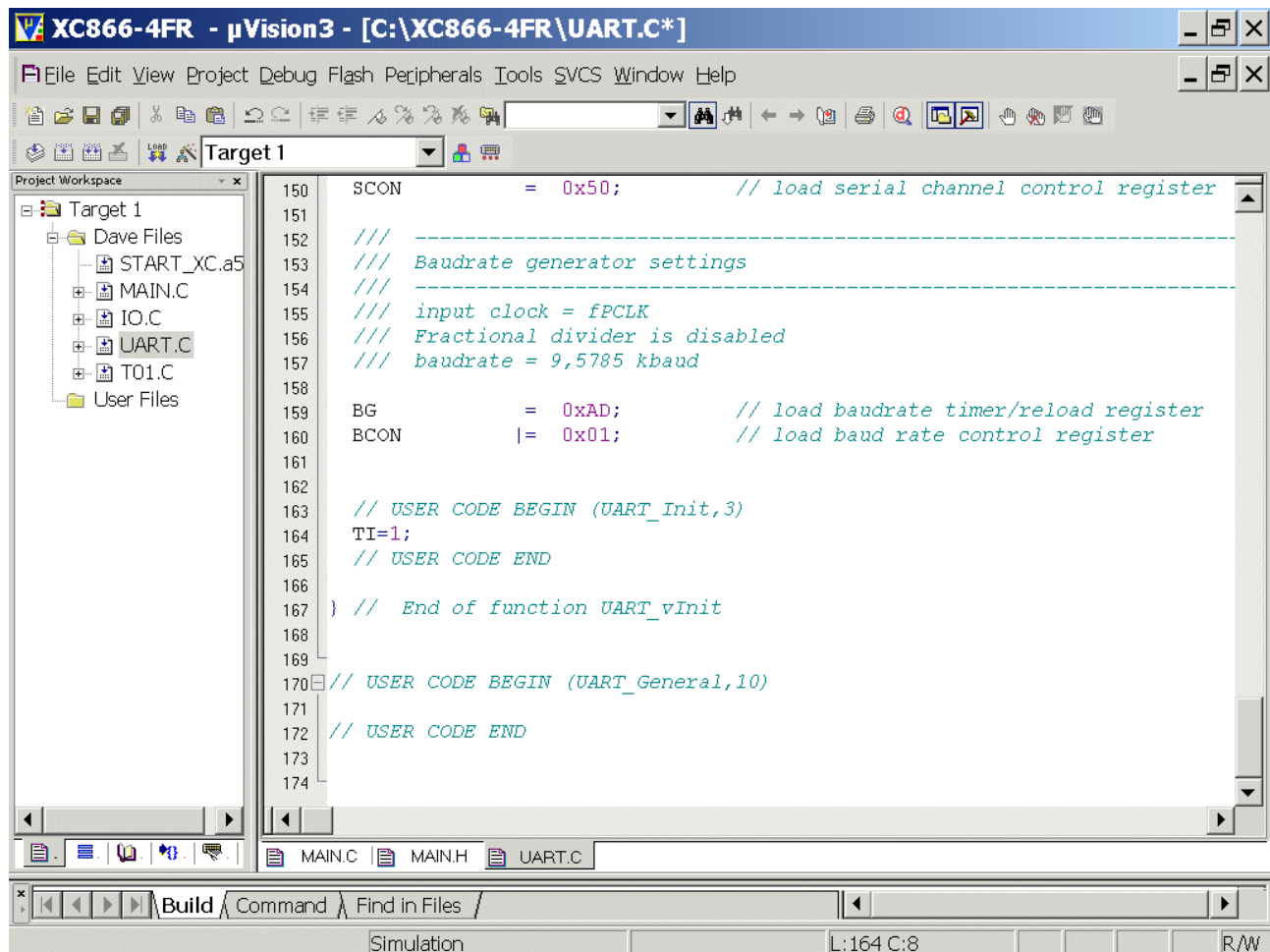
```

The status bar at the bottom shows "Simulation", "L:488 C:19", and "R/W".

Double click **UART.C**

Insert in the UART_vInit function: (to start printf()):

```
TI=1;
```



The screenshot shows the µVision3 IDE interface. The main window displays the source code for UART.C. The code includes initialization for the serial channel control register (SCON), baudrate generator settings, and baudrate timer/reload register (BG). A user-defined code block is inserted between lines 163 and 165, containing the line `TI=1;`. The IDE also shows a project workspace on the left with files like MAIN.C, IO.C, UART.C, and T01.C. The status bar at the bottom indicates the current position is at line 164, column 8.

```

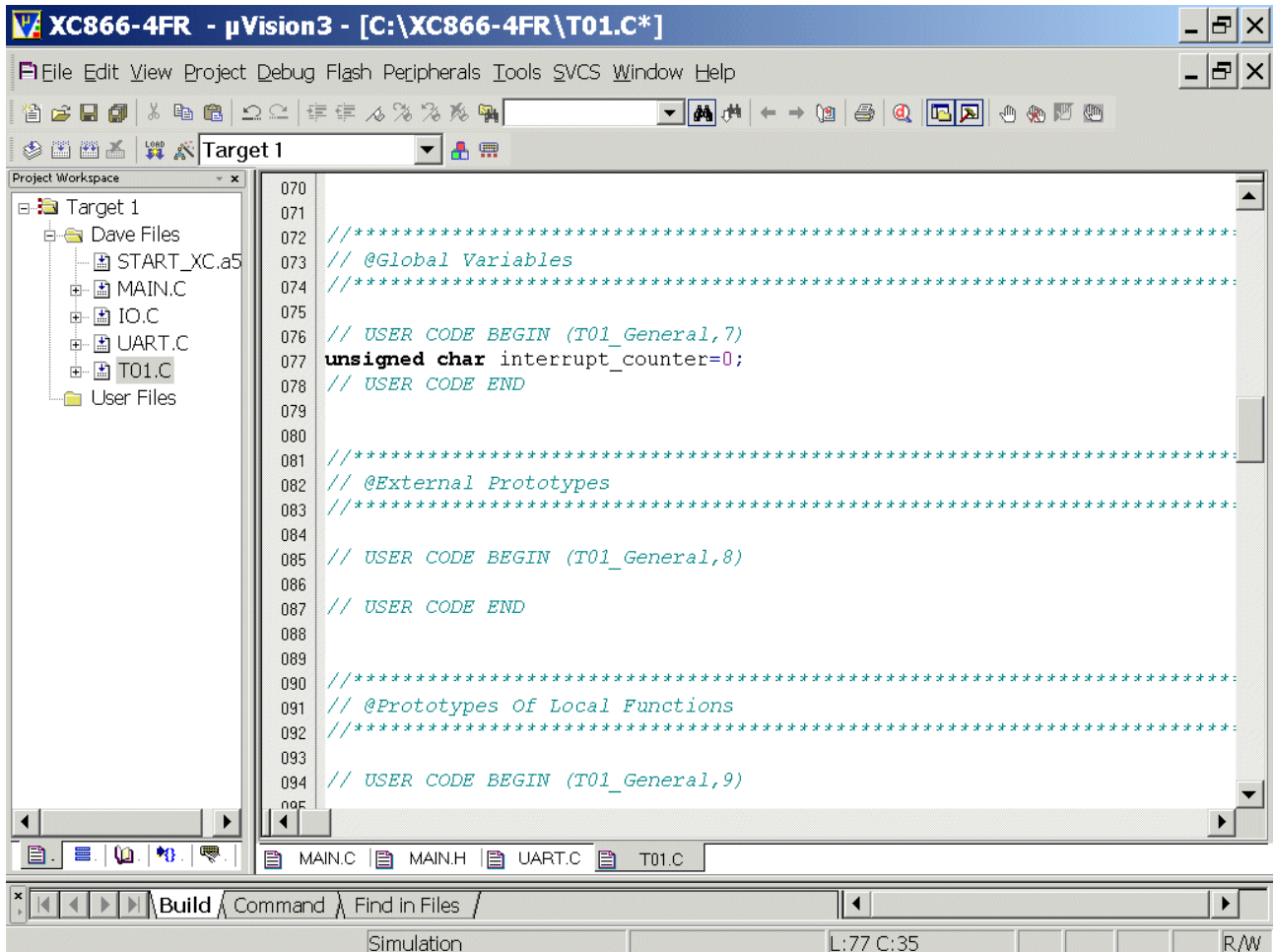
150  SCON          = 0x50;           // load serial channel control register
151
152  /// -----
153  /// Baudrate generator settings
154  /// -----
155  /// input clock = fPCLK
156  /// Fractional divider is disabled
157  /// baudrate = 9,5785 kbaud
158
159  BG            = 0xAD;           // load baudrate timer/reload register
160  BCON          |= 0x01;         // load baud rate control register
161
162
163  // USER CODE BEGIN (UART_Init,3)
164  TI=1;
165  // USER CODE END
166
167 } // End of function UART_vInit
168
169
170 // USER CODE BEGIN (UART_General,10)
171
172 // USER CODE END
173
174

```

Double click T01.C

Insert the following global variable:

```
unsigned char interrupt_counter=0;
```



Double click T01.C

Insert code for T0 interrupt service routine:

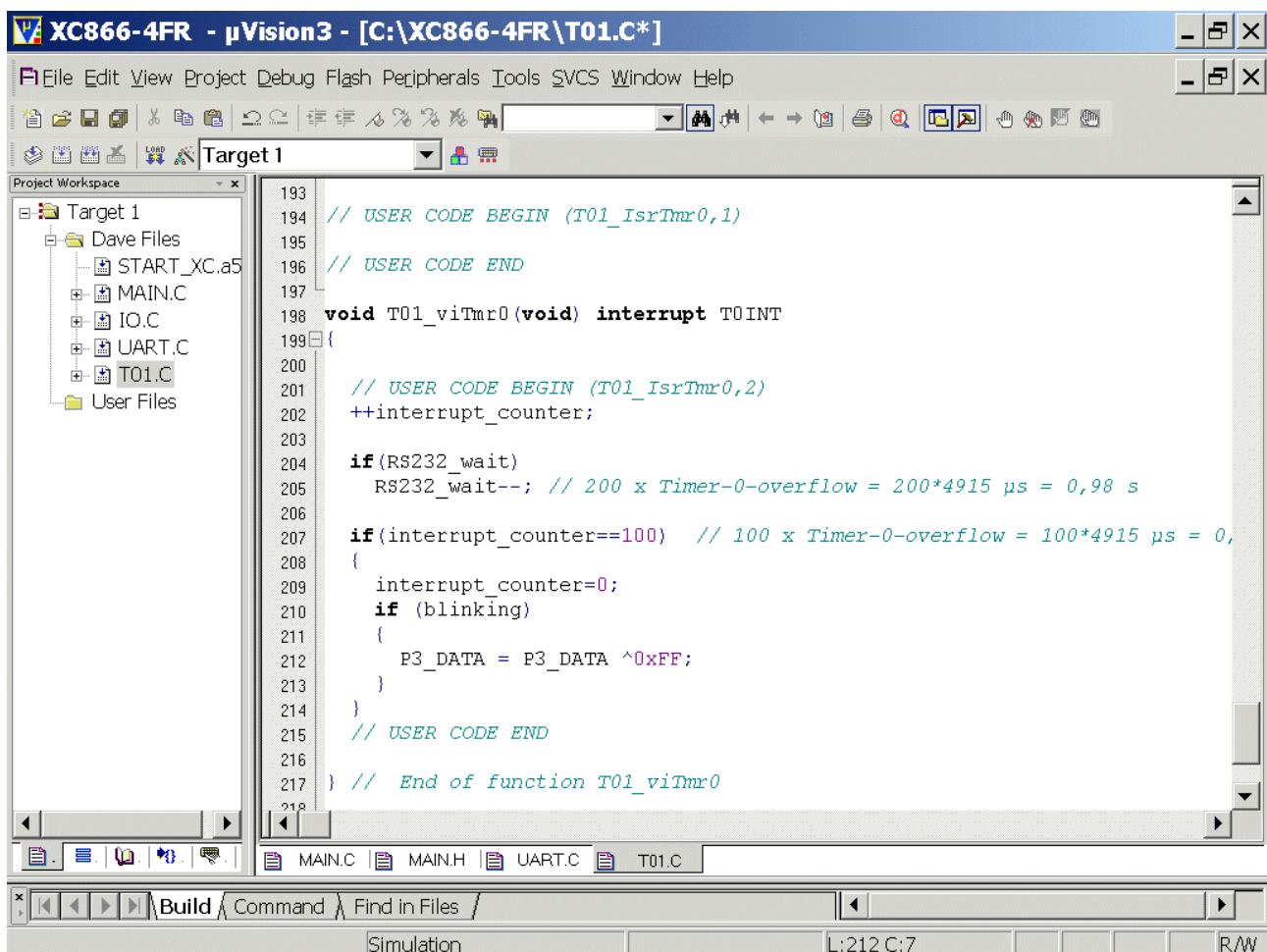
```

++interrupt_counter;

if(RS232_wait)
    RS232_wait--; // 200 x Timer-0-overflow = 200*4915 µs = 0,98 s

if(interrupt_counter==100) // 100 x Timer-0-overflow = 100*4915 µs = 0,5 s
{
    interrupt_counter=0;
    if (blinking)
    {
        P3_DATA = P3_DATA ^0xFF;
    }
}

```

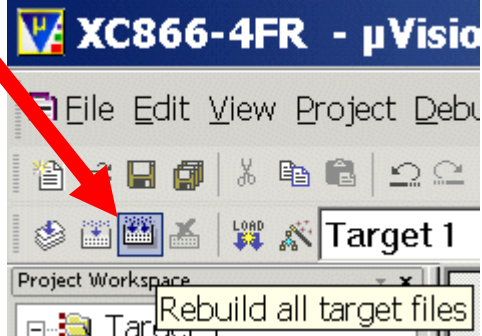


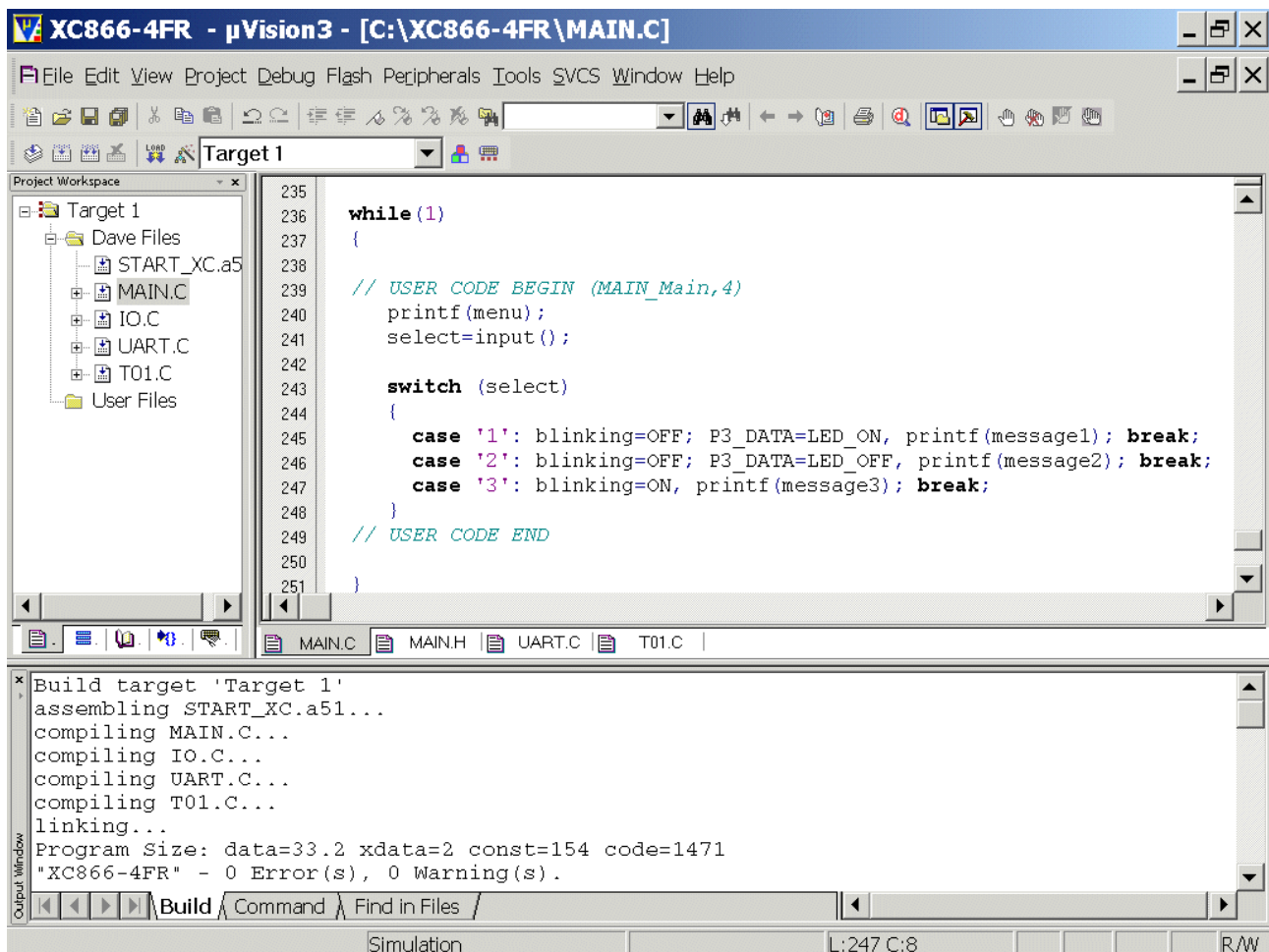
```

193
194 // USER CODE BEGIN (T01_IsrTmr0,1)
195
196 // USER CODE END
197
198 void T01_viTmr0(void) interrupt T0INT
199 {
200
201 // USER CODE BEGIN (T01_IsrTmr0,2)
202 ++interrupt_counter;
203
204 if(RS232_wait)
205     RS232_wait--; // 200 x Timer-0-overflow = 200*4915 µs = 0,98 s
206
207 if(interrupt_counter==100) // 100 x Timer-0-overflow = 100*4915 µs = 0,
208 {
209     interrupt_counter=0;
210     if (blinking)
211     {
212         P3_DATA = P3_DATA ^0xFF;
213     }
214 }
215 // USER CODE END
216
217 } // End of function T01_viTmr0
218

```

Generate your application program – generate the hex file for Memtool:

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



The screenshot shows the µVision3 IDE interface. The main window displays the source code for MAIN.C, which includes a while loop and a switch statement. The output window at the bottom shows the build process for Target 1, including assembly and compilation steps, and reports a successful build with 0 errors and 0 warnings.

```

235
236 while (1)
237 {
238
239 // USER CODE BEGIN (MAIN_Main,4)
240 printf(menu);
241 select=input();
242
243 switch (select)
244 {
245 case '1': blinking=OFF; P3_DATA=LED_ON, printf(message1); break;
246 case '2': blinking=OFF; P3_DATA=LED_OFF, printf(message2); break;
247 case '3': blinking=ON, printf(message3); break;
248 }
249 // USER CODE END
250
251 }
    
```

```

Build target 'Target 1'
assembling START_XC.a51...
compiling MAIN.C...
compiling IO.C...
compiling UART.C...
compiling T01.C...
linking...
Program Size: data=33.2 xdata=2 const=154 code=1471
"XC866-4FR" - 0 Error(s), 0 Warning(s).
    
```

Now you can close your project and μ Vision 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

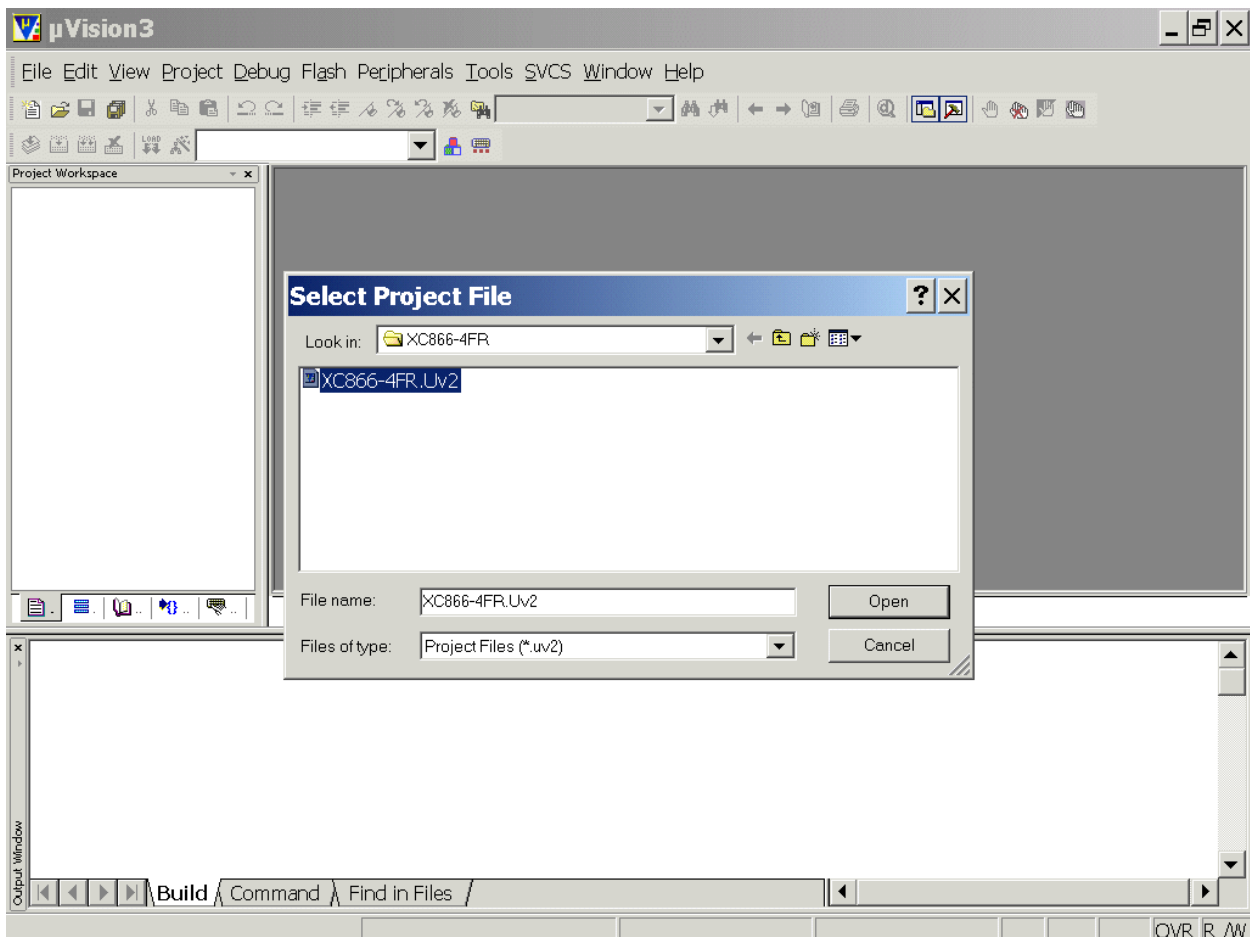
If you see an open project – close it: **Project - Close Project**

Project - Open Project

Select Project File: **Look in:** choose C:\XC866-4FR

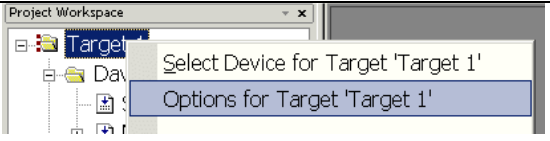
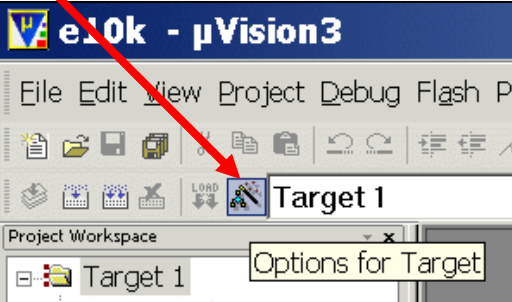
Select Project File: **Files of type:** choose Project Files (*.uv2)
choose XC866-4FR.Uv2

Open

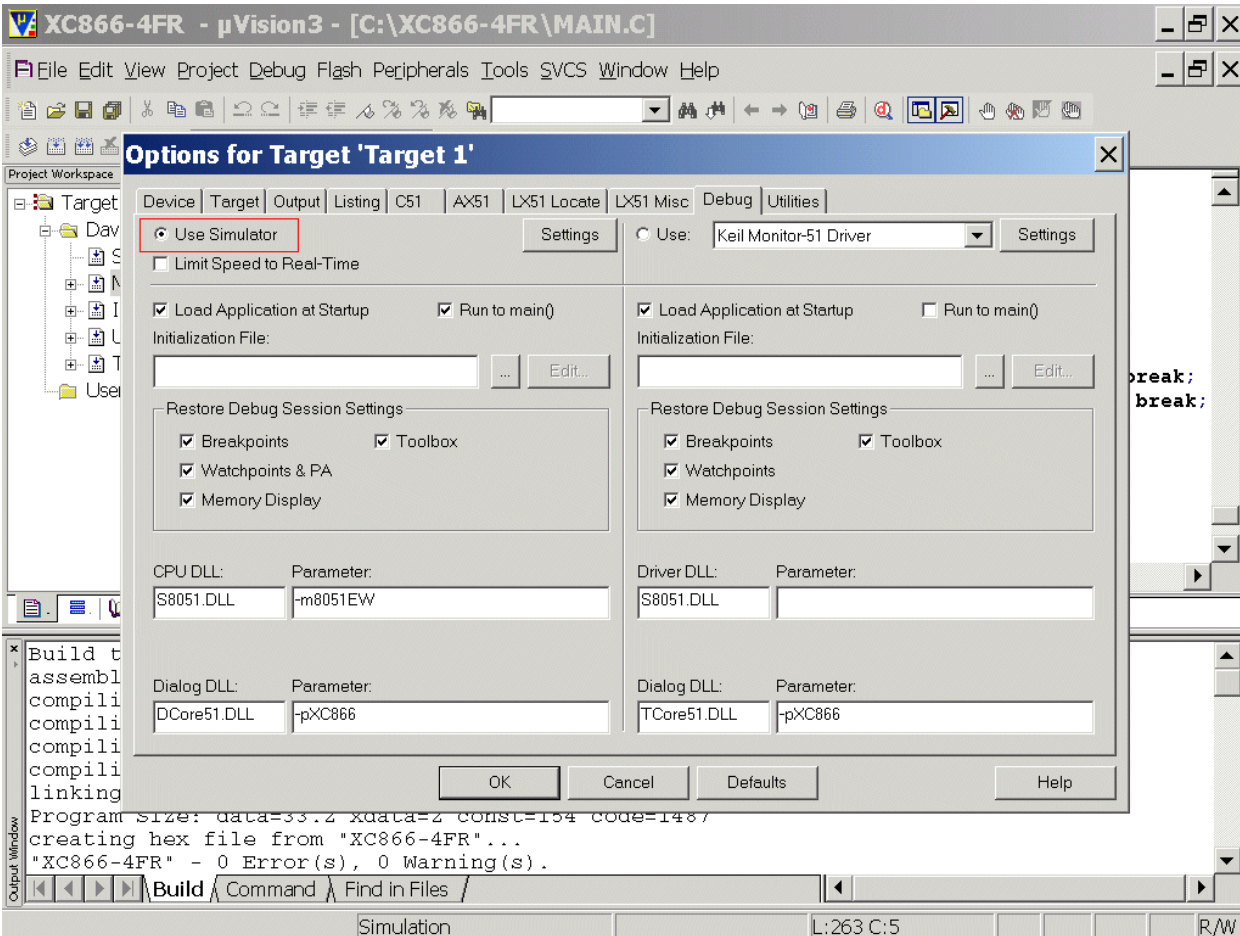


Using the simulator (first we will test our program inside the Keil Simulator):

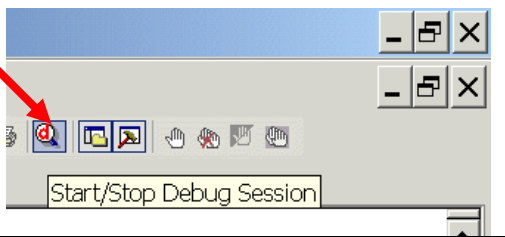
Check the configuration of the μ Vision simulator

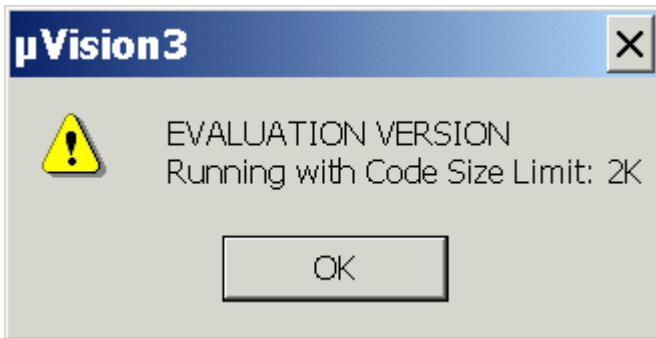
<p>mouse position: (Project Workspace, Files): Target1 click right mouse button Options for Target 'Target1'</p>	<p>or click</p>
	

Options for Target 'Target1': Debug: check Use Simulator



OK

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



OK

Project Workspace

Register	Value
r0	0x1f
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x01
r7	0x00

```

223 void main(void)
224 {
225     // USER CODE BEGIN (MAIN_Main,2)
226
227     // USER CODE END
228
229     MAIN_vInit();
230
231     // USER CODE BEGIN (MAIN_Main,3)
232     P3_DATA=LED_OFF;
233     while (RS232_wait);
234     // USER CODE END
235
236     while (1)
237     {
238
239         // USER CODE BEGIN (MAIN_Main,4)
240         printf(menu);
241         select=input();
242
243         switch (select)

```

Output Window

Load "C:\\XC866-4FR\\XC866-4FR"

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList

Build Command Find in Files

Watches

Locals Watch #1 Watch

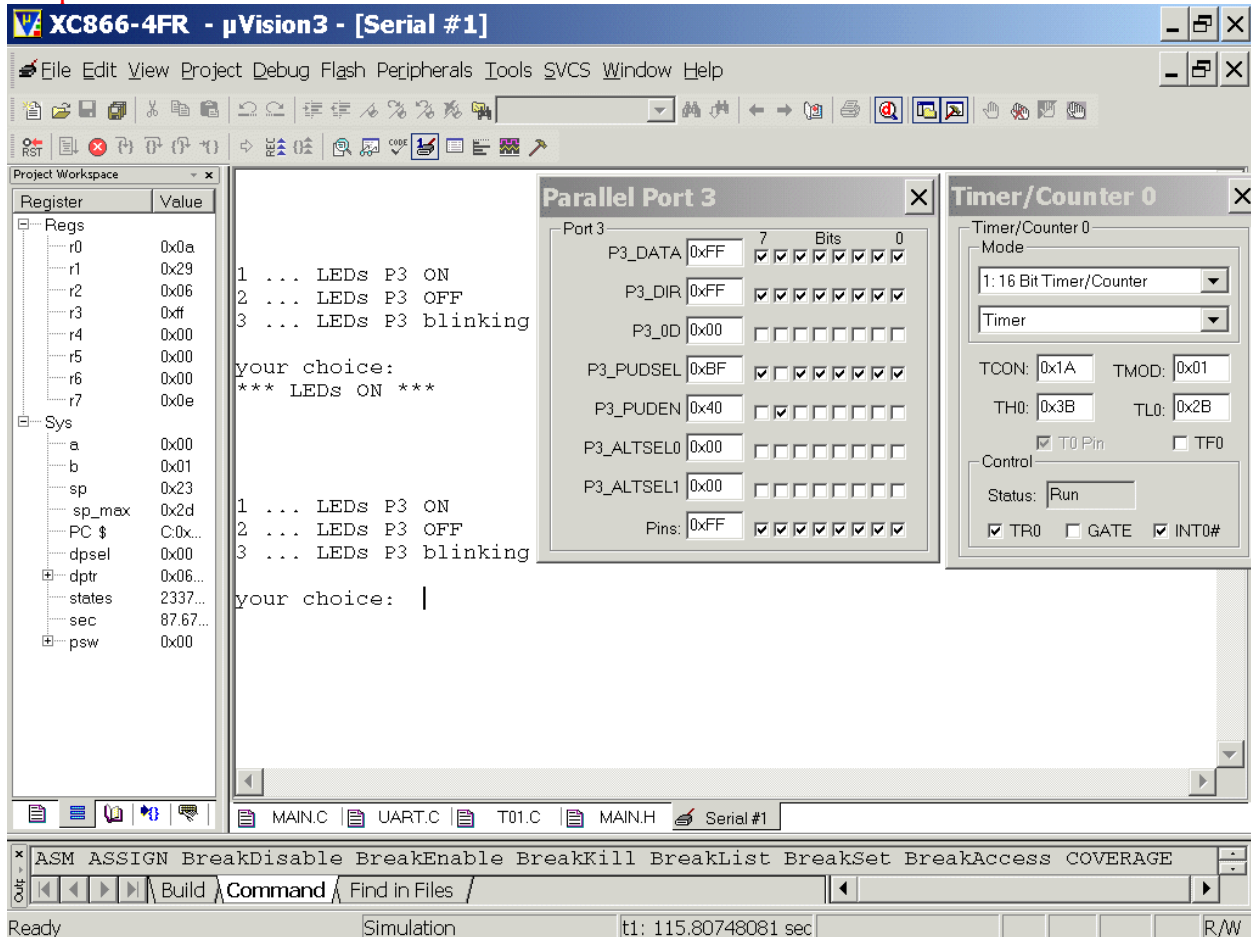
Ready Simulation t1: 0.00043654 sec L:229 C:1 R/W

Debug - Run

View - Serial Window #1

Peripherals - I/O-Ports – Port3

Peripherals – Timer – Timer0



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 3” window.

Now you can close your simulator session:

Debug - Stop Running

Debug - Start/Stop Debug Session

Now you can close your project and µVision 3:

Project
Close Project

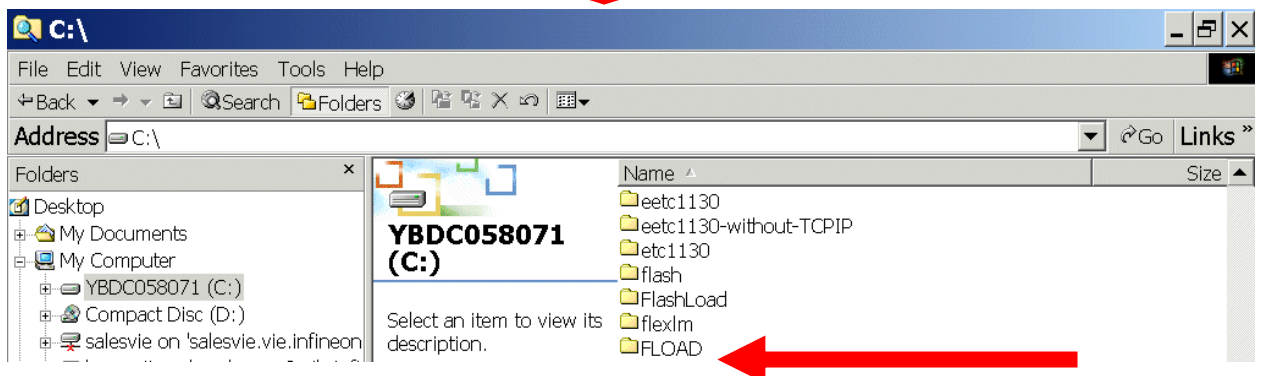
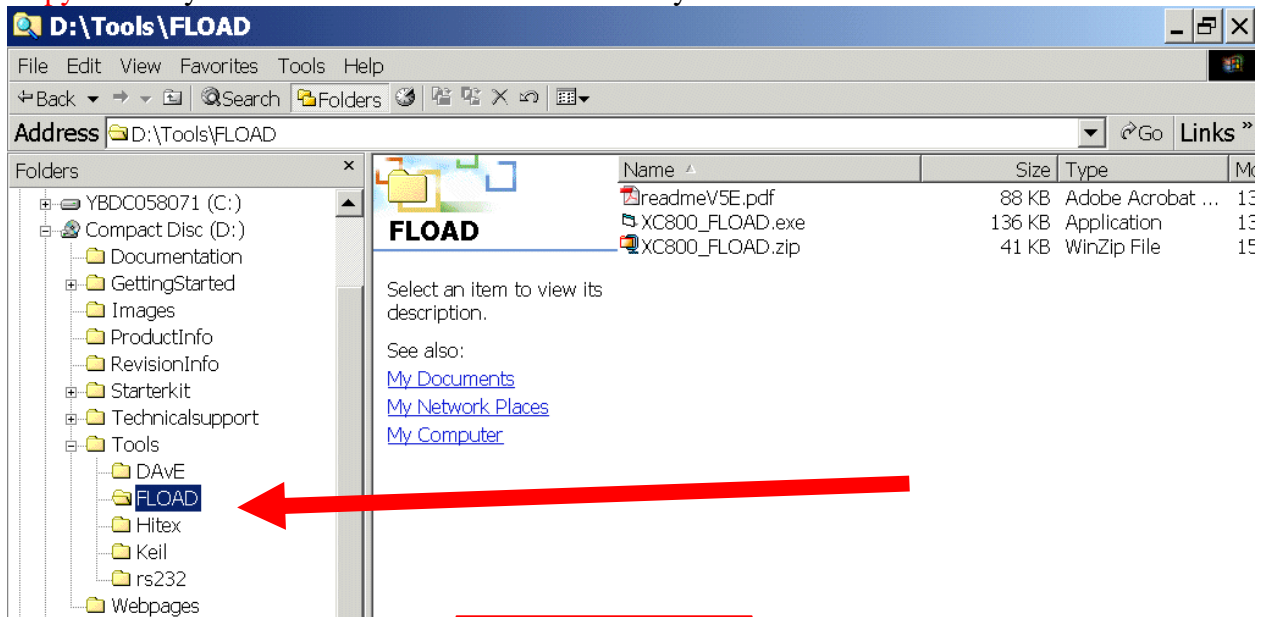
File – Exit

Using real hardware:


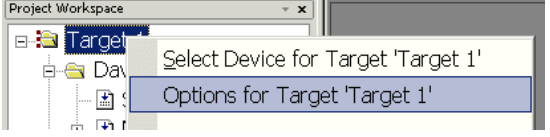
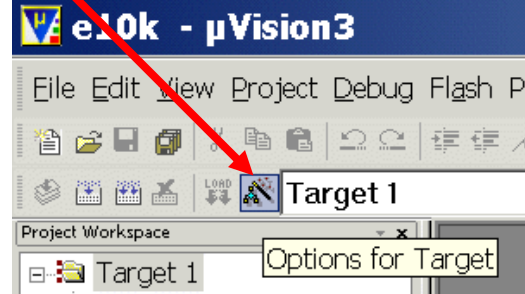
You can program now the hex output file **XC866-4FR.H86** with **XC800_FLOAD** into your real hardware:

(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.)

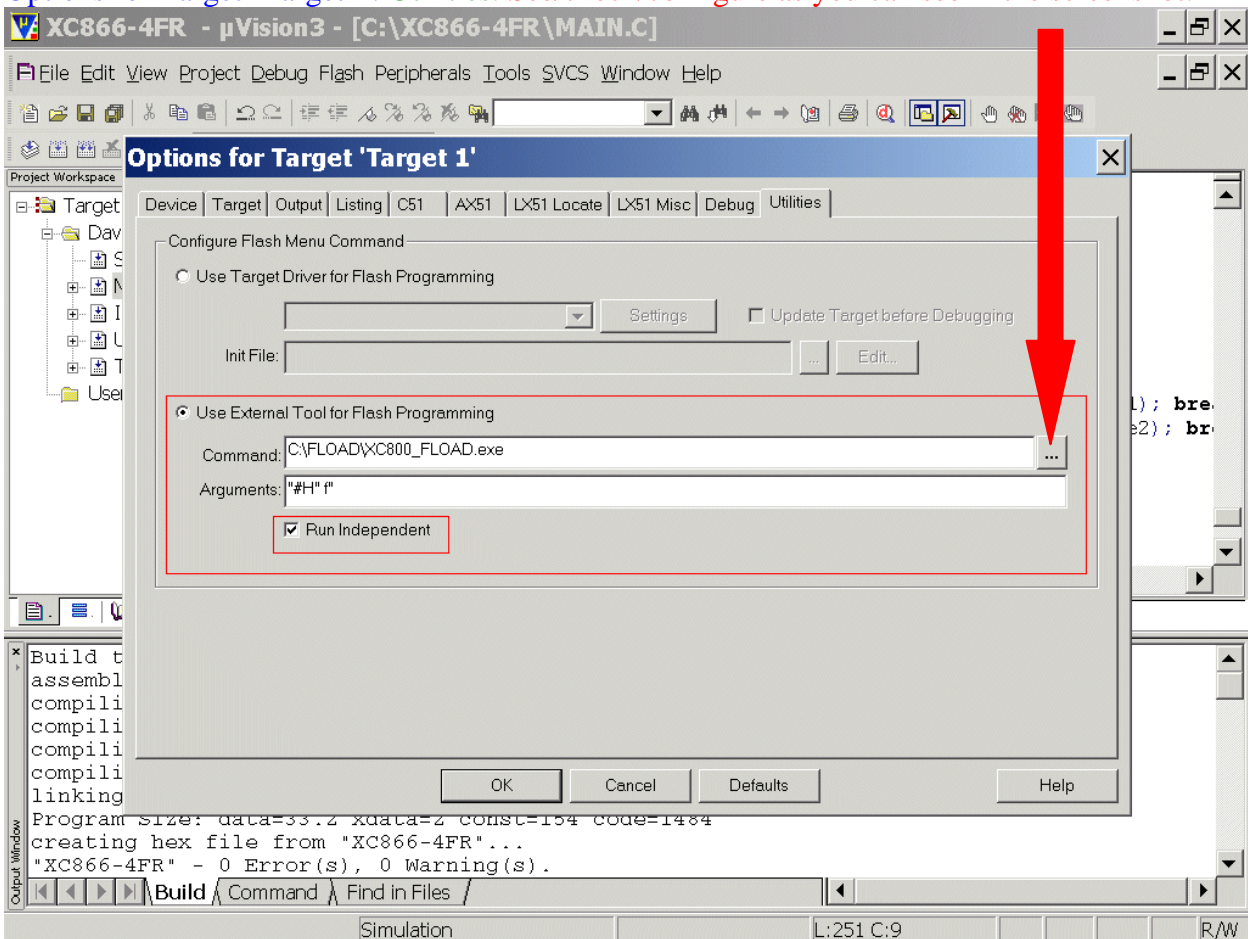
Copy directory **FLOAD** from the Starter-Kit-CD to your local Drive:



Open your μ Vision-Project:

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

Options for Target 'Target1': Utilities: See/check/configure as you can see in the screenshot:

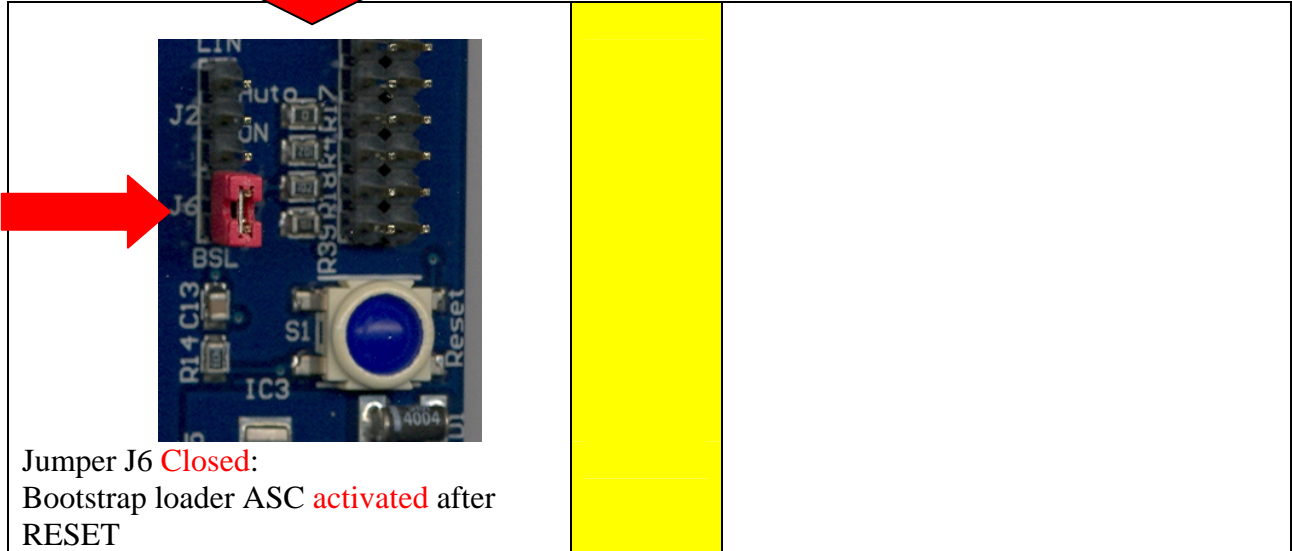
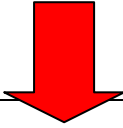


The screenshot shows the 'Options for Target 'Target 1'' dialog box with the 'Utilities' tab selected. The 'Use External Tool for Flash Programming' option is chosen. The 'Command' field contains 'C:\FLOAD\XC800_FLOAD.exe' and the 'Run Independent' checkbox is checked. A red box highlights these two elements, and a red arrow points to the ellipsis button next to the command field.

OK

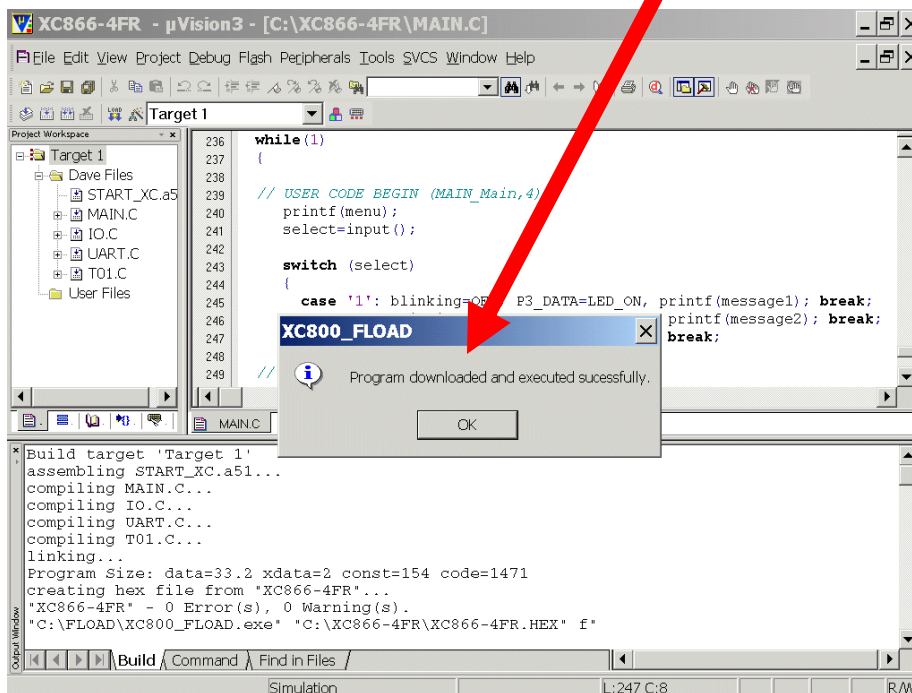
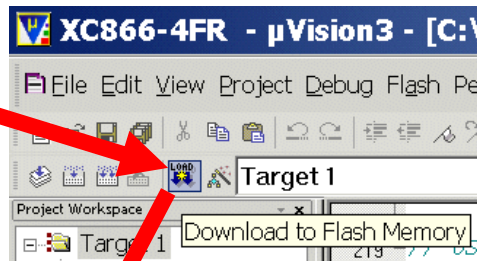
Activate Bootstrap Loader ASC

Jumper J6:



Jumper J6 **Closed**:
Bootstrap loader ASC **activated** after
RESET

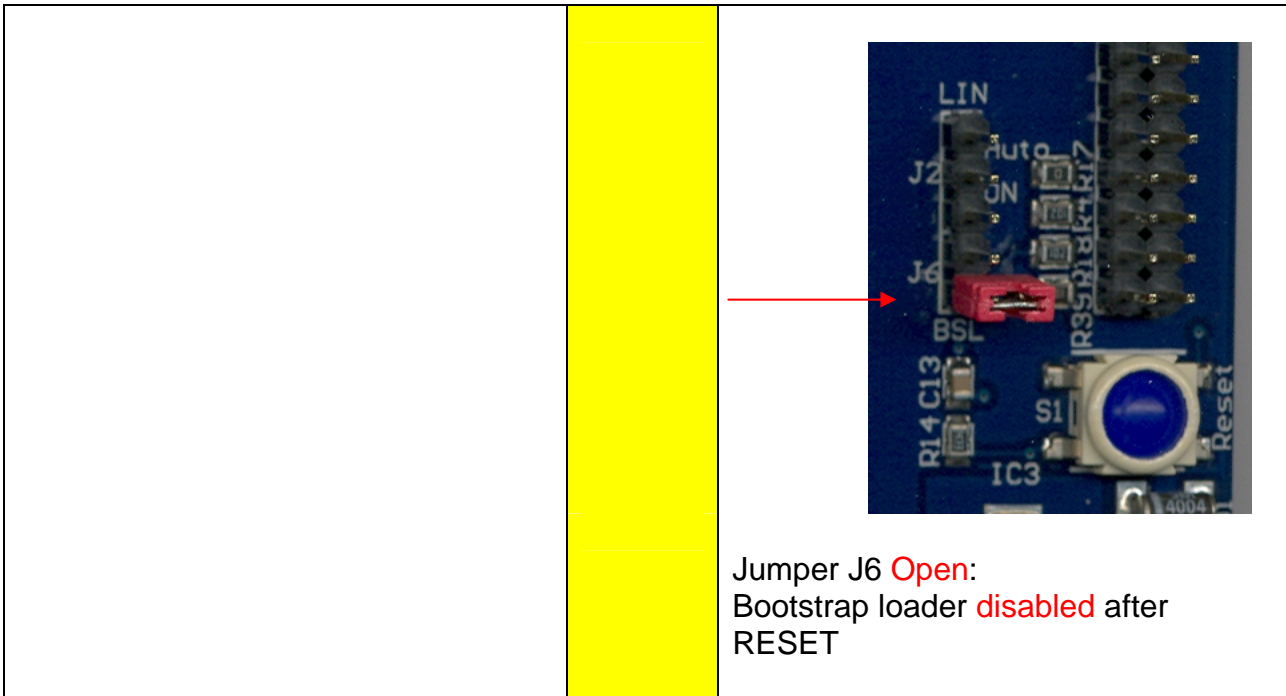
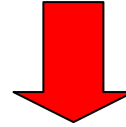
Click



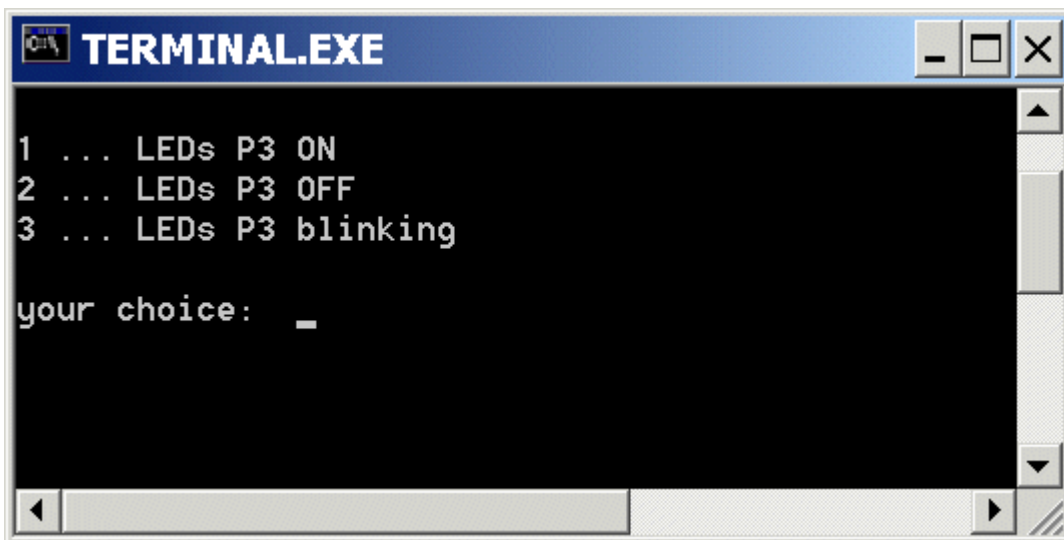
OK

Standard Start

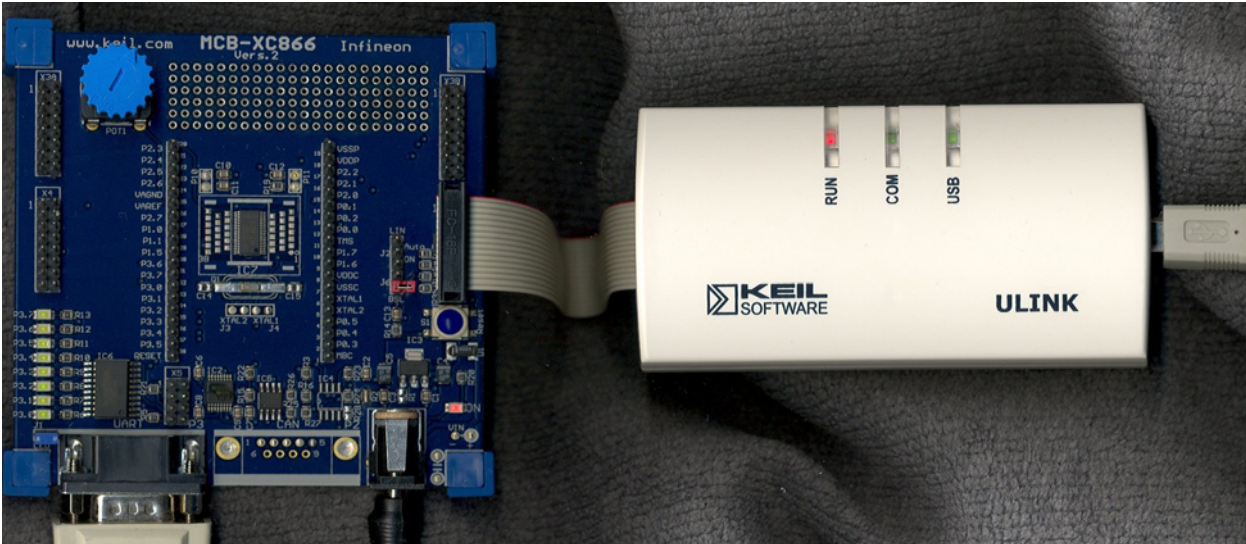
Jumper J6:



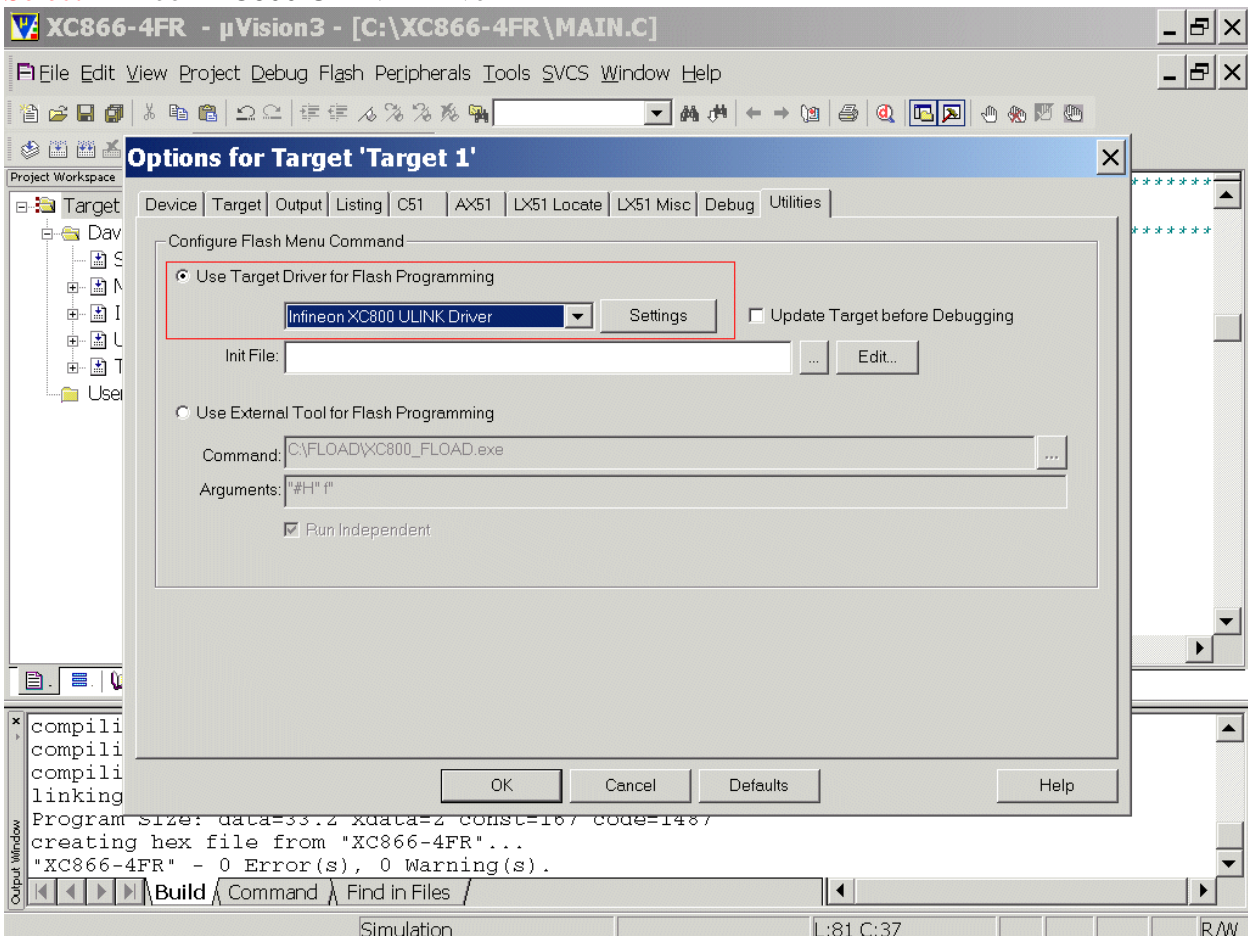
Your application is running:



Using ULINK (OnChipFlash-Programming and Debugging (using the JTAG interface)):

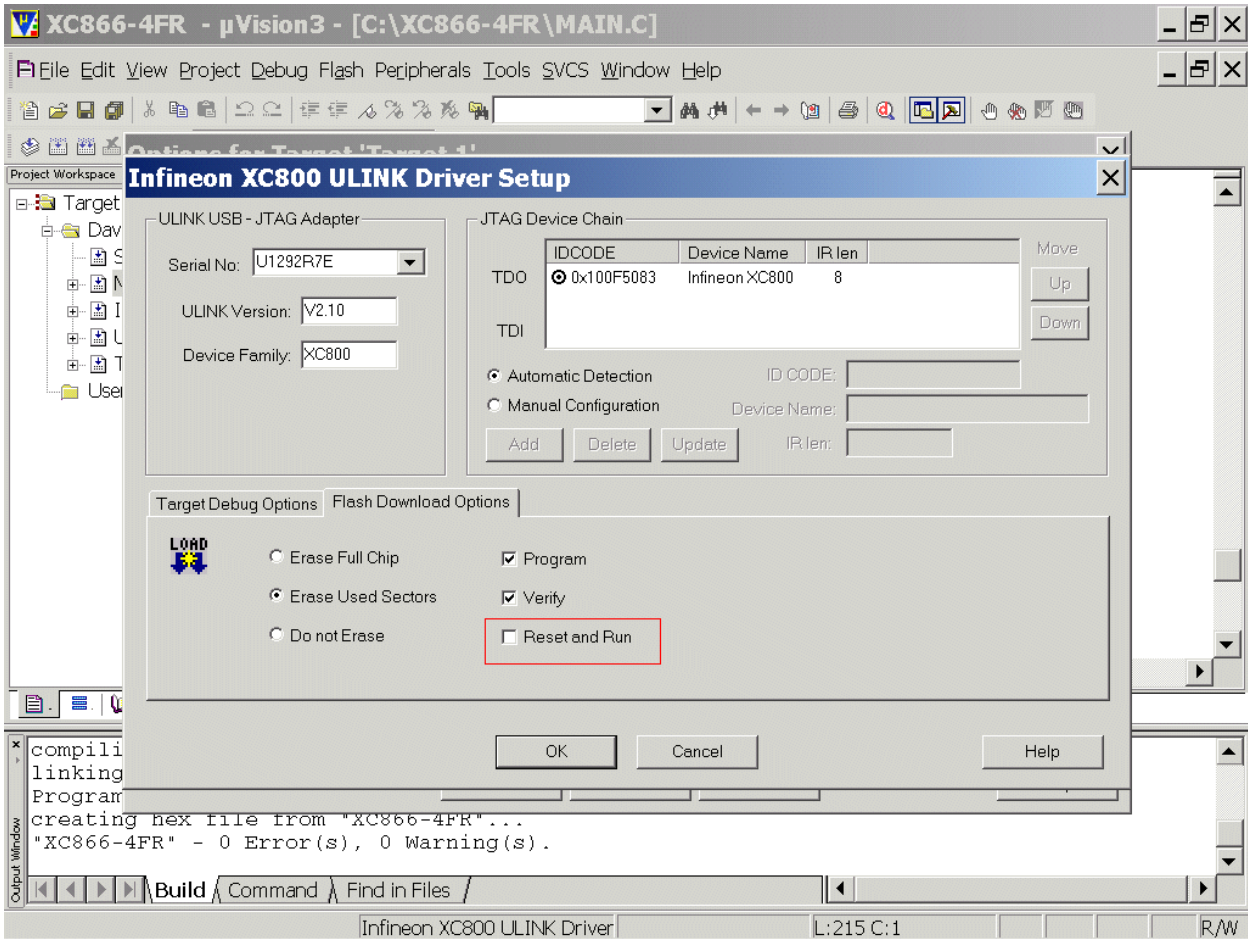


Utilities: **Configure Flash Menu Command:** click  Use Target Driver for Flash Programming
Select Infineon XC800 ULINK Driver



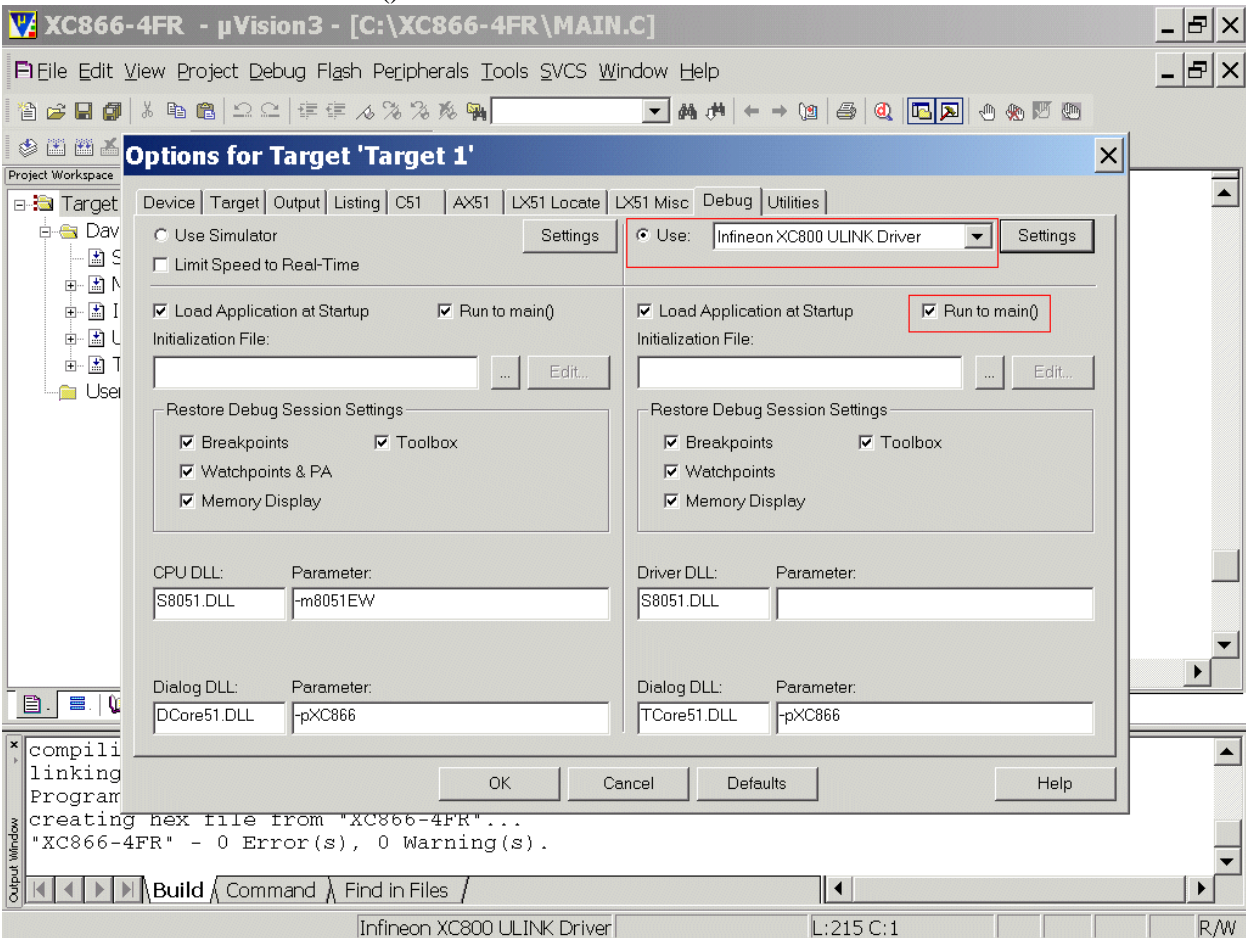
Click Settings

Check/configure: Flash Download Options: Reset and Run



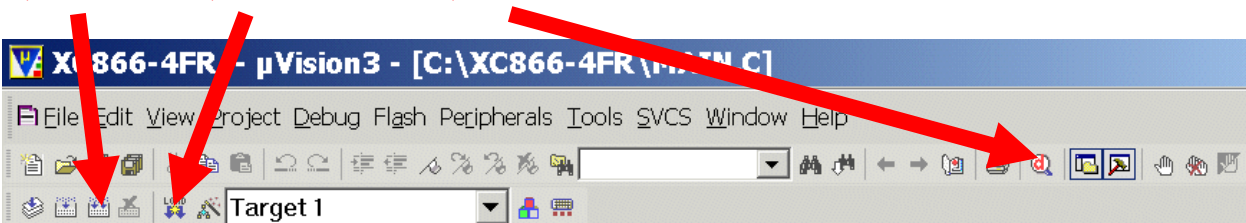
OK

Debug: click Use select Infineon XC800 ULINK Driver
click/check Run to main()

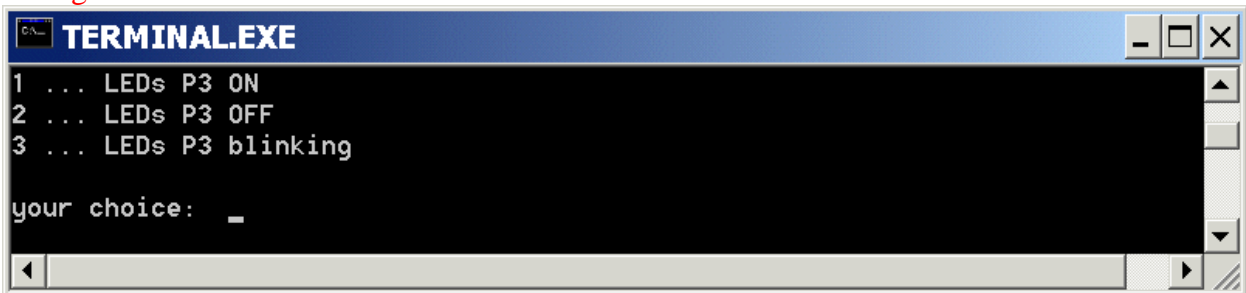


OK

1.) click 2.) click 3.) click

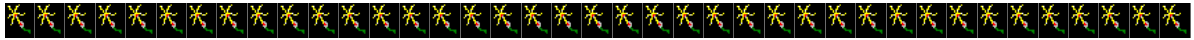


Debug – Run



In this appnote you have learned how to use the XC866 board together with the Keil compiler.

5.) Feedback (XC866-4FR): 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>