XC2287M HOT

Getting Started Exercise 1

Get familiar with the tool chain and uses a timer triggered LED toggle

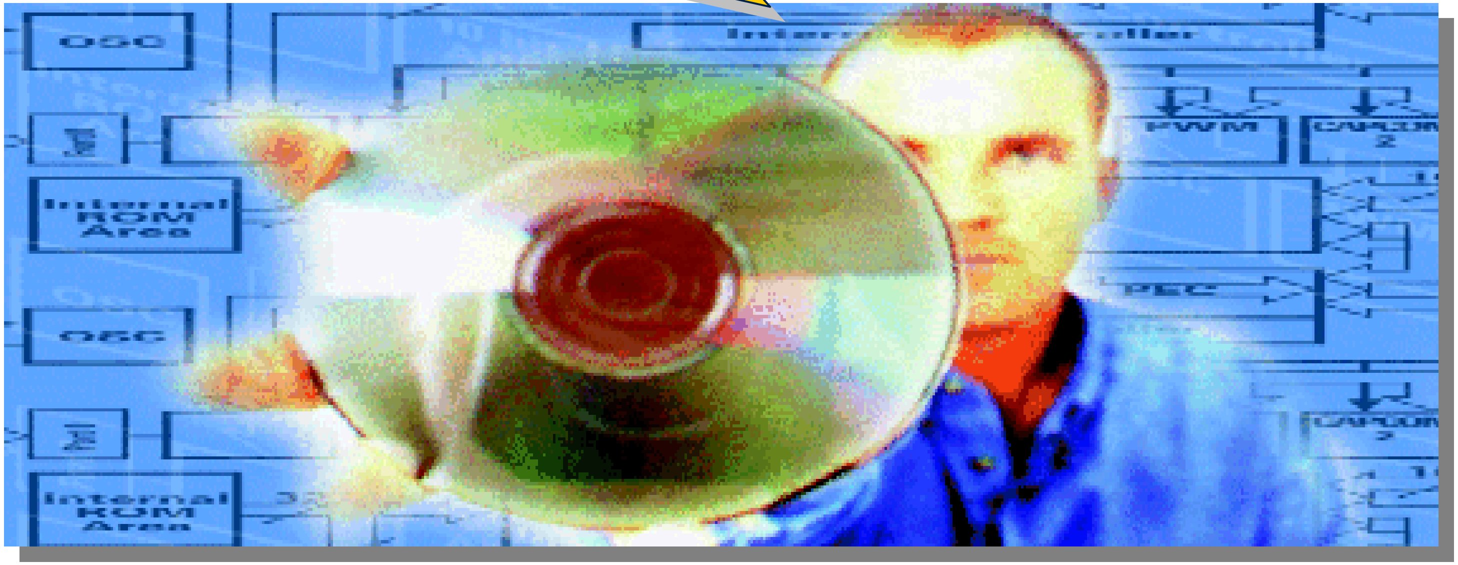
Device: XC2287M-104F80
Compiler: Tasking Viper 2.4r1
Code Generator: DAvE 2.1

Infineon

Never stop thinking

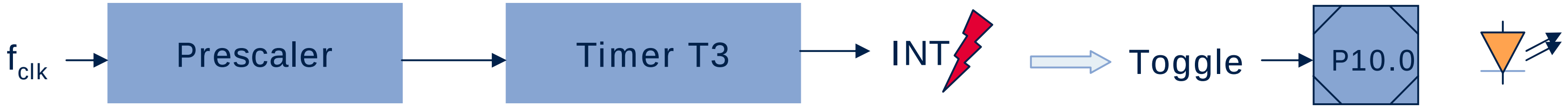
Pin/LED toggle using a simple timer

Let’s get started now!

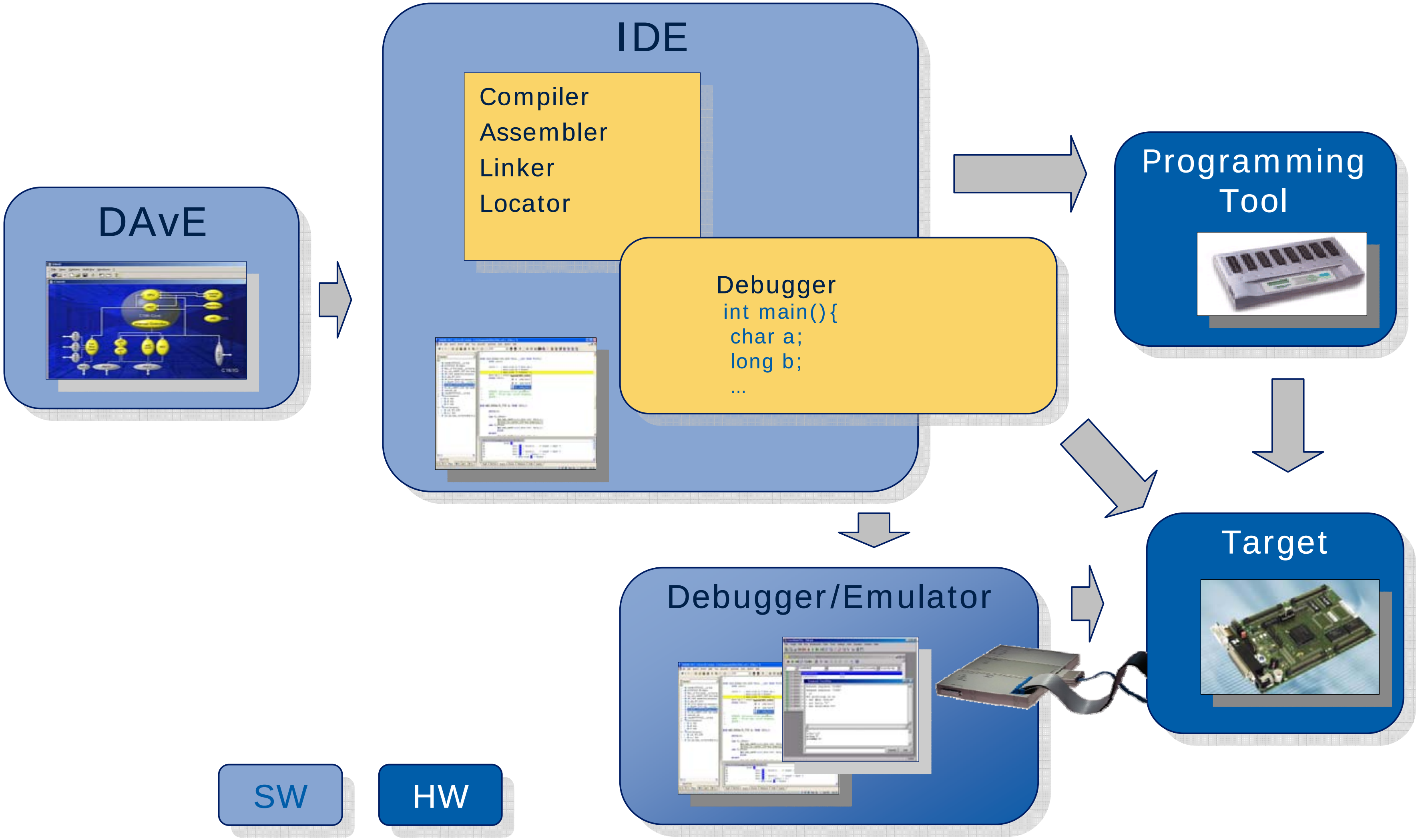


Getting Started 1
Objective

- Very simple example
  - See interaction with different tools
  - Check tool setup/installation
- Setup Prescaler and Timer T3 of GPT12
- Using timer interrupt to toggle port pin



Getting Started 1
Interaction of Development Tools



```
int main(){
 char a;
 long b;
 ...
```

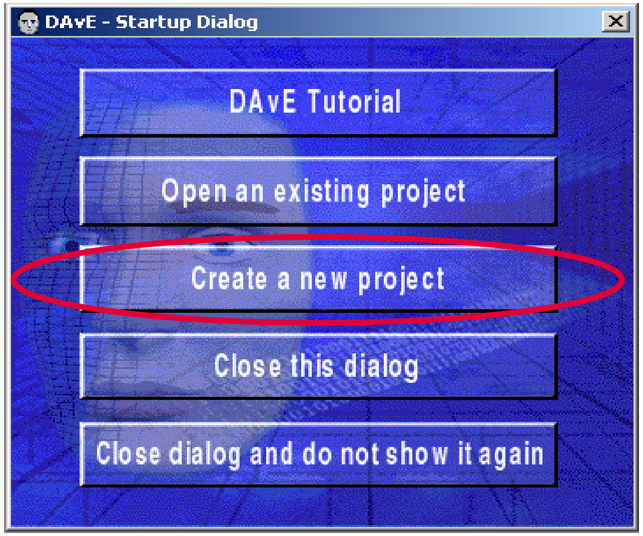
Getting Started 1
Overview

- DAvE
  - □ General Project Settings
  - □ Setup Timer T3
  - □ Port Settings
  - □ Generate Code (Peripheral Initialization; C Code)
- Compiler
  - □ Import your DAvE Project
  - □ Include Peripheral Initialization
  - □ Add Application C Code (Interrupt routine)
  - □ Compile Code
- Debugger
  - □ Download/Flash Compiled Code
  - □ Execute/Debug Code

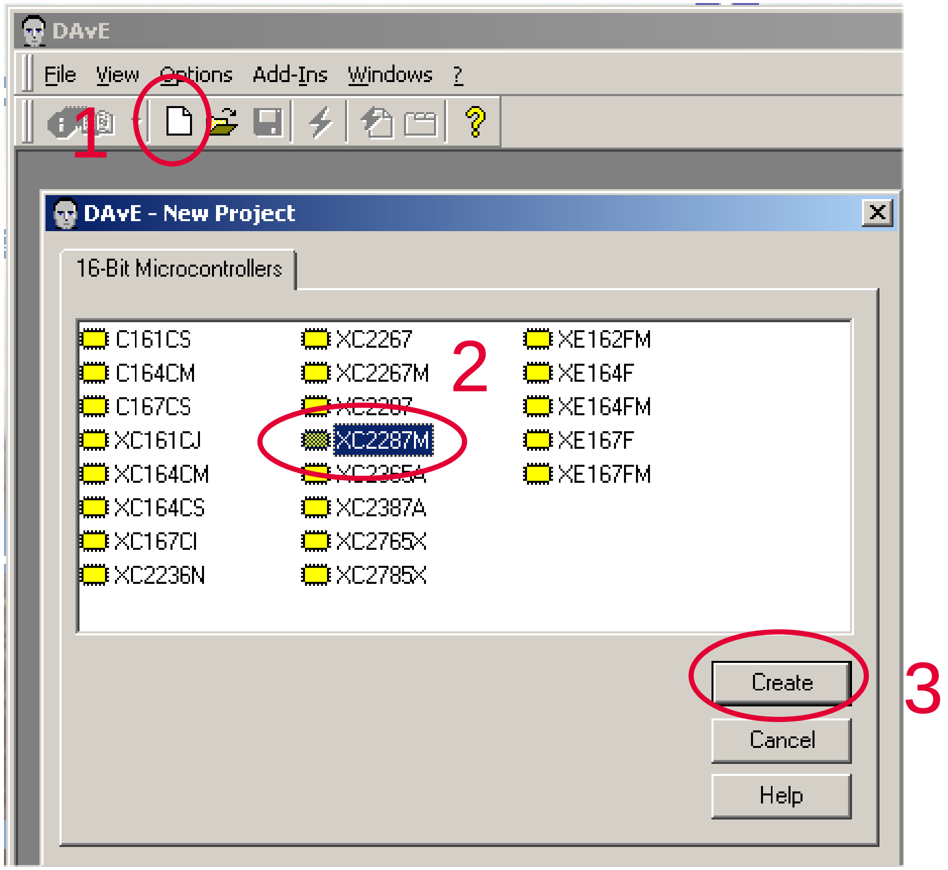
Getting Started 1 - DAvE Configurations
Start DAvE

- **Start DAvE**
  - □ Click on the DAvE

- **Create a new project** (Startup Dialog pop up automatically)
  - □ Click on 'Create a new project' or select File -> New
  - □ Select microcontroller: 'XC2287M'



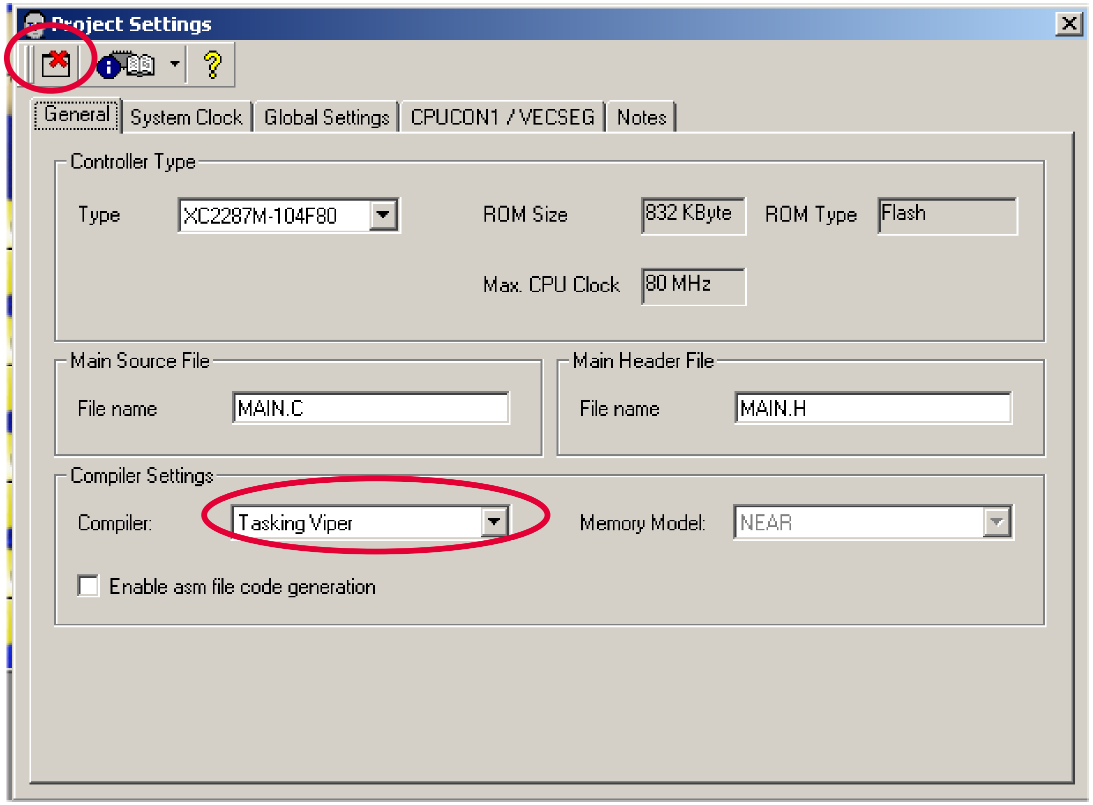
Getting Started 1 - DAvE Configurations
Select Device



C161CS
C164CM
C167CS
XC161CJ
XC164CM
XC164CS
XC167CI
XC2236N
XC2267
XC2267M
XC2287
XC2287M
XC2365A
XC2387A
XC2765X
XC2785X
XE162FM
XE164F
XE164FM
XE167F
XE167FM

Getting Started 1 - DAvE Configurations Project Settings

- Project Settings
- Close the window



Controller Type

Type: XC2287M-104F80

ROM Size: 832 KByte

ROM Type: Flash

Max. CPU Clock: 80 MHz

Main Source File

File name: MAIN.C

Main Header File

File name: MAIN.H

Compiler Settings

Compiler: Tasking Viper

Memory Model: NEAR

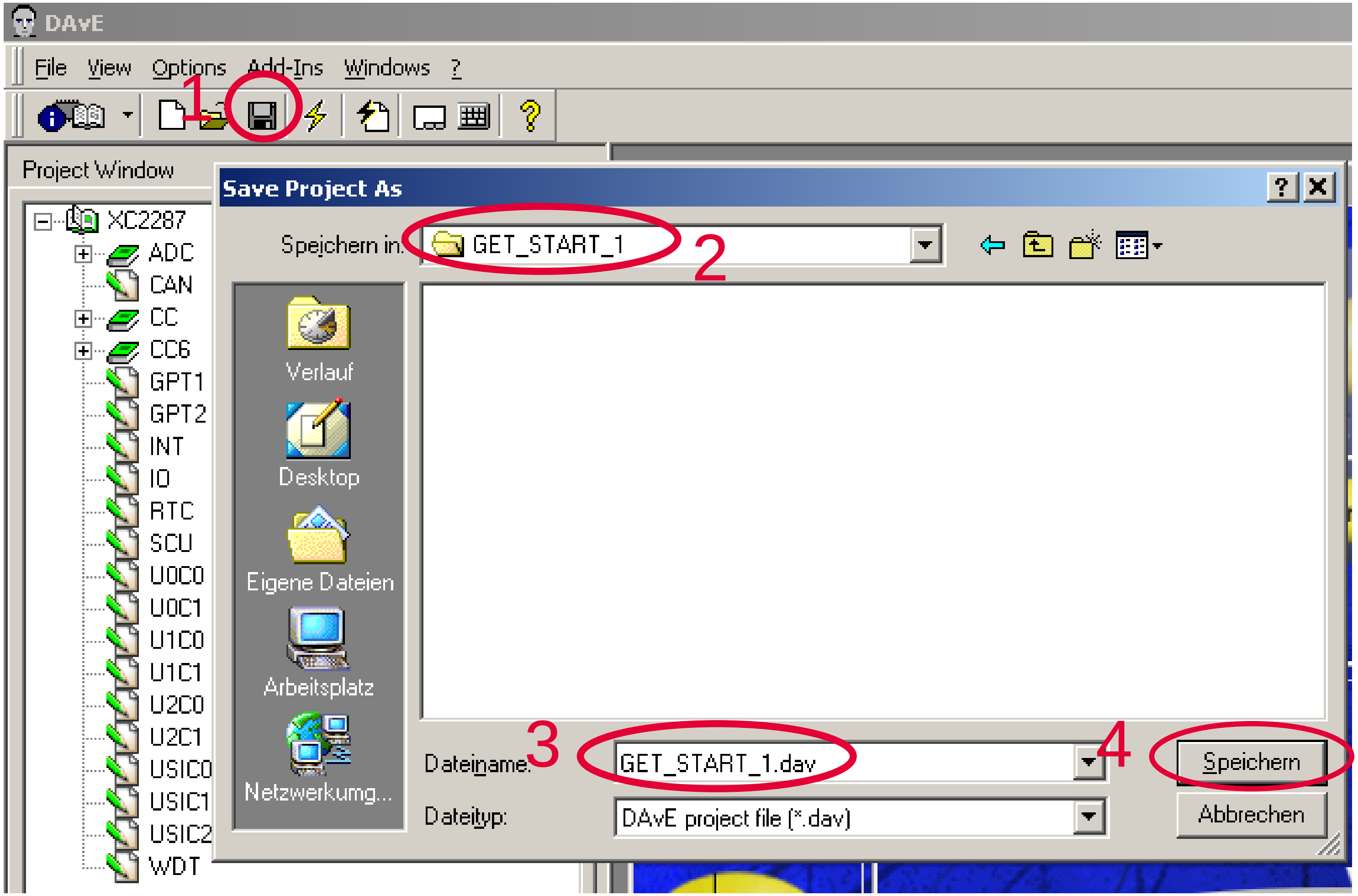
☐ Enable asm file code generation

Getting Started 1 - DAvE Configurations
Save DAvE Project

- Save your DAvE project 💾
  - Path: C:\IFX_HOT\XC2287M\Examples\Dave\GET_START_1
  - Project name: GET_START_1\GET_START_1.dav

Getting Started 1 - DAvE Configurations Save DAvE Project

- Save your DAvE Project File

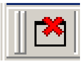


Speichern in: GET_START_1

Dateiname: GET_START_1.dav

Dateityp: DAvE project file (*.dav)

Getting Started 1 - DAvE Configurations GPT1 settings

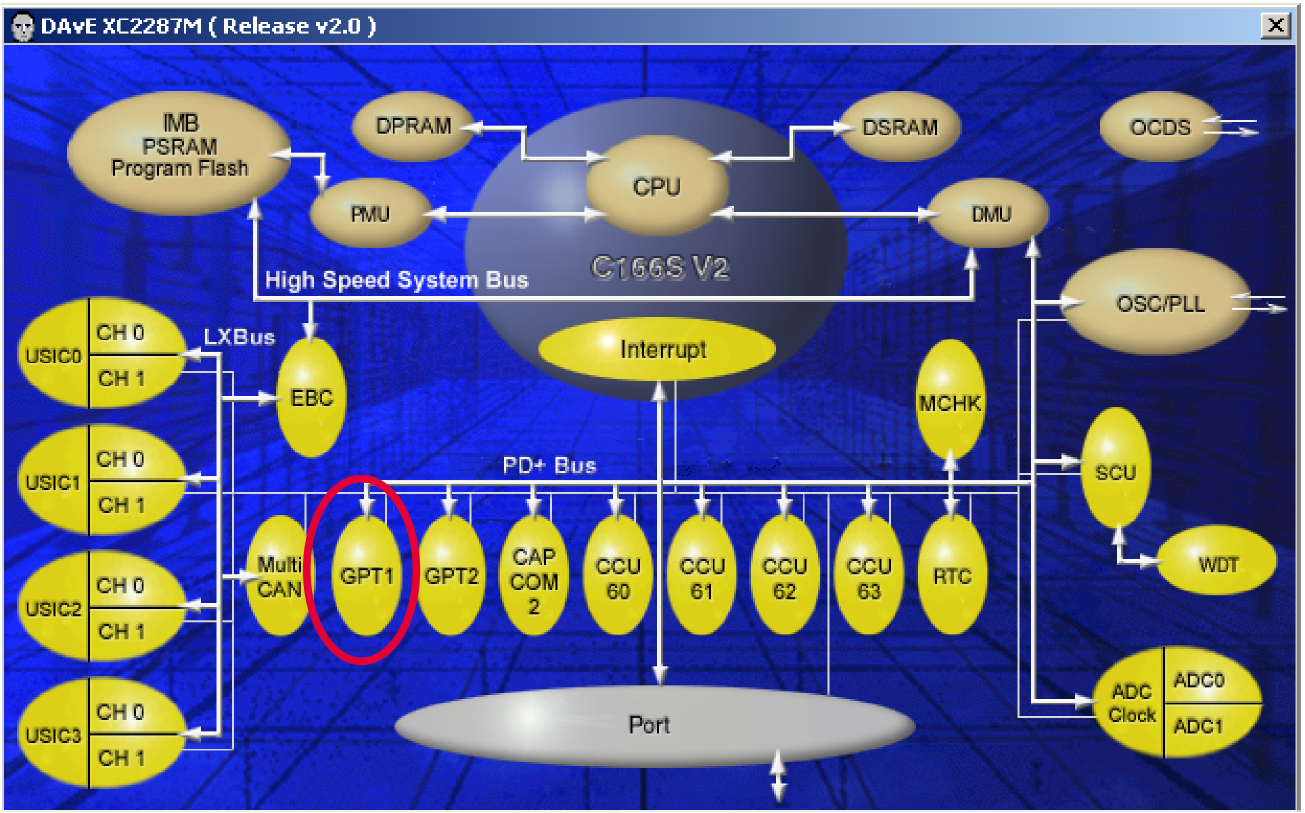
- **Timer Settings**
  - □ Enable the module (enable clock path)
  - □ Set Prescaler to get ~ 1 second timer overflow
  - □ Generate interrupt on overflow
  - □ Configure interrupt routine
  - □ Include timer setup function
  - □ Close GPT1 setup window 

Getting Started 1 - DAvE Configurations GPT1 settings

Timer Settings

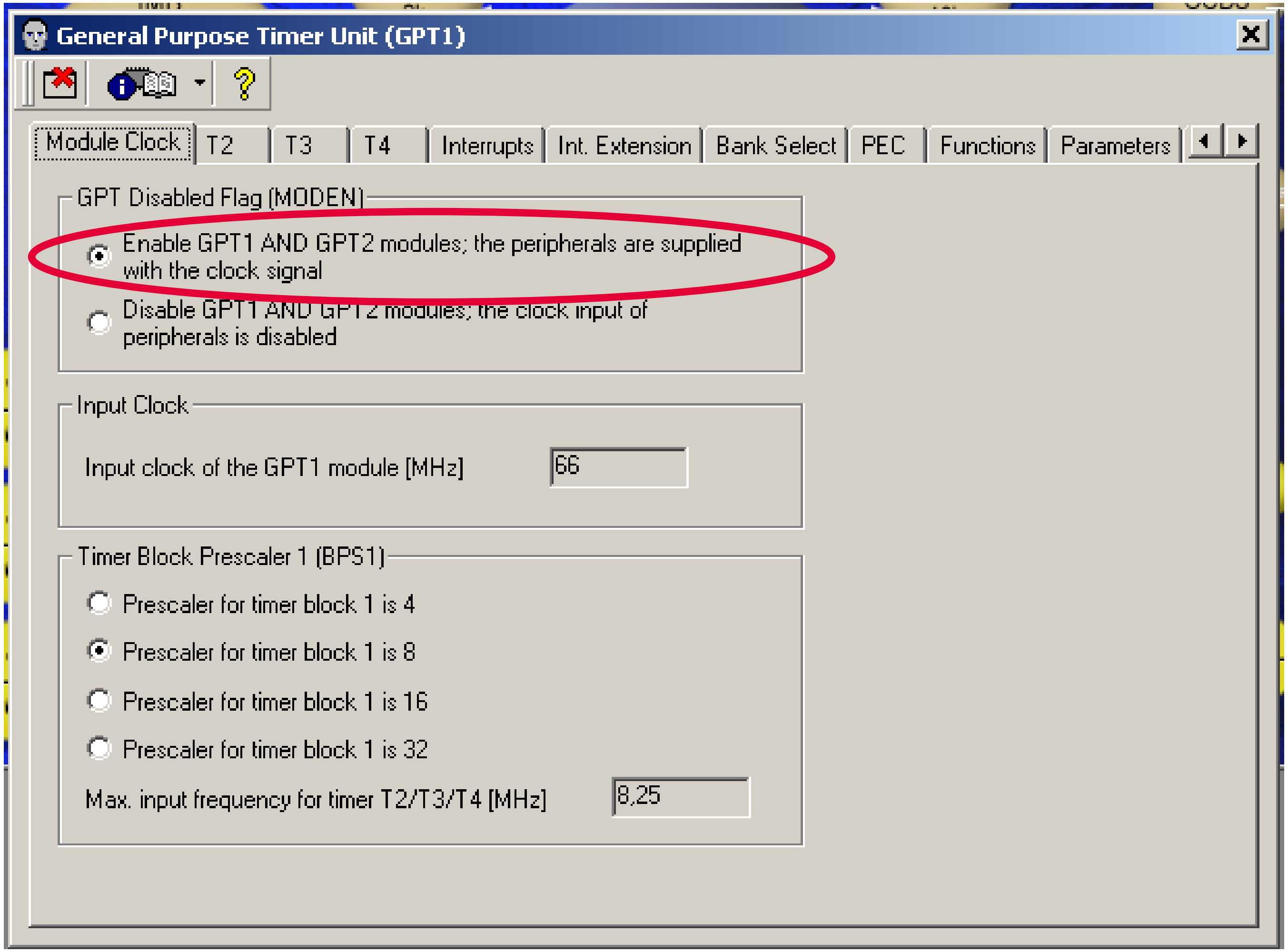
GPT1 :

- Click on


DAvE XC2287M (Release v2.0)

Getting Started 1 - DAvE Configurations GPT1 settings

- Module Clock settings



GPT Disabled Flag (MODEN)

● Enable GPT1 AND GPT2 modules; the peripherals are supplied with the clock signal

○ Disable GPT1 AND GPT2 modules; the clock input of peripherals is disabled

Input Clock

Input clock of the GPT1 module [MHz] 66

Timer Block Prescaler 1 (BPS1)

○ Prescaler for timer block 1 is 4

● Prescaler for timer block 1 is 8

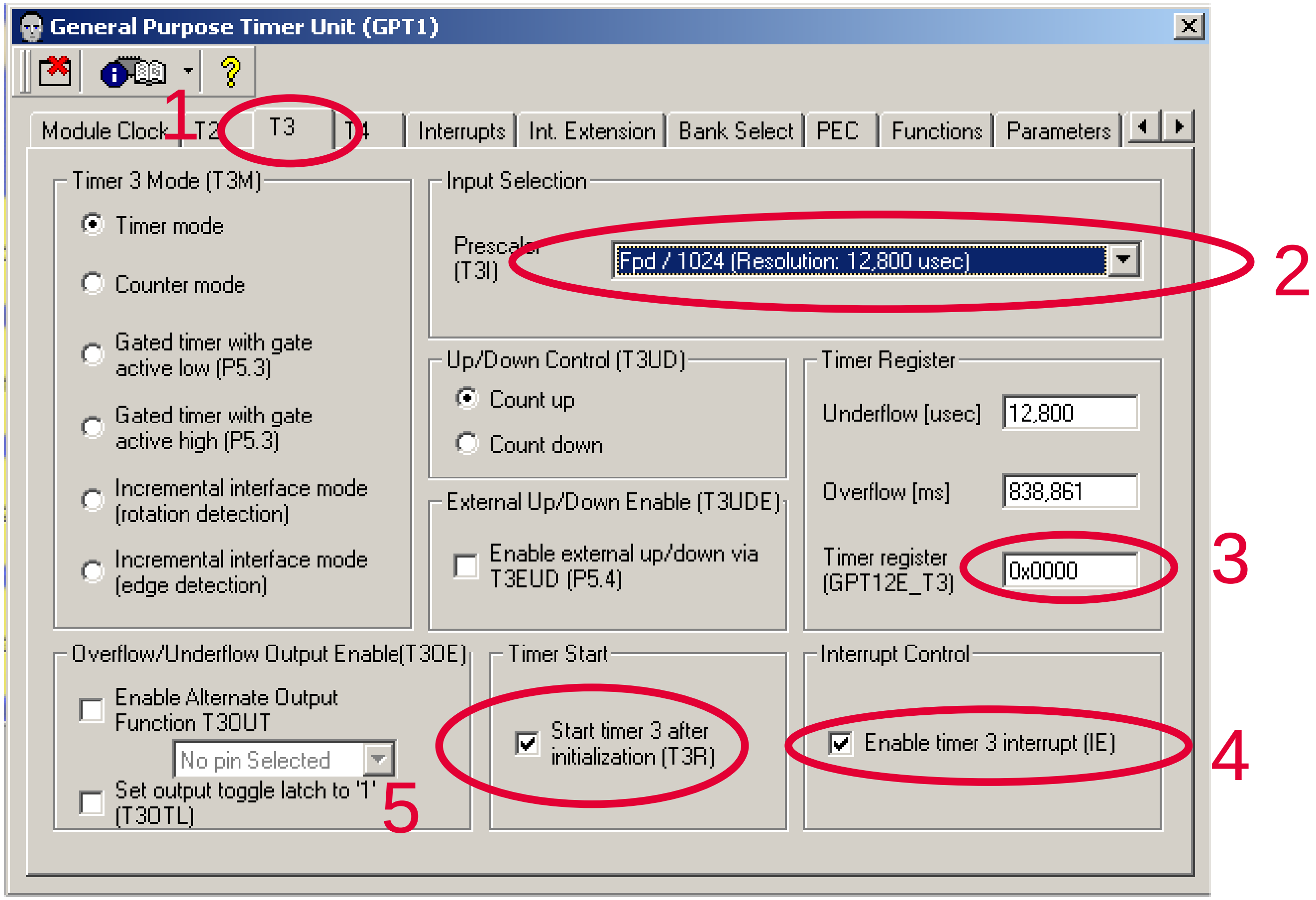
○ Prescaler for timer block 1 is 16

○ Prescaler for timer block 1 is 32

Max. input frequency for timer T2/T3/T4 [MHz] 8.25

Getting Started 1 - DAvE Configurations
GPT1 settings

- T3 settings



General Purpose Timer Unit (GPT1)

Timer 3 Mode (T3M)

● Timer mode
○ Counter mode
○ Gated timer with gate active low (P5.3)
○ Gated timer with gate active high (P5.3)
○ Incremental interface mode (rotation detection)
○ Incremental interface mode (edge detection)

Input Selection

Prescaler (T3I): Fpd / 1024 (Resolution: 12,800 usec)

Up/Down Control (T3UD)

● Count up
○ Count down

External Up/Down Enable (T3UDE)

□ Enable external up/down via T3EUD (P5.4)

Timer Register

Underflow [usec]: 12,800
Overflow [ms]: 838,861
Timer register (GPT12E_T3): 0x0000

Overflow/Underflow Output Enable(T3OE)

□ Enable Alternate Output Function T3OUT
No pin Selected
□ Set output toggle latch to '1' (T3OTL)

Timer Start

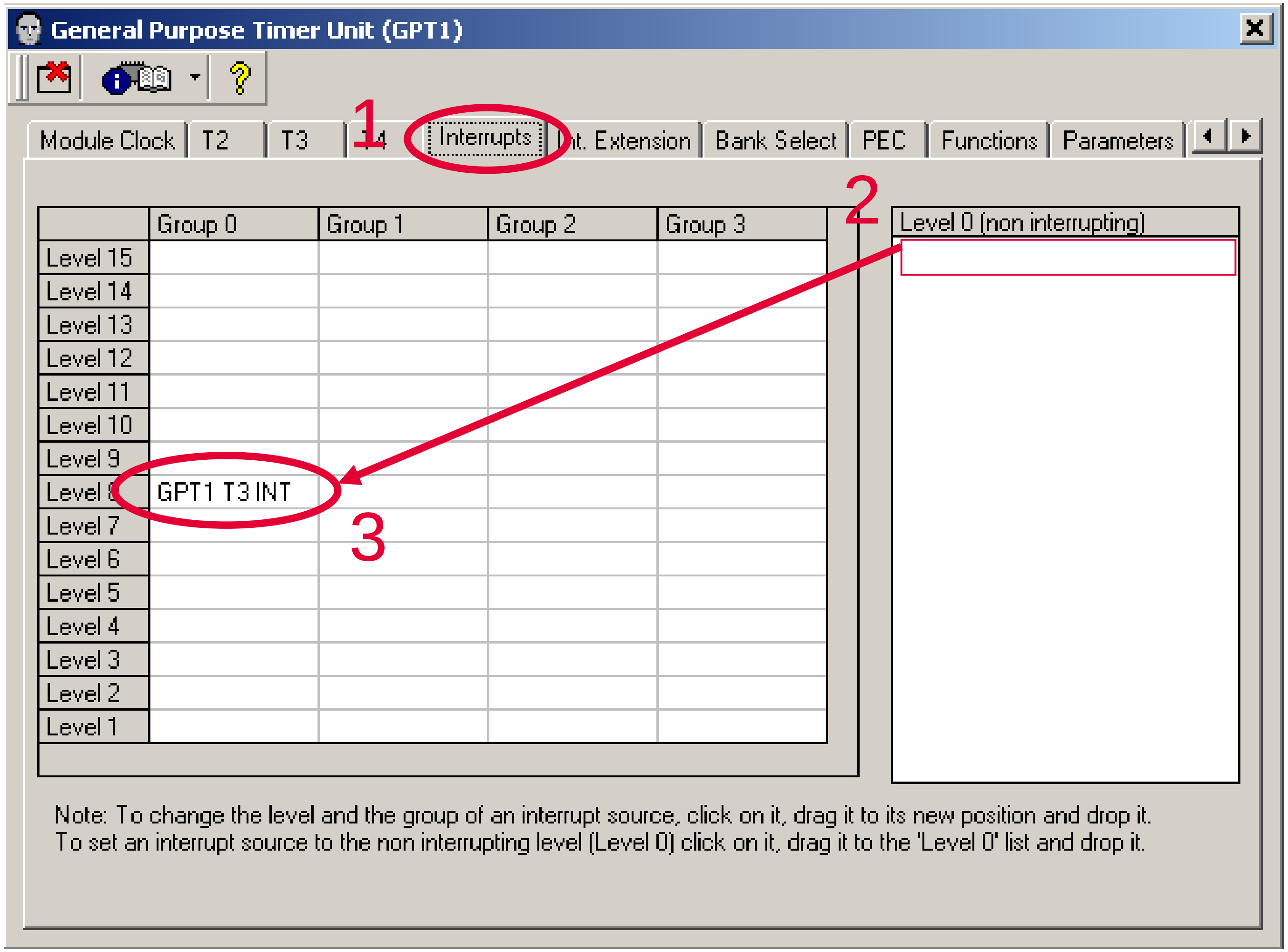
☑ Start timer 3 after initialization (T3R)

Interrupt Control

☑ Enable timer 3 interrupt (IE)

Getting Started 1 - DAvE Configurations GPT1 settings

- Interrupts settings



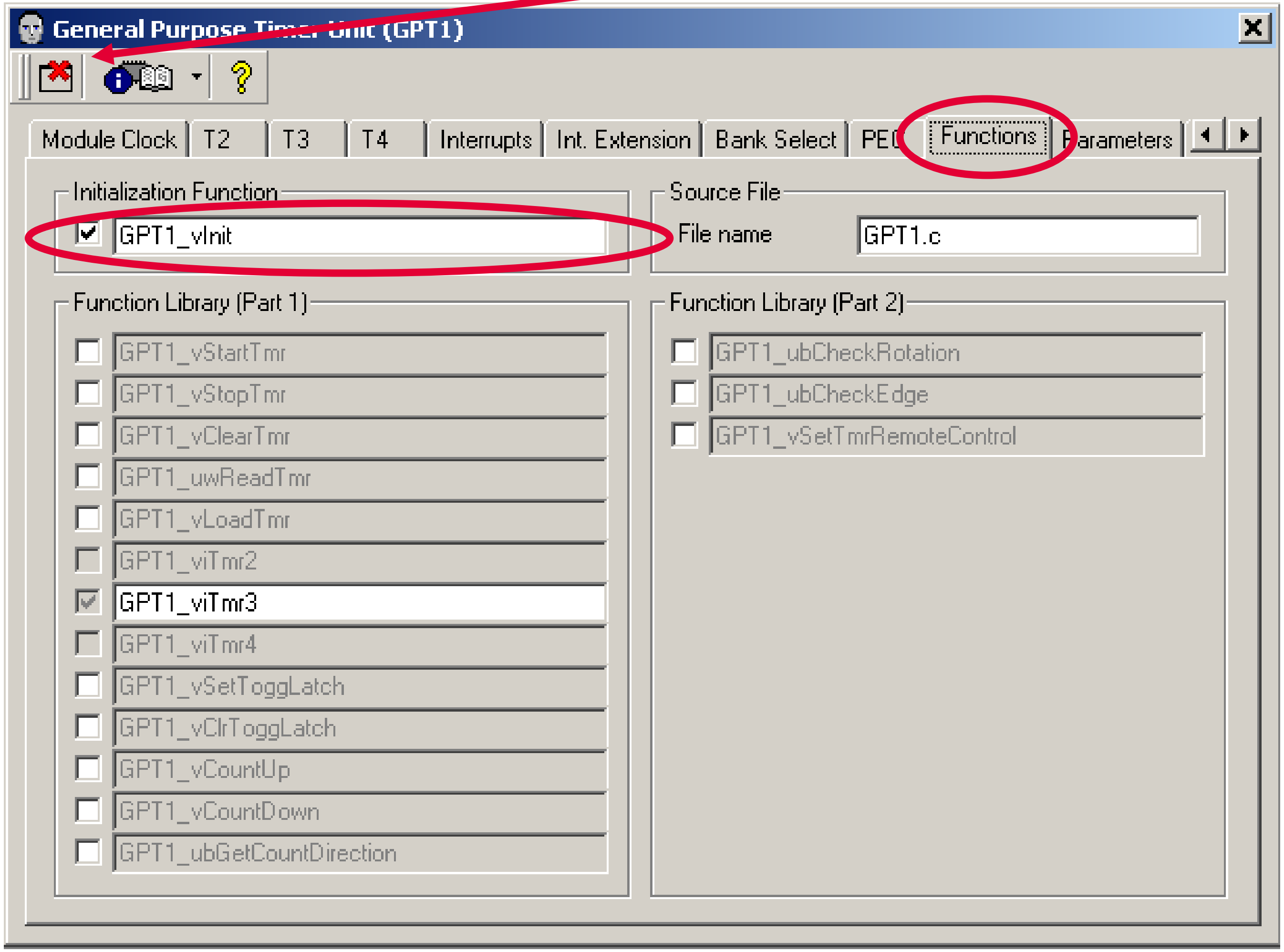
	Group 0	Group 1	Group 2	Group 3
Level 15				
Level 14				
Level 13				
Level 12				
Level 11				
Level 10				
Level 9				
Level 8	GPT1 T3 INT			
Level 7				
Level 6				
Level 5				
Level 4				
Level 3				
Level 2				
Level 1				

Level 0 (non interrupting)

Note: To change the level and the group of an interrupt source, click on it, drag it to its new position and drop it.
To set an interrupt source to the non interrupting level (Level 0) click on it, drag it to the 'Level 0' list and drop it.

Getting Started 1 - DAvE Configurations GPT1 settings

- Functions settings and close GPT1 window



Initialization Function

☑ GPT1_vInit

Source File

File name: GPT1.c

Function Library (Part 1)

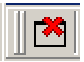
- ☐ GPT1_vStartTmr
- ☐ GPT1_vStopTmr
- ☐ GPT1_vClearTmr
- ☐ GPT1_uwReadTmr
- ☐ GPT1_vLoadTmr
- ☐ GPT1_viTmr2
- ☑ GPT1_viTmr3
- ☐ GPT1_viTmr4
- ☐ GPT1_vSetToggLatch
- ☐ GPT1_vClrToggLatch
- ☐ GPT1_vCountUp
- ☐ GPT1_vCountDown
- ☐ GPT1_ubGetCountDirection

Function Library (Part 2)

- ☐ GPT1_ubCheckRotation
- ☐ GPT1_ubCheckEdge
- ☐ GPT1_vSetTmrRemoteControl

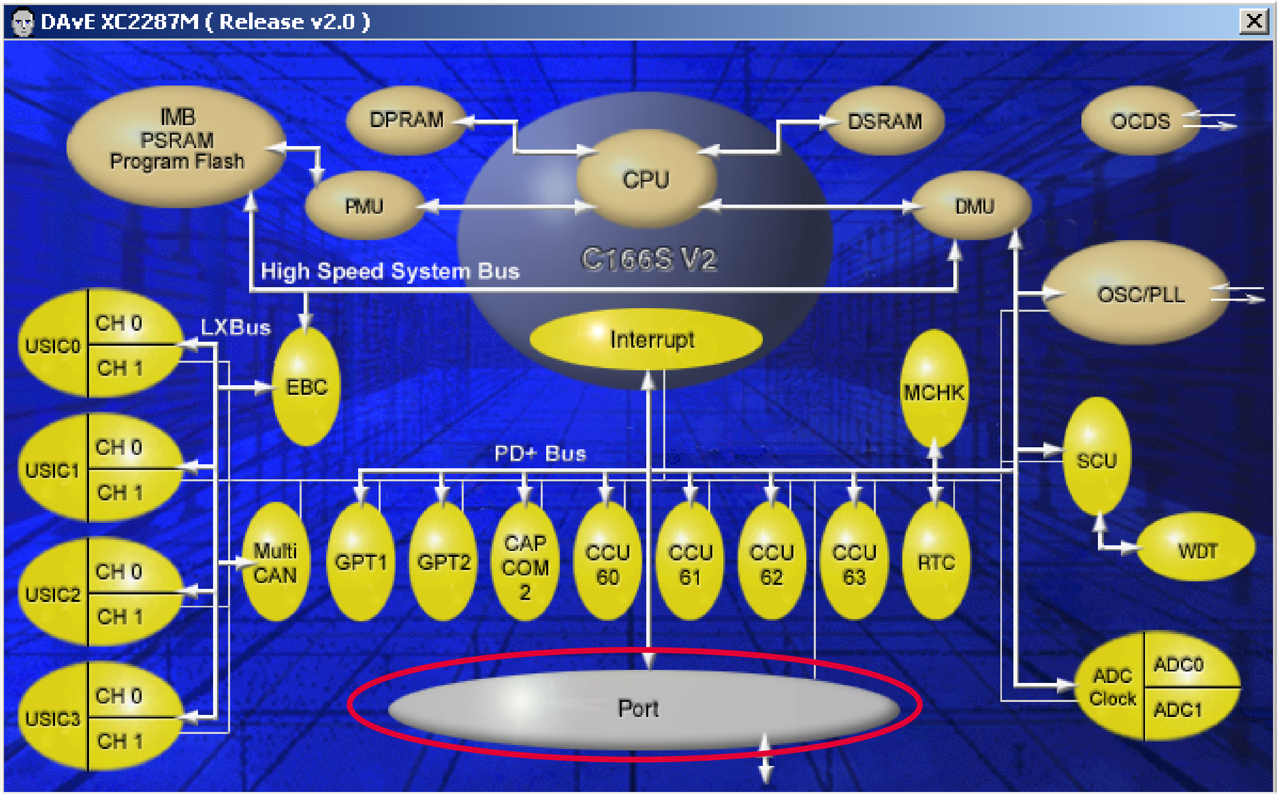
Getting Started 1 - DAvE Configurations
Port settings

- **Port Settings**
  - □ LED connected to P10.0
  - □ Use P10.0 as general output
  - □ Include port setup function
  - □ Include port toggle function

  - □ Close port setup window 

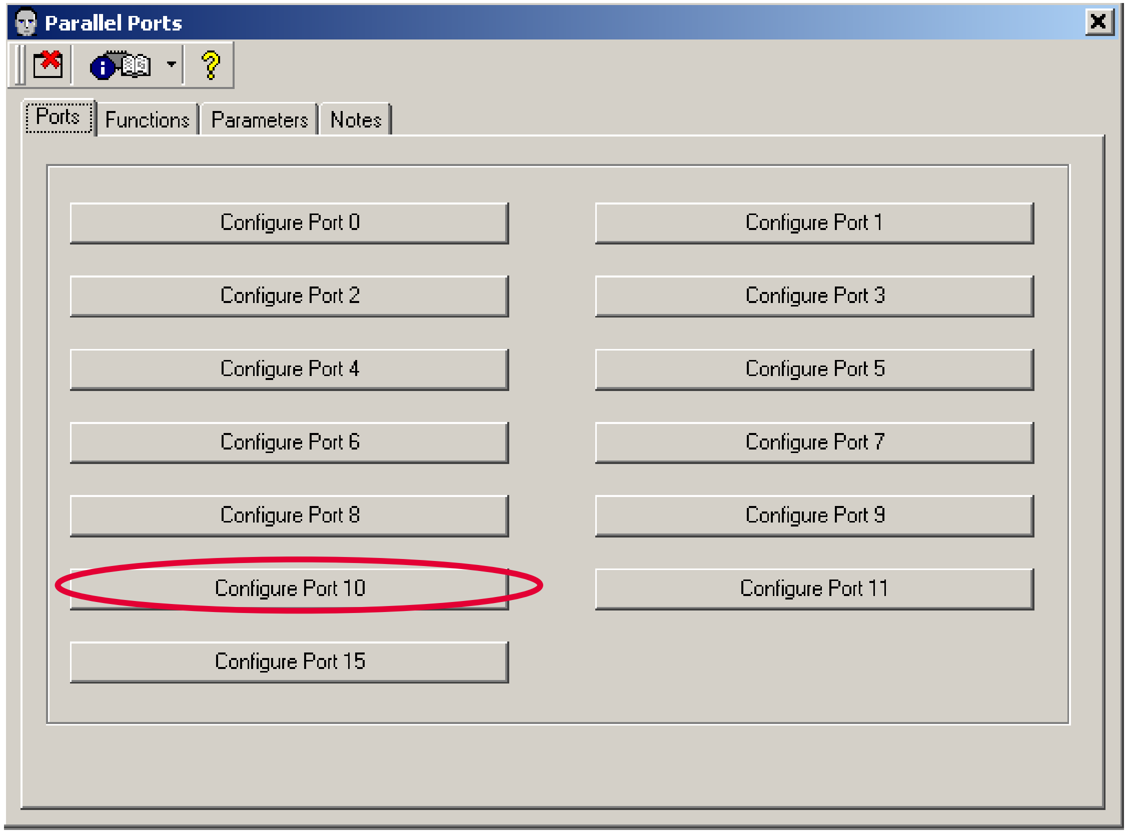
Getting Started 1 - DAvE Configurations
Port settings

- **Port Settings**
  - **Port:**
    - Click on



Getting Started 1 - DAvE Configurations
Port settings

- **Port Settings**
  - **Ports:**
    - Configure Port 10

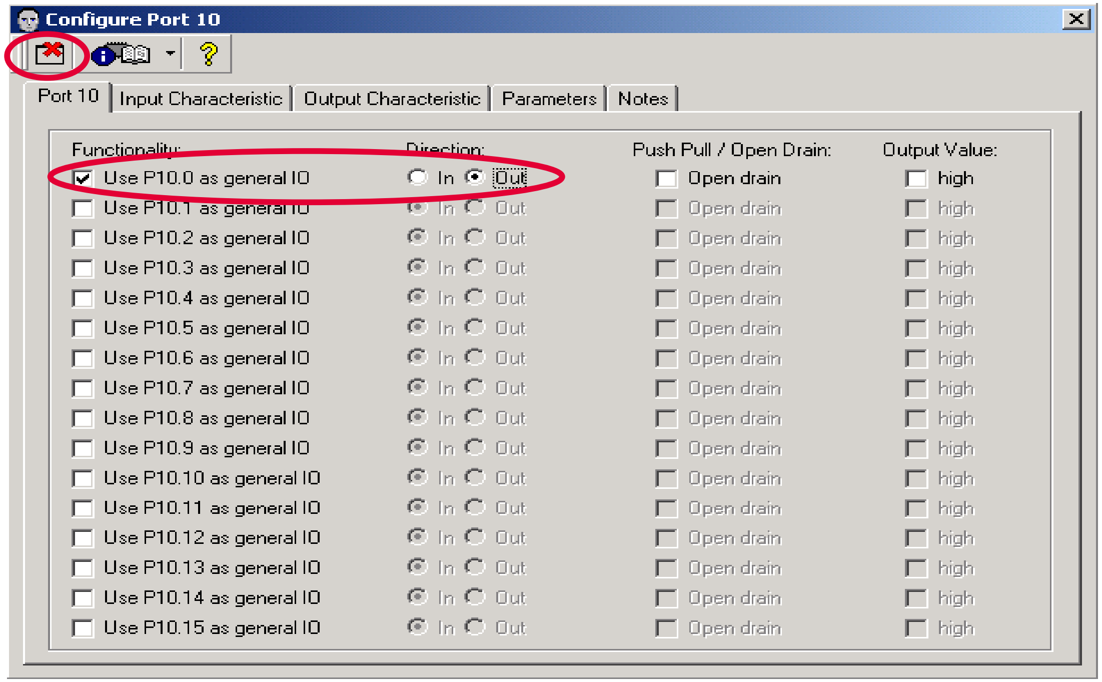


Getting Started 1 - DAvE Configurations
Port settings

Configure Port 10

Port 10:

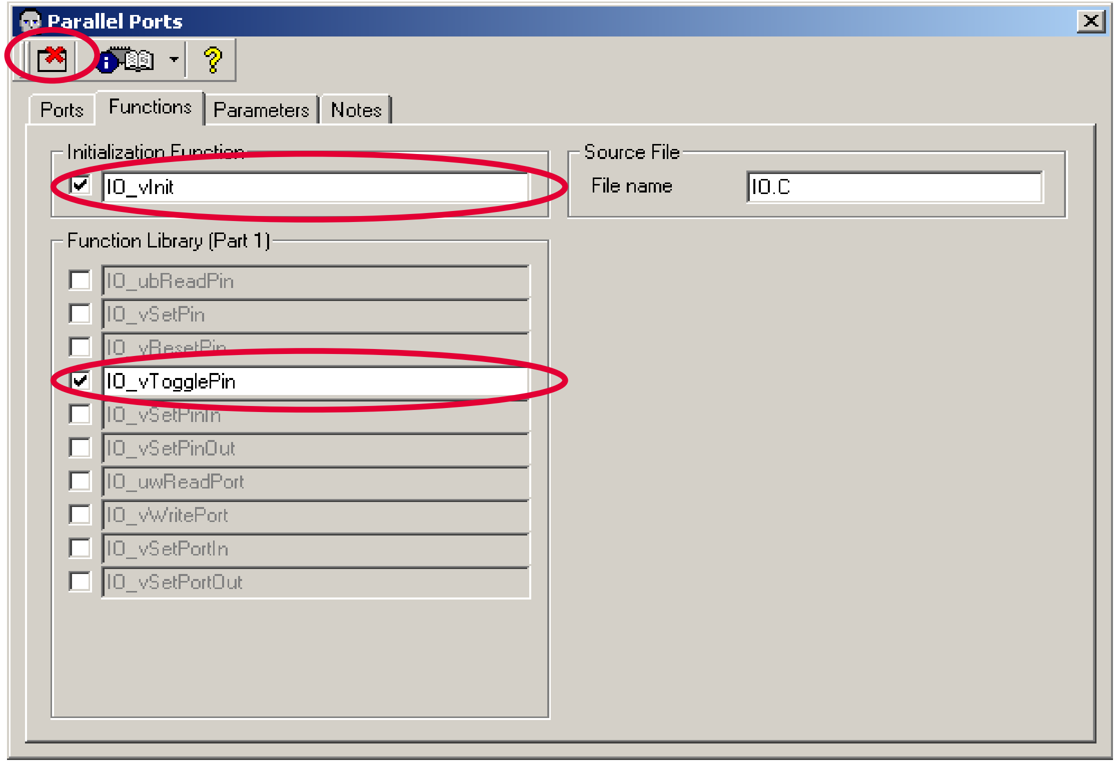
- Use P10.0 as general IO
- Set Direction to Out
- Close the window



Functionality:	Direction:	Push Pull / Open Drain:	Output Value:
☑ Use P10.0 as general IO	○ In ● Out	☐ Open drain	☐ high
☐ Use P10.1 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.2 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.3 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.4 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.5 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.6 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.7 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.8 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.9 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.10 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.11 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.12 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.13 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.14 as general IO	● In ○ Out	☐ Open drain	☐ high
☐ Use P10.15 as general IO	● In ○ Out	☐ Open drain	☐ high

Getting Started 1 - DAvE Configurations
Port settings

- **Parallel Ports**
  - **Functions:**
    - Include 'IO_vInit'
    - Include 'IO_vTogglePin'



Initialization Function
☑ IO_vInit

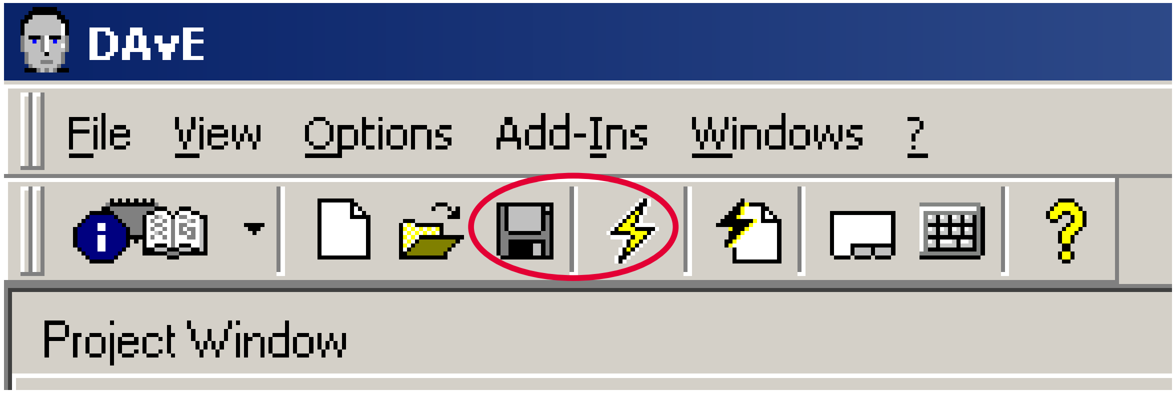
Source File
File name: IO.C

Function Library (Part 1)
☐ IO_ubReadPin
☐ IO_vSetPin
☐ IO_vResetPin
☑ IO_vTogglePin
☐ IO_vSetPinIn
☐ IO_vSetPinOut
☐ IO_uwReadPort
☐ IO_vWritePort
☐ IO_vSetPortIn
☐ IO_vSetPortOut

Getting Started 1 - DAvE Configurations
Save DAvE Project and Generate Code

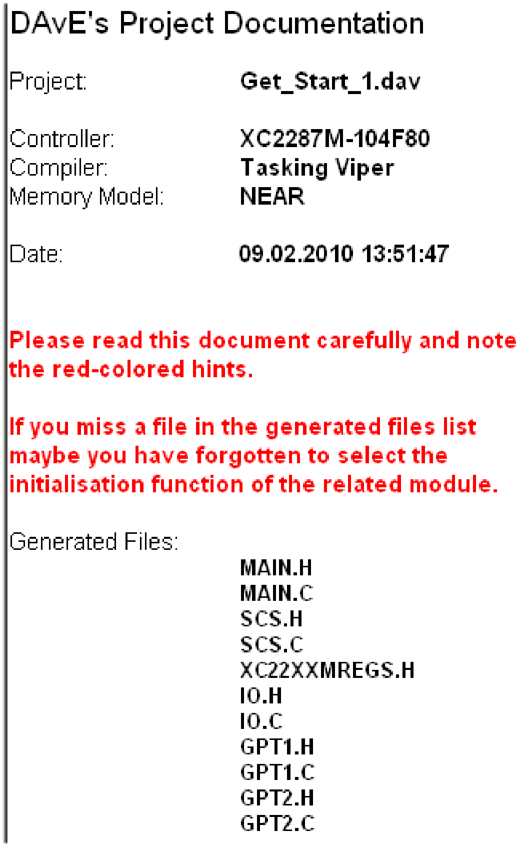
- Save your DAvE Project File 💾
- Generated Code ⚡

1



Getting Started 1 - DAvE Configurations Generated Code

- DAvE generated code files are
  - 'MAIN.c', 'MAIN.h'
  - 'SCS.c', 'SCS.h'
  - 'IO.c', 'IO.h'
  - 'GPT1.c', 'GPT1.h'
  - 'XC22xxREGS.h'



DAvE's Project Documentation

Project: **Get_Start_1.dav**

Controller: **XC2287M-104F80**
Compiler: **Tasking Viper**
Memory Model: **NEAR**

Date: **09.02.2010 13:51:47**

Please read this document carefully and note the red-colored hints.

If you miss a file in the generated files list maybe you have forgotten to select the initialisation function of the related module.

Generated Files:

MAIN.H
MAIN.C
SCS.H
SCS.C
XC22XXMREGS.H
IO.H
IO.C
GPT1.H
GPT1.C
GPT2.H
GPT2.C

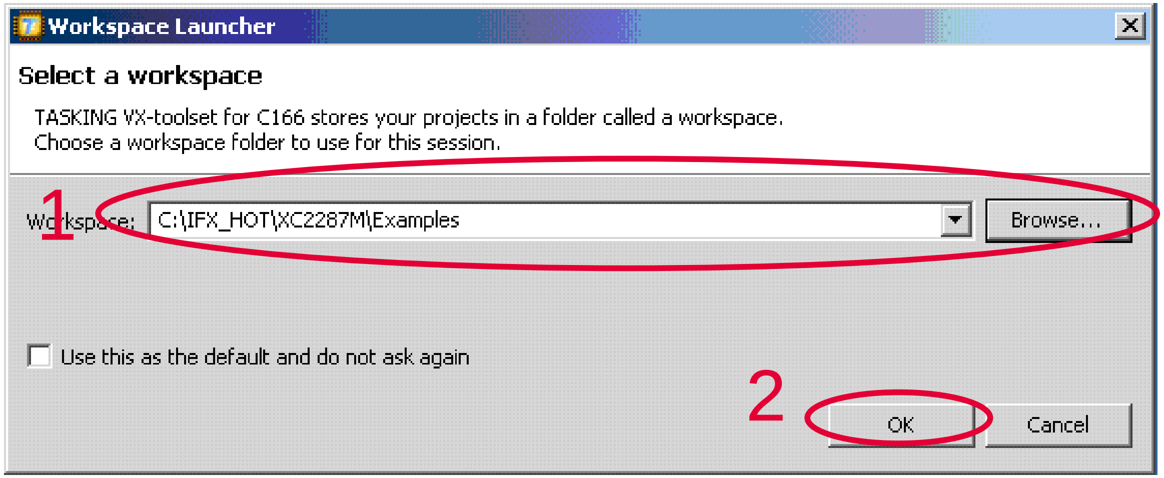
- In general:
  - if the included function is a macro it is included in the '.h' file
  - if the included function is a function it is included in the '.c' file

Getting Started 1– Tasking VX Toolset

- Start Tasking Viper Tool Set

VX V2.2r1

- Open Project Work Space
  - Browse to **“c:\IFX_HOT\XC2287M\Examples”**
  - Click ‘OK’



Workspace Launcher

Select a workspace

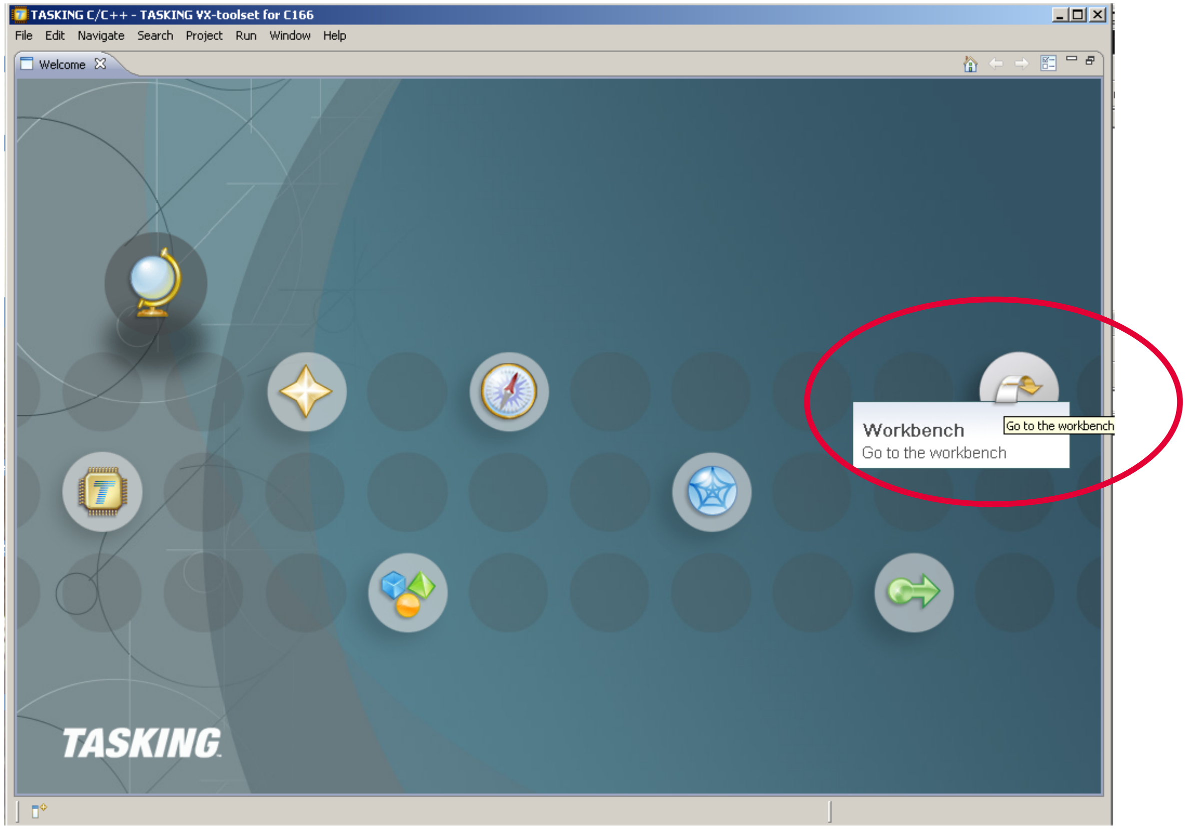
TASKING VX-toolset for C166 stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

1 Workspace: C:\IFX_HOT\XC2287M\Examples

□ Use this as the default and do not ask again

Getting Started 1– Tasking VX Toolset

- Create New Project
  - Click on Workbench (if not already there...)

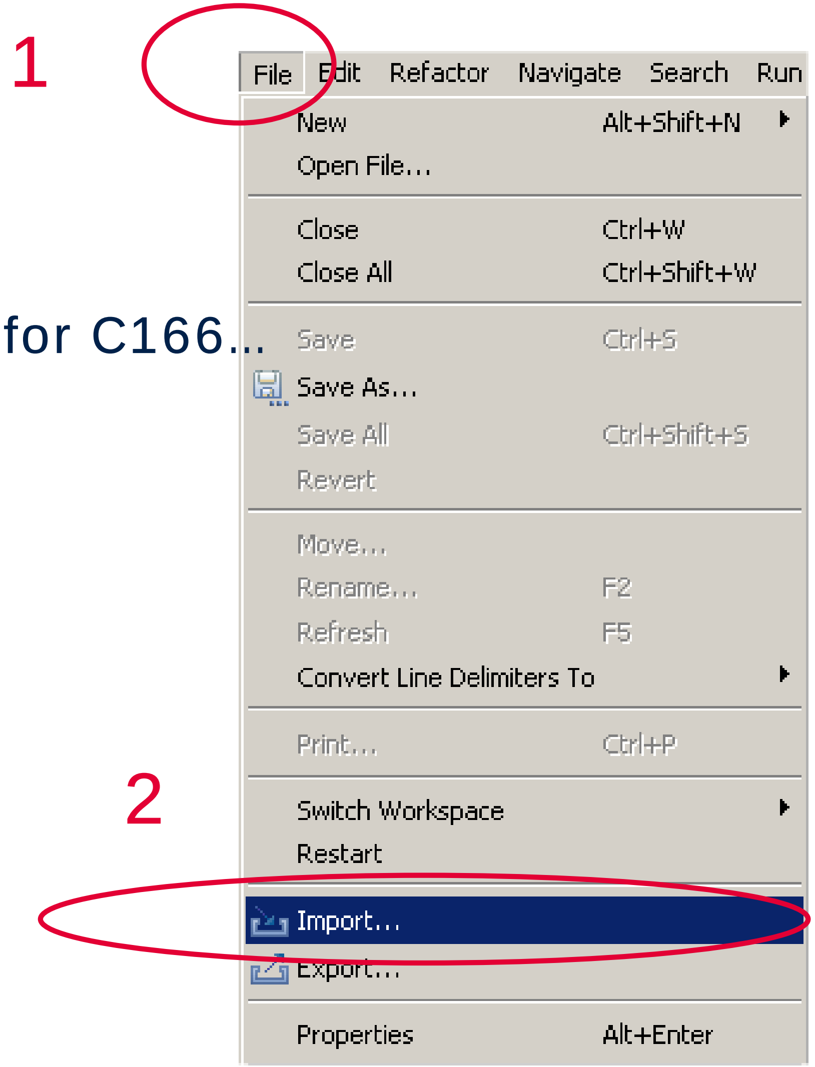

TASKING C/C++ - TASKING VX-toolset for C166
File Edit Navigate Search Project Run Window Help
Welcome
TASKING

Workbench
Go to the workbench

Go to the workbench

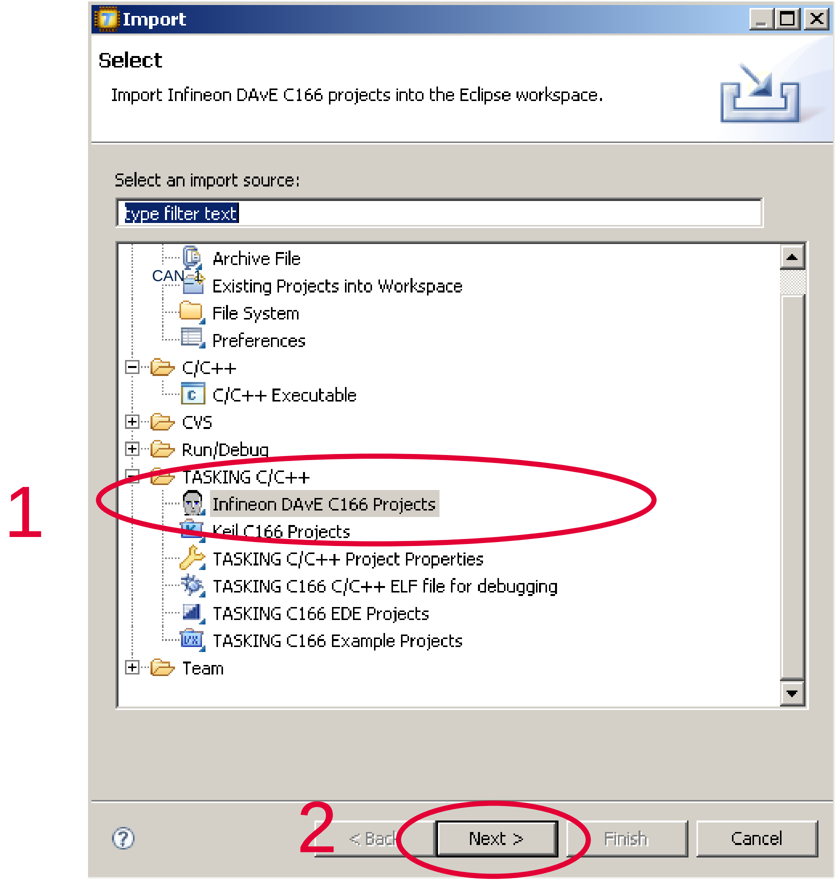
Getting Started 1– Tasking VX Toolset

- Import DAVE Project
  - Click on File -> Import
  - Select Tasking VX-toolset for C166…
  - Click ‘OK’



Getting Started 1– Tasking VX Toolset

- Import DAVE Project
  - Click `Infineon DAvE C166 Project´
  - Click ‘Next’



Import

Select

Import Infineon DAvE C166 projects into the Eclipse workspace.

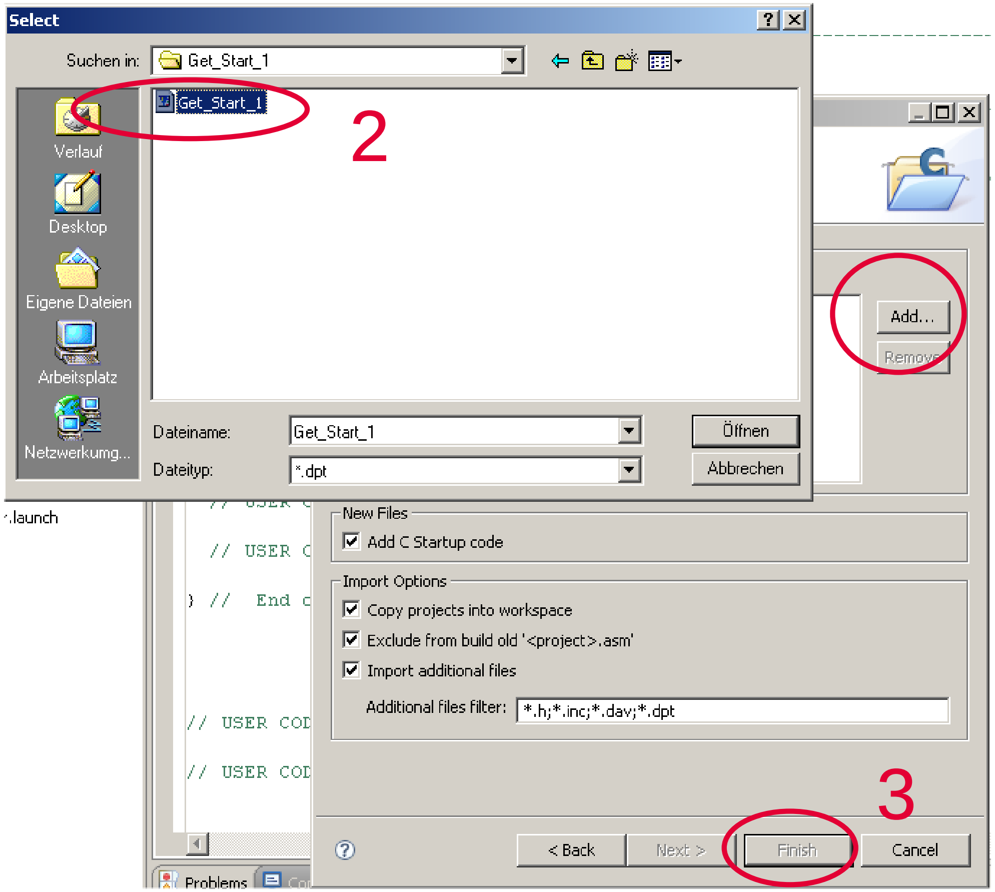
Select an import source:

type filter text

Archive File
CAN
Existing Projects into Workspace
File System
Preferences
C/C++
C/C++ Executable
CVS
Run/Debug
TASKING C/C++
Infineon DAvE C166 Projects
Keil C166 Projects
TASKING C/C++ Project Properties
TASKING C166 C/C++ ELF file for debugging
TASKING C166 EDE Projects
TASKING C166 Example Projects
Team
1
2
< Back
Next >
Finish
Cancel

Getting Started 1– Tasking VX Toolset

- Import DAvE Project
  - Add Dave Project `Get_Start_1´
  - Click `Finish´



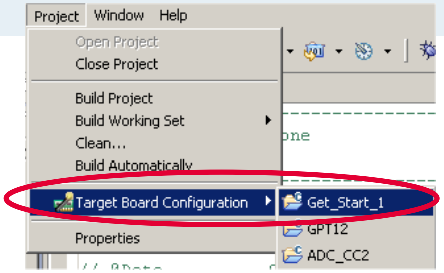
New Files
☑ Add C Startup code

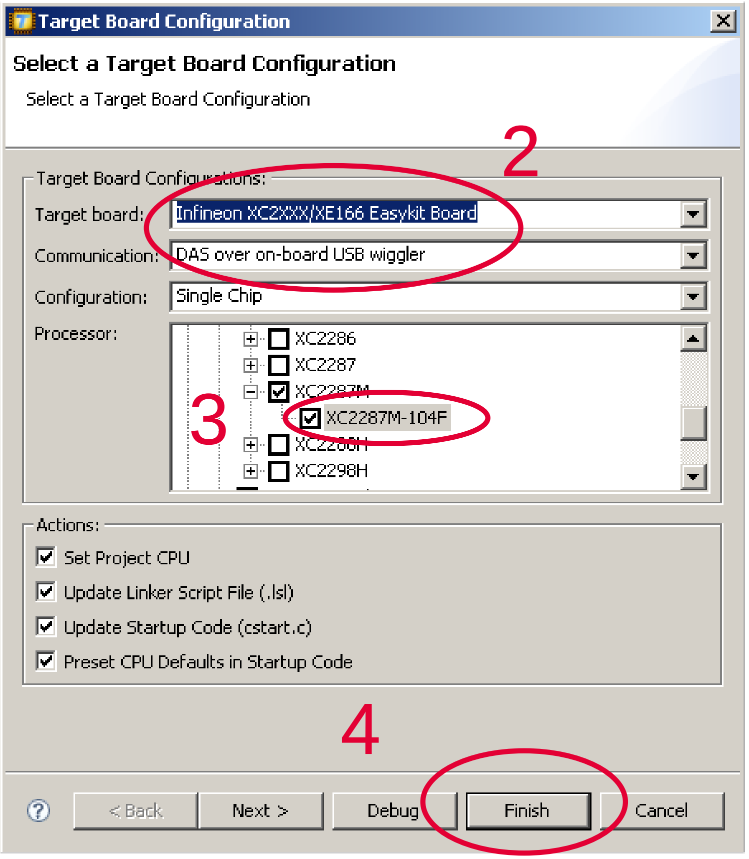
Import Options
☑ Copy projects into workspace
☑ Exclude from build old '<project>.asm'
☑ Import additional files
Additional files filter: *.h;*.inc;*.dav;*.dpt

Getting Started 1– Tasking VX Toolset

- Configure Target Board
  - □ Select the project in the navigator
  - □ Select 'Project/Target Board Configuration'
  - □ Select 'Infineon XC2000/XE166 Easykit Board'
  - □ Choose `XC2287M-104F´
  - □ Click `Finish´

1





Target Board Configuration

Select a Target Board Configuration

Select a Target Board Configuration

Target Board Configurations:

Target board: Infineon XC2XXX/XE166 Easykit Board

Communication: DAS over on-board USB wiggler

Configuration: Single Chip

Processor:
- □ XC2286
- □ XC2287
- ☑ XC2287M
  - ☑ XC2287M-104F
- □ XC2286H
- □ XC2298H

Actions:
- ☑ Set Project CPU
- ☑ Update Linker Script File (.lsl)
- ☑ Update Startup Code (cstart.c)
- ☑ Preset CPU Defaults in Startup Code

Getting Started 1– Tasking VX Toolset

- Software Hint
  - □ DAvE doesn't change code that is inserted in the 'USER CODE' sections if you let DAvE regenerate the code. Therefore, whenever adding code to the generated code, write it into a 'USER CODE' section.

    The code you really have to add looks like this:

```
while(1)
{
// USER CODE BEGIN (Main,4)

BlinkLED();

// USER CODE END
}
```

Getting Started 1 - Complete code
Edit File ‘GPT1.C’

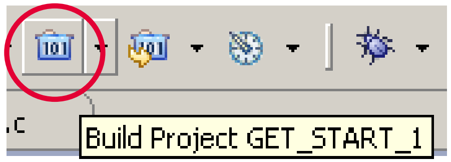
- In the ISR function 'void GPT1_viTmr3(void)'
(almost at the end)

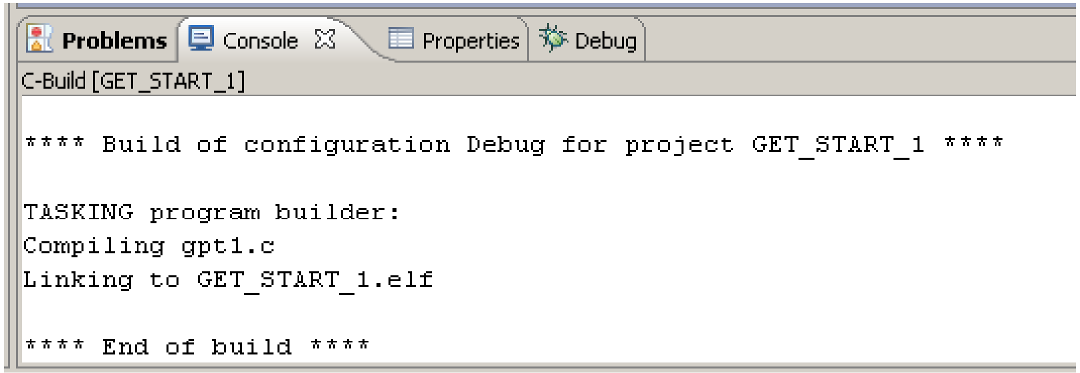
```
_interrupt(T3INT)  void GPT1_viTmr3(void)
{
  // USER CODE BEGIN (Tmr3,2)

  IO_vTogglePin(IO_P10_0);

  // USER CODE END


  // USER CODE BEGIN (Tmr3,5)

  // USER CODE END

} //  End of function GPT1_viTmr3
```

Getting Started 1– Tasking VX Toolset Build Project

- Click on ‘Build Project GET_START_1’

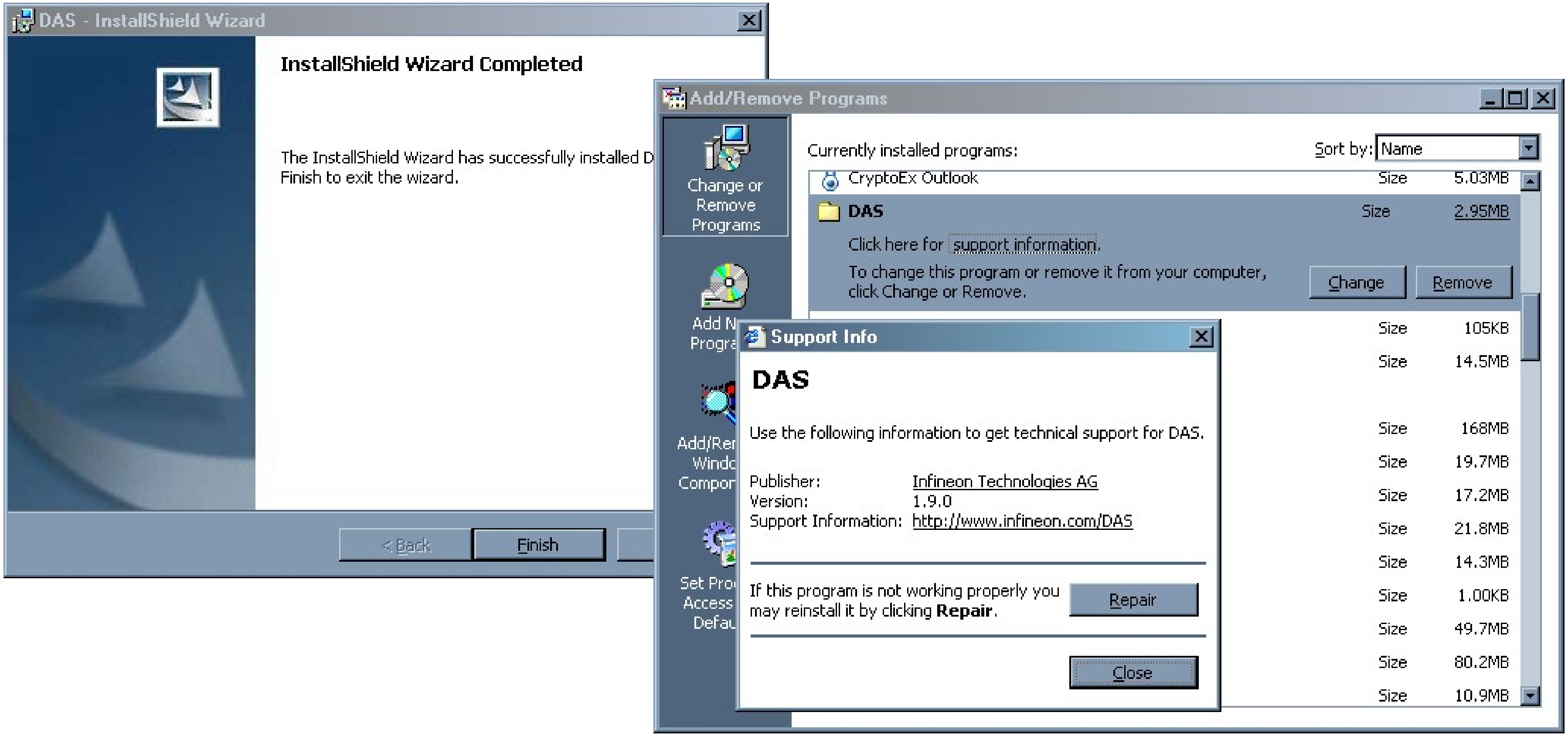
1





```
**** Build of configuration Debug for project GET_START_1 ****

TASKING program builder:
Compiling gpt1.c
Linking to GET_START_1.elf

**** End of build ****
```

Getting Started 1 - Device Access Server
Check for the latest DAS version



DAS - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed D
Finish to exit the wizard.

Add/Remove Programs

Currently installed programs: Sort by: Name

CryptoEx Outlook Size 5.03MB

DAS Size 2.95MB

Click here for support information.

To change this program or remove it from your computer, click Change or Remove.

Size 105KB
Size 14.5MB
Size 168MB
Size 19.7MB
Size 17.2MB
Size 21.8MB
Size 14.3MB
Size 1.00KB
Size 49.7MB
Size 80.2MB
Size 10.9MB

Support Info

DAS

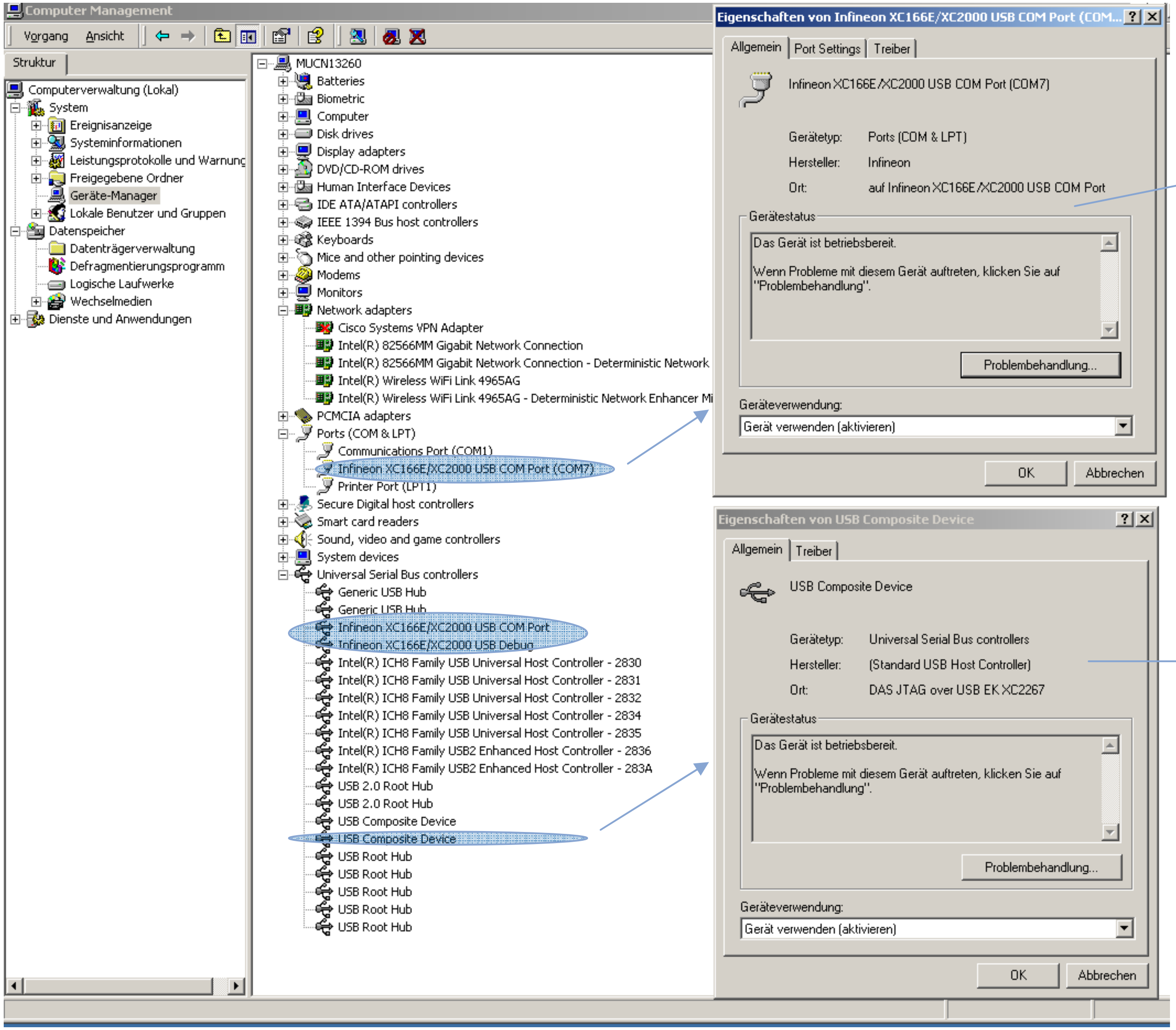
Use the following information to get technical support for DAS.

Publisher: Infineon Technologies AG
Version: 1.9.0
Support Information: http://www.infineon.com/DAS

If this program is not working properly you may reinstall it by clicking **Repair**.

Note: It is recommended to use the latest DAS version.
Download the latest version at www.infineon.com\DAS

Getting Started 1 - Device Access Server
1.) Checking USB connections



- Computerverwaltung (Lokal)
  - System
    - Ereignisanzeige
    - Systeminformationen
    - Leistungsprotokolle und Warnung
    - Freigegebene Ordner
    - Geräte-Manager
    - Lokale Benutzer und Gruppen
  - Datenspeicher
    - Datenträgerverwaltung
    - Defragmentierungsprogramm
    - Logische Laufwerke
    - Wechselmedien
  - Dienste und Anwendungen

- MUCN13260
  - Batteries
  - Biometric
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - IEEE 1394 Bus host controllers
  - Keyboards
  - Mice and other pointing devices
  - Modems
  - Monitors
  - Network adapters
    - Cisco Systems VPN Adapter
    - Intel(R) 82566MM Gigabit Network Connection
    - Intel(R) 82566MM Gigabit Network Connection - Deterministic Network
    - Intel(R) Wireless WiFi Link 4965AG
    - Intel(R) Wireless WiFi Link 4965AG - Deterministic Network Enhancer Mi
  - PCMCIA adapters
  - Ports (COM & LPT)
    - Communications Port (COM1)
    - Infineon XC166E/XC2000 USB COM Port (COM7)
    - Printer Port (LPT1)
  - Secure Digital host controllers
  - Smart card readers
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers
    - Generic USB Hub
    - Generic USB Hub
    - Infineon XC166E/XC2000 USB COM Port
    - Infineon XC166E/XC2000 USB Debug
    - Intel(R) ICH8 Family USB Universal Host Controller - 2830
    - Intel(R) ICH8 Family USB Universal Host Controller - 2831
    - Intel(R) ICH8 Family USB Universal Host Controller - 2832
    - Intel(R) ICH8 Family USB Universal Host Controller - 2834
    - Intel(R) ICH8 Family USB Universal Host Controller - 2835
    - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836
    - Intel(R) ICH8 Family USB2 Enhanced Host Controller - 283A
    - USB 2.0 Root Hub
    - USB 2.0 Root Hub
    - USB Composite Device
    - USB Composite Device
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub
    - USB Root Hub

Eigenschaften von Infineon XC166E/XC2000 USB COM Port (COM...

Infineon XC166E/XC2000 USB COM Port (COM7)

Gerätetyp: Ports (COM & LPT)

Hersteller: Infineon

Ort: auf Infineon XC166E/XC2000 USB COM Port

Gerätestatus

Das Gerät ist betriebsbereit.

Wenn Probleme mit diesem Gerät auftreten, klicken Sie auf "Problembehandlung".

Geräteverwendung:

Gerät verwenden (aktivieren)

Eigenschaften von USB Composite Device

USB Composite Device

Gerätetyp: Universal Serial Bus controllers

Hersteller: (Standard USB Host Controller)

Ort: DAS JTAG over USB EK XC2267

Gerätestatus

Das Gerät ist betriebsbereit.

Wenn Probleme mit diesem Gerät auftreten, klicken Sie auf "Problembehandlung".

Geräteverwendung:

Gerät verwenden (aktivieren)

This gets identified only when COM port is used

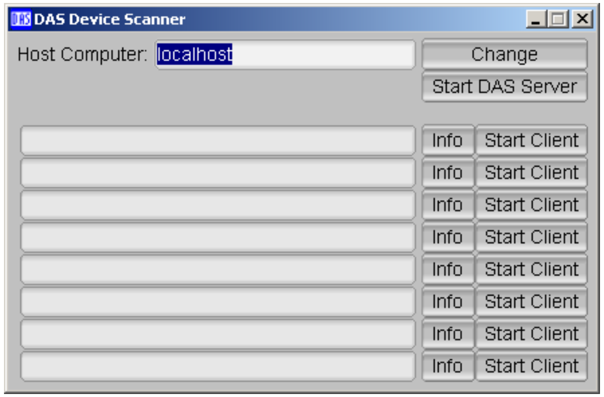
- Via the USB interface on the Easykit with FTDI chip

The DAS JTAG composite device gets identified

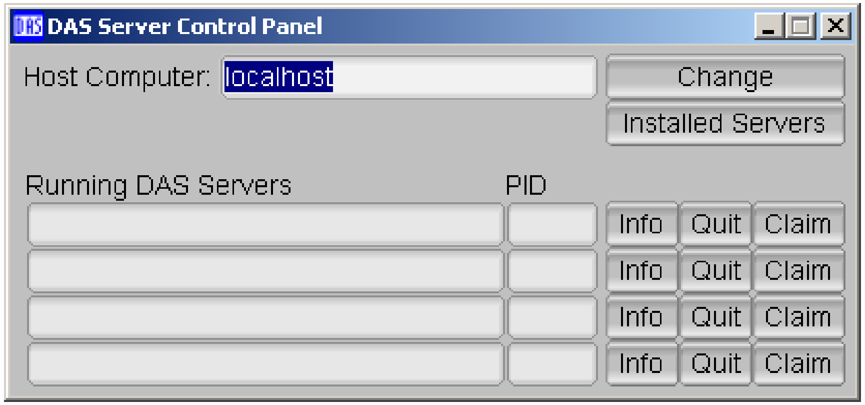
- When miniWiggler is connected
- When USB Wiggler Box is connected
- Via the USB interface on the Easykits with FTDI chip

Getting Started 1 - Device Access Server
2.) Check DAS status

1. Start DAS device scanner
2. Start DAS Server Control panel

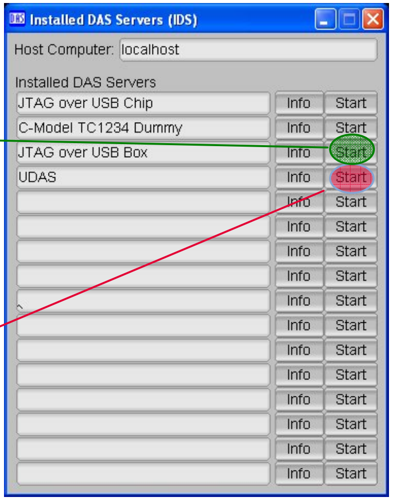

DAS Device Scanner

Host Computer: localhost


DAS Server Control Panel

Host Computer: localhost

3. If DAS device scanner does not show any device, start the appropriate DAS server


Installed DAS Servers (IDS)

Host Computer: localhost

Installed DAS Servers

JTAG over USB Chip

C-Model TC1234 Dummy

JTAG over USB Box

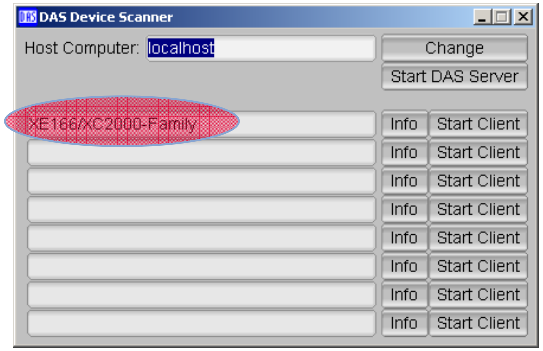
UDAS

Incase you are connected via the USB Wiggler box, then start „JTAG over USB Box“

Incase you are connected via the FTDI chip or mini wiggler, then start „UDAS“

Getting Started 1 - Device Access Server
3.) Starting the servers manually

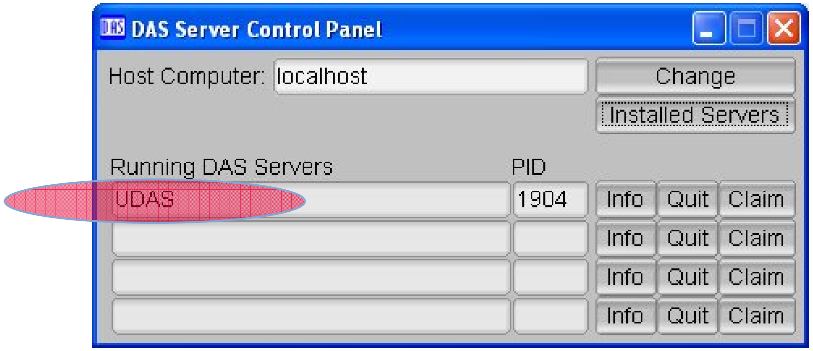
4. Incase „UDAS“ server is started and XC2000 easykit is connected via on-chip FTDI or via separate miniWiggler, following status changes could be noted



DAS Device Scanner

Host Computer: localhost

XE166/XC2000-Family

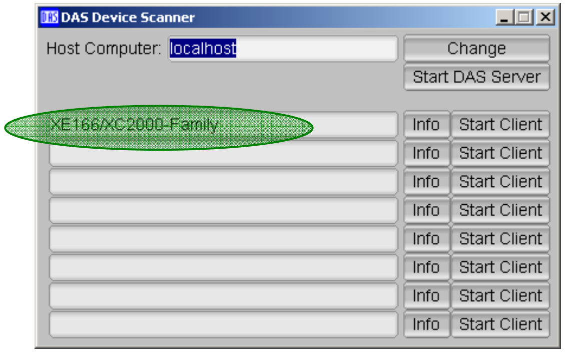


DAS Server Control Panel

Host Computer: localhost

Running DAS Servers	PID
UDAS	1904

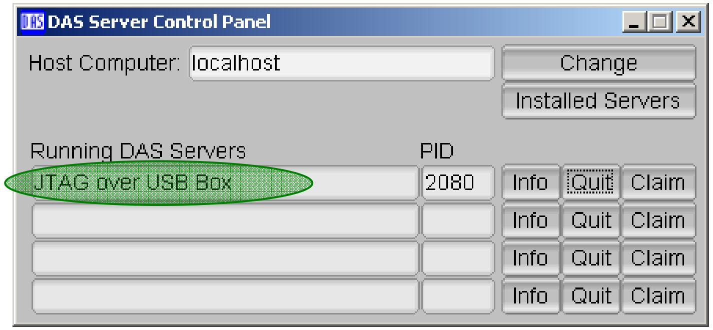
5. Incase „JTAG over USB Box“ server is started and XC2000 starter kit is connected via Wiggler box, following status changes could be noted



DAS Device Scanner

Host Computer: localhost

XE166/XC2000-Family



DAS Server Control Panel

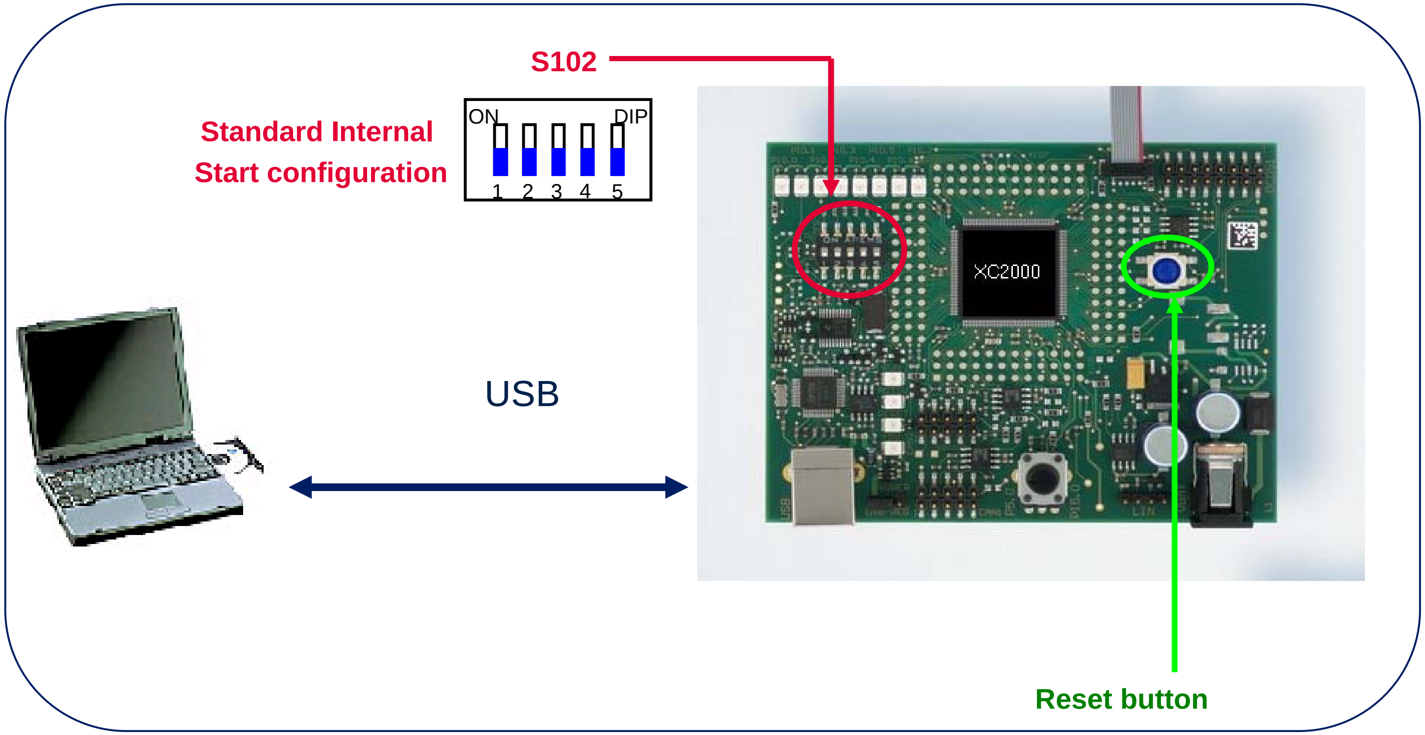
Host Computer: localhost

Running DAS Servers	PID
JTAG over USB Box	2080

Getting Started 1
Connect XC2287M Board

Internal start from Flash

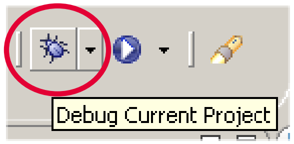
- □ Connect XC2287M Board to PC
- □ Check the DIP switch settings, S102: **OFF-OFF-OFF-OFF-OFF**
- □ Reset the board (press the reset button)



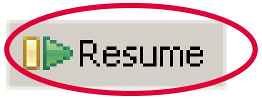
Getting Started 1– Tasking VX Toolset Run Debugger

1

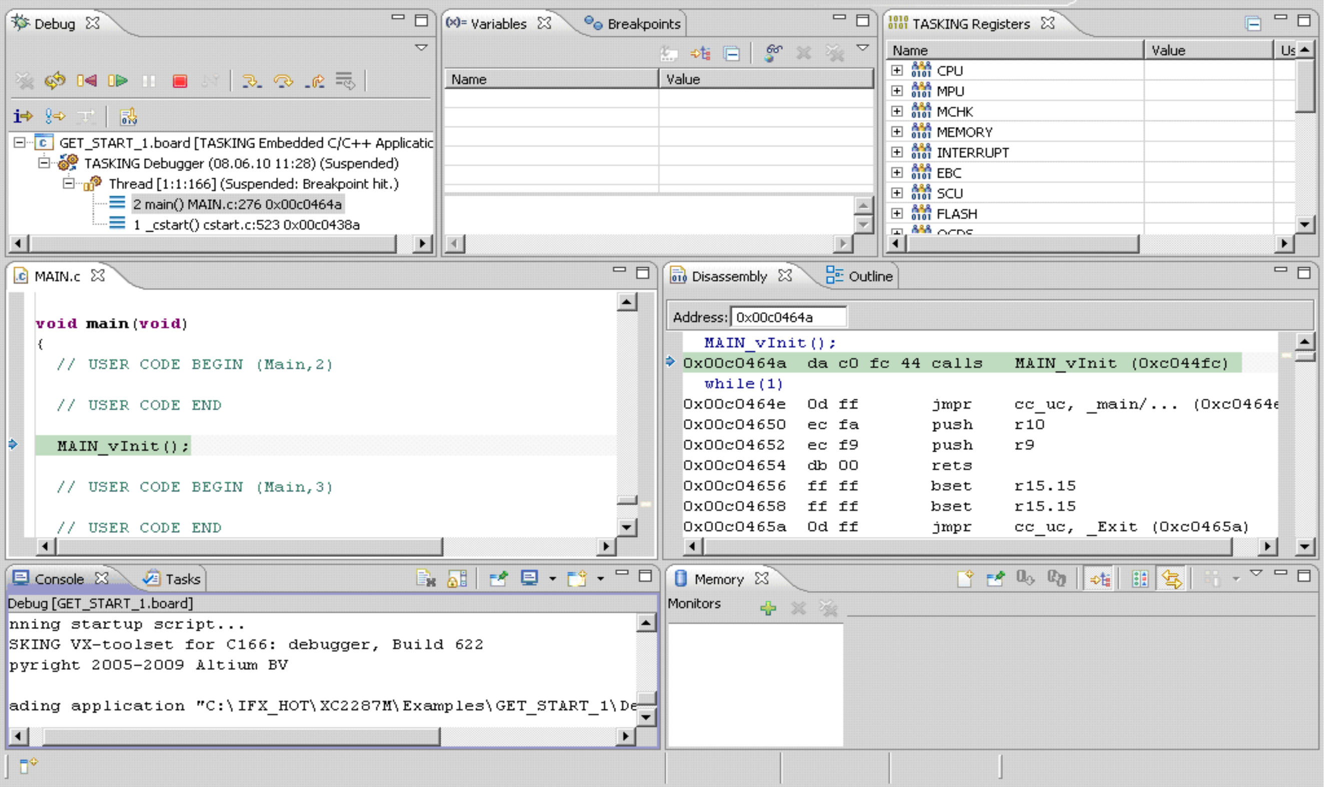
- Click on



2



- Click on ‘Resume’ and start program



```
GET_START_1.board [TASKING Embedded C/C++ Applicatic
  TASKING Debugger (08.06.10 11:28) (Suspended)
    Thread [1:1:166] (Suspended: Breakpoint hit.)
      2 main() MAIN.c:276 0x00c0464a
      1 _cstart() cstart.c:523 0x00c0438a
```

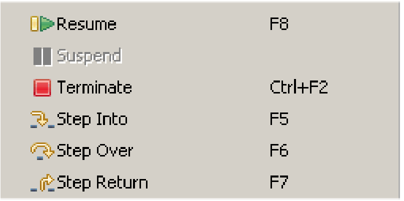
```
void main(void)
{
  // USER CODE BEGIN (Main,2)

  // USER CODE END

  MAIN_vInit();

  // USER CODE BEGIN (Main,3)

  // USER CODE END
```

Address: 0x00c0464a

```
  MAIN_vInit();
0x00c0464a  da c0 fc 44 calls   MAIN_vInit (0xc044fc)
  while(1)
0x00c0464e  0d ff       jmpr    cc_uc, _main/... (0xc0464e
0x00c04650  ec fa       push    r10
0x00c04652  ec f9       push    r9
0x00c04654  db 00       rets
0x00c04656  ff ff       bset    r15.15
0x00c04658  ff ff       bset    r15.15
0x00c0465a  0d ff       jmpr    cc_uc, _Exit (0xc0465a)
```

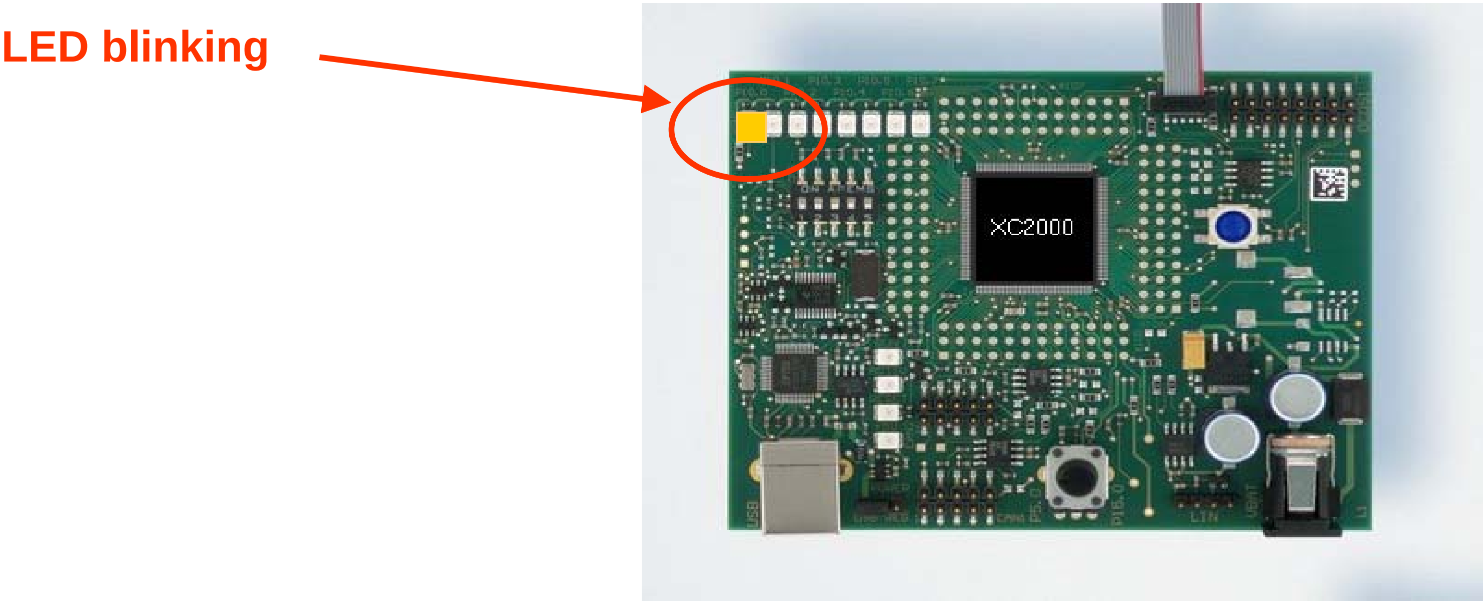
Debug [GET_START_1.board]

```
nning startup script...
SKING VX-toolset for C166: debugger, Build 622
pyright 2005-2009 Altium BV

ading application "C:\IFX_HOT\XC2287M\Examples\GET_START_1\De
```



Getting Started 1
See Result

- The yellow LED will toggle



Getting Started 1
Verification

- Verification: (stop the program)
  - Go to Tasking debugger
  - Click on ‘RESET’
  - ⇒ LED is no more toggling; stays constant off

Never stop thinking