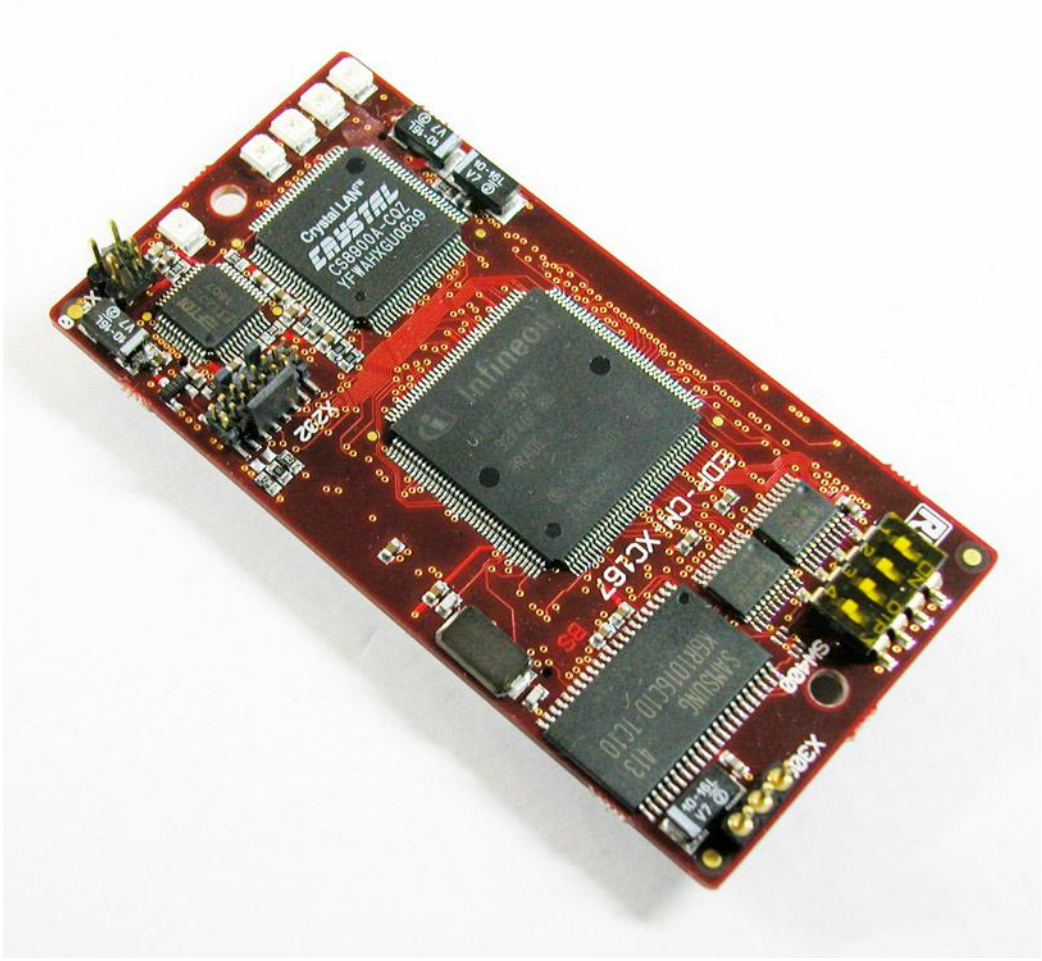
Getting Started Guide
RS-EDP & XC167 CPU Module

Version 2

10th June 2010



Contents

1. Introduction

To get the most out of the RS-EDP platform it's important to understand the concept of the RS-EDP system correctly. This is detailed in the user manuals for the Base Boards. The manual is called

RS EDP-BB-SystemBaseBoard User Manual Vx.docx.

The base boards come in both 2 position format and 4 position format and share a common user manual. Please read this manual to get an understanding of the system.

Each of the CPU Modules (CM or Command module) and Application Modules (AM) has its own user manual, so again these documents must be read to get an understanding on how to use the modules.

Each of the boards comes with a selection of software to fully exercise the RS-EDP Application Modules and the peripherals available on the MCU device.

In an RS-EDP system there is usually one Command Module (CM) and one or more Applications Modules (AM) plugged in to the Base Board (BB). The XC167 module has been designed as the CPU or Command Module for the system.

The 'Command Module' (also referred to as a CPU Module) in a system, dictates whether the whole system is a 3.3V one or a 5.0V one. This CPU module uses a 5.0V microprocessor and so the board is configured as such. If the user is unsure he can check the Vcc_CM signal on the break out connector on the Base Board to determine what voltage the system is.

There are 144 pins on the MCU and these are connected via various link options to the Base Board. The Base Board (BB) then provides these signals to the Application Modules thereby allowing the CPU Modules to communicate with the Application Modules.

As many of the CPU pins have more than one function it can make the mapping of the connections rather complex so there are additional support documents available to help you with this. The first is the Pin Allocation Spread Sheet. One spread sheet is available for each of the CPU Modules. The one for the NXP family of CPU Modules is called...

Pin Allocation - 144 pin XC167 Command Module RevXX.xlsx

The contents of this spreadsheet are also detailed in the user manual for the XC167 CPU Module.
This spreadsheet detailed which pins are mapped to the Base Board backplane and the various link options which need to be configured to connect them accordingly.

To get an appreciation of how the Application Modules are mapped to the backplane and how the CPU Module can connect to them, a Mapping Aid exists. The one for the XC167 module is called...

Mapping Aid RS-EDP - Complete – XC167 Rev.XX.pptx

This mapping aid is also present in the User Manual for the CPU Module and at a glance you can see what resources are required to get the best out of each Application Module.

The other useful documents you will need are the circuit diagram for the modules you want to connect with each other. These may be contained in the back of the user manual or available to download separately off the EDP web site.

So before you start to use the RS-EDP system make sure you have to hand the following documents.

- The Circuit Diagram of the modules you intend to use (in the user manual)
- The Base Board User Manual
- The XC167 CPU Module User Manual
- The Application Module User Manual
- The Pin Allocation Spreadsheet
- The Mapping Aid

In this getting started guide we will go through the process of getting the CPU Module up and running.

2. Development Tool Support

2.1 FTDI Based USB to JTAG Converter

On the XC167 CPU Modules is an FTDI chip which has flashed into it a USB to JTAG firmware set. The software allows a host PC to control the XC167 easily, and effectively removes the need for the user to have his own USB debugger. The USB end of this interface is bought down onto the Base Board and onto a mini USB connector. This miniUSB effectively allows the user to connect to a host PC and to run a debug session, including programming of the flash, without any additional debugger hardware tools. This means a software suite such as Keil uVision or Hitex HiTOP is all that is required to program and debug code.

The pre-programmed FTDI chip is present on all XC167 modules and allows the user to program and enter a debug session via HiTOP IDE. The version of HiTOP required is specifically designed for the RS-EDP platform. It is HiTOP53 for EDP XC167 and is available to download from the RS-EDP website. This software has a code size restriction on it, which allows the user to debug and program files up to this size. Larger files can be programmed but the debug information is lost.

HiTOP accesses the XC167 module via the miniUSB present on the Base Board. The base board routes the USB D+ and D- signals onto the XC167 CPU Module where the FTDI chip handles the interface to the JTAG connections. When the mini USB lead is present the FTDI chip is active and the XC167 device is under the control of the debugger. When the cable is unplugged, the FTDI chip is disconnected and the XC167 device runs in standalone mode.

The FTDI chip also has a spare RS232 virtual communications port on board, in additional to the USB to JTAG interface software. It is therefore possible to route some TTL level RS232 data via this chip therefore eliminating the requirement for a Communications Module, whilst debugging. Refer to the user manual for more information on these.

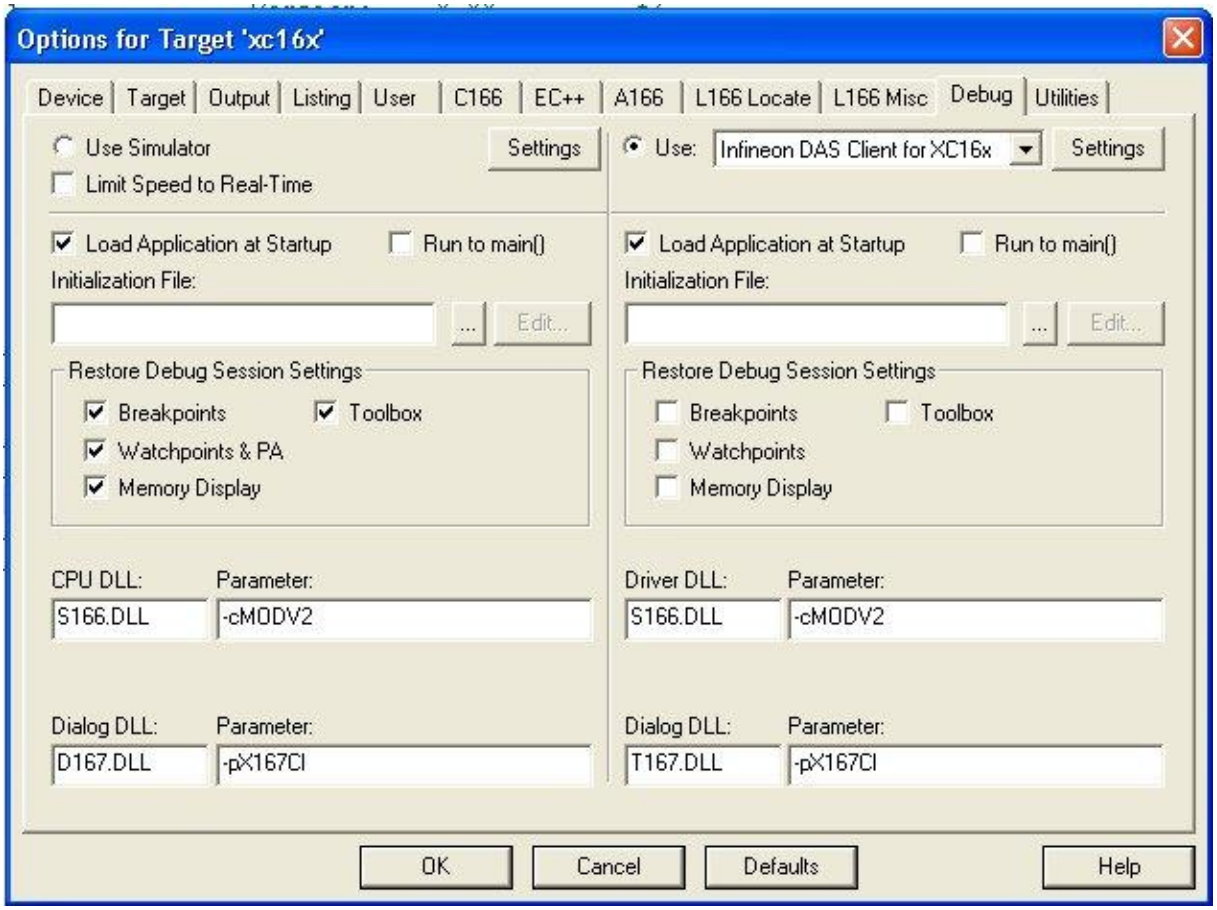
2.2 Keil uVision

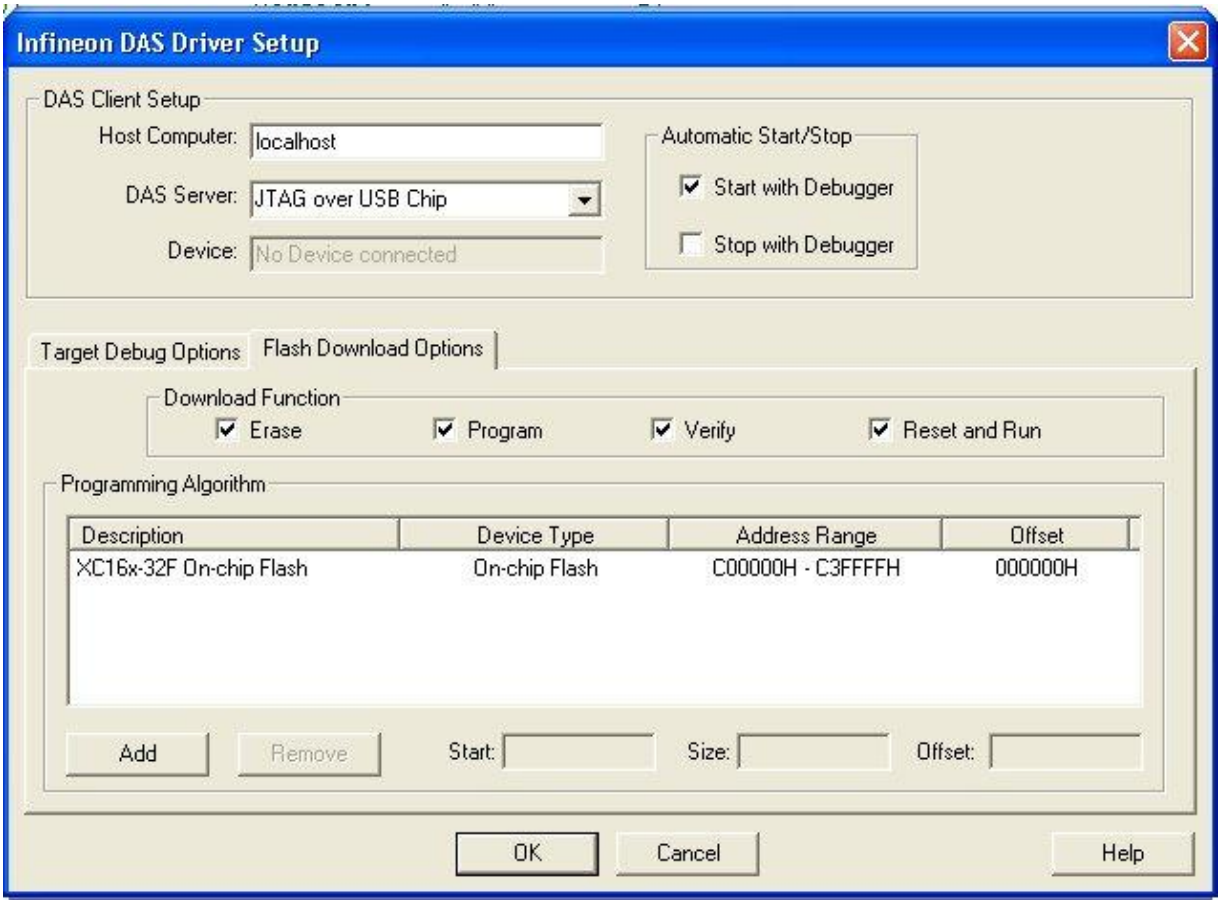
The Keil C Compiler for the XC167 is usually bundled together with the Keil uVision IDE. The user would write his code with this tool chain and then download and debug his code via a JTAG interface. Within Keil uVision the user can select a variety of interfaces, which include the standard ‘ULINK Driver for XC167’ and ‘Infineon DAS Client for XC167’.

The first one requires the use of a dongle (ULINK, ULINK2 or ULINK-ME) which interfaces between the Keil uVision IDE and the XC167 CPU module. As the XC167 CPU Module does not come with a ULINK debugger it is suggested to use the second option as detailed below. The standard ULINK tools are bought out onto a 16 way, double row, 0.1 inch pitch connector. This connector will need a reducing socket to interface to the XC167 daughter board. This connector is not provided and the user will need to make his own solution to utilise this interface with an existing ULINK tool.

The second solution, ‘Infineon DAS Client for XC167’ is designed to interface to the FTDI chip directly. As the board has been designed with the FTDI chip integrated into it, this means Keil uVision can directly connect to the XC167 module. This is done via the mini USB connector on the base board which is routed through the connectors to the XC167 CPU Module..

The settings are shown in the following pictures...


Options for Target 'xc16x'
Device
Target
Output
Listing
User
C166
EC++
A166
L166 Locate
L166 Misc
Debug
Utilities
Use Simulator
Settings
Limit Speed to Real-Time
Use:
Infineon DAS Client for XC16x
Settings
Load Application at Startup
Run to main()
Initialization File:
Edit...
Restore Debug Session Settings
Breakpoints
Toolbox
Watchpoints & PA
Memory Display
Watchpoints
CPU DLL:
Parameter:
S166.DLL
-cMODV2
Driver DLL:
Dialog DLL:
D167.DLL
-pX167CI
T167.DLL
OK
Cancel
Defaults
Help


Infineon DAS Driver Setup
DAS Client Setup
Host Computer:
localhost
DAS Server:
JTAG over USB Chip
Device:
No Device connected
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
XC16x-32F On-chip Flash
On-chip Flash
C00000H - C3FFFFH
000000H
Add
Remove
Start:
Size:
Offset:
OK
Cancel
Help

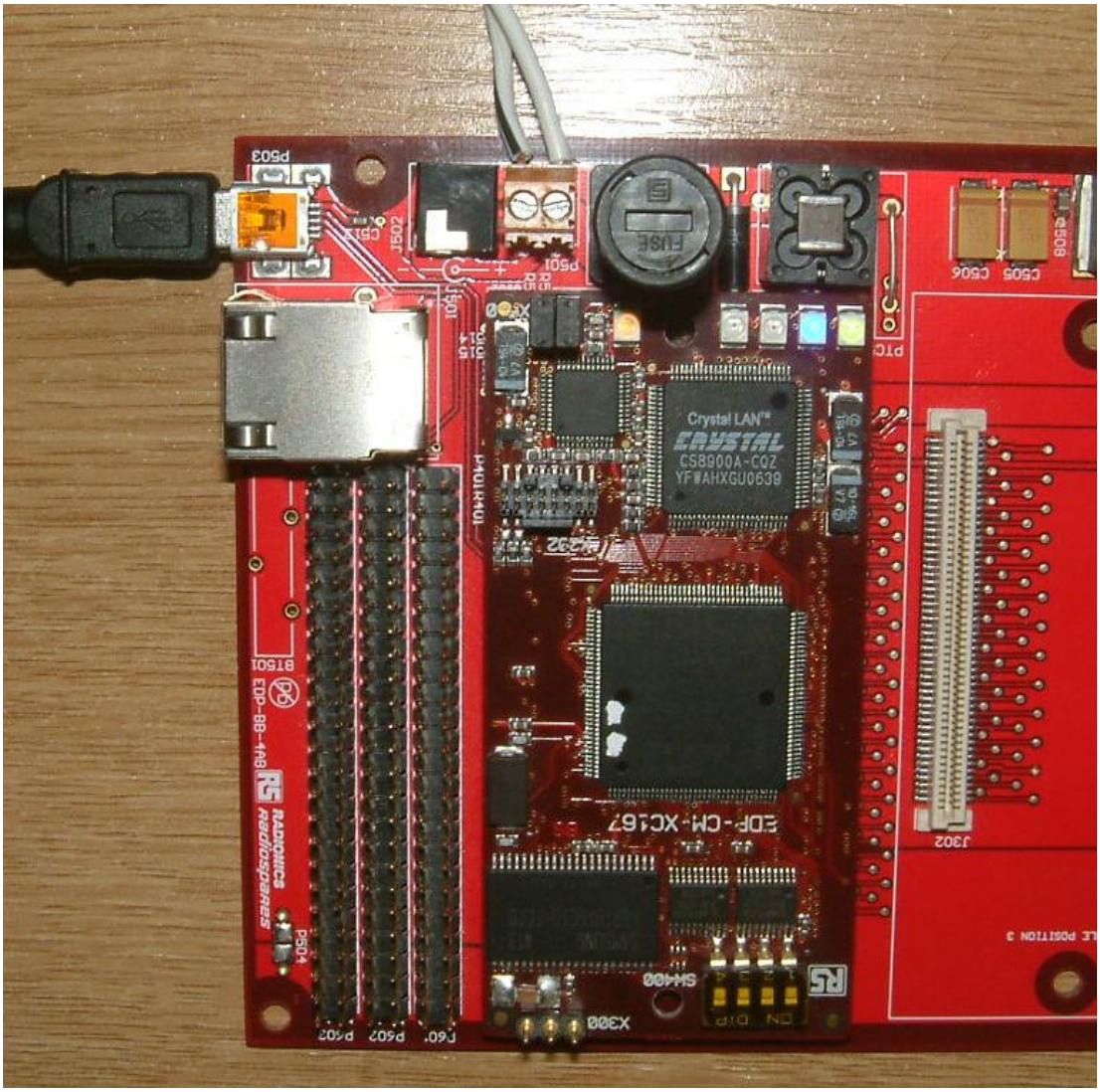
2.3 Hitex HiTOP

Hitex's own debugger and IDE combination is called HiTOP. There is a version of this specifically for the XC167 and the RS-EDP system. This is called HiTOP53 for RS-EDP-XC167.
This version of HiTOP uses the FTDI chip built into the XC167 CPU module.

HiTOP is used extensively as a debugger but not necessarily as a tool to author software. This is because the debug capabilities of the HiTOP system are very good and many customers prefer HiTOP to do the actual debugging on a target device. Also HiTOP has historically supported the larger classical bond out emulation systems that was prevalent to the advent of JTAG debugging.
With this in mind, there are lots of existing XC167 customers who have used HiTOP and have familiarity with it.

It is possible to author with Keil uVision and to generate a debug file which can then be bought in to HiTOP for debugging. This is done by creating a blank project within HiTOP and pointing to the relevant debug file.

To use HiTOP with the RS-EDP system you will need to connect a USB lead from the miniUSB connection on the base board to the host PC. The FTDI chip on the base board will handle the JTAG interface to the chip and communication to the host PC.



Base board showing mini USB connection

3. Quick Start

For the purposes of this quick start guide we will assume a 2 slot Base Board, a DC Motor Drive MC1 Module and an XC167 CPU Module. We will assume the user has downloaded and installed Keil uVision 4.00 or later and has downloaded and installed HiTOP53 for EDP-XC167.

1. Insert the Motor Drive Module into one of the vacant slots on the base board
2. Insert the CPU Module into the remaining slot on the base board
3. Connect the motor to the Motor Drive module. See user manual for MC1 Module.
4. Connect a 12V DC power supply to the Base Board. Use either the screw in type power terminals or use a wall cube PSU with centre positive and the power in jack.
5. Open Keil uVision and load in the project 'PID_V3_Noweb' for the Xc167 CPU Module.
6. Compile and build the project. The resulting output files should now be available to flash into the MCU and a debugging session can be started. See below.

3.1 Keil uVision Debugging Users

1. Connect the miniUSB lead to the base board.
2. Click on the debug icon. The Chip should be erased by the Keil Tool, followed by the flashing of the software and then the loading of the debug file into the debugger. Simply click on run to start the program execution.
3. If all has gone correctly, the program execution should have started. You can then stop the code and single step through it etc.
4. The project you have loaded in will automatically have the options selected for the 'JTAG over USB'. If they are not refer to the section above on Keil for correct settings.

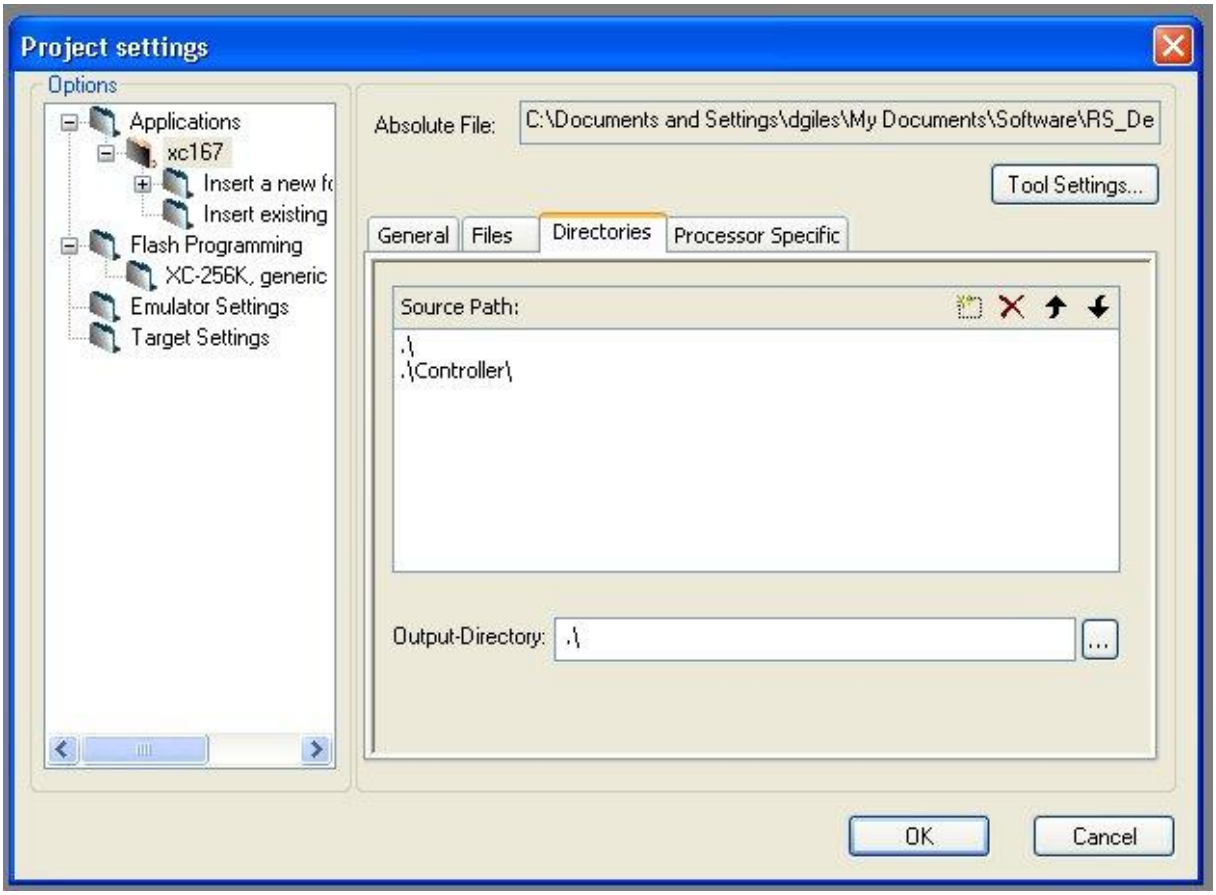
3.2 Hitex HiTOP for RS-EDP XC167

1. Connect the miniUSB lead to the base board.
2. Start HiTOP for RS-EDP-XC167.
3. Click on 'Open Existing Project'
4. Open the HiTOP project 'pid.htp'. This is a prebuilt HiTOP project which has some basic scripts included within it.
5. Click on the 'I want to continue evaluation' option.
6. When the download application dialogue box appears, do the following. Make sure the project XC167 tick box is checked; Make sure the enable FLASH programming tick box is checked. Then click on OK.
7. The HiTOP IDE will then program the XC167 device via the FTDI USB to JTAG converter.
8. Once the FLASH has been programmed, click on the 'Run' icon on the tool bar.
9. The debugger will then execute the startup code and break on the first line of the main application.
10. Click the green traffic light go icon to execute the program.
11. If all has gone correctly, you should be able to stop the code and single step through t etc.

If you are creating a new HiTOP project from scratch and importing a debug file from Keil uVision, follow the following procedure.

1. Connect the miniUSB lead to the base board.
2. Start HiTOP53 for RSEDP-XC167.
3. Click 'Create a New Project' and then next.
4. Click on the radio button – 'Create an empty project' and then next.
5. Enter a project name such as HiTOP_MotorDrivePrj
6. Click on the Browse locate a directory into which you want to locate the HiTOP project. Select the same root directory where the Keil uVision project is located.
7. Click on save and then click Next.
8. In the Tool Selection dialogue, click on RS-EDP-CM-XC167

9. In the controller selection, click on Infineon and XC167 respectively.
10. In the ‘port selection’ dialogue ensure the USB interface radio button is selected. Do not enter anything in the serial number box.
11. Leave the start up script blank and press next…
12. The project setting box should now appear automatically.
13. In the applications box we need to enter the name of the xc167 file we want to import. There is no visible extension on this file name. This contains all the debug information and symbol names etc that will be used by HiTOP for the debugging session. Click the Applications box and then the new entry box and then use the file browser which pops up to locate the xc167 file. This file is normally located in the root directory where the project is located.
14. Once this has been located, the path and file name should be automatically filled in by HiTOP
15. Uncheck the radio icon ‘Load Applications automatically at start up’ and tick the ‘Enable automatic detection of modified applications’ tick box.
16. From the Xc167 tab, select the directories tab and enter the location where the source files are kept. An example is shown below.



17. Now we need to tell HiTOP about the type of FLASH the XC167 MCU has. Within the Project Setting, highlight the flash programming option tab, and a new set of dialogue will appear. Click on ‘Add FLASH’ icon.
18. In the Add FLASH device dialogue box select, ‘Infineon’ and the ’XC 256k Generic’ option. Fill in the base address of 0x0.
19. The FLASH programming memory areas should now be visible in the ‘Installed FLASH Devices’ window.
20. In the ‘RAM Start Address’ window type in 0xe00000
21. In the ‘RAM Size’ window type in 0x800
22. Save the HiTOP project before attempting to download the code. This will ensure the project is saved as HiTOP may crash if the settings are not correct.
23. Click on the ’Debug’ tab and then select ‘Download’.
24. The ‘Download Applications’ dialogue box will then appear. Ensure the ‘Enable FLASH Download’ tick box is checked. Click on ‘OK’.
25. Click on OK. HiTOP will now automatically load the debug file and flash the MCU with the new code. Whilst you will be able to view the source files in this IDE you will not be able to modify them. To modify the source file you will need to go back to uVision. By changing the source files and then recompiling, HiTOP will now automatically detect a change in the debug image and will ask you to reload it.
26. To RESET the device click on the ‘TR’ icon button, as this is the target RESET.
27. You can then run and application or single step through the code in the normal debug fashion.
28. HiTOP can be used to write code also, using the GNU Tool Chain, or the Tasking Tool Chain. No examples of this are given here.

4. Code Size Restrictions

4.1 Keil uVision and the Keil C Compiler

The Keil tool chain allows you to compile and link code up to 8k bytes. Any more than this and uVision will complain. You can of course upgrade to a full license and several licensing models are available including educational licenses for universities and colleges. Floating licenses and Node locked licenses are available and time limited 12 month licenses also.

4.2 HiTOP for RS-EDP-XC167

The HiTOP debugger is code size restricted to 8k. HiTOP is upgradable from Hitex UK LTD to a full license. For customers that wish to use a Tasking Tool Chain with the RS-EDP it makes sense to upgrade HiTOP to the full version. This is significantly less costly than upgrading Keil uVision.

If you have a larger image from Keil uVision than you have the HiTOP licence for, then the image is still loadable into FLASH but the debug information is not available.

Most customer will use either Tasking or Keil uVision to author and compile the code, and then use HiTOP to do the actual debugging. Using a blank project and importing the debug files into this is the most popular method of using HiTOP. It is possible to completely author the whole application in HiTOP including editing and compiling but in practice most customers prefer to use their own editor.