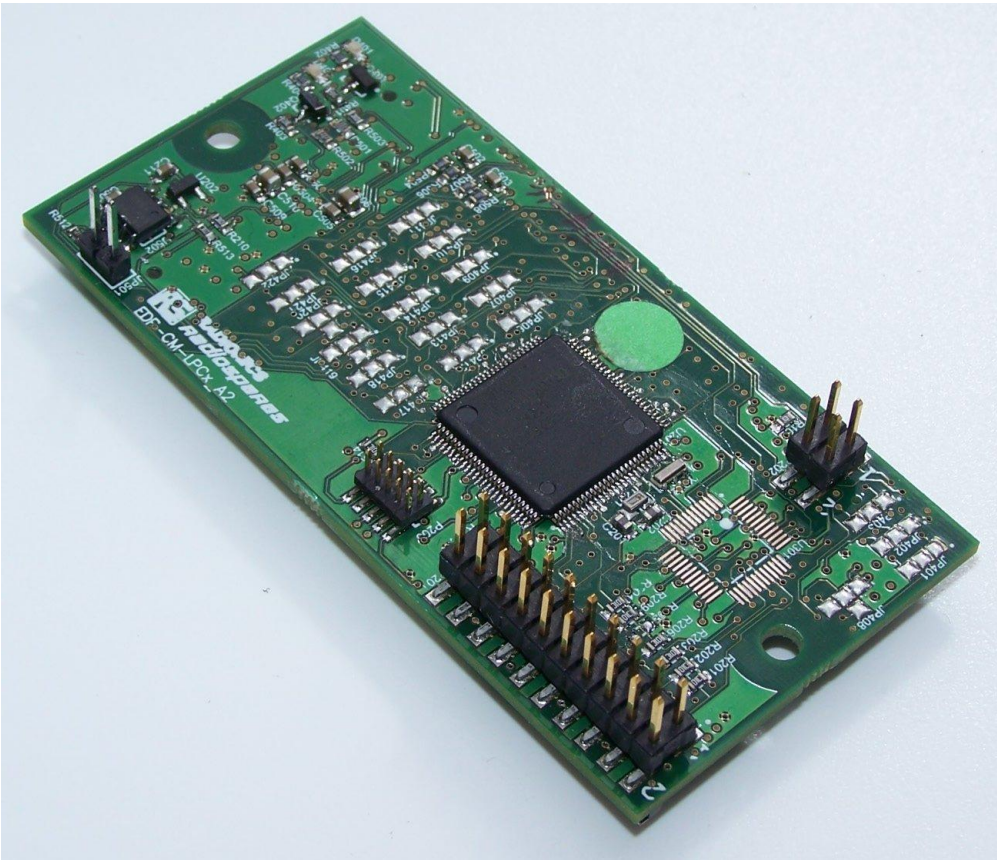




Getting Started Guide RS-EDP & LPC CPU Module

Version 2
10th June 2010



Contents

1.	Introduction	3
2.	Development Tool Support	4
2.1	Keil Debug Tools	6
2.2	Hitex Cortino	7
2.3	Hitex Cortino 2	9
2.4	JLINK.....	9
3.	Terminal Emulation	9
4.	Quick Start	9
4.1	Keil ULINK & ULINK-ME Users	10
4.2	Hitex HiTOP and Cortino Users	10
5.	Code Size Restrictions	11
5.1	Keil uVision and the ARM Real View Compiler	11
5.2	HiTOP for Cortino.....	11

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 Complete.pdf

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) 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 its own suite 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 NXP LPC module has been designed as the Command Module for the system, and this will be configured initially to talk to the Communications Module via a UART.

The 'Command Module' (sometimes 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 use a 3.3V microprocessor and so the board is configured as such. The user can check the Vcc_CM signal on the break out connector on the Base Board to determine what voltage the system is.

There are 100 pins on the MCU and these are connected via various link option 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 - 100 pin NXP Command Module RevXX.xlsx

The contents of this spreadsheet are also detailed in the user manual for the LPC1768 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 LPC1768 module is called...

Mapping Aid RS-EDP - Complete - NXP LPC1768 Rev.XX.pptx
(Name may vary slightly for other NXP modules)

This mapping aid is also present in the User Manual for the LPC 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 (back of user manual)
- The Base Board User Manual
- The LPC 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 basic RS-EDP Test Suite up and running on your CPU Module. This Test Suite is available for most of the CPU Modules and provides a way of exercising the RS-EDP system. It is a good way to test the system to ensure all of the various components are working correctly.

The Test Suite for all the CPU Modules uses the basic ASC0_RX and ASC0_TX signals on the Base Board back plane to send ASCII text strings to the Communications Module. The Communications Module has a physical layer RS232 level shifter on board which outputs its data onto a 9 way female D Connector a RS232.

A host terminal on the PC can connect to this D connector. The user can select the various tests via a menu option.

To get the system working correctly therefore it is important to ensure the ASC0_Rx and ASC0_Tx are connected correctly via the various link options. If you do not have a Communications module the user can simply bread board an RS232 level shifter for the ASC0_TX and ASC0_RX signals.

2. Development Tool Support

The CPU Module has a standard 20 way IDC Pin Header which supports both the standard JTAG and the newer SWD debug options.

JTAG is a classic debug interface that supports two break points and is used by pretty much by all of the standard ARM7 and ARM9 devices available on the market today. This interface has allowed many low cost debuggers access to the silicon to enable break point set up and single step execution of code etc. The LPC1768 supports JTAG although the CPU Core is of the newer Cortex type rather than the classic ARM7/ARM9 type.

The SWD interface is available on all new Cortex devices only. No ARM7/ARM9 based devices have a SWD interface on them. Some Cortex devices however support both JTAG and SWD interface, whilst some of the lower cost Cortex devices support only SWD. The LPC1768 supports both. This is so those customers who have already purchased ARM7/ARM9 tools can use them with the newer Cortex based part without having to purchase a new debugger.

The SWD interface uses the same MCU pins as JTAG so in many cases the user can simply plug in a debug tool into the 20 way connector IDC Connector and the debugger will decided which one to use.

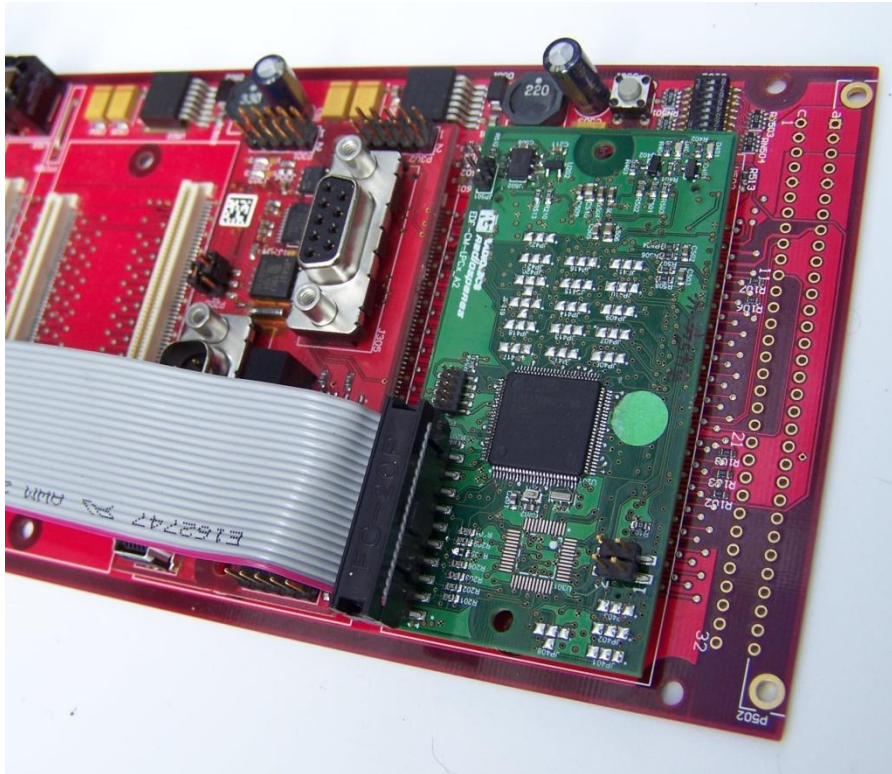
The big advantage of SWD over JTAG is that it provides eight hardware breakpoints instead of two, which is a big improvement. The interface is also much faster allowing for the examination of registers without having to stop code execution etc.

To complicate matters further ARM Ltd has developed a new physical connector for the SWD interface. This is a small 10 way mini header which allows for a smaller footprint for the connection.

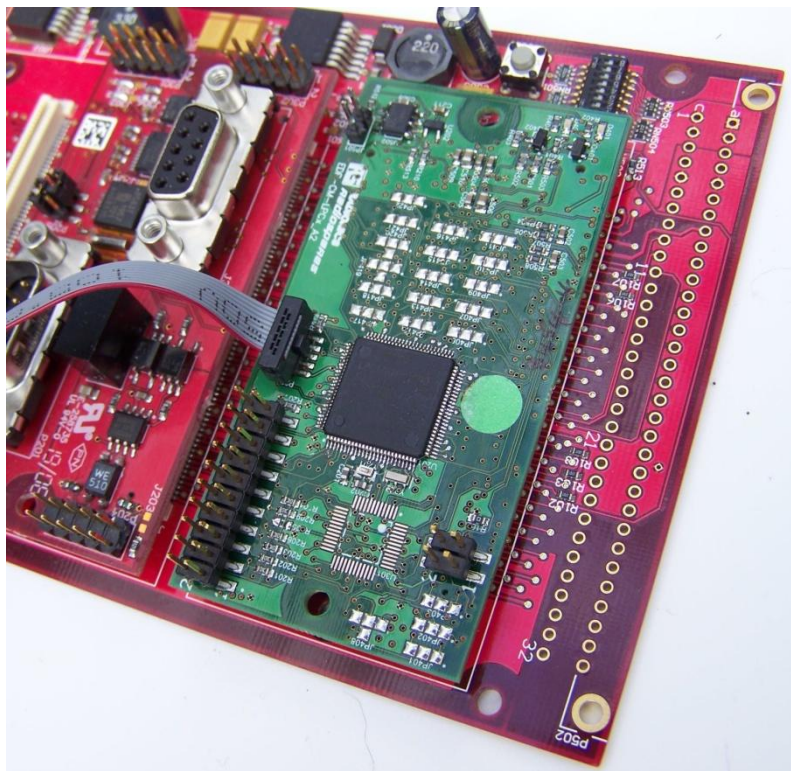
The LPC1768 CPU Module has the best of both worlds. It has both the 20 way classic header, which allows for debugging of the silicon with classic JTAG tool and SWD, plus it has a 10 way mini header for connection of SWD tools only.

Device	Core	Debug I/F	Access
LPC1113	Cortex M0	SWD Only	Use 20 pin header or 10pin mini header
LPC1343	Cortex M3	SWD Only	Use 20 pin header or 10pin mini header
LPC1768	Cortex M3	SWD & JTAG	Use 20 pin header or 10pin mini header
LPC2368	ARM7TDMI	JTAG Only	Use 20 pin header only

Device	Debug Tools Supported
LPC1113	ULINK2, ULINK-ME
LPC1343	ULINK2, ULINK-ME
LPC1768	ULINK2, ULINK-ME, Cortino
LPC2368	ULINK2, ULINK-ME, Tantino for ARM



20 Way Header with debug ribbon cable extender inserted



10 way mini connector with debug cable assembly inserted

2.1 Keil Debug Tools

The Keil debug tools are called ULINK, ULINK2 and ULINK-ME. The latter two are electrically identical. The ULINK2 comes in a plastic box and has various connection options whilst the ULINK-ME is a bare board PCB. The ULINK-ME generally comes bundled in the Keil development board and is not available to purchase separately. A variant of the ULINK-ME is available to purchase separately and it designed to work with the RS-EDP platform.

All three of these tools can be connected to the LPC1768 module via the 20 way connector. The ULINK2 has a flexible ribbon cable assembly which can be interchanged from the 20 way standard IDC connector to the newer 10 way mini one. There is no need to do this with however as SWD is also available via the 20 way IDC connector.

The ULINK2 tool is very versatile and supports pretty much all ARM7/ARM9 and Cortex M series cores. It can accommodate both JTAG and SWD debugging.

The purchase price of the ULINK2 is about £250 or so. The firmware is upgraded periodically via the Keil uVision development environment, and is the most versatile of the debuggers currently on the market.



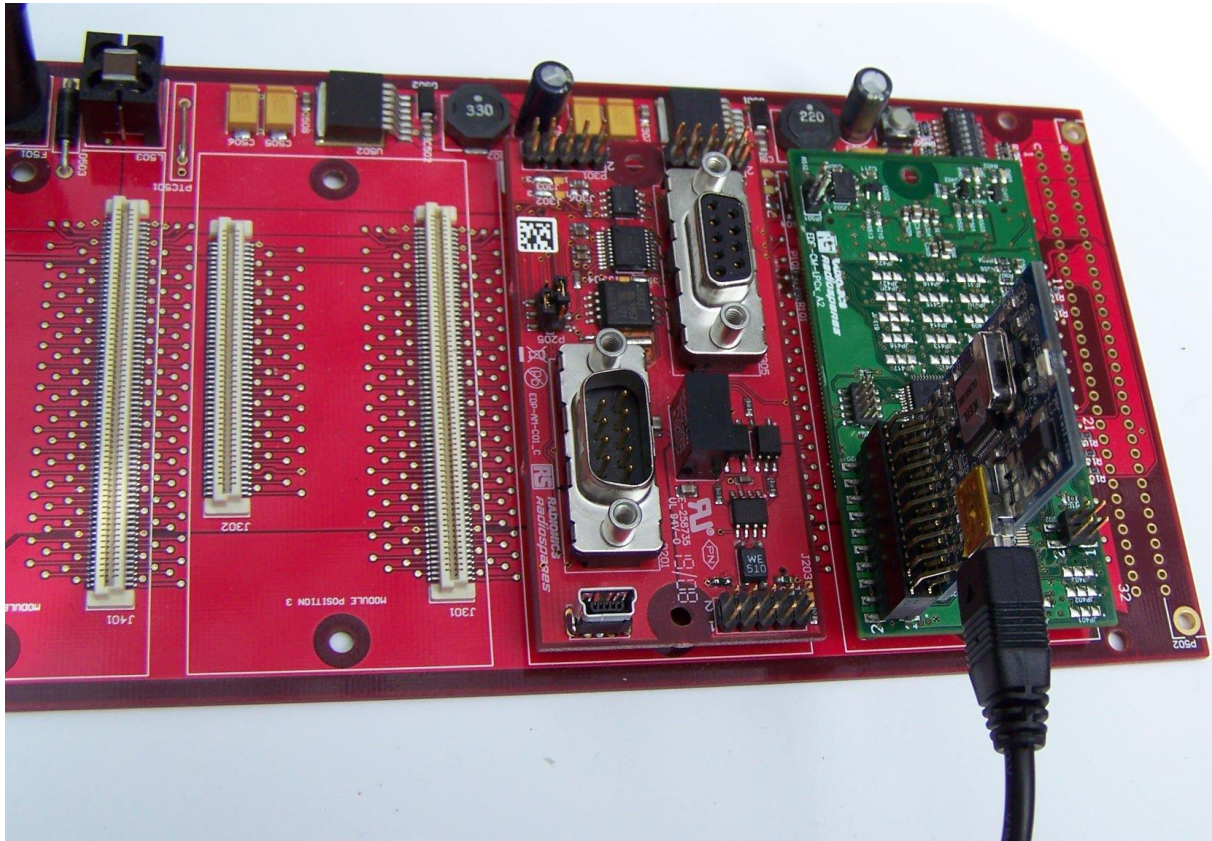
ULINK2 – Capable of both SWD and JTAG debugging. Cable assembly can be removed and replaced with smaller cable assembly for 10pin mini header. Inside the box is a row of connectors for the various cable assemblies that can be fitted.



ULINK-ME

This is electrically identical to the ULINK2 unit above. The ULINK-ME usually comes in the development kits made by Keil (Owned by ARM). The tool has all the facilities of the ULINK2 above but does not have a protected surround.

The software inside both the tools is upgraded automatically by Keil uVision when it is plugged in to the host computer. Both the tools support classic ARM7/ARM9 architectures as well as the later Cortex devices. Both tools can support JTAG and SWD.



Note correct orientation of the debugger on the CPU Module

2.2 Hitex Cortino

The Hitex Cortino tool can be purchased in a kit from Hitex. It is a tool that uses the standard JTAG programming interface found on many of the classic tools and as such will only support those Cortex based devices that have the JTAG interface on them. Fortunately the LPC1768 device supports both JTAG and SWD so the Cortino will work comfortably with the LPC1768 module via the 20 way IDC connector in JTAG mode.

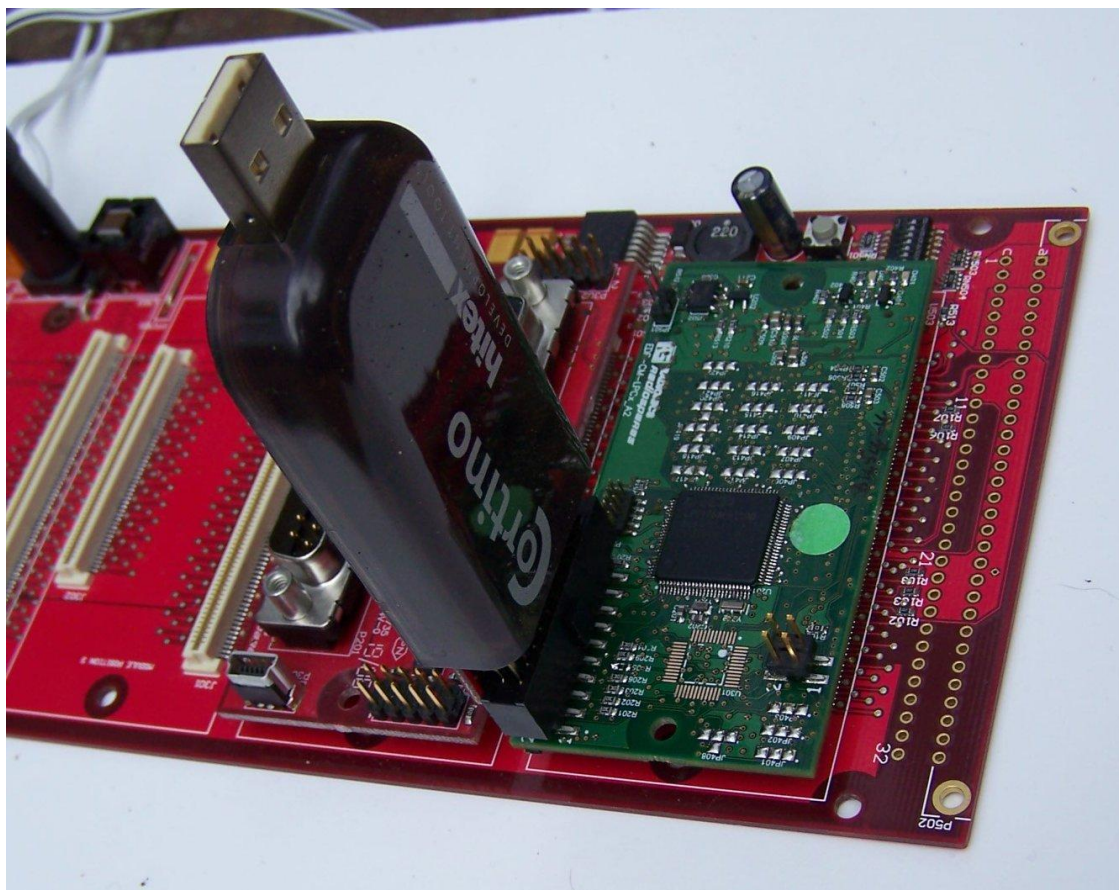
The Cortino debug dongle requires a debugger software tool called HiTOP to be installed on the host PC. This is a tremendously powerful IDE. The tool can be used as a standard editor for program creation or it can be used just as a simple debugger for programming/debugging by importing the debug files from other tool chains such as Keil uVision or IAR Embedded Workbench.

A free of charge version of the HiTOP tool can be downloaded from the Hitex web site, which allows for program generation (up to 16k/32k size restriction) using a GNU C Compiler Tool Chain or a Tasking C Compiler tool chain. HiTOP can also be used as a standalone debugger for use with the debug files generated by Keil uVision (*.axf) or IAR Embedded Workbench.

The provided software for the RS-EDP has been authored in Keil uVision so in this case the *.AXF debug file can be imported into an empty project in HiTOP and the MCU memory can be programmed and the software debugged.



The Cortino debuggers are usually supplied in a kit along with a CD and a few extra cables for the USB and programming interface. The distribution CD for this tool usually contains application projects to demonstrate the Cortino working with a GNU Compiler and the Tasking Compiler. For the RS-EDP platform all software has been authored in Keil uVision and the debug files imported into an empty project.



Note correct orientation of the debugger on the CPU Module

2.3 Hitex Cortino 2

The Cortino2 is the evolution of the Cortino. The big change is the move from classic JTAG interface to SWD. This means the Cortino 2 will be able to support eight hardware breakpoints rather than two and will be able to program and debug all of the Cortex variants including those lower cost smaller footprint devices which do not have the legacy JTAG interface on them. The LPC1768 device will therefore be compatible with the Cortino2 as the silicon supports both JTAG and SWD.

Cortino2 like Cortino will use HiTOP as a development/debug IDE. The user will need to check with the Hitex website to see if a different version of HiTOP is required to support this tool.

The Cortino2 will be re-spun into a different physical form which is the official debug tool for the RS-EDP range of NXP Modules. At the time of writing this article the Cortino2 is under development.

2.4 JLINK

IAR Embedded Workbench is another powerful tool for supporting ARM development. IAR have their own debug tool called JLINK. The JLINK tool provides support for their IDE Debugger called CSPY. For those customer who already have IAR Embedded Workbench then the LPC1768 Module is compatible with these tools as the provided interface for JTAG and SWD are industry standard. No software support examples are provided for the IAR tool chain, but it should not be too difficult to port a lot of the provided software over to support this software tool chain. The primary difference will be in the start up code and possible exception handling, which may require some special IAR directives.

3. Terminal Emulation

Most of the provided software requires the use of a terminal emulator on the host PC. Such a terminal emulator is provided free of charge in Windows. It is called Hyper Terminal and is available under Accessories – Communications in the start panel of Windows. Unfortunately my experience of Hyper Terminal has not been a positive one and I found it has some problems and generally cause Windows to crash especially when used with RS232 to USB converters. Unplugging a USB converter whilst Hyper Terminal is using it generally causes Windows to react badly. With this in mind it is better to spend some time finding one to download off the internet. A popular one which is very stable is called MTTY and has been tested and found reliable.

4. Quick Start

For the purposes of this quick start guide we will assume a 2 slot Base Board, a Communications Module and an LPC CPU Module. We will assume the user has downloaded and installed Keil uVision4.10 or later and has downloaded and installed HiTOP for Cortino2

1. Insert the Communication 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. Insert a 9 way serial cable into the female 9 way D Connector on the communication Module.
4. Connect the other end of the serial cable to your PC. If you do not have a classic serial port, use an RS232 to USB Converter.
5. Connect a 12V DC power supply to the Base Board. Use either the screw in power terminals, or a wall cube PSU with centre positive and the power in jack.
6. Connect an oscilloscope to the ASCO_TX and ASCO_RX signals lines present on the break out connector on the Base Board (optional). The connector and pin numbers can be found in either of the User Manuals for the LPC Module, the User Manual for the Base Board or in the Pin Allocation Spread Sheet document.

7. On the host PC select the required serial port, and set it up for 115,200 baud, 8 data bits, one stop bit and no parity, and then enable it.
8. Open Keil uVision and load in the project RSEDP_Test_Suite for the LPC CPU Module.
9. 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.

4.1 Keil ULINK & ULINK-ME Users

1. Connect the ULINK or ULINK-ME to either the 20 way IDC connector, or the 10 way mini connector on the CPU Module.
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 once the program execution has started some serial traffic streams should appear on the terminal emulation program. (Hyper Terminal etc).

4.2 Hitex HiTOP and Cortino Users (LPC1768 only)

1. Start HiTOP for Cortino.
2. Click 'Create a New Project' and then next
3. Click on the radio button – 'Create an empty project' and then next
4. Enter a project name such as HiTOP_TestSuitePrj
5. 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.
6. Click on save and then click Next.
7. In the Tool Selection dialogue, click on Cortino
8. In the controller selection, click on NXP and the LPC1768 device respectively.
9. In the port selection dialogue ensure the USB interface radio button is selected. Do not enter anything in the serial number box.
10. In the start up script do not enter anything. Click on next.
11. The project setting box should now appear automatically.
12. In the applications box we need to enter the name of the *.axf file we want to import. 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 AXF file. The AXF file is normally located in the 'output' sub directory where the project is located.
13. Once this has been located the path and file name should be automatically filled in by HiTOP
14. Click on the radio icon 'Load Applications automatically at start up' and tick the 'Enable automatic detection of modified applications' tick box.
15. 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.
16. Click on RESET. You may need to do this twice to correctly assert the RESET vector. Then press on the Green traffic light button to run the code.
17. If all has gone correctly you should be able to see some serial traffic on the terminal emulator.(Hyper Terminal etc).
18. Once you have created a HiTOP project you can save it and can come back in to HiTOP at a later date and simply reload the project. You do not need to create a new one from scratch each time.
19. 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.
20. It is possible to enable scripts which will help with completing some of the more common tasks. These scripts are easy to set up. See the help from with HiTOP for more information.

5. Code Size Restrictions

5.1 Keil uVision and the ARM Real View Compiler

The Keil tool chain allows you to compile and link code up to 32k 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.

5.2 HiTOP for Cortino

The HiTOP debugger is code size restricted. HiTOP is also upgradable from HITEK to a full license which is keyed to the debugger serial number. For customers that wish to use a GNU 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.