AN11321

Porting the CMSIS-DAP debugger to the Cortex-M0 platform

Rev. 1 — 30 January 2013 Application note

Document information

Info	Content
Keywords	LPC11Uxx, CMSIS-DAP, Debugger, LPC11U37
Abstract	This application note illustrates how to port the CMSIS-DAP application to a Cortex-M0 platform and then to use the Cortex-M as a debugger.

Revision history

Rev	Date	Description
1	20130130	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

This application note demonstrates the procedure used to port the CMSIS-DAP debug interface to the Cortex-M0 based LPC11U37 MCU from the Cortex-M4 platform. This MCU belongs to the LPC11UXX microcontroller family. The peripheral complement of the LPC11UXX series includes up to 128 kB of flash memory, up to 12 kB of data memory, up to 4 kB EEPROM, USB Device interface, 1 UART, 1 SSP controller, SPI interface, I^2C-bus interface, 8-channel 10-bit ADC, 4 general purpose timer/PWMs, and up to 54 general purpose I/O pins. Also present is an on-chip ROM containing In-System Programming (ISP) capability (a bootloader) supporting UART and USB flash programming, as well as APIs for user code.

The CMSIS-DAP debug interface supports both the JTAG (Joint Test Action Group) and SWD (Serial Wire Debug) interfaces.

1.1 JTAG

JTAG was established primarily to permit testing the integrity of signals between multiple chips in electronics systems. JTAG helps narrow down the anomalies present in modern printed circuit boards. This is basically a 5-pin interface. The pins are as described in Fig 1.

Pin	Description
TCK	Clock
TMS	State Machine Control
TDI	Data In
TDO	Data Out
TRST	Reset

Fig 1. Description of the JTAG hardware signals

JTAG uses 4 lines (5 when reset is utilized) to connect to boundary scan, test, and debugging components on the chip.

1.2 SWD

The Serial Wire Debug (SWD) interface utilizes only two lines for debugging: the clock and the data signal as shown in Fig 2. The Serial Wire Debug interface usually reuses the TCK (SWCLK) and TMS (SWDIO) pins of the JTAG interface. The real time trace capability is made available using a third line: Serial Wire Output (SWO). The instrumentation trace macrocell sends its data to the host over this line; this line usually overlays the JTAG Serial Out pin. The pins of the SWD interface are as shown below.

Pin	Description
SWCLK	Clock
SWDIO	Data In and Out
-	-
SWO	Serial Wire Output
-	-

Fig 2. Description of the SWD hardware signals

2. CMSIS-DAP introduction

CMSIS-DAP (Cortex Microcontroller Software Interface Standard-Debug Access Port) is the interface firmware for a Debug Unit that connects the Debug Port to USB. Debuggers, which execute on a host computer, connect via USB to the Debug Unit and to the Device that runs the application software. The Debug Unit connects via JTAG or SWD to the target device. ARM Cortex processors provide the CoreSight Debug and Trace Unit[1]. CMSIS-DAP supports target devices that contain one or more Cortex processors[2]. The system as a whole would appear as shown below in Fig 3.

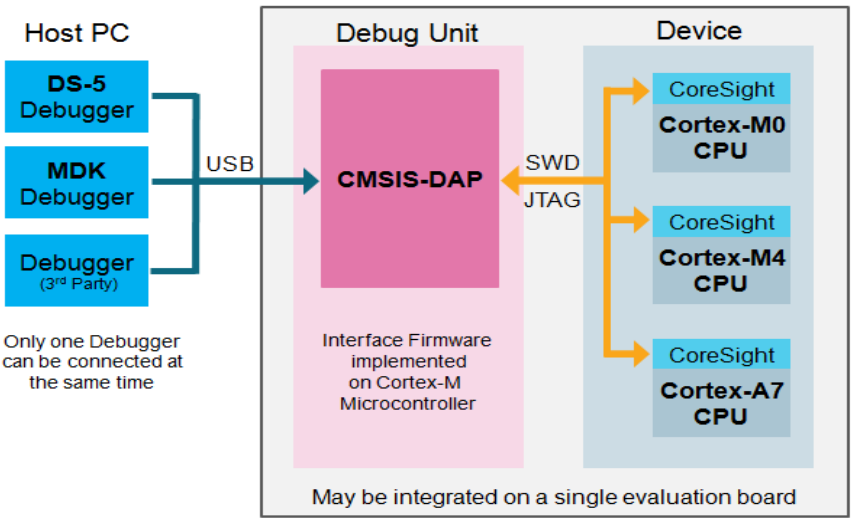

Fig 3. Overview of the system containing the CMSIS DAP, Device and Host PC

The Debug Unit can be an integral part of the evaluation board or an external debug adapter. The Debug interface by itself would look as shown in Fig 4.

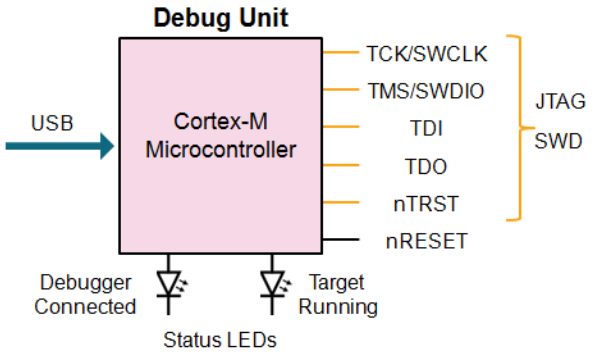

Fig 4. Overview of the system containing the CMSIS DAP, Device and Host PC

CMSIS-DAP is designed for Debug Units with Cortex-M microcontroller. Thus the CMSIS-DAP firmware can be deployed in a Debug Unit that has a Cortex-M MCU.

There is a set of hardware requirements that must be met in order to implement a CMSIS DAP debugger. They are as shown below:

- Cortex-M0, Cortex-M0+, Cortex-M3, or Cortex-M4 processor-based microcontroller.
- CPU Clock: 48 MHz or higher; the microcontroller must have a SYSTICK timer.
- RAM: 8 kB or more (The firmware can be modified in order to make it work on a processor that has lesser RAM, however, this may reduce the performance); Flash ROM: 16 kB or more.
- Full-speed or High-speed USB Device Peripheral.
- Seven standard I/O pins for JTAG/SWD Interface.
- Optional two I/O pins for Status LEDs.

Apart from these, the MDK-ARM v4.54 or later is also required. The MDK debugger contains the CMSIS-DAP Debugger driver.

3. Porting the firmware to the Cortex-M0 platform

3.1 Hardware selection

As stated earlier, the LPC11U37 MCU would be used. We use the NGX evaluation board with the LPC11U37 MCU (Fig 5). The features of this board are:

- Two layer PCB (FR-4 material)
- Power: DC 6.5 V with power LED on-board linear regulators generate +3.3 V/500 mA and +5 V/500 mA from the USB connector (as alternate power source)
- 10-pin, 20-pin Cortex debug connector for SWD (Serial Wire Debug)
- ISP, wake-up, external interrupt and reset switches
- 12.0000 MHz crystal for MCU, 32 kHz crystal for RTC
- Extension headers for all microcontroller pins
- RS232 connector, PS2 connector, Micro SD/MMC card connector, USB type-B mini connector with link-LED

- 64 x 128 graphical LCD with Backlight control
- High accuracy external RTC connected on I^2C-bus
- RTC battery holder
- 10 kOhm potentiometer for ADC

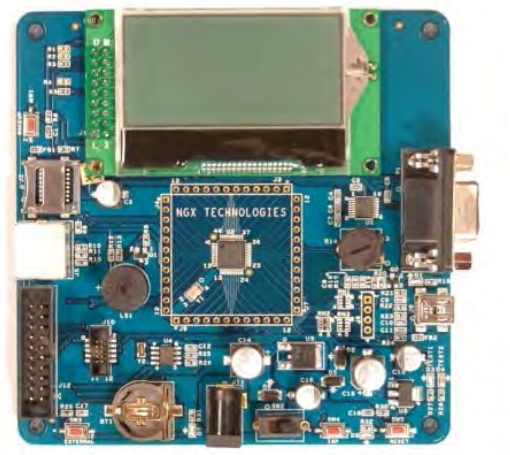

Fig 5. LPC11U37 MCU based NGX board

The NGX board needs to be interfaced/connected to the device that is being debugged. An Embedded Artists JTAG 20-pin to 10-pin converter as shown in the Fig 6 is used in order to debug the target board. The header helps to interface the GPIO pins of the debugger (that act as JTAG/SWD pins) to the target board.

Fig 6. Embedded Artists 10-pin to 20-pin JTAG adapter

Fig 7 shows the pin details of the 20 and 10-pin JTAG header interface. The designated lines from the debugger should be connected to the corresponding lines in the 20-pin header.

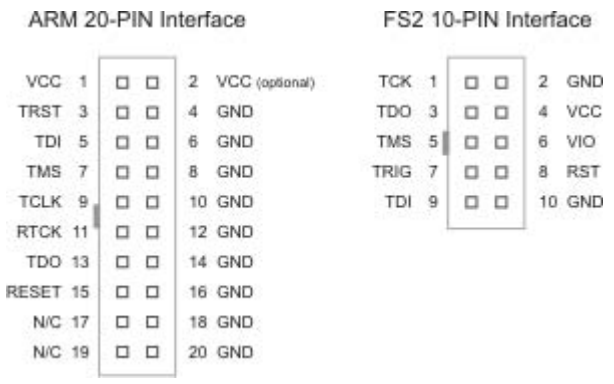

Fig 7. 20-pin and 10-pin JTAG pin details

3.2 Software implementation

As described earlier, the CMSIS-DAP firmware can be ported to any MCU that is Cortex-M based. The following steps give a brief description of the changes.

1. Debug Unit Processor: Select an appropriate microcontroller and replace the CMSIS-CORE file along with the startup file.
2. Configure the I/O ports and Debug unit: The DAP_Config file needs to be modified in order to adapt/customize the signals.
3. Configure the USB peripheral: The files in the USB driver need to be replaced with the relevant files for the new device. Modify/replace the usb_config_USB0 file settings as appropriate.
4. CMSIS-DAP Vendor commands: The CMSIS DAP firmware may be extended with the commands that are specific to a Debug Unit.

3.2.1 Debug Unit Processor

ARM (Keil) provides the CMSIS core and Startup files for every microcontroller that they support. These files are included in the MDK-ARM installation. The CMSIS core file is usually found in the C:\Keil\ARM\RV31 folder (assuming Keil was installed in the C directory). The startup_LPC11Uxx.s file is vendor provided; this is found in the vendor specific folders. In our case the start up file is located in the C:\Keil\ARM\Startup\NXP folder.

3.2.2 Configure the I/O ports and Debug unit

The I/O ports are configured in the DAP_Config file. Before doing this, the GPIO ports need to be associated to the Debug ports that will be interfaced to the target board that is to be debugged. We would need seven standard I/O pins for the JTAG/SWD device interface. Fig 8 shows the pin configuration used.

Functionality	Port	Header on NGX board
TCK/SWCLK	P1_19	J4_2
TMS/SWDIO	P1_20	J9_1
TDI	P1_22	J3_3
TDO	P1_25	J4_1
nRESET_OE	P1_27	J4_12
nReset	P1_26	J4_11
SWDIO/OE	P1_21	J7_2
LED_CONNECTED_OUT	P0_23	On-board LED, Test 1
LED_RUNNING_OUT	P0_22	On-board LED, Test 2

Fig 8. Pin configuration details

The pins are configured as GPIO pins in the DAP_Config file. Their corresponding directions (direction of data transfer) are also set here. This is done in the DAP_SETUP() function (Fig 9). The explanation to set the direction of the ports is out of scope of this Application Note and can be found in the LPC11Uxx User Manual[3]. The pin directions for pins SWCLK/TCK, SWDIO/TMS, and SWDIO_OE are set to output, TDO as input, nRESET as an input, and nRESET_OE configured as an output with its output latch set to a low level.

```
static __inline void DAP_SETUP (void) {

  /* Configure I/O pins                                          */
  LPC_IOCON->PIO1_19 = FUNC_0 | PULL_UP_ENABLED;  /* SWCLK/TCK */
  LPC_IOCON->PIO1_20 = FUNC_0 | PULL_UP_ENABLED;  /* SWDIO/TMS */
  LPC_IOCON->PIO1_21 = FUNC_0 | PULL_UP_ENABLED;  /* SWDIO_OE  */
  LPC_IOCON->PIO1_22 = FUNC_0 | PULL_UP_ENABLED;  /* TDI:      */
  LPC_IOCON->PIO1_25 = FUNC_0 | PULL_UP_ENABLED;  /* TDO:      */
  LPC_IOCON->PIO1_26 = FUNC_0 | OPENDRAIN;        /* nRESET:   */
  LPC_IOCON->PIO1_27 = FUNC_0 | PULL_UP_ENABLED;  /* nRESET_OE */
  LPC_IOCON->PIO0_23 = FUNC_0 | PULL_UP_ENABLED;  /* LED:      */
  LPC_IOCON->PIO0_22 = FUNC_0 | PULL_UP_ENABLED;  /* LED:      */

  /* Configure: SWCLK/TCK, SWDIO/TMS, SWDIO_OE, TDI as outputs (high level)  */
  /*            TDO as input                                                 */
  /*            nRESET as input with output latch set to low level           */
  /*            nRESET_OE as output (low level)                              */
  LPC_GPIO->SET[PIN_SWCLK_TCK_PORT]  =  (1 << PIN_SWCLK_TCK_BIT);
  LPC_GPIO->SET[PIN_SWDIO_TMS_PORT]  =  (1 << PIN_SWDIO_TMS_BIT);
  LPC_GPIO->SET[PIN_SWDIO_OE_PORT]   =  (1 << PIN_SWDIO_OE_BIT);
  LPC_GPIO->SET[PIN_TDI_PORT]        =  (1 << PIN_TDI_BIT);
  LPC_GPIO->CLR[PIN_nRESET_PORT]     =  (1 << PIN_nRESET_BIT);
  LPC_GPIO->CLR[PIN_nRESET_OE_PORT]  =  (1 << PIN_nRESET_OE_BIT);
  LPC_GPIO->DIR[PIN_SWCLK_TCK_PORT] |=  (1 << PIN_SWCLK_TCK_BIT);
  LPC_GPIO->DIR[PIN_SWDIO_TMS_PORT] |=  (1 << PIN_SWDIO_TMS_BIT);
  LPC_GPIO->DIR[PIN_SWDIO_OE_PORT]  |=  (1 << PIN_SWDIO_OE_BIT);
  LPC_GPIO->DIR[PIN_TDI_PORT]       |=  (1 << PIN_TDI_BIT);
  LPC_GPIO->DIR[PIN_TDO_PORT]       &= ~(1 << PIN_TDO_BIT);
  LPC_GPIO->DIR[PIN_nRESET_PORT]    &= ~(1 << PIN_nRESET_BIT);
  LPC_GPIO->DIR[PIN_nRESET_OE_PORT] |=  (1 << PIN_nRESET_OE_BIT);

  /* Configure: LED as output (turned off) */
  LPC_GPIO->CLR[LED_CONNECTED_PORT]  =  (1 << LED_CONNECTED_BIT);
  LPC_GPIO->DIR[LED_CONNECTED_PORT] |=  (1 << LED_CONNECTED_BIT);
  LPC_GPIO->CLR[LEDRUNNING_CONNECTED_PORT]  =  (1 << LEDRUNNING_CONNECTED_BIT);
  LPC_GPIO->DIR[LEDRUNNING_CONNECTED_PORT] |=  (1 << LEDRUNNING_CONNECTED_BIT);
}
```

Fig 9. DAP_SETUP() function

3.2.3 Configure the USB peripheral

The USB_Config file contains the details of the USB peripheral type. It specifies/defines the properties of a USB peripheral like the type of device (HID, CDC, etc.), the endpoint types, the size of the reports, Vendor ID, Product ID, power setting, Product Strings, etc. This file is to be modified or replaced accordingly. The Cortex-M4 MCU originally used had two onboard USB controllers whereas the LPC11U37 has a single USB controller. It is up to the user's discretion to either modify the USB_Config in the original project or replace it with a USB_Config that was written for the specific platform. We chose to replace it with the file present in the directory C:\Keil\ARM\RL\USB.

The file should be added to the project and the configuration can be done using the Configuration Wizard. The changes that are to be done are depicted in Fig 10.

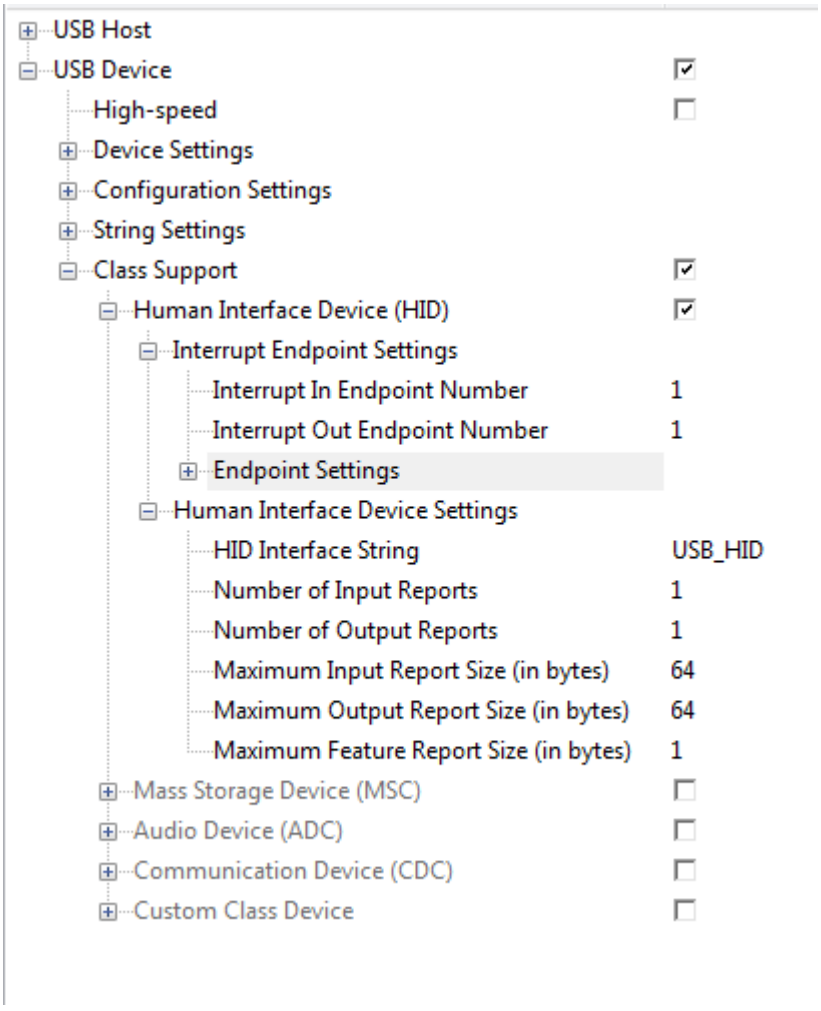

Fig 10. USB_Config changes

It can be seen that the CMSIS-DAP is an HID (Human Interface Device) class interface. The maximum input report size and maximum output report size are 64 bytes in length. It has two endpoints, both being of type Interrupt. One of the endpoints is an IN and the other is an OUT endpoint.

3.2.4 CMSIS-DAP Vendor commands

There are no CMSIS-DAP vendor commands implemented. A vendor can choose to implement custom commands and integrate them into the system.

Once these changes are implemented, the code will be built and flashed onto the NGX board.

4. Deploying CMSIS DAP debugger

The project is built and the NGX development board is flashed once the changes in the software are implemented as described in the previous section. The GPIO pins that are

configured as the debug ports are to be connected to the 10-pin to 20-pin header. The connections from the NGX board to the converter are shown in the Fig 11.

Pin/Functionality	Pin on the converter
TCK/SWCLK(J4_2)	9
TMS/SWDIO(J9_1)	7
TDI(J3_3)	5
TDO(J4_1)	13
nReset(J4_11)	15
GND(J3_5)	10

Fig 11. NGX board to Pin converter connection

The development board can now be used as a debugger. The new debugger connected to the header is shown in Fig 12 below.

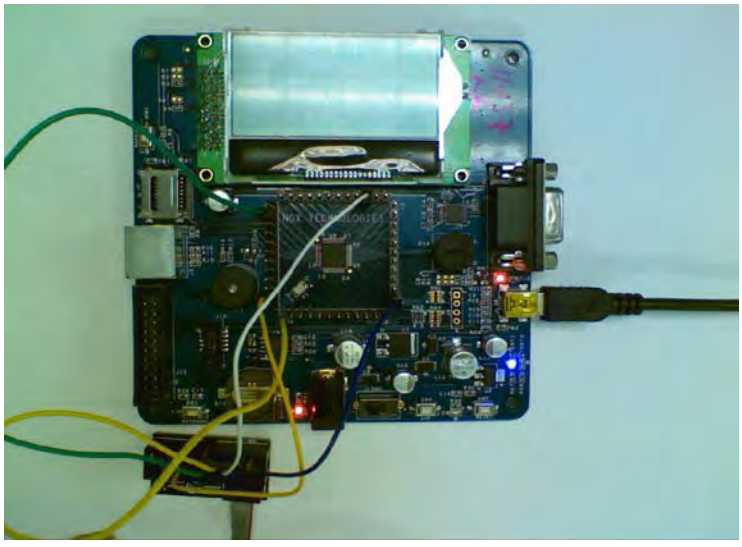

Fig 12. Debugger connected to the Pin converter header

To use it as a debugger, there is a small change required in the MDK-ARM IDE. The “CMSIS-DAP Debugger” option needs to be selected in the drop-down of the “Utilities” tab in the project options window (see Fig 13). Note that this must also be done in the project that will use the CMSIS-DAP as the debugger.

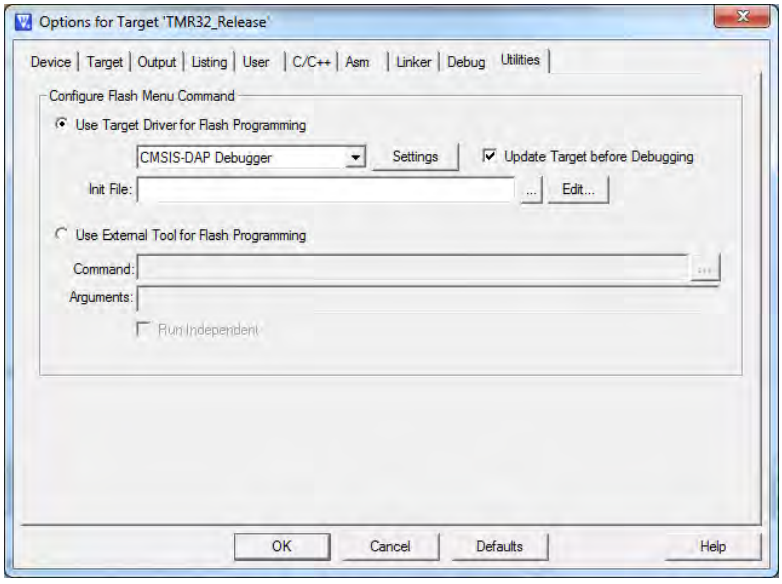

Fig 13. Project settings

Pressing the “Settings” button (see Fig 13) and choosing the “Debug” tab will provide the details of the connected Debugger. The name, firmware version, and serial number of the debugger will be displayed (see Fig 14).

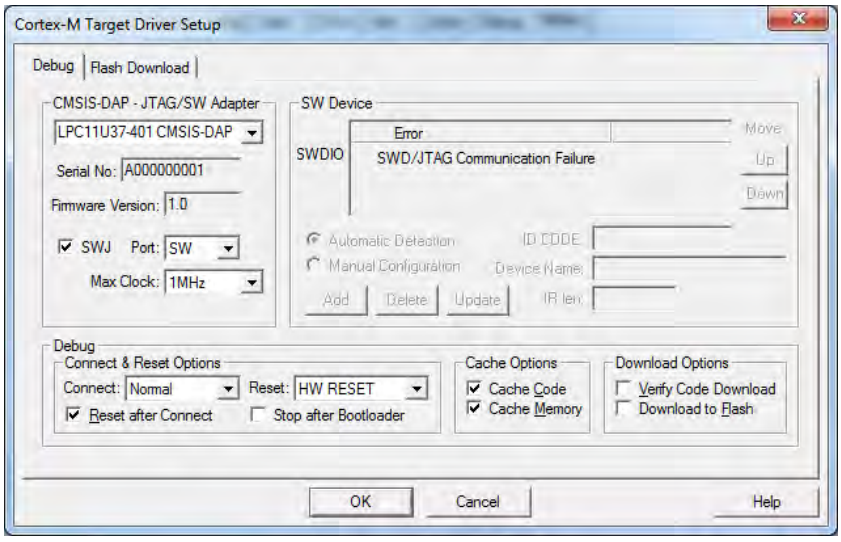

Fig 14. Debugger details

5. Testing the debugger

This section takes a look at the procedure to program the flash memory of an LPC11U14 device using the debugger. We program the flash memory of the LPC11U14 based board (MCB11U10) with a simple Blinky (Timer32) application.

The 10-pin header is connected to the JTAG interface on the MCB11U10 board and the board is powered up. The target core is visible in the “Cortex-M Target Driver Setup” window as shown below (Fig 15).

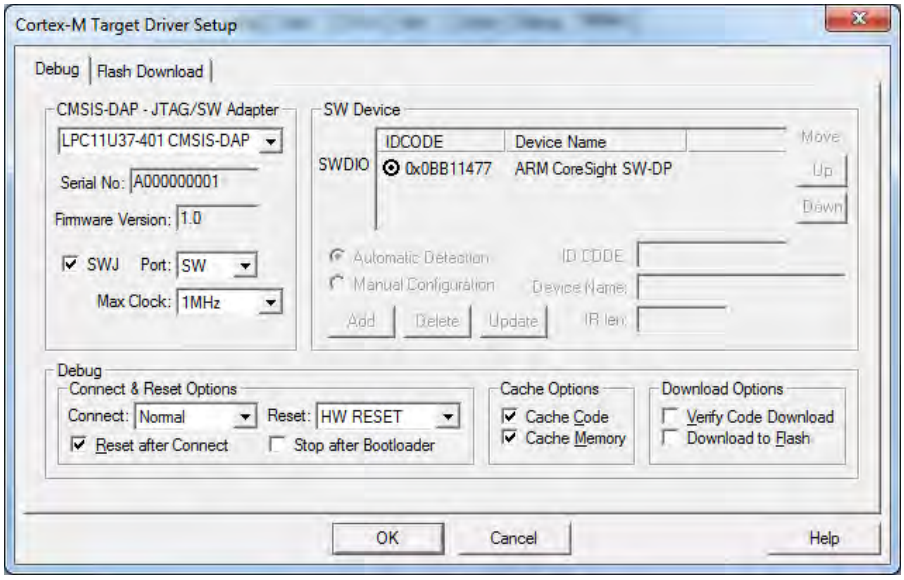

Fig 15. Target core details

Once the core is verified, the application is built by selecting “Build all target files” in the “Project” menu and then flashed by selecting “Download” from the “Flash” menu.

6. References

[1] CoreSight On-chip Debug & Trace IP, http://www.arm.com/products/system-ip/debug-trace/index.php

[2] CMSIS-DAP Debugger User's Guide, http://www.keil.com/support/man/docs/dapdebug/dapdebug_introduction.htm

[3] LPC11UXX user manual (UM10462), http://www.nxp.com/documents/user_manual/UM10462.pdf

7. Legal information

7.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

7.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

8. List of figures

9. Contents

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 30 January 2013
Document identifier: AN11321