Freescale Semiconductor, Inc. Application Note

Document Number: AN5229 Rev. 0, 01/2016

ARM DS-5 Development Studio Debug i.MX6UL-EVK

1. Introduction

This application note is intended to introduce the ARM DS-5 IDE debug functions based on an MX6UL EVK board.

Contents

1.	Introduction	1
2.	ARM [®] DS-5 Development Studio Introduction.	2
3.	i.MX 6UltraLite-Low-power, Secure, ARM® Co	ortex [®] -A7
	Core	2
4.	Requirements	4
5.	ARM DS-5 IDE	5
	5.1. How to install	6
	5.2. How to create a project	8
6.	RealView-ICE Debug	10
	6.1. Debug Bare Metal Code	10
	6.2. Debug U-Boot Code	
	6.3. Debug Linux kernel code	
7.	J-LINK Debug	
	7.1. Install software	
	7.2. Using J-Link to debug i.MX6UL	
8.	Revision history	



2. ARM® DS-5 Development Studio Introduction

The ARM DS-5 is a professional software development solution for Linux-based and bare-metal embedded systems, covering all the stages in development, from boot code and kernel porting to application debug.

DS-5 includes the following component tools:

- 1. Eclipse-based IDE combines software development with the compilation technology of the DS-5 tools.
- 2. DS-5 Compilation Tools.
- 3. DS-5 Debugger, together with a supported debug target, enables debugging of application programs and complete control over the flow of program execution to quickly isolate and correct errors.

DS-5 has three different editions: Community, Professional, and Ultimate. This Application Note is based on the ARM DS-5 Ultimate Edition.

3. i.MX 6UltraLite-Low-power, Secure, ARM[®] Cortex[®]-A7 Core

Expanding the i.MX 6 series, the i.MX 6UltraLite is a high performance, ultra-efficient processor family featuring an advanced implementation of a single ARM Cortex-A7 core, which operates at speeds up to 528 MHz. The i.MX 6UltraLite applications processor includes an integrated power management module that reduces the complexity of external power supply and simplifies power sequencing. Each processor in this family provides various memory interfaces, including 16-bit LPDDR2, DDR3, DDR3L, raw and managed NAND flash, NOR flash, eMMC, QSPI SPI, and a wide range of other interfaces for connecting peripherals such as WLAN, BluetoothTM, GPS, displays, and camera.

- Features
 - ARM Cortex-A7 @ 528 MHz, 128 KB L2 cache
 - Parallel LCD Display up to WXGA (1366x768)
 - 8/10/16/24-bit Parallel Camera Sensor Interface
 - 16-bit LP-DDR2, DDR3/DDR3L
 - 8/16-bit Parallel NOR FLASH / PSRAM
 - Dual-channel Quad-SPI NOR FLASH
 - 8-bit Raw NAND FLASH with 40-bit ECC
 - 2xMMC 4.5/SD 3.0/SDIO Port
 - 2xUSB 2.0 OTG, HS/FS, Device or Host with PHY
 - Audio Interfaces include 3x I2S/SAI, S/PDIF Tx/Rx
 - 2x10/100 Ethernet with IEEE 1588
 - 2x12-bit ADC, up to 10 input channel total, with resistive touch controller (4-wire/5-

wire)

- Partial PMU Integration
- Security Block: TRNG, Crypto Engine (AES/TDES/SHA/RSA with DPA), Tamper Monitor, Secure Boot, SIMV2/EVMSIM X 2, OTF DRAM Encryption, PCI4.0 precertification
- 14x14 289MAPBGA 0.8mm pitch
- 9x9 289 MAPBGA 0.5mm pitch



Figure 1. i.MX 6UL MPU Block Diagram

4. Requirements

- ARM DS-5 Software for Linux or Windows
- ARM RealView-ICE Debugger or J-LINK Debugger
- i.MX 6UtraLite EVK Rev.C (HW rework needed)
- JTAT Connection Between Debug and i.MX 6UtraLite EVK
- Ethernet/USB Connection Between Host PC and Debugger



Figure 2. i.MX 6UL-EVK board

NOTE:

The JTAG Port pin is reused by Audio codec WM8960 SAI2 pin on the i.MX 6 UL-EVK board. In order to correctly connect to the JTAG port of the board with the RealView-ICE Debugger or the JLINK debugger, five resistances R1432, R1407, R1431, R1433, and R1434 need to be removed. This is shown in the following figure.



Figure 3. i.MX 6UL-EVK Hardware rework for JTAG pin

The RealView-ICE Debugger is connected to the i.MX 6UL-EVK board via the JTAG port:



Figure 4. i.MX 6UL-EVK connection with RealView-ICE Debugger

5. ARM DS-5 IDE

The ARM Development Studio (DS-5) IDE is based on the Eclipse Platform.

The workbench is the main development environment where you can manage individual projects, associated sub-folders, and source files. It uses a single folder called a workspace to store files and folders related to specific projects. A typical workbench window contains one or more perspectives, a set of related views, editors, menus, and toolbars.

ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

DS-5 uses the C/C++ and DS-5 Debug perspectives.

Further details can be found at: <u>http://ds.arm.com/</u>

5.1. How to install

The ARM DS-5 can be found at: <u>http://ds.arm.com/downloads/</u>. Install it by following the step by step wizard.

Open DS-5 IDE, click the Help menu, select the ARM License Manager, and generate a 30-day free license.

C 45944			C Add License	- 0 ×
ARM License Manager	2.5		Choose network interface	
View and edit licenses Configure licenses and diagnose licensing problems.		\rightarrow	Choose a network interface that the license will be locked to	
Configuration Diagnostics			The new license will be locked to one of your computer's network interfaces. It is recommen physical network interface. The license will stop working if a virtual network interface is select adverse that schemeness.	ded to select a ted and its MAC
	Add License		Network interface: D48ED928F7F7 - Intel(R) 82579LM Gigabit Network Connection	-
C Add License	191			
Select License Type Select the ture of Scene that you would like to use				
O Use a license file, license server, serial number, or activation code				
Generate a 30-day evaluation license for DS-5 Ultimate Edition				
			(?) <back next=""> Finish</back>	Cancel
			_	
(?) < Back Next >	Finish	Cancel		
		<i>C</i>		
		C Add License		
		Developer a	count details	
		Enter the ARM	A developer (Silver) account details	
	-			
		Enter account	details:	
		Email:		
		Password:		
View and edit licenses		Don't have an	account? Click here to reset your password.	
Configure acenses and diagnose acensing problems.				
Configuration Diagnostics				
DS-5 Ultimate Edition (30-day Evaluation).lic	Add License			
	Delete License			
Select the toolkit that you intend to use:				
ARM DS-5 Ultimate Edition (Evaluation)	:			
		?	< Back Next > Finish	Cancel
	-			



When connecting the RealView-ICE Debugger with the Host PC via Ethernet/USB for the first time, you may need to update the debugger firmware. To do this, open the Debug Hardware Update in the Windows Start Menu and update the firmware with the following wizard:



Figure 6. Update the RealView-ICE Debugger firmware

ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

5.2. How to create a project

Eclipse IDE can show a range of programs. There are two classes of project; the first class of project requires ARM DS-5 to build and the binary image. Create the project as follows:

			-		
N	Vew	Alt+Shift+N ►	C.	Makefile Project with Existing Code	
C	Open File		C-	C++ Project	
c	lose	Ctrl+W	C	C Project	
c	lose All	Ctrl+Shift+W	Project	Project	
s Is	ave	Ctrl+S	Convert to a C/C++ Project (Adds C/C++ Nature)		
s	ave As	Ctrl+Shift+S	Source Folder Source Folder		
s	ave All				
R	levert	7766707665			
				Header File	
N	Aove			File from Template	
R	lename	F2	G Class	Class	
R	Refresh	FS			
c	Convert Line Delimiters To	,		Other	Ctrl+N

Figure 7. Create a built project

You must select the correct Toolchains to build this project:

Project Create C project of selected type	
Project name: uart-mx6ul	
Use default location	
.ocation: E:\local_program\imx6ul\uart-mx6ul	Browse
Choose file system: default 👻	
Project type:	Toolchains:
 Executable Empty Project Hello World ANSI C Project Shared Library Static Library Makefile project 	Cygwin GCC GCC 4.x [arm-linux-gnueabihf] (DS-5 built-in) MinGW GCC
Show project types and toolchains only if the	ey are supported on the platform

Figure 8. Select ARM Compiler 5 Toolchains for building

The second class of project is when ARM DS-5 is not used to build the project. The project can be built by Linux Host. You must import existing source codes:

ARM DS-5 IDE

File	Edit Source Refactor Navigate Sear	ch Proje	t Run Window Help	
	New Alt+Shift	N .	Makefile Project with Existing Code	
	Open File Close Ctrl-	تم ۲۰	C++ Project C Project Project	
	Close All Ctrl+Shift- Save Ctrl Save As Save All Ctrl+Shift Revert	+S 63 +S 63 •S 63	Convert to a C/C++ Project (Adds C/C++ Nature) Source Folder Folder Source File	
2	Move Rename Refresh Convert Line Delimiters To	F2 ©	Header File File from Template Class Other	Ctrl+N

Project Name		
linux_imx_3.14.38		
Existing Code Location		
H:\work\code\linux_imx_3.14.3	3	Browse
Languages V C V C++ Toolchain for Indexer Settings	Browse For Folder Select root directory of existing code	
<none> ARM Compiler 5 (DS-5 built-in) Cygwin GCC GCC4x [arm-linux-gnueabihf] MinGW GCC</none>	Video WDJDownload Work Code	
Show only available toolchain		

Figure 9. Create a no built project

6. RealView-ICE Debug

Realview-ICE is a powerful ARM core debugger, it supports JTAG port connection.



Figure 10. RealView-ICE debugger

6.1. Debug Bare Metal Code

Open ARM DS-5 and create a built project as shown in section 5.2. Import the source code to the project:

O Import		C Import	
Select Import resources from the local file system into an existing project.	Ľ	File system Import resources from the local file system.	
Select an import source: type filter text		From directory: E\Debug Tools\ARM DS-S\uart-mx6ul Image: Comparison of the system of the s	Browse
 > Remote Systems > Run/Debug > Scatter File Editor > Target Configuration Editor > Team 		Filter Types Select All Into folder: uart-imx6ul Options Overwrite existing resources without warning Create top-level folder Advanced >>	Browse
(?) < Back Next > Finish	Cancel		Cancel

Figure 11. Import the source code to the project

Configure the Project Properties then select the ARM C Compiler Target:



Figure 12. Select the ARM C Compiler 5 Target

ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

Select the ARM Assembler 5 Target Configuration:



Figure 13. Select the ARM Assembler 5 Target

Select the ARM Linker Target Configuration:



Figure 14. Select the ARM C Linker Target

Select Image Layout (Scatter file) Configuration,



Execution region

Figure 15. Select the scatter file

Build the project; right click at the project title to select Build Project:

C/C++ - uart-	тхби	l/main.c - Eclipse Platform			
File Edit Source	ce F	Refactor Navigate Search Projec	t Run Wind	ow Help	
📑 🕶 📰 🗞 🗠	8 6) 🔹 🔦 👻 🔜 🔍 🔜 🔂 🔹 🚳	• 🖻 • 🞯	• 🚸 • Ο	
Project E 83	<u>م</u> (۱	Streamli	3		-
		□ ☆ ▽ ○> (-
иаrt-mxбu	\supset	New	- UARTI TXD	_TX_DATA_	
Include	•	Go Into	,	RX DATA	-
Debug		Open in New Window			
Main.c	_	Open in New Window		·(const cha	
b le uart.c		Сору		*puart = (
▷ 🗈 uart.h		Delete		1 · · · ·	
i scatter		Move		t->uts & U	🏦 Problems 🖉 Tasks 🕼 Console 🖏 🔲 Properties 🔳 Commands
		Rename		[0] = -src	CDT Build Console Juart-myoull
	2	Import			
	èse -	Export		bug_uart(s	4150 286 38 32 0 0 KOM IOTAIS
	<	Build Project			
		Clean Project		14	
	8	Refresh	F5	(&uart1, 1	
		Close Project			Total RO Size (Code + RO Data) 4188 (4.09KB)
		Close Unrelated Projects		sts("Hello	10tal RW Size (RW Data + 21 Data) 4476 (4.378)
		Build Configurations	•	lo from i.	10cal ROM Size (Code + RO Data + RW Data) 4220 ($4.12kb$)
		Make Targets			
		Index	,		
		Show in Remote Systems view		Console 33	Interface with the property week and a set
		Debug As		s time.	Finished building target: uart-mxbul.axt
		Run As	,		
		Team	•		
		Compare With	,		15:49:55 Build Finished (took 105.155ms)
		Restore from Local History			
1	*	Run C/C++ Code Analysis			
1		Properties	Alt+Enter		
		-			

Figure 16. Build the project

Configure the DS-5 debugger connection:

	C Debug Configurations		
	Create, manage, and run configuration () [Debugger]: Debugging from a symbol), but no symbol files defined in the Files tab	\$
Window Help	0	Name: i.MX6 Ultralite-RealView ICE	
 Control of the second second	Image: Section of the section of th	Connection File File File File File File File File	II

Figure 17. Configure DS-5 debugger connection

NOTE:

if there is no "i.MX6 UltralLite(Generic)" item in Connection->Select Target, copy folder "i.MX6 UltraLite(Generic)" in "mx6ul_evk_ds-5\DS-5-Board-i.MX6 UL\Boards\Freescale" to path "\$(ARM DS-5 Install path)\DS-5 v5.22.0\sw\debugger\configdb\Boards\Freescale".

Configure the DS-5 Debugger Files:

eate, manage, and run configurations reate, edit or choose a configuration to	launch a DS-5 debugging session.		20
	Name: i.MX6 UltraLite-RealView ICE	 Open Select a file: Suart-mx6ul cproject project Project Debug main.d main.d main.d objects.mk objects.mk subdir.mk uart-mx6ul.axf uart.o include main.c 	Cancel
lter matched 21 of 22 items		(Apply Reyert

Figure 18. Configure the DS-5 Debugger Files Connection

To initialize the target's chip register, select Debug Configurations-> Debugger-> Run target initialization debugger script.

] 🗎 🗶 🛱 🍄 🔹	Name:
type filter text C C/C++ Application	🗢 Connection 🔀 Files 👫 Debugger 🛛 🎯 OS Awareness 🕬 Arguments 🚟 Environment
C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application DS-5 Debugger	Run control Connect only Debug from entry point Debug from symbol main Run target initialization debugger script (dc / m)
New_configuration	E\Debug Tools\ARM DS-S\scripts\JMX6UL_DDR3L_1GB_16bit_default.ds File System Workspace
e ^w IronPython unittest ^w Java Applet	Run debug initialization debugger script (.ds / .py)
Java Application 2 ³ Jython run	Execute debugger commands
 Jython unittest Launch Group 	
PyDev Django PyDev Google App Run PyDev Google App Run	
Python unittest	•
C Zylin Embedded debug (Cygwin)	Host working directory

Figure 19. Configure target initialization debugger script

When the debug connects successfully, the DS-5 Debug UI is displayed:



Figure 20. DS-5 Debugger Connected View

Debug and Run:



Figure 21. DS-5 Debug view

ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

6.2. Debug U-Boot Code

Before debug, the U-Boot must be compiled on a Linux Host.

Open the ARM DS-5, create a no-built project as in section 5.2, and import your U-Boot source code.

			New Project
			Import Existing Code Create a new Makefile project from existing code in that same directory
		-	Project Name u-boot-imx
		-	Existing Code Location H:\work\code\u-boot-imx Browse
0	C/C++ - Eclipse Platform	Project Run Window Hela	Languages Image: Image Control Contro Contro Control Control Control Control Control Control
-	New Alt+Shift+N >	Makefile Project with Existing Code	Toolchain for Indexer Settings
	Open File Close Ctrl+W Close All Ctrl+Shift+W	C++ Project C Project Project	<none> ARM Compiler 5 (DS-5 built-in) Cygwin GCC</none>
	Save Ctrl+S Save As Save All Ctrl+Shift+S Revert	Convert to a C/C++ Project (Adds C/C++ Nature) Source Folder Folder Source File	MinGW GCC
£	Move Rename F2 Refresh F5	Header File File from Template Class Other Ctrl-N	Show only available toolchains that support this platform Finish Cancel

Figure 22. Create U-boot project

After the project has been created, use the following command to setup a boot SD card:

\$ sudo dd if=u-boot.imx of=/dev/sdc bs=512 seek=2

Connect your i.MX6UL-EVK board Debug UART to your PC and open the PC's serial terminal. Turn on the power switch to boot the MX6UL EVK board; the serial terminal will then display the U-Boot log if setup has been successful. Quickly type any key in the terminal view to stop the U-Boot at its command line:

B COM19 - PuTTY
A
CPU: Freescale i.MX6UL rev1.0 at 396 MHz
CPU: Thermal invalid data, fuse: 0x0
CPU: Temperature: Can't find sensor device
Reset cause: POR
Board: MX6UL EVK
I2C: ready
DRAM: 512 MiB
MMC: FSL_SDHC: 0, FSL_SDHC: 1
Card did not respond to voltage select!
MMC init failed
Using default environment
Di
DISPIAY: IFI43AB (480x272)
In: serial
Out: Serial
Err: Serial
Nat. EC1
Net: FLU
With any key to stop autoboots.
->

Figure 23. U-boot command prompt line

Configure the DS-5 Debug Connection:



Figure 24. DS-5 Debugger Connections Configuration

Configure the DS-5 Debugger Configuration and click on start Debug at the bottom right:

Debug Configurations	
Create, manage, and run configurations	launch a DS-5 debugging session
Create, edit or choose a configuration to a	aunch a US-5 debugging session.
	Name: i.MX6 UltraLite-Uboot
type filter text	🖇 Connection 🔚 Files 👫 Debugger) 🚳 OS Awareness 🚧 Arguments 🌌 Environment
 C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application ♣ DS-5 Debugger ♣ i.MX6 DualLite-RealView ICE ♣ i.MX6 DualLite-RealView ICE 	Run control Connect only Debug from entry point Debug from symbol main Run target initialization debugger script (.ds / .py) Run debug initialization debugger script (ds / .py)
 i.MX6 UltraLite-RealView ICE i.MX6 UltraLite-Uboot MX6Q-U-boot Debug 	File System Workspace
 IronPython Run IronPython unittest Java Applet Java Application Jut Junit Jython run 	
e [¥] Jython unittest ▶ Launch Group ⓓ PyDev Django	Host working directory Actions to perform after the connection is established Image: Use default [workspace_loc:) File System
 ▲ PyDev Google App Run ✓ Python Run ✓ Python unittest ☑ Remote Java Application 	Paths Source search directory S(workspace_loc:/uboot-imx) File System Workspace
Filter matched 24 of 25 items	Apply Revert
?	Debug Close

Figure 25. DS-5 Debugger Configurations

When connected successfully, open the DS-5 Debug UI, click on Run and then interrupt it when debug starts:



Figure 26. DS-5 Debugger Connected View



After the U-Boot has been interrupted, then Load, Add Symbols File and click on U-Boot:

Figure 27. Load Symbol File

Run and then Interrupt the U-Boot, you can then see where Cortex-A7 has stopped and the functions in the stack:

DS-5 Debug - uboot-imx/fs/fat/fat.c - Eclipse Platform	Sun Window Hale	0 0
	8 • 6 c • c •	Quick Access 📓 🗟 🐼 🍋
Debug C N Project E Remote C Debug Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y	Commands 23 History Scripts Commands 23 History Scripts Linked: Test MADN(ING(ING66): Unable to check the image is the wait ERROR(CMD361): Wait for stopped interrupted interrupt Execution stopped at: S:0x9FF75FFC S:0x9FF75FFC TST r3,0x20 add-symbol-file "E:\workspace\uboot-imx\u-boot" wait continue interrupt Execution stopped at: S:0x9FF75FFC S:0x9FF75FFC 1150,0 char "nextname = NULL; *	Me Varia 9 Brea 22 重 Regi 行 Expr 10 Func 平日 業 強 亞 文 · · · · · · · · · · · · · · · · · ·
<pre> @ facc 03 1144 firsttime = 1; 1145 1146 while (isdir) { 1147 int startsect = mydata->data_begin 1148 + START(denptr) * mydata->clust 1149 dir_satay_dant; 1149 char *nextname = NULL; 1151 dent = *dentptr; 1152 dent = *dentptr; 1153 idx = dirdelim(subname); 1156 1157 if (idx >= 0) { 1158 subname[idx] = '\0'; 1159 nextname = subname idx + 1; 1160 /* Handle multiple delimiters */ 1161 while (ISDIRDELDW(*nextname)) 1162 if (dols && *nextname == '\8') </pre>	size;	sa_ 13

Figure 28. U-boot Debug View

6.3. Debug Linux kernel code

Before debugging, the Linux kernel code should be compiled on a Linux Host.

Enable and compile the kernel debug info options; this will include the symbol information for the debugger in the 'vmlinux' kernel executable:



Figure 29. Configure kernel make menu

NOTE:

If your i.MX6UL-EVK board runs Linux kernel version "imx_3.14.38_6ul_ga", you must apply this patch for JTAG: 0001-For-DS-5-debug-and-uart-console-work-normally.patch

Then build the kernel on Linux Host.

Setup a boot SD card following the MX6UL Linux User Guide.

Connect your i.MX6UL-EVK board Debug UART to a PC, open the serial terminal of the PC, turn on the power switch to boot the MX6UL EVK board, the serial terminal will display the kernel log if setup successfully.



Figure 30. Kernel command prompt line

ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

Configure the DS-5 Debug connection:

	C Debug Configurations		
	Create, manage, and run configurations		×.
		Nime: I.MX6 UltraLite-Linux	
	type filter text	- Connection Files & Debugger @ OS Awareness 🕬 Arguments 🗮 Environment	
	C C/C++ Application C C/C++ Attach to Application C C/C++ Postmortem Debugger C C/C++ Remote Application	Select target Select the manufacturer, board, project type and debug operation to use. Currently selected: Freescale / IMMS UltraUte(Generic) / Linux Kernel and/or Device Driver Debug / Debug Cortex-A7	
	DS-5 Debugger MXKS Dualitie-RealView ICE MXKS Quad-RealView ICE MXKS Ultralitie-RealView ICE MXKS Ultralitie-RealView ICE MXKS Ultralitie-Veloot MXKQ-U-boot Debug	Filter platforms	•
lindow Help	 Test IronPython Run IronPython unittest 	Linux Agglication Debug Linux Kernel and/or Device Driver Debug	
G • (***) • (****) • (*****) • (*******) # 1 i.MX6 Ultralite-RealView ICE * 2 i.MX6 Quad-RealView ICE * 3 i.MX6 DualLite-RealView ICE	Iava Applet Java Applet Java Applet Java Applet Journal Journal Johon ruin Johon unitest Launch Group Johon Group Johon Group Johon Group Johon Group Johon Group	Sinetis L Series (Generic) Twrk-LS2021A Vybrid Vr3ax Vybrid Vr3ax Vybrid Vr5w Target Connection RealWiew ICE	The currently selected
JART1_ (D */ JART1_ Debug Configurations Organize Favorites	Python Run Vertical Python unittest Remote Java Application	DS-5 Debugger will connect to a RealView ICE to debug a Linux kernel. Connections Linux Kernel Debug Connection USB:137940077	Browse.
t_str(const char *src)	Elter matched 25 of 26 items	C	Apply Revert

Figure 31. Configure DS-5 Debug Connection

NOTE:

If there is no "i.MX6 UltralLite(Generic)" item in Connection->Select Target, then copy the folder "i.MX6 UltraLite(Generic)" in "mx6ul_evk_ds-5\DS-5-Board-i.MX6 UL\Boards\Freescale" to path "\$(ARM DS-5 Install path)\DS-5 v5.22.0\sw\debugger\configdb\Boards\Freescale". Configure the DS-5 Debug Debugger Configuration, and start the debug:

 Debug Configurations Create, manage, and run configurations Create, edit or choose a configuration to 	aunch a DS-5 debugging session.	12 12 12 12 12 12 12 12 12 12 12 12 12 1
Create, edit or choose a configuration to ype filter text C/C++ Application C/C++ Astach to Application C/C++ Postmortem Debugger C/C++ Remote Application C/C++ Remote Apple C/C++ Remote Apple Discrete C/C++ Remote Apple C/C++ Remote Apple C/C++ Remote Application C/C++ Remote Application C/C++ Remote Application C/C++ Remote Application C/C++ Remote Application	aunch a DS-5 debugging session.	Workspace
Filter matched 25 of 26 items		bug Close

Figure 32. Configure DS-5 Debug Connection

The DS-5 Debug UI will then open, and Try RUN, or Interrupt menu.



Figure 33. Configure DS-5 Debugger Connected View

Interrupt the Linux and then Load Symbols File:



Figure 34. Load Linux kernel symbol file

J-LINK Debug

Run and then interrupt the Linux, you can then see where Cortex-A7 has stopped and the functions in the stack:



Figure 35. Linux kernel debug view

7. J-LINK Debug

ARM DS-5 IDE supports the J-LINK debugger when Debug Configurations select Zylin Embedded debug(Native) plugin. It based on GDB command between GDB-server and GDB-clinet.

7.1. Install software

Open DS-5, select Help -> Install New Software. To install the zylincdt plugin use the following parameters:

Name: zylincdt Location: <u>http://opensource.zylin.com/zylincdt</u>

J-LINK Debug

C Add Re	pository	×
Name:	zylincdt	Local
Location:	http://opensource.zylin.com/zylincdt	Archive
?	ОК	Cancel

Figure 36. Add zylincdt plugin

Install the J-Link software and documentation pack at https://www.segger.com/jlinksoftware.html



Figure 37. J-Link software

After the installation is complete you will find the J-Link software in the start menu.

Next, install the GDB tools (yagarto-bu-2.23.1_gcc-4.7.2-c-c++_nl-1.20.0_gdb-7.5.1_eabi_20121222.exe) from <u>http://sourceforge.net/projects/yagarto</u>

F http://sourceforge.net/projects/yagarto/		오 🗸 🖒 🥔 Zylin AS - open source 🖾 SEGGER - The Embe
sourceforge	Search	Browse Enterprise Blog Jobs Deals Help
SOLUTION CENTERS Go Parallel	Resources News	vsletters
ManageEngine ServiceDesk Plus Home / Browse / Development / Bu YAGARTO Brought to you by: mifi	UIII TOOIS / YAGARTO	IT Help Desk Software
Summary Files Revie	ws Support Wiki	ki News Code
★ 4.4 Stars (20) ↓ 182 Downloads (This Ⅲ Last Update: 2013-0	Week) 4-24	Sf Download Jagarto-bu-2 23.1_goc-4.7.2-o-c++_nexe Browse All Files

Figure 38. GDB tools Download

7.2. Using J-Link to debug i.MX6UL

Open ARM DS-5, use File -> Import to import the demo code, or you can create a project as in setion 5.2.

Import	
Select Create new projects from an archive file or directory.	Ľ
Select an import source:	
type filter text	
 General Archive File Existing Projects into Workspace File System Preferences C/C++ C/C++ Project Settings Existing Code as Makefile Project CVS Remote Systems Run/Debug Scatter File Editor Target Configuration Editor Team 	E
Sack Next > Finis	sh Cancel

Figure 39. Import the demo project



Configure the Project Properties, select ARM C Compiler 5 Target,



ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

J-LINK Debug

Configure the ARM Assembler 5 target:

Properties for uart-mx6ul				
type filter text	Settings			$ \diamondsuit \bullet \bullet \diamond \bullet \bullet$
 Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings 	Configuration: Debug [Active] Tool Settings Build Steps (Armonic Settings Build Steps (Armonic Settings Build Steps (Armonic Settings Build Steps (Armonic Settings Settings) Armonic Settings (Armonic Settings) Armonic Settings (Armonic Settings) Armonic Settings (Armonic Settings) Armonic Settings (Armonic Settings) Armonic Settings) Armonic Settings (Armonic Settings) Armonic Settings) Armonic Settings (Armonic Settings) Armonic Setti	 Build Artifact Binary Parse Target CPU (cpu) Byte order Instruction set Interworking (apcs=/interww Disable unaligned accesses (- Target FPU (fpu) Floating-point mode (fpmode) Floating-point PCS (apcs) 	rs Error Parsers Cortex-A7.no_neon.no_vfp Default Default ork -no_unaligned_access) Default Default	Manage Configurations
?				OK Cancel

Figure 41. Configure ARM Assembler 5 target

Configure the ARM Linker 5 target:



Figure 42. Configure ARM Linker 5 target

J-LINK Debug

Select Image Layout (scatter file) configuration:

Properties for uart-mx6ul	
type filter text	Settings 🗘 🕆 🖒 🔻 🔻
 Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Project References Run/Debug Settings 	Configuration: Debug [Active]
?	OK Cancel

Figure 43. Configure ARM Linker Image Layout

Click the Build button to build the demo code.

C/C++ - uart-mx6ul/main.c - Eclipse Platform	🔋 Problems 🖉 Tasks 📮 Console 🔉 🔲 Properties 🗖 Commands
File Edit Source Refactor Navigate Search Project Run	CDT Build Console Juart-mxbuil
i d • 2 • 4 + 9 (€) a a d • 6 • 6 • (4150 286 38 32 0 0 ROM Total
Project Explorer 💱 📓 (Build 'Debug' for project 'uart-mußul'	
	Total R0 Size (Code + R0 Data)4188 (4.09kB)Total RW Size (RW Data + ZI Data)4476 (4.37kB)Total R0M Size (Code + R0 Data + RW Data)4220 (4.12kB)
⊳ ku uart.h ≧ jlink.gdbinit	'Finished building target: uart-mx6ul.axf'
📕 scatter.scat	15:49:33 Build Finished (took 10s.153ms)

Figure 44. Build the project

DS-5 Debug Connection Configuration and select GDB debugger path installed before:

Debug Configurations		×
Create, manage, and run configurations		
		Mar.
	Name: Zylin JLink GDB Debug-Native	
type hiter text	📄 Main 📏 Debugger 🗷 Commands 🦻 Source 🖾 Environment 🗔 Common	
I.MX6 Quad-RealView ICE	Project (optional):	
4 i.MX6 UltraLite-OBDS	uart-mx6ul B	irowse
🐴 i.MX6 UltraLite-RealView ICE	C/C++ Appliestion.	
A MX6 UltraLite-Uboot	C:\sync\DS-5 Workspace\uart-mx6u\Debug\uart-mx6ul.axf Search Project B	rowse
A Test		
🏘 ZLG i.MX6UL JLink	A hbhiraron cousoic	
e ^T IronPython Run		
P IronPython unittest		
Java Application		
Ju JUnit		
e ²⁷ Jython run		
Jython unittest E Launch Group		
d PyDev Django		
a PyDev Google App Run		
er Python Run		
Python unittest		
FG Zylin Embedded debug (Cygwin)		
 IC Zylin Embedded debag (Netive) 		
I I I I I I I I I I I I I I I I I I I		
Filter matched 32 of 33 items	Apply	Revert
?	Debug	Close
Debug Configurations		×
Debug Configurations Create, manage, and run configurations		×
Debug Configurations Create, manage, and run configurations		∞
Debug Configurations Create, manage, and run configurations		×
Debug Configurations Create, manage, and run configurations Debug Configurations	Name: Zylin JLink GDB Debug-Native	×
Debug Configurations Create, manage, and run configurations Debug Configurations	Name: Zylin JLink GDB Debug·Native	×
Debug Configurations Create, manage, and run configurations Image: the set of the se	Name: Zylin JLink GDB Debug-Native	×
Debug Configurations Create, manage, and run configurations Image: provide the state of the stat	Name: Zylin JLink GDB Debug-Native Main (参 Debugger Commands) Source Environment Common Debugger: Embedded GDB	×
Debug Configurations Create, manage, and run configurations Image: the second sec	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Debugger: Embedded GDB Stop on startup at: Advanced	×
Debug Configurations Create, manage, and run configurations Image: Image of the second se	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Advanced	×
Debug Configurations Create, manage, and run configurations Image: Image in the image in th	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main	×
Debug Configurations Create, manage, and run configurations Image: the set of the s	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main	×
Debug Configurations Create, manage, and run configurations Image: the set of the s	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugger: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe	×
Debug Configurations Create, manage, and run configurations Image: the set of the s	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe GDB command file: Bro	×
Debug Configurations Create, manage, and run configurations Image: the second	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)	××××××××××××××××××××××××××××××××××××××
Debug Configurations Create, manage, and run configurations Image: and run configurations Image: and run configurations Image: and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set.	xwse
Debug Configurations Create, manage, and run configurations Image: the second	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard	xwse
Debug Configurations Create, manage, and run configurations Image: and run configurations Image: and run configurations Image: and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Environment Common Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro (Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set. Standard Protocol: mi	XWSE
Debug Configurations Create, manage, and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set. Standard Protocol: mi Verbose console mode Main	XWSE
Debug Configurations Create, manage, and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Use full file path to set breakpoints	xwse
Debug Configurations Create, manage, and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Use full file path to set breakpoints Use full file path to set breakpoints	XWSE
Debug Configurations Create, manage, and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Use full file path to set breakpoints	XWSE
Debug Configurations Create, manage, and run configurations ()	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Source Stop on startup at: Advanced Debugger Options Main GDB debugter: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: ni Use full file path to set breakpoints	XWSE
Debug Configurations Create, manage, and run configurations Image: Ima	Name: Zylin JLink GDB Debug-Native Main Commands Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugger: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Use full file path to set breakpoints	XWSE
Debug Configurations Create, manage, and run configurations ()	Name: Zylin JLink GDB Debug-Native Main Debugger Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugger: C:\yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro (Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard • Protocol: mi • Use full file path to set breakpoints	XWSE
Debug Configurations Create, manage, and run configurations ()	Name: Zylin JLink GDB Debug-Native Main Debugger Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Bro Main GDB debugter: C:tyagarto-20121222\bin\arm-none-eabi-gdb.exe GDB debugter: C:tyagarto-20121222\bin\arm-none-eabi-gdb.exe Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Verbose console mode Use full file path to set breakpoints	xwse
Debug Configurations Create, manage, and run configurations ()	Name: Zylin JLink GDB Debug-Native Main Debugger Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugger: C:\vagarto-20121222\bin\arm-none-eabi-gdb.exe GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set Standard Protocol: mi Use full file path to set breakpoints	xwse wwse
Debug Configurations Create, manage, and run configurations with the second s	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Debugger: Embedded GDB Stop on startup at: Advanced Debugger Options Main GDB debugter: C:yagarto-20121222\bin\arm-none-eabi-gdb.exe Bro GDB command file: Bro Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: Standard Protocol: mi Verbose console mode Use full file path to set breakpoints	wyse



ARM DS-5 Development Studio Debug i.MX6UL-EVK, Application Note, Rev. 0, 01/2016

Input six "Initialize" commands:

- target remote localhost:2331
- monitor reset
- monitor sleep 300
- monitor reg cpsr = 0xd3
- load
- break main

Debug Configurations		×
Create, manage, and run configurations		\$
Image: Second	Name: Zylin JLink GDB Debug-Native Main Debugger Commands Help/tips on how to setup GDB init script Initialize' commands target remote localhost2331 monitor reset monitor reg cpsr = 0xd3 load "Run' commands	× v
(?)	Debug	lose

Figure 46. GDB command

J-LINK Debug

Open the Segger J-Link GDB Server:

SE	SEGGER J-Link GDB Server V4.98b - Config				
	Connection to J-Link				
<	💿 USB 🔲 Serial No.				
	C TCP/IP				
	Target device				
\triangleleft	Little endian 💌				
	Target interface				
	Speed	Misc. settings			
	 Auto selection 	✓ Init registers			
	C Adaptive clocking				
	○ 1000 💌 kHz				
Command line option					
	-select USB -device Cortex-A7 -if JTAG -speed auto -ir				
		OK Cancel Exit			

Figure 47. J-Link GDB Server

The following parameters indicate that J-Link has connected with the target board successfully:

SEGGER J-Link GDB Server V4.98b					
File Help					
GDB Waiting for connection Initial JTAG	speed Auto	nost only n top log window ate logfile download gs on start			
Log output Clear Hog Silent mode: off Single run mode: off Target connection timeout: 5000 J-Link related settings J-Link host interface: USB J-Link script: none J-Link settings file: none Target device: Corte Target interface: JTA Target entice: OKHz Target endian: litt	ns ex-A7				
Connecting to J-Link J-Link is connected. Firmware: J-Link V9 compiled Aug 28 2015 17:49:48 Hardware: V9.10 S/N: 59101075 Feature(s): GDB, RDI, FlashBP, FlashDL, JFlash Checking target voltage Target voltage: 3.25 V Listening on TCP/IP port 2331 Connecting to targetWARNING: At least one of the connected devices is WARNING: At least one of the connected devices is not JTAG compliant (IE)					
J-Link found 3 JTAG devices, Total IRLen = 13 JTAG ID: 0x5BA00477 (Cortex-A7) Connected to target Waiting for GDB connection					
0 Bytes downloaded	3 JTAG devices, IRLen=13	1.			

Figure 48. J-Link Connected With Target

Select the Debug button to debug the project:

C/C++ - Eclipse Platform	
File Edit Source Refactor Navigate Search Project Run Window Help	
■・日本2 多・多・3 ×▶日=およりよモス 3 @・6	3 • 2 • 6 • * • 0 • 9 • 19 % • 1 1 2 • 5 • •
Project Explore X R Streamline D., C	Tc 1 Zylin JLink GDB Debug-Native
	2 i.MX6 UltraLite-OBDS
	n 3 Test
imxbEncrypt	4 ZLG i.MX6UL JLink
Content of the second s	5 i.MX6 UltraLite-Uboot
a wart-mxbui	6 MX6O-U-boot Debug
Debug	7 i.MX6 UltraLite-Linux
b Ch include	& A i MX5 Litralite-RealView ICF
h R main c	
N P wart c	Debug As +
b is watch	Debug Configurations
ilink adbinit	Organize Favorites
scatter.scat	
uboot-imx	



Stop the Debugger to watch the call stack and some debug information:



Figure 50. J-Link GDB Debug View

8. Revision history

Revision number	Date	Substantive changes
0	01/2016	Initial release

 Table 1.
 Revision history

How to Reach Us:

Home Page: freescale.com

Web Support: freescale.com/support Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off.

ARM, the ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All other product or service names are the property of their respective owners. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document Number: AN5229 Rev. 0 01/2016



