# ARM JTAG DONGLE $D/\gamma$

# **User Guide**



### INTRODUCTION

ARM JTAG DONGLE *DIY* is a wiggler-compatible JTAG board supporting RDI interface for ARM microcontrollers. It provides a low cost alternative for students or amateur engineers to debug and program ARM microcontrollers. This product is offered in unassembled form (Figure 1). However, only few components are required to finish the board so any averaged electronic engineer will be able to finish it within 5 minutes.



Figure 1 ARM JTAG DONGLE DIY

# SOFTWARE

ARM JTAG DONGLE *DIY* is compatible with H-JTAG, a free JTAG debug agent for ARM. H-JTAG supports most of the popular debuggers, like SDT2.51, ADS1.2, REALVIEW and IAR. With the help of H-JTAG, you can debug all the ARM7/ARM9 based processors.

Web site of H-JTAG http://www.hjtag.com/index.html

H-JTAG package includes the H-JTAG server for connecting your PC to an ARM micro, and H-FLASHER will also be installed for programming different NOR flash and on-chip flash.

Manuals and operation instructions can be found under the H-JTAG web pages.

### SETUP

Follow the schematic to finish the JTAG board. At one end, connect the DB-25 male port to the parallel port of your PC by a straight extension cable (not included). At the other end, connect the 20-pins flat cable (supplied with our package) from JTAG board to your evaluation board. Pin-out of the 20-pins flat cable is found under the schematic. Figure 2 shows the setup for our LPC2103 evaluation board as an example. Some ARM micros will check on the status of DBGSEL pin before entering debug mode. One needs to short this pin to high for debug. On LPC2103-Eval-1A it is the P5 jumper to short prior to a successful JTAG connection. Otherwise, the JTAG board will fail to halt the micro therefore failing to enter to debug state.



Figure 2 Connect ARM JTAG board to an ARM evaluation board via 20-pins flat cable

Apply power to the ARM evaluation board. There is no power supply for JTAG board because it draws power from the evaluation board. Launch H-JTAG software (Figure 3) to connect to the ARM microcontroller.

MPLAB IDE v8.02	🛅 Tencent Software	•	
	💼 EAGLE Layout Editor 4.16r2	•	🔁 H-Converter
Microsoft Office Wor	🕎 Keil uVision3		🔁 H-Flasher
	Philips Semiconductors	►	🚼 H-JTAG
All Programs 👂	m ATMEL	•	付 Uninstall
	i PenPowerJR	►	🔁 User Manual (CN)
	🖮 Н-ЈТАБ	×	🔁 User Manual (EN)
🏭 start	🛅 Internet Banking of Bank of Communications	•	
🚑 start	Internet Banking of Bank of Communications	•	🔎 User Manual (EN)

Figure 3 Launch H-JTAG software

H-JTAG Server automatically displays the IDCODE it detects. Different devices have different ID codes. For example, the code for LPC2103 is 0x4F1F0F0F.



If you are using Atmel's AT91SAM7S256, the IDCODE will be different.



Normally H-JTAG Server will automatically detect the interface and establish a connection. Just in case there is an error, go to **Jtag Settings** under **Settings** main menu to bring up the setup menu. Make sure Wiggler option has been selected with nTRST set to Pin2 D0. Try different values for TCK Speed for stability test. Finally, minimize H-JTAG Server. Don't close it.

Jtag Settings	
Jtag Selection	Wiggler Pin Assignment
• Wiggler (Predefined)	TMS Pin3 D1 🖃
C Sdt Jtag (Predefined) C User Defined	TCK Pin4 D2 🔽
Reset Signal Output	TDI Pin5 D3 💌
nTRST output inverted nSRST output inverted	TDO Pin11 Busy 💌
	nTRST Pin2 D0 💌
TCK Control	nSRST NO SYS RST 💌
OK	Cancel

Figure 6 Jtag Settings

# PROGRAM DEBUG

This example illustrates the procedure to debug under an evaluation copy of RealView which can be obtained after registration to Keil at <a href="https://www.keil.com/demo/eval/arm.htm">https://www.keil.com/demo/eval/arm.htm</a>. There is a 16KB program code limitation to this evaluation suite but it would be enough for a demonstration. For more details of operation, one may refer to the user guide of one of our evaluation boards for LPC2103 at the following web site.

### http://www.techtoys.com.hk/ARM boards/LPC2103 Eval 1A/LPC2103 Eval 1A.htm

Launch µVision3 to open the Blinky LED project for simplicity. It can be downloaded from the same web site above under Doc 04. Bring up the **Options** window and click on the **Debug Tab**. Check **RDI Interface Driver** and click on **Settings** button. Please make sure the **Run to main()** checkbox and **Load Application at Startup** have been checked.

Device   Target   Output   Listing   User   C/C++   Asm	Linker Debug Utilities	
Limit Speed to Real-Time		
I Load Application at Startup I Run to main() Initialization File:	✓ Load Application at Startup ✓ Run to main() Initialization File:	
Edit	Edit	
Restore Debug Session Settings		
Image: Weakpoints         Image: Weakpoints		
Vatchpoints & PA		
Memory Display	Memory Display	
CPUIDUL Promotor	Driver DI L: Parameter	
SABM DU -d PC2100	SABM DLL Parameter.	
, ,	1 1	
Dialog DLL: Parameter:	Dialog DLL: Parameter:	
DARMP.DLL -pLPC2103	TARMP.DLL -pLPC2103	
OK Ca	ncel Defaults Help	

A click on **Settings** will bring up the window for **RDI Interface Driver Setup**. Browse to H-JTAG.dll file which should have been installed under the H-JTAG server directory. The options **Cache Code** and **Cache Memory** are optional. Click **OK** to exit.

Options for Target 'Target 1'	X
Device     Target     Output     Listing     User     C/C++     Asm     Linker     Debug     Utilities       C     Use Simulator     Settings     Image: Settings     Image: Settings     Image: Settings     Image: Settings	Settings
RDI Interface Driver Setup	
Browse for RDI Driver DLL C:\Program Files\HJTAG.HJTAG.dl	
Browse for ToolConf File	
Debug Cache Options Cache Code Cache Memory Cache Memory	
OK Cancel	Help
DARMP.DLL PLPC2103 TARMP.DLL PLPC2103	
OK Cancel Defaults	Help
Figure 8 RDI Interface Driver Setup for H-JTA	G

Click on *Utilities Tab* to bring up the flash programming option. We need to download the program to LPC2103 prior to debug. Check the option *Use External Tool for Flash Programming* since we are using H-Flasher for download. Browse to H-Flash.exe and check *Run Independent* option.

Server

Options for Targ	t 'Target 1'	
Device Target (	Jutput Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash I	1enu Command	
C Use Target	Driver for Flash Programming	
	IDI Interface Driver 🗸 Settings 🔽 Update Target before Debugging	
Init File:		
Use Externa	Tool for Flash Programming	
Command:	:\Program Files\HJTAG\H-Flasher.exe	
Arguments:		=
L. L.	Run Independent	
	DK Cancel Defaults H	eln

Figure 9 Use External Tool for Flash Programming

Click **OK** to exit the configuration window and go back to main menu. First, under **Flash**, click **Download**.



This action will bring up the H-Flasher application automatically for you. Highlight **Flash Selection** at the left, and choose your target chip, NXP $\rightarrow$ LPC2103 in this case.

H-Flasher		
New Load Save Save	As Options Exit About	
Program Wizard	>> Flash Selection	
1 Flash Selection	. NXP	Vendor: NXP
2 Configuration	- LPC2101	PartNo: LPC2103
3 Init Script	LPC2102	Type: On-Chip Flash
4 Programming	LPC2104 LPC2105	Sector: 8
P H-Flasher Help	LPC2106	Size: 32 KB
	LPC2114 LPC2119	ID: 0x0004FF11
	LPC2124	Width: 8-Bit
	- LPC2125	
	LPC2132	
	- LPC2136	
	LPC2138	

Figure 11 H-Flasher user interface

Highlight **Programming** at the left, and a click on **Check** button on the right side near the top section will detect the microcontroller. It is to the right of **Src File** text box, click on the browse (...) button to search for your target hex code, in my case it is under D:ARM\LPC2103\Keil\Blinky\Blinky.hex.

Please make sure the correct file has been selected. Once everything has been set, click on *Program* button to download the hex code to LPC2103. You may also perform other task like erase, blank check, and confine programming region from the pull-down manual below. This H-Flasher application can run independently therefore it serves as a standalone programming tool for ARM micros out of zero cost!

H-Flasher			
New Load Save Save	As Options	Exit About	
Program Wizard	>> Prog	ramming - LPC2103	
1 Flash Selection	Flash:	LPC2103 0x0004FF11	Check
2 Configuration	Target:	ARM7TDMI-S Little-Endian	
3 Init Script	Tupe:		Deserver
4 Programming	Type.		Frogram
7 H-Flasher Help	Src File:	D:\ARM\LPC2103\Keil\Blinky\Blinky.he>	
	Dist Addr:		
	From:	Entire Chip 🔹	Erase
	To:	Entire Chip	Blank
	Address:	Size:	Read
<u> </u>			

Figure 12 H-Flasher programming

After finished, minimize the H-Flasher application. Go back µVision3, under **Debug→Start/Stop Debug Session** or **Ctrl+F5** for short key to start debugging. You will see the debug window with registers R0-R15, CPSR, and SPSR which are the core of ARM microcontroller at the left panel. A cursor will stop at the first statement which is sysInit() in this case because the **Run to main()** option has been checked under the Debug option. Otherwise program stepping will begin from the startup code Startup.s assembly file. It may be a good chance to step through it to know what happen before main(void) is executed. Out of curiosity, just uncheck the **Run to main()** option and start all over again (no H-Flasher download required) to repeat the process but this time, Startup.s is stepped through.

Press F10 to step over each statement and observe the LED with LED\_ON, LED\_OFF stepped over. It will be switched on and off under your control.

