Exercise 1: Setting up Free toolchain For ARM[®] Cortex[®]-M Application development Eclipse GNU C/C++ GDB Server for Windows



Revision History

Revision	Date	Author	Description
Number			
0.1	040ct2014	JW	First draft.
0.2	26Nov2014	JW	Generalized Eclipse installation.
			Added SEGGER J-Link GDB support.
0.3	04Feb2015	JW	Updated GNU tools section.
			Updated Eclipse section.
			Fixed table of contents links.
0.4	10Aug2016	JW	Updated SEGGER J-Link support to version 6.00e
0.5	13Nov2022	JW	Updated to latest Eclipse (Embedded)
			Updated link to ARM GNU Tool chain (GCC compiler/linker)
			Added NXP example project (LPC824)
			Added SDK for MCUXpresso Config tool
0.6	06Dec2022	JW	Added MAKE support (2.2.1 NOTE 8)

Audience

This document is intended for the ARM Cortex-M beginner and seasoned developers interested in evaluating ARM Cortex-M platforms.



Table of Contents

Revision History	. 2
Audience	. 2
1.0 Installing GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)	.4
2.0 Installing Eclipse	. 5
2.1 Tool chain demo projects	.7
2.1.1 Hello World	. 8
3.0 Installing SEGGER J-Link utilities	18



1.0 Installing GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)

The latest version of GNU tools for ARM embedded processors can be downloaded at <u>Arm GNU</u> <u>Toolchain</u>. Select the *Download Arm GNU Toolchain* button, follow instructions. Once downloaded, use the *gcc-arm-none-eabi-xx.x-20xx.xx-win32.exe* to install the toolchain. Do not include it in your PATH as developers can use many versions of the arm-none-eabi-gcc tools.



Figure 1 Arm GNU Toolchain



2.0 Installing Eclipse

The eclipse IDE requires the JAVA runtime engine so if you do not have JAVA RTE installed to go <u>http://www.java.com/en/download/</u> and install it. Note, as the tool chains function in both 32-bit and 64-bit architecture, install both the 32-bit and 64-bit versions of JAVA.

Note, the Eclipse installer for 2022-09 R includes a JAVA Runtime Environment for Windows, as well as SEGGER J-link support.

Once you have verified that JAVA RTE has been installed go to

https://www.eclipse.org/downloads/packages/release/2022-09/r/eclipse-ide-embedded-cc-developers and download the latest versions of *Eclipse IDE for C/C++ Developers* and install it (Eclipse IDE 2022-09, as of v 0.5 of this document). Select the *Download x86_64* button, follow instructions. Please do not forget to donate to support the Eclipse community.

GF Eclipse IDE for Embedded C/C++ x +					-	0	×
\leftarrow \rightarrow C \textcircled{a} https://www.eclipse.org/downloads/packages/release/2022-09/r/ecl	lipse-ide-embedded-cc-develope	ers A	ŵ	₹⁄≡	œ (
ECLIPSE	Projects Wo	rking Groups	Mem	bers	More	-	
Home / Downloads / Packages / Release / Eclipse IDE 2022-09 / R / Eclipse IDE for Embedded C/C++ Dev	elopers						
Eclipse Installer Eclipse Packages Eclipse Developer Builds 🗸							
Eclipse IDE for Embedded C/C++ Develope	rs	Er	OR/ terprise F	ACL Pack for	E clipse		
Package Description	Download Links		\square	D			
An IDE for Embedded C/C++ developers. It includes managed cross build plug-ins (Arm and RISC- V) and debug plug-ins (SEGGER J-Link, OpenOCD, pyocd, and QEMU), plus a number of templates to create ready to run blinky projects.	Windows x86_64 macOS x86_64 AArch64 Linux x86_64 AArch64		OF		Ļ		1
To avoid compatibility issues with pre 6.x plug-ins, it is recommended to create a new workspace with the new version and import the projects there.	Downloaded 12,962 Times			Downl	oad	J	
This package includes:	 Checksums 	т	e Eclipse l	nstaller 20	022-09 R		
C/C++ Development Tools Git integration for Eclipse	Bugzilla	no	w includes Window	a JRE for vs and Lin	r macOS, iux.		
Detailed features list	Open Bugs: 0			•			
Maintained by: Eclipse Packaging Project	Resolved Bugs: 0		(
	File a Bug on this Package						
	Github	Get I	Eclipse all your fa	vorite de	2022-0 esktop IDE)9	
			ра	ckages.			
	File an issue on GitHub		Downlo	oad x86_	_64		
	New and Noteworthy	Dowr	load Pac	kages 1	Need Help	?	

Figure 2 Eclipse Foundation CDT page,



Installing the Eclipse IDE is straight forward; launch the installer, eclipse-inst-jre-win64. Select Eclipse IDE for Embedded C/C++ Developers (this Eclipse release contains plug-ins that simplify ARM tools setup and embedded application development). As Eclipse is updated regularly, extract the archive in a directory suited to the version, i.e. for this exercise C:\GCC-Eclipse-NXP-DEMO.



Figure 3 Select Eclipse IDE for Embedded C/C++ Developers



Figure 4 Eclipse Installer folders



2.1 Tool chain demo projects

For this exercise we use NXP's *LPCXpresso board for LPC824* (single CORE Cortex-M0+) PN: OM13071, an inexpensive and readily available eval board for ARM Cortex M0+ development. In this section we will describe how to create the Eclipse project "Hello World." Please note that the steps may be exhaustive however they are for the benefit of new users to the Eclipse development environment.

NOTEs:

- This project is limited to ARM[®] Cortex[™]-M0+ core registers and basic modification to the linker script for LPC824 memory map.
- Projects use a Segger J-LINK debug probe.
- Exercises are path dependent.



Figure 5 Demo system block diagram



2.1.1 Hello World

Go to the Eclipse install directory, our path "C:\GCC-Eclipse-NXP-DEMO\eclipse" and launch eclipse.exe, eclipse.exe,

1. In the Eclipse IDE Launcher dialog Workspace field enter "C:\GCC-Eclipse-NXP-DEMO\XPRESSO-LPC824 then select the *Launch* button.



Figure 6 Project workspace

2. Select File->New->C/C++ project:

•	EclipseEmbedded-LPC824 - Eclipse IDE									-	- 0	1	×
File	Edit Source Refactor Navigate	Search Project	Ru	n Window Help									
	New	Alt+Shift+N >	ē:	Makefile Project with Existing Code			티셴ㅋ	51 - 1 5	c? (>	* 5	- 10	5	_
	Open File		C	C/C++ Project							Q	1	1
۵.	Open Projects from File System			Project				• •	문이	×	B "1	-	
	Recent Files	>	C	Convert to a C/C++ Autotools Proje	ct								
	Close Editor	Ctrl+W	0	Convert to a C/C++ Project (Adds C	/C++ Nature)				There i	is no a	ctive edi	or that	t
	Close All Editors	Ctrl+Shift+W	63	Source Folder					provid	les an c	outline.		
(R)	Save	Ctrl+S	6	Folder									
123	Save As		C	Source File									
1	Save All	Ctrl+Shift+S	h	Header File									
	Revert		Ľ	File from Template									
	Maus		G	Class									
-	Rename	F2		Example									
s)	Refresh	E5	-a	Other		Ctrl+N							
~	Convert Line Delimiters To		Ē										
	Print	Ctrl+P											
20	Import												
2	Export												
_	Properties	Alt+Enter											
	Switch Worksnace	>											
	Restart												
	Exit												
_		Problems	×	a Tasks 📮 Console 🔲 Properti	в						T	8 •	
		Description		^	Resource	Path		Location		Туре			
0 iten	is selected									: 0	al m	= 2	• @

Figure 7 New Project



3. Use the *C* Managed Build Template then select Next> button:



Figure 8 Eclipse managed C project

4. Project type *Hello World Arm Cortex-M C/C++* then select the *Next>* button"

•		_	
C Project			
Create C project of selected type			Ż
Project name: HelloWorld			
Use default location			
Location: C:\GCC-Eclipse-NXP-DEMO\XPRESSO-	LPC824\HelloWorld		Browse
Choose file system: default ~			
Project type:	Toolchains:		
GNU Autotools	Arm Cross GCC		
Executable			
Empty Project			
Hello World ANSI C Project			
Hello World Arm C Project			
Hello World RISC-V C Project			
ADuCM36x C/C++ Project			
Hello World Arm Cortex-M C/C++ Project			
Freescale Processor Expert C/C++ Project			
Freescale Kinetis KLxx C/C++ Project			
SiFive RISC-V C/C++ Project			
STM32F0xx C/C++ Project			
STM32F1xx C/C++ Project			
STM32E2xx C/C++ Project			
STM32F3xx C/C++ Project			
STM32E4xx C/C++ Project			
STM32E7xx C/C++ Project			
Hello World Arm OEMU x Pack C/C++ Project			
Hello World RISC-V OEMU xPack C/C++ Proj.			
Shared Library			
Static Library			
Makefile project			
< >			
Show project types and toolshoins and toolshoins	ure supported on the platforms		
Show project types and toolchains only if they a	re supported on the platform		
? < <u>B</u> ack	<u>N</u> ext > <u>F</u> inis	h	Cancel

Figure 9 Default "Hello World!" Project



5. The LPC824 is a Cortex-M0+ device with 32K of FLASH, 8K of RAM and use the internal 12 MHz oscillator. The first four (4) fields will have to be set to those values below then select *Next>*:

•	_		×
Target processor sett	ings		4
Select the target proces	ssor family and define flash and RAM sizes.		4
			_
Processor fore:	Cortex-M0+		~
Clock (Hz):	12000000		
Flash size (kB):	32		
RAM size (kB).	8		
Use system calls:	Freestanding (no POSIX system calls)		\sim
Trace output:	Semihosting DEBUG channel		\sim
Check some warnings			
Check most warnings			
Enable -Werror			
Use -Og on debug			
Use newlib nano			
Use link optimizations			
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cance	el

Figure 10 Platform configuration



6. Use the default folder options for this project select *Next>* and use the default configuration then select *Next>*:

٠	- D	ı x	•	-
Folders Define the project fold	Jers.	Ž	Select Configurations Select platforms and configurations you wish to deploy on	Ď
Include folder: Source folder:]nclude src		Project type: Executable Toolchaims: Arm Cross GCC Configurations:	
System folder:	system		✓ ⊕ Debug ✓ ⊕ Release	Select all
CMSIS library folder:	cmsis			Deselect all
Linker scripts folder:	Idscripts			
Vendor CMSIS name:	DEVICE			
			Ad	lvanced settings
			Use "Advanced settings" button to edit project's properties.	
			Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property pages.	
?	< <u>g</u> ack <u>Next</u> Einish Ca	ancel	(?) < Back Next> Einish	Cancel

Figure 11 Default project folder

Figure 12 Build configurations

7. Ensure the correct Toolchain is correct and path to GCC:

Arm Toolchains Paths	← → ⇒ 8				
Configure the location where various GNU Arm toolchains are installed. The values are stored in the workspace (not in the project). They are used for all build configurations of this project, and override the workspace or global paths.					
Toolchain name: GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)					
Toolchain folder: C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10\bin Browse	xPack				
On macOS use Shift+Cmd+'.' to show the hidden folders while browsing the file system. xpm uses a .content folder	to store the binaries.				

Figure 13 Tool chain paths



Additional note; As this is a managed MAKE project, make.exe must be present in the eclipse build tool path. It is up to the developer where and how to set the PATH variable; this can be done several ways, for ECLIPES managed projects we suggest Properties →C/C++ Build →Settings [Toolchains TAB] (part of the project settings). The down fall to this method when using REPOs, if the development team does not use similar make.exe path the project will fail to build... Alternatively, a "user" batch file for Windows can be created to set the environment and launch Eclipse. If make is not present, it can be downloaded from the web. Once available, set the Build tools folder for this exercise; make.exe is located in C:\Program Files (x86)\GnuWin32\bin.

Properties for HelloWorld		— 🗆 X
type filter text	Settings	⟨¬ ▼ ¬ < 8
 Resource Builders C/C++ Build Build Variables 	Configuration: Debu	ug [Active]
Logging Settings	🛞 Toolchains 🔳	Devices 🟮 Container Settings 🎤 Build Steps 🙅 Build Artifact 🗟 Binary Parsers 🥴 Error Parsers 🔹 🕨
Tool Chain Editor	Name:	GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)
Linux Tools Path	Architecture:	ARM (AArch32) ~
> MCU	Prefix:	arm-none-eabi-
Project Natures Project References	Suffix:	
Run/Debug Settings	C compiler:	gcc
Task Tags > Validation	C++ compiler:	g++
WikiText	Archiver:	ar
-	Hex/Bin converter:	objcopy
	Listing generator:	objdump
	Size command:	size
	Build command:	make
	Remove command:	rm
	Prefer local xpace	cs/.bin path
6	Toolchain path:	C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10\bin
	Build tools path:	(to change it use the <u>global</u> or <u>workspace</u> preferences pages or the <u>project</u> properties page)
		(to change it use the <u>global</u> or <u>workspace</u> preferences pages or the <u>project</u> properties page)
Preferences		– 🗆 X
type filter text	Global Build Tool	s Path 🗇 🖛 🖏
Global Build Tools Pat Global OpenOCD Pat Global pyOCD Path Global OFMI Paths	The locations where specifically, they are Build tools folder:	various Eclipse Embedded CDT build tools are installed. Unless defined more used for all projects in all workspaces. C:\Program Files (x86)\GnuWin32\bin Browse xPack
Glabal DISC V Taalab		Restore Defaults Apply
? 1 1 0		Apply and Close Cancel Apply and Close Cancel

Figure 14 Setting make.exe path!



2.1.1.1 Building and debugger Project

The Eclipse Hello World ARM Cortex-M project provide basic CMSIS support, in our exercise the Cortex-M0+ Core as described in ARM[®] Cortex[™]-M0+ Devices, Generic User Guide.

1. The linker script memory map must be updated to point to the base address of on-board RAM, set:

```
RAM (xrw) : ORIGIN = 0x10000000, LENGTH = 8K
```

```
EclipseEmbedded-LPC824 - HelloWorld/Idscripts/mem.ld - Eclipse IDE
                                                                                                                                                                    ×
<u>File Edit Source Refactor Navigate Search Project Run Window Help</u>
्र 🔡 🖬
🎦 Project Explorer 🗙 🗖 🗖 📝 main.c 📄 mem.ld 🗙
                                                                                                                                                                     æ
                 E ⊈ ♥ 8 28 /*
29 * Memory Spaces Definitions.
                                                                                                                                                                              臣
✓ → HelloWorld
                                                                                                                                                                              ۲
                                       30 *
  > 🔊 Includes

    Need modifying for a specific board.
    FLASH.ORIGIN: starting address of flash
    FLASH.LENGTH: length of flash

                                                                                                                                                                              -
  > 🐸 src
  > 🐸 system
                                      33 * FLASH.LENGIN: length of reash
34 * RAM.ORIGIN: starting address of RAM bank 0
35 * RAM.LENGTH: length of RAM bank 0
  > 🗁 include
  V 🗁 Idscripts
                                      36 *
        libs.ld
                                       37 * The values below can be addressed in further linker scripts
       📄 mem.ld
                                       38 * using functions like 'ORIGIN(RAM)' or 'LENGTH(RAM)'.
        sections.ld
                                      39 */
                                       40
                                       41 MEMORY
                                       1-1

    FLASH (rx) : ORIGIN = 0x00000000, LENGTH = 32K
    RAM (xrw) : ORIGIN = 0x10000000, LENGTH = 8K

                                       46 /
                                            * Optional sections; define the origin and length to match
* the the specific requirements of your hardware. The zero
                                       47
                                       48
                                             * length prevents inadvertent allocation.
                                       49
                                              */
                                       50
                                            CCMRAM (xrw) : ORIGIN = 0x10000000, LENGTH = 0
                                       51
                                            ELASHB1 (rx) : ORIGIN = 0x00000000, LENGTH = 0
EXTMEMB0 (rx) : ORIGIN = 0x00000000, LENGTH = 0
EXTMEMB1 (rx) : ORIGIN = 0x00000000, LENGTH = 0
                                       52
                                       53
                                       54
                                           EXTMEMB2 (rx) : ORIGIN = 0x00000000, LENGTH = 0
EXTMEMB2 (rx) : ORIGIN = 0x000000000, LENGTH = 0
EXTMEMB3 (rx) : ORIGIN = 0x000000000, LENGTH = 0
                                       55
                                       56
                                       57 }
                                       58
                                       59 /*
                                      60 * For external ram use something like:
61 * RAM (xrw) : ORIGIN = 0x64000000, LENGTH = 2048K
                                       62 *
                                       63 * For special RAM areas use something like:
                                                                                                                                                                     - 0
                                                                                                                                                             7 8
                                      🖹 Problems 🗙 🧔 Tasks 📮 Console 🔲 Properties
                                     0 items
                                      Description
                                                                                           Resource
                                                                                                           Path
                                                                                                                                 Location
                                                                                                                                                 Туре
                                                                                 44:34[10]
                                    Writable
                                                          Insert
                                                                                                                                                          🔄 📶 🎓 🎢 🔇
```

Figure 15 Platform memory map



2. Finally build the project, in the **Project Explorer** select *HelloWorld->Build Project*:

EclipseEmbedded-IPC824 - HelloWorld/Idscripts/mem.ld - Eclipse IDE –							×		
File Edit So	ource	Refactor Navigate Search	Project Run V	indow Help ▼ 💽 ▼ 🧏 ▼ 🎦 🖋 🕬 🖉 ■ 🗊 🗎 🖶 🏪 🔌 🕹 ½ ▼ 🗍	- 🏷 🗘	¢	• =	+	
							Q i	B	Ec
Project Exp	lorer	🗙 😐 🖬 🗈 main.c	📄 mem.ld 🗙				-		8
		E 🕏 🍸 🖇 28 /*						^	85
V 😂 HelloW		29 - Hen	ory spaces Det	initions.					۲
> D Inclu		Go Into		 a specific board. tarting address of flash 					_
> 🤒 syste		Onen in New Window		ngth of flash					
> 🁝 Debu		Show In	Alt+Shift+W >	ting address of KAM bank 0					
S inclusion of the second s		Show in Local Terminal	,	an be addressed in further linker scripts					
📄 lit		Conv	Ctrl+C	ke 'ORIGIN(RAM)' or 'LENGTH(RAM)'.					
🗎 m	iß.	Paste	Ctrl+V						
E 36	×	Delete	Delete						
		Source	>	= 0x00000000, LENGTH = 32K					
		Move		= 0x10000000, LENGTH = 8K					
		Rename	F2	and the shares of the set the set to					
	<u>रे</u> न	Import		requirements of your hardware. The zero					
	4	Export		inadvertent allocation.					
		Build Project		IN = 0×10000000, LENGTH = 0					
		Clean Project		SIN = 0×000000000, LENGIH = 0 IGIN = 0×00000000, LENGIH = 0					
	8	Refresh	F5	IGIN = 0x00000000, LENGTH = 0					
		Close Project		IGIN = 0x00000000, LENGTH = 0					
		Close Unrelated Project							
		Build Targets	>	and a second data to the second se					
		Index	>	IN = 0x64000000, LENGTH = 2048K					
		Build Configurations	>	leas use something like:					
		Profiling Tools	>	· · ···			>	1	
	0	Run As	>	nsole X 🗆 Properties 🛛 🗶 🗘 🔐 🔀 🚍 😹 🕞	e 🗉	- 19			
	夺	Debug As	,						
	01	Profile As	,					1	
		Run C/C++ Code Analysis							
	~	Team	>						
		Compare With	>						
		Validate							
		Configure	>						
😂 HelloWorld		Source	>			1	C 🖬	- 7	0

Figure 16 Build Project

EclipseEmbedded-LPC824 - HelloW	/orld/ldscripts/mem.ld - Eclipse IDE	-		×
File Edit Source Refactor Naviga	ate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp			
📑 • 📰 💿 😸 • 🗞 • 🗟 👔	🛔 = 🚳 = 🗗 = 🎯 = 🗱 = O = 💁 = 💁 = 🎒 🛷 = 📴 🖩 🗉 🗳 🕹 🖉 = 🕅 =	· • • • • •	- <> •	1
		G	1 1 1	2
			、 . u	
Project Explorer X	[c] main.c			0 8
Image: Section Section Image: Section Section	<pre>2 // 'Percory Spaces Definitions. 30 * 'Percory Spaces Definitions. 31 * Need modifying for a specific board. 52 * FLASH.OKIGIN: starting address of flash 32 * FLASH.OKIGIN: starting address of flash 33 * FLASH.OKIGIN: starting address of RAM bank 0 34 * FLASH.OKIGIN: starting address of RAM bank 0 35 * The values below can be addressed in further linker scripts 35 * The values below can be addressed in further linker scripts 36 * * * * * * * * * * * * * * * * * * *</pre>	e • •	, , ,	
				× .
•				 Image: Image: Ima

Figure 17 Successful build



3. The hardware debug interface must be configured before used. From the button bar select the debug ICON, **
drop list and from the list box select *Debug Configurations...*

EclipseEmbedded-LPC824 - HelloWorld/Idscripts/mem.Id - Eclipse IDE					
File Edit Source Refactor Navig	jate Search Project Run Window Help				
	📸 ▾ 🚳 ▾ 💽 ▾ 🞯 ▾ ! 🞋 ▾ 🔘 ▾ 🚱 ▾ 🂁 ▾ ! 🎒 🖉 ▾ ! 🔛 🗐 👖 ! 🖳 ! 🏪 ! 🍬 🕹 ! ½ ▾ 🖓 ▾ 🏷 😅				
	(no launch history)				
陷 Project Explorer 🗙 🗖 🗖	i main.c				
E 🕏 7 🕴	28 /* Debug Configurations				
✓ ﷺ HelloWorld	29 * Memory Spaces 30 * Organize Favorites				
> 🎎 Binaries	31 * Need modifying for a specific board.				
> 🔊 Includes	32 * FLASH.ORIGIN: starting address of flash				
> 📇 src	33 * FLASH.LENGTH: length of flash				
> 😕 system	34 * RAM.ORIGIN: starting address of RAM bank 0				
> 📂 Debug	36 *				
> 🗁 include	37 * The values below can be addressed in further linker scripts				
🗸 🗁 Idscripts	38 * using functions like 'ORIGIN(RAM)' or 'LENGTH(RAM)'.				
libs.ld	39 */				
mem.ld	40				
sections.ld	41 MEMORY 42 {				

Figure 18 GDB Hardware Configuration

4. From the items list select *GDB SEGGER J-Link Debugging* and double click it to bring up the HelloWorld debug configuration dialog:

Debug Configurations		_		
Create, manage, and run configurations			Ś	
Image: Solution in the second seco	 Configure launch settings from this dialog: Press the 'New Configuration' button to create a configuration of the selected type. Press the 'New Prototype' button to create a launch configuration prototype of the selected type. Press the 'Export' button to export the selected configurations. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Delete' button to configure filtering options. Select launch configuration(s) and then select 'Link Prototype' menu item to link a prototype. Select launch configuration(s) and then select 'Willink Prototype' menu item to unlink a prototype. Select launch configuration(s) and then select 'Reset with Prototype Values' menu item to reset with prototype values. Edit or view an existing configuration by selecting it. 			
Filter matched 17 of 17 items				
?	Depu	ıg	Close	

Figure 19 Select SEGGER J-LINK GDB configuration



 From the List Box select *HelloWorld Debug* then select the *Debugger* tab. In the *Device name* field enter LPC824M201 and in the *USB serial or IP name/address* field enter your J-Link product serial number. Select the *Debug* button and agree to the debug perspective. The developer is encouraged to investigate other tabs for available configuration.

Debug Configurations Create, manage, and run configurations				- □ ×
	Name: HelloWorld De	ehua		
tune filter text				
	I Link CDR Server Se	ger 🗭 startup 🦃 source 🔝 Commo	on 🚡 SVD Path	
C/C++ Application	Start the Link (etup GDB senver locally	Connect to running target	
C/C++ Container Launcher				N . 11
C/C++ Postmortem Debugger	Executable path:	Sulink_path}/Sulink_gdbserver}	Bro	variables
C/C++ Remote Application	Actual executable:	C:\Program Files (x86)\SEGGER\JLink/J	ILinkGDBServerCL.exe	
GDB Hardware Debugging		(to enange it use the globar or despace	ce preferences pages or the <u>project</u> prope	erties page)
C GDB OpenOCD Debugging	Device nan e:	LPC824M201	Sup	ported device names
C GDB PyOCD Debugging	Endianness:			
C GDB QEMU aarch64 Debugging	Connection:		(US) serial or IP nar	me/address)
GDB QEMU gnuarmeclipse Debugging (Der	Interface:	● SWD ○ JTAG		
GDB QEMU riscv32 Debugging	Initial speed:	○ Auto ○ Adaptive ④ Fixed	1000 kHz	
GDB QEMU riscv64 Debugging	GDB port:	2331		
GDB SEGGER J-Link Debugging	SWO port:	2332	Verify downloads 🔽 Initial	ize registers on start
Launch Group	Telnet nort:	2333	I ocal host only Silent	
	icine por			
	Log file:			Browse
	Other options:	-singlerun -strict -timeout 0 -nogui		
	Allocate console	e for the GDB server	Allocate console for semihosting an	nd SWO
	GDB Client Setup			
	Executable name:	{cross_prefix}gdb{cross_suffix}	Bro	wse Variables
	Actual executable:	arm-none-eabi-odb		
	Other antians:	[]]		
	Commands:			
	Commands:	set mem inaccessible-by-default off		<u>^</u>
				~
	Remote Target			
	Host name or IP ad	Idress: localhost		
	Port number	2331		
		2551		
	Force thread list up	pdate on suspend		
				Restore defaults
Filter matched 17 of 17 items			Re⊻ert	Apply
0			Debu	g Close

Figure 20 SEGGER J-Link hardware configuration for GDB.



6. Finally run the project from the button bar select the resume button or F8 to run the project. Note the output in the console window:

EclipseEmbedded-LPC824 - Hello	World/sro	:/main.c - Eclip	se IDE						_		×
<u>File Edit Source Refactor Navi</u>	gate Se	<u>a</u> rch <u>P</u> roject	<u>Run W</u> indow <u>H</u> elp			-					
i 📑 ▼ 🗒 🕼 i 🏘 ▼ 🚺 ▼ '	% ▼ i Ø	9 🔗 🔻 🏄	🖳 🔌 🕩 💷 🔳	N 3. O .r. i + =	5 🖂 🕹 🕍 🔻	 ∱ ▼	*> <>	<-> ▼ <-	> + 🛃		
									Q	. 18	10 *
🎋 Debug 🗙 🏠 Project Expl		🚺 main.c 🗙	📄 mem.ld 🛛 💽 r	main() at main.c:58 0x3a4	-		(x)= V	× 💁 B	ዥ E	묾 P	
E 🔌 🗆	i ⇒ 8	46 // 47 //				^			£_ ⇒t	6 📬	🖻 🖇
HelloWorld Debug [GDB SEGGE] HelloWorld elf	R J-Link	48					Name	Ţ	ype	Value	
Thread #1 57005 (Running)	g : User F	49⊖// Sam 50 // the	ple pragmas to cop end of this funct	pe with warnings. Ple tion, used to pop the	ase note the re. compiler diagn						
JLinkGDBServerCL.exe		51 #pragm	a GCC diagnostic p	oush	amaton"						
arm-none-eabi-gdb		53 #prage	a GCC diagnostic i	ignored "-Wmissing-de	clarations"						
jag serning and svvv		54 #prag	na GCC diagnostic i	ignored "-Wreturn-typ	e"						
		560 int									
		57 main (58 {	int argc, char* ar	rgv[])							
		59 // N	Normally at this st	tage most of the micr	ocontroller sub						
		60 // t	rom the startup fi	ile, before calling m	ain().						
		62 // (see system/src/cor	rtexm/initialize-hard	lware.c)						
		64 // c	or add the addition	nal initialisation he	re, for example						
		65 // 66 // H	AL Init():								
		67									
		68 // 1 69 // i	n this sample the if you do not add t	bysteminit() function the clo	on is just a pla ock will remain o						
		70 //t	the reset value, us	sually a relatively]	ow speed RC clo						
		72 // 5	end a greeting to	the trace device (sk	ipped on Release						
		73 trac 74	e_puts("Hello Arm:	World!");							
		75 // A	At this stage the s	system clock should h	ave already been						
		76 // a 77 trac	e_printf("System c	clock: %u Hz\n", Syst	emCoreClock);						
		78 79 +ime	er start ():								
		80									
		81 int 82	seconds = 0;								
		83 // I	infinite loop								
		85 {	(L)								
		86	timer_sleep (TIMER	R_FREQUENCY_HZ);	-	~					\lor
	- 1	<		-	>	-	<				
		🖳 Console 🔅	🗙 🚻 Registers 🔡 Pr	roblems 🜔 Executables	Rebugger Consc	le []	Memory				
		HelloWorld De	hug IGDB SEGGER I-Link	k Debugging]			6 % 1	ak 🔠 🗷			• 🖸 •
		SEGGER J-Li	nk GDB Server V7.5	64b - Terminal output	channel						~
		Hello Arm W System cloc	lorld! k: 12000000 Hz								
		Second 1									
		Second 2 Second 3									
		Second 4									
		Second 6									
		Second 7 Second 8									
		Second 9									
		Second 10 Second 11									
											~
<	>	1									
	Writable		Smart Insert	84:12:3304					Ð	10 🖻	7 🛇

Figure 21 Debug Console output



3.0 Installing SEGGER J-Link utilities

The SEGGER J-Link hardware and software utilities allow developers the means to download firmware to the target platform as well as debug applications in the target platform. As of version 0.4 of this document we will describe how to install and set up SEGGERS's J-Link ARM versions 6.00e. The process outlined below is the same regardless of SEGGER J-Link utilities version.

Go to <u>http://www.segger.com/jlink-software.html</u> and from the J-Link software & documentation pack for Windows select the Download button. Download the archive and extract it in the directory of your choice then launch the setup allocation. Follow instruction to completion.

Once installation has completed connect the J-Link device to the PC, USB enumeration should take place. Once Windows has enumerated the device, connect the J-Link to the target platform (in this case the NXP LPC824 eval board) and launch the J-Link GDB server (type J-Link GDB Server in the windows search box). Alternatively select the windows start Icon the scroll to SEGGER – J-Link Vx.xxx and choose J-Link GDB Server Vx.xxx.

The GDB Server Configuration dialog will appear, see (Figure 21 SEGGER GDB server Configuration). Ensure the following are set:

- Connection to J-Link: USB
- Target interface: SWD
- Speed: Auto selection
- Target device: Note, the developer must select the device, in our example the NXP LPC284 Xpresso evaluation board is the target so we will select the LPC824 MCU. Click the navigation button and the device explorer dialog will appear see (Figure 22 Target Device Settings: NXP LPC284).



🔝 SEGGER J-Link GDB Server V7.82c Config 🛛 🗙
Connection to J-Link
USB
O ICP/IP
Target device
LPC824M201
Little Endian 🔻
Flash banks
BaseAddr Name Loader
0x00000000 Internal flash Default 🔻
Target interface
SWD
Speed
Auto Selection
Script file (optional)
Server settings
Init registers
✓ Localhost only
Generate logfile
GDB port 2 2331
SWO port 2 2332
Telnet port 2 2333
Command line option
-select USB=600000394 -device LPC824M201 -endian little - if SWD -speed auto -noir -LocalhostOnly -nologtofile -port 2331 -SWOPort 2332 -TelnetPort 2333
<u>QK</u> <u>C</u> ancel

Figure 22 SEGGER GDB server Configuration

Manufacturer	Device	Core	NumCores	Flach Size	
	V 1PC824		Filter	Filter	
IXP	LPC824M201	Cortex-M0	1	32 KB	

Figure 23 Target Device Settings: NXP LPC284

Once the correct device has been selected select *OK* on the Target device setting dialog then select the *OK* button on the GDB server configuration dialog. If the J-Link device successfully connects to the target the SEGGER J-Link GDB Server dialog will appear see (Figure 23 SEGGER J-Link GDB Server: Connection StatusDialog). Congratulation you are ready to debug...

SEGO	GER J-Link GDB Server V7.82	c				-		×
File He	elp							
	- 1							
GDB	Waiting for connection					Star	y on <u>t</u> op	
J-Link	Connected		SWD	4000 kH	z	Sho	w log <u>w</u> ind	wob
Device	LPC824M201 (Halted)		3.32V	little end	lian	🗌 🗌 Gen	erate logf	ile
Logfile	(Not enabled)					🗹 🗹	fy downloa	ad
<u>C</u> lear l	Log							
SEGGER	J-Link GDB Server V7.82	c GU	I Version					
JLinkA	RM.dll V7.82c (DLL compi	led	Nov 16 2022 17:01:56)					
G0	OB Server start settings		-					
GDBInit	t file:	non	ê •					
SWD nav	woutput listening port:	233	2					
Termina	al I/O port:	233	3					
Accept	remote connection:	100	alhost only					
Generat	te logfile:	off						
Verify	download:	on						
Init re	egs on start:	off						
Silent	mode:	off						
Single	run mode:	off						
Target	connection timeout:	500	0 ms					
	D-Link related settings-		-					
J-Link	Host interface:	OZR	_					
J-Link	script:	non	e					
J-LINK	Sectings file.	non						
Target	device:	I PC	- 824M281					
Target	device parameters:	non	e					
Target	interface:	SWD						
Target	interface speed:	400	8kHz					
Target	endian:	lit	tle					
Connect	ting to J-Link							
Select:	is connected.	Flar	h hank no @ ('Internal fla	ch' A o	(av			
Selecte	car 1 link V10 compiled	105	12 2022 14:50:26	1511 (8.6	xo).			
Handwar	ce: Via aa	sep	22 2022 17.39:30					
S/N:								
Feature	e(s): RDI, FlashBP, Flas	hDL,	JFlash, GDB					
Checkin	ng target voltage		-					
Target	voltage: 3.32 V							
Listeni	ing on TCP/IP port 2331							
Connect	ting to target							
Connect	ted to target							
Waiting	g for GDB connection							
0 bytes do	wnloaded				Connected to target			

Figure 24 SEGGER J-Link GDB Server: Connection Status

