Exercise 2: Eclipse / ARM GCC Toolchain MCUXpresso SDK and Config Tools for LPCXpresso824MAX board under Windows



Revision History

Revision Number	Date	Author	Description
0.1	040ct2014	1\\/	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

Audience

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

Acknowledgements

The authors have been supported in this work by NXP Semiconductors. Select images in this text were reprinted with the permission of NXP Semiconductors.



Table of Contents

Revision History
Audience
Acknowledgements2
1 NXP SDK for LPCXpresso 824MAX board4
1.1 Importing SDK
1.2 Configuration
1.2.1 Clock Tree Configuration13
1.2.2 Pin Configuration17
1.2.3 Peripheral Configuration19
1.2.4 Code Generation
2 Hardware Requirements
2.1 LPC824Max modifications22
3 Importing project into Eclipse
3.1 Cross Assembler FLAGS27
3.2 C Cross Compiler FLAGS
3.3 Cross Linker FLAGS29
3.4 Include Paths



1 NXP SDK for LPCXpresso 824MAX board

This exercise picks up on the Eclipse/ARM free tool chain set up document. The focus is on how to generate a GCC NXP Xpresso SDK for the LPCXpresso 824MAX board and import it into the Eclipse IDE for Embedded C/C++ Developers. Using this code generation tool aids in navigating the MCU platforms user manual, provides access to the complete register set and more important provides a graphical representation of the MCU clock tree!

NOTE: NXP SDK generator will deliver the GCC sources as a CMake based project. For this exercise we will not use CMAKE scripts and batch files. We will import the sources as an Eclipse *C Managed Build* project. We will explore CMake managed projects on our next document Exercise 3 *How to import an Empty or Existing CMake Project into the Eclipse IDE.*

- 1. Go to NXP.com, if you do not have a login account create one
- 2. Login to your account, from the NPX home page navigate to:

PRODUCTS → ARM MCUs → General Purpose MCUs → LPC800 Arm Cortex-M0+ This will take you to the LPC800 Series landing page:



Figure 1 © 2022 NXP B.V.



3. From the landing page select LPC82x MCU Family, scroll down to Software and select the *MCUXpresso Software Development Kit (SDK)* **DOWNLOAD OPTIONS** button:

Software		
 NXP (8) Partners (23) 		Sort by Relevance V
FILTER BY Embedded Software BSP, Drivers and Middleware Development Software IDE and Build Tools	BSP, DRIVERS AND MIDDLEWARE MCUXpresso Software Development Kit (SDK) FEATURED BSP, DRIVERS AND MIDDLEWARE MCUXpresso Config Tools - Pins, Clocks, Peripherals FEATURED IDE AND BUILD TOOLS MCUXpresso Integrated Development Environment (IDE) FEATURED	DOWNLOAD OPTIONS
	Additional software available. View our featured partner solutions. BSP, DRIVERS AND MIDDLEWARE LPC824 Example Code Bundle Keil ZIP Rev 1 Aug 29, 2016 817.0 KB LPC824-EX-CODE-KEIL	DOWNLOAD L
	Sign in required BSP, DRIVERS AND MIDDLEWARE LPCOpen Software Development Platform LPC8XX	DOWNLOAD OPTIONS

Figure 2 © 2022 NXP B.V.



 From the MCUXpresso Software Development Kit (SDK) landing page select the MCUXpresso SDK

 SDK Builder DOWNLOAD button. Once at the SDK Builder page click the Select Development Board button:

Downloads		
NXP (1)	√ Filter by keyword	
FILTER BY Embedded Software BSP, Drivers and Middleware	1 download BSP, DRIVERS AND MIDDLEWARE MCUXpresso SDK - SDK Builder EXTERNAL Rev 1.0 Mar 2, 2017 MCUXPRESSO-SDK	

Figure 3 © 2022 NXP B.V.

NP				
(MCUXpresso Six brings open s applications to speed your software d specific to your processor of evaluate to your software d sector by our processor of evaluate sector by our processor of evaluate More strangly recommends you update address for Bluetooh Low Energy so	SDK Builder source drows, modifieware, and reference example towicpment. Castornice and download an SDK to toated selectors. A Access My SDK Dashboard ate to the lattest version of MCUXpresso SDK that contains en enhance stack implementations.	unital socurly	SDK Gecure Prov
	OVERVIEW	SOFTWARE AND TOOLS	DEVELOPER RESOURCES	
	Getting started with MCL Do you have a development boar Start by clicking on Select Development	JXpresso SDK is simple. rd? I Board to download a customized SDK for that specific platfic	xm.	
	Are you returning and seeking pr Click on Dashboard for quick access to	reviously downloaded SDKs? your history of downloads.		

Figure 4 © 2022 NXP B.V.



5. In the *Search for Hardware* field enter *LPCXpresso824MAX* select the board then select the **Build MCUXpresso SDK** button [vX.XX.X] (Located on the right side of the landing page):

← → C @ 0	https://mcuxpresso.nxp.com/en/select	A* G 🕼 🖷 🌒 …
NXP MCUXpresso	SDK Builder	3 9 2 4 4) 1
SOK Dashboard SOK Dashboard BUILD SOK Select baard / Processor Middeware (0) Camples (0) Totchain (0f) Processor Parametrics (0) ADMINISTRATION Notifications Preferences DYMMI AIDS	SDK Builder Select Development Board Search for your board or kit to get started. Search for Hardware LPCxpresso824MAX Select a Board, Kit, or Processor LPCxpresso55516 (LPC55516) LPCxpresso55556 (LPC55528) LPCXpresso55558 (LPC55528)	Selection Details
MCUXpresso IDE MCUXpresso IDE Coniji Tool Omine data MCUXpresso Secure Provisioning Tool	LPCXpressosS46 (LPCSSS46) LPCXpressos0526 (LPCSSS69) LPCXpresso802 (LPC802) LPCXpresso804 (LPC804) LPCXpresso812MAX (LPC812) LPCXpresso824MAX (LPC824) LPCXpresso824MAX (LPC845) MW	Build MCUXpresso SDK v2 12 0 Boaru Conferention Matched Hardware Platforms Found Conference Found Conference Boards: (22), Kills: (23), Processors: (47)
	ON dsc ds	Filtering Criteria - Reset all Required Middleware Middleware filtering not applied Required Example Projects Example Project filtering not applied Required Toolchains Toolchains filtering not applied Processor Parametric Filtering Processor Parametric Filtering not applied

Figure 5 © 2022 NXP B.V.



6. With this exercise the GNU GCC tools are used, ensure the Toolchain/IDE the GCC ICON is highlighted. Select the **DOWNLOAD SDK** button, this will generate the SDK and once complete will take the developer their SDK Dashboard page, select the download ICON for the LPCXpresso824MAX (GCC ARM Embedded toolchain). Select the *Download SDK* icon in the *SDK_2.xx.x_LPCXpresso824MAX* Archive:

← C A 🗅 https://mauxpresso.nxp.com/en/builder?hw=LPCXpresso824MAX A ^t G 🏂 🏟					
NCO MCUXpresso SDK Builder 😵 🗩 🖪 🔺					
SDK Dashboard BUILD SDK Select Board / Processor Middleware (0) Contain (0f) Processor Parametrics (0f) ADMINISTRATION Notifications	Build SDK for LPCX presso824MAX Generate a downloadable SDK archive for use with desktop MCUX presso Tools. Developer Environment Setting Selections here (oversting hod system, tookhain or middleware) will impact files and examples projects included in the SDK and Generated Projects Hot OS Met OS Search	SDK 2.12.0 (released 2022-07- Version 14)			
Preferences	Name Category Description	Dependencies			
DOWNLOADS MCUXpresso IDE MCUXpresso Config Tools	CMSIS DSP Library CMSIS DSP Lib CMSIS DSP Lib CMSIS DSP Software Library DOWNLOAD SDK				
Offline data MCUXpresso Secure Provisioning Tool	Can't find the middleware you are looking for? Let's try to use "Filtering Criteria" optionst By clicking at Middleware selection page you can find and select desired middleware you are looking for. C up as required for your configuration. When Middleware is set as required, the right side under 'Matched Hardware Platforms' section easily allows you your criteria by one click (not just for Middleware only).	In Middleware selection page you can set it to see results of HW platforms matching			

Figure 6 © 2022 NXP B.V.

← C @ bttp	ps://mcuxpresso.nxp.com/en/dashboard?uvid=4774588tto_vault=true				A [∿] t∂	ć 🖻 😩	
	SDK Builder				o 🗩	E 🔺 🕫	
SDK Dashboard	MCUXpresso SDK Dashboard Access, Download, and Share your requested SDK Builds.		Showing 1 of 1 Archives	[Search	Q	
Orient Lobert Treatment Moderware (0) Toolchare (0)		SDK_2.12.0_LPCXpresso824MAX	[‡] ≇ 2.12.0 ∰ <u>NXP.com</u> ®	LPCXpresso824MAX	2022-12-24	Remove SDK Rebuild SDK Config Tools Share SDK Download SD	
MCUXpresso IDE MCUXpresso Config Tools Offline data MCUXpresso Secure Provisioning Tool							

Figure 7 © 2022 NXP B.V.



7. From the **Download** dialog select Download SDK Archive including documentation [Agree to the Software Terms and Conditions]. Download the archive to a location to be accessed later:

MCUXpresso SDK Dashboard Access, Download, and Share your requested SDK Builds.	Showi	ng 1 of 1 Archives		Search	Q
	SDK_2.12.0_LPCXpress0824MAX Windows VCCCARM Embedded 10-021.10 None And SDK Description	2.12.0 NXP.com ⁶⁷	LPCXpressel2dMAX	2022-12-24	Remove SDK Rebuild SDK Config Tools Share SDK Share SDK
	Downloads MCUXpresso SDK Download SDK Archive including docume Download Statustance Example Project Online Documentation View SDK API Reference Manual MCUXPresso Config Tools Download Config Tools Data	ntation (20 MB)			

Figure 8 © 2022 NXP B.V.

 Return to the MCUXpresso page, <u>https://mcuxpresso.nxp.com/en/welcome</u> and select the CFG icon , this will take you to the MCUXpresso Config tools landing page. The Config tool is a GUI that use the SDK to easily configure and generate C sources for the LPC824. Note: Bookmark the MCUXpresso welcome page as this page provides quick button access to all NXP MCUXpresso tools.

	DE HE A (2) (2 - CFG) (2 -
MCUXpresso SDK Builder The MCUXpresso SDK brings open source drivers, middleware, and reference example applications to speed your software development. Customize and download an SDK	
specific to your processor or evaluation board selections. Solect Development Board Access My SDK Dashboard NKP stongly recommends you update to the latest version of MCUXpresso SDK that contains essential security updates for Bluetooth Low Energy software stack implementations.	Security Security (IDE) Config

Figure 9 © 2022 NXP B.V.



9. Select the DOWNLOADS button, once at the Downloads section select the *CODE GENERATION TOOLS <MCUXpresso Config Tools, Windows Installer [FEATURED]>* to begin the download:



Figure 10 © 2022 NXP B.V.

Downloads		
• NXP (35)	√ Filter by keyword	
FILTER BY	1-5 of 35 downloads	Sort by Relevance ~
Embedded Software BSP, Drivers and Middleware Development Software	CODE GENERATION TOOLS MCUXpresso Config Tools, Mac OS Package FEATURED PKG Rev 12.1 Sep 29, 2022 316753 KB MCUXPRESSO-CT-MAC-V12.1 Sign in required	DOWNLOAD
Code Generation Tools	CODE GENERATION TOOLS MCUXpresso Config Tools, Linux DEB Package FEATURED BIN Rev 12.1 Sep 29, 2022 321415 KB MCUXPRESSO-CT-LINUX64-DEB-V12.1 Sign in required CODE GENERATION TOOLS MCUXpresso Config Tools, Windows Installer FEATURED EXE. Rev 12.1 Sep 29, 2022 334659 KB MCUXPRESSO-CT-WIN64-W12.1	DOWNLOAD
	Sign in required BSP, DRIVERS AND MIDDLEWARE MCUXpresso SDK - SDK Builder EXTERNAL Rev 1.0 Mar 2, 2017 MCUXPRESSO-SDK	DOWNLOAD 🖉
	CODE GENERATION TOOLS MCUXpresso Config Tools, Windows installer EXE Rev 8.0 Jul 22, 2020 202453 KB MCUXPRESSO-CT-WIN64-V8 Sign in required	DOWNLOAD

Figure 11 © 2022 NXP B.V.



10. Please download the *MCUXpresso Config Tools User's Guide* for detailed instructions on installation and use:



Figure 12 © 2022 NXP B.V.



1.1 Importing SDK

After successful installation of the MCUXpresso Config tools, unpack the SDK for LPCXpresso824MAX board downloaded in section 2 (our path C:\NXP_DEMO) then launch the MCUXpresso Config tools. By default, the *Start development* dialog will appear. From this dialog select the *Create a new configuration and project based on an SDK example or a "hello world" project* radio button and select the Next button:

Start development			×
Select a way to start development			
In Create a new configuration and project based on an SDK example or a "hello world" project			
Use this option to clone an SDK example or create a hello world project for a supported IDE/toolchain			
O Create a new configuration based on an existing IDE/toolchain project			
Select toolchain project		Brow	se
Use this option to create the Pins, Clocks, and/or Peripherals configuration of an existing Keil uVision, IAR Embedded Workbench, CodeWarrior for MCU with SDK, or ARM GCC project. Once created, this option will support directly updating the Pins, Clocks, and Peripheral files associated with the IDE/toolchain project.			
Open an existing configuration			
Use this option to edit an existing configuration file (.mex).			
Select an existing configuration (*.mex)	~	Brow	ise
Always open last configuration			
Create a new standalone configuration for a processor, board, or kit Use this option to create a new frame. Clocks, and/or Peripherals configuration for a selected processor or board without association to a toolchain project. Generated source code can be exported to a specified folder. It is possible to associate the configuration for a selected processor or board without association to a toolchain project. Generated source code can be exported to a specified folder. It is possible to associate the configuration for a selected processor or board without association to a toolchain project. Generated source code can be exported to a specified folder. It is possible to associate the configuration for a selected processor or board without association to a toolchain project. Generated source code can be exported to a specified folder. It is possible to associate the configuration for a selected processor or board without association to a toolchain project. Generated source code can be exported to a specified folder. It is possible to associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board without associate the configuration for a selected processor or board or board by Config Tools into the toolchain project directory and then open it using the "Open an existing configuration".	ration to a	any	
< <u>Rack</u> <u>Next></u> Einish		Cance	el

Figure 13 © 2022 NXP B.V.

The next dialog will allow us to select project paths and options. For this exercise we will work in the C:\NXP_DEMO folder, this is where the SKD archive was unpacked. In the *SDK Path* group use the Browse... button to navigate to the SDK folder. Once the path is set the *Toolchain* group should have only one option "*GCC ARM Emedded XX-XXXX.XX*" if not use the dropdown to select the correct toolchain. If the GCC ARM toolchain is not available go to 2 and start again... In the *Action* group select the *Clone selected example for board kit* radio button and in the *SDK Example* group choose the *led_blinky* select C:\NXP_DEMO as the working directory and final select the Finish button:

Start development		-		×
Create a new configuration and project based on an SDK example or a "hello world" project				
Clone project(s): led_blinky				
SDK Path		SDK Example		
C:\NXP_DEMO\SDK_2_12_0_LPCXpresso824MAX v	Browse	type filter text		
SDK can be downloaded from https://mcuxpresso.nxp.com		V LPCXpresso824MAX board		
Toolchain		✓ demo_apps hello world		
GCC ARM Embedded 10-2021.10	~	led_blinky		
Artion		power_mode_switch_lpc > driver_examples		
Create "hello world" project for LPCXpresso824MAX board		,		
Clone selected example for board or kit				
		Base project directory (workspace)		
		C:\NXP_DEMO	✓ Brov	vse
		Project name		
		led_blinky		
		< <u>₿ack</u> Next > Einish	Cano	el:

Figure 14 © 2022 NXP B.V.



1.2 Configuration

The *Config Tools Overview* dialog will appear once the base code configuration has been generated. We will use this dialog to initialize the system clocks, required peripherals and pin assignment.

1.2.1 Clock Tree Configuration

By default, the clock tool is disabled, enable the clock tool by selecting the slider button then click on the Clocks icon:

✓ Configuration - HW Info							
Processor: LPC824							
Part number: LPC824M201JHI33							
Core: Cortex-M0P							
Board: LPCXpresso824							
SDK Version: ksdk2_0							
Clocks Configures the initialization of the core, system, bus, and peripheral clocks.							
✓ Generated code							
└ Update code enabled							
board\clock_config.c							
board\clock_config.h							
Functional groups BOARD_BootClockRUN							

Figure 15 © 2022 NXP B.V.

This will bring up the *Clocks Diagram* tab, a GUI that can be used to initialize all system clocks. Using this GUI we will initialize the system clock to use the 12 MHz crystal oscillator enable the PLL to boost the clock to 60 MHz then divide it by two (2) resulting in a system clock of 30 MHz (maximum for the LPC824). We will then direct the **main_clk** to a CLKOUT PIN, divide by 60 MHz/250 = 240 kHz to verify the system clock.



1) Connect the XTALIN and XTALOUT PINs; right click on the **sys_osc** block and select *connected* from the drop down:



Figure 16 © 2022 NXP B.V.

2) Right click on the SYSPLLCLKSEL MUX and from the list box select Crystal Oscillator (SYSOSC):



Figure 17 © 2022 NXP B.V.



3) For the next setting we will use the *Details* tab (located to the right of the clock tree diagram). Double click on the **PLL** block, set **M_MULT** to * 5 (boosting the oscillator clock to 60 MHz):

Overview 🔓 Code Preview	Registers	\blacksquare Details \times	🗘 Clock Consumers	
Element Details: SYSCON.PLL				
Name	C L	Value		Accuracy
🖃 PLL				
PLL Frequency		60 MHz		
PLL	Ī	Power-up		
N_DIV	🔒 /	/ 1		
M_MULT	•	* 5		

Figure 18 © 2022 NXP B.V.

4) Next double click on the MAINCLKSEL MUX and from the list box select pll_clk:



Figure 19 © 2022 NXP B.V.



5) Once this select has been mase the toll will automatically set the **SYSAHBCLKDIV** to two (2) producing the desired 30 MHz system clock:





6) From the **CLKOUTSEL** MUX select **main_clk** and set the CLKOUTDIV block and divide by 250) to produce a 240 kHz clock signal at this pin:



Figure 21 © 2022 NXP B.V.

7) The final clock to configure is the UART; in the **UARTCLKDIV** block set the divide value to 30 resulting in a 2 MHz reference for the all UARTS:



```
Figure 22 © 2022 NXP B.V.
```



1.2.2 Pin Configuration

Once the clock trees have been configured, we need to set the PIN MUX (IOCON) to route the XTAL, CLKOUT and UART pins. In the upper right corner of the Config Tools select the PIN configuration icon:





1) By default, PIO0_12 will be configured as GPIO (for the red led "blinky"). In the identifier column of the PIN table select the RED_LED from the list box:

	🔁 Routing Details 🖃 🖻 🗖																
Pin	Pins Signals (P type filter text																
Ro	uting	Details for BO	ARD_InitPins	1	• • •												
#		Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO initial state	Mode	Invert	Hysteresis	Open drain	Digital filter	CLK_DIV	DAC Mode	I2C Mode
	2	GPIO	PIO0, 12	->	[2] PIO0_12	SW2/D1[1]/P3[45]/J2[5]/P0_12-RED-ISP	LED_RED 🔺	Output	Logical 0	PullUp	Disabled	Enabled	Disabled	Bypass	DIV0	n/a	n/a
							LED_RED										
							SW2	Din identifier	used for #define co	de generation							
							Not Specified		used for #define co	de generation							
<																	

Figure 24 © 2022 NXP B.V.



2) Next, we will configure the balance of pins. From the Peripheral Signals tab select the **SWD**, **SYSCON** and **UARTO** check boxes. From the SWD enable the SWCLK [PIN6/PIO0_3] and SWDIO [PIN7/PIO0_2] to dedicate these pins to the debug interface. From the SYSCON enable the XTALIN [PIN18/PIO0_8], XTALOUT [PIN17/PIO0_9] for the crystal oscillator connection. From the UARTO select TXD [PIN30/PIO0_19] and RXD [PIN29/PIO0_20] and route the CLKOUT to [PIN24/PIO0_0]:



Figure 25 © 2022 NXP B.V.



1.2.3 Peripheral Configuration

By default, the SDK provides code for a console, this will be connected to peripheral UARTO. Enable peripheral configuration then right click on the Peripherals icon.

V Peripherals
Configures the initialization of the SDK peripheral drivers.
✓ Generated code
Update code enabled
board\peripherals.c
board\peripherals.h
V Functional groups
BOARD_InitPeripherals

Figure 26 © 2022 NXP B.V.

Once open select UARTO and modify the Baud rate field to 115200.

	🛕 💈 Update	e Code 🔻 Functional Group BOARD_InitPeripherals	P 🕞 🖉 💝 🗄 🕇		
	omponents 🦞	Peripherals ×	🖞 USARTO 🗙		- 8
Y	type filter text	11	Universal Synchronous	s/Asynchronous Receiver/Transmitter (USART) [Peripheral drivers (Device specific)]	
Peri	pheral	Used in	Name USART0		Custom name
H	ACOMP ADC0		Mode Polling	Peripheral USART0	~
	CRC		 General configuration 		Preset Custom y
H	DMA0 GPIO				
	I2C0		Clock source		· · · · · · · · · · · · · · · · · · ·
	12C1		Clock source frequency		`
Н	12C2		Baud rate [bps]		
	MRTO		Operation mode	Asynchronous mode	~
	PINT		Parity mode	Parity disabled	¥
	SCT0		Number of stop bits	1	~
Η	SPIO		Data length	8 bits	~
	USARTO	USARTO	Enable peripheral loopback		
	USART1	031110	Enable Rx		
	USART2		Enable Tx		
	WKT			Synchronous mode settings	
	WWDT		Clearly and arity	Filing edge	
			Clock polarity	r anning euge	•
			Enable continuous clock		
			chable hardware flow control		

Figure 27 © 2022 NXP B.V.



1.2.4 Code Generation

Finally select the *Update Code* tab to generate the files. Open the led_blinky folder and all C sources will be generated along with CMAKE scripts and batch file. As indicated, we will not be using the CMAKE scripts and batch files for this exercise however they will be used in the next exercise. One additional file is generated, led_blinky.mex, this is the MCUXpresso Config Tools Setting File. This file can be used to launch the Config Tools and load in that last save configuration for the project.

Image: Image						
	by Delete Rename Oroanize	New item • Easy access • New	Open ~ Select all Select none Select selection Select			
$\leftarrow \rightarrow \checkmark \uparrow$ \checkmark This PC \rightarrow OS (C:) \rightarrow NXP_DEMO	> led_blinky			~	ō	, O Sei
 >	Name board CMSIS component device doc drivers LPC824	Date modified 2022-12-27 10:57 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM	Type File folder File folder File folder File folder File folder File folder		Size	
	source startup utilities armgcc.cmake	2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM	File folder File folder File folder CMAKE File			3 KB
	 build_all.bat build_all.sh build_debug.bat build_debug.sh build_release.bat build_release.sh clean.bat 	2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM	Windows Batch File Shell Script Windows Batch File Shell Script Windows Batch File Shell Script Windows Batch File			1 KB 1 KB 1 KB 1 KB 1 KB 1 KB 1 KB
	 clean.sh CMakeLists.txt flags.cmake led_blinky.mex 	2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 9:09 AM 2022-12-27 10:57 AM	Shell Script Text Document CMAKE File MCUXpresso Config Tools Settings Fi	ile		1 KB 3 KB 4 KB 15 KB



2 Hardware Requirements

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 development. In this section we will describe the hardware required and how to modify the LPCXpresso824MAX board.

- PC running Windows.
- External 5 VDC power supply
- Segger J-LINK debug probe.
- NXP LPCXpresso824MAX Evaluation board.



Figure 28 Exercise 2 block diagram



2.1 LPC824Max modifications

This section lists all modification to the LPCXpresso824MAX board.

 12 MHz oscillator (Y2): This project use Y2 (12 MHz Crystal oscillator) as the reference for the LPC824 system clock. Jumpers SJ3 and SJ4 must be solder bridged between pads 1 and 2 to route crystal to PIN18 (PIO0_8/XLATIN) and PIN17 (PIO0_9/XTALOUT).



Figure 29 © 2022 NXP B.V.

• JP1 On-chip debug probe disable. This project does NOT use the LPC11U35 debug probe. If not provided, solder in a 2-PIN rectangular header (standard .100 pitch). Insert a jumper shunt to disable the on-chip debug probe.



Figure 30 © 2022 NXP B.V.



Power. There are several ways to power the LPCXpresso824MAX board externally. Our exercise uses P3 PIN1 (GND) and PIN2 (VIN). Solder in a 2-PIN rectangular header on P3.1 and P3.2 and connect an external 5.0 VDC source to P3.1 (GND) and P3.2 (5 VDC). Note: D6 reverse polarity protection is provided.



Figure 31 © 2022 NXP B.V.



3 Importing project into Eclipse

Launch Eclipse, create the workspace under the NXP_DEMO folder:

Eclipse IDE Launcher			×
Select a directory as workspace Eclipse IDE uses the workspace directory to store its preferences and development artifact	ts.		
Workspace: ⁹ C.\NXP_DEMO	~	<u>B</u> rowse.	
□ <u>U</u> se this as the default and do not ask again ▶ <u>R</u> ecent Workspaces			
Launch		Cancel	I

Select the *Create a new Embedded C/C++ project* \rightarrow *C Managed Build*. For *Project name* select the **led_blinky** folder as all sources were generated under that folder, *Project type*; **EMPTY Project** and *Toolchains:* **ARM Cross GCC**:

•	—	L X
C Project		-
Directory with specified name already exists.		
Project name: led_blinky		
Use <u>d</u> efault location		
Location: C:\NXP_DEMO\led_blinky		Browse
Choose file system: default \sim		
Project type: Toolc	hains:	
GNU Autotools Executable Empty Project Hello World ANSI C Project Hello World ANSI C Project Hello World ARSC-V C Project ADucM36x C/C++ Project Hello World Arm Cortex-M C/C++ Project Freescale Processor Expert C/C++ Project SiFive RISC-V C/C++ Project STM32F0xx C/C++ Project STM32F0xx C/C++ Project STM32F2xx C/C++ Project STM32F2xx C/C++ Project STM32F3xx C/C++ Project STM32F3xx C/C++ Project STM32F3xx C/C++ Project STM32F7xx C/C++ Project Hello World Arm QEMU xPack C/C++ Project Hello World RISC-V QEMU xPack C/C++ Project Hello World RISC-V QEMU xPack C/C++ Project Wakefile project ✓	Arm Cross GCC Cross GCC MinGW GCC RISC-V Cross GCC	
Show project types and toolchains only if they are suppo	orted on the platform	
?	lext > Einish	Cancel



Ensure *Toolchain name* and *Toolchain path* are correct:

•		_		×			
GNU Arm Cross Select the toolch	Toolchain ain and configure path. AArch64 projects require an AArch64 toolchain!			2			
Toolchain name:	GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc)			~			
Toolchain path:	C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10\bin	Brows	e x	ack			
On macOS use Sł	On macOS use Shift+Cmd+'.' to show the hidden folders while browsing the file system. xpm uses a .content folder to store the binaries.						
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Canc	el			

The project will import all generated files:

NXP_DEMO - Eclipse IDE						-		×
<u>File Edit Source Refactor Navigate Source</u>	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow	/ <u>H</u> elp						
	🖞 - 🕞 - 🕲 - 掾 - 🔘	- 🤬 - 💁 - 🍅	🛷 🔻 🗗 🔲	п 🗉 🏪 🔌	念詞▼福▼	*> <* <> .		
						C		
							`	
Project Explorer X						0 × 🖲 B	″1	
🖻 🖘 🍸 🕴								
✓					The	ere is no active	editor th	nat
> 🗊 Includes					pro	ovides an outlin	e.	
> 🔁 blinky								
> 🔁 board								
> 🗁 device								
> 🔁 doc								
> 🔁 drivers								
> 👝 📖								
> 🔁 LPC824								
> 🔁 lpcxpresso824max								
> 👝 red_blinky								
> 🔁 source								
> 🔁 startup								
> 🔁 utilities								
amgcccmake								
build all.sh								
build_debug.bat								
build_debug.sh								
build_release.bat								
build_release.sh								
💿 clean.bat						_		
lean.sh	👫 Problems 🗙 🖉 Tasks	🖳 Console 🔲 Prope	erties				j õ	
	0 errors, 1 warning, 0 others							
Indes.cmake	Description		Resource	Path	Location	Туре		
	> 💧 Warnings (1 item)							
1								
C /lad bliplay/utilities							M -	20
/ied_bilnky/utilities						· · · · · · · · · · · · · · · · · · ·		/ 🔘





In the project properties ensure Arm family is correct "cortex-m0plus":



3.1 Cross Assembler FLAGS

Properties \rightarrow C/C++ Build \rightarrow Settings \rightarrow GNU Arm Cross Assembler, Other assembler flags:



-DDEBUG -D__STARTUP_CLEAR_BSS -mcpu=cortex-m0plus -mfloat-abi=soft -mthumb



3.2 C Cross Compiler FLAGS

Properties \rightarrow C/C++ Build \rightarrow Settings \rightarrow GNU Arm Cross C Compiler, Other compiler flags:

-DDEBUG -DCPU_LPC824 -DCPU_LPC824M201JHI33 -g -mcpu=cortex-m0plus -Wall -mfloat-abi=soft -mthumb -MMD -MP -mapcs





3.3 Cross Linker FLAGS

Properties → C/C++ Build → Settings → GNU Arm Cross C Linker, Other linker flags:

-g -mcpu=cortex-m0plus -Wall -mfloat-abi=soft -fno-common -ffunction-sections -fdata-sections -ffreestanding -fno-builtin -mthumb -mapcs

-Xlinker [option]: --gc-sections --static -z muldefs

Check Boxes:

Use newlib-nano (--specs=nano.specs) Do not use syscalls (--specs=nosys.specs)

Properties for led_blinky			– 🗆 X
type filter text	Settings		< ⇒ ⇒ < 8
 > Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor C/C++ General Linux Tools Path MCU Project Natures Project Natures Project References Run/Debug Settings Task Tags Validation WikiText 	Configuration: Debug [Active]	s 🔋 Container Settings 🎤 Build Steps 😤 Build Artifact 🗟 Binary Parsers 🤣 Error Parsers	 ✓ Manage Configurations
		Linker flags (-Xlinker (option)) 	ର ର କ୍ରି ନି୍ୟ <u>ହ</u> ା
	> 🛞 GNU Arm Cross Print Size	Other objects	원 원 원 원
		Generate map "\$[BuildAttifactFileBaseName].map" Coss reference (-Xlinker print-map) Print link map (-Xlinker print-map) Use newline nano (-specs-nanospecs) Use float with nano caraf (-u_printf_float) Use float with nano caraf (-u_scanf_float) Ø Do not use syscalls (specs=nosys.specs) Verbreie (-v) Other linker flags -g -mcpu=cortex-m0plus -Wall -mfloat-abi=soft -fno-common -ffunction-sections -f/data-sections -ffreestanding -	fno-builtin -mthumb -mapcs
?		E Contraction of the second seco	pply and Close Cancel



3.4 Include Paths

Properties \rightarrow C/C++ General \rightarrow Paths and Symbols \rightarrow Includes:



Properties for led_blinky			— 🗆 X						
type filter text	Paths and Symbols		← ▼ ⇒ 8						
> Resource									
Builders	Configurations Dahum [Asting]		Marrie Carfornition						
> C/C++ Build	Configuration: Debug [Active]		 Manage Configurations 						
V C/C++ General									
> Code Analysis									
File Types	includes # Symbols 🖺 Libraries	s 📂 Library Paths 🗁 Source Location 🖹 References							
Formatter	Languages	Include directories	Add						
Indexer	Assesses		Addin						
Language Mappings	GNUC	管/led_blinky/CMSIS							
Paths and Symbols	SNOC	Part / Ied_blinky/board							
Preprocessor Include Pat		響/led_blinky/device	Delete						
Profiling Categories		/ied_blinky/drivers	Export						
Linux Tools Path		"#/led_blinky/utilities							
> MCU		//ed_blinky/component/uart	Mar 11						
Project Natures			Wove Up						
Project References			Move Down						
Run/Debug Settings									
Task Tags									
> Validation									
WikiText	"Preprocessor Include Paths Macros etc." property page may define additional entries								
	Chevy huilt in values								
	Show built-in values								
	🖓 Import Settings 🛞 Export Set	tings							
			Portors Defaults						
< >			Restore Deraults Apply						
?			Apply and Close Cancel						



Finally build and debug:





er text /C++ Application /C++ Attach to Application /C++ Container Launcher /C++ Postmortem Debugger /C++ Remote Application	J-Link GDB Server St	ger 🕨 St								
/C++ Application /C++ Attach to Application /C++ Container Launcher /C++ Postmortem Debugger /C++ Remote Application	J-Link GDB Server S	jei 📂 Si	Main X Debugger Startup Re Source Common P SVD Dath							
/C++ Attach to Application /C++ Container Launcher /C++ Postmortem Debugger /C++ Remote Application	J-LINK ODD SERVER S	- Link GDB Server Setun								
/C++ Container Launcher /C++ Postmortem Debugger /C++ Remote Application	Start the J-Link GDB server locally									
/C++ Postmortem Debugger /C++ Remote Application	Executable path: S(ilink path)/S(ilink odbserver)					Browse	Variables			
	Actual eventables [C/Drogram Eiler/EEGEEP/II ink/II inkGDPSenrerCl. even									
	Actual executable.	(to chan	ne it use the glob	or workspace	e preferences page	or the project prope	erties nage)			
DB Hardware Debugging	Device name:	LPC824	M201		e preferences page	sor the <u>project</u> prop	Supported	device name		
D Debugging)ebugging	Endianness:	Little					<u>oupported</u>			
h64 Debugging	Connection:	● USB	⊖ IP			(USB serial or IP nan	ne/address)			
rm Debugging	Interface:	● SWD	⊖ JTAG							
v32 Debugging	Initial speed:	O Auto	◯ Adaptiv	e 💿 Fixed	1000 kHz					
scv64 Debugging	GDB port:	2331								
ink Debugging bug	SWO port:	2332				Verify downloads	Initialize regi	sters on star		
)	Telnet port:	2333				Local host only	Silent			
	Log file							Province		
	Eug nie.							browse		
	Uther options: -singlerun -strict -timeout 0 -nogui									
	Allocate console for the GDB server									
	GDB Client Setup									
	Executable name: \${cross_prefix}gdb\${cross_suffix}				Browse	Variables				
	Actual executable: arm-none-eabi-gdb									
	Other options:									
	Commands:	set men	n inaccessible-by	default off				1		
								~		
	Remote Target									
	Host name or IP ad	dress: lo	ocalhost							
	Port number:	2	331							
	Force thread list up	pdate on s	suspend							
								Restore defa		
>							Devent	Annta		



Using a scope, probe PIO0_0 P3.10 of the LPCXpresso 284MAX board. If clocks were configured correctly a 240 kHz square wave should be seen at this pin:



Additional verification, the RED LED should toggle every 1000 ms:

The solution is available upon request, reference AN122222-001.

support@wojotech.com

