

Teaching online electronics, microcontrollers and programming in Higher Education

Programing of embedded systems

1. Wprowadzenie

Lead Partner: Warsaw University of Technology

Authors: Daniel Krol

University of Applied Sciences in Tarnow



1. Wprowadzenie

Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

© Copyright 2021 - 2023 the ENGINE Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1. Wprowadzenie

I. Układ LPC804

 LPC804 należy do niedrogiej, 32-bitowej rodziny mikrokontrolerów opartych na rdzeniu Arm Cortex-M0+ i działających z częstotliwością taktowania do 15 MHz. LPC804 posiada 32 kB pamięci flash i 4 kB SRAM. Urządzenia peryferyjne LPC804 to: silnik CRC, dwa interfejsy magistrali I2C, dwa układy USART, jeden interfejs SPI, pojemnościowy interfejs dotykowy (*Cap Touch*), jeden timer typu *multi-rate*, jeden timer typu *self-wake-up*, jeden uniwersalny 32-bitowy licznik/timer, jeden 12-bitowy przetwornik A/C, jeden 10-bitowy przetwornik C/A, jeden komparator analogowy, porty I/O z funkcją konfiguracji za pomocą macierzy przełączników, silnik dopasowywania wzorców wejściowych, programowalna jednostka logiczna (PLU) oraz do 30 wyprowadzeń we/wy ogólnego przeznaczenia [1].



Source: NXP

2. Płytka prototypowa wraz z modułami rozszerzeń:





Source: NXP

1. Wprowadzenie

II. Zintegrowane środowisko programistyczne MCUX presso IDE

Przejdź do strony: <u>https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE</u>, dokonaj darmowej rejestracji konta. Instalator jest dostępny dla systemów operacyjnych Windows, MacOS oraz Linux:

Product Download

М	CUX	(presso IDE						
F	iles	License Keys	Notes					O Down
Sho	w All	Files						
+	File (Description		\$	File Size	\$	File Name	
+	MCU	XpressoIDE 11.4.1 -	Linux		928.6	MB	L mcuxpressoide-11.4.1_6260.x86_64.deb.bin	
+	MCU	XpressoIDE 11.4.1 -	Mac		885.1	MB	MCUXpressolDE_11.4.1_6260.pkg	
+	MCU	XpressoIDE 11.4.1 -	Windows	,	833.3	MB	MCUXpressoIDE_11.4.1_6260.exe	

2. Zainstaluj MCUXpresso na swoim komputerze:



3. Następnie przejdź na stronę: https://mcuxpresso.nxp.com/en/welcome

1. Wprowadzenie



4. Naciśnij *Select Development Board*, wybierz płytkę prototypowa z układem LPC804 a następnie naciśnij *Build MCUXPresso SDK* (aktualna wersja):

NXP MCUXpresso	SDK Builder		0 🗭 E 🔺 🕹
SDK Dashboard BUILD SDK	Select Development Board Search for your board or kit to get started.	Select	on Details
Middleware (0) Examples (0) Toolchain (Off)	Search for Hardware		LPCXpresso804
NNC MCUXpressols Stock Dashboard Back Dashboard Back Dashboard If Modeware (b) If Modeware (b) If Modware (b) If Modwar	ः Select a Board, Kit, or Processor		LPCXpresso Development Board for LPC804
Notifications	LPCXpresso55S06 (LPC55S06)	Bull	a MCUApresso SDK V2.10.0 * Additional Details
Preferences	LPCXpresso55S16 (LPC55S16)	Match	ed Hardware Platforms
DOWNLOADS	LPCXpresso55S28 (LPC55S28)	Found	HW solutions that match your criteria.
MCUXpresso IDE	LPCXpresso55S69 (LPC55S69)	(Board	s: 121), Kits: 77), Processors: (395))
MCUXpresso	LPCXpresso802 (LPC802)		
Config Tools	LPCXpresso804 (LPC804)	Filterin	g Criteria - Reset all
Offline data	LPCXpresso812MAX (LPC812)	R	equired Middleware
MCUXpresso Secure Provisioning Tool	LPCXpresso824MAX (LPC824)		Middleware filtering not applied
court from any four	LPCXpresso845MAX (LPC845)	В	equired Example Projects
	► MW		Example Project filtering not applied
	→ QN	p	equired Toolchains
	→ dsc	<u></u>	Toolchains filtering not applied
	▶ I.MX		
	 Kits 	P	rocessor Parametric Filtering Processor Parametric Filtering not applied
BULD SOK Sector	 Processors 		g
	 Deprecated 		

Programing of embedded systems 1. Wprowadzenie

- 5. Wybierz system operacyjny i zaznacz *Toolchain / IDE: MCUXpresso*. Następnie wybierz *Select All* available components:
- 6. Po chwili gotowe *SDK* pojawi się w *Dashboard*. Kliknij *Download SDK* i w następnym oknie wybierz Download SDK Archive:



7. Otwórz *MCUXpresso IDE* and przeciągnij (Drag and Drop) archiwum SDK do zakładki *Installed SDKs*:

•••	o workspace_LPC804 - LPC804_Project/source/LPC804_Project.c - MCUXpresso IDE			
🖆 • 🖩 🕲 • 🗞 • 📓 전 🏷 🔛 🕷 🖛 🗰 14 3. (5. 2) 등	🕱 🗅 🗇 🐘 🔍 🗞 🕹 🔕 🌢 🔗 👗 🕸 + 🍢 🏘 🍁 🕗 🖓 - 🖗 🖉 🖉 - 🖗 👘 🖉 - 🚱 🖉		् 🖻 🔀	🖷 🗤 🕴 🔿 I
🍐 Project Explorer 😫 🚟 Registers 💠 Faults 🐕 Peripherals+ 👘 👘	LPC804_Project.c 23	- 0	GE Outline 22 (*)= Global Variables	8 🗆
Orect Captor 2: Capto	<pre>X</pre>		C III Could Variable There is no active editor that provides an outline.	
Project: LPC804_Project [Debug] v Create or import a project	<pre>68 tight while() loop */ 69asm volatile ("ngg"); 70 } </pre>			
Second Seco	1 Installed SDKs	inory SS (I)- Hea	p and Stack Usage 전 문 같 때 전 문 전 문 전 문 전 문 전 문 전 문 전 문 전 문 전 문 전	j 15 - 3 **

Programing of embedded systems 1. Wprowadzenie

8. Przejdź do Import SDK examples (obrazek powyżej) i wybierz lpcxpresso804:



9. Wybierz demo_apps i zaznacz przykład led_blinky. Kliknij Finish:

• • •	SDK Import Wizard					SDK Import Wizard			
🔏 The source from the SDK will be copied into the workspace. If you want to use linked	files, please unzip the 'SDK_2.x_LPCXpresso804' SDK.	NP 🖆						N	Pe
Import projects			Advanced Setti	ngs					
Project name prefix: lpcxpresso804	Project name suffix:		* C/C++ Library Settings						
Use default location			Set library type (and host	ing variant) Dectils (solute)	ef)				
Location: JJsers/danie//Documents/MCUXpressolDE_11.4.1_6260/workspace_LPCB		Browse	~~~~						
Project Type	Project Options		Redit: Use hoating po	nt version of print! ather then string based print!			eewtionano: Use hoating p lead bhano: Use floating p	ioint version of print/	
C Project C++ Project C Static Library C++ Static Library	SDK Debug Console 🕓 Semihost 🥥 UART 🔿 Exam	pie default							
	Copy sources		Redrect SDK "PRINTP"	to C library -print-			redirect printfyscare to I M	1	
Evanoies									
Chana M. Diar									
Name	Description	Version							
Bethyland Bethyland	The indefined dama prove the "Well broks" along the the second of the The indefined dama proves the second	ng ng StOL (MT Gha dia mananana make contu GPu	Language standard (2014) MCULINARE Link application to MAA Memory distalls Default LinkServer Flash Tipus Flash Roka R	U C99 (-toti-gnu99) Driver Name PROGRAM, FLASH BOOT, FLASH SAAM WP_SRAM WP_SRAM	Alas Fissh Fissh2 RAM RAM2	Location 0x0 0x7000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x100000 0x10000 0x10000 0x10000 0x10000 0x10000 0x100000 0x100000 0x10000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x1000000 0x100000 0x100000 0x100000 0x100000 0x100000 0x1000000 0x1000000 0x1000000 0x100000000	Size Ov7Ho Ox80 Ox40 Ox40 Ox20 Defete	Diver LPCIO _N _32.ctx	Bresse
0	< Back Next >	Cancel Finish	0				< Back	Next > Cancel	Finish

1. Wprowadzenie

10. Zbuduj projekt przez naciśnięcie *Build*:



11. Podłącz płytkę *LPCXpresso804* do komputera za pomocą interfejsu USB, oznaczonego jako *Emulator*:



Programing of embedded systems 1. Wprowadzenie

12. Zaprogramuj mikrokontroler naciskając GUI Flash Tool:



13. Pozostaw domyślne ustawienia w poszczególnych oknach programatora i naciśnij *Run*:

• • •	Probes discovered		• • •	GUI Flash Tool	
Connect to target: LPC804 1 probe found. Select the probe to use			GUI Flash Tool for: MCUXpresso IDE Li Program file into fl	inkServer (inc. CMSIS-DAP) probes lash: lpcxpresso804 led blinky.axf	
Available attached probes			alter.		
Name	Serial number / ID / Nickna Type	Manufacturer	Target: LPC804		
LS LPC11U3x CMSIS-DAP v1.0.4	02014020 LinkServer	NXP Semiconductors	Probe Options		
			Probe specific options		
			Connect script		Workspace File System
			Default Flash Driver		Workspace File System
			Reset Handling	Default	0
			Flash Reset Handling	Default	0
MCUXpresso IDE LinkServer (inc. CMS	Gisable) SIS-DAP) probes		Boot ROM Stall		
P&E Micro probes SEGGER J-Link probes			Wire Speed		
			Reset the target on connection	Disable use of preconnect script	
Probe search options			Target Operations		
Search again			Select the target flash operation to p	erform	
			Program Erase		
			Actions		
Ø	Cancel	OK	Select the action to perform		
			• Program	Program (mass erase first)	
			U venty only	Check file areas blank	
			Options		
			Select the options to apply		
			File to program	pace_loc)/lpcxpresso804_led_blinky/Debug/lpcxpresso804_led_blinky.ax	Workspace File System
			Format to use for programming	o axf ○ bin	
			Base address	0×0	
			Reset target on completion		
			Coneral Ontions		
			Flash programming tool options		
			Additional options		
			Repeat on completion Z Enable	e flash hashing Preview command	
			Clear console	-	
					Cancel Run

14. Diod LED na płytce prototypowej powinna zacząć błyskać.

- 1. Wprowadzenie
- 15. Wybierz tryb *Debug*:

	workspace_LPC804 - Ipcxpresso804_led_blinky/source/led_blinky.c - MCUXpresso IDE		
1 • 🖩 🐚 • • • • • • • • • • • • • • • • • •	19 19 19 20 19 20 20 20 20 20 20 20 20 20 20 20 20 20		ର୍ 🔡 🔀 🖷 🗤 🕴 🗘 🞚
project Explorer 🕱 🕮 Registers 🎋 Faults 🧏 Peripherals+ 👘 🗖	👔 led_blinky.c 🐹 👘 🗇	📴 Outline 🐹 🗱 Global Variables	🖂 1ª2 💐 🖋 🕈 🕴 🗖 🗆
Image: Section 1 Image: Section 1<	<pre>iv/ view iv/ iv/ iv/ iv/ iv/ iv/ iv/ iv/ iv/ iv/</pre>	Compared in the second se	
 Build your project 	🍘 Installed SDKs 🔲 Properties 🖹 Problems 📮 Console 🕱 🎆 Image Info 🙀 Debugger Console 🧏 Offline Peripherals 👘 🗖 🚺 Memory 🕱 🚥 Heap	p and Stack Usage	📑 🛃 🚚 🐯 🖏 i 😑 i
Build	COT Build Consele Resurgeree RAd lad bliebel	K %	
Clean	BODI_FLASH: 0 GB 128 B 0.00%		
- Debug your project	52-6 4004 10-10-10-10-10-10-10-10-10-10-10-10-10-1		
😿 🎋 Debug 🌾 Terminate, Build and Debug	rinisme ouilaing target: (ptxpressoow_led_blinky.axt makeno-print-directory post-build Performin post-build stee		
 Miscellaneous 	arm-none-eabi-size "lpcxpresso804_led_blinky.axf"; # arm-none-eabi-objcopy -v -0 binary "lpcxpresso804_led_bl taxt data bs der bay filename		
Edit project settings	6488 8 724 7220 1c34 lpcxpresso804_led_blinky.axf		
MCUXpresso Config Tools>>			
M MCUXpresso Config Tools>> ঔ Quick Settings>> ♥ Export project(s) to archive (zip)	19:25:20 Build Finished. 0 errors, 0 warnings. (took 670ms)		

16. Przez naciskanie Step Over (F6) mazes wykonywać program krokowo. Naciśniecie Instruction Stepping Mode otwiera okno Disassemblera, w którym możesz zobaczyć instrukcje assemblera wygenerowane przez kompilator. Przejdź do zakładki Peripherals i znajdź GPIO a następne rejestr B0_13:

• • •				e workspace_LPC804 - Ipcxpresso804_Ied_blinky/source/led_blinky.c - MCUXpresso IDE	
📬 • 🔝 🐚 🛞 • 🗞 • 📷 🔤	0 9 🖸 🗴 🕨 🗉	B 14 B	👁 .e. 🔍 🗷	19 11 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	< 2 4 10 ● 10 4 10 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
🍐 Project Explorer 🛛 👫 Registe	rs 🐐 Faults 🛼 Peripher	rals+ 🔀		1/2 Debug 22	🚼 Outline 🖾 🕪 Global Variables 🛛 🖹 🖧 💘 🔍 🗮 🖇 🗂
Name	Value	Acces	E C S	Compresse804_led_blinky LinkServer Debug (C)C++ (NXP Semiconductors) MCU Application] Compresse804_led_blinky.axt [LPC804 (contex-m0plus)]	1 pin_mux.h 1 board.h
> 🛃 ACOMP			0x40024000	 ^w Inread #11 (Suspensed : Breakpoint) ^w main() = text biology (State) ^w main() = text biology (State) ^w main() = text biology (State)	# BOARD_LED_PORT
> 🔀 ADC			0x4001c000	am-none-shi-odh (10.190,20201028)	V g systickCounter : volatile uint32 t
> 🛃 CAPT			0x40060000		SysTick_Handler(void) : void
> 🔀 CRC			0x5000000		 SysTick_DelayTicks(uint32_t) : void
> CTIMERO			0x40038000		main(void) : int
			0x40014000	ici led_blinky.c 🕄	
> 111180 0	0x01	PW/	0xa0000000	32 {	
> 1111 B0 1	0x01	RW	0xa0000001	34 }	
> 100 B0 2	0x00	RW	0xa0000002	35 }	
> ##80_3	0x01	RW	0xa0000003	36 370 unid funTick DelauTicke(uist33 t. a)	
> #*** 80_4	0x01	RW	0xa0000004	3/ a volu systick_betayticks(din(sz_t n/ 38 {	
> ###B0_5	0x01	RW	0xa0000005	39 g_systickCounter = n;	
> ###B0_6	0x00	RW	0xa0000006	40 while (g_systickCounter != 0U)	
> 000 80_7	0x01	RW	0xa0000007	41 1 42 }	Disassembly 22
> 600 80_8	0x01	RW	0xa0000008	43 }	Enter location here 🗧 👔 🏠 🕵 📑 🖻
> 11180_9	0x01	RW	Uxa0000009	44	00000518: movs r2, #250 : 0xfa
> IIIB0_10	0x01	RW	0xa000000a	42°€/#Σ 46 ★ dbrief Main function	0000051a: lsls r1, r2, #2
) 100 00 10	0x01	RW	0xa000000B	47 */	0000051c: movs r0, r3
2 1000_12	0x01	DIM	0xa0000000	480 int main(void)	88888522: movs r3, r8
PRYTE	0x00	RW	[0]	49 (50 /* Thit output LED GDTO */	00800524: movs r0, r3
> 1000 B0_14	0x01	RW	0xa000000e	51 GPID POTLINIT(GPID, BOARD LED PORT);	00000526: bl 0x46c <systick_config></systick_config>
> 80_15	0x01	RW	0xa000000f	52 /* Board pin init */	00000524: SUDS F3, F0, #0 0000052c: beg.n 0x530 <main+52></main+52>
> 100 B0_16	0x01	RW	0xa0000010	53 BOARD_INITROOTPINS(); 54 BOARD_CITEROST(); 55 BOARD_CITEROST();	59 while (1)
> ###B0_17	0x01	RW	0xa0000011	55	0000052e: b.n 0x52e <main+50></main+50>
> ###B0_18	0x01	RW	0xa0000012	56 /* Set systick reload value to generate ims interrupt */	67 Systick_Delay(icks(10000); 00000530: movs r3, #250 : 0xfa
> B0_19	0x01	RW	0xa0000013	57 if (SysTick_Config(SystemCoreClock / 1000U))	00000532: lsls r3, r3, #2
> ###B0_20	0x01	RW	0xa0000014	59 while (1)	00000534: movs r0, r3
Quickstart Panel (*)- Var	riables 💁 Breakpoints		- 0	60 { 61 }	68 GPIO_PORTToggle(GPIO, BOARD_LED_PORT, 1u << 8
MCUXpresso ID Project: Ipcxpresso804	E - Quickstart Pane _led_blinky [Debug]	el	_	62) 63 while (1) 65 (♥ 00000534: m0% (3, #120 ; 0.00 00000532: lsls - 72, r3, #6 00000532e: movs r3, #160 ; 0xa0 00000540: lsls r3, r3, #24 00000540: novs r1, #0
 Create or import a project 				00 /* Detay 1000 Hg */ 1000U); 67 SysTick DetayTicks(1000U);	00000544: movs r0, r3 00000546: bl 0x14bc <gpio porttoggle=""></gpio>
New project				69 69 GP10_POTTIOGGLe(GP10, BUARD_LED_PUKT, 1u << BUARD_LED_PIN); 69)	67 SysTick_DelayTicks(1008U); 0000054a; b.p. 0x520 cmaju(52)
Import SDK example	le(s)			70 }	0000054c: movs r4, r0
- Build your project(s) in	on ne system.				0000054e: asrs r0, r0, #32
Build your project				1 Installed SUKs II Properties I Problems Console X III Installed SUKs III Properties III Memory X IVI Her	ip and Stack Usage 12 to
Clean				Ipcxpresso804_led_blinky LinkServer Debug (c)(C++ (NXP Semiconductors) MCU Application) Millionesse Semibodition Tellect console for 'Incrvnesse884 led blinky LinkServer Debug' started on port 58874	<u> </u>
 Debug your project 		(🖸 • 🔛 • 🔜 •	presignesso selected and president of the peripherson comparison of search of the sear	
Debug	d Debug				
 Miscellaneous 					
Edit project settings					
MCUXpresso Config Tools>	•				
Quick Settings>>					
Export project(s) to archive					

1. Wprowadzenie

- 17. Sprawdź, jak zmienia się wartość rejestru *B0_13*, gdy dioda jest zapalona oraz zgaszona.
- 18. Wartość rejestru *B0_13* można zmienić bezpośrednio w zakładce *Peripherals* wpisując 0 lub 1.

III. Zadania

1. Sprawdź działanie innych funkcji *Debugera* w menu *Run*:

Run	RTOS	Analysis	Window	Help
∎► Re	esume			F8
Su	Ispend			
🔳 Te	rminate		H	GF2
💦 Di	sconnect	t		
🜏 St	ep Into			F5
🔁 St	ep Over			F6
_n? St	ep Returi	n		F7
⇒] Rι	un to Line	÷	H	8 R
🔊 U:	se Step F	ilters		
Step I	nto Selec	tion	H	S F5

2. Sprawdź inne przykłady dostarczone z pakietem SDK

Dokumentacja

- 1. LPC804 Data Sheet, https://www.nxp.com/
- 2. LPC804 User manual, https://www.nxp.com/
- 3. User Manual for LPCXpresso804 Board, https://www.nxp.com/