

Teaching online electronics, microcontrollers and programming in Higher Education

#### **Programing of embedded systems**

3. Timers and counters

Lead Partner: Warsaw University of Technology

**Authors: Daniel Krol** 

University of Applied Sciences in Tarnow



3. Timers and counters

### 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 Where other published and unpublished source materials have been used, these have been acknowledged.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.

3. Timers and counters

#### I. System Timer

- 1. Create a new project for the *LPCXpresso804* board as in the previous manual and name the project eg *Lab02*.
- 2. Configure three GPIO lines to control the RGB LEDs. From the Functional Group menu, select the *BOARD\_InitLEDsPins* preset, then activate it by selecting the flag icon on the left, as in the previous manual. Select *Update Code* to generate the code based on the entered configuration.
- 3. Modify the program code by adding system timer support:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"
bool g_pinState = false;
void SysTick_Handler(void)
   GPI0_PinWrite(BOARD_INITLEDSPINS_LED_RED_GPI0, BOARD_INITLEDSPINS_LED_RED_PORT, BOARD_INITLEDSPINS_LED_RED_PIN,
g_pinState^=true);
}
 * @brief Application entry point.
int main(void) {
              * Init board hardware. */
            BOARD_InitBootPins();
BOARD_InitBootClocks();
            BOARD InitBootPeripherals()
BOARD_InitDebugConsole();
#endif
            SysTick_Config(SystemCoreClock / 10U); // 10 Hz
            while(1) {
            }
            return 0 :
```

Build a project, program the microcontroller and check the operation. The led should change state 10 times per second (5 flashes per second).

#### II. Delay function

- 1. Create a new project for the *LPCXpresso804* board and name it eg *Lab02\_2*.
- 2. As before, configure three GPIO lines to control the RGB LEDs. Modify the program code as in the example below:

3. Timers and counters

*/ int main(	void) {
	<pre>/* Init board hardware. */ BOARD_InitBootPins(); BOARD_InitBootClocks(); BOARD_InitBootPeripherals();</pre>
#ifndef B	DARD_INIT_DEBUG_CONSOLE_PERIPHERAL /* Init FSL debug console. */ BOARD InitDebugConsole();
#endif	
	<pre>SysTick_Config(SystemCoreClock / 1000U); // 1 ms</pre>
	while(1) {
	GPI0_PinWrite(BOARD_INITLEDSPINS_LED_RED_GPI0, BOARD_INITLEDSPINS_LED_RED_PORT, BOARD_INITLEDSPINS_LED_RED_PIN, g_pinState ^= true);
	<pre>delay_ms(500); }</pre>
}	return 0 ;

Build a project, program the microcontroller and check the operation. The led should change state 2 times per second (1 flash every second).

3. Rebuild the project in the *Release* configuration by changing the settings in the drop-down menu next to the Build icon:



Build a project, program the microcontroller and check the operation. Due to compiler optimization, the *g\_systickCounter* variable is not "refreshed" in the while loop inside the *delay\_ms* function. Hence, the LED will stop flashing.

4. To force the value of the *g\_systickCounter* variable to "refresh" each time, add the *volatile* modifier:



3. Timers and counters

	while(1)		
		GPIO_PinWrite(BOARD_INITLEDSPINS_LED_RED_GP BOARD_INITLEDSPINS_LED_RED_PO BOARD_INITLEDSPINS_LED_RED_PI g_pinState ^= true);	PIO, DRT, N,
	}	delay_ms(500);	
}	return 0		

Build a project, program the microcontroller and check the operation. The led should change state 2 times per second (1 flash every second) as it did in *Debug* mode.

#### III. CTIMER - Match mode

1. Create a new project for the *LPCXpresso804* board and name it eg *Lab02\_3*. Add the *ctimer* driver:

• • •	S	DK Wizard		
A The source from the SDK will be control of	opied into the workspace. If you want to use linked files, pleas	e unzip the 'SDK_2.x_LPCXpresso804' SDK.	NF	G
Configure the project				
Project name: Lab02_3		Project name suffix:		
Use default location				
Location: /Users/daniel/Documents/	/MCUXpressoIDE_11.4.1_6260/workspace_LPC804/Lab02_3			Browse
Device Packages	Board	Project Type	Project Options	
<ul> <li>LPC804M101JDH24</li> <li>LPC804M101JDH20</li> <li>LPC804M111JDH24</li> </ul>	<ul> <li>Default board files</li> <li>Empty board files</li> </ul>	C Project C++ Project     C Static Library C++ Static Library	SDK Debug Console Semihost CMSIS-Core Copy sources Import other files	<ul> <li>UART</li> </ul>
Components		•	Components selection summary	
Add or remove SDK software compo	prents	U.	componente cenection caninary	
Operating Systems Drivers CMS	SIS Drivers Utilities Middleware Board Components Abstra	action Laver Software Components	type to filter	
		_ ~	Name Description	Versio Info
Drivers			Operating Systems	
type to filter			> Utilities	
Name	Description	Version Info	_	
🗌 🐁 capt	CAPT Driver	2.1.0		
🗹 🚯 clock	Clock Driver	2.3.1		
🗹 🚯 common	COMMON Driver	2.3.0		
ctimer	CTimer Driver	2.2.1		
🗌 🙀 dac	DAC Driver	2.0.2		
🗹 🚯 gpio	GPIO Driver	2.1.7		
🗌 🎧 i2c	I2C Driver	2.1.0		
🗌 🚯 iap	IAP Driver	2.0.4		
🗹 🚯 iocon	IOCON Driver	2.0.1		
Ipc_acomp	LPC_ACOMP Driver	2.1.0		
🗌 🚯 lpc_crc	CRC Driver	2.1.1		
🗆 🎧 mrt	MRT Driver	2.0.3		
🗌 🚯 pint	PINT Driver	2.1.8		
🔲 🙀 plu	PLU Driver	2.2.1		
V power_no_lib	Power Driver	2.0.0		
reset	Reset Driver	2.0.1		
?		< Back	Next > Cancel	Finish

#### 3. Timers and counters

2. To do this, right-click on the project name and select *MCUXpresso Config Tools -> Open Pheriperals*:

000			i wo	kspace_LPC804 - Lab02_3	source/Lab02_3.c - M	CUXpresso IDE		
📑 • 🔛 🐚 📸 🛛 Lab02	L3 V	🜴 🛕 📴 Update Code 🔹 Functional Group 🛛 🛙	OARD_InitPeripherals	T 🖻 💛 🕒 🖬	> 10 16 2. 70 .0. 6	• 💁 🙋 🖉 • 💷 👌 • 🕾	🏷 😅 😓 + 🔿 - 📷	<ul> <li>A 18 × 0 170</li> </ul>
🎕 Components 23 🦞 Per	ipherals	🗆 📄 🌴 Overview 23 🔓 Code Preview						
	0	↑↓ > Configuration - General Info						
		Configuration - HW Info						
Peripheral driver	s (Device specific)	Processor: LPC804						
		Part number: LPC804M101JDH24						
Custom in	itialization	Core: Cortex-M0P						
		Board: LPCXpresso804						
		SDK Version: ksdk2_0						
		> Project						
		<ul> <li>Peripherals</li> </ul>						
		Configures the initi	aliza 😑 💿 🔵	Select configu	ration component			
		( th )	Colorst which component	to should be offered. Everysted I	with a PDF and and			
			arrest mitch componen	Supported I	iy the sork package [+]			
			type filter text					
		M. Committed and	Configuration componer	t Component description		Category		
		Ciledate and analisi	d CRC	CRC engine (CRC)		Peripheral drivers (Device s		
		<ul> <li>Opdate code enabled</li> </ul>	A DAC	Standard counter/timer Digital to Analog Converter (DAC	1	Peripheral drivers (Device s Peripheral drivers (Device s		
		board/peripherals.c	GPIO	General Purpose I/O (GPIO)	,	Peripheral drivers (Device s		
		board/peripherals.h	8 12C	Inter-Integrated Circuit (I2C)		Peripheral drivers (Device s		
		<ul> <li>Functional groups</li> </ul>	A PINT	Multi-Rate Timer (MRT) Pin interrupt and pattern match	PINTI	Peripheral drivers (Device s Peripheral drivers (Device s		
			A PLU	Programmable Logic Unit	/	Peripheral drivers (Device s		
		BookeD_interentinenas	💩 SPI	Serial Peripheral Interface (SPI)		Peripheral drivers (Device s		
		V Other tools	USART	Universal Synchronous/Asynchro	nous Receiver/Transmitter (	USART) Peripheral drivers (Device s		
			Open SDK compone	nts manager C	ancel	OK		
		A Problems 82						В 7 -
		type filter text						
		Level ~ Resource	Issue Desigheral UP	Origin Origin	D InitOERIUC LIADTRing	Target Parioherals: BOARD JoitDerioherals	Type Validation	
		Warning 12C0	Peripheral I20	0 is not initialized Pins:BOAF	D_Initi2CPins	Peripherals: BOARD_InitPeripherals	Validation	
Lab02_3								U NXP LPC804* (Lab02_3)
~								

3. Configure the *CTIMER0* pheriperal:

• • •			¢ W	orkspace_LPC804 - Lab0	2_3/source/Lab02_3.c - N	/ICUXpresso IDE					
📬 🖬 🐘 🔛 Lab02_3	• 🕋 🔺	Update Code • Functional Group	IOARD_InitPeripherals	। 🗠 🏷 🦷 🦷	■[> 0 % × * *	s• 💁 🧶 🛷 I 🖉 🗄 - 51			2	् 🔡 🗡	(# m ? 🔿 🖩
Components 🖾 🦞 Peripherals								A Overview 23	C Code Preview		
		Standard counter/timer (Pe	ipheral drivers (Device spec	cific)]			· · · · · · · · · · · · · · · · · · ·	> Configura	ition - General Info		
Perinheral drivers (Device specific)		Name CTIMERO					Custom name	<ul> <li>Configura</li> </ul>	ition - HW Info		
		Mode Input Capture/Match			Peripheral C1	IMERO	•	Processor Part number	: LPC804		
CTIMER0		V <sup>4</sup> Timer/counter general configurati	0				Preset Ourtom	Core	Cortex-MOP		
Custom initialization		hd. Times equates and investion						Board	: LPCXpresso804		
		· miler counter comgutation						SDK Version	ksdk2_0		
		Timer mode	Timer (bus clock source)			00100 001		> Project			
		Bus clock source	System clock - BUARD_B	COLOCKERO 18M: 9 MH2, BOAF	RD_BootClockER024M: 12 MHz	BOARD_BOOTCIOCKERO30M: 15 MHz	•				
		Times input for meaning action	9 MHZ (BUARD_BOOTCIOC	KFRU18M)				<ul> <li>Periphera</li> </ul>	ls		
		Calculated prescaler	1						conigcres the initialization	or the SDK periphera	a crivers.
		Calculated timer input frequency	9 MHz; 111.111 ns					(Ψ	)		
		Timer counter period	500 ms								
		Start timer in initialization code	1								
		Match channels +	X					✓ Generate	d code		
								Update co	de enabled		
		Match_0						board/p	eripherals.c		
		Channel ID	Match_0					board/p	eripherals.h		
		Match channel	Match channel 0				•	V Functiona	l groups		
		Channel frequency/period/offset	4500000					BOARD.	InitPeripherals		
		Channel period (ticks)	4500000 dfeet 2 Hz 500 mr								
		Enable counter reset on match						<ul> <li>Other too</li> </ul>	ls		
		Enable counter stop on match									
		Output control	No action is taken				•		) (ПП)		
		Initial output value	Low				<b>T</b>				
		Enable match interrupt request									
		<ul> <li>Interrupt and callback settings</li> </ul>									
		Configuration enabled (a ma	tch/capture/pwm channel in	terrupt is enabled)							
		✓ Interrupt									
		Interrupt C	TIMER0_IRQn				۲	A Basklama 22			
		Enable priority initialization						Problems 23			0 (
		Priority 0						type filter text	Becourse	leque	
		Callback mode Singl	callback				٣	S Warning	CTIMER0.ctimer_match.0	Match outp	ut of the match 0 cl
		Shared callback function cbTin	er					S Warning	USART0	Peripheral U	USARTO is not initial
								to warning	1200	Peripheral I	1200 is not inicialized
	O showed of earlies	CTREES Is not set of Match and a state		have other than a stand has					ž da		

#### 3. Timers and counters

4. Press *Update Code* and modify the program code by adding the *cbTimer* function, the definition of which was generated in the file *peripherals.h*:



Build a project, program the microcontroller, start the debugger console and check the program operation.

5. Go back to *MCUXpresso Config Tools-> Pheriperals* and activate the hardware output of the *Match* block:



3. Timers and counters

- 6. In the picture showing the microcontroller, click on CTIMER.
- 7. In the open dialog boxes select MATCH, 0 and then PIO0\_11, respectively:

000								0 workspace_LP	C804 - Labuz_3/sc	urce/Labuz_3.c - M	COApresso IDE			
1 📑 • 🗟 🦉	Lab02_3		• 🔺 🔺 🛛	Update Code	Functional Gro	BOARD	InitPins	• 👼	••••••	0.020.00	• 💁 🧶 🛷 刘 🖄	- (0 + <b>0</b> + <b>0</b> + (0 + (0 + (0 + (0 + (0 + (0 + (0 +	Q 18	8 🗶 🖷 🗤 🕴 🖓 🔠
🗄 Pins 😫	Peripheral Signa	als					- 0	😳 Package 🕄				ଷ୍ ପ୍ (ବ ଼ି 🖃 🖻 🖻	A Overview 💱 🛃 Code Preview 🔢 Registers	- 0
990	₩₩ -0 0+ 0	🗲 🤹 😥 🔍 typ											> Configuration - General Info	
Pin	Pin name	Label	Identifier	Arduino UNO	R3 Cor SPI	GPIO	USART						Configuration - HW Info	
1	PIO0_18	CN6[1]/CN3[9]/PI		CN3[9] (D9)	PIO0_18[.	] PIO0_18	PIO0_18[.							
2	PIO0_16/ACMP_L	CN6[2]/CN5[2]/U		CN5[2] (A1)	PIO0_16[.	_] PIO0_16	PIO0_16[.		F				Processor: LPC804	
3	PIO0_17/ADC_9	CN6[3]/CN5[3]/PI	LED RED-S1	CN5[3] (A2) CN8[4] (D4)	PICO_17[.	1 800 13	PIO0_17[.		0		v + 6 8		Part number: LPC804M101JDH24	
5	PI00_12	\$2/CN8[6]/CN6[5	LED_GREEN:S2	CN8[6] (D2)	PIO0_12[	1 PIO0_12	PI00_12[.		ě.				Core: Cortex-M0P	
6	RESETN/PIO0_5	CN6[6]/CN1[10]/	DEBUG_SWD_RE	CN4[3] (RST)	PIO0_5[	] PIO0_5	PIO0_5[		2 2 2	- C - (	//67 //67	3	Board: LPCXpresso804	
7	PIO0_4/ADC_11	CN6[7]/CN8[7]/C	DEBUG_UART_TX	CN8[7] (D1);	CN5 PIO0_4[	] PIO0_4	PIO0_4[		01 01 0	USS NO		ž	SDK Version: ksdk2_0	
8	SWCLK/PI00_3	CN6[8]/CN1[4]/U	DEBUG_SWD_SW		PI00_3[	] PIO0_3	P100_3[						> Project	
9	SWDIO/PI00_2	CN6[9]/CN1[2]/U	DEBUG_SWD_SW	010/01/02	PIO0_2[	] PIO0_2	PIO0_2[						2 Tropage	
11	PIO0_10/ADC_7	CN6[10]/CN8[3]/	LED_BLUE	CN6[3] (05)	PIG0_11[.	1 PIO0_11	PIO0_11[. PIO0_10[.						V Pins	
12	PIO0_21/ACMP_I5	CN6[12]/CN8[5]/		CN8[5] (D3)	PIO0_21[.	_] PIO0_21	PIO0_21[.						Configures pin routing, including function	inal electrical pin
13	PIO0_20	CN7[12]/CN3[10]		CN3[10] (D8)	PICO_20[.	_] PIO0_20	PIO0_20[.						properties, voltage/power rails, and run-	time pin configuration.
14	PIO0_15/ADC_8	CN7[11]/CN3[8]/		CN3[8] (D10)	PIO0_15[.	] PIO0_15	PIO0_15[.		ACMP AD	CO CAPT	CTIMERO DACO CPIC			
15	PIO0_1/ADC_0/A	CN7[10]/CN3[5]/		CN3[5] (D13)	; CN PIO0_1[	] PIO0_1	PI00_1[		SUPPLY SW	D SYSCON	USARTO USARTI WKT			
16	PIO0_9/ADC_4	CN7[9]/CN3[6]/R		CN3[6] (D12)	PICO_9[	1 PIO0_9	PI00_9[							
18	VDD	CN7[7]/JP5/R37/		0110[7][011]	1100_0[	, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	nee of m						an Annual to the	
√ 19	VSS	GND											<ul> <li>Generated code</li> </ul>	
20	PIO0_7/ADC_1/A	CN7[5]/CN3[2]/J	I2C_SDA	CN3[2] (D14	000	Periphe	ral CTIMER	2					✓ Update code enabled	
21	VREFP	CN7[4]/JP7/MCU	DEDUG LIADT DV	010(0) (00)	All CTIMER0 sign	als for routing	-		_ LPC804	M101IDH24	SSOPA 24 package		board/pin_mux.c	
23	PIO0_14/ACMP L.	CN7[2]/CN3[1]/J	I2C SCL	CN3[1] (D15	CAPTURE, 0	- 21 pins							board/pin_mux.h	
24	PIO0_19/DACOUT	CN7[1]/CN8[2]/PI		CN8[2] (D6)	CAPTURE, 1	- 21 pins								
					CAPTURE, 2	- 21 pins			* ~ •			2	<ul> <li>Functional groups</li> </ul>	
					MATCH, 1 - 2	21 pins			8 9 9	6 6 5	WDC DO T	2	BOARD_InitPins	
					MATCH, 2 - 2	21 pins			6 (P)			( ) Y	BOARD Initi EDaBins	
					MATCH, 3 - 2	21 pins			2 2		8			
									e/W		B.	2	BOARD_InitDEBUG_UARTPINS	
					Route All	Unroute A			1.00				P BOARD_InitSWD_DEBUGPins	
					Males are also			Devide a Detaile devi	5		😑 💿 😑 Signal CTIME	RO, MATCH, O	P BOARD_InitI2CPins	
					Make sure privaç	gnai assignme	nt is correct in	Routing Details view.			Reutable also ( sincel souther			
								Done			181 SWCLK/PIO0 3		1- contognitor rotarity	
Routing (	Details										[7] PIO0_4/ADC_11		M Other tools	
Pins Sign	als Q type filter										[6] RESETN/PIO0_5		• Other tools	
Routing De	stails for ROARD Joit	Dine 1									[20] PIO0_7/ADC_1/ADC [17] PIO0_8/ADC_5	IPVREF		
#	Perioheral 1	Sinnal Arrow	Routed ninisin Labe	i ideo	tifier Directi	on GRIO	initial sta Mor	e invert	Hysteresis Open	drain DAC Mode	[16] PIO0_9/ADC_4			
10	CTIMERO I	MATCH, 0	[10] PIOO_11 CN6	[10]/CN LED	BLUE Output	t n/a	Pull	Jp Disabled	Enabled Disat	ied n/a	[11] PIO0_10/ADC_7	TOLVIN	A Problems 🕄	B 🍸 😁 🗖
											[5] PIQ0_12	(TODAIN		
											[4] PIO0_13/ADC_10		Level V Resource Issi	ue
											[23] PIOO_14/ACMP_I3/4	DC_2	Sa Warning USARTO Per	ipheral USARTO is not initiali
													S Warning 12C0 Per	ipheral I2C0 is not initialized
												Done		
Lab02_3												1	U <u>NXP LPC804</u> • (La	ab02_3)

- 8. Press *Done* in the individual dialog boxes and then *Update Code*.
- 9. Build a project, program the microcontroller and check the operation. The LED (blue) should change state 2 times per second (1 flash every second).

#### 3. Timers and counters

#### IV. CTIMER - PWM mode

1. Go to *Pheriperals* and change the configuration of CTIMER0 to PWM and set the values as below:

CTIMERO 🔀		
Standard counter/timer	Peripheral drivers (Device specific)]	
ame CTIMERO		Custom name
Inde DWM	Peripheral CTINEDO	
ING PWW	· Polyheral CHIMERU	
<ul> <li>Timer/counter general configura</li> </ul>	đion	Preset Custom
<ul> <li>Timer counter configuration</li> </ul>		
Timer mode	Timer (bus clock source)	•
Bus clock source	System clock - BOARD_BootClockFR018M: 9 MHz, BOARD_BootClockFR024M: 12 MHz, BOARD_BootClockFR030M: 15 MHz	T
Clock source frequency	9 MHz (BOARD_BootClockFR018M)	¥
Timer input frequency/prescaler	1	
Calculated prescaler	1	
Calculated timer input frequency	9 MHz; 111.111 ns	
Start timer in initialization code		
<ul> <li>PWM channels configuration</li> </ul>		
PWM period channel	PWM channel 3	•
PWM frequency/period	9000	
PWM period [ticks]	9000	
Calculated PWM frequency/period	1 kHz; 1 ms	
✓ PWM channels +	x	
PWM_0		
Channel ID	PWM_0	
PWM channel	PWM channel 0	٣
PWM duty	50	
PWM duty period [ticks]	50	
Calculated PWM duty period	5.556 µs	
Enable PWM duty interrupt requ	est 🗌	
<ul> <li>Interrupt and callback setting:</li> </ul>		
Configuration enabled (a	natch/capture/pwm channel interrupt is enabled)	
<ul> <li>Interrupt</li> </ul>		
Interrupt  Interrupt	CTIMER0_IRQn	T
Interrupt Interrupt Enable priority initialization	CTIMERO_JRQn	T
Interrupt Interrupt Enable priority initialization Priority	CTIMERO_IRQn	T
Interrupt Interrupt Enable priority initialization Priority Callback mode Sin	CTIMERO_IRQn 0 Is collipsek	· · · · · · · · · · · · · · · · · · ·
Interrupt Interrupt Enable priority initialization Priority Callback mode Sin Change collback function	CTIMERO_IRQn  CIMERO_IRQn  cimetal control of the c	

#### 2. Press *Update Code* and modify the program code:



3. Timers and counters



Adding the *EnableIRQ* function is necessary due to a bug in Config Tools (does not set the interrupt activation flag in the generated code).

Build a project, program the microcontroller and check the operation. Open the terminal and check the LED brightness control operation.

#### V. Exercises

- 1. Add an additional PWM channels (PWM\_1 and PWM\_2) to *CTIMER0*, connect its outputs to PIO0\_12 (Green LED) and PIO0\_13 (Red LED). Write a program to control LEDs brightness using the terminal. Send a mark:
  - a: Blue PWM ++ z: Blue PWM - s: Green PWM ++ x: Green PWM - d: Red PWM ++ c: Red PWM - -