

Teaching online electronics, microcontrollers and programming in Higher Education

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

2. Debbuger console and GPIO

Lead Partner: Warsaw University of Technology

**Authors: Daniel Krol** 

University of Applied Sciences in Tarnow



2. Debbuger console and GPIO

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 Where other published and unpublished source materials have been used, these have been acknowledged.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

2. Debbuger console and GPIO

#### I. New project and debugger console

1. Create a new project for the *LPCXpresso804* board:



2. Name the project eg *Lab01* and keep the default configuration:

		SDK Wizard	1				SDK Wizard			
The source from the SDK will be copil	ied into the workspace. If you want to use linked files, p	slease unzip the 'SDK_2.x_LPCXpresso804' SDK.	NP 🖆				JUK WILLU		N	P 🖆
Configure the project				Advanced project	t settings					
Project name: Lab01		Project name suffix:		+ C/C++ Library Settings						
Use default location				Set library type (and hostin	a variant)	0				
Location: /Users/daniel/Documents/Mi			Browne				•			
Device Parkenes	Board	Project Type	Project Onlines	Redib: Use floating point	: version of printf			NewlibNano: Use floating	point version of printf	
O LPC804M101JDH24	O Default board files	C Project C++ Project	SDK Debug Console Semihost Q UART		care print					
C LPC804M101JDH20	<ul> <li>Empty board files</li> </ul>	C Static Library C++ Static Library	CMSIS-Core	Redirect SDK "PRINTF" t	o C library "printf"			Redirect printf/scarf to ITI	M	
O LPC804M111JDH24			C Lopy sources	Minclude semihost HardFa	ult handler			Redirect printl/scard to UA	URT	
A. 1884 A. 1114 A. 1114 A.										
Add or remove SDK potterre compose	urba	e	components selection summary (E) (E)							
Operating Systems Drivers CMSIS	Drivers Utilities Middleware Board Components A	bstraction Laver Software Components	type to filter	- MOULC Compiler						
Petrone		17 % D D	Name Description Versio Info							
Linnis		8 % C C	Operating Systems	Language standard Comp	oller default					8
type to filter			> E Utilities	* MCU Linker						
Name	ADC Driver	2.5.0		Unk application to RAM						
apt	CAPT Driver	2.1.0		- Henry Oreformities						
C de clock	Clock Driver COMMON Driver	2.3.1		<ul> <li>Memory consignation</li> </ul>						
Cimer	CTimer Driver	2.2.1		Memory decails						
🔲 🙀 dəc	DAC Driver	2.0.2		Default LinkServer Flash D	tiver					Browse
2 2 gpio	GPID Driver I2C Driver	2.1.7		Type	Name	Alias	Location	Size	Driver	
0 8 kp	IAP Driver	2.0.4		Flash	PROGRAM_FLASH	Flash	0x0	0x7180	LPC80x_32.cfx	
🖸 💩 iscon	IDCON Driver	2.0.1		Flash	BOOT_FLASH SRAM	Flash2 BAM	0x7180 0x10000000	0x90 0xfe0		A
bc_acomp	LPC_ACOMP Driver	2.1.0		RAM	MP_SRAM	RAM2	0x10000fe0	0x20		
be_crc	CRC Driver	2.1.1								
- Apiet	PINT Driver	2.1.8		Add Flash Add R	AM		Split	Join Delete		
🗆 💩 plu	PLU Driver	2.2.1								
🖉 🔆 power_na_Rb	Power Driver	2.0.0		import Merge	. Export. Gene					
?		< Back	Next > Cancel Finish	0				< Back	Next > Cancel	Finish

2. Debbuger console and GPIO

#### 3. A code skeleton will be generated:

```
/**
 * @file
* @brief
                  Lab01.c
Application entry point.
  */
*/
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC604.h"
#include "fi_debug_console.h"
  /* TODO: insert other include files here. */
/* TODO: insert other definitions and declarations here. */
 * @brief Application entry point.
int main(void) {
       /* Init board hardware. */
BOARD_InitBootClocks();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
      /* Init FSL debug console
BOARD_InitDebugConsole();
#endif
      PRINTF("Hello World\n");
      /* Force the counter to be placed into memory. */
volatile static int i = 0;
      /* Enter an infinite loop, just incrementing a counter. */
while(1) {
             i++ :
             /* 'Dummy' NOP to allow source level single stepping of
    tight while() loop */
    __asm volatile ("nop");
      3
      return 0 ;
```

Add a "\r" tag to the end of the text in the PRINTF function.

4. Connect the *LPCXpresso804* board with the USB interface labeled *Emulator* to the computer:



2. Debbuger console and GPIO

5. Right click on the project name, select Properties and then go to C/C++ Build -> Settings and select Preprocessor. Check the value of the SDK\_DEBUGCONSOLE constant, which should be set to 1 (otherwise set to 1), as in the picture below. The constant activates the debugger console that uses UART. The other two constants disable real number support by default (a limited version of the library that takes up less memory is used). To display real types in the console, remove the CR\_INTEGER\_PRINTF constant and change the value of the PRINTF\_FLOAT\_ENABLE constant to 1.

	- Da 12 🗮 🕱		Properties for Lab01	
Project Explorer 22 🕮 Registers 🏇 Faults 🛼 Peripherals+	= □ # � <b>Ø</b> - 8	type filter text	Settings	↓ → ↓ 8
• Construction     Norm     >       • Construction     • Initial     >       • Constru	C • C • 15 •	<ul> <li>Tesource Builders Builders Build Varables Environment Logging MCU settings</li> <li>Brodowner MCU settings</li> <li>Orce-+ General MCUXpress Config Tools Project Natures Project References Project References Task Tage</li> <li>Validation</li> </ul>	Configuration: Debug [Active] Tool Setting: Pluid steps Puid Atfact in the Date: Da	Manage Configurations         y Parsers         () Error Parsers         nostdinc)         () () () () () () () () () () () () () (
Miscellars Source >				Restore Defaults Apply
Edit proje     Properties     X I      MOUXprove commy receive     Quick Settings>		?		Cancel Apply and Close

6. Build the project by clicking Build and then program the layout by clicking GUI Flash Tool, leaving the default settings in the following programmer windows:

•••	c workspace_LPC804 - Lab01/source/Lab01.c - MCUXpresso IDE	
🖆 • 🖩 🕲   🗞 • 🚯   🖉 🏷 💷   🔍 🕨 🗉 🖬 🗚 20-00   🗟 🛠	10 10 10 20 20 20 20 20 20 20 20 20 20 20 20 20	
🍋 Project Explorer 🐹 👯 Registers 🎋 Faults 🧏 Peripherals+ 👘 🗖	د Lab01.c ک	
E 🕏 7 🖶 🍫 🕱 🕯	2⊕ * Copyright 2016-2021 NXP[]	
✓ Kab01 <debug></debug>	310 /**	
Project Settings	32 * @file Lab01.c	
> 🖑 Binaries	33 * @brief Application entry point.	
> 🗊 Includes	34 */	
> 😕 CMSIS	36 #include <\$(010.n>	
l i internetti i i i i i i i i i i i i i i i i i i	So waterdad boundin	

7. Launch the terminal:



2. Debbuger console and GPIO

8. In the settings, select *Serial Terminal* and the baud rate 9600. In the serial port field, select /*dev/cu.usbmodem020140202* from the list of available ports. In Windows, select *COMMx* port. Note, depending on the version of the evaluation board, in the place of "X" there may appear no other code than in the picture below:

Choose termir	al: Serial Terminal	6
Settings		
Serial port:	/dev/cu.usbmodem020140202	_
Baud rate:	9600	
Data size:	8	0
Parity:	None	0
Stop bits:	1	0
Encoding:	Default (ISO-8859-1)	0

9. Press the RESET button on the *LPCXpresso804* board. The text sent with the *PRINTF* function should appear in the terminal window:

👘 Installed SDK	Properties	🖹 Problei	ms 📃 Console	🎤 Terminal 🔀	🚮 Image Info	🛃 Debugger Cor	1	Offline P	eriph	_	. 🗆
						E	1	ф <b>В</b>		1	ð 🖉
📃 /dev/cu.usbmo	dem020140202	2 23									
Hello World											

10. Write a simple "echo" program that prints the received characters to the console, preceded by the string "Character:". To do this, modify the code in the main function as below:



Build a project, program the microcontroller and check the program operation in the terminal console.

2. Debbuger console and GPIO

#### II. Controlling of LEDs

- 1. Create a new project for the *LPCXpresso804* board and name it *Lab01\_1*.
- 2. You must configure 3 *GPIO* lines to control the RGB LEDs. To do this, rightclick on the project name and select *MCUXpresso Config Tools -> Open Pins*, as in the picture below:

extremes to this watherers A Lan	ts 7 Peripherals+	Le Labol.c 32 la fsl_debug_console.h
	8378%	2# * Copyright 2016-2021 NXP
New	,	310/4*
Go Into		32 * @file Lab@l.c 33 * @frief Application entry point.
		34 ×/
Open in New Window		36 #include "Solid"
Show In	∠36 W 3	37 finclude "peripherals.h" 38 finclude "pin mus.h"
Show in Local Terminal	>	39 #include "Clock config.h"
Copy	36 C	40 #include [fsl_debug_console.h"
Paste	36 V	A 43 /* TODD: insert other include files here. */
Delete		244 /* TODD: insert other definitions and declarations here. */
iurce	>	45 46©/*
ove		47 * @brief Application entry point.
name		49e int main(void) {
moort		50 51 char c
A Export		52 /* Init board hardware. */
Exportan		53 BOARD_InitBootCocks(): 54 BOARD InitBootCocks():
luild Project		55 BOARD_InitBootPeripherals();
san Project		56 FITMET BUAKD_INIT_DEBUG_CUNSULE_PENIPHERAL 57 /* Init FSL debug console. */
Refresh	FS	58 BOARD_InitDebugConsole();
e Project		59 menalt 60
e Unrelated Projects		61 while(1) {
d Oradlau matterna		63 PRINTF("Please enter a character\r\n");
ingurations		64 c=GETCHAR(); 65 PDINTE("(haracter: hc/r/o"), c);
rgets	,	66 ) · · · · · · · · · · · · · · · · · ·
	,	6/ return 0;
ling Tools	>	69
Run As	>	
Debug As	>	
rofile As	>	
Restore from Local History		
aunch Configurations	>	
Utilities	>	🕲 bereicht 2014 💭 Bereichte 🕐 Berline 💭 Berline 101 🕅 🖓 Tereicht 🔍 Berline 104 🔍 Berline 104 🔍 Berline 104 🔍 Berline 104
🔀 SDK Management	>	installed SDK in Properties in Properties in Console 22 de lemma in Image Into up beougger Con 22 Utiline venon - U Memory 22 UP Head
Tools	>	
Validate		10:35:49 ++++ Incremental Build of configuration Debug for project Lab01 ++++
MCUXpresso Config Tools	> (iii) Open Pins	-r -j15 all
Run C/C++ Code Analysis	UI Open Clocks	- Notifing to be done for act -
Team	> 🕴 Open Periphe	als 6:49 Build Finished. @ errors, @ warnings. (took 153ms)
Compare With	>	
Configure	> Provision Open Tools O	terview .
Source	>	

3. From the *Functional Group* menu select the *BOARD\_InitLEDsPins* preset, then activate it by selecting the flag icon on the left. The window now shows the automatically configured lines connected to RGB LEDs on the prototype board:

000							ė	workspace	_LPC804 -	Lab01_2/so	urce/Lab01.c	- MCUXpre	ISO IDE					
📬 - 🔢 🍕	a 📸 Lab01_2		🔹 🏀 🖪	💈 Update Code 🔹 🖡 Fu	unctional Grou	BOARD_	InitLEDsPins		<b>7</b> 🗐 💜	> 🕒 🕞 1	i ili 3. % i	k 🚳 - 💁	· 🥭 🛷 •		- 🏷 🗳 🗇 - 🔿 -	<b>1</b>	c	2 🖻 🗶 🖲 ጣ የ 🔿
Pins 33	Peripheral Signa	/s				BOARD_ BOARD	InitPins InitLEDsPins								Q Q (? 🗇 📼 🤊	🗧 🌴 Overview 😫 💽 Code	Preview 🚺 Registers	
880	WW -00-0	🗲 💈 🕖 🔍 ty				BOARD_	hitDEBUG_UARTP	ns								> Configuration - Ge	neral Info	
Pin	Pin name	Label	Identifier	Arduino UNO R3 Cr	oi SPI	G BOARD_	Initi2CPins											
1	PI00_18	CN6[1]/CN3[9]/PL		CN3[9] (D9)	P100_18[]	P BOARD_	InitBUTTONsPins									<ul> <li>Configuration - HV</li> </ul>	linto	
2	PIO0_16/ACMP_L.	CN6[2]/CN5[2]/U.		CN5[2] (A1)	P100_16[]	P100_76	PI00_16[.		-							Processor: LPC80	4	
3	PIC0_17/ADC_9	CN6[3]/CN5[3]/PI.		CN5[3] (A2)	P100_17[	PI00_17	PIO0_17[.			8		<i>v</i> .	4 10	.5		Part number: LPC80	4M101JDH24	
4	PICO_13	S1/CN8[4]/CN8[4.	LED_RED;S1	CN8[4] (D4)	PI00_13[]	P100_73	PID0_13[.			š		× ×	8 8	Q.		Core: Cortes	MOP	
6	RESETNIDIOD 5	CN8/81/CN1/101/	DEBLIG SWD RE	CN4[3] (PST)	PIO0_12[]	800.5	PIO0_12[.			1 1 0		8.4	A/6	20 20		Board: LPCXp	resso804	
7	PIO0_4/ADC_11	CN6[7]/CN8[7]/C.	DEBUG_UART_TX	CN8[7] (D1); CN5.	. P100_4[]	PI00_4	PID0_4[		1		76FF	1 8 ĝ	00	00		SDK Version: ksdk2	_0	
8	SWCLK/PIO0_3	CN6[8]/CN1[4]/U.	. DEBUG_SWD_SW.		P100_3[]	PI00_3	PIO0_3[				>		a. a.	a. a.				
9	SWDIO/PIO0_2	CN6[9]/CN1[2]/U.	. DEBUG_SWD_SW.		P100_2[]	Pi00_2	PID0_2[									> Project		
10	PI00_11	CN6[10]/CN8[3]/.	LED_BLUE	CN8[3] (D5)	PIO0_11[]	PI00_11	PID0_11[.									Plan.		
11	PIO0_10/ADC_7	CN6[11]/VR1/CN		CN5[1] (A0)	P100_10[]	PI00_10	PIO0_10[.									<ul> <li>Pins</li> </ul>		
12	PIOD_21/ACMP_15	CN7[12]/CN8[5]/.		CN2(10) (D3)	PIO0_21[]	PIOU_21	PIOD_21[.										Argures prin routing, including certies, voltage/power rails.	I functional electrical pin and run-time pin configuration.
14	PICO 15/ADC 8	CN7[11]/CN3[8]/		CN3[8] (D10)	PIO0 15[]	PID0 15	PIO0_15[		AC	MP ADCO	CAPT	CTIMERO	DACO	GPIO			,	
15	PIO0_1/ADC_0/A	CN7[10]/CN3[5]/.		CN3[5] (D13); CN.	. PIO0_1[]	PI00_1	PIO0_1[		120	0 I2C1	LVLSHFT	PINT	PLU	SPID				
16	PIC0_9/ADC_4	CN7[9]/CN3[6]/R.		CN3[6] (D12)	P100_9[]	PI00_9	PIO0_9[		SU	PPLY SWD	SYSCON	USARTO	USART1	WKT			n 🗛	
17	PIO0_8/ADC_5	CN7[8]/CN3[7]/R		CN3[7] (D11)	P100_8[]	PI00_8	PIO0_8[											
✓ 18	VDD	CN7[7]/JP5/R37/														<ul> <li>Generated code</li> </ul>		
19	VSS	GND	100.004	0000001/004/01/000	000 7/ 1	000.7	0100 77									✓ Update code epable	ed	
20	VIDEED	CN7[5][CN3[2][U	Izc_sbw	UN3[2] (D14); UN.		100_7	PID0_7[											
22	PIOD 0/ACMP 11	CN7[3]/CN8[8]/J	DEBUG UART RX	CN8[8] (D0)	P100_0[]	P100.0	PIO0_01			LPC804M	4101JDH2	4 – TSSOF	A 24 pac	kage		Doardypin_mux.c		
23	PICO_14/ACMP_L.	CN7[2]/CN3[1]/J	I2C_SCL	CN3[1] (D15)	PIO0_14[]	PI00_14	PID0_14[.					_			_	board/pin_mux.h		
24	PIO0_19/DACOUT	CN7[1]/CN8[2]/PL		CN8[2] (D6)	PIO0_19[]	PI00_19	PIO0_19[.									and the state of a second		
										8 0 0	12	7 A	20	0 5		<ul> <li>Functional groups</li> </ul>		
										n a a	0 0	01	INS 00	AD NO		BOARD_InitPins		
										1 5 5			-	01.14		BOARD_InitLEDs	Pins	
										5 G				100			10 UADTO- 10	
										1919 1917				- E		P BOIND_INDEBC	S_CONCEPTINS AS	
										8						BOARD_InitSWD.	DEBUGPins	
										6						BOARD_Initi2CPI	ns 🕰	
																P BOARD_INIBUTT	UNSPINS	
Routing I	Details														E *			
Dine Sion	ale 0, turne filter															<ul> <li>Other tools</li> </ul>		
Final Sign	and the most																	
Routing D	etails for BOARD_Ini.	. 3 🕻	8 ~ ~													(00) (	D )	
# Per	ripheral Signal	Arrow Routed	pin/signal Label			Identifier	Direction 4	PIO initial state	e Mode	Invert	Hysteresis	Open drain	DAC Mode					
10 GP	10 PI00, 11	-> [10] Pl	00_11 CN6[10]/	CN8[3]/D4/PIO0_11/	M_PIO0_11	LED_BLUE	Output I	ogical 1	Inactive	Disabled	Enabled	Disabled	n/a			A Problems 33		BY
5 GP	10 PI00, 12	-> [5] PIC	10_12 S2/CN8[6	[/CN6[5]/D3/M_PIO0	_12/PIO0_12	LED_GREEN	Output I	ogical 1	Inactive	Disabled	Enabled	Disabled	n/a					
- GP		> (4) PIC	0 I/CN8[4	11-0140[#]/Dzhw_P100		CCO_RED	o a upor	orgenedit i	- NPLOVE	Unsaured	Shapleo	unsacted	.00			Level ~ Resource	:e	Issue
																Saming USART	j.	Peripheral USARTO is not in
																S Warning 12C0		Peripheral I2C0 is not initial
sb01_2																	O NXP LPC8	04* (Lab01_2)

2. Debbuger console and GPIO

4. Active presets can be checked by clicking the *Functional group* properties icon. The list of presets is displayed in the opened window. You can edit them and add your own:

000							e workspac	e_LPC804 - L	.ab01_2/sour	ce/Lab01.c -	MCUXpres	so IDE				
📑 • 🖂 I	🐚 🔜 Lab01_2		• 🛪 🔺	Update Code	• • Functional Group	BOARD_InitLEDsPins		🗖 🔍 🗠	0.410	19 3. O .6	S- 9.	🥭 \land 刘	[创+羽+	🗠 🗢 🚓 🖘 😅		오 😥 🗶 💷 🗤 🕴 🔿 📓
🗄 Pins 🕄	Peripheral Signa	als				- 0	😳 Package 😒						Q	Q (? 🖓 🖼 🖱 🗖	🔺 Overview 🕮 🔥 Code Preview 🕕 Regi	iters 🗖 🗆
Pin C	} ₩ ₩ -	🗲 💈 😥 🔍 typ	e filter text	Arduino UNC	R3 Cor SPI	GPIO USART									> Configuration - General Info	
1	PIO0_18	CN6[1]/CN3[9]/Pl		CN3[9] (D9)	PI00_18[]	PIO0_18 PIO0_18[.									Configuration - HW Info	
2	PIO0_16/ACMP_J	CN6[2]/CN5[2]/U		CN5[2] (A1)	PI00_16[]	PIO0_16 PIO0_16[.		E							Processor: LPC804	
4	PI00_13	S1/CN8[4]/CN6[4	LED_RED:S1	CN8[4] (D4)	PI00_13[]	PI00_73 PI00_13[,		Ö			5	7 0 8	2		Part number: EPC804M101JDH24	
5	PI00_12	S2/CN8[6]/CN6[5	LED_GREEN;S2	CN8[6] (D2)	PI00_12[]	PIO0_12 PIO0_12[.		ND/6			ADK.	ADC /ADC			Core: Contex-Mor	
6	RESETN/PIOO_5	CN6[6]/CN1[10]/	DEBUG_SWD_RE	CN4[3] (RST	F) PIO0_5[]	PIO0_5 PIO0_5[		10	2 2	ь ў "	0 8	6 6 6	2 2		COX Manian India 0	
8	PIOU_4/ADC_11 SWCLK/PIO0_3	CN6[/]/CN8[/]/C	DEBUG_UART_TX	CN8[7] (D1);	PIO0_4[]	PIO0_4 PIO0_4[		Did	Did Did	VS VS	07 X	NA NA	2 2		dok version: ksok2_0	
9	SWDIO/PIO0_2	CN6[9]/CN1[2]/U	DEBUG_SWD_SW.		PIO0_2[]	PID0_2 PID0_2[									> Project	
10	PI00_11	CN6[10]/CN8[3]/	LED_BLUE	CN8[3] (D5)	PIO0_11[]	PIO0_11 PIO0_11[.									1. B.	
11	PIO0_10/ADC_7	CN6[11]/VR1/CN		CN5[1] (A0)	PIO0_10[]	PIO0_10 PIO0_10[.									<ul> <li>Pins</li> </ul>	
13	PIO0_20	CN7[12]/CN3[10]		CN3[10] (D8	PIO0_20[]	PI00_20 PI00_20[.									properties, voltage/pow	r rais, and run-time pin configuration.
14	PI00_15/ADC_8	CN7[11]/CN3[8]/		CN3[8] (D10	PIO0_15[]	PI00_15 PI00_15[.		ACM	P ADCO	CAPT	CTIMERO	DAC0	CPIO			
15	PIO0_1/ADC_0/A	CN7[10]/CN3[5]/		CN3[5] (D13	3); CN PIO0_1[]	PIO0_1 PIO0_1[		SUPP	LY SWD	SYSCON	USARTO	USART1	WKT			
10	PIOD_9/ADC_4	CN7[9]/CN3[6]/R		CN3[6] (D12 CN3[7] (D11	2) PIOU_9[] (	PIO0_9 PIO0_9[										
18	VDD	CN7[7]/JP5/R37/						Functional gr	oup propertie	15					Generated code	
✓ 19	VSS	GND													V Undate code enabled	
20	VREEP	CN7[5]/CN3[2]/J	IZC_SDA	CN3[2] (D	Functional groups	010 ^ ~	Name:	BOARD_InitLED	sPins							
22	PIO0_0/ACMP_I1	CN7[3]/CN8[8]/J	DEBUG_UART_RX	CN8[8] (DI	BOARD InitPins		-	Set custom #	define prefix						Boardypin_make	
23	PIO0_14/ACMP_I	CN7[2]/CN3[1]/J	I2C_SCL	CN3[1] (D	BOARD_InitLEDs	Pina	Prefix:								board/pin_mux.h	
24	PIO0_19/DACOUT	CN7[1]/CN8[2]/Pl		CN8[2] (DI	BOARD_InitDEBU	IG_UARTPins		Clock gate er	nable						<ul> <li>Functional groups</li> </ul>	
					BOARD_InitSWD_	_DEBUGPins	Description:	Configures pin	routing and opti	onally pin electi	ical features.					
					BOARD_InitBUTT	ONsPins									19 BORRD_INTERN	
															BOARD_InitLEDsPins	
_															BOARD_InitDEBUG_UARTPins 🚵	
_															BOARD_InitSWD_DEBUGPins	
															BOARD Initi2CPins	
_																
_															P BOWKD_INIBOTTOWSHINS	
Routing	Details													<b>=</b> - 0	V Other tools	
Pins Sig	anals Q type filter														$\sim$	
Routing	Details for BOARD_Ini.	3 🕻														
# P	eripheral Signal	Arrow Routed	pin/signal Label												A Techner M	
10 0	PIO PIO0, 11	-> [10] PK	0.12 S2/CNB(6	CN8[3]/D4/PI											Problems 24	
4 0	PIO PIO0, 13	-> [4] PIO	0_13 \$1/CN8[4	]/CN6[4]/D2/				Called by def	ault initialization	function						
															Level V Resource	Issue
												Cancel	ОК		Warning 12C0	Peripheral 05ker10 is not initialized
Lab01_2															U <u>NX</u>	P LPC804* (Lab01_2)

5. Select *Update Code* (the picture above) to generate the code based on the entered configuration. The code will be added to files marked with the "change" icon:



By clicking on change you can see what changes will be made to individual files with the source code.

2. Debbuger console and GPIO

- 6. Accept the changes with the OK button.
- 7. The constants describing the individual lines controlling the RGB LEDs were generated in the *pin\_mux.h* file:

尾 Lab	o01.c	🔓 pin_mux.h δ	3			- 0
280	9/*!					
29	* @br:	Lef Configure	s pin routing and	optionally pin electric	al features.	
30	*					
32	void B	DARD_InitPins	(void); /* Functi	on assigned for the Cort	ex-M0P */	
33	****		VC EN 0	- Lobard Fachla bustons		
34	#define	IOCON_PIO_H	NV DI 0x00u /:	*:<@brief Enable hystere *!<@brief Input not inve	sis */ rt */	
36	#define	IOCON_PIO_M	ODE_INACT 0x00u /	*!<@brief No addition pi	n function */	
37	#define	10CON_PI0_0	D_DI 0x00u /	*!<@brief Disables Open-	drain function */	
396	)/*! @na	ame PIO0_11 (	number 10), CN6[1	0]/CN8[3]/D4/PI00_11/M_P	100_11	
40	@{ */	/				
41 42	/* Symb	ols to be us	ed with GPIO driv	er */		
43	#define	BOARD_INITL	EDSPINS_LED_BLUE_	GPIO GPIO	<pre>/*!&lt;@brief GPIO peripheral base pointer */</pre>	
44	#define	BOARD_INITL BOARD_INITL	EDSPINS_LED_BLUE_	GPIO_PIN_MASK (10 << 110 PORT AU	/*!<@brief GPIO pin mask */ /*!<@brief PORT device index: 0 */	
46	#define	BOARD_INITL	EDSPINS_LED_BLUE_	PIN 11U	/*!<@brief PORT pin number */	
47	#define	BOARD_INITL	EDSPINS_LED_BLUE_	PIN_MASK (1U << 11U)	/*!<@brief PORT pin mask */	
40					/* @/ */	
500	/*! @na	ame PIO0_12 (	number 5), S2/CN8	<pre>[6]/CN6[5]/D3/M_PI00_12/</pre>	PI00_12	
51	@{ */					
53	/* Symb	ools to be us	ed with GPIO driv	er */		
54	#define	BOARD_INITL	EDSPINS_LED_GREEN	_GPIO GPIO GRID RIN MASK (11 er 12	/*!<@brief GPIO peripheral base pointer */	
56	#define	BOARD_INITL	EDSPINS_LED_GREEN	_PORT ØU	/*!<@brief PORT device index: 0 */	
57	#define	BOARD_INITL	EDSPINS_LED_GREEN	PIN 12U	/*!<@brief PORT pin number */	
58	#detine	BUARD_INITL	EDSPINS_LED_GREEN	_PIN_MASK (10 << 120)	/*!<@DF1ET PORT pin mask */ /* @} */	
60						
616	e/*!@na @{*/	ame PIO0_13 ( /	number 4), S1/CN8	[4]/CN6[4]/D2/M_PI00_13/	PI00_13	
63	et					
64	/* Symt	pols to be us	ed with GPIO driv	er */	/*L-Shriaf CDIO peripheral base pointer */	
66	#define	BOARD_INITL	EDSPINS_LED_RED_G	PIO_PIN_MASK (1U << 13U)	/*!<@brief GPIO pin mask */	
67	#define	BOARD_INITL	EDSPINS_LED_RED_P	DRT ØU	/*!<@brief PORT device index: 0 */	
69	#define	BOARD_INITL BOARD_INITL	EDSPINS_LED_RED_P EDSPINS_LED_RED_P	IN 130 IN MASK (10 << 130)	/*!<@brief PORT pin mask */	
70					/* @} */	
71	) /*!					
73	* @br:	ief Configure	s pin routing and	optionally pin electric	al features.	
74	*					
76	void B	DARD_InitLEDs	Pins(void); /* Fu	nction assigned for the	Cortex-M0P */	

8. Modify the code in the main function so that sending "a" causes the LED to light red. In turn, sending the "z" character should extinguish the red component:



Build a project, program the microcontroller and check the program operation.

2. Debbuger console and GPIO

#### **III. Button operation**

- 1. Create a new project for the *LPCXpresso804* board and name it *Lab01\_3*.
- 2. Open the pin configuration tool again: *MCUXpresso Config Tools -> Open Pins*. Add the *BOARD\_InitBUTTONsPins* preset and then click *Update Code*.
- 3. Modify the code in the main function so that pressing the S1 and S2 buttons displays the appropriate messages:

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "fil_debug_console.h"
  /* TODO: insert other include files here. */
 /* TODO: insert other definitions and declarations here. */
 * @brief Application entry point.
 int main(void) {
               /* Init board hardware. */
               BOARD_InitBootPins();
BOARD_InitBootFins();
BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
/* Init FSL debug_console. */
DODDD_InitBootPeripheral();
              BOARD InitDebugConsole();
 #endif
              PRINTF("Start\r\n");
              while(1) {
                         if(GPI0_PinRead(BOARD_INITBUTTONSPINS_S1_GPI0, BOARD_INITBUTTONSPINS_S1_PORT,
BOARD_INITBUTTONSPINS_S1_PIN) == 0)
                            PRINTF("S1\r\n");
if(GPI0_PinRead(BOARD_INITBUTTONSPINS_S2_GPI0, BOARD_INITBUTTONSPINS_S2_PORT,
BOARD_INITBUTTONSPINS_S2_PIN) == 0)
                            PRINTF("S2\r\n");
               }
               return 0 ;
```

Build a project, program the microcontroller and check the program operation.

4. Modify the program so that it reacts only to a single button press (falling edge detection):

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "clock_config.h"
#include "fsl_debug_console.h"
/* TODO: insert other include files here. */
/* TODO: insert other definitions and declarations here. */
/*
* (@brief Application entry point.
*/
int main(void) {
    bool sw1=false, tm1=false;
    bool sw2=false, tm2=false;
        /* Init board hardware. */
        BOARD_InitBootPeripherals();
        BOARD_InitBootPeripherals();
#ifndef BOARD_InitDootPins();
        BOARD_InitDootPeris();
#ifndef BOARD_InitBootPeripherals();
#ifndef BOARD_InitDootPeris();
#
```

#### **Programing of embedded systems** 2. Debbuger console and GPIO



Build a project, program the microcontroller and check the program operation.

#### **IV. Exercises**

- 1. Modify the LED control program so that it is possible to control three RGB diodes. Send a mark:
  - a: Red-On z: Red-Off s: Green-On x: Green-Off d: Blue-On c: Blue-Off
- 2. Write the same program using the switch-case statement.
- 3. Modify the button program so that it is possible to detect when a button is released.