

Teaching online electronics, microcontrollers and programming in Higher Education

### **Programing of embedded systems**

**10**. Parent application - virtual serial port

Lead Partner: Warsaw University of Technology

Authors: Daniel Krol

University of Applied Sciences in Tarnow



10. Parent application - virtual serial port

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 Where other published and unpublished source materials have been used, these have been acknowledged.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

10. Parent application - virtual serial port

#### I. Sterownik LED

 Należy skonfigurować 3 linie GPIO do sterowania poszczególnymi diodami RGB, tak jak na pierwszych zajęciach. W tym celu kliknij prawym przyciskiem na nazwie projektu i wybierz MCUXpresso Config Tools -> Open Pins. Z menu Functional Group wybierz preset BOARD\_InitLEDsPins, a następnie aktywuj go zaznaczając ikonę flagi po lewej stronie:

000						e workspace_LPC55	is69 - ZPSW_La	b01/source/ZPSW	_Lab01.c - M	ICUXpresso IDI				
	ZPSW_Lat	01 💌 🌪 🗛	Update Code + Func	tional Group	BOARD_InitLED	sPins 🔺 🛃	B   U \ \ D	0.000.000	0.016-9	• 🙋 🖉 • 📝	1월 - 월 <b>· 영 약 수 ·</b> 수 · [] -		Q	
Dine S2	Derinheral Signa				BOARD_InitPin		Expansion Meade					A Quandam St	Code Preview III Registers	
					BOARD_InitLED	UG_UARTPins	copartition reade				ddt 20 a	H OVER NEW 54	C Code Flerien ( negisters	
991	10.10.000	7 (2) Q type filter text			BOARD_InitSW	DEBUGPins						> Configurat	ion - General Info	
Pin	Pin name	Label Identifier	Arduino UNO R3 Cor	SPI	BOARD_Initi2C	Pins TONsPins						V Configurat	ion - HW Info	
1	PI00_18	CN6[1]/CN3[9]/PIO0	CN3[9] (D9)	SPI0:SCK[]	SPIO:PIO0.18 U	ARTC	J					Processor:	LPC804	
2	PI00_16/ACMP_14/A	CN6[2]/CN5[2]/U4[1	CN5[2] (A1)	SPI0:SCK[]	GPIO:PIO0,16 US	ARTC						Part number:	LPC804M101JDH24	
3	PIOD_17/ADC_9	CN6[3]/CN6[3]/PIOL S1/CN8[4]/CN6[4]/OLED_RED-S1	CNB[3] (A2) 3	SPI0:SCK[]	GPI0:PI00,17 US	ARTC	+					Core:	Cortex-M0P	
5	PI00_12	S2/CN8161/CN6151/C LED_GREEN-S2	CN8[6] (D2)	SPI0:SCK[]	GPIO:PIO0.12 U	ARTC	00			A 4 6	Ű	Board:	LPCXpresso804	
6	RESETN/PIO0_5	CN6[6]/CN1[10]/S3/ DEBUG_SWD_RES	E CN4[3] (RST)	SPI0:SCK[]	GPIO:PIO0,5 U	ARTC	Đ,		1	A A A	QV .	SDK Version:	ksdk2_0	
7	PI00_4	CN6[7]/CN8[7]/CN5 DEBUG_UART_TX	CN8[7] (D1); CN5[5] 5	SPI0:SCK[]	GPIO:PIO0,4	ARTC	51.0	0 4 0		6 6 7	0.15			
8	SWCLK/PIO0_3	CN6[8]/CN1[4]/U1[1 DEBUG_SWD_SWD	DC 1	SPI0:SCK[]	GPIO:PIO0,3 U	ARTC	04 94	OID BRV OID	NDI N	2 2 2	8 8	> Project		
10	PI00 11(ADC 6/WK	CN6[9]/CN1[2]/OT[1 DEBOG_SWD_SWD	CN8[3] (D5)	SPI0:SCK[]	GPI0:PI00,2 01	ARTC						Dine		
11	PI00_10/ADC_7	CN6[11]/VR1/CN5[1]	CN5[1] (A0)	SPI0:SCK[]	GPIO:PIO0,10 US	ARTC			-			• • • • • • •		
12	PI00_21/ACMP_I5	CN6[12]/CN8[5]/PIO	CN8[5] (D3)	SP10:SCK[]	GPIO:PIO0,21 U	ARTC							properties, voltage/power rails, a	and run-time pin configuration.
13	PI00_20	CN7[12]/CN3[10]/PK	CN3[10] (D8) 5	SPI0:SCK[]	GPIO:PIO0,20 U	ARTC								
14	PIOD_15/ADC_8	CN7[11]/CN3[5]/PIO	CN3[8] (D10) 3	SPID:SCK[]	GPI0:PI00,15 0	ARTC	ACMP	ADC0	стия	0 0400	CRIO			
16	PI00_9/ADC_4	CN7[9]/CN3[6]/R13/	CN3[6] (D12)	SPI0:SCK[]	GPIO:PIO0,9 U	ARTC	12C0	12C1 LVL5	HFT PINT	PLU	SPID		🕘 💼 🏦	
17	PI00_8/ADC_5	CN7[8]/CN3[7]/R20/	CN3[7] (D11)	SPI0:SCK[]	GPIO:PIO0,8 U	ARTC	SUPPLY	SWD SYSC	ON USARTO	0 USART1	WKT			
18	VDD	CN7[7]/JP5/R37/VDE										Generated	code	
20	V55 PIDD 7/ADC 1/ACM	CN7(51/CN3(21/JP25)2C_SD4	CN3[2] (D14): CN5[4	SPI0:SCK[ ]	GPIO:PIO0 7 1	ARTI						Update co	ie enabled	
21	VREFP	CN7[4]/JP7/MCU_VF	energy (energy energy	or re-0 or ()								board/pi	n_mux.c	
22	PI00_0	CN7[3]/CN8[8]/JP2/ DEBUG_UART_RX	CN8[8] (D0)	SPI0:SCK[]	GPIO:PIO0,0	ARTC	1.00		124 7554	DA 24		board/pi	n_mux.h	
23	PIO0_14/ACMP_13/A	CN7[2]/CN3[1]/JP4/I I2C_SCL	CN3[1] (D15)	SPI0:SCK[]	GPIO:PIO0,14 US	ARTC		.804M101JDF	24 - 1550	ля 24 раск	age			
29	PI00_19/040001	CN/[I]/CN8[2]/PID0	CH0[2] (D0) :	5P10/3GA[]	GPI0-PI00, 19 0:	ARTC						<ul> <li>Functional</li> </ul>	groups	
												BOARD_	InitPins	
								2 2 3 3	Es o			BOARD_	InitLEDsPins	
							014	7/A	8 5 6	2 K K	9CM			
_							9	8			0 2	10 00000		
							140	6			H 004	P BOARD_	InitSWD_DEBUGPins	
							2	)				BOARD_	Initi2CPins 🏭	
							Gid					P BOARD_	InitBUTTONsPins	
												<ul> <li>Other tool</li> </ul>	5	
-											-			
Routing	Details										<b>E</b>	(70	) (Ψ)	
Pins Sig	nals Q type filter											A Problems 23		в 7 - о
Routing	etails for BOARD											Channel Billion Land		
				la ser	-	lane of the second								
# Pi	enpneral Signal	Arrow Routed pin/signal Label		Identifie	r Directio	n GPIO initial state Mo	de Invert	Hysteresis	upen drain	DAC Mode		Level ~	Resource	Issue
7 12	SARTO RXD SARTO TXD	[22] PIO0_0 CN7[3]/C	N8[8]/JP2/PIO0_0 N8[7]/CN5[5]/JP24/PIO	DEBUG	UART_RX Input	n/a Pul	IUp Disable IUp Disable	d Enabled	Disabled	n/a n/a		🔈 Warning	USARTO	Peripheral USART0 is not initiali
												S Warning	12C0	Peripheral I2C0 is not initialized
ZPSW_Lab0	1												U NXP LPCBC	4* (ZPSW_Lab01)
									_					

- 2. Wybierz Update Code i zaakceptuj zmiany przyciskiem OK.
- 3. Zmodyfikuj kod w funkcji *main*, tak aby odebranie odpowiedniego znaku odpowiadało sterowaniu poszczególnymi diodami LED:
  - a: Red-On z: Red-Off s: Green-On x: Green-Off d: Blue-On
  - c: Blue-Off

10. Parent application - virtual serial port



Zbuduj projekt i zaprogramuj układ.

10. Parent application - virtual serial port

### II. Aplikacja nadrzędna

1. Uruchom środowisko Qt Creator i utwórz nowy projekt Qt Widgets Application:

•	New Project	
Vybierz szablon:		All Templates
Projects Application (Qt) Application (Qt for Python) Library Inne projekty Projekt nieużywający Qt Zaimportuj projekt	Qt Widgets Application         Aplikacja konsolowa Qt         Aplikacja Qt Quick	Creates a widget-based Qt application that contains a Qt Designer-based main window. Preselects a desktop Qt for building the application if available. <b>Supported Platforms:</b> • Desktop • Symulator iOS • Urządzenie iOS • WebAssembly Runtime
		Anuluj Choose

2. Nazwij go LED\_Controller:

🗼 Położenie	Położenie projektu
System budowania	This wizard generates a Qt Widgets Application project. The application derives by default
Szczegóły	from QApplication and includes an empty widget.
Translation	
Zestawy narzędzi	
Podsumowanie	
	Utwárz w: // Jasza/donia//Documents/DDO JEOTS/DraiostoOt
	wybierz
	Ustaw jako domyślne położenie projektów

10. Parent application - virtual serial port

3. Jako system budowania wybierz *qmake*:

		Qt Widgets Application		
Położenie	Zdefiniuj system b	udowania		
款 System budowania	System budowania:	amaka		
Szczegóły	System budowania.	quinte		
Translation				
Zestawy narzędzi				
Podsumowanie				
Anuluj			Wróć Kon	tynuuj
Anuluj			WFOC Kom	lynduj

- 4. W kolejnych oknach pozostaw domyślne ustawienia.
- 5. W oknie wyboru zestawu narzędzi wybierz *Qt 6.3 for macOS* (*MinGW* na Windows):

	Wybór zestawu narzędzi								
System budowania Szczegóły Translation	The following kits can be used for project LED_Controller: Type to filter kits by name Zaznacz wszystkie zestawy narzędzi								
Zestawy narzędzi Podsumowanie	Qt 6.3.0 for iOS	Szczegóły•							
	🔽 📮 Qt 6.3.0 for macOS	Szczegóły•							
	Replacement for "Qt 6.2.3 for iOS"	Szczegóły•							
	Replacement for "Qt 6.2.3 for macOS"	Szczegóły•							
	Replacement for "Qt 6.2.4 for macOS"	Szczegóły•							
Anului									

10. Parent application - virtual serial port

6. Widok struktury wygenerowanego projektu:



7. W pliku projektu *LED\_Controller.pro* dodaj bibliotekę serialport:



8. Przejdź do plik mainwindow.h i zmodyfikuj kod:



10. Parent application - virtual serial port



9. Przejdź do plik *mainwindow.cpp* i zmodyfikuj kod:



10. Parent application - virtual serial port

- 10. Podłącz płytkę z mikrokontrolerem do portu USB.
- 11. Zbuduj projekt i uruchom aplikację. Na pasku statusu powinien pojawić się tekst z nazwą wirtualnego portu szeregowego:



12. Wciśnij reset na płytce mikrokontrolera. W oknie *Komunikaty aplikacji*, w Qt *Creator* powinien pojawić się tekst wysłany przez mikrokontroler:

33	}	
35	connect(&serial &0S	MainWindow
36	connect(&serial, &QS	
37		
38	<pre>rawData.clear();</pre>	
39	}	
40	Mai pWi pdow: Mai pWi pdow(	
42		
43	delete ui;	
44	}	
45		
46 ~	void MainWindow::readDat	
48	i rawData.append(seria	
49 ~	if(rawData.size() >=	
50		
51	rawData= rawData	Device: cu.usbmodem020140202
52		
53	quedug()< <rawbat< td=""><td>4,</td></rawbat<>	4,
Komunil	katy aplikacji 🛛 🛓 🗸 🗸	🕒 🕨 🖕 🌣 🔑 Filtr
LED_	Controller	

17:04:03: Uruchamianie /Users/daniel/Documents/PROJECTS/ProjectsQt/1\_Dydaktyka/build-LED\_Controller-Qt\_6\_3\_0\_for\_mc "LPC804 Start..."

13. Zamknij aplikację.

. . .

14. Przejdź do *Formularze -> mainwindow.ui* i wstaw (przeciągnij) na formatkę *label*:

	🗈 🍃 mainwindow.ui															
	Filtr															
iii	> Layouts	Wp	isz t	tutaj												
Powitanie	> Spacers															
	> Buttons		111			<b>.</b>			÷ 1							
Edvcia	> Item Views (Model-Based)	- 1 - E 🖬	Text	labe	1				Ē -							
	Item Widgets (Item-Based)	- e - e - 📕				• <b>=</b> • • •			•							
	> Containers															
Design	> Input Widgets															
<b></b>	Display widgets															
	Label															
Debug	AI Text Browser															
ىر	‰ Graphics View															
Projekty	12 Calendar Widget															
-	W LCD Number															
?	ECD Number															
Pomoc	Progress Bar															
	Horizontal Line															
	Vertical Line															
	OpenGL Widget					 	 	 		 	 	 	 	 	 	
	QQuickWidget															

10. Parent application - virtual serial port

15. Przejdź do metody *readData* w pliku *mainwindow.cpp* i dodaj kod:



16. Zbuduj i uruchom aplikację a następnie wciśnij reset na płytce z mikrokontrolerem. Odebrany tekst powinien wyświetlić się na widżecie *label*:

• •	MainWindow	
LPC804 Start		
vice: cu usbmodem02	0140202	

17. Zamknij aplikację, przejdź do *Formularze -> mainwindow.ui* i wstaw na formatkę przycisk *PushButton*:



18. Zmień etykietę na *Red* przez podwójne kliknięcie lub we właściwościach, w kolumnie po prawej stronie okna *Qt Creator*:





10. Parent application - virtual serial port

19. Zmień nazwę obiektu *pushButton* na *redButton* oraz ustaw właściwość *checkable* na *true* we właściwościach obiektu po prawej stronie okna Qt *Creator*:



20. Klikając prawym przyciskiem, z menu kontekstowego, wybierz *Przejdź do slotu…* Następnie wybierz sygnał *clicked*:

TextLabel									
1 💼 1 1 1 1 1 💼 1 1 1									
Red	· · · · · · · · · · · · · · · · · · ·								
	Zmień tekst								
. 7 7	Zimen tekstill								
	Zmioń pozwo obiektu								
	Zillen nazwę obiektu								
	Przekształć w								
	Zmień podpowiedź								
	Zmiań "co to iest"								
	Zimen cotojest								
	Zmień arkusz stylu								
	Ograniczenia rozmiaru	Þ					-	1	
	Zastąp								
	Przejdź do slotu								
	Przenieś na spód								
	Przenieś na wierzch								
	Wytnii								
	Skoniui								
	Skopiaj								
	WKIej								
	Zaznacz wszystko								
	Usuń								

QAbstractButton
clicked()
 clicked(bool)
pressed()
released()
toggled(bool)
<ul> <li>QWidget</li> </ul>
customContextMenuRequested(QPoint)
windowlconChanged(Qlcon)
windowlconTextChanged(QString)
windowTitleChanged(QString)
V QObject
destroyed()
objectNameChanged(OString)
object talleentinged (gotting)

21. W pliku *mainwindow.cpp* pojawi się slot (definicja w pliku *mainwindow.h*) on\_redButton\_clicked:



22. Dodaj kod wysyłający dane do mikrokontrolera:

10. Parent application - virtual serial port



23. Zbuduj i uruchom aplikację a następnie wciśnij przycisk *Red*. Czerwona dioda LED powinna świecić gdy przycisk jest wciśnięty i gasnąć gdy jest wyciśnięty:

MainWindow	e e MainWindow
TextLabel	TextLabel
Red	Red
vice: cu.usbmodem020140202	Device: cu.usbmodem020140202

#### III. Zadania

- 1. Dodaj dodatkowe przyciski do sterowania zieloną i niebieską diodą LED.
- 2. Rozmieść widżety w siatce formatki i zwiększ rozmiar czcionki w widżecie *label* na 40:

🕂 🗧 🌠 🚮 🖬 🖬 🗧			Filtr
			label : QLabel
Wpisz tutai			Właściwość
			palette
	· · · · · · • • • • · · · · · · · · · ·		✓ font
			Rodzina
			Wielkość pun
levti ahei			Pogrubiony
ICALLADO			Kursywa
			Podkreślony
3			Przekreślony
Ped	Green	Blue	Kerning
	Gleen	 Diue	Antyaliasing
			cursor
			mouseTracking

3. Dodaj wysyłanie informacji z mikrokontrolera o zapaleniu lub zgaszeniu poszczególnych diod LED i wyświetlanie ich na widżecie *label*:

<ul> <li>MainWindow</li> </ul>	MainWindow
BLUE On	BLUE Off
Red Green Blue	Red Green Blue
Device: cu.usbmodem020140202	Device: cu.usbmodem020140202