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

GRANT NUMBER: 2020-1-PL01-KA226-HE-095653

Declaration

Copyright

Funding Disclaimer

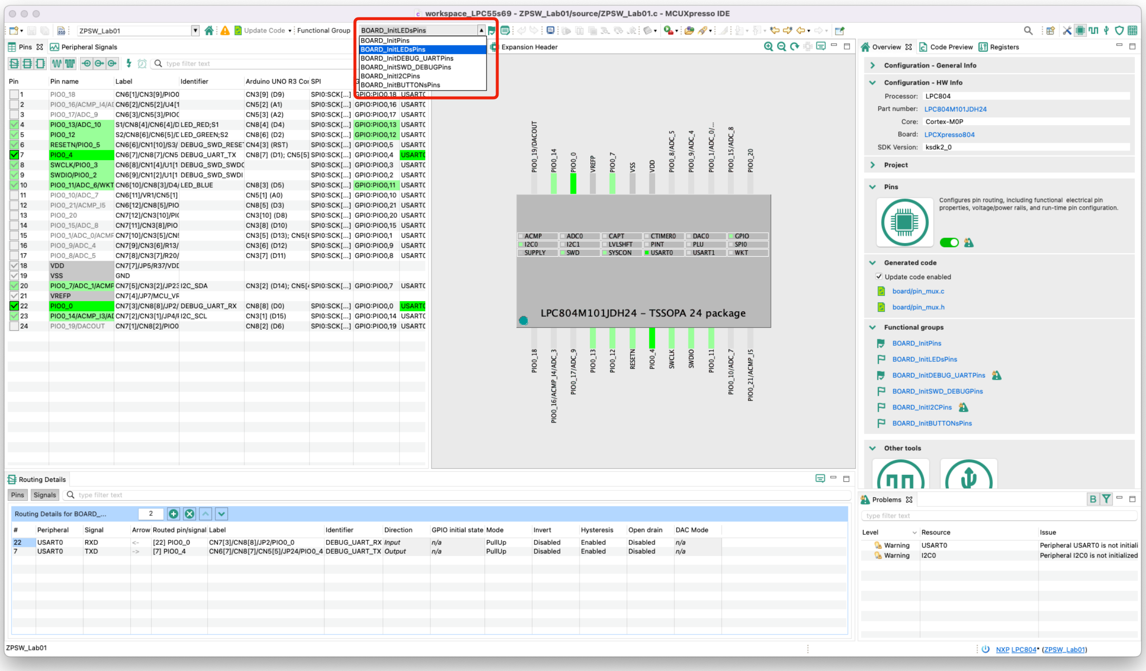
This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Programing of embedded systems

10. Parent application - virtual serial port

I. LED driver

1. You should configure 3 *GPIO* lines to control individual *RGB LEDs*, just like in the first manual. To do this, right-click on the project name and select *MCUXpresso Config Tools -> Open Pins*. From the *Functional Group* menu, select the *BOARD_InitLEDsPins* preset, then activate it by selecting the flag icon on the left:



workspace_LPC55s69 - ZPSW_Lab01/source/ZPSW_Lab01.c - MCUXpresso IDE

ZPSW_Lab01 · Update Code · Functional Group: BOARD_InitLEDsPins

- BOARD_InitPins
- BOARD_InitLEDsPins
- BOARD_InitDEBUG_UARTPins
- BOARD_InitSWD_DEBUGPins
- BOARD_InitI2CPins
- BOARD_InitBUTTONsPins

Pin	Pin name	Label	Identifier	Arduino UNO R3 Co	SPI		
1	PIO0_18	CN6[1]/CN3[9]/PIO0		CN3[9] (D9)	SPI0:SCK[...]	GPIO:PIO0,18	USARTC
2	PIO0_16/ACMP_I4/AI	CN6[2]/CN5[2]/U4[1		CN5[2] (A1)	SPI0:SCK[...]	GPIO:PIO0,16	USARTC
3	PIO0_17/ADC_9	CN6[3]/CN5[3]/PIOC		CN5[3] (A2)	SPI0:SCK[...]	GPIO:PIO0,17	USARTC
✓ 4	PIO0_13/ADC_10	S1/CN8[4]/CN6[4]/D	LED_RED;S1	CN8[4] (D4)	SPI0:SCK[...]	GPIO:PIO0,13	USARTC
✓ 5	PIO0_12	S2/CN8[6]/CN6[5]/D	LED_GREEN;S2	CN8[6] (D2)	SPI0:SCK[...]	GPIO:PIO0,12	USARTC
✓ 6	RESETN/PIO0_5	CN6[6]/CN1[10]/S3/	DEBUG_SWD_RESET	CN4[3] (RST)	SPI0:SCK[...]	GPIO:PIO0,5	USARTC
✓ 7	PIO0_4	CN6[7]/CN8[7]/CN5	DEBUG_UART_TX	CN8[7] (D1); CN5[5]	SPI0:SCK[...]	GPIO:PIO0,4	USARTC
✓ 8	SWCLK/PIO0_3	CN6[8]/CN1[4]/U1[1	DEBUG_SWD_SWDC		SPI0:SCK[...]	GPIO:PIO0,3	USARTC
✓ 9	SWDIO/PIO0_2	CN6[9]/CN1[2]/U1[1	DEBUG_SWD_SWDI		SPI0:SCK[...]	GPIO:PIO0,2	USARTC
✓ 10	PIO0_11/ADC_6/WKT	CN6[10]/CN8[3]/D4/	LED_BLUE	CN8[3] (D5)	SPI0:SCK[...]	GPIO:PIO0,11	USARTC
11	PIO0_10/ADC_7	CN6[11]/VR1/CN5[1]		CN5[1] (A0)	SPI0:SCK[...]	GPIO:PIO0,10	USARTC
12	PIO0_21/ACMP_I5	CN6[12]/CN8[5]/PIO		CN8[5] (D3)	SPI0:SCK[...]	GPIO:PIO0,21	USARTC
13	PIO0_20	CN7[12]/CN3[10]/PIC		CN3[10] (D8)	SPI0:SCK[...]	GPIO:PIO0,20	USARTC
14	PIO0_15/ADC_8	CN7[11]/CN3[8]/PIO		CN3[8] (D10)	SPI0:SCK[...]	GPIO:PIO0,15	USARTC
15	PIO0_1/ADC_0/ACMF	CN7[10]/CN3[5]/CN5		CN3[5] (D13); CN5[6	SPI0:SCK[...]	GPIO:PIO0,1	USARTC
16	PIO0_9/ADC_4	CN7[9]/CN3[6]/R13/		CN3[6] (D12)	SPI0:SCK[...]	GPIO:PIO0,9	USARTC
17	PIO0_8/ADC_5	CN7[8]/CN3[7]/R20/		CN3[7] (D11)	SPI0:SCK[...]	GPIO:PIO0,8	USARTC
✓ 18	VDD	CN7[7]/JP5/R37/VDD					
✓ 19	VSS	GND					
✓ 20	PIO0_7/ADC_1/ACMF	CN7[5]/CN3[2]/JP23	I2C_SDA	CN3[2] (D14); CN5[4	SPI0:SCK[...]	GPIO:PIO0,7	USARTC
✓ 21	VREFP	CN7[4]/JP7/MCU_VF					
✓ 22	PIO0_0	CN7[3]/CN8[8]/JP2/	DEBUG_UART_RX	CN8[8] (D0)	SPI0:SCK[...]	GPIO:PIO0,0	USARTC
✓ 23	PIO0_14/ACMP_I3/AI	CN7[2]/CN3[1]/JP4/F	I2C_SCL	CN3[1] (D15)	SPI0:SCK[...]	GPIO:PIO0,14	USARTC
24	PIO0_19/DACOUT	CN7[1]/CN8[2]/PIO0		CN8[2] (D6)	SPI0:SCK[...]	GPIO:PIO0,19	USARTC

LPC804M101JDH24 – TSSOPA 24 package

Overview · Configuration - General Info · Configuration - HW Info

Processor: LPC804
Part number: LPC804M101JDH24
Core: Cortex-M0P
Board: LPCXpresso804
SDK Version: ksdk2_0

Pins: Configures pin routing, including functional electrical pin properties, voltage/power rails, and run-time pin configuration.

Generated code: ✓ Update code enabled; board/pin_mux.c; board/pin_mux.h

Functional groups: BOARD_InitPins, BOARD_InitLEDsPins, BOARD_InitDEBUG_UARTPins, BOARD_InitSWD_DEBUGPins, BOARD_InitI2CPins, BOARD_InitBUTTONsPins

Routing Details for BOARD_... 2

#	Peripheral	Signal	Arrow	Routed pin/signal	Label	Identifier	Direction	GPIO initial state	Mode	Invert	Hysteresis	Open drain	DAC Mode
22	USART0	RXD	<-	[22] PIO0_0	CN7[3]/CN8[8]/JP2/PIO0_0	DEBUG_UART_RX	*Input*	*n/a*	PullUp	Disabled	Enabled	Disabled	*n/a*
7	USART0	TXD	->	[7] PIO0_4	CN6[7]/CN8[7]/CN5[5]/JP24/PIO0_4	DEBUG_UART_TX	*Output*	*n/a*	PullUp	Disabled	Enabled	Disabled	*n/a*

Problems

Level	Resource	Issue
Warning	USART0	Peripheral USART0 is not initiali
Warning	I2C0	Peripheral I2C0 is not initialized

ZPSW_Lab01 · NXP LPC804* (ZPSW_Lab01)

2. Select *Update Code* and accept the changes with the *OK* button.
3. Modify the code in the main function so that receiving the appropriate character corresponds to the control of individual *LEDs*:

   a: Red-On
   z: Red-Off
   s: Green-On
   x: Green-Off
   d: Blue-On
   c: Blue-Off

```
#include <stdio.h>
#include "board.h"
#include "peripherals.h"
#include "pin_mux.h"
#include "clock_config.h"
#include "LPC804.h"
#include "fsl_debug_console.h"

/*
 * @brief   Application entry point.
 */
int main(void) {

          /* Init board hardware. */
          BOARD_InitBootPins();
          BOARD_InitBootClocks();
          BOARD_InitBootPeripherals();
#ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
          /* Init FSL debug console. */
          BOARD_InitDebugConsole();
#endif

          PRINTF("LPC804 Start...\r\n");
```

```
        char c;

        while(1) {

                c=GETCHAR();

                if( c== 'a') {
                        // On
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_RED_GPIO,
                                        BOARD_INITLEDSPINS_LED_RED_PORT,
                                        BOARD_INITLEDSPINS_LED_RED_PIN,
                                        0);

                }
                if(c == 'z') {
                        // Off
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_RED_GPIO,
                                        BOARD_INITLEDSPINS_LED_RED_PORT,
                                        BOARD_INITLEDSPINS_LED_RED_PIN,
                                        1);
                }
                if( c== 's') {
                        // On
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_GREEN_GPIO,
                                        BOARD_INITLEDSPINS_LED_GREEN_PORT,
                                        BOARD_INITLEDSPINS_LED_GREEN_PIN,
                                        0);

                }
                if(c == 'x') {
                        // Off
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_GREEN_GPIO,
                                        BOARD_INITLEDSPINS_LED_GREEN_PORT,
                                        BOARD_INITLEDSPINS_LED_GREEN_PIN,
                                        1);
                }
                if( c== 'd') {
                        // On
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_BLUE_GPIO,
                                        BOARD_INITLEDSPINS_LED_BLUE_PORT,
                                        BOARD_INITLEDSPINS_LED_BLUE_PIN,
                                        0);

                }
                if(c == 'c') {
                        // Off
                        GPIO_PinWrite(BOARD_INITLEDSPINS_LED_BLUE_GPIO,
                                        BOARD_INITLEDSPINS_LED_BLUE_PORT,
                                        BOARD_INITLEDSPINS_LED_BLUE_PIN,
                                        1);
                }
        }

        return 0 ;
}
```

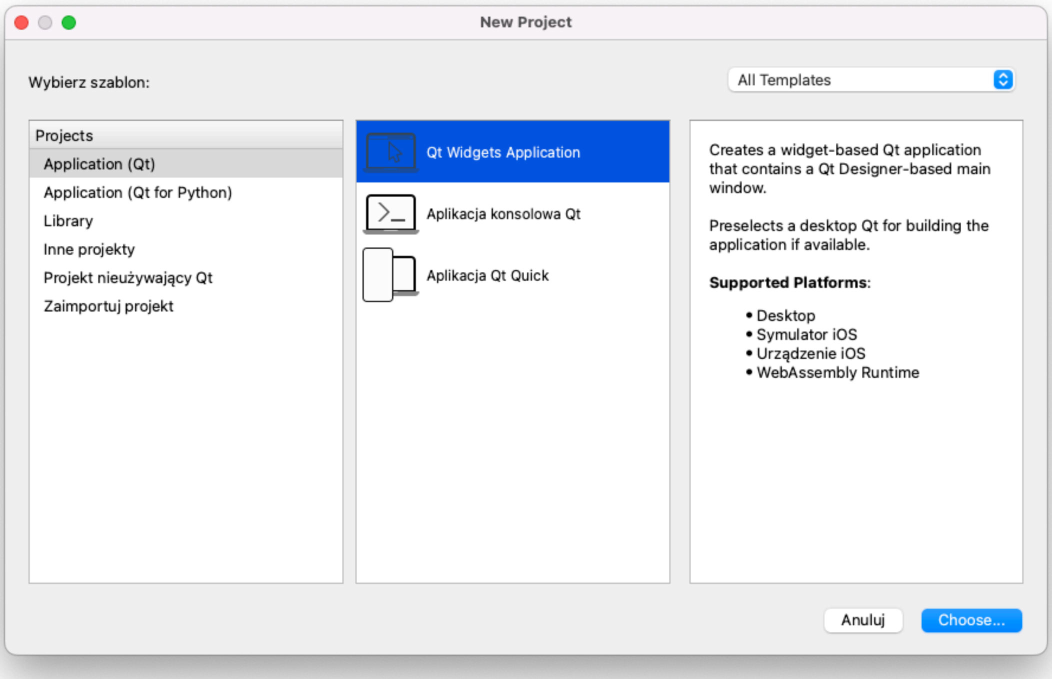
Build a project and program the microcontroller.

Programing of embedded systems

10. Parent application - virtual serial port

II. Parent application

1. Launch *Qt Creator* and create a new *Qt Widgets Application* project:



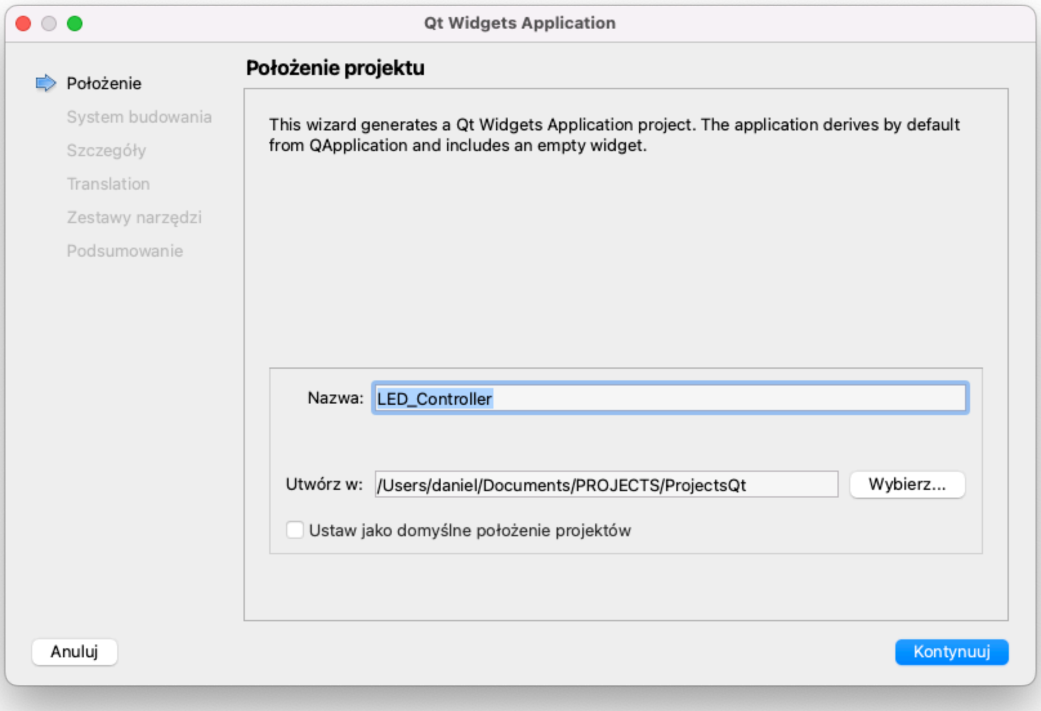
Creates a widget-based Qt application that contains a Qt Designer-based main window.

Preselects a desktop Qt for building the application if available.

Supported Platforms:

- Desktop
- Symulator iOS
- Urządzenie iOS
- WebAssembly Runtime

2. Name it *LED_Controller*:



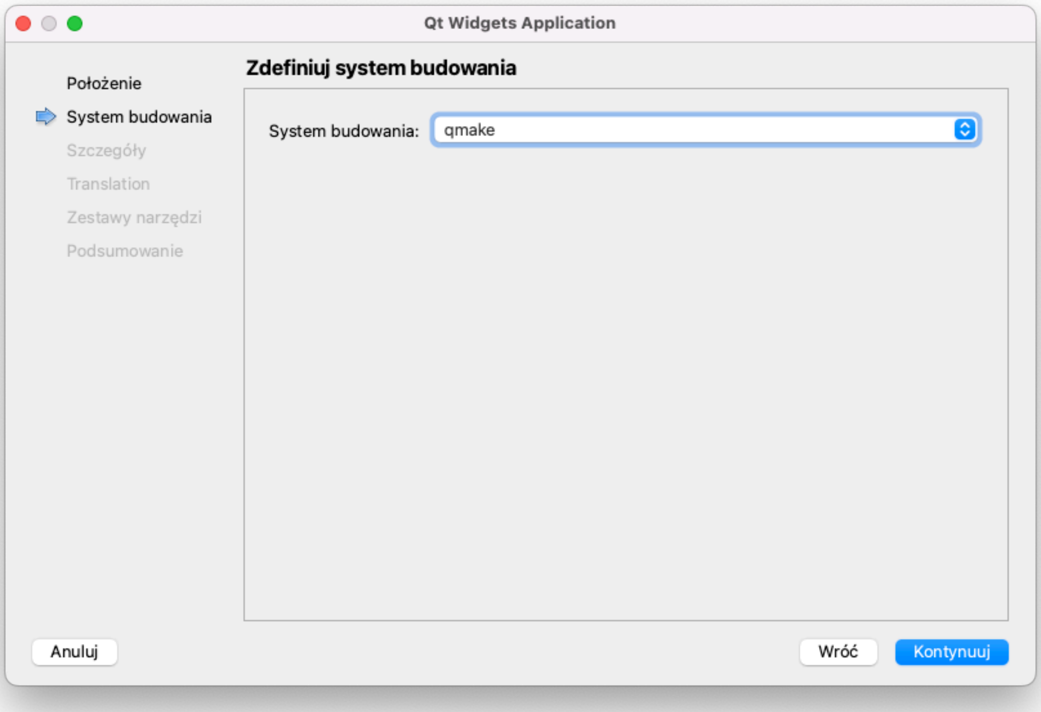
This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.

Nazwa: LED_Controller

Utwórz w: /Users/daniel/Documents/PROJECTS/ProjectsQt

☐ Ustaw jako domyślne położenie projektów

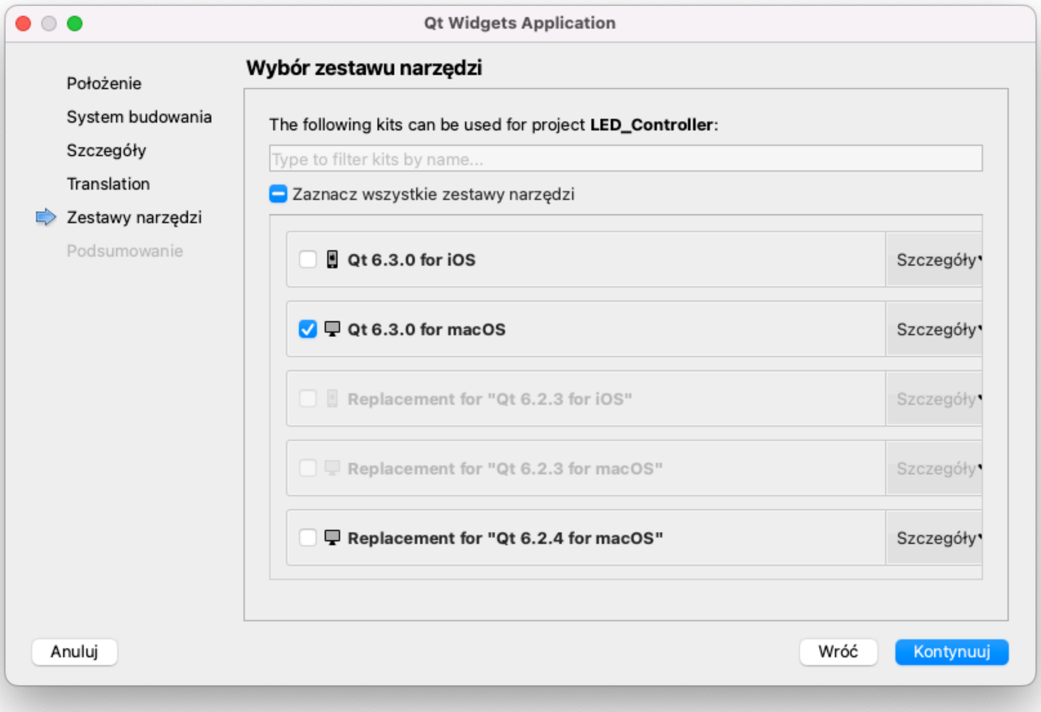
3. Select *qmake* as the build system:



Zdefiniuj system budowania

System budowania: qmake

4. In the following windows, leave the default settings.
5. In the toolkit selection window, select Qt 6.3x for *macOS* (*MinGW* on *Windows*):



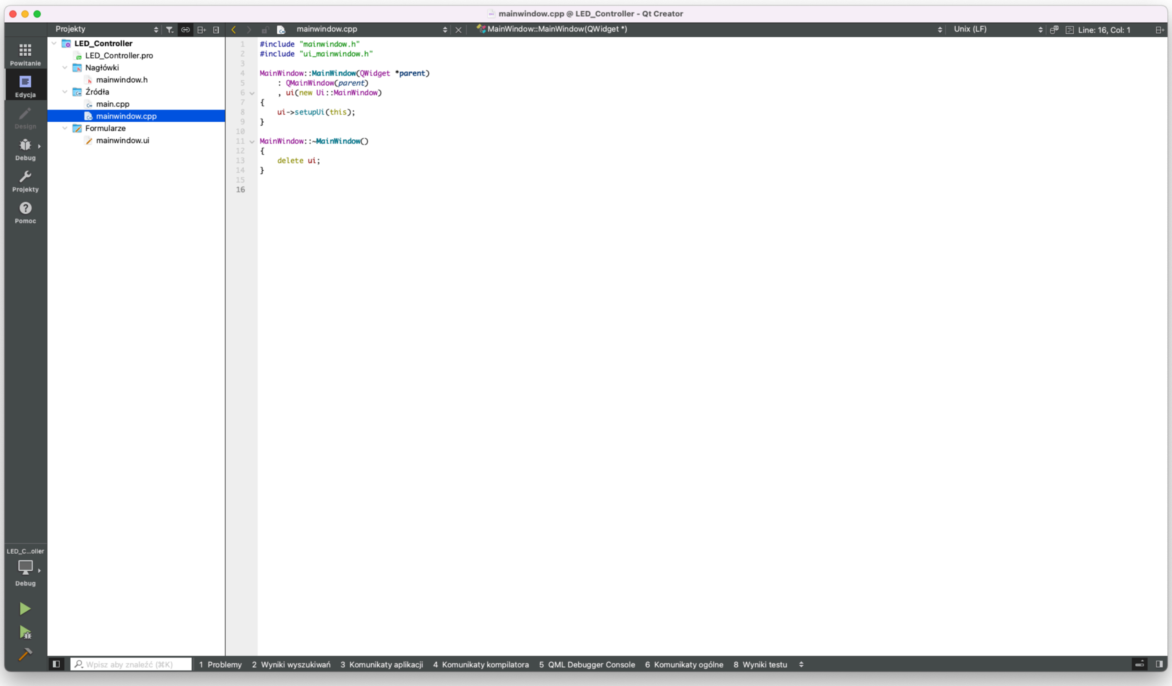
Wybór zestawu narzędzi

The following kits can be used for project **LED_Controller**:

Zaznacz wszystkie zestawy narzędzi

- [ ] Qt 6.3.0 for iOS
- [✓] Qt 6.3.0 for macOS
- [ ] Replacement for "Qt 6.2.3 for iOS"
- [ ] Replacement for "Qt 6.2.3 for macOS"
- [ ] Replacement for "Qt 6.2.4 for macOS"

6. Structure view of the generated project:



```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

```

7. In the LED_Controller.pro project file, add the *serialport* library:

```
QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets serialport

CONFIG += c++17

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs deprecated before Qt 6.0.0

SOURCES += \
    main.cpp \
    mainwindow.cpp

HEADERS += \
    mainwindow.h

FORMS += \
    mainwindow.ui

# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /opt/$${TARGET}/bin
!isEmpty(target.path): INSTALLS += target
```

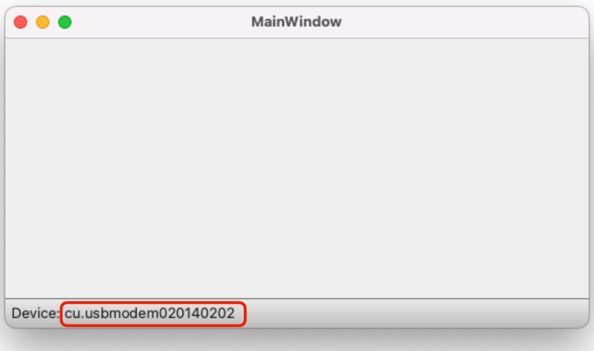
8. Go to the *mainwindow.h* file and modify the code:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QtSerialPort/QSerialPort>
#include <QtSerialPort/QSerialPortInfo>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;
    QSerialPort serial;
    QByteArray cmdData;
    QByteArray rawData;

private slots:
    void readData();
    void error(QSerialPort::SerialPortError error);

};
#endif // MAINWINDOW_H
```

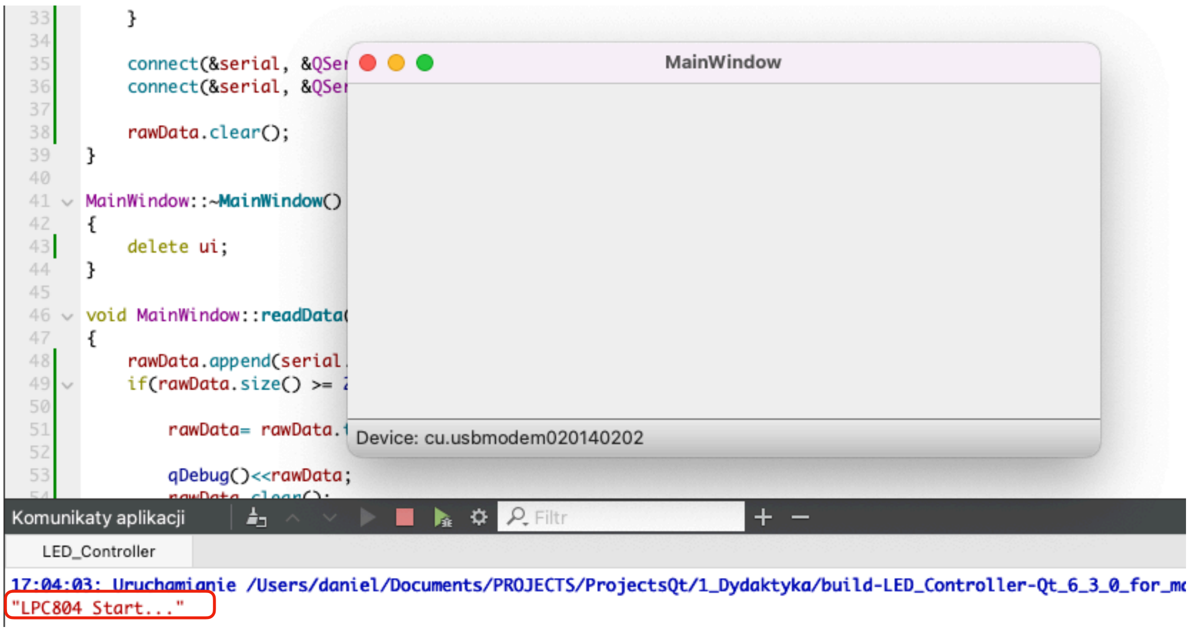
9. Go to the *mainwindow.cpp* file and modify the code:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    ui->statusbar->showMessage("No device");
    QString portname;
    const auto infos = QSerialPortInfo::availablePorts();
    for (const QSerialPortInfo &info : infos) {

        // if(info.portName()=="cu.usbmodem020140202") { // OS Windows e.g "COM1"
        // if(info.description() =="LPC11U3x CMSIS-DAP v1.0.4") {
        if(info.manufacturer() =="NXP Semiconductors") {

            portname=info.portName();
            serial.setPortName(portname);
            if (serial.open(QIODevice::ReadWrite)) {

                ui->statusbar->showMessage(tr("Device: %1").arg(info.portName()));
                serial.setBaudRate( serial.Baud9600,  serial.AllDirections);
                serial.clear();
            } else {

                ui->statusbar->showMessage(tr("Can't open %1, error code
%2") .arg(serial.portName()).arg(serial.error()));
                return;
            }
            break;
        }
    }

    connect(&serial, &QSerialPort::readyRead, this, &MainWindow::readData);
    connect(&serial, &QSerialPort::errorOccurred, this, &MainWindow::error);

    rawData.clear();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::readData()
{
    rawData.append(serial.readAll());
    if(rawData.size() >= 2 && (rawData.last(2)) == "\r\n") {

        rawData= rawData.trimmed();  // delete \r\n

        qDebug()<<rawData;
        rawData.clear();
    }
}

void MainWindow::error(QSerialPort::SerialPortError error)
{
    qDebug()<<error;
    qDebug()<<"---------";
}
```

10. Connect the board with the microcontroller to the *USB* port.
11. Build a project and run the application. The text with the name of the virtual serial port should appear in the *status bar*:



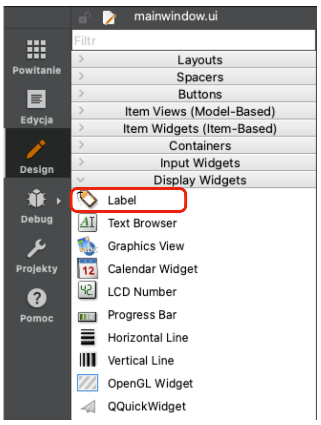
12. Press the reset on the microcontroller board. In the *Application Messages* window, the text sent by the microcontroller should appear in *Qt Creator*:

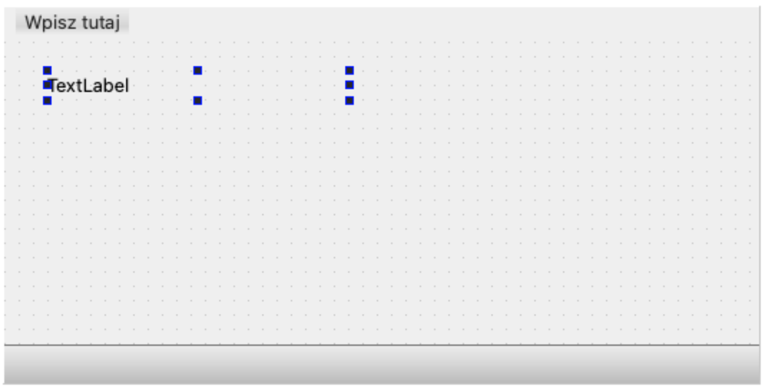


```
    }

    connect(&serial, &QSer
    connect(&serial, &QSer

    rawData.clear();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::readData(
{
    rawData.append(serial.
    if(rawData.size() >= 2

        rawData= rawData.

        qDebug()<<rawData;
```

```
17:04:03: Uruchamianie /Users/daniel/Documents/PROJECTS/ProjectsQt/1_Dydaktyka/build-LED_Controller-Qt_6_3_0_for_mc
"LPC804 Start..."
```

13. Close the application.
14. Go to *Forms* -> *mainwindow.ui* and insert (drag) *label* widget onto the form:

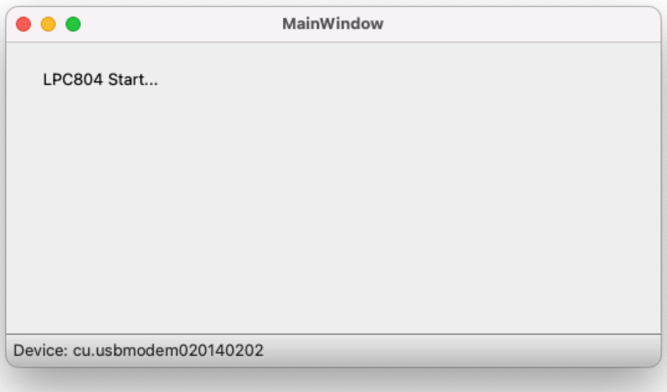




15. Go to the *readData* method in *mainwindow.cpp* and add the code:

```
void MainWindow::readData()
{
    rawData.append(serial.readAll());
    if(rawData.size() >= 2 && (rawData.last(2)) == "\r\n") {

        rawData= rawData.trimmed();  // delete \r\n

        ui->label->setText(rawData);
        qDebug()<<rawData;
        rawData.clear();
    }
}
```

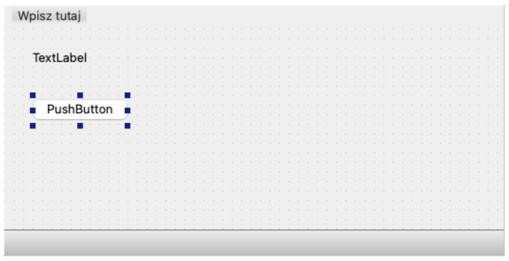
16. Build and run the application and then press the reset button on the microcontroller board. The received text should be displayed on the *label* widget:



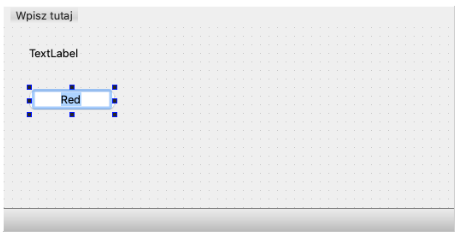
```
LPC804 Start...
```

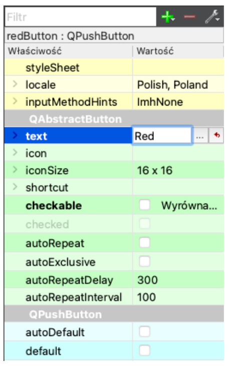
```
Device: cu.usbmodem020140202
```

17. Close the application, go to *Forms* -> *mainwindow.ui* and insert the *PushButton* button on the form:



18. Change the *label* to *Red* by double-clicking or in the properties, column on the right side of the *Qt Creator window*:

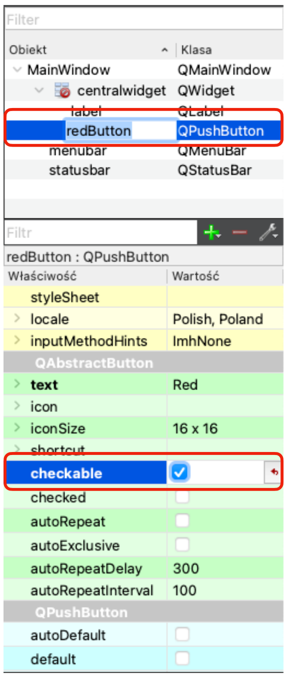




redButton : QPushButton

Właściwość	Wartość
styleSheet	
locale	Polish, Poland
inputMethodHints	ImhNone
QAbstractButton	
text	Red
icon	
iconSize	16 x 16
shortcut	
checkable	☐ Wyrówna...
checked	☐
autoRepeat	☐
autoExclusive	☐
autoRepeatDelay	300
autoRepeatInterval	100
QPushButton	
autoDefault	☐
default	☐

19. Rename the *pushButton* object to *redButton* and set the checkable property to true in the object properties on the right side of the *Qt Creator window*:

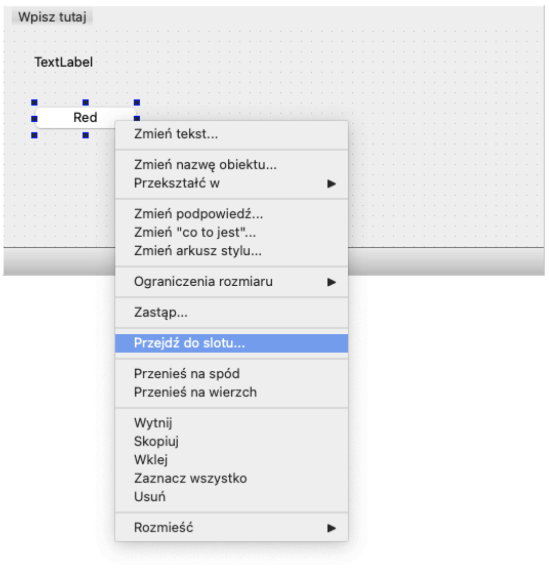


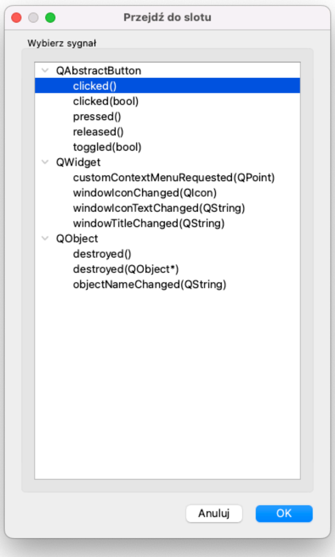
Obiekt	Klasa
MainWindow	QMainWindow
centralwidget	QWidget
label	QLabel
redButton	QPushButton
menubar	QMenuBar
statusbar	QStatusBar

redButton : QPushButton

Właściwość	Wartość
styleSheet	
locale	Polish, Poland
inputMethodHints	ImhNone
QAbstractButton	
text	Red
icon	
iconSize	16 x 16
shortcut	
checkable	✓
checked	
autoRepeat	
autoExclusive	
autoRepeatDelay	300
autoRepeatInterval	100
QPushButton	
autoDefault	
default	

20. By right-clicking, from the context menu, select *Go to Slot...* Then select the clicked signal:




Przejdź do slotu

Wybierz sygnał

```
QAbstractButton
    clicked()
    clicked(bool)
    pressed()
    released()
    toggled(bool)
QWidget
    customContextMenuRequested(QPoint)
    windowIconChanged(QIcon)
    windowIconTextChanged(QString)
    windowTitleChanged(QString)
QObject
    destroyed()
    destroyed(QObject*)
    objectNameChanged(QString)
```

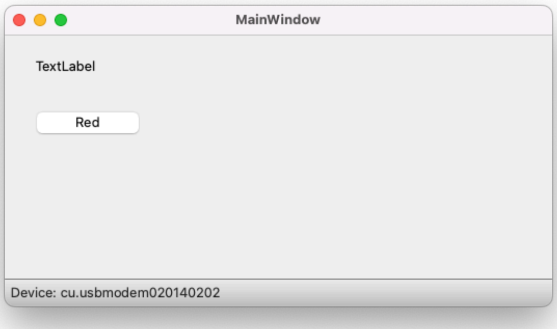
21. A slot will appear in the *mainwindow.cpp* file (definition in the *mainwindow.h* file) *on_redButton_clicked*:

```
void MainWindow::on_redButton_clicked()
{

}
```

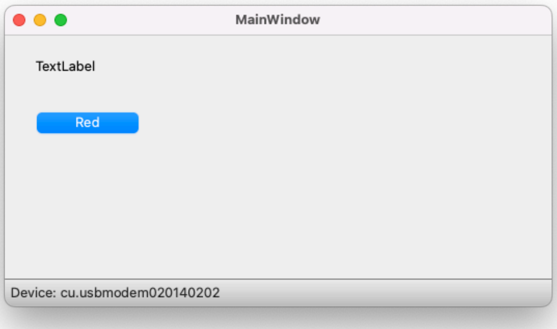
22. Add code that sends data to the microcontroller:

```
void MainWindow::on_redButton_clicked()
{
    cmdData.clear();
    cmdData.append(ui->redButton->isChecked() ? 'a' : 'z');
    serial.write(cmdData);
}
```

23. Build and run the application and then press the *Red* button. The red *LED* should glow when the button is pushed in and go out when the button is pushed out:



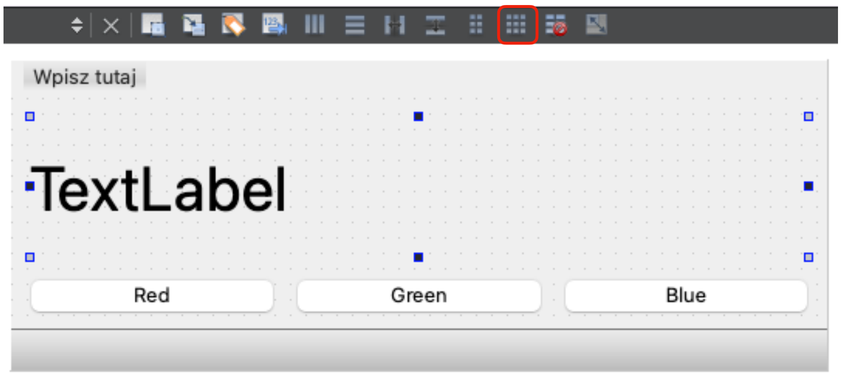
Device: cu.usbmodem020140202

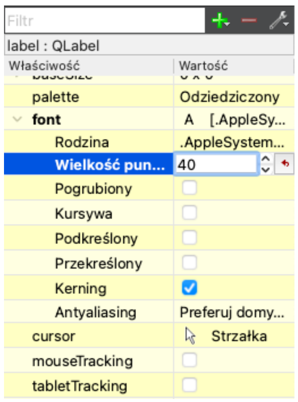


Device: cu.usbmodem020140202

III. Exercises

1. Add extra buttons to control the *green* and *blue LEDs*.
2. Arrange widgets in the form grid and increase the *font size* of the *label* widget to 40:



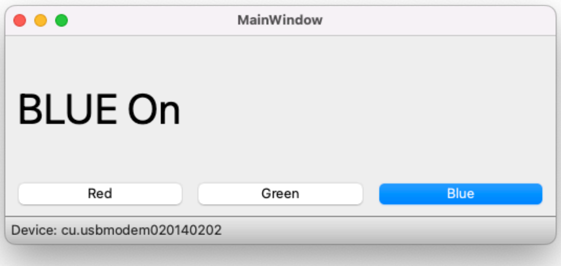


Filtr

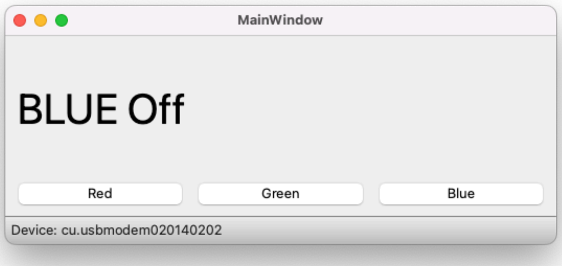
label : QLabel

Właściwość	Wartość
palette	Odziedziczony
font	A [.AppleSy...
Rodzina	.AppleSystem...
Wielkość pun...	40
Pogrubiony	☐
Kursywa	☐
Podkreślony	☐
Przekreślony	☐
Kerning	✓
Antyaliasing	Preferuj domy...
cursor	Strzałka
mouseTracking	☐
tabletTracking	☐

3. Add sending information from the microcontroller about turning *on* or *off* individual *LEDs* and displaying them on the *label* widget:



Device: cu.usbmodem020140202



Device: cu.usbmodem020140202