

Teaching online electronics, microcontrollers and programming in Higher Education

Programing of embedded systems

5. Termometr cyfrowy I2C

Lead Partner: Warsaw University of Technology

Authors: Daniel Krol

University of Applied Sciences in Tarnow



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 Where other published and unpublished source materials have been used, these have been acknowledged.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

I. Konfiguracja interfejsu I2C

- 1. Stwórz nowy projekt dla płyty *LPCXpresso804*:
- 2. Nazwij projekt np. ZPSW_Lab06 i dodaj sterownik I2C:

| Config | igure the project | | | | | | | | |
|--|--|--|---|---|-------------|--|--|--|--|
| ect name: | ZPSW_Lab6 | | Project name suffix: Search | | | | | | |
| Use default | It location | | | | | | | | |
| ation: /Us | sers/daniel/Documents/MCU) | (pressolDE_11.5.0_7232/workspace_LPC55s69/ZP | SW_Lab6 | | Browse. | | | | |
| uise Deele | | Deced | Parad Parinet Tura Parinet Online | | | | | | |
| | ages | Doard | Project Type | Project Options | | | | | |
| LPC804MI | 1101JDH24 | Empty board files | C Project C++ Project | CMSIS-Core | a UARI | | | | |
| LPC804M1 | IIIIIJDH20 | 0-14 | C Static Library C++ Static Library | Copy sources | | | | | |
| LPC804M1 | 1111JDH24 | | | Import other files | | | | | |
| moonente | | | | Components selection summary | | | | | |
| d or romou | va SDK software components | | | oomponente serection summary | - | | | | |
| perating Sv | vetame Drivere CMSIS Dr | vere I Itilities Middleware Board Components A | hetraction Laver Software Componente | type to filter | | | | | |
| seruting by | Jacenta Britera Conolo Br | vers otilities middleware board components A | baraction cayer bortware components | Name | Versio Infe | | | | |
| | | | | Name Description | | | | | |
| Drivers | | | 2 💥 🖽 🖻 | > E Drivers | | | | | |
| Drivers type to fil | | | | Drivers Operating Systems | | | | | |
| Drivers type to fil Name | ilter | Description | Version Info | Drivers Operating Systems Utilities | | | | | |
| Drivers type to fil Name | ilter | Description | Version Info | Borivers Gerating Systems Utilities | | | | | |
| Drivers type to fil Name | ilter | Description ADC Driver CAPT Driver | Version Info 2.5.0 2.10 | B Drivers B Operating Systems B Utilities | | | | | |
| Drivers type to fil Name | ilter adc capt clock | Description ADC Driver CAPT Driver Clock Driver | Version Info 2.5.0 2.1.0 | → Drivers → Operating Systems → Utilities | | | | | |
| Drivers | adc capt clock common | Description ADC Driver CAPT Driver Clock Driver COMMON Driver | Version Info 2.5.0 2.3.1 2.3.1 | Second Systems Second S | | | | | |
| Drivers type to fil Name O O O O O O O O O O O O O O O O O O O | ilter adc capt clock common citimer | Description ADC Driver CAPT Driver Clock Driver COMMON Driver CTIMEr Driver | Version Info 2.5.0 2.3.1 2.3.1 2.2.1 | Venine Solution So | | | | | |
| Drivers type to fil Name O O O O O O O O O O O O O O O O O O O | ilter eapt copt clock common ctimer dae | Description ADC Driver CAPT Driver Clock Driver COMMON Driver CTimer Driver DAC Driver | Version Info 2.5.0 2.3.1 2.3.1 2.2.1 2.0.2 | Venine Description | | | | | |
| Drivers type to fil Name | ilter eact capt clock common ctimer dac galo | Description ADC Driver CAPT Driver COMMON Driver CTIMer Driver DAC Driver GPID Driver | Version Info 2.5.0 2.1.0 2.3.1 2.3.1 2.2.1 2.2.1 2.0.2 2.1.7 | teane Description | | | | | |
| Drivers type to fil Name O O O O O O O O O O O O O O O O O O O | ilter adc icapt iclock icommon ictimer idac gpio i2c | Description ADC Driver CAPT Driver Clock Driver CTimer Driver DRC Driver GPIO Driver IZC Driver | Version Info 2.5.0 2.3.1 2.3.1 2.2.1 2.0.2 2.1.7 2.1.0 2.1.0 2.1.0 | Vescription | | | | | |
| Drivers type to fil Name | ilter adc capt clock common ctimer dac gpio 12c iap | Description ADC Driver CAPT Driver COMMON Driver CTIMEr Driver DAC Driver GPIO Driver I2D Driver I4D Driver | Version Info 2.5.0 2.1.0 2.3.1 2.3.1 2.2.1 2.0.2 2.1.7 2.0.2 2.1.7 2.0.6 | teane (Jesc-Ipuol ≥ Drivers ≥ Operating Systems ≥ Utilities | | | | | |
| Drivers type to fil Name | ilter adc capt .clock common ctimer .dac .gpio .l2c .iap .iap .iap | Description ADC Driver CAPT Driver COMMON Driver CTimer Driver DAC Driver GPID Driver IZD Driver IAD Driver IAD Driver | Version Info 2.5.0 2.1.0 2.3.1 2.3.1 2.2.1 2.0.2 2.1.7 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.0 2.0.1 2.0.0 2.0.1 | Venine Description | | | | | |
| Drivers Type to fill Name Control Cont | ilter ade capt clock common ctimer dae gelo 12e 12e 14p 16p 16p 16p 16p | Description ADC Driver CAPT Driver COMMON Driver CTIMER Driver GPIO Driver IZD Driver IZD Driver IQD Driver IQD Driver IQCON Driver IQCON Driver | Version Info 2.5.0 2.1.0 2.3.1 2.2.1 2.2.1 2.1.7 2.1.7 2.0.6 2.0.1 2.1.0 2.0.1 2.1.0 | teane (Jescipuol ≥ Drivers ≥ Operating Systems ≥ Utilities | | | | | |
| Drivers Type to fil Name Control Contr | ilter adc capt common ctimer dac gelio lize lize liap inon hpc_scomp hpc_sc | Description ACC Driver CAPT Driver COMMON Driver CTIMET Driver DRC Driver RCD Driver RCD Driver ICD Driver ICD Driver ICCON Driver LPC, ACOMP Driver CRC Driver | Version Info 2.5.0 2.1.0 2.3.1 2.3.1 2.3.1 2.2.1 2.0.2 2.1.7 2.1.0 2.0.6 2.0.1 2.0.1 2.0.1 2.1.1 | teane Sources | | | | | |
| Drivers type to fill Name Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q | ilter ade ceapt clock common citimer dae gelo li2e hipe_acom hipe_acom hipe_acom hipe_acom | Description ADC Driver CAPT Driver COMMON Driver CTimer Driver DAC Driver GPID Driver I2D Driver I2D Driver I2D Driver I2D Driver I2C CACMP Driver CRC Driver MRT Driver | Version Info 2.5.0 2.1.0 2.3.1 2.2.1 2.2.1 2.0.2 2.1.7 2.0.6 2.0.6 2.0.1 2.1.0 2.1.1 2.0.4 | teane (Jeschuson) ≥ Drivers ≥ Operating Systems ≥ Utilities | | | | | |
| Drivers type to fil Name G G G G G G G G G G G G G | ilter adc capt capt common celock common celock common celock common celock common celoc com foce com | Description ADC Driver CAPT Driver COMMON Driver CTIMER Driver DAC Driver I2C Driver I2C Driver I2C Driver I2C Driver I2C Driver CRC Driver CRC Driver MRT Driver PINT Driver | Version Info 25.0 21.0 2.31 2.31 2.21 2.02 21.0 20.8 20.1 2.10 2.0 2.10 2.0.1 2.10 2.0.2 2.11 2.0.1 2.10 2.0.3 2.0.1 2.0.1 2.11 2.0.4 2.0.4 2.0.9 9 | teane perception ≥ Operating Systems ≥ Utilities | | | | | |
| Drivers | ilter vade ceapt clock common citimer dae gelo vade gelo vade hep_acom hep | Description ADC Driver CAPT Driver COMMON Driver CTimer Driver DAC Driver GPID Driver IAD Driver IAD Driver ICCON Driver ICCON Driver CRC Driver MRT Driver PINT Driver PINT Driver PINT Driver | Version Info 2.5.0 2.1.0 2.3.1 2.3.1 2.2.1 2.0.2 2.1.7 2.0.6 2.0.6 2.0.1 2.1.1 2.0.4 2.1.1 2.0.4 2.1.1 2.0.4 2.1.9 2.2.1 | teame ≥ Drivers ≥ Operating Systems ≥ Utilities | | | | | |

3. Przejdź do Config Tools -> Open Pins. Z menu Functional Group wybierz preset BOARD_Initl2CPins, i aktywuj go przez zaznaczenie ikony flagi po lewej stronie:

| | | | | | | 6 | workspace_LP | C55s69 - ZPSW | _Lab03 | _1/source/ | ZPSW_LabC | 03.c - MC | UXpresso I | IDE | | | | |
|-----------|--------------------------|--------------------|------------------|----------------------|---------------|------------------|----------------------|-----------------|--------|------------|------------|-----------|------------|----------|---------------|--------------------------------|--------------------------------|------------------------------------|
| 📑 🖬 🖬 🛙 | 🗟 👼 ZPSW_Lal | 6 | 💌 🔺 🔒 💈 | Update Code 🔹 Fun | ctional Group | BOARD_Initi2CPi | \$ | 💽 💌 💷 💜 🛸 | 0 | > 10 16 3 | N (N (R (| <u>م</u> | ر • 🔗 😂 | | - 🐎 🗳 🗇 - 🗇 🛃 | | | ፍ 📴 🗙 🖲 🗤 🕴 🖸 🔛 |
| 📰 Pins 😂 | Peripheral Signa | ls | | | | | 🗆 🏮 Package | 🕄 🏮 Expansion H | Header | | | | | | ଷ୍ ପ୍ 🕈 💿 📼 🖻 | A Overview 🔀 | Code Preview 🔢 Registers | - 8 |
| 990 | }₩₩-●●• | 🗲 💈 🔯 🔍 type | | | | | | | | | | | | | | > Configurat | tion - General Info | |
| Pin | Pin name | Label | Identifier | Arduino UNO R3 Cor | SPI | GPIO USA | т | | | | | | | | | V Configurat | tion - HW Info | |
| 1 | PI00_18 | CN6[1]/CN3[9]/PIOC | 5 | CN3[9] (D9) | SPI0:SCK[] | GPIO:PIO0,18 USA | ITC | | | | | | | | | Processor | LPC804 | |
| 2 | PI00_16/ACMP_14/A | CN6[2]/CN5[2]/U4[| 1 | CN5[2] (A1) | SPI0:SCK[| GPIO:PIO0,16 USA | ITC | | | | | | | | | Part number: | L PC804M101 IDH24 | |
| 3 | PI00_17/ADC_9 | CN6[3]/CN5[3]/PIOI | ¢ | CN5[3] (A2) | SPI0:SCK[| GPIO:PIO0,17 USA | ITC | | | | | | | | | Core | Contex-MOP | |
| 4 | PI00_13/ADC_10 | S1/CN8[4]/CN6[4]/C | LED_RED;S1 | CN8[4] (D4) | SPI0:SCK[| GPIO:PIO0,13 USA | ITC | 10 | | | | 10 | 4 | 60 | | Core. | Contex-more | |
| 6 | PIDU_12 RESETNIDIOD 5 | S2/CN8[6]/CN6[5]/C | DEBLIG SWD RESE | CN8[6] (D2) | SPID:SCK[| GPIO:PIOU,12 USA | rrc rrc | - Nor | | | | × | 8 8 | QQ QQ | | Board: | LPCXpressoa04 | |
| 7 | PI00_4/ADC_11 | CN6[7]/CN8[7]/CN5 | DEBUG_UART_TX | CN8[7] (D1); CN5[5] | SPI0:SCK[| GPIO:PIO0,4 USA | ITC | /61 | 7 | 9 | 5 | 8/A | A/6 | 20 22 | | SDK version: | KSGK2_U | |
| 8 | SWCLK/PIO0_3 | CN6[8]/CN1[4]/U1[1 | DEBUG_SWD_SWD | c | SPI0:SCK[| GPIO:PIO0,3 USA | ITC | 8 | 8 | 00 492 | 8 <u>0</u> | 8 8 | 0 0 | 8 8 | | > Project | | |
| 9 | SWDIO/PIO0_2 | CN6[9]/CN1[2]/U1[1 | DEBUG_SWD_SWD | | SPI0:SCK[] | GPIO:PIO0,2 USA | ITC | • | 1 | 1 | î î | 2 6 | ~ ~ | e e | | | | |
| 10 | PI00_11/ADC_6/WK | CN6[10]/CN8[3]/D4 | / LED_BLUE | CN8[3] (D5) | SPI0:SCK[] | GPIO:PIO0,11 USA | ITC | | | | | | | | | Pins | | |
| 11 | PI00_10/ADC_7 | CN6[11]/VR1/CN5[1] | | CN5[1] (A0) | SPI0:SCK[| GPIO:PIO0,10 USA | TC | | | | | | | | | | Configures pin routing, includ | ing functional electrical pin |
| 12 | PIDD_21(ACMP_I5 | CN6[12]/CN8[5]/PIC | | CN8[5] (03) | SPID:SCK[| GPI0:PI00,21 USA | TTC | | | | | | | | | | properties, voltage/power rail | s, and run-time pin configuration. |
| 14 | PIO0_15/ADC_8 | CN7[11]/CN3[8]/PIO | | CN3[8] (D10) | SPI0:SCK[| GPI0:PI00.15 USA | TC | | | | | | | | | | | |
| 15 | PIO0_1/ADC_0/ACM | CN7[10]/CN3[5]/CN | ŧ | CN3[5] (D13); CN5[| SPID:SCK[| GPIO:PIO0,1 USA | ITC | ACM | IP I | ADCO | CAPT | CTIMERO | DACO | GPIO | | | | |
| 16 | PI00_9/ADC_4 | CN7[9]/CN3[6]/R13 | (| CN3[6] (D12) | SPI0:SCK[| GPIO:PIO0,9 USA | ITC | I2C0 | | 12C1 | LVLSHFT | PINT | PLU | SPIO | | | C 4 | |
| 17 | PI00_8/ADC_5 | CN7[8]/CN3[7]/R20 | / | CN3[7] (D11) | SPI0:SCK[] | GPIO:PIO0,8 USA | ITC | SUPP | PLY | SWD | SYSCON | USARTO | USART1 | WKT | | | | |
| 18 | VDD | CN7[7]/JP5/R37/VDI | C | | | | | | | | | | | | | Generated | code | |
| ✓ 19 | VSS | GND | | 010101 (014) 0101 | CONCONT | 00000007 | | | | | | | | | | Update cor | de enabled | |
| 20 | VIDEED | CN7[5]/CN3[2]/JP2. | E CONTRACTOR | CNS[2] (D14); CNS[- | SPIU-SUK[| GPI0.PI00,7 USK | iii. | | | | | | | | | board/pi | n_mux.c | |
| 22 | PIO0_0/ACMP_I1 | CN7[3]/CN8[8]/JP2 | DEBUG UART RX | CN8[8] (D0) | SPI0:SCK[] | GPIO:PIO0.0 USA | ITC | | | | | | | | | board/oi | n mux h | |
| 23 | PI00_14 | CN7[2]/CN3[1]/JP4/ | 12C_SCL | CN3[1] (D15) | SPI0:SCK[| GPIO:PIO0,14 USA | TC | | LPC8 | 04M101 | JDH24 – | TSSOP/ | 24 pac | kage | | | | |
| 24 | PIO0_19/DACOUT | CN7[1]/CN8[2]/PIOC |) | CN8[2] (D6) | SPI0:SCK[| GPIO:PIO0,19 USA | ITC | - | | | | | | | | Functional | groups | |
| | | | | | | | | | | | | | | | | BOARD | InitDine | |
| | | | | | | | | 8 | 5 | 6 | IN IS | 7 3 | 0 1 | 5 5 | | 10 00000 | | |
| | | | | | | | | 8 | ğ | 8 8 I | 8 8 | NC NO | ow o | Q N | | P BOARD_ | InitLEDsPins | |
| | | | | | | | | E | 14/ | 121 | £ " | | E | ,10/. | | BOARD_ | InitDEBUG_UARTPins | |
| | | | | | | | | | 8 | 0 | | | | 5 0 | | EI noven | | |
| | | | | | | | | | 6/M | e. | | | | P10 | | La posono- | initswo_bebooPins | |
| | | | | | | | | | 5 | | | | | | | BOARD_ | Initi2CPins 😩 | |
| | | | | | | | | | Old | | | | | | | P BOARD_ | InitBUTTONsPins | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | V Other tool | | |
| | | | | | | | | | | | | | | | | | - | |
| | | | | | | | | | | | | | | | | | | |
| Routing | Details | | | | | | | | | | | | | | 🖃 = 🗆 | (ПП | ነ (ወ ነ | |
| Pins Sic | nals Q type filter | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | Problems 🕅 | | BY |
| Routing | Details for BOARD_In | 2 🖯 | | | | | | | | | | | | | | | | |
| # P | eripheral Signal | Arrow Routed | pin/signal Label | | Identifie | r Direction | GPIO initial state M | /lode Inver | 1 | Hysteresis | Open drain | n DAC N | fode | | | Level ~ | Resource | Issue |
| 23 12 | CO SCL | - [23] PIC | 0_14 CN7[2]/CN | I3[1]/JP4/PIO0_14 | I2C_SCI | Not Specified | n/a F | PullUp Disat | bled | Enabled | Disabled | n/a | | | | Sk Warning | USARTO | Peripheral USART0 is not initiali |
| 20 12 | CO SDA | - [20] PIC | 00_7 CN7[5]/CN | 13[2]/JP23/CN5[4]/PI | 00_7 12C_SD | Not Specified | n/a F | PullUp Disat | bled | Enabled | Disabled | n/a | | | | S Warning | 12C0 | Peripheral I2C0 is not initialized |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| ZPSW_Lab6 | | | | | | | | | | | | | | | | | U NXP LPC | 804* (ZPSW_Lab6) |
| - | | | | | | | | | | | | | | | | | | |

4. Przejdź do zakładki *Clocks* a następnie kliknij dwukrotnie bloczek *FRO_OSC* i zmień częstotliwość generatora *FRO_OSC* na 30 MHz:



5. Następnie kliknij dwukrotnie bloczek *I2C0CLKSEL* i wybierz *main_clk* (15 MHz):



6. Przejdź do zakładki *Peripherals,* wybierz *Peripheral drivers* i zaznacz *I2C* z listy sterowników:

| 000 | | | | c worksp | ice_LPC55s69 - ZP | SW_Lab02/source/ZPSW_Lab0 | 12.c - MCUXpresso IDE | | | | | |
|--------------------------------------|-------|--------------------------------|---|--------------------------|--------------------------|-----------------------------------|---|-----------------------------------|--|------------------------------|--|--|
| 📑 🖬 🐘 🔜 ZPSW_Lab02 | • 🔺 🔺 | 🚺 Update Code 🔹 | Functional Group | BOARD_InitPeripherals | • 🖻 😅 🥠 | 9 💼 N 💷 (b 🖷 x, 4 | 3 M 🚳 🖓 🖌 🎒 🛷 🖌 🗄 | 월 - 종 · 영 수 · 수 · <u>명</u> | | ዲ 🗄 🗶 🖷 🗤 የ 🗘 🖩 | | |
| Components 😵 🦞 Peripherals | - 0 | A Overview 🕅 | Code Preview | | | | | | | - 0 | | |
| | 0 | > Configura | ation - General Info | | | | | | | | | |
| | | V Configura | | | | | | | | | | |
| Peripheral drivers (Device specific) | | Processor | : LPC804 | | | | | | | | | |
| | | Part number | LPC804M101JDH24 | | | | | | | | | |
| Custom initialization | 0 | Core | Cortex-M0P | | | | | | | | | |
| | | Board | LPCXpresso804 | | | | | | | | | |
| | | SDK Version | : KSGK2_0 | | | | | | | | | |
| | | > Project | | | | | | | | | | |
| | | Periphera | ls | | | | | | | | | |
| | | | Configures the in | vitializ 😑 💿 🔵 | Sele | ct configuration component | | | | | | |
| | | (U | | Select which component | ents should be offered | Present in the tool-chain project | | | | | | |
| | | | | type filter text | | | | | | | | |
| | | | | Configuration component | ent Component descript | ion | Category | | | | | |
| | | Generated | d code | GPIO | General Purpose I/O | (GPIO) | Peripheral drivers (Device s | | | | | |
| | | Update co | ode enabled | 12C | Inter-Integrated Circ | uit (I2C) | C) Peripheral drivers (Device s | | | | | |
| | | board/p | Doard/perpheratis.c USART Universal Synchronous/Receiver/Transmitter (USART) Peripheral drivers (Device (| | | | | | | | | |
| | | Doard/p | enprierais.n | | | | | | | | | |
| | | Functional | al groups | | | | | | | | | |
| | | BOARD, | BOARD_InitPeripherals 🔒 | | | | | | | | | |
| | | Other tool | 45 | | | | | | | | | |
| | | \sim | Deen SDK components manager Cancel 0K | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | וחי) (|) | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | A Problems 32 | | | | | | | | BT | | |
| | | | - | | | 1 | | - | | | | |
| | | Level ~ | Resource | Issue Decisioneral II | CADTO is not initialized | Origin | Target Desinherels: ROADD, InitDesinherels | Type | | | | |
| | | 🔈 Warning | 1200 | Peripheral I2 | .C0 is not initialized | Pins:BOARD_InitI2CPins | Peripherals: BOARD_InitPeripherals | Validation | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| ZPSW_Lab02 | | | | | | | | | | () NXP PC804* (ZPSW Lab02) | | |
| - | | | | | | | | • | | | | |

7. Wybierz interfejs *I2C0* i zmień domyślą prędkość transmisji na 400 000 bps:

| Ĉ | 2 0201 £ | | | | | | | | | |
|---|---|--|--------|--|--|----------|-------|--|--|--|
| 1 | Inter-Integrated Circuit (I2C) (Peripheral drivers (Device specific)) | | | | | | | | | |
| N | ame I2C0 | | | | | Custom n | ame 🗌 | | | |
| м | Mode Polling V Peripheral 12C0 | | | | | | | | | |
| ~ | ✓ I2C general configuration Preset Custo | | | | | | ¥ | | | |
| | I2C mode | Master n | node | | | | ¥ | | | |
| | Clock source | I2C0 clock - BOARD_BootClockFRO18M: 15 MHz, BOARD_BootClockFRO24M: Inactive, BOARD_BootClockFRO30M: Inactive | | | | | | | | |
| Clock source frequency 15 MHz (BOARD_BootClockFR018M) | | | | | | | ¥ | | | |
| | ✓ Master configuration | | | | | | | | | |
| | Enable master mode | | | | | | | | | |
| | Baud rate in bits per second | d | 400000 | | | | | | | |
| | Enable internal timeout function | | | | | | | | | |

8. Kliknij Update Code w celu wygenerowania konfiguracji interfejsu I2C0.

II. Wyświetlacz graficzny

1. Dodaj do projektu (przeciągnij pliki do workspace) bibliotekę wyświetlacza OLED:



2. Przejdź do głównego pliku projektu i zmodyfikuj kod jak poniżej:



3. Podłącz wyświetlacz do płytki prototypowej według poniższego schematu:



4. Zbuduj projekt, zaprogramuj układ i sprawdź działanie sterownika wyświetlacza.

III. Biblioteka termometru LM75B

- 1. Utwórz pliki biblioteki termometru. W tym celu kliknij prawym klawiszem myszy na folderze *source* w *workspace* a następnie wybierz *New->Header File* i nazwij go *LM75B.h*.
- 2. Analogicznie, klikając prawym klawiszem myszy na folderze source w workspace, wybierz New->Source File i nazwij go LM75B.c :



3. Przejdź do pliku *LM75.h* i dopisz kod:

```
#ifndef LM75B_H_
#define LM75B_H_
#include "fsl_i2c.h"
#define LM75_REG_TEMP (0x00) // Temperature Register
#define LM75_REG_CONF (0x01) // Configuration Register
#define LM75_ADDR (0x48) // LM75 address
void LM75B_Init(I2C_Type *base);
float LM75B_Read();
#endif /* LM75B_H_ */
```

4. Przejdź do pliku *LM75B.c* i dopisz kod:

```
#include "LM75B.h"
static I2C_Type *I2C_base=NULL;
void LM75B_Init(I2C_Type *base) {
    I2C_base=base;
    char data_write[2];
    data_write[0] = LM75_REG_CONF;
    data_write[1] = 0x02;
    if (kStatus_Success == I2C_MasterStart(I2C_base, LM75_ADDR, kI2C_Write)) {
        I2C_MasterWriteBlocking(I2C_base, &data_write[0], 2, kI2C_TransferDefaultFlag);
        I2C_MasterStop(I2C_base);
    }
float LM75B_Read() {
    char data_read[2];
    char data_write[1];
    float temp;
    int16_t v;
    data_write[0] = LM75_REG_TEMP;
```

| | if (kStat | <pre>tus_Success == I2C_MasterStart(I2C_base, LM75_ADDR, kI2C_Write)) {</pre> |
|---|-----------------------|---|
| | | <pre>I2C_MasterWriteBlocking(I2C_base, &data_write[0], 1, kI2C_TransferNoStopFlag);</pre> |
| | | I2C_MasterRepeatedStart(I2C_base, LM75_ADDR, <i>kI2C_Read</i>); I2C_MasterReadBlocking(I2C_base, &data_read[0], 2, <i>kI2C_TransferDefaultFlag</i>); |
| | } | I2C_MasterStop(I2C_base); |
| | v= (data_ temp = v | <pre>_read[0] << 8) data_read[1]; / 256.0; // temperature value in Celsius</pre> |
| } | return te | emp; |

 Przejdź do ustawień projektu. Klikając prawym klawiszem myszy na nazwie projektu wybierz Properties a następnie Settings -> Preprocessor. Zmień flagę PRINTF_FLOAT_ENABLE na 1 oraz usuń flagę CR_INTEGER_PRINTF:



6. Przejdź do głównego pliku programu i zmodyfikuj kod:



7. Zbuduj projekt, zaprogramuj układ i sprawdź odczyt temperatury (w celu zmiany temperatury można delikatnie dotknąć układu *LM75B*).

IV. Proste GUI

1. Dodaj sytuację wyświetlacza 7-segmentowego dla wyświetlania temperatury:

```
#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"
#include "lefsl_debug_console.h"
#include "LM75B.h"
 char sbuff[32];
 float temp;
  * @brief Application entry point.
 int main(void) {
                 /* Init board hardware. */
BOARD_InitBootPins();
BOARD_InitBootClocks();
BOARD_InitBootPeripherals();
 #ifndef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
                               FSL debug conso
                 BOARD_InitDebugConsole();
#endif
                  /* Initialize OLED */
                 OLED_Init(I2C0_PERIPHERAL);
                 /* Initialize LM75 */
LM75B_Init(I2C0_PERIPHERAL);
                 while(1) {
                                   temp = LM75B Read():
                                  OLED_Clear_Screen(0);
                                  OLED_7segf(0, 4, temp, 4, 1, 1);
OLED_Puts(105, 1, "C");
                                   OLED_Refresh_Gram();
                  }
                  return 0 ;
```

2. Zbuduj projekt, zaprogramuj układ i sprawdź odczyt temperatury.

3. Dodaj bargraf:

```
#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"
#include "oled.h"
#include "LM75B.h"
char sbuff[32];
float temp;
#define T_MIN
                          0
#define T_MAX
                          40
void Bargraph(uint8_t x, uint8_t y, uint8_t w, uint8_t h, float min, float max, float v) {
             if(v<min) {</pre>
                           v=min;
             if(v>max) {
                          v=max:
             v = ((v-min)*w)/(max-min);
             OLED_Draw_Rect(x , y, x+w-1, y+h-1, 1);
OLED_Draw_Fill_Rect(x+2, y+2, x+v-3 , y+h-3, 1);
}
 * @brief Application entry point.
int main(void) {
* Init FSL debug conso
             BOARD_InitDebugConsole();
#endif
                Initialize OLED */
             OLED_Init(I2C0_PERIPHERAL);
             /* Initialize LM75 */
LM75B_Init(I2C0_PERIPHERAL);
             while(1) {
                          temp = LM75B_Read();
                          OLED_Clear_Screen(0);
                          OLED_7segf(0, 4, temp, 4, 1, 1);
OLED_Puts(105, 1, "C");
                          Bargraph(0, 45, 128, 8, T_MIN, T_MAX, temp);
                          sprintf(sbuff, "%d", T_MIN);
OLED_Puts(0, 7, sbuff);
sprintf(sbuff, "%3d", T_MAX);
OLED_Puts(110, 7, sbuff);
                          OLED_Refresh_Gram();
             }
             return 0 ;
```

4. Zbuduj projekt, zaprogramuj układ i sprawdź odczyt temperatury.

V. Zadania

- 1. Sprawdź wskazania barografu dla różnych zakresów *T_MIN* i *T_MAX*.
- 2. Zaimplementuj filtr (*moving average filter*), uśredniający określoną liczbę pomiarów, dany równaniem:

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k) \, dla \, n = 0, \, 1, \, 2, \, 3, \, \dots$$

w funkcji FilterAVG:

```
#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"
#include "lefsl_debug_console.h"
 char sbuff[32];
float temp:
#define T_MIN
#define T_MAX
#define N 16
                                0
                                40
 float FilterAVG(float x) {
 void Bargraph(uint8_t x, uint8_t y, uint8_t w, uint8_t h, float min, float max, float v) {
                if(v<min) {</pre>
                                v=min;
                if(v>max) {
                                v=max;
                v = ((v-min)*w)/(max-min);
                OLED_Draw_Rect(x , y, x+w-1, y+h-1, 1);
OLED_Draw_Fill_Rect(x+2, y+2, x+v-3 , y+h-3, 1);
  * @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 cons
                BOARD_InitDebugConsole();
#endif
                   Initialize OLED :
                OLED_Init(I2C0_PERIPHERAL);
                /* Initialize LM75 */
LM75B_Init(I2C0_PERIPHERAL);
                while(1) {
                                temp = LM75B_Read();
temp = FilterAVG(temp);
                                OLED_Clear_Screen(0);
                               OLED_7segf(0, 4, temp, 4, 1, 1);
OLED_Puts(105, 1, "C");
                                Bargraph(0, 45, 128, 8, T_MIN, T_MAX, temp);
                               sprintf(sbuff, "%d", T_MIN);
OLED_Puts(0, 7, sbuff);
sprintf(sbuff, "%3d", T_MAX);
OLED_Puts(110, 7, sbuff);
                                OLED_Refresh_Gram();
                }
                return 0 :
```