ENGINE

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

Erasmus+

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

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 Where other published and unpublished source materials have been used, these have been acknowledged.

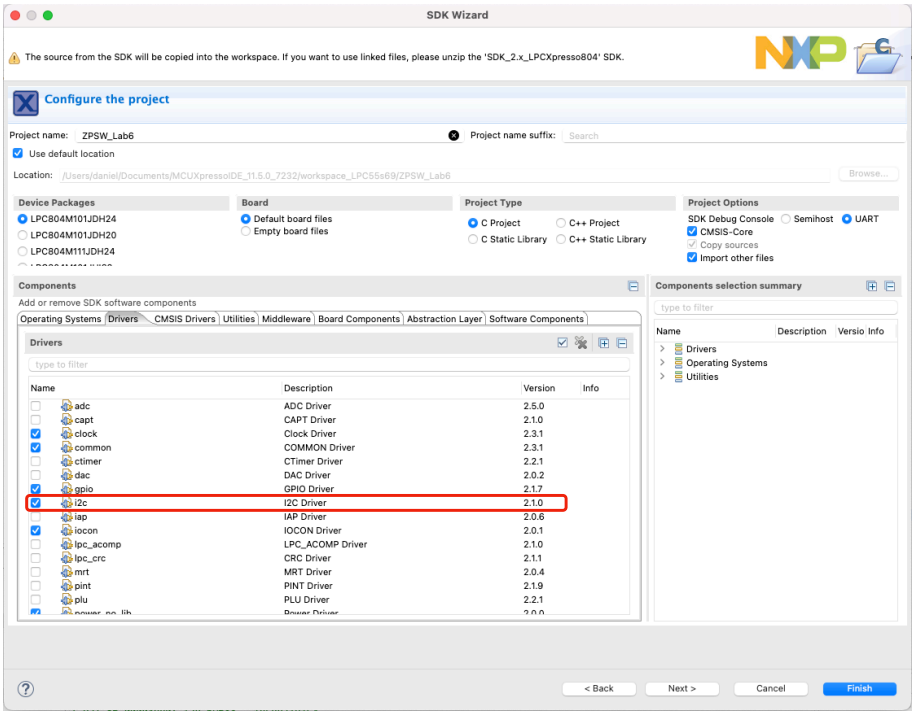
Copyright

Funding Disclaimer

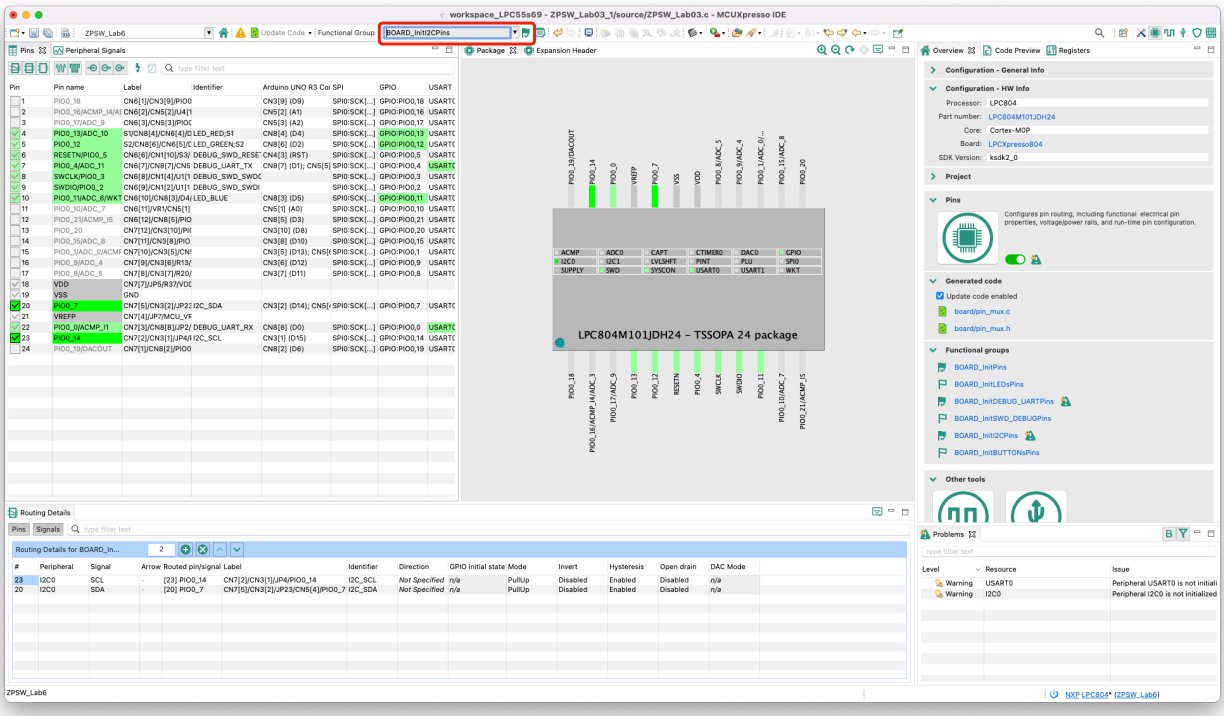
This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

I. Konfiguracja interfejsu I2C

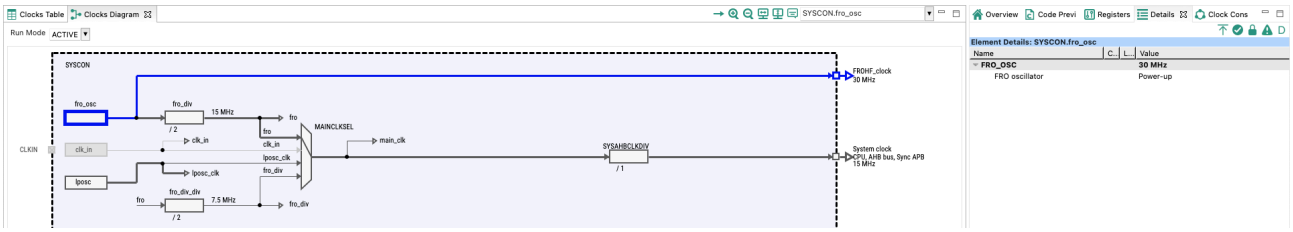
1. Stwórz nowy projekt dla płyty *LPCXpresso804*:
2. Nazwij projekt np. *ZPSW_Lab06* i dodaj sterownik *I2C*:



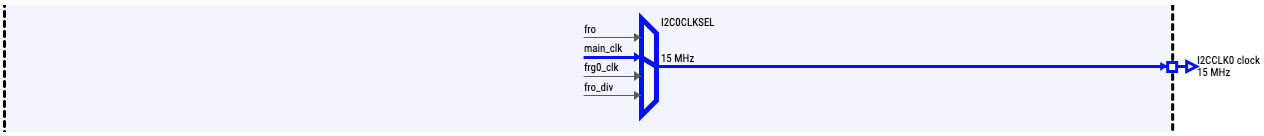
3. Przejdź do *Config Tools -> Open Pins*. Z menu *Functional Group* wybierz preset *BOARD_InitI2CPins*, i aktywuj go przez zaznaczenie ikony flagi po lewej stronie:



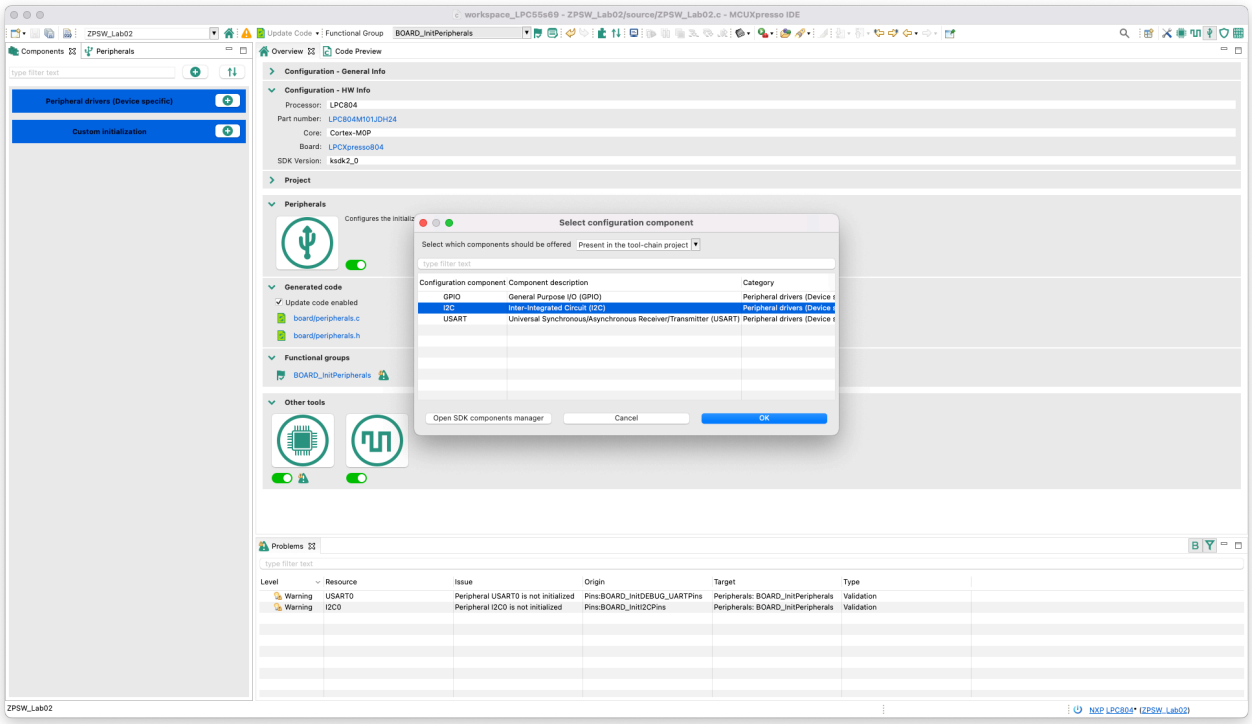
4. Przejdź do zakładki *Clocks* a następnie kliknij dwukrotnie bloczek *FRO_OSC* i zmień częstotliwość generatora *FRO_OSC* na 30 MHz:



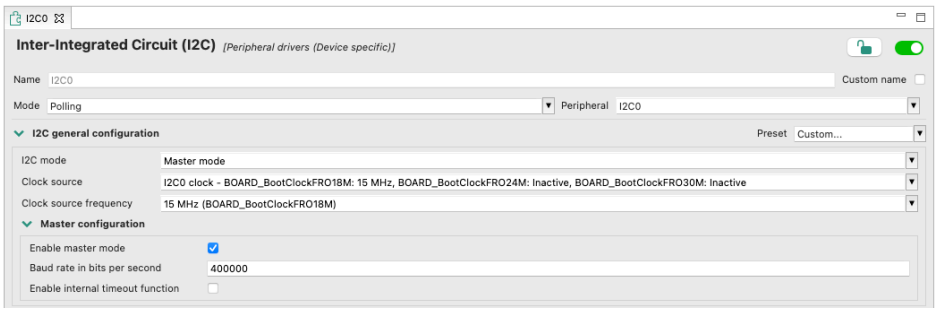
5. Następnie kliknij dwukrotnie bloczek *I2C0CLKSEL* i wybierz *main_clk* (15 MHz):



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



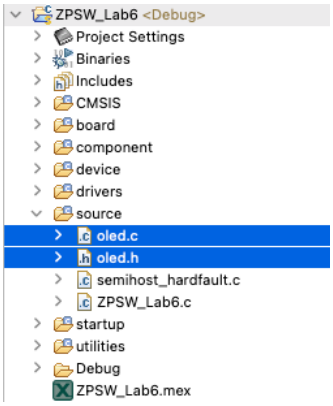
7. Wybierz interfejs *I2C0* i zmień domyślą prędkość transmisji na 400 000 bps:



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

II. Wyświetlacz graficzny

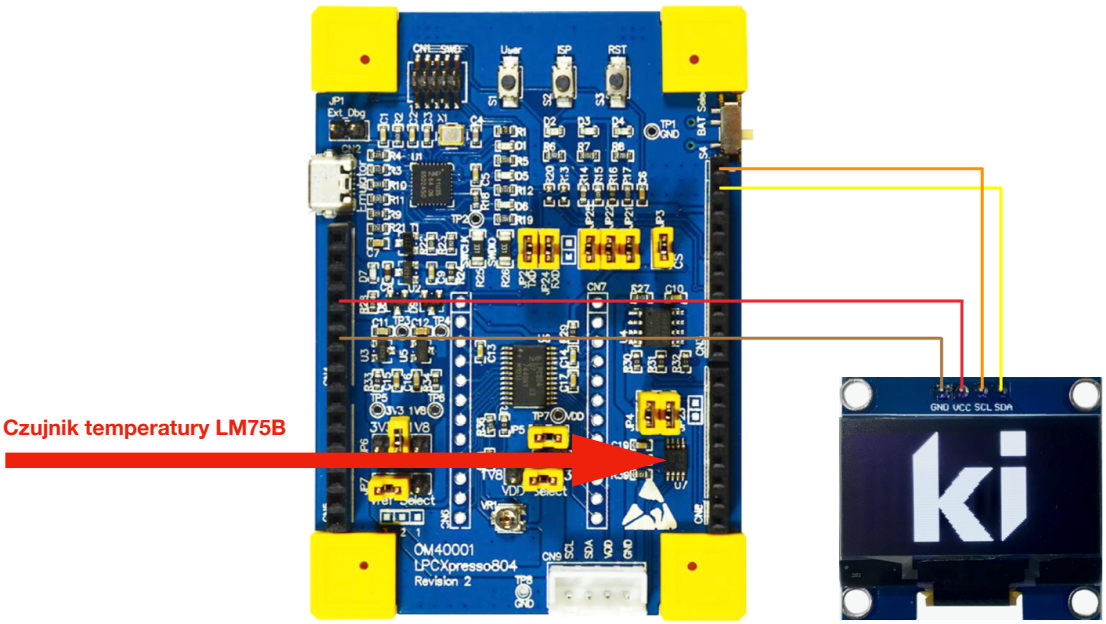
1. Dodaj do projektu (przeciągnij pliki do workspace) bibliotekę wyświetlacza OLED:



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

```
#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"
/*
 * @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
          /* Initialize OLED */
          OLED_Init(I2C0_PERIPHERAL);
          OLED_Draw_Bitmap(LogoKI);
          OLED_Refresh_Gram();

          while(1) {

          }
          return 0 ;
}
```

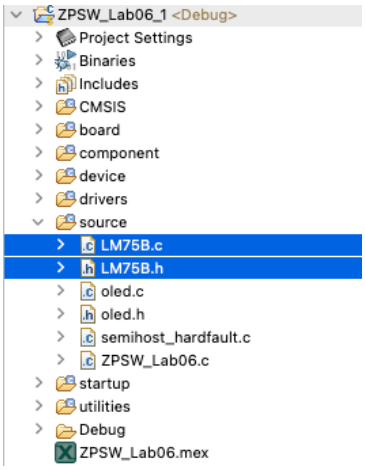
3. Podłącz wyświetlacz do płytki prototypowej według poniższego schematu:



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:

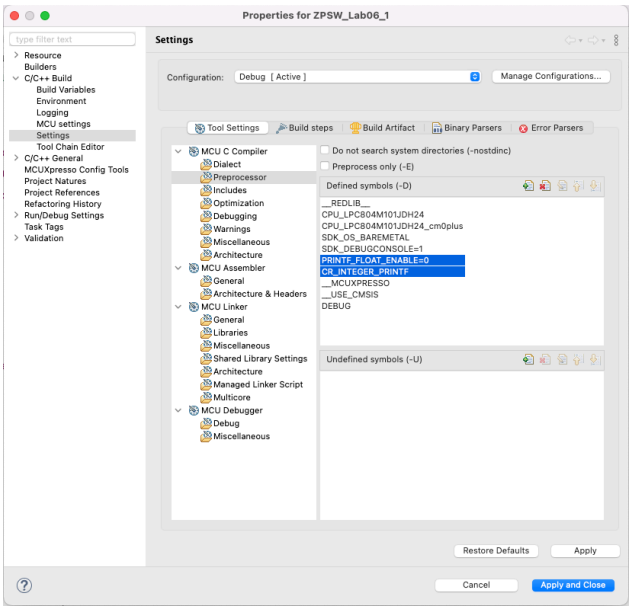
```
#ifndef LM75B_H_
#define LM75B_H_

#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_ */
```

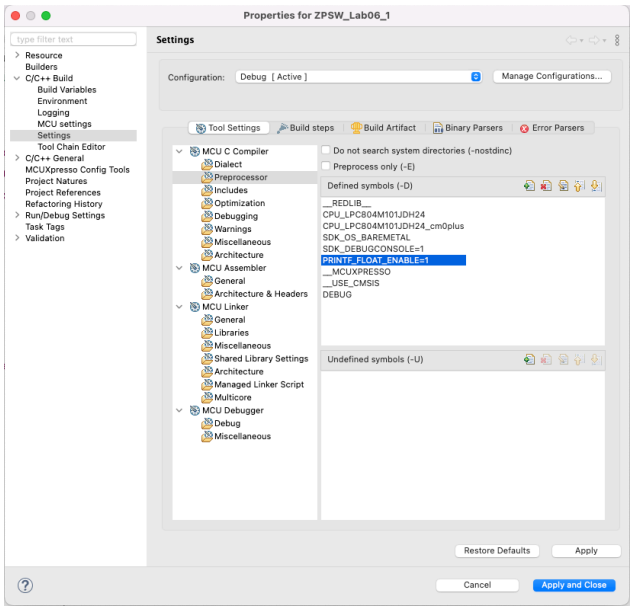
4. Przejdź do pliku *LM75B.c* i dopisz kod:

```
#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 (kStatus_Success == I2C_MasterStart(I2C_base, LM75_ADDR, kI2C_Write)) {

                I2C_MasterWriteBlocking(I2C_base, &data_write[0], 1, kI2C_TransferNoStopFlag);

                I2C_MasterRepeatedStart(I2C_base, LM75_ADDR, kI2C_Read);
                I2C_MasterReadBlocking(I2C_base, &data_read[0], 2, kI2C_TransferDefaultFlag);

                I2C_MasterStop(I2C_base);
        }

        v= (data_read[0] << 8) | data_read[1];
        temp = v / 256.0; // temperature value in Celsius

        return temp;
}
```

5. 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:

```
#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;

/*
 * @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

        /* Initialize OLED */
        OLED_Init(I2C0_PERIPHERAL);
        OLED_Draw_Bitmap(LogoKI);
        OLED_Refresh_Gram();

        /* Initialize LM75 */
```

```
          LM75B_Init(I2C0_PERIPHERAL);

          while(1) {

                    OLED_Clear_Screen(0);

                    temp = LM75B_Read();

                    sprintf(sbuff, "t: %.3f C", temp);
                    OLED_Puts(0,0, sbuff);

                    OLED_Refresh_Gram();
          }
          return 0 ;
}
```

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 "oled.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
          /* Init FSL debug console. */
          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) {
                    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 console. */
          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:

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

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 "oled.h"
#include "LM75B.h"

char sbuff[32];
float temp;

#define T_MIN       0
#define T_MAX       40
#define N 16

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) {
                    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 console. */
          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 ;
}
```