

Podstawy techniki mikroprocesorowej 2

Lab 4 - Wyświetlacz LCD z układem HD44780

Imię i Nazwisko:	Remigiusz Mielcarz, Grzegorz Salzburg
Nr indeksu:	252887, 252912

Termin zajęć: dzień tygodnia, godzina:	Środa 14:10-17:10 TP
Numer grupy ćwiczeniowej:	Y03-45f
Data wykonania ćwiczenia:	08.12.2021
Termin do oddania sprawozdania:	19.01.2022
Prowadzący kurs:	Dr inż. Krzysztof Halawa

Spis treści

1	Cel ćwiczenia	1
2	Zadania do wykonania	2
3	Schemat układu	2
4	Schemat i konfiguracja pinów mikrokontrolera	3
5	Kod programu	3
5.1	main.c	3
5.2	lcd.c	7
5.3	main.h	12
5.4	lcd.h	13

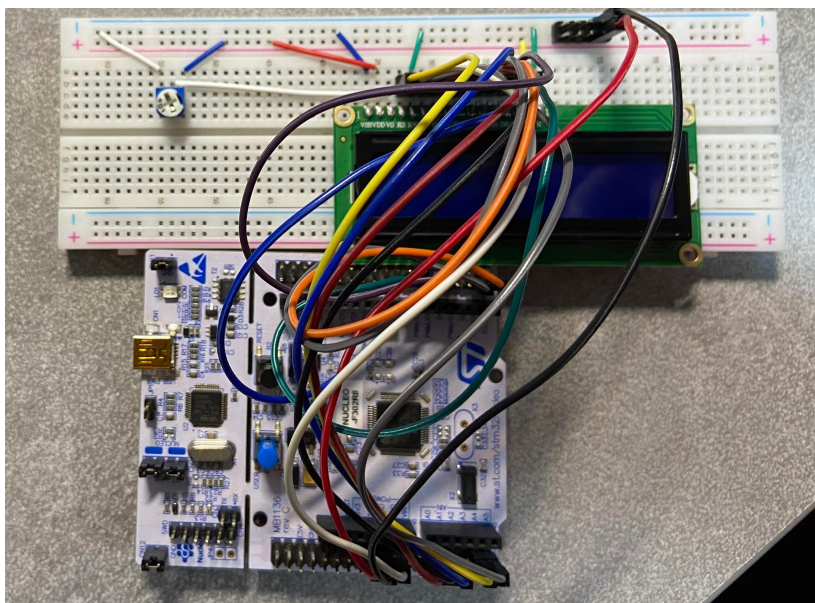
1 Cel ćwiczenia

Głównym celem ćwiczenia jest zapoznanie się z działaniem i budową wyświetlacza LCD opartego na układzie HD44780 przy użyciu Mikrokontrolera STM32f302.

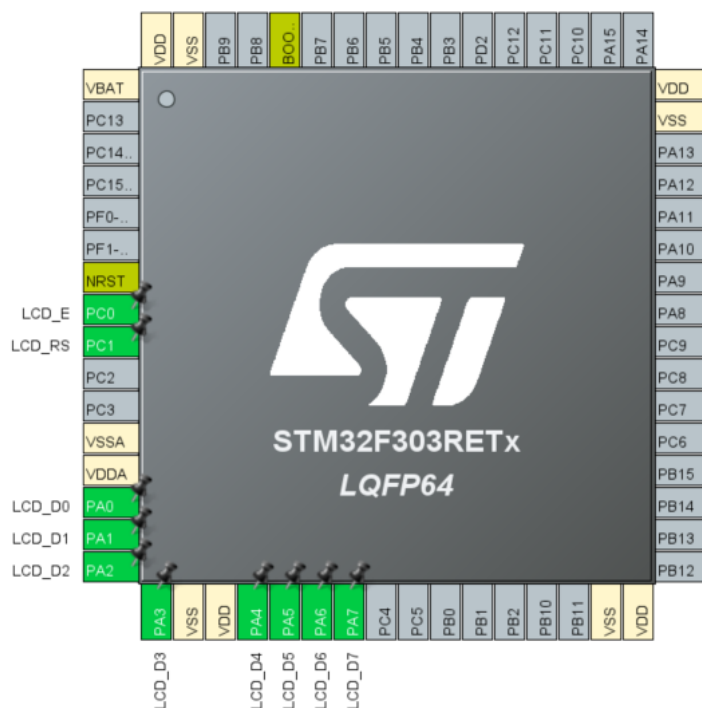
2 Zadania do wykonania

- Implementacja funkcji inicjalizacji wyświetlacza
- Zdefiniowanie i wyświetlenie 2 znaków na ekranie wyświetlacza

3 Schemat układu



4 Schemat i konfiguracja pinów mikrokontrolera



Rysunek 1:

5 Kod programu

5.1 main.c

```
1 #include "main.h"
2
3 /* Private includes -----*/
4 /* USER CODE BEGIN Includes */
5 #include "lcd.h"
6 /* USER CODE END Includes */
7
8 /* Private function prototypes -----*/
9 void SystemClock_Config(void);
10 static void MX_GPIO_Init(void);
11
12 int main(void)
13 {
14     /* USER CODE BEGIN 1 */
15
16     /* USER CODE END 1 */
17
18     /* MCU Configuration -----*/
```

```

19
20 /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
21 HAL_Init();
22
23 /* Configure the system clock */
24 SystemClock_Config();
25
26 /* Initialize all configured peripherals */
27 MX_GPIO_Init();
28 /* USER CODE BEGIN 2 */
29
30 LCD_Init();
31 pierwszy_znak();
32 drugi_znak();
33 LCD_Instruction(1, 1);
34 LCD_Instruction(0, 1);
35
36 /* USER CODE END 2 */
37
38 /* Infinite loop */
39 /* USER CODE BEGIN WHILE */
40 while (1)
41 {
42     /* USER CODE END WHILE */
43     clear_display();
44     HAL_Delay(10);
45     LCD_Instruction(1, 1);
46     HAL_Delay(1000);
47     clear_display();
48     HAL_Delay(10);
49     LCD_Instruction(0, 1);
50     HAL_Delay(1000);
51
52     /* USER CODE BEGIN 3 */
53 }
54 /* USER CODE END 3 */
55 }
56
57 /**
58  * @brief System Clock Configuration
59  * @retval None
60  */
61 void SystemClock_Config(void)
62 {
63     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
64     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
65

```

```

66  /** Initializes the RCC Oscillators according to the specified parameters
67  * in the RCC_OscInitTypeDef structure.
68  */
69  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
70  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
71  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
72  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
73  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
74  {
75      Error_Handler();
76  }
77  /** Initializes the CPU, AHB and APB buses clocks
78  */
79  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
80                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
81  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
82  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
83  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
84  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
85
86  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) != HAL_OK)
87  {
88      Error_Handler();
89  }
90 }
91
92 /**
93  * @brief GPIO Initialization Function
94  * @param None
95  * @retval None
96  */
97 static void MX_GPIO_Init(void)
98 {
99     GPIO_InitTypeDef GPIO_InitStruct = {0};
100
101     /* GPIO Ports Clock Enable */
102     __HAL_RCC_GPIOC_CLK_ENABLE();
103     __HAL_RCC_GPIOA_CLK_ENABLE();
104
105     /*Configure GPIO pin Output Level */
106     HAL_GPIO_WritePin(GPIOC, LCD_E_Pin|LCD_RS_Pin, GPIO_PIN_RESET);
107
108     /*Configure GPIO pin Output Level */
109     HAL_GPIO_WritePin(GPIOA, LCD_D0_Pin|LCD_D1_Pin|LCD_D2_Pin|LCD_D3_Pin
110                       |LCD_D4_Pin|LCD_D5_Pin|LCD_D6_Pin, GPIO_PIN_RESET);
111
112     /*Configure GPIO pins : LCD_E_Pin LCD_RS_Pin */

```

```

113 GPIO_InitStruct.Pin = LCD_E_Pin|LCD_RS_Pin;
114 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
115 GPIO_InitStruct.Pull = GPIO_NOPULL;
116 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
117 HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
118
119 /*Configure GPIO pins : LCD_D0_Pin LCD_D1_Pin LCD_D2_Pin LCD_D3_Pin
120                      LCD_D4_Pin LCD_D5_Pin LCD_D6_Pin */
121 GPIO_InitStruct.Pin = LCD_D0_Pin|LCD_D1_Pin|LCD_D2_Pin|LCD_D3_Pin
122                      |LCD_D4_Pin|LCD_D5_Pin|LCD_D6_Pin;
123 GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
124 GPIO_InitStruct.Pull = GPIO_NOPULL;
125 GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
126 HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
127
128 /*Configure GPIO pin : LCD_D7_Pin */
129 GPIO_InitStruct.Pin = LCD_D7_Pin;
130 GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
131 GPIO_InitStruct.Pull = GPIO_NOPULL;
132 HAL_GPIO_Init(LCD_D7_GPIO_Port, &GPIO_InitStruct);
133
134 }

```

5.2 lcd.c

```
1 #include "main.h"
2 #include "lcd.h"
3
4 void LCD_Instruction (uint8_t DB, uint8_t RS)
5 {
6     HAL_Delay(2);
7     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
8
9     if (RS)
10         HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, SET );
11     else
12         HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
13
14     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, DB & (1<<7) ? SET : RESET);
15     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, DB & (1<<6) ? SET : RESET);
16     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, DB & (1<<5) ? SET : RESET);
17     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, DB & (1<<4) ? SET : RESET);
18     HAL_GPIO_WritePin (GPIOA, LCD_D3_Pin, DB & (1<<3) ? SET : RESET);
19     HAL_GPIO_WritePin (GPIOA, LCD_D2_Pin, DB & (1<<2) ? SET : RESET);
20     HAL_GPIO_WritePin (GPIOA, LCD_D1_Pin, DB & (1<<1) ? SET : RESET);
21     HAL_GPIO_WritePin (GPIOA, LCD_D0_Pin, DB & (1<<0) ? SET : RESET);
22     HAL_Delay(1);      //For long RC time constants
23     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
24 }
25
26 void display_ON()
27 {
28     LCD_Instruction(0b00001110 , 0);
29     HAL_Delay(2);
30 }
31
32 void display_OFF()
33 {
34
35     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
36     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
37     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
38     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
39     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
40     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
41     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
42     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
43     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_RESET);
44
45     HAL_Delay(2);
46 }
```

```

47 }
48
49 void clear_display()
50 {
51
52
53     LCD_Instruction (0b00000001, 0);
54
55 }
56
57 void return_home()
58 {
59
60     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
61     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
62     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
63     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
64     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
65     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
66     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
67     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET);
68
69 }
70
71 void entry_mode()
72 {
73
74     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_1, GPIO_PIN_RESET);
75     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
76     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
77     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
78     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
79     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
80     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
81     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET); // I/D
82     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);    // S
83
84 }
85
86 void LCD_Init()
87 {
88     HAL_Delay (50);           //wait for more then 15ms after Vcc raises to 4.5V
89                               //+ wait for more then 40ms
90     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
91
92     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
93     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);

```



```

94     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
95     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, SET);
96     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, SET);
97     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
98     HAL_Delay(7);
99     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
100
101     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
102     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
103     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
104     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, SET);
105     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, SET);
106     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
107     HAL_Delay(2);
108     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
109
110     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
111     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
112     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
113     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, SET);
114     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, SET);
115     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
116     HAL_Delay(1);
117     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
118
119     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
120     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
121     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
122     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, SET);
123     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, SET);
124     HAL_GPIO_WritePin (GPIOA, LCD_D3_Pin, SET);
125     HAL_GPIO_WritePin (GPIOA, LCD_D2_Pin, RESET);
126     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
127     HAL_Delay(5);
128     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );;
129
130     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
131     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
132     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
133     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, RESET);
134     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, RESET);
135     HAL_GPIO_WritePin (GPIOA, LCD_D3_Pin, SET);
136     HAL_GPIO_WritePin (GPIOA, LCD_D2_Pin, SET);
137     HAL_GPIO_WritePin (GPIOA, LCD_D1_Pin, SET);
138     HAL_GPIO_WritePin (GPIOA, LCD_D0_Pin, RESET);
139     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
140     HAL_Delay(6);

```

```

141     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
142
143     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
144     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
145     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
146     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, RESET);
147     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, RESET);
148     HAL_GPIO_WritePin (GPIOA, LCD_D3_Pin, RESET);
149     HAL_GPIO_WritePin (GPIOA, LCD_D2_Pin, RESET);
150     HAL_GPIO_WritePin (GPIOA, LCD_D1_Pin, RESET);
151     HAL_GPIO_WritePin (GPIOA, LCD_D0_Pin, SET);
152     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
153     HAL_Delay(4);
154     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, SET );
155
156     HAL_GPIO_WritePin (GPIOC, LCD_RS_Pin, RESET );
157     HAL_GPIO_WritePin (GPIOA, LCD_D7_Pin, RESET);
158     HAL_GPIO_WritePin (GPIOA, LCD_D6_Pin, RESET);
159     HAL_GPIO_WritePin (GPIOA, LCD_D5_Pin, RESET);
160     HAL_GPIO_WritePin (GPIOA, LCD_D4_Pin, RESET);
161     HAL_GPIO_WritePin (GPIOA, LCD_D3_Pin, RESET);
162     HAL_GPIO_WritePin (GPIOA, LCD_D2_Pin, SET);
163     HAL_GPIO_WritePin (GPIOA, LCD_D1_Pin, SET);
164     HAL_GPIO_WritePin (GPIOA, LCD_D0_Pin, RESET);
165     HAL_GPIO_WritePin (GPIOC, LCD_E_Pin, RESET );
166     HAL_Delay(4);
167 }
168
169 void wyswietl_na_lcd(char *tekst)
170 {
171
172     while(*tekst)
173     {
174         LCD_Instruction(*tekst++, 1);
175     }
176 }
177
178 void pierwszy_znak()
179 {
180
181     LCD_Instruction(0b01000000,0);
182     HAL_Delay(2);
183     LCD_Instruction(0b00000000,1);
184     HAL_Delay(2);
185     LCD_Instruction(0b00000000,1);
186     HAL_Delay(2);
187     LCD_Instruction(0b00001010,1);

```

```

188     HAL_Delay(2);
189     LCD_Instruction(0b00010101,1);
190     HAL_Delay(2);
191     LCD_Instruction(0b00010001,1);
192     HAL_Delay(2);
193     LCD_Instruction(0b00001010,1);
194     HAL_Delay(2);
195     LCD_Instruction(0b00000100,1);
196     HAL_Delay(2);
197     LCD_Instruction(0b00000000,1);
198     HAL_Delay(2);
199     LCD_Instruction(0b10000000,0);
200     HAL_Delay(2);
201
202 }
203
204 void drugi_znak()
205 {
206
207     LCD_Instruction(0b01001000,0);
208     HAL_Delay(2);
209     LCD_Instruction(0b00000000,1);
210     HAL_Delay(2);
211     LCD_Instruction(0b00000000,1);
212     HAL_Delay(2);
213     LCD_Instruction(0b00001010,1);
214     HAL_Delay(2);
215     LCD_Instruction(0b00011111,1);
216     HAL_Delay(2);
217     LCD_Instruction(0b00011011,1);
218     HAL_Delay(2);
219     LCD_Instruction(0b00001110,1);
220     HAL_Delay(2);
221     LCD_Instruction(0b00000100,1);
222     HAL_Delay(2);
223     LCD_Instruction(0b00000000,1);
224     HAL_Delay(2);
225     LCD_Instruction(0b10000000,0);
226     HAL_Delay(2);
227
228 }

```

5.3 main.h

```
1 #ifndef __MAIN_H
2 #define __MAIN_H
3
4 #ifdef __cplusplus
5 extern "C" {
6 #endif
7
8 /* Includes -----*/
9 #include "stm32f3xx_hal.h"
10
11 /* Private includes -----*/
12 /* USER CODE BEGIN Includes */
13 #include "lcd.h"
14 /* USER CODE END Includes */
15
16 /* Exported types -----*/
17 /* USER CODE BEGIN ET */
18
19 /* USER CODE END ET */
20
21 /* Exported constants -----*/
22 /* USER CODE BEGIN EC */
23
24 /* USER CODE END EC */
25
26 /* Exported macro -----*/
27 /* USER CODE BEGIN EM */
28
29 /* USER CODE END EM */
30
31 /* Exported functions prototypes -----*/
32 void Error_Handler(void);
33
34 /* USER CODE BEGIN EFP */
35
36 /* USER CODE END EFP */
37
38 /* Private defines -----*/
39 #define LCD_E_Pin GPIO_PIN_0
40 #define LCD_E_GPIO_Port GPIOC
41 #define LCD_RS_Pin GPIO_PIN_1
42 #define LCD_RS_GPIO_Port GPIOC
43 #define LCD_D0_Pin GPIO_PIN_0
44 #define LCD_D0_GPIO_Port GPIOA
45 #define LCD_D1_Pin GPIO_PIN_1
46 #define LCD_D1_GPIO_Port GPIOA
```

```

47 #define LCD_D2_Pin  GPIO_PIN_2
48 #define LCD_D2_GPIO_Port  GPIOA
49 #define LCD_D3_Pin  GPIO_PIN_3
50 #define LCD_D3_GPIO_Port  GPIOA
51 #define LCD_D4_Pin  GPIO_PIN_4
52 #define LCD_D4_GPIO_Port  GPIOA
53 #define LCD_D5_Pin  GPIO_PIN_5
54 #define LCD_D5_GPIO_Port  GPIOA
55 #define LCD_D6_Pin  GPIO_PIN_6
56 #define LCD_D6_GPIO_Port  GPIOA
57 #define LCD_D7_Pin  GPIO_PIN_7
58 #define LCD_D7_GPIO_Port  GPIOA
59 /* USER CODE BEGIN Private defines */
60
61 /* USER CODE END Private defines */
62
63 #ifndef __cplusplus
64 }
65 #endif
66
67 #endif /* __MAIN_H */

```

5.4 lcd.h

```

1 #pragma once
2
3 #include <stdint.h>
4
5 void display_ON();
6
7 void display_OFF();
8
9 void clear_display();
10
11 void return_home();
12
13 void LCD_Init();
14
15 void LCD_Instruction (uint8_t DB, uint8_t RS);
16
17 void wyswietl_na_lcd(char *tekst);
18
19 void pierwszy_znak();
20
21 void drugi_znak();

```