

5η Εργαστηριακή Άσκηση

Εργαστήριο Μικροϋπολογιστών

Ανδρέας Στάμος (03120018), Κωνσταντίνος Πίκουλας (03120112)

Νοέμβριος 2023

Περιεχόμενα

1	Ζήτημα 5.1	1
2	Ζήτημα 5.2	2
3	Ζήτημα 5.3	2
	Α' Κοινές συναρτήσεις για το TWI και την επικοινωνία με το PCA9555	3
	Β' Κοινές συναρτήσεις για την επικοινωνία με την LCD οθόνη μέσω PCA9555	4

1 Ζήτημα 5.1

Το IO_PORT 0 ρυθμίζεται ως έξοδος. Μέσα σε έναν βρόχο, διαβάζουμε τα A, B, C, D από το PINB, υπολογίζουμε την ζητούμενη λογική συνάρτηση και στέλνουμε το αποτέλεσμα στο PCA9555 στο IO_PORT 0.

Οι κοινές συναρτήσεις για το TWI και την επικοινωνία με το PCA9555 βρίσκονται στο Παράρτημα Α'.

Ο κώδικας C είναι ο παρακάτω:

```
1  #define F_CPU 16000000UL
2  #include "../common/twi_pca9555.c"
3  #include<avr/io.h>
4  #include<stdint.h>
5
6  #define A PINB
7  #define B (PINB >> 1)
8  #define C (PINB >> 2)
9  #define D (PINB >> 3)
10
11 int main(void) {
12     //portb as input
13     DDRB = 0x00;
14     PORTB = 0xff;
15
16     twi_init();
17
18     //PCA9555 port 0 as output
19     PCA9555_0_write(REG_CONFIGURATION_0, 0x00);
20
21     while (1) {
22         uint8_t res = 0;
23         //calc f0
24         res |= (~(A & B) | (~B & C & D)) & 0x01;
25         //calc f1
26         res |= ((A & C & (B | D)) & 0x01) << 1;
27
28         PCA9555_0_write(REG_OUTPUT_0, res);
```

```

29     }
30 }

```

2 Ζήτημα 5.2

Χρησιμοποιούμε την γραμμή IO1_0 ως γείωση (γράφοντας 0) σε αυτή. Έτσι όταν ο χρήστης πατήσει κάποιο κουμπί, η αντίστοιχη γραμμή εισόδου γειώνεται, όπως την διαβάζουμε ως 0. Επειδή τα bits, τα δείχνουμε με την ίδια σειρά στο LEDs, μπορούμε απλά να κάνουμε shift 4 θέσεις δεξιά αυτό που διαβάσαμε. Επίσης απαιτείται να υπολογίσουμε την λογική άρνηση αυτού που διαβάσαμε, ώστε να μεταβούμε από την αρνητική λογική των κουμπιών όταν τα διαβάζουμε, στην θετική λογική των LEDs.

Οι κοινές συναρτήσεις για το TWI και την επικοινωνία με το PCA9555 βρίσκονται στο Παράρτημα [Α'](#).

Ο κώδικας C είναι ο παρακάτω:

```

1  #define F_CPU 16000000UL
2  #include "../common/twi_pca9555.c"
3
4  #define SWAP_NIBBLE(op) { asm("swap %0" : "+r" (op)); }
5
6  int main() {
7      twi_init();
8
9      //pca9555 io0 as output
10     PCA9555_0_write(REG_CONFIGURATION_0, 0x00);
11     //pca9555 io1[0] as output, io1[4:7] as input;
12     PCA9555_0_write(REG_CONFIGURATION_1, 0xF0);
13
14     //io1_0 pull down to zero.
15     PCA9555_0_write(REG_OUTPUT_1, 0x00);
16     while (1) {
17         uint8_t pressed = PCA9555_0_read(REG_INPUT_1);
18         //SWAP_NIBBLE(pressed);
19         pressed = ~pressed >> 4;
20         PCA9555_0_write(REG_OUTPUT_0, pressed);
21     }
22 }

```

3 Ζήτημα 5.3

Χρησιμοποιήσαμε τους κώδικες της 4ης εργαστηριακής σειράς, αλλάζοντας την εγγραφή στο PORTD σε εγγραφή στο IO PORT 1. Προκειμένου να γλιτώσουμε την ανάγνωση που συνέβαινε για να διατηρούμε πάνω στο PORTD αν πρόκειται για command ή data, περνάμε το αν πρόκειται για command ή data ως όρισμα στις αντίστοιχες συναρτήσεις.

Επίσης, προκειμένου να κάνουμε τις αλλαγές γραμμής, χρησιμοποιήσαμε την εντολή του μικροελεγκτή ST7066U της LCD οθόνης: Set DDRAM Address. Για την 1η γραμμή, θέτουμε την διεύθυνση DDRAM σε 0x00 και για την 2η γραμμή θέτουμε την διεύθυνση DDRAM σε 0x40 (σύμφωνα με το datasheet).

Οι κοινές συναρτήσεις για την επικοινωνία με την LCD οθόνη μέσω του PCA9555 βρίσκονται στο Παράρτημα [Β'](#).

Οι κοινές συναρτήσεις για το TWI και την επικοινωνία με το PCA9555 βρίσκονται στο Παράρτημα [Α'](#).

Ο κώδικας C είναι ο παρακάτω:

```

1  #define F_CPU 16000000UL
2
3  #include "../common/lcd.c"
4
5  const char message1[] = "JOHN"; //null-terminated
6  const char message2[] = "SMITH"; //null-terminated
7
8  int main(void) {
9      lcd_init();
10
11     for (uint8_t i=0; i<sizeof(message1)-1; ++i) lcd_data(message1[i]);

```

```

12     lcd_goto_line2();
13     for (uint8_t i=0; i<sizeof(message2)-1; ++i) lcd_data(message2[i]);
14 }
15

```

A' Κοινές συναρτήσεις για το TWI και την επικοινωνία με το PCA9555

Το αρχείο twi_pca9555_.c που χρησιμοποιείται στους παραπάνω κώδικες, για το TWI και την επικοινωνία με το PCA9555 είναι το εξής:

```

1  #ifndef __TWI_PCA9555_C
2  #define __TWI_PCA9555_C
3
4  #include<avr/io.h>
5  #include<util/twi.h>
6  #include<stdint.h>
7
8  #define REG_INPUT_0          0
9  #define REG_INPUT_1          1
10 #define REG_OUTPUT_0         2
11 #define REG_OUTPUT_1         3
12 #define REG_POLARITY_INV_0    4
13 #define REG_POLARITY_INV_1    5
14 #define REG_CONFIGURATION_0   6
15 #define REG_CONFIGURATION_1   7
16
17 #define PCA9555_0_ADDRESS 0x40
18
19 #define SCL_CLOCK 100000L
20 #define TWBR0_VALUE ((F_CPU/SCL_CLOCK-16)/2)
21
22 #define TWO_STATUS (TWSR0 & 0xF8)
23
24
25 void twi_init(void) {
26     TWSR0 = 0; //prescaler = 1
27     TWBR0 = TWBR0_VALUE;
28 }
29
30 unsigned char twi_start(uint8_t address) {
31     //7 msb of address are address and lsb is set if R and not set if W
32
33     TWCRO = 1<<TWSTA | 1<<TWINT | 1<<TWEN; //START
34     while (!(TWCRO & (1<<TWINT))); //wait till START transmitted
35     if (TWO_STATUS != TW_START && TWO_STATUS != TW_REP_START) return 1; //failed
36
37     TWDRO = address; //SLA_W or SLA_R
38     TWCRO = 1<<TWINT | 1<<TWEN;
39
40     while (!(TWCRO & (1<<TWINT))); //wait till SLA transmitted
41
42     //if SLA_R
43     if (address & 0x01) {
44         if (TWO_STATUS != TW_MR_SLA_ACK) return 1; //failed
45     } else {
46         //if SLA_W
47         if (TWO_STATUS != TW_MT_SLA_ACK) return 1; //failed
48     }
49

```

```

50     return 0;
51 }
52
53 void twi_stop(void) {
54     TWCRO = 1<<TWSTO | 1<<TWINT | 1<<TWEN; //STOP
55     while (!(TWCRO & (1<<TWSTO))); //wait till STOP transmitted
56 }
57
58 unsigned char twi_write(uint8_t data) {
59     TWDRO = data;
60     TWCRO = 1<<TWINT | 1<<TWEN;
61     while (!(TWCRO & (1<<TWINT))); //wait till transmitted
62     if (TWO_STATUS != TW_MR_DATA_ACK) return 1;
63     return 0;
64 }
65
66 unsigned char twi_readAck(void) {
67     TWCRO = 1<<TWINT | 1<<TWEA | 1<<TWEN;
68     while (!(TWCRO & (1<<TWINT))); //wait till received
69     return TWDRO;
70 }
71
72 unsigned char twi_readNak(void) {
73     TWCRO = 1<<TWINT | 1<<TWEN;
74     while (!(TWCRO & (1<<TWINT))); //wait till received
75     return TWDRO;
76 }
77
78 void twi_start_wait(uint8_t address) {
79     while (twi_start(address));
80 }
81
82 void PCA9555_0_write(uint8_t reg, uint8_t value) {
83     twi_start_wait(PCA9555_0_ADDRESS | TW_WRITE);
84     twi_write(reg); //what if PCA9555 NACKs?
85     twi_write(value);
86     twi_stop();
87 }
88
89 uint8_t PCA9555_0_read(uint8_t reg) {
90     uint8_t ret_val;
91     twi_start_wait(PCA9555_0_ADDRESS | TW_WRITE);
92     twi_write(reg); //what if PCA9555 NACKs?
93     twi_start(PCA9555_0_ADDRESS | TW_READ); //what if fails? should we wait instead?
94     ret_val = twi_readNak();
95     twi_stop();
96     return ret_val;
97 }
98
99 #endif
100

```

B' Κοινές συναρτήσεις για την επικοινωνία με την LCD οθόνη μέσω PCA9555

Το αρχείο lcd.c που χρησιμοποιείται στους παραπάνω κώδικες, για την επικοινωνία με την LCD οθόνη μέσω του PCA9555, είναι το εξής:

```

1  #ifndef __LCD_C
2  #define __LCD_C

```

```

3
4  #include<util/delay.h>
5  #include<avr/io.h>
6  #include "twi_pca9555.c"
7
8  #define NOP() { asm volatile ("nop"); }
9
10 #define RS_DATA 1
11 #define RS_COMMAND 0
12
13 void write_2_nibbles(uint8_t command, uint8_t type) {
14     PCA9555_0_write(REG_OUTPUT_0, (command & 0xf0) | type<<2 | 1<<3);
15     NOP();
16     NOP();
17     PCA9555_0_write(REG_OUTPUT_0, (command & 0xf0) | type<<2);
18
19     PCA9555_0_write(REG_OUTPUT_0, ((command & 0x0f) << 4) | type<<2 | 1<<3);
20     NOP();
21     NOP();
22     PCA9555_0_write(REG_OUTPUT_0, ((command & 0x0f) << 4) | type<<2);
23 }
24
25 void lcd_data(uint8_t data) {
26     write_2_nibbles(data, RS_DATA);
27     _delay_ms(0.250);
28 }
29
30 void lcd_command(uint8_t command) {
31     write_2_nibbles(command, RS_COMMAND);
32     _delay_ms(0.250);
33 }
34
35 void lcd_goto_line1(void) {
36     lcd_command((1<<7) | 0x00); //set ddram address to 0x00
37 }
38
39 void lcd_goto_line2(void) {
40     lcd_command((1<<7) | 0x40); //set ddram address to 0x40
41 }
42
43 void lcd_clear_display(void) {
44     lcd_command(0x01);
45     _delay_ms(5);
46 }
47
48 void lcd_init(void) {
49     twi_init();
50
51     //configure pca9555_io0 as output
52     PCA9555_0_write(REG_CONFIGURATION_0, 0x00);
53
54     _delay_ms(200); // wait for screen to initialize
55     for(uint8_t i=0; i<3; ++i) {
56         PCA9555_0_write(REG_OUTPUT_0, 0x38); // set 8 bit mode 3 times
57         NOP();
58         NOP();
59         PCA9555_0_write(REG_OUTPUT_0, 0x30);
60         _delay_ms(0.250);
61     }
62
63     // switch to 4bit mode

```

```
64     PCA9555_0_write(REG_OUTPUT_0, 0x28);
65     NOP();
66     NOP();
67     PCA9555_0_write(REG_OUTPUT_0, 0x20);
68     _delay_ms(0.250);
69
70     lcd_command(0x28);
71     lcd_command(0x0c);
72     lcd_clear_display();
73 }
74
75 #endif
76
```