ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΑΚΑΔ. ΕΤΟΣ 2024-2025

ΑΘΗΝΑ 8 Νοεμβρίου 2024

7η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"

ΟΜΑΔΑ 23

Συνεργάτες

Νικόλαος Αναγνώστου
03121818

Νικόλαος Λάππας
03121098

Ζήτημα 7.1: Στην συγκεκριμένη άσκηση κληθήκαμε να υλοποιήσουμε σε γλώσσα C τις συναρτήσεις που δίνονται σε assembly για την επικοινωνία του μικροελεγκτή ATmega328PB με τον αισθητήρα θερμοκρασίας DS1820/DS18B20. Θα πρέπει να προσέχουμε ότι κάνουμε χρήση τον ακροδέκτη PD4 και να τροποποιούμε μόνο αυτόν όπως ακριβώς ορίζουν οι συναρτήσεις, και πως σε αυτόν τον ακροδέκτη να είναι συνδεδεμένος μόνο ένας αισθητήρας. Ζητείται επιπλέον μια συνάρτηση, την ονομάσαμε $GetTemperature()$ και είναι τύπου int16_t, η οποία χρησιμοποιεί τις συναρτήσεις που δίνονται σε assembly και επιστρέφει την τιμή που κατέγραψε ο αισθητήρας. Ακολουθούν οι συναρτήσεις με αναλυτικούς κώδικες:

```
#define F_CPU 16000000UL

#include<avr/io.h>
#include<avr/interrupt.h>
#include<util/delay.h>
#include<stdbool.h>

// Returns true if a connected device is found (PD4 = 0)
bool one_wire_reset()
{
  DDRD |= (1 << PD4);         // Set PD4 as output
  PORTD &= ~(1 << PD4);       // Clear PD4
  _delay_us(480);       // Delay 480 usec

  DDRD &= ~(1 << PD4);        // Set PD4 as input
  PORTD &= ~(1 << PD4);       // Disable pull-up resistor
  _delay_us(100);       // Delay 100 usec

  uint8_t input = PIND & (1 << PD4);   // Read input
  _delay_us(380);       // Delay 380 usec

  // If device is detected (PD4 = 0) -> return true
  if (input == 0x10) {return false;} // PD4 = 1
  return true;                  // PD4 = 0
}


uint8_t one_wire_receive_bit()
{
  DDRD |= (1 << PD4);         // Set PD4 as output
  PORTD &= ~(1 << PD4);       // Clear PD4
  _delay_us(2);               // Delay 2 usec

  DDRD &= ~(1 << PD4);        // Set PD4 as input
  PORTD &= ~(1 << PD4);       // Disable pull-up resistor
  _delay_us(10);              // Delay 10 usec

  uint8_t bit_to_receive = (PIND & (1 << PD4)) ? 1 : 0;
  _delay_us(49);              // Delay 49 usec
```

```
  return bit_to_receive;
}

void one_wire_transmit_bit(uint8_t bit_to_transmit)
{
  DDRD |= (1 << PD4);          // Set PD4 as output
  PORTD &= ~(1 << PD4);        // Clear PD4
  _delay_us(2);                // Delay 2 usec

  //PORTD |= (bit_to_transmit & 0x10);   // Send PD4 bit to connected device
  PORTD = (PORTD & ~(1 << PD4)) | ((bit_to_transmit & 0x01) ? (1 << PD4) : 0);
  _delay_us(58);               // Delay 58 usec

  DDRD &= ~(1 << PD4);         // Set PD4 as input
  PORTD &= ~(1 << PD4);        // Disable pull-up resistor
  _delay_us(1);                // Delay 1 usec
}

uint8_t one_wire_receive_byte()
{
  uint8_t received_byte = 0x00;      // Store the byte (8-bit) we received
  for (uint8_t i = 0; i < 8; i++)
  {
    uint8_t received_bit = one_wire_receive_bit();
    received_byte |= (received_bit << i);          // Logical shift left, because DS18B20 send LSB first
    // Logical OR to insert new bit into byte sequence
  }
  return received_byte;
}

void one_wire_transmit_byte(uint8_t byte_to_transmit)
{
  for (uint8_t i = 0; i < 8; i++)
  {
    uint8_t send_bit = (byte_to_transmit >> i) & 0x01;// Bit to transmit now in position bit 0
    one_wire_transmit_bit(send_bit);
  }
}

int16_t GetTemperature()
{
  bool connected_device = one_wire_reset();   // Check for connected device
  if (!connected_device) return 0x8000;      // Error in connection return 0x8000

  one_wire_transmit_byte(0xCC);            // Only one device
  one_wire_transmit_byte(0x44);           // Begin counting temperature

  while (!one_wire_receive_bit());    // Wait until the above counting terminates
```

```
  one_wire_reset();                 // Re-initialize
  one_wire_transmit_byte(0xCC);
  one_wire_transmit_byte(0xBE);          // Read 16-bit result of temperature value

  uint16_t temperature = 0;
  temperature |= one_wire_receive_byte();    // 8-bit LSB of the total 16-bit value
  // Shift the 8-bit value 8 times to the left, OR with the previous 8-bit value
  // And take the temperature value of 16-bit
  temperature |= ((uint16_t)one_wire_receive_byte() << 8);

  return temperature;
}
```

Ζήτημα 7.2: Σε αυτή την άσκηση κληθήκαμε να χρησιμοποιήσουμε τις συναρτήσεις που κατασκευάσαμε προηγουμένως και να εντάξουμε την εκτύπωση της θερμοκρασίας. Συγκεκριμένα, χρησιμοποιήσαμε τον τρόπο με το PCA9555 προκειμένου να αποφύγουμε οποιαδήποτε ατασθαλία, αφού τόσο το LCD όσο και το θερμόμετρο επηρεάζει το PORTD και το PD4 αντίστοιχα. (Θα μπορούσε να γίνει το LCD κανονικά και με το PORTB, απλά με προσοχή όταν γίνεται χρήση του PD4 από τον αισθητήρα θερμοκρασία να μην κάνουμε εκτύπωση με το LCD). Παρακάτω ακολουθεί η συνάρτηση $int\ main()$ διότι οι υπόλοιπες συναρτήσεις είναι της 7.1 και το LCD με PCA υπάρχει ήδη σε προηγούμενη σειρά:

```
int main(){
  DDRB = 0xff;
  DDRC = 0x00;

  twi_init();
  PCA9555_0_write(REG_CONFIGURATION_0, 0x00); // EXT_PORT0 -> output

  lcd_init();

  while(1)
  {

    lcd_clear_display();

    uint16_t temperature = GetTemperature();


    if (temperature == 0x8000) // NO Device 9 bits no need for extra line
    {
      lcd_data('N');
      lcd_data('O');
      lcd_data(' ');
      lcd_data('D');
      lcd_data('e');
```

```
    lcd_data('v');
    lcd_data('i');
    lcd_data('c');
    lcd_data('e');
  }
  else {

    if ((temperature & 0x8000) > 0) {
      temperature = ~temperature + 1;
      lcd_data('-');
    } //if negative convert to its value
    else lcd_data('+');

    int dekadika = 0;
    //for now
    if((temperature&0x01)==0x01) dekadika += 625;
    temperature = temperature>>1;
    if((temperature&0x01)==0x01) dekadika += 1250;
    temperature = temperature>>1;
    if((temperature&0x01)==0x01) dekadika += 2500;
    temperature = temperature>>1;
    if((temperature&0x01)==0x01) dekadika += 5000;
    temperature = temperature >> 1 ;




    int result = 0;
    for (int i = 0; i <= 6; i++){
      if (temperature & 0x0001 > 0) result += (int)pow(2,i);
      temperature = temperature >> 1;
    }
    bool is_zero = false;
    if(result/100!=0) {
      lcd_data('0'+(result/100));
      result = result % 100;
    }
    else is_zero = true;



    if(result/10==0 && is_zero==true) ;
    else{
      lcd_data('0'+(result/10));
      result = result % 10;
    }
    lcd_data('0'+result);
    //might add something here
    lcd_data('.');
```

```
        result = dekadika/1000;
        lcd_data('0'+result);
        dekadika %= 1000;

        result = dekadika/100;
        lcd_data('0'+result);
        dekadika %= 100;

        result = dekadika/10;
        lcd_data('0'+result);

        dekadika %= 10;
        lcd_data('0'+dekadika);



        lcd_data(' ');
        lcd_data(223);
        lcd_data('C');

      }
      _delay_ms(750);

   }
   return 0;
}
```