8η Εργαστηριακή Αναφορά

**Ομάδα 20**

Παπαδόπουλος Χαράλαμπος 03120199

Στρίφτης Γεώργιος 03121200

|  |
| --- |
| #define F\_CPU 16000000UL  #include<avr/io.h>  #include<avr/interrupt.h>  #include<util/delay.h>  #include <stdio.h>  #include <string.h>  #include "setup\_LCD\_PEX.h"  #include "setup\_TWI.h"  #include "setup\_OWI.h"  #include "setup\_USART.h"  #include "setup\_KEYBOARD.h"  static uint8\_t temp\_l, temp\_h;  extern uint16\_t pressed\_keys;  typedef enum {  NURSE\_CALL, // 0  OK, // 1  CHECK\_TEMP, // 2  CHECK\_PRESSURE // 3  } STATUS\_t;  void comms();  const char\* status\_to\_string(STATUS\_t s);  void transmit\_state(double temperature, double pressure, STATUS\_t status);  void transmit\_payload(double temperature, double pressure, int team, STATUS\_t status);  STATUS\_t check\_button();  double measure\_temp();  double read\_adc();  void transmit\_esp();  void setup() {  twi\_init();  \_delay\_ms(50); // Delay 500 ms for better stability  PCA9555\_0\_write(REG\_CONFIGURATION\_0, 0x00); // Set as output  lcd\_init();  \_delay\_ms(50);  lcd\_clear\_display();  \_delay\_us(100);  usart\_init(103);  ADMUX = 0b01000000; // ADC right-adjusted, select ADC2  ADCSRA = 0b10000111; // Enable ADC with a prescaler of 128  setup\_keyboard();  \_delay\_ms(50); // Delay 500 ms for better stability  }  int main()  {  setup();  double pressure;  STATUS\_t status;  double temperature;  int team = 20;      while(1){  lcd\_clear\_display();  lcd\_set\_cursor(0, 0);    comms();  status = check\_button();  pressure = read\_adc();  temperature = measure\_temp();    if((pressure < 4 || pressure > 12) && status != NURSE\_CALL)  status = CHECK\_PRESSURE;  else if ((temperature <= 34.0 || temperature >= 37.0) && status != NURSE\_CALL)  status = CHECK\_TEMP;    transmit\_state(temperature, pressure, status);  transmit\_payload(temperature, pressure, team, status);  transmit\_esp();    }  }  double measure\_temp()  {  int16\_t temperature = 0;  if(one\_wire\_reset()) {  one\_wire\_transmit\_byte(0xCC);  one\_wire\_transmit\_byte(0x44);  while(!one\_wire\_receive\_bit()) {  //busy waiting  }  if(one\_wire\_reset()) {  one\_wire\_transmit\_byte(0xCC);  one\_wire\_transmit\_byte(0xBE);  temp\_l = one\_wire\_receive\_byte();  temp\_h = one\_wire\_receive\_byte();  }  else {  temp\_l = 0x00;  temp\_h = 0x80;  }  }  else {  temp\_l = 0x00;  temp\_h = 0x80;  }    temperature = (temp\_h & 0b00000111) << 8;  temperature |= temp\_l;  double result = ((double)temperature \* 0.0625) + 10.0;      return result;  }  double read\_adc()  {  double adc;  double output;  ADCSRA |= (1 << ADSC);  while (ADCSRA & (1 << ADSC));  adc = ADC;  output = (adc \* 20) / 1024;  return output;  }  void comms()  {  char message1[] = "ESP:connect";  char message2[] = "ESP:url:\"http://192.168.1.250:5000/data\"";  char c;  for(int i = 0; i < strlen(message1); i++) {  usart\_transmit(message1[i]);  \_delay\_us(10);  }  usart\_transmit('\n');    lcd\_data('1');  lcd\_data('.');  do {  c = usart\_receive();  if(c != '\n')  lcd\_data(c);  } while(c != '\n');      lcd\_set\_cursor(1, 0);  for(int i = 0; i < strlen(message2); i++) {  usart\_transmit(message2[i]);  \_delay\_us(10);  }  usart\_transmit('\n');    lcd\_data('2');  lcd\_data('.');  do {  c = usart\_receive();  if(c != '\n')  lcd\_data(c);  } while(c != '\n');  }  const char\* status\_to\_string(STATUS\_t s) {  switch (s) {  case OK: return "OK";  case NURSE\_CALL: return "NURSE CALL";  case CHECK\_TEMP: return "CHECK TEMP";  case CHECK\_PRESSURE: return "CHECK PRESSURE";  }  return "OK";  }  void transmit\_state(double temperature, double pressure, STATUS\_t status)  {  char pres\_out[5];  char status\_out[16];  char temp\_out[5];  lcd\_clear\_display();  lcd\_data('T');  lcd\_data(':');  sprintf(temp\_out, "%.1f", temperature);  for (int i = 0; i < strlen(temp\_out); i++)  lcd\_data(temp\_out[i]);    lcd\_data(' ');    lcd\_data('P');  lcd\_data(':');  snprintf(pres\_out, 5, "%.2f", pressure);  for (int i = 0; i < strlen(pres\_out); i++)  lcd\_data(pres\_out[i]);    lcd\_set\_cursor(1, 0);  snprintf(status\_out, 16, status\_to\_string(status));  for (int i = 0; i < strlen(status\_out); i++)  lcd\_data(status\_out[i]);    \_delay\_ms(2000);  }  void transmit\_payload(double temperature, double pressure, int team, STATUS\_t status)  {  char payload[256] = {0};  char c;  snprintf(payload, 256,  "ESP:payload:[{\"name\": \"temperature\",\"value\": \"%.2f\"},"  "{\"name\": \"pressure\",\"value\": \"%.2f\"},"  "{\"name\": \"team\",\"value\": \"%d\"},"  "{\"name\": \"status\",\"value\": \"%s\"}]",  temperature, pressure, team, status\_to\_string(status));    for(int i = 0; i < strlen(payload); i++) {  usart\_transmit(payload[i]);  \_delay\_us(10);  }  usart\_transmit('\n');    lcd\_clear\_display();  lcd\_data('3');  lcd\_data('.');  do {  c = usart\_receive();  if(c != '\n')  lcd\_data(c);  } while(c != '\n');  \_delay\_ms(2000);  return;  }  void transmit\_esp()  {  char message[] = "ESP:transmit";  char c;    for(int i = 0; i < strlen(message); i++) {  usart\_transmit(message[i]);  \_delay\_us(10);  }  usart\_transmit('\n');      lcd\_set\_cursor(1, 0);  lcd\_data('4');  lcd\_data('.');  do {  c = usart\_receive();  if(c != '\n')  lcd\_data(c);  } while(c != '\n');    \_delay\_ms(1000);  }  STATUS\_t check\_button()  {  int num = 0;  STATUS\_t status = OK;  int del = 0;  while(del < 500) {  scan\_keypad\_rising\_edge();  if (pressed\_keys != 0) {  num = keypad\_to\_ascii();  if( num == 48)  status = NURSE\_CALL;  else if(num == 35)  status = OK;  }  \_delay\_us(100);  del++;  }  return status;  } |