



ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ

ΟΜΑΔΑ 1
3 ΕΓΡΑΣΙΑ/ΕΓΡΑΣΤΗΡΙΟ



30 ΜΑΙΟΥ 2022

Ανδρονίκου, Δημήτριος 9836 dimitrios@ece.auth.gr
Γιαννόπουλος, Νικόλαος 9629 ngiannop@ece.auth.gr

Φιλοσοφία κώδικα

Αρχικά, γράφουμε μια βιβλιοθήκη ώστε να μπορούμε να επικοινωνούμε με τον αισθητήρα. Σύμφωνα με το manual του αισθητήρα, χρειάζονται 3 συναρτήσεις για να επικοινωνήσουμε, καθώς η επικοινωνία χωρίζεται σε 3 βασικά στάδια.

Πρώτα, απ' το Nucleo, στέλνουμε κάποιο σήμα για να αρχίσουμε την επικοινωνία.

```
6 #define PIN PA_10
7
8 void DHT11_Start(void){
9     // Start pin transfer data
10    gpio_set_mode(PIN, Output);    // Set PA_10 as output
11    gpio_set(PIN,0);               // PA_10 set low (see manual DTH11)
12    delay_ms(18);                  // wait for 18 ms
13    gpio_set(PIN,1);               // PA_10 set high (see manual DTH11)
14    delay_us(20);                  // wait for 20 us
15    gpio_set_mode(PIN, Input);     // Start receiving
16 }
17
```

Έπειτα, περιμένουμε απάντηση ότι ο αισθητήρας, έλαβε το αρχικό σήμα και είναι έτοιμος να μεταδώσει την πληροφορία του.

```
18 uint8_t DHT11_check_response(void){
19     uint8_t response = 0;
20     delay_us(40);                  // Wait for DHT response middle signal
21     if(!(gpio_get(PIN))){
22         delay_us(80);              // DHT middle Pull up
23         if((gpio_get(PIN)) response = 1; // if the pin is high, response is ok
24         else response = 0;         // if response == 0 is false response
25     }
26     while(gpio_get(PIN));
27     return response;
28 }
29
```

Τέλος λαμβάνουμε τα byte (8-bit), της πληροφορίας.

```
31 uint8_t DHT11_Read(void){
32     uint8_t i,j;
33     for (j = 0; j < 8; j++){
34         while(!(gpio_get(PIN))); // wait for the pin to go high
35         delay_us(28);            // wait for 28 us
36         if(!(gpio_get(PIN))){    // if the pin is low
37             i&=~(1<<(7-j));      // write 0
38         }else i |= (1<<(7-j));    // if the pin is high, write 1
39         while((gpio_get(PIN)));  // wait for the pin to go low
40     }
41     return i;
42 }
43
```

Στη συνέχεια, στη main, ορίζουμε την συνάρτηση αρχικοποίησης και την συνάρτηση get για το switch καθώς και της global μεταβλητές που χρειαζόμαστε για τα καταλληλά μηνύματα αλλά και την παραλαβή των δεδομένων από τον αισθητήρα.

```

15 float temp = -100.0;          // tempure
16 int sRate = 0 , counter = 0;  // Sampling Rate propably global to chenaghe it when switch press it
17
18
19
20 void switch_init(void){
21     //define P_SW PC13
22     gpio_set_mode(P_SW, PullUp); // 1 switch
23
24 }
25
26 int switch_get(Pin pin){
27     return !gpio_get(pin); // active low
28 }
29
30 uint8_t RH_bytel , RH_byt2, Temp_bytel, Temp_byte2, SUM, presence;
31
32 char charTemp[64];
33 char charsRate[30];

```

Έπειτα, ορίζομαι το πως θα λειτουργεί ο timer στο interrupt του.

```

43 void timer_ISR(void){
44     DHT11_Start();
45     presence = DHT11_check_response(); // if (presensce == 1 ) continue
46     if (presence != 1){
47         printf("Error");
48     }
49     RH_bytel = DHT11_Read();
50     RH_byt2 = DHT11_Read();
51     Temp_bytel = DHT11_Read();
52     Temp_byte2 = DHT11_Read();
53     SUM = DHT11_Read();
54
55     temp = (float) Temp_bytel; // getting float value of temp
56
57
58
59
60     sprintf(charTemp,"%f", temp);
61     sprintf(charsRate,"%d", sRate);
62
63     uart_print("Temp= ");
64     uart_print(charTemp);
65     uart_print("\r\n");
66
67     uart_print("Sample Rate= ");
68     uart_print(charsRate);
69     uart_print("\r\n");
70 }

```

Φτιάχνουμε, επίσης, κάποιες συναρτήσεις που ελέγχουμε αν έχει πατηθεί το κουμπί, αν έχει πατηθεί μονές φορές ή ζυγές και μια συνάρτηση που καλείται όταν πατηθεί το κουμπί.

```

64 int boolFist = 0;
65 int boolZhgos = 0;
66 int boolMonos = 0;
67
68 void firstTime() {
69     if (boolFist == 1) {
70         timer_disable();
71         delay_ms(AEM2 * 1000);
72         //delay_ms(AEM1 * 1000);
73         boolFist = 0;
74         timer_enable();
75     }
76     //sRate = AEM1;
77     sRate = AEM2;
78 }
79
80 void zhgos_ISR() {
81     if (boolZhgos == 1) {
82         timer_disable();
83         delay_ms(4 * 10000);
84         boolZhgos = 0;
85         timer_enable();
86     }
87     sRate = 4; // set new rate
88 }
89
90 void monos_ISR() {
91     if (boolMonos == 1) {
92         timer_disable();
93         delay_ms(3 * 10000);
94         boolMonos = 0;
95         timer_enable();
96     }
97     sRate = 3; // set new rate.
98 }
99
100 void OnPressButton_isr() {
101     counter++;
102     if (counter == 1) {boolFist = 1; firstTime();}
103     if ((counter%2) == 0) {boolZhgos = 1; zhgos_ISR();}
104     if ((counter%3) == 0) {boolMonos = 1; monos_ISR();}
105 }

```

Στη main(), τέλος, αρχικοποιούμε κάποιες μεταβλητές, όπως επίσης τη uart, τον διακόπτη, τον αισθητήρα και τα led. Ενεργοποιούμε τη σειρά προτεραιότητας των interrupts.

Μέσα στο while, είναι η βασική λειτουργία με βάση τη λαμβανόμενη θερμοκρασία.

Όλα τα υπόλοιπα έχουν υλοποιηθεί στις προηγούμενες συναρτήσεις.

```

107 int main() {
108
109     uart_init(9600);           // Baud -> 9600
110     uart_enable();
111
112
113     leds_init();
114     switch_init();
115     gpio_set_trigger(P_SW, Rising);
116     gpio_set_callback(P_SW, OnPressButton_isr);
117
118     sRate = 1;
119     timer_init(1000000); // set period for 1 seconds
120
121     timer_set_callback(timer_ISR); // sets timer_ISR to get the tempure from sensor
122
123     timer_enable();
124     __enable_irq();
125
126     while(1) {
127
128         if(temp < 20.0) {
129             leds_set(0,0,1); // set the leds to blue
130         } else if( temp > 25.0) {
131             leds_set(1,0,0); // set the leds to red
132         } else {
133             leds_set(0,1,0); // set the leds to green
134         }
135     }
136 }
137

```

Προβλήματα

Οι βασικές δυσκολίες του εργαστηρίου ήταν η δημιουργία της βιβλιοθήκης για τον αισθητήρα, από τις οδηγίες του αισθητήρα.

Testing

Τα τεστ έγιναν πάνω στο board όπου το στην αρχή με ρυθμό 1 sec μας εμφανίζει τα αποτελέσματα στην οθόνη και μετά το πρώτο πάτημα φαίνεται η καθυστέρηση που δημιουργείται και λειτουργεί σωστά και για τα μονά/ζυγά πατήματα.