

3^ο Εργαστήριο Μικροεπεξεργαστές και Περιφερειακά

Νίκος Λαδιάς

Χρήστος Τριφηνόπουλος

Υλοποίηση

Στην παρούσα άσκηση μας ζητείτε να διαβάζουμε περιοδικά την θερμοκρασία από τον αισθητήρα DHT11, για τον οποίο μας δόθηκε το τεχνικό datasheet και υλοποιήθηκε έτσι ο driver του αισθητήρα για να λαμβάνουμε τα δεδομένα από αυτόν.

Αρχικά ο μικροεπεξεργαστής πρέπει να στείλει κάποια σήματα συγκεκριμένης διάρκειας (σύμφωνα με το manual) στον αισθητήρα για να αρχίσει να λαμβάνει δεδομένα.

```
gpio_set_mode(pin, Output);  
    // REQUEST SAMPLE  
gpio_set(pin, 0);  
delay_ms(20); //wait for DHT to detect start signal  
gpio_set(pin, 1);  
delay_us(40); //wait for DHT to respond  
  
gpio_set_mode(pin, Input); //turn mode to input to receive sensor signal
```

Ο εντοπισμός κάποιου σφάλματος στο σήμα από τον αισθητήρα το οποίο θα το έκανε να ήταν μόνιμο σε επίπεδο HIGH υλοποιείται ως εξής:

```
loopCnt = TIMEOUT;  
while(gpio_get(pin) == 1)  
    if (loopCnt-- == 0) return -2;
```

Όπου το TIMEOUT είναι ένας αρκετά μεγάλος αριθμός ώστε να γνωρίζουμε ότι υπάρχει σφάλμα αν πρόλαβε να τρέξει αυτή η λούπα τόσες φορές χωρίς να πέσει στο LOW το σήμα που έρχεται από τον αισθητήρα.

Στην συνέχεια αφού μετρήσουμε την διάρκεια που το λαμβανόμενο σήμα βρίσκεται στην θέση HIGH, ο παρακάτω κώδικας αποφασίζει αν το εκάστοτε bit είναι 1 ή 0 και με ένα bitwise operation(μετατόπιση του ψηφίου 1 κατά έναν αριθμό μπιτ που αρχικοποιείται σε 7 και καταλήγει στο 0 καθώς λαμβάνουμε πρώτο το MSB και τελευταίο το LSB) τοποθετεί την τιμή στο κατάλληλο bit στην μνήμη του μικροεπεξεργαστή.

```
if (dur > 45) bits[idx] |= (1 << currentBit); //use bitwise or to modify the current bit  
if (currentBit == 0)    //LSB reached, go to next byte  
{  
    currentBit = 7;  
    idx++;  
}  
else currentBit--;
```

Έπειτα με την χρήση timer ISR, εκτελούμε τις εναλλαγές στις περιόδους εμφάνισης της θερμοκρασίας στην οθόνη. Με gpio ISR υλοποιήσαμε τον διακόπτη . Το τύπωμα της θερμοκρασίας γίνεται στην οθόνη του keil μέσα στην ISR ρουτίνα του timer. Κάθε εναλλαγή περιόδου γίνεται με βάση την εκφώνηση και το πάτημα του διακόπτη. Ο διακόπτης είναι στο Pin P_SW και είναι by default στο pull up, ενώ γίνεται triggered σε rising edge. Ο αισθητήρας είναι συνδεδεμένος στο pin PA_0.

```
void timer_callback_isr(){
    counter++;
    if (counter%2==0){
        temp=temperature_read();
        if(T==2){
            printf("temperature: %d",temp);
            printf("sampling period: 2s");
        }
    }
    if (T==3 && counter%3==0){
        printf("temperature: %d",temp);
        printf("sampling period: 2s");
    }
    else if (T==4 && counter%4==0){
        printf("temperature: %d",temp);
        printf("sampling period: 2s");
    }
    else if (T==8 && counter%8==0){
        printf("temperature: %d",temp);
        printf("sampling period: 2s");
    }
}

void button_press_isr(){
    buttonCnt++;
    if (buttonCnt==1)
        T=8;
    else if(buttonCnt%2==0)
        T=4;
    else
        T=3;
}
```

Στην main αρχικοποιήσαμε τα callbacks (timer&button) και την σειριακή επικοινωνία uart.

```
int main(){

    uart_init(9600);    //Initialize UART protocol
    uart_enable();

    timer_init(1000000);
    timer_set_callback(timer_callback_isr);
    timer_enable();

    gpio_set_callback(P_SW, button_press_isr);
    gpio_set_mode(P_SW, PullUp);
    gpio_set_trigger(P_SW, Rising);

    __enable_irq();
}
```

Στην συνέχεια με τον παρακάτω κώδικα υλοποιείται η λειτουργία με την οποία ελέγχουμε συνεχώς την τιμή της θερμοκρασίας (η οποία αλλάζει με την κλήση των callbacks) και ανάβουμε τα κατάλληλα led.

```
while(1){
    if(temp<20)
        leds_set(0,0,1); //blue led on //uart_print("blue led on");
    else if(temp>25)
        leds_set(1,0,0); //red led on
    else
        leds_set(0,1,0); //green led on
}
```