



ΑΘΗΝΑ, 21/11/2025

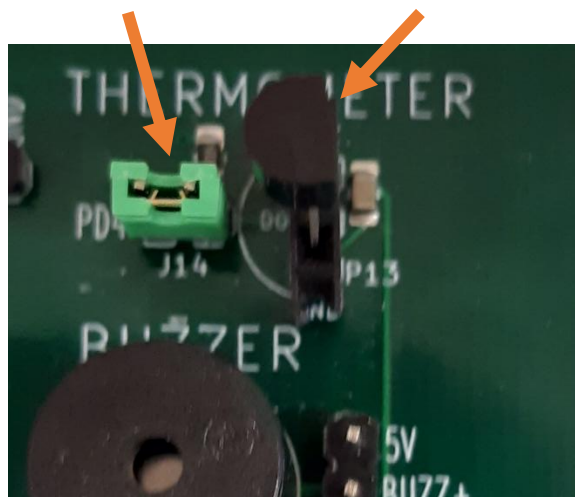
**7<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"**  
**Αισθητήρας Θερμοκρασίας DS1820 στην κάρτα ntuAboard\_G1**

**Εξέταση – Επίδειξη: Παρασκευή 28/11/2025**  
**Προθεσμία για παράδοση Έκθεσης: Τρίτη 2/12/2025 (23:59)**

**Χρήση Αισθητήρα Θερμοκρασίας DS1820 και Σειριακή Επικοινωνία 1-wire**

Η σειριακή επικοινωνία ενός καλωδίου (1-wire serial communication) είναι ένα πρωτόκολλο που επιτρέπει τη μεταφορά δεδομένων από μία μόνο καλωδιακή σύνδεση. Η διαδικασία συντονίζεται από ένα μικροελεγκτή σε ρόλο master. Το πλεονέκτημα του συγκεκριμένου πρωτοκόλλου είναι ότι απαιτεί έναν μόνο ακροδέκτη του μικροελεγκτή. Όλες οι συσκευές που συνδέονται σε αυτόν τον ακροδέκτη έχουν έναν προκαθορισμένο κωδικό slave, που επιτρέπει στο μικροελεγκτή να αναγνωρίζει εύκολα με ποια συσκευή επικοινωνεί κάθε φορά.

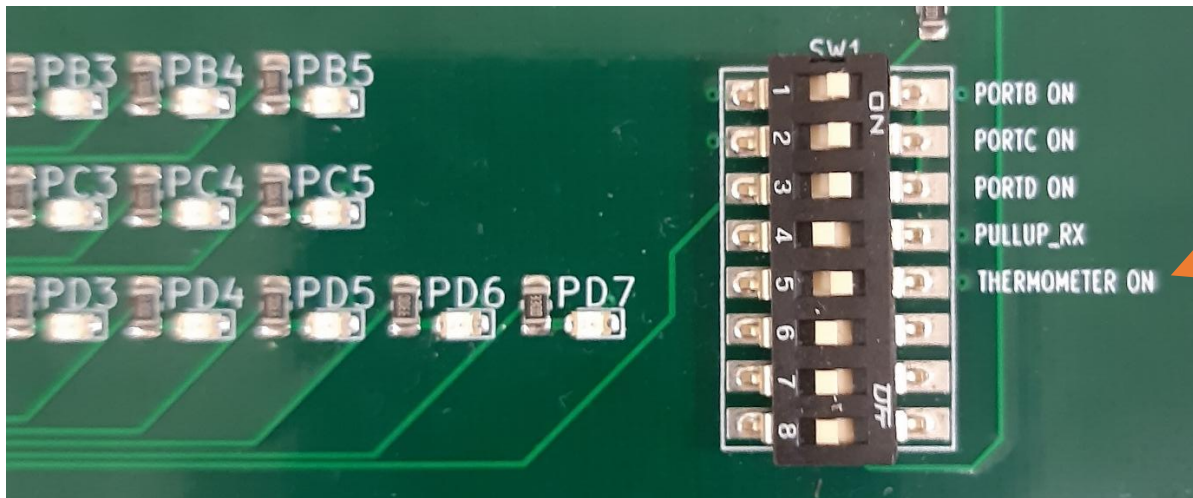
Η αναπτυξιακή πλακέτα *ntuAboard\_G1* είναι εφοδιασμένη με τον αισθητήρα θερμοκρασίας DS1820 (σχήμα 7.1) που χρησιμοποιεί το πρωτόκολλο σειριακής επικοινωνίας ενός καλωδίου. Μετράει θερμοκρασίες από -55°C έως 125°C και παρέχει ακρίβεια  $\pm 0.5^\circ\text{C}$  στην περιοχή θερμοκρασιών από -10°C έως 85°C. Για την τροφοδοσία του απαιτείται συνεχής τάση από 3V έως 5V. Ο αισθητήρας χρειάζεται μέγιστο χρόνο 750msec για να υπολογίσει τη θερμοκρασία με ακρίβεια 9 bit.



**Σχήμα 7.1** Αισθητήρας θερμοκρασίας DS1820.

Στην αναπτυξιακή πλακέτα *ntuAboard\_G1* ο αισθητήρας θερμοκρασίας DS1820 τοποθετείται στη δεξιά πάνω γωνία και για να συνδεθεί με τον ακροδέκτη PD4 του μικροελεγκτή, απαιτείται η χρήση ενός βραχυκυκλωτήρα στο κονέκτορα J14, όπως φαίνεται στο σχήμα 7.1.

Η σύνδεση της τροφοδοσίας στον αισθητήρα θερμοκρασίας DS1820 γίνεται δια μέσω του DIP Switch SW1-5 (THERMOMETER ON) όπως φαίνεται στο Σχήμα 7.2 παρακάτω



**Σχήμα 7.2** Σύνδεση τροφοδοσίας στον αισθητήραDS1820

Η επικοινωνία του μικροελεγκτή με τον αισθητήρα θερμοκρασίας πραγματοποιείται δια μέσω του ακροδέκτη PD4. Η τοποθέτηση του αισθητήρα στην πλακέτα *ntuAboard\_G1* γίνεται σε ειδική βάση, προσέχοντας το τυπωμένο ημικύκλιο στην πλακέτα να συμπίπτει με την ημικυκλική πλευρά του αισθητήρα, όπως στο σχήμα 7.1. Το ολοκληρωμένο DS1820 δεν είναι ένας απλός αισθητήρας αλλά μια σύνθετη συσκευή με ενσωματωμένο μετατροπέα ADC και πρωτόκολλο επικοινωνίας. Αν συνδεθεί λανθασμένα θα πάθει βλάβη.

Η επικοινωνία ενός καλωδίου είναι ένα πολύ απλό πρωτόκολλο που διαβάζει και γράφει σειριακά δεδομένα τηρώντας συγκεκριμένα χρονικά παράθυρα, όπως φαίνεται στο τεχνικό εγχειρίδιο του DS1820. Η σωστή λειτουργία του πρωτοκόλλου απαιτεί και την ύπαρξη μιας αντίστασης πρόσδεσης (pull-up) στον ακροδέκτη PD4.

Στη συνέχεια περιγράφεται ο κώδικας για την επικοινωνία του DS1820 με το μικροελεγκτή ATmega328PB που βρίσκεται πάνω στην αναπτυξιακή πλακέτα *ntuAboard\_G1*.

Η ρουτίνα αρχικοποίησης **one\_wire\_reset** εκπέμπει έναν παλμό αρχικοποίησης στη γραμμή και ανιχνεύει εάν υπάρχει κάποια συνδεδεμένη συσκευή. Ο ακροδέκτης PD4 αρχικά ρυθμίζεται ως έξοδος και τίθεται σε κατάσταση λογικού 0, για 480usec. Στη συνέχεια μετά από 100usec, διαβάζεται η κατάσταση της θύρας PD, και ακολουθεί μία επιπλέον χρονική καθυστέρηση 380usec ώστε να ολοκληρωθεί ο κύκλος αρχικοποίησης. Αν κάποια συνδεδεμένη συσκευή αποκριθεί και θέσει το PD4 σε κατάσταση λογικού 0 τότε επιστρέφεται η τιμή 1 στον καταχωρητή r24, διαφορετικά επιστρέφεται η τιμή 0.

one\_wire\_reset:

```

sbi DDRD, PD4      ; set PD4 as output

cbi PORTD, PD4     ;
ldi r24, low(480)   ; 480 usec reset pulse
ldi r25, high(480)  ;
rcall wait_usec    ;

cbi DDRD, PD4      ; set PD4 as input
cbi PORTD, PD4     ; disable pull-up

ldi r24, 100        ; wait 100 usec for connected devices
ldi r25, 0          ; to transmit the presence pulse
rcall wait_usec    ;

in r24, PIND        ; read PORTD

push r24            ; save PORTD

ldi r24, low(380)   ;
ldi r25, high(380)  ; wait for 380 usec
rcall wait_usec    ;

pop r25             ; retrieve PORTD

clr r24             ; if a connected device is
sbrs r25, PD4       ; detected(PD4=0) return 1
ldi r24, 0x01       ; else return 0
ret

```

Η ρουτίνα *one\_wire\_receive\_bit* έχει ως αντικείμενο το διάβασμα ενός bit, το οποίο αποθηκεύεται στον καταχωρητή **r24**, από τον αισθητήρα DS1820 δια μέσω του ακροδέκτη PD4, σύμφωνα με το πρωτόκολλο επικοινωνίας 1-wire serial communication.. Ξεκινά με εντολές που ρυθμίζουν τη θύρα PD ως έξοδο και ο ακροδέκτης PD4 τίθεται σε λογικό 0 για 2usec. Στη συνέχεια ρυθμίζει τη θύρα PD ως είσοδο, περιμένει 10usec, διαβάζει τον ακροδέκτη PD4 και γράφει την τιμή που διάβασε στον καταχωρητή r24. Τέλος δημιουργείται μια ακόμη χρονική καθυστέρηση 49usec (48usec για ολοκλήρωση του κύκλου ανάγνωσης και 1usec για αποκατάσταση) και η ρουτίνα επιστέφει.

one\_wire\_receive\_bit:

```

sbi DDRD, PD4      ; set PD4 as output

cbi PORTD, PD4     ;
ldi r24, 0x02      ; time slot 2 usec
ldi r25, 0x00      ;
rcall wait_usec    ;

cbi DDRD, PD4      ; set PD4 as input
cbi PORTD, PD4     ; disable pull-up

ldi r24, 10        ;
ldi r25, 0         ; wait 10 usec
rcall wait_usec    ;

clr r24            ;
sbic PIND, PD4     ; r24 = PD4
ldi r24, 1         ;
push r24

ldi r24, 49        ;
ldi r25, 0         ; delay 49 usec to meet the standards
rcall wait_usec    ;

pop r24
ret

```

Αντίστοιχα η ρουτίνα *one\_wire\_transmit\_bit*, έχει ως αντικείμενο την εγγραφή ενός bit, το οποίο βρίσκεται καταχωρητή r24, στον αισθητήρα DS1820 δια μέσω του ακροδέκτη PD4. Ξεκινά με εντολές που ρυθμίζουν τη θύρα PORTD ως έξοδο και ο ακροδέκτης PD4 τίθεται σε λογικό 0 για 2usec. Στη συνέχεια ο ακροδέκτης PD4 διατηρείται σε λογικό 0 ή μεταβαίνει σε λογικό 1 ανάλογα με την τιμή του καταχωρητή r24 και περιμένει 58usec μέχρι ο αισθητήρας θερμοκρασίας να διαβάσει αυτή την τιμή. Τέλος, η θύρα PORTD ρυθμίζεται ως είσοδος, δημιουργείται μια ακόμη χρονική καθυστέρηση 1usec για αποκατάσταση και η ρουτίνα επιστρέφει.

one\_wire\_transmit\_bit:

```

push r24                ; save output bit

sbi DDRD, PD4           ; set PD4 as output

cbi PORTD, PD4          ;
ldi r24, 0x02           ; time slot 2 usec
ldi r25, 0x00           ;
rcall wait_usec         ;

pop r24                 ; retrieve output bit

sbrc r24, 0             ;
sbi PORTD, PD4          ; PD4 = r24[0]
sbrs r24, 0             ;
cbi PORTD, PD4          ;

ldi r24, 58             ; wait 58 usec for connected
ldi r25, 0              ; device to sample the line
rcall wait_usec         ;

cbi DDRD, PD4           ; set PD4 as input
cbi PORTD, PD4          ; disable pull-up

ldi r24, 0x01           ;
ldi r25, 0x00           ; recovery time 1 usec
rcall wait_usec         ;

ret

```

Η ρουτίνα **one\_wire\_receive\_byte** έχει ως αντικείμενο τη λήψη ενός byte από τον αισθητήρα DS1820, καλώντας τη ρουτίνα **one\_wire\_receive\_bit** 8 φορές ξεκινώντας από το λιγότερο σημαντικό bit. Το λαμβανόμενο byte αποθηκεύεται στον καταχωρητή **r24**.

```
one_wire_receive_byte:
    ldi r27,8                ; 8 repetitions
    clr r26

loop_:
    rcall one_wire_receive_bit ; r24[0] holds the received bit

    lsr r26

    sbrc r24,0                ; logical or with 0x00 if returned bit is 0,
    ldi r24,0x80                ; or
    or r26,r24                 ; with 0x80 if returned bit is 1

    dec r27
    brne loop_
    mov r24,r26
    ret
```

Η ρουτίνα **one\_wire\_transmit\_byte** έχει ως αντικείμενο την εγγραφή ενός byte στον αισθητήρα DS1820, καλώντας τη **one\_wire\_transmit\_bit** 8 φορές ξεκινώντας από το λιγότερο σημαντικό bit. Το εκπεμπόμενο byte αποθηκεύεται στον καταχωρητή **r24**.

```
one_wire_transmit_byte:
    mov r26,r24
    ldi r27,8                ; 8 repetitions

_one_more_:
    clr r24                    ;
    sbrc r26,0                ; load r24[0] with the transmitted bit
    ldi r24,0x01                ;

    rcall one_wire_transmit_bit

    lsr r26
    dec r27
    brne _one_more_
    ret
```

## Τα ζητούμενα της 7<sup>ης</sup> εργαστηριακής άσκησης

**Ζήτημα 7.1 (α)** Να υλοποιηθεί ένα πρόγραμμα σε γλώσσα C, για την επικοινωνία του μικροελεγκτή ATmega328PB με τον αισθητήρα θερμοκρασίας DS18B20 στην κάρτα *NtuAboard\_G1*.

Το πρόγραμμα να περιλαμβάνει μόνο τις στοιχειώδεις λειτουργίες, λαμβάνοντας υπόψη ότι στον ακροδέκτη PD4 είναι συνδεδεμένος μόνο ένας αισθητήρας. Οι στοιχειώδεις αυτές λειτουργίες στηρίζονται στις εντολές που δέχεται ο αισθητήρας και παρουσιάζονται στο τεχνικό εγχειρίδιου του DS18B20.

Οι εντολές αυτές μπορούν να χωριστούν σε **εντολές μνήμης**, με τις οποίες επιλέγεται η συσκευή με την οποία θα γίνει η επικοινωνία, και **εντολές λειτουργίας**, με τις οποίες υλοποιούνται λειτουργίες που έχουν σχέση με τη **μέτρηση θερμοκρασίας**. Ενδιαμέσως, ο αισθητήρας βρίσκεται σε κατάσταση **χαμηλής κατανάλωσης ισχύος** και επανέρχεται σε κατάσταση λειτουργίας με μια **εντολή αρχικοποίησης**.

Ειδικότερα να γραφεί ρουτίνα που να επιστρέφει στον διπλό καταχωρητή r25:r24 την τιμή της θερμοκρασίας και σε περίπτωση που δεν υπάρχει συνδεδεμένη συσκευή να επιστρέφει την τιμή 0x8000.

Ποιο αναλυτικά η ρουτίνα να περιλαμβάνει τα εξής:

- Αρχικοποίηση και έλεγχος αν υπάρχει συνδεδεμένη συσκευή με την εκτέλεση της ρουτίνας *one\_wire\_reset*.
- Χρήση της ρουτίνας *one\_wire\_transmit\_byte* για αποστολή της εντολής **0xCC**, η οποία παρακάμπτει την επιλογή συσκευής από διάδρομο πολλών συσκευών, μιας και είναι δεδομένο ότι στην κάρτα *NtuAboard\_G1* υπάρχει μόνο μία συνδεδεμένη συσκευή.
- Χρήση της ρουτίνας *one\_wire\_transmit\_byte* για αποστολή της εντολής **0x44**, η οποία ξεκινάει μια μέτρηση θερμοκρασίας. Όταν η μέτρηση θερμοκρασίας ολοκληρωθεί, το DS18B20 σηματοδοτεί τον τερματισμό της μετατροπής με την αποστολή ενός bit με τιμή 1, το οποίο ελέγχεται με την εκτέλεση της ρουτίνας *one\_wire\_receive\_bit*.
- Νέα αρχικοποίηση της συνδεδεμένης συσκευής με την εκτέλεση της ρουτίνας *one\_wire\_reset*.
- Χρήση της ρουτίνας *one\_wire\_transmit\_byte* για αποστολή της εντολής **0xCC**.
- Χρήση της ρουτίνας *one\_wire\_transmit\_byte* για αποστολή της εντολής **0xBE** για ανάγνωση των 16 bit της μετρημένης θερμοκρασίας, η οποία πραγματοποιείται με κλίση της ρουτίνας *one\_wire\_receive\_byte* δύο φορές και αποθήκευση της τιμής στο ζεύγος καταχωρητών r25:r24. Η τιμή που διαβάζεται, σύμφωνα με όσα αναφέρονται στο τεχνικό εγχειρίδιο, είναι ένας αριθμός σε μορφή συμπληρώματος ως προς 2. Τα 8 λιγότερα σημαντικά bit αντιστοιχούν σε θερμοκρασία βαθμών Κελσίου, με **ακρίβεια 0.5°C** ανά bit, και τα 8 περισσότερο σημαντικά αποτελούν επέκταση προσήμου (στην προκαθορισμένη λειτουργία του αισθητήρα). Για παράδειγμα, ο αριθμός 0x0032 αντιστοιχεί σε θερμοκρασία 25°C και ο 0xFFFF σε -0.5°C.
- Η ρουτίνα συμπληρώνεται από μια εντολή που επιστρέφει την τιμή 0x8000 σε περίπτωση που δεν υπάρχει συσκευή συνδεδεμένη στον ακροδέκτη PD4.

Το πρόγραμμα πρέπει να έχει παρόμοια δομή με την assembly, δηλ. οι συναρτήσεις της C να αντιστοιχούν υποχρεωτικά στις ρουτίνες που σας δίνονται.

Αντί για τον αισθητήρα DS18B20 μπορεί να χρησιμοποιηθεί και ο αισθητήρας DS18B20 με τη διαφορά ότι στην προεπιλεγμένη ρύθμιση της λειτουργίας του DS18B20, η τιμή της θερμοκρασίας μετριέται με ακρίβεια 0.0625 °C και δίνεται σε μορφή συμπληρώματος ως προς 2 σύμφωνα με τον παρακάτω πίνακα:

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
<b>LS BYTE</b>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
<b>MS BYTE</b>	S	S	S	S	S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>

S = SIGN

Για παράδειγμα, ο αριθμός 0x0190 αντιστοιχεί σε θερμοκρασία 25°C και ο 0xFFFF σε -0.5°C.

**Ζήτημα 7.2** Γράψτε πρόγραμμα σε C που με την χρήση της προηγούμενης ρουτίνας να απεικονίζει τη θερμοκρασία σε °C στο LCD display σε δεκαδική τιμή τριών ψηφίων με το πρόσημο (**-55°C** έως **+125°C**).

Επίσης στην περίπτωση που δεν υπάρχει συσκευή συνδεδεμένη να εμφανίζει το μήνυμα **“NO Device”**.