

# Μικροεπεξεργαστές και Περιφερειακά

## Εργασία 3<sup>η</sup>: Smart Thermostat

### Authors

Χριστοφορίδης Σάββας AEM: 9147 email: [schristofo@ece.auth.gr](mailto:schristofo@ece.auth.gr)

Πορτοκαλίδης Σταύρος AEM: 9334 email: [stavport@ece.auth.gr](mailto:stavport@ece.auth.gr)

Code: [https://github.com/sportokalidis/stm32-Nucleo\\_Smart\\_Thermostat](https://github.com/sportokalidis/stm32-Nucleo_Smart_Thermostat)

Video: <https://www.dropbox.com/s/4pwgph595wfj2fl/Demo.mp4?dl=0>

### Περιγραφή Προβλήματος

Σκοπός της παρούσας εργασίας είναι η δημιουργία μιας ενσωματωμένης συσκευής, η οποία θα υλοποιεί ένα έξυπνο θερμόμετρο-θερμοστάτη. Για το σκοπό αυτό, θα γίνει χρήση του μικροελεγκτή ARM-M4 σε συνδυασμό με αισθητήρες θερμοκρασίας (DS18B20) και εγγύτητας (HC-SR04).

### Εξήγηση Διαδικασίας

Η υλοποίηση του «έξυπνου θερμοστάτη» βασίζεται στην ανάγνωση της θερμοκρασίας του χώρου με χρήση ενός αισθητήρα θερμοκρασίας. Ο αισθητήρας θερμοκρασίας παίρνει μετρήσεις ανά 5 δευτερόλεπτα και αποθηκεύει τις 24 τελευταίες μετρήσεις, δηλαδή ο κάθε κύκλος μετρήσεων διαρκεί 2 λεπτά. Μετά το πέρας των 2 λεπτών το ενσωματωμένο σύστημα υπολογίζει την μέση τιμή των θερμοκρασιών που έχουν μετρηθεί και την παρουσιάζει σε μια LCD οθόνη με το μήνυμα «Mean Temp is “yy.y”». Το μήνυμα προβάλλεται στην οθόνη για 10 δευτερόλεπτα. Κάθε φορά που θα παρουσιάζεται η μέση τιμή, το ενσωματωμένο σύστημα ξεκινάει εκ νέου ένα κύκλο μετρήσεων, μετρώντας πάλι την θερμοκρασία ανά 5 δευτερόλεπτα.

Εκτός από την παραπάνω λειτουργία, αν η θερμοκρασία του χώρου ανέβει πάνω από μια ορισμένη τιμή (στην υλοποίησή μας 20 °C), τότε το ενσωματωμένο σύστημα ανάβει ένα κόκκινο Led, επίσης, αν η θερμοκρασία πέσει κάτω από μία ορισμένη τιμή (15 °C), τότε το σύστημα ανάβει ένα πράσινο Led. Και στις δύο περιπτώσεις παρουσιάζεται στην LCD οθόνη ανάλογο μήνυμα «Temp > 20» ή «Temp < 15». Επιπλέον, αν η θερμοκρασία ανέβει πάνω από μια ορισμένη τιμή, στην υλοποίησή μας ίδια με την παραπάνω (20 °C), κλείνει ένας διακόπτης για να εκτελεστεί μια διεργασία (π.χ. εκκίνηση λειτουργίας ενός ανεμιστήρα), αυτό γίνεται αισθητό με το άναμμα ενός κίτρινου Led.

Τέλος, ο θερμοστάτης διαθέτει έναν αισθητήρα εγγύτητας με τον οποίο ανιχνεύει αν κάποιος έχει πλησιάσει την συσκευή. Όταν ανιχνεύσει ότι κάποιος έχει πλησιάσει κοντύτερα από μια απόσταση, θα εμφανίζει στην οθόνη την τελευταία μέτρηση της θερμοκρασίας (αυτή που παίρνει κάθε 5 δευτερόλεπτα), καθώς και την τελευταία μέση

τιμή της θερμοκρασίας από το προηγούμενο δίδεπτο. Το μήνυμα που εμφανίζεται στην LCD οθόνη είναι «T:yy.y, A: zz.z».

## Εξήγηση Υλοποίησης

- Pins της πλακέτας Nucleo stm32 F401RE που χρησιμοποιούνται στην υλοποίηση:

Name	Pin	Name	Pin
LCD (DB4)	PB_4	LCD (V <sub>DD</sub> )	5 V
LCD (DB5)	PB_5	LCD (V <sub>SS</sub> )	GND
LCD (DB6)	PB_3	Red Led	PA_5
LCD (DB7)	PA_10	Green Led	PA_6
LCD (R/W)	PA_8	Yellow Led	PA_7
LCD (R <sub>s</sub> )	PA_9	DS18B20 (DQ)	PA_0
LCD (K)	GND	HC-SR04 (echo)	PB_9
LCD (A)	3.3 V	HC-SR04 (Trig)	PB_8
LCD (E)	PB_10	HC-SR04 (V <sub>cc</sub> )	5 V
LCD (V <sub>o</sub> )	GND	HC-SR04 (GND)	GND

- Driver για αισθητήρα θερμοκρασίας DS18B20

Για την μέτρηση της θερμοκρασίας του χώρου με τον αισθητήρα βασιστήκαμε στο manual του DS18B20 [1]:

### Initialization

Για να λειτουργήσει ο αισθητήρας πρέπει να τον αρχικοποιούμε κάθε φορά. Σύμφωνα με το manual, η αρχικοποίηση του αισθητήρα επιτυγχάνεται θέτοντα το data pin (DQ) low για 480usec και στην συνέχεια διαβάζουμε το pin αυτό για να δούμε αν απάντησε ο αισθητήρας με παλμό. Η ρουτίνα που υλοποιεί την αρχικοποίηση είναι η `temperature_start()`.

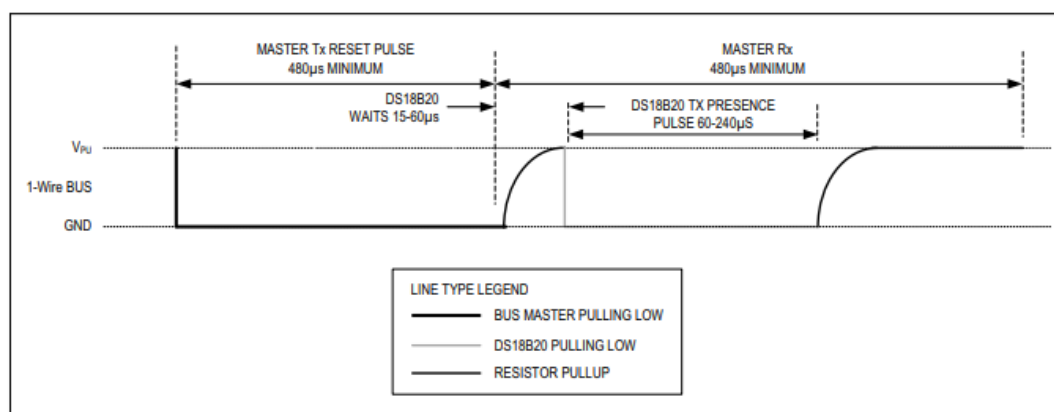
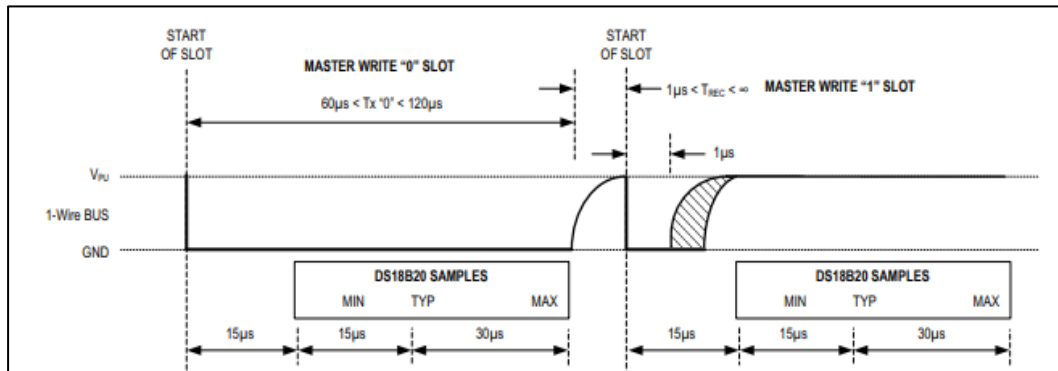


Figure 15. Initialization Timing

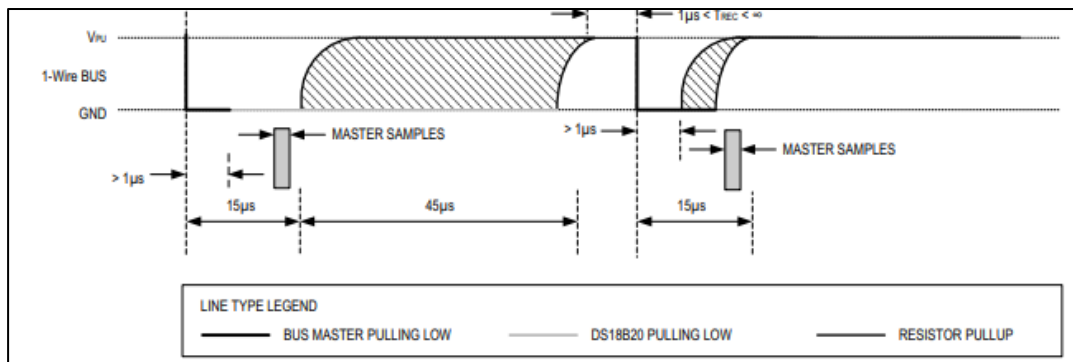
### Write

Για να γράψουμε 1 στον αισθητήρα (δηλαδή ο master να γράψει 1 στο bus προς τον slave), πρέπει αφού κρατήσουμε low το pin για 1usec, να το «απελευθερώσουμε» εντός 15 usec. Ενώ για να γράψουμε 0 στο πρέπει να κρατήσουμε το pin low για τουλάχιστον 60 usec. Η ρουτίνα που υλοποιεί το write στον αισθητήρα είναι η `temperature_write(uint8_t data)`. Το data είναι μια 8-bit λέξη, της οποίας τα bit περνούν από μια μάσκα και στέλνονται ένα προς ένα στον αισθητήρα.



### Read

Για να ξεκινήσει η διαδικασία του reading πρέπει ο master να κρατήσει το pin low για τουλάχιστον 1 usec. Μετά από αυτό, ο slave αρχίζει και στέλνει bits μέσω του bus. Στέλνει 1 αν έχει το pin high, ενώ στέλνει 0 αν το κρατάει low. Η ρουτίνα που υλοποιεί το read είναι η `temperature_read()`. Η οποία επιστρέφει μια 8-bit λέξη, που είναι αυτή που σχηματίστηκε από τα bit που έστειλε ο slave.

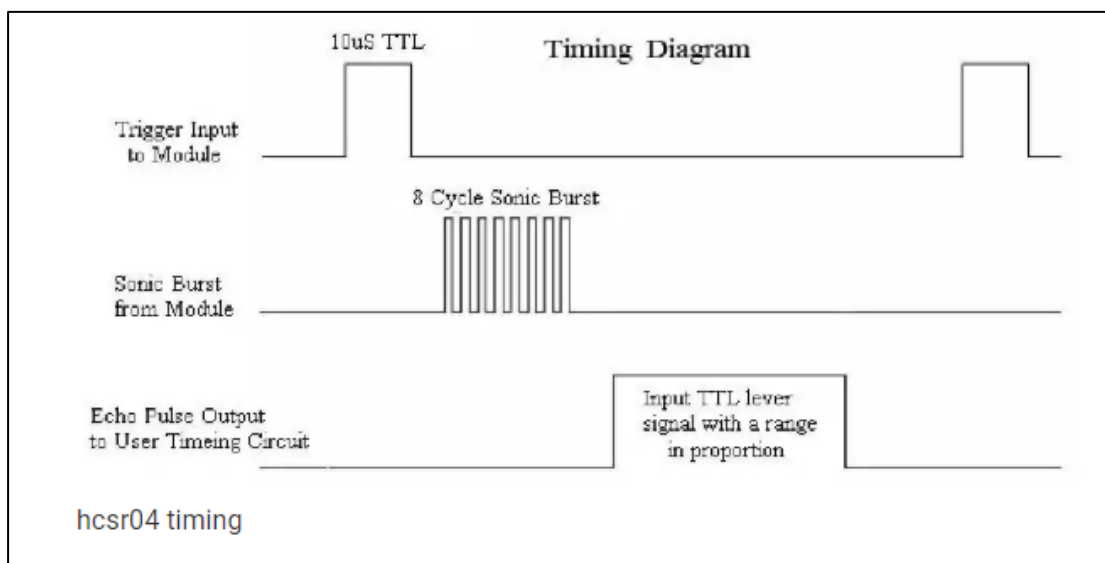


### Result

Τέλος, υπάρχει η μέθοδος `result`, η οποία είναι αυτή που βάζει σε σωστή σειρά όλες τις μεθόδους με τα κατάλληλα delays, σύμφωνα πάλι με το manual, ώστε να πραγματοποιηθεί η επικοινωνία με τον αισθητήρα. Στο τέλος, επιστρέφει ένα float που είναι η τιμή της θερμοκρασίας σε βαθμούς Κελσίου.

- Driver για αισθητήρα εγγύτητας HC-SR04

Στην περίπτωση του αισθητήρα εγγύτητας, η διαδικασία είναι πιο απλή. Για την ενεργοποίησή του χρειάζεται να θέσουμε το trig pin high για 10 usec. Μετά από αυτό ο αισθητήρας μας απαντάει στο echo pin με ένα παλμό. Εμείς αρκεί να μετρήσουμε το πλάτος του παλμού που μας έστειλε, ο οποίος είναι ανάλογο της απόστασης του αντικειμένου που βρίσκεται μπροστά. Χρησιμοποιώντας την σχέση: **distance = sensor\_time \* 0.034/2**. Παίρνουμε την μέτρηση για την απόσταση. Η ρουτίνα που υλοποιεί αυτή τη διαδικασία για την επικοινωνία με τον αισθητήρα εγγύτητας είναι η distance\_read(). Η distance\_read() επιστρέφει ένα 16-bit uint, που είναι η ζητούμενη απόσταση του αντικειμένου.



- Υλοποίηση main() function

Αρχικά, στη main() function γίνονται όλες οι απαραίτητες αρχικοποιήσεις και δηλώσεις μεταβλητών. Μέσα στις αρχικοποιήσεις γίνεται initialize και enable και ένας interrupt timer, ο οποίος δημιουργεί ένα Interrupt κάθε 1 δευτερόλεπτο. Η ρουτίνα η οποία εκτελείται όταν «σκάει» ένα interrupt είναι η timer\_isr(), η οποία με την χρήση counter, είναι υπεύθυνη να μετράει χρόνο πέντε, δέκα και εκατόν είκοσι δευτερολέπτων. Κάθε φορά που το χρονικό διάστημα περνάει, η ρουτίνα μεταβάλλει το αντίστοιχο flag (five\_sec\_flag, ten\_sec\_flag, two\_min\_flag). Στη συνέχεια, στην main() function υπάρχει μια while(1) loop, μέσα στην οποία γίνεται ο έλεγχος των παραπάνω flags. Αν το five\_sec\_flag == 1, το ενσωματωμένο σύστημα παίρνει μια μέτρηση θερμοκρασίας και την αποθηκεύει. Έπειτα, ελέγχει αν την τιμή της θερμοκρασίας και αν είναι απαραίτητο εκτελεί την κατάλληλη λειτουργία. Αν το two\_min\_flag == 1, τότε, το ενσωματωμένο σύστημα υπολογίζει την μέση τιμή των μετρήσεων, στα δύο λεπτά που πέρασαν και εμφανίζει ένα μήνυμα στην οθόνη για 10 δευτερόλεπτα. Τέλος, υπάρχει και ένας έλεγχος για τον αισθητήρα εγγύτητας. Αν ο αισθητήρας εντοπίσει αντικείμενο, σε μια απόσταση μικρότερη μιας δεδομένης τιμής (που ορίζεται στα #DEF), τότε το ενσωματωμένο σύστημα εμφανίζει στην οθόνη LCD το κατάλληλο μήνυμα.

## Sources

- [1] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [2] <http://web.eece.maine.edu/~zhu/book/lab/HC-SR04%20User%20Manual.pdf>
- [3] <https://controllerstech.com/ds18b20-and-stm32/>
- [4] <https://www.embedded-computing.com/articles/measuring-code-execution-time-on-arm-cortex-m-mcus>