



Μικροεπεξεργαστές και περιφερειακά: Thermostat

Φοιτητές: Εμμανουήλ Μιχάλαϊνας (9070)

michalain@ece.auth.gr

Παναγιώτης Καρελής (9099)

karelisp@ece.auth.gr

Καθηγητής: Ιωάννης Παπαευσταθίου

Ημερομηνία: 27 Ιουνίου 2020

Περίληψη: Η παρούσα αναφορά παρουσιάζει τα σχόλια και τα συμπεράσματα που προέκυψαν από την εκπόνηση της 3^{ης} εργασίας του μαθήματος “Μικροεπεξεργαστές και περιφερειακά” - Τομέας Ηλεκτρονικής – Υπολογιστών – 8ο εξάμηνο. Ζητείται η ανάπτυξη ενός προγράμματος για την πλακέτα STM32F401RE, με χρήση του keil μVision, με σκοπό τον έλεγχο ενός ψηφιακού θερμομέτρου, ενός αισθητήρα εγγύτητας και μιας lcd οθόνης.

Λέξεις κλειδιά: Μικροεπεξεργαστές και περιφερειακά, ARM, Cortex-M4, Keil uVision, STM32F401RE, αισθητήρας θερμότητας, αισθητήρας εγγύτητας, lcd οθόνη, GPIO.

1 ΑΙΣΘΗΤΗΡΑΣ ΕΓΓΥΤΗΤΑΣ

Η επικοινωνία με τον αισθητήρα εγγύτητας είναι αρκετά απλή. Αρχικά, ρυθμίζονται οι ακροδέκτες μέσω των οποίων τον συνδέουμε στην πλακέτα (GPIO – General Purpose Input Output). Η λήψη μιας μέτρησης απόστασης γίνεται μέσω της συνάρτησης `get_distance()`, βλέπε `proximity.c` και `proximity.h`. Για να ληφθεί μέτρηση, στέλνεται ένας λογικός παλμός 1 διάρκειας 10μs. Στη συνέχεια αναμένουμε λογικό παλμό 1 στον ακροδέκτη “Echo” τον οποίο χρονομετρούμε μέσω διαδοχικών καθυστερήσεων `delay_cycles(960)`. Η χρονομέτρηση γίνεται δηλαδή όπως και στην δεύτερη εργασία, που μετριοούνταν οι χρόνοι αντανakλαστικών του χρήστη. Με κατάλληλη μετατροπή παίρνουμε την απόσταση που μετρήθηκε με σφάλμα περίπου 10mm.

2 ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜΟΤΗΤΑΣ

Για την επικοινωνία με το θερμομέτρο απαιτείται η υλοποίηση του πρωτοκόλλου 1-wire™. Η υλοποίηση έγινε χρησιμοποιώντας την USART6 της πλακέτας σε half-duplex mode. Η ευελιξία στην επιλογή του ρυθμού επικοινωνίας (baud rate) που παρέχει η UART, επιτρέπει την απευθείας υλοποίηση των χρονοθυρίδων (time-slots) του 1-wire™. Σε όλα τα παρακάτω, χρησιμοποιούνται 8bit, no parity, 1stop bit, no flow-control UART.

Πιο συγκεκριμένα, το πρωτόκολλο αποτελείται από 3 στάδια:

- Στάδιο 1. Initialization
- Στάδιο 2. ROM Command
- Στάδιο 3. DS18B20 Function Command

Για το Στάδιο 1 ο microcontroller χρειάζεται να στείλει έναν reset pulse και να ανιχνεύσει έναν presence pulse. Με τη χρήση baud rate 9600, αποστέλλοντας το byte 0xF0 δημιουργείται ένας κατάλληλος reset pulse. Επειδή στην 1-wire γραμμή είναι συνδεδεμένη μόνο μια συσκευή, λαμβάνουμε σωστή την τιμή 0xEE ως presence pulse.

Για τα Στάδια 1 και 2 γίνεται χρήση baud rate 115200. Στα στάδια αυτά απαιτείται η δημιουργία bit read/write χρονοθυρίδων. Για ένα read time-slot αποστέλλεται το σύμβολο 0xFF κι έπειτα στη UART λαμβάνεται από το θερμόμετρο μια byte τιμή της οποίας το LSB είναι το bit που επέστρεψε το θερμόμετρο. Σε ένα write time-slot, για την αποστολή του bit 1, αποστέλλουμε μέσω UART το byte 0xFF, ενώ για το bit 0, αποστέλλουμε το byte 0x00.

Όλες οι παραπάνω λειτουργίες επιτυγχάνονται με τις συναρτήσεις onewire_transaction_init(), onewire_read(), onewire_write(). Επιπλέον υλοποιούνται και οι onewire_read_byte() και onewire_write_byte() οι οποίες απλά καλούν τις αντίστοιχες τους 8 φορές.

Πηγή: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/2/214.html>

3 ΟΘΟΝΗ ΚΑΙ LED

Για τον έλεγχο της οθόνης και των led χρησιμοποιούμε τις έτοιμες βιβλιοθήκες που δίνονται.

4 ΠΡΟΣΕΓΓΙΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η λογική που ακολουθούμε αποτυπώνεται στο αρχείο main.c. Αρχικά ορίζουμε macro για τον έλεγχο της συμπεριφοράς του συστήματος. Ρυθμίζουμε έναν χρονιστή στο 1 δευτερόλεπτο, χρησιμοποιώντας την έτοιμη βιβλιοθήκη timer.c και timer.h. Η συνάρτηση χειρισμού διακοπών αυξάνει κατάλληλα μια μεταβλητή που παίρνει τιμές από 0 έως 120 (δευτερόλεπτα).

Το σύστημα, ξυπνάει κάθε ένα δευτερόλεπτο και κάνει τις παρακάτω διεργασίες. Αρχικά, παίρνει μέτρηση θερμοκρασίας αν το τρέχον δευτερόλεπτο είναι πολλαπλάσιο του πέντε και εκχωρεί τις κατάλληλες τιμές στις μεταβλητές average και prev_average κάθε δύο λεπτά. Έπειτα, σετάρει το κόκκινο και μπλε LED ανάλογα με τη πιο πρόσφατη μέτρηση θερμοκρασίας. Το πράσινο LED (το ρελέ), ρυθμίζεται με βάση τη μέση θερμοκρασία. Στη συνέχεια κάνει μέτρηση της απόστασης μέσω του αισθητήρα εγγύτητας. Τελευταία διεργασία είναι η εκτύπωση των κατάλληλων μηνυμάτων στην LCD οθόνη αν χρειάζεται. Μετά, ο microcontroller μπαίνει σε low power mode μέχρι την έναρξη του επόμενου δευτερολέπτου.

Η εκτύπωση των μηνυμάτων στην LCD γίνεται με την εξής προτεραιότητα:

High/Low Temperature msg > Human nearby msg > Average Temperature msg (για 10s κάθε 2min).

Προφανώς, το μήνυμα High/Low Temperature μένει στην οθόνη μέχρις ότου η θερμοκρασία είναι εντός των οριακών τιμών. Επίσης αν μέσα στο χρονικό παράθυρο των 10s (για το Average Temperature msg) εμφανιστεί άνθρωπος σε κοντινή απόσταση, κι αυτός φύγει πριν την λήξη του παραθύρου, το καθιερωμένο μήνυμα θα εμφανιστεί ξανά μέχρι τη λήξη των 10s.

Η επιλογή του ενός δευτερολέπτου ώστε να ξυπνάει το σύστημα είναι ικανοποιητική και ενεργειακά, αλλά και από πλευράς γρήγορης απόκρισης συστήματος, όταν πλησιάσει άνθρωπος, σύμφωνα με τα αποτελέσματα που πήραμε από τη δεύτερη εργασία για τα ανθρώπινα αντανakλαστικά. Στη χειρότερη περίπτωση ο χρόνος απόκρισης είναι 1s ενώ στη μέση είναι 0.5s.

5 ΠΡΟΒΛΗΜΑΤΑ ΠΟΥ ΑΝΤΙΜΕΤΩΠΙΣΤΗΚΑΝ

Το πρώτο πρόβλημα που αντιμετωπίσαμε ήταν η σωστή επιλογή κβάντου κύκλων για χρονομέτρηση του παλμού που επιστρέφει ο αισθητήρας εγγύτητας. Οι συναρτήσεις delay εισάγουν επιπρόσθετους κύκλους που δεν υπολογίζονται, οδηγώντας σε χαμηλότερες μετρήσεις από τις πραγματικές. Επιλογή λιγότερων κύκλων καθυστέρησης σημαίνει καλύτερη ακρίβεια λόγω κβάντισης, αλλά χαμηλότερη ακρίβεια λόγω υψηλού επιπλέον φόρτου. Επιλέξαμε να καθυστερούμε 960 κύκλους, που αντιστοιχούν σε 60μs για τα 16MHz του ρολογιού της πλακέτας.

Ένα πολύ σημαντικό πρόβλημα που αντιμετωπίσαμε ήταν η χρήση του πρωτοκόλλου 1-wire™. Οι έτοιμες βιβλιοθήκες `uart.c`, `uart.h` εξυπηρετούν full duplex επικοινωνίες μέσω του περιφερειακού USART2, ωστόσο το θερμόμετρο απαιτεί επικοινωνία half duplex. Με μια μικρή τροποποίηση (ενός bit στον κατάλληλο καταχωρητή) αυτό μπορεί να επιτευχθεί. Ωστόσο η πλακέτα επικοινωνεί με τον υπολογιστή μέσω usb, πάλι μέσω του USART2, οπότε υπάρχει διένεξη που αποτρέπει την σωστή επικοινωνία. Η διένεξη (conflict) αυτή, προκύπτει γιατί δύο συσκευές είναι συνδεδεμένες στο άκρο Rx της UART, ο υπολογιστής και το θερμόμετρο. Όταν το θερμόμετρο προσπαθεί να οδηγήσει τη γραμμή στο low, δεν την αφήνει το usb γιατί αυτό είναι συνέχεια στο high (καθώς δεν στέλνει δεδομένα μέσω της serial port). Η λύση μας ήταν να γράψουμε δική μας συνάρτηση για να χρησιμοποιήσουμε το περιφερειακό USART6, ακολουθώντας το reference manual της πλακέτας.