

Εργασία 3η

Γιώργος Σελιβάνωφ, Λαμπρινός Χατζηγιάννου

June 7, 2024

1 Γενικά

Η δομή του προγράμματος παρουσιάζει αρκετές ομοιότητες με εκείνη της δεύτερης εργασίας. Οι κυριότερες διαφορές είναι η εισαγωγή δεύτερου timer (`tim.h`), ο οποίος, με δομή παρόμοια με τον δοθέντα timer χρησιμοποιείται αντί `delay` για τον ρυθμό δειγματοληψίας, και η ανάπτυξη `drivers` για τα εμπλεκόμενα περιφερειακά.

2 DHT: Θερμοκρασία και υγρασία

Για τον έλεγχο του χρήστη, όπως άλλωστε απαιτούνταν, γράφτηκαν οι κατάλληλοι `drivers` (`dht.c/h`). Ουσιαστικά, ακολουθώντας τα αρχεία προδιαγραφών, και μεριμνώντας για τις συνθήκες ελαττωματικών μετρήσεων, γίνεται η ερμηνεία των δεδομένων που εισέρχονται στο Pin, και αποθηκεύονται στην επιθυμητή μορφή.

3 ISRs

3.1 TouchSensor ISR

Ρύθμιση του ISR ούτως ώστε η κλήση του να συνεπάγεται (ανάλογα το πλήθος των προηγούμενων πατημάτων)

- την αρχική συμπεριφορά
- την συμπεριφορά για ζυγό αριθμό "κλήσεων"
- την συμπεριφορά για μονό αριθμό κλήσεων.

3.2 Temperature (Timer) ISR

Από την εκφώνηση ζητείται η επανειλημμένη (σε μεταβλητά κατά την διάρκεια του προγράμματος διαστήματα) μέτρηση θερμοκρασίας (ή και υγρασίας). Αυτό, το πετύχαμε με την χρήση του δικού μας *δευτερεύοντος* timer, ο οποίος είχε δυνατότητα να προκαλεί ISR μέχρι και 18 δευτερόλεπτα (που είναι άλλωστε το ανώτατο δυνατό όριο για τις προδιαγραφές του συστήματος).

3.3 Blink ISR

Ακόμα μία εξεταζόμενη περίπτωση είναι η συμπεριφορά του LED ανάλογα με τις τιμές της θερμοκρασίας. Ο έλεγχος, λαμβάνει χώρα κατά την διάρκεια του check message, και ανάλογα με τις υπάρχουσες τιμές επιδρά επί του *πρωτεύοντος timer* (από τους driver του εργαστηρίου). Στην περίπτωση που είναι ενεργοποιημένος, η `blink_isr` είναι υπεύθυνη για την περιοδική αλλαγή των καταστάσεων του LED.

4 Εκτύπωση

Κρατήθηκε παρόμοια μεθοδολογία με ύπαρξη `msg_queue`, προς αποφυγή της χρήσης εκτυπώσεων εντός των ISR. Μια μικρή λεπτομέρεια στο συγκεκριμένο είναι πως για την “καλύτερη” λειτουργία βάσει των προδιαγραφών σχηματίστηκε και `float_queue`.

```
1 while(1)
2 {
3     if(!checkMessg()) {
4         __WFI();
5     }
6 }
```

Ακόμα **σημαντικό** είναι να τονιστεί πως αποφασίστηκε η εξής “απλοποίηση” για λόγους μνήμης/ταχύτητας: Χρησιμοποιείται το ίδιο Queue τόσο για θερμοκρασία, υγρασία αλλά και ρυθμό δειγματοληψίας. Για την διασφάλιση της κατάλληλης ερμηνείας μεταξύ των, δόθηκε ξεχωριστό value range, στο καθένα από αυτά μήκους 255, καθώς οι τιμές τους εκ φύσεως δεν μπορούν να είναι μεγαλύτερες από την μέγιστη αποθηκεύσιμη σε 8 bit. Κατά την εκτύπωση γίνεται ο κατάλληλος διαχωρισμός/ερμηνεία:

```
1 if(msg < 255) {
2     // Temperature logic
3     ...
4     sprintf(buffer, "Temperature: %f C.\r\n", (double)msg);
5 } else if (msg < 510) {
6     // Humidity
7     sprintf(buffer, "Humidity: %f %%\r\n", (double)(msg-255));
8 } else {
9     // Sensor rate
10    sprintf(buffer, "Sensor Rate: %f s .\r\n", (double)(msg-510));
11 }
```
