Μικροεπεξεργαστές και περιφερειακά, 2η εργαστηριακή εργασία.

Λαδιάς Νικόλαος
Τριφηνόπουλος Χρήστος

Για την συγκεκριμένη εργασία υλοποιήσαμε 3 interrupt service routines (ISR):

1. **uart_rx_isr:** Καλείται κάθε φορά που ο χρήστης καταχώρει κάποιο γράμμα του επιθέτου του ή πατάει τα πλήκτρα backspace/enter. Αποθηκέυει τα δεδομένα που λαμβάνει από τον χρήστη σε μία ουρά. (queue)

```
// ISR for character receive (receiving char is in argument)
void uart_rx_isr(unsigned char rx) {
  if ((rx >= 'a' && rx <= 'z') || (rx >= 'A' && rx <= 'Z') || rx==0x7F || rx==0x0A ) {
       queue_enqueue(&rx_queue, rx); // Store the received character
  }
}
```

2. **timer_callback_isr:** Είναι ένας timer ο οποίος καλείται κάθε 100 κύκλους ρολογιού. Αν η μεταβλητή lightsOn είναι 1, δηλαδή ο χρήστης έχει πατήσει enter, ανάβει το led σε πράσινο ή κόκκινο χρώμα, ανάλογα με την κατάσταση της μεταβλητής vowel. Στην συνέχεια αλλάζει την τιμή της μεταβλητής lightsOn ξανά σε 0.
3. **button_press_isr:** Καλείται κάθε φορά που ο χρήστης πατάει το κουμπί και αλλάζει το χρώμα του led σε μπλε. Αν είναι ήδη μπλε, το σβήνει.

Στην main αρχικά υλοποιούμε τις κατάλληλες αρχικοποιήσεις των μεταβλητών και των interrupt service routines. Αφού αρχικοποιήσουμε και την επικοινωνία μέσω uart protocol καθώς και την ουρά στην οποία εισάγουμε τα γράμματα του εισαγόμενου από τον χρήστη επιθέτου, αρχικοποιούμε και κατάλληλα τα LEDs του nucleo και τον timer. Εκτός από τον ορισμό των callbacks χρησιμοποιούμε και την **__enable_irq()** για να ενεργοποιήσουμε τα interrupts. Στην συνέχεια ρυθμίζουμε το κουμπί ώστε να ενεργοποιεί την αντίστοιχη ISR με το πάτημα(**gpio_set_trigger(P_SW, Rising)**) . Να σημειωθεί πως το mode της ενεργοποίησης του κουμπιού είναι Falling, καθώς αυτό βρίσκεται by default σε κατάσταση 1**(gpio_set_mode(P_SW, PullUp)).**

Στην συνέχεια έχουμε δύο εμφωλευμένες while loops:

- Η εξωτερική, με κάθε προσπέλασή της, υλοποιεί ολόκληρη την διαδικασία κατά την οποία το σύστημα ζητάει από τον χρήστη το επίθετο του και αυτός το γράφει και πατάει enter. Τρέχει συνεχώς και δεν τερματίζει ποτέ.
- Η εσωτερική με κάθε προσπέλασή της διαχειρίζεται ένα input από τον χρήστη και τερματίζει όταν αυτός πατήσει το πλήκτρο “enter”.

Υπάρχει και μία Τρίτη while η οποία υλοποιεί την αναμονή για user input με την __WFI.

Να σημειωθεί πως οι επιτρεπόμενοι εισαγόμενοι χαρακτήρες που εισάγονται στην ουρά όταν καλείται η ISR είναι λατινικοί πεζοί και κεφαλαίοι.

Στην επαναληπτική διαδικασία της εσωτερικής λούπας, ελέγχεται αν η ουρά στην οποία εισάγονται οι χαρακτήρες είναι άδεια. Αν αφαιρώντας χαρακτήρα η συνάρτηση της ουράς dequeue επιστρέψει 0, τότε είναι άδεια και με κατάλληλο έλεγχο κάθε φορά που είναι άδεια μέσα στην εσωτερική while, περιμένουμε για interrupt από το user input του χρήστη , με την εμφωλευμένη αντίστοιχη αυτή while.

Στην main, ελέγχεται αν ο εισαγόμενος χαρακτήρας είναι backspace (δηλαδή διαγραφή του χαρακτήρα που εισήχθη), αν ναι , τότε το index της ουράς μεταφέρεται μια θέση πίσω, ενώ στην επόμενη εισαγωγή θα γίνει overwrite από τον χαρακτήρα που εισάγεται. Για τον χαρακτήρα enter, η μεταβλητή lightsOn γίνεται 1. Επιπλέον, αν ο τελευταίος χαρακτήρας ήταν φωνήεν στέλνουμε μέσω UART ένα μήνυμα στον χρήστη πως άναψε το κόκκινο LED, ενώ αν ήταν σύμφωνο το πράσινο , τέλος σπάμε την εσωτερική λούπα και ξεκινάμε πάλι την διαδικασία εισαγωγής από τον χρήστη μετά από 1 δευτερόλεπτο .

Είναι σημαντικό να συνειδητοποιήσει κανείς πως η ταχύτητα ελέγχου της εισαγωγής του Enter άρα και της ενεργοποίησης του αντίστοιχου LED, είναι η συχνότητα που καλείται η **timer_callback_isr(void),** δηλαδή ανά **100 κύκλους ρολογιού.**

Για οποιονδήποτε άλλο χαρακτήρα που εισάγεται προχωράει ο index κατά ένα και σώζεται σε τοπική μεταβλητή για να ελεγχθεί αν είναι σύμφωνο ή φωνήεν . Έτσι ώστε να ανάψει το αντίστοιχο LED με βάση τις οδηγίες της άσκησης. Για τον έλεγχο του φωνήεν ή σύμφωνο , μια μεταβλητή vowel γίνεται ίση με ένα αν ο εισαγόμενος χαρακτήρας είναι ένας από όλους τους πιθανούς των φωνήεν (κεφαλαίος ή πεζός), αυτό ελέγχεται με ένα απλό OR assignment με τους πιθανούς χαρακτήρες.

Διαδικασία Testing

Για την δοκιμή στο εργαστήριο, αφού έγιναν μικροδιορθώσεις ο κώδικας έτρεξε και τύπωνε κανονικά τις ανάλογες τιμές στη σειριακή θύρα , έστελνε R ή G αντίστοιχα με χρήση του UART, για το κόκκινο led ή το πράσινο. Επίσης για κάθε πάτημα του κουμπιού τύπωνε “Blue on” και για κάθε δεύτερο πάτημα “Blue off”. Η σειριακή επικοινωνία καθώς και το πάτημα έτσι επιβεβαιώθηκαν σε πραγματικό χρόνο, καθώς και το κάλεσμα των Interrupt Service Routines για την διεκπεραίωση των ζητούμενων.

Κατά την διάρκεια του εργαστηρίου αλλάξαμε κάποια κομμάτια του κώδικα για να τρέχει σωστά πάνω στην πλακέτα. Πχ είχαμε γράψει **gpio_set_trigger(P_SW, Falling)** αρχικά, αλλά τρέχοντας τον κώδικα στην πλακέτα καταλάβαμε ότι δεν «έπιανε» όλα τα πατήματα του κουμπιού και το αλλάξαμε σε **Rising.** Σε περίπτωση που δεν τρέχει ο παρόν κώδικας σε πλακέτα θα πρέπει να αντικατασταθεί ο δεκαεξαδικός χαρακτήρας του new line (0x0A) με τον χαρακτήρα \n. Αν δεν τρέχει ακόμα παρόμοια αλλαγή πρέπει να γίνει και για το backspace.