

ΕΡΓΑΣΤΗΡΙΟ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΝΑΦΟΡΑ 2ΟΥ ΕΡΓΑΣΤΗΡΙΟΥ

ΠΑΠΑΔΟΠΟΥΛΟΣ ΠΑΝΑΓΙΩΤΗΣ | ΑΛΕΞΑΝΔΡΟΣ ΚΑΡΓΙΩΤΗΣ

Εισαγωγή

Η παρούσα αναφορά έχει ως στόχο την υλοποίηση ενός προγράμματος που διαβάζει ακέραιους αριθμούς μέσω UART, αναλύει τα ψηφία τους μέσω timer ISR ανά 0.5 δευτερόλεπτο και πραγματοποιεί αλλάζει το ενσωματωμένο LED ανάλογα το ψηφίο. Παράλληλα, το κουμπί του nucleo μπορεί να επηρεάσει τη συμπεριφορά του LED κατά την ανάλυση.

Δομή και υλοποίηση

Το πρόγραμμα αναπτύχθηκε εξολοκλήρου σε γλώσσα C, αξιοποιώντας τις βιβλιοθήκες για διαχείριση UART, timer, διακοπών και GPIO που μας δόθηκαν στο πλαίσιο του μαθήματος.

1. **UART** για λήψη και προβολή δεδομένων από τον χρήστη.
2. **Timer**: Ένας ενιαίος timer λειτουργεί με περίοδο 100ms. Μετρά εσωτερικά τα ticks και ενεργοποιεί την επεξεργασία νέου ψηφίου κάθε 500ms.
3. **Blink χωρίς delay**: Για ζυγά ψηφία ενεργοποιείται non blocking λογική. Το LED ανάβει και μετά από 2 ticks (200ms) σβήνει. Ο κύκλος ολοκληρώνεται στα 400ms.
4. **Διακοπές EXTI** για ανίχνευση πατήματος κουμπιού μέσω του PC_13.
5. **EXTI Interrupt**: Το κουμπί του Nucleo (PC_13) ενεργοποιεί διακοπή. Αν το LED είναι ενεργό, «παγώνει» την κατάσταση, διατηρώντας την τρέχουσα φωτεινότητα μέχρι νέα εντολή.
6. **Input Buffering**: Υλοποιήθηκε staging buffer για συνεχή καταχώριση χαρακτήρων, υποστηρίζοντας σβήσιμο με backspace και ανανέωση input με Enter.
7. **Ανάλυση αριθμού**: Η λογική ανάλυσης επιτρέπει:
 - toggle για μονά ψηφία
 - blink για ζυγά
 - επανάληψη αν ο αριθμός τελειώνει σε -
 - άμεση διακοπή της προηγούμενης ανάλυσης με νέο input
8. **Προτεραιότητα διακοπών**: Ορίστηκε προτεραιότητα EXTI πάνω από τη UART με `NVIC_SetPriority`, σύμφωνα με την εκφώνηση.

Προβλήματα

- Τα drivers που μας δόθηκαν στο πλαίσιο του μαθήματος δεν ήταν ευρέως διαδεδομένα, έτσι ήταν δύσκολο να βρούμε ανάλογο κώδικα και έπρεπε να γίνει αρκετό testing πειραματικά για να εξακριβωθεί η ακριβής λειτουργία ορισμένων functions.
- Ήταν δύσκολη η υλοποίηση blinking LED με την χρήση μόνο του SysTick όπως χρησιμοποιείται στους drivers. Έγινε προσπάθεια για χρήση TIMx Timers αλλά λόγω δυσκολιών στην υλοποίηση εν τέλει επιλέχθηκε η υλοποίηση που περιγράφηκε παραπάνω.
- Η ανάγκη για διακοπή του processing των αριθμών με το που βάζουμε νέο input μας ταλαιπώρησε στην υλοποίηση, κυρίως ως προς την διαχείριση του timer και των buffers

Testing

Αρχικά δοκιμάστηκαν τα inputs που δίνονται στο task:

Input: 52719

Digit 5 -> Toggle LED

Digit 2 -> Blink LED

Digit 7 -> Toggle LED

Interrupt: Button pressed. LED locked. Count = 1

Digit 1 -> Skipped LED action Digit 9 -> Skipped LED action

Interrupt: Button pressed. LED unlocked. Count = 2

End of sequence. Waiting for new number...

Input: 81-

Digit 8 -> Blink LED

Digit 1 -> Toggle LED

Digit 8 -> Blink LED

Digit 1 -> Toggle LED

Input: 42

Digit 4 -> Blink LED

Digit 2 -> Blink LED

End of sequence. Waiting for new number...

Έτσι επαληθεύτηκε η σωστή λειτουργία του LED, το interrupt από το button, το ότι το '-' εκτυπώνει μέσω UART επαναλαμβανόμενα τον πιο πρόσφατο αριθμό και το ότι η ανάλυση ενός αριθμού σταματά όταν εισάγουμε έναν νέο.

Έπειτα, έγινε έλεγχος ότι ο κώδικας λειτουργεί σωστά και για corner cases, πχ εισαγωγή και διαγραφή του '-' πριν πατήσω Enter, εισαγωγή μόνο του '-', εισαγωγή γραμμάτων κλπ. Τέλος επιβεβαιώθηκε ότι το κουμπί δεν κάνει "bounce".