

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.

Κύρια λειτουργικά μέρη:

1. Λήψη αριθμού μέσω UART

Ο χρήστης εισάγει έναν αριθμό ο οποίος αποθηκεύεται δυναμικά σε buffer. Η εισαγωγή επιβεβαιώνεται με το Enter (\n). Ο χαρακτήρας - στο τέλος σημαίνει επανάληψη του ίδιου αριθμού.

2. Ανάλυση ψηφίων με Timer Interrupt

Με κάθε χρονική διακοπή (κάθε 0.5 sec) γίνεται επεξεργασία ενός μόνο ψηφίου:

- **Μονός αριθμός:** το LED εναλλάσσει κατάσταση (toggle).
- **Ζυγός αριθμός:** το LED αναβοσβήνει (200ms ON - 200ms OFF – 100ms ON).

3. Διακοπή με Button (PC13)

Όταν πατηθεί το κουμπί:

- Το LED «κλειδώνει» και δεν αντιδρά στα ψηφία.
- Οι ενέργειες συνεχίζουν να καταγράφονται μέσω UART.
- Με δεύτερο πάτημα, η λειτουργία επανέρχεται.
- Το κουμπί έχει προτεραιότητα έναντι του UART input.

4. Διαχείριση πολλαπλών εισόδων

Αν ο χρήστης εισάγει νέο αριθμό κατά τη διάρκεια επεξεργασίας, η τρέχουσα διαδικασία διακόπτεται άμεσα και αρχίζει νέα ανάλυση.

5. Άλλα στοιχεία

- Διαχείριση backspace, overflow του buffer και έλεγχος έγκυρων χαρακτήρων.

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 το blinking στην ίδια ISR με την χρήση delay
- Εγινε προσπάθεια για υλοποίηση 2^{ου} timer τεχνητά, μέσω flag που καλούσε μια άλλη ISR κάθε 200ms. Η υλοποίηση αυτή εγκαταλείφθηκε αφού ήταν αισθητά πιο αργή.
- 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".