

Μικροεπεξεργαστές και Περιφερειακά 2

Panagiotis Koutris
10671

Christos Alexopoulos
10618

May 19, 2025

Περιγραφή και Υλοποίηση Εργασίας

Η τελική έκδοση της εργασίας υλοποιεί ένα διαδραστικό πρόγραμμα σε γλώσσα C που αναλύει αριθμούς μέσω UART, χρησιμοποιεί χρονόμετρο για την περιοδική ανάλυση ψηφίων και επιτρέπει παρεμβολές μέσω button interrupt. Η δομή είναι διαχωρισμένη μεταξύ ISR και main loop, ώστε οι διακοπές να είναι ελαφριές και ασφαλείς.

1. Είσοδος Αριθμού μέσω UART

Ο χρήστης εισάγει έναν ακέραιο αριθμό μέσω σειριακής κονσόλας. Οι αποδεκτοί χαρακτήρες (0-9, '-', Enter) φιλτράρονται στο UART ISR και τοποθετούνται σε queue. Το Enter ('\r') ενεργοποιεί τη σημαία `input_ready`, ώστε η main loop να ξεκινήσει ανάλυση.

2. Ανάλυση Ψηφίων με Timer

Η ανάλυση ξεκινά μετά το Enter. Κάθε 500ms, εξετάζεται ένα ψηφίο:

- Αν είναι ζυγό: το LED αρχίζει να αναβοσβήνει κάθε 200ms.
- Αν είναι μονό: το LED κάνει μία εναλλαγή κατάστασης (toggle).

Ο έλεγχος των 500ms/200ms γίνεται μέσω counter στο timer ISR, με χρήση flags `timer500_flag` και `timer200_flag` που διαχειρίζονται αποκλειστικά από τη main loop.

3. Blink χωρίς delay

Το αναβόσβημα του LED δεν γίνεται με delay. Αντί αυτού, μια σημαία `blinking_active` ενεργοποιείται μόνο για ζυγά ψηφία, και κάθε 200ms γίνεται toggle μέσω του flag `timer200_flag`. Αυτό εξασφαλίζει ότι το πρόγραμμα δεν μπλοκάρει.

4. Button Interrupt και Κλείδωμα LED

Η διακοπή από το κουμπί απλώς θέτει τη σημαία `button_event_flag`, ενώ η πραγματική λειτουργία (κλείδωμα/ξεκλείδωμα του LED) γίνεται στη `main`. Το LED παγώνει στην τρέχουσα κατάσταση, αλλά η ανάλυση συνεχίζεται κανονικά. Με νέο πάτημα, ξεκλειδώνει.

5. Νέος Αριθμός Κατά την Ανάλυση

Αν ο χρήστης πατήσει `Enter` κατά την ανάλυση, η σημαία `new_input_override` ενεργοποιείται και η ανάλυση τερματίζεται άμεσα στην επόμενη διακοπή 500ms. Το νέο input αντικαθιστά το παλιό και ξεκινά νέα ανάλυση.

6. Ειδική Περίπτωση – Repeat Mode

Αν ο αριθμός τελειώνει σε παύλα (π.χ. 123-), η ανάλυση των ψηφίων ξεκινά ξανά από την αρχή μόλις τελειώσει, δημιουργώντας επαναληπτική λειτουργία. Η μόνη έξοδος είναι νέα είσοδος.

7. Άλλες Παρατηρήσεις

- Χρησιμοποιήθηκαν helper συναρτήσεις `led_on/off/toggle()` για καθαρό έλεγχο LED.
- Η μεταβλητή `led_on_state` διατηρεί την τρέχουσα κατάσταση LED, ώστε το `toggle` να είναι αξιόπιστο.
- Δεν γίνεται καθόλου λογική μέσα σε ISR — μόνο θέσεις σημάτων.

Προβλήματα και Testing

- Αρχικά χρησιμοποιήθηκε `volatile bool`, το οποίο προκάλούσε αστάθειες λόγω του τρόπου αποθήκευσης από τον compiler. Αντικαταστάθηκε με `volatile uint8_t`.
- Επαληθεύτηκε ότι οι σημαίες ανανεώνονται μόνο από ISRs, και ότι ελέγχονται/καθαρίζονται μόνο από τη `main`.
- Το `blinking` παραμένει συγχρονισμένο με το `timer`, ακόμη και σε `repeat mode` ή `button lock`.
- Όλες οι edge cases (`pause`, `restart`, `override`, `loop`) δοκιμάστηκαν με επιτυχία σε STM32 Nucleo M4.