

# Μικροεπεξεργαστές και Περιφερειακά

## Lab 2

Ομάδα 35

Μαρινόπουλος Χριστόφορος 10522

Σπυριδόπουλος Σταύρος 10845

Εαρινό εξάμηνο 2025

### Υλοποίηση Συναρτήσεων

Στο αρχείο `main.c` υλοποιήθηκε η λειτουργία του προγράμματος σύμφωνα με τις απαιτήσεις της άσκησης. Το πρόγραμμα διαχειρίζεται είσοδο αριθμών και του χαρακτήρα - μέσω UART, επεξεργάζεται τα ψηφία με χρονισμό, ελέγχει LED μέσω GPIO και διαχειρίζεται interrupts από κουμπί.

### Interrupts και Priorities

Η `main()` αρχικοποιεί τα εξής:

- UART με ρυθμό 115200 baud και callback για διαχείριση εισόδου (`uart_rx_isr`).
- Timer με περίοδο 0.5 sec για συγχρονισμό επεξεργασίας ψηφίων.
- GPIO για το LED (`PA_5`) και το κουμπί USER (`P_SW`) με interrupts σε Falling Edge.

Προτεραιότητες στα Interrupts:

- Button (priority 0)
- UART I/O (priority 2)

Το κουμπί έχει υψηλότερη προτεραιότητα (priority 0) έναντι του UART (priority 2).

Η είσοδος της UART φιλτράρεται ώστε να αποδεχτεί μόνο αριθμούς (0-9), παύλα (-), backspace (0x7F) και Enter (0x13). Οι μη έγκυροι χαρακτήρες αγνοούνται.

### Logging

Κάθε ψηφίο, που έχει εισαχθεί στην UART, επεξεργάζεται διαδοχικά κάθε 0.5 sec:

- Ζυγά ψηφία (even): Το LED αλλάζει 2 φορές κατάσταση με 200ms ανάμεσα στην κάθε αλλαγή.
- Μονά ψηφία (odd): Το LED αλλάζει κατάσταση και παραμένει σταθερό.

Στην περίπτωση που έχει απενεργοποιηθεί το LED (μέσω του κουμπιού USER), κάθε ενέργεια αναφέρεται στη UART (π.χ., Digit 5 -> Toggle LED). Αυτό συμβαίνει όταν `led_enabled==false`.

### Button Interrupt

Η συνάρτηση `button_isr` καλείται ως interrupt και εκτελείται όταν πατιέται το κουμπί:

- Αριθμός πατήματος μονός: Κλειδώνει το LED στην τρέχουσα κατάσταση, τυπώνει μήνυμα με τον αριθμό των μέχρι στιγμής πατημάτων (π.χ., Count = 1).
- Αριθμός πατήματος ζυγός: Ξεκλειδώνει το LED και τυπώνει μήνυμα με τον αριθμό των μέχρι στιγμής πατημάτων (π.χ., Count = 2).

Σε περίπτωση σύγκρουσης interrupts, το κουμπί έχει προτεραιότητα έναντι του UART.

## Αυτόματη Επαναφορά και Ειδικές Περιπτώσεις

Νέα είσοδος: Αν εισαχθεί νέος αριθμός κατά τη διάρκεια επεξεργασίας, η τρέχουσα διαδικασία διακόπτεται και ξεκινά νέα.

Επανάληψη με παύλα (-): Αν η είσοδος της UART τελειώνει σε - πριν πατηθεί το enter, η διαδικασία επεξεργασίας της εισόδου επαναλαμβάνεται συνεχώς (loop) μέχρι να υπάρξει νέα είσοδος.

## Testing

Για τον έλεγχο χρησιμοποιήθηκαν οι ακόλουθες περιπτώσεις:

**Εισαγωγή αριθμού χωρίς παύλα (π.χ., 52719):**

Το LED αλλάζει κατάσταση για μονά ψηφία και αναβοσβήνει για ζυγά. Παρεμβολή κουμπιού κλειδώνει/ξεκλειδώνει το LED και εμφανίζει σωστά τον αριθμό πατημάτων.

**Εισαγωγή αριθμού με παύλα (π.χ., 81-):**

Η επεξεργασία επαναλαμβάνεται συνεχώς μέχρι να εντοπιστεί νέα είσοδος.

**Ακύρωση επαναφοράς:**

Εισαγωγή νέου αριθμού κατά τη διάρκεια επεξεργασίας διακόπτει άμεσα την τρέχουσα διαδικασία.

**Μη έγκυροι χαρακτήρες (π.χ., a12b):**

Αγνοούνται και μόνο τα αριθμητικά ψηφία και οι παύλες αποθηκεύονται στην ουρά.

## Προβλήματα

Το κύριο πρόβλημα που αντιμετωπίστηκε ήταν η κλήση της ρουτίνας interrupt button\_isr η οποία καλούταν μόνο όταν πατιώταν το κουμπί RESET. Συμπεριφορά αρκετά περιέργη διότι στην main είχε οριστεί να καλείται μόνο όταν πατιέται το κουμπί USER.