

Αναφορά - 2η Εργασία

Εργαστηρίου

Μικροεπεξεργαστών

Καργιώτης Αλέξανδρος 10662
Παπαδόπουλος Παναγιώτης 10697

Εισαγωγή

Η παρούσα εργασία αφορά την ανάπτυξη ενός ενσωματωμένου συστήματος με χρήση του μικροελεγκτή STM32, το οποίο λαμβάνει ακέραιους αριθμούς μέσω σειριακής θύρας UART και πραγματοποιεί ανάλυση των ψηφίων τους με χρονισμό μέσω του timer. Στόχος ήταν η αλληλεπίδραση του χρήστη με το σύστημα σε πραγματικό χρόνο, με χρήση κουμπού, LED και χρονισμών, ώστε να εξεταστεί η διαχείριση διακοπών και η οργάνωση λογικής κατάστασης σε embedded περιβάλλον.

Υλοποίηση

Η εφαρμογή υλοποιήθηκε σε γλώσσα C και περιλαμβάνει χρήση των εξής βασικών στοιχείων:

- **UART** για λήψη και προβολή δεδομένων από τον χρήστη.
- **Timer (SysTick)** για χρονισμό της ανάλυσης των ψηφίων ανά 500ms.
- **Διακοπές EXTI** για ανίχνευση πατήματος κουμπού μέσω του PC_13.
- **Κατάσταση LED** που αλλάζει με βάση την τιμή κάθε ψηφίου:
 - Για **ζυγό ψηφίο**, γίνεται αναβόσβημα LED (200ms ON / 200ms OFF).
 - Για **μονό ψηφίο**, γίνεται toggle και παραμένει στη νέα κατάσταση.
- Υποστήριξη “**παγώματος**” του LED όταν πατηθεί το κουμπί.
- Υποστήριξη εισαγωγής νέου αριθμού ανά πάσα στιγμή — με άμεση διακοπή της τρέχουσας ανάλυσης και επανεκκίνηση με το νέο input.

Η λογική διαχωρίστηκε σε:

- **Επεξεργασία χαρακτήρων** με buffer staging,
- **Κύριο loop** που αντιδρά σε flags (digit_ready, new_input_received, led_locked),
- **Σύντομες συναρτήσεις διακοπής (ISR)** για χρονισμό και κουμπί.

Κατά την αρχικοποίηση ορίστηκαν οι προτεραιότητες των διακοπών μέσω `NVIC_SetPriority`, με την EXTI να έχει ανώτερη προτεραιότητα από τη σειριακή UART, ώστε να τηρείται η απαίτηση της εκφώνησης περί προτεραιότητας του κουμπιού.

Προβλήματα & Αντιμετώπιση

Κατά την ανάπτυξη αντιμετωπίστηκαν αρκετές προκλήσεις:

- **Σύγχυση στην προτεραιότητα των διακοπών**, με αποτέλεσμα ο handler του κουμπιού να μην εκτελείται. Η λύση δόθηκε με κατάλληλη ρύθμιση των `NVIC_SetPriority()` ώστε η EXTI15_10 να έχει υψηλότερη προτεραιότητα από το `USART2_IRQn`.
- **Μη προβλεπόμενη συμπεριφορά στον buffer εισόδου**, όταν ο χρήστης εισήγαγε νέο αριθμό κατά τη διάρκεια ανάλυσης. Το πρόβλημα λύθηκε με χρήση **διπλού buffer** (`input_staging[]` για πληκτρολόγηση, `input_buffer[]` για ανάλυση), ώστε να αποφεύγεται η αλλοίωση της τρέχουσας ακολουθίας.
- **Η διακοπή του κουμπιού δεν εκτελούνταν**, παρότι το EXTI handler καλούταν. Η αιτία εντοπίστηκε στο ότι το callback ανέμενε `int num` ως bitmask (`1 << pin`), ενώ στην πράξη το driver περνούσε απλώς το `pin index`. Η σύγκριση διορθώθηκε σε `if (num == 13)` ή απλώς αφαιρέθηκε εντελώς, αφού χρησιμοποιείται μόνο μία διακοπή.

Δοκιμές & Επιβεβαίωση Λειτουργίας

Το σύστημα δοκιμάστηκε με διάφορες εισόδους (π.χ. 52719, 42-, 777) και σε κάθε περίπτωση παρατηρήθηκε ορθή λειτουργία του χρονισμού, σωστή εναλλαγή LED και αντιστοίχιση των εντολών UART. Επίσης επαληθεύτηκε ότι:

- Η εισαγωγή νέου αριθμού διακόπτει την παλιά ακολουθία.
- Το κουμπί παγώνει σωστά τη λειτουργία LED και μετράει τα πατήματα.
- Η ακολουθία – επαναλαμβάνεται μέχρι να εισαχθεί νέος αριθμός.

Η ορθή συμπεριφορά του LED και των UART logs επαληθεύτηκε με χρήση του Tera Term.

Συμπεράσματα

Η εργασία αποτέλεσε ουσιαστική άσκηση στον προγραμματισμό embedded συστημάτων, στη διαχείριση διακοπών και στον συγχρονισμό μεταξύ UART, timer και GPIO. Αντιμετωπίστηκαν και επιλύθηκαν ρεαλιστικά προβλήματα που εμφανίζονται συχνά σε συστήματα πραγματικού χρόνου, και το τελικό αποτέλεσμα είναι σταθερό, επεκτάσιμο και πλήρως λειτουργικό.