

# Μικροεπεξεργαστές και Περιφερειακά

Εαρινό Εξάμηνο 2025

## 2<sup>η</sup> Εργασία/Εργαστήριο

Πληροφορίες κ. Δημήτρης Καρανάσσης: [dkaranassos@ece.auth.gr](mailto:dkaranassos@ece.auth.gr)

κ. Άγγελος Αθανασιάδης: [angelathan@ece.auth.gr](mailto:angelathan@ece.auth.gr)

Διεξαγωγή εργαστηρίου : 8 Μαΐου 2025 ώρες εργαστηρίων

Η εργασία θα πραγματοποιηθεί σε ομάδες των 2 ατόμων

Η παρούσα εργασία καλύπτει τον προγραμματισμό σε assembly ενός μικρό-ελεγκτή ARM με χρήση των εργαλείων Keil όπως σας έχουν παρουσιαστεί στο 2<sup>ο</sup> εργαστηριακό μάθημα. Στα πλαίσια της εργασίας θα γράψετε ένα πρόγραμμα, σε C, το οποίο διαμορφώνεται ως εξής:

1. **Είσοδος Αριθμού:** Ο χρήστης εισάγει έναν ακέραιο αριθμό (π.χ. 52719) μέσω UART.
2. **Ανάλυση Ψηφίων με Timer:**
  - Κάθε ψηφίο αναλύεται διαδοχικά κάθε 0.5 sec (χρήση timer interrupt).
  - **Ζυγό αριθμό (even):** το LED αναβοσβήνει κάθε 200ms (200ms ON - 200ms OFF).
  - **Μονό αριθμό (odd):** το LED αλλάζει την παρούσα κατάσταση κατάσταση (toggle) και την κρατάει σταθερή.
  - **Κάθε ενέργεια** εκτυπώνεται στο UART.
1. **Interrupt από το Button:** Αν το κουμπί πατηθεί κατά τη διάρκεια της ανάλυσης, τότε:
  - Το LED παγώνει στην τρέχουσα κατάσταση.
  - Εκτυπώνεται το εξής μήνυμα που περιέχει τον αριθμό των counts που έχει πατηθεί το button από την εκκίνηση του board μέχρι τώρα: Interrupt: Button pressed. LED locked. Count = X
  - Η ανάλυση συνεχίζεται χωρίς LED ενέργειες, αλλά με UART logging.
  - Σε σύγκρουση UART input & button interrupt, το κουμπί έχει προτεραιότητα.
3. **Έξοδος από την κλείδωση:** Με δεύτερο πάτημα του κουμπιού, η λειτουργία LED επανέρχεται.
4. **Νέος Αριθμός:**
  - Μόλις ολοκληρωθεί η ανάλυση όλων των ψηφίων, ζητείται αυτόματα νέος αριθμός από τον χρήστη.
  - Ο χρήστης μπορεί οποιαδήποτε στιγμή κατά τη διάρκεια της ανάλυσης ενός αριθμού να εισάγει νέο αριθμό. Σε αυτή την περίπτωση:
    - i. Η τρέχουσα διαδικασία σταματά αμέσως.
    - ii. Ξεκινά νέα ανάλυση από την αρχή, σαν να δόθηκε εξ αρχής μόνο ο νέος αριθμός.

5. **Ειδική Περίπτωση - Τέλος σε "-":** Αν ο αριθμός που εισάγεται τελειώνει σε παύλα (π.χ. 5271-), τότε η διαδικασία:
- Επαναλαμβάνεται σε loop συνεχώς για τον συγκεκριμένο αριθμό.
  - Η μόνη περίπτωση να διακοπεί είναι η είσοδος νέου αριθμού, οπότε όλα ξεκινούν εκ νέου από την αρχή.
6. **Οποιοδήποτε άλλος χαρακτήρας εκτός από νούμερο ή το "-" αγνοείται από το πρόγραμμα.**

### **Παράδειγμα UART Output:**

*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*  
*...*  
*(New input received)*  
*Input: 42*  
*Digit 4 -> Blink LED*  
*Digit 2 -> Blink LED*  
*End of sequence. Waiting for new number...*

Προτείνεται στο Keil να επιλέξετε τον μικρο-ελεγκτή NUCLEO M4 που σας έχει υποδειχθεί (και το ανάλογο Board) και περιγράφεται αναλυτικά και στο υλικό που έχει αναρτηθεί στο elearning.

**\*Σημείωση:** Σε περίπτωση που πατηθεί το κουμπί ενώ ο χρήστης καταχωρεί το input του θα πρέπει να έχει προτεραιότητα η είσοδος από το διακόπτη

**\*\*Σημείωση:** Για την επικοινωνία UART της πλακέτας με τον υπολογιστή σας μπορείτε να χρησιμοποιήσετε το [Tera Term](#) ή κάποιο άλλο πρόγραμμα το οποίο υποστηρίζει σειριακή επικοινωνία.

### **Παράδοση Εργασίας**

Η παράδοση την εργασίας θα γίνει μέσω του elearning και τα παραδοτέα της εργασίας θα είναι α) ένα αρχείο με τον κώδικα σας και σχόλια (το οποίο θα μπορούμε να τρέξουμε και εμείς στο Keil) και β) μια 2σέλιδη αναφορά που θα περιγράφετε τι κάνατε, ποια προβλήματα αντιμετωπίσατε και πως κάνατε testing.