



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΑΦΟΡΑ 2^{ου} ΕΡΓΑΣΤΗΡΙΟΥ ΓΙΑ ΤΟ ΜΑΘΗΜΑ Μικροεπεξεργαστές και Περιφερειακά

Νικόλαος Κυλιντηρέας 10908, Τοραμανίδου
Χριστοδούλη 10800

Σημείωση: ο κώδικας είναι και στο github: <https://github.com/NicKyliis/UART>

Εισαγωγή

Για την εκπόνηση του 2^{ου} εργαστηρίου, αρχικά δημιουργήσαμε ένα blueprint του κώδικα, που χρησιμοποιήσαμε ως δείκτη για το κύριο μέρος. Το αρχείο αυτό (blueprint.c) μπορεί να εκτελεστεί σε υπολογιστή και περιγράφει με προσομοίωση τη βασική λογική για τις περισσότερες λειτουργίες που ζητούνται.

Στη συνέχεια, γράψαμε ένα test αρχείο βασιζόμενο σε κώδικα που δίνεται στη θεωρία προκειμένου να ελέγχουμε την λειτουργία της UART στην πλακέτα. Σε αυτό το σημείο πειραματιστήκαμε και με τις υπόλοιπες λειτουργίες που χρειάζονται για την πραγματική (και όχι προσομοίωση) λειτουργία, όπως αρχικοποιήσεις, ανάγνωση κατάστασης led ή button και interrupts με timer.

Τέλος, δημιουργήθηκε το ζητούμενο αρχείο main.c το οποίο είναι σύμφωνο με τις προδιαγραφές που περιγράφονται στο παραδοτέο. Η σύνδεση με το NUCLEO έγινε μέσω του putty, όπου και ελέγχθηκε η λειτουργία του.

Δομή και περιγραφή του κώδικα

Η δομή του βασικού μας αρχείου μπορεί να περιγραφεί ως εξής:

1. Ορισμός/Αρχικοποίηση μεταβλητών: Στο πρώτο κομμάτι του κώδικα, ορίζουμε το μέγιστο χώρο για το buffer που «κρατάει» τη συμβολοσειρά του χρήστη, καθώς και τον αριθμό pin του κουμπιού (13) για αναγνωσιμότητα. Επίσης, ορίζουμε ως volatile int τα flags που θα χρειαστούμε, για μέγιστη ασφάλεια έναντι του compiler.
2. Αρχικοποίηση βοηθητικών/interrupt συναρτήσεων: Σε αυτό το σημείο δηλώνονται όλες οι συναρτήσεις που χρησιμοποιεί το πρόγραμμα. Για μέγιστη οργάνωση δημιουργήσαμε τις setLedOn/Off, η οποίες

περιλαμβάνουν και εκτύπωση μέσω UART, ενώ η συνάρτηση ledBlinker ορίζει τον τρόπο που θα αναβοσβήσει το LED, ανάλογα με το ψηφίο που επεξεργάζεται (άρτιος ή περιττός). Η συνάρτηση analysis είναι ένα time interrupt που καλείται κάθε 100ms και μετράει 200 και 500ms. Τέλος, οι uart_rx_isr και button_interrupt είναι interrupts εισόδου και καλούνται αμέσως μόλις ο χρήστης πληκτρολογήσει καινούριο νούμερο ή πατήσει το κουμπί αντίστοιχα.

3. Κύρια συνάρτηση main: Εδώ αρχικοποιούμε ό,τι χρειάζεται η πλακέτα (LED, timer, callbacks) και ξεκινάμε έναν ατέρμων βρόγχο (while(1)) για την εισαγωγή ψηφίων και την ειδική περίπτωση (να τελειώνει σε - το αλφαριθμητικό του χρήστη). Τέλος, δίνεται προτεραιότητα σε button interrupt έναντι του UART interrupt.

Να σημειωθεί ότι όπου έχουμε while(1) προσθέτουμε την εντολή __WFI() ώστε να μπορούμε να «πιάσουμε» οποιοδήποτε interrupt έρθει όσο περιμένει ο επεξεργαστής. Αυτή είναι και η βασική διαφορά με την εντολή delay(). Επίσης, εφόσον διαβάζουμε μόνο χαρακτήρες 0-9 και τον χαρακτήρα -, όλοι οι υπόλοιποι αγνοούνται από το πρόγραμμα, όπως ζητείται.

Testing και προβλήματα που αντιμετωπίστηκαν

Για να ελέγξουμε ότι ο κώδικας λειτουργεί σωστά, φορτώσαμε στα πλαίσια του εργαστηρίου τον κώδικα που περιεγράφηκε παραπάνω στην πλακέτα NUCLEO M4. Ως διεπαφή χρησιμοποιήσαμε το putty επιλέγοντας σειριακή σύνδεση. Κατά τη δοκιμή, εκτελέστηκαν όλοι οι συνδυασμοί των περιπτώσεων, στους οποίους το πρόγραμμα ανταποκρινόταν σύμφωνα με τις προδιαγραφές.

Τα βασικά προβλήματα που αντιμετωπίσαμε ήταν το ποια εξαρτήματα χρειάζονται αρχικοποίηση και πώς και πού να τα αρχικοποιήσουμε. Επίσης, το button φαίνεται να δουλεύει σωστά μόνο όταν το trigger mode που του θέτουμε είναι το Rising. Τέλος, για την ειδική περίπτωση (-) ήταν πολύπλοκο λόγω της δομής του κώδικα να το θέσουμε έτσι ώστε να διακόπτεται μόνο με UART input, οπότε και τελικά χρησιμοποιήσαμε την εντολή goto.