Μικροεπεξεργαστές και Περιφερειακά

Εργασία 2η : Human Reaction Time

Authors

Χριστοφορίδης Σάββας AEM: 9147 email: schristofo@ece.auth.gr
Πορτοκαλίδης Σταύρος AEM: 9334 email: stavport@ece.auth.gr

Code: https://github.com/sportokalidis/stm32-Nucleo_Human_Reaction_Time

Περιγραφή Προβλήματος

Σκοπός της δεύτερης εργασίας είναι η δημιουργία μιας ενσωματωμένης συσκευής, η οποία θα μετρά πόσο γρήγορα ένα άτομο μπορεί να πατήσει ένα κουμπί ως απάντηση σε ένα LED που ανάβει. Για τον σκοπό αυτό θα χρησιμοποιηθούν το switch (PC_13) και το LED (PA_5) που βρίσκονται πάνω στην πλακέτα "nucleo_stm32f401RE".

Εξήγηση Διαδικασίας Επίλυσης

Η διαδικασία η οποία ακολουθείται για την επίλυση της άσκησης είναι απλή και βασίζεται στην λειτουργία των interrupts. Αρχικά, η άσκηση ζητάει το ενσωματωμένο να έχει διπλή λειτουργία ανάλογα με την τιμή που θα έχει μια μεταβλητή #DEF. Στη πρώτη λειτουργία ο χρήστης περιμένει να ανάψει το LED για να αντιδράσει, ενώ στη δεύτερη ο χρήστης περιμένει να σβήσει το LED για να αντιδράσει. Στο πρόγραμμα ανάλογα με την λειτουργία στην οποία βρισκόμαστε, ο χρήστης περιμένει ένα τυχαίο χρονικό διάστημα από 0 έως 8 seconds, να αλλάξει η κατάσταση του LED και να πατήσει το κουμπί. Με το πάτημα του κουμπιού, ενεργοποιείται ένα interrupt, το **button_press_isr()**. Για την ομαλή λειτουργία του προγράμματος, όταν ένα interrupt εκτελείται, ελέγχει πρώτα εάν η κατάσταση του LED έχει μεταβληθεί. Για την μέτρηση του χρόνου αντίδρασης μετράμε τους κύκλους ρολογιού από την στιγμή που άναψε το LED, μέχρι την στιγμή που ενεργοποιείται το interrupt και διαιρούμε με την συχνότητα του επεξεργαστή. Το πείραμα εκτελείται πέντε φορές και στο τέλος υπολογίζει τον μέσο χρόνο αντίδρασης για τα πέντε πειράματα. Το πρόγραμμα δεν τερματίζει και ο χρήστης μπορεί να εκτελέσει το πείραμα όσες φορές θέλει.

Εξήγηση Υλοποίησης

Αρχικά, η επίτευξη της διπλής λειτουργίας γίνεται με τον ορισμό της μεταβλητής MOD. Όταν το MOD είναι 0 το ενσωματωμένο λειτουργεί στην πρώτη λειτουργία, ενώ όταν είναι 1 λειτουργεί στην δεύτερη λειτουργία. Για τον έλεγχο του LED χρησιμοποιούμε την συνάρτηση **leds_set(PA_5, PA_6, PA_7)**, όταν είναι leds_set(1,0,0) το LED ανάβει. Στο κώδικα, χρησιμοποιούμε κάποια σχέση συναρτήσει του MOD για να ελέγξουμε το LED ώστε να ικανοποιούμε και τις δύο λειτουργίες, π.χ. το (MOD+1)%2 δίνει 1 για MOD=0, ενώ δίνει 0 για MOD=1. Αρχικά, στην main() γίνονται όλες οι απαραίτητες αρχικοποιήσεις και δηλώσεις μεταβλητών. Στην συνέχεια, προχωράμε σε μια while(1) loop, όπου για κάθε επανάληψη του πειράματος, δημιουργούμε μια τεχνητή καθυστέρηση με χρήση της delay_ms(), από 0 έως 8 seconds και μεταβάλουμε την κατάσταση του LED. Με το πάτημα του κουμπιού από τον χρήστη ενεργοποιείται το interrupt **button_press_isr().** Στη ρουτίνα αυτή, αρχικά, γίνεται ο έλεγχος εάν η κατάσταση του LED έχει μεταβληθεί, η συνθήκη ελέγχου είναι **signal==1**, αν η τιμή δεν είναι 1, η ρουτίνα τερματίζει χωρίς να κάνει κάτι. Στη περίπτωση που είναι 1, γίνεται ο τελικός υπολογισμός του χρόνου αντίδρασης, καθώς επίσης αλλάζει και η κατάσταση του LED. Για τον υπολογισμό των κύκλων ρολογιού χρησιμοποιούμε έναν free running counter του Debug Watch and Trace module όπως αναφέρει στο [1].

Προβλήματα που Αντιμετωπίστηκαν

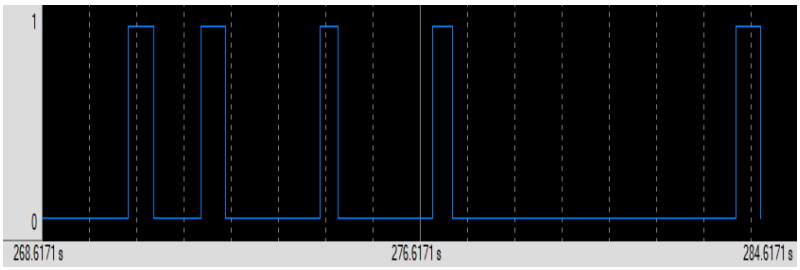
Ένα από τα προβλήματα που αντιμετωπίστηκαν ήταν η μέτρηση χρόνου καθώς δεν υπήρχε πληθώρα επιλογών για να υπολογιστεί ο χρόνος αντίδρασης του χρήστη. Επίσης, ένα κομμάτι που πήρε αρκετό χρόνο ήταν η κατανόηση της αντιστοίχισης των pins της πλακέτας και των αντίστοιχων integer αριθμών τους. Τέλος, ένα πρόβλημα που χρειάστηκε να λυθεί ήταν η παρεμβολή των interrupts σε "ευαίσθητο" κομμάτι κώδικα που ήταν υπεύθυνο για αλλαγή της κατάστασης του LED και την εκτέλεση της εντολής **delay_ms()**.

Testing

Το testing πραγματοποιήθηκε μέσω του ST-Link Debugger πάνω στην πλακέτα και χρησιμοποιώντας ως είσοδο το κουμπί B1 της πλακέτας και ως έξοδο το LED LD2. Για την επιβεβαίωση της ορθότητας του κώδικα στο testing εξετάστηκαν όλες οι οριακές συνθήκες όπως το συνεχόμενο ή το πρόωρο πάτημα του κουμπιού. Επιπλέον, για παρακολούθηση και έλεγχο της συμπεριφοράς του προγράμματος χρησιμοποιήθηκε εκτός από το LED εξόδου και η συνάρτηση **printf()** εκτυπώνοντας αποτελέσματα και τυχόν σφάλματα. Τέλος, για την επιβεβαίωση των αποτελεσμάτων χρησιμοποιήσαμε το εργαλείο του Keil, Logic Analyzer, όπου αναπαραστήσαμε σε ένα γράφημα με μορφή παλμών, τον χρόνο αντίδρασης του χρήστη, ουσιαστικά, είναι αναπαράσταση των τιμών της global μεταβλητής **signal** συναρτήσει του χρόνου. Το αποτέλεσμα που πήραμε, επιβεβαίωσαν τα αποτελέσματα των μετρήσεων μας.

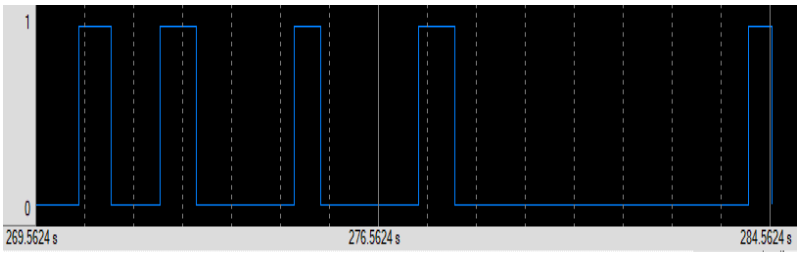
Μετρήσεις και Πειράματα

	Χρόνος Αντίδρασης
1ο	524 ms
2ο	523 ms
3ο	388 ms
4ο	415 ms
5ο	525 ms
Μ.Ο.	**474 ms**



Εικόνα 1: Λειτουργία 1η: MOD=0

	Χρόνος Αντίδρασης
1ο	647 ms
2ο	746 ms
3ο	540 ms
4ο	729 ms
5ο	496 ms
Μ.Ο.	**631.59 ms**



Εικόνα 2: Λειτουργία 2η: MOD=1

Sources

[1] https://www.embedded-computing.com/articles/measuring-code-execution-time-on-arm-cortex-m-mcus