Εργαστήριο Μαθήματος: “Μικροεπεξεργαστές & Περιφερειακά”

2η Εργασία

16/5/2024

Ομάδα 18:
Μαμουγιώργη Μαρία 10533
Ξακουστού Αιμιλία 10324

ΕΙΣΑΓΩΓΗ

Για τη διεκπεραίωση της συγκεκριμένης εργασίας υλοποιήθηκε μια συνάρτηση main.c χρησιμοποιώντας παράλληλα και τους drivers που δόθηκαν. Βασικός σκοπός της είναι η δημιουργία ISR ανάλογα με το ΑΕΜ που εισάγει ο χρήστης μέσω της σειριακής διεπαφής UART. Έτσι επιτρέπεται στο μικρο-ελεγκτή να ανταποκρίνεται άμεσα στα εισερχόμενα δεδομένα δίχως να αναγκάζεται να σπαταλά πόρους της κεντρικής μονάδας επεξεργασίας.

ΟΡΙΣΜΟΙ ΜΕΤΑΒΛΗΤΩΝ

Αρχικά καθορίζονται το led και το button που αντιστοιχούν στον μικρο-ελεγκτή NUCLEO M4. Συνεπώς το led αντιστοιχεί στο pin PA5 και το button στο PC13. Έπειτα ορίζονται κάποιες στατικές μεταβλητές, ο counter για να υπολογίζεται ο αριθμός που πατήθηκε το κουμπί, ένας πίνακας εκατό στοιχείων ώστε να γίνει επιτυχώς η εκτύπωση μέσω της uart_print του uart.h και το last_digit, το οποίο αντιπροσωπεύει το τελευταίο ψηφίο του ΑΕΜ που δόθηκε από τον χρήστη.

ΣΥΝΑΡΤΗΣΗ TIMER_ISR

Η συγκεκριμένη συνάρτηση είναι ουσιαστικά το ISR του timer στην οποία χρησιμοποιείται η gpio_toggle(LED_PIN) του gpio.h όπου αλλάζει την κατάσταση του led από ON σε OFF και αντίστροφα.

ΣΥΝΑΡΤΗΣΗ ODD_ISR

Πρόκειται για το ISR που καλείται όταν το ΑΕΜ που δίνει ο χρήστης έχει περιττό/μονό τελευταίο ψηφίο . Αρχικά μέσω της time_enable από το timer.h ενεργοποιείται ο timer ο οποίος ανά 0,5 sec θα αλλάζει την κατάσταση του led, το οποίο επιτυγχάνεται όπως και προηγουμένως με την gpio_toggle(LED_PIN). Έπειτα πραγματοποιείται η σειριακή εκτύπωση μέσω της UART.

ΣΥΝΑΡΤΗΣΗ EVEN_ISR

Σε περίπτωση που το last_digit είναι άρτιος/ζυγός αριθμός καλείται μία άλλη ISR στην οποία απενεργοποιείται ο timer μέσω της timer_disable του timer.h, αφού πλέον δεν χρειάζεται και έτσι το led κρατάει την τελευταία κατάστασή του. Ακολουθεί ένας έλεγχος για το αν η τελευταία κατάσταση του led πριν κληθεί η συνάρτηση είναι ON ή OFF. Έτσι αν η τιμή που επιστρέφει η gpio_get(LED_PIN) του gpio.h ,η οποία πρόκειται για αυτήν την τιμή, είναι 1 τότε το led είναι αναμμένο και εκτυπώνεται το αντίστοιχο μήνυμα μέσω της uart_print. Αντίστροφα αν η συνάρτηση επιστρέφει 0 σημαίνει ότι το led είναι σβηστό.

ΣΥΝΑΡΤΗΣΗ BUTTON_PRESS_ISR

Αν πατηθεί το κουμπί (BUTTON_PIN) σε οποιαδήποτε στιγμή πρέπει να αλλάζει η κατάσταση του led αντίθετα. Δηλαδή αν είναι αναμμένο το led να σβήνει και αντίστροφα. Για αυτό αρχικά απενεργοποιείται ο timer μέσω της timer_disable και αυξάνεται η μεταβλητή counter κατά ένα αφού πατήθηκε το κουμπί. Στη συνέχεια γίνεται ένας έλεγχος της τελευταίας κατάστασης του led

προκειμένου να γίνουν οι απαραίτητες αντίστοιχες ενέργειες. Αν η τιμή που επιστρέφει η συνάρτηση gpio_get(LED_PIN) είναι 0 τότε η !gpio_get(LED_PIN) είναι 1 και μέσω της gpio_set(LED_PIN,LED_ON) του gpio.h, τίθεται το LED_PIN το λογικό 1 και έτσι μέσω αυτής το led ανάβει. Χρησιμοποιώντας την sprintf, uart_print γίνονται οι αντίστοιχες εκτυπώσεις. Όμοια και για αν η συνάρτηση gpio_get(LED_PIN) επιστρέψει 1, τότε το !gpio_get(LED_PIN) θα δώσει 0 οπότε το led θα σβήσει.

ΑΡΧΙΚΟΠΟΙΗΣΕΙΣ ΚΑΙ ΚΛΗΣΕΙΣ ISR ΣΤΗΝ MAIN

Προκειμένου να αρχικοποιηθεί ο timer χρησιμοποιείται η timer_init του timer.h όπου βάζουμε ως περίοδο 0,5 seconds (500000 microseconds), όσο αναφέρεται και στην εκφώνηση. Ξανά καλείται η timer_disable αφού η ενεργοποίηση του γίνεται μόνο όταν καλείται η odd_isr. Τέλος χρησιμοποιείται από το timer.h η timer_set_callback (timer_isr) η οποία περνάει ένα call back στο API. Ουσιαστικά όταν συμβαίνει ένα interrupt καλείται μέσω αυτής της συνάρτησης η timer_isr περιοδικά.
Ακολουθεί η δήλωση του LED_PIN μέσω της gpio_set_mode του gpio.h ως έξοδος προκειμένου να υπάρχει πιο ολοκληρωμένος έλεγχος. Επίσης αρχικοποιείται μέσω της gpio_set η κατάσταση του led ώστε να είναι αρχικά σβηστό.
Η κλήση των συναρτήσεων όσων αφορά το BUTTON_PIN επιτρέπει στον μικροελεγκτή να ανταποκρίνεται αποτελεσματικά και αποδοτικά στα πατήματα του κουμπιού. Συγκεκριμένα για κάθε μια από τις συναρτήσεις του gpio.h ισχύει ότι η gpio_set_mode(BUTTON_PIN, PullUp) καλείται ώστε να θέσει το κουμπί στην κατάσταση Pull_up , δηλαδή λογικό 1 όταν δεν πατιέται. Με αυτό εξασφαλίζεται ένα προεπιλεγμένο επίπεδο για το κουμπί όταν δεν έχει πατηθεί. Η gpio_set_trigger (BUTTON_PIN, Rising) καλείται προκειμένου να καθοριστεί πως θα εκτελεστεί η αντίστοιχη ISR, δηλαδή η button_press_isr . Θα εκτελεστεί όταν έχουμε rising edge δηλαδή μετάβαση από low σε high value, δηλαδή από το 0 σε 1. Τέλος γίνεται κλήση της gpio_set_callback(BUTTON_PIN, button_press_isr) του gpio.h ώστε κάθε φορά που πατιέται το κουμπί να καλείται η button_press_isr και να εκτελούνται όσα προαναφέρθηκαν.

ΚΛΗΣΕΙΣ ISR ΣΤΗΝ MAIN

Έχοντας ήδη καθορίσει ότι το AEM θα αποθηκεύεται ως πίνακας στο buffer και ότι το τελευταίο στοιχείο buff[buff_index -1] ισούται με τον μηδενικό χαρακτήρα,το last_digit ορίζεται ως το προτελευταίο στοιχείο με τον ίδιο τρόπο, last_digit=buff[buff_index - 2]. Στη συνέχεια προκειμένου να καθοριστεί αν το τελευταίο ψηφίο του AEM είναι μονός ή ζυγός αριθμός πραγματοποιείται η πράξη υπολογισμού του υπολοίπου last_digit=last_digit%2. Έτσι τελικά μέσω ενός if-else ελέγχου, αν το υπόλοιπο ισούται με 0 πρόκειται για ζυγός αλλιώς για μονός αριθμός, καλούνται οι αντίστοιχες συναρτήσεις ISR, even_isr ή odd_isr και πραγματοποιείται ότι αναλύθηκε προηγουμένως.

PROBLEMS

Το πρόβλημα που ήρθαμε αντιμέτωπες ήταν η αρχικοποίηση και ο καθορισμός του κουμπιού. Για αυτό χρησιμοποιήθηκαν συναρτήσεις του gpio.h (gpio_set_mode, gpio_set_trigger,gpio_set_callback) που αναλύθηκαν προηγουμένως ώστε να λυθεί. Καθώς δοκιμάζαμε, προσπαθήσαμε να ενεργοποιήσουμε το interrupt όταν το κουμπί είναι ήδη πατημένο και αφήνεται. Δηλαδή ακολουθώντας Falling mode ,που σημαίνει ότι το interrupt διεγείρεται όταν το GPIO BUTTON_PIN πηγαίνει από το 1 στο 0. Ωστόσο θέτοντάς το σε αυτή την κατάσταση δε λειτούργησε όπως θα θέλαμε. Ο AEM είναι ακέραιος αριθμός, οπότε η UART πρέπει να δέχεται μόνο ακέραιους αριθμούς και τον enter χαρακτήρα. Για αυτό προσαρμόστηκαν ανάλογα στην uart_rx_isr οι τιμές που μπορεί να δεχθεί η μεταβλητή rx, σύμφωνα με τον πίνακα ASCII.

TESTING

Με τη χρήση του build και του debugger του Keil και του PuTTy, έγιναν δοκιμές για το αν υλοποιούνται τα ζητούμενα. Ως είσοδοι χρησιμοποιήθηκαν ποικίλες και διαφορετικές τιμές και το κουμπί πατήθηκε αρκετές φορές σε οποιαδήποτε στιγμή. Για οποιαδήποτε τιμή εισόδου που αντιστοιχεί σε AEM και για οποιοδήποτε πάτημα του κουμπιού η κατάσταση του led, ήταν η επιθυμητή.