Μικροεπεξεργαστές και Περιφερειακά
Εαρινό Εξάμηνο 2023

2η Εργασία/Εργαστήριο

Αρχικά στη συνάρτηση main του κώδικα ενεργοποιήθηκαν τα IRQ interrupts με την συνάρτηση __enable_irq(), αρχικοποιήθηκε η ουρά queue με την εντολή queue_init(&queue, 128). Έγιναν οι παρακάτω αρχικοποιήσεις που αφορούν την uart επικοινωνία και τα interrupts.

```
uart_init(115200);
uart_set_rx_callback(uart_rx_isr);
uart_enable();

leds_init();
leds_set(0,0,0);
```

Όσον αφορά τον διακόπτη, αρχικοποιήθηκε σε κατάσταση PullUp και θέσαμε το interrupt να ενεργοποιείται με το Rising mode. Δηλαδή, όταν δεν είναι πατημένος ο διακόπτης είναι σε υψηλή τάση, ενώ όταν είναι πατημένος, είναι στο ground. Έτσι, όταν αφήνεται ξανά ο διακόπτης μεταβαίνει από κατάσταση γείωσης σε κατάσταση υψηλής τάσης και ενεργοποιείται το interrupt. Όταν πατιέται ο διακόπτης, εκτελείται η gpio_set_callback(P_SW, switch_press_isr).

```
gpio_set_mode(P_SW, PullUp);
gpio_set_trigger(P_SW, Rising);
gpio_set_callback(P_SW, switch_press_isr);
```

Εκτός της main ορίστηκαν δύο συναρτήσεις, η void uart_rx_isr(uint8_t a) και η void switch_press_isr(int status).

```
void uart_rx_isr(uint8_t a){
  if((char)a != '\r'){
    queue_enqueue(&rx_queue, a);
    uart_print("added to queue \r\n");
    c++;
  }
  else{

    ledFlag = 1;
    uart_print(" flag = 1 \r\n");
  }
```

```
void switch_press_isr(int status){
      switchFlag = 1;
}
```

Η void uart_rx_isr(uint8_t a) καλείται μέσω της uart_set_rx_callback(uart_rx_isr) όταν εισάγεται οποιοσδήποτε χαρακτήρας του πληκτρολογίου.

Η uart_rx_isr(uint8_t a) γεμίζει μία ουρά με τους χαρακτήρες που δίνονται από το πληκτρολόγιο και για κάθε χαρακτήρα αυξάνει τον counter c κατά 1. Μόλις πατηθεί το πλήκτρο «enter» σταματάει να προσθέτει χαρακτήρες στην ουρά και αλλάζει το ledFlag από 0 σε 1.

Η switch_press_isr(int status) αλλάζει το switchFlag σε 1 κάθε φορά που καλείται.

Στη συνέχεια της main, μετά τις αρχικοποιήσεις, υπάρχει ένα infinite loop όπου ανάλογα με τα flags που έχουν οριστεί εκτελούνται ορισμένες λειτουργίες. Συγκεκριμένα, αν το switchFlag είναι ίσο με 1, σημαίνει ότι έχει πατηθεί ο διακόπτης και ελέγχεται η κατάσταση του led. Αν το led είναι αναμμένο το σβήνει, αν είναι σβηστό το ανάβει, ενώ και στις δύο περιπτώσεις αυξάνει τον counter που μετράει το πλήθος των πατημάτων του διακόπτη και τυπώνεται αυτός ο αριθμός μέσω της uart. Αφού τυπωθεί το μήνυμα μηδενίζεται η τιμή του switchFlag. Για παράδειγμα:

```
if(gpio_get(P_LED_R)){
                                /*if the switch is pressed and the LED is already activated*/
    gpio_set(P_LED_R, 0);       /* deactivates the LED*/
    counter++;                  /*ups the counter by 1*/
    sprintf(s2, "The number of switch has been pressed is: %d\r\n", counter);
    uart_print(s2);             /*prints the number of times the switch was pressed*/
```

Έπειτα γίνεται έλεγχος για το ledFlag. Αν αυτό είναι ίσο με 1, σημαίνει ότι έχει δοθεί ένα ΑΕΜ. Στη συνέχεια σε έναν integer j αφαιρείται από την τιμή ASCII του τελευταίου στοιχείου της ουράς η τιμή ASCII του μηδέν ώστε να πάρουμε τον ακέραιο. Έπειτα ελέγχεται αν αυτό είναι περιττός ή άρτιος αριθμός ώστε να σβήσει ή να ανάψει το led αντίστοιχα.

```
j = (int)rx_queue.data[4] - 48;

if(j%2 == 0){
  gpio_set(P_LED_R, 0);
}
else{
  gpio_set(P_LED_R, 1);
}
```

Τέλος, επαναφέρουμε τις παρακάτω μεταβλητές στο 0 ώστε να μπορεί να επαναχρησιμοποιηθεί η ουρά και θέτουμε την τιμή του ledFlag ίση με το μηδέν.

```
rx_queue.head = 0;
rx_queue.tail = 0;
ledFlag = 0;
```

Να σημειωθεί ότι σε περίπτωση που πατηθεί ο διακόπτης και εισαχθεί ένα ΑΕΜ ταυτόχρονα δίνεται προτεραιότητα στον διακόπτη όπως φαίνεται και στον έλεγχο: if(ledFlag == 1 && buttonFlag == 0).

Σιταρίδης Παναγιώτης ΑΕΜ: 10249

Τσιτσάνου Άννα ΑΕΜ: 10051