

Μικροεπεξεργαστές και Περιφερειακά

Εργαστήριο 2

Πάκας Αλέξιος-Γεώργιος

Ρίζος Γεώργιος

-Ξεκινάμε δηλώνοντας τη **volatile** μεταβλητή *counter*, η οποία θα χρησιμοποιείται για να μετράμε το πόσες φορές έχει πατηθεί το κουμπί, και μαζί το string που απαιτείται για να το τυπώνουμε μέσω **UART**.

Έπειτα δηλώνεται το **Queue** *rx_queue*, στο οποίο θα καταχωρούνται οι χαρακτήρες που δέχεται το πρόγραμμα μέσω **UART**.

Μετά φτιάχνω τους **interrupt handlers** για την καταχώρηση των χαρακτήρων και για το πάτημα του button.

Void uart_rx_isr:

Η συνάρτηση *uart_rx_isr* είναι μια διακοπτική ρουτίνα υπηρεσιών (ISR - Interrupt Service Routine) που καλείται κάθε φορά που λαμβάνεται ένας χαρακτήρας μέσω της μετάδοσης *uart*. Η γραμμή *queue_enqueue(&rx_queue, rx)*; καλεί τη συνάρτηση *queue_enqueue* για να εισάγει τον ληφθέντα χαρακτήρα *rx* στην ουρά *rx_queue*. Η συνάρτηση *queue_enqueue* προσθέτει τον χαρακτήρα *rx* στο τέλος της ουράς *rx_queue*. Εάν η ουρά είναι πλήρης τότε έχουμε δυο σενάρια. Πρώτα ελεγχουμε αν το *rx* είναι ζυγός ή μονός χρησιμοποιώντας *modulo*. Εάν είναι μονός τότε χρησιμοποιώντας την *gpio_set(P_LED_G, 1)*; , αναβουμε το *led*, βάζουμε 5sec wait με την *timer_init*, και ομοια με την *gpio_set(P_LED_G, 0)*; , κλείνουμε το *led* ή λούπα τρέχει μέχρι να έχουμε κάποιο άλλο ISR.

Εάν έχουμε ζυγο αριθμο, καλείται *timer_disable()* για να σταματήσουμε τον timer, και ύστερα αναβουμε και κλείνουμε το λαμπάκι με την χρήση προηγούμενων συναρτήσεων. Τέλος καλούμε την *queue_init* για να πάρουμε καινούργιο AEM.

void button_press_isr(int status):

Καλείται με το rising edge του πατήματος του button.

Ελέγχει εάν το **LED** είναι σβηστό, και εάν είναι, τότε το ανάβει και τυπώνει την αλλαγή μέσω της **UART**. Εάν είναι αναμμένο, τότε το σβήνει και επίσης τυπώνει την αλλαγή μέσω **UART**.

Πριν επιστρέψει στην **main**, αυξάνει κατά 1 τον **counter** και τυπώνει το πόσες φορές έχει πατηθεί το button.

int main(void):

Πρώτα αρχικοποιούνται τα **LEDs** και μετά θέτουμε το πράσινο **LED** σαν *output* και το button σαν *resistive pull-up input*.

Αρχικοποιούμε τον **counter** στο 0.

Ενεργοποιούμε την επικοινωνία **UART** με ρυθμό μετάδοσης bit 115200 και ορίζουμε την επιστροφή από τη συνάρτηση που ζητάει τον χαρακτήρα να είναι ο interrupt handler που φτιάξαμε παραπάνω.

Ομοίως, ενεργοποιούμε το button **P_SW** να πυροδοτείται με το πάτημα και ορίζουμε την επιστροφή απ' αυτό να είναι ανάλογος interrupt handler.

Τέλος, το πρόγραμμα εισέρχεται στον ατέρμονο while loop, (δηλαδή, while(1)), όπου δίνονται οι χαρακτήρες

Μόλις δοθεί το πλήρες AEM, ο ISR θα δράσει αναλόγως για το τελευταίο ψηφίο, και έπειτα θα ξαναζητήσει AEM, ενώ παράλληλα καθ' όλη τη διάρκεια αυτή, μπορεί να πατηθεί το κουμπί και να δράσει ο κατάλληλος ISR.