

HPY411- Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

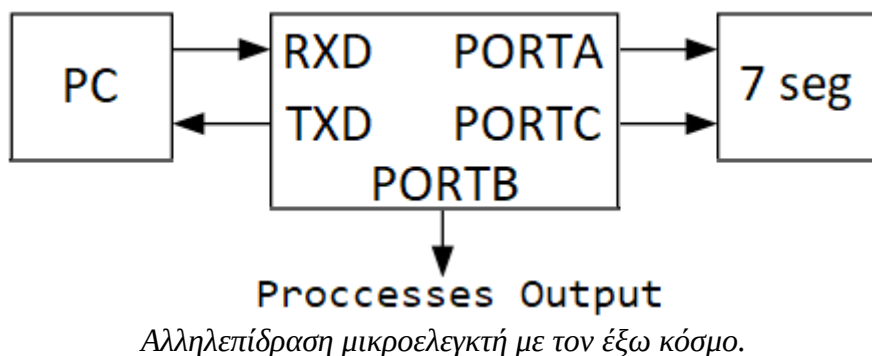
Εργαστήριο 7

LAB41145851

27/11/2020

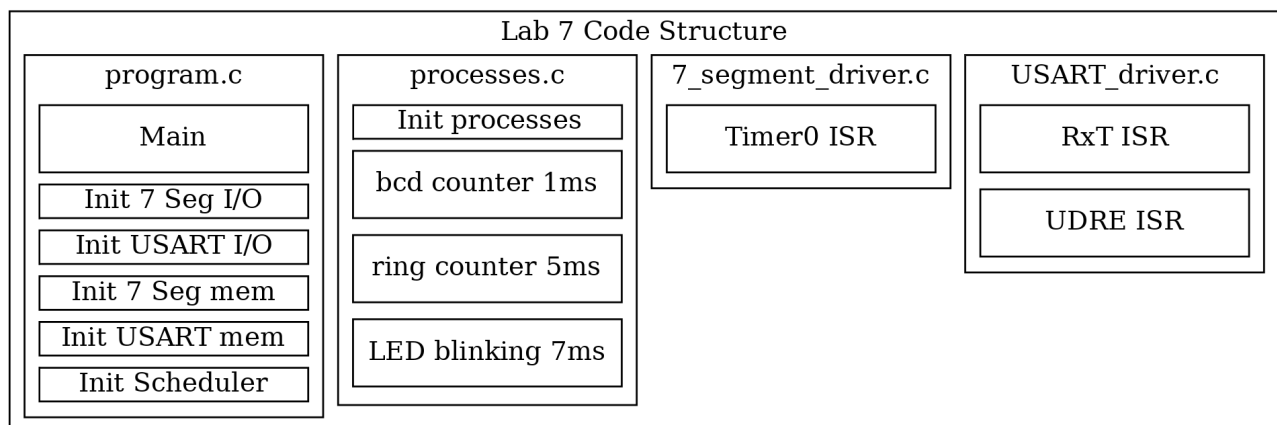
Εμμανουήλ Πετράκος AM 2014030009

Στο 7ο εργαστήριο προετοιμάζεται το σύστημα για την υλοποίηση δρομολογητή διεργασιών. Η PORTB χρησιμοποιείται ως η κοινή έξοδος όλων των διεργασιών. Επίσης, έχει παραμείνει η λειτουργικότητα του εργαστηρίου 5.



Επεξήγηση προσέγγισης

Καταρχάς, έχει επεκταθεί η συνάρτηση εξυπηρέτησης interrupts του receiver στον USART driver ώστε να δέχεται εντολές ενεργοποίησης/απενεργοποίησης διεργασιών. Επίσης, στο αρχείο processes.c, έχουν υλοποιηθεί 3 απλές διεργασίες για την δοκιμή του δρομολογητή. Τέλος, υπάρχουν δύο νέες συναρτήσεις αρχικοποίησης, μία για τον δρομολογητή και μία για τις διεργασίες.



Με τις νέες προδιαγραφές, η αντιμετώπιση των αριθμών που λαμβάνει ο receiver εξαρτάται από προηγούμενα frames. Άρα ο driver πρέπει να έχει μνήμη. Για αυτό, χρησιμοποιείται μια θέση μνήμης (receiver status) όπου στο ξεκίνημα ενός μηνύματος, αν αυτό πρόκειται να έχει αριθμούς, κρατάει τον τύπο του. Πέρα από αυτό, η λειτουργία του driver παραμένει ως έχει. Στον παρακάτω πίνακα φαίνεται αναλυτικά η αντιμετώπιση κάθε frame.

Είσοδος	Ενέργεια
C	Καθαρισμός δεδομένων 7 segment
N	Καθαρισμός δεδομένων 7 segment, receiver status = N
S	receiver status = S
Q	receiver status = Q
A, T, <CR>	Καμία
<LF>	Ενεργοποίηση Interrupt απάντησης (UDRIE), αύξηση μετρητή εκκρεμών απαντήσεων, καθαρισμός receiver status
Αριθμός	ASCII → bcd <ul style="list-style-type: none"> • Αν receiver status = N, αποθήκευση αριθμού στα δεδομένα 7 segment • Αν receiver status = S, ενεργοποίηση αντίστοιχου process • Αν receiver status = Q, απενεργοποίηση αντίστοιχου process

Ο ατέρμον βρόγχος της main συνάρτησης έχει επεκταθεί ώστε να καλεί τις ενεργοποιημένες διεργασίες την μία μετά την άλλη. Για να θυμάται το σύστημα ποιες είναι ενεργοποιημένες, χρησιμοποιείται μια θέση μνήμης (Scheduler Control) η οποία περιγράφεται παρακάτω.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	SCPE3	SCPE2	SCPE1	Scheduler Control
Init	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved bits**

Θα χρησιμοποιηθούν στην ολοκλήρωση του scheduler.

- **Bits 2..0 – SCPE3, SCPE2, SCPE1: Scheduler Process Enable 3, 2, 1**

Τα bits 2 έως 0 καθορίζουν ποιες διεργασίες είναι ενεργοποιημένες. 1 για ενεργοποίηση, 0 για απενεργοποίηση.

Για την δοκιμή του δρομολογητή, έχουν δημιουργηθεί 3 απλές διεργασίες με κοινή έξοδο την θύρα B. Και οι τρεις περιέχουν κώδικα καθυστέρησης, ώστε να προσομοιώνουν πιο σύνθετες διεργασίες και να δοκιμάσουν την ανοχή του συστήματος σε απόκλιση των timeslices. Ο scheduler είναι non-preemptive και πρέπει να τελειώσει μια διεργασία για να ξεκινήσει η επόμενη. Είναι υλοποιημένες ως συναρτήσεις σε C, δηλαδή υλοποιούνται με call και return.

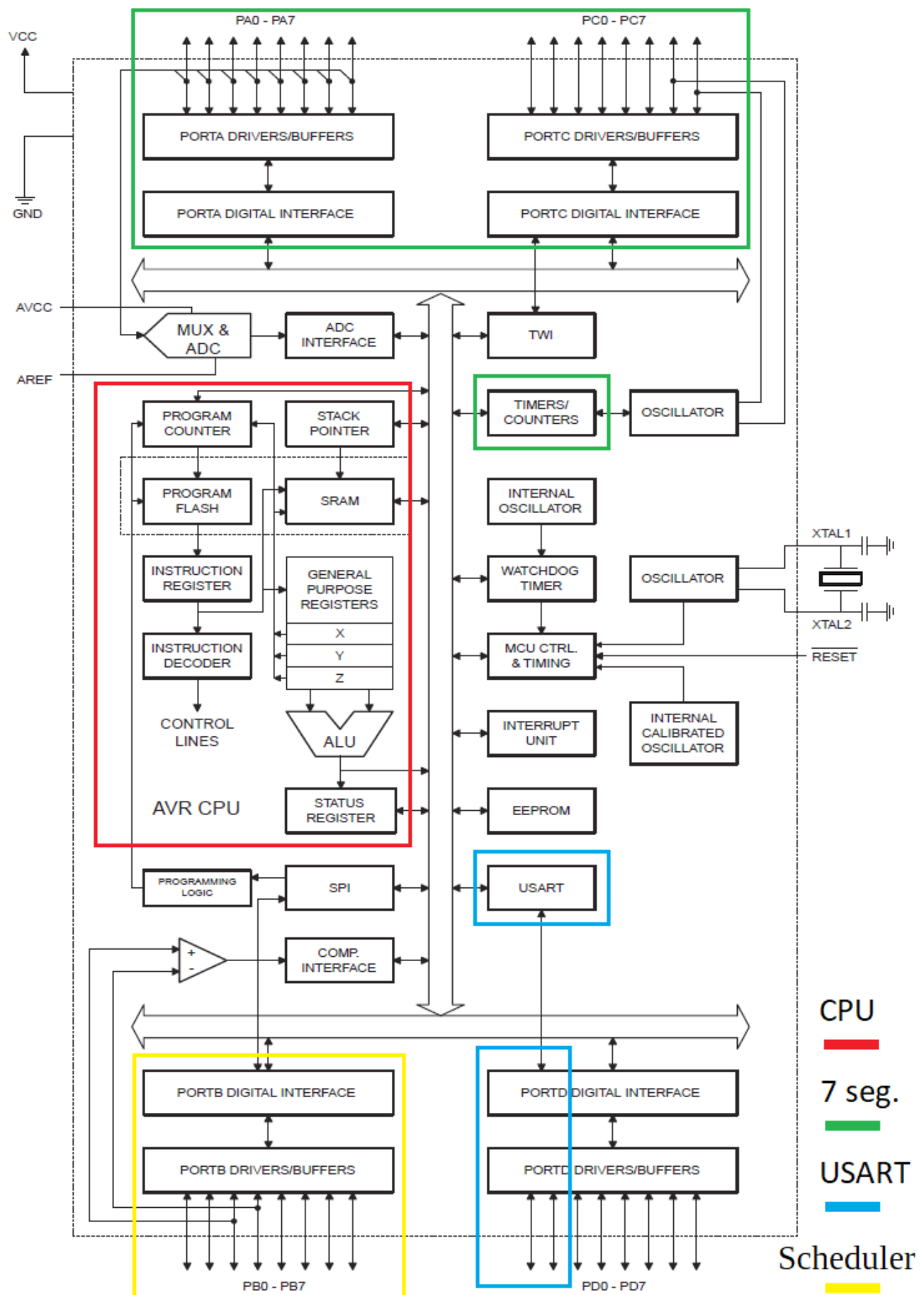
- `bcd_counter_1ms()`: Απλός αύξον μετρητής με καθυστέρηση 1ms. Χρησιμοποιεί την θέση μνήμης `Bcd_counter_data`.
- `bcd_counter_1ms()`: Μετρητής δακτυλίου με καθυστέρηση 5ms. Χρησιμοποιεί την θέση μνήμης `ring_counter_data`.
- `LED_blinking_7ms()`: Αντιστρέφει την τιμή της θέσης μνήμης `LED_blinking_data` με καθυστέρηση 7ms.

Παρακάτω φαίνεται πως χαρτογραφεί την μνήμη ο compiler και υπογραμμίζονται οι αλλαγές για την υλοποίηση του δρομολογητή.

0x0000	Register address space
...	
0x005F	
0x0060	Scheduler control
0x0061	OK transmits left
0x0062	Transmitter status
0x0063	Receiver status
0x0064	7 segment format 0
...	...
0x006e	7 segment format A
0x006f	Data LSB
...	...
0x0076	Data MSB
0x0077	LED blinking data
0x0078	Ring counter data
0x0079	Bcd counter data
...	Unused memory
...	Stack
...	
0x045f	

Χάρτης μνήμης

Τέλος, παρουσιάζεται ο χάρτης χρησιμοποιούμενων πόρων. Η θύρα B χρησιμοποιείται ως η κοινή έξοδος των διεργασιών που ελέγχονται από τον scheduler



Πειραματική Διαδικασία

Ο έλεγχος του συστήματος είναι σπασμένος σε δύο κομμάτια ώστε να είναι πιο εύκολη η ανάλυση του. Στο πρώτο κομμάτι δοκιμάζονται οι αλλαγές στο USART ενώ στο δεύτερο η λειτουργία των διεργασιών.

Στέλνοντας τα παρακάτω μηνύματα μέσω του stimulus file και θέτοντας ένα breakpoint στο τέλος της συνάρτησης εξυπηρέτησης τους, φαίνεται η κατάσταση της μνήμης του δρομολογητή μετά από αυτά.

Εντολή	Κατάσταση Scheduler Control (MSB → LSB)							
-	0	0	0	0	0	0	0	0
S1<CR><LF>	0	0	0	0	0	0	0	1
S2<CR><LF>	0	0	0	0	0	0	1	1
S3<CR><LF>	0	0	0	0	0	1	1	1
Q1<CR><LF>	0	0	0	0	0	1	1	0
Q2<CR><LF>	0	0	0	0	0	1	0	0
Q3<CR><LF>	0	0	0	0	0	0	0	0

Επίσης, στο αρχείο lab.log εμφανίζονται 6 απαντήσεις OK<CR><LF>.

Αν έχουν ενεργοποιηθεί και οι τρεις διεργασίες, στο μέχρι στιγμής σύστημα πρέπει να καλείται η μία μετά την άλλη. Θέτοντας από ένα breakpoint στο τέλος κάθε διεργασίας, παρακολουθούνται οι αλλαγές στην κοινή θύρα εξόδου. Επίσης, κοιτώντας την μνήμη κάθε μίας καθώς και το stop watch από το παράθυρο Processor Status, φαίνεται πιο αναλυτικά η λειτουργία του συστήματος.

Τρέχοντας τρεις φορές όλες τις διεργασίες παρουσιάζονται τα παρακάτω αποτελέσματα.

Χρόνος	PORTB	Bcd_counter_data	ring_counter_data	LED_blinking data
0 ms	0x00	0x00	0x80	0x00
1 ms	0x01	0x01	0x80	0x00
6 ms	0x01	0x01	0x01	0x00
13 ms	0xFF	0x01	0x01	0xFF
14 ms	0x02	0x02	0x01	0xFF
19 ms	0x02	0x02	0x02	0xFF
26 ms	0x00	0x02	0x02	0x00
27 ms	0x03	0x03	0x02	0x00
32 ms	0x04	0x03	0x04	0x00
39 ms	0xFF	0x03	0x04	0xFF

Από τον παραπάνω πίνακα πηγάζουν μερικά συμπεράσματα:

- Το διάστημα μεταξύ δύο διαδοχικών breakpoints είναι 1, μετά 5 και μετά 7 ms, δηλαδή ανταποκρίνεται στις καθυστερήσεις των διεργασιών και την σειρά με την οποία εκτελούνται.
- Οι διεργασίες που δεν εκτελούνται κάποια στιγμή, έχουν αποθηκευμένα τα δεδομένα τους και όταν ξαναέρχεται η σειρά τους συνεχίζουν από εκεί που είχαν μείνει. Δηλαδή δεν έχουν αντίληψη του context switch.

Δηλαδή, το σύστημα λειτουργεί όπως έχει σχεδιαστεί.

Ανάλυση & Παρατηρήσεις

Για την ολοκλήρωση του δρομολογητή πρέπει να γίνουν δύο ακόμα αλλαγές.

- Υλοποίηση timeslices.
- Αλλαγή τρέχον διεργασίας με αλλαγή timeslice. Μέχρι στιγμής τρέχει μία φορά η κάθε μια και δίνει τον έλεγχο του επεξεργαστή στην επόμενη.

Πηγές

ATmega16 manual

<http://ww1.microchip.com/downloads/en/devicedoc/doc2466.pdf>

AVR Libc User Manual

<http://savannah.nongnu.org/download/avr-libc/avr-libc-user-manual-2.0.0.pdf.bz2>