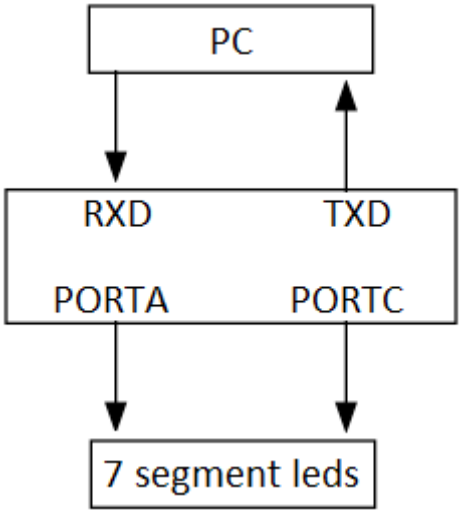
HPY411- Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

Εργαστήριο 3

LAB41145851 **24/10/2020**

Εμμανουήλ Πετράκος ΑΜ 2014030009

Στο παρόν εργαστήριο υλοποιείται το πρόγραμμα οδήγησης της σειριακής συσκευής επικοινωνίας USART και ενσωματώνεται στο ήδη υλοποιημένο σύστημα. Ως είσοδος θεωρούνται τα δεδομένα που φτάνουν στον δέκτη του USART, ενώ ως έξοδος οι απαντήσεις στον πομπό του USART και τα 7 segment LEDs που ελέγχονται από τις θύρες A και C.



Top level schematic

Επεξήγηση προσέγγισης

Έχουν γίνει δύο σημαντικές αλλαγές στον κώδικα του προηγούμενου εργαστηρίου. Πρώτον, στο λεξικό των 7 segment έχει προστεθεί ένας ακόμα χαρακτήρας (A) με τιμή 0xFF και χρησιμοποιείται για να απενεργοποιηθούν όλα τα LED ενός ψηφίου. Κατά την αρχικοποίηση, όλες οι θέσεις μνήμης δεδομένων γράφονται με αυτόν τον χαρακτήρα μέσω της νέας ρουτίνας clr_7seg_data. Η δεύτερη αλλαγή αφορά την αποθήκευση στη μνήμη δεδομένων προς εμφάνιση. Δημιουργήθηκε η ρουτίνα save_to_7seg_data και κάθε φορά που καλείται αποθηκεύει ένα χαρακτήρα στη χαμηλότερη θέση μνήμης δεδομένων, ενώ όλα τα παλιά δεδομένα ανεβαίνουν μία θέση. Ότι υπήρχε στην τελευταία θέση χάνεται.

Για να χρησιμοποιηθεί το USART το πρώτο που πρέπει να ρυθμιστεί είναι το baudrate. Σύμφωνα με τις προδιαγραφές, η επιθυμητή τιμή του είναι τα 9600bps. Αυτό μπορεί να υλοποιηθεί χρησιμοποιώντας τον εσωτερικό μετρητή UBRR. Η τιμή του για ασύγχρονη κανονική λειτουργία υπολογίζεται από τον παρακάτω τύπο.

$$UBRR = \frac{f_{clk}}{16 * BAUD} - 1 = \frac{10\,MHz}{16 * 9600} - 1 = 64.1041$$

Επειδή ο μετρητής δέχεται ακέραιους αριθμούς, έχει την τιμή 64 και το πραγματικό baudrate είναι:

$$BAUD_{closest\,match} = \frac{f_{clk}}{16 * (UBRR + 1)} = \frac{10\,MHz}{16 * (64 + 1)} = 9615{,}3846\,bps$$

Με λάθος: $$Error[\%] = (\frac{BAUD_{closest\,match}}{BAUD} - 1) = +0.16\,\%$$

Σύμφωνα με το manual του ATmega16 (σελίδα 159, πίνακας 61), το λάθος είναι εντός των προτεινόμενων ορίων για οποιαδήποτε ρύθμιση και δεν χρειάζεται κάποια άλλη ενέργεια.

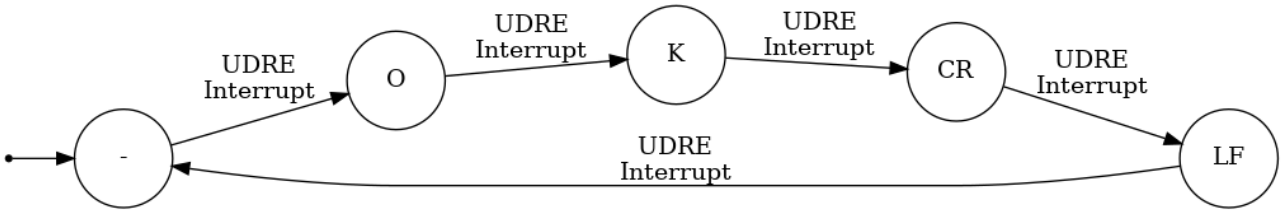
Η αρχικοποίηση του USART γίνεται στην ρουτίνα init_USART_driver κατά την εκκίνηση του προγράμματος. Οι τιμές των καταχωρητών ελέγχου που χρειάζονται για τις προδιαγραφές, φαίνονται στο παρακάτω πίνακα.

Διεύθυνση	Τιμή	Ρύθμιση
UBRR	64	Baudrate = 9600bps @ 10MHz F_CPU
UCSRB		
RXEN	1	Ενεργοποίηση receiver
RXCIE	1	Ενεργοποίηση receiver interrupts
TXEN	1	Ενεργοποίηση transmitter
UCSZ2	0	8 bit frame
UCSRC		
UMSEL	0	Ασύγχρονη λειτουργία
UPM1:0	00	0 bit Parity
UCSZ1:0	11	8 bit frame
USBS	0	1 stop bit

Όποτε ολοκληρώνεται η μετάδοση ενός χαρακτήρα προς τον μικροελεγκτή, εμφανίζεται ένα interrupt. Η εξυπηρέτηση του γίνεται στην ρουτίνα ISR_URXC. Επειδή οι προδιαγραφές έχουν ξεκαθαρίσει ότι τα μηνύματα που έρχονται είναι πάντα σωστά και η αντιμετώπιση των χαρακτήρων είναι ανεξάρτητη των προηγούμενων/επόμενων, δεν χρειάζεται μια μηχανή πεπερασμένων καταστάσεων. Η λειτουργία της ρουτίνας είναι αντιδραστική και φαίνεται στον παρακάτω πίνακα.

Είσοδος	Ενέργεια
C, N	Καθαρισμός δεδομένων μέσω της ρουτίνας clr_7seg_data
A, T, <CR>	Τίποτα
<LF>	Ενεργοποίηση Interrupt απάντησης(UDRIE), αύξηση μετρητή εκκρεμών απαντήσεων
0-9	ASCII → bcd, αποθήκευση αριθμού μέσω της ρουτίνας save_to_7seg_data

Η αποστολή των απαντήσεων γίνεται μέσω των interrupts που ενεργοποιεί ο transmitter όταν ο buffer του έχει διαθέσιμο χώρο(Tx Data Register Empty). Οι απαντήσεις αποτελούνται από τέσσερις χαρακτήρες που πρέπει να σταλθούν σειριακά, για αυτό χρειάζεται η μηχανή πεπερασμένων καταστάσεων που παρουσιάζεται στην παρακάτω εικόνα. Επίσης, χρησιμοποιείται η θέση μνήμης 0x0073 για να αποθηκεύεται η κατάσταση. Στην αρχική κατάσταση δεν γίνεται εκπομπή, ενώ στις υπόλοιπες στέλνεται ο αντίστοιχος χαρακτήρας. Η απάντηση αποτελείται από 4 χαρακτήρες ενώ η μικρότερη εντολή από 3, με αποτέλεσμα να μπορεί να ενεργοποιηθεί νέα απάντηση πριν η FSM φτάσει σε ηρεμία (κατάσταση -) και να χαθεί. Για αυτό, υπάρχει ο μετρητής εκκρεμών απαντήσεων που υλοποιείται στη θέση μνήμης 0x0074. Όποτε ξεκινάει η αποστολή μια απάντησης μειώνεται κατά 1, ενώ αν στο τέλος της έχει την τιμή 0 απενεργοποιείται το interrupt του transmitter.



FSM αποστολής απαντήσεων

Τέλος, φαίνεται η δομή της μνήμης του προγράμματος και με γκρι χρώμα υπογραμμίζονται οι θέσεις που προστέθηκαν στο παρόν εργαστήριο. Συνολικά χρησιμοποιούνται 21 byte μνήμης, 19 για τον οδηγό των 7 segment και 2 για τον οδηγό του USART.

0x0000 ... 0x005F	Register address space
0x0060 ... 0x0067	Data LSB ... Data MSB
0x0068 ...	7 segment format 0 ...
0x0072	7 segment format A
0x0073	Transmitter state
0x0074	Remaining transmits
0x0075 ...	Free Memory

Χάρτης μνήμης

Πειραματική Διαδικασία

Ο έλεγχος του προγράμματος βασίστηκε στον τρόπο προσομοίωσης που αναρτήθηκε στα χρήσιμα έγραφα του μαθήματος. Τα interrupt και τα δεδομένα του receiver ελέγχονται μέσω του usart.stim αρχείου. Επίσης, επειδή ο buffer του transmitter (UDR) είναι write-only και δεν μπορεί να γίνει log, οι απαντήσεις μπαίνουν ταυτόχρονα στον buffer και τον καταχωρητή TCNT2 και γίνεται log σε αυτόν. Έτσι, φαίνονται οι τιμές που μπαίνουν στον buffer μαζί με τις πραγματικές καθυστερήσεις. Τέλος, αφού η υλοποίηση των 7 segments έχει μείνει ίδια πέρα από την εγγραφή στη μνήμη, παρακολουθώντας αυτή μπορούμε να καταλάβουμε τι εμφανίζεται. Χρησιμοποιώντας ένα breakpoint στην ρουτίνα εξυπηρέτησης των interrupt του receiver σε σημείο που φτάνει μόνο όταν έρχεται <LF>, φαίνεται η κατάσταση της μνήμης στο τέλος κάθε μηνύματος.

Στο παρακάτω πίνακα παρουσιάζονται οι εντολές που υλοποιεί το usart.stim καθώς και η κατάσταση της μνήμης μετά από αυτές.

Εντολή	Κατάσταση μνήμης δεδομένων, 0x0060 – 0x0067 (LSB → MSB)							
-	0a	0a	0a	0a	0a	0a	0a	0a
N123<CR><LF>	03	02	01	0a	0a	0a	0a	0a
N1<CR><LF>	01	0a	0a	0a	0a	0a	0a	0a
AT<CR><LF>	01	0a	0a	0a	0a	0a	0a	0a
C<CR><LF>	0a	0a	0a	0a	0a	0a	0a	0a

Στο lab.log φαίνονται οι απαντήσεις προς το PC και η καθυστέρηση μεταξύ των χαρακτήρων:

```
#10058
TCNT2 = 0x4f
#25
TCNT2 = 0x4b
#10399
TCNT2 = 0x0d
#10401
TCNT2 = 0x0a
#25175
TCNT2 = 0x4f
#25
TCNT2 = 0x4b
#10419
TCNT2 = 0x0d
#10401
TCNT2 = 0x0a
#25156
TCNT2 = 0x4f
#25
TCNT2 = 0x4b
#10437
TCNT2 = 0x0d
#10403
TCNT2 = 0x0a
#23134
TCNT2 = 0x4f
#25
TCNT2 = 0x4b
#10443
TCNT2 = 0x0d
#10403
TCNT2 = 0x0a
```

Ανάλυση & Παρατηρήσεις

Από τον τελευταίο πίνακα φαίνεται ότι οι εντολές λειτουργούν σύμφωνα με τις προδιαγραφές. Οι NX<CR><LF> αποθηκεύουν τον αριθμό X ενώ αφαιρούν ότι υπήρχε, η εντολή AT<CR><LF> δεν προκαλεί κάποια αλλαγή στην μνήμη και η εντολή C<CR><LF> καθαρίζει όλα τα δεδομένα.

Στο lab.log φαίνεται ότι το σύστημα στέλνει την απάντηση OK<CR><LF> 4 φορές, μία για κάθε εντολή. Οι μεγάλες καθυστερήσεις (~25000 κύκλοι) οφείλονται στην καθυστέρηση μεταξύ των εισερχόμενων εντολών στο usart.stim. Αντίθετα, οι υπόλοιπες καθυστερήσεις οφείλονται στο σύστημα. Οι μικρές (25 κύκλοι) συμβαίνουν γιατί ο buffer έχει διαθέσιμο χώρο και μπορεί να εξυπηρετήσει δύο interrupt κατευθείαν. Οι καθυστερήσεις των 1ms (~10000 κύκλοι) εμφανίζονται γιατί ο buffer είναι γεμάτος και πρέπει να τελειώσει μια αποστολή πριν δεχτεί νέο χαρακτήρα.

Η πιο υπολογιστικά απαιτητική περίπτωση είναι να έρχονται συνέχεια μηνύματα 8 αριθμών, καθώς η αποθήκευση στη μνήμη είναι η πιο χρονοβόρα λειτουργία των interrupts. Λαμβάνοντας υπόψιν ότι ένα τέτοιο μήνυμα χρειάζεται 11ms για να ολοκληρωθεί, το χρόνο επεξεργασίας του κάθε χαρακτήρα και την απάντηση προς το PC, το USART απασχολεί το CPU περίπου 90 κύκλους κάθε millisecond. Δηλαδή, στη χειρότερη περίπτωση χρειάζεται το 0.9% της υπολογιστικής δύναμης. Πρακτικά, δε θα βρεθεί σε τέτοια κατάσταση και υπό κανονική λειτουργία οι απασχόληση του CPU θα είναι τάξης μεγέθους μικρότερη.

Ο μετρητής των εναπομενόντων μεταδόσεων υλοποιείται σε μια θέση μνήμης 8 bits. Σαν αποτέλεσμα, αν μαζευτούν πάνω από 255 θα υπάρξει overflow και θα χαθούν. Μέχρι στιγμής, οι προδιαγραφές δεν έχουν δείξει ότι το σύστημα υπό κανονική λειτουργία θα βρεθεί σε τέτοια κατάσταση. Πρέπει όμως να ληφθεί υπόψιν σε μελλοντικά εργαστήρια αν θα αλλάξουν την επικοινωνία με το PC.

Πηγές

ATmega16 manual
http://ww1.microchip.com/downloads/en/devicedoc/doc2466.pdf

Simulator2 Stimuli in AVR Studio
https://www.mikrocontroller.net/attachment/160977/stimuli.txt

“Προσομοίωση USART” του Α.Κ.
Χρήσιμα Έγγραφα Μαθήματος