ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ

Εργαστηριακή Άσκηση 3η

Επικοινωνία με σειριακή θύρα

Σωτήριος Μιχαήλ
2015030140

ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

Εισαγωγή

Σκοπός είναι η υλοποίηση ενός προγράμματος το οποίο επικοινωνεί μέσω της optional (communication) θύρας RS232 του STK500 με ένα τερματικό (λ.χ. PuTTY) σε PC, και υλοποιεί μερικές απλές εντολές, καθώς και μία οθόνη 7-segment, η οποία δείχνει αριθμούς τους οποίους έχουμε αποθηκεύσει μέσω μίας εκ των προαναφερθέντων εντολών.

Οι εντολές που υλοποιήθηκαν είναι οι εξής:

Εντολή από PC (ASCII)	**Επεξήγηση Ορίσματος X**	**Ενέργεια Εντολής**	**Απάντηση προς PC (ASCII)**
AT<CR><LF>	-	Απλή Απάντηση OK με το <LF>, καμμία άλλη ενέργεια	OK<CR><LF>
C<CR><LF>		Καθαρισμός οθόνης και απάντηση OK μετά το <LF>	OK<CR><LF>
N<X>CR><LF>	<X>: 1-8 χαρακτήρες ASCII που όλοι αντιστοιχούν σε δεκαδικό ψηφίο	Καθαρισμός οθονης, αποθήκευση ορισμάτων στην μνημη της οθόνης, απάντηση OK μετά το <LF>	OK<CR><LF>

Η βασική ροή του προγράμματος παρουσιάζεται παρακάτω, σε μορφή ψευδοκώδικα:

```
START
    INITIALIZE_STACK( )
    INITIALIZE_UART( )
    INITIALIZE_TIMER_0( )
    WHILE(1) {
        UPDATE_DISPLAY()
        IF (RECEIVE INTERRUPT) {
            RECEIVEINTERRUPTHANDLER()
            REPLY()
        }
    }
```

Παρακάτω παρουσιάζονται περισσότερες λεπτομέρειες για την υλοποίηση κάθε μέρους αυτού του προγράμματος.

Αρχικοποιήσεις

Οι αρχικοποιήσεις για υλοποίηση σε μικροελεγκτή βασίζονται στο ποιόν μικροελεγκτή χρησιμοποιούμε και στην συχνότητα του ρολογιού του. Σε αυτή τη περίπτωση, έχουμε τον μικροελεγκτή ATmega16L και χρησιμοποιούμε το εσωτερικό του ρολόι στο 1MHz. Επομένως, πρέπει να υπολογίσουμε ανάλογα κάποιες τιμές για την αρχικοποίηση της διεπαφής U(S)ART καθώς και του timer0, που χρησιμοποιούμε για την ενημέρωση της οθόνης.

Η επικοινωνία με το PC, καθώς και η ενημέρωση της οθόνης γίνονται μέσω interrupts, επομένως, πρέπει να αρχικοποιήσουμε και τους interrupt handlers στην αρχή του προγράμματος, στις σωστές διευθύνσεις, ανάλογα με το interrupt που έχουμε.

Διευθύνσεις μνήμης

Στις προδιαγραφές του προγράμματος, ορίζεται η αποθήκευση των δεδομένων εισόδου τα οποία προορίζονται να εμφανιστούν στην οθόνη, δηλαδή ψηφία σε ASCII, στη μνήμη RAM του ελεγκτή. Επομένως, γίνεται αρχικοποίηση (allocation) χώρου στη RAM, μεγέθους 8 byte, καθώς αυτός είναι ο μέγιστος αριθμός ψηφίων της εντολής “N<X>”, στη διεύθυνση 0x60 με 0x67 της RAM.

Για την εμφάνιση του ψηφίου που αποθηκεύεται σε μορφή κωδικού ASCII, πρέπει να γίνει μία μετατροπή σε BCD, το οποίο γίνεται με μία μάσκα στα 4 λιγότερο σημαντικά bits, και ένα lookup table, το οποίο περιέχει τους κωδικούς για την εμφάνιση των ψηφίων στην οθόνη 7-segment. Το lookup table βρίσκεται στη ROM, στη θέση μνήμης 0x20 με 0x28.

Για το πρόγραμμα

Πριν από όλες τις υπόλοιπες αρχικοποιήσεις, πρέπει πρώτα να αρχικοποιήσουμε δύο πράγματα. Τη διεύθυνση του κυρίως προγράμματος, του MAIN, το οποίο βρίσκεται στη διεύθυνση 0x00 στη ROM, για την περίπτωση που έχουμε RESET, καθώς και της στοίβας, έτσι ώστε ο ελεγκτής να επιστρέφει στο σωστό σημείο του κυρίως προγράμματος μετά από ένα interrupt.

Για τα interrupts

Για την ορθή λειτουργία της οθόνης, χρησιμοποιούμε τον timer 0 και το interrupt υπερχείλισης του. Ο ελεγκτής για αυτό το interrupt επομένως βρίσκεται στη διεύθυνση 0x12.

Για την επικοινωνία UART μέσω της θύρας RS232, θέλουμε να έχουμε ένα interrupt κάθε φορά που έχουμε μία ολοκληρωμένη μετάδοση στον buffer του UART, τον UDR. Επομένως, θέτουμε τα bits RXC, TXC του καταχωρητή UCSRA σε 1, και τα RXEN, TXEN και RXCIE του καταχωρητή UCSRB σε 1. Το RXCIE είναι το bit το οποίο ενεργοποιεί το interrupt για μία ολοκληρωμένη μετάδοση, και απαιτεί να είναι και το RXC bit ενεργό. Ο ελεγκτής του RXCIE interrupt βρίσκεται στη διεύθυνση 0x16.

Για την διεπαφή U(S)ART

Σε αυτή την εργαστηριακή άσκηση, χρησιμοποιούμε το USART ως απλό UART, δηλαδή, έχουμε ασύγχρονη επικοινωνία με το PC. Ο στόχος για την ταχύτητα επικοινωνίας με το PC είναι τα 9600 baud, επομένως, πρέπει να φορτώσουμε τη κατάλληλη τιμή στον καταχωρητή UBBRL. Για να υπολογίσουμε τη τιμή αυτή, χρησιμοποιείται ο παρακάτω τύπος:

$$UBBRL = \frac{f_{oscillator}}{16\ x\ target_{baud}} = \frac{1\,MHz}{16\,x\,9600} = 5.5 \approx 5$$

Για την σωστή επικοινωνία με το PC μέσω του RS232, πρέπει να ορίσουμε και τη διαμόρφωση των μηνυμάτων που μεταφέρονται. Αυτό γίνεται μέσω των bits UCSZ2..0 του καταχωρητή UCSRC. Καθώς ο καταχωρητής UCSRC μοιράζεται τη διεύθυνσή του με έναν άλλον καταχωρητή για εξοικονόμηση διευθύνσεων, χρειάζεται να θέσουμε και το URSEL bit σε 1, έτσι ώστε να γράψουμε τις σωστές τιμές για τα UCSZ2..0. Σε αυτήν την άσκηση, τα μηνύματα έχουν 1 stop bit και δεν έχουν parity. Επομένως, γράφουμε 1 στα bits UCSZ1, UCSZ0 και URSEL.

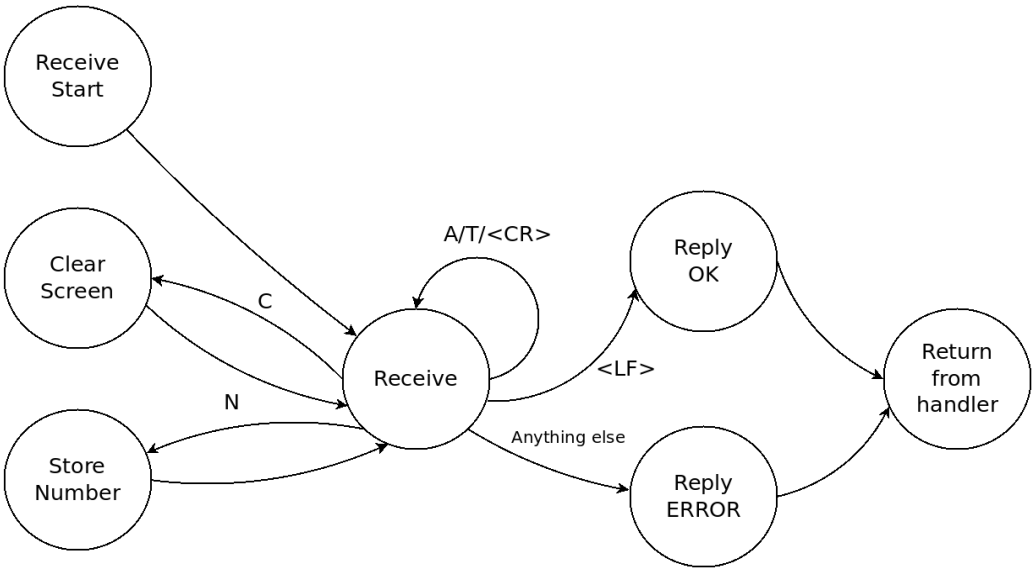
Για τον timer 0

Μέσω του μετρητή αυτού, θέλουμε να ανανεώνουμε την οθόνη κάθε 4 ms. Καθώς έχουμε ρολόι 1 MHz αντί των 10 MHz, έπρεπε να γίνει μία αλλαγή στην αρχικοποίηση του timer, σε σχέση με τις 2 προηγούμενες εργαστηριακές ασκήσεις. Χρησιμοποιήθηκε ο prescaler $f_{clk}/1024$, θέτοντας 1 τα bit CS00 και CS02, και η σωστή τιμή του TCCR0 για overflow κάθε 4 ms προκύπτει από τον τύπο:

$$f_{timer0} = f_{clk}/1024 \approx 1\,kHz \Rightarrow T_{timer0} \approx 1\,ms \Rightarrow 1\,ms\ x(256 - TCCR0) = 4\,ms \Rightarrow TCCR0 = 4$$

Διαχείριση RXCIE interrupt και είσοδος μέσω RS232

Η βασική προσθήκη στην υλοποίηση της $2^{ης}$ εργαστηριακής άσκησης, είναι ο RXCIE interrupt handler, ο οποίος ουσιαστικά υλοποιεί την είσοδο δεδομένων από το PC μέσω τις θύρας RS232, και επομένως, όλες τις εντολές των προδιαγραφών, μέσω κλήσεων κατάλληλων βοηθητικών συναρτήσεων. Ο διαχειριστής αυτού το interrupt είναι ουσιαστικά μία μηχανή πεπερασμένων καταστάσεων, η οποία αλλάζει κατάσταση ανάλογα τον χαρακτήρα που λαμβάνεται από το UDR. Η βασική ροή του διαχειριστή παρουσιάζεται με το παρακάτω διάγραμμα καταστάσεων:



Για τον καθαρισμό της οθόνης μέσω της εντολής “C<CR><LF>”, μία βοηθητική συνάρτηση θέτει την οθόνη σε 0xFF, και καθαρίζει την μνήμη στην οποία βρίσκονται τα ψηφία εισόδου αποθηκευμένα.

Η αποθήκευση εισόδου περιλαμβάνει επίσης χρήση μίας βοηθητικής συνάρτησης, η οποία ολισθαίνει τα bytes στη μνήμη RAM κατά μία θέση κάθε φορά, έτσι ώστε τα ψηφία τα οποία έρχονται κατά μία σειρά να βρίσκονται στη σωστή θέση μνήμης έτσι ώστε να εμφανίζονται χωρίς “trailing zeroes” στην οθόνη.

Έξοδος μέσω RS232

Οι δύο συναρτήσεις απάντησης προς το PC, όπου η REPLY απαντά “OK<CR><LF>” για μία επιτυχή αναγνώριση και εκτέλεση εντολής, και η ERREPLY απαντά “ERROR<CR><LF>” για μία αποτυχία αναγνώρισης εντολής, υλοποιούνται με την TRANSMT, μία συνάρτηση η οποία παίρνει ως όρισμα έναν χαρακτήρα στο R17, και τον αποστέλλει στο UDR. Η συνάρτηση TRANSMT παρουσιάζεται ως αυτούσιο κομμάτι κώδικα:

```
TRANSMT:
        SBIS  UCSRA,UDRE
        RJMP  TRANSMT
        OUT   UDR,R17
        RET
```

Οθόνη 7-segment

Η υλοποίηση της οθόνης, καθώς και του timer που χρησιμοποιείται για την ανανέωσή της, έχει καλυφθεί εκτενώς στις αναφορές των προηγούμενων δύο εργαστηρίων. Η μόνη αλλαγή που έγινε σε αυτό το κομμάτι κώδικα ήταν η διαφορετική αρχικοποίηση του timer, καθώς τώρα έχουμε το πραγματικό ρολόι του 1 MHz του ATmega16 στην αναπτυξιακή πλατφόρμα STK500, αντί του 10 MHz του προσομοιωτή του Atmel Studio 7.

Καταχωρητές

Για την ευκολότερη κατανόηση του κώδικα, παρατίθεται ένας πίνακας με τους καταχωρητές που χρησιμοποιούνται στο πρόγραμμα, και την ειδική χρήση του καθενός, εάν υπάρχει:

Καταχωρητής	Χρήση
R16	Γενικής προσωρινής χρήσης
R17	Όρισμα για τη συνάρτηση μετάδοσης
R18	Τιμή που ανακτάται από τη RAM
R19	Γενικής προσωρινής χρήσης
R20	Αριθμός ψηφίων που έχουν αποθηκευθεί
R21	Γενικής προσωρινής χρήσης
R22	Γενικής προσωρινής χρήσης
R23	Αριθμός ψηφίων που δεν έχουν ανανεωθεί στην οθόνη
R24	Γενικής προσωρινής χρήσης

Οι καταχωρητές από το R25 εώς το R30 είναι ειδικοί καταχωρητές των 16 bit στον AVR, που χρησιμοποιούνται ως pointers. Σε αυτό το πρόγραμμα, χρησιμοποιείται ο καταχωρητής-δείκτης X για την ανάκτηση και την αποθήκευση δεδομένων στη RAM, και ο καταχωρητής-δείκτης Z για την αντιστοίχηση ενός ψηφίου BCD με τον κατάλληλο κωδικό για 7-segment display.

Διεξαγωγή εργαστηρίου – προσομοίωση και επιβεβαίωση λειτουργίας

Σε αυτή την εργαστηριακή άσκηση εμφανίζεται ένας περιορισμός της προσομοίωσης, καθώς δεν είναι δυνατό να προσομοιώσουμε δεδομένα τα οποία εισέρχονται από τη θύρα RS232. Επομένως, έπρεπε να γίνουν κάποιες αλλαγές στον κώδικα (οι οποίες αναφέρονται σε ανακοίνωση καθώς και σε αρχείο που έχει αναρτηθεί στα "Χρήσιμα Έγγραφα" του μαθήματος). Η κύρια αλλαγή είναι η χρήση ενός καταχωρητή διαφορετικού του UDR, καθώς αυτός ο καταχωρητής δεν είναι δυνατό να εγγραφεί με άλλο τρόπο εκτός της θύρας RS232. Με αυτές τις αλλαγές, καθώς και με τη χρήση ενός stimulus file, το οποίο αναθέτει τιμές σε συγκεκριμένους καταχωρητές μετά το πέρας κάποιων κύκλων του ρολογιού τους οποίους εμείς ορίζουμε (σε αυτή τη περίπτωση, 2000 κύκλοι), το πρόγραμμα έλαβε ως είσοδο τις τρεις εντολές των προδιαγραφών, και για τις εντολές AT και C στο αρχείο καταγραφής του προσομοιωτή εγγράφεται η απάντηση "OK<CR><LF>", όπως παρατίθεται:

```
#15
TCNT2 = 0x4f
#12
TCNT2 = 0x4b
#12
TCNT2 = 0x0d
#12
TCNT2 = 0x0a
```

Η απάντηση του προγράμματος γράφεται στον καταχωρητή TCNT2, καθώς ο προσομοιωτής δε μπορεί να εμφανίσει τα δεδομένα του UDR.

Για την εντολή "N<X>", η οποία αποθηκεύει ψηφία για την εμφάνισή τους στην οθόνη, ο προσομοιωτής δε γράφει κάποια απάντηση στο αρχείο καταγραφής. Πιθανό αίτιο αυτού είναι η εσφαλμένη λειτουργία του καταχωρητή-δείκτη X, ο οποίος ανακτά δεδομένα από τη μνήμη RAM, και βγαίνει από τα όρια των δεδομένων που έχουν τεθεί.

Όσον αφορά την αρχικοποίηση των καταχωρητών του UART, η ορθή λειτουργία της φαίνεται στο παρακάτω στιγμιότυπο του προσομοιωτή: