

Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

αναφορα ΔΕΥΤΕΡΗΣ εργαστηριακης ασκησης

Καλογεράκης Στέφανος | Ζερβάκης Αρης

Σκοπός εργαστηριακής άσκησης

Η πρώτη εργαστηριακή άσκηση είχε στόχο την περετέρω εξοικείωση με το περιβάλλον ανάπτυξης για μικροελεγκτή AVR, την σειριακή θύρα RS-232 και με το κατέβασμα κώδικα στην αναπτυξιακή πλακέτα STK500.

Περιγραφή

Aρχικά, έπρεπε να δημιουργήσουμε μια σειριακή θύρα και με τη χρήση αυτής, να στέλνουμε και να λαμβάνουμε απλούς χαρακτήρες ASCII. Έπειτα, να συνδέσουμε τον υπολογιστή μας με τη σειριακή θύρα με τερματικό στον υπολογιστή μας, και τέλος, χρησιμοποιώντας τον timer του πρώτου εργαστηρίου να στέλνουμε μία τελίτσα στο PC κάθε δευτερόλεπτο.

Υλοποίηση

Πρώτο μας μέλημα ήταν η σύνδεση της πλακέτας με τον υπολογιστή, εγκαθιστώντας τους σωστούς drivers για το καλώδιο RS232-to-usb, καθώς και η βραχυκύκλωση των σωστών pins πάνω στο STK500 ώστε να δούμε τα σωστά αποτελέσματα κατα την αποστολή και τη λήψη συμβόλων. Έτσι βραχυκυκλώσαμε τα pins ISP6PIN και SPROG3 καθώς και τα PD0, PD1 με τα RXD και ΤΧD αντίστοιχα. Κατά την υλοποίηση μας χωρίσαμε τη λειτουργία της σειριακής θύρας σε 2 επιμέρους λειτουργίες. Την αποστολή και τη λήψη χαρακτήρων.

***Αποστολή Συμβόλων***

Πρώτο μας μέλημα ήταν να επιλέξουμε την ταχύτητα μετάδοσης (BAUD Rate). Αναζητώντας στο datasheet του atmega16L βρήκαμε την κατάλληλη σχέση, την εφαρμόσαμε και καταλήξαμε σε Rate=9600Baud.



Για τη σύνδεση στο τερματικό του υπολογιστή μας με τη σειριακή θύρα για την αποστολή και λήψη συμβόλων επιλέξαμε το πρόγραμμα Tera Term, καθαρά για λόγους ευκολίας και εύκολης παραμετροποίησης.

Για την ενεργοποήση του transmitter έπρεπε να αλλάξουμε 2 απο τα 3 control registers του. Πρώτα το δεύτερο, το transmitter enable και έπειτα το τρίτο για το Character Size.



Για το transmit πρέπει να προσθέσουμε μια transmission buffer. Eτσι δημιουργούμε ένα array χαρακτήρων με 2 indexes, το read position και το write position για να καθορίσουμε την αρχή και το τέλος του μηνύματος. Έπειτα δημιουργούμε 2 μεθόδους, Την appendSerial και την serialWrite.

H appendSerial είναι υπεύθυνη για να προσθέτει χαρακτήρες στη buffer και η serialWrite για να προσθέτει στη buffer strings χαρακτήρων.

Τέλος ορίσαμε το Interrupt Vector (USART\_TX\_vect) το οποίο ειναι αναγκαίο για να καθορίσουμε το τέλος του transmission, όταν όλοι οι χαρακτήρες στο UDR register έχουν γίνει shift.

***Λήψη Συμβόλων***

Για τη λήψη συμβόλων ακολουθήσαμε παρόμοια διαδικασία με την αποστολή. Αρχικά έπρεπε να ενεργοποιήσουμε τον receiver. Ενεργοποιήσαμε λοιπόν το receiver enable, ρυθμίσαμε το parity όπως ακριβώς και στον transmitterr, καθώς και το Character Size.



Για το receive δημιουργήσαμε και πάλι 2 μεθόδους, την getChar και την peekChar, όπως και μια buffer με 2 indexes όπως και στην περίπτωση αποστολής. Αυτή τη φορά ο στόχος μας είναι να αφαιρούμε,δηλαδή να διαβάζουμε τους χαρακτήρες απο τη buffer, σε αντίθεση με το transmission.

Η μέθοδος getChar είναι υπεύθυνη για να διαβάζει τον χαρακτήρα που της υποδυκνύει το index στην buffer ενώ η μέθοδος peekChar διαβάζει τον χαρακτήρα αν και μόνο αν αυτός βρίσκεται στην κορυφή της buffer.

Σύνοψη

Στο δευτερο εργαστήριο του μαθήματος σε επαφή με τη σειριακή θύρα και την αποστολή αλλα και τη λήψη απλών συμβόλων ASCII, ενώ εξοικιωνόμαστε περισσοτερο με το περιβάλλον του atmel studio αλλά και με την πλακέτα