



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 411 – ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021

Καθ. Α. Δόλλας

PROJECT – ΕΝΑ ΠΑΙΧΝΙΔΙ SUDOKU

ΕΚΔΟΣΗ : 1.0

Προθεσμία: Πέμπτη 23 Δεκεμβρίου 2021 (τελευταία μέρα μαθημάτων πριν κλείσουμε για Χριστούγεννα) για την τελική ολοκλήρωση αλλά και προθεσμία για το Milestone Ηλεκτρονική υποβολή στο eClass και συμμετοχή στο ηλεκτρονικό τουρνουά χρονομέτρησης λύσεων

Σκοπός

Σκοπός του Project είναι ένα παιχνίδι Sudoku που επιλύει ένα πλέγμα 9X9 στο STK500. Η διεπαφή είναι με σειριακή θύρα, μέσω της οποίας η πλατφόρμα δέχεται εντολές και δεδομένα εισόδου (σε ASCII) και επιστρέφει το συμπληρωμένο πίνακα, ή κωδικούς σφάλματος ή κάποιο μήνυμα (π.χ. τελείωσα, κλπ.).

Υπάρχουν πολλοί αλγόριθμοι, καθώς και κώδικες στο διαδίκτυο, **αλλά εσείς πρέπει να γράψετε κώδικα από την αρχή** (μπορείτε βεβαίως να πάρετε τον αλγόριθμο από την βιβλιογραφία ή τις δημοσιευμένες εφαρμογές).

Για να εξοικειωθείτε με το παιχνίδι Sudoku μπορείτε να επισκεφτείτε το site: <https://www.websudoku.com/> ή άλλο συναφές. Τα βήματα και τα παραδοτέα παρατίθενται παρακάτω.

Λίγα λόγια για το παιχνίδι Sudoku

Οι κανόνες του Sudoku είναι απλοί, έχουμε ένα πίνακα 9X9 που αποτελείται από εννέα πίνακες 3X3. Κάθε πίνακας 3X3 περιλαμβάνει στα εννέακελιά του όλους τους αριθμούς 1-9, ακριβώς από μία φορά τον καθένα. Στον πλήρη πίνακα 9X9, κάθε στήλη αλλά και κάθε σειρά περιλαμβάνουν από μία φορά κάθε αριθμό από 1-9. Η επίλυση του πίνακα περιλαμβάνει αρχικά δεδομένα (που θεωρούμε ότι οδηγούν πάντα σε τουλάχιστον μία λύση). Τα αρχικά δεδομένα τιμές για κάποια κελιά (αυτές οι τιμές δεν μπορούν να αλλάξουν), καθώς και 'κενά' κελιά, οι τιμές των οποίων πρέπει να προσδιοριστούν. Όσο λιγότερα κελιά είναι συμπληρωμένα τόσο δυσκολότερη είναι η επίλυση του Sudoku.

Παράδειγμα:

Ενδεικτικός Αρχικός Πίνακας –
δεν προβλέπει συντεταγμένες κελιών

5		6		4	3			9
					1			
3			9		8	7	4	
	4	5		1		3	8	6
6								1
8	1	2		3		5	9	
	7	8	1		2			3
			3					
2			6	8		1		4

Λύση του παραπάνω πίνακα, με bold έχουμε τα «αγκυροβολημένα» κελιά της εκφώνησης

5	8	6	7	4	3	2	1	9
7	9	4	5	2	1	6	3	8
3	2	1	9	6	8	7	4	5
9	4	5	2	1	7	3	8	6
6	3	7	8	9	5	4	2	1
8	1	2	4	3	6	5	9	7
4	7	8	1	5	2	9	6	3
1	6	9	3	7	4	8	5	2
2	5	3	6	8	9	1	7	4

Πίνακας όπως θα τον χρησιμοποιήσουμε - Έχει συντεταγμένες **X** (κόκκινες) και **Y** (πράσινες - θετική κατεύθυνση προς τα κάτω).

	1	2	3	4	5	6	7	8	9
1	5		6		4	3			9
2						1			
3	3			9		8	7	4	
4		4	5		1		3	8	6
5	6								1
6	8	1	2		3		5	9	
7		7	8	1		2			3
8				3					
9	2			6	8		1		4

Παρατηρούμε ότι η αρχική δομή του πίνακα έχει μεταβλητό αριθμό ορισμένων κελιών ανά πίνακα 3X3, καθώς και ανά στήλη και σειρά. Αυτό εν γένει θα ισχύει και για μας.

Κανόνες για το δικό μας παιχνίδι

Για να έχουμε ενιαία πλατφόρμα, θα σας δοθεί κρύσταλλος 10MHz (με μικρή επιφύλαξη για τυχόν προβλήματα). Με βάση αυτόν τον κρύσταλλο θα πρέπει να καθορίσετε τους κώδικές σας. Ο χρόνος απόκρισης του συστήματός σας καθορίζεται από τον χρόνο τελικής επίλυσης του πίνακα. Η επιθυμητή ταχύτητα και το πρωτόκολλο της σειριακής θύρας είναι **9600baud, 8Bits, 1Stop Bit, No Parity**, αν και μπορεί να ανακοινωθεί κάποια διαφορετική σε περίπτωση προβλημάτων με το καλώδιο USB - σειρακή.

Το αναπτυξιακό STK-500 έχει διακόπτες αλλά και οκτώ LED (LE00 - LE07). Όσο λειτουργεί το σύστημά σας τα LED πρέπει να δείχνουν την πρόοδο της επίλυσης του πίνακα Sudoku. Αυτό σημαίνει ότι θέλουμε το LED00 να είναι αναμμένο όταν έχουμε 10 ή περισσότερα κελιά με κάποια μη μηδενική τιμή (που μπορεί και να είναι λάθος και να αλλάξει αργότερα) το LED01 αν έχουμε 20 ή περισσότερα κελιά, κοκ., με το LE07 να είναι αναμμένο με 80 ή περισσότερα κελιά προφανώς με 80 κελιά από τα 81 να έχουν κάποια τιμή είτε σχεδόν ολοκληρώθηκε η εκτέλεση, ή θα κάνουμε (ανάλογα και με τον αλγόριθμο backtrack και θα μειωθούν τα κελιά με μη μηδενική τιμή). Ουσιαστικά τα LED θα δείχνουν μία «μπάρα» με την πρόοδο της επίλυσης του Sudoku. ΒΟΗΘΗΜΑ: διαλέξετε κάποια θύρα για τα LED που να μην «πέφτει πάνω» σε TXD, RXD ή άλλα σήματα που σας χρειάζονται, ώστε να κάνετε την σύνδεση μία φορά και να μην χρειάζεται να βάζετε Jumpers μετά τον προγραμματισμό του AVR.

Το πρωτόκολλο επικοινωνίας, αρχικοποίησης και παιξίματος είναι το παρακάτω, με προοπτική να προστεθούν ή ενδεχόμενα να αλλάξουν στο μέλλον λίγες εντολές εάν προκύπτει ότι χρειάζονται.

Εντολή από PC ή απάντηση από το PC (ASCII)	Ενέργεια Εντολής	Απάντηση προς PC (ASCII) ή εντολή από AVR προς PC
AT<CR><LF>	Απλή Απάντηση OK μετά το <LF>, καμμία άλλη ενέργεια	OK<CR><LF>
C<CR><LF>	Καθαρισμός δεδομένων πίνακα (αρχικοποίηση κελιών στο 0) και οθόνης LED (όλα τα LED σβυσμένα) και απάντηση OK μετά το <LF>, ισοδύναμο με αρχικοποίηση παιχνιδιού	OK<CR><LF>
N<X><Y><VALUE><CR><LF>	Τιμές κελιών, απάντηση από τον AVR OK μετά το <LF> <X>, <Y>, <VALUE>: [1-9] χαρακτήρας ASCII που όλοι αντιστοιχούν σε δεκαδικό ψηφίο	OK<CR><LF>
P<CR><LF>	Play New Game (ξεκινάμε να λύνουμε τον πίνακα, δεν υπάρχουν άλλα δεδομένα).	OK<CR><LF>
S<CR><LF>	Από AVR: Done (τελείωσε η επεξεργασία, ο AVR είναι έτοιμος να αρχίσει να αποστέλλει αποτελέσματα), η απάντηση S από το PC σημαίνει «στείλε το πρώτο αποτέλεσμα κελιού»	D<CR><LF>
T<CR><LF>	Τιμές κελιών, απάντηση T από το PC μετά το <LF> κάθε μηνύματος ώστε να ξεκινήσει ο AVR να στέλνει το επόμενο. <X>, <Y>, <VALUE>: [1-9] χαρακτήρας ASCII που όλοι αντιστοιχούν σε δεκαδικό ψηφίο	N<X><Y><VALUE><CR><LF>
OK<CR><LF>	Done sending results από AVR, η απάντηση OK από PC σημαίνει ότι τα έλαβα όλα	D<CR><LF>
	Τα παρακάτω «αχρείαστα να 'ναι»	
B<CR><LF>	Break – Σταμάτησε τους υπολογισμούς ή την μετάδοση	OK<CR><LF>

	αποτελεσμάτων αλλά μην αρχικοποιήσεις τους πίνακες ακόμη – κάτι σαν warm start της εφαρμογής (π.χ. για παρατήρηση προόδου αν φαίνεται ότι «κρέμασε» η εφαρμογή)	
D<X><Y><CR><LF>	Debug – Στείλε μου τα περιεχόμενα του κελιού (X,Y). Η απάντηση N περιλαμβάνει τις συντεταγμένες και την τιμή του κελιού (ΠΡΟΧΟΧΗ: η τιμή μπορεί να περιλαμβάνει και το 0 αν δεν έχει ακόμη καθοριστεί η τιμή του κελιού)	N<X><Y><VALUE>CR><LF>

Διεξαγωγή Έργου

Από λειτουργικής απόψεως, η διαδικασία είναι η εξής: κατεβάζουμε κώδικα στον AVR, και αλλάζουμε την σειριακή θύρα από την διεπαφή (connector) προγραμματισμού στην διεπαφή όπου το PC επικοινωνεί μέσω TXD/RXD με το σύστημά μας. Κατόπιν ξεκινάει η επικοινωνία από μία εφαρμογή που είναι υπό ανάπτυξη με τον AVR, π.χ. με AT για να ξεκινήσει επικοινωνία. Όταν φορτωθούν όλα τα μη μηδενικά κελιά κάθε παιχνιδιού (θα υπάρχουν πολλά παιχνίδια, διαφορετικής δυσκολίας, για κάθε ομάδα). Η εφαρμογή στο PC (το ίδιο PC θα χρησιμοποιηθεί για την εξέταση όλων των projects) θα κατεβάζει τα «γνωστά» κελιά, και θα μετράει τον χρόνο επίλυσης του Sudoku. Ο χρόνος θα μετριέται από το P (Play) που θα στέλνει το PC έως το D (Done) που θα λαμβάνει ως απάντηση από τον AVR. Εδώ υπάρχει ένα ενδιαφέρον θέμα που ανακύπτει για το αν μετράμε τον χρόνο του <CR><LF> – θα το συζητήσουμε στο μάθημα αυτό. Μετά το D(One) το PC θα ανεβάζει τα αποτελέσματα από τον AVR και θα ελέγχει ότι η λύση είναι σωστή. Οποιαδήποτε σωστή λύση μετράει ως τέτοια, χωρίς απαίτηση να ταιριάζει με εκείνη που είχε προϋπολογιστεί ή να πρέπει να βρεθούν άνω της λίας λύσεις.

Τα datasets θα αντιστοιχούν σε χρόνο εκτέλεσης κάθε Sudoku από λίγα δευτερόλεπτα έως λίγα λεπτά, με επιφύλαξη ότι μπορεί να θέλουμε να δοκιμάσουμε και πιο ακραίες περιπτώσεις και (συμπληρωματικά) να δώσουμε και Sudoku που χρειάζονται δεκάδες λεπτά εκτέλεσης. Η ιδέα είναι ότι μεταξύ των πιο «αργών» κωδίκων και των πιο «γρήγορων» κωδίκων μπορεί να έχουμε ακόμη και 10:1 χρόνο εκτέλεσης, οπότε θα θέλουμε να δοκιμάσουμε και πιο δύσκολες περιπτώσεις. Επίσης, ενδέχεται κάποιοι αλγόριθμοι να δουλεύουν καλύτερα σε «δύσκολα» παιχνίδια, οπότε θα

θέλουμε να δούμε την πρόοδο σε χρόνο εκτέλεσης όσο το παιχνίδι γίνεται πιο δύσκολο.

Το Project θα έχει ένα Milestone:

- M1 (Δευτέρα 6 Δεκεμβρίου, στο eClass) Υλοποίηση διεπαφής σειριακής θύρας για υποστήριξη όλων των εντολών και απαντήσεων, υλοποίηση ανάμματος LED, υλοποίηση πίνακα στην εσωτερική μνήμη, μαζί με τις δομές δεδομένων για χρήση μνήμης και μελέτη αλγορίθμου που θα υλοποιηθεί. Ενδεχόμενη μερική υλοποίηση του έργου, χωρίς κάποια συγκεκριμένη απαίτηση.
- Ολοκλήρωση Project: Τετάρτη 22 Δεκεμβρίου, με υποβολή στο eClass όλων των κωδίκων και της τελικής αναφοράς.

Παρουσίαση /Εξέταση

Η εξέταση των ομάδων στο Project με μορφή τουρνουά θα γίνει για όλους/όλες μαζί σε μία ημέρα, πιθανόν στην εβδομάδα από το πέρας των μαθημάτων και πριν την εξεταστική. Η παρουσία όλων σε ολόκληρη την ημέρα αυτή είναι υποχρεωτική.

Η αναφορά που θα κατατεθεί μαζί με τους κώδικες πρέπει να είναι πλήρης, με κατάλληλες ενότητες, με block diagrams του κώδικα, με ανάλυση του αλγορίθμου, με ανάλυση χρήσης της μνήμης του AVR και των σχετικών δομών, και πρέπει να συνοδεύεται στην on-line υποβολή από όλους τους κώδικες, πλήρεις και σε σωστή τεκμηρίωση (block comments, line comments) στην Αγγλική γλώσσα. Το Milestone υποβάλλεται on-line με την δική του αναφορά, καθώς και τους σχετικούς κώδικες αλλά και την διαδικασία επιβεβαίωσης αυτών. Η τελική αναφορά περιλαμβάνει (ξανά) και όσα μέρη των αναφορών Milestone και Εργασιρών χρειάζονται ώστε να είναι πλήρης και να διαβάζεται αυτοτελώς.

ΠΡΟΣΟΧΗ (τα ξέρετε, αλλά τα ξαναθυμίζουμε)!

- 1) Η προεργασία να είναι σε ηλεκτρονική μορφή και μαζί με αρχεία με κώδικες που να μπορούμε να εκτελέσουμε. Το αρχείο πρέπει να το υποβάλλετε στο eClass.
- 2) Η έλλειψη προετοιμασίας ή επαρκούς τεκμηρίωσης οδηγεί σε απόρριψη.
- 3) Η διαπίστωση αντιγραφής σε οποιοδήποτε σκέλος του Project οδηγεί στην απόρριψη από το σύνολο των εργαστηριακών ασκήσεων, άρα και του μαθήματος. Αυτό γίνεται οποιαδήποτε στιγμή στη διάρκεια του εξαμήνου. Ως αντιγραφή νοείται και μέρος της αναφοράς, π.χ. σχήματα.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ! ☺