

1 Εισαγωγή

Σκοπός του Project είναι η επίλυση ενός παιχνιδιού Sudoku (9x9 πλέγμα) χρησιμοποιώντας έναν μικροελεγκτή AVR. Η διεπαφή του μικροελεγκτή με τον εξωτερικό κόσμο υλοποιείται με χρήση της σειριακής θύρας RS232 και με τη χρήση του τερματικού προγράμματος putty. Η επίλυση του Sudoku θα πραγματοποιηθεί σε μεταγενέστερο χρόνο χρησιμοποιώντας έναν αλγόριθμο οποίος και θα αποφασίσει με βάση την απόδοση του.

2 Τεχνολογία

Για την υλοποίηση του project χρησιμοποιήθηκε η πλακέτα STK500 με τον μικροελεγκτή ATmega16L με συχνότητα ρολογιού 10MHz (εξωτερικός κρύσταλλος). Τα μοντέλα του μικροελεγκτή και του κρυστάλλου τηρούν προδιαγραφές οι οποίες είναι κοινές για όλες τις ομάδες. Η συγγραφή του κώδικα πραγματοποιήθηκε στη γλώσσα C με τον avr-gcc compiler, ενώ για την προσομοίωση του κώδικα χρησιμοποιήθηκε η πλατφόρμα Microchip Studio. Για την επικοινωνία της πλακέτα με τον εξωτερικό κόσμο χρησιμοποιήθηκε ένα καλώδιο USB σε Serial RS232 (βύσμα DB9).

3 Περιγραφή της υλοποίησης

Το πρότζεκτ αποτελείται από 4 μέρη τα οποία είναι η διεπαφή της σειριακής θύρας, η επεξεργασία των εντολών, ο αλγόριθμος επίλυσης του Sudoku και η οθόνη η οποία εμφανίζει τη πρόοδο επίλυσης του Sudoku.

3.1 Σειριακή Θύρα RS232

Η επικοινωνία του μικροελεγκτή με το χρήστη επιτυγχάνεται μέσω της σειριακής θύρας ορίζοντας τις παραμέτρους της έτσι ώστε να τηρούν τις προδιαγραφές:

- BAUD rate 9600 bps
- 8 data bits
- No parity
- 1 stop bit

Για το σετάρισμα του BAUD rate στην τιμή 9600 φορρώθηκε στον 16-bit καταχωρητή UBRR η τιμή $F_{CPU}/16/USART_BAUDRATE - 1 = 10 * 10^6 / 16 / 9600 - 1 = 64.10$, το οποίο απλοποιείται στο 64 (ο UBRR δέχεται ακέραιες τιμές).

Στη συνέχεια, ενεργοποιείται η λήψη και η μετάδοση δεδομένων μέσω των flags RXEN και TXEN του UCSRB. Τα interrupt flags τόσο της λήψης όσο και της μετάδοσης, ενεργοποιούνται προκειμένου ο μικροελεγκτής να λαμβάνει και να μεταδίδει δεδομένα με χρήση interrupts αντί για polling. Το τελευταίο δεν προτιμήθηκε λόγω χαμηλότερης απόδοσης.

Τέλος, για την ρύθμιση 8 data bits, αρχικά ενεργοποιείται ο UCSRC (ίδιος με τον UBRRH) με το URSEL bit και τίθενται τα UCSZ0 και UCSZ1 στην τιμή 1. Για τη ρύθμιση του parity και του stop bit, δεν απαιτείται κάποια ενέργεια καθώς είναι αυτόματα ρυθμισμένα στην επιθυμητή τιμή (0).

Listing 1: Κώδικας αρχικοποίησης UART

```
static inline void initUART()  
{  
    // Load upper 8-bits of the baud rate value into  
    // the high byte of the UBRR register  
    UBRRH = (BAUD_PRESCALE)>>8;  
    // Load lower 8-bits of the baud rate value into  
    // the low byte of the UBRR register  
    UBRL = BAUD_PRESCALE;  
    // Enable reception and transmission circuitry  
    UCSRB = (1 << RXEN) | (1 << TXEN);  
    // Use 8-bit character sizes  
    UCSRC = (1 << URSEL) | (1 << UCSZ1) | (1 << UCSZ0);  
    // Enable the USART RXC and TXC interrupts  
    UCSRB |= (1 << RXCIE) | (1 << TXCIE);  
}
```

3.2 Επεξεργασία των εντολών

Οι εντολές που υποστηρίζει το ενσωματωμένο σύστημα παρατίθενται στο Table 1. Κατά το χρόνο σύνταξης της παρούσας αναφοράς, οι εντολές εκτελούνται στο Timer 2 interrupt. Συγκεκριμένα, κάθε φορά που εκτελείται το Timer 2 interrupt, διαβάζεται ο receive buffer και αποκωδικοποιείται μία νέα εντολή, εφόσον υπάρχει. Η εκτέλεση των εντολών, γίνεται και αυτή στο timer interrupt έχει πλεονεκτήματα και μειονεκτήματα.

Εντολή από PC ή AVR	Ενέργεια εντολής	Απάντηση προς PC ή AVR
"AT\r\n" (Από PC)	Έναρξη επικοινωνίας με τον AVR	"OK\r\n" (Προς AVR)
"C\r\n" (Από PC)	Καθαρισμός πίνακα και οθόνης	"OK\r\n" (Προς AVR)
"NXYVAL\r\n" (Από PC)	Τιμή κελιού (VAL) sudoku[X][Y], X,Y ∈ [1,9]	"OK\r\n" (Προς AVR)
"P\r\n" (Από PC)	Ξεκινάει να παίζει-λύνει το παιχνίδι	"OK\r\n" (Προς AVR)
"S\r\n" (Από PC)	Ρωτάει τον AVR αν τέλειωσε η λύση του Sudoku	"D\r\n" αν έχει τελειώσει (Προς AVR)
"T\r\n" (Από PC)	Στέλνει τμές κελιών στο PC, μία τιμή κάθε φορά που λαμβάνει αυτή την εντολή	"NXYVAL\r\n" (Προς AVR)
"D\r\n" (Από AVR)	Ρωτάει τον AVR αν του έστειλε όλα τα δεδομένα	"OK\r\n" (Προς PC)
"B\r\n" (Από PC)	Σταμάτησε τους υπολογισμούς ή τη μετάδοση αποτελεσμάτων (warm start)	"OK\r\n" (Προς AVR)
"DXYVAL\r\n" (Από PC)	Στέλνει τα περιεχόμενα του κελιού sudoku[X][Y]	"NXYVAL\r\n" (Προς AVR)

Table 1: Σετ εντολών

3.3 Αλγόριθμός Επίλυσης Sudoku

3.4 Οθόνη - Progress Bar

Για την απεικόνιση της προόδου επίλυσης του Sudoku χρησιμοποιήθηκαν τα LEDs κοινής ανόδου (CA) της πλακέτας. Συγκεκριμένα κατά τη διάρκεια επίλυσης του Sudoku πραγματοποιείται περιοδικά έλεγχος σχετικά με το πόσα κελιά του πίνακα είναι συμπληρωμένα, με βάση αυτόν τον αριθμό ανάβει ο ανάλογος αριθμός LEDs (Table 2). Για την υλοποίηση της περιοδικής ανανέωσης των LEDs χρησιμοποιήθηκε ο Timer 0 σε λειτουργία output compare θέτωντας το καταχωρητή OCR0 στη τιμή 150 κατά την αρχικοποίηση του Timer 0. Ο prescaler τέθηκε στην τιμή 256 θέτωντας στο καταχωρητή TCCR0 το bit CS02 ίσο με 1. Η τιμή του OCR0 και του prescaler ενδέχεται να αλλάζουν καθώς δεν έχει ελεγχθεί πάνω στη πλακέτα

ο ρυθμός ανανέωσης των LEDs και η επίδραση στην απόδοση της επίλυσης του Sudoku.

LEDs	Clues found
LEDs off	< 10
LED00	≥ 10
LED00-01	≥ 20
LED00-02	≥ 30
LED00-03	≥ 40
LED00-04	≥ 50
LED00-05	≥ 60
LED00-06	≥ 70
LED00-07	≥ 80

Table 2: Progress Bar

4 Έλεγχος σωστής λειτουργίας

Η σωστή λειτουργία του συστήματος ελέγχθηκε για κάθε υποσύστημα ξεχωριστά.

Η επιβεβαίωση της ορθής λειτουργίας τόσο της σειριακής θύρας όσο και της εκτέλεσης των εντολών ΔΕΝ κατέστη δυνατή. Αυτό συνέβη καθώς η αναπτυξιακή πλακέτα STK500 η οποία μας δόθηκε είχε πρόβλημα στη μετάδοση δεδομένων πίσω στο PC.

Η διαπίστωση του προβλήματος της πλάκετας έγινε δυνατή με τη χρήση πλάκετας άλλης ομάδας. Συγκεκριμένα ανατύχθηκε ένα απλό πρόγραμμα το οποίο επιτρέπει στον αποστολέα τους χαρακτήρες που του έστειλε. Στη δική μας πλακέτα αυτό που επέστρεφε το πρόγραμμα ήταν σκουπίδια ενώ στη πλακέτα της άλλης ομάδας επιστρέφονταν οι χαρακτήρες που είχαν μεταδοθεί. Επίσης διαπιστώθηκε ότι το καλώδιο δεν ήταν αυτός ή αιτία του προβλήματος χρησιμοποιώντας το καλώδιο της άλλης ομάδας.

Η απεικόνιση της προόδου λειτουργήσε κατόπιν πολλαπλών ελέγχων πάνω στη πλακέτα καθώς τόσο τα LEDs όσο και το PORT A λειτουργούσαν ορθά.