



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ (Α.Π.Θ.)

ΗΥ0901 ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ

Εργασία Εργαστηρίου: Άσκησή Με Σιλό Αποθήκευσης

Αντωνιάδης Δημήτριος (8462): akdimitri@auth.gr
Δημητριάδης Βασίλειος (8404): dimvasdim@auth.gr

23 Μαΐου 2019

Περιεχόμενα

1	Εισαγωγή.	2
2	Λειτουργία προγράμματος.	2
3	Αποτελέσματα.	6
4	Επίλογος.	6
5	Παράρτημα.	7

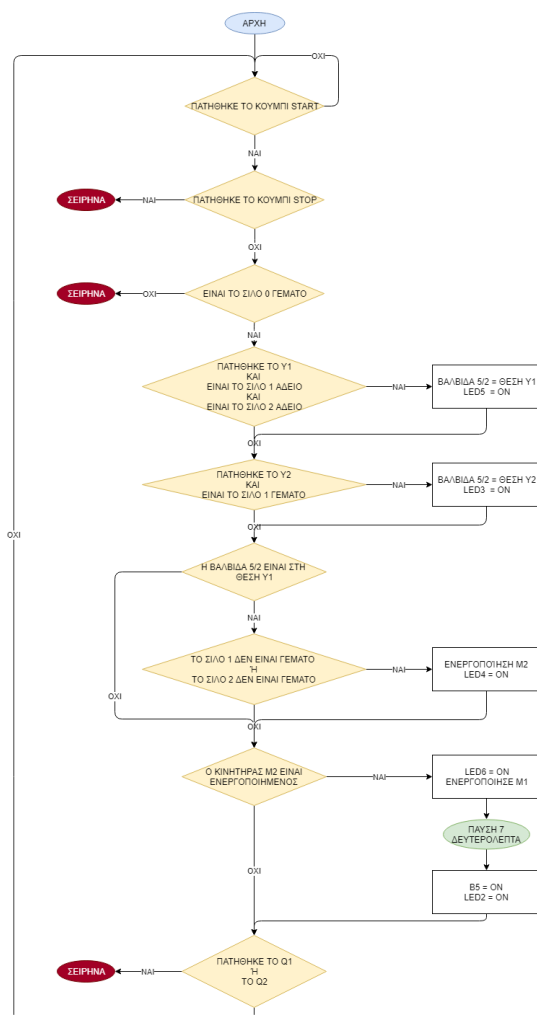
1 Εισαγωγή.

Το παρόν έγγραφο αποτελεί την αναφορά της εργασίας εργαστηρίου που πραγματοποιήθηκε στο πλαίσιο του μαθήματος *Μικροεπεξεργαστές και Περιφερειακά*. Σκοπός της εργασίας ήταν ο προγραμματισμός του μικροελεγκτή **ATmega16** προκειμένου αυτός να ελέγχει και να ρυθμίζει μία δομή αποθήκευσης, η οποία περιλάμβανε 3 σιλό αποθήκευσης. Πιο συγκεκριμένα, το ζητούμενο ήταν ο αποδοτικός διαμερισμός των αποθηκευμένων υλικών από το Silo 0 στα Silo 1 και Silo 2 μέσω μίας γραμμής μεταφοράς.

Παραπάνω έγινε μία σύντομη αναφορά του σκοπού της εργασίας. Στην επόμενη (2^η) ενότητα παρουσιάζεται η λειτουργία του προγράμματος και στην τρίτη (3^η) ενότητα τα αποτελέσματα λειτουργικότητας του προγράμματος. Τέλος, η τέταρτη (4^η) ενότητα αποτελεί τον επίλογο της εργασίας αυτής.

2 Λειτουργία προγράμματος.

Η λειτουργία του προγράμματος είναι σχετικά απλή και φαίνεται στο παρακάτω διάγραμμα.



Σχήμα 1: Διάγραμμα ροής κυρίως προγράμματος.

Αρχικά, το πρόγραμμα περιμένει να πατηθεί το START button. Αν πατηθεί τότε εκτελείται το πρόγραμμα.

Από τη στιγμή που πατηθεί το START button το πρόγραμμα μπαίνει σε ένα συνεχές loop. Το πρώτο πράγμα που ελέγχεται όταν το πρόγραμμα μπει σε αυτό το συνεχές loop είναι να ελέγξει αν πατήθηκε το START button. Αν το START button πατήθηκε τότε ενεργοποιείται η σειρήνα, αλλιώς συνεχίζει κανονικά η εκτέλεση του προγράμματος. Ο επόμενος έλεγχος αφορά το σιλό αποθήκευσης, αν το σιλό αποθήκευσης είναι άδειο, τότε ενεργοποιείται η σειρήνα, αλλιώς συνεχίζει η εκτέλεση του προγράμματος.

Στη συνέχεια ελέγχεται αν πατήθηκε το κουμπί προσομοίωσης για τη θέση Y1. Τότε αν το σιλό 1 είναι άδειο και το σιλό 2 είναι άδειο, η βαλβίδα 5/2 τοποθετείται στη θέση Y1 και ανάβει το LED5.

Αν πατήθηκε το κουμπί προσομοίωσης Y2 και το σιλό 1 είναι γεμάτο, τότε μετακινείται η βαλβίδα στη θέση Y2 και το LED3 ανάβει.

Τώρα, αν η βαλβίδα 5/2 βρίσκεται στη θέση Y1 και το σιλό 1 ή το σιλό 2 δεν είναι γεμάτο ενεργοποιείται ο κινητήρας M2 και ανάβει το LED4.

Στη συνέχεια, ελέγχεται αν ο κινητήρας M2 είναι ενεργοποιημένος. Τότε ανάβει το LED6 και ενεργοποιείται ο M1. Το πρόγραμμα περιμένει 7 δευτερόλεπτα για να πιάσει ο κινη-

τήρας την κατάλληλη ταχύτητα και ανάβει το LED2.

Τέλος, αν έχει πατηθεί κάποιο από τα κουμπιά που προσομοιώνουν τα θερμικά Q1, Q2 ηχεί η σειράνα.

Καθώς εκτελείται το παραπάνω πρόγραμμα, ο μικροελεγκτής ανά τακτά χρονικά διαστήματα, διακόπτει τη ροή του προγράμματος προκειμένου να διαβάζει τις στάθμες των τριών σιλό.

Η υλοποίηση του προγράμματος βασίστηκε στο Datasheet του ATmega16. [1]

Για την εκτέλεση του προγράμματος χρησιμοποιούνται 2 καταχωρητές οι οποίοι ονομάζονται στο πρόγραμμα

- COntrol REgister A (COREA)
- COntrol REgister B (COREB)

Τα bits των καταχωρητών αυτών μεταχειρίζονται ως εξής:
COREA:

B1	B2	B3	B4	A1	Y1	Y2	-
7	6	5	4	3	2	1	0

COREB:

M1	M2	-	-	-	B5	-	-
7	6	5	4	3	2	1	0

Σημειώνεται ότι για τα bits B1, B2, B3, B4, A1, το 1 σημαίνει ότι έχει το σιλό έχει φτάσει τη στάθμη αυτή και το 0 ότι ακόμη δεν έχει φτάσει τη στάθμη αυτή το αντίστοιχο σιλό.

Επιπλέον, κατά την εκτέλεση του προγράμματος χρησιμοποιούνται δύο Interrupt Vectors. Αυτά τα δύο είναι τα εξής:

- \$0000 RESET
- \$0012 TIM0_OVF

Στο RESET αρχικοποιείται ο ADC, τα INPUT-OUTPUT PINS, και ο TIMER0 OVERFLOW VECTOR.

Για την αρχικοποίηση του ADC τίθεται ο ADCSRA:

1	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

Το bit 7: ADEN ενεργοποιεί τον ADC. Ακόμη, ενεργοποιώντας τα bits: 0,1,2, ο prescaler τίθεται σε $\text{clk}/128$, ώστε να δουλεύει σε συχνότητα 125 kHz. Αυτό γίνεται διότι στο datasheet αναφέρεται ότι ο ADC δουλεύει καλύτερα σε συχνότητες 50-200 kHz. [1]

Για την αρχικοποίηση του TIMER0 OVERFLOW το bit 0 του TIMSK τίθεται σε 1 και ο καταχωρητής TCCR0 τίθεται στην τιμή:

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Δηλαδή ο prescaler του τίθεται σε συχνότητα $\text{clk}/1 = 16 \text{ MHz}$. Άρα ο TIM0_OVF εκτελείται κάθε 16 μικροδευτερόλεπτα.

Η διεύθυνση του διανύσματος RESET τοποθετείται στη θέση \$002A όπως προτείνεται στο datasheet. [1]

```

.ORG 0x02A
RESET:
    LDI R16, HIGH(RAMEND)      ; Main program start
    OUT SPH, R16               ; Set stack pointer to top of RAM
5    LDI R16, LOW(RAMEND)
    OUT SPL, R16

    LDI COREA, 0b00000000      ; Initialize CONTROL REGISTER A
    LDI COREB, 0b00000000      ; Initialize CONTROL REGISTER B
10    LDI R16, 0b11111111      ; Define PORTB -> OUTPUT
    OUT DDRB, R16              ;
    OUT PORTB, R16             ; TURN OFF LEDS

    LDI R16, 0b00000000        ; Define PORTD -> INPUT
    OUT DDRD, R16              ;
15    LDI R16, 0b11111111      ; Activate PULL UP Resistors
    OUT PORTD, R16             ;

    LDI R16, 0b10000111        ; ENABLE ADC CONVERTER, SET PRESCALING TO CLK
    /128 -> 125 kHz
    OUT ADCSRA, R16

    LDI R16, 0b00011111        ; ENABLE PULL UP RESISTORS OF PA0-PA4
    OUT PORTA, R16
25    LDI R16, 0xFA             ; Set COUNTER0 to 0xFA to FORCE interrupt
    OUT TCNT0, R16
    LDI R16, 0b00000001        ; Enable TIM0_OVF Interrupt
    IN R17, TIMSK
30    OR R17, R16
    OUT TIMSK, R17
    LDI R16, 0b00000001        ; Start COUNTER0, Normal Mode, Prescaler =
    clk/1
    OUT TCCR0, R16
35    SEI

```

Η λειτουργία του TIMER0 φαίνεται σε μορφή ψευδογλώσσας στον παρακάτω αλγόριθμο.

Algorithm 1: TIMER0 OVERFLOW VECTOR

Data: Potentiometers Values

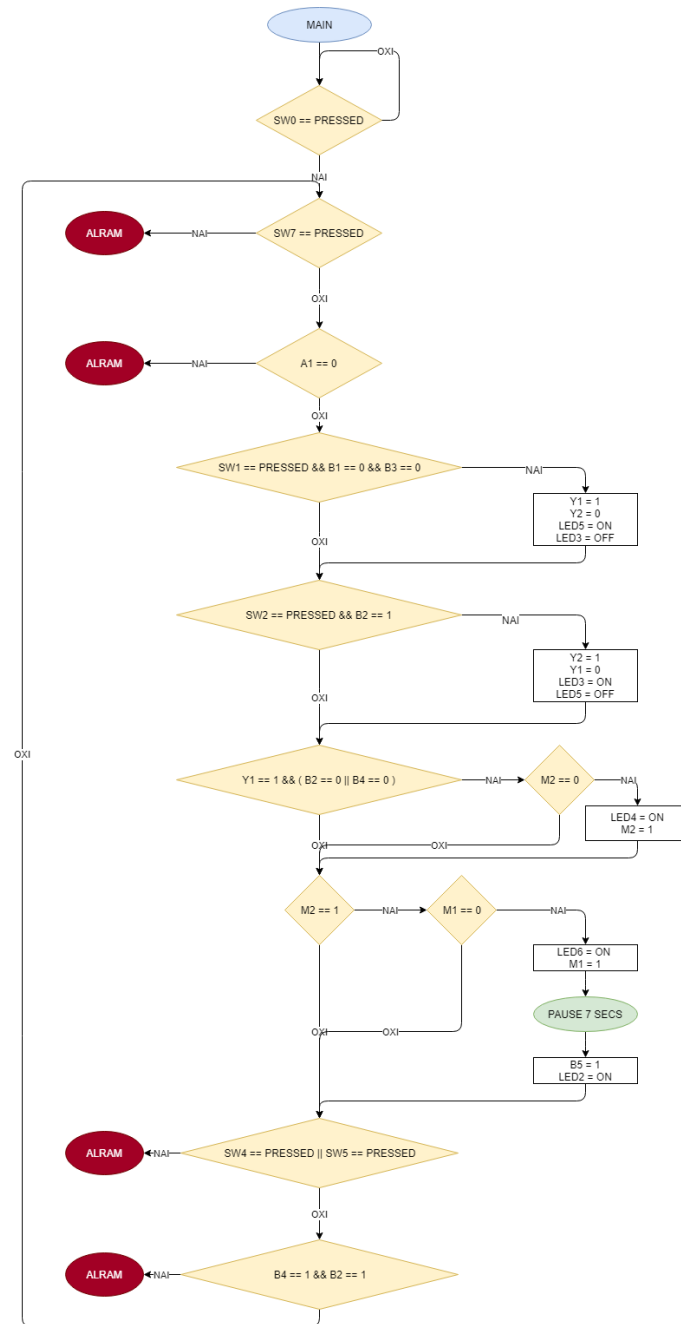
Result: Define B1,B2,B3,B4,A1 Values

- 1 Save SREG
 - 2 Read PA0-4 using ADC
 - 3 Set/Clear COREA: 7-3 /* B1, B2, B3, B4, A1 */
 - 4 Restore SREG
-

Η καθυστέρησή 7 δευτερολέπτων πραγματοποιείται με τη συνάρτηση PAUSE, η οποία με 4 καταχωρητές εκτελεί 112000000 κύκλους ρολογιού στα 16 MHz.

Η συνάρτηση ALARM ενεργοποιεί το buzzer χρησιμοποιώντας έναν παλμό 8 kHz με duty cycle = 50%. Παράλληλα, περιμένει το πάτημα του κουμπιού ACKNOWLEDGEMENT και εκκινεί ξανά το πρόγραμμα από το VECTOR RESET. Κατά τη διάρκεια εκτέλεσης της συνάρτησης ALARM απενεργοποιούνται τα Interrupts.

Τέλος, η συνάρτηση MAIN φαίνεται στο παρακάτω διάγραμμα πως πραγματοποιήθηκε:



Σχήμα 2: MAIN

Ο αλγόριθμος της συνάρτησης MAIN περιλαμβάνεται ολόκληρος σε επίπεδο καταχωρητών στο παράρτημα σε μορφή διαγράμματος.

3 Αποτελέσματα.

Το πρόγραμμα αρχικά προσομοιώθηκε στο AVR Studio. Η συνάρτηση TIM0_OVF τροποποιήθηκε ώστε να εκτελεί μία NOP. Κάθε φορά που καλούνται η συνάρτηση αυτή, γίνονταν τα αντίστοιχα edits στους καταχωρητές COREA, COREB. Το πρόγραμμα απεδείχθη πλήρως λειτουργικό.

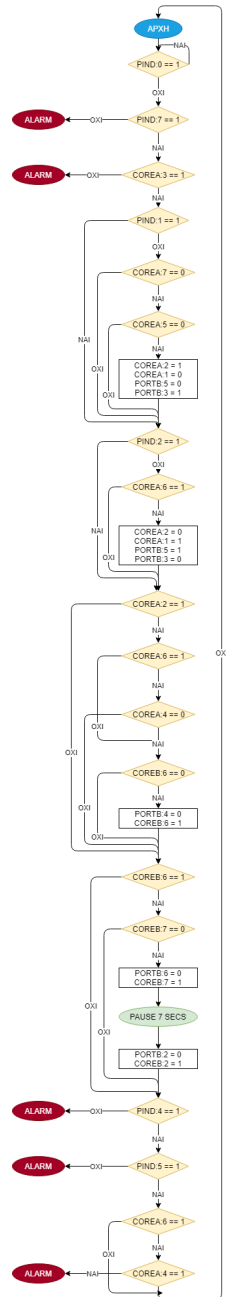
4 Επίλογος.

Συνοψίζοντας, στο έγγραφο αυτό παρουσιάστηκε η εργασία που πραγματοποιήθηκε στο πλαίσιο του εργαστηρίου του μαθήματος *Μικροεπεξεργαστές και Περιφερειακά*. Στην αναφορά αυτή, επεξηγήθηκε η λύση της εργασίας και οι παράγοντες που οδήγησαν στη λύση αυτή. Τέλος, το πρόγραμμα που υλοποιήθηκε δοκιμάστηκε στη δοκιμαστική πλακέτα του εργαστηρίου και απεδείχθη πλήρως λειτουργικό.

Αναφορές

[1] ATmega16 Datasheet. <http://ww1.microchip.com/downloads/en/devicedoc/doc2466.pdf>.

5 Παράρτημα.



Σχήμα 3: MAIN (silos.asm)