Άσκηση με Σιλό

Group D

Στιβακτάκης Νικόλαος
ΑΕΜ:8351
email: nikostiv@ece.auth.gr

Μπεκιάρης Θεοφάνης
ΑΕΜ:8200
email: theompek@ece.auth.gr

Στόχος της συγκεκριμένης εργαστηριακής άσκησης ήταν ο έλεγχος της διαδικασίας εγκατάστασης πλήρωσης υλικού.Για την προσομοίωση του προγράμματος χρησιμοποιήθηκε το AVR studio της atmel ενώ ο πρόγραμμα δοκιμάστηκε και στην αναπτυξιακή κάρτα STK500.

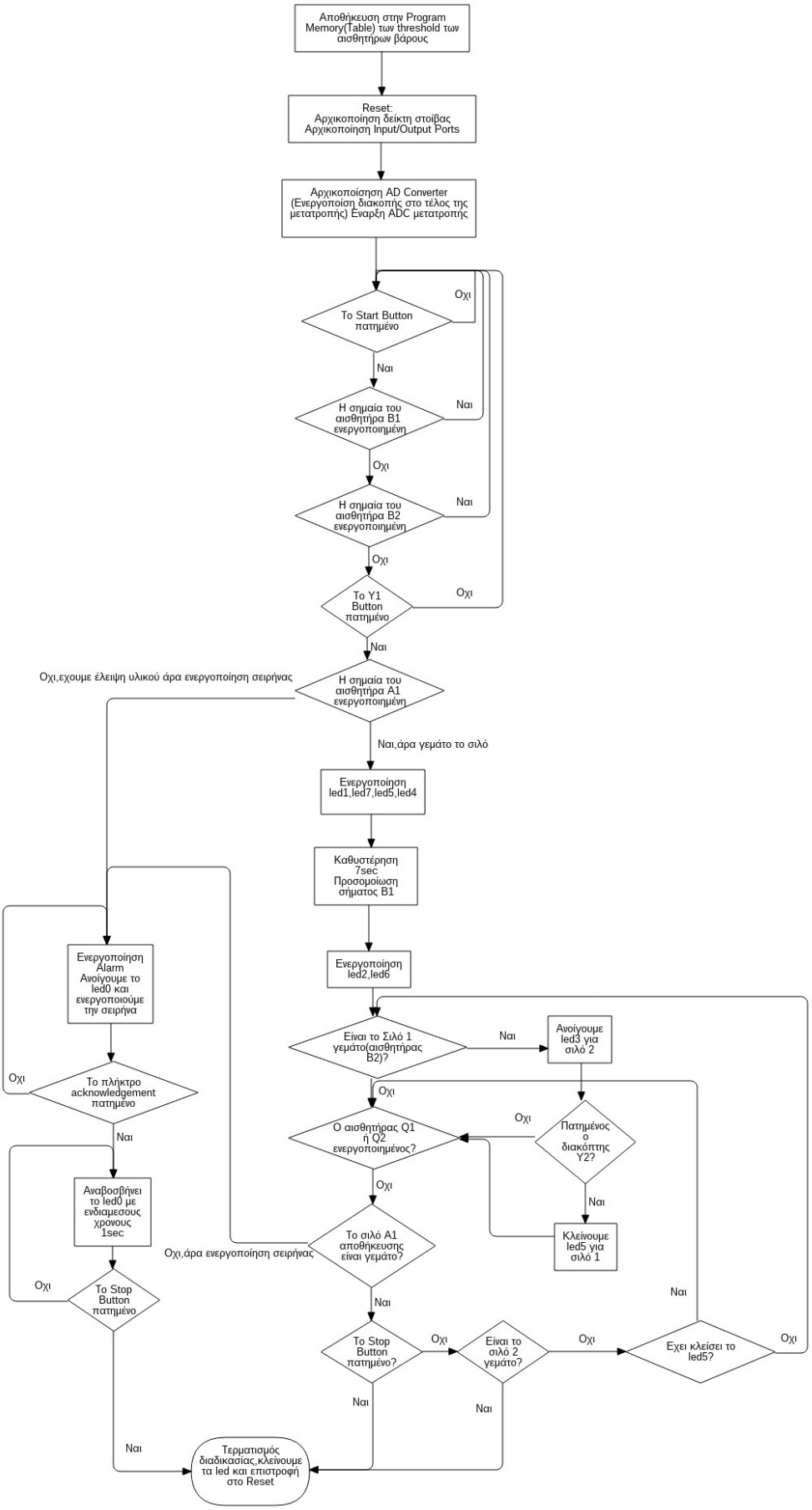
Συνοπτικά η διαδικασία που καλούμαστε να ελέγξουμε αφορά την μεταφορά υλικού από ένα σιλό μεγάλων διαστάσεων σε δύο σιλό μικρότερης χωρητικότητας. Η διαδικασία αυτή,όσο βρίσκεται σε εξέλιξη, υλοποιεί διαφορετικές ενέργειες όπως ο έλεγχος κάποιων αισθητήρων οι οποίοι προσδιορίζουν το βάρος των σιλό, ή και ενέργειες αναγνώριση σημάτων(π.χ. button εκκίνησης, μετακίνηση αγωγού εκφόρτωσης Y1,Y2).

Μέσω του Analog/Digital converter προσομοιώνουμε τους αισθητήρες βάρος, με ρύθμιση του ποτενσιόμετρου στις τιμές που εμείς ορίσαμε σαν “χαμηλή” και “υψηλή”. Μια τιμή θεωρείται “χαμηλή” αν είναι μικρότερη των 2.5V και “υψηλή” αν είναι μεγαλύτερη των 2.5V (lowByte, highByte). Κάθε φορά που έχουμε ένας αισθητήρας ξεπερνάει τα όρια που έχουμε θέσει στον πίνακα Table ενεργοποιείται η σημαία αισθητήρα sensors και κατόπιν ελέγχεται για τον την αλλαγή της ροής του προγράμματος.

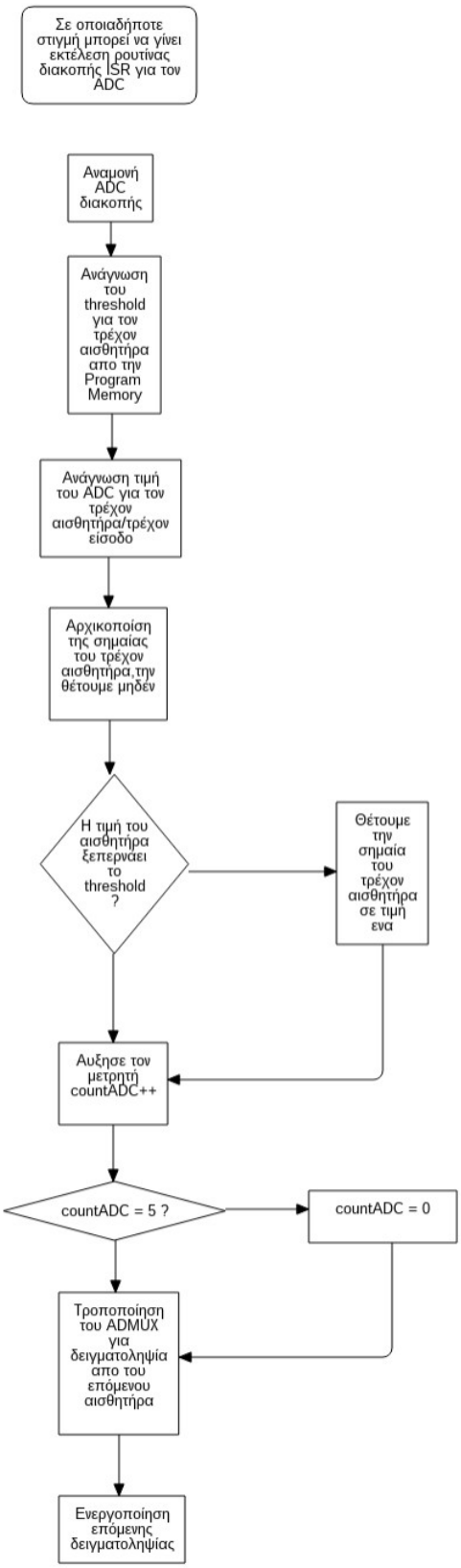
Ενεργοποιούμε τον ADC να διαβάζει σειριακά τους αισθητήρες και στο τέλος κάθε δειγματοληψίας εκτελείτε διακοπή.Μέσα στην ρουτίνα διακοπής ελέγχεται αν η τιμή ξεπερνάει το threshold και αν το ξεπερνάει τίθεται η αντίστοιχη σημαία σε λογικό 1 διαφορετικά μένει στην τιμή μηδέν.Επιπλέον στο τέλος στην ρουτίνας ο ADC αρχικοποιείται και καλείται να εκτελέσει δειγματοληψία για τον επόμενο αισθητήρα.

Θεωρήθηκε ότι η χρήση διαγράμματος ροής σε συνδυασμό με τα σχόλια δίπλα από σημαντικές εντολές προς επεξήγηση, είναι επαρκής για την οπτικοποίηση της διαδικασίας, και της κατανόησης της κεντρικής ιδέας για τον έλεγχο της διαδικασίας πλήρωσης υλικού.

Διάγραμμα ροής:



Εκτέλεση ρουτίνας διακοπής ISR για τον ADC:



Πρόγραμμα:

```
;8ewroume clk_cpu = 4MHz

.include "m16def.inc"

.def temp = r16
.def countADC = r17
.def lowByte = r18
.def highByte = r19
.def adcLow = r20
.def adcHigh = r21
.def sensors = r23 ;Kataxorhths shmaiwn twn shmatwn
;Shmaies shmatwn barous kai sta8mhs
.equ A1f = 0
.equ B1f = 1
.equ B3f = 2
.equ B2f = 3
.equ B4f = 4
;8eshs twn bits diakoptwn/shmatwn sto portD eisodou
.equ startBtn = 0
.equ Y1 = 1
.equ Y2 = 2
.equ Q1 = 4
.equ Q2 = 5
.equ Acknlg = 6
.equ StopBtn = 7
;8eshs twn bits shmatwn/led sto portB e3odou
.equ H1 = 0
.equ A1 = 1
.equ B5 = 2
.equ Silo2 = 3
.equ M2 = 4
.equ Silo1 = 5
.equ M1 = 6
.equ Run = 7
.equ input = 0x10  ;input = PIND      = 0x10
.equ output = 0x18 ;output = PORTB          = 0x18
.equ timer1sec = 61630
.equ Hsetp1 = HIGH(timer1sec)
.equ Lsetp1 = LOW(timer1sec)
.equ timer7sec = 38192
```

```
.equ Hsetp7 = HIGH(timer7sec)
.equ Lsetp7 = LOW(timer7sec)

.cseg

.org 0x00
jmp reset

.org 0x1C
jmp ADC_ISR

.org 0x100
Table:
.DW 0x0200,0x0201,0x0202,0x0203,0x0204

reset:
ldi temp,low(RAMEND) ; Arxikopoihsh stack pointer
out SPL,temp
ldi temp,high(RAMEND)
out SPH,temp
;Arxikopoihsh Ports eisodoy/e3odou
;--------PortC-----------
sbi DDRC,0 ;Orizoume to bit 0 tou portC e3odo gia thn seirhna
cbi PORTC,0
;--------PortB-----------
ser temp ;Orizoume to portB ws e3odo gia ta led
out DDRB,temp
out output,temp ;Ta led kleista
;--------RortD-----------
clr temp;
out DDRD,temp ;Orizoume to portD ws eisodo gia switches
ser temp
out PORTD,temp ;Energwpoihsh antistasewn prosdeshs
clr sensors ;Arxikopoihsh timwn shmaiwn gia thn katastash twn es8.barous

;Arxikopoihsh ADC
clr countADC
ldi temp,0b01000000 ;REF0:1 = 01 -> AVcc, ADLAR=0
out ADMUX,temp      ;Arxizoume me MUX4:0=00000,anagnwsh tou ADC0(PA0),ais8hthras
barous A1
ldi temp,0b11001101 ;ADEN=1,ADCS=1,ADPS2:0 = 101 -> diairesh me 32 -> clk_cpu=4MHz
tote 9.600 deigmata/sec
out ADCSRA,temp     ;Energopoish ADC
sei                         ;Energopoihsh interrupts
```

```
startCond:
sbic input,startBtn ;An path8ei to plhktro start 1->0 3ekinaei h diadikasia
rjmp startCond
sbrc sensors,B1f    ;Prepei to silo 1 na einai adeio(ais8hthras barous b1)
rjmp startCond
sbrc sensors,B2f    ;Prepei to silo 2 na einai adeio(ais8hthras barous b1)
rjmp startCond
sbic input,Y1       ;An einai pathmeno to plhktro Y1 gia ton silo1
rjmp startCond
sbrs sensors,A1f    ;An einai gemato to Silo Apo8ukeushs A1f = 0 sunexhse
jmp alarm           ;diaforetika energopoieitai to alarm

;H diadikasia 3ekinhse
cbi output,A1    ;anoigoume led1
cbi output,Run   ;Anoigoume Led7
cbi output,Silo1 ;Anoigoume Led5
cbi output,M2    ;Anoigoume Led4,kinhsh metaforikhs tainias

rcall delayTimer7sec ;Perimenoyme to shma B5,prosomoiwsh me timer

cbi output,B5  ;Anoigoume Led2,energopihsh shmatos B5
cbi output,M1 ;Anoigoume Led6,ekenwsh ulikou mesw valvidas

loop:
sbrs sensors,B2f  ;Otan gemisei to Silo1 B2f(0->1)
rjmp subloop
cbi output,Silo2 ;anoigoume led3 gia to silo2
sbic input,Y2    ;An path8ei o diakopths Y2(1->0)
sbi output,Silo1 ;kleinoyme led5 gia silo1

subloop:
sbis input,Q1
rjmp alarm
sbis input,Q2
rjmp alarm
sbrs sensors,A1f
rjmp alarm
sbrc sensors,B4f ;an to Silo2(B4f->1) einai gemato tote
rjmp  stop       ;telos diadikasias
sbis input,StopBtn
rjmp stop
sbis output,Silo1 ;Otan exei gemisei to silo 1 phgaine kateu8eian sthn subloop
rjmp loop
```

```
rjmp subloop


alarm:
ser temp
out output,temp
cbi output,H1 ;Anoigoume led0 gia thn endei3ei sfalmatos
sbi PORTC,0   ;Anoigoume thn seirhna
wait:
sbic input,Acknlg
rjmp wait
cbi PORTC,0   ;Kleinoume thn seirhna
blink:
sbi output,H1
rcall delayTimer1sec
cbi output,H1
rcall delayTimer1sec
sbic input,StopBtn  ;Anavosbhnei to led seirhnas mexri na path8ei to plhktro Stop
rjmp blink

stop:
ser temp          ;Ekκa8arish led
out output,temp
rjmp startCond


ADC_ISR:
push temp
ldi ZH,HIGH(Table*2)
ldi ZL,LOW(Table*2)
lsl countADC    ;Pollaplasiasmos me 2
ldi temp,0x00
add ZL,countADC ;Pernoume apo thn mnhmh programmatos to orio
adc ZH,temp     ;to opoio den prepei na 3epernaei o ka8e ais8hthras
lpm lowByte,z+
lpm highByte,z
in adcLow,ADCL
in adcHigh,ADCH
lsr countADC    ;Epanafora tou countADC



;Mhdenizoume arxika to flag tou antistoixou
;ais8hthra pou e3etazoume
```

```
cpi countADC,0
in temp,SREG
sbrc temp,1       ;An countADC=0 tote SREG(Z)=1 kai ara
cbr sensors,1<<A1f ;arxikopoioume sthn timh mhden to A1f
cpi countADC,1    ;diaforetika oi shmaies menoun ws exoun
in temp,SREG   ;To idio isxuei gia ka8e periptwsh pou akolou8ei
sbrc temp,1
cbr sensors,1<<B1f
cpi countADC,2
in temp,SREG
sbrc temp,1
cbr sensors,1<<B2f
cpi countADC,3
in temp,SREG
sbrc temp,1
cbr sensors,1<<B3f
cpi countADC,4
in temp,SREG
sbrc temp,1
cbr sensors,1<<B4f

;Elegxoume an o ais8hthras 3epernaei thn epitrepth timh
cp lowByte,adcLow
cpc highByte,adcHigh
brge setNextADC

setFlag:
;8etoume thn antistoixh shmaia pou dhlwnei
;oti o ais8hthras energopoieitai
cpi countADC,0
in temp,SREG
sbrc temp,1       ;An countADC=0 tote SREG(Z)=1 kai ara
sbr sensors,1<<A1f ;topo8eteitai sthn timh ena(1) to A1f
cpi countADC,1    ;diaforetika oi shmaies menoun ws exoun
in temp,SREG   ;To idio isxuei gia ka8e periptwsh pou akolou8ei
sbrc temp,1
sbr sensors,1<<B1f
cpi countADC,2
in temp,SREG
sbrc temp,1
sbr sensors,1<<B2f
cpi countADC,3
in temp,SREG
sbrc temp,1
```

```
sbr sensors,1<<B3f
cpi countADC,4
in temp,SREG
sbrc temp,1
sbr sensors,1<<B4f

setNextADC:
inc countADC   ;Au3hse gia thn metatroph ths epomenhs eisodou ADC
cpi countADC,5
in temp,SREG
sbrc temp,1 ;An ftaseis sthn teleutaia eisodo(ADC4) 3ana mhdenise
clr countADC
in temp,ADMUX
andi temp,0b11100000 ;Mhdenizoume ta bits MUX4:0
or temp,countADC     ;Fortonoume sta bits MUX4:0 thn 8esh
out ADMUX,temp       ;ths epomenhs eisodou ADC, 0<=countADC<=4 ->
00000000<=countADC<=00000100
sbi ADCSRA,6         ;Energopoioume thn epomenh metatroph ADC 8etontas ADSC=1
pop temp
reti


;Orizoume routina pou energopoiei ton timer1 gia ka8usterish 1 sec
;Exoume clk_cpu = 4MHz ara gia prescaler = 1024 8eloume 3906 metrhseis
;ara arxikopoioume ton timer1 se timh 65536-3906 = 61630
delayTimer1sec:
push temp
ldi temp,1<<TOV1
out TIFR,temp       ;Arxikopoihsh shmaias TOV1
ldi temp,Hsetp1     ;Apo8hkeush arxikhs timhs ston timer1
out TCNT1H,temp
ldi temp,Lsetp1
out TCNT1L,temp
ldi temp,0b00000101 ;Prescales = 1024
out TCCR1B,temp
waitTimer1:
sbis input,StopBtn ;An path8ei to Stop Button stamataei h leitourgia
rjmp stop
in temp,TIFR
sbrs temp,TOV1
rjmp waitTimer1
pop temp
ret
```

```
;Routina gia ka8usterish 7sec
;Exoume clk_cpu = 4MHz ara 7sec kai gia prescaler = 1024 8eloume  27344 metrhseis
;ara arxikopoioume ton timer1 se timh 65536-27344 = 38192
delayTimer7sec:
push temp
ldi temp,1<<TOV1
out TIFR,temp       ;Arxikopoihsh shmaias TOV1
ldi temp,Hsetp7     ;Apo8hkeush arxikhs timhs ston timer1
out TCNT1H,temp
ldi temp,Lsetp7
out TCNT1L,temp
ldi temp,0b00000101 ;Prescales = 1024
out TCCR1B,temp
waitTimer7:
sbis input,StopBtn ;An path8ei to Stop Button stamataei h leitourgia
rjmp stop
in temp,TIFR
sbrs temp,TOV1
rjmp waitTimer7
pop temp
ret
```