



**4<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**  
**ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Εργαστήριο Μικροϋπολογιστών"**  
**Χρήση Οθόνης 2×16 Χαρακτήρων στον AVR**

**Εξέταση – Επίδειξη: Παρασκευή 7/11/2025**  
**Προθεσμία για παράδοση Έκθεσης: Τρίτη 11/11/2025 (23:59)**

**Αλφαριθμητική Οθόνη 2×16 Χαρακτήρων**

Η οθόνη είναι υγρού κρυστάλλου (LCD) με χαμηλή κατανάλωση και έχει ενσωματωμένο ένα μικροελεγκτή. Ο ενσωματωμένος μικροελεγκτής εμπεριέχει δύο μνήμες για τη δημιουργία χαρακτήρων: την CG-ROM (Character Generation ROM) και την CG-RAM (Character Generation RAM). Επίσης έχει μία μνήμη RAM στην οποία κρατούνται προσωρινά οι χαρακτήρες που απεικονίζονται κάθε φορά στην οθόνη την DD\_RAM (Display Data RAM).

Ακροδέκτες της οθόνης

Στη συνέχεια παρατίθενται οι ακροδέκτες της οθόνης και οι αντίστοιχες λειτουργίες τους:

<b>DB[0:7]</b>	Τρικατάστατος αμφίδρομος διάδρομος δεδομένων 8-γραμμών, με τον οποίο η οθόνη ανταλλάσσει δεδομένα με τον εξωτερικό μικροεπεξεργαστή. Υπάρχει δυνατότητα χρήσης μόνο 4 γραμμών, οπότε η ανταλλαγή των δεδομένων γίνεται σε δύο φάσεις, δια μέσω των ακροδεκτών DB[7:4] ενώ οι ακροδέκτες DB[0:3] δεν χρησιμοποιούνται. Τα 4 περισσότερα σημαντικά bit πρέπει να μεταφέρονται πρώτα.
<b>E'</b>	Σήμα που ενεργοποιεί την οθόνη. Η ενεργοποίηση γίνεται Αν $E'=0$ .
<b>R/W'</b>	Σήμα που καθορίζει αν πρόκειται να γίνει εγγραφή ή διάβασμα από την οθόνη ( $R/W'=0$ εγγραφή, $R/W'=1$ διάβασμα).
<b>RS</b>	Το σήμα RS καθορίζει αν το περιεχόμενο των DB[0:7] είναι διεύθυνση/εντολή ή αν είναι δεδομένα που πρόκειται να διαβαστούν ή να εγγραφούν ( $RS=0$ διεύθυνση/εντολή, $RS=1$ δεδομένα).
<b>V0</b>	Ρυθμίζει τη φωτεινότητα της οθόνης.
<b>VDD</b>	Συνδέεται με την τάση τροφοδοσίας (3,3 volt).
<b>VSS</b>	Συνδέεται με τη γη (0 volt).

Καταχωρητές

Ο εσωτερικός μικροελεγκτής της LCD οθόνης ενσωματώνει δύο καταχωρητές των 8-bit, τον καταχωρητή εντολών (instruction register -IR) και τον καταχωρητή δεδομένων (data register- DR).

Ο καταχωρητής IR αποθηκεύει τον κώδικα των εντολών, ή τη διεύθυνση μιας θέσης της μνήμης DD-RAM ή της μνήμης CG-RAM. Στον καταχωρητή IR γίνεται μόνο εγγραφή και όχι ανάγνωση από τον εξωτερικό μικροελεγκτή. Ο καταχωρητής DR αποθηκεύει προσωρινά τα δεδομένα που διαβάζονται ή γράφονται στην DD-RAM ή στην CG-RAM.

Σημαία απασχόλησης (Busy Flag-BF)

Όταν η σημαία απασχόλησης είναι 1, η οθόνη είναι σε κατάσταση εσωτερικής λειτουργίας και δεν γίνεται δεκτή καμία άλλη εντολή. Όταν  $RS=0$  και  $R/W'=1$ , η σημαία απασχόλησης εμφανίζεται στην έξοδο DB7. Η επόμενη εντολή πρέπει να δοθεί, μετά την εξασφάλιση ότι η σημαία απασχόλησης είναι 0.

#### Μετρητής Διεύθυνση (Address Counter -AC)

Ο μετρητής διεύθυνση (AC) παρέχει διευθύνσεις σε αμφότερες τις μνήμες DD-RAM και CG-RAM. Όταν μια διεύθυνση μιας εντολής εγγράφεται στον IR, η διεύθυνση στέλνεται από τον IR στον AC.

Μετά την εγγραφή ή την ανάγνωση από την DD-RAM ή την CG-RAM, η AC αυτόματα αυξάνεται κατά 1 (ή ελαττώνεται κατά 1). Το περιεχόμενο του καταχωρητή AC δίνεται στη έξοδο μέσω των DB[0:6] όταν RS = 0 και R/W' = 1.

RS	R/W'	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DD-RAM or CG-RAM)
1	1	DR read as an internal operation (DD-RAM or CG-RAM to DR)

**Πίνακας 4.1:** Επιλογή Καταχωρητή

#### Εσωτερική μνήμη DD-RAM

Η DD-RAM αποθηκεύει τα δεδομένα που πρόκειται να απεικονιστούν στην οθόνη. Η χωρητικότητά της είναι 80 byte. Ο κώδικας της διεύθυνσης μιας θέσης μνήμης DD-RAM είναι των επτά Bit. Οι χαρακτήρες που οι κώδικες τους βρίσκονται στις θέσεις 0x00 έως 0x0F της DD-RAM τοποθετούνται(απεικονίζονται) στη γραμμή 1 της οθόνης, ενώ οι κώδικες που βρίσκονται στις θέσεις 0x40 έως 0x4F απεικονίζονται στη γραμμή 2 όπως φαίνονται στον παρακάτω πίνακα:

Γραμμή_1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Γραμμή_2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

**Πίνακας 4.2:** Απεικόνιση χαρακτήρων

Αν πρέπει να απεικονιστούν και άλλοι χαρακτήρες που οι κώδικες τους είναι αποθηκευμένοι στη DD-RAM, εκτός από τους 32 χαρακτήρες που αναφέρθηκαν πιο πάνω, τότε πρέπει να γίνει ολίσθηση στο περιεχόμενο της οθόνης δεξιά ή αριστερά.

#### Εσωτερική μνήμη CG-ROM

Η CG-ROM χρησιμοποιείται για τη δημιουργία εικόνων χαρακτήρων 5×8 κουκίδων από κωδικούς χαρακτήρων των 8 bit. Η αντιστοιχία αριθμών των 8 bit και χαρακτήρων φαίνεται στον ακόλουθο πίνακα.

Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)	±		0	P	'	P	5	6	Δ	'	r	P	P	τ	
LLLH	CG RAM (2)	≡	!	1	A	Q	a	q	Q	a	i	'	J	+	γ	υ
LLHL	CG RAM (3)	7	"	2	B	R	b	r	é	R	é	Δ	'	∞	8	Σ
LLHH	CG RAM (4)	L	#	3	C	S	c	s	8	δ	ç	'	P	7	ε	ψ
LHLL	CG RAM (5)	7	*	4	D	T	t	t	8	δ	t	'	+	7	ζ	ω
LHLH	CG RAM (6)	7	%	5	E	U	e	u	8	δ	e	'	+	7	ζ	ω
LHHL	CG RAM (7)	7	8	6	F	V	v	v	8	δ	v	'	+	7	ζ	ω
LHHH	CG RAM (8)	7	'	7	G	W	w	w	8	δ	w	'	+	7	ζ	ω
HLLL	CG RAM (1)	7	(	8	H	X	x	x	8	δ	x	'	+	7	ζ	ω
HLLH	CG RAM (2)	7	)	9	I	Y	y	y	8	δ	y	'	+	7	ζ	ω
HLHL	CG RAM (3)	×	*	#	J	Z	z	z	8	δ	z	'	+	7	ζ	ω
HLHH	CG RAM (4)	7	+	#	K	C	k	c	8	δ	c	'	+	7	ζ	ω
HHLL	CG RAM (5)	≡	.	<	L	\	l	l	8	δ	l	'	+	7	ζ	ω
HHLH	CG RAM (6)	∞	—	≡	M	]m	]m	]m	8	δ	]m	'	+	7	ζ	ω
HHHL	CG RAM (7)	#	.	>	N	^	n	^	8	δ	^	'	+	7	ζ	ω
HHHH	CG RAM (8)	≡	/	?0	_	o	ω	Δ	ç	ω	'	+	7	ζ	ω	ω

Πίνακας 4.3: Αντιστοιχία αριθμών των 8 bit και χαρακτήρων.

### Εντολές της οθόνης

Επειδή στην κάρτα ntuAboard\_G1 δεν υπάρχει δυνατότητα ανάγνωσης της σημαίας BUSY FLAG για να γνωρίζουμε πότε η οθόνη είναι έτοιμη να δεχτεί νέα εντολή πρέπει να εισάγουμε καθυστέρηση μεταξύ των διαδοχικών εντολών. Ο χρόνος που χρειάζεται για την εκτέλεση κάθε εντολής είναι μεταβλητός και εξαρτάται από την κάθε εντολή. Το σύνολο εντολών της οθόνης περιέχεται στον ακόλουθο πίνακα.

Instruction	Instruction Code										Description	Execution time (fosc=270Khz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "00H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	—	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display.	39 $\mu$ s
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and blinking of cursor (B) on/off control bit.	39 $\mu$ s
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	—	—	Set cursor moving and display shift control bit, and the direction, without changing of DDRAM data.	39 $\mu$ s
Function Set	0	0	0	0	1	DL	N	F	—	—	Set interface data length (DL:8-bit/4-bit), numbers of display line (N:2-line/1-line)and, display font type (F:5×11 dots/5×8 dots)	39 $\mu$ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 $\mu$ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 $\mu$ s
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 $\mu$ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 $\mu$ s
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 $\mu$ s

\* "—" : don't care

**Πίνακας 4.4:** Το σύνολο εντολών της οθόνης

**Παρατήρηση:** Πειραματικά έχει φανεί ότι οι οθόνες που χρησιμοποιούνται στην κάρτα ntuAboard\_G1 δεν μπορούν να ανταποκριθούν στους χρόνους που φαίνονται στον παραπάνω πίνακα. Για το σκοπό αυτό, στα τμήματα του κώδικα που ακολουθούν, έχουν χρησιμοποιηθεί μεγαλύτερες καθυστερήσεις. Αυτό πρέπει να ληφθεί υπόψη και στον κώδικα που θα δημιουργήσετε.

Στη συνέχεια δίνεται η επεξήγηση των παραμέτρων που χρησιμοποιούνται στις παραπάνω εντολές.

Το AC<sub>x</sub>, x=0 έως 6, συμβολίζει ένα bit του address counter.

Το D<sub>x</sub>, x=0 έως 7, συμβολίζει ένα bit δεδομένων.

Το BF συμβολίζει το busy flag.

Στον πίνακα που ακολουθεί παρουσιάζεται η ρύθμιση του τρόπου λειτουργίας της οθόνης στην είσοδο ή έξοδο δεδομένων:

S	I/D	Λειτουργία
0	0	Η διεύθυνση της DD-RAM ή της CG-RAM μειώνεται κατά 1 όταν γράφουν δεδομένα. Ο κέρσορας μετακινείται προς τα αριστερά.
0	1	Η διεύθυνση της DD-RAM ή της CG-RAM αυξάνεται κατά 1 όταν γράφουν δεδομένα. Ο κέρσορας μετακινείται προς τα δεξιά.
1	0	Το περιεχόμενο της οθόνης ολισθαίνει προς τα δεξιά όταν γράφουμε στην DD-RAM (όχι όταν διαβάζουμε). Η θέση του κέρσορα δεν μεταβάλλεται.
1	1	Το περιεχόμενο της οθόνης ολισθαίνει προς τα αριστερά όταν γράφουμε στην DD-RAM (όχι όταν διαβάζουμε). Η θέση του κέρσορα δεν μεταβάλλεται.

**Πίνακας 4.5:** Ρύθμιση του τρόπου λειτουργίας της οθόνης στην είσοδο ή έξοδο δεδομένων

Στον πίνακα που ακολουθεί παρουσιάζεται η ρύθμιση του τρόπου ολίσθησης της οθόνης:

S/C	R/L	Λειτουργία
0	0	Η θέση του κέρσορα μετακινείται προς τα αριστερά. (Ο AC μειώνεται κατά 1).
0	1	Η θέση του κέρσορα μετακινείται προς τα δεξιά. (Ο AC αυξάνεται κατά 1).
1	0	Το περιεχόμενο της οθόνης ολισθαίνει προς τα αριστερά μαζί με τον κέρσορα. (Ο AC δεν μεταβάλλεται).
1	1	Το περιεχόμενο της οθόνης ολισθαίνει προς τα δεξιά μαζί με τον κέρσορα. (Ο AC δεν μεταβάλλεται).

**Πίνακας 4.6:** Ρύθμιση του τρόπου ολίσθησης της οθόνης

Αν D=1 τότε η μονάδα απεικόνισης της οθόνης είναι ανοιχτή. Αν D=0 τότε η μονάδα απεικόνισης της οθόνης είναι κλειστή. Και στις δυο περιπτώσεις τα δεδομένα της DD-RAM μένουν αμετάβλητα.

Αν C=1 τότε ο κέρσορας απεικονίζεται στην οθόνη. Αν C=0 τότε ο κέρσορας δεν απεικονίζεται στην οθόνη.

Αν B=1 τότε ο χαρακτήρας πάνω από τον κέρσορα αναβοσβήνει. Αν B=0 τότε ο χαρακτήρας δεν αναβοσβήνει.

Αν DL=1 τότε το μήκος των δεδομένων είναι 8 bit. Αν DL=0 τότε το μήκος των δεδομένων είναι 4 bit.

Στον παρακάτω πίνακα παρουσιάζονται οι κωδικοί των βασικών εντολών της οθόνης:

A/A	Hex Code	Command to LCD instruction Register
1	1	Clear display screen
2	2	Return home
3	4	Decrement cursor (shift cursor to left)
4	6	Increment cursor (shift cursor to right)
5	5	Shift display right
6	7	Shift display left
7	8	Display off, cursor off
8	0A	Display off, cursor on
9	0C	Display on, cursor off
10	0E	Display on, cursor blinking
11	0F	Display on, cursor blinking
12	10	Shift cursor position to left
13	14	Shift the cursor position to the right
14	18	Shift the entire display to the left
15	1C	Shift the entire display to the right
16	80	Force cursor to the beginning ( 1st line)
17	C0	Force cursor to the beginning ( 2nd line)
18	38	2 lines and 5×7 matrix

**Πίνακας 4.7:** Κωδικοί των βασικών εντολών της οθόνης

Τέλος για να γίνει δεκτή μια εντολή από τον ελεγκτή πρέπει να τηρούνται οι χρόνοι του παρακάτω πίνακα:

Ta=25°C, VDD=5.0V

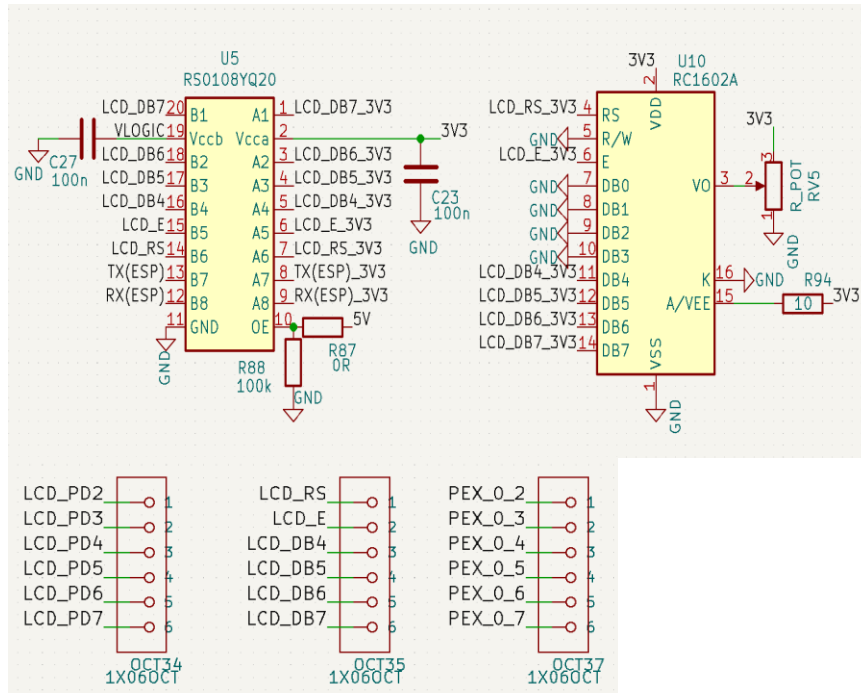
Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	T <sub>C</sub>	1200	—	—	ns
Enable pulse width	T <sub>PW</sub>	140	—	—	ns
Enable rise/fall time	T <sub>R</sub> , T <sub>F</sub>	—	—	25	ns
Address set-up time (RS, R/W to E)	t <sub>AS</sub>	0	—	—	ns
Address hold time	t <sub>AH</sub>	10	—	—	ns
Data set-up time	t <sub>DSW</sub>	40	—	—	ns
Data hold time	t <sub>HI</sub>	10	—	—	ns

**Πίνακας 4.8:** Χρονισμός εντολών οθόνης

## Χρήση της οθόνης LCD 2x16 της κάρτας ntuAboard\_G1.

Η κάρτα ntuAboard\_G1 διαθέτει μία LCD οθόνη 2x16 χαρακτήρων. Μεταξύ της οθόνης και του μικροελεγκτή παρεμβάλλεται το ολοκληρωμένο RS0108YQ20 (8-Bit Bidirectional Voltage Level Translator) για προσαρμογή του επιπέδου της τάσης του μικροελεγκτή (5Volt) με το επίπεδο της τάσης της οθόνης (3.3 Volt).

Η συνδεσμολογία της οθόνης LCD με τον Μικροελεγκτή ATmega328PB φαίνεται στο παρακάτω σχήμα:



**Σχήμα 4.1:** Συνδεσμολογία της οθόνης LCD

Κάνοντας χρήση των κονεκτόρων OCT34, OCT35 και OCT37 μπορούμε να επιλέξουμε εάν οι ακροδέκτες ελέγχου της LCD οθόνης (OCT35) θα συνδεθούν στο PORTD του ATmega328PB (OCT34) ή στο PORT EXPANDER (OCT37). Ανάλογα με τη σύνδεση, οι ακροδέκτες [LCD\_PD2 : LCD\_PD7] ή οι ακροδέκτες [PEX\_0\_7 : PEX\_0\_2] πρέπει να είναι ρυθμισμένοι για έξοδο. Η επικοινωνία μεταξύ οθόνης και του μικροελεγκτή γίνεται με λέξεις των 4 bit.

### Κώδικας επικοινωνίας μεταξύ μικροελεγκτή και οθόνης

Αρχικά χρειαζόμαστε μια ρουτίνα που θα μεταφέρει τα δύο τμήματα των 4 bit κάθε εντολής. Η ρουτίνα θα πρέπει να αφήνει ανεπηρέαστους τους ακροδέκτες που επιλέγουν μεταξύ καταχωρητή εντολών και καταχωρητή δεδομένων, ώστε να μπορεί να χρησιμοποιηθεί και για τις δύο λειτουργίες.

### **Ρουτίνα: write\_2\_nibbles**

*write\_2\_nibbles:*

```
push r24 ; save r24(LCD_Data)

in r25 ,PIND ; read PIND

andi r25 ,0x0f ;
andi r24 ,0xf0 ; r24[3:0] Holds previus PORTD[3:0]
add r24 ,r25 ; r24[7:4] <-- LCD_Data_High_Byte
out PORTD ,r24 ;
```

```

sbi PORTD ,PD3          ; Enable Pulse
nop
nop
cbi PORTD ,PD3

pop r24                  ; Recover r24(LCD_Data)
swap r24                 ;
andi r24 ,0xf0           ; r24[3:0] Holds previus PORTD[3:0]
add r24 ,r25             ; r24[7:4] <-- LCD_Data_Low_Byte
out PORTD ,r24

sbi PORTD ,PD3          ; Enable Pulse
nop
nop
cbi PORTD ,PD3

ret

```

Στη συνέχεια, με βάση την προηγούμενη ρουτίνα θα δημιουργήσουμε δύο άλλες. Η μία θα στέλνει εντολές στην οθόνη και η άλλη δεδομένα.

#### **Ρουτίνα: lcd\_data**

Αποστολή ενός byte δεδομένων στον ελεγκτή της οθόνης lcd. Ο ελεγκτής πρέπει να βρίσκεται σε 4 bit mode. Το byte που μεταδίδεται είναι αποθηκευμένο στον καταχωρητή r24

```

lcd_data:
sbi PORTD ,PD2          ; LCD_RS=1(PD2=1), Data
rcall write_2_nibbles   ; send data
ldi r24 ,250            ;
ldi r25 ,0              ; Wait 250uSec
rcall wait_usec
ret

```

#### **Ρουτίνα: lcd\_command**

Αποστολή μιας εντολής στον ελεγκτή της οθόνης lcd.

Ο ελεγκτής πρέπει να βρίσκεται σε 4 bit mode.

Η εντολή που μεταδίδεται είναι αποθηκευμένη στον καταχωρητή r24

```

lcd_command:
cbi PORTD ,PD2          ; LCD_RS=0(PD2=0), Instruction
rcall write_2_nibbles   ; send Instruction
ldi r24 ,250            ;
ldi r25 ,0              ; Wait 250uSec
rcall wait_usec
ret

```

#### **Ρουτίνα: lcd\_clear\_disply**

Καθαρισμός της οθόνης LCD :

```

lcd_clear_display:

```

```

ldi r24 ,0x01          ; clear display command
rcall lcd_command

ldi r24 ,low(5)         ;
ldi r25 ,high(5)        ; Wait 5 mSec
rcall wait_msec         ;

ret

```

Για να μπορεί να χρησιμοποιηθεί οθόνη πρέπει να αρχικοποιηθεί στην επιθυμητή κατάσταση. Όταν η οθόνη τροφοδοτείται με ρεύμα για πρώτη φορά πραγματοποιείται μια εσωτερική αρχικοποίηση και για αυτό απαιτείται χρονική καθυστέρηση 200 ms πριν την τελική αρχικοποίηση.

Κάθε φορά που προγραμματίζουμε τον Μικροελεγκτή, ξεκινάει η εκτέλεση του κώδικα από την αρχή. Η οθόνη όμως βρίσκεται στην κατάσταση που την αφήσαμε την προηγούμενη φορά. Ο κώδικας που θα κάνει την αρχικοποίηση της οθόνης δεν γνωρίζει αν ο ελεγκτής της οθόνης βρίσκεται σε 8 bit mode ή σε 4 bit mode. Για το λόγο αυτό αρχικά στέλνεται 3 φορές η εντολή 0x30 (function set) για 8 bit mode. Αν ο ελεγκτής είναι σε 8 bit mode δεν θα αλλάξει κάτι, αν όμως είναι σε 4 bit mode θα μεταβεί σε 8 bit mode. Στη συνέχεια στέλνεται η εντολή 0x20 για να οδηγηθεί η οθόνη σε 4 bit mode. Μόλις είμαστε βέβαιοι για την μορφή που πρέπει να στέλνουμε τις εντολές μπορούμε να προχωρήσουμε με την αρχικοποίηση.

#### **Ρουτίνα: lcd\_init**

Αρχικοποίηση και ρυθμίσεις της οθόνης LCD όπως παρουσιάζεται παρακάτω:

```

;DL = 0    4 bit mode,
;N = 1     2 lines
;F = 0     5×8 dots
;D = 1     display on
;C = 0     cursor off
;B = 0     blinking off
;I/D = 1   DD-RAM address auto increment
;SH = 0    shift of entire display off

```

lcd\_init:

```

ldi r24 ,low(200)       ;
ldi r25 ,high(200)      ; Wait 200 mSec
rcall wait_msec         ;

ldi r24 ,0x30            ; command to switch to 8 bit mode
out PORTD ,r24           ;
sbi PORTD ,PD3           ; Enable Pulse
nop
nop
cbi PORTD ,PD3
ldi r24 ,250             ;
ldi r25 ,0               ; Wait 250uSec
rcall wait_usec         ;

ldi r24 ,0x30            ; command to switch to 8 bit mode
out PORTD ,r24           ;

```

```

sbi PORTD ,PD3          ; Enable Pulse
nop
nop
cbi PORTD ,PD3
ldi r24 ,250             ;
ldi r25 ,0               ; Wait 250uSec
rcall wait_usec         ;

ldi r24 ,0x30            ; command to switch to 8 bit mode
out PORTD ,r24           ;
sbi PORTD ,PD3          ; Enable Pulse
nop
nop
cbi PORTD ,PD3
ldi r24 ,250             ;
ldi r25 ,0               ; Wait 250uSec
rcall wait_usec         ;

ldi r24 ,0x20            ; command to switch to 4 bit mode
out PORTD ,r24           ;
sbi PORTD ,PD3          ; Enable Pulse
nop
nop
cbi PORTD ,PD3
ldi r24 ,250             ;
ldi r25 ,0               ; Wait 250uSec
rcall wait_usec         ;

ldi r24 ,0x28            ; 5x8 dots, 2 lines
rcall lcd_command

ldi r24 ,0x0c            ; display on, cursor off
rcall lcd_command
rcall lcd_clear_display

ldi r24 ,0x06            ; Increase address, no display shift
rcall lcd_command       ;
ret

```

### **Παράδειγμα**

Ο κώδικας που ακολουθεί προξενεί επαναληπτικά:

Καθαρισμό της οθόνης - Καθυστέρηση 1 sec - Απεικόνιση του χαρακτήρα 'Α' στην οθόνη - Καθυστέρηση 1 sec.

```
.include "m328PBdef.inc" ;ATmega328P microcontroller definition
```

```

.equ PD0=0
.equ PD1=1
.equ PD2=2
.equ PD3=3
.equ PD4=4
.equ PD5=5

```

```
.equ PD6=6
.equ PD7=7
```

*main:*

```
ldi r24, low(RAMEND)
out SPL, r24
ldi r24, high(RAMEND)
out SPH, r24          ; init stack pointer

ser r24
out DDRD, r24         ; set PORTD as output
```

```
clr r24
```

```
rcall lcd_init
```

```
ldi r24, low(100)
ldi r25, high(100)    ; delay 100 mS
rcall wait_msec
```

*main1:*

```
rcall lcd_clear_display
```

```
ldi r24, low(1000)
ldi r25, high(1000)
rcall wait_msec       ; delay 1 Sec
```

```
ldi r24, 'A'
call lcd_data
```

```
ldi r24, low(1000)
ldi r25, high(1000)
rcall wait_msec       ; delay 1 Sec
jmp main1
```

Παραδείγματα ρουτινών χρονοκαθυστερήσεων που χρησιμοποιούνται στα τμήματα κώδικα παραπάνω:

*wait\_msec:*

```
push r24          ; 2 cycles
push r25          ; 2 cycles
ldi r24, low(999) ; 1 cycle
ldi r25, high(999) ; 1 cycle
rcall wait_usec   ; 998.375 usec
pop r25           ; 2 cycles
pop r24           ; 2 cycles
nop              ; 1 cycle
nop              ; 1 cycle
```

```
sbiw r24, 1 ; 2 cycles
brne wait_msec ; 1 or 2 cycles
```

```
ret ; 4 cycles
```

```
wait_usec:  
    sbiw r24,1           ; 2 cycles (2/16 usec)  
    call delay_8cycles   ; 4+8=12 cycles  
    brne wait_usec       ; 1 or 2 cycles  
    ret
```

```
delay_8cycles:  
    nop  
    nop  
    nop  
    nop  
    ret
```

## Τα ζητούμενα της 4<sup>ης</sup> εργαστηριακής άσκησης

### Ζήτημα 4.1

Στην κάρτα ntuAboard\_G1 να γίνει η σύνδεση του POT4 με την είσοδο A3 του ADC, μέσω κατάλληλου συνδετήρα(Jumper) και να γραφεί πρόγραμμα **σε assembly** για τον ATmega328PB το οποίο θα ξεκινάει μια μετατροπή του ADC κάθε 1Sec. Ο ADC Θα μετατρέπει την τάση που υπάρχει κάθε φορά την είσοδο A1. Η ανάγνωση των δεδομένων του ADC πρέπει να γίνεται μέσα στην ρουτίνα εξυπηρέτησης της διακοπής ολοκλήρωσης μετατροπής του ADC. Η διακοπή αυτή (ADC) μεταφέρει τον έλεγχο στην διεύθυνση **0x02A**, αν είναι ενεργοποιημένη η αντίστοιχη διακοπή (από το bit ADIE του ADCSRA) καθώς και το flag επίτρεψης όλων των διακοπών. Η τιμή μέτρησης του ADC να μετατρέπεται σε τάση και να εκτυπώνονται στην LCD οθόνη, αρχίζοντας κάθε φορά από τον πρώτο χαρακτήρα της πρώτης γραμμής, με ακρίβεια δύο δεκαδικών ψηφίων (δεν χρειάζεται στρογγυλοποίηση).

Η τάση δίνεται από τον τύπο:

$$V_{IN} = \frac{ADC}{1024} V_{REF}$$

Όπου:

- $V_{IN}$  η τάση στην αναλογική είσοδο A3 του μικροελεγκτή.
- ADC η τιμή που διαβάζεται από τον ADC (10bit, από 0-1023)
- $V_{REF}$  η τάση αναφοράς που έχει οριστεί στα 5V.

### Ζήτημα 4.2

Να γραφεί σε γλώσσα C, πρόγραμμα για τον ATmega328PB το οποίο θα ξεκινάει μια ADC μετατροπή, όπως στο Ζήτημα 4.1. Δεν θα κάνει χρήση της διακοπής ολοκλήρωσης μετατροπής του ADC και θα περιμένει μέχρι να ολοκληρωθεί η ADC μετατροπή ελέγχοντας το bit ADSC του ADCSRA το οποίο γίνεται 0 μόλις ολοκληρωθεί η μετατροπή (Polling). Θα εκτυπώνει την τάση στην LCD οθόνη με ακρίβεια δύο δεκαδικών ψηφίων.

### Ζήτημα 4.3

Να δημιουργηθεί κώδικας **σε C** για την επιτήρηση ενός χώρου όπου υπάρχει αυξημένος κίνδυνος ύπαρξης μονοξειδίου του άνθρακα (CO). Ο αισθητήρας CO είναι συνδεδεμένος στην αναλογική είσοδο A3 του μικροελεγκτή.

Καθ' όλη την διάρκεια πρέπει να διαβάζεται η τιμή του αισθητήρα ανά 100 ms (μικρές αποκλίσεις είναι αποδεκτές) και να εμφανίζεται μια ένδειξη του επιπέδου του αερίου στα LED PB0-PB5.

Αν οποιαδήποτε στιγμή η συγκέντρωση του CO ξεπεράσει τα 75ppm να τυπώνεται στην LCD το μήνυμα GAS DETECTED και να αναβοσβήνουν τα αντίστοιχα LED στα PB0-PB5 αναλόγως του επιπέδου του αερίου.

Το επίπεδο θα πρέπει να συνεχίζει να διαβάζεται (και να εμφανίζεται στα LED) και αν επανέλθει σε επίπεδο χαμηλότερο των 75ppm τα LEDs να σταματούν να αναβοσβήνουν και να τυπώνεται στην LCD το μήνυμα CLEAR. Είναι καλή πρακτική να διατηρείτε τις ρουτίνες εξυπηρέτησης διακοπών όσο το δυνατόν μικρότερες σε χρονική διάρκεια. Παρόλα αυτά αν το πρόγραμμα σας είναι λειτουργικό δεν θα υπάρξει αρνητική επίπτωση στην βαθμολογία.

Για την μέτρηση της συγκέντρωσης του CO χρησιμοποιείται αισθητήρας του τύπου ULPSM-CO-968-001:

([https://helios.ntua.gr/pluginfile.php/250446/mod\\_folder/content/0/ULPSM-CO-968-001.pdf?forcedownload=1](https://helios.ntua.gr/pluginfile.php/250446/mod_folder/content/0/ULPSM-CO-968-001.pdf?forcedownload=1)),

ο οποίος βρίσκεται σε σταθερές συνθήκες θερμοκρασίας (20°C) και υγρασίας (40%). Δίνεται ότι  $V_{gas0}=0.1V$  και sensitivity code 129nA/ppm. Για να προσομοιώσετε την έξοδο που θα είχε ο αισθητήρας CO να χρησιμοποιήσετε το ποτενσιόμετρο POT4 στην κάρτα ntuAboard\_G1 το οποίο παράγει μια DC τάση στο εύρος 0-5V.