

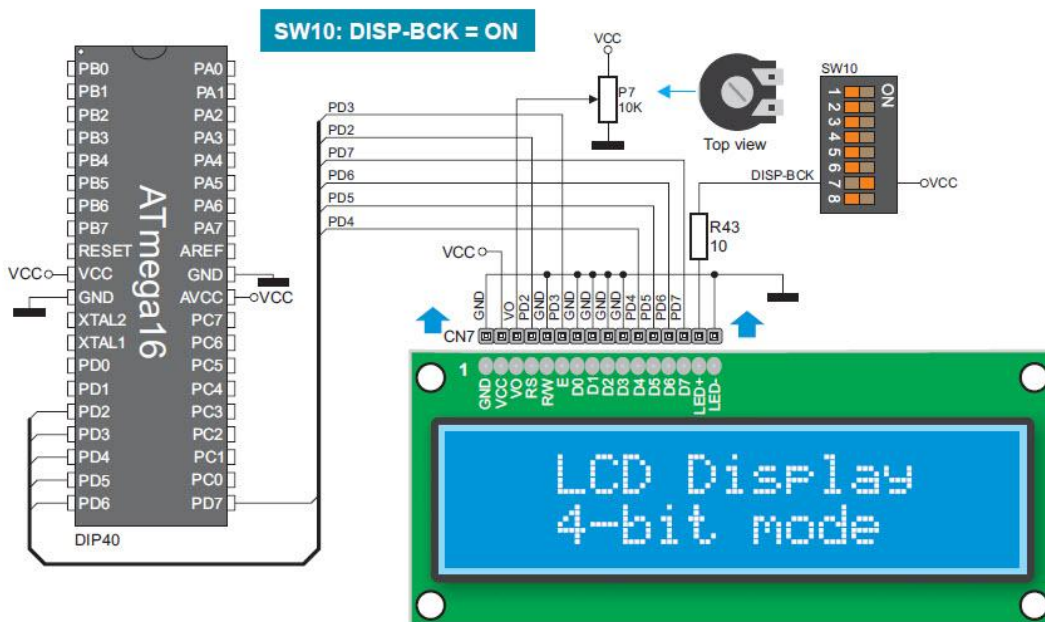
Εργαστηριακές ασκήσεις στον Μικροελεγκτή AVR

4^η Εργαστηριακή Άσκηση του AVR – Χρήση οθόνη

Στην άσκηση αυτή θα γίνει μελέτη χρήσης της αλφαριθμητικής οθόνη LCD 2×16 χαρακτήρων (επικοινωνία μεταξύ οθόνης και του μικροελεγκτή γίνεται με λέξεις των 4 bit). Στην συνέχεια παρουσιάζεται η συσκευή αυτή σε συνδυασμό με το αντίστοιχο λογισμικό οδήγησής της.

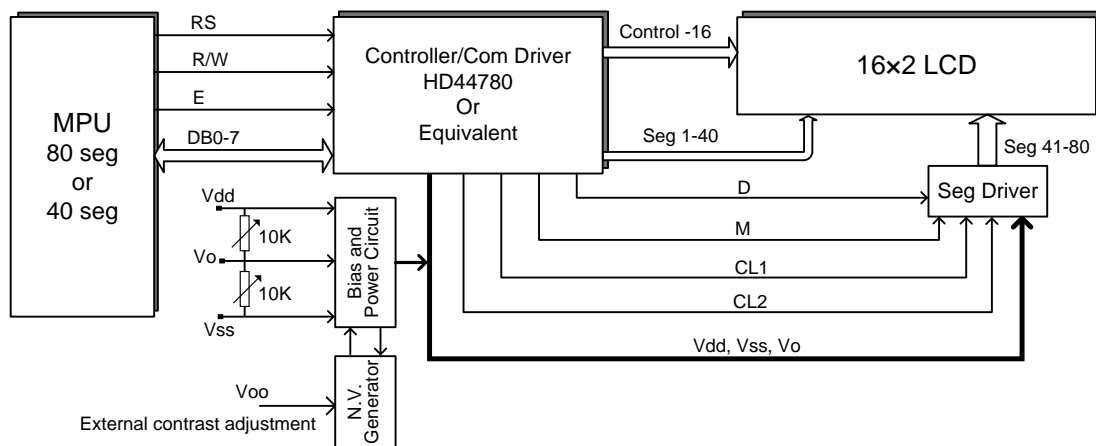
Αλφαριθμητική Οθόνη Χαρακτήρων 2×16

Στην αναπτυξιακή πλακέτα EasyAVR6 μια πολύ χρήσιμη περιφερειακή συσκευή είναι η πρόσθετη οθόνη χαρακτήρων 2×16. Η συνδεσμολογία γίνεται με τους 6 περισσότερο σημαντικούς ακροδέκτες της θύρας PORTD (PD2-PD7), όπως φαίνεται στο σχήμα 7.1. Οι ακροδέκτες αυτοί συνδέονται σε ακροδέκτες του ελεγκτή της οθόνης WH1602B, του οποίου το εγχειρίδιο τεχνικών προδιαγραφών παρατίθεται στις επόμενες σελίδες. Από τους 6 ακροδέκτες, οι 2 λιγότερο σημαντικοί (PD2 και PD3) είναι ακροδέκτες ελέγχου ενώ οι υπόλοιποι 4 σχηματίζουν λέξεις των 4 bit με τις οποίες γίνεται η επικοινωνία μικροελεγκτή – ελεγκτή οθόνης.

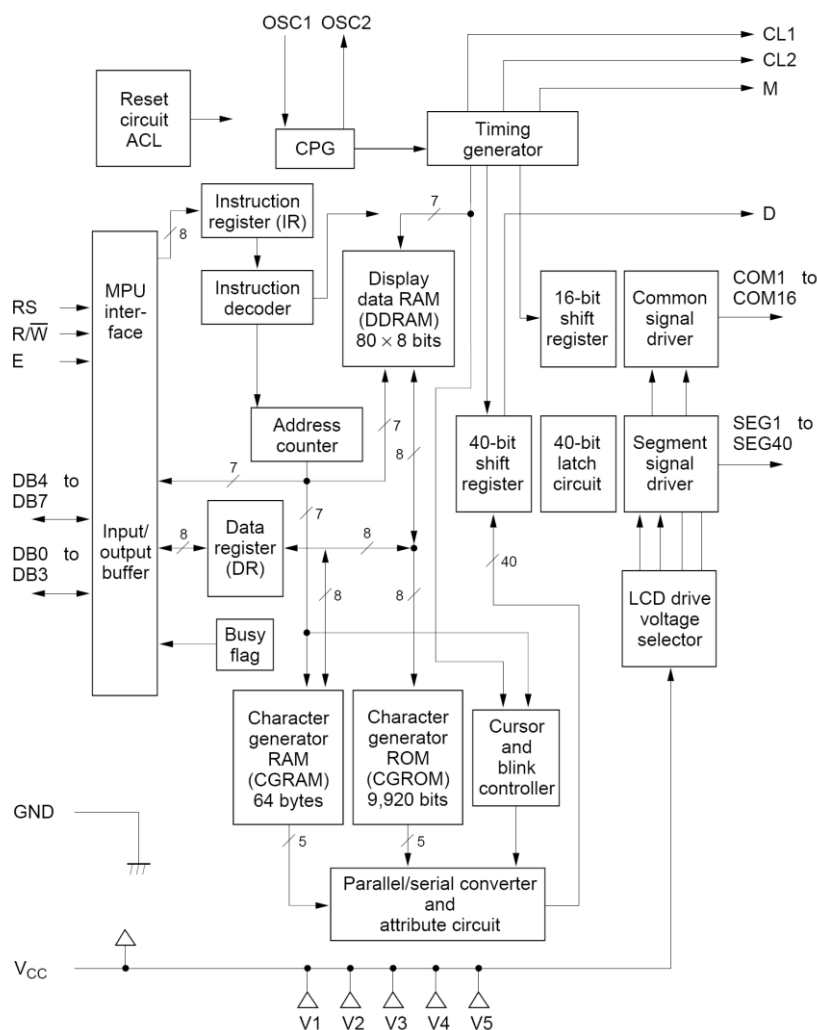


Σχήμα 7.1. Κυκλωματικό διάγραμμα οθόνης χαρακτήρων 2x16.

Για να χρησιμοποιήσουμε την αλφαριθμητική οθόνη χαρακτήρων του easyAVR6 είναι απαραίτητο να εξοικειωθούμε με τη δομή και τη λειτουργία του. Η εσωτερική οργάνωση της αλφαριθμητικής οθόνης χαρακτήρων παρουσιάζεται στο ακόλουθο σχήμα.



Κεντρικό ρόλο παίζει ο ελεγκτής HD44780, ο οποίος αναλαμβάνει να παρουσιάσει στην οθόνη υγρών κρυστάλλων τους αλφαριθμητικούς χαρακτήρες καθώς και την αλληλεπίδραση με τον χρήστη. Το μπλοκ διάγραμμα του ελεγκτή HD44780 φαίνεται στο παρακάτω σχήμα.



Περιγραφή λειτουργία

Η μονάδα οθόνης LCD περιλαμβάνει ελεγκτή που έχει δύο 8-bit καταχωρητές, έναν καταχωρητή εντολών (instruction register -IR) και έναν καταχωρητή δεδομένων (data register- DR).

Καταχωρητές

Ο καταχωρητής IR αποθηκεύει κωδικούς εντολών, όπως καθαρισμός οθόνης ολίσθηση δρομέα, και πληροφορίες για τη διεύθυνση των δεδομένων της RAM απεικόνισης (display data RAM -DDRAM) και για τη γεννήτρια χαρακτήρων RAM (character generator RAM- CGRAM). Ο καταχωρητής IR εγγράφεται μόνο από τον Μικροελεγκτή. Ο καταχωρητής DR αποθηκεύει προσωρινά τα δεδομένα για να εγγράφουν σε DDRAM ή CGRAM και προσωρινά αποθηκεύει δεδομένα που πρέπει να διαβαστούν από την DDRAM ή την CGRAM. Τα δεδομένα γράφονται στον DR από τον Μικροελεγκτή αυτόματα μεταφέρονται σε DDRAM ή CGRAM από μια εσωτερική λειτουργία. Το DR χρησιμοποιείται επίσης για αποθήκευση κατά την ανάγνωση δεδομένων από DDRAM ή CGRAM. Όταν πληροφορίες διεύθυνσης εγγράφονται στο IR, τα αντίστοιχα δεδομένα αποθηκεύονται στον DR από την DDRAM ή την CGRAM μέσω μιας εσωτερικής λειτουργίας. Η μεταφορά δεδομένων στον MPU ολοκληρώνεται όταν διαβάσει τον DR. Μετά την ανάγνωση, τα δεδομένα των DDRAM ή CGRAM της επόμενης διεύθυνσης αποστέλλονται στον DR για την επόμενη ανάγνωση από τον Μικροελεγκτή. Με το σήμα επιλογής καταχωρητή (register selector -RS), οι δύο αυτοί καταχωρητές μπορούν να επιλεγούν (Πίνακας 1).

Σημαία απασχόλησης (Busy Flag-BF)

Όταν η σημαία απασχόλησης είναι 1, η HD44780U είναι σε κατάσταση εσωτερικής λειτουργίας, και επόμενη εντολή δεν γίνεται δεκτή. Όταν RS = 0 και R / W = 1 (Πίνακας 1), η σημαία απασχόλησης εμφανίζεται στην έξοδο DB7. Η επόμενη εντολή πρέπει να είναι 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 DDRAM και CGRAM. Όταν μια διεύθυνση μιας εντολής εγγράφεται στον IR, η διεύθυνση στέλνεται από τον IR στον AC. Η επιλογή είτε της DDRAM ή της CGRAM καθορίζεται επίσης από την εντολή.

Μετά την εγγραφή προς (ή την ανάγνωση από) τις DDRAM ή CGRAM, η AC αυτόματα αυξάνεται κατά 1 (ή ελαττώνεται κατά 1). Το περιεχόμενο του καταχωρητή AC δίνεται στη έξοδο μέσω των DB0 - DB6 όταν RS = 0 και R / W = 1 (βλέπε Πίνακα 1).

Πίνακας 1. Επιλογή Καταχωρητή

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 DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Περιγραφή των εσωτερικών δομικών μονάδων του ελεγκτή

Μνήμες

Ο ελεγκτής διαθέτει συνολικά 3 μνήμες, δύο τύπου RAM και μια τύπου ROM. Η ROM δημιουργίας χαρακτήρων χρησιμοποιείται για τη δημιουργία εικόνων χαρακτήρων 5×8 κουκίδων ή 5×10 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	0	P	'	P	5	E	Δ			Γ	Β
LLLH	CG RAM (2)	≡	!	1	A	Q	a	4	Q	a	i				J	γ
LLHL	CG RAM (3)	7	"	2	B	R	b	r	e	E	Δ				o	z
LLHH	CG RAM (4)	L	#	3	C	S	c	s	Δ	Δ	Δ				P	ψ
LHLL	CG RAM (5)	J	\$	4	D	T	d	t	Δ	Δ	Δ				4	ζ
LHLH	CG RAM (6)	J	%	5	E	U	e	u	Δ	Δ	E				Δ	η
LHHL	CG RAM (7)	J	&	6	F	V	f	v	Δ	Δ	Δ				u	θ
LHHH	CG RAM (8)	J	'	7	G	W	g	w	Δ	Δ	Δ				x	ι
HLLL	CG RAM (1)	J	(8	H	X	h	x	Δ	Δ	Δ				÷	κ
HLLH	CG RAM (2)	J)	9	I	Y	i	y	Δ	Δ	Δ				Δ	λ
HLHL	CG RAM (3)	*	*		J	Z	j	z	Δ	Δ	Δ				Δ	μ
HLHH	CG RAM (4)	J	+		K	C	k	c	Δ	Δ	Δ				Δ	ν
HHLL	CG RAM (5)	≡	,	<	L	\	l	l	Δ	Δ	Δ				Δ	ξ
HHLH	CG RAM (6)	Δ	Δ	≡	M	Δ	m	Δ	Δ	Δ	Δ				Δ	π
HHHL	CG RAM (7)	Δ	.	>	N	^	n	^	Δ	Δ	Δ				Δ	ρ
HHHH	CG RAM (8)	Δ	/	?	O	_	o	_	Δ	Δ	Δ				Δ	σ

Η RAM δημιουργίας χαρακτήρων (character generator RAM- CGRAM) εκτελεί την ίδια λειτουργία με την ROM δημιουργίας χαρακτήρων, με την διαφορά ότι οι χαρακτήρες που απεικονίζονται πρέπει να δημιουργηθούν από τον χρήστη και να αποθηκευτούν σε αυτή.

Η RAM απεικόνισης δεδομένων (display data RAM –**DDRAM**) αποθηκεύει τα δεδομένα που πρόκειται να απεικονιστούν στην οθόνη. Η χωρητικότητά της είναι 80 byte. Η αντιστοιχία μεταξύ θέσεων στην RAM απεικόνισης χαρακτήρων και θέσεων στην οθόνη φαίνεται στην ακόλουθη εικόνα. Παρατηρήστε ότι αρκετές θέσεις της μνήμης δεν απεικονίζονται στην οθόνη και πως όταν η οθόνη έχει διαμόρφωση για απεικόνιση

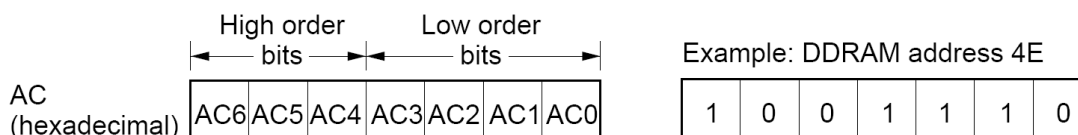
χαρακτήρων σε δύο γραμμές, οι διευθύνσεις που απεικονίζονται στην πρώτη και δεύτερη γραμμή της οθόνης δεν είναι συνεχόμενες.

DDRAM Address

Display Data RAM (DDRAM)

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80×8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. See Figure 1 for the relationships between DDRAM addresses and positions on the liquid crystal display.

The DDRAM address (A_{DD}) is set in the address counter (AC) as hexadecimal.



1-line display ($N = 0$) (Figure 2)

— When there are fewer than 80 display characters, the display begins at the head position. For example, if using only the HD44780, 8 characters are displayed. See Figure 3.

When the display shift operation is performed, the DDRAM address shifts.

Display position (digit)	1	2	3	4	5			79	80
DDRAM address (hexadecimal)	00	01	02	03	04		4E	4F

2-line display ($N = 1$) (Figure 4)

— Case 1: When the number of display characters is less than 40×2 lines, the two lines are displayed from the head. Note that the first line end address and the second line start address are not consecutive. For example, when just the HD44780 is used, 8 characters \times 2 lines are displayed.

Display position	1	2	3	4	5			39	40
DDRAM address (hexadecimal)	00	01	02	03	04		26	27
	40	41	42	43	44		66	67

When display shift operation is performed, the DDRAM address shifts.

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07
	40	41	42	43	44	45	46	47
For shift left	01	02	03	04	05	06	07	08
	41	42	43	44	45	46	47	48
For shift right	27	00	01	02	03	04	05	06
	67	40	41	42	43	44	45	46

— Case 2: For a 16-character \times 2-line display, the HD44780 can be extended using one 40-output

extension driver.

When display shift operation is performed, the DDRAM address shifts.

Display position	1	2	3	4	5	6	7	8
DDRAM address	00	01	02	03	04	05	06	07

For shift left	01	02	03	04	05	06	07	08
----------------	----	----	----	----	----	----	----	----

For shift right	4F	00	01	02	03	04	05	06
-----------------	----	----	----	----	----	----	----	----

1-Line by 8-Character Display Example

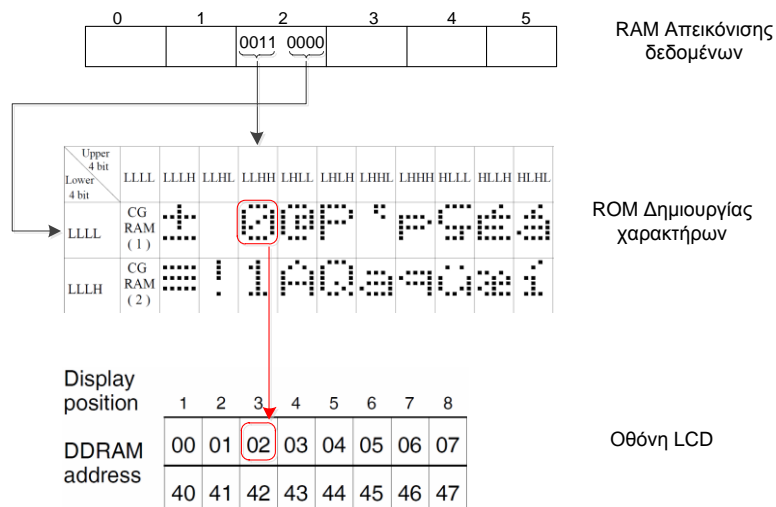
Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	HD44780U display								Extension driver display							

For shift left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50

For shift right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

2-Line by 16-Character Display Example

Ο ρόλος των μνημών RAM και ROM στην απεικόνιση δεδομένων στην οθόνη αποδίδεται από το παρακάτω σχεδιάγραμμα.



Ο ελεγκτής HD44780 έχει δύο καταχωρητές των 8 bit που μπορούν να προσπελαστούν άμεσα από το χρήστη. Τον καταχωρητή εντολών και τον καταχωρητή δεδομένων.

Ο καταχωρητής εντολών αποθηκεύει δυαδικούς κωδικούς εντολών και διευθύνσεις για τις μνήμες RAM του ελεγκτή.

Ο καταχωρητής δεδομένων αποθηκεύει προσωρινά τα δεδομένα που πρόκειται να αποθηκευτούν στις RAM του ελεγκτή ή δεδομένα που διαβάστηκαν από αυτές. Δεδομένα που αποθηκεύονται στον καταχωρητή δεδομένων μεταφέρονται αυτόματα στην κατάλληλη RAM. Όταν ο καταχωρητής εντολών δεχτεί πληροφορίες διεύθυνσης, δεδομένα αποθηκεύονται στον καταχωρητή δεδομένων από την κατάλληλη RAM αυτόματα.

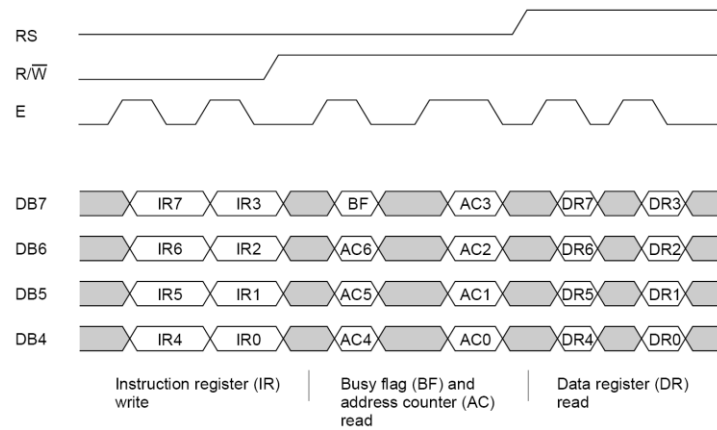
Στο μετρητή διευθύνσεων βρίσκεται αποθηκευμένη η τρέχουσα διεύθυνση της RAM απεικόνισης δεδομένων ή της RAM δημιουργίας χαρακτήρων. Το περιεχόμενό του μπορεί να τροποποιηθεί με μια από τις εντολές που αλλάζουν την διεύθυνση κάποιας από τις RAM του ελεγκτή. Το ποια μνήμη επιλέγεται εξαρτάται από την εντολή. Η διεύθυνση που βρίσκεται αποθηκευμένη σε αυτόν τον καταχωρητή αυξάνεται κατά 1 (ή μειώνεται κατά 1) αυτόματα με κάθε ανάγνωση/εγγραφή από τη μνήμη.

Η διεπαφή με την οποία κάποιος ανταλλάσσει εντολές με τον ελεγκτή φαίνεται στον παρακάτω πίνακα.

Pin No.	Symbol	Level	Description
1	V _{SS}	0V	Ground
2	V _{DD}	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	—	LED +
16	K	—	LED —

Ο ελεγκτής HD44780 μπορεί να δεχτεί εντολές είτε σε μια ενιαία εντολή των 8 bit είτε σε μια εντολή που αποστέλλεται σε δύο διαδοχικά τμήματα των 4 bit. Στο αναπτυξιακό δεν μπορούμε να στείλουμε εντολές στον ελεγκτή σε μορφή 8 bit, συνεπώς δεν θα ασχοληθούμε άλλο με την μεταφορά εντολών σε αυτή τη μορφή.

Όταν οι εντολές μεταφέρονται σε δύο τμήματα των 4 bit χρησιμοποιούνται μόνο οι γραμμές DB7 – DB4. Τα 4 περισσότερο σημαντικά bit πρέπει να μεταφερθούν πρώτα. Επειδή δεν υπάρχει δυνατότητα ανάγνωσης της σημαίας BUSY FLAG για να γνωρίζουμε πότε ο ελεγκτής είναι έτοιμος να δεχτεί νέα εντολή πρέπει να εισάγουμε καθυστέρηση μεταξύ των διαδοχικών εντολών. Ο χρόνος που χρειάζεται ο ελεγκτής HD44780 για να εκτελέσει κάθε εντολή είναι μεταβλητός και εξαρτάται από την εντολή.



Το σύνολο εντολών του ελεγκτή 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 μ 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 μ 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 μ 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:5x11 dots/5x8 dots)	39 μ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 μ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 μ 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 μ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 μ s
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 μ s

* "—" : don't care

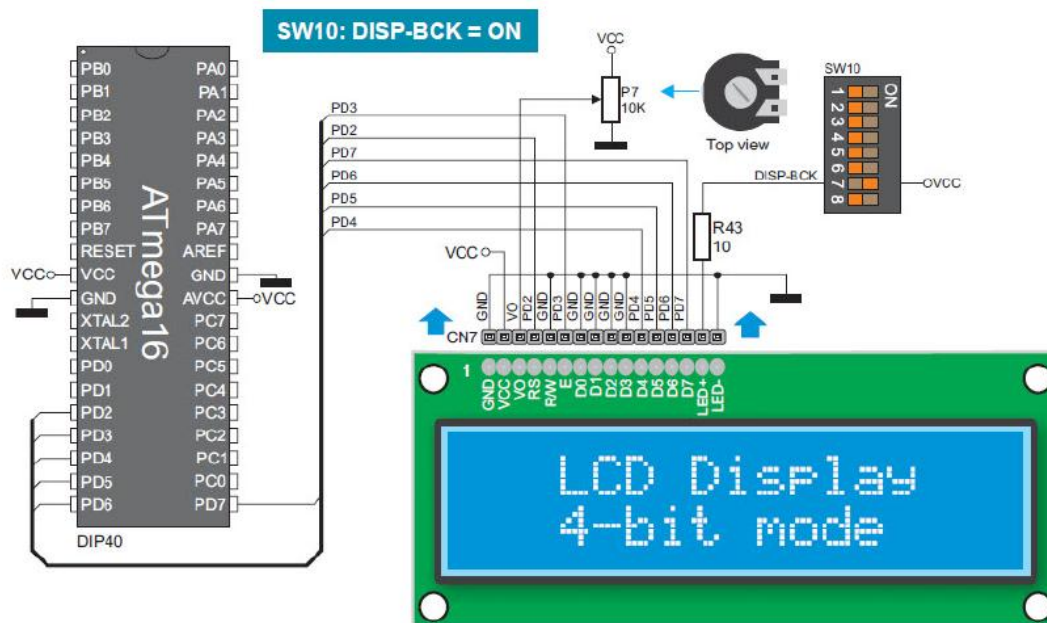
Τέλος για να γίνει δεκτή μια εντολή από τον ελεγκτή πρέπει να τηρούνται οι χρόνοι του παρακάτω πίνακα.

Ta=25°C, VDD=5.0V

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	T _C	1200	—	—	ns
Enable pulse width	T _{PW}	140	—	—	ns
Enable rise/fall time	T _R , T _F	—	—	25	ns
Address set-up time (RS, R/W to E)	t _{AS}	0	—	—	ns
Address hold time	t _{AH}	10	—	—	ns
Data set-up time	t _{DSW}	40	—	—	ns
Data hold time	t _H	10	—	—	ns

Χρήση της οθόνης LCD του αναπτυξιακού easyAVR6.

Η συνδεσμολογία της οθόνης lcd με τον μικροελεγκτή φαίνεται στο παρακάτω σχήμα, απ' όπου μπορούμε να συμπεράνουμε ότι οι ακροδέκτες PD7 – PD2 πρέπει να είναι ρυθμισμένοι για έξοδο.



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

Αποστολή ενός byte, 4 bit τη φορά στον ελεγκτή της οθόνης lcd. Το λογικό επίπεδο που βρίσκεται ο ακροδέκτης που αντιστοιχεί στο σήμα R/S δεν επηρεάζεται.

Είσοδος: Το byte που μεταδίδεται είναι αποθηκευμένο στον καταχωρητή r24

Έξοδος: -

Καταχωρητές: r25:r24

Καλούμενες υπορουτίνες: -

write_2_nibbles:

```
push r24 ; στέλνει τα 4 MSB
in r25 ,PIND ; διαβάζονται τα 4 LSB και τα ξαναστέλνουμε
andi r25 ,0x0f ; για να μην χαλάσουμε την όποια προηγούμενη κατάσταση
andi r24 ,0xf0 ; απομονώνονται τα 4 MSB και
add r24 ,r25 ; συνδυάζονται με τα προϋπάρχοντα 4 LSB
out PORTD ,r24 ; και δίνονται στην έξοδο
sbi PORTD ,PD3 ; δημιουργείται παλμός Enable στον ακροδέκτη PD3
cbi PORTD ,PD3 ; PD3=1 και μετά PD3=0
pop r24 ; στέλνει τα 4 LSB. Ανακτάται το byte.
swap r24 ; εναλλάσσονται τα 4 MSB με τα 4 LSB
andi r24 ,0xf0 ; που με την σειρά τους αποστέλλονται
add r24 ,r25
out PORTD ,r24
sbi PORTD ,PD3 ; Νέος παλμός Enable
cbi PORTD ,PD3
ret
```

Στη συνέχεια, με βάση την προηγούμενη ρουτίνα μπορούμε να 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.

Είσοδος: Το byte που μεταδίδεται είναι αποθηκευμένο στον καταχωρητή r24

Καταχωρητές: r25:r24

Καλούμενες υπορουτίνες: wait_usec, write_2_nibbles

lcd_data:

```
sbi PORTD ,PD2      ; επιλογή του καταχωρητή δεδομένων (PD2=1)
rcall write_2_nibbles ; αποστολή του byte
ldi r24 ,43          ; αναμονή 43μsec μέχρι να ολοκληρωθεί η λήψη
ldi r25 ,0            ; των δεδομένων από τον ελεγκτή της lcd
rcall wait_usec
ret
```

Ρουτίνα: lcd_command

Αποστολή μιας εντολής στον ελεγκτή της οθόνης lcd. Ο ελεγκτής πρέπει να βρίσκεται σε 4 bit mode.

Είσοδος: Η εντολή που μεταδίδεται είναι αποθηκευμένη στον καταχωρητή r24

Έξοδος: -

Καταχωρητές: r25:r24

Καλούμενες υπορουτίνες: wait_usec, write_2_nibbles

lcd_command:

```
cbi PORTD ,PD2      ; επιλογή του καταχωρητή εντολών (PD2=1)
rcall write_2_nibbles ; αποστολή της εντολής και αναμονή 39μsec
ldi r24 ,39          ; για την ολοκλήρωση της εκτέλεσης της από τον ελεγκτή της lcd.
ldi r25 ,0            ; ΣΗΜ.: υπάρχουν δύο εντολές, οι clear display και return home,
rcall wait_usec      ; που απαιτούν σημαντικά μεγαλύτερο χρονικό διάστημα.
ret
```

Τώρα που μπορούμε να στείλουμε εντολές και δεδομένα στην οθόνη μένει να την αρχικοποιήσουμε στην επιθυμητή κατάσταση, ώστε να μπορεί να χρησιμοποιηθεί. Όταν η οθόνη τροφοδοτείται με ρεύμα για πρώτη φορά ο ελεγκτής HD44780 πραγματοποιεί μια εσωτερική αρχικοποίηση και για αυτό απαιτείται να περιμένουμε 40 ms. Στη συνέχεια ο ελεγκτής βρίσκεται σε 8 bit mode και είναι έτοιμος να λάβει εντολές. Ο κώδικας που θα κάνει την αρχικοποίηση δεν πρέπει να βασίζεται στο ότι ο ελεγκτής βρίσκεται σε 8 bit mode, διότι αυτό δεν είναι πάντα αληθές. Κάθε φορά που προγραμματίζουμε τον μικροελεγκτή αυτός ξεκινάει την εκτέλεση του κώδικα από την αρχή, η οθόνη όμως βρίσκεται στην κατάσταση που την αφήσαμε την προηγούμενη φορά. Για να οδηγήσουμε την οθόνη σε 4 bit mode στέλνουμε δύο φορές την εντολή 0x30 (function set) για 8 bit mode. Η συγκεκριμένη εντολή (μεταξύ άλλων) ρυθμίζει τον ελεγκτή να δέχεται εντολές και δεδομένα σε ένα ενιαίο κομμάτι των 8 bit. Τα 4 λιγότερο σημαντικά bit μας είναι αδιάφορα. 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. 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 DDRAM address auto increment

SH = 0 shift of entire display off

Είσοδος: -

Έξοδος: -

Καταχωρητές: r25:r24

Καλούμενες υπορουτίνες: wait_msec, wait_usec, lcd_command

lcd_init:

```
ldi r24 ,40          ; Όταν ο ελεγκτής της lcd τροφοδοτείται με
ldi r25 ,0           ; ρεύμα εκτελεί την δική του αρχικοποίηση.
rcall wait_msec      ; Αναμονή 40 msec μέχρι αυτή να ολοκληρωθεί.

ldi r24 ,0x30         ; εντολή μετάβασης σε 8 bit mode
out PORTD ,r24       ; επειδή δεν μπορούμε να είμαστε βέβαιοι
sbi PORTD ,PD3       ; για τη διαμόρφωση εισόδου του ελεγκτή
cbi PORTD ,PD3       ; της οθόνης, η εντολή αποστέλλεται δύο φορές
ldi r24 ,39
ldi r25 ,0           ; εάν ο ελεγκτής της οθόνης βρίσκεται σε 8-bit mode
rcall wait_usec      ; δεν θα συμβεί τίποτα, αλλά αν ο ελεγκτής έχει διαμόρφωση
                    ; εισόδου 4 bit θα μεταβεί σε διαμόρφωση 8 bit

ldi r24 ,0x30
out PORTD ,r24
sbi PORTD ,PD3
cbi PORTD ,PD3
ldi r24 ,39
ldi r25 ,0
rcall wait_usec

ldi r24 ,0x20         ; αλλαγή σε 4-bit mode
out PORTD ,r24
sbi PORTD ,PD3
cbi PORTD ,PD3
ldi r24 ,39
ldi r25 ,0
rcall wait_usec

ldi r24 ,0x28         ; επιλογή χαρακτήρων μεγέθους 5x8 κουκίδων
rcall lcd_command    ; και εμφάνιση δύο γραμμών στην οθόνη

ldi r24 ,0x0c         ; ενεργοποίηση της οθόνης, απόκρυψη του κέρσορα
rcall lcd_command

ldi r24 ,0x01         ; καθαρισμός της οθόνης
rcall lcd_command
ldi r24 ,low(1530)
ldi r25 ,high(1530)
rcall wait_usec

ldi r24 ,0x06         ; ενεργοποίηση αυτόματης αύξησης κατά 1 της διεύθυνσης
rcall lcd_command    ; που είναι αποθηκευμένη στον μετρητή διευθύνσεων και
                    ; απενεργοποίηση της ολίσθησης ολόκληρης της οθόνης

ret
```

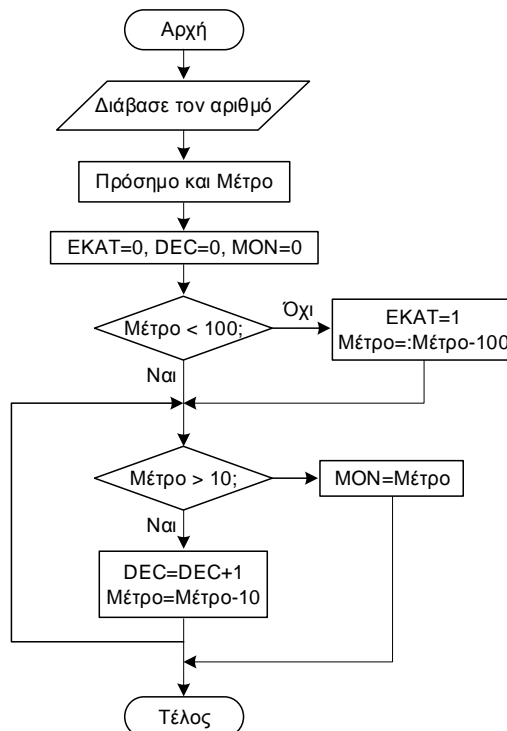
Τα ζητούμενα της 4^{ης} εργαστηριακής άσκησης του AVR

Ζήτημα 4.1 Γράψτε ένα πρόγραμμα που να απεικονίζει στην οθόνη LCD το πλήκτρο που πατήθηκε τελευταίο στο keypad. Μέχρι να πατηθεί κάποιο πλήκτρο, η οθόνη πρέπει να εμφανίζει την ένδειξη **KENO**. Τόσο η ένδειξη **KENO** όσο και το πλήκτρο που πατήθηκε να εμφανίζονται στην πάνω αριστερή θέση της οθόνης. Ο κέρσορας να μην είναι ορατός.

Ζήτημα 4.2 Γράψτε ένα πρόγραμμα που να εξομοιώνει ένα σύστημα συναγερμού. Υποθέτουμε ότι τα push buttons PB0-PB7 αντιστοιχούν στις εξόδους των αισθητήρων. Αν κάποιο από αυτά πατηθεί (ενεργοποιηθεί) τότε μέσα σε 5 sec πρέπει να πληκτρολογηθεί στο keypad ο αριθμός της ομάδας (δύο ψηφία) και ένας μονοψήφιος (1-4) ακόμα αριθμός που να δηλώνει την βάρδια. Αν είναι σωστά, να εμφανίζεται το μήνυμα **ALARM OFF** στο LCD display. Αλλιώς τίθεται ο συναγερμός με το συνεχόμενο άναμμα-σβήσιμο των leds PA0-PA7 (με περίοδο ~0.5 sec, 0.4 sec ON και 0.1 sec OFF) που υποθέτουμε ότι αντιστοιχεί στην ενεργοποίηση της σειρήνας. Ταυτόχρονα να εμφανίζεται στο LCD display το μήνυμα **ALARM ON**. Κατά την εισαγωγή του κωδικού να εμφανίζονται τα πλήκτρα που πατάει ο χρήστης στο LCD display καθώς και ο κέρσορας. Όλα τα μηνύματα να ξεκινούν από την πάνω αριστερή θέση του display, ενώ το υπόλοιπο να είναι κενό.

Ζήτημα 4.3 Γράψτε πρόγραμμα που να απεικονίζει στο LCD display την δυαδική τιμή (υποθέτουμε ότι είναι σε συμπλήρωμα ως προς 1) που διαβάζει από την θύρα PORTA και τη δεκαδική τιμή του σε μορφή τριών ψηφίων με το πρόσημο (δηλαδή αν από τη θύρα διαβαστεί π.χ. ο αριθμός 0110 0001, τότε ξεκινώντας από την πάνω αριστερή θέση του display να εμφανίζει **01100001=+97** ενώ αν δοθεί ο αριθμός 1000 0000 να εμφανίζεται το **10000000=-127**). Προσοχή στο να περιλαμβάνει την διπλή αναπαράσταση του μηδενός **0=11111111**. Η διαδικασία να είναι συνεχόμενη.

Δίνεται στο Σχήμα 7.2 ένα διάγραμμα ροής για τη μετατροπή ενός προσημασμένου δυαδικού αριθμού σε BCD μορφή που αν θέλετε μπορείτε να ακολουθήσετε.



Σχήμα 7.2 Διάγραμμα ροής για τη μετατροπή προσημασμένου δυαδικού αριθμού σε BCD μορφή

Ζήτημα 4.4 Δημιουργήστε ένα ψηφιακό χρονόμετρο. Το πρόγραμμα σας θα πρέπει να απεικονίζει στην οθόνη LCD το χρόνο που έχει περάσει από τη στιγμή που ενεργοποιήθηκε το χρονόμετρο στη μορφή:

Λεπτά:Δευτερόλεπτα

Τα λεπτά και τα δευτερόλεπτα θα πρέπει να απεικονίζονται με δύο δεκαδικά ψηφία και όταν το χρονόμετρο φτάνει την τιμή (59:59) θα πρέπει να ξεκινάει πάλι από την αρχή. Το χρονόμετρο θα μηδενίζεται και θα ξεκινάει όταν πατάμε και αφήνουμε το πλήκτρο PB0 και με την προϋπόθεση ότι τα πλήκτρα PB1 και PB2 δεν είναι ενεργοποιημένα. Θα σταματάει όσο πατάμε το πλήκτρο PB1, όταν το αφήνουμε να συνεχίζει την μέτρηση και να σταματάει οριστικά (χωρίς να χαλάσει η ένδειξη) όταν πατήσουμε το πλήκτρο PB2. Κατά την έναρξη λειτουργίας του προγράμματός σας θα πρέπει να εμφανίζεται στην οθόνη η ένδειξη 00:00 και το χρονόμετρο να είναι εκτός λειτουργίας. Κάντε όσες άλλες υποθέσεις σας χρειάζονται για την υλοποίηση.