



ΕΝΟΤΗΤΑ 6: LCD ΟΘΟΝΕΣ (ΥΓΡΩΝ ΚΡΥΣΤΑΛΛΩΝ)

ΣΤΟΧΟΙ

Όπως μπορείτε να φανταστείτε υπάρχει μεγάλη γκάμα περιφερειακών εισόδου-εξόδου. Μέχρι τώρα, χρησιμοποιήσαμε τα πιο κοινά από αυτά – απλά και οικονομικά πλήκτρα ή διακόπτες σαν συσκευές εισόδου και LED σαν συσκευές εξόδου.

Πλέον όμως ήρθε η ώρα να γνωρίσουμε άλλα, πιο σημαντικά και περισσότερο γνωστά ψηφιακά περιφερειακά. Σε αυτή την ενότητα θα εξοικειωθούμε με τις οθόνες LCD (οθόνες υγρών κρυστάλλων) ως περιφερειακό εξόδου που μας επιτρέπει να απεικονίσουμε οποιοδήποτε είδος πληροφορίας επιθυμούμε, συμπεριλαμβανομένων αριθμών, γραμμάτων και συμβόλων.

ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ

- ΟΘΟΝΕΣ LCD
- ΤΟ ΣΕΤ ΧΑΡΑΚΤΗΡΩΝ
- GRAPHIC CHARACTERS
- Η ΒΙΒΛΙΟΘΗΚΗ LiquidCrystal.h
- Η ΜΝΗΜΗ ΔΕΔΟΜΕΝΩΝ EEPROM

ΕΝΟΤΗΤΑ ΕΞΑΣΚΗΣΗΣ

- Η ΣΥΝΔΕΣΜΟΛΟΓΙΑ ΤΩΝ ΟΘΟΝΩΝ LCD
- ΠΑΡΑΔΕΙΓΜΑ 1: Hello world!
- ΠΑΡΑΔΕΙΓΜΑ 2: Η οθόνη
- ΠΑΡΑΔΕΙΓΜΑ 3: Ο δρομέας
- ΠΑΡΑΔΕΙΓΜΑ 4: Αναβόσβημα
- ΠΑΡΑΔΕΙΓΜΑ 5: Κατεύθυνση κειμένου



- ΠΑΡΑΔΕΙΓΜΑ 6: Κύλιση
- ΠΑΡΑΔΕΙΓΜΑ 7: Αυτόματη κύλιση
- ΠΑΡΑΔΕΙΓΜΑ 8: Ειδικοί χαρακτήρες
- ΠΑΡΑΔΕΙΓΜΑ 9: Απεικόνιση
- ΠΑΡΑΔΕΙΓΜΑ 10: Απεικόνιση φυσικών αριθμών
- ΠΑΡΑΔΕΙΓΜΑ 11: Απεικόνιση αριθμών κινητής υποδιαστολής
- ΠΑΡΑΔΕΙΓΜΑ 12: Μενού
- ΠΑΡΑΔΕΙΓΜΑ 13: Λίστα επιλογών
- ΠΑΡΑΔΕΙΓΜΑ 14: Μία τελευταία βελτίωση

ΑΠΑΙΤΟΥΜΕΝΑ ΥΛΙΚΑ

- Φορητός ή σταθερός υπολογιστής.
- Το περιβάλλον *Arduino IDE*. Τα δοθέν συμπληρωματικό υλικό θα πρέπει να είναι ήδη εγκατεστημένο και ρυθμισμένο.
- Αναπτυξιακή πλακέτα *Arduino UNO*.
- Καλώδιο *USB*.



TABLE OF CONTENTS

ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ	4
1. ΟΘΟΝΕΣ LCD	4
2. ΤΟ ΣΕΤ ΧΑΡΑΚΤΗΡΩΝ	6
3. ΟΙ ΓΡΑΦΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	7
4. Η ΒΙΒΛΙΟΘΗΚΗ LIQUIDCRYSTAL.H	9
5. Η ΜΝΗΜΗ ΔΕΔΟΜΕΝΩΝ ΕΕΡΡΟΜ	17
ΕΝΟΤΗΤΑ ΕΞΑΣΚΗΣΗΣ	19
6. ΣΥΝΔΕΟΝΤΑΣ ΤΗΝ ΟΘΟΝΗ LCD	19
7. ΠΑΡΑΔΕΙΓΜΑ 1: HELLO WORLD!	22
8. ΠΑΡΑΔΕΙΓΜΑ 2: Η ΟΘΟΝΗ	22
9. ΠΑΡΑΔΕΙΓΜΑ 3: Ο ΔΡΟΜΕΑΣ	22
10. ΠΑΡΑΔΕΙΓΜΑ 4: ΑΝΑΒΟΣΒΗΜΑ	22
11. ΠΑΡΑΔΕΙΓΜΑ 5: ΚΑΤΕΥΘΥΝΣΗ ΚΕΙΜΕΝΟΥ	23
12. ΠΑΡΑΔΕΙΓΜΑ 6: ΚΥΛΙΣΗ	23
13. ΠΑΡΑΔΕΙΓΜΑ 7: ΑΥΤΟΜΑΤΗ ΚΥΛΙΣΗ	24
14. ΠΑΡΑΔΕΙΓΜΑ 8: ΕΙΔΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ	25
15. ΠΑΡΑΔΕΙΓΜΑ 9: ΑΠΕΙΚΟΝΙΣΗ	25
16. ΠΑΡΑΔΕΙΓΜΑ 10: ΑΠΕΙΚΟΝΙΣΗ ΦΥΣΙΚΩΝ ΑΡΙΘΜΩΝ	26
17. ΠΑΡΑΔΕΙΓΜΑ 11: ΑΠΕΙΚΟΝΙΣΗ ΑΡΙΘΜΩΝ ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ	26
18. ΠΑΡΑΔΕΙΓΜΑ 12: ΜΕΝΟΥ	27
19. ΠΑΡΑΔΕΙΓΜΑ 13: ΛΙΣΤΑ ΕΠΙΛΟΓΩΝ	29
20. ΠΑΡΑΔΕΙΓΜΑ 14: ΜΙΑ ΤΕΛΕΥΤΑΙΑ ΒΕΛΤΙΩΣΗ	29
ΠΑΡΑΠΟΜΠΕΣ	31

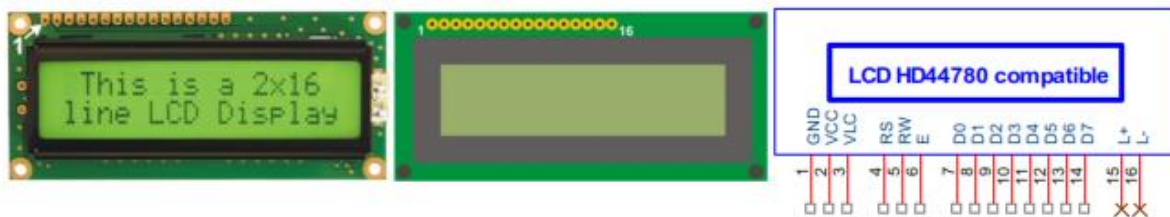
ΕΝΟΤΗΤΑ ΘΕΩΡΙΑΣ

1. ΟΘΟΝΕΣ LCD

Είναι ένα περιφερειακό εξόδο το οποίο, όχι μόνο απεικονίζει αριθμούς, αλλά επίσης και κάθε είδος χαρακτήρα, ή και απλά γραφικά. Έχουν χιλιάδες χρήσεις και λογικά, ήδη θα τις έχετε δει στην καθημερινότητά σας. Θα τις συναντήσετε με διαφορετικό αριθμό γραμμών ή διαφορετικό αριθμό χαρακτήρων ανά γραμμή. Συναντώνται επίσης με χρωματιστό οπίσθιο φωτισμό αλλά και με χαρακτηριστικές διαφορετικών χρωμάτων και μεγεθών.

Περιλαμβάνουν τον δικό τους ελεγκτή ο οποίος αναλαμβάνει όλες τις εσωτερικές τους διεργασίες. Οι περισσότερες από αυτές, βασίζονται στον δημοφιλή ελεγκτή HD44780 της Hitachi, γι' αυτό και δεν έχει ιδιαίτερη σημασία το ακριβές μέγεθος της οθόνης που χρησιμοποιείτε (2 x 16 ή 4 x 20).

Θα χρησιμοποιήσουμε μία οθόνη LCD 2 x 16. Παρακάτω (Σχήμα 1) αναπαρίσταται η μορφή της, καθώς και εμφανίζεται το ηλεκτρικό της διάγραμμα.

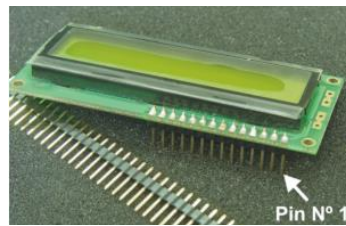


Σχήμα 1

Μία οθόνη LCD είναι ένα ψηφιακό περιφερειακό. Τα σήματά της μπορούν να συνδεθούν απευθείας με τους ακροδέκτες εισόδου και εξόδου του ελεγκτή. Χωρίζονται σε τρεις κατηγορίες:

- **Τροφοδοσία**: Οι ακροδέκτες αυτοί τροφοδοτούν την οθόνη με την απαιτούμενη για τη λειτουργία της ισχύ, ενώ η τάση συννηθέστερα είναι +5V και 0V (GND). Υπάρχει ένας ακόμα ακροδέκτης, ο VLC, ο οποίος παρέχει μία τάση κυμαινόμενη ανάμεσα σε 0V-5V και χρησιμοποιείται για να ρυθμίσουμε την αντίθεση της οθόνης.
- **Έλεγχος**: Αυτά είναι τα σήματα που καθορίζουν κατά πόσο η οθόνη λαμβάνει εντολές ή δεδομένα, εάν πρόκειται να διαβάσουμε ή να γράψουμε σε αυτή, καθώς και κατά πόσο είναι ενεργοποιημένη ή όχι.
- **Δεδομένα**: Ο ελεγκτής χρησιμοποιεί αυτούς τους ακροδέκτες για να αποστείλει δεδομένα και εντολές στην οθόνη.

Ο ακροδέκτης #1 είναι ο πρώτος στα αριστερά, όπως δείχνει και το (Σχήμα 2).



Σχήμα 2

Ο παρακάτω πίνακας περιγράφει τη λειτουργία των ακροδεκτών της οθόνης.

# ΑΚΡΔ.	ΟΝΟΜΑ	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ
1	Vss	Τροφοδοσία	Γείωση τροφοδοσίας (0V)
2	Vdd	Τροφοδοσία	Θετική πηγή τροφοδοσίας (+5V)
3	VLC	Τροφοδοσία	Ρύθμιση αντίθεσης: τάση που κυμαίνεται ανάμεσα σε 0V-5V.
4	RS	Είσοδος	Έξοδος από τον ελεγκτή. Η τιμή του καθορίζει τη λειτουργία της οθόνης, επιλέγοντας ανάμεσα σε δεδομένα ή εντολές. RS=0 - Αποστολή εντολών RS=1 - Αποστολή δεδομένων (ASCII τιμές)
5	R/W	Είσοδος	Έξοδος από τον ελεγκτή. Η τιμή του καθορίζει τη λειτουργία της οθόνης, καθορίζοντας κατά πόσο διαβάζουμε, ή γράφουμε σε αυτή: R/W=0 – Ο ελεγκτής γράφει στην οθόνη R/W=1 – Ο ελεγκτής διαβάζει από την οθόνη
6	E	Είσοδος	Έξοδος από τον ελεγκτή. Ενεργοποιεί ή απενεργοποιεί την οθόνη: E=0 – Οθόνη ανενεργή E=1 – Οθόνη ενεργή
7-14	DB0:DB7	Είσοδος / Έξοδος	Γραμμές σήματος για μεταφορά δεδομένων και εντολών. Το τι από τα δύο αποστέλλεται, καθορίζεται από τον ακροδέκτη RS. Τα σήματα DB0:DB7 χρησιμοποιούνται ως διεπαφή 8-bit Τα σήματα DB4:DB7 χρησιμοποιούνται ως διεπαφή 4-bit
15	L+	Τροφοδοσία	Θετική πηγή τροφοδοσίας για τον οπίσθιο φωτισμό (+5V)
16	L-	Τροφοδοσία	Γείωση για τον οπίσθιο φωτισμό (0V)

Δε θα χρησιμοποιήσουμε τους ακροδέκτες 15 και 16: είναι προαιρετικοί. Είναι πολύ πιθανό να μην υπάρχουν καν σε ορισμένα μοντέλα οθονών, καθώς αφορούν τον οπίσθιο φωτισμό, χαρακτηριστικό που δεν έχουν όλες οι οθόνες.

Οι οθόνες συγκαταλέγονται ανάμεσα στα πιο χρήσιμα και πολύπλευρα περιφερειακά. Όπως έχουμε αναφέρει ήδη, έχουν τη δυνατότητα απεικόνισης πολλών ειδών μηνυμάτων που αποτελούνται από κείμενο, αριθμούς και σύμβολα, ενώ έχουν την δυνατότητα να δημιουργήσουν αρκετά εφέ, όπως κίνηση από τα αριστερά προς τα δεξιά και αντίστροφα, τρεμόπαιγμα, κύλιση, κλπ. Μία οθόνη LCD έχει ξεχωριστό, δικό της ελεγκτή.



Η οθόνη LCD που θα χρησιμοποιήσουμε η οποία και περιλαμβάνεται στο kit προτεινόμενων υλικών παρουσιάζεται στο παραπάνω σχήμα. Η οθόνη αποτελείται από δύο σειρές των 16 χαρακτήρων εκάστη (2 x 16). Πιθανώς να πρέπει να κολλήσετε μία σειρά ακροδεκτών, πράγμα που θα σας διευκολύνει στο να την χρησιμοποιήσετε μαζί με την πλακέτα δοκιμών.

2. ΤΟ ΣΕΤ ΧΑΡΑΚΤΗΡΩΝ

Η επικοινωνία ανάμεσα στο Arduino και την οθόνη LCD επιτυγχάνεται μέσα από τους ακροδέκτες DB0 έως DB7. Μπορείτε να χρησιμοποιήσετε και τους οκτώ ακροδέκτες, ή, όπως στην περίπτωση μας, μόνο τους τέσσερις. Αυτό χαρακτηρίζεται ως «διεπαφή 4-bit».

Το Arduino στέλνει εντολές και δεδομένα στην οθόνη μέσα από αυτούς τους ακροδέκτες. Η οθόνη, στη συνέχεια, μπορεί να εκτελέσει ένα σύνολο διαφορετικών λειτουργιών: κύλιση, τρεμόπαιγμα, απαλοιφή χαρακτήρων, τοποθέτηση του δρομέα, κλπ. Μέσα από αυτούς τους ακροδέκτες στέλνονται επίσης οι ASCII κώδικες των χαρακτήρων που θέλετε να απεικονίσετε οι οποίοι έχουν μέγεθος 8-bit. Εάν λειτουργείτε την οθόνη με διεπαφή 8-bit, κάθε χαρακτήρας μεταφέρεται με μία μετάδοση. Η διεπαφή 4-bit, από την άλλη, χρησιμοποιεί δύο μεταδόσεις για κάθε χαρακτήρα. Είναι συνεπώς λίγο πιο αργή, αλλά χρησιμοποιεί λιγότερα καλώδια. Σε κάθε περίπτωση όμως, δε χρειάζεται να ανησυχείτε ιδιαίτερα γι' αυτό: οι συναρτήσεις που θα μελετήσουμε θα μας επιτρέψουν να χειριστούμε την οθόνη με ιδιαίτερη ευκολία.

Το παρακάτω σχήμα (Σχήμα 3) αναπαριστά ένα δείγμα από το σετ χαρακτήρων που έχουν τη δυνατότητα να απεικονίσουν οι οθόνες LCD. Το σετ χαρακτήρων ορίζεται από τον κατασκευαστή. Η εσωτερική μνήμη της οθόνης (ROM) περιέχει τον ορισμό των χαρακτήρων αυτών, και μπορεί να διαφέρει ανάμεσα σε διαφορετικά μοντέλα και εκδόσεις.

Τα τέσσερα λιγότερα σημαντικά bit, B3 έως B0, αναπαρίστανται στις γραμμές στα αριστερά του πίνακα. Τα τέσσερα πιο σημαντικά, από B7 έως B4, αναπαρίστανται στην κορυφή του πίνακα στις στήλες με δυαδικό κώδικα. Το μόνο που έχετε να κάνετε για να κωδικοποιήσετε ένα χαρακτήρα, είναι να τον επιλέξετε και να παρατηρήσετε σε ποια στήλη και γραμμή βρίσκεται. Ένα παράδειγμα: ο χαρακτήρας 'F' είναι στη στήλη 4 (0100) και στη γραμμή 6 (0110). Ο δυαδικός του κώδικας, ως εκ τούτου, είναι 0100 0110 (0x46) ο οποίος και ανταποκρίνεται στον ASCII κώδικα για τον συγκεκριμένο χαρακτήρα. Αναφέραμε προηγουμένως πως δε χρησιμοποιούν όλες οι οθόνες το ίδιο σετ χαρακτήρων αλλά εξαρτάται από τον κατασκευαστή, το μοντέλο, κλπ. Παρ' όλα αυτά, οι κώδικες για τους «στάνταρ» ASCII χαρακτήρες, είναι κοινói σε όλες τις οθόνες. Αυτοί είναι στις στήλες 2 (0010) έως και 7 (0111).

CG RAM Address	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)			0@P`P									-9εαp			
0001	(2)			!1AQa9									α74äq			
0010	(3)			"2BRbr									「イツ×pθ			
0011	(4)			#3CScs									」ウテεεω			
0100	(5)			\$4DTdt									、イトμΩ			
0101	(6)			%5EUeu									・オナ1εÜ			
0110	(7)			&6FUfv									ヲカニヨρΣ			
0111	(8)			'7GW9w									フキヌラgπ			
1000	(1)			<8HXhx									イフネリJε			
1001	(2)			>9IYiy									ウケル'y			
1010	(3)			*:JZjz									エコハレjキ			
1011	(4)			+;K[k<									オサヒロ*片			
1100	(5)			,<L¥ll									ホシフワφ円			
1101	(6)			-=M]m>									ユスへンも÷			
1110	(7)			.>N^n→									ヨセホ`ñ			
1111	(8)			/?O_ο←									ッソマ°ö			

Figure 3

3. ΟΙ ΓΡΑΦΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ

Μπορείτε να δημιουργήσετε έως και οκτώ γραφικούς χαρακτήρες των 5 x 8 σημείων ή "pixel". Κάθε χαρακτήρας αριθμείται από το 0 έως το 7 και απαιτεί συνολικά 8 bytes για να οριστεί. Η οθόνη LCD έχει μία εσωτερική μνήμη RAM που ονομάζεται CGRAM γι' αυτή τη χρήση. Άπαξ και έχουν οριστεί, μπορείτε να τους απεικονίσετε αποστέλλοντας τον αριθμό που αντιστοιχεί στον συγκεκριμένο χαρακτήρα (0 έως



7). Μην ξεχνάτε: οι γραφικοί χαρακτήρες αποθηκεύονται στη μνήμη RAM. Εάν το σύστημα σταματήσει να τροφοδοτείται με ενέργεια, χάνονται, ενώ το Arduino θα πρέπει να του ορίσει ξανά.

Οι γραφικοί χαρακτήρες ορίζονται εισάγοντας bytes σε συνεχείς θέσεις στη μνήμη CGRAM. Το δυαδικό μοτίβο των bytes αυτών καθορίζει και κάθε χαρακτήρα. Η CGRAM είναι μία πτητική μνήμη ικανή να αποθηκεύσει συνολικά 64 bytes. Ένας χαρακτήρας 5 x 8 σημείων απαιτεί 8 bytes για τον ορισμό του. Οπότε μπορείτε να ορίσετε έως και οκτώ διαφορετικούς χαρακτήρες (8 x 8) με όποιο τρόπο επιθυμείτε.

Ο παρακάτω πίνακας (Σχήμα 4) περιέχει ένα παράδειγμα με τέσσερις χαρακτήρες 5 x 8. Οι χαρακτήρες έχουν εισαχθεί στις 32 πρώτες θέσεις της μνήμης CGRAM. Ο πρώτος τοποθετήθηκε στις θέσεις 0 έως 7, ο δεύτερος στις θέσεις 8 έως 15, και ούτε καθεξής.

Κάθε bit με τιμή «1» ενεργοποιεί το αντίστοιχο pixel στην οθόνη LCD. Ο πρώτος γραφικός χαρακτήρας της CGRAM απεικονίζεται στέλνοντας τον αριθμό «0» σα να ήταν μέρος του ASCII κώδικα. Ο δεύτερος απεικονίζεται στέλνοντας τον αριθμό «1», και ούτε καθεξής.

Το μόνο που έχετε να κάνετε για να ορίσετε τους χαρακτήρες είναι να δημιουργήσετε έναν πίνακα για κάθε χαρακτήρα. Έπειτα, τα περιεχόμενα του κάθε πίνακα αντιγράφονται στη CGRAM της οθόνης χρησιμοποιώντας τη συνάρτηση **createChar()** – την οποία και θα μελετήσουμε παρακάτω. Πρέπει να δημιουργήσετε έναν πίνακα για κάθε χαρακτήρα που επιθυμείτε να ορίσετε- σε αυτό το παράδειγμα έχουμε τέσσερις.

1. byte heart[8] = {10,21,17,17,17,10,4,0};

2. byte smile[8] = {B0,B01010,B0,B00100,B00000,B10001,B01110,B0};

3. byte letter_ñ [8] = {31,0,24,27,17,17,17,0};

4. byte letter_ú [8] = {2,4,17,17,17,19,13,0};

CHARACTER	Bits on CGRAM Memory	Decimal	CGRAM Address	Character Code
0	0 0 0 0 1 0 1 0	10	0	0
	0 0 0 1 0 1 0 1	21	1	
	0 0 0 1 0 0 0 1	17	2	
	0 0 0 1 0 0 0 1	17	3	
	0 0 0 1 0 0 0 1	17	4	
	0 0 0 0 1 0 1 0	10	5	
	0 0 0 0 0 1 0 0	4	6	
	0 0 0 0 0 0 0 0	0	7	
1	0 0 0 0 0 0 0 0	0	8	1
	0 0 0 0 1 0 1 0	10	9	
	0 0 0 0 0 0 0 0	0	10	
	0 0 0 0 0 1 0 0	4	11	
	0 0 0 0 0 0 0 0	0	12	
	0 0 0 1 0 0 0 1	17	13	
	0 0 0 0 1 1 1 0	14	14	
	0 0 0 0 0 0 0 0	0	15	
2	0 0 0 1 1 1 1 1	31	16	2
	0 0 0 0 0 0 0 0	0	17	
	0 0 0 1 0 1 1 0	22	18	
	0 0 0 1 1 0 0 1	25	19	
	0 0 0 1 0 0 0 1	17	20	
	0 0 0 1 0 0 0 1	17	21	
	0 0 0 1 0 0 0 1	17	22	
	0 0 0 0 0 0 0 0	0	23	
3	0 0 0 0 0 0 1 0	2	24	3
	0 0 0 0 0 1 0 0	4	25	
	0 0 0 1 0 0 0 1	17	26	
	0 0 0 1 0 0 0 1	17	27	
	0 0 0 1 0 0 0 1	17	28	
	0 0 0 1 0 0 1 1	19	29	
	0 0 0 0 1 1 0 1	13	30	
	0 0 0 0 0 0 0 0	0	31	

Figure 4

4. Η ΒΙΒΛΙΟΘΗΚΗ LIQUIDCRYSTAL.H

Έως τώρα πιθανώς να έχετε εξοικειωθεί με την έννοια της «βιβλιοθήκης» και με το πως να διαχειριστείτε το εύρος των συναρτήσεων που εμπεριέχει. Το Arduino συνοδεύεται από έναν ευρύ αριθμό βιβλιοθηκών, αλλά εδώ θα περιοριστούμε στην “LiquidCrystal.h” η οποία περιέχει συναρτήσεις κατάλληλες για τον έλεγχο και τη διαχείριση οθονών LCD βασισμένων στον ελεγκτή HD44780 της Hitachi (ή αντίστοιχους).

Θα μελετήσουμε τις πιο χαρακτηριστικές και σημαντικές συναρτήσεις. Γενικότερα, προτείνεται να επισκεφτείτε www.arduino.cc όπου και θα βρείτε πληθώρα πληροφοριών και παραδειγμάτων για όλες.

Συνάρτηση: LiquidCrystal()

Η συνάρτηση αυτή κατασκευάζει ένα αντικείμενο τύπου “LiquidCrystal” και εγκαθιδρύει τη σύνδεση μεταξύ του Arduino και της οθόνης LCD. Μπορούμε να επιλέξουμε ανάμεσα σε διεπαφή 4- ή 8-bit. Εάν επιλέξετε 4-bit διεπαφή, μην χρησιμοποιήσετε τα σήματα D0 έως D3. Μπορείτε επίσης να επιλέξετε εάν θα αξιοποιήσετε το σήμα R/W, ή αν θα το κρατήσετε γειωμένο, όπως στο παράδειγμα που θα εξετάσουμε.



Σύνταξη:

```
LiquidCrystal var(RS,E,D4,D5,D6,D7); // Για μία διεπαφή 4 bit χωρίς το σήμα R/W  
LiquidCrystal var(RS,RW,E,D4,D5,D6,D7); // Για μία διεπαφή 4 bit με το σήμα R/W  
LiquidCrystal var(RS,E,D0,D1,D2,D3,D4,D5,D6,D7); // Για μία διεπαφή 8 bit χωρίς το σήμα R/W  
LiquidCrystal var(RS,RW,E,D0,D1,D2,D3,D4,D5,D6,D7); // Για μία διεπαφή 8 bit με το σήμα R/W
```

var: το όνομα της μεταβλητής που έχετε αναθέσει στην οθόνη LCD που θέλετε να ελέγξετε.

RS: ο ακροδέκτης του Arduino που έχει συνδεθεί με τον ακροδέκτη RS της οθόνης.

RW: ο ακροδέκτης του Arduino που έχει συνδεθεί με τον ακροδέκτη R/W της οθόνης (εάν πρόκειται να τον χρησιμοποιήσουμε).

E: ο ακροδέκτης του Arduino που έχει συνδεθεί με τον ακροδέκτη E της οθόνης.

D0-D7: οι ακροδέκτες του Arduino που έχουν συνδεθεί με τα σήματα DB0-DB7 της οθόνης. Εάν δεν έχουν υποδειχθεί ακροδέκτες για τα σήματα DB0-DB3, υποθέτουμε διεπαφή 4-bit interface και αξιοποιούμε μόνο τα σήματα DB4-DB7.

Παράδειγμα:

```
LiquidCrystal lcd(7,8,9,10,11,12); //Εγκαθιδρύει τη σύνδεση μίας οθόνης που ονομάζουμε "lcd".  
// Οι ακροδέκτες του Arduino D7 έως D12 συνδέονται με τους  
// ακροδέκτες RS, E, DB4, DB5, DB6 και DB7 της οθόνης. Το  
// σήμα R/W συνδέεται στη γείωση (GND).
```

- **Συνάρτηση: Begin()**

Η συνάρτηση αυτή εκκινεί τη λειτουργία της οθόνης και αναθέτει σε αυτή τον αριθμό των σειρών και τον αριθμό των χαρακτήρων ανά γραμμή σύμφωνα με το μοντέλο που έχουμε. Στην περίπτωση μας, πρόκειται να χρησιμοποιήσουμε μία οθόνη 16 x 2.

Σύνταξη:

```
var.begin(c,f);
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LiquidCrystal().

c: ο αριθμός των στηλών.



f: ο αριθμός των σειρών.

Παράδειγμα:

LiquidCrystal lcd(7,8,9,10,11,12); // Συνδεσμολογία της οθόνης

lcd.begin(16,2); // Πρόκειται για μία οθόνη 16 x 2

- **Συνάρτηση: setCursor()**

Η συνάρτηση αυτή τοποθετεί το δρομέα στο σημείο που επιθυμούμε. Οι χαρακτήρες που εγγράφουμε θα εμφανιστούν στην τρέχουσα θέση του δρομέα.

Σύνταξη:

```
var.setCursor(c,f);
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LiquidCrystal().

c: ο αριθμός των στηλών (ξεκινώντας από το 0).

f: ο αριθμός των γραμμών (ξεκινώντας από το 0).

Παράδειγμα:

LiquidCrystal lcd(7,8,9,10,11,12); // Συνδεσμολογία της οθόνης

lcd.setCursor(3,0); // Τοποθετεί τον δρομέα στη θέση 3 (4^{ος} χαρακτήρας)

// της γραμμής 0 (η 1^η γραμμή).

- **Συνάρτηση: home()**

Η συνάρτηση αυτή τοποθετεί το δρομέα στη θέση 0,0 της οθόνης (επάνω αριστερά). Δεν απαλείφει ότι είχε απεικονιστεί μέχρι πριν.

Σύνταξη:

```
var.home();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LiquidCrystal().

Συνάρτηση: clear()

Η συνάρτηση αυτή απαλείφει ότι είχε απεικονιστεί πριν στην οθόνη και τοποθετεί τον δρομέα στη θέση 0,0 της οθόνης (επάνω αριστερά).



Σύνταξη:

```
var.clear();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.

Παράδειγμα:

```
LiquidCrystal lcd(7,8,9,10,11,12); // Συνδεσμολογία της οθόνης.
```

```
lcd.clear(); // Καθαρισμός της οθόνης.
```

- **Συνάρτηση: write()**

Η συνάρτηση αυτή εγγράφει έναν χαρακτήρα στην τρέχουσα θέση του δρομέα.

Σύνταξη:

```
var.write(char);
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.

char: Ο χαρακτήρας που επιθυμούμε να εγγράψουμε.

Παράδειγμα:

```
lcd.write('A'); // Εγγράφει το 'A'
```

- **Συνάρτηση: print()**

Αυτή η συνάρτηση εκτυπώνει στην LCD οθόνη ξεκινώντας από την τρέχουσα θέση του δρομέα.

Σύνταξη:

```
var.print(data);
```

```
var.print(data,base);
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.



data: Τα δεδομένα που θέλουμε να εκτυπώσουμε. Αυτά μπορούν να είναι char, int, long, float ή string.

base: Προαιρετικό. Καταδεικνύει την επιθυμητή αριθμητική βάση: BIN=Δυαδική; DEC=Δεκαδική (προεπιλεγμένο); OCT=Οκταδική; HEX=Δεκαεξαδική; ή N=n^ο δεκαδικά ψηφία για αριθμούς κινητής υποδιαστολής (με προεπιλεγμένο το 2).

Παραδείγματα:

int A=19;

float PI=3.1416;

lcd.print(A,HEX); // Εκτυπώνει το A σε δεξαεξαδικό (1910 = 1316)

lcd.print("Hello"); // Εκτυπώνει "Hello"

lcd.print(PI*2,4); // Εκτυπώνει 6.2832 (τέσσερα δεκαδικά)

- **Συνάρτηση: cursor()**

Η συνάρτηση αυτή απεικονίζει το δρομέα πάνω στην LCD οθόνη στην τρέχουσα θέση του ως μία κάτω παύλα (_). Αυτό είναι και το σημείο στο οποίο θα εγγραφεί τυχόν επόμενος χαρακτήρας.

Σύνταξη:

var.cursor();

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LiquidCrystal().

- **Συνάρτηση: noCursor()**

Κρύβει το δρομέα.

Σύνταξη:

var.noCursor();

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LiquidCrystal().



Συνάρτηση: **blink()**

Η συνάρτηση αυτή απεικονίζει το δρομέα πάνω στην LCD οθόνη στην τρέχουσα θέση του σαν ένα συμπαγές διακοπτόμενο σύμβολο (█). Αυτό είναι και το σημείο στο οποίο θα εγγραφεί τυχόν επόμενος χαρακτήρας.

Σύνταξη:

```
var.blink();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

Συνάρτηση: **noBlink()**

Κρύβει το συμπαγή διακοπτόμενο δρομέα (█).

Σύνταξη:

```
var.noBlink();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

- **Συνάρτηση: noDisplay()**

Αυτή η συνάρτηση απενεργοποιεί την LCD οθόνη, χωρίς όμως να χάνονται τυχόν περιεχόμενα που έχουν απεικονιστεί σε αυτήν, ούτε η θέση του δρομέα.

Σύνταξη:

```
var.noDisplay();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

- **Συνάρτηση: display()**

Ενεργοποιεί την LCD οθόνη και ανακτά τα περιεχόμενά της πριν την κλήση noDisplay().

Σύνταξη:

```
var.display();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

Συνάρτηση: **scrollDisplayLeft()**



Αυτή η συνάρτηση μετακινεί το περιεχόμενο (τυχόν κείμενο, αλλά και το δρομέα) της οθόνης μία θέση προς τα αριστερά.

Σύνταξη:

```
var.scrollDisplayLeft();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

- **Function: scrollDisplayRight()**

Αυτή η συνάρτηση μετακινεί το περιεχόμενο (τυχόν κείμενο, αλλά και το δρομέα) της οθόνης μία θέση προς τα δεξιά.

Σύνταξη:

```
var.scrollDisplayRight();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

- **Συνάρτηση: leftToRight()**

Αυτή η συνάρτηση ορίζει την κατεύθυνση προς την οποία εγγράφονται νέοι χαρακτήρες από το δρομέα: από αριστερά προς τα δεξιά. Δεν επηρεάζει τους ήδη εγγεγραμμένους χαρακτήρες.

Σύνταξη:

```
var.LeftToRight();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().

Συνάρτηση: rightToLeft()

Αυτή η συνάρτηση ορίζει την κατεύθυνση προς την οποία εγγράφονται νέοι χαρακτήρες από το δρομέα: από δεξιά προς τα αριστερά. Δεν επηρεάζει τους ήδη εγγεγραμμένους χαρακτήρες.

Σύνταξη:

```
var.RightToLeft();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την LyquidCrystal().



Συνάρτηση: `autoscroll()`

Η συνάρτηση αυτή ενεργοποιεί την κύλιση (ή αλλιώς, την αυτόματη μετακίνηση των περιεχομένων της οθόνης). Κάθε φορά που ένας χαρακτήρας αποστέλλεται στην οθόνη, η συνάρτηση τον εμφανίζει και στη συνέχεια μετακινεί τα υπόλοιπα περιεχόμενα κατά μία θέση. Εάν η κατεύθυνση έχει οριστεί ως από αριστερά προς τα δεξιά (με τη συνάρτηση `leftToRight()`), τα περιεχόμενα μετακινούνται προς τα αριστερά. Εάν η κατεύθυνση έχει οριστεί ως από δεξιά προς τα αριστερά (με τη συνάρτηση `rightToLeft()`), τα περιεχόμενα μετακινούνται προς τα δεξιά.

Σύνταξη:

```
var.autoscroll();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.

Συνάρτηση: `noAutoscroll()`

Αυτή η συνάρτηση απενεργοποιεί την αυτόματη κύλιση των περιεχομένων της οθόνης.

Σύνταξη:

```
var.noAutoscroll();
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.

- **Συνάρτηση: `createChar()`**

Αυτή η συνάρτηση δημιουργεί ένα χαρακτήρα ορισμένο από τον χρήστη. Είναι ικανή να δημιουργήσει έως και οκτώ χαρακτήρες μεγέθους 5 x 8 pixel, αριθμημένους από το 0 έως και το 7. Ένας πίνακας των 8 bytes, ένα για κάθε γραμμή, καθορίζει την εμφάνιση (ή το σχέδιο) του χαρακτήρα. Τα πέντε λιγότερο σημαντικά bit του κάθε byte αντιστοιχούν στα 5 pixel που σχηματίζουν κάθε γραμμή του χαρακτήρα. Άπαξ και η συνάρτηση έχει δημιουργήσει ένα χαρακτήρα, μπορείτε να τον απεικονίσετε στην οθόνη υποδεικνύοντας τον αριθμό του.

Σύνταξη:

```
var.createChar(n,data);
```

var: αυτό είναι το όνομα του αντικειμένου της οθόνης που δημιουργήσαμε με την `LyquidCrystal()`.

n: αντιπροσωπεύει τον αριθμό του χαρακτήρα που ορίζεται (από το 0 έως και το 7) .

data: το όνομα του πίνακα που περιέχει τα bytes που ορίζουν τον ειδικό χαρακτήρα.



Παράδειγμα:

```
bytearrowhead[8]={B00100,B01110,B10101,B00100,B00100,B00100, B00100, B00100};  
// Δημιουργεί τον μεγέθους 8-byte πίνακα "arrowhead" ο οποίος και ορίζει το γραφικό χαρακτήρα.  
  
lcd.createChar(0, arrowhead);  
  
// Δημιουργεί τον νέο χαρακτήρα (με νούμερο 0) από το περιεχόμενο του πίνακα "arrowhead"  
  
lcd.write(byte(0));  
  
// Απεικονίζει τον νέο χαρακτήρα (με νούμερο 0) που δημιουργήσαμε.
```

5. Η ΜΝΗΜΗ ΔΕΔΟΜΕΝΩΝ EEPROM

Εάν πιστεύετε πως τα παρακάτω δεν έχουν ιδιαίτερη συνάφεια με το κεφάλαιο που εξετάζουμε, θα έχετε απόλυτο δίκιο: δεν έχουν. Τυχαίνει όμως να είναι μία καλή ευκαιρία να μιλήσουμε για τη μνήμη δεδομένων EEPROM του ελεγκτή Arduino UNO. Θα σας βοηθήσει να βελτιώσετε πολλά από τα τρέχοντα και μελλοντικά σας project.

Οι ελεγκτές που απαρτίζουν την πλατφόρμα του Arduino, περιλαμβάνουν και έναν ειδικό τύπο μνήμης: τη μνήμη EEPROM. Αυτή σας επιτρέπει να διαβάσετε ή να γράψετε δεδομένα των 8-bit, ή αλλιώς, τιμές από το 0 έως και το 255. Έχει όμως και ένα ειδικό χαρακτηριστικό: συγκρατεί όλη την πληροφορία που έχετε αποθηκεύσει ακόμα και αν διακοπεί η τροφοδοσία της. Δεν διαγράφονται τα δεδομένα.

Γι' αυτό το λόγο είναι ιδανική για την αποθήκευση μη-πτητικής πληροφορίας που μπορεί να τροποποιηθεί, όπως κωδικούς πρόσβασης, παραμέτρους λειτουργίας, αριθμούς τηλεφώνων, κωδικούς, κλπ. Όπως θα γνωρίζετε ήδη, το Arduino NANO περιλαμβάνει τον ελεγκτή ATMEGA328 ο οποίος και περιέχει 1KB (1024 byte) μνήμης EEPROM.

Έως τώρα γνωρίζετε αρκετά για να μπορέσετε να τη χρησιμοποιήσετε. Σας παρέχεται επίσης μία βιβλιοθήκη, η "EEPROM.h". Περιέχει μόνο δύο συναρτήσεις, τις οποίες μπορείτε να χρησιμοποιήσετε για να διαβάσετε ή να γράψετε δεδομένα στη μνήμη. Ως συνήθως, θα πρέπει να περιλάβετε τη βιβλιοθήκη στο πρόγραμμά σας, χρησιμοποιώντας την οδηγία **#include <EEPROM.h>**.



- **Συνάρτηση:read()**

Αυτή η συνάρτηση διαβάζει την 8-bit πληροφορία που βρίσκεται στη μνήμη EEPROM.

Σύνταξη:

```
EEPROM.read(dir);
```

dir: Καθορίζει τη θέση από την οποία θα διαβάσετε. Η μνήμη EEPROM αποτελείται από 1024 διακριτές θέσεις, συνεπώς, το δυνατό εύρος είναι από 0 έως και 1023.

Παράδειγμα:

```
#include EEPROM.h           // Περιλαμβάνει τη βιβλιοθήκη.  
  
byte value;                 // Μεταβλητή 8-bit.  
  
value = EEPROM.read(279);   // Διαβάζει το byte στη θέση 279.
```

- **Συνάρτηση:write()**

Αυτή η συνάρτηση εγγράφει μία τιμή 8-bit (από 0 έως και 255) σε μία από τις διαθέσιμες θέσεις της μνήμης EEPROM.

Σύνταξη:

```
EEPROM.write(dir, value);
```

dir: Καθορίζει τη θέση από την οποία θα διαβάσετε. Η μνήμη EEPROM αποτελείται από 1024 διακριτές θέσεις, συνεπώς, το δυνατό εύρος είναι από 0 έως και 1023.

value: Αυτή είναι η τιμή 8-bit που θέλετε να εγγράψετε στην υποδειχθείσα θέση.

Παράδειγμα:

```
EEPROM.write(982, 33);           // Εγγράφει την τιμή 33 στη θέση 982.
```

ΣΗΜΑΝΤΙΚΟ: Η μνήμη EEPROM έχει περιορισμένη «αντοχή» η οποία ανέρχεται σε 100,000 κύκλους εγγραφής/σβησίματος. Μία εγγραφή στην EEPROM μπορεί να χρειαστεί έως και 3.3 mS (μιλιδευτερόλεπτα) για να ολοκληρωθεί, οπότε θα πρέπει να προσέχετε πόσο συχνά γράφετε σε αυτή, καθώς αν υπερβείτε το όριο μπορεί να απενεργοποιηθεί. Κρατήστε το στο μυαλό σας.

ΕΝΟΤΗΤΑ ΕΞΑΣΚΗΣΗΣ

1. ΣΥΝΔΕΟΝΤΑΣ ΤΗΝ ΟΘΟΝΗ LCD

Θα ξεκινήσουμε την ενότητα εξάσκησης συνδέοντας την LCD οθόνη με το Arduino. Παρατηρήστε προσεκτικά τη συνδεσμολογία και βεβαιωθείτε πως την έχετε υλοποιήσει σωστά. Παρουσιάζεται το σχηματικό διάγραμμα:

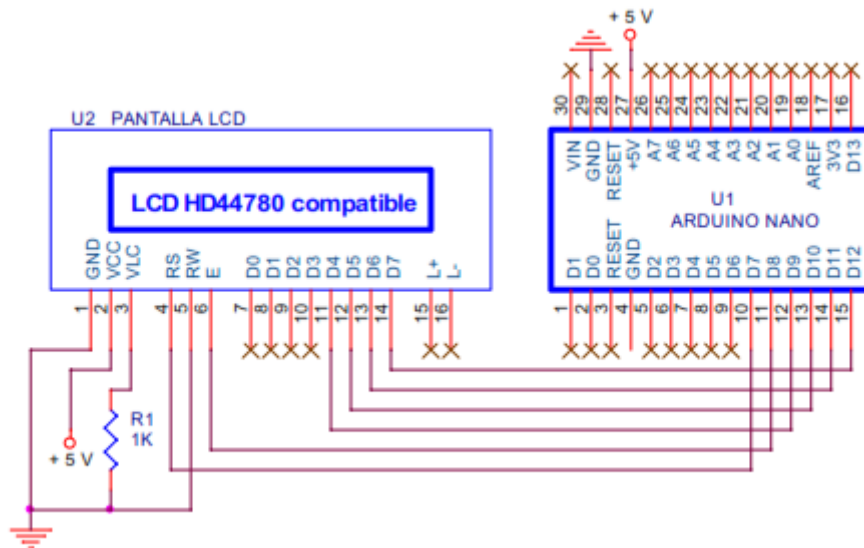


Figure 5

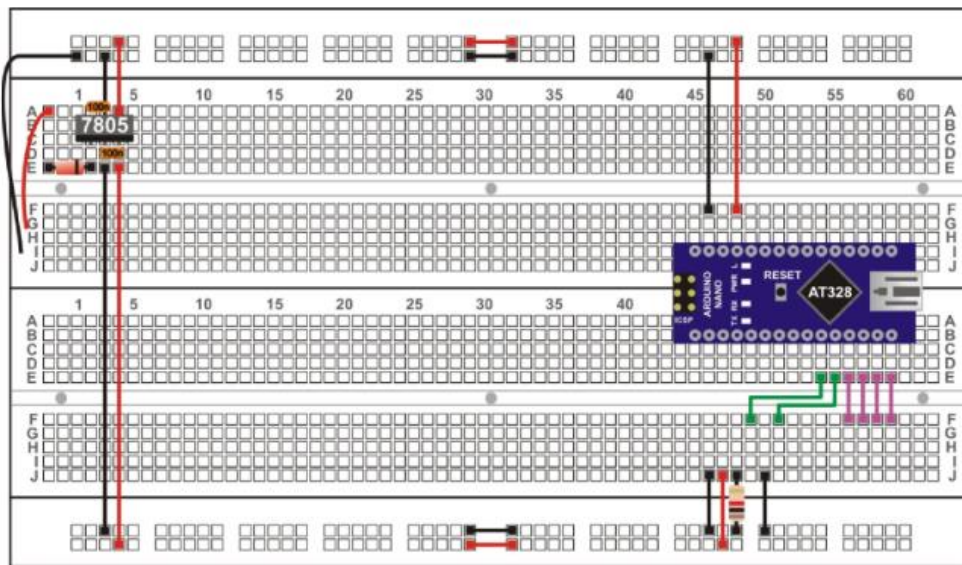
Οι ακροδέκτες D9 έως D12 του Arduino συνδέονται με τους ακροδέκτες DB4 έως DB7 της οθόνης. Ελεγχόμενο από το πρόγραμμά σας, το Arduino θα στέλνει τις εντολές και τα δεδομένα που πρόκειται να απεικονιστούν στην οθόνη. Τα D7 και D8 ελέγχουν τα σήματα RS και E, ενώ το σήμα R/W πρέπει να γειωθεί (σύνδεση στα 0V).

Η οθόνη τροφοδοτείται με +5V από τους ακροδέκτες VCC (τροφοδοσία) και GND (γείωση). Ταυτόχρονα δίνουμε μία κυμαινόμενη τάση ανάμεσα σε 0V και 5V στον ακροδέκτη VLC, με τον αριθμό 3, για να ρυθμίσουμε την αντίθεση. Μπορούμε επίσης να τον συνδέσουμε απλά στη γη, μέσω της R1, μίας αντίστασης. Η τιμή της αντίστασης μπορεί να διαφέρει ανάλογα με το μοντέλο της οθόνης.

Μιας και είναι πολύ δύσκολο να γνωρίζουμε τι τύπο οθόνης έχουμε στα χέρια μας, θα ξεκινήσουμε με μία αντίσταση στο 1KΩ. Εάν με αυτή την τιμή οι χαρακτήρες είναι δυσδιάκριτοι, μπορείτε να αλλάξετε την αντίσταση με μία μεγαλύτερη, για παράδειγμα, στα 4,7KΩ. Πιθανότατα να πετύχετε την ιδανική αντίσταση κάπου ανάμεσα σ' αυτές τις δύο τιμές.

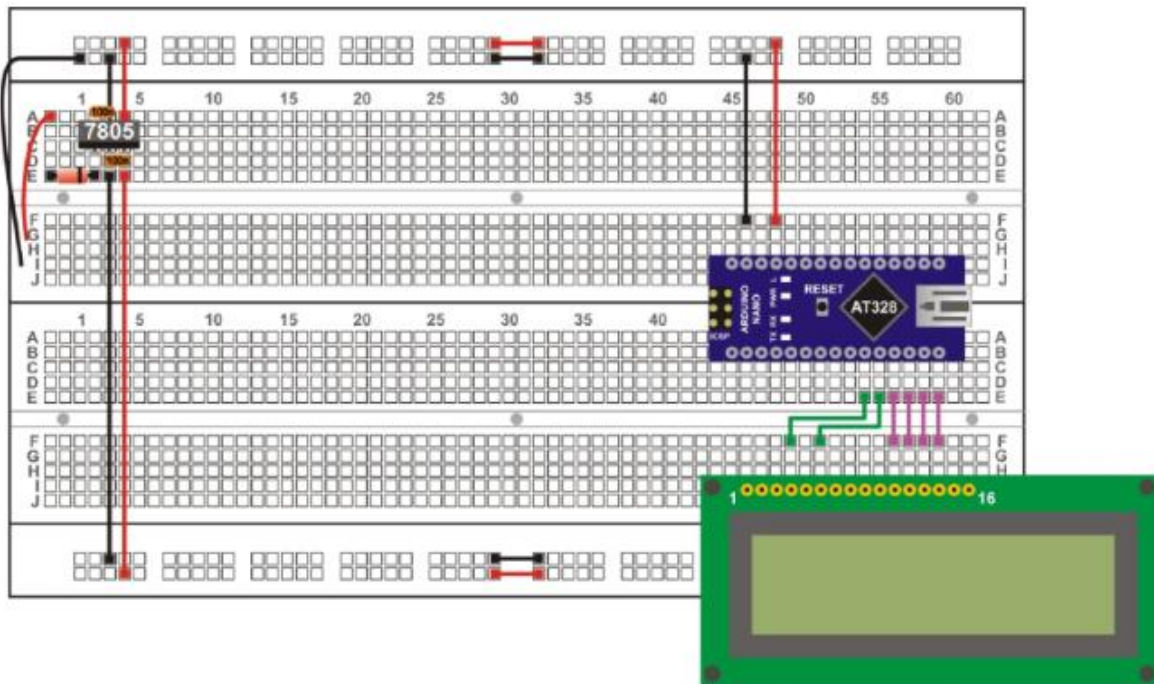
Χρησιμοποιούμε τους ακροδέκτες 15 και 16 της οθόνης για ρυθμίσουμε τον οπίσθιο φωτισμό, αν και δεν έχουν όλα τα μοντέλα αυτή τη δυνατότητα. Εδώ, δε θα τους χρησιμοποιήσουμε, ενώ είναι πολύ πιθανό η οθόνη που έχετε στα χέρια σας να μην έχει καν αυτούς τους δύο ακροδέκτες. Εάν τους έχει, μη τους συνδέσετε.

Συστήνουμε να ξεκινήσετε τη συνδεσμολογία με τα καλώδια όπως στο σχήμα παρακάτω. Θα τοποθετήσετε αργότερα την οθόνη και μερικά καλώδια μπορεί να καλυφθούν. Παρατηρήστε το σχήμα.



Σχήμα 6

Τώρα τοποθετήστε την οθόνη LCD. Βεβαιωθείτε πως ο ακροδέκτης 1 βρίσκεται τοποθετημένος στην ίδια στήλη από οπές με το μαύρο καλώδιο της γείωσης, ο ακροδέκτης 2 στην ίδια στήλη με το κόκκινο καλώδιο τροφοδοσίας +5V, ο ακροδέκτης 3 στην ίδια στήλη με την αντίσταση, και ούτε καθεξής.



Σχήμα 7

Πιθανώς να σκεφτείτε κάποια διαφορετική προσέγγιση για τη σύνδεση της οθόνης, η οποία ίσως και να είναι καλύτερη. Θα πρέπει να θυμάστε, παρόλα αυτά, πως η οθόνη LCD θα παραμείνει συνδεδεμένη για σχεδόν ολόκληρη την υπόλοιπη ενότητα. Γι' αυτό το λόγο θα πρέπει να μεριμνήσετε να αφήσετε αρκετό χώρο στο ράστερ για τα υπόλοιπα εξαρτήματα και περιφερειακά που θα χρησιμοποιήσουμε.



2. ΠΑΡΑΔΕΙΓΜΑ 1: Hello world!

Αυτό είναι το πρώτο πρόγραμμα που θα χρησιμοποιήσουμε για να εμφανίσουμε το διάσημο μήνυμα στην LCD οθόνη.

Μπορείτε να περιλάβετε τη σχετική βιβλιοθήκη χρησιμοποιώντας την **#include**. Μπορείτε να εγκαθιδρύσετε το αντικείμενο `lcd` και να ορίσετε τη συνδεσμολογία σας χρησιμοποιώντας την **LiquidCrystal()**. Βεβαιωθείτε πως η συνδεσμολογία που ορίζετε στον κώδικά σας ανταποκρίνεται σε αυτή που αντιστοιχεί στο σχηματικό διάγραμμα που υλοποιήσατε προηγουμένως.

Ορίστε ως μοντέλο οθόνης το 16 x 2 χρησιμοποιώντας τη συνάρτηση **lcd.begin(16,2)** μέσα στην **setup()**.

Τέλος, στο κύριο σώμα του προγράμματος, στη **loop()**, μεταδώστε το μήνυμα που θέλετε να εμφανίσετε χρησιμοποιώντας τη συνάρτηση **lcd.print()**. Η εντολή εμφάνισης του μηνύματος πραγματοποιείται μέσα σε έναν ατέρμονα βρόγχο που ονομάζεται **while(1)** ο οποίος δε κάνει κάτι πέρα από αυτό. Στην ουσία, δε θα είχε καμία διαφορά εάν εμφανίζαμε το μήνυμα στο τέλος της εκτέλεσης.

3. ΠΑΡΑΔΕΙΓΜΑ 2: Η οθόνη

Προτείνουμε να δοκιμάσετε να εκτελέσετε τα επόμενα παραδείγματα για να εξοικειωθείτε με τις συναρτήσεις που περιλαμβάνει η βιβλιοθήκη “**LiquidCrystal.h**” και με τα διάφορα εφέ που αυτές δημιουργούν.

Πρόκειται να χρησιμοποιήσετε τις συναρτήσεις **noDisplay()** και **display()** για να ενεργοποιήσετε ή να απενεργοποιήσετε την οθόνη: η οθόνη θα εμφανίσει οτιδήποτε περιεχόμενο έχει, ή θα «αδειάσει», αντίστοιχα.


4. ΠΑΡΑΔΕΙΓΜΑ 3: Ο δρομέας

Ο δρομέας καταδεικνύει το σημείο πάνω στην οθόνη στο οποίο θα εμφανιστεί ο επόμενος χαρακτήρας που θα ληφθεί. Αυτό το παράδειγμα χρησιμοποιεί τις συναρτήσεις **cursor()** και **noCursor()** για να ενεργοποιήσει, ή να απενεργοποιήσει, το σύμβολο του δρομέα.

Θεωρητικά, ο δρομέας θα πρέπει να εμφανίζεται σαν μία κάτω παύλα (`_`).

5. ΠΑΡΑΔΕΙΓΜΑ 4: Αναβόσβημα



Ο δρομέας μπορεί επίσης να φαίνεται σαν ένα συμπαγές διακοπτόμενο σύμβολο () το οποίο καταδεικνύει το σημείο πάνω στην οθόνη στο οποίο θα εμφανιστεί ο επόμενος χαρακτήρας που θα ληφθεί. Γι' αυτό χρησιμοποιούνται οι συναρτήσεις **blink()** και **noBlink()**.

Ο δρομέας αναβοσβήνει αυτόνομα, ελεγχόμενος από την LCD οθόνη. Στο παράδειγμα αυτό, ο δρομέας είναι ενεργός και αφήνεται να αναβοσβήσει για τρία δευτερόλεπτα πριν απενεργοποιηθεί για άλλα τρία, έτσι ώστε να κατανοήσετε καλύτερα αυτές τις συναρτήσεις.

6. ΠΑΡΑΔΕΙΓΜΑ 5: Κατεύθυνση κειμένου

Χρησιμοποιώντας τις συναρτήσεις `leftToRight()` και `rightToLeft()` μπορείτε να καθορίσετε πως θα εμφανίζεται στην οθόνη το κείμενο που γράφετε. Μπορείτε να αναπαραστήσετε πληροφορία από τα αριστερά προς τα δεξιά, αλλά και από τα δεξιά προς τα αριστερά. Από προεπιλογή, ο δρομέας γράφει από αριστερά προς τα δεξιά, όπως ακριβώς γράφουμε και με το χέρι.

Το παρακάτω παράδειγμα παρουσιάζει έναν ακόμα τρόπο απεικόνισης κειμένου. Αυτή τη φορά, το κείμενο καταχωρείται σε έναν πίνακα.

```
char Mens1[]={"I Love Arduino"}; // Μήνυμα προς εμφάνιση
```

Δύο **for()** επαναλήψεις αποστέλλουν έναν-έναν χαρακτήρα του πίνακα στην οθόνη με χρονικά διαστήματα των 0.150 δευτερολέπτων μεταξύ τους χρησιμοποιώντας τη συνάρτηση **lcd.write()**. Ο δρομέας ξεκινάει από το θέση 0 της γραμμής 0 (η γραμμή 0 είναι η πρώτη της οθόνης LCD) και ξεκινάει να εμφανίζει τους χαρακτήρες από αριστερά προς τα δεξιά, ως συνήθως. Ο δρομέας πηγαίνει στη γραμμή 1 και θέση 15 (την τελευταία θέση στη δεύτερη γραμμή) στη δεύτερη **for()** και ξεκινάει να γράφει τους χαρακτήρες από τα δεξιά προς τα αριστερά.

Έπειτα τροποποιούμε το χρονικό διάστημα σε τρία δευτερόλεπτα και η συνάρτηση **lcd.clear()** απαλείφει τα περιεχόμενα της οθόνης. Έπειτα υπάρχει μία τελευταία ακολουθία ενός δευτερολέπτου, και ο κύκλος ξεκινάει ξανά.

7. ΠΑΡΑΔΕΙΓΜΑ 6: Κύλιση

Οι συναρτήσεις **scrollDisplayLeft()** και **scrollDisplayRight()** σας επιτρέπουν να επιτύχετε μερικά ωραία εφέ – τα οποία πιθανότατα θα έχετε δει σε διαφημιστικές πινακίδες.

Αυτές οι συναρτήσεις μετακινούν οτιδήποτε απεικονίζεται στην οθόνη προς τα αριστερά ή προς τα δεξιά. Αυτό το παράδειγμα απεικονίζει ένα κείμενο δύο γραμμών στην οθόνη. Στη συνέχεια ολόκληρο



το κείμενο ξεκινάει να μετακινείται προς τα αριστερά από κάθε θέση στην επόμενη με χρονικά διαστήματα 3 δευτερολέπτων.

Μόλις ο τελευταίος χαρακτήρας κρυφτεί από την αριστερή μεριά στην οθόνη, το κείμενο ξεκινάει να μετακινείται ξανά έως ότου να κρυφτεί ένας χαρακτήρας από τη δεξιά μεριά της οθόνης. Μετά ξεκινάει να μετακινείται προς τα αριστερά ξανά μέχρι να επιστρέψει στην αρχική του θέση. Ύστερα από αυτό, η οθόνη ανάβει και σβήνει και λίγο.

8. ΠΑΡΑΔΕΙΓΜΑ 7: Αυτόματη κύλιση

Ακόμα ένα παράδειγμα που θα σας επιτρέψει να δημιουργήσετε εντυπωσιακά εφέ. Η συνάρτηση **autoscroll()** μετακινεί όλο το κείμενο μία θέση προς τα αριστερά κάθε φορά που προστίθεται ένας χαρακτήρας, ενώ η **noAutoscroll()** απενεργοποιεί αυτό το χαρακτηριστικό.

Όταν η αυτόματη κύλιση είναι ενεργή η συνάρτηση μετακινεί αυτόματα τους χαρακτήρες που υπάρχουν στην οθόνη. Στο προηγούμενο παράδειγμα, για να επιτύχουμε αυτό το εφέ, έπρεπε να χρησιμοποιήσουμε τις συναρτήσεις **scrollDisplayLeft()** και **scrollDisplayRight()**.

Η συνάρτηση μετακινεί τους χαρακτήρες αυτόματα, προς τα αριστερά (από προεπιλογή) ή προς τα δεξιά, χαρακτηριστικό που εξαρτάται από ποια συνάρτηση καλέσατε τελευταία: την **leftToRight()** ή την **RightToLeft()**.

Το παράδειγμα που ακολουθεί χρησιμοποιεί δύο κείμενα τα οποία είχαν οριστεί σε δύο διαφορετικούς πίνακες:

```
char Mess1[] = { "ARDUINO" };           // Μήνυμα 1
```

```
char Mess2[] = { "I Love you" };       // Μήνυμα 2
```

Η πρώτη επανάληψη **for()** απεικονίζει το μήνυμα "Mess1" στην οθόνη χαρακτήρα-προς-χαρακτήρα. Εγγράφει αυτόματα τους χαρακτήρες από τα αριστερά προς τα δεξιά με διαστήματα τριών δευτερολέπτων ανάμεσά τους.

Η δεύτερη επανάληψη **for()** απεικονίζει το μήνυμα "Mess2" στην οθόνη χαρακτήρα-προς-χαρακτήρα. Παρόλα αυτά, πριν την εκτέλεση της επανάληψης, ο δρομέας τοποθετείται στην τελευταία θέση της δεύτερης γραμμής της οθόνης και εκτελείται η συνάρτηση **lcd.autoscroll()**. Εγγράφει τους χαρακτήρες από τα αριστερά προς τα δεξιά, αλλά κάθε φορά που εμφανίζει ένα νέο χαρακτήρα, μετακινεί αυτόματα τα περιεχόμενα της οθόνης προς τα αριστερά (από προεπιλογή).

Τέλος, εκτελείται η συνάρτηση **lcd.noAutoscroll()** ακολουθούμενη από την **lcd.clear()** η οποία καθαρίζει τα περιεχόμενα της οθόνης και ο κύκλος επαναλαμβάνεται ξανά.



9. ΠΑΡΑΔΕΙΓΜΑ 8: Ειδικοί χαρακτήρες

Για να ολοκληρώσουμε αυτό το κομμάτι παραδειγμάτων που παρουσιάζουν τις πιο χρήσιμες συναρτήσεις της βιβλιοθήκης “**LiquidCrystal.h**”, θα κατασκευάσουμε και θα χρησιμοποιήσουμε τους δικούς μας γραφικούς χαρακτήρες.

Για να ήμαστε πιο συγκεκριμένοι, θα κατασκευάσουμε πέντε χαρακτήρες: μία καρδιά, ένα χαρούμενο πρόσωπο, ένα σοβαρό πρόσωπο, μία φιγούρα με τα χέρια πάνω και άλλη μία με τα χέρια κάτω.

Κάθε ένας από αυτούς ορίζεται χρησιμοποιώντας έναν πίνακα. Αυτοί είναι οι πέντε χαρακτήρες: **heart[8]**; **happy[8]**; **serious[8]**; **down[8]** και **up[8]**. Κάθε ένας από αυτούς τους χαρακτήρες περιέχει μία δυαδική περιγραφή σχετικά με το ποια «σημεία» (pixels) θα πρέπει να ενεργοποιηθούν σε κάθε μία από τις περιπτώσεις.

Η συνάρτηση **lcd.createChar()** μεταφέρει τα περιεχόμενα των πινάκων στη μνήμη CGRAM της οθόνης κατά την κλήση της συνάρτησης **setup()**. Η συνάρτηση μεταφέρει συνολικά 40 byte, οκτώ για κάθε χαρακτήρα. Επιπροσθέτως, σε κάθε έναν από τους χαρακτήρες ανατίθεται ένας αριθμός από το 0 έως το 4.

```
lcd.createChar(0, heart);           // Δημιουργία νέου χαρακτήρα n° 0
lcd.createChar(1, happy);          // Δημιουργία νέου χαρακτήρα n° 1
lcd.createChar(2, serious);        // Δημιουργία νέου χαρακτήρα n° 2
lcd.createChar(3, down);           // Δημιουργία νέου χαρακτήρα n° 3
lcd.createChar(4, up);              // Δημιουργία νέου χαρακτήρα n° 4
```

Τέλος, μπορείτε να απεικονίσετε καθέναν από τους γραφικούς χαρακτήρες στην οθόνη στο σημείο που βρίσκεται ο δρομέας σα να ήταν ένας οποιοσδήποτε ASCII χαρακτήρας. Το μόνο που έχετε να κάνετε είναι να χρησιμοποιήσετε τη συνάρτηση **lcd.write(n)**, όπου “n” να ισούται με τον αριθμό που έχετε αντιστοιχήσει σε κάθε χαρακτήρα, όπως κάναμε στο παράδειγμα, με τους αριθμούς από το 0 έως το 4.

10. ΠΑΡΑΔΕΙΓΜΑ 9: Απεικόνιση

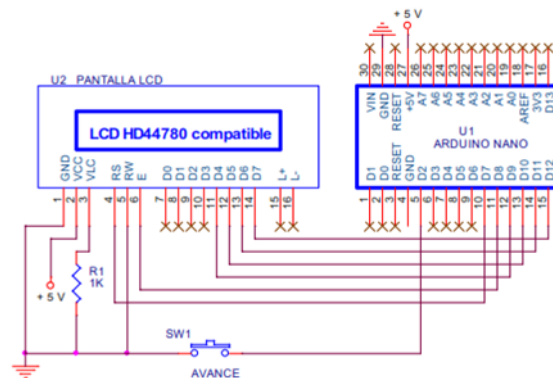
Ένα ιδιαίτερα πρακτικό και χρήσιμο παράδειγμα. Θα δημιουργήσουμε μία σύνδεση ανάμεσα στο σειριακό κανάλι του υπολογιστή σας και την LCD οθόνη. Εκ των πραγμάτων, το Arduino NANO θα βρίσκεται ανάμεσά τους.

Ο ελεγκτής λειτουργεί σαν μία «γέφυρα» ανάμεσά τους. Από τη μία πλευρά συνδέεται με τον υπολογιστή μέσω ενός καλωδίου USB ενώ από την άλλη, συνδέεται με την οθόνη με τον ίδιο τρόπο που τον συνδέαμε μέχρι τώρα. Χρησιμοποιώντας ένα σειριακό τερματικό, θα αποστείλετε έναν αριθμό χαρακτήρων από τον υπολογιστή χρησιμοποιώντας το σειριακό κανάλι: αυτοί θα ληφθούν από το Arduino και θα εμφανιστούν στην LCD οθόνη.

11. ΠΑΡΑΔΕΙΓΜΑ 10: Απεικόνιση φυσικών αριθμών

Έχετε ήδη δει πως μπορούμε να απεικονίσουμε κάθε είδος κειμένου και χαρακτήρων στην οθόνη LCD και πως να παράγουμε διαφορετικά ήδη εφέ. Μπορείτε επίσης να απεικονίσετε αριθμούς.

Αυτό το παράδειγμα πρόκειται να εμφανίσει του αριθμούς από το 1 έως το 15 σε σειρά κάθε φορά που πατάτε το πλήκτρο που είναι συνδεδεμένο στον ακροδέκτη D2. Επιπροσθέτως, ο αριθμός θα εμφανίζεται σε δεκαδική, δεκαεξαδική και δυαδική μορφή.



Σχήμα 8

Με βάση τα όσα πλέον γνωρίζετε, το πρόγραμμα αυτό δε θα πρέπει να σας δυσκολέψει. Μία μικρή λεπτομέρεια που πιθανώς να σας φανεί χρήσιμη είναι η εξής: μιας και οι αριθμοί που θα απεικονίζετε έχουν διαφορετικά μήκη, είναι καλή ιδέα να διαγράψετε ολόκληρη την πρώτη γραμμή της οθόνης πριν τους εμφανίσετε. Μπορείτε να το κάνετε αυτό γεμίζοντας την πρώτη γραμμή με κενά διαστήματα (« »). Έτσι δε θα μένουν 'υπολείμματα'.

Ρίξτε μια προσεκτική ματιά στο πρόγραμμα. Δοκιμάστε να σβήσετε τις δυο πρώτες συναρτήσεις της **loop()** και να παρατηρήσετε τι συμβαίνει: έτσι θα κατανοήσετε τι εννοούμε λέγοντας 'υπολείμματα'.

12. ΠΑΡΑΔΕΙΓΜΑ 11: Απεικόνιση αριθμών κινητής υποδιαστολής

Συνεχίζοντας από το παραπάνω παράδειγμα, θα μιλήσουμε για την απεικόνιση αριθμών κινητής υποδιαστολής. Σε αυτή την περίπτωση, πρόκειται να εμφανίσουμε την τετραγωνική ρίζα του αριθμού N με ακρίβεια δύο δεκαδικών ψηφίων και το N πολλαπλασιασμένο με το 'π' (3,14159) με ακρίβεια

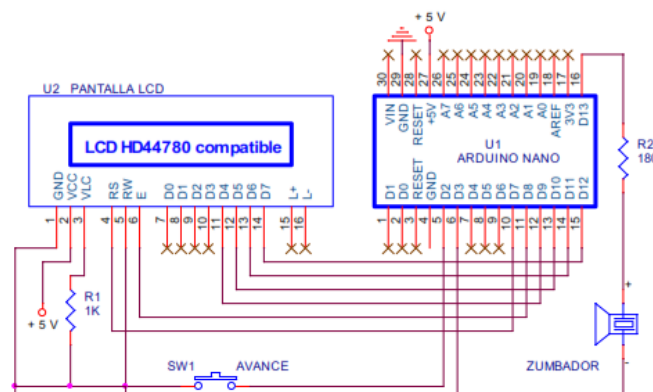
τεσσάρων δεκαδικών ψηφίων. Ο αριθμός N αυξάνεται κάθε φορά που πατάτε το πλήκτρο που είναι συνδεδεμένο στον ακροδέκτη D2. Ξεκινάει από το 0 και καταλήγει στο 9.

Τόσο το σχηματικό διάγραμμα, όσο και η συνδεσμολογία, είναι τα ίδια με το παράδειγμα 10.

13. ΠΑΡΑΔΕΙΓΜΑ 12: Μενού

Μία οθόνη LCD είναι ιδανικό περιφερειακό για να κατασκευάσετε μία εύχρηστη διεπαφή χρήστη. An LCD screen in an ideal peripheral for making an intuitive user interface. Στην ουσία υπάρχουν μενού επιλογών που σας επιτρέπουν να επιλέξετε από διαφορετικές λειτουργίες προς εκτέλεση.

Σε αυτό το παράδειγμα η ιδέα είναι να κατασκευάσουμε ένα μενού από έξι επιλογές που εμφανίζονται στην οθόνη η μία μετά την άλλη. Χρησιμοποιώντας ένα πλήκτρο για να προχωρήσουμε στην επόμενη επιλογή και άλλο ένα για να επιστρέψουμε στην προηγούμενη, θα ήμαστε σε θέση να πλοηγηθούμε στο μενού και να εξετάσουμε τις διαθέσιμες επιλογές. Παρακάτω παρουσιάζεται το σχηματικό διάγραμμα:



Σχήμα 9

Η συνδεσμολογία ανάμεσα στο ελεγκτή Arduino και την οθόνη είναι η ίδια με αυτή που χρησιμοποιούσαμε μέχρι τώρα. Σε αυτή την περίπτωση όμως, υπάρχει ένα δεύτερο πλήκτρο που ονομάζεται SW2. Το πλήκτρο SW1 συνδέεται στον ακροδέκτη D2 και οι διαθέσιμες επιλογές παρουσιάζονται στην οθόνη από την πρώτη, έως την τελευταία. Όταν πατηθεί το πλήκτρο SW2 που είναι συνδεδεμένο στον ακροδέκτη D3 βλέπουμε τις επιλογές με αντίστροφη σειρά: από την τελευταία έως την πρώτη. Και οι δύο ακροδέκτες έχουν ενεργοποιημένες τις pull-up αντιστάσεις τους. Κατ' αυτό τον τρόπο γλυτώνουμε τις δύο εξωτερικές αντιστάσεις που θα έπρεπε να χρησιμοποιήσουμε. Η οθόνη πραγματοποιεί κάθετη κύλιση για να παρουσιάσει όλες τις διαθέσιμες επιλογές του μενού τη μία μετά την άλλη. Θα χρησιμοποιήσουμε επίσης ένα πιεζοηλεκτρικό buzzer συνδεδεμένο στον ακροδέκτη D13. Κατ' αυτό τον τρόπο, όταν πατιέται είτε το πλήκτρο SW1, ή το SW2, θα ειδοποιούμαστε ηχητικά.



Παρατηρείστε το πρόγραμμα και εξετάστε τον τρόπο λειτουργίας του. Θα πούμε μόνο πως προσπαθήσαμε να το υλοποιήσουμε με όσο γίνεται πιο κατανοητό τρόπο. Σίγουρα θα υπάρχουν και άλλες λύσεις, ενώ δεν αυτή που παρουσιάζουμε, δεν θα είναι αναγκαστικά η καλύτερη. Ελπίζω να μπορέσετε να βελτιώσετε το πρόγραμμα με δική σας πρωτοβουλία.

Αρχή:

Πρώτα απ' όλα, περιλαμβάνουμε τη βιβλιοθήκη "LiquidCrystal.h" και συνδέουμε την οθόνη: τίποτα καινούριο μέχρι εδώ. Έπειτα, δηλώνουμε τις μεταβλητές Option και N. Η πρώτη περιέχει την επιλογή που είναι ενεργή την τρέχουσα στιγμή και η δεύτερη τον αριθμό των διαθέσιμων επιλογών του μενού. Θα πρέπει επίσης να δηλώσουμε έναν arrowhead πίνακα που ορίζει τους γραφικούς χαρακτήρες.

Συναρτήσεις:

Γι' αυτό το project, έχουν δημιουργηθεί δύο συναρτήσεις: η visualize() και η beep(). Η πρώτη εμφανίζει στην οθόνη το μήνυμα που αντιστοιχεί στην επιλεγείσα επιλογή βάσει της μεταβλητής "Option". Ενώ τη συνάρτηση beep(), τη γνωρίζουμε ήδη: παράγει έναν τόνο.

setup():

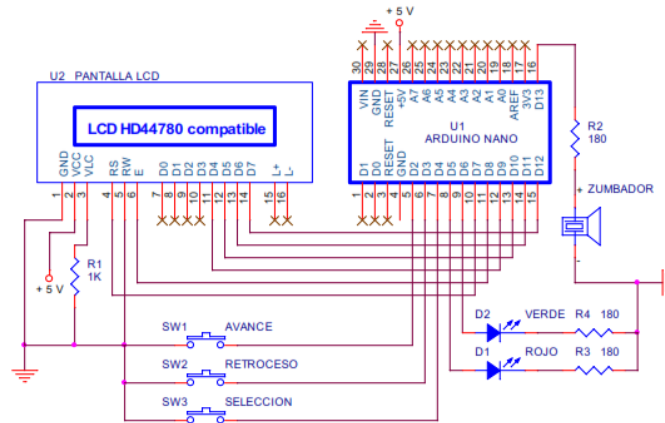
Ορίζουμε τους ακροδέκτες D2 και D3 σαν εισόδους με ενεργές τις εσωτερικές αντιστάσεις pull-up, τον ακροδέκτη D13 σαν έξοδο, την οθόνη ως οθόνη 2 γραμμών και 16 χαρακτήρων, ενώ τέλος, δημιουργούμε τον γραφικό χαρακτήρα n° 0 (ένα βέλος). Δεν πραγματοποιούμε κάτι σημαντικό εδώ.

loop():

- Ξεκινάει καθαρίζοντας την οθόνη και εμφανίζοντας τον γραφικό χαρακτήρα n° 0 (το βέλος).
- Τοποθετεί τον δρομέα στην πρώτη γραμμή και απεικονίζει την ενεργή επιλογή βάσει της μεταβλητής "option". Χρησιμοποιεί τη συνάρτηση visualize().
- Τοποθετεί τον δρομέα στη δεύτερη γραμμή και απεικονίζει την επόμενη επιλογή (option+1). Και πάλι, χρησιμοποιεί τη συνάρτηση visualize().
- Περιμένει έως ότου κάποιος πιέσει κάποιο από τα πλήκτρα SW1 και SW2 που είναι συνδεδεμένα στους ακροδέκτες D2 και D3.
- Εάν πατηθεί το πλήκτρο του ακροδέκτη D2, πραγματοποιεί απόσβεση αναπήδησης (debouncing) και περιμένει έως ότου το πλήκτρο αφεθεί. Δημιουργεί έναν σφυριχτό ήχο και αυξάνει την τιμή της μεταβλητής "option". Πηγαίνει στην επόμενη επιλογή του μενού.
- Εάν πατηθεί το πλήκτρο του ακροδέκτη D3, πραγματοποιεί απόσβεση αναπήδησης (debouncing) και περιμένει έως ότου το πλήκτρο αφεθεί. Δημιουργεί έναν σφυριχτό ήχο και μειώνει την τιμή της μεταβλητής "option". Πηγαίνει στην προηγούμενη επιλογή του μενού.

14. ΠΑΡΑΔΕΙΓΜΑ 13: ΛΙΣΤΑ ΕΠΙΛΟΓΩΝ

Φανταστείτε το σαν μία «επέκταση» του προηγούμενου παραδείγματος. Παρατηρήστε το σχηματικό διάγραμμα.



Σχήμα 10

Προσθέτουμε ένα πλήκτρο --SW3- στον ακροδέκτη D4; το πλήκτρο αυτό εκτελεί την επιλεχθείσα επιλογή. Προσθέτουμε επιπλέον δύο ενδεικτικές λυχνίες LED: μία κόκκινη και μία πράσινη, συνδεδεμένες αντίστοιχα στους ακροδέκτες D5 και D6. Αναπαριστούν το αποτέλεσμα της εκτέλεσης των επιλογών.

Το πρόγραμμα αυτό μοιάζει πολύ με αυτό του προηγούμενου παραδείγματος. Η μόνη διαφορά είναι η συνάρτηση execute() η οποία και καλείται κάθε φορά που πατείτε το πλήκτρο SW3. Αυτή η νέα συνάρτηση ελέγχει την τιμή την μεταβλητής "option" και ανάλογα με την επιλεχθείσα επιλογή, ενεργοποιεί ή απενεργοποιεί το πράσινο ή το κόκκινο LED που είναι συνδεδεμένα στους ακροδέκτες D5 και D6.

15. ΠΑΡΑΔΕΙΓΜΑ 14: Μία τελευταία βελτίωση

Φανταστείτε μία εφαρμογή που δεν έχει μόνο έξι επιλογές στο μενού της όπως αυτή που είδαμε: έχει δεκάδες. Κάθε φορά που εκκινείτε το σύστημα ή το επανεκκινείτε με το πλήκτρο RESET, ξεκινάει πάντα έχοντας επιλεγμένη την πρώτη. Θα πρέπει να προσπεράσετε όλες τις επιλογές μέχρι να βρείτε αυτή που θέλετε.

Ίσως να ήταν καλή ιδέα η οθόνη LCD να έδειχνε την τελευταία επιλογή που διαλέξατε έτσι ώστε να μην την ψάχνετε κάθε φορά που επανεκκινείτε το σύστημα. Μαντεύετε τι θα μπορούσατε να χρησιμοποιήσουμε; Αυτό είναι η μνήμη EEPROM. Κάθε φορά που διαλέγουμε μία νέα επιλογή, αυτή αποθηκεύεται στη μνήμη EEPROM. Ουσιαστικά, κάθε επιλογή αναπαριστάται με τίποτα περισσότερο από έναν αριθμό – αυτό που χρησιμοποιούμε στη μεταβλητή "option".



Κάθε φορά που το σύστημα επανεκκινείται, στη μεταβλητή αυτή ανατίθεται η τελευταία τιμή που καταγράφηκε στη μνήμη EEPROM. Χρησιμοποιείται τη θέση 1 από τις 1024 που είναι συνολικά διαθέσιμες.

Ανεβάστε το πρόγραμμα και βεβαιωθείτε πως λειτουργεί. Επιλέξτε και εκτελέστε τις διάφορες λειτουργίες. Έπειτα επανεκκινήστε το σύστημα. Παρατηρήστε πως η τελευταία επιλογή που κάνατε εμφανίζεται στην οθόνη.



ΠΑΡΑΠΟΜΠΕΣ

BIBΛΙΑ

- [1]. **EXPLORING ARDUINO**, Jeremy Blum, Ed.Willey
- [2]. **Practical ARDUINO**, Jonathan Oxe & Hugh Blemings, Ed.Technology in action
- [3]. **Programing Arduino, Next Steps**, Simon Monk
- [4]. **Sensores y actuadores, Aplicaciones con ARDUINO**, Leonel G.Corona Ramirez, Griselda S. Abarca Jiménez, Jesús Mares Carreño, Ed.Patria
- [5]. **ARDUINO: Curso práctico de formación**, Oscar Torrente Artero, Ed.RC libros
- [6]. **30 ARDUINO PROJECTS FOR THE EVIL GENIUS**, Simon Monk, Ed. TAB
- [7]. **Beginning C for ARDUINO**, Jack Purdum, Ph.D., Ed.Technology in action
- [8]. **ARDUINO programing notebook**, Brian W.Evans

ΙΣΤΟΣΕΛΙΔΕΣ

- [1]. <https://www.arduino.cc/>
- [2]. <https://www.prometec.net/>
- [3]. <http://blog.bricogeek.com/>
- [4]. <https://aprendiendoarduino.wordpress.com/>
- [5]. <https://www.sparkfun.com>