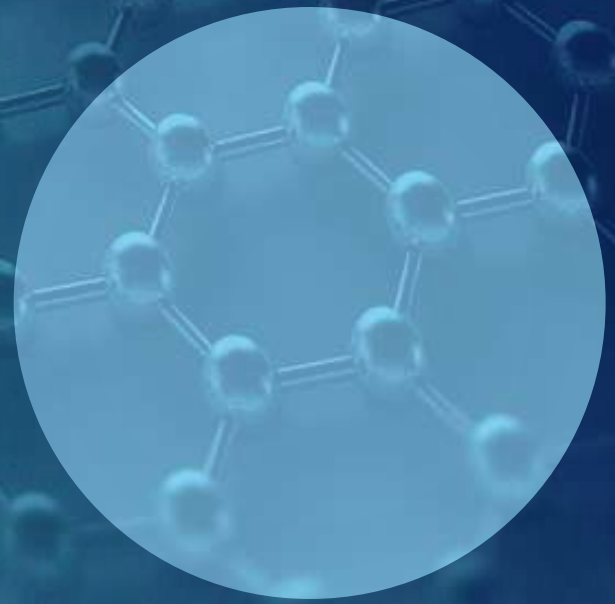
Πανεπιστήμιο Δυτικής Μακεδονίας

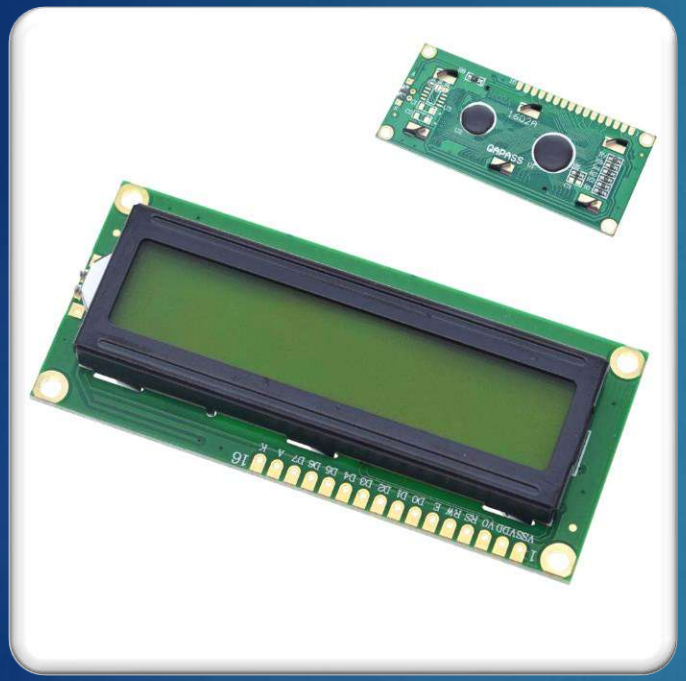
Οθόνη LCD



Δρ. Ζιούζιος Δημήτρης

dziouzios@uowm.gr

Τι είναι μια οθόνη LCD;



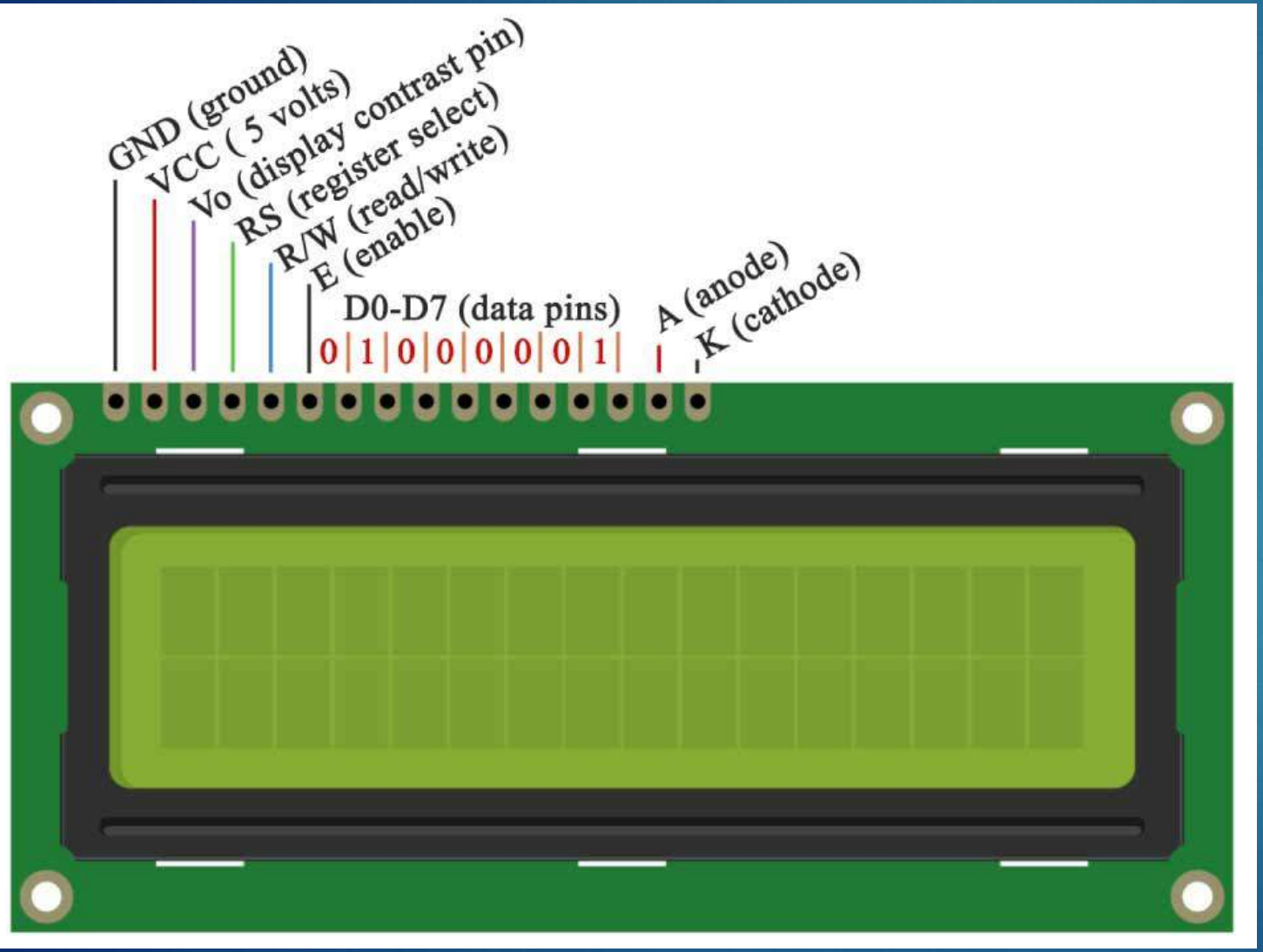
Μια LCD (Liquid Crystal Display) είναι μια οθόνη που χρησιμοποιεί υγρούς κρυστάλλους για να εμφανίζει χαρακτήρες, αριθμούς ή εικόνες. Είναι πολύ δημοφιλής σε ηλεκτρονικές εφαρμογές, όπως Arduino projects, υπολογιστές, τηλέφωνα και τηλεοράσεις.

- Πώς λειτουργεί μια LCD;Μια LCD αποτελείται από δύο στρώματα πολωμένου γυαλιού που περιέχουν ένα στρώμα υγρών κρυστάλλων. Όταν ένα ηλεκτρικό ρεύμα περνά από αυτά, αλλάζουν την πόλωση του φωτός και επιτρέπουν ή εμποδίζουν τη διέλευσή του, δημιουργώντας εικόνες ή κείμενο.

- Είδη LCD οθονών: Υπάρχουν διάφοροι τύποι LCD, ανάλογα με το μέγεθος, τον τρόπο ελέγχου και τη χρήση:

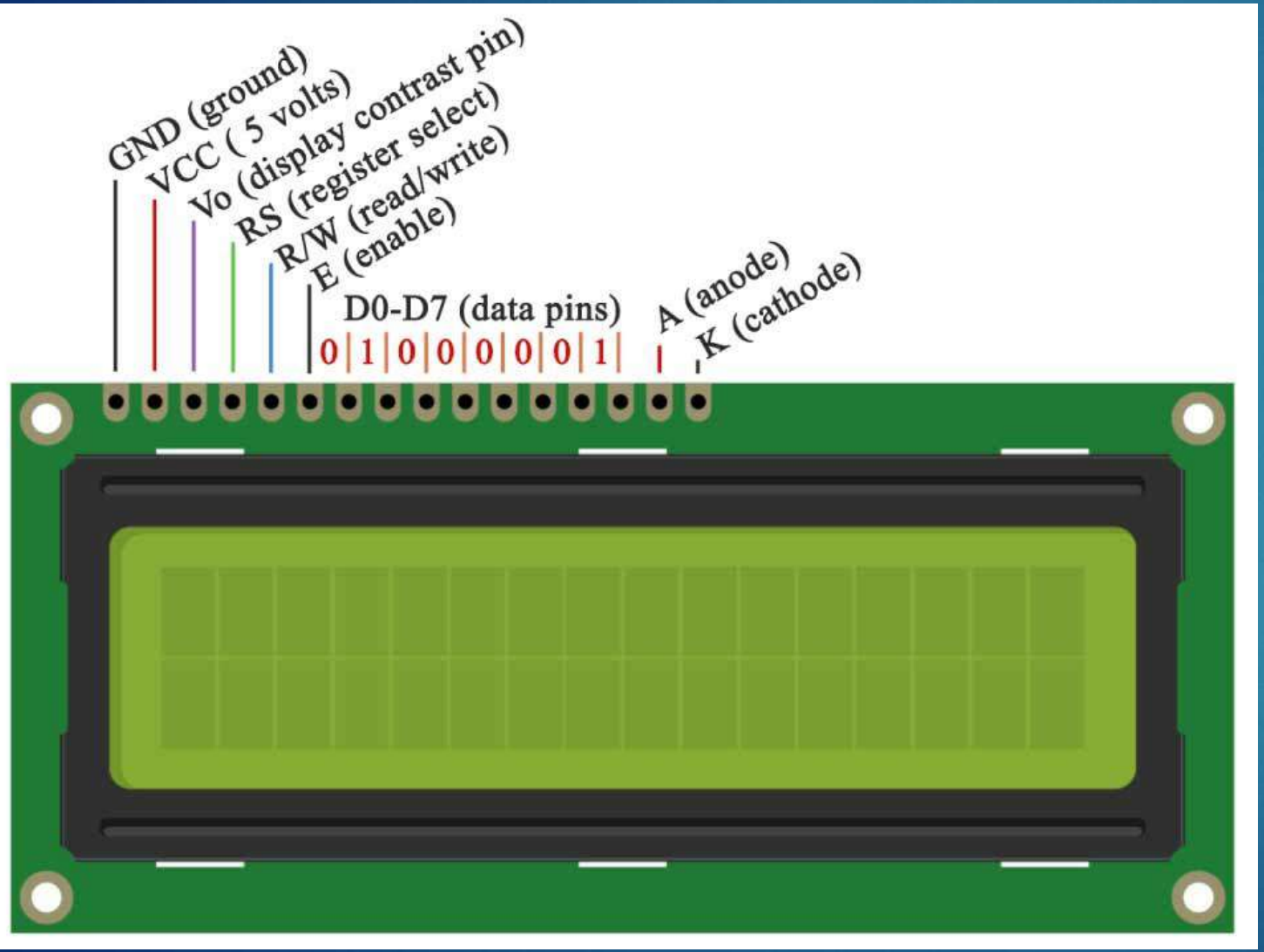
1. LCD 16x2 & 20x4 → Μικρές οθόνες με χαρακτήρες, συχνά σε Arduino projects.
2. Graphic LCD → Δείχνουν pixel-based γραφικά αντί για απλά γράμματα.
3. TFT LCD → Πολύχρωμες LCD για προηγμένες εφαρμογές, όπως smartphones.

Οθόνη LCD



LCD 16x2	Περιγραφή
VSS	Γείωση
VDD	Τροφοδοσία 5V
V0	Ρύθμιση αντίθεσης
RS	Επιλογή καταχωρητή
RW	Ανάγνωση/Εγγραφή
E	Ενεργοποίηση
D4	Δεδομένα bit 4
D5	Δεδομένα bit 5
D6	Δεδομένα bit 6
D7	Δεδομένα bit 7
A	Οπίσθιος φωτισμός 5V
K	Γείωση οπίσθιου φωτισμού

Οθόνη LCD



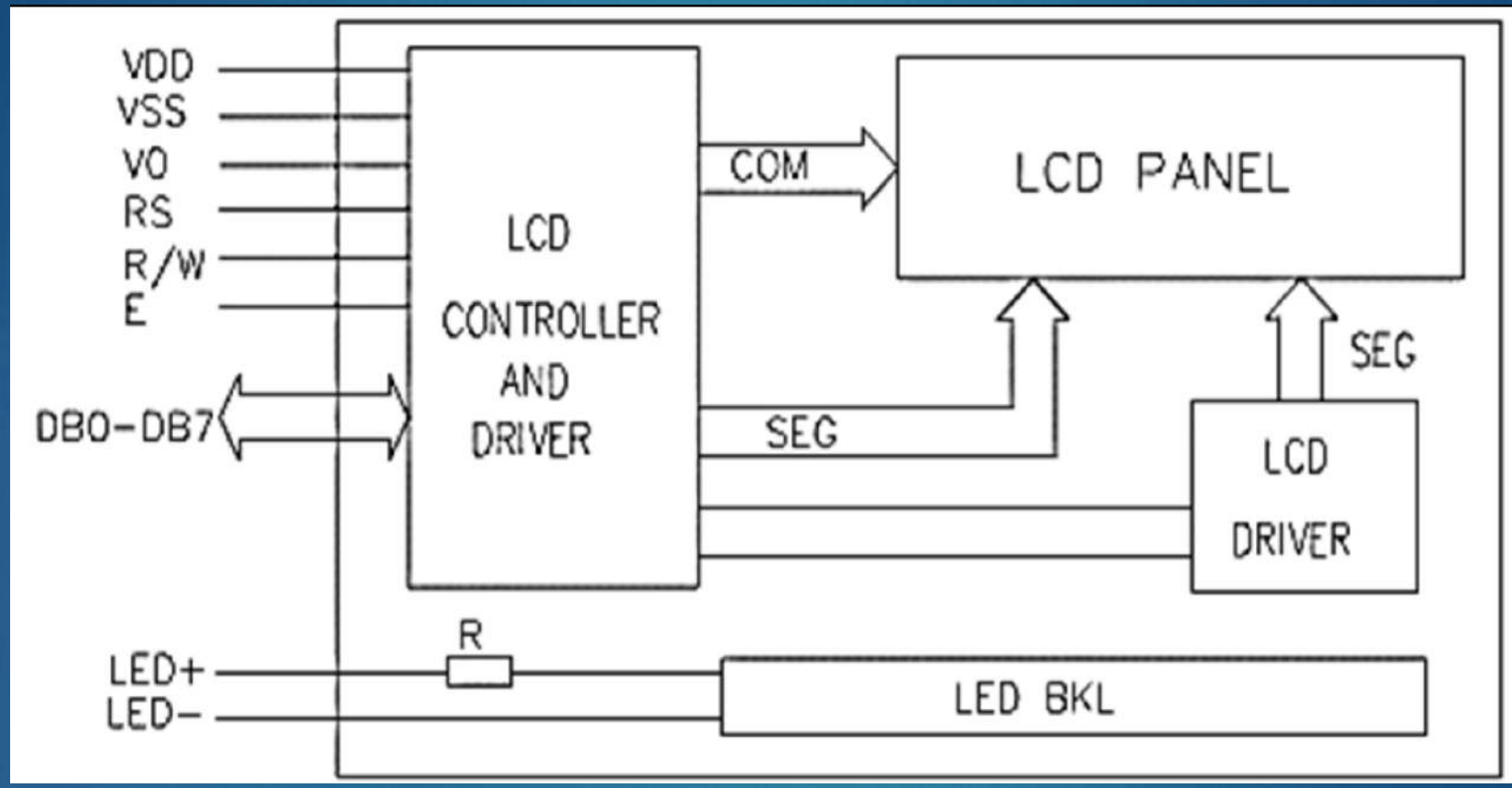
*Διαθέτει οθόνη με 2 σειρές των 16 χαρακτήρων.

*Κάθε χαρακτήρας έχει διαστάσεις 5χ8 pixel

*204 χαρακτήρες είναι αποθηκευμένοι στη μνήμη του display .

*Ένας ενσωματωμένος μικροελεγκτής δέχεται εντολές από εξωτερική συσκευή και διαχειρίζεται την λειτουργία της LCD οθόνης.

Οθόνη LCD


VDD
VSS
VO
RS
R/W
E
DB0-DB7
LCD
CONTROLLER
AND
DRIVER
COM
LCD PANEL
SEG
SEG
LCD
DRIVER
R
LED+
LED-
LED BKL

Οθόνη LCD

Τι είναι το RS στην LCD 16x2;

Η ακίδα **RS (Register Select)** είναι μια από τις βασικές ακίδες ελέγχου της LCD 16x2.
Χρησιμοποιείται για να **επιλέξει ποιον τύπο δεδομένων στέλνουμε στην οθόνη**.

1. **Εντολές (Commands)**, όπως **καθαρισμός οθόνης ή μετακίνηση κέρσορα**
2. **Χαρακτήρες (Data)**, δηλαδή **το κείμενο που εμφανίζεται στην οθόνη**

RS	Λειτουργία
LOW (0)	Εντολές (Commands)
HIGH (1)	Χαρακτήρες (Data)

Παράδειγμα λειτουργίας

Όταν στέλνουμε μια εντολή, όπως lcd.clear(), η RS μπαίνει σε LOW (0).

Όταν στέλνουμε ένα γράμμα, όπως lcd.print("A"), η RS μπαίνει σε HIGH (1).

Οθόνη LCD

Ποιος ο ρόλος της ακίδας RW;
Η ακίδα **RW (Read/Write)** καθορίζει αν η οθόνη LCD θα:

1. **Διαβάζει δεδομένα** από το Arduino (**Read Mode**)
2. **Γράφει δεδομένα** στην οθόνη (**Write Mode**)

Write Mode (Εγγραφή Δεδομένων)
Όταν η ακίδα **RW είναι συνδεδεμένη στο GND**, η οθόνη λειτουργεί **μόνο σε εγγραφή**.
✓ Αυτός είναι ο **συνηθισμένος τρόπος** λειτουργίας σε Arduino.
✓ Το **Arduino στέλνει εντολές & χαρακτήρες στην LCD** χωρίς να χρειάζεται να διαβάσει δεδομένα από αυτήν.

RW → GND

Read Mode (Ανάγνωση Δεδομένων)
Αν η RW **συνδεθεί στο 5V**, τότε η οθόνη μπαίνει σε λειτουργία **ανάγνωσης**.

✓ Σε αυτήν τη λειτουργία, το Arduino μπορεί να **διαβάσει την κατάσταση της LCD**.
✓ Χρησιμοποιείται σπάνια, γιατί η LCD απλά δείχνει ό,τι της στέλνει το Arduino.

RW → 5V

Οθόνη LCD

Τι είναι η ακίδα E (Enable) στην LCD 16x2;
Η ακίδα **E (Enable)** είναι μία από τις βασικές ακίδες ελέγχου της LCD.

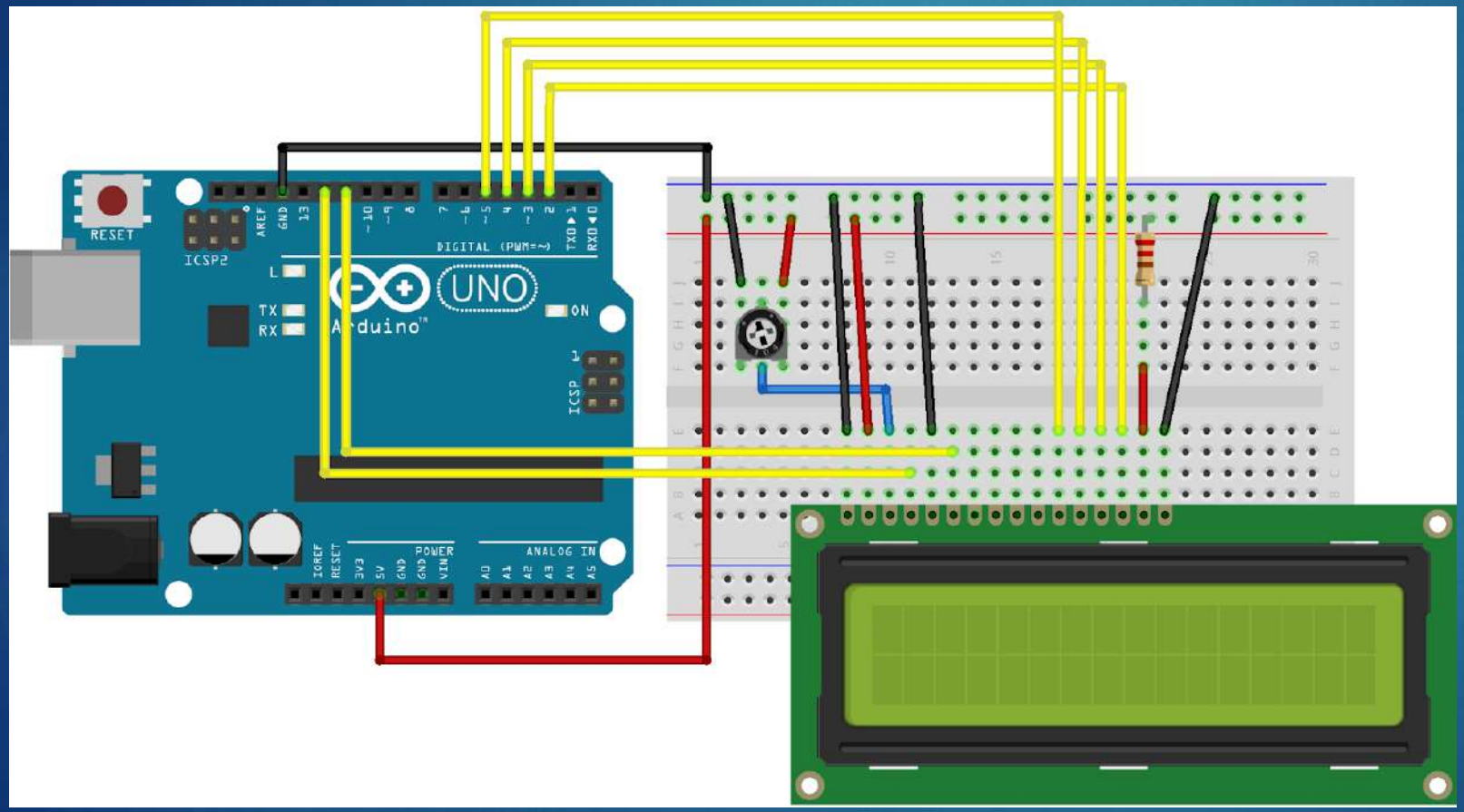
Ρόλος:
•Χρησιμοποιείται για να **ενεργοποιήσει τη μεταφορά δεδομένων** από το Arduino στην οθόνη.
•Κάθε φορά που θέλουμε να στείλουμε **εντολές ή χαρακτήρες**, πρέπει να **παλμοδοτήσουμε (pulse) την E**.
•Αντί να στέλνουμε συνεχώς δεδομένα, η LCD διαβάζει τα δεδομένα **μόνο όταν η E αλλάξει από HIGH → LOW**.

Η LCD **διαβάζει τα δεδομένα** όταν η ακίδα **E αλλάξει από HIGH σε LOW** (falling edge trigger).

Βήματα μεταφοράς δεδομένων:
1. **Θέτουμε τα δεδομένα στις ακίδες D4-D7**
2. **Ορίζουμε RS (για εντολή ή δεδομένα)**
3. **Κάνουμε την E HIGH και μετά LOW** (για να τα "διαβάσει" η LCD)

Δύο τρόποι διασύνδεσης

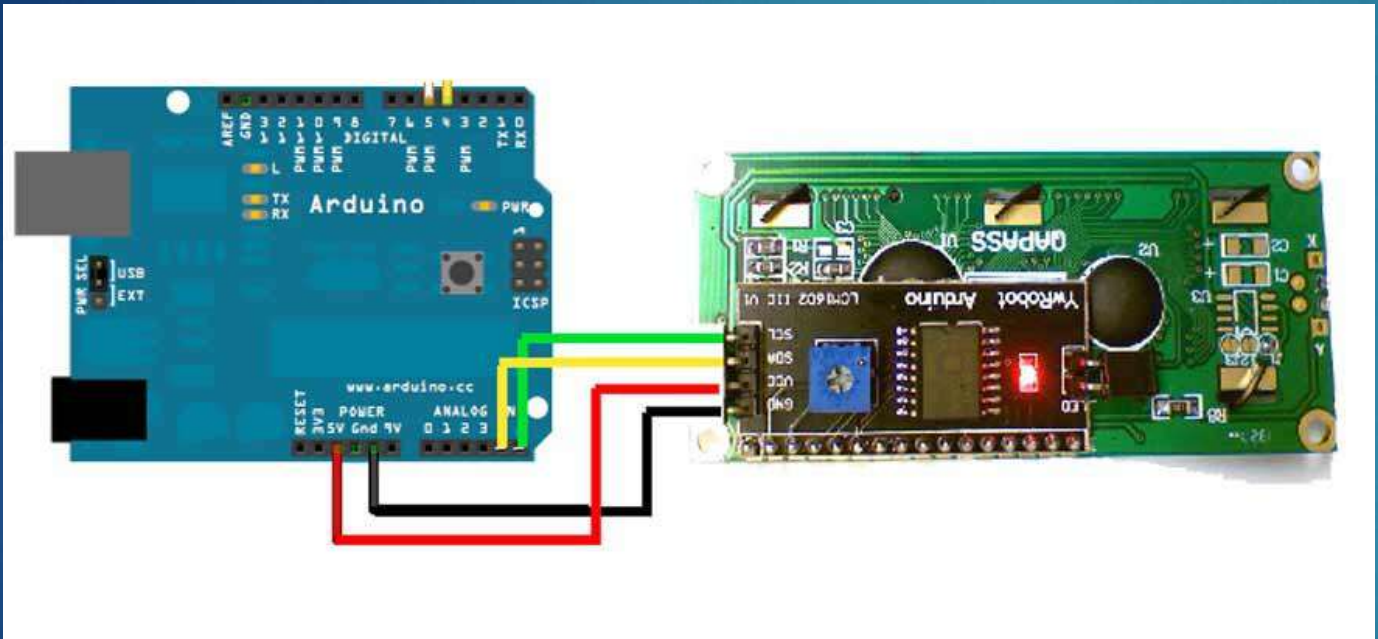
1ος τρόπος



Αυτή η σύνδεση δεσμεύει 6 εξόδους του Arduino και αν θέλουμε να χειριστούμε το backlight από το arduino, δηλαδή να το ενεργοποιούμε και να το απενεργοποιούμε κατά βούληση, τότε χρειαζόμαστε 7 εξόδους.

Δύο τρόποι διασύνδεσης

2ος τρόπος



Η σύνδεση της οθόνης μέσω I2C γίνεται με 4 καλώδια. Τα 2 είναι για την τροφοδοσία +5V και GND. Τα άλλα 2 για clock και data.

I2C (Inter-Integrated Circuit)

Το I2C (Inter-Integrated Circuit) είναι ένα πρωτόκολλο σειριακής επικοινωνίας που χρησιμοποιείται για τη σύνδεση χαμηλής ταχύτητας περιφερειακών σε ενσωματωμένα συστήματα. Αναπτύχθηκε από την Philips Semiconductor (νυν NXP) και χρησιμοποιείται ευρέως σε μικροελεγκτές, αισθητήρες, οθόνες LCD, EEPROMs, RTCs και πολλά άλλα κυκλώματα.

I2C (Inter-Integrated Circuit)

Βασικά Χαρακτηριστικά του I2C:

1.Δίαυλος δύο καλωδίων:

1. **SCL (Serial Clock Line):** Ρολόι που συγχρονίζει τη μεταφορά δεδομένων.
2. **SDA (Serial Data Line):** Μεταφορά δεδομένων μεταξύ συσκευών.

2.Υποστηρίζει πολλαπλές συσκευές (multi-master, multi-slave):

1. Μπορεί να υπάρχει ένας ή περισσότεροι **masters** (ελεγκτές).
2. Πολλαπλοί **slaves** (υποσυστήματα) μπορούν να συνδέονται στον ίδιο δίαυλο.

3.Διευθυνσιοδότηση συσκευών:

1. Κάθε συσκευή έχει μια μοναδική διεύθυνση 7-bit (ή 10-bit σε κάποιες περιπτώσεις).

4.Ταχύτητες επικοινωνίας:

1. **Standard Mode:** 100 kbit/s
2. **Fast Mode:** 400 kbit/s
3. **Fast Mode Plus:** 1 Mbit/s
4. **High-Speed Mode:** 3.4 Mbit/s
5. **Ultra-Fast Mode:** 5 Mbit/s

5.Χρήση pull-up αντιστάσεων:

1. Οι γραμμές SCL (Serial Data Line) και SDA (Serial Clock Line)είναι ανοικτού συλλέκτη (open-drain) και χρειάζονται εξωτερικές pull-up αντιστάσεις (~4.7kΩ ή 10kΩ συνήθως).

I2C (Inter-Integrated Circuit)

Πλεονεκτήματα του I2C:

✅ Μόνο δύο καλώδια για πολλαπλές συσκευές.
✅ Υποστηρίζει πολλαπλούς **masters** και **slaves**.
✅ Ευρέως διαδεδομένο και χρησιμοποιούμενο σε πολλές εφαρμογές.

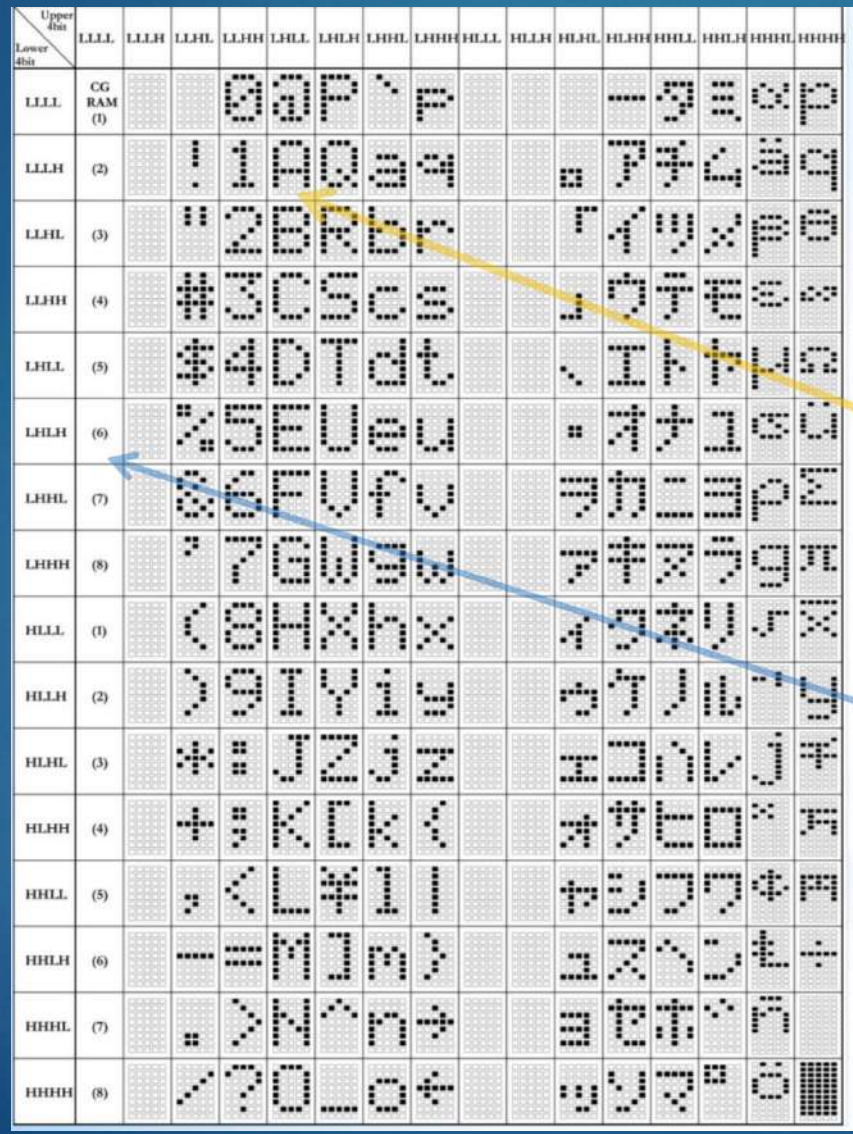
Μειονεκτήματα του I2C:

❌ Περιορισμένη ταχύτητα συγκριτικά με το **SPI**.
❌ Χρειάζεται pull-up αντιστάσεις που αυξάνουν την κατανάλωση ενέργειας.
❌ Περισσότερη πολυπλοκότητα στην επικοινωνία σε σχέση με το **UART**.

Πού χρησιμοποιείται το I2C;

- Αισθητήρες (π.χ. θερμοκρασίας, πίεσης, επιτάχυνσης)
- Οθόνες LCD/OLED
- EEPROMs και RTCs (ρολόγια πραγματικού χρόνου)
- Συστήματα ενσωματωμένων ελεγκτών (Arduino, Raspberry Pi, ESP32)

Μοτίβο Χαρακτήρων



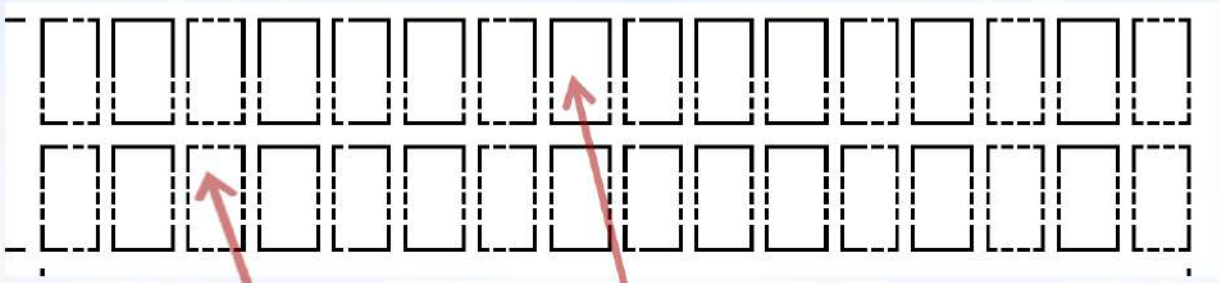
Ενσωματωμένοι χαρακτήρες

CGROM
Character Generator ROM

Για παράδειγμα:
η διεύθυνση
01000001 (LHLLLLLH)
περιέχει το σχήμα A

CGRAM
Character Generator RAM
Μπορούν να αποθηκευτούν
8 σχήματα

Μοτίβο Χαρακτήρων

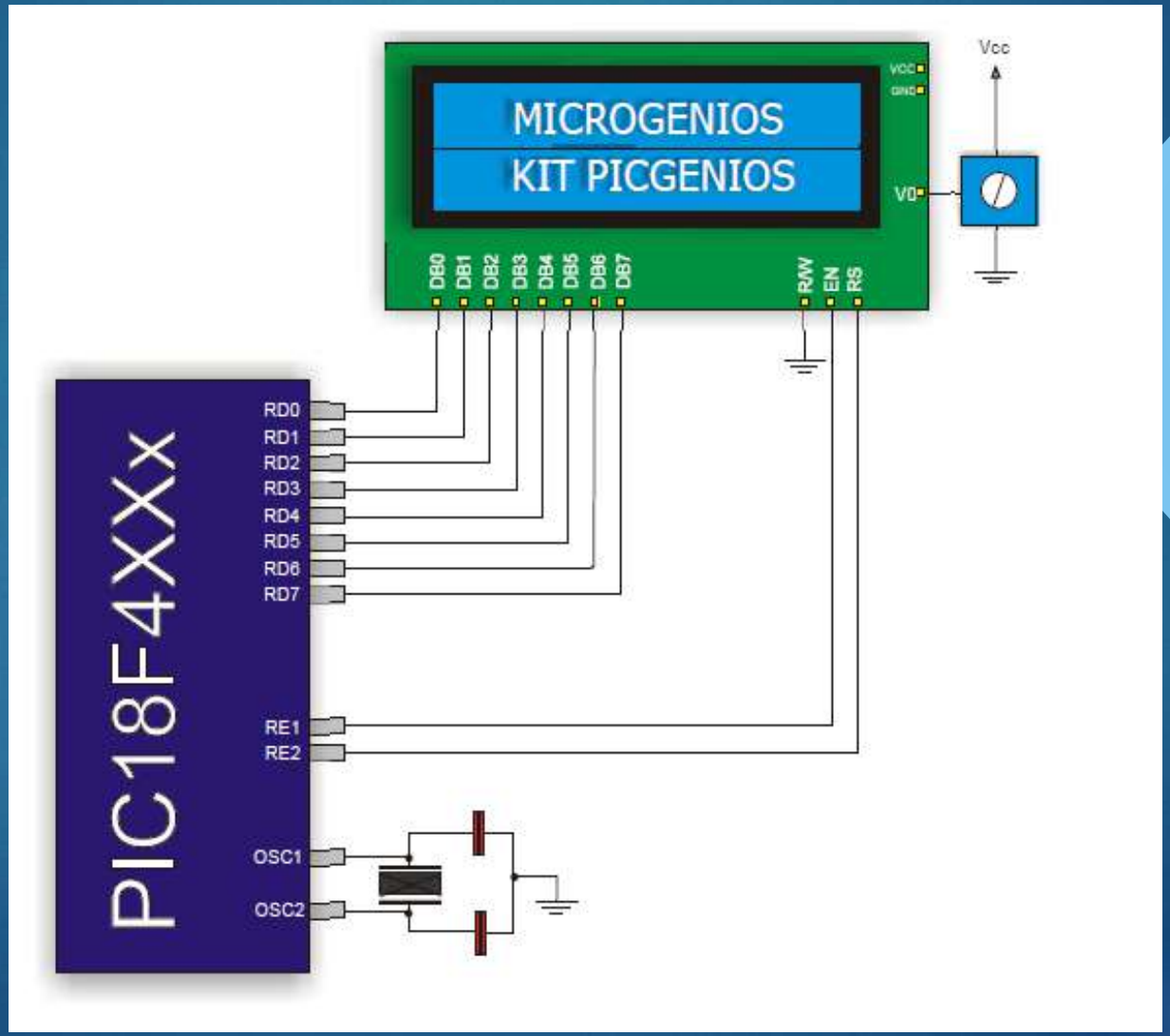


Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Μνήμη Δεδομένων Οθόνης

Display Data RAM (DDRAM)

Συνδεσμολογία LCD display στην εικονική Πλακέτα.



Παράδειγμα εντολών

Για να ορίσουμε την θέση στην οθόνη που θα εμφανιστεί ο χαρακτήρας
Set RS ->0
Set R/W->0 Set E->1
Set DB7-DB0-> 10000111 (Θέση οθόνης 07)
Set E->0 Ολοκληρώνεται η εντολή

Για να ορίσουμε ποιος χαρακτήρας θα εμφανιστεί.
Set RS ->1
Set R/W->0 Set E->1
Set DB7-DB0-> 01000001 Διεύθυνση μνήμης CGROM
Set E->0 Ολοκληρώνεται η εντολή

Παράδειγμα κώδικα σε MICROC

Εντολές για την εμφάνιση του χαρακτήρα «Κ» στην 2 σειρά και 9 θέση.

```
TRISE.F1 =0;
TRISE.F2 =0;
TRISD=0x00;
PORTE.F1 = 0;
……..//Αρχικοποίηση του LCD display.

//Ορίζουμε την θέση στην οθόνη
    PORTE.F2 = 0;                          //RS ->0
                                           //R/W->0 (είναι γειωμένο)
    PORTE.F1 = 1;                          //E->1
    PORTD = 0xC7;                          //Set DB7-DB0-> 11000111 (Θέση οθόνης 0x47)
    PORTE.F1= 0;                           //Set E->0 Ολοκληρώνεται η εντολή
//Για να ορίσουμε ποιος χαρακτήρας θα εμφανιστεί.
PORTE.F2 = 1;                              // Set RS ->1
                                           //Set R/W->0
PORTE.F1 = 1;                              //Set E->1
PORTD =0x4B;                               // Set DB7-DB0-> 01001011 Διεύθυνση μνήμης CGROM
                                           // PORTD = 'K'; ASCII του Κ είναι 0x4B.
 PORTE.F1 = 0;                             //Set E->0 Ολοκληρώνεται η εντολή
```

Βιβλιοθήκη MicroC για LCD display

* **Lcd8_Config**(&PORTC,&PORTD,0,1,2,6,5,4,3,7,1,2,0);

* **Lcd8_Cmd**(LCD_CLEAR);

LCD_FIRST_ROW	Move cursor to 1st row
LCD_CLEAR	Clear display
LCD_RETURN_HOME	Return cursor to home position
LCD_CURSOR_OFF	Turn off cursor
LCD_UNDERLINE_ON	Underline cursor on
LCD_BLINK_CURSOR_ON	Blink cursor on
LCD_MOVE_CURSOR_LEFT	Move cursor left without changing DDRAM
LCD_MOVE_CURSOR_RIGHT	Move cursor right without changing DDRAM
LCD_TURN_ON	Turn LCD display on
LCD_TURN_OFF	Turn LCD display off
LCD_SHIFT_LEFT	Shift display left
LCD_SHIFT_RIGHT	Shift display right

* **Lcd8_Out**(1, 3, "Hello!");
* **Lcd8_Out_Cp**("Here!");
* **Lcd8_Chr**(2, 3, 'i');
* **Lcd8_Chr_Cp**('e');

Βιβλιοθήκη MicroC PRO για LCD display

* LCD_Init();

* LCD_Cmd(LCD_CLEAR);

LCD_FIRST_ROW	Move cursor to 1st row
LCD_CLEAR	Clear display
LCD_RETURN_HOME	Return cursor to home position
LCD_CURSOR_OFF	Turn off cursor
LCD_UNDERLINE_ON	Underline cursor on
LCD_BLINK_CURSOR_ON	Blink cursor on
LCD_MOVE_CURSOR_LEFT	Move cursor left without changing DDRAM
LCD_MOVE_CURSOR_RIGHT	Move cursor right without changing DDRAM
LCD_TURN_ON	Turn LCD display on
LCD_TURN_OFF	Turn LCD display off
LCD_SHIFT_LEFT	Shift display left
LCD_SHIFT_RIGHT	Shift display right

* LCD_Out(1, 3, "Hello!");
* LCD_Out_Cp("Here!");
* LCD_Chr(2, 3, 'i');
* LCD_Chr_Cp('e');

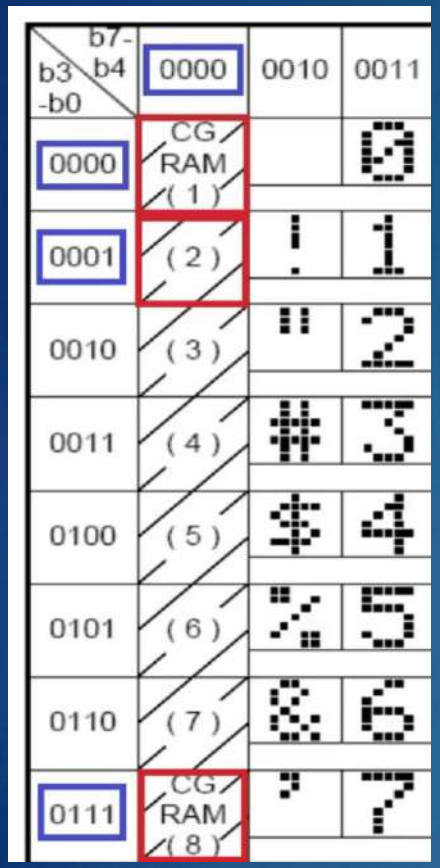
Παράδειγμα χρήσης βιβλιοθήκης Lcd για τη λειτουργία του LCD Display (MIKROC PRO).

```
sbit LCD_RS at RE2_bit;
sbit LCD_EN at RE1_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at Rd7_bit;

sbit LCD_RS_Direction at TRISE2_bit;
sbit LCD_EN_Direction at TRISE1_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;

char *text = "MyTextHere";
void main() {
        // Initialize LCD as defined above
        LCD_Init();
        LCD_Cmd(LCD_CURSOR_OFF); // Turn off cursor
        LCD_Out(1, 4, text); // Print text on LCD
}
```

Παράδειγμα κώδικα για Arduino

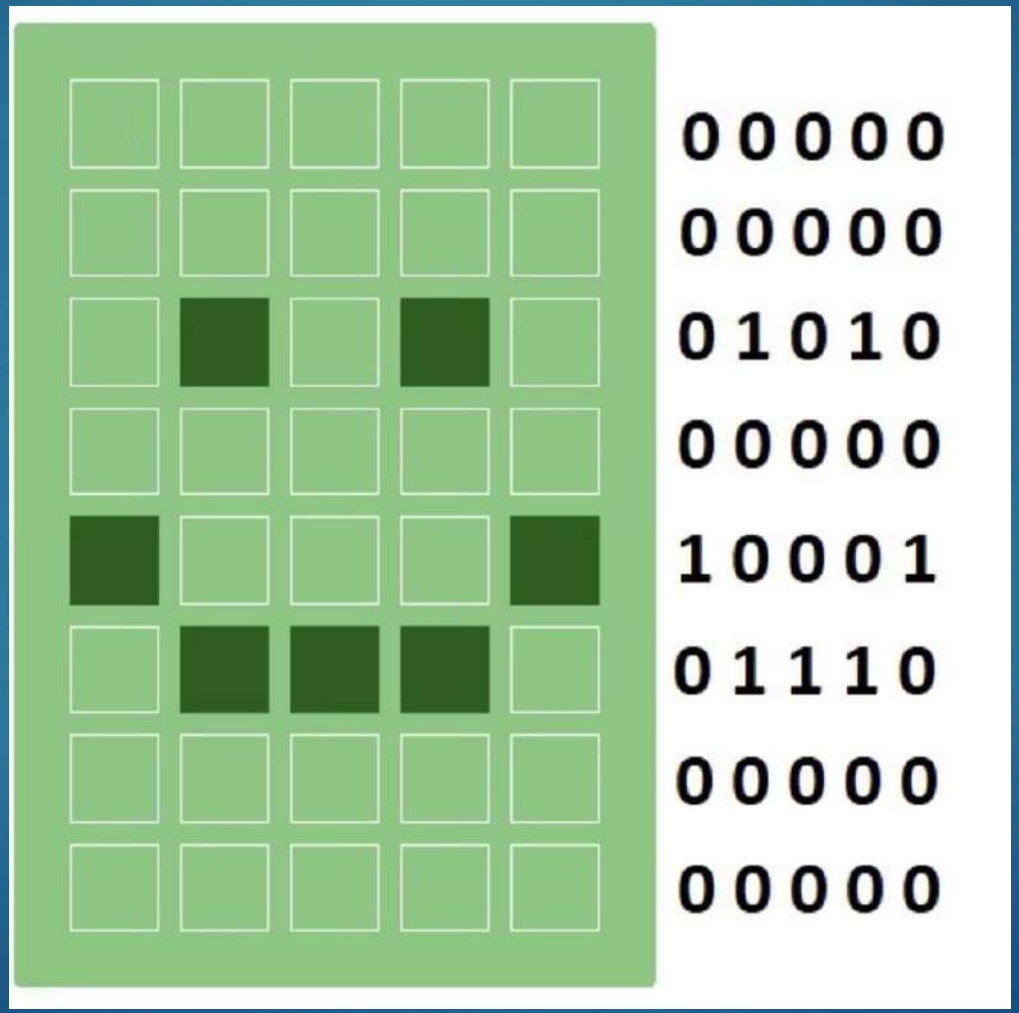
```
#include <LiquidCrystal.h> // Εισαγωγή της LCD βιβλιοθήκης

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
// Δημιουργεί ένα αντικείμενο LCD. Παράμετροι: (rs, enable, d4, d5, d6, d7)

void setup()
{
     lcd.begin(16, 2);  // Δηλώνει τον αριθμό των στηλών και των γραμμών του LCD.

     lcd.clear();       // Καθαρίζει την οθόνη του LCD

}
void loop()
{
     lcd.print(" Hello world!");  // Τυπώνει ένα μήνυμα στη μονάδα LCD.

     lcd.setCursor(0, 1);         // μετακινεί τον κέρσορα στη στήλη 0, και στη γραμμή 1
     lcd.print(" LCD Tutorial"); // Τυπώνει ένα μήνυμα στη LCD.

    while(1);
 }
```

Δημιουργία προσαρμοσμένου χαρακτήρα MicroC




```
PORTD =0b01001000; //Set CGRAM  01001000 καθορίζει τη διεύθυνση που θα χρησιμοποιηθεί στην CGRAM
PORTE.F2 = 0;            //RS = 0 σημαίνει ότι επιλέγουμε να γράψουμε στην CGRAM (1 στην DDRAM)
PORTE.F1=1;              // pulse to enable η LCD "διαβάζει" τα δεδομένα ή τις εντολές που στέλνονται μέσω του PORTD
PORTE.F1=0;

PORTD =0b00011111; //Write 11111 to 1st row
PORTE.F2 = 1;            //RS = 1 to write to address specified above.
PORTE.F1=1;              // pulse to enable
PORTE.F1=0;
.
...
…

PORTD = 0b10001000; //Set DDRAM ADDRESS to Specify position in display.
PORTE.F2 = 0;
PORTE.F1=1;        // pulse to enable
PORTE.F1=0;

PORTD =  0b00000001;   //write from CCGRAM 0  001;
PORTE.F2 = 1;
PORTE.F1=1;        // pulse to enable
PORTE.F1=0;
```

Δημιουργία προσαρμοσμένου χαρακτήρα



Τέλος 3ης διάλεξης

Δρ. Δημήτρης Ζιούζιος

dziouzios@uowm.gr