

## Ψηφιακά Συστήματα Μετρήσεων

Arduino # Raspberry Pi  
Processing # Python # MATLAB

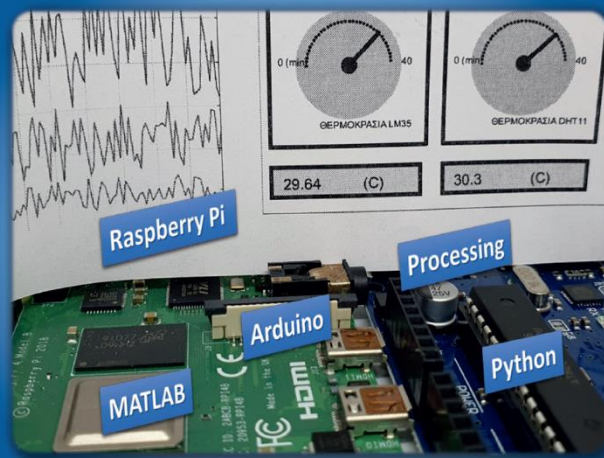
Το βιβλίο αυτό «απαντά» στις πραγματικές ανάγκες ενός μηχανικού, στην κατεύθυνση της ανάπτυξης ενός λειτουργικού και ολοκληρωμένου ψηφιακού συστήματος μετρήσεων. Παρουσιάζει τις επικρατέστερες τεχνολογίες και μεθοδολογίες για την ανάπτυξη των εφαρμογών. Δηλαδή, συγκεντρώνει γνώσεις με μια ενιαία αντίληψη και όχι αποσπασματικά, όπως έχουν συνηθίσει οι μηχανικοί σήμερα, που καλούνται να ανατρέξουν σε τελείως διαφορετικά βιβλία μεταξύ τους, μη γνωρίζοντας ακριβώς ποια εργαλεία θα πρέπει να χρησιμοποιήσουν.

Το βιβλίο είναι εστιασμένο στην ανάπτυξη λογισμικού, εκεί δηλαδή που βρίσκεται ο πυρήνας των σύγχρονων ψηφιακών συστημάτων μέτρησης.

Το βιβλίο συνοδεύεται από πλούσιο υλικό που είναι διαθέσιμο στην ιστοσελίδα [panosparazoglou.gr](http://panosparazoglou.gr)

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΖΟΓΛΟΥ

## Ψηφιακά Συστήματα Μετρήσεων



# Διαφάνειες

# Υλικό βιβλίου

Περισσότερο υλικό στο  
[panosparazoglou.gr](http://panosparazoglou.gr)

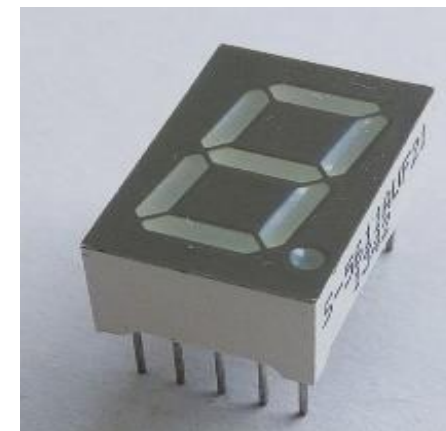
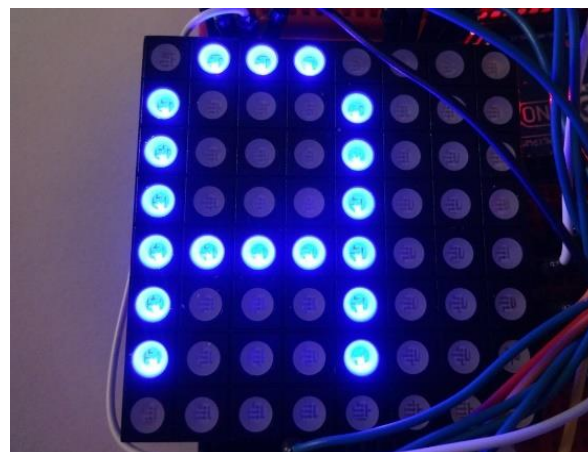
# Κεφάλαιο 6



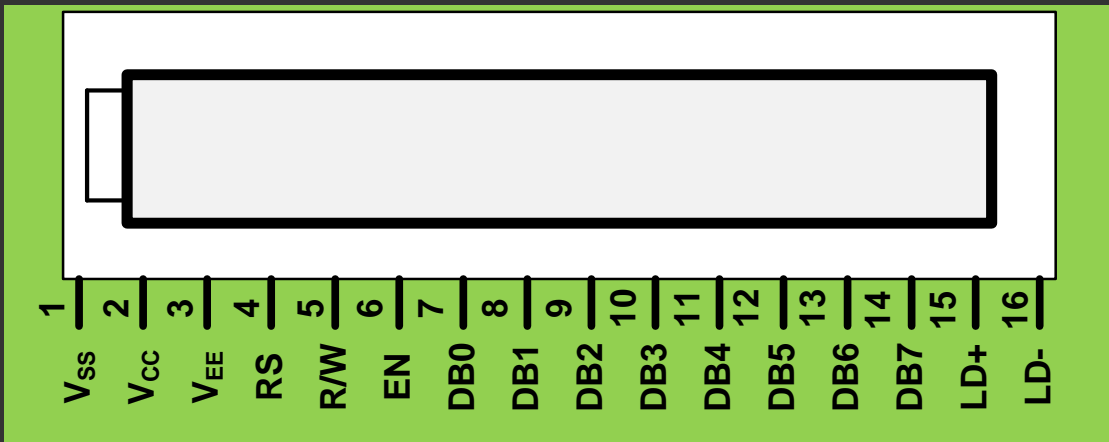
# Απεικόνιση μετρήσεων σε ηλεκτρονικές διατάξεις



# Διασύνδεση με μονάδα απεικόνισης



# Οθόνη LCD (1)



Αριθμός ακροδέκτη	Ονομασία	Χρήση
1	$V_{SS}$	Γείωση
2	$V_{CC}/V_{DD}$	+5V (τροφοδοσία)
3	$V_{EE}/V_0$	Έλεγχος αντίθεσης (contrast)
4	RS	Επιλογή καταχωρητή (0=καταχωρητής εντολών, 1= καταχωρητής δεδομένων)
5	R/W	Επιλογή εγγραφής ή ανάγνωσης (0=εγγραφή, 1=ανάγνωση)
6	EN	Ενεργοποίηση λειτουργίας (Enable)
7-14	DB0-DB7	Γραμμές δεδομένων (Data Bus)
15	LD+	+5V τροφοδοσία για οπίσθιο φωτισμό*
16	LD-	Γείωση για οπίσθιο φωτισμό*

\* Οι οθόνες LCD που δεν υποστηρίζουν οπίσθιο φωτισμό, διαθέτουν 14 ακροδέκτες



# Οθόνη LCD (2)

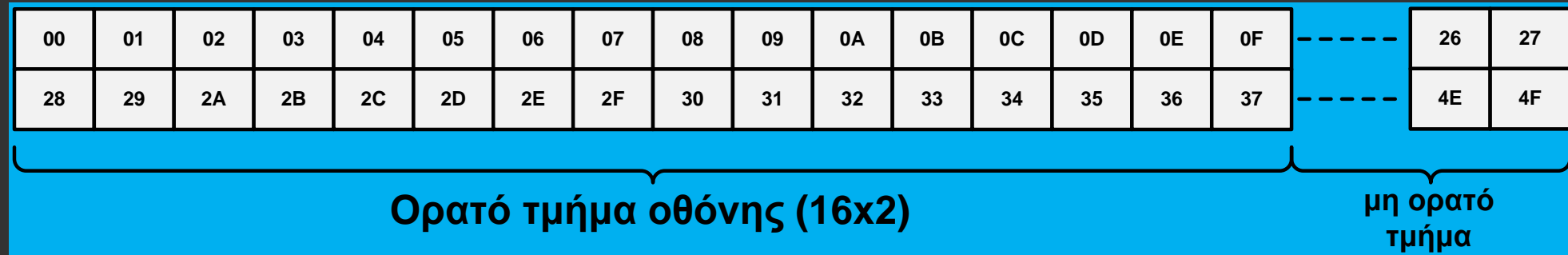
## Καταχωρητές

**DR** – Data Register (καταχωρητής δεδομένων). Δεδομένα που σχετίζονται με τη μνήμη της οθόνης (ό,τι εμφανίζεται σε αυτή)

**IR** – Instruction Register (καταχωρητής εντολών). Έλεγχος λειτουργίας

# Οθόνη LCD (3)

## DDRAM οθόνης



- Πρώτη γραμμή (διευθύνσεις 00 έως 0F)
- Δεύτερη γραμμή (διευθύνσεις 28 έως 37)

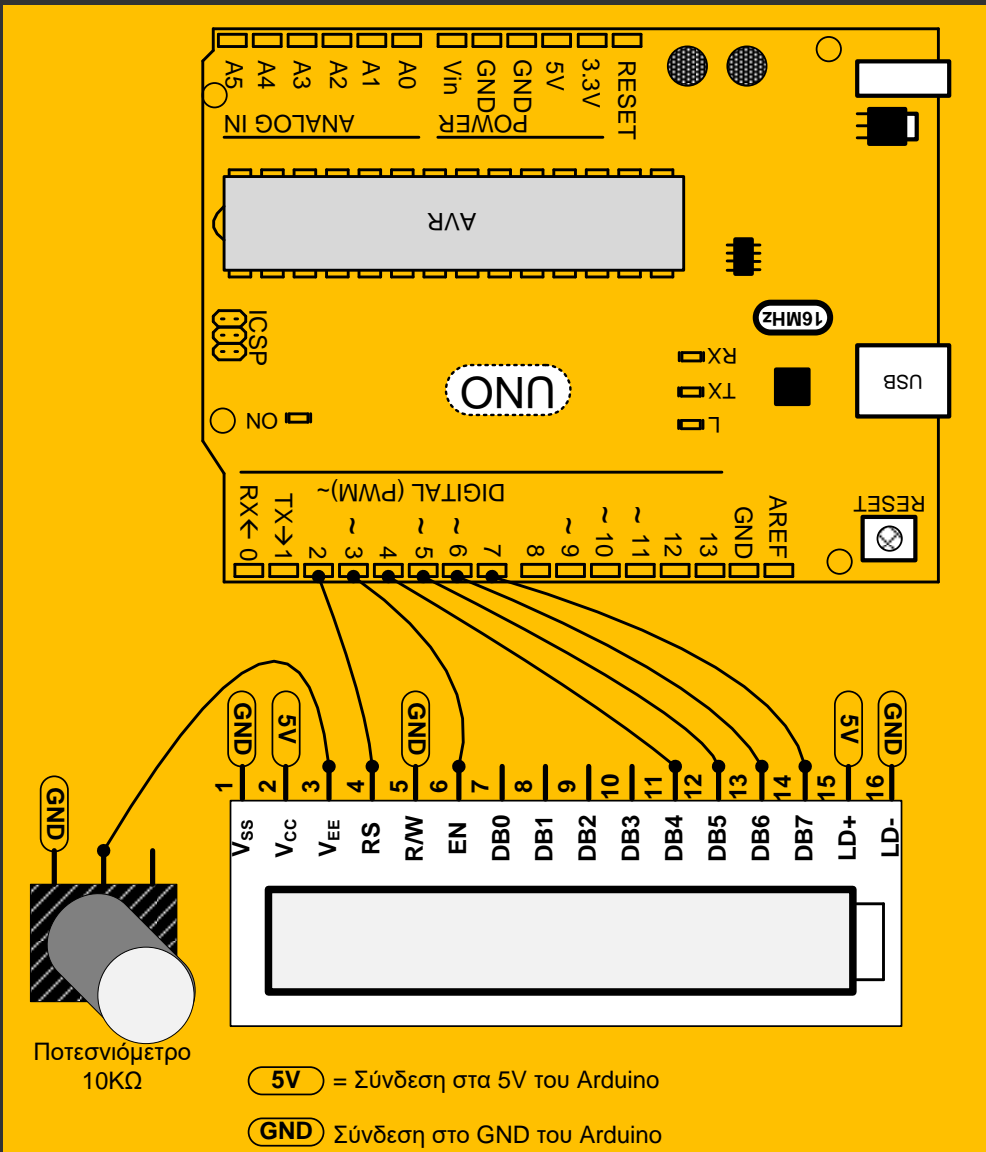
# Οθόνη LCD (4)

## Ενδεικτικές λειτουργίες

Κώδικας (δεκαεξαδικό)	Περιγραφή λειτουργίας
01	Καθαρισμός οθόνης
04	Μετακίνηση δρομέα αριστερά
06	Μετακίνηση δρομέα δεξιά
05	Ολίσθηση περιεχομένου οθόνης δεξιά
07	Ολίσθηση περιεχομένου οθόνης αριστερά
08	Σβήσιμο οθόνης, σβήσιμο δρομέα
0A	Σβήσιμο οθόνης, ενεργοποίηση δρομέα
0C	Ενεργοποίηση οθόνης, σβήσιμο δρομέα
0E	Ενεργοποίηση οθόνης, αναβόσβησμα δρομέα
10	Μετακίνηση δρομέα αριστερά
14	Μετακίνηση δρομέα δεξιά
80	Μετακίνηση δρομέα στην αρχή της πρώτης γραμμής
C0	Μετακίνηση δρομέα στην αρχή της δεύτερης γραμμής



# LCD – Arduino



```
#include <LiquidCrystal.h>

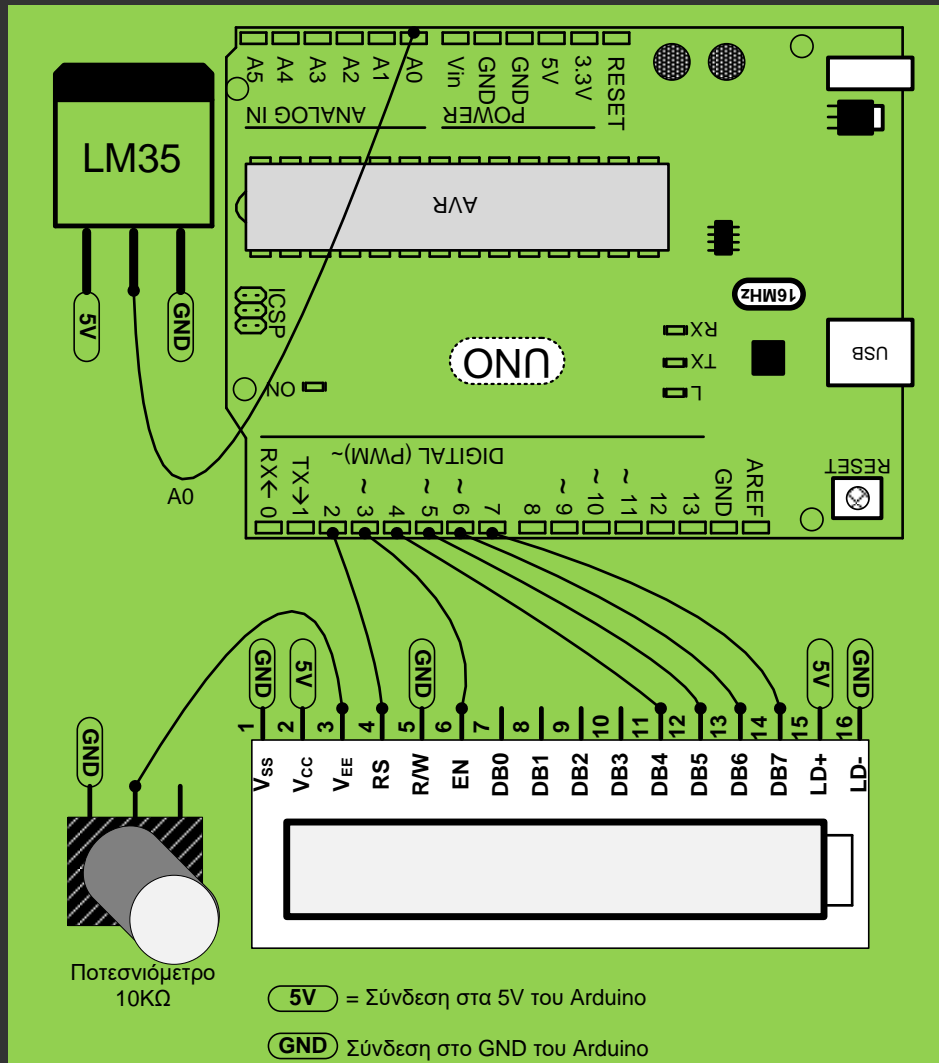
LiquidCrystal lcd(2,3,4,5,6,7);

void setup()
{
  lcd.begin(16, 2);
  lcd.print("Hello, LCD16x2");
}

void loop()
{
  lcd.setCursor(0, 1);
  lcd.print("Seconds:");
  lcd.print(millis()/1000);
}
```

HELLO, LCD16X2

# LCD – Arduino - Αισθητήρας



```
#include <LiquidCrystal.h>  
float Q=1.074;  
LiquidCrystal lcd(2,3,4,5,6,7);
```

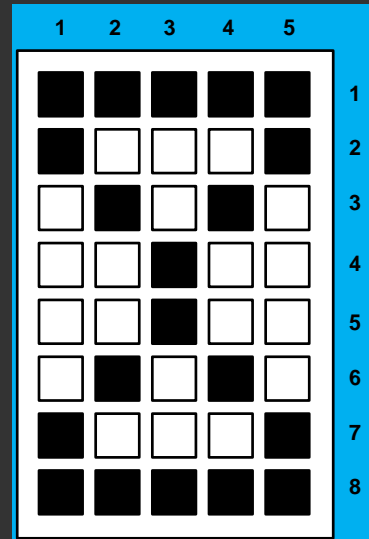
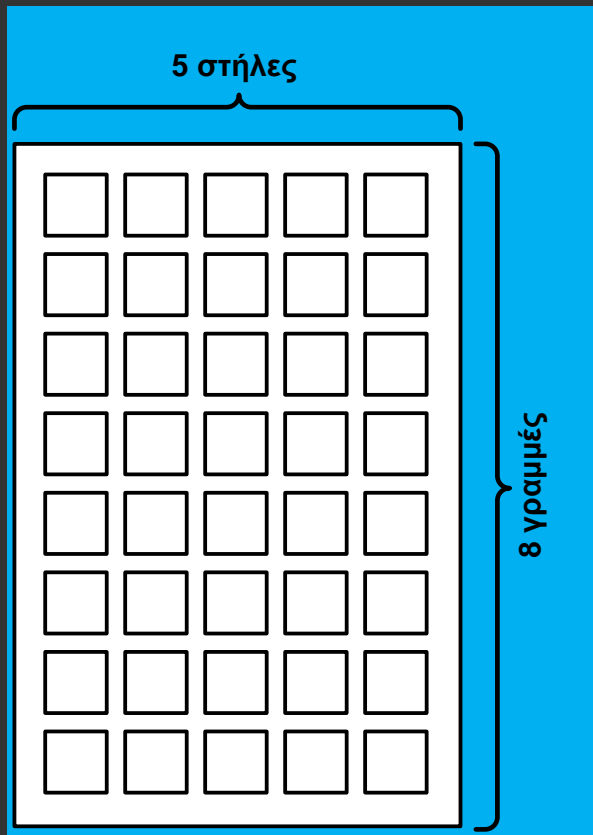
```
void setup()
```

```
{  
  analogReference(INTERNAL);  
  lcd.begin(16, 2);  
  lcd.print("Thermokrasia");  
  lcd.setCursor(6,1);  
  lcd.print("C");  
}
```

```
void loop()
```

```
{  
  lcd.setCursor(0, 1);  
  float temp=(analogRead(0)*Q)/10;  
  lcd.print(temp);  
  delay(500);  
}
```

# LCD – Δημιουργία συμβόλων (1)



Αριθμός γραμμής	Κωδικοποίηση
1	xxx11111
2	xxx10001
3	xxx01010
4	xxx00100
5	xxx00100
6	xxx01010
7	xxx10001
8	xxx11111



```
byte newsymbol[8]={0b11111, 0b10001, 0b01010,  
0b00100, 0b00100, 0b01010, 0b10001, 0b11111};
```

# LCD – Δημιουργία συμβόλων (2)

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
byte newsymbol[8]={0b11111, 0b10001, 0b01010, 0b00100, 0b00100,
0b01010, 0b10001, 0b11111};

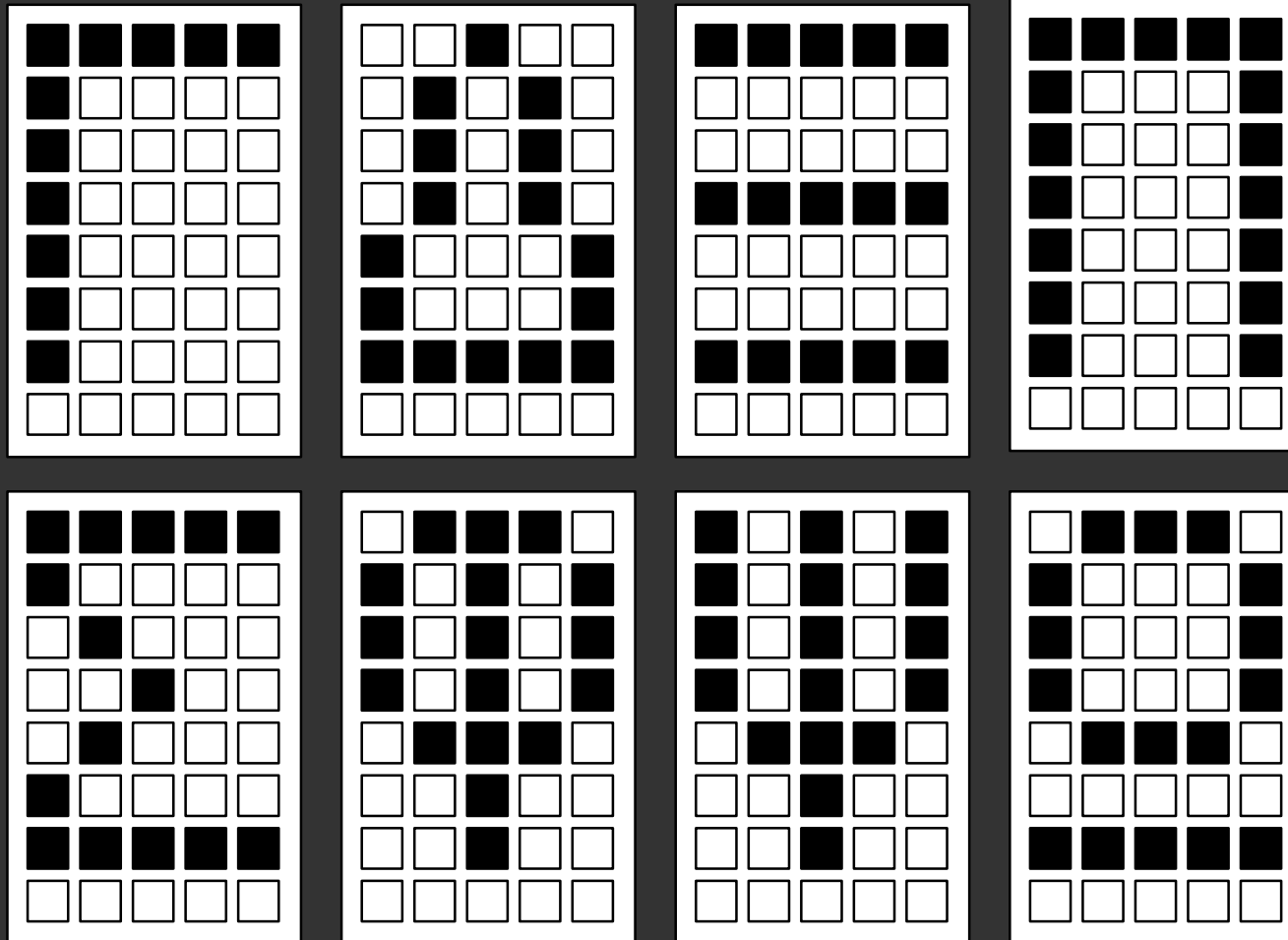
void setup()
{
  lcd.createChar(1,newsymbol);
  lcd.begin(16, 2);

  lcd.write(1);
}

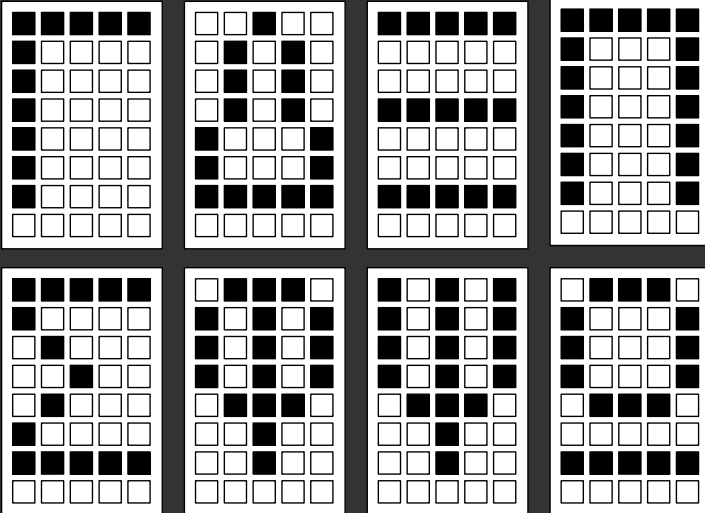
void loop()
{
}
}
```



# LCD – Ελληνικά σύμβολα (1)



# LCD – Ελληνικά σύμβολα (2)



Σύμβολο	Κωδικοποίηση
‘Γ’	11111, 10000, 10000, 10000, 10000, 10000, 10000, 00000
‘Δ’	00100, 01010, 01010, 01010, 10001, 10001, 11111, 00000
‘Ξ’	11111,00000,00000,11111,00000,00000, 11111, 00000
‘Π’	11111, 10001, 10001, 10001, 10001, 10001, 10001, 00000
‘Σ’	11111,10000, 01000, 00100, 01000,10000,11111, 00000
‘Φ’	01110, 10101, 10101, 10101,01110,00100, 00100, 00000
‘Ψ’	10101, 10101, 10101, 10101, 01110, 00100, 00100, 00000
‘Ω’	01110, 10001, 10001, 10001, 01110,00000,11111,00000

# LCD – Ελληνικά σύμβολα (3)

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
byte newsymbol[8][8]=
{
{0b11111, 0b10000, 0b10000, 0b10000, 0b10000, 0b10000,
0b10000, 0b00000},
{0b00100, 0b01010, 0b01010, 0b01010, 0b10001, 0b10001,
0b11111, 0b00000},
{0b11111, 0b00000, 0b00000, 0b11111, 0b00000, 0b00000,
0b11111, 0b00000},
{0b11111, 0b10001, 0b10001, 0b10001, 0b10001, 0b10001,
0b10001, 0b00000},
{0b11111, 0b10000, 0b01000, 0b00100, 0b01000, 0b10000,
0b11111, 0b00000},
{0b01110, 0b10101, 0b10101, 0b10101, 0b01110, 0b00100,
0b00100, 0b00000},
{0b10101, 0b10101, 0b10101, 0b10101, 0b01110, 0b00100,
0b00100, 0b00000},
{0b01110, 0b10001, 0b10001, 0b10001, 0b01110, 0b00000,
0b11111, 0b00000}
};
```

```
void setup()
{
  for(int i=0;i<8;i++)
    lcd.createChar(i+1,newsymbol[i]);

  lcd.begin(16, 2);
  lcd.print("new symbols:");
  for (int i=1; i<=8; i++)
  {
    lcd.setCursor(i,1); lcd.write(i);
  }
}

void loop()
{
}
```

new symbols

# LCD – Ελληνικά σύμβολα (4)

```
#include <LiquidCrystal.h>
float Q=1.074;
LiquidCrystal lcd(2,3,4,5,6,7);
byte sigma[8]={0b11111, 0b10000, 0b01000, 0b00100,
0b01000, 0b10000, 0b11111, 0b00000};

void setup()
{
  analogReference(INTERNAL);
  lcd.createChar(1,sigma);

  lcd.begin(16, 2);

  lcd.print("8EPMOKPA"); lcd.write(1); lcd.print("IA");
  lcd.setCursor(6,1); lcd.write(223); lcd.print("C");
}
```

```
void loop()
{

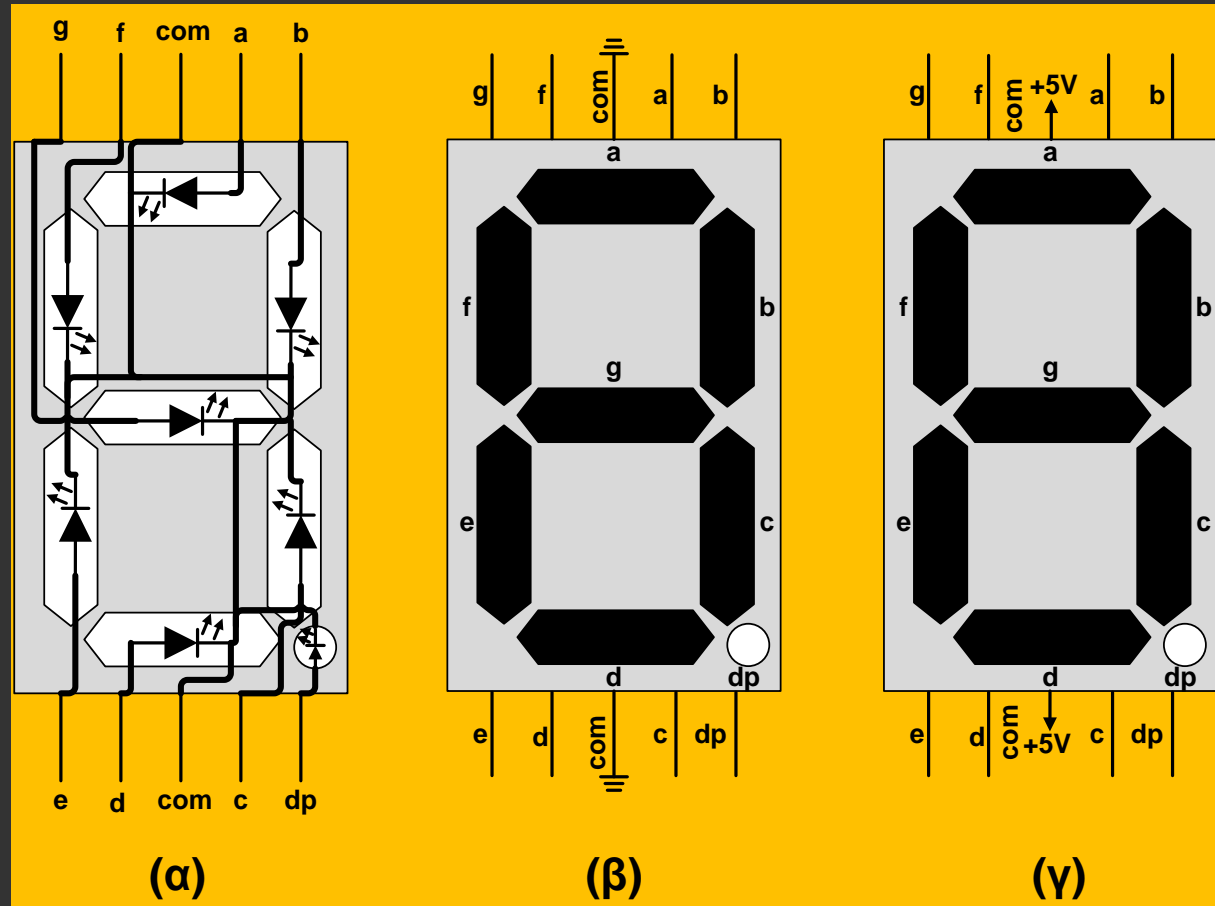
  lcd.setCursor(0, 1);
  float temp=(analogRead(0)*Q)/10;
  lcd.print(temp);
  delay(500);

}
```

8EPMOKPA  
223.45 °C

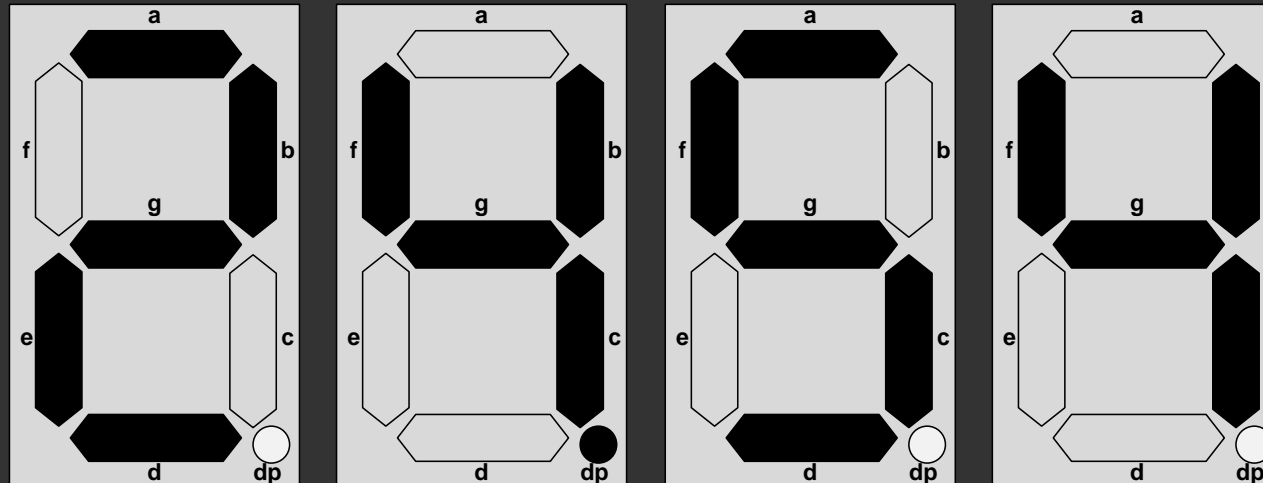


# Μονάδα επτά τμημάτων (1)



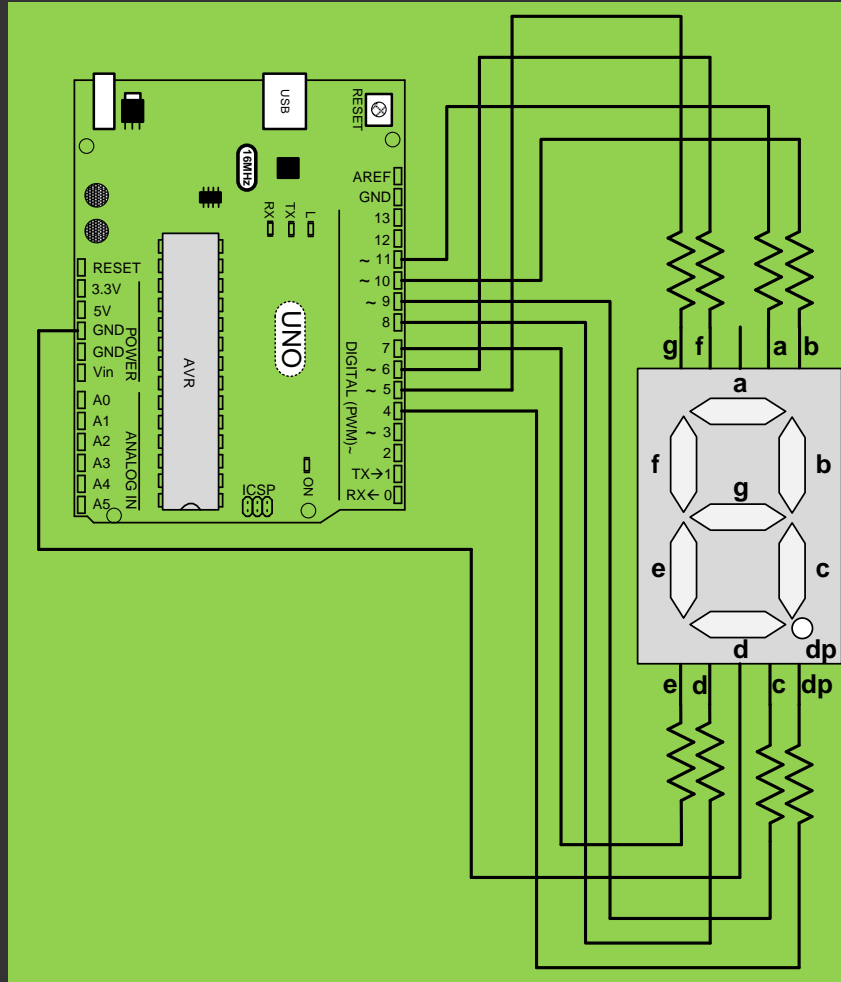
# Μονάδα επτά τμημάτων (2)

## Παράδειγμα



Αριθμός μονάδας (από αριστερά)	Ψηφίο που απεικονίζεται	Πλήθος ενεργοποιημένων LED (τμημάτων)	Σύνολο κατανάλωσης ρεύματος (ενδεικτικά)
1 <sup>η</sup>	2	5	$5 \times 15\text{mA} = 75\text{mA}$
2 <sup>η</sup>	4	5	$5 \times 15\text{mA} = 75\text{mA}$
3 <sup>η</sup>	5	5	$5 \times 15\text{mA} = 75\text{mA}$
4 <sup>η</sup>	4	4	$4 \times 15\text{mA} = 60\text{mA}$
			Σύνολο 275mA

# Έλεγχος μονάδας (1)



Ακροδέκτης Arduino	Ακροδέκτης μονάδας 7SD
11	a
10	b
9	c
8	d
7	e
6	f
5	g
4	dp
GND	Γείωση

# Έλεγχος μονάδας (2)

```
byte hex[16]=
{
//από αριστερά προς τα δεξιά
//d p g f e d c b a
  0b00111111, //0
  0b00000110, //1
  0b01011011, //2
  0b01001111, //3
  0b01100110, //4
  0b01101101, //5
  0b01111101, //6
  0b00000111, //7
  0b01111111, //8
  0b01101111, //9
  0b01110111, //A
  0b01111100, //B
  0b01011000, //C
  0b01011110, //D
  0b01111001, //E
  0b01110001, //F
};
```

```
void setup()
{
  int i;
  for (i=4;i<=11;i++)
    pinMode(i,OUTPUT);
}

void loop()
{
  int i,j;
  int d;
  for (i=0;i<=15;i++)
  {
    for (j=7;j>=0;j--)
      digitalWrite(11-j,bitRead(hex[i],j));

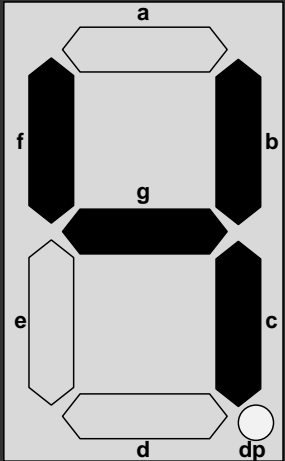
    delay(1000);
  }
}
```

# Εμφάνιση ψηφίων

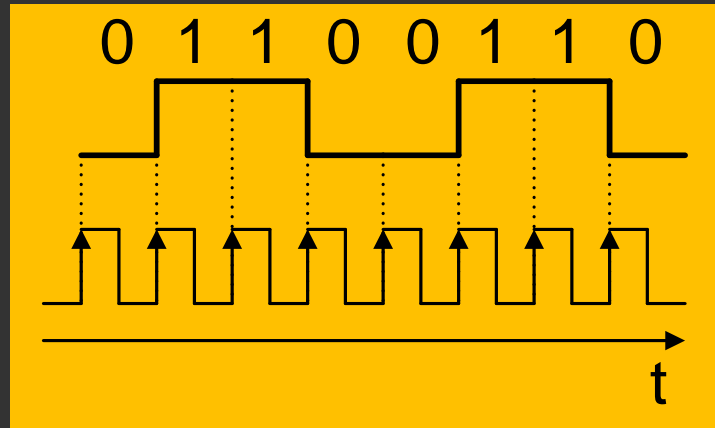


# Καταχωρητής ολίσθησης (1)

## Παράδειγμα

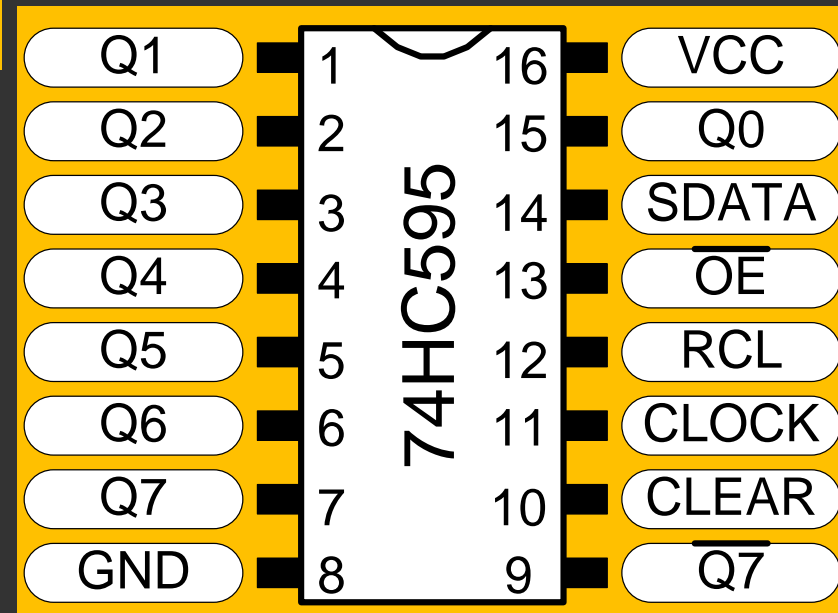
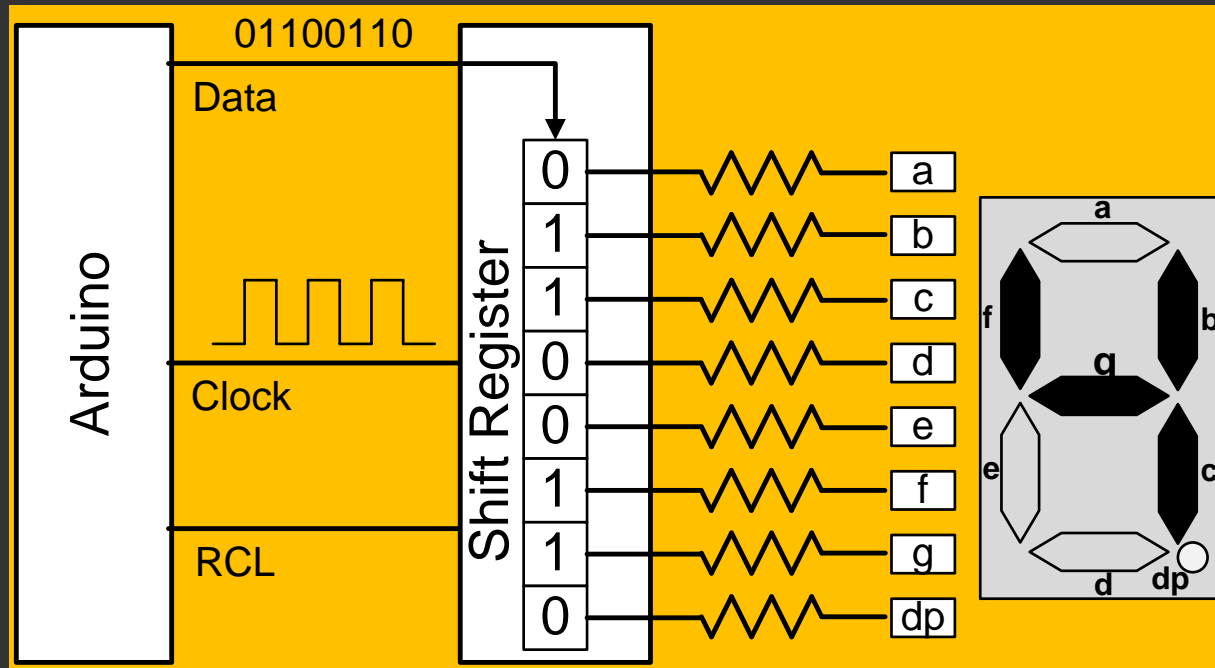


dp	g	f	e	d	c	b	a
0	1	1	0	0	1	1	0

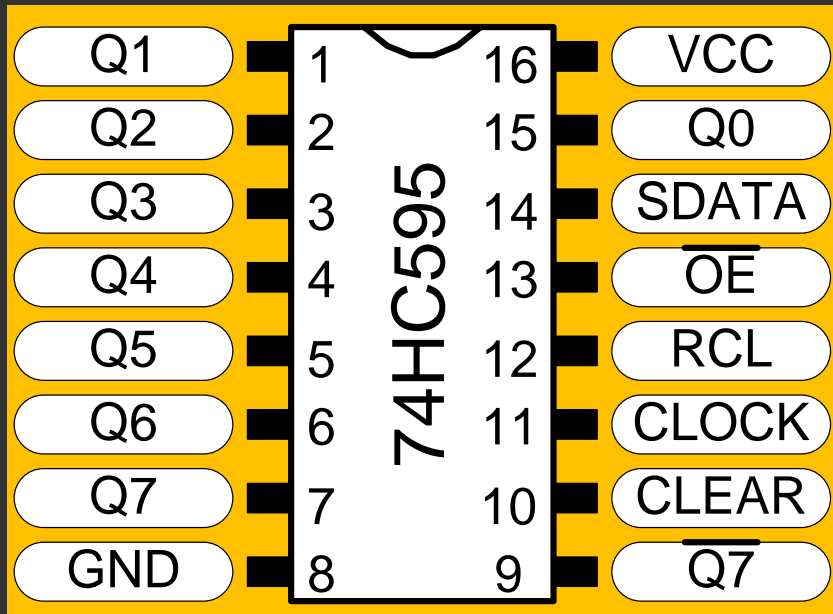


Παλμός Clock	Περιεχόμενο καταχωρητή ολίσθησης (bit)							
	b7	b6	b5	b4	b3	b2	b1	b0
Αρχικό περιεχόμενο (άγνωστο)	X	X	X	X	X	X	X	X
1 <sup>ος</sup>	X	X	X	X	X	X	X	0
2 <sup>ος</sup>	X	X	X	X	X	X	0	1
3 <sup>ος</sup>	X	X	X	X	X	0	1	1
4 <sup>ος</sup>	X	X	X	X	0	1	1	0
5 <sup>ος</sup>	X	X	X	0	1	1	0	0
6 <sup>ος</sup>	X	X	0	1	1	0	0	1
7 <sup>ος</sup>	X	0	1	1	0	0	1	1
8 <sup>ος</sup>	0	1	1	0	0	1	1	0

# Καταχωρητής ολίσθησης (2)



# Καταχωρητής ολίσθησης (3)



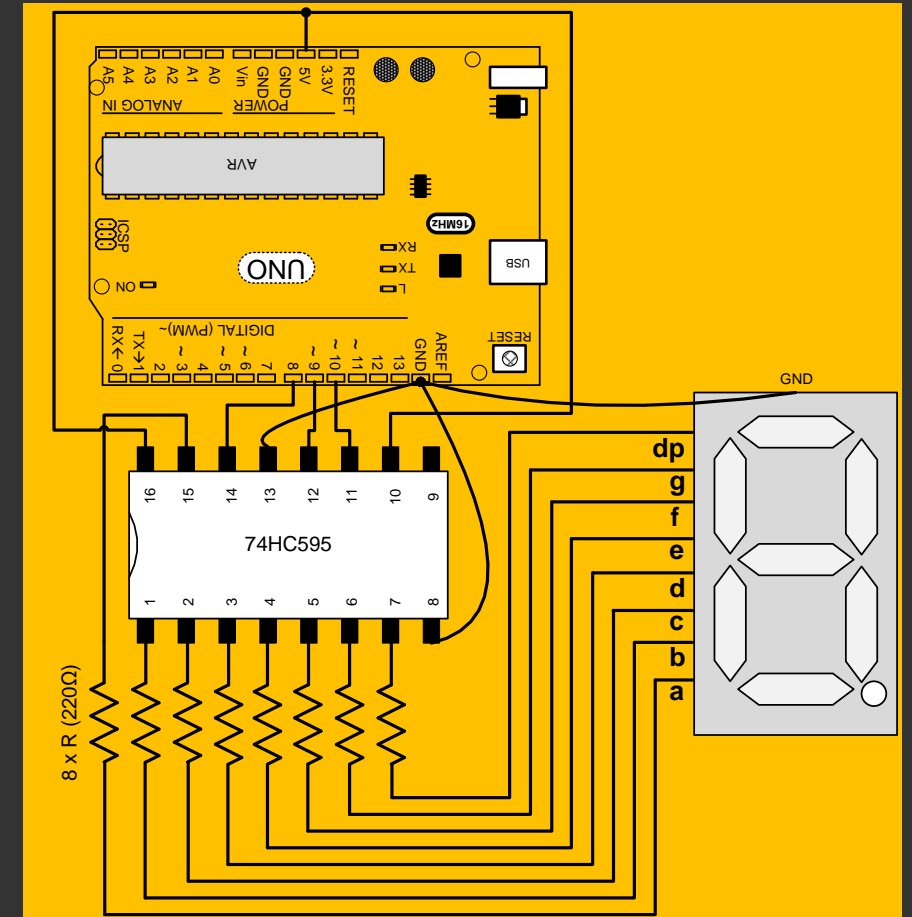
Αριθμός ακροδέκτη	Ονομασία	Λειτουργία
1-7, 15	Q0-Q7	Το 8bit περιεχόμενο του καταχωρητή ολίσθησης, διαθέσιμο σε 8 αντίστοιχους ακροδέκτες
8	GND	Γείωση
9	Q7 (ενεργοποίηση στο μηδέν)	Χρησιμοποιείται σε περίπτωση διασύνδεσης με επόμενο κύκλωμα 74HC595
10	CLEAR	Reset περιεχομένου
11	CLOCK	Σήμα ρολογιού για σειριακή προώθηση ψηφίο προς ψηφίο
12	RCL	Επιλογή «εμφάνισης» της εξόδου
13	OE (ενεργοποίηση στο μηδέν)	Ενεργοποίηση των ακροδεκτών εξόδου (αλλιώς βρίσκονται σε κατάσταση υψηλής αντίστασης)
14	SDATA	«Παροχή» σειριακών δεδομένων για το «γέμισμα» του καταχωρητή ολίσθησης
16	VCC	Τροφοδοσία (+5V)



# Χρήση Καταχωρητή ολίσθησης

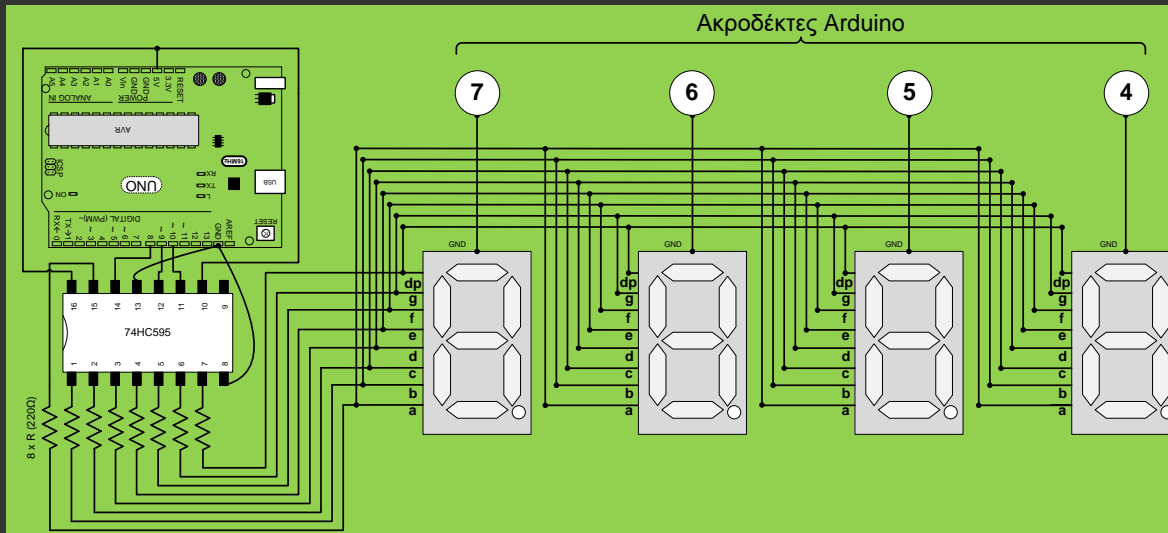
```
#define Data 8
#define Clock 10
#define L 9
byte hex[16]=
{
  0b00111111, //0
  0b00000110, //1
  0b01011011, //2
  0b01001111, //3
  0b01100110, //4
  0b01101101, //5
  0b01111101, //6
  0b00000111, //7
  0b01111111, //8
  0b01101111, //9
  0b01110111, //A
  0b01111100, //B
  0b01011000, //C
  0b01011110, //D
  0b01111001, //E
  0b01110001, //F
};
```

```
void setup()
{
  pinMode(Data,OUTPUT);
  pinMode(Clock,OUTPUT);
  pinMode(L, OUTPUT);
}
void loop()
{
  int i;
  for (i=0;i<=15;i++)
  {
    digitalWrite(L,LOW);
    shiftOut(Data,Clock, MSBFIRST, hex[i]);
    digitalWrite(L,HIGH);
    delay(1000);
  }
}
```



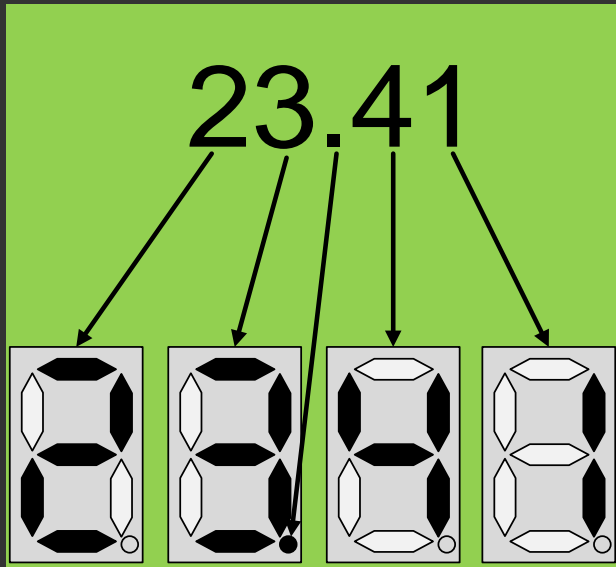


# Έλεγχος – Χρονική πολυπλεξία (2)

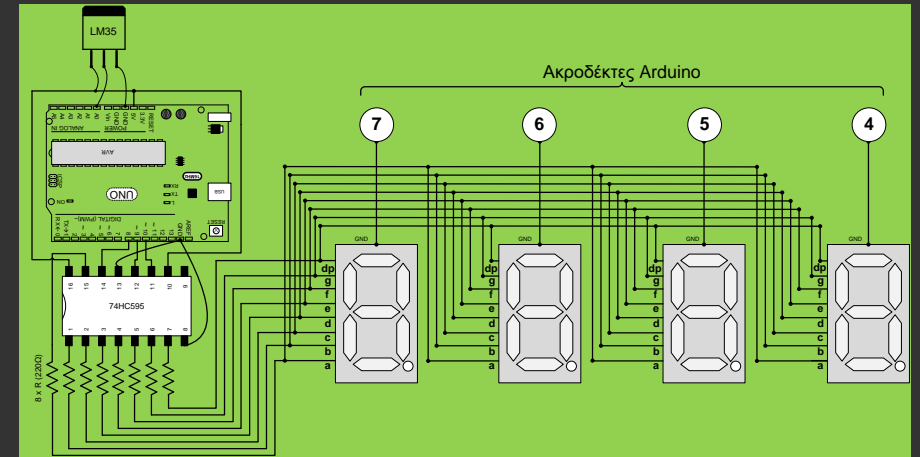
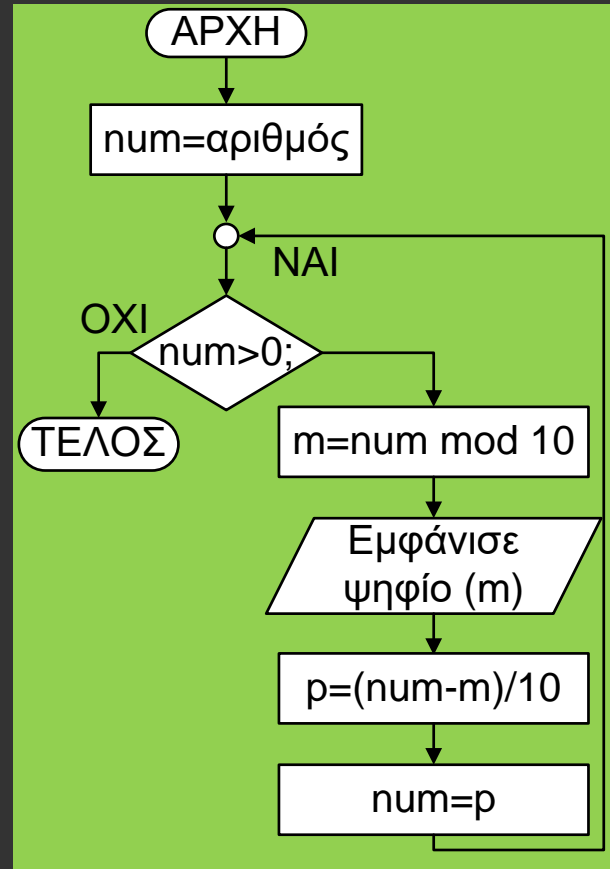
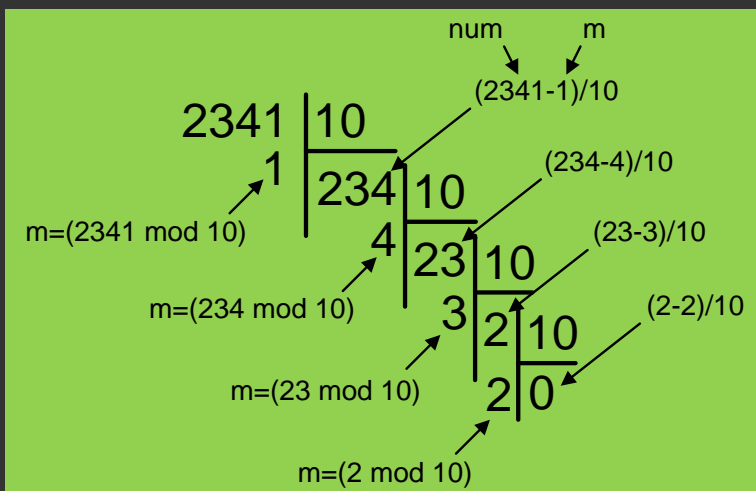


Ακροδέκτης 74HC595	Ακροδέκτης Arduino
10,16	5V
11	10
12	9
8,13	GND
14	8
	Μονάδες 7SD (κοινοί σε όλες τις μονάδα)
15	a
1	b
2	c
3	d
4	e
5	f
6	g
7	dp
Ακροδέκτης Arduino	
7	Γείωση 1 <sup>ης</sup> μονάδας 7SD
6	Γείωση 2 <sup>ης</sup> μονάδας 7SD
5	Γείωση 3 <sup>ης</sup> μονάδας 7SD
4	Γείωση 4 <sup>ης</sup> μονάδας 7SD

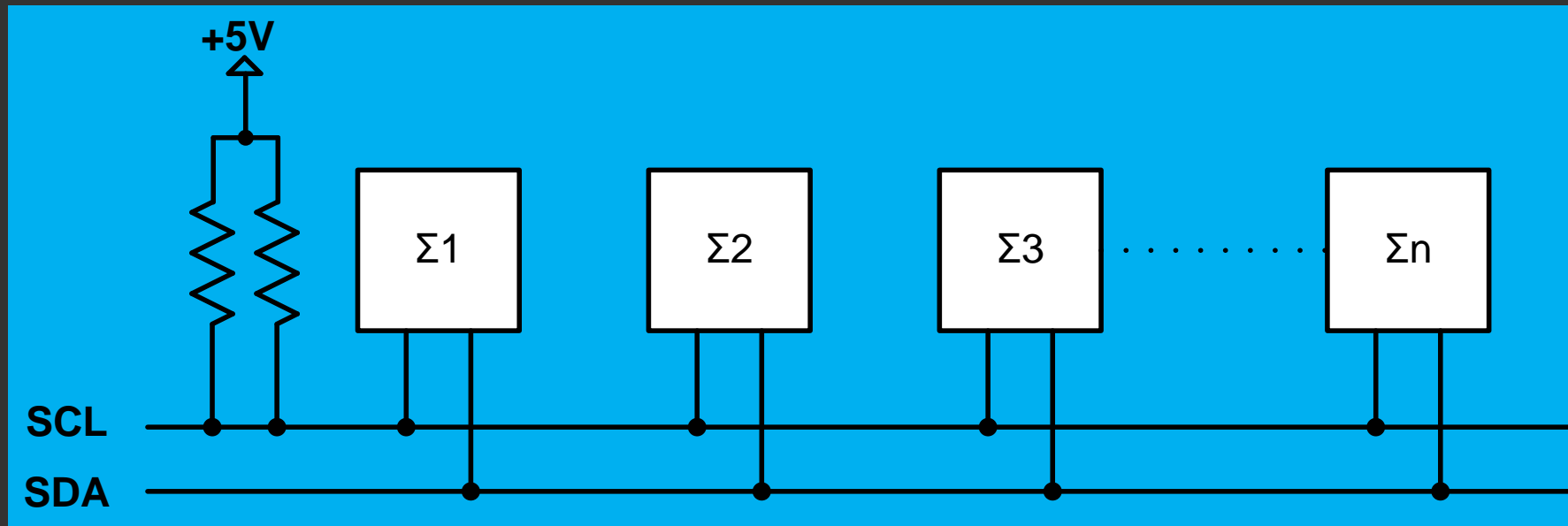
# Εμφάνιση αριθμού – Χρονική πολυπλεξία (1)



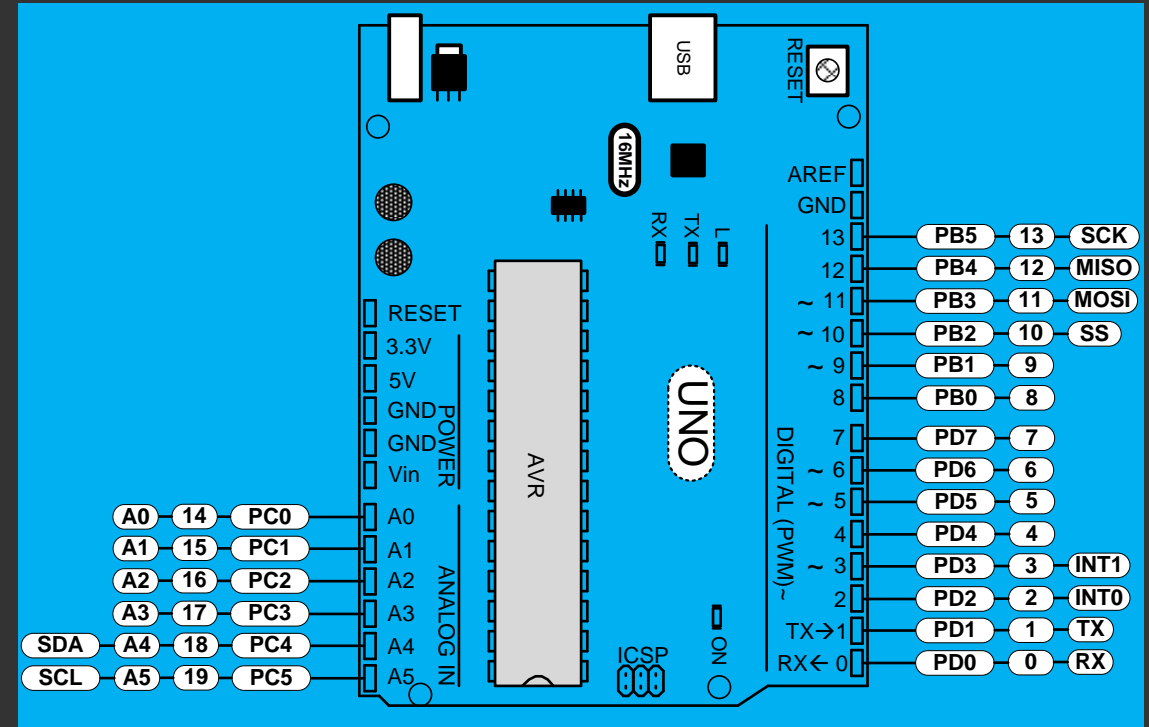
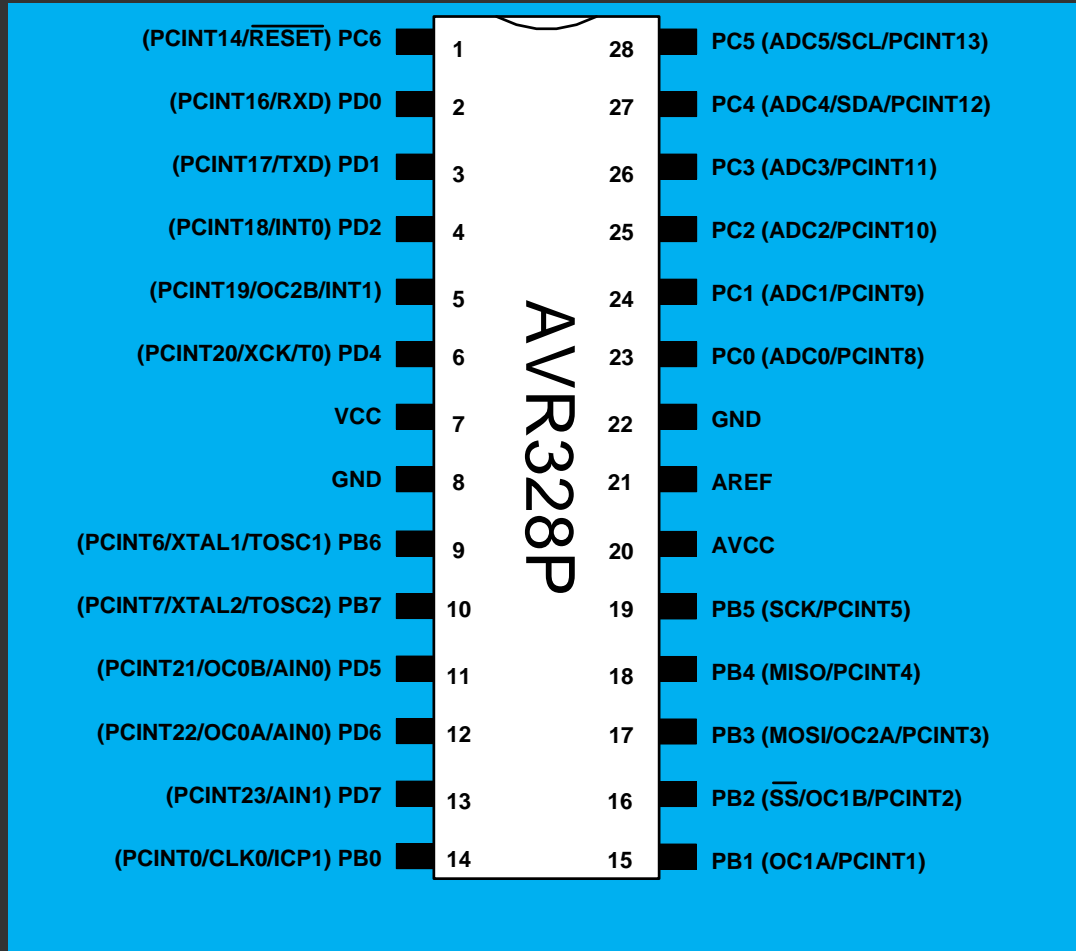
## Εξαγωγή ψηφίων



# LCD – I<sup>2</sup>C (1)

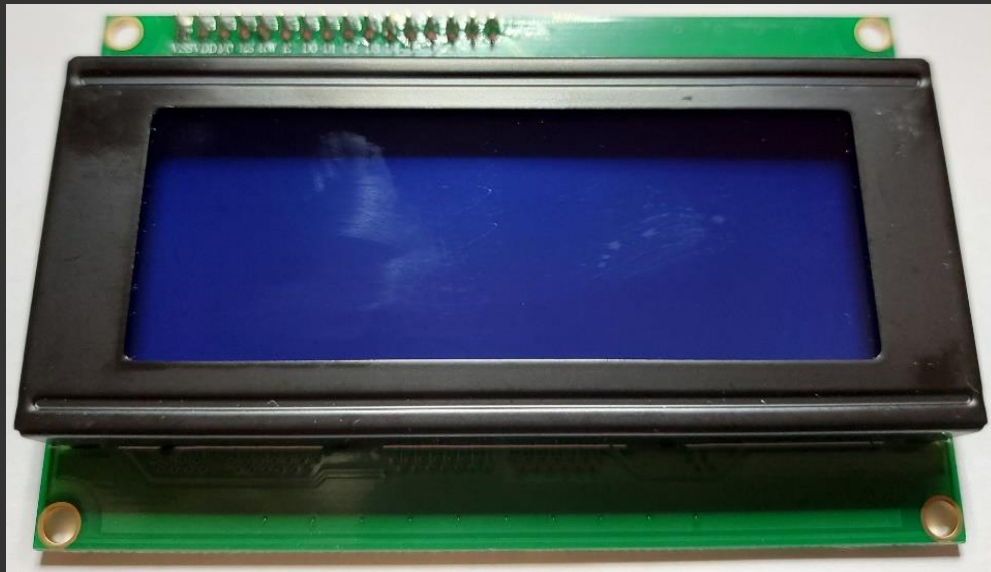
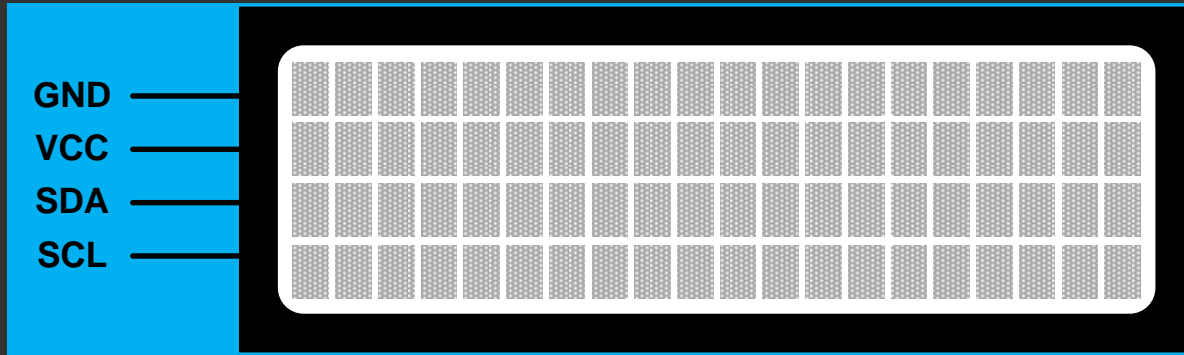


# LCD – I<sup>2</sup>C (2)



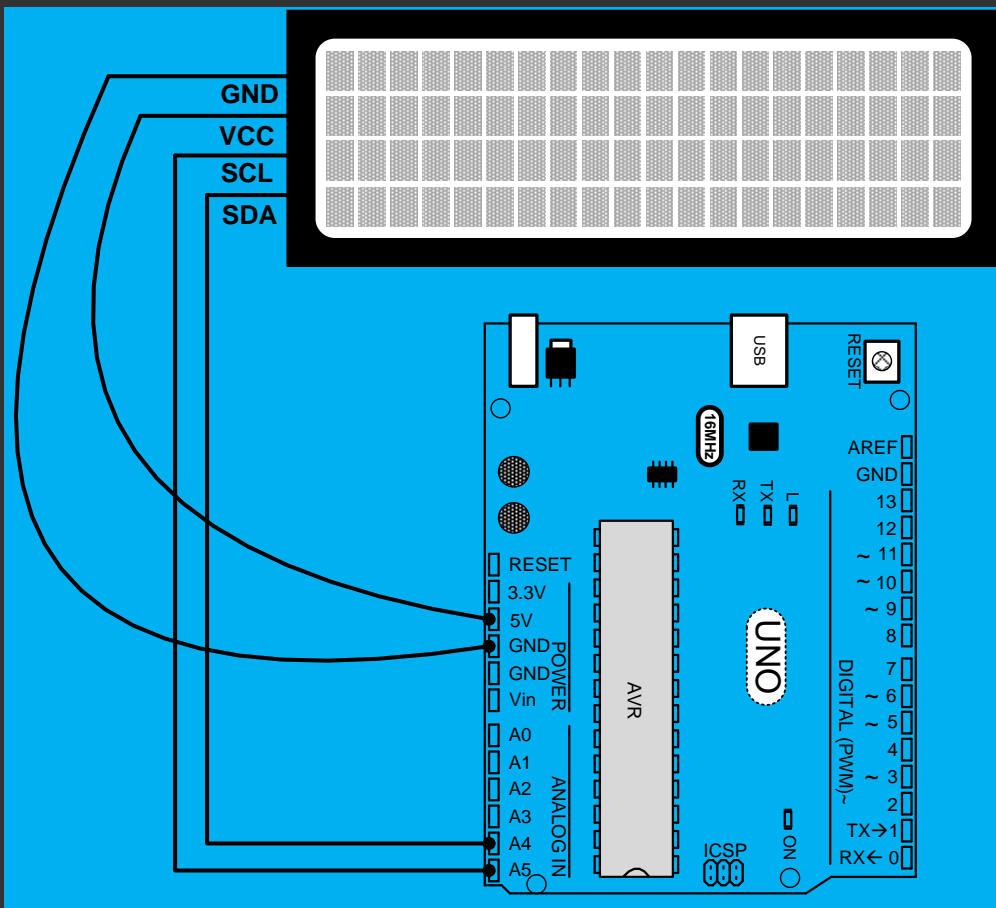
ακροδέκτες μικροελεγκτή στην πλακέτα Arduino / διπλή λειτουργία

# LCD - I<sup>2</sup>C (3)



# LCD – I<sup>2</sup>C (4)

## Εύρεση διεύθυνσης



```
#include <Wire.h>
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  Serial.println("START\n");
}
```

```
void loop()
{
  byte comstatus, addr;
  int numdev;
  Serial.println("Searching...");
  numdev = 0;
  for (addr = 1; addr < 127; addr++ )
  {
    Wire.beginTransmission(addr);
    comstatus = Wire.endTransmission();
    if (comstatus == 0)
    {
      Serial.print("Device found : address : 0x");
      Serial.print(addr, HEX);
      numdev++;
    }
  }
  if (numdev == 0)
    Serial.println("Device not found\n");

  Serial.println("\nEND!\n");
  delay(2000);
}
```



# LCD – I<sup>2</sup>C (5)

## Δοκιμή



```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3f,20,4);

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.setCursor(7,0);
  lcd.print("HELLO");
  lcd.setCursor(7,1);
  lcd.print("I2C LCD");
  lcd.setCursor(7,2);
  lcd.print("20x4");
  lcd.setCursor(7,3);
  lcd.print("TEST!!!"); }

void loop()
{
}
```

