# Σύνδεση περιφερειακών στο Arduino

# Connecting Arduino to a PIR Sensor

## PIR sensor

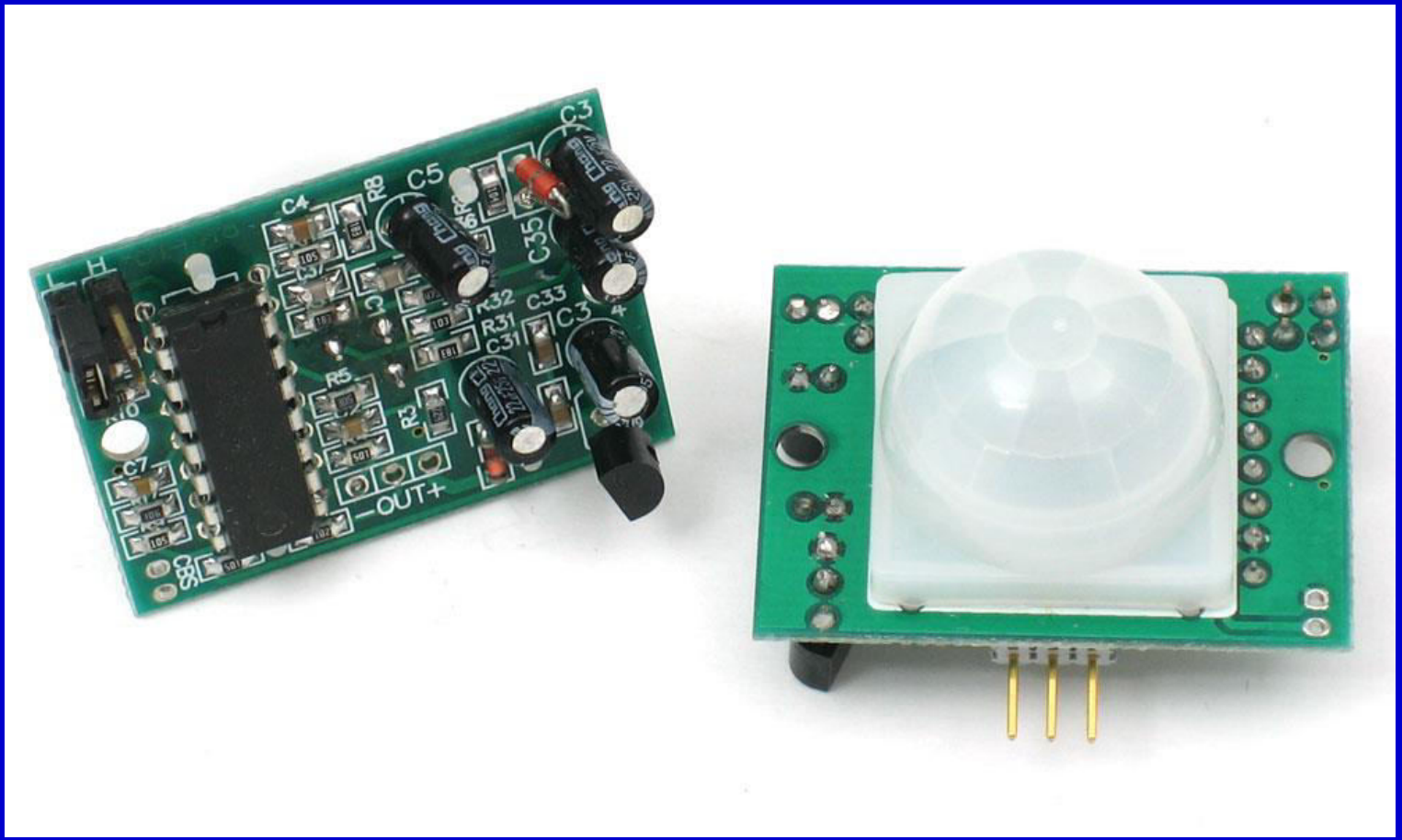
A **passive infrared sensor** (**PIR sensor**) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors.

# Αισθητήρας Ανίχνευσης Κίνησης HC-SR501

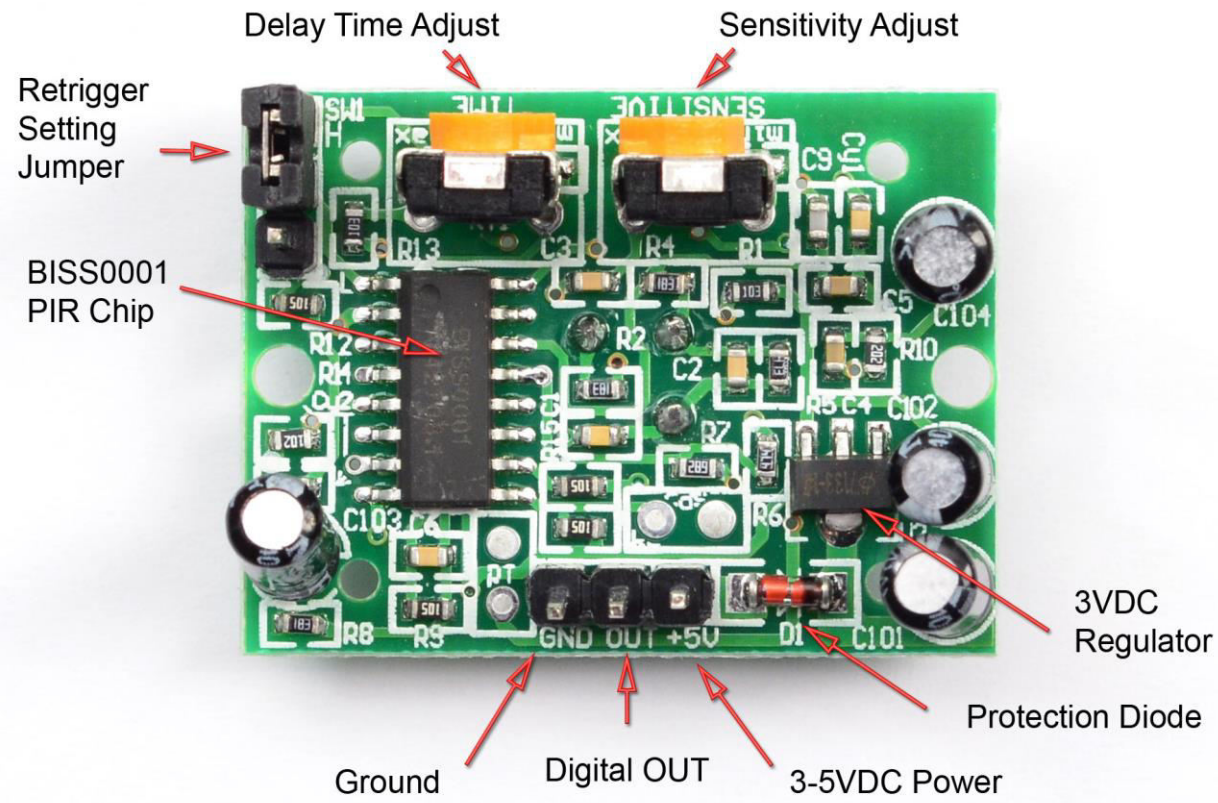
**Τεχνικά Χαρακτηριστικά**:

Τάση Εισόδου 5 - 12 V DC
Κατανάλωση <60uA
Σήμα Εξόδου 3.3V TTL
Απόσταση Ανίχνευσης 3-6 μέτρα
Γωνία Ανίχνευσης <140°
Χρόνος ενεργοποίησης 5 - 200 sec
Χρόνος καθυστέρησης 2 - 4 sec

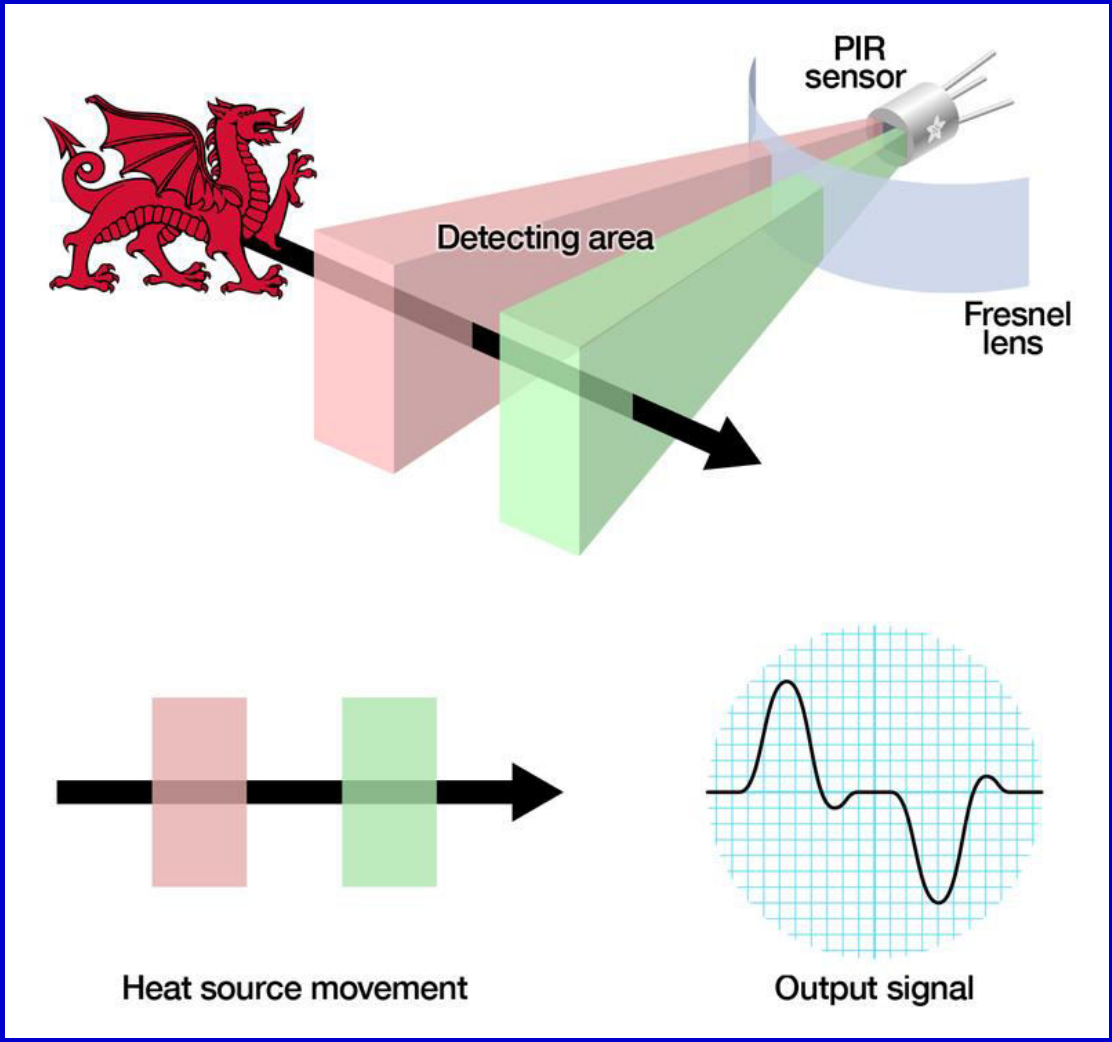
# PIR sensor



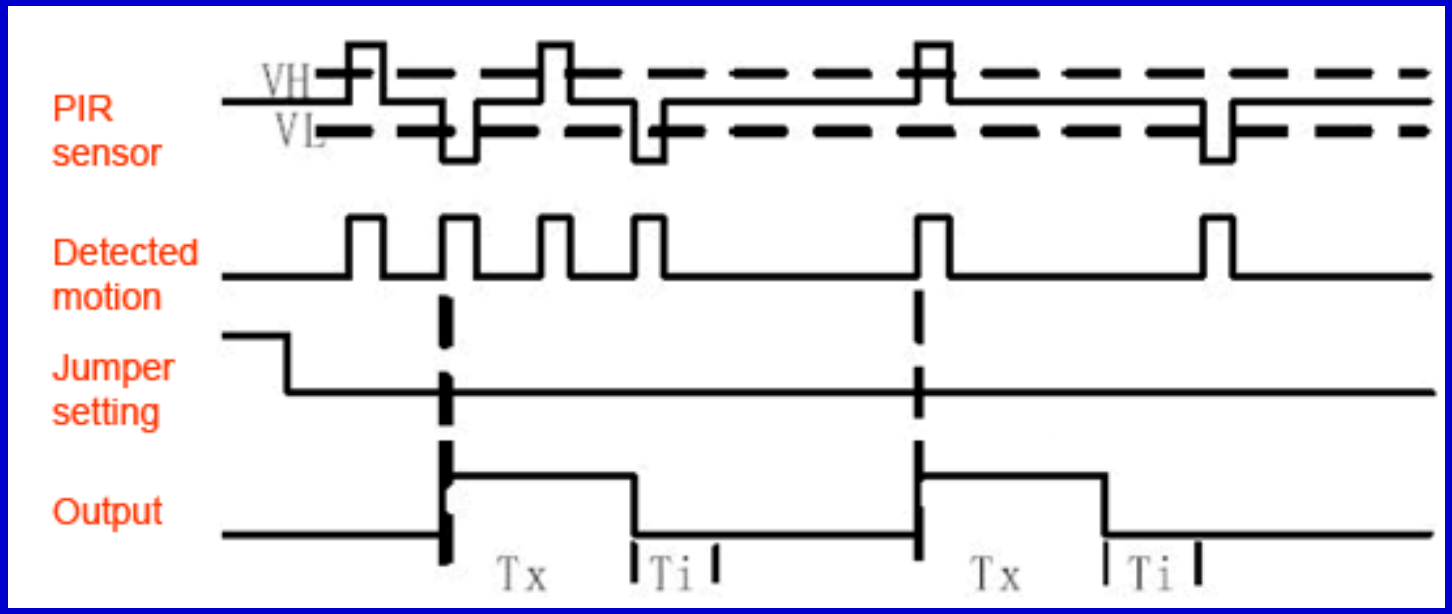
# PIR sensor design



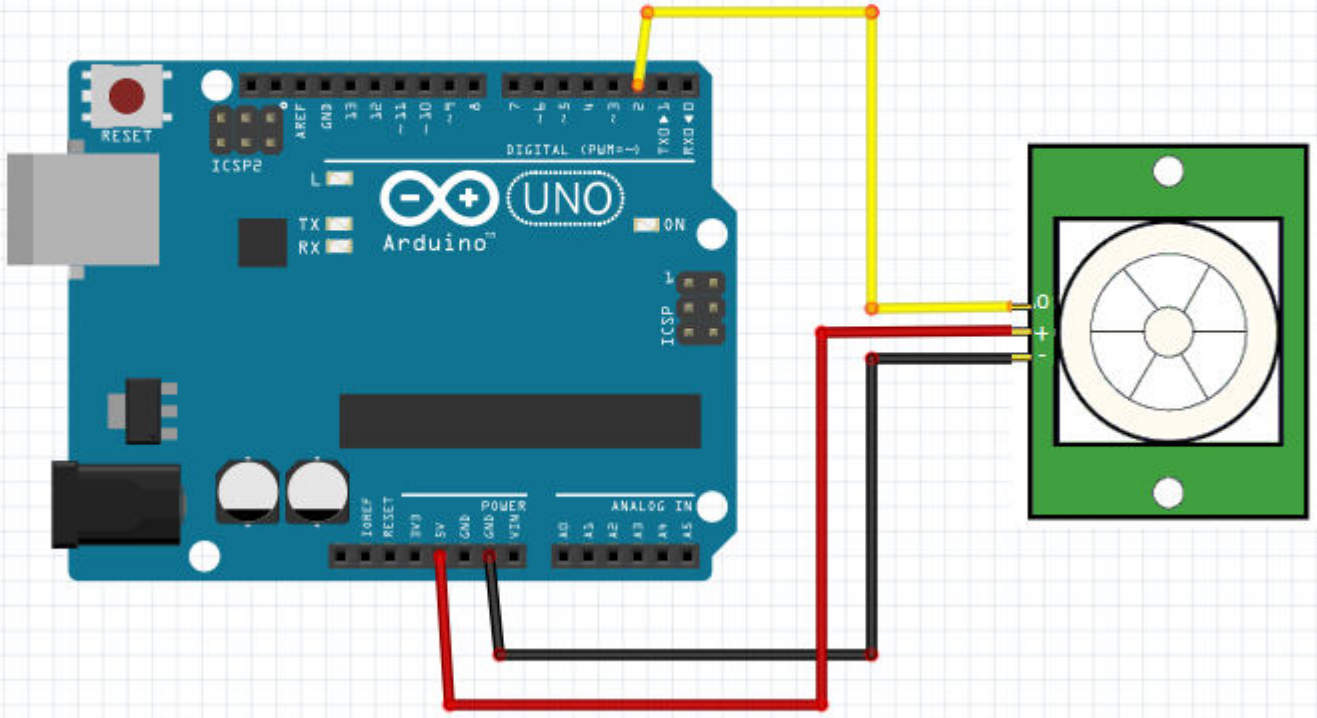
# PIR sensor operation



# PIR sensor operation



# Arduino Tutorial
# How to use PIR Motion Sensor
www.ardumotive.com




fritzing

```
/*
* PIR sensor tester
*/

int inputPin = 2;           // choose the input pin (for PIR sensor)
int pirState = LOW;     // we start, assuming no motion detected
int val = 0;                 // variable for reading the pin status
void setup() {
  pinMode(inputPin, INPUT); // declare sensor as input
  Serial.begin(9600);
}
```

```
void loop(){
   val = digitalRead(inputPin);     // read input value
    if (val == HIGH) {                  // check if the input is HIGH
        if (pirState == LOW) {
        // we have just turned on
          Serial.println("Motion detected!");
          // We only want to print on the output change, not state
          pirState = HIGH;
         }
      }
      else {
      if (pirState == HIGH){
      // we have just turned of
       Serial.println("Motion ended!");
       // We only want to print on the output change, not state
        pirState = LOW;
        }
      }
  }
```

## Libraries

Οι *libraries* είναι συλλογές κώδικα που καθιστούν εύκολο να συνδεθούν στο Arduino sensors, displays, modules, etc.
Για παράδειγμα, η ενσωματωμένη στο IDE LiquidCrystal library κάνει εύκολη την επικοινωνία με character LCD displays.
Υπάρχουν διαθέσιμες εκατοντάδες **additional libraries** στο Internet. Για να χρησιμοποιηθούν οι additional libraries, χρειάζεται να εγκατασταθούν.

## Libraries

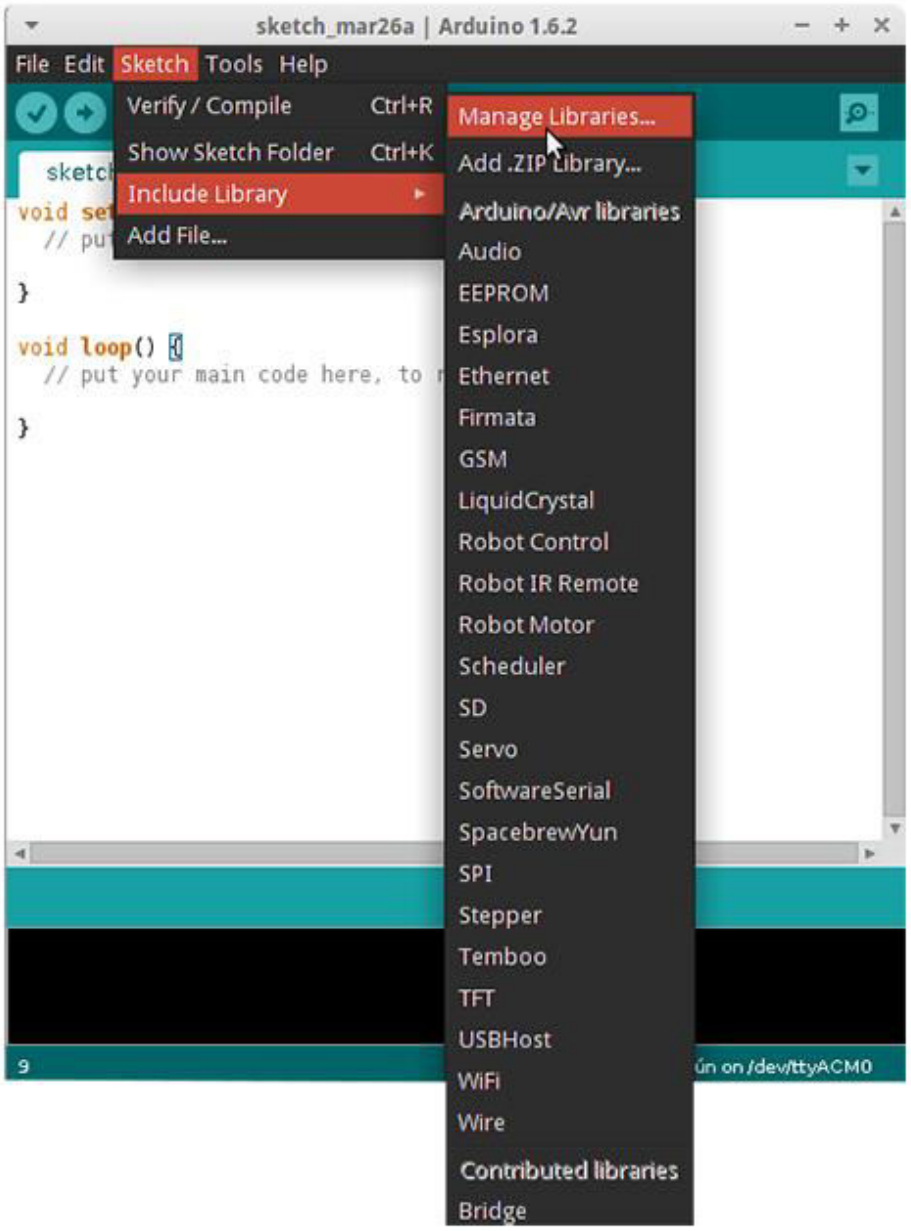
The Arduino environment can be extended through the use of libraries, just like most programming platforms.
Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from **Sketch > Import Library**.
A number of libraries come installed with the IDE, but you can also download or create your own. See these instructions for details on installing libraries.
To **install a new library** into your Arduino IDE you can use the Library Manager. Open the IDE and click to the "Sketch" menu and then *Include Library*

**Standard Libraries**

EEPROM - reading and writing to "permanent" storage.
Ethernet - for connecting to the internet using the Arduino Ethernet Shield
GSM - for connecting to a GSM/GRPS network with the GSM shield.
LiquidCrystal - for controlling liquid crystal displays (LCDs)
SD - for reading and writing SD cards
SPI - for communicating with devices using the Serial Peripheral Interface (SPI) Bus

SoftwareSerial - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's NewSoftSerial library as SoftwareSerial.
Servo - for controlling servo motors
Stepper - for controlling stepper motors
TFT - for drawing text, images and shapes on the Arduino TFT screen
WiFi - for connecting to the internet using the Arduino WiFi shield
Wire - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

# Install new Arduino Libraries



**#define**

*#define* is a useful C component that allows the programmer to give a name to a constant value before the program is compiled. Defined constants in Arduino don't take up any program memory space on the chip.
The compiler will replace references to these constants with the defined value at compile time.

**#**define, has no semicolon terminator

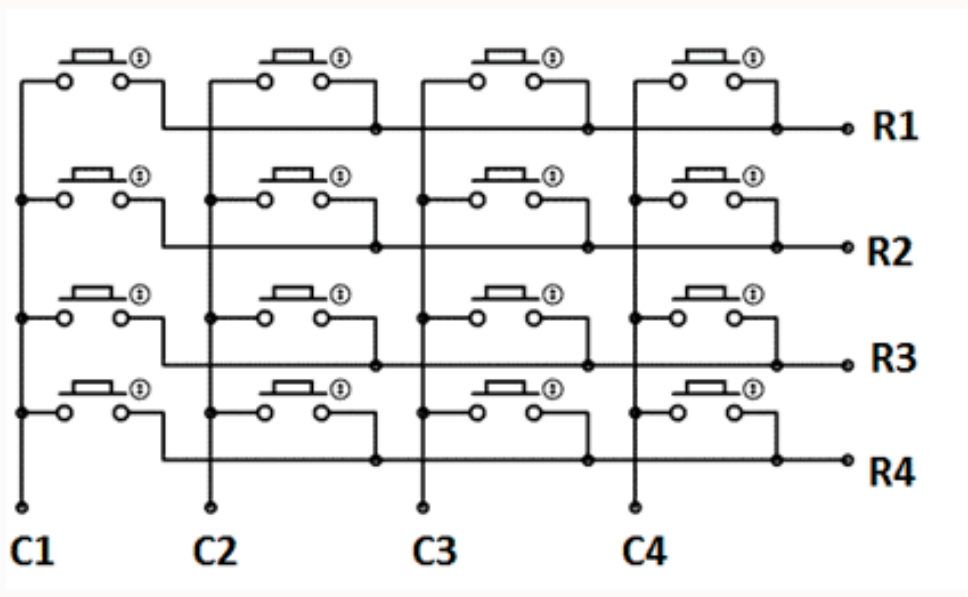
```
#define constantName value
#define ledPin 3  // The compiler will replace any mention of
                  // ledPin with the value 3 at compile time.
```

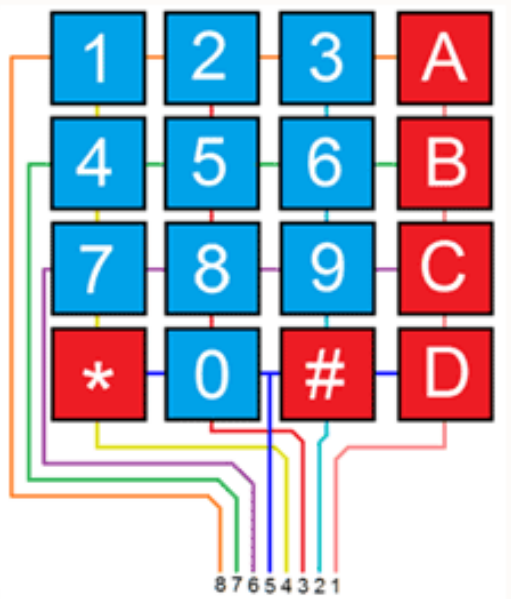
**#include**

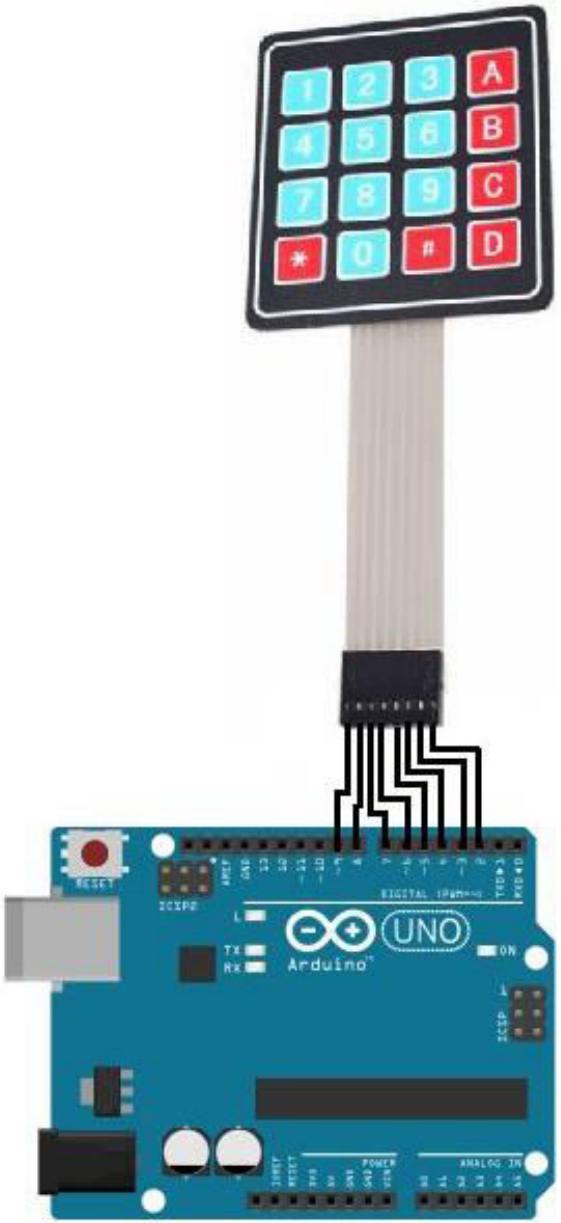
***#include*** is used to include outside libraries in your sketch. This gives the programmer access to a large group of standard C libraries (groups of pre-made functions), and also libraries written especially for Arduino.

**#**include, has no semicolon terminator.

# 4x4 matrix keyboard



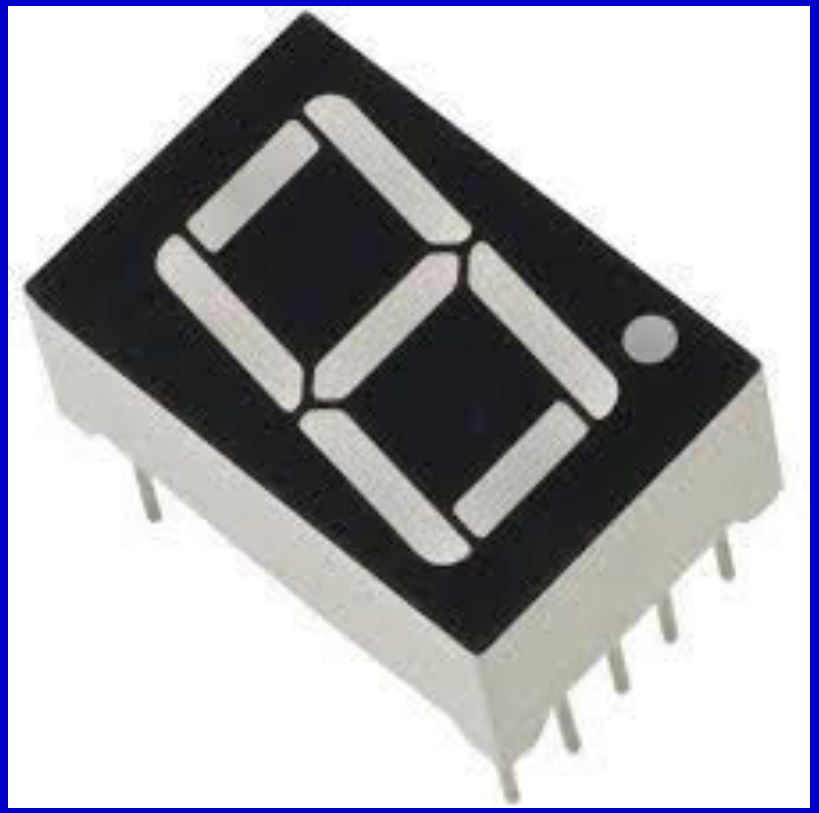

1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D
RESET
ICSP2
L
TX
RX
UNO
Arduino
ON
ICSP
POWER
ANALOG IN

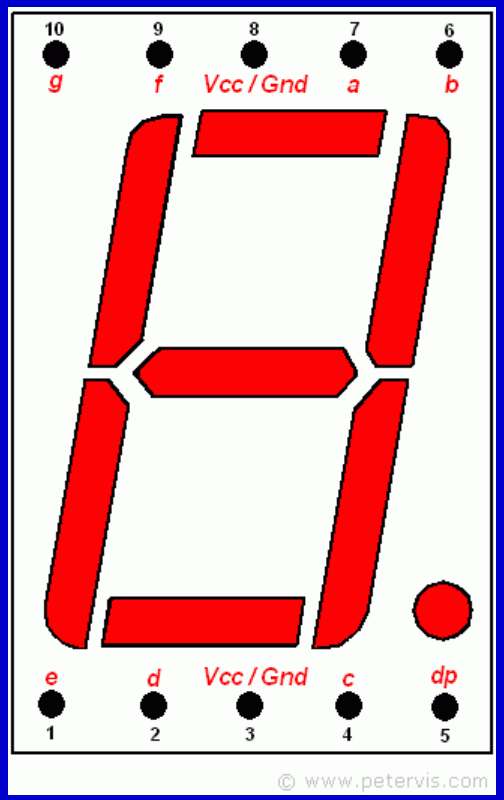
```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;
//define the symbols on the buttons of the keypads
char Keys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2,3,4,5};
byte colPins[COLS] = {6,7,8,9};
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad(makeKeymap(Keys),
rowPins, colPins, ROWS, COLS);
```

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Please press the keyboard:");
}

void  loop()
{
  char key = customKeypad.getKey();
  if(key!=NO_KEY)
  {
    Serial.print("Key Value : ");
    Serial.println(key);
  }
}
```

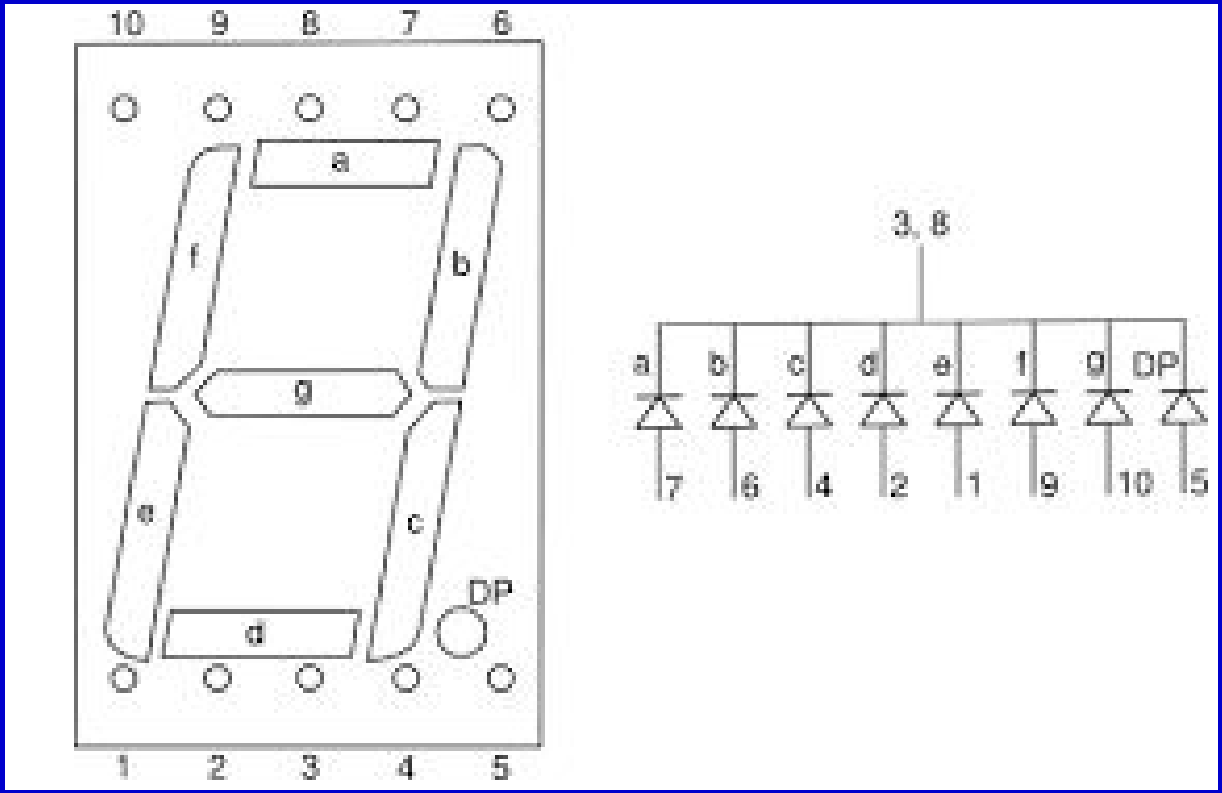
# Σύνδεση Arduino με seven segment display

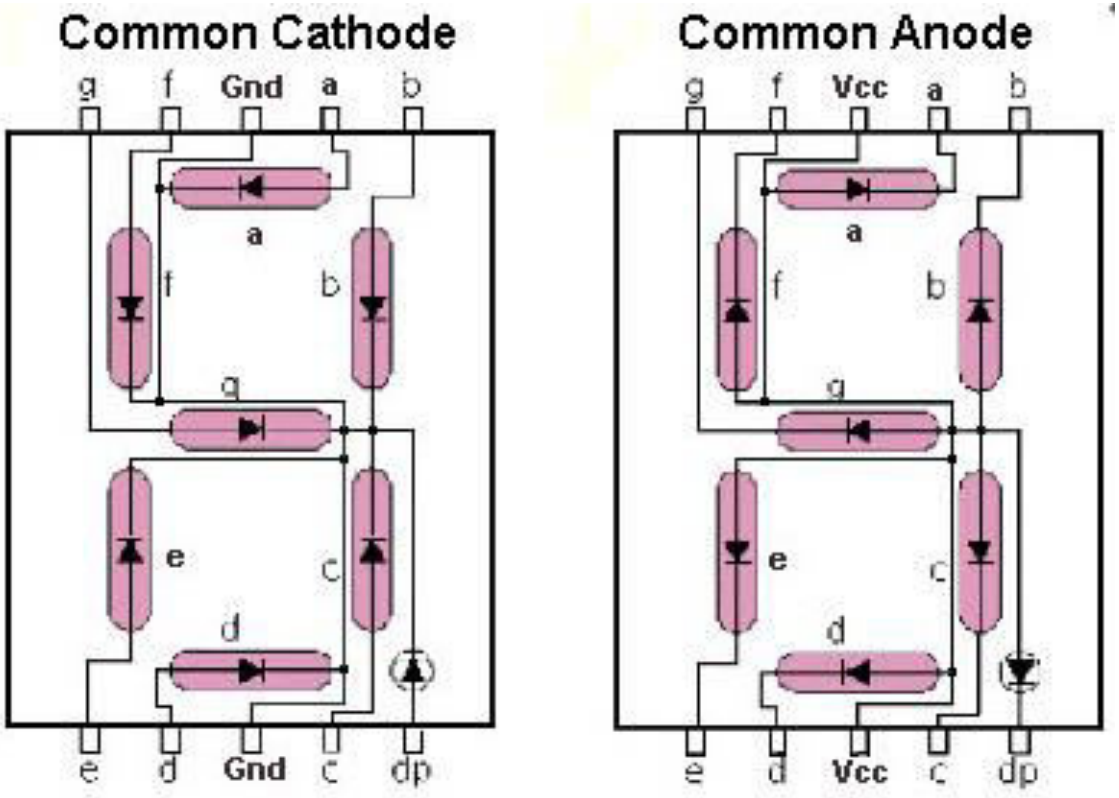
# Seven Segment Display



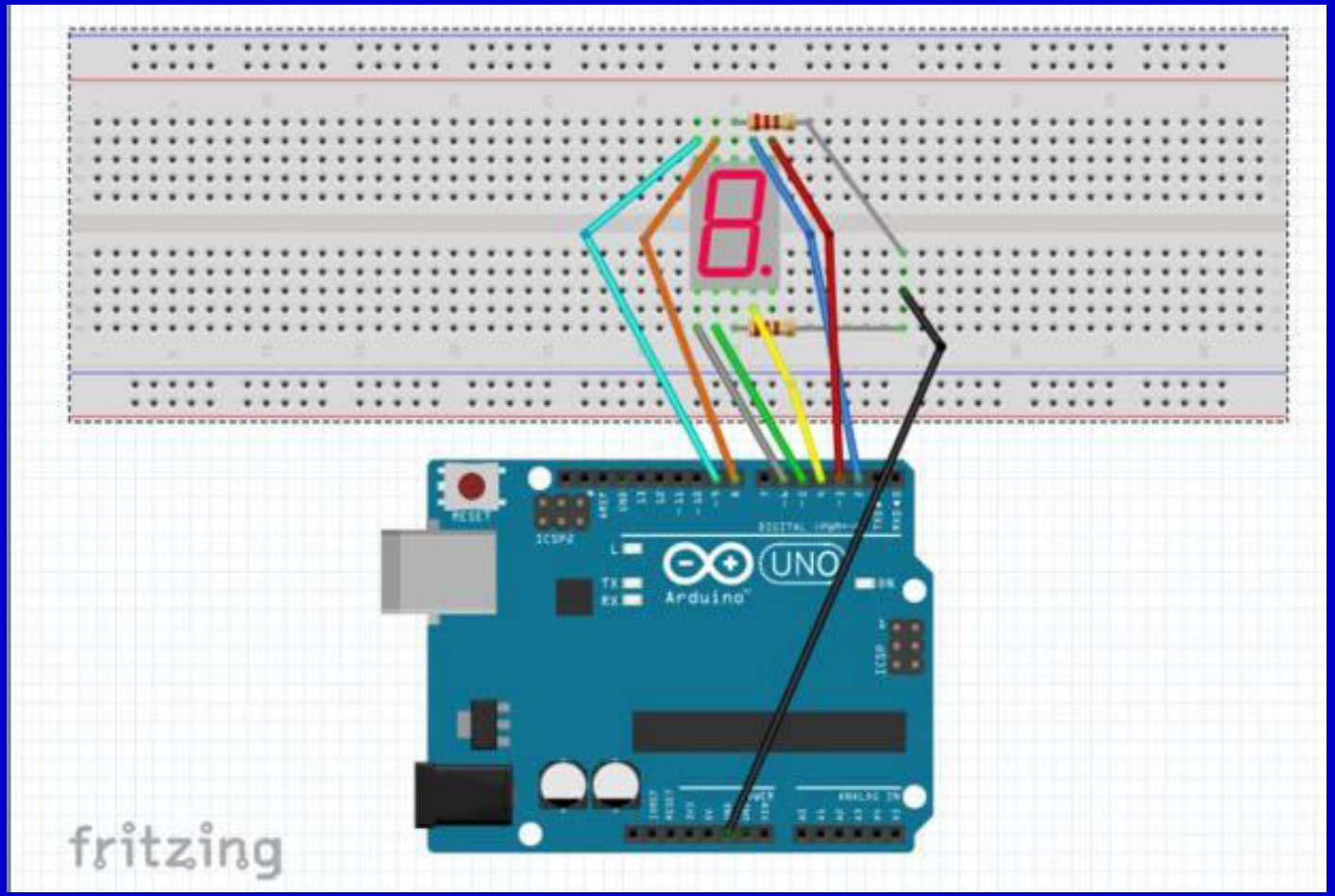


# Seven-Segment Display κοινής καθόδου




Common Cathode
g
f
Gnd
a
b
a
f
b
g
e
c
d
e
d
Gnd
c
dp
Common Anode
g
f
Vcc
a
b
a
f
b
g
e
c
d
e
d
Vcc
c
dp

# Σύνδεση 7-segment display στο Arduino




R=220 Ohm

## Κώδικας του Arduino

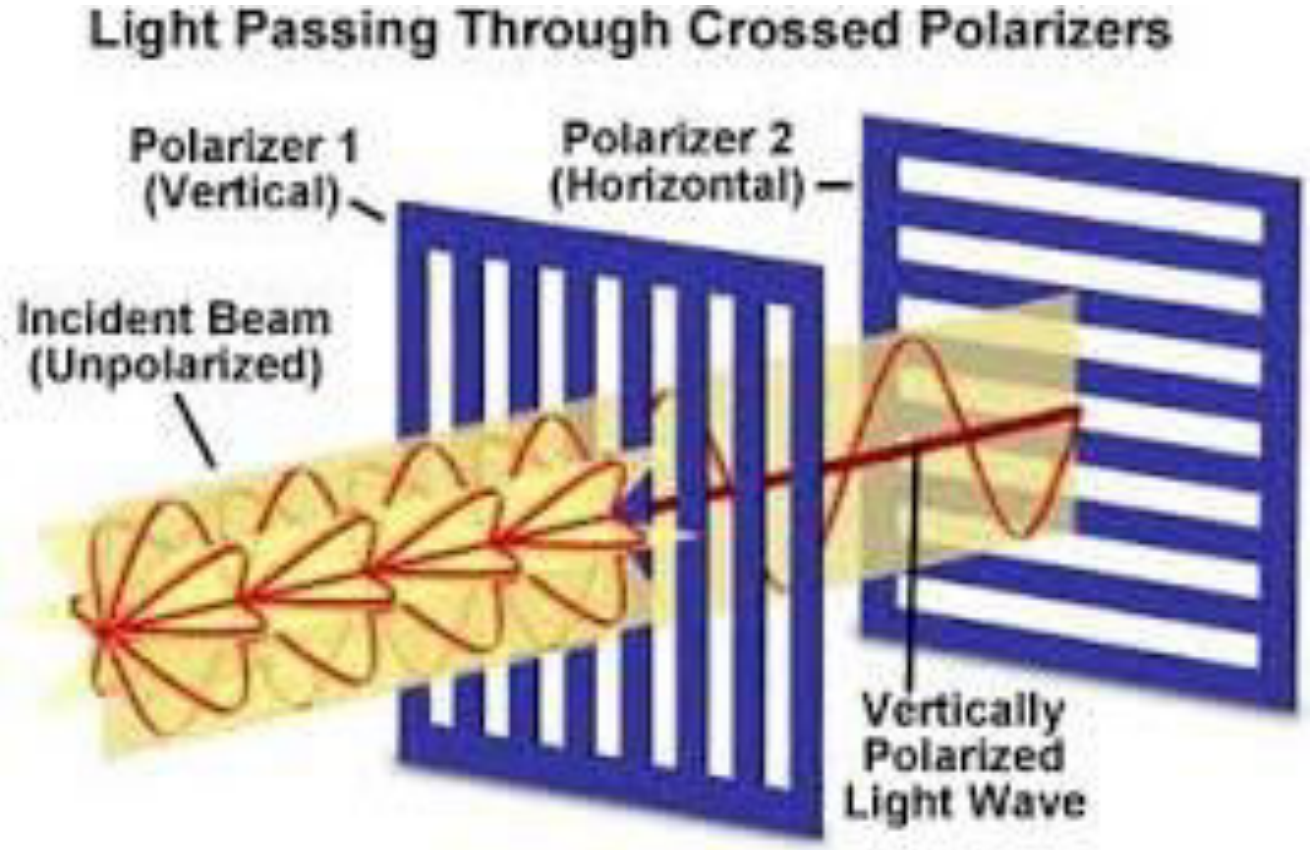
```
int a = 2;  //For displaying segment "a“
int b = 3;  //For displaying segment "b“
int c = 4;  //For displaying segment "c“
int d = 5;  //For displaying segment "d“
int e = 6;  //For displaying segment "e“
int f = 8;  //For displaying segment "f“
int g = 9;  //For displaying segment "g“

void setup() {
pinMode(a, OUTPUT);  //A
pinMode(b, OUTPUT);  //B
pinMode(c, OUTPUT);  //C
pinMode(d, OUTPUT);  //D
pinMode(e, OUTPUT);  //E
pinMode(f, OUTPUT);  //F
pinMode(g, OUTPUT);  //G
}
```

```
void displayDigit(int digit){
//Conditions for displaying segment a
if(digit!=1 && digit != 4) digitalWrite(a,HIGH);
//Conditions for displaying segment b
if(digit != 5 && digit != 6) digitalWrite(b,HIGH);
//Conditions for displaying segment c
if(digit !=2) digitalWrite(c,HIGH);
//Conditions for displaying segment d
if(digit != 1 && digit !=4 && digit !=7) digitalWrite(d,HIGH);
//Conditions for displaying segment e
if(digit == 2 || digit ==6 || digit == 8 || digit==0)
digitalWrite(e,HIGH);
//Conditions for displaying segment f
if(digit != 1 && digit !=2 && digit!=3 && digit !=7)
digitalWrite(f,HIGH);
if (digit!=0 && digit!=1 && digit !=7) digitalWrite(g,HIGH);
}
```

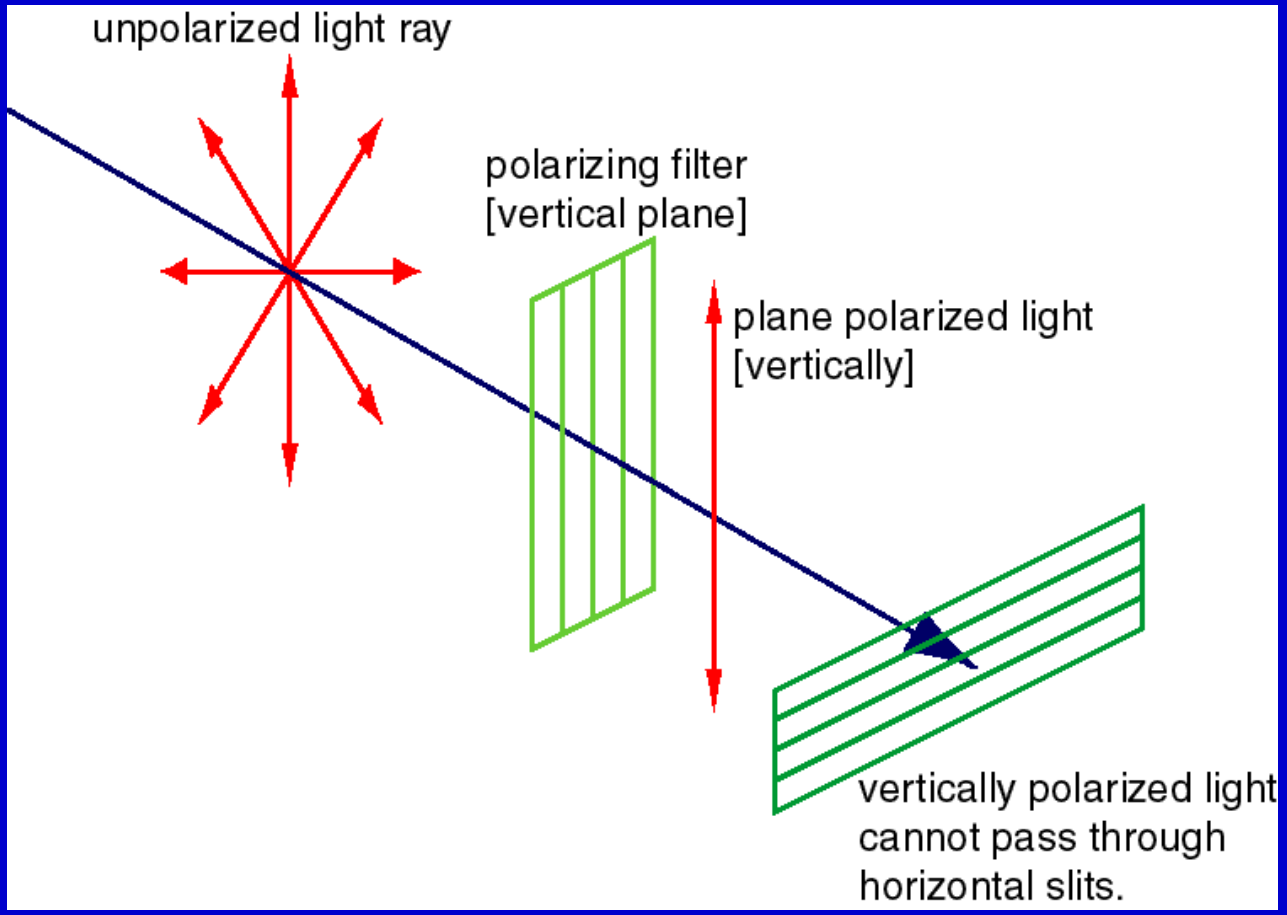
```
void turnOff(){
digitalWrite(a,LOW);
digitalWrite(b,LOW);
digitalWrite(c,LOW);
digitalWrite(d,LOW);
digitalWrite(e,LOW);
digitalWrite(f,LOW);
digitalWrite(g,LOW);
}

  void loop() {
  for(int i=0; i<10; i++) {
     displayDigit(i);
     delay(1000);
     turnOff();
      }
  }
```

# Σύνδεση Arduino με οθόνη LCD με χρήση της αντίστοιχης library

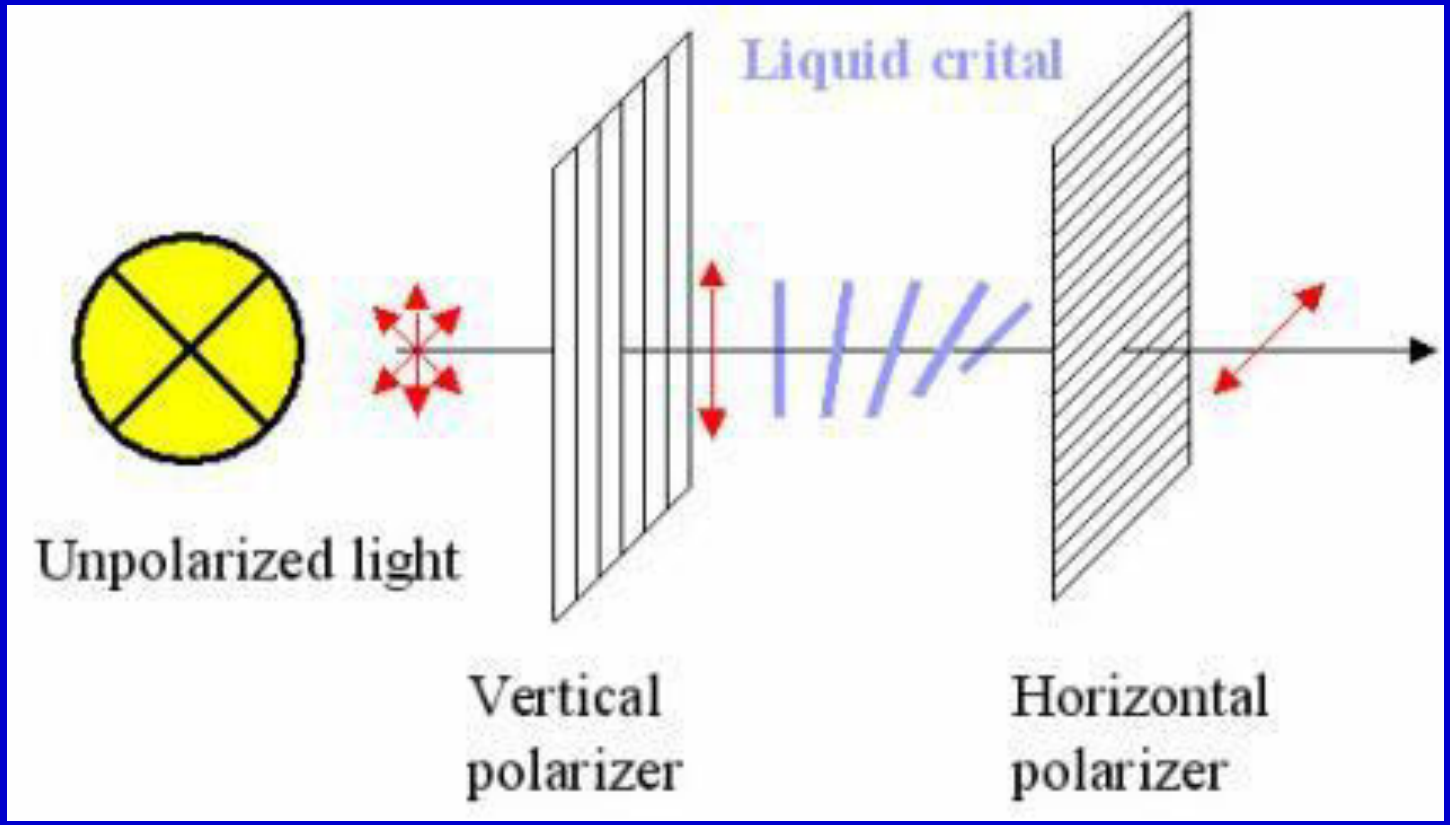
# Πόλωση του φωτός

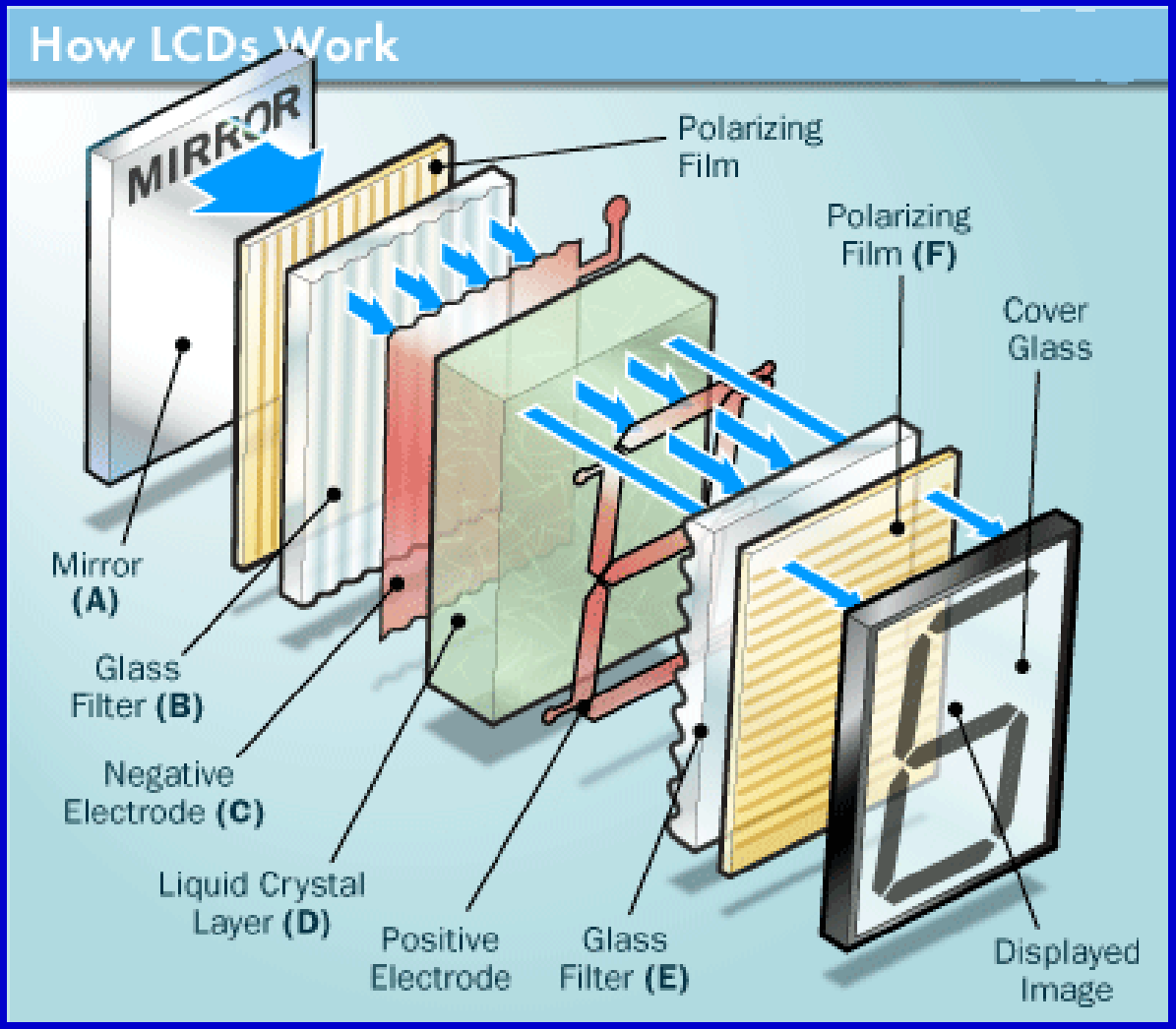


# Διέλευση του φωτός από οριζόντιο και κάθετο πολωτή

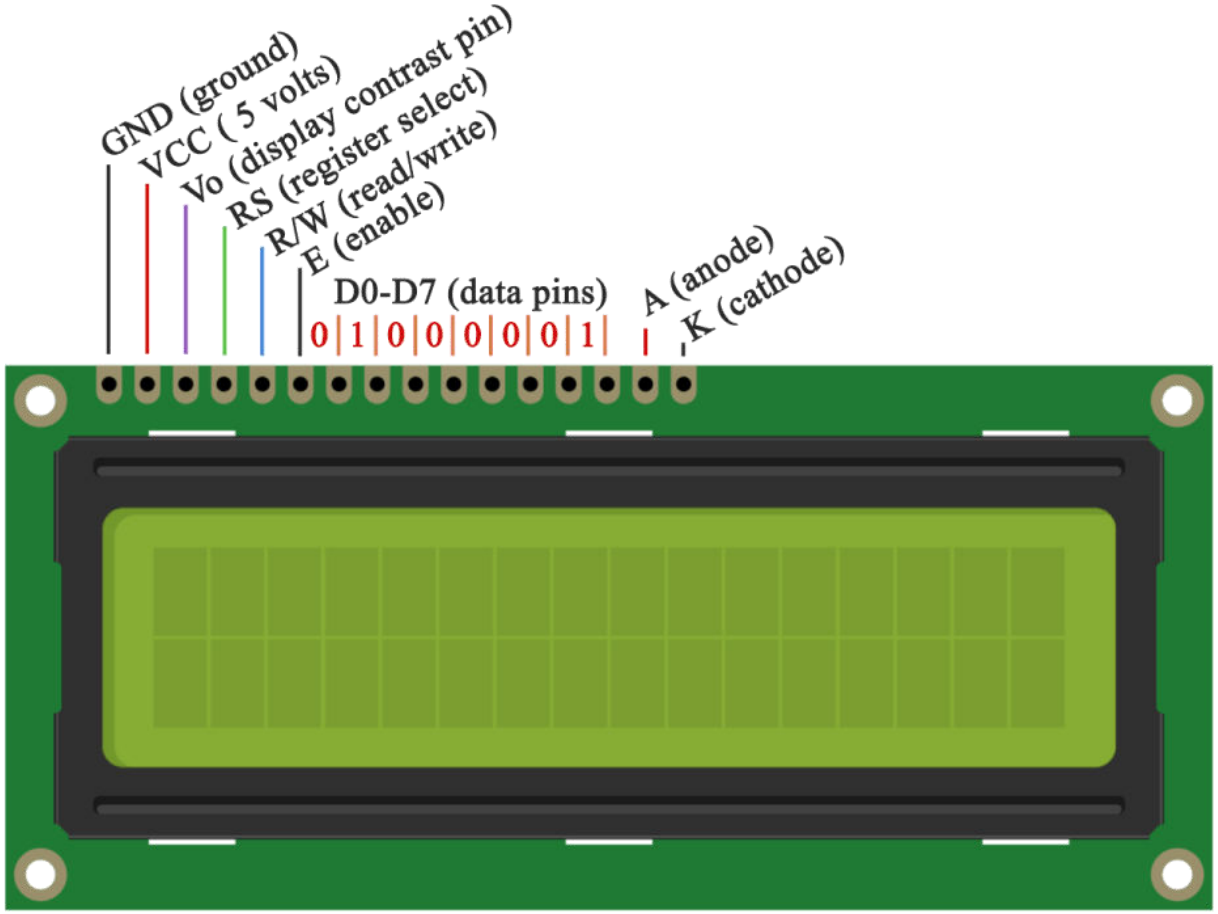


# Χρήση οπτικών κρυστάλλων για την περιστροφή του επιπέδου ταλάντωσης του πολωμένου φωτός

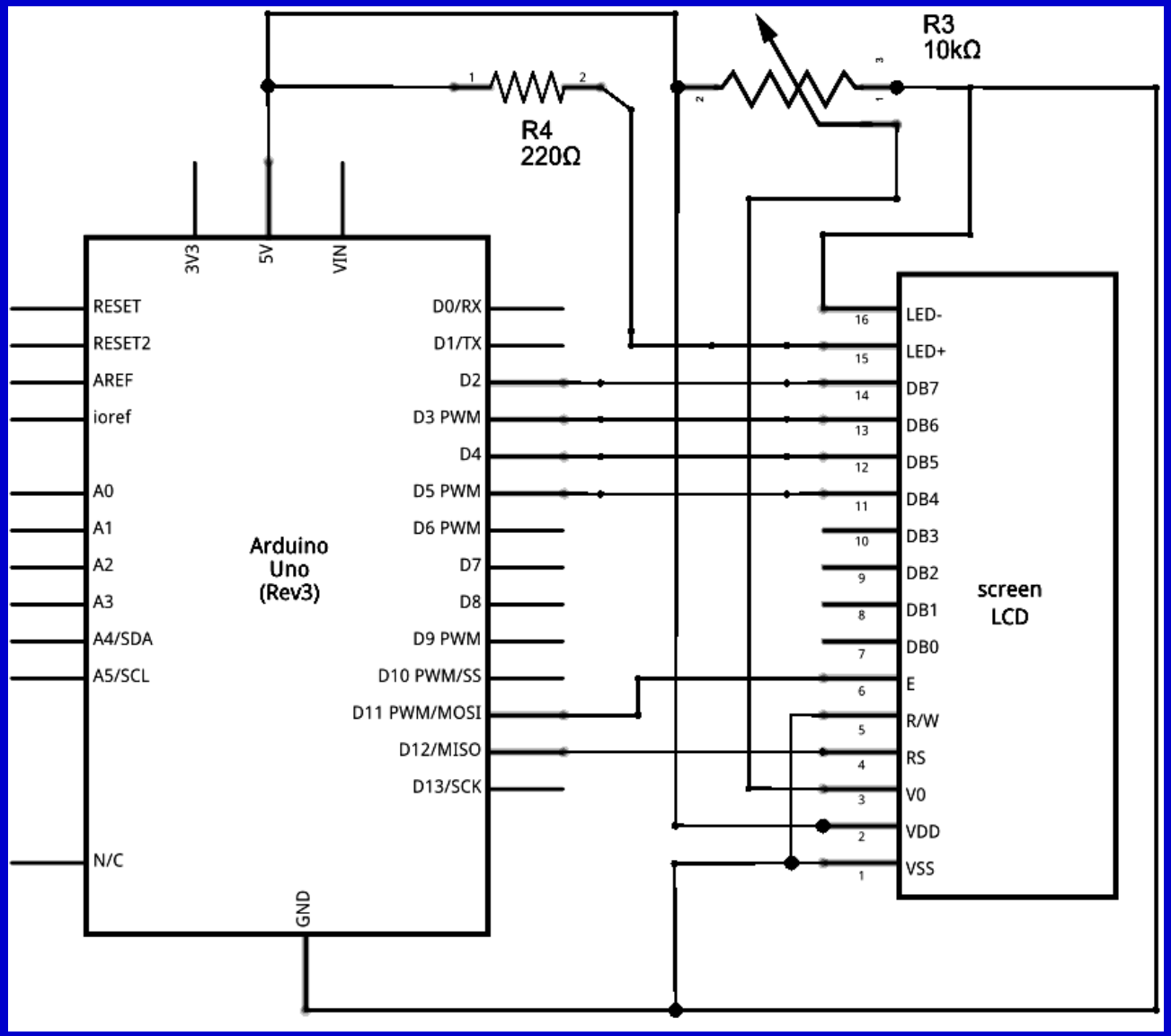



How LCDs Work
MIRROR
Polarizing Film
Polarizing Film (F)
Cover Glass
Mirror (A)
Glass Filter (B)
Negative Electrode (C)
Liquid Crystal Layer (D)
Positive Electrode
Glass Filter (E)
Displayed Image

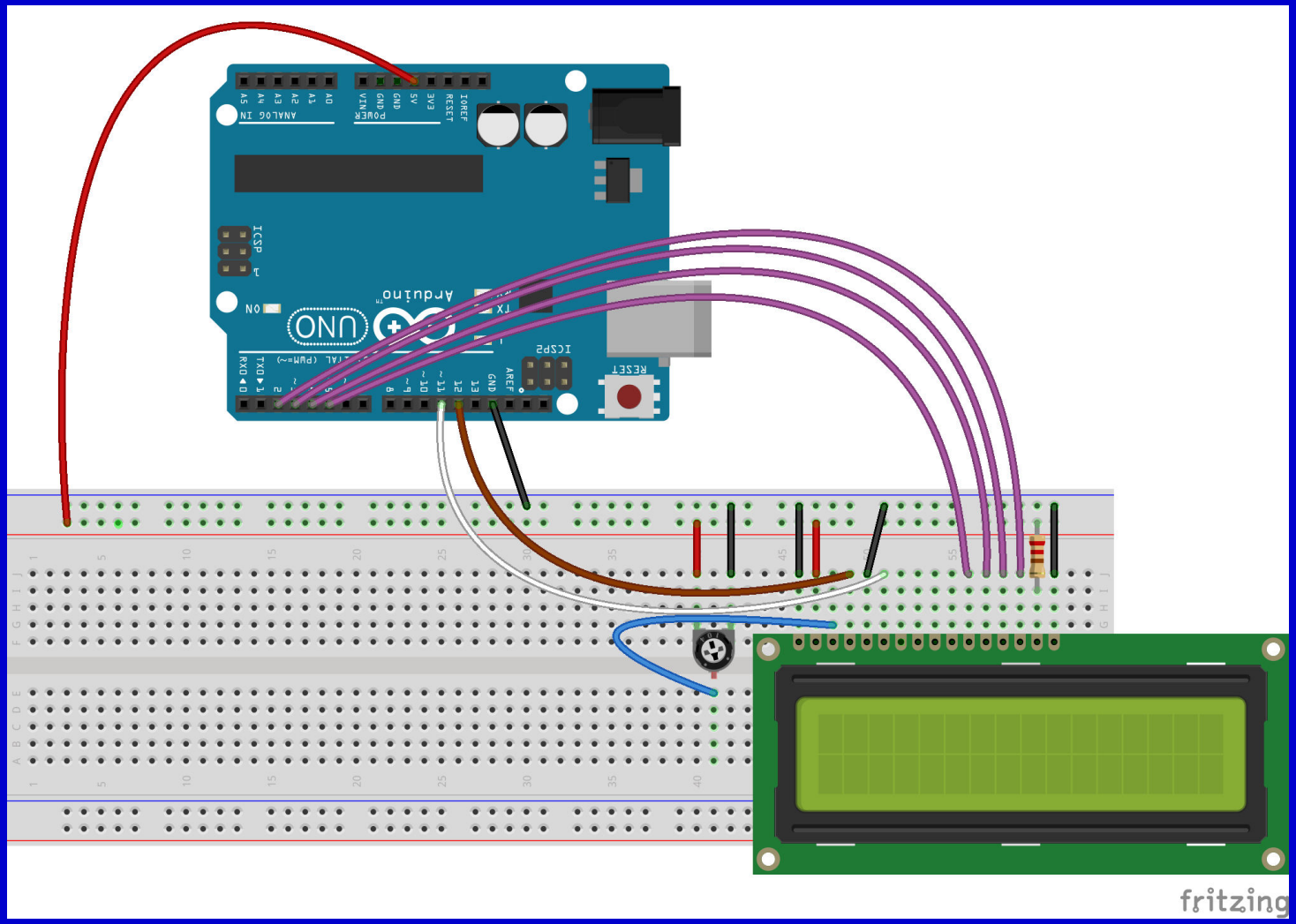
# LCD screen


GND (ground)
VCC ( 5 volts)
Vo (display contrast pin)
RS (register select)
R/W (read/write)
E (enable)
D0-D7 (data pins)
0 1 0 0 0 0 0 1
A (anode)
K (cathode)

# Σύνδεση του Arduino με οθόνη LCD



# Σύνδεση Arduino με οθόνη LCD



## Liquid Crystal Display (LCD) Library

The **LiquidCrystal** library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines).

## Συναρτήσεις της LiquidCrystal Display Library

### lcd.begin(cols, rows)

**Description**
Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display.
begin() needs to be called before any other LCD library commands.

**Parameters**
cols: the number of columns that the display has
rows: the number of rows that the display has

**LiquidCrystal(rs, enable, d4, d5, d6, d7)**

**Description**
Creates a variable of type LiquidCrystal.

**Parameters**
**rs:** the number of the Arduino pin that is connected to the RS pin on the LCD.
**enable:** the number of the Arduino pin that is connected to the enable pin on the LCD.
**d4, d5, d6, d7:** The number of the Arduino pins

**Πρόγραμμα που εμφανίζει στην οθόνη LCD την φράση hello world και το πόσος χρόνος πέρασε από την έναρξή του**

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

  void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
}
```

# Πρόγραμμα δημιουργίας σειριακής πόρτας με χρήση library

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11);
// creates a "virtual" serial port/UART
// connect BT module TX to D10
// connect BT module RX to D11
// connect BT Vcc to 5V, GND to GND
void setup() {
   //set digital pin to control as an output
     pinMode(13, OUTPUT);
   //set the data rate for the Software
     Serial port BT.begin(9600);
   // Send test message to other device
     BT.println("Hello from Arduino");
}
char a;
// stores incoming character from other device
```

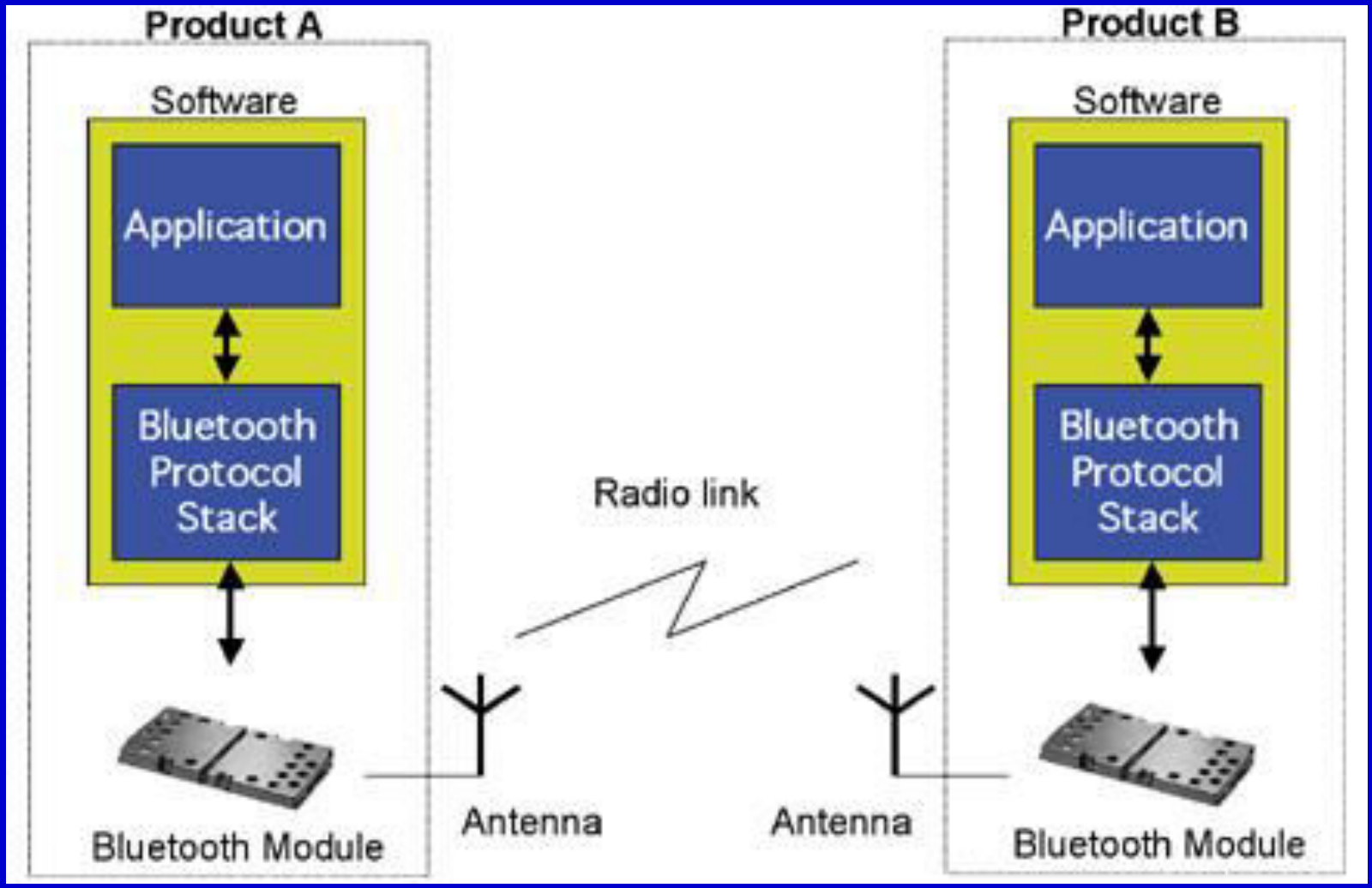
```
void loop() {
if (BT.available()) // if text arrived in from BT serial...
  {
   a=(BT.read());
      if (a=='1') {
      digitalWrite(13, HIGH);
   BT.println("LED on");
  }
    if (a=='2') {
    digitalWrite(13, LOW);
    BT.println("LED off");
 }
if (a=='?') {
      BT.println("Send '1' to turn LED on");
      BT.println("Send '2' to turn LED on");
    }
  }
} // loop
```

# Σύνδεση Arduino με Smartphone Android μέσω Bluetooth

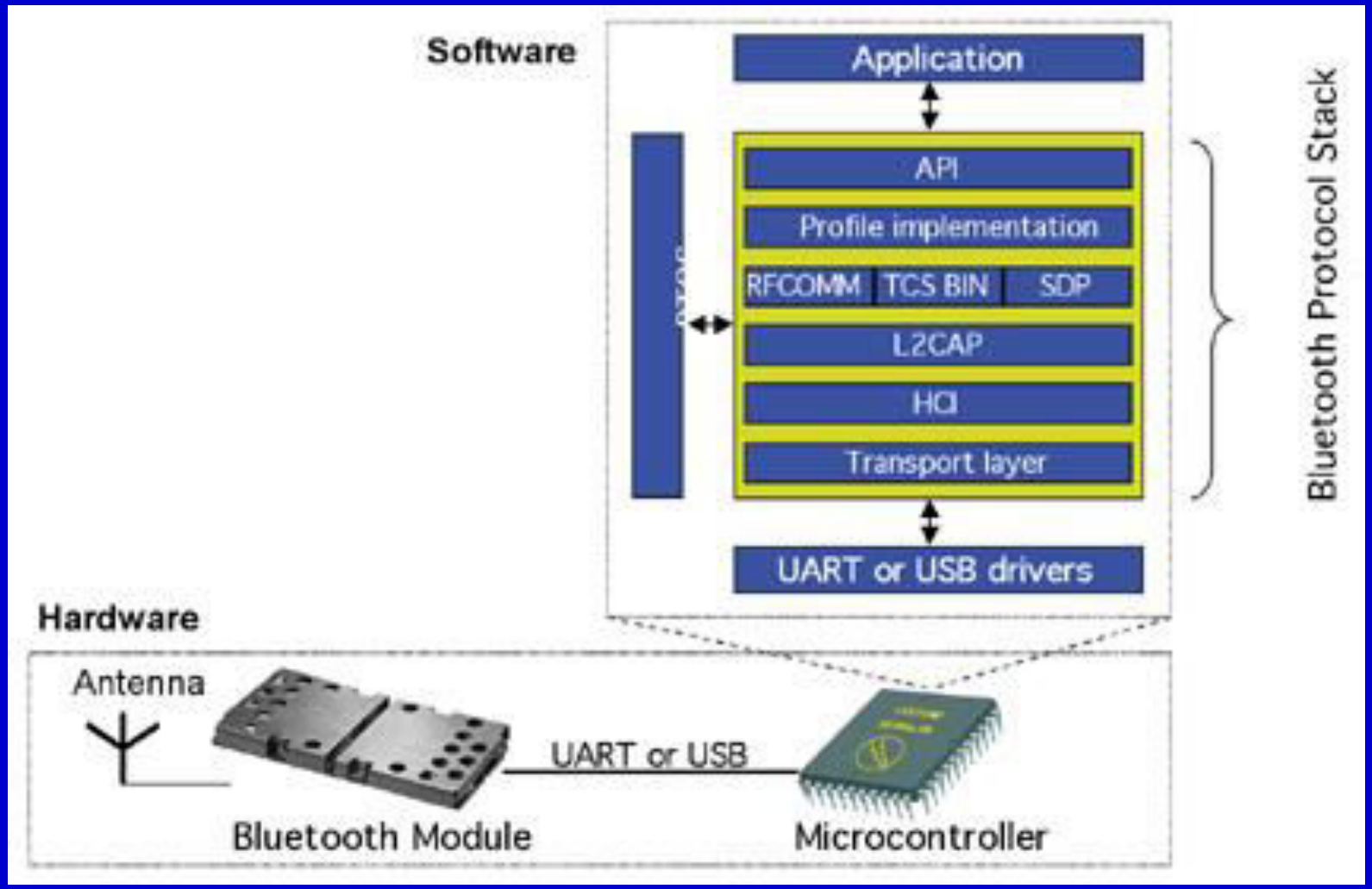
## Bluetooth

**Bluetooth** is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables. It can connect several devices, overcoming problems of synchronization.

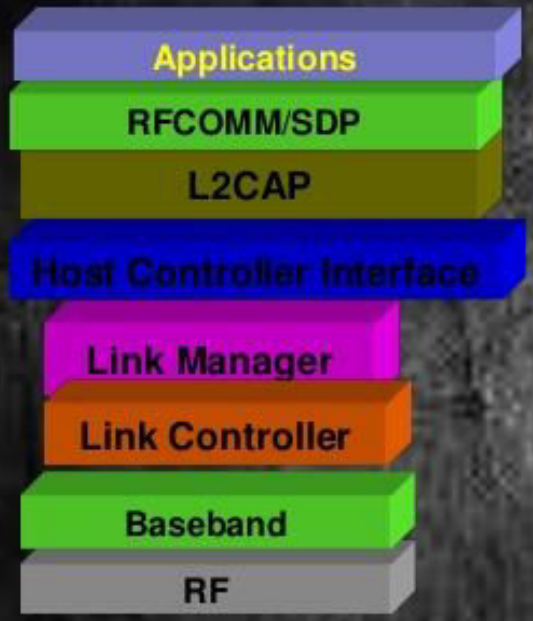
# Επικοινωνία με Blootooth

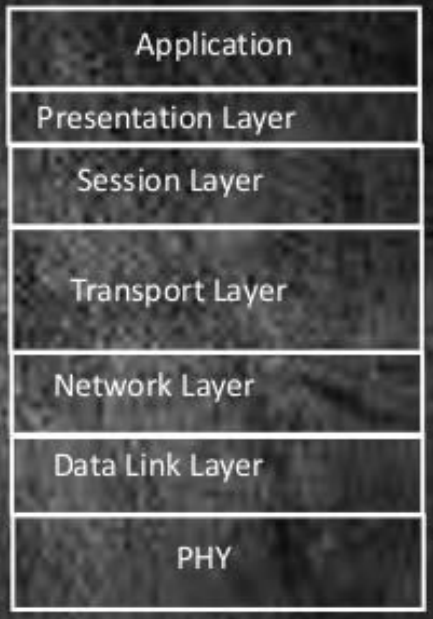


# Αρχιτεκτονική του Bluetooth



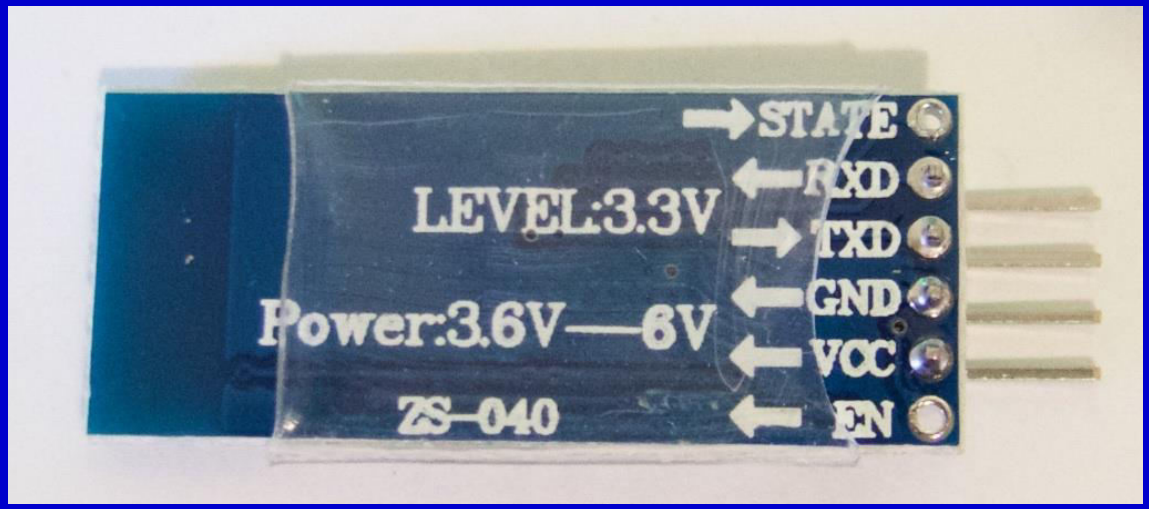
# Bluetooth Protocol Stack

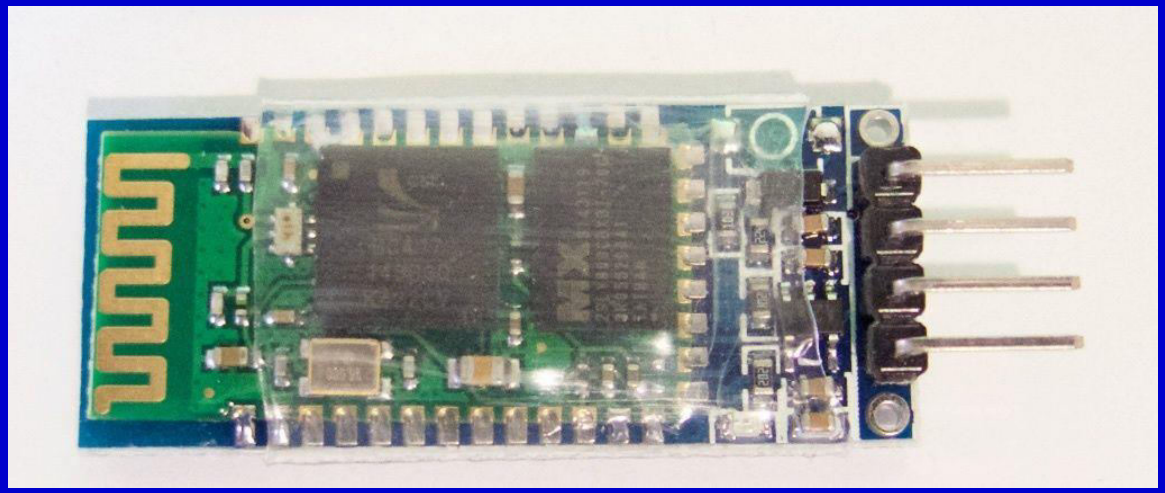



OSI/ISO

# Bluetooth module HC-06



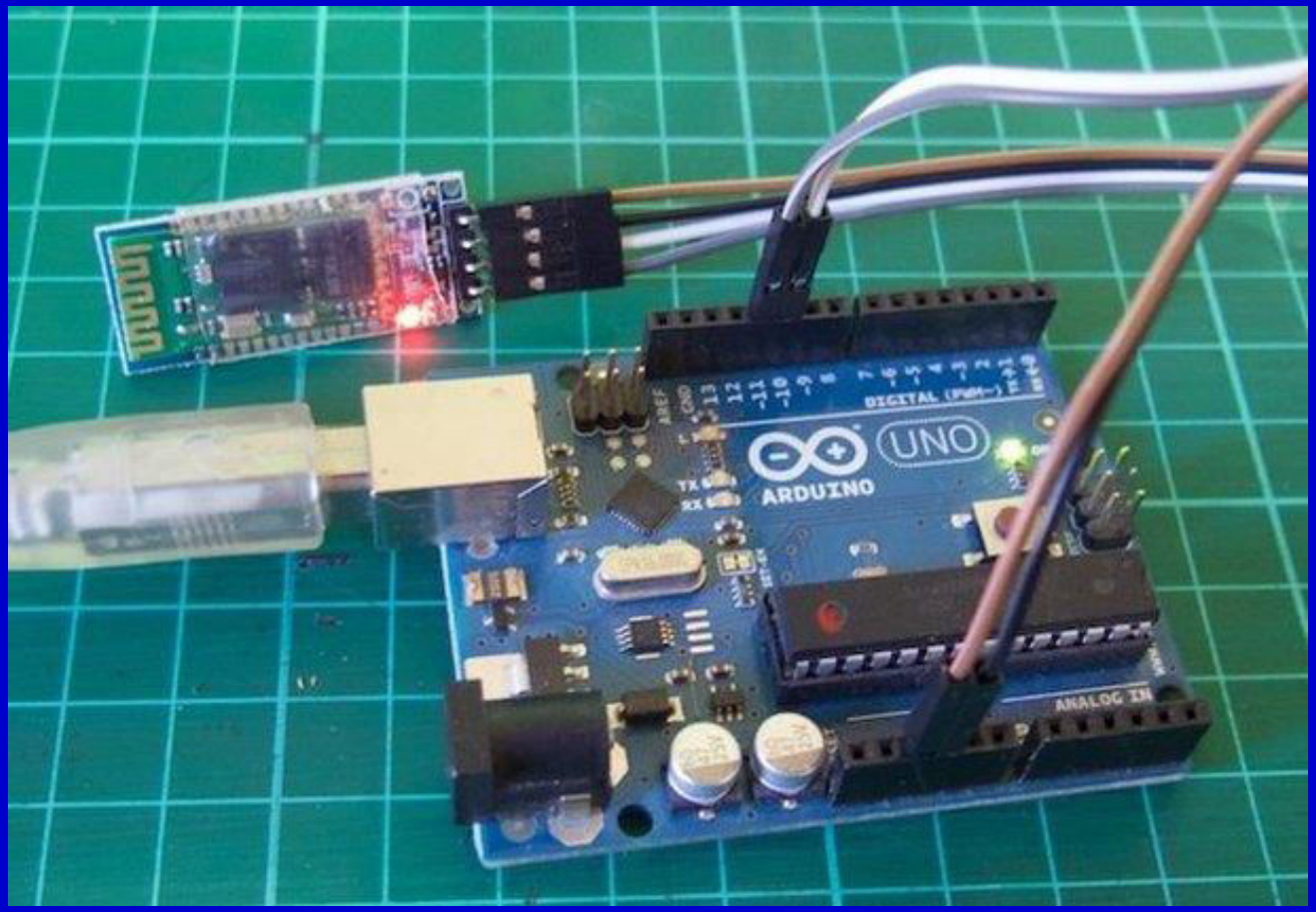



**Συνδεσμολογία HC-06 με Arduino**

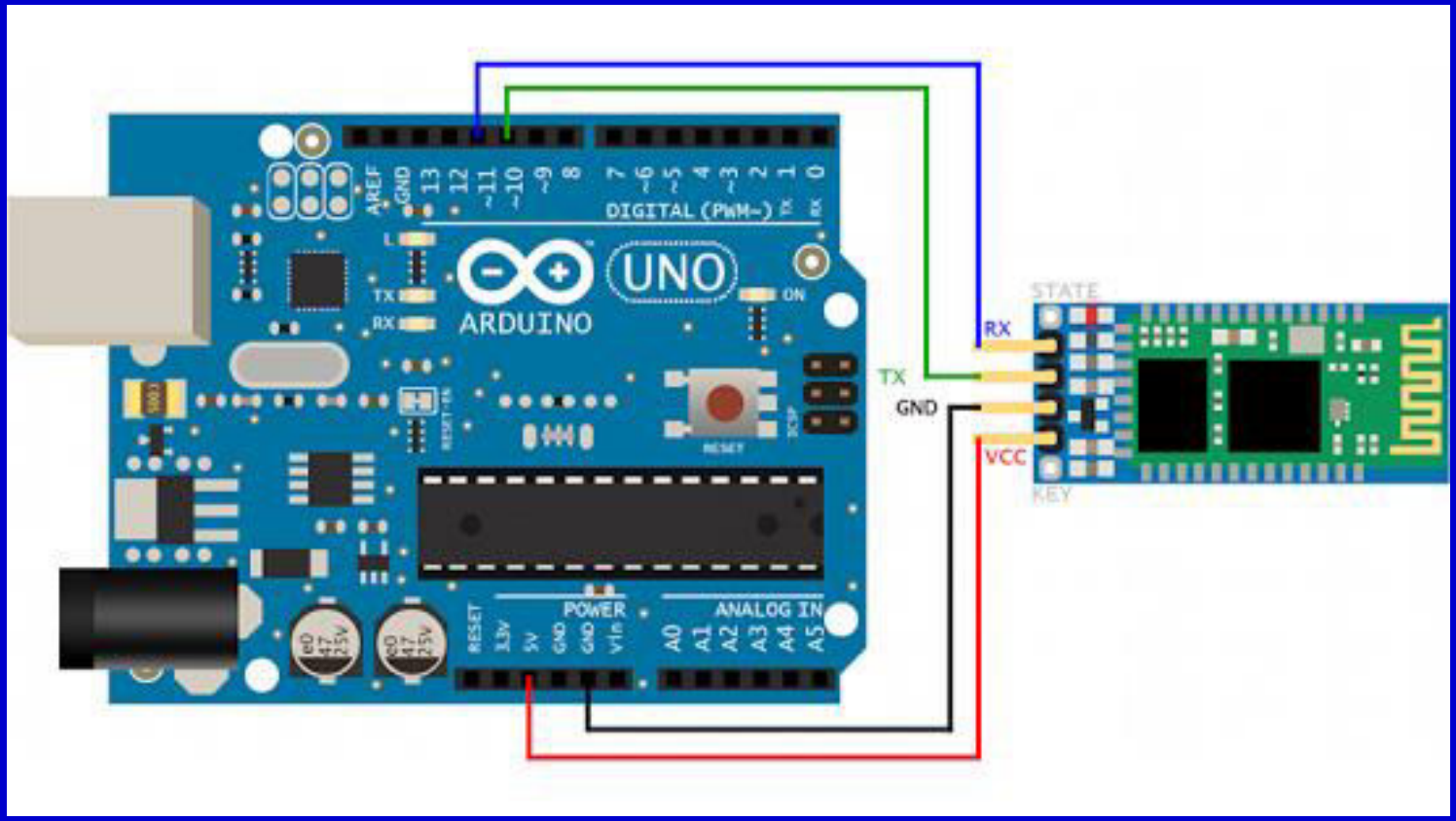
5V to Vcc
GND to GND
D10 (RX Arduino) to TX (HC-09)
D11 (TX Arduino) to RX (HC-09)

Οι ακίδες D10, D11 χρησιμοποιούνται σαν RX (receive) και TX (transmit) για υλοποίηση με Library της ασύγχρονης σειριακής επικοινωνίας.

# Συνδεσμολογία HC-06 με Arduino



# Συνδεσμολογία HC-06 με Arduino



```
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11);

void setup() {
 // set digital pin to control as an output
pinMode(13, OUTPUT);
// set the data rate for the Software
Serial port BT.begin(9600);
// Send test message to other device
BT.println("Hello from Arduino");
 }

char a;
// stores incoming character from other device
```

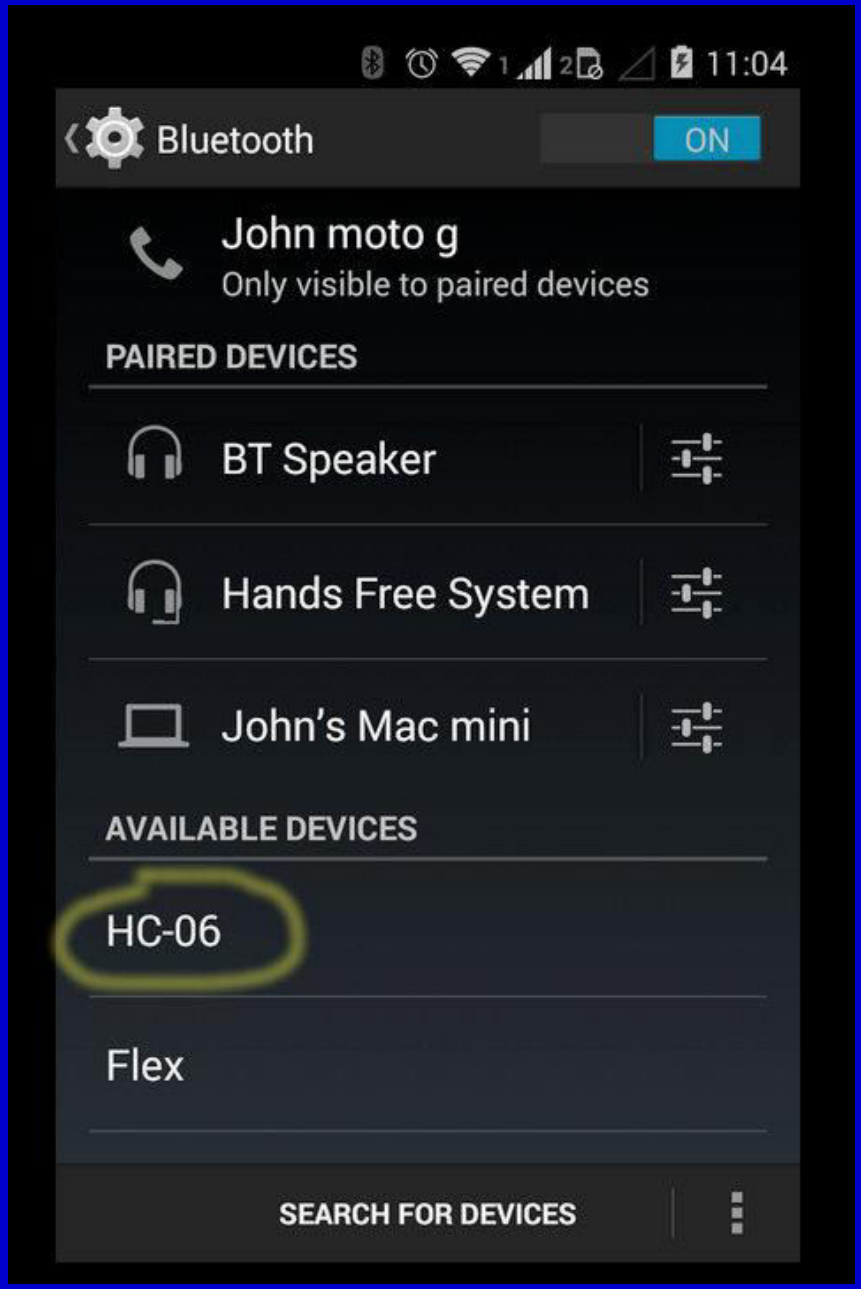
```
void loop() {
 if (BT.available()) // if text arrived in from  BT serial...
 {
  a=(BT.read());
   if (a=='1') {
   digitalWrite(13, HIGH);
   BT.println("LED on");
    }
   if (a=='2') {
   digitalWrite(13, LOW);
   BT.println("LED off");
    }
   if (a=='?') {
   BT.println("Send '1' to turn LED on");
   BT.println("Send '2' to turn LED on");
   }
 }
}
```

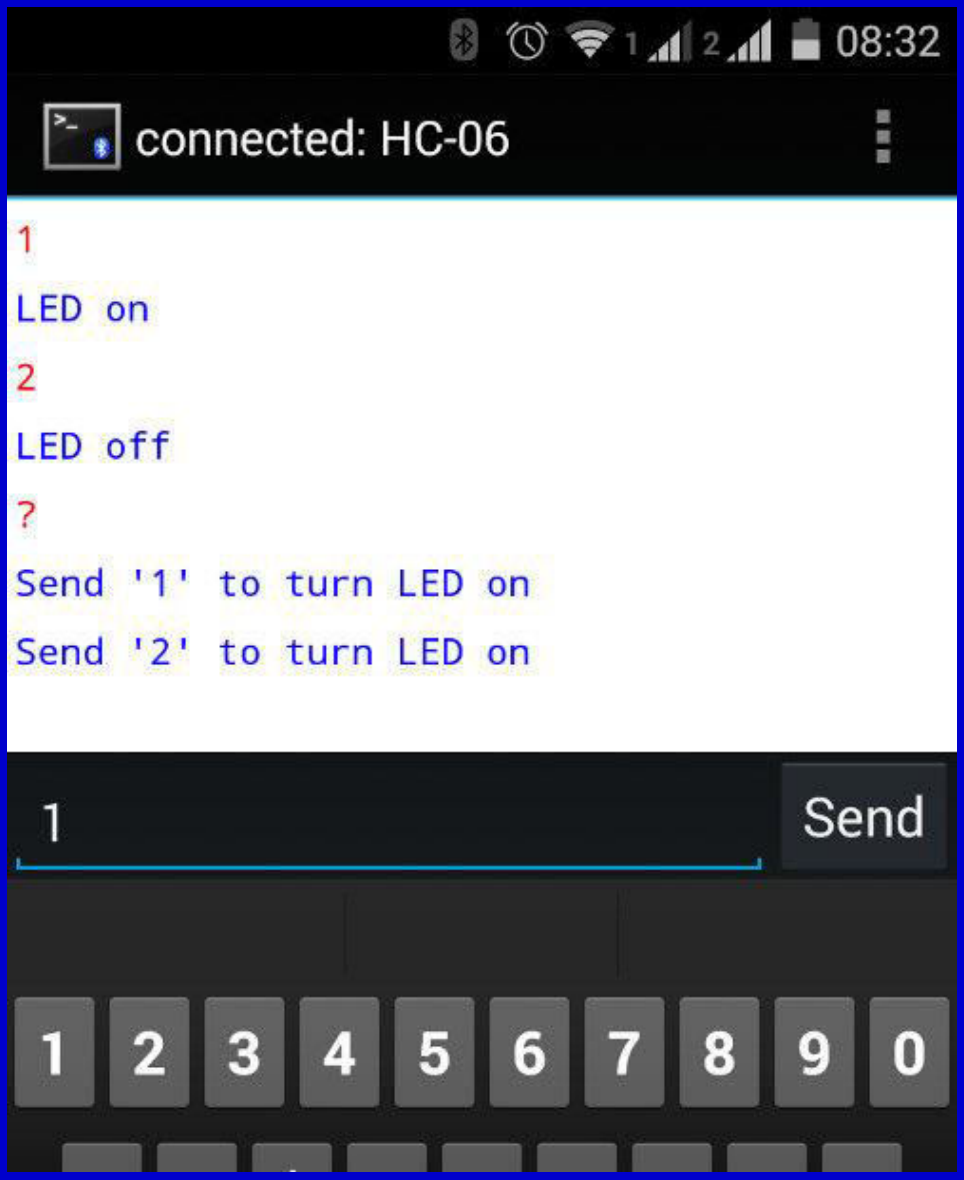
We used “Bluetooth Terminal" by qwerty as it was free and worked, however you can choose your own.
Once the app has been installed, you will need to pair your Bluetooth module to the smartphone. To do this, enter the Bluetooth menu inside Settings, then search for new devices. After a moment the device "HC06" will appear, for example - see the image in this step.

Tap the HC-09 in the list, and you will then be asked for the PIN - it is 1234. Finally, open your terminal app on the smartphone, and select "Connect a device" from the app menu. Select the HC-06 option and then wait a moment. The LED on the Bluetooth module should stay on and the app will show
"connected: HC-06".

**Link για το Bluetooth Terminal**

https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal&hl=en


11:04
Bluetooth
ON
John moto g
Only visible to paired devices
PAIRED DEVICES
BT Speaker
Hands Free System
John's Mac mini
AVAILABLE DEVICES
HC-06
Flex
SEARCH FOR DEVICES

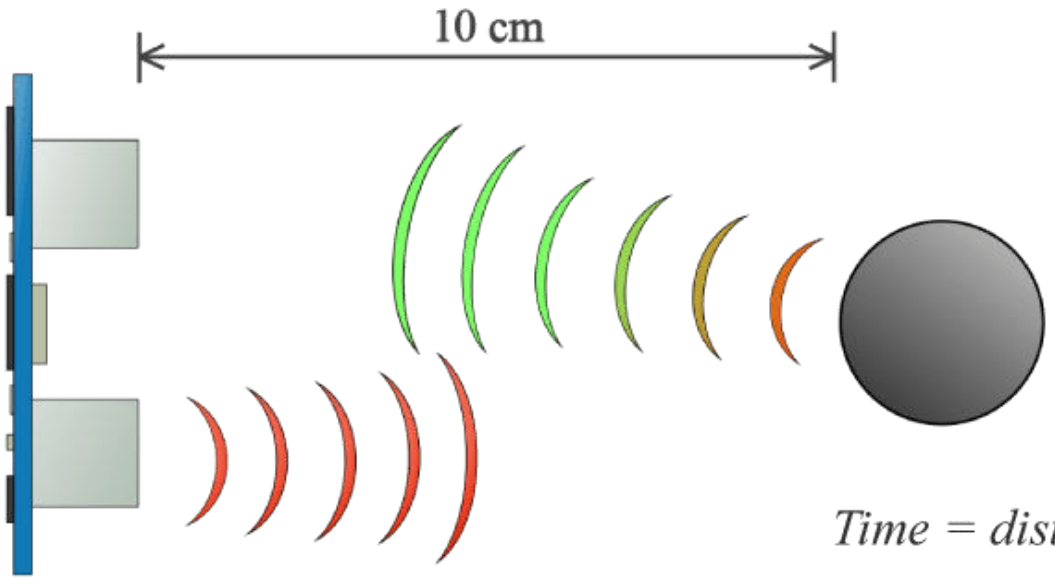

08:32
connected: HC-06
1
LED on
2
LED off
?
Send '1' to turn LED on
Send '2' to turn LED on
1
Send
1 2 3 4 5 6 7 8 9 0

# Σύνδεση αισθητήρα απόστασης με υπερήχους στο Arduino

# Αισθητήρας απόστασης με υπερήχους

Τεχνικά χαρακτηριστικά του αισθητήρα απόστασης

| | |
|---|---|
| Working Voltage | DC 5 V |
| Working Current | 15mA |
| Working Frequency | 40Hz |
| Max Range | 4m |
| Min Range | 2cm |
| MeasuringAngle | 15 degree |
| Trigger Input Signal | 10uS TTL pulse |
| Echo Output Signal | Input TTL lever signal and the range in proportion |
| Dimension | 45*20*15mm |




*speed of sound:*

***v = 340 m/s***

***v = 0,034 cm/µs***

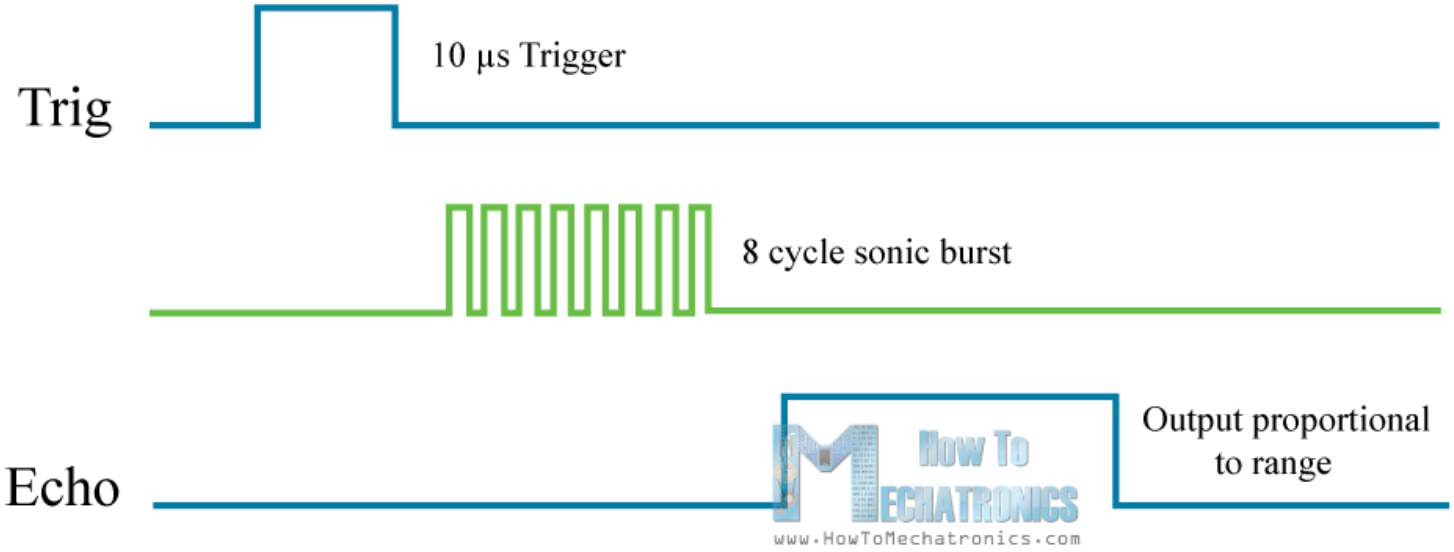
*Time = distance / speed:*

***t = s / v = 10 / 0,034 = 294 µs***

*Distance:*

***s = t · 0,034 / 2***

## Λειτουργία του αισθητήρα απόστασης



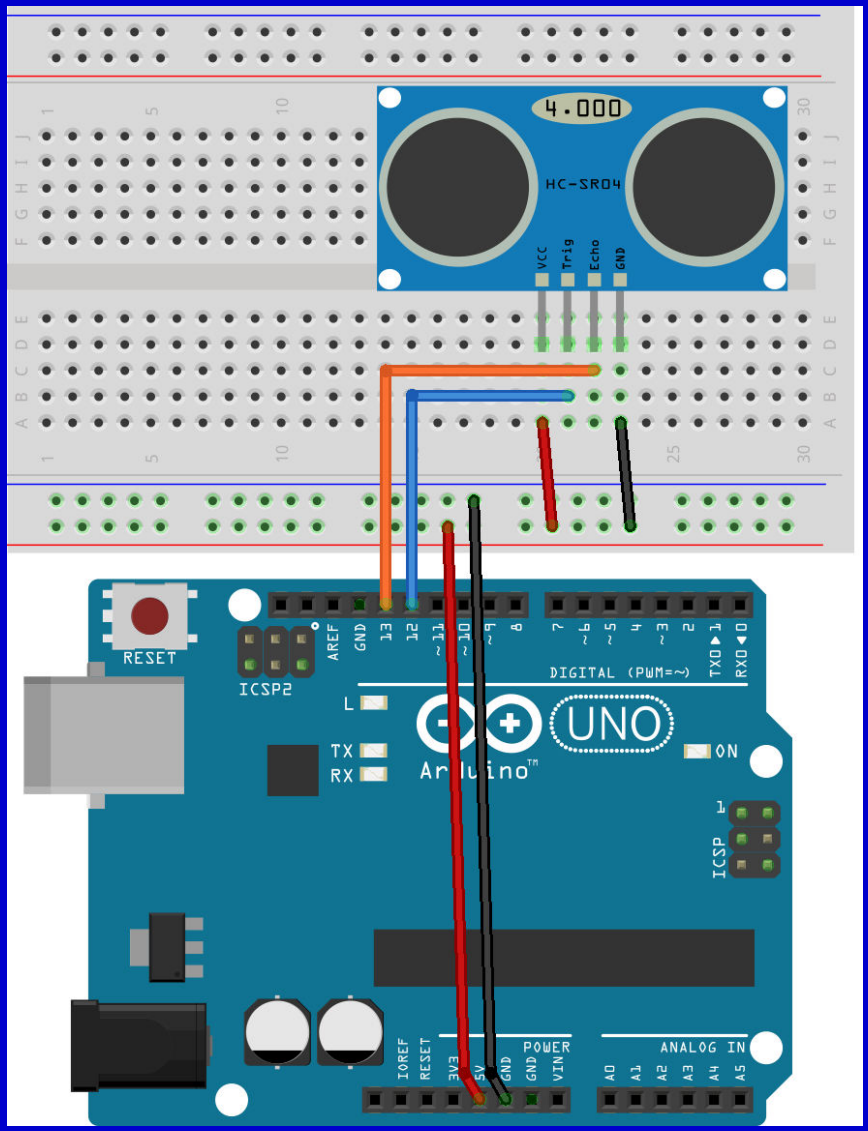
Σύνδεση του αισθητήρα απόστασης με το Arduino

Trig → 9 pin

Echo → 10 pin

GND →GND

Vcc → 5 Volt



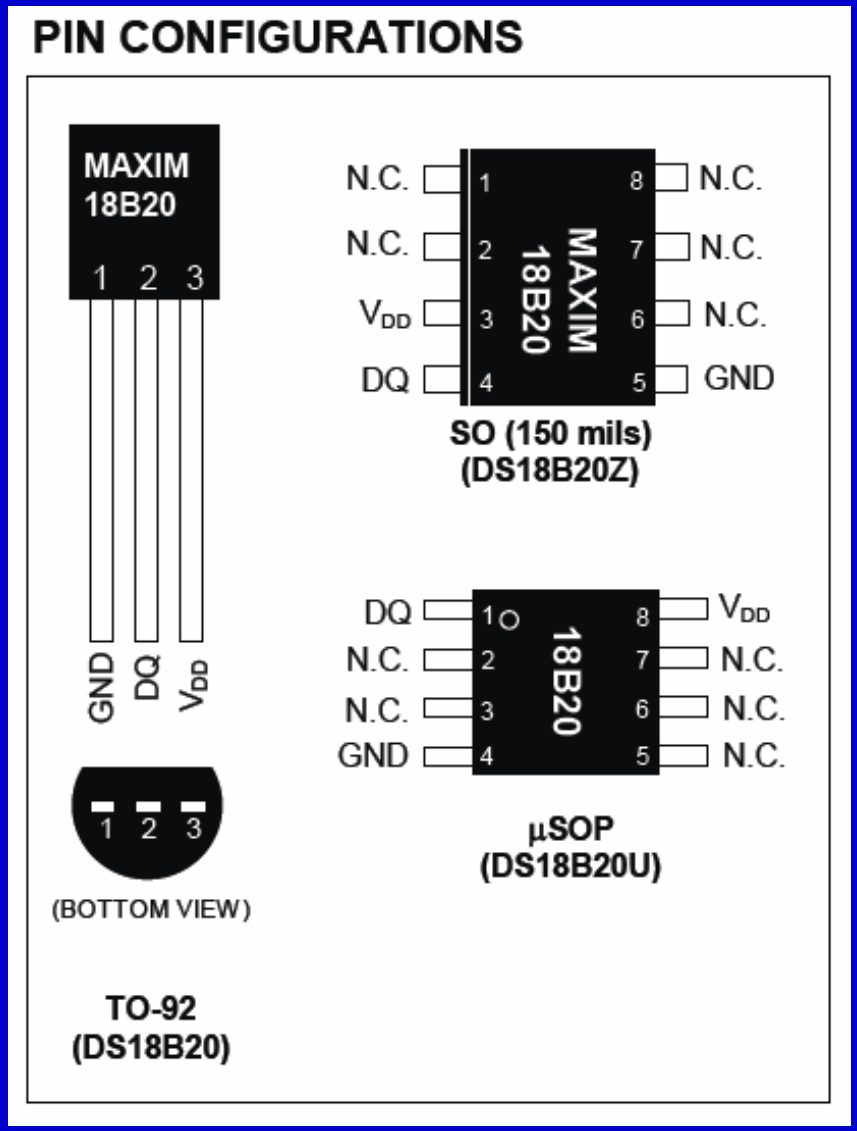
## Πρόγραμμα μέτρησης της απόστασης

```
//defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
//defines variables
long duration;
int distance;

 void setup() {
   pinMode(trigPin, OUTPUT);
   // Sets the trigPin as an Output
   pinMode(echoPin, INPUT);
   // Sets the echoPin as an Input
   Serial.begin(9600);
   // Starts the serial communication
 }
```

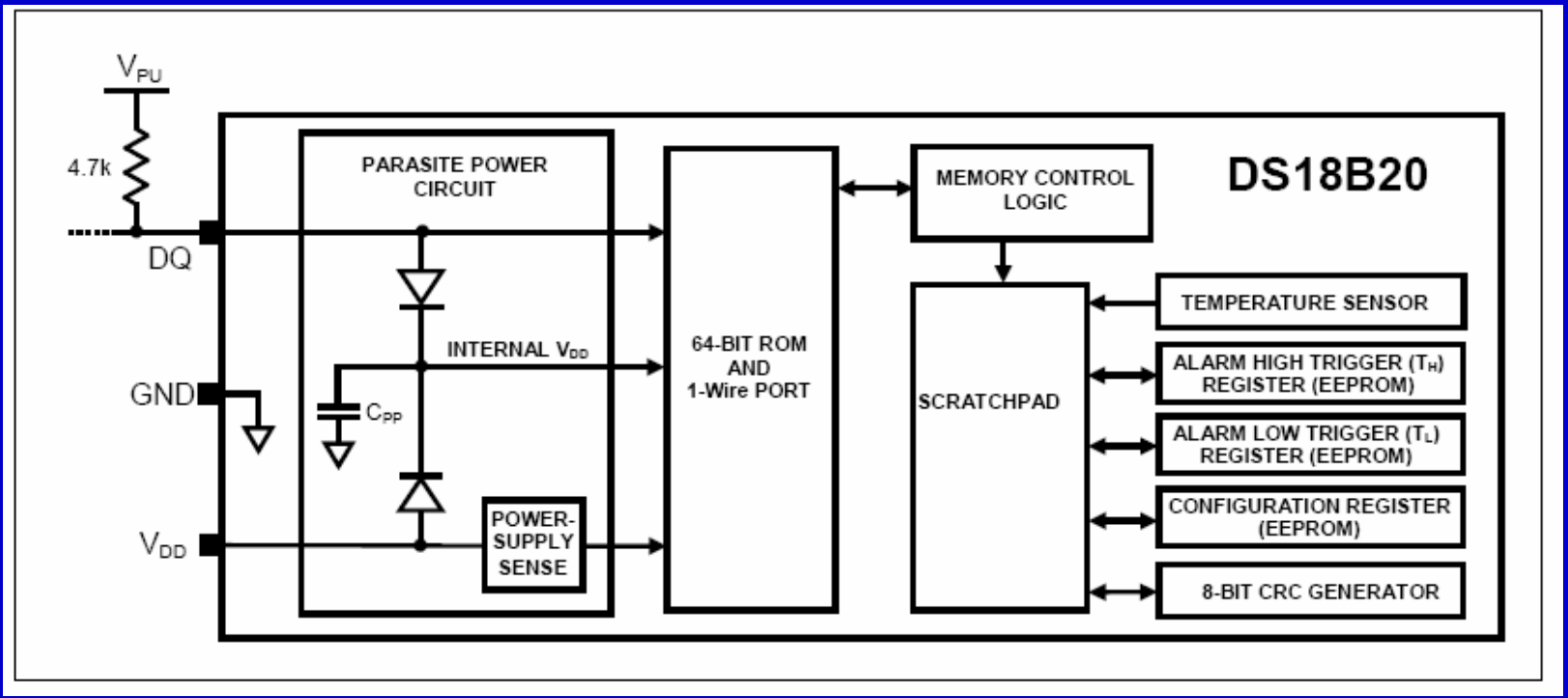
```
void loop() {
  //Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  //Sets the trigPin on HIGH state for 10 ms
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  //Reads the echoPin, returns the sound wave
  // travel time in microseconds
  duration=pulseIn(echoPin, HIGH);
  //Calculating the distance
  distance= duration*0.034/2;
  //Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```

# Σύνδεση ψηφιακού αισθητήρα θερμοκρασίας στο Arduino

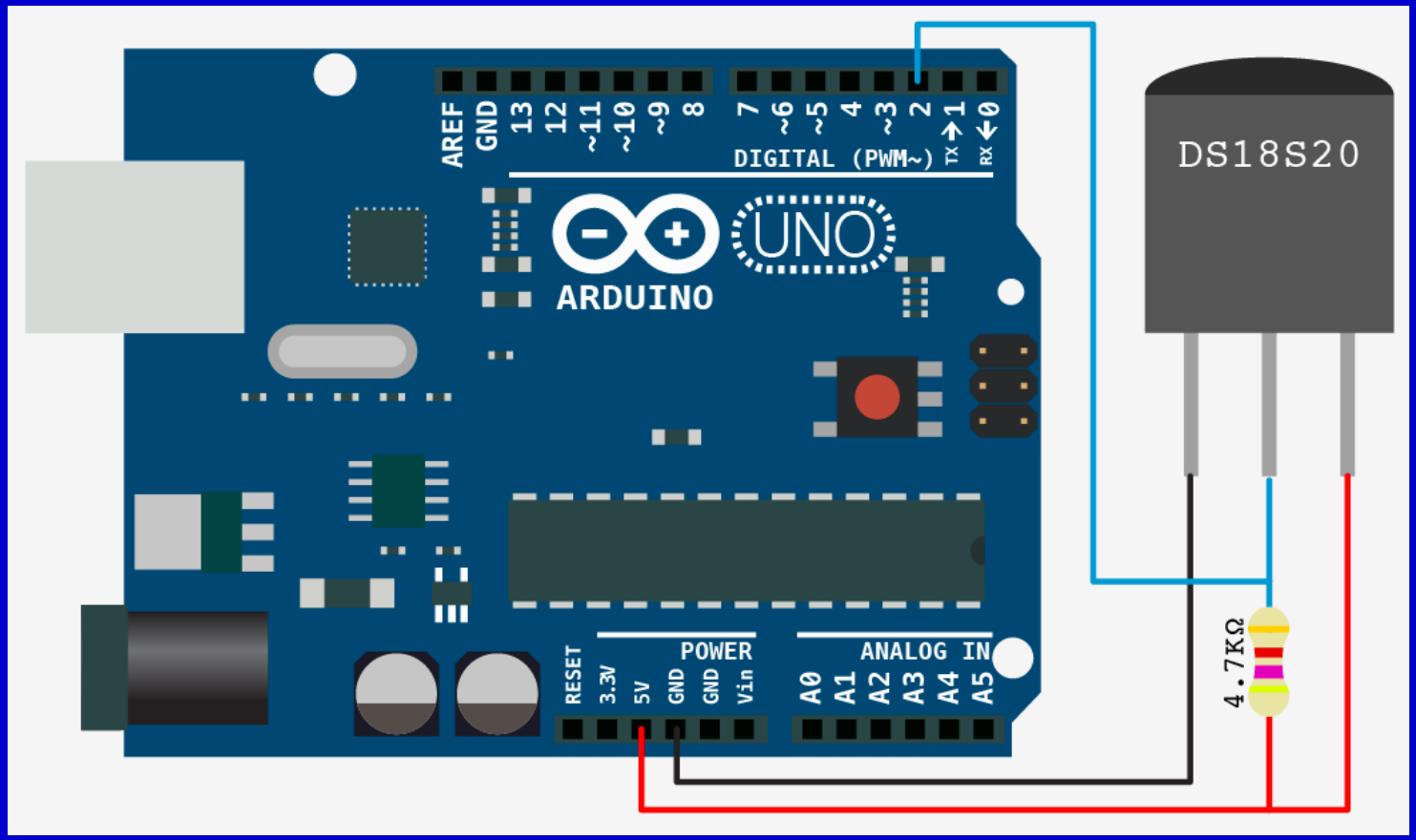
# Αισθητήρας θερμοκρασίας DS18B20



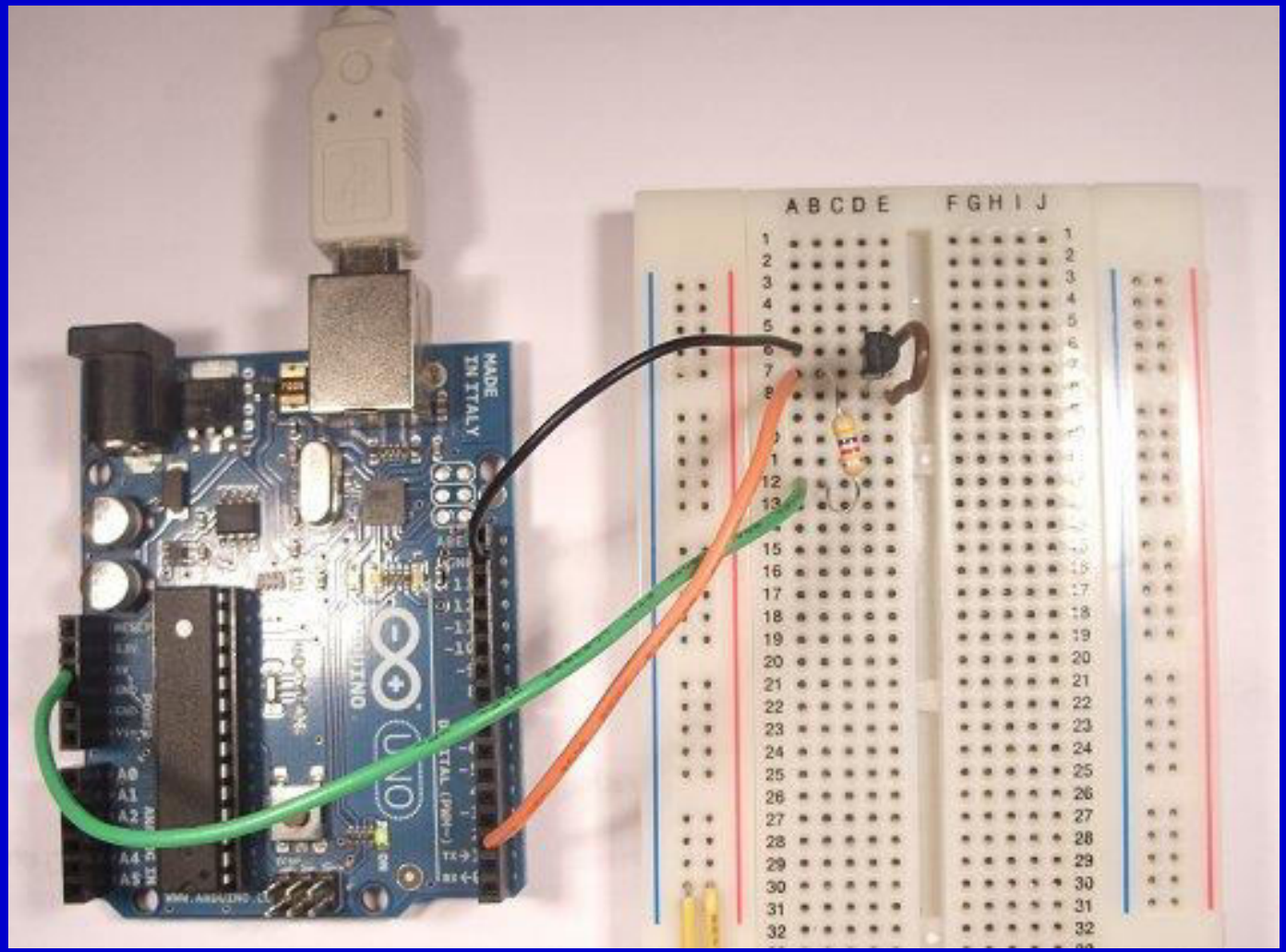
# Δομή του αισθητήρας θερμοκρασίας MAXIM 18B20



# Σύνδεση του αισθητήρα θερμοκρασίας στο Arduino



# Σύνδεση του αισθητήρα θερμοκρασίας στο Arduino



## Library του αισθητήρα DS18B20 για το Arduino

The one wire Digital Temperature Sensor-DS18B20 from Maxim (formerly Dallas) is a chip for measuring temperature. Luckily, there is a Dallas Temperature library for the Arduino which makes using this sensor very easy.
The most recent version of this library can be found at

**Download a copy of the Library for DS 18B20**

https://github.com/milesburton/Arduino-Temperature-Control-Library

**Download a copy of the OneWire library**

https://github.com/PaulStoffregen/OneWire

Drag and drop the libraries into your **arduino/libraries folder.**

```
#include <OneWire.h>
#include <Arduino-Temperature-Control-Library>
// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2
// Setup a oneWire instance to communicate with any OneWire
device
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup(void) {
// start serial port
Serial.begin(9600);
Serial.println("Dallas Temperature IC Control Library Demo");
// Start up the library
sensors.begin();
}
```

```
void loop(void)
{
  // call sensors.requestTemperatures() to issue a global
     temperature
  // request to all devices on the bus
  Serial.print(" Requesting temperatures...");
  sensors.requestTemperatures();
  // Send the command to get temperatures
  Serial.println("DONE");
  Serial.print("Temperature for Device 1 is: ");
  Serial.print(sensors.getTempCByIndex(0));
  // Why "byIndex"?
  // You can have more than one IC on the same bus.
  // 0 refers to the first IC on the wire
 }
```

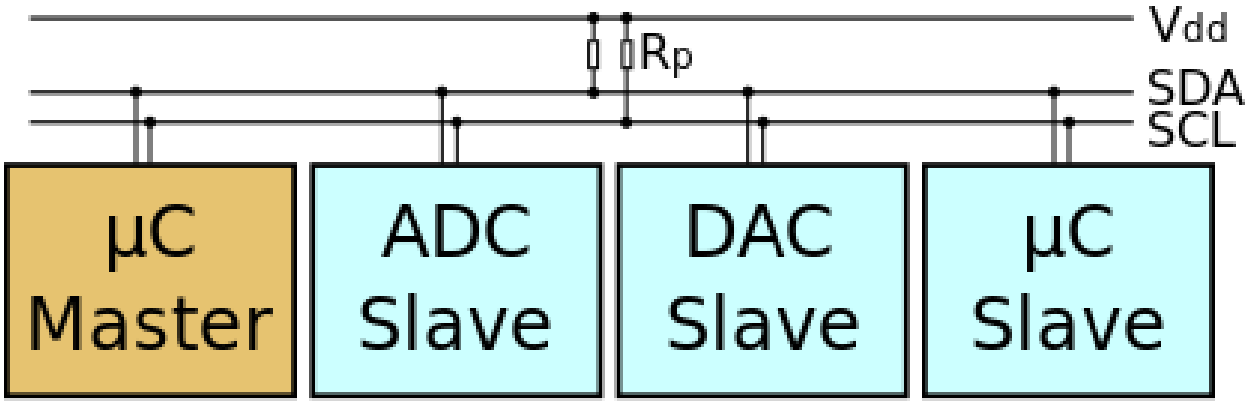
## Έξοδος του προγράμματος

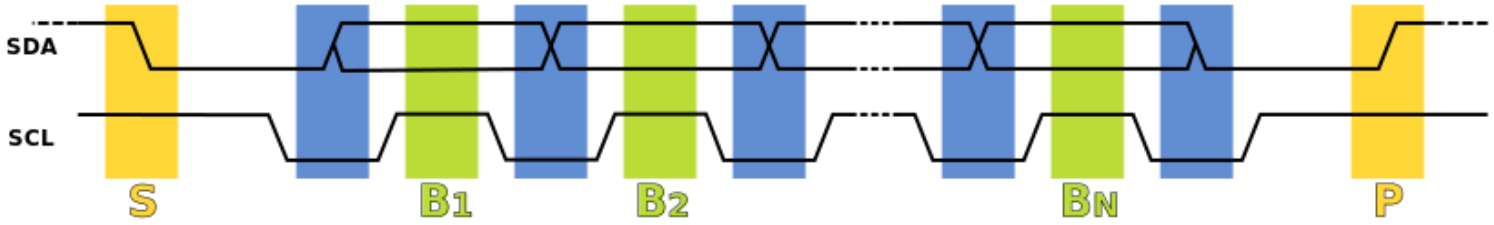
Dallas Temperature IC Control Library Demo
Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...DONE
Temperature for Device 1 is: 24.37 Requesting temperatures...

**Δίαυλοι TWI or I2C**

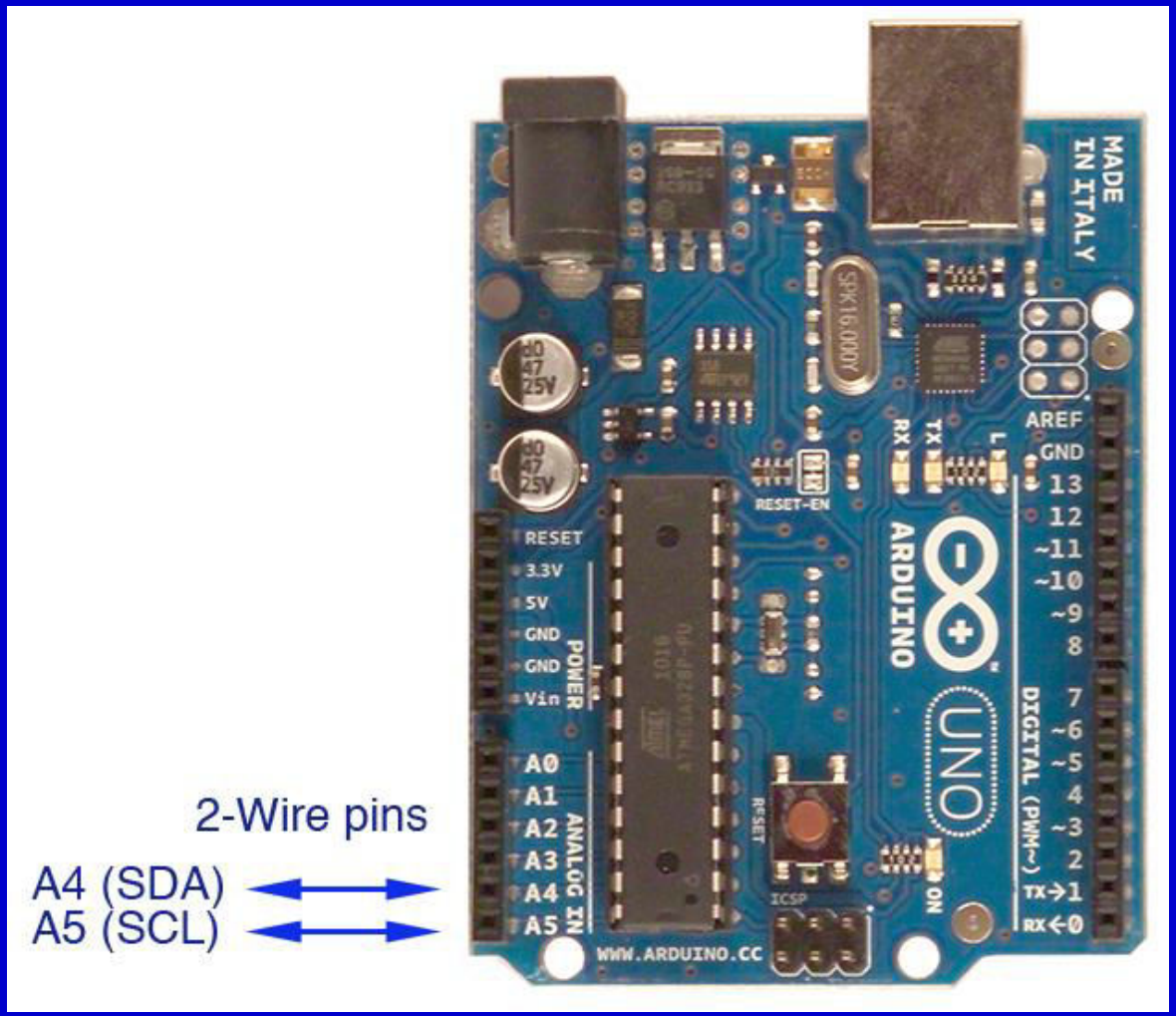
I2C (*Inter-Integrated Circuit*) is a multi-master, multi-slave, single-ended, serial computer bus invented by Philips Semiconductor (now NXP Semiconductors). It is typically used for attaching lower-speed peripheral ICs to processors and microcontrollers.

Δίαυλος I2C


Vdd
Rp
SDA
SCL
μC
Master
ADC
Slave
DAC
Slave
μC
Slave


SDA
SCL
S
B1
B2
BN
P

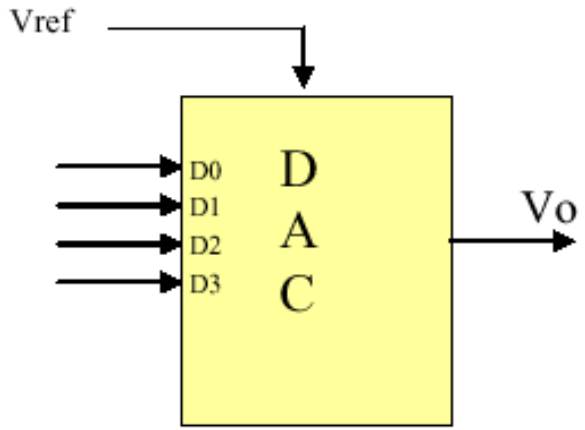
# Ακίδες I2C του Arduino



**Digital-to-analog converter**

Ο *digital-to-analog converter* (DAC, D/A, D2A or D-to-A) είναι ένα σύστημα που μετατρέπει ψηφιακά δεδομένα (συνήθως δυαδικά) σε αναλογικό σήμα. Ο analog-to-digital converter (ADC) κάνει την αντίστροφη λειτουργία. Αντίθετα με τα αναλογικά σήματα τα ψηφιακά δεδομένα μπορούν να μεταδοθούν, να χειρισθούν και να αποθηκευτούν εύκολα και χωρίς παραμορφώσεις. Χρειάζεται όμως ένας DAC για να μετατρέψει το ψηφιακό σήμα σε αναλογικό για να χρησιμοποιηθεί σαν είσοδος σε ακουστικά ή σε ενισχυτή ήχου.

# DAC των 4 bit



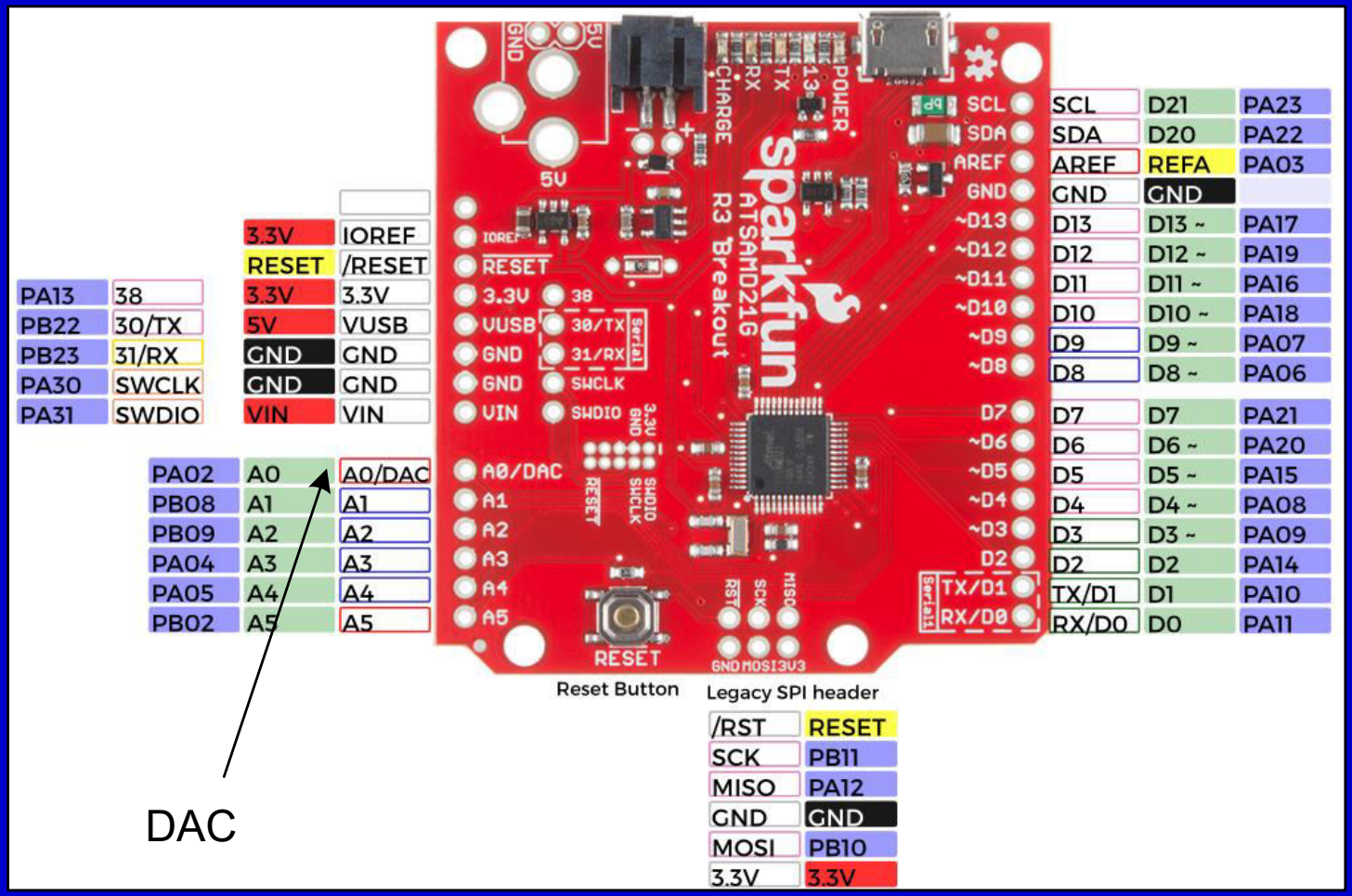

$$Vo = Vref\left(\frac{D3}{2} + \frac{D2}{4} + \frac{D1}{8} + \frac{D0}{16}\right) \text{ or } \qquad Vo = Vref\left(\frac{D}{2^n}\right) = Vref\left(\frac{D}{16}\right)$$

where D= decimal value of the digital input.

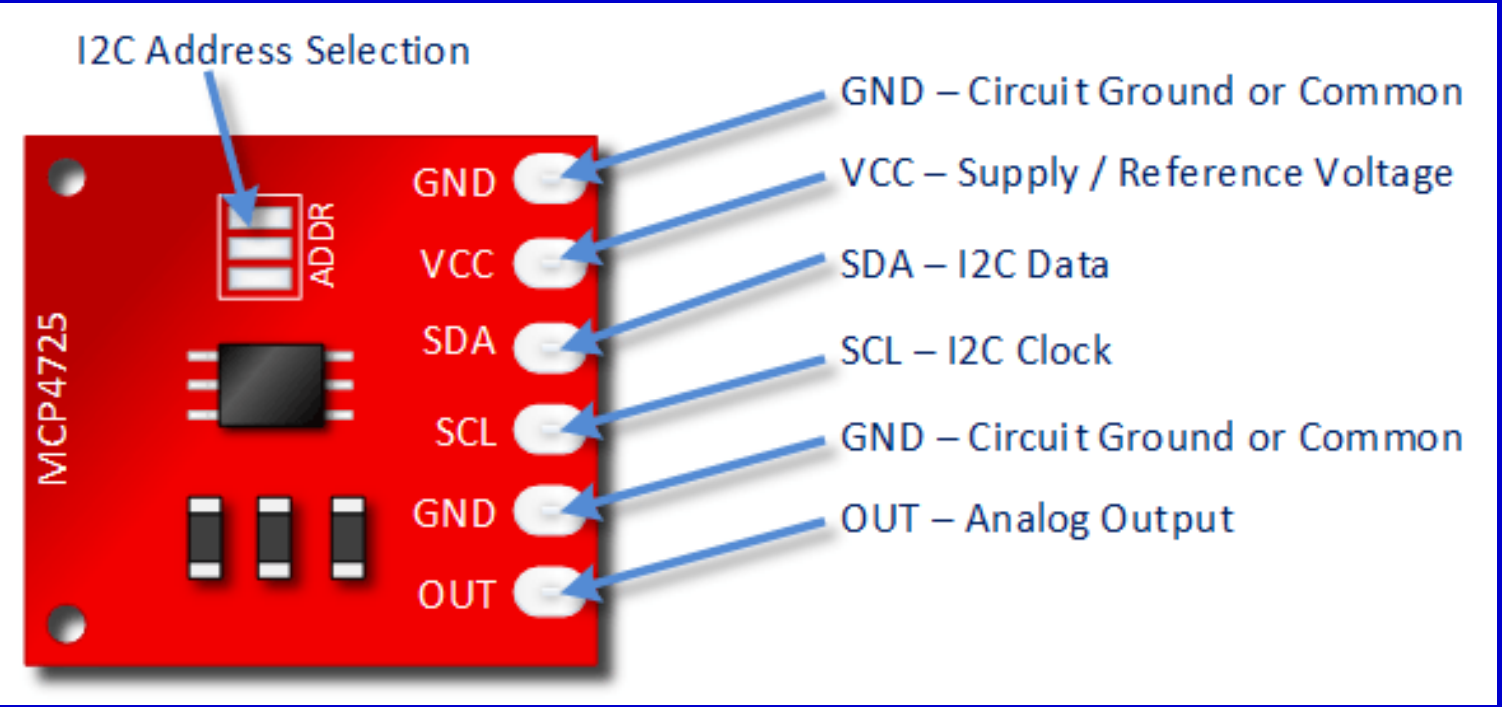
## DAC of Arduino Zero

The Arduino Zero has the following hardware capabilities: 10 pins which default to 8-bit PWM, like the AVR-based boards. These can be changed to 12-bit resolution.
1 pin with 10-bit DAC (Digital-to-Analog Converter).
By setting the write resolution to 10, you can use analogWrite() with values between 0 and 1023 to exploit the full DAC resolution.

# DAC of Arduino zero



# DAC MCP4725

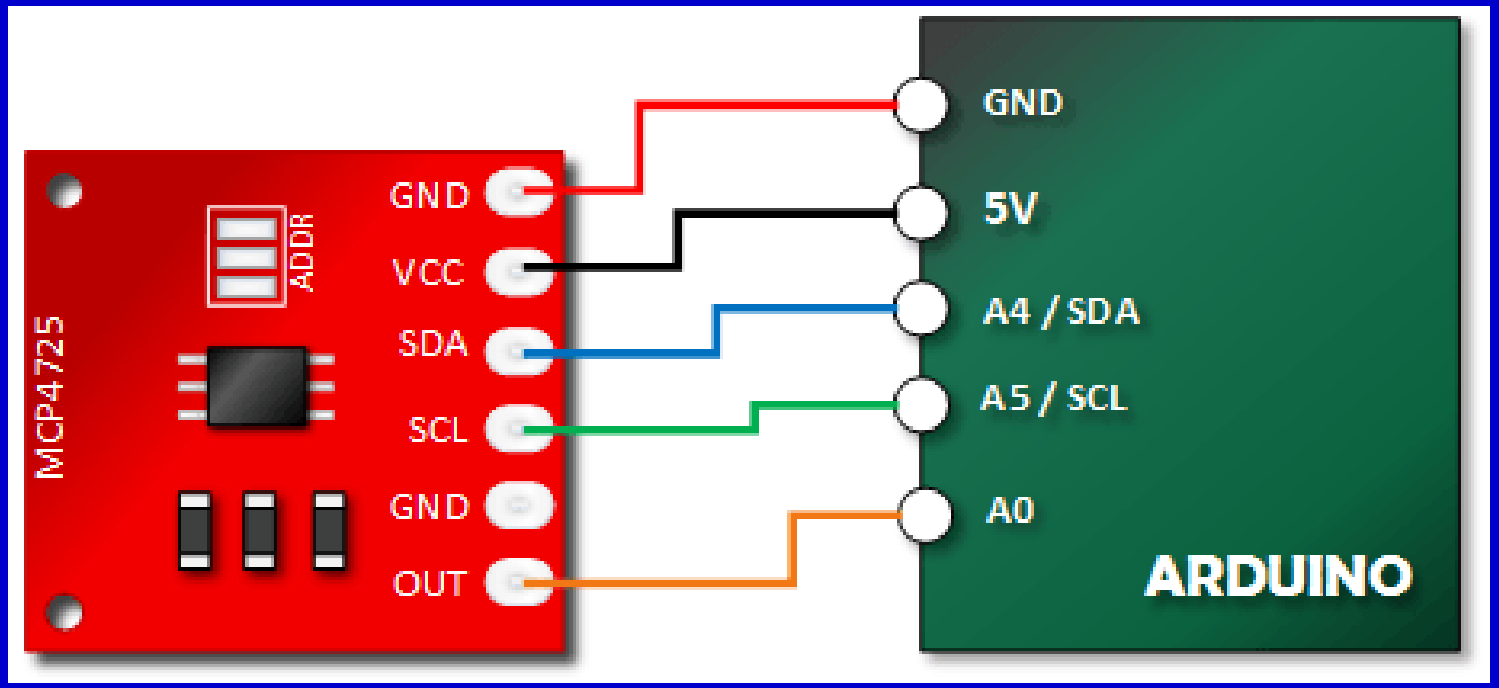



Address 0x60

## DAC MCP4725


Address 0x62.

## Σύνδεση DAC με Arduino



**Library του MCP4725**

https://github.com/adafruit/Adafruit_MCP4725

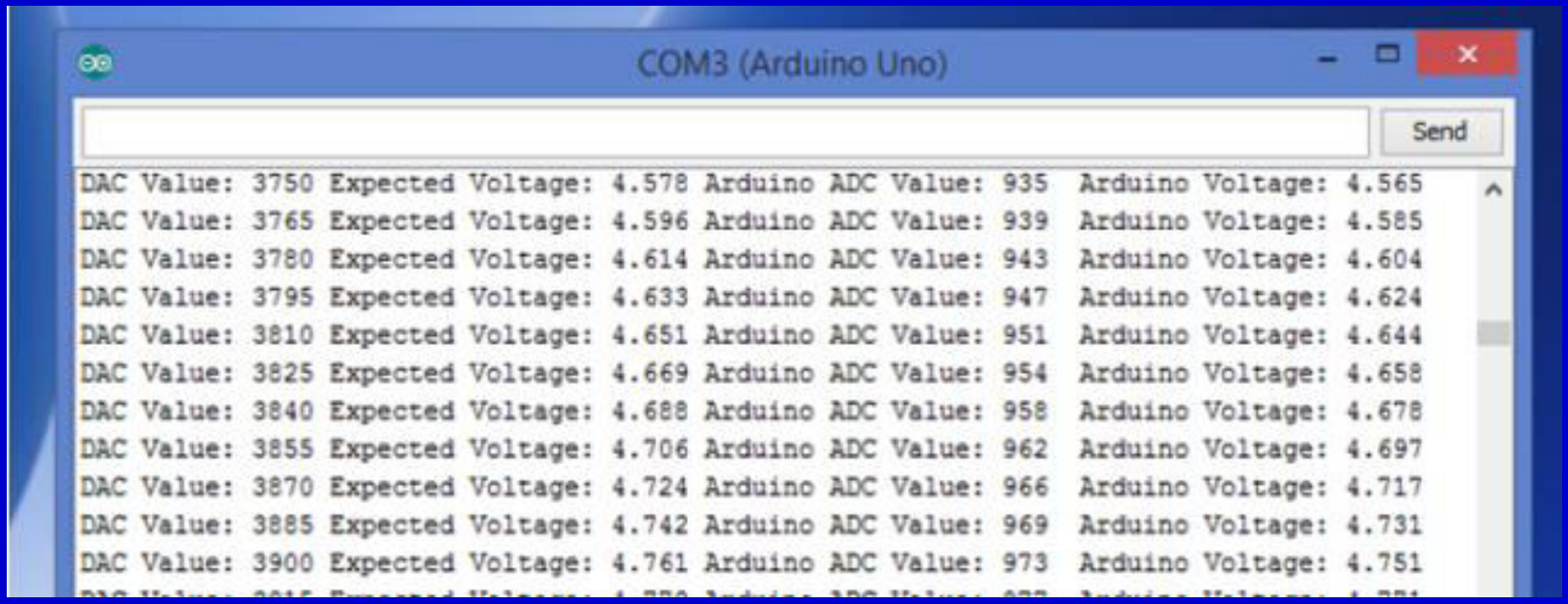
– We’re incrementing the DAC by 15 from zero to full scale.
– We’re measuring the output of the DAC using an analog pin.
– We’re calculating what we think we have as an output from the DAC and what the Arduino claims we have.
– We’re displaying the results.

```
#Include <Wire.h>
#include <Adafruit_MCP4725.h>
#define voltsIn A0
Adafruit_MCP4725 dac; // constructor

void setup(void)
{
  Serial.begin(9600);
  dac.begin(0x60); // The I2C Address
}
```

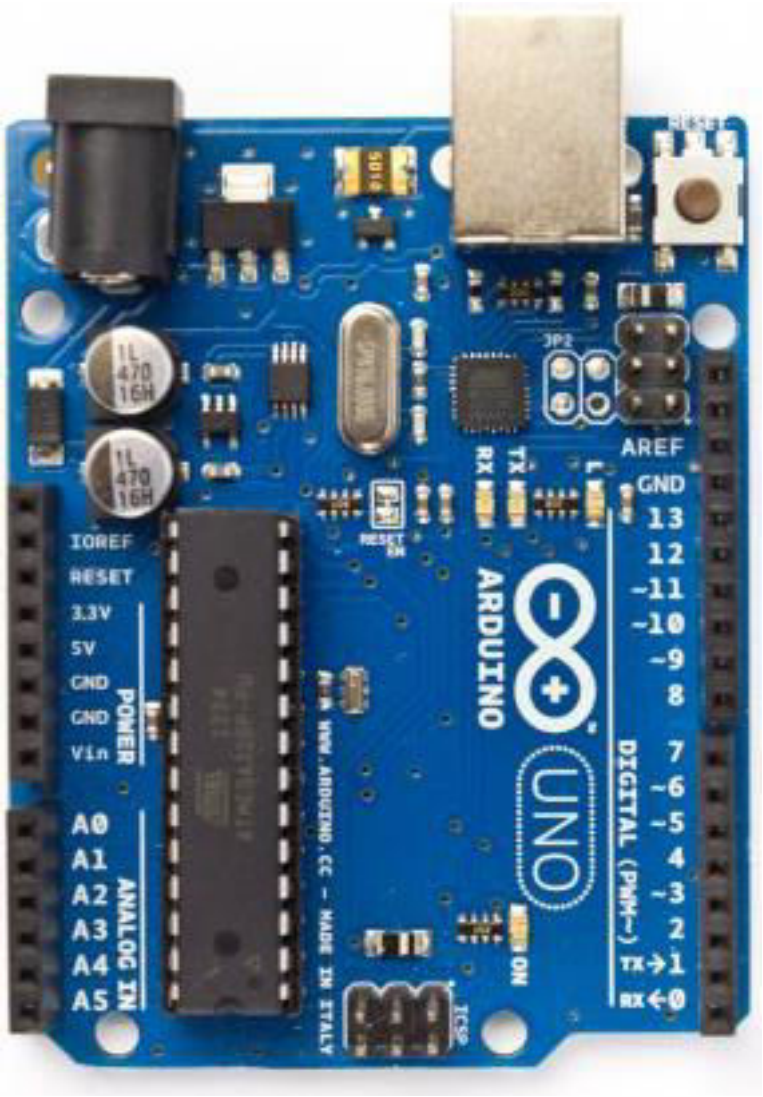
```
void loop(void) {
   uint32_t  dac_value;
   int adcValueRead = 0;

   float voltageRead = 0;
   float dac_expected_output;
   for (dac_value=0; dac_value<4096; dac_value=dac_value +15)
    {
     dac_expected_output = (5.0/4096.0) * dac_value;
     dac.setVoltage(dac_value, false);
     delay(250);
     adcValueRead = analogRead(voltsIn);
     voltageRead = (adcValueRead * 5.0 )/1024.0;
```

```
  Serial.print("DAC Value:");
  Serial.print(dac_value);

  Serial.print("\tExpected Voltage: ");
  Serial.print(dac_expected_output, 3);

  Serial.print("\tArduino ADC Value: ");
  Serial.print(adcValueRead);

  Serial.print("\tArduino Voltage: ");
  Serial.println(voltageRead,3);
 }
}
```

# Έξοδος

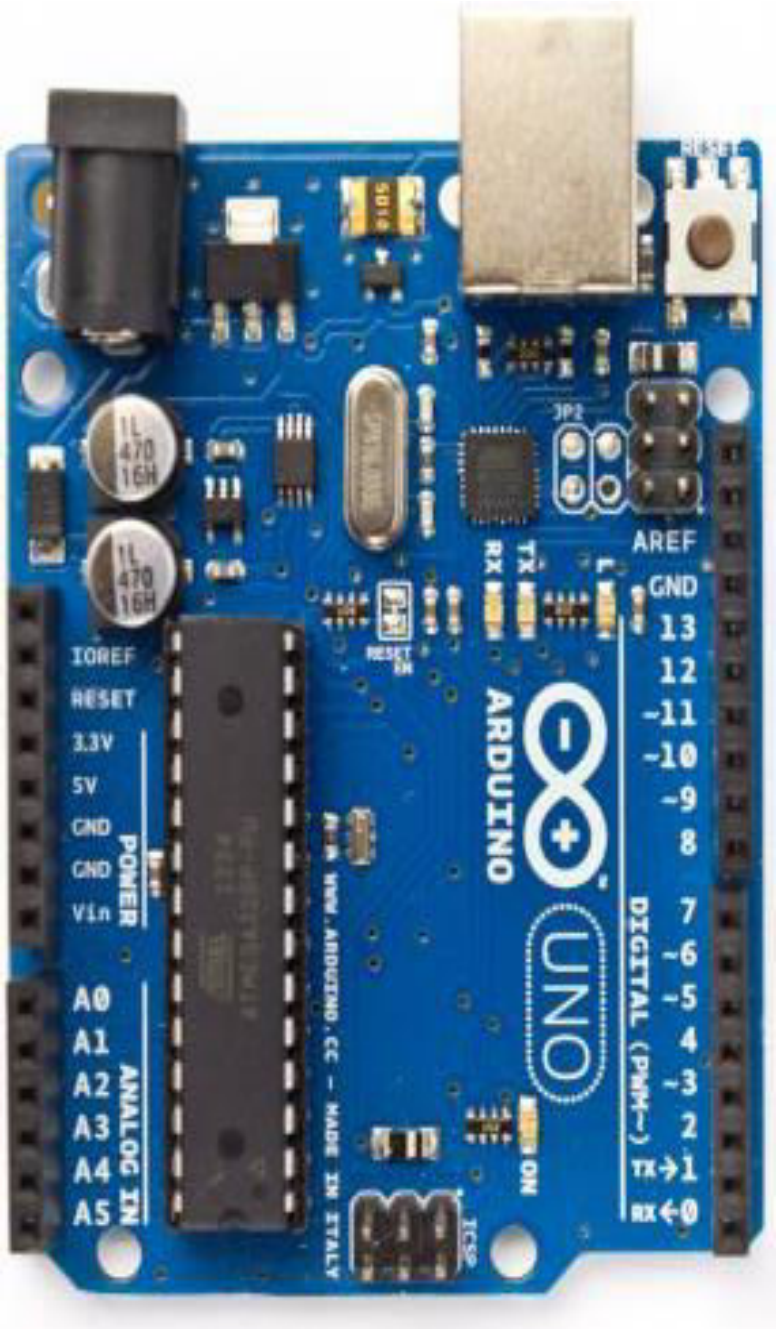

COM3 (Arduino Uno)
Send
DAC Value: 3750 Expected Voltage: 4.578 Arduino ADC Value: 935 Arduino Voltage: 4.565
DAC Value: 3765 Expected Voltage: 4.596 Arduino ADC Value: 939 Arduino Voltage: 4.585
DAC Value: 3780 Expected Voltage: 4.614 Arduino ADC Value: 943 Arduino Voltage: 4.604
DAC Value: 3795 Expected Voltage: 4.633 Arduino ADC Value: 947 Arduino Voltage: 4.624
DAC Value: 3810 Expected Voltage: 4.651 Arduino ADC Value: 951 Arduino Voltage: 4.644
DAC Value: 3825 Expected Voltage: 4.669 Arduino ADC Value: 954 Arduino Voltage: 4.658
DAC Value: 3840 Expected Voltage: 4.688 Arduino ADC Value: 958 Arduino Voltage: 4.678
DAC Value: 3855 Expected Voltage: 4.706 Arduino ADC Value: 962 Arduino Voltage: 4.697
DAC Value: 3870 Expected Voltage: 4.724 Arduino ADC Value: 966 Arduino Voltage: 4.717
DAC Value: 3885 Expected Voltage: 4.742 Arduino ADC Value: 969 Arduino Voltage: 4.731
DAC Value: 3900 Expected Voltage: 4.761 Arduino ADC Value: 973 Arduino Voltage: 4.751

1. Κάντε τις κατάλληλες συνδέσεις μεταξύ του Arduino και του PIR Sensor που δίδονται στην συνέχεια.


IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
WWW.ARDUINO.CC – MADE IN ITALY


3
2
1
GND
OUT
VCC (+5V)

2. Κάντε τις κατάλληλες συνδέσεις μεταξύ του Arduino και του bleutooth module που δίδονται στην συνέχεια.


ARDUINO
UNO
POWER
IOREF
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5
DIGITAL (PWM~)
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
RX
TX
ON
WWW.ARDUINO.CC — MADE IN ITALY


PIO5
3V3
PIO4
WAKEUP
VCC
GND
TXD
RXD
STATE