

## Ψηφιακά Συστήματα Μετρήσεων

Arduino # Raspberry Pi  
Processing # Python # MATLAB

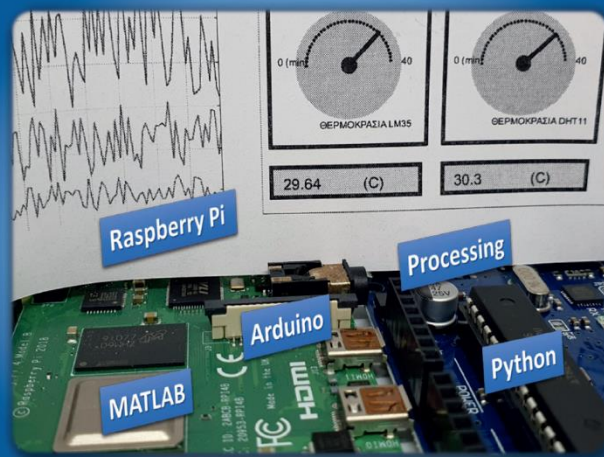
Το βιβλίο αυτό «απαντά» στις πραγματικές ανάγκες ενός μηχανικού, στην κατεύθυνση της ανάπτυξης ενός λειτουργικού και ολοκληρωμένου ψηφιακού συστήματος μετρήσεων. Παρουσιάζει τις επικρατέστερες τεχνολογίες και μεθοδολογίες για την ανάπτυξη των εφαρμογών. Δηλαδή, συγκεντρώνει γνώσεις με μια ενιαία αντίληψη 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)

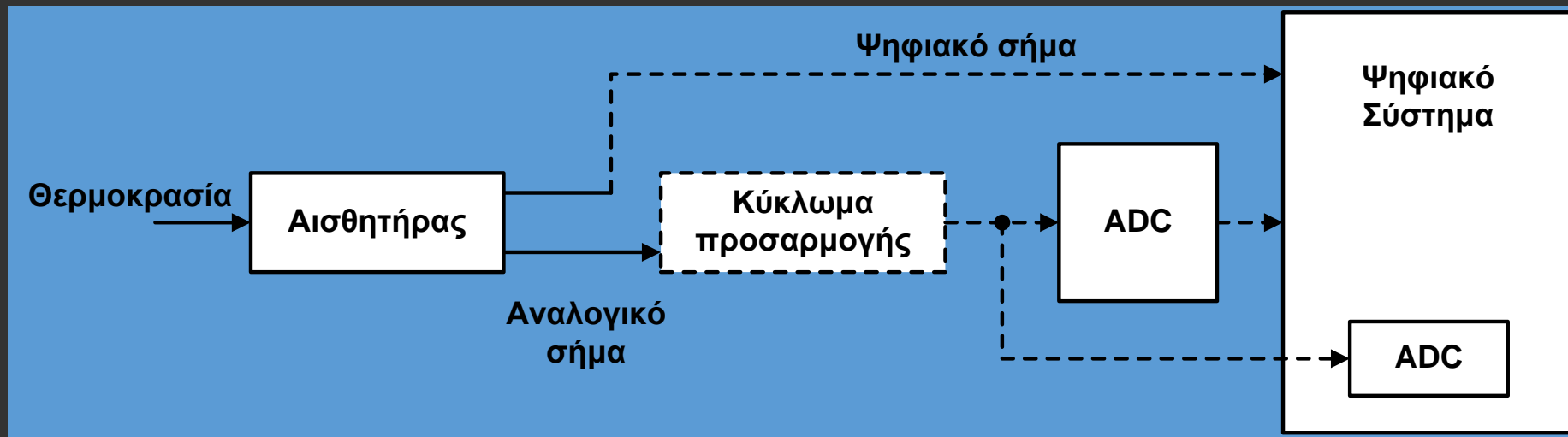
# Κεφάλαιο 5



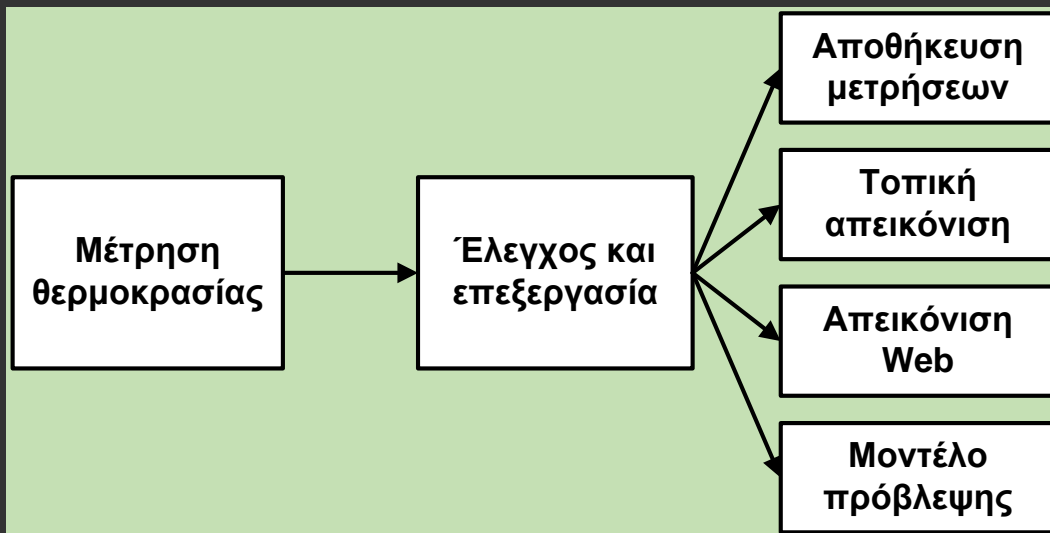
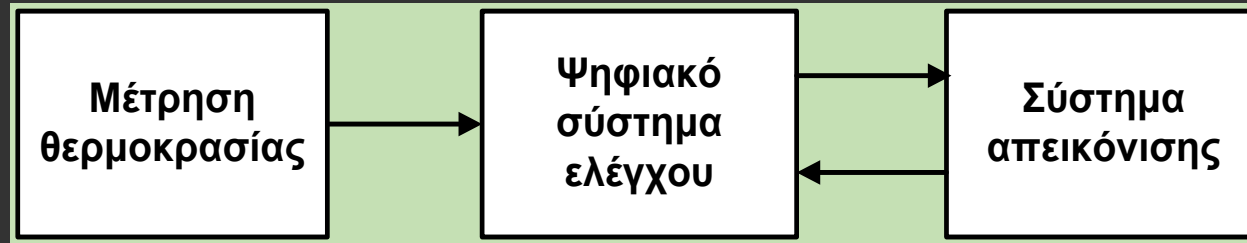
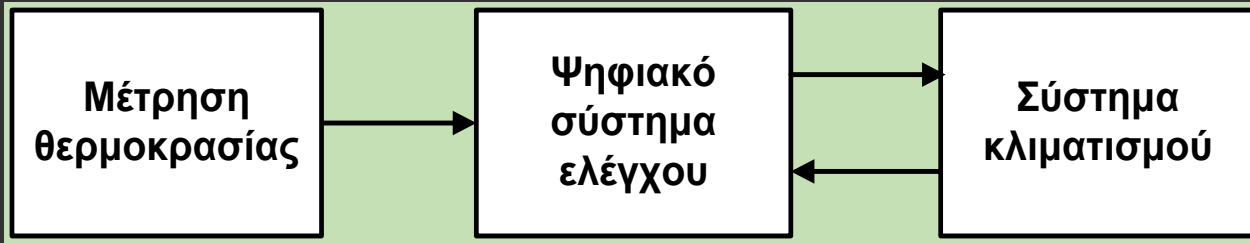
# Μετρήσεις με βασικούς αισθητήρες στο Arduino



# Μέτρηση θερμοκρασίας (1)



# Μέτρηση θερμοκρασίας (2)



Παραδείγματα συστημάτων

# Μέτρηση θερμοκρασίας (3)

## Κλίμακες μέτρησης

- Διεθνές σύστημα μονάδων SI → κλίμακα Κέλβιν ( $^{\circ}\text{K}$ )
- Κλίμακα κελσίου ( $^{\circ}\text{C}$ )
- Κλίμακα Φαρενάιτ ( $^{\circ}\text{F}$ )

## Διαφορές αισθητήρων

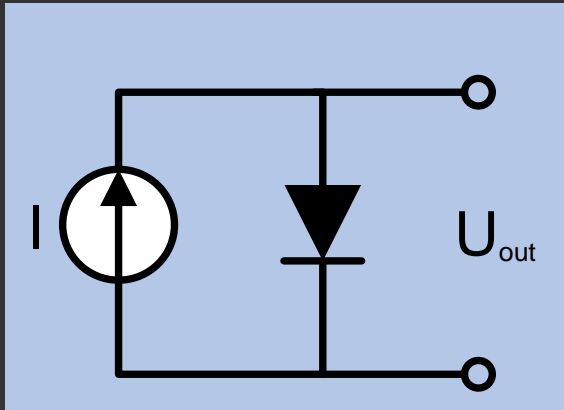
- Εύρος θερμοκρασιών που μετρούν
- Κλίμακα θερμοκρασίας
- Ευαισθησία
- Είδος απόκρισης (γραμμική ή όχι) που αφορά στη σχέση της μεταβολής του φυσικού μεγέθους με το σήμα εξόδου του αισθητήρα
- Ακρίβεια μέτρησης
- Αξιοπιστία
- Κόστος
- Φυσικό μέγεθος

# Μέτρηση θερμοκρασίας (4)

## Δημοφιλείς αισθητήρες

- Αισθητήρες βασισμένοι σε ημιαγωγούς
- Θερμοζεύγη
- Θερμίστορ
- Αισθητήρες αγωγών (RTD)

# Αρχιτεκτονική κυκλωμάτων για αισθητήρες ημιαγωγών (1)



$$I = I_s \left( e^{\frac{q_e U_{out}}{kT}} - 1 \right)$$

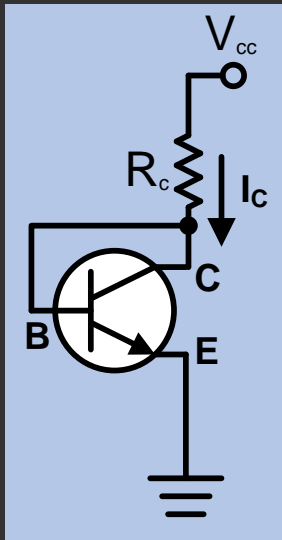
$$U_{out} = \frac{kT}{q_e} \ln \left( \frac{I}{I_s} + 1 \right)$$

$I_s$  : ανάστροφο ρεύμα κορεσμού

$q_e$  : φορτίο του ηλεκτρονίου ( $1.66 \times 10^{-9} \text{C}$ )

$K$  : σταθερά Boltzmann ( $1.38 \times 10^{-23} \text{ J/K}$ )

$T$  : θερμοκρασία σε Kelvin



$$I_c = \frac{V_{CC} - V_{BE}}{R_c}$$

$$I_s \left( e^{\frac{q_e U_{out}}{kT}} - 1 \right) = \frac{V_{CC} - U_{out}}{R_c}$$

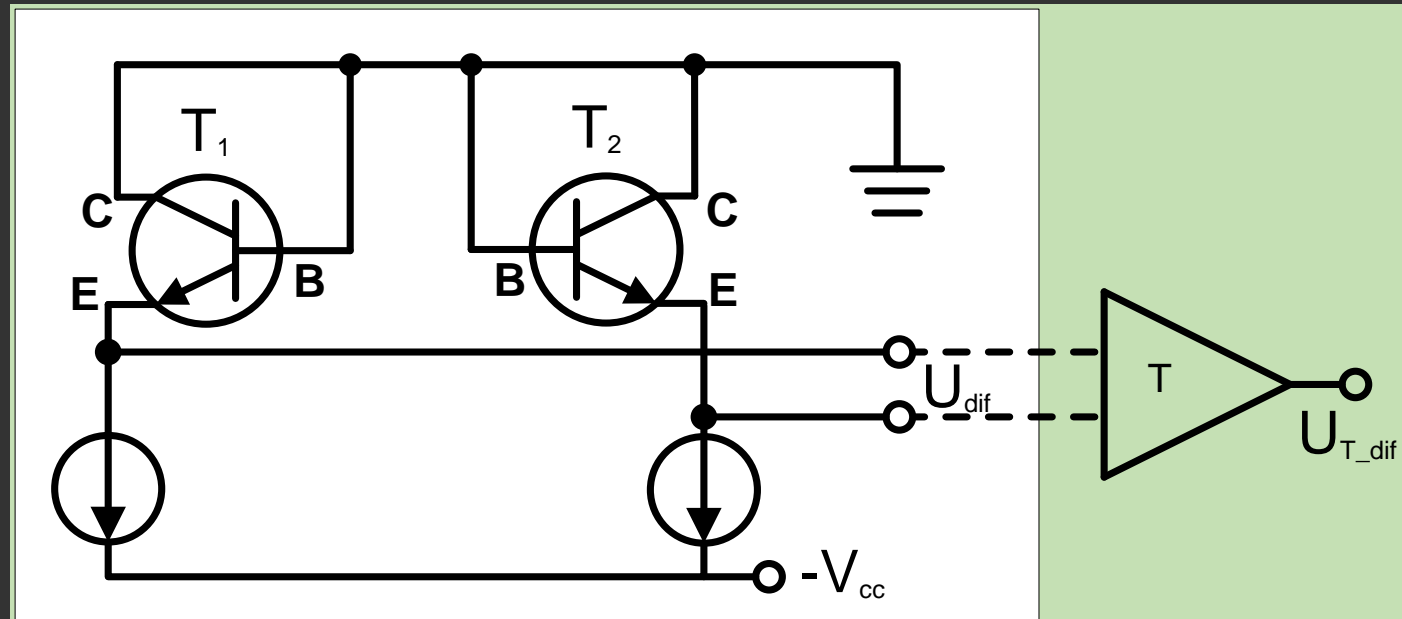
Τρανζίστορ ως δίοδος



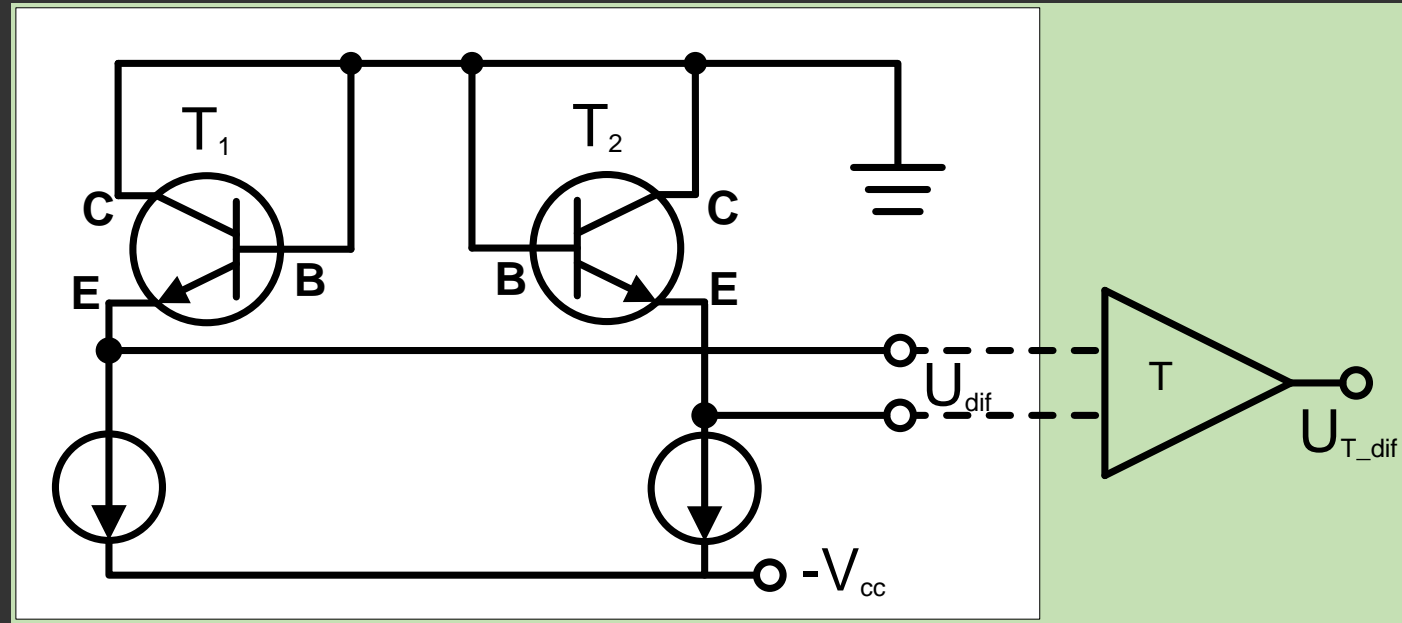
# Αρχιτεκτονική κυκλωμάτων για αισθητήρες ημιαγωγών (2)

- Ρεύμα διόδου & τάση  $\rightarrow$  μεταβάλλονται με τη θερμοκρασία
- Μεταβολή του ρεύματος  $\rightarrow$  ανεπιθύμητη, επηρεάζει τη ζητούμενη γραμμική συμπεριφορά

**ΛΥΣΗ:** Καθρέφτης ρεύματος



# Αρχιτεκτονική κυκλωμάτων για αισθητήρες ημιαγωγών (3)



Αν τα ρεύματα  $I_{S1} = I_{S2}$

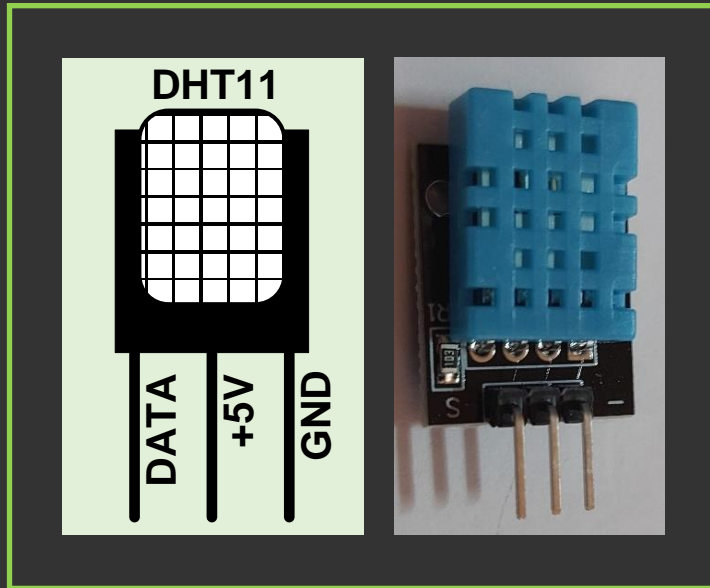
$$U_{dif} = \frac{kT}{q_e} \ln \left( \frac{I_1}{I_2} \right)$$

Σταθερός λόγος  $I_1/I_2 \rightarrow$  διαφορική τάση  $U_{dif}$  ανάλογη της θερμοκρασίας

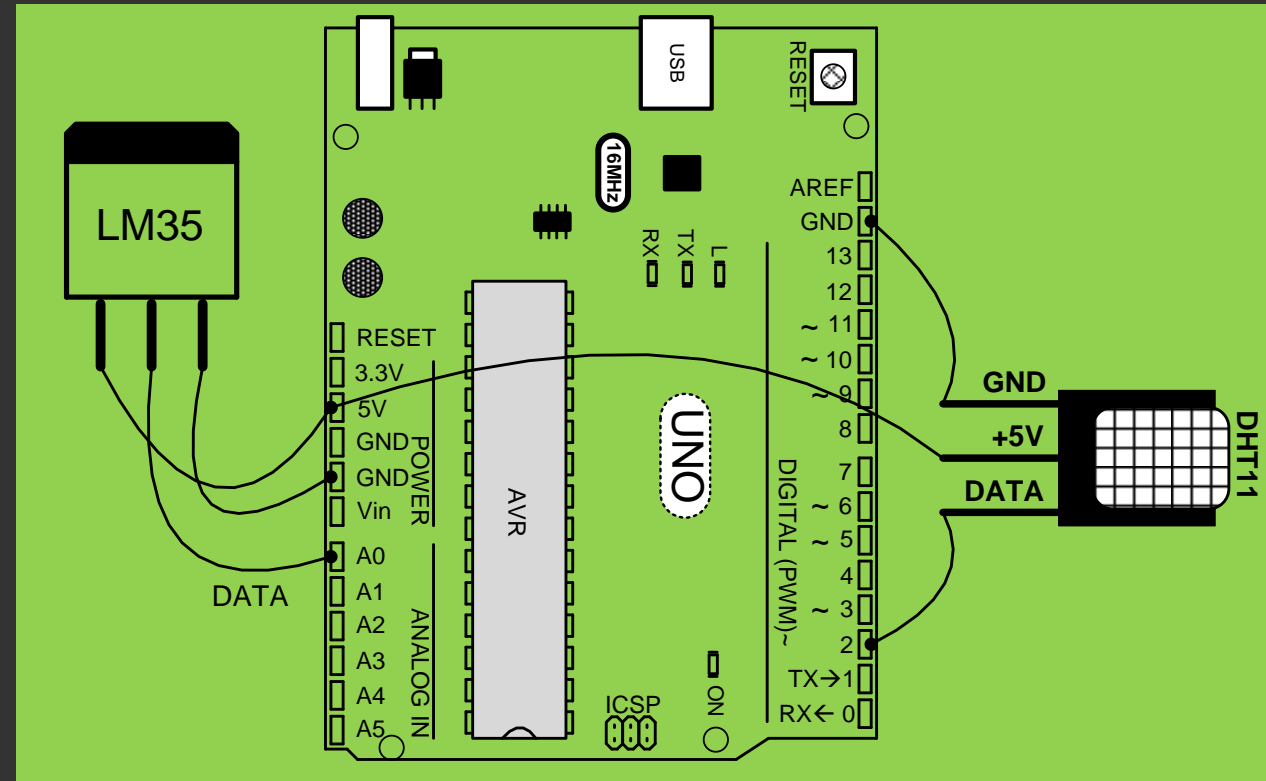




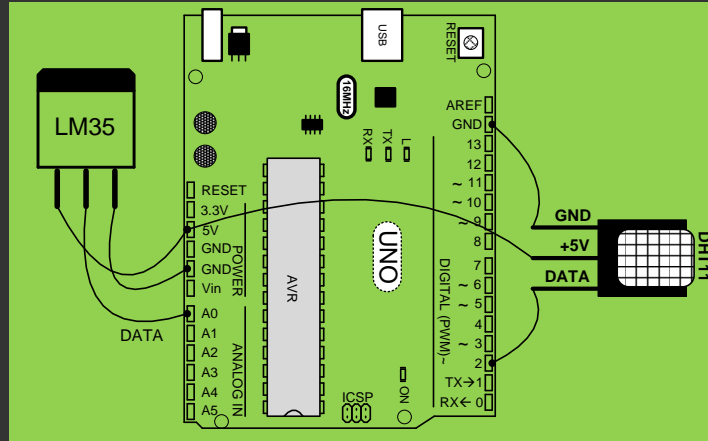
# DHT11 (ψηφιακός) - Arduino



## Κύκλωμα με δύο αισθητήρες



# LM35 & DHT 11 - Κώδικας



```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600);
  dht.begin();
}
```

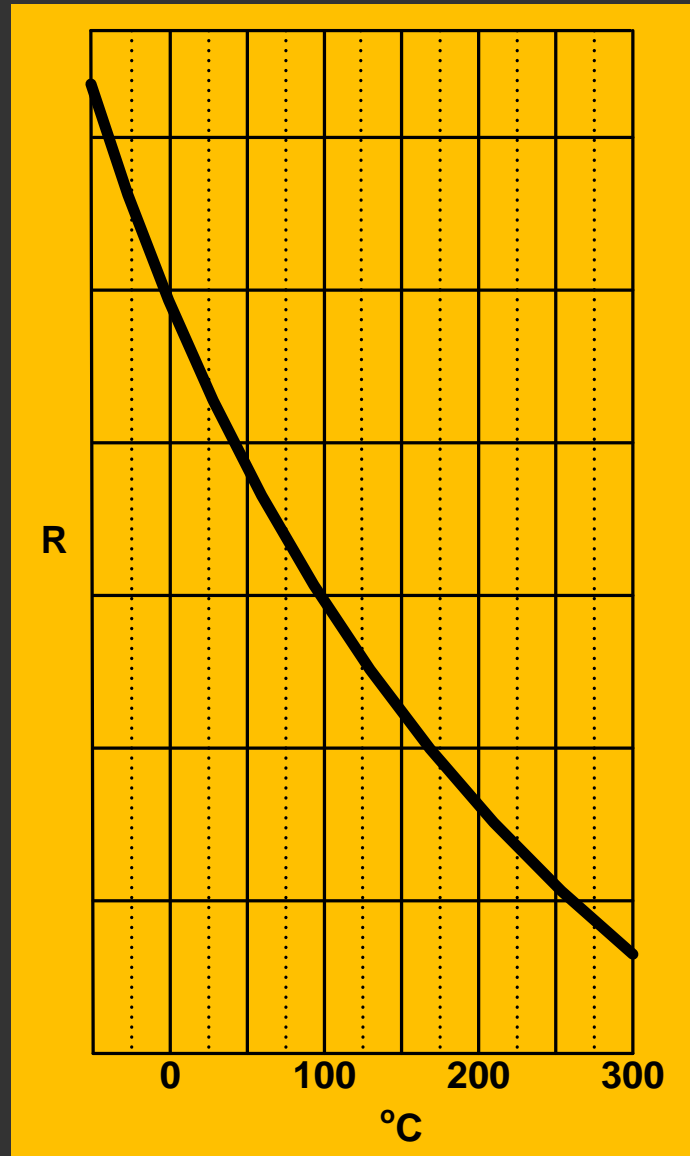
```
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t))
  {
    Serial.println("DHT sensor error");
    return;
  }

  Serial.print("H: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temp: ");
  Serial.print(t);
  Serial.println(" *C ");

  delay(2000);
}
```

# Θερμίστορ

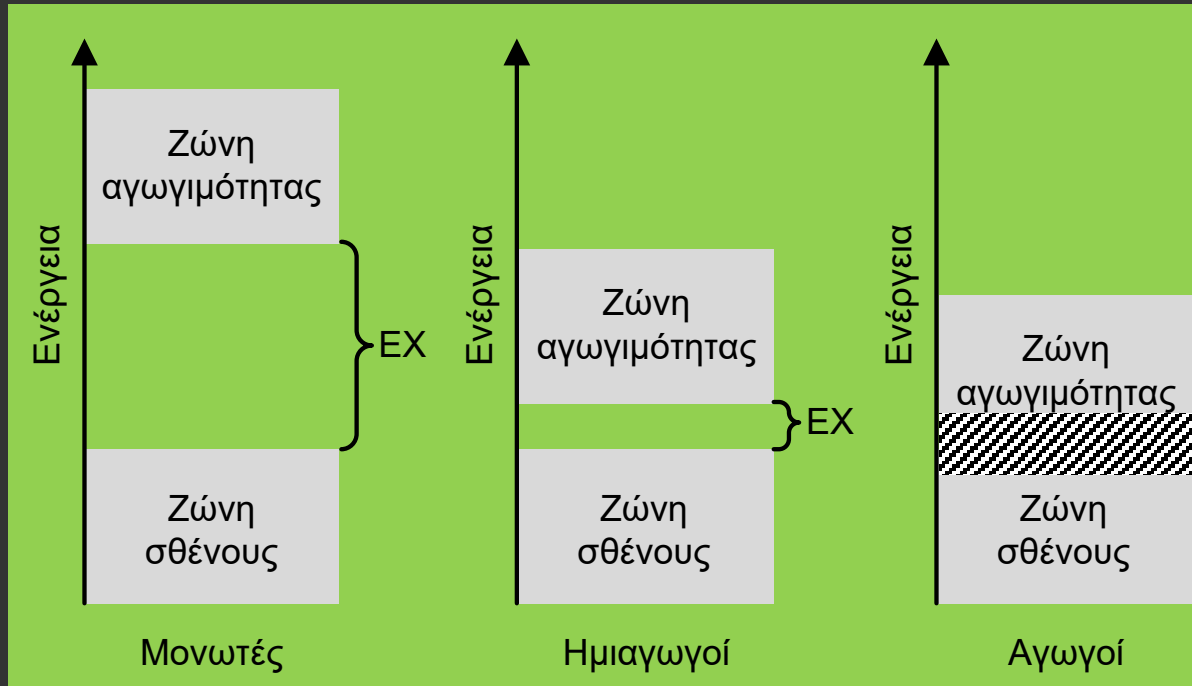


$$R = R_0 e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

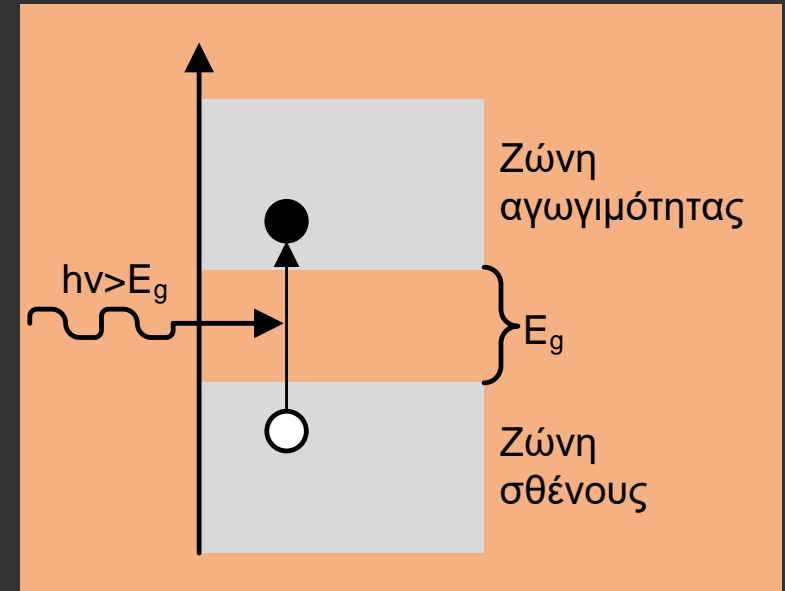
$$T = \frac{1}{\frac{\ln\left(\frac{R}{R_0}\right)}{B} + \frac{1}{T_0}}$$

# Μέτρηση φωτός

## Η Φωτοαντίσταση ως ημιαγωγός



Ενεργειακές στάθμες

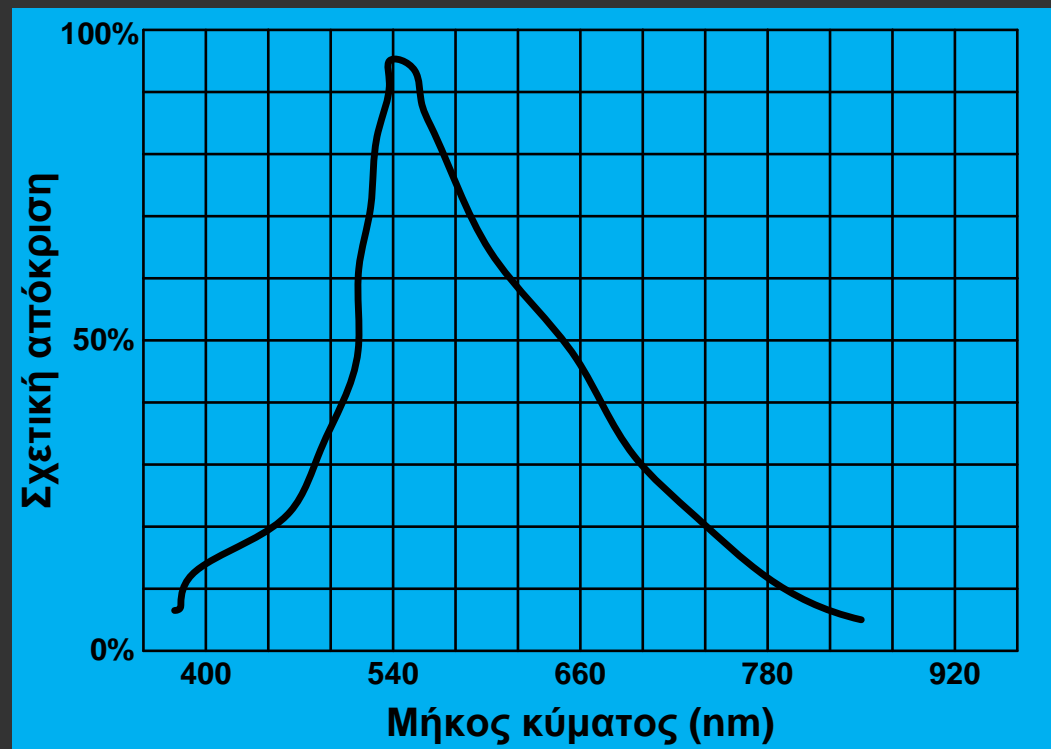
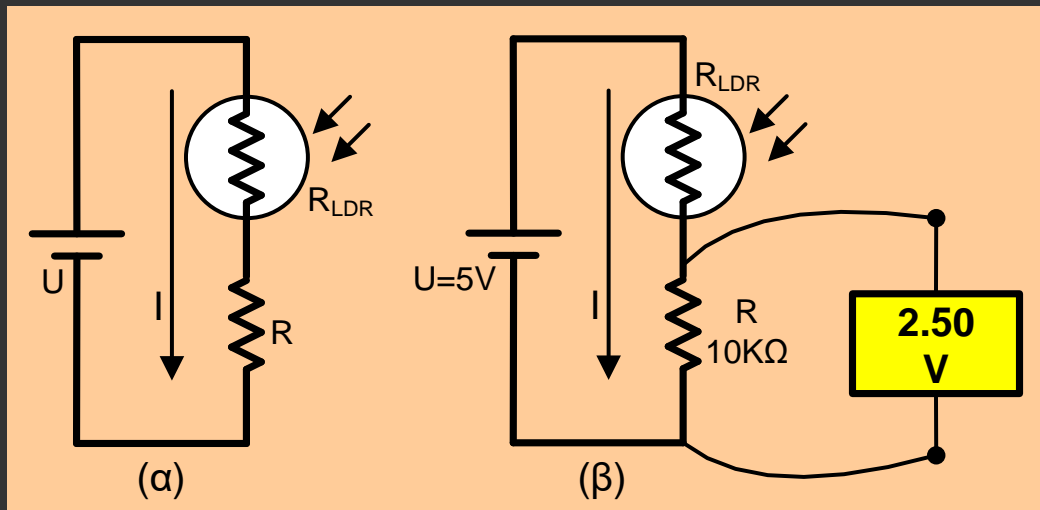


$$E_{ph} = hf = h \frac{c}{\lambda}$$



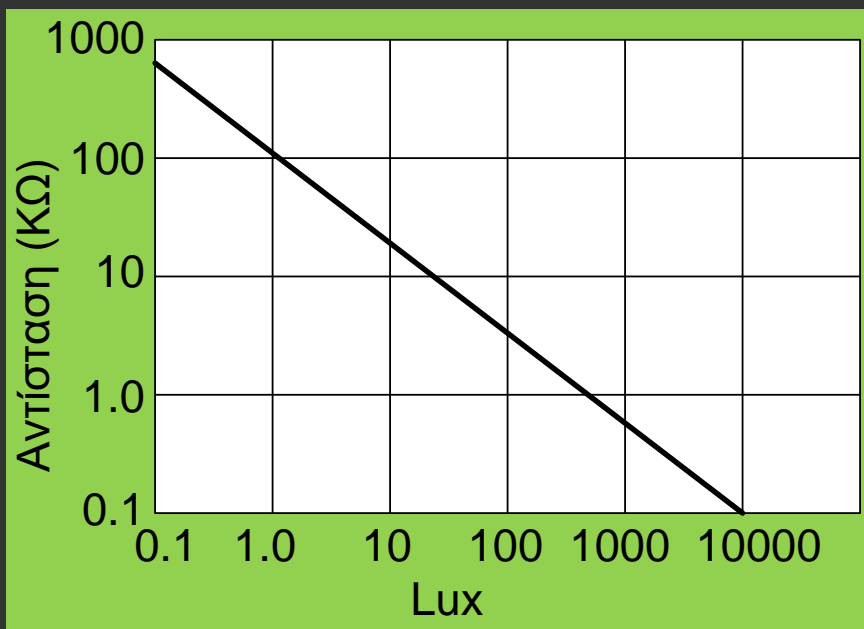
# Μέτρηση φωτός

## Η φωτοαντίσταση σε διαιρέτη τάσης



Απόκριση αισθητήρα

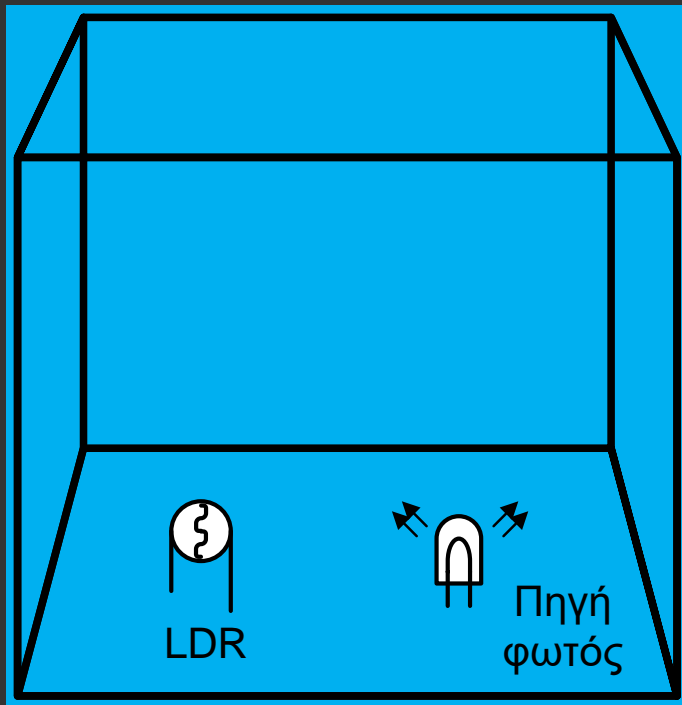
Αντίσταση συναρτήσει της φωτεινότητας



# Μέτρηση φωτός

## Μετρώντας τις συνθήκες φωτισμού (1)

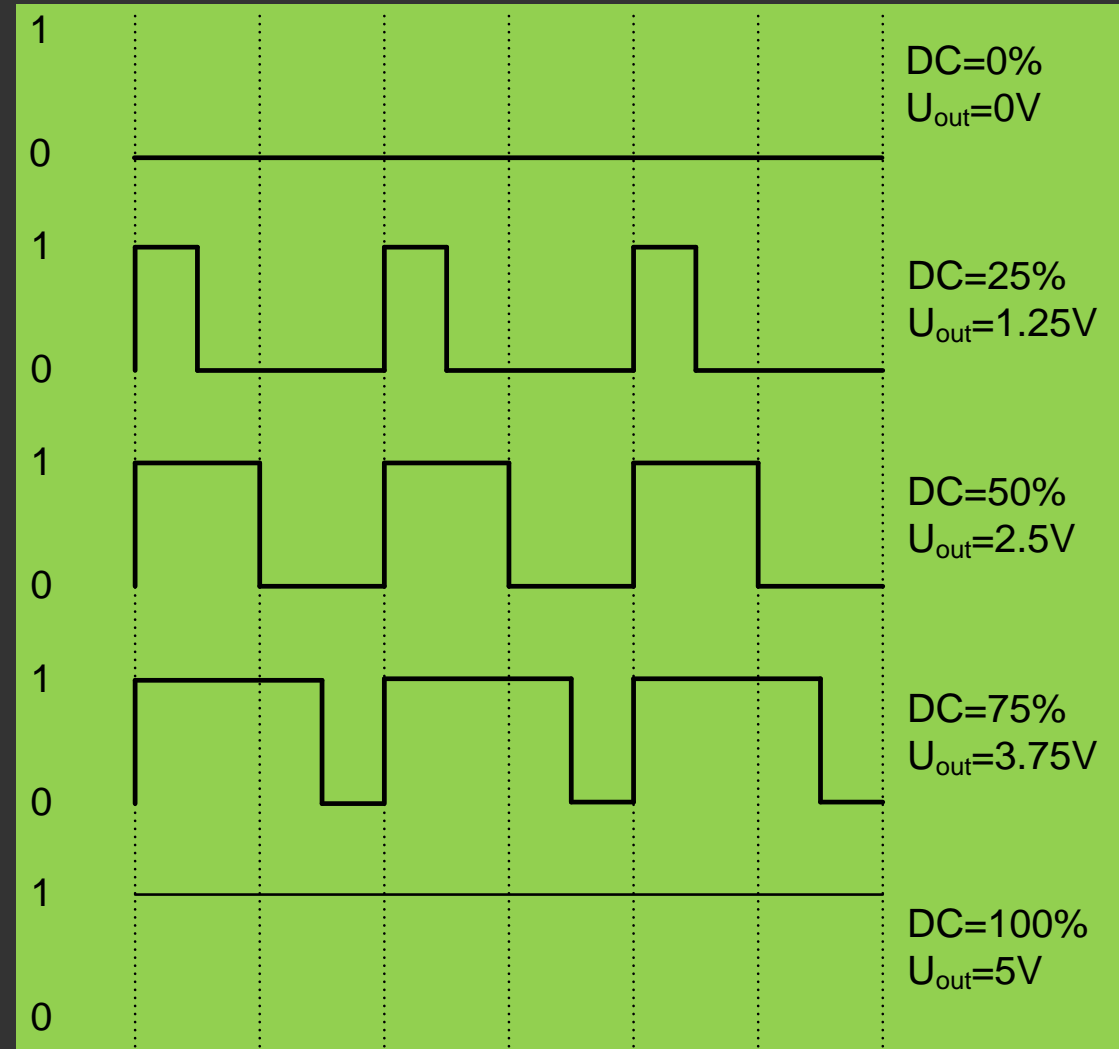
### Πειραματική διάταξη



# Μέτρηση φωτός

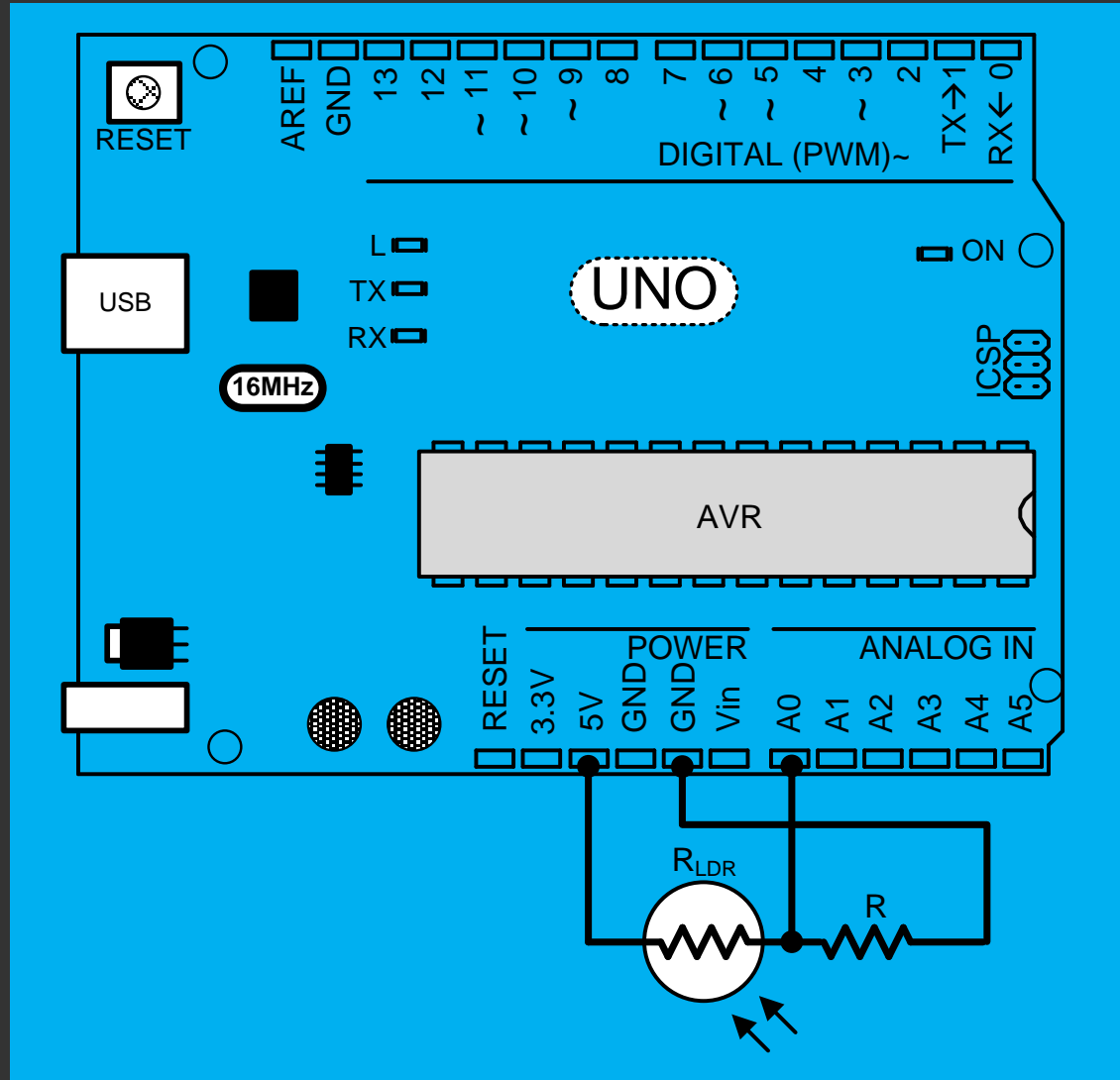
Μετρώντας τις συνθήκες φωτισμού (2)

Έλεγχος φωτεινότητας με PWM



# Μέτρηση φωτός

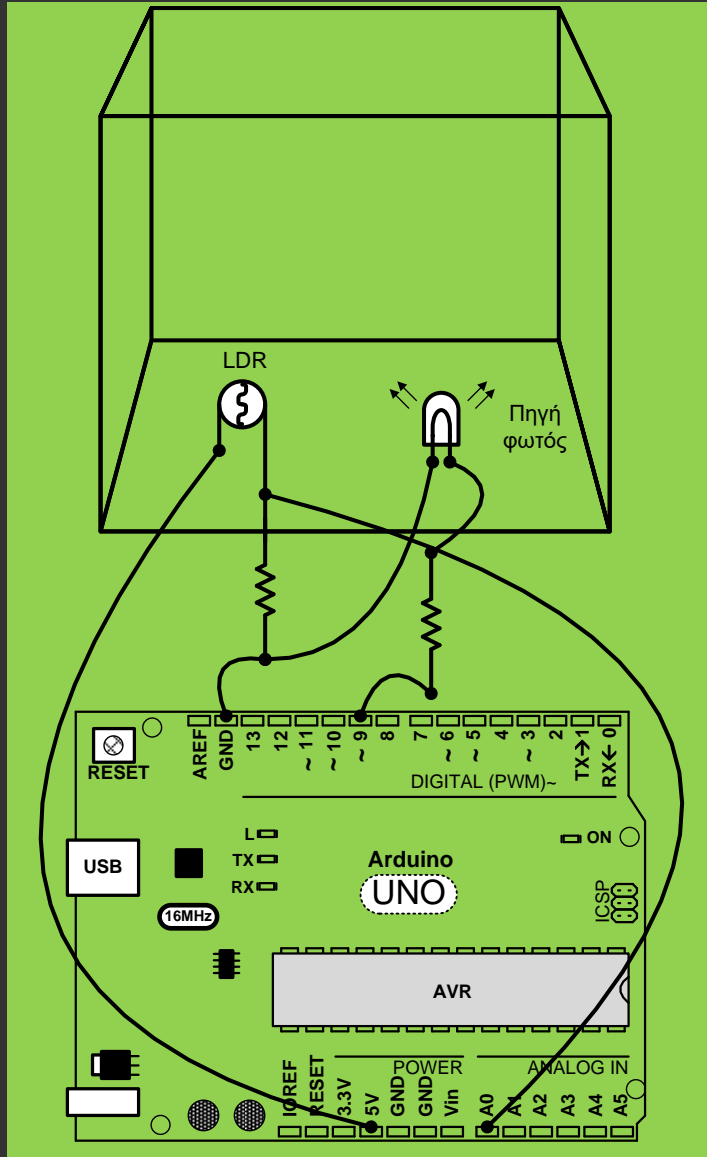
## Αρχικό κύκλωμα - Κώδικας



```
float Q=4.88;  
void setup()  
{  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  int a=analogRead(0);  
  float (a*Q)/1000;  
  Serial.println(Vin);  
  delay(1000);  
}
```

# Μέτρηση φωτός

## Πλήρες πείραμα - Κώδικας

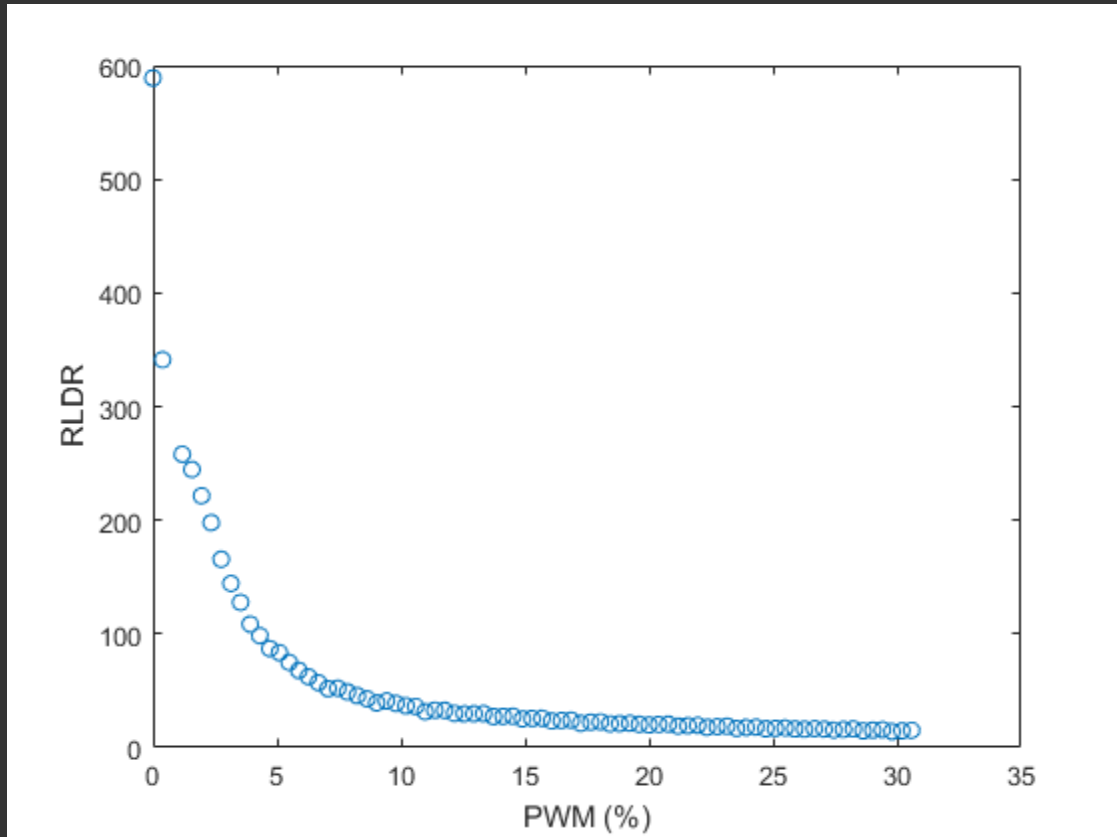


```
float Q=4.88;
int pwmpin=9;
void setup()
{
  Serial.begin(9600);
  pinMode(pwmpin,OUTPUT);
}
```

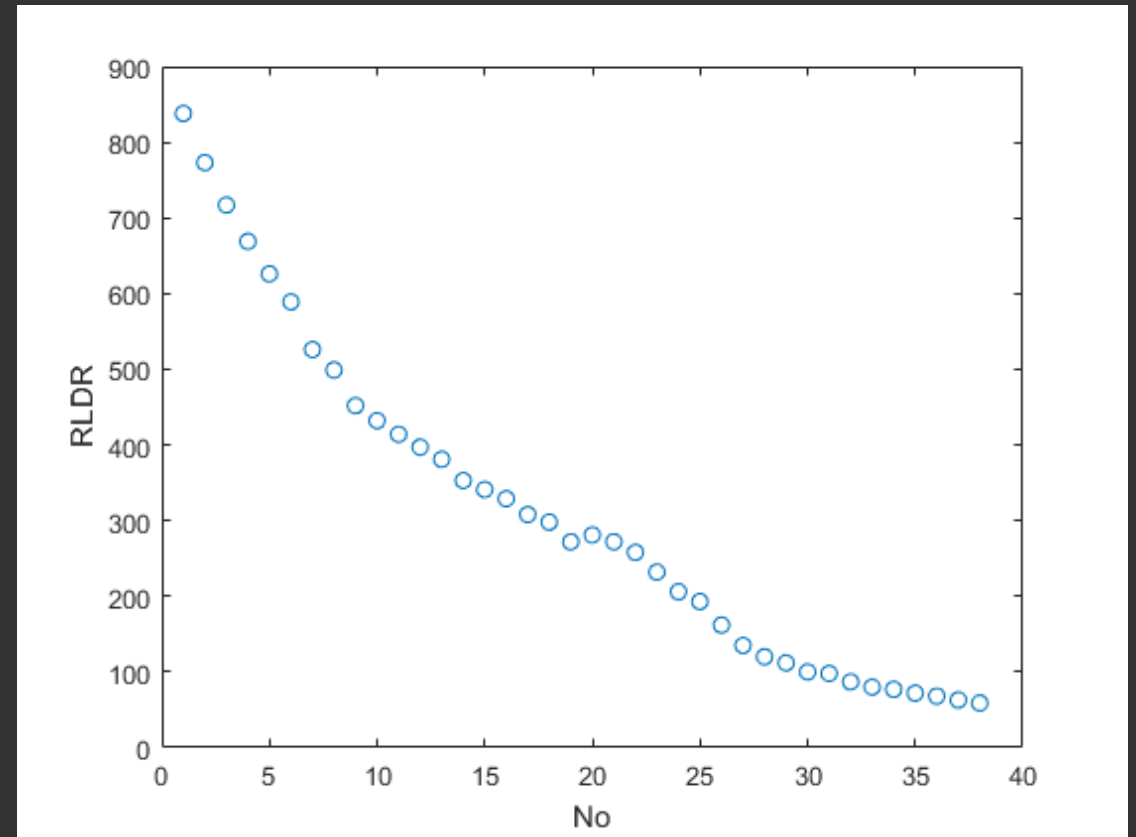
```
void loop()
{
  int i=0;
  while(i<=255)
  {
    analogWrite(pwmpin,i);
    int a=analogRead(0);
    float (a*Q)/1000;
    float I=Vin/9940;
    float ((5-Vin)/I)/1000;
    float pr=((float)i/255)*100;;
    Serial.print(""); Serial.print(pr);
    Serial.print("%,"); Serial.print(Vin);
    Serial.print(","); Serial.println(RLDR);
    i+=1;
  }
  analogWrite(pwmpin,0);
  delay(2000);
}
```

# Μέτρηση φωτός

## Ενδεικτικές μετρήσεις

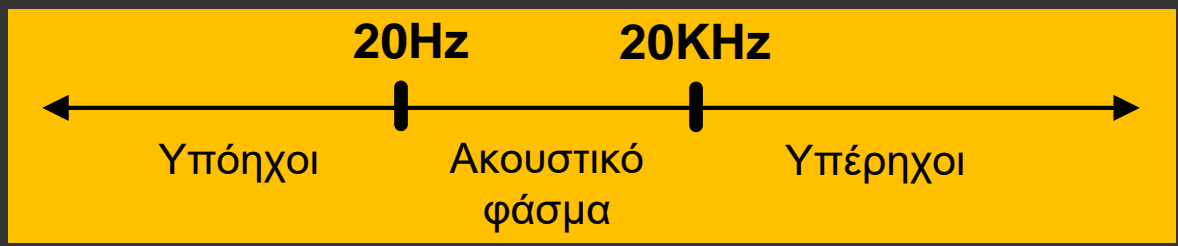


RLDR (KΩ) συναρτήσεϊ του PWM



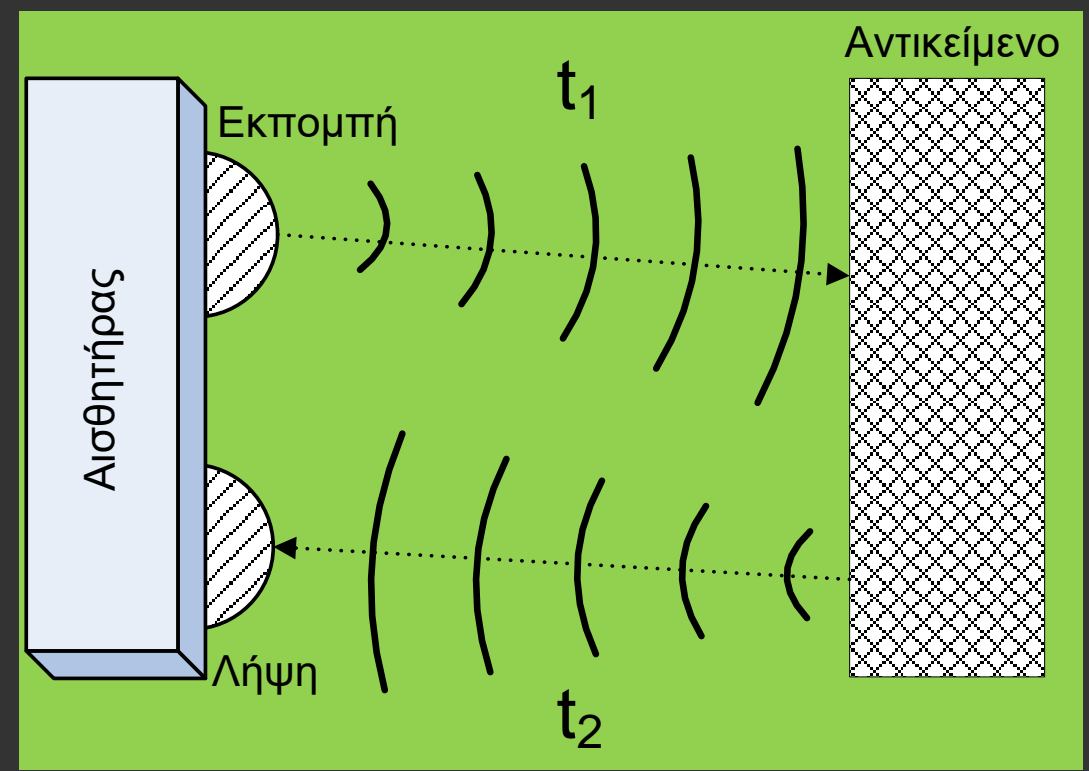
Μείωση της RLDR με φυσικό φως

# Μέτρηση απόστασης



Φάσμα ήχων

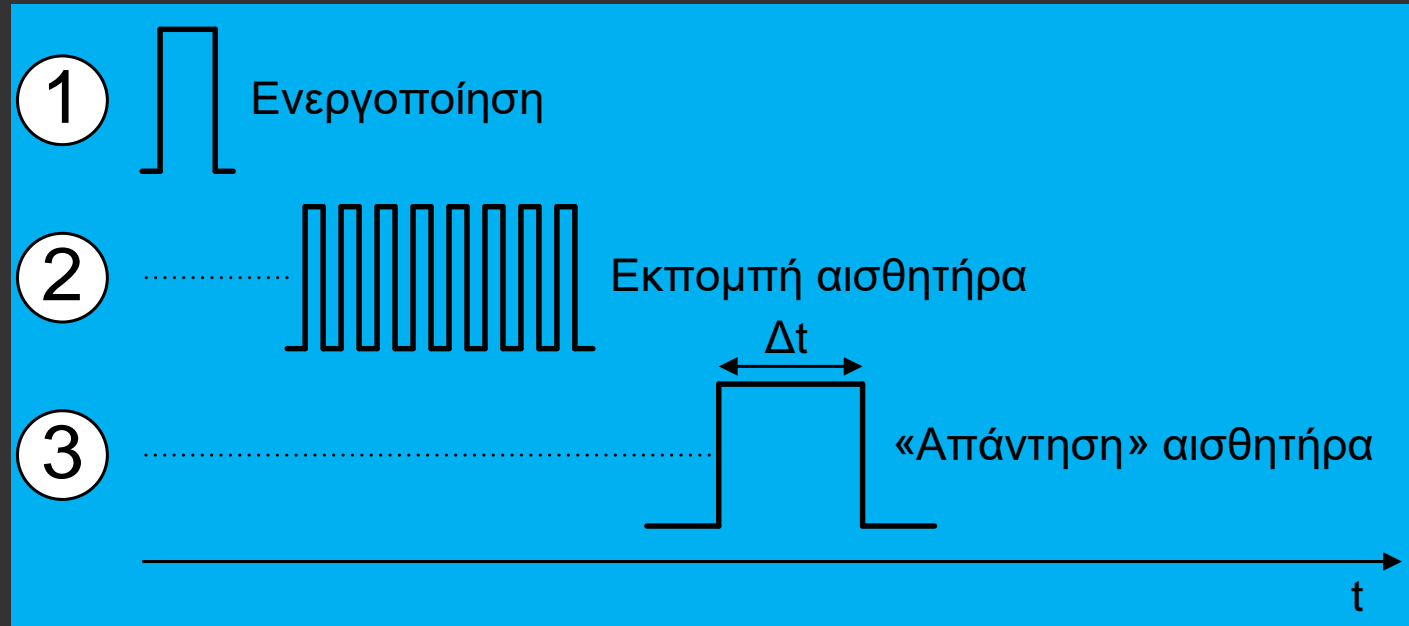
## Αισθητήρας HC-SR04



Σχηματική αναπαράσταση ανάκλασης ηχητικού σήματος

$$t = \frac{t_1 + t_2}{2}$$

# Μέτρηση απόστασης Αισθητήρας HC-SR04



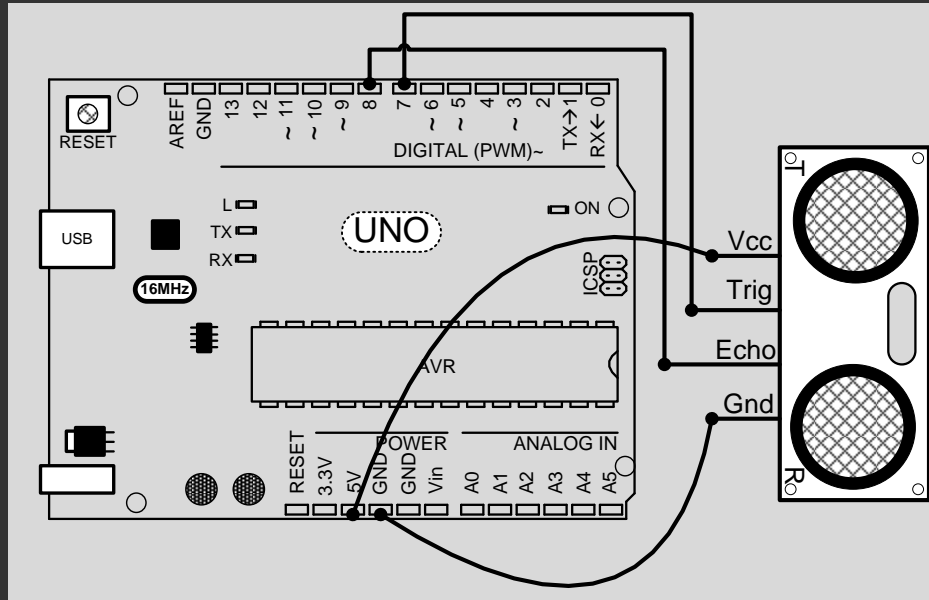
$$d = U_s \frac{\Delta t}{2}$$

$$U_s = U_{s0} \sqrt{1 + \frac{T(^{\circ}C)}{273.15}}$$



# Μέτρηση απόστασης

## Δοκιμή Αισθητήρα HC-SR04 (1)



Ακροδέκτης Arduino	Ακροδέκτης αισθητήρα	Περιγραφή
5V	Vcc	Θετικός ακροδέκτης τροφοδοσίας
GND	Gnd	Γείωση
7	Trig	Σήμα ενεργοποίησης
8	Echo	Έξοδος αισθητήρα

# Μέτρηση απόστασης

## Δοκιμή Αισθητήρα HC-SR04 (2)

```
int trig=7;
int echo=8;

void setup()
{
    pinMode(trig,OUTPUT);
    pinMode(echo,INPUT);
    Serial.begin(9600);
}
```

```
void loop()
{
    long dur;
    float cm;

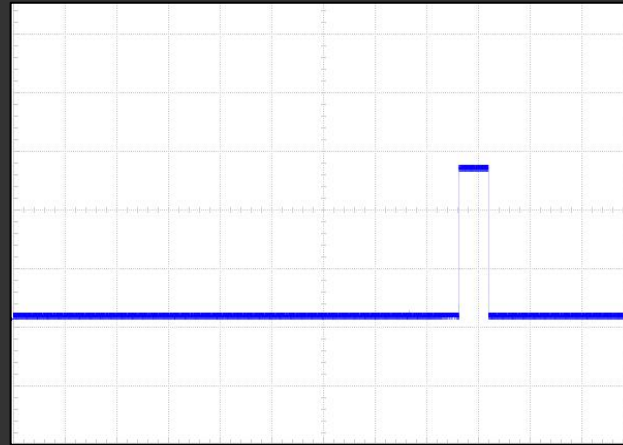
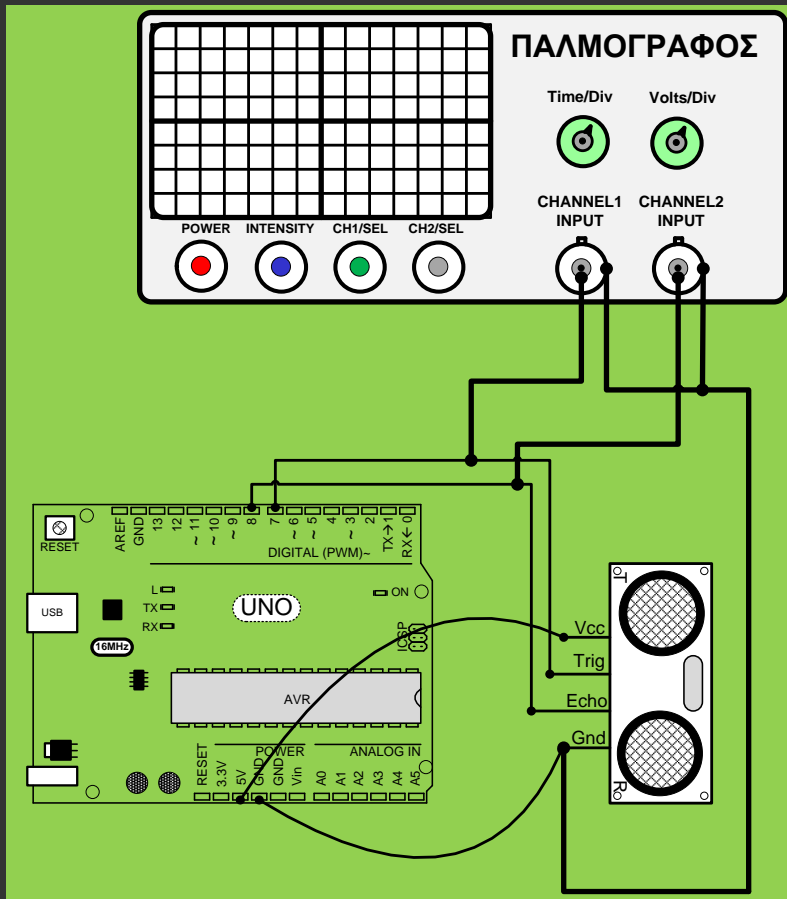
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);

    dur=pulseIn(echo,HIGH);
    cm=micro2cm(dur);
    Serial.print("microsec="); Serial.print(dur);
    Serial.print(",cm=");Serial.println(cm,2);
    delay(500);
}

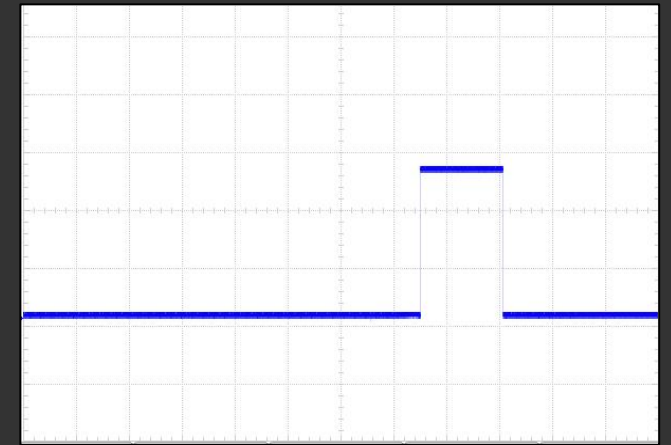
float micro2cm(long microsec)
{
    return (0.034*microsec/2);
}
```

# Μέτρηση απόστασης

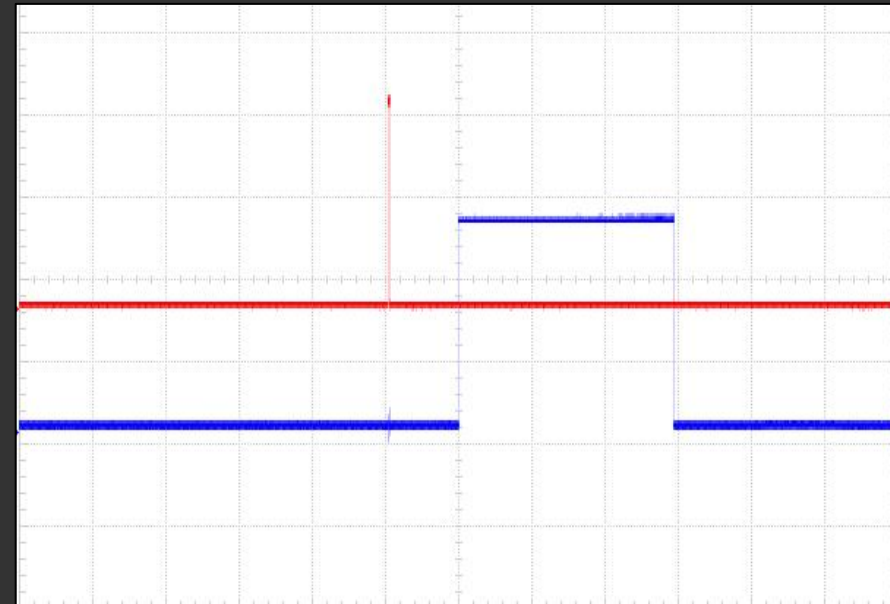
## Μετρήσεις με παλμογράφο



Απόσταση 4.93cm

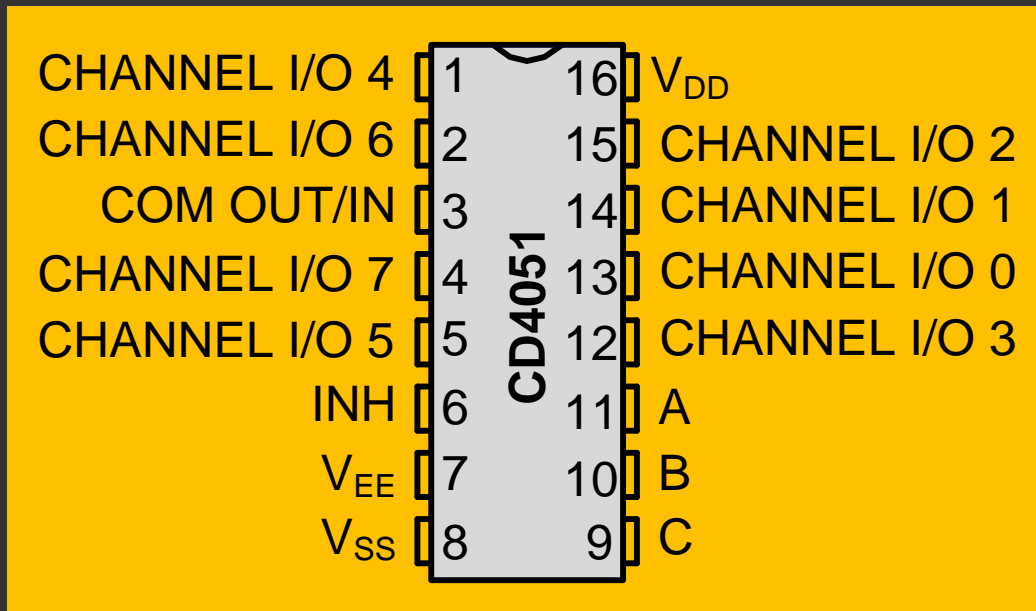


Απόσταση 13.2cm



Παλμός ενεργοποίησης  
και παλμός εξόδου αισθητήρα

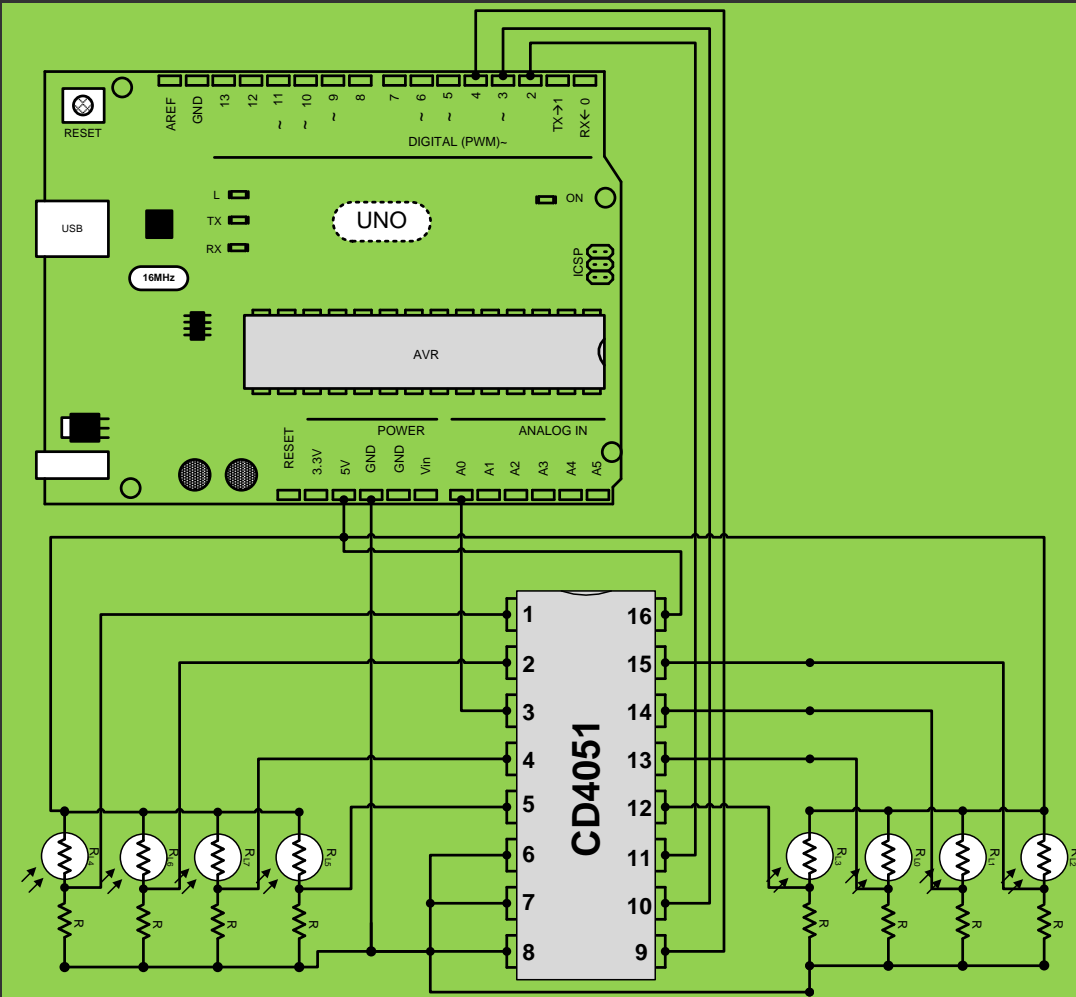
# Αναλογική πολυπλεξία (1)



ΕΙΣΟΔΟΙ				Ενεργό κανάλι
INH	C	B	A	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	Κανένα

# Αναλογική πολυπλεξία (2)

## Πειραματική διάταξη



Αισθητήρας	CD4051		Arduino
	Αρ. Ακροδέκτη	Ονομασία	
0	13	Channel I/O 0	
1	14	Channel I/O 1	
2	15	Channel I/O 2	
3	12	Channel I/O 3	
4	1	Channel I/O 4	
5	5	Channel I/O 5	
6	2	Channel I/O 6	
7	4	Channel I/O 7	
	6,7,8	INH, VEE, VSS	GND (γείωση)
	9	C	4
	10	B	3
	11	A	2
	3	COM OUT/IN	A0
	16	VDD	+5V

# Αναλογική πολυπλεξία (3)

## Κώδικας

```
byte pin[3]={4,3,2};

void setup()
{
  Serial.begin(9600);
  for(byte i=0;i<3;i++)
    pinMode(pin[i],OUTPUT);
}
```

```
void loop()
{
  byte bit3[3]={0,0,0};

  for(byte m=0;m<8;m++)
  {
    if (m>0)
    {
      if ((m%2)==0) bit3[1]=!bit3[1];
      if ((m%4)==0) bit3[0]=!bit3[0];
    }
  }

  for(byte p=0;p<3;p++)
    Serial.print(bit3[p]);

  Serial.println();

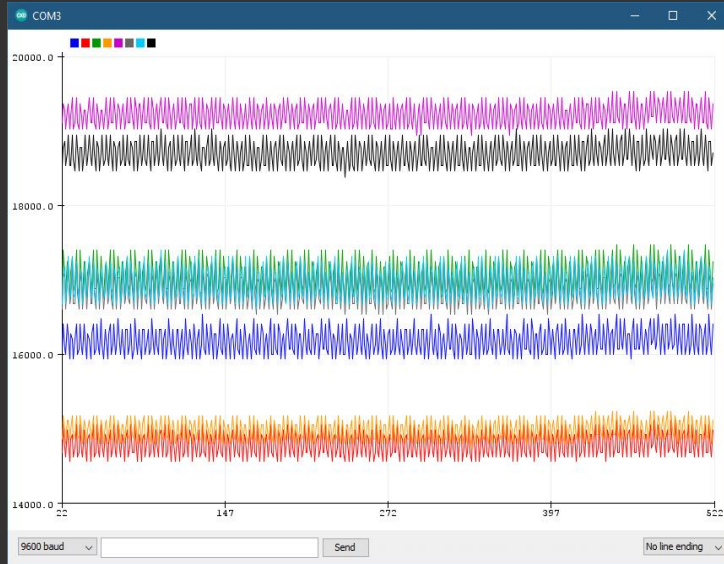
  for(byte i=0;i<3;i++)
    digitalWrite(pin[i],bit3[i]);
```

```
int analog=analogRead(0);
float V0=(analog*4.88)/1000;
float I=V0/10000;
float Ima=I*1000;
float VL=5-V0;
float RL=VL/I;

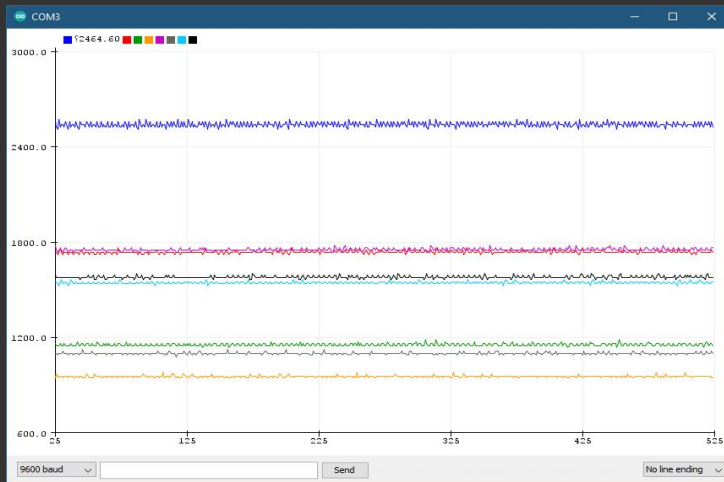
Serial.print("PhotoResistor #");
Serial.print(m); Serial.print("=");
Serial.print(RL); Serial.println(" Ohm");
delay(100);
bit3[2]=!bit3[2];
}
Serial.println("===");
delay(10000);
}
```

# Αναλογική πολυπλεξία (4)

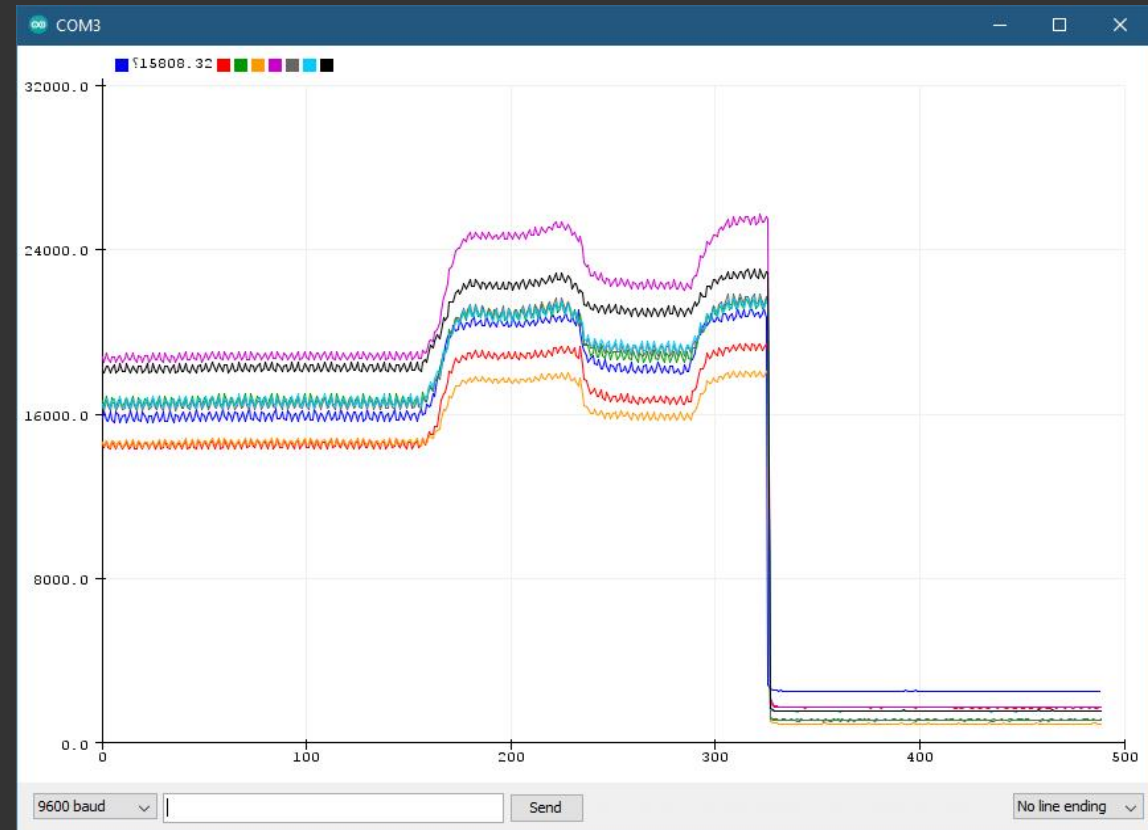
## Μετρήσεις από «ίδιους» αισθητήρες



Συνθήκες χαμηλού φυσικού φωτισμού

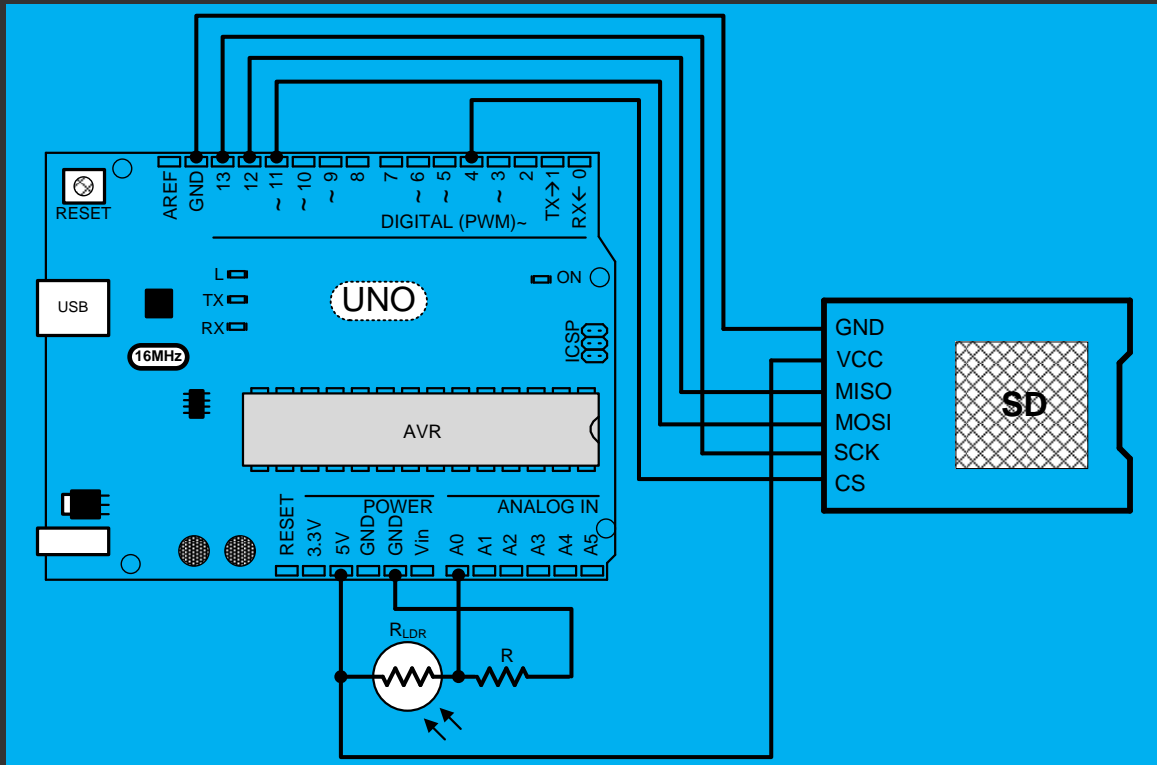
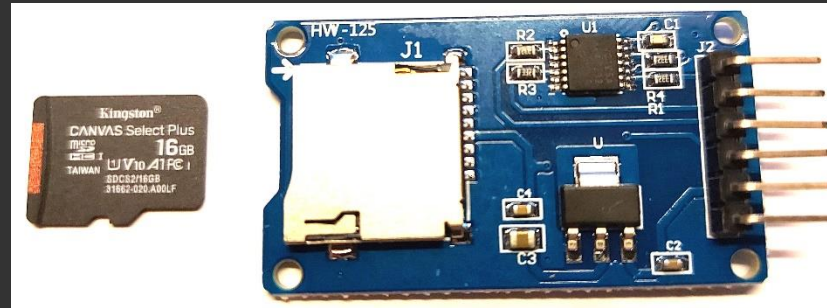


Συνθήκες ισχυρότερου φωτισμού (τεχνητού)



Μεταβολή συνθηκών φωτισμού

# Αποθήκευση μετρήσεων (1)



Ακροδέκτης Arduino	Ακροδέκτης CARD READER
+5V	+5V
GND	GND
4 (ψηφιακός)	CS
11 (ψηφιακός)	MOSI
12 (ψηφιακός)	MISO
13 (ψηφιακός)	SCK



# Αποθήκευση μετρήσεων (2) Κώδικας

```
#include <SD.h>
byte cs=4;
File datafile;

int counter=0;
void setup()
{
  Serial.begin(9600);
  delay(3000);
  Serial.println("Initializing SD ...");
  while (!SD.begin(cs))
    Serial.println("*** ERROR");
  Serial.println("SD
OK!");datafile=SD.open("csvfile.csv",
FILE_WRITE);
}
```

```
void loop()
{
  int analog=analogRead(0);
  float V0=(analog*4.88)/1000;
  float I=V0/10000;
  float Ima=I*1000;
  float VL=5-V0;
  float RL=VL/I;
  if (datafile)
  {
    Serial.print(counter); Serial.print("/ ");
    Serial.println(RL);
    datafile.println(RL);
  }
  counter++;
  if (counter==100)
  {
    datafile.close();
    Serial.println("SD file closed");
    while(1);
  }
  delay(1000);
}
```