



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΗΛΕΚΤΡΟΝΙΚΩΝ ΑΙΣΘΗΤΗΡΙΩΝ

Εισαγωγή στο Arduino

Μέρος Β: Λογισμικό





Arduino IDE

Arduino IDE

- Είναι το **open-source γραφικό περιβάλλον που επιτρέπει:**
 - Τη **συγγραφή του κώδικα**
 - Την **επικοινωνία με το Arduino**
- Είναι γραμμένο σε **Java** και βασίζεται στη γλώσσα **Processing**.
- Ο προγραμματισμός των Arduino γίνεται ουσιαστικά σε **C++**.
- Είναι συμβατό με **Windows, Linux** και **Mac OS X**.
- Πλέον υπάρχει και σε **Web έκδοση**.

The screenshot shows the Arduino IDE interface. The title bar reads "New_project | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations. The main area displays the code for a new project:

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

A message at the bottom left says "Done Saving." and the status bar at the bottom right indicates "Arduino/Genuino Uno on COM5".

Στοιχεία γραφικού περιβάλλοντος

Έλεγχος κώδικα

Φόρτωση στο Arduino

Δημιουργία/Άνοιγμα/
Αποθήκευση αρχείου

Το κάθε πρόγραμμα
γραμμένο για Arduino
ονομάζεται Sketch
(Σχέδιο) και έχει
κατάληξη .ino

```

Blink | Arduino 1.8.5
File Edit Sketch Tools Help
Sketch: /Blink
Blink
Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/



// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}

Done compiling.

Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.

```

Εμφάνιση
παραθύρου
σειριακής
επικοινωνίας
(Serial Monitor)

Περιοχή
συγγραφής κώδικα

Περιοχή ειδοποιήσεων/
πληροφοριών
σφαλμάτων

Επιλογή πλακέτας & Θύρας

Tools -> Board -> Arduino/Genuino Uno

```
File Edit Sketch Tools Help
demo
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

Done Saving.

1
Arduino Nano, ATmega328P (Old Bootloader) on COM8
```

Tools -> Port -> COM...

```
File Edit Sketch Tools Help
demo
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

Done Saving.

1
Arduino/Genuino Uno on COM8
```

Επιπρόσθετο βήμα για Linux (εντολές στo Terminal):

*sudo usermod -a -G dialout <username>
sudo chmod a+rw /dev/ttyACM0*

Βιβλιοθήκες

Συλλογές κώδικα και συναρτήσεων που δεν περιλαμβάνονται στη βασική γλώσσα του Arduino.

Εισαγωγή βιβλιοθήκης:

- Από τον **Library Manager**:
Arduino IDE -> Sketch -> Include Library -> Manage Libraries...
- Από αρχείο **.ZIP**:
Arduino IDE -> Sketch -> Include Library -> Add .ZIP Library...

Κλήση βιβλιοθήκης:

- Από το μενού:
Arduino IDE -> Sketch -> Include Library -> Επιλογή της βιβλιοθήκης
- Απευθείας:
`#include <Όνομα βιβλιοθήκης.h>`
π.χ. `#include <LiquidCrystal.h>` (χωρίς ερωτηματικό)



Στοιχεία της γλώσσας

Δομή κώδικα (1/2)

Βασικές συναρτήσεις:

void setup ():

Κώδικας που εκτελείται μόνο μια φορά,
κατά την εκκίνηση του προγράμματος.

void loop ():

Κώδικας που εκτελείται συνεχώς, όσο
τροφοδοτείται το Arduino.

- Το Arduino απενεργοποιείται μόνο με διακοπή της τροφοδοσίας του.
- Η εκτέλεση του κώδικα μπορεί να επανεκκινηθεί με το πλήκτρο Reset.
- Ο κώδικας παραμένει στη μνήμη flash του Arduino, ακόμη και μετά την απενεργοποίησή του.

The screenshot shows the Arduino IDE interface with a project titled "New_project". The code editor displays the following code:

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Blue curly braces and arrows point from the text descriptions of the two functions to their respective definitions in the code editor. The "setup()" function is associated with the brace above it, and the "loop()" function is associated with the brace below it.

Δομή κώδικα (2/2)

Σχόλια:

Περιοχή του κώδικα που αγνοείται κατά την εκτέλεση.

Ορίζονται με:

- `/* & */` (για πολλές γραμμές)
- `//` (για 1 γραμμή)

Ορισμός μεταβλητής:

- **Τύπος μεταβλητής**
- **Όνομα μεταβλητής**
- **Αρχική τιμή μεταβλητής**

Κλήση συναρτήσεων:

- **Όνομα συνάρτησης**
- **Παράμετροι συνάρτησης**

Προσοχή στο ";" στο τέλος των γραμμών!

```

/*
Blink

Turns an LED on for one second, then off for one second, repeatedly.

modified 8 May 2014 by Scott Fitzgerald
modified 2 Sep 2016 by Arturo Guadalupi
modified 8 Sep 2016 by Colby Newman

This example code is in the public domain
http://www.arduino.cc/en/Tutorial/Blink

*/
// Define LED pin as pin D13.
int led = 13;

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(led, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}

```

Βασικά είδη μεταβλητών

Όνομα	Τύπος	Μέγεθος (bits)	Τιμές
boolean	λογική σχέση	8	HIGH/LOW (5/0V)
char	χαρακτήρας	8	-128 έως 127 (ASCII)
byte	απόλυτος ακέραιος	8	0 έως 255
int	ακέραιος	16	-32,768 έως 32,767
word	απόλυτος ακέραιος	16	0 έως 65,535
long	ακέραιος	32	-2,147,483,648 έως 2,147,483,647
unsigned long	απόλυτος ακέραιος	32	0 έως 4,294,967,295
float	δεκαδικός	32	-3.4028235E38 έως 3.4028235E38

Διανύσματα (Arrays)

Ομάδα μεταβλητών που καλούνται βάσει ενός δείκτη θέσης.

Ορισμός διανύσματος:

Τύπος Όνομα[πλήθος στοιχείων] = {a, b, c, d, ...}

π.χ.

```
int myArray[3];  
int myVals[4] = {5, 8, 3, 2};
```

Εκχώρηση του i στοιχείου στη μεταβλητή x:

x = Όνομα διανύσματος[i]

π.χ.

```
x = myVals[2];
```

Η αρίθμηση των στοιχείων ενός διανύσματος αρχίζει από το 0.

Άρα η τιμή που θα λάβει το x στο παραπάνω παράδειγμα θα είναι 3.

Βασικές συναρτήσεις

Όνομα	Παράδειγμα	Λειτουργία
pinMode(pin, mode)	pinMode(13,OUTPUT)	Ορίζει τη λειτουργία ενός ψηφιακού pin σε Είσοδο ή Έξοδο
digitalRead(pin)	digitalRead(6)	Διαβάζει την τιμή ενός ψηφιακού pin (HIGH ή LOW)
digitalWrite(pin, value)	digitalWrite(4, HIGH)	Εκχωρεί τιμή σε ένα ψηφιακό pin (HIGH ή LOW)
analogRead(pin)	analogRead(A0)	Διαβάζει την τιμή ενός αναλογικού pin (0-1023)
analogWrite(pin, value)	analogWrite(9, 127)	Εκχωρεί τιμή σε ένα ψηφιακό PWM pin (0-255)
delay(ms)	delay(1000)	Παύει την εκτέλεση του προγράμματος για x milliseconds
millis() micros()	millis() micros()	Εμφανίζει το χρονικό διάστημα από την εκκίνηση της εκτέλεσης του προγράμματος, σε ms ή μs, αντίστοιχα
Serial.begin(bps)	Serial.begin(9600)	Ορίζει τον ρυθμό μετάδοσης (σε bits per second) των δεδομένων κατά τη σειριακή επικοινωνία
Serial.print(variable)	Serial.print(value)	Εμφανίζει την τιμή της μεταβλητής μέσω της σειριακής επικοινωνίας

Τελεστές

Συγκριτικοί τελεστές:

$x == y$	ίσο
$x != y$	διάφορο
$x < y$	μικρότερο
$x > y$	μεγαλύτερο
$x <= y$	μικρότερο ή ίσο
$x >= y$	μεγαλύτερο ή ίσο

Λογικοί τελεστές:

!	(όχι)
&&	(και)
	(ή)

if, else if, else

```
if (συνθήκη1)
{
    κώδικας...
}
else if (συνθήκη2)
{
    κώδικας...
}
else
{
    κώδικας...
}
```



```
if (voltage >= 8.5)
{
    Serial.print("Good");
}
else if (voltage >= 7.5 && voltage < 8.5)
{
    Serial.print("Low battery");
}
else
{
    Serial.print("Shutting down");
}
```

while, for

while (συνθήκη)

{

κώδικας...

}

π.χ.

```
int i = 0;  
while(i < 100)
```

{

```
i++;
```

}

for (αρχικοποίηση; συνθήκη; αύξηση/μείωση)

{

κώδικας...

}

π.χ.

```
for (int i=0; i <= 255; i++)
```

{

```
analogWrite(PWMpin, i);  
delay(10);
```

}

Interrupts

Διακόπτουν τη ροή του κώδικα, ώστε να εκτελεστεί κάποιο άλλο τμήμα του, όταν γίνει αντιληπτό συγκεκριμένο συμβάν.

attachInterrupt(αντιστοίχιση pin, συνάρτηση, τύπος);

- αντιστοίχιση pin: ένας αριθμός που αντιστοιχεί στο pin του Arduino που θα αντιληφθεί το συμβάν. Για το Arduino UNO: **0** για το **pin 2** και **1** για το **pin 3**.
- συνάρτηση: η συνάρτηση που θα κληθεί μετά το συμβάν.
- τύπος:
 - **RISING**: όταν η τιμή του pin αλλάξει από LOW σε HIGH
 - **FALLING**: όταν η τιμή του pin αλλάξει από HIGH σε LOW
 - **CHANGE**: όταν η τιμή του pin αλλάξει γενικά
 - **LOW**: όταν η τιμή του pin είναι LOW
- Η **attachInterrupt** εισάγεται στο **setup()**

π.χ.

attachInterrupt (0, Alarm, CHANGE);

Όταν αλλάζει η κατάσταση του pin 2, διακόπτεται η ροή του κώδικα και εκτελείται η συνάρτηση **Alarm()**.



Παραδείγματα

Παράδειγμα I: Blink

Arduino IDE -> File -> Examples -> 01.Basics -> Blink

Πρόγραμμα που ανάβει και σθήνει το ενσωματωμένο LED του Arduino, ανά 1 δευτερόλεπτο

Ορισμός του pin που αντιστοιχεί στο ενσωματωμένο LED (pin D13), ως pin εξόδου

Ενεργοποίηση του pin D13

Παραμονή στην ίδια κατάσταση για 1s (1000ms)

Απενεργοποίηση του pin D13

Παραμονή στην ίδια κατάσταση για 1s

```
/*
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Colby Newman
 */

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink

*/
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
}
```

Σχόλια, περιγραφή της λειτουργίας κ.λπ.

setup(): Εκτέλεση μόνο κατά την έναρξη

loop(): Επανάληψη του βρόχου συνεχώς

Παράδειγμα II: PhysicalPixel

Arduino IDE -> File -> Examples -> 04.Communication -> PhysicalPixel

Πρόγραμμα που λαμβάνει ως είσοδο εντολή από
υπολογιστή, για να ανάψει ένα LED

Ορισμός μιας σταθεράς που αντιστοιχεί στο LED (pin D13)

const: Ορισμός σταθεράς (η τιμή που εκχωρείται
δεν μπορεί να αλλάξει μέσα στο πρόγραμμα)

Ορισμός μιας μεταβλητής τύπου int

Έναρξη σειριακής επικοινωνίας στα 9600bps

Ορισμός του pin D13 ως pin εξόδου

Εάν υπάρχουν δεδομένα από τη σειριακή επικοινωνία...

...να εκχωρηθεί η τιμή στη μεταβλητή incomingByte και...

...εάν η τιμή είναι H, να ανάψει το LED

Αλλιώς, εάν η τιμή είναι L, να σβήσει το LED

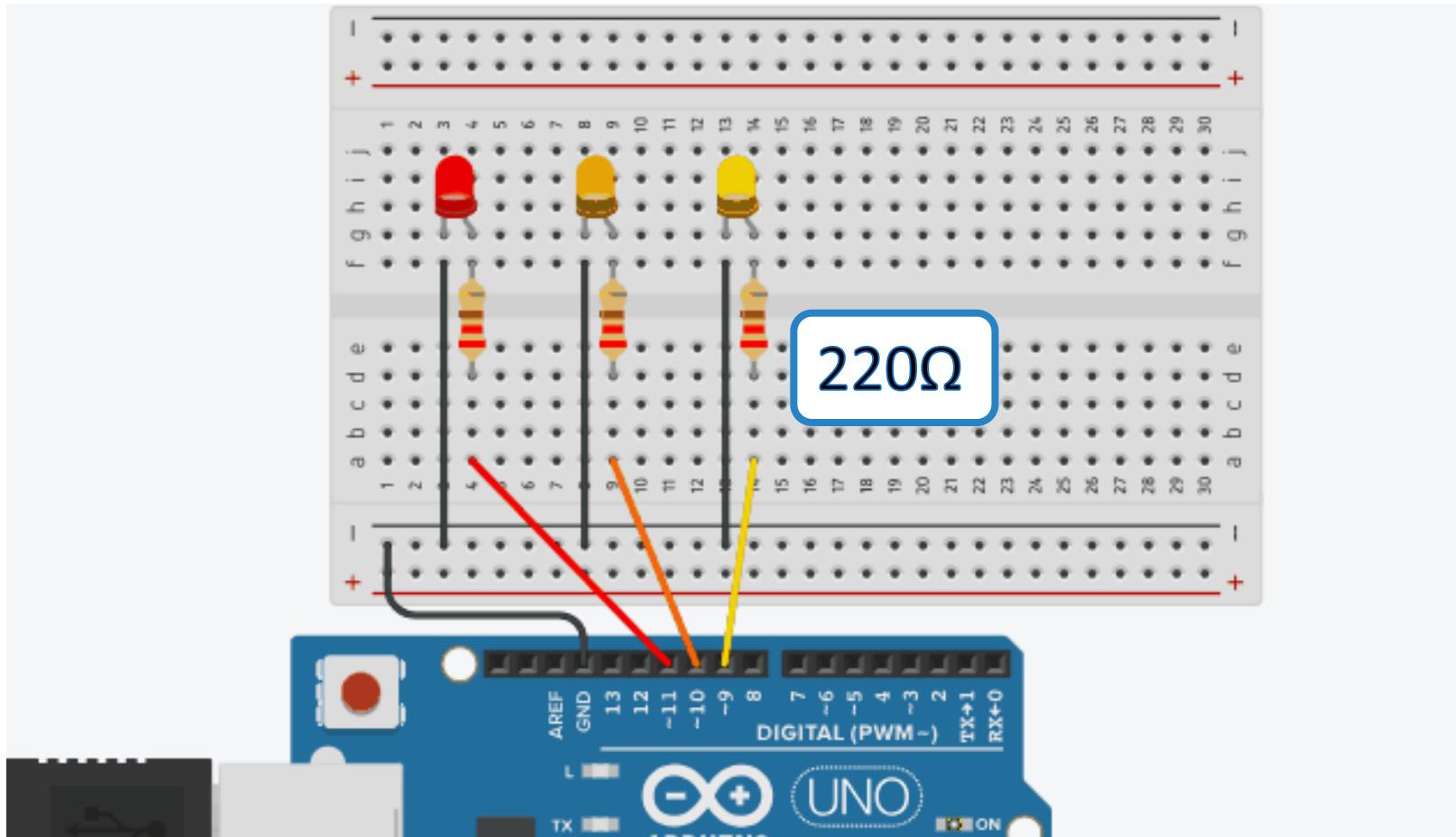
```
const int ledPin = 13;
int incomingByte;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

Παράδειγμα III: LED Fire (1/2)

Πρόγραμμα που προσομοιώνει το φως της φωτιάς, χρησιμοποιώντας LEDs



Παράδειγμα III: LED Fire (2/2)

Ορισμός σταθερών που αντιστοιχούν στα pins τριών LEDs (προσοχή: πρέπει να είναι PWM pins)

Ορισμός των pins ως pins εξόδου

randomSeed(πηγή): Ορισμός πηγής τυχαίων αριθμών (π.χ. από μία αναλογική είσοδο).
random(min, max): Παραγωγή τυχαίας τιμής στο εύρος (min, max).

Λήψη τυχαίων αριθμών βάσει της εισόδου A0

Εκχώρηση τυχαίου duty cycle (δηλ. τυχαίας φωτεινότητας) σε κάθε LED

Ορισμός τυχαίας τιμής καθυστέρησης

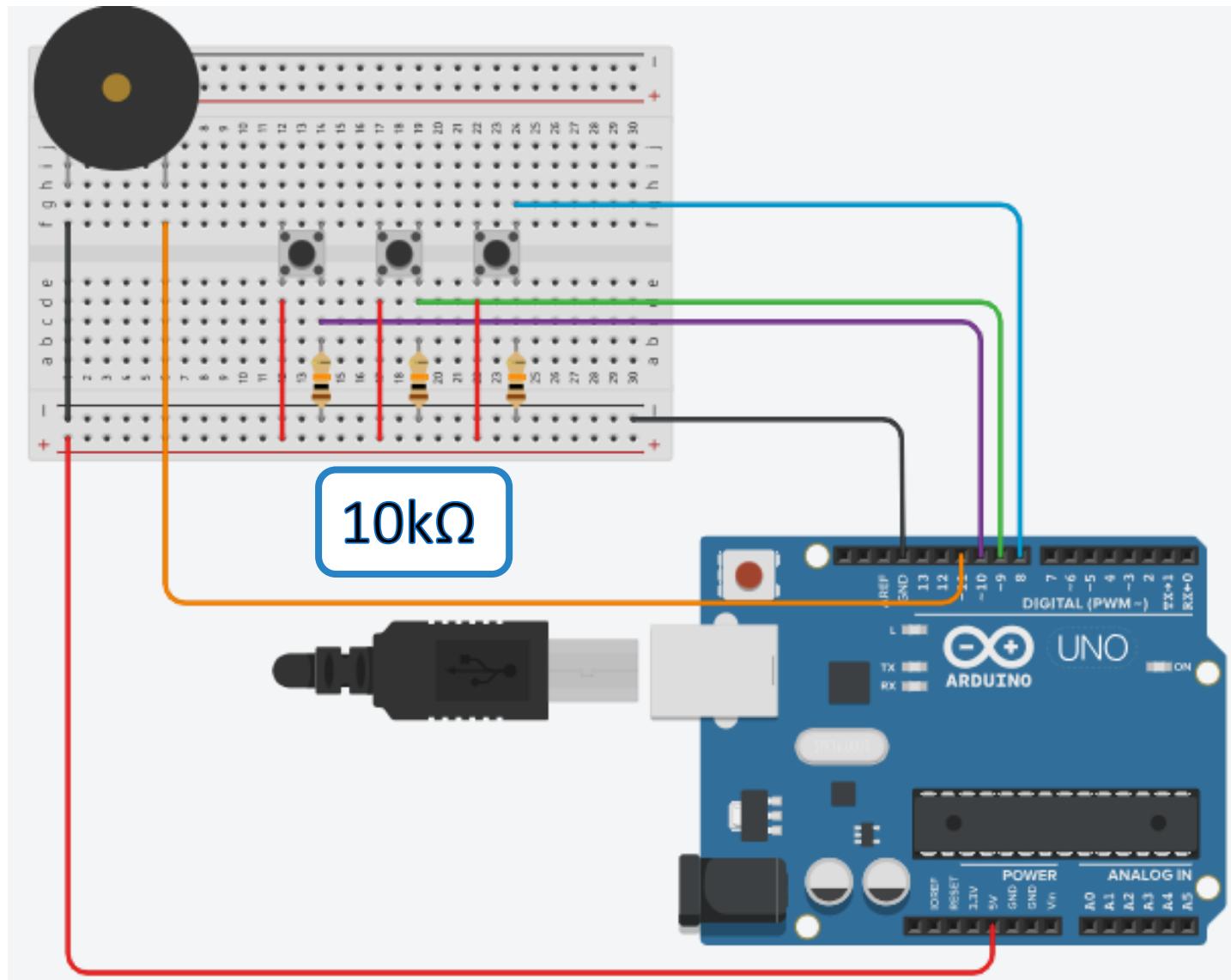
```
int ledPin1 = 9;
int ledPin2 = 10;
int ledPin3 = 11;

void setup()
{
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
    randomSeed(analogRead(A0));
}

void loop()
{
    analogWrite(ledPin1, random(0, 130) + 125);
    analogWrite(ledPin2, random(0, 130) + 125);
    analogWrite(ledPin3, random(0, 130) + 125);
    delay(random(0, 120));
}
```

Παράδειγμα IV: Piano (1/2)

Πρόγραμμα που παίζει 3 νότες με το πάτημα αντίστοιχων πλήκτρων



Παράδειγμα IV: Piano (2/2)

```
int speaker_pin = 11;
int Re_pin = 10;
int Fa_pin = 9;
int Sol_pin = 8;
```

} Ορισμός των pins

```
int notes[] = {294, 349, 392};

void setup() {
    pinMode(Re_pin, INPUT);
    pinMode(Fa_pin, INPUT);
    pinMode(Sol_pin, INPUT);
    pinMode(speaker_pin, OUTPUT);
}
```

Ορισμός διανύσματος
με στοιχεία 3 νότες
(συχνότητες)

} Ορισμός των pins ως
εξόδου και εισόδου

Όσο είναι πατημένο το αντίστοιχο κουμπί...

...να αναπαράγεται η αντίστοιχη νότα από το ηχείο,
για 200ms

```
void loop() {
    while (digitalRead(Re_pin) == HIGH)
    {
        tone(speaker_pin, notes[0], 200);
    }
    while (digitalRead(Fa_pin) == HIGH)
    {
        tone(speaker_pin, notes[1], 200);
    }
    while (digitalRead(Sol_pin) == HIGH)
    {
        tone(speaker_pin, notes[2], 200);
    }
}
```

tone(pin, συχνότητα, διάρκεια): Συνάρτηση για αναπαραγωγή ήχου

Βαθμονόμηση αισθητήρα (1/2)

Εάν γνωρίζουμε τη σχέση τάσης εξόδου και μετρούμενου μεγέθους:

- π.χ. Στο datasheet του αισθητήρα θερμοκρασίας **TMP36** αναγράφεται ότι η σχέση τάσης εξόδου και θερμοκρασίας είναι **10mV/°C**, με ένα offset **500mV**.
- Η ανάλυση του Analog-to-Digital Converter (ADC) του Arduino UNO είναι **10-bit** (0-1023).
- Επομένως, η μετρούμενη τάση σε **mV** θα είναι:
 - Για **V_{cc}=5V**: $V_{out} = \text{τιμή_εισόδου} \cdot \frac{5000}{1024}$
 - Για **V_{cc}=3.3V**: $V_{out} = \text{τιμή_εισόδου} \cdot \frac{3300}{1024}$
- Τελικά, βάσει των παραπάνω σχέσεων, η θερμοκρασία σε **°C** για τον συγκεκριμένο αισθητήρα θα είναι:

$$Temp_C = \frac{V_{out} - 500}{10}$$

Βαθμονόμηση αισθητήρα (2/2)

Εάν γνωρίζουμε τη σχέση εύρους μέτρησης και εύρους τάσης:

- π.χ. Στο datasheet του αισθητήρα Hall SS49 παρουσιάζεται διάγραμμα, βάσει του οποίου τα 1000 Gauss αντιστοιχούν στα 4V και τα -1000 Gauss στο 1V.
- Χρησιμοποιούμε τη συνάρτηση **map()**:

map(μεταβλητή, αρχ. min, αρχ. max, τελ. min, τελ. max):
Συνάρτηση για αντιστοίχιση τιμών από ένα εύρος σε άλλο

- Όπως και προηγουμένως, λόγω της ανάλυσης των 10-bit, έχουμε:

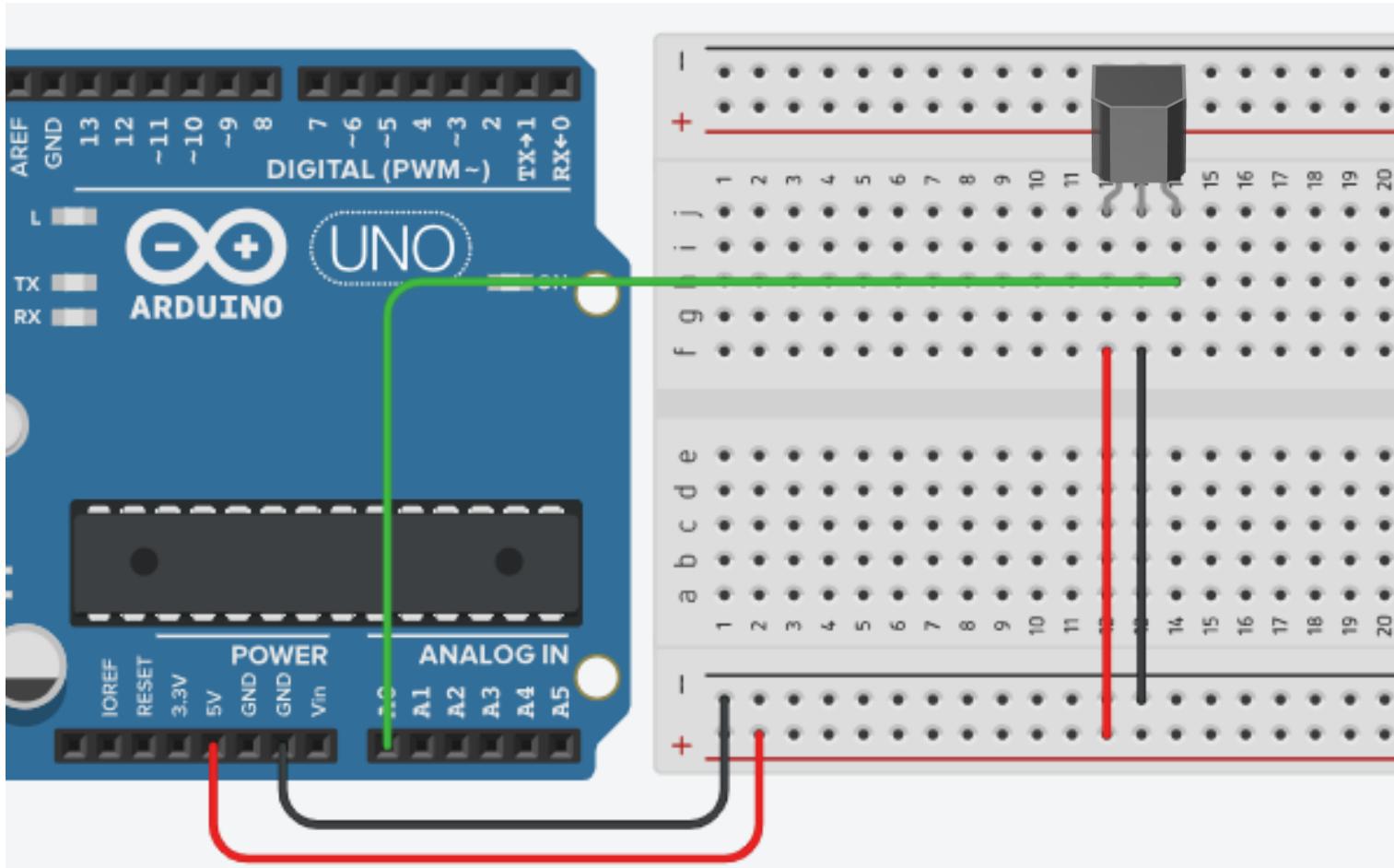
$$1V \rightarrow 1024/5 \approx 205 \quad \& \quad 4V \rightarrow (1024/5) \cdot 4 \approx 819$$

Άρα, ορίζουμε μία μεταβλητή (έστω **field**), που θα λάβει τιμές βάσει της συνάρτησης **map()**:

map(field, 205, 819, -1000, 1000)

Παράδειγμα V: Hall SS49 (1/2)

Πρόγραμμα που διαβάζει την έξοδο ενός αισθητήρα Hall SS49



Παράδειγμα V: Hall SS49 (2/2)

```

int pin = A0;
float vcc = 5000;
float cal = 0.014;
float offset = 521;

int value = 0;
float voltage = 0.0;
float field = 0.0;

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

```

Έναρξη σειριακής
επικοινωνίας στα
9600bps

Ορισμός pin εισόδου,
τάσης τροφοδοσίας,
τιμής βαθμονόμησης
& τιμής απόκλισης

Ορισμός και αρχικοποίηση
μεταβλητών

Υπολογισμός τιμής τάσης & πεδίου

Εμφάνιση αποτελεσμάτων

Λήψη νέας μέτρησης κάθε 1 s

```

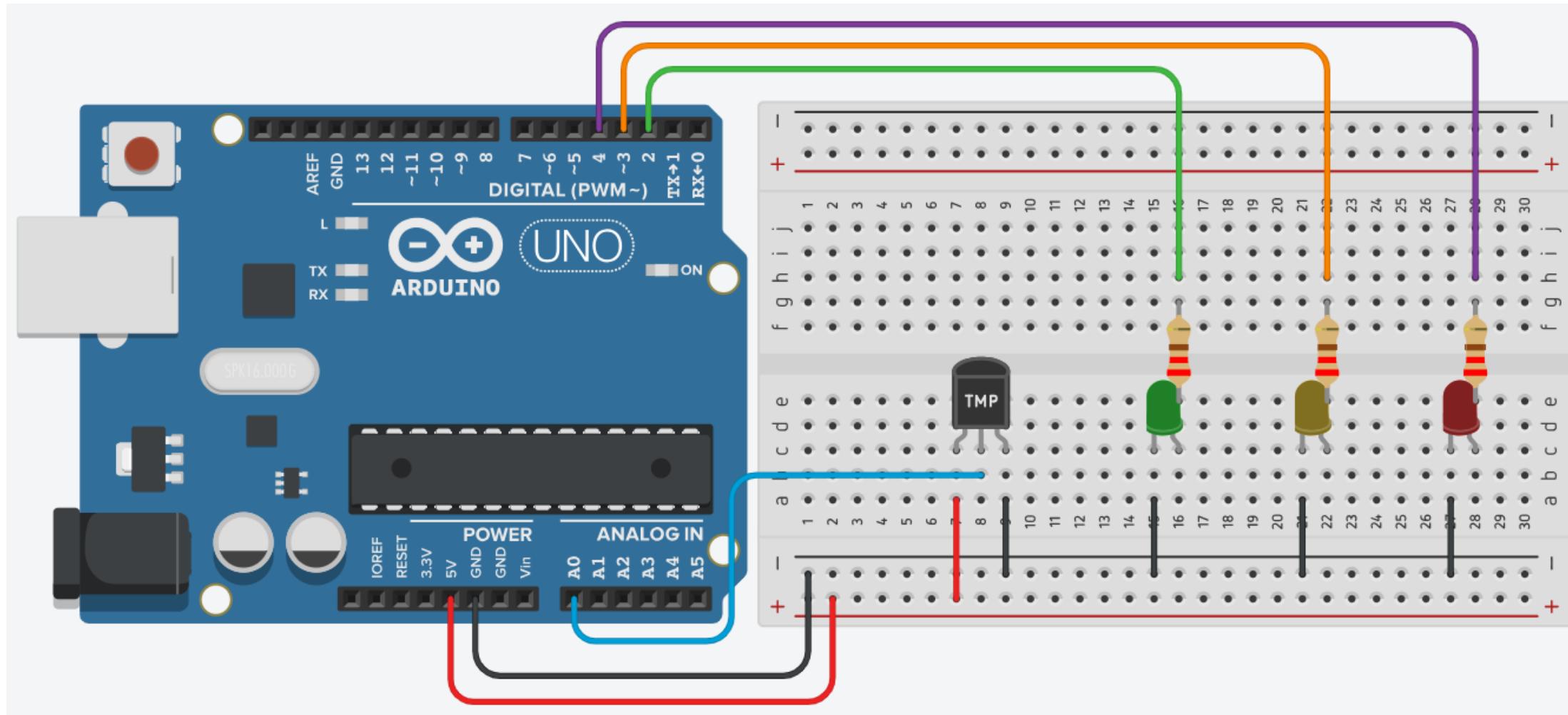
void loop() {
    value = analogRead(pin) - offset; //A
    voltage = value * (vcc / 1024.0); //M
    field = (voltage) / cal;

    //Εμφάνιση αποτελεσμάτων
    Serial.print("Value: ");
    Serial.print(value);
    Serial.print("\t");
    Serial.print(voltage);
    Serial.println(" mV");
    Serial.print("Magnetic Field: ");
    Serial.print(field);
    Serial.println(" uT");
    Serial.println("=====");
    //Λήψη νέας τιμής κάθε 1sec
    delay(1000);
}

```

Παράδειγμα VI: TMP36 (1/2)

Πρόγραμμα που διαβάζει την έξοδο ενός αισθητήρα θερμοκρασίας TMP36 και ανέβει αναλόγως ένα LED



Παράδειγμα VI: TMP36 (2/2)

```

int green_pin = 2;
int orange_pin = 3;
int red_pin = 4;
int tempPin = A0;

void setup()
{
    pinMode(green_pin, OUTPUT);
    pinMode(orange_pin, OUTPUT);
    pinMode(red_pin, OUTPUT);
    Serial.begin(9600);
}

```

} Ορισμός των pins

Ανάγνωση, υπολογισμός και εμφάνιση
ακέραιας τιμής θερμοκρασίας

} Ορισμός των pins ως
εξόδου και εισόδου

Ενεργοποίηση αντίστοιχου LED,
αναλόγως της θερμοκρασίας

Έναρξη σειριακής
επικοινωνίας στα
9600bps

Λήψη νέας μέτρησης κάθε 1 s

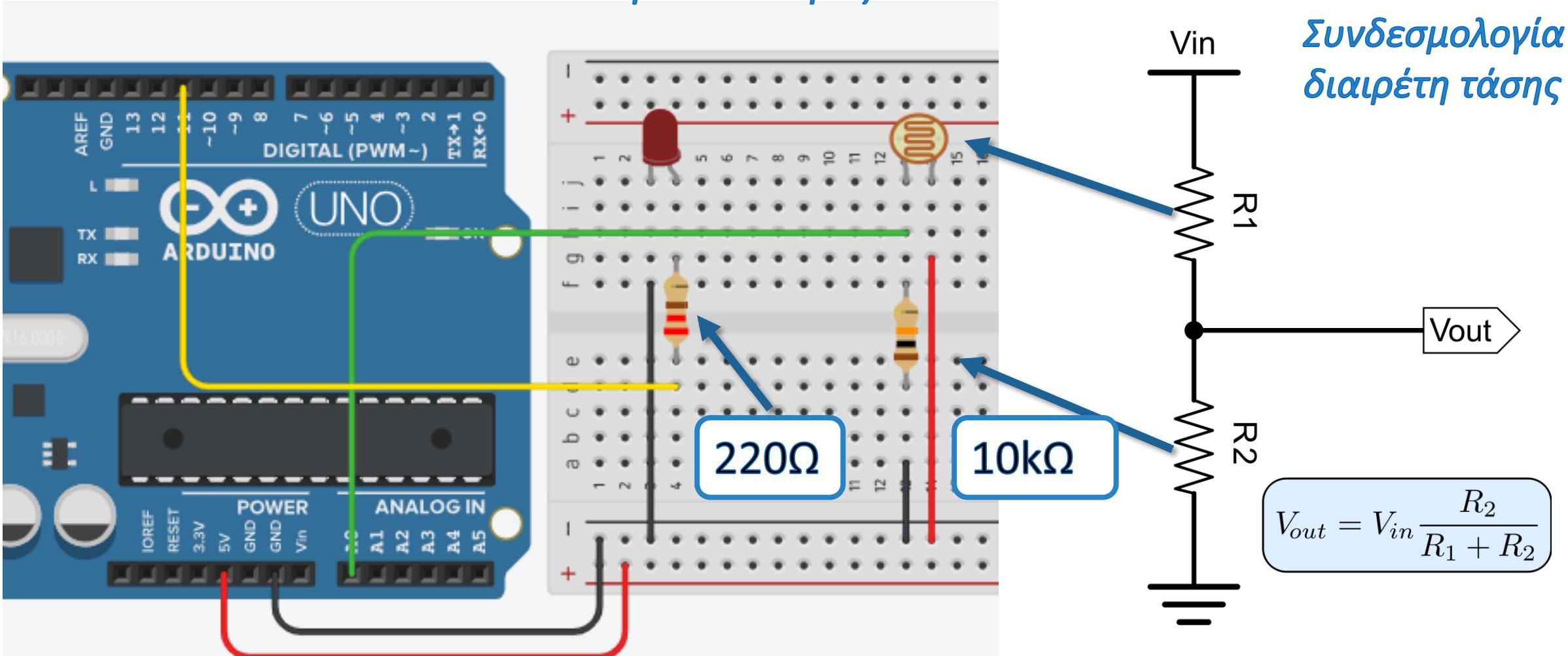
```

void loop()
{
    int tempValue = analogRead(tempPin);
    float temp = tempValue * 5000.0 / 1024.0;
    float temp_C = (temp - 500) / 10;
    Serial.println(tempValue);
    Serial.print("Temp = ");
    Serial.print(temp_C, 0);
    Serial.println("C");
    if (temp_C <= 25) {
        digitalWrite(green_pin, HIGH);
        digitalWrite(orange_pin, LOW);
        digitalWrite(red_pin, LOW);
    }
    else if (temp_C > 25 && temp_C <= 35) {
        digitalWrite(green_pin, LOW);
        digitalWrite(orange_pin, HIGH);
        digitalWrite(red_pin, LOW);
    }
    else {
        digitalWrite(green_pin, LOW);
        digitalWrite(orange_pin, LOW);
        digitalWrite(red_pin, HIGH);
    }
    delay(1000);
}

```

Παράδειγμα VII: Αισθητήρας φωτεινότητας (1/3)

Πρόγραμμα που διαβάζει ψηφιακά και αναλογικά την έξοδο ενός αισθητήρα IR και συγχεί αναλόγως ένα LED



Παράδειγμα VII: Αισθητήρας φωτεινότητας (2/3)

```
int incomingByte;
int value;
}

void setup() {
    pinMode(11, OUTPUT);
    Serial.begin(9600);
    Serial.println("Please enter A for Analog or D for
}
```

Ορισμός του pin11 ως
pin εξόδου, έναρξη
σειριακής επικοινωνίας
στα 9600bps και
εμφάνιση μηνύματος για
επιλογή λειτουργίας

Ορισμός
μεταβλητών

Εάν υπάρχουν δεδομένα από τη
σειριακή επικοινωνία...

Please enter A for Analog or D for

Εάν αποσταλεί A, να εκτελείται η
analog() μέχρι να αποσταλούν
ξανά δεδομένα

Εάν αποσταλεί D, να εκτελείται η
digital() μέχρι να αποσταλούν
ξανά δεδομένα

Άλλιώς, να εμφανιστεί μήνυμα
σφάλματος

```
void loop() {
    if (Serial.available() > 0) {
        incomingByte = Serial.read();
        if (incomingByte == 'A') {
            while (Serial.available() == 0) {
                analog();
            }
        } else if (incomingByte == 'D') {
            while (Serial.available() == 0) {
                digital();
            }
        } else {
            Serial.println("Unknown command");
        }
    }
}
```

Παράδειγμα VII: Αισθητήρας φωτεινότητας (3/3)

Ανάγνωση και εμφάνιση τιμής αισθητήρα IR

Μετατροπή της αναλογικής τιμής στο εύρος (0, 255) και εκχώρηση της τιμής ως duty cycle του LED

Ανάγνωση και εμφάνιση τιμής αισθητήρα IR

Μετατροπή της αναλογικής τιμής στις τιμές LOW/ HIGH και εκχώρηση της τιμής ως κατάσταση του LED



```
void analog() {
    value = analogRead(A0);
    Serial.print(value);
    Serial.print("\t");
    int analogValue = map(value, 512, 1023, 0, 255);
    analogWrite(11, analogValue);
    Serial.println(analogValue);
}
```

```
void digital() {
    value = analogRead(A0);
    Serial.print(value);
    Serial.print("\t");
    boolean digitalValue = map(value, 0, 600, LOW, HIGH);
    digitalWrite(11, digitalValue);
    Serial.println(digitalValue);
}
```

Προσοχή στις ρυθμίσεις του Serial Monitor:

No line ending

9600 baud

Περισσότερα...

Πληροφορίες για τη γλώσσα προγραμματισμού του Arduino:
<https://www.arduino.cc/reference/en/>

Λήψη του Arduino IDE:
<https://www.arduino.cc/en/Main/Software>

Κοινότητα του Arduino:
<https://playground.arduino.cc/>