

CAN  
SVT  
IN GREECE



SPIN  
SPACE INNOVATION

# Arduino Teachers Workshop

## Εισαγωγή στο Arduino

- Προγραμματισμός στο Arduino IDE
- Επικοινωνία με τον υπολογιστή μέσω USB
- Πλακέτα εύκολων συνδέσεων Breadboard

## Projects

- Led Blink
- Αναλογική ανάγνωση Ποτενσιόμετρου
- Έλεγχος φωτεινότητας Led μέσω Ποτενσιόμετρου με χρήση PWM
- Αυτόματη λάμπα με χρήση φωτοκύτταρου
- Έλεγχος κινητήρα Servo
- Ρομποτικός βραχίονας

# Τύποι Arduino



Arduino Uno



Arduino Leonardo



Arduino Mega ADK



Arduino Ethernet



LilyPad Arduino SimpleSnap



LilyPad Arduino



Arduino Due



Arduino Yún



Arduino Mega 2560



Arduino Mini



Arduino Nano



Arduino Pro Mini



Arduino Tre



Arduino Micro



LilyPad Arduino USB



LilyPad Arduino Simple

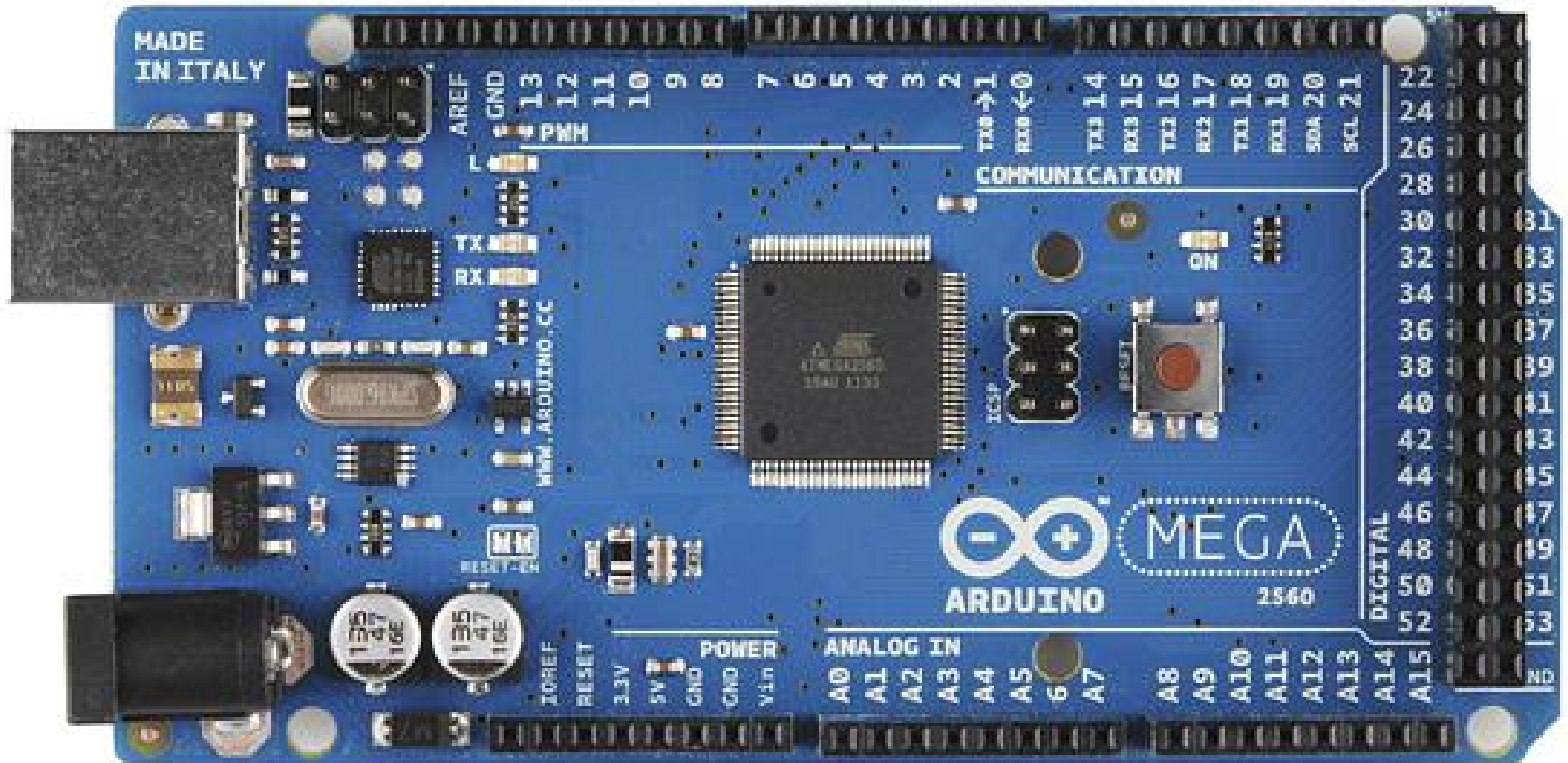


Arduino Pro



Arduino Fio

# Arduino MEGA 2560



Για περισσότερες πληροφορίες δείτε το: [arduino.cc](http://arduino.cc)

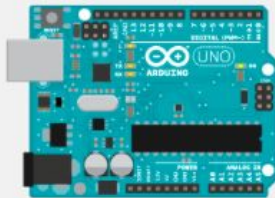


Search the Arduino Website

[Home](#) [Buy](#) [Download](#) [Products](#) [Learning](#) [Forum](#) [Support](#) [Blog](#)

[LOG IN](#) [SIGN UP](#)

WHAT IS ARDUINO?



[BUY AN ARDUINO](#)



[LEARN ARDUINO](#)



Arduino's sister brand for products sold outside the US.



[BLOG](#)

5 ROBOTIC BARTENDERS TO CELEBRATE CINCO DE MAYO



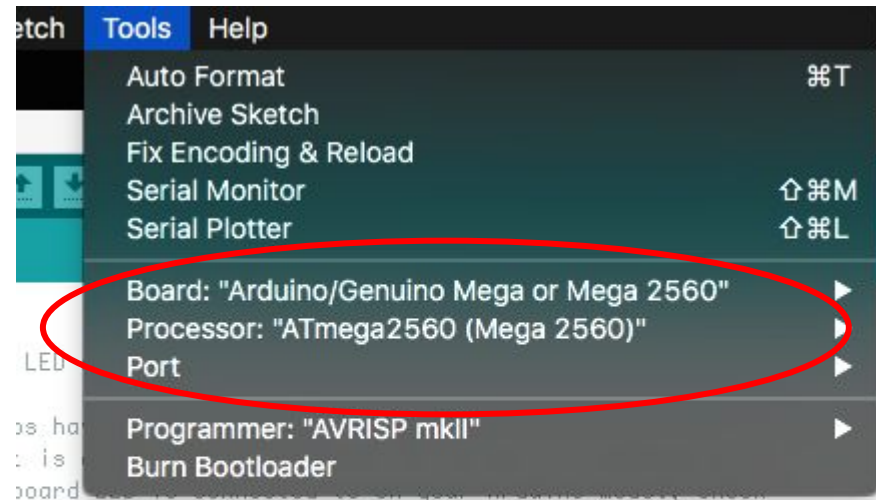
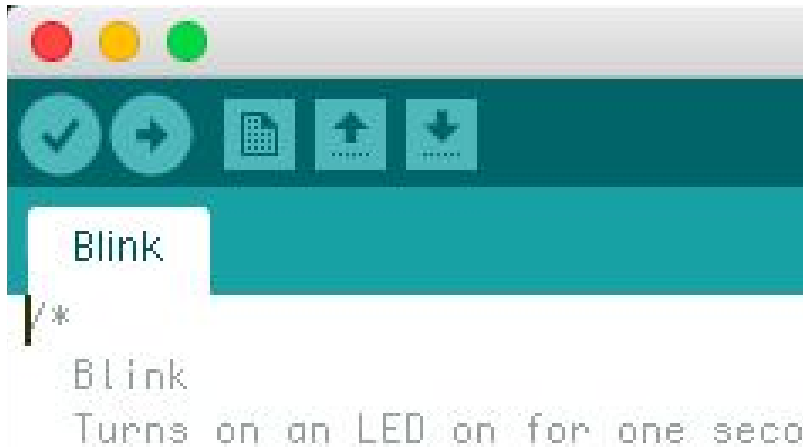
[GENUINO UNO, THE CLASSIC BOARD TO GET STARTED!](#)  
[SHOP NOW](#)



[BLOG](#)

# Προγραμματισμός σε Arduino IDE

cansat.gr/downloads/workshop.zip



# Προγραμματισμός

## loop ()

- Η λειτουργία αυτή μας επιτρέπει να καλούμε το περιεχόμενο της διαδοχικά επ' αόριστον.
- Χρησιμοποιείτε για να ελέγξει τις εισόδους του μικροελεγκτή, να επικοινωνήσει με άλλες συσκευές και να βγάλει εξόδους σε διάφορες μορφές.
- Κάθε φορά που εκτελείτε το περιεχόμενο της ο μικροελεγκτής την **τρέχει ξανά από την αρχή**.

## Παράδειγμα κώδικα

```
int buttonPin = 3;
void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}
void loop()
{
  if (digitalRead(buttonPin) == HIGH)
    Serial.write('H');
  else
    Serial.write('L');
  delay(1000);
}
```

# Προγραμματισμός

## setup ()

- Η λειτουργία αυτή καλείται όταν ξεκινά ένα πρόγραμμα.
- Χρησιμοποιείται για να προετοιμάσει μεταβλητές, τους τρόπους επαφών, εισαγωγή βιβλιοθηκών, κ.λπ.
- Η λειτουργία **setup()** θα τρέξει **μόνο μία φορά**, μετά από κάθε εκκίνηση ή επαναφορά (reset) τις πλακέτας Arduino.

## Παράδειγμα κώδικα

```
int buttonPin = 3;
```

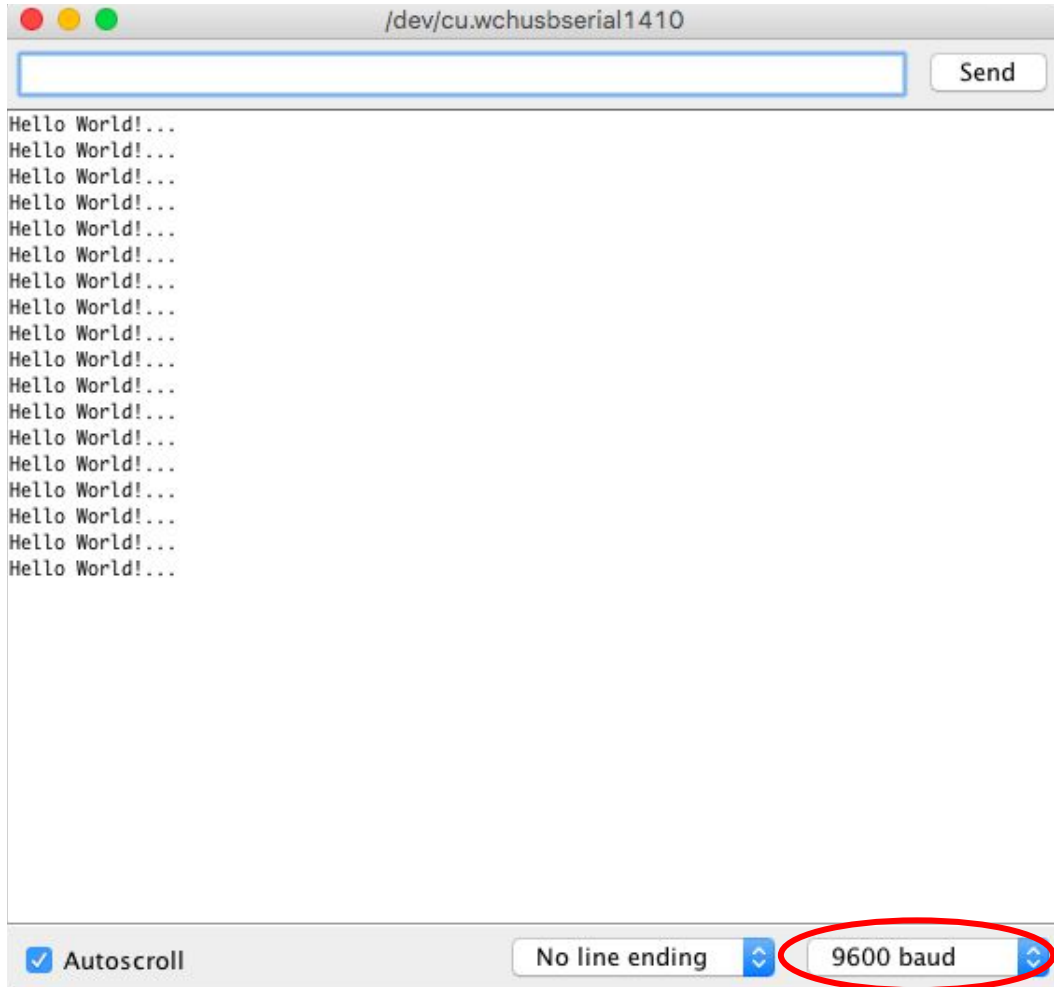
### void setup ()

```
{  
  Serial.begin (9600);  
  pinMode (buttonPin, INPUT);  
}
```

```
///...///
```

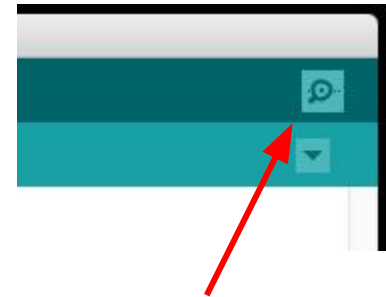


# ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΤΟΝ ΥΠΟΛΟΓΙΣΤΗ ΜΕΣΩ USB



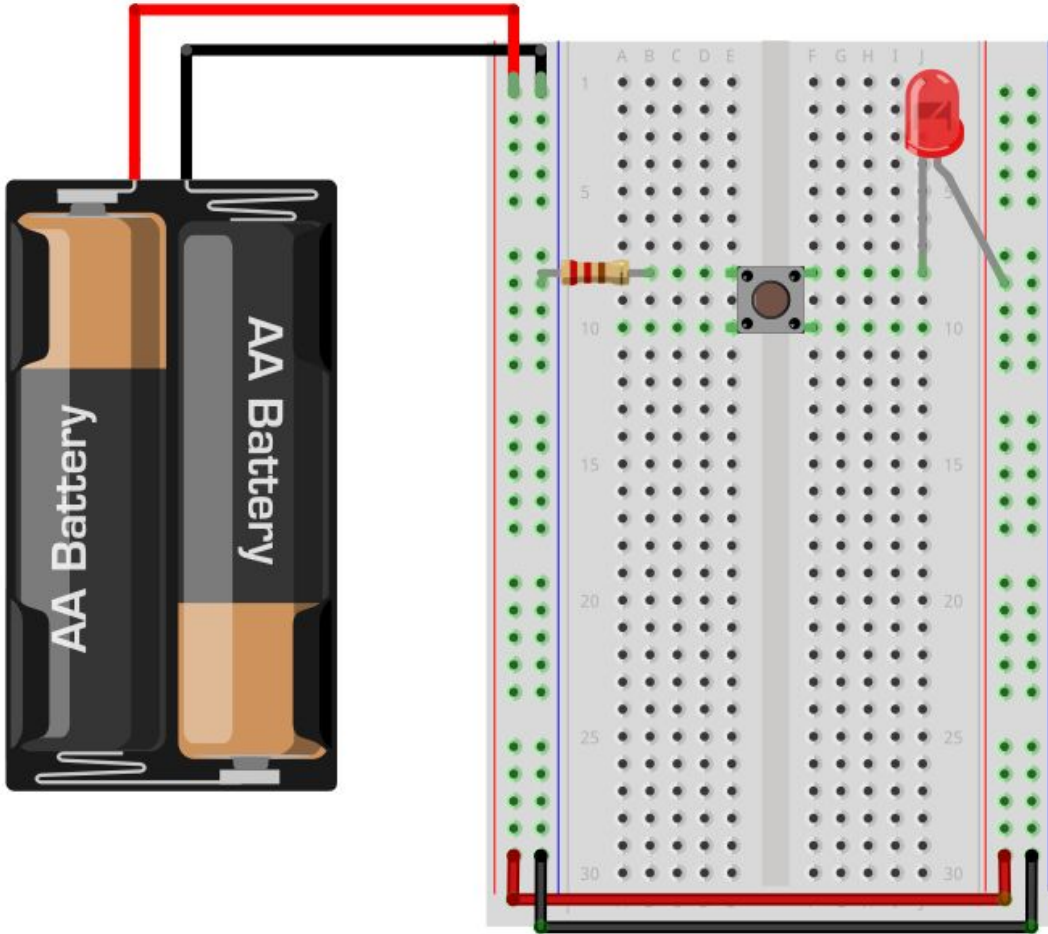
## Κώδικας

```
void setup()  
{  
  Serial.begin(9600);  
}  
void loop()  
{  
  Serial.println("Hello World!");  
  delay(1000);  
}
```



Serial Monitor

# Πλακέτα εύκολων συνδέσεων Breadboard



## Μεσαίο μέρος

- οι τρύπες συνδέονται οριζόντια.
- χωρίζετε στην μέση για ευκολία συνδέσεων.

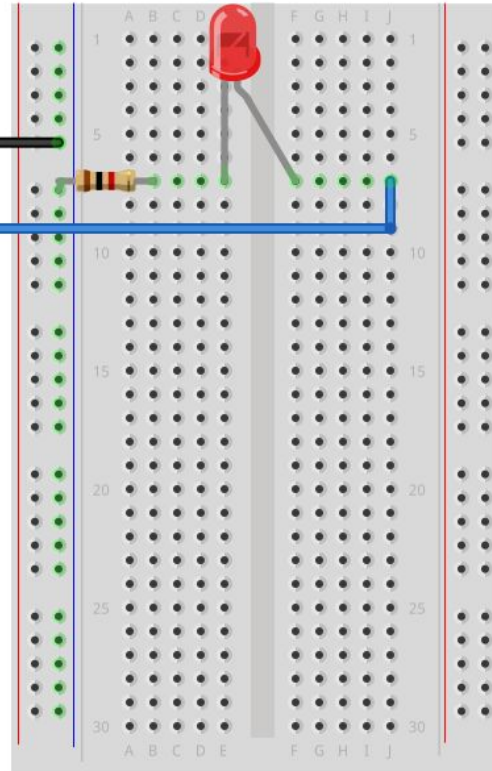
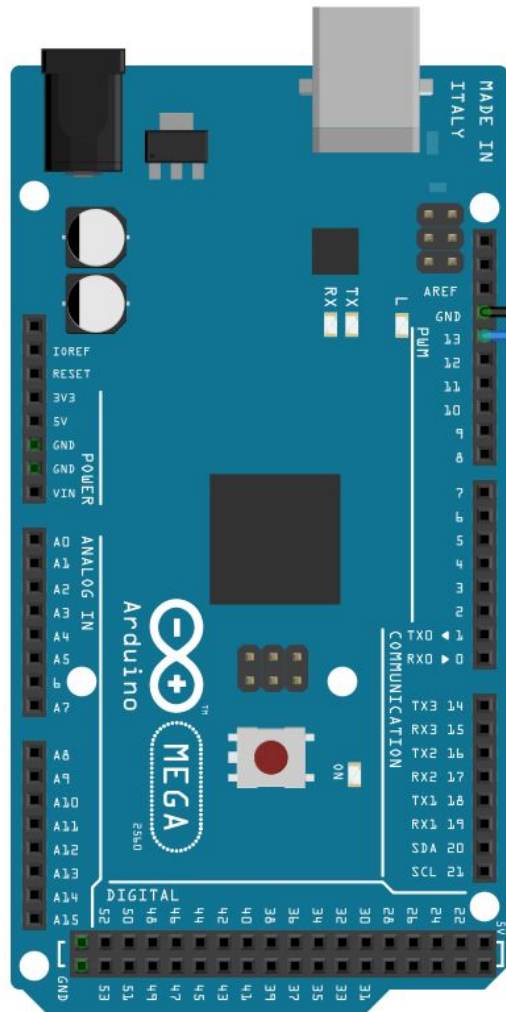
## Πλαϊνό Μέρος

- οι τρύπες συνδέονται κάθετα.
- χρησιμοποιείται συνήθως για γραμμές τροφοδοσίας

# L.E.D. Blink

## Κώδικας

```
int led = 13;  
void setup()  
{  
  pinMode(led, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

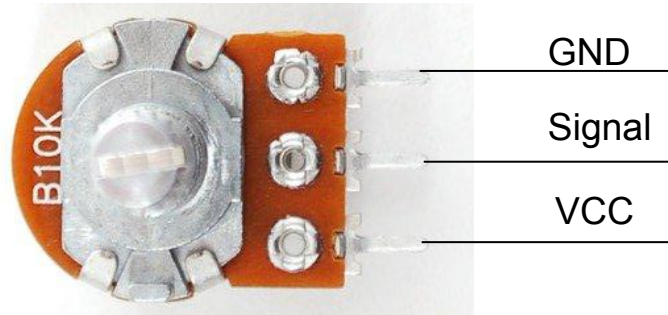
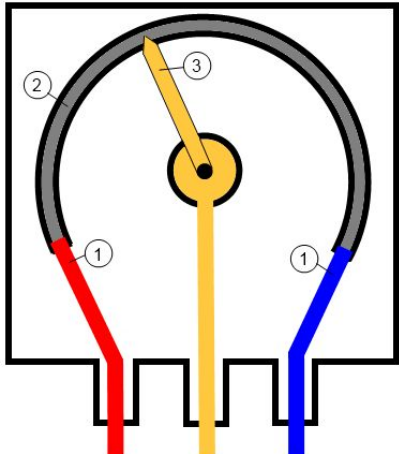


(+) anode  
(-) cathode



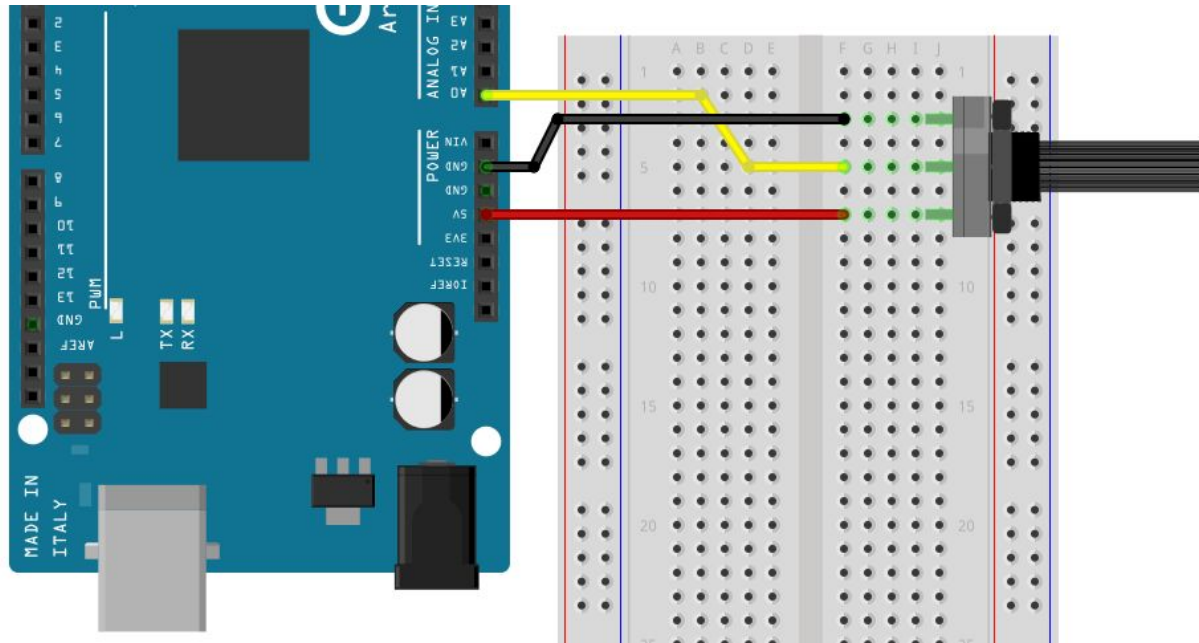
1KΩ

# Αναλογική ανάγνωση Ποτενσιόμετρου

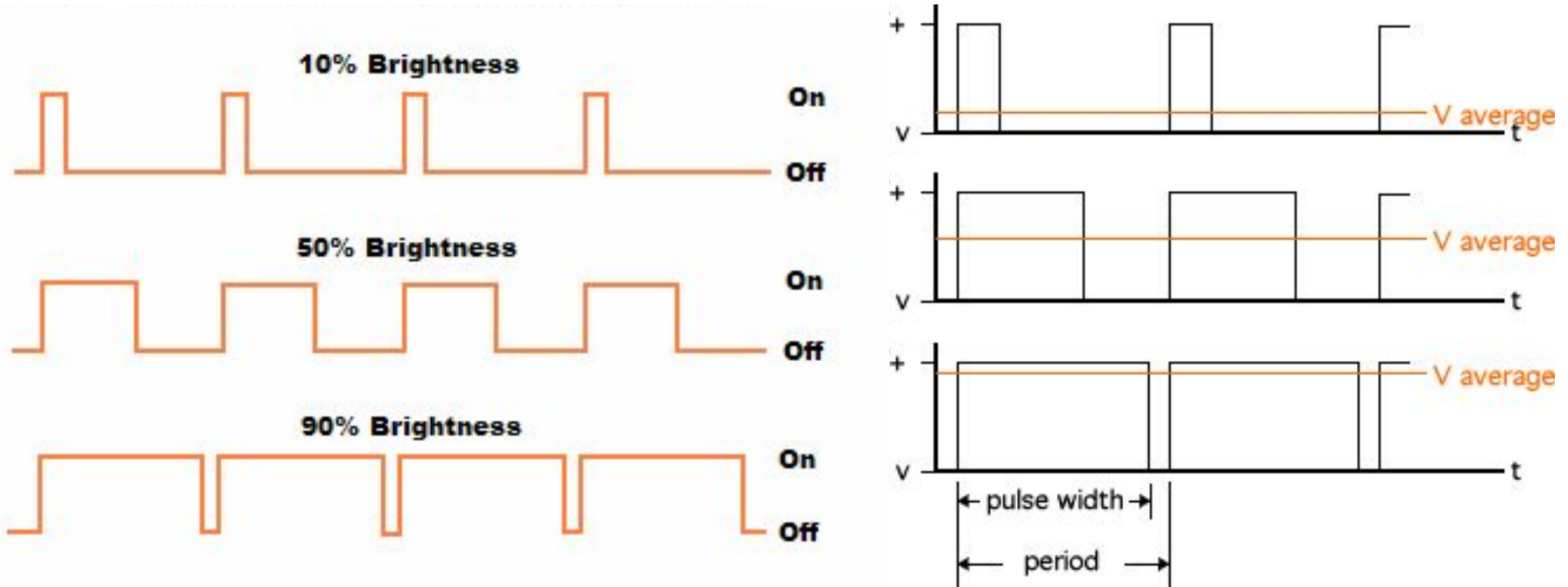


## Κώδικας

```
int potPin = A0;  
int value;  
void setup()  
{  
  Serial.begin(9600);  
}  
void loop()  
{  
  value = analogRead(potPin);  
  Serial.println(value);  
  delay(10);  
}
```



# PWM = Pulse Width Modulation



**analogWrite(pin-name, value)**

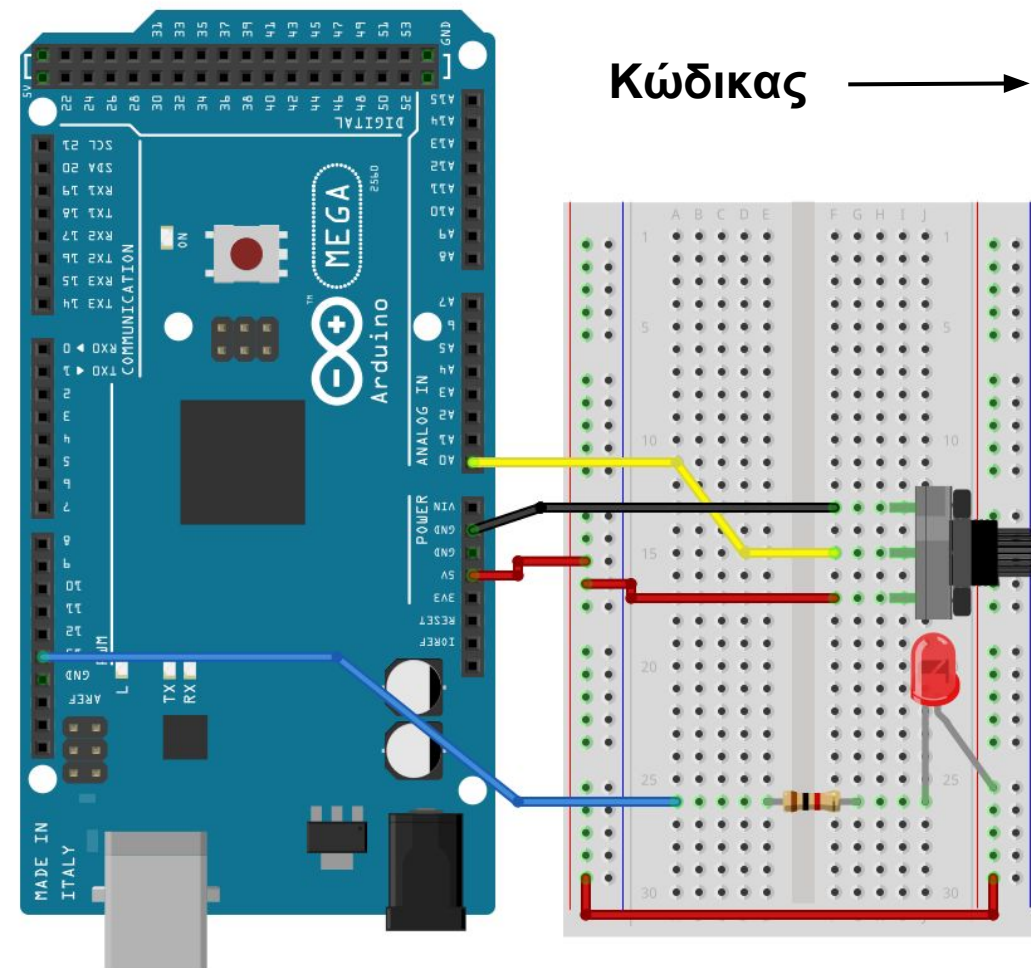
- υλοποιεί PWM αναθέτοντας το value στο dutycycle της παλμοσειράς.
- το value παίρνει τιμές 0-255 (8bit).

# Έλεγχος φωτεινότητας Led μέσω Ποτενσιόμετρου με χρήση PWM

```
int ledPin = 9;  
int potPin = A0;  
int brightness; // 0-255 (8bit)  
int value; // 0-1023 (10bit)
```

Κώδικας →

```
void setup(){  
  pinMode(ledPin, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop(){  
  value = analogRead(potPin);  
  brightness = map(value,0,1023,0,255);  
  analogWrite(ledPin, brightness);  
  Serial.println(value);  
}
```



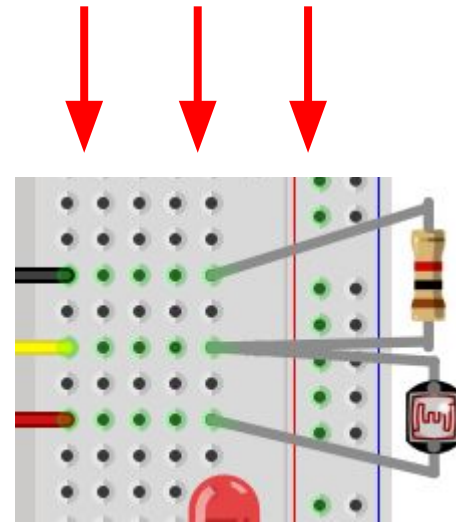
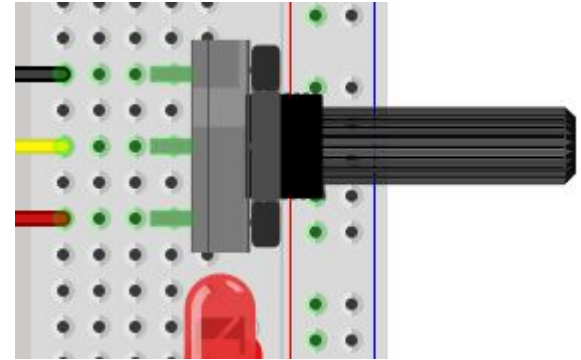
# Αυτόματη λάμπα με χρήση φωτοκύτταρου

Στην breadboard αλλάζουμε μόνο το ποτενσιόμετρο με μία αντίσταση και το φωτοκύτταρο.

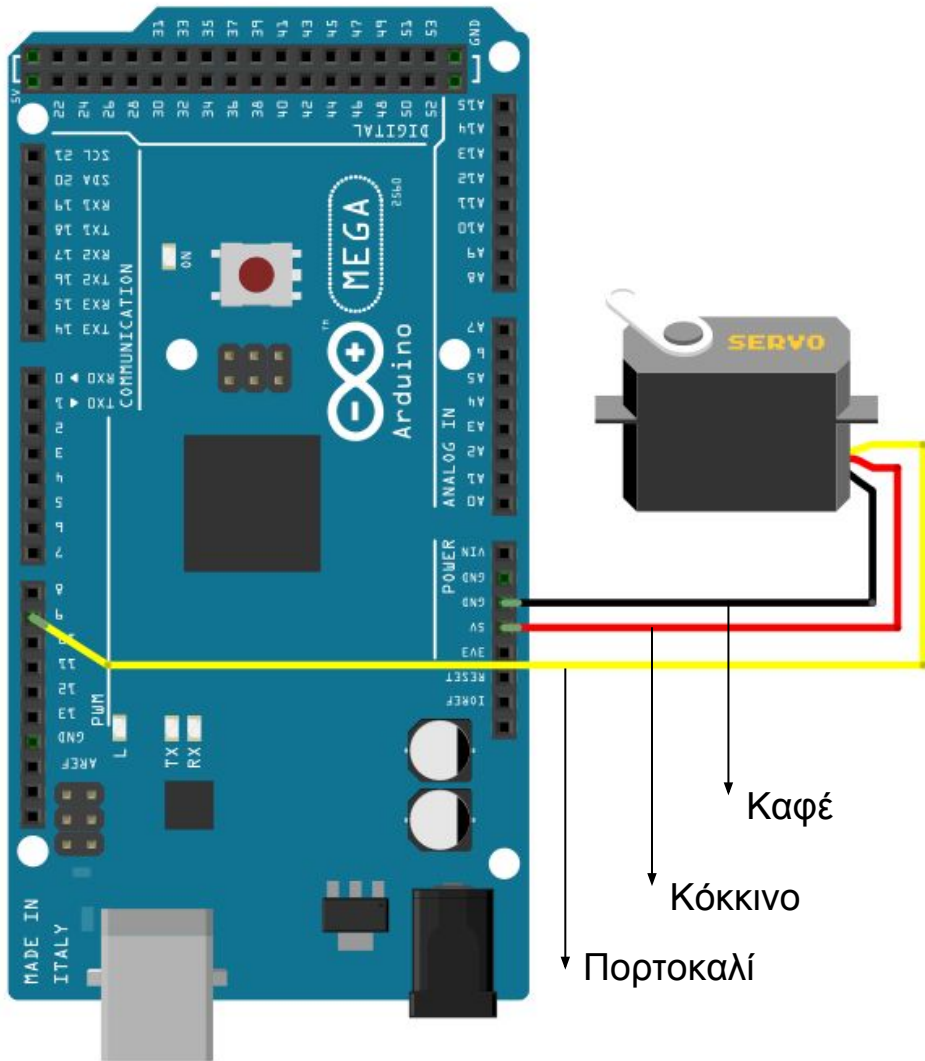
Ο κώδικας παραμένει ο ίδιος. **Γιατί;**

## Ερώτηση:

Πώς θα προσαρμόσουμε τον κώδικα μας ώστε το led να ανταποκρίνεται καλύτερα στο φως του δωματίου?



# Έλεγχος κινητήρα Servo



## Κώδικας

```
#include <Servo.h>
Servo myservo;
int servoPin = 9;
int pos = 0; // 0-180 degrees
void setup() {
  myservo.attach(servoPin);
}
void loop() {
  for (pos = 0; pos <= 180; pos++){
    myservo.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos--){
    myservo.write(pos);
    delay(15);
  }
}
```



# Ρομποτικός Βραχίονας

## **Τηλεχειριζόμενος βραχίονας:**

Συνδυάσετε τον κινητήρα με το ποτενσιόμετρο για να φτιάξετε ένα ρομποτικό βραχίονα. Χρησιμοποιήστε την συνδεσμολογία του ποτενσιόμετρου που είδαμε νωρίτερα. Κάντε αλλαγές στον κώδικα του ελέγχου φωτεινότητας LED μέσω ποτενσιόμετρου ώστε η έξοδος να είναι ο κινητήρας. Προσοχή ο κινητήρας δέχεται μοίρες (0-180).

## **Αυτόματος βάση περιστροφής φωτοβολταϊκών πάνελ:**

Συνδυάσετε τον κινητήρα με το φωτοκύτταρο για να φτιάξετε μια βάση περιστροφής φωτοβολταϊκών πάνελ ανάλογα με τον ήλιο. Χρησιμοποιήστε την συνδεσμολογία του φωτοκύτταρου που είδαμε νωρίτερα. Κάντε αλλαγές στον κώδικα ελέγχου του κινητήρα ώστε όταν η φωτεινότητα ξεπερνάει ένα όριο ο κινητήρας να αλλάζει σταδιακά θέση από 180 μοίρες σε 0.



Ευχαριστούμε! Απορίες;

Για εκδήλωση ενδιαφέροντος και ερωτήσεις:  
[info@cansat.gr](mailto:info@cansat.gr)