Μηχατρονικά Συστήματα Ι
Τμήμα Μηχανολόγων Μηχανικών Τ.Ε. Τ.Ε.Ι. Κρήτης

Παραδείγματα χρήσης του μικροελεγκτή Arduino

Εφαρμογές για το εργαστήριο

Ενότητα 2η :Προγραμματισμός μικροελεγκτή

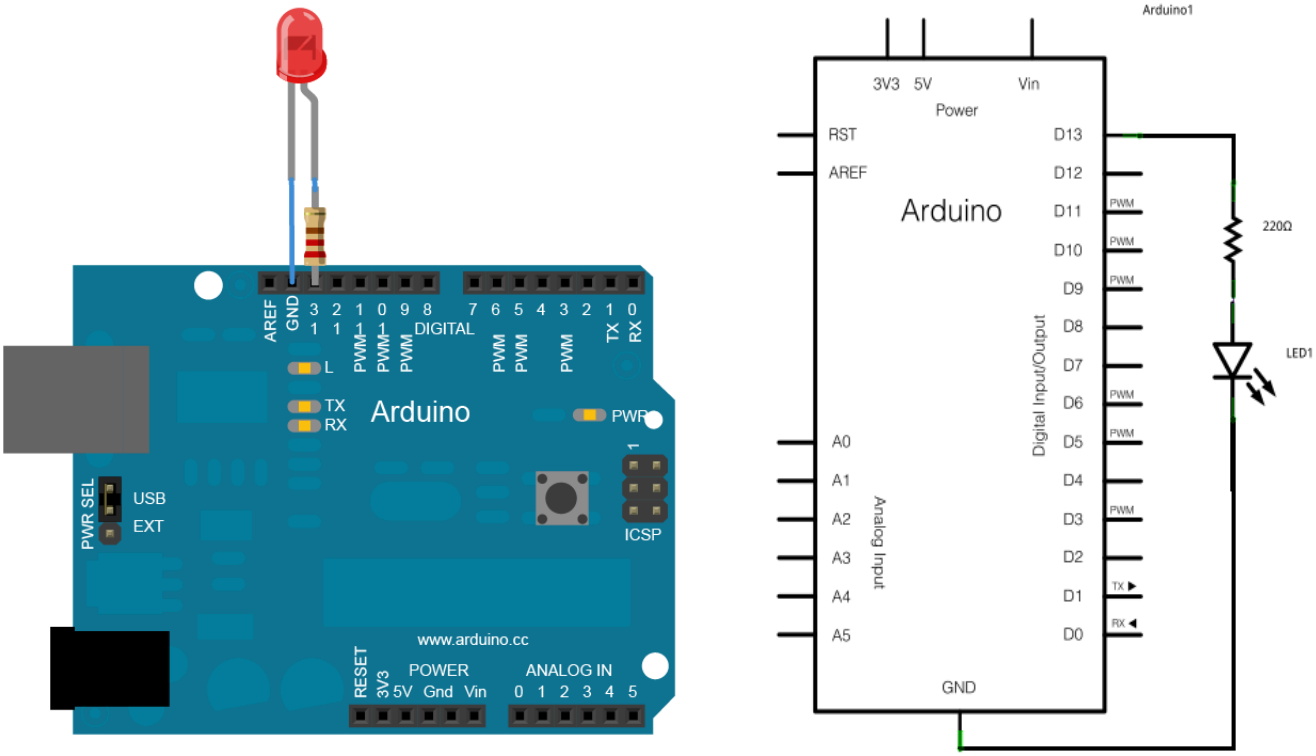
Δρ. Φασουλάς Γιάννης, Επίκ. Καθηγητής

Νοέμβριος 2016

Παραδείγματα μπορείτε να βρείτε στην διεύθυνση http://www.arduino.cc/en/Tutorial/HomePage

Δραστηριότητα 1: Ενεργοποίηση και απενεργοποίηση LED με την συνάρτηση delay()

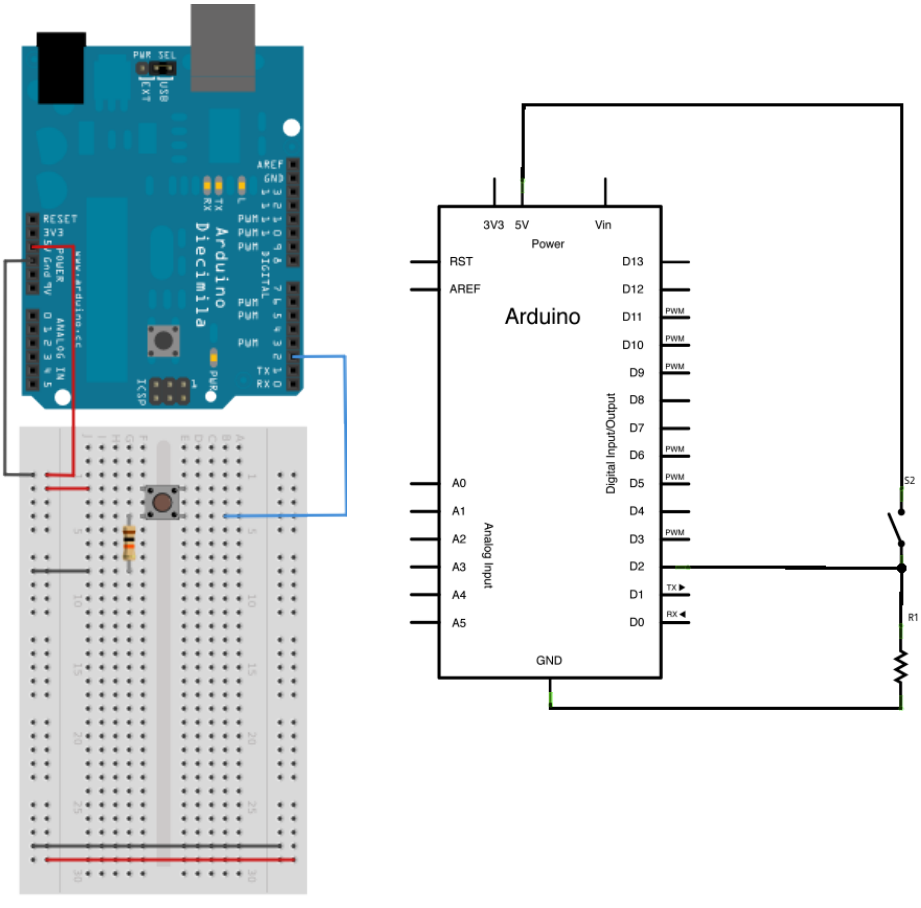
```
 http://arduino.cc/en/Tutorial/Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  */

int led = 13; // Pin 13 has an LED connected on most Arduino boards, give it a name:

void setup(){              // the setup routine runs once when you press reset:
  pinMode(led, OUTPUT);    // initialize the digital pin as an output.
            }

void loop()  {               // the loop routine runs over and over again forever:
  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
            }
```



Δραστηριότητα 2: Ενεργοποίηση και απενεργοποίηση LED με μπουτόν

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

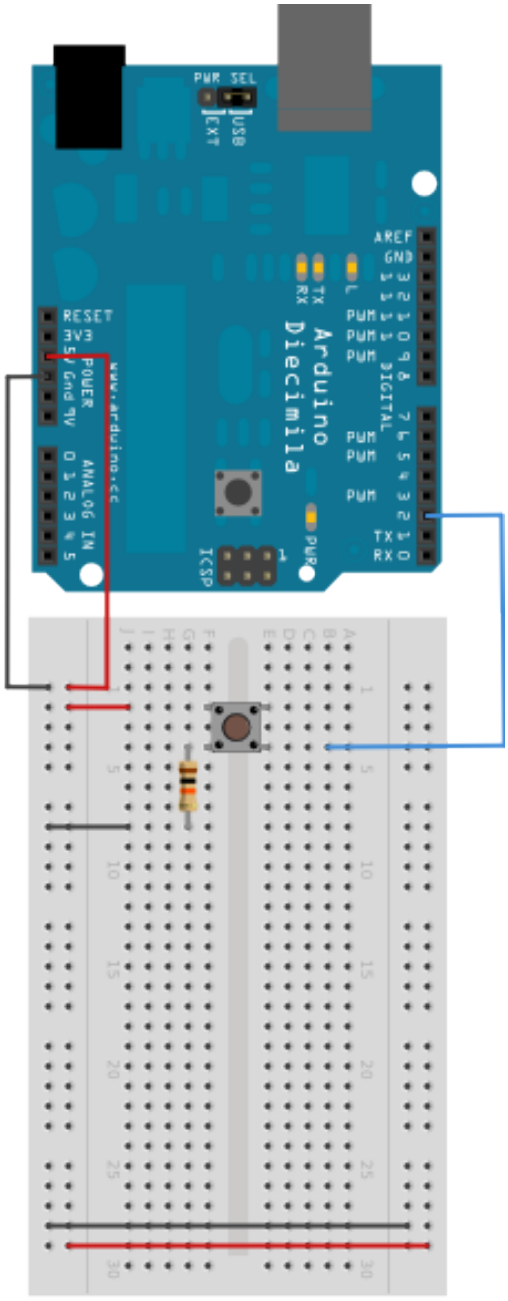
```
/*
  Button

 Turns on and off a light emitting diode(LED) connected to digital
 pin 13, when pressing a pushbutton attached to pin 2.

 * Note: on most Arduinos there is already an LED on the board
 attached to pin 13.
  */

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;      // variable for reading the pushbutton status

void setup() {
  pinMode(ledPin, OUTPUT);       // initialize the LED pin as an output:
  pinMode(buttonPin, INPUT);      // initialize the pushbutton pin as an input:
             }

void loop(){
  buttonState = digitalRead(buttonPin); // read the state of the pushbutton value:
  if (buttonState == HIGH) {            // check if the pushbutton is pressed.
      digitalWrite(ledPin, HIGH);      // if it is, the buttonState is HIGH, turn LED on:
                            }
  else {
        digitalWrite(ledPin, LOW); // turn LED off:
       }
           }
```

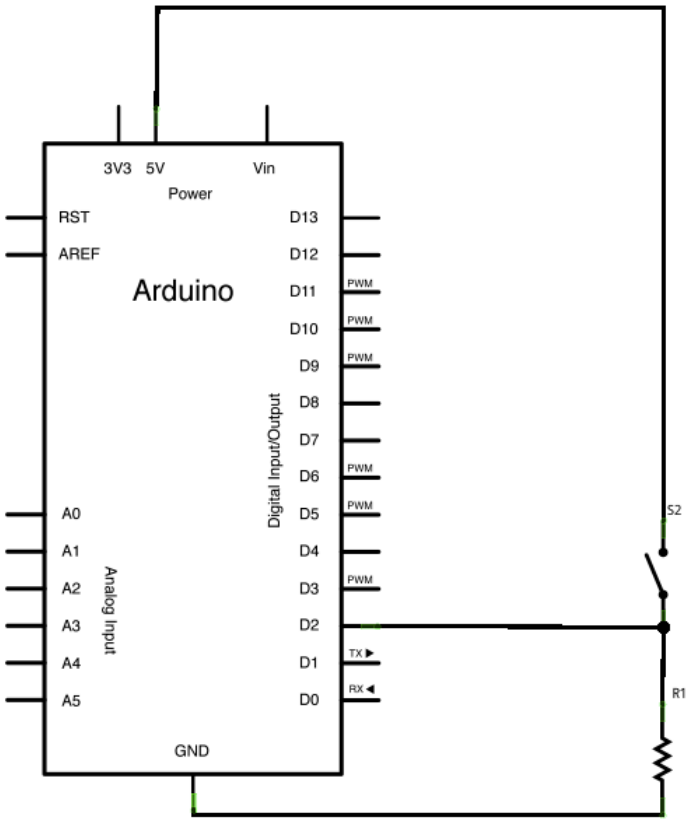


Δραστηριότητα 3: Αποστολή της κατάστασης μίας ψηφιακής εισόδου στο σειριακό μόνιτορ του υπολογιστή.

http://arduino.cc/en/Tutorial/DigitalReadSerial

```
/*
 Reads a digital input on pin 2, prints the result to the serial monitor
  */

int pushButton = 2; // digital pin 2 has a pushbutton attached to it. Give it a name:

void setup() {  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  pinMode(pushButton, INPUT);    // make the pushbutton's pin an input:
              }

void loop() {
  int buttonState = digitalRead(pushButton); // read the input pin:
  Serial.println(buttonState);                // print out the state of the button:
  delay(1);                                  // delay in between reads for stability
            }
```

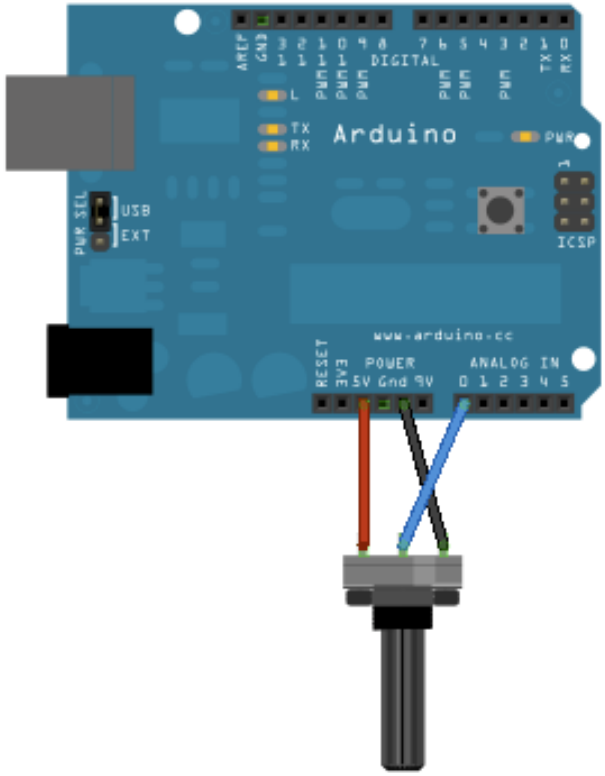


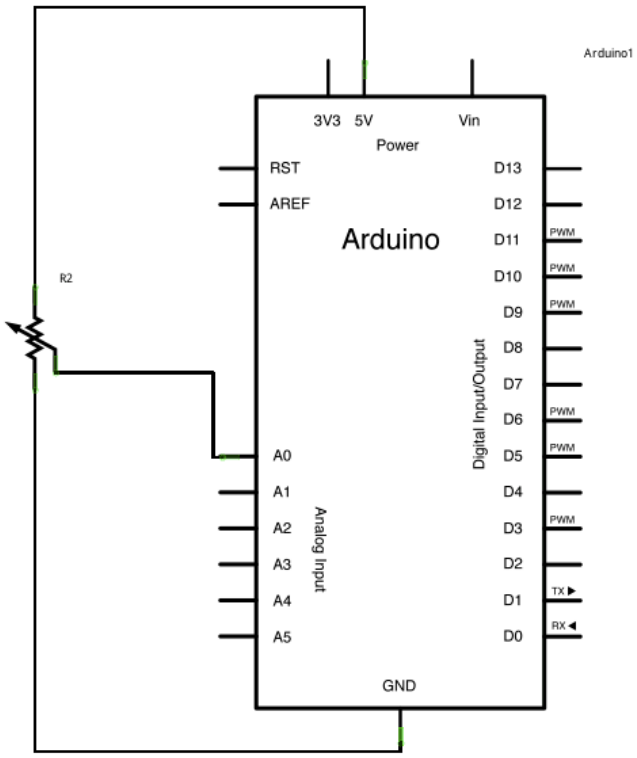


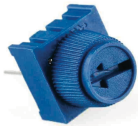
Δραστηριότητα 4: Καταγραφή αναλογικού σήματος και άμεση αποστολή της αντίστοιχης ψηφιακής τιμής στο σειριακό μόνιτορ του υπολογιστή.

http://arduino.cc/en/Tutorial/AnalogReadSerial

```
/*
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.
 */

void setup() {
  Serial.begin(9600);   // initialize serial communication at 9600 bits per second:
            }

void loop() {

  int sensorValue = analogRead(A0); // read the input on analog pin 0:
  Serial.println(sensorValue);     // print out the value you read and change line:
  delay(1);                       // delay in between reads for stability
           }
```



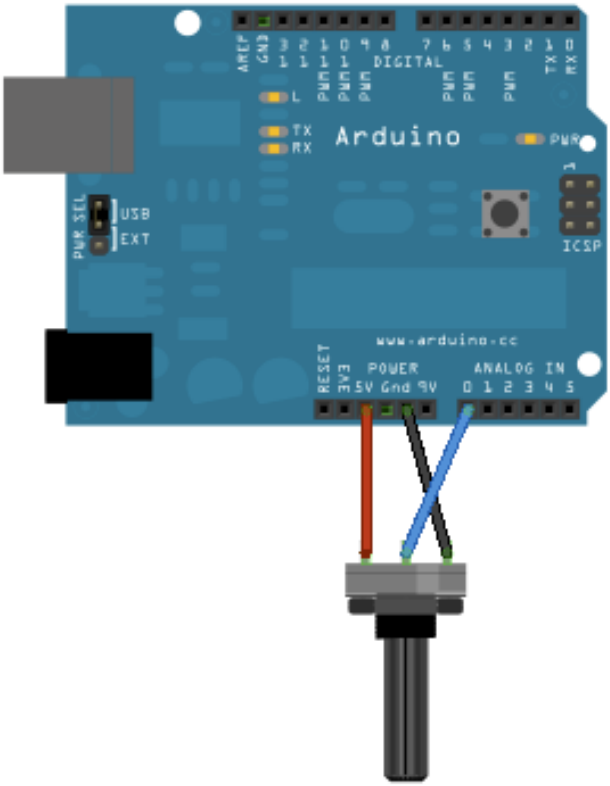


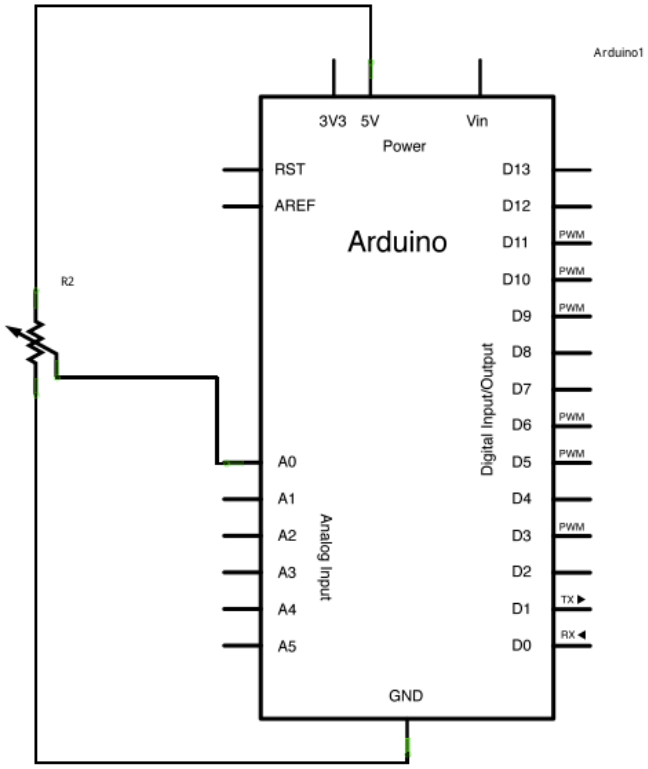


Δραστηριότητα 5: Υλοποίηση ψηφιακού βολτομέτρου με την χρήση του μικροελεγκτή

http://arduino.cc/en/Tutorial/ReadAnalogVoltage

```
/*
  ReadAnalogVoltage
  Reads an analog input on pin 0, converts it to voltage, and prints the result to the serial
  monitor. Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and
ground.
  */

void setup() {
  Serial.begin(9600); // initialize serial communication at 9600 bits per second:
            }

void loop() {
  int sensorValue = analogRead(A0);               // read the input on analog pin A0
  float voltage = sensorValue * (5.0 / 1024.0); // compute Q (analog quantization size)
  Serial.println(voltage);                    // print out the value you read:
            }
```

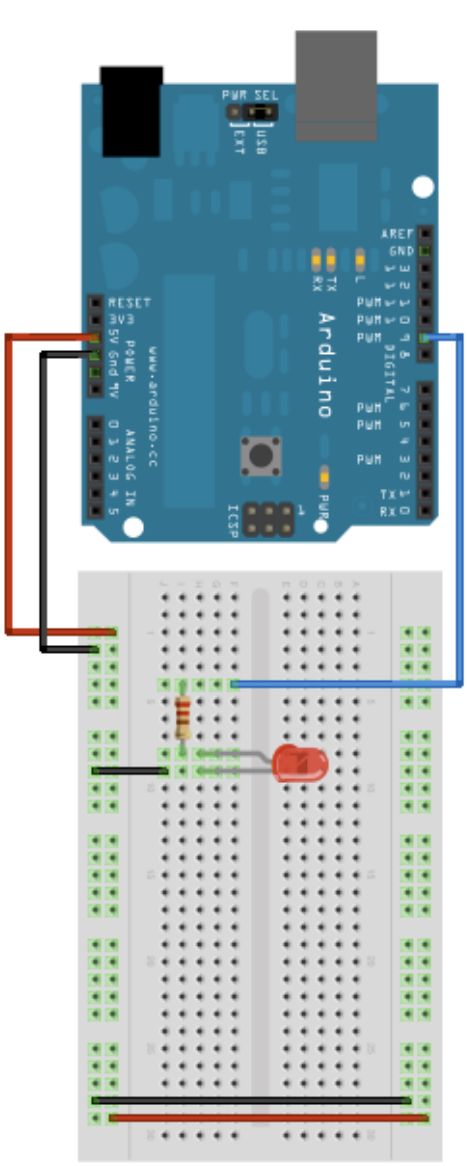


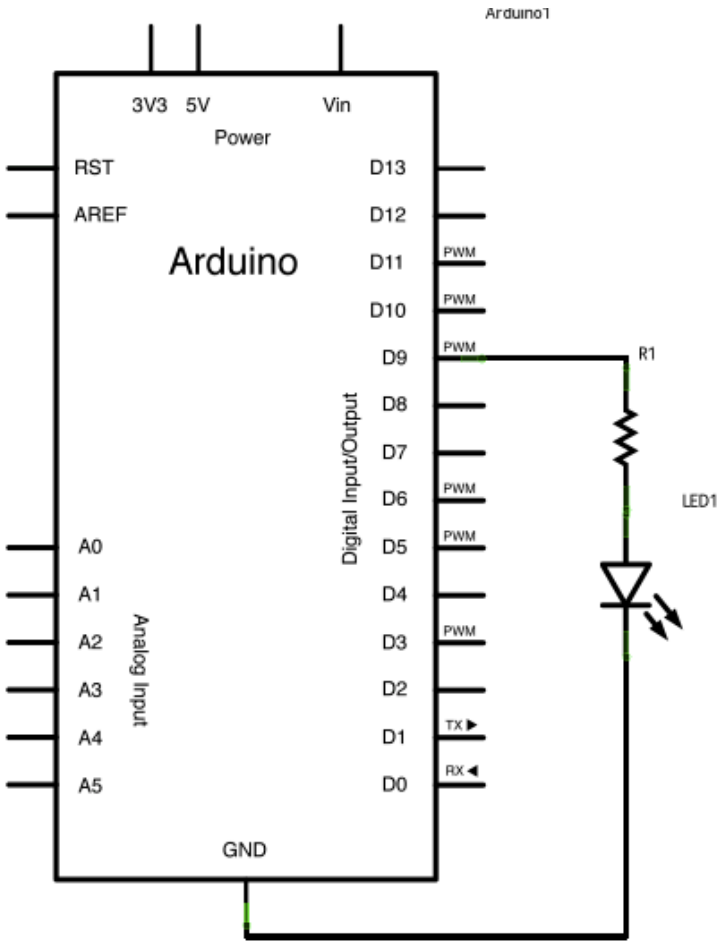


Δραστηριότητα 6: Ρύθμιση φωτεινότητας LED με ψευδοαναλοκή έξοδο PWM

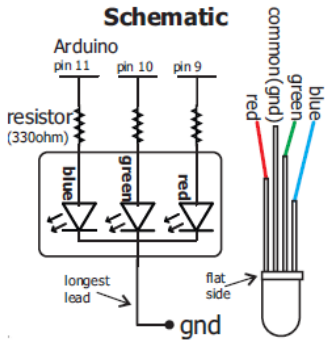
http://arduino.cc/en/Tutorial/Fade

```
/*
 This example shows how to fade an LED on pin 9 using the analogWrite() function.
 */

int led = 9;           // the pin that the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

void setup()  {
  pinMode(led, OUTPUT); // declare pin 9 to be an output:
              }

void loop()  {
  analogWrite(led, brightness);               // set the brightness of pin 9:
  brightness = brightness + fadeAmount;      // change the brightness for next time through the loop
    if (brightness == 0 || brightness == 255) {// reverse the direction of the fading at the ends of
                                                                                    //the fade
                                            fadeAmount = -fadeAmount ;
                                           }
  delay(30);                          // wait for 30 milliseconds to see the dimming effect
              }
```





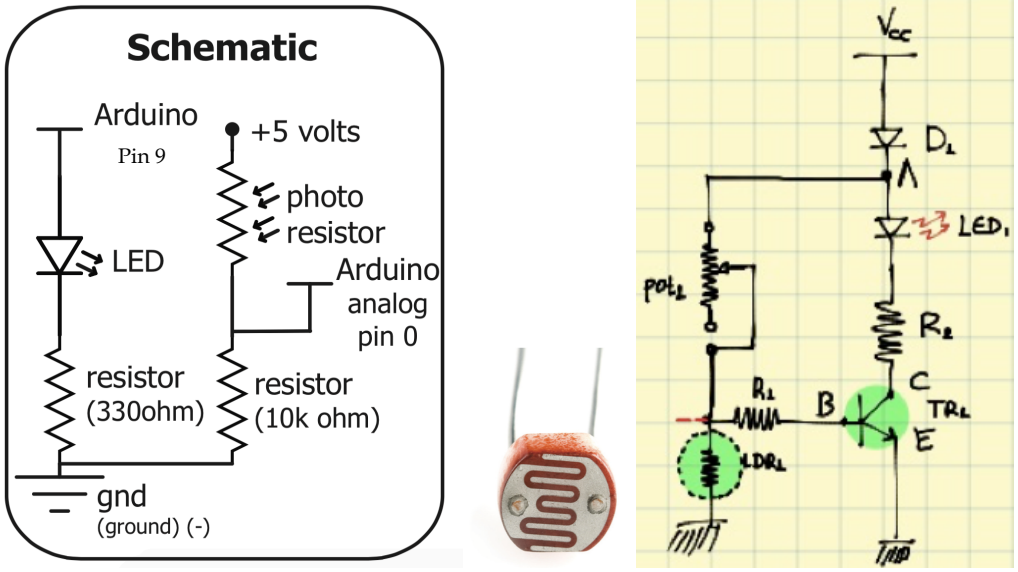
Δραστηριότητα 7: Ρύθμιση φωτεινότητας LED τριπλού χρώματος RGB με εξόδους PWM




```
// LED leads connected to PWM pins
const int RED_LED_PIN = 9;
const int GREEN_LED_PIN = 10;
const int BLUE_LED_PIN = 11;

                      // Used to store the current intensity level of the individual LEDs
int redIntensity = 0;
int greenIntensity = 0;
int blueIntensity = 0;

                     // Length of time we spend showing each color
const int DISPLAY_TIME = 100; // In milliseconds


void setup() {  // No setup required.}

void loop() {
                // Cycle color from red through to green
                // (In this loop we move from 100% red, 0% green to 0% red, 100% green)
  for (greenIntensity = 0; greenIntensity <= 255; greenIntensity+=5) {
        redIntensity = 255-greenIntensity;
        analogWrite(GREEN_LED_PIN, greenIntensity);
        analogWrite(RED_LED_PIN, redIntensity);
        delay(DISPLAY_TIME);
  }

                // Cycle color from green through to blue
                // (In this loop we move from 100% green, 0% blue to 0% green, 100% blue)
  for (blueIntensity = 0; blueIntensity <= 255; blueIntensity+=5) {
        greenIntensity = 255-blueIntensity;
        analogWrite(BLUE_LED_PIN, blueIntensity);
        analogWrite(GREEN_LED_PIN, greenIntensity);
        delay(DISPLAY_TIME);
  }

                // Cycle cycle from blue through to red
                // (In this loop we move from 100% blue, 0% red to 0% blue, 100% red)
  for (redIntensity = 0; redIntensity <= 255; redIntensity+=5) {
        blueIntensity = 255-redIntensity;
        analogWrite(RED_LED_PIN, redIntensity);
        analogWrite(BLUE_LED_PIN, blueIntensity);
        delay(DISPLAY_TIME);
  }
}
```

Δραστηριότητα 8: Έλεγχος φωτεινότητας με Φωτοαντίσταση LDR

Σε αυτή την εργαστηριακή άσκηση χρησιμοποιούμε μια φωτοαντίσταση ως αισθητήριο μέτρησης φωτεινότητας, η οποία με την σειρά της θέλουμε να ρυθμίζει την φωτοβολία ενός LED που συνδέουμε στο pin 13 του μικροελεγκτή. Για την μέτρηση της φωτεινότητας υλοποιείστε τον διαιρέτη τάσης που παρουσιάζεται παρακάτω και οδηγείστε το σήμα μέτρησης από το διαιρέτη τάσης στην αναλογική είσοδο A0. Στην συνέχεια γράψτε τον κώδικα που δίνεται και φορτώστε τον στον μικροελεγκτή. Τέλος πειραματιστείτε με τη λειτουργία του αισθητήρα.



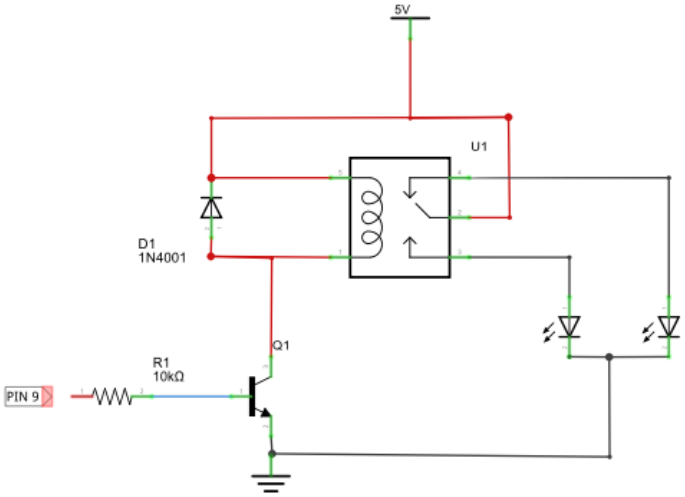
Αντίστοιχό κύκλωμα με τη χρήση τρανζίστορ

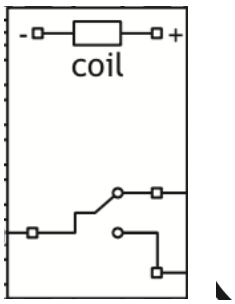
```
/*
 * A simple programme that will change the intensity of
 * an LED based  * on the amount of light incident on
 * the photo resistor.
 *
 */

//PhotoResistor Pin
int lightPin = 0; //the analog pin the photoresistor is
                  //connected to
                  //the photoresistor is not calibrated to any units so
                  //this is simply a raw sensor value (relative light)
//LED Pin
int ledPin = 9;   //the pin the LED is connected to
                  //we are controlling brightness so
                  //we use one of the PWM (pulse width
                  // modulation pins)

void setup() { pinMode(ledPin, OUTPUT); //sets the led pin to output }

void loop()
   {
       int lightLevel = analogRead(lightPin); //Read the
                                              // lightlevel
        lightLevel = map(lightLevel, 0, 900, 0, 255);
                //adjust the value 0 to 900 to
               //span 0 to 255

        lightLevel = constrain(lightLevel, 0, 255);//make sure the
                                                   //value is betwween
                                                  //0 and 255
             analogWrite(ledPin, lightLevel);  //write the value
   }
```

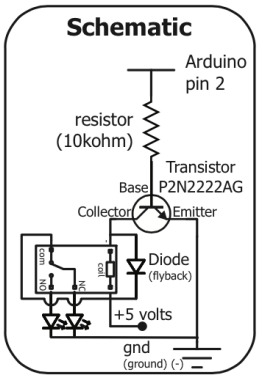
Δραστηριότητα 9: Οδήγηση Ρελέ με την χρήση τρανζίστορ και μικροελεγκτή

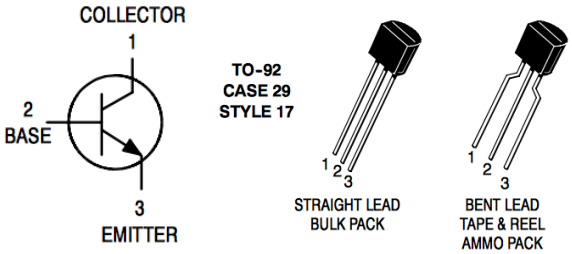
Πραγματοποιήστε το παρακάτω κύκλωμα προκειμένου να γίνει έλεγχος ενεργοποίησης του ρελέ μέσα από απο τον μικροελεγκτή Arduino.

```
int ledPin =  9;


void setup()   {

  pinMode(ledPin, OUTPUT);
}


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









MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Value	Unit
Collector-Emitter Voltage	V_{CEO}	40	Vdc
Collector-Base Voltage	V_{CBO}	75	Vdc
Emitter-Base Voltage	V_{EBO}	6.0	Vdc
Collector Current - Continuous	I_C	600	mAdc
Total Device Dissipation @ T_A = 25°C Derate above 25°C	P_D	625 5.0	mW mW/°C
Total Device Dissipation @ T_C = 25°C Derate above 25°C	P_D	1.5 12	W mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}	-55 to +150	°C