

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Arduino si indicatoare pentru CPU si memoria RAM

In acest tutorial vei descoperi cum se poate construi un bord cu 2 indicatoare pentru activitatea procesorului si capacitatea memoriei RAM. Vei descoperi cum se citesc cele 2 valori printr-un script Python si cum se vor transmite, serial, catre o placa Arduino.

Ca indicatoare vei folosi 2 servomotoare conectate la o placa Arduino. In functie de valorile primite pe seriala placa va modifica pozitia bratelor servomotoarelor.

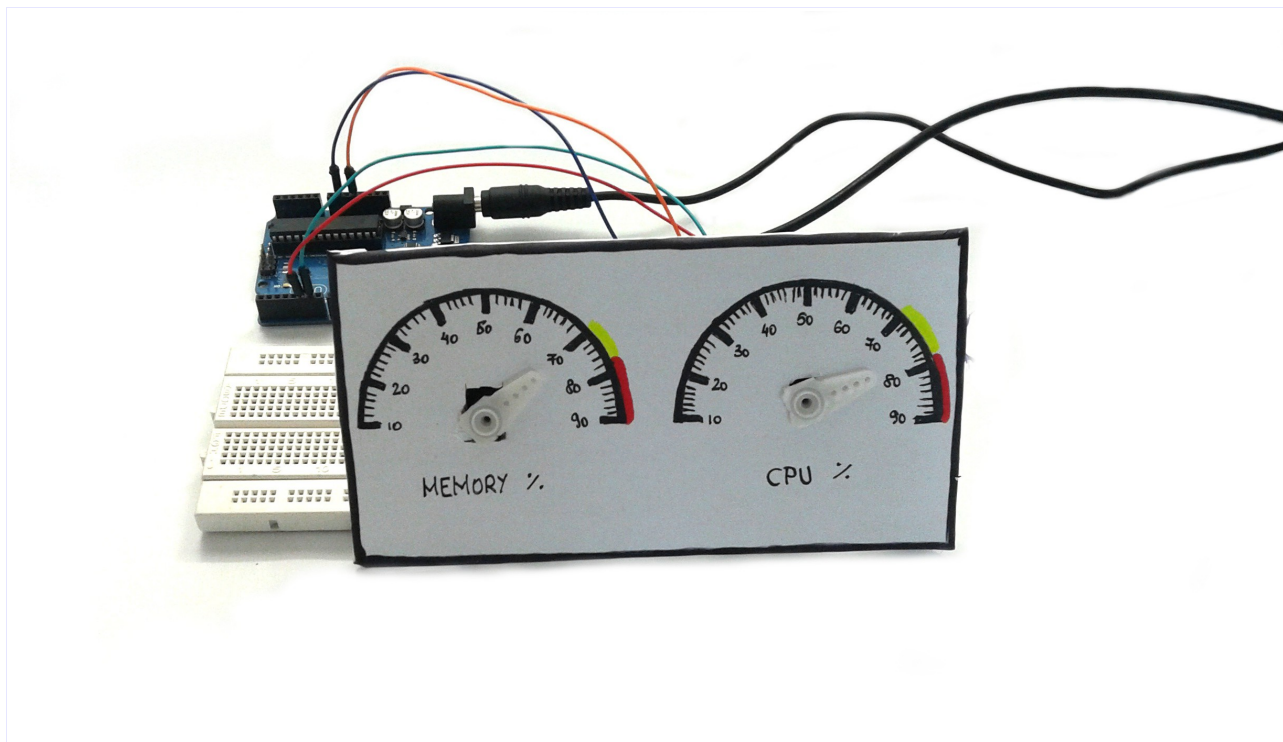
Vei avea nevoie de urmatoarele componente:

- O placa Arduino - <http://www.robofun.ro/arduino>
- 2 servomotoare - <http://www.robofun.ro/mecanice/servo>
- Un alimentator extern Arduino
http://www.robofun.ro/surse_de_alimentare/alimentatoare/alimentator-extern-arduino
- Un breadboard - <http://www.robofun.ro/breadboard>
- Un cablu USB - http://www.robofun.ro/cabluri/cablu_usb_1_5m_male_a_male_b

Cum construiesc indicatorul ?

In tutorialul de fata indicatorul cu cele 2 servomotoare a fost realizat dintr-o bucata de carton peste care s-a lipit o coala alba desenata cu 2 cadrane. S-au dat 2 gauri pe unde se fixeaza bratele servomotoarelor iar servomotoarele s-au lipit cu banda dublu adeziva.

Panoul va arata ca in imaginea de mai jos.



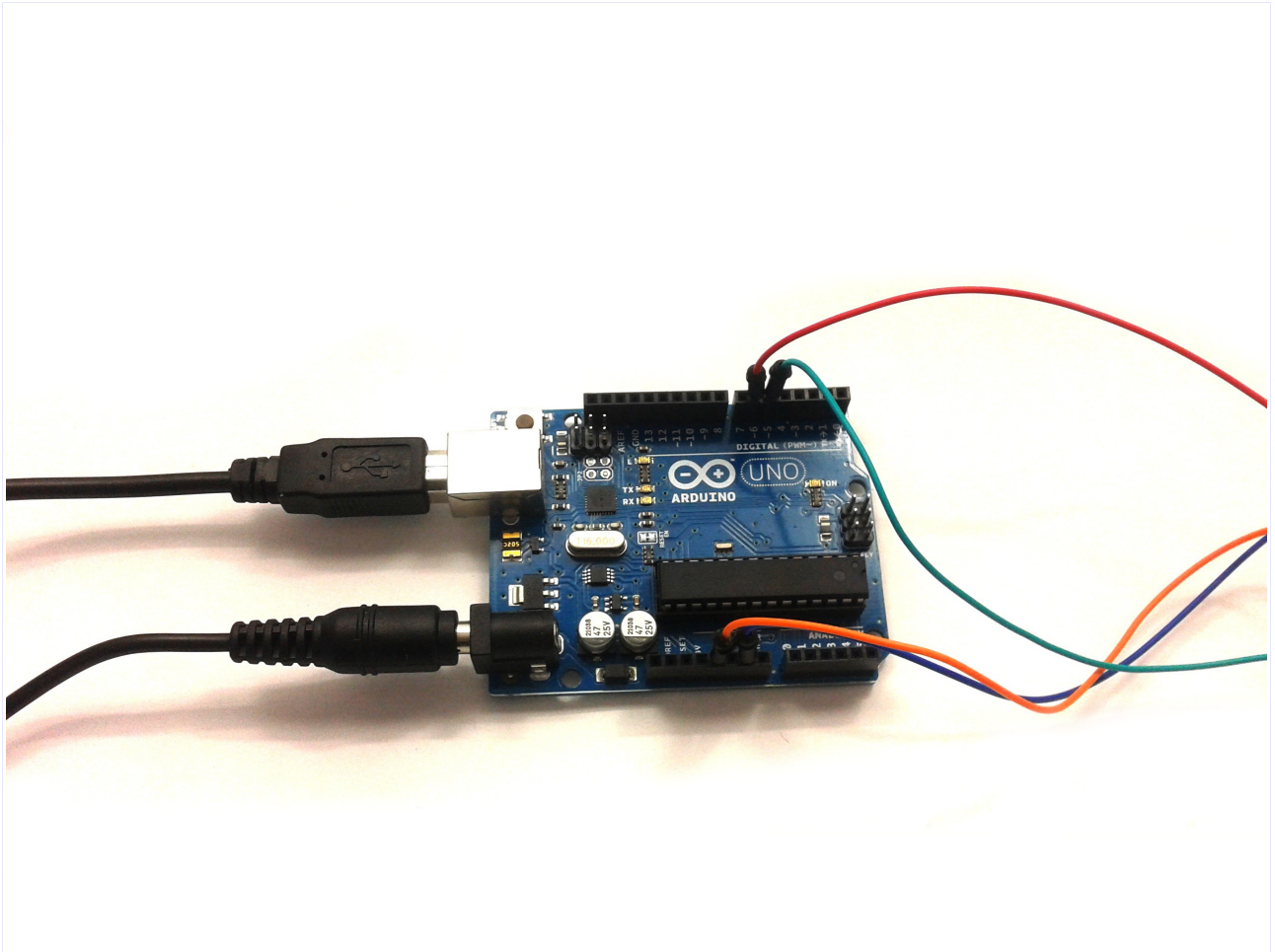
Cum se conecteaza servomotoarele ?

Cele 2 servomotoare se vor conecta urmand tabelul de mai jos. Te vei folosi de breadboard pentru a conecta alimentariile servomotoarelor.

Arduino 5V	Servomotor 1 fir Rosu
Arduino GND	Servomotor 1 fir Negru
Arduino digital 5	Servomotor 1 fir Alb
Arduino 5V	Servomotor 2 fir Rosu
Arduino GND	Servomotor 2 fir Negru
Arduino digital 6	Servomotor 2 fir Alb

Deoarece servomotoarele consuma mai mult decat poate oferi portul USB al calculatorului, vei conecta alimentatorul extern de 9V in mufa placii Arduino.

In final vei obtine urmatorul montaj.



Codul sursa pentru Arduino.

Codul sursa pentru placa Arduino este listat mai jos. Se copiaza cu copy/paste si se incarca in placa.

```
#include <Servo.h> //Used to control the Pan/Tilt Servos

//These are variables that hold the servo IDs.
char tiltChannel=0, panChannel=1;

//These are the objects for each servo.
Servo servoTilt, servoPan;

//This is a character that will hold data from the Serial port.
char serialChar=0;
```

```

void setup(){
    servoTilt.attach(5); //The Tilt servo is attached to pin 5.
    servoPan.attach(6); //The Pan servo is attached to pin 6.
    servoTilt.write(70); //Initially put the servos both
    servoPan.write(160); //at 90 degrees.
    //delay(5000);
    Serial.begin(9600); //Set up a serial connection for 9600 bps.
}

void loop(){
    while(Serial.available() <=0); //Wait for a character on the
    serial port.
    serialChar = Serial.read(); //Copy the character from the
    serial port to the variable
    if(serialChar == tiltChannel){ //Check to see if the character
    is the servo ID for the tilt servo
        while(Serial.available() <=0); //Wait for the second command
        byte from the serial port.
        servoTilt.write(180 - Serial.read()); //Set the tilt servo
        position to the value of the second command byte received on the
        serial port
    }
    else if(serialChar == panChannel){ //Check to see if the initial
    serial character was the servo ID for the pan servo.
        while(Serial.available() <= 0); //Wait for the second command
        byte from the serial port.
        servoPan.write(180 - Serial.read()); //Set the pan servo
        position to the value of the second command byte received from the
        serial port.
    }
    //If the character is not the pan or tilt servo ID, it is
    ignored.
}

```

Codul sursa Python.

Codul sursa de mai jos se executa in Python 2.7

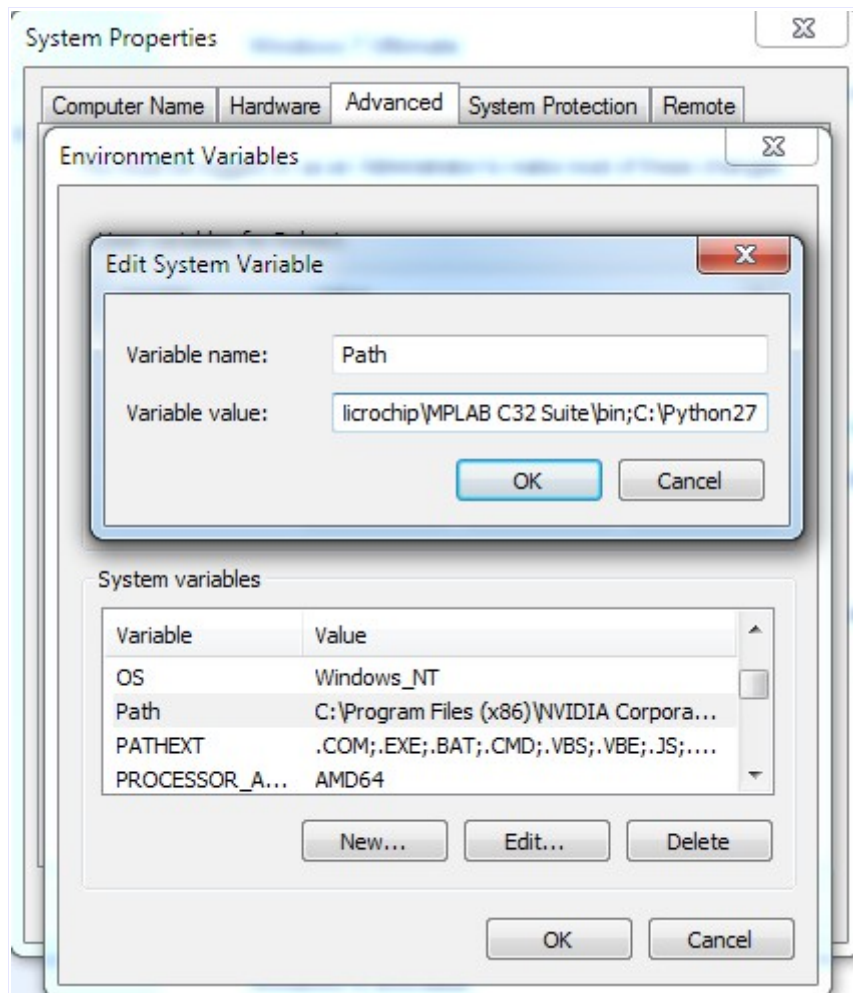
- Inainte de toate se va instala Python de la urmatoarea adresa:
<http://www.python.org/download/releases/2.7/>
- Se adauga linia de mai jos in System Path. In Windows 7 System Path se acceseaza din
 Computer – Properties – Advanced System Settings – Advanced – Environment Variables.

Linia ce trebuie adaugata:

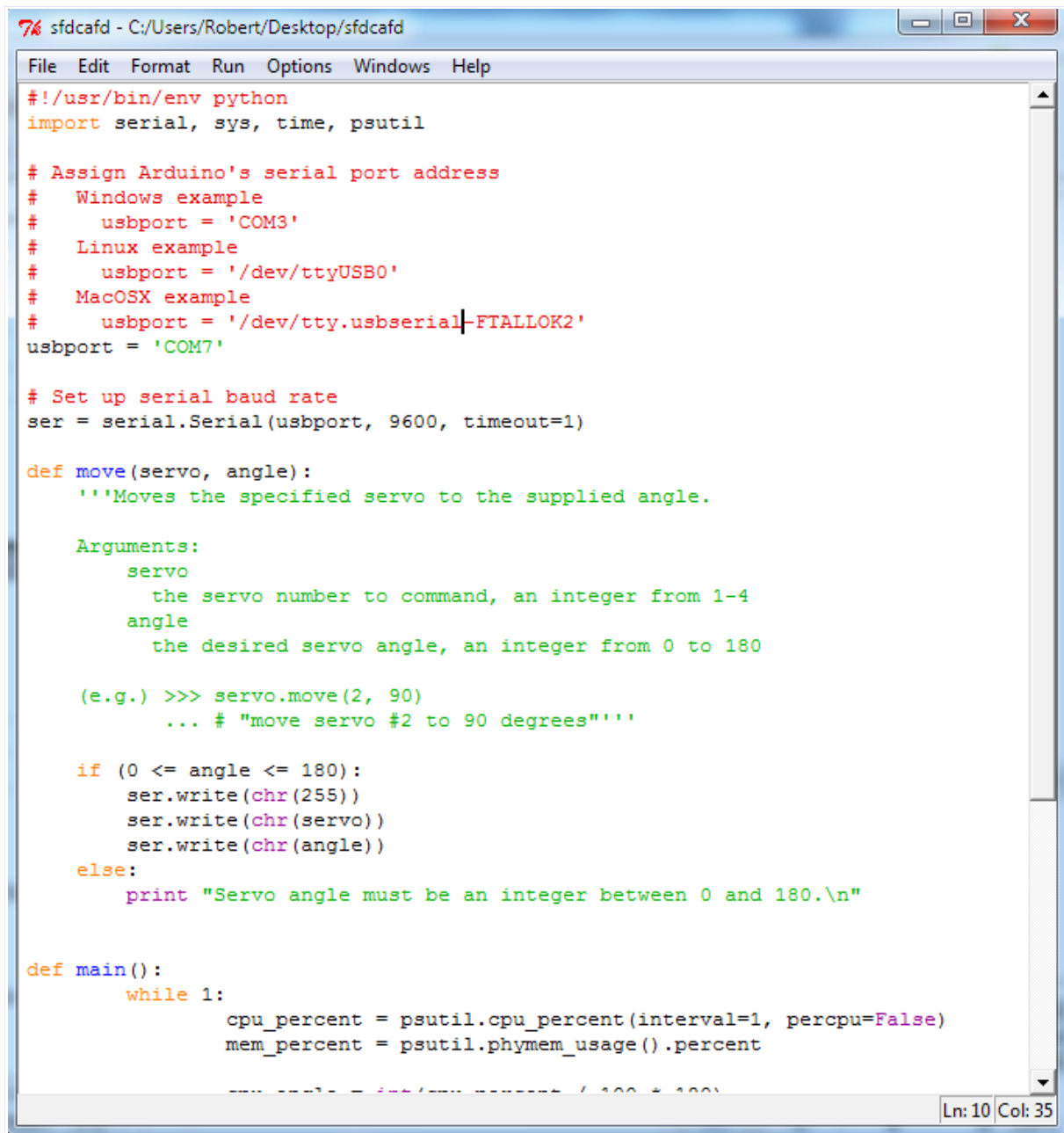
<http://www.robofun.ro/forum>

C:\Python27

- Asa arata System Path cu linia de mai sus.



- Se deschide IDLE(Python GUI) si din File – New Window se creaza fereastra in care vei copia codul sursa de mai jos.



```
#!/usr/bin/env python
import serial, sys, time, psutil

# Assign Arduino's serial port address
# Windows example
# usbport = 'COM3'
# Linux example
# usbport = '/dev/ttyUSB0'
# MacOSX example
# usbport = '/dev/tty.usbserial-FTALLOK2'
usbport = 'COM7'

# Set up serial baud rate
ser = serial.Serial(usbport, 9600, timeout=1)

def move(servo, angle):
    '''Moves the specified servo to the supplied angle.

    Arguments:
        servo
            the servo number to command, an integer from 1-4
        angle
            the desired servo angle, an integer from 0 to 180

    (e.g.) >>> servo.move(2, 90)
            ... # "move servo #2 to 90 degrees"'''

    if (0 <= angle <= 180):
        ser.write(chr(255))
        ser.write(chr(servo))
        ser.write(chr(angle))
    else:
        print "Servo angle must be an integer between 0 and 180.\n"

def main():
    while 1:
        cpu_percent = psutil.cpu_percent(interval=1, percpu=False)
        mem_percent = psutil.phymem_usage().percent
        # ... more code ...
```

- Urmeaza sa modifici o linie care reprezinta defapt portul USB la care este conectata placa Arduino.

usbport = 'COM7'

- In cazul de fata placa Arduino a fost conectata la portul COM7. Portul se poate identifica direct din mediul Arduino.
- Urmeaza sa rulezi programul din Run – Run Module.

- Din acest moment script-ul Python va transmite catre placa Arduino 3 valori dintre care 2 sunt valorile la care se vor deplasa bratele servomotoarelor. Un servomotor este responsabil cu indicarea nivelului de utilizare al procesorului iar celalalt servomotor se ocupa cu indicarea capacitatii memoriei RAM.
- In functie de sistemul de operare pe care ai instalat Python, este posibil sa primești erori cu privire la lipsa unor librării. Adica este posibil ca unul din modulele serial, sys, time si psutil sa iti lipseasca. In acest caz vei instala librăriile lipsa.


```

#!/usr/bin/env python
import serial, sys, time, psutil

# Assign Arduino's serial port address
# Windows example
#     usbport = 'COM3'
# Linux example
#     usbport = '/dev/ttyUSB0'
# MacOSX example
#     usbport = '/dev/tty.usbserial-FTALLOK2'
usbport = 'COM7'

# Set up serial baud rate
ser = serial.Serial(usbport, 9600, timeout=1)

def move(servo, angle):
    '''Moves the specified servo to the supplied angle.

    Arguments:
        servo
            the servo number to command, an integer from 1-4
        angle
            the desired servo angle, an integer from 0 to 180

    (e.g.) >>> servo.move(2, 90)
            ... # "move servo #2 to 90 degrees"'''

    if (0 <= angle <= 180):
        ser.write(chr(255))
        ser.write(chr(servo))
        ser.write(chr(angle))
    else:
        print "Servo angle must be an integer between 0 and 180.\n"

def main():
    while 1:
        cpu_percent = psutil.cpu_percent(interval=1,
        percpu=False)
        mem_percent = psutil.phymem_usage().percent

        cpu_angle = int(cpu_percent / 100 * 180)
        mem_angle = int(mem_percent / 100 * 180)

        cpu_str = "CPU: " + str(cpu_percent) + " % (" +
str(cpu_angle) + "/180)"
        mem_str = "Mem: " + str(mem_percent) + " % (" +
str(mem_angle) + "/180)"

```

```
    print cpu_str
    print mem_str

    move(0, cpu_angle)
    move(1, mem_angle)

if __name__ == "__main__":
    sys.exit(main())
```