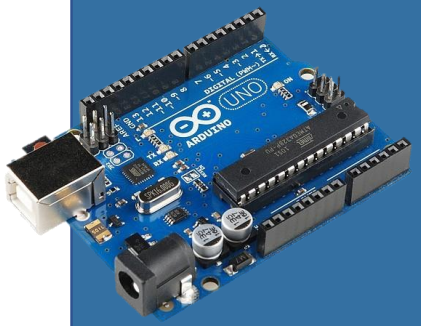
python

Aplicaciones de Escritorio con Tkinter

Turbo Código



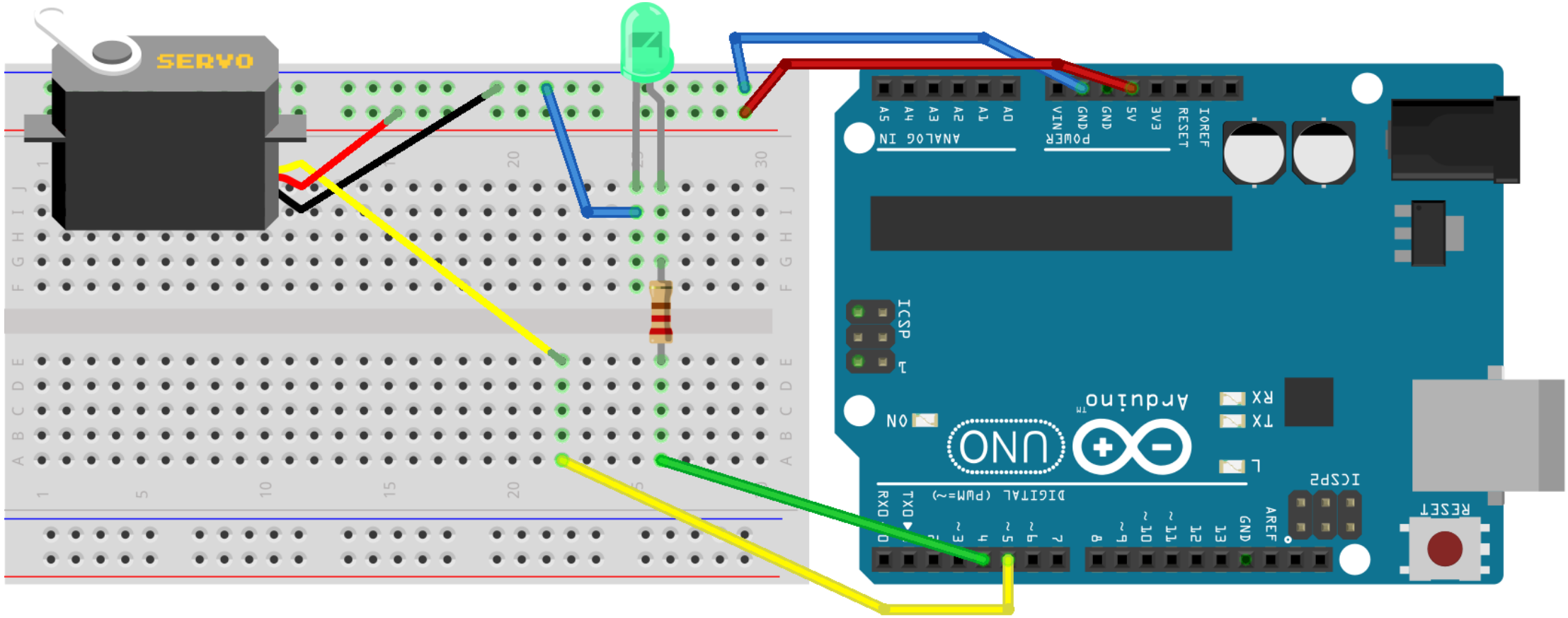
07

Comunicación Serial con Arduino

(Pyserial,Threading, Checkbutton, Scale)


Comunicación serial Python y Arduino
12
python
Turbo Código
ARDUINO
Envía Datos
De Python
a Arduino

Circuito



Turbo Código
https://www.youtube.com/user/juarezefren

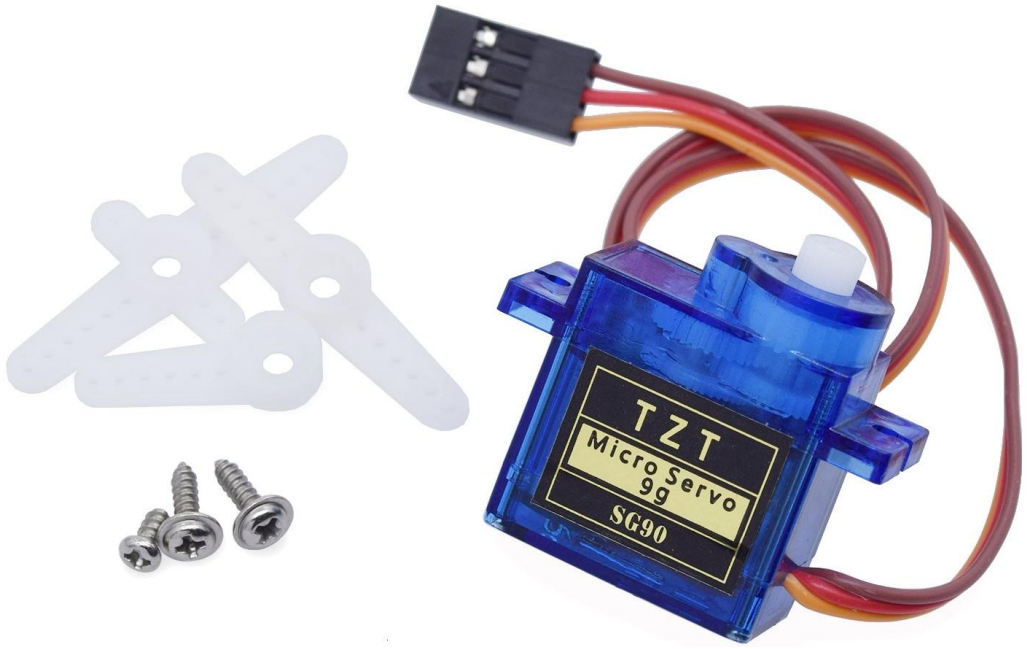
Servo SG90 1.5k

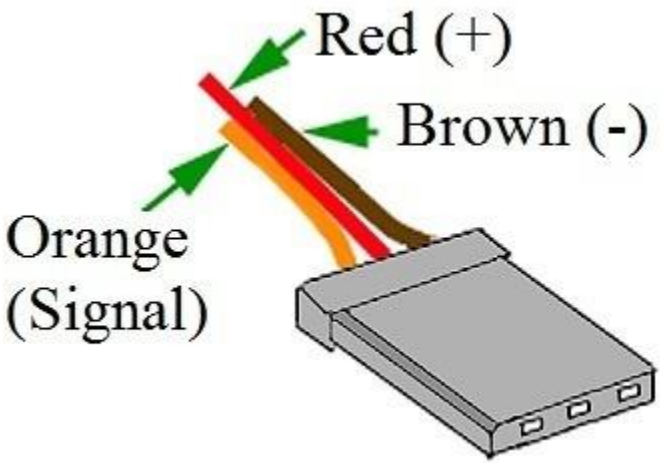
- PIN_MOT=5

- #include <Servo.h>
- Servo mot;
- mot.attach(PIN_MOT);
- mot.write(value);

- value: 0-180

- https://www.arduino.cc/en/reference/servo





Código Arduino Parte1

```
#include <Servo.h>
#define PIN_LED 4
#define PIN_MOT 5

Servo mot;
int vmot=0,vled=0,pos;
String cad,cad1,cad2;

void setup() {
  Serial.begin(9600);
  delay(30);
  pinMode(PIN_LED, OUTPUT);
  digitalWrite(PIN_LED,0);
  mot.attach(PIN_MOT);
  mot.write(0);
}
```

```
void loop() {
  if(Serial.available()){
    cad = Serial.readString();
    pos = cad.indexOf(',');
    cad1= cad.substring(0,pos);
    cad2= cad.substring(pos+1);
    if(vled != cad1.toInt()){
      vled = cad1.toInt();
      digitalWrite(PIN_LED,vled);
    }
    if(vmot != cad2.toInt()){
      vmot = cad2.toInt();
      mot.write(vmot);
    }
    Serial.println(cad1);
    Serial.println(cad2);
  }
}
```

Dividir una cadena separada por comas

- cad = 1 2 3 , 8 8

pos = 3

- pos = cad.indexOf(‘,’);

- cad1 = cad.substring(0,pos);
- cad2 = cad.substring(pos+1);

cad1 = 1 2 3

cad2 = 8 8

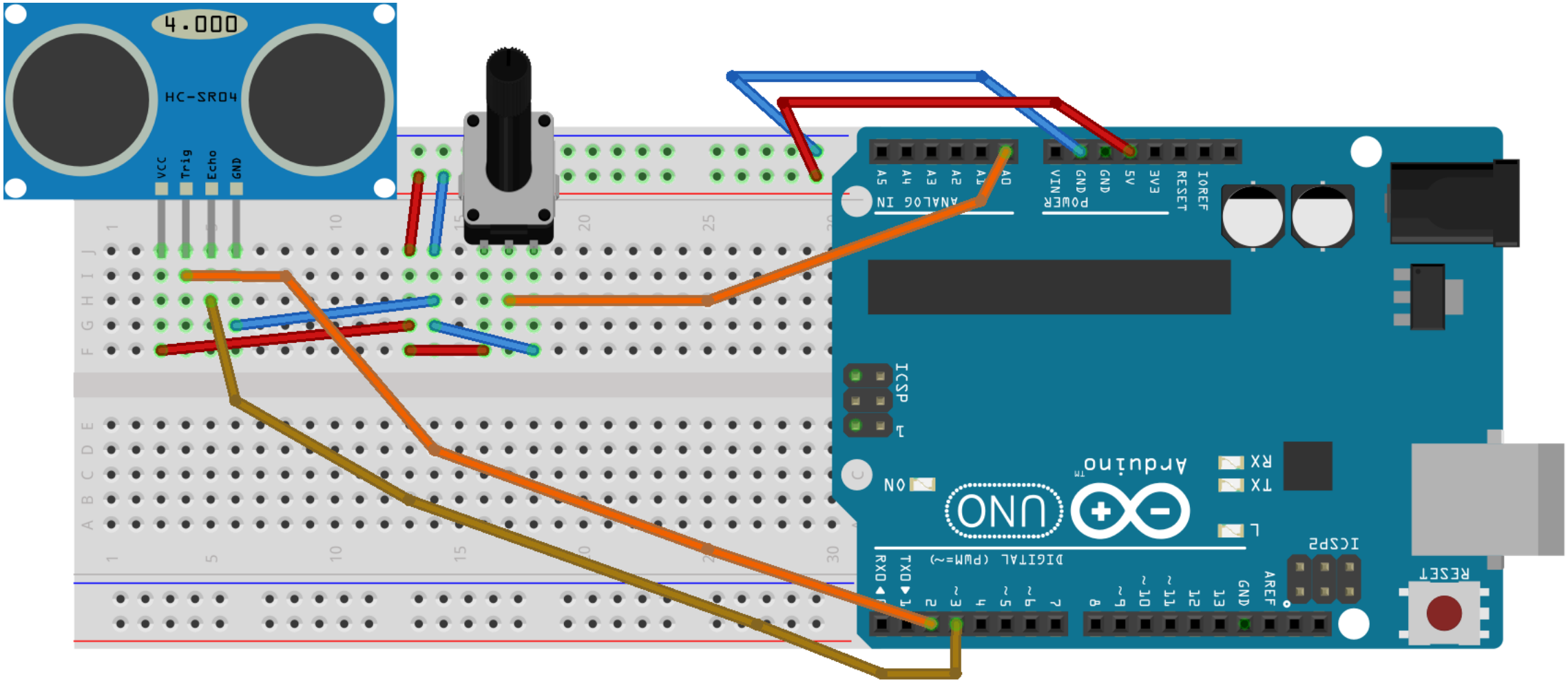
Documentación:

Clase String Arduino:
https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/

Clase Serial Arduino:
https://www.arduino.cc/reference/en/language/functions/communication/serial/

Turbo Código
https://www.youtube.com/user/juarezefren

Circuito

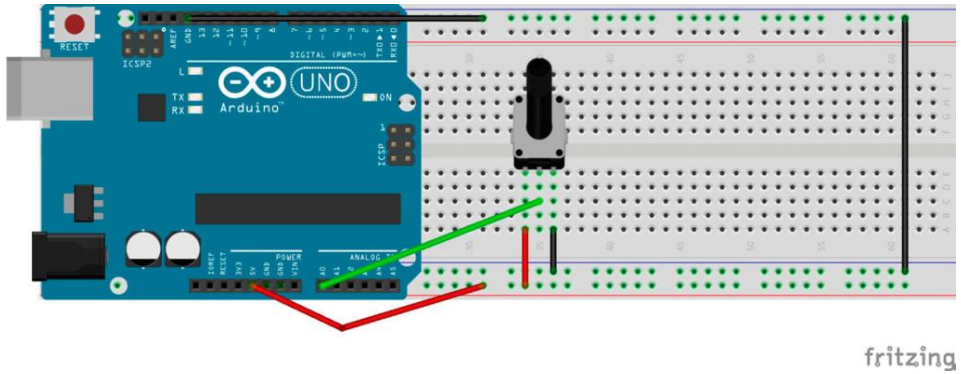


Turbo
https://www.youtube.com/user/juarezefren

fritzing

Potenciometro

- value_pot = analogRead(PIN_POT);
- value_pot = map(value_pot , 0, 1023, 0, 100);
- Serial.println(value_pot);



Ultrasonico

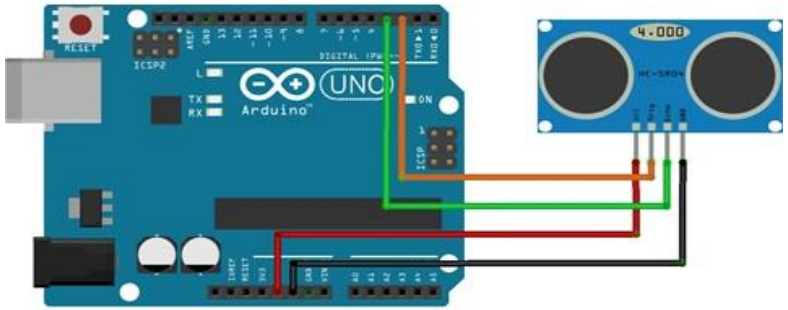
- Rango de 2 cm a 400 cm
- Mide el tiempo, en microsegundos
- Velocidad Sonido = 343 M/S

$$343\frac{m}{s} * 100\frac{cm}{m} * \frac{1}{1{,}000{,}000}\frac{s}{\mu s} = \frac{1}{29.15}\frac{cm}{\mu s}$$

$$Distancia\ (cm) = \frac{Tiempo\ (\mu s)}{29.15 * 2}$$

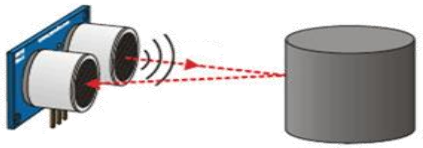
$$Distancia\ (cm) = \frac{Tiempo\ (\mu s)}{58.3}$$

```
float getDistance(int p_trig, int p_echo){
  float _val;
  digitalWrite(p_trig, LOW);  //para genera
  delayMicroseconds(4);

  digitalWrite(p_trig, HIGH);  //generamos
  delayMicroseconds(10);
  digitalWrite(p_trig, LOW);

  _val = pulseIn(p_echo, HIGH);
  _val = _val/58.3;
  if(_val >= 2 and _val <= 400)
    return _val;
  return -1;
}
```



Trig -> 2

Echo -> 3



Turbo Código
https://www.youtube.com/user/juarezefren

Algo parecido a multihilo en Arduino

Ejecuta una función cada cierto tiempo
-- se puede hacer también utilizando millis() --

- https://github.com/sstaub/Ticker

How to use

First, include the TimerObject to your project:

```
#include "Ticker.h"
```

Now, you can create a new object in setup():

```
Ticker tickerObject(callbackFunction, 1000);
tickerObject.start(); //start the ticker.
```

In your loop(), add:

```
tickerObject.update(); //it will check the Ticker
and if necessary, it will run the callback function.
```

Installation

1. "Download":https://github.com/sstaub/Ticker/archive/master.zip the Master branch from GitHub.
2. Unzip and modify the folder name to "Ticker"
3. Move the modified folder on your Library folder (On your `Libraries` folder inside Sketchbooks or Arduino software).

No usar delay() en el loop()

Código Arduino Parte 2

```
#include <Ticker.h>

#define PIN_TRIG 2
#define PIN_ECHO 3
#define PIN_POT A0

void fnDistancia(){
  float distancia;
  distancia = getDistance(PIN_TRIG,PIN_ECHO);
  Serial.println("dis:" + String(distancia,2));
}
Ticker ticDistancia( fnDistancia,1000);

int pot=-1;
void fnPotenc(){
  int value_pot;
  value_pot = analogRead(PIN_POT);
  if (pot != value_pot){
    pot = value_pot;
    Serial.println("pot:" + String(value_pot));
  }
}
Ticker ticPotenc( fnPotenc,500);
```

```
float getDistance(int p_trig, int p_echo){ //Obtiene la distancia en cm (2-400),
  float _val;
  digitalWrite(p_trig, LOW);  //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds(4);
  digitalWrite(p_trig, HIGH);  //generamos Trigger (disparo) de 10us
  delayMicroseconds(10);
  digitalWrite(p_trig, LOW);
  _val = pulseIn(p_echo, HIGH);
  _val = _val/58.3;
  if(_val >= 2 and _val <= 400)
    return _val;
  return -1;
}

void setup() {
  Serial.begin(9600);
  delay(30);
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  ticDistancia.start();
  ticPotenc.start();
}

void loop() {
  ticDistancia.update();
  ticPotenc.update();
}
```

Enviando valores con el formato “label:value”

- cad = “mot:135”

pos = 3

- pos = cad.indexOf(‘:’);

label = “mot”

- label = cad.substring(0,pos);
- value = cad.substring(pos+1);

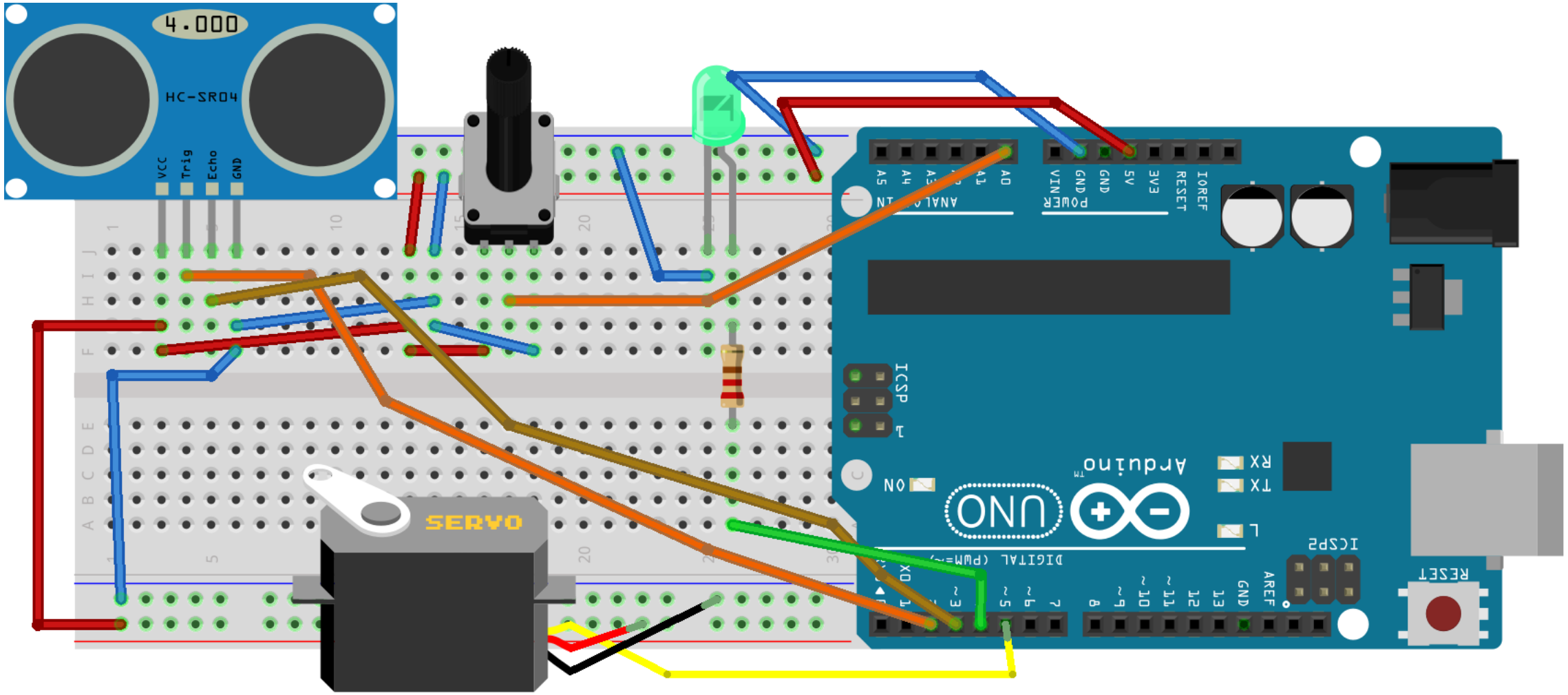
value = 135

Documentación:

Clase String Arduino:
https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/

Clase Serial Arduino:
https://www.arduino.cc/reference/en/language/functions/communication/serial/

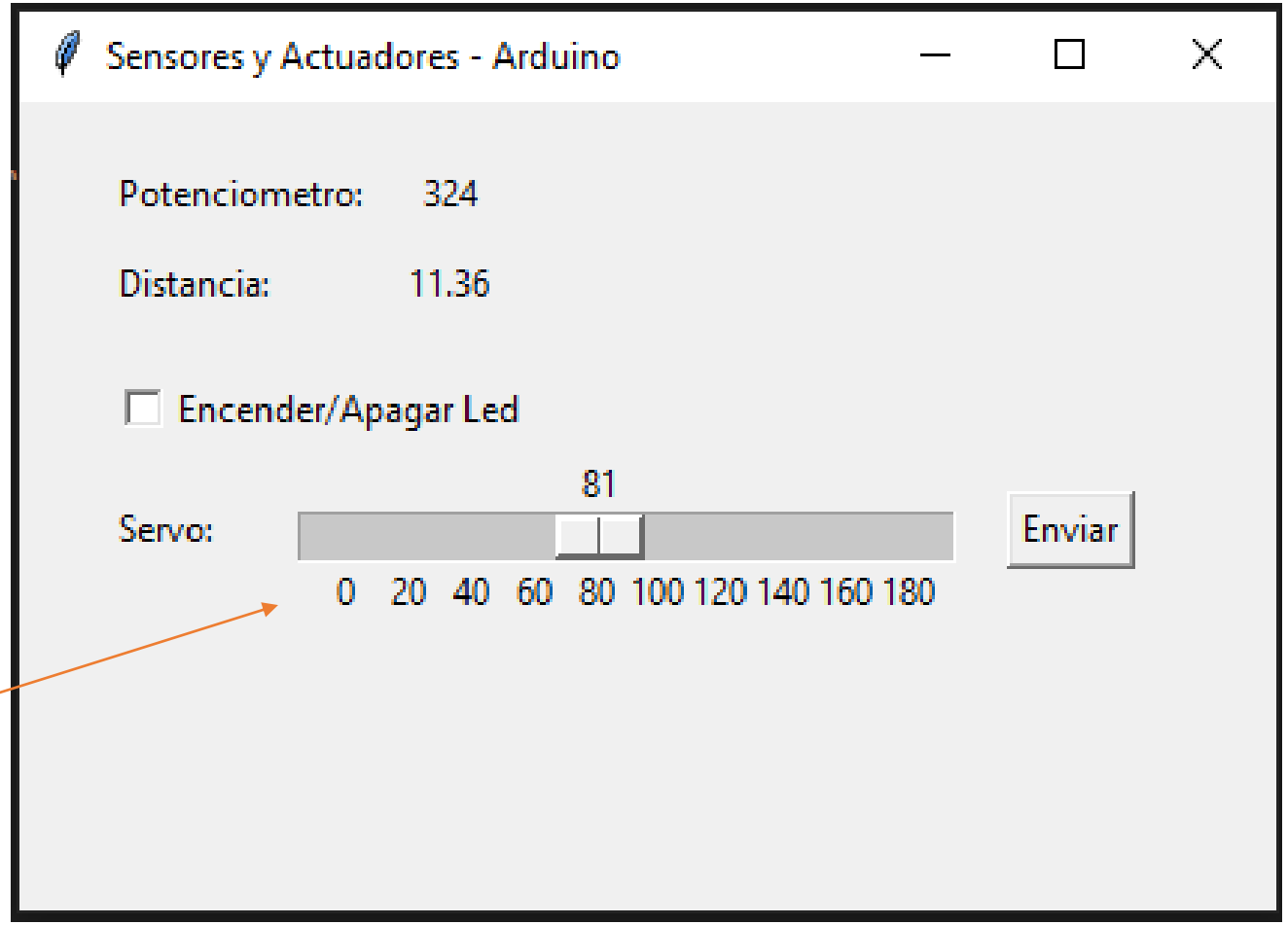
Circuito



Turb

https:

Python
(Diseño de la ventana)



Scale

Pyserial (instalación)

- Instalación
  - pip install pyserial

- Lista tus dispositivos serie:
  - python -m serial.tools.list_ports

Pyserial (uso)

- import serial
- **dev** = serial.Serial("COM4", 9600)

- **cad** = “1,180”

- **dev**.write(**cad**.encode('ascii’))

- **dev**.close()

Pyserial (lectura)

- import serial
- **dev** = serial.Serial("COM4", 9600)

- val =**dev**.readline()

- **cad** = val.decode('ascii')

- **dev**.close()