

Comunicazione Python-Arduino

Prof. Fischetti Pietro

Codice Arduino che scrive un dato sulla porta seriale:

```
wser.ino
int d=0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(d);
  d= !d; //Toggle
}
```

Codice Python per leggere dalla porta seriale tramite libreria serial

```
RSer.py
import serial
import string

PORT='COM5' #NB! Sostituire la propria porta seriale a cui e' connessa Arduino

ser = serial.Serial(port=PORT,baudrate=9600)

print ser
print("connected to: " + ser.portstr)
count=1

while True:
  r= ser.readline()
  if(len(r)>0):
    print string.strip(r)
print ser.isOpen()
ser.close()
print ser.isOpen
```

Nella figura che segue i dati ricevuti dal codice Python


```

void loop() {
  y = A*sin(2*3.14*F*t);
  Serial.println(y);
  t+=sampling_interval; //Incremento del tempo
}

```

RSerVW.py – codice python che disegna il segnale che arriva sulla seriale

```

import serial
import time
import csv
import matplotlib
matplotlib.use("tkAgg")
import matplotlib.pyplot as plt
import numpy as np

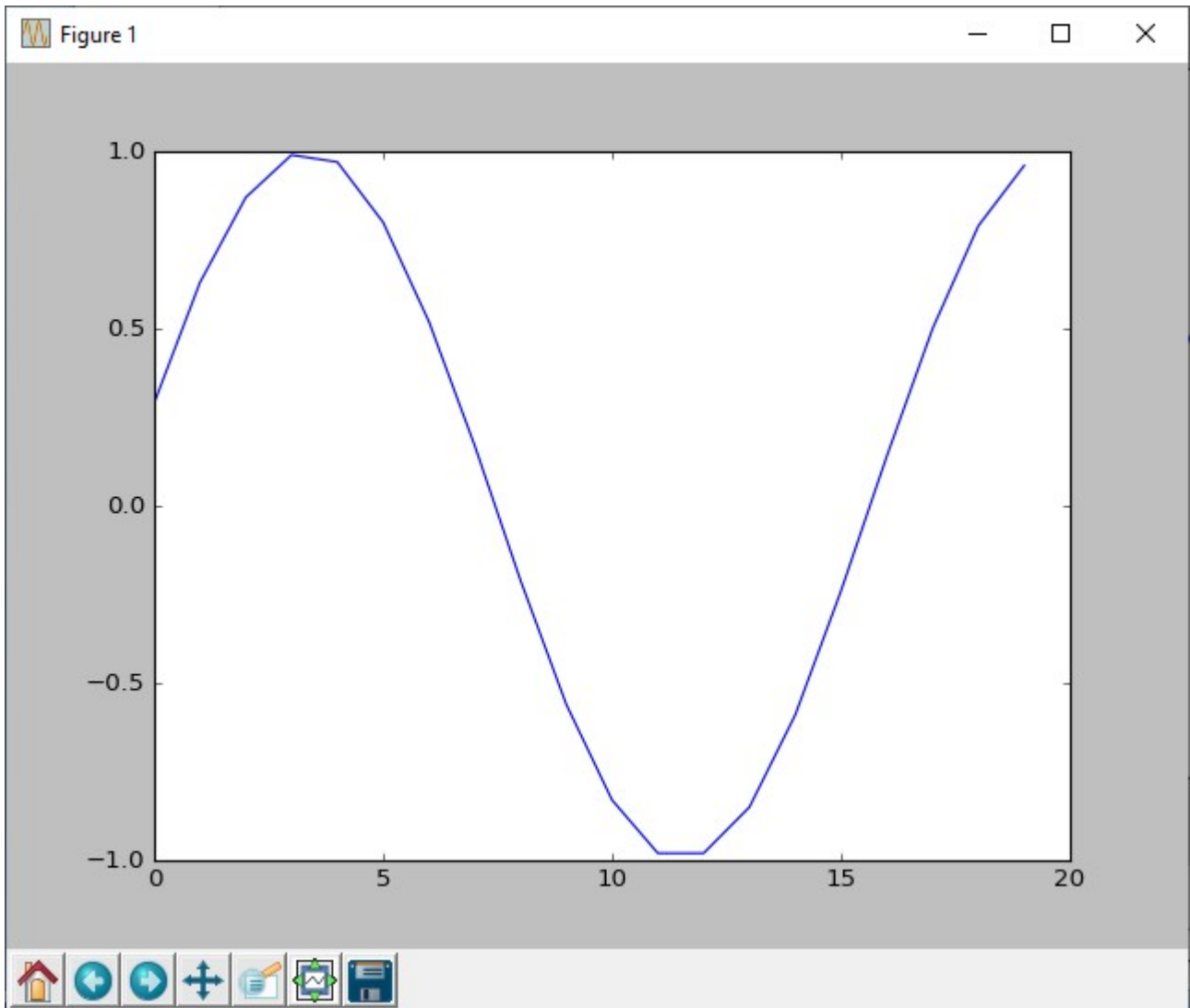
ser = serial.Serial('COM5')
ser.flushInput()

plot_window = 20
y_var = np.array(np.zeros([plot_window]))

plt.ion()
fig, ax = plt.subplots()
line, = ax.plot(y_var)

while True:
    try:
        ser_bytes = ser.readline()
        try:
            decoded_bytes = float(ser_bytes[0:len(ser_bytes)-2].decode("utf-8"))
            print(decoded_bytes)
        except:
            continue
        with open("test_data.csv","a") as f:
            writer = csv.writer(f,delimiter=",")
            writer.writerow([time.time(),decoded_bytes])
        y_var = np.append(y_var,decoded_bytes)
        y_var = y_var[1:plot_window+1]
        line.set_ydata(y_var)
        ax.relim()
        ax.autoscale_view()
        fig.canvas.draw()
        fig.canvas.flush_events()
    except:
        print("Keyboard Interrupt")
        break
ser.close()

```



WSer.py – Invia un carattere letto da tastiera

```
import serial
import string

ser = serial.Serial('COM5') # open serial port
print(ser.name)    # check which port was really used
try:
    c=raw_input()
    while(len(c)>0):
        ser.write(c)
        c=raw_input()

finally:
    ser.close()
```

Rser.ino – legge un dato da seriale e se 1 accende il Led 13 se 0 lo spegne

```
void setup() {
```

```

Serial.begin(9600);
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, LOW);
}

void loop() {
  if (Serial.available ()> 0) {
    char ch = Serial.read ();
    if (ch == '0') {
      digitalWrite(LED_BUILTIN, LOW);
    }
    else if (ch == '1') {
      digitalWrite(LED_BUILTIN, HIGH);
    }
  }
}
}

```

Wser.py Scrive su Porta seriale

```

import serial
import string

ser = serial.Serial('COM5') # open serial port
print(ser.name)      # check which port was really used
try:
  c=raw_input()
  while(len(c)>0):
    ser.write(c)
    c=raw_input()

finally:
  ser.close()

```

Rser.ino – Legge un numero da seriale e se 1 accende il led 13 altrimenti se 0 lo spegne

```

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
}

void loop() {
  if (Serial.available ()> 0) {
    char ch = Serial.read ();
    if (ch == '0') {
      digitalWrite(LED_BUILTIN, LOW);
    }
    else if (ch == '1') {
      digitalWrite(LED_BUILTIN, HIGH);
    }
  }
}
}

```

