

Proyecto 1

Proyectos TICS 1

Nicolás Flores - Christian Lopez

I. PRESENTACIÓN

En este informe se describirá el prototipo final creado, se explicará como se debe armar y como se debe utilizar. Además se mostrará un diagrama que explicará como están ensamblados los componentes del prototipo. Así como también se especificará las tecnologías utilizadas y se explicará con detalle los resultados de pruebas, el trabajo realizado y las experiencias que vivimos al realizar este proyecto.

II. DESARROLLO

A. Descripción del prototipo

El prototipo consta de 3 sensores diferentes. Uno de temperatura, presión atmosférica y de flujo. Este último fue creado con un motor de corriente continua y una hélice, la cual permite que el rotor del motor gire cuando pasa el agua a través de este sensor.

El motor está dentro de un tubo de PVC, lo cual permite canalizar el agua que se está analizando.

Los sensores de temperatura, humedad y presión atmosférica permiten determinar el clima (a grosso modo). De esta forma podremos predecir si lloverá o no.

B. Diagrama de componentes

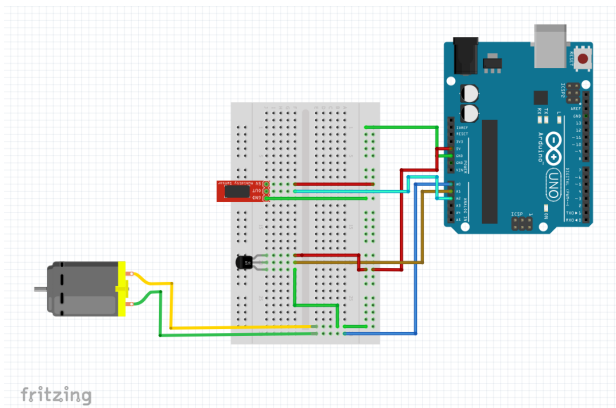


Figura 1. Gráfico Físico prototipo

C. Modo de armado

Nuestro artefacto podrá ser instalado al borde de un río que esté cerca de un pueblo o ciudad.

Para esto simplemente se debe crear una base que esté a la altura del río que será analizado.

Escuela de Informática y Telecomunicaciones
e-mail: nicolas.floress@mail.udp.cl

D. Proceso realizado

Para este artefacto creamos y calibramos un sensor de flujo con un motor CC (como se mencionó anteriormente).

Los pasos que seguimos para construir este sensor fueron los siguientes:

1. Creamos una hélice con una tapa de botella plástica

- Recortamos y moldeamos la tapa hasta que encontramos la forma ideal para que esta gire cuando choca con el agua.

2. Soldamos dos cables a las terminaciones del motor.

- Conectamos el cable negativo a un pin ground del Arduino.
- Conectamos el cable positivo a un pin analógico del Arduino.

Después de la creación del sensor se implementó (creo) un programa para recibir el voltaje que entregaba el sensor de flujo (el motor) cada 3 segundos, esta información es recibida por un servidor (notebook), la cual es analizada, graficada y guardada.

Para poder analizar la información entregada por los sensores se creó un programa con el lenguaje de programación Python3, en este programa se utilizaron diferentes librerías:

- **Serial:** permite recibir información desde el Arduino
- **Matplotlib:** permite graficar los datos guardados
- **Numpy:** permite guardar la información recibida en un archivo en forma de arreglo.

El programa recibe los datos desde el Arduino, los cuales son guardados en una lista y en un arreglo. Los datos guardados en la lista son graficados en tiempo real, cabe destacar que cada vez que se agrega un dato a la lista se elimina un dato antiguo, de esta forma el gráfico siempre tiene el mismo tamaño. Por otra parte los datos que son agregados al arreglo son guardados para que estos sean analizados y graficados posteriormente (en el futuro).

De los datos analizados se obtiene un valor promedio, el cual al ser comparado con un dato que es ingresado saber si sobrepasa el valor promedio enviar alerta.

III. CÓDIGOS DEL PROTOTIPO

- Este es el código que utiliza el arduino para recibir los datos de los sensores:

```
const int pot_f=0; const int pot_t=1;
const int pot_p=2;

void setup() { Serial.begin(9600); }

long timedPulsos( pulso_total, seconds)
{
    long startedAt = millis();
    while( millis() - startedAt < seconds *1000)
    {
        int pulso=analogRead(pot_p);
        if (pulso > 30){ pulso_total=pulso+pulso_total;
        ; }
    }
    return pulso_total;
}

long timedTemperatura( float temp_total, int seconds)
{
    long startedAt = millis();
    int count_T=0;
    while( millis() - startedAt < seconds *1000)
    {
        float temp=analogRead(pot_t);
        temp_total=temp+temp_total;
        count_T+=1;
    }
    return temp_total/count_T;
}

void loop() {
    int secs=15;
    Serial.println( timedPulsos(0,secs) );
    Serial.println( timedTemperatura(0,secs) );
}
```

- Código para graficar los datos guardados:

```
import numpy as np
import matplotlib.pyplot as plt
a = np.load('data.npy')
plt.title('Pulsos Guardados')
plt.ylabel('Pulsos recibidos')
plt.xlabel('Muestras')
plt.plot(a)
plt.show()
```

IV. PRUEBAS REALIZADAS

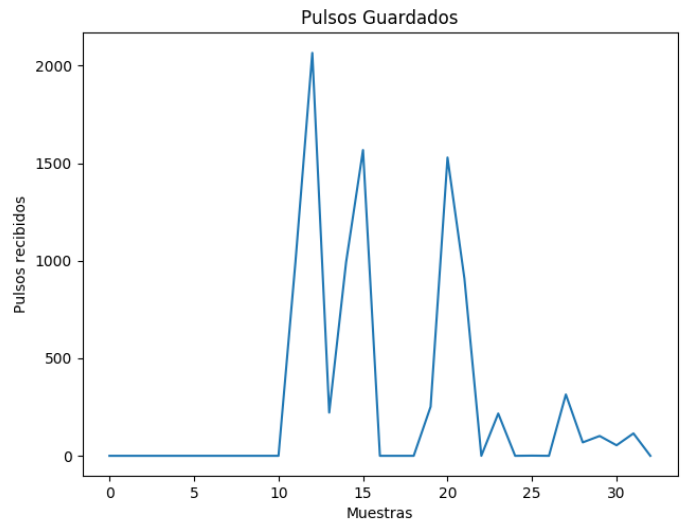


Figura 2. Gráfico de prueba

- Código para Analizar, graficar (de forma simultanea) y guardar los datos:

```
import serial import matplotlib.pyplot import
numpy
values = []
values_save=np.array([0,0])

serialArduino = serial.Serial("COM4", 9600)
def plotValues():
    plt.ylabel('Values')
    plt.plot(values, 'rx-', label='values')
    plt.legend(loc='upper right')
def doAtExit():
    serialArduino.close()
    print("Close serial ()")
    print(values_save)
    np.save('data{}'.format(af), values_save)
for i in range(0,11):
    values.append(0)
while True:
    while (serialArduino.inWaiting()==0):
        pass
    valueRead = serialArduino.readline(500)
    valueInInt = int(valueRead)
    if (valueInInt>=5 or valueInInt==0):
        print("readline ()")
    try:
        print(valueInInt)
        if valueInInt >= 5 or valueInInt==0 :
            values.append(valueInInt)
            values_save=np.insert(values_save,
                values_save.size, valueInInt)
            values.pop(0)
            drawnow(plotValues)
```

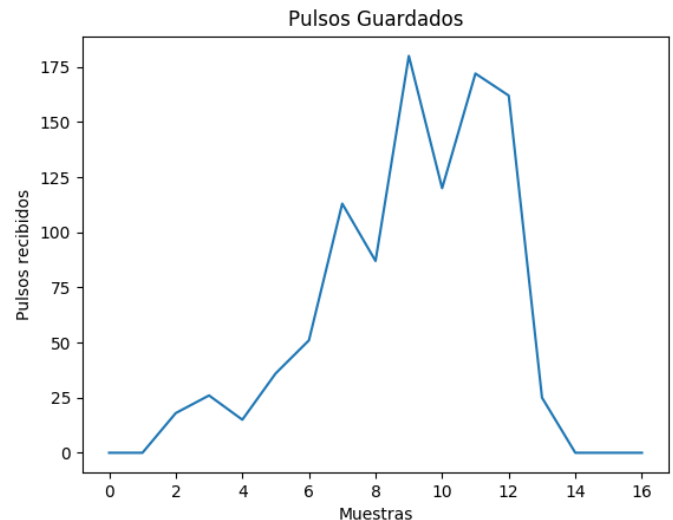


Figura 3. Gráfico de prueba

V. REPOSITORIO

<https://github.com/NigusFs/proyecto-1-tics1>