

Universidad de Sevilla  
Centro Nacional de Aceleradores  
Grupo de Física Nuclear Básica

## Sistema de control por correo

Andrés Gómez Ramos

agomez30@us.es

17 Abril 2017



## **Contribución**

El objetivo principal del proyecto es diseñar un sistema flexible que permita al usuario comunicarse con su ordenador de trabajo mediante correos permitiendo mandar y recibir información cuando el usuario y el ordenador lo necesiten.

## **Sumario**

El propósito principal es crear un sistema que permita al ordenador enviar mensajes automáticamente e interpretar los mensajes que reciba.

## Tabla de contenidos

Contribución .....	iv
Sumario .....	iv
1 Introducción .....	1
2 El problema ingenieril.....	1
3 Requerimientos .....	1
4 Posible solución .....	1
4.1 Solución .....	<b>¡Error! Marcador no definido.</b>
5 Análisis ingenieril de la solución planteada.....	1
6 Conclusión .....	4
7 Consideraciones .....	4

## **1 Introducción**

Se pretende realizar un sistema que permita informar al usuario de cambios realizados en ciertos archivos del ordenador.

## **2 El problema ingenieril**

Se debe diseñar un procedimiento para medir resistencias superiores a 100 MΩ. Se debe seleccionar un procedimiento preciso y cómodo de utilizar en la Faraday cup del acelerador.

## **3 Requerimientos**

El sistema debe informar al usuario en poco tiempo de los cambios efectuados.

## **4 Posible solución**

## **5 Análisis ingenieril de las soluciones planteadas**

### **5.1 Sistema para enviar mensajes**

Sistema para enviar mensajes. “sendmessge.py”.

```
#!/usr/bin/python2.7
```

```
1 import smtplib
```

```
2 server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
```

```
3 server.ehlo()
```

```
4 gmail_user = "
```

```
5 gmail_password = ''
6 server.login(gmail_user, gmail_password)
7 server.sendmail(gmail_user, 'contacto@gmail.com', 'hellopy')
8 server.close
```

Se debe disminuir la seguridad de la cuenta de Gmail en:

Mi cuenta -> Inicio de sesión y seguridad -> permitir el acceso de aplicaciones menos seguras: SÍ.

Este sistema permite enviar un mensaje al contacto que se encuentra en la línea 7 (pueden ser varios contactos), en este caso “contacto@gmail.com” en este caso el mensaje es “hellopy”.

Se recomienda crear un usuario Gmail expresamente para ser utilizado por el ordenador.

## 5.2 Sistema para recibir e interpretar mensajes

```
6 import imaplib
7
8 mailserver = imaplib.IMAP4_SSL('imap.gmail.com', 993)
9 gmail_user = '@gmail.com'
10 gmail_password = ''
11 mailserver.login(gmail_user, gmail_password)
12
13 status, count = mailserver.select('Inbox')
14 status, data = mailserver.fetch(count[0], '(RFC822)')
15
16 print(data)
17
18
19
20 dato = str(data)
21
22 stron = "turnon";
23
24 print (dato.find(stron))
25 aux = dato.find(stron)
26
27 if aux == -1:
28     print ('no encontrado')
29 else:
30     print ('encontrado')
31
32 (1)
33
34 mailserver.close()
35 mailserver.logout()
```

En la línea 15 se conecta al usuario Gmail. En la línea 17 se entra en la bandeja de entrada. En la línea 18 se guarda en “data” la información del último mensaje recibido. En la línea 24 se convierte la información de data en cadena de caracteres y se guarda en “dato”. A partir de la línea 26 se busca unos caracteres específicos, en este caso “turnon” en “dato”. Si no lo encuentra, muestra “-1”, si lo encuentra muestra la posición en la cadena de caracteres en la que se encuentra.

## 6 Conclusión

El sistema de envío de mensajes permite al ordenador informarnos por correo de un hecho relevante previamente programado.

El sistema de lectura permite al usuario, mandando una cadena de caracteres previamente programada para que el sistema para recibir mensajes devuelva un valor distinto a “-1”, ejecutar un comando previamente programado.

Para que el código Python se ejecute cada cierto tiempo, se recomienda utilizar un sistema de ejecución controlada como el programa “Crontab” en Linux.

## 7 Consideraciones

Es posible combinar un ordenador con una salida GPIO con una placa Arduino permitiendo el control del Arduino conectando sus salidas digitales a las salidas GPIO del ordenador.

A continuación se muestra un ejemplo de control del led de arduino mediante este sistema.

### 7.1 Combinación con Arduino

Código de Python

```
#EJEMPLO DE BLINKING CON RASPBERRY PI
```

```
#Escrito por G14r3
```

```
import RPi.GPIO as GPIO #importamos la libreria y cambiamos su nombre por "GPIO"
```

```
import time #necesario para los delays
```

```
#establecemos el sistema de numeracion que queramos, en mi caso BCM
```

```
GPIO.setmode(GPIO.BCM)
```

```
#configuramos el pin GPIO17 como una salida
```

```
GPIO.setup(17, GPIO.OUT)
```

```
#encendemos y apagamos el led 5 veces
```

```
for i in range(0,5):
```

```
    GPIO.output(17, GPIO.HIGH)
```

```
    time.sleep(1)
```

```
    GPIO.output(17, GPIO.LOW)
```

```
    time.sleep(1)
```

```
GPIO.cleanup()
```

Se conecta la salida GPIO17 a la salida digital del Arduino 8 comportándose los dos pines de la misma forma.

Código de arduino

```
// the setup function runs once when you press reset or power the board
```

```
void setup() {
```

```
    // initialize digital pin 13 as an output.
```

```
    pinMode(13, OUTPUT);
```

```
    pinMode(8, INPUT);
```

```
}
```

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
    if (8 == HIGH) {
```

```
        digitalWrite(13, HIGH);
```

```
    } else { digitalWrite(13, LOW); }
```

```
}
```

Si el pin 8 se encuentra en HIGH el led se encenderá. Si está en LOW se apagará.



Se puede diseñar un voltímetro con precisión de añadiendo un voltaje de referencia de 4,096 V mediante el regulador LM4040DIZ-4.1/NOPB (5) de Texas Instruments.

Tal y como se muestra en el enlace: <http://www.skillbank.co.uk/arduino/measure.htm>

Permitiendo crear un sistema de control de tensión por correo. Además esta misma placa de Arduino podría ser un sistema de control ya que podría conectarse a un interruptor que se abra cuando las condiciones de tensión salgan de unos límites de seguridad.

## **7.2 Uso como sistema portátil**

También es posible diseñar un sistema portátil con una raspberry y una batería externa de 10000mAh, permitiendo una autonomía de 2 a 4 días.