

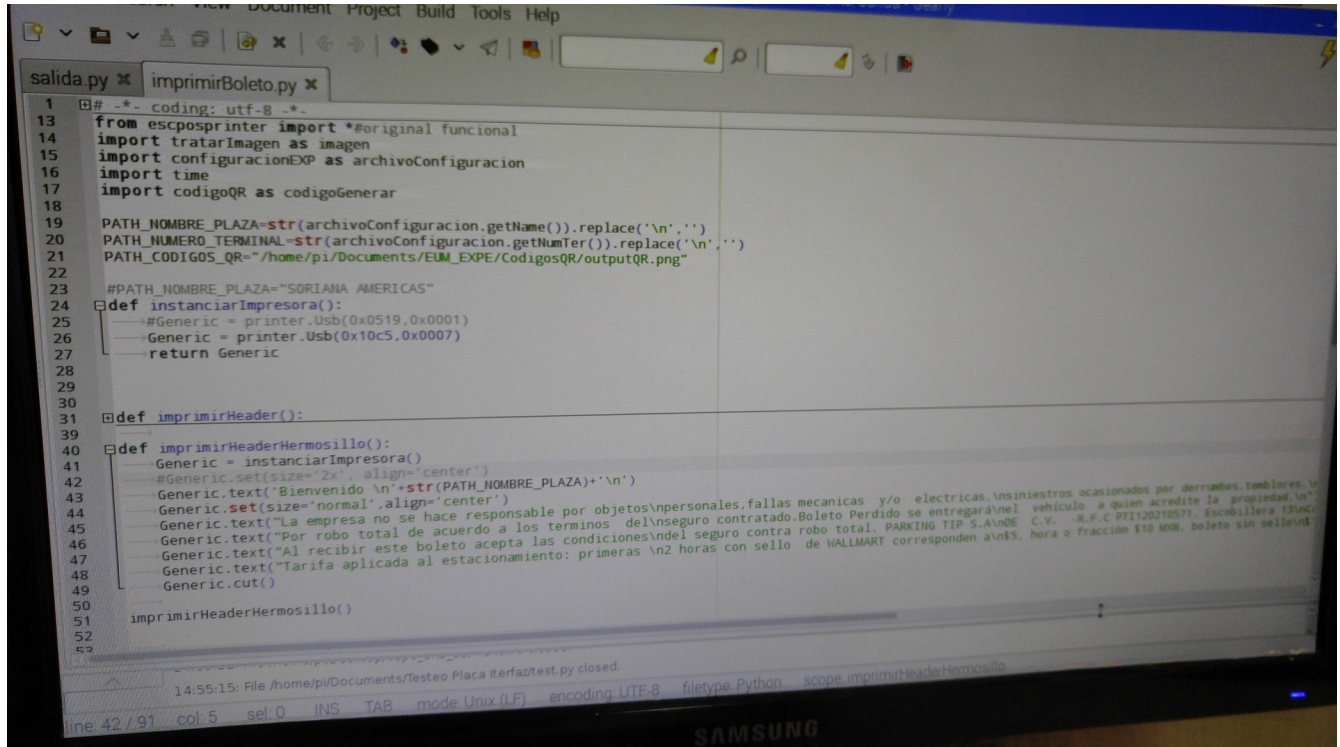
En esta imagen vemos el puerto usb de la impresora sk1-31

```
Bus 001 Device 023: ID 10c5:0007 Sanei Electric, Inc.  
Bus 001 Device 004: ID 046d:c534 Logitech, Inc.
```

En el código de python nos conectamos por ese puerto en la línea 26

Lo restante son instrucciones escpos con lo que imprimimos texto, imágenes y códigos QR (es lo que nos interesa imprimir).

Todo el código funciona con impresoras con compatibilidad escpos como el caso de la Tup900.



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53  
# coding: utf-8  
from escposprinter import * # original funcional  
import tratarImagen as imagen  
import configuracionEXP as archivoConfiguracion  
import time  
import codigoQR as codigoGenerar  
  
PATH_NOMBRE_PLAZA = str(archivoConfiguracion.getName().replace('\n', ''))  
PATH_NUMERO_TERMINAL = str(archivoConfiguracion.getNumTer().replace('\n', ''))  
PATH_CODIGO5_QR = "/home/pi/Documents/EUM_EXPE/CodigosQR/outputQR.png"  
  
# PATH_NOMBRE_PLAZA = "SORIANA AMERICAS"  
  
def instanciarImpresora():  
    # Generic = printer.Usb(0x0519, 0x0001)  
    Generic = printer.Usb(0x10c5, 0x0007)  
    return Generic  
  
def imprimirHeader():  
  
def imprimirHeaderHermosillo():  
    Generic = instanciarImpresora()  
    # Generic.set(size=24, align='center')  
    Generic.text('Bienvenido \n' + str(PATH_NOMBRE_PLAZA) + '\n')  
    Generic.set(size='normal', align='center')  
    Generic.text("La empresa no se hace responsable por objetos \n personales, fallas mecanicas y/o electricas \n sinistros ocasionados por derrumbes, temblores, v  
    Generic.text("Por robo total de acuerdo a los terminos del \n seguro contratado. Boleto Perdido se entregará el vehículo a quien acredite la propiedad \n")  
    Generic.text("Al recibir este boleto acepta las condiciones \n del seguro contra robo total. PARKING TIP $ 5.00 C.V. - R.F.C PTI120210571, Escobillera (B) MC  
    Generic.text("Tarifa aplicada al estacionamiento: primeras \n 2 horas con sello de WALMART corresponden a \n $5. hora o fracción $10 MNR. boleto sin sello \n")  
    Generic.cut()  
  
imprimirHeaderHermosillo()  
  
14:55:15: File /home/pi/Documents/Testeo Placa Interfaz/test.py closed.  
line: 42 / 91 col: 5 sel: 0 INS TAB mode: Unix (LF) encoding: UTF-8 filetype: Python scope: imprimirHeaderHermosillo
```

Con la SK1-31 tenemos estos resultados.

```
pi@raspberrypi:~/Documents/Salida$
pi@raspberrypi:~/Documents/Salida$
pi@raspberrypi:~/Documents/Salida$ sudo python imprimirBoleto.py
Traceback (most recent call last):
  File "imprimirBoleto.py", line 51, in <module>
    imprimirHeaderHermosillo()
  File "imprimirBoleto.py", line 43, in imprimirHeaderHermosillo
    Generic.text('Bienvenido \n'+str(PATH_NOMBRE_PLAZA)+'\n')
  File "/usr/local/lib/python2.7/dist-packages/escposprinter/escpos.py", line 276, in text
    self._raw(text)
  File "/usr/local/lib/python2.7/dist-packages/escposprinter/printer.py", line 58, in _raw
    self.device.write(self.out_ep, msg, self.interface)
  File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 940, in write
    intf, ep = self.ctx.setup_request(self, endpoint)
  File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 102, in wrapper
    return f(self, *args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 215, in setup_request
    intf, ep = self.get_interface_and_endpoint(device, endpoint_address)
  File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 102, in wrapper
    return f(self, *args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 231, in get_interface_and_endpoint
    raise ValueError('Invalid endpoint address ' + hex(endpoint_address))
ValueError: Invalid endpoint address 0x1
pi@raspberrypi:~/Documents/Salida$
```

SAMSUNG