

Desarrollo de un Turnero Digital Usando FPGA's Laboratorio de Electrónica Digital Módulo: 4, Aplicación de Escritorio

Héctor F. JIMÉNEZ S. Sebastian ZAPATA
hfjimenez@utp.edu.co sebastzapata93@gmail.com
PGP KEY ID: 0xB05AD7B8 PGP KEY ID: 0xfffff

Francisco GALLEGO
juanfran16@utp.edu.co

Fecha de Entrega: 16 noviembre, 2016
Profesor: Ing.Msc(c) Ramiro Andres Barrios Valencia

1 OBJETIVOS

- Desarrollar el aplicativo de referencia Turno-Puesto de atención.

2 MARCO TEÓRICO

En este modulo desarrollamos el aplicativo utilizando como lenguaje de programación *python* que posee soporte para el manejo de comunicaciones utilizando el protocolo rs232, la librería estándar se llama ***pyserial***, específicamente para nuestro desarrollado utilizamos la siguientes opciones:

1. 8 bits **serial.EIGHTBITS**
2. 2 bits de parada **serial.STOPBITS_TWO**
3. paridad impar **serial.PARITY_ODD**
4. velocidad de **9600** baudios.

para nosotros simular el envío de comunicación serial lo realizamos en dos sistemas operativos para windows utilizamos un emulador de puertos *Virtual Serial*

Port en modo pair que nos permite crear virtualmente puertos/dispositivo que utilizara el protocolo RS232. En Gnu-Linux utilizamos **socat**, un emulador multipropuesta que permite crear dispositivos de comunicaciones virtuales unidireccionales. la forma de crear dos dispositivos en Linux es :

```
socat -d -d pty,raw,echo=0 pty,raw,echo=0
```

Listing 1: Creacion de dos dispositivos ubicados en **/dev/pts/2**, **/dev/pts/3**.

En este sistema se debe agregar las siguientes librerias a pyserial para poder abrir y acceder al puerto serial : **rtscts=True,dsrdtr=True**

La idea principal de este modulo era simular como seria la integracion entre el hardware y el aplicativo que se desarrollo, nosotros el aplicativo lo dividimos principalmente en 3 modulos, escritos en 3 scripts diferentes de python que son :

1. **Generador de Turnos**: Es un script en python que se encarga de generar los turnos solicitados por el usuario
2. **ModuloA,B**: Es es el aplicativo al que tienen acceso los asesores o mesas que resuelve el turno.
3. **Visualizador de Turnos**(*Turnero*): Es el que se encarga de mostrar el turno actual en pantalla para el publico.

Para la interfaz gráfica se utilizo las librerias que dispone python como **pygame** el cual nos permite de una manera simple y rápida crear ventanas y entornos.

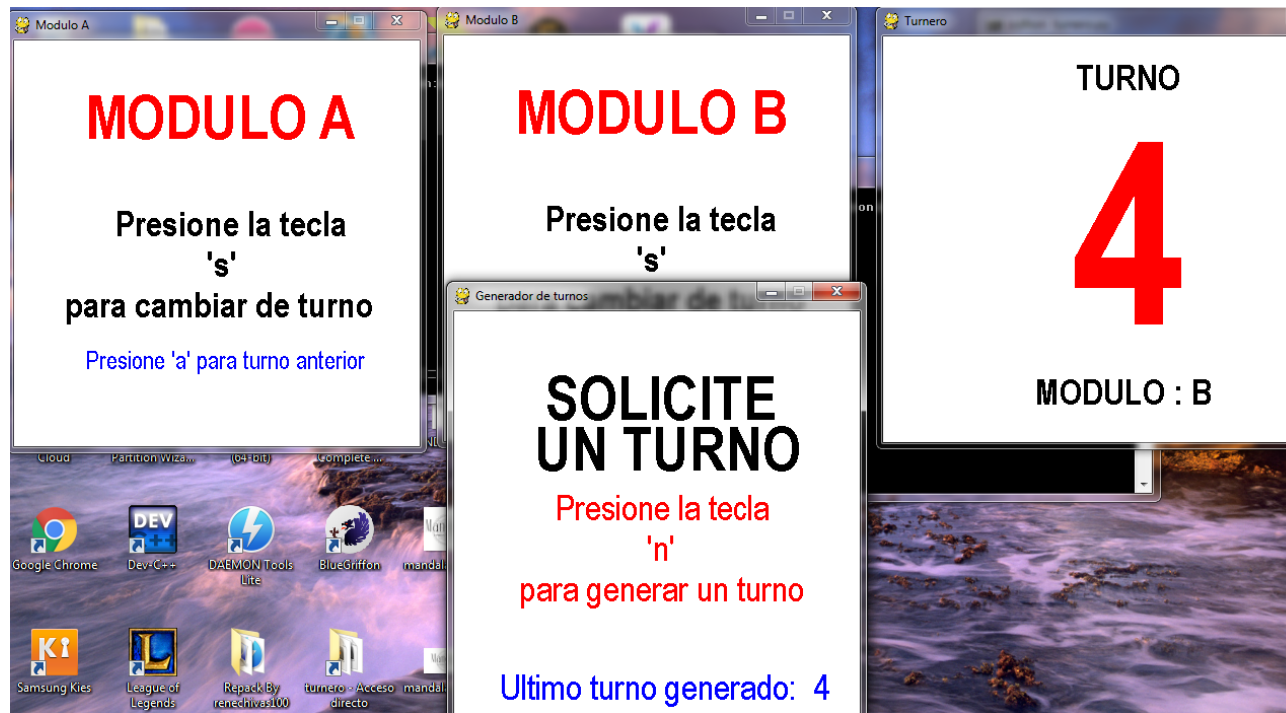


Figure 1: Esquema de Conexión Aplicación Web.

```
# -*- coding: utf-8 -*-
import pygame                                #Importar libreria necesaria
import serial
ANCHO=400                                    #Dimensiones de la ventana
ALTO=400
BLANCO=(255,255,255) #constante de colores
ROJO=(255,0,0)
VERDE=(0,255,255)
AZUL=(0,0,255)
NEGRO=(0,0,0)

pygame.init()                                #objeto que inicializa el entorno
pantalla=pygame.display.set_mode([ANCHO,ALTO]) #creamos ventana
pygame.display.set_caption("Generador De Turnos")
```

Listing 2: Creación de ventanas en todos los scripts.

Teniendo en cuenta esto para cada script lo que nosotros hemos realizado en nuestra aplicación simulada es que el usuario una vez presione la tecla **n** uti-

lizamos la opción de eventos que ofrece pygame y enviamos esto por serial a nuestro visualizador.

```
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        fin=True
    elif event.type == pygame.KEYUP:
        if event.key == pygame.K_n:
            if activado==0:
                #serial
                x=s.write(turn)
                print ("Envio : -- nuevo turno ")
                #pygame
                activado=1
                turnos+=1
            if turnos==100:
                turnos=1
```

Listing 3: Gestión de eventos, presión de teclas .

Una vez se detecta el evento de la tecla enviamos esto el numero d eturno generado.

3 ANEXOS

los demás archivos son adicionales que resuelven la misma implementación, se encuentran en el repositorio¹ *Nota:* Las referencias utilizadas se encuentran en los pies de página. Si requiere de manera detallada estas contacte con *miembros del equipo*.

1. <https://github.com/heticor915/DigitalElectronicsLab/tree/master/Reports/Module4>