

FICHA [01]: Electrónica Digital & Raspberry Pi

Práctica : **Control de 3 diodos LED**

Nivel de dificultad teoría: **básico**

Nivel de dificultad hardware: **básico**

Nivel de dificultad software: **básico**

Nota : precisa el KIT básico de prácticas de [Webtronika](#) (Ref. [KIT001](#))

DESCRIPCIÓN DE LA PRÁCTICA

Conexión de 3 diodos LED al puerwto GPIO de la tarjeta Raspberry Pi para operarlos vía software en diversos modos. Cada diodo se activa desde una patilla del puerto GPIO configurada como salida.

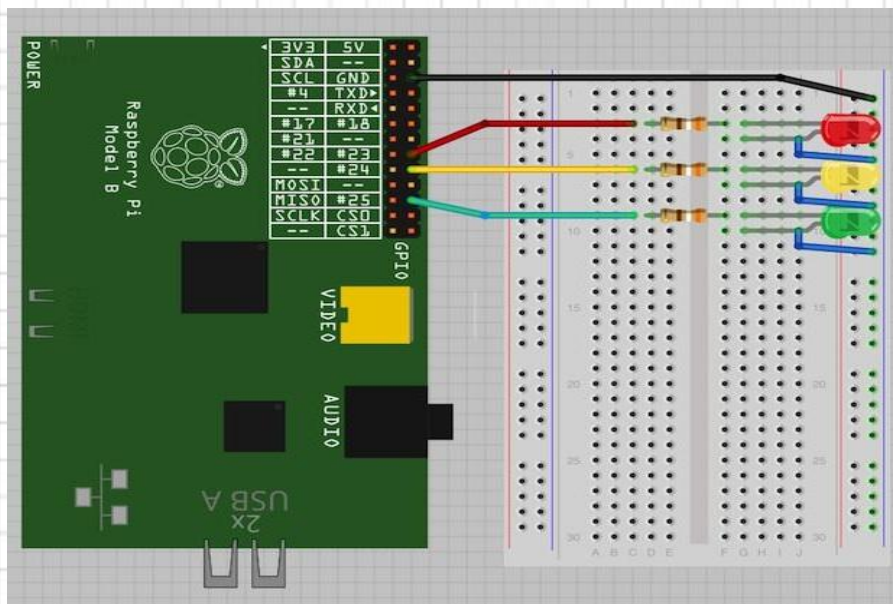
El circuito es muy simple, un diodo LED (¡ojo! tiene polaridad) y una resistencia limitadora en serie.

La intensidad consumida por pin de salida es inferior a los 10 mA. Recomendados para la Raspberry, realmente – al usar una resistencia de 390 ohmios – la intensidad que circula por el circuito no llega a los 4 mA.

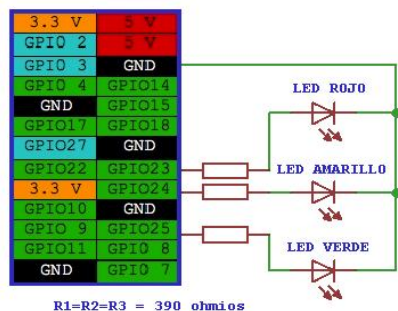
MATERIALES UTILIZADOS

- 1 Tarjeta Raspberry Pi (con s.o. Linux Wheezy instalado)
- 1 Placa prototipo pequeña
- 3 Resistencia de 390 ohmios (1/4 W)
- 1 didodo LED rojo
- 1 didodo LED amarillo
- 1 didodo LED verde
- Cables para prácticas (macho-hembra + macho-macho)

ESQUEMA ELECTRONICO

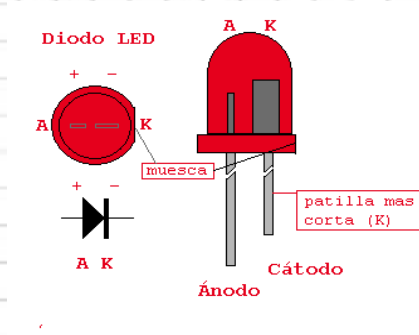


Esquema Fritzing + librerías Adafruit



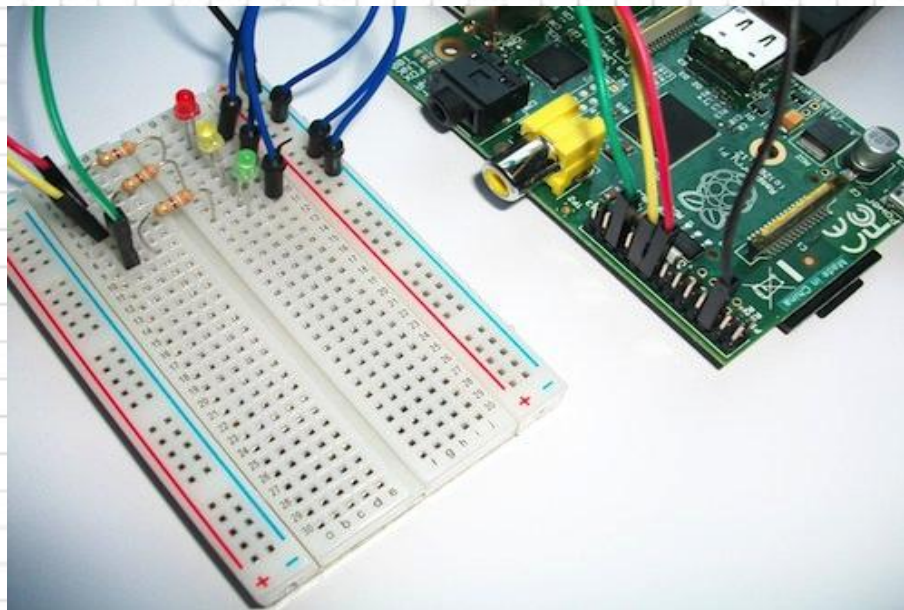
OPERATIVA DEL MONTAJE

- Apagamos la tarjeta Raspberry Pi (`sudo halt`)
- Conectamos a la placa de prototipos los componentes empleados de acuerdo al esquema mostrado.
- Prestamos **especial atención** a la polaridad de los diodos LED.
-



Detalle de conexionado del diodo LED

- Conectamos los 4 hilos desde la placa de prototipos al puerto GPIO de la Raspberry Pi.
- Nos aseguramos de que el cableado es el correcto
- Arrancamos la Raspberry Pi.
- Ejecutamos el software de control del montaje.



Esquema del montaje en la placa de prototipos

OPERATIVA SOFTWARE

Las librerías (python) para el control del puerto GPIO vienen ya pre-instaladas en el sistema operativo Raspbian. Conviene, no obstante, estar actualizado a la última versión (Wheezy).

PROGRAMA EJEMPLO (py_leds.py desarrollado en lenguaje python.)

Este programa puede copiarse tal como está y guardarse en un fichero llamado **py_leds.py** en nuestra Raspberry Pi. Para ponerlo en marcha hemos de seguir cuidadosamente estos pasos:

- (1) Tener el sistema operativo Raspbian Wheezy instalado en nuestra Raspberry Pi.
- (2) Efectuar correctamente el conexionado del circuito a nuestra Raspberry Pi.
- (3) Vamos al directorio del programa y damos al fichero **py_leds.py** permisos de ejecución con

```
sudo chmod 777 py_leds.py
```

- (5) En un terminal, desde el directorio del programa, ejecutamos (**IMPORTANTE** : como root)

```
sudo python py_leds.py
```

Listado del programa py_leds.py

```
# =====
# - Programa para manejo de diodos LED -
# =====
# - J.C.G.P. - (c) Webtronika 2013 -
# =====
#!/usr/bin/python
import sys
import curses
import time
import RPi.GPIO as GPIO

# -----
# - Rutina de configuracion de puertos en GPIO -----
# -----
def configGPIO(param1):
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(23, GPIO.OUT)
    GPIO.setup(24, GPIO.OUT)
    GPIO.setup(25, GPIO.OUT)
    return 0

# -----
# - Rutina activa LED -----
# Activa LED segun parametro (1=Rojo ; 2=Amarillo ; 3=Verde)
# param1 = 0 ==> se activan los tres diodos LED
# -----
def LEDon(param1):
    if param1 == 1:
        GPIO.output(23, True)
    elif param1 == 2:
        GPIO.output(24, True)
    elif param1 == 3:
        GPIO.output(25, True)
    else:
        GPIO.output(23, True)
        GPIO.output(24, True)
        GPIO.output(25, True)
    return 0

# -----
# - Rutina desactiva LED -----
# Desactiva LED segun parametro (1=Rojo ; 2=Amarillo ; 3=Verde)
# param1 = 0 ==> se desactivan los tres diodos LED
# -----
def LEDoff(param1):
    if param1 == 1:
        GPIO.output(23, False)
    elif param1 == 2:
        GPIO.output(24, False)
    elif param1 == 3:
        GPIO.output(25, False)
    else:
        GPIO.output(23, False)
        GPIO.output(24, False)
        GPIO.output(25, False)
    return 0

# -----
# - Rutina secuencia LED -----
# Activa 3 LEDs secuencialmente con retardo = param1
# -----
def LEDsec(param1):
    LEDoff(0)
    for i in range(3):
        LEDon(i+1)
        time.sleep(param1)
    for i in range(3):
        LEDoff(i+1)
        time.sleep(param1)
    return 0
```

```

# -----
# #####
# - PROGRAMA PRINCIPAL -----
#
# Utiliza curses para visualizar salida a pantalla
#
# #####
# -----
stdscr = curses.initscr()
curses.cbreak()
curses.start_color()
curses.init_pair(1, curses.COLOR_WHITE, curses.COLOR_BLUE)
curses.init_pair(2, curses.COLOR_YELLOW, curses.COLOR_BLUE)
hsize = curses.COLS
vsize = curses.LINES
curses.curs_set(0)
curses.noecho
stdscr.border(0)
stdscr.keypad(1)
stdscr.nodelay(1)
configPIO(0)
LEDoff(0)
try:
    stdscr.bkgd(curses.color_pair(2))
    LEDsec(0.05) # Parpadeo (TEST INICIAL) secuencial de los 3 LEDs
    while True: # Rutina principal del programa
        char = stdscr.getch()
        if (char == 81 or char == 113): # Tecla Q/q
            break
        if (char == 76 or char == 108): # Tecla L/l
            LEDsec(0.05)
        if (char == 49): # Tecla 1
            LEDon(1)
        if (char == 50): # Tecla 2
            LEDoff(1)
        if (char == 51): # Tecla 3
            LEDon(2)
        if (char == 52): # Tecla 4
            LEDoff(2)
        if (char == 53): # Tecla 5
            LEDon(3)
        if (char == 54): # Tecla 6
            LEDoff(3)
        else:
            stdscr.addstr(vsize/6, (hsize/2)-18, "Test de Diodos LED", curses.color_pair(1))
            stdscr.addstr((vsize/5)+2, (hsize/2)- 22, "[L] - secuencia de colores")
            stdscr.addstr((vsize/5)+4, (hsize/2)- 22, "[1]/[2] - activa/desactiva LED rojo")
            stdscr.addstr((vsize/5)+6, (hsize/2)- 22, "[3]/[4] - activa/desactiva LED rojo")
            stdscr.addstr((vsize/5)+8, (hsize/2)- 22, "[5]/[6] - activa/desactiva LED rojo")
            stdscr.addstr(vsize-4, (hsize/2)- 22, "Pulsa [Q] para salir")
            stdscr.refresh()
    finally:
        curses.nocbreak()
        stdscr.keypad(0)
        curses.echo()
        curses.endwin()

```

ENLACES DE INTERÉS

[RASPERRY PI](#): página oficial del proyecto Raspberry Pi.

[DIVERTEKA](#): notas técnicas de utilidad que pueden usarse con el Kit de Webtronika.

[FRITZING](#): programa gratuito para representación de prototipos desarrollado en la Universidad de Potsdam.

© Copyrights

Raspberry Pi es una marca registrada de la **Raspberry Pi Foundation**.

Fritzing es un programa para la representación de prototipos.

Nota : Webtronika no se responsabiliza de cualquier daño en los componentes empleados en la práctica. Es responsabilidad del usuario final la cuidadosa verificación y la toma de precauciones adecuadas para evitar posibles daños a los mismos.