

FICHA [10]: Electrónica Digital & Raspberry Pi

Práctica : **Medida de Temperatura**

Nivel de dificultad teoría: **básico**

Nivel de dificultad hardware: **básico**

Nivel de dificultad software: **medio**

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

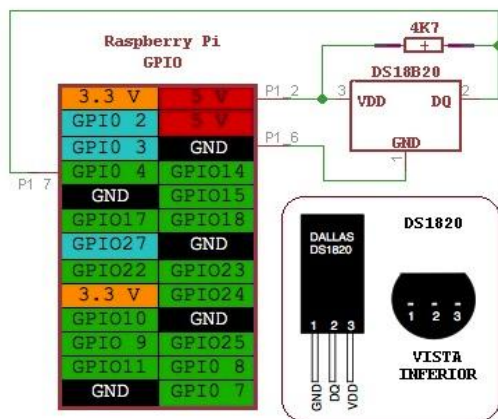
DESCRIPCIÓN DE LA PRÁCTICA

Conexión de un sensor de temperatura que opera en tecnología 1-Wire a la tarjeta Raspberry Pi. Mediante un software adecuado procederemos a medir la temperatura.

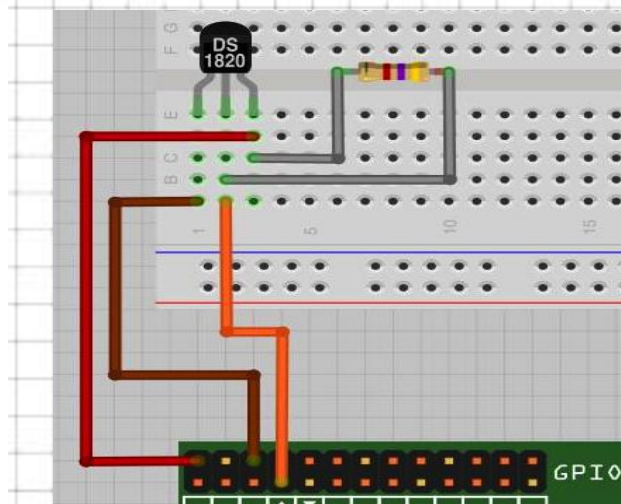
MATERIALES UTILIZADOS

- 1 Tarjeta Raspberry Pi (con s.o. Linux Wheezy instalado)
- 1 Placa prototipo pequeña
- 1 Resistencia de 4K7 ohmios (1/4 W)
- 3 Cables para prácticas (macho-hembra)
- 1 Sensor DS18B20

ESQUEMA ELECTRONICO



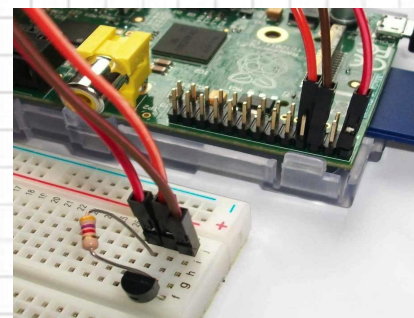
Esquema Eagle



Esquema Fritzing + librerías Adafruit

DETALLE DE MONTAJE

- Apagamos la tarjeta Raspberry Pi (`sudo halt`)
- Conectamos a la placa de prototipos la resistencia (pull-up) de datos (4K7) y el sensor 18B20 de acuerdo al esquema mostrado.
- Conectamos los 3 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 sensor.



OPERATIVA DEL MONTAJE

La patilla GPIO-4 del puerto **GPIO** (pin 7) de la tarjeta Raspberry Pi está preparada ya para utilizarse como entrada de tipo 1-Wire. El software de la misma precisa la carga de unos módulos que ya existen en el núcleo (kernel) del sistema operativo Linux, los cuales no se activan por defecto cuando esta arranca.

El protocolo **1-Wire** es un diseño de la firma **Dallas Semiconductor**, que hoy forma parte de **Maxim Integrated Products, Inc.**

OPERATIVA SOFTWARE

El software preciso para manejar el sensor de temperatura parte de la carga de los módulos que precisa la comunicación 1-Wire y, después, del lenguaje elegido para su implementación. El proceso puede efectuarse desde el propio terminal Linux sin programa alguno. Esto es lo que hay que hacer :

Accedemos al terminal y tecleamos (con privilegios de root)

```
sudo modprobe wire
sudo modprobe w1-gpio
sudo modprobe w1-therm
```

Ahora, - si el sensor DS1820 está correctamente conectado - se habrá creado una carpeta con el ID del dispositivo sensor en `/sys/bus/w1/devices/`

IMPORTANTE : cada dispositivo DS1820 tiene un ID diferente, por lo que los pasos siguientes han de ejecutarse con el ID de nuestro sensor. Saber el ID de nuestro sensor es imprescindible para desarrollar cualquier aplicación de lectura para el mismo.

Por ejemplo (con el sensor usado por nosotros) si ejecutamos

```
ls /sys/bus/w1/devices/
```

obtendremos

```
10-0008027f3664 w1_bus_master1
```

por lo que el ID de nuestro sensor DS1820 sería **10-0008027f3664**

Ejecutamos ahora el comando “more” para visualizar la lectura del sensor.
(nota : usando el ID es del sensor de Webtronika, deberemos usar el ID de nuestro sensor)

```
more /sys/bus/w1/devices/10-0008027f3664/w1_slave
```

```
28 00 4b 46 ff ff 0d 10 57 : crc=57 YES
28 00 4b 46 ff ff 0d 10 57 t=19937
```

vemos que la lectura (en milésimas de grado) es 19937 por lo que sabemos que nuestro sensor está midiendo **19,9 °C**

PROGRAMA EJEMPLO (lenguaje Python)

Programa ejemplo **mide.py** desarrollado en lenguaje python. Véanse notas más abajo.

```
# =====
# - Programa para lectura de sensor 1820 - Webtronika 2013 -
# =====

#!/usr/bin/python
import sys
import os
import curses
import time

# - Carga de modulos para OneWire -----
os.system("sudo modprobe wire")
os.system("sudo modprobe wl-gpio")
os.system("sudo modprobe wl-therm")

# - Rutina de lectura del sensor -----
# El valor de temperatura se devuelve por defecto con 3 decimales
# puede devolver 1 o 2 posiciones decimales ejecutando:
# python mide.py 1 // python mide.py 2
# -----
def dametemp(param1):
    tfile = open("/sys/bus/w1/devices/10-0008027f3664/w1_slave")
    lectura = tfile.read()
    tfile.close()
    linea2 = lectura.split("\n")[1]
    temperaturaleida = linea2.split(" ")[9]
    temperatura = float(temperaturaleida[2:])
    temperatura = temperatura / 1000
    rtemp = round (temperatura,ndec)
    return rtemp

# - Detecta el argumento -----
# si no existe, => ndec=3 (valor por defecto para decimales)
# -----
narg = len(sys.argv)
if narg > 1:
    ndec = int(sys.argv[1])
else:
    ndec = 3

# - Programa principal -----
# Utiliza curses para visualizar salida a pantalla
# Muestra lectura del sensor y refresca pantalla cada 500 ms.
# -----
stdscr = curses.initscr()
curses.cbreak()
curses.start_color()
# Definicion de paletas de colores, borde
curses.init_pair(1, curses.COLOR_RED, curses.COLOR_BLACK)
curses.init_pair(2, curses.COLOR_YELLOW, curses.COLOR_BLACK)
stdscr.border(0)
hsize = curses.COLS
vsize = curses.LINES
stdscr.keypad(1)
stdscr.nodelay(1)
try:
    curses.curs_set(0)
# Rutina principal del programa
    while True:
        char = stdscr.getch()
        if (char == 113 or char == 81):
            break # quit
        else:
            stdscr.addstr(vsize/6, (hsize/2)-14,"Sensor 1820 - Raspberry Pi",curses.color_pair(1))
            stdscr.addstr((vsize/5)+2, (hsize/2)- 14,"Temperatura = ")
            stdscr.addstr((vsize/5)+2, (hsize/2) ,str(dametemp(ndec)),curses.color_pair(2))
            stdscr.addstr(vsize-4, (hsize/2)- 14,"Pulsa (Q) para salir")

            stdscr.refresh()
            time.sleep(0.5)
# Sale educadamente del programa
finally:
    curses.nocbreak()
    stdscr.keypad(0)
    curses.echo()
    curses.endwin()
```

Notas sobre el programa :

Este programa puede copiarse tal como está y guardarse en un fichero llamado **mide.py** en nuestra Raspberry Pi. También puede ser descargado desde la página de [DIVERTEKA](#) y copiarse a la carpeta de nuestra elección en la 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 sensor DS1820 a nuestra Raspberry Pi.
- (3) **;; IMPORTANTE !!** => Editar el código de **mide.py** y cambiar la línea con el ID del dispositivo por el nuestro.

Cambiamos **10-0008027f3664** por ejemplo por **XX-XXXXXXXXXXXX** en la línea
`tfile = open("/sys/bus/w1/devices/10-0008027f3664/w1_slave")`

(**XX-XXXXXXXXXXXX** será el ID de nuestro dispositivo, el cual podemos averiguar efectuando los comandos comentados desde el terminal en la sección **OPERATIVA SOFTWARE** de este documento)

- (4) Vamos al directorio del programa y damos al fichero **mide.py** permisos de ejecución con

`sudo chmod 777 mide.py`

- (5) En un terminal, desde el directorio del programa, ejecutamos

`python mide.py`

- (6) El programa no recorta los decimales mostrados, Podemos forzar a que muestre – por ejemplo – una sola posición decimal ejecutándolo con el parametro 1, o sea :

`python mide.py 1`

ENLACES DE INTERÉS

[DS1820](#): documento sobre el sensor DS1820 (formato PDF) de la firma Maxim.

[RASPBERRY 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

1-Wire es un protocolo de comunicaciones serie diseñado por **Dallas Semiconductor** (ahora **MAXIM**).

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

Fritzing es un programa para la representación de prototipos.

Adafruit es una página para el fomentar el aprendizaje online de electrónica.

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.