

6.3 TLC1543.py y VARIABLES.py

Tal y como hemos visto en la [teoría del TLC1543 ¿Cómo está conectado?](#) añadimos estas líneas al archivo **VARIABLES.py**

```
##SENSOR SIGUELINEAS

CS = 5
Clock = 25
Address = 24
DataOut = 23

##SENSOR SIGUELINEAS

CS = 5
Clock = 25
Address = 24
DataOut = 23
```

Script Damebit

En la [teoría del TLC1543 ¿Cómo funciona?](#) tenemos que obtener el bit de una posición dada de un número dado. [Aquí](#) hay un pequeño script para hacerlo (dale al play para ejecutarlo):

<https://repl.it/@deleyva/ObtenerBitEntero?lite=true>

TLC1543.py

Tal y como hemos visto en la [teoría del TLC1543 ¿Cómo funciona?](#) podemos hacer una librería que tenga una función **SENSORLINEA(cual)** que nos devuelva el valor que lee el sensor *cual*:

- Importamos las variables de **VARIABLES.py**
- Luego realizamos una función **SACADIRECCION** que active la salida ADDRESS según sus bits basándonos en la función **Damebit** que hemos visto.
- Activamos 4 golpes de reloj sacando la dirección **ADDRESS** con la función **SACADIRECCION**
- Hacemos 6 pulsos de **CLOCK** perdidos
- Hacemos 10 pulsos de **CLOCK** pero leyendo el valor **DATAOUT** y convirtiendo esos bits en un número decimal, ese será el valor que devolverá la función **SENSORLINEA(cual)**
- Grabamos esto en un archivo **TLC1543.py**

```
import RPi.GPIO as GPIO
import time

from VARIABLES import *

#####
#función de manipulación de bits
#ver https://repl.it/@javierquintana/ObtenerBitEntero
#####
def SACADIRECCION(x,n):
    if (x & (1<=n)):
        GPIO.output(Address,GPIO.HIGH)
    else:
        GPIO.output(Address,GPIO.LOW)

#####
#función de obtener lectura del sensor siguelíneas
#cual = el número del siguelíneas a leer 0-4
#####
def SENSORLINEA(cual):
    #activo el chip
    GPIO.output(CS,GPIO.LOW)
    for i in range(4):
        #Pongo en Address el bit de cual empezando por MSB
        SACADIRECCION(cual,3-i)
```

```
#Flanco de subida de Clock para que lo lea
GPIO.output(Clock,GPIO.LOW)
GPIO.output(Clock,GPIO.HIGH)
#ahora 6 pulsos perdidos
for i in range(6):
    GPIO.output(Clock,GPIO.LOW)
    GPIO.output(Clock,GPIO.HIGH)
#vamos a darle un tiempo para que calcue la convesión A/D
#A/D Conversion Interval =
time.sleep(0.001)
#leemos el número
#formula valor = sumatorio (potenciasde2 * bit leido)
#potenciasde2 = 2 elevado al peso del bit
#el peso del primer bit es MSB luego 9 y acaba en 0 o LSB
valor=0
for i in range (10):
    GPIO.output(Clock,GPIO.LOW)
    GPIO.output(Clock,GPIO.HIGH)
    valor=valor+GPIO.input(DataOut)*(2**(9-i))
#desactivamos el chip
GPIO.output(CS,GPIO.HIGH)
return valor
```

Revision #3

Created 21 March 2022 11:57:33 by Equipo CATEDU

Updated 22 March 2022 13:28:35 by Equipo CATEDU