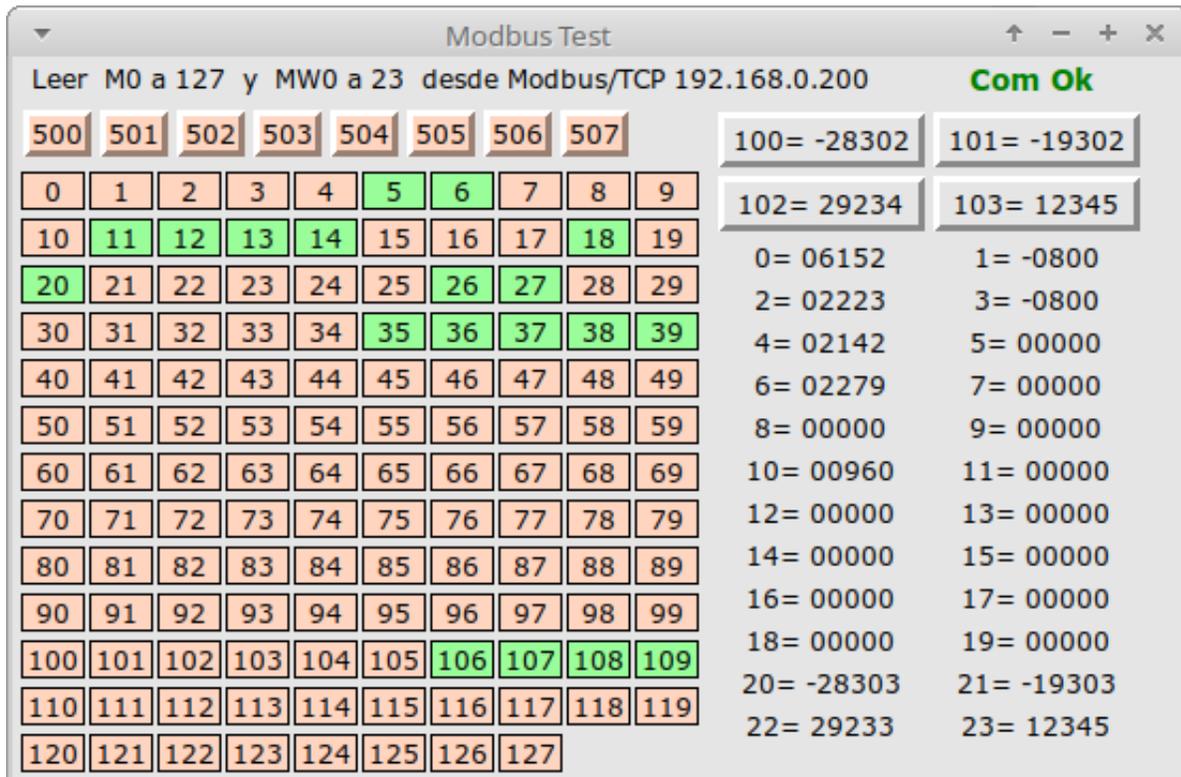


ModbusTest2

Esta pequeña aplicación, es una herramienta para probar la comunicación con un servidor Modbus/TCP.

Fue creada solo para fines de prueba, con el curso de capacitación, de programación de PLC. Es peligroso usarla para escribir datos hacia un PLC controlando un proceso real; no lo hagas, a menos que estés muy seguro de lo que estas haciendo.

Al ejecutar la aplicación vemos la ventana de la imagen.



En la esquina superior derecha, si se logra comunicar con el PLC dice **Com Ok**, en cambio si no logra comunicación dirá **Sin Com**

Los **botones** en la parte superior izquierda, son las discretas que se pueden leer y escribir.

Debajo, a la izquierda, las luces que muestran los estados de las discretas que esta leyendo. En rojo si están en 0, en verde si están en 1.

Los **botones** en la parte superior derecha, son las análogas que se pueden leer y escribir.

Las listas en la parte inferior derecha son los valores de las direcciones análogas que se están leyendo.

La configuración la lee, cuando se inicia, desde el archivo "config.txt" que debe estar en la misma carpeta sobre la que se esta ejecutando la aplicación. Esta puede ser cambiada con cualquier editor de texto.

Un ejemplo de "config.txt"

```
[Modbus]
ip = 192.168.0.200
port = 502
timeout = 0.2

[Digital Read Only]
init = 0
count = 128

[Analog Read Only]
init = 0
count = 24

[Digital Read And Write]
init = 500
count = 8

[Analog Read And Write]
init = 100
count = 4
```

¿ Como obtener una copia de esta herramienta ?

Alternativa 1 **el camino del impaciente**, solo quieres usarla, y no te interesa aprender. Descargar desde osdn.net/users/rolfds/pf/ModbusTest/ una versión compilada solo para Windows de 64bit.

Alternativa 2 **el camino del autodidacta**, si además de usarla, quieres aprender como funciona, y usarla en Windows, Linux y Mac por favor continua leyendo lo que sigue después, por que es software libre, y puedes copiarlo, usarlo y modificarlo, bajo los términos de la licencia GNU General Public License, versión 3.

Rolf Dahl-skog Stade. (Abril 2020)

ModbusTest2 esta programado en lenguaje python.

¿Que es python? Python es un lenguaje de programación multiplataforma y de código abierto (libre), diseñado para ser fácil de leer. es.wikipedia.org/wiki/Python

Antes de que te pierdas en muchos videos de youtube, que solo te van a confundir, **consejos básicos necesarios** que no aparecen en los muchos cursos que hay en Internet :

1.- Ir a la pagina www.python.org y descarga el instalador de la actual versión estable de python **3**. No desde la tienda de aplicaciones de Microsoft.

Importante: NO lo instales en la ruta que propone por defecto Windows. Selecciona instalación personalizada, y ponlo en una ruta simple (corta), como por ejemplo en `c:\python` esto te facilitará la vida después.

2.- Un programa en este lenguaje, es un archivo de texto plano, sin formato. Puede ser creado, o modificado, con cualquier editor de texto, sino fuera por que este lenguaje no usa `;` `{ }` ni ningún otro carácter extraño para delimita las instrucciones, para esto solo usa la [indentacion](#) o sangría de las líneas.

Si agregas o eliminas espacios al comienzo de una línea, o cambias espacios por tabulación, o cambias tabulación por espacios, fácilmente puedes dañar el programa. *Esto es motivo de confusión, se ve igual pero a mi no me funciona.* Por eso es bueno que puedas ver estos caracteres, y cualquier editores de texto no los muestra. Es muy recomendable usar un editor de texto que si muestre los caracteres espacio y tabulación, y muestre la indentacion.

Usa [Geany](http://www.geany.org) que además tiene resaltado de código, plegado automático, y varias utilidades mas. También es libre. www.geany.org
Dentro de las preferencias de Geany, habilitar la opción: Editor -> Mostrar espacios en blanco.

3.- El programa que escribes, solo es un archivo de texto con extensión `.py` no es un ejecutable. Para ejecutarlo debes pasárselo al interprete de lenguaje, que lo analiza, convierte y ejecuta.

Si con tu explorador de archivos al hacer click derecho, abrir, sobre `tuarchivo.py` no pasa nada, o te dice que no reconoce el tipo de archivo. Falta enseñarle a tu computador, que hacer con estos archivos, y la ruta hasta el interprete de python. Como establecer las rutas, depende de que cada versión de sistema operativo.

Al menos, enseñarle a Geany en que ruta instalaste python ...

Geany → menu → construir → establecer comando de construcion, y frente al boton "python" escribe: `c:\laRutaDondeEsta\python.exe "%f"`

Para que al presionar ejecutar, llame al interprete de python y le pase la direccion de tu archivo.

4.- Cualquier aplicación de utilidad, es mucho mas que **un** solo archivo. Generalmente son muchos archivos, que se importan, como módulos o librerías (colecciones de archivos). Algunos de estos módulos, los mas usados, ya se instalan por defecto, al instalar el lenguaje python, si necesitas otros debes agregarlos a tu sistema.

En **PyPi** (Python Package Index) pypi.org encuentras cientos de estos modulos, libres para descargarlos y usarlos

Si en tu programa pones `import tkinter` todos los archivo de la librería tkinter ahora también son parte de tu programa.

Si en tu programa pones `import pymodbus` probablemente te dirá "Error, no se encuentra este modulo". Muy pocas personas usan una librería Modbus, esta librería hay que instalar primero.

5.- ¿ Como agregar una librería ?

Cuando instalaste python, también se instalo una aplicación que se llama **pypi** que es justamente para eso, instalar paquetes desde pypi.org

En un terminal (consola de comandos) ingresa:

<code>pip install pymodbus</code>	para agregar la librería Modbus
<code>pip uninstall PIL</code>	quita la librería PIL
<code>pip install Pillow</code>	agregar la librería manipulación de imágenes
<code>pip list</code>	muestra que paquetes ya has agregado
<code>pip help</code>	

6.- Si la aplicación no funciona, como encuentro que falla ?

Python siempre envía todos los mensajes de error hacia el terminal (consola de comandos). Ejecútala desde un terminal, y te mostrara donde esta cualquier problema que encuentre.

Si lo ejecutas con Geany el terminal continuará abierto, incluso aunque tu programa falle y se caiga, para que puedas ver en que linea del programa esta el error.

Mi primera ventana

Usando [Geany](#) menu → archivo → nuevo. Escribe (o copia) ...

```
from tkinter import * # Carga módulo de widgets estándar
ventana = Tk()
ventana.geometry('300x200') # anchura x altura
ventana.configure(bg = 'beige')
ventana.title('Mi ventana')
boton = Button(ventana, text='Salir', command=quit).pack(side=BOTTOM)
ventana.mainloop()
```

menu → archivo → Guardar ...

menu → construir → ejecutar

Felicitaciones. Creaste tu primera ventana !!!

Ahora puedes abrir `ModbusTest2.py` u otros ejemplos, y comenzar a jugar.

<https://python-para-impacientes.blogspot.com/p/indice.html>

<https://python-para-impacientes.blogspot.com/p/tutorial-de-tkinter.html>

<https://docs.python-guide.org/>

<https://learnxinyminutes.com/docs/es-es/python-es/>

<https://docs.python.org/3/> documentacion python version 3

<https://docs.python.org/3/library/tkinter.html>

<https://devfreebooks.github.io/python/>