

TCPModbusRemote v.01

Paul D. Fincato

2 May 2017

Freely distributable under the Creative Commons license

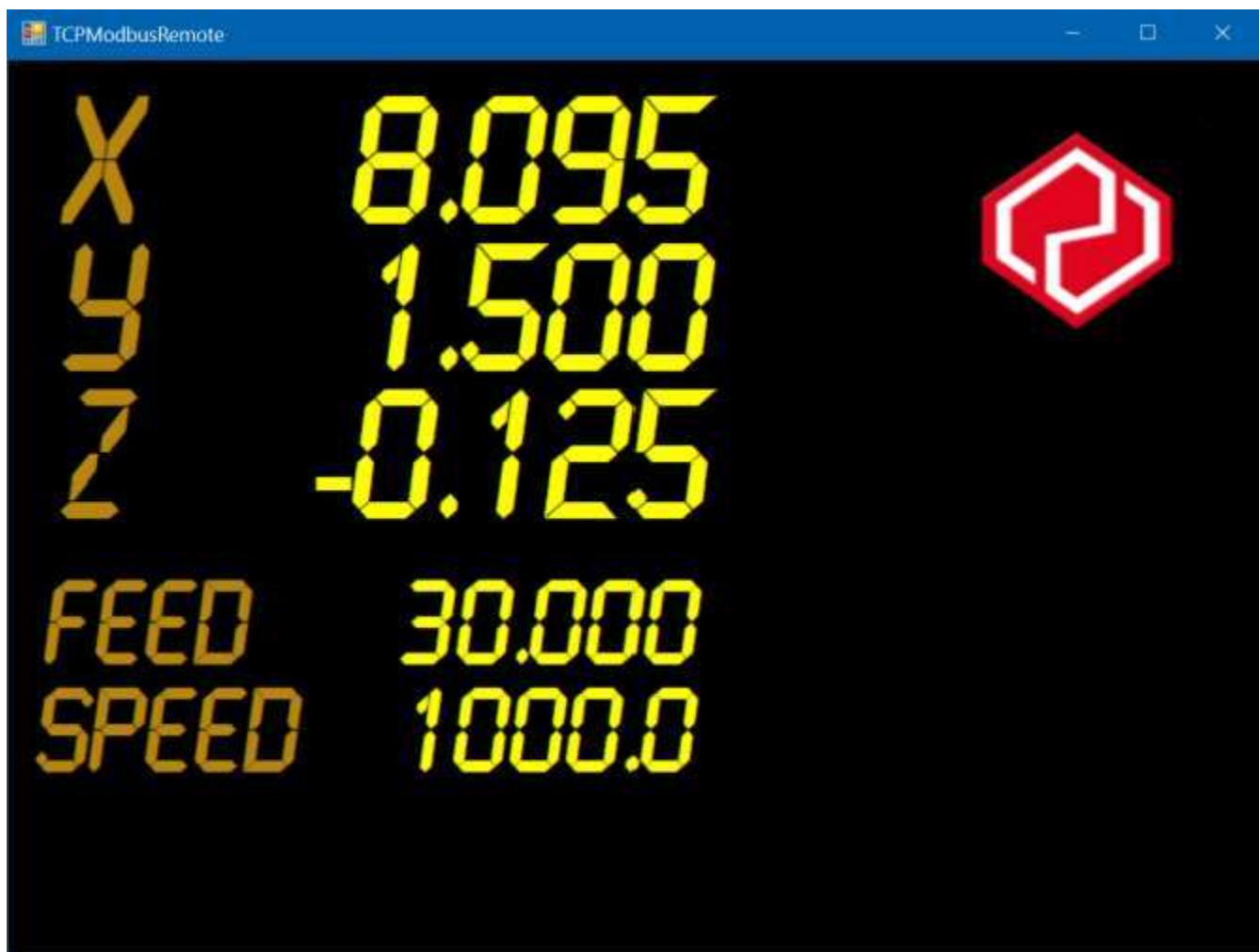
Allows an old tablet or PC to be used as 'wallboard' DRO indicator. This should work very nicely if the remote has a WiFi connection to the same network as the UCCNC PC. This is more of a proof of concept than a completed project, but the functionality provided is complete.

Developed with SharpDevelop 5.1 on Windows 10. Not tested on other Windows versions, but is expected to work.

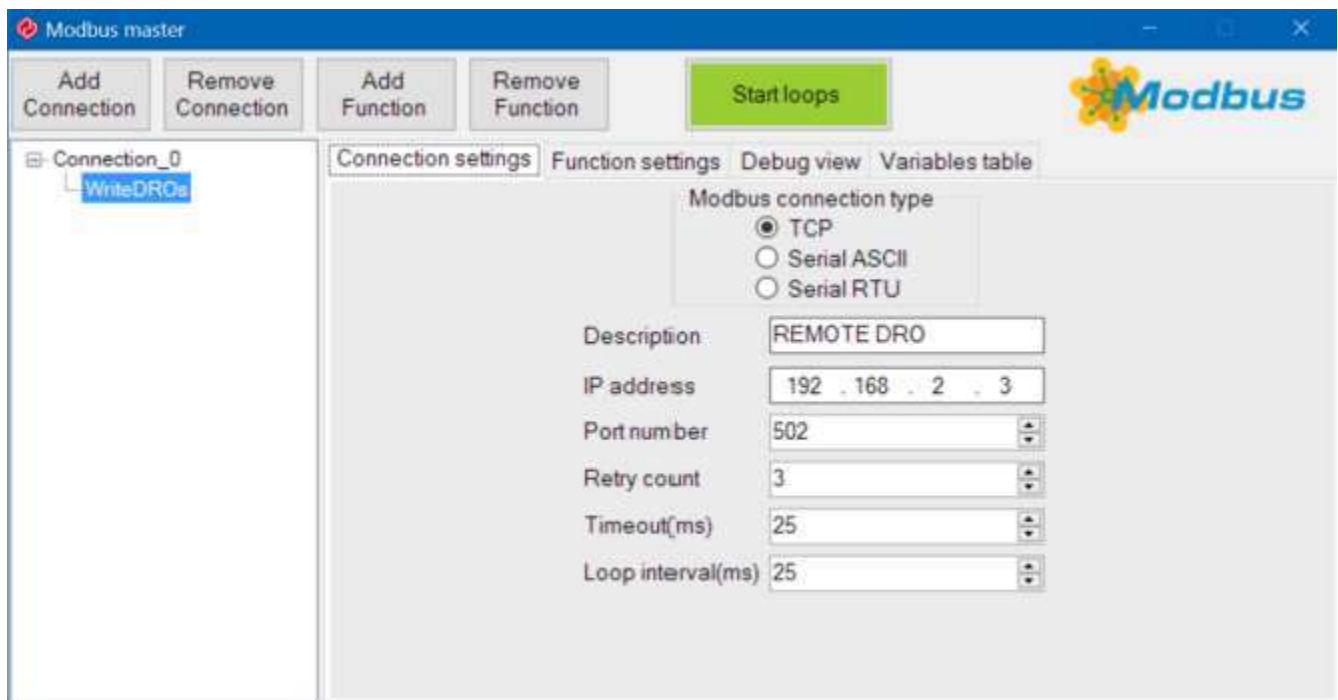
Prerequisites:

The LCD font in the 'support' needs to be installed on the machine running the remote.
.NET framework 4.0 runtime on the remote.

The remote display



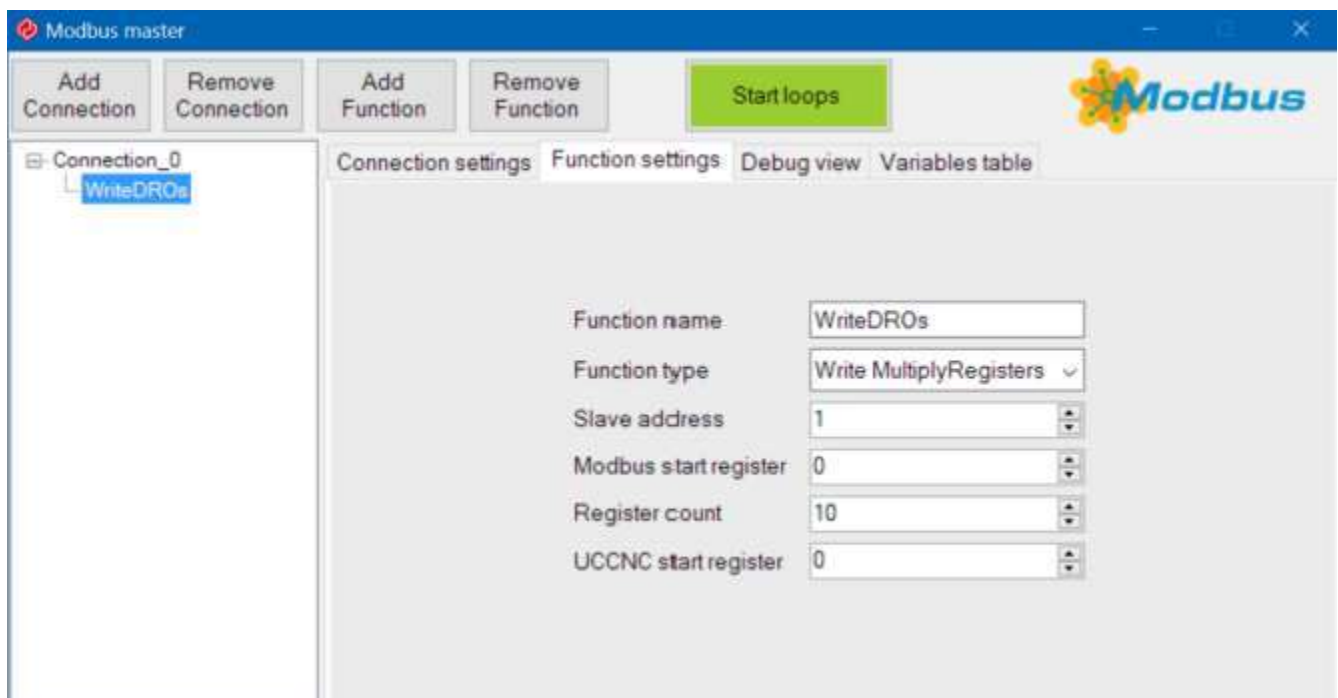
Set the IP address to the machine running the remote DRO software. It can also be on the the same machine running UCCNC for testing. Run 'cmd' and type 'ipconfig /all' if you don't know the machine's address.



The screenshot shows the 'Modbus master' application window. The 'Connection settings' tab is active. On the left, a tree view shows 'Connection_0' with a sub-item 'WriteDROs'. The main area contains the following settings:

- Modbus connection type: ☒ TCP, ☐ Serial ASCII, ☐ Serial RTU
- Description: REMOTE DRO
- IP address: 192 . 168 . 2 . 3
- Port number: 502
- Retry count: 3
- Timeout(ms): 25
- Loop interval(ms): 25

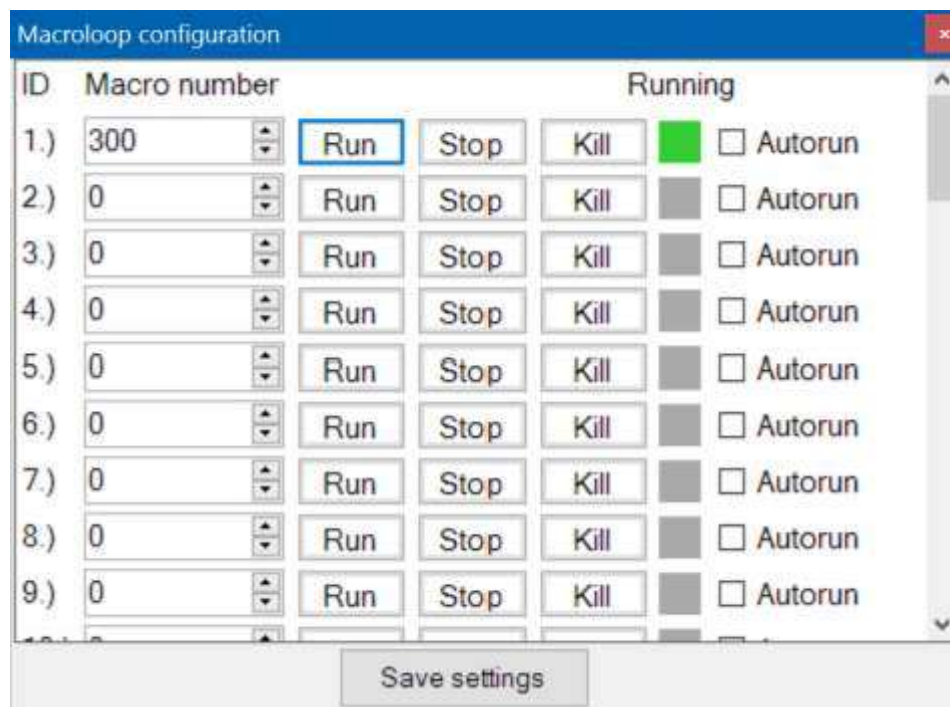
A function to write the values must be created. Register count must be 10 for the current macro



The screenshot shows the 'Modbus master' application window with the 'Function settings' tab active. The 'WriteDROs' function is selected in the tree view. The main area contains the following settings:

- Function name: WriteDROs
- Function type: Write MultiplyRegisters
- Slave address: 1
- Modbus start register: 0
- Register count: 10
- UCCNC start register: 0

M300.txt macro needs copied your default profile (The number 300 was arbitrary and could be changed) Check the autorun if desired, once testing is complete.



TECHNICAL DETAILS

The EasyModbus DLL takes care of all the heavy lifting. The only tricky part is transmitting floating point numbers to the remote. Modbus can only work with a maximum of 16 bit unsigned integers. Floats are converted to integers in the macro, transmitted and converted back to floats in the remote. The BitConvertor class helps here.

In the remote, the Modbus class runs in a separate thread, so the Invoke method is used to communicate values to the GUI thread.

CREDITS

EasyModbusTCP/UDP/RTU .NET from sourceforge.net -Using this package made the remote C# programming a cinch!

Samuel Reynolds for the LCD font

CNCDrive for a great product - UCCNC

SharpDevelop 5.1