



PowerTools Pro Application Note
PTAN #14, rev. 1, 9/30/2009
Applicable Products: PowerTools Pro

PowerTools Pro
Latency
Communications
Settings

Introduction

PowerTools Pro communicates with the drive using Modbus RTU (serial) and Modbus TCP (ethernet) protocols. In most cases, the user does not have to modify the default communications setting, but occasionally network latency can cause communications issues.

Latency is when communication packets take longer than expected to arrive. When talking about Ethernet, latency can be a function of physical distance, network traffic and time of day. However, in the serial communications environment, we also need to deal with latency. This is especially important when we introduce an environment that does not include a physical serial port. Devices that can fall into this category are:

- USB to serial devices (dongles)
- Ethernet to serial devices

Their common component is a virtual serial port installed that simulates the COM port, but in reality redirects the data packet to either a USB device or over the Ethernet to a remote device.

PowerTools is unable to tell the difference between a virtual serial port and a physical serial port. PowerTools was written to have aggressive communications response with the drive because it assumed that the drive was hard-wired to the PC. Therefore, when using a virtual serial port, the user is responsible for determining that potential serial communication problems are a result of latency between PowerTools and the drive.

The rest of this application note details the modifications to the PowerTools initialization file to deal with latency issues

Basic Setup

The following table lists the parameters that can be changed in the PowerTools 'ini' configuration file (emc_fm3.ini or emc.ini).

CServer	<i>These parameters are to extend the behavior of CServer. They are primarily for the "Serial over Ethernet" devices. ALL VALUES ARE IN MILLI-SECONDS</i>		
	Parameter	Default	Notes
	ExtraMsgTimeout	0	Timeout for receiving data within a response packet after receiving a partial packet.
	ExtraPacketTimeout	0	Timeout for receiving the first data from a response packet.
	InterPacketDelayOverride	0	If device does not support "WaitCommEvent()" sleep (m_iInterPacketDelayOverride) instead.
	ErrorFlushTime	200	Time spent after a timeout or CRC error to receive data and ignore it.
	ExtraRetries	0	Extra Retries
	ExtraPacketDelayOverride	10	Extends the delay between consecutive packets (for multi-drop)
	DirectReadTimeout	100000	The timeout for the response from a direct read.
ActivePort	-1	-1 indicates CServer is not active. Otherwise it is the COM port number CServer is actively using.	

Understanding the CServer Parameters

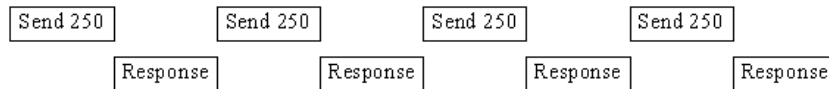
In order to understand the parameters definitions, we need to understand how Modbus packets work. A Modbus packet is up to 255 bytes long. Once the master has sent its packet, the slave will wait 3.5 characters (1 byte plus start/stop bits) after the last byte is transmitted before it starts to process the received bytes. Once processed, the slave then transmits a response. The master will wait 3.5 characters after the last byte is received before it processes the received bytes. As you may already have noticed, there needs to be a 3.5 character gap between sent and received packets. Each packet sent by the master will have one packet response.

PowerTools quite often will request data that requires the slave to transmit more than 255 bytes, or send data that is greater than 255 bytes. In this case the data packet is fragmented into smaller pieces so that it can be transmitted within the required 255 byte limit. As far as Modbus is concerned, each fragmented packet is considered one complete Modbus message. It is the higher protocol that fragments, and re-assembles these packets into the required send and responses PowerTools requires.

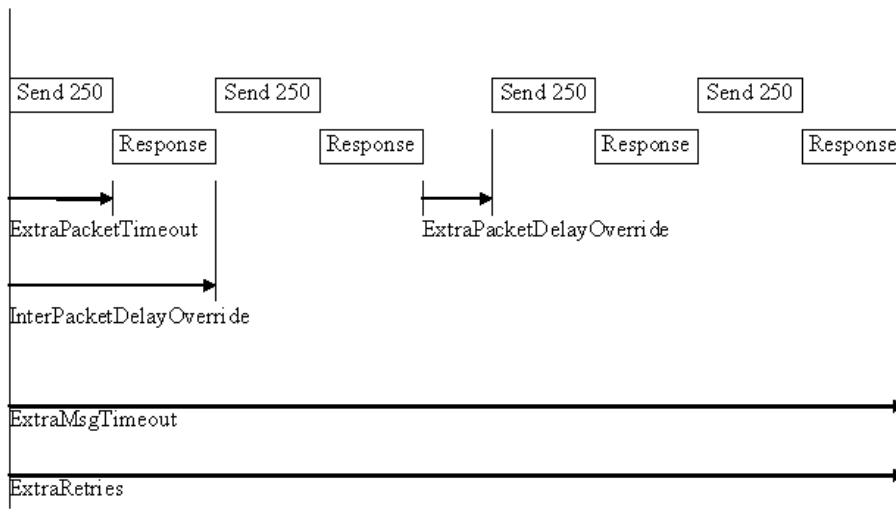
For example:



would be sent as:



Using this fragmented picture, we can now show how each parameter relates to latency:



Setting the CServer Parameters

The PowerTools Pro initialization file is called 'emc_fm3.ini' and is located within the PowerTools Pro installation directory. This file contains initialization and screen setup information for use by PowerTools.

The file can be opened with a simple text editor like Notepad. Don't worry about corrupting the file; the file can be deleted without any harm to the software operation. If the file is not found, PowerTools will recreate the default file when it starts.

The default setup for the CServer looks like this:

```
[CServer]
ActivePort=-1
```

Modifying the CServer, simply add in any additional parameters:

```
[CServer]
ActivePort=3
ExtraMsgTimeout=50
ExtraPacketTimeout=15
ExtraPacketDelayOverride=100
```

Save the file and restart PowerTools.

The values show above are example only, your system will require its own values that must be determined by educated trial and error.