GE Industrial Systems
We bring good things to life.

Technical Note #78

DDE update time

Subject: Modbus RTU device update time

Applies To: Any Modbus network utilizing a RS-485 half-duplex standard

Modbus RTU Background:

The Mobus RTU protocol is a master/slave transaction where a master device generates a query and a slave device generates a response. A slave should never give a response unless told by the master.

The time it takes for the device to answer is the "device response time". The "device update time" is how quickly the queried data is updated in the master for that particular device. Note: Network baud rate accounts for only a small percentage of the device response time. Internal processing of the Modbus command by the device takes up the majority of the time.

The nature of this protocol is that only one device can be queried at a time on the same network. If a network has only one device on it that the "device response time" and the "device update time" should be roughly equal.

What if there are multiple devices on one line? If the master needs to query all the devices it must query them one at a time. Let's suppose a network has 3 devices on it(device 1, 2 and 3) and the mater needs data from all three. To get this data the master would do the following:

1. Query device one for data,
2. Listen for the response
3. Receives the response from device one
4. Query device two for data,
5. Listen for the response
6. Receives the response from device two
7. Query device three for data,
8. Listen for the response
9. Receives the response from device three

The whole process is then repeated as needed. From the steps outline above

it can be seen that the "device update time" is dependent on how fast the other devices on the network can respond. Let's suppose that each device has a device update time of 1 second. This means that the master will see fresh data from each device every three seconds(1+1+1=3). The more devices added to the network the longer each device's update time!

Device update times for PMCS:

PMCS utilizes a wide range of device ranging from breaker units to motor controllers to meters. Each device has a unique register map and varying amounts of registers. When a PMCS wizards calls for register information the DDE server "looks" at the request and creates a Modbus request according to Modus rules.

If all the registers needed can fit into one Modbus return message (125 registers in order) than only one request is made. However if the amount of registers is too many for one return message than the server breaks the wizard request into two or more Modbus requests.

Devices with small register maps, such as an MVT, generally need only one Modbus request with a device response time of about 500 Milliseconds. But larger devices (MLPQM, EPM3720,etc) may require 4 or more Modbus requests with a device response time that could be as much as three seconds.

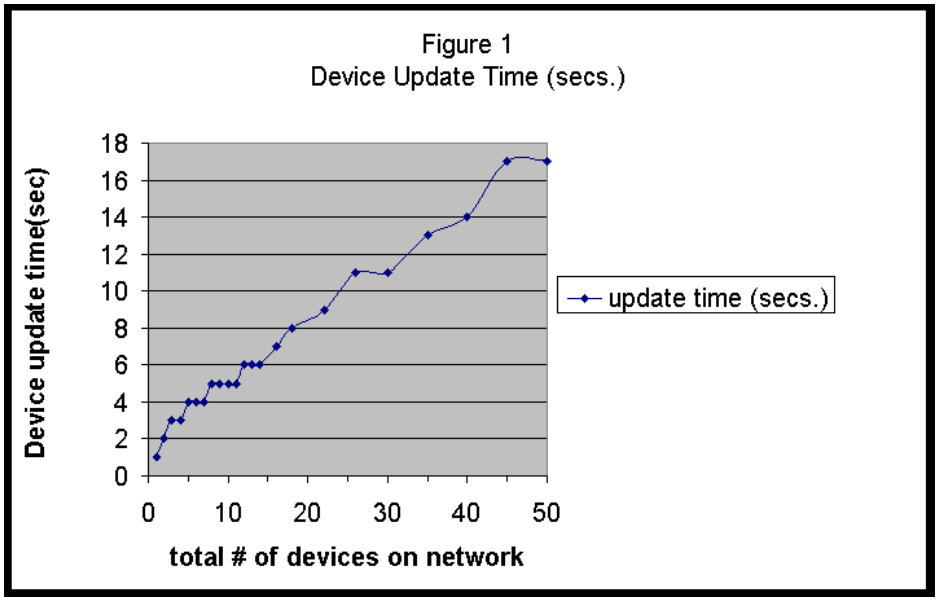
Test Cases:

Note: The following data was derived from test racks in the lab. Data from real customer PMCS systems will be added to this appnote as they become available.

Case 1: Testing was done in the lab using 48 MVT's and 2 EPM3710's. PMCS 6.0 was used and an application was developed which would ask for all large PMCS faceplate registers from 1 device, than 2 devices, than three devices, etc, etc.

Results:

See Figure 1.



Analysis:

The distribution suggests a roughly linear system. Doubling the number of MVT's almost doubles the device response time.

PMCS System Recommendation:

The quickest possible PMCS system would have only one device per network. For 99% of real world users this is impractical due to cost, device layout, wiring requirements, etc. Also these update times are assuming that all the devices on the network will be polled on a regular basis due to historical trending, alarm monitoring, etc. If none of this is occurring in an application than the device update time should be equal to the **number of devices actually being queried** regardless of the actual number of device on the network

Related Notes: N/A

Last Revised July 1, 1999