MODBUS DIAGNOSTICS

Taught by:
Douglas Novy

1

Modbus Diagnostic Topics

- Modbus Command Monitor
  - Observer the commands being received by the logger
    - (8864 is the Modbus Slave)
- Error/Status Debugging in Holding Register
  - Diagnose server.cfg based issues
    - (8864 is the Modbus Master)
- TCP Dump & Wireshark
  - Record and check all ethernet/serial traffic to the logger

2

2

MODBUS COMMAND MONITOR

3

Modbus Command Monitor

The 8864 Data Controller (v5.04r13 and above) displays all incoming Modbus commands.

From the Home Screen:

(S) Status Menu

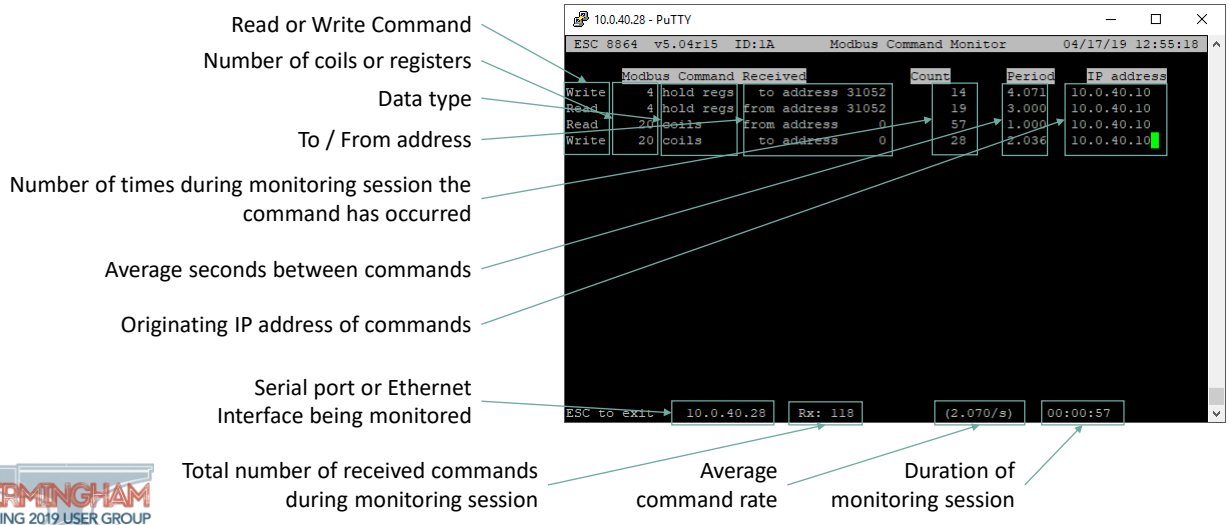
(V) View Modbus Master Status

(V) Modbus Command Monitor

Select Ethernet Port of interest

4

Modbus Command Monitor



ERROR/STATUS DEBUGGING

Error/Status Codes

The ESC Data Controllers have dedicated registers in the Client Table for the purposes of troubleshooting Modbus error/status codes.

These codes are especially helpful when troubleshooting errors in the server.cfg file

(Logger is the Modbus master)

7

Example server.cfg file

Error/Status Codes

server.cfg - Notepad

File Edit Format View Help

```
# COMPANY    : ESC
# DATE       :
# CREATED BY: ESC ENGINEERING
# EDIT DATE :
# EDITED BY :
[Module]
Module Name : 8864-Modbus-Master

[8864 Client 0]
Minimum Command Delay    :    10 #Minimum number of msec's between commands
Response Timeout         :  2000 #Response message timeout (0-65535 mSec)
Retry Count              :     0 #Response failure retry count
Float Flag               :    No #Use floating-point data type Y=Yes, N=No
Float Start              :  7000 #Register offset in message for floats
Float Offset             :  2000 #Internal Address for floats

[8864 Client 0 Commands]
START
#        Internal  Poll      Reg    Swap  IP              Serv  Slave    Func    MB Address
# Enable  Address   Interval  Count  Code  Address         Port  Address  Code    In Device
#-------------------------------------------------------------------------------------------
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        1       XX        #  READ coil
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        2       XX        #  READ discrete input
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        3       XX        #  READ registers - (4XXXX)
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        4       XX        #  READ input registers - (3XXXX)
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        5       XX        # WRITE coil
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        6       XX        # WRITE register -- (single)
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        15      XX        # WRITE coil ------ (multiple)
# 1       XXX       10        1      0     XX.XX.XX.XXX    502   1        16      XX        # WRITE registers - (multiple)
#-------------------------------------------------------------------------------------------

# READ INDIVIDUAL COILS FROM REMOTE DEVICE AND SAVE TO LOCAL HOLDING REGISTERS 400-409
  1       400       50        1      0     10.10.10.10     502   1          1       0          #READ COIL 0

# WRITE CHANNEL DATA TO REMOTE DEVICE, FLIPPING LSW and MSW, FROM HOLDING REGISTERS 303-306
  1       303       50        1      0     10.10.10.10     502   1          16      1          #Write 2nd reg UNITON

# WRITE INDIVIDUAL COILS FROM DATA MAPPED TO A SINGLE HOLDING REGISTER 500 (ASK ABOUT MULTIPLYING BY 16)
  1      8000       50        1      0     10.10.10.10     502   1          5       17         #WRITE COIL 17

# READ CHANNEL DATA FROM A DIFFERENT REMOTE DEVICE, FLIPPING LSW and MSW, AND STORING IN HOLDING REGISTERS 351-356
  1       351       50        1      0     10.10.10.12     502   1          3       1          #READ 2nd reg of HG_ON
END


[8864 Data Map]
START
#        Internal  Poll      Reg    Swap  Modbus  MB Address
# Enable  Address   Interval  Count  Code  Func    in Slave
#-------------------------------------------------------------------------

 1       303       100       16     0     3       9 # READ 16 REGSITERS (8 FLOAT CHANNELS) AND STORE IN HOLDING REGISTERS 303 and up
 1       400       100       16     0     15      0 # WRITE 16 COILS FROM HOLDING REGISTER 400 to COILS 0 and up
END
```

8

Error/Status Codes

The following table includes the default starting addresses of these error/status registers.

Port	Default Error/Status Registers Starting Address
Ethernet Port Client 0	6000
8864 Serial Port 0	6200
8864 Serial Port 1	6400
8864 Serial Port 2	6600
8864 Serial Port 3	6800
Ethernet Port Client 1	7000
Ethernet Port Client 2	7200
Ethernet Port Client 3	7400
Ethernet Port Client 4	7600
Ethernet Port Client 5	7800
Ethernet Port Client 6	8000
Ethernet Port Client 7	8200
Ethernet Port Client 8	8400
Ethernet Port Client 9	8600

9

Error/Status Codes

There are 10 dedicated registers, each reserved for different status information.

The following registers indicate error/status codes for each command.

Offset	Error/Status Description
0	Number of commands sent.
1	Number of responses received.
2	Number of command errors.
3	Not used.
4	Not used.
5	Not used.
6	Not used.
7	Not used.
8	Current error code (see table below)
9	Last error code (see table below)
10	Latest error/status code for command 1
11	Latest error/status code for command 2
:	:
N	Latest error/status code for command N

10

Error/Status Codes

The following tables shows the possible error/status codes that are possible for each command.

Error Code	Error/Status Description
0	No error.
1	Standard Modbus protocol error – Invalid function code
2	Standard Modbus protocol error – Address validation error
3	Standard Modbus protocol error – Data validation error
4,5,6	Standard Modbus protocol error – Command specific errors
-11 (0xFFF5)	Serial port – response timeout.
-33 (0xFFDF)	Ethernet port – no connection established.
-36 (0xFFDC)	Ethernet port – response timeout.

[6000] **#** of Commands Sent

[6001] **#** of Responses Received

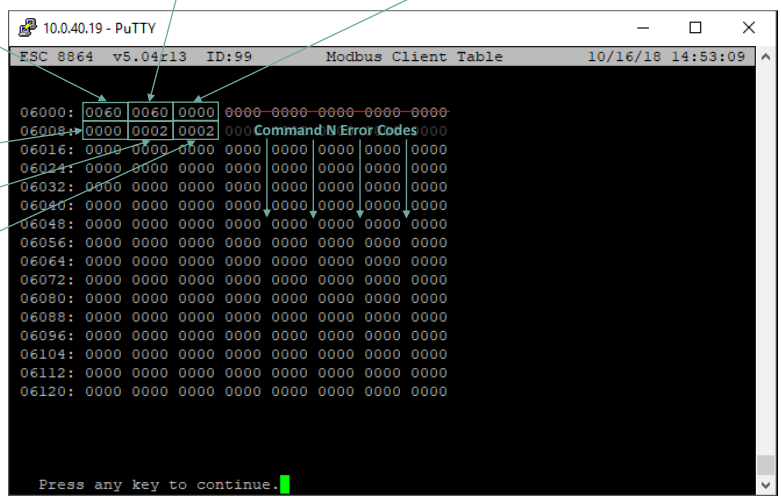
[6002] **#** of Errors

[6008] Current Error **Code**

[6009] Last Error **Code**

[6010] Command 1 Error **Code**

[6011] Command 2 Error Code
[6012] Command 3 Error Code
...
[6NNN] Command N Error Code
(Up to Registers 6199)



Error/Status Codes

Client 0:

06000 Register

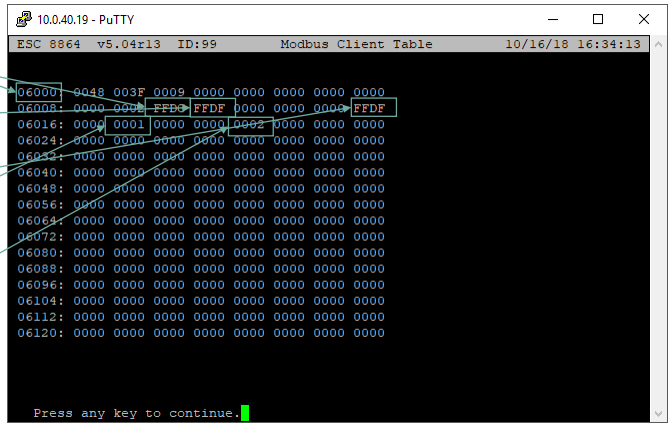
[6010] Command 1: Error FFDC
ethernet response timeout

[6011] Command 2: Error FFDF
no connection established

[6015] Command 6: Error FFDF
no connection established

[6017] Command 8: Error 1
invalid function code

[6020] Command 11: Error 2
address validation error



13

TCP DUMP

14

What Is TCP Dump?

TCP Dump allows you to record all network traffic in & out of the controller for troubleshooting purposes.

15

15

How Do I Capture a TCP Dump?

From home screen…

(S) Status Menu

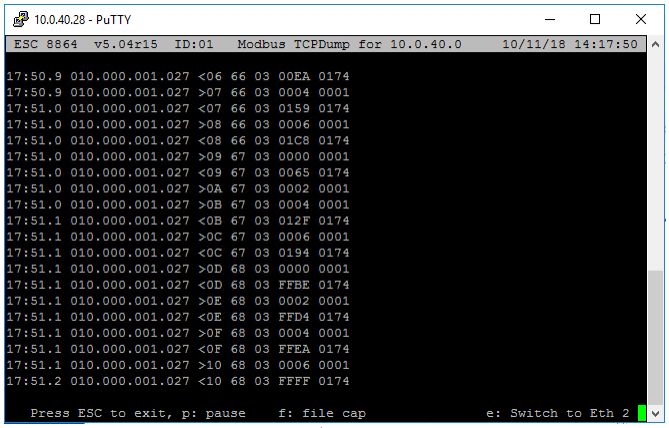
(V) View Modbus Master Status

(M) Modbus TCPdump

16

16

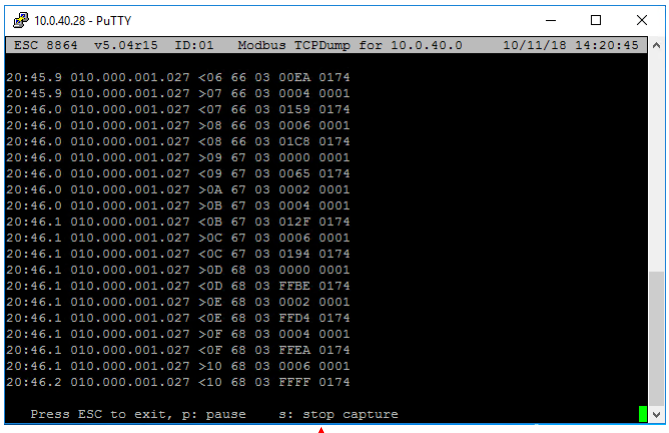
Start Capture



Push ‘f’ to capture the TCP dump

17

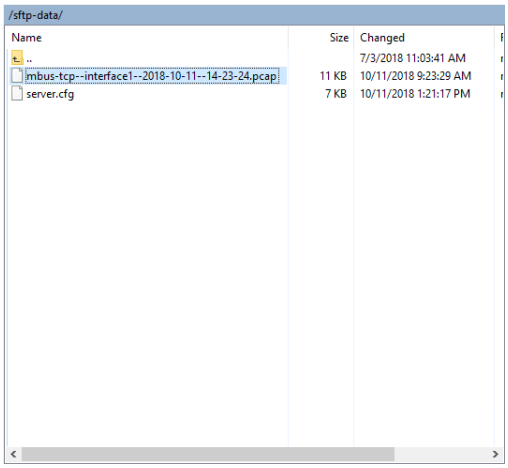
Stop Capture



Push ‘s’ to stop the TCP dump

18

Retrieve File Using WinSCP

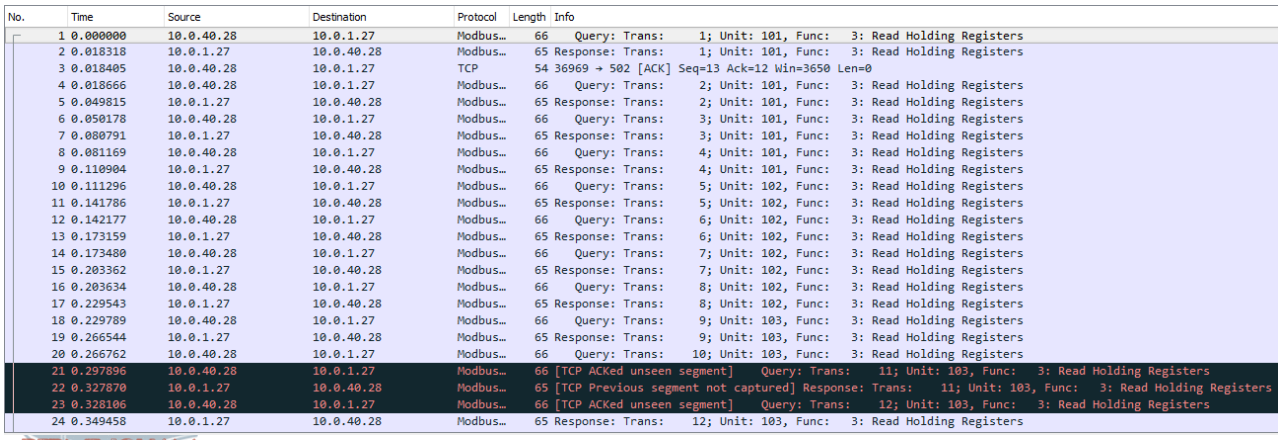


19

Enter Wireshark…

20

Wireshark Example



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 1; Unit: 101, Func: 3: Read Holding Registers
2	0.018318	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 1; Unit: 101, Func: 3: Read Holding Registers
3	0.018405	10.0.40.28	10.0.1.27	TCP	54	36969 → 502 [ACK] Seq=13 Ack=12 Win=3650 Len=0
4	0.018666	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 2; Unit: 101, Func: 3: Read Holding Registers
5	0.049815	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 2; Unit: 101, Func: 3: Read Holding Registers
6	0.050178	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 3; Unit: 101, Func: 3: Read Holding Registers
7	0.080791	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 3; Unit: 101, Func: 3: Read Holding Registers
8	0.081169	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 4; Unit: 101, Func: 3: Read Holding Registers
9	0.110904	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 4; Unit: 101, Func: 3: Read Holding Registers
10	0.111296	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 5; Unit: 102, Func: 3: Read Holding Registers
11	0.141786	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 5; Unit: 102, Func: 3: Read Holding Registers
12	0.142177	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 6; Unit: 102, Func: 3: Read Holding Registers
13	0.173159	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 6; Unit: 102, Func: 3: Read Holding Registers
14	0.173480	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 7; Unit: 102, Func: 3: Read Holding Registers
15	0.203362	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 7; Unit: 102, Func: 3: Read Holding Registers
16	0.203634	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 8; Unit: 102, Func: 3: Read Holding Registers
17	0.229543	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 8; Unit: 102, Func: 3: Read Holding Registers
18	0.229789	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 9; Unit: 103, Func: 3: Read Holding Registers
19	0.266544	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 9; Unit: 103, Func: 3: Read Holding Registers
20	0.266762	10.0.40.28	10.0.1.27	Modbus…	66	Query: Trans: 10; Unit: 103, Func: 3: Read Holding Registers
21	0.297896	10.0.40.28	10.0.1.27	Modbus…	66	[TCP ACKed unseen segment] Query: Trans: 11; Unit: 103, Func: 3: Read Holding Registers
22	0.327870	10.0.1.27	10.0.40.28	Modbus…	65	[TCP Previous segment not captured] Response: Trans: 11; Unit: 103, Func: 3: Read Holding Registers
23	0.328106	10.0.40.28	10.0.1.27	Modbus…	66	[TCP ACKed unseen segment] Query: Trans: 12; Unit: 103, Func: 3: Read Holding Registers
24	0.349458	10.0.1.27	10.0.40.28	Modbus…	65	Response: Trans: 12; Unit: 103, Func: 3: Read Holding Registers

Questions?