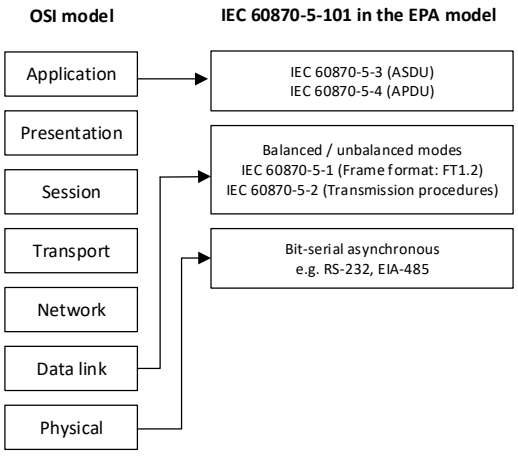
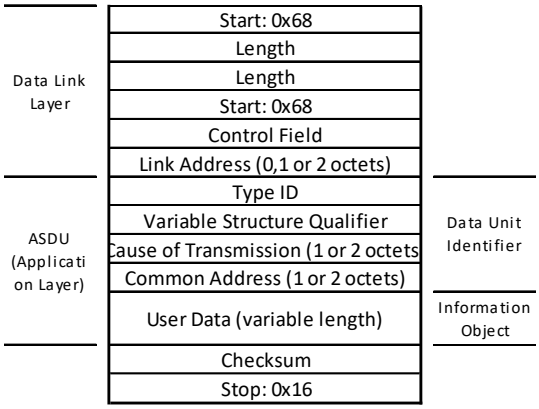
SCADA Protocols IEC 60870-5

- IEC 60870-5 is a collection of standards for SCADA
  - » DNP3 was developed at same time – out of draft version
- IEC 60870-5-101
  - » Serial over low bandwidth channel
  - » First version in 1994 – called T101
- IEC 60870-5-104
  - » Specifies the TCP/IP suite in the network and transport layers to provide LAN or WAN operation
  - » Added in 2000

IEC 60870-5 in OSI Model

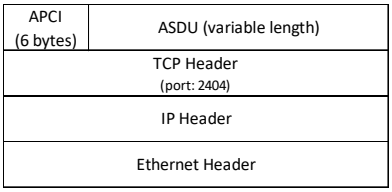


T101 Message Format

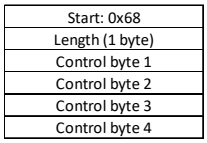


T104 in TCP/IP

- Defines network operation of T101 over TCP/IP



- Application Protocol Control Information (ACPI) is a control field



SCADA Protocols: MODBUS

- Created in 1970's by Modicon
  - » Has passed through several owners over the years
  - » Initial developed for PLC
  - » Widespread use made it a de-facto standard
- https://modbus.org

SCADA Variants

- MODBUS ASCII
  - » ASCII symbols
- MODBUS RTU
  - » Binary digital data messages
- MODBUS Plus
  - » Extension from one group – not widely used
- MODBUS TCP/IP
- MODBUS UDP

SCADA Protocols: MODBUS

- Created in 1970's by Modicon
  - » Has passed through several owners over the years
  - » Initial developed for PLC
  - » Widespread use made it a de-facto standard
- https://modbus.org

SCADA Variants

- MODBUS ASCII
  - » ASCII symbols
- MODBUS RTU
  - » Binary digital data messages
- MODBUS Plus
  - » Extension from one group – not widely used
- MODBUS TCP/IP
- MODBUS UDP
- All use OSI Layers 1, 2 and 7

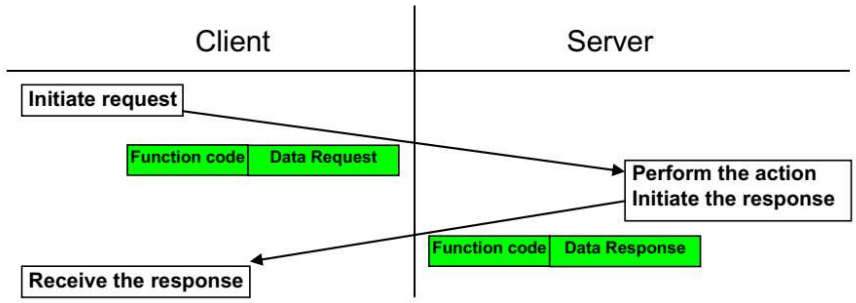
MODBUS Pros

- Easy to read/troubleshoot
- Has server device addressing
- Has built-in data integrity checking (CRC)
- Readily available on most devices
- Since sent over physical media can observe traffic to troubleshoot
  - » Serial COM port sniffer
  - » Wireshark or similar TCP/IP or UDP

MODBUS Pros and Cons

- Pros
  - » Easy to read/troubleshoot
  - » Has server device addressing
  - » Has built-in data integrity checking (CRC)
  - » Readily available on most devices
- Cons
  - » Message start/start all time based
  - » Does not have client device addressing
  - » Message size limitations slows updates, especially on serial channels
  - » No data quality, timestamps without adding special logic
  - » While TCP/IP security is available, not commonly used

Error Free MODBUS Transaction (polling)



- Server given a device address
- From: **MODBUS APPLICATION PROTOCOL SPECIFICATION, V1.1b3** (http://www.modbus.org)

11

Data units

- ADU → Application Data Unit
  - » Entire message with layer 2 addressing

|-------PDU-------------------|

ADDR	DATA	CRC

- PDU→ Protocol Data Unit

Function Code	Data

12

Data Types

- Discrete Input → Single read only bit
- Coils→ Single read/write bit
- Input Registers→ 16-bit read-only word
- Holding Registers→ 16-bit read/write word

- Big-Endian when more than single byte is used to describe message
  - » The most significant byte is sent first

Numbering of Registers

- Discrete inputs: 10,001-20,000
- Coils: 0-10,000
- Input registers: 30,001-40,000
- Holding register: 40,001-50,000

Function Codes

Code	1/16-bit	Description	I/O Range
01	1-bit	Read coils	00001 – 10000
02	1-bit	Read contacts	10001 – 20000
05	1-bit	Write a single coil	00001 – 10000
15	1-bit	Write mulitple coils	00001 – 10000
03	16-bit	Read holding registers	40001 – 50000
04	16-bit	Read input registers	30001 – 40000
06	16-bit	Write single register	40001 – 50000
16	16-bit	Write mulitple registers	40001 – 50000
22	16-bit	Mask write register	40001 – 50000
23	16-bit	Read/write mulitple registers	40001 – 50000
24	16-bit	Read FIFO queue	40001 – 50000

Message Example

- Query
  - » 02 03 1F A4 00 1F 42 06
    - 02 //address 2
    - 03 //read holding register
    - 1F A4 // *40001* + **8100**
    - 00 1F // 31 POINTS
    - 42 06 // crc

Message Example

- Response
  - » 02 03 3E 80 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F9 CB 05 25 68 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 67 F4
    - 02//address 2
    - 03//read holding register
    - 3E // (62 bytes * 8bits/byte)/16bits = 31 registers
    - 80 01//register 1
    - 00 00//register 2
    - ……//register 3 - 30
    - 00 07//register 31
    - 67 F4 // crc

Write Multiple Registers

Request

Function code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	0x0001 to 0x007B
Byte Count	1 Byte	2 x **N***
Registers Value	**N*** x 2 Bytes	value

***N** = Quantity of Registers

Response

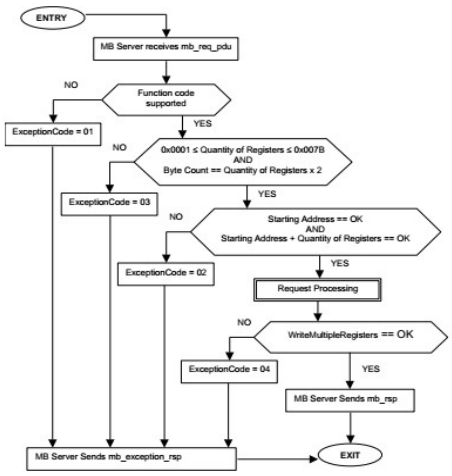
Function code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 123 (0x7B)

Error

Error code	1 Byte	**0x90**
Exception code	1 Byte	01 or 02 or 03 or 04

- From: **MODBUS APPLICATION PROTOCOL SPECIFICATION, V1.1b3** (http://www.modbus.org)

Write Multiple Registers Block Diagram



- From: **MODBUS APPLICATION PROTOCOL SPECIFICATION, V1.1b3** (http://www.modbus.org)

19

Exceptions

- Exception Response
  - Add 80_H to the request
- Exception Codes
  - 01_H = Illegal Function
  - 02_H = Illegal Data Address
  - 03_H = Illegal Data Value
  - 04_H = Slave Device Failure
  - 05_H = Ack
  - 06_H = Slave Device Busy
  - 07_H = Neg Ack
  - 08_H = Memory Parity Error

20

Exception Response Example

– Exception Response Example
  - Request - "Read Coil" (Function Code 01_H)
  - Response adds 80_H in response for Function Code 81_H
  - Data replaced with Exception Code 02_H for "Illegal Data Address"

Request		Response	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	01	Function	81
Starting Address Hi	04	Exception Code	02
Starting Address Lo	A1		
Quantity of Outputs Hi	00		
Quantity of Outputs Lo	01		

Example MODBUS Network Architecture

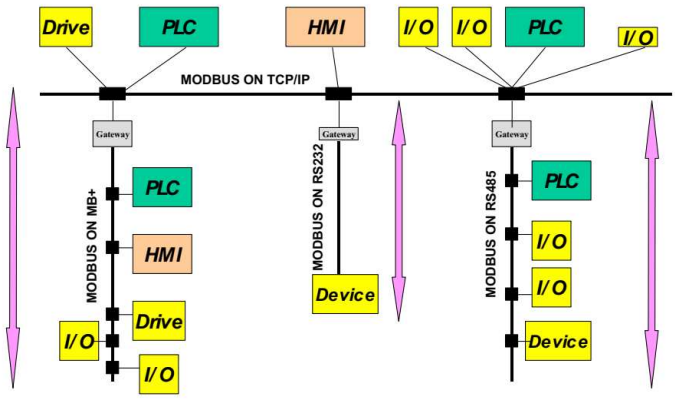


Figure 2: Example of MODBUS Network Architecture

From: **MODBUS APPLICATION PROTOCOL SPECIFICATION, V1.1b3** (http://www.modbus.org)