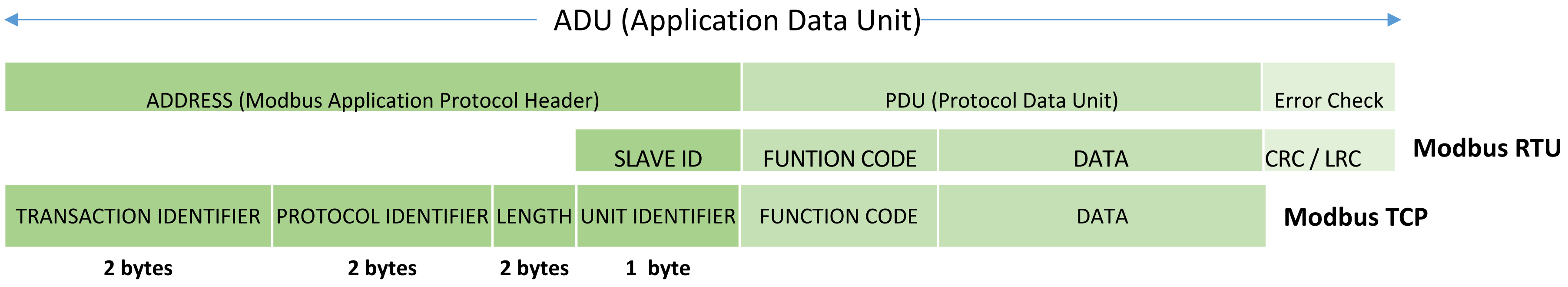
Modbus Protocol Stack

OSI Layers	Modbus RTU	Modbus Plus	Modbus TCP/IP
Application Layer (L7)	Modbus Application Layer		
Presentation Layer (L6)			
Session Layer (L5)			
Transport Layer (L4)			TCP (Port 502)
Network Layer (L3)			IP
Link Layer (L2)	Master / Slave	Modbus+ / HDLC	Ethernet II / 802.3 / MAC / LLC
Physical Layer (L1)	RS-232 / RS-485	Physical Layer	Ethernet Physical Layer

Should be Known

ADU	Application Data Unit
HDLC	High level Data Link Control
MB	MODBUS Protocol
MBAP	MODBUS Application Protocol
PDU	Protocol Data Unit
Modbus Master	Modbus Client
Modbus Slave	Modbus Server

General Modbus Frame



ADU (Application Data Unit)

ADDRESS (Modbus Application Protocol Header) | PDU (Protocol Data Unit) | Error Check

SLAVE ID | FUNTION CODE | DATA | CRC / LRC — **Modbus RTU**

TRANSACTION IDENTIFIER | PROTOCOL IDENTIFIER | LENGTH | UNIT IDENTIFIER | FUNCTION CODE | DATA — **Modbus TCP**

2 bytes | **2 bytes** | **2 bytes** | **1 byte**

RS232 / RS485 **ADU** = 253 bytes + Server address (1 byte) + CRC (2 bytes) = **256 bytes**.
TCP MODBUS **ADU** = 253 bytes + MBAP (7 bytes) = **260 bytes**.

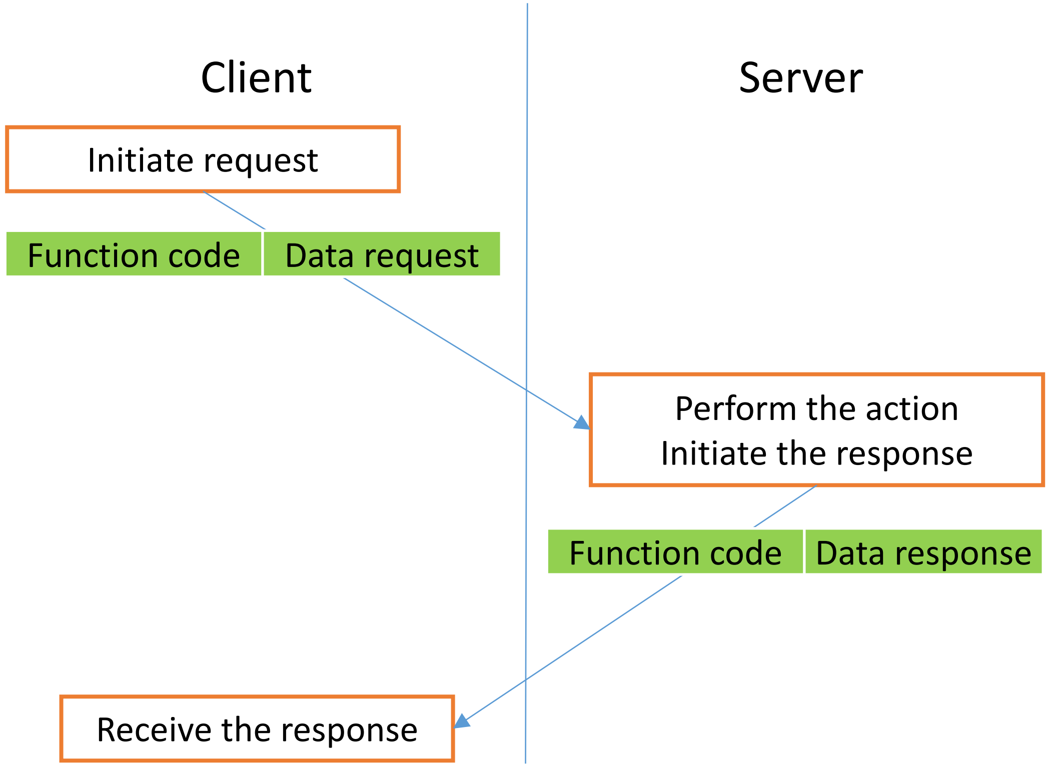
Modbus Data Types

Primary Tables	Object Type	Type of	Data References Type (Memory Block)	Description
Discrete Inputs	Single bit	Read-Only	1xxx	Provided by a Modbus device's I/O system.
Coils Outputs	Single bit	Read / Write	0xxx	Can be alterable by Master.
Input Registers	16-bit word	Read-Only	3xxx	Provided by a Modbus device's I/O system.
Holding Registers	16-bit word	Read / Write	4xxx	Can be alterable by Master.

by Seda Narli

Modbus Transaction

Error Free



Client

Server

Initiate request

Function code | Data request

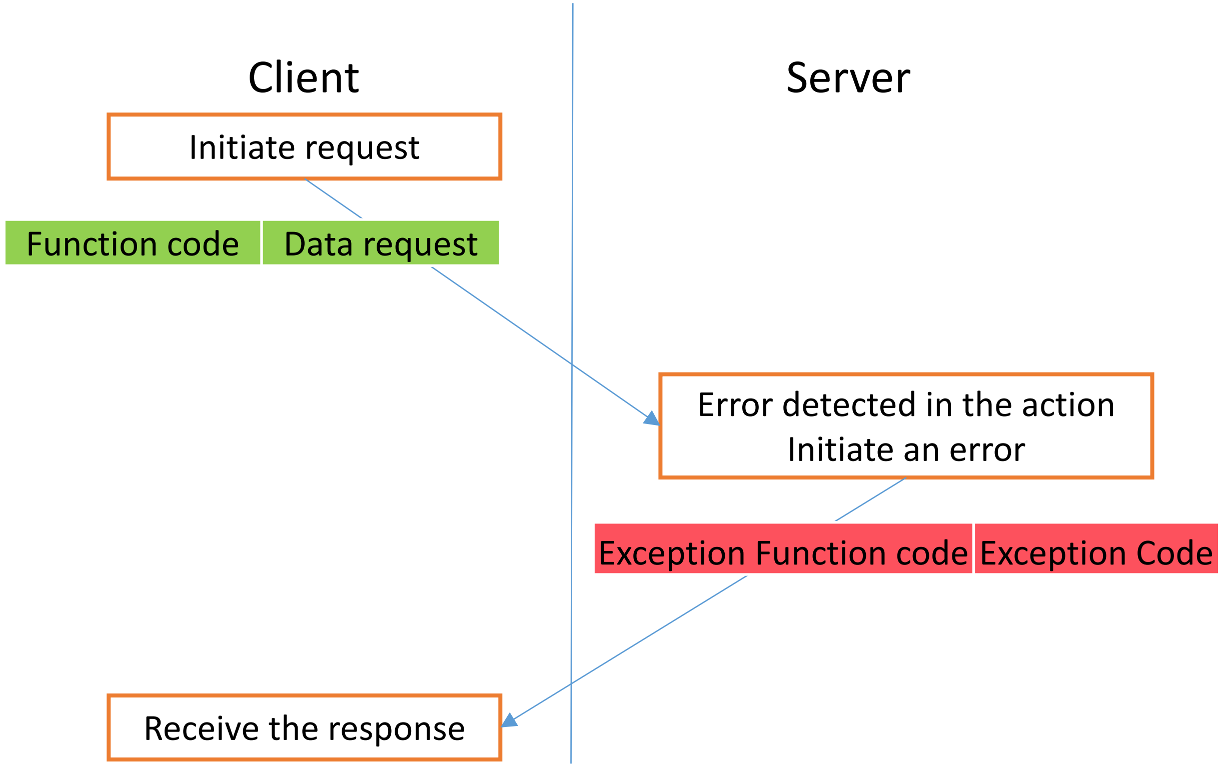
Perform the action
Initiate the response

Function code | Data response

Receive the response

The server simply echoes to the request the original function code.

Exception Response



Client

Server

Initiate request

Function code | Data request

Error detected in the action
Initiate an error

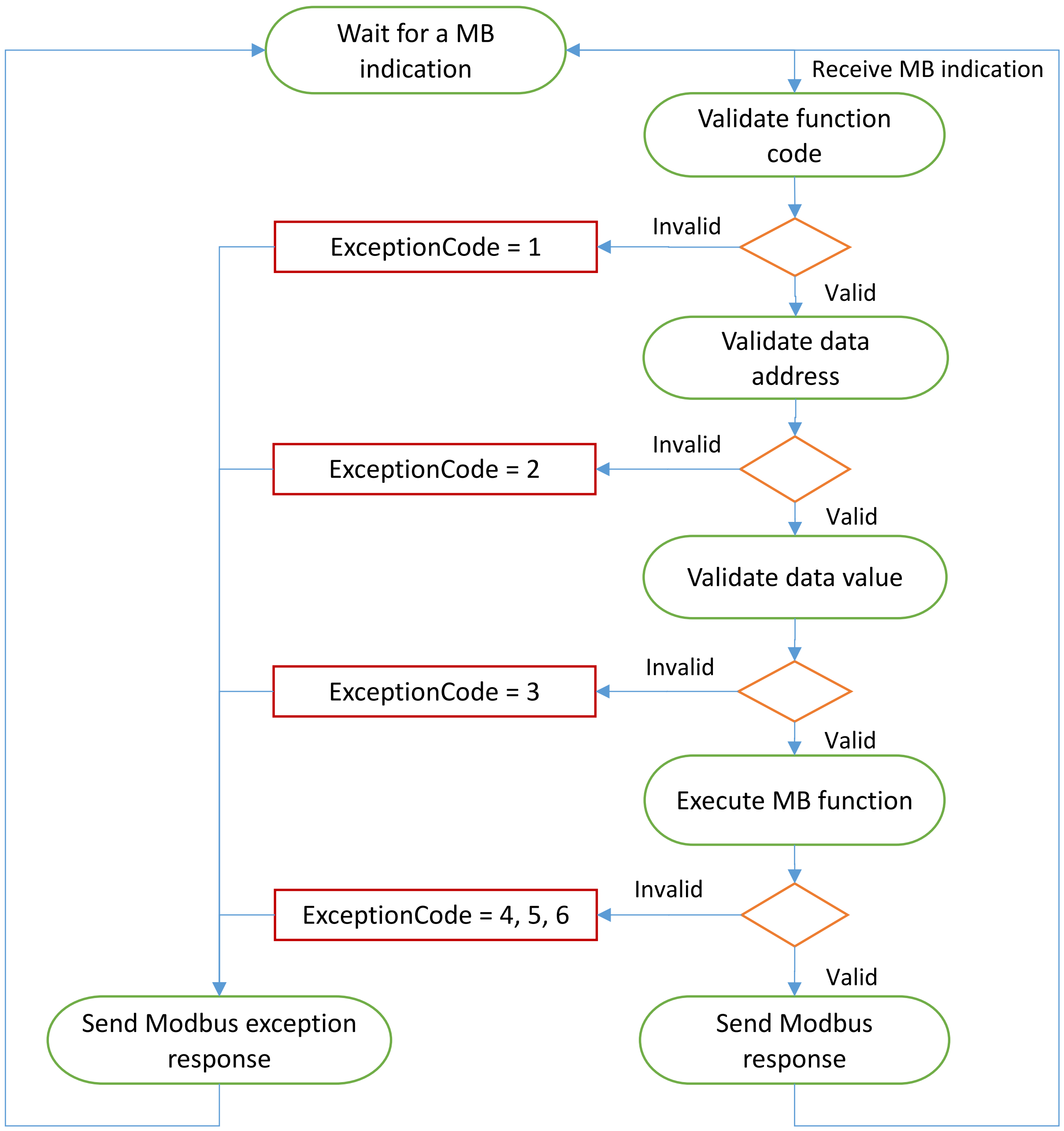
Exception Function code | Exception Code

Receive the response

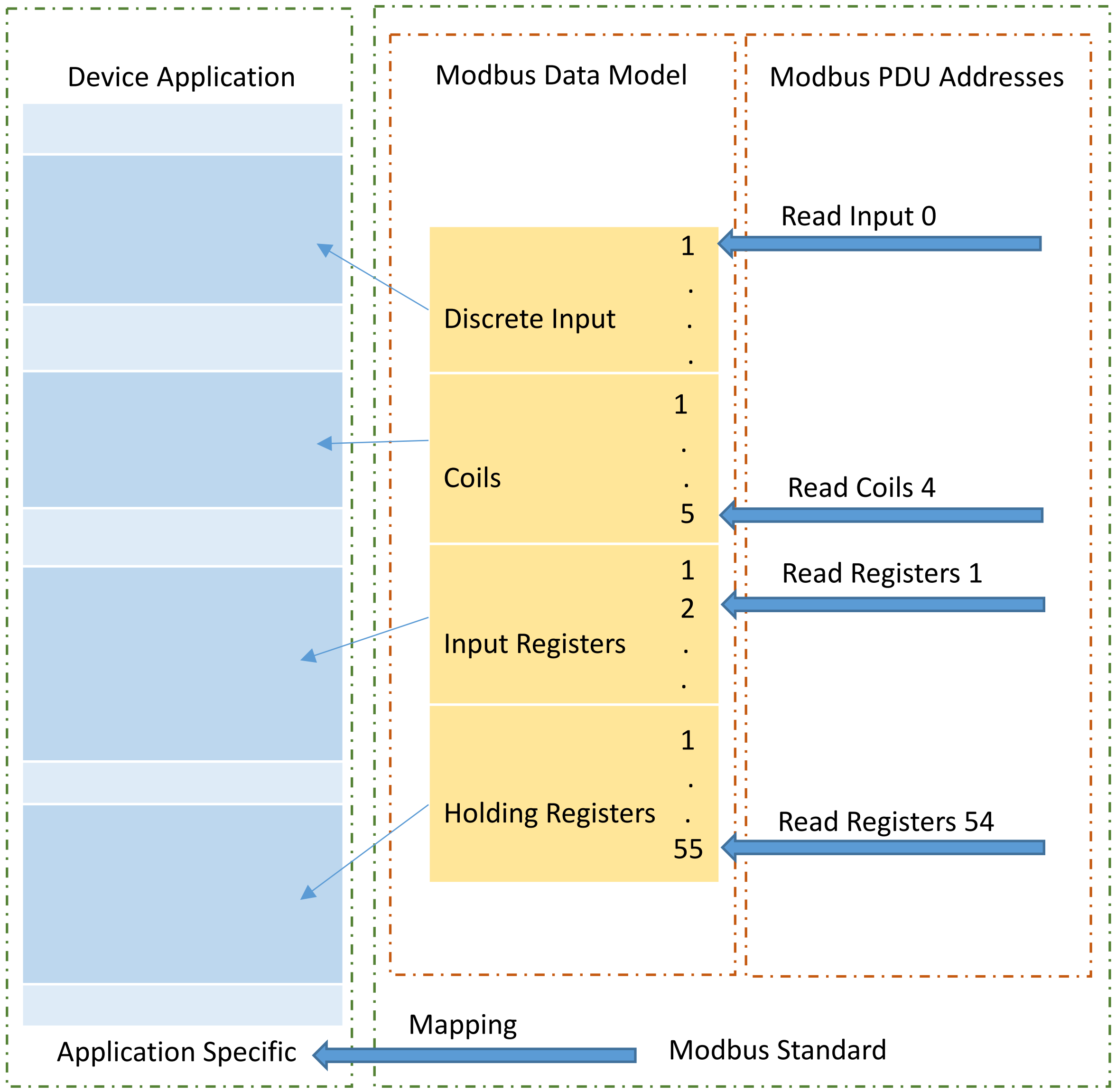
The server returns a code that is equivalent to the original function code from the request PDU with its most significant bit set to logic 1 (Hex 0x80).

by Seda Narli

Modbus Transaction State Diagram

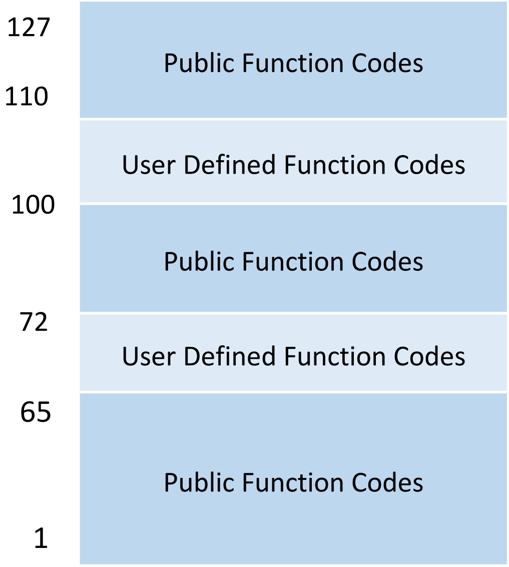


Modbus Addressing Model



by *Seda Narli*

Modbus Function Code Categories



Modbus Function Codes

Code	Hex	Function	Operation	Type	
01	01	Read Coils	Reads the bit data (N bits)	Single Bit Access	Data Access
02	02	Read Discrete Inputs	Reads the bit data		
05	05	Write Single Coil	Writes the bit data (one bit)		
15	0F	Write Multiple Coils	Writes the bit data (N bits)		
03	03	Read Holding Registers	Reads the integer type/character type/status word floating-point type data (N words)	16 Bit Access	
04	04	Read Input Register	Reads the integer type/character type/status word floating-point type data		
06	06	Write Single Register	Writes the integer type/character type/status word floating-point type data (one word)		
16	10	Write Multiple Registers	Writes the integer type/character type/status word floating-point type data (N words)		
22	16	Mask Write Register	Modifies the contents of a specified holding register using a combination of an AND mask, and OR mask, and the register’s current contents.		
23	17	Read/Write Multiple Registers	One read operation and one write operation in a single Modbus transaction.		
24	18	Read FIFO queue	Reads the contents of a FIFO queue of register in a remote device.		
20	14	Read File Record	Performs a file record read.	File Record Access	
21	15	Write File Record	Performs a file record write.		
07	07	Read Exception Status	Reads contents of eight Exception Status outputs in a remote device.	Diagnostics	
08	08	Diagnostic (Serial Line)	Tests for the checking to communication and internal error.		
11	0B	Get Com event counter	Gets status word and an event count from the remote device’s comm event counter.		
12	0C	Get Com Event Log	Gets status word, event count, message count, and field of event bytes from remote device.		
17	11	Report Server ID (Serial Line)	Reads the current status and other information specific to a remote device.		

Modbus Exception Codes

Code	Name	Description
01	Illegal Function	Function code is not valid or implemented.
02	Illegal Data Address	Object address is not valid for the Slave.
03	Illegal Data Value	Writing value is not value valid for the addressed object.
04	Slave Device Failure	Fatal error ocurred during the requested operation.
05	Acknowledge	The slave device may return an Acknowledge response after receiving a function query.
06	Slave Device Busy	The slave device is busy processing a function or task.
08	Memory Parity Error	The extended file area failed to pass a consistency check.
0A	Gateway Path Unavalible	The gateway is misconfigured or overloaded.
0B	Gateway Target Device Failed to Respond	The device is not present on the network.

by Seda Narli

Modbus Sample Transaction

The following is an example of a Modbus request for obtaining the AO value of the holding registers from registers # 40108 to 40110 with the address of the device 17.

11 03 006B 0003 7687

Modbus RTU	Slave ID	Inquiry	CRC
Modbus RTU	11	03 006B 0003	7687
Modbus TCP	0001 0000 0006 11	03 006B 0003	
Modbus TCP	MBAP Header	PDU	
Modbus TCP	ADU, Application Data Unit		

Modbus TCP Request		
0001	Transaction identifier	Transaction Identifier
0000	Protocol identifier (will always be 0000 for the Modbus Protocol)	Protocol Identifier
0006	Length (6 bytes are followed)	Message Length
11	The device address (17 = 11 hex)	Unit Identifier
03	Function code (read Analog Output Holding Registers)	Function Code
006B	First address register (107 = 40108-40001 = 6B hex)	Data Address of the first register
0003	The number of required registers (read 3 registers 40108 by 40110)	The total number of registers

Modbus RTU Request	
11	Device address SlaveID (17 = 11 hex)
03	Function Code (read Analog Output Holding Registers)
006B	Address of the first register (40108-40001 = 107 = 6B hex)
0003	The number of required registers (reading 3 registers from 40108 to 40110)
7687	Checksum CRC

Modbus TCP Response		
0001	Transaction identifier	Transaction Identifier
0000	Protocol identifier	Protocol Identifier
0009	The length (9 bytes are followed)	Message Length
11	The device address (17 = 11 hex)	Unit Identifier
03	Function code (read Analog Output Holding Registers)	Function Code
06	The number of bytes later (6 bytes are followed)	Byte Count (2 * The number of required registers)
02	Value of the high register bit (02 hex)	Register value Hi (AO0)
2B	Early discharge value register (2B hex)	Register value Lo (AO0)
00	Value of the high register bit (00 hex)	Register value Hi (AO1)
64	Value of the low register bit (64 hex)	Register value Lo (AO1)
00	Value of the high register bit (00 hex)	Register value Hi (AO2)
7F	Early discharge value register (7F hex)	Register value Lo (AO2)

Modbus TCP Response with Error	
0001	Transaction Identifier
0000	Protocol Identifier
0006	Message Length
11	Device Address
83	Functional code with changed bit
02	Exception Code

References :

- *http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf*
- *https://ipc2u.com/articles/knowledge-base/detailed-description-of-the-modbus-tcp-protocol-with-command-examples/*

by Seda Narli