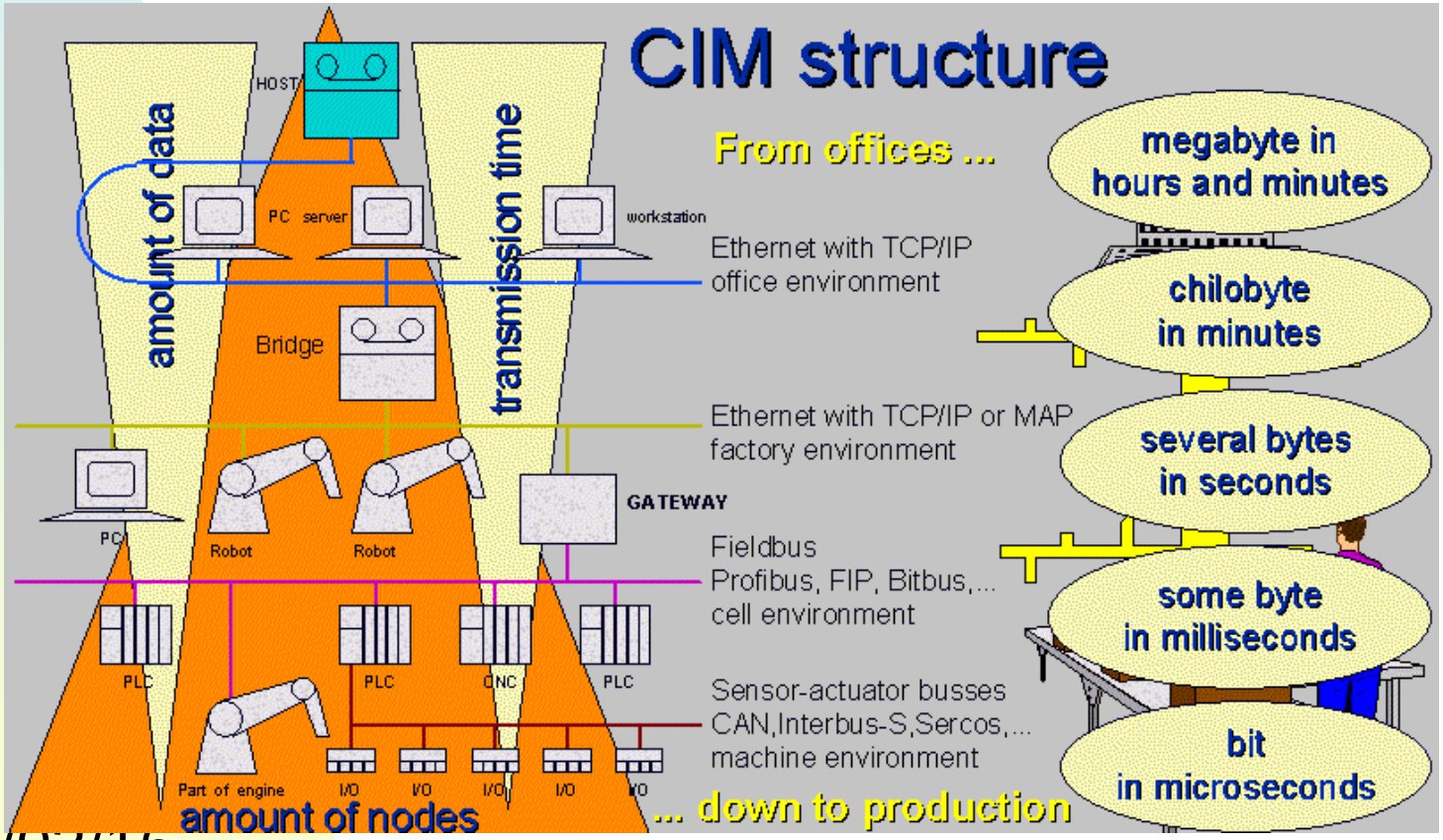
INSTITUTO FEDERAL
CEARÁ

Protocolo industrial aberto - MODBUS

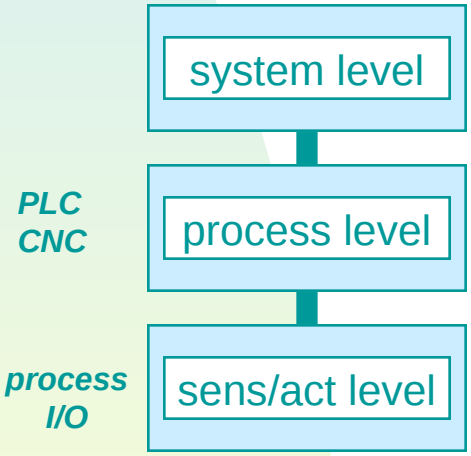
PEDRO URBANO B. DE ALBUQUERQUE

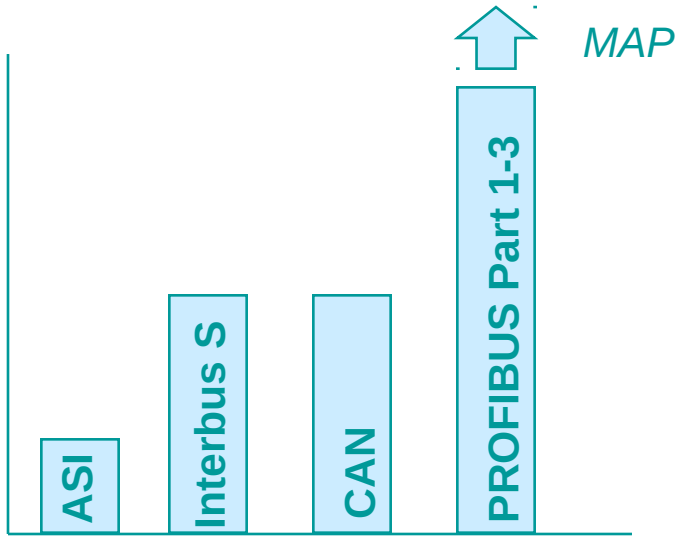
REDES INDUSTRIAIS

- Estrutura de um **CIM**



Fieldbus - Faixas de aplicação





BARRAMENTOS DE CAMPO

- **MODBUS – *MODICON*** (MODBUS-IDA.org community)
- **PROFIBUS – SIEMENS (**PROFIBUS User Organization)
- **MAP** (Manufacturing Automation Protocol) **- OSI**
- **EPA** (Enhanced Performace Archiiteture) **- MAP2.2**
- **WorldFIP (**Factory Information Protocol) **- AB +**
- **ISP** (Interoperable Systems Projet) **- SIEMENS +**
- **SP50 - ISA** (International Standards Association)
- **FOUNDATION** (Foundation Fieldbus)

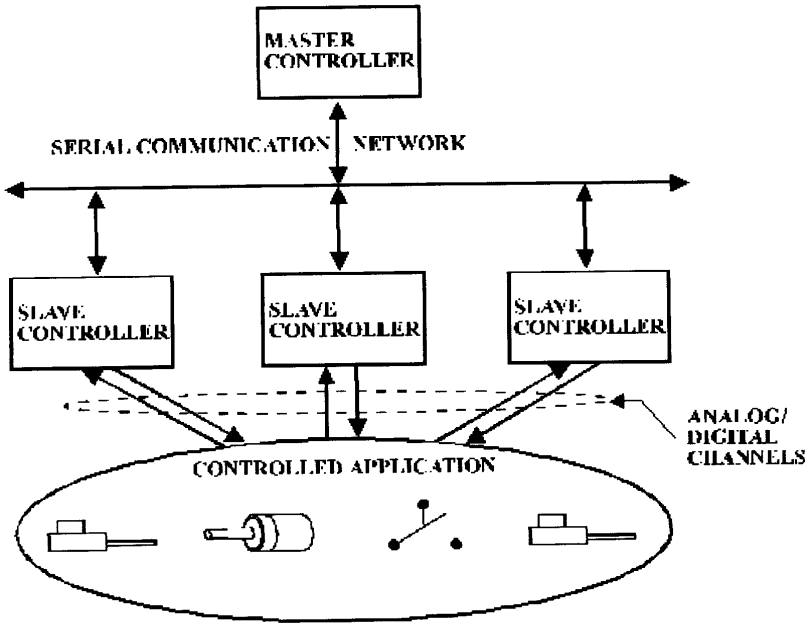
MODBUS

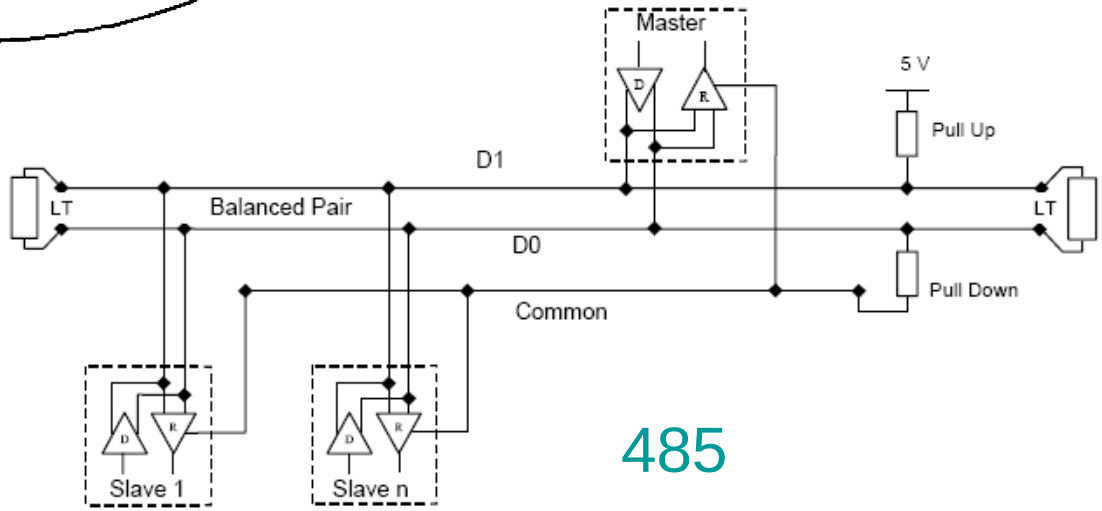
Abreviações:

ADU - *Application Data Unit*
HDLC - *High level Data Link Control*
HMI - *Human Machine Interface*
IETF- *Internet Engineering Task Force*
I/O - *Input/Output*
IP - *Internet Protocol*
MAC - *Medium Access Control*
MB - *MODBUS Protocol*
MBAP - *MODBUS Application Protocol*
PDU - *Protocol Data Unit*
PLC - *Programmable Logic Controller*
TCP - *Transport Control Protocol*

MODBUS

Arquitetura Distribuída hierárquica mestre escravo

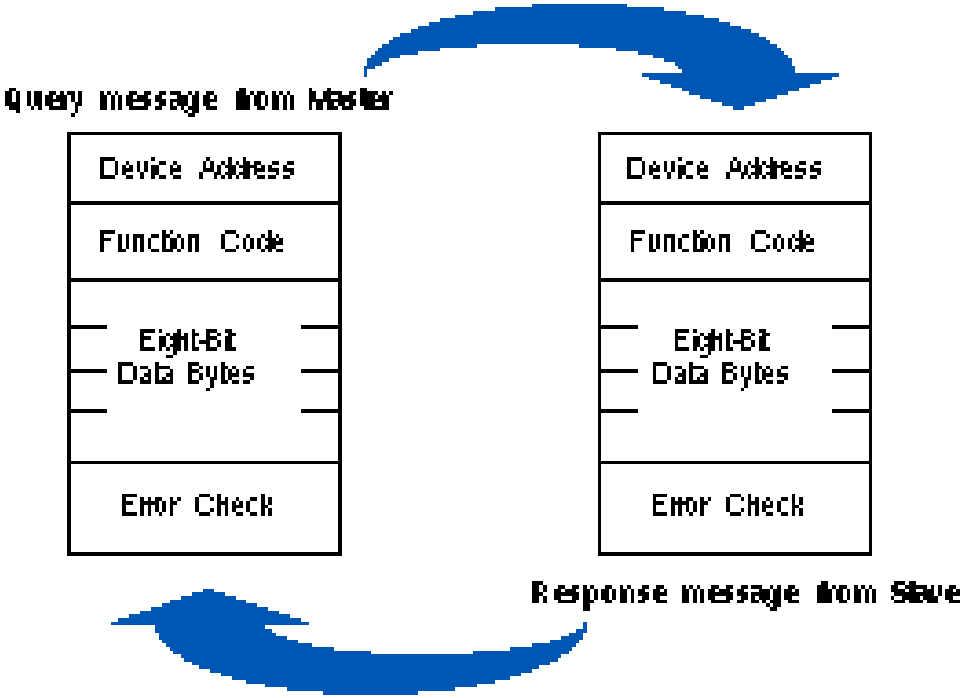




MODBUS

- Mestre-Escravo
- Modo Pergunta/resposta

Modos de menssagem



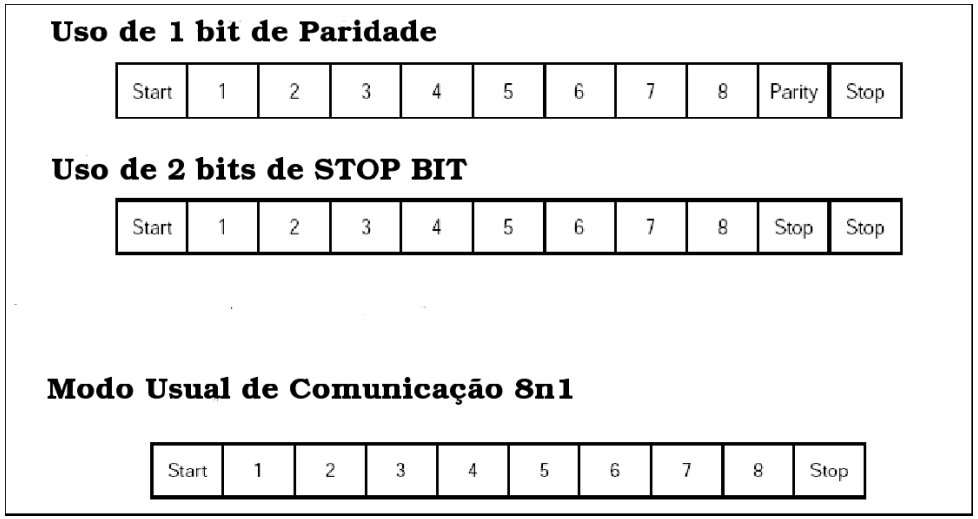
- MODBUS ASCII – transmite dados codificados em caracteres ASCII (*American Standard Code for Information Interchange*) de sete bits. Apesar de gerar mensagens legíveis por pessoas este modo consome mais recursos da rede.
- MODBUS RTU – neste modo os dados são transmitidos em formato binário de oito bits, no modo RTU(*Remote Terminal Unit*) - cada *byte* na mensagem contém dois caracteres hexadecimais de quatro *bits* cada.

CRC-16 (*Cyclic Redundancy Check)*

MODBUS

- Transmissão serial Assíncrona:

Vários modos de operação (Ex: 7n1, 8n1, 7o2,8o1);

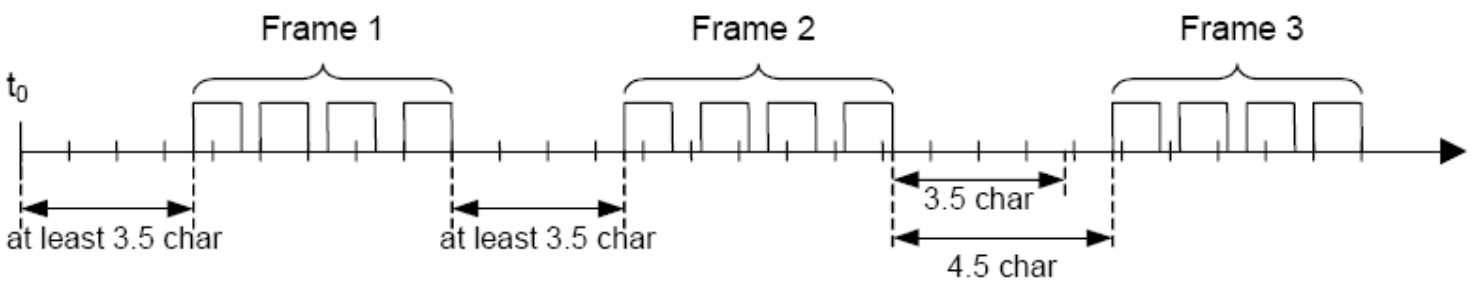


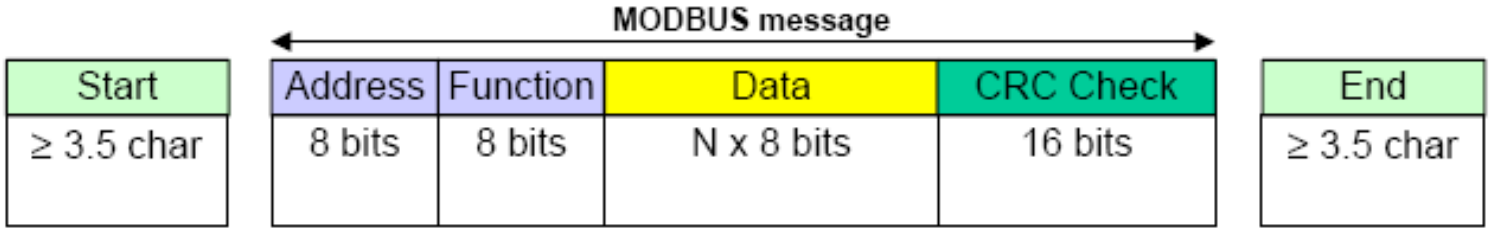
Exemplo: 11001100 - O Nº de bits 1 no frame são quatro.

Se a paridade usada for “par”, o bit de paridade será um 0, fazendo que o Nº de bits 1 seja um número par (quatro).
Se a paridade for “impar”, o bit de paridade será um 1, fazendo a quantidade de bits 1 impar (cinco).

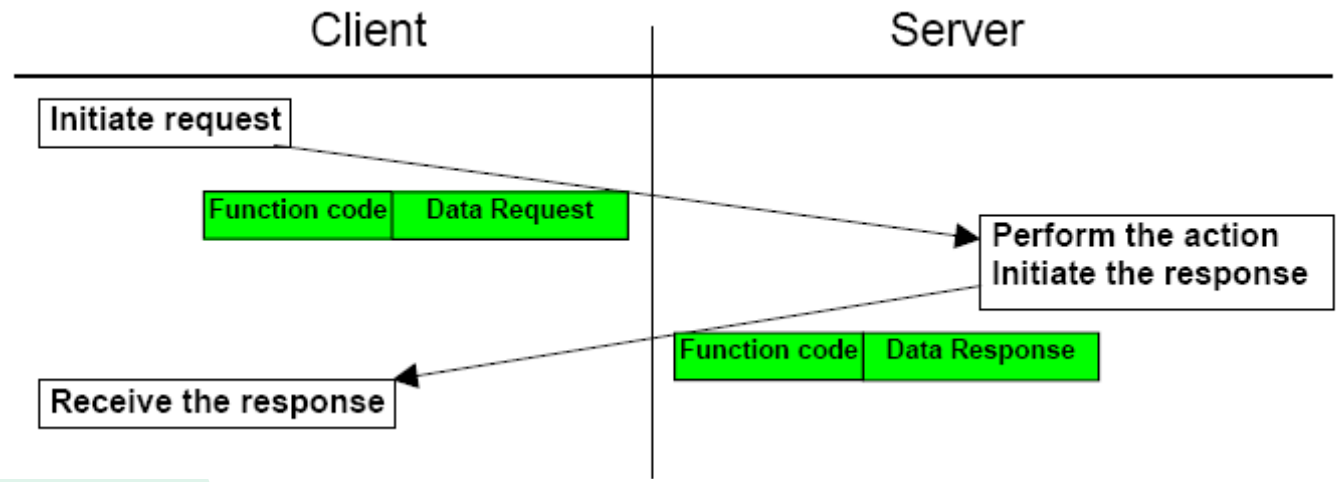
MODBUS – *Frame (telegrama, mensagem)*

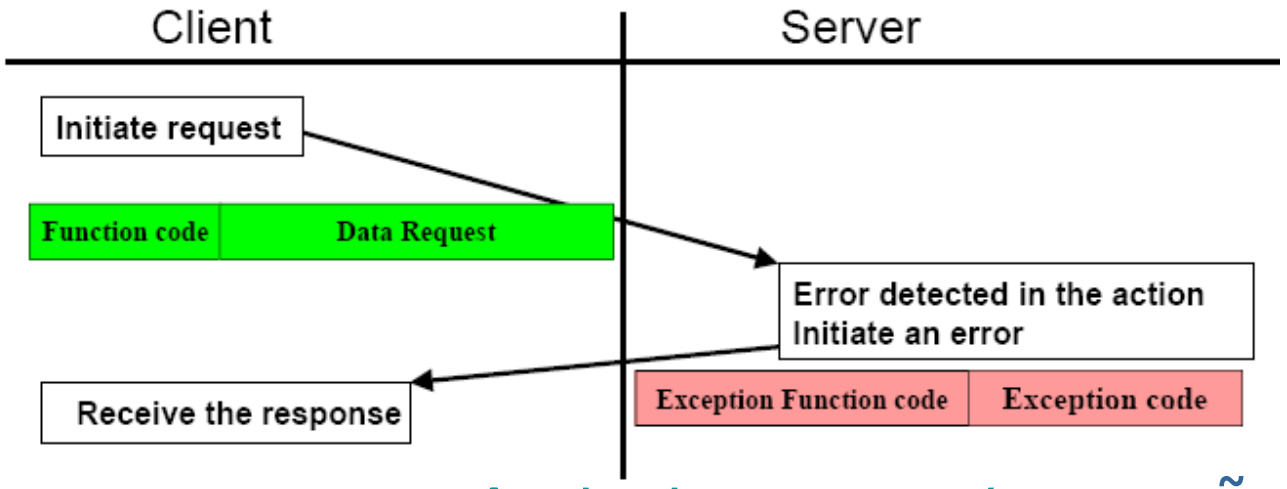
START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	N X 8 BITS	16 BITS	T1-T2-T3-T4





MODBUS

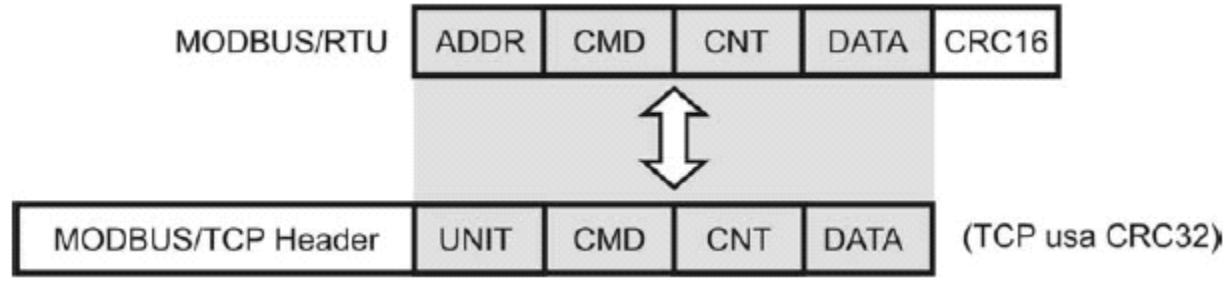




Ocorrência de um erro (exceção)

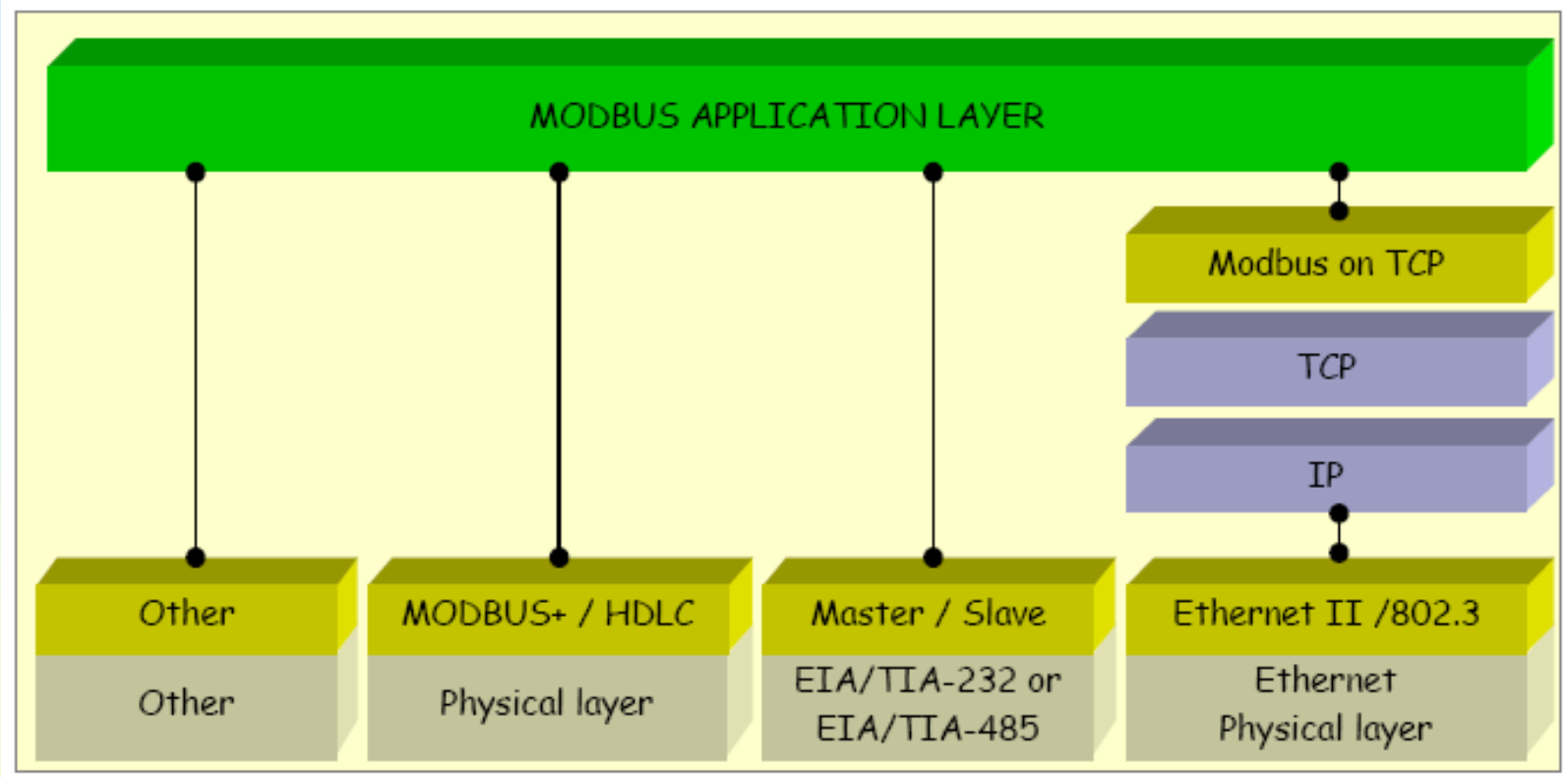
Variações do protocolo MODBUS:

MODBUS/TCP – Aqui os dados são encapsulados em formato binário em quadros para autilização do meio físico Ethernet (IEEE 802.3). Quando o MODBUS/TCP é utilizado, o mecanismo de controle e acesso é o CSMA-CD (próprio da rede Ethernet) e as estações utilizam o modelo cliente-servidor.

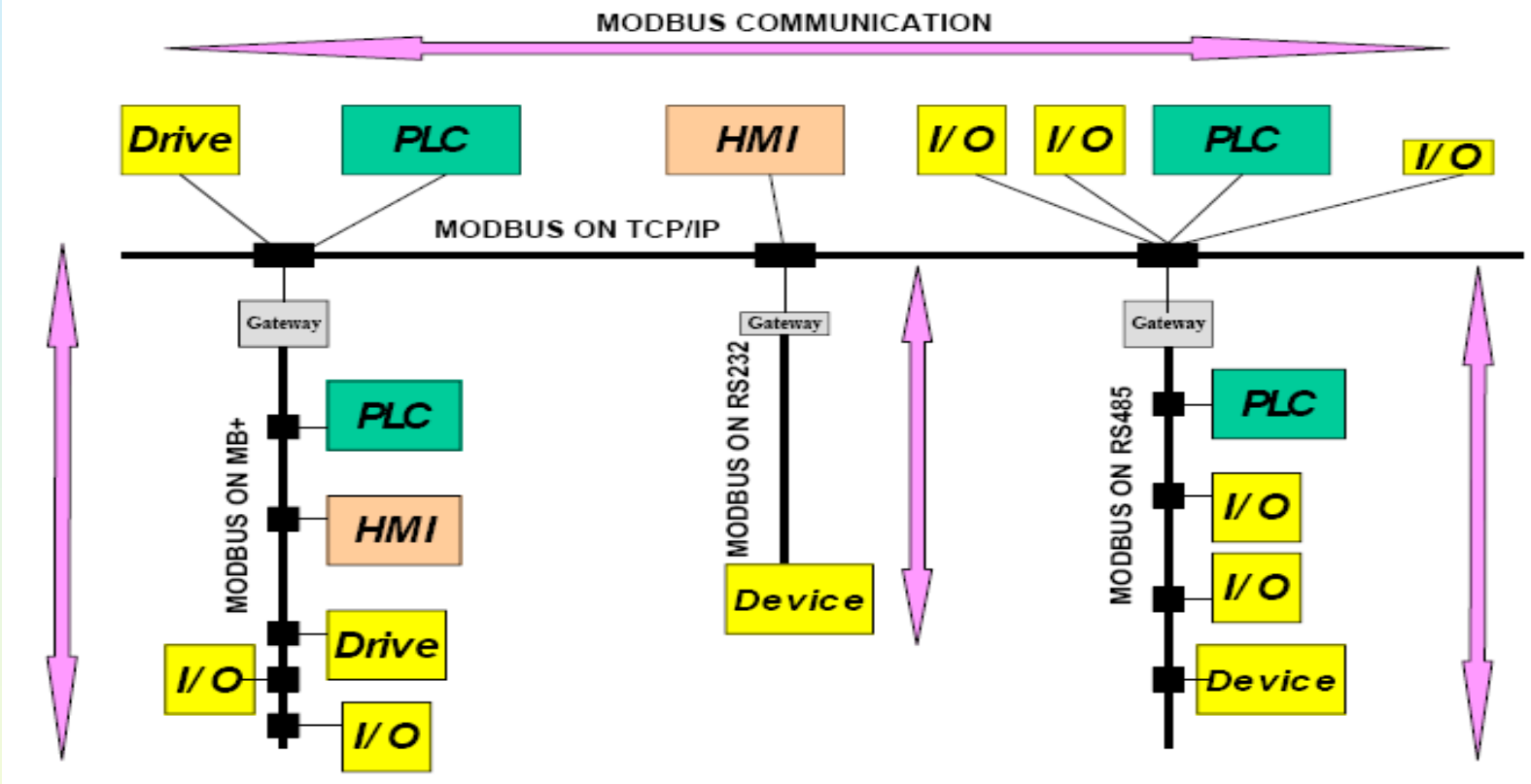


- **Modbus Plus** – Versão que possui vários recursos adicionais de roteamento, diagnóstico, endereçamento e consistência de dados. Esta versão ainda é mantida sob domínio da Schneider Electric e só pode ser implantada sob licença deste fabricante.

Variações do protocolo MODBUS:



Variações do protocolo MODBUS:

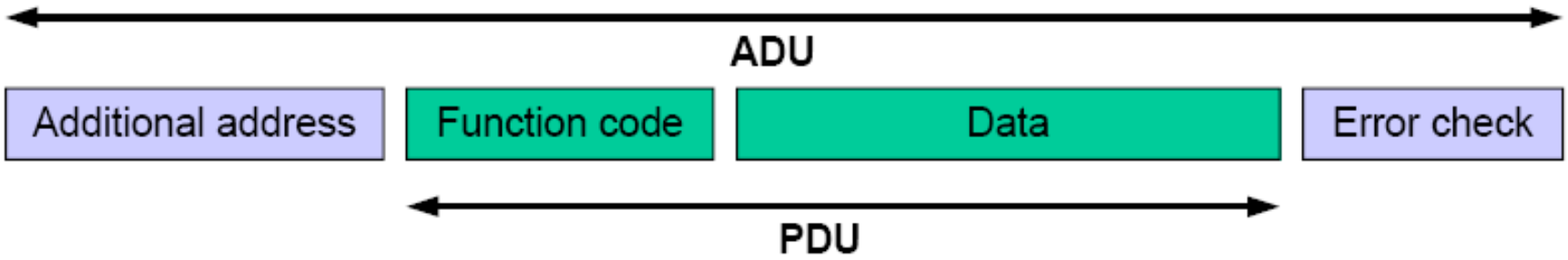


Mensagens/endereços *em* MODBUS

- **Uma mensagem MODBUS pode ser uma seqüência que varia desde alguns poucos bytes (menos de 10) até algumas centenas (máximo de 256 bytes).**

256bytes - Server address (1 byte) - CRC (2bytes) = **253 bytes**.

- **O endereço pode variar de 1 a 247, sendo possível, portanto, haver 1 mestre e 247 escravos.**



Principais funções dos mestres / escravos

- Mestre(master):
  - Comunicação com os outros níveis
  - Interpretação dos comandos
  - Sincronização do sistema
  - Coordenação
  - Cálculos

- Escravo(Slave):
  - Atuar em tarefas localizadas
  - Processamentodos sinais
  - Medidas
  - Manipular o evento conforme o predeterminado

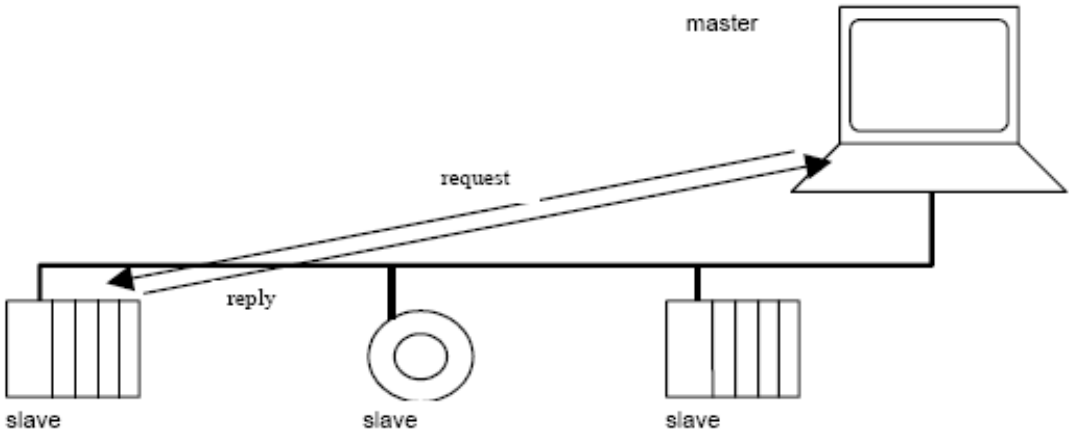
Principais funções para troca de mensagens:

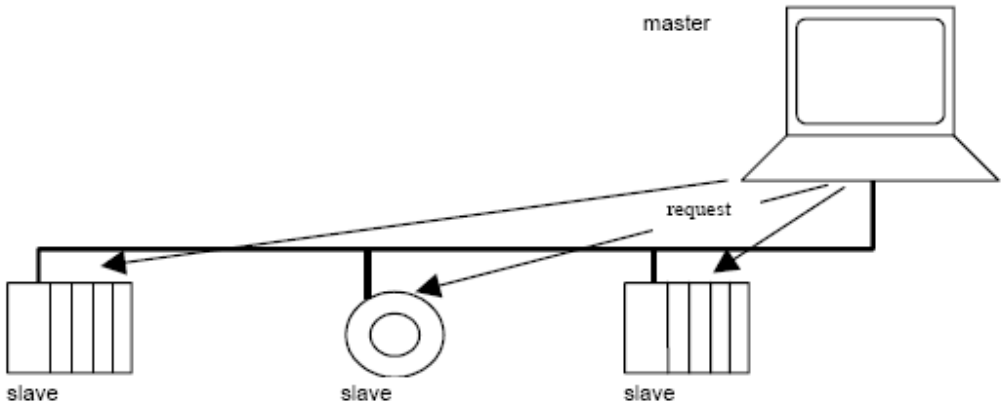
- 1. leitura de dados;
- 2. escrita de Dados e
- 3. difusão de dados (*Broadcast*):

- O protocolo Modbus define os seguintes tipos de dados:

  - Dados de 1 bit:
    - - Bobinas (*coils*): podem ser lidos do escravo ou escritos no escravo;
    - - Entradas (*inputs*): somente podem ser lidos do escravo;
- Dados de 16 bit (ou registros - *registers*):
  - - Retentivos (holding): podem ser lidos do escravo ou escritos no escravo;
  - - Entradas (*inputs*): somente podem ser lidos do escravo;

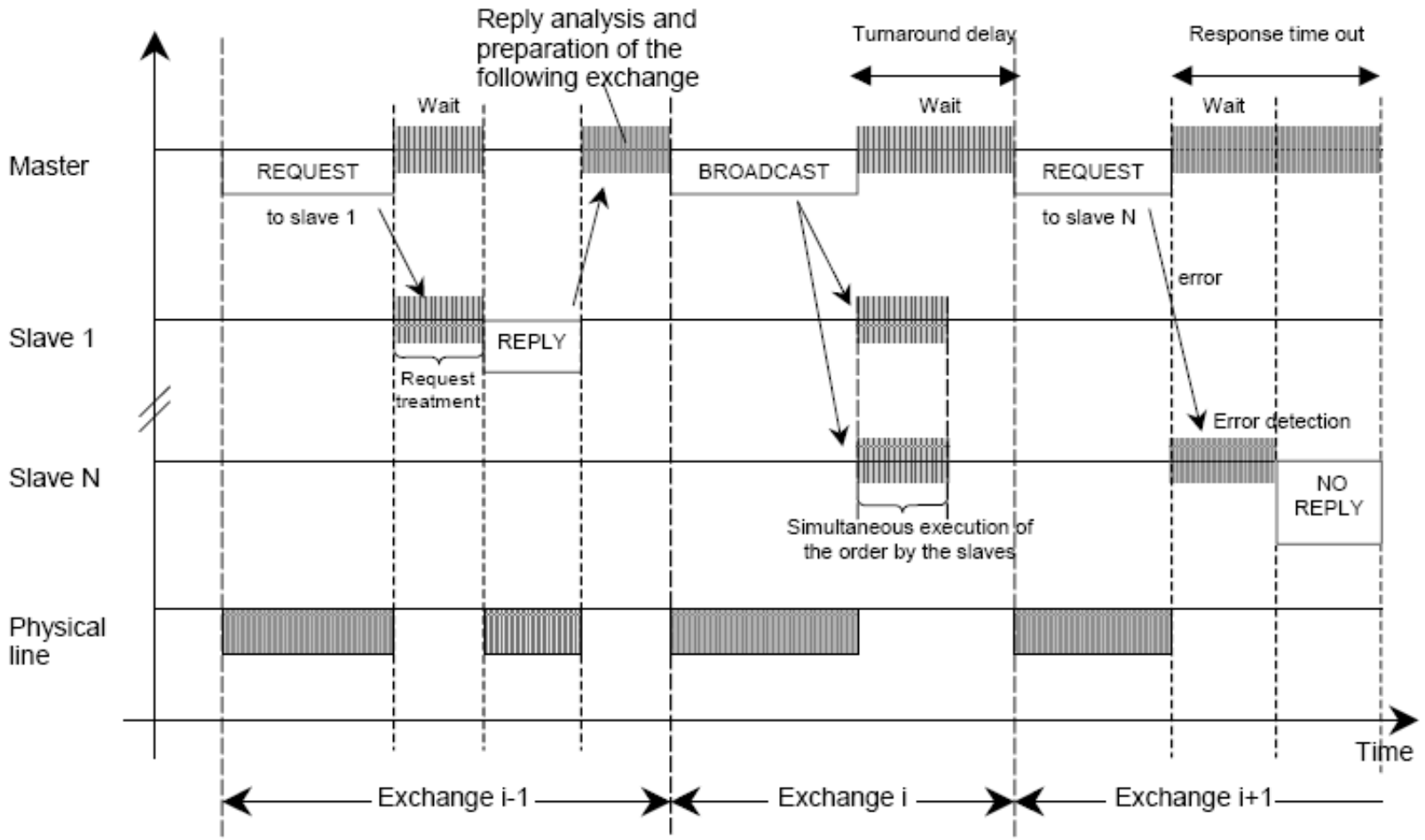
MODBUS - Modos de mensagens:





Difusão

MODBUS - Modos de mensagens:



Código hexadecimal das funções do MODBUS

Código Hex	Função
01	Leitura de um bit (bobina)
02	Leitura de n bits
03	Leitura de n palavras – Registros Retentivos
04	Leitura de n palavras – Registros de Entrada
05	Escrita de 1 bit – Simples Bobina
06	Escrita de 1 palavra – Preset um Registro
07	Leitura rápida de 1 byte – Status de Execução
0F	Escrita de n bits
10	Escrita de n palavras

MODBUS - Endereçamento:

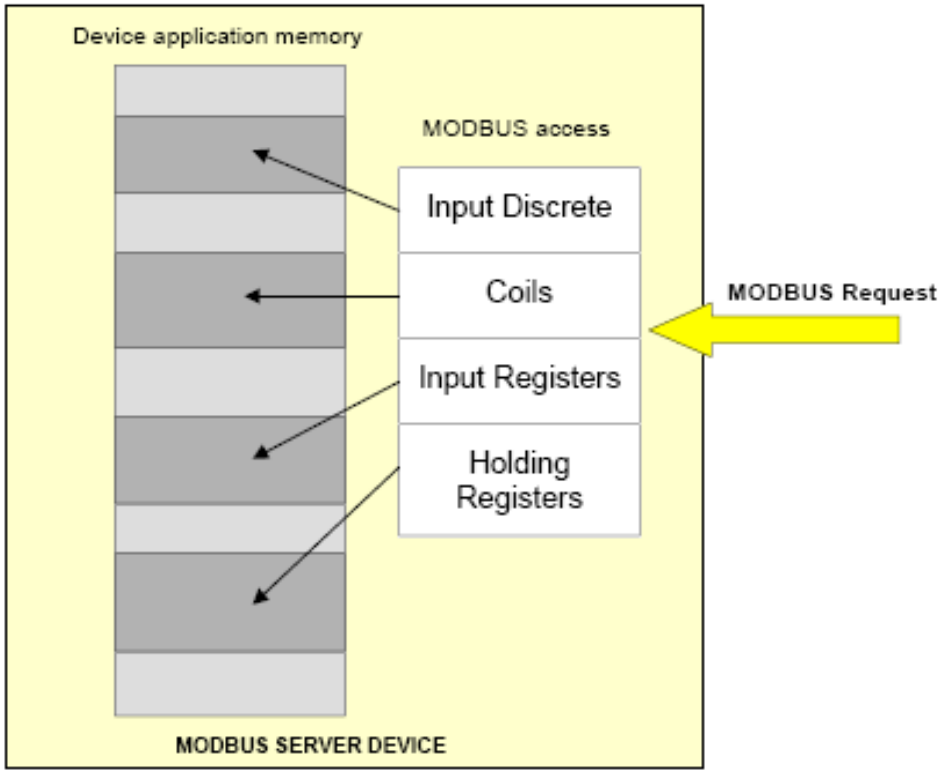
Endereçamentos lógico dos dados

Cada um dos tipos de dados definidos anteriormente pode ter até 9999 operandos ou variáveis. Cada operando deve ter um endereço lógico para diferenciá-lo dos demais operandos. Existe uma faixa de endereços destinada aos operandos de cada tipo de dados, conforme relacionado a seguir:

- **coils: 00001 a 09999;**
- **Inputs: 10001 a 19999;**
- **input registers 30001 a 39999;**
- **holding registers: 40001 a 49999.**

MODBUS - Endereçamento:

Endereçamentos lógico dos dados



MODBUS - mensagens:

Pergunta:

- O número do "escravo" (1 byte), que designa o destinatário da mensagem;
- O código da função a realizar (1 byte), que designa um comando de escrita ou leitura sobre os escravos;
- O endereço respectivo (2 bytes), que designa a posição de memória (endereço inicial dos dados) do escravo;
  - Byte mais significativo;
  - Byte menos significativo;
- Os dados a transmitir (2 bytes), que designa o número de registros (dados) a ser lido do escravo;

  Byte mais significativo número de registros;

  Byte menos significativo número de registros;
- Uma palavra de controle (2 bytes), CRC-16

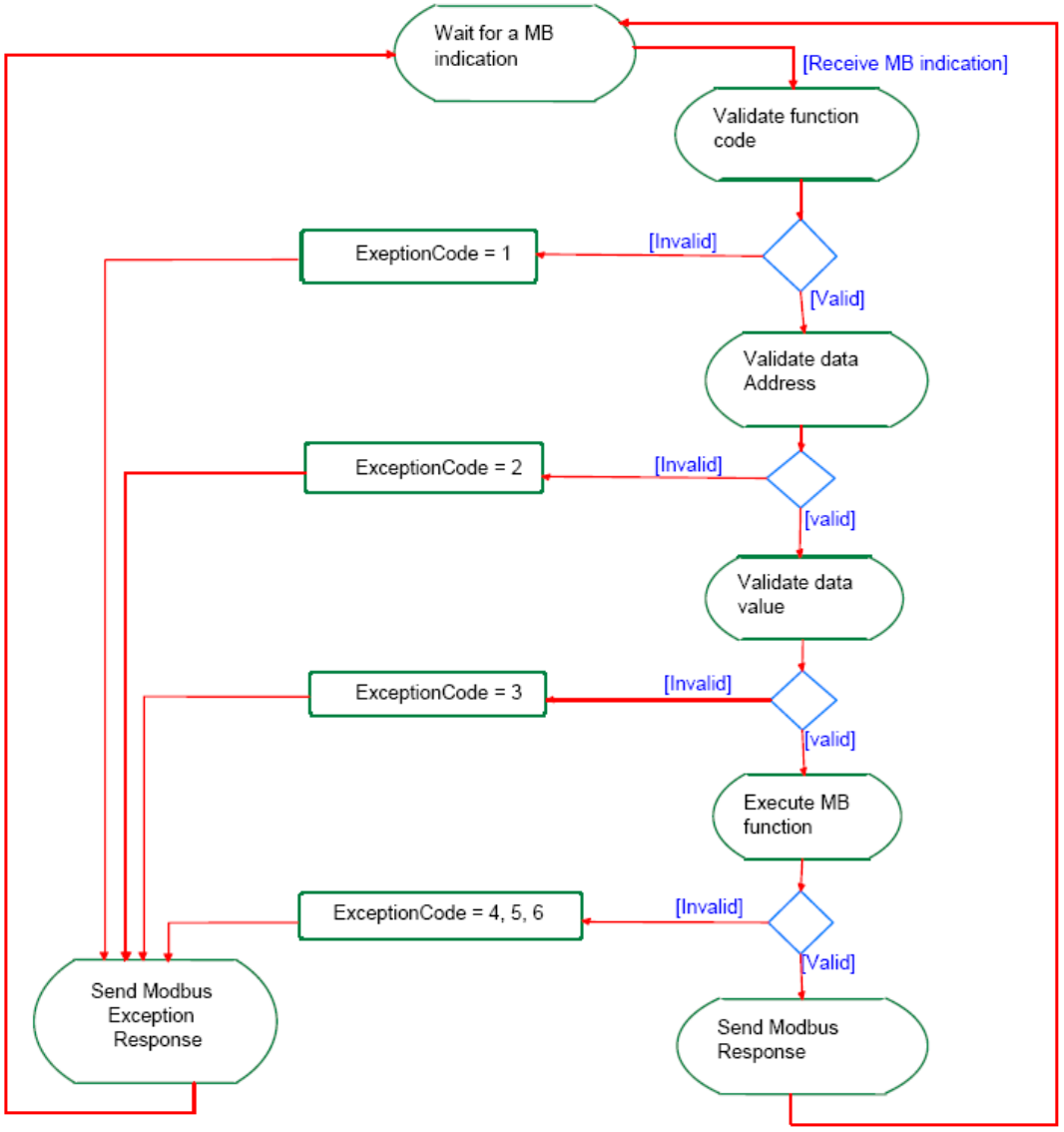
MODBUS - mensagens:

Resposta:

- O número do "escravo" (1 byte), ao qual se solicitou os dados;
- O código da função realizada (1 byte), que designa um comando de escrita ou leitura sobre os escravos;
- A quantidade de bytes da resposta (1 byte);
- Os dados solicitados, organizados da seguinte forma:
  - Byte mais significativo;
  - Byte menos significativo;
- Uma palavra de controle (2 bytes) -CRC-16

MODBUS - mensagens:

Resposta:



MODBUS - mensagens:

Exemplo de resposta com exceção:

The function code (01) – Função de *Read Output Status*.
Solicita o status de uma saída no endereço 1185

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		

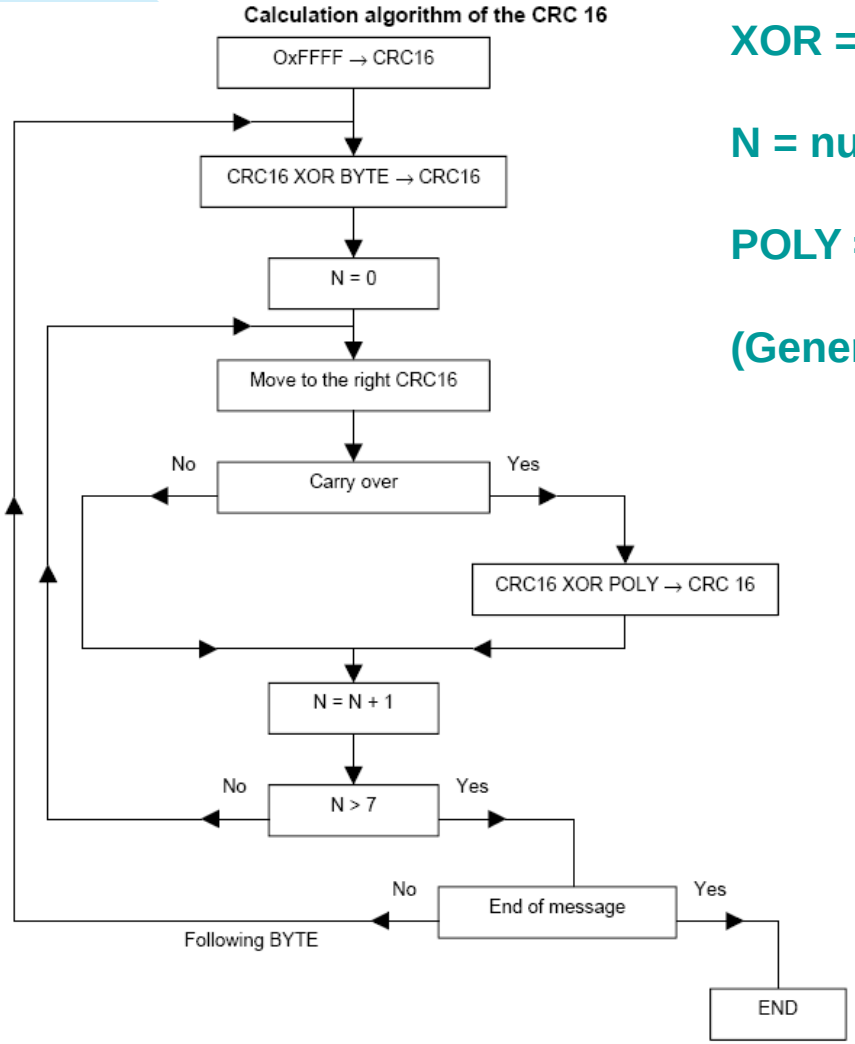
Pergunta *(Query) (Request)*

Field Name	(hex)	ASCII	RTU 8 - Bit Field
Slave Addres	06	0 6	0000 0110
Function	03	0 3	0000 0011
Starting Address HI	00	0 0	0000 0000
Starting Address LO	6B	6 B	0110 1011
Nº. of Registers HI	00	0 0	0000 0000
Nº. of Registers LO	03	0 3	0000 0011
Error Check		LRC (2 chars.)	CRC (16 bits)

• *Resposta (Response) (Replay)*

Field Name	**(hex)**
Slave Addres	06
Function	03
Byte Count	06
Data HI	02
Data LO	2B
Data HI	00
Data LO	00
Data HI	00
Data LO	63
Error Check	CRC (16 bits)

CRC - Algoritmo



XOR = exclusive or

N = number of information bits

POLY = CRC 16 = 1010 0000 0000 0001 (A001hex)

(Generating polynomial = $1 + x^2 + x^{15} + x^{16}$)

CRC - Algoritmo

Example of CRC calculation (frame 02 07)

CRC register initialization		1111	1111	1111	1111
XOR 1st character		0000	0000	0000	0010
		1111	1111	1111	1101
	Move 1	0111	1111	1111	1110|1
Flag to 1, XOR polynomial		1010	0000	0000	0001
		1101	1111	1111	1111
	Move 2	0110	1111	1111	1111|1
Flag to 1, XOR polynomial		1010	0000	0000	0001
		1100	1111	1111	1110
	Move 3	0110	0111	1111	1111 | 0
	Move 4	0011	0011	1111	1111 | 1
		1010	0000	0000	0001
		1001	0011	1111	1110
	Move 5	0100	1001	1111	1111 | 0
	Move 6	0010	0100	1111	1111 | 1
		1010	0000	0000	0001
		1000	0100	1111	1110
	Move 7	0100	0010	0111	1111 | 0
	Move 8	0010	0001	0011	1111 | 1
		1010	0000	0000	0001

CRC - Algoritmo

	1000	0001	0011	1110
XOR 2nd character	0000	0000	0000	0111
	1000	0001	0011	1001
Move 1	0100	0000	1001	1100 | 1
	1010	0000	0000	0001
	1110	0000	1001	1101
Move 2	0111	0000	0100	1110 | 1
	1010	0000	0000	0001
	1101	0000	0100	1111
Move 3	0110	1000	0010	0111 | 1
	1010	0000	0000	0001
	1100	1000	0010	0110
Move 4	0110	0100	0001	0011 | 0
Move 5	0011	0010	0000	1001 | 1
	1010	0000	0000	0001
	1001	0010	0000	1000
Move 6	0100	1001	0000	0100 | 0
Move 7	0010	0100	1000	0010 | 0
Move 8	0001	0010	0100	0001 | 0
			Most significant	least significant

The CRC 16 of the frame is then: 4112

?

OBRIGADO!!!!!

purbano@dpmengenharia.com.br

purbano@cefetce.br

INSTITUTO FEDERAL
CEARÁ