

6. Protocolos Digitais de Comunicação de Chão de Fábrica HART e Modbus

Diante de sistemas de automação compostos por múltiplos controladores de processos, CLPs, transmissores de campo e elementos finais de controle para o fechamento de malhas de controle e de intertravamento, nota-se a clara necessidade de troca de variáveis entre os instrumentos do sistema. Tradicionalmente a troca de dados entre sensores, controladores e atuadores é realizada de forma “ponto a ponto”, com a instalação de cabeamento dedicado entre cada sensor ou atuador e o respectivo canal em um cartão de entrada ou de saída do controlador correspondente.

Com o avanço da microeletrônica e da capacidade de processamento dos sistemas microprocessador, a partir do final da década de 1960 os fabricantes de instrumentos para automação industrial desenvolveram instrumentos com a capacidade de processamento digital local e, conseqüentemente, com a capacidade de comunicar-se com outros dispositivos por meio de protocolos de comunicação digitais.

Este capítulo trata de dois dos primeiros protocolos digitais surgidos para a comunicação em ambiente de chão de fábrica: o Modbus e o HART. Ambos os protocolos se firmaram como padrões de fato na indústria e hoje são considerados como “universais”, cada qual no seu setor, o HART no setor de indústrias de processo e o Modbus nas indústrias de manufatura.

Protocolo Modbus

O protocolo Modbus é um protocolo de aplicação para sistemas industriais, originalmente destinado a conectar CLPs. Foi desenvolvido no final da década de 60 pela empresa Modicon e desde 1979 é um padrão industrial de fato, com um grande número de fornecedores e de instrumentos compatíveis. Atualmente o protocolo oferece amplo suporte a redes do tipo Serial e Ethernet, pela porta reservada de sistema de número 502 do protocolo TCP/IP.

Este protocolo coordena sua troca de mensagens entre as estações pelo mecanismo PERGUNTA-RESPOSTA ou MESTRE-ESCRAVO e diferencia alguns serviços e funções definidos por um parâmetro dentro de cada mensagem denominado “Function Code”.

Os tipos mais comuns de sistemas Modbus são os seguintes:

- MODBUS Serial em RS485: é uma rede de baixo custo que usa o modelo master/slave para acesso ao meio físico com uma taxa de transmissão de 1.200 a 115 Kbits/s.

- MODBUS TCP/IP: usa o TCP/IP e Ethernet 10 Mbit/s ou 100 Mbits/s para transmitir mensagens MODBUS.

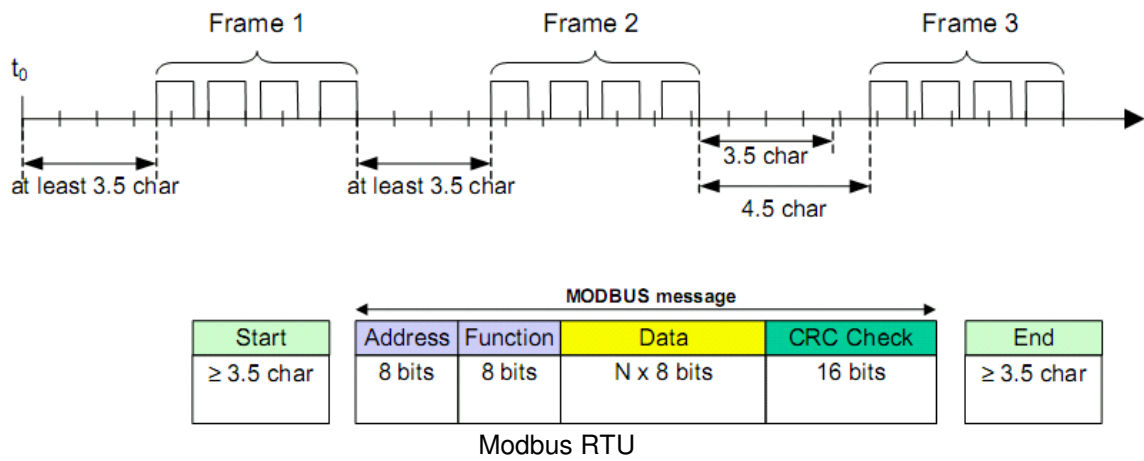
O protocolo MODBUS é encontrado ainda em 2 versões segundo o modo de transmissão:

- Modo de transmissão ASCII: Cada byte de 8 bits na mensagem é transmitido como 2 caracteres ASCII. Entre as mensagens deve-se respeitar um intervalo mínimo de 1 segundo.

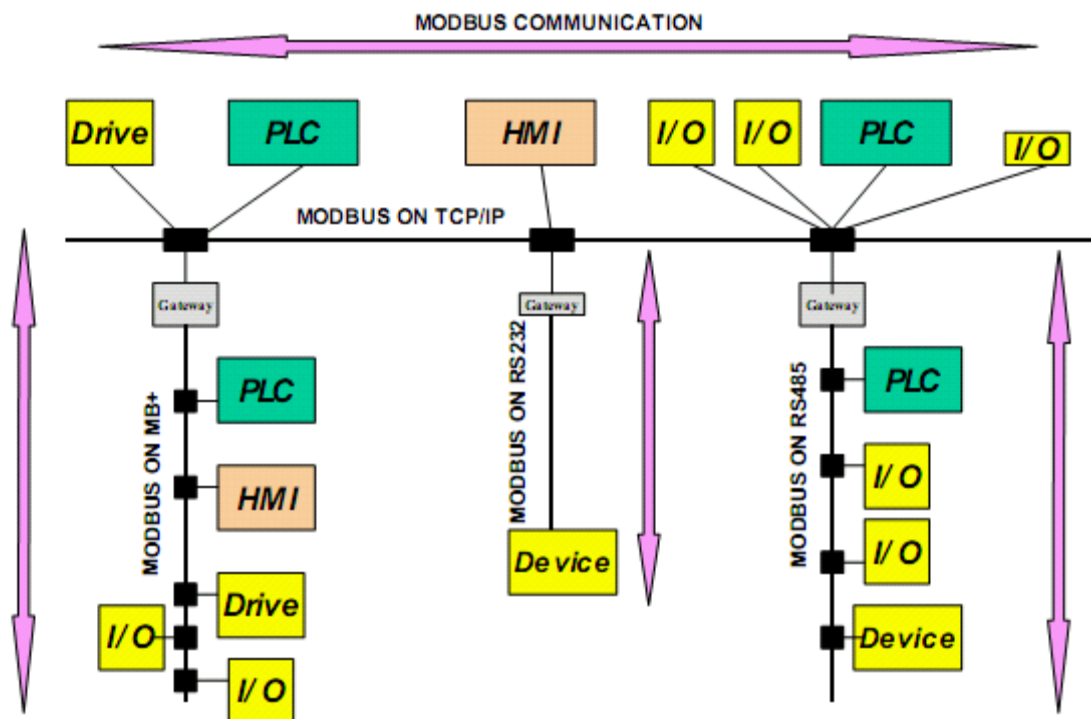
Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

Modbus ASCII

- Modo de transmissão RTU: Cada byte de 8 bits na mensagem é transmitido como 2 caracteres hexadecimais.



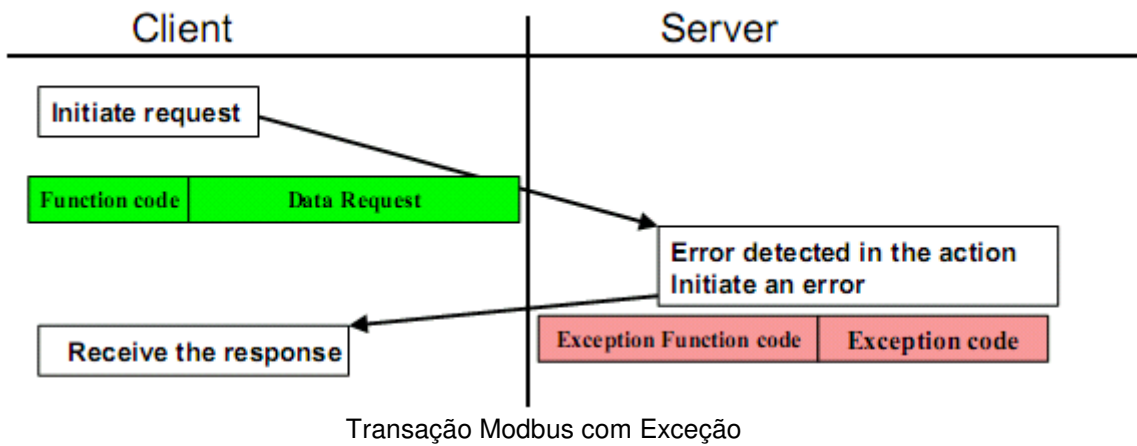
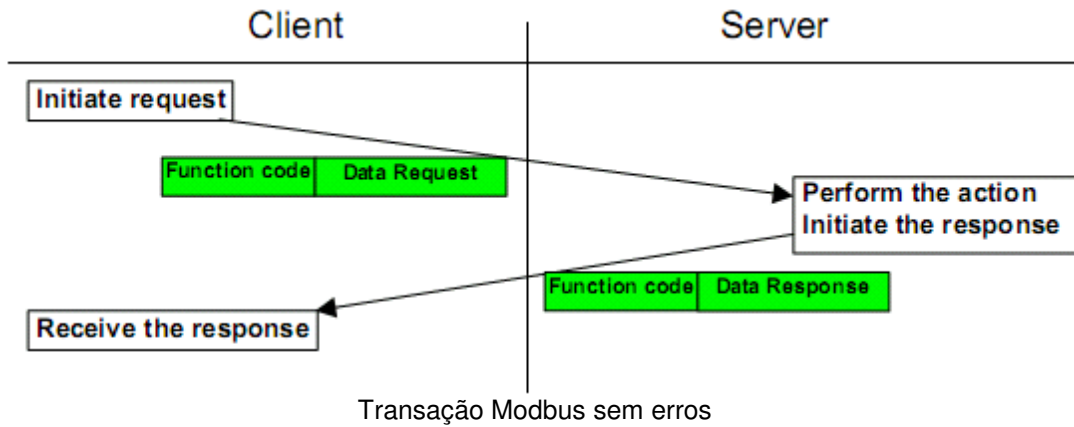
Contexto de Aplicação



O protocolo possui estrutura simples e define somente três tipos de mensagens:

- MODBUS Request PDU
- MODBUS Response PDU
- MODBUS Exception Response PDU

Estes tipos de mensagens relacionam-se conforme os diagramas a seguir dentro do modelo de comunicação Pergunta-Resposta.



Estrutura da Mensagem Modbus

A estrutura do frame ou mensagem é a mesma para a requisição (mensagens do master para o slave) e para a resposta (mensagens do slave para o master), segundo as figuras a seguir.

Modbus RTU



Silencio >= 3,5 caracteres

Modbus ASCII

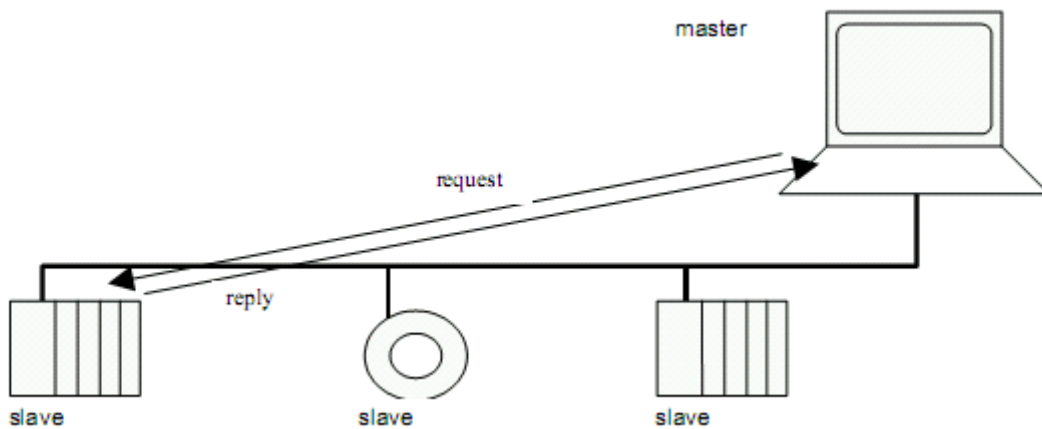


Campo Address ou Endereço:

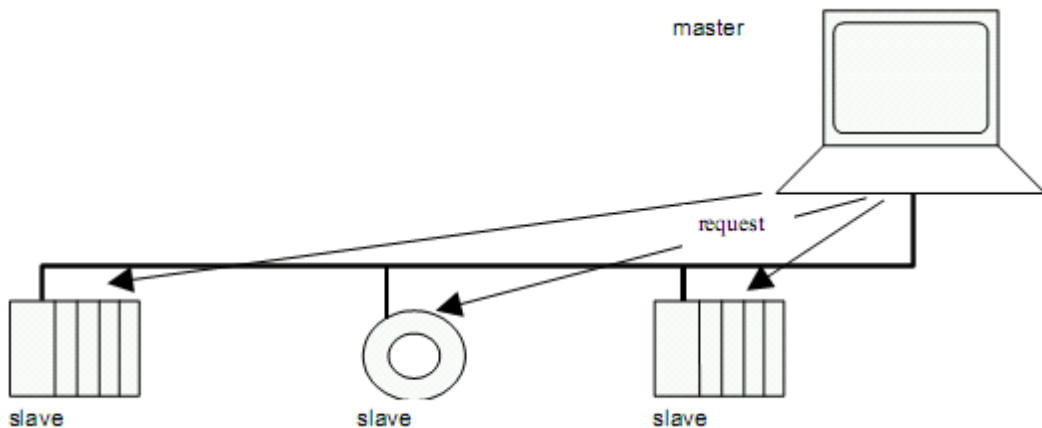
- Faixa de endereços válidos para escravos: 0 a 247 (decimal).

Dispositivos escravos são endereçados na faixa de 1 a 247. Os endereços de 248 a 255 são reservados.

O Valor 0 é reservado para mensagens de broadcast (sem resposta).



Tipo de comunicação em Unicast (endereçada)



Tipo de comunicação broadcast (um para todos), com endereço "0"

Na pergunta ou request, o mestre endereça o escravo colocando o endereço deste no campo address.

Na resposta ou response, o escravo coloca seu próprio endereço no campo address para informar ao mestre qual escravo responde.

Campo Function:

Códigos válidos na faixa de 1 a 255 (decimal).

Request : O campo function especifica qual tipo de ação o escravo deve executar.

Response : Para uma resposta normal, o escravo simplesmente repete o function code original. Para uma resposta excepcional, o escravo repete o function code original com seu bit mais significativo setado (=1).

Campo Data:

Códigos válidos na faixa de 0 a 255 decimal.

Request: O campo data contém informações adicionais que o escravo deve usar para realizar a ação definida pelo código da função. Pode incluir itens como registradores, endereços, quantidade de itens a serem manipulados, etc...

Response : Em ausência de erro, o campo dado contém o dado requisitado. Em caso de erro, o campo dado contém um código de exceção que o aplicativo do mestre pode utilizar para determinar ações a serem tomadas.

Campo Checksum:

Códigos válidos na faixa de 0 a 255 decimal.

O Modbus RTU usa CRC: Cyclycal Reduncy Check (2 byte)

O Modbus ASCII usa LRC: Longitudinal Redundancy Check (1 bytes)

Request : O checksum é calculado pelo mestre e enviado para o escravo.

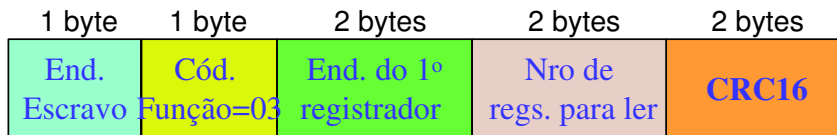
Response : O checksum é re-calculado pelo escravo e comparado com o valor enviado pelo mestre.

Se uma diferença é detectada, o escravo não responde ao mestre.

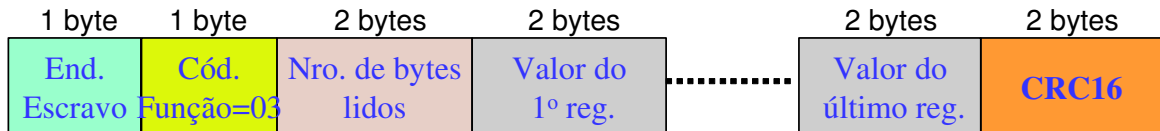
Exemplos:

1. Código de Função 03: Read Holding Registers

Pergunta:

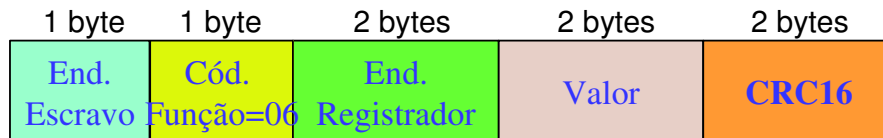


Resposta:

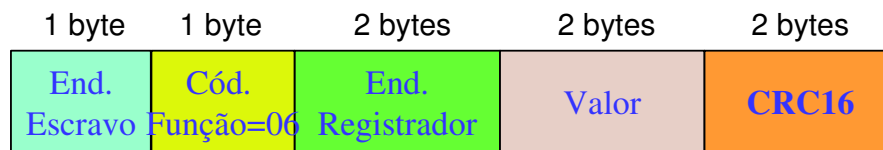


2. Código de Função 06: Write Single Register

Pergunta:

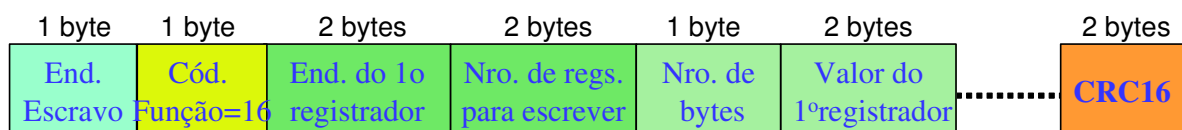


Resposta:



3. Código de Função 16: Write Multiple Registers

Pergunta:



Resposta:

1 byte	1 byte	2 bytes	2 bytes	2 bytes
End. Escravo	Cód. Função=16	End. do 1o registrador	Nro. de regs. para escrever	CRC16

Métodos de Checagem de Erros:

- Checagem de Paridade: Paridade par ou impar podem ser escolhidas para cada caracter.
- Checagem de Mensagem: Os algoritmos LRC ou CRC são aplicados para todas as mensagens.
- Fluxo contínuo: Toda a mensagem deve ser transmitida como um fluxo contínuo. Se um intervalo de silêncio (mais de que o tempo de 1.5 caracter no modo RTU ou 1 segundo no modo ASCII) aconteça durante a transmissão da mensagem, o receptor descarta a mensagem toda e assume que o próximo byte já é o campo de endereço de uma nova mensagem.

- Codificação dos Dados:

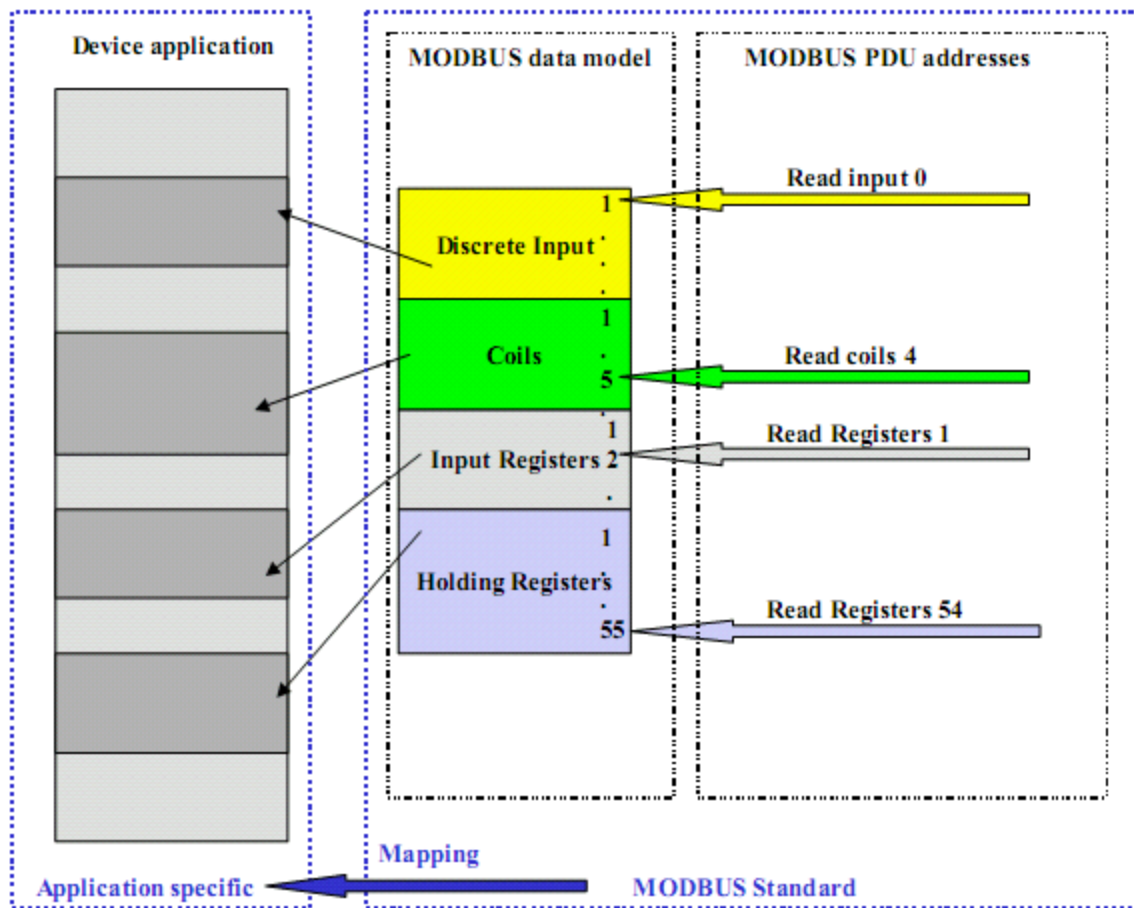
Ex.

Tamanho do Registro	Valor
16 - bits	0x1234 O primeiro byte enviado é 0x12 e depois 0x34 (formato big-Endian)

Em cada mensagem, cada dado é endereçado na faixa de 1 a 65535 dentro da tabela primária à qual pertence, compreendendo-se quatro blocos de dados (ou tabelas primárias) como na figura. A numeração dos dados dentro de cada bloco é seqüencial de 1 a n, sendo que as quatro tabelas primárias podem ser "sobrepostas" na área de memória do instrumento.

Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

O mapeamento entre o modelo de dados do Modbus e o mapeamento de cada instrumento é realizado previamente e de responsabilidade dos fabricantes. O endereço físico do dado dentro da memória do instrumento não deve ser confundido com sua referência ou endereço Modbus.



Modo de endereçamento Modbus

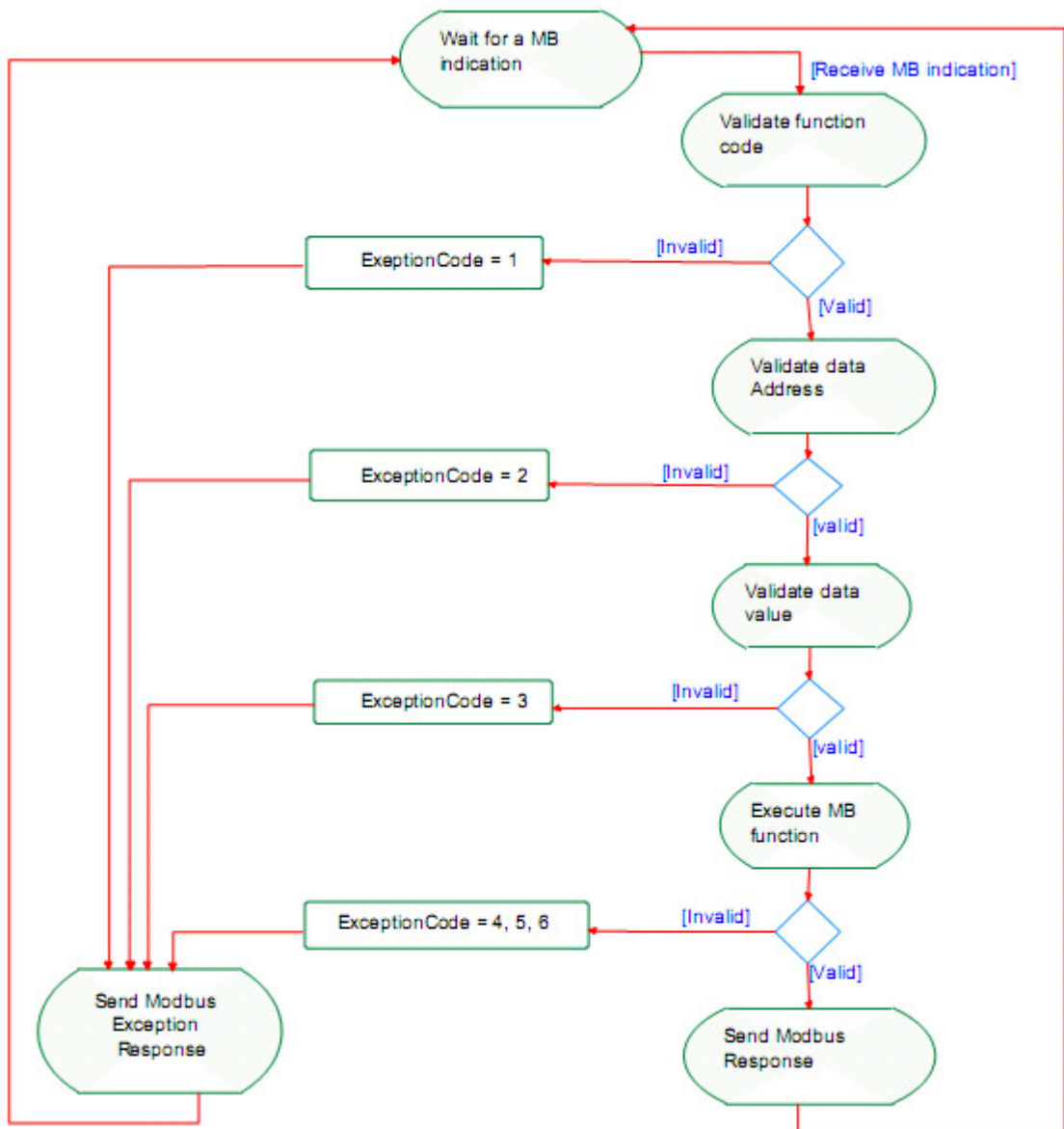


Diagrama de estados de uma transação Modbus

Relação de Códigos de Exceção para Respostas Negativas

MODBUS Exception Codes		
Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (address-wise at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 "Illegal Data Address" since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05	ACKNOWLEDGE	Specialized use in conjunction with programming commands. The server (or slave) has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client (or master). The client (or master) can next issue a Poll Program Complete message to determine if processing is completed.
06	SLAVE DEVICE BUSY	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long-duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.
08	MEMORY PARITY ERROR	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check.
		The server (or slave) attempted to read record file, but detected a parity error in the memory. The client (or master) can retry the request, but service may be required on the server (or slave) device.
0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Tipos de Function Codes:

- Públicos: uso público, único e bem documentado.

- User Defined (65 a 72 e 100 a 110 decimal)
- Reservados, usados apenas por algumas companhias para compatibilidade com produtos antigos, não são disponíveis para os usuários.

				Function Codes		(hex)	Section
				code	Sub code		
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
		Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access	Read File record		20		14	6.14
		Write File record		21		15	6.15
	Diagnostics	Read Exception status		07	00-18,20	07	6.7
Diagnostic		08	08	6.8			
Get Com event counter		11	0B	6.9			
Get Com Event Log		12	0C	6.10			
Report Slave ID		17	11	6.13			
Read device Identification		43	14	2B	6.21		
Other	Encapsulated Interface Transport		43	13,14	2B	6.19	

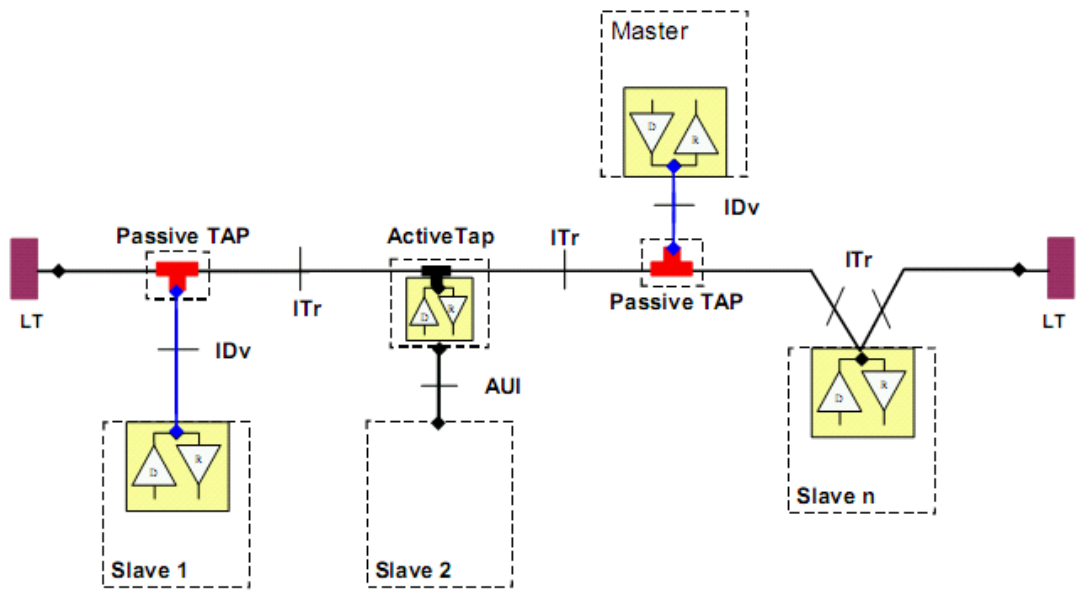
Definições dos Function Codes Públicos

Instalação e Projeto de Sistemas Modbus – Esquema Padrão do Modbus RS485

Tabela de Características Elétricas do Barramento Modbus RS485

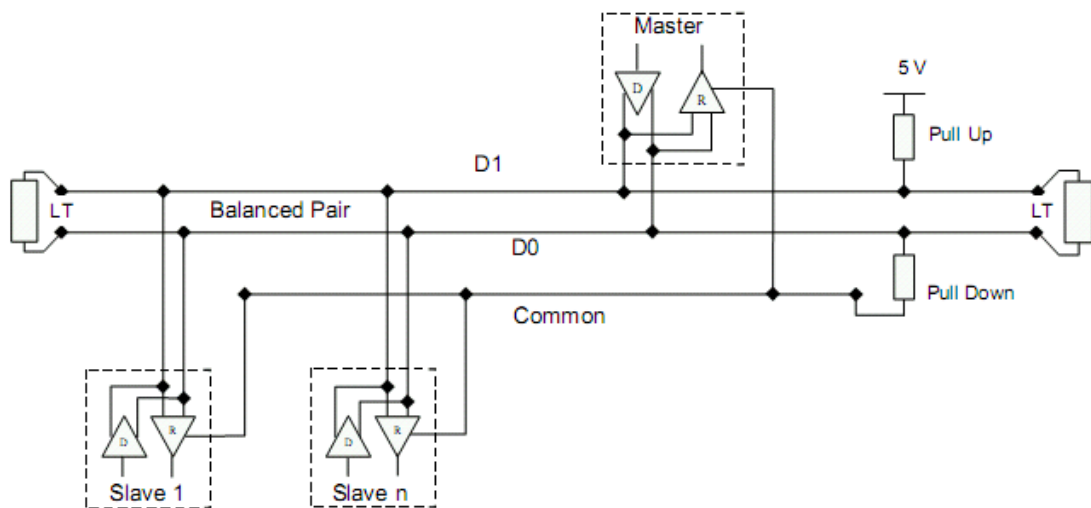
Comprimento máximo do barramento	1000 m @ 9600 bps com condutor 26AWG
Máximo número. de estações sem repetidores	32 (31 escravos)
Comprimento máximo de derivações	20 m para uma derivação 40 m divididos pelas n derivações (multi tap)
Aterramento do circuito comum	Preferencialmente em um ponto no mestre
Terminador de barramento	120 Ω - 0,25Wm em série com 1nF 10V
Polarização do Barramento	Sim (Pull Up/Pull Down @ R = 450~650 Ω)

Os transmissores são conectados na configuração RS485 a 2 fios, mas opcionalmente pode-se conectar a 4 fios (dois fios em par trançado para o sinal de comunicação bidirecional balanceado e dois para a alimentação em 24V). Em algumas redes conectam-se os transmissores em um barramento de 3 fios, um deles para a referência de tensão comum. A taxa típica de comunicação Modbus é de 9600 bits/s. Outras taxas de comunicação podem ser opcionalmente configuradas: 1200, 2400, 4800, 38400 bps, 56 Kbps ou 115 Kbps.



Infraestrutura do barramento serial

Detalhamento da topologia a dois fios:



<u>Required Circuits</u>		For device	<u>Required on device</u>	EIA/TIA-485 name	Description
on ITr	on IDv				
D1	D1	I/O	X	B/B'	Transceiver terminal 1, V1 Voltage (V1 > V0 for binary 1 [OFF] state)
D0	D0	I/O	X	A/A'	Transceiver terminal 0, V0 Voltage (V0 > V1 for binary 0 [ON] state)
Common	Common	--	X	C/C'	Signal and optional Power Supply Common

Pinagem típica de conectores RJ45 e Serial de 9 pinos para redes Modbus de 2 fios:

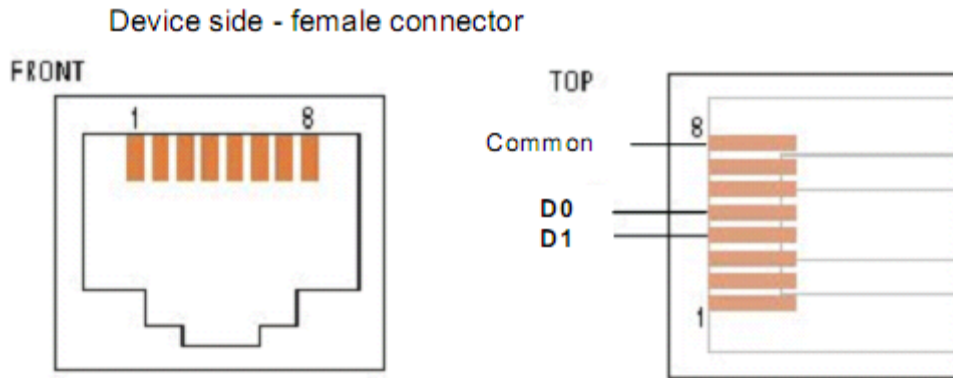
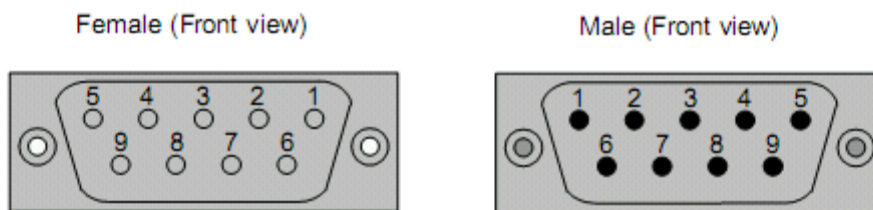


Figure 24: 2W- MODBUS on RJ45 connector (required pin-out)



Pin on RJ45	Pin on D9-shell	Level of requirement	IDv Circuit	ITr Circuit	EIA/TIA-485 name	Description for IDv
3	3	optional	PMC	--	--	Port Mode Control
4	5	required	D1	D1	B/B'	Transceiver terminal 1, V1 Voltage (V1 > V0 for binary 1 [OFF] state)
5	9	required	D0	D0	A/A'	Transceiver terminal 0, V0 Voltage (V0 > V1 for binary 0 [ON] state)
7	2	recommended	VP	--	--	Positive 5...24 V D.C. Power Supply
8	1	required	Common	Common	C/C'	Signal and Power Supply Common

Loops de Corrente 4 a 20 mA

O loop de corrente 4-20 mA é um padrão de sinalização de sensores muito robusto freqüentemente utilizado em indústrias de processo. Loops de corrente são ideais para a transmissão de dados pela sua maior imunidade (intrínseca por utilizar modulação em corrente, não em tensão elétrica) a ruídos eletro-magnéticos. Sua origem no cenário industrial como padrão de comunicação data de 1972 e ainda hoje muitos projetos de automação utilizam total ou parcialmente este tipo de comunicação analógica entre sensores, transmissores e CLPs ou controladores.

Em um loop de 4-20 mA, a corrente de sinalização do sinal flui por todos os componentes do circuito, independentemente das características elétricas de cada um. Cada um dos componentes é submetido, portanto, a uma queda de tensão pela passagem da corrente de sinalização sobre si. Entretanto, a corrente no circuito não é afetada pelas quedas de tensão em cada componente, desde que a tensão de alimentação da fonte do circuito for maior que a somatória de todas as quedas de tensão do circuito para a máxima corrente de sinalização de 20 mA.

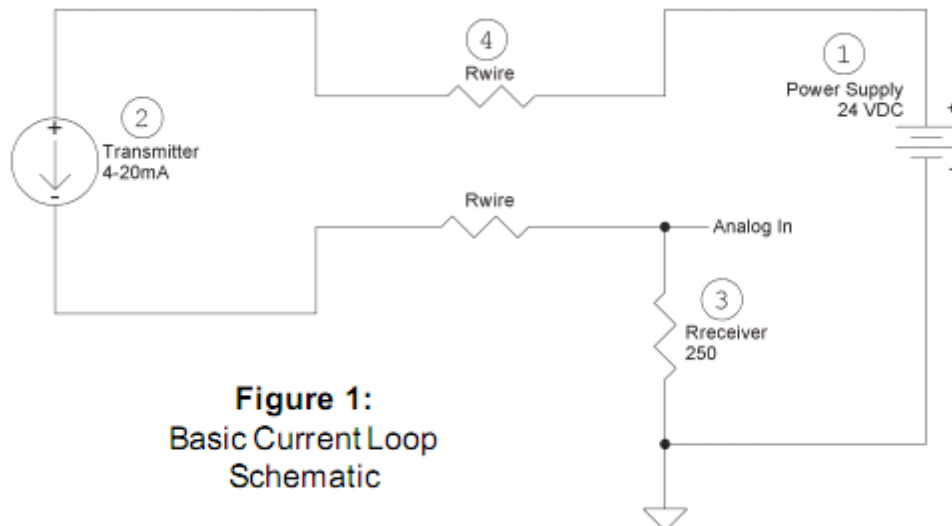


Figure 1:
Basic Current Loop
Schematic

De acordo com a figura anterior, uma resistência interna para a corrente de loop de 250 Ω em um instrumento receptor causará uma queda de tensão de 1V @ 4mA e de 5V @ 20mA, níveis adequados para uma entrada analógica em V_{DC} em CLPs, por exemplo.

A fonte de alimentação para o loop de corrente deve sempre ser DC, para loops com transmissores a 2 fios, usualmente se empregam fontes de 12, 15, 24 ou 36V_{DC}.

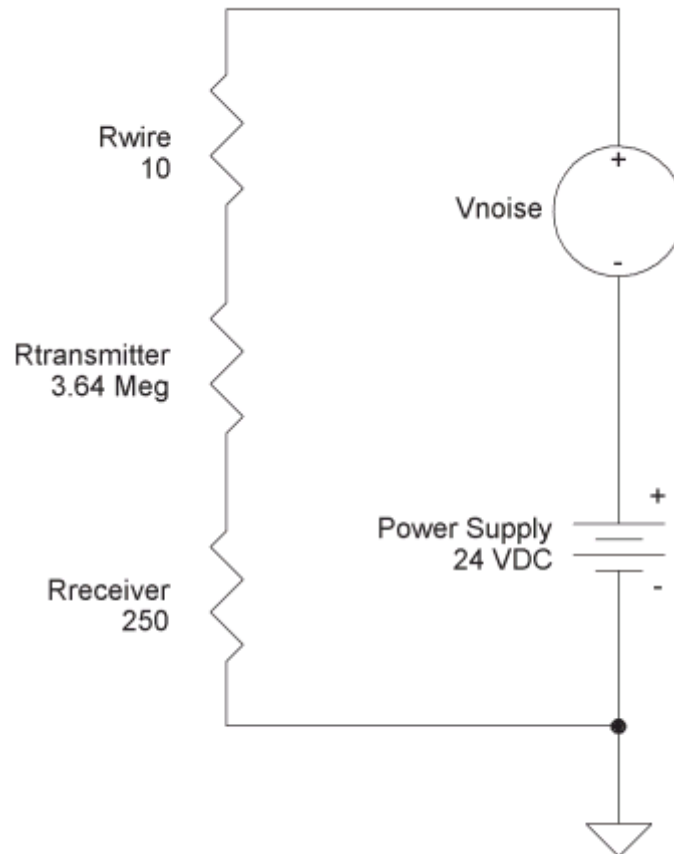
O instrumento transmissor é o elemento principal de um loop de corrente. Ele transforma a informação da variável física medida em um valor de corrente no loop. A corrente de 4mA representa o limite inferior do range do instrumento e a corrente de 20mA, o limite superior. A alimentação dos transmissores é de acordo com uma faixa tolerável de tensão DC, sendo que quando alimentado por uma tensão igual ao limite inferior da faixa e uma corrente de 4mA, o transmissor já deve operar corretamente. Deve-se considerar que o transmissor consome de 7 a 15 V_{DC} do loop para sua própria alimentação.

Devido ao fato de ser muito mais fácil se medir uma tensão elétrica a uma corrente elétrica, os receptores de loop de corrente empregam em geral um resistor de recepção para transformar o sinal de corrente em um sinal de tensão elétrica. Conforme mencionado, o valor mais comum é de 250 Ω , porém, em função da aplicação, encontram-se receptores com resistências de 50 Ω a 1000 Ω .

A passagem de corrente por condutores metálicos produzem uma queda de tensão ao longo dos condutores proporcional à bitola e ao comprimento destes. Para se calcular a queda de tensão em um loop de corrente, deve-se portanto utilizar a lei de Ohm considerando-se a seguinte tabela de resistividade do condutor em unidades norte americanas:

Table 1 Copper Wire Resistance @ 20°C (68°F)		
American Wire Gauge	Ohms per 1,000 feet (305 meters)	
	Solid Wire	Stranded Wire
14	2.53 Ω	2.73 Ω
16	4.02 Ω	4.35 Ω
18	6.39 Ω	6.92 Ω
20	10.15 Ω	10.90 Ω
22	16.14 Ω	16.50 Ω
24	25.67 Ω	27.70 Ω

A grande vantagem da transmissão por loops de corrente é a alta insensibilidade à ruídos externos. Todo transmissor de corrente possui uma resistência associada ao circuito de saída. Idealmente esta resistência deve ser infinita. Em transmissores reais, entretanto, tal resistência é muito alta, porém finita.

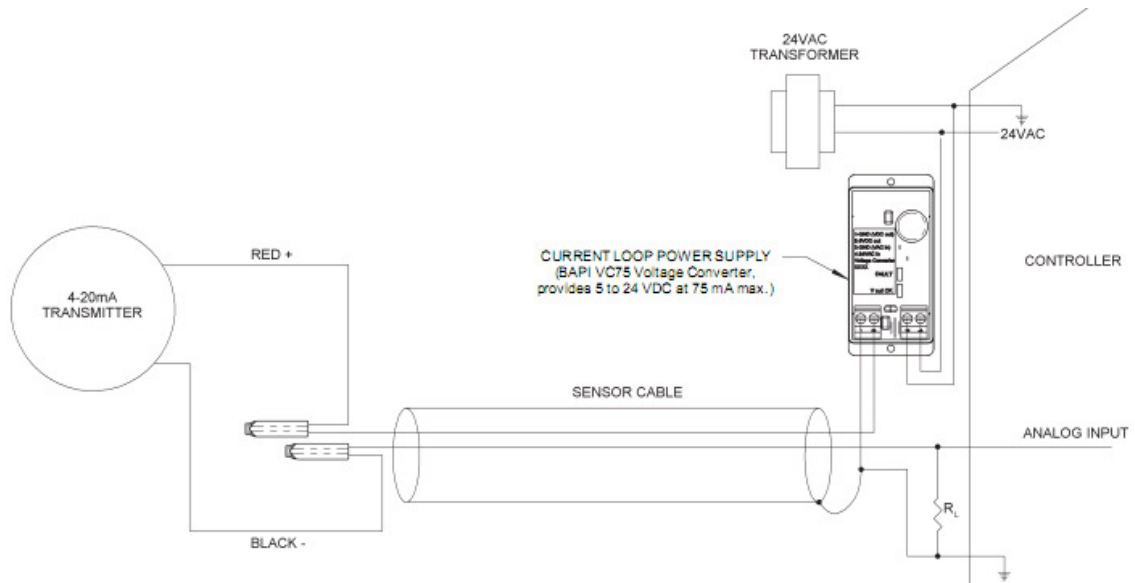


Pelo circuito indicado acima, uma fonte de tensão é adicionada ao loop de corrente para modelar o acoplamento de um ruído externo. Pela alta resistência de saída dos transmissores (ex. $3.64\text{M}\Omega$), quase toda a tensão de ruído é direcionada ao transmissor e apenas uma pequena fração desta é sentida pelo receptor, desta forma, o ruído causa um mínimo efeito no controle do processo.

No exemplo dado, se a tensão de ruído for de 20V, apenas 0,0014 V será sentida pelo receptor. O mesmo princípio se aplica a flutuações na tensão de alimentação.

Projeto de Loops de 4 a 20mA

Primeiramente deve-se calcular a resistência do cabeamento, em função do seu comprimento, bitola e temperatura de trabalho. Deve-se considerar um coeficiente de alteração da resistividade elétrica em função da temperatura de $0,393\% / ^\circ\text{C}$ para o Cobre.

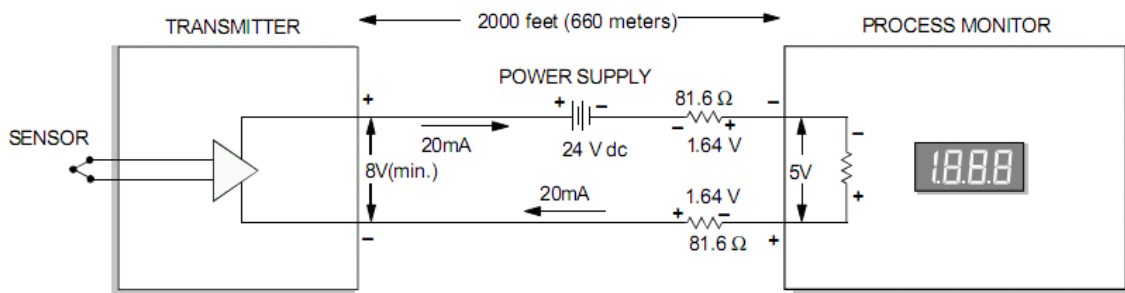


Em seguida, calcula-se a queda de tensão no circuito para as condições extremas de corrente: 4mA e 20mA. Inicia-se pelo cálculo em 20mA. Neste cálculo deve-se considerar a queda de tensão correspondente ao condutor (resistência calculada previamente para máxima temperatura de trabalho, portanto máxima resistência), ao transmissor (dado de catálogo, por exemplo, 15V de queda de tensão mínima) e ao receptor, em função de sua carga (ex. 250 Ohms).

A partir da queda de tensão total para a condição de 20mA, especifica-se a fonte. Por exemplo, se a queda total for de 21V, especifica-se uma fonte de 24V. O excedente de 3V é direcionado ao transmissor, assim, este terá uma alimentação de, neste caso hipotético, 18V e não 15V). Ter uma margem de tensão acima da tensão mínima para a alimentação do transmissor é uma boa prática para se garantir a integridade do loop frente ao envelhecimento dos elementos e a oxidação dos contatos.

A queda de tensão no circuito deve também ser calculada para a condição de 4mA em mínima resistência dos condutores (mínima temperatura possível de trabalho) para se garantir que nesta condição o transmissor não será alimentado com uma tensão superior ao seu máximo limite especificado por catálogo, em geral definido pela sua capacidade de dissipação de potência.

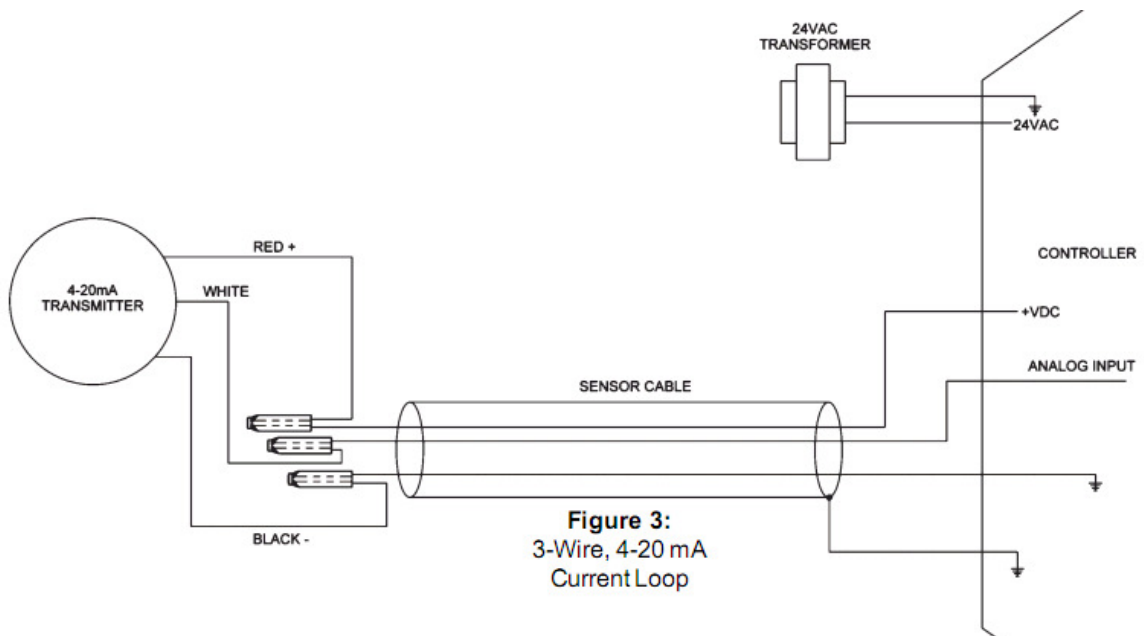
Por exemplo, na condição de 4mA com baixa temperatura do cabeamento e fonte de 24V, a queda hipotética de tensão em um transmissor seria de 22,8V. Caso a faixa de alimentação do instrumento fosse de até 24V, então o circuito estaria bem dimensionado. Entretanto, caso a fonte de alimentação utilizada fosse de 36V_{DC}, então seriam necessários resistores em série no circuito para não sobrecarregar o transmissor.



Exemplo de distribuição de tensão elétrica entre os elementos do loop de corrente

A utilização de circuitos 4-20mA a três fios deve ser considerada quando o transmissor em questão consome mais de que 20mA para seu funcionamento, neste caso, a fonte deve ser

fornecer energia para um circuito de alimentação do transmissor e também para o circuito de sinalização da variável, conforme o exemplo da figura a seguir.



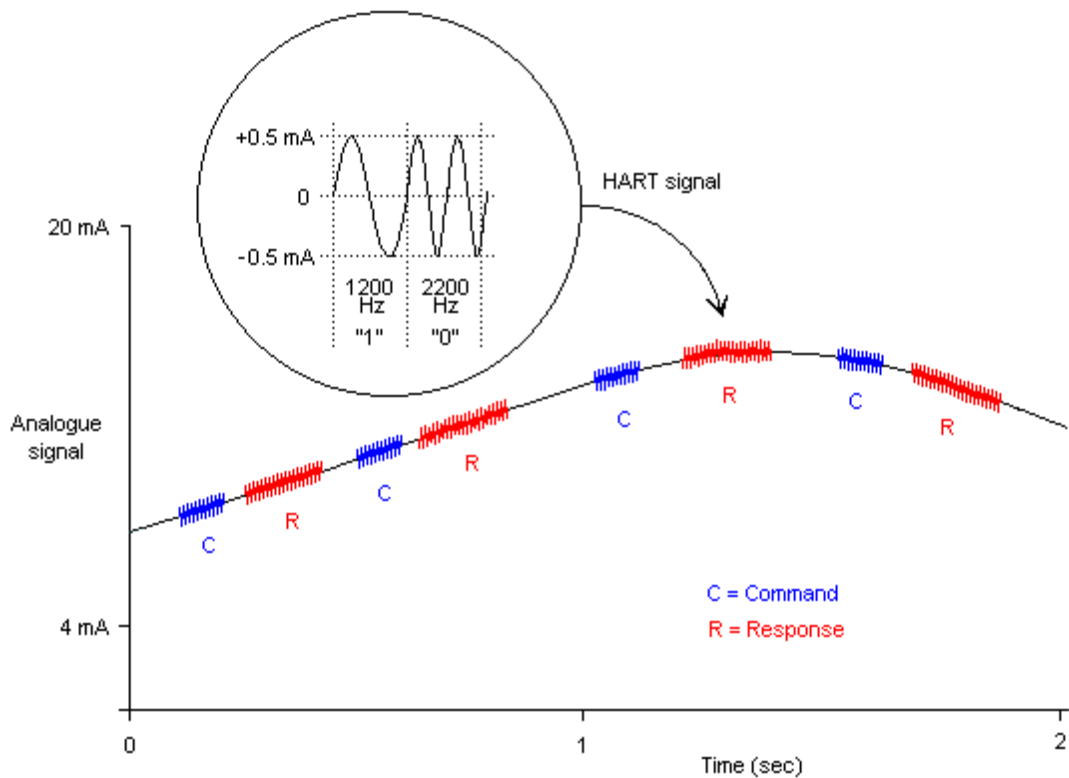
Protocolo HART

O protocolo HART (Highway Addressable Remote Transducer) foi desenvolvido em 1980 pela empresa norte americana Rosemount, que mais tarde fez dele, assim como o Modbus, um protocolo aberto.

O HART é 100% compatível com o sistema 4-20mA, motivo de sua imensa aceitação no mercado de automação industrial, podendo utilizar toda a estrutura de cabeamento e de rede já instalado, bastando apenas a substituição de instrumentos antigos por instrumentos HART “inteligentes”. Originalmente o protocolo foi projetado para realizar operações de calibração e ajustes de variáveis remotamente ao mesmo tempo em que o transmissor envia a variável de processo pelo loop de corrente, foi o primeiro esquema digital, bi-direcional para processos sem eliminar o sinal analógico.

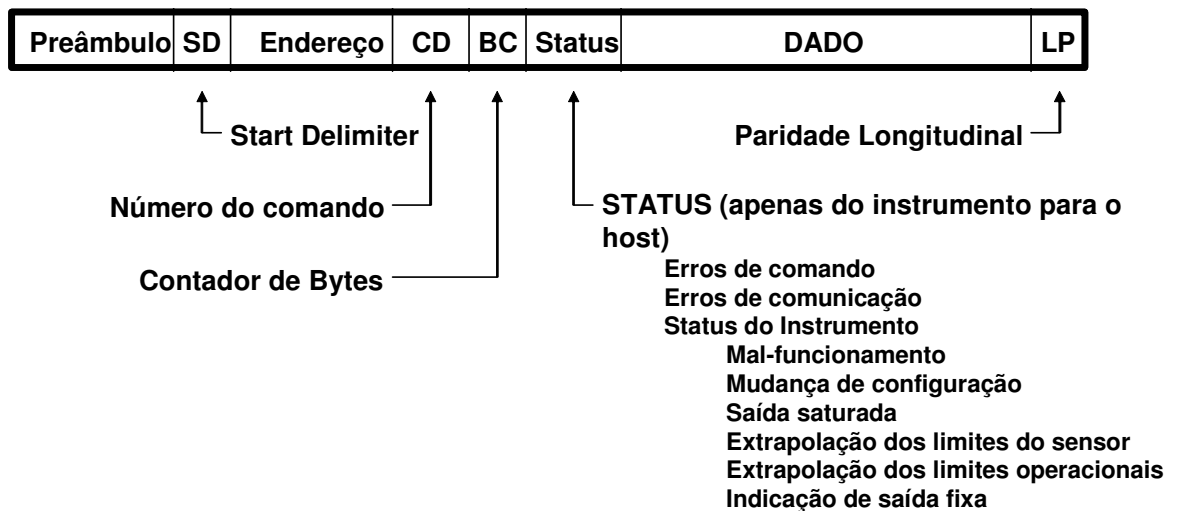
A comunicação estabelecida com o protocolo HART faz uso de sinalização binária com chaveamento de freqüências pelo padrão “Bell 202 Frequency Shift Keying”, com a freqüência de 1200 Hz representando o bit 1 e a freqüência de 2200 Hz representando o bit 0 a uma taxa de 1200bps sobreposta à comunicação analógica. Ambas as freqüências da sinalização digital estão bem acima da freqüência do sinal analógico, tradicionalmente entre 0 e 10Hz, o que garante a não interferência entre ambas.

Nos sistemas atuais, a comunicação HART em geral emprega-se para a transmissão de informações acerca do instrumento, informações sobre calibração, transmissão de variáveis secundárias de processo, diagnósticos, status e alertas.



Representação da comunicação digital HART sobreposta ao loop de corrente

Da mesma forma que o protocolo Modbus, o HART é baseado em comunicação MESTRE-ESCRAVO, ou PERGUNTA-RESPOSTA. A estrutura das mensagens de comunicação é definida por norma da seguinte forma:



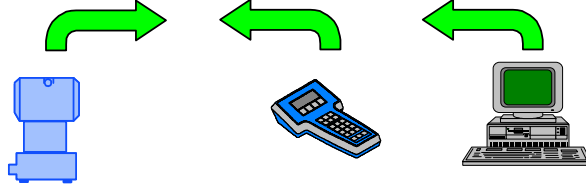
A função de cada mensagem é definida pelo comando a que esta se destina. A família de comandos definida no protocolo HART é configurada em três grupos:

Comandos Universais

Model ID
Tag ID
Description

Message
Date
Range Values

Instrument Limits
Process Measurements
Device Status



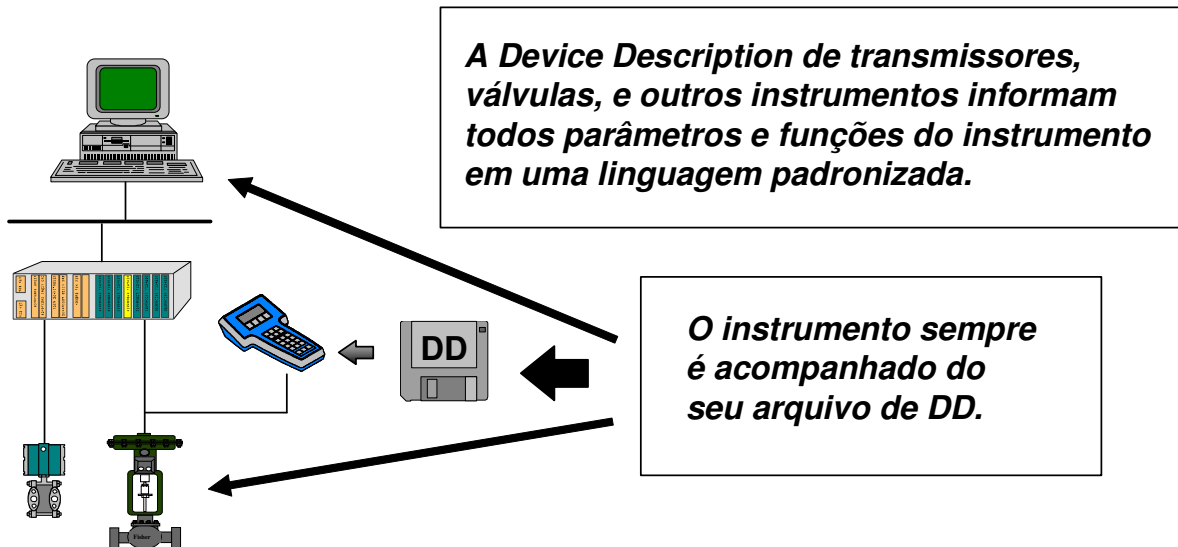
Comandos de prática comum

Read Variables
Calibration (zero, span)
Initiate Self Tests
Serial Number
Time Constant Values

Comandos específicos do fabricante

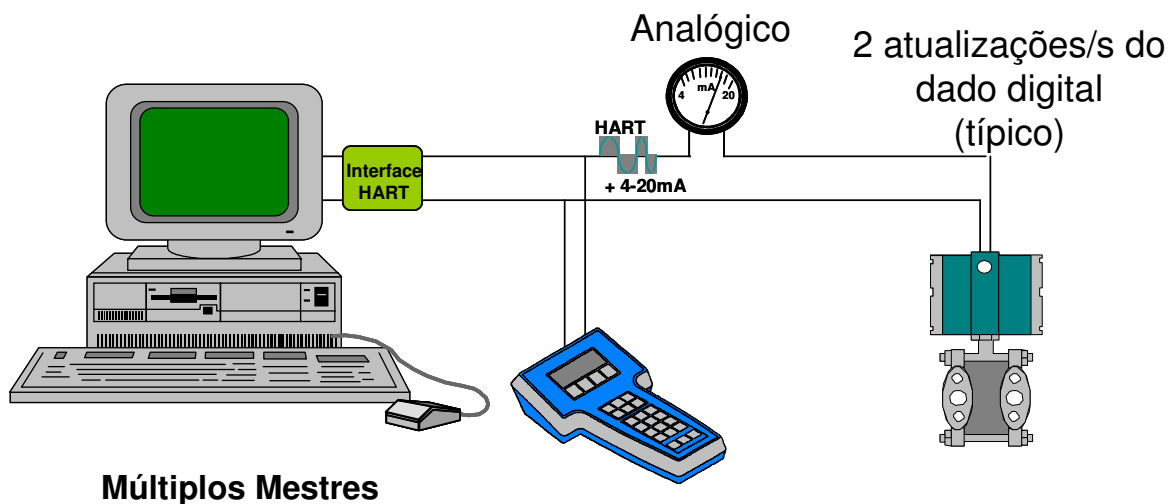
Model Specific Functions
Start, Stop, or Clear Totalizer
Select Primary Variable
Special Calibration Options
PID - SP, Tuning, Etc.

Uma importante inovação do protocolo HART para os sistemas de comunicação industrial foi a introdução de “datasheets” eletrônicos para os dispositivos, na forma de arquivos de “Device Description” ou simplesmente DD. Através da leitura e interpretação destes arquivos que realizam a função de “drivers”, o dispositivo mestre passa a conhecer todas as funcionalidades disponíveis em seu(s) dispositivo(s) escravo(s), tornando mais simples a programação de sistemas supervisórios e de procedimentos de calibração e de manobras em plantas de processo.

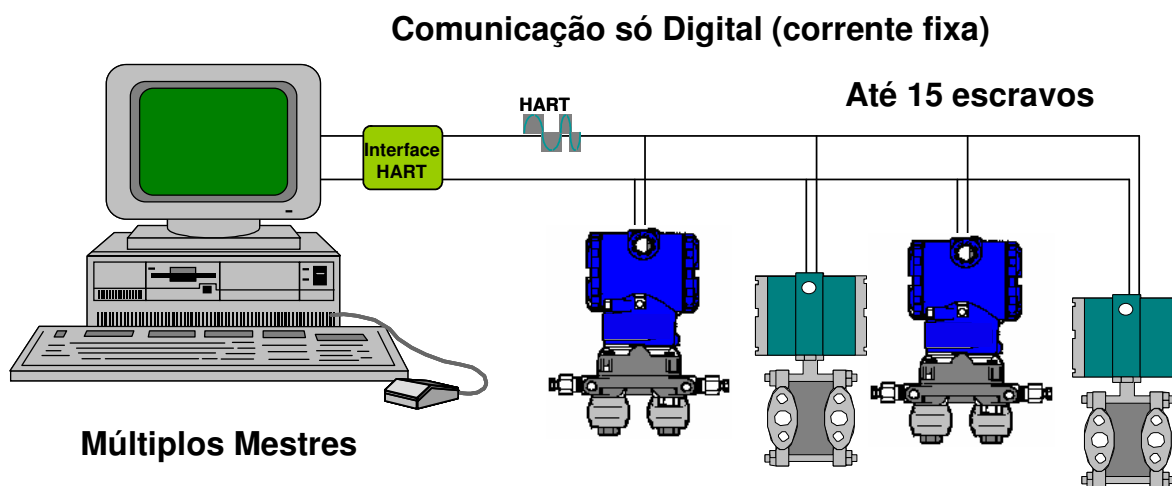


Arquiteturas de Loops de Controle com o protocolo HART

1. Múltiplos Mestres e um dispositivo Escravo



2. Comunicação Multidrop (múltiplos escravos)

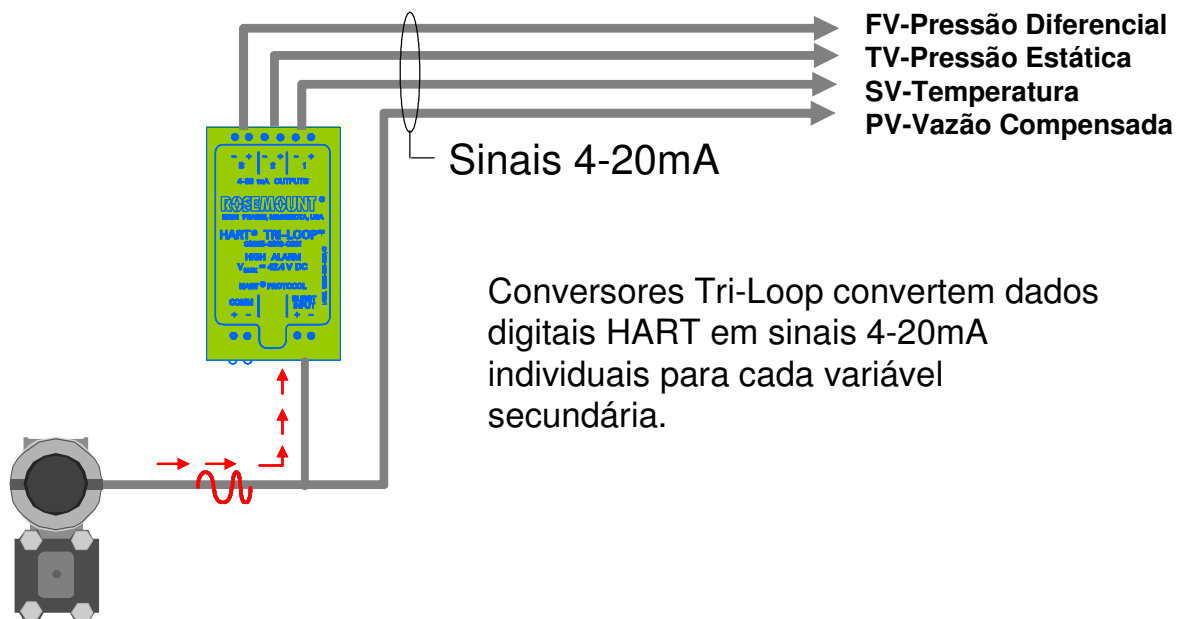


Neste modo de comunicação, o nível de corrente do loop analógico é fixo e somente a comunicação digital é ativada.

Integração de dispositivos HART a sistemas de automação

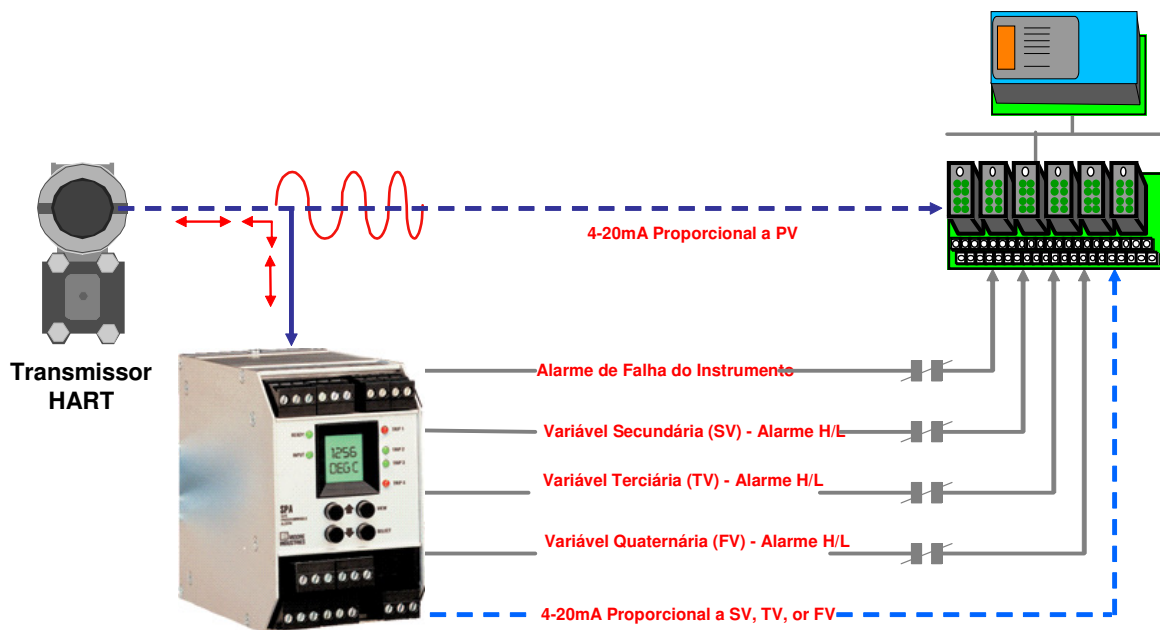
Algumas alternativas para a integração de dispositivos HART a sistemas de automação serão apresentadas a seguir.

1. Conversores de variáveis secundárias HART em loops de 4-20mA



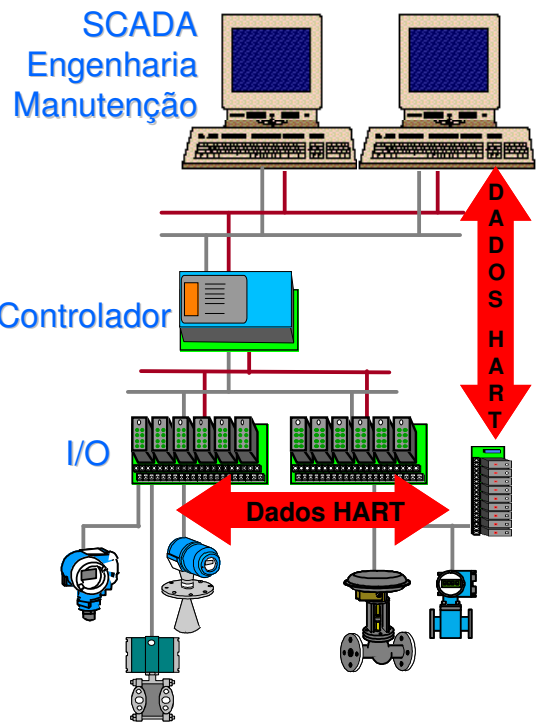
2. Conversores SPA

Convertem dados digitais HART em sinais 4-20mA individuais para cada variável secundária e/ou variáveis de alarme e status em sinais discretos (saídas a relé).

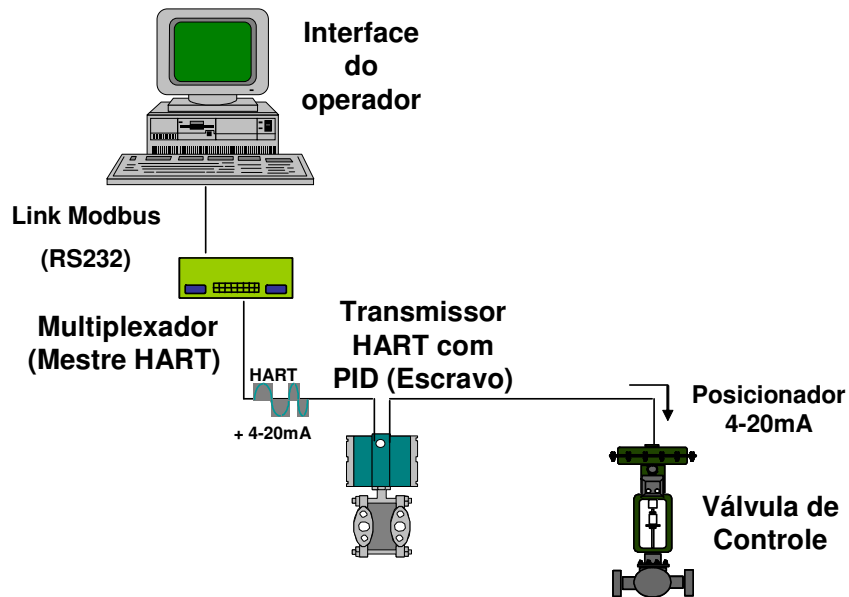


3. Conversores do tipo Gateway HART

Convertem o protocolo HART para outros tipos de protocolos de campo.



4. Integração via Multiplexadores HART



Neste exemplo, a saída do algoritmo de controle PID incluído no transmissor HART modula o sinal analógico entre 4-20mA no loop de corrente para o instrumento posicionador da válvula de controle. As variáveis SP, PV, MV e os parâmetros de sintonia do PID são comunicados digitalmente via HART para o computador Host do operador.