



HI tecnologia

Automação Industrial

Nota de Aplicação

Protocolos de Comunicação nos Equipamentos GII, GII_DUO e G3

HI Tecnologia Indústria e Comércio Ltda.

Documento de acesso Público



Apresentação

Este documento foi elaborado pela **HI Tecnologia Indústria e Comércio Ltda.** Quaisquer dúvidas ou esclarecimentos sobre as informações contidas neste documento podem ser obtidas diretamente com o nosso departamento de suporte a clientes, através do telefone (19) 2139.1700 ou do email suporte@hitecnologia.com.br. Favor mencionar as informações a seguir para que possamos identificar os dados relativos a este documento.

Título documento: Protocolos de Comunicação nos Equipamentos GII, GII_DUO e G3
Referência do documento: ENA.00044
Versão do documento 1.02

HI Tecnologia Indústria e Comércio Ltda.

Sede: Av. Dr. Armando de Sales Oliveira, 445.

Cidade: Campinas – SP

Fone: +55 (19) 2139.1700

CEP: 13076-015

Portal Web: www.hitecnologia.com.br

Contatos

Vendas: vendas@hitecnologia.com.br

Suporte Técnico: suporte@hitecnologia.com.br

Engenharia de Aplicação: engenharia@hitecnologia.com.br

FAQ: faq.webhi.com.br

Portal de documentação On line: doc.hitecnologia.com.br

Forum: forum.hitecnologia.com.br



Índice

1	Abrangência do Documento	4
2	Introdução	5
2.1	Informação Copyright	5
2.2	Isenção de Responsabilidade	5
2.3	Sugestões	5
3	Referências	5
4	Compatibilidade	6
5	Base de dados dos equipamentos GII, GII_DUO e G3	7
5.1	Formato dos tipos de dados	8
5.1.1	R – Contato	9
5.1.2	M Memórias Inteiras	9
5.1.3	L Memórias Inteiras Longas	10
5.1.4	D - Memórias Reais	10
5.1.5	X - Texto (Strings)	11
6	Protocolos de comunicação disponíveis	12
6.1	Operação dos equipamentos GII, GII_DUO e G3 no modo Escravo	13
6.1.1	Protocolo SCP-HI no modo escravo	13
6.1.2	Protocolo MODBUS no modo escravo	20
6.2	Operação dos equipamentos GII, GII_DUO e G3 no modo Mestre	27
6.2.1	Protocolo SCP-HI no modo Mestre	28
6.2.2	Protocolo MODBUS no modo Mestre	33
6.3	Protocolos ASCII	40
6.3.1	Dispositivos que enviam informações em forma de Texto	40
6.3.2	Dispositivos que recebem informações em forma de Texto	41
6.3.3	Operação de frames não ASCII	41
7	Arquiteturas de comunicação	42
7.1	Comunicação Ponto a Ponto	42
7.2	Comunicação em rede RS485	42
7.3	Comunicação em rede Ethernet	43
	Controle do Documento	44
	Considerações gerais	44



1 Abrangência do Documento

Este documento abrange os seguintes Controladores nas plataformas especificadas abaixo:

Equipamentos			Plataforma					Abrangência
Tipo	Família	Modelo	GI	GII	GII Duo	G3	G3S	✓
Controladores	MCI02	MCI02	X					
		MCI02-QC	X					
	ZAP500	ZAP500/BX/BXH	X					
		ZTK500/501	X					
	ZAP900	eZAP900/901, ZAP900/901		X				✓
		eZTK/ZTK900, ZAP900-BXH		X				✓
	ZAP91X	ZAP910 / ZTK910					X	✓
		ZAP911					X	✓
		eZAP910 / eZTK910					X	✓
		eZAP911					X	✓
		ZAP910-BXH					X	✓
		ZAP910-S / ZTK910-S						X
		ZAP911-S						X
		eZAP910-S / eZTK910-S						X
		eZAP9911-S						X
		ZAP910-BXH-S						X
	FLEX950	FLEX950-PLC		X				✓
	P7C	CPU300				X		✓
		CPU301, PPU305					X	✓
		CPU302, PPU306						X
NEON	CPU400					X		
IHMs	MMI600	MMI600/601		X				
	MM650	MMI650		X				
	MMI800	MMI800		X				
	FLEX950	FLEX950-IHM		X				
	GTI100	GTI100-RS/GTI00-ET						



2 Introdução

Este documento tem como objetivo descrever os recursos de comunicação disponíveis nos equipamentos da HI tecnologia classificados como GII, GII_DUO e G3, descrever os protocolos de comunicação utilizados e, como a base de dados dos equipamentos GII, GII_DUO e G3 pode ser acessada (lida ou escrita), utilizando estes protocolos.

- O documento é dividido nas seguintes seções:
- Referências de documentações sobre os protocolos mencionados
- Apresentação dos protocolos disponíveis nos equipamentos GII, GII_DUO e G3.
- Base de dados dos equipamentos GII, GII_DUO e G3
- Protocolos de comunicação disponíveis e sua inter-relação com as bases de dados dos equipamentos
- Exemplos de Arquiteturas de Comunicação

2.1 Informação Copyright

Este documento é de propriedade da HI Tecnologia Indústria e Comércio Ltda. © 2006, sendo distribuído de acordo com os termos apresentados a seguir.

- Este documento pode ser distribuído no seu todo, ou em partes, em qualquer meio físico ou eletrônico, desde que os direitos de copyright sejam mantidos em todas as cópias.

2.2 Isenção de Responsabilidade

A utilização dos conceitos, exemplos e outros elementos deste documento é responsabilidade exclusiva do usuário. A HI Tecnologia Indústria e Comércio Ltda. não poderá ser responsabilizada por qualquer dano ou prejuízo decorrente da utilização das informações contidas neste documento.

2.3 Sugestões

Sugestões são bem vindas. Por favor, envie seus comentários para suporte@hitecnologia.com.br
Novas versões deste documento podem ser liberadas sem aviso prévio. Caso tenha interesse neste conteúdo acesse o site da HI Tecnologia regularmente para verificar se existem atualizações liberadas deste documento.

3 Referências

Todos os documentos e aplicativos referenciados abaixo estão disponíveis para *download* no site da HI Tecnologia: www.hitecnologia.com.br



Documentos	Referências
Notas de Aplicação	
ENA.00008	Controladores HI com Protocolo MODBUS (*1).
ENA.00019	Comunicação Remota com os Controladores da HI (Função RCB) (*1).
ENA.00022	Configuração dos canais de comunicação dos Controladores HI (*1).
ENA.00026	TCP/IP com PLC's HI via conversor ESC710 ou ESC713
ENA.00030	Integrando controladores em rede Ethernet com protocolo UDP
ENA.00043	Interface dos Controladores HI-GII com Dispositivos via Protocolo ASCII
ENA.00045	Comunicação remota com controladores HI-GII.
Controlador P7C	
PMU 10730000	PMU 10730000 Módulo CPU300 (formato PDF)
PMU 10730100	PMU 10730100 Módulo CPU301 (formato PDF)
PMU 10730100	PMU 10730100 Módulo PPU305 (formato PDF).
Controlador ZAP900 / 901 / 900 BXH	
PET 108.001.00	Controlador Lógico Programável ZAP900/901 (formato PDF).
Controlador eZAP900/ eZAP901	
PET 108.002.00	
PET.110.001.00	
<i>Softwares Aplicativos (download gratuito)</i>	
SPDSW	Software de programação dos controladores HI, em ambiente <i>Windows</i> .

4 Compatibilidade

As funcionalidades descritas neste documento estão disponíveis a partir das seguintes versões

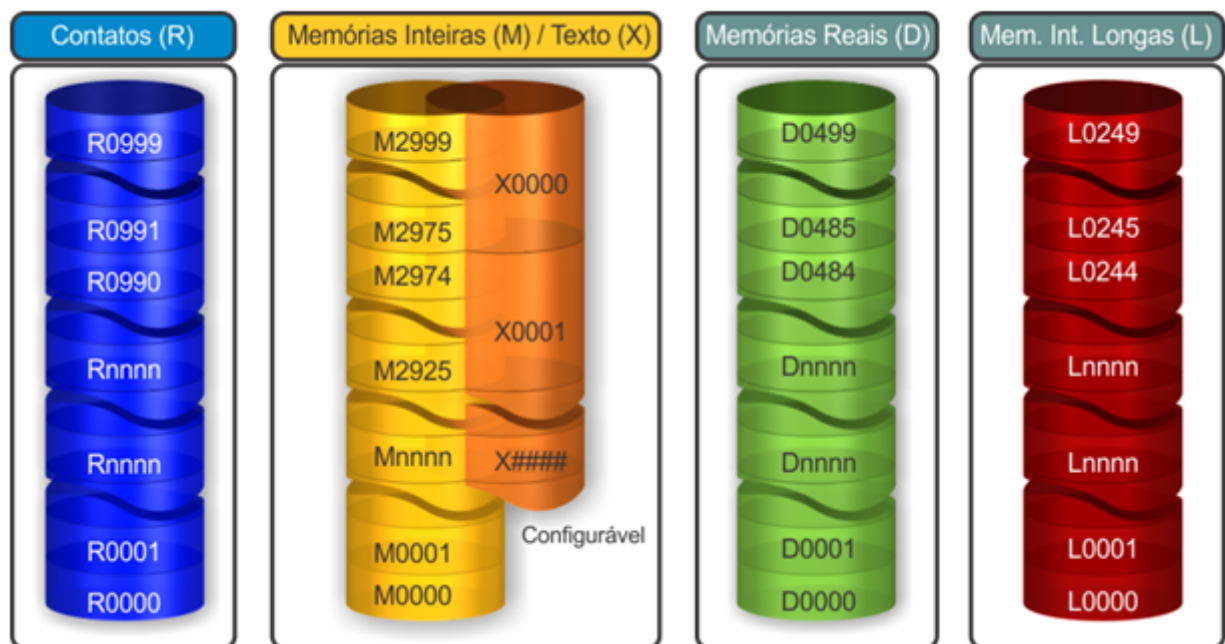
- Aplicativo SPDSW: Versão 2.5.00 ou superior.
- Controlador ZAP900 ou ZAP901: *Firmware* versão 1.3.03 ou superior.
- Controladores ZAP91X (ZAP910 ou ZAP911 ou eZAP910 ou eZAP911): *Firmware* versão 1.2.01 ou superior.
- MMI800 ou MMI650: *Firmware* versão 2.2.06 ou superior.



- P7C-CPU300: *Firmware* versão 2.0.11 ou superior.
- P7C-CPU301: *Firmware* versão 1.3.02 ou superior.
- P7C-PPU305: *Firmware* versão 1.0.02 ou superior.

5 Base de dados dos equipamentos GII, GII_DUO e G3

O objetivo principal da comunicação nos equipamentos é permitir a troca de informações entre os mesmos. Para tanto, é necessário que estas informações estejam em um formato que possa ser interpretado corretamente por ambos. Estas informações estão agrupadas nos equipamentos GII, GII_DUO e G3 em uma base de dados comum conforme apresentado na figura a seguir:



Base de Dados dos Equipamentos GII, GII_DUO e G3

Como apresentado na figura acima, a base de dados dos equipamentos GII, GII_DUO e G3 é composta por alguns tipos de dados:



Tipo	Identificador	Tamanho	Descrição
Boleano	R	1 byte	Flag lógico (0: Falso) (-1: Verdadeiro);
Inteiro	M	2 bytes	Valor inteiro de 16 bits com sinal (-32768 .. +32767);
Inteiro Longo	L	4 bytes	Valor inteiro de 32 bits com sinal (-2.147.483.648 a .. +2.147.483.647);
String	X	50 bytes	Texto com tamanho máximo de 48 bytes. Cada string ocupa o espaço de 25 memórias M consecutivas;
Real	D	4 bytes	Valor real de 32 bits padrão IEEE (1.18 x 10E-38 .. 3.40 x 10E+38),

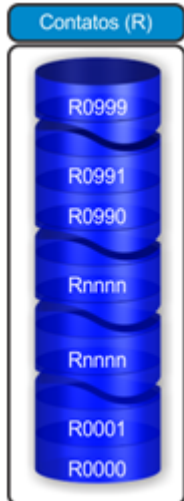
1. O número de variáveis de cada tipo é dependente do equipamento e do *firmware* utilizado.
2. Os tipos de dados R, M e D sempre existem nos equipamentos, e podem ser acessados mesmo que não exista um programa de aplicação carregado.
3. O tipo de dado L somente existe a partir dos equipamentos G3, e pode ser acessado mesmo que não exista um programa de aplicação carregado.
4. O tipo de variável string é opcional e, portanto pode não existir. O número de variáveis alocadas deste tipo é definido dinamicamente pelo programa de aplicação carregado.
5. Os tipos de dados, Inteiro e String, compartilham o mesmo espaço de memória (vide item 4.1.4) O motivo desta estratégia se deve ao fato de que o protocolo MODBUS não define um tipo de dado string e, portanto, não existem funções de comunicação para leitura e escrita destas variáveis. Estando as *strings* mapeadas na mesma base de memórias inteiras, estas podem ser acessadas através das funções de leitura e
6. Todas as variáveis são referenciadas através do seu índice que, para todos os tipos começa em zero. Por exemplo, a quarta memória inteira da base de dados é endereçada como M0003, a segunda string da base de dados é referenciada como X0001, a quarta variável lógica é acessada como R0003.

5.1 Formato dos tipos de dados

Cada tipo de dado definido na base dos equipamentos GII, GII_DUO e G3 necessita de uma quantidade de memória distinta e pode representar valores ou informações específicas. Os itens seguintes apresentam o formato e o tipo de informação armazenada em cada tipo de dado.



5.1.1 R – Contato

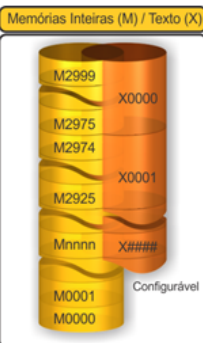


Este tipo de dado é utilizado para armazenar informações lógicas (Binárias). É normalmente utilizado para representar estados tais como Ligado/Desligado, Falso/Verdadeiro, Não/Sim e outros. A tabela seguinte resume as características deste tipo de dado.

Contato	
Identificador do tipo de dado	R
Tamanho do tipo de dado	1 <i>byte</i>
Faixa de Valores	0 (00h) ou -1 (FFh)
Faixa de Valores	1000 (PLC)

(*1) – A quantidade pode variar em função do equipamento e do tipo de *firmware* carregado. Consulte a especificação do produto utilizado para se certificar deste valor.

5.1.2 M Memórias Inteiras



Este tipo de dado é utilizado para armazenar valores inteiros com sinal que podem ser representados em 16 *bits*. A tabela seguinte resume as características deste tipo de dado.

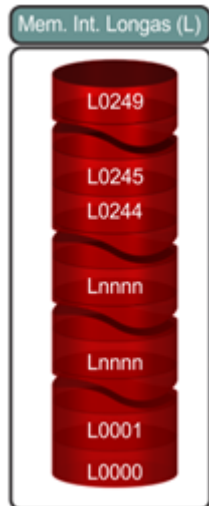
Memórias Inteiras	
Identificador do tipo de dado	M
Tamanho do tipo de dado	2 bytes (16 <i>bits</i>)
Notação	Primeiro <i>byte</i> é o LSB
Faixa de Valores	-32768 .. + 32767

Quantidade (*1) 3000 (PLC),

(*1)– A quantidade pode variar em função do equipamento e do tipo de *firmware* carregado. Consulte a especificação do produto utilizado para se certificar deste valor.



5.1.3 L Memórias Inteiras Longas



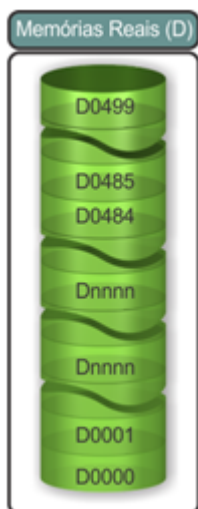
Este tipo de dado é utilizado para armazenar valores inteiros longos com sinal que podem ser representados em 32 *bits*. Tipo de dado disponível a partir dos equipamentos G3. A tabela seguinte resume as características deste tipo de dado.

Memórias Inteiras Longas	
Identificador do tipo de dado	L
Tamanho do tipo de dado	4 bytes (32 <i>bits</i>)
Notação	Primeiro <i>byte</i> é o LSB
Faixa de Valores	-2.147.483.648 .. +2.147.483.647

Quantidade (*1) 250 (PLC).

(*1) – A quantidade pode variar em função do equipamento e do tipo de *firmware* carregado. Consulte a especificação do produto utilizado para se certificar deste valor.

5.1.4 D - Memórias Reais



Este tipo de dado é utilizado para armazenar valores reais em ponto flutuante que podem ser representados em 32 *bits*. A tabela seguinte resume as características deste tipo de dado.

Memórias Reais	
Identificador do tipo de dado	D
Tamanho do tipo de dado	Tamanho do tipo de dado
Notação	IEEE
Faixa de Valores	$1.175494351 \times 10^{-38} \dots 3.4028235 \times 10^{38}$
Quantidade (*1)	500 (PLC).

(*1) – A quantidade pode variar em função do equipamento e do tipo de *firmware* carregado. Consulte a especificação do produto utilizado para se certificar deste valor.



5.1.5 X - Texto (*Strings*)

	<p>Este tipo de dado é utilizado para armazenar uma sequência de <i>bytes</i> que podem ou não representar um texto. Este tipo de dado compartilha o mesmo espaço de memória do tipo M (memória Inteira) e sua alocação se dá conforme indicado a seguir:</p> <p>Cada variável do tipo Texto possui 50 <i>bytes</i> e, portanto, quando alocada "consome" 25 memórias inteiras (M). Os textos são alocados a partir do topo da base do tipo M. Sendo assim, considerando um equipamento com 3000 memórias inteiras, o texto X0000 irá ocupar o espaço alocado para as memórias 2975 a 2999 (25 memórias). A próxima variável X0001 será alocada nas 25 memórias anteriores, ou seja, de 2950 a 2974 e assim sucessivamente. Para se obter a memória M inicial M_i, da variável Texto X000N utilize a seguinte equação:</p> $M_i = \text{MAX_M} - ((n + 1) * 25)$ <p>onde:</p> <p>MAX_M - número máximo de variáveis do tipo M disponíveis no equipamento associado.</p> <p>N número da variável do tipo Texto a ser identificada</p>
--	--

Finalmente, cabe ressaltar que, este tipo de dado é opcional e pode não estar disponível em todos os equipamentos.

Quando disponível, o número de textos presentes no equipamento é variável e definido pelo programa de aplicação corrente. A tabela seguinte resume as características deste tipo de dado.

Texto (String)	
Identificador do tipo de dado	X
Tamanho do tipo de dado	48 bytes
Notação	Vide item 4.1.4.1
Faixa de Valores	Não aplicável
Quantidade (*1)	0 .. 120 (PLC)

– A quantidade pode variar em função do equipamento e do tipo de *firmware* carregado. Consulte a especificação do produto utilizado para se certificar deste valor.

5.1.5.1 Formato interno do tipo Texto (*String*)

A variável do tipo Texto dentro dos equipamentos GII, GII_DUO e G3 possui o seguinte formato de dados:



BC (1 byte)	TEXTO (0..48 bytes)	DEL (1 byte)
Espaço alocado para a variável = 50 bytes		

BC Contador de bytes válidos no texto associado (primeiro *byte* da sequência de valores da variável).

Texto Vetor de caracteres que compõem o texto associado. Note que esta sequência de *bytes* pode representar um texto ASCII bem como uma sequência de *bytes* qualquer (ex. uma sequência de códigos de controle a serem enviados para a impressora para configuração da mesma).

DEL Delimitador de fim do texto (*Byte* com valor zero). Este valor é inserido imediatamente após o último *byte* válido do texto, independente do tamanho do mesmo.

Baseado no padrão anteriormente apresentado, o texto "HI TECNOLOGIA" é armazenado em uma variável do tipo texto conforme ilustrado a seguir:

Índice do Caractere	Valor Decimal	Valor ASCII
0	13	
1	72	H
2	73	I
3	32	
4	84	T
5	69	E
6	67	C
7	78	N
8	79	O
9	76	L
10	79	O
11	71	G
12	73	I
13	65	A
14	0	
..	?	
49	?	

6 Protocolos de comunicação disponíveis

Os equipamentos da HI Tecnologia classificados com GII, GII_DUO e G3 suportam comunicação através dos seguintes protocolos de comunicação:



- SCP-HI
- MODBUS-RTU
- MODBUS -TCP
- Protocolos ASCII

A tabela a seguir apresenta as principais características e compara os protocolos implementados:

Características	SCP-HI	MODBUS-RTU	MODBUS-TCP	ASCII
Modelo de comunicação	Mestre/Escravo	Mestre/Escravo	Mestre/Escravo	Nenhum
Operação via RS232-C	Sim	Sim	Não	Sim
Operação em rede RS485	Sim	Sim	Não	Sim (*1)
Operação em rede Ethernet	Sim	Não	Sim	Sim (*1)
Número de Estações	252 (1..252)	247 (1..247)	Não definido (*2)	(*1)
Porta de comunicação (TCP)	na	na	502	na
Suporta redirecionamento (*3)	Sim	Sim	Sim	Não

na – não se aplica.

(*1) – Depende da especificação do protocolo do equipamento conectado.

(*2) – Cada equipamento possui, ou está associado a um endereço IP próprio.

(*3) – Redirecionamento de pacotes é um recurso exclusivo dos equipamentos da HI Tecnologia, não fazendo parte da especificação de nenhum dos protocolos utilizados.

Como já mencionado anteriormente, não é escopo deste documento discutir a implementação de cada um dos protocolos. Na sessão de referências podem ser obtidas informações adicionais sobre os protocolos MODBUS e SCP-HI.

Apesar de possível a comunicação com qualquer dos protocolos acima, os mesmos não estão disponíveis em todos os tipos de equipamentos. Por exemplo, até a presente data, o protocolo ASCII não está habilitado para as MMI's GII (MMI650 e MMI800). Verifique na especificação do equipamento quais os protocolos habilitados para operação.

6.1 Operação dos equipamentos GII, GII_DUO e G3 no modo Escravo

Neste modo de operação, o equipamento remoto opera como mestre da rede de comunicação (Cliente), enquanto o equipamento GII, GII_DUO e G3 opera como Servidor de Comunicação (Escravo), conforme exemplificado na figura seguinte, especificamente para o equipamento GII.

6.1.1 Protocolo SCP-HI no modo escravo



Utilizando o protocolo SCP-HI o equipamento escravo disponibiliza acesso para leitura e escrita de variáveis dos tipos R, M e D. O tipo de dado X (*string*) pode ser obtido lendo-se o conjunto de memórias M correspondentes.

A tabela a seguir apresenta os tipos de transações possíveis de serem realizadas por um equipamento GII, GII_DUO e G3 operando em modo escravo com protocolo SCP-HI:

	Leitura				Escrita				Função	Parâmetros
	R	M	D	X	R	M	D	X		
1	√								ReadVarType R	Estação, End. inicial, Quantidade;
2		√							ReadVarType M	Estação, End. inicial, Quantidade;
3			√						ReadVarType D	Estação, End. inicial, Quantidade;
4(a)				√					ReadVarType M	Estação, End. inicial, Quantidade;
5					√				WriteVarType R	Estação, End. inicial, Quantidade, Valores;
6						√			WriteVarType M	Estação, End. inicial, Quantidade, Valores;
7							√		WriteVarType D	Estação, End. inicial, Quantidade, Valores;
8(b)								√	WriteVarType M	Estação, End. inicial, Quantidade, Valores.

Obs.:

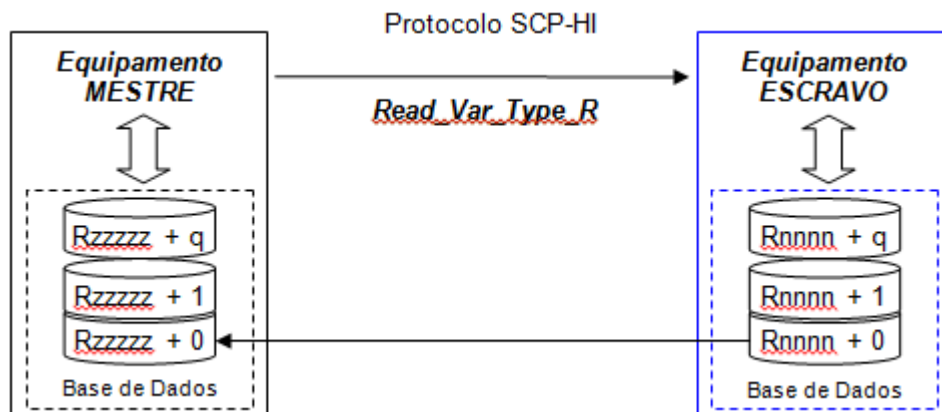
- a – Para leitura de variáveis do tipo *string* deve-se utilizar o comando SCP-HI "ReadVarType M" lendo a partir da memória M inicial da string, as 25 memórias consecutivas.
- b – Para escrita de variáveis do tipo *string* deve-se utilizar o comando SCP-HI "WriteVarType M" escrevendo a partir da memória M inicial da string, as 25 memórias consecutivas.



6.1.1.1 Tipo R (Contatos)

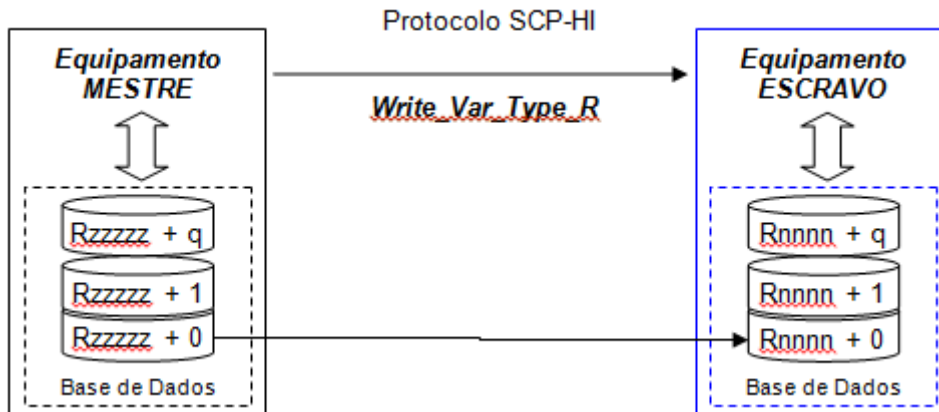
6.1.1.1.1 Leitura

Função SCP-HI	READ_VAR_TYPE_R	
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..249; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.1.1.2 Escrita

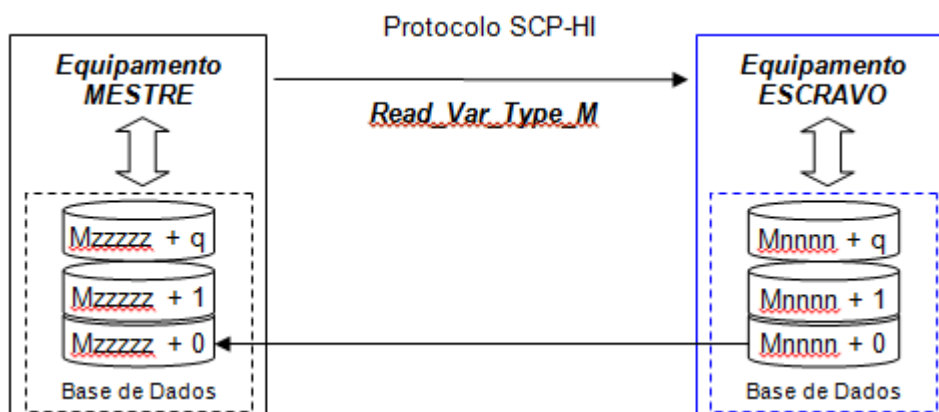
Função SCP-HI	WRITE_VAR_TYPE_R	
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial a ser alterada
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..246; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.1.2 Tipo M (Memórias Inteiras)

6.1.1.2.1 Leitura

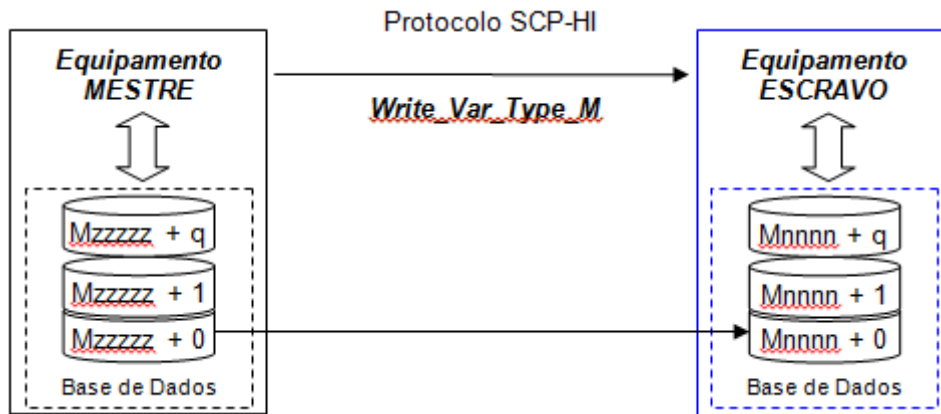
Função SCP-HI	READ_VAR_TYPE_M	
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..124; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.





6.1.1.2.2 Escrita

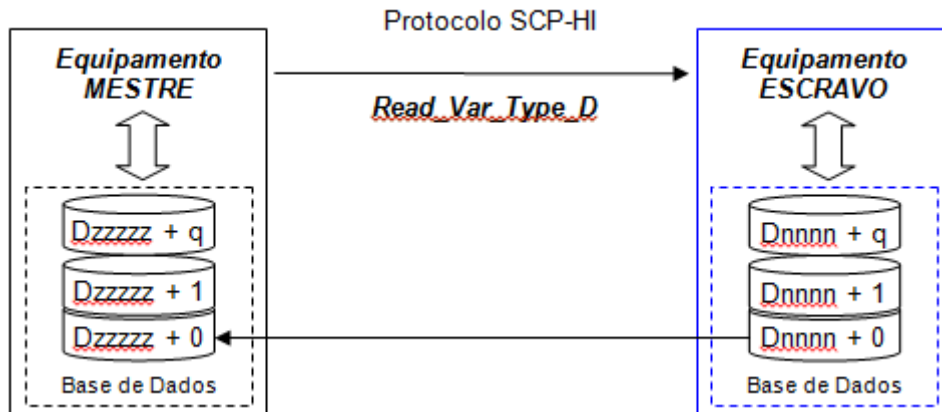
Função SCP-HI	WRITE_VAR_TYPE_M	
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..123; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.1.3 Tipo D (Memórias Reais)

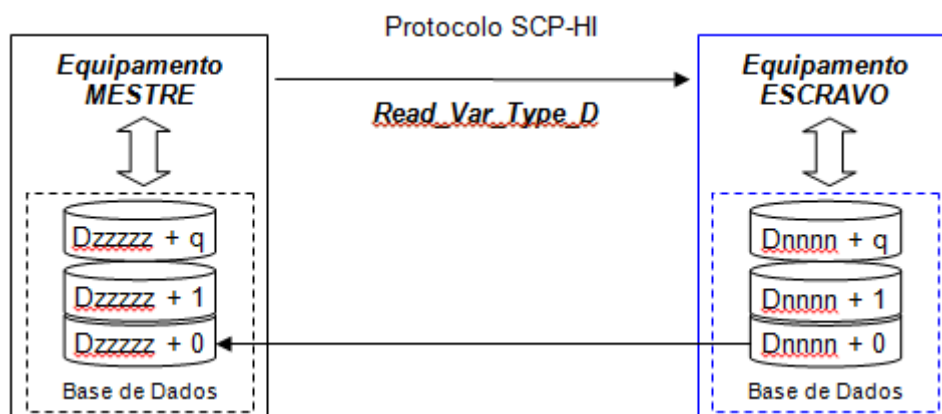
6.1.1.3.1 Leitura

Função SCP-HI	READ_VAR_TYPE_D	
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..62.; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.1.3.2 Escrita

Função SCP-HI		WRITE_VAR_TYPE_D
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..61; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

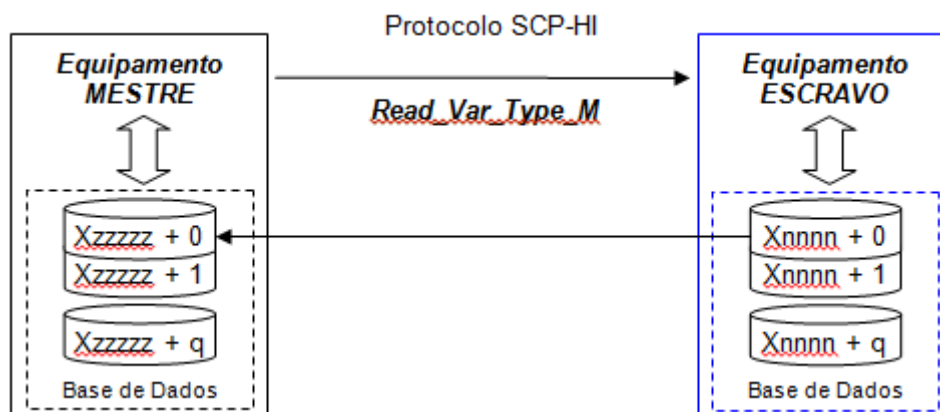




6.1.1.4 Tipo X (Texto)

6.1.1.4.1 Leitura

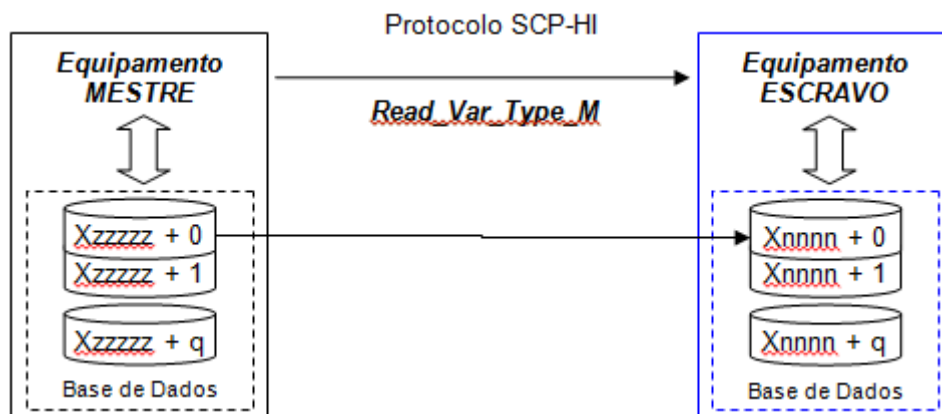
Função SCP-HI		<i>READ_VAR_TYPE_M</i>
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial associada ao texto a ser obtido. Para se obter a memória associada ao início do texto utilize a seguinte fórmula: $M_i = MAX_M - ((x + 1) * 25)$ onde: MAX_M número máximo de variáveis do tipo M disponíveis no equipamento associado. x número da variável do tipo Texto a ser identificada.
	VQuant	Quantidade de Textos a partir da variável inicial * 25 O parâmetro VQuant pode variar entre 1..100 ou equivale a solicitar 1 .. 4 textos. O limite máximo pode ser menor que o especificado anteriormente dependendo no número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.1.4.2 Escrita



Função SCP-HI		READ_VAR_TYPE_X
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial associada ao texto a ser escrito. Para se obter a memória associada ao início do texto utilize a seguinte fórmula: $M_i = MAX_M - ((x + 1) * 25)$ onde: MAX_M número máximo de variáveis do tipo M disponíveis no equipamento associado. x número da variável do tipo Texto a ser identificada.
	VQuant	Quantidade de Textos a partir da variável inicial * 25 O parâmetro VQuant pode variar entre 1..100 ou equivale a escrever de 1 .. 4 textos. O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.2 Protocolo MODBUS no modo escravo

Utilizando o protocolo MODBUS-RTU ou MODBUS-TCP o equipamento escravo disponibiliza acesso para leitura e escrita de variáveis tipo R, M e D. O tipo de dados X (string) pode ser obtido lendo-se o conjunto de memórias M correspondentes.

A tabela a seguir apresenta os tipos de transações possíveis de serem realizadas por um equipamento GII, GII_DUO e G3 operando em modo escravo com protocolo MODBUS:



	Leitura				Escrita				Função	Parâmetros
	R	M	D	X	R	M	D	X		
1	√								Read Coils	Estação, end. Inicial, quantidade
2		√							Read Holding Registers	Estação, end. Inicial, quantidade
3 (a)			√						Read Holding Registers	Estação, end. Inicial, quantidade
4 (b)				√					Read Holding Registers	Estação, end. Inicial, quantidade
5					√				Write Single Coil Write Multiple Coils	Estação, endereço, valor Estação, end. Inicial, quantidade, valores
6						√			Write Single Register Write Holding Registers	Estação, endereço, valor Estação, end. Inicial, quantidade, valores
7 (c)							√		Write Holding Registers	Estação, end. Inicial, quantidade, valores
8 (d)								√	Write Holding Registers	Estação, end. Inicial, quantidade, valores

Obs.:

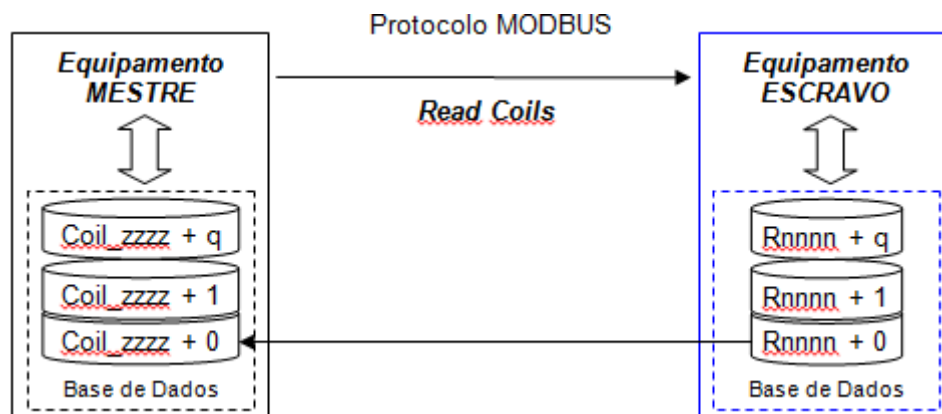
- Para leitura da base dados de memórias reais (D), deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS. Como o tipo de dado D é de 32 bits (4 *bytes*), deve-se solicitar 2 registros para cada valor a ser obtido. Por exemplo, para leitura da variável D0002 deve-se montar um comando Modbus "*Read Holding Registers*" especificando 10002 para o endereço e 2 para a quantidade.
- Para leitura de variáveis do tipo string deve-se utilizar o comando MODBUS "*Read Holding Registers*" lendo a partir da memória M inicial da *string* (vide item 4.1.4), as 25 memórias consecutivas.
- Para escrita na base de dados de memórias reais (D), deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS. Como o tipo de dado D é de 32 bits (4 *bytes*), deve-se enviar 2 registros para cada valor a ser escrito. Por exemplo, para escrever da variável D0005 deve-se montar um comando MODBUS "*Write Multiple Registers*" especificando 10005 para o endereço e 2 para a quantidade. Note que os 2 registros enviados devem ser um valor IEEE em ponto flutuante válido.
- Para escrita de variáveis do tipo string deve-se utilizar o comando MODBUS "*Write Holding Registers*" escrevendo a partir da memória M inicial da *string* (vide item 4.1.4), as 25 memórias consecutivas.

6.1.2.1 Tipo R (Contatos)

6.1.2.1.1 Leitura

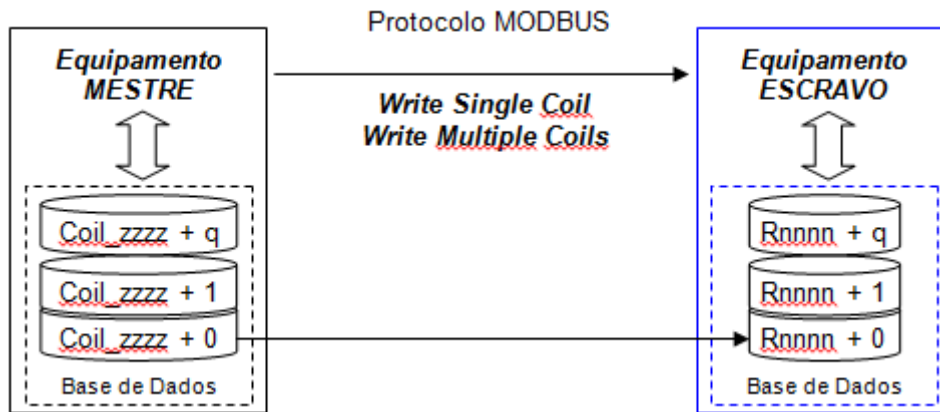


Função MODBUS		READ COILS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial (R) a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..2000; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado



6.1.2.1.2 Escrita

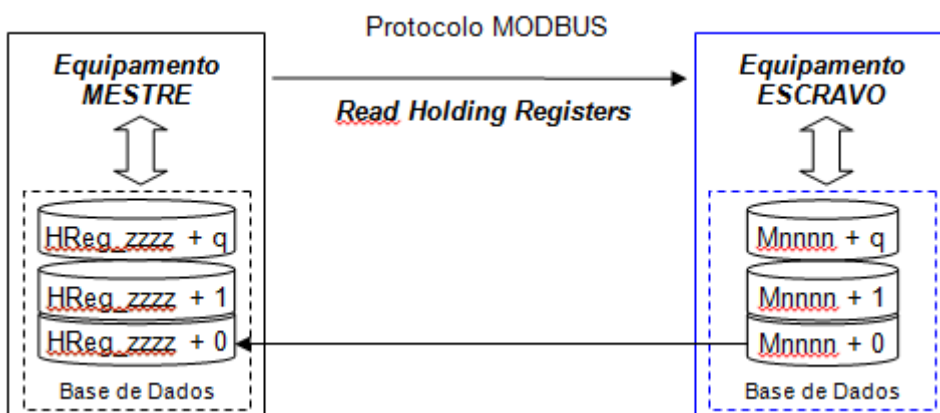
Função MODBUS		WRITE SINGLE COIL WRITE_MULTIPLE_COILS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial (R) a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..1968; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.2.2 Tipo M (Memórias Inteiras)

6.1.2.2.1 Leitura

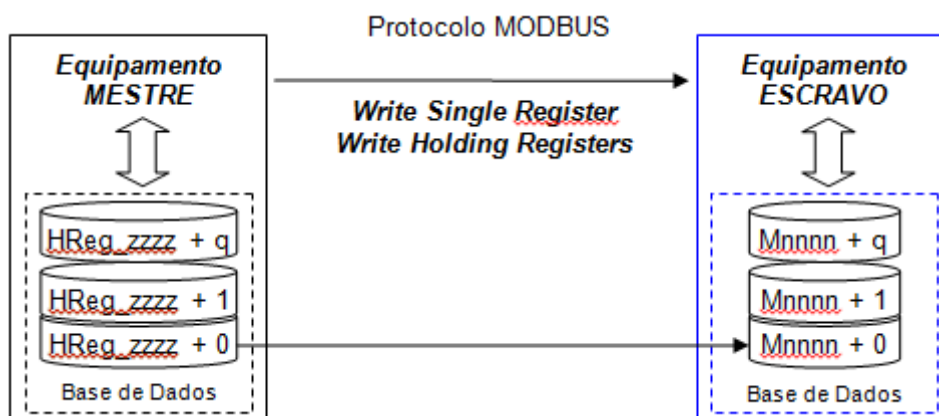
Função MODBUS		READ HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira (M) inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.





6.1.2.2.2 Escrita

Função MODBUS		WRITE SINGLE REGISTER WRITE HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira (M) inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..123; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.1.2.3 Tipo D (Memórias Reais)

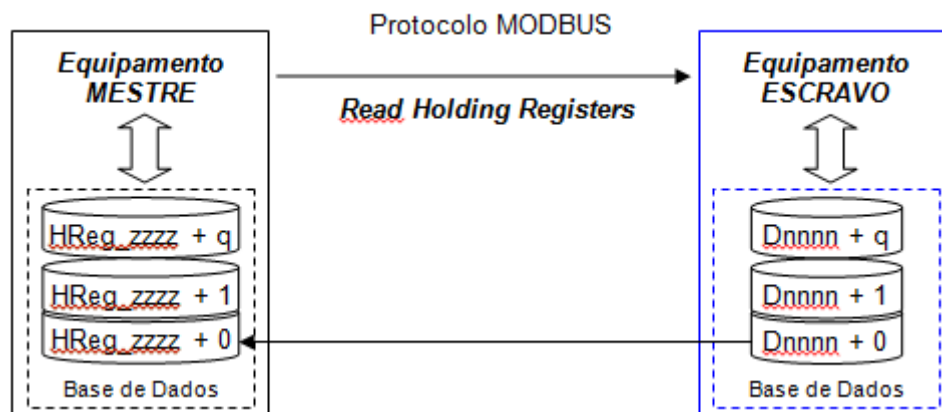
6.1.2.3.1 Leitura

Função MODBUS		READ HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real (D) inicial a ser obtida acrescentado do offset de 10000;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.


Obs.:

O protocolo MODBUS não possui suporte nativo para acesso a variáveis reais. Portanto, os valores reais obtidos do equipamento GII, GII_DUO e G3 remoto serão transferidos para registros do equipamento MODBUS mestre. O tipo de equipamento mestre utilizado definirá onde os valores reais obtidos serão salvos.

Note que, para leitura da base dados de memórias reais (D), utilizando-se o protocolo MODBUS, deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS. Como o tipo de dado D é de 32 bits (4 bytes), deve-se solicitar 2 registros para cada valor real a ser obtido do equipamento remoto.

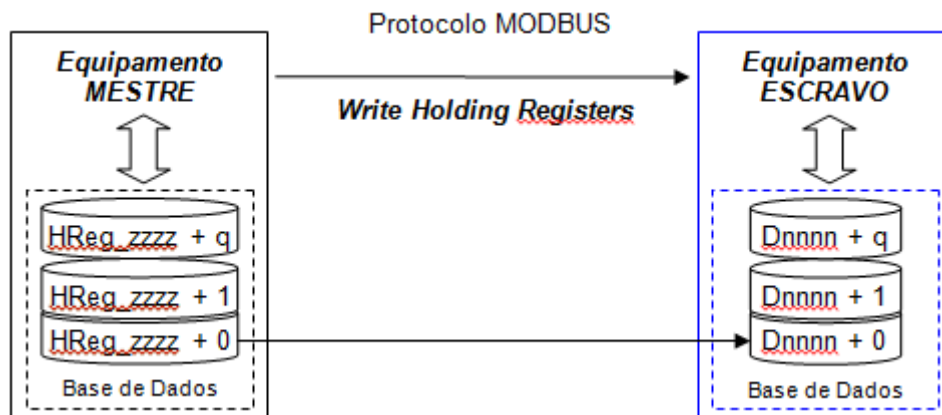

6.1.2.3.2 Escrita

Função MODBUS		WRITE HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real (D) inicial a ser obtida acrescentado do offset de 10000;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..123 O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

Obs.:

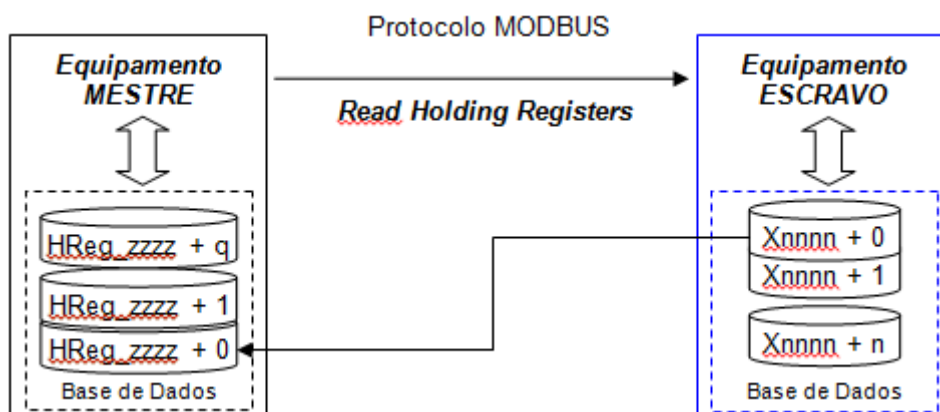
O protocolo MODBUS não possui suporte nativo para acesso a variáveis reais. Portanto, os valores reais escritos no equipamento GII, GII_DUO e G3 remoto devem ser obtidos da base *Holding Register* do equipamento MODBUS mestre. O equipamento mestre é que deve definir onde os valores reais serão obtidos para serem enviados para o equipamento remoto.

Note que, para escrita na base dados de memórias reais D, utilizando-se o protocolo MODBUS, deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS. Como o tipo de dado D é de 32 bits (4 bytes), deve-se enviar 2 registros Holding Register para cada valor real D a ser transferido para o equipamento remoto.



6.1.2.4 Tipo X (Texto)

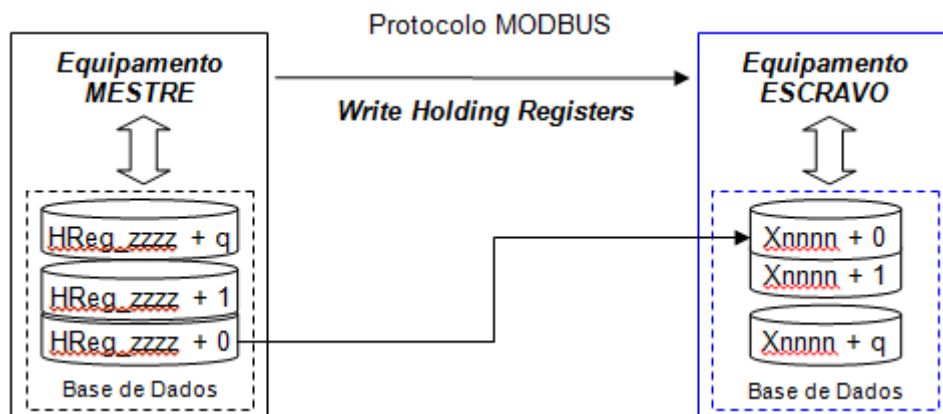
Função MODBUS		READ HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial associada ao início do texto a ser obtido. Para se obter a memória associada ao início do texto utilize a seguinte fórmula: $M_i = MAX_M - ((x + 1) * 25)$ onde: MAX_M número máximo de variáveis do tipo M disponíveis no equipamento associado. x número da variável do tipo Texto a ser identificada.
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.





6.1.2.4.1 Escrita

Função MODBUS		WRITE HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial associada ao início do texto a ser alterado. Para se obter a memória associada ao início do texto utilize a seguinte fórmula: $M_i = MAX_M - ((x + 1) * 25)$ onde: MAX_M número máximo de variáveis do tipo M disponíveis no equipamento associado. x número da variável do tipo Texto a ser identificada.
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..120; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2 Operação dos equipamentos GII, GII_DUO e G3 no modo Mestre

Neste modo de operação, o equipamento GII, GII_DUO e G3 opera como cliente (Mestre), enquanto o equipamento remoto opera como Servidor de Comunicação (Escravo), conforme exemplificado na figura seguinte.



Nesta condição, o equipamento GII, GII_DUO e G3 como Mestre de uma conexão deve montar o comando de requisição para ser enviado para o equipamento remoto. Este comando deve ser montado obedecendo a especificação do protocolo que estiver sendo utilizado na conexão. Para implementar recursos de comunicação independentes do protocolo utilizado, os equipamentos GII, GII_DUO e G3 operam com tipos de dados genéricos que quando especificados, são mapeados nos comandos apropriados de cada protocolo pelo respectivo *driver* de comunicação. Desta forma, o usuário do programa de aplicação especifica o tipo de dado a ser escrito ou lido no equipamento remoto de forma transparente para o protocolo utilizado. Os tipos de dados existentes estão definidos na tabela a seguir:

Tipo de Dado	Código (HEX)	Leitura	Escrita	Definição
DEVICE_TYPE_BOOL_8	000 (00h)	✓		Variável booleana (lógica)
	128 (80h)		✓	
DEVICE_TYPE_INTEGER_16	001 (01h)	✓		Variável inteira de 16 bits (com sinal)
	129 (81h)		✓	
DEVICE_TYPE_REAL_32	002 (02h)	✓		Variável real de 32 bits com ponto flutuante (notação IEEE)
	130 (82h)		✓	
DEVICE_TYPE_ANALOG_I16	005 (05h)	✓		Variável associada à entrada analógica de processo (16 bits)
DEVICE_TYPE_DIGITAL_I16	006 (06h)	✓		Variável associada à entrada digital de processo (mapeada em 16 bits)

Para informações mais detalhadas sobre as funções de comunicação dos PLC's GII, GII_DUO e G3 operando como Mestre consulte a nota de aplicação ENA0004500 (Comunicação Remota com Controladores HI-GII e GII_DUO). Este documento está disponível para *download* em www.hitecnologia.com.br.

6.2.1 Protocolo SCP-HI no modo Mestre

Utilizando o protocolo SCP-HI no modo mestre, é possível enviar e ler no equipamento escravo, variáveis tipo R, M e D. O tipo de dados X (*string*) pode ser acessado, endereçando o conjunto de memórias M correspondentes.

A tabela a seguir apresenta o mapeamento dos tipos genéricos definidos na tabela do item 5.2 e os comandos específicos do protocolo SCP-HI, com os tipos da base de dados enviados ou obtidos:



Tipo de Dado	Leitura	Escrita	Função SCP-HI	Base Mestre	Base Escravo
DEVICE_TYPE_BOOL_8	√		ReadVarType R	R	R
		√	WriteVarType R	R	R
DEVICE_TYPE_INTEGER_16	√		ReadVarType M	M	M
		√	WriteVarType M	M	M
DEVICE_TYPE_REAL_32	√		ReadVarType D	D	D
		√	WriteVarType D	D	D
DEVICE_TYPE_ANALOG_I16	√		ReadVarType M	nd	
DEVICE_TYPE_DIGITAL_I16	√		ReadVarType M	nd	

Obs:

nd – Não disponível no protocolo SCP-HI.

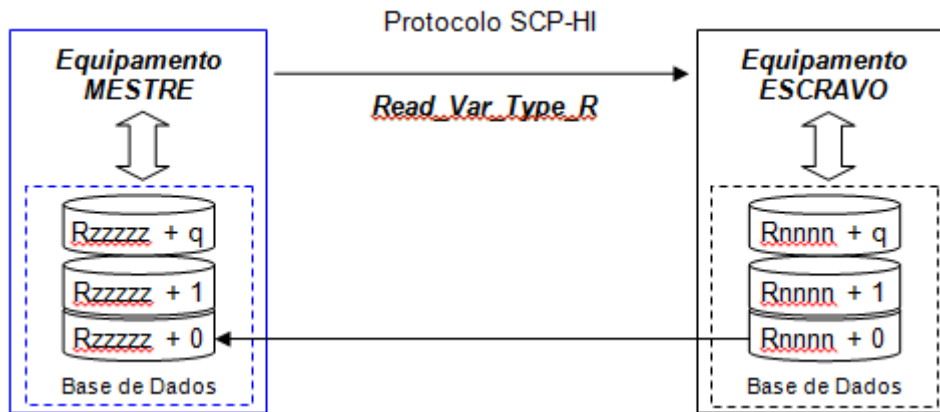
a – Para leitura de variáveis do tipo string deve-se utilizar o tipo genérico *DEVICE_TYPE_INTEGER_16* lendo a partir da memória M inicial da string (vide item 4.1.4), as 25 memórias consecutivas.

b – Para escrita de variáveis do tipo string deve-se utilizar o tipo genérico *DEVICE_TYPE_INTEGER_16* escrevendo a partir da memória M inicial da string (vide item 4.1.4), as 25 memórias consecutivas.

6.2.1.1 Tipo *DEVICE_TYPE_BOOL_8*

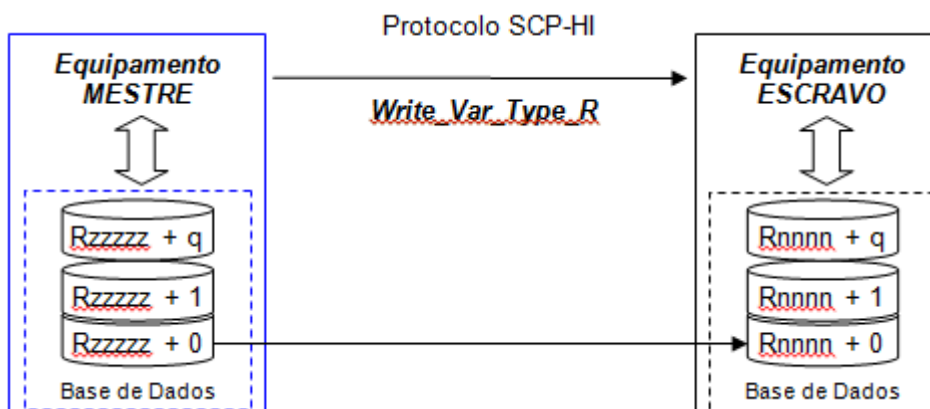
6.2.1.1.1 Leitura

Função SCP-HI		READ_VAR_TYPE_R
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..249; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2.1.1.2 Escrita

Função SCP-HI		WRITE_VAR_TYPE_R
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo contato inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..246; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.





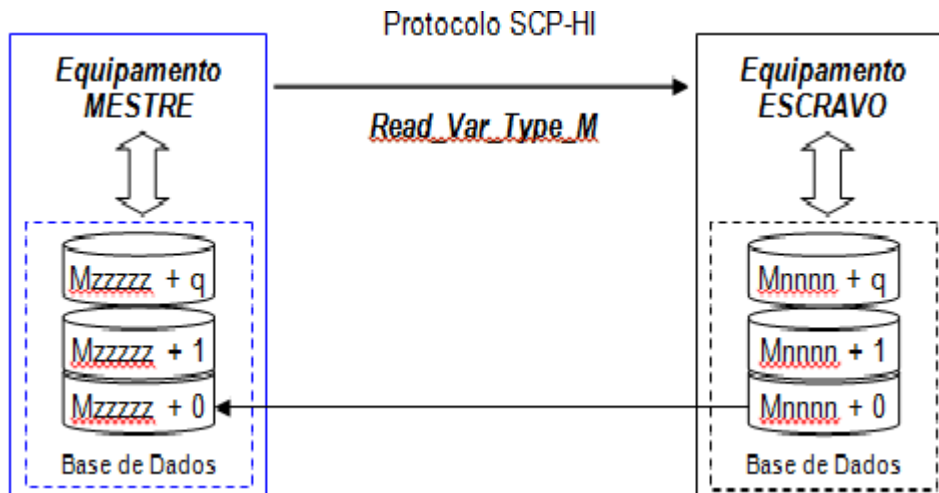
6.2.1.2 Tipo *DEVICE_TYPE_INTEGER_16*

6.2.1.2.1 Leitura

Função SCP-HI		READ_VAR_TYPE_M
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

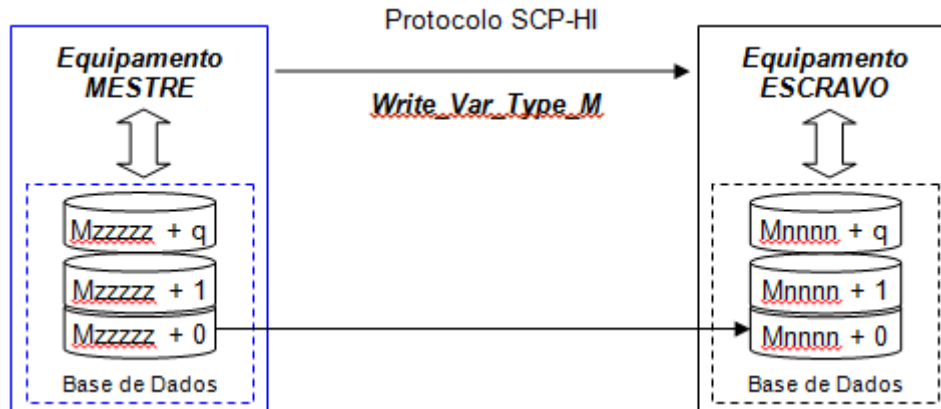
Obs.:

- a – Os tipos de dados ***DEVICE_TYPE_BIT_16***, ***DEVICE_TYPE_ANALOG_I16*** e ***DEVICE_TYPE_DIGITAL_I16***, quando utilizados, mapeiam em comandos de comunicação equivalentes.



6.2.1.2.2 Escrita

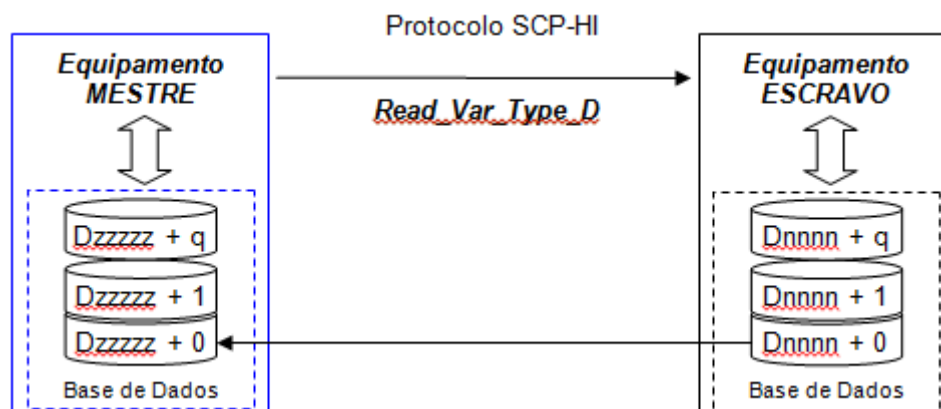
Função SCP-HI		WRITE_VAR_TYPE_M
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..123;



6.2.1.3 Tipo *DEVICE_TYPE_REAL_32*

6.2.1.3.1 Leitura

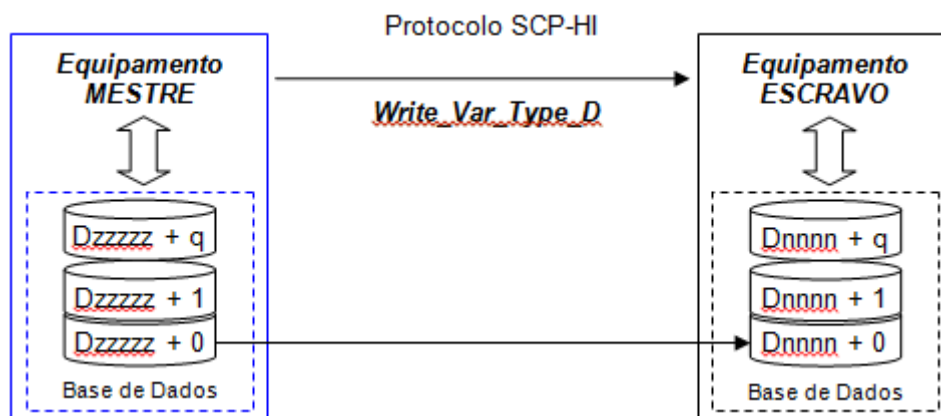
Função SCP-HI		READ_VAR_TYPE_D
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..62; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.





6.2.1.3.2 Escrita

Função SCP-HI		WRITE_VAR_TYPE_D
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..61; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2.2 Protocolo MODBUS no modo Mestre

Utilizando o protocolo MODBUS-RTU ou MODBUS-TCP no modo mestre, é possível enviar e ler no equipamento escravo, variáveis tipo *coil*, *input status*, *input registers* e *holding register*.

A tabela a seguir apresenta o mapeamento dos tipos genéricos definidos na tabela do item 5.2 e os comandos específicos do protocolo MODBUS, com os tipos da base de dados enviados ou obtidos:

Tipo de Dado	Leitura	Escrita	Função MODBUS	Base Mestre	Base Escravo
DEVICE_TYPE_BOOL_8	✓		Read Coils	R	Coils
		✓ ✓	Write Single Coil (*a) Write Multiple Coils	R R	Coil Coils
DEVICE_TYPE_INTEGER_16	✓		Read Holding Registers	M	Holding Registers
		✓	Write Single Register	M	Holding Registers



			(*b) Write Holding Registers		Holding Registers
DEVICE_TYPE_REAL_32	√		Read Holding Registers (*c)	D	Holding Registers
		√	Write Holding Registers (*d)	D	Holding Registers
DEVICE_TYPE_ANALOG_I16	√		Read Input Status	M	Input Status
DEVICE_TYPE_DIGITAL_I16	√		Read Discrete Inputs	M	Discrete Inputs

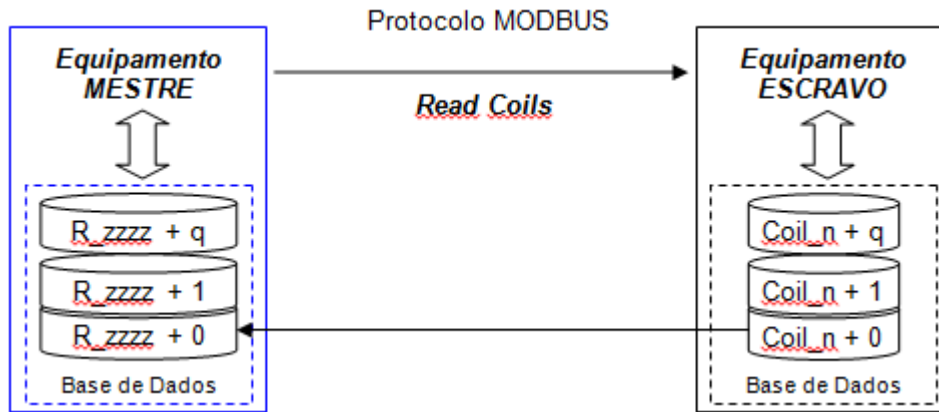
Obs.:

- a – Se a quantidade de variáveis especificada for 1 é utilizada a função MODBUS *WRITE_SINGLE_COIL*, caso contrário é utilizada a função MODBUS *WRITE_MULTIPLE_COILS*.
- b – Se a quantidade de variáveis especificada for 1, é utilizada a função MODBUS *WRITE_SINGLE_REGISTER*, caso contrário é utilizada a função MODBUS *WRITE_MULTIPLE_REGISTERS*.
- c – Quando utilizado o tipo *DEVICE_TYPE_REAL_32* para leitura, a quantidade de variáveis especificada é multiplicada por 2 pelo *driver*, pois a função MODBUS utilizada (*READ_MULTIPLE_REGISTERS*) obtém um valor de 16 *bits* para cada item solicitado.
- d – Quando utilizado o tipo *DEVICE_TYPE_REAL_32* para escrita, a quantidade de variáveis especificada é multiplicada por 2 pelo *driver*, pois a função MODBUS utilizada (*WRITE_MULTIPLE_REGISTERS*) envia um valor de 16 bits para cada item solicitado.

6.2.2.1 Tipo *DEVICE_TYPE_BOOL_8*

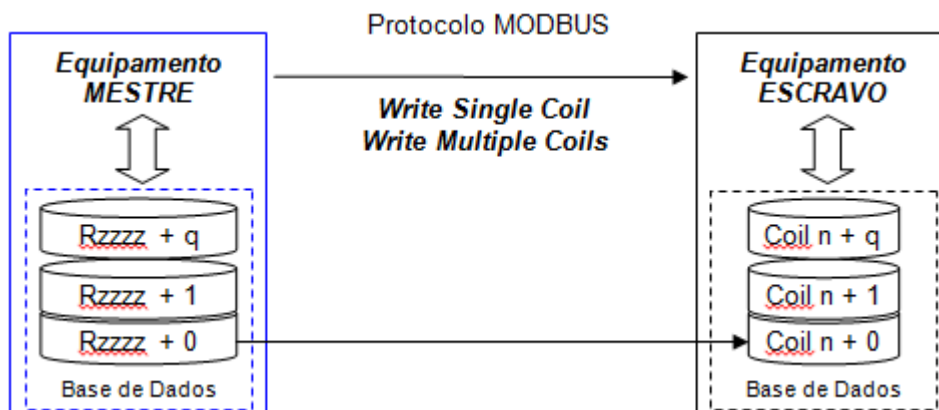
6.2.2.1.1 Leitura

Função MODBUS		READ COILS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo coil a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..2000; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2.2.2 Escrita

Função MODBUS		WRITE SINGLE COIL WRITE_MULTIPLE_COILS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo coil inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..1968; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

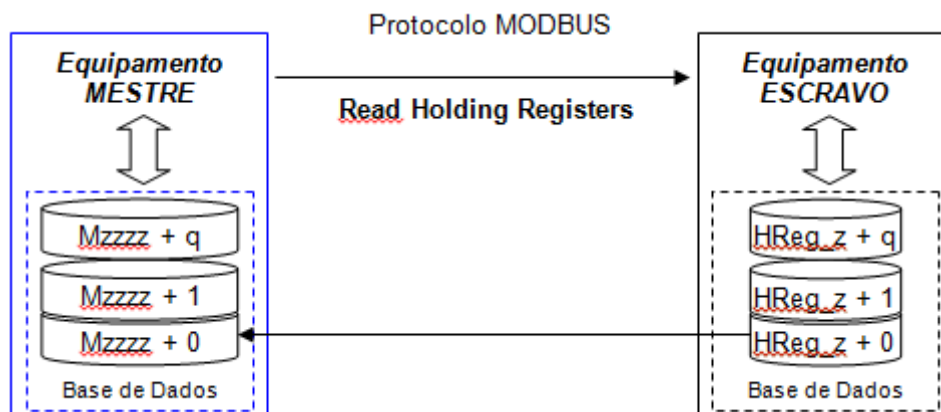




6.2.2.3 Tipo *DEVICE_TYPE_INTEGER_16*

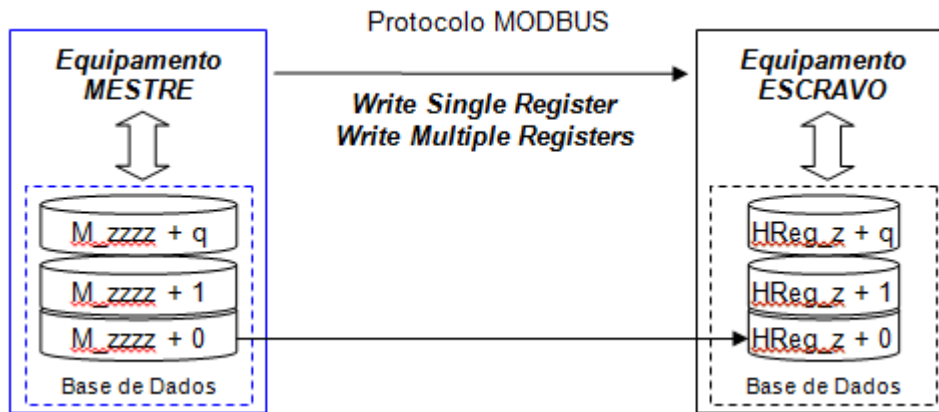
6.2.2.3.1 Leitura

Função MODBUS		READ HOLDING REGISTER
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo Holding Register inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2.2.3.2 Escrita

Função MODBUS		WRITE SINGLE REGISTER WRITE MULTIPLE REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória inteira (M) inicial a ser alterada;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..123; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.2.2.4 Tipo *DEVICE_TYPE_REAL_32*

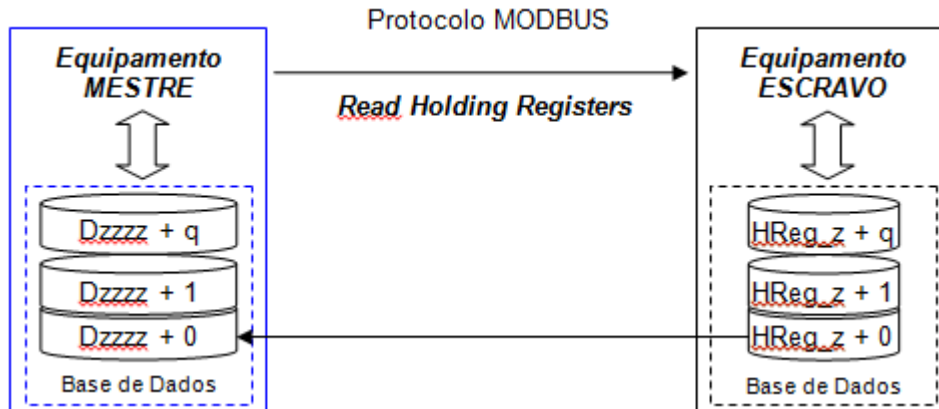
6.2.2.4.1 Leitura

Função MODBUS		READ HOLDING REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real (D) inicial a ser obtida acrescentado do offset de 10000;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..62; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

Obs.:

O protocolo MODBUS não possui suporte nativo para acesso a variáveis reais. Portanto, os valores reais obtidos do equipamento GII, GII_DUO e G3 remoto serão transferidos para registros do equipamento MODBUS mestre. O tipo de equipamento mestre utilizado definirá onde os valores reais obtidos serão salvos.

Note que, para leitura da base dados de memórias reais (D), utilizando-se o protocolo MODBUS, deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS. Como o tipo de dado D é de 32 *bits* (4 *bytes*), o *driver* solicita 2 registros para cada valor real a ser obtido do equipamento remoto.



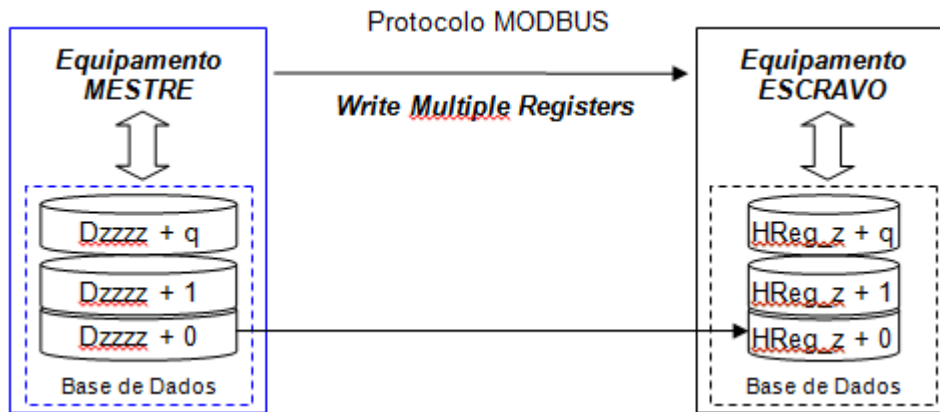
6.2.2.4.2 Escrita

Função MODBUS		WRITE MULTIPLE REGISTERS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo memória real (M) inicial a ser obtida acrescentado do offset de 10000;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..61; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

Obs.:

O protocolo MODBUS não possui suporte nativo para acesso a variáveis reais. Portanto, os valores reais escritos no equipamento GII, GII_DUO e G3 remoto devem ser obtidos da base Holding Register, do equipamento MODBUS mestre. O equipamento mestre definirá onde os valores reais serão obtidos para serem enviados para o equipamento remoto.

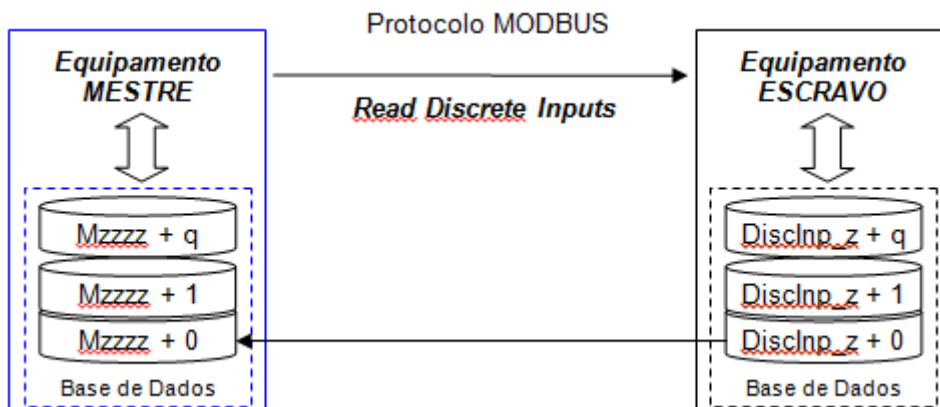
Note que, para escrita na base dados de memórias reais (D), utilizando-se o protocolo MODBUS, deve-se especificar um *offset* de 10000 no campo de endereço inicial do comando MODBUS.



6.2.2.5 Tipo *DEVICE_TYPE_DIGITAL_I16*

6.2.2.5.1 Leitura

Função MODBUS		READ DISCRETE INPUTS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo Discrete Input inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..2000; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.

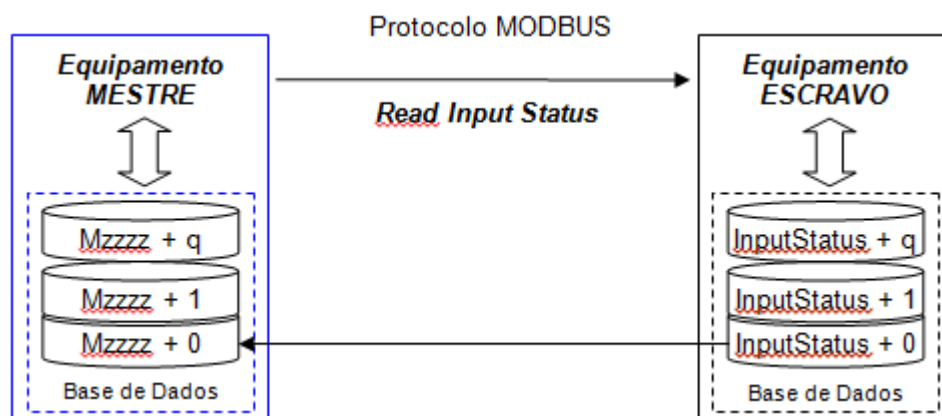




6.2.2.6 Tipo *DEVICE_TYPE_ANALOG_I16*

6.2.2.6.1 Leitura

Função MODBUS		READ INPUT STATUS
Parâmetros	Station	Número do equipamento remoto (escravo);
	VIni	Endereço da variável do tipo Input Status inicial a ser obtida;
	VQuant	Quantidade de variáveis a partir da variável inicial; O parâmetro VQuant pode variar entre 1..125; O limite máximo pode ser menor que o especificado anteriormente dependendo do número máximo de variáveis do tipo, disponíveis no equipamento associado.



6.3 Protocolos ASCII

O protocolo ASCII é normalmente utilizado para interfacear dispositivos que recebem ou enviam informações em forma de texto. Para informações mais detalhadas sobre a utilização do protocolo ASCII nos equipamentos GII, GII_DUO e G3, consulte a nota de aplicação ENA0004300 (Interface dos Controladores HI-GII com dispositivos via protocolo ASCII). Este documento está disponível para *download* em www.hitecnologia.com.br.

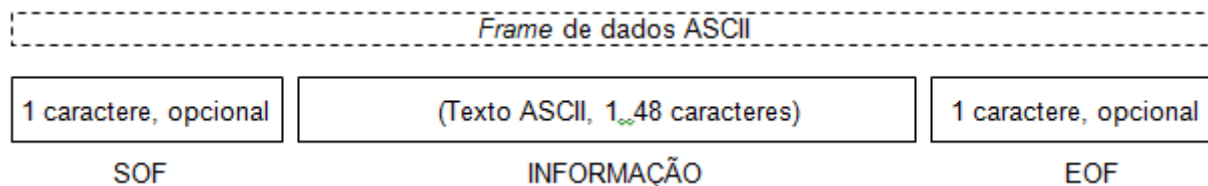
A necessidade de seleção do protocolo ASCII para o respectivo canal de comunicação é dependente do tipo de recurso de comunicação a ser utilizado, conforme detalhado a seguir.

6.3.1 Dispositivos que enviam informações em forma de Texto

Estes dispositivos obtêm informações do processo, formatam estes dados como um Texto e enviam através de canais de comunicação para serem tratados. Como exemplo pode citar: leitores de código de barra, *scanners*, leitores de cartões magnéticos, leitores biométricos etc. Dependendo do equipamento utilizado, o texto enviado



pode ser delimitado com caracteres especiais no início e/ou no final do mesmo. Desta forma o formato geral do *frame* é o seguinte:



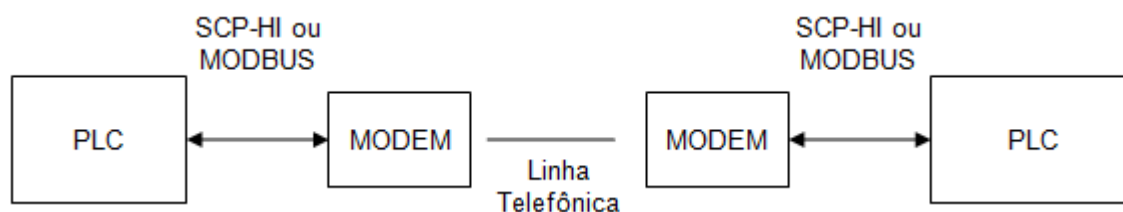
Para receber *frames* com este formato, é necessário que o equipamento GII, GII_DUO e G3 utilizado tenha a sua porta de comunicação configurada para operação com protocolo ASCII.

A especificação dos caracteres SOF e EOF, bem como a habilitação dos mesmos é realizada na configuração do canal de operação associado. Quando habilitada operação com SOF e/ou EOF estes caracteres são filtrados pelo driver de recepção, não sendo armazenados na variável texto Xnnn associada.

6.3.2 Dispositivos que recebem informações em forma de Texto

Estes dispositivos aguardam a recepção de *frames* de Textos pelo canal de comunicação. Como exemplo pode citar: impressoras, modems, painéis de *displays* de *Led's* etc. Neste caso, o *frame* de informação a ser enviado deve ser criado em uma variável de texto (Xnnnn) qualquer do equipamento GII, GII_DUO e G3. A utilização ou não de caracteres de início de *frame* (SOF) ou final de *frame* (EOF) é definida pelo protocolo do equipamento que será conectado e, quando necessária, devem ser inseridos no texto antes do envio pelo canal de comunicação.

É importante notar que, para o envio de um *frame* ASCII qualquer, não é necessário que o canal de comunicação associado esteja configurado como ASCII. A função SCB de transmissão de dados ASCII está operacional independente do tipo de protocolo configurado para o canal de comunicação. Desta forma, pode-se, por exemplo, utilizar um canal serial do equipamento, configurado no protocolo SCP-HI (ou MODBUS -RTU) para enviar informações de conexão para um *modem* e posteriormente comunicar-se com o protocolo configurado, conforme ilustrado a seguir:



6.3.3 Operação de *frames* não ASCII

Todas as aplicações que utilizam comunicação nos exemplos citados anteriormente montam os *frames* de informação utilizando caracteres ASCII padrão. Entretanto o *frame* recebido ou enviado pelas funções de



comunicação não necessita necessariamente de ser ASCII. Qualquer sequência de valores (bytes) pode ser montada e transmitida pelo canal de comunicação bem como, recebida pelo mesmo.

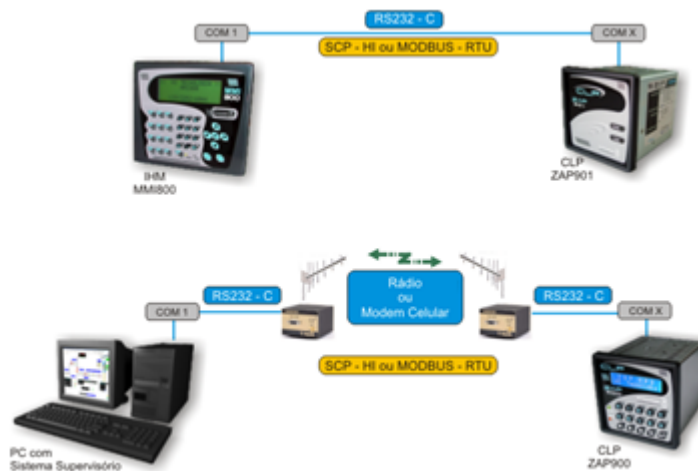
Em outras palavras, o protocolo ASCII permite envio e recepção de um conjunto de *bytes* em qualquer formato. A única restrição imposta pelo *driver* é que a quantidade de dados recebida ou enviada em cada transação de comunicação (envio ou recepção) não seja superior a 48 *bytes*.

7 Arquiteturas de comunicação

Neste capítulo são apresentados exemplos de topologias usuais para comunicação utilizando os protocolos disponíveis nos equipamentos GII, GII_DUO e G3.

7.1 Comunicação Ponto a Ponto

Para este tipo de configuração de comunicação o equipamento que irá controlar o processo de comunicação é configurado como mestre enquanto que o outro é configurado como escravo. Os protocolos suportados para este tipo de comunicação são normalmente SCP-HI, MODBUS -RTU e ASCII. O meio físico de comunicação pode ser RS232-C, RS485, radio/Modem ou Modem celular, conforme indicado nas figuras a seguir:



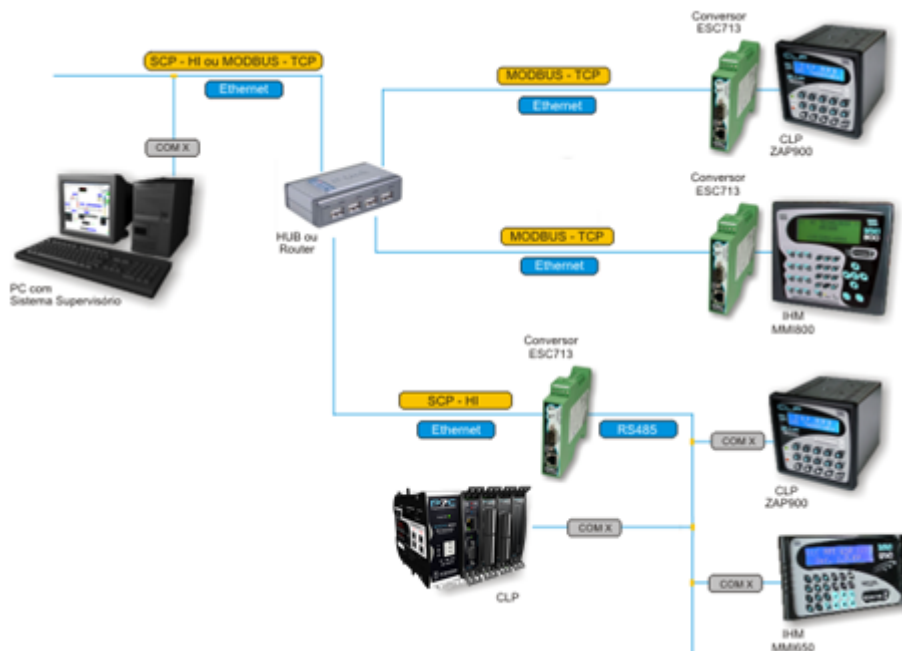
7.2 Comunicação em rede RS485

Para este tipo de configuração de comunicação o equipamento que controlará o processo de comunicação é configurado como mestre enquanto todos os demais equipamentos devem ser configurados como escravo. Os protocolos suportados para este tipo de comunicação são normalmente SCP-HI, MODBUS-RTU. O meio físico de comunicação pode ser RS485 ou radio/Modem, conforme indicado nas figuras a seguir:



7.3 Comunicação em rede *Ethernet*

Para este tipo de configuração de comunicação o equipamento que irá controlar o processo de comunicação é configurado como mestre enquanto todos os demais equipamentos devem ser configurados como escravo. Os protocolos suportados para este tipo de comunicação são normalmente SCP-HI, MODBUS-TCP. O meio físico de comunicação é *Ethernet* com camada TCP-IP ou UDP, conforme indicado na figura a seguir:





Controle do Documento

Considerações gerais

- Este documento é dinâmico, estando sujeito a revisões, comentários e sugestões. Toda e qualquer sugestão para seu aprimoramento deve ser encaminhada ao departamento de suporte ao cliente da **HI Tecnologia Indústria e Comércio Ltda.**, fornecendo os dados especificados na "Apresentação" deste documento.
- Os direitos autorais deste documento são de propriedade da **HI Tecnologia Indústria e Comércio Ltda.**

Controle de Alterações do Documento

Data Liberação	Revisão	Descrição	Elaborado por	Revisado por	Aprovado por
05/01/2017	2	Documento revisado e migrado para o novo ambiente de documentação. Revisada a tabela de controle do documento para manter histórico dos responsáveis por elaboração, revisão e aprovação	N/a	Maria Villela	Isaías Ribeiro
16/08/2013	1	Acrescentadas referências aos controladores P7C GII_DUO e G3	N/a	Wendel Bonfá	Isaías Ribeiro
11/09/2006	0	Documento Original	Heber Scachetti	Wendel Bonfá	Isaías Ribeiro