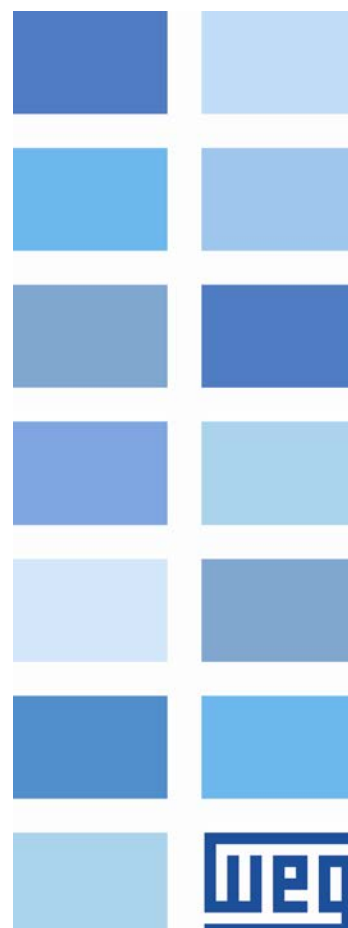


Serial

SIW600 V1.3X

Manual de Comunicação

Idioma: Português





Manual de Comunicação Serial RS485

Série: SIW600

Idioma: Português

Número do Documento: 10004654463 / 00

Versão de Software: 1.3X

Data de Publicação: 12/2016

Sumário

SOBRE O MANUAL	5
ABREVIações E DEFINIções.....	5
REPRESENTAÇÃO NUMÉRICA.....	5
1 INTRODUÇÃO À COMUNICAÇÃO SERIAL	6
2 FUNÇÕES DOS PINOS DO CONECTOR RS485	7
2.1 RS485.....	7
2.2 Conexão com a Rede RS485.....	7
3. PROGRAMAÇÃO DO INVERSOR	9
3.1 SÍMBOLOS E PROPRIEDADES.....	9
3.2 PROGRAMAÇÕES VIA HMI.....	9
P0308 – ENDEREÇO SERIAL.....	9
P0310 – TAXA DE TRANSMISSÃO SERIAL.....	9
P0312 – PROTOCOLO SERIAL.....	10
4 PROTOCOLO MODBUS-RTU	11
4.1 MODOS DE TRANSMISSÃO.....	11
4.2 ESTRUTURA RTU MODO MENSAGEM.....	11
4.2.1 Endereço.....	11
4.2.2 Código de Função.....	11
4.2.3 Campo de Dados.....	12
4.2.4 CRC.....	12
4.2.5 Tempo entre Mensagens.....	12
4.3 OPERAÇÃO DO SIW600 NA REDE MODBUS-RTU.....	13
4.3.1 Funções Disponíveis e Tempos de Resposta.....	13
4.3.2 Endereçamento dos Dados e Offset.....	13
4.4 DESCRIÇÃO DETALHADA DAS FUNÇÕES.....	17
4.4.1 Função 03 – Leitura de Grupos de Registradores.....	17
4.4.2 Função 06 – Escrita em um Registrador.....	18
4.4.3 Função 16 – Escrita em Múltiplos Registradores.....	19
4.4.4 Função 43 – Leitura de Identificação do Produto.....	20
4.4.5 Erros de Comunicação.....	21
5 CONFIGURAÇÃO DA POTÊNCIA REATIVA	23
5.1 FP UNITÁRIO.....	23
5.2 FP FIXO.....	23
5.3 FP VARIÁVEL.....	24
5.4 CONTROLE KVAR.....	25
APÊNDICES	27
APÊNDICE A. TABELA ASCII.....	27
APÊNDICE B. CÁLCULO DO CRC UTILIZANDO TABELAS.....	28
APÊNDICE C. CÁLCULO CRC UTILIZANDO DESLOCAMENTO DE REGISTRADORES.....	29

Sobre o Manual

Este manual fornece a descrição necessária para a operação do inversor solar SIW600, utilizando a interface de comunicação serial RS485.

Abreviações e Definições

ASCII	Código Padrão Americano para o Intercâmbio de Informação
CRC	Verificação cíclica de redundância
EIA	Aliança das indústrias eletrônicas
RTU	Unidade remota de terminação

Representação Numérica

Números decimais são representados através de dígitos sem sufixo.

Números hexadecimais são representados com a letra 'h' depois do número.

1 Introdução à Comunicação Serial

Em uma interface serial, os bits de dados são enviados sequencialmente através de um canal de comunicação ou um barramento. Diversas tecnologias utilizam comunicação serial para a transferência de dados, incluindo as interfaces RS232 e RS485.

As normas que especificam os padrões RS232 e RS485, no entanto, não especificam o formato nem a sequência de caracteres para a transmissão e recepção de dados. Deste modo, além da interface, é necessário identificar também o protocolo utilizado para a comunicação, que é o protocolo Modbus-RTU.

A seguir, as características da interface de comunicação serial RS485 presente no SIW600 serão apresentadas, bem como o protocolo para a utilização desta interface.

2 Funções dos Pinos do Conector RS485

2.1 RS485

Figura 2.1 - Configuração do conector
Device side - female connector / conector fêmea

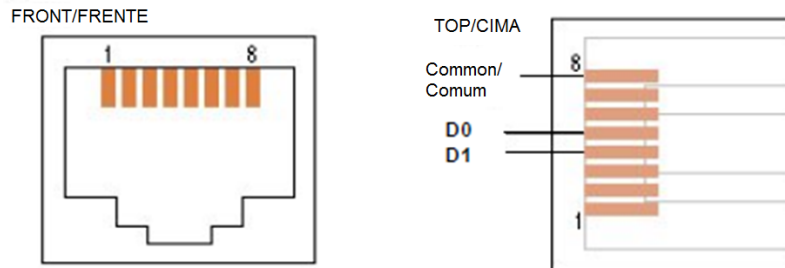


Tabela 2.1 - RJ45/RS485 e atribuições dos pinos do conector

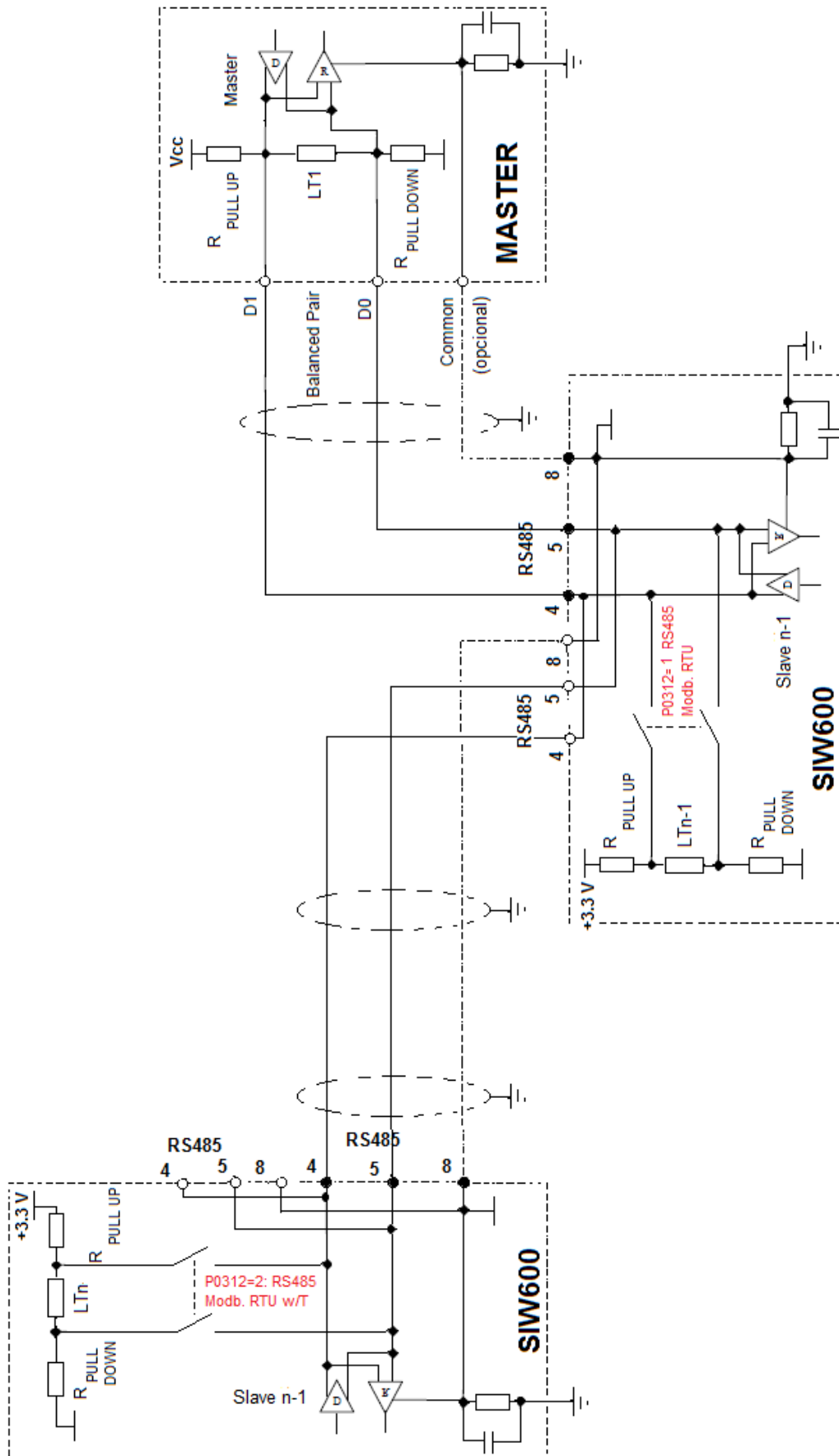
Pino	Nome	Função
1	NC	
2	NC	
3	NC	
4	B/B'(D1)	RxD/TxD Positivo
5	A/A'(D0)	RxD/TxD Negativo
6	NC	
7	NC	
8	GND	0V do circuito RS485 (conectado ao PE (chassis) através de um capacitor ou resistor).

2.2 Conexão com a Rede RS485

Os seguintes pontos devem ser observados para a ligação do inversor, utilizando a interface RS485:

- ☑ É recomendado o uso de um cabo com par trançado blindado;
- ☑ É recomendado também que o cabo possua mais um fio para a conexão do sinal de referência (GND). Caso o cabo não possua o fio adicional, deve-se deixar o sinal GND desconectado;
- ☑ A passagem do cabo deve ser feita separadamente (se possível distante) dos cabos para a alimentação de potência;
- ☑ Todos os dispositivos de rede devem estar devidamente aterrados, preferencialmente na mesma ligação ao terra. A blindagem do cabo também deve estar aterrada;
- ☑ Habilitar os resistores de terminação em apenas dois pontos, nos extremos do barramento principal, mesmo que existam derivações a partir do barramento. Para ativar os resistores de terminação, é preciso ativar internamente: P0312=2: RS485 Modb. RTU w/T. Ver Figura 2 para detalhes.

Figura 2.2 - Conexão com a Rede RS485



3 Programação do Inversor

3.1 Símbolos e Propriedades

Símbolo	Descrição
RO	"Apenas Leitura". Parâmetro visível através do display LCD ou pela comunicação serial.
RO-Serial	"Apenas Leitura". Parâmetro visível apenas pela comunicação serial.
RW	"Leitura/Escrita". Parâmetro que pode ser lido ou alterado através do display LCD ou da comunicação serial.
RW-Serial	"Leitura/Escrita". Parâmetro que pode ser lido ou alterado apenas através da comunicação serial.

3.2 Programações Via HMI

Os parâmetros a seguir, são utilizados para configurar a comunicação serial.

P0308 – Endereço serial

Faixa de

Valores: 1 até 247

Padrão: 1

Propriedades: RW

Grupo de acesso via HMI:

01 GRUPO DE PARÂMETROS.

└ 02 Parametrização.

└ 03 P0308 Endereço Serial.

Descrição:

Permite a programação do endereço utilizado para a comunicação serial do inversor. É necessário que cada dispositivo existente na rede possua um endereço diferente. Os endereços válidos para este parâmetro são de 1 até 247 (Modbus-RTU).

P0310 – Taxa de Transmissão Serial

Faixa de valores: 0 = 9600 bits/s

Padrão: 0

1 = 19200 bits/s

2 = 38400 bits/s

3 = 57600 bits/s

Propriedades: RW

Grupo de acesso via HMI:

01 GRUPO DE PARÂMETROS.

└ 02 Parametrização.

└ 03 P0310 - Taxa de Transmissão Serial.

Descrição:

Permite a programação da taxa de transmissão desejada para a interface serial, em bits por segundo. A taxa deve ser igual para todos os dispositivos ligados à rede.



NOTA!

A configuração da interface serial é fixa: 8 bits de dados, sem paridade, 2 stop bits.

P0312 – Protocolo Serial

Faixa de valores : 0 = Inativo **Padrão:** 0
1 = RS485 Modb. RTU
2 = RS485 Modb. RTU w/T

Propriedade: RW

Grupo de acesso via HMI:

01 GRUPO DE PARÂMETROS.

└ 02 Parametrização.

└└ 03 P0312 – Protocolo Serial.

Descrição:

Permite a seleção do protocolo para a interface serial, nesse caso o Modbus RTU.

Ao ativar o Modbus-RTU através da opção 2, é possível ativar a terminação de 120Ω através de um interruptor.

A descrição detalhada do protocolo Modbus-RTU está nos próximos tópicos deste manual.

4 Protocolo Modbus-RTU

O protocolo Modbus-RTU foi inicialmente desenvolvido em 1979. Hoje em dia, é um protocolo aberto amplamente difundido, sendo utilizado em diversos equipamentos. A comunicação Modbus-RTU do inversor solar SIW600 foi desenvolvida utilizando como base os seguintes documentos:

- ☑ *MODBUS Protocol Reference Guide Rev. J*, MODICON, June 1996.
- ☑ *MODBUS Application Protocol Specification*, MODBUS.ORG, May 8th 2002.
- ☑ *MODBUS over Serial Line*, MODBUS.ORG, December 2nd 2002.

Estes documentos definem o formato das mensagens utilizadas para os elementos que constituem a rede Modbus, os serviços (ou funções) que podem ser disponibilizados, assim como a forma que estes elementos trocam dados na rede.

4.1 Modos de Transmissão

Estão definidos dois modos de transmissão na especificação do protocolo: ASCII e RTU. Os modos definem a maneira como os bytes da mensagem são transmitidos. Não é possível utilizar dois modos de transmissão em uma mesma rede.

O inversor solar SIW600 utiliza o modo RTU para a transmissão do telegrama. Os bytes são transmitidos em formato hexadecimal.

4.2 Estrutura RTU Modo Mensagem

A estrutura Modbus-RTU utiliza o Sistema mestre – escravo para a troca de mensagens. Ele permite até 247 escravos, mas apenas um mestre. Toda comunicação inicia com o mestre realizando uma solicitação a um escravo, que responde ao mestre com o que lhe foi solicitado. Em ambos os telegramas (pergunta e resposta), a estrutura utilizada é a mesma: Endereço, Código da Função, Dados e CRC.

Apenas o campo de dados pode ter um tamanho variável, dependendo do que está a ser solicitado.

- **Mestre (telegrama de pedido):**

Endereço (1 byte)	Função (1 byte)	Campo de dados de solicitação (n bytes)	CRC (2 bytes)
----------------------	--------------------	--	------------------

- **Slave (telegrama de resposta):**

Endereço (1 byte)	Função (1 byte)	Campo de dados de resposta (n bytes)	CRC (2 bytes)
----------------------	--------------------	---	------------------

4.2.1 Endereço

O mestre inicia a comunicação, enviando o byte com o endereço do escravo, para o qual a mensagem é destinada. Ao enviar a resposta, o escravo também inicia o telegrama com o seu próprio endereço. O mestre também pode enviar uma mensagem para o endereço de 0 (zero), o que significa que, a mensagem está destinada a todos os escravos na rede (*broadcast*). Neste caso, nenhum escravo irá responder ao mestre.

4.2.2 Código de Função

Este campo também contém um único byte, onde o mestre especifica o tipo de serviço ou função solicitada ao escravo (leitura, escrita, etc.). De acordo com o protocolo, cada função é utilizada para acessar um tipo específico de dado.

No SIW600, os dados relativos aos parâmetros estão registrados como registradores do tipo *holding* (ver item 4.3.2).

4.2.3 Campo de Dados

Campo com tamanho variável. O formato e conteúdo deste campo dependem da função utilizada e dos valores transmitidos. Este campo está descrito juntamente com a descrição das funções (ver Tabela 4.2).

4.2.4 CRC

A última parte do telegrama é o campo para checagem de erros de transmissão. O método utilizado é o CRC-16 (Checagem cíclica de redundância). Este campo é formado por dois bytes, onde primeiro é transmitido o byte menos significativo (CRC-), e depois o mais significativo (CRC+). A forma de cálculo do CRC é descrita na especificação do protocolo, informações para sua implementação são fornecidas nos Apêndices B e C.

4.2.5 Tempo entre Mensagens

No modo RTU não existe um caractere específico que indique o início ou o fim de um telegrama. A indicação de quando uma nova mensagem começa ou quando ela termina é feita pela ausência de transmissão de dados na rede, por um tempo mínimo de 3,5 vezes o tempo de transmissão de um byte de dados (11 bits¹). Sendo assim, caso um telegrama tenha iniciado após a decorrência deste tempo mínimo, os elementos da rede irão assumir que o primeiro caractere recebido representa o início de um novo telegrama. E da mesma forma, os elementos da rede irão assumir que o telegrama chegou ao fim, quando recebidos os bytes do telegrama, este tempo decorra novamente.

Se durante a transmissão de um telegrama, o tempo entre os bytes for maior que este tempo mínimo, o telegrama será considerado inválido, pois o inversor irá descartar os bytes já recebidos e montará um novo telegrama com os bytes que estiverem sendo transmitidos.

Para taxas de comunicação superiores a 19200 bits/s, os tempos utilizados são os mesmos que para esta taxa. A Tabela 4.1 mostra os tempos para diferentes taxas de comunicação:

Figura 4.1 - Telegrama e o tempo de transmissão

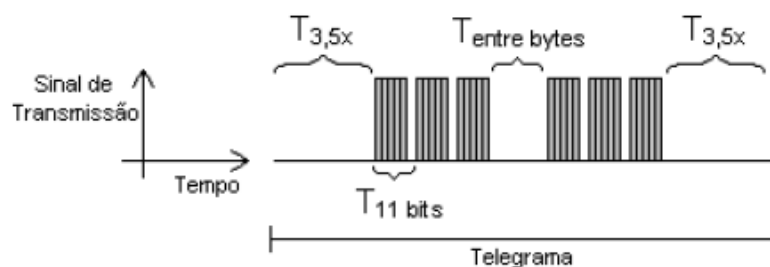


Tabela 4.1 - Taxas de comunicação e tempos envolvidos na transmissão de telegramas

Taxa de comunicação	$T_{11 \text{ bits}}$	$T_{3,5x}$
9600 bits/s	1,146 ms	4,010 ms
19200 bits/s	573 μ s	2,005 ms
38400 bits/s	573 μ s	2,005 ms
57600 bits/s	573 μ s	2,005 ms

- $T_{11 \text{ bits}}$ = Tempo para transmitir uma palavra do telegrama.
- $T_{\text{between bytes}}$ = Tempo entre bytes (não pode ser maior que $3,5x$).
- $T_{3,5x}$ = Intervalo mínimo para indicar começo e fim de telegrama ($3,5 \times T_{11 \text{ bits}}$).

¹ Sempre é considerado o tempo de 11 bits como o tempo para transmissão de um byte, mesmo que no parâmetro P0311 seja programado um formato de telegrama onde cada byte possua apenas 10 bits.

4.3 Operação do SIW600 na Rede Modbus-RTU

○ SIW600 possui as seguintes características quando operando em rede Modbus-RTU:

- ☑ Conexão da rede via interface serial RS-485 (ver item 2);
- ☑ Endereçamento, taxa de transmissão e formato dos bytes definidos através de parâmetros (ver item Tabela 4.2);
- ☑ Permite a parametrização e controle do inversor através do acesso aos parâmetros.

4.3.1 Funções Disponíveis e Tempos de Resposta

Na especificação do protocolo Modbus-RTU são definidas funções utilizadas para acessar diferentes tipos de registradores. No SIW600, os parâmetros foram definidos como sendo registradores do tipo *holding*. Para acessar estes registradores, foram disponibilizados os seguintes serviços (ou funções):

- ☑ *Read Holding Registers*
Descrição: leitura de bloco de registradores do tipo *holding*.
Código de função: 03.
- ☑ *Write Single Register*
Descrição: escrita em um único registrador do tipo *holding*.
Código de função: 06.
- ☑ *Write Multiple Registers*
Descrição: escrita em bloco de registradores do tipo *holding*.
Código de função: 16.
- ☑ *Read Device Identification*
Descrição: identificação do modelo do drive.
Código de função: 43.

○ tempo de resposta do inversor SIW600, do final na transmissão do mestre até o início da resposta do escravo varia de 50 a 100 ms, para qualquer uma das funções acima.

4.3.2 Endereçamento dos Dados e Offset

○ endereçamento dos dados no SIW600 é feito com offset igual a zero, o que significa que o número do endereço equivale ao número dado. Os parâmetros são disponibilizados a partir do endereço 0 (zero). A Tabela 4.2 ilustra o endereçamento dos parâmetros, que podem ser acessados como registradores do tipo *holding*:

Tabela 4.2 – Endereço dos dados para a interface Modbus-RTU

Parâmetros								
Número Do Parâmetro	Descrição	Endereço de dados Modbus		R/W via Serial	Unit	Tipo de Dados	Faixa de Valores	Padrão de fabrica
		Decimal	Hexadecimal					
P0002	Tensão de Linha CA	0	0000h	R	V	Float32	0 até 2000	-
		1	0001h					
P0003	Corrente de Linha	2	0002h	R	A	Float32	-1000 até 1000	-
		3	0003h					
P0004	Tensão CC	4	0004h	R	V	Float32	-2000 até 2000	-
		5	0005h					
P0007	Frequência da Rede	6	0006h	R	Hz	Float32	40 até 70	-
		7	0007h					
P0009	Potência Ativa CA	8	0008h	R	kW	Float32	-1000.0 até 1000.0	-
		9	0009h					
P0010	Potência Reativa CA	10	000Ah	R	KVAr	Float32	-3276.0 até 3276.0	-
		11	000Bh					
P0011	Fator de Potência	12	000Ch	R		Float32	0 até 1.00	-
		13	000Dh					
P0023	Versão de Software	14	000Eh	R		Uint16	0 até 65535	-
-	§	15	000Fh	RW		-	-	-
P0030	Temperatura IGBTs Fase R	16	0010h	R	°C	Float32	-20.0 até +150.0	-
		17	0011h					
P0031	Temperatura IGBTs Fase S	18	0012h	R	°C	Float32	-20.0 até +150.0	-
		19	0013h					
P0032	Temperatura IGBTs Fase T	20	0014h	R	°C	Float32	-20.0 até +150.0	-
		21	0015h					
P0033	Temperatura IGBTs Boost	22	0016h	R	°C	Float32	-20.0 até +150.0	-
		23	0017h					
P0034	Temperatura Interna	24	0018h	R	°C	Float32	-20.0 até +150.0	-
		25	0019h					
P0042	Horas Energizado	26	001Ah	R	H	Float32	0 até 760.000	-
		27	001Bh					
P0043	Horas Habilitado	28	001Ch	R	H	Float32	0 até 438000	-
		29	001Dh					
P0044	Contador kWh	30	001Eh	R	kWh	Float32	0 até 3,504,00	-
		31	001Fh					
P0045	Horas Ventil. Ligado	32	0020h	R	H	Float32	0 até 175200	-
		33	0021h					
P0046	Estado Inversor/Boost	34	0022h	R		Enum(Int16)	0=Pronto 1=Ger Energia 2=Sync 3=Falha 4=Alarme 5=Startup	-
-	§	35	0023h	RW		-	-	-
P0047	Humidade Relativa Interna	36	0024h	R	%	Float32	0 até 100.0	-
		37	0025h					
P0048	Alarme Atual	38	0026h	R		Uint16	0 até 999	-
-	§	39	0027h	RW		-	-	-
P0049	Falha Atual	40	0028h	R		Uint16	0 até 999	-
-	§	41	0029h	RW		-	-	-
P0050	Última Falha	42	002Ah	R		Uint16	0 até 999	-
-	§	43	002Bh	RW		-	-	-
P0051	Dia/Mês Última Falha	44	002Ch	R		Uint16	0000 até 3112 (DD:MM)	-
-	§	45	002Dh	RW		-	-	-
P0052	Ano Última falha	46	002Eh	R		Uint16	0 até 99	-
-	§	47	002Fh	RW		-	-	-
P0053	Hora Última falha	48	0030h	R		Uint16	0000 até 2359 (HH:MM)	-
-	§	49	0031h	RW		-	-	-
P0194	Dia	50	0032h	RW		Uint16	1 até 31	01
-	§	51	0033h	RW		-	-	-

Tabela 4.2 – Endereço dos dados para a interface Modbus-RTU

Número Do Parâmetro	Descrição	Endereço de dados Modbus		R/W via Serial	Unit	Tipo de Dados	Faixa de Valores	Padrão de fábrica
		Decimal	Hexadecimal					
		P0195	Mês					
-	§	53	0035h	RW		-	-	-
P0196	Ano	54	0036h	RW		Uint16	0 até 99	16
-	§	55	0037h	RW		-	-	-
P0197	Hora	56	0038h	RW		Uint16	0 até 23	00
-	§	57	0039h	RW		-	-	-
P0198	Minutos	58	003Ah	RW		Uint16	0 até 59	00
-	§	59	003Bh	RW		-	-	-
P0199	Segundos	60	003Ch	RW		Uint16	0 até 59	00
-	§	61	003Dh	RW		-	-	-
P0200	Senha	62	003Eh	RW		Uint16	00000 até 65535	00005
-	§	63	003Fh	RW		-	-	-
P0201	Idioma	64	0040h	RW		Enum(Int16)	0=Português 1=Inglês	0
-	§	65	0041h	RW		-	-	-
P0204	Carregar/Salvar Parâmetros	66	0042h	RW		Enum(Int16)	0=Sem Função 1=Sem Função 2=Reset P0045 3=Reset P0043 4=Reset P0044 5=Padrão Fábrica 6=Reset Hist. Falhas	0
-	§	67	0043h	RW		-	-	-
P0295	Corr. Nominal Inversor	68	0044h	R		Enum(Int16)	0	0
-	§	69	0045h	RW		-	-	-
P0296	Tensão de Linha da Rede	70	0046h	RW		Enum(Int16)	0=380V 1=440V	0
-	§	71	0047h	RW		-	-	-
P0300	Código do País	72	0048h	R		Enum(Int16)	0	0
-	§	73	0049h	RW		-	-	-
P0308	Endereço Serial	74	004Ah	RW		Uint16	1 até 247	1
-	§	75	004Bh	RW		-	-	-
P0310	Taxa Comunic. Serial	76	004Ch	RW		Enum(Int16)	0=9600 bits/s 1=19200 bits/s 2=38400 bits/s 3=57600 bits/s	0
-	§	77	004Dh	RW		-	-	-
P0312	Protocolo Serial	78	004Eh	RW		Enum(Int16)	0=OFF 1=RS485 Modb. RTU 2=RS485 Modb. RTU w/T	0
-	§	79	004Fh	RW		-	-	-
P0320	Tensão Fase R	80	0050h	R	V	Float32	0 até 2000	-
		81	0051h					
P0321	Tensão Fase S	82	0052h	R	V	Float32	0 até 2000	-
		83	0053h					
P0322	Tensão Fase T	84	0054h	R	V	Float32	0 até 2000	-
		85	0055h					
P0323	Potência Arranjo FV1	86	0056h	R	kW	Float32	-1000.0 até 1000.0	-
		87	0057h					
P0324	Tensão Arranjo FV1	88	0058h	R	V	Float32	-2000.0 até 2000.0	-
		89	0059h					
P0325	Corrente Arranjo FV1	90	005Ah	R	A	Float32	-1000 até 1000	-
		91	005Bh					
P0326	Potência Arranjo FV2	92	005Ch	R	kW	Float32	-1000.0 até	-

Tabela 4.2 – Endereço dos dados para a interface Modbus-RTU

Parâmetros								
Número Do Parâmetro	Descrição	Endereço de dados Modbus		R/W via Serial	Unit	Tipo de Dados	Faixa de Valores	Padrão de fabrica
		Decimal	Hexadecimal					
		93	005Dh				1000.0	
P0327	Tensão Arranjo FV2	94	005Eh	R	V	Float32	-2000 até 2000	-
		95	005Fh					
P0328	Corrente Arranjo FV2	96	0060h	R	A	Float32	-1000 até 1000	-
		97	0061h					
P0920	Corrente I Fase R	98	0062h	R	A	Float32	-1000 até 1000	-
		99	0063h					
P0921	Corrente I Fase S	100	0064h	R	A	Float32	-1000 até 1000	-
		101	0065h					
P0922	Corrente I Fase W	102	0066h	R	A	Float32	-1000 até 1000	-
		103	0067h					
P0926	Tensão DC- Ponto Médio	104	0068h	R	V	Float32	-2000 até 2000	-
		105	0069h					
P0927	Tensão DC+ Ponto Médio	106	006Ah	R	V	Float32	-2000 até 2000	-
		107	006Bh					
P0928	Tensão N-Rede N-Inv.	108	006Ch	R	V	Float32	-2000 até 2000	-
		109	006Dh					
P0965	Resposta do Modo de Geração	110	006Eh	RW		Enum(Int16)	0=FP unitário 1=FP fixo 2=FP Variável 3=Controle kVAr	0
-	§	111	006Fh	RW		-	-	-
P0966	FP Fixo	112	0070h	RW		Float32	0.900 até 0.999	0.999
		113	0071h					
P0967	Modo FP Fixo	114	0072h	RW		Enum(Int16)	0=Indutivo 1=Capacitivo	0
-	§	115	0073h	RW		-	-	-
P0975	Modo de Partida	116	0074h	RW		Enum(Int16)	0=Local (Automático) 1=Remoto (Modbus)	0
-	§	117	0075h	RW		-	-	-
P0976	Partida Remota	118	0076h	RW		Enum(Int16)	0=Inativa 1=Ativa	0
-	§	119	0077h	RW		-	-	-
P10003	Revisão SVN DSP	120	0078h	RW		UInt16	0 até 65535	0
-	§	121	0079h	RW		-	-	-
P10004	Revisão SVN ARM	122	007Ah	R		UInt16	0 até 65535	0
-	§	123	007Bh	RW		-	-	-
P10005	Número serial	124	007Ch	R		UInt32	0 até 65535	-
		125	007Dh					
P10006	Tipo de dispositivo	126	007Eh	R		UInt16	0 até 65535	600
-	§	127	007Fh	RW		-	-	-
P10007	Potência nominal	128	0080h	R	W	UInt32	-	20000
		129	0081h					
-	#	130	007Ch	RW		Unit16	0 até 65535	0
-	#	131	007Dh	RW		-	-	-
-	#	132	007Eh	RW	%	Float32	0 até 100	0
		133	007D					
P10008	Limitação da Potência Ativa	134	007E	RW	%	Float32	0 até 100	100.0
-	#	135	007Fh	RW		-	-	-
P0970	Potência Reativa	136	0080h	RW	%	Float32	-50 até 50	0
-	§	137	0081h	RW	%			
P0990	Tensão inicial Func FPxP	138	0082h	RW	%	Float32	90 até 110	104
-	§	139	0083h	RW	%			
P0991	Tensão final Func FPxP	140	0084h	RW	%	Float32	90 até 110	100

Tabela 4.2 – Endereço dos dados para a interface Modbus-RTU

Número Do Parâmetro	Descrição	Endereço de dados Modbus		R/W via Serial	Unit	Tipo de Dados	Faixa de Valores	Padrão de fábrica
		Decimal	Hexadecimal					
-	§	141	0085h	RW	%			
P0992	Potência Ponto A Func FPxP	142	0086h	RW	%	Float32	5 até 100	20
-	§	143	0087h	RW	%			
P0993	FP Ponto A Func FPxP	144	0088h	RW		Float32	0.9 até 0.999	0.999
-	§	145	0089h	RW				
P0994	Potência Ponto B Func FPxP	146	008Ah	RW	%	Float32	5 até 100	50
-	§	147	008Bh	RW	%			
P0995	FP Ponto B Func FPxP	148	008Ch	RW		Float32	0.9 até 0.999	0.999
-	§	149	008Dh	RW				
P0996	FP Ponto C Func FPxP	150	008Eh	RW		Float32	0.9 até 0.999	0.900
P	§	151	008Fh	RW				
P0126	Tensão Inicial de Varredura	1152	0480h	RW	V	Float32	250 até 1000	400
-	§	1153	0481h	RW				



NOTA!

- Todos os parâmetros são tratados como registradores do tipo *holding*. Dependendo do mestre utilizado, estes registros podem ser referenciados a partir do endereço base 40000 ou 4x. Neste caso, no entanto, em que o endereço tem de ser programado no mestre para um parâmetro, é o endereço mostrado na Tabela 4.2. Consulte a documentação do mestre para saber como acessar registradores do tipo *holding*;
- §: Registradores reservados sem nenhuma função. Estes retornam 0x0000 quando lidos;
- #: Registradores internos da WEG. Uso proibido.

4.4 Descrição Detalhada das Funções

Neste item é feita uma descrição detalhada das funções disponíveis no SIW600 para comunicação Modbus-RTU. Para a elaboração dos telegramas, é importante observar o seguinte:

- Os valores são sempre transmitidos em hexadecimal.
- O endereço de um dado, o número de dados e o valor dos registradores são sempre representados em 16 bits. Por isso, é necessário transmitir estes campos utilizando dois bytes – superior (*high*) e inferior (*low*);
- Os telegramas, tanto para pergunta, quanto para resposta, não podem ultrapassar 64 bytes;
- Os valores transmitidos são sempre representados em bytes. Um registro é sempre composto por dois bytes. O tipo *float* é composto por dois registradores. O mestre deve converter os bytes para o valor nativo.

4.4.1 Função 03 – Leitura de Grupos de Registradores

Essa função lê o conteúdo de um grupo de registradores, que necessariamente devem estar em sequência numérica. Esta função possui a seguinte estrutura para os telegramas de leitura e resposta (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do registrador inicial (byte high)	Campo Byte Count
Endereço do registrador inicial (byte low)	Dado 1 (high)
Número de registradores (byte high)	Dado 1 (low)
Número de registradores (byte low)	Dado 2 (high)
CRC-	Dado 2 (low)
CRC+	etc...
	CRC-
	CRC+

Exemplo 1: Leitura do idioma (P0201) e Carregar/Salvar Parâmetro (P0204) do SIW600 localizado no endereço 1 (assumindo que P0201 = 1 e P0204 = 0).

- Endereço: 1 = 01h (1 byte)
- Número do primeiro parâmetro: 64 = 0040h (2 bytes)
- Valor do primeiro parâmetro: 1 = 0001h (2 bytes)
- Valor do registrador reservado: 0 = 0000h (2 bytes)
- Valor do Segundo parâmetro: 0 = 0000h (2 bytes)

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do escravo	01h	Endereço do escravo	01h
Função	03h	Função	03h
Registrador inicial (high)	00h	Byte Count	06h
Registrador inicial (low)	40h	P0201 (high)	00h
Número de registros (high)	00h	P0201 (low)	01h
Número de registros (low)	03h	Registrador Reservado (high)	00h
CRC-	04h	Registrador Reservado (low)	00h
CRC+	1Fh	P0204 (high)	00h
		P0204 (low)	00h
		CRC-	1Ch
		CRC+	B5h

4.4.2 Função 06 – Escrita em um Registrador

Esta função é utilizada para escrever um valor para um único registrador. Possui a seguinte estrutura (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do registrador (high byte)	Endereço do registrador (byte high)
Endereço do registrador (low byte)	Endereço do registrador (byte low)
Valor do registrador (byte high)	Valor do registrador (byte high)
Valor do registrador (byte low)	Valor do registrador (byte low)
CRC-	CRC-
CRC+	CRC+

Exemplo 2: escrevendo 1 para o idioma (P0201) do SIW600 localizado no endereço 1.

- Endereço: 1 = 01h (1 byte)
- Número do parâmetro: 64 = 0040 (2 bytes)
- Valor do parâmetro: 0000h (2 bytes)

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do escravo	01h	Endereço do escravo	01h
Função	06h	Função	06h
Registrador (high)	00h	Registrador (high)	00h
Registrador (low)	40h	Registrador (low)	40h
Valor (high)	00h	Valor (high)	00h
Valor (low)	00h	Valor (low)	00h
CRC-	88h	CRC-	88h
CRC+	1Eh	CRC+	1Eh

Note que, para esta função, a resposta do escravo é uma cópia idêntica da pergunta feita pelo mestre.

4.4.3 Função 16 – Escrita em Múltiplos Registradores

Esta função permite escrever para um grupo de registradores, que devem estar em seqüência numérica. Também pode ser usada para escrever em um único registrador (os valores são sempre hexadecimal, e cada campo representa um byte):

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
Endereço do registrador inicial (byte high)	Endereço do registrador inicial (byte high)
Endereço do registrador inicial (byte low)	Endereço do registrador inicial (byte low)
Número de registradores (byte high)	Número de registradores (byte high)
Número de registradores (byte low)	Número de registradores (byte low)
Byte Count (Nr. de bytes de dados)	CRC-
Dado 1 (high)	CRC+
Dado 1 (low)	
Dado 2 (high)	
Dado 2 (low)	
etc...	
CRC-	
CRC+	

Exemplo 3: escrita do PF fixo (P0966) igual á 0.957 e modo PF fixo (P0967) igual a 1 no SIW600 localizado no endereço 1.

Valores convertidos para hexadecimal:

- Endereço: 1 = 01h (1 byte)
- Número do primeiro parâmetro: 112 = 0070h (2 bytes)
- Valor do primeiro parâmetro $0.957^2 = \text{FDF4h}$ (registrador 112) 3F74h (registrador 113) (4 bytes)
- Valor do Segundo parâmetro: 1 = 0001h (registrador 114) (2 bytes)

²Tomado 0,957 como um exemplo para número de ponto flutuante. Ele é representado como 0x3F74FDF4 na pequena máquina *endianness* de acordo com a norma IEEE-754. Observe a ordem de bytes: Byte3 = 3Fh, Byte2 = 74H, Byte1 = FDH, Byte0 = F4H.

Solicitação (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do escravo	01h	Endereço do escravo	01h
Função	10h	Função	10h
Registrador inicial (high)	00h	Registrador (high)	00h
Registrador inicial (low)	70h	Registrador (low)	70h
Número de registradores (high)	00h	Valor (high)	00h
Número de registradores (low)	03h	Valor (low)	03h
Byte Count	06h	CRC-	81h
P0966 (Byte1)	FDh	CRC+	D3h
P0966 (Byte0)	F4h		
P0966 (Byte3)	3Fh		
P0966 (Byte2)	74h		
P0967 (high)	00h		
P0967 (low)	01h		
CRC-	CDh		
CRC+	5Dh		

4.4.4 Função 43 – Leitura de Identificação do Produto

Função auxiliar que permite a leitura do fabricante, modelo e versão de firmware do produto. Possui a seguinte estrutura:

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função
MEI Type	MEI Type
Código de leitura	Nível de conformidade
Número do Objeto	Mais palavras
CRC-	Próximo objeto
CRC+	Número de objetos
	Código do primeiro objeto
	Tamanho do primeiro objeto
	Valor do primeiro objeto (n bytes)
	Código do Segundo objeto
	Tamanho do Segundo objeto
	Valor do Segundo objeto (n bytes)
	etc...
	CRC-
	CRC+

Esta função permite a leitura de três categorias de informações: Básica, Regular e Estendida, e cada categoria é formada por um grupo de objetos. Cada objeto é formado por uma sequência de caracteres ASCII. Para o SIW600, apenas informações básicas estão disponíveis, formadas por três objetos:

- Objeto 00h – VendorName: sempre 'WEG';
- Objeto 01h – ProductCode: formado pelo código do produto (SIW600) mais a tensão e a corrente nominal do inversor (ex. 'SIW600 220 - 230 V 10A / 8A');
- Objeto 02h – MajorMinorRevision: indica a versão de firmware do drive, no formato 'VX.XX'.

O código de leitura indica quais as categorias de informações são lidas, e se os objetos são acessados em sequência ou individualmente. No caso, o SIW600 suporta os códigos 01 (informações básicas em sequência), e 04 (acesso individual aos objetos). Os demais campos são especificados pelo protocolo.

Exemplo 4: leitura de informações básicas em sequência, a partir do objeto 01h, de um SIW600 localizado no endereço 1:

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do escravo	01h	Endereço do escravo	01h
Função	2Bh	Função	2Bh
MEI Type	0Eh	MEI Type	0Eh
Código de leitura	01h	Código de leitura	01h
Número do objeto	01h	Nível de conformidade	81h
CRC-	B1h	Mais palavras	00h
CRC+	B7h	Próximo objeto	00h
		Número de objetos	02h
		Código do objeto	01h
		Tamanho do objeto	20h
		Valor do objeto	'SIW600 380 - 440 V 30.4 A / 26.2 A'
		Código do objeto	02h
		Tamanho do objeto	05h
		Valor do objeto	'V1.00'
		CRC-	50h
		CRC+	12h

Neste exemplo, os valores dos objetos não estão representados em forma hexadecimal, mas usam o caractere correspondente em ASCII. Ex.: para o objeto 02h, o valor 'V1.00' foi transmitido como sendo cinco caracteres ASCII, que em formato hexadecimal possuem os valores 56h ('V'), 31h ('1'), 2Eh ('.'), 30h ('0') e 30h ('0').

4.4.5 Erros de Comunicação

Erros de comunicação podem ocorrer tanto na transmissão dos telegramas quanto no conteúdo dos telegramas transmitidos. De acordo com o tipo do erro, o SIW600 poderá ou não, mandar uma resposta para o mestre. Quando o mestre envia uma mensagem para um drive configurado em um determinado endereço da rede, o inversor não irá responder ao mestre caso ocorra:

- Erro no bit de paridade;
- Erro no CRC;
- Timeout* entre os bytes transmitidos (3.5 vezes o tempo de transmissão de um byte).

Nestes casos, o mestre deverá detectar a ocorrência dos erros pelo *timeout* na espera da resposta do escravo. No caso de uma recepção com sucesso, durante o tratamento do telegrama, o inversor pode detectar problemas e enviar uma mensagem de erro, indicando o tipo de problema encontrado:

- Função inválida (código do erro = 1): a função solicitada não está implementada para o equipamento.
- Endereço de dado inválido (código do erro = 2): o endereço do dado (parâmetro) não existe.
- Valor de dado inválido (código do erro = 3): ocorre nas seguintes situações:
 - Valor está fora da faixa permitida.
 - Escrita em dado que não pode ser alterado (registrador somente leitura).



NOTA!

É importante que seja possível identificar no mestre qual o tipo de erro ocorrido para poder diagnosticar problemas durante a comunicação.

No caso de ocorrência de algum destes erros, o escravo deve retornar uma mensagem para o mestre que indica o tipo do erro ocorrido. As mensagens de erro enviadas pelo escravo possuem a seguinte estrutura:

Pergunta (Mestre)	Resposta (Escravo)
Endereço do escravo	Endereço do escravo
Função	Função (com o bit mais significativo em 1)
Dado	Código do erro
CRC-	CRC-
CRC+	CRC+

Exemplo 5: o mestre solicita para o escravo no endereço 1 a escrita no parâmetro localizado no registrador 99 (parâmetro inexistente):

Pergunta (Mestre)		Resposta (Escravo)	
Campo	Valor	Campo	Valor
Endereço do escravo	01h	Endereço do escravo	01h
Função	06h	Função	86h
Registrador (high)	00h	Código do erro	02h
Registrador (low)	63h	CRC-	C3h
Valor (high)	00h	CRC+	A1h
Valor (low)	00h		
CRC-	79h		
CRC+	D4h		

5 Configuração da Potência Reativa

O SIW600 também pode ser configurado para fornecer ou absorver potência reativa. São disponibilizados quatro modos de operação distintos, conforme Tabela 5.1, os quais se referem às características da energia injetada (fator de potência, FP, ou energia reativa) pelo SIW600 no ponto de conexão com a rede elétrica.

Tabela 5.1 - Modos de operação

P0965	Descrição	Resumo	Status indicado
0 (valor de fábrica)	FP Unitário	FP próximo ao unitário	-
1	FP Fixo	FP fixo conforme parâmetro	FPcontrol
2	FP Variável	FP variável conforme curva pré-estabelecida	FPcontrol
3	Controle kVAr	Reativo (kVAr) fixo conforme parâmetro	qcontrol

Os modos de operação 1 a 3 se destinam a uma melhor interconectividade do inversor com o sistema elétrico, conforme normas NBR 16149 e NBR 16150. O modo de operação escolhido pode ser visualizado através do canto superior esquerdo da HMI (status do inversor).

5.1 FP Unitário

Nesse modo de operação, quando P0965=0 (valor de fábrica), o FP do SIW600 é próximo ao unitário em toda sua faixa de operação.

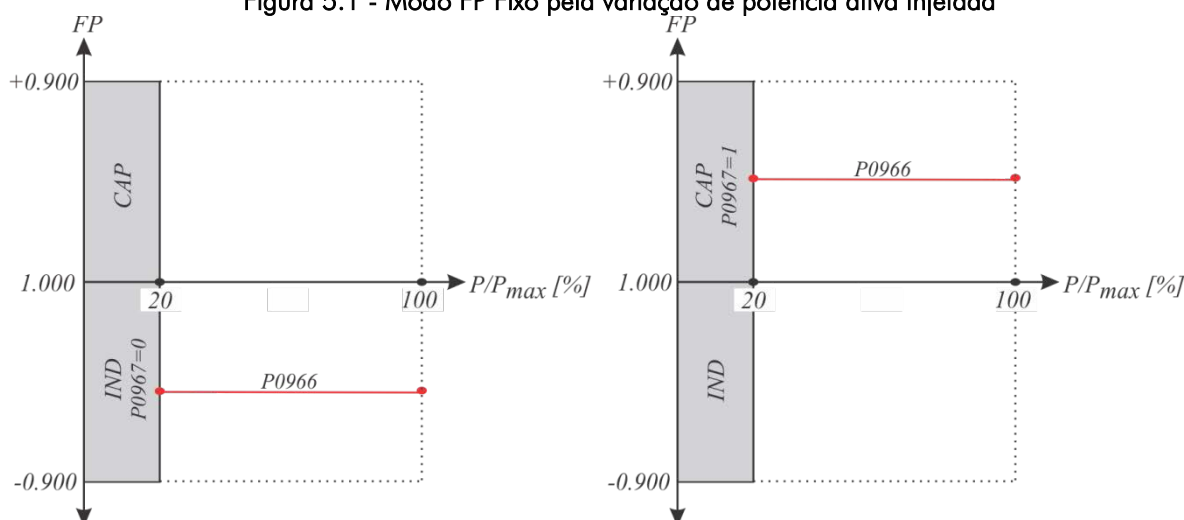
5.2 FP Fixo

Nesse modo de operação, quando P0965=1, o FP do SIW600 é fixo e é configurado em P0966 (0.900 até 0.999) e em P0967 (0=Indutivo e 1=Capacitivo). Uma vez ativado em P0965, esse modo irá operar sempre quando a potência ativa de saída permanecer acima de 20% da potência nominal por, pelo menos, 1 segundo, conforme ilustrado na Figura 5.1. Abaixo de 20% da potência nominal, o FP será próximo ao unitário.

Tabela 5.2 - Modo FP Fixo

Parâmetros	Descrição	Valor e faixa
P0966	FP Fixo	0.999 (0.900 até 0.999)
P0967	Modo FP Fixo	0 ("0" Indutivo / "1" Capacitivo)

Figura 5.1 - Modo FP Fixo pela variação de potência ativa injetada



A potência ativa máxima, P_{max} , é o produto da potência aparente nominal, S_n , do modelo do inversor solar com o fator de potência, FP, definido pela concessionária da rede elétrica, conforme Equação 5.1.

Equação 5.1 - Potência ativa máxima para o controle do FP

$$P_{\max} = S_n \cdot FP$$

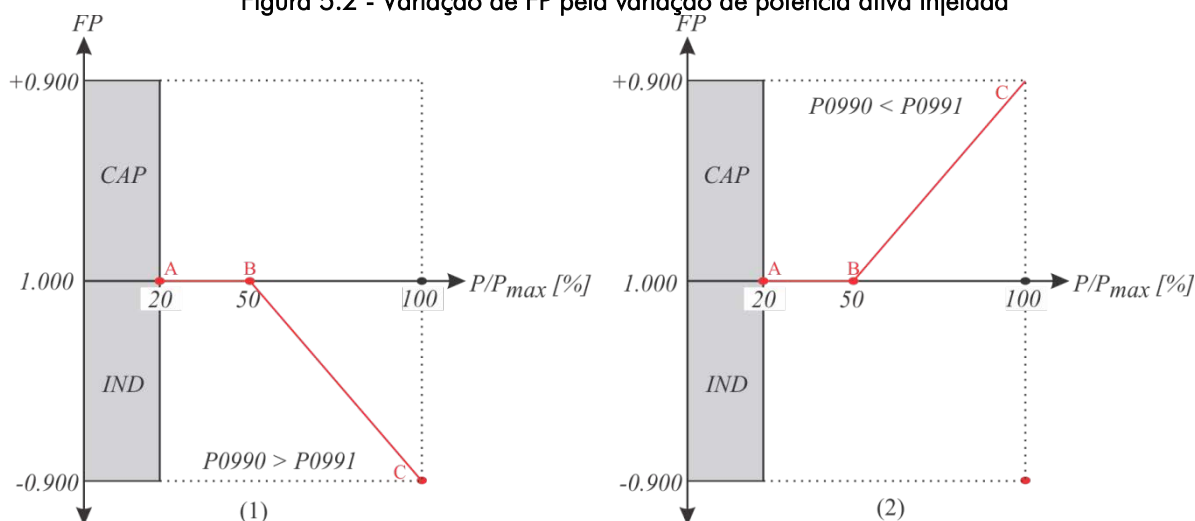
5.3 FP Variável

Nesse modo de operação, quando P0965=2, o SIW600 controla o FP conforme curva configurada nos parâmetros P0990 até P0996. O ajuste desses parâmetros deve ser feito de conforme a definição da concessionária da rede elétrica no ponto de conexão do inversor solar. Os parâmetros para esse modo de operação são descritos na Tabela 5.3.

Tabela 5.3 - Modo FP Variável

Parâmetros	Descrição	Valor e faixa
P0990	Tensão Inic FuncFPxP	104% (90% até 110%)
P0991	Tensão Final FuncFPxP	100% (90% até 110%)
P0992	Potência Ponto A FuncFPxP	20% (5% até 100%)
P0993	FP Ponto A FuncFPxP	0,999 (0,900 até 0,999)
P0994	Potência Ponto B FuncFPxP	50% (5% até 100%)
P0995	FP Ponto B FuncFPxP	0,999 (0,900 até 0,999)
P0996	FP Ponto C FuncFPxP	0,900 (0,900 até 0,999)

Figura 5.2 - Variação de FP pela variação de potência ativa injetada



Conforme Figura 5.2, pode-se verificar duas curvas distintas para injeção de reativos na rede. A curva (1) é recomendada pela norma NBR 16149 e destina-se ao suporte à rede elétrica em instalações onde exista elevação da tensão no ponto de conexão, cujas prováveis causas são a elevada geração de energia e/ou baixa demanda. A curva (2) se destina ao suporte à rede elétrica em instalações onde exista diminuição da tensão no ponto de conexão, cuja provável causa seja a elevada impedância entre o gerador principal e o ponto de conexão.

Tabela 5.4 - Descrição detalhada dos parâmetros do modo FP variável

Parâmetro	Descrição
P0990	Configura a tensão de início em que o algoritmo inicia a operar, dado como valor percentual em relação à tensão nominal de operação. Se $P0990 > P0991$ o algoritmo considera os pontos A,B,C indutivos e irá iniciar a operação se a tensão da rede ficar acima de P0990 por 1seg. Se $P0990 < P0991$ o algoritmo considera os pontos A,B,C capacitivos e irá iniciar a operação se a tensão da rede ficar abaixo de P0990 por 1seg.
P0991	Configura a tensão final em que o algoritmo finaliza a operação, dado como valor percentual em relação à tensão nominal de operação. Se $P0990 > P0991$ o algoritmo irá interromper a operação se a tensão da rede ficar abaixo de P0991 por 1seg. Se $P0990 < P0991$ o algoritmo irá interromper a operação se a tensão da rede ficar acima de P0991 por 1seg.
P0992 P0994	Configura a potência ativa, respectivamente, dos pontos A e B em relação à tensão nominal de operação. A potência ativa do ponto C é sempre fixa em 100%.
P0993 P0995 P0996	Configura o FP, respectivamente, dos pontos A, B e C.



NOTA!

O SIW600 mostra a falha F056 se os pontos A, B e C foram configurados erroneamente. A Falha irá aparecer quando o inversor tentar entrar em operação (estado "Ger Energia"). Esses erros podem ser gerados de duas formas abaixo listadas:

- ➔ Quando a $P0994 > 95\%$ ou $P0992 > P0994 \cdot 5\%$;
- ➔ Quando $P0990 = P0991$.



NOTA!

Se o usuário configurar indevidamente os pontos A, B e C, e tentar habilitar $P0965 = 2$, o software emitirá a falha F056 quando o inversor tentar iniciar a geração de energia (estado "Ger Energia").

5.4 Controle kVAR

Nesse modo de operação, quando $P0965 = 3$, o SIW600 realiza o controle de reativos conforme configurado no parâmetro P0970. Esta funcionalidade tem como objetivo possibilitar que o usuário possa, remotamente ou localmente, controlar a injeção de reativos indutivos ou capacitivos gerados pelo SIW600 no ponto de conexão com o sistema elétrico.

Essa funcionalidade possibilita que o inversor, além de injetar potência ativa na rede, possa também compensar reativos que, possivelmente, são gerados por cargas conectadas nas proximidades do ponto de conexão. A Figura 5.3 ilustra o desempenho deste modo de operação conforme norma NBR 16149.

Figura 5.3 - Modo de operação Controle kVAR conforme NBR 16149

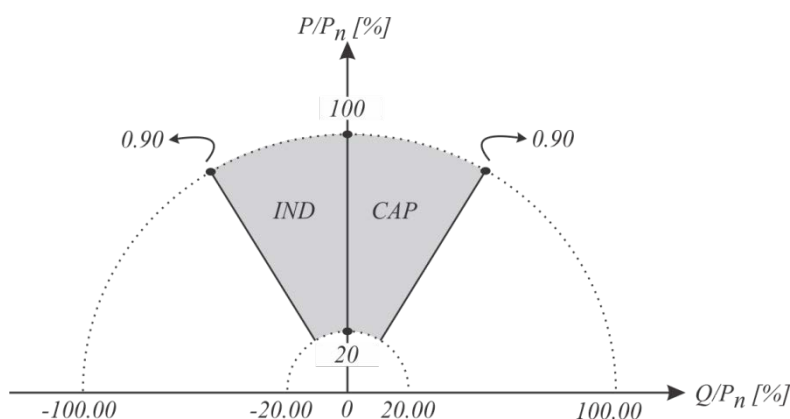


Tabela 5.5 - Modo Controle kVAr

Parâmetros	Descrição	Valor e faixa
P0970	Potência Reativa CA	0.00% (-50.00% até 50.00%)

Tabela 5.6 - Descrição detalhada do parâmetro do modo controle kVAr

Parâmetro	Descrição
P0970	Parâmetro que define a referência para a injeção de potência reativa na rede. Valor negativo para esse parâmetro proporciona injeção de reativos indutivos, de modo que valor positivo proporciona injeção de capacitivos.

APÊNDICES

Apêndice A. Tabela ASCII

Tabela 1 - Caracteres ASCII

Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr	Dec	Hex	Chr
0	00	NUL (Null char.)	32	20	Sp	64	40	@	96	60	`
1	01	SOH (Start of Header)	33	21	!	65	41	A	97	61	a
2	02	STX (Start of Text)	34	22	"	66	42	B	98	62	b
3	03	ETX (End of Text)	35	23	#	67	43	C	99	63	c
4	04	EOT (End of Transmission)	36	24	\$	68	44	D	100	64	d
5	05	ENQ (Enquiry)	37	25	%	69	45	E	101	65	e
6	06	ACK (Acknowledgment)	38	26	&	70	46	F	102	66	f
7	07	BEL (Bell)	39	27	'	71	47	G	103	67	g
8	08	BS (Backspace)	40	28	(72	48	H	104	68	h
9	09	HT (Horizontal Tab)	41	29)	73	49	I	105	69	i
10	0A	LF (Line Feed)	42	2A	*	74	4A	J	106	6A	j
11	0B	VT (Vertical Tab)	43	2B	+	75	4B	K	107	6B	k
12	0C	FF (Form Feed)	44	2C	,	76	4C	L	108	6C	l
13	0D	CR (Carriage Return)	45	2D	-	77	4D	M	109	6D	m
14	0E	SO (Shift Out)	46	2E	.	78	4E	N	110	6E	n
15	0F	SI (Shift In)	47	2F	/	79	4F	O	111	6F	o
16	10	DLE (Data Link Escape)	48	30	0	80	50	P	112	70	p
17	11	DC1 (Device Control 1)	49	31	1	81	51	Q	113	71	q
18	12	DC2 (Device Control 2)	50	32	2	82	52	R	114	72	r
19	13	DC3 (Device Control 3)	51	33	3	83	53	S	115	73	s
20	14	DC4 (Device Control 4)	52	34	4	84	54	T	116	74	t
21	15	NAK (Negative Acknowledgement)	53	35	5	85	55	U	117	75	u
22	16	SYN (Synchronous Idle)	54	36	6	86	56	V	118	76	v
23	17	ETB (End of Trans. Block)	55	37	7	87	57	W	119	77	w
24	18	CAN (Cancel)	56	38	8	88	58	X	120	78	x
25	19	EM (End of Medium)	57	39	9	89	59	Y	121	79	y
26	1A	SUB (Substitute)	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC (Escape)	59	3B	;	91	5B	[123	7B	{
28	1C	FS (File Separator)	60	3C	<	92	5C	\	124	7C	
29	1D	GS (Group Separator)	61	3D	=	93	5D]	125	7D	}
30	1E	RS (Record Separator)	62	3E	>	94	5E	^	126	7E	~
31	1F	US (Unit Separator)	63	3F	?	95	5F	_	127	7F	DEL

Apêndice B. Cálculo do CRC Utilizando Tabelas

A seguir é apresentada uma função, utilizando linguagem de programação “C”, que implementa o cálculo do CRC para o protocolo Modbus-RTU. O cálculo utiliza duas tabelas para fornecer valores pré-calculados dos deslocamentos necessários para a realização do cálculo. O algoritmo foi obtido e é explicado nos documentos referenciados no item 4.

```
/* Table of CRC values for high-order byte */
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41 };

/* Table of CRC values for low-order byte */
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,
0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 };

/* The function returns the CRC as a unsigned short type */
unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg; /* message to calculate CRC upon */
unsigned short usDataLen; /* quantity of bytes in message */
{
    unsigned char uchCRCHi = 0xFF; /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF; /* low byte of CRC initialized */
    unsigned uIndex; /* will index into CRC lookup table */
    while (usDataLen--) /* pass through message buffer */
    {
        uIndex = uchCRCLo ^ *puchMsgg++; /* calculate the CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex];
        uchCRCHi = auchCRCLo[uIndex];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}
```

Apêndice C. Cálculo CRC Utilizando Deslocamento de Registradores

O algoritmo para o cálculo do CRC comunicação Modbus-RTU usando o deslocamento dos registros é descrito neste item. O algoritmo foi obtido a partir e é explicado nos documentos referidos no item 4.

O cálculo do CRC é iniciado pelo carregamento da primeira variável de 16 bits (referenciado a partir de agora como variável CRC) com o valor FFFFh. Depois a rotina segue executando passo a passo:

1. O primeiro byte da mensagem é submetido (apenas os bits de dados – start bit, bit de paridade e stop bit não são usados) a uma XOR (OR Exclusiva) lógica com os 8 bits menos significativos da variável CRC, retornando o resultado da própria variável CRC;
2. Em seguida, a variável CRC é deslocada uma posição para a direita, no sentido do bit menos significativo, e a posição do bit mais significativo é preenchida com 0 (zero);
3. Após este deslocamento, o bit de flag (bit que foi deslocado para fora da variável CRC) é analisado, ocorrendo o seguinte:
 - Se o bit for igual a 0 (zero), nada é feito;
 - Se o bit for igual a 1, o conteúdo da variável CRC é submetido a uma lógica XOR com um valor constante de A001h e o resultado é retornado para a variável CRC.
4. Os passos 2 e 3 são repetidos até que oito mudanças sejam realizadas;
5. Os passos 1 a 4 são repetidos utilizando o próximo bit da mensagem, até toda a mensagem ser processada.

O conteúdo final da variável CRC é o valor do campo de CRC, que é transmitida no fim do telegrama. A parte menos significativa, é transmitida primeiro (CRC-) e em seguida a parte mais significativa (CRC +).