

Protocolo de Controle de Baixo Nível

Versão 0.9
23 de agosto de 2013

Bruno Martins
bruno.martins@lnls.br
Grupo de Controle

Histórico de Revisões

Revisão	Mudanças
0.9 23/08/2013	<ul style="list-style-type: none"> • Alterações e Correções: <ul style="list-style-type: none"> ◦ Comando Criar Grupo de Variáveis retorna o comando OK quando realizado com sucesso. ◦ Limite de 8 grupos, ao invés de 128. ◦ Limite de 64 curvas, ao invés de 128. ◦ Especificação do comando Status. • Remoções: <ul style="list-style-type: none"> ◦ Comando Grupo de Variáveis Criado.
0.8 17/07/2013	<ul style="list-style-type: none"> • Alterações e correções: <ul style="list-style-type: none"> ◦ Erros no comando Recalcular Checksum de Curva. ◦ Interpretação do comando TAMANHO de uma mensagem. ◦ Reorganização do Histórico de Revisões. • Adições: <ul style="list-style-type: none"> ◦ Cabeçalho e rodapé oficiais na página de título. ◦ Comandos Operação Binária em uma Variável e Operação Binária em Grupo de Variáveis. ◦ Tabela contendo todos os comandos de forma resumida. • Remoções: <ul style="list-style-type: none"> ◦ Comandos de Multicast. ◦ Comando Erro Interno.
0.7 06/06/2013	<ul style="list-style-type: none"> • Adições: <ul style="list-style-type: none"> ◦ Comando Recalcular Checksum de Curva. • Remoções: <ul style="list-style-type: none"> ◦ Comandos de Escrita e Leitura sincronizadas. ◦ Checagem de ajustes de valor nos comandos de Escrita
0.6 26/04/2013	<ul style="list-style-type: none"> • Alterações e Correções: <ul style="list-style-type: none"> ◦ Nome do documento: de “Especificação do Protocolo da Rede Serial de Controle” para “Protocolo de Controle de Baixo Nível”. ◦ Divisão da especificação em duas partes: especificação do meio físico e do formato dos pacotes em uma rede serial e especificação dos possíveis conteúdos dos pacotes. ◦ Comportamento do comando Dados de Post-Mortem. ◦ Significado do campo TAMANHO do cabeçalho do Protocolo. • Adições: <ul style="list-style-type: none"> ◦ Comando Valor Inválido. ◦ Comando Ping. ◦ Definição de Curva e dos comandos de consulta e transferência de Curvas. • Remoções: <ul style="list-style-type: none"> ◦ Comandos de Post-Mortem em favor dos comandos de Curva.

<p>0.5 18/04/2013</p>	<ul style="list-style-type: none"> • Alterações e Correções: <ul style="list-style-type: none"> ◦ Nome do documento: de “Especificação do Protocolo PUC” para “Especificação do Protocolo da Rede Serial de Controle”. ◦ Comando Lista de Grupos de Variáveis semelhante ao comando Lista de Variáveis. ◦ Comando Ler um Grupo de Variáveis semelhante ao comando Ler uma Variável. ◦ Opção Postergar Escrita dos comandos Escrever em uma Variável e Escrever em um Grupo de Variáveis ocupa apenas um <i>bit</i>, não mais um <i>byte</i> inteiro. ◦ Comando Grupo de Variáveis para não mais retorna se cada Variável é de escrita ou leitura (informação redundante). ◦ Respostas de comandos que deveriam ser usados apenas em <i>Multicast</i>. ◦ Comando Grupo Criado indica se o grupo criado pode ser escrito (<i>bit</i> mais significativo). • Adições: <ul style="list-style-type: none"> ◦ Quadro de Histórico de Revisões. ◦ Definições de Mestre e Escravo. ◦ Definição de comandos obrigatórios. ◦ Comando Somente Leitura. ◦ Comando Erro Interno. • Remoções: <ul style="list-style-type: none"> ◦ Comandos Variável Somente Leitura e Grupo contém Variáveis de Leitura.
<p>0.4 25/03/2013</p>	<ul style="list-style-type: none"> • Alterações e Correções: <ul style="list-style-type: none"> ◦ Comando de Remover de Grupo de Variável para remover todos os Grupos. ◦ Comando de Desinscrever de Grupo de Multicast para se desinscrever de todos os Grupos. • Adições: <ul style="list-style-type: none"> ◦ Comando Grupo de Variáveis, para ser utilizado como resposta ao comando Criar Grupo de Variáveis.
<p>0.3 22/03/2013</p>	<ul style="list-style-type: none"> • Alterações e Correções: <ul style="list-style-type: none"> ◦ Relocação dos comandos Consultar Grupo de Variáveis e Grupo de Variáveis para a seção Comandos de Consulta. ◦ Explicitação da faixa de valores possíveis de serem utilizados nos campos ID e TAMANHO das Variáveis. ◦ Seção Comandos de Manipulação de Grupos de Variáveis renomeada para Comandos de Manipulação de Grupos. ◦ Comandos de Escrita acomodam Escritas Sincronizada. • Adições: <ul style="list-style-type: none"> ◦ Campo TIPO para Grupos de Variáveis. ◦ Grupos de <i>Multicast</i> e seus comandos associados. ◦ Comandos de Leitura Sincronizada. • Remoções:

	<ul style="list-style-type: none"> ◦ Comandos exclusivos de Escrita Sincronizada.
0.2 18/03/2013	<ul style="list-style-type: none"> • Adições: <ul style="list-style-type: none"> ◦ Comandos de Escrita Sincronizada. ◦ Cinco novos códigos de erro.
0.1 14/03/2013	<ul style="list-style-type: none"> • Versão Inicial.

Sumário

1	Introdução.....	1
2	Comunicação Serial.....	1
2.1	Endereçamento.....	2
2.1.1	Grupos de Multicast.....	2
3	Protocolo de Controle de Baixo Nível.....	3
3.1	Rede, Mensagem, Comando, Mestre e Nó (ou Escravo).....	3
3.2	Variável.....	3
3.3	Grupo de Variáveis.....	4
3.4	Curva.....	4
3.5	Tipo de Protocolo.....	5
3.6	Formato da Mensagem.....	5
3.7	Configuração Hipotética.....	5
4	Comandos do Protocolo de Baixo Nível.....	7
4.1	Comandos de Consulta (0x0_).....	9
4.1.1	Consultar Status (0x00).....	9
4.1.2	Status (0x01).....	9
4.1.3	Consultar Lista de Variáveis (0x02).....	9
4.1.4	Lista de Variáveis (0x03).....	10
4.1.5	Consultar Lista de Grupos de Variáveis (0x04).....	10
4.1.6	Lista de Grupos de Variáveis (0x05).....	11
4.1.7	Consultar Grupo de Variáveis (0x06).....	11
4.1.8	Grupo de Variáveis (0x07).....	12
4.1.9	Consultar Lista de Curvas (0x08).....	12
4.1.10	Lista de Curvas (0x09).....	13
4.2	Comandos de Leitura (0x1_).....	14
4.2.1	Ler Variável (0x10).....	14
4.2.2	Leitura de uma Variável (0x11).....	15
4.2.3	Ler um Grupo de Variáveis (0x12).....	15
4.2.4	Leitura de um Grupo de Variáveis (0x13).....	16
4.3	Comandos de Escrita (0x2_).....	17
4.3.1	Escrever em uma Variável (0x20).....	17
4.3.2	Escrever em Grupo de Variáveis (0x22).....	18
4.3.3	Operação Binária em uma Variável (0x24).....	19
4.3.4	Operação Binária em Grupo de Variáveis (0x26).....	20
4.4	Comandos de Manipulação de Grupos de Variáveis(0x3_).....	22
4.4.1	Criar Grupo de Variáveis (0x30).....	22
4.4.2	Remover Todos os Grupos de Variáveis (0x32).....	23
4.5	Comandos de Transferência de Curvas (0x4_).....	24
4.5.1	Transmitir Bloco de Curva (0x40).....	24
4.5.2	Bloco de Curva (0x41).....	25
4.5.3	Recalcular Checksum de Curva (0x42).....	26
4.6	Comandos de Erro (0xE_).....	27
4.6.1	OK (0xE0).....	27
4.6.2	Mensagem Mal Formada (0xE1).....	27
4.6.3	Operação Não Suportada (0xE2).....	27

4.6.4	ID Inválido (0xE3).....	27
4.6.5	Valor Inválido (0xE4).....	27
4.6.6	Tamanho da Carga Inválido (0xE5).....	27
4.6.7	Somente Leitura (0xE6).....	27
4.6.8	Memória Insuficiente (0xE7).....	27

1 Introdução

A Rede de Controle para o Sirius será construída sobre dois meios de comunicação entre os equipamentos: Ethernet e Serial RS 485. O Grupo de Controle utilizará Ethernet nos níveis mais altos da hierarquia da Rede de Controle; os equipamentos de atuação e medição, no nível mais baixo da hierarquia, serão interconectados através da rede Serial.

Este documento descreve a comunicação entre os dispositivos que compõem o chamado “Baixo Nível” do Sistema de Controle. Este nível compreende todos os dispositivos que controlam diretamente um equipamento ligado ao acelerador: BPM, fonte de corrente, sensor de temperatura, etc. A descrição desta comunicação se divide em duas partes.

A primeira parte dedica-se a especificar a transmissão de um pacote no meio físico de comunicação das redes Seriais RS 485, bem como o formato dos pacotes que circularão neste meio. Esta especificação deve ser seguida por todos os equipamentos que se conectem a uma rede serial com um *Single Board Computer* – SBC – do Grupo CON como mestre. A segunda parte especifica as mensagens do Protocolo de Controle de Baixo Nível, detalhando seus formatos e funções.

Este protocolo foi desenvolvido com a PUC em mente. Entretanto, ele foi projetado de modo a permitir sua utilização por outros *hardwares* de outros grupos do LNLs. A PUC – Placa Universal de Controle – é um *hardware* desenvolvido pelo Grupo de Controle com o propósito de controlar e adquirir dados de equipamentos do acelerador através de entradas e saídas analógicas e digitais, formando a camada mais baixa do Sistema de Controle. Grupos que desejem implementar seu próprio *hardware* de aquisição e controle podem utilizar a PUC como referência de implementação.

2 Comunicação Serial

Na rede Serial RS 485 todos os dados de um pacote devem ser transmitidos em binário, ou seja, nenhum valor de *byte* deve ter significado especial. O fim de pacote deve ser indicado por um silêncio na linha serial de duração equivalente à duração de dois *bytes*. Por exemplo, em uma rede serial 10 Mbps um silêncio de 1.6 μ s após a transmissão de um pacote marca seu fim. Se o Mestre não obter resposta de um Nó em 1 ms o pacote deve ser dado como perdido e reenviado.

Endereçamento		Mensagem								Checksum
DESTINO	ORIGEM									CHECKSUM

Tabela 1- Estrutura de um pacote da Rede Serial

Os pacotes que circulam na rede serial **devem** ter seu formato bem definido. **Devem** existir dois *bytes* para endereçamento, nesta ordem: DESTINO e ORIGEM. O *byte* de ORIGEM **deve** indicar o endereço do transmissor da mensagem. O *byte* DESTINO, entretanto, **pode** indicar tanto um dispositivo ou um grupo de dispositivos aos quais se endereça a mensagem. **Deve** existir, também, um *byte* ao final da mensagem, contendo seu CHECKSUM, como apresentado na Tabela 1. O CHECKSUM **deve** conter a soma de todos os *bytes* anteriores, em complemento de 2, de modo que a soma de todos os *bytes* do pacote seja nula. Cada pacote tem a função de transmitir uma Mensagem, que deve ser interpretada e executada de acordo com o Protocolo de Controle de Baixo Nível.

2.1 Endereçamento

Os dispositivos em uma rede serial devem ser endereçados por um número simples entre 0 (zero) e 31, inclusive. Esta faixa restringe a existência de apenas trinta e dois dispositivos por rede serial. O dispositivo de endereço 0 (zero) é denominado Mestre da rede Serial, sendo os outros dispositivos chamados Nós ou Escravos. Toda rede deve ser composta por **exatamente** um mestre e **no mínimo** um Nó. Os Nós de uma rede Serial devem receber e interpretar pacotes enviados diretamente a ele ou ao um Grupo de *Multicast* do qual ele faça parte. Cada Nó deve responder **apenas** a pacotes enviados diretamente a ele, ou seja, pacotes para Grupos de *Multicast* **não devem** ser respondidos.

2.1.1 Grupos de *Multicast*

Dispositivos podem ser agrupados nos chamados Grupos de *Multicast*. Os Grupos de *Multicast* podem assumir endereços entre 240 e 255, constituindo dezesseis possíveis Grupos. O Grupo de endereço 255 é um Grupo especial denominado *Broadcast*. Todos os dispositivos em uma rede serial **devem** pertencer ao Grupo de *Broadcast*. Um Nó pode pertencer a mais de um Grupo de *Multicast*. Reforça-se que pacotes enviados a Grupos de *Multicast* **não devem** ser respondidos pelos destinatários.

As faixas de endereços em uma Rede Serial estão especificadas na Tabela 2.

Endereço	Usado para
0	Mestre da Rede
1 a 31	Identificação de Nó
32 a 239	RESERVADO
240 a 254	Grupo de <i>Multicast</i>
255	<i>Broadcast</i>

Tabela 2- Endereços na Rede Serial

3 Protocolo de Controle de Baixo Nível

Nesta seção especificam-se termos e conceitos que serão usados no restante deste documento.

3.1 Rede, Mensagem, Comando, Mestre e Nó (ou Escravo)

Os dispositivos conectados pelo Protocolo de Controle de Baixo Nível constituem uma **Rede**. Os componentes da **Rede** se comunicam através da troca de **Mensagens**. Cada **Mensagem** contém um **Comando**, que pode ser tanto um pedido de execução de uma ação quanto uma resposta à tal execução.

Assume-se que os dispositivos da **Rede** atuam em um de dois possíveis papéis: **Mestre** ou **Nó** (também chamado de **Escravo**). Deve existir **exatamente** um **Mestre** por **Rede**. A quantidade de **Nós** por **Rede** não é limitada pelo Protocolo de Controle de Baixo Nível.

3.2 Variável

A Variável é a entidade central do Protocolo. Cada Nó da rede é responsável por gerenciar um número determinado de Variáveis. Cada Variável corresponde a um valor independente que pode ser lido e, em alguns casos, também escrito. Cada Variável deve ter 4 informações associadas a ela, de acordo com a Tabela 3. É importante ressaltar que uma Variável de escrita **deve** poder ser lida. Por exemplo, caso se leia o valor de uma Variável correspondente a um conversor D/A, o valor retornado é o último valor de ajuste recebido. Já a escrita em uma Variável de leitura **não deve** ser permitida. Os ID's das Variáveis devem ser **contínuos** e começar em 0, ou seja, se houver 4 Variáveis em um Nó, por exemplo, seus ID's **devem** ser, **necessariamente**, 0, 1, 2 e 3.

Informação	Descrição
ID	Número de 0 a 127 que identifica univocamente a Variável dentro do Nó
TIPO	Indica se a Variável é de leitura (0) ou de escrita (1)
TAMANHO	Indica quantos <i>bytes</i> são necessários para armazenar o valor da Variável, de 1 a 127
VALOR	Armazena o valor da Variável

Tabela 3- Informações associadas a cada Variável

Estas informações são obtidas pelo Mestre da rede através de comandos específicos de consulta (Comandos de Consulta (0x0)).

3.3 Grupo de Variáveis

É possível que sejam definidos Grupos de Variáveis para que certos conjuntos de Variáveis possam ser lidos e/ou escritos com um único comando. Cada Grupo de Variáveis **deve** conter três informações: um número de identificação (ID), seu tipo (0: contém variáveis de leitura e escrita, 1: contém somente variáveis de escrita) e uma lista com os ID's das Variáveis pertencentes àquele Grupo. **Devem** existir, no mínimo, três Grupos de Variáveis, descritos na Tabela 4, chamados Grupos Padrão. O ID de um Grupo pode assumir valores entre 0 e 7, inclusive. Os Grupos Padrão **não devem** ser modificados ou excluídos. Sua existência com os ID's indicados **deve** ser garantida.

ID	TIPO	Descrição do grupo
0	0	Todas as Variáveis
1	0	Variáveis de Leitura
2	1	Variáveis de Escrita

Tabela 4- Grupos Padrão

Além dos Grupos Padrão, novos Grupos podem ser criados e excluídos através dos Comandos de Manipulação de Grupos de Variáveis(0x3). Os Grupos, assim como as Variáveis, **devem** ter seus ID's contínuos. Ou seja, se houver 5 Grupos em um nó, por exemplo, **necessariamente** seus ID's serão 0, 1, 2, 3, e 4.

3.4 Curva

Curva é o nome dado a uma sequência grande de *bytes*, que podem ou não estar relacionados entre si. Os valores de Curva podem ser transmitidos tanto do Mestre para o Nó quanto do Nó para o Mestre. As Curvas armazenadas em um Nó **devem** ter cinco informações associadas, descritas na Tabela 39.

Informação	Descrição
ID	Número de 0 a 63 que identifica univocamente a Curva dentro do Nó
TIPO	Indica se a Curva é de leitura (0) ou de escrita (1)
TAMANHO	Indica o número de blocos contidos na Curva, menos 1.
VALORES	Armazena os valores da Curva
CHECKSUM	Número de 16 <i>bytes</i> que representa o <i>hash</i> MD5 de todos os VALORES da Curva

Tabela 5- Informações associadas a cada Curva

Os ID's das Curvas **devem** ser contínuos e começar em 0, ou seja, se houver 4 Curvas em um Nó, por exemplo, seus ID's **devem** ser, necessariamente, 0, 1, 2 e 3. O TIPO **deve** indicar se os valores da Curva podem ser escritos (1) (curvas de Ciclagem ou Rampa, por exemplo) ou apenas lidos (0) (curva de Post-Mortem, por exemplo). Cada Curva **deve** ser limitada a um tamanho de 256 blocos, sendo cada bloco de tamanho 16384 bytes, totalizando um máximo de 4MB por Curva. O campo TAMANHO **deve** armazenar o número de blocos da Curva, subtraindo 1. Cada Curva no Nó **pode** (não é obrigatório) possuir um valor de CHECKSUM associado a ela. Este CHECKSUM **deve** ser calculado através da função *hash* MD5. Caso a Curva não possua CHECKSUM, seu valor **deve** ser 0 (zero).

3.5 Tipo de Protocolo

O Protocolo aqui descrito é um protocolo do tipo *token* (ou de ficha). Só tem permissão de transmissão pela Rede aquele que possui a ficha. No caso deste Protocolo a ficha é implícita. Assim, toda comunicação é iniciada pelo Mestre. Uma vez que o Mestre envia uma mensagem diretamente a um dos Escravos, fica implícito que este Escravo possui a ficha até enviar uma resposta, momento em que a ficha retorna ao Mestre. O Protocolo não armazena estado, ou seja, cada par comando/resposta representa uma única transação independente.

3.6 Formato da Mensagem

Uma mensagem do Protocolo deve ter, no mínimo, dois *bytes*, constituintes de seu cabeçalho: COMANDO e TAMANHO. O campo COMANDO especifica qual o comando que deve ser executado pelo Nó. Os códigos de comandos existentes no Protocolo estão descritos na seção Comandos do Protocolo de Baixo Nível. O campo TAMANHO indica quantos *bytes* estão contidos na Carga Útil do pacote. Caso o comando não contenha carga útil, o campo TAMANHO deve conter o valor 0 (zero). A estrutura da mensagem está ilustrada na Tabela 6. O campo TAMANHO pode assumir valores de 0 a 254. O valor 255 é um **caso especial** e deve ser interpretado como o tamanho da carga do comando Bloco de Curva (0x41), ou seja, 16386 *bytes*.

Cabeçalho		Carga Útil							
COMANDO	TAMANHO								

Tabela 6- Estrutura de uma Mensagem do Protocolo de Controle de Baixo Nível

3.7 Configuração Hipotética

Nos exemplos de comandos considera-se a presença de uma PUC – Placa Universal de Controle - na Rede. A PUC foi projetada para ser flexível, admitindo diferentes configurações, no que tange o número de conversores A/D, D/A e entradas e saídas digitais. Assim, considera-se em todos os exemplos da seção Comandos do Protocolo de Baixo Nível uma PUC com 4 conversores A/D, 4 conversores D/A, 1 *byte* de saídas digitais e 1 *byte* de entradas digitais. Assim, a PUC é responsável por 10 Variáveis: uma para cada conversor A/D, uma para cada conversor D/A e uma para cada *byte* de entrada e saída digital.

As PUC's atribuem ID's mais baixos aos conversores A/D, depois aos conversores D/A, seguidos das entradas digitais e, por fim, das saídas digitais. Deste modo, para a PUC hipotética, as 10 Variáveis teriam suas informações como na Tabela 7. Os conversores da PUC têm precisão de 18 *bits*, sendo necessários, no mínimo, 3 *bytes* de espaço para armazenar seu VALOR. Os Grupos de Variáveis presentes na PUC, após a inicialização, estão apresentados na Tabela 8.

Na inicialização a PUC participa de apenas um Grupo serial de *Multicast*: o de *broadcast*.

Variável	ID	TIPO	TAMANHO
Primeiro ADC	0	0	3
Segundo ADC	1	0	3
Terceiro ADC	2	0	3
Quarto ADC	3	0	3
Primeiro DAC	4	1	3
Segundo DAC	5	1	3
Terceiro DAC	6	1	3
Quarto DAC	7	1	3
Entrada Digital	8	0	1
Saída Digital	9	1	1

Tabela 7- Variáveis na PUC hipotética

ID	TIPO	Descrição do grupo	Lista de Variáveis na PUC de exemplo
0	0	Todas as Variáveis	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
1	0	Variáveis de Leitura	0, 1, 2, 3, 8
2	1	Variáveis de Escrita	4, 5, 6, 7, 9

Tabela 8- Grupos de Variáveis padrão na PUC hipotética

4 Comandos do Protocolo de Baixo Nível

Os códigos aceitos no campo COMANDO das mensagens estão descritos nesta seção. Estes comandos estão divididos em classes de comandos, sendo agrupados pela sua semelhança semântica. Cada comando consiste de um *byte*, sendo seu *nibble* mais significativo indicativo de sua classe. Em geral, segue-se a convenção de que comandos pares são enviados pelo Mestre para um Escravo e comandos ímpares são enviados de um Escravo para o Mestre, existindo exceções (por exemplo, nos casos de códigos de erro (seção Comandos de Erro (0xE_)). Se o Nó receber um comando com um formato inesperado, ou seja, com o número de *bytes* indicado no campo TAMANHO diferente do número de *bytes* de fato enviados no corpo da mensagem, **deve-se** retornar o comando Mensagem Mal Formada (0xE1). Existe um conjunto de comandos cuja implementação deve ser obrigatória. Estes comandos estão devidamente identificados através da informação “OBRIGATÓRIO?”, na seção respectiva de cada comando. Caso um comando não obrigatório enviado a um Nó não tenha sido implementado nele, **deve-se** responder com o comando Operação Não Suportada (0xE2). Caso o comando possua Carga Útil, é apresentado um exemplo de sua utilização. Caso o comando não possua Carga Útil, sua mensagem se resume a dois *bytes*: o primeiro *byte* contendo seu código de comando e o segundo *byte* contendo seu tamanho, 0 (zero).

Um resumo de todos os comandos é apresentado na Tabela 9. Descrições detalhadas de cada comando são feitas nas seções subsequentes.

Cód	Nome	Sentido	Carga
0x00	Consultar Status	M → N	
0x01	Status	M ← N	Depende do dispositivo
0x02	Consultar Lista de Variáveis	M → N	
0x03	Lista de Variáveis	M ← N	[Tipo, Tamanho] * (nº de Vars)
0x04	Consultar Lista de Grupos de Variáveis	M → N	
0x05	Lista de Grupos de Variáveis	M ← N	[Tipo, Tamanho] * (nº de Grupos)
0x06	Consultar Grupo de Variáveis	M → N	[ID Grupo]
0x07	Grupo de Variáveis	M ← N	[ID Var] * (nº de Vars no Grupo)
0x08	Consultar Lista de Curvas	M → N	
0x09	Lista de Curvas	M ← N	[Tipo, Tamanho, Checksum] * (nº de Curvas)
0x10	Ler Variável	M → N	[ID Var]
0x11	Leitura de uma Variável	M ← N	[Valor]
0x12	Ler um Grupo de Variáveis	M → N	[ID do Grupo]
0x13	Leitura de um Grupo de Variáveis	M ← N	[Valor] * (nº de Vars no Grupo)
0x20	Escrever em uma Variável	M → N	[ID, Valor da Var]
0x22	Escrever em Grupo de Variáveis	M → N	[ID do Grupo], [Valor]*(nº de Vars no Grupo)
0x24	Operação Binária em uma Variável	M → N	[ID da Var, Operação, Máscara]
0x26	Operação Binária em Grupo de Variáveis	M → N	[ID do Grupo, Operação], [Máscara] * (nº de Vars no Grupo)
0x30	Criar Grupo de Variáveis	M → N	[ID Var] * (nº Vars desejadas)
0x32	Remover Todos os Grupos de Variáveis	M → N	
0x40	Transmitir Bloco de Curva	M → N	[ID Curva, offset do bloco]
0x41	Bloco de Curva	M ↔ N	[ID Curva, offset do bloco, dados]
0x42	Recalcular Checksum de Curva	M → N	[ID Curva]
0xE0	OK	M ← N	
0xE1	Mensagem Mal Formada	M ← N	
0xE2	Operação Não Suportada	M ← N	
0xE3	ID Inválido	M ← N	
0xE4	Valor Inválido	M ← N	
0xE5	Tamanho da Carga Inválido	M ← N	
0xE6	Somente Leitura	M ← N	
0xE7	Memória Insuficiente	M ← N	

Tabela 9- Resumo dos Comandos do Protocolo

4.1 Comandos de Consulta (0x0_)

Os comandos de consulta são utilizados para se obter informações sobre os parâmetros de operação de um Nó.

4.1.1 Consultar *Status* (0x00)

COMANDO: 0x00

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Pedido para que o Nó retorne informações sobre seu *status* de operação

SENTIDO: Mestre → Nó

A resposta a esse comando **deve** ser o comando Status (0x01).

4.1.2 *Status* (0x01)

COMANDO: 0x01

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Informações sobre o *status* de operação do remetente

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: Definida pelo dispositivo, entre 1 e 64 *bytes*

ESTRUTURA DA CARGA: Definida pelo dispositivo, entre 1 e 64 *bytes*

O comando *Status* pode conter até 64 *bytes* de informação, cujos significados devem ser especificados pelo dispositivo.

4.1.3 Consultar Lista de Variáveis (0x02)

COMANDO: 0x02

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Pedido para que o Nó retorne sua lista de Variáveis

SENTIDO: Mestre → Nó

A resposta a esse comando **deve** ser o comando Lista de Variáveis (0x03).

4.1.4 Lista de Variáveis (0x03)

COMANDO: 0x03

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Lista de Variáveis controladas pelo nó

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: (número de Variáveis)

ESTRUTURA DA CARGA: Tabela 10

EXEMPLO: Tabela 11

As Variáveis **devem** ser retornadas na Carga da Mensagem, uma a uma, na sua ordem de ID. **Não deve** existir descontinuidade nos ID's das Variáveis. A primeira Variável é considerada com valor de ID sendo 0. Para cada Variável, **deve-se** retornar um *byte* de informação. O *bit* mais significativo deste *byte* **deve** indicar se a Variável é de leitura (*bit* = 0) ou escrita (*bit* = 1). Os sete *bits* restantes **devem** conter o tamanho do VALOR da Variável. Assim, pode-se notar, no exemplo apresentado na Tabela 11, que os quatro primeiros *bytes* da Carga da Mensagem indicam quatro Variáveis de leitura de tamanho 3. Já os quatro *bytes* seguintes indicam quatro Variáveis de escrita de tamanho 3. Por fim, os dois últimos *bytes* indicam uma Variável de leitura e uma Variável de escrita, ambas de tamanho 1.

Carga		
Tipo (1 <i>bit</i>) Tamanho da primeira Variável (7 <i>bits</i>)	...	Tipo (1 <i>bit</i>) Tamanho da última Variável (7 <i>bits</i>)

Tabela 10- Estrutura do comando Lista de Variáveis

Cabeçalho		Carga									
03	0A	03	03	03	03	83	83	83	83	01	81

Tabela 11- Exemplo do comando Lista de Variáveis

4.1.5 Consultar Lista de Grupos de Variáveis (0x04)

COMANDO: 0x04

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Pedido da lista de Grupos de Variáveis disponíveis no escravo

SENTIDO: Mestre → Nó

A resposta a esse comando **deve** ser o comando Lista de Grupos de Variáveis (0x05).

4.1.6 Lista de Grupos de Variáveis (0x05)

COMANDO: 0x05

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Lista de Grupos de Variáveis no escravo

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: (número de Grupos de Variáveis)

ESTRUTURA DA CARGA: Tabela 12

EXEMPLO: Tabela 13

Os Grupos de Variáveis do nó **devem** ser retornados na Carga da Mensagem, um a um, na sua ordem de ID. **Não deve** existir descontinuidade nos ID's dos Grupos. O primeiro Grupo é considerado com valor de ID sendo 0. Para cada Grupo, **deve-se** retornar um *byte* de informação. O *bit* mais significativo deste *byte* **deve** indicar se o Grupo é de leitura (*bit* = 0) ou escrita (*bit* = 1). Os sete *bits* restantes **devem** conter a quantidade de Variáveis no Grupo. O exemplo lista os três Grupos padrão em um nó. O primeiro grupo é de leitura (*bit* mais significativo 0) e contém dez Variáveis. O segundo também é de leitura e contém cinco Variáveis. Por fim, o último Grupo é de escrita (*bit* mais significativo 1) e contém cinco Variáveis.

Carga		
Tipo (1 <i>bit</i>) Número de Variáveis no primeiro Grupo (7 <i>bits</i>)	...	Tipo (1 <i>bit</i>) Número de Variáveis no último Grupo (7 <i>bits</i>)

Tabela 12- Estrutura do comando Lista de Grupos de Variáveis

Cabeçalho		Carga		
05	03	0A	05	85

Tabela 13- Exemplo de comando Lista de Grupos de Variáveis

4.1.7 Consultar Grupo de Variáveis (0x06)

COMANDO: 0x06

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Consulta para se obter a lista de Variáveis pertencentes a um Grupo

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1

ESTRUTURA DA CARGA: Tabela 14

EXEMPLO: Tabela 15

- **Deve-se** responder a esse comando com o comando Grupo de Variáveis (0x07).
- Caso o ID do Grupo não exista, **deve-se** responder com o comando ID Inválido (0xE3).
- Caso o tamanho da Carga seja diferente de 1, **deve-se** retornar o comando Tamanho da Carga Inválido (0xE5).

O exemplo faz uma consulta ao Grupo de ID 2.

Carga
ID do Grupo

Tabela 14- Estrutura do comando Consultar Grupo de Variáveis

Cabeçalho		Carga
06	01	02

Tabela 15- Exemplo de comando Consultar Grupo de Variáveis

4.1.8 Grupo de Variáveis (0x07)

COMANDO: 0x07

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Lista de Variáveis que compõem um Grupo

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: (quantidade de Variáveis no Grupo consultado)

ESTRUTURA DA CARGA: Tabela 16

EXEMPLO: Tabela 17

As Variáveis contidas no Grupo de Variáveis consultado **devem** ser retornados na Carga da Mensagem, uma a uma, na sua ordem de ID. O exemplo consiste na resposta à consulta ao Grupo de Variáveis de escrita (ID 2). Esta resposta inclui os ID's dos quatro conversores D/A e da saída digital.

Carga		
ID da primeira Variável no Grupo	...	ID da última Variável no Grupo

Tabela 16- Estrutura do comando Grupo de Variáveis

Cabeçalho		Carga					
07	05	04	05	06	07	09	

Tabela 17- Exemplo de comando Grupo de Variáveis

4.1.9 Consultar Lista de Curvas (0x08)

COMANDO: 0x08

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Pedido para que o Nó retorne sua lista de Curvas

SENTIDO: Mestre → Nó

A resposta a esse comando **deve** ser o comando Lista de Curvas (0x09).

4.1.10 Lista de Curvas (0x09)

COMANDO: 0x09

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Lista de Curvas no Nó

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: 18*(número de Curvas)

ESTRUTURA DA CARGA: Tabela 18

EXEMPLO: Tabela 19

As Curvas **devem** ser retornadas na Carga da Mensagem, uma a uma, na sua ordem de ID. **Não deve** existir descontinuidade nos ID's das Curvas. A primeira Curva é considerada com valor de ID sendo 0. Para cada Curvas, **deve-se** retornar 18 bytes de informação. O primeiro byte **deve** indicar se a Curva é de leitura (byte = 0) ou escrita (byte = 1). O segundo byte **deve** conter o número de blocos da Curva, menos 1. Os dezesseis bytes restantes **devem** conter o Checksum da Curva (byte mais significativo primeiro) ou o valor 0 (zero) caso a Curva não possua Checksum.

No exemplo apresentado na Tabela 19 mostra-se a resposta de um Nó que contenha uma Curva de leitura, de tamanho 32 blocos e que não possui Checksum.

Carga										
Tipo (1ª Curva)	Blocos-1 (1ª Curva)	Checksum (mais sig.) (1ª Curva)	...	Checksum (menos sig.) (1ª Curva)	...	Tipo (ult. Curva)	Blocos-1 (ult. Curva)	Checksum (mais sig.) (ult. Curva)	...	Checksum (menos sig.) (primeira Curva)

Tabela 18- Estrutura do comando Lista de Curvas

Cabeçalho		Carga															
09	12	00	31	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Tabela 19- Exemplo do comando Lista de Curvas

4.2 Comandos de Leitura (0x1_)

4.2.1 Ler Variável (0x10)

COMANDO: 0x10

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Leitura do valor de uma Variável

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1

ESTRUTURA DA CARGA: Tabela 20

EXEMPLO: Tabela 21

Este comando faz o pedido de leitura do VALOR de uma Variável. O *bit* mais significativo do campo ID da Variável é reservado e **deve** ser 0 (zero).

- **Deve-se** responder a esse comando com o comando Leitura de uma Variável (0x11).
- Caso não exista uma variável com o ID passado, **deve-se** retornar o comando ID Inválido (0xE3).
- Caso o tamanho da Carga seja diferente de 1, **deve-se** retornar o comando Tamanho da Carga Inválido (0xE5).

O exemplo faz a leitura da Variável de ID 3 (último conversor A/D).

Carga
ID da Variável

Tabela 20- Estrutura do comando Ler Variável

Cabeçalho		Carga
10	01	03

Tabela 21- Exemplo de comando Ler Variável

4.2.2 Leitura de uma Variável (0x11)

COMANDO: 0x11

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Valor de uma Variável

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: (tamanho do campo VALOR da Variável)

ESTRUTURA DA CARGA: Tabela 22

EXEMPLO: Tabela 23

O exemplo consiste na resposta à leitura do VALOR da Variável com ID 3, considerando que este valor seja o valor máximo alcançado pelo conversor A/D (todos os 18 bits altos).

Carga		
Primeiro <i>byte</i> (mais significativo) do VALOR da Variável	...	Último <i>byte</i> (menos significativo) do VALOR da Variável

Tabela 22- Estrutura do comando Leitura de uma Variável

Cabeçalho		Carga		
11	03	03	FF	FF

Tabela 23- Exemplo de comando Leitura de uma Variável

4.2.3 Ler um Grupo de Variáveis (0x12)

COMANDO: 0x12

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Leitura dos valores de um Grupo de Variáveis

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1

ESTRUTURA DA CARGA: Tabela 24

EXEMPLO: Tabela 25

Este comando faz o pedido de leitura do campo VALOR das Variáveis em um Grupo. O *bit* mais significativo do campo ID do Grupo é reservado e **deve** ser 0 (zero).

- **Deve-se** responder a esse comando com o comando Leitura de um Grupo de Variáveis (0x13).
- Caso não exista um Grupo com o ID passado, **deve-se** retornar o comando ID Inválido (0xE3).
- Caso o tamanho da Carga seja diferente de 1, **deve-se** retornar o comando Tamanho da Carga Inválido (0xE5).

O exemplo faz a leitura do Grupo de Variáveis de ID 1 (todas as Variáveis de leitura).

Carga
ID do Grupo

Tabela 24- Estrutura do comando Ler Grupo de Variáveis

Cabeçalho	Carga
12 01	01

Tabela 25- Exemplo de comando Ler Grupo de Variáveis

4.2.4 Leitura de um Grupo de Variáveis (0x13)

COMANDO: 0x13

DESCRIÇÃO: Contém os valores das Variáveis de um Grupo de Variáveis

OBRIGATÓRIO?: Sim

SENTIDO: Nó → Mestre

TAMANHO DA CARGA: (soma dos tamanhos dos campos VALOR das Variáveis)

ESTRUTURA DA CARGA: Tabela 26

EXEMPLO: Tabela 27

Deve-se retornar o VALOR de cada Variável na ordem crescente de ID das Variáveis. O exemplo consiste na resposta à leitura dos valores de todas as Variáveis de leitura (Grupo de ID 1), considerando que os conversores A/D estejam com seu valor máximo (todos os 18 bits altos) e que a entrada digital tenha o valor 0xAA.

Carga						
Primeiro byte da primeira Variável	...	Último byte da primeira Variável	...	Primeiro byte da última Variável	...	Último byte da última Variável

Tabela 26- Estrutura do comando Leitura de um Grupo de Variáveis

Cabeçalho	Carga												
13 0C	03	FF	FF	03	FF	FF	03	FF	FF	03	FF	FF	AA

Tabela 27- Exemplo de comando Leitura de um Grupo de Variáveis

4.3 Comandos de Escrita (0x2_)

4.3.1 Escrever em uma Variável (0x20)

COMANDO: 0x20

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Escrita no valor de uma Variável

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1 + (tamanho do VALOR da Variável)

ESTRUTURA DA CARGA: Tabela 28

EXEMPLO: Tabela 29

O comando de escrita de uma Variável **deve** ser enviado **apenas** para Variáveis de escrita. O *bit* mais significativo do campo ID da Variável é reservado e **deve** ser 0 (zero). Caso algum erro aconteça, a escrita **deve** ser ignorada pelo Nó. O comando a ser retornado depende de alguns fatores:

- Caso o comando tenha sido executado com sucesso, **deve-se** retornar o comando OK (0xE0);
- Caso a ID da Variável não exista, **deve-se** retornar o comando ID Inválido (0xE3);
- Caso o tamanho da Carga seja diferente do tamanho esperado, **deve-se** responder com o comando Tamanho da Carga Inválido (0xE5);
- Caso o ID de Variável passado seja de uma Variável de leitura, **deve-se** responder com o comando Somente Leitura (0xE6).

O exemplo ilustra o ajuste imediato do valor do primeiro conversor D/A (ID 4) para 0x1BBBB.

Carga			
ID da Variável	Primeiro byte (mais significativo) do VALOR da Variável	...	Último byte (menos significativo) do VALOR da Variável

Tabela 28- Estrutura do comando Escrever em uma Variável

Cabeçalho		Carga			
20	04	04	01	BB	BB

Tabela 29- Exemplo de comando Escrever em uma Variável

4.3.2 Escrever em Grupo de Variáveis (0x22)

COMANDO: 0x22

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Escreve nos valores das Variáveis de um Grupo de Variáveis

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1 + (soma dos tamanhos dos campos VALOR das Variáveis)

ESTRUTURA DA CARGA: Tabela 30

EXEMPLO: Tabela 31

O comando de escrita em um Grupo de Variáveis deve ser enviado para Grupos contendo apenas Variáveis de escrita. O *bit* mais significativo do campo ID do Grupo é reservado e **deve** ser 0 (zero). Caso se tente escrever em um Grupo com ao menos uma Variável de leitura, o Nó **deve** ignorar todas as escritas e retornar o código de erro apropriado.

- Caso o comando tenha sido executado com sucesso, **deve-se** retornar o comando OK (0xE0);
- Caso a ID do Grupo de Variável não exista, **deve-se** retornar o comando ID Inválido (0xE3);
- Caso o tamanho da Carga seja diferente do tamanho esperado, **deve-se** responder com o comando Tamanho da Carga Inválido (0xE5);
- Caso o ID do Grupo de Variáveis se refira a um Grupo com ao menos uma Variável de leitura, **deve-se** responder com o comando Somente Leitura (0xE6);

O exemplo ilustra o ajuste imediato de todos os conversores D/A para 0x1BBBB e da saída digital para 0xCC, usando-se o Grupo de ID 2.

Carga							
ID do Grupo	Primeiro <i>byte</i> da primeira Variável	...	Último <i>byte</i> da primeira Variável	...	Primeiro <i>byte</i> da última Variável	...	Último <i>byte</i> da última Variável

Tabela 30- Estrutura do comando Escrever em Grupo de Variáveis

Cabeçalho		Carga													
22	0E	02	01	BB	BB	01	BB	BB	01	BB	BB	01	BB	BB	CC

Tabela 31- Exemplo de comando Escrever em Grupo de Variáveis

4.3.3 Operação Binária em uma Variável (0x24)

COMANDO: 0x24

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Realiza uma Operação Binária no valor de uma Variável

SENTIDO: Mestre → Nô

TAMANHO DA CARGA: 2 + (tamanho do VALOR da Variável)

ESTRUTURA DA CARGA: Tabela 33

EXEMPLO: Tabela 34

O comando de Operação Binária em uma Variável **deve** ser enviado **apenas** para Variáveis de escrita. O *bit* mais significativo do campo ID da Variável é reservado e **deve** ser 0 (zero). Caso algum erro aconteça, a Operação **deve** ser ignorada pelo Nô. O código da operação a ser feita **deve** estar entre os códigos da Tabela 32. O comando a ser retornado depende de alguns fatores:

- Caso o comando tenha sido executado com sucesso, **deve-se** retornar o comando OK (0xE0);
- Caso a operação requisitada não esteja entre as possíveis Operações Binárias, **deve-se** retornar o comando Operação Não Suportada (0xE2).
- Caso a ID da Variável não exista, **deve-se** retornar o comando ID Inválido (0xE3);
- Caso o tamanho da Carga seja diferente do tamanho esperado, **deve-se** responder com o comando Tamanho da Carga Inválido (0xE5);
- Caso o ID de Variável passado seja de uma Variável de leitura, **deve-se** responder com o comando Somente Leitura (0xE6).

Código	Operação	Descrição
0x53 ('S')	SET bits	'Liga' (torna 1) os <i>bits</i> especificados pela máscara.
0x43 ('C')	CLEAR bits	'Desliga' (torna 0) os <i>bits</i> especificados pela máscara.
0x54 ('T')	TOGGLE bits	Inverte os <i>bits</i> especificados pela máscara.
0x41 ('A')	AND	Realiza um AND lógico entre o valor da Variável e a máscara.
0x4F ('O')	OR	Realiza um OR lógico entre o valor da Variável e a máscara.
0x58 ('X')	XOR	Realiza um XOR lógico entre o valor da Variável e a máscara.

Tabela 32- Códigos das possíveis Operações Binárias

O exemplo ilustra uma operação OR que liga 4 *bits* mais significativos da saída digital.

Carga				
ID da Variável	Código da Operação	Primeiro byte (mais significativo) da máscara	...	Último byte (menos significativo) da máscara

Tabela 33- Estrutura do comando Operação Binária em uma Variável

Cabeçalho		Carga		
24	03	09	4F	F0

Tabela 34- Exemplo de comando Operação Binária em uma Variável

4.3.4 Operação Binária em Grupo de Variáveis (0x26)

COMANDO: 0x26

OBRIGATÓRIO?: Sim

DESCRIÇÃO: Realiza uma Operação Binária em um Grupo de Variáveis

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 2 + (soma dos tamanhos dos campos VALOR das Variáveis)

ESTRUTURA DA CARGA: Tabela 35

EXEMPLO: Tabela 36

O comando de Operação Binária em um Grupo de Variáveis **deve** ser enviado **apenas** para Grupos de escrita. O *bit* mais significativo do campo ID do Grupo é reservado e **deve** ser 0 (zero). Caso algum erro aconteça, a Operação **deve** ser ignorada pelo Nó. O código da operação a ser feita **deve** estar entre os códigos da Tabela 32, apresentada no comando Operação Binária em uma Variável (0x24). Caso se tente realizar uma Operação Binária em um Grupo com ao menos uma Variável de leitura, o Nó **deve** ignorar todas as Operações Binárias e retornar o código de erro apropriado. O comando a ser retornado depende de alguns fatores:

- Caso o comando tenha sido executado com sucesso, **deve-se** retornar o comando OK (0xE0);
- Caso a operação requisitada não esteja entre as possíveis Operações Binárias, **deve-se** retornar o comando Operação Não Suportada (0xE2).
- Caso a ID do Grupo de Variável não exista, **deve-se** retornar o comando ID Inválido (0xE3);
- Caso o tamanho da Carga seja diferente do tamanho esperado, **deve-se** responder com o comando Tamanho da Carga Inválido (0xE5);
- Caso o ID do Grupo de Variáveis se refira a um Grupo com ao menos uma Variável de leitura, **deve-se** responder com o comando Somente Leitura (0xE6);

O exemplo ilustra uma Operação Binária OR que liga os 4 *bits* menos significativos de todas as Variáveis, usando-se o Grupo de ID 2.

Carga							
ID do Grupo	Código da operação	Primeiro byte da primeira máscara	...	Último byte da primeira máscara	...	Primeiro byte da última máscara	Último byte da última máscara

Tabela 35- Estrutura do comando Operação Binária em Grupo de Variáveis

Cabeçalho		Carga													
26	0F	02	4F	00	00	0F	00	00	0F	00	00	0F	00	00	0F

Tabela 36- Exemplo de comando Operação Binária em Grupo de Variáveis

4.4 Comandos de Manipulação de Grupos de Variáveis(0x3_)

4.4.1 Criar Grupo de Variáveis (0x30)

COMANDO: 0x30

OBRIGATÓRIO?: Não

DESCRIÇÃO: Cria um novo Grupo de Variáveis

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: (número de Variáveis no Grupo)

ESTRUTURA DA CARGA: Tabela 37

EXEMPLO: Tabela 38

Este comando cria um novo Grupo de Variáveis, para ser adicionado aos Grupos já existentes. **Deve-se** usar sete *bits* para o ID do Grupo, permitindo a existência de 128 Grupos de Variáveis. Porém, três desses Grupos são pré-definidos, como descrito na seção Grupo de Variáveis. **Deve-se**, neste comando, especificar uma lista com os ID's das Variáveis que pertencerão a este Grupo. O ID do Grupo é atribuído automaticamente, sendo **necessariamente** igual ao valor do ID do último Grupo criado, somado a 1.

- Caso a criação aconteça com sucesso, **deve-se** retornar o comando OK (0xE0);
- Caso o número de Variáveis seja nulo ou maior que a quantidade de Variáveis existentes, **deve-se** retornar o comando Tamanho da Carga Inválido (0xE5);
- Caso não haja memória suficiente para a criação do Grupo, **deve-se** retornar o comando Memória Insuficiente (0xE7);

O exemplo ilustra a criação de um Grupo contendo todos os conversores D/A.

Carga		
ID da primeira Variável do Grupo	...	ID da última variável do Grupo

Tabela 37- Estrutura do comando Criar Grupo de Variáveis

Cabeçalho		Carga			
30	04	04	05	06	07

Tabela 38- Exemplo de comando Criar Grupo de Variáveis

4.4.2 Remover Todos os Grupos de Variáveis (0x32)

COMANDO: 0x32

OBRIGATÓRIO?: Não

DESCRIÇÃO: Pedido de remoção de todos os Grupo de Variáveis

SENTIDO: Mestre → Nó

Este comando **deve** remover todos os Grupos de Variáveis criados, **exceto** os grupos padrão, descritos na seção Grupo de Variáveis. **Deve-se** retornar o comando OK (0xE0).

4.5 Comandos de Transferência de Curvas (0x4_)

Nesta seção estão descritos os comandos para transferência de Curva.

4.5.1 Transmitir Bloco de Curva (0x40)

COMANDO: 0x40

OBRIGATÓRIO?: Não

DESCRIÇÃO: Pedido para que um Nó transmita um bloco de Curva

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 2

ESTRUTURA DA CARGA: Tabela 39

EXEMPLO: Tabela 40

Este comando pede ao Nó para que ele transmita um determinado bloco de uma determinada Curva.

- Caso os dados do comando sejam válidos, **deve-se** retornar o comando Bloco de Curva (0x41);
- Caso o ID da Curva seja inválido, **deve-se** retornar o comando ID Inválido (0xE3);
- Caso o *offset* do bloco seja inválido (maior que o número de blocos da Curva), **deve-se** retornar o comando Valor Inválido (0xE4);

O exemplo ilustra o pedido do quinto bloco da Curva de ID 3.

Carga	
ID da Curva	Offset do bloco

Tabela 39- Estrutura do comando Transmitir Bloco de Curva

Cabeçalho		Carga	
40	02	03	04

Tabela 40- Exemplo de comando Transmitir Bloco de Curva

4.5.2 Bloco de Curva (0x41)

COMANDO: 0x41

OBRIGATÓRIO?: Não

DESCRIÇÃO: Transmissão de um bloco de uma Curva

SENTIDO: Mestre → Nó ou Nó → Mestre

TAMANHO DA CARGA: 16386

ESTRUTURA DA CARGA: Tabela 41

EXEMPLO: Tabela 42

Transmissão de um bloco de Curva tanto pelo Nó quanto pelo Mestre. Se a transmissão for feita do Mestre para o Nó, entende-se como uma escrita no bloco indicado. Neste caso, o Checksum da Curva especificada é **zerado**. O cálculo do Checksum deve ser realizado após o término do envio de todos os blocos da Curva, através do comando Recalcular Checksum de Curva (0x42).

- Se o comando foi enviado do Mestre para o Nó:
 - Caso os dados do comando sejam válidos, **deve-se** retornar o comando OK (0xE0);
 - Caso o ID da Curva seja inválido, **deve-se** retornar o comando ID Inválido (0xE3);
 - Caso o *offset* do bloco seja inválido (maior que o número de blocos da Curva), **deve-se** retornar o comando Valor Inválido (0xE4);

O exemplo ilustra a transmissão do quinto bloco da Curva de ID 3 cujos *bytes* têm todos o valor 0xDD.

Carga				
ID da Curva	Offset do bloco	Primeiro byte do bloco	...	Último byte do bloco

Tabela 41- Estrutura do comando Bloco de Curva

Cabeçalho		Carga				
41	FF	03	04	DD	...	DD

Tabela 42- Exemplo de comando Bloco de Curva

4.5.3 Recalcular Checksum de Curva (0x42)

COMANDO: 0x42

OBRIGATÓRIO?: Não

DESCRIÇÃO: Pedido para que o Nó recalcule o Checksum da curva de escrita especificada.

SENTIDO: Mestre → Nó

TAMANHO DA CARGA: 1

ESTRUTURA DA CARGA: Tabela 43

EXEMPLO: Tabela 9

Este comando faz com que o Checksum da Curva especificada seja recalculado pelo Nó. Deve-se enviar este comando ao Nó toda vez que uma Curva de escrita for alterada.

- Caso os dados do comando sejam válidos, **deve-se** retornar o comando OK (0xE0);
- Caso o ID da Curva seja inválido, **deve-se** retornar o comando ID Inválido (0xE3);

O exemplo ilustra o pedido de recálculo do Checksum da curva de ID 0.

Carga
ID da Curva

Tabela 43- Estrutura do comando Recalcular Checksum de Curva

Cabeçalho	Carga
42	01 00

Tabela 44- Exemplo de comando Recalcular Checksum de Curva

4.6 Comandos de Erro (0xE_)

Todos os Comandos de Erro obedecem o sentido Nó → Mestre e não possuem carga útil.

4.6.1 OK (0xE0)

Indica que o último comando enviado foi executado com sucesso.

4.6.2 Mensagem Mal Formada (0xE1)

Indica que o número de *bytes* recebidos difere do tamanho da mensagem especificado no campo TAMANHO da mensagem.

4.6.3 Operação Não Suportada (0xE2)

Indica que o comando requisitado não foi implementado.

4.6.4 ID Inválido (0xE3)

Indica que foi passado um ID inválido no comando anterior.

4.6.5 Valor Inválido (0xE4)

Indica que um valor passado no comando anterior não está dentro da faixa de valores aceitáveis.

4.6.6 Tamanho da Carga Inválido (0xE5)

A Carga da última mensagem recebida tinha um tamanho diferente do esperado pelo comando.

4.6.7 Somente Leitura (0xE6)

Foi tentada uma escrita em uma entidade de leitura.

4.6.8 Memória Insuficiente (0xE7)

O comando anterior falhou por falta de memória disponível.