



# **Comandos de Pesagem para Protocolo ALFA Instrumentos**

# Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

## 1. Introdução

Este documento descreve o Protocolo de Comunicação Serial Assíncrona de Dados e deve ser utilizado para propiciar a comunicação de dados entre os equipamentos da Alfa Instrumentos e dispositivos que necessitem receber informações de pesagem.

O tipo de interligação para comunicação serial mais simples é o ponto à ponto, e o mais elaborado é o multiponto, tendo os dois suas vantagens e desvantagens. O Protocolo de Comunicação Serial da Alfa Instrumentos foi desenvolvido para poder trabalhar com ambos. A seguir, trataremos do tipo de interligação multiponto, bem como de seu protocolo.

## 2. Conceitos de Mestre /Escravo

Dois conceitos importantes no protocolo são os relativos ao equipamento mestre e ao equipamento escravo. Em qualquer interligação na qual esteja sendo utilizado um protocolo, necessariamente deverá existir um e somente um equipamento mestre e pelo menos um equipamento escravo.

Defini-se como mestre o equipamento responsável por toda a iniciativa do processo de comunicação, ou seja, uma troca de dados entre os equipamentos em qualquer dos dois sentidos, é iniciado e finalizado pelo mestre. Defini-se como escravo o equipamento que, no processo de comunicação, só realiza alguma função sob requisição e controle do mestre.

Um equipamento pode ser mestre ou escravo dependendo somente de sua aplicação no processo. Isto quer dizer que para um determinado processo, um equipamento pode estar configurado como mestre, sendo que, em outro processo o mesmo equipamento pode estar configurado como escravo.

Na comunicação entre um mestre e um escravo existem duas situações possíveis:

- a) mestre deseja enviar/receber dados para/do escravo
- b) escravo deseja enviar/receber dados para/do mestre

No primeiro caso, como o mestre tem o poder de iniciar o processo de comunicação, ele envia uma mensagem a qualquer momento, de acordo com suas necessidades e independente do estado do escravo, requisitando a este a realização de uma determinada função. Ao receber a mensagem, o escravo executa a função requerida e envia uma resposta contendo o resultado da função requisitada. Este processo de comunicação é chamado de **SELECT**.

No segundo caso, como o escravo não pode tomar a iniciativa de começar o processo de comunicação, ele aguarda que o mestre pergunte se o escravo deseja enviar/receber alguma mensagem, e só quando isto ocorrer o escravo envia sua mensagem requisitando ao mestre a realização de determinada função. Ao receber a mensagem enviada pelo escravo, o mestre realiza a função solicitada e envia a resposta contendo o resultado da função. Esse processo recebe o nome de **POLLING**.

## 3. Protocolo em termos de enlace

O estágio de enlace se preocupa exclusivamente com os procedimentos a serem seguidos na transmissão e recepção das mensagens, recuperação de erros, sincronismo entre as mensagens, sincronismo entre os equipamentos em relação à comunicação, conexão e desconexão, etc., deixando para o nível de aplicação o tratamento das informações relativas ao processo que se deseja supervisionar e/ou controlar.

O controle da comunicação é disciplinado por códigos ASCII em sequências especiais.

Os procedimentos referentes ao sincronismo entre as mensagens e equipamentos são implementados pelos processos de "Select" e "Polling".

### 3.1 Formato das mensagens

Todas as mensagens enviadas no protocolo são classificadas em **mensagens de comando** e **mensagens de resposta**. As mensagens de comando são aquelas que contêm um comando para a execução de uma atividade por parte de quem a está recebendo (o receptor). Um comando pode ser uma requisição de dados, uma ordem para uma mudança de estado, uma carga de programa no receptor, uma requisição de status do equipamento, a alteração de um parâmetro do receptor, etc..

As mensagens de resposta são aquelas que contêm a resposta de um comando recebido. Uma resposta pode conter dados requisitados, status do equipamento, confirmação da carga de programa, confirmação da alteração do parâmetro requisitado, etc..

O formato das mensagens, quer seja de comando ou resposta, é o seguinte:

DLE	STX	DST	SRC	blocos de informações	DLE	ETX	BCC
-----	-----	-----	-----	-----------------------	-----	-----	-----

- DLE (Data Link Escape): código ASCII = 10H - Este caracter tem a função de informar ao equipamento que está recebendo a mensagem, que o próximo byte da mensagem é um caracter ASCII de controle da comunicação (STX, ETX, etc.). Isto permite que não exista confusão entre os caracteres de controle e os outros bytes constituintes da mensagem. Se o equipamento receptor receber um DLE após outro DLE, deve interpretar isso como apenas um DLE, isto quer dizer que caso se queira enviar um DLE como dado da mensagem, deve ser enviado DLE duas vezes.
- STX (STart of Text): código ASCII = 02H.
- DLE STX: sequência que indica o início de um bloco de mensagem.
- DST (DeSTino). Identifica o endereço do equipamento da rede de comunicação que deve receber a mensagem.
- SRC (SouRCe). Identifica o endereço do equipamento da rede de comunicação que esta enviando a mensagem.
- Blocos de Informações: contém os dados binários referentes ao processo e que são tratados pelo protocolo na aplicação. No caso desses dados se referirem a comando, a mensagem completa, incluindo os caracteres de controle (DLE, STX, ETX, BCC), forma um "Quadro de Comando". Já no caso dos dados serem referentes a uma resposta, a mensagem completa forma um "Quadro de Resposta".
- ETX (End of TeXt): código ASCII = 03H.
- DLE ETX: sequência que indica se a mensagem de comando ou resposta terminou, não existindo nenhum outro bloco a ser enviado.

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

- BCC (Block Check Character) - Permite a verificação da consistência dos dados recebidos, isto é, se ocorreu algum erro durante a transmissão ou recepção do bloco da mensagem. O BCC é calculado a partir de STX até ETX. É calculado através da lógica OU EXCLUSIVO do byte de dados atual com o BCC anterior, seguido de um Rotate Left de uma posição. Defini-se 00H como valor inicial do BCC.

### 3.2 Procedimentos de transmissão e recepção de mensagens:

A transmissão e recepção de uma mensagem, do ponto de vista de seus procedimentos de detecção e recuperação de erros, segue sempre uma mesma sequência e independe do sentido da transmissão (mestre para escravo ou vice-versa), e do tipo da mensagem (Comando ou Resposta). No entanto os procedimentos, relativos ao sincronismo da comunicação entre os equipamentos, dependem da definição mestre e escravo para uma rede multiponto e/ou da definição "Source" e "Sink" para comunicação ponto-a-ponto.

Este item descreve os procedimentos a serem seguidos na transmissão e recepção do bloco que compõem uma mensagem segundo o conceito mestre e escravo.

#### 3.2.1 Escravo:

O escravo pode se comportar de duas formas distintas:

- recepção de mensagens enviadas pelo mestre
- transmissão de mensagens para o mestre, quando solicitado

##### 3.2.1.1 Recepção:

Na recepção de mensagens, o escravo analisa os dados recebidos e pode considerar o bloco recebido como:

Inálido - quando o endereço destino (DST) não corresponde ao endereço do escravo ou não reconhece o delimitador de início ou fim do bloco (DLE STX, DLE ETX). Neste caso o escravo não responde ao mestre.

Correto - Neste caso o escravo transmite ao mestre a sequência de reconhecimento do bloco ( ACK - positive Acknowledgement = 06H). Ao responder ACK ao mestre, está finalizado o processo de recepção para o escravo.

Errado - Neste caso o escravo transmite ao mestre a sequência de erro de recepção (NAK - negative Acknowledgement = 15H ).

Ocupado para atender - Neste caso o escravo transmite ao mestre a sequência de impossibilidade de atender ( WAK - Wait Acknowledgement = 14H DC4). Ao responder WAK, o mestre finaliza o processo de transmissão para o escravo, e a retransmissão da mensagem é decidida pela aplicação.

##### 3.2.1.2 Transmissão:

Na transmissão de mensagens para o mestre, o escravo analisa o pedido recebido como:

Nenhuma mensagem ou mensagem sendo montada - Neste caso o escravo transmite ao mestre a sequência de impossibilidade de enviar uma mensagem, pois não há nenhuma mensagem a ser enviada (EOT- End of Transmission = 04H). Ao responder EOT, o mestre poderá assumir uma entre duas possibilidades conforme decisão da aplicação, executando uma sequência de pedidos de transmissão de dados do escravo até um limite definido pela aplicação, ou finaliza o processo de recepção e a decisão de um novo pedido de transmissão de dados do escravo fica a critério da aplicação.

Ocupado para atender - Neste caso o escravo transmite ao mestre a sequência de impossibilidade de atender ( WAK - Wait Acknowledgement = 14H ). Ao responder WAK, o mestre finaliza o processo de recepção do escravo ou executa uma sequência de pedidos de transmissão de dados do escravo até um limite definido pela aplicação.

Transmite a mensagem - Transmite a mensagem conforme o formato padrão.

#### 3.2.2 Mestre:

Do mesmo modo que o escravo, o mestre pode ter dois comportamentos distintos:

- transmissor de mensagens para o escravo
- receptor de mensagens enviadas pelo escravo

##### 3.2.2.1 Transmissor:

Como transmissor, o mestre envia a mensagem e dispara uma contagem máxima (timeout) de espera de confirmação de recepção pelo escravo. Podem ocorrer as seguintes situações com relação à resposta esperada:

Ocorreu Timeout - Caso ocorra o timeout de espera de confirmação de resposta, o mestre executa sequência de "retries", ou seja, retransmite a mensagem enviada anteriormente. Esse procedimento de retry se repete até o escravo responder ACK ou NAK, ou até ocorrer o número máximo de "retries" permitidos. Caso ocorra o estouro do número de "retries", o mestre assume a ocorrência de um erro de comunicação.

Recebeu NAK - Caso receba uma resposta negativa, a mensagem enviada o mestre executa a sequência de "retries" idêntica ao caso de ocorrência de timeout.

Recebeu ACK - Ao receber ACK da mensagem enviada, o mestre finaliza o processo de transmissão desta mensagem.

Não compreendeu a resposta ACK ou a resposta NAK - O transmissor executa sequência de "retries", ou seja, retransmite a mensagem enviada anteriormente. Esse procedimento de "retry" se repete até o escravo responder ACK ou NAK, ou até o número máximo de "retries" ser atingido.

Recebeu WAK - Ao receber o WAK o mestre finaliza o processo de transmissão e a retransmissão fica a cargo da aplicação.

##### 3.2.2.2 Receptor

Como receptor o mestre questiona o escravo se possui alguma mensagem a ser enviada, podendo ocorrer as seguintes situações:

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

Recebeu WAK - Ao receber WAK, pode se comportar de dois modos independentes, conforme decisão da aplicação: executa uma sequência de “retries” do pedido de transmissão ou finaliza a recepção, e um novo pedido é feito pela aplicação.

Recebeu EOT - Ao receber EOT o mestre finaliza o processo de recepção e um novo pedido fica a cargo da aplicação.

Recebe a mensagem do escravo - Nesta situação, após receber a mensagem o mestre envia os seguintes caracteres de controle: ACK, confirmando para o escravo que a recepção da mensagem enviada foi bem sucedida, ou NAK, informando para o escravo que houve erro na recepção da mensagem e solicitando uma retransmissão.

Ocorreu Timeout - Existem duas formas de timeout possíveis: durante o pedido de transmissão de dados ao escravo não ocorrendo nenhuma resposta ou resposta não compreendida, ou caso a mensagem recebida tenha sido ignorada. Nesta situação o mestre finaliza o processo de recepção, deixando a interpretação do timeout e/ou o pedido de retransmissão a cargo da aplicação.

### 3.2.3 Notas

O número de “retries” máximo é 2 (dois). Isto significa que ocorreram 3 (três) transmissões para uma situação de falha: a transmissão inicial e as duas retransmissões provocadas pelo “retry”. Todos os timeouts desse procedimento devem ser de 500 ms.

### 3.3 Select

Como discutido no item 2, o processo de “Select” começa quando o mestre necessita enviar dados para o escravo.

#### 3.3.1 Descrição

##### 3.3.1.1 Comportamento do mestre

Caso não esteja em andamento um processo de comunicação, o mestre transmite ao escravo a sequência DLE EOT (End Of Text) enviando em seguida uma mensagem, de acordo com o procedimento de transmissão e recepção de mensagens descrito no item 3.2.

No caso de ocorrência de falha de comunicação na transmissão da mensagem, o mestre tenta a recuperação da comunicação retransmitindo a mensagem (“retry”).

Caso o número de “retries” atinja o máximo permitido, o mestre executa seu procedimento de falha de comunicação (alarme, mensagem no display, etc.).

Ao receber ACK, o mestre é informado que a mensagem foi corretamente recebida pelo escravo. Neste momento o mestre finaliza o processo de “Select” podendo, desta maneira, pedir uma transmissão de dados pelo escravo (“Polling”-item 3.4) ou iniciar um novo “Select” para transmitir uma nova mensagem.

Caso ocorra “timeout” de espera pela confirmação da mensagem enviada para o escravo, o mestre tenta uma sequência de “retries”. Se ocorrer o estouro do número de “retries” permitidos o mestre executa o procedimento de falha de comunicação.

##### 3.3.1.2 Comportamento do escravo

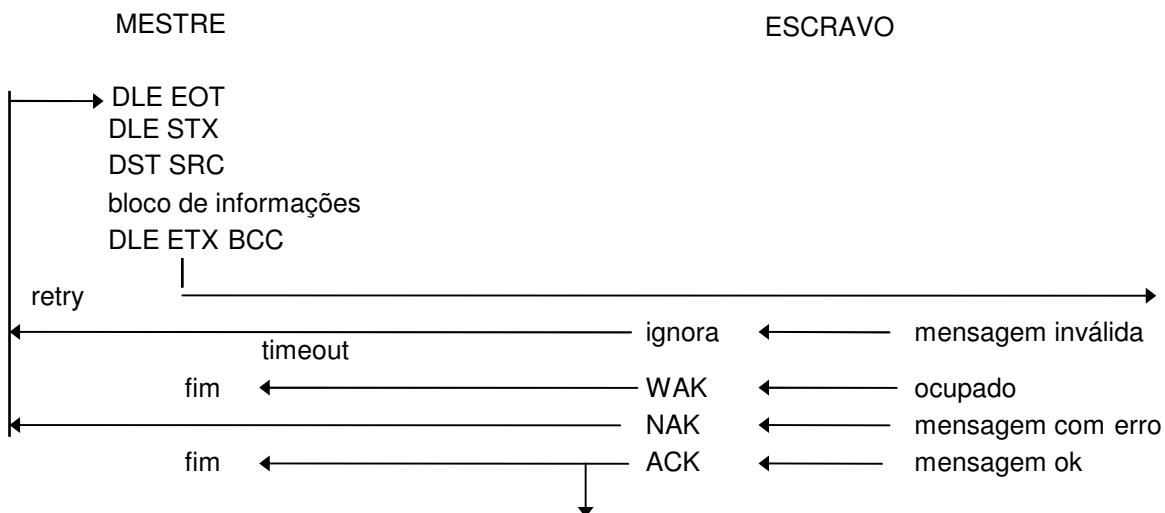
Quando não está ocorrendo um processo de comunicação todos os escravos da rede estão à espera de um “Select” ou um “Polling”. Ao receber a sequência de “Select” (DLE EOT), todos os escravos conectados à rede entendem que está se iniciando um “Select”. Os escravos disparam um timeout e passam a aguardar a recepção da mensagem.

Caso ocorra o timeout de espera da mensagem, os escravos ignoram o ocorrido e voltam a aguardar um novo “Select” ou um “Polling”. Ao iniciar a recepção da mensagem os escravos verificam se o endereço destino da mensagem corresponde ao seu endereço de rede. Os escravos não endereçados ignoram o “Select” e voltam a aguardar um novo “Select” ou um “Polling”.

Após a recepção correta de toda a mensagem do mestre, o escravo endereçado processa o comando e passa a enviar o Quadro de Resposta seguindo o procedimento de transmissão e recepção de mensagens como descrito no item 3.2. No caso de ocorrência de algum problema na transmissão do Quadro de Resposta, o escravo executa seu procedimento de falha de comunicação voltando a aguardar um novo “Select” ou “Polling” do mestre para reiniciar seu processo de comunicação.

Ao receber ACK, o escravo considera que finalizou o “Select” e volta a aguardar um novo “Select” ou um “Polling”.

### 3.3.2 Esquema de Transmissão e Recepção (SELECT)



# Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

**Obs. a sequência DLE EOT deve ser enviada somente quando é necessário “resetar” todos os escravos da linha.**

### 3.3 Notas

O número de “retries” máximo é 2 (dois). Isto significa que ocorreram 3 (três) transmissões para uma situação de falha: a transmissão inicial e as duas retransmissões provocadas pelo “retry”.

Caracteres de Controle utilizados no processo de Select:

EOT (End Of Transmission): Código ASCII = 04H - Indica ao escravo o início de um “Select” (“resseta a linha”).

WAK (Wait Acknowledgement): Código ASCII = 14H (DC4)

Os timeouts envolvidos no processo de Select devem ser de 500ms.

### 3.4 Polling

O processo de “Polling” é utilizado pelo mestre para verificar se o escravo possui algum comando pendente a ser enviado.

#### 3.4.1 Descrição

##### 3.4.1.1 Comportamento do Mestre

Caso não esteja em andamento um processo de comunicação, o mestre transmite ao escravo a sequência DLE ENQ DST, onde DST indica o endereço do escravo que está sendo interrogado. Após enviar o ENQ o mestre dispara uma contagem de tempo máximo (timeout) para aguardar uma resposta positiva ou negativa do escravo. Caso ocorra o timeout, o mestre finaliza o “Polling”, deixando a cargo da aplicação a decisão de uma nova sequência de “Polling”. No caso de estouro do número máximo de “retries”, o mestre executa o procedimento de falha de comunicação (alarme, mensagem no display, etc.).

Caso o mestre receba como resposta do escravo a sequência DLE EOT, significa que o escravo não possui nenhuma mensagem a ser enviada, e o mestre encerra o “Polling”.

Se o mestre receber a sequência DLE WAK significa que o escravo está impossibilitado de enviar a mensagem nesse momento. O mestre faz ou não uma sequência de retries segundo sua aplicação.

Após a recepção correta de todo o Quadro de Comando do escravo, o mestre processa o comando recebido e passa a enviar o Quadro de Resposta de acordo com o procedimento de transmissão e recepção de mensagens como descrito no item 3.2.

Caso ocorra algum problema na transmissão do Quadro de Resposta o mestre executa um “retry” de todo o “Polling”, até o número máximo de “retries” permitido e caso não consiga que o escravo receba o Quadro de Resposta corretamente, executa sua sequência de falha de comunicação.

Ao receber ACK o mestre considera finalizado o “Polling”.

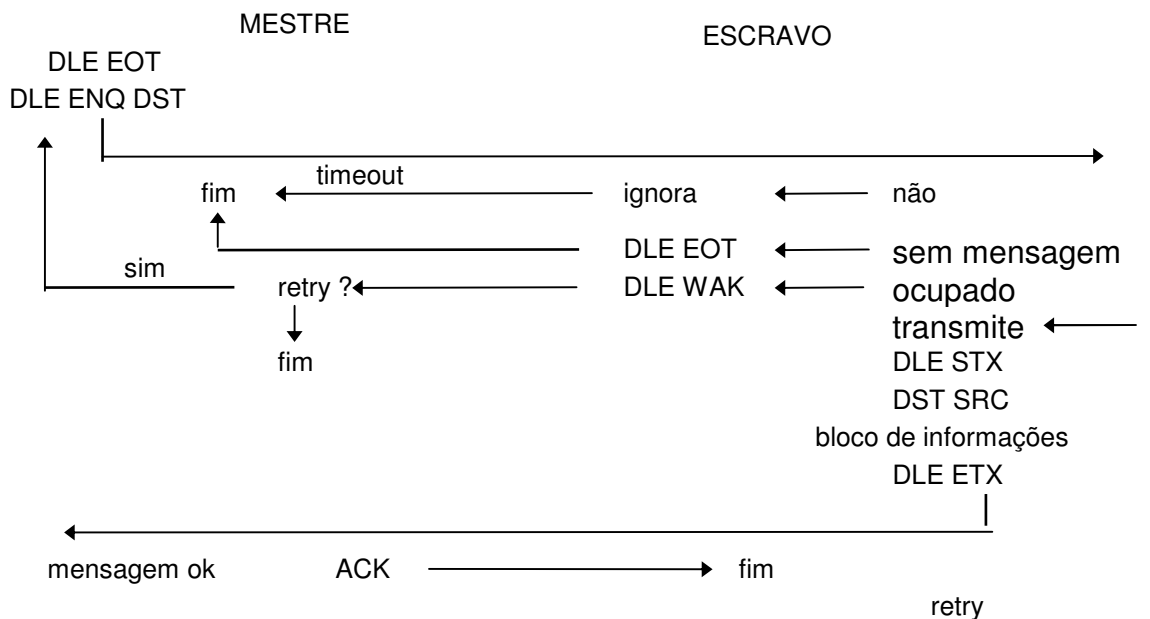
##### 3.4.1.2 Comportamento do Escravo

Ao receberem a sequência DLE ENQ DST todos os escravos da rede verificam se o campo DST (destino da mensagem) contém seu endereço de rede. Os escravos não endereçados ignoram o “Polling” e voltam a aguardar novo “Polling” ou “Select”. O escravo endereçado verifica se possui algum comando a enviar ao mestre e caso não exista envia a sequência DLE EOT ao mestre encerrando o processo de “Polling”.

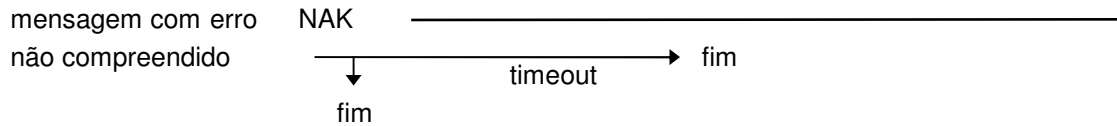
No caso de possuir uma mensagem pendente, envia imediatamente a mensagem ao mestre, seguindo os procedimentos de transmissão e recepção de mensagens como descrito no item 3.2. Quando ocorre falha de comunicação, o escravo não responde mais ao mestre até receber um novo “Polling” ou um “Select”.

Se o escravo estiver ocupado ele transmite ao mestre a sequência DLE WAK informando ao mestre a impossibilidade de enviar qualquer mensagem e volta a esperar um novo “Select” ou “Polling”.

#### 3.4.2 Esquema de transmissão e recepção (Polling)



## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0



**obs. a sequência DLE EOT enviada pelo mestre ao escravo deve ser utilizada somente quando for necessário “resetar” todos os escravos da linha.**

### 3.4.3 Notas

O número de “retries” máximo é 2 (dois). Isto significa que ocorreram 3 (três) transmissões para uma situação de falha: a transmissão inicial e as duas retransmissões provocadas pelo “retry”.

Caracteres de Controle utilizados no processo de Polling:

EOT (End Of Transmission): Código ASCII = 04H

ENQ (ENQUIRE): Código ASCII = 05H

WAK (Wait Acknowledge) : Código ASCII = 14H (DC4)

DLE EOT: Quando enviado pelo escravo ao mestre indica que não existe mensagem a ser enviada. Deve portanto ocorrer novo “select”.

DLE WAK: Quando enviado pelo escravo ao mestre indica que está impossibilitado de enviar a mensagem. Se ocorrer outro “polling” estará vinculado ao último “select”.

DLE ENQ: Indica ao escravo o início de um processo de “Polling”.

NAK (Negative Acknowledgement) Código ASCII = 15H. Resposta informando que uma string foi recebida com erro.

ACK (positive Acknowledgement) Código ASCII = 06H. Resposta informando que uma string foi recebida corretamente.

Os timeouts envolvidos no processo de “Polling” devem ser de 500ms.

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

### 4. Protocolo em termos de Aplicação

O protocolo do lado da aplicação se preocupa com o tratamento das informações embutidas em uma mensagem enviada ou recebida pela rede de comunicação.

Quando operando com protocolo ALFA Instrumentos, os parâmetros para comunicação são os seguintes:

- 19200 bps, 1 START BIT, 8 DATA BITS, SEM paridade, 1 STOP BIT

#### 4.1 Quadro de comando

Como já foi dito, Quadro de Comando é o nome dado à mensagem enviada na rede pelo equipamento que deseja requisitar ao outro equipamento conectado a execução de determinada tarefa.

A tarefa requisitada pode ser o envio de dados de processo, envio de dados de alarme, a alteração de parâmetros internos, a alteração de programação, etc..

O Quadro de Comando é constituído pelas sequências de controle e o bloco de informações com os dados referentes a um comando.

O comando é formado pelo código de comando seguido de seus parâmetros.

CMD	PARÂMETROS
-----	------------

O código de comando é um número binário de 00H até 7FH, isto é, podem existir até 128 códigos de comando + códigos de resposta sendo que cada código possui uma determinada função e aplicação que estão descritas no item Conjunto de Comandos.

#### 4.2 Quadro de resposta

Quadro de Resposta é o nome dado à mensagem enviada através da rede de comunicação pelo equipamento que executou um comando requisitado. Essa mensagem pode conter dados de processo, status do equipamento, dados de alarme, confirmação de alteração de parâmetros internos, confirmação de alteração de programação, etc..

O Quadro de Resposta é constituído pelas sequências de controle mais o bloco de informações com os dados referentes à resposta.

RSP	PARÂMETROS
-----	------------

#### 4.3 Conjunto de comandos

A seguir a apresentação e descrição do conjunto de comandos executados pela rede Alfa. Novos comandos podem ser criados conforme a necessidade de utilização.

##### 4.3.1 Verificação do status do escravo (01H)

Utilizado para reconhecimento pelo mestre da posição atual do escravo.

CMD= 01H
----------

A resposta enviada pelo escravo :

RSP= 01H	POSIÇÃO
----------	---------

00H = estado de indicação de pesos  
01H a 0DH = estado de ajuste de parâmetros.  
0EH a 11H = estado de ajuste de valores de setpoints  
13H a 18H = estado de autocalibração

##### 4.3.2 Inicialização individual dos escravos (02H)

Utilizado para colocação de todos os escravos em um ponto definido do processo. No caso específico do Indicadores de Pesagem o ponto do processo é a indicação de pesos ( estado de indicação de pesos). Este comando também permite confirmar ao escravo que o mestre identificou a alteração de parâmetros do equipamento via display local (sinalizada pelo escravo através do status do comando 08H) e já atualizou a nova configuração.

CMD= 02H
----------

A resposta enviada pelo escravo confirma a execução do comando.

RSP= 02H
----------

##### 4.3.3 Programação de Setpoints (03H) modelos 3104/7

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=03H	STATUS	SETPOINT 1	-----	SETPOINT N
---------	--------	------------	-------	------------

status (1byte) = bit 7 em 1.  
bit 6 em 0 - não gravar em memória não volátil.  
1 - gravar em memória não volátil.  
bits 0 a 5 = número de setpoints em binário.

Setpoints = cada valor é composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

**Obs.:** O número de setpoints igual a 4, sendo o último igual ao valor programado para opção da configuração "VAZIA". Todos valores devem sempre ser passados independentemente à quantidade de valores a serem alterados.

### 4.3.4 Programação do Setpoint0 (30H) modelos 3104C/7C

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=30H	STATUS	SETPOINT0	TRAVA0
---------	--------	-----------	--------

STATUS (1byte) = bit 7 fixo em 1  
bit 6 em 0 – não gravar em memória não volátil  
em 1 – gravar em memória não volátil  
bits 5 ..1 – fixos em 0  
bit 0 em 0 – setpoint0 opera como VAZIA  
1 – SETPOINT0 opera como NIVEL0

SETPOINT0 = composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )  
TRAVA0 = '0' (30H - sem trava), '1' (31H - com trava).

### 4.3.5 Programação do Setpoint1 (31H) modelos 3104C/7C

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=31H	STATUS	SETPOINT1	TRAVA1
---------	--------	-----------	--------

STATUS (1byte) = bit 7 fixo em 1  
bit 6 em 0 – não gravar em memória não volátil  
em 1 – gravar em memória não volátil  
bits 5 ..0 – fixos em 0

SETPOINT1 = composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )  
TRAVA1 = '0' (30H - sem trava), '1' (31H - com trava).

### 4.3.6 Programação do Setpoint2 (32H) modelos 3104C/7C

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=32H	STATUS	SETPOINT2	TRAVA2
---------	--------	-----------	--------

STATUS (1byte) = bit 7 fixo em 1  
bit 6 em 0 – não gravar em memória não volátil  
em 1 – gravar em memória não volátil  
bits 5 ..1 – fixos em 0  
bit 0 em 0 – setpoint0 opera como VAZIA  
1 – SETPOINT0 opera como NIVEL0

SETPOINT2 = composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )  
TRAVA2 = '0' (30H - sem trava), '1' (31H - com trava).

### 4.3.7 Programação do Setpoint3 (33H) modelos 3104C/7C

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=33H	STATUS	SETPOINT3	TRAVA3
---------	--------	-----------	--------

STATUS (1byte) = bit 7 fixo em 1  
bit 6 em 0 – não gravar em memória não volátil  
em 1 – gravar em memória não volátil  
bits 5 ..0 – fixos em 0

SETPOINT3 = composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )  
TRAVA3 = '0' (30H - sem trava), '1' (31H - com trava).

### 4.3.8 Verificação dos setpoints programados (04H) - modelos 3104/7

Utilizado para verificação de todos valores de nível de corte programados.

CMD= 04H
----------

A resposta enviada segue formato:

RSP=04H	STATUS	SETPOINT 1	-----	SETPOINT N
---------	--------	------------	-------	------------

status (1 byte) = bit 7 em 1.  
bit 6 em 0.  
bits 0 a 5 número de setpoints em binário.

Setpoints = cada valor é composto de 5 bytes em ASCII.

**Obs.** O setpoint 0, refere-se a programação da opção "VAZIA".

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

### 4.3.9 Programação da configuração dos Setpoints (05H)

Utilizado para programação de todos parâmetros de configuração dos setpoints.

CMD=05H	HISTERESE	LÓGICA RELE	STATUS	TRAVA 1
---------	-----------	-------------	--------	---------

-----	TRAVA N
-------	---------

histerese (2 bytes ASCII) = valor em % de 00 a 99.

lógica rele (1 byte ASCII) = '0' (30H - normalmente aberto), '1' (31H - normalmente fechado).

status (3 bytes) = bit 7 em 1.

bit 6\* = 0 não gravar configuração em memória não volátil.

1 gravar configuração em memória não volátil.

bits 0 a 5 número de setpoints em binário. Se igual a 4, define SETPOINT0 operando como NIVEL0

trava (1 byte ASCII) = '0' (30H - sem trava), '1' (31H - com trava).

\* cuidado com o envio do status bit 6, pois a memória não volátil é uma E2PROM, com vida de gravação limitada à aproximadamente 100.000 gravações.

### 4.3.10 Verificação da configuração dos setpoints programados (06H)

Utilizado para verificação da configuração de todos setpoints programados .

CMD=06H
---------

A resposta enviada segue formato:

RSP=06H	HISTERESE	LÓGICA RELE	N. SETPOINTS	TRAVA 1
---------	-----------	-------------	--------------	---------

-----	TRAVA N
-------	---------

histerese (2 bytes ASCII) = valor em % de '00' a '99'.

lógica relê (1 byte ASCII) = '0' (30H - normalmente aberto), '1' (31H - normalmente fechado).

n.setpoints (1 byte) = bit 7 em 1

bit 6 em 0

bits 0 a 5: número de setpoints em binário

trava x (3 bytes ASCII) = '0' (30H - sem trava), '1' (31H - com trava)

### 4.3.11 Programação do endereço do escravo (07H)

Coloca um novo endereço de nó da rede ao escravo de endereço "0". Todo escravo quando é conectado a primeira vez assume seu endereço como "0", a não ser que programado manualmente.

CMD=07H	NOVO ENDEREÇO
---------	---------------

novo endereço (2 bytes ASCII) = de '00' à '99'.

Ao receber este comando imediatamente o escravo assume o novo endereço programado, rejeitando a comunicação com o endereço anterior.

### 4.3.12 Requisição de Pesos e status geral (08H)

Solicita ao escravo o envio dos pesos e status geral do Transmissor.

CMD=08H
---------

A resposta enviada pelo escravo:

RSP=08H	STATUS 1	STATUS 2	PESO DISPLAY	TARA
---------	----------	----------	--------------	------

status 1:

bit 7 = 1

bit 6 = Alarme de sobrecarga

bit 5 = Alarme de saturação

bit 4 = Detecção de movimento de peso

bit 3 = peso negativo no display

bit 0 a 2 = posição do ponto decimal em binário

status 2:

bit 7 = 1

bit 6 = 1, ocorreu alteração local de parâmetros via display/teclado

bit 5 = sistema em modo PESO BRUTO

bit 4 = sistema operando com ZERO FIXO

bit 3 = passagem pelo nível de corte 0

bit 2 = passagem pelo nível de corte 3

bit 1 = passagem pelo nível de corte 2

bit 0 = passagem pelo nível de corte 1

peso display = 5 bytes ASCII

tara = 5 bytes ASCII

### 4.3.13 Acionamento de funções via rede (09H)

CMD=09H	TECLAS
---------	--------

teclas :

- bit 0 = zero
- bit 1 = tara
- bit 2 = RESERVADO
- bit 3 = destara
- bit 4 = destrava
- bit 5 = imprime (3104C/7C)
- bit 6 = acumula (3104C/7C)
- bit 7 = RESERVADO

### 4.3.14 Programação da faixa de peso para atuação da saída analógica (0AH) - modelo 3107

Determinação da faixa de atuação 4/20mA ou 0/20mA e a sua referência de pesagem ( peso bruto ou Líquido);

CMD=0AH	REFERÊNCIA	PESO 4 MA OU 0 MA	PESO 20 MA
---------	------------	-------------------	------------

- peso 4mA ou 0mA = 5 bytes em ASCII.
- peso 20mA = 5 bytes em ASCII.
- referência = 1 byte em ASCII, ('B' - peso bruto), ('L' - peso líquido)

### 4.3.15 Verificação da faixa de peso para atuação da saída analógica (0BH) - modelo 3107

Verificação da faixa de atuação 4/20mA

CMD=0BH
---------

A resposta enviada segue formato:

RSP=0BH	REFERÊNCIA	PESO 4 MA OU 0 MA	PESO 20 MA
---------	------------	-------------------	------------

- peso 4mA ou 0mA = 5 bytes em ASCII.
- peso 20mA = 5 bytes em ASCII.
- referência = 1 byte em ASCII, ('B' - peso bruto), ('L' - peso líquido)

### 4.3.16 Verificação e controle acumulador (0CH) - modelos 3104/6/7

Utilizado nos equipamentos que possuam função de acumulador de peso. Através deste comando pode-se ler o valor acumulado, zerar e obter informações de status.

CMD=0CH	STATUS
---------	--------

- status (1 byte) = bit 0 em 1 - ordem para acumular
- bit 1 em 1 - ordem para limpar o valor acumulado

A resposta enviada pelo escravo confirma a execução do comando e o valor atual acumulado no equipamento.

RSP=0CH	STATUS	VALOR ACUMULADO	ÚLTIMO PESO ACUMULADO
---------	--------	-----------------	-----------------------

- status (1 byte) = bit 6 em 1 - o equipamento não pode acumular pois ultrapassou a capacidade do buffer
- bit 5 em 1 - não pode acumular devido a balança estar em movimento.
- bit 4 em 1 - não pode acumular devido a operação já ter sido realizada com comando manual ou através da rede de comunicação, sem ter sido liberada (volta do Indicador a 0)
- bit 0,1 e 2 - posição da casa decimal do valor acumulado.

- valor acumulado = 8 bytes em ASCII
- último peso acumulado = 5 bytes em ASCII

### 4.3.17 Leitura dos dados de acumulação (0EH) - modelos 3104C/7C

Utilizado nos equipamentos que possuam função de acumulador de peso. Através deste comando pode-se ler todos os dados relativos à função de acumulação.

CMD=0EH
---------

A resposta enviada segue formato:

RSP=0EH	STATUS	TOTAL ACUMULADO	TOTALIZAÇÕES	PESO MÉDIO
---------	--------	-----------------	--------------	------------

- status (1 byte) = bits 7 ..3 = 0
- bit 0,1 e 2 - posição da casa decimal do valor acumulado.

- valor acumulado = 8 bytes em ASCII
- totalizações = 5 bytes em ASCII
- peso médio = 5 bytes em ASCII

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

### 4.3.18 Programação do Relógio – Calendário (10H) - modelo 3104C/7C

Programa DATA/HORA.

CMD=10	DIA	MES	ANO	HORAS	MINUTOS	SEGUNDO
--------	-----	-----	-----	-------	---------	---------

dia = 2 bytes em ASCII  
mes = 2 bytes em ASCII  
ano = 2 bytes em ASCII, contendo apenas os dígitos da dezena e unidade  
horas = 2 bytes em ASCII, no padrão 00..23  
minutos = 2 bytes em ASCII  
segundos = 2 bytes em ASCII

### 4.3.19 Leitura do Relógio – Calendário (11H) - modelo 3104C/7C

Leitura dos dados de DATA/HORA.

CMD=11H
---------

A resposta enviada segue formato:

CMD=11H	DIA	MES	ANO	HORAS	MIN	SEG	SEMANA
---------	-----	-----	-----	-------	-----	-----	--------

dia = 2 bytes em ASCII  
mes = 2 bytes em ASCII  
ano = 2 bytes em ASCII, contendo apenas os dígitos da dezena e unidade  
horas = 2 bytes em ASCII, no padrão 00..23  
minutos = 2 bytes em ASCII  
segundos = 2 bytes em ASCII  
semana = 1 byte em ASCII: 31H = segunda, 32H = terça, ... 36H = sábado, 37H = domingo

### 4.3.20 Captura do Peso ALVO – função SOBRA-FALTA (12H) - modelo 3104C/7C

Lê o peso aplicado à balança definindo-o com peso ALVO.

CMD=12H
---------

A resposta enviada pelo escravo:

RSP=12H	PESO ALVO
---------	-----------

peso alvo = 5 bytes ASCII

### 4.3.21 Programação do PESO ALVO e TOLERÂNCIA (13H) - modelo 3104C/7C

Programa PESO ALVO e respectiva tolerância.

CMD=13H	PESO ALVO	TOLERÂNCIA
---------	-----------	------------

função = 1 byte em ASCII, ('0' - desabilitada), ('1' - habilitada)  
visual = 1 byte em ASCII, ('0' - pontual), ('1' - bargraph)

### 4.3.22 Leitura do Peso ALVO e TOLERÂNCIA (14H) - modelo 3104C/7C

Solicita ao escravo o envio dos pesos e status geral do Transmissor.

CMD=14H
---------

A resposta enviada pelo escravo:

RSP=14H	PESO ALVO	TOLERÂNCIA
---------	-----------	------------

peso alvo = 5 bytes ASCII  
tolerância = 5 bytes ASCII

### 4.3.23 Programação dos parâmetros da função SOBRA-FALTA (15H) - modelo 3104C/7C

Determinação da faixa de atuação 4/20mA ou 0/20mA e a sua referência de pesagem ( peso bruto ou Líquido);

CMD=15H	FUNÇÃO	VISUAL	TOLERÂNCIA	SAÍDAS	ALARME
---------	--------	--------	------------	--------	--------

função = 1 byte em ASCII, ('0' - desabilitada), ('1' - habilitada)  
visual = 1 byte em ASCII, ('0' - pontual), ('1' - bargraph)  
tolerância = 1 byte em ASCII, ('0' - percentual), ('1' - numérica)  
saídas = 1 byte em ASCII, ('0' - desabilitadas), ('1' - habilitadas)  
quando habilitadas as saídas de níveis serão atuadas nas seguintes condições:  
- SETPOINT0: peso na balança <= 1% do peso ALVO  
- SETPOINT1: peso na balança ABAIXO do peso: ALVO – TOLERÂNCIA  
- SETPOINT2: peso na balança em torno do peso: ALVO + / – TOLERÂNCIA  
- SETPOINT3: peso na balança ACIMA do peso: ALVO + TOLERÂNCIA  
alarme = 1 byte em ASCII, ('0' - desabilitado), ('1' - habilitado)

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

### 4.3.24 Leitura dos parâmetros da função SOBRA-FALTA (16H) - modelo 3104C/7C

Leitura dos dados de DATA/HORA.

CMD=16H
---------

A resposta enviada segue formato:

CMD=16H	FUNÇÃO	VISUAL	TOLERÂNCIA	SAÍDAS	ALARME
---------	--------	--------	------------	--------	--------

função = 1 byte em ASCII, ('0' - desabilitada), ('1' - habilitada)  
visual = 1 byte em ASCII, ('0' - pontual), ('1' - bargraph)  
tolerância = 1 byte em ASCII, ('0' - percentual), ('1' - numérica)  
saídas = 1 byte em ASCII, ('0' - desabilitadas), ('1' - habilitadas)  
alarme = 1 byte em ASCII, ('0' - desabilitado), ('1' - habilitado)

### 4.3.25 Programação da função ZERO (17H) - modelo 3104C/7C

Programa modo de operação da função ZERO.

CMD=17H	MODO	BOOT
---------	------	------

modo = 1 byte em ASCII:  
'0' = função desabilitada  
'1' = modo automático  
'2' = modo manual  
'3' = modo automático e manual  
boot = 1 byte em ASCII, ('0' - busca de zero ao ligar desabilitada), ('1' - busca habilitada)

### 4.3.26 Programação do modo de operação da TARA (19H) - modelo 3104C/7C

Programa modo de operação da TARA.

CMD=19H	MODO
---------	------

modo = 1 byte em ASCII:  
'0' = não sucessiva  
'1' = não sucessiva e memorizada  
'2' = sucessiva  
'3' = sucessiva e memorizada  
'4' = editável  
'5' = editável e memorizada  
'8' = desabilitada

### 4.3.27 Programação da tipo do FILTRO (1BH) - modelo 3104C/7C

Programa o filtro a ser utilizado.

CMD=1BH	FILTRO	VISUALIZAÇÃO
---------	--------	--------------

filtro = 1 byte em ASCII:  
'0','1','2' = para aplicações que requeiram repostas rápidas em plataformas com capacidade inferior a 120 kg  
'3','4','5','6' = para aplicações que requeiram repostas rápidas em plataformas com capacidade superior a 120 kg  
'7','8' = para aplicações com cargas móveis  
'9' = para aplicações com auto grau de vibrações  
visualização = 1 byte em ASCII, ('0' - não mostra peso até estabilizar), ('1' - sempre mostra o peso)

### 4.3.28 Programação do valor da TARA EDITÁVEL (55H) - modelo 3104C/7C

Programa valor da TARA.

CMD=55H	VALOR TARA
---------	------------

valor tara = 5 bytes ASCII

### 4.3.29 Processo de Calibração Completo

A calibração dos Transmissores Digital, consiste em comandos que permitem fazer leitura de sistema vazio (sem peso), leitura de sistema com peso de calibração e passagem de parâmetros. Estes comandos (etapas) podem ser realizados em qualquer sequência, mas com a finalização do processo utilizando o comando "54H" o qual faz a geração da constante de calibração e armazenamento dos parâmetros.

#### 4.3.29.1 Passagem de parâmetros de calibração (50H)

Faz a passagem de todos parâmetros existentes no menu de calibração do Transmissor Digital.

Os parâmetros de calibração podem ser alterados em qualquer instante, mesmo que já se tenha feito um processo de calibração, não esquecendo de se utilizar o comando "54H" para finalização.

CMD=50H	PARAMETROS
---------	------------

Parâmetros (todos em ASCII, nesta ordem):  
casas decimais (1 byte) = '0,1,2,3 ou 4'.

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

degrau (1 byte)	=	'1, 2 ou 5'.
peso de calibração		(5 bytes)
capacidade máxima		(5 bytes)
autozero (1 byte)	=	('0' - zero manual e automático desativados),('1' - zero automático),('2' - zero manual),('3' - zero manual e automático ativados).
filtro digital (1 byte)	=	('0' - rápido a '8' - lento),
tara (1 byte)	=	('0' - não sucessiva),('1' - não sucessiva memorizada),('2' - sucessiva), ('3' - sucessiva memorizada)

### 4.3.29.2 Verificação dos parâmetros de calibração (51H)

Obtém todos os parâmetros de calibração do equipamento.

**CMD=51H**

resposta do Transmissor ao comando:

**RSP=51H**      **PARÂMETROS**

Parâmetros (todos em ASCII, nesta ordem):

casas decimais (1 byte)	=	'0,1,2,3 ou 4'.
degrau (1 byte)	=	'1, 2 ou 5'.
peso de calibração		(5 bytes)
capacidade máxima		(5 bytes)
autozero (1 byte)	=	('0' - zero manual e automático desativados),('1' - zero automático),('2' - zero manual),('3' - zero manual e automático ativados).
filtro digital (1 byte)	=	('0' - rápido a '8' - lento),
tara (1 byte)	=	('0' - não sucessiva),('1' - não sucessiva memorizada),('2' - sucessiva), ('3' - sucessiva memorizada)

### 4.3.29.3 Etapa de calibração sem peso (52H)

Determina o início de autocalibração com sistema sem peso. Neste ponto o sistema de pesagem não deve conter nenhum peso sobre o sistema e que os acessórios que fazem parte do peso morto estejam em seus lugares de trabalho.

**CMD=52H**

A resposta enviada pelo Transmissor ao terminar esta etapa.

**RSP=52H**      **STATUS**

status (1 byte em ASCII) = ('0' - sem erro),(3 - peso instável),( 6 - conversão fora dos limites).

### 4.3.29.4 Etapa de calibração com peso (53H)

Determina o início de autocalibração com o peso de calibração. Neste ponto o peso deve estar colocado sobre o sistema.

**CMD=53H**

A resposta enviada pelo Transmissor ao terminar esta etapa.

**RSP=53H**      **STATUS**

status (1 byte em ASCII) = ('0' - sem erro),(3 - peso instável),( 6 - conversão fora dos limites).

### 4.3.29.5 Geração da constante de calibração (54H)

Calcula a constante e armazena parâmetros de calibração. Este comando deve ser feito sempre ao final do processo de calibração.

**CMD=54H**

A resposta enviada pelo Transmissor ao terminar esta etapa.

**RSP=54H**      **STATUS**

status (1 byte em ASCII) = ('0' - sem erro),('1' - peso balança vazia >= peso de calibração),('2' - span insuficiente),('9' - peso de calibração > capacidade máxima)

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

\*\*\*\*\* COMANDOS EXCLUSIVOS PARA O INDICADOR 3108 \*\*\*\*\*

### 4.3.30 Programação de Setpoints (11H) modelo 3108

Utilizado para programação de valores de nível de corte (setpoints) do escravo.

CMD=11H	STATUS	SETPOINT 1	-----	SETPOINT 4
---------	--------	------------	-------	------------

status (1byte) = bit 7 em 1.

bit 6 em 0 - não gravar em memória não volátil.

1 - gravar em memória não volátil.

bits 0 a 5 número de setpoints em binário.

Setpoints = cada valor é composto de 5 bytes em ASCII ( ex. setpoint =10000 --> 31 30 30 30 30 )

**Obs.:** Os setpoints 1,2 e 4 são referenciados ao peso, e o setpoint 3 com a vazão.

O número de setpoints igual a 4, sendo o último igual ao valor programado para opção da configuração "VAZIA".

Todos valores devem sempre ser passados independentemente à quantidade de valores a serem alterados.

### 4.3.31 Verificação dos setpoints programados (12H) modelo 3108

Utilizado para verificação de todos valores de nível de corte programados.

CMD= 12H
----------

A resposta enviada segue formato:

RSP=12H	STATUS	SETPOINT 1	-----	SETPOINT 4
---------	--------	------------	-------	------------

status (1 byte) = bit 7 em 1.

bit 6 em 0.

bits 0 a 5 número de setpoints em binário.

Setpoints = cada valor é composto de 5 bytes em ASCII.

**Obs.** O setpoint 4, refere-se a programação da opção "VAZIA".

### 4.3.32 Programação da faixa de vazão para atuação da saída analógica (13H) modelo 3108

Determinação da faixa de atuação 4/20mA ou 0/20mA e a taxa de amostragem ;

CMD=13H	TAXA AMOSTRA	VAZÃO 4 MA OU 0 MA	VAZÃO 20 MA
---------	--------------	--------------------	-------------

vazao 4mA ou 0mA = 5 bytes em ASCII.

vazao 20mA = 5 bytes em ASCII.

Taxa de amostragem = 1 byte em ASCII :

'0' ( 0,1 seg), '1'( 1 seg), '2'(10 seg), '3'(60 seg).

### 4.3.33 Verificação da faixa de vazão para atuação da saída analógica (14H) modelo 3108

Verificação da faixa de atuação 4/20mA

CMD=14H
---------

A resposta enviada segue formato:

RSP=14H	TAXA AMOSTRA	VAZÃO 4 MA OU 0 MA	VAZÃO 20 MA
---------	--------------	--------------------	-------------

vazao 4mA ou 0mA = 5 bytes em ASCII.

vazao 20mA = 5 bytes em ASCII.

Taxa de amostragem = 1 byte em ASCII :

'0' ( 0,1 seg), '1'( 1 seg), '2'(10 seg), '3'(60 seg).

### 4.3.34 Requisição de vazão e status (15H) modelo 3108

Solicita ao escravo o envio dos pesos e status geral do Transmissor.

CMD=15H
---------

A resposta enviada pelo escravo:

RSP=15H	STATUS 1	STATUS 2	VAZÃO DISPLAY
---------	----------	----------	---------------

vazão display (5 bytes ASCII)

status 1:

bit 7 = 1

bit 6 = Alarme de sobrecarga (referente ao peso)

bit 5 = Alarme de saturação (referente ao peso)

bit 4 = Alarme medida de vazão > 65535 kg/min

bit 3 = vazão negativa no display

bit 0 a 2 = posição do ponto decimal em binário

status 2:

bit 7 = 1

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

bit 6 = 1 , ocorreu alteração local de parâmetros via display/teclado  
bit 3 = passagem pelo nível de corte 0  
bit 2 = passagem pelo nível de corte 3  
bit 1 = passagem pelo nível de corte 2  
bit 0 = passagem pelo nível de corte 1

\*\*\*\*\* *COMANDOS EXCLUSIVOS PARA O INDICADOR 3106* \*\*\*\*\*

### 4.3.35 Requisição do último Peso Armazenado no Acumulador (18H) - modelo 3106

Solicita ao escravo o envio dos pesos e status geral do Transmissor.

CMD=18H
---------

A resposta enviada pelo escravo:

RSP=18H	PESO ARMAZENADO
---------	-----------------

peso armazenado (5 bytes ASCII)

### 4.3.36 Limpa último Peso Armazenado no Acumulador (19H) - modelo 3106

Ao enviar este comando o mestre indica ao equipamento que pode retirar o sinalização de peso armazenado (ver comando anterior).

CMD=19H
---------

A resposta enviada pelo escravo:

RSP=19H
---------

## Protocolo de Comunicação Serial Alfa Instrumentos versão 2.0

### **Protocolo de Comunicação - exemplo de uma transmissão completa e correta de mensagem entre mestre-escravo**

comando de requisição peso e status geral (08H)  
 indicação de peso 29.998 kg  
 tara= 00.000 kg  
 endereço indicador = 1, endereço do mestre = 0

**MESTRE**

**ESCRAVO(3104)**

DLE STX DST SRC CMD DLE ETX BCC →  
 10H 02H 01H 00H 08H 10H 03H A6H

← ACK  
 06H

DLE ENQ DST →  
 10H 05H 01H

← DLE STX DST SRC CMD ----- INFORMAÇÃO ----- DLE ETX BCC  
 10H 02H 00H 01H 08H 83H 83H 32H 39H 39H 39H 38H 30H 30H 30H 30H 30H 10H 03H 0FH

ACK →  
 06H

BCC= é calculado de STX até ETX (inclusive) sendo o OU exclusivo do byte de dados atual, com o BCC anterior seguido de um rotate left de uma posição.

No exemplo acima na transmissão do mestre:

0000 0000	0001 1100
0000 0010 02H (STX)	0011 1000 rotate
-----	0001 0000 10H (DLE)
0000 0010 xor	-----
0000 0100 rotate	0010 1000 xor
0000 0001 01H (DST)	0101 0000 rotate
-----	0000 0011 03H (ETX)
0000 0101 xor	-----
0000 1010 rotate	0101 0011 xor
0000 0000 00H (SRC)	1010 0110 rotate
-----	
0000 1010 xor	BCC= A6H
0001 0100 rotate	
0000 1000 08H (CMD)	
-----	
0001 1100 xor	

observação: quando ocorrer um dado de valor igual a um caracter de controle DLE, deve-se enviar duas vezes o valor DLE para que o receptor interprete como dado. exemplo : vamos supor que o endereço do escravo = 16 (10H) a transmissão fica da seguinte forma (display indicando 29.999 kg):

**MESTRE**

**ESCRAVO(3104)**

DLE STX DST SRC CMD DLE ETX BCC →  
 10H 02H 10H 10H 00H 08H 10H 03H 84H

← ACK  
 06H

DLE ENQ DST →  
 10H 05H 10H 10H

← DLE STX DST SRC CMD ----- INFORMAÇÃO ----- DLE ETX BCC  
 10H 02H 00H 10H 10H 08H 83H 83H 32H 39H 39H 39H 39H 30H 30H 30H 30H 30H 10H 03H 1FH

ACK →  
 06H

Para o cálculo do BCC deve-se considerar apenas um DLE quando for duplicado.



R. Cel. Mário de Azevedo, 138 São Paulo - SP

02710-020 Brasil

Tel: (11) 3952-2299 Fax: (11) 3961-4266

SAC: 0800-772-2910

<http://www.alfainstrumentos.com.br>