

Sistema de automação

PC - Horustech - TWC

Protocolo de Comunicação HRS-1-Terminal

DT472

Sumário

[Introdução 3](#_Toc421094304)

[Estrutura do Comando 3](#_Toc421094305)

[1. ENVIO DE MENSAGENS AO TERMINAL 4](#_Toc421094306)

[2. REQUISIÇÃO DE RESPOSTAS OU PERGUNTAS DO TERMINAL 5](#_Toc421094307)

[3. LIMPA PENDÊNCIAS DE TERMINAL 6](#_Toc421094308)

[4. LIMPA PENDÊNCIA E ENVIO DE MENSAGENS AO TERMINAL 6](#_Toc421094309)

[5. PEDIDO DE STATUS DOS TERMINAIS 6](#_Toc421094310)

[6. GRAVAÇÃO DE IDENTIFICADOR V1.0 7](#_Toc421094311)

[7. ABASTECIMENTO HORUSTECH 8](#_Toc421094312)

[8. ABASTECIMENTO COMPANYTEC - CBC 9](#_Toc421094313)

[9. FORMATAÇÃO DAS MENSAGENS CODIFICADAS 10](#_Toc421094314)

[10. Códigos de erro 15](#_Toc421094315)

[11. Códigos de Status de execução 15](#_Toc421094316)

[12. Cálculo de Check-Sum 16](#_Toc421094317)

Convenções Gerais

# Introdução

* Todos os comandos são iniciados com o caractere “>“, seguido de”? ”para consulta e ”! ”para resposta da automação.
* Após os caracteres de início de bloco, o protocolo informa o tamanho dos dados, em hexadecimal.
* O campo de dados é dividido em duas partes, o índice do comando, informado em hexadecimal de dois caracteres (00..FF), e os parâmetros, que serão de tamanho variável, de acordo com a informação do cabeçalho.
* Campos “bico” são representados em decimal, de 0 a 99;
* Campos tanque e combustível são representados em decimal, de 0 a 99;
* Campos número do terminal são representados em decimal, de 0 a 99;
* A Companytec recomenda que a comunicação com o Terminal seja desenvolvida de forma independente a da automação, se possível deve-se utilizar até mesmo outra porta para comunicação, como por exemplo a 771. Esta recomendação se dá devido a velocidade necessária as comunicações do software gerencial com o TWC.

# Estrutura do Comando

Todos os comandos são definidos como figura abaixo, os comandos consistem em uma camada de transporte, uma região de dados e um checksum respectivamente.

|  |
| --- |
| >PCCCCX.... KK |

* Camada de Transporte:

> : Delimitador;

P : Tipo de Comando:

? : Consulta para automação;

! : Resposta da automação;

C[4] :Tamanho do campo DATA em Hexadecimal;

* Camada de Dados:
* X[2..65535] :Dados do comando:

Tipo [2] : Índice do comando;

Parâmetros [0..65532] : Parâmetros auxiliares do comando;

* Checksum:
* K[2] : Somatório dos valores ASCII dos caracteres do comando, sendo desprezado o byte mais significativo. O caractere delimitador ‘>’ é excluído do calculo;

OBS: Todos comandos podem ser setados para responder com o numero de série do console. Esta função é ativada quando o bit 7 do índice de comando está ligado.

**Observação: Os comandos presentes neste documento estão implementados somente no concentrador Horustech, com versão superior a 6.4**.

Comandos

# ENVIO DE MENSAGENS AO TERMINAL

* Descrição: O comando *Envio de Mensagens* é utilizado para transmitir mensagens codificadas aos terminais na rede.
* Tamanho: 00xx este comando possui tamanho variável, e atualmente o tamanho do campo DATA está limitado a 180 caracteres.
* Índice: 43 (0x2B)
* Estrutura do comando:

>?CCCC2Bdd0Eiitttttttt.......KK

* 2B [2] : Índice do comando;
* dd[2] : número do terminal;
* 0E [2]: sub-comando de grupo de mensagens;
* ii[2] : index de tipo de mensagem;
* ttt.. [variável] : texto codificado da mensagem (máximo de 180 caracteres);
* KK[2]: checksum;
* Resposta : A resposta do comando retornará uma mensagem de OK de foi possível agendar a entrega da mensagem ao terminal, ou uma mensagem de erro caso não seja possível entregar a mensagem.
* Resposta de OK:
  + >!00042B00B9
* Resposta com erro:
  + >!00042BExKK
    - x[1]: Representa o [código do erro](#_Códigos_de_erro), para o caso especifico do erro ‘6’ a mensagem codificada fica na memória e é entregue ao terminal somente quando o mesmo entra na rede.

Exemplo:

>?00422B020E80terminal|T1 Deseja Cancelar?|R|R|T2 1:SIM 2:NAO|T1 |EN10078

* No exemplo acima, o terminal é o de numero “02”.
* O índice é de mensagem dinâmica “80”.
* Mensagem codificada “terminal|T1 Deseja Cancelar?|R|R|T2 1:SIM 2:NAO|T1 |EN100”
* Resposta >!00042B00BC

# REQUISIÇÃO DE RESPOSTAS OU PERGUNTAS DO TERMINAL

* Descrição: O comando *Requisição de Respostas* é utilizado para ler respostas do terminal ou alguma pergunta originada no terminal. Este comando é endereçado a um determinado terminal, ou se for para o endereço zero, se torna uma requisição *broadcast,* retornando os dados do terminal de menor número que possui resposta pendente;
* Tamanho: 04 (0x0004)
* Índice: 44 ( 0x2C)
* Estrutura do comando:

>?CCCC2CnnKK

* + 2C [2]: Índice do comando;
  + nn [2]: número do terminal (00 para *broadcast*);
* Resposta : O comando Requisição retornará até 180 caracteres, conforme demonstração abaixo. No caso de memória vazia, a resposta terá tamanho dois;

Uma pergunta efetuada pelo software gerencial pode não ser respondida, isto é, cancelada pelo usuário, basta que este digite cancelar no teclado do terminal, neste caso a resposta do terminal virá com o campo Mensagem codificada contendo “|G”;

Tamanho: Variável [2..110] ([0x0002 .. 0x006E])

* Estrutura resposta:

>!CCCC2Cdd0Eiitttttttt.......KK

* + 2C [2]: Índice do comando;
  + dd [2] : Número do terminal;
  + 0E [2] : sub-comando;
  + ii [2] : index de tipo de mensagem;
  + ttttt... [x] : Mensagem codificada;

Exemplo:

Resposta vazia:

Comando : >?00042C01D9

Resposta : >!00022C58

Resposta com mensagem:

Comando : >?00042C01D9

Resposta : >!000F2C010E80||||||2C4

* + terminal de número “01”;
  + mensagem do tipo “80”, isto é, resposta a uma pergunta dinâmica do PC ;
  + texto da mensagem “||||||2”;

Resposta com mensagem de cancelamento pelo teclado terminal:

Comando : >?00042C01D9

Resposta : >!000A2C010E80|G68

* + terminal de número “01”;
  + mensagem do tipo “80”, isto é, resposta a uma pergunta dinâmica do PC ;
  + texto da mensagem “|G”;

# LIMPA PENDÊNCIAS DE TERMINAL

* Descrição: O comando *Limpa Pendências* é utilizado para apagar uma mensagem pendente na memória do console, ou seja, confirmando que a última mensagem do terminal foi lida e entendida. Caso este comando não seja executado uma mensagem "sistema não responde" é apresentada no display do terminal. Este comando é endereçado, mas se o endereço for “00” as pendências de todos os terminais serão apagadas.
* Tamanho: 04 (0x0004)
* Índice: 45 (0x2D)
* Estrutura do comando:

>?CCCC2DnnKK

* 2D [2]: Índice do comando;
* nn [2]: número do terminal (00 para *broadcast*);
* Resposta: O comando *Limpa Pendências* retornará sempre a resposta padrão de OK:
  + >!00042D00BB

# LIMPA PENDÊNCIA E ENVIO DE MENSAGENS AO TERMINAL

* Descrição: Este comando é idêntico ao comando *Envio de Mensagens*, porém ele executa automaticamente um comando de *Limpa Pendências* para o terminal o qual esta sendo enviada a nova mensagem. Ele serve para acelerar as comunicações com o terminal quando várias perguntas e respostas são solicitadas em sequência.
* Índice: 46 (0x2E)

# PEDIDO DE STATUS DOS TERMINAIS

* Descrição: O Comando *Pedido de Status* é utilizado para informar quais terminais estão on-line, ou seja quais terminais estão ligados e operando na rede.
* Tamanho: 02 (0x0002)
* Índice: 47 (0x2F)
* Estrutura do comando:

>?CCCC2FKK

* 2F [2]: Índice comando;
* Resposta: O comando *Pedido de Status* retornará os números dos terminais em ordem crescente, que estiverem *on-line*.
* Tamanho: Variável
* Estrutura da resposta:

>!CCCC2Fn1n2.....nxKK

* n1 [2]: número do primeiro terminal;
* n2 [2]: número do segundo terminal;
* ...
* nx [2]: número do ultimo terminal;
* Exemplo:

Nenhum terminal *on-line*.

Comando : >?00022F79

Resposta : >!00022F5B

Somente o terminal 02 *on-line*.

Comando : >?00022F79

Resposta : >!00042F02BF

# GRAVAÇÃO DE IDENTIFICADOR V1.0

* Descrição: O comando de *Gravação de Identificador V1.0* é utilizado para gravar códigos de identificadores na memória da automação, após esse procedimento, os cartões reconhecidos pela automação, podem liberar o bico para abastecimento sem intervenção do computador, sendo assim, as liberações passam a ser controladas pelo console. Este comando possui separadores (‘;’ ponto e vírgula) que devem ser mantidos sempre entre os campos do comando.
* Tamanho: 68 (0x0044)
* Índice: 13 (0x0D)
* Estrutura do comando:
* >?CCCC0DTTTTTTTTTTTTTTTT;LL;10;FFFF;CCCCCCCC;nnnnnnnnnnnnnnnnnnnnnnnnnnnnn;DD,ddKK
  + 0D [2]: Índice comando;
  + TTTTTTTTTTTTTTTT[16]: Código de identificador;
  + L [2]: Controle:
  + 27: Libera bomba para abastecimento;
  + 04: Cliente, não libera bomba para abastecimento;
  + 10[2]: Versão do comando, reservado para uso futuro, manter fixo “10”;
  + FFFF[4]: Flags de função do cartão, reservado para uso futuro, manter fixo “FFFF”;
  + CCCCCCCC[8]: Código do frentista ou cliente no sistema (em caso de não uso manter “FFFFFFFF”);

Este campo será utilizado, por exemplo, para mostrar o código de um cliente no sistema quando o cartão deste é coletado pelo terminal.

* + nnn.....[30]: Campo que deve conter o nome do proprietário do cartão informado, este nome aparecerá na tela de log do terminal, caso possua menos que 30 caracteres deve-se adicionar tantos caracteres “espaço”, 0x20 em hexadecimal, quantos forem necessários para que o campo possua ao final 30 caracteres;
  + DD,dd [5]: Campo desconto em percentual (em caso de não uso manter “00,00”);
* Resposta: Retornará a posição do identificador em memória, ou um código de erro caso não tenha sido possível realizar sua gravação.
* Tamanho: 24 (0x0018) / 4 (0x0004) {Erro}.
* Estrutura resposta:

>!CCCC0DNNNNNNTTTTTTTTTTTTTTTTKK

* + 0D [02]: Índice do tipo do comando;
  + NNNNNN [6]: Posição em que o identificador foi armazenado;
  + T [16]: Código do identificador armazenado;

>!CCCC0DCDKK

* + 0D [02]: Índice do tipo do comando;
  + C [1]: Códigos de Status de execução;
  + D [1]: Código do erro;

Exemplo:

Comando : >?004B0DB3CF6C81BF84AE55;27;10;FFFF;00015341;AIRES AZEVEDO SOUZA ;01,00C0

Resposta : >!00180D000021B3CF6C81BF84AE5545

# ABASTECIMENTO HORUSTECH

* Descrição: O comando de *Abastecimento Horustech* é utilizado para leitura de um abastecimento finalizado. Comando indicado para desenvolvedores que tenham implementado o protocolo Horustech ou estejam iniciando sua integração com nossa automação.
* Tamanho: 02 (0x0002)
* Índice: 02 (0x02)
* Estrutura do comando:
* >?CCCC02KK
  + 02 [2]: Índice comando;
* Resposta: O comando *Abastecimento* retornará até 116 caracteres, conforme demonstração abaixo. No caso de memória vazia a resposta terá tamanho dois.
* Tamanho: Variavel [2...116] ([0x0002 .. 0x0074]).
* Estrutura resposta:

>!CCCC02NNNNNNBBCCAATTTTTTLLLLLLPPPPXYZttttddmmyyhhnneeeeeeeeeeEEEEEEEEEEiiiiiiiiiiiiiiiiIIIIIIIIIIIIIIIIaaaaaaaaKK

* 02 [2]: Índice do comando;
* NNNNNN [6]: Índice do abastecimento na memória da automação;
* BB [2]: Número do bico;
* CC [2]: Código do combustível;
* AA [2]: Número do tanque fornecedor do bico;
* TTTTTT [6]: Valor total abastecido;
* LLLLLL [6]: Volume total fornecido;
* PPPP [4]: Preço unitário praticado;
* X [1]: Número de casas decimais do campo total (T[6]);
* Y[1]: Número de casas decimais do campo volume (L[6]);
* Z[1]: Número de casas decimais do campo preço unitário (P[4]);
* tttt [4]: Tempo de duração do abastecimento;
* ddmmyyhhnn [10]: Data do abastecimento em dia, mês, ano, hora e minuto;
* eeeeeeeeee [10]: Totalizador inicial do abastecimento;
* EEEEEEEEEE [10]: Totalizador final do abastecimento;
* iiiiiiiiiiiiiiii [16]: Identificador 1;
* IIIIIIIIIIIIIIII [16]: Identificador 2: veja tabela 01 deste documento.
* aaaaaaaa [8]: Odômetro/horímetro digitado no terminal.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Caracteres de Controle** | **Caracteres ‘0’ (0x30)** | **I[16]** |
| Tag/Cartão cliente | Não se Aplica | Não se Aplica | IIIIIIIIIIIIIIII |
| CNPJ | CC | Não se Aplica | CCxxxxxxxxxxxxxx |
| CPF | FF | 3 | FF000yyyyyyyyyyy |
| Placa do automóvel | AA | 7 | AA000000kkkkkkk |

Tabela 01 – Formatos do Campo Identificador 2

Os caracteres de controle ocupam os dois primeiros dígitos do campo identificador 2.

O campo Odômetro/horímetro substituiu o campo que era usado para volume do tanque.

Exemplo:

Comando : >?00020263

Resposta : >!006B0200007201000000683402004834092330100220415140201328511370132853142

B3CF6CA5B7ECDAFAFF0000296217808200000000F5

Obs: Para o abastecimento acima foi informado o CPF 029.621.780-82 do cliente (FF00002962178082).

# ABASTECIMENTO COMPANYTEC - CBC

* Descrição: O comando de *Abastecimento Companytec* é utilizado para leitura de um abastecimento finalizado. Comando indicado para desenvolvedores que tenham implementado o protocolo Companytec.
* Tamanho: 02 (0x0002)
* Índice: 162 (0xA2)
* Estrutura do comando:
* (&A2KK)
  + A2 [2]: Índice comando;
  + KK : Checksum, que neste caso é 99;
* Resposta: O comando *Abastecimento Companytec* retornará até 116 caracteres, conforme demonstração abaixo. No caso de memória vazia a resposta será ”(0)”.
* Tamanho: Variavel [1...116] ([0x0001 .. 0x0074]).
* Estrutura resposta:

(a2$SSSSSSSSTTTTTTLLLLLLPPPPVVCCCCBBDDHHMMNNAARRRRRRffffffffffiiiiiiiiiibbccttvvvvvvvvIIIIIIIIIIIIIIIIJJJJJJJJJJJJJJJJnnKK)

* a2 [2]: Índice do comando;
* $ [1]: Caractere de habilitação;
* S [8]: Número de série;
* T [6]: Total a pagar (bombas mecânicas retornam “000000”);
* L [6]: Volume abastecido (Listros ou m³);
* P [4]: Preço unitário (em bombas mecânicas é necessário gravar);
* V [2]: Código de vírgula (Tabela de códigos em DT435 – Protocolo de comunicação Companytec);
* C [4]: Tempo de abastecimento (Hexadecimal);
* B [2]: Código de bico;
* D [2]: Dia;
* H [2]: Hora;
* M[2]: Minuto;
* N [2]: Mês;
* A[2]: Ano;
* R[6]: índice do abastecimento na memória;
* f[10]: Encerrante final (xxxxxxxx,xx);
* i[10]: Encerrante inicial (xxxxxxxx,xx);
* b[2]: Número do bico na pista;
* c[2]: Tipo de combustível;
* t[2]: Número do tanque;
* v[8]: Odômetro/horímetro digitado no terminal.
* I[16]: Identificador 1;
* J[16]: Identificador 2: veja tabela 02 deste documento. .
* N[2]: Status de integridade de memória de abastecimentos (00 = OK);
* K[2]: Cheksum;

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Caracteres de Controle** | **Caracteres ‘0’ (0x30)** | **I[16]** |
| Tag/Cartão cliente | Não se Aplica | Não se Aplica | IIIIIIIIIIIIIIII |
| CNPJ | CC | Não se Aplica | CCxxxxxxxxxxxxxx |
| CPF | FF | 3 | FF000yyyyyyyyyyy |
| Placa do automóvel | AA | 7 | AA000000kkkkkkk |

Tabela 02 – Formatos do Campo Identificador 2

Comando : (&A299)

Resposta : (a2FFFFFFFFF00988302899234093E00110422141304150000730132856041013285314201

000000000000B3CF6CA5B7ECDAFAFF000029621780820061)

# FORMATAÇÃO DAS MENSAGENS CODIFICADAS

* Descrição: As mensagens codificadas são uma combinação de textos para escrever na tela do terminal e comandos que dizem ao terminal como e onde escrever estes textos e quais entradas de dados serão solicitadas ao terminal.

As mensagens codificadas podem ser usadas para armazenar solicitações do tipo *menu* na memória não volátil do terminal ou usadas para fazer solicitações e informes de forma dinâmica e instantânea na tela do terminal. As mensagens do tipo *menu* ficam em memória e são acessadas via tecla *menu* do usuário a qualquer momento. Para diferenciar entre estas mensagens existe o index de tipo na mensagem.

As mensagens são divididas em três partes:

* + Controle da mensagem;
  + Título da mensagem;
  + Corpo da mensagem;

**Controle da Mensagem:** O controle da mensagem é formado por três informações:

* + **Endereço do terminal:** Endereço lógico atribuído a um determinado terminal (configurável).
  + **Sub-comando da mensagem:** atualmente é fixo “0E” e está reservado para uso futuro.
  + **Index de tipo de mensagem:** usado par indicar a forma de uso ou onde armazenar a mensagem.

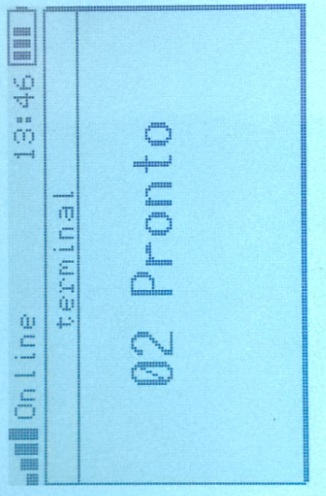
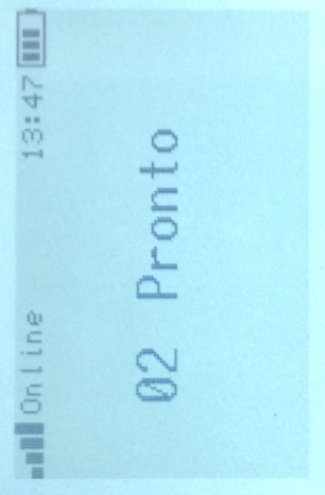
**Título da Mensagem:** O titulo é um texto que pode ser apresentado no topo da tela. Quando este é informado o terminal gera uma caixa no display com o texto na parte de cima. Atualmente limitado a 18 caracteres.

Figura 01 – Mensagem com título Figura 02 – Mensagem sem título

**Corpo da Mensagem:** O corpo da mensagem contém as demais informações iniciando sempre com o caractere de controle ‘|’ (barra vertical), este caractere serve como separador dos demais campos no corpo da mensagem. A mensagem completa pode ter quantos campos forem possíveis dentro do limite máximo de 180 caracteres.

Cada campo pode possuir somente uma função, cuja lista está na tabela 1 abaixo:

|  |  |
| --- | --- |
| **CÓDIGO** | **FUNÇÃO** |
| **Tx**   * x = ‘1’ Letra tamanho 1 * x = ‘2’ Letra tamanho 2 | Exibe texto |
| **Eyxxzn...**   * y = ‘A’ Alfa numérico * y = ‘S’ Alfa * y = ‘C’ CPF/CNPJ * y = ‘N’ Numérico * y = ‘M’ Mascarado * xx = Número total de caracteres * z = Número de dígitos após a vírgula * zn... = Máscara   + zn = ‘K’ Imprime ‘\*’.   + zn = outros caracteres (- , /) informados serão fixos na posição.   Exemplos:     |  |  |  |  |  | | --- | --- | --- | --- | --- | | Comando | EN030 | EN031 | EC | ES02 | | Terminal | 000 | 00.0 | 00002962178083 | AB | | Comando | EM02KK | EA02 | EM08SSS-NNNN | EN020 | | Terminal | \*\* | A2 | CSC-2013 | 74 |  |  | | --- | | Comando : >?00302B010E81Terminal|D0025|T1Insira o CPF ...|R|R|ECB1 | | Resposta : >!001B2C010E81|||||00002962178083E0 |  |  | | --- | | Comando : >?00382B010E81Terminal|D0025|T1Digite a Senha (3):|R|R|EM03NKK36 | | Resposta : >!00102C010E81|||||555A1 | | Entrada de digitação  **Entrada CPF/CNPJ,** nestecaso a validade do número informado pelo usuário será averiguada pelo terminal, a entrada solicitada é fixa de 14 caracteres numéricos, portanto não é necessário informar o campo “xx”.  **Entrada Numérica**, neste caso o parâmetro z deve ser informado com a quantidade de dígitos após a vírgula esperado e o parâmetro ‘n’ não informado.  **Entrada Mascarada**, os parâmetros ‘zn..’ devem ser informados conforme o tipo de máscara, isto é, para cada caractere digitado um caractere de máscara correspondente deve ser informado.  **Máscara para digitação de senhas**, para esta aplicação em que se deseja que o usuário não visualize na tela os caracteres digitados no teclado, isto é, os caracteres são substituídos por ‘\*’, deverá ser informado o caractere ‘K’ como parâmetro da máscara. |
| **C** | Leitura de cartão RFID |
| **B** | Leitura de código de barras (opcional) |
| **R** | Retorno de linha ou nova linha |
| **A**  Estrutura da Resposta: **nk.kk**   * **n** = Nível da bateria (0 a 4) * **k.kk** =Tensão da bateria (2 casas decimais)   Ex:   |  | | --- | | Comando : >?002B2B010E81Terminal|T1Pressione Enter ...|A|KC4A | | Resposta : >!00112C010E81||**13.37**|K52 | | Retorna Nível e Tensão da bateria  O nível da bateria corresponde ao VU do dispositivo, isto é, nível de bateria 1 corresponde a 1 barra do indicador.  Recomenda-se a recarga do equipamento quando o nível de bateria for inferior a 2. |
| **L** | Retorna código do cartão logado no terminal |
| **N** | Retorna numero de série do terminal |
| **V** | Retorna Versão do firmware do terminal |
| **F** | Limpa o display |
| **Jnn**   * n[2] = Número do menu a ser executado (20,21,29). | Execução de Menu |
| **Intttt**   * n = Quantidade de beeps (1 a 9) * t[4] = Intervalo entre os beeps (0 a 9999 milisegundos) | Beep  O tempo de beep é o mesmo de intervalo entre os beeps. |
| **Hx**   * x = ‘1’ Letra tamanho 1 * x = ‘2’ Letra tamanho 2 | Exibe número lógico do terminal |
| **G** | Cancelamento de Operação  Sempre que for solicitada uma entrada para o usuário e este pressionar ‘X’ o software gerencial receberá o comando ‘G’ indicando o cancelamento de sua solicitação. |
| **Mxttttt...**   * x = ‘G’ grava versão do *menu* * x = ‘L’ retorna leitura da versão * tttt... = texto com exatos 20 caracteres identificando a versão do *menu.* | Versão do *Menu*  O sistema gestor pode adotar o formato que quiser com tamanho fixo de 20 caracteres. Para leitura não informar o campo “ttttt...” |
| **Kx**   * x = ’C’ aguarda *enter* com confirmação * x = ‘N’ aguarda *enter* sem confirmação | Aguarda a digitação da tecla “*enter*”  A confirmação gera uma resposta ‘K’ ao PC |
| **Dxxyy**   * xx = posição do display horizontal * yy = posição display vertical | Posiciona a entrada de dados no display  Os valores são de 0 a 99 (relativo) . |

Tabela 02 - Funções

OBS: Novas funções podem ser criadas se necessário.

**Index de tipo:** o campo index de tipo é usado para informar ao terminal o tipo da mensagem enviada, atualmente existem dois tipos:

* **Menu**

Valores de 20 a 2B representam cada *menu* na tela. Ou seja, para gravar o *menu* 2.1 o index é “21” e assim por diante. Este index deve ser usado na hora de gravar o texto codificado no terminal e deve ser comparado para saber de qual *menu* se trata uma determinada requisição do terminal, para este index de tipo o campo de título deve ser obrigatoriamente informado.

* **Dinâmico**

Valores de 0x80 a 0x9F podem ser usados para identificar uma pergunta dinâmica, facilitando assim verificar de qual pergunta a resposta se trata. Como o desenvolvedor vai criar suas próprias perguntas ele pode criar um index para se organizar na hora de receber as respostas sem precisar saber o que foi perguntado antes para o terminal. Este parâmetro aceita valores hexadecimal.

* **Menus Especiais**

O *menu* “2A" se configurado no terminal, é chamado quando a tecla “veículo” é pressionada.

O *menu* “2B” se configurado no terminal, é chamado quando a tecla “bomba” é pressionada.

O *menu* “2C” se configurado, representará a tela inicial do terminal substituindo a tela padrão, para este *menu (tela)* deve ser fornecido somente o comando de *exibe texto*.

O *menu* “2D” se configurado, substituirá a mensagem padrão “Aguarda PC”, um comando *exibe texto* com o texto desejado deve ser fornecido para este menu.

Estes menus devem ser habilitados no terminal, através do *menu* *sistema*, para mais informações ver DT 473 – Manual de Instalação TWC01.

**Exemplos de gravação de menu:**

Tx: >?003E2B010E2CTela Inicial Teste|D1820|T2Companytec|D2250|T2SoluçõesDC

Rx: >!00042B00B9

Tx: >?003E2B020E2AAbastecer|D0030|T1Leia a Tag do Veiculo ...|D1860|C|KC07

Rx: >!00042B00B9

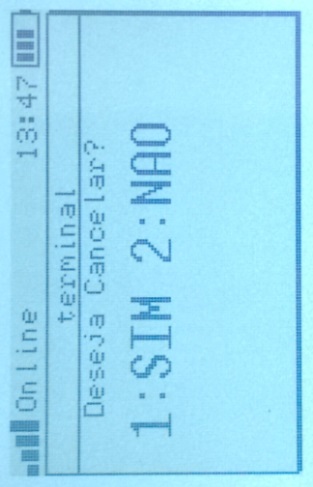
Tx: >?003D2B020E2BConsulta Cartoes|D0030|T1Leia o Cartao ...|D1860|C|KC19

Rx: >!00042B00B9

Tx: >?004D2B020E81Terminal|T1Erro!|R|R|T1Sistema Offline|R|R|T1Procure o Gerente ...|KNCC

Rx: >!00042B00B9

**Exemplos de texto codificado:**

Tx: >?00422B020E80terminal|T1 Deseja Cancelar?|R|R|T2 1:SIM 2:NAO |R|EN01043

Rx: >!00042B00B9

* Texto enviado ao terminal 2 Figura 03 -
* Index dinâmico “80” Texto Titulo: “terminal” codificado
* Texto em tamanho 1 na primeira linha com

do display “ Deseja Cancelar?” entrada

* Avance duas linhas “|R|R” numérica
* Texto na quarta linha em tamanho 2

“ 1:SIM 2:NAO ”

* Espera digitação de um caractere numérico na quinta linha sem casas após a vírgula “|EN010”

Resposta:

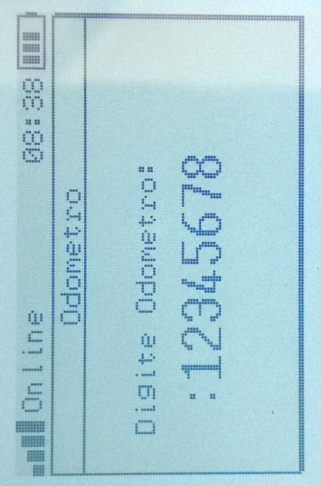
Tx: >?00042C00D8 (*requisição de respostas ou perguntas do terminal*)

Rx: >!000F2C020E80**||||||2**C5

Observe que para cada campo retorna uma reposta, mesmo que esta resposta seja vazia. Neste exemplo somente o quinto campo tinha retorno, que neste caso foi digitado a tecla <2> e a tecla <*enter>*.

Tx: >?00392B020E81Odometro|R|T1 Digite Odometro:|R|R|T2 :|EN080|VA1

Rx: >!00042B00B9



* Texto enviado ao terminal 2;
* Index dinâmico “81”;
* Titulo: “Odometro”;
* Texto na segunda linha do display “ Digite Odometro:”;
* Avance uma linha “|R”;
* Texto tamanho 2 na quinta linha “ :”;
* Espera digitação de 8 caracteres numéricos e nenhum após a vírgula “|EN080”;
* Retorna versão do firmware do terminal “|V”.

Figura 04 – Texto codificado com solicitação de odômetro

Resposta:

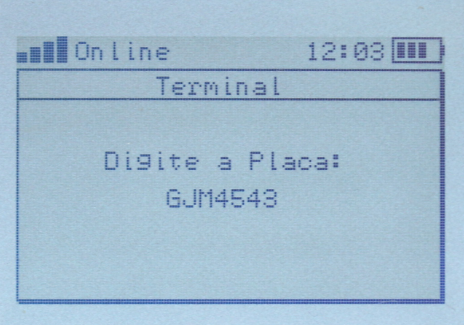
Tx: >?00042C00D8 (*requisição de respostas ou perguntas do terminal*)

Rx: >!00192C020E81||||||12345678|1009

* Seis barras retornam;
* Resposta ao pedido de digitação de numérico “|12345678”.
* Resposta ao pedido de versão de firmware “|10” equivale a “V 1.0”.

Tx: >?003A2B020E80Terminal|D2025|T1Digite a Placa:|D3545|EM07SSSNNNND5

Rx: >!00042B00B9



* Texto enviado ao terminal 2;
* Index dinâmico “80”;
* Título: “Terminal”;
* Posição relativa do cursor para a escrita: “(x,y) = (20,25)”;
* Texto tamanho 1: “Digite a Placa:”
* Posição relativa do cursor para a entrada: “(x,y) = (35,45)”;
* Solicita uma entrada de 7 caracteres com a máscara: ”SSSNNNN”, isto é, os 3 primeiros caracteres devem ser alfa e os próximos 4 deverão ser obrigatoriamente numéricos.

Resposta: Figura 05 – Solicitação de entrada mascarada

Tx: >?00042C00D8 (*requisição de respostas ou perguntas do terminal*)

Rx: >!00132C010E80||||GJM454336

Placa digitada pelo usuário: “GJM4543”

# Códigos de erro

Descrição: Os códigos abaixo são utilizados por todas as respostas do protocolo que possam conter um erro, esses códigos são numerados de 0 a 16 (0x00 a 0x0F):

* 0: Sucesso
* 1: Número de bico posição A já utilizado/ buffer do terminal ocupado;
* 2: Número de bico posição B já utilizado;
* 3: Número de bico posição C já utilizado;
* 4: Número de bico posição D já utilizado;
* 5: Erro de resposta da bomba/ pendência existente;
* 6: Erro de timeout da bomba/ terminal offline;
* 7: Número de bico inexistente/ terminal inexistente;
* 8: Bico abastecendo;
* 9: Modo inválido;
* A: Identificador diferente/ terminal não inicializado;
* B: Erro ao apagar identificador;
* C: Erro parâmetro inválido;
* D: Erro Certificado inválido;
* E: Erro Comando inválido;

# Códigos de Status de execução

Descrição: Os códigos abaixo são utilizados por todas as respostas do protocolo que possam conter erros, informando a situação da execução do comando:

* 0: Sucesso;
* E: Erro;

# 12. Cálculo de Check-Sum

Descrição: Somatório em Hexa-Decimal dispostos em dois caracteres adicionados às linhas de comando e resposta do equipamento, esse somatório garante que os dados estejam íntegros ao serem recebidos nas extremidades de comunicação (PC ⇔ Automação).

O cálculo do Check-Sum HRS-1, é realizado com um acumulador, onde são acrescidos os valores ASCII de cada caractere presente após o caractere de início de bloco nas linhas de comando e resposta.

Para manter um tamanho fixo de dois caracteres, as casas mais significativas são removidas do resultado do somatório, até que seu tamanho pré-estabelecido seja respeitado.

Exemplo

>!000A01AALB P A34

CONTROLE =!

CONTADOR = 000A

DATA = 01AALB P A

CHECK-SUM = 34

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Caractere | ASCII (Dec) | ASCII (Hexa) | Acum. (Dec) | Acum. (Hexa) |
| ! | 33 | 21 | 33 | 021 |
| 0 | 48 | 30 | 81 | 051 |
| 0 | 48 | 30 | 128 | 81 |
| 0 | 48 | 30 | 177 | 0B1 |
| A | 65 | 41 | 242 | 0F2 |
| 0 | 48 | 30 | 290 | 122 |
| 1 | 49 | 31 | 339 | 153 |
| A | 65 | 41 | 404 | 194 |
| A | 65 | 41 | 469 | 1D5 |
| L | 76 | 4C | 545 | 221 |
| B | 66 | 42 | 611 | 263 |
| <SPACE> | 32 | 20 | 643 | 283 |
| P | 80 | 50 | 723 | 2D3 |
| <SPACE> | 32 | 20 | 755 | 2F3 |
| A | 65 | 41 | 820 | 334 |

Tabela 03 - Relação para Caracteres