

A comunicação entre a "família" de Terminais e Coletores TED/PAT_UDP (denominados a partir de agora unicamente como TED) e o HOST é realizada via protocolo **UDP**, utilizando o modelo **cliente/servidor**, onde o TED é o cliente UDP e o HOST (PC) é o servidor UDP.

- Tanto o HOST como o TED podem enviar comandos um para o outro. No caso do TED, o dispositivo pode enviar para o HOST texto inserido pelo usuário via teclado do TED, ou dados enviados por dispositivos conectados ao TED, como leitor de código de barras externo tipo pistola, leitor de código de barras embutido internamente no TED, balança e sensores digitais. diversos (medidores, contadores, etc.)

Para que cada parte (TED e HOST) possa encontrar o IP do outro nó na rede, é preciso ser estabelecido um processo de conexão.

Esse processo pode ser dar através de endereço de IP fixo configurado pelo usuário, ou através de IP dinâmico (automático) concedido pelo roteador / switch, iniciado pelo TED, detalhado abaixo:

- Ao ser ligado, o TED começa a enviar um pacote de 4 bytes em broadcast a cada 2 segundos na **porta 55555** via socket UDP:
 - 0x00 0x00 0x00 0x00 (hexadecimal)
 - O HOST, que está rodando o software + DLL, recebe esse pacote (o broadcast chega em todos os dispositivos da rede). Então, a aplicação no HOST analisa o pacote e descobre o IP de origem, ou seja, o IP do TED;
 - Em seguida, o HOST envia o texto 'Conectado' num pacote para o TED na **porta 8** via protocolo UDP:
 - 0x20 0x00 0x00 0x09 0x43 0x67 0x6E 0x65 0x63 0x74 0x61 0x64 0x67 (hexadecimal)
 - Desta forma, o TED descobre o IP do HOST ao receber esse pacote e se conecta logicamente a ele. Daí em diante, o TED usará a porta 8 em UDP para se comunicar com o HOST.
- 0xFF, após o qual deve retornar a 0x00;
 - **Tamanho do campo DADOS:** número de bytes do campo DADOS inserido no pacote. Caso não exista dado a ser enviado, o seu valor deve ser 0x00. Este campo limita o tamanho máximo de dados. Por ser somente um byte, seu limite é 0xFF.
 - **DADOS:** dados referentes ao comando, limitado em 255 bytes, visto o limite imposto

Protocolo

O protocolo entre HOST e TED é definido em comandos e respostas.

Comando

Cada comando deve seguir o seguinte formato:

<i>Pacote</i>	ID do comando	Contador de tentativas	Contador de comandos	Tamanho do campo DADOS	DADOS
<i>Tamanho em bytes</i>	1	1	1	1	Até 255 bytes

- **ID do comando:** define o código do comando que deve ser executado, seja no HOST, seja no TED;
- **Contador de tentativas:** contador iniciado em 0x00 e incrementado em cada tentativa de transmissão de qualquer pacote para o outro nó. O número máximo de tentativas para envio de um comando é 3. Portanto, esse valor pode variar de 0x00 a 0x02;
- **Contador de comandos:** contador incrementado a cada comando enviado para o outro nó, independente do outro contador das tentativas. Para as n tentativas de envio de um comando, deve ser usado o mesmo contador de comandos, o qual deve ser incrementado somente no envio de um novo comando. Seu valor máximo é pelo campo anterior, de tamanho.

Resposta

Cada resposta deve seguir o seguinte formato:

<i>Pacote</i>	ID do comando (0x80)	Contador de tentativas	Contador de comandos	Tamanho do campo DADOS	DADOS
<i>Tamanho em bytes</i>	1	1	1	1	Até 255 bytes

- **ID do comando:** deve ser o valor fixo de 0x80, que indica resposta a qualquer comando;
- **Contador de tentativas:** valor do contador de tentativas recebido no pacote do comando que está sendo respondido;
- **Contador de comandos:** valor do contador de comandos recebido no pacote do comando que está sendo respondido;
- **Tamanho do campo DADOS:** número de bytes do campo DADOS inserido no pacote. Caso não exista dado a ser enviado, o seu valor deve ser 0x00. Este campo limita o tamanho máximo de dados. Por ser somente um byte, seu limite é 0xFF.

- **DADOS:** dados referentes à resposta, limitado em 255 bytes, visto o limite imposto pelo campo anterior, de tamanho.

Exemplo

Caso 1: TED recebe resposta do HOST na primeira tentativa

O TED envia para o HOST o pacote acima e obtém resposta. O pacote de resposta deve ser:

HOST -> TED: 0x80 0x00 0x22 0x00

Caso 2: TED não recebe resposta do HOST e nenhuma tentativa

O TED envia o pacote acima para o HOST, mas não obtém resposta. O HOST pode estar desligado, ou a aplicação pode estar fechada. Então o TED tentará transmitir mais duas vezes, resultando nos seguintes pacotes enviados ao HOST:

TED -> HOST: 0x01 0x00 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41

TED -> HOST: 0x01 0x01 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41

TED -> HOST: 0x01 0x02 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41

Comando enviado pelo TED ao HOST:

<i>Pacote</i>	ID do comando	Contador de tentativas	Contador de comandos	Tamanho do campo DADOS	DADOS
<i>Valor</i>	0x01	0x00	0x22	0x06	"BANANA"

- **ID do comando:** 0x01 (texto enviado do TED ao HOST);
- **Contador de tentativas:** iniciado em 0x00 e incrementado até 0x02 em cada tentativa de transmissão do pacote;
- **Contador de comandos:** dado o número de comandos enviado pelo TED até o momento, o comando atual tem o contador com o valor 0x22;
- **Tamanho do campo dados:** Como será enviado o texto "BANANA", que possui 6 caracteres, esse valor deve ser setado neste campo;
- **DADOS:** o texto "BANANA"

O contador de tentativas (segundo byte) é incrementado a cada tentativa (0x00, 0x01 e 0x02).

Caso 3: HOST perde a primeira tentativa de comando

O TED tenta transmitir o comando, mas por algum problema de transmissão (colisão na rede, ruído ou outro motivo) a primeira tentativa não chega ao HOST. Mas o pacote enviado na segunda tentativa chega normalmente. Vai acontecer então o seguinte tráfego de dados:

```
TED -> HOST: 0x01 0x00 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41
TED -> HOST: 0x01 0x01 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41
HOST -> TED: 0x80 0x01 0x22 0x00
```

Neste exemplo, o HOST perdeu o primeiro pacote por algum problema de ruído. Mas quando o TED transmitiu o segundo pacote, o HOST respondeu. Ao receber esse pacote, o TED pára de transmitir novas tentativas porque percebe que o HOST já recebeu o dado corretamente, já que contador de tentativas é 0x01 e coincidiu com o mesmo contador de tentativas do comando enviado.

Caso 4: TED perde a primeira tentativa de resposta

Agora suponha que, por algum motivo, a resposta do HOST (confirmação de recebimento) não chegue ao TED (por ruído, colisão ou outro motivo). O TED vai interpretar que o HOST ainda não recebeu o dado e o transmitirá novamente. O HOST, ao receber o comando repetido, dado que o seu contador de comandos 0x22 é idêntico ao do comando anterior, vai desprezá-lo porque já o processou. Se fosse realmente um novo comando, o TED teria incrementado este byte para 0x23. Esse contador é um número incrementado não a cada tentativa de transmissão do mesmo comando, mas a cada novo comando que o TED tenta transmitir.

Vai acontecer então o seguinte tráfego de dados:

```
TED -> HOST: 0x01 0x00 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41
HOST -> TED: 0x80 0x00 0x22 0x00 (perdido)
TED -> HOST: 0x01 0x01 0x22 0x06 0x42 0x41 0x4E 0x41 0x4E 0x41
HOST -> TED: 0x80 0x01 0x22 0x00
```

Este protocolo de confirmação de envio e confirmação de recebimento acontece nos dois sentidos: do TED para o HOST e do HOST para o TED e em todos os comandos implementados.

Comandos enviados pelo TED ao HOST

Obs.: Os modelos TED fixos possuem 2 portas seriais RS-232C + 1 Entrada Digital + 1 Saída Digital em seu conector DB-09 na parte traseira, para conexões externas.

Comando de envio de dados do TED (0x01 a 0x06)

O TED envia um comando ao HOST para indicar que algum novo dado foi disponibilizado pelo TED, como um texto digitado pelo usuário, um código de barra disponibilizado pela leitora de código de barras, entrada de dados externos pelas seriais RS232, entrada de sinal digital, etc.

Nesse comando pode ser inserida ou não a informação da origem do dado, através de headers. Por padrão o envio dessa informação não está habilitado. Para habilitar os headers é preciso enviar um comando ao TED previamente.

Caso os headers não estejam habilitados (padrão), o TED envia somente o comando **0x01** e o dado disponibilizado, não indicando a sua origem. Caso contrário, o TED envia os comandos **0x01 a 0x06**, os quais indicam a origem do dado. Veja a seguir o que cada comando indica.

Header desabilitado (padrão)

Somente o comando 0x01 é enviado pelo TED, independente da origem.

Comando 0x01 - Texto digitado pelo usuário (“BANANA”):

```
TED -> HOST: 0x01 0x00 0x22 0x07 0x42 0x41 0x4E 0x41 0x4E 0x41
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x01 - Leitura de código de barra com comunicação USB (“7891040042517”):

```
TED -> HOST: 0x01 0x00 0x22 0x0E 0x37 0x38 0x39 0x31 0x30 0x34
0x30 0x30 0x34 0x32 0x35 0x31 0x37 0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x01 - Leitura de código de barra com comunicação serial (“7891040042517”):

```
TED -> HOST: 0x01 0x00 0x22 0x0E 0x37 0x38 0x39 0x31 0x30 0x34
0x30 0x30 0x34 0x32 0x35 0x31 0x37 0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x01 - Leitura da porta serial 1 (“123456”):

Obs.: Nesta porta serial pode estar conectada uma balança, que envia o peso no momento da pesagem, por exemplo.

Parâmetros da comunicação serial: default 9600,n,8,1

```
TED -> HOST: 0x01 0x00 0x22 0x07 0x31 0x32 0x33 0x34 0x35 0x36
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x01 - Leitura da porta serial 2 (“123456”):

Obs.: Nesta porta serial pode estar conectado um paquímetro digital p.explo., que envia a medida no momento da medição , por exemplo.

Parâmetros da comunicação serial: default 9600,n,8,1.

```
TED -> HOST: 0x01 0x00 0x22 0x07 0x31 0x32 0x33 0x34 0x35 0x36
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Header habilitado

Com o header habilitado, o campo “ID do comando” pode assumir mais de um valor dependendo da origem do dado.

Comando 0x01 - Texto digitado pelo usuário (“BANANA”):

```
TED -> HOST: 0x01 0x00 0x22 0x07 0x42 0x41 0x4E 0x41 0x4E 0x41
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x02 - Leitura de código de barra com comunicação USB (“7891040042517”):

```
TED -> HOST: 0x02 0x00 0x22 0x0E 0x37 0x38 0x39 0x31 0x30 0x34
0x30 0x30 0x34 0x32 0x35 0x31 0x37 0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x03 - Leitura de código de barra com comunicação serial (“7891040042517”):

```
TED -> HOST: 0x03 0x00 0x22 0x0E 0x37 0x38 0x39 0x31 0x30 0x34
0x30 0x30 0x34 0x32 0x35 0x31 0x37 0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x04 - Leitura da porta serial 1 (“123456”):

Obs.: Nesta porta serial por estar conectada uma balança, que envia o peso no momento da pesagem, por exemplo.

Parâmetros da comunicação serial: default 9600,n,8,1.

```
TED -> HOST: 0x04 0x00 0x22 0x07 0x31 0x32 0x33 0x34 0x35 0x36
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comando 0x05 - Leitura da porta serial 2 (“123456”):

Obs.: Nesta porta serial por estar conectada um paquímetro digital, que envia a medida no momento da medição, por exemplo.

Parâmetros da comunicação serial: default 9600,n,8,1

```
TED -> HOST: 0x05 0x00 0x22 0x07 0x31 0x32 0x33 0x34 0x35 0x36
0x0D
HOST -> TED: 0x80 0x00 0x22 0x00
```

Comandos enviados pelo HOST ao TED

Obs.: Os modelos TED fixos possuem 2 portas seriais RS-232C + 1 Entrada Digital + 1 Saída Digital em seu conector DB-09 na parte traseira, para conexões externas.

Comando de escrita de mensagem no display do TED (0x01)

O HOST envia uma mensagem disponibilizada no display TED. Exemplo do envio do texto “ABC”:

```
HOST -> TED: 0x01 0x00 0x11 0x03 0x41 0x42 0x43
TED -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é variável, dependendo do tamanho do texto.

Comando para tocar beep de inicialização (0x02)

É tocado no TED um beep de inicialização. Esse padrão de beep é fixo no TED.

```
HOST -> TED: 0x02 0x00 0x11 0x00
TED -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre zero.

Comando para apagar display (0x03)

O conteúdo exibido no display do TED é apagado por completo.

```
HOST -> TED:  0x03 0x00 0x11 0x00
```

```
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre zero.

Comando para tocar "n" beeps (buzzer) (0x05)

São tocados no TED *n* beeps, onde *n* é passado no único byte do campo DADOS. No exemplo abaixo o comando faz tocar 4 beeps:

```
HOST -> TED:  0x05 0x00 0x11 0x01 0x04
```

```
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre um.

Comando para escrita na porta serial auxiliar 1 (0x06)

Esse comando é usado para transmissão de dados para dispositivos externos via porta serial usando o conector DB-09.

Parâmetros da comunicação serial: default 9600,n,8,1.

No exemplo abaixo é enviado pelo HOST um comando para envio do texto "ABC" na porta serial auxiliar 1 do TED, onde pode estar conectada uma impressora, por exemplo:

```
HOST -> TED:  0x06 0x00 0x11 0x03 0x41 0x42 0x43
```

```
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é variável, dependendo do tamanho do texto.

Comando para escrita na porta serial auxiliar 2 (0x07)

Esse comando é usado para transmissão de dados para dispositivos externos via porta serial usando o conector DB-09.

Parâmetros da comunicação serial: default 9600,n,8,1.

No exemplo abaixo é enviado pelo HOST um comando para envio do texto "ABC" na porta serial auxiliar 2 do TED, onde pode estar conectado a um display informativo de leds , por exemplo:

```
HOST -> TED:  0x07 0x00 0x11 0x03 0x41 0x42 0x43
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é variável, dependendo do tamanho do texto.

Comando para habilitar/desabilitar leitura da serial auxiliar 1 (0x08)

Esse comando é usado para habilitar ou desabilitar a leitura dos dados da porta serial 1.

No exemplo abaixo é enviado pelo HOST um comando para habilitar a leitura (valor 1):

```
HOST -> TED:  0x08 0x00 0x11 0x01 0x01
TED  -> HOST: 0x88 0x00 0x11 0x00
```

No exemplo abaixo é enviado pelo HOST um comando para desabilitar a leitura (valor 0):

```
HOST -> TED:  0x08 0x00 0x11 0x01 0x00
TED  -> HOST: 0x88 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre um.

Comando para habilitar/desabilitar leitura da serial auxiliar 2 (0x09)

Esse comando é usado para habilitar ou desabilitar a leitura dos dados da porta serial 2.

No exemplo abaixo é enviado pelo HOST um comando para habilitar a leitura (valor 1):

```
HOST -> TED:  0x09 0x00 0x11 0x01 0x01
TED  -> HOST: 0x89 0x00 0x11 0x00
```

No exemplo abaixo é enviado pelo HOST um comando para desabilitar a leitura (valor 0):

```
HOST -> TED:  0x09 0x00 0x11 0x01 0x00
TED  -> HOST: 0x89 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre um.

Comando para ler a entrada digital (0x0D)

Esse comando é usado para ler o estado da entrada digital, podendo ser 0 ou 1.

No exemplo abaixo é enviado pelo HOST um comando para ler o valor da entrada digital:

```
HOST -> TED:  0x0D 0x00 0x11 0x00
TED  -> HOST: 0x80 0x00 0x11 0x02 0x00 0x0D (valor 0)
ou
TED  -> HOST: 0x80 0x00 0x11 0x02 0x01 0x0D (valor 1)
```

O tamanho do campo DADOS é sempre dois, pois contém o valor da entrada digital e o caractere 0x0D.

Comando para ligar a saída digital (0x0E)

Esse comando é usado para ligar a saída digital, ajustando o seu nível lógico para 1.

No exemplo abaixo é enviado pelo HOST um comando para ajustar o valor 1 na saída digital:

```
HOST -> TED:  0x0E 0x00 0x11 0x00
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre zero.

Comando para desligar a saída digital (0x0F)

Esse comando é usado para desligar a saída digital, ajustando o seu nível lógico para 0.

No exemplo abaixo é enviado pelo HOST um comando para ajustar o valor 0 na saída digital:

```
HOST -> TED:  0x0F 0x00 0x11 0x00
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre zero.

Comando para limpar a lista de atalhos (0x11)

Esse comando é usado para limpar o conteúdo da lista de atalhos apresentada para o usuário no TED.

Todos os atalhos do TED são apagados. Dessa forma, não é exibido o menu de atalhos quando pressionadas as teclas de entrada do menu. Assim que um comando de adicionar página (0x12) for enviado, o menu é exibido novamente quando requisitado.

```
HOST -> TED:  0x11 0x00 0x11 0x00
TED  -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre zero.

Comando para adicionar uma página à lista de atalhos (0x12)

Adiciona uma página no menu de atalhos atual, limitado a até 4 páginas no total. Cada página pode conter até 7 atalhos de 15 caracteres (19 caracteres por linha menos 4 usados para exibir o índice do item no display).

Cada linha de 19 caracteres é exibida no display no seguinte formato:

```
(X) 123456789012345
```

Para enviar um comando para adicionar uma página, no campo DADO deve conter até 7 itens da página de forma sequencial. Cada item deve conter 16 bytes (15 bytes de texto + 1 caractere NULL), o tamanho da string referente à linha. Mesmo que o item tenha menos que 15 caracteres (como "BANANA", por exemplo), os bytes seguintes devem ser preenchidos com 0x00 até completar os 15 bytes. O item "BANANA", por exemplo, deve ser escrito assim:

```
0x42 0x41 0x4E 0x41 0x4E 0x41 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00
```

São inseridos 9 bytes 0x00 de padding mais 1 byte 0x00 para terminação de string. No total, 10 bytes 0x00.

Veja um exemplo de comando para adicionar uma página com os seguintes itens:

```
NOMEDOITEM_1
NOMEDOITEM_2
NOMEDOITEM_3
NOMEDOITEM_4
NOMEDOITEM_5
NOMEDOITEM_6
NOMEDOITEM_7
```

Comando:

HOST -> TED:

```
0x12 0x00 0x11 0x70
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x11 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x12 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x13 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x14 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x15 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x16 0x00
0x00 0x00 0x00
0x4E 0x4F 0x4D 0x45 0x44 0x4F 0x49 0x54 0x45 0x4D 0x5F 0x17 0x00
0x00 0x00 0x00
```

TED -> HOST: 0x80 0x00 0x11 0x00

Cada item possui 12 caracteres, então foram necessários 4 bytes 0x00 de padding por item. O tamanho total do campo de dados é 112 bytes (0x70).

Para se configurar um conjunto novo de 3 páginas, por exemplo, pode ser executada a seguinte sequência:

- enviar comando "limpar lista atual" (0x11);
- enviar comando "adicionar página" (0x12) com o conteúdo da primeira página;
- enviar comando "adicionar página" (0x12) com o conteúdo da segunda página;
- enviar comando "adicionar página" (0x12) com o conteúdo da terceira página.

Comando para habilitar headers (0x13)

Esse comando é usado para habilitar ou desabilitar o envio de headers para o HOST.

No exemplo abaixo é enviado pelo HOST um comando para habilitar os headers (valor 1):

```
HOST -> TED: 0x13 0x00 0x11 0x01 0x01
TED -> HOST: 0x80 0x00 0x11 0x00
```

No exemplo abaixo é enviado pelo HOST um comando para desabilitar os headers (valor 0):

```
HOST -> TED: 0x13 0x00 0x11 0x01 0x00
TED -> HOST: 0x80 0x00 0x11 0x00
```

O tamanho do campo DADOS é sempre um.