

## Descrição geral do TED

A comunicação entre o TED e o host é realizada utilizando o protocolo de transporte TCP/IP, estabelecendo uma arquitetura cliente-servidor. O host, no papel de cliente TCP, pode enviar comandos para o TED, como, por exemplo, escrever texto no display, limpar display, etc. Já o TED, sendo o cliente da comunicação, envia dados de entrada (digitação pelo usuário, leitura de código de barras e outros) para o host na forma de requisição TCP.

A comunicação é ponto-a-ponto, ou seja, cada um se comunica diretamente com o outro. Para que um identifique a outra ponta na rede, o TED envia periodicamente um stream em broadcast via protocolo **UDP**, o qual deve ser respondido pelo host. Dessa forma, ambos tomam conhecimento de seus IPs na rede para poderem enviar comandos um para o outro.

## Descrição do protocolo UDP usado para conexão

Ao ser ligado, o TED começa a enviar um pacote de 1 byte (0x00) em broadcast a cada 10 segundos na **porta 55555** via socket UDP.

O HOST, que está escutando a porta 55555 usando um socket UDP, recebe esse pacote (o broadcast chega em todos os dispositivos da rede). Então, a aplicação no HOST analisa os dados recebidos e identifica o IP de origem, ou seja, o IP do TED.

Em seguida, o HOST deve enviar o texto **'Conectado'** num pacote para o TED na porta 55555 via protocolo UDP. 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 este IP para envio de dados ao host usando o protocolo TCP.

## Descrição do protocolo TCP do TED

O TED executa internamente um servidor TCP. Todos os dados utilizados nos pacotes TCP, seja um comando, seja uma resposta, são formatados no padrão JSON.

Veja abaixo alguns exemplos de comandos suportados pelo TED, que podem ser enviados pelo host, e suas respectivas respostas.

Obs.: O IP do TED é 192.168.10.2 e o IP do host é 192.168.10.3 nos exemplos.

## Comandos enviados pelo HOST ao TED

### Comando de escrita de mensagem no display do TED

Comando do host para o TED:

```
Body: {"cmd": 1, "args": {"text": "olá TED!"} }
```

Resposta do TED para o host:

```
Body: {}
```

### Comando para tocar beep de inicialização

Comando do host para o TED:

```
Body: {"cmd": 2}
```

Resposta do TED para o host:

```
Body: {}
```

### Comando para apagar display

Comando do host para o TED:

```
Body: {"cmd": 3}
```

Resposta do TED para o host:

```
Body: {}
```

### Comando para tocar "n" beeps (buzzer)

Comando do host para o TED:

```
Body: {"cmd": 5, "args": {"qtty": 1} }
```

Resposta do TED para o host:

```
Body: {}
```

O membro "qtty" contém a quantidade desejada de beeps.

### Comando para escrita na porta serial auxiliar COM1

Comando do host para o TED:

```
Body: { "cmd": 6, "args": { "data": "ABCDEF12345" } }
```

Resposta do TED para o host:

```
Body: {}
```

O texto enviado está contido no campo “data” e deve conter caracteres ASCII printáveis somente.

## Comando para escrita na porta serial auxiliar COM2

Comando do host para o TED:

```
Body: { "cmd": 7, "args": { "data": "ABCDEF12345" } }
```

Resposta do TED para o host:

```
Body: {}
```

O texto enviado está contido no campo “data” e deve conter caracteres ASCII printáveis somente.

## Comando para habilitar/desabilitar leitura da porta serial auxiliar COM1

Comando do host para o TED:

```
Body: { "cmd": 8, "args": { "enabled": true } }
```

Resposta do TED para o host:

```
Body: {}
```

O flag que indica se a leitura deve ser habilitada ou não está contido no campo “enabled”, que deve ser *true* ou *false*.

## Comando para habilitar/desabilitar leitura da porta serial auxiliar COM2

Comando do host para o TED:

```
Body: { "cmd": 9, "args": { "enabled": true } }
```

Resposta do TED para o host:

```
Body: {}
```

O flag que indica se a leitura deve ser habilitada ou não está contido no campo “enabled”, que deve ser *true* ou *false*.

## Comando para ler a entrada digital

Comando do host para o TED:

```
Body: {"cmd": 13}
```

Resposta do TED para o host:

```
Body: {"value": 0}
```

A resposta está contida no campo "value" do body (0 ou 1).

## Comando para ligar saída digital

Comando do host para o TED:

```
Body: {"cmd": 14}
```

Resposta do TED para o host:

```
Body: {}
```

## Comando para desligar saída digital

Comando do host para o TED:

```
Body: {"cmd": 15}
```

Resposta do TED para o host:

```
Body: {}
```

## Comando para limpar a lista de atalhos

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 for enviado, o menu é exibido novamente quando requisitado.

Comando do host para o TED:

```
Body: {"cmd": 17}
```

Resposta do TED para o host:

```
Body: {}
```

## Comando para adicionar uma página à lista de atalhos

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

Veja o formato do comando que deve ser enviado ao TED:

Comando do host para o TED:

```
Body:
{
  "cmd": 18,
  "args": {
    "11": "123456789012345",
    "12": "2222222222222222",
    "13": "3333333333333333",
    "14": "4444444444444444",
    "15": "5555555555555555",
    "16": "6666666666666666",
    "17": "7777777777777777"
  }
}
```

Resposta do TED para o host:

```
Body: {}
```

Neste comando fazemos uso do membro “args” para passar o conteúdo de cada linha, que é representado pelos membros “11”, “12”, ..., “17”. Cada linha deve conter no máximo 15 caracteres, como explicado acima.

## Comandos enviados pelo TED ao HOST

No sentido contrário, do TED para o host, o padrão é o mesmo.

### Comando para enviar dados digitados no teclado pelo usuário

Comando do TED para o host:

```
Body: {"source": 1, "text": "coxinha"}
```

Resposta do host para o TED:

```
Body: {}
```

O campo “source” indica a origem do texto. Os valores possíveis são:

- teclado: 0;
- leitora de código de barra USB: 1;
- leitora de código de barra serial: 2;

- serial 1: 3;
- serial 2: 4;
- entrada digital: 5.