

# **BOLETIM TÉCNICO INFORMATIVO**

## **BTI 44/15-A**

**PROTOCOLO DE MENSAGENS DAS PLACAS DE ENTRADAS E SAÍDAS INTELIGENTES**  
**MARCA ZUCHI - LINHA MULTI-IO**  
**PROTOCOLO ASCII - LONGO**

Para o conhecimento de: Usuários e Parceiros Integradores

Assunto: Descrição do protocolo de mensagens

Data de criação: Fevereiro/2015

Data da última atualização: Setembro/2015

## **ÍNDICE:**

### **1. Aplicação**

### **2. Protocolo de Comunicação**

#### **2.1 Mensagens Enviadas à Placa**

##### **2.1.1 Mensagens de Escrita**

##### **2.1.2 Mensagens de Leitura**

##### **2.1.3 Mensagens de Execução**

#### **2.2 Mensagens Enviadas pela Placa**

##### **2.2.1 Mensagens de Resposta**

##### **2.2.2 Mensagens de Alteração de Estado Operacional**

#### **2.3 Composição das Mensagens**

##### **2.3.1 Campo 1**

##### **2.3.2 Campo 2**

##### **2.3.3 Campo 3**

##### **2.3.4 Campo 3**

##### **2.3.5 Campo N**

### **3. Direitos de uso**

## **1. Aplicação**

As Placas da Linha Multi-O, entradas e saídas inteligentes, podem operar com diversos protocolos de comunicação sejam proprietários da Zuchi, protocolos industriais como MODBUS ou DNP3, além de protocolos personalizados, de acordo com a necessidade de cada cliente integrador ou fabricante de equipamentos.

Esta BTI visa informar detalhes referentes as mensagens do protocolo de comunicação desenvolvido pela Zuchi e denominado como PROTOCOLO MULTI-IO ASCII, passíveis de uso nas Placas de Entradas e Saídas Inteligentes da Linha Multi-IO dispostas com interface de comunicação do tipo Ethernet sob o protocolo TCP/IP.

## **2. Protocolo de Comunicação**

As mensagens que compõem o PROTOCOLO MULTI-IO ASCII são classificadas de 2 formas: mensagens enviadas para as placas MULTI-IO e mensagens recebidas das placas MULTI-IO.

As mensagens enviadas para as placas podem ser de 3 tipos: mensagens de escrita, leitura ou execução (read, write, execution).

Já as mensagens recebidas das placas podem ser de 2 tipos: respostas de mensagens de escrita, leitura ou execução ou mensagens de alteração de estado operacional, ou seja, mensagens geradas espontaneamente pelas placas quando algum de seus I/Os sofrem alguma mudança de estado lógico (answer,unsolicited).

Toda mensagem enviada para a placa é respondida pela mesma oferecendo ao software de controle um retorno (feedback), ou seja, reconhecimento do recebimento da mensagem (acknowledge).

### **2.1 Mensagens Enviadas para a Placa**

#### **2.1.1 Mensagens de Escrita (Write)**

São mensagens enviadas para a placa afim de se alterar o estado lógico de alguma saída ou saídas da placa, variáveis ou parâmetros operacionais, por exemplo:

- Configuração dos estados lógicos das saídas após a energização da placa;
- Alteração de estados lógicos das saídas da placa (relés) e etc;

### **2.1.2 Mensagens de Leitura (Read)**

São mensagens enviadas a placa afim de se ler o estado lógico de alguma de suas entradas ou saídas, variáveis ou parâmetros operacionais, por exemplo:

- Versão do firmware das placas;
- Estados lógicos das entradas e saídas da placa e etc;

### **2.1.3 Mensagens de Execução (Execution)**

São mensagens enviadas a placa afim de se executar algumas tarefas operacionais internas, por exemplo:

- Reinicialização (reset) da placa;
- Retorno das configurações ao padrão de fábrica e etc;

## **2.2 Mensagens Enviadas pela Placa**

### **2.2.1 Mensagens de Resposta (Answer)**

As mensagens de resposta são mensagens enviadas pela placa em resposta aos comandos de escrita, leitura ou execução podendo ser a resposta efetiva de um comando de leitura ou apenas um feedback de um comando de escrita ou execução.

### **2.2.2 Mensagens de Alteração de Estado Operacional (Unsolicited)**

As mensagens de alteração de estado operacional são mensagens enviadas pela placa em razão da alteração de uma de suas entradas. Este tipo de mensagem é gerada de forma espontânea pela placa toda vez que uma entrada sofre alguma alteração de estado lógico.

## **2.3 Composição das Mensagens**

As mensagens do PROTOCOLO MULTI-IO ASCII são compostas por N campos, sendo cada campo separado por um caractere ‘;’ (ponto e vírgula), inclusive o último campo, finalizada com 1 caractere ASCII CR (carriage return) e 1 caractere ASCII LF (line feed).

O PROTOCOLO MULTI-IO ASCII adicionalmente pode utilizar um campo final contendo o cálculo do CHECKSUM ou CRC da mensagem e também utilizar

criptografia nas mensagens aumentando a segurança na via de trânsito das mensagens.

Consulte a Zuchi caso estes requisitos sejam necessários em sua aplicação.

Exemplo:

CAMPO\_1;CAMPO\_2;[CAMPO\_3];[CAMPO\_4];[CAMPO\_N];+CR+LF

No exemplo acima, o CAMPO\_3, CAMPO\_4 e CAMPO\_N que se encontram entre '[' ']' (colchetes) são opcionais, ou seja, em algumas mensagens do protocolo estes campos podem ou não serem utilizados.

Todos os campos do protocolo são alfanuméricos, ou seja, os comandos, funções, identificação das variáveis da placa, estados operacionais e valores, são representados por números e/ou letras.

### 2.3.1 Campo 1

O CAMPO\_1 contém o endereço da placa e é identificado com os 3 últimos octetos do endereço MAC da placa.

O endereço MAC (Media Access Control) é um endereço físico associado à interfaces Ethernet de comunicação da placa que conectam dispositivos à rede de computadores (LAN). O MAC é um endereço único, não havendo dois dispositivos com a mesma numeração e é usado para controle de acesso em redes de computadores. A Zuchi dispõe de registro e autorização junto ao IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos), órgão responsável pelo gerenciamento das licenças de MAC em todo o mundo assegurando a seus usuários um controle dos endereços de MAC.

Apesar do endereço MAC ser formado por um conjunto de 6 bytes separados por ':' (dois pontos), sendo cada byte representado por dois algarismos em notação hexadecimal, como por exemplo: "40:D8:55:10:40:00", o PROTOCOLO MULTI-IO ASCII faz uso apenas dos 3 últimos octetos do endereço MAC sem a presença dos ':'.

O CAMPO\_1 está presente tanto nas mensagens de leitura, escrita e ações.

Exemplo de mensagem com destaque ao CAMPO\_1:

**104000**;E;1S;L;+CR+LF // Liga o relé da saída 1

### 2.3.2 Campo 2

O CAMPO\_2 contém o tipo de mensagem representado por uma letra sendo:

L - Se a mensagem for de leitura (read)

E - Se a mensagem for de escrita (write)

X - Se a mensagem for de execução (execution)

A - Se a mensagem for de ação, também chamada alteração de estado (unsolicited)

O CAMPO\_2 está sempre presente tanto nas mensagens de leitura, escrita e ações. Exemplo de mensagem com destaque ao CAMPO\_2:

104000;**E**;1S;L;+CR+LF // Liga o relé da saída 1

104000;**L**;1E;+CR+LF // Lê o estado da entrada 1

104000;**X**;RS;+CR+LF // Força um reset da placa

104000;**A**;1E;0;+CR+LF // Mudança de estado lógico na entrada 1, de 1 para 0

### 2.3.3 Campo 3

O CAMPO\_3 pode conter o endereço de um único IO da placa, o endereço de um conjunto de IOs ou ainda o endereço de uma variável operacional da placa, identificados por números e letras, sendo:

Para uma Placa Multi-IO 2E2S (2 entradas e 2 saídas)

Endereços de IOs:

1S - Saída número 1

2S - Saída número 2

1E - Entrada número 1

2E - Entrada número 2

TS - Todas as saídas

TE - Todas as entradas

ES - Todas as entradas e todas as saídas

Variáveis Operacionais:

MM - Habilita e desabilita a memória dos estados das saídas fazendo com que na energização ou re-energização da placa as saídas retornem ao último estado que se encontravam antes da desenergização

NS - Habilita e desabilita a geração automática de mensagens após a alteração do estado de uma das entradas (mensagens não solicitadas - unsolicited)

EI - Estado inicial das saídas após a energização ou re-energização da placa

EM - Estado da memória das saídas

O CAMPO\_3 está sempre presente tanto nas mensagens de leitura, escrita e ações.

Exemplo de mensagem com destaque ao CAMPO\_3:

104000;E;1S;L;+CR+LF // Liga o relé da saída 1

104000;E;TS;L;+CR+LF // Liga todos os relés

104000;L;ES;+CR+LF // Lê o estado de todas as entradas e saídas

#### 2.3.4 Campo 4

O CAMPO\_4 pode conter um estado lógico ou função a ser executada em uma determinada saída ou conjunto de saídas quando a mensagem for de escrita ou ainda, o estado de uma entrada ou saída, o estado de todas as entradas, o estado de todas as saídas ou ainda o estado de todas as entradas e saídas, identificados por números e letras, sendo:

0 - Desliga ou desligada

1 - Liga ou ligada

D - Desliga ou desligada

L - Liga ou ligada

I - Inverte

T - Temporiza

P - Pulsa

O CAMPO\_4 está sempre presente nas mensagens de escrita e respostas de leitura e ações.

Exemplo de mensagens com destaque ao CAMPO\_4:

104000;E;1S;L;+CR+LF // Liga o relé da saída 1

104000;E;TS;L;+CR+LF // Liga todos os relés das saídas

#### 2.3.5 Campo N

Campos adicionais citados acima como o CAMPO\_N, podem ser necessários na composição de algumas mensagens, como as mensagens de temporização e pulsos nas saídas a relés das placas.

As mensagens de escrita do tipo T (TEMPORIZAÇÃO), necessitam de mais 2 campos adicionais para informação do tempo de duração da temporização e a unidade de tempo (milisegundos, Segundos, Minutos ou Horas) portanto, um CAMPO\_5 e um CAMPO\_6 serão necessários para complementar a mensagem do Tipo T, ficando assim:

CAMPO\_5 - Tempo

CAMPO\_6 - m se milisegundos, S se segundos, M se minutos, H se horas

Exemplos:

104000;E;1S;T;100;m;+CR+LF// Temporiza o relé da saída 1 por 100 milisegundos

104000;E;1S;T;1;S;+CR+LF // Temporiza o relé da saída 1 por 1 segundo

104000;E;2S;T;5;M;+CR+LF // Temporiza o relé da saída 2 por 5 minutos

As mensagens de escrita do tipo P (PULSOS), necessitam de mais 5 campos para informação da quantidade de pulsos ou ciclos, tempo de duração e unidade de tempo, tempo de intervalo e unidade de temporização portanto, um CAMPO\_5, CAMPO\_6, CAMPO\_7, CAMPO\_8 e CAMPO\_9 serão necessários para complementar a mensagem do Tipo P, ficando assim:

CAMPO\_5 - Quantidade de pulsos ou ciclos

CAMPO\_6 - Tempo de duração

CAMPO\_7 - m se milisegundos, S se segundos, M se minutos, H se horas

CAMPO\_8 - Tempo de intervalo

CAMPO\_9 - m se milisegundos, S se segundos, M se minutos, H se horas

Exemplos:

104000;E;1S;P;5;200;m;200;m;+CR+LF// Pulsa o relé da saída 5 vezes 1 com pulsos de 200ms de duração e intervalo

### **Observação Importante:**

Os comandos T e P nas Placas Multi-IO consideram o estado atual dos relés, ou seja, invertem e retornam os estados dos relés a cada ciclo, certifique-se portanto de assegurar o estado inicial dos relés antes de enviar estes comandos.

### **3. Lista de Mensagens**

Exemplos de mensagens disponíveis para a placa 2E2S.



### 3.1 Mensagens de Escrita (Write)

#### 3.1.1 Escrita em IO ou escrita nas saídas

Endereços disponíveis:

1S - Saída 1

2S - Saída 2

TS - Todas as saídas

**Mensagem enviada: (write)**

**104000**;E;1S;L;+CR+LF ou **104000**;E;1S;1;+CR+LF // Liga o relé da saída 1

**Mensagem recebida: (acknowledge)**

**104000**;E;1S;1;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;1S;D;+CR+LF ou **104000**;E;1S;0;+CR+LF // Desliga o relé da saída 1

**Mensagem recebida: (acknowledge)**

**104000**;E;1S;0;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;1S;I;+CR+LF // Inverte o estado do relé 1

**Mensagem recebida: (acknowledge)**

**104000**;E;1S;I;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;TS;L;+CR+LF // Liga todos os relés

**Mensagem recebida: (acknowledge)**

**104000**;E;TS;L;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;TS;I;+CR+LF // Inverte todos os relés

**Mensagem recebida: (acknowledge)**

**104000**;E;TS;I;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;1S;T;100;m;+CR+LF // Temporiza o relé 1 por 100 milissegundos

**Mensagem recebida: (acknowledge)**

**104000**;E;1S;T;100;m;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;1S;T;10;S;+CR+LF // Temporiza o relé 1 por 10 segundos

**Mensagem recebida: (acknowledge)**

104000;E;1S;T;1;S;+CR+LF

**Mensagem enviada: (write)**

104000;E;1S;T;5;M;+CR+LF // Temporiza o relé 1 por 5 minutos

**Mensagem recebida: (acknowledge)**

104000;E;1S;T;5;M;+CR+LF

**Mensagem enviada: (write)**

104000;E;1S;T;1;H;+CR+LF // Temporiza o relé 1 por 1 hora

**Mensagem recebida: (acknowledge)**

104000;E;1S;T;1;H;+CR+LF

**Mensagem enviada: (write)**

104000;E;1S;P;5;200;m;200;m;+CR+LF // Pulsa 5 vezes o relé 1 com durações de 200 milissegundos e intervalos de 200ms

**Mensagem recebida: (acknowledge)**

104000;E;1S;P;5;200;m;200;m;+CR+LF

### 3.1.2 Escrita em Variáveis Operacionais

Endereços disponíveis:

NS - Habilita e desabilita mensagens de alteração de estado das entradas (mensagens não solicitadas - unsolicited)

MM - Habilita e desabilita a memória do estado das saídas

EI - Define o valor inicial de cada saída na energização da placa

**Mensagem enviada: (write)**

104000;E;NS;0;+CR+LF // Desabilita a geração de mensagens não solicitadas

**Mensagem recebida: (acknowledge)**

104000;E;NS;0;+CR+LF

**Mensagem enviada: (write)**

104000;E;NS;1;+CR+LF // Habilita a geração de mensagens não solicitadas

**Mensagem recebida: (acknowledge)**

104000;E;NS;1;+CR+LF

**Mensagem enviada: (write)**

104000;E;MM;0;+CR+LF // Desabilita a memória dos estados das saídas

**Mensagem recebida: (acknowledge)**

104000;E;MM;0;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;MM;1;+CR+LF // Habilita a memória dos estados das saídas

**Mensagem recebida: (acknowledge)**

**104000**;E;MM;1;+CR+LF

**Mensagem enviada: (write)**

**104000**;E;EI;10;+CR+LF // Relé 1 ligado e relé 2 desligado na energização

**Mensagem recebida: (acknowledge)**

**104000**;E;EI;10;+CR+LF

### 3.2 Mensagens de leitura (Read)

#### 3.2.1 Leitura de IO - entradas e saídas

Endereços disponíveis:

1S - Saída 1

2S - Saída 2

1E - Entrada 1

2E - Entrada 2

TS - Todas as saídas

TE - Todas as entradas

ES - Todas as entradas e saídas

**Mensagem enviada: (write)**

**104000**;L;1E;+CR+LF // Leitura do estado da entrada 1

**Mensagem recebida: (acknowledge)**

**104000**;L;1E;1;+CR+LF // Entrada 1 em 1

**Mensagem enviada: (write)**

**104000**;L;1S;+CR+LF // Leitura do estado da saída 1

**Mensagem recebida: (acknowledge)**

**104000**;L;1S;0;+CR+LF // Saída 1 em 0 (relé desatracado)

**Mensagem enviada: (write)**

**104000**;L;ES;+CR+LF // Leitura do estado das entradas e saídas

**Mensagem recebida: (acknowledge)**

**104000**;L;ES;1011;+CR+LF // Entrada 1 em 1, entrada 2 em 0, saída 1 em 1 e saída 2 em 1, respectivamente

#### 3.2.2 Leitura de Variáveis Operacionais

Endereços disponíveis:

NS - Habilita e desabilita mensagens de alteração de estado das entradas (mensagens não solicitadas - unsolicited)

MM - Habilita e desabilita a memória do estado das saídas

EI - Estado inicial de cada saída na energização da placa

EM - Estado atual da memória de cada saída

**Mensagem enviada: (write)**

**104000**;L;NS;+CR+LF // Leitura do estado das mensagens de alteração de estados das entradas

**Mensagem recebida: (acknowledge)**

**104000**;L;NS;1;+CR+LF // Mensagens de alteração de estados das entradas habilitadas

**Mensagem enviada: (write)**

**104000**;L;MM;+CR+LF // Leitura do estado da memória das saídas

**Mensagem recebida: (acknowledge)**

**104000**;L;MM;0;+CR+LF // Memória do estado das saídas desabilitada

**Mensagem enviada: (write)**

**104000**;L;EI;+CR+LF // Leitura do estado inicial das saídas após a energização

**Mensagem recebida: (acknowledge)**

**104000**;L;EI;11;+CR+LF // Atracar os 2 relés após a energização da placa

#### **4. Direitos de uso**

O protocolo de comunicação das Placas Multi-IO fabricadas pela Zuchi é de domínio público (freeware).

Qualquer dúvida, ajuda ou sugestão por favor entre em contato conosco pelos canais abaixo descritos.

A Zuchi reserva o direito de alterar as especificações deste documento sem prévio aviso.

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

E-mail: [suporte@zuchi.com.br](mailto:suporte@zuchi.com.br)

Skype: pedrozuchi

Tels: (32) 3218 6367 ou (32) 3218 6292