

Action.NET – IEC-60870-5-101
Slave Communication Protocol
Versão 2016.2.1

00054.03

Março, 2019

Sumário

| | |
|---|----------|
| 1. INFORMAÇÃO GERAL..... | 1 |
| 1.1 Sumário..... | 1 |
| 1.2 Objetos de dados suportados (ASDUs)..... | 1 |
| 1.3 Funcionamento geral | 2 |
| 2. CONFIGURAÇÃO DO CANAL..... | 3 |
| 2.1 Opções de protocolo | 3 |
| 2.2 Interface..... | 4 |
| 2.3 Settings | 4 |
| 2.4 Timeouts..... | 5 |
| 3. CONFIGURAÇÃO DE “NODES” | 6 |
| 3.1 Parâmetros..... | 6 |
| 4. CONFIGURAÇÃO DE PONTOS | 7 |
| 4.1 Geral..... | 7 |
| 4.2 Tipos de pontos..... | 7 |
| 4.3 Endereço do ponto | 9 |
| 4.4 Parâmetro de comando..... | 9 |
| 4.4.1 Configuração do Parâmetro | 9 |
| 4.4.2 Utilização do parâmetro no protocolo Servidor (Slave) | 10 |
| 4.5 Access Type..... | 10 |

1. INFORMAÇÃO GERAL

1.1 Sumário

Communication Driver Name: IEC8705101S

Current Version: 2016. 2.1

Implementation DLL: T.ProtocolDriver.IEC8705101S.dll

Protocol: IEC-60870-5-101 Slave standard protocol

Interface: Serial

Description: O protocolo IEC870501S implementa a comunicação com estações clientes compatíveis com este protocolo, atuando como estação escrava (servidora).

Clients types supported: Qualquer cliente compatível com IEC-60870-5-101.

Communication block size: Maximum 250 bytes, formato FT 1.2;

Protocol Options: tamanhos em bytes de endereços, senha para comandos, timeouts e tipo de timestamp.

Multi-threading: Configurável pelo usuário, default é uma threads para todos os nós da rede..

Max number of nodes: user defined

PC Hardware requirements: Standard Serial Port;

PC Software requirements: ActionNET system.

1.2 Objetos de dados suportados (ASDUs)

O protocolo utiliza as ASDUs definidas para o IEC-60870-5-101 e seus tipos de objetos de dados. Somente suporta a comunicação serial.

M_SP_NA: 1 - Single-point information ;

M_DP_NA: 3 - Double-point information ;

M_ST_NA: 5 - Step position;

M_BO_NA: 7 - Bitstring with 32 bits ;

M_ME_NA: 9 - Measured value, normalized ;

M_ME_NB: 11 - Measured value, scaled value ;

M_ME_NC: 13 - Measured value Float;

M_IT_NA: 15 - Integrated totals ;

C_SC_NA: 45 - Single command ;

C_DC_NA: 46 - Double command ;

C_RC_NA: 47 - Regulating step command ;

C_SE_NA: 48 - Set point command, normalized value ;

C_SE_NC: 50 - Set point command, 32 bits floating point ;

C_BO_NA: 51- Write Bitstring de 32 bits;

E também todas as variantes com timestamp de 24 e de 56 bits das ASDUs acima. No cadastramento de pontos utiliza-se os códigos acima, mas ao enviar mudanças não solicitadas são utilizadas para pontos digitais (SP, DP, ST e BO) as variantes com data e timestamp como definido na opção do protocolo. As medidas analógicas são sempre enviadas sem timestamp.

1.3 Funcionamento geral

O protocolo IEC-870-5-101 está implementado no modo Servidor no inicia aguardando comandos de reset e solicitação de status na Camada de Ligação (Link Layer) de IEDs que implementam o protocolo IEC-870-5-101 Cliente ou Mestre. O frame usado para a troca de mensagens é o FT 1.2, versão não balanceada (escravo não envia eventos não solicitados). Várias parametrizações estão disponíveis para acomodar perfis diferentes de implementações do protocolo, conhecido por certa flexibilidade definida na própria norma.

Esse módulo responde a pedidos de leitura de variáveis - analógicas e digitais - transmissão de eventos e execução de comandos. A implementação tem as seguintes características:

- Responde a pedidos de leitura cíclica (amostragem geral) de pontos digitais simples/duplos e analógicos e de Contadores
- Envia, quando solicitadas Classe1 e Classe2, alterações de estados de pontos digitais e de mudanças em medidas analógicas, considerando banda morta e tempo para integridade.
- Utiliza alternativamente time tag de 24 ou 56 bits de comprimento, no envio de alterações em SP, DP, ST e BO;
- Aceita comandos para pontos digitais simples e duplos, regulating step, BitString e set points para medidas analógicas;
- Suporta comando direto, select e Select and execute
- Faz tratamento da qualidade dos pontos (QDS);

2. CONFIGURAÇÃO DO CANAL

2.1 Opções de protocolo

CommonAddress Num Octets – Número de bytes utilizados no endereço da estação escrava para a camada de aplicação. Pode ser 1 ou 2 bytes.

LinkAddress Num Octets – Número de bytes utilizados no endereço da estação escrava para a camada de enlace (Link Layer). Pode ser 1 ou 2 bytes.

ObjectAddress Num Octets - Número de bytes utilizados no endereço dos objetos de dados. Pode ser 1, 2 ou 3 bytes.

CauseOfTransmission Num Octets – Número de bytes utilizados para a informação de causa de transmissão do dado. Pode ser 1 ou 2 bytes.

Password for commands – Este parâmetro deve ser utilizado, se os comandos aqui recebidos serão enviados por um outro canal de comunicação que também está usando senha para o envio de comandos. No caso a senha aqui definida deve ser igual a definida para o protocolo que enviará o comando.

Timeout (ms) – Tempo de espera em milissegundos para o recebimento de uma resposta após envio de uma solicitação qualquer. É uma verificação funcional, que não leva em consideração o tipo de solicitação ou resposta, apenas verifica se há respostas sendo recebidas no canal.

UseCP56Tags – Deve ser escolhida a opção **True**, para que os envios de alterações de pontos definidos como digitais utilizem as variações de ASDUs que utilizam o timestamp no formato de 56 bits. Caso contrário serão utilizados timestamp de 24 bits.

Maximum Changes to send a message – Para melhorar o desempenho do módulo de comunicação ao enviar mudanças em medidas analógicas, pode-se definir este número como o máximo de mudanças que devem ser acumuladas para serem enviadas em uma mensagem única, ao invés do envio de uma medida em cada mensagem. O número considerado bom é 30 medidas, que é o valor default. Esta acumulação é utilizada quando as alterações em medições são recebidas pelo módulo de comunicação, como eventos. (AccessType com WriteEventEnable)

Max time to send analog changes (ms) – Este tempo é definido como o máximo de espera para o envio de uma mensagem com mudanças em medidas analógicas. Se desde que iniciou o acumulo de medições para a mesma mensagem este tempo se esgotar antes da chegada do número de medidas acima definido, este módulo enviará a mensagem com as medidas que já chegaram. Este tempo por default é definido em 3 segundos. Se for utilizado o modo Get analog changes by sampling (veja abaixo), este é o tempo a ser usado como intervalo entre duas amostragens.

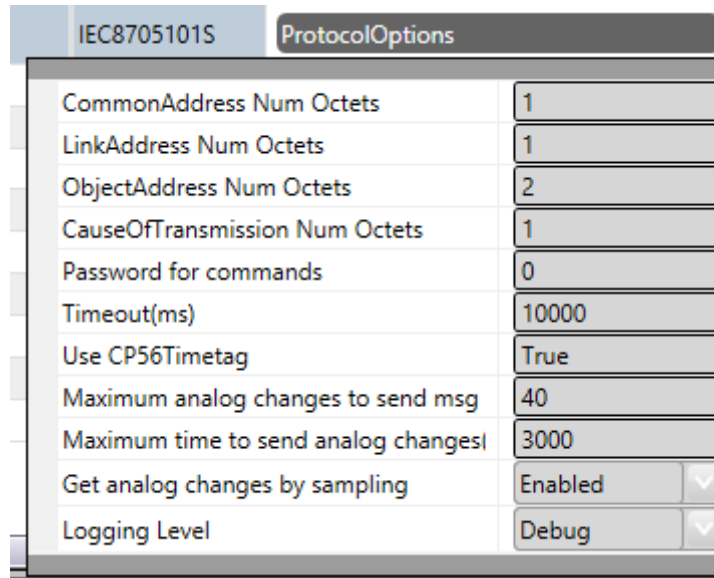
Get analog changes by sample – Alternativamente ao modo de recebimento de alterações de valores de tags, (pelo uso de AccessType com WriteEventEnable), pode-se escolher o uso do modo de amostragem, pelo módulo de comunicação, das alterações ocorridas em tags. Neste modo, periodicamente, é feita a verificação dos valores atuais contra os últimos valores enviados. Desta forma a alteração somente é considerada, e o novo valor enviado ao cliente, se a diferença absoluta entre o valor atual e o ultimo enviado é maior que o atributo Deadband do tag. Para o uso deste modo deve-se usar o AccessType na table Points, para estes tags de medições, com WriteEventEnable desativado.

O intervalo de tempo entre cada duas verificações será o definido no parâmetro acima, **Max time to send analog changes (ms)**.

Logging Level – Pode-se escolher nesta lista o modo de funcionamento do log criado pelo módulo de comunicação.

| | | |
|----------------------|-------|--|
| Logging level | Debug | Todas as mensagens são registradas no LOG. |
|----------------------|-------|--|

| | | |
|--|---------|--|
| | Info | Apenas as mensagens de Info, Warning e Error são registradas no LOG. |
| | Warning | Apenas as mensagens de Warning e Error são registradas no LOG. |
| | Error | Apenas as mensagens de Error são registradas no LOG. |



2.2 Interface

A interface de comunicação utilizada é denominada **CustomSerial**, por ser implementada não através das bibliotecas básicas do Action.NET mas sim por uma biblioteca de comunicação serial integrada ao próprio protocolo de comunicação.

2.3 Settings

Canal Serial:

Com: Porta de comunicação serial utilizada;

BaudRate

DataBits: 8

StopBits: 1ou 2

Parity: None, Even, Odd;

DTR: on, off

RTS: on,off

CTS: on, off;

Configurações da porta devem ser compatíveis com as do IED Cliente:

| Settings | |
|-----------------|-------------------------------------|
| Serial Settings | |
| Com Port | com1 |
| Baud Rate | 115200 |
| Data bits | 8 |
| Stop Bits | 1 |
| Parity | none |
| DTR | DtrOn |
| RTS | RtsOn |
| CTS | <input checked="" type="checkbox"/> |
| CtsWait | 100 |
| InvalidValues | |

2.4 Timeouts

Normalmente as definições de timeout default são apropriadas para a maioria das instalações. Caso for necessário podem ser alterados os valores em milissegundos definidos na figura abaixo.

| Settings | | Timeout |
|---------------------|--|---------|
| com1;115200;8;1;... | | Timeout |
| Tx | | 2000 |
| RxStart | | 2000 |
| RxFinish | | 1000 |
| NextByte | | 500 |
| Retry | | 0 |
| Unsolicited | | 5000 |

3. CONFIGURAÇÃO DE “NODES”

Cada nó (node) é uma estação servidora. Pode-se configurar uma única estação servidora para cada canal.

3.1 Parâmetros

Common Address – É o endereço do IED na camada de aplicação

LinkAddress – É o endereço do IED para a camada de enlace (Link Layer)

Clock Adjust – Em algumas implementações, em que não há GPS acoplado ao IED Escravo (servidor), o cliente pode mandar uma solicitação de sincronismo, pela qual o módulo de comunicação fará alteração da data-hora do computador em que está sendo executado.

Tag for Comm status – Neste campo pode ser indicado o nome de um tag existente no projeto para receber indicação de sucesso/falha na comunicação, do ponto de vista funcional. Quando são feitas solicitações o módulo aguarda um máximo de **Timeout** milissegundos (definido em Protocol Options, acima) para o recebimento de uma resposta. No caso de falha o valor deste tag será colocado em ZERO. No caso de sucesso o valor será colocado em UM.

| PrimaryStation | |
|---|--------------------------------------|
| 192.168.0.127;20000;5;Disabled;Disabled;10000;0;30000;1000;1000;0;0;... | |
| 1;1;True;Tag.TGIEC.COMMOK | |
| 1 | CommonAddress 1 |
| 192.1 | LinkAddress 1 |
| 127.0 | Clock adjust True |
| 172.2 | Tag for comm status Tag.TGIEC.COMMOK |
| 10.0. | |

Backup Station – Não utilizado para estações em modo servidor.

4. CONFIGURAÇÃO DE PONTOS

4.1 Geral

Os pontos podem ser de entrada ou de saída. Os pontos de entrada, isto é, os pontos que são adquiridos pelo protocolo cliente tem basicamente dois parâmetros principais: O tipo de ponto e o seu endereço. Os pontos de saída, utilizados para telecomandos, possuem além do tipo e do endereço um parâmetro para especificar a operação de saída. No mapa de endereços de um IED nunca são repetidos endereços. Os endereços são únicos não importando o tipo do ponto.

4.2 Tipos de pontos

O módulo de comunicação no modo Escravo, implementa:

- Recebimento de data e hora para sincronismo;
- Resposta a pedido de interrogação geral
- Envio de frames de informação, não solicitados, devido alterações do dado na memória.
- Time tag de 24 ou de 56 bits de comprimento;
- Recepção de Comandos de Pontos digitais simples, duplos, Regulating steps e BitStrings;
- Recebimento de comandos de Set Points para pontos analógicos
- Recepção de Comando select before operate;
- Tratamento da qualidade do ponto (QDS);

Os tipos de pontos implementados são definidos pelos próprios objetos de dados definidos na norma IEC, a seguir apresentados.

M_SP_NA: 1 - Single-point information

Ponto de entrada binario simples, valor 0 ou 1. As variantes enviadas pelo servidor são com “timetag” M_SP_TA(=2) ou M_SP_TB(=30) quando envio espontaneo, ou a própria M_SP_NA_1, sem timestamp, nas respostas à Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_DP_NA: 3 - Double-point information ;

Ponto de entrada duplo, que pode assumir os estados 0 a 3. Normalmente utilizados na sinalização de estados de chaves e disjuntores. As variantes enviadas pelo servidor são com “timetag” M_DP_TA (=4) ou M_DP_TB(=31) quando envio espontaneo, ou a própria M_DP_NA, sem timestamp, nas respostas à Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_ST_NA: 5 - Step position;

Valor de passo ou step, na faixa de -64 a +63, utilizado principalmente para posição de tap de transformadores ou outra informação de posição. As variantes enviadas pelo servidor são com “timetag” M_ST_TA ou M_ST_TB quando envio espontaneo, ou a própria M_ST_NA, sem timestamp, nas respostas à Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_BO_NA: 7 - Bitstring with 32 bits ;

Informação de estados binarios como uma cadeia de 32bits. Não é feita qualquer manipulação pelo driver. A configuração é tratada como um número long. As variantes enviadas pelo servidor são com “timetag” quando envio espontaneo M_BO_TA ou M_BO_TB(=33) ou a própria M_BO_NA nas respostas à Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_ME_NA: 9 - Measured value, normalized ;

Medição analógica normalizada de 16 bits com sinal. Valor entre -32768 e + 32767. É calculado como um número real entre 0 e 1 antes de ser atribuído ao tag em tempo real. Deve-se usar “scalling” para reproduzir o valor em unidade de engenharia. A variante enviada pelo servidor é a mesma sem “timetag” M_ME_NA tanto para o caso de alterações como as respostas às Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_ME_NB: 11 - Measured value, scaled value ;

Medição analógica escalar utilizada para transmissão de grandezas analógicas. Também de 16 bits, valor entre -32768 e 32767. A variante enviada pelo servidor é a mesma sem “timetag” M_ME_NB tanto para o caso de alterações como as respostas às Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_ME_NC: 13 - Measured value short floating point;

Medição analógica em formato de número real fracionário, utilizada para transmissão de grandezas analógicas. As medidas são campos de 32 bits no formato IEEE STD 754, que implementa números de ponto flutuante. A variante enviada pelo servidor é a mesma sem “timetag” M_ME_NC tanto para o caso de alterações como as respostas às Interrogações Gerais. No cadastramento somente é utilizado este tipo.

M_IT_NA: 15 - Integrated totals ;

Medição analógica inteira com sinal. Medidas com 32 bits inteiro. A variante enviada pelo servidor é a mesma sem “timetag” M_IT_NA tanto para o caso de alterações como as respostas às Interrogações Gerais. No cadastramento somente é utilizado este tipo.

C_SC_NA: 45 - Single command ;

Comando para ponto simples (1 bit). Detalhes do comando podem ser escolhidos clicando o botão a direita do campo. Também pode ser entrado diretamente o número que é o código de comando resultante da escolha dos detalhes. Cada ponto será parametrizado estáticamente na tabela POINTS, de forma que deverá ser configurado um ponto para abertura e outro para fechamento de chaves de um bit.

C_DC_NA: 46 - Double command ;

Comando para ponto duplo (2 bits). Detalhes do comando podem ser escolhidos clicando o botão a direita do campo. Também pode ser entrado diretamente o número que é o código de comando resultante da escolha dos detalhes. Cada ponto será parametrizado estáticamente na tabela POINTS, de forma que deverá ser configurado um ponto para abertura e outro para fechamento de chaves com sinalização de dois bits.

C_RC_NA: 47 - Regulating step command ;

Comando para regulagem de passo (step) normalmente utilizado para envio de pulsos subir ou descer comutadores de “tap” de transformadores. Detalhes do comando podem ser escolhidos clicando o botão a direita do campo. Também pode ser entrado diretamente o número que é o código de comando resultante da escolha dos detalhes. Cada ponto será parametrizado estáticamente na tabela POINTS, de forma que deverá ser configurado um ponto para subir e outro para descer a posição do “tap”.

C_SE_NA: 48 - Set point command, normalized value ;

Para envio de set points de 16 bits, normalizado para IEDS que suportam este tipo de comando.

O valor a ser enviado é o que estiver no momento como valor do “tag” cujo endereço foi enviado no comando.

C_SE_NC: 50 - Set point command, short floating point value ;

Para envio de set points de 32 bits, em formato de ponto flutuante IEEE STD 764, para IEDs que suportam este tipo de comando. O valor a ser enviado é o que estiver no momento como valor do “tag” cujo endereço foi enviado no comando.

C_BO_NA: 51- Write Bitstring de 32 bits

Para a escrita no IED servidor de uma Informação de estados binários como uma cadeia de 32bits. Não é feita qualquer manipulação pelo driver. A configuração é tratada como um numero long sem sinal. O valor a ser enviado é o que estiver no momento como valor do “tag” cujo endereço foi enviado no comando. O tipo do tag deve ser “long” ou AnalogInt, isto é um inteiro de 32 bits.

4.3 Endereço do ponto

O campo **Address** a ser preenchido no cadastramento de pontos é o que a norma chama de “Information Object Address”. Trata-se de um número com número de bytes como definido no Protocol Options.

4.4 Parâmetro de comando

O parâmetro de comando é para os tipos de comandos implementados um código de um byte, que detalha o que e como o comando deve ser executado pelo IED. Nesta implementação, ao ser cadastrado um ponto com tipo de comando de saída, aparece este campo para ser preenchido. Se já se sabe o código que se quer utilizar basta digitá-lo no campo. Se não se sabe, clica-se no botão a direita do campo para que seja mostrada janela com as ações e detalhes que podem ser escolhidos.

4.4.1 Configuração do Parâmetro

Os códigos gerados pela escolha de itens na janela de definição do parâmetro de comando são formados pelo cálculo da soma de duas parcelas (A e B), a primeira indicativa da ação, e segunda de detalhes da operação, como definidas abaixo:

Para Comando Simples C_SC_NA:

- 0 = Desliga (A)
- 1 = Ligar (A)
- 0 = Nenhum detalhe (B)
- 4 = Pulso Curto (B)
- 8 = Pulso Longo (B)
- 12= Sinal Persistente (B)

Para Comando Duplo C_DC_NA:

- 1 = Desligar (A)
- 2 = Ligar (A)
- 0 = Nenhum detalhe (B)
- 4 = Pulso Curto (B)
- 8 = Pulso Longo (B)
- 12= Sinal Persistente (B)

Para Comando de Regulagem (de Tap) C_RC_NA:

00054.03

| | |
|-----------------------|-----|
| 1 = Descer | (A) |
| 2 = Subir | (A) |
| 0 = Nenhum detalhe | (B) |
| 4 = Pulso Curto | (B) |
| 8 = Pulso Longo | (B) |
| 12= Sinal Persistente | (B) |

A opção restante é se o comando será do tipo **Select** – apenas para selecionar o dispositivo a ser comandado, ou se é **Execute** - isto é para enviar o comando de ação propriamente. No caso **Select** deve-se somar 128 ao código até aqui obtido pela soma das parcelas A e B.

Exemplo: código = 9, em um comando simples significa *Pulso longo* para *Ligar* dispositivo;

4.4.2 Utilização do parâmetro no protocolo Servidor (Slave)

O servidor ao receber um comando do mestre (cliente) faz a sua execução de acordo com o parâmetro vindo na mensagem. **O parâmetro definido na base de dados do servidor não é utilizado**, podendo ser configurado de qualquer forma.

O comportamento do servidor na execução do comando é a seguir explanado.

Select / Execute

SELECT - não haverá execução propriamente, isto é, não haverá alteração na memória do servidor. Uma mensagem será enviada ao log (Trace), indicando a modalidade SELECT, se o ponto de saída foi de fato encontrado no servidor. Caso o ponto não exista uma mensagem de erro “POINT NOT FOUND!”, aparecerá no log.

EXECUTE- O comando será executado normalmente e aparecerá no log, mensagem indicativa do fato com a indicação EXECUTE.

Opções de Detalhes – Opção B

0 – Nenhum – Será feito apenas um “toggle” no estado do ponto destino do comando. (se zero vai para 1 se 1 vai para zero), qualquer que seja o valor da parcela A.

4 – Pulso Curto – O valor da parcela A será colocado no ponto de destino, mantido assim por 100 ms, e após será restaurado o valor original.

8 – Pulso Longo – O valor da parcela A será colocado no ponto de destino, mantido assim por 1000 ms, e após será restaurado o valor original.

12 – Persistente – O valor da parcela A será colocado no ponto de destino e deixado assim.

4.5 Access Type

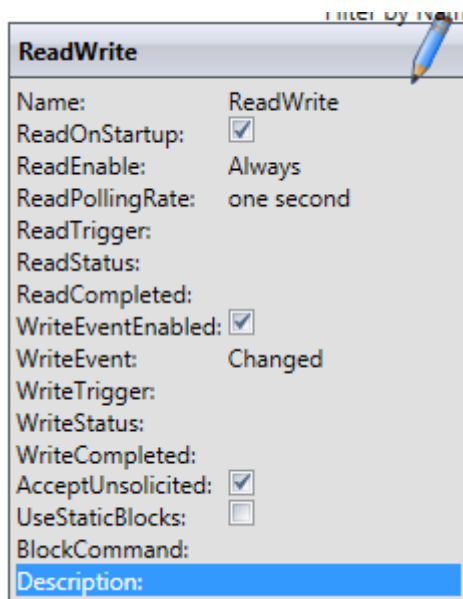
Por ser um módulo de comunicação em modo escravo (servidor) há a necessidade de algumas características próprias para a parametrização do campo **Access Type** da tabela **Points**:

Para pontos com tipos que são de leitura: (usando eventos ao ocorrem alterações de estado)

- M_SP_NA: 1 - Single-point information ;
- M_DP_NA: 3 - Double-point information ;
- M_ST_NA: 5 - Step position;
- M_BO_NA: 7 - Bitstring with 32 bits ;
- M_ME_NA: 9 - Measured value, normalized ;
- M_ME_NB: 11 - Measured value, scaled value ;
- M_ME_NC: 13 - Measured value Float;
- M_IT_NA: 15 - Integrated totals ;

O Access Type deve ser definido com: (como na figura ReadWrite)

- ReadOnStartup= On;
- ReadPooling= Always;
- ReadPoolongRate: 500 mili
- WriteEventEnable = On
- WriteEvent= Changed;
- AcceptUnsolicitited = On;



Para pontos com tipos de MEDIDAS - (usando amostragem para identificar alterações de valor- Opção Get analog changes by sampling)

- M_ME_NA: 9 - Measured value, normalized ;
- M_ME_NB: 11 - Measured value, scaled value ;
- M_ME_NC: 13 - Measured value Float;
- M_IT_NA: 15 - Integrated totals ;

O Access Type deve ser definido com: (como na figura SlaveAna)

ReadOnStartup= On;

ReadPooling= Always;

ReadPoolongRate: one second

WriteEventEnable = off

WriteEvent= Changed;

AccepUnsolicitited = off;

| SlaveAna | |
|--------------------|-------------------------------------|
| Name: | SlaveAna |
| ReadOnStartup: | <input checked="" type="checkbox"/> |
| ReadEnable: | Never |
| ReadPollingRate: | one second |
| ReadTrigger: | |
| WriteEventEnabled: | <input type="checkbox"/> |
| WriteEvent: | Changed |
| WriteTrigger: | |
| AcceptUnsolicited: | <input type="checkbox"/> |
| UseStaticBlocks: | <input type="checkbox"/> |
| BlockCommand: | |
| Description: | |

Para pontos com tipos que são de comandos:

C_SC_NA: 45 - Single command ;

C_DC_NA: 46 - Double command ;

C_RC_NA: 47 - Regulating step command ;

C_SE_NA: 48 - Set point command, normalized value ;

C_SE_NC: 50 - Set point command, 32 bits floating point ;

C_BO_NA: 51- Write Bitstring de 32 bits

O Access Type deve ser definido com: (como na figura WriteSlave)

ReadPooling = Never;

WriteEventEnable = off

WriteEvent= Changed;

| WriteSlave | |
|--------------------|--------------------------|
| Name: | WriteSlave |
| ReadOnStartup: | <input type="checkbox"/> |
| ReadEnable: | Never |
| ReadPollingRate: | one second |
| ReadTrigger: | |
| WriteEventEnabled: | <input type="checkbox"/> |
| WriteEvent: | Changed |
| WriteTrigger: | |
| AcceptUnsolicited: | <input type="checkbox"/> |
| UseStaticBlocks: | <input type="checkbox"/> |
| BlockCommand: | |
| Description: | |