

## ActionNET – DNP 3.0

### Slave Communication Protocol

## Sumário

<b>1. INFORMAÇÃO GERAL.....</b>	<b>1</b>
1.1 Sumário.....	1
1.2 Objetos de dados suportados.....	1
1.3 Funcionamento geral .....	5
<b>2. CONFIGURAÇÃO DO CANAL.....</b>	<b>6</b>
2.1 Opções de protocolo .....	6
2.2 Settings .....	7
2.3 Timeout.....	8
2.4 Estado Inicial.....	8
<b>3. CONFIGURAÇÃO DE “NODES” .....</b>	<b>9</b>
3.1 Dados dos Canais (primário e backup).....	9
<b>4. CONFIGURAÇÃO DE PONTOS .....</b>	<b>10</b>
4.1 Geral.....	10
4.2 Tipos de pontos.....	10
4.3 Endereço do ponto .....	13
4.4 Access Type.....	15



## 1. INFORMAÇÃO GERAL

### 1.1 Sumário

**Communication Driver Name:** DNP30S

**Current Version:** 1.0

**Implementation DLL:** T.ProtocolDriver.DNP30S.dll

**Protocol:** DNP3.0 Slave standard protocol

**Interface:** TCP/IP ou Serial

**Description:** O driver é utilizado para a comunicação em modo escravo (ou Servidor) com supervisórios ou outros equipamentos que utilizam o protocolo DNP3 Nível 2 em Modo MESTRE (ou Cliente). A comunicação pode ser feita através de canal serial ponto a ponto ou utilizando rede local Ethernet e protocolo TCP-IP como escravo único de um endereço IP.

**Masters types supported:** Qualquer equipamento em Modo MESTRE compatível DNP 3.0

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

**Protocol Options:** Modo "LinkConfirm" e endereço da estação mestre.

**Multi-threading:** Configurável pelo usuário, default é cinco threads para cda nó da rede.

**Max number of nodes:** user defined

**PC Hardware requirements:** Standard PC Ethernet interface board, RS485 or RS232 port

**PC Software requirements:** ActionNET system.

### 1.2 Objetos de dados suportados

O quadro abaixo apresenta os objetos DNP e suas variantes, suportados por esta implementação.

Objeto			Requisição (Mestre)		Resposta (Escravo)	
Obj.	Var	Descrição	Function Codes (decimal)	Qualifier Codes (Hex)	Function Codes (decimal)	Qualifier Codes (Hex)
1	0	Binary Input (any variation)	1	00,01,06		
			22	00,01,06	129	
1	1	Single Bit Binary Input (packed)	1	00,01,06	129	00, 01
1	2	Binary Input with status			129	00, 01
2	0	Binary Input event (any variation)	1	06,07,08		
2	1	Binary Input change without time	1	06,07,08	129,130	17,18
2	2	Binary Input change with absolut time	1	06,07,08	129,130	17,18
2	3	Binary Input change with relative time	1	06,07,08	129,130	17,18
3	0	Double bit Binary input – Any variation	22	00,01,06		
3	1	Double-bit Binary Input – Packed	1	00,01,06	129	00, 01
3	2	Double-bit Binary Input – With flags	1	00,01,06	129	00, 01

4	0	Double-bit Binary Input Event – Any Variation	1	06,07,08		
4	1	Double-bit Binary Input Event – whitout time	1	06,07,08	129,130	17,18
4	2	Double-bit Binary Input Event- with absolut time	1	06,07,08	129,130	17,18
4	3	Double-bit Binary Input Event - with relative time	1	06,07,08	129,130	17,18
10	1	Binary Output – Any Variation	1	00,01,06		
10	2	Binary Output – status with flags	1	00,01,06	129	
12	1	Control relay output block	3,4,5	17,28	129	Echo of request
20	0	Binary Counter – all variations				
20	1	Counter – 32-bit with flag	1	00,01,06	129	00, 01
20	2	Counter – 16-bit with flag	1	00,01,06	129	00, 01
20	5	Counter – 32-bit without flag	1	00,01,06	129	00, 01
20	6	Counter – 16-bit without flag	1	00,01,06	129	00, 01
21	0	Frozen counter – all variations				
21	1	Frozen Counter – 32-bit with flag	1	00,01,06	129	00, 01
21	2	Frozen Counter – 16-bit with flag	1	00,01,06	129	00, 01
21	3	Frozen Counter – 32-bit without flag	1	00,01,06	129	00, 01
21	4	Frozen Counter – 16-bit without flag	1	00,01,06	129	00, 01
22	0	Counter Event – Any Variation	1	06		
22	1	Counter Event – 32-bit with flag	1	06,07,08	129,130	17,18
22	2	Counter Event – 16-bit with flag	1	06,07,08	129,130	17,18
23	0	Frozen Counter Event – Any Variation	1	06,07,08		
23	1	Frozen Counter Event – 32-bit with flag	1	06,07,08	129,130	17,18
23	2	Frozen Counter Event – 16-bit with flag	1	06,07,08	129,130	17,18
30	0	Analog Input – all variations	1, 22	00,01,06		
30	1	32 Bits Analog Input	1	00,01,06	129	00, 01
30	2	16 Bit Analog input with flag	1	00,01,06	129	00, 01
30	3	32 Bits Analog Input without flag	1	00,01,06	129	00, 01
30	4	16 Bit Analog input without flag	1	00,01,06	129	00, 01
30	5	Short Floating Point (32bits)	1	00,01,06	129	00, 01
32	0	Analog Input event – all variations	1	06,07,08		
32	1	32 Bits Analog Input event	1	06,07,08	129,130	17,18
32	2	16 Bit Analog event without flag	1	06,07,08	129,130	17,18
32	3	32 Bit Analog event with flag	1	06,07,08	129,130	17,18
32	4	16 Bit Analog event with flag	1	06,07,08	129,130	17,18
32	5	Analog input event single float –without time	1	06,07,08	129,130	17,18

32	7	Analog input event single float –with time	1	06,07,08	129,130	17,18
40	0	Analog Output Status–any variation	1			
40	1	Analog Output Status – 32bits with flag	1	00,01,06	129	00, 01
40	2	Analog Output Status -16bits with flag	1	00,01,06	129	00, 01
40	3	Analog output status – Single float with flag	1	00,01,06	129	00, 01
41	1	32Bit Analog output block	3,4,5,6	17,28	129	Echo of request
41	2	16 Bit Analog output block	3,4,5,6	17,28	129	Echo of request
41	3	Analog output block – Single float	3,4,5,6	17,28	129	Echo of request
50	1	Time and Data – Absolut time	1,2	0x07	129	07
51	1	Time and Date CTO – Absolute time, synchronized			129,130	07
51	2	Time and Date CTO – Absolute time, unsynchronized			129,130	07
52	1	Time Delay – Coarse			129	07
52	2	Time Delay – Fine			129	07
60	1	Class 0 data	1	0x06		
60	2	Class 1 data	1,20,21	06,07,08		
60	3	Class 2 data	1,20,21	06,07,08		
60	4	Class 3 data	1,20,21	06,07,08		
80	1	Internal indications	1,2	00,01	129	01

Na implementação do protocolo o mestre só executa as requisições destacadas em azul. O equipamento servidor, responde utilizando as respostas destacadas em amarelo. Observe-se que cabe ao equipamento servidor decidir como será a resposta e o mestre deve suportar todas as funções do nível 2 possíveis de serem utilizadas como resposta.

Os objetos, variações de objeto, códigos de função e qualificadores tem seus significados padronizados no DNP. Abaixo são mostradas as tabelas de código de função e qualificador:

Código de Função	Descrição	Origem
1	Ler	Mestre
2	Escrever	Mestre
3	Selecionar	Mestre
4	Operar	Mestre
5	Operar direto (sem seleção)	Mestre
6	Operar direto (sem ack)	Mestre
7	Congela Imediatamente	Mestre

8	Congela Imediatamente (sem ack)	Mestre
9	Congela e limpa	Mestre
10	Congela e lê	Mestre
13	Restart (Cold)	Mestre
14	Restart (Warm)	Mestre
20	Habilita mensagem não solicitada	Mestre
21	Desabilita mensagem não solicitada	Mestre
22	Assinala classe a objeto	Mestre
23	Medida com delay	Mestre
129	Resposta	Escravo
130	Resposta não solicitada (não existe no nível 2)	Escravo

Código de Qualificador	Uso em requisição	Uso em resposta
00,01	Um intervalo de pontos estáticos (classe 0) ou um único ponto com um número	Objeto estático
06	Todos os pontos	Inválido
07,08	Uma quantidade limitada de eventos. Um ponto simples sem número (isto é uma data / hora)	Um ponto simples sem número (isto é uma data / hora)
17, 28	Controles (usualmente um ou mais pontos não relacionados)	Objetos evento (usualmente um ou mais pontos não relacionados)

O DNP tem o conceito de classes de dados, sendo definidas quatro classes:

**Classe 0:** Corresponde ao ponto estático, analógico ou digital. Seu conteúdo é o valor de uma variável analógica ou digital, de entrada ou saída, em um dado momento;

**Classe 1, 2 e 3:** Corresponde a eventos de transição de estados ou de variáveis de classe 0 ou situações internas à remota / relé que causam o evento.

O que ocorre normalmente nos IED's ao utilizar-se o DNP, é associar à variação de estado de variáveis digitais ou valores de banda morta de variáveis analógicas as classes 1, 2 e 3. Dessa forma, a modificação do estado / valor dessas variáveis causará eventos que serão transmitidos através do pedido de eventos das respectivas classes (60/2, 60/3 e 60/4). Periodicamente, pode ser feita uma leitura cíclica para verificação de integridade. Essa leitura, corresponde a um pedido de classe 0 (60/1).

**Observações:**

- 1) Nesta implementação é considerado automaticamente que variáveis digitais Tipo BI, quando sofrerem alteração serão enviadas como de Classe 1, na forma do Objeto 2 com variação 2 (Binary input with time stamp).
- 2) Nesta implementação é considerado automaticamente que variáveis analógicas AI, AIF, quando sofrerem alteração serão enviadas como de Classe 2, na forma dos Objetos 30 variação 3, para AI e como 30 variação 5 para o AIF.

### 1.3 Funcionamento geral

A sequência normal de operação do escravo é:

- Caso seja necessário na instalação o mestre deverá enviar mensagem de sincronismo (50,1) periodicamente. A data e horário recebido será entendido como UCT (ou GMT), e será utilizada para alterar o relógio do computador “host”.
- Ao iniciar a execução, nas primeiras mensagens de resposta com, enviará flags de IIN indicando que IED reiniciou (restart). O mestre deverá então executar uma escrita de “clear device flag” (80/1) para limpar esta indicação;
- Sempre que houver alteração do estado de digitais ou analógicas, que sejam configurados na tabela POINTS, com *AccessType como ReadWrite*, será enviada mensagem de evento como Classe 1 ou Classe 2. Caso não se desejar eventos em alguns pontos usar para estes *AccessType = Read*;
- Quando o modo escravo recebe um pedido de comando de saída digital ou saída analógica, através software envia a respectiva requisição para o IED (12/1 ou 41/2);

## 2. CONFIGURAÇÃO DO CANAL

### 2.1 Opções de protocolo

**LinkConfirmMode** – Modo do protocolo em que na camada de enlace todas as solicitações recebem resposta de confirmação de seu recebimento. Por default este campo é “enable”.

**MasterStationID** - Número entre 1 e 65534, unívoco e corresponde ao endereço da estação mestre. Assim, um mestre quando envia mensagem a este escravo deve informar seu endereço. Por default este campo é “65534”.

**Password for commands:** Para aumentar a segurança no envio de comandos, normalmente iniciados apenas por uma mudança no estado de um tag, é possível especificar nos módulos Clientes uma senha de até 9 algarismos, para o comando. Aqui neste módulo servidor deve ser especificada a senha utilizada por este módulo Servidor para gerar o comando para o módulo Cliente que de fato enviará o comando. Esta senha deve ser a mesma que a utilizada pelo módulo Cliente destinatário do comando.

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

<b>Logging level</b>	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.

**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.

**Max time to send analog changes (ms)** – Se for utilizado o modo Get analog changes by sampling, este é o tempo a ser usado como intervalo entre duas amostragens. Este tempo por default é definido em 3 segundos.



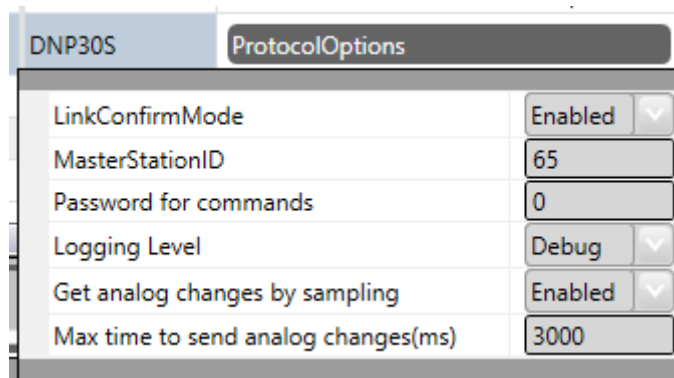


Figura 1 – Opções do Protocolo

## 2.2 Settings

### Canais Serial:

Com: Porta de comunicação do computador 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 no mestre e escravo.

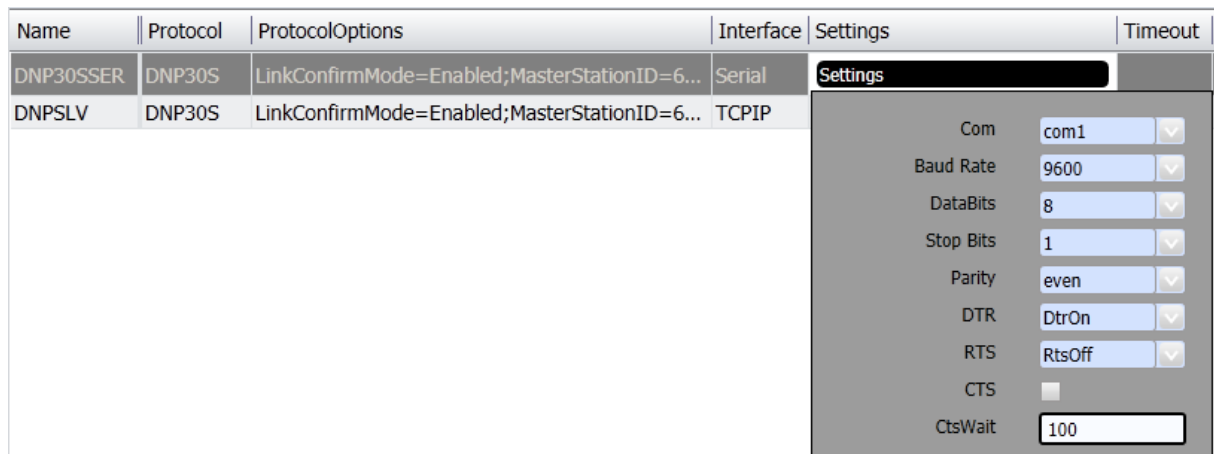


Figura 2 – Settings de Canal Serial

### TCP/IP channels:

- **ServerMode** e **AcceptUnsolicited** são pré marcados e não podem ser alterados;
- **ListeningPort** - Deve-se definir a porta no micro onde este canal escravo vai aceitar conexões.
- **MaxSimultaneousConnections**- Deve ser 1.
- **ShareNodeSameIP**: Aceita IED's com o mesmo IP.

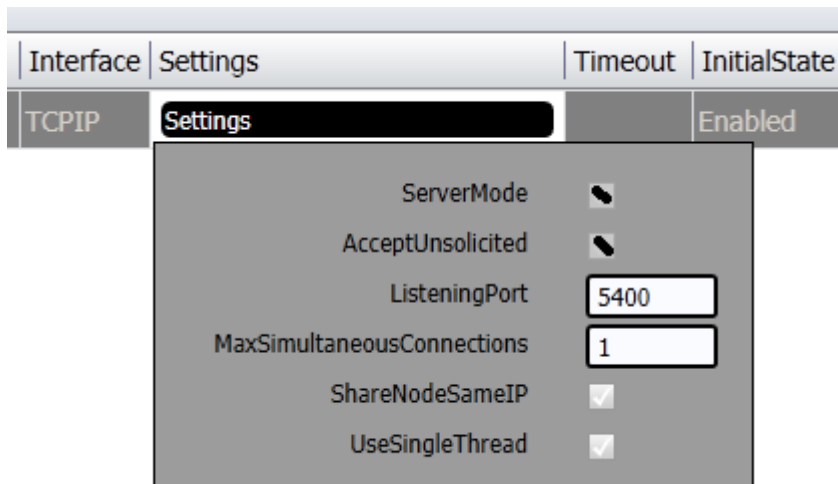


Figura 3 – Settings de TCP/IP

### 2.3 Timeout

Define tempos limites para a transmissão e recepção de caracteres de mensagens e o número de retries.

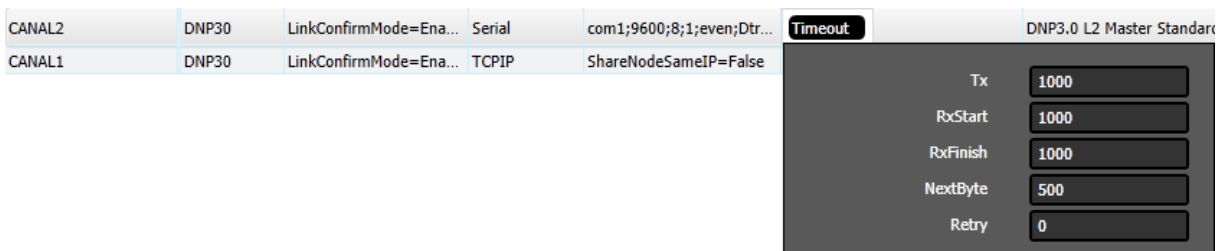


Figura 4 – Atributos de Timeout

### 2.4 Estado Inicial

Define como o canal iniciará: habilitado ou Desabilitado. Se o canal estiver desabilitado, não serão enviadas nem recebidas mensagens através do mesmo, isto é o canal ficará inativo.

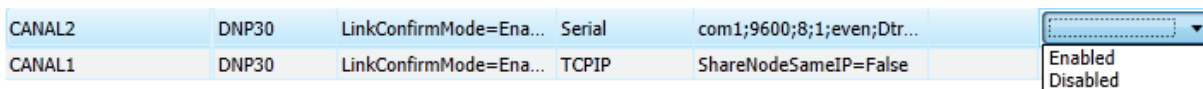


Figura 5 – Estado inicial do Canal

### 3. CONFIGURAÇÃO DE “NODES”

Deve ser configurado um único nó como estação servidora em cada canal. Caso seja necessário servir a mais de um cliente devem ser configurados tantos canais quantos são os clientes.

#### 3.1 Dados dos Canais (primário e backup)

Conjunto de atributos associados ao nodo (canal), referentes ao seu endereço e outros atributos apresentados abaixo:

abled;Enabled;Tag.TGDNP30.COMMOK	
IP	0
Port	0
SlaveID	2
IgnoreBitOnLine	Enabled
WaitForIdleToCmd	Enabled
Tag for comm status	Tag.TGDNP30.COMMOK

Figura 6 – Dados dos Canais

Para o caso de comunicação via Tcp-Ip:

**Endereço IP** – Não utilizado.

**Port** - Não utilizado

Para ambos TCP-IP e Serial:

**SlaveID** –Endereço da estação Escrava, definida por este canal.

**IgnoreBitOnLine** - Se “enabled” indica que o driver deve ignorar a indicação de “BitOnLine” que é inserida pelo IED quando ocorre falha ou normalidade em um ponto, segundo o critério do IED.

**WaitForIdleToCmd** – Se “enabled” indica que um comando somente será enviado quando a comunicação de amostragem estiver em repouso, isto é, não estiver ocorrendo.

**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. O módulo aguarda um máximo de **Timeout (=8000ms)** ara o recebimento de um pedido do cliente. No caso de falha o valor deste tag será colocado em ZERO. No caso de sucesso o valor será colocado em UM.

## 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 servidos pelo protocolo tem basicamente dois parâmetros principais: O tipo de ponto e o seu endereço.

No mapa de endereços de um IED os endereços reiniciam para cada tipo de ponto.

Os estados ou valores dos pontos são reportados ao cliente por Objetos de Informação definidos no protocolo. Estes objetos possuem variações como com ou sem “timestamp”. Sempre que o escravo reportar mudanças em classes 1 ou 2 , utilizará a variação com “timestamp” utilizando o ultimo “timestamps” correspondente de um ponto no ActionNET. Quando o escravo reportar leituras gerais de analógicos, contadores ou digitais este protocolo utilizará variação sem “timestamp”.

O módulo de comunicação no modo Escravo, implementa as seguintes operações:

- Respostas a pedidos de leitura de pontos digitais;
- Respostas a pedidos de leitura de variáveis analógicas;
- Respostas a pedidos de Leitura de contadores;
- Execução de solicitação de comandos de Pontos digitais e analógicos;

### 4.2 Tipos de pontos

Os tipos de pontos implementados são definidos utilizando os objetos de dados definidos na norma, a seguir apresentados. Para cada tipo de ponto quaisquer que sejam as variações de objetos recebidos do IED , com “flag” ou sem, com timestamp ou sem, os valores adquiridos serão colocados em pontos com o tipo entre os abaixo listados. Na tabela **pontos**, no campo “*address*” escolhe-se o tipo do ponto e especifica-se seu endereço.

ID	TagName	Node	Address	DataType	AccessType
273	TGDNP30.AI.T044	NODNP30	AI:0044	Native	Read
1058	TGDNP30.AI.T044			Native	ReadWrite
1314	TGDNP30.AI.T044			Native	Read
272	TGDNP30.AI.T045			Native	Read
1059	TGDNP30.AI.T045			Native	ReadWrite
1313	TGDNP30.AI.T045			Native	Read
271	TGDNP30.AI.T046			Native	Read
1060	TGDNP30.AI.T046			Native	ReadWrite
1312	TGDNP30.AI.T046			Native	Read
270	TGDNP30.AI.T047			Native	Read
1061	TGDNP30.AI.T047			Native	ReadWrite
1311	TGDNP30.AI.T047			Native	Read
269	TGDNP30.AI.T048			Native	Read
1062	TGDNP30.AI.T048			Native	ReadWrite
1310	TGDNP30.AI.T048			Native	Read

### AI - Analog Input

Medição analógica escalar utilizada para transmissão de grandezas analógicas. Os dados de tags com este tipo de ponto são enviados através do objeto 32 variação 1. São numeros de 32 bits inteiros sem timesatmp.

### BI - Binary Input

Ponto de entrada binario simples, valor 0 ou 1. Os dados de tags com este tipo de ponto são enviados através do objeto 2 variação 2, quando por motivo de alteração (evento), com timestamp, ou através do objeto 2 var. 1, quando por leitura geral (classe 0).

### DBI - Double bits Binary Input

Ponto de entrada binario duplo, valor entre 0 e 3. Os dados de tags com este tipo de ponto são enviados através do objeto 3 e variação 2. quando por alteração (evento), com timestamp, ou através do objeto 3 var. 1, quando por leitura geral (classe 0).

### RAO – Read Analog Output

Não utilizado na implementação do modo escravo.

### CRO - Control Relay Output

Ponto de saída digital utilizado para receber comandos de chaves e disjuntores. É esperado o objeto DNP 12, incluído o *Control Code* (8 bits) para indicação do tipo de comando e detalhes de execução.

No cadastramento destes pontos em modo escravo, é preciso definir o control code idêntico ao uque será nviado pelo IED Mestre, para que se possa identificar o tag utilizado no Escravo. No escravo somente é executada a alteração do valor do ponto do Action.NET. Para que o comando chegue ao destino final em um IED o tag afetado deverá ser utilizado por um outro protocolo Cliente, que o enviará ao IED.

No caso de controle de chaves e disjuntores recomenda-se a utilização de dois Tags: um para o Trip e outro para o Close.

Os valores possíveis recebidos do cliente são apresentados na tabela abaixo com as respectivas ações associadas:

Control Code	Ação protocolo	Ação no tag Action.NET executada no escravo
1	Saída Pulse ON	Altera para 1 e a seguir para Zero
2	Saída Pulse OFF	Altera para Zero e a seguir para 1
3	Saída Latch ON	Altera para 1
4	Saída Latch OFF	Altera para ZERO
65	Saída Pulse ON + Close	Altera para 1 e a seguir para Zero
66	Saída Pulse OFF + Close	Altera para Zero e a seguir para 1
67	Saída Latch ON + Close	Altera para 1
68	Saída Latch OFF + Close	Altera para ZERO
129	Saída Pulse ON + Trip	Altera para 1 e a seguir para Zero
130	Saída Pulse OFF + Trip	Altera para Zero e a seguir para 1
131	Saída Latch ON + Trip	Altera para 1
132	Saída Latch OFF + Trip	Altera para ZERO

+ 16	Enfileirar (Queue) + Trip	Nada faz
+ 32	Limpar (Clear) + Trip	Nada faz

### CT – Counter

Contador binário de 16 ou 32bits, enviado através do objetos 22 variação 1. Este numero possui o ultimo estado do tag de contagem, no instante em que é enviado.

### FRZ – Frozen Counter

Contador binário de 32 bits, enviado através do objeto 23 variação 1, que contém a informação de um contador utilizado como um tag.

### AO - Analog Output Status or Block (16bits)

Quando este modo escravo recebe mensagem com Function Code de escrita (2 = Write ou 4=Operate, ou 5= DirectOperate ) e este objeto ou 41 (valor requerido para ser alcançado na saída analógica) na suas variações 2 (16 bits), este valor é escrito no tag definido na tabela POINTS para o endereço recebido no objeto..

### AOL - Analog Output long (32 bits)

Quando este modo escravo recebe mensagem com Function Code de escrita (2 = Write) e este objeto 41 (valor requerido para ser alcançado na saída analógica) na suas variações 1 (32 bits), este valor é escrito no tag definido na tabela POINTS para o endereço recebido no objeto..

### AIF - Analog Input Floating (32 bits)

Medição analógica utilizada para transmissão de grandezas analógicas. Tags com este tipo devem ser tipificados no Action.Net como reais. Seu valor é enviado através do objeto 32 variação 5. São numeros de ponto flutuante de 32 ou 64 bits, formato IEEE-754.

### IIN - Internal Indications

Como é o padrão do protocolo os status dos Internal Indications ou status de resposta de comandos são enviados em todas as mensagens de resposta do escravo, para utilização pelo cliente. A tabela mostra o formato destas duas palavras de 16 bits.

BIT	Origem	Descrição	Conteúdo
0	IIN	Broadcast	Retorna 1 quando escravo recebe uma mensagem de broadcast (endereço = FFFF)
1	IIN	Classe 1	Retorna 1 se escravo tem eventos classe 1
2	IIN	Classe 2	Retorna 1 se escravo tem eventos classe 2
3	IIN	Classe 3	Retorna 1 se escravo tem eventos classe 3
4	IIN	Sincronismo de relógio	Retorna 1 se escravo pede sincronização de relógio
5	IIN	Saídas em local	Retorna 1 se algumas ou todas as saídas estão em local
6	IIN	Problema	Retorna 1 se escravo com problema
7	IIN	Reinício	Retorna 1 se escravo reiniciou (restart)

8	IIN	Função não implementada	Retorna 1 se função enviada pelo mestre não está implementada no escravo
9	IIN	Objeto desconhecido	Retorna 1 se escravo não tem determinado objeto ou classe não tem esse objeto
10	IIN	Dado inválido	Retorna 1 se tem parâmetro inválido no qualificador ou intervalo de endereços está inválido
11	IIN	Overflow	Retorna 1 se buffer do escravo teve overflow
12	IIN	Ocupado	Retorna 1 se requisição foi entendida mas já está sendo executada
13	IIN	Dado corrompido	Retorna 1 se dados de parametrização estão corrompidos
14	IIN	Reservado	Sempre 0
15	IIN	Reservado	Sempre 0

Status field: 8 bits enviados como resposta de um comando.

BIT	Origem	Descrição	Conteúdo
0	Status	Comando aceito	Retorna 1 após comando correto
1	Status	Comando não aceito	Retorna 1 se ocorreu time out entre tempo de select e operate.
2	Status	Falta select	Retorna 1 se foi feito um operate sem prévio select
3	Status	Erro formato	Retorna 1 se comando tem erro de formato
4	Status	Controle não suportado	Retorna 1 se operação não é suportada
5	Status	Fila cheia	Retorna 1 se fila de requisição do escravo está cheia ou ponto já está ativo
6	Status	Erro hardware	Retorna 1 se ocorreu um erro de hardware do dispositivo ao tentar executar o comando
7		Não utilizado	

### 4.3 Endereço do ponto

O campo **Address** a ser preenchido no cadastramento de pontos é o que a norma chama de “Index”. Trata-se de um número de 16 bits que é o índice [0 a n-1] indicativo de cada um dos pontos do mesmo tipo mapeados dentro do IED.

Abaixo, a título de exemplo é apresentada uma tabela de pontos preenchida com diferentes tipos de ponto. Os pontos tipo saída digital (CRO), conforme citado acima, além do endereço possuem o código de controle.

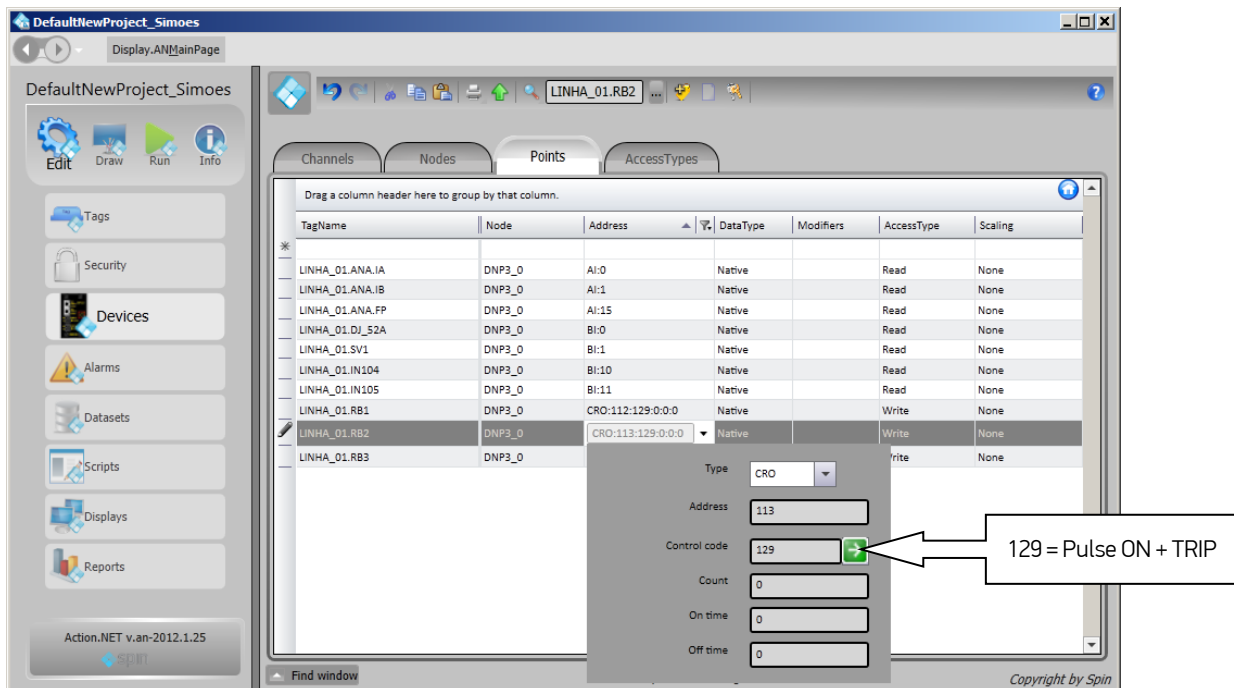
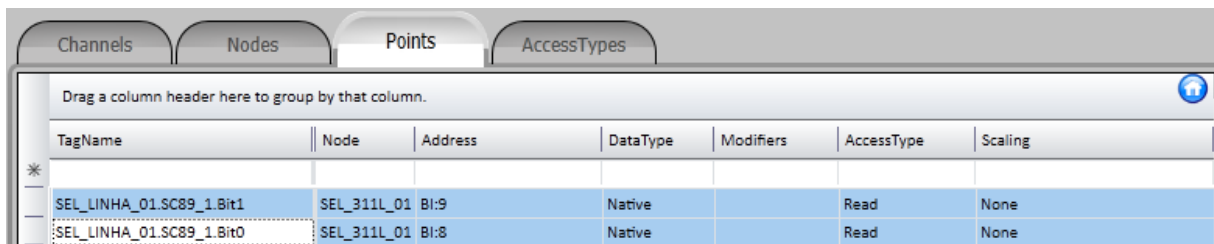


Figura 12 – Exemplo da tabela de Points com diferentes tipos de pontos

Para implementar pontos de entrada digital discretos, basta usar o atributo “Bit” de um Tag, para cada um dos pontos que definem o valor da entrada digital discreta. Assim, por exemplo, uma seccionadora que tem dois contatos para definir seu estado:

Tag	ENDEREÇO	Complemento
SEL_LINHA_01.SC89_1	8	Seccionadora aberta
SEL_LINHA_01.SC89_1	9	Seccionadora Fechada

É definida como um Tag AnalogInt e na tabela node se usa o atributo Bit desta variável de 16 bits (AnalogInt) para endereçar os dois pontos, como na tabela abaixo:



Os valores assumidos pela variável SEL\_LINHA\_01.SC89\_1 serão:

Tag	Bit 0	Bit 1	VALOR	SIGNIFICADO
SEL_LINHA_01.SC89_1	0	0	0	INDEFINIDO
SEL_LINHA_01.SC89_1	1	0	1	ABERTA
SEL_LINHA_01.SC89_1	0	1	2	FECHADA
SEL_LINHA_01.SC89_1	1	1	3	ERRO



## 4.4 Access Type

Por ser um módulo de comunicação em modo 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)**

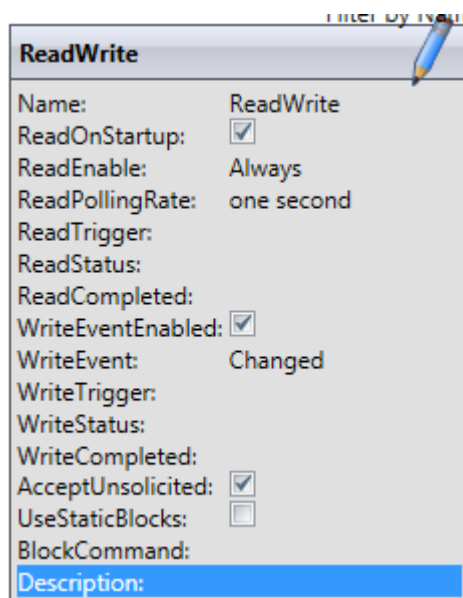
**Access Type deve ser definido com:**

ReadOnStartup= On;

ReadPooling= Never;

WriteEvent= Changed;

AcceptUnsolicited = On;



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

Pontos tipo AI, AIF, CT EAO e FRZ.

Utilizar writeEventEnable = false: como na figura **SlaveAna**

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:

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:	