



## FAÇA FÁCIL: DRIVER IGS PARA COMUNICAÇÃO DE PROTOCOLOS

### 1. PROPRIETÁRIOS INTRODUÇÃO

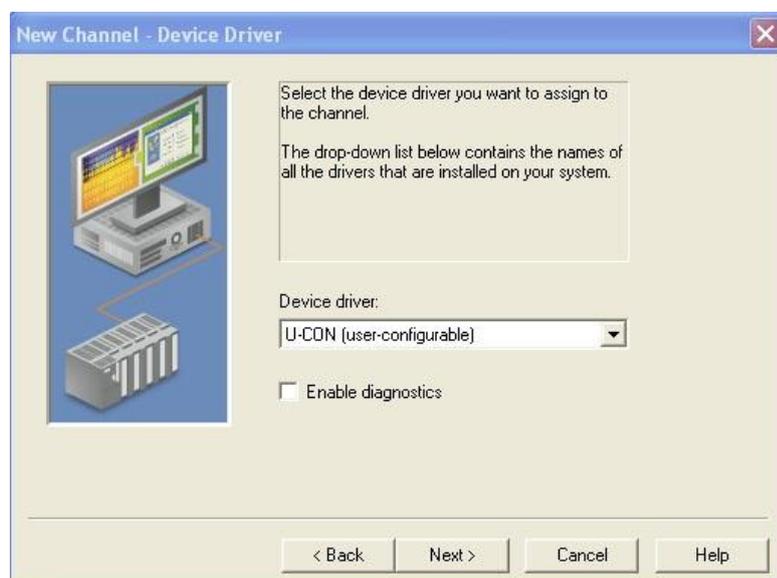
O Driver IGS possui um módulo de configuração que possibilita a comunicação com protocolos proprietários. Trata-se do Driver **U-CON** (Configurável pelo Usuário), o qual pode ser programado para trabalhar com uma variedade muito ampla de dispositivos e é uma ferramenta confiável para se conectar via Ethernet e Serial. Isso permite a comunicação com diversos aplicativos, incluindo HMI/SCADA, Historiador, MES, ERP e inúmeras aplicações personalizadas. Perfis de drivers são criados usando o Editor de Transação integrado ao driver e os elementos de transação são selecionados a partir de menus sensíveis ao contexto, eliminando assim a necessidade de aprender linguagens de script e reduzindo a possibilidade de erros.

### 2. CONFIGURAÇÃO DO DRIVER

São quatro as etapas necessárias para configurar o Driver U-CON. Os usuários devem definir um canal servidor, um dispositivo servidor, perfil de dispositivo e, em seguida, testar e depurar a configuração. Embora as duas primeiras etapas sejam relativamente simples, as duas últimas provavelmente irão exigir uma quantidade significativa de esforço e atenção.

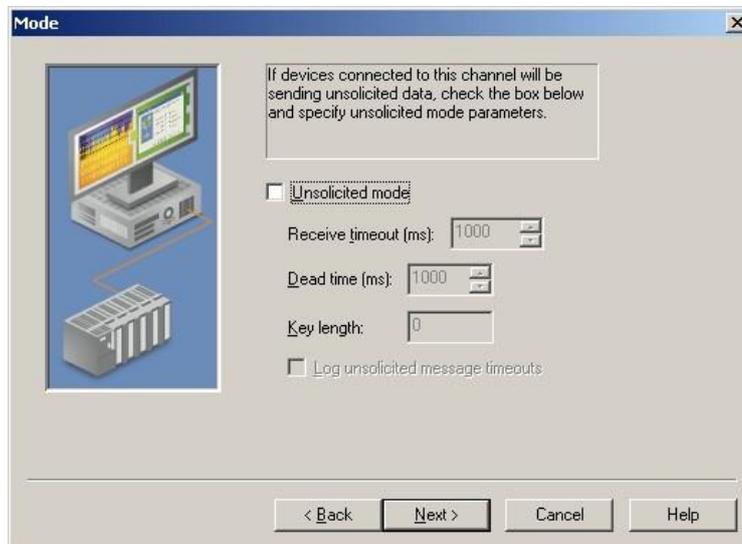
### 3. Etapa 1 - Definição De Um Canal Server

O primeiro passo na criação de qualquer projeto no IGS é a definição de um canal. Muitos dispositivos podem ser conectados a um único canal, desde que todos eles possam usar o mesmo protocolo e Driver. Neste caso, execute o Assistente de Criação do servidor Canal e escolha o driver U-CON da lista de drivers disponíveis.





Em seguida, especifique os parâmetros de comunicação (por exemplo uma taxa de transmissão, paridade, número de bits de dados e etc) que são necessários para o dispositivo que será especificado. Na caixa de diálogo final, o modo de U-CON Driver é especificado. O diálogo deve aparecer como mostrado a seguir:



A requisição de dados poderá ser *solicited* (padrão) ou *unsolicited*. No modo *solicited*, o driver vai solicitar dados de cada dispositivo periodicamente (até 100 vezes ou mais por segundo por Tag). Para o *unsolicited*, o DRIVER não solicita os dados do dispositivo, mas sim, aguarda o envio dos mesmos pelo dispositivo. Cabe ao dispositivo o modo a ser utilizado.

Uma vez que o modo do canal do DRIVER é definido, ele não poderá ser alterado, pois todas as configurações subsequentes passarão a ser incompatíveis.

Nota 1: É necessário segregar todos os dispositivos que emitem dados não solicitados a um ou mais canais que são específicos para a comunicação não solicitada.

Nota 2: Se o encapsulamento Ethernet for usado, certifique-se de configurar seu modo de operação para coincidir com esta configuração.

Para a comunicação *unsolicited*, três parâmetros adicionais deverão ser definidos:

**Receive timeout:** Se uma mensagem completa não for recebida (quer devido a um problema de hardware ou um comando definido incorretamente) o driver irá assumir que o próximo caractere recebido é o começo de outra mensagem



**Dead Time:** O tempo morto é necessário para que o driver possa voltar a se sincronizar com o dispositivo(s) depois de receber uma mensagem com uma chave desconhecida. Se uma mensagem é irreconhecível, o driver não saberá onde a mensagem termina e a próxima começa. A forma de controlar essa situação é receber a mensagem inteira não reconhecida e, então, esperar por algum período de tempo. Este tempo morto deverá ser igual à quantidade de tempo necessária para assumir que o próximo byte recebido é o começo de outra mensagem. Valores razoáveis dependem do dispositivo de destino e deve ser o menor possível. O tempo em milissegundos, entre bytes em uma mensagem é aproximadamente 8000/ baud rate. Visto que o período de tempo morto é sempre iniciado cada vez que um byte é recebido, certifique-se de não definir um valor muito grande: o driver iria ver mensagens individuais como um único fluxo irreconhecível.

**Key length:** O comprimento da chave de transação diz ao driver quantos caracteres usar como chaves de transação. Esses caracteres devem ser os primeiros caracteres em uma mensagem. O protocolo utilizado em um determinado canal deve ser tal de modo que as chaves de mesmo comprimento venham a identificar todas as possíveis mensagens. O comprimento da chave pode variar entre 0 e 32 caracteres.

Antes da definição desses parâmetros, deve-se verificar como o driver lida com os dados não solicitados. Após a recepção de uma mensagem não solicitada, o driver deve determinar o que da transação definida pelo usuário deverá ser usado para a interpretação da mensagem. Para tornar isso possível, o usuário deve associar cada definição de transação com alguma propriedade única de mensagens de um determinado tipo.

Por exemplo, um dispositivo pode relatar mudanças na entrada 1 como

IN01xxxx onde xxxx é um valor de 4 bytes, e mudanças na entrada 2 como IN02xxxx. Neste caso, IN01 (chave) representa a transação que atualiza uma tag INPUT\_1 e IN02 outra que atualiza uma tag INPUT\_2. O driver pode consultar a transação adequada usando os quatro primeiros bytes de qualquer mensagem deste dispositivo, em particular como chaves. Se o protocolo não se presta ao uso de tais chaves, ainda é possível usar este driver especificando um comprimento zero para chave. Mesmo assim, o UCON poderá ser utilizado. Um scanner que envia pacotes começando com os valores de dados brutos seria um exemplo. Nestes casos, o comprimento da chave de transação deve ser definida como zero. Isto irá forçar o driver a usar a primeira transação não solicitada definida no canal para interpretar todos os pacotes de entrada.

### **Nota: O documento do driver fornece mais informações sobre transações não solicitadas e as chaves de transações**

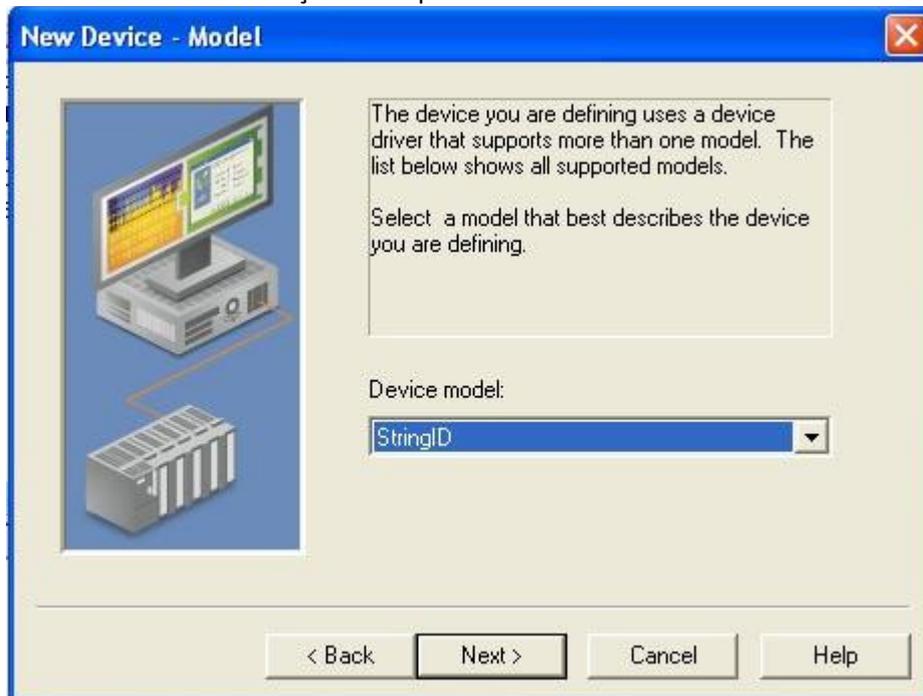
Na janela de configuração mostrada anteriormente, a opção "*log unsolicited message timeout*", quando habilitada, poderá auxiliar no diagnóstico de falhas de comunicação. Nesse caso, mensagens são registradas no Event Log cada vez que o período de recebimento for maior que o tempo limite definido para a recepção de uma mensagem não solicitada. Isso poderá ocorrer devido ao tráfego de rede ou dispositivos de gateway, transações configuradas incorretamente, ou comandos de pausa nas transações.



Outro detalhe importante é a alocação de dispositivos que usam protocolos diferentes em canais separados. É possível misturar protocolos em um canal não solicitado, desde que as chaves de transação possam ter o mesmo comprimento e sejam únicas. Lembre-se também que o modo de canal não pode ser alterado depois que o canal for definido. Certifique-se que o modo de canal a ser selecionado é o correto. Finalmente, os usuários não devem misturar no mesmo canal dispositivos que enviam dados não solicitados com aqueles que trabalham com requisição solicitada.

#### 4. Etapa 2 - Definição de um Device Server

Em seguida, um dispositivo deve ser definido. A definição do ID do dispositivo só terá significado se as transações ou comandos de teste de identificação do dispositivo usar Escreva ID.



#### 5. Etapa 3 - Definição de um perfil de dispositivo

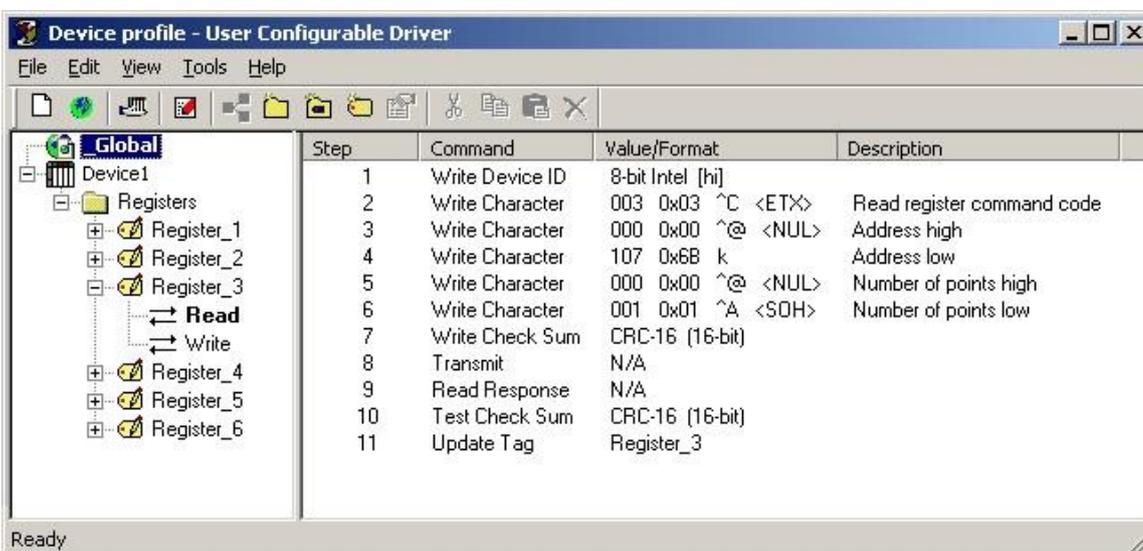
O U-CON Driver (Configurável pelo Usuário) exige que o usuário defina um perfil de dispositivo para cada dispositivo de destino. Um perfil de dispositivo inclui uma definição de cada tag que o DRIVER vai servir, assim como a seqüência de comandos necessários para realizar os pedidos de leitura e escrita. Este trabalho é feito usando o Editor de Transação, que é a interface gráfica do usuário do Driver U-CON (Configurável pelo Usuário).

Para chamar o Editor de transação, clique duas vezes no dispositivo e, em seguida, selecione a guia Editor de Transação.

Em seguida, clique em Iniciar o Editor de Transação.



O Editor de transação pode ser usado para construir grupos de tags e sequências de comandos da transação. Seu perfil definido pelo usuário é mostrado a seguir.



Uma vez que um perfil de dispositivo é criado, as definições de tag e transação podem ser enviados para o servidor clicando Server Update na barra de ferramentas ou no menu principal. Se o Editor de transação é fechado, os



usuários terão a chance de atualizar o servidor. As tags e grupos previamente definidos com o Editor de Transação serão geradas automaticamente no servidor. Lembre-se, as mudanças não foram salvas em arquivo neste momento. Desse modo, salve o projeto do servidor cada vez que um dos perfis de dispositivo é editado. Perfis de dispositivos são uma extensão do projeto de servidor padrão e são salvos como parte do arquivo de projeto de servidor (.OPF).

## 6. Etapa 4 - Testando e depurando a Configuração

Para testar o novo driver criado, inicialmente deve-se conectar o dispositivo (driver criado) à aplicação cliente para validar o processo de leitura e escrita de dados. Se houver algum problema, consulte a janela Diagnosticos, que é uma ferramenta muito útil na depuração do perfil e de falhas de comunicação.

A partir dessa etapa, o projeto piloto pode ser usado. Uma vez que um perfil de DRIVER foi criado e carregado, o Driver U-CON deve funcionar como qualquer outro driver plug-in para o servidor. Mudanças são feitas para o perfil a qualquer momento, basta garantir que o driver esteja *off line* (sem conexão com os aplicativos cliente) e em seguida, chamar o editor de Transação.