

REDES INDUSTRIAIS

Guilherme Moraes Rosa 144450022

Daniel S. Crepalde 134450075

Tadeu Augusto S. Duarte 114450046

CANOPEN

A rede CANopen é uma rede baseada no protocolo CAN, o que significa dizer que ela utiliza telegramas CAN para troca de dados na rede.

Protocolo CAN

O protocolo CAN é um protocolo de comunicação serial que descreve os serviços da camada 2 do modelo ISO/OSI (camada de enlace de dados). Uma rede de área de controle (barramento CAN) é um padrão de barramento de veículo robusto projetado para permitir que microcontroladores e dispositivos se comuniquem entre si em aplicativos sem um computador host. É um protocolo baseado em mensagens, projetado originalmente para fiação elétrica multiplex dentro de automóveis para economizar cobre, mas também é usado em muitos outros contextos.

Barramento CAN

O desenvolvimento do barramento CAN começou em 1983 na Robert Bosch GmbH. O protocolo foi oficialmente lançado em 1986 na conferência da Society of Automotive Engineers (SAE) em Detroit, Michigan. Os primeiros chips controladores CAN, produzidos pela Intel e Philips, surgiram no mercado em 1987. Lançado em 1991, o Mercedes-Benz W140 foi o primeiro veículo de produção a apresentar um sistema de fiação multiplex baseado em CAN.

A Bosch publicou várias versões da especificação CAN e a mais recente é a CAN 2.0 publicada em 1991. Esta especificação tem duas partes; a parte A é para o formato padrão com um identificador de 11 bits e a parte B é para o formato estendido com um identificador de 29 bits. Um dispositivo CAN que usa identificadores de 11 bits é comumente chamado de CAN 2.0A e um dispositivo CAN que usa identificadores de 29 bits é comumente chamado de CAN 2.0B. Esses padrões estão disponíveis gratuitamente na Bosch, juntamente com outras especificações.

Em 1993, a Organização Internacional de Normalização (ISO) divulgou a norma CAN 11898, posteriormente reestruturada em duas partes; ISO 11898-1, que cobre a camada de enlace de dados, e ISO 11898-2, que cobre a camada física CAN para CAN de alta velocidade. A ISO 11898-3 foi lançada mais tarde e cobre a camada física CAN para CAN de baixa velocidade e tolerante a falhas. Os padrões da camada física ISO 11898-2 e ISO 11898-3 não fazem parte da especificação CAN CAN 2.0 da Bosch. Esses padrões podem ser comprados da ISO.

A Bosch ainda está ativa na ampliação dos padrões CAN. Em 2012, a Bosch lançou o CAN FD 1.0 ou CAN com taxa de dados flexível. Esta especificação usa um formato de quadro diferente que permite um comprimento de dados diferente, bem como, opcionalmente, alternar para uma taxa de bits mais rápida após a decisão da arbitragem. O CAN FD é compatível com as redes CAN 2.0 existentes, para que novos dispositivos CAN FD possam coexistir na mesma rede com os dispositivos CAN existentes.

O barramento CAN é um dos cinco protocolos usados no padrão de diagnósticos do veículo de diagnóstico on-board (OBD) -II. O padrão OBD-II é obrigatório para todos os carros e caminhões leves vendidos nos Estados Unidos desde 1996. O padrão EOBD é obrigatório para

todos os veículos a gasolina vendidos na União Europeia desde 2001 e todos os veículos a diesel desde 2004.

O barramento CAN, a camada de enlace de dados do CANopen, só pode transmitir pacotes curtos que consistem em um id de 11 bits, um bit de solicitação de transmissão remota (RTR) e 0 a 8 bytes de dados. O padrão CANopen divide o ID do quadro CAN de 11 bits em um código de função de 4 bits e o ID de nó CANopen de 7 bits. Isso limita o número de dispositivos em uma rede CANopen a 127 (sendo 0 reservado para transmissão). Uma extensão para o padrão de barramento CAN (CAN 2.0 B) permite IDs de quadros estendidos de 29 bits, mas na prática redes CANopen grandes o suficiente para precisar do intervalo de ID estendido raramente são vistas.

No CANopen, o id de 11 bits de um quadro CAN é conhecido como identificador de objeto de comunicação ou COB-ID. No caso de uma colisão de transmissão, a arbitragem de barramento usada no barramento CAN permite que o quadro com o menor ID seja transmitido primeiro e sem atraso. A utilização de um número de código baixo para funções críticas de tempo garante o menor atraso possível.

Rede CANopen

CANopen é um protocolo de comunicação e especificação de perfil de dispositivo para sistemas embarcados usados em automação. Em termos do modelo OSI, o CANopen implementa as camadas acima e inclui a camada de rede. O padrão CANopen consiste em um esquema de endereçamento, vários protocolos de comunicação pequenos e uma camada de aplicação definida por um perfil de dispositivo. Os protocolos de comunicação têm suporte para gerenciamento de rede, monitoramento de dispositivos e comunicação entre nós, incluindo uma camada de transporte simples para segmentação / dessegmentação de mensagens. O protocolo de nível mais baixo que implementa o enlace de dados e as camadas físicas é normalmente a Rede de Área do Controlador (CAN), embora os dispositivos que usam outros meios de comunicação (como Ethernet Powerlink, EtherCAT) também possam implementar o perfil do dispositivo CANopen.

O dispositivo CANopen básico e os perfis de comunicação são fornecidos na especificação CiA 301 lançada pela CAN em Automation. Perfis para dispositivos mais especializados são construídos sobre este perfil básico e são especificados em vários outros padrões liberados pela CAN em Automação, como o CiA 401 para módulos I / O e CiA 402 para controle de movimento.

Modelo do dispositivo

Cada dispositivo CANopen tem que implementar certos recursos padrão em seu software de controle:

- . Uma unidade de comunicação implementa os protocolos para envio de mensagens com os outros nós na rede.

- . Iniciar e redefinir o dispositivo é controlado por meio de uma máquina de estado. Deve conter os estados Inicialização, Pré-operacional, Operacional e Interrompido. As transições entre estados são feitas emitindo um objeto de comunicação de gerenciamento de rede (NMT) para o dispositivo.

. O dicionário de objetos é uma matriz de variáveis com um índice de 16 bits. Além disso, cada variável pode ter um sub índice de 8 bits. As variáveis podem ser usadas para configurar o dispositivo e refletir seu ambiente, ou seja, conter dados de medição.

. A parte de aplicação do dispositivo realmente executa a função desejada do dispositivo, depois que a máquina de estado é definida para o estado operacional. A aplicação é configurada por variáveis no dicionário de objetos e os dados são enviados e recebidos através da camada de comunicação.

Dicionário de objetos

Os dispositivos CANopen devem ter um dicionário de objetos, que é usado para configuração e comunicação com o dispositivo. Uma entrada no dicionário de objetos é definida por:

Index - endereço de 16 bits do objeto no dicionário

Nome do objeto (Tipo / tamanho do objeto) - um tipo simbólico do objeto na entrada, como uma matriz, registro ou variável simples

Nome - uma string que descreve a entrada

Tipo - fornece o tipo de dados da variável (ou o tipo de dados de todas as variáveis de uma matriz)

Atributo - que fornece informações sobre os direitos de acesso para essa entrada, pode ser leitura / gravação, somente leitura ou somente gravação

O campo Obrigatório / Opcional (M / O) - define se um dispositivo em conformidade com a especificação do dispositivo deve implementar este objeto ou não

Os tipos de dados básicos para valores de dicionário de objeto, como booleanos, inteiros e flutuantes, são definidos no padrão (seu tamanho em bits é armazenado opcionalmente na definição de tipo relacionada, intervalo de índice 0x0001–0x001F), bem como tipos de dados compostos, como cadeias de caracteres, matrizes e registros (definidos no intervalo de índice 0x0040–0x025F). Os tipos de dados compostos podem ser sub-indexados com um índice de 8 bits; o valor no subíndice 0 de uma matriz ou registro indica o número de elementos na estrutura de dados e é do tipo UNSIGNED8.

Modelos de comunicação

Diferentes tipos de modelos de comunicação são usados no sistema de mensagens entre nós CANopen.

Mestre / escravo - um nó CANopen é designado como o mestre, que envia ou solicita dados dos escravos. O protocolo NMT é um exemplo de um modelo de comunicação mestre / escravo.

Cliente / servidor - é implementado no protocolo SDO, onde o cliente SDO envia dados (o índice e subíndice de dicionário de objetos) para um servidor SDO, que responde com um ou mais pacotes SDO contendo os dados solicitados (o conteúdo do dicionário de objetos no índice fornecido).

Produtor / consumidor - é usado nos protocolos Heartbeat e Node Guarding. No modelo push de produtor / consumidor, o produtor envia dados ao consumidor sem uma solicitação específica, enquanto no modelo pull, o consumidor precisa solicitar os dados do produtor.

Mensagens

Mensagem de gerenciamento de rede (NMT) - são usados para emitir comandos de alteração de máquina de estado (por exemplo, para iniciar e parar os dispositivos), detectar inicialização de dispositivos remotos e condições de erro. O protocolo de controle do módulo é usado pelo mestre NMT para alterar o estado dos dispositivos. O COB-ID do quadro CAN deste protocolo é sempre 0, o que significa que ele tem um código de função 0 e ID de nó 0, o que significa que cada nó na rede processará essa mensagem. O ID do nó real, para o qual o comando é destinado, é fornecido na parte de dados da mensagem (no segundo byte). Isso também pode ser 0, significando que todos os dispositivos no barramento devem ir para o estado indicado.

Heartbeat - é usado para monitorar os nós na rede e verificar se eles estão ativos. Um produtor de heartbeat (geralmente um dispositivo escravo) envia periodicamente uma mensagem com o código de função binária 1110 e seu ID de nó ($\text{COB-ID} = 0x700 + \text{ID de nó}$). A parte de dados do quadro contém um byte indicando o status do nó. O consumidor de heartbeat lê essas mensagens. Se as mensagens não chegarem dentro de um determinado limite de tempo (definido no dicionário de objetos dos dispositivos), o consumidor poderá tomar medidas para, por exemplo, redefinir o dispositivo ou indicar um erro.

Mensagem de inicialização - os dispositivos CANopen são necessários para fazer a transição do estado Inicializando para Pré-operacional automaticamente durante a inicialização. Quando essa transição é feita, uma única mensagem de pulsação é enviada ao barramento.

Objeto de Dados de Serviço (SDO) - é usado para configurar e ler valores do dicionário de objetos de um dispositivo remoto. O dispositivo cujo dicionário de objetos é acessado é o servidor SDO e o dispositivo que acessa o dispositivo remoto é o cliente SDO. A comunicação é sempre iniciada pelo cliente SDO. Na terminologia CANopen, a comunicação é visualizada a partir do servidor SDO, para que uma leitura de um dicionário de objetos resulte em um upload de SDO e uma gravação em uma entrada de dicionário seja um download de SDO.

Como os valores do dicionário de objetos podem ser maiores que o limite de oito bytes de um quadro CAN, o protocolo SDO implementa a segmentação e a dessegmentação de mensagens mais longas. Na verdade, existem dois desses protocolos: download / upload de SDO e download / upload de SDO Block. A transferência de bloco SDO é uma adição mais recente ao padrão, o que permite que grandes quantidades de dados sejam transferidas com um pouco menos de sobrecarga de protocolo.

Os COB-IDs das respectivas mensagens de transferência SDO do cliente para o servidor e do servidor para o cliente podem ser definidos no dicionário de objetos. Até 128 servidores SDO podem ser configurados no dicionário de objetos nos endereços 0x1200 - 0x127F. Da mesma forma, as conexões de cliente SDO do dispositivo podem ser configuradas com variáveis em 0x1280 - 0x12FF. No entanto, o conjunto de conexão predefinido define um canal SDO que pode ser usado mesmo após a inicialização (no estado pré-operacional) para

configurar o dispositivo. As IDs de COB deste canal são 0x600 + ID de nó para receber e ID de nó 0x580 + para transmissão.

PDO (Process Data Object) - é usado para processar dados em tempo real entre vários nós. Você pode transferir até 8 bytes (64 bits) de dados por um PDO de ou para o dispositivo. Um PDO pode conter várias entradas do dicionário de objetos e os objetos em um PDO são configuráveis usando as entradas de dicionário de objetos de mapeamento e parâmetro.

Existem dois tipos de PDOs: transmitir e receber PDOs (TPDO e RPDO). O primeiro é para dados provenientes do dispositivo (o dispositivo é um produtor de dados) e o segundo é para os dados que vão para o dispositivo (o dispositivo é um consumidor de dados); isto é, com o RPDO você pode enviar dados para o dispositivo e com o TPDO você pode ler os dados do dispositivo. No conjunto de conexão predefinido, há identificadores para quatro (4) TPDOs e quatro (4) RPDOs disponíveis. Com a configuração 512 PDOs são possíveis.

Os PDOs podem ser enviados de forma síncrona ou assíncrona. Os PDOs síncronos são enviados após a mensagem SYNC, enquanto as mensagens assíncronas são enviadas após o acionamento interno ou externo. Por exemplo, você pode fazer uma solicitação para um dispositivo para transmitir TPDO que contém dados necessários, enviando um TPDO vazio com o sinalizador RTR (se o dispositivo estiver configurado para aceitar solicitações TPDO).

Com os RPDOs, você pode, por exemplo, iniciar dois dispositivos simultaneamente. Você só precisa mapear o mesmo RPDO em dois ou mais dispositivos diferentes e certificar-se de que esses RPDOs sejam mapeados com o mesmo COB-ID.

Synchronization Object (SYNC) - O Sync-Producer fornece o sinal de sincronização para o Sync-Consumer. Quando o Sync-Consumer recebe o sinal, eles começam a executar suas tarefas síncronas.

Em geral, a fixação do tempo de transmissão de mensagens PDO síncronas, juntamente com a periodicidade de transmissão do objeto Sync, garante que os dispositivos sensores podem organizar variáveis de processo de amostra e que dispositivos atuadores podem aplicar sua atuação de maneira coordenada. O identificador do objeto Sync está disponível no índice 1005h.

Time Stamp Object (TIME) - Normalmente, o objeto Carimbo de data / hora representa um tempo como um campo de 6 bytes: uma contagem de milissegundos após a meia-noite (no máximo 27 bits, armazenados em um campo de 32 bits) e um número não assinado de 16 bits de dias desde 1º de janeiro 1984. (Isso vai transbordar em 7 de junho de 2163.)

Algumas aplicações críticas de tempo, especialmente em grandes redes com taxas de transmissão reduzidas, exigem uma sincronização muito precisa; pode ser necessário sincronizar os relógios locais com uma precisão na ordem de microssegundos. Isto é conseguido usando o protocolo opcional de sincronização de alta resolução que emprega uma forma especial de mensagem de timestamp para ajustar a deriva inevitável dos relógios locais.

O timestamp de alta resolução é codificado como unsigned32 com uma resolução de 1 microssegundo, o que significa que o contador de tempo reinicia a cada 72 minutos. Ele é configurado mapeando o registro de data e hora de alta resolução (objeto 1013h) em um PDO.

Mensagem de Objeto de Emergência (EMCY) - As mensagens de emergência são acionadas pela ocorrência de uma situação de erro fatal interno do dispositivo e são transmitidas do dispositivo de aplicativo em questão para os outros dispositivos com alta prioridade. Isso os torna adequados para alertas de erro de tipo de interrupção. Um Telegrama de Emergência pode ser enviado apenas uma vez por "evento de erro", ou seja, as mensagens de emergência não devem ser repetidas. Enquanto nenhum novo erro ocorrer em um dispositivo, nenhuma outra mensagem de emergência deve ser enviada. Por meio do perfil de comunicação CANopen, os códigos de erro de emergência definidos, o registro de erro e as informações adicionais específicas do dispositivo são especificados nos perfis do dispositivo.

Aplicações

Indústria automobilística - O automóvel moderno pode ter até 70 unidades de controle eletrônico (ECU) para vários subsistemas. Normalmente, o maior processador é a unidade de controle do motor. Outros são usados para transmissão, airbags, ABS, cruise control, sistemas de áudio, janelas elétricas, portas, ajuste de espelhos, sistemas de bateria e recarga para carros híbridos / elétricos, etc. Alguns deles formam subsistemas independentes, mas as comunicações entre outros são essenciais. Um subsistema pode precisar controlar os atuadores ou receber feedback dos sensores. O padrão CAN foi concebido para preencher essa necessidade. Uma das principais vantagens é que a interconexão entre diferentes sistemas de veículos pode permitir que uma ampla gama de recursos de segurança, economia e conveniência sejam implementados usando apenas software - funcionalidade que acrescentaria custo e complexidade se tais recursos fossem "hard wired" usando sistemas elétricos tradicionais.

Exemplos incluem:

Partida / parada automática: Várias entradas de sensor ao redor do veículo (sensores de velocidade, ângulo de direção, ativação / desativação do ar condicionado, temperatura do motor) são agrupadas através do barramento CAN para determinar se o motor pode ser desligado quando parado para melhorar a economia de combustível e emissões.

Freios de estacionamento elétricos: A funcionalidade "hill hold" recebe informações do sensor de inclinação do veículo (também usado pelo alarme) e dos sensores de velocidade (também usados pelo ABS, controle do motor e controle de tração) através do barramento CAN para determinar se o veículo é parado em um declive. Da mesma forma, as entradas dos sensores do cinto de segurança (parte dos controles do airbag) são alimentadas pelo barramento CAN para determinar se os cintos de segurança estão apertados, para que o freio de estacionamento seja liberado automaticamente ao se deslocar.

Sistemas de assistência ao estacionamento: quando o motorista engata marcha à ré, a unidade de controle da transmissão pode enviar um sinal via barramento CAN para ativar o sistema de sensor de estacionamento e o módulo de controle de porta do espelho lateral do passageiro para inclinar para baixo para mostrar a posição do calçada. O barramento CAN também recebe entradas do sensor de chuva para acionar o limpador de pára-brisa traseiro ao inverter.

Auto lane assist / colision avoidance systems: As entradas dos sensores de estacionamento também são usadas pelo barramento CAN para alimentar dados de proximidade externa para sistemas de assistência ao motorista, como aviso de partida de faixa

e, mais recentemente, esses sinais viajam através do barramento CAN para acionar o freio por fio em sistemas de prevenção de colisão ativos.

Limpeza do freio automático: A entrada é feita do sensor de chuva (usado principalmente para os limpadores automáticos do pára-brisa) através do barramento CAN ao módulo ABS para iniciar uma aplicação imperceptível dos freios enquanto dirige para limpar a umidade dos rotores do freio. Alguns modelos Audi e BMW de alto desempenho incorporam esse recurso.

Outras aplicações

. O protocolo de barramento CAN foi usado no sistema de câmbio eletrônico Shimano DI2 para bicicletas de estrada desde 2009, e também é usado pelos sistemas Ansmann e BionX em seu motor de acionamento direto.

. O barramento CAN também é usado como fieldbus em ambientes gerais de automação, principalmente devido ao baixo custo de alguns controladores e processadores CAN.

. Os fabricantes, incluindo o NISMO, pretendem utilizar os dados do bus CAN para recriar as corridas reais no videogame Gran Turismo 6, utilizando a função GPS Data Logger, que permitiria aos jogadores correr contra voltas reais.

. O MPL (Modular Prosthetic Limb) do Laboratório de Física Aplicada da Universidade Johns Hopkins usa um barramento CAN local para facilitar a comunicação entre servos e microcontroladores no braço protético.

. As equipes da FIRST Robotics Competition usam amplamente o barramento CAN para se comunicar entre o roboRIO e outros módulos de controle do robô.