

## Aula prática 4

### Estudo do barramento CAN

#### Objectivos

- Familiarização com o protocolo CAN
- Transmissão e recepção de tramas CAN em micro-controladores PIC
- Observação dos mecanismos de prioridade do CAN

#### Descrição

O protocolo CAN (*Controller Area Network*) foi desenvolvido em meados dos anos 80 pela Robert Bosch GmbH, para aplicações automóveis. Este protocolo apresenta uma arquitectura *multi-master*, operando habitualmente sobre par entrançado. A transmissão é efectuada em *broadcast*, sendo a decisão de aceitação tomada ao nível do nó receptor. A codificação dos bits é NRZ (*Non return to zero*), sendo a sincronização efectuada por uma técnica de *bit stuffing* (por cada 5 bits consecutivos de igual polaridade é adicionado um bit extra de polaridade inversa). O comprimento máximo da rede depende da velocidade de transmissão (e.g. 40m @ 1 Mbps; 1300m @ 50 Kbps).

O mecanismo de arbitragem empregue pelo protocolo CAN é baseado em CSMA com arbitragem bit-a-bit não destrutiva. Quando um nó pretende transmitir uma mensagem, tem primeiro de aguardar que o barramento fique livre. Depois inicia a transmissão, bit-a-bit, e observa simultaneamente o estado do barramento, retirando-se do processo de arbitragem logo que o nível observado seja diferente do transmitido. Este mecanismo assegura que quando mais que um nó inicia a transmissão simultaneamente o que tiver o ID com maior prioridade tem sucesso na transmissão. Esta propriedade permite, em certas condições, obter garantias quanto à latência máxima associada a cada mensagem, tornando este protocolo apetecível para aplicações industriais com requisitos tempo-real.

#### Procedimentos

1. Descarregar o ficheiro disponível na página da disciplina relativo ao tutorial CAN.  
Analisar o código e observar as primitivas de configuração, envio e recepção de tramas CAN.  
Prestar especial atenção à configuração dos registos de máscara e filtro.
2. Compilar e testar o programa. Para isso:

- i) Crie uma pasta no drive “E:” (Atenção: o nome desta pasta não deve conter caracteres especiais e/ou espaços) e coloque os ficheiros que descarregou nessa pasta.
  - ii) Usando o Hi-Tide criar um novo projecto que inclua os ficheiros “can.c” e “send.c”. Alterar a linha “#define GID 0x#” de forma a que “#” corresponda ao seu grupo.
  - iv) Compilar o programa e descarrega-lo para a placa “DETPIC”. Ligar esta à dos colegas do lado. Ligue um *led* à ficha de expansão como indicado abaixo e observar o comportamento do programa.
3. Ainda em conjunto com os colegas do lado altere o programa de forma a que apenas uma das placas envie periodicamente uma trama CAN. Observe o conteúdo da trama no osciloscópio:
- Quando o campo de dados tem um conteúdo fixo, com 16 bits;
  - Quando o campo de dados contém um contador de 16 bit, incrementado de uma unidade em cada trama enviada;
  - Explique o comportamento observado.
4. Crie um programa que envie a cada 5ms uma trama CAN, com 8 bytes de dados e identificador na forma 0x70N, N={1,2,..8}. Por cada 5 tramas enviada o *led* deve comutar de estado.
- Coloque no campo ID (N) o número do seu grupo e interligue todos os nodos. Comparar a taxa a que o *led* pisca nos vários grupos.
  - Rodar os identificadores com os outros grupos e observar eventuais alterações no comportamento;
  - Explique o comportamento observado.

