

Realization of an application under μ C/OSII (1)

1 Purpose

- Study the possibilities of the CAN network
- Create a simple application sending and receiving messages

Useful documents: see folder "Useful Documents"

- Datasheet of the dsPIC "Section 21 - Enhanced Controller Area Network.pdf"
- Table of conversion of the ASCII codes "ASCII.pdf"

2 Work

For this laboratory, each group gets an Explorer 16 board equipped with a dsPIC33 and an extension board with a keyboard and a DB9 connector for the CAN network. Plug the yellow twisted pair in this connector to be connected to the CAN network.

Your work consists in programming the CAN peripheral of the dsPIC to send a message, then to receive messages transmitted continuously by a PC of the laboratory.

1) Read the following useful information in the datasheet of CAN:

- the review on pages 2-5
- page 35 on transmission
- pages 41 to 43 on reception

As you can see on figure 21-1, the dsPIC contains 32 buffers for transmission / reception, which can be configured either in reception, or in transmission. These buffers are not directly in the CAN peripheral. [Explain which mechanism enables to access to the CAN peripheral and what are the advantages of this mechanism.](#)

The buffers for reception use a particular mechanism to filter the identifiers of the messages. Try to explain this mechanism. [Which are their advantages and disadvantages?](#)

2) Open the project named **OSCan**.

A library of functions has been developed to facilitate the use of the CAN peripheral (see files CanDspic.c and CanDspic.h).

3) Open and [examine the file CanDspic.h](#). Observe the structure that has been set for the CAN buffers. Refer to section 21.4 of the pdf for more details.

4) In the function main(), configure the CAN peripheral for a bits rate of 500kbps. An initializing function allows you to configure automatically the first 7 buffers in reception and the 8th buffer in transmission.

5) [Send continuously a first CAN message on the network \(at a rate of 1s\)](#). This message must contain an ASCII string of maximum 8 characters of your choice. For tables x_a , choose $(0x180 + x)$ as an identifier. For tables x_b , choose $(0x190 + x)$ as identifier. To check that your message is correctly sent, one of the nodes of the lab has been programmed to monitor and display all the messages passing on the bus.

6) You shall now receive a messages transmitted on the network. A PC node has been programmed to send permanently 4 messages on the network. Your goal is to find which are these messages and their

contents. The reception of the messages should be done by interruption. You will find a skeleton of the interrupt service routine in CANRxInterrupt.c.

- [configure the reception of CAN messages and activate the interruptions in main\(\)](#)
- [write the interrupt service allowing to find a first message. Don't forget to use OsIntEnter\(\) and OSIntExit\(\) \(see reference manual pp. 29 – 31\).](#)
- [modify your code to receive the other messages.](#) Once you have received the 4 messages, decode them using the ASCII table and follow the instructions they contain.