

Exercise 7CAN_2 - Transmitting and Receiving a Standard CAN Message with the CAN Modules 1 and 2*

❑ Objective:

- Generate one Standard CAN (11-bit-Identifier) Message with the on-chip CAN Module 1
 - Use Message Object 1
 - Use Identifier 0x011
 - Use 4 Data Bytes containing the data 0x0a, 0x0b, 0x0c, 0x0d
- Receive this Standard CAN Message with the on-chip CAN Module 2
 - Use Message Object 1
 - Use Identifier 0x011
- Toggle Port 2.0 when Message Object 1 of CAN Module 2 receives the Message generated by CAN Module 1
- Use the Flash Tools and burn the program into the on-board FLASH



❑ You will need to connect the two CAN Interfaces with a serial interface cable - both ends of the cable need to be female

* This exercise is possible with the C167CS only

Exercise 7CAN_2 - DAvE Configurations

- ☐ **Start DAvE 2.0**



- ☐ **Select “Create a new project” from the Startup Dialog or click** 
- ☐ **Select the microcontroller C167CS* and click “Create”**
(if this microcontroller is not on the list, you need to re-install it from the DAvE 2.0 CD ROM*)
- ☐ **DAvE will create the project**
- ☐ **Save your project by selecting “File | Save” or press** 
 - Browse to directory “c:\hot167_2\7can_2\”
 - Enter project name: “7CAN_2”
 - Click “Save”
- ☐ **You will see the C167CS block diagram and the Project Settings Window (configuration see next slide)**
- ☐ **To get back to the Project Settings window in case you close it: Select “File | Project Settings”**

* C167CS not yet supported by DAvE 2.0 CD ROM. See “Hints regarding DAvE.”

Exercise 7CAN_2 - DAvE Configurations (cont.)

❑ Project Settings:

- General:
 - Select Tasking Compiler, SMALL model
- System Clock:
 - External Oscillator Frequency: Set to 5 MHz
- Startup Configuration:
 - Bus Type after Reset: Set to 16 bit DEMUX
 - Write Configuration: Pin #WR and #BHE operates as #WRL and #WRH

Exercise 7CAN_2 - DAvE Configurations (cont.)

❑ **Configure CAN Module:**

- Because DAvE 2.0 does not yet support the C167CS with the on-chip CAN Module 2, the files CAN1.C, CAN2.C, CAN1.H, CAN2.H are included in the directory c:\hot167_2\7can_2
- The files contain the same code as if it would have been generated by DAvE

❑ **Configure Port 2:**

- Port 2:
 - Enable P2.0 / CC0IO as general IO / Out
- Functions:
 - Include Port Initialization Function IO_vInit

Exercise 7CAN_2 - DAvE Configurations (cont.)

❑ **Configure GPT1:**

- Auxiliary Timer 2:
 - Timer 2 Mode: Reload Timer 3 with T2
 - Input Selection: Any transition of T3OTL
 - T2 Register: Set to 0x4C4B
- Core Timer 3
 - Up/Down Control: Count down
 - Timer Start: Start T3 after initialization (this will generate a 500ms period of T3)
 - T3 Register: Set to 0x4C4B
 - Input Selection/Prescaler: Fcpu/512
 - Alternate Output Function: enable alternate output function (use T3OUT)
 - Interrupt Control: Enable T3 interrupt
- Functions:
 - Include GPT1 initialization function GT1_vInit
- Interrupts:
 - Place interrupt of T3 in level 14, group 1 of the Interrupt Vector Table

Exercise 7CAN_2 - DAvE Configurations (cont.)

☐ **Configure External Bus Controller:**

- BUSCON 0 (Flash):
 - Memory Cycles Time Control: 1 waitstate
 - Memory Tristate Control: No Waitstate
 - ALE Lengthening Control: Normal ALE Signal
- BUSCON 1 (RAM):
 - Enable external bus
 - Bus Configuration (BTYP): 16 bit DMUX
 - Memory Cycles Time Control: 1 waitstate
 - Memory Tristate Control: No Waitstate
 - Address Area (ADDRSEL 1):
 - Set Windows Size to 64 Kbyte
 - Set Required start address (A23..A12) to 0x040
- Note: These settings have no effect until Tasking EDE can read the '*.dpt' and '*.asm' files generated by DAvE 2.0.
Some special EDE settings are necessary (see below.)

☐ **Generate Code ()**

☐ **DAvE will show you all the files that he has generated (File Viewer is opened automatically)**

Exercise 7CAN_2 - EDE Configurations

- ❑ **Start Tasking EDE for C166**
- ❑ **Create new Project (Project | New):**
 - browse to directory c:\hot167_2\7can_2\
 - enter file name: 7can_2
 - click “save”, then verify the path and click “OK”
- ❑ **Edit Project:**
(window Project | Properties | Files opens up automatically)
 - Add all C-files (from c:\hot167_2\7can_2)
 - Add START.ASM (from c:\dc166\lib\src\)
 - Click “OK”
- ❑ **Load Files (Project | Load Files):**
 - Click “Invert” | “OK”
- ❑ **Select CPU (EDE | CPU Options):**
 - Select CPU type C167CR or C167CS, click “OK”

Exercise 7CAN_2 - EDE Configurations (cont.)

☐ **Select Output Format (EDE | Linker/Locator Options):**

- Format tab:
 - Select Intel HEX Format (Intel hex8 records)

☐ **Configure ROM / RAM Areas (EDE | Linker/Locator Options)**

- Memory tab:
 - ROM area: Enter 0h-7FFFh
 - RAM area: Enter 40000h-4FFFFh

☐ **Define C167 Symbol (EDE | Macro Preprocessor | Project Options):**

- Macros: Enter C167

Exercise 7CAN_2 - EDE Configurations (cont.)

❑ Edit MAIN.C (Project_Init()-function, Main()-function):

- Edit Project_Init()-function:

```
// USER CODE BEGIN (Project_Init,1)
CAN1_vInit(); // initialize CAN module 1
CAN2_vInit(); // initialize CAN module 2
// USER CODE END
```

- Edit Main()-function:

```
// USER CODE BEGIN (Main,2)
while(1);
// USER CODE END
```

Exercise 7CAN_2 - EDE Configurations (cont.)

❑ Edit MAIN.H (File | Open, enter Filename 'main.h', click 'Open')

- Edit Project Includes:

```
// USER CODE BEGIN (MainHeader,1)
#include "CAN1.H"
#include "CAN2.H"
// USER CODE END
```

❑ Edit GPT1.C (GT1_vilsrTmr3()-function):

```
// USER CODE BEGIN (GT1_IsrTmr3,1)
CAN1_vTransmit(1);
// USER CODE END
```

❑ Edit CAN2.C:

- Edit CAN2_vilsr()-function:

```
// USER CODE BEGIN (CAN_IsrRxOk,1)
P2_0 = ! P2_0; // Toggle pin
// USER CODE END
```

Exercise 7CAN_2 - EDE Configurations (cont.)

- ❑ **Set / verify SYSCON / BUSCON / ADDRSEL / XPERCON Register (EDE | CPU Options):**
 - SYSCON tab:
 - Select Enable XBUS Peripherals
 - XPERCON tab:
 - Select Enable on-chip CAN module 1
 - Select Enable on-chip CAN module 2
 - Select Enable on-chip 2K XRAM memory module
 - BUSCON0 tab (for external Flash Memory on the Starter Kit board):
 - Select Enable External Bus
 - Memory Cycle Time: Select one wait state
 - BUSCON1 tab (for external RAM Memory on the Starter Kit board):
 - Select Enable Address Window
 - Memory Cycle Time: Select one wait state
 - External Bus Configuration: Select 16 bit de-multiplexed
 - ADDRSEL1 tab (for external RAM Memory on the SK board):
 - Range Size: select 64KB window
 - Range Start Address: enter 040h for bits A23...A12 (A11...A0 are not relevant (0) => RAM starts at address 40000h)

Exercise 7CAN_2 - EDE Configurations (cont.)

☐ **Build Project (Project | Build)** 

Exercise 7CAN_2 - Using the Flash Tool

- ❑ **To demonstrate the use of the Flashtool that is included on the Starter Kit CD ROM, the output HEX-file shall now be programmed to the flash memory.**
- ❑ **Step by Step to use the Flashtool:**
 - Connect your evaluation board to a power supply and the PC
 - Put blue DIP switch #1 to the ON position
 - Older Starter kits: close pins 1+2 of jumper 2 (usually red)
 - Press reset button S1 on kitCON-167
 - Run flasht.exe in ..\cdrom\startkit\sk_167\flash\
 - Press (2) to erase entire Flash-Area and (Y) for yes
 - Press (4) - Load INTEL-Hexfile to download the Hex-code to the evaluation board
 - Press F2 and enter the complete path of the hex-file (.hex):
c:\hot167_2\7can_2\7can_2.hex
 - Press F1 and (Y) to exit Flashtool

Exercise 7CAN_2 - Running the Program

- ❑ **Connect the two CAN Interfaces with a serial interface cable - both ends of the cable need to be female**

- ❑ **Start the application by**
 - switching DIP Switch 1 to OFF
(older boards: remove red jumper JP2),
 - pushing the reset button S1.

- ❑ **Program Verification:**
 - The toggling of LED D20 indicates that the Port 2.0 toggles
 - The Interrupt Service Routine of the CAN Module 2 is entered every time the LED changes its state
 - This means that the CAN Module 2 receives the CAN Message that was sent by CAN Module 1
 - If available, use an oscilloscope and/or a CAN analyzer to monitor the bus activity