

Exercise 7CAN_1 - Transmitting a Standard CAN Message with the CAN Module

❑ Objective:

- Generate one Standard CAN (11-bit-Identifier) Message with the on-chip CAN Module*
- Evaluate every identifier bit of incoming messages
- Use maximum bus speed of 1 Mbaud
- Use Message Object 1
- Use Identifier 0x123
- Use 8 Data Bytes containing the data 0x00, 0x11, ..., 0x77.



* Hints:

As long as no receiving node is connected to the kitCON-167, the C167CR will not receive an acknowledge for its transmission and therefore will keep on trying to transmit the CAN message “forever”. Connect another Starter Kit with CAN Capability or a CAN Analyzer to the CAN bus to generate real CAN data transfers.

Exercise 7CAN_1 - DAvE Configurations

- ☐ **Start DAvE 2.0**



- ☐ **Select “Create a new project” from the Startup Dialog or click** 
- ☐ **Select the microcontroller C167CR/CS* 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_1\7can_1\”
 - Enter project name: “7CAN_1”
 - Click “Save”
- ☐ **You will see the C167CR/CS 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_1 - DAvE Configurations (cont.)

❑ Project Settings:

- General:
 - Select Keil 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_1 - DAvE Configurations (cont.)

☐ **Configure CAN Module:**

- Control:
 - Global Mask Register: 0x7FF
(evaluate every bit of incoming identifiers)
- Baudrate:
 - Baudrate: Enter desired Baudrate: 1000 Kbaud
- Objects:
 - Configure Object 5:
 - Enable Control: Enable Message Object
 - Identifier Selection: Standard 11 bit
 - Message Direction: Transmit Data Frames
 - Data Length Code: Select 8 Data Bytes
 - Data fields: Enter Data Bytes (0x00, 0x11, ..., 0x77)
 - Arbitration Register: Enter Identifier 11-bit: 0x123
 - Close
- Functions: Include functions CAN_vInit and CAN_vTransmit

☐ **Generate Code ()**

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

Exercise 7CAN_1 - µVision2 Configurations

- ❑ **Start Keil µVision2**
- ❑ **Create new Project: '7can_1.uv2'**
 - Go to Project | New Project
 - Browse to directory c:\hot167_1\7can_1\
 - Enter file name: 7can_1.uv2
- ❑ **Select Target Device:**
 - Go to Project | Select Device for Target 'Target 1'
 - Double-click 'Siemens'
 - Select 'C167CR-LM' or 'C167CS-32FM'
 - Click 'OK'

Exercise 7CAN_1 - µVision2 Configurations (cont.)

☐ **Add Files:**

- Click the right mouse button on 'Source Group 1' in the Project Window on the left hand side
- Click 'Add Files to Group 'Source Group 1''
- Select all C files and click 'add' (they appear in the Project Window)
- (Optional: Add your assembler startup file)
- Click 'Close'
- Double-click all files in the Project Window to open them

☐ **Select Target Options:**

Go to Project | Options for Target 'Target 1'

- Select External Memory configuration:
 - Go to '**Target**' tab
 - If available, uncheck the box to use / allocate the On-chip (Code) ROM
 - External Memory #1: RAM: Start 0x4000, Size 0x3FFF
 - External Memory #2: ROM: Start 0x1000, Size 0x2FFF
(those are example values and they need to be adapted
e.g. for programming the code into the on-board Flash memory)

Exercise 7CAN_1 - µVision2 Configurations (cont.)

- Select Linker Options:
 - Go to '**L166 Misc**' tab
 - Interrupt Vector Table Address: Enter 0x0
 - Reserve: Enter 08H-0BH, 0ACH-0AFH
(to reserve the Interrupt vector locations for NMI (Non-Maskable Interrupt) and Serial Interface Receive Interrupt)
- Set Debug Options:
 - Go to '**Debug**' tab
 - Click 'Use Keil Monitor-166 Driver' (upper right hand corner)
 - Click 'Load Application at Startup' and 'Go till main()'
 - Click 'Settings' (upper right hand corner)
 - Monitor configuration: select 'Phytec KC167'
 - Stop Program Execution with: select 'Serial Interrupt or NMI'
 - Click 'OK'
- Click 'OK' to close the Target Option window

Exercise 7CAN_1 - µVision2 Configurations (cont.)

❑ Edit MAIN.C (main()-function):

- Transmit Message Object 5
- Include endless loop:

```
// USER CODE BEGIN (Main,2)
    CAN_vTransmit(5);
    while(1) {};
// USER CODE END
```


Exercise 7CAN_1 - Running the Program

☐ **Reset Target Hardware (Press Reset Button on Starter Kit)**

☐ **Build Project (Project | Rebuild Target, or click )**

☐ **Run integrated Debugger from within μ Vision2**



- Debug | Start / Stop Debug Session (click 'OK' when prompted)
- The Debugger will load the Keil Monitor into the kitCON-167's RAM via bootstrap loader
- Object file c:\hot167_1\7can_1\7can_1 will be loaded automatically and the debugger will go to main().

☐ **Go! (Debug | Go)**

☐ **Program Verification: Connect Scope to**

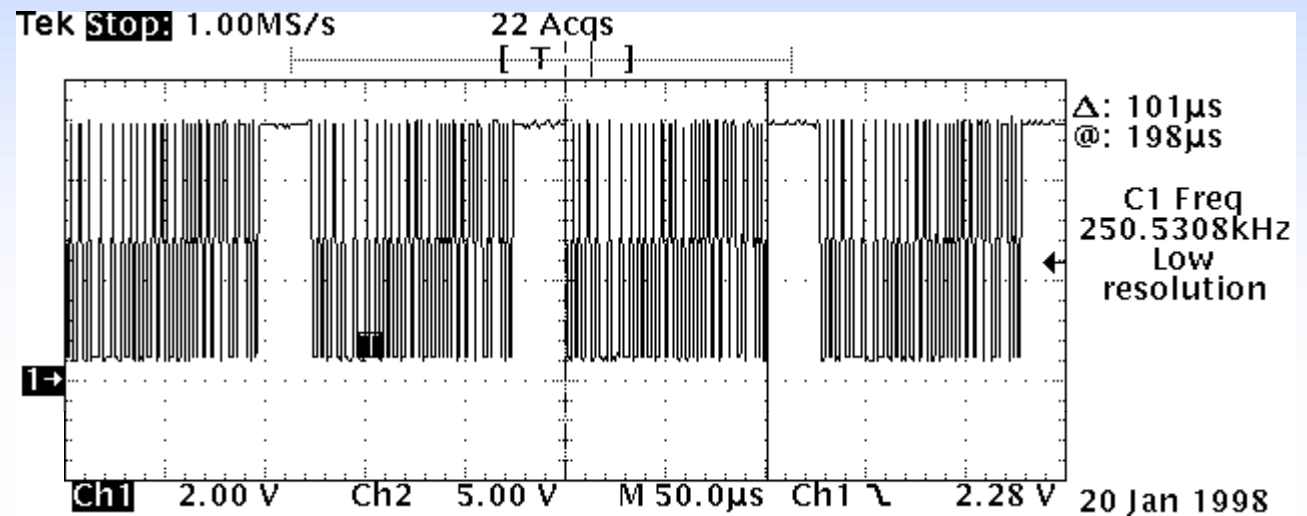
- P4.6 / CAN_TxD (connector X3 pin 40, called A22/TXDC)

Exercise 7CAN_1 - Screenshot: CAN Message

50.0 μ s / Div

no
acknowledge
received

P4.6 / CAN_TxD



10.0 μ s / Div

1 us per bit
= 1 Mbit/s

P4.6 / CAN_TxD

