

**Infineon**

# ***XC886/888CLM***

## ***Getting Started on MCAN - ADC***

V1.0 May 2006

# DAvE Setup and Code Generation

---

We will use DAvE to create a project to demonstrate the feature of interrupt, MCAN, and ADC.

ADC will be configured to the parallel conversion mode with autoscan.

ADC will constantly scan the analog input channel 7 (thru P2.7). Interrupt will be generated every time the conversion is done. In the interrupt routine, the conversion result will be transmitted thru CAN NODE1.

CAN NODE0 is configured with Message Object 16, to be able to received any frame in the CAN Bus. When the valid frame is received, the interrupt will be generated. In the interrupt routine, the received data will be send to PORT3.

CAN NODE1 is configured with Message Object 1, to be able to transmit a frame, which consist of 1 data byte (the result of ADC conversion).



DAvE

# DAvE Setup and Code Generation

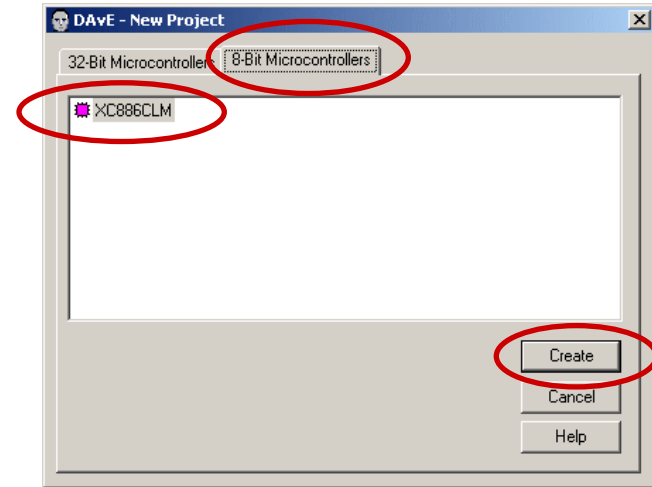
- Step1: Creating the new project

*File → New*

- Step2: Select 8-bit Microcontroller

*XC886CLM*

and then click the *“Create”* button.

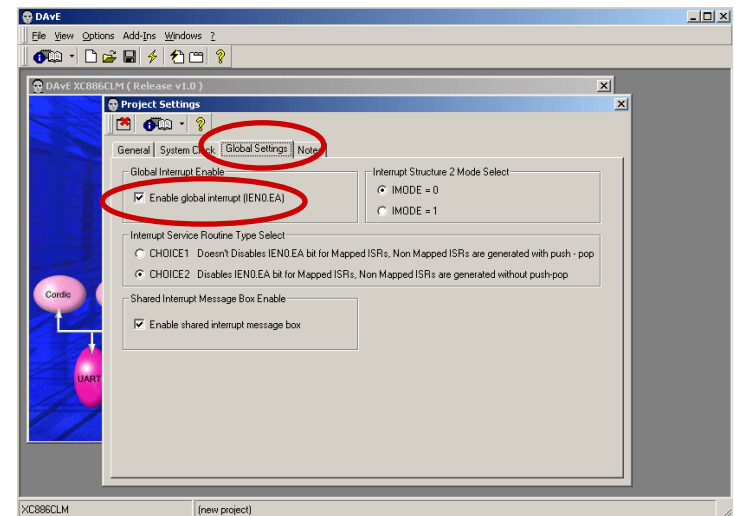


- Step3: Inspecting General Settings

Click the *“Global Setting”* tab.  
Select the *“Enable Global Interrupt”* checkbox.

(If it is not yet selected)

Close the interface by clicking



stop thinking  
never

# DAvE Setup and Code Generation

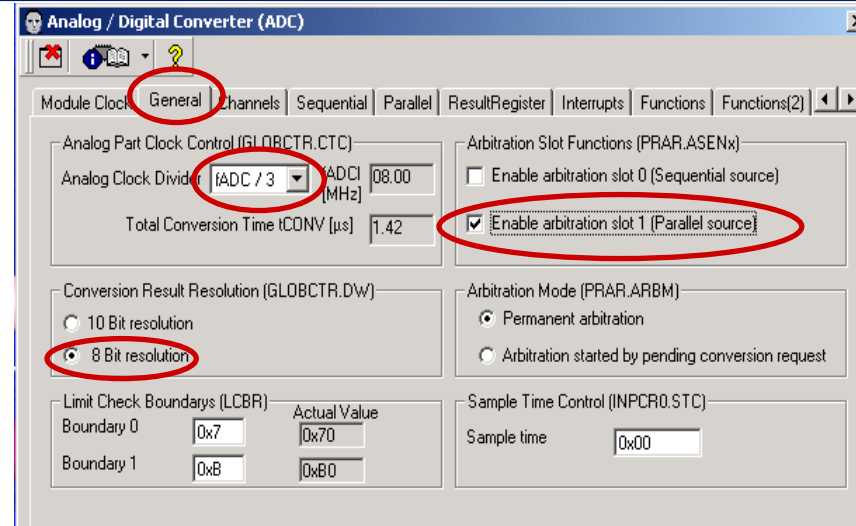
## Step 4.1: Configuring ADC Module

Click “**ADC**” bubble.



On the ADC menu that appears, Click “**General**” tab and do the following:

- ) In the Analog Clock Divider field:  
Select “**fADC / 3**”
- ) Select “**8 bit resolution**”
- ) Select “**Enable arbitration slot 1**”



## Step 4.2: Configuring ADC Module

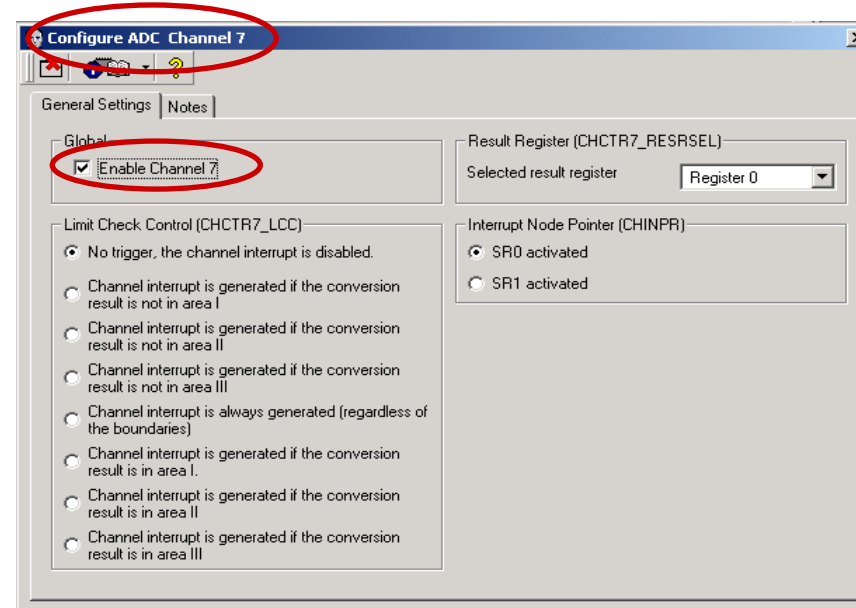
Click “**Channels**” tab.

Click “**Configure Channel 7**” button.

New pop up window will appear.

Select “**Enable Channel 7**”.

Close the new pop up window by clicking



# DAvE Setup and Code Generation

## Step 4.3: Configuring ADC Module

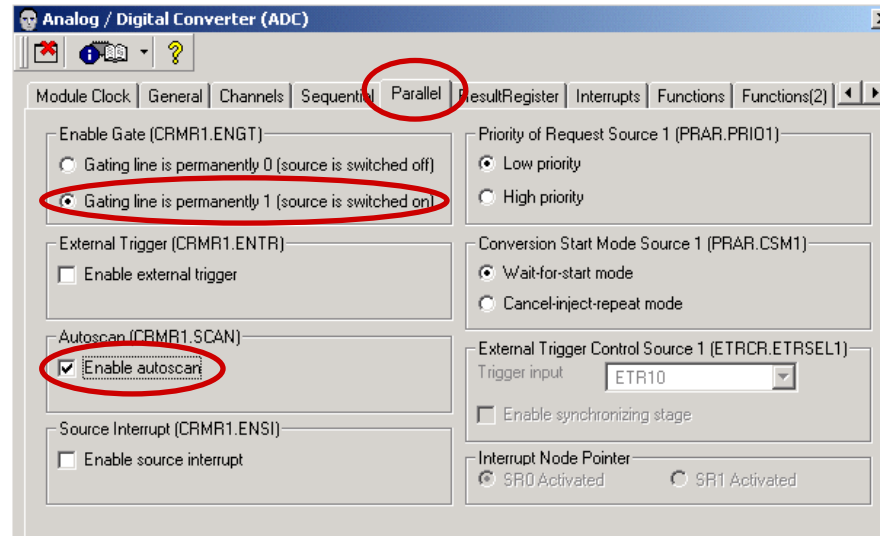
Click "**Parallel**" tab.

Select the following:

**"Gating line is permanently 1"**

and

**"Enable autoscan"**



## Step 4.4: Configuring ADC Module

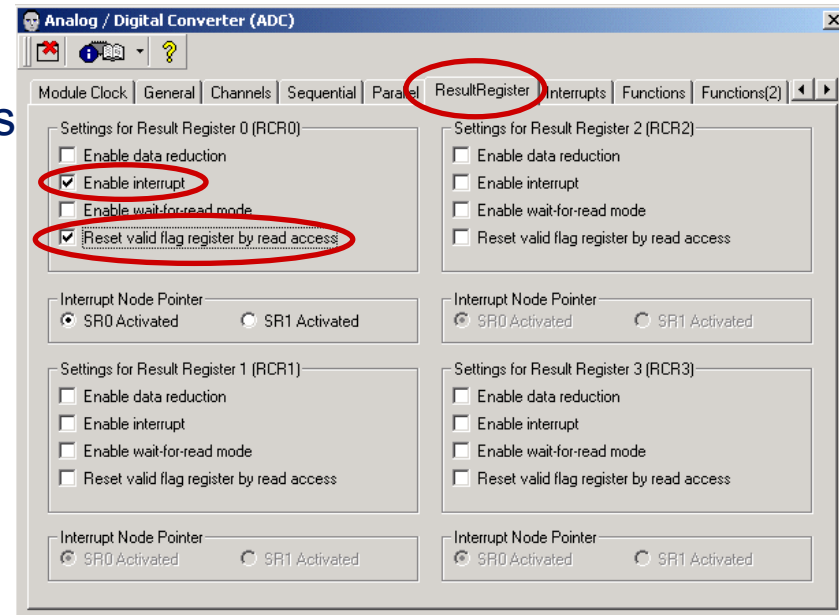
Click "**Result Register**" tab.

Select the following under the "Settings for Result Register 0 (RCR0)"

**"Enable interrupt"**

and

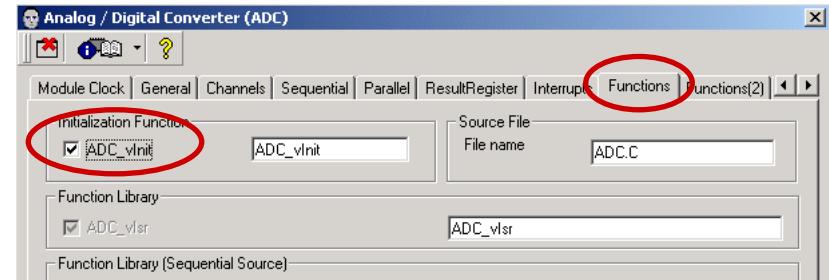
**"Reset valid flag register by read access"**



# DAvE Setup and Code Generation

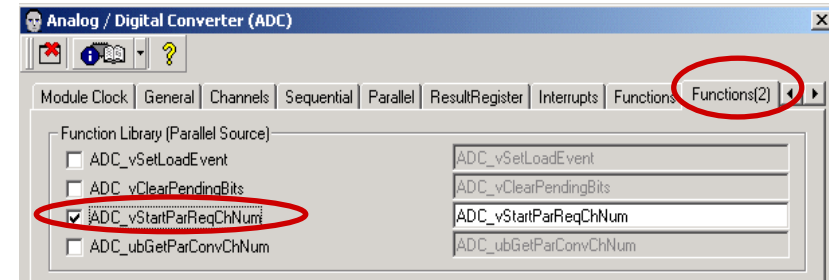
## Step 4.5: Configuring ADC Module

Click “**Functions**” tab.  
 Select the following:  
 “**ADC\_vInit**”



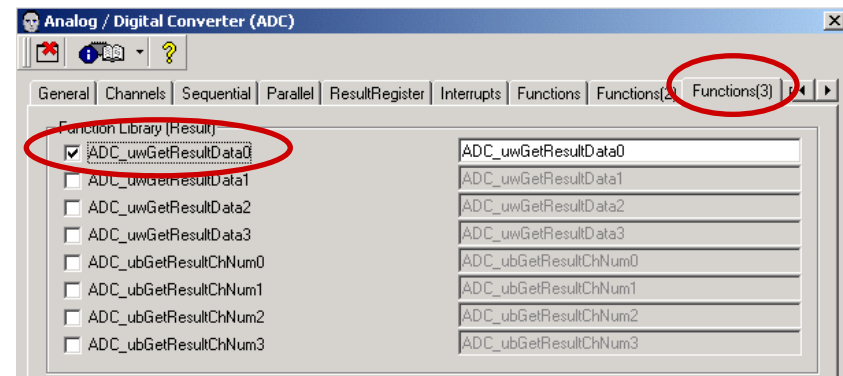
## Step 4.6: Configuring ADC Module

Click “**Function(2)**” tab.  
 Select the following:  
 “**ADC\_vStartParReqChNum**”



## Step 4.7: Configuring ADC Module

Click “**Function(3)**” tab.  
 Select the following:  
 “**ADC\_uwGetResultData0**”



Now, close the window by clicking



# DAvE Setup and Code Generation

---

## ■ Step 5.1: Configuring MCAN Module

Click "**Multi CAN**" bubble.



DAvE

The next few slides will show the MCAN configuration step by step from:

Configuring NODE0

Configuring NODE1

Configuring LIST

Configuring Message Object (MO) – MO1 and MO16



# DAvE Setup and Code Generation

## ■ Step 5.2: Configuring MCAN NODE 0

On the MultiCAN menu that appears,

Click “**Nodes**” tab, click the “**Node 0**” button, and do the following:

-) Click “**Control**” tab and select the following options:

**“Enable the Loop-Back mode (LBM)”**

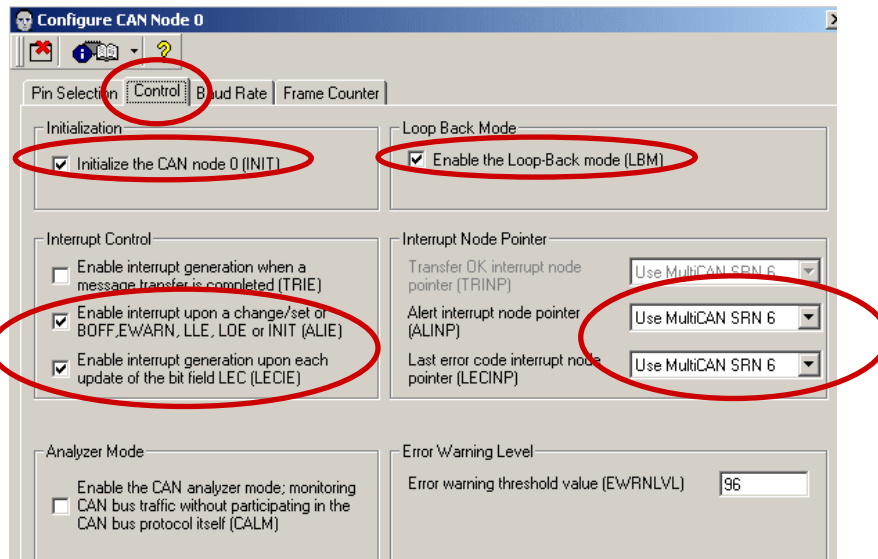
**“Enable interrupt upon a change/set of BOFF, EWARN, LLE, LOE, or INIT (ALIE)”**

**“Enable interrupt generation upon each update of the bit field LEC”**

**“Initialize the CAN node 0 (INIT)”**

Select (ALINP): **Use MultiCAN SRN 6**

Select (LECINP): **Use MultiCAN SRN 6**





# DAvE Setup and Code Generation

## ■ Step 5.2: Configuring MCAN NODE 0 (Cont'd)

-) Click "**Baud Rate**" tab and select the following options:

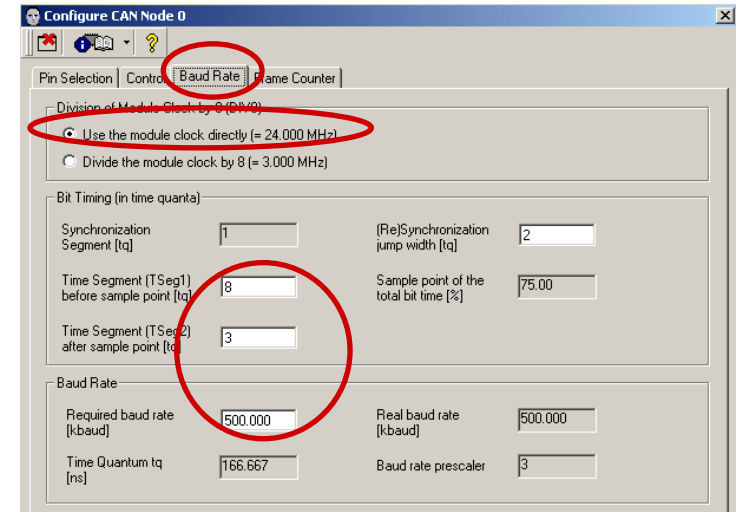
**"Use the module clock directly"**

Enter Time Segment (TSeg1): 8

Enter Time Segment (TSeg2): 3

Enter Required baud rate: 500.000

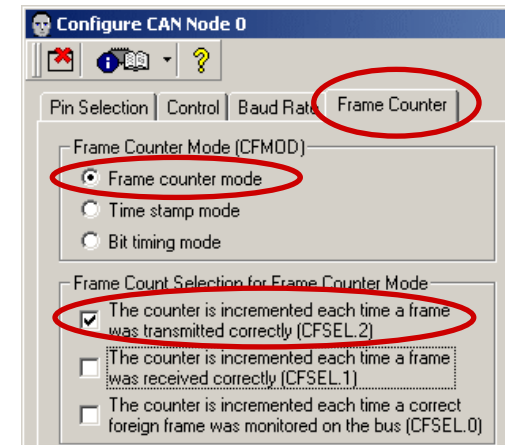
Press Enter afterward.



-) Click "**Frame Counter**" tab and select the following options:

**"Frame counter mode"**

**"The counter is incremented each time a frame was transmitted correctly"**



Click to close the dialog.



DAvE

stop thinking  
Never

# DAvE Setup and Code Generation

## ■ Step 5.3: Configuring MCAN NODE 1

On the MultiCAN menu that appears,

Click “**Nodes**” tab, click the “**Node 1**” button, and do the following:

-) Click “**Control**” tab and select the following options:

**“Enable the Loop-Back mode (LBM)”**

**“Enable interrupt upon a change/set of BOFF, EWARN, LLE, LOE, or INIT (ALIE)”**

**“Enable interrupt generation upon each update of the bit field LEC”**

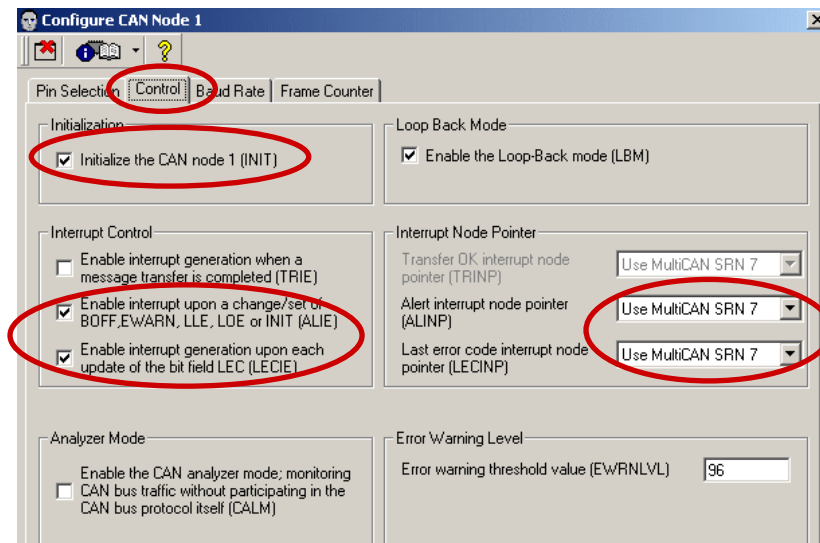
**“Initialize the CAN node 1”**

Select (ALINP): **Use MultiCAN SRN 7**

Select (LECINP): **Use MultiCAN SRN 7**



DAvE



# DAvE Setup and Code Generation

## ■ Step 5.3: Configuring MCAN NODE 1 (Cont'd)

-) Click "**Baud Rate**" tab and select the following options:

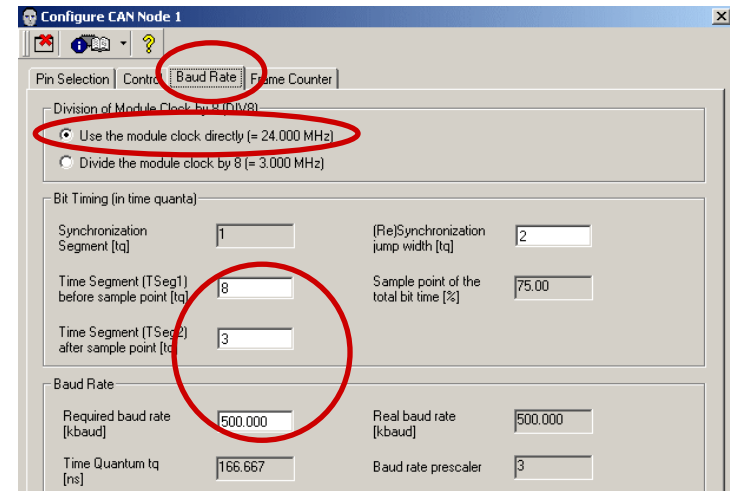
**"Use the module clock directly"**

Enter Time Segment (TSeg1): 8

Enter Time Segment (TSeg2): 3

Enter Required baud rate: 500.000

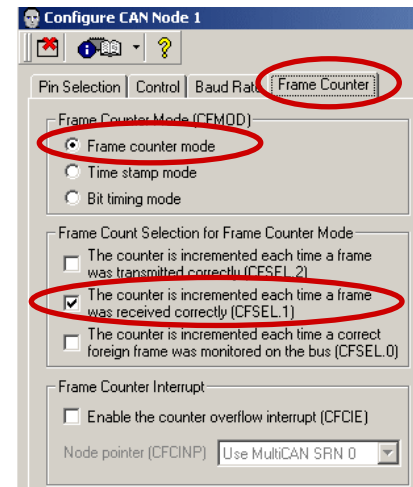
Press Enter afterward.



-) Click "**Frame Counter**" tab and select the following options:

**"Frame counter mode"**

**"The counter is incremented each time a frame was received correctly"**



Click to close the dialog.



# DAvE Setup and Code Generation

## ■ Step 5.4: Configuring LIST

Click “**List**” tab and configure the List as follow:

Drag MO1 from the right window and put it in the first row of List 1 (Node 0).

Drag MO16 from the right window and put it in the first row of List 2 (Node 1).



The screenshot shows the 'MultiCAN Controller' software interface. The 'Lists' tab is selected and highlighted with a red circle. The interface displays three columns: 'List 1 (Node 0)', 'List 2 (Node 1)', and 'List 0 (unallocated MOs)'. The 'List 0' column contains a list of message objects from MO0 to MO28. The 'List 1' and 'List 2' columns are empty except for the first row of 'List 1', which contains 'MO1', and the first row of 'List 2', which contains 'MO16'. A red oval highlights the first row of 'List 1' and the first row of 'List 2'. A note at the bottom of the window reads: 'Note: To change the level (position in the list) and the list of an message object (MO), click on it, drag it to its new position and drop it. To set an message object to the unallocated list, click on it, drag it to the 'List 0' list and drop it.'

Level	List 1 (Node 0)	List 2 (Node 1)	List 0 (unallocated MOs)
Level 15			MO0
Level 14			MO10
Level 13			MO11
Level 12			MO12
Level 11			MO13
Level 10			MO14
Level 9			MO15
Level 8			MO17
Level 7			MO18
Level 6			MO19
Level 5			MO2
Level 4			MO20
Level 3			MO21
Level 2			MO22
Level 1	MO1	MO16	MO23
			MO24
			MO25
			MO26
			MO27
			MO28

# DAvE Setup and Code Generation

## ■ Step 5.5: Configuring Message Objects

Click on the “**MOs**” tab, and then click on “**MO 1**” button.

Configure MO 1 as follow:

Select “**Enable message object 1(MSGVAL)**”

Select “**Transmit data frames, receive and answer remote frames**”

Select Data Length: **1 data bytes**

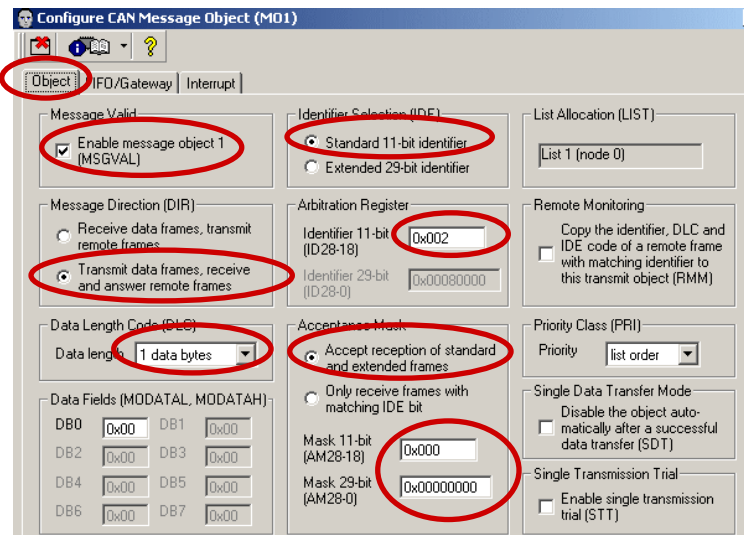
Select “**Standard 11-bit identifier**”

Enter Identifier 11-bit: **0x002**

Select “**Accept reception of standard and extended frames**”

Enter Mask 11-bit: **0x000**

Enter Mask 29-bit: **0x00000000**



Close the dialog by clicking



# DAvE Setup and Code Generation

## ■ Step 5.5: Configuring Message Objects (Cont'd)

Click on **“MO 16”** button.

Configure MO 16 as follow:

Select **“Enable message object 16(MSGVAL)”**

Select **“Receive data frames, transmit remote frames”**

Select Data Length: **8 data bytes**

Select **“Standard 11-bit identifier”**

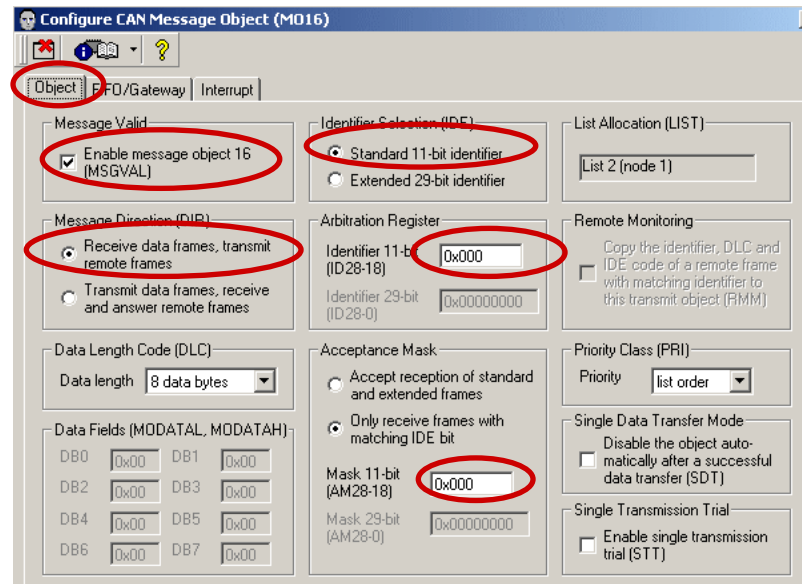
Enter Identifier 11-bit: **0x000**

Select **“Only receive frames with matching IDE bit”**

Enter Mask 11-bit: **0x000**



DAvE



# DAvE Setup and Code Generation

## ■ Step 5.5: Configuring Message Objects (Cont'd)

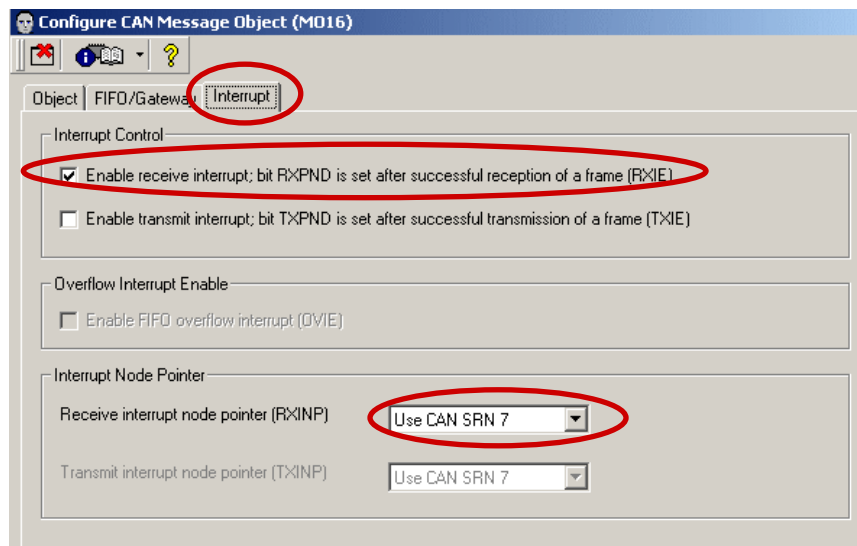
Click on "**Interrupt**" tab.

Select "**Enable receive interrupt; bit RXPND is set after successful reception of a frame (RXIE)**"

Select RXINP to: "**Use CAN SRN 7**"



DAvE



Close the dialog by clicking



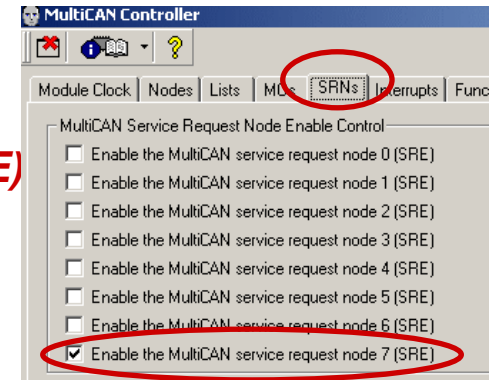


# DAvE Setup and Code Generation

## Step 5.6: Configuring SRN and interrupt

Click on the “**SRNs**” tab.

Select “**Enable the MultiCAN service request node 7 (SRE)**”



Click on the “**Interrupt**” tab

Make sure to select “**Level 14**” into “**Priority 1**”

Click on the “**Functions**” tab.

Select “**CAN\_vlnit**” function

Level	Interrupt Source	Priority 0	Priority 1	Priority 2	Priority 3
Level 0	Non Maskable Interrupt (NMI)	Highest Priority (can't be changed)			
Level 1	External Interrupt 0	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 2	Timer 0 Interrupt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 3	External Interrupt 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 4	Timer 1 Interrupt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 5	UART Interrupt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 6	Timer 2 / BRG / MultiCAN Node 0 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 7	ADC / MultiCAN Node 1 and 2 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 8	SSC Interrupt	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 9	External 2 / T21 / UART1 / BRG1 Intrpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 10	External [6:3] / MultiCAN Node 3 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 11	CCU6 Node 0 / MultiCAN Node 4 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 12	CCU6 Node 1 / MultiCAN Node 5 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 13	CCU6 Node 2 / MultiCAN Node 6 Interpts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Level 14	CCU6 Node 3 / MultiCAN Node 7 Interpts	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Click on the “**Functions2**” tab.

Select :

“**CAN\_vTransmit**”

“**CAN\_vLoadData**”

functions.

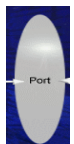
Close the dialog by clicking



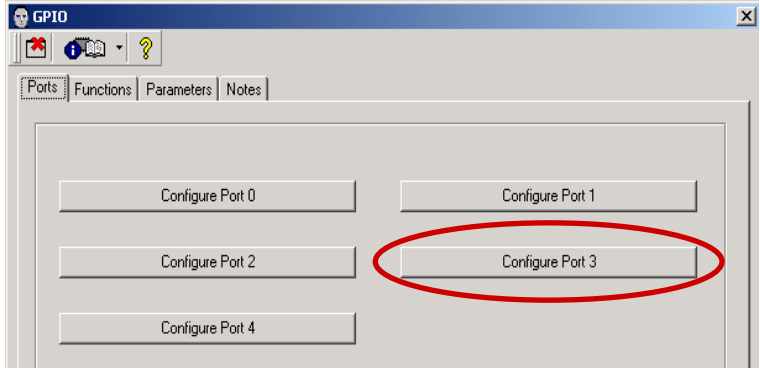
# DAvE Setup and Code Generation

## Step 6.1: Configuring Port Module

Click "**Port**" bubble.



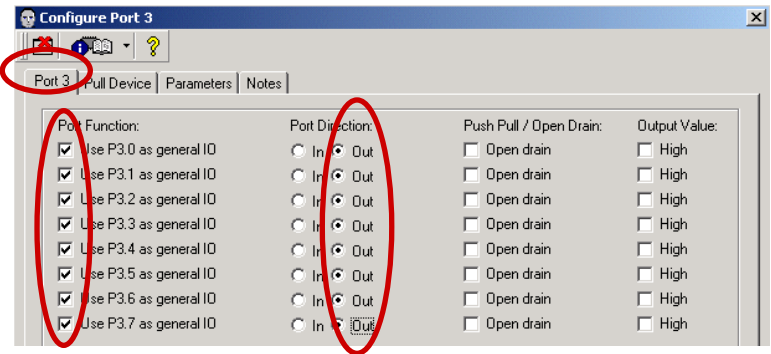
On the GPIO menu that appears, click on "**Configure Port3**" button



## Step 6.2: Configuring Port Module

Select the box for **P3.0 to P3.7 as general IO**

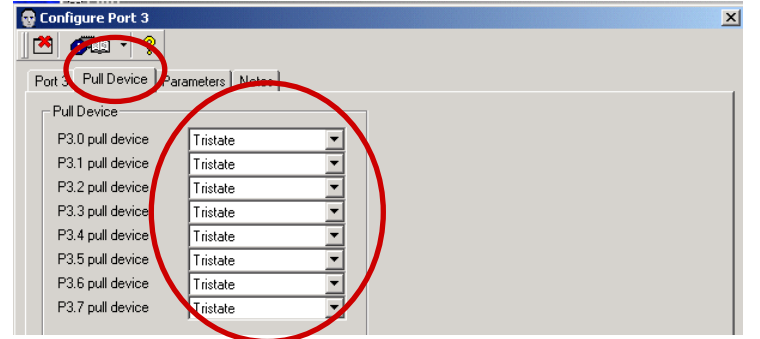
Select the "**Out**" for the Port Direction.



## Step 6.3: Configuring Port Module

Click "**Pull Device**" tab. Ensure all fields are set to "**Tristate**"

Close the interface by clicking



stop thinking  
Never

# DAvE Setup and Code Generation

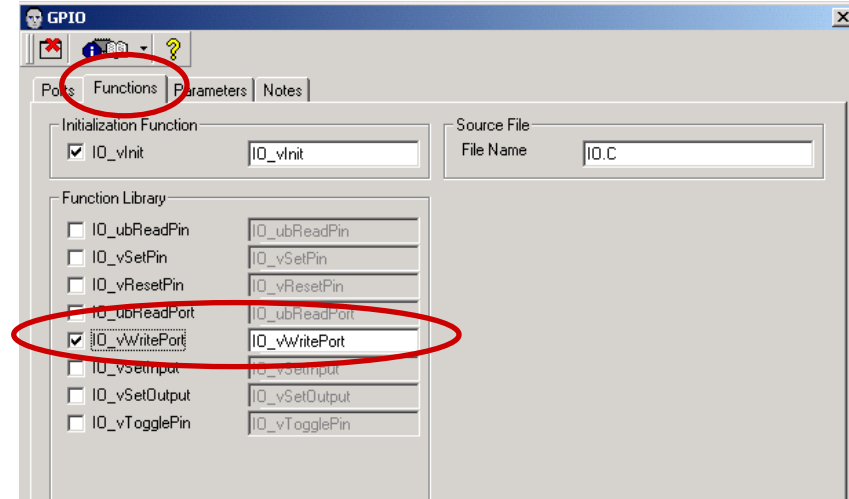
## Step 6.4: Configuring Port Module

On the GPIO menu, click on **“Function”** tab. Select the following checkboxes:

**“IO\_vInit”**

**“IO\_vWritePort”**

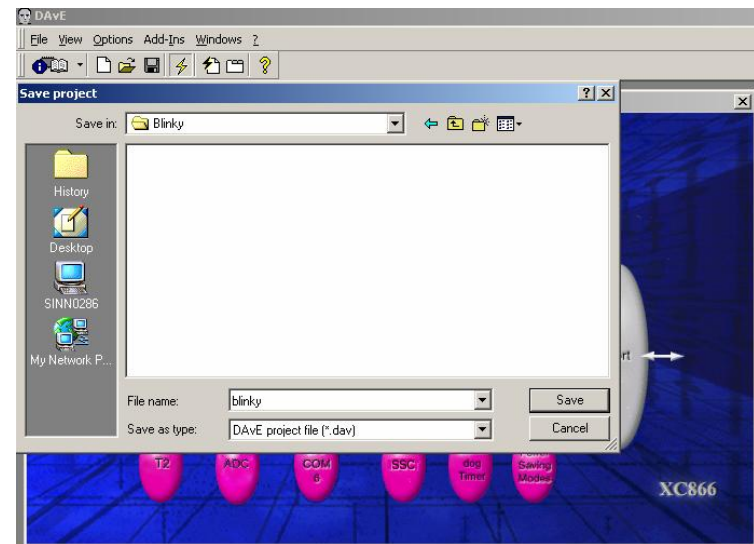
Close the interface by clicking 



## Step 7: Generating Code and Save.

Click  button to generate the code.

Enter File name as **“mcan\_adc”** to save your project as DAVE project file (\*.dav).



# DAvE Code Modification with KEIL uVision

- Start the KEIL uVision by clicking the icon  in the desktop.

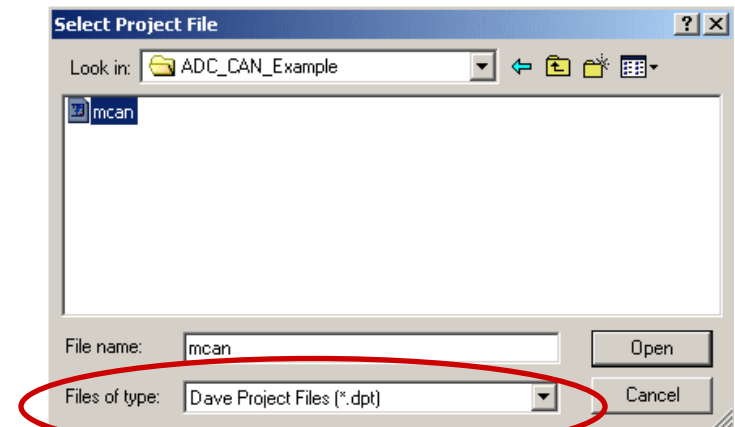
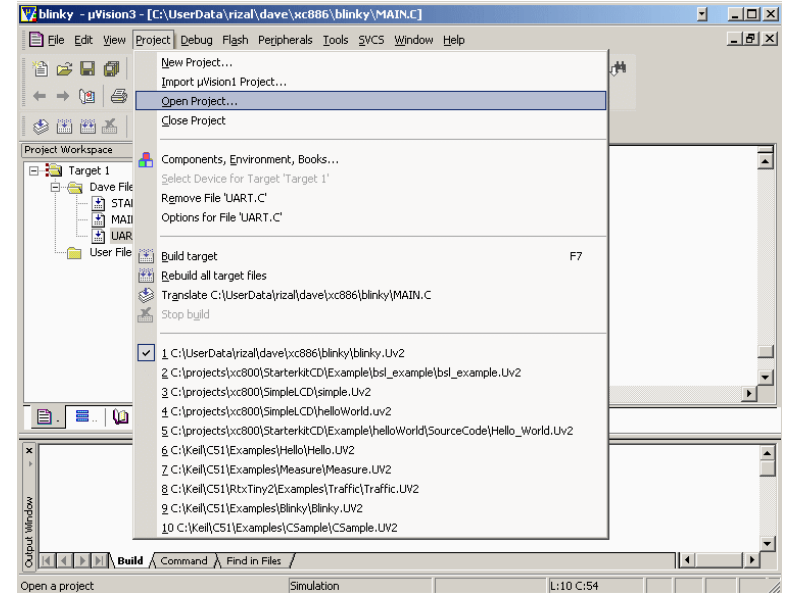
- Import your DAVe project by:

***Project → Open Project***

Set the Files of types to:

***Dave Project Files (\*.dpt)***

Select the project that you have just generated in DAVe Setup session.

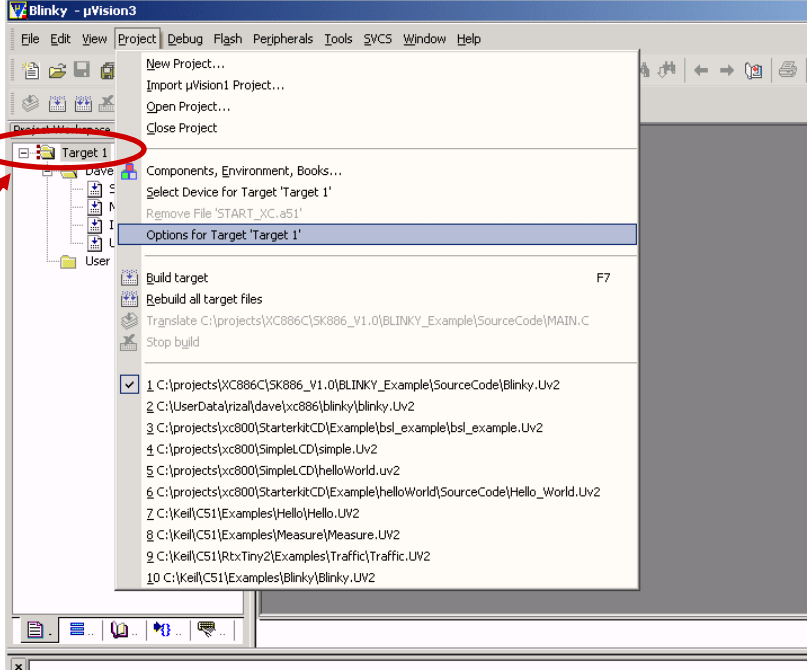


# DAvE Code Modification with KEIL uVision

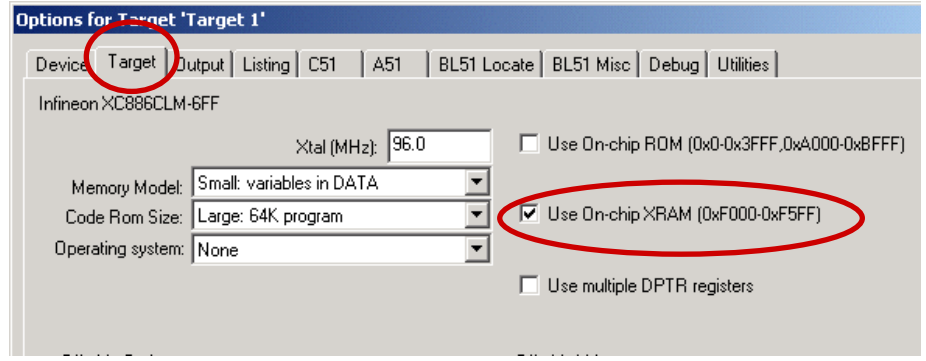
- After the project is opened, configure the option using:

**Project → Option for Target**

(To see this menu, you must make sure that **Target 1** in the left window is selected)



- Click the **“Target”** tab and select the box: **“Use On-chip XRAM”**



# DAvE Code Modification with KEIL uVision

- Select **Main.C** from the left window of uVision (expand the **Target 1** to see this entry) so we can modify this file.
- Goto line 214 of Main.C and insert the following line:

```

MAIN_vInit();

// USER CODE BEGIN (MAIN_Main,3)
// Start the Parallel Conversion of Channel 7 // Add this Line
ADC_vStartParReqChNum(0x80); // Add this Line
// USER CODE END
    
```

*Make sure that you only add the new code between "USER CODE BEGIN" and "USER CODE END"*

- Select **SHARED\_INT.C** from the left window of uVision so we can modify this file.
- Goto line 188 of SHARED\_INT.C and insert the following line:

```

void SHINT_viXINTR6lSr(void) interrupt XINTR6INT
{
// USER CODE BEGIN (SHINT_XINTR6lSr,2)
un_32bit adcResult_old; // Add this Line
un_32bit adcResult_new; // Add this Line
// USER CODE END
    
```

*Make sure that you only add the new code between "USER CODE BEGIN" and "USER CODE END"*



DAvE

# DAvE Code Modification with KEIL uVision

- Goto line 203 of SHARED\_INT.C and insert the following line:

```
IRCON1 &= ~(ubyte)0x08; // clear ADCSRC0
```

```
// USER CODE BEGIN (SHINT_XINTR6Isr,3)
```

```
// Get the ADC Result and transmit it thru CAN Node 1
```

```
adcResult_new.ubDB[3] = ADC_uwGetResultData0() & 0xFF; // Add this Line
```

```
if (adcResult_old.ubDB[3] != adcResult_new.ubDB[3]) { // Add this Line
```

```
    CAN_vLoadData(0x1, &adcResult_new.ulVal); // Add this Line
```

```
    CAN_vTransmit(0x1); // Add this Line
```

```
    adcResult_old.ubDB[3] = adcResult_new.ubDB[3]; // Add this Line
```

```
}
```

```
// USER CODE END
```

*// Add this Line*  
*// Add this Line*  
*// Add this Line*  
*// Add this Line*  
*// Add this Line*

***Make sure that you only add the new code between "USER CODE BEGIN" and "USER CODE END"***

- Copy the commented code from line 449-476 and paste them at line 479 between (after that, remove all of the comment "//"):

```
// USER CODE BEGIN (SRN7_OBJ,1)
```

```
// USER CODE END
```



DAvE



# DAvE Code Modification with KEIL uVision

- Goto line 496 of SHARED\_INT.C (after un-commenting the code, you will be able to see this line)

```

else
    {
        // The CAN controller has stored a new message into this object.
        CAN_vWriteCANAddress(CAN_MODATAL(ubTempMsgID)); // Add this line
        CAN_vReadEN(); // Add this line
        IO_vWritePort(P3, CAN_DATA0); // Add this line
    }

```

- Now, compile the code by clicking the Compile icon, after that, download and run the code.  
You can rotate the Potentiometer and see the LED changing
- Check if there is an error in the output window. You can always refer to the project: “mcan” located at: Examples\SoftwareExample\MCAN

CONGRATULATION !!

YOU SHOULD SEE THE LED CHANGING NOW



DAvE