

56800/E SCI Hands-On Exercise

56800/E Training



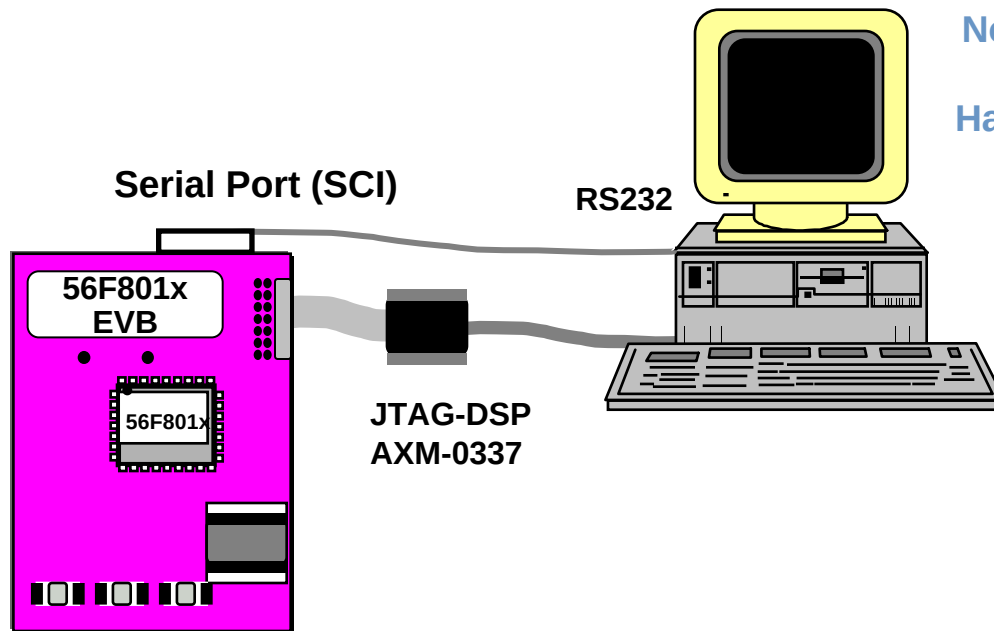
Slide 1

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2004

Task Description

Develop a monitor application that uses SCI in messages between PC and DSC56F801x EVB.

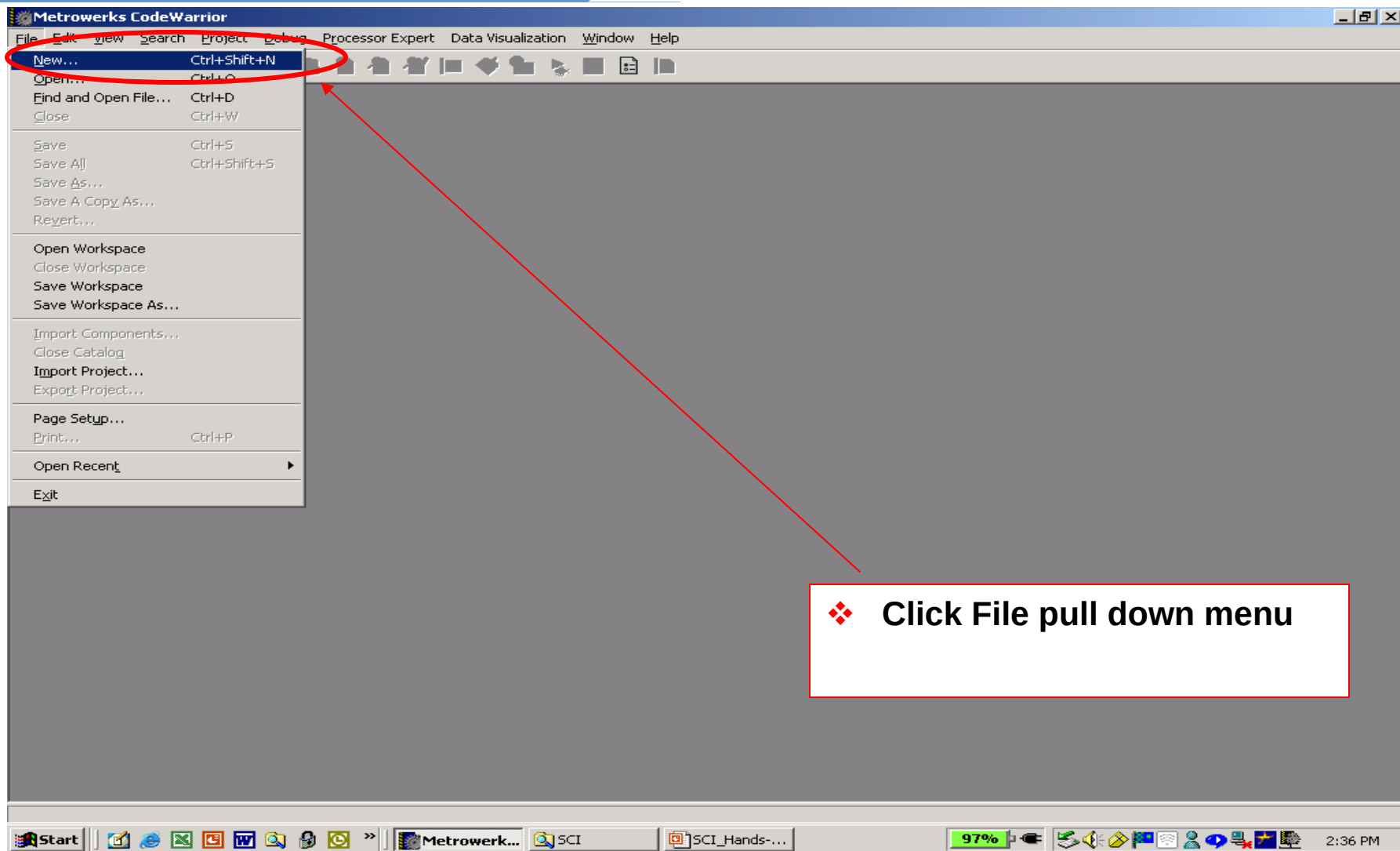
HyperTerminal (Console) for
CAN Data Display
via RS232,
Bits/sec : 9600
No parity : none
Stop bit : 1
Hardware : None



Approach

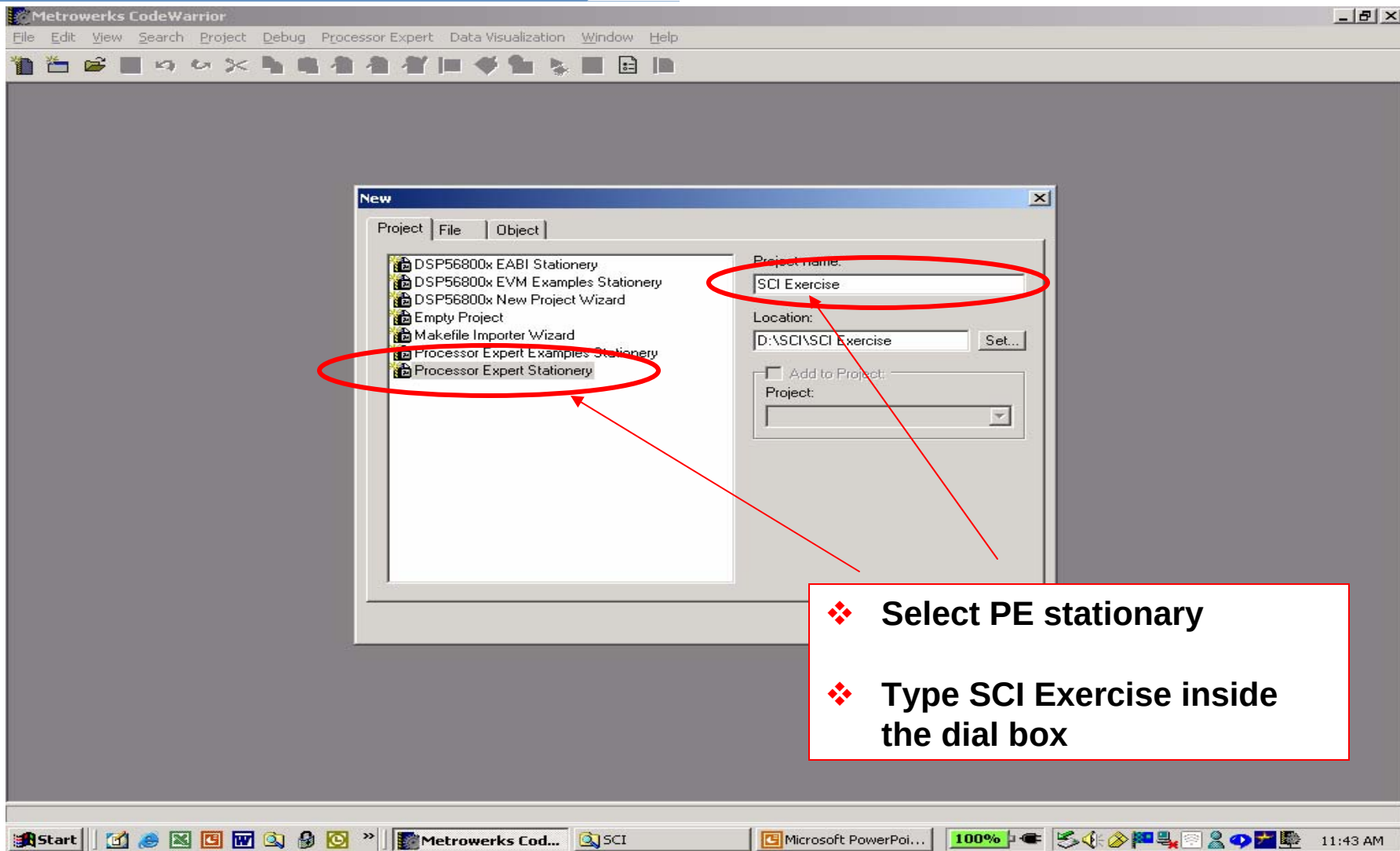
- ❖ **Use Processor Expert Beans to implement Target System application**
 - ✓ **AsynchroSerial**
 - ✓ **Download and Execute on 56F801x Demo Board**

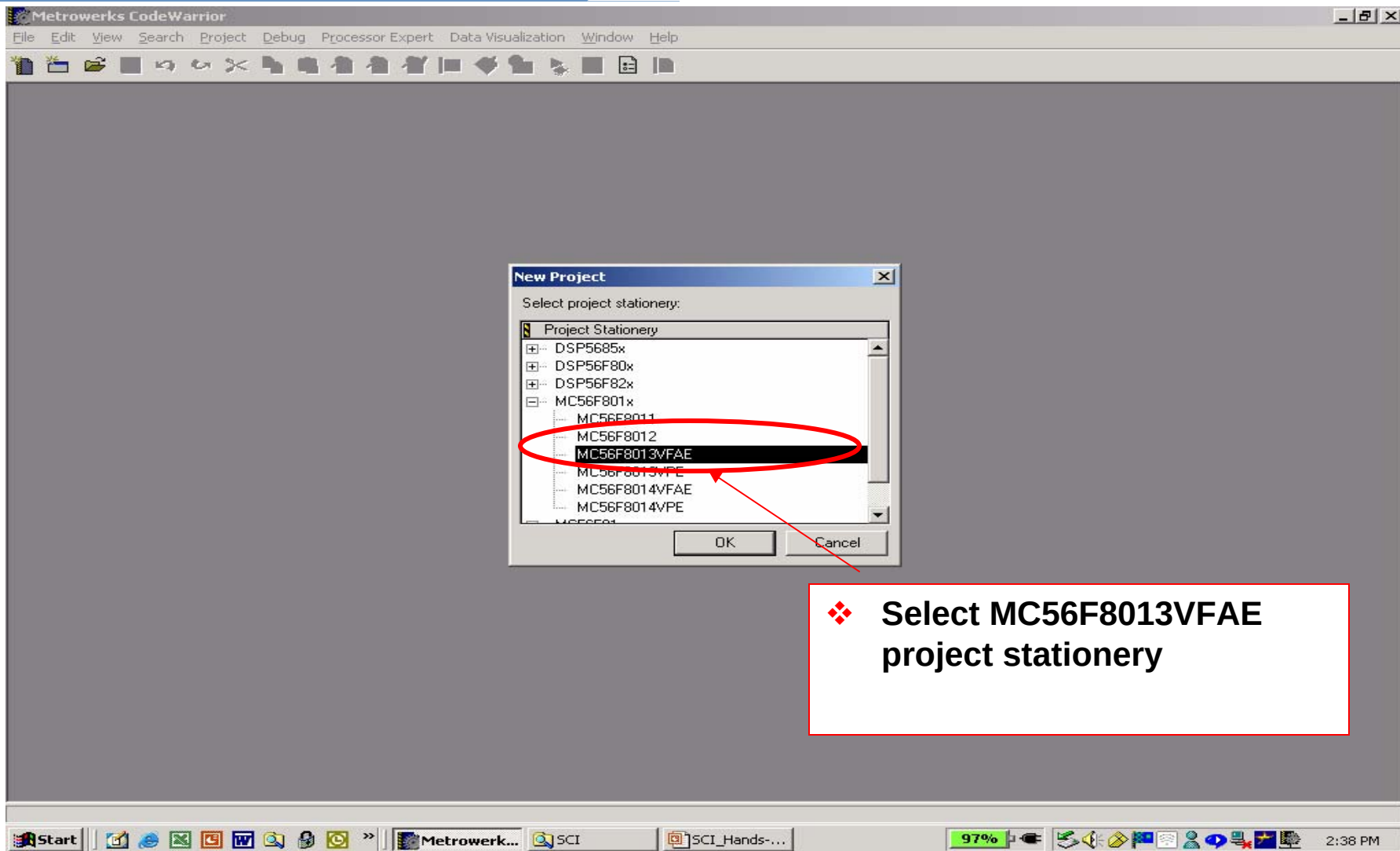
56800/E Training



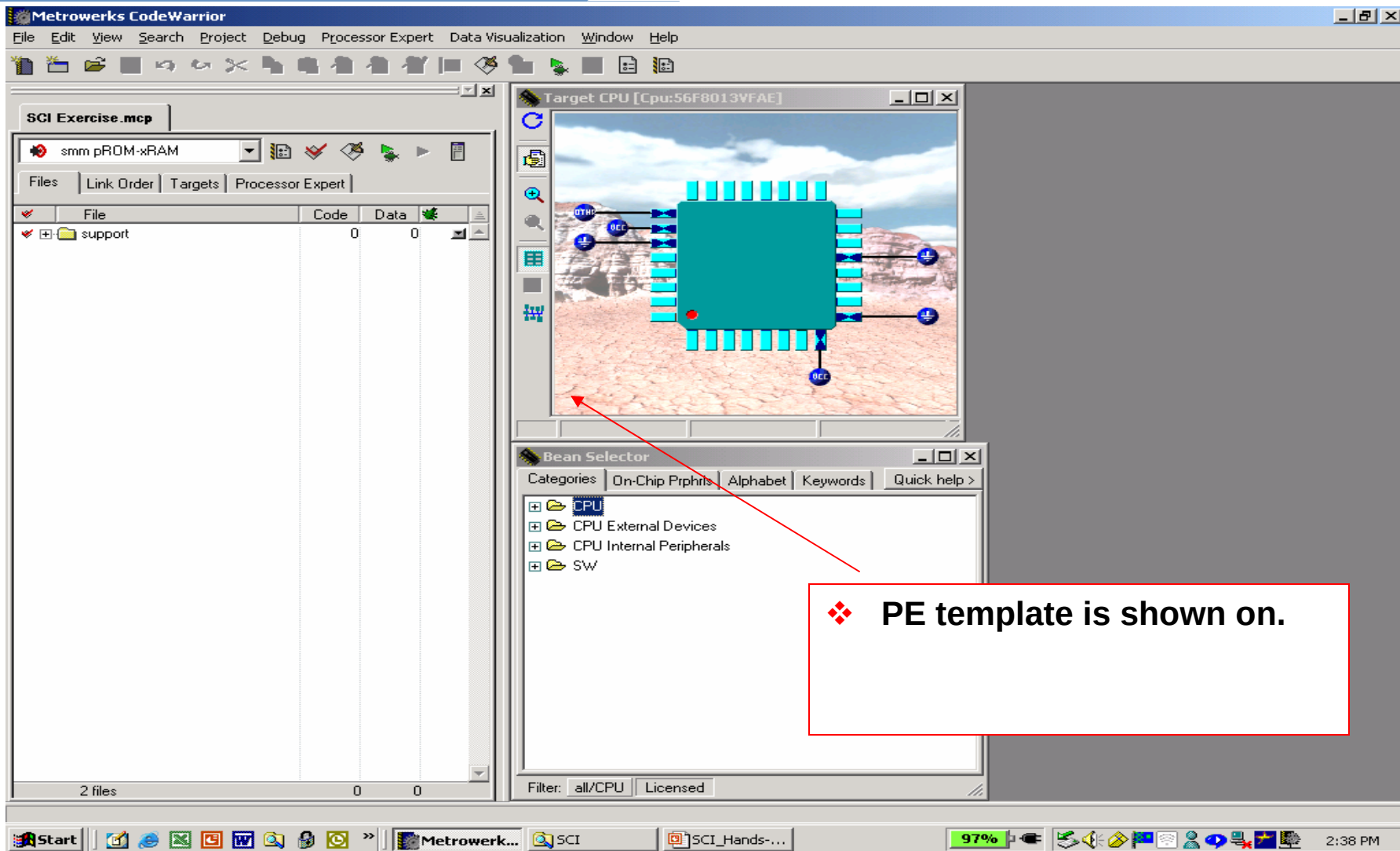
Slide 4

Freescall™ and the Freescall logo are trademarks of Freescall Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescall Semiconductor, Inc. 2004





56800/E Training



Slide 7

Freescall™ and the Freescall logo are trademarks of Freescall Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescall Semiconductor, Inc. 2004

Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

SCI Exercise.mcp

Files Link Order Targets Processor Expert

Configurations

- smm pROM-xRAM

Operating System

CPU's

- Cpu:56F8013VFAE

Beans

- AS1:AsynchroSerial

User Modules

- SCI_Exercise.c:main

Generated Modules

- External Modules
- Documentation
- PESL

Target CPU [Cpu:56F8013VFAE]

32 TDO_GPIOD1 TDO_GPIOD1 (none)

Bean Selector

Categories On-Chip Prphrls Alphabet Keywords Quick help

- CPU
- CPU External Devices
- CPU Internal Peripherals
- Communication
 - AsynchroMaster
 - AsynchroSerial**
 - AsynchroSlave
- FreescaleCAN
- FreescaleHI8
- FreescaleSSI
- InternalI2C
- SynchroMaster
- SynchroSlave
- Converter

Filter: all/CPU Licensed

Bean Inspector AS1:AsynchroSerial

Bean Items Visibility Help < > Peripheral Initialization >

Properties Methods Events Comment

Bean name AS1

Channel SCI

Interrupt service/event Disabled

Settings

- Parity none
- Width 8 bits
- Stop bit 1

SCI output mode

- LIN slave mode Disabled

Receiver

- Enabled
- RxD GPIOB6_RX
- RxD pin signal

Transmitter

- Enabled
- TxD GPIOB7_TX
- TxD pin signal

Baud rate

Break signal Disabled

Wakeup condition Idle line wak

Transmitter output Not inverted

Stop in wait mode no

Initialization

- Enabled in init. code yes
- Events enabled in init. yes

CPU clock/speed selection

❖ Kick AsynchroSerial Bean under Communication PE menu

❖ Then, kick Baud rate dial box under the properties of Bean Inspector AS1:AsynchroSerial

The image shows the Metrowerks CodeWarrior IDE with the 'Timing - Baud rate' dialog box open. The 'Runtime setting' is set to 'fixed value'. The 'Init. value' section shows 'requested baud rate' set to 9600, 'unit' set to 'baud', and 'Error allowed' set to 1.5%. The 'Speed mode' section shows 'Adjusted value' as 9615.385 baud, 'Prescaler' as 208, and 'Error' as 0.16%. The 'Possible settings' section lists 'Closest values: 9569baud, 9615baud;' and 'Possible in all speed modes: 244-2000000 baud;'. A red circle highlights the 'requested baud rate' field, and a red arrow points to it from a text box that says '❖ Type 9600 baud rate inside the dial box'.

The screenshot shows the Metrowerks CodeWarrior IDE. The 'Processor Expert' menu is open, and the 'Generate Code' option is highlighted with a red circle. A red arrow points from this menu item to a text box. The 'Bean Inspector' for 'AS1:AsynchroSerial' is visible on the right, showing various configuration settings. The 'Bean Selector' on the left shows the 'AsynchroSerial' bean selected under the 'Communication' category.

❖ Select Generate Code 'SCI Exercise.mcp' under PE pull down menu to generate the code

Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

Code generation

Processor Expert is working. Please wait.

SCI Exercise

smm pROM-xRAM

Files Link Order Targets Processor Expert

Configurations

- smm pROM-xRAM

Operating System

CPU

- Cpu:56F8013VFAE

Beans

- AS1:AsynchroSerial

User Modules

- SCI_Exercise.c:main

Generated Modules

- External Modules
- Documentation
- PESL

Target CPU [Cpu:56F8013VFAE]

Code Generation

Project: SCI_Exercise

Module: Cpu

Current line: 624 Total lines: 6469

Errors: 0 Warnings: 0 Hints: 0

Cancel

Bean Inspector AS1:AsynchroSerial

Bean Items Visibility Help < > Peripheral Initialization >

Properties Methods Events Comment

Bean name AS1

Channel SCI

Interrupt service/event Disabled

Settings

- Parity none
- Width 8 bits
- Stop bit 1

SCI output mode Normal

Receiver Enabled

- RxD GPIOB6_RX
- RxD pin signal

Transmitter Enabled

- TxD GPIOB7_TX
- TxD pin signal

Baud rate 9600 baud

Break signal Disabled

Wakeup condition Idle line wak

Transmitter output Not inverted

Stop in wait mode no

Initialization

- Enabled in init. code yes
- Events enabled in init. yes

CPU clock/speed selection

- High speed mode This bean er
- Low speed mode This bean di
- Slow speed mode This bean di

BASIC ADVANCED EXPERT Bean Level: High Le

Errors: 0, warnings: 0, hints: 0

Start Metrowerk... SCI SCI_Hands-... 97% 2:39 PM

The screenshot shows the Metrowerks CodeWarrior IDE. On the left, the 'Project Explorer' pane displays the project structure for 'SCI_Exercise.mcp'. The 'User Modules' folder is expanded, and 'SCI_Exercise.c:main' is highlighted with a red circle. A red arrow points from this circle to the main code editor. The main editor displays the contents of 'SCI_Exercise.c', which includes header files and a main function. A red box with a diamond icon and text is overlaid on the editor, pointing to the main function area.

❖ Open up the SCI_Exercise.c:main file to edit user code.

The screenshot shows the Metrowerks CodeWarrior IDE. On the left, the 'Processor Expert' pane displays a list of beans for the 'AS1:AsynchroSerial' component. The 'SendChar' bean is selected and circled in red. A red arrow points from this circle to a text box on the right. The main editor window shows the source code for 'SCI_Exercise.c', which includes headers for 'AS1.h' and 'PE_*.h'. The code defines a 'main' function that initializes the processor and then enters a loop. The 'Communication' pane at the bottom shows the 'AsynchroSerial' bean selected. A status bar at the bottom indicates 97% CPU usage and the time is 2:45 PM.

Select PE Bean SendChar for sending out characters

Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

SCI_Exercise.mcp

smm pROM-xRAM

Files Link Order Targets Processor Expert

AS1:AsynchroSerial

- BeforeNewSpeed
- AfterNewSpeed
- OnError
- OnRxChar
- OnTxChar
- OnFullRxBuf
- OnFreeTxBuf
- OnBreak
- OnTxComplete
- Enable
- Disable
- EnableEvent
- DisableEvent
- RecvChar
- SendChar**
- RecvBlock
- SendBlock
- ClearRxBuf
- ClearTxBuf
- CharsInRxBuf
- GetCharsInRxBuf
- CharsInTxBuf
- GetCharsInTxBuf
- SetBaudRateMode
- GetError
- GetBreak
- SetBreak
- TurnTxOn
- TurnTxOff
- TurnBxOn
- TurnBxOff

2 files

SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /*** Processor Expert internal initialization. DON'T REMOVE THIS! ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization. ***/

    /* Write your code here */
    for(;;) {AS1_SendChar()}
}

/* END SCI_Exercise */
/*
*****
** This file was created by UNIS Processor Expert 2.96 [03.11.03]
** for the Freescale 56800 series of microcontrollers.
**
*/
```

Line 42 Col 12

Communication

- AsynchroMaster
- AsynchroSerial
- AsynchroSlave
- FreescaleCAN
- FreescaleHI8
- FreescaleSSI
- InternalI2C
- SynchroMaster
- SynchroSlave
- Converter

Events enabled in init.

CPU clock/speed selection

Put PE Bean SendChar to SCI_Exercise.c:main by Drag-n-Drop Methods.

SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /*** Processor Expert internal initialization. DON'T REMOVE THIS ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.

    /* Write your code here */

    for(;;) {
        AS1_SendChar('A');
        AS1_SendChar('e');
        AS1_SendChar('l');
        AS1_SendChar('l');
        AS1_SendChar('o');
        AS1_SendChar('\r');
        AS1_SendChar('\n');
    }
}
```

Line 49 Col 23

❖ Made more characters to send out by Copy-n-Paste methods.

The image shows the Metrowerks CodeWarrior IDE with the Bean Inspector for AS1:AsynchroSerial. The 'Methods' tab is selected, and a red circle highlights the 'Methods' tab. A red arrow points from the 'Methods' tab to the 'SendChar' method in the code editor.

Code Editor (SCI_Exercise.c):

```
#include "AS1.h"
/* Include shared modules, which are u
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /** Processor Expert internal initi
    PE_low_level_init();
    /** End of Processor Expert interne

    /* Write your code here */

    for(;;) {
        AS1_SendChar('H');
        AS1_SendChar('e');
        AS1_SendChar('l');
        AS1_SendChar('l');
        AS1_SendChar('o');
        AS1_SendChar('\r');
        AS1_SendChar('\n');
    }
}
```

Bean Inspector AS1:AsynchroSerial:

Bean	Item	Value	Unit
AS1	Bean name	AS1	
AS1	Channel	SCI	SCI
AS1	Interrupt service/event	Disabled	
AS1	Settings		
AS1	Parity	none	nonv
AS1	Width	8 bits	8 bit
AS1	Stop bit	1	1
AS1	SCI output mode	Normal	
AS1	LIN slave mode	Disabled	
AS1	Receiver	Enabled	
AS1	RxD	GPIOB6_RX	GPIO
AS1	RxD pin signal		
AS1	Transmitter	Enabled	
AS1	TxD	GPIOB7_TX	GPIO
AS1	TxD pin signal		
AS1	Baud rate	9600 baud	high
AS1	Break signal	Disabled	
AS1	Wakeup condition	Idle line wak	
AS1	Transmitter output	Not inverted	
AS1	Stop in wait mode	no	
AS1	Initialization		
AS1	Enabled in init. code	yes	
AS1	Events enabled in init.	yes	
AS1	CPU clock/speed selection		
AS1	High speed mode	This bean er	This
AS1	Low speed mode	This bean di	This
AS1	Slow speed mode	This bean di	This

- ❖ We need to wait for the 1st character send out completely if we want to send out 2nd character in SCI communication.
- ❖ kick Methods page of Bean Inspector AS1:AsynchroSerial

Bean Inspector AS1:AsynchroSerial

Bean	Items	Visibility	Help	Comment
Enable		don't generate code		
Disable		don't generate code		
EnableEvent		don't generate code		
DisableEvent		don't generate code		
RecvChar		generate code		
SendChar		generate code		
RecvBlock		don't generate code		
SendBlock		don't generate code		
ClearRxBuf		don't generate code		
ClearTxBuf		don't generate code		
CharsInRxBuf		don't generate code		
GetCharsInRxBuf		generate code		
CharsInTxBuf		don't generate code		
GetCharsInTxBuf		generate code		
SetBaudRateMode		don't generate code		
GetError		don't generate code		
GetBreak		don't generate code		
SetBreak		don't generate code		
TurnTxOn		don't generate code		
TurnTxOff		don't generate code		
TurnRxOn		don't generate code		
TurnRxOff		don't generate code		
SetIdle		don't generate code		
LoopMode		don't generate code		
ConnectPin		don't generate code		
GetRxIdle		don't generate code		
GetTxComplete		don't generate code		

❖ Enable GetRxIdle PE Bean by kicking this dial box

Bean Inspector AS1:AsynchroSerial

Bean	Properties	Methods	Events	Comment
Enable	☒		don't generate code	
Disable	☒		don't generate code	
EnableEvent	☒		don't generate code	
DisableEvent	☒		don't generate code	
RecvChar	☒		generate code	
SendChar	☒		generate code	
RecvBlock	☒		don't generate code	
SendBlock	☒		don't generate code	
ClearRxBuf	☒		don't generate code	
ClearTxBuf	☒		don't generate code	
CharsInRxBuf	☒		don't generate code	
GetCharsInRxBuf	☒		generate code	
CharsInTxBuf	☒		don't generate code	
GetCharsInTxBuf	☒		generate code	
SetBaudRateMode	☒		don't generate code	
GetError	☒		don't generate code	
GetBreak	☒		don't generate code	
SetBreak	☒		don't generate code	
TurnTxOn	☒		don't generate code	
TurnTxOff	☒		don't generate code	
TurnRxOn	☒		don't generate code	
TurnRxOff	☒		don't generate code	
SetIdle	☒		don't generate code	
LoopMode	☒		don't generate code	
ConnectPin	☒		don't generate code	
GetRxIdle	☒		generate code	
GetTxComplete	☒		don't generate code	

❖ Enable GetTxComplete PE Bean by kicking this dial box

Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

SCI Exercise.mcp

smm pROM-xRAM

Files Link Order Targets Processor Expert

AS1:AsynchroSerial

- ☒ BeforeNewSpeed
- ☒ AfterNewSpeed
- ☒ OnError
- ☒ OnRxChar
- ☒ OnTxChar
- ☒ OnFullRxBuf
- ☒ OnFreeTxBuf
- ☒ OnBreak
- ☒ OnTxComplete
- ☒ Enable
- ☒ Disable
- ☒ EnableEvent
- ☒ DisableEvent
- ☒ RecvChar
- ☒ SendChar
- ☒ RecvBlock
- ☒ SendBlock
- ☒ ClearRxBuf
- ☒ ClearTxBuf
- ☒ CharsInRxBuf
- ☒ GetCharsInRxBuf
- ☒ CharsInTxBuf
- ☒ GetCharsInTxBuf
- ☒ SetBaudRateMode
- ☒ GetError
- ☒ GetBreak
- ☒ SetBreak
- ☒ TurnTxOn
- ☒ TurnTxOff
- ☒ TurnRxOn
- ☒ TurnRxOff

2 files 0 0

SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are u
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /** Processor Expert internal initi
    PE_low_level_init();
    /** End of Processor Expert interne

    /* Write your code here */

    for(;;) {
        AS1_SendChar('H');
        AS1_SendChar('e');
        AS1_SendChar('l');
        AS1_SendChar('l');
        AS1_SendChar('o');
        AS1_SendChar('\r');
        AS1_SendChar('\n');
    }
}
```

Bean Inspector AS1:AsynchroSerial

Bean Items Visibility Help < > Peripheral Initialization

Properties	Methods	Events	Comment
☒ Enable	don't generate code	☒	
☒ Disable	don't generate code	☒	
☒ EnableEvent	don't generate code	☒	
☒ DisableEvent	don't generate code	☒	
☒ RecvChar	generate code	☒	
☒ SendChar	generate code	☒	
☒ RecvBlock	don't generate code	☒	
☒ SendBlock	don't generate code	☒	
☒ ClearRxBuf	don't generate code	☒	
☒ ClearTxBuf	don't generate code	☒	
☒ CharsInRxBuf	don't generate code	☒	
☒ GetCharsInRxBuf	generate code	☒	
☒ CharsInTxBuf	don't generate code	☒	
☒ GetCharsInTxBuf	generate code	☒	
☒ SetBaudRateMode	don't generate code	☒	
☒ GetError	don't generate code	☒	
☒ GetBreak	don't generate code	☒	
☒ SetBreak	don't generate code	☒	
☒ TurnTxOn	don't generate code	☒	
☒ TurnTxOff	don't generate code	☒	
☒ TurnRxOn	don't generate code	☒	
☒ TurnRxOff	don't generate code	☒	
☒ GetTxComplete	generate code	☒	

ADVANCED EXPERT Bean Level: High

Errors: 0, warnings: 0, hints: 1

Hint: File startup\56F80xx_init.asm copied to the pro

Start Metrowerks SCI SCI_Hands-... 97% 2:53 PM

❖ Scroll up the screen to reach GetTxComplete PE Bean by kick the scroll bar

Metrowerks CodeWarrior

SCI_Exercise.mcp

smm pROM-xRAM

Files | Link Order | Targets | Processor Expert

DisableEvent
☒ RecvChar
☒ SendChar
☒ SendBlock
☒ ClearRxBuf
☒ ClearTxBuf
☒ CharsInRxBuf
☒ GetCharsInRxBuf
☒ CharsInTxBuf
☒ GetCharsInTxBuf
☒ SetBaudRateMode
☒ GetError
☒ GetBreak
☒ SetBreak
☒ TurnTxOn
☒ TurnTxOff
☒ TurnRxOn
☒ TurnRxOff
☒ SetIdle
☒ LoopMode
☒ ConnectPin
☒ GetRxIdle
☒ GetTxComplete

User Modules
☒ SCI_Exercise.c:main

Generated Modules
☒ External Modules
☒ Documentation
☒ PEST

2 files

SCI_Exercise.c

Path: D:\Data\Technical Training\DSC training\Exam...\SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /*** Processor Expert internal initialization. DON'T REMOVE THIS ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.

    /* Write your code here */
    for(;;) {
        AS1_SendChar('H'); AS1_GetTxComplete()

        AS1_SendChar('e');
        AS1_SendChar('l');
        AS1_SendChar('l');
        AS1_SendChar('o');
        AS1_SendChar('\r');
        AS1_SendChar('\n');
    }
}
```

Line 43 Col 22

Communication
☒ AsynchroMaster
☒ AsynchroSlave
☒ FreescaleCAN
☒ FreescaleH18
☒ FreescaleSSI
☒ InternalI2C
☒ SynchroMaster
☒ SynchroSlave
☒ Converter

Serial don't generate code
 LoopMode don't generate code

❖ Put PE Bean GetTxComplete to SCI_Exercise.c:main by Drag-n-Drop Methods.

Start | Metrowerks... | SCI | SCI_Hands... | 97% | 2:55 PM

Metrowerks CodeWarrior

SCI_Exercise.mcp

smm pROM-xRAM

Files | Link Order | Targets | Processor Expert

DisableEvent
☒ RecvChar
☒ SendChar
☒ SendBlock
☒ ClearRxBuf
☒ ClearTxBuf
☒ CharsInRxBuf
☒ GetCharsInRxBuf
☒ CharsInTxBuf
☒ GetCharsInTxBuf
☒ SetBaudRateMode
☒ GetError
☒ GetBreak
☒ SetBreak
☒ TurnTxOn
☒ TurnTxOff
☒ TurnRxOn
☒ TurnRxOff
☒ SetIdle
☒ LoopMode
☒ ConnectPin
☒ GetRxIdle
☒ GetTxComplete

User Modules
☒ SCI_Exercise.c:main

Generated Modules
 External Modules
 Documentation
 PESL

2 files

SCI_Exercise.c

Path: D:\Data\Technical Training\56800 training\Exam...\SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
  /*** Processor Expert internal initialization. DON'T REMOVE THIS ***/
  PE_low_level_init();
  /*** End of Processor Expert internal initialization.

  /* Write your code here */

  for(;;) {
    AS1_SendChar('H');
    AS1_SendChar('e');
    AS1_SendChar('l');
    AS1_SendChar('l');
    AS1_SendChar('o');
    AS1_SendChar('\r');
    AS1_SendChar('\n');
  }
}
```

Line 43 Col 54

Communication
☒ AsynchroMaster
☒ AsynchroSerial
☒ AsynchroSlave
☒ FreescaleCAN
☒ FreescaleH18
☒ FreescaleSSI
☒ Internal2C
☒ SynchroMaster
☒ SynchroSlave
☒ Converter

SetIdle don't generate code
 LoopMode don't generate code

Hint: File startup\56800x_init.asm copied to the project

Start | Metrowerks... | SCI | SCI_Hands... | 97% | 2:56 PM

❖ Modify it as while statement.

Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

SCI_Exercise.mcp

smm pROM-xRAM

Files Link Order Targets Processor Expert

- ☒ DisableEvent
- ☒ RecvChar
- ☒ SendChar
- ☒ RecvBlock
- ☒ SendBlock
- ☒ ClearRxBuf
- ☒ ClearTxBuf
- ☒ CharsInRxBuf
- ☒ GetCharsInRxBuf
- ☒ CharsInTxBuf
- ☒ GetCharsInTxBuf
- ☒ SetBaudRateMode
- ☒ GetError
- ☒ GetBreak
- ☒ SetBreak
- ☒ TurnTxOn
- ☒ TurnTxOff
- ☒ TurnRxOn
- ☒ TurnRxOff
- ☒ SetIdle
- ☒ LoopMode
- ☒ ConnectPin
- ☒ GetRxIdle
- ☒ GetTxComplete

User Modules

- SCI_Exercise.c:main

Generated Modules

External Modules

Documentation

PESL

2 files 0 0

SCI_Exercise.c

Path: D:\Data\Technical Training\56800 training\Exam...\SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
    /*** Processor Expert internal initialization. DON'T REMOVE THIS ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.

    /* Write your code here */

    for(;;) {
        AS1_SendChar('H'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('e'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('l'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('l'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('o'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('r'); while(AS1_GetTxComplete()==0){};
        AS1_SendChar('\n'); while(AS1_GetTxComplete()==0){};
    }
}
```

Line 49 Col 55

Communication

- AsynchroMaster
- AsynchroSerial
- AsynchroSlave
- FreescallCAN
- FreescallH18
- FreescallSSI
- Internal2C
- SynchroMaster
- SynchroSlave
- Converter

SetIdle don't generate code

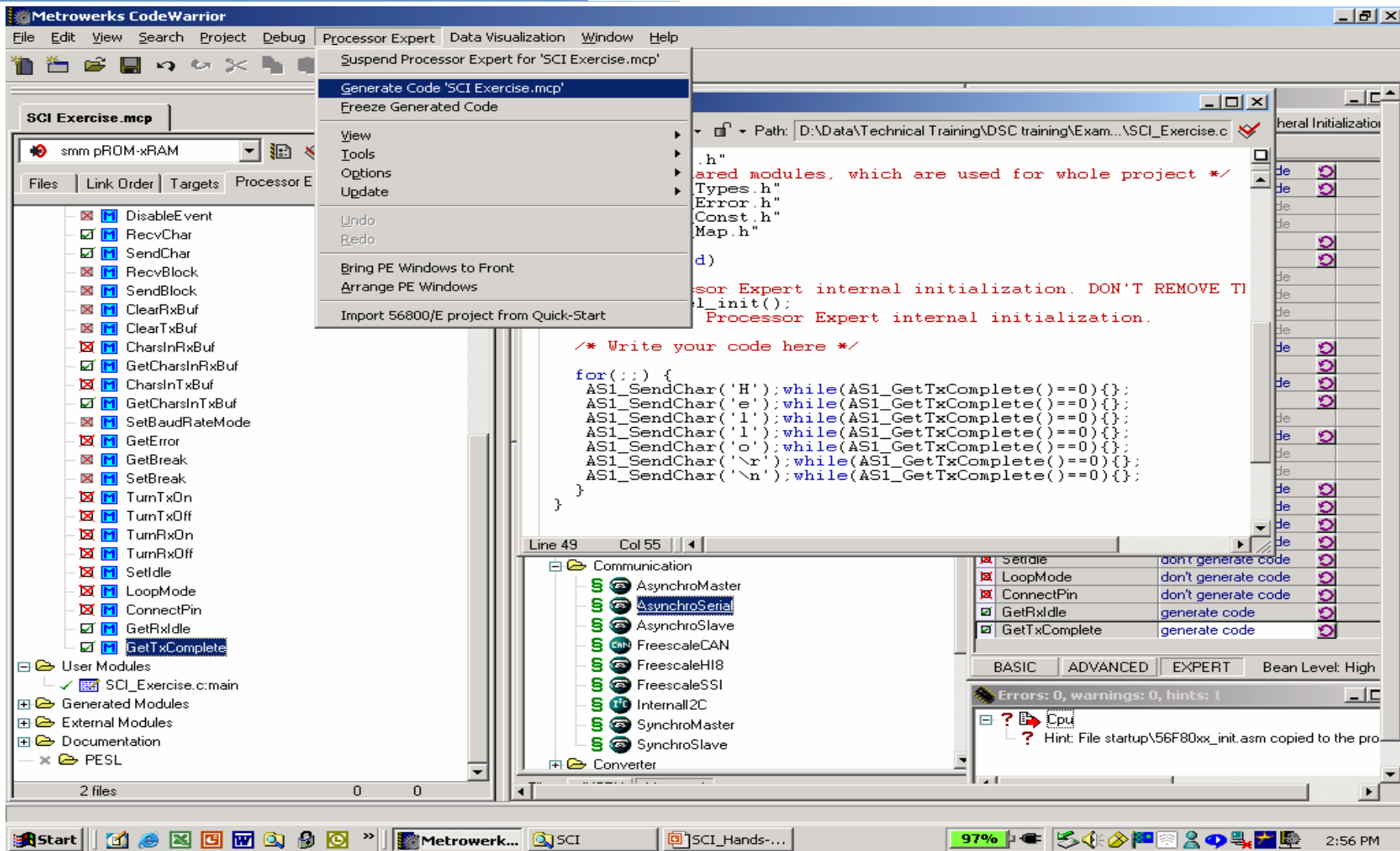
LoopMode don't generate code

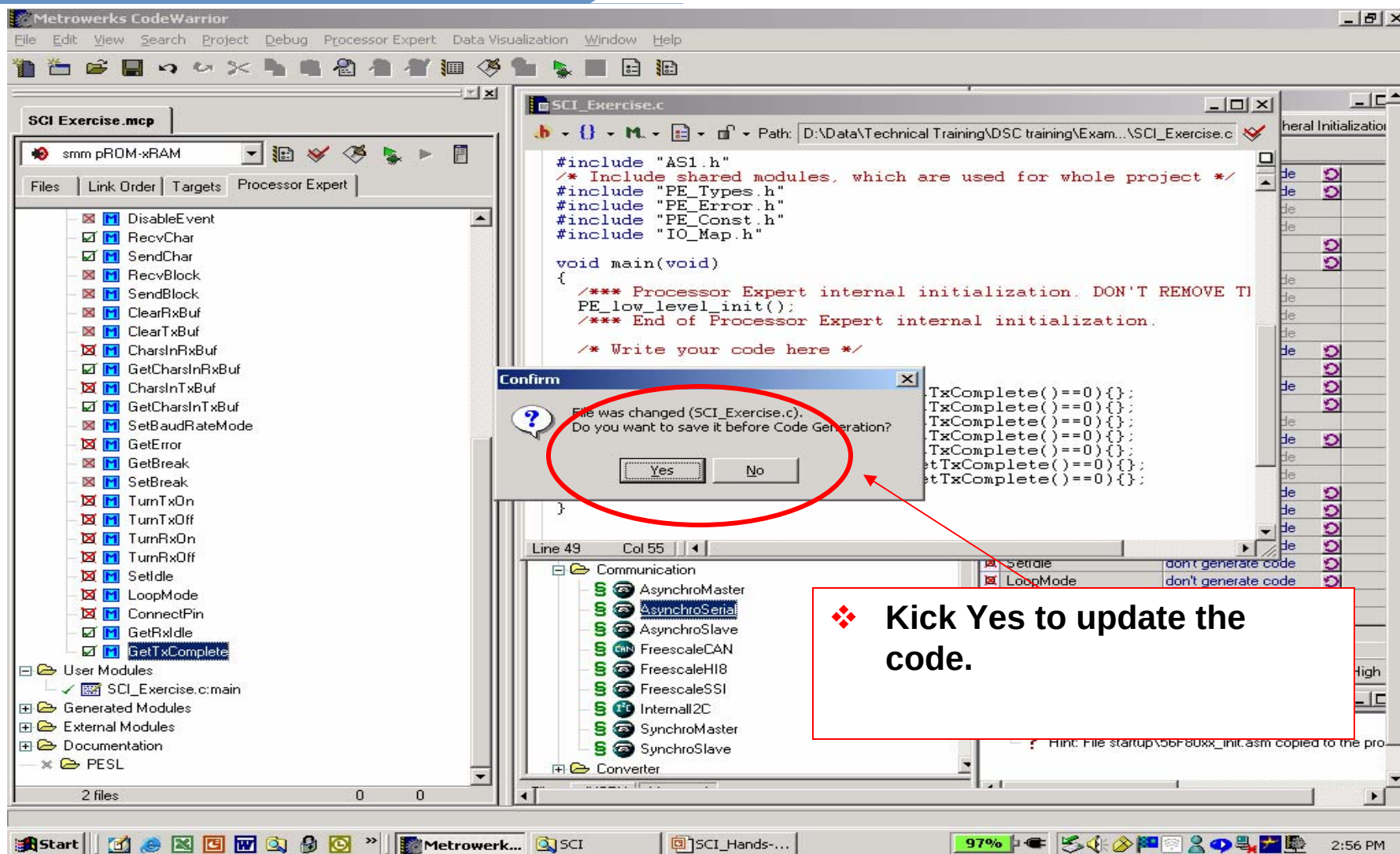
Hint: File startup\56800x_init.asm copied to the pro...

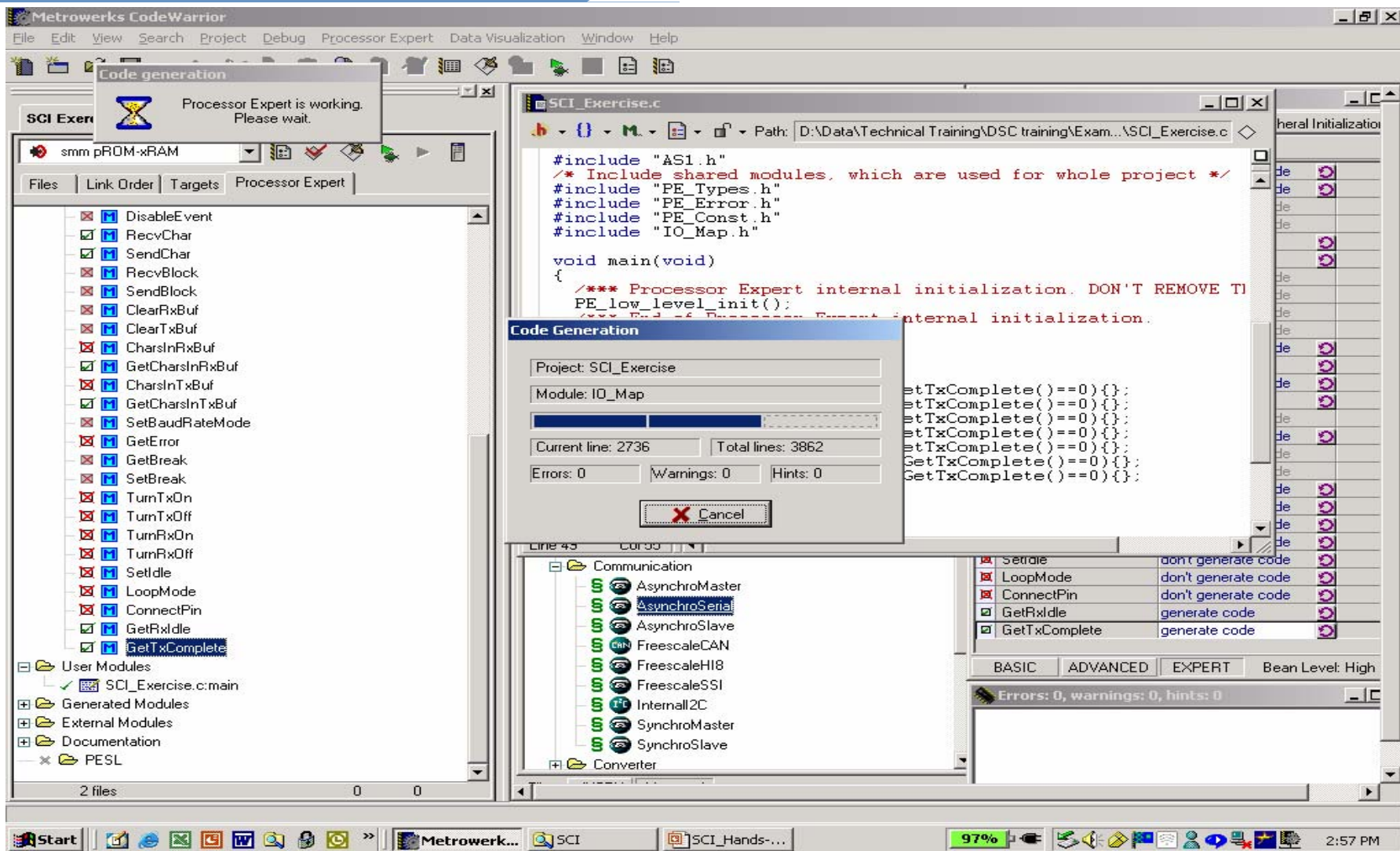
97% 2:56 PM

❖ Make every while statement for every character sending out.

56800/E Training







Metrowerks CodeWarrior

File Edit View Search Project Debug Processor Expert Data Visualization Window Help

SCI_Exercise.mcp

smm pROM-xRAM

Files Link Order Targets Processor Expert **Debug**

DisableEvent
☒ RecvChar
☒ SendChar
☒ RecvBlock
☒ SendBlock
☒ ClearRxBuf
☒ ClearTxBuf
☒ CharsInRxBuf
☒ GetCharsInRxBuf
☒ CharsInTxBuf
☒ GetCharsInTxBuf
☒ SetBaudRateMode
☒ GetError
☒ GetBreak
☒ SetBreak
☒ TurnTxOn
☒ TurnTxOff
☒ TurnRxOn
☒ TurnRxOff
☒ SetIdle
☒ LoopMode
☒ ConnectPin
☒ GetRxIdle
☒ GetTxComplete

User Modules
☒ SCI_Exercise.c:main

Generated Modules
 External Modules
 Documentation
 PESL

2 files 0 0

SCI_Exercise.c

Path: D:\Data\Technical Training\56800 training\Exam...\SCI_Exercise.c

```
#include "AS1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
  /*** Processor Expert internal initialization. DON'T REMOVE THIS ***/
  PE_low_level_init();
  /*** End of Processor Expert internal initialization.

  /* Write your code here */

  for(;;) {
    AS1_SendChar('H'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('e'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('l'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('l'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('o'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('\r'); while(AS1_GetTxComplete()==0){};
    AS1_SendChar('\n'); while(AS1_GetTxComplete()==0){};
  }
}
```

Line 49 Col 55

Communication
☒ AsynchroMaster
☒ AsynchroSerial
☒ AsynchroSlave
☒ FreescaleCAN
☒ FreescaleH18
☒ FreescaleSSI
☒ Internal2C
☒ SynchroMaster
☒ SynchroSlave
☒ Converter

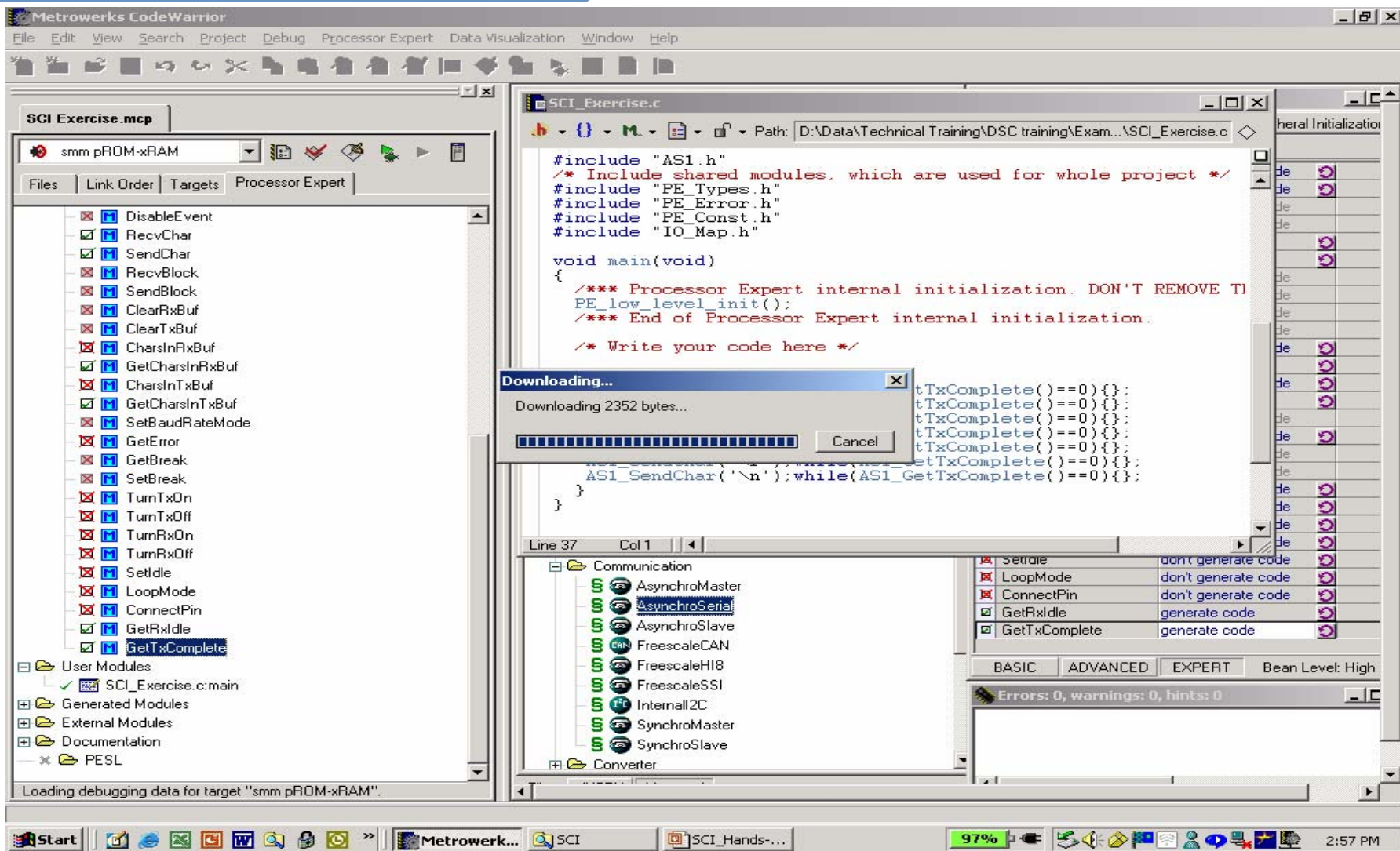
SetIdle don't generate code
 LoopMode don't generate code

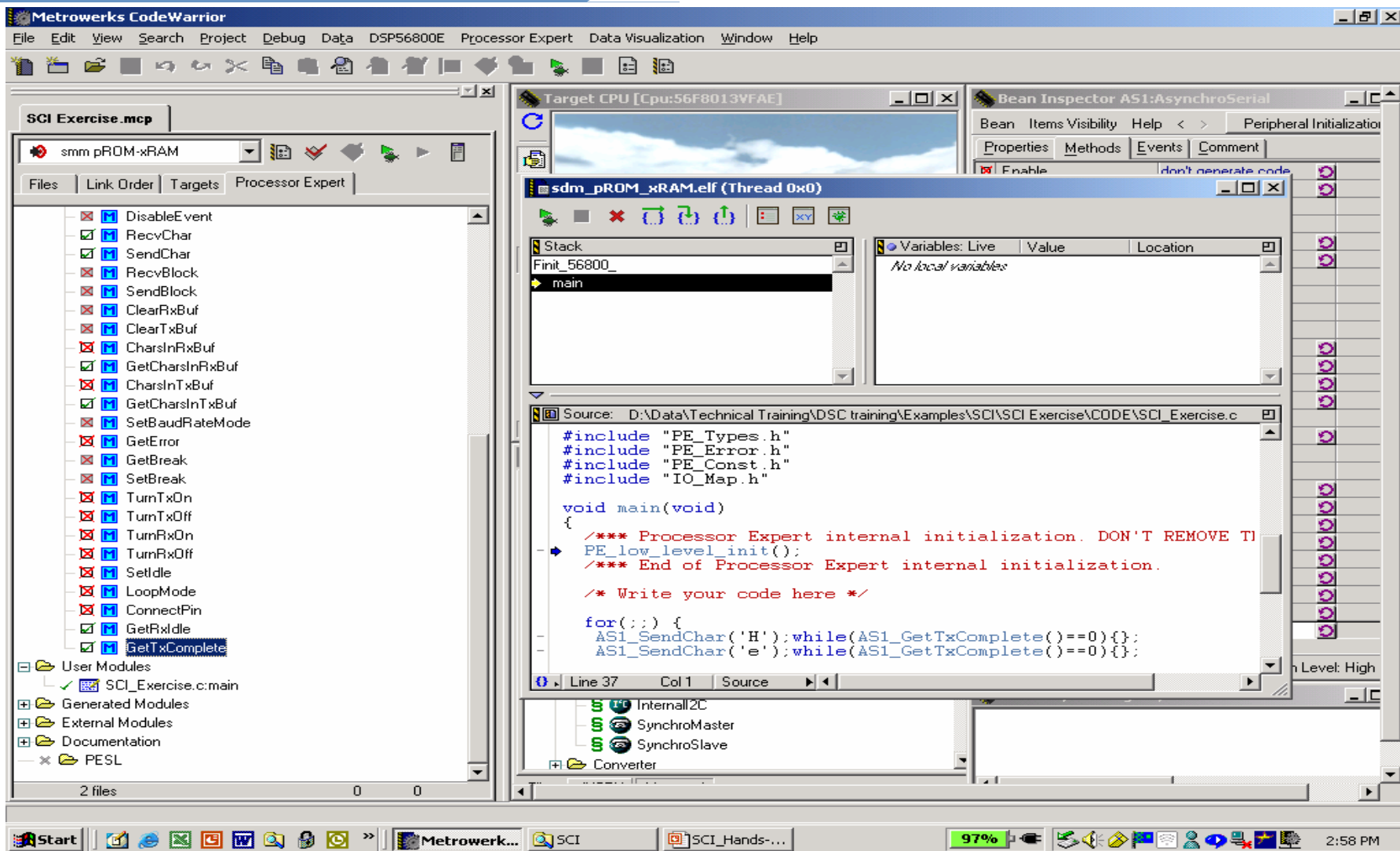
❖ Kick green arrow to download the code to 56F801xEVB .

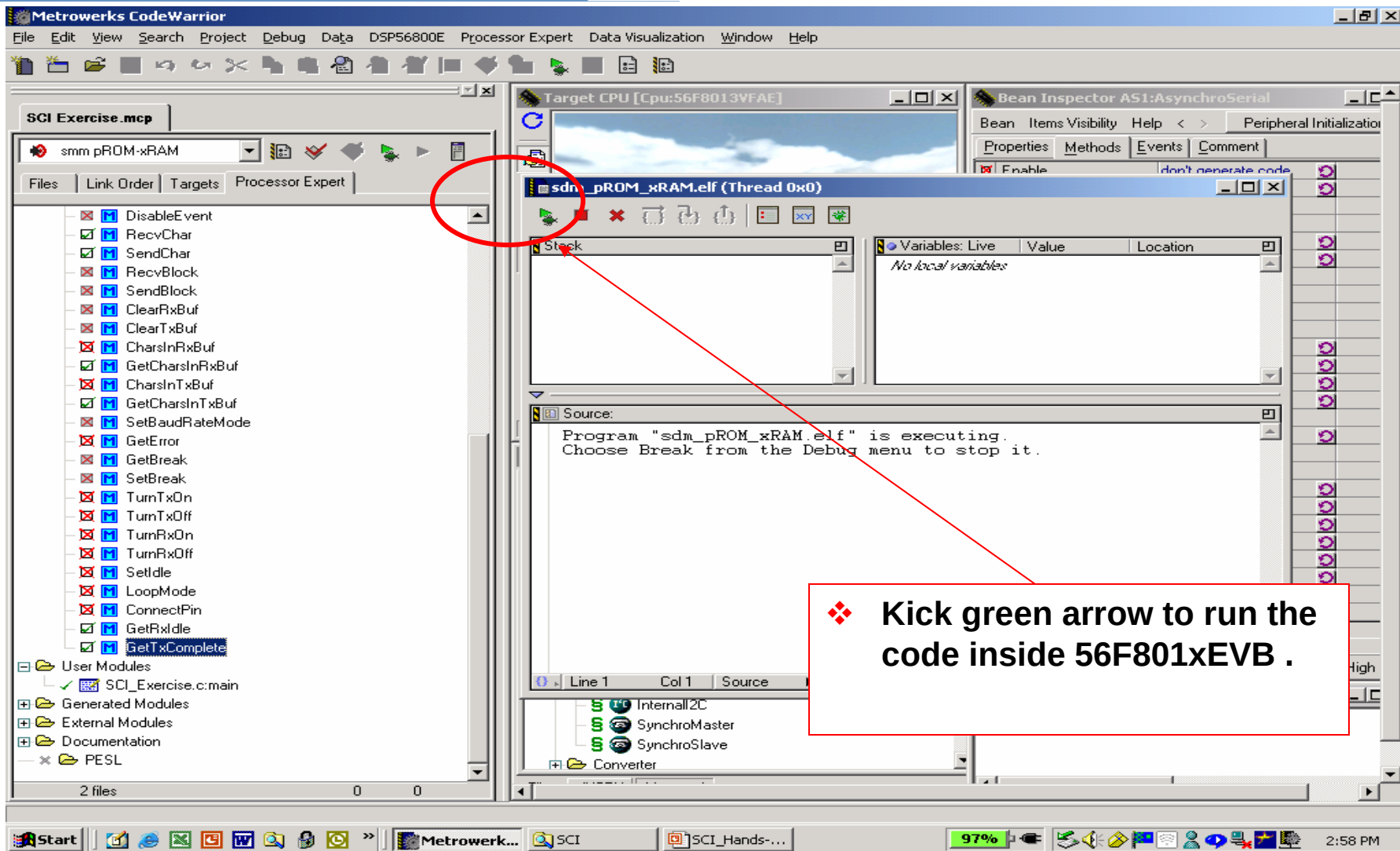
Start Metrowerks... SCI SCI_Hands... 97% 2:57 PM

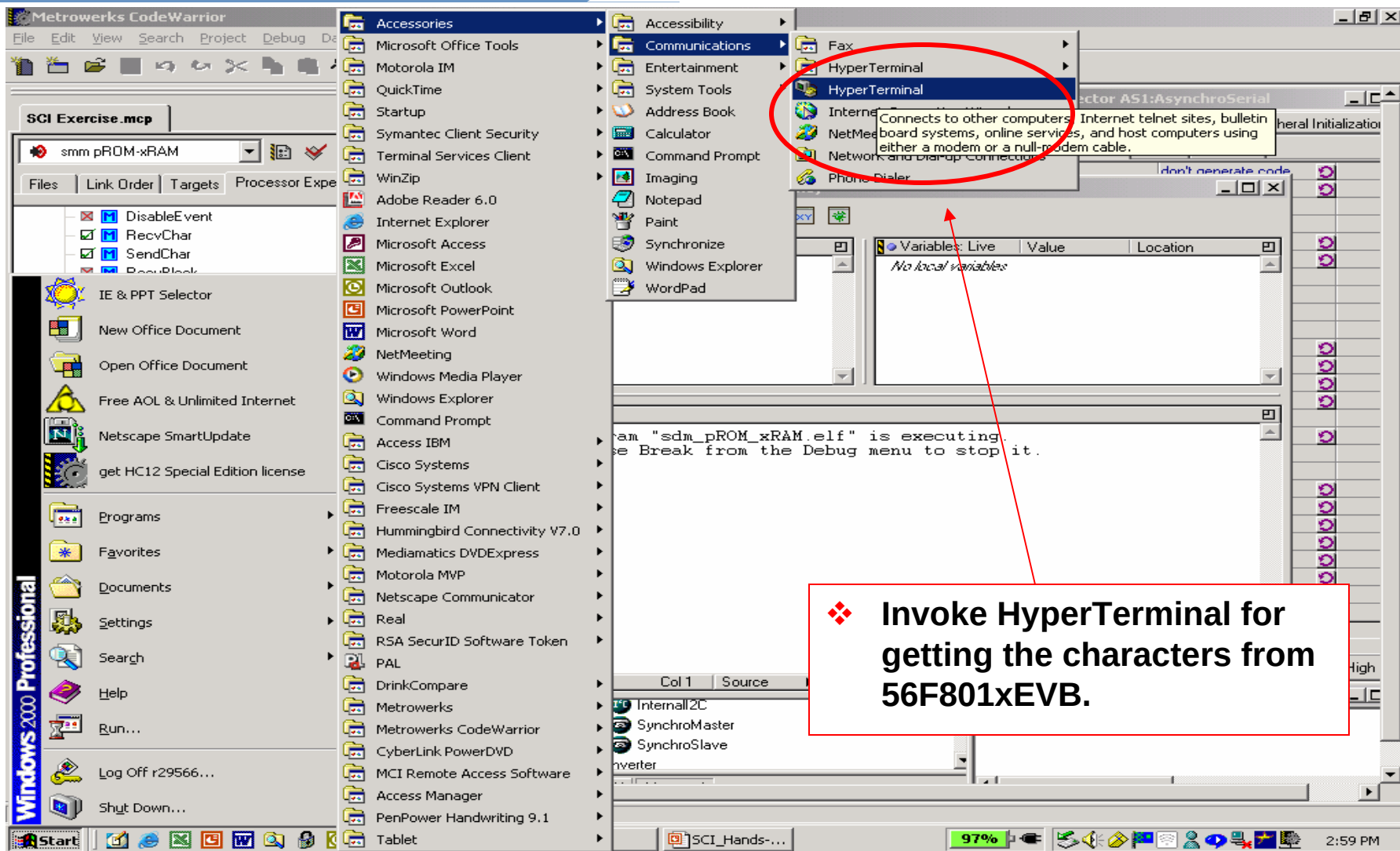
56800/E Training

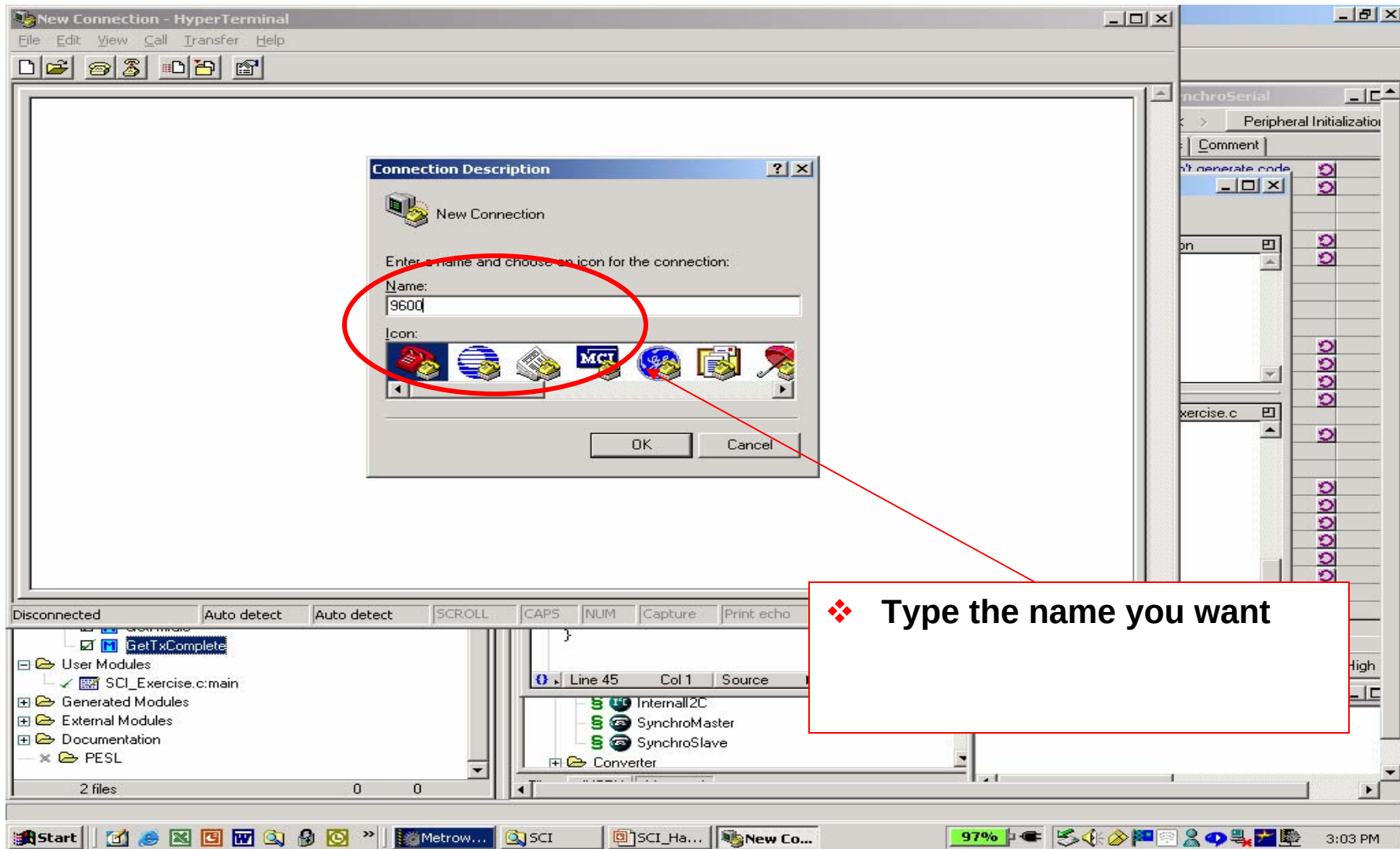
The screenshot shows the Metrowerks CodeWarrior IDE interface. The main window displays the source code for `SCI_Exercise.c`. The code includes headers for `AS1.h`, `PE_Types.h`, `PE_Error.h`, `PE_Const.h`, and `IO_Map.h`. The `main` function is defined, which includes comments for internal initialization and a loop for writing data to the SCI module. A dialog box titled "Building SCI_Exercise.mcp" is open, showing the compilation progress for the file `56F80xx_init.asm`. The dialog indicates that the file is currently compiling, with a total of 9 files and 28030 lines. The left sidebar shows the project structure, including the "User Modules" folder containing `SCI_Exercise.c:main`. The bottom status bar indicates "Errors: 0, warnings: 0, hints: 0".

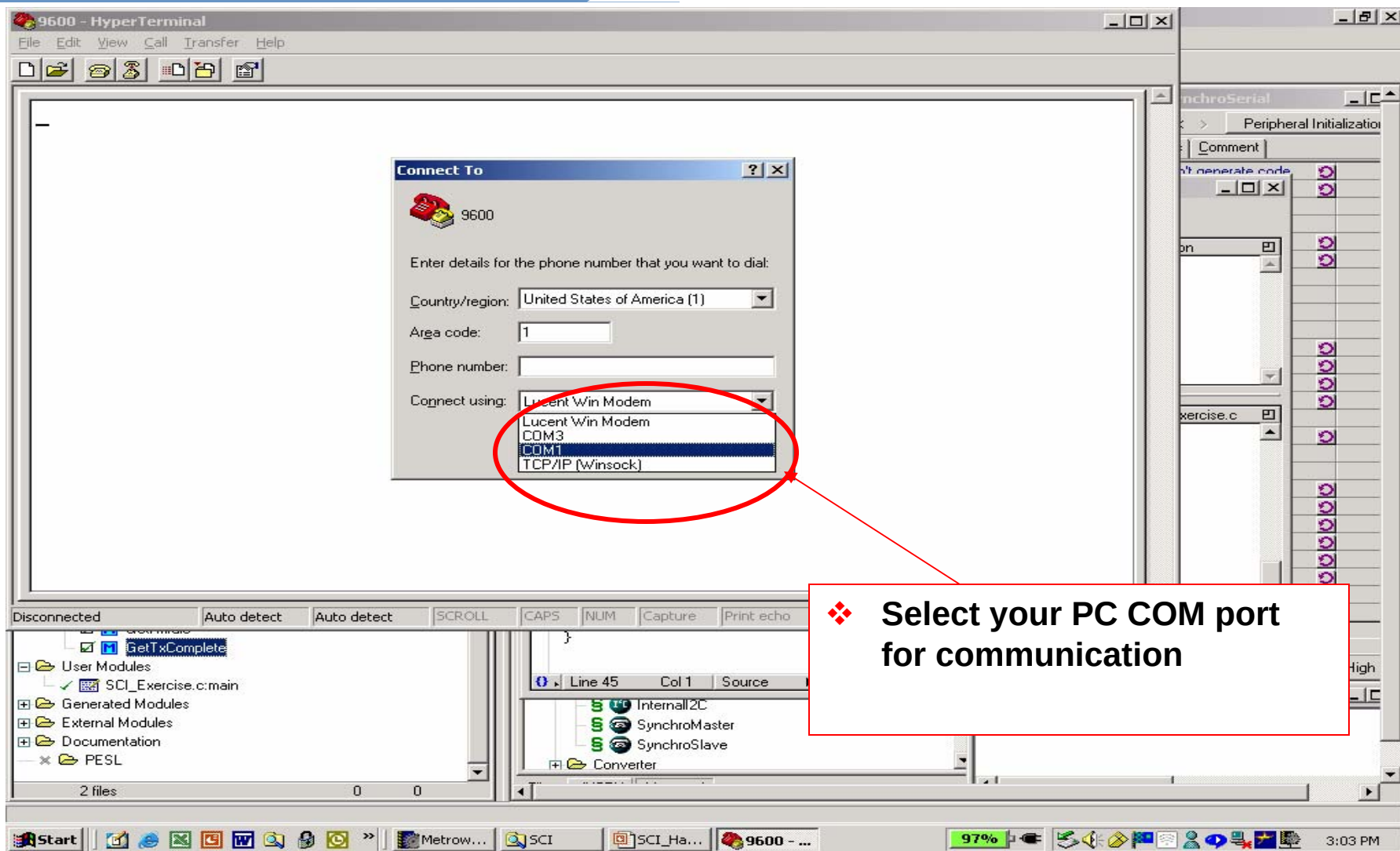












56800/E Training

COM1 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

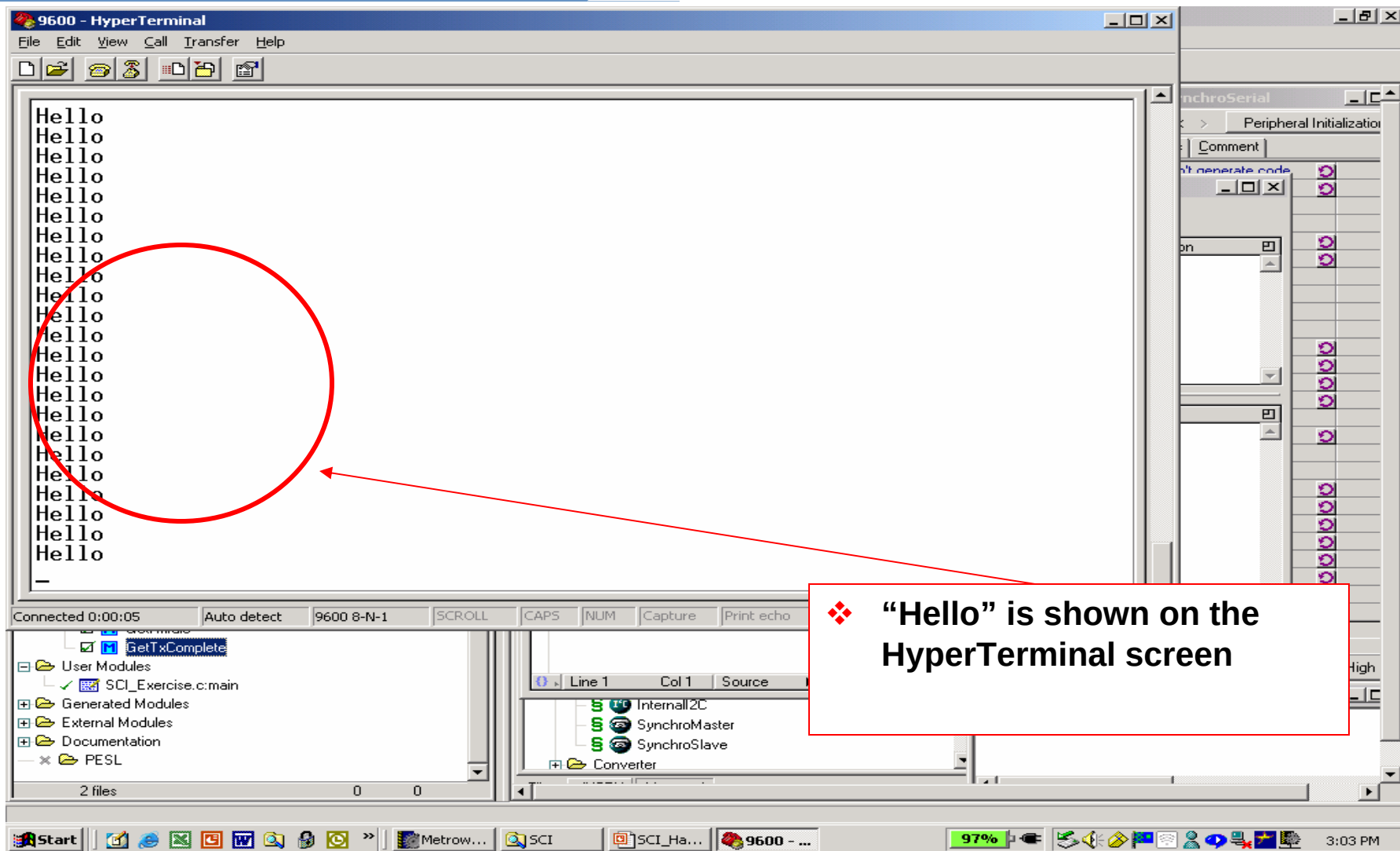
Flow control: None

Restore Defaults

OK Cancel Apply

❖ Select 9600 baud rate, 8 data bit, no parity, 1 stop bit, none flow control

❖ Then, kick OK



Summary

56800/E Training

Summary

- ❖ Understand the hardware and software support available for the 56800E Hybrid Controller product line.
- ❖ Demonstrated the ease of developing applications using CodeWarrior development tools with Processor Expert™ technology.

56800E



Thank You!!!