

Introduction:

This demo program gives an idea about interfacing multiplexed seven segment display with seven segment data input. The interfacing can be verified using a 4 digit counter program which will counts from H'0000 to H'FFFF.

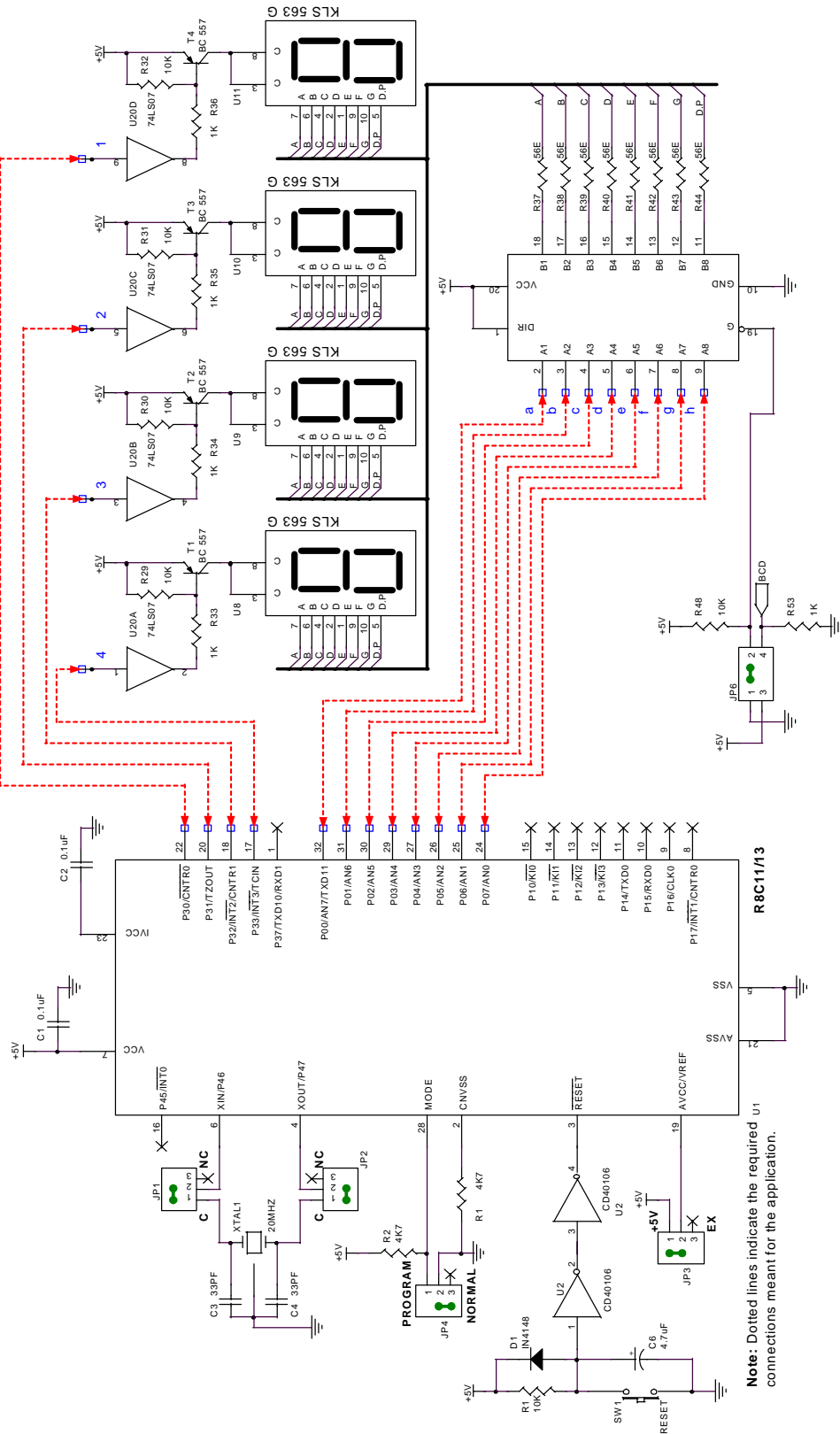
Hardware:

4 digits of seven segment LED displays are connected in multiplexed mode with seven segment data input.

Port line	Display Lines
P00	Segment a
P01	Segment b
P02	Segment c
P03	Segment d
P04	Segment e
P05	Segment f
P06	Segment g
P07	Segment dp
P30	Digit Selection Control for digit 1
P31	Digit Selection Control for digit 2
P32	Digit Selection Control for digit 3
P33	Digit Selection Control for digit 4

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P

Circuit:



Connections:

1. Connect port lines P00 to P07 to seven segment input from a to dp of seven segment display.
2. Connect port lines P30 to P33 to digit selection input of seven segment display.

Functional Description:

In this module, a four digit counter program is provided to study the interfacing of seven segment display in multiplexed mode with seven segment data input. The 4 digit counter will be displayed on the display counting from H'0000 to H'FFFF.

Registers Used:

PD0 - Port 0 Direction Register
PD3 - Port 3 Direction Register
TXMR - Timer X mode Register
PREX - Prescaler X Register
TX - Timer X Register
TCSS - Timer Count Source Setting Register

Port Direction Register:

Symbol	Address	After reset
PD0	00E216	0016
PD1	00E316	0016
PD3	00E716	0016
PD4	00EA16	0016

Bit symbol	Bit name	Function	RW
PDI_0	Port Pi0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PDI_1	Port Pi1 direction bit		RW
PDI_2	Port Pi2 direction bit		RW
PDI_3	Port Pi3 direction bit		RW
PDI_4	Port Pi4 direction bit		RW
PDI_5	Port Pi5 direction bit		RW
PDI_6	Port Pi6 direction bit		RW
PDI_7	Port Pi7 direction bit		RW

Notes:

1. The PD0 register must be written to by the next instruction after setting the PRC2 bit in the PRCR register to "1" (write enabled).
2. Nothing is assigned to the PD3_4 to PD3_6 bits in the PD3 register. When writing to the PD3_4 to PD3_6 bits, write "0" (input mode). When read, its content is indeterminate.
3. Nothing is assigned to the PD4_0 to PD4_4, PD4_6 and PD4_7 bits in the PD4 register. When writing to the PD4_0 to PD4_4, PD4_6 and PD4_7 bits, write "0" (input mode). When read, its content is indeterminate.

PD0 is set to H'FF to select all lines of port 0 (P00 to P07) as output lines to control seven segment display. Port lines P30 to P33 are set as output lines by moving data H'0F to PD3 register.

TXMR - Timer X Mode Register:

Symbol	Address	After reset
TXMR	008B16	0016

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b15:0 0 0 : Timer mode or pulse period measurement mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
ROEDG	INT1/CNTR0 polarity switching bit ¹	Function varies with each operation mode	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/CNTR0 select bit	Function varies depending on operation mode	RW
TXMOD2	Operation mode select bit 2	0 : Except in pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Active edge reception flag	Function varies depending on operation mode.	RW
TXUND	Timer X under flow flag	Function varies depending on operation mode.	RW

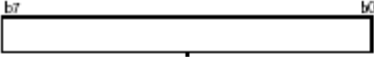
Notes:

1. The IR bit in the INT1IC register may be set to "1" (interrupt requested) when the ROEDG bit is rewritten. Refer to the paragraph 19.2.5 "Changing Interrupt Factor" in the Usage Notes Reference Book.

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P

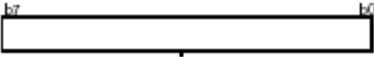
Timer X mode register TXMR is loaded with H'00 to set timer mode.

PREX -Prescaler X Register:

	Symbol PREX	Address 008C16	After reset FF16																									
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Mode</th> <th style="width: 45%;">Function</th> <th style="width: 15%;">Setting range</th> <th style="width: 15%;">RW</th> </tr> </thead> <tbody> <tr> <td>Timer mode</td> <td>Internal count source is counted</td> <td>0016 to FF16</td> <td>RW</td> </tr> <tr> <td>Pulse output mode</td> <td>Internal count source is counted</td> <td>0016 to FF16</td> <td>RW</td> </tr> <tr> <td>Event counter mode</td> <td>Externally input pulses are counted</td> <td>0016 to FF16</td> <td>RW</td> </tr> <tr> <td>Pulse width measurement mode</td> <td>Pulse width of externally input pulses is measured (Internal count source is counted)</td> <td>0016 to FF16</td> <td>RW</td> </tr> <tr> <td>Pulse period measurement mode</td> <td>Pulse period of externally input pulses is measured (Internal count source is counted)</td> <td>0016 to FF16</td> <td>RW</td> </tr> </tbody> </table>				Mode	Function	Setting range	RW	Timer mode	Internal count source is counted	0016 to FF16	RW	Pulse output mode	Internal count source is counted	0016 to FF16	RW	Event counter mode	Externally input pulses are counted	0016 to FF16	RW	Pulse width measurement mode	Pulse width of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW	Pulse period measurement mode	Pulse period of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW
Mode	Function	Setting range	RW																									
Timer mode	Internal count source is counted	0016 to FF16	RW																									
Pulse output mode	Internal count source is counted	0016 to FF16	RW																									
Event counter mode	Externally input pulses are counted	0016 to FF16	RW																									
Pulse width measurement mode	Pulse width of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW																									
Pulse period measurement mode	Pulse period of externally input pulses is measured (Internal count source is counted)	0016 to FF16	RW																									

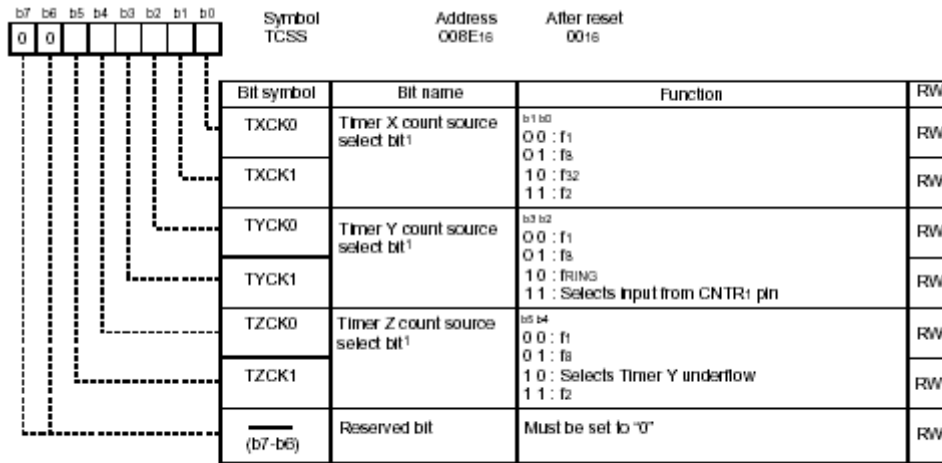
The prescaler register PREX is loaded with 100 to divide the input selected clock to timer X by 100.

TX - Timer X Register:

	Symbol TX	Address 008D16	After reset FF16							
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 55%;">Function</th> <th style="width: 15%;">Setting range</th> <th style="width: 30%;">RW</th> </tr> </thead> <tbody> <tr> <td>Underflow of Prescaler X is counted</td> <td>0016 to FF16</td> <td>RW</td> </tr> </tbody> </table>				Function	Setting range	RW	Underflow of Prescaler X is counted	0016 to FF16	RW
Function	Setting range	RW								
Underflow of Prescaler X is counted	0016 to FF16	RW								

The timer X register TX loaded with 100 to generate an interrupt at 1 milli second rate.

TCSS - Timer Count Source Setting Register:



Notes:
1. Avoid switching a count source, while a counter is in progress. Timer counter must be stopped before switching a count source.

The data H'03 is moved to Timer Count Source Setting Register TCSS to select "f2" as source clock for Timer X.

Software Description:

In this module, a four digit counter program is provided to study the interfacing of seven segment display in multiplexed mode with seven segment data input.

Timer X is used to generate an interrupt at a rate of one milli second. In the timer X interrupt service routine, the seven segment display is refreshed and the milli seconds are counted to get one second delay. This milli second counter is used in main loop to get 1 second delay.

In the main loop, the timer X is initialized to generate interrupt at 1KHZ rate to refresh the display and wait for one second. After a second, the count value is increment by one and displayed on the 4 digit display. The process is repeated.

The files used in this module are listed below:

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P

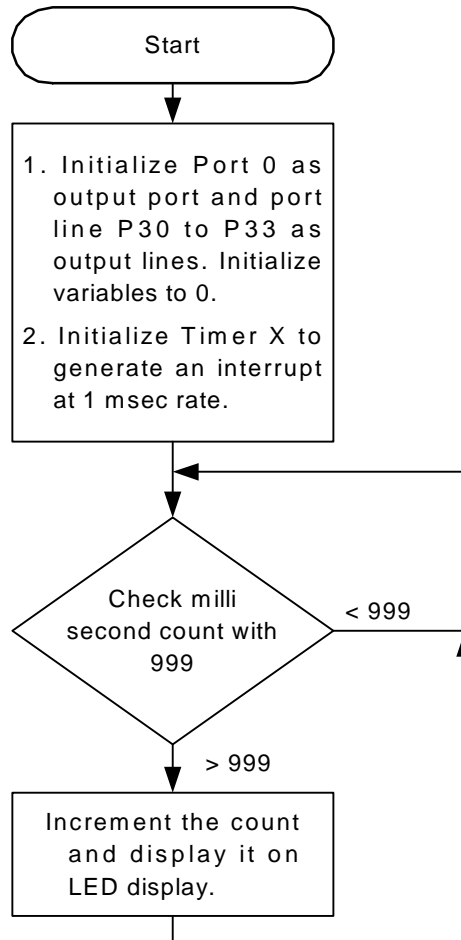
<i>Files</i>	<i>Description</i>
Demo21.C	The only file for this module, has the processing function for the 4 digit counter , timer X interrupt service routine and initialization routines for seven segment display, timer X etc.

The functions in the file “Demo21.C” and short descriptions are listed below:

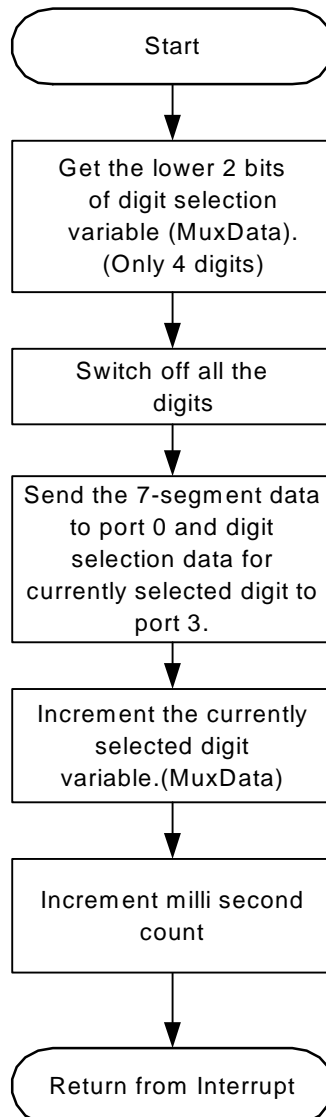
<i>Functions</i>	<i>Description</i>
Main	A 4 digit counter which will counts from H'0000 to H'FFFF. Input: None. Output : None.
ProcessTimer_X_Int	Timer X interrupt service routine. Here the seven segment display was refreshed and the milli seconds are counted to get one second delay. Input: None. Output : None.
InitializeTimerX	Timer X is initialized to generate an interrupt at a rate of one milli second. Input: None. Output : None.
InitializeSevenSegmentLED	This routine will initialize I/O lines used for seven segment display, Timer X and variables used for refreshing display. Input: None. Output : None.
Display4Digits	Displays the given 4 digit number on seven segment display. Input: 4 Digit number. Output : None.

Program Flow:

For Main Loop



For Timer X Interrupt Service Routine



Execute Application:

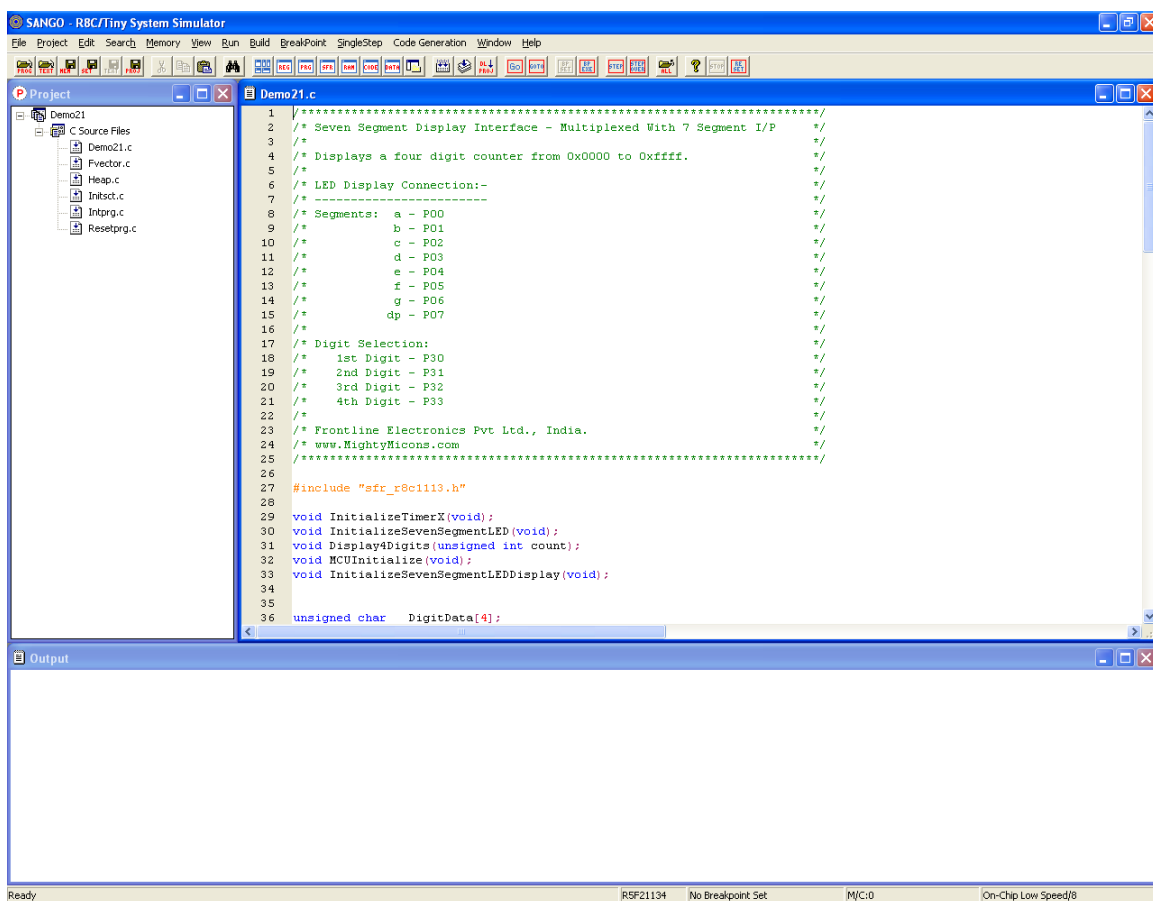
After reset, the 4 digit counter will be displayed on the display counting from H'0000 to H'FFFF.

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P

Use Topview Simulator to Verify the Design.

Open the project Demo21 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo21.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler output.

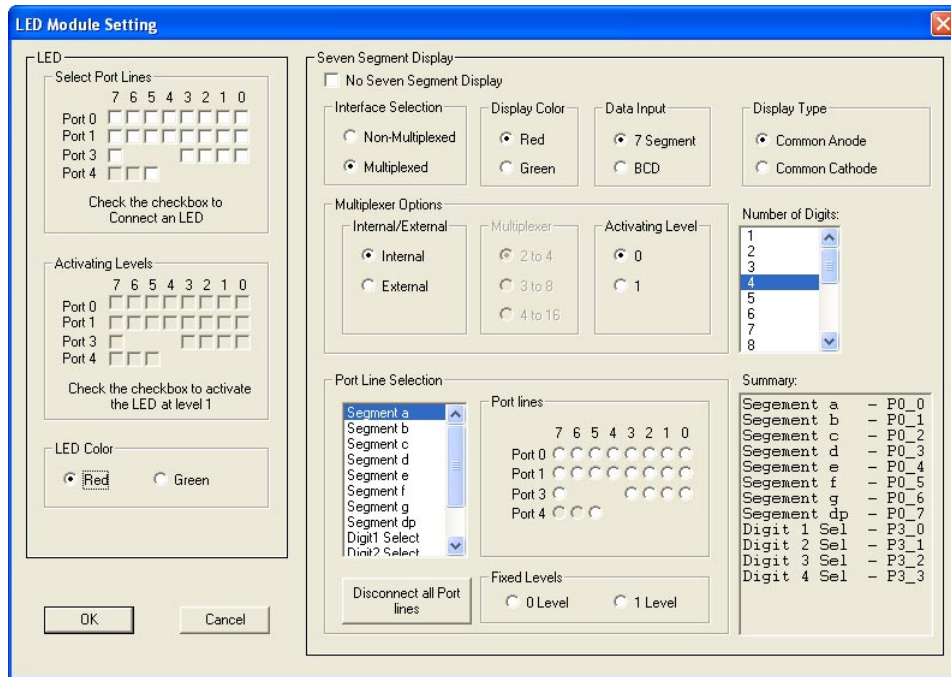
Use **Project -> Download Project** from main menu to download the .mot file into the simulator's memory for simulation.



```
1 /******  
2 /* Seven Segment Display Interface - Multiplexed With 7 Segment I/P */  
3 /*  
4 /* Displays a four digit counter from 0x0000 to 0xffff.  
5 /*  
6 /* LED Display Connection:-  
7 /* -----  
8 /* Segments: a - P00  
9 /*           b - P01  
10 /*          c - P02  
11 /*          d - P03  
12 /*          e - P04  
13 /*          f - P05  
14 /*          g - P06  
15 /*          dp - P07  
16 /*  
17 /* Digit Selection:  
18 /* 1st Digit - P30  
19 /* 2nd Digit - P31  
20 /* 3rd Digit - P32  
21 /* 4th Digit - P33  
22 /*  
23 /* Frontline Electronics Pvt Ltd., India.  
24 /* www.MightyMicons.com  
25 /******  
26  
27 #include "sfr_r8c1113.h"  
28  
29 void InitializeTimerX(void);  
30 void InitializeSevenSegmentLED(void);  
31 void Display4Digits(unsigned int count);  
32 void MCUInitialize(void);  
33 void InitializeSevenSegmentLEDDisplay(void);  
34  
35  
36 unsigned char DigitData[4];
```

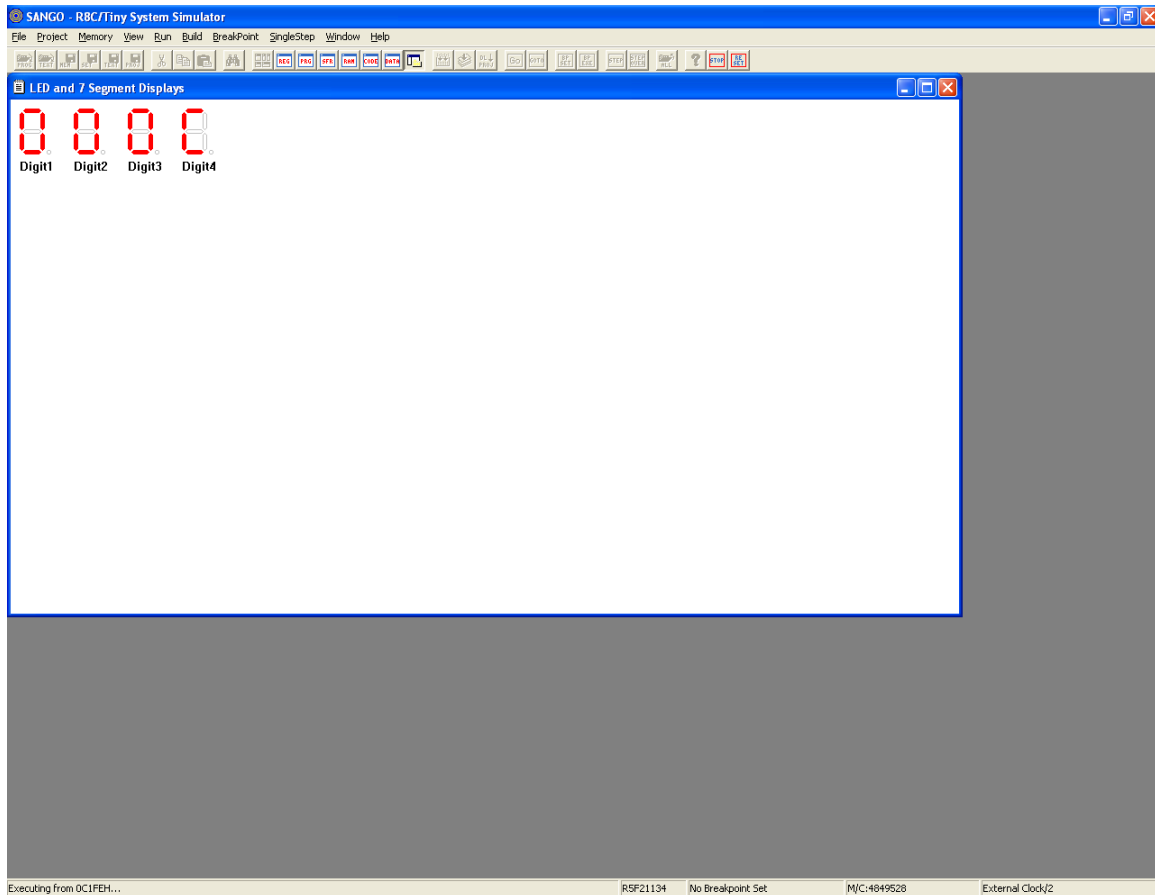
Do the settings to the LED modules as shown. Connect seven segment LEDdisplay control and segment lines to port lines P30 to P33 and P00 to P07.

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P



Then open the LED window and arrange them as shown below.

Demo 21 - 4 Digit Multiplexed 7-Segment Display With 7 Segment I/P



Download the program using **Download Project** command in **Project** menu.

Run the program using **Go** command in **Run** menu.

A 4 digit counter will be displayed on the display counting from H'0000 to H'FFFF.