Introduction:

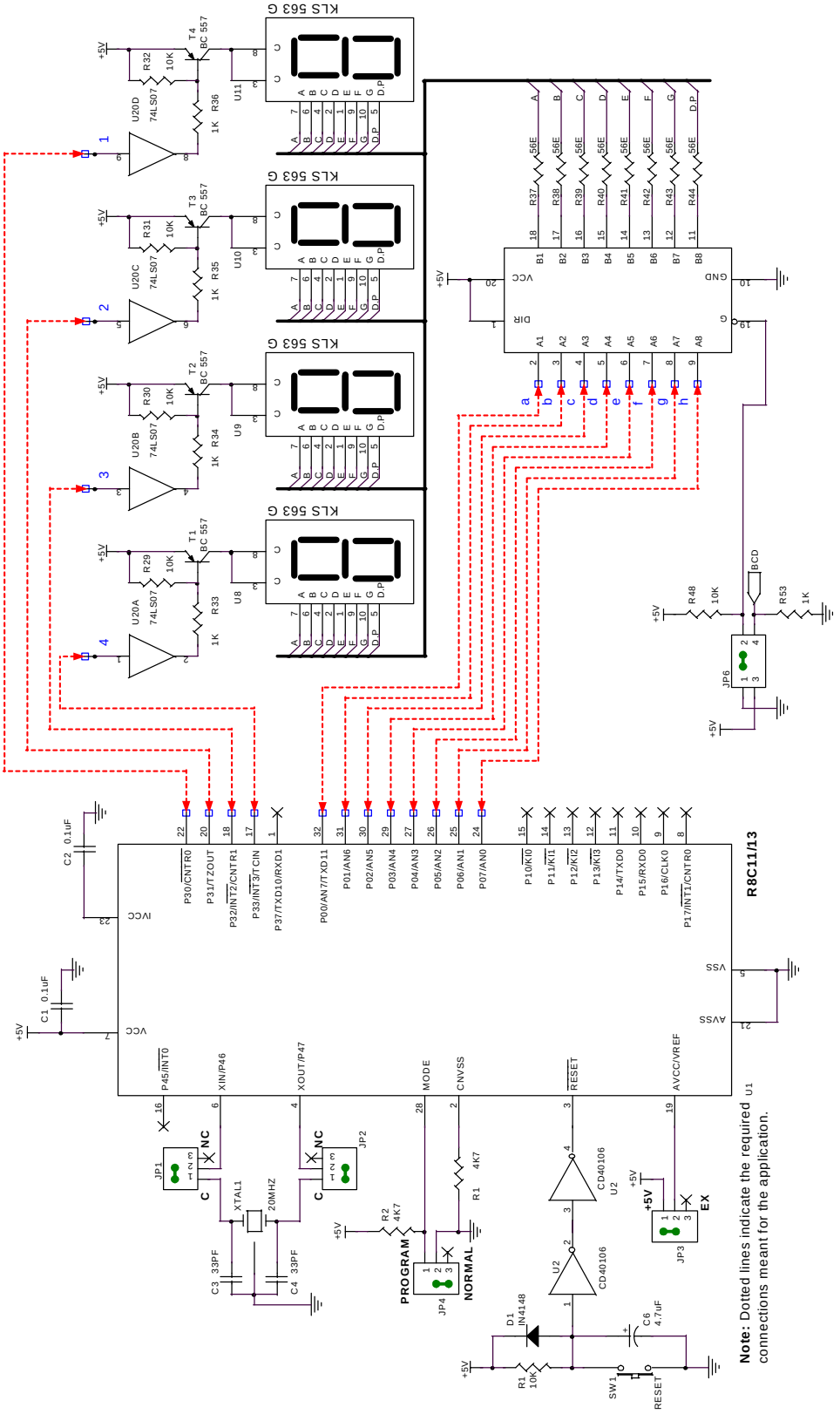
This demo gives an idea about interfacing multiplexed seven segment display with seven segment data input. The interfacing can be verified using a 4 digit counter program which will counts from H'0000 to H'FFFF.

Demo Hardware:

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

Connect P0 to the seven segments sequentially and the Port lines P30, P31, P32 and P33 to the digit selection lines.

Circuit Connection:



Connections:

Port lines	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

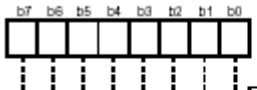
Functional Description:

In this demo, 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:



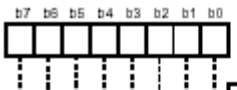
Symbol	Address	After reset
PD0	00E2₁₆	00₁₆
PD1	00E3₁₆	00₁₆
PD3	00E7₁₆	00₁₆
PD4	00EA₁₆	00₁₆

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:



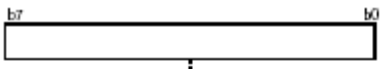
Symbol	Address	After reset
TXMR	008B₁₆	00₁₆

Bit symbol	Bit name	Function	RW
TXMOD0	Operation mode select bit 0, 1	b1 b0 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
R0EDG	$\overline{INT}$1/CNTR0 polarity switching bit[1]	Function varies with each operation mode	RW
TXS	Timer X count start flag	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P30/$\overline{CNTR_0}$ 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 R0EDG bit is rewritten. Refer to the paragraph 19.2.5 "Changing Interrupt Factor" in the Usage Notes Reference Book.

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

PREX -Prescaler X Register:

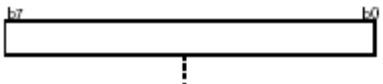


Symbol	Address	After reset
PREX	008C16	FF16

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:

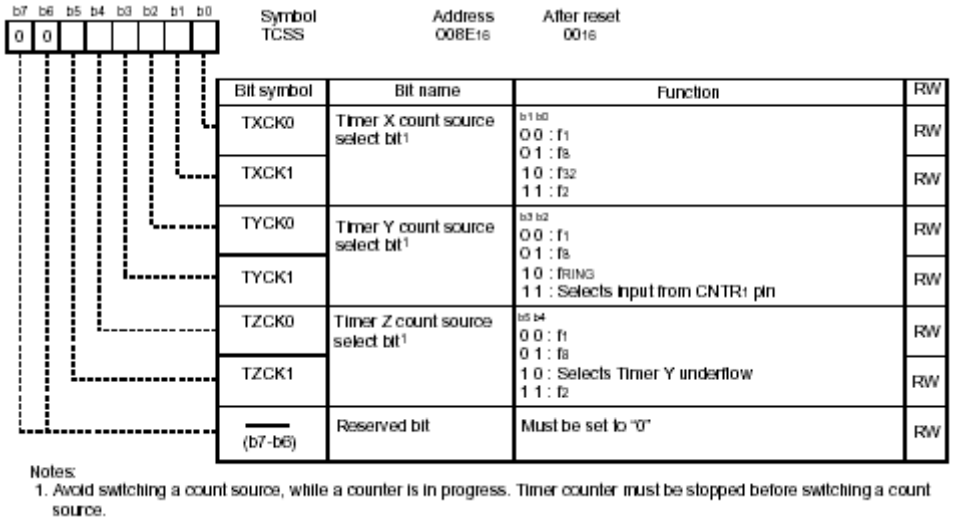


Symbol	Address	After reset
TX	008D16	FF16

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:



b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol: TCSS | Address: 008E16 | After reset: 0016

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit[1]	b1 b0 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : f2	RW
TXCK1			RW
TYCK0	Timer Y count source select bit[1]	b3 b2 0 0 : f1 0 1 : f8 1 0 : fRING 1 1 : Selects input from CNTR1 pin	RW
TYCK1			RW
TZCK0	Timer Z count source select bit[1]	b5 b4 0 0 : f1 0 1 : f8 1 0 : Selects Timer Y underflow 1 1 : f2	RW
TZCK1			RW
— (b7-b6)	Reserved bit	Must be set to "0"	RW

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 demo, 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 demo are listed below:

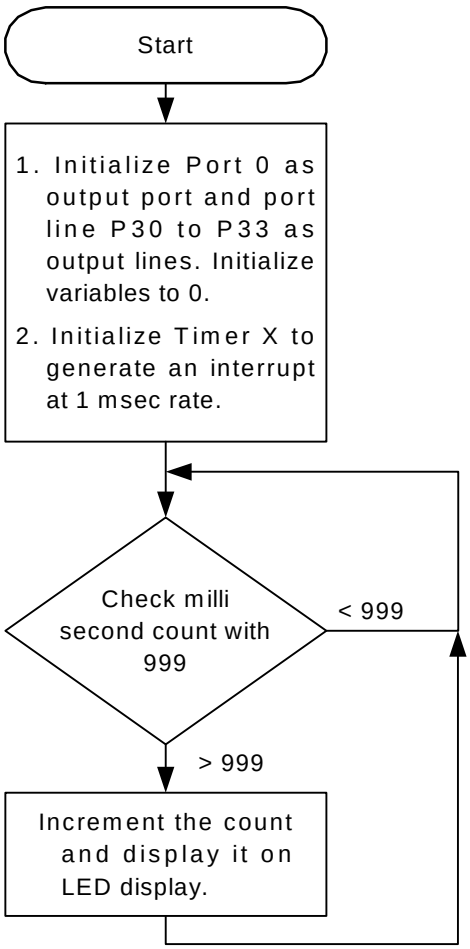
Files	*Description*
Demo2.C	The only file for this module, has the processing function for the 4 digit counter , timer X interrupt service routine and initialization routines for seven segment display, timer X etc.

The functions in the file **Demo2.C** and short descriptions are listed below:

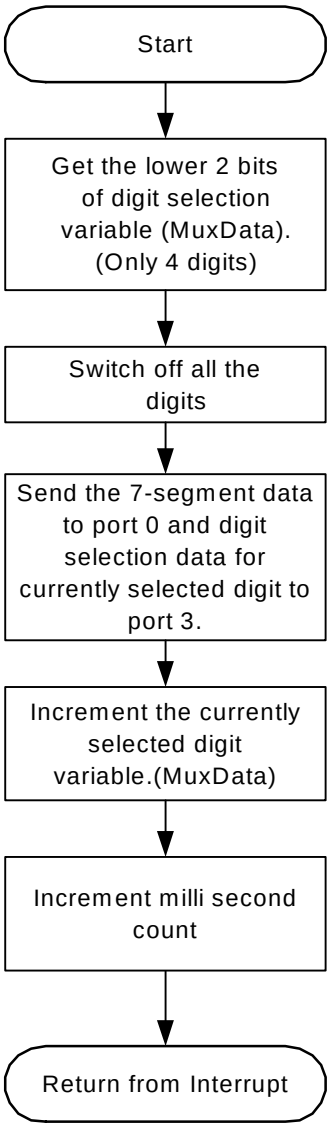
Files	*Description*
main	A 4 digit counter which will counts from H'0000 to H'FFFF. **Input**: None. **Output** : None.
LEDInterrupt	Timer X interrupt service routine. Here the seven segment display was refreshed and the milli seconds are counted to get one second delay. **Input**: None. **Output** : None.
InitLEDDisplay	Timer X is initialized to generate an interrupt at a rate of one milli second.Timer X and variables used for refreshing display. **Input**: None. **Output** : None.
Initialize7SegmentMuxDisplay	This routine will initialize I/O lines used for seven segment 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



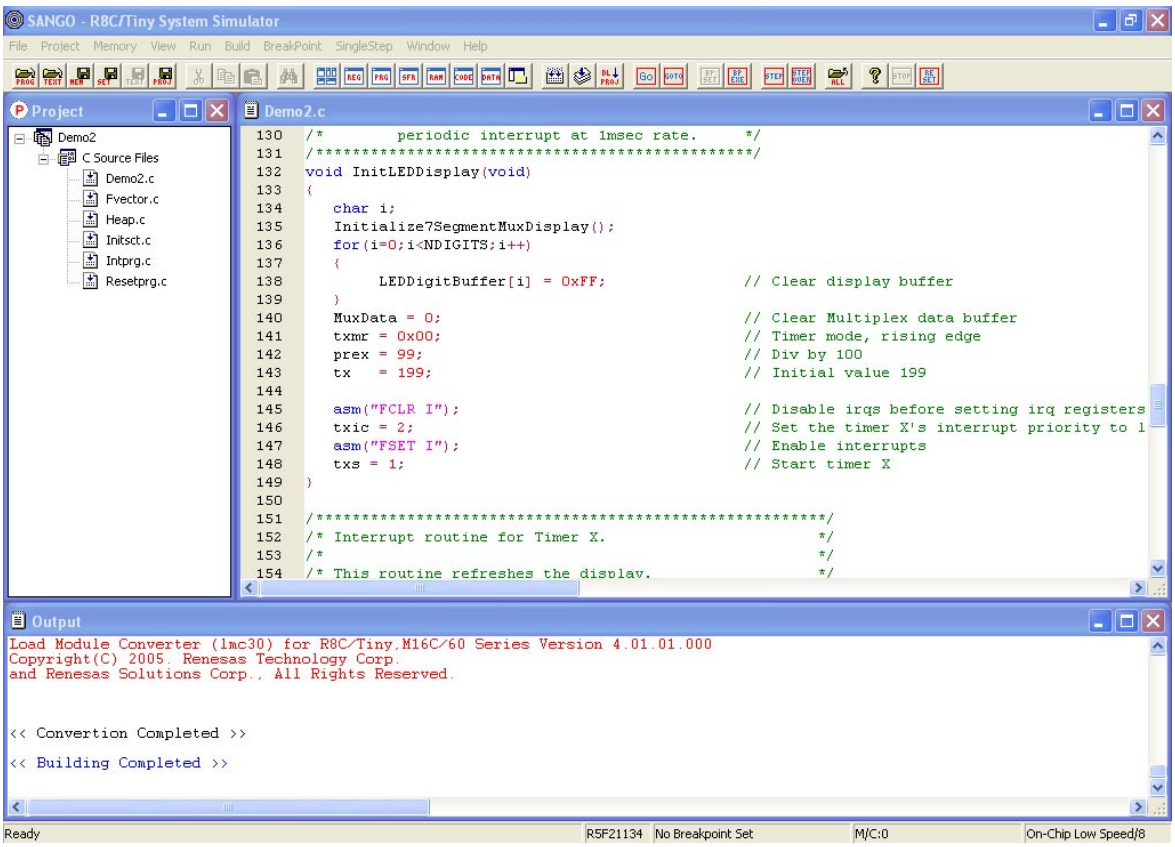
Execute Demo:

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

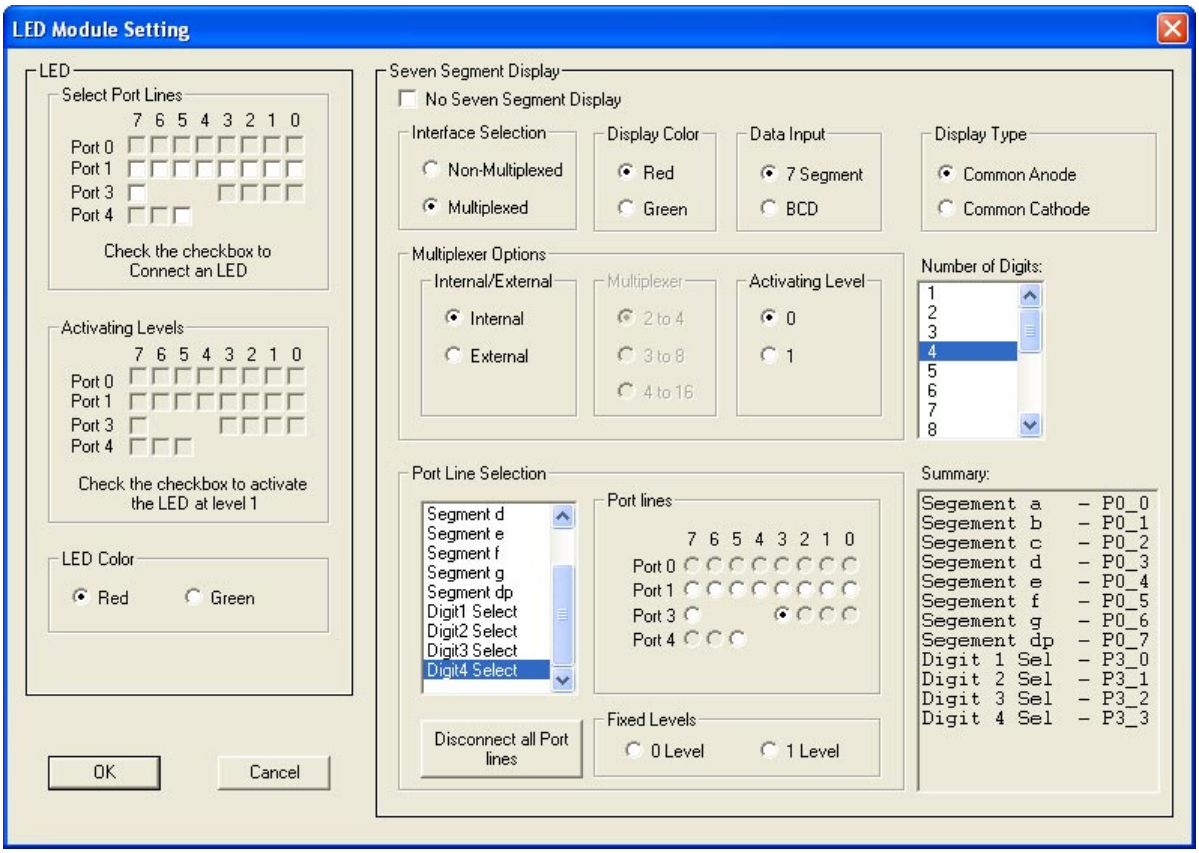
Use Topview Simulator to Verify the Design.

Open the project Demo2 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo2.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

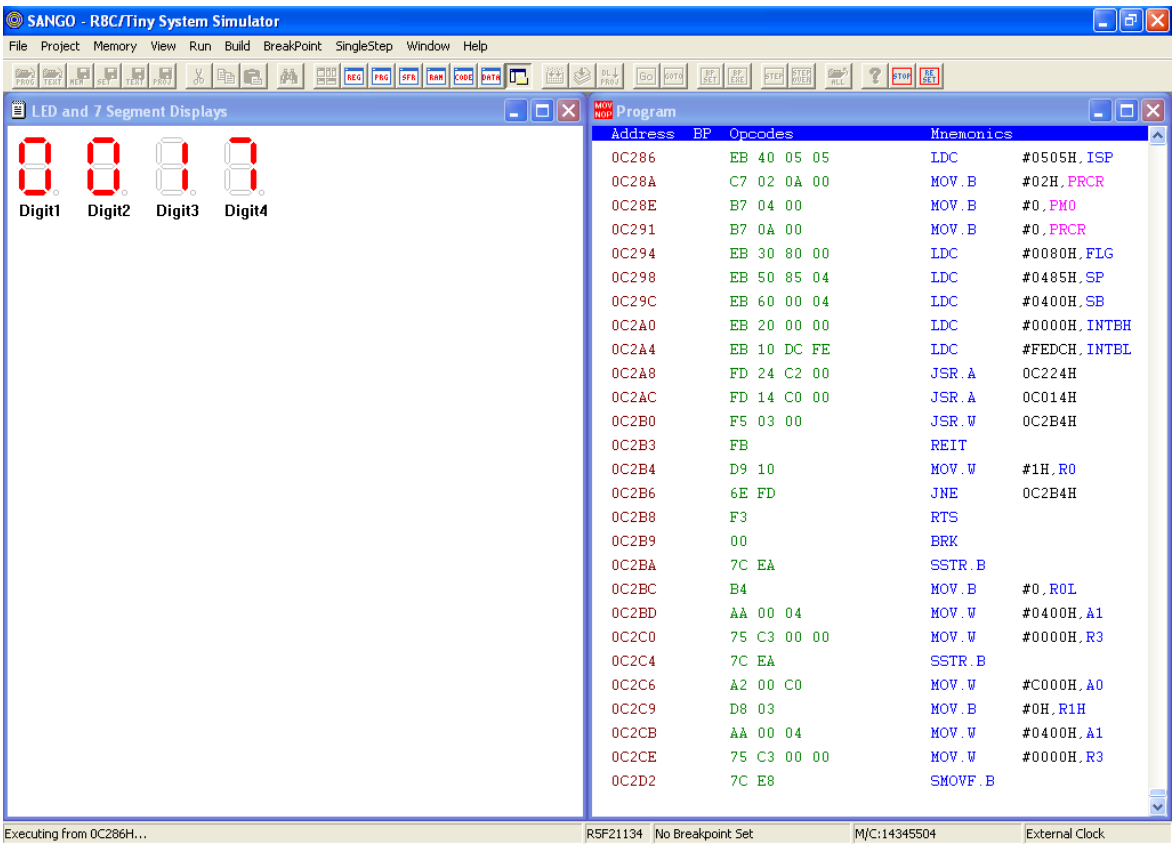
Use **Project** -> **Download Project** from main menu to download the Demo2.mot file into the simulator's memory for simulation.



Open the LED Module settings window and do the settings to the 7-segment LED module as shown. Connect 7 segments of the display to the port lines P00 to P07 and the 4 digit selection lines to P30, P31, P32 and P33 respectively using radio buttons.



Then open the **LED window** usign the option **View** -> **External Modules** -> **LED** as shown below and the Program Window.



Run the program using **Go** from the **Run** menu. The program will display a 4 digit counter in the 7-segment display and counts from H'0000 to H'FFFF and it is repeated.