

Introduction:

This Demo discusses a complete application based on R8C11/13 microcontroller. The application is based on a digital clock with the simple interaction. Four digits of seven segment LED display are used to display the stop clock along with three switches for the operation.

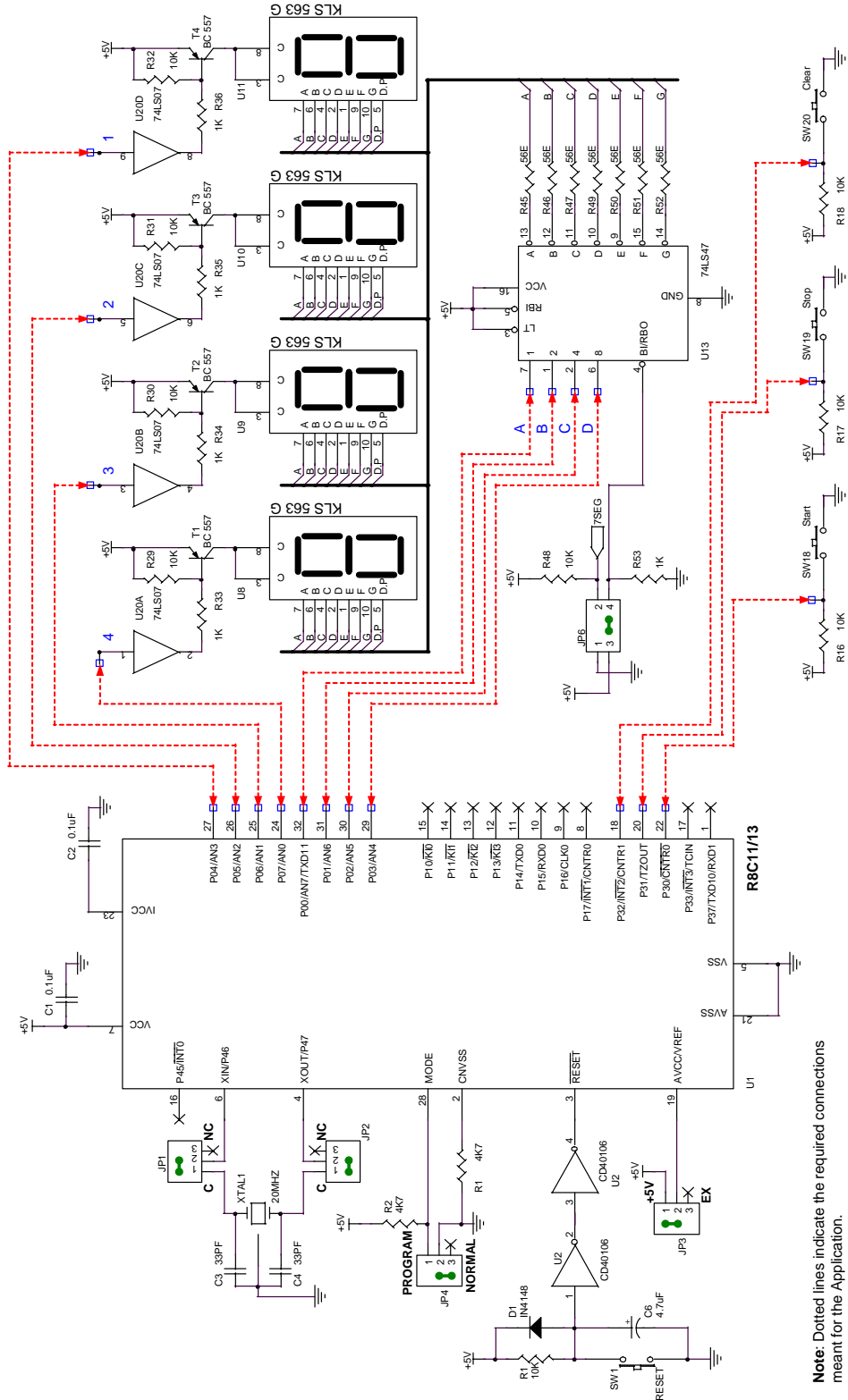
Demo Hardware:

The hardware contains 4 digits of seven segment LED displays connected in the multiplexed format. Three push button switches are connected to port lines to get Start, Stop and Clear commands.

Port line	Display Lines
P00	BCD I/P A
P01	BCD I/P B
P02	BCD I/P C
P03	BCD I/P D
P04	Digit Selection Control for digit 1
P05	Digit Selection Control for digit 2
P06	Digit Selection Control for digit 3
P07	Digit Selection Control for digit 4

Port line	Push Button Swicthes
P30	Start Switch
P31	Stop Switch
P32	Clear Switch

Circuit Connection Using FE EADS:



Connections:

1. Connect port lines P00 to P03 to BCD inputs of seven segment display.
2. Connect port lines P04 to P07 to digit selection input of seven segment display.
3. Connect port lines P30 to P32 to three push button switches.

Functional Description:

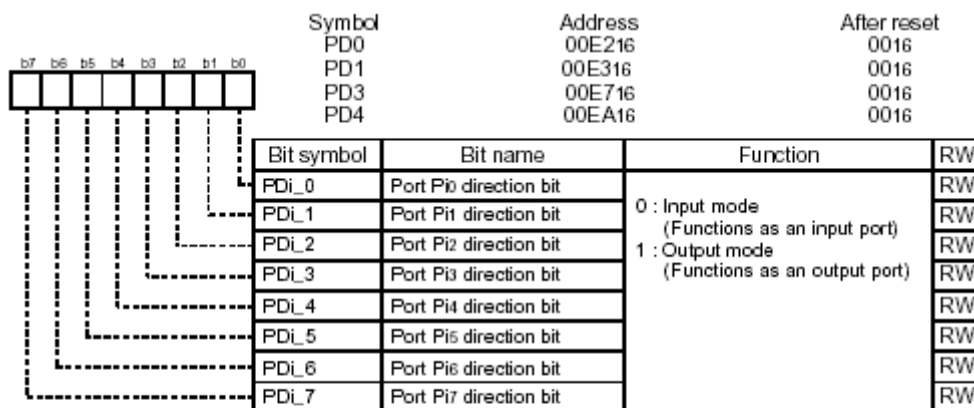
In this module, a complete project for the 4 digit stop clock is provided. 4 digits of seven segment display are used for displaying the seconds and 3 numbers of push button switches are used to give Start, Stop and Clear commands. The seven segment display is used in multiplexed mode with BCD input.

Timer X of the R8C Tiny is used to generate an interrupt at a rate of one milli second. In the timer X interrupt service routine, the LED display is refreshed and the milli seconds are counted to get one second delay for counting the seconds after the start command from the user.

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 Pi Direction Register:

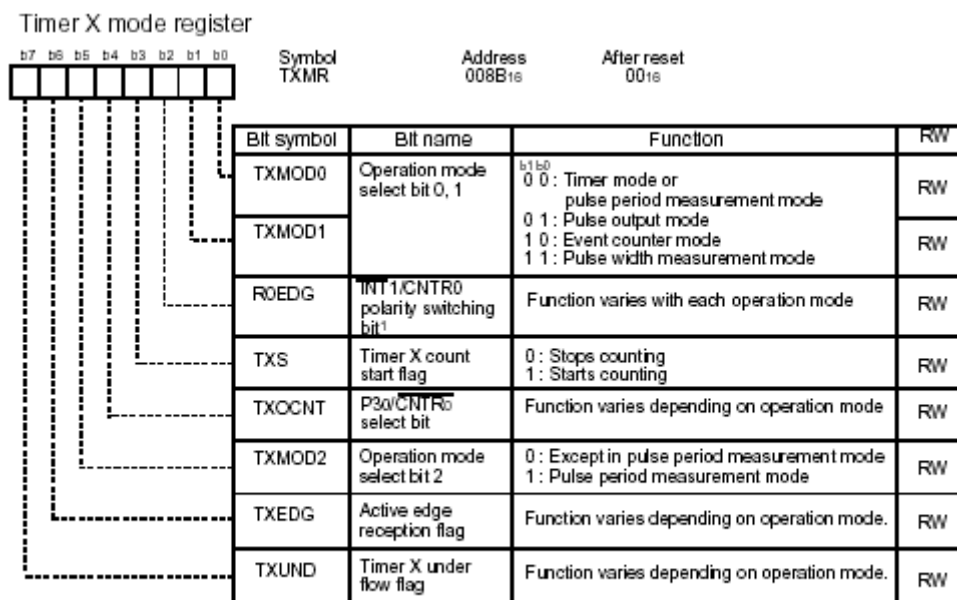


Notes:

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 the seven segment display. The port 3 lines P30 to P32 are selected as input lines after the reset.

Timer X Mode Register:



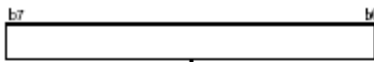
Notes:

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.

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

Prescaler X Register

Prescaler X Register



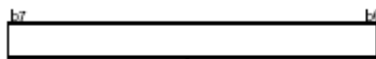
Symbol: PREX
Address: 008C16
After reset: FF16

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 selected clock meant for the timer X.

Timer X Register

Timer X Register

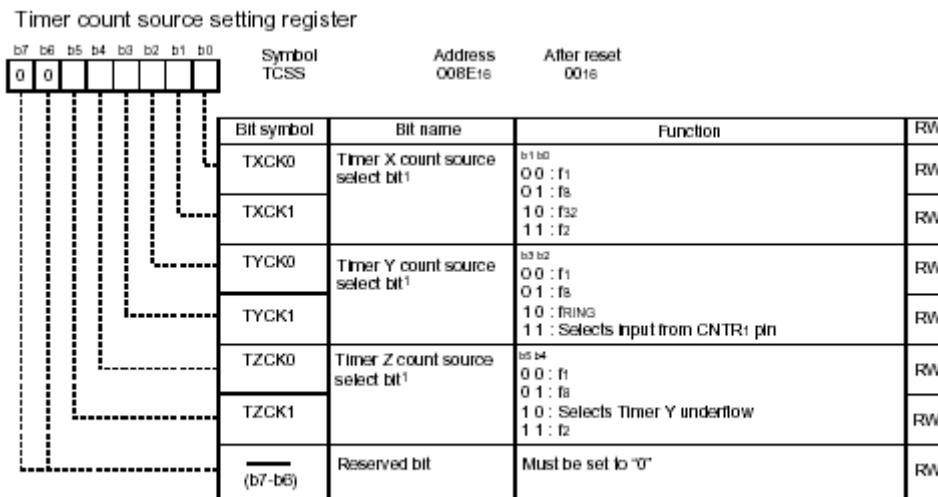


Symbol: TX
Address: 008D16
After reset: FF16

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.

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 the main loop, two types of operations are carried out according to the clock status. If the clock is in active running condition, only stop switth was enabled during switch scanning to issue a stop command. In the active stop clock condition, clear command is disabled. When the clock is stopped, the start and clear switches are enabled to get start and clear commands. After the start command, the clock running variable is set to clock running state and after the clear command, the variable holding the seconds count will be cleared to 0.

The files used in this module are listed below:

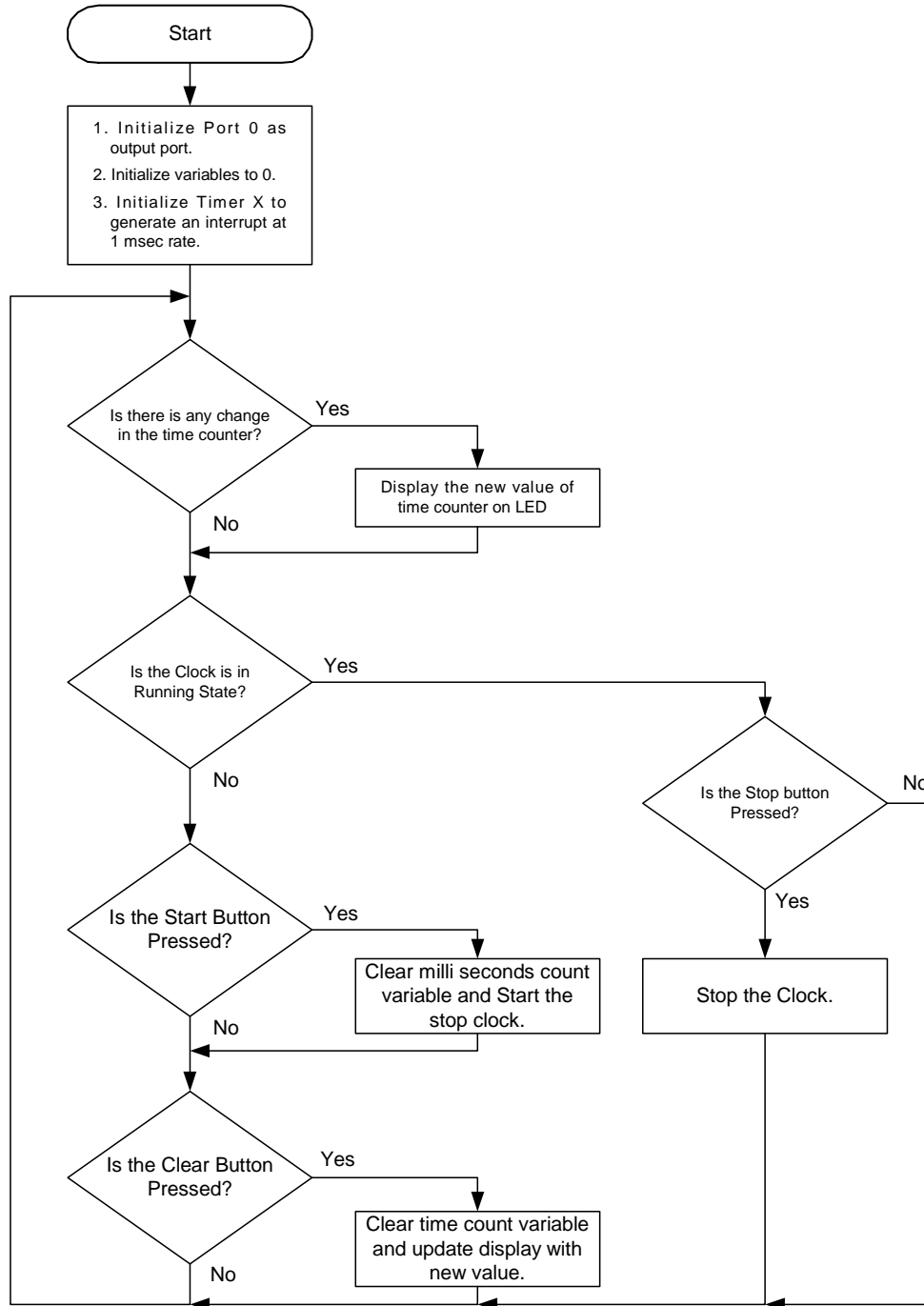
Files	Description
R8C1113_FE_A19.C	The only file for this module, has the processing function for the stop clock, timer X interrupt service routine and initialization routines for seven segment display, timer X etc.

The functions in the file **R8C1113_FE_A19.C** and short descriptions are listed below:

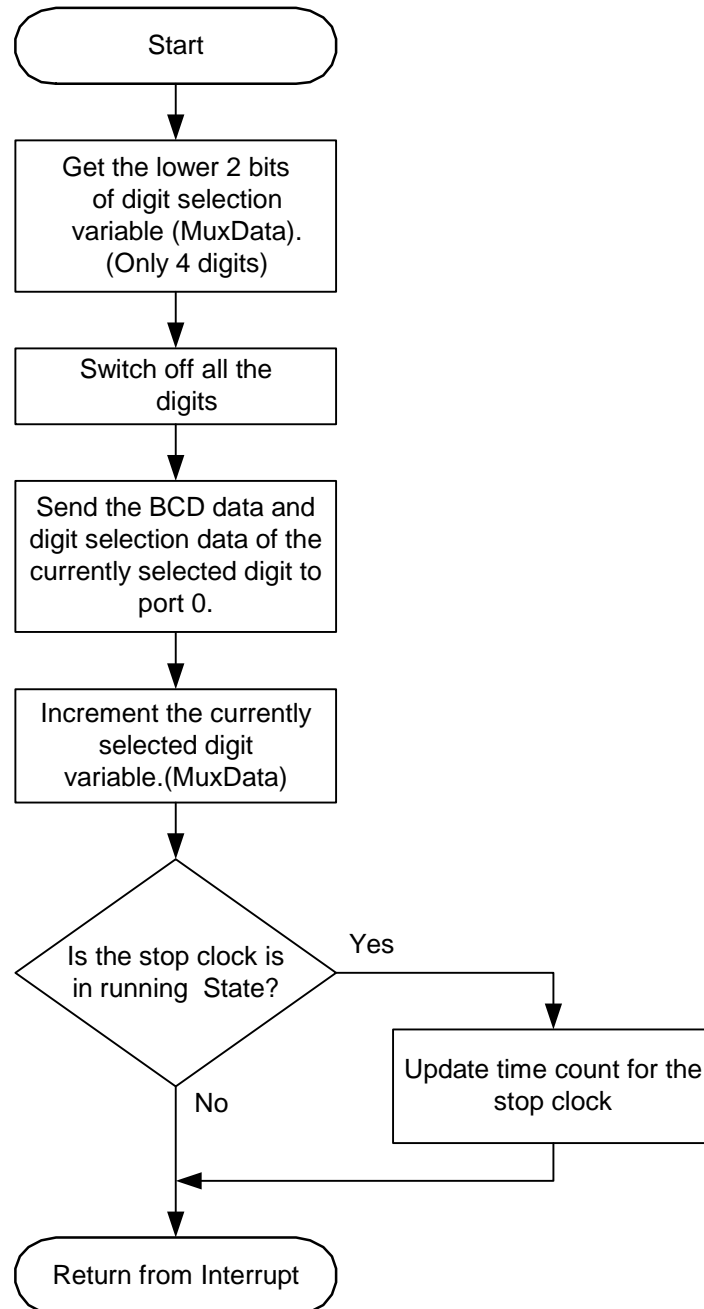
<i>Files</i>	<i>Description</i>
ProcessStopClock	Checks the Start, Stop and Clear switch status and starts the stop clock or stops the stop clock or clears the clock contents. Displays the counted seconds after start command. Inpu
ProcessTimer_X_Int	Timer X interrupt service routine. Here the seven segment display was refreshed and the milli seconds are counted to get one second delay. The seconds are counted after the start of the stop clock.
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 : Non
Display4Digits	Displays the given 4 digit number on seven segment display. Input: 4 Digit number. Output : None.
BCDIncrement	Increments the given 4 digit number by one in BCD format and returns the same to called function. Input: 4 Digit number. Output : Incremented value of given nu
InitializeSevenSegmentLED	This routine will initialize I/O lines used for seven segment display, Timer X and variables used for refreshing display. Input: None. Output :
Display4Digits	Displays the given 4 digit number on seven segment display. Input: 4 Digit number. Output : None.
BCDIncrement	Increments the given 4 digit number by one in BCD format and returns the same to called function. Input: 4 Digit number. Output : Incremented value.

Program Flow:

For Main Loop



For Timer X Interrupt Service Routine



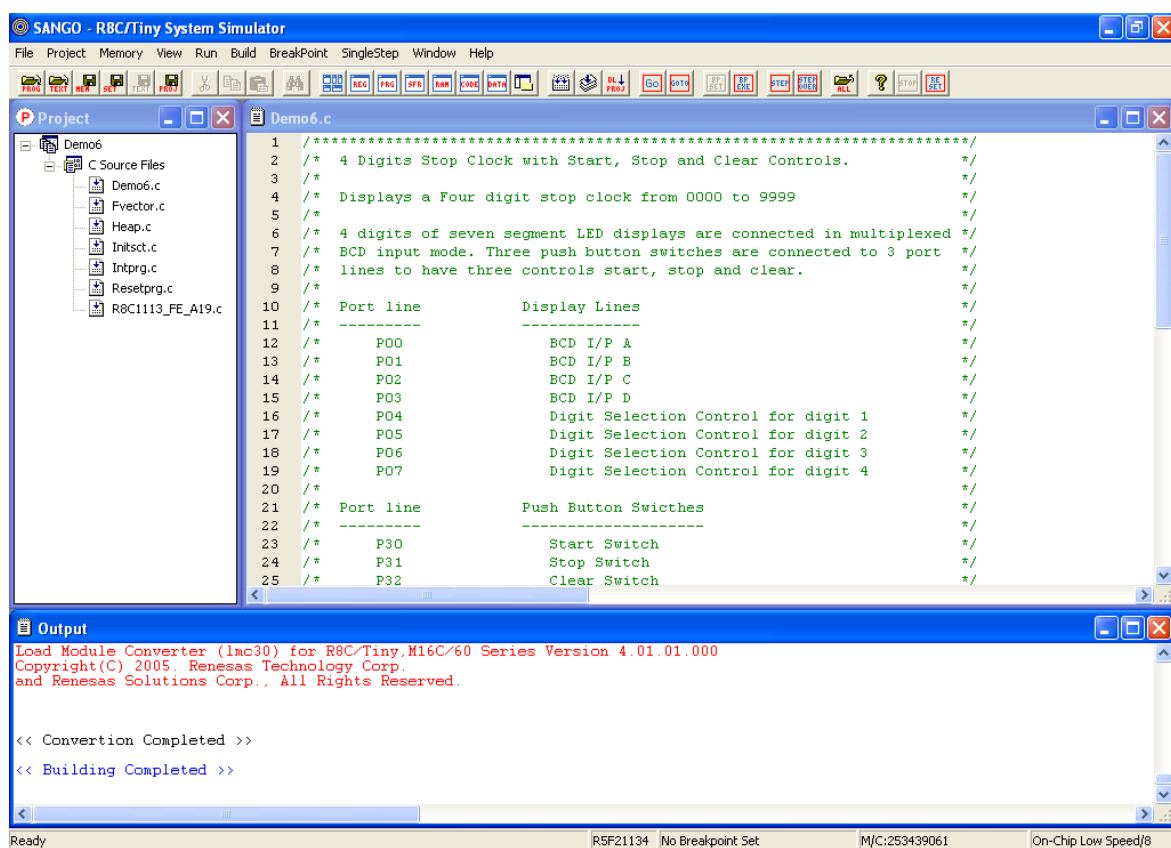
Execute Demo:

After reset, the stop clock starts showing its initial value 0000. The program will wait for start command from the user. After the start command, the clock will start counting the time and displayed in four digits from 0000 to 9999 seconds. When the stop button is pressed, the clock will be stopped and the display will show the current time count. The time count will be cleared to 0000 if the clear button is pressed when the clock is in stop condition.

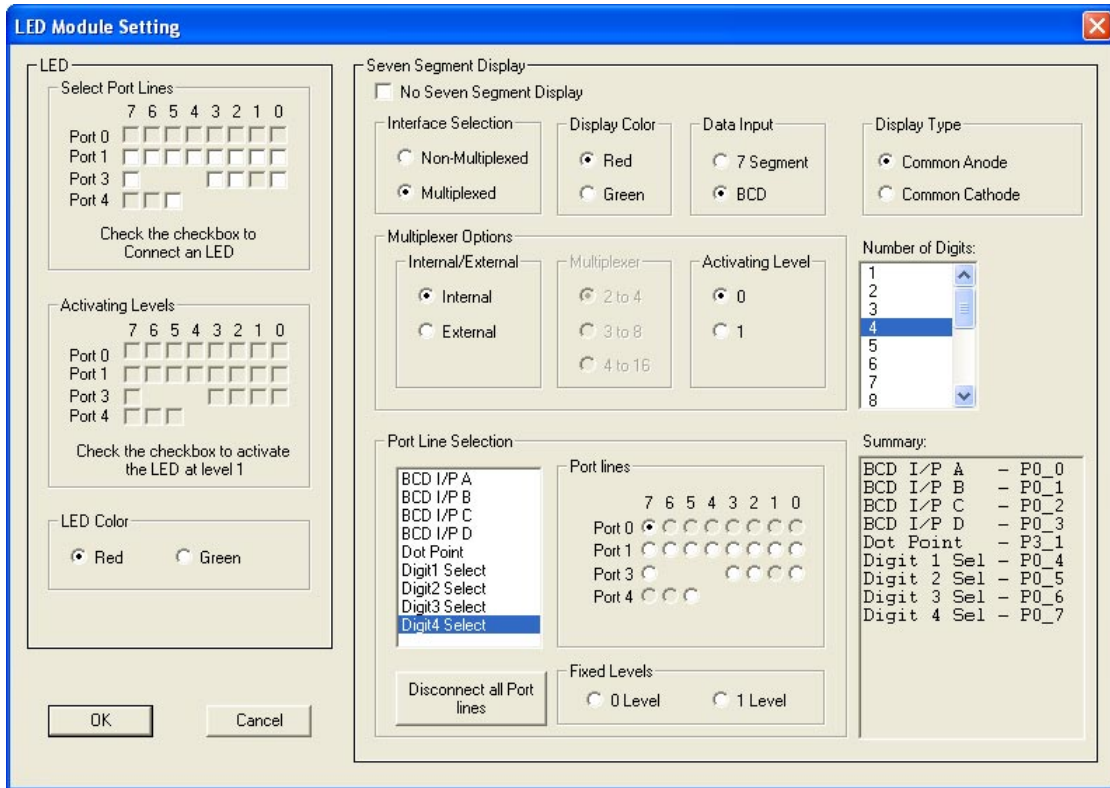
Use Topview Simulator to Verify the Design.

Open the project Demo4 in the R8C/Tiny System Simulator using **Open Project** option from **Project** menu. The project window opens up along with the Demo4.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler output.

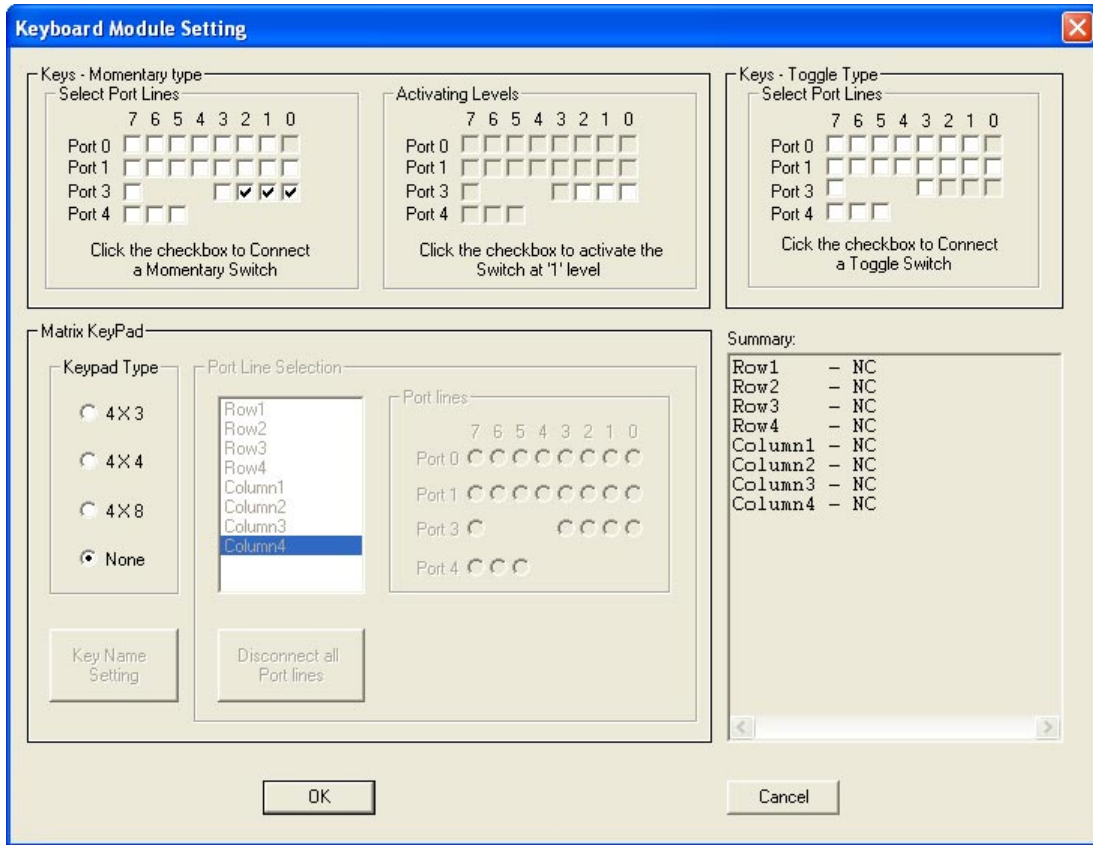
Use **Project -> Download Project** from main menu to download the Demo4.mot file into the simulator's memory for simulation.



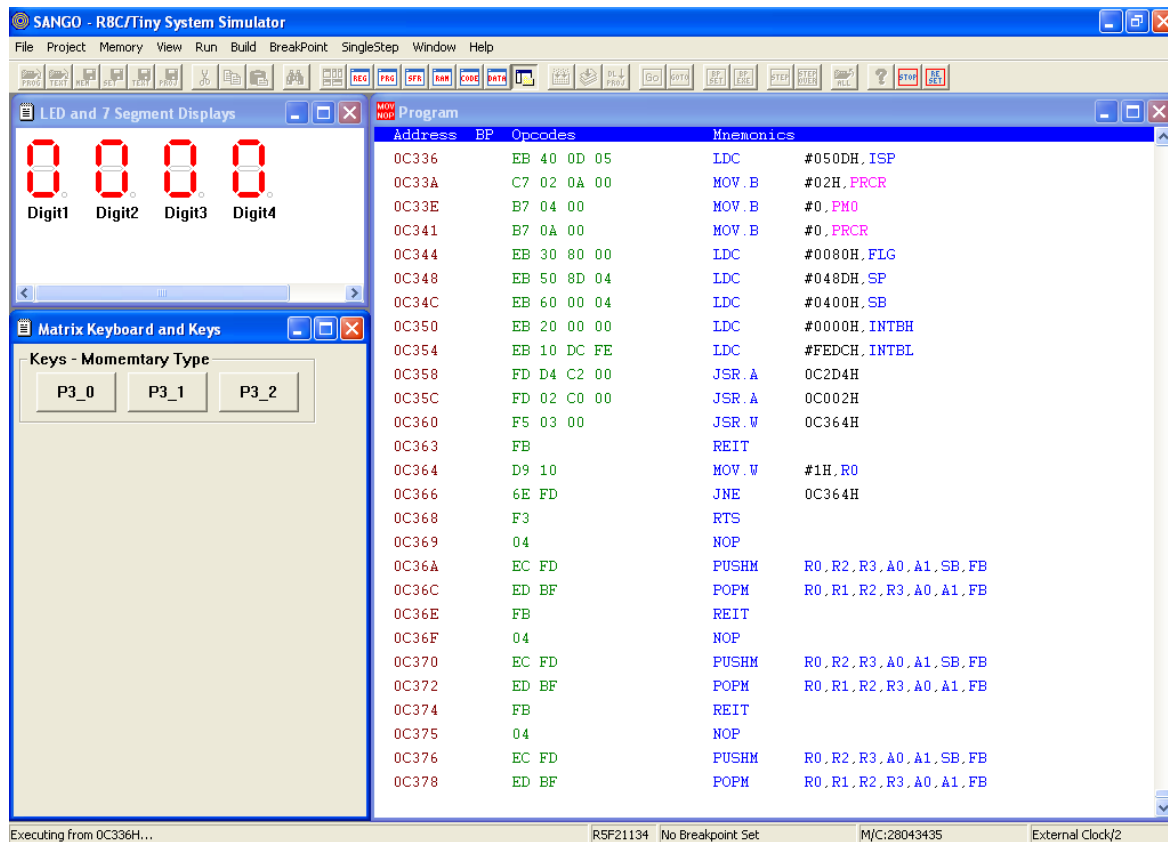
Open the LED Module settings window and do the settings to the 7-segment LED module as shown. Connect BCD input of the display to the port lines P00 to P03 and the 4 digit selection lines to P04 to P07 respectively.



Open the Keyboard Module settings window and do the settings. Connect the rows to P14 to P17 and columns to P10 to P13.



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



Run the program using **Go** from the **Run** menu.

After reset, the stop clock starts showing its initial value 0000. The program will wait for start command from the user. After the start command, the clock will start counting the time and displayed in four digits from 0000 to 9999 seconds. When the stop button is pressed, the clock will be stopped and the display will show the current time count. The time count will be cleared to 0000 if the clear button is pressed when the clock is in stop condition.