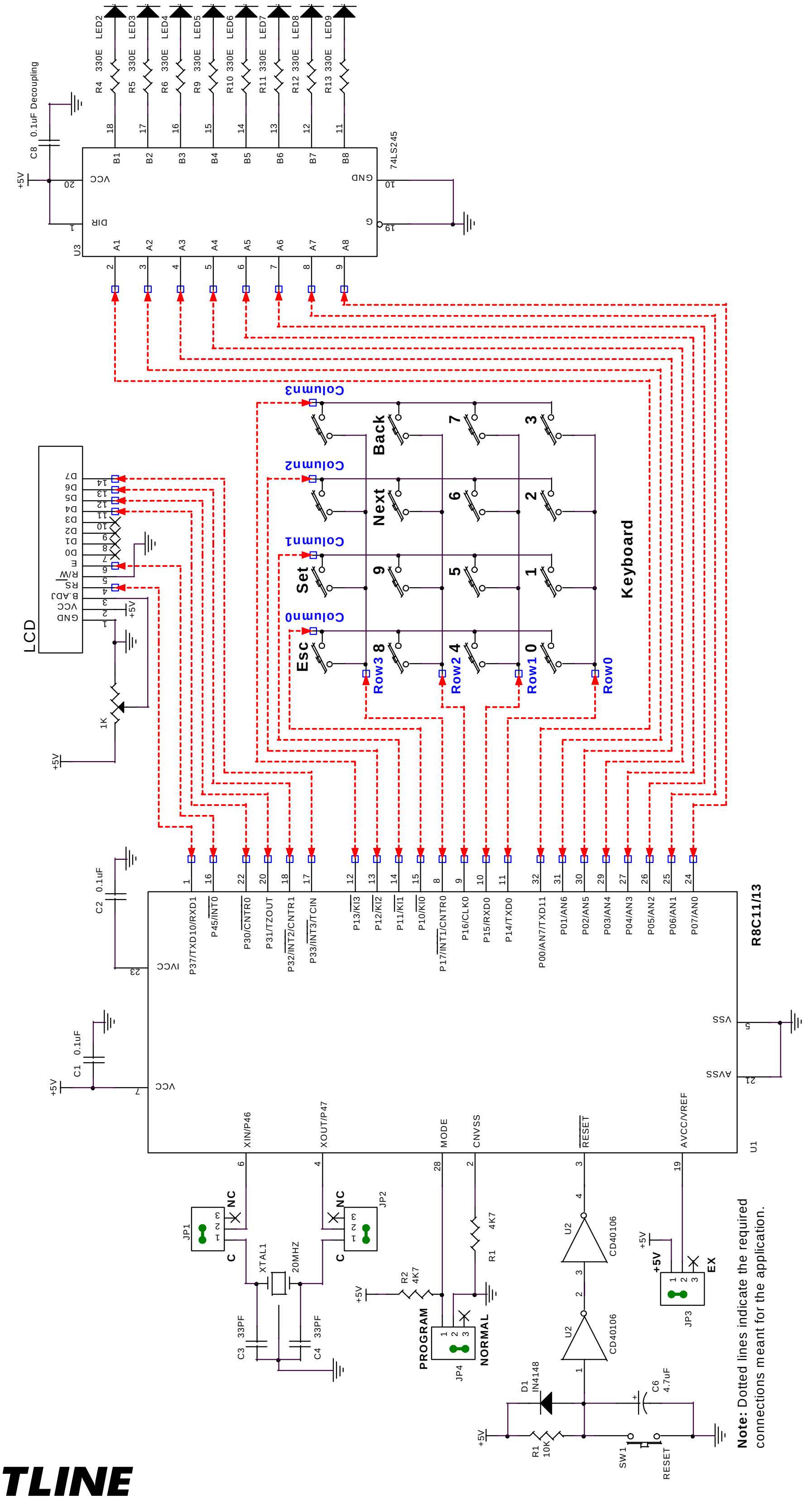
Introduction:

This demo gives details of a complete project, an 8 channel sequential controller which will switch on 8 channels sequentially one by one for a predefined time. For interaction, a LCD module of 2 line x 16 characters and a key pad of 16 keys are provided.

Hardware:

The project hardware contains a LCD 2 line x 16 characters interfaced with the microcontroller using 4 bit data bus. A small key pad with 16 keys is also included for the user interaction. 8 numbers of point LEDs are used to indicate the channel outputs.

Circuit:



Note: Dotted lines indicate the required connections meant for the application.

Connections:

1. Connect port lines P30 to P33 to D4 to D7 line of LCD.
2. Connect port lines P37 and P45 to RS and E lines of LCD.
3. Connect R/W line of LCD to GND.
4. Connect port lines P10 to P13 to Column0 to Column3 of 4X4 matrix keyboard.
5. Connect port lines P14 to P17 to Row0 to Row3 of 4X4 matrix keyboard.
6. Connect port lines P00 to P07 to eight point LEDs.

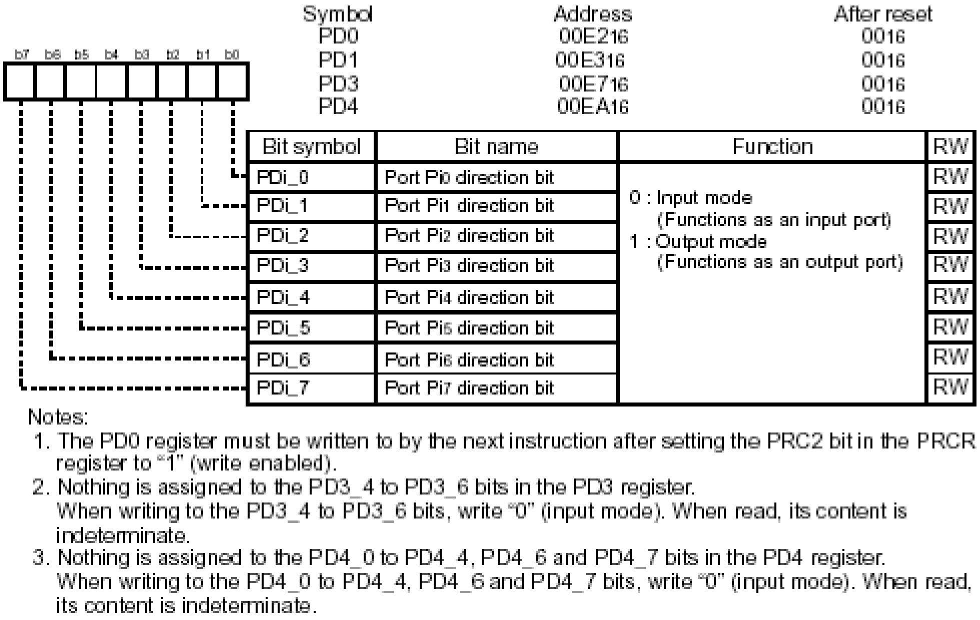
Functional Description:

In this module, a complete project for 8 channel sequential controller is studied . The sequential controller has 8 outputs and the user can set individual on time for each channel. Two line by 16 characters LCD is used for displaying the RTC data, 4 X 4 matrix keyboard is used to enter the user setting, Timer X is used generate 1 second delay and port 0 is used for controlling 8 outputs.

Registers Used:

PD0	- Port 0 Direction Register
PD1	- Port 1 Direction Register
PD3	- Port 3 Direction Register
PD4	- Port 4 Direction Register
PUR0	- Pull-up control register 0
KIEN	- Key input enable register
KUPIC	- Key input Interrupt control register
TXMR	- Timer X mode register
PREX	- Prescaler X Register
TX	- Timer X Register
TCSS	- Timer count source setting register

Port Direction Registers:



Symbol	Address	After reset
PD0	$00E2_{16}$	00_{16}
PD1	$00E3_{16}$	00_{16}
PD3	$00E7_{16}$	00_{16}
PD4	$00EA_{16}$	00_{16}

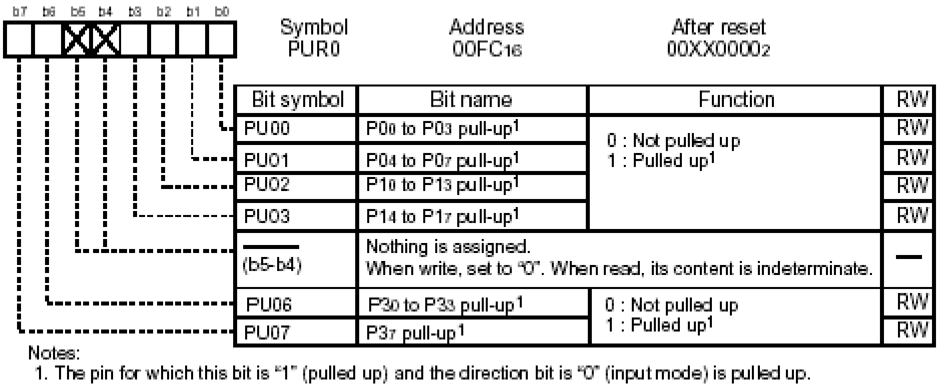
Bit symbol	Bit name	Function	RW
PDi_0	Port Pi_0 direction bit	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	RW
PDi_1	Port Pi_1 direction bit		RW
PDi_2	Port Pi_2 direction bit		RW
PDi_3	Port Pi_3 direction bit		RW
PDi_4	Port Pi_4 direction bit		RW
PDi_5	Port Pi_5 direction bit		RW
PDi_6	Port Pi_6 direction bit		RW
PDi_7	Port Pi_7 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 the port lines P00 to P07 as output lines to control the LED outputs. PD1 is loaded with H'0F to select port lines P14 to P17 as output lines to connect matrix keyboard. The port line P45 is selected as output line by setting bit PD45 in PD4 register. The port lines P30 to P33 and P37 are selected as output lines by moving H'8F to PD3 register.

PUR0 - Pull-up Control Register:



Symbol	Address	After reset
PUR0	$00FC_{16}$	$00XX0000_2$

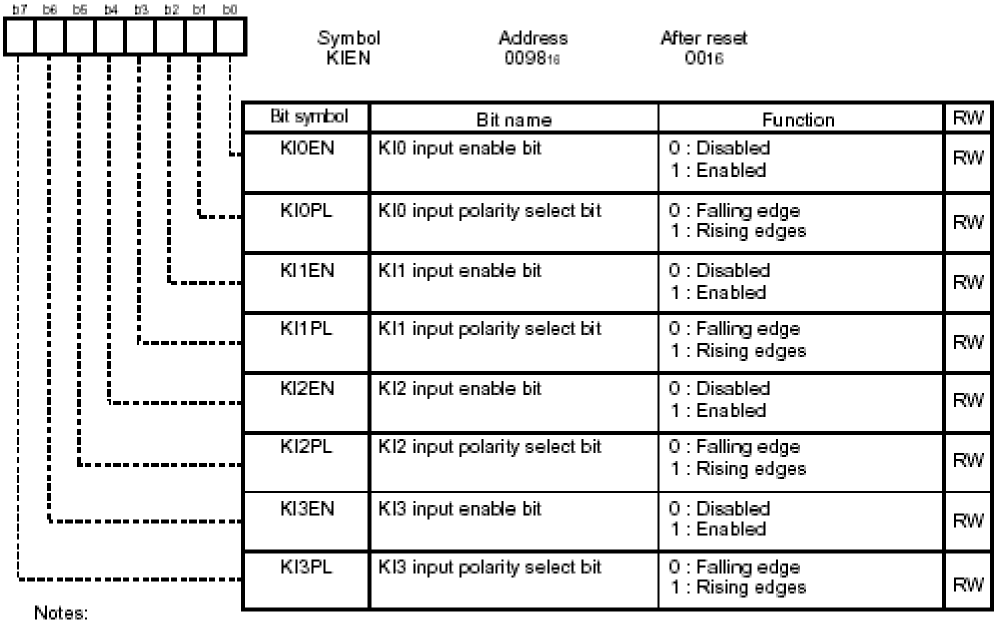
Bit symbol	Bit name	Function	RW
PU00	$P0_0$ to $P0_3$ pull-up[1]	0 : Not pulled up 1 : Pulled up[1]	RW
PU01	$P0_4$ to $P0_7$ pull-up[1]		RW
PU02	$P1_0$ to $P1_3$ pull-up[1]		RW
PU03	$P1_4$ to $P1_7$ pull-up[1]		RW
— (b5-b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
PU06	$P3_0$ to $P3_3$ pull-up[1]	0 : Not pulled up 1 : Pulled up[1]	RW
PU07	$P3_7$ pull-up[1]		RW

Notes:

1. The pin for which this bit is "1" (pulled up) and the direction bit is "0" (input mode) is pulled up.

The bit PU02 is set to 1 to enable internal pull ups for port lines P10 to P13.

KEIN - Key Input Enable Register:



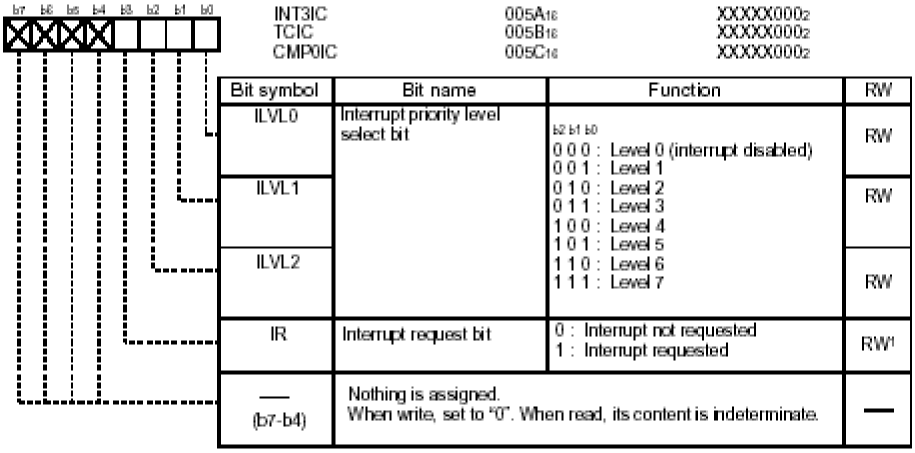
Symbol	Address	After reset
KIEN	0098_{16}	00_{16}

Bit symbol	Bit name	Function	RW
KI0EN	KI0 input enable bit	0 : Disabled 1 : Enabled	RW
KI0PL	KI0 input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI1EN	KI1 input enable bit	0 : Disabled 1 : Enabled	RW
KI1PL	KI1 input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI2EN	KI2 input enable bit	0 : Disabled 1 : Enabled	RW
KI2PL	KI2 input polarity select bit	0 : Falling edge 1 : Rising edges	RW
KI3EN	KI3 input enable bit	0 : Disabled 1 : Enabled	RW
KI3PL	KI3 input polarity select bit	0 : Falling edge 1 : Rising edges	RW

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

The register KIEN is loaded with H'55 to enable KI0 to KI3 key input interrupts and to select falling edge detection.

KUPIC - Key Input Interrupt Control Register:

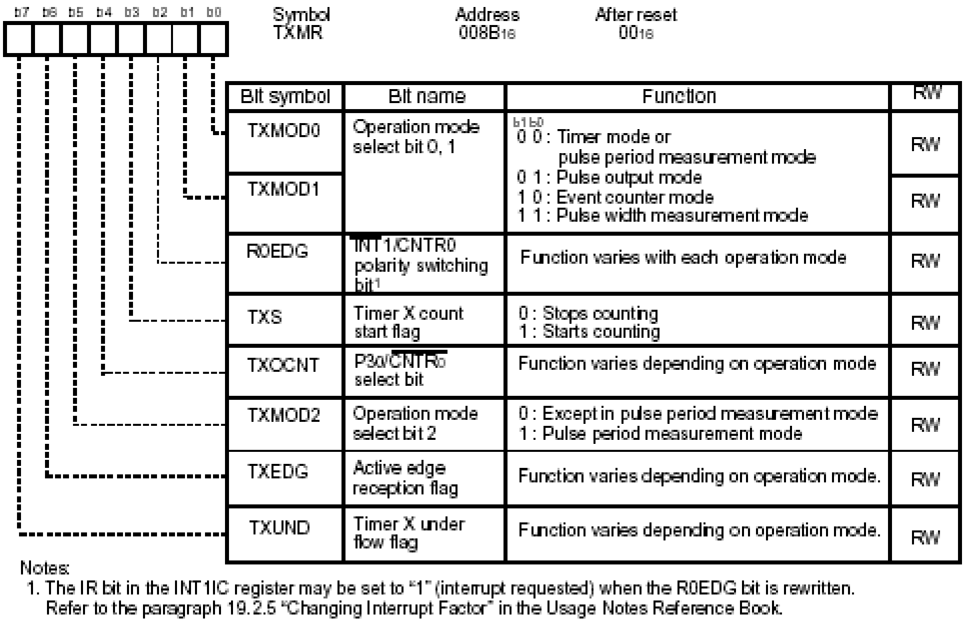


INT3IC	$005A_{16}$	$XXXXX000_2$
TCIC	$005B_{16}$	$XXXXX000_2$
CMP0IC	$005C_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW[1]
— (b7-b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

The priority level 2 is selected for key input interrupt by moving H'02 to KUPIC register.

TXMR - Timer X Mode Register:


b7 b6 b5 b4 b3 b2 b1 b0

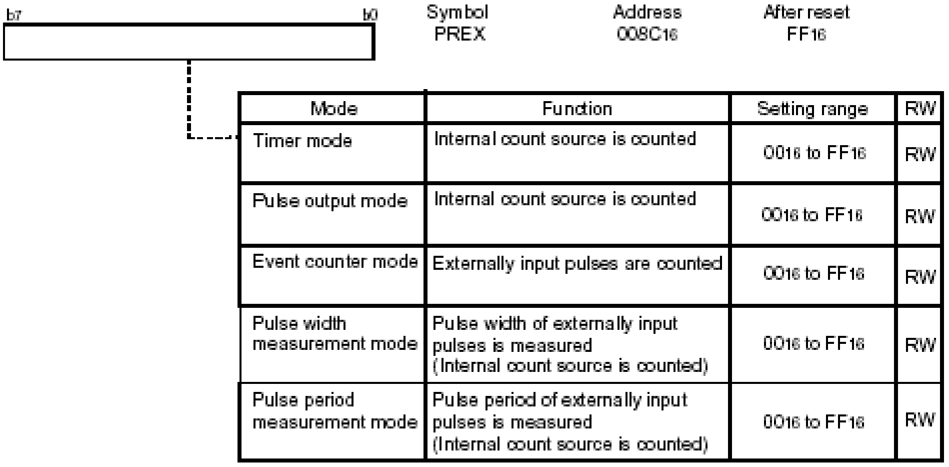
Symbol	Address	After reset
TXMR	$008B_{16}$	00_{16}

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	$\overline{\text{INT1}}$/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	$P3_0/\overline{\text{CNTR}_0}$ 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 register TXMR is loaded with H'00 to set timer mode.

PREX - Prescaler X Register:

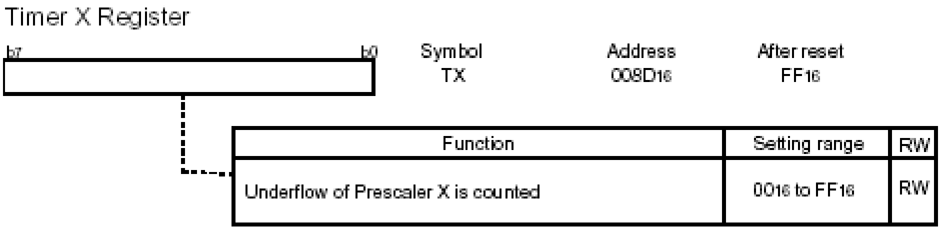


Symbol	Address	After reset
PREX	$008C_{16}$	FF_{16}

Mode	Function	Setting range	RW
Timer mode	Internal count source is counted	00_{16} to FF_{16}	RW
Pulse output mode	Internal count source is counted	00_{16} to FF_{16}	RW
Event counter mode	Externally input pulses are counted	00_{16} to FF_{16}	RW
Pulse width measurement mode	Pulse width of externally input pulses is measured (Internal count source is counted)	00_{16} to FF_{16}	RW
Pulse period measurement mode	Pulse period of externally input pulses is measured (Internal count source is counted)	00_{16} to FF_{16}	RW

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

TX - Timer X Register:



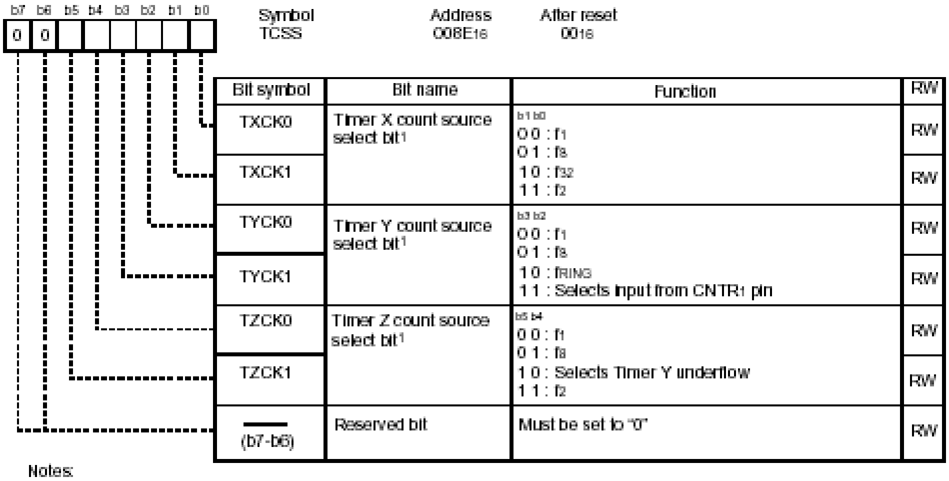
Timer X Register

Symbol	Address	After reset
TX	$008D_{16}$	FF_{16}

Function	Setting range	RW
Underflow of Prescaler X is counted	00_{16} to FF_{16}	RW

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

TCSS - Timer Count Source Setting Register:



Symbol	Address	After reset
TCSS	$008E_{16}$	00_{16}

Bit symbol	Bit name	Function	RW
TXCK0	Timer X count source select bit[1]	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_2	RW
TXCK1			RW
TYCK0	Timer Y count source select bit[1]	b3 b2 0 0 : f_1 0 1 : f_8 1 0 : f_{RING} 1 1 : Selects input from $CNTR_1$ pin	RW
TYCK1			RW
TZCK0	Timer Z count source select bit[1]	b5 b4 0 0 : f_1 0 1 : f_8 1 0 : Selects Timer Y underflow 1 1 : f_2	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:

After reset, the following initialization are taking place before entering into the main loop.

1. Initialize the port 0 as output to control outputs.
2. Initialize the port lines P14 to P17 as output lines and enable KI0 to KI3 key input interrupts.
3. Initialize port lines P30 to P33, P37 and P45 as output lines and initialize the LCD in 4 bit mode.
4. A message “**`Sequential Contr`**” is displayed on the 1st line of LCD.
5. Initialize Timer X to generate an interrupt at a rate of 1 KHZ.

Key input interrupt is generated for any key press in the keyboard. In the Key Input (KI0 to KI3) interrupt service routine, the pressed key value is identified.

In the main loop, the keyboard status is checked and the control goes to setting function if the “SET” key was pressed. Otherwise the program will decrement the currently selected channel’s on time for every second and activates the output line meant for that channel till the on time becomes zero. The currently selected channel number gets incremented and the loop is repeated until the last channel. After this, the control starts repeating the whole thing again.

The files used in this module are listed below:

Files	Description
Demo7.C	Main file for this module, will sequentially switches on 8 channels one after another for the programmed on time and checks the keyboard for user interaction.
R8C1113_FE_LCD_4Bit.C	LCD routines to initialize LCD, cursor on/off, display a message etc.
R8C1113_FE_LCD_4Bit.H	Declarations of functions in R8C1113_FE_LCD_4Bit.C
R8C1113_FE_Keyboard.C	Keybaord routines to initialize Key input interrupts, I/O lines used by keyboard and key input interrupt service routine.
R8C1113_FE_Keyboard.H	Declarations of functions in R8C1113_FE_Keyboard.C

The functions in the file “Demo7.C” and short descriptions are listed below:

Functions	*Description*
main	This is main function of sequential controller and will switch 8 channels one after another on with the set time. Checks the keyboard for "SET" key press. **Input**: None. **Output** : None.
InitializeSequentialController	Initialize I/O lines, Keyboard and LCD. **Input**: None. **Output** : None.
ProcessTimer_X_Int	Timer X interrupt service routine and will count the milli seconds and set a flag after 1000 milli seconds. **Input**: None. **Output** : None.
InitializeTimerX	Initializes Timer X to generate an interrupt for every one milli second. **Input**: None. **Output** : None.
DisplayChannelNumberOntime	Displays Channel number and on time of the currently selected channel. **Input** : None. **Output** : None.
ConvertBCD	Convert two digit hex number into BCD number. **Input**: None. **Output** : None.
DisplayLCD2DigitBCD	Displays the given 2 digit hex number in BCD format on LCD at given location. **Input**: Line number, character position and data. **Output** : None.
BCDDecrement	Decrements the given 2 digit number by one in BCD format. **Input**: 2 digit BCD data. **Output** : Decremented 2 digit BCD data.

The functions in the file “R8C1113_FE_LCD_4Bit.C” and short descriptions are listed below:

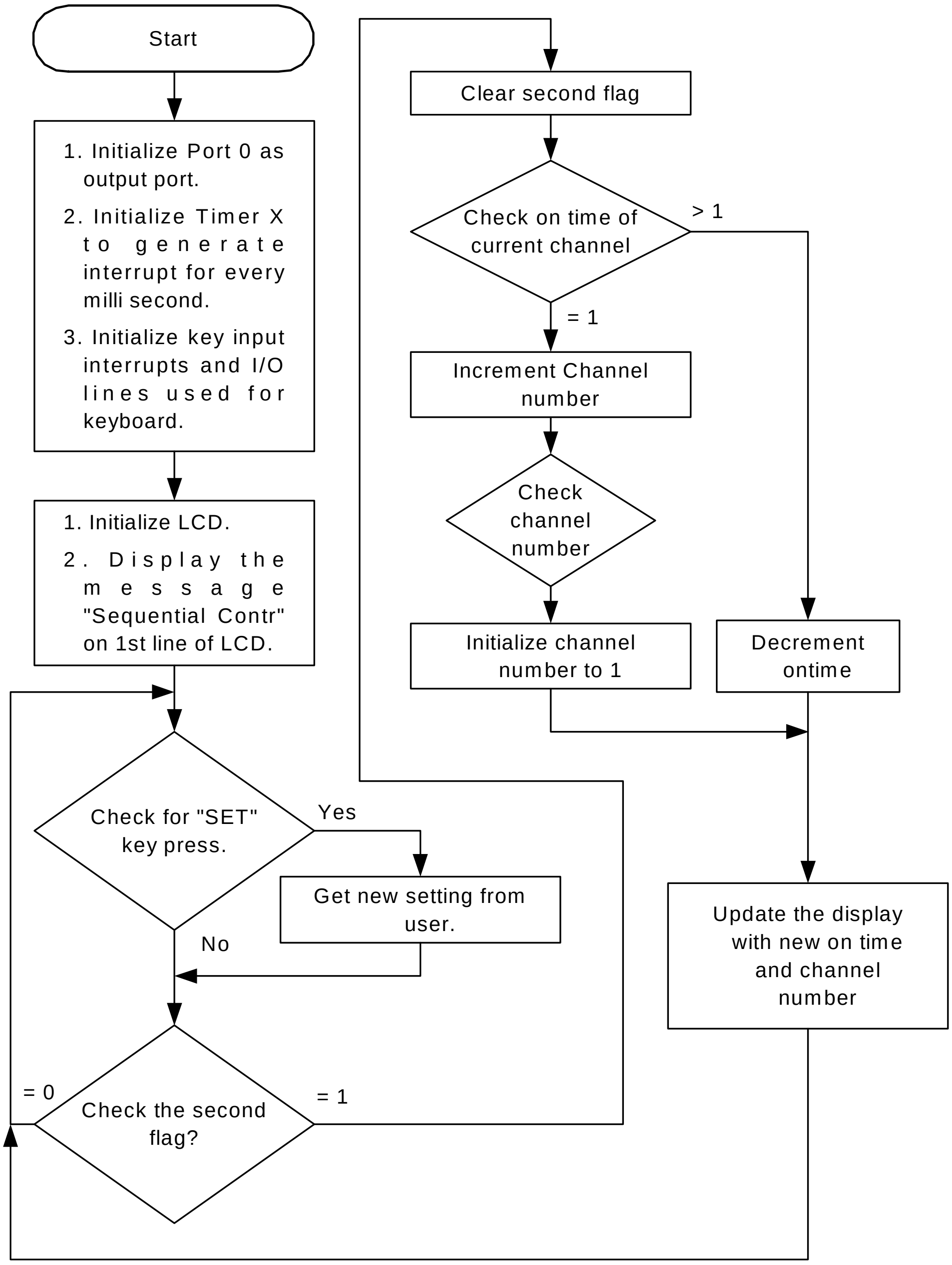
Functions	***Description***
DisplayLCD	Displays a message (16 Characters) on LCD on the given line number. **Input**: Line number and message string. **Output** : None.
DisplayLCD2Digit	Displays the given 2 digit number on LCD at given location. **Input**: Line number, character position and data. **Output** : None.
CursorON	Makes the cursor visible on LCD. **Input**: None. **Output** : None.
CursorOFF	Hides the cursor. **Input**: None. **Output** : None.
InitializeLCD	Initializes the I/O lines used by LCD and LCD in 4 bit mode. Clears the LCD. **Input**: None. **Output** : None.
WriteDataLCD	Write a data byte to LCD. **Input**: Data Byte. **Output** : None.
WriteCommandLCD	Write a command byte to LCD. **Input**: Command Byte. **Output** : None.

The functions in the file “R8C1113_FE_Keyboard.c” and short descriptions are listed below:

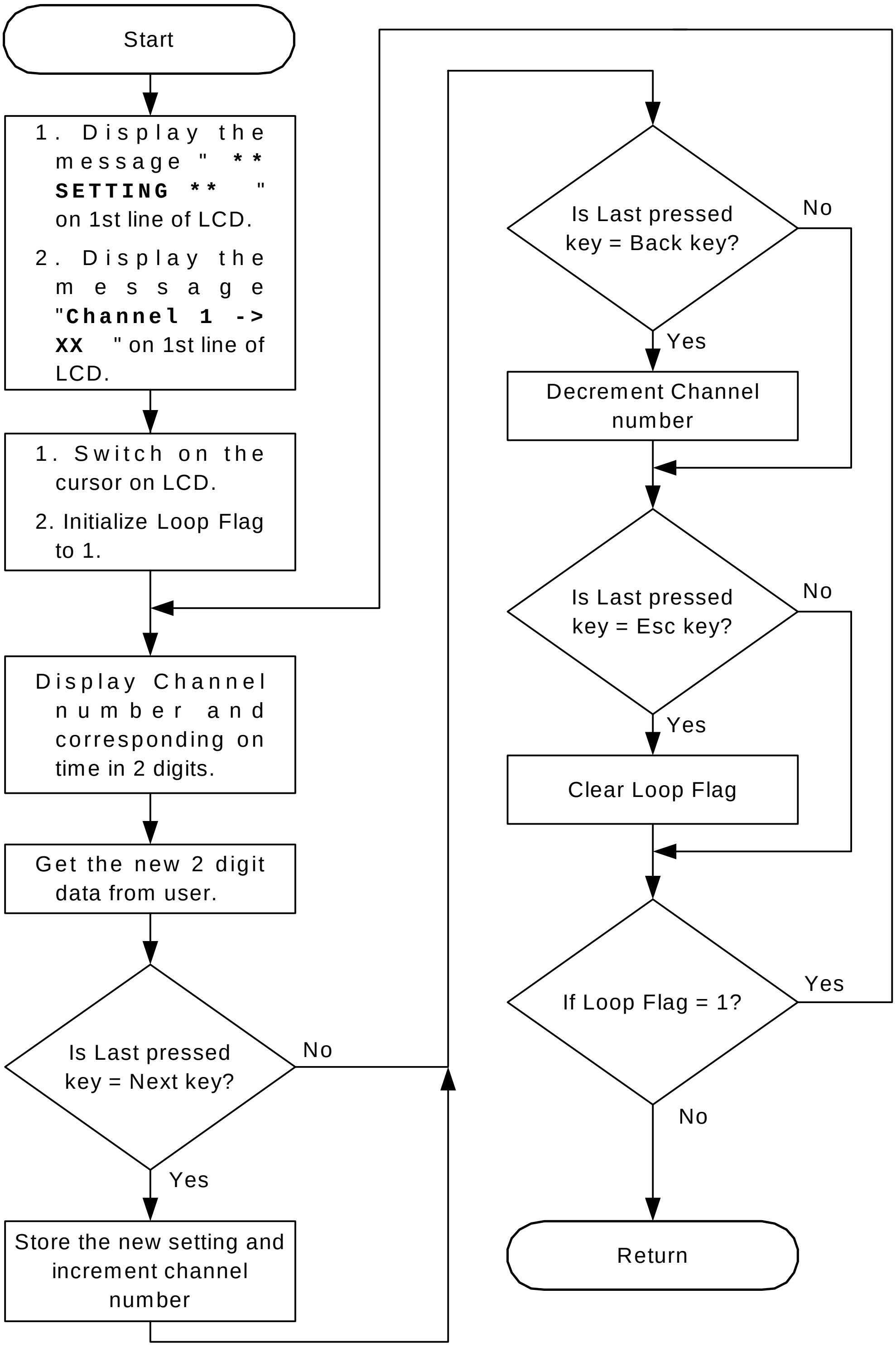
Functions	***Description***
ReadKeyboardStatus	Returns the keyboard **Input**: None. **Output** : Keyboard Status.
WaitForKeyPress	Waits for a key press and returns the key code of the pressed key. **Input**: None. **Output** : Key Code.
ReadKeyCode	Reads and returns the last pressed key code without waiting for a key press. **Input**: None. **Output** : Key Code.
InitializeKeyboard	Initializes the I/O lines used by keyboard and enables Key input (KI0 to KI3) interrupts. **Input**: None. **Output** : None.
ProcessKey_Int	Interrupt service routine for key input interrupt. **Input**: None. **Output** : None.

Program Flow:

For Main Loop


Start
1. Initialize Port 0 as output port.
2. Initialize Timer X to generate interrupt for every milli second.
3. Initialize key input interrupts and I/O lines used for keyboard.
1. Initialize LCD.
2. Display the message "Sequential Contr" on 1st line of LCD.
Check for "SET" key press.
Yes
Get new setting from user.
No
Check the second flag?
= 0
= 1
Clear second flag
Check on time of current channel
> 1
= 1
Increment Channel number
Check channel number
Initialize channel number to 1
Decrement ontime
Update the display with new on time and channel number

### For Setting Routine


Start
1. Display the message " ** SETTING ** " on 1st line of LCD.
2. Display the message "Channel 1 -> XX " on 1st line of LCD.
1. Switch on the cursor on LCD.
2. Initialize Loop Flag to 1.
Display Channel number and corresponding on time in 2 digits.
Get the new 2 digit data from user.
Is Last pressed key = Next key?
No
Yes
Store the new setting and increment channel number
Is Last pressed key = Back key?
No
Yes
Decrement Channel number
Is Last pressed key = Esc key?
No
Yes
Clear Loop Flag
If Loop Flag = 1?
Yes
No
Return

Execute Demo:

After reset, the program will display a message “**Sequential Contr**” on the first line of LCD and the channel number and on time in second line as shown below:

```
“Sequential Contr
Channel  x: yy   “
```

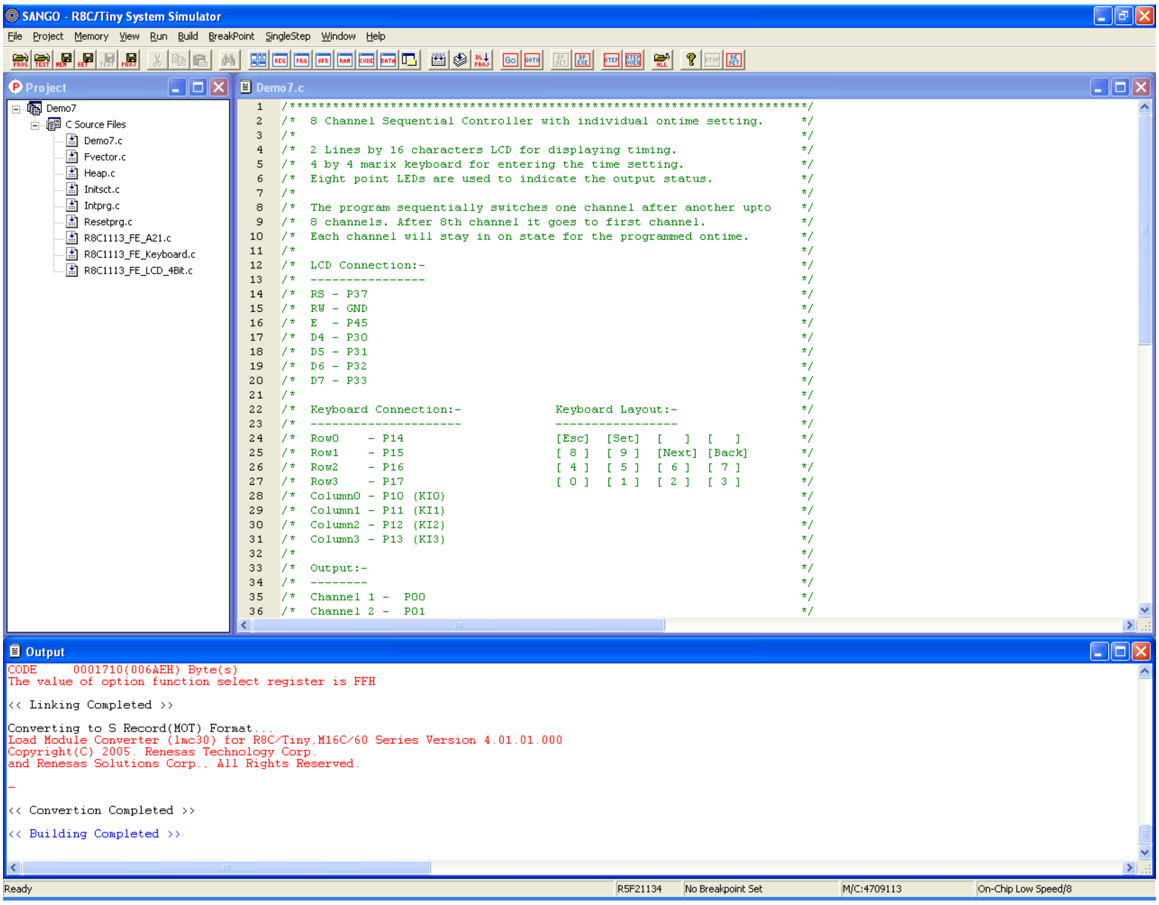
Where x is the current channel number and yy is the on time of the current channel.

The program switches the outputs on one after other for the set time.

Use Topview Simulator to Verify the Design.

Open the project Demo7 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo7.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

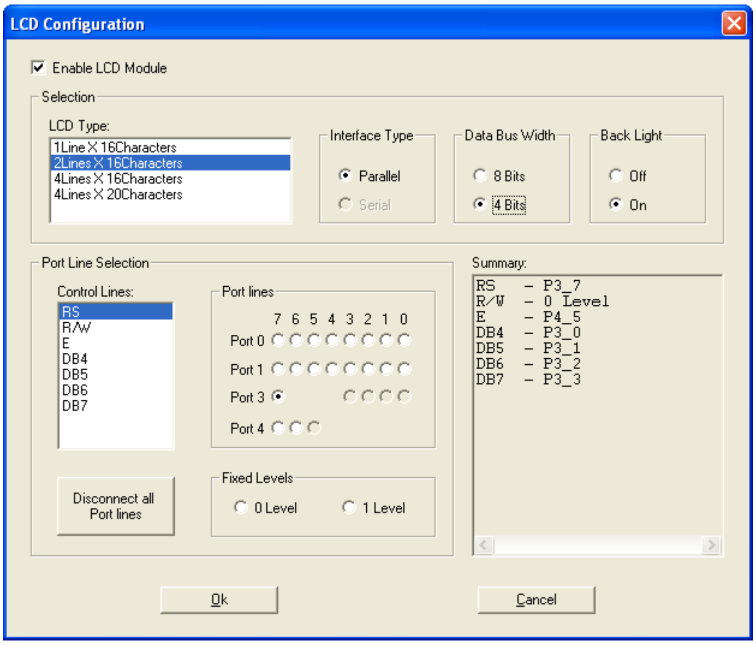
Use **Project** -> **Download Project** from main menu to download the .mot file into the simulator's memory for simulation.



```
/************************************************************************/
/*  8 Channel Sequential Controller with individual ontime setting.     */
/*                                                                      */
/*  2 Lines by 16 characters LCD for displaying timing.                 */
/*  4 by 4 marix keyboard for entering the time setting.                */
/*  Eight point LEDs are used to indicate the output status.            */
/*                                                                      */
/*  The program sequentially switches one channel after another upto    */
/*  8 channels. After 8th channel it goes to first channel.             */
/*  Each channel will stay in on state for the programmed ontime.       */
/*                                                                      */
/*  LCD Connection:-                                                    */
/*  ----------------                                                    */
/*  RS - P37                                                            */
/*  RW - GND                                                            */
/*  E  - P45                                                            */
/*  D4 - P30                                                            */
/*  D5 - P31                                                            */
/*  D6 - P32                                                            */
/*  D7 - P33                                                            */
/*                                                                      */
/*  Keyboard Connection:-               Keyboard Layout:-               */
/*  ----------------------              -----------------               */
/*  Row0    - P14                       [Esc]  [Set]  [   ]  [   ]      */
/*  Row1    - P15                       [ 8 ]  [ 9 ]  [Next] [Back]     */
/*  Row2    - P16                       [ 4 ]  [ 5 ]  [ 6 ]  [ 7 ]      */
/*  Row3    - P17                       [ 0 ]  [ 1 ]  [ 2 ]  [ 3 ]      */
/*  Column0 - P10 (KI0)                                                 */
/*  Column1 - P11 (KI1)                                                 */
/*  Column2 - P12 (KI2)                                                 */
/*  Column3 - P13 (KI3)                                                 */
/*                                                                      */
/*  Output:-                                                            */
/*  --------                                                            */
/*  Channel 1 -  P00                                                    */
/*  Channel 2 -  P01                                                    */
```

```
CODE     0001710(006AEH) Byte(s)
The value of option function select register is FFH

<< Linking Completed >>

Converting to S Record(MOT) Format...
Load Module Converter (lmc30) for R8C/Tiny,M16C/60 Series Version 4.01.01.000
Copyright(C) 2005. Renesas Technology Corp.
and Renesas Solutions Corp., All Rights Reserved.

-

<< Convertion Completed >>

<< Building Completed >>
```

Do the settings to the LCD modules as shown. Connect LCD control and data lines to port lines P37, P33 to P30 and P45.



LCD Configuration

☑ Enable LCD Module

Selection

LCD Type:
1Line X 16Characters
2Lines X 16Characters (selected)
4Lines X 16Characters
4Lines X 20Characters

Interface Type: ● Parallel ○ Serial

Data Bus Width: ○ 8 Bits ● 4 Bits

Back Light: ○ Off ● On

Port Line Selection

Control Lines: RS (selected), R/W, E, DB4, DB5, DB6, DB7

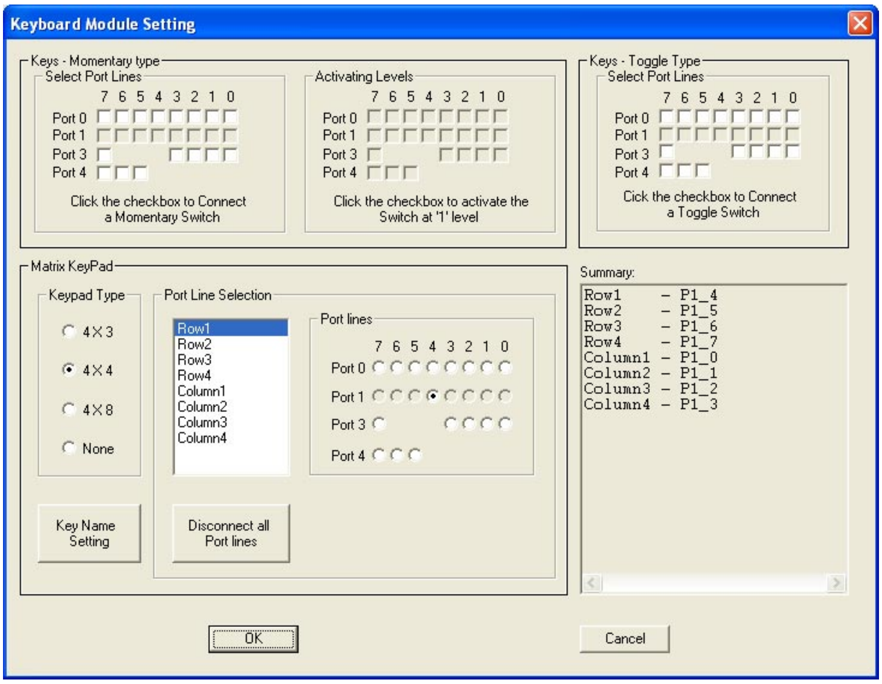
Port lines: 7 6 5 4 3 2 1 0 — Port 0, Port 1, Port 3 (● 7), Port 4

Fixed Levels: ○ 0 Level ○ 1 Level

Summary:

```
RS   - P3_7
R/W  - 0 Level
E    - P4_5
DB4  - P3_0
DB5  - P3_1
DB6  - P3_2
DB7  - P3_3
```

Make the setting for keyboard module as shown below:



Keyboard Module Setting

Keys - Momentary type — Select Port Lines: 7 6 5 4 3 2 1 0 — Port 0, Port 1, Port 3, Port 4 (none checked)

Click the checkbox to Connect a Momentary Switch

Activating Levels: 7 6 5 4 3 2 1 0 — Port 0, Port 1, Port 3, Port 4 (none checked)

Click the checkbox to activate the Switch at '1' level

Keys - Toggle Type — Select Port Lines: 7 6 5 4 3 2 1 0 — Port 0, Port 1, Port 3, Port 4 (none checked)

Cick the checkbox to Connect a Toggle Switch

Matrix KeyPad

Keypad Type: ○ 4X3 ● 4X4 ○ 4X8 ○ None

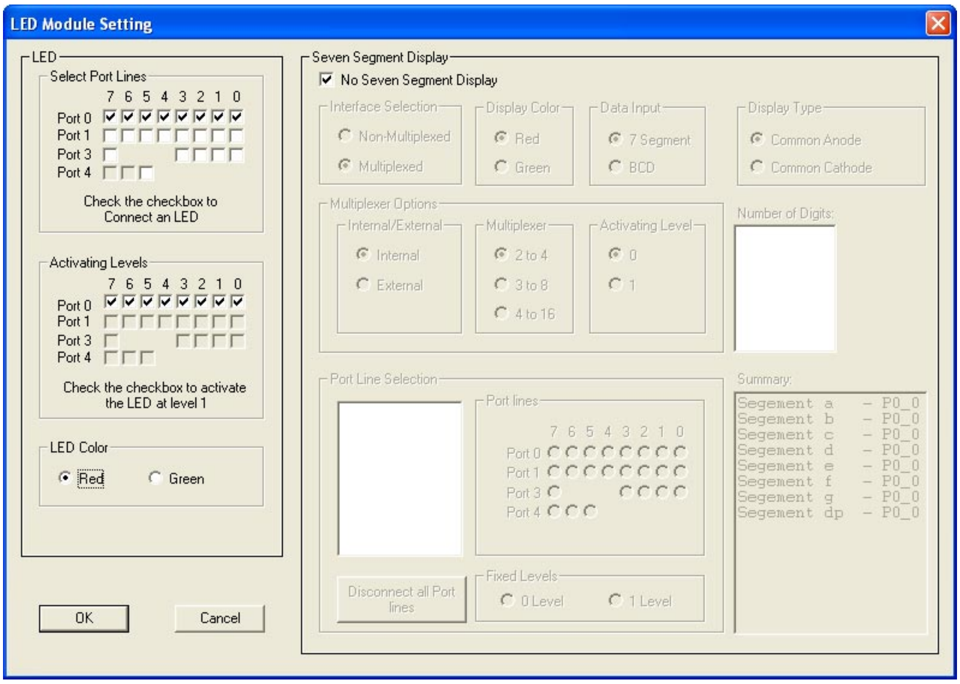
Port Line Selection: Row1 (selected), Row2, Row3, Row4, Column1, Column2, Column3, Column4

Port lines: 7 6 5 4 3 2 1 0 — Port 0, Port 1 (● 4), Port 3, Port 4

Summary:

```
Row1    - P1_4
Row2    - P1_5
Row3    - P1_6
Row4    - P1_7
Column1 - P1_0
Column2 - P1_1
Column3 - P1_2
Column4 - P1_3
```

Connect 8 numbers of point LEDs to the port lines P00 to P07 using LED module setting.



LED Module Setting

LED

Select Port Lines

	7	6	5	4	3	2	1	0
Port 0	☑	☑	☑	☑	☑	☑	☑	☑
Port 1	□	□	□	□	□	□	□	□
Port 3	□				□	□	□	□
Port 4	□	□	□					

Check the checkbox to Connect an LED

Activating Levels

	7	6	5	4	3	2	1	0
Port 0	☑	☑	☑	☑	☑	☑	☑	☑
Port 1	□	□	□	□	□	□	□	□
Port 3	□				□	□	□	□
Port 4	□	□	□					

Check the checkbox to activate the LED at level 1

LED Color

● Red ○ Green

Seven Segment Display

☑ No Seven Segment Display

Interface Selection: ○ Non-Multiplexed ● Multiplexed

Display Color: ● Red ○ Green

Data Input: ● 7 Segment ○ BCD

Display Type: ● Common Anode ○ Common Cathode

Multiplexer Options

Internal/External: ● Internal ○ External

Multiplexer: ● 2 to 4 ○ 3 to 8 ○ 4 to 16

Activating Level: ● 0 ○ 1

Number of Digits:

Port Line Selection

Port lines

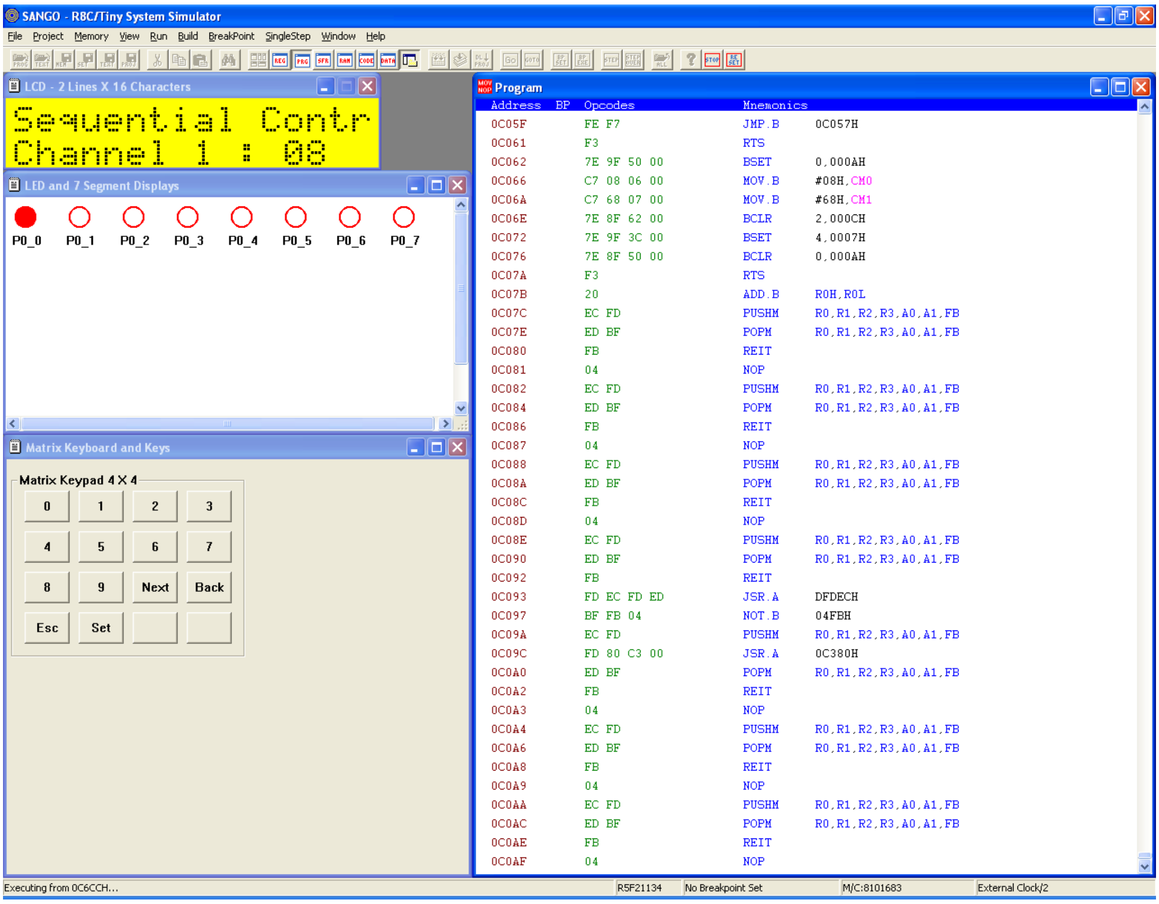
	7	6	5	4	3	2	1	0
Port 0	○	○	○	○	○	○	○	○
Port 1	○	○	○	○	○	○	○	○
Port 3	○				○	○	○	○
Port 4	○	○	○					

Fixed Levels: ○ 0 Level ○ 1 Level

Summary:

```
Segement a   - P0_0
Segement b   - P0_0
Segement c   - P0_0
Segement d   - P0_0
Segement e   - P0_0
Segement f   - P0_0
Segement g   - P0_0
Segement dp  - P0_0
```

Then open the LED, LCD, Keyboard windows and arranage them as shown below.


SANGO - R8C/Tiny System Simulator
File Project Memory View Run Build BreakPoint SingleStep Window Help
LCD - 2 Lines X 16 Characters
LED and 7 Segment Displays
P0_0 P0_1 P0_2 P0_3 P0_4 P0_5 P0_6 P0_7
Matrix Keyboard and Keys
Matrix Keypad 4 X 4
0 1 2 3
4 5 6 7
8 9 Next Back
Esc Set
Program

```
Sequential Contr
Channel  1  : 08
```

```
Address  BP  Opcodes       Mnemonics
0C05F        FE F7         JMP.B    0C057H
0C061        F3            RTS
0C062        7E 9F 50 00   BSET     0,000AH
0C066        C7 08 06 00   MOV.B    #08H,CM0
0C06A        C7 68 07 00   MOV.B    #68H,CM1
0C06E        7E 8F 62 00   BCLR     2,000CH
0C072        7E 9F 3C 00   BSET     4,0007H
0C076        7E 8F 50 00   BCLR     0,000AH
0C07A        F3            RTS
0C07B        20            ADD.B    R0H,R0L
0C07C        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C07E        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C080        FB            REIT
0C081        04            NOP
0C082        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C084        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C086        FB            REIT
0C087        04            NOP
0C088        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C08A        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C08C        FB            REIT
0C08D        04            NOP
0C08E        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C090        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C092        FB            REIT
0C093        FD EC FD ED   JSR.A    DFDECH
0C097        BF FB 04      NOT.B    04FBH
0C09A        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C09C        FD 80 C3 00   JSR.A    0C380H
0C0A0        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C0A2        FB            REIT
0C0A3        04            NOP
0C0A4        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C0A6        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C0A8        FB            REIT
0C0A9        04            NOP
0C0AA        EC FD         PUSHM    R0,R1,R2,R3,A0,A1,FB
0C0AC        ED BF         POPM     R0,R1,R2,R3,A0,A1,FB
0C0AE        FB            REIT
0C0AF        04            NOP
```

```
Executing from 0C6CCH...    R5F21134    No Breakpoint Set    M/C:8101683    External Clock/2
```

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

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

The program sequentially switches one channel after another upto 8 channels. After 8th channel it goes to first channel. Each channel will stay in on state for the programmed ontime. The channel number and ontime are displayed in LCD as shown below:

```
"Sequential Contr
Channel  1: 07    "
```