

### Introduction:

This demo describes a complete project meant for industrial timing applications. This is a complete one with all the required features to manage various time-based operations in the industrial environments. This module is the Programmable Timer with the facility to activate its output based on the Real Time Clock. The timer has the facility to activate its output for eight different timings.

To implement this project, a LCD module of 2 line x 16 characters, or key matrix of 16 keys are used for the user interaction. A real time clock is also included to get timing information for the project . A buzzer is connected with the micon to indicate the output condition. In real, this buzzer can be replaced by a relay or any other suitable output element.

### Hardware:

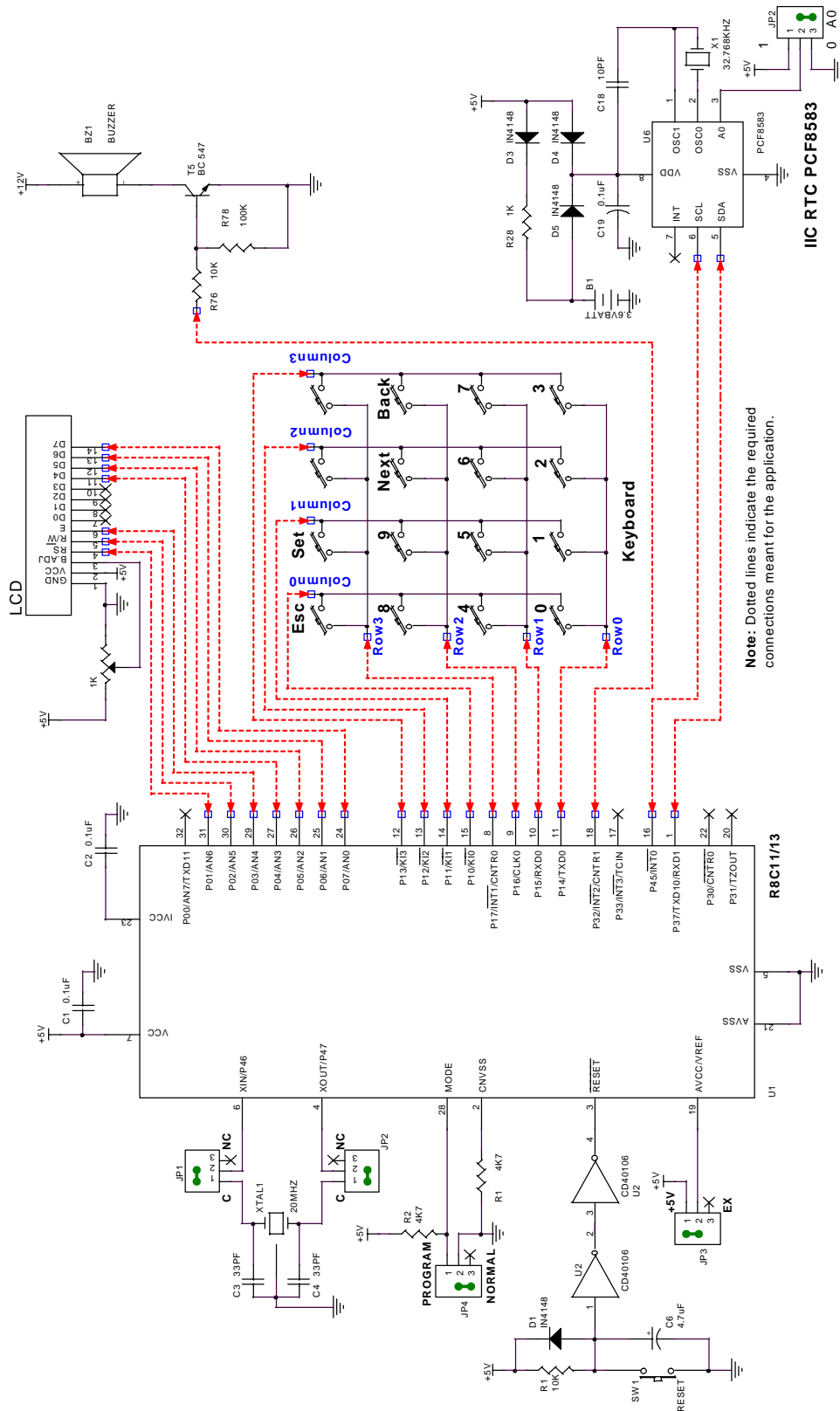
This project, Programmable Timer, is based on the Real Time Clock device, PCF8583 along with user interaction elements; 2 line x 16 character LCD module, a keypad with 16 keys. A buzzer is also included to indicate the output status.

The LCD module is interfaced with the micon using 4 bit data bus to save the port lines. Micon's port lines are simulated for the IIC bus to interface the IIC bus based real time clock.

During the operation, the micon keeps comparing the current time with the set time. When the time matches, the controller activates the output for the predefined time.

A buzzer is connected with the micon to indicate the output status.

Circuit:



### Connections:

1. Connect port lines P04 to P07 to D4 to D7 line of LCD.
2. Connect port lines P01, P02 and P03 to RS, R/W and E lines of LCD.
3. Connect port lines P10 to P13 to Column0 to Column3 of 4X4 matrix keyboard.
4. Connect port lines P14 to P17 to Row0 to Row3 of 4X4 matrix keyboard.
5. Connect P45 to SCL and P37 to SDA lines of IIC bus for RTC.
6. Connect Port line P32 to either a point LED to get visual output or to buzzer to get sound output.

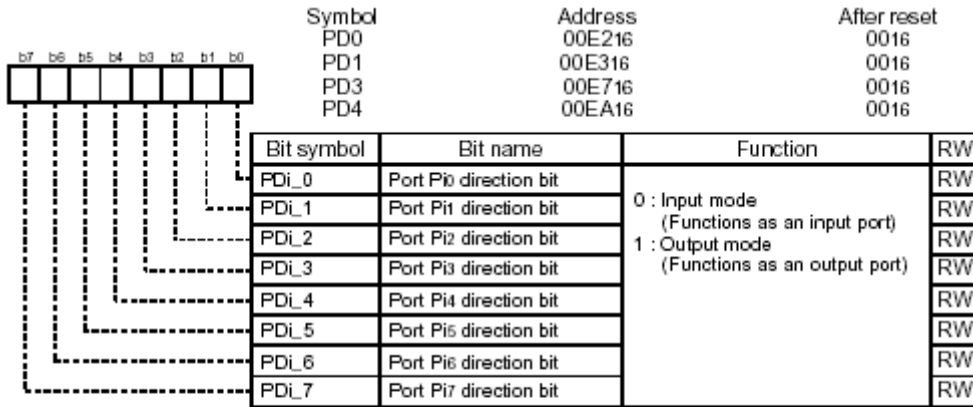
### Functional Description:

In this module, a complete project for the programmable timer is provided. The programmable timer has one output indication and user can set upto 8 timings. 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, IIC RTC is used to maintain the calendar and one LED/Buzzer is used for the output.

### 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

**Port Direction Registers:**

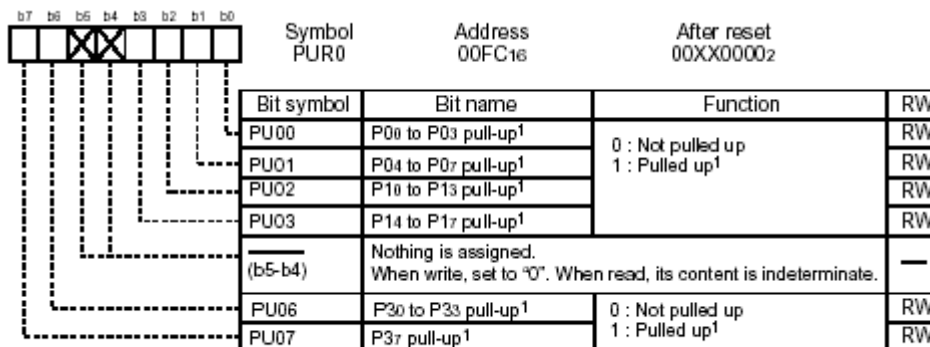


**Notes:**

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'FE to select the port lines P00 to P07 as output lines to control the LCD. 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 line P32 is selected as output line by setting bit PD32 in PD3 register.

**PUR0 - Pull-up Control Register:**

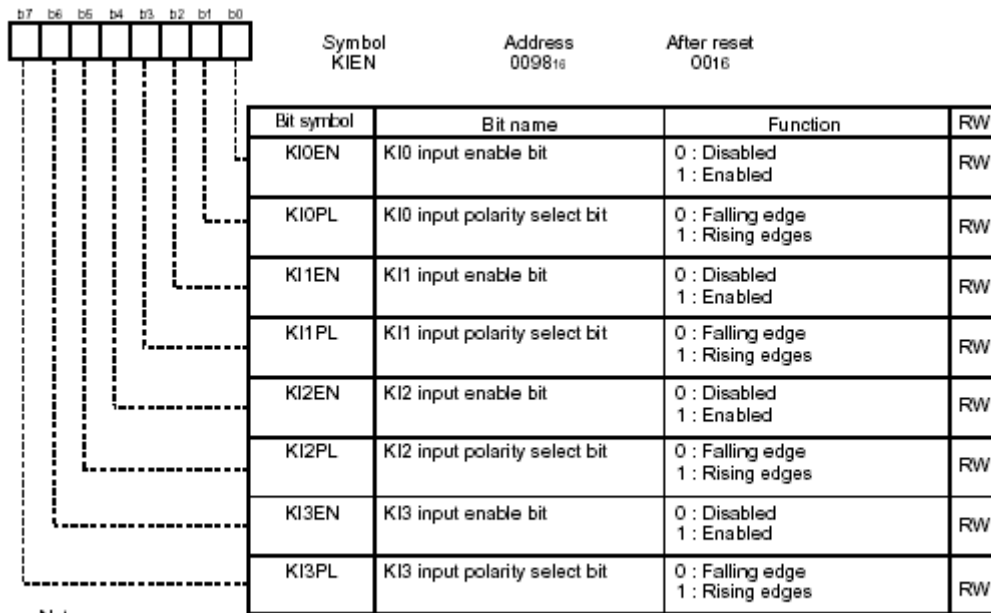


**Notes:**

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.

**KIEN - Key Input Enable Register:**

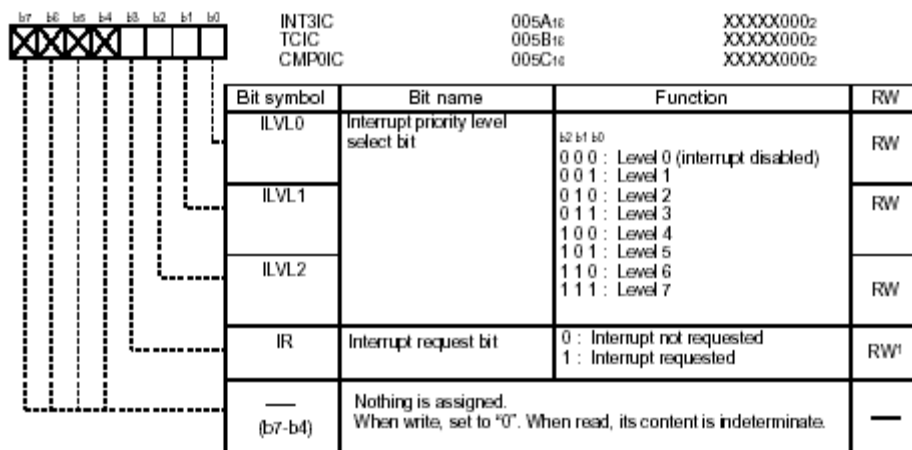


Notes:

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

**KUPIC - Key Input Interrupt Control Register:**



### Software Description:

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

1. Initialize the port line P32 as output line.
2. Read the user setting stored in RTC RAM area and place the same in the local variables.
3. Initialize the port lines P14 to P17 as output lines and enable KI0 to KI3 interrupts.
4. Initialize Port lines P01 to P07 as output line and initialize the LCD in 4 bit mode.

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

In the main loop, the keyboard status is checked and the control goes to setting function if the "SET" key is pressed. Otherwise the calender data are read from RTC and they are displayed in the first line of LCD and the last read timing is compared with user set timing, then a buzzer will be switched on if they are equal.

The files used in this module are listed below:

<i>Files</i>	<i>Description</i>
Demo8.C	Main file for this module, will read and display the RTC data from RTC and compares the user set timings with the current time and then activates the output if they are equal and checks the keyboard for user setting.
R8C1113_FE_SimIICRoutines.C	Basic routines to access the IIC bus such as Giving start bit, Stop bit, reading a byte serially, sending a byte serially etc.
R8C1113_FE_SimIICRoutines.H	Declarations of functions in R8C1113_FE_SimIICRoutines.c
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 "Demo8.C" and short descriptions are listed below:

<i>Functions</i>	<i>Description</i>
main	<p>Reads RTC data and display it on LCD.                      Compares the current time with the user set timing and activates one output if they are equal.                      Checks the keyboard for "SET" key press.  <b>Input:</b> None.  <b>Output :</b> None.</p>
InitializeProgrammbleTimer	<p>Initialize I/O lines, Keyboard, LCD and read user time setting in RTC  <b>Input:</b> None.  <b>Output :</b> None.</p>
ReadSetting	<p>Reads the user time setting stored in RTC and place it in local variables.  <b>Input:</b> None.  <b>Output :</b> None.</p>
ReadRTCData	<p>Reads the current time from RTC device.  <b>Input:</b> None.  <b>Output :</b> None.</p>
DisplayRTCData	<p>Displays RTC data read from RTC on LCD.  <b>Input :</b> None.  <b>Output :</b> None.</p>
CompareTiming	<p>Compares current time read from RTC with the user set data and activates the output if they are equal.  <b>Input:</b> None.  <b>Output :</b> None.</p>
SwitchOnOutput	<p>Switches on the output LED connected at port line P32 for a predefined time and displays on status of output at 2nd line of LCD.  <b>Input:</b> None.  <b>Output :</b> None.</p>
ConvertBCD	<p>Converts a digit Hex number into BCD.  <b>Input:</b> 2 digit Hex data.  <b>Output :</b> 2 digit BCD data.</p>



<i>Functions</i>	<i>Description</i>
DisplayLCD2DigitBCD	Displays the 2 digit Hex variable in BCD format <b>Input:</b> 2 digit Hex data. <b>Output :</b> None.
GetSetting	Gets the user time setting through keyboard. <b>Input:</b> None. <b>Output :</b> None.
StoreSetting	Stores the user setting to RTC RAM area. <b>Input:</b> None. <b>Output :</b> None.
DisplayLCD4Digit	Displays a 4 digit data (Hour:Minute) on LCD. <b>Input:</b> Line number, Character Position and data to be displayed in 4 digits. <b>Output :</b> None.
Get4DigitData	Gets 4 digit data through keyboard from user. <b>Input:</b> None. <b>Output :</b> Last pressed key code.
ReadByteRTC	Reads a byte from RTC. <b>Input:</b> Address of the location. <b>Output :</b> Read data from RTC.
WriteByteRTC	Writes a byte to RTC. <b>Input:</b> Address and data. <b>Output :</b> None.

The functions in the file "R8C1113\_FE\_SimIICRoutines.C" and short descriptions are listed below:

<i>Functions</i>	<i>Description</i>
PollAck	Sends the <b>SlaveAck</b> command and wait until slave acknowledges. This routine can be used to sense the completion of a write cycle. <b>Input:</b> None. <b>Output :</b> None.
SendOneByteIIC	Sends a byte (command or address) to the slave IIC device serially. <b>Input:</b> Data to be send. <b>Output :</b> None.
StartBit	Generates a start condition for IIC bus. <b>Input:</b> None. <b>Output :</b> None.
StopBit	Generates a stop condition for IIC bus. <b>Input:</b> None. <b>Output :</b> None.
SlaveAck	Gets acknowledge from the slave. <b>Input:</b> None. <b>Output :</b> None.
ReadOneByteIIC	Rends a byte from the slave IIC device serially. <b>Input:</b> Data to be send. <b>Output :</b> None.

The functions in the file "R8C1113\_FE\_LCD\_4Bit.C" and short descriptions are listed below:

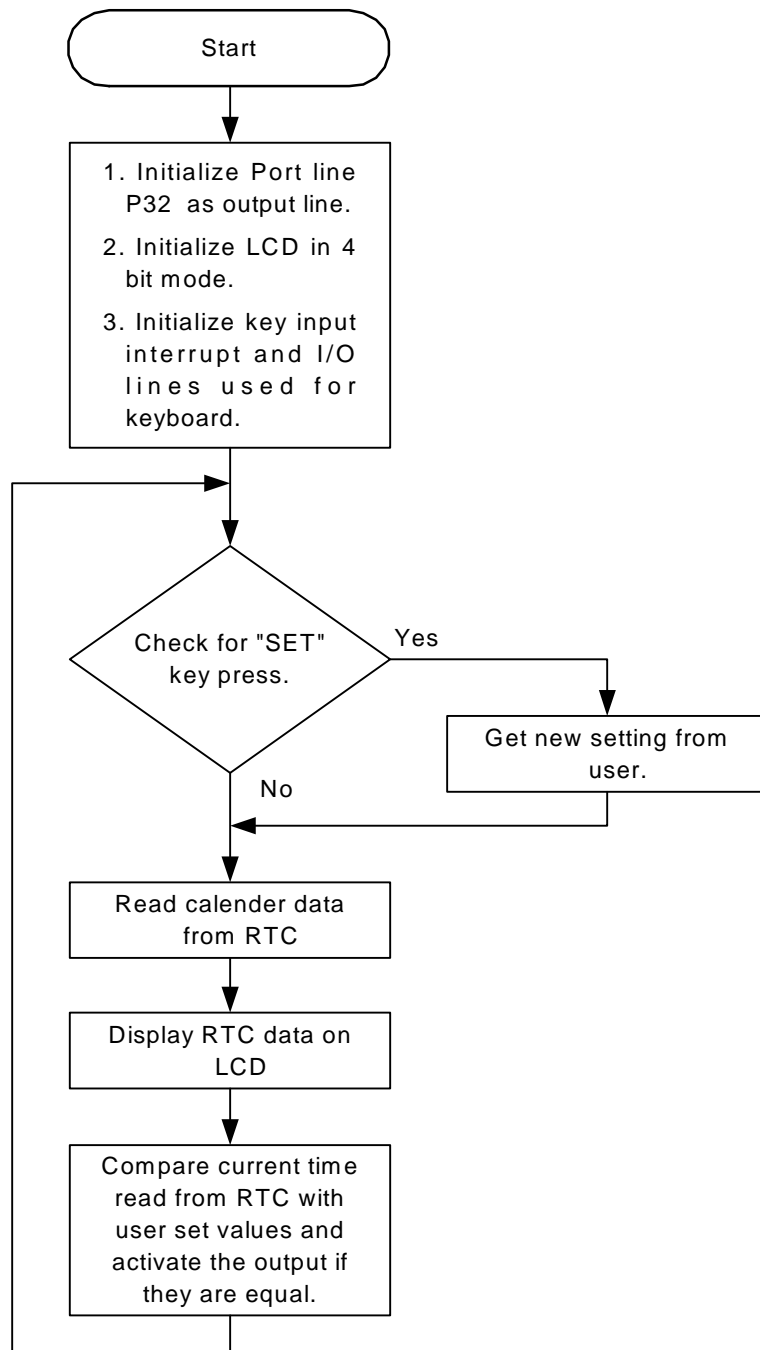
<b>Functions</b>	<b>Description</b>
DisplayLCD	Displays a message (16 Characters) on LCD on the given line number. <b>Input:</b> Line number and message string. <b>Output :</b> None.
DisplayLCD2Digit	Displays the given 2 digit number on LCD at given location. <b>Input:</b> Line number, character position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

The functions in the file "R8C1113\_FE\_Keyboard.c" and short descriptions are listed below:

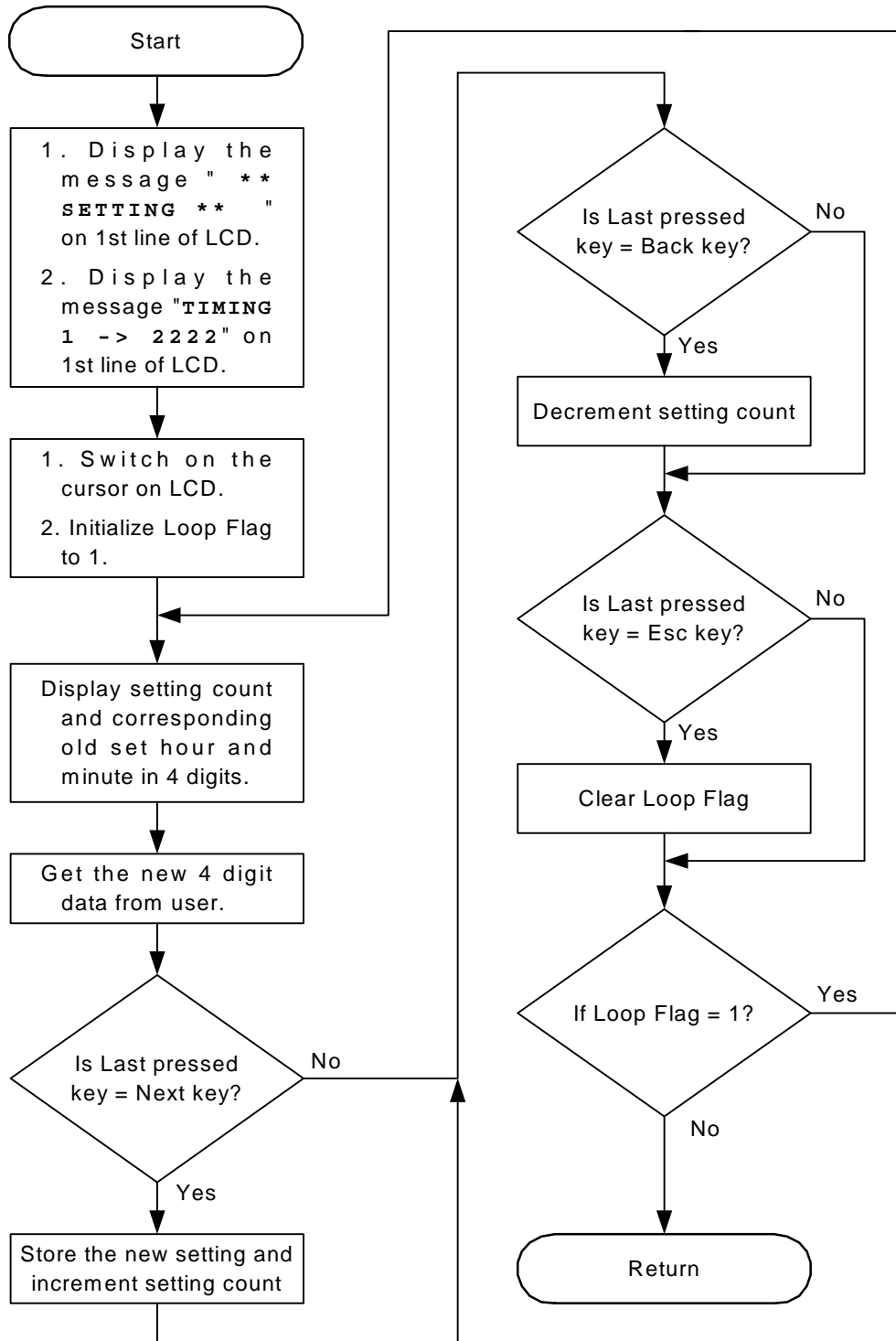
<i>Functions</i>	<i>Description</i>
ReadKeyboardStatus	Returns the keyboard <b>Input:</b> None. <b>Output :</b> Keyboard Status.
WaitForKeyPress	Waits for a key press and returns the key code of the pressed key. <b>Input:</b> None. <b>Output :</b> Key Code.
ReadKeyCode	Reads and returns the last pressed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

Program Flow:

For Main Loop



For Setting Routine



### Execute Demo:

After reset, the program will display the date, month, year, hour, minute and seconds on first line of LCD and the second line is kept blank as shown below:

```
"12JAN06 12:23:37
```

```
"
```

The program will check the current time with the user set time and activates the output if they are equal. During output activation a message will be displayed in the second line of LCD as shown below:

```
"12JAN06 12:23:37
```

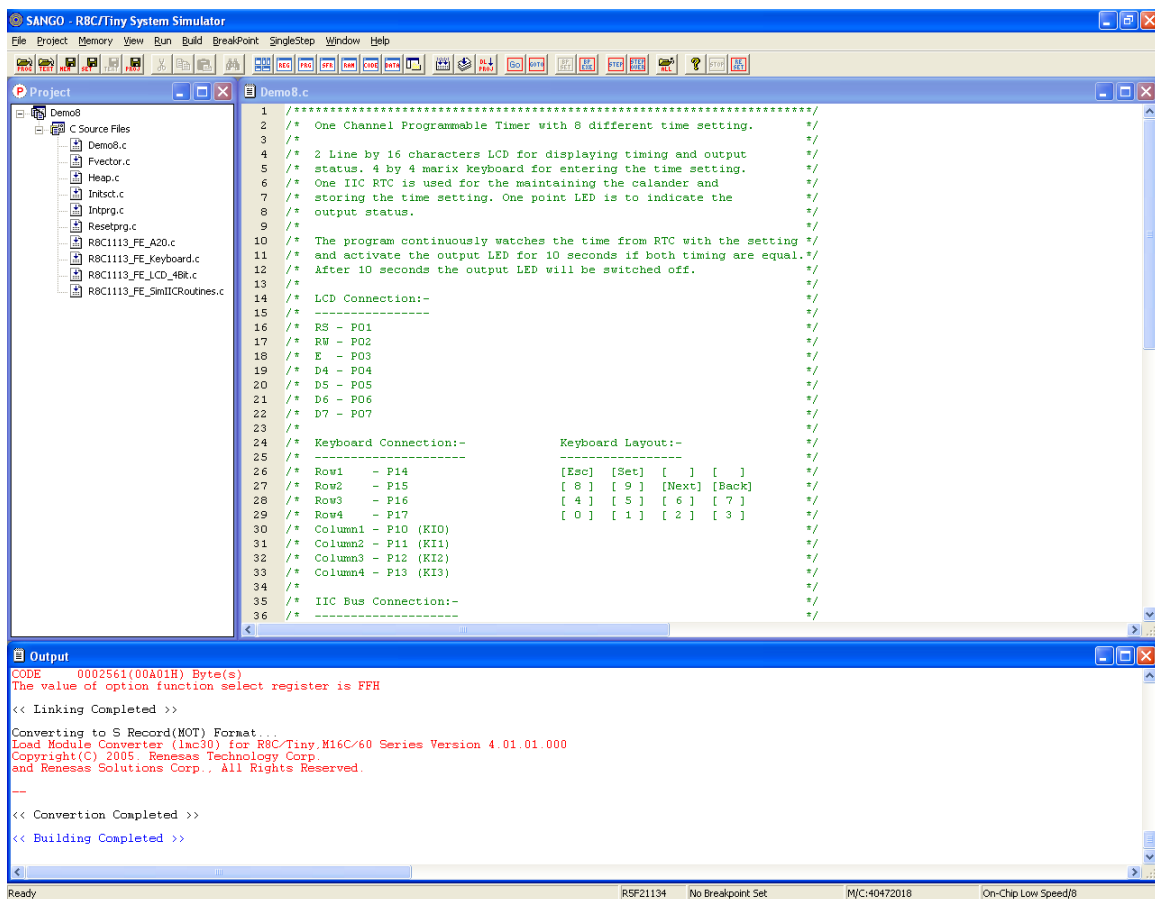
```
RELAY IS ON ->xx "
```

Where xx is the predefined on time of the output and gets decremented for every second. when it reaches zero the output will be deactivated and the second line of LCD will be cleared.

**Use Topview Simulator to Verify the Design.**

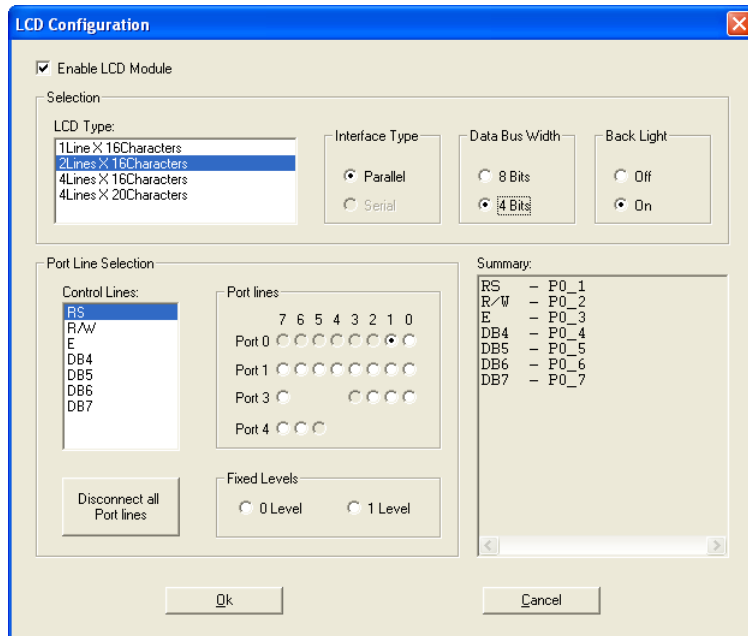
Open the project Demo8 in the R8C/Tiny System Simulator using **Open Project** option from **Project** menu. The project window opens up along with the Demo8.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler output.

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

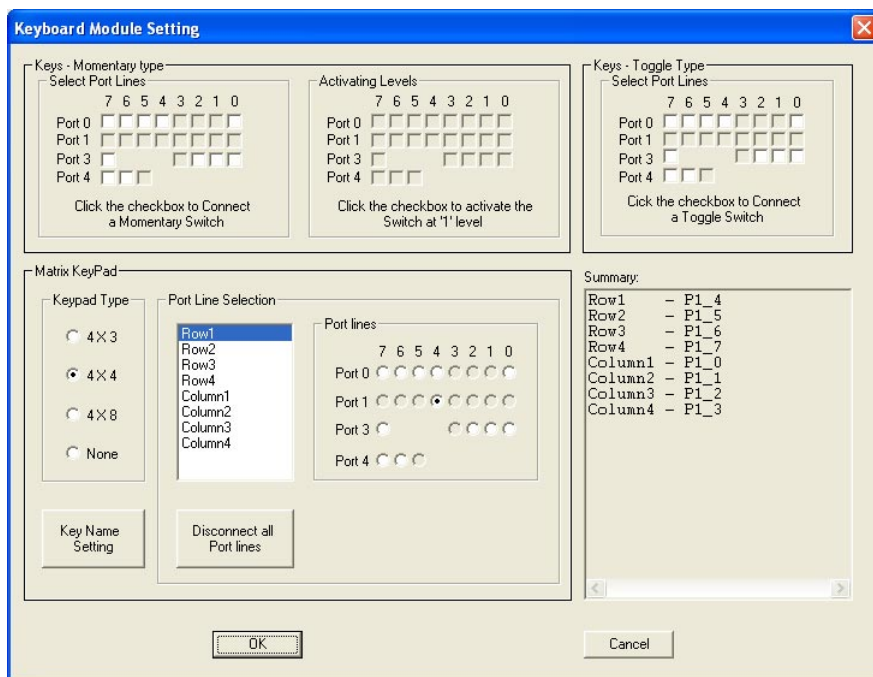


Do the settings to the LCD modules as shown. Connect LCD control and data lines to port lines P01 to P07.

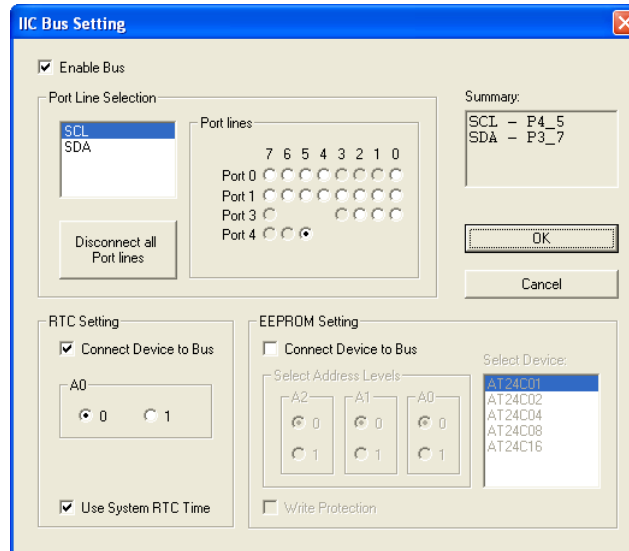




Make the setting for keyboard module as shown below:

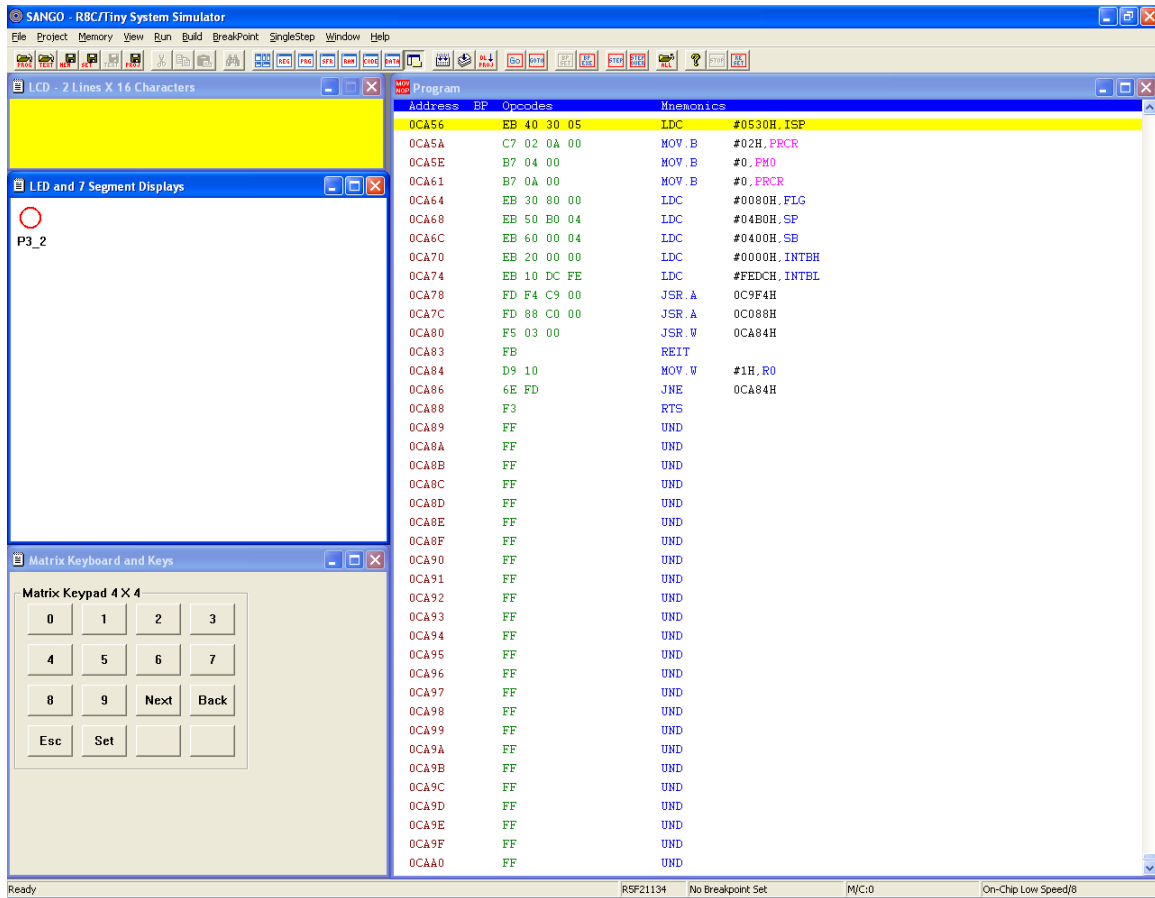


Connect the port lines P45 and P37 to IIC lines SCL and SDA using IIC module setting and enable RTC function as shown below:



Connect one point LED to the port line P32 using LED module setting.

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



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

Run the program using **Go** command in **Run** menu. The message,

The program continuously watches the time from RTC with the setting and activate the output LED for 10 seconds if both timing are equal. After 10 seconds the output LED will be switched off.