Introduction:

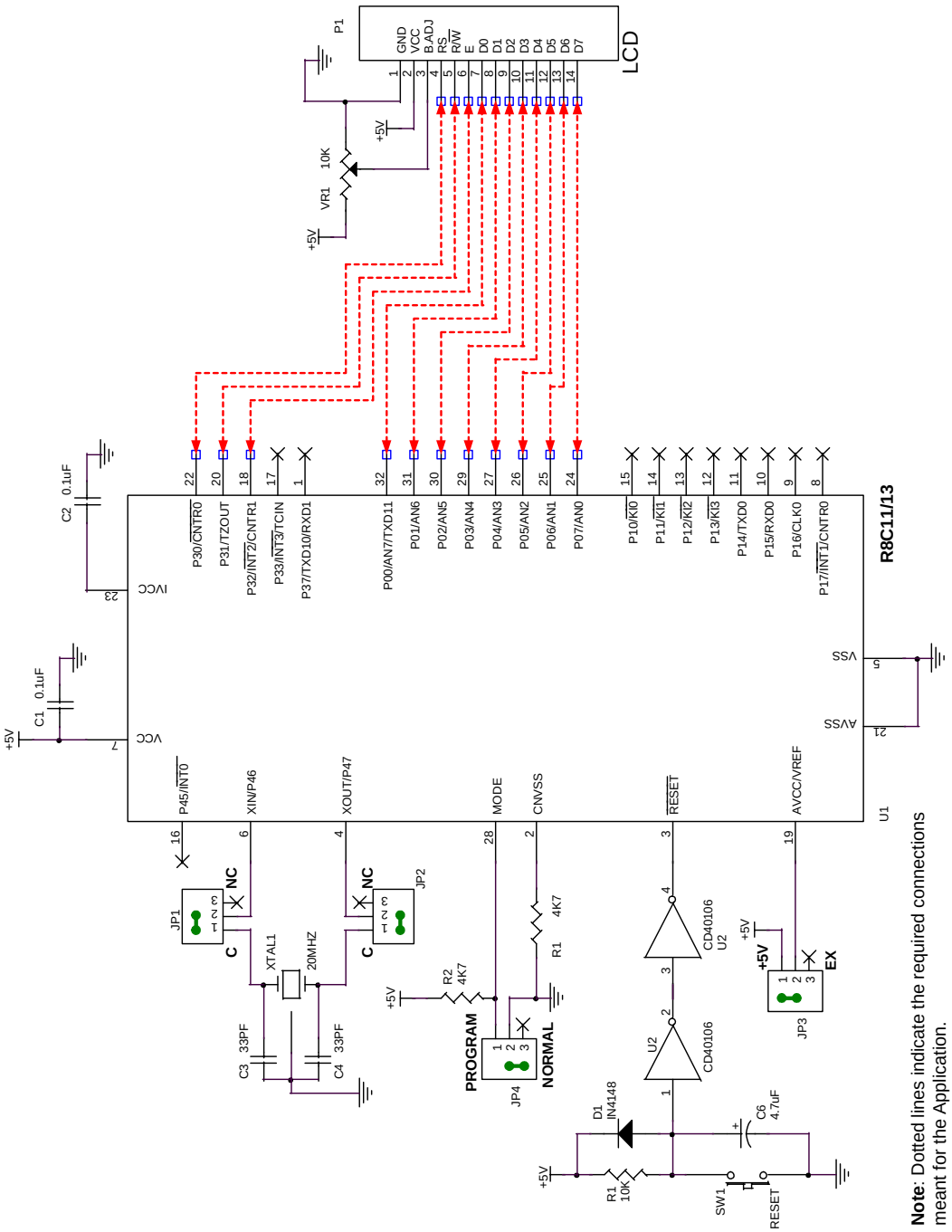
This study discusses the possibilities using LCD in an embedded application. The LCD module can be interfaced either using the 8 bit or 4 bit bus. Here the LCD module is connected using an 8 bit data bus.

Demo Hardware:

The LCD module interfacing is simple one. Normally, LCD modules can be interfaced either using 8 bit data bus or 4 bit data bus. Apart from the data bus, few control lines, RS R/W and E are required.

For a control using an 8 bit bus, totally 11 port lines are required. Here the data bus is connected with Port 0 and the control lines are interfaced with P30-P32 port lines.

Circuit Connection Using FE EADS:



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

Connections:

D0 -> P00
D1 -> P01
D2 -> P02
D3 -> P03
D4 -> P04
D5 -> P05
D6 -> P06
D7 -> P07

RS -> P30
RW -> P31
E -> P32

Functional Description:

LCD Module consists of the LCD and a Controller. The LCD controller has Display Data RAM (DDRAM), Character Generator ROM (CGROM), Instruction Register (IR), Data Register (DR) and Address Counter (AC).

DDRAM has a capacity of 80 bytes and it holds the display data. DR register temporarily holds the display data before writing it to DDRAM. This can be read or written directly. The capacity of CGROM is 9920 bits for 240 characters. It holds the pixel details of each ASCII character in two different formats, 5 X 8 dots and 5 X 10 dots.

The LCD controller gets data from the DDRAM, which is an ASCII value of a character and reads the equivalent display data from the CGROM and sends it to the LCD for displaying. This is the basic operation that takes place in the LCD module.

The communication with the LCD Module is through the IR and DR registers. The IR register stores the Instructions codes like clear display, cursor shift and etc. When an address is written into IR, it is sent to the Address counter (AC). The address counter holds the address of the DDRAM always.

Some of the LCD commands are.

- Clear Display
- Return Home
- Entry Mode Set
- Display On/off control
- Cursor or Display shift
- Function Set.
- Set DDRAM address.
- Write data to DDRAM.
- Read data from DDRAM.

Instruction	RS,R/W	DB7 to DB0	Description
Clear Display	0 0	0 0 0 0 0 0 0 1	Clears Display and sets DDRAM address 0 in Address Counter
Return Home	0 0	0 0 0 0 0 0 1 -	Sets DDRAM address 0 in AC. Returns display to home position.
Entry Mode Set	0 0	0 0 0 0 0 1 I/D S	Sets cursor move direction & display shift. I/D = 1- Increment by 1, S = 0 – No shift
Display On/Off Control	0 0	0 0 0 0 1 D C B	Sets display(D) on/off, Cursor on/off (C), blinking of cursor position (B)
Cursor or Dislay Shift	0 0	0 0 0 1 S/C R/L - -	Moves Cursor and shifts displaywithout changing DDRAM contents.
Function Set	0 0	0 0 1 DL N F - -	Sets interface Data length (DL), no. of display lines(N) and Character Font(F)
Set DDRAM Address	0 0	1 A A A A A A A	Sets DDRAM address
Read Status	0 1	BF AC AC AC AC AC AC AC	Reads busy flag (BF) and Address counter (AC) contents.
Read Data from DDRAM	1 1	Read Data	Reads data from DDRAM
Write Data to DDRAM	1 0	Write Data	Writes data to DDRAM

Generally the following are the timing requirements for the LCD operations. The application programs, should take them into account.

- Clear Display - 1640 microseconds
- Return Home - 1640 microseconds.
- Entry Mode Set - 40 microseconds.
- Display On/off control - 40 microseconds.
- Cursor or Display shift - 40 microseconds.
- Function Set - 40 microseconds.
- Set DDRAM address - 40 microseconds.
- Write data to DDRAM - 40 microseconds.
- Read data from DDRAM - 40 microseconds.

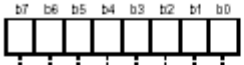
The above timing are assumed for the standard oscillator frequency of LCD, fosc = 250KHz.

Registers used:

PD0 - Port 0 Direction Register
P0 - Port 0 Data Register
PD3 - Port 0 Direction Register
P3 - Port 0 Data Register
PRCR - Protect Register

Port Pi Direction Register

Port Pi direction register (i=0, 1, 3, 4)[1, 2, 3]



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 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.

To set a port line as output line, load the corresponding bit of the direction register with a value of 1 level and for the output line the value should be 0 level. After reset all port lines are set to input mode, which indicates that all direction registers are already set with a value 0.

Port Pi Data Register

Port Pi register (i=0, 1, 3, 4)[1, 2]

Symbol	Address	After reset
P0	00E0$_{16}$	Indeterminate
P1	00E1$_{16}$	Indeterminate
P3	00E5$_{16}$	Indeterminate
P4	00E8$_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
Pi_0	Port Pi0 bit	The pin level on any I/O port which is set for input mode can be read by reading the corresponding bit in this register. The pin level on any I/O port which is set for output mode can be controlled by writing to the corresponding bit in this register 0 : "L" level 1 : "H" level (i = 0, 1, 3, 4)	RW
Pi_1	Port Pi1 bit		RW
Pi_2	Port Pi2 bit		RW
Pi_3	Port Pi3 bit		RW
Pi_4	Port Pi4 bit		RW
Pi_5	Port Pi5 bit		RW
Pi_6	Port Pi6 bit		RW
Pi_7	Port Pi7 bit		RW

Notes:
1. Nothing is assigned to the P3_4 to P3_6 bits in the P3 register. When writing to the P3_4 to P3_6 bits, write "0" ("L" level). When read, its content is indeterminate.
2. Nothing is assigned to the P4_0 to P4_4 bits in the P4 register. When writing to the P4_0 to P4_4 bits, write "0" ("L" level). When read, its content is indeterminate.

To write a data to a port, write the data in the respective port register (Pi). To read a data from the port, read from the respective port register.

Protect Register

Protect register

b7	b6	b5	b4	b3	b2	b1	b0
		0	0				

Symbol PRCR | Address 000A16 | After reset 00XXX0002

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enable write to CM0, CM1, OCD, HR0, HR1 registers 0 : Write protected 1 : Write enabled	RW
PRC1	Protect bit 1	Enable write to PM0, PM1 registers 0 : Write protected 1 : Write enabled	RW
PRC2	Protect bit 2	Enable write to PD0 register 0 : Write protected 1 : Write enabled[1]	RW
PRC3	Protect bit 3	Enable write to VCR2, D4INT registers 0 : Write protected 1 : Write enabled	RW
$\overline{\text{(b5-b4)}}$	Reserved bit	When write, should set to "0"	RW
$\overline{\text{(b7-b6)}}$	Reserved bit	When read, its content is "0".	RO

Notes:
1. The PRC2 bit is set to "0" by writing to any address after setting it to "1". Other bits are not set to "0" by writing to any address, and must therefore be set to "0" in a program.

Protect Register provides protection to important system control registers and Port 0 direction register (PD0). Before writing to PD0 set the bit 2 (PRC2) of PRCR register. After writing to PD0 clear the bit PRC2, which provides protection and disables further writing in the PD0.

Software Description:

LCD Initialization:

Care should be taken in initializing the LCD in 8 bits mode. Otherwise LCD initialization will not be proper. The details are provided based on the Hitachi's standard LCD controller chip IC HD44780.

LCD is interfaced with the controller using 8-bit bus. LCD is enabled with the control line E. The R/W in the LCD is connected to a port line. The RS line is used to select between the command word and data word.

Separate routines are available to initialize LCD, to write a message in the LCD, display a 2 digit hexadecimal data on the LCD and other necessary routines.

Message Display:

To the Message display routine, the starting location, line number and a pointer to the message is sent. The routine positions the cursor to the specified line and the location. Then using the pointer the ASCII value of the charcter is read and it is displayed on the LCD. The process is repeated until the end of the message.

The files used in this module are listed below:

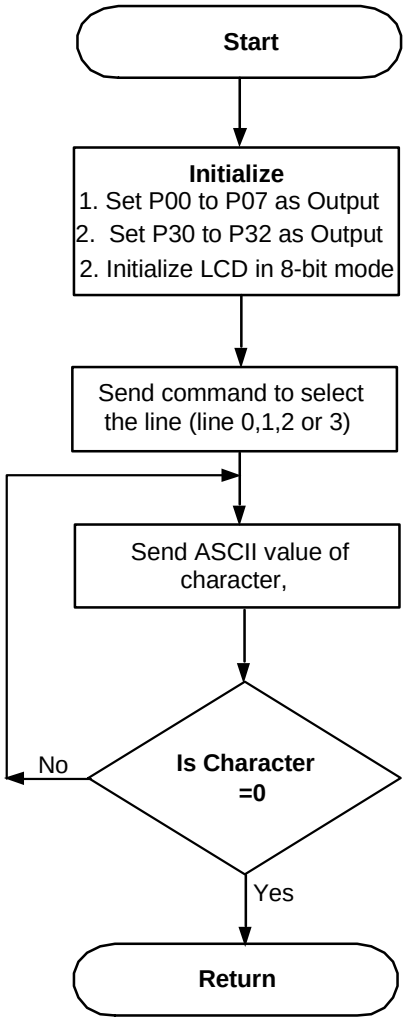
Files	*Description*
Demo3.C	LCD routines to initialize LCD, cursor on/off, display a message etc.
LCDDisplay.h	Declarations of functions in Demo3.C

The **Demo3.C** contains all routines required for the LCD module to operate in 8 bit mode.

Files	*Description*
Main	Main routine Displays a message of 4 lines on LCD **Input**: Line number and message string. **Output** : None.
DisplayString	Displays a string on the selected line of LCD. **Input**: Line number, character position and string pointer. **Output** : None.
Clear LCD	Clears LCD. **Input**: None. **Output** : None.
ReturnHomeLCD	Positions the cursor to the first line and first character . **Input**: None. **Output** : None.
InitializeLCD	Initializes the I/O lines used by LCD and LCD in 8 bit mode. Clears the LCD. **Input**: None. **Output** : None.
SendData	Write a data byte to LCD. **Input**: Data Byte. **Output** : None.
SendCommand	Write a command byte to LCD. **Input**: Command Byte. **Output** : None.

Program Flow:

Flow for Message Display:



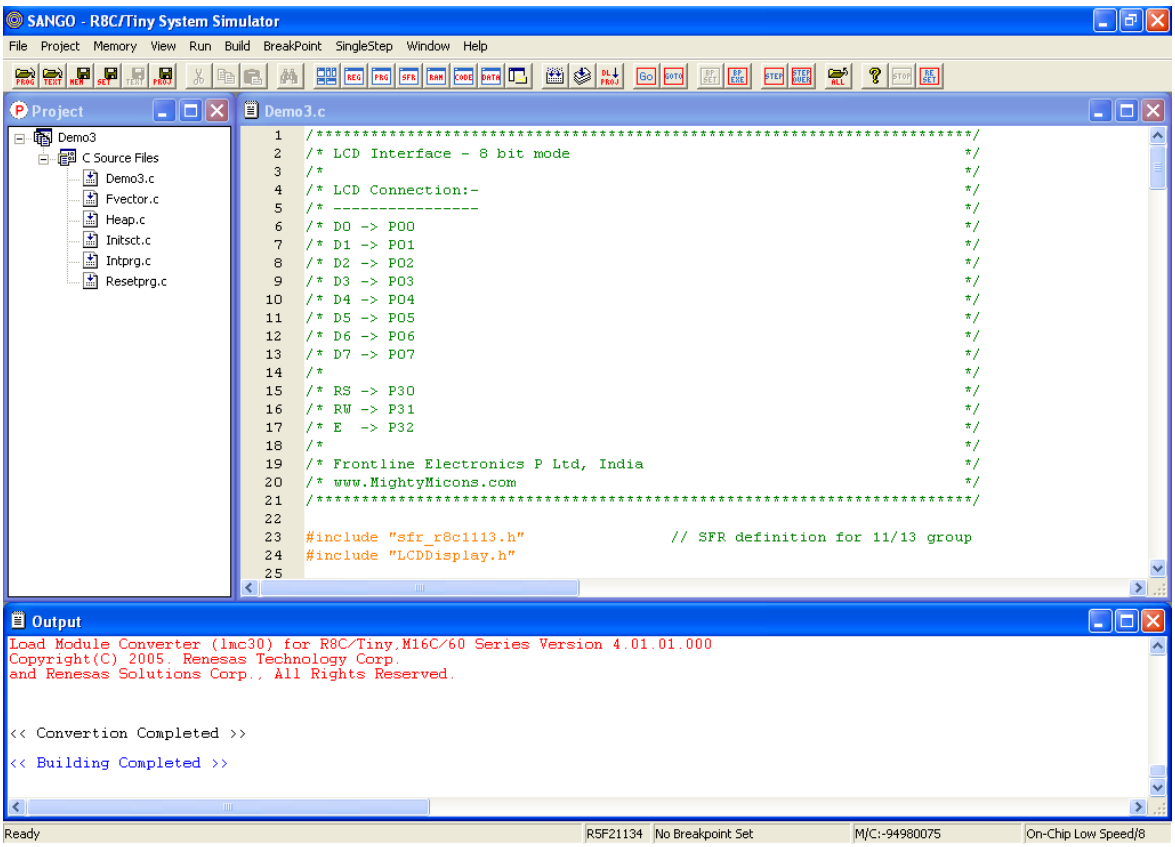
Execute Demo:

A 4 line message gets displayed on the LCD.

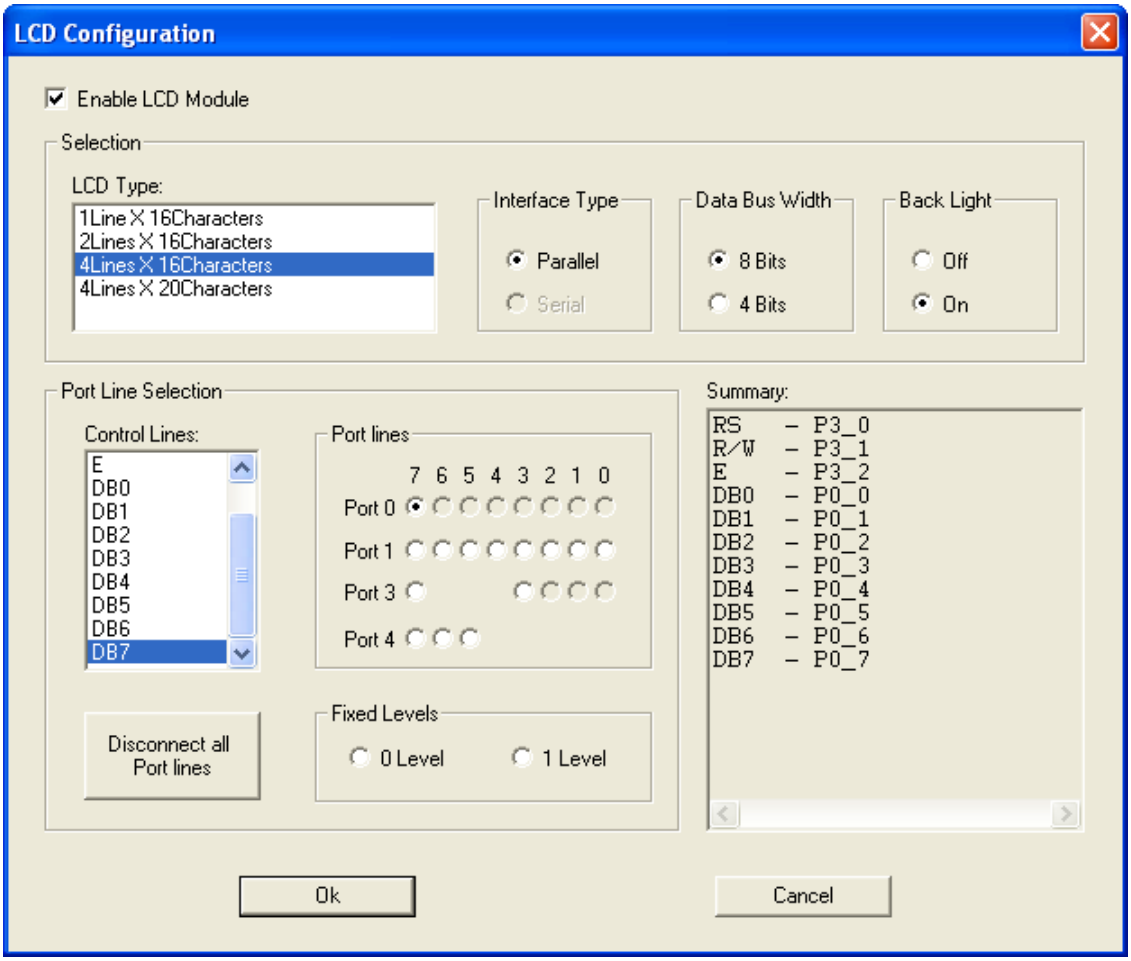
Use Topview Simulator to Verify the Design.

Open the project Demo3 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo3.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

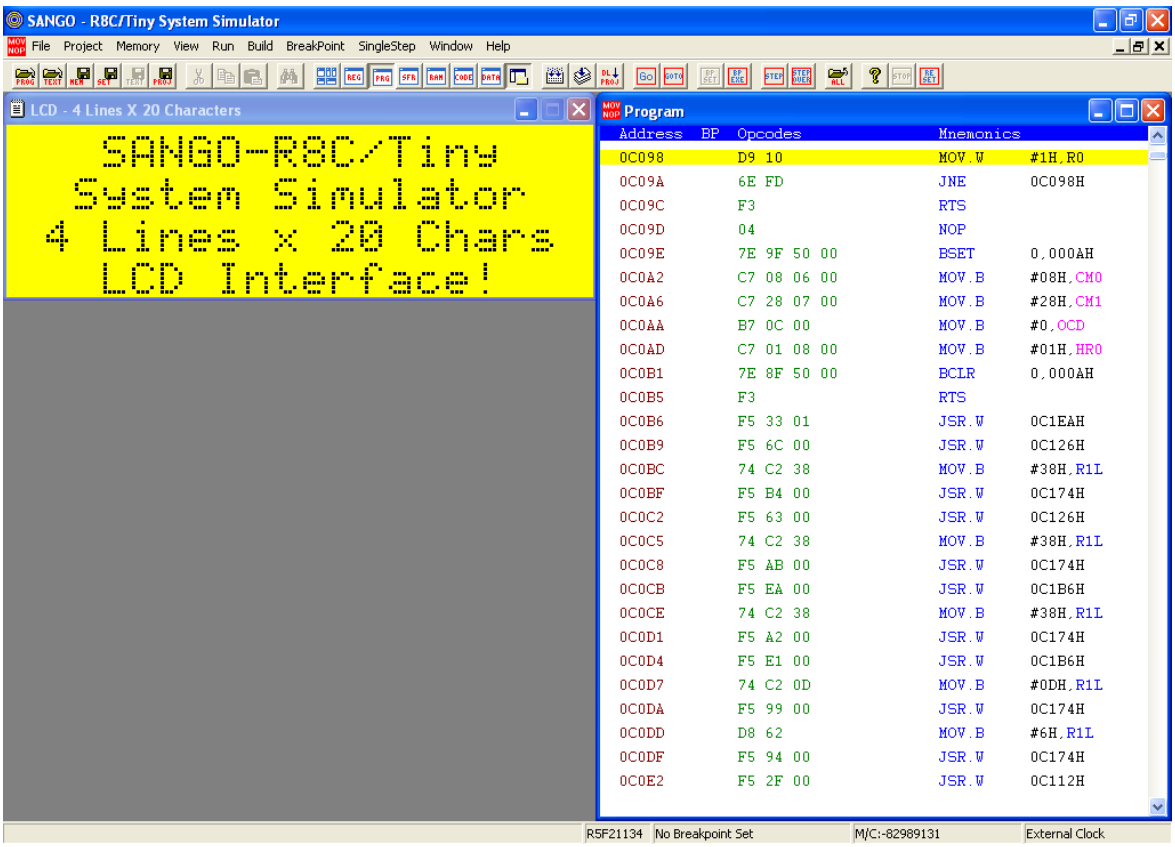
Use **Project** -> **Download Project** from main menu to download the Demo3.mot file into the simulator's memory for simulation.



Open the LCD Module settings window and do the settings to the LCD module as shown. Connect Data lines D0 to D7 of the display to the port lines P00 to P07. Connect the control lines RS, R/W and E to P30, P31 and P32 respectively.



Then open the **LCD window** usign the option **View** -> **External Modules** -> **LCD** and the Program Window.



Run the program using **Go** from the **Run** menu. The program will display a 4 line message on the LCD.