Introduction:

Gives an idea about using the LIN bus to transfer a set of data from one R8C 24/25 based system (called Master node) to another R8C 24/25 based system (called slave node). The pressed key value is sent to the slave using the TXD line of the serial port 0.

Hardware:

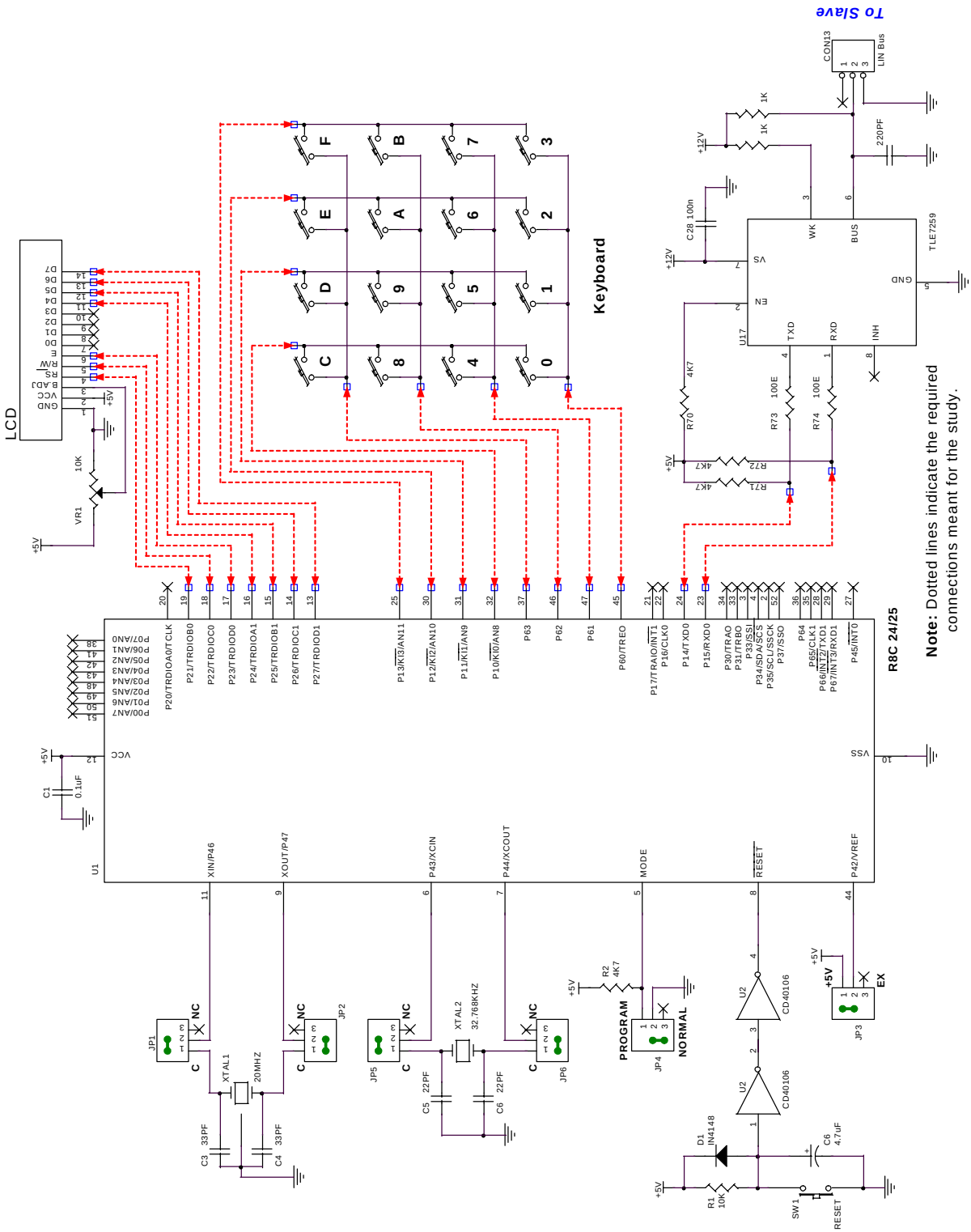
For studying the hardware LIN, one master and a slave node is required. The micon's LIN acts as master and simulated slaves nodes can be used as slave. The slave have it's own ID. The master will send the data to the slave by sending the Syn break signal, IDs first and the data next.

A 2 line by 16 characters LCD is used for the display purpose. 4 X 4 matrix keyboard is used for selecting the data.

Port lines P21 to P23 are used to control LCD and connected to RS, R/W and E pins of LCD. The port lines P24 to P27 are connected to upper data bus D4 to D7 of LCD to send data and command.

Port lines P10 to P13 are connected to the column lines and port lines P14 to P17 are connected to the row lines of the matrix keyboard.

Circuit:


To Slave
Keyboard
LCD
R8C 24/25
TLE7259
CD40106
Note: Dotted lines indicate the required connections meant for the study.

Connections:

LCD Connection:-

RS - P21
RW - P22
E - P23
D4 - P24
D5 - P25
D6 - P26
D7 - P27

Keyboard Connection:-

Row0 - P60
Row1 - P61
Row2 - P62
Row3 - P63
Column0 - P10 (KI0)
Column1 - P11 (KI1)
Column2 - P12 (KI2)
Column3 - P13 (KI3)

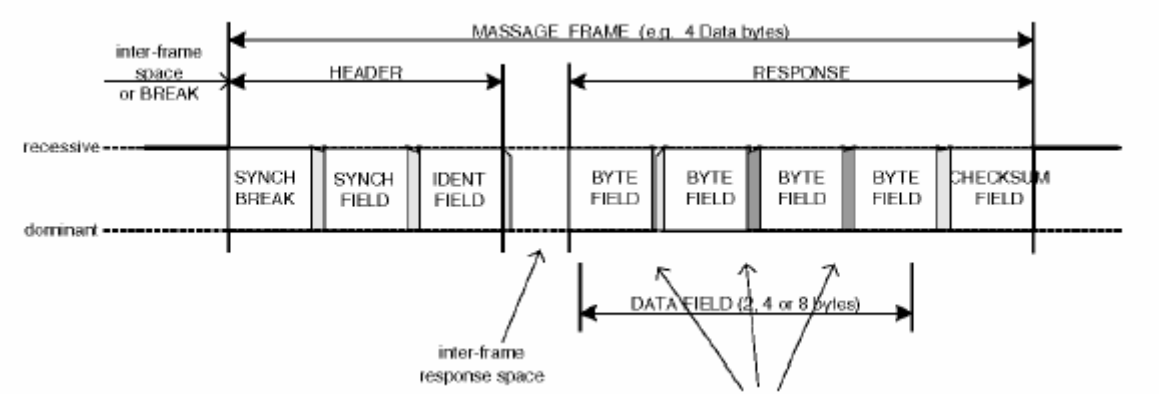
Functional Description:

The serial communication protocol LIN is qualified to link distributed systems especially in automotive applications. LIN will be used as a standardized local subbus within clustered systems. Moreover, this bus means a low cost silicon implementation, because LIN is based on a standard UART data format.

LIN is a single-master multiple-slave bus system. One master node and several slaves build up the bus system. The LIN bus is a single-wire bus system. The master node contains a master task and a slave task, whereas slave nodes only contain slave tasks only.

A fixed message format is defined for the LIN protocol. Each LIN message starts with a header consisting of a synchbreak followed by a synchfield and an identifier-field. Only the master task

may send this frame. Next 2, 4, or 8 bytes code the message data. A checksum field completes the LIN message (Figure 5). This response to the header is sent either by the slave task of the master node or by one of the several slaves nodes hooked up to the LIN bus.



LIN Message Frame

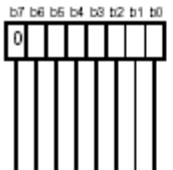
Main features of the LIN bus

- Single-master / multiple-slave
- Use of an ordinary UART / SCI
- Self synchronization
- Low-cost single-wire (enhanced ISO 9141)
- Easy connection with single-wire-transceivers
- Speed up to 20 kBits/sec
- Guaranteed latency times
- Message Frame contains 2,4 or 8 data bytes
- Multi-cast reception
- Checksum calculation (inverted modulo 256 checksum)
- Recognition of defective nodes
- Error-Detection
- Low cost

Registers Used:

U0MR - UART0 Transmit/Receive Mode register
U0C0 - UART0 Transmit/Receive Control register 0
U0C1 - UART0 Transmit/Receive Control register 1
U0BRG - UART0 bit rate register
TRAMR - Timer RA mode Register
TRAPRE - Timer RA Prescaler Register
TRA - Timer RA Register
LINCR - LIN Control Register
LINST - LIN Status REgister

U0MR - UART0 Transmit/Receive Mode Register:



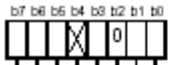
Symbol	Address	After Reset
U0MR	00A0h	00h
U1MR	00A8h	00h

Bit Symbol	Bit Name	Function	RW
SMD0	Serial Interface Mode Select Bit[2]	b2 b1 b0 0 0 0 : Serial interface disabled 0 0 1 : Clock synchronous serial I/O mode 1 0 0 : UART mode transfer data 7 bits long 1 0 1 : UART mode transfer data 8 bits long 1 1 0 : UART mode transfer data 9 bits long Other than above : Do not set	RW
SMD1			RW
SMD2			RW
CKDIR	Internal / External Clock Select Bit	0 : Internal clock 1 : External clock[1]	RW
STPS	Stop Bit Length Select Bit	0 : 1 Stop Bit 1 : 2 Stop Bits	RW
PRY	Odd / Even Parity Select Bit	Enables when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity Enable Bit	0 : Parity disabled 1 : Parity enabled	RW
— (b7)	Reserved Bit	Set to "0"	RW

NOTES :
1. Set the PD1_6 bit in the PD1 register to "0" (input).
2. Do not set the SMD2 to SMD0 bits in the U1MR register to any values other than "000b", "100b", "101b" and "110b".

The data H'05 is moved to U0MR register to select No parity, One stop bit and Internal Clock.

U1C0 - UART1 Transmit/Receive Control Register 0:



Symbol	Address	After Reset
U0C0	00A4h	00001000b
U1C0	00ACh	00001000b

Bit Symbol	Bit Name	Function	RW
CLK0	BRG Count Source Select Bit	b1 b0 0 0 : Selects f1 0 1 : Selects f8 1 0 : Selects f32 1 1 : Do not set	RW
CLK1			RW
— (b2)	Reserved Bit	Set to "0"	RW
TXEPT	Transmit Register Empty Flag	0 : Data in transmit register (during transmit) 1 : No data in transmit register (transmit completed)	RO
— (b4)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—
NCH	Data Output Select Bit	0 : TXDi pin is a pin of CMOS output 1 : TXDi pin is a pin of N-channel open drain output	RW
CKPOL	CLK Polarity Select Bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	RW
UFORM	Transfer Format Select Bit	0 : LSB first 1 : MSB first	RW

Data H'00 is set to register U1C0 register to select following options:

f1SIO clock is selected,

TXD1 pin is a pin of CMOS,

Transmit data is output at falling edge of transfer clock and receive data is input at rising edge,

LSB First.

U1C1 - UART1 Transmit/Receive Control Register 1:

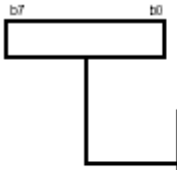
Symbol	Address	After Reset
U0C1	00A9h	00000010b
U1C1	00ADh	00000010b

Bit Symbol	Bit Name	Function	RW
TE	Transmit Enable Bit(1)	0 : Disables transmit 1 : Enables transmit	RW
TI	Transmit Buffer Empty Flag	0 : Data in UiTB register 1 : No data in UiTB register	RO
RE	Receive Enable Bit	0 : Disables receive 1 : Enables receive	RW
RI	Receive Complete Flag(1)	0 : No data in UiRB register 1 : Data in UiRB register	RO
UiIRS	UARTi Transmit Interrupt Cause Select Bit	0 : Transmit buffer empty (TI=1) 1 : Transmit completed (TXEPT=1)	RW
UiRRM	UARTi Continuous Receive Mode Enable Bit	0 : Disables continuous receive mode 1 : Enables continuous receive mode (This bit is disabled in UART mode)	RW
— (b7-b4)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

NOTES :
1. The RI bit is set to "0" when the higher byte of the UiRB register is read out.

The register U0C1 is loaded with H'05 to enable transmission and reception.

U0BRG - UART 0 Bit Rate Register:



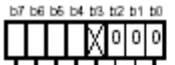
Symbol	Address	After Reset
U0BRG	00A1h	Indeterminate
U1BRG	00A9h	Indeterminate

Function	Setting Range	RW
Assuming that set value is n, UiBRG divides the count source by n+1	00h to FFh	WO

NOTES :
1. Write to this register while the serial I/O is neither transmitting nor receiving.
2. Use the MOV instruction to write to this register.

The baud rate 9600 is selected by loading data 129 to U0BRG register.

TRAMR - Timer RA Mode Register:



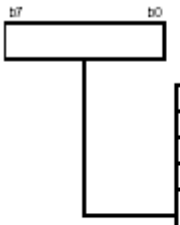
Symbol: TRAMR | Address: 0102h | After Reset: 00h

Bit Symbol	Bit Name	Function	RW
TMOD0(1)	Timer RA Operation Mode Select Bit	b2 b1 b0 0 0 0 : Timer mode	RW
TMOD1(1)			RW
TMOD2(1)			RW
— (b3)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—
TCK0(2)	Timer RA Count Source Select Bit	b5 b5 b4 0 0 0 : f1 0 0 1 : f8 0 1 0 : fOCO 0 1 1 : f2 1 0 0 : fC32 1 0 1 : } Do not set 1 1 0 : } Do not set 1 1 1 : } Do not set	RW
TCK1(2)			RW
TCK2(2)			RW
TCKCUT(2)	Timer RA Count Source Cutoff Bit	0 : Provides count source 1 : Cuts off count source	RW

NOTES :
1. Select operation mode while the count stops.
2. Do not switch or cut off a count source during a count operation. Stop the timer count before switching or cutting off a count source.

Timer RA mode register TRAMR is loaded with H'00 to set the timer RA in timer mode and to select f1 as source clock..

TRAPRE - Timer RA Prescaler Register:



Symbol: TRAPRE | Address: 0103h | After Reset: FFh

Mode	Function	Setting Range	RW
Timer Mode	Counts an internal count source	00h to FFh	RW
Pulse Output Mode	Counts an internal count source	00h to FFh	RW
Event Counter Mode	Counts an external count source	00h to FFh	RW
Pulse Width Measurement Mode	Measure pulse width of input pulses from external (counts internal count source)	00h to FFh	RW
Pulse Period Measurement Mode	Measure pulse period of input pulses from external (counts internal count source)	00h to FFh	RW

NOTES :
1. The TRAPRE register is initialized by the TSTOP bit in the TRACR register.

The prescaler register TRAPRE is loaded with 149 to divide the selected input clock to the timer RA by 150.

TRA - Timer RA Register:

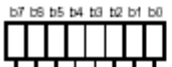
Symbol	Address	After Reset
TRA	0104h	FFh

Mode	Function	Setting Range	RW
All modes	Counts of an underflow of the Timer RA Prescaler Register	00h to FFh	RW

NOTES :
1. The TRA register is initialized by the TSTOP bit in the TRACR register.

The timer RA register TRA is loaded with 199 to generate a sync break signal of width 1500 micro seconds.

LINCR - LIN Control Register:



Symbol	Address	After Reset
LINCR	0106h	00h

Bit Symbol	Bit Name	Function	RW
SFIE	Sync Field measurement-completed interrupt enable bit	0 : Disables Sync Field measurement-completed interrupt 1 : Enables Sync Field measurement-completed interrupt	RW
SBIE	Sync Break detection interrupt enable bit	0 : Disables Sync Break detection interrupt 1 : Enables Sync Break detection interrupt	RW
BCIE	Bus collision detection interrupt enable bit	0 : Disables bus collision detection interrupt 1 : Enables bus collision detection interrupt	RW
RXDSF	RXD0 input status flag	0 : RXD0 input enabled 1 : RXD0 input disabled	RO
LSTART	Sync Break detection start bit(1)	When this bit is set to "1", Timer RA input is enabled and RXD0 input is disabled. When read, its content is "0".	RW
SBE	RXD0 input unmasking timing select bit (effective in only slave mode)	0 : Unmasked after Sync Break is detected 1 : Unmasked after Sync Field measurement is completed	RW
MST	LIN operation mode setting bit(2)	0 : Slave mode (Sync Break detection circuit actuated) 1 : Master mode (timer RA output OR'ed with TXD0)	RW
LINE	LIN operation start bit	0 : Causes LIN to stop 1 : Causes LIN to start operating (Inputs to timer RA and UART0 are prohibited immediately after this bit is set to "1")	RW

NOTES:
1. After setting the LSTART bit, confirm that the RXDSF flag is set to "1" before Sync Break input starts.
2. Before changing LIN operation modes, temporarily stop the LIN operation (LINE bit = 0).

Master mode is selected by setting the MST bit in LINCR register.

LINST - LIN Status register:

Symbol: LINST | Address: 0107h | After Reset: 00h

Bit Symbol	Bit Name	Function	RW
SFDCT	Sync Field measurement-completed flag	"1" shows Sync Field measurement completed.	RO
SBDCT	Sync Break detection flag	"1" shows Sync Break detected	RO
BCDCT	Bus collision detection flag	"1" shows Bus collision detected	RO
B0CLR	SFDCT bit clear bit	When this bit is set to "1", SFDCT bit is set to "0". When read, its content is "0".	RW
B1CLR	SBDCT bit clear bit	When this bit is set to "1", SBDCT bit is set to "0". When read, its content is "0".	RW
B2CLR	BCDCT bit clear bit	When this bit is set to "1", BCDCT bit is set to "0". When read, its content is "0".	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is "0".		—

The LIN status register is loaded with the data H'00.

Software Description:

Data from H'00 to H'0F is sent through LIN to the slave. A 2 Line X 16 characters LCD is used to display message and 4 X 4 matrix keyboard is used to enter data. The serial port 0 is configured to send data at 9600 baud rate @ 20MHZ.

After reset,

1. The external crystal oscillator is selected as clock source for MCU and other peripherals.
2. Serial port 0 is initialized in 9 bits mode at 9600.
3. LCD is initialized in 4 bit mode.
4. Keyboard is initialized.
5. The Message "LIN Study-Master" is displayed on first line of LCD.

After the initialization, in a continuous loop, the keyboard is scanned. The pressed key value is sent to the slave through LIN bus by sending the Sync break signal, Sync byte (H'55) and ID of

the selected slave first and then the key value with a dummy data. The pressed key value is displayed in the second line of LCD.

The files used in this module are listed below:

Files	*Description*
Demo22.C	Main file for this module, will read key value and send the same to the slaves through LIN bus. Displays the key value and the selected slave on the LCD.
R8C2425_FE_LCD_4Bit.C	LCD routines to initialize LCD, cursor on/off, display a message etc.
R8C2425_FE_LCD_4Bit.H	Declarations of functions in R8C2425_FE_LCD_4Bit.C
R8C2425_FE_Keyboard.C	Keyboard routines to initialize Key input interrupts, I/O lines used by keyboard and key input interrupt service routine.
R8C2425_FE_Keyboard.H	Declarations of functions in R8C2425_FE_Keyboard.C

The functions in the file "**Demo22.C**" and short descriptions are listed below:

Files	*Description*
Main	Reads the pressed key value and sends to slave through LIN bus. **Input**: None. **Output** : None.
MCUInitialize	Selects the external crystal oscillator as clock source for the CPU and other peripherals. **Input**: None. **Output** : None.
InitializeLIN	Initializes the Hardware LIN in master mode. **Input**: None. **Output** : None.
SendOneFrameLIN	Sends one frame to the slave. **Input**: Selected slave and data. **Output** : None.

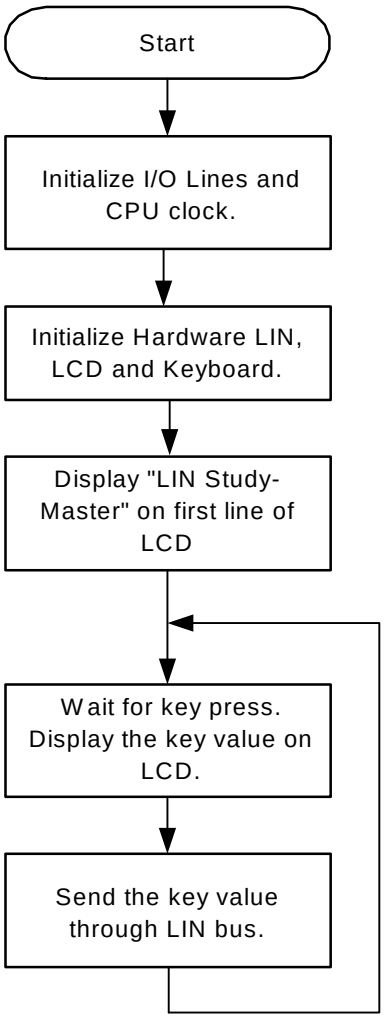
The functions in the file "**R8C2425_FE_LCD_4Bit.C**" and short descriptions are listed below:

Files	*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 "**R8C2425_FE_Keyboard.C**" and short descriptions are listed below:

Files	*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:



Execute Demo:

After reset, the message,

“**LIN Study-Master**
Node:1 Data: “

is displayed on the LCD.

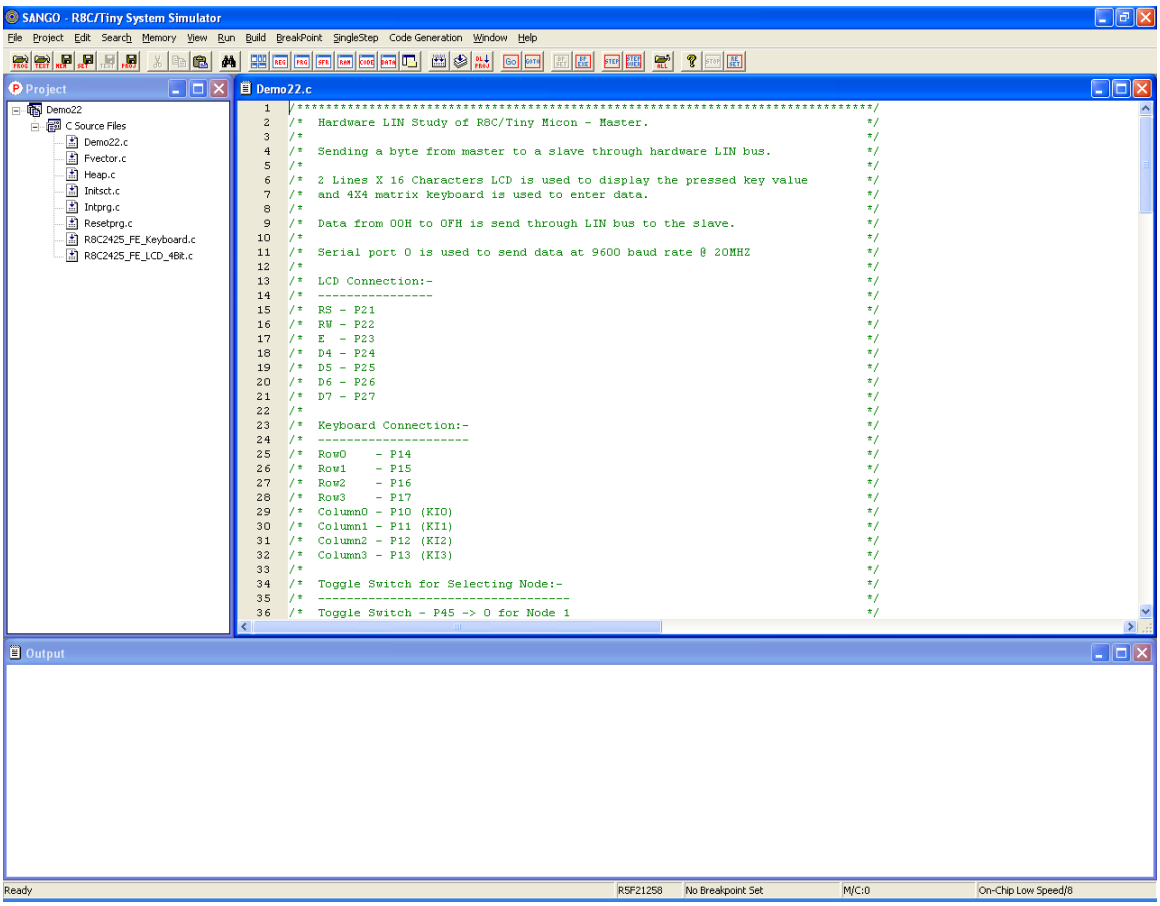
The pressed key value (H’00 to H’0F) is sent through hardware LIN after displaying the key value on LCD as shown below:

“**LIN Study-Master**
Node:1 Data:04 “

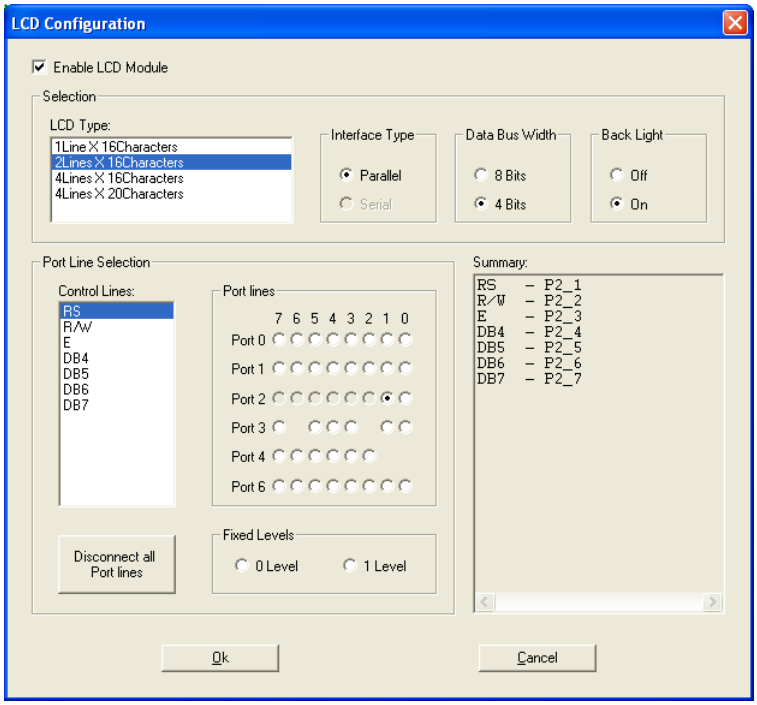
Use Topview Simulator to Verify the Design.

Open the project Demo22 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo22.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

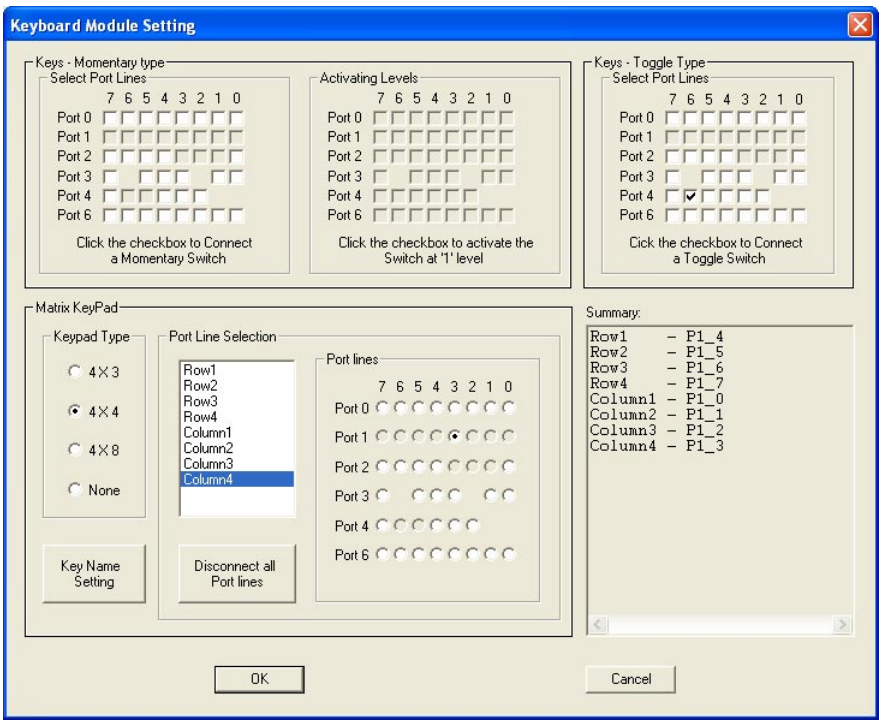
Use **Project** -> **Download Project** from main menu to download the .mot file into the simulator's memory for simulation.



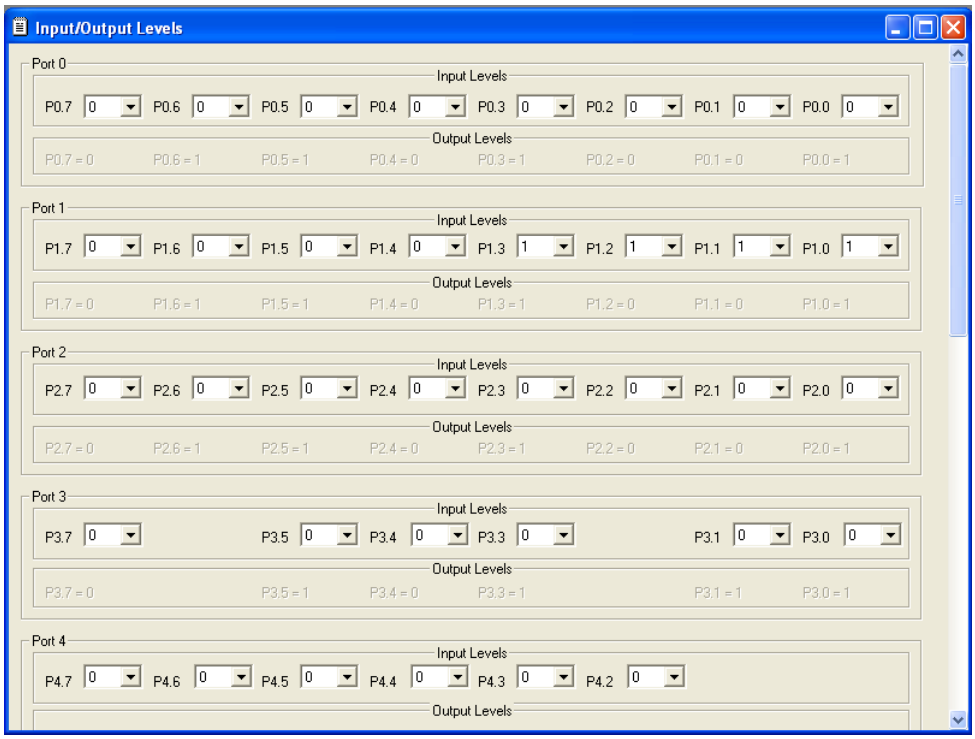
Do the settings to the LCD modules as shown. Connect LCD control and data lines to port lines P21 to P27.



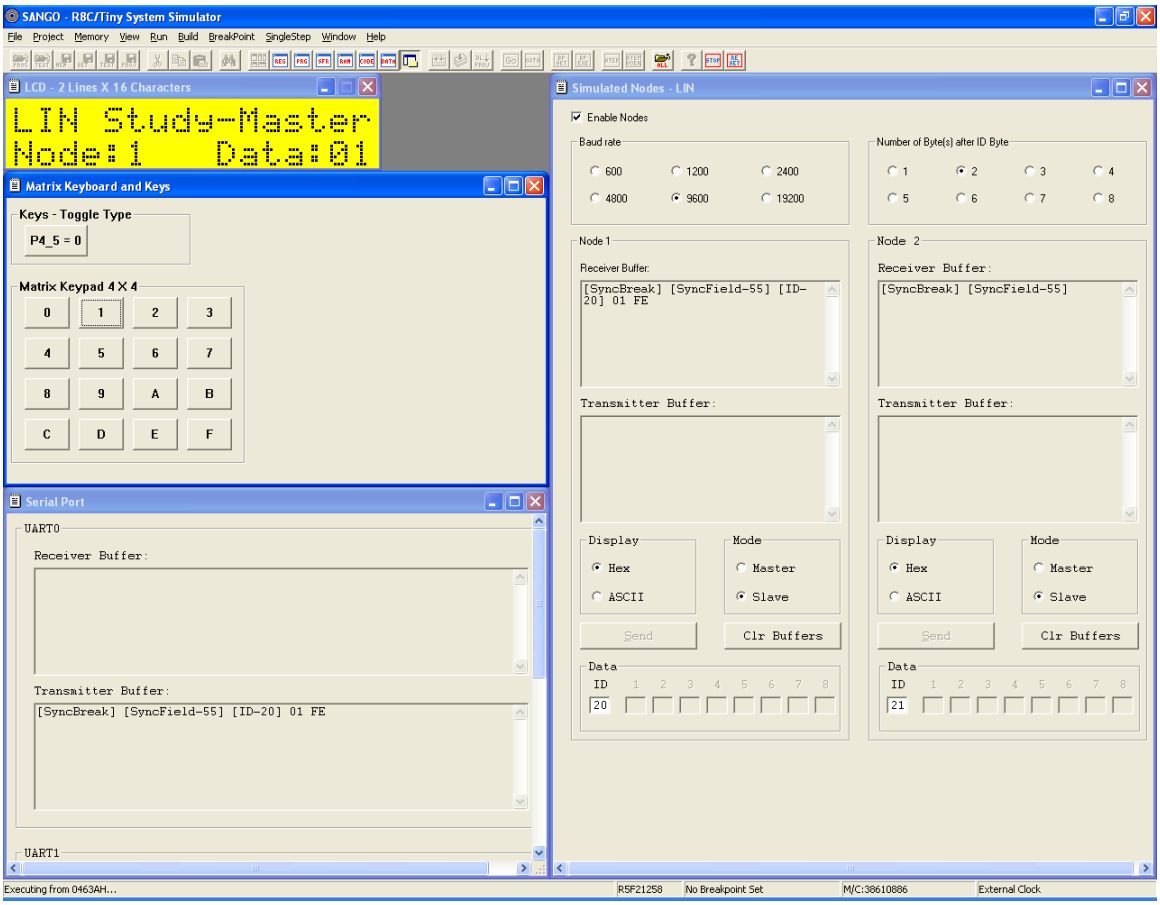
Make the setting for keyboard module as shown below:



Open the I/O window and set the input levels of P10 to P13 to 1.



Then open the LCD, Keyboard, Serial port and LIN Nodes windows and arranage them as shown below.



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

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

"**LIN Study-Master**

Node:1 Data: "

will be displayed in LCD.

Now press any key in the 4X4 matrix keyboard and the key value is sent to the selected node through LIN.

Format of LIN message:

[SyncBreak] [SyncField-55] [ID-20] 01 FE

Here "01" is Key value and "FE" is the checksum.