

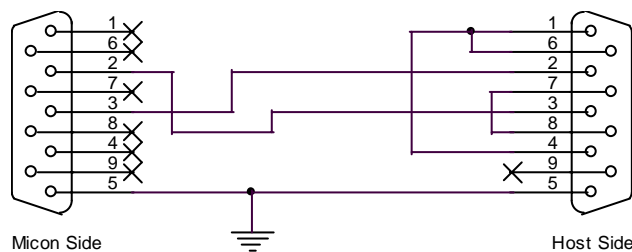
Introduction:

This demo program gives an idea about receiving data from the host using the serial port in asynchronous mode at 115200 baud using serial port 0. The received data will be stored in the RAM area of the micon. The starting address and number of bytes are also sent from host along with data.

Hardware:

The serial port lines RXD0 and TXD0 are level shifted using the RS232 level shifter device MAX232 and terminated in a 9 pin DIN connector. The host serial port lines and the micon's serial port lines connected using a serial port cable.

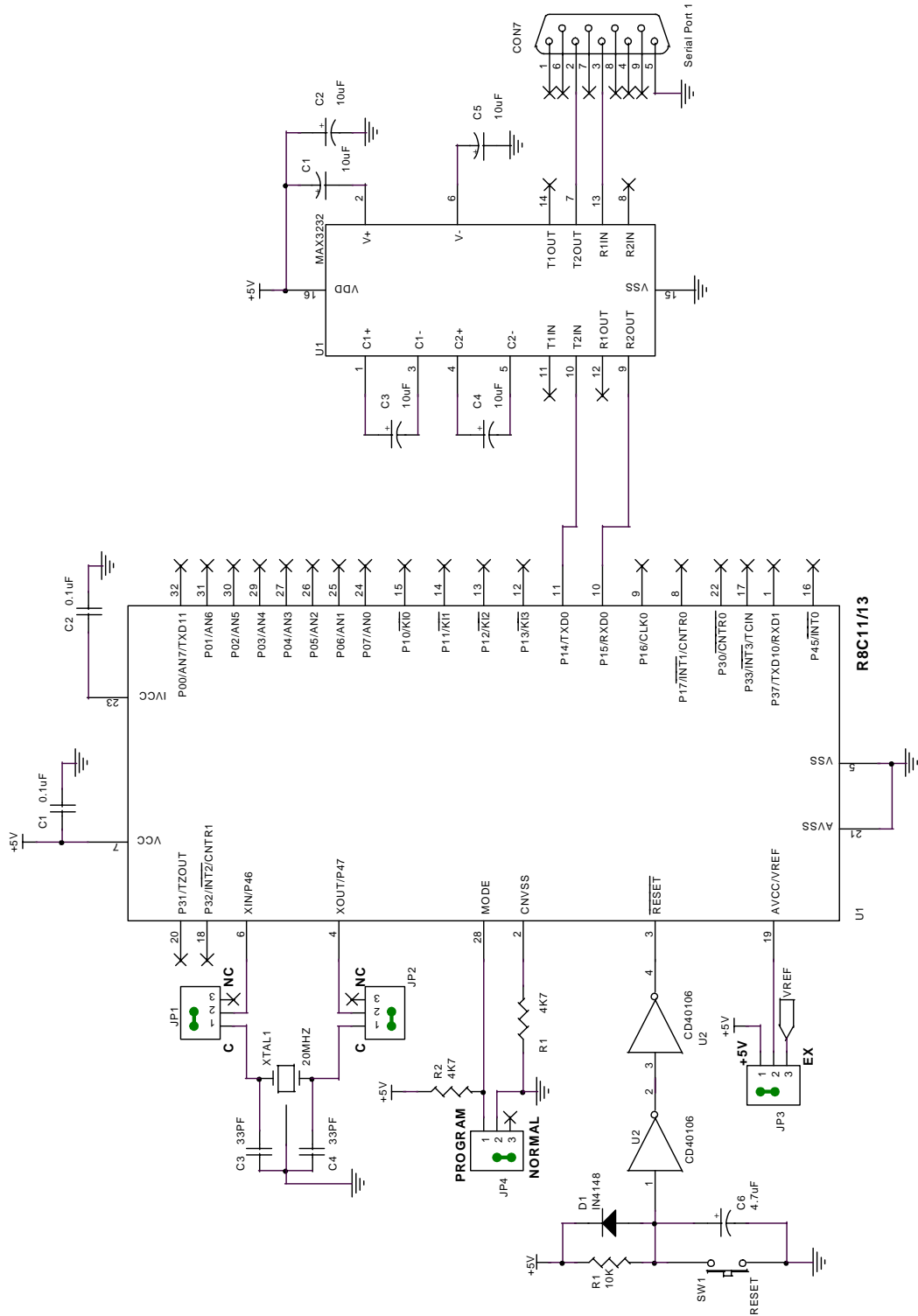
For minimum configuration, three lines are to be connected between the host and the micon namely RXD, TXD and Ground. The ground of micon circuit is connected directly to the host ground. The RXD line of micon is connected to the TXD line of host and the TXD line of micon is connected to the RXD line of host as shown below:



The pins 1, 6, 4, 7 & 8 of the 9 pin DIN connector at host are connected as shown to loop the hardware handshaking signals.

Demo 17 - Receiving data from Host through Serial Port

Circuit:



Functional Description:

In asynchronous mode, each transmitted or received character begins with a start bit and ends with one or two stop bits. Serial communication is synchronized one character at a time. The transmitting and receiving sections of the SCI are independent, so full-duplex communication is possible. The transmitter and the receiver are both double-buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

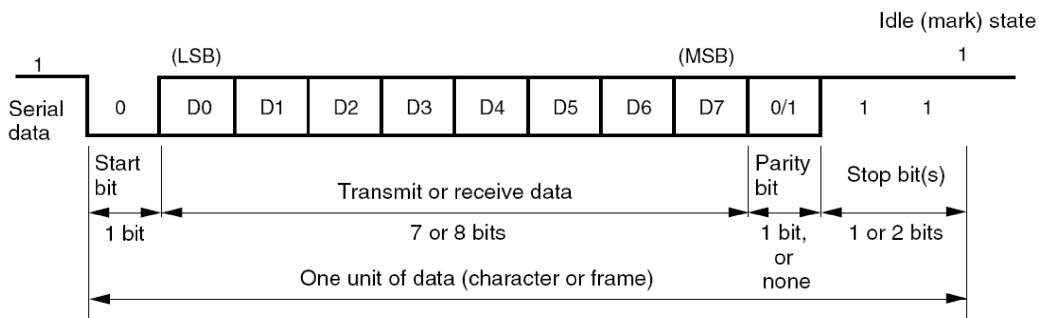


Figure shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and one or two stop bits (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

Registers Used:

- U0MR - UART0 transmit/receive mode register
- U0C0 - UART0 transmit/receive control register 0
- U0C1 - UART0 transmit/receive control register 1

U0MR - UART0 Transmit/Receive Mode Register:

Bit	Symbol	Address	After reset
b7	U0MR	00A016	0016
b6	U1MR	00A816	0016

Bit symbol	Bit name	Function	RW
SMD0	Serial interface mode select bit ²	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 Do not set except above	RW
SMD1			RW
SMD2			RW
CKDIR	Internal/external clock select bit ³	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	Effective 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

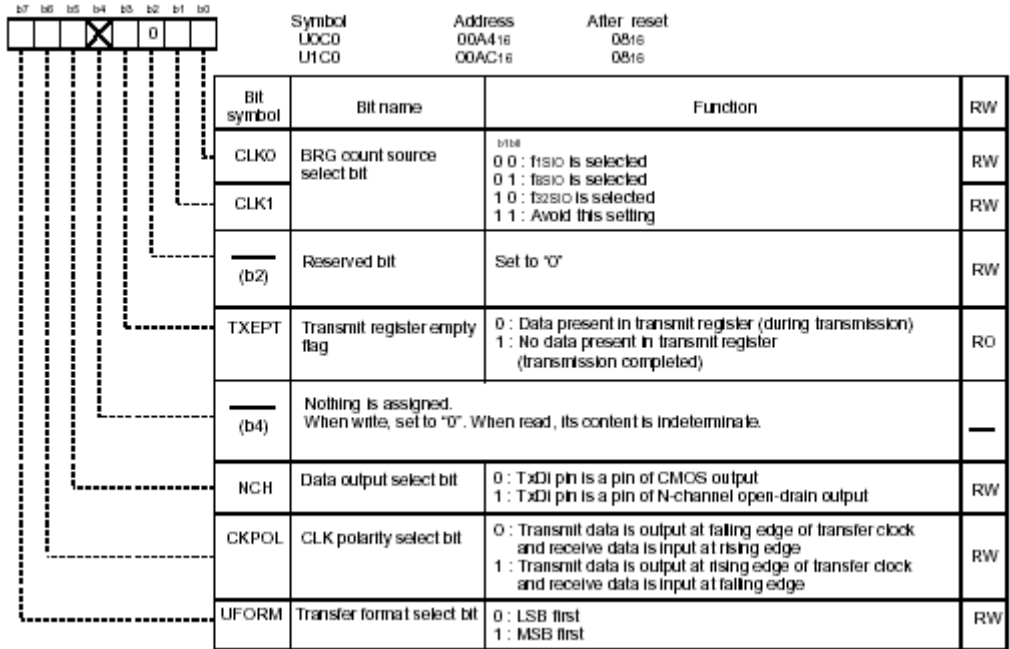
Notes:

1. Must set the P1_6 bit in the PD1 register to "0" (input).
2. For the U1MR register, the SMD2 to SMD0 bits must not be set except the followings: "000", "100", "101", or "110".
3. Must set the CKDIR bit to "0" (internal clock) in UART1.

U1MR register is initialized with the data H'05 to select following options:

1. UART mode transfer data 8 bits long,
2. Internal clock,
3. 1 Stop bit,
4. No parity.

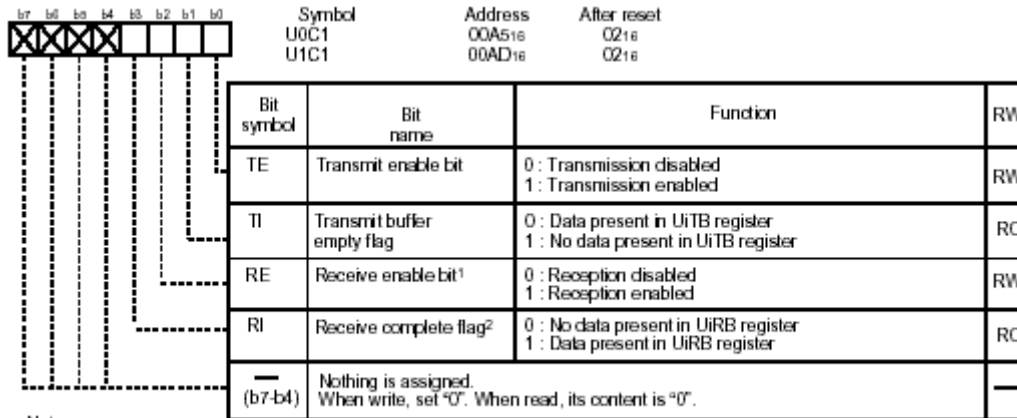
U0C0 - UART0 Transmit/Receive Control Register 0:



Data H'00 is set to register U0C0 register to select following options:

1. f1SIO clock is selected,
2. TXD1 pin is a pin of CMOS,
3. Transmit data is output at falling edge of transfer clock and receive data is input at rising edge,
4. LSB First.

U0C1 - UART1 Transmit/Receive Control Register 1:



Notes:
 1. As for the UART1, set the TXD1EN bit in the UCON register before setting this bit to reception enabled.
 2. The RI bit is set to "0" when the higher byte of the UiRB register is read.

Bit RE is set 1 to enable reception.

Software Description:

The data received from host through serial port is stored in RAM area of micon. The serial port 0 is used in standard 8 bit UART mode with no parity and one stop bit. The serial port 0 is configured to read data at 115200 baud rate @ 20MHZ.

After reset,

1. The external crystal oscillator is selected as clock source for MCU and other peripherals.
2. The serial port 0 is initialized in 8 bit mode with one stop bit and no parity configuration at 115200 baud rate @ 20MHz.

After the initialization, the control will read two bytes of data from host for address of RAM area. Then the number of bytes is read from host as one byte data.

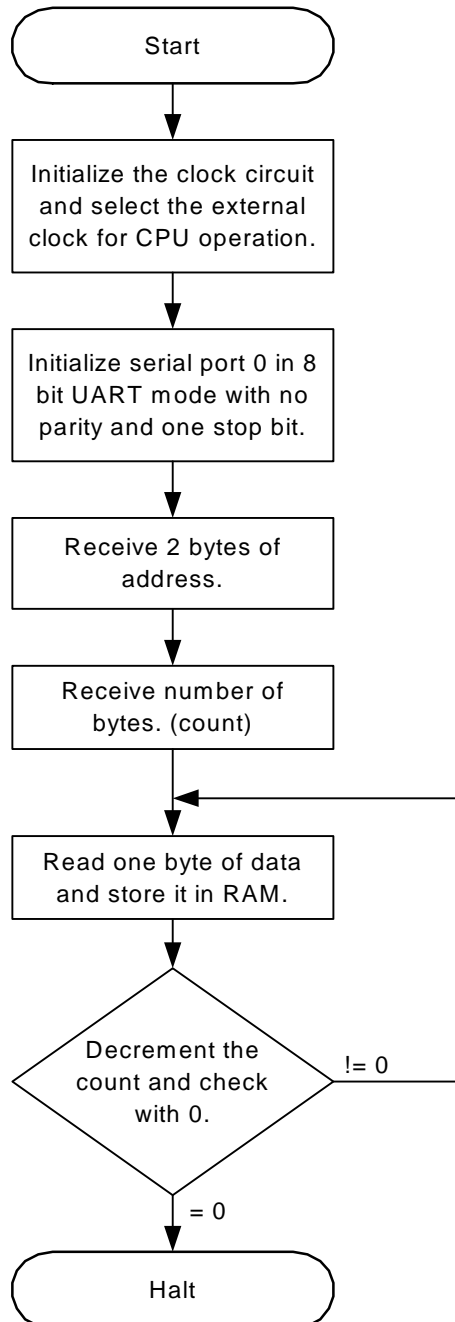
Finally the actual data are read from host through serial port 0 and stored in RAM area.

Demo 17 - Receiving data from Host through Serial Port

The functions in the file "Demo17.C" and short descriptions are listed below:

<i>Functions</i>	<i>Description</i>
main	Initializes the serial port 0 and sends the programmed message through serial port0. Input: None. Output : None.
ClockInitialization	Selects the external crystal oscillator as clock source for the CPU and other peripherals. Input: None. Output : None.
InitializeSerialPortChannel0	Initializes the serial port 0 in 8 bit mode at 9600 baud rate. Input: None. Output : None.
ReadByteSerialPort0	Sends the given message to host through serial port 0. Input: Message. Output : None.

Program Flow:



Execute Demo:

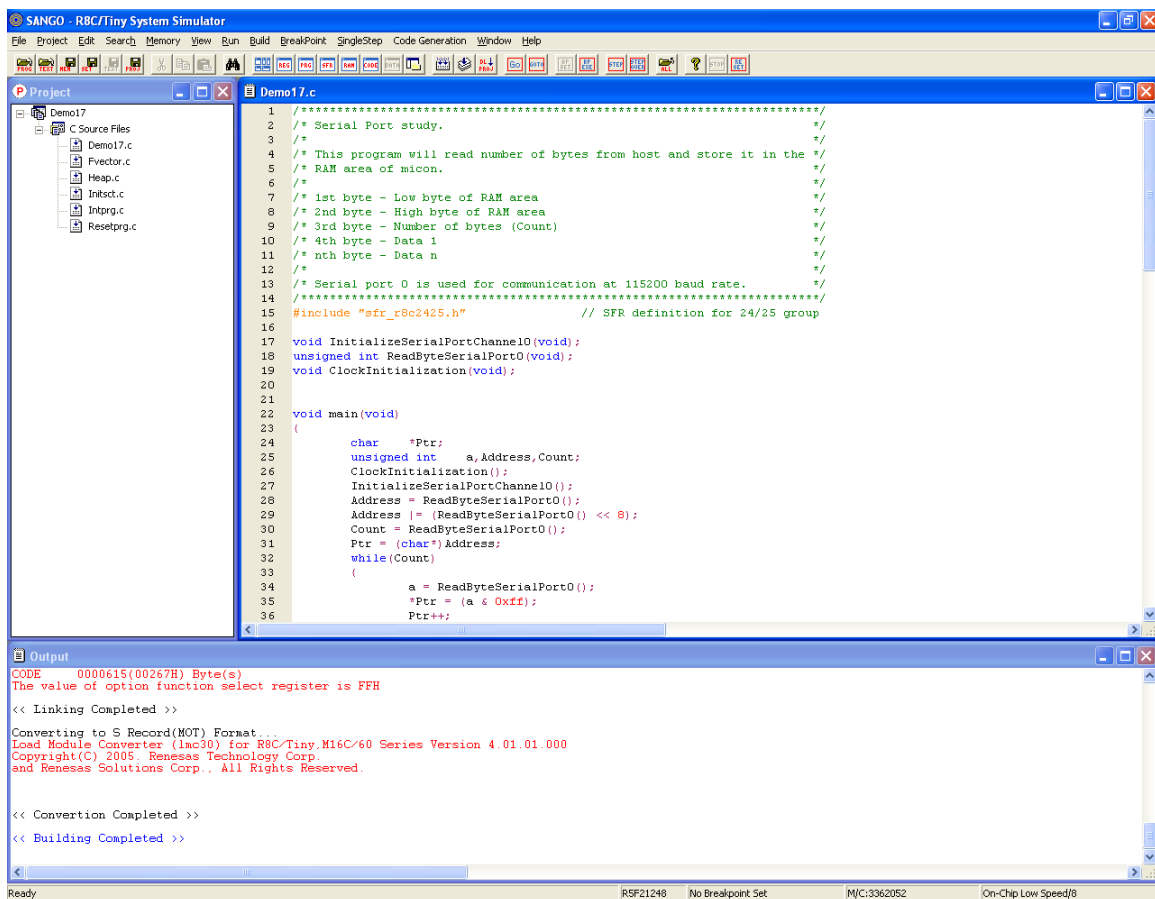
After reset, the data sent from the host are received through serial port 0 and stored in RAM area.

Demo 17 - Receiving data from Host through Serial Port

Use Topview Simulator to Verify the Design.

Open the project Demo17 in the R8C/Tiny System Simulator using **Open Project** option from **Project** menu. The project window opens up along with the Demo17.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.



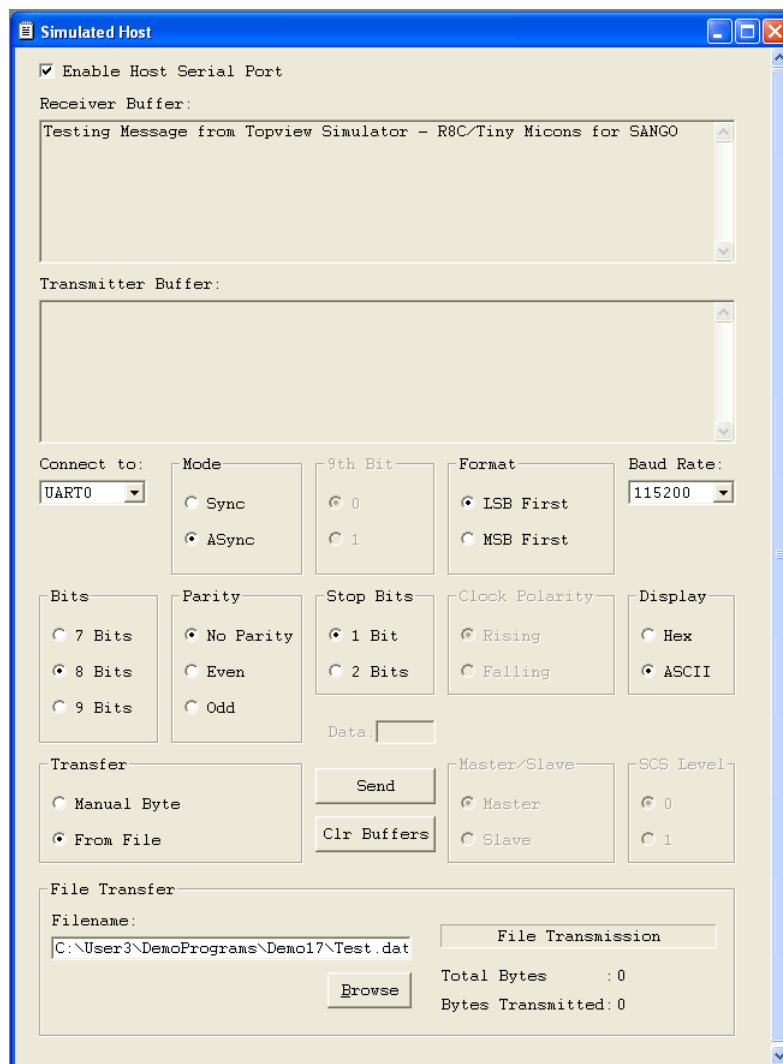
```
1  /******  
2  /* Serial Port study.  
3  /*  
4  /* This program will read number of bytes from host and store it in the  
5  /* RAM area of micon.  
6  /*  
7  /* 1st byte - Low byte of RAM area  
8  /* 2nd byte - High byte of RAM area  
9  /* 3rd byte - Number of bytes (Count)  
10 /* 4th byte - Data 1  
11 /* nth byte - Data n  
12 /*  
13 /* Serial port 0 is used for communication at 115200 baud rate.  
14 /******  
15 #include "sfr_r8c2425.h" // SFR definition for 24/25 group  
16  
17 void InitializeSerialPortChannel0(void);  
18 unsigned int ReadByteSerialPort0(void);  
19 void ClockInitialization(void);  
20  
21  
22 void main(void)  
23 {  
24     char *Ptr;  
25     unsigned int a, Address, Count;  
26     ClockInitialization();  
27     InitializeSerialPortChannel0();  
28     Address = ReadByteSerialPort0();  
29     Address |= (ReadByteSerialPort0() << 8);  
30     Count = ReadByteSerialPort0();  
31     Ptr = (char*)Address;  
32     while(Count)  
33     {  
34         a = ReadByteSerialPort0();  
35         *Ptr = (a & 0xFF);  
36         Ptr++;
```

Output
CODE 0000615(00267H) 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.
<< Conversion Completed >>
<< Building Completed >>

Ready RSP21248 No Breakpoint Set M/C:3362052 On-Chip Low Speed/8

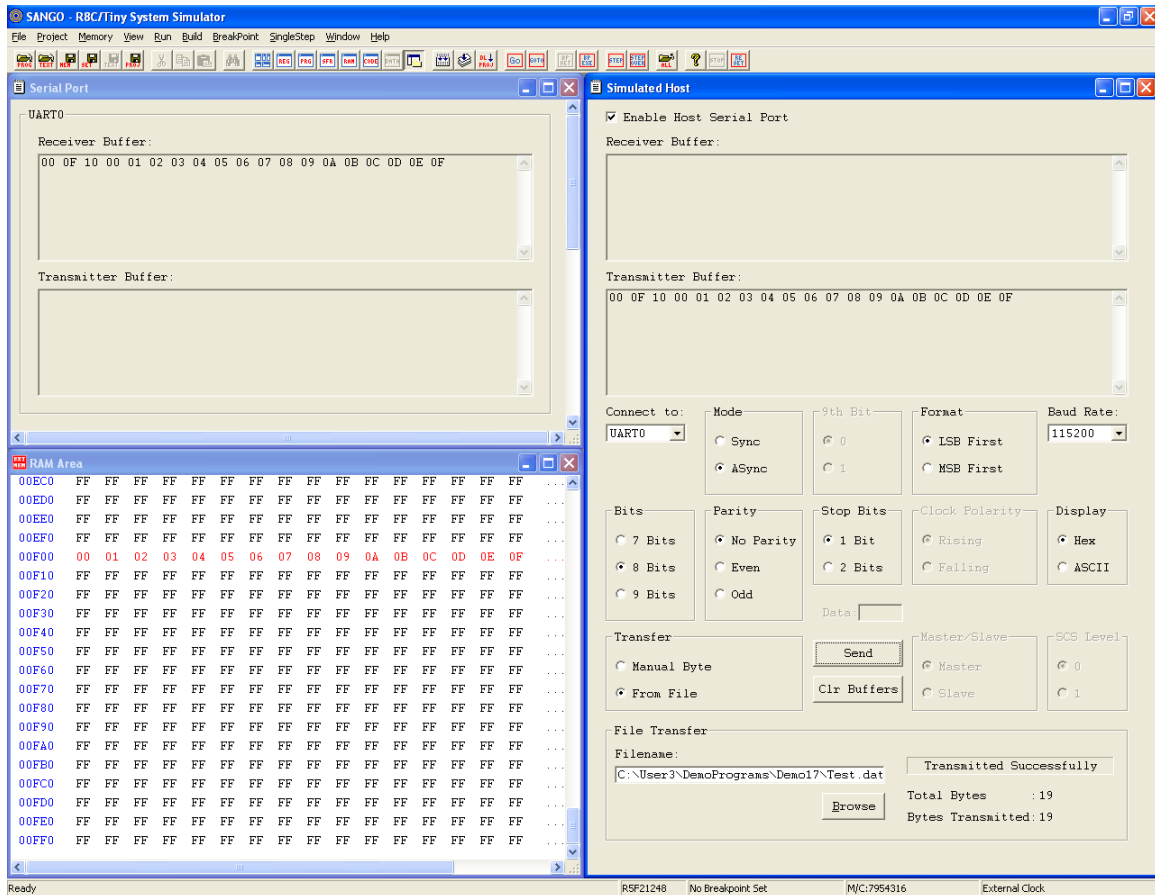
Open the simulated host window make the setting as shown below:

Demo 17 - Receiving data from Host through Serial Port



Open Serial port, Simulated host and RAM area windows and arrange them as shown below.

Demo 17 - Receiving data from Host through Serial Port



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

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

In this demo program, a group of data is sent from the simulated host window to micon's serial port 0. The data sent from the host received in micon and stored in the RAM area.

The group of data is already created as as data file in the name of "test.dat" and the data from this file is sent from host.

Now press the **Send** button in host window, this will read the data from the given file and sends the data to serial port 0.