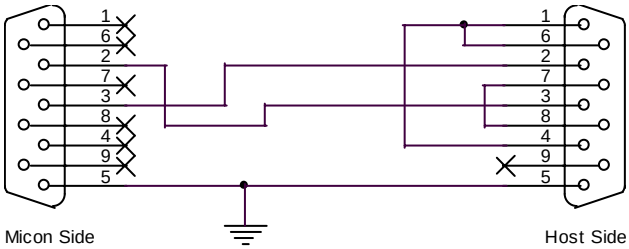
Introduction:

Gives an idea about sending a message to the host using the serial port in asynchronous mode at 9600 baud using serial port 0.

Hardware:

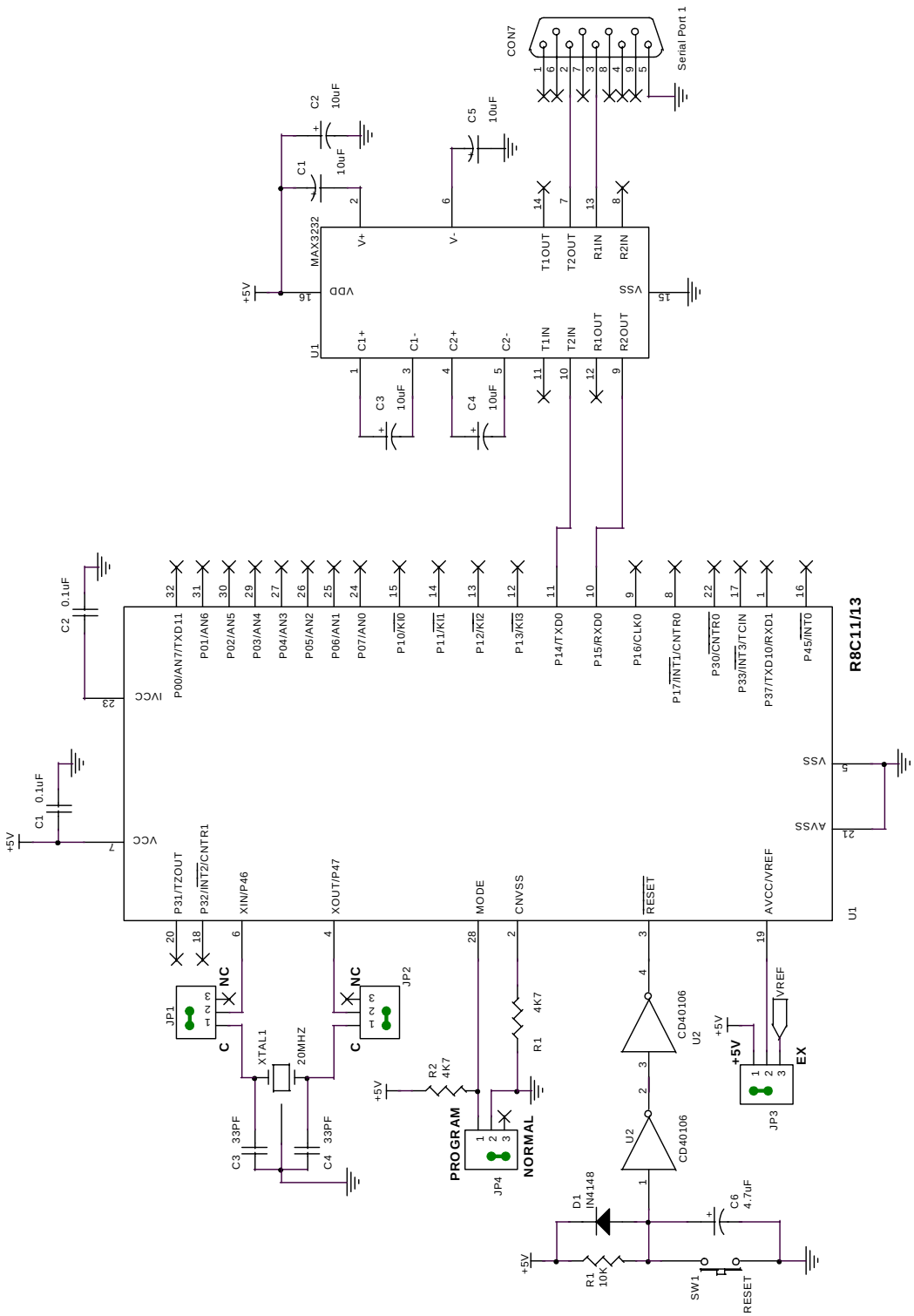
The serial port lines RXD0 and TXD0 are level shifted using the RS232 level shifter device MAX232 and terminated in a 9 pin DIN connector. The host serial port lines and the micon's serial port lines connected using a serial port cable.

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.

Circuit:


CON7
Serial Port 1
MAX3232
U1
C1
C2
C3
C4
C5
10uF
+5V
T1OUT
T2OUT
R1IN
R2IN
T1IN
T2IN
R1OUT
R2OUT
C1+
C1-
C2+
C2-
V+
V-
VDD
VSS
R8C11/13
U1
P00/AN7/TXD11
P01/AN6
P02/AN5
P03/AN4
P04/AN3
P05/AN2
P06/AN1
P07/AN0
P10/KI0
P11/KI1
P12/KI2
P13/KI3
P14/TXD0
P15/RXD0
P16/CLK0
P17/INT1/CNTR0
P30/CNTR0
P33/INT3/TCIN
P37/TXD10/RXD1
P45/INT0
P31/TZOUT
P32/INT2/CNTR1
XIN/P46
XOUT/P47
MODE
CNVSS
RESET
AVCC/VREF
VCC
AVSS
0.1uF
JP1
JP2
JP3
JP4
NC
XTAL1
20MHZ
33PF
4K7
R1
R2
PROGRAM
NORMAL
CD40106
U2
VREF
EX
D1
IN4148
C6
4.7uF
10K
SW1
RESET

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.

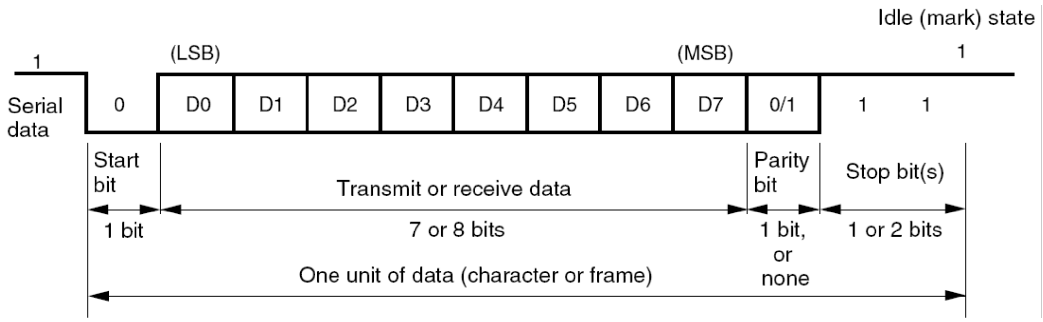


Figure shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One 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:

Symbol	Address	After reset
U0MR	00A0₁₆	00₁₆
U1MR	00A8₁₆	00₁₆

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 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: "000₂", "100₂", "101₂", or "110₂".
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:

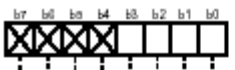
Symbol	Address	After reset
U0C0	00A4₁₆	08₁₆
U1C0	00AC₁₆	08₁₆

Bit symbol	Bit name	Function	RW
CLK0	BRG count source select bit	b1 b0 0 0 : f1SIO is selected 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Avoid this setting	RW
CLK1			RW
— (b2)	Reserved bit	Set to "0"	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
— (b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
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 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:



Symbol	Address	After reset
U0C1	00A5₁₆	02₁₆
U1C1	00AD₁₆	02₁₆

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UiTB register 1 : No data present in UiTB register	RO
RE	Receive enable bit[1]	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete flag[2]	0 : No data present in UiRB register 1 : Data present in UiRB register	RO
— (b7-b4)	Nothing is assigned. When write, set "0". When read, its content is "0".		—

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.

Bits TE and TI are set 1 to enable transmission.

Software Description:

The programmed message is sent to the host through serial port 0 in standard 8 bit UART mode with no parity and one stop bit. The serial port 0 is configured to send data at 9600 baud rate @ 20MHZ.

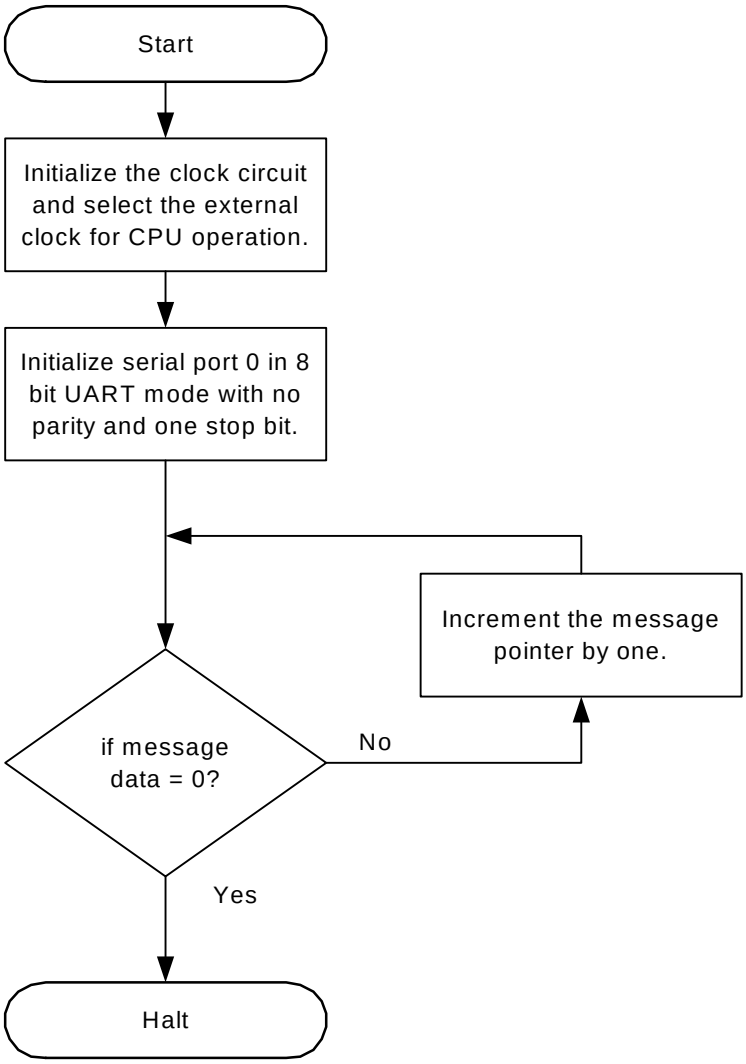
After reset,

1. The external crystal oscillator is selected as clock source for MCU and other peripherals.
2. The serial port 0 is initialized in 8 bit mode with one stop bit and no parity configuration at 9600 baud rate @ 20MHz.
3. The Message "**Testing Message from Topview Simulator - R8C/Tiny Micons for SANGO**" is sent to the host through serial port 0.

The functions in the file “Demo16.C” and short descriptions are listed below:

Functions	*Description*
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.
SendMessage	Sends the given message to host through serial port 0. **Input**: Message. **Output** : None.

Program Flow:



Execute Demo:

After reset, the message,

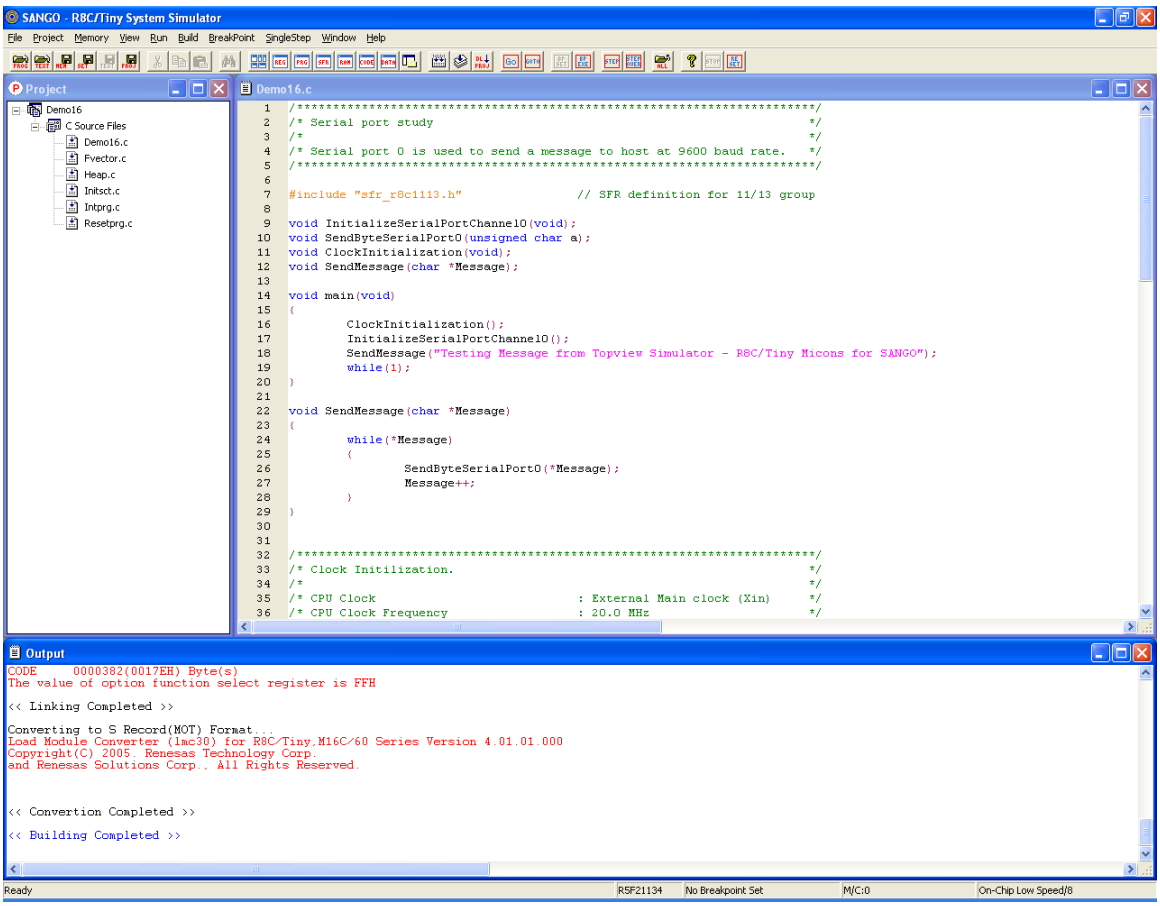
"**Testing Message from Topview Simulator - R8C/Tiny Micons for SANGO**"

is sent through serial port 0.

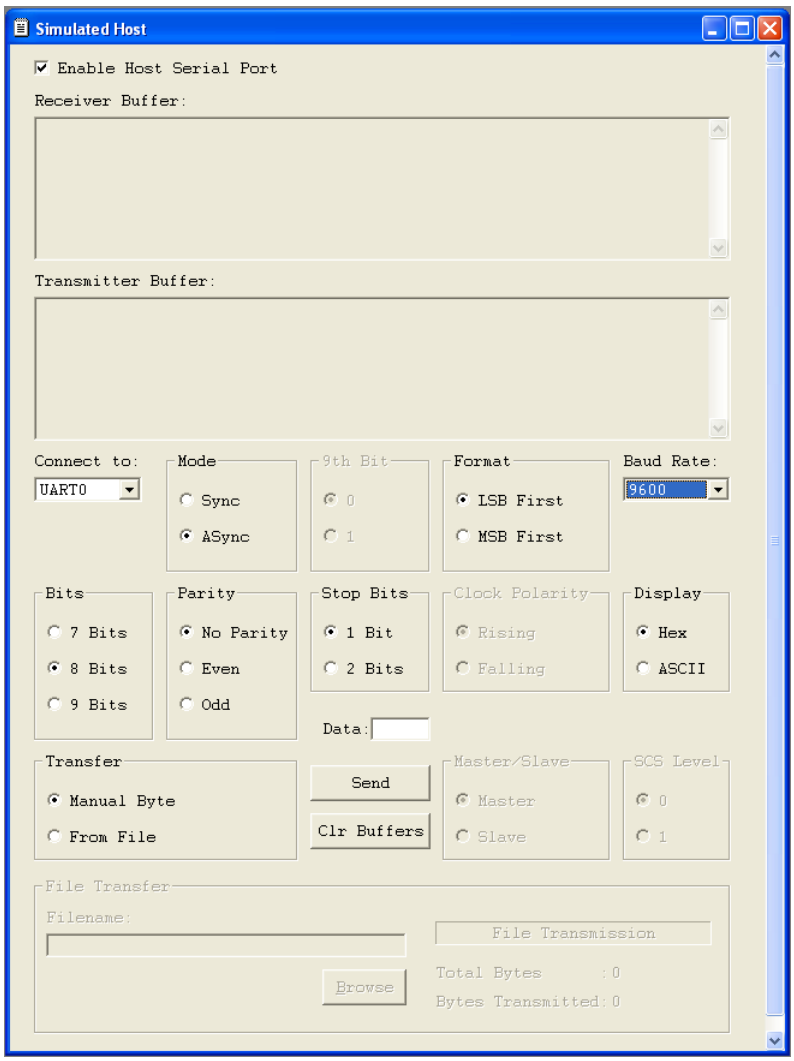
Use Topview Simulator to Verify the Design.

Open the project Demo16 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo16.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

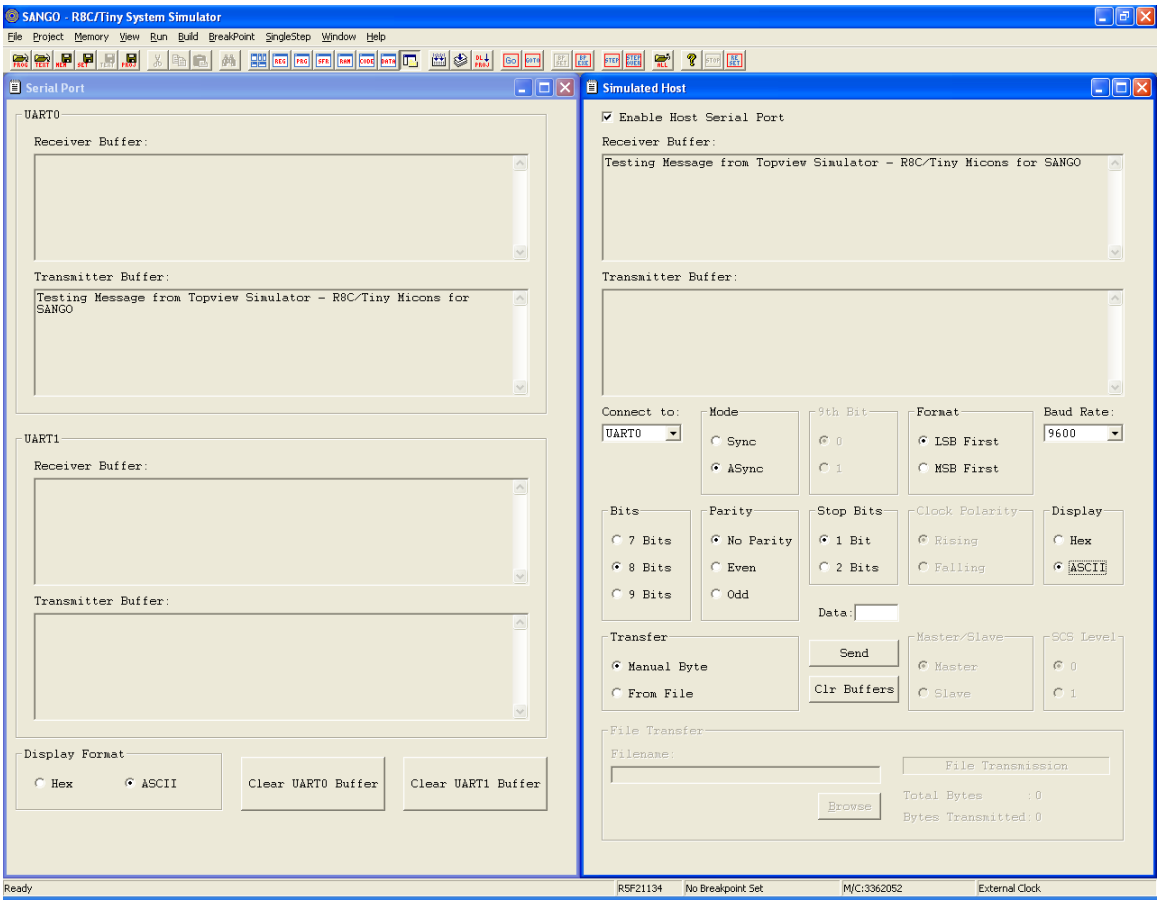
Use **Project** -> **Download Project** from main menu to download the .mot file into the simulator's memory for simulation.



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



Open Serial port and Simulated host 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,

"**`Testing Message from Topview Simulator - R8C/Tiny Micons for SANGO`**"

will be sent through serial port 0.