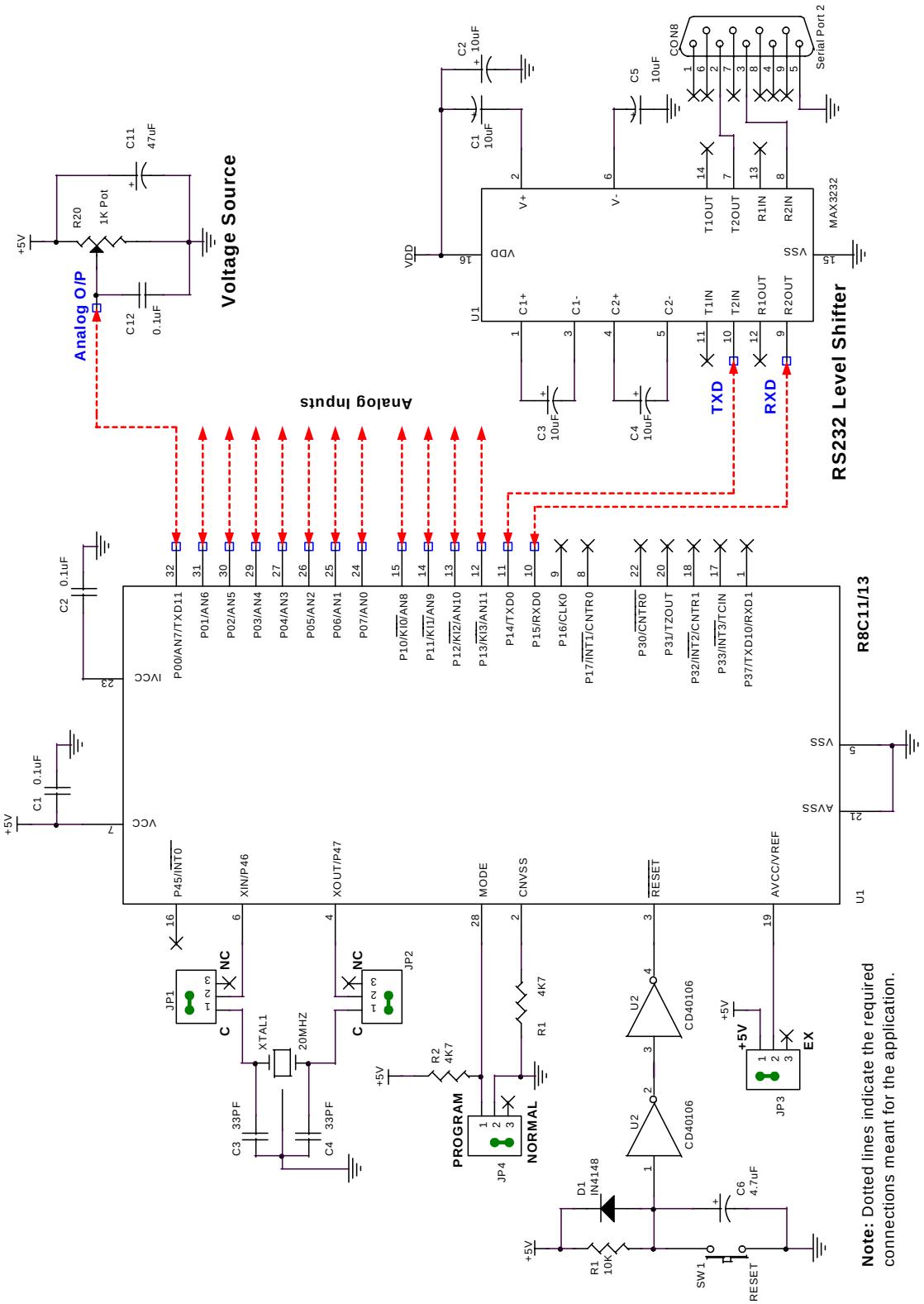
Introduction:

In this demo, the program will read all 12 channels of ADC and send the data through serial port. Serial port 0 is used for sending data to PC at 9600 baudrate. The conversion is started as soon as the character 'S' or 's' is received from PC through serial port.

Hardware:

Serial port 0 is used to send the ADC data to PC. The Vref is given externally as +5V.

Circuit:


Voltage Source
Analog O/P
R20
1K Pot
C11
47uF
C12
0.1uF
+5V
Analog Inputs
RS232 Level Shifter
MAX3232
U1
C1
C2
C3
C4
C5
10uF
CON8
Serial Port 2
VDD
VSS
V+
V-
T1OUT
T2OUT
R1IN
R2IN
C1+
C1-
C2+
C2-
T1IN
T2IN
R1OUT
R2OUT
TXD
RXD
R8C11/13
P00/AN7/TXD11
P01/AN6
P02/AN5
P03/AN4
P04/AN3
P05/AN2
P06/AN1
P07/AN0
P10/KI0/AN8
P11/KI1/AN9
P12/KI2/AN10
P13/KI3/AN11
P14/TXD0
P15/RXD0
P16/CLK0
P17/INT1/CNTR0
P30/CNTR0
P31/TZOUT
P32/INT2/CNTR1
P33/INT3/TCIN
P37/TXD10/RXD1
P45/INT0
XIN/P46
XOUT/P47
MODE
CNVSS
RESET
AVCC/VREF
AVSS
VCC
JP1
JP2
JP3
JP4
NC
C
XTAL1
20MHZ
33PF
R1
4K7
R2
4K7
PROGRAM
NORMAL
EX
U2
CD40106
D1
IN4148
R1
10K
C6
4.7uF
SW1
RESET
Note: Dotted lines indicate the required connections meant for the application.

Connections:

1. Connect the analog volatge to ADC inputs.
2. Connect +5V to Vref pin.

Functional Description:

All the 12 channels are sampled one after another and the sampled data are send through serial port 0 at 9600 baud rate to PC.

Insteat of PC user can use simulated host to read and display the data sent from R8C/Tiny micon through serial port 0 at 9600 baudrate. The conversion is started as soon as the character 'S' or 's' is received from PC through serial port.

Registers Used:

ADCON0 - AD Control register 0
ADCON1 - AD Control register 1
ADCON2 - AD Control register 2
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

ADCON0 - AD Control Register 0:

Symbol: ADCON0 | Address: 00D6₁₆ | After reset: 00000XXX₂

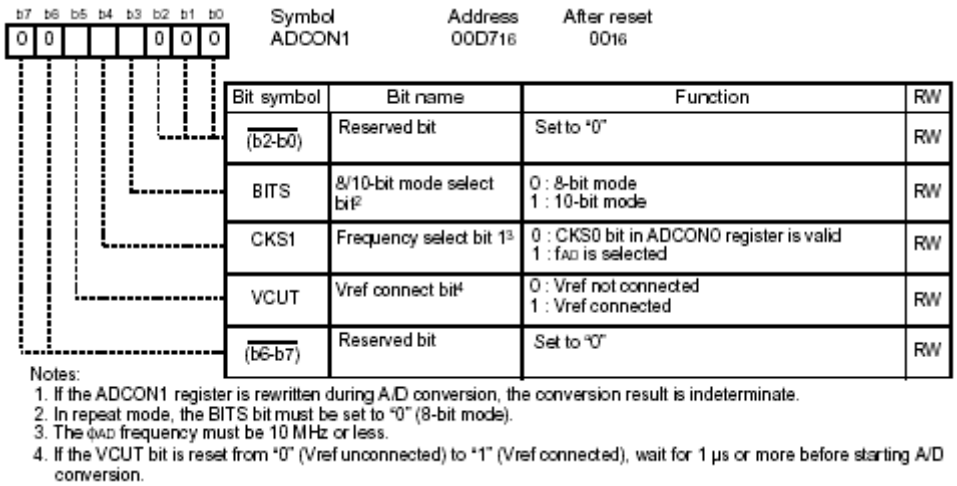
Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit	See Note 4.	RW
CH1			RW
CH2			RW
MD	AD operation mode select bit[2]	0 : One-shot mode 1 : Repeat mode	RW
ADGSEL0	AD input group select bit[4]	0: Port P0 group selected (AN0 to AN7) 1: Port P1 group selected (AN8 to AN11)	RW
(b6)	Reserved bit	Set to "0"	RW
ADST	A/D conversion start flag	0 : A/D conversion disabled 1 : A/D conversion started	RW
CKS0	Frequency select bit 0[3]	0 : fAD/4 is selected 1 : fAD/2 is selected	RW

Notes:
1. If the ADCON register is rewritten during A/D conversion, the conversion result is indeterminate.
2. When changing A/D operation mode, set analog input pin again.
3. This bit is valid when the CKS1 bit in the ADCON1 register is set to "0".
4. The analog input pin can be selected by a combination of the CH2 to CH0 bits and ADGSEL0 bit as follows:

CH2 to CH0	ADGSEL0=0	ADGSEL0=1
000₂	AN0	Avoid these settings
001₂	AN1	
010₂	AN2	
011₂	AN3	
100₂	AN4	AN8
101₂	AN5	AN9
110₂	AN6	AN10
111₂	AN7	AN11

The register ADCON0 is loaded with the data H'01 to select one shot mode and clock as fad/4.

ADCON1 - AD Control Register 1:



b7	b6	b5	b4	b3	b2	b1	b0
0	0				0	0	0

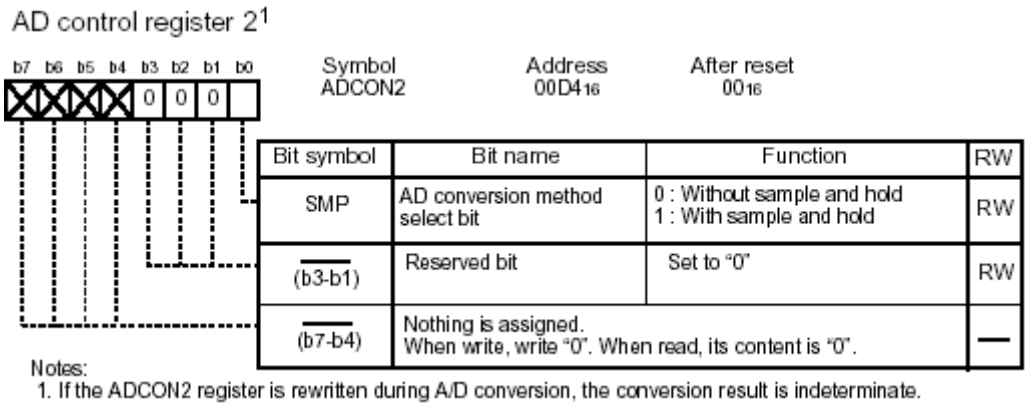
Symbol: ADCON1 — Address: 00D7₁₆ — After reset: 00₁₆

Bit symbol	Bit name	Function	RW
$\overline{(b2\text{-}b0)}$	Reserved bit	Set to "0"	RW
BITS	8/10-bit mode select bit[2]	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit 1[3]	0 : CKS0 bit in ADCON0 register is valid 1 : f_{AD} is selected	RW
VCUT	Vref connect bit[4]	0 : Vref not connected 1 : Vref connected	RW
$\overline{(b6\text{-}b7)}$	Reserved bit	Set to "0"	RW

Notes:
1. If the ADCON1 register is rewritten during A/D conversion, the conversion result is indeterminate.
2. In repeat mode, the BITS bit must be set to "0" (8-bit mode).
3. The ϕ_{AD} frequency must be 10 MHz or less.
4. If the VCUT bit is reset from "0" (Vref unconnected) to "1" (Vref connected), wait for 1 µs or more before starting A/D conversion.

The data H'28 is moved to ADCON1 register to select 10 bit mode and external Vref.

ADCON2 - AD Control Register 2:



AD control register 2[1]

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	0	0	0	

Symbol: ADCON2 — Address: 00D4₁₆ — After reset: 00₁₆

Bit symbol	Bit name	Function	RW
SMP	AD conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
$\overline{(b3\text{-}b1)}$	Reserved bit	Set to "0"	RW
$\overline{(b7\text{-}b4)}$	Nothing is assigned. When write, write "0". When read, its content is "0".		—

Notes:
1. If the ADCON2 register is rewritten during A/D conversion, the conversion result is indeterminate.

The register ADCON2 is initialized with the data H'01 to enable sample and hold function.

U0MR - UART 0 Transmit/Receive Mode Register:

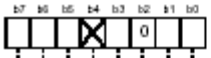
Symbol	Address	After reset
U0MR	00A0₁₆	00₁₆
U1MR	00A8₁₆	00₁₆

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.

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

U0C0 - UART 0 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 : f_{1SIO} is selected 0 1 : f_{8SIO} is selected 1 0 : f_{32SIO} 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

The register U0C0 is loaded with the data H'00 to select clock as f1SIO and LSB first.

U0C1 - UART 0 Transmit/Receive Control Register 1:

UARTi transmit/receive control register 1 (i=0, 1)

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

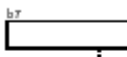
Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	RW
TI	Transmit buffer empty flag	0 : Data present in UITB register 1 : No data present in UITB register	RO
RE	Receive enable bit[1]	0 : Reception disabled 1 : Reception enabled	RW
RI	Receive complete 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.

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

U0BRG - UART 0 Bit Rate Register:

UARTi bit rate register[1, 2] (i=0, 1)



Symbol	Address	After reset
U0BRG	00A1₁₆	Indeterminate
U1BRG	00A9₁₆	Indeterminate

Function	Setting range	RW
Assuming that set value = n, UiBRG divides the count source by n + 1	00₁₆ to FF₁₆	WO

Notes:
1. Write to this register while serial interface is neither transmitting nor receiving.
2. Use MOV instruction to write to this register.

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

Software Description:

After reset, the serial port is initialized to 9600 baud rate and a message as shown below will be sent through serial port 0.

```
"              *** R8C/13 - ADC Demo ***
<<< Press 'S' or 's' to get data of 12 channels >>> "
```

The ADC is used in 10 bit mode with sample hold function. When a character 'S' or 's' is received from host through the serial port, all 12 channels are sampled one after another and the sampled data are stored in a array. After sampling all channels, the sampled data array is sent to PC through serial port 0 at 9600 baud rate. Format of messages sent from R8C/Tiny micon to PC:

```
*** R8C/13 - ADC Demo ***
Channel 00 - 100H
Channel 01 - 100H
Channel 02 - 100H
Channel 03 - 100H
Channel 04 - 100H
Channel 05 - 100H
Channel 06 - 100H
Channel 07 - 100H
Channel 08 - 100H
Channel 09 - 100H
Channel 10 - 100H
Channel 11 - 100H
```

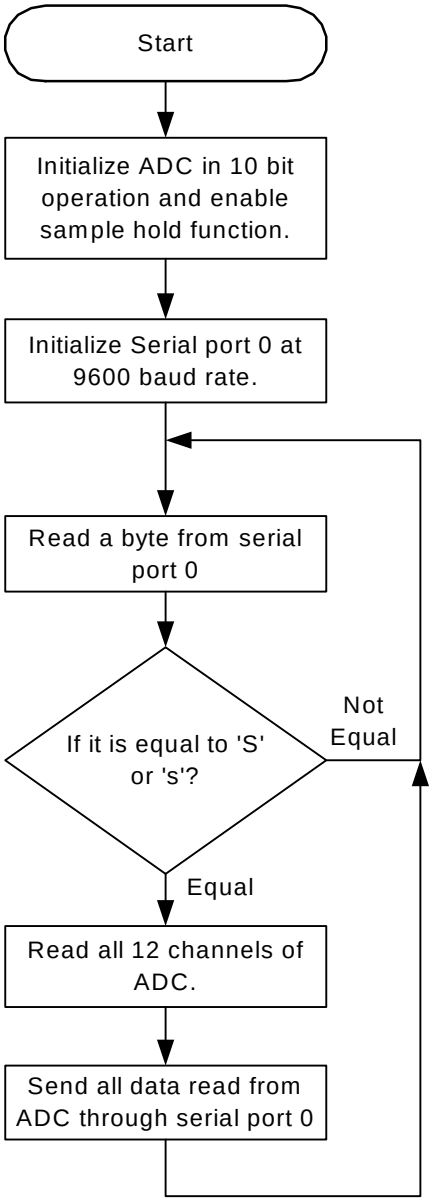
The files used in this module are listed below:

Files	*Description*
Demo20.C	Initializes the serial port 0 to 9600 baud rate and ADC in 10 bit mode with sample and hold function. Reads all 12 channels of ADC and sent the data to PC through the serial port 0.

The functions in the file “Demo20.C” and short descriptions are listed below:

Files	*Description*
Main	Sequentially read all 12 channels of ADC and send the sampled data to PC through serial port 0. **Input**: None. **Output** : None.
InitializeSequentialSamplingADC	Initializes the ADC in one shot mode with sample and hold function and 10 bit mode. Also initializes serial port at 9600 baud rate. **Input**: None. **Output** : None.
ReadADC	Reads the given channel and returns the 10 bit digital value. **Input**: Channel number. **Output** : Digital value.
InitializeSerialPort	Initializes Serial Port 0 with 9600 baud rate **Input**: None. **Output** : None.
SendMessage	Sends the given message through serial port 0. **Input**: Message to be sent. **Output** : None.
ReadAllChannels	Reads and stores all 12 channels of ADC **Input**: None. **Output** : None.

Program Flow:



Execute Application:

After reset, the program will initilaize serial port 0 at 9600 baud rate and ADC.The message

" ***** R8C/13 - ADC Demo *****

<<< Press 'S' or 's' to get data of 12 channels >>>"

is sent to PC through serial port 0.

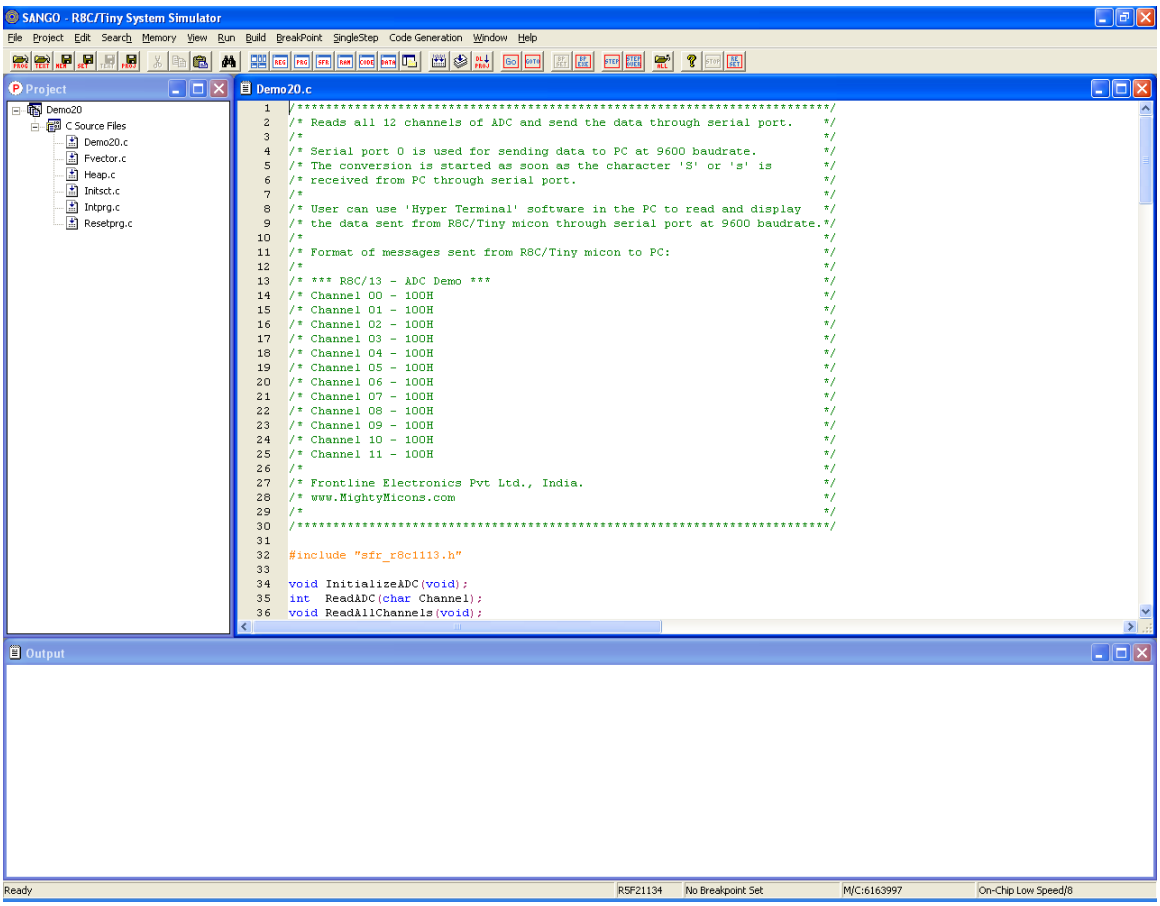
The control waits for ‘S’ or ‘s’ character from host. After getting the character ‘S’ or ‘s’ from host, the 12 channels of ADC are sampled one after another and the sampled data are sent through serial port 0 as shown below:

```
*** R8C/13 - ADC Demo ***
Channel 00 - 100H
Channel 01 - 100H
Channel 02 - 100H
Channel 03 - 100H
Channel 04 - 100H
Channel 05 - 100H
Channel 06 - 100H
Channel 07 - 100H
Channel 08 - 100H
Channel 09 - 100H
Channel 10 - 100H
Channel 11 - 100H
```

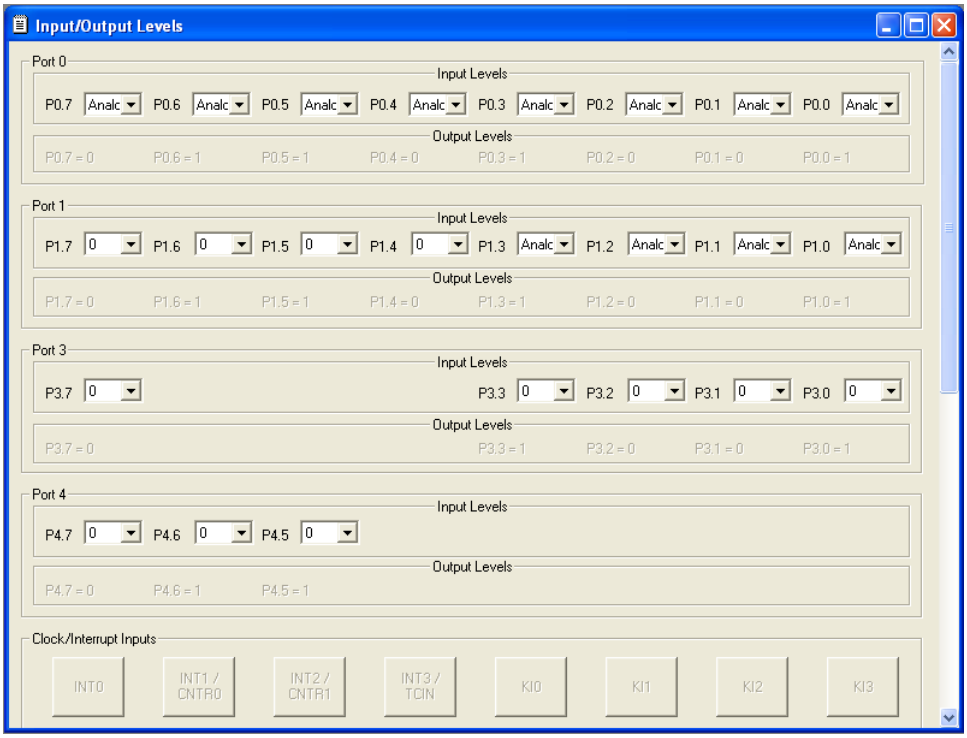
Use Topview Simulator to Verify the Design.

Open the project Demo20 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo20.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

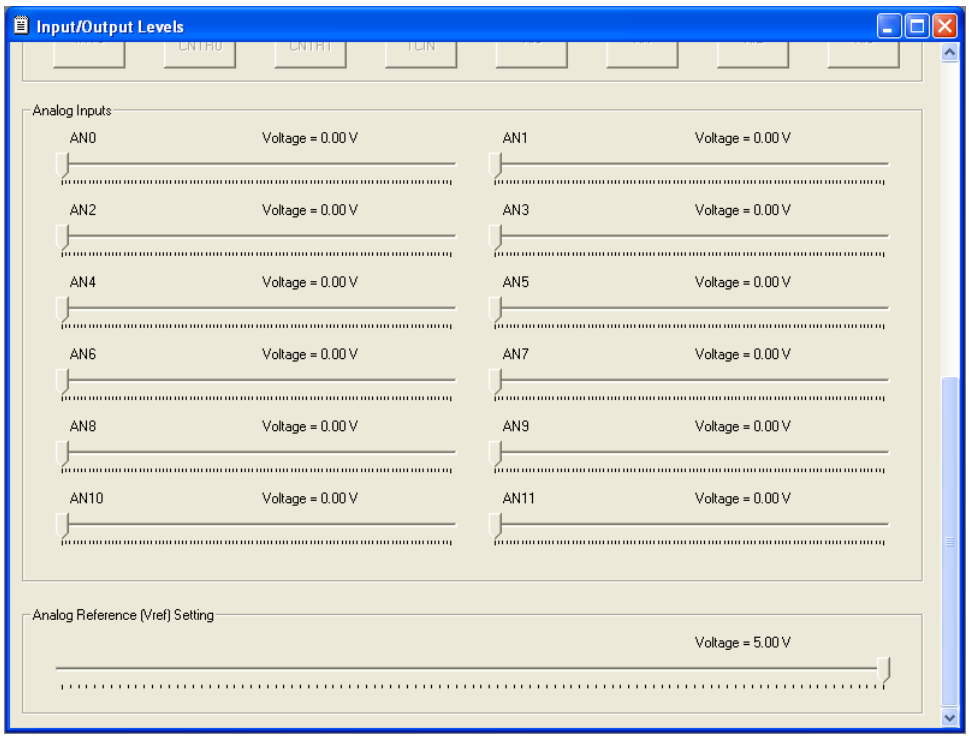
Use **Project** -> **Download Project** from main menu to download the .mot file into the simulator's memory for simulation.

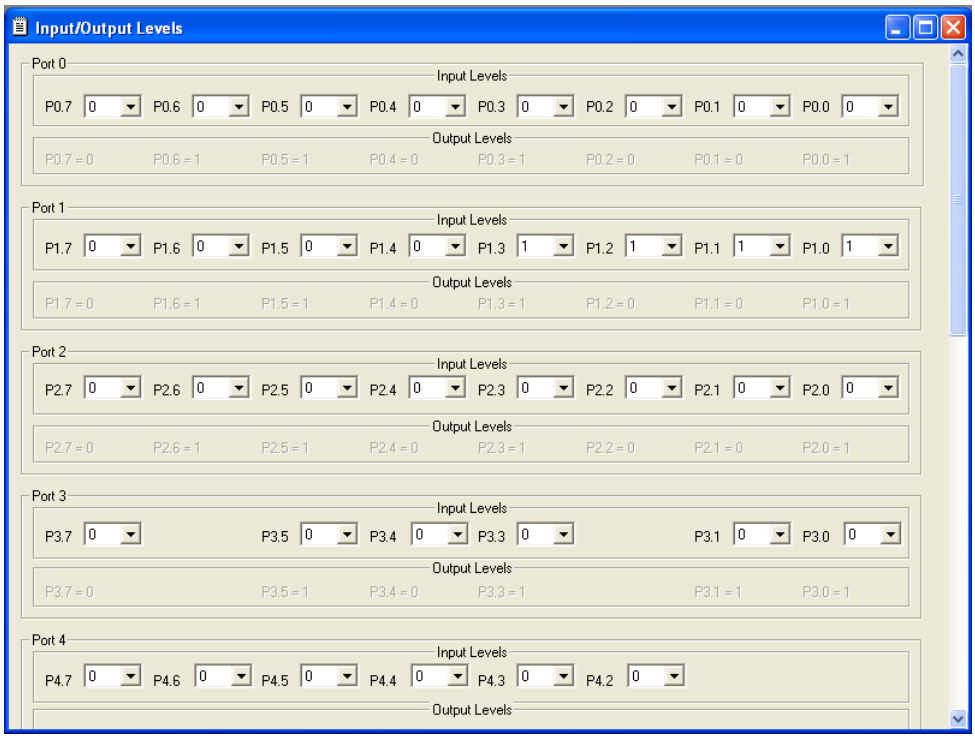


Open the I/O window and connect the variable analog voltage source to analog inputs by selecting the option "Analog" from the input levels of the portlines P00 to P07 and P10 to P13.

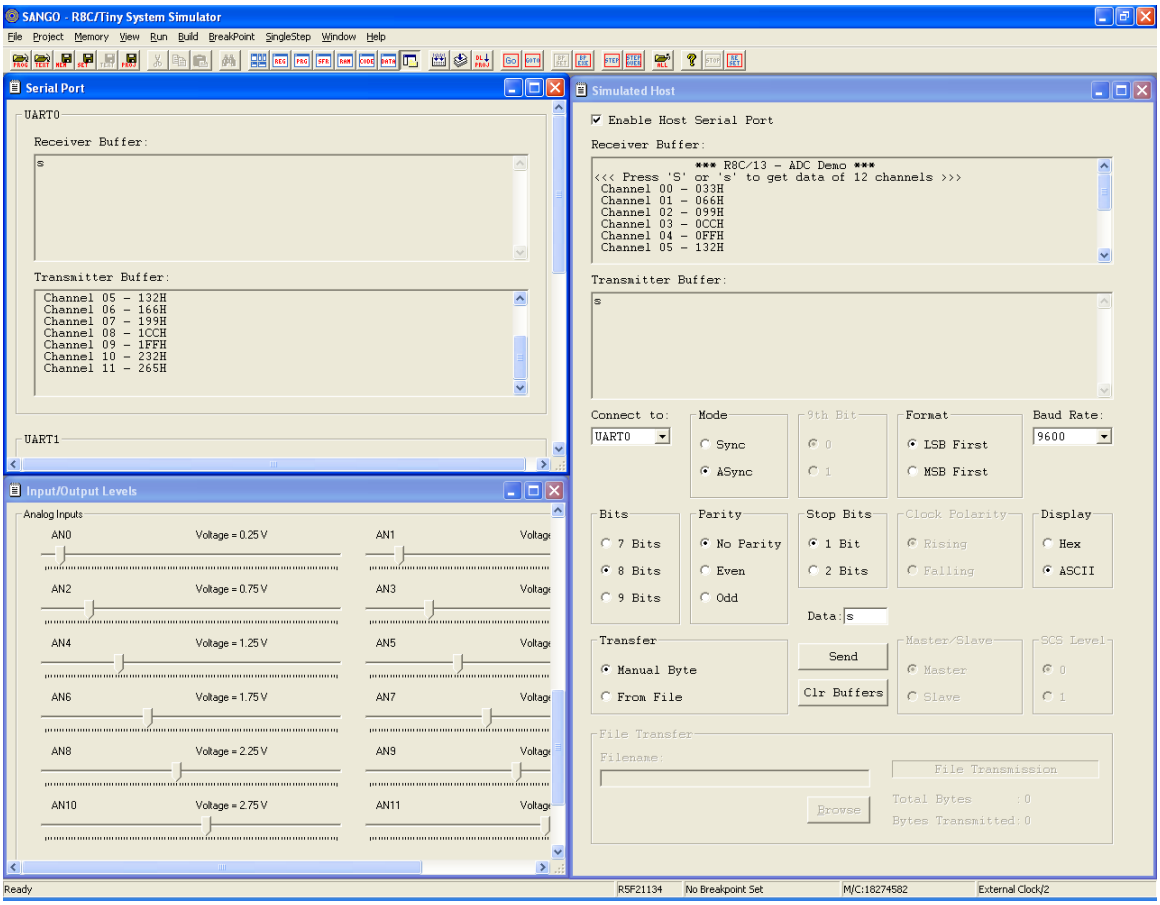


Then move the scroll bar to down to view the variable analog voltage sources and set 5V to Vref pin using analog reference setting.





Then open serial port and simulated host window and arranage them. Enabled the host function and do setting for the host 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,

```
*** R8C/13 - ADC Demo ***
<<< Press 'S' or 's' to get data of 12 channels >>>
```

is sent to the host through serial port 0. The host will capture this message and display it in receiver buffer. Now send a character ‘S’ or ‘s’ from host by typing the character ‘S’ or ‘s’ in the edit class above the send button. This will send the typed character to the micon’s serial port.

After receiving the character ‘s’ or ‘S’ from host, micon will read all 12 channels and send to the host as shown below:

```
Channel 00 - 033H
Channel 01 - 066H
Channel 02 - 099H
Channel 03 - 0CCH
Channel 04 - 0FFH
Channel 05 - 132H
Channel 06 - 166H
Channel 07 - 199H
Channel 08 - 1CCH
Channel 09 - 1FFH
Channel 10 - 232H
Channel 11 - 265H
```