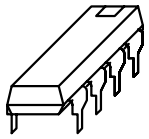
Protean Logic Inc.

Sales: (303) 828 9156 w Fax: (303) 828 9316 w http: //www.protean-logic.com

RSB509B



Product Information

Overview

Protean's **RSB509b** is an easy to use, low cost IC that buffers RS232 serial data for a host microprocessor. The **RSB509b** stores serial data received on the input pin and retransmits the stored data to the host through the interface pin. The **RSB509b** frees the host microprocessor from serial data timing concerns. The real time considerations of RS232 interrupt handling are thus eliminated or reduced. Processors that emulate RS232 hardware via software, like the FBASIC TICkit or the PARALLAX Basic STAMP, can perform other tasks while the **RSB509b** buffers serial data. The processor can retrieve the stored data when it has the time to spare by pulsing the interface pin.

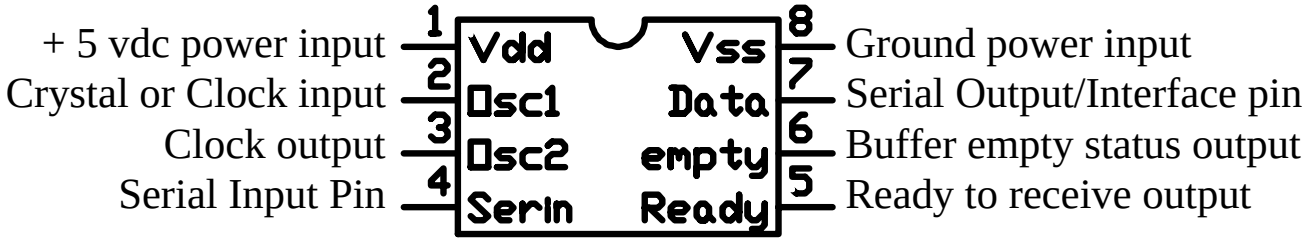
The **RSB509b** can store up to 32 bytes of received data. When the host processor is ready to receive the data it signals the RSB509 to transmit by creating a small pulse on the interface pin. When using a TICkit or a STAMP as the host processor, the user program simply outputs a high on the interface pin immediately before the rs_receive or serin line.

Connecting the RSB509b is easy. Only a few pull down resistors and a crystal or resonator is required. The RSB509b only requires one connection to the host processor.

Features

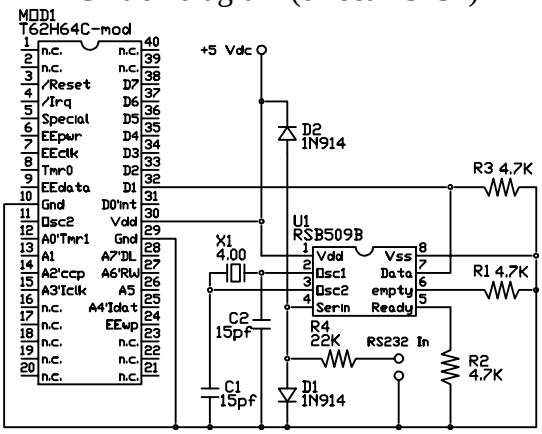
- Small 8-pin plastic DIP package. Requires 4mhz clock source (crystal, resonator or square wave) and pull down resistors for data, empty, and ready outputs.
- 32 Byte buffer transparantly receives data and retransmits it to the host when requested by the.
- Only one host processor pin required to completely interface to RSB509b.
- Pulse protocol on the interface pin is used to program the RSB509b operation modes and to regulate data transfere.
- Programmable input baud rates 9600, 4800, 2400, 1200. Lower baud rates can be achieved by using lower oscillator frequency on the RSB509b.
- Receive input pin programmable for inverted (pull down) or non-inverted (RS232 driver output) signal polarity.
- Communicates with host at 9600 baud via open drain pin.
- Host indicates it is ready to receive by bringing the data line high momentarily. Host emulation routine makes interface pin an input when it starts which inherently signals the RSB509b to send data from the buffer.
- Programmable single byte or burst mode transfere to host.
- Programmable byte match before buffer option for addressable packet reception.
- Programmable break required before byte match option.
- Buffer empty output and buffer full (/ready) output for handshaking and buffer status.

RSB509 Pin Assignments

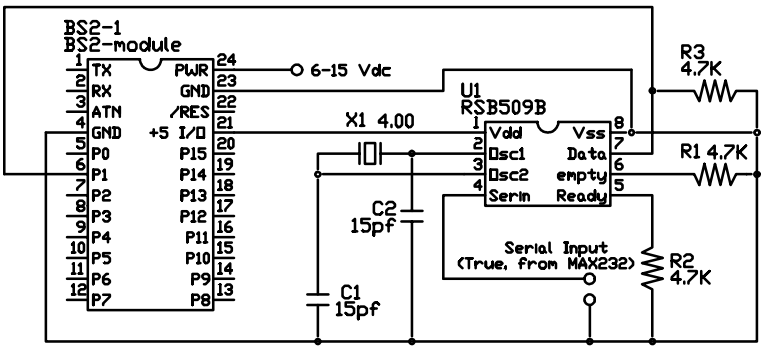


Example Connection Diagrams

TICkit 62 diagram (direct RS232)



Basic Stamp II diagram (buffered RS232)



Example Software Control of RSB509b

TICkit 62 Sample Program

```
; Program for RSB509A serial buffering

DEF tic62_c
LIB fbasic.lib
LIB rsb509b.lib

FUNC none main
BEGIN
    ; generate a long pulse on interface pin to
    ; signal initialization to RSB509
    pin_high( pin_d0 )
    delay( 10 )
    =( in_err, pin_in( pin_d0 ))

    ; no match byte used but one must be sent
    ; as a place holder during initialization
    rs_param_set( rs_invert | rs_9600 | pin_d0 )
    rs_send( ' ' )

    ; serial input is inverted and 9600 baud.
    rs_send( rsb509_invert | rsb509_baud1 )

    ; wait for RSB509 to reset.
    delay( 100 )
    REPEAT
        ; generate a quick pulse for each
        ; byte to be read. pin high creates start
        ; of pulse. rs_receive creates end of pulse
        ; when it makes the pin an input.
        pin_high( pin_d1 )

        ; pin D1 will be used for input
        rs_param_set( rs_invert | rs_9600 | pin_d0 )

        ; read each byte with a short wait
        =( in_val, rs_receive( 100, 0b, in_err ))
        IF in_err
        ELSE
            rs_param_set( debug_pin )
            con_out_char( in_val )
        ENDIF

    LOOP

ENDFUN
```

Basic Stamp II Sample Program

```
' sample program for RSB509 on STAMP II

inval    VAR    byte

' generate a 10 ms pulse to signal initialization
HIGH 1
PAUSE 10
INPUT 1

' pin P1 is used for input.
' we are using open, inverted, 9600, 8N1 format
' no match byte is used but we must send a byte to
' hold our place during initialization
SEROUT 1, 49236, [" "]

' pin P1 is used for input.
' we are using open, inverted, 9600, 8N1 format
' serial input is inverted and 9600 baud
SEROUT 1, 49236, [%00000100]

' wait for RSB509 to reset
PAUSE 100

around:
' generate a quick pulse for each byte
' to be read. HIGH creates the start of
' the pulse. SERIN creates the end of the
' pulse when it makes the pin an input
' because the interface pin is pulled low
HIGH 1

' pin P1 is used for input.
' we are using open, inverted, 9600, 8N1 format
' if a byte is not received within 1 ms branch to
' "around" to poll again.
' data will only be displayed when byte is received
SERIN 1, 16468, 1, around, [inval]

' show character received on debug window
DEBUG inval

GOTO around
END
```