

AppKit:

Using the SN75176 in an RS485 Network

This AppKit shows how to create a [bi-directional](#) RS485 network using the Texas Instruments SN75176 Differential Bus Transceiver chip and a Parallax BASIC Stamp single-board computer.

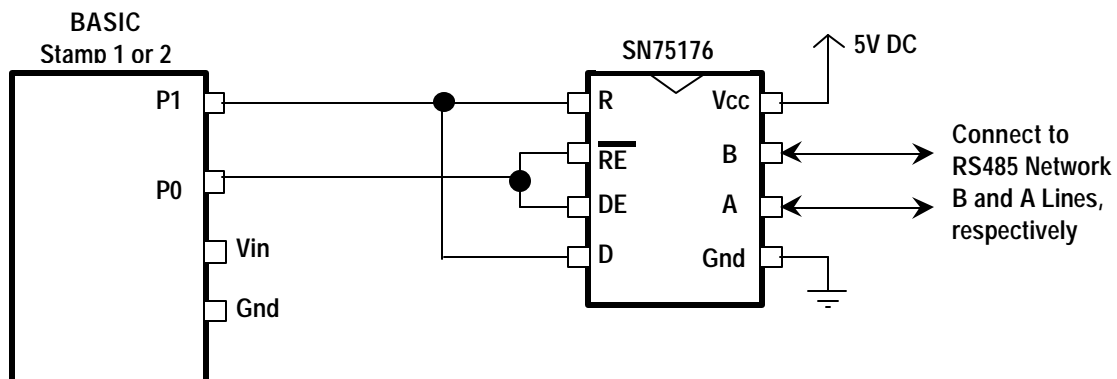
Description

RS485 is commonly used to provide strong serial signals able to withstand long cable distances (up to 4000 feet) at high baud rates in potentially noisy electrical environments. Two wires carrying an RS485 signal (the A and B lines) provide a signal base from which many devices can communicate. Twisted pair wire is recommended for long distances, but for short distances we recommend using 24 gauge wire. Up to 32 devices can be connected to an RS485 data line with the SN75176 and communicate using a data protocol. This is referred to as an RS485 network, or an RS485 drop network.

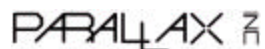
The heart of this communication is the Texas Instruments SN75176 Differential Bus Transceiver chip. This chip converts RS485 signals to RS232 TTL-level signals allowing devices that traditionally communicate over standard RS232 serial connections to communicate over a single two-wire RS485 network.

Hardware Interface

The SN75176 chip has only 8 pins. Here is how it is connected to the BASIC Stamp:



The SN75176 requires 30 mA, therefore we recommend driving only one of these chips from the Stamp's Vdd pin provided no other components require current. Setting RE (Not Receiver Enable) to Low, the R (Receiver) pin is enabled, allowing the Stamp to receive any data coming over the A and B RS485 network lines. Setting DE (Driver Enable) to High allows the Stamp to transmit data over the RS485 network. Vcc and Vdd are electronic terms with the same meaning.



Receiving and transmitting serial data using the BASIC Stamp works like this:

To Transmit data:

- 1) Set P0 to High.
- 2) Use SEROUT on P1 to transmit the data.

To Receive Data:

- 1) Set P0 to Low.
- 2) Use SERIN to receive serial data on P1

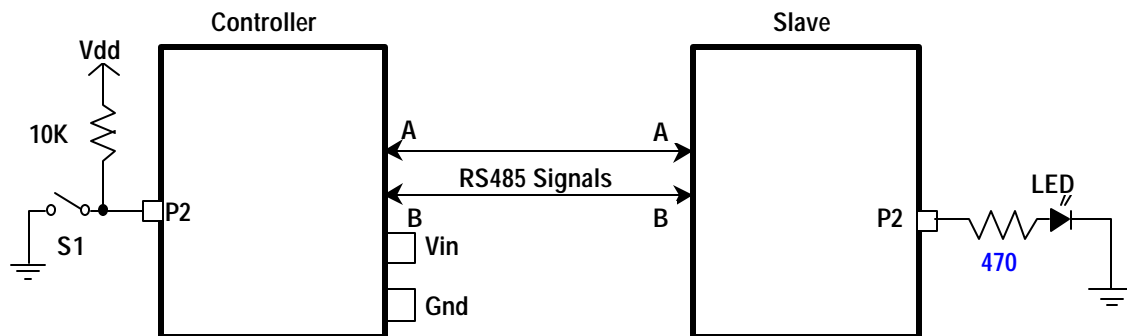
Setting Up the Network

An RS485 network operates with a controller/slave communication protocol. One controller commands everything, and one or more slaves respond to the commands. This example uses one BASIC Stamp as the controller, and one BASIC Stamp as the slave. The circuit shown below can easily be expanded to have up to 32 BASIC Stamps communicating on one RS485 network by connecting additional slaves to the network A and B lines.

Construct the network by following these steps (see figure below):

- 1) Build two identical circuits as shown below. Designate one to be the controller and the other to be the slave.
- 2) On the controller connect the supplied switch between the BASIC Stamp's P2 pin and ground, and the 10K resistor between P2 and Vcc as a pull-up resistor. This makes the line +5 V unless a button is pushed in which case it goes to 0 V.
- 3) On the slave connect the supplied LED and resistor in series between the BASIC Stamp's P2 pin and ground.
- 4) Program the controller Stamp with the supplied program "control.bas" if you are using a Stamp 1, or "control.bs2" if you are using a Stamp 2.
- 5) Program the slave BASIC Stamp with the supplied program "slave.bas" if you are using a Stamp 1, or "slave.bs2" if you are using a Stamp 2.
- 6) Flick the switch on the Controller. The Slave LED will turn on or off depending on the controller's switch position.

See the program listings for details of the software communication process.



Tips:

- For more technical details on RS485, see the "RS-422 and RS-485 Application Note" available from <http://www.bb-elec.com>. This application note tells you all you'll want to know about RS485.
- See <http://www.jdrichards.com> for more RS485 information and a program listing that demonstrates a network protocol using the BASIC Stamp.
- Complications may arise when the distance between communicating units becomes large. A terminator resistor may be needed at the end of a network between the A and B lines when the network length reaches more than several hundred feet. See the App Note mentioned above at: <http://www.bb-elec.com> for more details on termination.
- If the RS485 communication does not work, you may have the A and B lines switched.

BASIC Stamp I (BS1-IC) and BASIC Stamp Ver. D Program Listing

```
' Program: control.bas (BS1 to RS485 interface via a SN75176 chip)
,
' This program interfaces a BASIC Stamp 1 to an RS485 network
' using the SN75176 Differential Bus Transceiver chip from
' TI. This program is meant to operate with another Stamp 1
' connected to the same RS485 network and is running the
' slave.bs1 program.
,
' Pins 2 and 3 of the SN75176 chip are connected to pin 0 of
' the Stamp 1. Pins 1 and 4 of the SN75176 chip are
' connected to Pin 1 of the Stamp 1.
,
' This program expects an active-low button or switch to be
' connected to pin 2 of the Stamp 1. When the button or switch
' sets pin 2 low, the character "L" is sent over the RS485
' network. When the button or switch sets pin 2 high then
' the character "H" is sent over the network.
,
' Note. Setting pin 0 on the Stamp 1 high puts the SN75176 into
' transmit mode. So any serial data transmitted out of pin 1 on
' the Stamp 1 will be transmitted over the RS485 network.
,
'===== Initialize =====
input 2                'Make pin 2 the button input
output 0              'Make pin 0 an output
high 0                'Put the SN75176 into transmit mode

'===== Begin Program =====
if pin2<>1 then loop1:  'If pin 2 is initially High, send...
    SEROUT 1,N2400,("H")  '... an "H" over the RS485 network

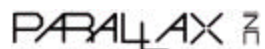
'===== Main Loop =====
loop1:
    BUTTON 2,0,255,0,b0,1,preloop2  'Wait till pin 2 is low
    goto loop1

preloop2:
    SEROUT 1,N2400,("L")  'Send a "L" over the RS485 network.

loop2:
    BUTTON 2,1,255,250,b0,1,loop_again 'Wait till pin 2 goes high
    goto loop2

loop_again:
    SEROUT 1,N2400,("H")  'Send an "H" over the rs485 network.

goto loop1:            'Loop forever
```



```
' Program: slave.bas (BS1 to RSS485 interface via a SN75176 chip)
'
' This program interfaces a BASIC Stamp 1 to an RS485 network
' using the SN75176 Differential Bus Transceiver chip from
' TI. This program is meant to operate with another Stamp 1
' connected to the same RS485 network and is running the
' control.bs1 program.
'
' Pins 2 and 3 of the SN75176 chip are connected to pin 0 of
' the Stamp 1. Pins 1 and 4 of the SN75176 chip are
' connected to Pin 1 of the Stamp 1.
'
' This program expects an LED and resistor in series to be connected to
' pin 2 of the Stamp 1. When an "H" comes across the RS485 network
' pin 2 is set high, turning on the LED. When an "L" is received
' pin 2 of the Stamp 1 is turned off.
'
' Note: Setting pin 0 on the Stamp 1 low puts the SN75176 into
' receive mode. So any serial data received on pin 1 of the
' Stamp 1 will be read in with the SERIN command.
'
'===== Initialize=====
output 2           'Make pin 2 the LED connected pin.
output 0           'Make pin 0 an output pin.
low 0              'Put the SN75176 into receive mode.

'===== Begin Program =====
loop1:
SERIN 1,N2400,b0   'Read a byte of data coming in.

if b0<>"H" then is_low      'If H, then ...
  high 2                 '... set pin 2 high, truning on LED
  goto loop1

is_low:             'If not an H, then turn off the LED
  low 2

goto loop1          'Loop forever
```

BASIC Stamp II (BS2-IC) Program Listing

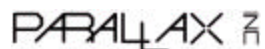
```
' Program: control.bs2 (BS2 to RS485 interface via a SN75176 chip)
,
' This program interfaces a BASIC Stamp 2 to an RS485 network
' using the SN75176 Differential Bus Transceiver chip from
' TI. This program is meant to operate with another Stamp 2
' connected to the same RS485 network and is running the
' slave.bs2 program.
,
' Pins 2 and 3 of the SN75176 chip are connected to pin 0 of
' the Stamp 2. Pins 1 and 4 of the SN75176 chip are
' connected to Pin 1 of the Stamp 2.
,
' This program expects an active-low button or switch to be
' connected to pin 2 of the Stamp2. When the button or switch
' sets pin 2 low, the character "L" is sent over the RS485
' network. When the button or switch sets pin 2 high then
' the character "H" is sent over the network.
,
' Note. Setting pin 0 on the Stamp 2 high puts the SN75176 into
' transmit mode. So any serial data transmitted out of pin 1 on
' the Stamp 2 will be transmitted over the RS485 network.
,
'===== Variables =====
btnWk var byte           'Workspace for the BUTTON command
'===== Initialize =====
input 2                 'Make pin 2 the button input
output 0                'Make pin 0 an output
high 0                  'Put the SN75176 into transmit mode
'===== Begin Program =====
if(in2<>1)then loop1:   'If pin 2 is initially High, send...
  SEROUT 1,16468,1,["H"] '... an "H" over the RS485 network
,
'===== Main Loop =====
loop1:
  BUTTON 2,0,255,0,btnWk,1,preloop2 'Wait till pin 2 is low
  goto loop1

preloop2:
  SEROUT 1,16468,1,["L"]           'Send a "L" over the RS485 network.

loop2:
  BUTTON 2,1,255,250,btnWk,1,loop_again 'Wait till pin 2 goes high
  goto loop2

loop_again:
  SEROUT 1,16468,1,["H"]           'Send an "H" over the rs485 network.

goto loop1:                   'Loop forever
' Program: slave.bs2 (BS2 to RSS485 interface via a SN75176 chip)
```



```

'
' This program interfaces a BASIC Stamp 2 to an RS485 network
' using the SN75176 Differential Bus Transceiver chip from
' TI. This program is meant to operate with another Stamp 2
' connected to the same RS485 network and is running the
' control.bs2 program.
'
' Pins 2 and 3 of the SN75176 chip are connected to pin 0 of
' the Stamp 2. Pins 1 and 4 of the SN75176 chip are
' connected to Pin 1 of the Stamp 2.
'
' This program expects an LED and resistor in series to be connected to
' pin 2 of the Stamp 2. When an "H" comes across the RS485 network
' pin 2 is set high, turning on the LED. When an "L" is received
' pin 2 of the Stamp 2 is turned off.
'
' Note: Setting pin 0 on the Stamp 2 low puts the SN75176 into
' receive mode. So any serial data received on pin 1 of the
' Stamp 2 will be read in with the SERIN command.
'
'===== Variables =====
string var byte           "Used to hold the "H" or "L"

'===== Initialize=====
output 2                  'Make pin 2 the LED connected pin.
output 0                  'Make pin 0 an output pin.
low 0                     'Put the SN75176 into receive mode.

'===== Begin Program =====
loop1:
  SERIN 1,16468,60000,loop1,[string]
  'Read a byte of data coming in.
  if(string<>"H")then is_low 'If H, then ...
  high 2                    '... set pin 2 high, turning on LED
  goto loop1
is_low:                    'If not an H, then turn off the LED
  low 2
  goto loop1                'Loop forever

```