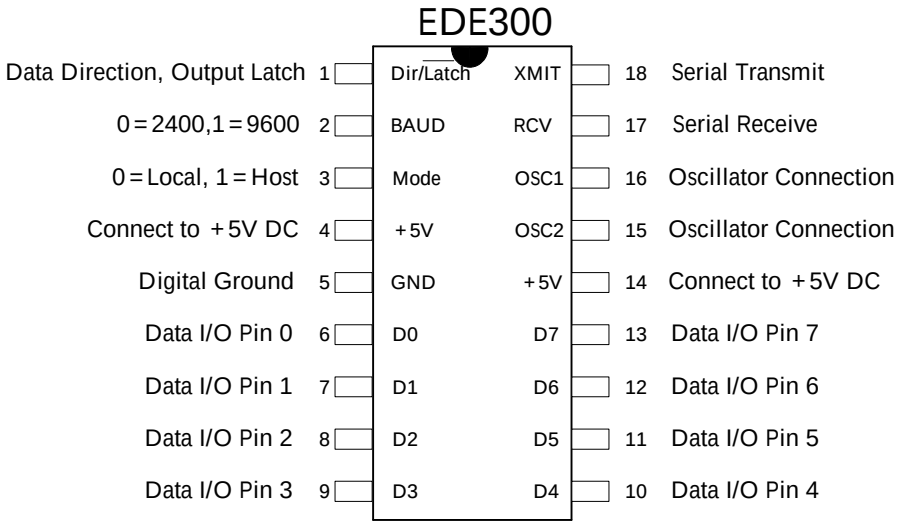
EDE300 Parallel/ Serial Transceiver IC



The EDE300 Parallel/ Serial Transceiver IC is a 5 volt, 18 pin package designed to conveniently convert 8-bit parallel data into serial format and vice-versa. The EDE300 is ideal for logging data to a PC, controlling hardware via the PC serial port, communicating 8-bit data via a serial connection, and expanding I/O capabilities on microcontrollers and Stamps. The EDE300 is BAUD-selectable, offering both 2400 and 9600 BAUD communication. PC connection requires the use of a voltage level shifter such as the MAX233. The EDE300 can be configured as either a half-duplex transmitter, a half-duplex receiver, or a half-duplex bi-directional transceiver.

PIN DEFINITIONS

Flow Control Pins:

Data Direction, /Latch (PIN 1):	In *Local Control Mode* pin is an INPUT: Input of 0 = Parallel to Serial, 1 = Serial to Parallel. In *Host Control Mode*, pin is an OUTPUT: EDE300 drives this pin low on receive, high on transmit
Baud Rate (PIN 2):	0 = 2400 BAUD, (N 8 1); 1 = 9600 BAUD, (N 8 1)
Mode (PIN 3):	0 = *Local Control Mode*, 1 = *Host Control Mode*

Data Pins:

Serial Transmit (PIN 18) :	Serial data is output from the EDE300 via this pin
Serial Receive (PIN 17):	Serial data is input to the EDE300 via this pin
D0..D7 (PIN 6..PIN 13):	Data I/O Pins (DO is least significant bit)

Clock/ Power Pins:

OSC1,OSC2 (PIN 16,PIN 15):	4 MHz Resonator Connection
Power (PIN 14,PIN 4):	Connect to +5 VDC
Ground (PIN 5):	Connect to 0 VDC (GND)

LOCAL CONTROL MODE

In *Local Control Mode*, enabled by wiring the Mode pin low (Pin 3), directional control is selected by the condition of the Dir/Latch pin (Pin 1). When *Local Control Mode* is selected and the Dir/Latch pin is low, the EDE300 acts as a Parallel to Serial transmitter, continually reading the 8 data pins (D0..D7) and sending this value out the serial XMIT pin (Pin 18). When *Local Control Mode* is selected and the Dir/Latch pin is high, the EDE300 acts as a Serial to Parallel receiver, continually monitoring the RCV pin (Pin 17) and latching the received data onto the 8 data pins (D0..D7).

NOTE: Switching the Dir/Latch pin causes the data I/O pins (D0..D7) to switch from inputs to outputs and vice-versa. Care should be taken that data is not input into the 8 data pins while the EDE300 is in Serial to Parallel mode (Pin 17 high); otherwise damage may occur to the EDE300 or the external circuitry as both attempt to drive the I/O pins. See the *Host Control Mode* section's schematics for details on switching the data inputs and outputs with the EDE300. The following schematics depict the EDE300 in a basic, unidirectional hookup, transmitting (Figure One) and receiving (Figure Two) data to and from a PC via the PC serial port COM 1.

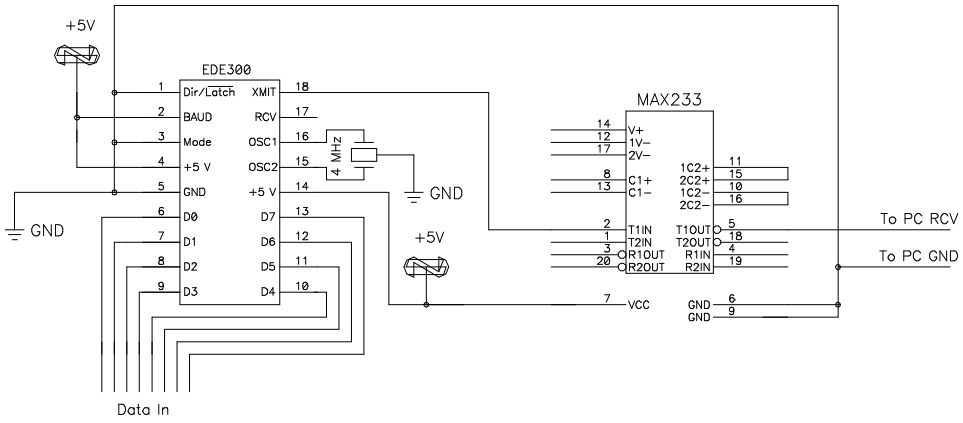


Figure One: Unidirectional Serial Transmission to PC

The following code, written in 'Turbo C', illustrates the reception of the EDE300's data stream. It can be used in conjunction with the hardware setup in Figure One.

```
#include <stdio.h>
char data;                    /* 8 BIT VARIABLE TO HOLD INCOMING DATA */
main()
```



```
{
while(1)                        /* BEGIN INFINITE LOOP */
        {
        data = inportb(0x3f8);  /* READ COM PORT #1 */
        printf("%d",data);      /* CAST 'CHAR'  TO 'INT' TO VIEW ACTUAL VALUE */
        if (data = = 0) exit(0); /* IF INPUT IS ZERO, EXIT INFINITE LOOP */
        }
}
```

NOTE: If your PC is not sending or receiving data properly in these examples, you may need to enter the following at the command prompt:

MODE COM1 9600 N 8 1

This will set up the COM1 port correctly for use at 9600 BAUD as is used in the hardware examples in this section.

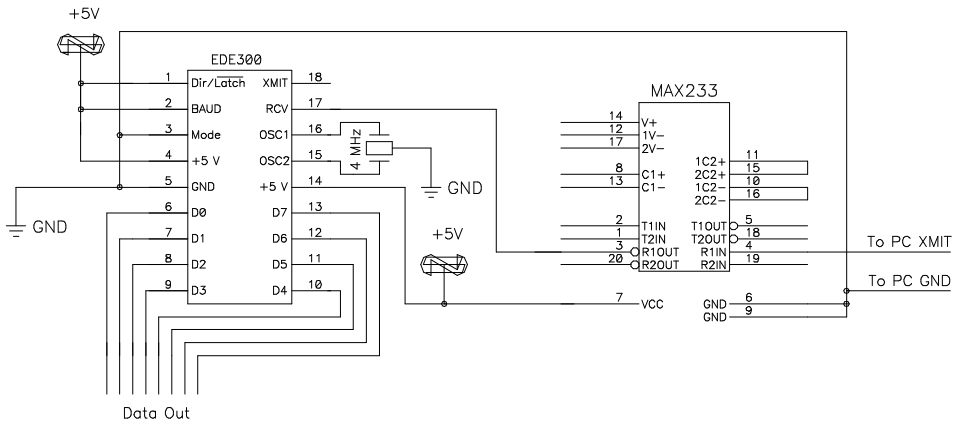


Figure Two: Unidirectional Serial Reception from PC

The following code, written in 'Turbo C', illustrates a data write to the EDE300. It can be used in conjunction with the hardware setup in Figure Two.

```
#include <stdio.h>
char data;                      /* 8-BIT VARIABLE TO HOLD OUTGOING DATA */
main()
{
data = 128;                     /* VALUE TO BE WRITTEN TO EDE300 */
outportb(0x3F8,data);           /* WRITE 'DATA' TO COM PORT #1 */
}
```


Notice that the MAX233 (or MAX232) voltage level shifter is needed for connection to a PC to meet the required RS-232 line voltage requirements. A level shifter, as is used in the above two examples, is not needed for connection to a Stamp or microcontroller, as can be seen in the 'BASIC STAMP CONNECTION' section, or between two EDE300's, as illustrated below.

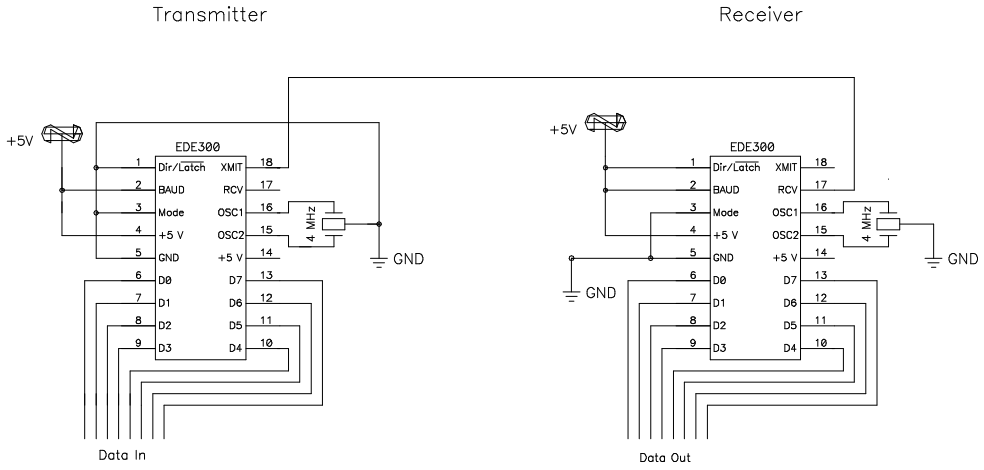


Figure Three: Two-Wire/ Wireless Data Transmission

This arrangement is useful when 8-bit data needs to be sent in one direction across a twisted pair or coaxial cable. A variation on this schematic is the addition of a RF transmitter/ receiver pair on the data line, providing a wireless 8-bit connection. For example, with the TX-99 and RE-99 transmitter/ receiver pair from Ming Microsystems (available from Digi-Key, 1-800-DIGI-KEY), an 8-bit wireless bus can be implemented. Serial data from the transmitting EDE300 is fed into the TX-99. The RE-99 receives the serial data and feeds it to the receiving EDE300. Alternately, short range transmission could be accomplished via an infrared transmitter/ receiver pair.

HOST CONTROL MODE

In *Host Control Mode*, selected by wiring the Mode (Pin 3) high, the EDE300 takes data-flow instructions from a host device connected to it serially instead of from the Data Direction Pin (Pin 1). The host, typically a PC, STAMP, or microcontroller, sends a one-byte command to the EDE300 instructing it to either receive one data byte and latch it onto the data pins (D0..D7) or to read the data pins and transmit this value serially.

The host should transmit a value of either "1" or "2" to the EDE300 in *Host Control Mode*. If an ASCII '1' is received (00110001b), the EDE300 will wait for one byte of data to be transmitted serially from the host and will then latch this data onto its data pins. The Dir/ Latch (Pin 1) pin will be held high. If an ASCII '2' is received (00110010b), the EDE300 will read its data pins and transmit this value serially back to the host. The Dir/ Latch (Pin 1) pin will be held low. This unidirectional *Host Control Mode* is convenient when too much data would be transmitted using *Local Control Mode*. Note that the ASCII characters "1" and "2" must be used, instead of the values one and two. This enables the EDE300 to be used with a terminal program as well.

Host Control Mode, however, is most useful when implemented as a bi-directional interface.

Making use of the Dir/ Latch pin, the EDE300 can simultaneously latch data from a PC/ Stamp and transmit data to the PC/ Stamp. The host PC/ Stamp would simply send a "1" command followed by an output data byte, and then send a "2" command to tell the EDE300 to transmit data back. This arrangement is shown in Figure Four below. The Dir/ Latch pin causes a data receive request to be latched onto the '373, and a data transmit request to read from the '245. *Note: use of the '245 may in some circumstances cause brief glitches on the '373 outputs. We recommend this arrangement only for output devices in which a small (0 to 25 ns) output glitch would be acceptable (eg. LED's, relay control, etc.).*

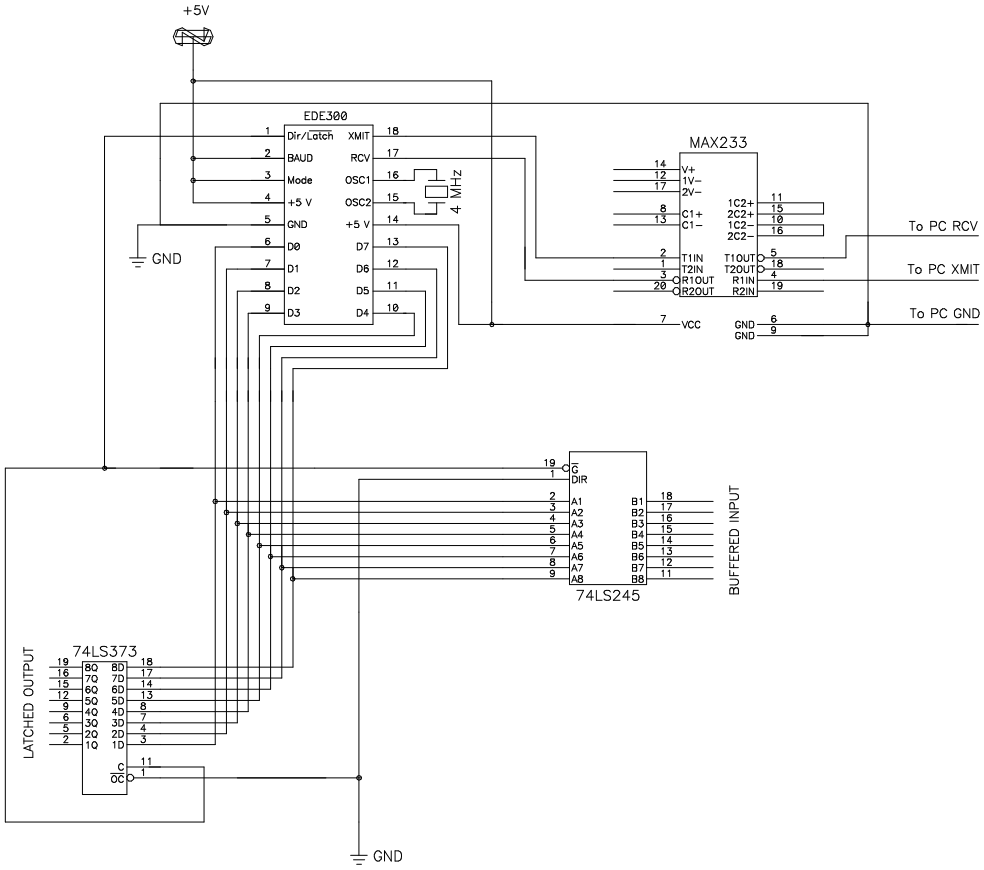


Figure Four: Simultaneous Bi-directional Host-Control Mode Arrangement

The following code, written in 'Turbo C', illustrates one method for using the EDE300 in bi-directional mode:

```
#include <stdio.h>
char value;                              /* main loop data variable */

send (char x)                            /* write 'x' to EDE300 */
{
while ((inportb(0x3FD) & 0x20) == 0);    /* hold until transmitter is ready */
outportb(0x3F8,0x31);                    /* select 'receive byte' on EDE300 */
while ((inportb(0x3FD) & 0x20) == 0);    /* hold until transmitter is ready */
outportb(0x3F8,x);                       /* write 'x' to EDE300 */
}
```



```
char receive()                          /* receive one byte from EDE300 */
{
char value;                             /* stores incoming byte */
value = inportb(0x3F8);                 /* flush input buffer */
while ((inportb(0x3FD) & 0x20) = = 0);  /* hold until transmitter is ready */
outportb(0x3F8,0x32);                   /* select 'send byte' on EDE300 */
while ((inportb(0x3FD) & 0x01) = = 0);  /* hold until a byte is received */
value = inportb(0x3F8);                 /* receive byte from EDE300 */
return (value);
}

main()                                  /* main program loop */
{
bioscom (0,0xE0|0x03,0);                /* set 9600 BAUD, 8 bit data */

do
        {
        value = receive();              /* read byte from EDE300 into 'value' */
        send (value);                   /* write 'value' back to EDE300 */
        }
while (value != 0);                     /* exit if 'value' = 0 */
}
```

This program reads one byte from the '245 via the EDE300 and writes the same value back to the '373 via the EDE300. This is a very simple example - it is intended only to illustrate the proper use of the EDE300's *Host Control Mode*. Notice that the subroutine 'receive()' returns one byte of data from the EDE300, and the subroutine 'send(value)' transmits the byte stored in value to the EDE300. In this example, the BAUD rate does not need to be set from the command prompt; it is set to 9600 BAUD in software.

BASIC STAMP CONNECTION

The EDE300 can make a very effective I/O expander for the BASIC Stamp™ or other microcontroller devices capable of transmitting and receiving serial data. In order to keep from overrunning the Stamp with data from the EDE300, it is recommended that the EDE300 be used in *Host Control Mode* when used with the Stamp, whether you are performing unidirectional input, unidirectional output, or half-duplex bi-directional I/O. This way, the host (Stamp) controls when it will receive a byte and when it will transmit a byte.

Bi-Directional Communication with BASIC Stamp™ I

To use the EDE300 with the BASIC Stamp™ in a bi-directional arrangement, refer to the schematic in Figure Four, above. As the EDE300 can communicate with a Stamp using standard +5V signals, no voltage level converter (MAX233) is needed for the connection. Simply remove it from the schematic shown in Figure Three, and wire in the Stamp as follows:

Stamp Pin 7 <-------> EDE300 Pin 17 (Serial Receive)
Stamp Pin 6 <-------> EDE300 Pin 18 (Serial Transmit)

Then connect the EDE300's BAUD Pin (Pin 2) to GND instead of +5V as is shown in the example for connection to a PC in Figure Four. This will tell the EDE300 to communicate at 2400 BAUD instead of 9600. Also, you will need to connect the Stamp's ground pin to the EDE300's ground pin.

Once connected, the following example code, running on the Stamp, illustrates the simple example of polling the '245's eight inputs and writing this value back to the '373's latched outputs.

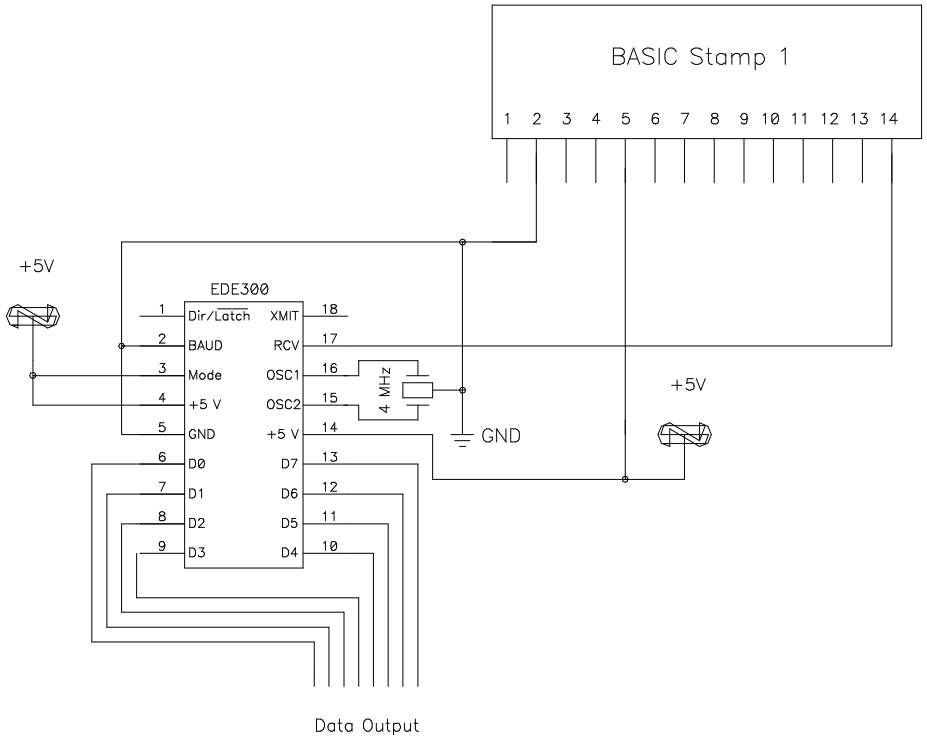
```
pause 200                       ' pause Stamp while all circuitry powers up
loop: serout 7,T2400, ("2")     ' send the character "2" out pin 7 at 2400 BAUD
serin 6,T2400,b2                ' read serial byte send back from EDE300, store in b2
serout 7,T2400,("1")            ' tell EDE300 to accept next byte of data
serout 7,T2400,(b2)             ' send value in b2 to EDE300 to be latched onto output of '373 IC
if b2 < > 0 then loop           ' continue until input byte = 0
```

If, however, you simply wish to use the EDE300 for unidirectional communication with a Stamp (to expand the I/O capabilities of the Stamp), you would not need the '245 or '373 IC's. The EDE300's eight data pins would serve as the eight input pins in unidirectional input mode, or as the eight output pins in unidirectional output mode.

NOTE: In the following two arrangements, the EDE300 is being used in *Host Control Mode*, which means that its Data Direction/Latch Pin (Pin 1) is made an output rather than an input (as it is in *Local Control Mode)*. In the following two examples leave this pin unconnected; data flow direction is determined by the host. The Data Direction/Latch Pin (Pin 1) should NEVER be driven while the EDE300 is in *Host Control Mode*.

Unidirectional Communication with BASIC Stamp™ I - Output Only

The following schematic will make eight digital outputs from one Stamp I/O pin.

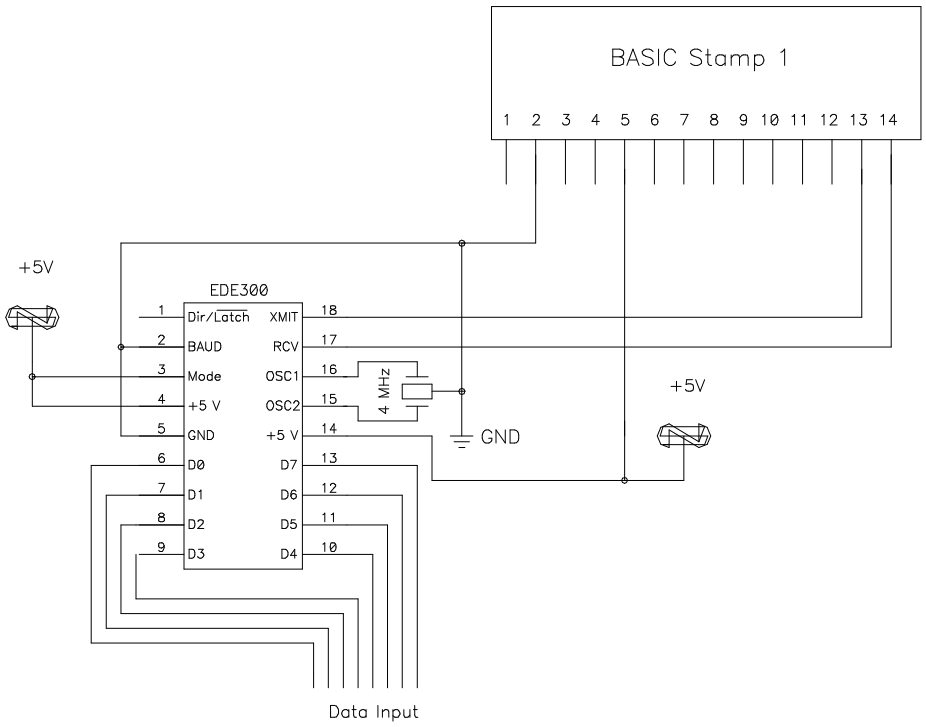


The following code, written for the BASIC Stamp™ I, when used with the above schematic, will loop from zero to ten, writing the appropriate binary value to the Output Port of the EDE300 each time, stopping with the binary value 00001010b latched onto the EDE300's data I/O port.

```
pause 200                       ' pause Stamp while EDE300 powers up
for b2 = 0 to 10
serout 7,T2400,("1")            ' tell EDE300 to accept next byte of data
serout 7,T2400,(b2)             ' send value in b2 to EDE300 to be latched
next b2                         ' repeat loop until b2 = ten
```

Unidirectional Communication with BASIC Stamp™ I - Input Only

The following schematic will make eight inputs from two Stamp pins. Notice that in the previous example only one pin was required; in this example, however, we still need to maintain a serial output pin on the stamp to tell the EDE300 to send data, as well as a serial input pin on the Stamp to receive that data.



The following code, written for the BASIC Stamp™ I, when used with the above schematic, will read the value on the EDE300's I/O port and return it to the PC programming the Stamp via a debug window.

```
pause 200                       ' pause Stamp while EDE300 powers up
serout 7,T2400,("2")            ' tell EDE300 to read its I/O port and transmit the value
serin 6,T2400,b2                ' read transmitted value into b2
debug b2                        ' return value in b2 to Stamp's programming PC via debug window
```


ABSOLUTE MAXIMUM RATINGS

Oscillator frequency ..... 4 MHz
Supply Voltage 7.5V
Ambient temperature under bias ..... -40°C to +125°C
Max. current sinked by output pin 25mA
Max. current sourced by output pin 25mA
Max. current sourced by all 8 data pins..... 200mA
Max. current sinked by all 8 data pins..... 200mA

STANDARD OPERATING CONDITIONS

Supply voltage ..... 3.0V to 5.5V
Operating temperature ..... 0°C to +70°C

The EDE300 IC is implemented as firmware on a PIC16C554 microcontroller, manufactured by Microchip Technology, Inc. For a more comprehensive technical summary of this device, please refer to the PIC16C554 datasheet (available from the E-Lab web site).

IMPORTANT NOTICE

E-LAB Digital Engineering, Inc. (E-LAB), reserves the right to change products or specifications without notice. Customers are advised to obtain the latest versions of product specifications, which should be considered when evaluating a product's appropriateness for a particular use.

THIS PRODUCT IS WARRANTED TO COMPLY WITH E-LAB'S SPECIFICATION SHEET AT THE TIME OF DELIVERY. BY USING THIS PRODUCT, CUSTOMER AGREES THAT IN NO EVENT SHALL E-LAB BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES AS A RESULT OF THE PERFORMANCE, OR FAILURE TO PERFORM, OF THIS PRODUCT.

E-LAB MAKES NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

E-LAB'S LIABILITY IS FOR A PERIOD NO GREATER THAN 90 DAYS FROM DATE OF SHIPMENT BY E-LAB AND IS LIMITED TO REPLACEMENT OF DEFECTIVE PRODUCT. This warranty covers only defects arising under normal use and not malfunctions resulting from misuse, abuse, modification, or repairs by anyone other than E-LAB.

E-LAB'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF E-LAB. Life support devices or systems are those which are intended to support or sustain life and whose failure to perform can be reasonably expected to result in a significant injury or death to the user. Critical components are those whose failure to perform can be reasonably expected to cause failure of a life support device or system or affect its safety or effectiveness.

COPYRIGHT NOTICE

CONTACTING US

We are continually updating our product line. Please contact us for our latest product information.

E-LAB Digital Engineering, Inc.
1600 N. 291 Hwy. Ste. 330
P.O. Box 520436
Independence, MO 64052-0436

Telephone: (816) 257-9954
FAX: (816) 257-9945

Internet:
www.elabinc.com

E-Mail: support@elabinc.com