HP 82920A
Current Loop Interface

Owner's Manual

HEWLETT
PACKARD

Edition 1 November 1984

Part Number
82920-90001

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Portable Computer Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

Printing History

Edition 1 November 1984 Mfg. No. 82920-90002

Specifications

Transmission Mode	Serial, asynchronous—simplex, half duplex, or full duplex.
Baud Rate	Up to 19,200.*
Back-Panel Connector	Male 25-pin D-subminiature.
Operating Limits	20 mA (active transmitter or receiver). 60 mA (passive transmitter or receiver).
Dimensions	Approximately 19.1 by 13.5 by 2.7 cm (7.5 by 5.3 by 1.1 in.).
Weight	218g (7.7 oz).
Operating Temperature	0° to 40°C (32° to 104°F).
Storage Temperature	−40° to 75°C (−40° to 167°F).
Operating Humidity	5% to 95% relative humidity at 40°C.

* Your computer's operating system may limit the baud rate to 9,600.

Potential for Radio/Television Interference (for U.S.A. Only)

The HP 82920A Current Loop Interface generates and uses radio frequency energy and, if not installed and used properly—that is, in strict accordance with the instructions in this manual—may cause interference to radio and television reception. It has been tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. In the unlikely event that your product does cause interference to radio or television reception (which can be determined by turning off the computer, removing the interface, then turning the computer back on) you are encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the interface cable with respect to the receiver.
- Move the computer and interface away from the receiver.
- Plug the computer into a different ac outlet so that the computer and the receiver are on different branch circuits.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet, prepared by the Federal Communications Commission, helpful: *How to Identify and Resolve Radio-TV Interference Problems*. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock Number 004-000-00345-4.

Contents

Illustrations

Tables

1 Setting Up the Interface

This chapter shows how to install the HP 82920A Current Loop Interface and connect a peripheral device. *Once you have connected a peripheral, read "Where To Go From Here" at the end of the chapter to determine where to turn for further instructions.*

Unpacking and Inspection

Before you unpack the interface, note that its printed-circuit assembly is exposed. The interface is shipped enclosed in an antistatic bag to protect it against electrostatic discharge (ESD) damage.

Observe the following precautions whenever handling, installing, or removing the interface:

- Whenever you remove your interface from the computer, store it in the original antistatic bag and shipping container.
- Always handle the interface by its metal back panel, not by the exposed circuit board.

Failure to observe these precautions may result in mechanical damage and/or electrostatic discharge (ESD) damage to the interface.

The shipping container should include the following items in addition to this manual:

- HP 82920A Current Loop Interface.
- Support guide.

Inspect the interface carefully for any signs of shipping damage. Notify your dealer immediately and file a claim with the carriers involved if the interface is damaged or if anything is missing.

Besides the HP 82920A Current Loop Interface and your computer, you need an HP 82975A Current Loop Cable to connect a peripheral device.

Installing the Current Loop Interface

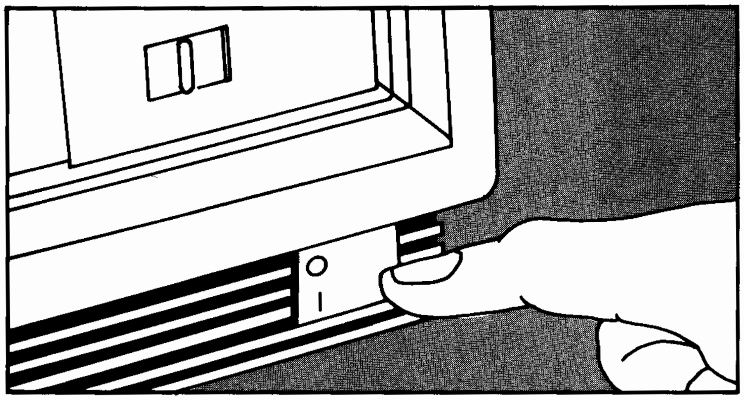
Note the following precautions before you install the interface.

Do not place fingers, tools, or other foreign objects into the plug-in ports of your computer or the connectors of the interface. Such actions may result in minor electrical shock hazard and interference with pacemaker devices worn by some persons. Damage to the connectors and internal circuitry of the computer and/or interface may also result.

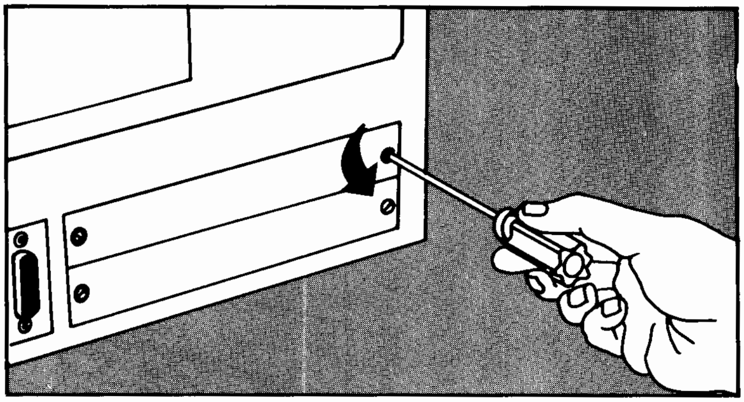
Always turn off the computer and any peripherals connected to it before installing or removing the interface. Failure to observe this procedure may result in damage to the computer, the interface, or both.

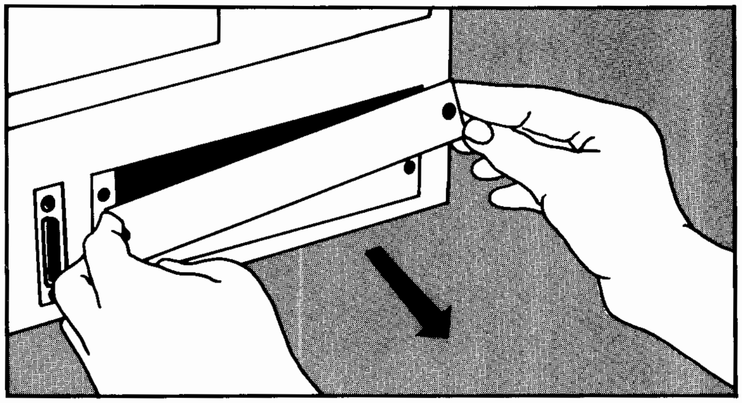
To install the interface, follow these steps:

1. Turn off the computer.

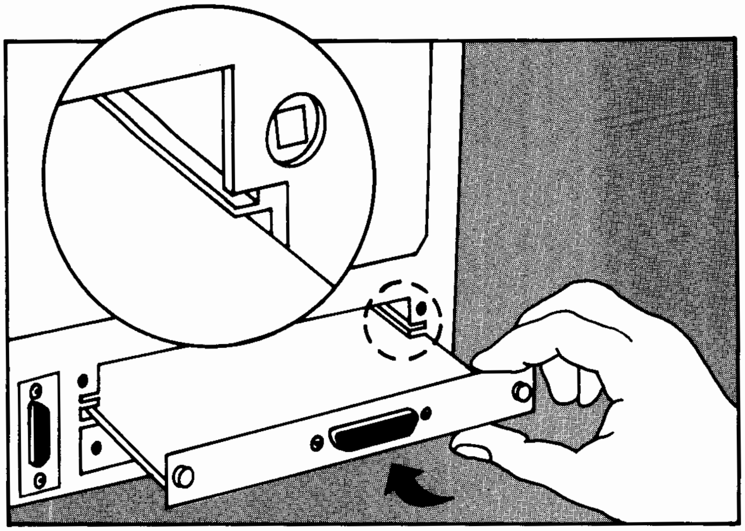


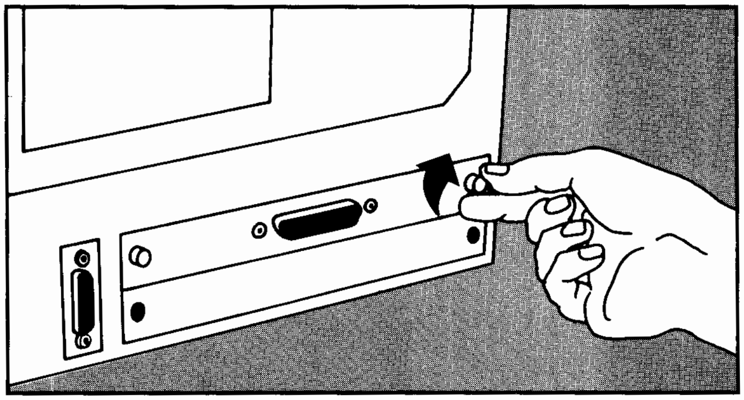
2. Uncover an I/O port (on back of computer).





3. Insert the interface.

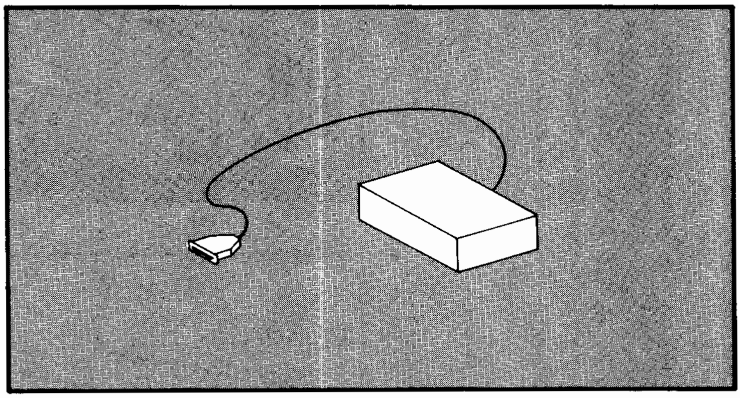




☝ You may install the interface in any unused I/O port. Be sure that the interface is right-side up and that the printed-circuit assembly is aligned in the guides.

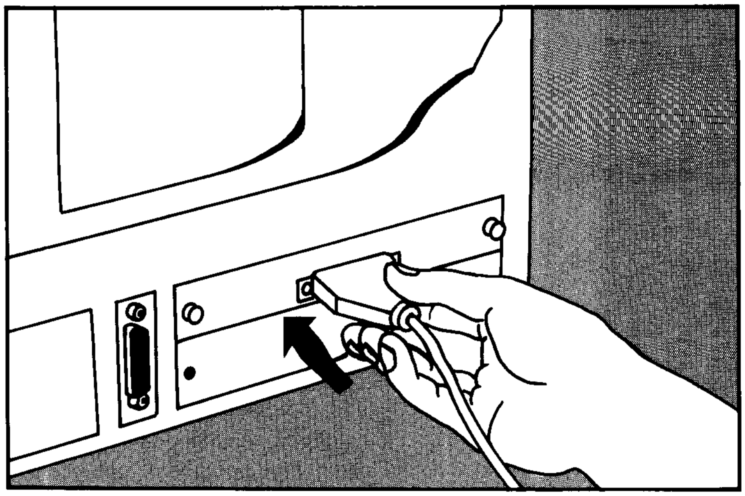
4. Install a cable on your peripheral (refer to "Connecting a Peripheral Device").*

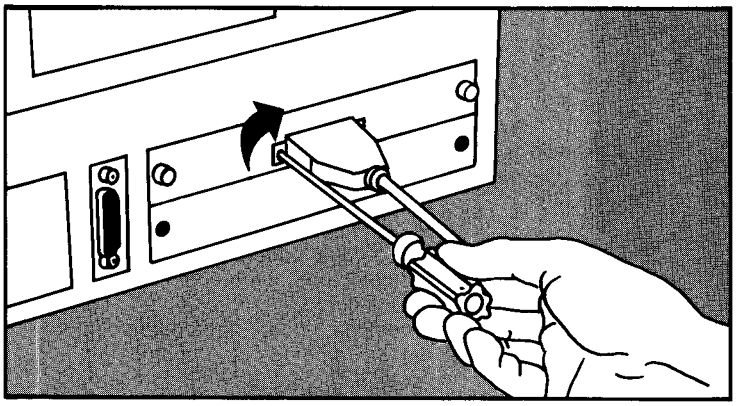
Follow the instructions under "Connecting a Peripheral Device" carefully. Damage to the interface and/or peripheral may result if you connect the cable improperly. Connect the interface only to peripherals that have a current loop transmitter and receiver. *Do not connect the interface to RS-232-C serial peripherals.* Doing so may result in damage to the interface.



* The HP 82975A Current Loop Cable has no connector at the peripheral end.

5. Plug the cable connector into the interface.





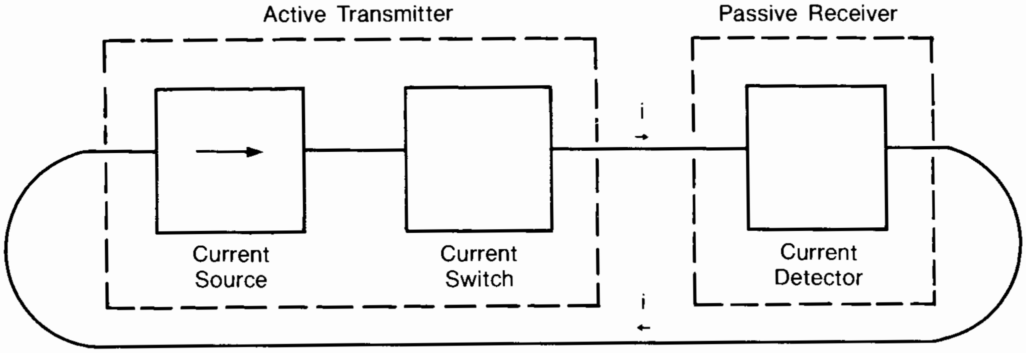
If you remove the interface for an extended period, replace the cover plate over the I/O port.

Current Loop Data Transmission

The current loop interface is similar to an RS-232-C serial interface. Both transmit ASCII encoded data in the bit-serial, asynchronous form, and can operate in the full-duplex, half-duplex, or simplex mode. However, the current loop interface transmits and receives data over two *current loops*. Logic states are represented by the flow of current rather than by voltages. A *space* (logic 0) is represented by a current less than or equal to 3 mA. A *mark* (logic 1) is represented by a current greater than or equal to 10 mA.

To complete a current loop, three components are required: a current source or sink, a current switch, and a current detector. The current source or sink can be located either at the transmitter end or the receiver end of the loop. If the transmitter contains the current source, it is called an *active transmitter*. Figure 1-1 is a block diagram of an active-transmitter current loop:

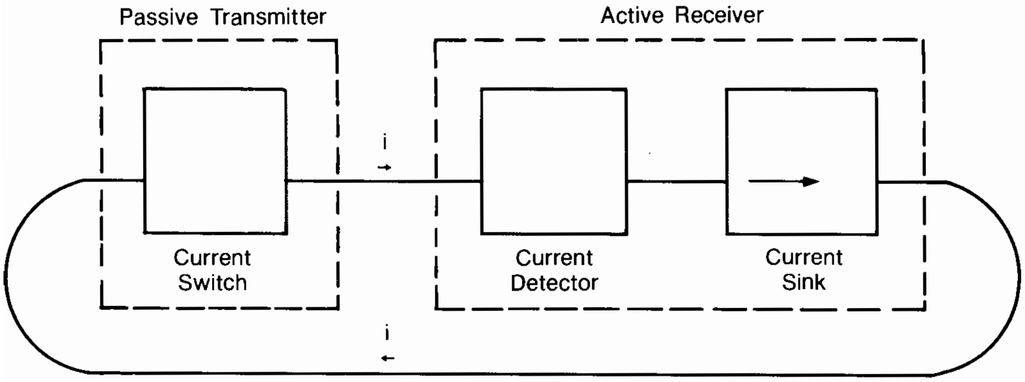
Figure 1-1. Active-Transmitter Current Loop



]

If the receiver contains a current sink, it is called an *active receiver.* Figure 1-2 is a block diagram of an active-receiver current loop:

Figure 1-2. Active-Receiver Current Loop



The current loops of figures 1-1 and 1-2 transmit data in only one direction, providing simplex data transmission. The HP 82920A Current Loop Interface provides both a current loop transmitter and receiver so that you can set up two independent current loops (one for output and one for input). This provides full-duplex capability. To complete the output current loop, connect the transmitter of the interface to the receiver of the peripheral. To complete the input current loop, connect the receiver of the interface to the transmitter of the peripheral.

In many applications a common ground wire is used for both the transmitter and receiver current loops. Thus, only three wires need to be connected.

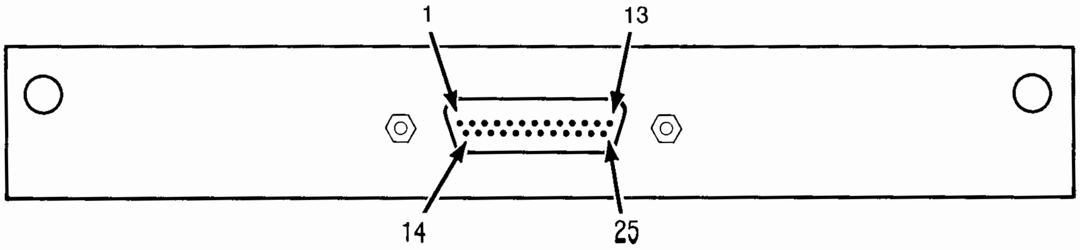
Connecting a Peripheral Device

You can connect the HP 82920A Current Loop Interface to most peripherals that have a current loop transmitter and receiver. Current loop data transmission is commonly used in teletype applications and for some medical instruments. Refer to the owner's manual for your peripheral to determine whether it is compatible with a current loop interface.

The HP 82975A Current Loop Cable plugs into the interface back-panel connector, but does not have a connector at the peripheral end. You need to wire this cable to your peripheral directly. The method of connection depends on your peripheral, but the first step is to identify the terminals of the current loop interface and the corresponding wire colors in the cable.

Figure 1-3 shows the back-panel connector of the current loop interface, and identifies its pins:

Figure 1-3. Back-Panel Connector



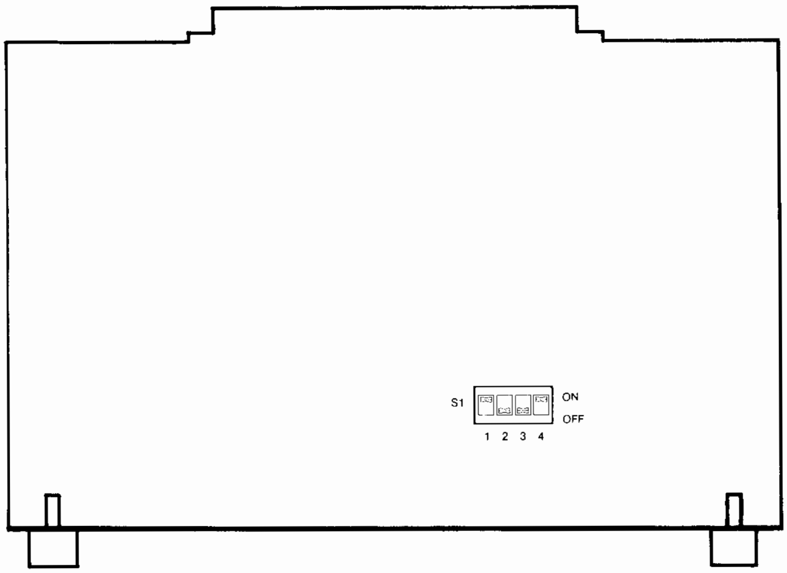
The current loop interface provides two transmitter terminals, two receiver terminals, a signal ground, and a shield ground. Table 1-1 gives the back-panel connector pin number of each terminal and ground connection, along with the corresponding wire color for the HP 82975A Current Loop Cable.

Table 1-1. Current Loop Terminals

Terminal Name	Designation	Pin Number	Wire Color
Transmitter +	+Tx	14	Black
Transmitter −	−Tx	2	Red
Receiver +	+Rx	3	Orange
Receiver −	−Rx	16	Brown
Signal Ground		7	Violet
Shield Ground		1	(Drain)

Switch assembly S1 provides four switches that allow you to configure the current loop transmitter and receiver. Figure 1-4 shows the location of S1 on the interface printed-circuit assembly (top view):

Figure 1-4. Configuration Switches



Switches 1 through 4 of S1 are identified as S1-1, S1-2, S1-3, and S1-4, respectively. These switches are closed in the "ON" position and open in the "OFF" position. They are set at the factory in the following positions:

S1-1	ON (closed).
S1-2	OFF (open).
S1-3	OFF (open).
S1-4	ON (closed).

You can configure the current loop transmitter by setting switches S1-1 and S1-2. Close switch S1-1 to connect terminal +Tx to the 20 mA current source if you need an active transmitter. Close switch S1-2 to connect +Tx to ground, but don't close both switches (S1-1 and S1-2) at the same time. Doing so will ground the 20 mA current source. This will not damage the circuit, but accomplishes no purpose.

You can configure the current loop receiver by setting switches S1-3 and S1-4. Close switch S1-3 to connect terminal −Rx to the 20 mA current sink if you need an active receiver. Close switch S1-4 to connect −Rx to ground, but don't close both switches (S1-3 and S1-4) at the same time. Doing so will ground the 20 mA current sink. This will not damage the circuit, but accomplishes no purpose.

Current Loop Configurations

You can independently configure the transmitter and receiver of the interface to be either active or passive, depending on the requirements of the peripheral:

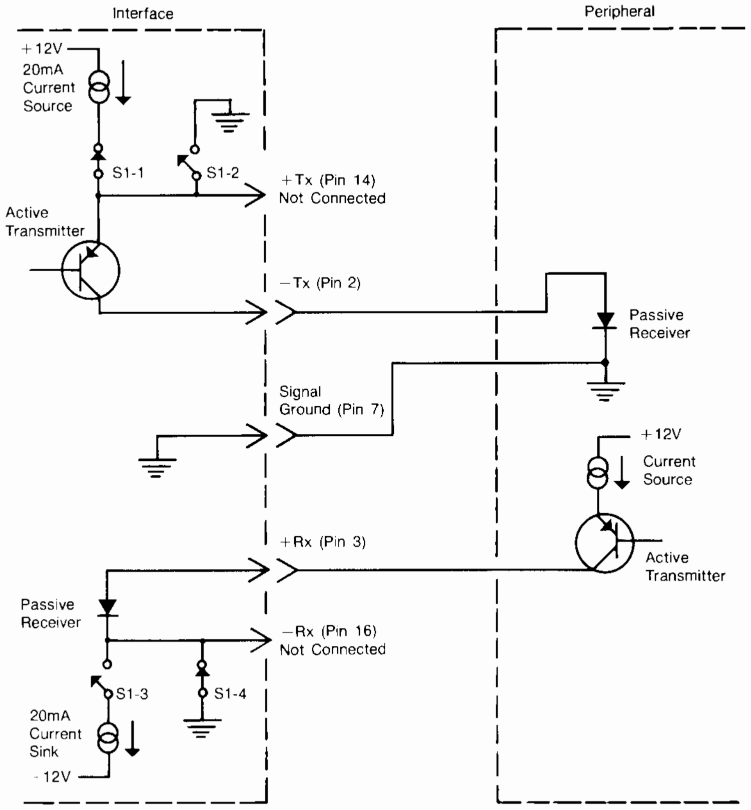
- For the output current loop, if the peripheral has an active receiver, the interface must have a passive transmitter. If the peripheral has a passive receiver, use an active transmitter.
- For the input current loop, if the peripheral has an active transmitter, the interface must have a passive receiver. If the peripheral has a passive transmitter, use an active receiver.

Refer to the owner's manual for your peripheral to determine the needed configuration.

Now let's look at some typical configurations.

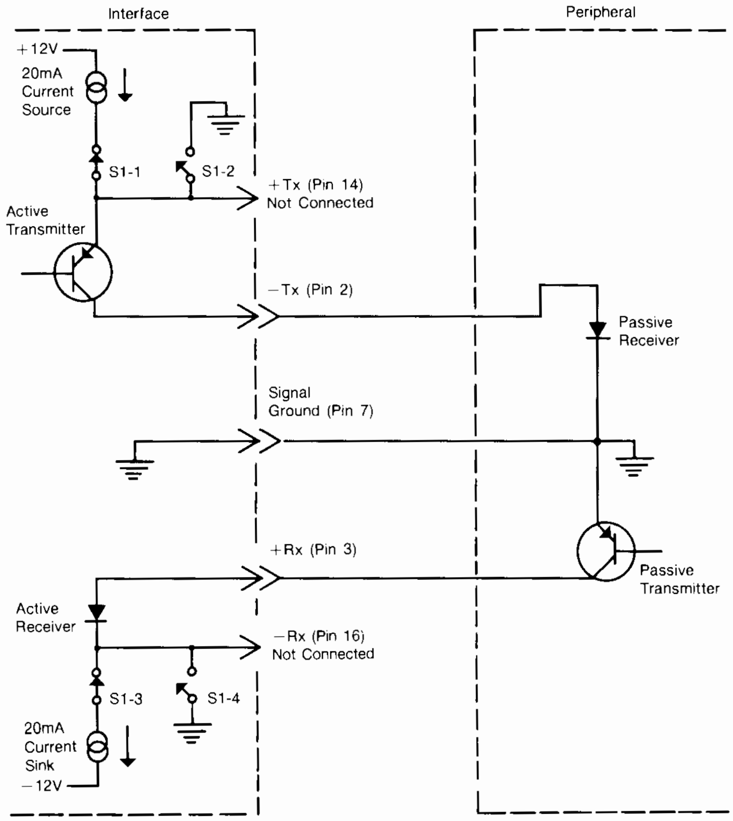
Figure 1-5 shows the interface with switches S1-1 through S1-4 in their factory setings. In this configuration the interface has an active transmitter and passive receiver, and thus can be connected to a peripheral with a passive receiver and active transmitter.

Figure 1-5. Active Transmitter, Passive Receiver



If your peripheral has a passive receiver and transmitter, you need to set the interface to the active transmitter-active receiver configuration. Figure 1-6 shows this configuration:

Figure 1-6. Active Transmitter and Receiver



If your peripheral has an active receiver and transmitter, you need to set the interface to the passive transmitter-passive receiver configuration. Figure 1-7 shows such a configuration with the current loops grounded to the interface through switches S1-2 and S1-4. The signal ground line brings the ground connection back to the peripheral.

Figure 1-7. Passive Transmitter and Receiver (Grounded at Interface)

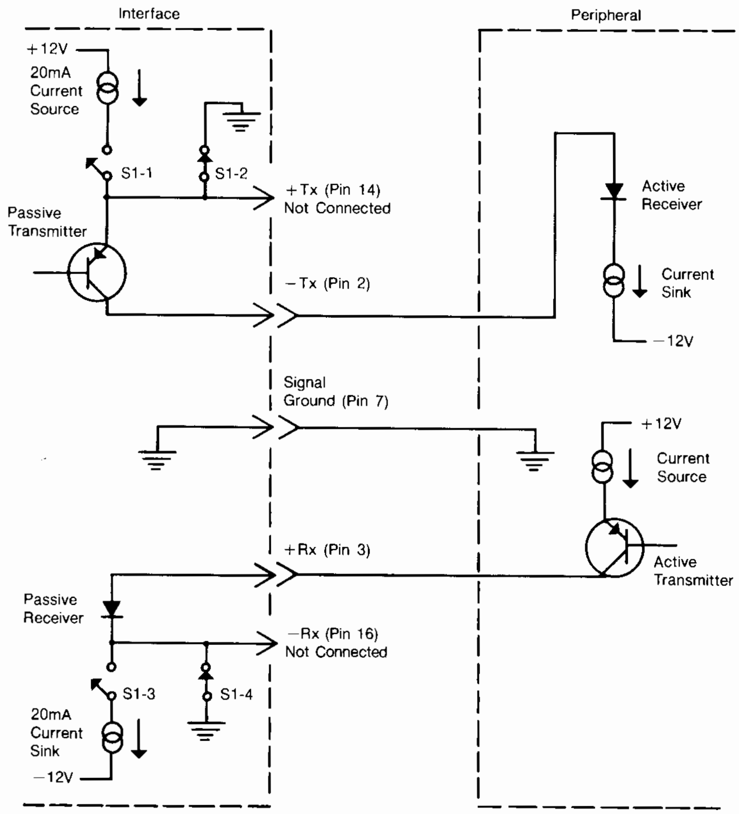
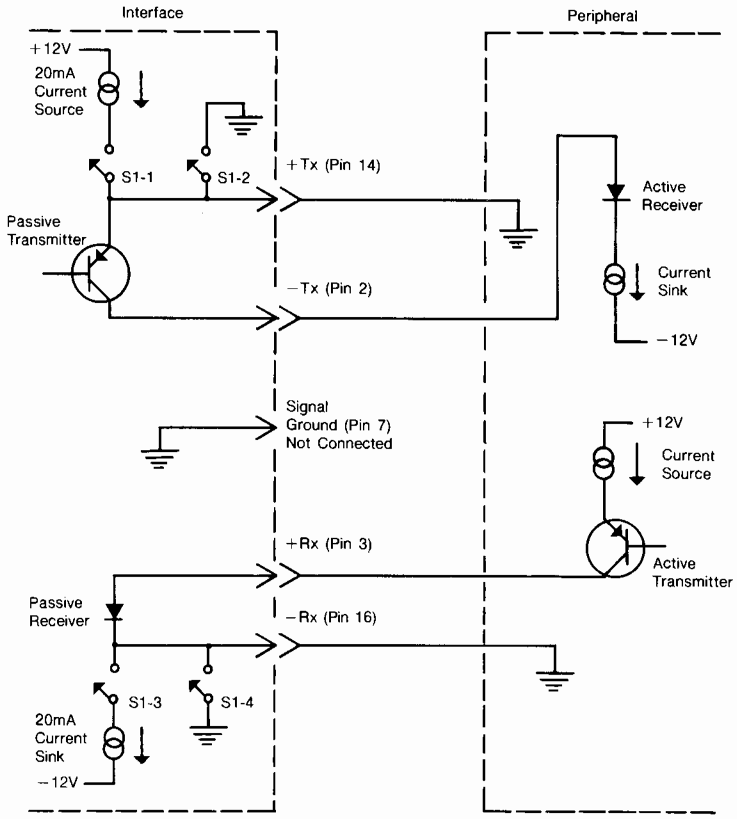


Figure 1-8 shows another passive transmitter-passive receiver configuration. In this case, however, each current loop is grounded separately to the peripheral using the +Tx and −Rx terminals.

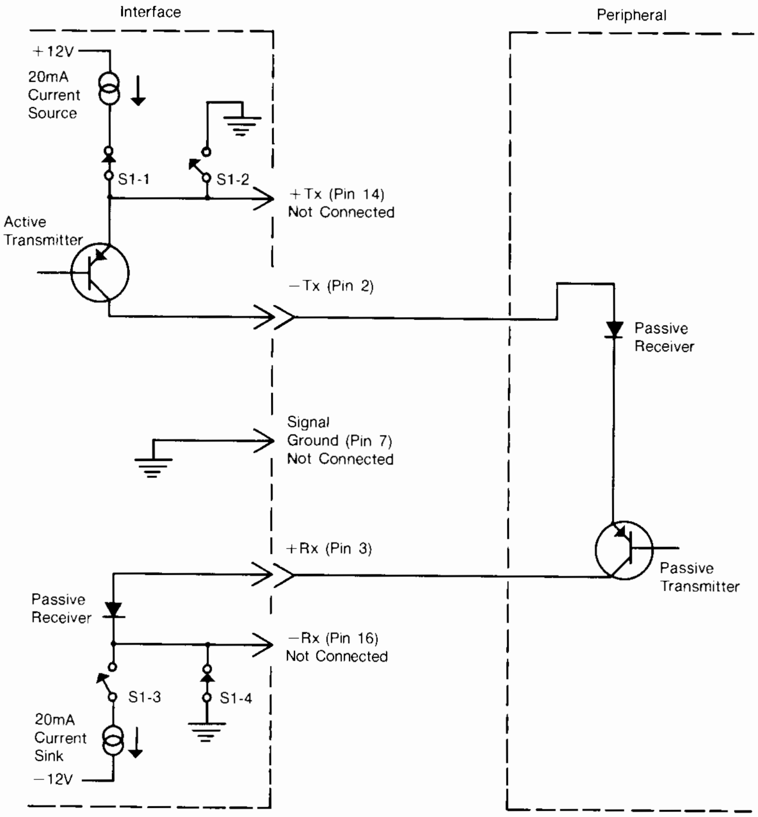
Figure 1-8. Passive Transmitter and Receiver (Grounded at Peripheral)



There are many possible configurations besides those shown. For example, you can configure the interface with a passive transmitter and active receiver, or you can employ a different grounding method. In any case, the configuration that you need depends on your application and your peripheral.

So far we have looked at full-duplex configurations. For half-duplex operation, you can connect the interface transmitter, peripheral receiver, peripheral transmitter, and interface receiver into one current loop. Only one current source or sink is needed. Figure 1-9 shows a typical half-duplex configuration:

Figure 1-9. Half-Duplex Configuration



For simplex operation, of course, only one transmitter and receiver are needed.

Operating Limits

The operating limits of the current loop transmitter and receiver depend on whether the active mode or the passive mode is in use.

In the *active* mode, the current loop transmitter can source up to 20 mA and the current loop receiver can sink up to 20 mA. Each of the 20 mA current supplies can supply 20 mA at up to a 9.9 V compliance (−20 mA into a 495 ohm load). The current loop interface is compatible with all standard 20 mA current loop devices.

In the *passive* mode, the interface can switch or detect up to 60 mA (if the peripheral is capable of sourcing this much current). Opto-isolators are used in both the transmitter and the receiver. The transmitter opto-isolator can switch up to 60 mA in the passive mode, but the off voltage across the switch element should not exceed 15 V. The receiver opto-isolator can detect up to 60 mA in the passive mode.

Where To Go From Here

Once you have installed the current loop interface and connected a peripheral, you need a program to control the input and output of data. The procedure to follow depends upon the program you are using:

- If you are using a Hewlett-Packard data communications program such as "Datacomm," refer to the user's manual supplied with the program.
- If you are writing your own program, refer to your programming language manuals. If you need to access the internal registers of the interface, refer to chapter 2 in this manual.
- If you are writing your own interface driver program, refer to your operating system documentation. Refer to chapter 2 in this manual for a description of the interface hardware, including the internal registers.

The documentation provided with your computer and its peripherals may also help you.

If you suspect a malfunction, refer to the diagnostic procedures in your computer owner's manual. Refer to the support guide included with the interface for service and warranty information.

2 Hardware Description

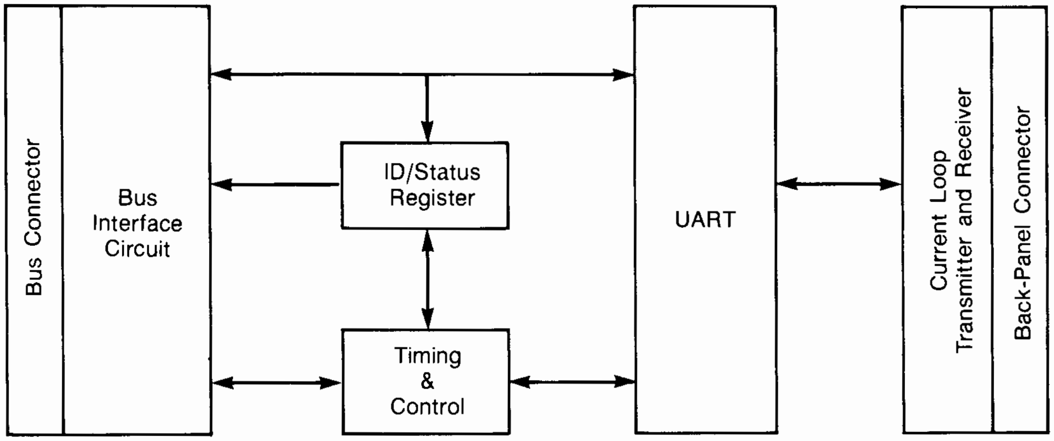
This chapter describes the circuitry and internal registers of the HP 82920A Current Loop Interface. It is intended for the advanced programmer who wants to control the operation of the interface by accessing its internal registers.

If you are using a Hewlett-Packard data communications program such as "Datacomm," you do not need to read this chapter. Follow the instructions in the manual provided with the program. If you are programming in HP-UX Technical BASIC, use the registers described in the *HP-UX Technical BASIC I/O Programming Guide* to control the interface, rather than the registers described here.

Functional Description

Figure 2-1 illustrates the major components of the HP 82920A Current Loop Interface. The interface uses a UART (Universal AsyAsynchronous Receiver Transmitter) integrated circuit to convert transmitted data to serial form, and to convert incoming serial characters to parallel form.

Figure 2-1. Interface Block Diagram



Let's look at each component in turn.

Bus Connector and Interface Circuit

The interface bus connector plugs into the I/O backplane connector on your computer. The bus interface circuit decodes addresses, generates and responds to handshaking signals, and provides data buffering.

Identification and Status Register

The identification and status register is a read/write register that identifies the current loop interface and controls its capability to interrupt the computer. The timing and control circuit controls access to this register. This register is covered in detail in the "Registers" section later in this chapter.

Timing and Control Circuit

The timing and control circuit generates the signals necessary to control the internal components of the HP 82920A Current Loop Interface (the bus interface circuit, identification and status register, and the dual channel UART).

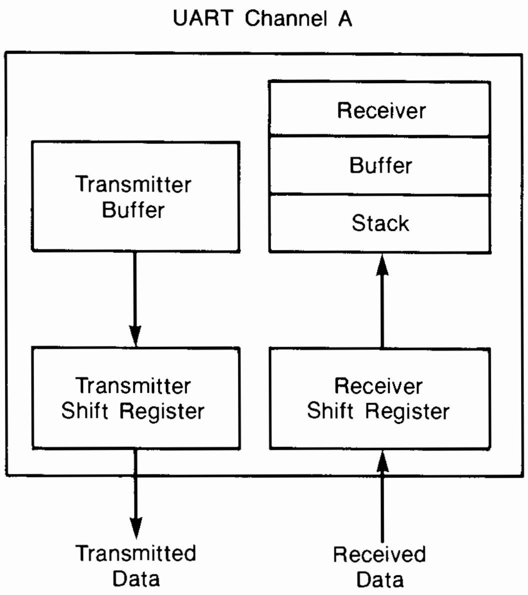
UART

The major component of the HP 82920A Current Loop Interface is the UART integrated circuit. The UART provides two data channels (channel A and channel B), but only channel A is used. We will refer to channel A as the *data channel*. The UART also provides several registers. You can control the operation of the interface by accessing these registers (refer to "Registers").

Data Channel. The data channel consists of a transmitter and receiver. You can control the operation of the data channel by accessing the channel A mode registers, command register, and status register. You can enable the transmitter and receiver independently, and you can operate them at different baud rates (in the normal operating mode). For full duplex operation, enable both the transmitter and the receiver.

The data channel has a transmitter buffer and shift register, and a receiver buffer and shift register. Figure 2-2 illustrates the operation of the transmitter and receiver:

Figure 2-2. Data Channel



Your program can output data by writing to the transmitter buffer. The data is then transferred to the transmitter shift register, which converts the data to serial form. Incoming serial data is converted to parallel form by the receiver shift register, then transferred to the receiver buffer (a three-character stack). The program can then read the data from the receiver buffer. The following discussion covers transmitter and receiver operation in detail.

Transmitter Operation. The transmitter, when enabled, indicates that it is ready to accept a character by setting the "transmitter ready" bit (bit 2 of Read Register 5) equal to "1". You can enable an interrupt for the "transmitter ready" condition. This interrupt is negated when a character is loaded into the transmitter buffer. The character is transferred from the transmitter buffer to the transmitter shift register once the shift register has completed transmission of the previous character. At this point "transmitter ready" is again asserted. Characters cannot be written into the transmitter buffer if the transmitter is disabled.

The transmitter shift register converts the parallel data from the computer to a serial bit stream on the Transmitted Data output. The transmitter automatically sends a start bit, followed by the programmed number of data bits (least-significant bit first), an optional parity bit, and the programmed number of stop bits. Each bit is transmitted as either a *space* (logic 0) or a *mark* (logic 1). If a new character is not available in the transmitter buffer when the stop bits have been sent, the Transmitted Data output remains in the marking (logic 1) state and the "transmitter empty" bit in the status register is set equal to "1". The "transmitter empty" bit is cleared and transmission resumes when a new character is written into the transmitter buffer by the computer. You can force the transmitter into a continuous spacing (logic 0) condition by issuing a "send break" command.

The Transmitted Data output of the data channel is connected to the current loop transmitter of the interface (refer to chapter 1). Outgoing data from the transmitter shift register is sent out on the output current loop.

Receiver Operation. The receiver, when enabled, searches for a valid start bit. When a start bit is detected, the programmed number of data bits, an optional parity bit, and the programmed number of stop bits are read into the receiver shift register. Each bit is read as either a space (logic 0) or a mark (logic 1). The shift register converts the data into parallel form, the data is transferred to the receiver buffer, and the "receiver ready" bit (bit 0 of Read Register 5) is set equal to "1".

The receiver buffer is a first-in-first-out (FIFO) stack with room for three characters. The first character transferred into the stack by the shift register is the first character read by the computer. Thus, the order of the incoming data is preserved, but buffering for four full characters is provided by the buffer and shift register. The "receiver ready" bit is cleared when the last character in the receiver buffer is read.

If the receiver buffer and receiver shift register are both full and a new character is received, the character in the shift register is overwritten by the new character and the "overrun error" bit is set in the status register. If a parity error, framing error, or received break is detected in the incoming data, the appropriate status bit is set.

The Received Data input of the data channel is connected to the current loop receiver of the interface (refer to chapter 1). Incoming data from the current loop receiver is input to the receiver shift register.

Baud Rate Generator and Timer. The UART provides both a baud rate generator and a timer. The baud rate generator provides 18 fixed baud rates. You can also use the timer to provide any desired baud rate in the operating range of the interface (up to 19,200). This baud rate is determined by the divisor contained in Registers 13 and 14. You can independently select baud rates for the transmitter and receiver by writing to the Channel A Baud Rate Select register (Write Register 5). Although you can use two different baud rates, only one of them can be provided by the timer.

Operating Modes. The data channel can be programmed to operate in one of four operating modes. You can specify the operating mode with bits 6 and 7 of Register 2.

The first mode is the *normal operating mode*, which allows the data channel to operate in either simplex, half-duplex, or full duplex. In this mode the transmitter and receiver operate independently.

The second mode is the *auto-echo mode*. In this mode the received data can be read by the computer, but it is also automatically retransmitted. The following conditions are true in this mode:

1. All received data is reclocked and transmitted by the transmitter.
2. The baud rate selected for the receiver is automatically used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be.
4. The "transmitter empty" and "transmitter ready" status bits are inactive.
5. Parity is checked if in use, but parity bits are just transmitted as received (not regenerated).
6. Character framing is checked, but the stop bits are just transmitted as received (not regenerated).
7. If a break is received, it is echoed just as received until the next valid start bit is received.
8. In auto-echo mode the computer cannot transmit. It can only receive.

The third mode of operation is a diagnostic mode known as *local loop-back mode.* The following conditions are true in this mode:

1. The output of the transmitter is internally connected to the input of the receiver.
2. The baud rate selected for the transmitter is also used for the receiver.
3. The transmitter must be enabled, but the receiver need not be.
4. The Transmitted Data output is held in the marking state.
5. The Received Data input is ignored.
6. The computer can send data to the transmitter and should receive the same data back from the receiver.

The fourth mode of operation, the *remote loop-back mode,* is also a diagnostic mode. The following conditions are true in this mode:

1. Data received from an external source is reclocked and transmitted back to the external source by the transmitter.
2. The baud rate selected for the receiver is also used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be.
4. The computer does not have access to the received data and the status register error bits are inactive.
5. The received parity is not checked or regenerated. Parity bits are just transmitted as received.
6. Character framing is not checked. Stop bits are just transmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

Current Loop Transmitter and Receiver

The UART data channel transmitter outputs data through the current loop transmitter, while the data channel receiver inputs incoming data from the current loop receiver. The current loop transmitter and receiver are covered in chapter 1.

Registers

You can program the operation of the HP 82920A Current Loop Interface by accessing its hardware registers. *The procedures for reading or writing to a register depend on your operating system and programming language. Refer to your operating system and language documentation for the correct statements.* Whatever the access method, you will need to identify each register with its least-significant address byte. These address bytes are given in the register descriptions that follow.

If you accidentally specify the wrong least-significant address byte, you may access an illegal register address. You may lose some data if you do this, but no damage to the interface or computer will result. You can return your system to normal operation by performing a hardware reset.

The HP 82920A Current Loop Interface and the HP 82919A Serial Interface both use the same dual-channel UART integrated circuit. However, the current loop interface uses only channel A, while the serial interface uses channels A and B.

The registers of the current loop interface are equivalent to the corresponding registers of the serial interface, and have the same numbers. This allows program compatibility between the two interfaces. However, the current loop interface does not implement all of the registers of the serial interface (for example, the channel B registers). Thus, the registers of the current loop interface are not numbered consecutively.

Register 0: Identification and Status

Register 0 contains the indentification bits of the HP 82920A Current Loop Interface, and also indicates whether interrupts are enabled, the level at which the interface can interrupt the computer, and whether an interrupt is pending.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Interrupt Enable	Interrupt Pending	Interrupt Level		Identification Bits "0010"			

To access Register 0:

- Least-significant byte of address = 01 (hexadecimal).
- Read and write access is allowed for bits 4 through 7. Bits 0 through 3 are read only.

Bits 0 Through 3: Identification. These bits are permanently set to "0010" (2), the identification code for the HP 82920A Current Loop Interface.

Bits 4 and 5: Interrupt Level. These bits specify the level at which the interface can issue an interrupt to the computer. The following table gives the priority levels specified by various settings of these bits (in order of increasing priority):

Table 2-1. Interrupt Level

Bit 5	Bit 4	Interrupt Priority Level
0	0	0 (lowest)
0	1	1
1	0	2
1	1	3 (highest)

Bit 6: Interrupt Pending. This bit, when equal to "1", indicates that an interrupt is pending. Bit 6 is a read/write bit. You can force a hardware interrupt by setting this bit to a "1" (provided interrupts are enabled).

Bit 7: Interrupt Enable. This bit, when set to a "1", enables the interrupting capability of the interface. Pending or forced interrupts are ignored when bit 7 is equal to "0".

Register 1: Channel A Mode Register I

This register and Register 2 are both accessed by addressing hexadecimal 39. The channel A mode register pointer determines which register is accessed. You can set the pointer to mode register I by writing a "reset mode register pointer" command to the channel A command register (Write Register 7). A hardware reset also resets the pointer. The pointer advances to mode register II after mode register I has been accessed (read or write).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used	Receiver Interrupt Select	Error Mode	Parity Mode		Parity Type	Data Bits Per Character	

To access Register 1:

- Least-significant byte of address = 39 (hexadecimal).
- The mode register pointer must be set at mode register I.
- The data channel transmitter and receiver must be reset before you change the contents of this register.
- Read and write access is allowed.

Bits 0 and 1: Data Bits Per Character. This field specifies the number of data bits per character:

Table 2-2. Data Bits Per Character

Bit 1	Bit 0	Number of Data Bits
0	0	5
0	1	6
1	0	7
1	1	8

Bit 2: Parity Type. Bit 2 specifies parity type. The effect of this bit depends upon the parity mode specified with bits 3 and 4. If "with parity" mode is selected, bit 2 equal to "0" specifies even parity, while "1" specifies odd parity. In "force parity" mode, bit 2 equal to "0" specifies "always zero" parity, while "1" specifies "always 1" parity.

Bits 3 and 4: Parity Mode. These bits specify parity mode. If "with parity" or "force parity" is selected, a parity bit is added to each transmitted character and the receiver performs a parity check on the incoming data.

Table 2-3. Parity Mode

Bit 4	Bit 3	Parity Mode
0	0	With Parity
0	1	Force Parity
1	0	No Parity

Bit 5: Error Mode. This bit selects the operating mode for the "framing error," "parity error," and "received break" bits in the channel A status register (Read Register 5). If bit 5 is equal to "0", the *character* mode is selected and status is provided on a character-by-character basis. If bit 5 is a "1", *block* mode is selected. In block mode the three error bits indicate the logical OR of the status for all characters that have reached the top of the channel A receiver buffer since the last "reset error status" command.

Bit 6: Receiver Interrupt Select. This bit selects the condition that results in a receiver interrupt. If this bit is equal to "0", an interrupt is generated on the channel A "receiver ready" condition. If this bit is equal to "1", an interrupt is generated on a channel A "receiver buffer full" condition (refer to Read Register 12).

Bit 7: Not Used. This bit has no effect. Set bit 7 equal to "0".

Register 2: Channel A Mode Register II

This register, like Register 1, is accessed by addressing hexadecimal 39. The mode register pointer moves to mode register II after mode register I has been accessed (read or write). Accessing mode register II does not change the pointer.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Mode Select		Not Used		Stop Bit Length			

To access Register 2:

- Least-significant byte of address = 39 (hexadecimal).
- The mode register pointer must be set at mode register II.
- The data channel transmitter and receiver must be reset before you change the contents of this register.
- Read and write access is allowed.

Bits 0 Through 3: Stop Bit Length. These bits specify stop bit length.* You can specify a stop bit length of either 1.0 or 2.0 if you have specified six, seven, or eight data bits per character (refer to Register 1). If you have specified five data bits per character, use a stop bit length of 1.5. Table 2-4 gives the allowable selections:

Table 2-4. Stop Bit Length

Value Written to Bits 3–0	Data Bits Per Character	Stop Bit Length
0111	6, 7, or 8	1.0
0111	5	1.5
1111	6, 7, or 8	2.0

Bits 4 and 5: Not Used. These bits have no effect. Set each bit equal to "0".

* You should always set bits 0, 1, and 2 equal to "1". Bit 3 can be equal to either "0" or "1".

Bits 6 and 7: Mode Select. These bits select the operating mode of the data channel as shown in table 2-5. Refer to "Operating Modes" for a description of each of the four modes.

Table 2-5. Operating Modes

Bit 7	Bit 6	Operating Mode
0	0	Normal
0	1	Auto Echo
1	0	Local Loop Back
1	1	Remote Loop Back

Read Register 5: Channel A Status

This register provides the current status of the data channel transmitter and receiver. The contents of this register are cleared to zero by a hardware reset.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Received Break	Framing Error	Parity Error	Overrun Error	Transmitter Empty	Transmitter Ready	Receiver Buffer Full	Receiver Ready

To access Read Register 5:

- Least-significant byte of address = 3B (hexadecimal).
- Read-only access is allowed.

Bit 0: Receiver Ready. This bit, when equal to "1", indicates that a character has been received and is waiting in the receiver buffer to be read by the computer. Bit 0 is cleared to "0" after the last character is read from the buffer.

Bit 1: Receiver Buffer Full. This bit, when equal to "1", indicates that all three character positions in the receiver buffer are occupied. Reading the receiver buffer clears this bit to "0" (unless another character is waiting in the receiver shift register).

Bit 2: Transmitter Ready. This bit, when equal to "1", indicates that the transmitter buffer is empty. This bit is cleared when a character is written to the transmitter buffer, and is set when the character is transferred to the transmitter shift register.

Bit 3: Transmitter Empty. This bit is set equal to "1" when the transmitter underruns. That is, when both the transmitter buffer and transmitter shift register are empty.

Bit 4: Overrun Error. This bit, when set, indicates that the receiver has overrun. This occurs when the receiver buffer is filled with three characters, a fourth character is in the receiver shift register, and the start bit of a fifth character is received. The character in the shift register (the fourth character) is lost.

Bit 5: Parity Error. If parity is enabled, this bit is set equal to "1" when a character is received with incorrect parity.

Bit 6: Framing Error. This bit, when set, indicates that a stop bit was not detected in an incoming character.

Bit 7: Received Break. This bit, when set, indicates a break condition has been detected. A break is indicated when an all zero character of the programmed length is received without a stop bit. When this bit is set, the "change in break" bit in the Interrupt Status register (Read Register 12) is also set.

If Read Register 5 is read when there are no characters in the receiver buffer (bit 0 = 0), bits 5 through 7 ("parity error," "framing error," and "received break") remain in whatever state they were in for the last character received. These bits are updated when a new character is placed in the receiver buffer.

Write Register 5: Channel A Baud Rate Select

This register selects the baud rate for the transmitter and receiver.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Receiver Baud Rate Select				Transmitter Baud Rate Select			

To access Write Register 5:

- Least-significant byte of address = 3B (hexadecimal).
- Write-only access is allowed.

Bits 0 Through 3: Transmitter Baud Rate Select. You can select one of 18 standard baud rates for the data channel transmitter by writing to this field, or you can specify the timer as the baud rate source. To specify a baud rate, write the appropriate value to bits 0 through 3 of this register and to bit 7 of the Auxiliary Control register (Write Register 11), as shown in table 2-6:

Table 2-6. **Baud Rate Selection**

Baud Rate Select Register Bits 3–0	**Baud Rate**	
	Auxiliary Control Register Bit 7 = 0	**Auxiliary Control Register Bit 7 = 1**
0000	50	75
0001	110	110
0010	134.5	134.5
0011	200	150
0100	300	300
0101	600	600
0110	1200	1200
0111	1050	2000
1000	2400	2400
1001	4800	4800
1010	7200	1800
1011	9600	9600
1100	38,400	19,200
1101	Timer	Timer

If you select the timer (binary 1101), the baud rate is determined by the 16-bit divisor contained in Registers 13 and 14. This allows you to select a non-standard baud rate.

Although you can select a baud rate of 38,400, operation at this rate is not supported.

Bits 4 Through 7: Receiver Baud Rate Select. This field selects the receiver baud rate (in conjunction with bit 7 of the auxiliary command register). The baud rate selections are exactly as listed for the transmitter in table 2-6, except that the control bits are written to the upper four bits rather than the lower four bits of the register.

Write Register 7: Channel A Command

You can use this register to send commands to the data channel (UART channel A). You can specify multiple commands in a single write to the register as long as the commands are not in conflict with each other.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used (Always "0")	Miscellaneous Commands			Disable Transmitter	Enable Transmitter	Disable Receiver	Enable Receiver

To access Write Register 7:

- Least-significant byte of address = 3D (hexadecimal).
- Write-only access is allowed.

Bit 0: Enable Receiver. Setting this bit enables operation of the data channel receiver and causes the receiver to initiate a search for a start bit.

Bit 1: Disable Receiver. Setting this bit terminates operation of the data channel receiver immediately—a character being received will be lost. The command has no effect on the receiver status bits or any other control registers.

Bit 2: Enable Transmitter. Setting this bit enables the data channel transmitter and asserts the "transmitter ready" status bit.

Bit 3: Disable Transmitter. Setting this bit terminates data channel transmitter operation and resets the "transmitter ready" and "transmitter empty" status bits. However, the transmitter will finish sending any characters currently in the transmitter buffer and shift register before becoming inactive.

Bits 4 Through 6: Miscellaneous Commands. The value of this three-bit field specifies a single command. The values and commands are listed in the following table:

Table 2-7. Miscellaneous Commands

Bits 6–4	Command
000	*No command.*
001	*Reset mode register pointer.* Resets the mode register pointer to mode register I.
010	*Reset receiver.* Disables the receiver and clears the receiver buffer (as if a hardware reset has occurred).
011	*Reset transmitter.* Resets the transmitter as if a hardware reset has occurred.
100	*Reset error status.* Clears the "received break," "parity error," "framing error," and "overrun error" bits in the Channel A Status register.
101	*Reset break change interrupt.* Clears the "change in break" bit in the Interrupt Status register (Read Register 12).
110	*Start break.* Forces the transmitter data output to the spacing state. If the transmitter is active, the break will be delayed until any characters in the transmitter buffer or shift register have been sent. The transmitter must be enabled for this command to be accepted.
111	*Stop break.* Forces the transmitter data output to the marking state. This output will remain in the marking state for one bit time before the next character, if any, is sent.

Bit 7: Unused. This bit *must* always remain equal to "0". Setting this bit may cause unpredictable UART operation.

Write Register 11: Auxiliary Control

This register provides a set of auxiliary commands that you can use to control the UART.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Baud Rate Select	Timer Clock Source			Not Used			

To access Write Register 11:

- Least-significant byte of address = 31 (hexadecimal).
- Write-only access is allowed.

Bits 0 Through 3: Not Used. These bits have no effect. Set these bits equal to "0".

Bits 4 Through 6: Timer Clock Source. This field selects the clock source that drives the timer. You may select either the crystal oscillator or the crystal oscillator divided by 16 as the clock source:

Table 2-8. Timer Clock Source

Bits 6–4	Clock Scource
110	Oscillator divided by 1.
111	Oscillator divided by 16.

Bit 7: Baud Rate Select. This bit is used in conjunction with the Channel A Baud Rate Select register (Write Register 5) to select baud rates. Refer to Write Register 5 for the available baud rates.

Read Register 12: Interrupt Status

When an interrupt occurs, read this register to determine its cause. When a bit in this register becomes set, an interrupt will result, but only if the corresponding bit in the Interrupt Mask register (Write Register 12) has been set.

A hardware reset clears all bits in this register.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used (Always "0")				Not Used (Ignore)	Change In Break	Receiver Ready or Receiver Buffer Full	Transmitter Ready

To access Read Register 12:

- Least-significant byte of address = 33 (hexadecimal).
- Read-only access is allowed.

Bit 0: Transmitter Ready. This bit is a duplicate of bit 2 of the Channel A Status register (Read Register 5).

Bit 1: Receiver Ready or Receiver Buffer Full. The function of this bit is programmed by bit 6 of Channel A Mode Register I (Register 1). Bit 1 can be either a duplicate of bit 0 ("receiver ready") or bit 1 ("receiver buffer full") of the Channel A Status register (Read Register 5).

Bit 2: Change In Break. This bit is set when the receiver detects the begining or end of a received break. It is reset by the "reset break change interrupt" command (refer to Write Register 7, the Channel A Command register).

Bit 3: Not Used. Ignore the value of this bit (usually "1").

Bits 4 Through 7: Not Used. These bits are always equal to "0".

Write Register 12: Interrupt Mask

You can enable interrupts for the bits in the Interrupt Status register by setting the corresponding bits in the Interrupt Mask register. If the bit in the Interrupt Mask register is equal to "1", an interrupt will occur when the corresponding bit in the Interrupt Status register becomes set.

A hardware reset clears all bits in this register.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not Used (Always "0")					Change In Break Interrupt	Receiver Ready or Receiver Buffer Full Interrupt	Transmitter Ready Interrupt

To access Write Register 12:

- Least-significant byte of address = 33 (hexadecimal).
- Write-only access is allowed.

Bit 3 of this register must always be equal to "0". If you set this bit, an interrupt may occur on every clock cycle!

Registers 13 and 14: Timer Divisor Latch

These registers together make up the 16-bit divisor latch for the timer. Register 13 contains the *most-significant byte;* Register 14, the *least-significant byte.*

To access Registers 13 and 14:

- Register 13: Least-significant address byte = 35 (hexadecimal).

- Register 14: Least-significant address byte = 37 (hexadecimal).
- Read and write access is allowed for both registers. However, reading these registers will produce the current count, not the original value. Reading these registers is not recommended.

You can use the timer to provide any desired baud rate in the operating range of the interface. The baud rate is determined by the 16-bit divisor contained in Registers 13 and 14, which divides the frequency of the timer's clock source. Since you may specify either the crystal oscillator divided by 1 or divided by 16 as the clock source for the timer, there are two equations for determining the divisor. If you are using the oscillator divided by 1, the equation is:

$$\text{Divisor} = 115{,}200 \div \text{Baud Rate}$$

If you are using the oscillator divided by 16, the following equation should be used:

$$\text{Divisor} = 7{,}200 \div \text{Baud Rate}$$

In either case, you should separate the 16 bits of the divisor into two bytes. Write the most-significant byte to Register 13 and the least-significant byte to Register 14. Calculate the values to write to these registers as follows:

$$\text{Register 13 value} = \text{Integer part of (Divisor} \div 256)$$

$$\text{Register 14 value} = \text{(Divisor) modulo 256}$$

If new values are written to Registers 13 and 14, the baud rate will change to the new rate once the current half cycle of the timer is completed.

A hardware reset will cause the timer to stop. You must execute a "start timer" command after a hardware reset. To do this, simply read address byte 25 (hexadecimal). A value of FF (hexadecimal) is returned and the timer starts.

Read Register 15: Channel A Receiver Buffer

The receiver buffer is the three-character first-in-first-out (FIFO) stack described under the subheading "Receiver Operation." Incoming data enters the FIFO stack from the receiver shift register. When you read this register, the character at the top of the stack (the first received) is returned. If the stack is empty, the value FF (hexadecimal) is returned.

To access Read Register 15:

- Least-significant byte of address = 3F (hexadecimal).
- Read-only access is allowed.

Write Register 15: Channel A Transmitter Buffer

The transmitter buffer is described under the subheading "Transmitter Operation." Data written to this register is transferred to the transmitter shift register, serialized, and transmitted.

To access Write Register 15:

- Least-significant byte of address = 3F (hexadecimal).
- Write-only access is allowed.