could be adjusted to read up to 800 cps. The cycle time at this speed is 1.25 milliseconds. The time lapse between the start and end of each of the signals, (data, sprocket and busy), and the time lapse between the various signals also differ widely between the different models.

Although the interaction of the control signals and the timing aspects of these signals differ between the various makes and models of paper tape readers and interfaces, the principle of the sequence of parallel data transfer remains the same.

5.3 IBM Personal Computer Printer Interface

Before it is possible to explain the method of interfacing an IBM personal computer or compatible (PC) to an NC machine controller, it is necessary to describe certain aspects of the PC's hardware.

The printer interface has a 25-pin, D-shell connector mounted on it. The connector protrudes through the rear of the panel of the system or expansion unit, where a cable may be attached. (See Figure 5.4).

Each of the 25 pins in the D-shell connector is controlled by an area in the PC's internal memory. By placing 8-bit binary values in certain memory locations, pins in the D-shell connector can be loaded with either 0V or 5V.

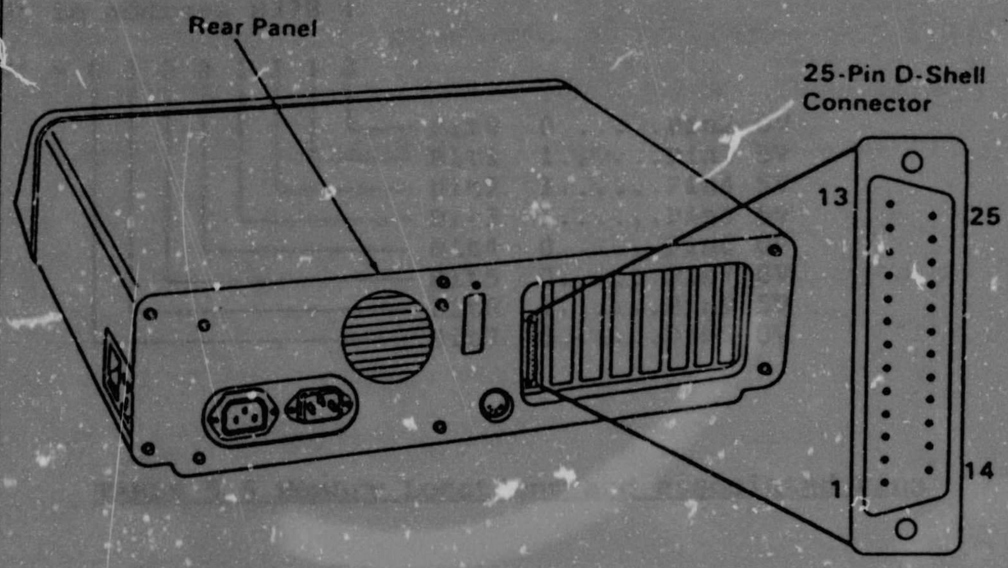


Fig 5.4 The Printer Interface and Connector

Conversely, by reading 8-bit binary values from certain other memory locations, voltages placed by external devices on the pins of the D-shell connector can be "read in". These input voltages supplied by the external device must be either 0V or 5V. The memory locations and associated pins used for the purposes of this project are listed in Table 5.5.

To illustrate the above process, consider the character "N" and it's associated 8-bit binary value of 0100 1110. If this value is loaded into hexadecimal address 378 (address H379), pins 2-9 on the D-shell connector are loaded with the following voltages :

"N" in address H378 :

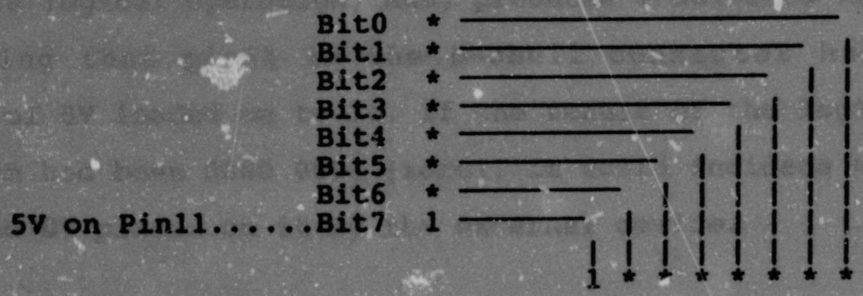
"N" = 0 1 0 0 1 1 1 0

```
Bit0  0......Pin2  0V
Bit1  1......Pin3  5V
Bit2  1......Pin4  5V
Bit3  1......Pin5  5V
Bit4  0......Pin6  0V
Bit5  0......Pin7  0V
Bit5  1......Pin8  5V
Bit7  0......Pin9  0V
```

TABLE 5.5 Memory Locations and associated Pins

<u>Address H378</u>	Bit0	Pin2
(Output to	Bit1	Pin3
external	Bit2	Pin4
device only)	Bit3	Pin5
	Bit4	Pin6
	Bit5	Pin7
	Bit6	Pin8
	Bit7	Pin9
<u>Address H37A</u>	Bit0	Pin1
(Output to		
external		
device only)		
<u>Address H379</u>	Bit7	Pin11
(Input from		
external		
device only)		

To illustrate the "reading" of voltages or input of voltages on pins in the D-shell connector, consider a voltage of 5V placed on pin 11 by and external device :



Address H379 : 8-bit binary value with bit7 set to 1

The character representing the 8-bit binary value in the above example will alter depending on the input voltages on the other seven pins associated with Address H379. In order to ascertain what voltage (either 0V or 5V) has been placed on pin11, it is necessary to extract the value of bit7 from the binary value in Address H379. To do this, the binary value in Address H379 must be "logically AND'ed" with the binary value 1000 0000. For example, assume the value of Address H379 was 1101 1000 :

	1101 1000	Value of Address H379
AND	<u>1000 0000</u>	
	1000 0000	

The result is the 8-bit binary value with all bit positions set to zero except bit7 which has the value of bit7 in the original 8-bit value.

The above logical operation, AND, produces a non-zero value indicating that pin11 in the D-shell connector had a voltage of 5V loaded on to it. If the result of the logical operation had been 0000 0000 (zero), it would indicate that pin11 had 0V placed on it by the external device.

5.4 Simulating Paper Tape Readers Using PC's

In section 5.3 it was explained that by loading the 8-bit binary value of the ASCII character "N" into address H378 produced a set of 8 TTL compatible voltages on pins 2-9 on the D-shell connector. These 8 voltages exactly match the 8 voltages produced by the paper tape reader in figure 5.2 when reading the character "N" from the paper tape. This is how the BTR (Behind Tape Reader) interface functions. Pins 2-9 on the D-shell connector are connected to tracks 1-8 on the paper tape reader interface after the amplification stage. By writing the 8-bit binary values of ASCII characters to address H378, the PC can emulate a paper tape reader by supplying the NC controller with TTL compatible data.

In order to supply the NC controller with a sprocket signal, pin1 on the D-shell connector is connected to the sprocket track on the paper tape reader interface after the amplification stage. By writing the binary value 0000 0001 to address H37A, pin1 on the D-shell connector can be loaded with 5V, (see Table 5.5), thereby supplying a separate signal. In order to terminate the sprocket signal, 0000 0000 is loaded into address H37A, placing 0V on pin1.

To identify the status of the busy signal, pin11 is connected to the busy signal supplied by the NC controller. By reading the value of address H379 and performing the logical "AND" operation, (as explained in section 5.3), the status of the busy signal can be ascertained. Table 5.6 lists the pin numbers of the D-shell connector and their corresponding signals on the paper tape reader interface.

TABLE 5.6 Cabling Connections

D-Shell Connector	Corresponding Reader Signals
Pin1	Sprocket
Pin2	Data track 1
Pin3	Data track 2
Pin4	Data track 3
Pin5	Data track 4
Pin6	Data track 5
Pin7	Data track 6
Pin8	Data track 7
Pin9	Data track 8
Pin11	Busy

Table 5.7 lists the hexadecimal addresses and the functions they perform.

TABLE 5.7 The Hex Addresses and their Functions

Address	Function
H378	Supply Sprocket Signal
H37A	Supply 8 Parallel Data
H379	Monitor Busy Signal

5.5 The BTR Interface Algorithm for NC Machines

In section 5.2 the interaction of the paper tape reader interface control signals was explained. The computer algorithm that simulates this interaction is as follows :

[1] READ ADDRESS H379
    If (value of H379) AND (1000 0000) = 1000 0000
        Then goto [1]   {Busy signal still high 5V}
        Else goto [2]   {Busy signal low 0V}

[2] WRITE CHARACTER TO H378
    {Supplies 8 parallel data signals}

[3] DELAY

[4] WRITE 0000 0001 TO H37A
    {Supplies sprocket signal}

[5] DELAY

[6] WRITE 0000 0000 TO H37A
{Terminates sprocket signal}

[7] DELAY

[8] WRITE 0000 0000 TO H378
{Terminates data signals}

[9] GOTO [1]

The above algorithm will produce the data transfer sequence illustrated in figure 5.3 in section 5.2. Appendix B explains the structure of the BTR Interface program code for NC machines. The code itself appears in Appendix C. For an interactive tutorial on the use of the BTR Interface for NC machines, refer to Appendix D.

6. BTR INTERFACE : CNC MACHINES

Section 5 explained the parallel method of data transfer used when interfacing a PC to an NC machine. This section will explain the method of data transfer between a PC and the more modern CNC machines involved in this project. The main difference between an NC machine and a CNC machine lies in the level of sophistication of the machine controller. Unlike the NC machines CNC machines have controllers that support fairly sophisticated electronic file management functions such as :

- "On-board" memory [illegible] : storage of entire part programs.
- Editing functions that facilitate alterations to part programs once they have been stored in memory.
- Facilities to copy part programs from memory to cartridge tape or floppy disks, (thereby releasing the memory).
- Facilities to punch/print part programs for hardcopy storage.
- Facilities to read in part programs from tape readers/host computers.

From an interfacing point of view, the main difference is the fact that the entire part program is transferred to the CNC controller before any machining occurs. Although parallel interfaces are available for the printing and punching of part programs, the interfaces used for the transfer of part programs to CNC controllers were serial interfaces.

6.1 Serial Data Transfer

In 1969 the EIA (Electronic Industries Association), Bell LAboratories and manufacturers of communications equipment cooperatively formulated and issued the EIA RS-232 standard. This almost immediately underwent minor revisions to become RS-232C. This standard has been widely accepted by industry and is the standard used by CNC machine controller manufacturers and therefore is also the standard used for serial data transfer in terms of this project.

Section 5 explained how the simultaneous transmission of 8 bit-voltages that constituted a byte form parallel data transfer. This form of data transfer is carried out byte by byte along 8 separate, discrete wires. Serial data transfer on the other hand, is carried out bit by bit. Here the bits constituting a byte (character) are transmitted along a single wire one after the other according to a predetermined rate of transfer (baud rate). The bits are represented by the following voltages :

Logical 1 : -15V

Logical 0 : +15V

For example, consider the character "N" being transmitted serially. It's 8-bit binary value of 0100 1110 is transmitted along a single wire with voltages of +15V and-15V placed on it at regular intervals. Figure 6.1 illustrates this process.

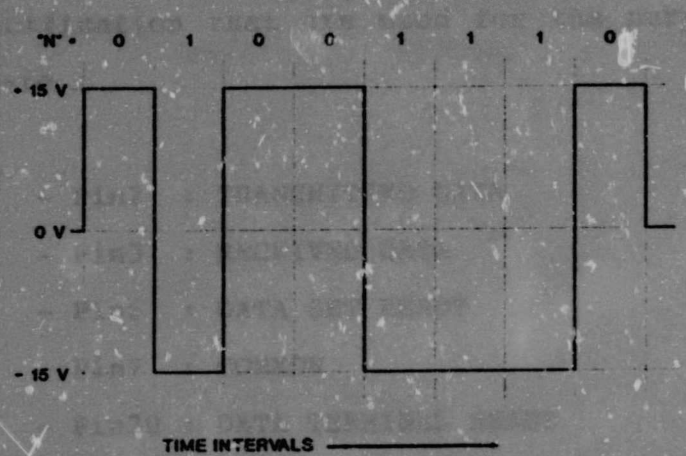


Fig 6.1 The Character "N" Tramsmitted Serially

6.2 EIA RS-232C Interface Control Signals and Pin Connections

The complete EIA RS-232C Interface specification incorporates the use of 21 wires and pin connections. However, for the purposes of this project, only 5 pin connections are used. The reason for this is that the complete specification provides for sophisticated control measures for data transfer over long distances at high speed. Many of these data transfer control measures are not required for simple, short distance data transfer as used between the PC's and the CNC controllers, and are therefore not in the scope of this project.

The serial interfaces on the PC's and the CNC controllers use the 25 pin D-shell connectors displayed in figure 5.4 in section 5. The pins specified in the complete EIA RS-232C specification that are used for the purposes of this project are :

- Pin2 : TRANSMITTED DATA
- Pin3 : RECEIVED DATA
- Pin6 : DATA SET READY
- Pin7 : COMMON
- Pin20 : DATA TERMINAL READY

Pin2 : TRANSMITTED DATA : This pin transmits data (bits as +15V or -15V) according to the specified baud rate.

Pin3 : RECEIVED DATA : This pin receives data. Pin2 on the PC is therefore connected to pin3 on the CNC controller. For data to be transferred from the CNC controller to the PC, pin2 on the CNC controller is connected to pin3 on the PC.

Pin6 : DATA SET READY : This pin signals the transmitting device that the receiving device is powered up and is ready to operate.

Pin7 : COMMON : This pin is mandatory in all RS-232C cabling connections. Pin7 on both devices are connected to each other. It provides the reference point for all interface voltages.

Pin20 : DATA TERMINAL READY : This pin signals the receiving device that the transmitting device is powered up and is ready to operate. Pin20 on the PC is therefore connected to pin6 on the CNC controller. Conversely, pin6 on the PC is connected to pin20 on the CNC controller. This allows the PC and CNC controller that they are powered up and ready to operate.

The pin connections between the PC's and the CNC controllers are graphically illustrated in figure 6.2.

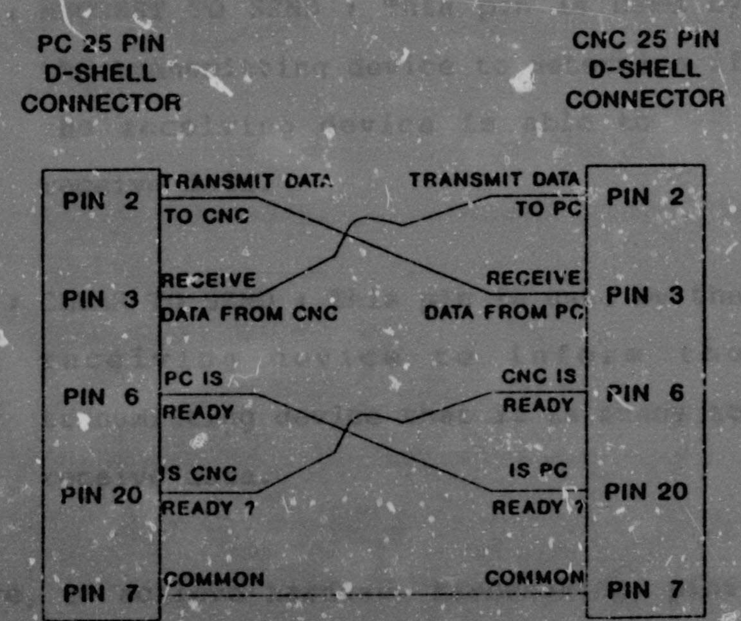


Fig 6.2 Pin Connections Between PC and CNC Controller