

Chapter 4. Programming

4.1. Command Attributes

All IFX commands can be categorized into the following groups: they are either immediate (I) or buffered (B); they are also either universal (U) or device specific (D). Commands are also characterized by being either sequence, parameter, execute or status commands.

Immediate commands are processed when they are received. Buffered commands are stored in a command buffer and processed in the order they were received, when the IFX is not otherwise busy.

All commands are either universal (U) or device specific (D). Universal commands do not need to be prefaced by a device identifier; device specific commands must be prefaced by a device identifier. All reportback commands are device specific. The reason for this is to avoid having all of the IFX indexer/drives in a daisy chain configuration from issuing a response and cluttering the serial communications line. Universal commands may also be prefaced by a device identifier, making all IFX units having the same address perform the command.

4.2. Three wire interface

The IFX uses a three wire implementation of RS-232C. It does not support handshaking of any form. The computer or terminal sending characters to the IFX should have its handshaking disabled through either hardware or software.

Serial communication transmits data one bit at a time. By transmitting bits at a predefined interval, the data can then be grouped into a byte. Serial communication utilizes a single line to transmit data and a second line to receive it.

4.2.1. Troubleshooting RS232 Communications

Procedures for troubleshooting 3 wire RS232 communications are as follows:

1. Make certain the transmit of the host is wired to the receive of the peripheral, and receive of the host is wired to the transmit of the peripheral.

NOTE: Try switching the receive and transmit wires on either the host or peripheral if you fail to get any communication.

2. Some serial ports require handshaking. You can then establish 3 wire communication by jumpering RTS to CTS (usually pins 4 and 5) and DSR to DTR (usually pins 5 and 25).
3. Configure the host and peripheral to the same baud rate, number of data bits, number of stop bits, and parity.

NOTE: The IFX is configurable in Baud Rate only.

4. If you receive double characters, for instance typing "A" and receiving "AA", your computer is set up for half duplex. Change the setup to full duplex.
5. Use DC common or signal ground as your reference, NOT earth ground.
6. Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to shield the cable to earth ground at one end only.
7. To test your terminal or terminal emulation software for proper 3 wire communication, unhook your peripheral device and transmit a character. You should not receive an echoed character. If so, you are in half duplex mode. Jumper the host's transmit and receive lines and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin-outs.

4.3. RS232 Daisy Chaining

Up to 8 IFX's may be daisy chained to a single RS232 port. It is necessary for each IFX to have an exclusive address setting so that each indexer may be accessed individually. Device address settings can be found in Appendix A. It is possible to chain more than 8 IFX's together, however, this will result in multiple indexers sharing the same device address and, consequentially, performing identical functions. Daisy Chaining Multiple IFX's require that the Receive line of each IFX be connected to the Transmit line of the IFX preceding it in the daisy chain, with the exception of the first IFX in the chain, which has it's Receive line connected to the Transmit line of the RS232 communications device (terminal). The Transmit line of the last IFX in the daisy chain is connected to the Receive line of the terminal. All IFX grounds for RS232 should be run individually back to the host terminal.

For Example,

A user has 3 IFX's that must be daisy chained to a single RS232 port. To accomplish this, the following steps are performed.

- 1) Each IFX is given a different device address. In this example, as in most cases, the address settings will be 1, 2, and 3. (See Appendix A for address settings)
- 2) The Transmit line of the terminal is connected to the Receive line of the first IFX in the daisy chain. It is not required that the first drive in the daisy chain be set at the lowest address setting. For reasons of clarity in this example, however, this will be the case.
- 3) The Transmit line of the first IFX is connected to the Receive line of the second IFX in the daisy chain.
- 4) The Transmit line of the second IFX is connected to the Receive line of the the third IFX in the daisy chain.
- 5) The Transmit of the third (and last) IFX is connected to the Receive line fo the terminal.

NOTE

All IFX grounds for RS232 should be run individually back to the host terminal.

Refer to Figure 4-1 for the above example.

4.3.1. Tips and Hints

Device specific commands require that a device address precede them. The IFX will not execute a device specific command if: there is no device address preceding the command, or, the device address setting in the IFX does not match the address preceding the command. Universal commands do not require device identifiers preceding them. A Universal command with no device address will be executed regardless of the address setting of the IFX. If a device address does precede a Universal command, it will only be executed by an IFX set to that particular address. (refer to Chapter 8 for definitions of Device specific and Universal commands).

The "E" (Enable RS232 communication) and "F" (Disable RS232 communication) commands are useful in a daisy chain for locking out particular drives from responding to Universal commands with no preceding device address.

NOTE

The "F" command will keep the IFX from executing any commands (except the "E" command) sent to it over the RS232 interface, but will not prevent the command from being echoed.

For Example,

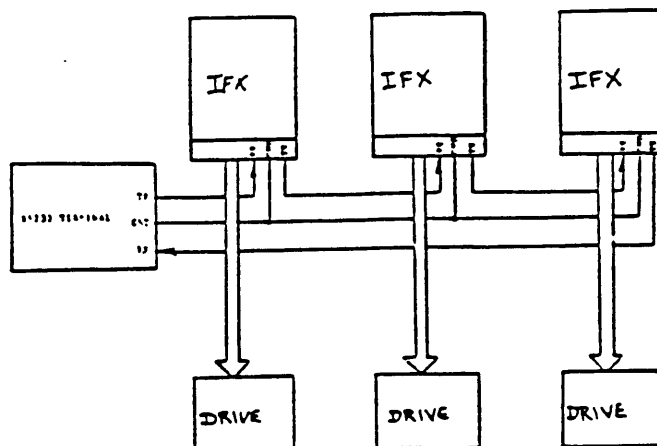
To Lock out the IFX set to device address 1, so that Universal commands are only executed by the IFX's set to address 2 and 3, the following step is performed.

- 1) 1F_ is sent over the RS232 line locking out the IFX at device address 1.

All Universal commands (with no preceding device address) will now be executed only by the IFX's at Device addresses 2 and 3. Entering a 2F_ in addition to 1F_ would allow only the IFX at device address 3 to execute Universal commands with no preceding device address. This eliminates the need to precede every command intended for a specific IFX (in this case, the IFX at device address 3) with a device address. Sending E_ over the RS232 line will re-enable all drives previously disabled with the "F" command. Preceding the "E" command with a device address will only re-enable the IFX set to that particular device address.

When using the "XU" command to upload the contents of a specific sequence from an IFX in a daisy chain, it is necessary to disable all the drives in the chain that come after the drive being queried with the "F" command as described above. This will prevent subsequent drives from executing the commands being sent back to the terminal. See description of "XU" command.

Figure 4-1: Multi-Axis Wiring



4.4. Sequence Programming

A sequence is a series of commands that is executed in order whenever the sequence is run. Immediate commands cannot be stored in a sequence just as they cannot be stored in the command buffer. The only "sequence legal" commands are buffered commands.

To begin the definition of a sequence the XD command immediately followed by sequence identifier number (1 to 7) and a delimiter must be entered. The XT command ends the sequence definition and automatically loads the sequence into the IFX's non-volatile memory. All commands entered after the XD command and before the XT command will be the commands executed (in the order entered when the sequence was defined) when the sequence is run.

NOTE

It may be necessary to erase a sequence (XEn) prior to defining it.

Example:

Keyboard input

XE1	(Erase Sequence #1)
XD1	(Definition of sequence #1 is begun)
A10	(A preset move with a distance
V10	(of one rev in the CW direction and
D25000	(an accel of 10 rev/sec/sec and a vel of
G	(10 revs/sec is defined and is told to Go)
H	(Direction of the next move is set to CCW)
G	(Previously defined move with direction
	now inverted is told to Go)
XT	(Definition of sequence #1 is terminated
	the sequence is stored in non-volatile
	memory)

Any time sequence number one is executed, the motor will turn 1 revolution clockwise and then 1 revolution counterclockwise. The methods for executing a sequence are described below.

A sequence can be run by entering the XR command immediately followed by a sequence identifier number (1 to 7) and a delimiter. It is also possible to have a sequence executed on power-up.

Once a sequence is defined it cannot be redefined until it has been deleted. A sequence may be deleted by entering the XE command immediately followed by a sequence identifier (1 to 7) and a delimiter.

The IFX has non-volatile memory allocated for storing 7 sequences of up to 256 characters each (including delimiters). However, one sequence cannot "borrow" any unused portion of another sequences allocated memory, 256 characters is the maximum for any sequence.

For more information on the implementation of the commands discussed in this section, refer to chapter 5.

During normal operation a command can specify the execution of any sequence stored in memory.

- a) Sequence identifiers: number from 1 - 7.
- b) When a sequence is downloaded the IFX automatically calculates a checksum for the sequence and stores it. A command can be used to verify the existence of a sequence (not erased) and the validity of its checksum. A sequence with a bad checksum will not be executed.
- c) An internal directory stores the length of each sequence and its checksum. The power-on execution sequence method and sequence number, if appropriate, are also stored in the non-volatile memory (EEPROM).
- d) Sequences must be explicitly deleted. A new sequence cannot be downloaded over an existing one.
- e) Sequences are downloaded by first sending a special start download command, which clears the command buffer. All subsequent commands are stored in the command buffer (but not executed) until an end sequence command is received. At this point the sequence is stored in the EEPROM. Any condition which prevents proper recording of the sequence is saved and can be accessed with a status command. The downloaded sequence can be tested with the run sequence command which executes any specified sequence.
- f) A query command sends the entire sequence back to the host (see XUn commands).
- g) EEPROM errors are reported only as a download failure mode. Other errors will have to be detected by the host, using various status commands.

4.5. Power on sequence execution

The IFX-Indexer/Drive may be programmed to execute a sequence of commands on power up, once, or continuously.

4.5.1. Normal mode: XPO

No sequence is executed upon power up. This is the default mode.

4.5.2. Single run: XP1-7

One of the sequences can be designated to be executed when the device is powered up. The sequence to be executed can be selected by prior RS-232C command (XP1-7), or with the Sequence Select inputs (XP8) (see definition of XP9 mode for Sequence Select input definition). The single sequence specified is executed once; control then passes to the normal command processor loop. Alternatively, power-on execution can be disabled with either the XP0 or XZ commands.

4.5.3. Continuous Run: XP9

The Sequence Select inputs are read: if the Sequence input is 1-7, the corresponding sequence is executed; if the Sequence input is 8, the IFX waits for it to change. Whenever a sequence is finished, the Sequence Select inputs are read for the next sequence to execute. If the previous sequence contains an XQ1 (sequence hold) command, a new sequence will not be run after the previous sequence until the Sequence Select inputs first read a value of 8 (all Sequence Select bits high). Need to reset if RS232 is sent.

Note: See Section 2.4 for more on Using Sequences

4.5.4. Miscellaneous

- a) Various EEPROM faults produce an error state. At this point only immediate commands are functional. The R and RC commands can be used to determine remotely the existence of this error state.
- b) A hardware fault could cause the microprocessor to be held in reset and the IFX will stop communicating.

4.6. Command Groups

In this section commands are broken into 4 groups. These groups are,

- Parameter - Commands that set parameters.
- Execute - Commands that cause operations to be performed.
- Sequence - Commands pertaining to sequences.
- Status - Commands that cause the IFX to send information to the terminal.

Application examples are given to help illustrate the usefulness of most or all of the commands within a given group.

4.6.1. Parameter Commands

A - Set acceleration.
D - Set distance for preset moves.
H - Set direction/Change direction of next move.
L - Loop.
MC - Set mode to continuous.
MN - Set mode to preset (normal).
MPA - Set position mode to absolute.
MPI - Set position mode to incremental.
N - End of loop.
V - Set velocity.

Example #1

A scientist wants to scan a molecular sample at a rate of 1 revolution per second. He finds that from time to time he wants to scan more quickly move past uninteresting portions of the sample. He can do this by making use of the A, MC, and V commands. The MC commands sets the move mode to continuous, (moves without a specified end point the motor will move until given a command to stop or it hits a limit switch).

To begin a slow scan the scientist would enter:

MC A10 V1 G

This will accelerate the motor from zero velocity to one rev per second in 0.1 seconds. To change the velocity to a faster one for passing uninteresting portions of the sample he would enter:

A100 V10 G

This will cause the motor to accelerate from 1 rev per second to 10 revs per second at an acceleration rate of 100 revs per second per second. Once he is past the uninteresting portions of the sample he can return to his original velocity by entering:

V1 G

Example #2

The same scientist has a second application requiring him to position a rotary table. The rotary table holds 8 test tubes, each test tube must be positioned beneath a valve that releases a measured amount of a chemical into each tube to react with a specimen already in the tube. The table is geared so that one revolution of the motor corresponds to the distance between each tube in the table. The scientist wants to,

- a. position a tube beneath the valve,
- b. release the chemical,
- c. repeat 8 times,

To do this he could use the D, L, N, MN, and T commands. The T command is an Execute command, its function is to introduce a time delay into the execution of the command string. The time delay will give the valve time to dispense the chemical into the tube. It has been determined that 2 seconds are required for this to take place. The operation can be initiated by entering:

MN A5 V10 D25000 L8 T2 G N

This will cause the table to position each tube beneath the valve and stop for 2 seconds while it receives the chemical from the tube and then index to the next tube and repeat the process until all 8 tubes have received the chemical, then the operation will stop (assume the use of a 25000 step/rev motor).

Example #3: Using the Y and H commands.

After the tubes have all received the chemical. Corks will be placed on them, and the table will be moved in an alternating motion at a high velocity and acceleration to agitate both the chemical and the specimen inside the tube. To do this the H command is used to reverse the direction after every move. This operation is achieved by entering:

A100 V15 D25000 L H G N

The absence of a number immediately following the L command will cause the loop to be executed indefinitely until the Y command (stop execution of the loop after the current pass) is entered.

4.6.2. Execute Commands

- C - Continue (end pause state).
- E - Unlock device.
- F - Lock device.
- G - Go (make a move).
- GH - Go Home.
- K - Kill (Stops motion immediately under any condition or mode).
- LD - Enable/disable limit inputs.
- PS - Pause execution of current command string.
- PZ - Set current position as absolute zero.
- Q1 - Sets velocity profiling mode
- Q0 - Clears velocity profiling mode
- RM - Set immediate velocity.
- S - Stop.
- ST - Disable/enable shutdown.
- T - Time delay.
- TR - Wait for trigger.
- U - Pause and wait for continue.
- Y - Terminate loop at end of current pass.
- Z - Clear command buffer.

4.6.2.1. Notes on velocity profiling mode:

Situations requiring non-linear accelerations may use the Q0, Q1 and RM commands. Q1 is used to enter the velocity profiling mode, and Q0 is used to exit. While in this mode the RM command is used to generate velocity values that are immediately implemented while the motor is moving. This means that the RM command must be sent to the IFX at the time the change in velocity is required. This creates a stairstep effect in velocity change. By implementing a large number of very small instantaneous velocity changes, a smooth, non-linear acceleration ramp can be achieved.

Assuming the IFX's communication protocol is set at 9600 baud, 1 ms is required to transmit a single character from the terminal to the IFX. The RM command is seven characters long (delimiter included) and therefore takes 7 msec to be transmitted from the terminal to the IFX. This limits the update rate of the velocity to once every 7 msec.

4.6.2.2. Using The Execute Commands

Example #1

An operator has an application which requires him to initialize the home position before beginning operation, but first, all mechanical hazards must be cleared in the system to avoid injury or damage to the equipment. To do this the PS, C, TR, LD, GH commands may be used. A sensor is connected to each of the trigger inputs indicating the presence or absence of mechanical hazards. A simple program is written that sends the following command string to the IFX.

PS LD3 TRX11 A10 GH-5 MPA

This will first pause execution of the command string with the PS command, so that the command string is loaded into the IFX but not executed. The user's computer controlling the IFX is programmed to then prompt the operator if he is ready to begin initialization. If so, a C is sent to the IFX and execution of the command string begins. The first command encountered by the IFX is the LD command. The LD3 command disables the limit inputs. The second command encountered is the TR command. In this example the TR command causes the IFX to wait until trigger inputs 2 and 3 are in a "high" state before executing the command string (trigger 1 is ignored). The trigger inputs will only be in the "high" (inputs off) state when the sensors indicate that all mechanical hazards are cleared. Once all hazards are clear, the motor will search for home in the CCW direction at 5 revolutions per second.

Once home is found, the IFX will set the position mode to absolute, and define the home position as absolute zero.

Example #2:

The operator then wants the motor to turn 2 revolutions CW and 1 revolution CCW and return to the Home position. To do this he enters:

MN A3 V5 D50000 G D25000 G GH

Example #3:

He then wants to send the motor in a continuous move in the CW direction and visually determine when he wants the motor to begin decelerating to a stop. To do this he enters:

MC G

The motor will now move in the CW direction at 5 revolutions per second. To cause the motor to decelerate to a stop at the currently defined accel/decel rate he enters:

S

If for any reason it becomes necessary for him to bring the motor to an immediate stop with infinite deceleration, he enters:

K

Execution of the K command will cause all move parameters to be lost.

4.6.3. Sequence Commands

XD - Start sequence definition.
XE - Delete sequence.
XP - Set power-on sequence mode.
XQ - Set/reset interrupted run mode.
XR - Run a sequence.
XRP - Run a sequence with a pause.
XT - End sequence definition.
XU - Upload sequence.
XZ - Set power on sequence to zero.

It is possible to have the IFX immediately begin executing the operations in the 3 execution examples above on power up by defining the commands within a sequence using the XD, XT, and XP commands. To define sequence one the operator enters:

```
XE1 XP1 XD1 LD1 TRX11 A10 GH-5 MPA PZ MN A3 V5 D25600 G D12800 G MC G  
XT
```

As soon as the XT command is entered, the entire sequence is stored in non-volatile memory.

In this example the operator would not power the unit on until he was ready for operation to begin. The XP command has enabled the single run power on execution mode. This means that when the unit is powered on the sequence whose assigned number immediately follows the XP command will be executed. The XD command signifies the start of the definition of a sequence. The number immediately following the XD command is the number assigned to the sequence. The XT command signifies the end of the sequence definition. All commands between the XD and XT commands will be performed when the sequence is executed. When the operator applies power to the IFX sequence number 1 will be executed. It is not necessary after power is applied for the operator to supply any further input to the IFX until he is ready to stop the sequence (S command) or unless it becomes

necessary to implement an emergency stop (K command).

If for some reason the operator wishes to check the contents of the sequence, he would enter:

LXU1

This would cause the IFX to send the contents of sequence number 1 to the terminal's screen. The 1 preceding the XU command is the device address which must be present since the XU command is a "device specific" command. We are assuming in this example that the IFX is set up at device address 1. After issuing this command the operator will see the following on his terminal's screen,

```
D1 TRX11 A10 GH-5 MPA PZ MN A3 V5 D50000 G D25000 G MC G
```

This is the command string that is performed when sequence number 1 is executed. If there is an error in the sequence, or if for any reason the operator wishes to change the sequence in some way, the sequence must first be deleted before it can be redefined. To delete sequence number 1 the operator enters:

XE1

Sequence number 1 is now deleted and may be re-entered.

If the operator does not wish to have the sequence execute on power up, there are two ways to cancel the execute mode set by the XP command. By issuing the XPO command, or it can be canceled by issuing the XZ command.

The XR command is used to execute a sequence manually. To execute sequence number 1 the operator enters:

XR1

Sequence number 1 would now execute.

4.6.4. Status Commands

B - Buffer status request.
BS - Buffer size request.
IS - Input Status request.
PR - Absolute position request.
R - Indexer status request.
RA - Limit status request.
RB - Loop, pause, shutdown, trigger status request.
RC - Stall detect, Go Home status request.
RS - Sequence move status.
RV - Software revision request.
TS - Trigger status request.
W - Request position on the fly.
XC - Request checksum for EEPROM.
XSD - Download status request.
XSP - Power on sequence mode request.
XSR - Run sequence status request.
XSS - Sequence status request.

All status commands are device specific and must be preceded by a device address.

The use of status request commands must be conducted in an orderly fashion. Commands should only be issued when the host is ready to read the response. New commands should not be sent until the response is received. In particular, after a buffered status command, an immediate status command should not be sent until the buffered command status response has been received by the host. If this injunction is not followed, the command responses will be intertwined, rendering the information useless.

There are two status commands that can be used to request position.

The PR command will report how many steps the motor has moved relative to the absolute zero position. This position is calculated by summing the total number of moves commanded since the IFX was at the absolute zero position, and can be erroneous if the motor has stalled. This command may only be used when the motor is not moving. If position reports are needed that are relative to the beginning of each individual move, the PZ command should be issued at the start of each move to zero the absolute position counter.

The W command is used to get position reports while the motor is moving. This is also a relative position report. The W command will report position back in two formats, binary (not in ASCII) or hexadecimal (in ASCII). If the W command is followed by a 1, the report will be in binary. If the W command is followed by a 3, the report will be in ASCII. This report will always be in two's complement notation.

4.6.4.1. I/O Status Requests

The RA command is useful when the motor will not move in either or both directions. The report back will indicate whether or not the last move was terminated by a limit switch activation, and if the current move is disabled by an active limit.

The RC command will indicate whether or not the last Go Home attempt was successful.

The TS command is useful for checking the status of the trigger inputs when it appears as though execution is being halted by the TR command and all conditions for matching the trigger input configuration defined by the TR commands appear to be met. It may also be used to initiate external actions by monitoring the trigger inputs manually with a computer controlling the IFX.

4.6.4.2. Sequence Status Requests

Sequence status commands include XC, XSD, XSP, XSR, XSS. The XC command can be used to verify that the memory in the EEPROM has not been corrupted in any way since last power down. The XC command will cause the IFX to send a checksum to the terminal that will be in the form of 3 decimal digits (000 - 255). The check sum may be recorded when sequences have been defined, so that it may be used for comparison later, such as each time the IFX is powered up.

The XSD command is useful for verifying that the last sequence download attempt was successful. If the attempt was not successful the IFX will indicate one of three possible reasons why

The XSP command is useful for obtaining the sequence number that will be executed on power up in the execute sequence on power up mode.

The XSR command will inform the user whether or not a sequence is in progress. If a sequence should be in progress and is not, this command will report back the reason why.

The XSS command will inform the user if the sequence indicated by the number immediately following (no space) the XSS command does in fact exist in memory, and if so, if that portion of memory has been corrupted or not.