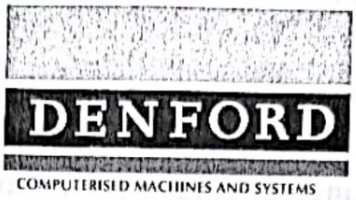
DENFORD LTD



Denford Limited
Birds Royd, Brighouse, West Yorkshire
HD6 1NB, England.
Telephone: 01484 712264
Fax: 01484 722160
Email:info@denford.co.uk

Computer Integrated Manufacturing Systems

CIM's

CIM EQUIPMENT INSTALLATION SUPPLEMENT

INCLUDING

Denford Machine Tool Controller
Fanuc 0 Machine Tool Controller
Fanuc 21i Machine Tool Controller
Denford Programmable Linear Slide
Mitsubishi RV-M1 Robot
Denford New Automatic Storage & Retrieval System (ASRS)

11.9.2002

Powered by CamScanner

Machine Tool Installation

The device drivers for both Denford lathes and millers communicate to the machine tools in two different ways: RS232 serial links and digital I/O signals.

RS232 is used for program download; Direct Numerical Control (DNC) and detailed state checking. An RS232 cable should connect from a serial port on the supervising Cell PC to the external serial port on the machine tool.

I/O signals are connected directly from the Interface Module to the machine's auxiliary input and output port. There are two input lines and two output lines which can be used to transmit low-level state information to the device driver.

RS232 serial communication

To communicate via the serial link, both machine tool and device driver must use the same settings. For the machine tool, the settings are stored in the 'options file' (.opt extension), the serial communications parameters have the prefix EXTCOMM.

EXTCOMM_ENABLE 1	enables serial communications
EXTCOMM_ROUTE 1	specifies the use of the com port
EXTCOMM_SDEVICE COM1	specifies the name of the com port
EXTCOMM_BAUD 4800	the baud rate (300, 1200, 2400, 4800, 9600)
EXTCOMM_PARITY 0	parity settings (0=none, 1=odd, 2=even)
EXTCOMM_DATABITS 8	specifies the number of data bits (7, 8)
EXTCOMM_STOPBITS 1	specifies the number of stop bits (1, 2)
EXTCOMM_LF 1	specifies whether Line Feed is used (0, 1)
EXTCOMM_ES 1	1st escape character
EXTCOMM_EE 2	2nd escape character

The address and interrupt request number for the machine's communications port is stored in the 'Go' file (.go extension).

Address, interrupt, name
$3E8 5 COM1

The name in the Go file must correspond to the name in the Options file.

Communications must use Hardware flow control (CTS/RTS).

For the device driver, the communications settings are stored in the Initialisation file (.ini extension).

Powered by CamScanner

.ini file config.

```
[COMMS]
Port          =    COM1          specifies which PC port to use
Settings      =    4800,n,8,1    the communication settings
Flags         =    H1            flow control flags
OutBufferSize =    1024          buffer sizes (bytes)
InBufferSize  =    1024
TimeOut       =    15000         RS232 timeout (ms)
```

The 'Port' name is Window's reference to the serial port to be used (see Window's Control Panel's Ports section). The 'Settings' specify the baud rate, parity (n=none, o=odd, e=even), data bits and stop bits. 'Flags' specify the flow control to be used, (H0=no handshaking, H1=hardware handshaking, H2=XON/XOFF flow control).

The settings for the machine and the device driver above do correspond so serial communications can take place.

There are two methods for transferring data to a Denford Machine controller, file transfer and Direct Numerical Control (DNC or drip-feeding).

File Transfer

The Denford controller recognises the command "$$MACHINE FROM <filename>", where 'filename' is a valid NC program. If this command is sent over the serial link the controller will load the specified program and immediately start execution. The file name holds a path to the program so, if the machine tool is part of a PC network, the path could specify a Network File-server.

DNC

The Denford controller also recognises the commands "$$MACHINE", "$$SCREEN" and "$$EXECUTE". These commands initiate drip-feeding of a NC program.

$$MACHINE executes each block as it is received.

$$SCREEN simulates each block as it is received.

$$EXECUTE executes each block in the current machine mode (AUTO or SIMULATE).

To drip feed a program, put the line '$$MACHINE' at the beginning of the file and the line '$$RESET' at the end. The '$$RESET' command flushes the RS232 buffer and resets the controller to its initial state.

Powered by CamScanner

I/O Signal Communication

In a Cell PC there is an Industrial I/O card. The I/O card provides 16 relay outputs and 16 photo-couple inputs. This allows the Cell PC to send signals directly to a machine tool via its auxiliary I/O port.

All Denford Machine Tools are connected to the I/O interface module in ports 2A or 2B; the lead has a QM style plug which fits into the auxiliary port. Some machines may already have an emergency stop terminator plug in the auxiliary port which allows the machine to operate without a remote emergency stop. When the machine is to be part of a CIM, the terminator plug should be replaced with the lead to the I/O interface module.

The auxiliary port provides two inputs to the machine tool, two outputs from the machine and an emergency stop circuit. The CIM software automatically energises the emergency stop circuit unless the remote emergency stop button is depressed. The inputs and outputs to and from the machine tool correspond to four 'M' codes:

M62 output 1 on
M63 output 2 on
M64 output 1 off
M65 output 2 off

There are four other 'M' codes which can be used to delay execution of a NC program until their condition is met:

M66 wait for input 1 to turn on
M67 wait for input 2 to turn on
M76 wait for input 1 to turn off
M77 wait for input 2 to turn off

Input/output 1 corresponds to bit 0 on the I/O interface module, input/output 2 corresponds to bit 1.

The initialisation file for a device driver contains a section headed '[IO PORT DATA]' which holds an interpretation of the IO signals. See the device driver manual for a detailed explanation.

Input 2 on a Denford Machine Tool can be used to remotely control the clamp. The clamp opens when a signal (+24V DC) is sent and closes

when it is removed. To allow the remote clamp to operate, the machine tool's options file (.opt extension) must have the line:

Powered by CamScanner

REMOTECLAMP 1 (MILLER)

REMOTECHUCK 1 (LATHE)

Structure of a CIM program for machine tools

To safely use a Denford Machine Tool in a CIM environment it is recommended that the programming technique explained below is used.

The NC program should be prefixed with the lines:

1. **M66 / M76** *(wait for input 1 to turn on / off)*
2. **M62** *(set output 1 to on)*
3. **M39** *(close the door)*

When this program is executed, it immediately halts at line 1 and waits for confirmation from an external signal (usually provided by the Denford device driver software). Line 2 sends a signal to the device driver to tell it that the machine is busy and line 3 closes the door ready for machining.

At the end of the program (line n) the 'busy signal' should be turned off with the lines:

n-2. **M38** *(open the door)*

n-1. **M64** *(set output 1 to off)*

n. **M30** *(end of program)*

The main program can either be directly inserted between the above two sections or it can be stored as a separately and called as a sub-program.

'M' code number 98 calls a subprogram, the parameter P holds the name of the program to be called. So in the example above, line 4 could be:

4. M98 P0003; *(call sub-program 0003)*

Program 0003 will perform the required actions and then execute the M

code 'M99' to return to the original calling program. Refer to the machine-tool programming manual for further help.

Powered by CamScanner

RS232 Functions of FANUC 0 Controllers

As well as the standard RS232 link capabilities of the FANUC 0 controls there are also a further two functions available for mass data transfer and specialist computer links.

DNC

By programming auxiliary code "M29" in "MDI" mode, The control will place itself in the "DNC" mode. Any valid NC blocks sent to the control will be executed but not stored in the controls memory and so a program of infinite length can be executed by the CNC machine.

The control will exit DNC mode if:

- An invalid block is transmitted.
- An "M02" or "M30" block is transmitted.
- Control/Reset is pressed.

During transmission the XON/XOFF protocol is used to control data flow.

"MINP" FUNCTION

With the "MINP" function enabled, the control will receive any valid NC block from a host computer and store these blocks in its internal memory. However, upon receipt of a program end code, it will automatically begin to execute the program just received until either:-

- "M02" or "M30" is executed, in which case the control will exit "MINP" mode.
- Auxiliary code "M15" is executed, in which case cycle execution will stop and RS232 input will resume.
- Control/Reset is pressed.

"MINP" mode us entered in one of two ways depending on the type of machine being used: note parameter PWE (Parameter Write Enable) should be set to 1 to alter the parameters.

1) **ON DENFORD FANUC 0 LATHES**

Set parameter 11, bit 7 to 1	use "MINP" input
Set diagnostic 452, bit 4 to 1	use "MINP" input
Set parameter 111 to 15	the M code M15 is not buffered

Powered by CamScanner

2) ON DENFORD FANUC 0 MILLERS

Set parameter 11, bit 7 to 1	use "MINP" input
Set diagnostic 451, bit 7 to 1	use "MINP" input
Set parameter 111 to 15	the M code M15 is not buffered

When parameter 11 is altered, PS alarm 00 occurs. This means that the controller should be switched off and on again with the new setting.

Please note that when the control starts to receive a NC program into its memory when in "MINP" mode, it will first erase any programs currently in its memory so the program protect key should be switched to OFF.

Baud rate is set with the BRATE1 parameter (552), note the I/O parameter must be set to 0 to use BRATE1.

BAUD : 4800
DATA : 7
STOP : 1
PARITY: EVEN

Again, data transmission is controlled using <XON> and <XOFF> protocol.

Start of program characters
"% ^M ^J" (CR/LF)

End of program characters
"M15 ^M ^J (CR/LF)
%"

All spaces are optional in a program, each line ends with a ^M and ^J (CR/LF)

Refer to the FANUC manual appendix for further details of parameters.

Powered by CamScanner

I/O Functions of FANUC 0 Controllers

The FANUC 0 controllers work in a similar way to the Denford controllers regarding auxiliary inputs and outputs, however various parameters must be set to enable them.

1) **ON DENFORD FANUC 0 LATHES**

Set parameter 452, bit 6 to 1	use auxiliary 2 for chuck
Set parameter 450, bit 7 to 0	enable M62/M64

2) **ON DENFORD FANUC 0 MILLERS**

Set parameter 450, bit 2 to 1	use auxiliary 2 for chuck
Set parameter 450, bit 7 to 0	enable M62/M64

RS232 Functions of FANUC 21i Controllers

As well as the standard RS232 link capabilities of the FANUC 21i controls there are also a further two functions available for mass data transfer and specialist computer links.

DNC

By programming auxiliary code "M29" in "MDI" mode, The control will place itself in the "DNC" mode. Any valid NC blocks sent to the control will be executed but not stored in the controls memory and so a program of infinite length can be executed by the CNC machine.

The control will exit DNC mode if:

- An invalid block is transmitted.
- An "M02" or "M30" block is transmitted.
- Control/Reset is pressed.

During transmission the XON/XOFF protocol is used to control data flow.

"MINP" FUNCTION

With the "MINP" function enabled, the control will receive any valid NC block from a host computer and store these blocks in its internal memory.

However, upon receipt of a program end code, it will automatically begin to execute the program just received until either:-

Powered by CamScanner

- "M02" or "M30" is executed, in which case the control will exit "MINP" mode.
- Auxiliary code "M15" is executed, in which case cycle execution will stop and RS232 input will resume.
- Control/Reset is pressed.

"MINP" mode us entered in one of two ways depending on the type of machine being used: note parameter PWE (Parameter Write Enable) offset setting should be set to 1 to alter the parameters.

1) **ON DENFORD FANUC 21i LATHES**

Set parameter 3201, bit 7 to 1	use "MINP" input
Set Keep Relay K1.4 to 1	use "MINP" input
Set parameter 3411 to 15	the M code M15 is not buffered

2) **ON DENFORD FANUC 21i MILLERS**

Set parameter 3201, bit 7 to 1	use "MINP" input
Set Keep Relay K1.4 to 1	use "MINP" input
Set parameter 3411 to 15	the M code M15 is not buffered

When parameter 3201 is altered, PS alarm 00 occurs. This means that the controller should be switched off and on again with the new setting.

Please note that when the control starts to receive a NC program into its memory when in "MINP" mode, it will first erase any programs currently in its memory so the program protect key should be switched to OFF.

To check the RS232 settings on the Fanuc 21i use the following procedure.

1. Press the **SYSTEM** Button.
2. Press the **SYSTEM SOFT** Button.
3. Press the Right hand extended menu Button three times so that the **ALL I/O** section is displayed.
4. Press the **SYSTEM SOFT** Button under the **ALL I/O** section and the RS232 setting will be displayed.

E.g.

Powered by CamScanner

I/O CHANNEL : 1
DEVICE NUM : 0
BAUD RATE : 9600
STOP BIT : 2

Parameter 0103 = 10 (Baud rate 4800)
Parameter 0103 = 11 (Baud rate 9600)
Parameter 0103 = 12 (Baud rate 19200)
Parameter 0101.0 = 0 (1 stop bit)
Parameter 0101.0 = 1 (2 stop bits)

Again, data transmission is controlled using <XON> and <XOFF> protocol.

Start of program characters
"% ^M ^J" (CR/LF)

End of program characters
"M15 ^M ^J (CR/LF)
%"

All spaces are optional in a program, each line ends with a ^M and ^J (CR/LF)

Refer to the FANUC manual appendix for further details of parameters.

I/O Functions of FANUC 21i Controllers

The FANUC 21i controllers work in a similar way to the Denford controllers regarding auxiliary inputs and outputs, however various parameters must be set to enable them.

1) **ON DENFORD FANUC 21i LATHES**

Set Keep Relay 1.1 to 1	use auxiliary 2 for chuck
Set Keep Relay 0.1 to 1	Programmable chuck

2) **ON DENFORD FANUC 21i MILLERS**

Set Keep Relay 1.1 to 1	use auxiliary 2 for vice
Set Keep Relay 0.0 to 1	Programmable vice

Powered by CamScanner

Structure of a CIM program for machine tools

To safely use a Denford FANUC Machine Tool in a CIM environment it is recommended that the programming technique explained below is used.

The NC program should be prefixed with the lines:

1.	**%**	*(every program should start with a '%')*
2.	**M66**	*(wait for Aux 1 to go High / On)*
3.	**M62**	*(set output 1 High / On)*
4.	**M39**	*(close the door)*

The door is closed in line 4 and line 3 sends a signal to the device driver to tell it that the machine is busy.

At the end of the program (line n-2) the 'busy signal' should be turned off with the lines:

n-3.	**M38**	*(open the door)*
n-2.	**M64**	*(set output 1 to off)*
n-1.	**M15**	*(resume "MINP" mode)*
n.	**%**	*(end of program)*

The main program can either be directly inserted between the above two sections or it can be stored as a separately and called as a sub-program.

'M' code number 98 calls a subprogram, the parameter P holds the name of the program to be called. So in the example above, line 4 could be:

4. M98 P0003; *(call sub-program 0003)*

Program 0003 will perform the required actions and then execute the M code 'M99' to return to the original calling program. Refer to the machine-tool programming manual for further help.

Powered by CamScanner

To down load a FNC file to the Fanuc 21i controller you need to get LSK flashing on the screen. On the control panel do the following steps

1. Press the EDIT (Hard) Button
2. Press the PROG (Hard) Button
3. Press the PRGRM (Soft) Button
4. Press the OPRT (Soft) Button
5. Press the ARROW RIGHT (Soft) Button
6. Press the READ (Soft) Button
7. Press the EXEC (Soft) Button

LSK should now be flashing on the screen

Alternatively

1. Press the MEMORY MODE (AUTO MODE) Button
2. Press the CYCLE START Button
   (PROGRAM PROTECT KEY IS SET TO "PROTECT MODE")

LSK should now be flashing on the screen

Note? The PROGRAM PROTECT KEY needs to be OFF

Robot Installation

Like machine tools, robots communicate with RS232 signals and digital I/O signals; however once a program has been downloaded to the robot, the RS232 cable can be removed because it is not used for sending or receiving status information while the robot is in cycle.

RS232 serial communication

The robot's control unit has an external RS232 port at the back; this should be connected to a computer which stores the robot program (either a cell computer or the host computer). The communications settings are altered with a set of switches in the side door on the right hand side. There are three sets of switches labelled SW1, SW2 and SW3, each is a bank of eight on/off switches where ON is signified by a switch in the upper position; here a 1 shall indicate ON (up) and a 0 shall indicate off (down).

Switch SW1 controls various operations but switch 1 controls the carriage return and line feed function for the RS232 operation. SW1

Powered by CamScanner

should be set to 01100001. Refer to the robot manual pages 2-15 and 2-16 for guidance.

Switch SW2 controls the asynchronous transmission including the number of stop bits, parity and character length. It should be set to: 01011110 which indicates 7 data bits, even parity and 1 stop bit. Refer to the robot manual Appendix 6 for guidance.

Switch SW3 controls the Baud rate and should be set to 00000010 which produces a baud rate of 4800.

To communicate to the robot, Windows Terminal should be used with the same settings, i.e. 4800 baud, 7 data bits, 1 stop bit, even parity and XON/XOFF flow control. The robot teach box should be switched OFF and the side setting switches (the two metallic toggle switches) should both be in the DOWN position to receive a program over the serial link.

The robot should now be able to respond to and receive commands. If 'WH' is typed into the Terminal window, the robot should reply with the positions of each of its axes (if it has been nested). To download a file,

choose **Transfers|Send Text File** from the menu of Terminal and select the robot file name to send. When the OK button is pressed the file will be sent to the robot. Either a program or a set of positions can be transferred using this procedure.

I/O Signal Communication

There are two commands in the robot language, which send and receive information to and from the robot's external I/O port. The commands are: OD (Output Direct) and ID (Input Direct).

Output Direct sends data unconditionally through the output port. It can send a number between 0 and 255 (decimal) and will maintain that output until it is changed by another OD instruction.

Input Data unconditionally reads from the input port and places the result into the robot's internal comparison register. The data can then be used by the robot to compare with constants and to make program flow decisions.

Structure of a CIM program for robots (Using a Robot RS232 Device Driver)

Powered by CamScanner

Utilising the RS232 Robot Device Driver means that Robot programs are down loaded from the Cell controller, to the Robot controller, and are immediately executed as and when they are required during the CIM System cycle.

When a program is downloaded to the robot controller using an RS232 Robot Device Driver, the first thing that must be done is to delete the old program from the robot's memory. This can be done by prefixing the program with the line

DL 1, 2048

The basic Robot program structure should be as follows :-

DL 1,2048	Deletes any program left in the Robot controller
10 OB +0	Turns Output bit 0 ON (Busy signal)
20	Main body of program
1000 OB +0	Turns Output bit 0 OFF (Idle signal)
RN	Run the program immediately

Structure of a CIM program for robots (Using a Robot I/O Device Driver)

When a program is downloaded to the robot, the first thing that must be done is to delete the old program from the robot's memory. This can be done by prefixing the program with the line

DL 1, 2048

This deletes program lines 1 to 2048, i.e. all possible line numbers.

The program should be written as a set of subroutines, each performing a simple sequence of actions. The subroutines can be executed in response to an external I/O signal.

The Denford robot programs all have three distinct sections:

1. An initialisation sub-routine

2. A main program loop

3. A set of movement sequences

The initialisation routine is called once at the beginning of the robot program. It may nest the robot and set various movement parameters

Powered by CamScanner

such as speed, acceleration and gripper strength.

The main loop continually reads the input port for data and continually sets the output port to 0 (idle). It appears in the program for a robot servicing a milling machine like this:

```
20 OD 0
30 ID
40 EQ 1, 600
50 EQ 3,300
60 GT 20
```

line 20 sets the output port to 0 (idle)

line 30 reads the input port and stores the result in the robot's internal comparison register

line 40 compares the register to '1', if the comparison is true, then the program jumps to line 600

line 50 compares the register to '3', if the comparison is true, then the program jumps to line 300

line 60 finishes the loop and jumps back to line 20 to start again.

The two subroutines called do two separate tasks; the first loads a component into the machine, the second takes a component from the machine.

Each subroutine executes a simple sequence of movements to manipulate a component and move it to the correct place. A subroutine may also execute 'interlocks', for instance: the robot may take a component and place it in the jaws of a vice on a milling machine; the subroutine is now finished so it returns an idle signal to allow the vice to close. The robot is in a potentially dangerous position because the only safe command to execute would be to leave the milling machine and park. If another subroutine were to run, the robot would collide with the machine tool. To avoid the collision the subroutine does not return to the main program loop where any subroutine could be triggered, but it starts another loop which scans for an input which triggers the one safe subroutine to execute. An interlock like this can be seen at line 244 in the miller-robot program

ASRS SETUP PROCEDURE

Powered by CamScanner