YUASA

SUDX
Programable
Indexer

Operations and Service Manual
Version 2.01

!!MANDATORY READ THIS!!

Introduction

Thank you for choosing the SUDX Programmable Table. These precision tables have been designed and manufactured for the highest possible durability, accuracy, and flexibility that is required for today diverse machining needs.

To make sure that you get the most out of your new unit, please completely read the aspects of programming and operating these units before actual setup and operation.

Special Cautions:

***1)** Power Source: 105 ~ 128 VAC*

2) Never apply voltages to lines 2 & 3 {Step Start} or 8 & 9 {Emergency Stop} on our remote Cable. **(We require an isolated relay to provide a contact closure only.)**

3) Place controller in a place free from oils, coolants, chips, and other foreign elements. Do not allow filters to become clogged or blocked. Air must be able to circulate freely else damage to the circuitry occur.

4) Always use a Filtered/Regulated/Lubricated air supply to these units. Failure will create rust and corrosion that will damage and foul these units air cylinders and braking systems.

5) Periodically check motor cabling for splits, breaks, and punctures that can allow coolants and other fluids to enter cabling or motor housing. If damage is found, please repair immediately to prevent possible motor, controller, &/or wiring damage. **Failure to do so can lead to very costly non-warranty repairs.**

6) Avoid tool and work pieces crashing into these units.

7) When mounting jigs, trunnions, and other fixtures, never use unnecessarily long set bolts. They will cause vibrations during rotation which can cause: oil leaks, tool breakage, etc.

8) When using trunnions and long work pieces, please use tailstocks with live centers or if you are using “A” frame supports, use bearings and not grease alone else undue stress and tension may cause damage to the motor, gears, or circuitry.

NEGLECTING THE ABOVE PRECAUTIONS WILL VOID OUR WARRANTY POLICY.

CREDIT POLICY

1) All returned goods must have prior authorization. The MRA (Merchandise Return Authorization) Number must appear prominently on the shipping label. **Please refer to your original invoice number when requesting MRA's. Merchandise sent in without a MRA# will be refused by our shipping department. All returns must be sent back pre-paid.**

2) Defective goods will be replaced, repaired, or credited to you **after** the item has been inspected by our Service Department; unless prior authorization has been made.

3) Merchandise will not receive credit if: it is no longer under warranty, has been misused, or has been altered or damaged by end-user.

4) If a unit is deemed defective and not repairable, credit will be issued. ***Should part(s) be missing from the returned unit(s), the part(s) price will be deducted from the customer's credit.***

5) Any warranty merchandise that is repairable, will be repaired and sent back to the customer; therefore, no credit will be issued.

6) There will be a 15% restocking charge on any and all items not returned in their original packaging &/or not covered by warranty.

7) No other policy, other than the foregoing, will be authorized or accepted by Yuasa International.

SERVICE POLICY

1) Warranty cards must be on file within 30 days of invoice date for a warranty repair. If warranty card is not on file, proof of purchase must be submitted when requesting service. If neither is supplied, ***all repairs will be charged to the customer.***

2) There will be a minimum charge of $120.00 for handling and inspection fee on all items sent in for inspection if no defect is found {warranty} or if customer declines service on unit(s) {non-warranty.} All such shipments will be sent back to customer "Freight Collect."

3) Units must not be previously altered, repaired, or serviced by anyone other than an Authorized Yuasa Service Center. **Any and all misuse of an item will void warranty.**

4) Before any item will be repaired, the customer will receive a written estimate which must be returned within 15 days, signed: Approve/Disapprove repair. If after 30 days we have not received authorization either way, there will be a $120.00 fee charged to the customer and merchandise will be shipped back "Freight Collect."

5) All repairs preformed a the Yuasa International Service Center: California, carry a 30 day warranty; warranty limited to only the exact items repaired or replaced.

6) All merchandise must be sent in Prepaid. All repairs covered as warranty, will have freight charges credited back to their account.

Table of Contents

Section 1: Lets get Started

Section 2: UDNC Panel Controls

Section 3: “i” Parameters and Alarm Descriptions

Section 4: Trouble Shooting

Section 5: Direct Operations through the RS-232

Section 6: Yuasa Terminal Program

Section 7: Maintenance, Service, “i” Parameters, and Table Dimensions

Section 1: Lets Get Started

Un-crating

This is a fairly minor task, yet an important one. Please make sure you maintain the warranty card, test report and manual, enclosed in your informational packet. You will notice that the warranty card provides you with center height specifications of your table, and the serial number. Save this data, as it provides us with important information, about your particular table or indexer. As a precaution, **we strongly recommend that you maintain the original crate for at least 30 days.**

Preliminary Set up

After carefully removing both the controller and table body, place them both on a clean table. First thread the motor cable connector, from the rotary table, to the flange (**marked motor**) on the controller. Make sure that this and all connections are threaded in ***all the way***. Then hook up the power cable to an AC power source, 105V to 125V. If you have the 5CA model, locate a clean (***Filtered, Regulated, Lubricated***) air line and affix it to the small air hose, leading from the air collet closer piston. If you have an air brake, attach the air hose (***w/FRL***) to the fitting. You are now ready to begin.

Simple Basic Programming

This section will only focus on simple programs, 90 deg., 60 deg., 45 deg, indexing, etc. There are three ways to write a program:

1) Use your computer, with the software supplied by Yuasa.

2) Use our Hand-held Computer Terminal (model HCT-001 or HCT-002).

3) The newest way, is to make use of the RS232 port on your computerized machining center. As each manufacturer is different in their design and macro needs, we can only state that our controller requires the ability to send and receive **ASCII** code, under the following conditions:

A) Minimum of 300 to a maximum of 19200 baud rate.

B) Either **8 bit, no parity** or **7 bit, even parity** Data Transfer.
Note: Data strings cannot contain extra characters; {ie: #, %, <>.}

C) A Carriage Return <CR> must be used at the end of each Data string.
Line Feeds <LF> are ignored by the UDNC.

Computer Programing

Since our controller accepts a standard ASCII string of commands, you can actually use any program software you have that can generate a program and download through the RS232 port. However, if you are using the YUASA TERMINAL software the following will apply.

Note: You must use a 80286/ 16 Mhz with 640k Ram or better computer with an operating system of DOS 5.0 of higher.

The Yuasa Terminal must be ran in DOS mode! You cannot run from a Windows, NT, DOS SHELL, or XTGold environments nor from these programs [DOS PROMPT] features. *{Note: Windows 95B, 95C, & 98 can run the Yuasa Terminal from their desktop. Microsoft fixed their comm port redirect functions but only on these (3) versions of Windows. 95 & 95A must restart into DOS.}*

Loading Program

If you haven't already, hook up your RS232 cable from your PC, to the RS232 port on the controller. ***Note: This controller uses a standard RS232 cable. Some CNC machines use a special cable that has pins #2 and #3 crossed to form a null-modem cable, if you use this null-modem cable, you will need a null-modem adapter. Never plug or unplug the RS232 cable when UDNC or computer is on! RS232 cables use +/- 12v transmit & receive lines which will cause harm to circuitry of both programmer and controller if these lines are active.***

1) **Loading program.** Insert Yuasa Terminal diskette into drive A, then type: **INSTALL**. This will automatically begin to load the YUASA TERMINAL. ***Note: Not all of the programs contained on the diskette will be transferred, only those required for Yuasa Terminal. Other programs are for separate operations and require you to enter separate INSTALL commands or for you to copy them directly to your hard drive or work diskette.***

2) **Configure Screen.** Confirm that the Terminal program is properly configured for your computer.

A) Com Port. Is the correct COM port selected? You have the choice of using either Com#1 or Com#2. ***{Com#1 is most often the 9 pin port whereas Com#2 is the 25 pin port.}*** Unless you are using a PS/2 style mouse, you probably have a mouse plugged into Com#1 . ***{DEFAULT = Com#2***}

B) Card Address. You have the choice of Card Address 0 - 7. Unless Specified prior to shipping, Card Address is A0. ***{FACTORY DEFAULT = 0}***

C) Baud Rate. As stated, you can communicate at baud rates between 300 and 9600. ***{FACTORY DEFAULT = 9600 BAUD}***

D) Data Bits. Again, you have the choice of 8 bit/none or 7 bit/even parity. ***{FACTORY DEFAULT = 8 bit/none}***

E) Time Out. Select the Communication Timeout that suits your computer. Older computers, 80286 - 386, may require higher numbers. 486 and better will work with default setting. ***{DEFAULT = 100}***

Type "Y" to save your new settings. The program will then connect to the controller. ***Note: Check to make sure that "MCC at A0 address? -- Yes" is show when you are hooked-up to the controller***

via the RS232 cable. If you are not connected, check your Com port, baud rate, and make sure your timeout is appropriate.

3) You are now in Terminal Mode, depress [F10] **Edit/Create Program**, the main program screen should appear.

4) Lets write our first program, **(Note: *Both YUASA TERMINAL and the controller are entirely case sensitive, all input must be in the correct format*)** type the following:

A0E (make sure you type the number zero, not the letter “O”, this addresses the card and clears the buffer, this must always be your first entry)

F1000 (This is the maximum feed rate. F1000 = 100%. Think of the last 0 as a % sign.)

G90B90 (G90 denotes an absolute movement and B90 is a 90 degree movement)

B180 (This movement is 180 absolute) Once you have input a G90, the CPU maintains absolute movements, unless you change it to incremental (G91)

B270 (270 degrees absolute)

M40 (This code allows the CPU to read the next movement in the program, in this case B0, and automatically see the M30, resetting the program and Block # .)

B0 (Machine or Work zero position)

M30 (End of program)

If you would like to save this program on the disk drive of your computer, depress the [F2] button, select the save file choice, (2), name the file, and enter. ***<u>Note: The Remote/Manual switch must be in manual mode or the program will not be stored.</u>***

Now download your program to the controller, using the “F3” key (download to card buffer and save.) ***Note: “Dwnld&Save File”[F3] will save your program, and all “i” parameters to the card. “Download File” [F5] will automatically take you from editor mode to terminal mode, where the small “s” can be entered to save the file, if desired.***

This program format can be duplicated, substituting different “B” movements, for example B60, B120, B180, B240, B300, B0, would be 60 degree movements. By simply inserting a minus sign “-” in front of the B movement you can move counter clockwise using G91 (Incremental) or stay in absolute and write the program in reverse; B300, B240, B180, etc.

There are a number of other ways of accomplishing these simple index movements, like using the loop (**G70**) function, but for a quick start, this is the easiest way to begin.

Hand Held Computer Terminal Programing

With the HCT connected to the UDNC controller, push the **[ON]** button. The screen should say “HOST FOUND.” Push the (OK) button which corresponds to the **[F]** key and you will be in the Yuasa Menu. You will find the following list of options:

OFF (Turns HCT off)

File Manager (Takes you to files that are stored in your hand held or lets you write files. ***Note: Files marked with “P##X#” are parameter files and “P” are program files. If you choose the parameter files that do not match your unit, your table will not run correctly.***)

New File (Create files to run your index and store them in the HCT.)

Terminal Menu (Diagnostic and MDI operations. See HCT manual for further explanation.)

Preferences (Lets you configure your HCT for special applications. ***Note: the factory default is A0/9600/8bit/none***.)

Card Status (Used for checking status of controllers MCC functions.)

Select **[Create File]** by using the arrow keys to highlight the name press [OK]. You will now be at a completely blank screen. Press the [α] key located beneath the **[ENTER]** bar. The α should now be seen in the top center of the screen; if not push button again. **<u>During programming this button must be active.</u>** Push the [**ENTER**] bar after each line entry. Lets write your first program.

A0E (A0 = Card Address Zero. E = Clear Buffer. *This must always start every program.*)

F1000 (This is the maximum feed rate. F1000 = 100%. Think of the last 0 as a % sign.)

G90B90 (G90 denotes an absolute movement and B90 is a 90 degree movement)

B180 (This movement is 180 absolute) Once you have input a G90, the CPU maintains absolute movements, unless you change it to incremental (G91)

B270 (270 degrees absolute)

M40 (This code allows the CPU to read the next movement in the program, in this case B0, and automatically see the M30, resetting the program and Block # .)

B0 (Machine or Work zero position)

M30 (End of program)

Now push the **[α]** key again and the **[ENTER]** bar. You will now be at a screen that has the cursor flashing behind the letter ”P”. ***Note: P = Program files. Files that have P##X2 are Parameters.*** Type the name of the file we have just created: **four** and press [ENTER]. You should now be at a screen that says: **EDIT, Download, Dwnld&Save,** and **Previous**. If you are still connected to the UDNC and the unit is on, select: **Dwnld&Save** to transfer file and save it in the UDNC.

Note: The Remote/Manual switch must be in manual mode or the program will not be stored.

Hand-held Lock-up or No Yuasa Screen:

If you are no longer in the Yuasa Program or your hand-held stops reacting, here’s what you need to do to restore the program:

1) Push the **[ON]** & **[C]** buttons at the same time.

2) Confirm that the screen now looks like this: {HOME} in the upper left of screen with the #’s 4, 3, 2, 1 running down the screen beneath it. And the black boxes contain Real/Vector/Base/etc.

3) Push the **[VAR]** key and the boxes will now contain {YUASA/MCC1/TIME/etc}. Push the button that is directly under the box that reads: **[YUASA]**.

4) The “NO HOST” screen should now appear. If not, please call the Yuasa Technical Support Dept at 800-421-9763.

Note: ***If your handheld gives a low battery warning ((•)) do not ignore it, if the batteries die or are removed the unit can only hold memory for about ten minutes.***

Simple Hook Up to your CNC Milling Machine/Machining Center.

Warning: If you are not an electrician or have not interfaced an M-function product before, we strongly recommend, having a qualified electrician, do this installation. It's like anything else, it's easy if you know how, and very dangerous if you don't.

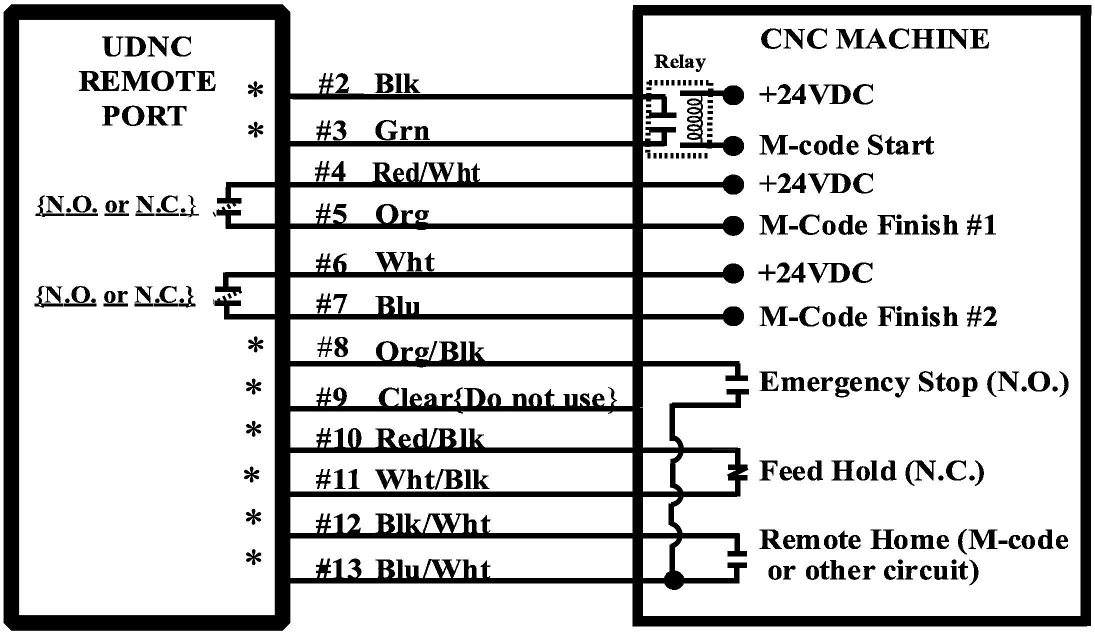
Supplied with your new rotary is what we classify as a remote cable. On the back side of the controller is a 14 pin socket labeled: **REMOTE**. You will notice that on end of the cable is a plug, this matches the stated socket, and on the other end are 13 loose wires. These wires have several functions, ***as shown in the electrical schematic drawing***, but for this quick interface, we will only concern ourselves with the two cycle start wires (#2 and #3 which are black & green) and the two cycle finish wires (# 4 and #5 which are red/white & orange).

Unfortunately, machine tool builders did not institute an industry standard when they were building their equipment, and as such provided us with no commonality to develop a single interface drawing. What I can offer you here (in addition to our electrical drawing) are some hints, on how to accomplish this task.

Warning: When installing the Remote Cable, pay very close attention to the colors of the wire, as some of the colors appear to be similar. (Red may look very much like Red/White) Connecting these wires incorrectly can create a very expensive & extensive repair.

Note: Before beginning this interface, compare our electrical schematic, with your CNC machine's electrical drawings, for actual signal information. All machines are dissimilar and may need to be installed differently.

Wiring Diagram for UDNC Remote Cable

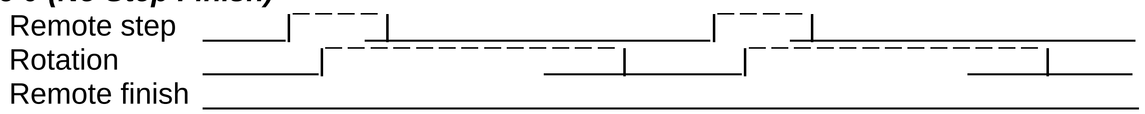


*** = No Voltage can ever be applied to these lines without damaging the internal circuitry.**

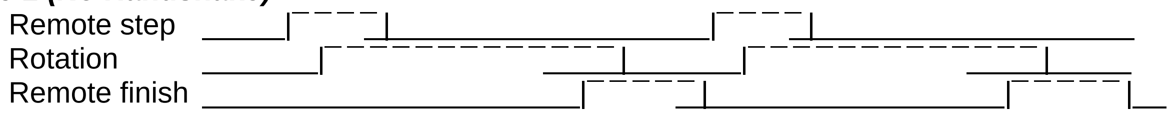
Step Finish Types

Based on Step start types:

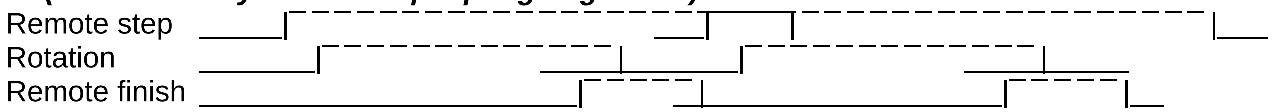
Type 0 ***(No Step Finish)***



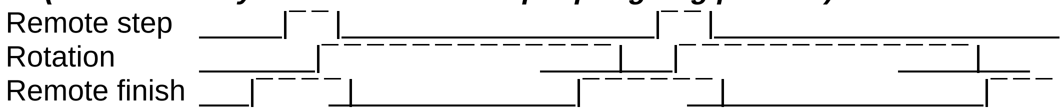
Type 1 ***(No Handshake)***



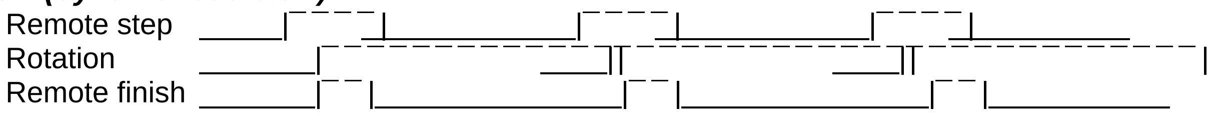
Type 2 ***(Handshake by remote step input going active)***



Type 3 ***(Handshake by second remote step input going passive)***



Type 4 ***(Synchronous Start)***



Remote Cycle Start (M-Code)

1) Determine first, where your M-code is outputting. If you're lucky the M-code has been wired out to a terminal strip. If you are very lucky, your CNC machine may also have included a relay for the M-code output.

2) If you are one of the extemely lucky people that have a relay already installed, simply hook wires #2 and #3 of our remote cable, to the normally open contacts on the relay, or to a terminal strip, if there are already wires running from the relay.

3) If your machine does not have this indexer M-code relay, you may need to install one. On most machines the M-code output, acts as a switch for a completed circuit, but some, output voltage, either 24V DC or 110V AC.

4) When adding the relay you must first determine whether you need a 24V DC or 110V AC, check your machine electrical drawings. Then make sure you have the correct type of relay, as some machines already have a receptacle in place.

5) If your machines M-code outputs voltage, say 24V then run a wire from the M-code output, either from a terminal point, or sometimes from a pin on a connector, like a Honda plug, to one side of the coil on the relay, (normally pin #7 and #8, but check with your meter) Hook another wire from the other side of the relay coil, to ground.

6) If the M-code output is a switch type (like most machines) Run a wire from the M-code, to one side of the coil, and voltage (normally 24V DC) to the other side.

7) Then always connect wires # 2 and #3 of the remote cable to the normally open contacts, of the relay, again check with your meter.

Note: To simplify this: Our machine requires a contact closure on wires 2&3 <u>(DO NOT APPLY VOLTAGE.)</u> Parameter i32 is programmable to the type of handshaking that most machine require.

<u>Remote Cycle Finish</u>

The way most machines work, once the M-start is initiated the rotary table controller will execute the next step in the program, driving the table to the desired position. Once in position, the rotary encoder verifies location and either a contact closure will occur, (if using normally open contacts) or a contact open will occur, (if using normally closed contacts). <u>Most CNC machines utilize normally open contacts, and operate with a momentary close (either 0.5 sec. or 1.0 sec.) or a Full Handshake (M Code stays active until we return a finish signal.)</u>

Again, it is imperative that you consult your electrical drawings for confirmation, but if your machine is looking for open contacts on the return signal, you would hook wire # 4 on our remote cable to +24V DC and line #5 to the M-Finish on your CNC machine.

There may be a number of different configurations to this formula. If your machine needs normally closed contacts on the return signal, Please correctly set up 'i" parameter 32. Please verify all connections with the electrical drawings, or contact your machine builder.

Note 1: ***To simplify this: Our machine gives a contact closure on wires 4&5.***

Note 2: ***For the finish type/contact polarity please check "i" parameter #32.***
For the delay time check "i" parameter #31. **{Default for i32 is 2:0}**

Section 2: UDNC Panel Controls

Front Panel Switches

1. **On and Off,** be sure to check incoming voltage, prior to powering up. The off switch, totally shuts off the entire system. Program memory and work zero position are stored in EAROM. ***Never connect or disconnect any of the cables while UDNC is on!***

2. **Program Select,** is used to access both the Main Program (**M**), and the seven additional sub programs (**0 -6**.) These programs are recognized, when either the M30, W-Z set, or W-Z return switches are seen or by simply changing the selector switch to another number setting. **(You must be in Manual Mode when you change selector switch.)**Selector switch "**M**" performs a **go to the beginning of the main buffer program.** Program selector switch 0 - 6 does a **go to the beginning of subprogram L0 - L6.** It is read and used when the W-Z set, W-Z return, Machine home, Power on, and the M30 commands are executed. It is also read when the program block number is zero. If no subprogram exists for the program selector switch, then the main buffer program will be run.

Note: ***To simplify this: The process for storing more than one set of indexer commands is to write one main program that contains up to 7 subprogram, which will fill the 8 positions on the selector switch.*** **{See Sub-Programs}**

3. **Jog +, Jog -,** the jog feedrate is controlled, either by using the **Feedrate Override,** or changing the value of i09.

4. **Feedrate Override,** controls both the jog feed and the program feed. This function operates in 10% settings, based on the value in i06.

5. **Manual/Remote.**

Manual Mode: is used for operating the front panel switches and for use of the pulse generator. In this mode, **the brake will be disabled and no remote I/O will be accepted.** In addition, the auto stepping function will be enabled either to a block number or through the program. A **Cycle Start** will cause the program to step through the each programmed move for each **Cycle Start** input.

Note: To use the brake while in manual mode, set "i12" to 1(on), then the pressing of Cycle Start will turn off the brake, index the machine, re-enable the brake. All other buttons will act as if in Remote mode. Pulse Generator must be disconnected if you select this option or severe damage to brake and motor will occur.

Remote Mode: the following front panel switches (W-Z set, W-Z return, Jog+ and Jog-, Cycle start, and Feed hold) will be disabled, and the remote I/O, will be enabled. The homing function from the front panel will be disabled. **Remote**, requires a remote cycle start to execute each program block. Going from **manual** to **remote**, will cause the program to stop at the next program block.

6. **Cycle Start/M-Z Return:** The cycle start button initiates a step start. Each depression of this button will cycle the rotary table, one individual movement. **To initiate a Machine Zero Return (M-Z), you must depress, and hold the cycle start switch for 5 seconds.** This will begin the search for the "C" channel of the encoder, which is classified as Machine Zero.

Note: On 5C, 5CA, and 132 models you must first jog to within four degrees of home, and then depress, and hold the cycle start switch for 5 seconds. This will begin the search for

the “C” channel of the encoder, which is classified as Machine Zero. These units “Home” at every 4 degrees.

7. **W-Z Set:** This **Work Zero** set button, sets the current position of the rotary unit as work zero. **Once you have depressed, this button, any previous work zero settings will be lost, and the controller will recognize this new position a work zero.** This switch is enabled only in the manual mode. Typing “**Z**” in the terminal mode will perform the motion card’s “Z” command.

8. **W-Z Return:** One depression of this button will return you to the current work zero setting. It will also cause the motion program to go to the beginning of the program, selected by the program select switch. This switch is enabled only in the manual mode.

9. **Feed Hold:** A single depression this button automatically sets the feedrate to zero, stopping all table and motor movement. Movement will resume by pushing the **Cycle Start** once.

10. **E-Stop:** Emergency stop is used to totally stop all movement and shuts down the controller. As before, work zero settings, programs and “i” variable’s are stored in EAROM.

M and G Codes.

The programming of this controller utilizes standard M and G codes, found in common CNC Machine Tool controls. Below is a listing of these codes, and their meaning. A more detailed description follows.

G-Codes.

G04	Dwell.
G17	Terminate Sub Routine.
G28	Machine Zero.
G30 to G39	Jump to Subroutine.
G70	Start Loop.
G71	End Loop.
G72	Auto Divide.
G80 to G89	Call Subroutines or conditional call with @.
G90	Absolute Positioning.
G91	Incremental Positioning.
G92	Circular Interpolation Radius, (for use with linear table)
G93	Zero Referenced Moves from second coordinate.

M-Codes

M00	Program Stop.
M18	Disable step finish relay #1
M19	Enable step finish relay #1
M20	Disable step finish relay #2
M21	Enable step finish relay #2

M Codes Continued:

M22	Disable brake on.

M23	Enable brake on.
M30	End of Program.
M36	Enable pulse generator.
M37	Disable pulse generator.
M38	Enable backlash.
M39	Disable backlash.
M40	Enable continuous motion.
M41	Disable continuous.
M42	Enable mirror image.
M43	Disable mirror image.

G-Code Description

G04, Dwell (7) digits, 1 to 8388607 millisecs:

A G04X1000 would cause the axis to remain in position one second before being ready to execute the next block. A G04 requires a step or rub command to be executed. It is a block of data that causes the "In position" register to report false during G04 execution.

Note #1: ***G04 dwells are not affected by "%". The dwell period starts at the beginning of decel. If decel is very long, the dwell time may go to zero.***

Note #2: ***For multiple axis use, when a dwell is proportional to a move size and a feedrate is required, use the "W" command.***

G17, Terminate a Group of Subroutine Blocks.

Started with L0 through L9 and terminated by G17. (See "L" command)

For example: L0 G04X500 M19 G04X1000 G17 is a subroutine that dwells a ½ second, turns relay #1 on, waits 1 second and continues at the program step after the G80 that called the subroutine.

G28, Go to Home Position (M-Z Return).

Under program control, and as specified by initialize parameters, i02, 03 and 04. If i04 is zero, home command is disregarded. When G28 is issued and i04 is non zero, the servo will move in the direction specified by i02 until the home switch is seen. It then looks for the encoder "**C**" channel. When the "**C**" channel is detected, the control executes the i03 move, terminating the homing procedure.

Note: ***On 5C, 5CA, and 132 the rotary table does not have home limit switches, it will be necessary to first jog the table to vernier zero, prior to executing G28, or the M-Z function on the control panel. This position is automatically Machine zero position.***

G30 to G39, Jump to Subroutine.

This command is similar to G80 (Call) except that it branches to subroutines L0 through L9. When a G17 or M30 is encountered, the program is terminated and the program pointer is returned to the beginning of the subprogram. G30 jumps to subroutine L0, G31 jumps to L1, etc.

In addition, both G30 and G80 commands can be made conditional by using the "@" conditional command. Refer to "@" command description for details.

The form of the conditional Jump and Call are respectively: G30@A.B and G80@A.B

Where A is an address or port, 0 to 65535, and B is a bit, 0 to 7. These commands allow the Jump or Call to be made based on the condition of an input at the specified address.

G70, G71 Start and End Loop

G70XNNNNN (5 digits) 0 to 65535. **0 = infinite**.

All blocks between G70 and G71 commands are executed as many times as specified in G70 command.

EXAMPLE (a):

A0E G70X15 F1000 G91B180 G71 M30 = Unit will incrementally move 180 degrees 15 times.

EXAMPLE (b):

A0E G70X0 F1000 G91B90 G71 M30 = Unit will move in increments of 90 degrees *without an additional start signal* at the end for the M30 command.

Note: ***A looping command may not be used inside a loop. The first loop will reject the statement of secondary loop command. (See L0 to L9 for rules governing loops and subroutines.)***

G72 Automatic Divide

G72XNNNNN (5 digits) 2 to 65535.

This command will instruct the unit to divide the next movement command into the number of equal divisions you require. The division takes care of residual encoder counts and insures that no cumulative error occurs during incrementation, and that the last movement, will execute correctly.

EXAMPLE: A0E F1000 G72X10 G91B360 M30. This program would evenly divide 360 degrees 10 times, producing individual movements of 36 degrees. In this case one additional start command will be required at the end of the M30.

To eliminate the need for an extra start command, write in an infinite loop.

EXAMPLE: A0E G70X0 F1000 G72X10 G91B360 G17 M30.

G80 through G89 (Call) Subroutines

Will execute up to 10 possible subroutines, as defined by L0 through L9, **(see "L" command.)** When encountered, the program pointer will jump to specified sub-routine and then returns to the normal program flow when a G17 is executed.

G90 Selects Absolute Data Positioning {Default Setting}

This command is seen as a default, if G91 (incremental) is not initiated. An absolute movement that is relative to zero.

EXAMPLE: A0E F1000 B180 B270 M30 = Move the rotary table, first to 180 degrees, then to 270 degrees, all referenced from zero.

G91 Selects Incremental Data Positioning

An incremental move, is one that is relative to the current position, not the zero location.

EXAMPLE: A0E F1000 G91B90 B180 M30 = First we would move 90 degrees, then an additional 180 degrees, from the 90 degree position, a total 270 degrees.

Note: ***G90 and G91 are modal commands will remain in effect until changed.***

G92 Circular Interpolation. (For linear applications)

When used with an "X" [58H], presets present position registers to values selected in this block. [G92X +/- NNNNNNN] 0 to 8388607 encoder lines.

G92X1000	sets current position = 10000 Encoder Counts
G92X0	sets current position = 0 [zero set]

When used with an “R” [52H], sets the circular interpolation radius, G92RNNNNNN. Where: NNNNN is 0 to 262139 encoder lines.

When used with a “C” [43H], sets the starting angle at which circular interpolation will be made, G92CNNNNN.
Where: NNNNN is 0 to 360000 in increments of .001 degrees.

G93 Command

This buffer command executes moves relative to initial “0” or home position, regardless of whether G92 or “Z” (zero set) was executed.

The format is G93+/-NNNNNNN where N is 0 to 8388607. This function permits the user to command the servo to go to a position relative to zero [power turn on position or home position, after “h” or “G28”], rather than a position relative to “Z” or G92.

EXAMPLE: After power turn on, position is “Zero.” If a move is executed to position 2000, then a “Z” command is sent, position will be set to zero again. Now, if a G93 is commanded, the servo will go to position 1500, which is displayed relative to where the “Z” was issued, but 500 counts from where power was turned on. If the displayed position needs to match the actual position relative to zero, then send a G93500 followed by a G92500.

This command is useful for tool changing applications, where the part reference zero may vary, but the tool change position is always the same, relative to the initial home zero.

M-Codes description

M00 Program Stop

This function causes the control to stop executing the buffer program, much like the Feed Hold button on the UDNC front panel. Use a Start, Run, Step, or Rub Out command to resume motion.

M18 Step Finish #1 Disable

Writing this code into a program block will disable the step finish signal on relay #1.

M19 Step Finish #1 Enable

Writing this will enable the finish signal on relay #1 if M18 has been used. It is not necessary to put this statement in a program, **the relay is turned on by factory default.**

M20 Step Finish #2 Disable

This command will disable the step finish signal on relay #2.

M21 Step Finish #2 Enable

This command will enable the step finish signal on relay #2 if a M20 has been used.

M22 Brake Function Disable

This command will disable the "Brake On" function in Remote mode.*{Default = Brake On in Remote Mode}*

M23 Brake Function Enable

This command will enable the "Brake On" function in Remote mode. *{Default = Brake On in Remote Mode}* ***Note: To enable brake function in Manual Mode, parameter i12 needs to be set to a value of 1. This command will not function in Manual Mode unless the above criteria is met.***

M30 End of program

All written programs must be terminated by an M30 command and carriage return. The M30 instruction is interpreted two different ways by the UDNC:

1) If the program consists of less than 448 blocks, the M30 will reset all "M" functions and feedrate instructions to their default values. It also sets the program execution pointer to the beginning of the program that is stored in the buffer.

2) If the program is larger than 448 blocks, the M30 will reset and purge all functions including the program buffer. This requires that the host computer retransmit the program before program execution can be started again. The initialize "M", "G", and Feedrate instructions are not reset, and need not be reloaded.

M*30 Rules*

Rule #1: A program without an M30 cannot be saved in the EAROM permanent memory.
Rule #2: When a program is written and terminated with the M30 instruction, a new program cannot be written unless the buffer is cleared, using the "z" or "E" command.

Rule #3: ***An M30 requires a cycle start,*** unless the move or moves before it have a M40 preceding. **{See M40}**

M36 Enable Serial Pulse Channel

This code will enable use of the *optional* pulse generator (SPG-100 or LPG-200.) The pulse generator (hand wheel) port will automatically be activated, and able to be accessed, once the "Manual/Remote" front panel switch is positioned in the manual setting. This will also disable the brake, allowing for instant use of the pulse generator.**{Factory Default}**

M37 Disable Serial Pulse Channel

This code will deactivate the handwheel port on the back panel of the controller. By default the hand wheel port is set as active.

M38/M39 Enable/Disable Backlash

As specified in the i05, backlash parameter. Default is 1600 giving you approx.: 1 degree of backlash. To activate the backlash feature, the M38 must directly follow the A0 line at the beginning of the main motion program. If Backlash is desired for all of the desired program, there is no need to add the M39 statement, if will be automatically disabled by the M30 command at the end of the program.

M40/M41 Enable/Disable Continuous Motion

This code must be used when it is desired to automatically execute an M30 command, without the need for a separate cycle input or to perform Helical operations.

EXAMPLE: A0E F1000 G90B90 B180 M40 B0 M30

With this program, the controller would know that after the B0 [work zero] block was executed, to immediately read the M30, without the need for an additional cycle input.

M00 and M30 are the only two M-codes which require a cycle, or rub command for execution; therefore the need to utilize the M40, prior to the last program move before the M30.

Another use of this function is for generating unusual motion profiles, such as hyperbolic, parabolic or arbitrary curves. This command provides a continuous motion string, blending different programmed moves and feedrates, with one simple step start command. This is extemely useful if you are attempting to develop a helical move, synchronizing an "X" axis move with a rotational move. Many times the lead, or pitch of a particular part cut will change, necessitating the need to alter the rotary feedrate and movement without stopping for another step start command. The movements and feeds entered between M40 and M41 will execute in a continuous fashion, permitting this type of move.

EXAMPLE: A0E M40 F500 B90 F250 B45 F100 B30 M41 M30

A common use of the M40/41 commands is to generate a delay before a move begins, as follows. **Note: The M41 does not have to be there, program will finish at M30 and reset without M41. If you want to give a second start command to reset program then use as above.**

M42/M43 Mirror Image Enable/Disable

This code will cause all program buffer moves to occur in reverse direction. The controller will output a negative signal for every positive move, it executes. On a two axis machine, this feature permits a part to be duplicated in all four quadrants. Note that all “i” parameters will execute independently of M42. Home direction, offset, and backlash are not affected by M42. Only motion buffer moves are mirrored, including G92, C, U and other move commands. Circular moves are also mirror imaged.

Non-Motion Buffer Control Commands

^Q	X on - turns on RS232C transmission.
^S	X off - turns off RS232C transmission.
^U	Unfreeze position, following error time click, no addressing.
^V	Freeze position, following error, time click, no addressing.
$	Reset
%	Feedrate % Override.
=	Return to pre-jog position.
?	Send status.
E	Purge buffer, prepare to accept new program.
H	Feed hold (stop)
I	Inn send motion parameter.
J	Jog +
O	Output DAC value.
Q	Stop.
R	Run.
S	Step.
Y	Send current position.
Z	Zero set.

b	Go to beginning of. This function is similar to the **“b*n*”** command: ***go to the beginning of program*.**
c	Copy RAM to EAROM.
d	Download EAROM to RAM.
e	Send error code.
f	Send following error.
h	Home.
i	Inn send stored parameter.
j	Jog -
k	Kill servo.
l	Learn position.
p	Send current position.
r	Send memory data.
s	Save program and parameters, in EAROM. **Must be entered in terminal mode**.
t	Self test.
w	Write memory data.
X	Send time click.
z	Purge buffer, prepare to accept new program.

Input Data Command Characters

Backspace	Back line pointer, one character.
CR	End of line character.
<CNTRL>X	Cancel input line.
Space	Separation character.
Comma	Separation character, conditional jump value delimiter.
Period	Conditional jump bit delimiter.
Slash	Comment delimiter.
Colon	“U” command delimiter, "i" Value delimiter.

Special, and Motion Commands

(B or X) Single AXIS Motion.

This command is used as an axis identifier. For most applications our use will be for rotary movements, but this controller can also be used for linear moves. For either axis movement, i.e., B180 (180 degrees) or B2.5000 (2 1/2" linear move using “T” instead of “F”) we utilize the “B or X” for the movement.

For a reverse direction move, simply input a minus sign, i.e., B-180, would be 180 degree move in the counterclockwise direction.

(F) Feedrate Setting

The (F) setting in the program, acts as a percentage of feed, based on the value setting of parameter i06.

For example: F100 sets the feedrate to 10%, of the maximum feedrate, value of i06. F1000 would set the feedrate to 100% of i06. Think of the last 0 as a % sign.
The “F” command is modal, and will remain in effect until changed.

(T) Time Mode Programming

Where: NNNNNNN is 1 to 8388607 in millisecs.

This command may be used whenever a specified move time is desired rather than a feedrate. Its primary use in the controller is if a timed programming is used on multiple axis, regardless of the move sizes on each axis, all moves will terminate simultaneously, thereby, providing an 8 axis linear interpolation capability.

Feedrate “F” programming, which is modal, can be used interchangeably with time programming “T” in any program. Care must be taken to re-specify the desired feedrate by placing an “F” command in the program immediately after the last “T” move.

In “T” mode, the automatic backlash (M38 /M39) is not permitted.

The time round off error for very small and slow moves is traded in favor of velocity round off error. This permits multiple controller systems to remain synchronized throughout the full dynamic range of the controlled operation.

The velocity round-off is small, approximately equivalent to 200 encoder counts, for a 2 hour move: i.e. if you tell the controller to move 2,000,000 counts in two hours, it will terminate the move exactly on time, but may make a 100 to 150 encoder count move at the very end of the time period, instead of equally spreading the move, thus creating a small velocity jerk. Time and position remain accurate. This error rapidly diminishes as the move times become shorter. The same move made in 20 seconds, would have an undetectable velocity error of ½ of an encoder count.

Time Constant: 1 second is equal to a “T” value of 2014

If you wanted to move 360 degrees in 1 minutes, you would first multiply 60 (seconds) and 2014 which equals: 120840. Your “T” value would appear as: T120840B360.

Helical Time Feed Formula:

To figure out what “T” value to use for a helical cut, use the formula below or use the TIMEMODE program located on the Yuasa Diskette.

1) Determine the **Radius** of the **Workpiece.** Radius is equal to the distance from the center of the workpiece to it’s outer edge minus the depth of the cut. *{If you measure your workpiece to be 4" in Radius and you are doing a 1/2" cut, your Radius = 3.5"}* **This is R.**

2) Determine how far you want the workpiece to rotate, {i.e.: 45 degrees.} **This is D.**

3) Determine how far the X axis cut across the workpiece is going to be, {i.e.: 12".} **This is X.**

4) Determine how many inches per minute you want the workpiece to travel, {i.e.:6".} **This is IPM.**

5) $\frac{\Pi \cdot R \cdot D}{180} = I$ (Π = 3.1415926536)

6) $\sqrt{I^2+X^2} = I_T$

7) $\frac{120840 \bullet I_T}{IPM} = T$

Using the examples stated we will figure the Formula as follows:

3.1415926536 X 3.5 X 45 / 180 = **2.74889**

$\sqrt{2.74889^2 + 12^2}$ **= 12.310824**

12.310824 X 120840 / 6 = **247939.98 or 247940**

Your programed “T” value would look like this: **T247940B45**

For helical timing of axes, set i32 to 4:0 or in the program insert I324:0 after A0E.

Note 1: ***In “T” mode, the servo computes a feedrate based on the specified time, and the size of the move for each block of data.***

Note 2: ***The “T” mode is a modal command and takes effect in the block where it appears, and remains in effect until another “T” or “F” command is given.***

Note 3: ***Care must be taken to specify a “T” time which will result in a feedrate within the machine’s capability. Otherwise, a Position Following error will be generated.***

Note 4: ***The accel and decel parameter, (“i08”), remain in effect during “T” moves.***

Note 5: ***The “T” time specified is the time from beginning of acceleration to the beginning of deceleration; therefore, in some applications it is necessary to make allowances for the decel time which is not included in the “T” move. To calculate decel time, please refer to “i08” parameter description.***

Note 6: ***If a “T” time is specified without a net motion occurring, it will be treated as a dwell “G04” command, except that the “%” command will alter the dwell time. (See G04 for further details.) The “T” time dwell should be used in synchronized motion applications, since it is more accurate than the G04.***

C Circular Move.

In conjunction with the G92RNNNNNNN, which specifies the radius, and the G92CNNNNNNN, which specifies the starting angle, the “C” command causes the controller to execute a motion with sinusoidal velocity profile. For example, the following program:

A0	**addresses the card.**
Z	**sets current position to zero.**
z or E	**clears buffer.**
F1000	**sets the feedrate to 100% of the “i06” value.**
G92R10000	**sets the radius to 10000.**
G92C0	**sets the starting angle to zero degrees.**
M30	**end of program.**

Send a “Rub” or an “S” or an “R” to execute the program. Will move the servo motor from “0” position to 20000 encoder counts with velocity reaching maximum peak at 10000. Having reached 20000, the motor will reverse and return to zero position.

To make the motor go to -20000 and back to “0”, specify a starting angle of 180 degrees by changing G92C0 to G92C180000.

The range of G92R is from 0 to 262139 encoder lines; and G92C is from 0 to 360000, in increments of .001 degrees or 3.6 arc. seconds, of resolution. Both the radius and the starting angle are initialized to zero on power turn-on, or reset. Once specified, the radius will remain at its specified value until changed. However, the starting angle will change incrementally as moves are executed. For example, if G92C0 is specified and a C90000 move (90 degrees) is executed, the starting angle will now be G92C90000, for the next move. If a different value is required, it must be entered into the program.

The “C” command’s range is from 0 to 8388607 and, therefore, allows up to 23.3 revolutions to be executed in a command line. The use of a loop (G70) command allows execution of continuous rotations.

It is possible and sometimes desirable to blend a linear, and a circular motion together. For example:

A0	addresses the card.
Z	sets current position to zero.
z or E	clears buffer.
F1000	sets feedrate to maximum of i06 value.
G92R500	sets radius to 500.
M40	enables continuous motion.
C90000	Circular move, 90 degrees to 500.
B9500	Move to 9500.
M41	Disable continuous run.
C90000	Move circular, to 90 deg. to 10000. Motion stops at 10000, G92 is now 180000.
M40	Enable continuous run.
C90000	Move circular 90 degrees.
B500	Move to 500.
M41	Disable continuous motion.
C90000	Move 90 degrees.
M30	End of program. Motor is at 0, and G92 is at 0.

The first step command will move the motor from 0 to 10000 by sinusoidally accelerating from 0 to 500, running continuously to 9500 and decelerating sinusoidally to 10000. The next step command will return the motor to zero following an identical path and accel/decel behavior.

Note: ***i08 (accel and decel) parameters should not be used when blending a linear move with an interpolated circular move.***

W Proportional wait or dwell

Even though the controller is equipped with a G04, dwell command, and a delay time proportional to % (feedrate override), when using the “T” (time mode) with a zero move: it is desirable, especially in multiple axis systems to provide a delay or wait time which is proportional to both the “%” command and the machine feedrate which includes “F” (feedrate and i06, machine feedrate parameter). The “W” command accomplishes this.

“W” must be followed by a number from 0 to +/- 8388607. Negative or positive number may be used, which is handled as follows:

W2000 will generate a 2 second delay if % is 1000, “F” is 1000, and i06 is 1000. Should any one of these values be reduced to 500, the delay will increase to 4 seconds.

In all cases, the number following the “W” command will take the same amount if time to execute as a move of the same size. W1500 will take the same amount of time as an B1500 (regular move), regardless of the “%”, “F” and “i06” values.

In a multi-axis system, the “W” command can be used on the idling axis while the other axis is/moving. A program can then be executed continuously with all stops and moves blending accurately.

When specifying idle time on one axis when the other axis is making a circular move, it is necessary to specify a “W” command size equal to the path of the other axis, not just the net move. For example, if an axis is making a 90 degree circular move, with a radius of 5000, the specified “W” wait time should be: [5000 X 2"PI”R] divided by 4 = 7854

R Radius Set, G92R

Unlike the “R”, Run command which starts the control, executing blocks consecutively without a stop, and can be used as a stand alone, on line non-buffer command, the “R” command, when used with the G92 command, is used to set the radius, for a circular interpolation move.

G92RNNNNN specifies the radius of the move and can be from 0 to 262139 encoder counts. This means that if the system is using 1000 encoder counts per inch, the largest circle can be 524.278 inches in diameter.

The G92R command is modal and will remain in effect until changed.

n No Operation

This command, when entered in a program will have no effect on operation, it occupies one block of storage in the buffer.

# Feedrate

This command can be executed from the controller card buffer. It is exactly equivalent to the on-line (non buffer) % command, and is used to change the control’s velocity on the fly.

z or E Purge Motion Buffer

Allowed only in stop. Clear buffer for new program data. Should always be used before new program is entered. Also, clears all M-functions and “I” parameters.

Z Zero set

Allowed only in stop. When received by the control card, will cause current position to be “0.0000” position.

Note: ***After zero set, if “p” or “Y” command (send position) is used, the position may not necessarily be “0” due to amplifier offset and other reasons. Adjust amplifier “balance” or activate the i13 parameter (bias compensation) to eliminate offset.***

h Home

Allowed only in stop. Go to home position will take place as specified by i02, i03, and i04 parameters for direction, speed and offset. If i04 is zero, homing will not take place.

Note 1: ***To recover from feed hold during home or jog functions, repeat home or jog command.***

Note 2: ***To recover from (+) or (-) machine limit switch during home or jog commands, clear limit switch by jogging, then repeat command.***

Note 3: ***A home command may also be generated by using an external switch to ground jog (+) and jog (-) inputs simultaneously.***

b Go to Beginning of Program. bn, (n=0 to 9)

Resets all M-functions and “I” parameters, but leaves all set up parameters (i) undisturbed. Allowed only in stop, the “b” command with a number of zero following is used to return to main program. Specifying a number from 0 to 9, sets the program pointer to one of ten subroutines and allows them to be executed on a stand alone basis, as if they were a complete independent program. The G17 (end of subroutine) is then interpreted as an M30 (end of program).

p or Y Send Current Position

The control sends a sign and 7 digit position data to host. Data is in ASCII with sign first (+ or -) followed by 7 digits that represents the units current position. Data transmission is terminated with a carriage return and a line feed. This command can be used with certain CNC for requesting “Current Index Position” when used with Macro RS232 direct operations. *{e.g.: If during a slow rotary move of*

33 degrees, you send "YYYYY<CR><LF>" the MCC will pole the encoder and send back the current position +23.2333 to your CNC or computer.}

Note: ***i15(1) or i15(2) parameter will place the control in a mode where the "f" and "p" or "Y" commands will transmit data as a 2's complement binary data instead of standard ASCII characters. See "i15" for details.***

s Save Program in EAROM [Permanent Memory]

This command is used to save a program into the EAROM permanent memory. The "i" parameters, are saved, together with the program.

Upon power turn on, or initialize, the EAROM program, along with its parameters are automatically loaded, and ready for execution.

Rules for "s"aving

Rule #1: *You must be in terminal program mode, and the function switch must be in "**Manual**" for the program and/or "i" parameters to be saved.*

Rule #2: *"s" is active only in the stop mode.*

Rule #3: *Each program block requires approximately 50 millisecs, to be saved.*

Rule #4: *Save operation stops when M30 is encountered.*

Rule #5: *If only set up parameters are to be saved, a program must be written or in the buffer containing an M30, in order to allow the save to be successful. For example, an F1000 M30, is an adequate dummy program.*

i Send Parameter Values. (nn)

Where nn represents a value from 00 to 99 and is used to cause the controller to send the specified parameter value. The control will send the appropriate value, 1 to 7 characters, followed by a <CR> and <LF> character. **{e.g.: i32 [ENTER]}**

This function can be used in either the hand terminal, or a computer, but most NC controllers do not have the ability to transmit lowercase letters.

A0 to A07 Control Card Address

Up to 8 controllers can be programmed, from our custom software package. Whenever, you begin a program, you must address the particular controller, which you're are writing a program to. If you only have one controller, than the card address, would be A0. It is also strongly recommended to clear the card buffer, every time you download a program. A program must always begin with an "A0E" (address card, clear buffer), and end with an "M30", (end of program).

Note: ***You must address the card, by typing an "A0" in terminal mode, prior to writing your program***.

“L” Functions

L0 to L9, defines one of ten possible subroutines, to be invoked by G80 to G89, or by G30 to G39. Using a “b” on line command, which executes the subroutines L0 to L9, with a bn value n=0 to 9 for b0 to b9.

Note: ***When writing a program, all subroutines, if any are used, must be entered first, followed by the program itself.***

General rules for subroutines and loops are:

1) Subroutines must be entered first, but their order can be random.
2) Subroutines used inside other subroutines, must be entered before the user subroutine is entered.
3) Subroutines may be inside subroutines up to 4 levels deep.
4) Loops may be used inside subroutines.
5) Loops may ***not*** be used inside loops.
6) Subroutines may be used inside loops.
7) Subroutines are defined by the “L0” to “L9” commands and terminated by the G17 command.
8) Loops are started with the G70 XN command and terminated by the G71 command.
9) Subroutines are invoked in the program by the “G80” to “G89” commands, which respectively call the appropriate routines or the “G30” to “G39” commands which respectively jump to the specified routine.

EXAMPLE:

```
A0E         /Address card A0 and clear the buffer/
L0          /Subroutine 0/
G90B90      /Absolute index to 90°/
B0          /Return to Work Zero/
G17         /Terminate Subroutine/
L1          /Subroutine 1/
G90B180     /Absolute index to 180°/
G80         /Go to Subroutine 0/
G17         /Terminate Subroutine/
L2          /Subroutine 2/
G90B30      /Absolute index to 30°/
G81         /Go to Subroutine 1/
G17         /Terminate Subroutine/
L3          /Subroutine 3/
G90B270     /Absolute index to 270°/
G82         /Go to Subroutine 2/
G17         /Terminate Subroutine/
F1000       /Set Feedrate to 100%/
G80         /Go to Subroutine 0/
G81         /Go to Subroutine 1/
G82         /Go to Subroutine 2/
G83         /Go to Subroutine 3/
G70X2       /Start Loop 2 times/
G83         /Go to Subroutine 3/
G71         /End Loop/
M30         /End Program/
```

Program sets feedrate to max. i06 value, calls subroutine #0, then subroutine #1, then subroutine # 2, then subroutine #3, starts a 2 times loop to execute subroutine #3, end program.

This program if run with the Program Select switch in the "M" position would index as follows: 90° 0° 180° 90° 0° 30° 180° 90° 0° 270° 30° 180° 90° 0° 270° 30° 180° 90° 0° 270° 30° 180° 90° 0° End.

Subroutines can be treated as subprogram, in that by using the G30-G39 jump command, or the “b” (beginning of program) b0 to b9, a subroutine can be selected and executed on a stand alone basis. Subroutines can be called or jumped to conditionally or unconditionally. A call will return to the main program, when the G17 is seen. A jump will not return but will instead treat a G17 as M30, or end of program.

Subroutines can be executed directly as a “stand alone” program by using the bn command (return to program start) where “n” can be used to specify the subroutine (program) to be run. “n” can be 0 to 9, if no “n” is used, the main program will be executed.

When G17 is executed, the program pointer will return to the beginning of the main program. Therefore, to execute a subprogram again, the “b” command must be sent.

A total of 10 subroutines (subprogram) may be stored as well as the main program. Generally, the main program should be used as a director, or a selector of subprogram.

The front panel, program selector switch, automatically identifies, either the “M” main program, or the 7 subprogram, 0-6.

Program EXAMPLES.

In this first example, we have first set the feedrate to 100% of the value of i06, then moved incrementally 90 deg., from zero, then incr. minus 180 degrees. Then we switched to absolute (G90) and moved to the 170 deg. position, relative to zero. Then back to incr. (G91) minus 95.3025 deg. then home. You will note that we used a G90B0 for work zero, you **must** be in ABS(G90) mode for a work zero return.

```
A0E
F1000
G91B90
B-180
G90B170
G91B-95.3025
M40
G90B0
M30
```

In this second example, we wanted to rotate 360° continually (without stopping) at a 50% feed, again based on the value in i06.

```
A0E
F500
M40
G70X0
G91B360
G71
M30
```

In this third example, we are positioning in 15 degree increments.

```
A0E
G70X0
F1000
G91B15
G71
M30
```

This example illustrates, how to use the auto divide (G72) dividing 360 degrees into 10 equal parts.

```
A0E
F1000
G72X10
G91B360
M30
```

Feedrate Table

The “F” value, when entered into a program block, represents a percentage of the value in “i” parameter 06. For example, F1000 = 100%, F100 = 10%, F10 = 1%, and so on. In addition, the feedrate override switch on the front of the controller will also affect the feed in percentages of 10% increments. *For example:* if you set i06 at its maximum setting, and used an “F” value of 1000, but set the feedrate override at 90% your feed would be 90% of the maximum feed capabilities. If you set the “F” value at 500, and the feedrate switch at 50%, your resulting feed would be half of the maximum, halved again (25%).

This control offers tremendous flexibility, specially in the feed department, permitting extemely slow speeds, up to fast rapids, even continuous movements at rapid speeds, if you like.

There are two formulas for calculating the feedrate.
One is easy and reasonably accurate, the other is more difficult but extemely precise.

#1) Easy Formula: ***RPMs X Multiplying Factor(MF) = i06 setting***
For 60:1 gear ratio **(*130*)**: **MF=2000**
For 72:1 gear ratio **(*5C, 5CA, 132, 140, 170*)**: **MF=2400**
For 90:1 gear ratio **(*220*)**: **MF=3000**
For 180:1 gear ratio **(*280, 320, 400*)**: **MF=6000**

EXAMPLES, **Assume an “F” value of 1000:**

GEAR RATIO	**REQUIRED RPM’S**	**MULTIPLYING FACTOR**	**SET I06 PARAMETER TO:**
72	5	2400	12000
72	26	2400	62400
72	33	2400	79200
72	62	2400	148000
90	2	3000	6000
90	18	3000	53999
90	25	3000	75000
90	30	3000	90000
180	2	6000	12000
180	7	6000	42000
180	10	6000	60000
180	15	6000	90000

EXACT Formula: Number of encoder lines, times the encoder multiplier, divided by the encoder counts per second, divided into 60 = Motor RPM,. divided by the gear ratio = table RPM

<u>EXAMPLE #1</u>, (For 5C, 132, 140, 170) Assumes an "F" value of 1000

2000 X 2 = 4000, divided by 300,000 = .01333, divided into 60 = 4500RPM (motor max), divided by 72 = 62.5 RPM table speed.

Note: ***With this example, your minimum positioning increment (resolution) would be 0.00125 deg.***

<u>EXAMPLE #2</u> (220) Assumes an "F" value of 1000

2000 X 2 = 4000, divided by 300,000 = .01333, divided into 60 = 4500RPM (motor max), divided by 90 = 50.00 RPM table speed.

<u>EXAMPLE #3</u> (280, 320, 400) Assumes an "F" value of 1000

2000 X 2 = 4000, divided by 180,000 = .0200, divided into 60 = 3000RPM (motor max), divided by 180 = 16.66 RPM table speed.

Note: ***The speed of movement, and resolution are very closely tied together. Keep in mind, the maximum amount of encoder counts, per second, the CPU can count is 300,000. The faster the motor rotates, the more encoder counts it generates, per second. Given this 300,000 maximum, the slower you rotate, the finer the resolution. As you go faster, you lose a little resolution. Your rotary unit is shipped with the times 2 multiplier. If you desire the times 4 multiplier, it is easily changed. A couple of "i" parameter settings, and a simple jumper switch. You can do it in the field, but give us a call, and we will work you through the first time, over the phone.***

Just as your new rotary, can move very fast, it can also move very, slow. Again it's a matter of changing the "F" value in your program, and altering "i06". The same formula above still applies, just input smaller numbers.

First let me confirm to you, the minimum settings, for the "F" and "i06". (For more info on these please see the "F" command, and "i06" parameter). "F1" is the minimum setting, which = .1% of the value in "i06". "i06" set at 1, would = one encoder count/ per half second. (My recommended lowest setting would be 250)

EXAMPLE of above: 2000 X 4 = 8,000 div. by 500 = 16 div. into 60 = 3.75 (motor rpm) X by .01 ("F" setting 1% of i06) = 0.0375 div. by 72(Gear Ratio) = 0.0005 rpm

SECTION 3: i PARAMETERS AND ALARM CODES

“i” Parameters.

The initialize, or set up parameters are used to tailor the operating characteristics of the control to the system in which it is being used. The EAROM will save any parameter changes, regardless of power shut down, provided the “s”ave, is entered. Don’t forget, you must be in the program terminal mode, with the Manual/Remote switch set to manual.

i00 Set Following Error.

Factory setting, -250. This allows a maximum deflection of .25° before sending a “Pos Follow” error alarm. {Changing this may cause your unit to not properly function. The factory setting is appropriate for 99.7% of all users needs}

i01 Set in Position Band.

Factory setting, 2. This is reflective to the motor/encoder accuracy and is preset to our motor specifications. **Do Not Change!** Incorrect values will seriously damage the encoder and/or controller, voiding your warranty.

i02 Set Home Direction.

One digit, “0” or “1”. Specifies direction in which home will take place. “0” is negative, and “1” is positive direction.

i03 Set Home Offset.

Factory Set: **3000**. During the homing operation, immediately after the encoder “C” channel is detected, this parameter will cause the axis to make a move by the specified amount and direction. The final position will be interpreted as the axis “zero” position. For homing instructions, please see “G28” and “M-Z” instructions.

i04 Set Home Feedrate.

Three digits, where NNN is 0 to 999 and represents the % of maximum feedrate, of “i06”, the axis will be moving, during the homing operation. An i04 value of 500 would be 50%, such as a value of 250 would be 25%, of the maximum value, set in i06.

i05 Set Backlash Compensation.

Five digits, based on encoder lines, and specifies the direction in which, and the number of encoder lines the axis will overshoot the desired position, and return. Approaching position always from the same direction, eliminates the effects mechanical looseness on positioning accuracy.

A “+” number causes the axis to overshoot position and return, when the axis motion is in the “-” direction. A “-” number will cause the axis to overshoot, when the axis is in the “+” direction. The sign of the backlash number indicates the direction, in which backlash take up will occur.

An i05 value of 1600 (72:1 G.R.), would overshoot by one degree, then return.

i06 Feedrate Set.

The maximum setting is 100,000 (60:1, 72:1 & 90:1) [90,000 on 180:1]which equals 100,000 [90,000] encoder counts per half second. This is the maximum amount of encoder counts, per half second, that the CPU can count, without causing a fault This setting will not take effect, until the “F” program feedrate is read. The “T” command generates its own feedrate, and is not affected by i06. Please refer to the feedrate table for more information, on this subject.

i07 Set Servo Loop Gain Factor.

3 Digits, Gain Control, i07NNN.

Where NNN is 1 to 255 and controls the responsiveness of the servo loop in responding to a position error. The larger the number the "racier" the servo loop becomes. This parameter, together with the "Accel/Decel" factor: i08, can be used to critically damp the servo loop. It is recommended that the gain be gradually increased or decreased until satisfactory response is obtained.

i07 is normally used to eliminate overshoots or slow position creep. Once i07 is optimized, then i08 can be used to set the desired acceleration time. {Factory setting is recommended!}

Note #1: ***It is important that the servo amp and motor/load combination be optimized before attempting to set i07.***

Note #2: *i07 may be executed in the Program Buffer by using I07.*

i08 Set Accel/Decel Factor.

Range 1 to 32767 with 1 being the shortest amount of time until indexer reaches full speed, and 32767 being the greatest amount. **Average value should be: 800 to 1200.**

Note: Less than 400 may make the lock up or give a "Pos Follow" error.

i09 Jog Feedrate Set.

Similar function as i06 except controls jog feedrate instead of program feed. The maximum setting must never exceed i06. Recommended setting 1500 to 15000.

i10 Set maximum positive travel limit.

7 digits. Measured in encoder counts. This parameter will prevent the servo from moving beyond the specified positive direction limit, even if it is commanded, or instructed to do so. If the travel limit is exceeded, the control will output an error signal and stop motion. Moves in the opposite direction, remain active.

Setting i10 to 288,000 would set the limit at the 180 degree position (72:1). Default -1 (off)

Note: ***To use this feature i20 should be set to "0"***

i11 Set maximum negative travel limit.

7 digits. Same as above, except controls travel in the negative direction. Default -1 (off)

i12 Enable/Disable Cycle start button in Remote Mode.

i12 set at 1 will enable the cycle start in the remote mode, allowing the brake to be used from the front panel. Setting this to 0 will disable this feature.

i13 Automatic position bias generator time constant.

From 0 to 127. Factory set, 90. Sets the rate at which the servo loop examines the system's position and following error drift and generates a bias term for compensating it. 0 (off), 1 = minimum compensation, 127 = maximum compensation.

i14 Velocity feedback gain.

Factory set, 20. Range 0 to 255. 0 (off). Controls the amount of negative feedback proportional to velocity. **Note:** *i14 may be executed in the Program Buffer by using I14.*

i15 Binary data mode.

Factory set, do not change.

i16 Pulse Generator (handwheel)

1 digit. This parameter is factory set to "0", to enable the pulse train to change desired as well as actual position. For example, rotating the pulse generator, 1600 encoder lines, would cause the rotary to move one degree. In this mode the pulse generator is automatically disabled by a programmed movement.

"i16" set to "1", causes the generator input to change actual position only, and does not disturb the desired position. For example, if the handwheel encoder, is rotated 1600 lines and the table position is zero, the control will move the motor 1600 lines, but the controller's position will remain at zero. This function is mainly used for setting offsets. When "i16" is set to "2" or "3", it creates a **special handwheel mode**, where the handwheel (pulse generator) input is used to create a time base for the control card. The result is that the handwheel can now be used to control the motor velocity as a function of the generator rate, and where the move distances can be used to change these rates. This mode can be used to generate **nonlinear lead screw patterns, or any other arbitrary motion patterns.** Also, if a common handwheel input is used on any number of controllers, they will execute their programs in precise synchronization.

i17 Velocity feed forward gain.

Note: *Should match i14.*

i18 Automatic bias generator mode control.

Factory setting, do not change.

i19 Pulse generator (handwheel) scale factor.

5 digits, Factory set. Controls the multiplication factor applied to the handwheel input.

Note: ***The handwheel model # LPG-200 will automatically adjust the scale factor.***

i20 Linear/Circular position range.

Factory set for unit gear ratio.

Gear Ratio	Encoder
60:1	-240000
72:1	-288000
90:1	-360000
180:1	-720000

i21 Torque limit control.

Factory set, do not change.

i22 MDI mode control.

One digit. “0” is for standard operating mode, and “1” places the control card in a manual data input mode where it is capable of executing all buffer commands directly and immediately upon reception. The MDI mode can be invoked at any time, and does not place any data into the buffer. This is considered an “on-line” command feature.

To enter MDI mode, type i221[CR]; the control is now ready to accept any move. G & M commands are executed upon reception of the [CR] command. Make certain that the feedrate is not zero, by typing an F1000. This mode can be used by a host computer, at any time without using the buffer.

To exit MDI mode, use the “S” step or the “R” run command, or type i220[CR]

This function is also used to “Allow a remote step via RS232 terminal” see this for more details.

i23 Automatic step delay & step to block number.

First value (the delay range 0-255 = 16msec. default = 0)

This is the pause time between automatic steps in the Manual mode.

The second value is the block_number which will be auto stepped to. The block_number range is 0-65535, default = 0.

The format is “i23delay:block_number”

This is an excellent parameter for recovering to a particular block number, after say braking a drill, or a machine crash. It allows you to enter a block number say, #49 and auto step to that block, without having to cycle every step to get there.

The procedure would be as follows: first W-Z return, then enter “in terminal mode” i2316:49[CR] This would identify “i23” set the delay time, then auto step to block #49 and stop.

If i23 is zero there will be no automatic stepping of the program.

If i23 **block** number is zero, the program will auto-step to the end of the program.

i23 is zeroed when a “Q” quit command is executed.

i24 Entry and display scale factor.

Factory set for gear ratio to encoder value. See chart below to match your unit.

Gear Ratio	i24 Value
60:1	2000:3
72:1	800:1
90:1	1000:1
180:1	2000:1

i25 Feed hold decel mode control.

Factory setting, 1 do not change.

i26 Prompt mode control.

Factory setting, do not change.

i27 Gain multiplication factor.

Factory setting, 3 do not change.

i28 Work Zero set position.

DO NOT CHANGE THIS PARAMETER, SEE “i” PARAMETER LIST.

Units are in encoder lines, with a range of 7 digits. For example, a setting of 144,000 in this parameter, would retain the work zero set position at 90 degrees from zero.

8,000 encoder counts/per rev. X 72 (gear ratio) = 576,000 div. by 4 = 144,000.

The work zero position is saved after power shut down in the EAROM.

i29 Brake off delay, before step.

Range = 0 -255, LSB = 16 msec., default is 32 = 512 msec.

0 defines a system with no brake.

The servo integrator gain will never be zeroed, if i29 = 0

Factory Set: Do not change!

i30 Brake on and remote step finish delay

Range = 0 - 255. LSB = 16 msec., default = 32 = 512 msec.

This “i” variable allows the delay of turning on the brake, and remote step finish signals. In order for the “i30” time to be used in determining the delay of the step finish signals, i29 must be a non zero value (at least 1).

i31 Step complete delay.

Range = 0 - 255, LSB = 16 msec., default = 32 = 512 msec.

Allows the pulse length adjustment of the remote step complete signal for the “i32” types 1 & 3.

i32 Step type, and polarity.

The format is “i32 type:polarity”

Type: 0 - No Step Finish. ***To see a visual representation of i32 types, refer to page 7 of this manual.***

1 - No handshake with a delay set by “i31”

2 - Handshake by remote step input, going low.

3 - Handshake by second remote step input, going high. The remote step should stay set until it sees the remote step finish change state, or remain set for at least the time set in “i29”

4 - Step Finish is synchronous with motion start and resets the step finish when the Remote CYCLE START resets.

Note: I32 ***is a motion program buffer command. It can be put into a motion program to change the Step Finish Type signal during the program.***

The type, must be the same for the step finish #1 & step finish #2.

Polarity Chart

Value	Step Finish #2	Step Finish #1
0	0	0
1	0	1
2	1	0
3	1	1

Note:
1 = Close to open to close.
0 = Open to close to open.

The polarity can be independent for step finish #1 and step finish #2. One can be set to "1" and the other to "0".

<u>Allow a remote step via an RS232 terminal.</u>

A. Proceed as follows, for using your moves.

1. Go to MDI mode "i22" = 1 (i221[CR])
2. Type the next move value with M18 (B45 M18) to disable the step finish #1 signal.
3. Type the next move value with M19 (B90 M19) to re-enable the step finish #1 signal.
4. Turn off MDI mode "i22" = 0 (i220[CR])

B. Proceed as follows for using the current control cards buffer moves:

1. To go into MDI, turn off step finish #1, and step by typing "i221 M18 S"[CR]
2. Typing "S" will step through remaining moves without step finish #1 signal.
3. To re-enable step finish #1 type i221 M19 S [CR]

The block number will be incorrect because, M18 and M19 commands were added.

Diagnostic Analysis.

This controller has a number of failure alarms, designed to troubleshoot problems, or just inform, if you entered an invalid input character. The front panel display, will indicate a numerical or letter type, of alarm code, as well as a brief “worded” description.

Alarm Code

	Failure Description
AXIS LIMIT	Motor or Cable defective or disconnected.
8032 CPU	Internal RAM or CPU Failure
PRG BUF RAM	EXTERNAL RAM fault
EAROM	EAROM failure
CMD FORMAT	(a) Invalid command or more than 80 characters input. (b) No <CR> after M30. (c) Tried to (s)ave while in Remote.
POS FOLLOW	Position Following Error
PROG LIMIT	Motion limit failure
RS232 XMT	Transmission error (check cable type & verify lines 2 & 3.)
EAROM XFER	EAROM save or load failure
U MOVE	"U" move failure
BRAKE CLAMP	Brake clamp did not release. (Check if airlines are crimped &/or solenoid wires.}

Section 4: Trouble Shooting the SUDX & UDNC

Communication Problems:

A) My computer does not communicate with the UDNC.

1) Confirm what type of cable you are using, Standard or Null, and that communication cables are plugged into proper ports. On most computers the 9 pin port is configured as COM 1.

If this is true on your computer, use a standard 9 pin to 25 pin cable and connect to the 25 pin RS232c port on the UDNC. Note: If you have a 9 pin to 9 pin Null Modem cable connect to the 9 pin RS232c port on the UDNC.

If this is not true on your computer, identify which port it is (ie: COM 2.) When you first access the Yuasa Terminal it will ask you to confirm your Port Settings, choose the appropriate port and confirm your changes.

If the 9 pin is COM 1 and it is being used by a mouse, use the 25 pin RS232c port on your computer, confirm its port numbers and specify this port when you first access the Yuasa Terminal. If the 25 pin is also in use, you must terminate one of the devices occupying a port.

2) Confirm that you are in DOS and not in Windows 3.1, 95, or NT. The Yuasa Terminal will not work properly in a multi-tasking environment. Do not use a Windows DOS prompt! You must be completely exited into DOS and have no other applications running.

3) **My computer communicates with my CNC but not to the UDNC.** Your CNC may require lines 2 and 3 of the cable to be crossed {Null Modem.} Use a Standard cable or attach a Null Modem adapter to the end of the cable.

4) **I don’t understand how to use the Yuasa Terminal Configure Menu.**
a) Confirm COM 1 or COM 2. (see #1)

b) Confirm BAUD rate. The default is 9600. This is an industry standard for most 286 and better computers. If you are unsure of what baud rate to use, leave it at 9600.

c) 8 bit or 7 bit. Our default is 8 bit. If you need to use a 7 bit rate, you will also need to configure the dip switches inside the UDNC. {See Dip Switch Configure.}

d) Millisecond time out: 35 (default) will work with 386/486x20Mhrz or better processors. If you are using a 286 or slower than 20Mhrz 386/486, increase the time out to 50ms. 50ms is a good starting point, your true time out may be slightly higher or lower.

e) Confirm the setting are correct and enter “Y” to save. Your next screen will list the cards addressed {A0 - A7} and confirmation of communication {Yes or NO.} If everything is correct A0 should say YES.

5) **I've tried everything and still no communications.** Unplug the power cord and remove the cover from the UDNC. Check the ribbon cables coming from the RS232c ports. They should both connect to a 3" x 3" circuit board. On the back of the board they will connect to the inner green circuit board of the 2 large boards in the center of the controller. Confirm that they are securely in their perspective socket. If these are fine and you still cannot communicate with the UDNC, call the Yuasa Service Dept at 800-421-9763 for assistance with the UDNC.

B) My CNC Machine does not communicate with UDNC.

1) Confirm what type of cable you are using, Standard or Null, and that communication cables are plugged into proper ports. Your CNC may require lines 2 and 3 of the cable to be crossed {Null Modem.} Use a Null Modem adapter to the end of the cable. Some machines may have a Null modem port, in which case a Null Modem cable will be converted to a Standard cable.

2) Confirm whether you are transmitting at 4800 or 9600 baud rate. The UDNC has a default setting of 9600. If you need 4800 baud rate, see Dip Switch Configure.

3) Confirm that your CNC has the ability to send and receive ASCII (American Standard Code of Information Interchange) such as Fanuc's "Macro 'B'" or Okuma's "User Task II." It is imperative that your CNC has this capability or you cannot communicate with the UDNC.

4) Confirm that the CNC protocols are properly configured and that you have used Carriage Returns <CR>. {Refer to your CNC manual for more information.}

5) **I've tried everything and still no communications.** Unplug the power cord and remove the cover from the UDNC. Check the ribbon cables coming from the RS232c ports. They should both connect to a 3" x 3" circuit board. On the back of the board they will connect to the inner green circuit board of the 2 large boards in the center of the controller. Confirm that they are securely in their perspective socket. If these are fine and you still cannot communicate with the UDNC, call the Yuasa Service Dept at 800-421-9763 and ask for technical assistance with the UDNC.

C) My HCT does not communicate with the UDNC.

1) Confirm that the TIC {Terminal Interface Cable) is securely plugged into both the HCT and the UDNC. Take care that you did not bend any of the 4 pins on the HCT.

2) Confirm that the batteries are not low and/or dying.

3) Confirm that the preferences are correctly set on the HCT. Turn on the HCT, confirm "NO HOST" and press the ENTER bar. Scroll down the menu using the DOWN ARROW key until you highlight "Preferences," push "F" key. { F = OK.} The next screen will have your choices in the black Boxes on the bottom of the screen. Each box corresponds with the button below it; A - F. Select (default setting) 9600/ 8 bit/ Even/ A0. {Using the NXT key changes the screen and options in the Boxes.} Push "ENTER" to complete the selection sequence and return to the main menu. Highlight "OFF"- "ENTER"- "ON" to see if you have established communications.

4) **I've tried everything and still no communications.** Unplug the power cord and remove the cover from the UDNC. Check the ribbon cables coming from the RS232c ports. They should both connect to a 3" x 3" circuit board. On the back of the board they will connect to the inner green circuit board of the 2 large boards in the center of the controller. Confirm that they are securely in their perspective socket. If these are fine and you still cannot communicate with the UDNC, call the Yuasa Service Dept at 800-421-9763 and ask for technical assistance with the UDNC.

Error Code Definitions:

AXIS LIMIT: Motor is restricted or has failed.
Causes:

1) Am phenol cable is not properly connected to UDNC. Make sure motor cable is properly snugged.

2) Stress to internal wires, {ie: broken motor cable, coolant seepage,} have caused lines "R," "S," or "V" to break or short. Perform a continuity test on these lines.

3) Possible bad amplifier. Unplug the UDNC. Remove the Am phenol cable from the UDNC. Take (3) 4" solid core copper wires and strip 1/4" of insulation off each end. Twist (1) end of all (3) wires together and electrical tape or cap. Insert (1) wire into the "R" socket of the UDNC Am phenol plug. Place the (2) remaining wires in "S" and "V" respectively. Apply power to UDNC and turn unit on; watch the display read out. ***If the AXIS LIMIT remains, your amplifier is bad.*** If the AXIS LIMIT is gone, you may have a damaged motor. ***Call the Yuasa Service Dept at 800-421-9763.***

EAROM: EAROM has failed. Call Yuasa Service Dept for repair.

POS FOLLOW: Position Following Error. Unit is binding.
Causes:

1) Parameter i06 is set too high. Do not exceed the recommended optimum stated in the SUDX Manual. If workpiece is off center or overtly heavy, decrease i06 or lower feed rate.

2) Unfiltered air lines have allowed condensation into the brake cylinder. Disconnect air lines and power to the indexer. Remove the back plate, thrust & brake plates from the unit and examine for rust and or corrosion in the chamber. If any corrosion is found, remove using a wire wheel or light grit sand paper until all traces are removed. {Be careful not to scare or damage chamber.} Reassemble unit and attach a Filter/Regulator/Lubricator to the airline before reconnecting to SUDX.

3) On smaller units (5C, 130, 132-5C) the micro collar maybe off set and restricting rotation. Loosen vernier ring and test unit.

4) Tool crash has created unknown damage to unit. Contact Yuasa Service Dept.

5) Check the oil level thru the sight glass on your unit. If oil is low, add until sight glass reads ½ full. On 5c/130/132-5c, remove the upper grease fitting and add some oil to the unit and replace fitting.

BRAKE CLAMP: Brake solenoid did not release and or brake is binding.

Causes:

1) Unfiltered air lines have allowed condensation into the brake cylinder. Disconnect air lines and power to the indexer. Remove the back plate, thrust & brake plates from the unit and examine for rust and or corrosion in the chamber. If any corrosion is found, remove using a wire wheel or light grit sand paper until all traces are removed. {Be careful not to scare or damage chamber.} Reassemble unit and attach a Filter/Regulator/Lubricator to the airline before reconnecting to SUDX.

2) Exhaust line has a kink in it and will not allow the air to pass. Remove the triangular plate that holds the air fittings from the SUDX. Examine the lines for constriction. If constricted, use electrical tape to reinforce tube or trim if tube length is excessive.

3) Air solenoid is malfunctioning. Push the Remote/Manual switch on the UDNC off and on and listen to the solenoid fore triggering. If no trigger is heard, solenoid is bad. Contact Yuasa Service Dept.

CMD FORMAT: Unrecognized command statement.
Cause:

1) Does the program downloaded have a M30 {end of program} statement at the end.

2) Command line is longer than 80 characters.

3) Is the cursor flashing behind the M30 when you access the file. If so, push “ENTER” to place an end of line statement.

4) Mistyped command was input. Examine input line for and inaccuracies.

Dip Switch Configuration:

With the cover removed from the UDNC and the control panel facing toward you, you will see 2 windows on the Blue (Amplifier) Board. Behind these 2 windows reside 2 banks of dip switches.

The bank on the left (BANK 1) is for adjusting bit transfer rate, {7 or 8 bit.} There are 8 dip switches on this bank. Switches 1 & 8 are set in the same direction where as 2 - 7 are opposed. This is our default setting of 8 bit. To change this to 7 bit, set switch 5 to match switches 1 & 8.

The bank closest to the center of the board (BANK 2) is used to set the baud rate and specify card address.

Card Address:
Switches 1, 2, & 3 control the address selection. Our default setting is for A0 in which case all 3 switches are facing the same direction, top of each switch pushed in. This position is refered to as 0, with the bottom pushed in the position is referred to as 1.

SW1	SW2	SW3	CARD ADDRESS
0	0	0	A0
1	0	0	A1

0	1	0	A2
1	1	0	A3
0	0	1	A4
1	0	1	A5
0	1	1	A6
1	1	1	A7

Switches 4, 5, & 6 control the baud rate of the transmission line. Our default setting is for 9600 baud, other bauds may be changed by setting the switches below.

SW4	SW5	SW6	BAUD RATE
0	1	0	9600
1	1	0	4800
0	0	1	2400
1	0	1	1200
0	1	1	600
1	1	1	300

0 = bottom of switch extended. ***1 = top of switch extended.***

Section 5: RS232 Interface

Direct Operation of SUDX, via RS232 port for CNC Controls.

Advantages for direct operation, opposed to M-Code use

1. You must only write ***a single CNC machine program***, including the rotary movements; ***no need to have a separate program for the SUDX controller.***

2. In viewing the CNC machines CRT; you always know the position of the rotary table, relative to any other program line.

3. When programming the SUDX in absolute (G90), ***the SUDX will always be in the correct location, even when re-starting the machine program at another line,*** (see "explanation of moves commands", below, (G91) for more detailed information.

4. If you utilize a CAM/CAD program for developing CNC machine files, you may use this format for incorporating the SUDX moves.

5. Macro programming permits the use of ***"variables"*** , which are very useful for creating "calls" for various types of functions, such as half indexes, math functions, etc.

6. ***Sub-Programs*** may also be used, for ease of programming repeat moves

The following information is only to be used as a guide. CNC machine controls may need different machine parameters and pin configurations for the RS232 cable, for each type and make of control. Please confirm all information with your particular machine parameters guide.

The requirements for the FANUC control is as follows:

1. You **MUST HAVE MACRO-B.**
2. An RS232 cable, (pin configuration, and gender may vary)
3. A spare, accessible M-Code.

The following is the procedure for interfacing an SUDX Programmable Rotary table, for direct operation, through the Fanuc controls, via an RS232 connection.

Getting the SUDX Controller ready.

1. Remove the controller cover, and change dip switch #1, bit 5 to open. This will change the data format to 7 bit, even parity, one stop. Fanuc controls can only communicate in this format. We normally ship the SUDX units with the 8 bit configuration, however, if you are just purchasing your unit, we would be happy to change this switch, prior to shipment.

2. The baud rate setting of the SUDX controller is 9600 bps. This can be changed by dip switch alterations inside the controller from 300 bps, to 19,200 bps. If you communicate with the FANUC control with other devises, and are locked into a baud rate other than 9600 bps, please refer to **page #40** for specific dip switch changes for the baud rate you desire. If 9600 is OK, then no

change is needed at the SUDX controller side.

Getting the Fanuc Control ready.

The following is a series of steps, in order to prepare the FANUC control, to communicate with the SUDX controller. Be advised that some of the parameter settings may change, depending on the particular control. The settings below, are for the FANUC OM. Confirm parameter settings for your individual control.

1. Depress the "Diagnos Param" button.
2. Push the " Page Down" key once.
3. Turn the function switch to "MDI" mode.
4. Change "PWE" to equal "1" , then "INPUT".
   At this point an ALARM will show.
5. Depress the "Diagnos Param" button.
6. "Page Up" once to the "I/O" page, use cursor to make sure I/O = 0.
7. "Page Down" to change parameters.

Change the following parameters.

0002   Bit #2 = 0 (sets data format)
          Bit #0 = 0   "   "   "
0057   Bit #4 = 1 (gives carriage return, line feed)
0552   =11   (sets to 9600 bps)
8. "Page up" to "PWE" =1 (setting #2)
9. "INPUT" "0"   "PWE" =0
10. Depress "PROG" button, and "RESET"

FANUC 20 & 21M CONTROL

PROTOCOL AND SET UP PARAMETERS FOR "DPRNT"

Note: Please use the following parameter settings only as a guide. Check you actual machine parameter listing for confirmation.

Param#	Setting	Description
0000	00001110	ASCII Code
0020	Chanel 0	I/O Channel
0100	10000000	LF/CR
0101	10000000	Stop Bit
0102	4	
0103	11	9600 bps
6001	00010000	CR/LF after "DPRNT"

Since all FANUC controls only work with 7 DATA BITS, it will be nessessary to OPEN DIP SWITCH # 1 PIN #5 in the SUDX controller. This will set the SUDX controller to 7 DATA BITS, EVEN PARITY, 1 STOP BIT.

You are now ready to write a TEST program on the FANUC control.

```
PROGRAM
program #

POPEN;
DPRNT [ A0EF1000G91B90M30 ];
PCLOS;
M21 ( example of CNC machine M-Code)
M30;
%
```

Once you execute this program the rotary table should move incrementally 90 degrees. Essentially what is happening here is that the RS232 port on the FANUC control is being opened for communication "POPEN" , then the control is being told to send the characters inside the square brackets [ ] out through the port "DPRNT" , in ASCII format. Once the SUDX controller receives this data, the CNC machine M-Code will automatically execute a move.

RS232 Interface to FANUC Controls
Addendum

The following is a method of placing the Macro "DPRNT" command line, in a variable. This allows the programmer to use a simple program line to execute any SUDX rotary movement.

<u>CNC Machine Macro Program</u> (sample)

```
%
:9010
POPEN;
DPRNT[A0EF1000G90B#2[33]M30];
PCLOS;
M21; (example of Machine M-Code)
M99;
%
```

Explanation

Programs beginning with a "9" are considered Macro program numbers. Some of these programs are called with G-Codes and others are called with M-Codes, (please refer to your control manual). In the DPRNT line note the "B#2", this requests the program to look @ the B "index movement" in the main machine program line. The [33] states that the index degrees will be 3 digits (maximum), for example "180" degrees, and to look 3 digits left of the decimal. If you need to use fractional degrees like "180.5025 degrees, then you may wish to input [73] 7 total digits, with 3 digits left of the decimal. Also note the "M21" after the "PCLOS". This is a sample of the M-Code needed to execute the DPRNT command line. **This M-Code will change depending on the brand of CNC machine.** This M-Code will be the spare M-Code accessed **via the REMOTE cable.**

Additional Parameter Settings.

#220 = 15 (for G15)
#10 Bit 4 = 1 (Macro Protect)

Main CNC Program Line Example

G15 B180 (This line in the main program would provide an absolute move of 180 degrees)

Note: You may wish to write another Macro program changing only the "G90" to "G91". In this case you would use a G16 instead of a G15 , and parameter #221 = 16. The program # would be 9011.

Special Note: This is only an example, the parameter settings and codes may be different for every control. Please refer to your program manual.

FADAL control is as follows:

1. You **MUST Have Macro Capability, CNC 88HS (1992) or new controls.**
2. An RS232 cable, (pin configuration, and gender described below)

Getting the SUDX Controller ready.

1. Remove the controller cover, and change dip switch #1, bit 5 to open. This will change the data format to 7 bit, even parity, one stop. Fadal controls can only communicate in this format. We normally ship the SUDX units with the 8 bit configuration, however, if you are just purchasing your unit, we would be happy to change this switch, prior to shipment.

2. The baud rate setting of the SUDX controller is 9600 bps. This can be changed by dip switch alterations inside the controller from 300 bps, to 19,200 bps. If you communicate with the Fadal control with other devises, and are locked into a baud rate other than 9600 bps, give us a call, 1 800 421 9763, and we will let you know the specific dip switch changes for the baud rate you desire. If 9600 is OK, then no change is needed at the SUDX controller side.

Getting the Fadal Control ready.

The following is a series of steps, in order to prepare the FADAL control, to communicate with the SUDX controller. Be advised that some of the parameter settings may change, depending on the particular control. Confirm parameter settings for your individual control.

Using the baud rate parameter, set the Fadal control to communicate at 9600BPS. Using an RS232 cable, connect the Fadal control to the SUDX controller, using the DB25 serial connector on both. For the pin configuration, see below.

You are now ready to write a TEST program on the FADAL control.

```
PROGRAM
program #
#WAIT
#SPRINT " A0EF1000G91B90M30 "
M20 (example of machine M Code)
```

Once you execute this program the rotary table should move incrementally 90 degrees. Essentially what is happening here is that the RS232 port on the FADAL control is being opened for communication , then the control is being told to send the characters inside the double quotes " " out through the port "#SPRINT" , in ASCII format. Once the SUDX controller receives this data, the CNC machine M-Code will automatically execute a move. Then the RS232 port on the FADAL is closed and the control will automatically continue to the next line.

Explanation for FADAL Addendum

The attached sheets explain a new method of commanding the SUDX to move to a desired position via an RS232 cable between the SUDX controller and the FADAL control. Using the FADAL MACRO function (#SPRINT) we are now able to **eliminate the need for the (M20, from the FADAL) for position confirmation.** The REMOTE cable is not even connected between the FADAL and the SUDX controller. **Commanded position and the verification of position is all accomplished via the RS232 cable.**

The theory for this is as follows. (Please refer to N5 of the MACRO program) First the ASCII data is output from the FADAL control to the SUDX controller commanding the rotary table to move to position. Once the move is complete (refer to N7 of the MACRO program) a "Y" (send current position) is sent in ASCII to the SUDX controller. Position (i.e. 90.000 , for 90 degrees) is sent from the SUDX controller to the FADAL control. Once the FADAL sees this data, (to a variable, refer to N13 and 14 of the MACRO program) the VMC proceeds to the next block in the main machine program. If the position data sent is not correct (i.e. 85.000, for 85 degrees, when the actual position should be 90 degrees) then the FADAL will go to ERROR (see N15 of the MACRO program).

This entire sequence is all placed into a MACRO program, which is "CALLED" in the main program. (see N3 of the TEST program). The actual data that the customer would need to input into the FADAL main program is quite simple, (see N4 & N11 of the TEST program). These simple commands are all that is needed to index the rotary table. Another MACRO program could be written for incremental moves, if desired. MACRO program # 09998 is set for absolute (G90) movements, as this is strongly recommended should the user need to re-start his machine program in the middle of the program. ABSOLUTE enables this feature as the rotary table will always move to actual desired position, regardless of it's previous move.

This MACRO and TEST program is now included on all program disks sent out with each SUDX unit. The FADAL user need only load these programs into the FADAL control (see page #1 of the ADDENDUM) and they are ready to go.

Prior to shipment of the SUDX we can (if requested) set the communication PROTOCOL correctly (7 data bit, even parity, one stop) and the customer requested baud rate (we normally ship at 9600 BPS).

MORI SEIKI TV-30 & 40

Parameters

```
# 7201   100 (G CODE Numbers)
#  7202   1    (G65P)
#7203     (any Sub Program #)
```

Sub Program List

```
00001
#100=#4003
POPEN
DPRNT[A1A0EF1000G#[20] B#2[33]M30A1]
PCLOS
M21
M99
```

Note#1: The above is only meant as a guide or reference. Please check your actual machine parameters and Macro program for verification.

Note#2 : The "A1" at the beginning and end of the program line may not be needed on your UDNC 100 controller.

Note#3: For set up and communication parameters, please see the general Mori TV-30 and 40 set up page.

OKUMA control is as follows:

1. You **MUST HAVE "USER TASK II"**
2. An RS232 cable, (pin configuration, and gender described below)
3. A spare, accessible M-Code, if you require a step finish signal.

Getting the SUDX Controller ready.

1. The baud rate setting of the SUDX controller is 9600 bps. This can be changed by dip switch alterations inside the controller from 300 bps, to 19,200 bps. If you communicate with the Okuma control with other devises, and are locked into a baud rate other than 9600 bps, give us a call, 1 800 421 9763, and we will let you know the specific dip switch changes for the baud rate you desire. If 9600 is OK, then no change is needed at the SUDX controller side.

Getting the Okuma Control ready.

The following is a series of steps, in order to prepare the Okuma control, to communicate with the SUDX controller. Be advised that some of the parameter settings may change, depending on the particular control. The settings below, are for the Okuma OSP 7000, 5020 and 5000. Confirm parameter settings for your individual control.

Change the following parameters.

Optional Parameter Word.
#39 = 9600

Optional Parameter Bit.
#1   01010011
#12 00111011

You are now ready to write a TEST program on the Okuma control.

```
PUT `A0EF1000G91`
PUT `B180M30`
PUT $0A0D
WRITE 0
M181
M2
```

Once you execute this program the rotary table should move incrementally 180 degrees (with the M181 output) Essentially what is happening here is that the RS232 port on the Okuma control is being opened for communication "PUT" , then the control is being told to send the characters inside the quotes, out through the port, in ASCII format. Once the SUDX controller receives this data, the M181 command will automatically execute a move. Then the RS232 port on the Okuma is closed, the control will automatically continue to the next line.

Yasnac I 80 & MX-3 Control, via. RS232 Port.

I 80

1. Go to the maintenance screen, depress soft key "ALLPARA" , then softkey #5, select RS232.
2. Next a set up screen will appear, input data as follows:
3. Make sure that the pin configuration for the RS323 cable between the UDNC controller and the I 80 control, are correct. Please refer to our config. diagram for the UDNC pin-out, and the machine diagram.. Most likely you will need a "NULL MODEM" adapter, switching pins #2 and #3. In addition, note that the DB-25 connector on the UDNC-100 is female, and the DB-25 on the CNC machine is usually female. You will need male ends for both sides of your RS232 cable.
4. To test a program, go to MDI mode on the Yasnac and input the following :

#3100=(A0EF1000G90B90M30)
M21 (example of machine M-Code)

This will initiate a movement of 90 degrees.
Note: Make sure that the "Manual/Remote" switch is on Remote.
Make sure that you input "A (the number zero) E and not the letter "O"
If it does not work at this point, re-check your RS232 connection, this is most likely where the problem is.

MX-3

The set up for this control is very similar to the I-80. The command (#3100) and program data is the same also. The following is a step procedure for communication protocol.

1. Depress the " PRM" switch.
2. Push the soft key " RS232"

This should bring you to the setting page: 3. Depress " PROG" then go to either "MDI" mode or write a program to test. Again, check pin configuration of the RS232 cable, and make sure you are connected to the correct port on the CNC machine.

The requirements for the Yasnac control is as follows:

1. You **MUST HAVE MACRO-B, this should be a standard control feature, but check to be sure.**
2. An RS232 cable, (pin configuration, and gender described below)
3. A spare, accessible M-Code, if you require a Step Finish signal.

Getting the Yasnac Control ready.

The following is a series of steps, in order to prepare the Yasnac control, to communicate with the SUDX controller. Be advised that some of the parameter settings may change, depending on the particular control. The settings below, are for the Yasnac I80 & MX-3. Confirm parameter settings for your individual control.

1. Go to maintenance screen, depress the soft key"ALLPARA", then softkey #5, select RS232C
2. Next, a set up screen will appear, input the data as follows:

You are now ready to write a TEST program on the Yasnac control.

```
PROGRAM
program #
#3100= (A0EF1000G91B90M30)
M21 (example of Machine M-Code)
M30;
%
```

	IN	**OUT**
Units	RS232C	RS232C
BPS	9600	9600
Stop Bit	1	1
Bits	8	8
Parity	Off	Off
Xon/Off	On	On
Rts/Ctl	Off	Off
Parity ISO	On	On
RS232	Output	
Channel No.	1 or 2	
ISO/EIA	ISO	
Control Code	ON	
Stop Bit	1	
Baud	9600	
Parity	OFF	

Section 6: Yuasa Terminal

1.0 YUASA TERMINAL VERSION 1.X

YUASA TERMINAL V1.X is an all-purpose programming tool for the UDNC-100 Controller. It allows for on-line terminal operations, program editing, uploading and downloading the UDNC-100. It is written in the “C” programming language.

SYSTEM REQUIREMENTS

YUASA TERMINAL Version 1.X runs on an IBM PC, PC/AT,80286, or compatibles with at least 640K and one disk drive. In addition, one standard serial port is required.

COPYING THE SUPPLIED DISKETTE

Making a backup copy of your diskette is essential for insuring that the programs are not destroyed if the original diskette is ever erased or damaged.

If you have two floppy disk drives, you may backup the original YUASA diskette as follows:

1. Insert your DOS diskette into Drive A and a blank diskette into Drive B and reset the computer.
2. At the A> prompt type FORMAT B: and press <ENTER>.
3. After the disk in Drive B is formatted, type: COPY A:*.* B: and press <ENTER>.
4. Remove the DOS diskette. Place the YUASA diskette in Drive A and press <ENTER> to the prompt.
5. After the original disk is copied, store it in a safe place and use the backup copy as your working master.

If you have a hard disk drive, you may back up the original YUASA diskette as follows:

1. At the C> prompt type: MD YUASA and press <ENTER>.
2. Type: CD YUASA and press <ENTER>.
3. Insert the supplied YUASA disk in the correct drive & Type: INSTALL then press <ENTER>.

Note: ***The supplied diskette operates under DOS Version 5.0 or greater***

INSTALLING YUASA TERMINAL ON YOUR C: drive

To install YUASA TERMINAL to your hard disk, insert the backup copy of the provided disk, go to the appropriate floppy drive, then type INSTALL and press <ENTER>.

RUNNING YUASA TERMINAL

To run YUASA TERMINAL from your hard disk, type CD YUASA and press <ENTER>.
Then type: YUASA and press <ENTER>.

After the two windows appear on the screen and a listing of the function keys appears in the right-hand window, you will be prompted with a question such as the following:

System Configured as:
Card Number: 0
Port: COM1
Baudrate: 9600
ASCII Type: 8 bit, no parity
Communication timeout (milliseconds): 100
Do you want to change it? (y/n)

Note: ***All UDNC-100 controllers are factory set to the above settings unless otherwise requested before shipping.***

If your Controller Card is configured for this setting, simply press <N> or <ENTER>. If you press <Y>, you will be allowed to configure YUASA TERMINAL to the proper card address (0-7), port type (com 1 or com 2), baudrate (19200, 9600, 4800, 2400, 1200, 600, 300), ASCII type (8 bit no parity, 7 bit even parity), and timeout setting which is proper for your controller setup. Also, refer to the Alt-F1 (Configure) key description in Section 2 of this manual for detailed instructions.

YUASA TERMINAL proceeds to poll the available cards (A0 through A7) and determines which cards are connected to your system. You will then be placed into Terminal Mode. If no cards are connected, or the communication port in your computer is not properly configured, you will get a message: 'No Cards found'.

Pressing the "Escape Key" allows you to exit YUASA TERMINAL and return to DOS. If you wish to try to specify the interface configuration again, use the ALT <F1> Key. YUASA TERMINAL will enter its normal terminal mode regardless of whether any Controller cards are connected to the computer.

2.0 YUASA TERMINAL MODE

After YUASA TERMINAL successfully determines that one or more cards are connected, you may proceed to communicate with the cards. The left-hand window will act as a direct terminal to the card. Anything you type will be sent to the card (once you press <ENTER>, and anything sent from the card will be promptly displayed on the screen.

EXAMPLE: Typing the letter "J" and pressing <ENTER> will cause the motor attached to the card to jog in the positive direction. Type another "J" or "Q" to stop the motion. Typing the lowercase letter "p" or "Y" will cause the Controller Card to send its position data which will be displayed on the next line of the screen.

Note: *Upon entering terminal mode for the first time, YUASA TERMINAL automatically addresses card A0. Thus the position data displayed by the "p" or "Y" command will be the position of card A0. To change the card being addressed, type "A" and the card's number (0-9) or letter (a-f) and press <ENTER>.*

In addition to the on-line commands described in the YUASA SUDX Manual, the following function keys

<F1> - EDIT I-VARIABLE

This function key retrieves up to 32 "i"-parameters of the card currently addressed. The parameters are divided into two screens. To display the next screen, press the <Pg Dn> key located on the numeric keypad. To display the previous screen, press the <Pg Up> key located on the numeric keypad. Once the parameter you wish to modify is displayed on the current screen, you may access it by positioning the cursor bar over the parameter. To move the cursor bar, use the Up Arrow and Down Arrow keys, and to modify the parameter, press <ENTER>.

After positioning the bar over the desired parameter and pressing <ENTER>, the following message will appear in the right-hand window:

```
CHANGE - ixx
Maximum:  xxxxxx
Minimum:  XXXXXX
Default:  XXXXXX
Units:    XXXXXX
Now:      XXXXXX
ENTER NEW VALUE:
=>
```

The values listed refer to the appropriate values of the parameter you selected. You may enter a new value, or press <ESC> to cancel the change. If you enter a value which exceeds the critical values of the parameter, a warning message will flash and you will be prompted to try again.

When you finish reviewing the i-parameters, press <Esc> to return to Terminal Mode, or "s" to save the parameters.

<F2> - DISPLAY I/O STAT

This key displays the current I/O status.

<F3> - DISPLAY CARD STAT

This function key displays the current card status.

<F4> - LIST PROG BUF

This key will display the buffer of the card currently addressed. You are prompted for the starting line number, after which each carriage return keystroke will display a page full of consecutive program lines. The axis data will be displayed in user units set by the “i24” variable. To return back to Terminal Mode, press <ESC>.

<F5> - MDI MODE ON/OFF

This function key allows you to enter the Manual Data Input Mode. In this mode motion commands will be executed as they are entered. “MDI On” will be displayed in the lower right window when MDI is active.

<F6> - OPEN LOOP MODE

This mode allows the user to test the motor and encoder in an open loop mode. The arrow keys allow the user to adjust the motor output in 0.1 and 0.01 volt increments. By changing this output, the user can drive the motor and determine if the encoder position feedback direction is correct before closing the servo loop. The <ESC> key will exit this mode and gradually return the output to zero.

Warning: ***Do not exceed 4.00 volts, as it may damage the Rotary Table and/or the Controller's Amp Board. Resulting in a costly non-warranty repair.***

<F7> - POS/FOLLOW/TIME

This function key displays the position, following error and time for the card connected to your system.

<F9> - DOS COMMANDS

This key exits you to DOS allowing you to perform DOS commands. Typing <exit> will return you to the YUASA program.

<F10> - FILE EDITOR

This key allows you to create, edit, download, and upload programs. See section 3.0 below for detailed instructions.

The following functions are activated by holding down the <Alt> key and pressing the appropriate function key. The alternate function key selections can be viewed on the right-hand screen by holding down the <Alt> key and pressing <Q>.

Alt <F1> - COMM SET UP

This function key allows you to re-configure and re-poll your system. You may change the communication parameters (port, baudrate, etc.) and re-poll the available cards.

Note: ***This function is automatically invoked when YUASA TERMINAL is first run. The current interface configuration is shown, and you are asked if you wish to change it. If you answer ‘Y’ or ‘y’, you will be asked which of the following interfaces you wish to configure:***

Card Number:(0-7)

This is used to address controllers that have been "Daisy Chained Together" via a serial cable.

Note 1: Factory ***default for a single controller is 0.***

Port: <1> COM1 <2> COM2

where selections 1 and 2 refer to the standard RS-232 ports on your PC.

When you select the RS-232 interface, you will select the baud rate of 9600 from a menu of choices. The rate you choose must match the setting on the DIP switch on the Controllers card(s), and, in general, the longer the cable is, the lower the baud rate should be.

Note 3: ***Factory default is 9600 baud, see Trouble Shooting if you need to change.***

ASCII Type: <0> 8 bit no parity   <1> 7 bit even parity

Note 4: ***Factory default is 8 bit no parity, call for information if you need to change.***

Enter Communication Timeout (msec): __ (MINIMUM 35) (Recommended: 60)

This puts a limit on the time the program waits for a handshake bit for reading or writing.

Alt <F2>- DISK ON/OFF

This key will toggle the DISK ON option. With DISK ON, most data received from the card will be echoed to the default disk drive under the filename "TERMDATA.MCC." This file may be examined by loading it through the file editor. When YUASA TERMINAL is first run, if the file TERMDATA.MCC already exists, it will be erased. However, all subsequent uses of DISK ON will append all incoming data to the file. When in the POS/FOLLOW/TIME mode, the data displayed will also be sent to disk. To view this file, you must turn off DISK ON. This will close the file so you can view it with the file editor.

Alt <F3>- PRINT ON/OFF

This key will toggle the PRINTER ON option. With PRINTER ON, all data received from the card will be echoed to any standard line printer as well as to the screen. Make sure that the line printer has paper and is On-Line.

Alt <F4>- DISPLAY OFF

This key stops data from being displayed on the terminal screen. This allows you to send data to the printer or disk at a faster rate.

Alt <F5>- SAV PRGR&I-VAR

This key will save the current Program Buffer and I-Variables currently in the RAM to the EPROM.

Alt <F6>- PRINT SCREEN

Will print the current screen to your printer.

Alt <F10>- HELP FILE

<1> - Load Help File : Allows the user to view a Help File(*.HLP).
<2> - Reformat Help File: Reformats the Help File for easier viewing.
<ESC> - EXIT: EXIT the Help Menu.

3.0 YUASA EDITOR

The editor is a word processor for Controller programs of up to 65000 characters. In this section of the program, lines typed in the left window are not sent to the Controller, but instead entered into the editor buffer. The line number corresponding to the line of the editor buffer to which the cursor is currently pointing is displayed in the right window. Movement of the cursor is possible with the four Arrow Keys. The following editing keys also apply:

PgUp - Displays the previous page (23 lines) of the current program.
PgDn - Displays the next page (23 lines) of the current program.
Home - Displays the first page (23 lines) of the current program.
End - Displays the last page (23 lines) of the current program.
Alt-I - Inserts a blank line at the current cursor position. All subsequent lines are moved down.
Alt-D - Deletes the line at the current cursor position. All subsequent lines are moved up.

Alt-C - Clears the current buffer and filename.

Ins - Toggles between the “INSERT ON” and “INSERT OFF” modes of the editor. Insert Off allows you to type over previously entered text, while Insert Off will move any previously entered text to the right of the cursor.

Note: ***Each line is limited to fifty-two characters and the editor is limited to 65000 characters. The editor will signal you when the last line is reached.***

In addition, the following function keys may be used:

<F1> - UPLOAD I-VARIABLES

This key uploads the current I-Variables into the displayed buffer file shown at the bottom of the screen.

<F2> - FILE UTILITIES

This key allows you to load, save, insert, or delete a program on the current disk. In addition, you may print the file currently in the editor or a file saved on a diskette. You will be presented with the following selection menu in the left window:

<1>	-	Load File	<4>	-	Delete File
<2>	-	Save File	<5>	-	Print Current File
<3>	-	Insert File	<6>	-	Print Disk File
<9>	-	DOS Commands	<ESC>	-	EXIT

Enter the number of your selection at the prompt. The selections perform the following tasks:

<1> - Load File : Allows you to bring a file in from disk to the editor buffer. Any program currently in the editor buffer will be erased.

<2> - Save File : Allows you to save the program currently in the editor buffer to disk.

Note: ***Filenames are limited to eight standard characters and unless you specify your own extension, the extension “.MCC” is automatically added to any filename. Disk and subdirectory specifiers can be entered in front of a filename, up to a total of 40 characters. The default filename for a newly created program in the editor buffer is NONAME.MCC.***

<3> - Insert File : Allows you to bring a file in from disk and insert it into the program currently in the editor buffer at the present cursor location.

<4> - Delete File : Allows you to delete a disk file.

<5> - Print Current File : Allows you to print the file currently in the editor buffer to the standard printer for the system.

<6> - Print Disk File : Allows you to print a file on disk to the standard file for the system.

<9> - DOS Commands : Allows you to perform DOS commands

<F3> - DWNLD&SAVE FILE

This key downloads the current buffer file to the controller and then saves the data to its internal EAROM.

<F4> - UPLOAD PROG BUF

This function key uploads the program buffer from any card and appends it to the program currently in the editor’s buffer.

<F5> - DOWNLOAD FILE

This key downloads the current buffer file to the UDNC controller. Before downloading you will be asked if you want the program buffer cleared before download. This is necessary if your file has motion program data and no “z” or “E”, clear the buffer, command. After the download you will be returned to Terminal Mode.

Note: ***The editor’s buffer is not erased; you may access it again by pressing <F10>. If there are any errors, hitting any key will return you to the file edit buffer.***

“i” Parameters List for the SUDX Series Indexers

SUDX-130		SUDX-5C(A)		SUDX-132/140/170		SUDX-220		SUDX-280/320/400	
60:1 Gear Ratio		*72:1 Gear Ratio*		*72:1 Gear Ratio*		*90:1 Gear Ratio*		*180:1 Gear Ratio*	
i00	-250	i00	-250	i00	-250	i00	-250	i00	-250
i01	2	i01	2	i01	2	i01	2	i01	2
i02	0	i02	0	i02	0	i02	0	i02	0
i03	3000	i03	3000	i03	3000	i03	3000	i03	3000
i04	250	i04	250	i04	250	i04	250	i04	250
i05	1600	i05	1600	i05	1600	i05	1600	i05	1600
i06	100000	i06	100000	i06	100000	i06	100000	i06	80000
i07	30	i07	35	i07	40	i07	45	i07	50
i08	800	i08	800	i08	800	i08	1000	i08	1200
i09	1500	i09	1500	i09	1500	i09	1500	i09	1500
i10	-1	i10	-1	i10	-1	i10	-1	i10	-1
i11	-1	i11	-1	i11	-1	i11	-1	i11	-1
i12	0	i12	0	i12	0	i12	0	i12	0
i13	90	i13	90	i13	90	i13	90	i13	90
i14	20	i14	20	i14	20	i14	20	i14	20
i15	0	i15	0	i15	0	i15	0	i15	0
i16	0	i16	0	i16	0	i16	0	i16	0
i17	20	i17	20	i17	20	i17	20	i17	20
i18	12	i18	12	i18	12	i18	12	i18	12
i19	1024	i19	1024	i19	1024	i19	1024	i19	1024
i20	-240000	i20	-288000	i20	-288000	i20	-360000	i20	-720000
i21	0	i21	0	i21	0	i21	0	i21	0
i22	0	i22	0	i22	0	i22	0	i22	0
i23	0:0	i23	0:0	i23	0:0	i23	0:0	i23	0:0
i24	2000:3	i24	800:1	i24	800:1	i24	1000:1	i24	2000:1
i25	1	i25	1	i25	1	i25	1	i25	1
i26	0	i26	0	i26	0	i26	0	i26	0
i27	3	i27	3	i27	3	i27	3	i27	3
i28	Factory Set	i28	Factory Set	i28	Factory Set	i28	Factory Set	i28	Factory Set
i29	32	i29	0	i29	32	i29	32	i29	32
i30	32	i30	32	i30	32	i30	32	i30	32
i31	32	i31	32	i31	32	i31	32	i31	32
i32	2:0	i32	2:0	i32	2:0	i32	2:0	i32	2:0

Note: On i29, i31, & i32, you may need to change these values depending upon your Finish signal requirements.