

Motion Coordinator SERIES 2:

All SERIES 2 Motion Coordinators feature Multi-Tasking TRIO BASIC. The controller can run up to 15 named BASIC programs (5 on an MC204) and a command line process simultaneously using pre-emptive multi-tasking. Each BASIC program has named local variables and can in addition use a shared variable array. The programs can use groups of axes freely in any combination.

The multi-tasking can make writing programs for complex machines far easier as a program can be written to perform each machine process independently yet they can share data and motion axes as required.

Additional BASIC commands are available on the SERIES 2 to control the multi-tasking and the real time clock.

PROGRAM CONTROLS:

COMPILE	- Compile a program
COPY	- Copies a program on the controller
DEL	- Deletes a program
DIR	- Directory of programs
EDIT	- Edit a program with VT100 terminal
EPROM	- Store programs to flash eeprom
NEW	- Clear a program
HALT	- Halts all processes
LIST	- List a program
PROCESS	- Lists running programs and priorities
RENAME	- Renames a program
RUNTYPE	- Sets run time priority and configuration
SELECT	- Selects a program for edit

REAL TIME CLOCK CONTROLS: (MC2 ONLY)

DATE	- Returns date
DATE\$	- Prints date
DAY	- Returns day of week
DAY\$	- Prints day of week
TIME	- Returns time
TIMES	- Prints time

ADDITIONAL CONTROLLER FUNCTIONS:

LOADSYSTEM-	Serially update system software
LOCK	- Prevent programs being edited or viewed
STORE	- Store updated system software to flash eeprom
UNLOCK	- Security coded reversal of LOCK

Motion Perfect:

Trio supply a Windows based application development program "Motion Perfect" with each Motion Coordinator. This program provides all the features required to setup, program and archive applications. The program communicates with the Motion Coordinator via the serial port.

- Project Manager
- Axis Parameter Display Change
- Automatically Load and Save Programs to PC
- Automatically Generate Initialization Program
- Multiple Terminal Windows may be opened
- Multi-Program "Windows" Style Screen Editor
- Load and Save Variables/Tables to PC
- Display/Change Inputs, Outputs and Flags
- Jog Axes
- Program Debugger
- On-line Help with Motion Perfect and Trio BASIC
- Trio Membrane Keypad Emulator
- Configure and Initialise CAN Amplifiers *
- 4-Channel Oscilloscope to Monitor Parameters & I/O
- Graphical view of Table data
- Configure Program Baudrate, Serial Port, Displays etc.

Trio BASIC Programming



Trio's Multi-Tasking BASIC is a flexible programming system used by the Motion Coordinator SERIES 2 range of programmable positioners. The syntax is similar to that of other structured BASIC systems. A PC is used to develop and test the application program which coordinates all the required motions and machine functions. Once complete, the PC may be removed and the controller set to run standalone.

FEATURES

- Fast structured BASIC for easy standalone machine programming
- Fully integrated with Trio's MotionPerfect application development software
- Comprehensive motion control functions for 1-16 axes (dependant on controller type)
- Multi-tasking of up to 14 tasks (5 on MC204)
- Axes may be servo or stepper motors
- Feedback from incremental or absolute encoder or resolver. Stepper position verification
- Free MotionPerfect support utilities run on PC compatibles
- BASIC programs stored in flash EPROM
- Versatile I/O Expansion options. 96 On MC2, up to 512 I/O via CAN interface on MC204.
- Wide range of operator interfaces
- Real maths, bit operators, and variables
- Linear, circular, and helical interpolation
- Cam profiles, software gearboxes
- High speed registration functions with hardware position capture.
- Continuous speed control
- Standalone or computer controlled operation
- Specialist user functions such as robot transformations can be incorporated into the system software.

Trio BASIC has over 200 commands designed to make programming motion functions quick and simple.

► Trio Motion Technology ◀

Unit 2 Empire Way, Gloucester. GL2 5HY
 Telephone: 01452 308332
 Fax: 01452 311884
 Web Site: www.triomotion.com
 Email: sales@triomotion.com

► Trio Motion Technology ◀

B R E A T H I N G L I F E I N T O M A C H I N E S

TRIO BASIC PROGRAMMING

MOTION CONTROL COMMANDS

Each axis can perform any move type independently while the others each simultaneously perform another. Groups of axes can be controlled together using interpolation or software gearboxes. Moves can be run on axes which are not physically present in the system and these can be added to superimpose two or more motion commands on to an axis.

- ACCEL - Set acceleration rate
- ADDAX - Adds axes demand positions for complex profiles
- BASE - Sets the group of axes for a motion command
- CAM - Move along CAM shape
- CAMBOX - Move along CAM shape using software gearbox
- CANCEL - Cancel a buffered or running move
- CONNECT - Connect using software gearbox
- CREEP - Set creeping speed
- DATUM - Set configurable datuming movement
- DECEL - Set deceleration rate
- DEFPOS - Define current position on a group of axes
- FORWARD - Set continuous forward motion
- MOVEABS - Move to absolute position. Any group of axes may be interpolated
- MOVECIRC - Move circular arc. Any group of 2 axes may be interpolated
- MHELICAL - Move helical arc
- MOVELINK - Move Link. Motion for threading, flying shears etc.
- MOVE - Move incremental position. Any group of axes may be interpolated
- MOVEMODIFY - Modify the end position of a move
- REVERSE - Set continuous reverse motion
- RAPIDSTOP - Cancels a move on all axes
- SPEED - Sets speed
- MERGE - Allows merge to be switched ON/OFF
- UNITS - Set number of encoder edges/steps in axis programming units

The motion commands allow for programming in USER UNITS. The axis variable UNITS may be set to the number of encoder edges/steps in the required mechanical units. If an axis has 2000 encoder edges/millimetre, and this axis variable is set to 2000. Move lengths would then be programmed in mm, speeds in mm/sec and acceleration rates in mm/sec/sec.

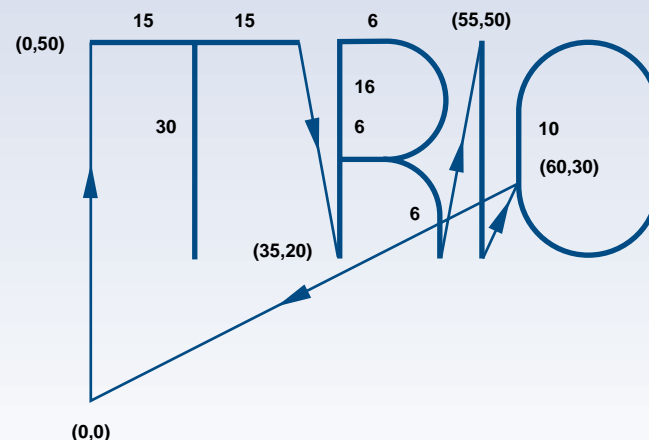
The BASIC program will buffer moves, allowing speeds and other settings to be changed "on the fly". Motion commands may be merged together seamlessly by the controller to allow continuous path profiles for glue laying, etc.

The move commands MOVELINK, CAMBOX, and CONNECT program moves which link the axis positions to the measured position of another axis. This allows for synchronization of machine movements to external webs, conveyors, spindles etc.

X-Y MOTION EXAMPLE PROGRAM:

```

\ Merges arcs and lines except when
\ forced to stop with WAIT IDLE
\
\ Output 15 is pen UP/DOWN
REPEAT
\ Plot TRIO - start with a T
MOVEABS(0,50)
WAIT IDLE
OP(15,ON)
MOVE(15,0)
MOVE(0,-30)
MOVE(0,30)
MOVE(15,0)
WAIT IDLE
OP(15,OFF)
\ Plot R:
MOVEABS(35,20)
WAIT IDLE
OP(15,ON)
MOVE(0,30)
MOVE(6,0)
MOVECIRC(8,-8,0,-8,1)
MOVE(-6,0)
WAIT IDLE
MOVE(6,0)
MOVECIRC(0,-16,0,-8,1)
MOVE(0,-6)
WAIT IDLE
OP(15,OFF)
\ Plot I:
MOVEABS(55,50)
WAIT IDLE
OP(15,ON)
MOVE(0,-30)
WAIT IDLE
OP(15,OFF)
\ Plot O:
MOVEABS(60,30)
WAIT IDLE
OP(15,ON)
MOVE(0,10)
MOVECIRC(20,0,10,0,1)
MOVE(0,-10)
MOVECIRC(-20,0,-10,0,1)
WAIT IDLE
OP(15,OFF)
\ Go back to start
MOVEABS(0,0)
GOTO START
    
```



PROGRAM LOOPS, SEQUENCES AND CONTROL:

- BASICERROR - Set branch for program error
- ELSE - IF..THEN..ELSE..construct
- ENDIF - IF ..THEN..ELSE termination for multiple line blocks
- FOR - FOR..NEXT loop construction
- GOTO - Branch to a label
- GOSUB - Branch to a subroutine
- HALT - Halts all processes *
- IDLE - Waits for move termination
- IF - Conditional branch
- NEXT - FOR..NEXT loop
- ON ... - Multiple GOTO / GOSUB via variable
- REPEAT - REPEAT UNTIL loop
- RETURN - Return from subroutine
- RUN - RUN program
- STEP - Set FOR..NEXT loop step
- STOP - Stop program
- THEN - IF..THEN..ELSE
- TO - FOR ..NEXT loop
- TRON - Set trace on
- TROFF - Set trace off
- UNTIL - REPEAT/WAIT loop
- WEND - Terminate WHILE ..WEND loop
- WHILE - Start WHILE ..WEND loop
- WAIT - Waits for condition
- WA - Waits for time

```

\
\ TRIO BASIC Structures Example Program:
\
start:
GOSUB initial
REPEAT
FOR y=0 TO 7
FOR x=0 TO 5

IF IN(6)=ON THEN
\ Move to pick up point 1:
MOVEABS(-340,-516.5)
ELSE
\ Move to pick up point 2:
MOVEABS(-220,-516.5)
ENDIF

GOSUB pick
MOVEABS(x*85,y*85)
GOSUB place

NEXT x
NEXT y
UNTIL IN(1)=OFF
    
```

COIL WINDING PROGRAMMING EXAMPLE:

```

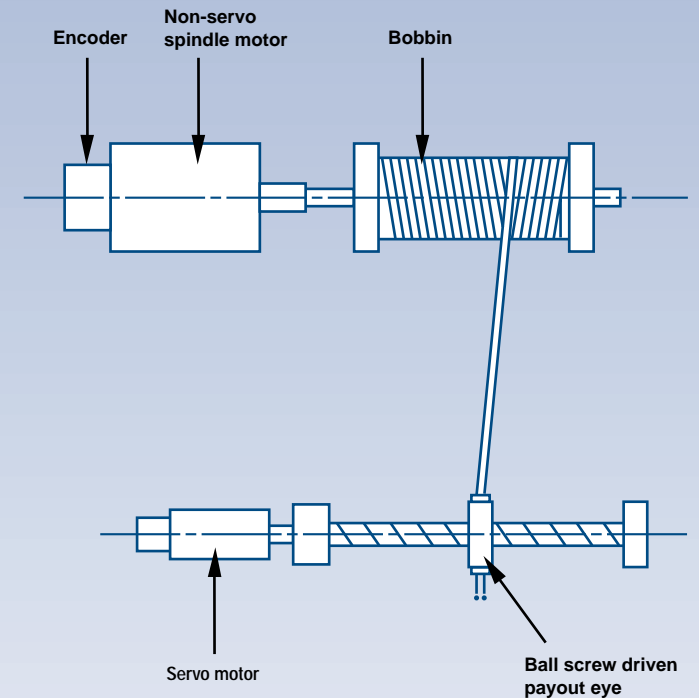
\
\ TRIO BASIC Coil Winding Example Program:
\
OP(motor,ON)' - Switch spindle motor on
FOR turn=1 TO 10

MOVELINK(50,25)
MOVELINK(-50,25)

NEXT turn
WAIT IDLE
OP(motor,OFF)
    
```

In this example the unit conversion factors UNITS are set so that the payout movements are in mm and the spindle position is measured in revolutions. The payout eye therefore moves 50mm over 25 revolutions of the spindle with the command MOVELINK(50,25). If it were desired to accelerate up over the first spindle revolution and decelerate over the final 3 the command would be MOVELINK(50,25,1,3).

MOVELINK and CAMBOX can be programmed to commence automatically relative to an absolute position on the link axis.



MOTION COORDINATOR SERIES

REGISTRATION FUNCTIONS

The MOTION COORDINATOR series features dedicated hardware for the capture of axis positions in less than 1u sec. 2 separate registration inputs are available independently on each axis. These facilities can be used to implement a wide variety of registration schemes with great accuracy.

Software Controls:

- REGIST - Determines input(s) / function / and polarity to be used
- MARK - Allows the programmer to test if mark has been captured
- MATCH - Compare Transition Pattern
- OPEN_WIN - Position at which window opens
- CLOSE_WIN - Position at which window closes
- RECORD - Record registration transitions
- REG_POS - Returns captured position/offset

Registration functions are available on Servo Encoder, Servo Resolver, Encoder Input and Stepper Encoder Daughter Boards.

PROGRAM MANIPULATION AND SYSTEM COMMANDS

- APPENDPROG - [MP] Used by Motion Perfect editor
- COPY - Copy program
- DEL - Delete program
- DIR - Directory of programs
- EDIT - Screen edit program
- EDPROG - [MP] Used by Motion Perfect editor
- EPROG - Transfer program to flash EPROM
- EX - Software Reset
- FREE - Returns free memory space
- INITIALISE - Resets parameters
- LIST - List program
- LOAD - Load program from PC
- LOADSYSTEM - Load system software
- MPE - [MP] Motion Perfect mode control
- NETSTAT - Network Status
- NEW - Deletes Program
- PMOVE - Process move buffer
- PROCNUMBER - Read current process number
- RENAME - Rename Program
- SELECT - Select active program
- SCOPE - Enable data capture facility
- SCOPE_POS - Current index in data storage
- SETSERIALNUMBER - Set OEM serial number for machine
- STORE - Store new system software in Eprom
- TRIGGER - Trigger the SCOPE function

CAM PROFILE PROGRAMMING EXAMPLE:

```

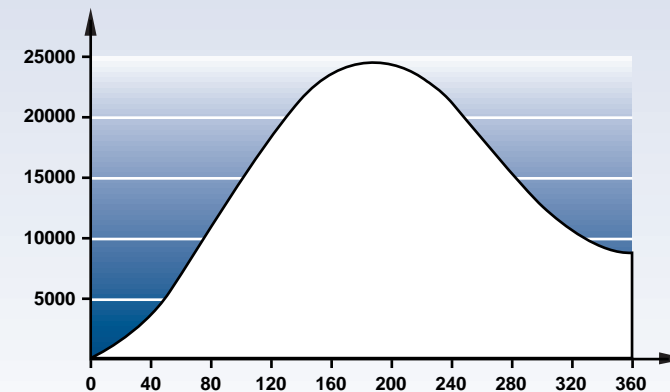
\
\ TRIO BASIC CAM Programming Example:
\
GOSUB loadtab

WHILE NOT (KEY)
    CAM (1,19,1.25,200)
WEND
STOP

\ This table holds the movement SHAPE.
\ It can be stretched in both X and Y
\ when used by the CAM command

loadtab:
TABLE (1,0,1103,3340,6500,10263,14236)
TABLE (7,18000,21160,23396,24500,24396)
TABLE (12,23160,21000,18236,15263)
TABLE (16,12500,10340,9103,9000)
RETURN
    
```

TABLE POSITION	DEGREES	VALUE
1	0	0
2	20	1103
3	40	3340
4	60	6500
5	80	10263
6	100	14236
7	120	18000
8	140	21160
9	160	23396
10	180	24500
11	200	24396
12	220	23160
13	240	21000
14	260	18236
15	280	15263
16	300	12500
17	320	10340
18	340	9103
19	360	9000



SYSTEM INFORMATION

- PROCESS - Lists running programs and priorities
- TIME - Returns Time *
- DATE - Returns Date *
- DAY - Returns Day of week *
- POWER_UP - Power up from FLASH or RAM
- NIO - Returns Number Input/Outputs
- TICKS - Returns process counters
- VERSION - Returns software version
- CONTROL - Returns controller type
- NETSTAT - Returns network status
- WDOG - Set watchdog relay state
- CHECKSUM - Read checksum
- ERROR_AXIS - First axis to trip on error
- ERROR_LINE - Program line at which error occurred
- INDEVICE - Read current input device
- OUTDEVICE - Read current output device
- LOCK - Prevent users examining or editing programs
- RUN_ERROR - Error code for last recorded error
- SERVO_PERIOD - Set servo update period

Constants

- OFF - 0
- ON - 1
- FALSE - 0
- TRUE - -1
- PI - 3.14159

REGISTRATION EXAMPLE:

MATHS AND VARIABLES

Trio BASIC has different types of variables for different functions. Named variables with 16 significant letters may be used which are local to each process. In addition Trio BASIC supports battery backed global variables using VR() and large tables of data using the TABLE() command.

- ABS - Absolute value
- ACOS - Arc Cos
- AND - Logical and bitwise AND
- ASIN - Arc Sin
- ATAN - Arc Tan
- ATAN2 - ATAN2(x,y) function
- CLEAR - Clear variables
- COS - Cos function
- EXP - Exponential
- FRAC - Frac function
- INT - Integer part
- LN - Logarithm
- MOD - Modulus function
- NOT - Not function
- OR - Logical and bitwise OR
- RESET - Clear local named variables to 0
- SGN - Sgn function
- SIN - Sin function
- SQR - Square root function
- TABLE - Global user data array
- TAN - Tan function
- TSIZE - Returns size of user data array
- XOR - XOR function
- VR - Shared battery backed memory array

INPUT/OUTPUT AND SERIAL PORT CONTROL

- AIN - Reads one of 4 12 bit analogue input channels
Note - Analog Inputs are an optional feature
- CHR - Allows for printing of control characters
- DATE\$ - Print date from real time clock *
- DAY\$ - Print day from real time clock *
- DEFKEY - Allows redefinition of membrane keypad keys
- FLAG - Read/Set PLC flag bits
- FLAGS - Read/Set multiple flag bits
- GET - Read character from serial channel
- IN - Read inputs individually or as a block
- INPUT - Allows number to be input to a variable
- LINPUT - Allows line of text to be input to an array
- KEY - Allows test if characters are available
- MARK - Test if REGIST function is completed
- OP - Set single or multiple outputs
- PRINT - Allows printing to serial devices and network
- PSWITCH - Set output to be on over position sector
- READPACKET - High data transfer from serial port
- REGIST - Set registration mode and window area
- SEND - Transmit message via fibre-optic network
- SETCOM - Configure serial port settings
- TIME\$ - Print time from real time clock *

```

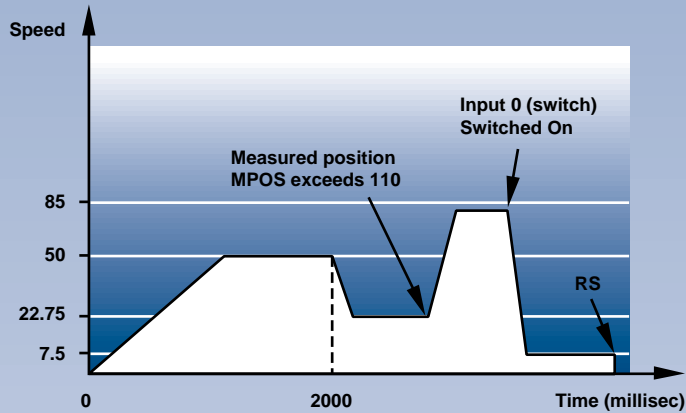
\
\ TrioBASIC Registration with CAM stretching
\
\ Set window open and close:
length=200
OPEN_WIN=10
CLOSE_WIN=length-10
GOSUB Initial
Loop:
TICKS=0' Set millisec counter to 0
IF MARK THEN
    offset=REG_POS
    ' Make offset -ve if at end of sheet
    IF ABS(offset-length)<offset THEN
        offset=offset-length
    ENDIF
    PRINT "Mark seen at:"offset[5.1]
ELSE
    offset=0
    PRINT "Mark not seen"
ENDIF
DEFPOS(0)
REGIST(3+768)' Allow mark end +/- 10mm
CAM(0,50,(length+offset*0.5)*cf,1000)
WAIT UNTIL TICKS>500
GOTO Loop
    
```

BREATHING LIFE INTO MACHINES

SPEED AND ACCELERATION CONTROL PROGRAMMING EXAMPLE:

```

Start Profile:
start:
  ACCEL=250
  DECEL=500
  FORWARD' Set forward until cancelled
  WA(2000)
  SPEED=22.75
  ACCEL=1000
  WAIT UNTIL MPOS>=110
  SPEED=85
  WAIT UNTIL IN(switch)=ON
  SPEED=7.5
  WA(500)
  RAPIDSTOP
  GOTO start
    
```



AXIS PARAMETERS (HELD FOR EACH MACHINE AXIS):

Axis parameters are dedicated function named variables held independently for each axis on the system. They can be read or written to as required by the program at any time.

- ATYPE - Axis Type
- UNITS - Unit conversion factor
- REPDIST - Machine Repeat Distance
- REP_OPTION - Set encoder repeat distance mode
- MICROSTEP - Enable microstepping mode
- PP_STEP - Stepper feedback encoder scaling factor
- SSI_BITS - SSI encoder resolution
- P_GAIN - Proportional Gain
- I_GAIN - Integral Gain
- D_GAIN - Derivative Gain
- OV_GAIN - Output Velocity Gain
- VFF_GAIN - Velocity Feedforward
- AFF_GAIN - Acceleration Feedforward (RESERVED)
- SPEED - Read or Set Speed
- CREEP - Creep Speed
- JOGSPEED - Jogging Speed
- MSPEED - Measured Speed
- VPSPEED - Velocity Profile Speed
- FHSPEED - Feedhold Speed
- ACCEL - Read Accel
- DECEL - Read Deceleration
- FASTDEC - Read Fast Deceleration
- MPOS - Measured Position
- DPOS - Demand Position
- OFFPOS - Demand position offset value
- SERVO - Servo ON/OFF control
- FELIMIT - Following Error Limit
- FE - Following Error
- FERANGE - Following Error Report Range *
- FEMIN - Stationary Following Error limit *
- FEGRAD - Following Error limit gradient *
- REG_POS - Registration Position
- MARK - Registration Event Flag
- OPEN_WIN - Registration Window Open
- CLOSE_WIN - Registration Window Close
- FWD_IN - Forward Limit Input
- REV_IN - Reverse Limit Input
- DATUM_IN - Datuming Input
- FHOLD_IN - Feedhold Input
- FAST_JOG - Fast Jog input
- FWD_JOG - Forward Jog Input
- REV_JOG - Reverse Jog input
- FSLIMIT - Forward Software Limit
- RSLIMIT - Reverse Software Limit
- DAC - Force Voltage to output
- DAC_OUT - Output generated by servo algorithm
- OUTLIMIT - Voltage Output Limit
- BOOST - Set the boost output on stepper board
- SRAMP - S ramp factor
- MERGE - Merge Moves Flag
- MTYPE - Read Move Type
- NTYPE - Read next move type
- ENDMOVE - Read position of end of move
- REMAIN - Read remainder of move
- LINKAX - Read link axis for Gearbox etc
- AXISSTATUS - Read Axis Status
- ERRORMASK - Error Mask
- FRAME - Set alternative coordinate transformation

APPLICATION EXAMPLE:

Filling Boxes with packing foam whilst they are moving on a conveyor which is not under control of the Motion Coordinator.

```

Example Motion Coordinator Program
Filling boxes moving on a conveyor:
start:  GOSUB initial
cycle:  WAIT UNTIL IN(0)=ON'      wait for box in position
        CONNECT(200/173,1) AXIS(0)' link axis 0 to conveyor now
        GOSUB fill_box'          do box filling sequence
        CANCEL AXIS(0)'          stop link to conveyor
        MOVEABS(0) AXIS(0)'      send carriage back to start
        GOTO cycle

fill_box: BASE(2)'              set default axis pointer to 2
        MOVEABS(50,50)'          goto start point in box
        WAIT IDLE '              wait till move finished
        OP(8,ON)'                lower nozzle via output 8
        WA(100)'                 wait 100 msec for nozzle
        OP(9,ON)'                foam on via output 9
        MOVEABS(150,50)'         move sequence relative to box:
        MOVEABS(150,150)
        MOVEABS(50,150)
        MOVEABS(100,100)
        WAIT IDLE                foam off
        OP(9,OFF)'               nozzle out
        OP(8,OFF)'               go back to park position
        MOVEABS(0,0)'
        RETURN

' Axis 0 is connected to linear slide running parallel to conveyor
' Axis 1 accepts encoder input from the conveyor
' Axis 2 has no physical connection but runs the moves parallel to the
' conveyor relative to the box. Movement on this axis is added
' to axis 0 using the ADPX command.
' Axis 3 is connected to linear slide at right angle to conveyor

Input 0 - box detect sensor
Output 8 - nozzle down output
Output 9 - foam on output

Set parameters not at defaults:
initial:
        ADPX(2) AXIS(0)'        add position from axis 2
        SERVO AXIS(0)=ON'       axis0 has motor connected
        SERVO AXIS(1)=OFF
        UNITS AXIS(2)=200
        SPEED AXIS(2)=500
        SERVO AXIS(3)=ON'       axis3 has motor connected
        UNITS AXIS(3)=200
        WDOG=ON'                enable amplifiers

        RETURN
    
```

