

AcroLoop Motion Controller

Overview

When configured with STEPware-100, Maple Systems' OIT Family Operator Interface Terminals (Maple OITs) can communicate with AcroLoop Motion Controllers. This document describes the various STEPware-100 settings and provides some simple programming examples.

Compatible Controllers	
Family	Model
AcroLoop	All

Communications Cable

The Maple OIT should be connected to the RS232 or RS422 communication port.

Refer to Technical Note 1061 for information on communication cable part numbers and cable assembly instructions. If you will be assembling your own communications cable, cable assembly instructions are also available on our web site at www.maple-systems.com.

WARNING: If your communications cable is not wired exactly as shown in our cable assembly instructions, damage to the Maple OIT or loss of communications can result.



STEPware-100 Settings

The following table lists the communications settings that must be configured in STEPware-100.

Please note:

- the Settings column lists STEPware-100's settings; for the AcroLoop Controllers
- the Options column lists STEPware-100's options; your Controller may not support every option

Name	Settings	Options	Important Notes
Baud Rate	9600	19200, 9600, 4800, 2400, 1200, 600, 300	Must match the controller port settings. Use the fastest baud rate supported by both.
Parity	None	Even, Odd, None, Mark, Space	
Data Bits	8	7, 8	
Stop Bits	1	1, 2	
Operating Mode	Block	Interactive, Block, Network Modes	Attached sample code requires Block Mode.
Line Terminator	CR	CR, LF, CR/LF, ETX	Attached sample code requires CR.
Turn-around Delay	No Delay	50, 100, 250, No Delay	
Handshaking	None	None, Xon/Xoff, RTS/CTS, both	
Password	00000	Any 5-digit value	Application specific
Local Echo	Enabled	Enabled/Disabled	
Local Setup Enabled	Enabled	Enabled/Disabled	
Local Keyboard Enabled	Enabled	Enabled/Disabled	
Key Click Enabled	Enabled	Enabled/Disabled	
Block Echo Enabled	Enabled	Enabled/Disabled	
Delayed Line Feed Enabled	Disabled	Enabled/Disabled	
Append Line Feed	Disabled	Enabled/Disabled	
Use 3-Wire RS485	Disabled	Enabled/Disabled	Available only in Network Mode.

NOTE: STEPware provides a full range of display and control functions. See STEPware Help, under *ESCAPE Control Commands*, for a complete list.

Sample Programs

Below are code fragments for performing common tasks.

In each of the following examples, the OIT should be set for **Block** mode. The Line Terminator should be set as **CR**.

Setting/Showing Drive Speed

```
100 OPEN "COM1:9600,N,8,1" AS #1      REM open serial port
110 PRINT #1, "New Speed: ";          REM text is sent to OIT
120 INPUT NewSpd$                     REM text is returned into NewSpd$
130 Speed=VAL(NewSpd$)                REM get new speed
140 VEL Speed                          REM set new speed
150 MOV X100                           REM start motor
160 PRINT "Speed is ", STR$(VEL)      REM speed is shown on OIT
170 DWL 2000                           REM wait for 2 seconds
180 CLOSE #1                           REM done w/ serial port
190 END                                REM all done
```

Setting/Showing Drive Position

```
100 OPEN "COM1:9600,N,8,1" AS #1      REM open serial port
110 PRINT #1, "New Position           REM text is sent to OIT
120 INPUT NewPos$                     REM text is returned into NewPos$
130 Pos=VAL(NewPos$)                 REM get new position
140 VEL 250                           REM set a speed
150 MOV XPos                           REM start motor
160 PRINT "Speed is ", STR$(P12290)  REM speed is shown on OIT
170 DWL 2000                           REM wait for 2 seconds
180 CLOSE #1                           REM done w/ serial port
190 END                                REM all done
```

Starting/Stopping the Motor from the OIT

Using Function Key ASCII Strings, the OIT can send messages to Start, Stop, and adjust the motor speed.

The OIT's Function Keys should be programmed as follows:

- **F1:** 1{CR}
- **F2:** 2{CR}
- **F3:** 3{CR}
- **F4:** 4{CR}

Each key should have the **Send Immediately** option checked.

Starting/Stopping the Motor from the OIT (Continued)

Use the following code:

```
100 REM F1 is Start, F2 is Stop, F3 is Increase Speed, F4 is Decrease Speed
110 ISpd=100 REM Initial Speed
120 SChg=25 REM Speed Inc/Dec Amount
130 MaxSpd=500 REM Max Speed
140 JOG VEL XISpd REM initial speed
150 OPEN "COM1:9600,N,8,1" AS #1 REM open serial port
160 INPUT, Cmd$ REM get new command
170 IF (Cmd$="1") THEN GOSUB 1000 REM Start Motor
180 IF (Cmd$="2") THEN GOSUB 1500 REM Stop Motor
190 IF (Cmd$="3") THEN GOSUB 2000 REM Increase Speed
200 IF (Cmd$="4") THEN GOSUB 2500 REM Decrease Speed
210 IF (BIT 1) THEN GOTO 160 REM Loop while Input 1 is ON
220 CLOSE #1 REM Done w/ serial port, close it
230 END REM all done

1000 JOG FWD X REM Start Motor Subroutine
1010 RETURN

1500 JOG OFF REM Stop Motor Subroutine
1510 RETURN

2000 ISpd=ISpd+SChg REM Increase Motor Speed Subroutine
2010 IF (ISpd>=MaxSpd) THEN GOTO 2030 REM if out-of-range, return
2020 JOG VEL XISpd REM set new speed
2030 RETURN REM back to caller

2500 ISpd=ISpd-SpdChg REM Decease Motor Speed Subroutine
2510 IF (ISpd<=0) THEN GOTO 2530 REM if out-of-range, return
2520 JOG VEL XISpd REM set new speed
2530 RETURN REM back to caller
```

Displaying a Pre-Programmed Message

The Motion Controller can instruct the OIT to display a pre-programmed message. This can be useful for showing alarm or status messages, without having to place code in the Motion Controller for a lot of Text messages.

The following code fragment will display message numbers 1-25.

```
100 MSG=0
110 REM format a message to the OIT: ESC m [msg num] STX
120 MSG=MSG+1
130 PRINT CHR$(27); "m"; STR$(MSG); CHR$(2);
140 IF MSG<25 THEN GOTO 120
150 END
```

Sounding the OIT's Built-In Buzzer

The Motion Controller can instruct the OIT to sound its buzzer for a specified number of seconds.

The following code fragment will sound the OIT's buzzer for 2 seconds.

```
100 REM format a message to the OIT: ESC g [seconds] STX
110 PRINT CHR$(27); "g2"; CHR$(2);
```

NOTE: STEPware provides a full range of display and control functions. See STEPware Help, under *ESCAPE Control Commands*, for a complete list.