

# Electro-Craft IQ-Series Servo Controllers

## Overview

When configured with STEPware-100, Maple Systems' OIT Family Operator Interface Terminals (Maple OITs) can communicate with Electro-Craft IQ-Series Servo Controllers. This document describes the various STEPware-100 settings and provides some simple IQ Basic examples.

| Compatible Controllers |                    |
|------------------------|--------------------|
| Family                 | Model              |
| Electro-Craft          | IQ-550, 2000, 5000 |

## Communications Cable

The Maple OIT should be connected to Serial Port 1.

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](http://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.

## Controller Settings

**NOTE:** The IQ Drive must have a firmware version of 2.10 or higher.



# 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 Electro-Craft IQ 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               | Xon/Xoff | 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

**NOTE:** The IQ Controller should be set for a **Generic Operator Terminal**, and the Maple OIT should be connected to Serial Port 1. The following is an IQ Master Manual excerpt, from the **IQ Master Environment/Parameter Menu/Serial** section, with appropriate information underlined.

## **Generic Operator Terminal**

The Generic check box selects the use of any generic operator terminal for the serial port 1 connection on P7. If this setting is cleared, the IQ Operator Terminal is selected. When a generic terminal is used, the IQ will not automatically send any special characters to the serial port. PRINT and READ statements can be used in a program to send any characters to the terminal. The built in status functions, Xkey functions, and Fkey functions cannot be used with a generic terminal.

The Generic check box (available for Port 1 only) is used to specify a generic operator terminal. No Electro-Craft Operator Terminal control codes will be automatically sent to the terminal when using PRINT, READ, and CLEAR commands if this box is selected. Control codes can still be sent to your terminal using the PRINT “^code” statement, however. The Generic option is available only in firmware versions 2.10 and later. If your firmware version is earlier than 2.10, the check box will be disabled.

If a generic operator terminal is used, the Op Term Key and the Codes list boxes are used to specify the codes the operator terminal generates. Select the specific key from the Op Term Key list box, then select the code for the key in the Codes list box. The code consists of two bytes—the high byte being the left most Code list box. If your codes are in hexadecimal, you must convert to decimal. For example, if the generic terminal sends ‘5Fh 0h’ for F1 ON and ‘6Fh 0h’ for F1 OFF, select ‘95 0’ for F1 ON and ‘111 0’ for F1 OFF. If the operator terminal sends only single byte codes, set the high byte of each key code to 0.

For keys that are not implemented, the key code should be set to 0. The Generic check box must be selected and the code for F1 ON must be non-zero so that the IQ will load the key codes on power up. The Generic check box does not have to be selected to set the key code values, however. If the operator terminal does not support the F1 ON function, a code that will never be sent must be set for the F1 ON code value. This option is available only in firmware versions 2.11 and later. If your firmware version is earlier than 2.11, the list boxes will be disabled.

When using PRINT “^Code”, the code must be in hexadecimal.

The Key Codes must be entered in order to activate the IQ’s Function Key Programs when a Function Key is pressed. It is not, however, required to use the Maple OITs’ Function Keys in this way. See the examples below.

## ***IQ Basic Examples***

Below are fragments of IQ Basic code 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**.

### **Additional Notes:**

- PRINT statements should only provide Port, String and Variable data.
- READ statements should only provide Port and Variable data.
- For cursor position, see below.

### **Setting/Showing Drive Speed**

```
PRINT #1 "New Speed: " ; text is sent to the OIT
READ V1 ; text is returned into NewSpd
MovV=V1 ; speed is set to new speed value
PRINT #1 "Speed is ",VCMD ; speed is shown on OIT
DELAY = 2 ; set pause time
MovV=0 ; set speed to 0
END ; all done
```

### **Setting/Showing Drive Position**

```
PRINT #1,"New Position: " ; text is sent to the OIT
READ #1, V1 ; text is returned into NewPos
MovP=V1 ; set position
WHILE V1>0 PRINT "Position is ",POSN
PRINT "Position is ",POSN
```

### **Setting Cursor Position**

```
; command format is ESC x [column] [line] STX
PRINT "^1Bx20^02" ; moves cursor to column 20, line 2
```

### **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:** 0{CR}
- **F2:** 1{CR}
- **F3:** 2{CR}
- **F4:** 3{CR}

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

## Starting/Stopping the Motor from the OIT (continued)

Use the following IQ Basic code:

```
; F1 is Start, F2 is Stop, F3 is Increase Speed, F4 is Decrease Speed
ASSIGN InitSpeed V1                ; Motor Initial Speed
ASSIGN SpdChange V2                ; Speed Increment/Decrement Amount
ASSIGN MaxSpeed V3                 ; Max Speed
ASSIGN Cmd V4                      ; Command from Operator Terminal

; initial values
InitSpeed=100
SpdChange=25
MaxSpeed=500

WHILE I1=ON                        ; Loop while Input #1 is ON
  READ Cmd                         ; get something from OIT
  IF Cmd=0 CALL M_START             ; Start Motor
  IF Cmd=1 CALL M_STOP              ; Stop Motor
  IF Cmd=2 CALL M_INCR              ; Increase Motor Speed
  IF Cmd=3 CALL M_DECR              ; Decrease Motor Speed

  ; invalid entry
  IF Cmd>3 PRINT"Invalid-Select F1-F4"
REPEAT

MovV=0                             ; stop motor
END                                 ; all done

; Start Motor Subroutine
SUB M_START
  MovV=InitSpeed
RET

; Stop Motor Subroutine
SUB M_STOP
  MovV=0 = 0
RET

; Increase Motor Speed
SUB M_INCR
  InitSpeed=InitSpd+SpdChange      ; get new speed
  IF InitSpeed>MaxSpeed JMP IncrDone ; don't exceed max speed
  MovV=InitSpeed                   ; set new speed
IncrDone:
RET

; Decrease Motor Speed
SUB M_DECR
  InitSpeed=InitSpeed-SpdChange     ; get new speed
  IF InitSpeed<1 JMP DecrDone       ; don't allow 0 or negative speed
  MovV=InitSpeed                    ; set new speed
DecrDone:
RET
```

## **Displaying a Pre-Programmed Message**

The servo 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 Servo for a lot of Text messages.

The following IQ Basic code will display message numbers 1-25.

```
V1=1
WHILE V1<25
  ; format a message to the OIT: ESC m [msg num] STX
  PRINT "^1Bm",V1,"^02"
  V1=V1+1
REPEAT
```

## **Sounding the OIT's Built-In Buzzer**

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

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

```
; format a message to the OIT: ESC g [seconds] STX
PRINT "^1Bg2^02"
```

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