Connect Tech Inc.

Meeting your objectives ... by design

Intellicon

Intelligent Serial Communications Adapters

User Manual

Connect Tech Inc
42 Arrow Road
Guelph, Ontario
N1K 1S6
Tel: (519) 836-1291
1-800-426-8979
Fax: (519) 836-4878
Email: sales@connecttech.com
FTP: ftp.connecttech.com
URL: http://www.connecttech.com

CTIM-00019, Revision 2.30 June 9 2000
Part Number: MAN001

LIMITED WARRANTY

Connect Tech Inc. warrants to the original purchaser of this Connect Tech hardware product that it is to be in good working order for a period of one year from the date of purchase from Connect Tech Inc. or an authorized Connect Tech dealer. Should this product, in Connect Tech Inc.'s opinion, fail to be in good working order during the warranty period, Connect Tech Inc. will, at its option, repair or replace this product at no charge, **provided that the product has not been subjected to abuse, misuse, accident, disaster or non Connect Tech Inc. authorized modification or repair.**

You may obtain Limited Warranty service by delivering this product to an authorized Connect Tech dealer or to Connect Tech Inc. along with proof of purchase. If this product is shipped, you agree to insure the product or assume risk of loss or damage in transit, and to prepay shipping charges to the warranty repair location.

The above warranty is the only warranty authorized by Connect Tech Inc. **UNDER NO CIRCUMSTANCES WILL CONNECT TECH INC. BE LIABLE IN ANY WAY FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF, OR INABILITY TO USE, SUCH PRODUCT.**

Copyright Notice

Trademark Acknowledgment

CLASS A COMPUTING DEVICE

Intellicon-2/-4/-6+2/-8/-Flex8

FCC

This equipment complies with the requirements in Part 15 of FCC Rules for a Class A computing device. Operation of this equipment in a residential area may cause unacceptable interference to radio and TV reception requiring the operator to take whatever steps are necessary to correct the interference.

DOC/IC

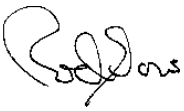
This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe est conforme à la norme NMB-003 du Canada.

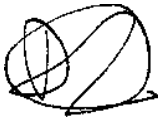
CE

Connect Tech Inc. declares that the product(s) covered by the contents of this manual have been tested and found compliant with the below listed standards as required by the Electromagnetic Compatibility (EMC) Directive for General Immunity Compliance, EN 50 0082.1.

EN 55022	Conducted and Radiated emissions
IEC 61000-4-2	Electrostatic Discharge
IEC 61000-4-3	Radiated Immunity
IEC 61000-4-4	Electrical Fast Transients



Rod Doré
Manager of Engineering



Dave Worthen
President

General

The above agency conformances were met by independent laboratory testing of Connect Tech Inc. product(s) with shielded cables, with metal hoods, attached to either the terminating connectors or cable assemblies supplied with the product(s). **Failure to follow good EMC/EMI compliant cabling practices may produce more emissions or less immunity than were obtained in laboratory measurements.**

Operation of this equipment in a residential area may cause unacceptable interference to radio a TV reception, requiring the user to take whatever steps necessary to correct the interference.

Table of Contents

Chapter 3: Software Installation (continued)

Chapter 3: Software Installation (continued)

Chapter 4: Appendices

Chapter 4: Appendices (continued)

List of Tables

List of Tables (continued)

List of Figures

List of Figures (continued)

List of Figures (continued)

Chapter 1: Introduction

Introduction

This section contains an overall description of the Intellicon manual, the Intellicon product, Connect Tech 's technical services, Return Merchandise Authorization and warranty repair policies. The organization of the information is as follows:

- Manual Overview - describes the structure of the manual and the conventions the manual uses.
- Product Overview - describes the Intellicon, its features and specifications.
- Customer Services Overview - describes the various customer support services available to users of Connect Tech products.
- RMA Overview - describes the RMA policy and procedures for all Connect Tech products.

Manual Overview

The purpose of the Intellicon User's Manual is to help you install the Intellicon adapter as effortlessly as possible. The manual includes three main sections:

- Introduction - the section you are currently reading, covers the Intellicon features; Connect Tech's customer services; and return merchandise authorization (RMA) policies and procedures.
- Hardware Installation - describes installation of the Intellicon adapter.
- Software Installation - discusses the installation of the Intellicon device drivers under various operating systems.

> ***Note:***
> *We recommend that you read both the **Hardware Installation** and **Software Installation** sections before you attempt to install the Intellicon adapter.*

- Appendices - the appendices contain information on electrical pinouts, product specifications such as power requirements; default jumper and DIP switch settings; RS-485/422 and 20mA Current Loop options; and other technical information.

Conventions & Symbols

This manual uses the following conventions:

- In most cases the font for file names and command statements is 9 point Arial Narrow Bold. An example is:

  DEVICE=C:\CTI\CTI.SYS a=D000 p=300 n=1 i=3 d=COM

- ***Note:** the "Note" convention informs you of important messages, exceptions , or special cases.*
- ***Example:** the "Example" convention outlines hardware and software installation examples.*
- ***Technical Tip:** the "Technical Tip" convention offers technical tips to assist you in hardware and software installation or problems*

- **WARNING**

  The "Warning" convention advises you to take certain precautions in order to avoid possible damage to your Connect Tech product.

- 

  This message warns you that your Connect Tech product is very sensitive to static electricity. Make sure that before handling the product you practice proper ESD procedures.

Product Overview

The Intellicon is a high performance intelligent multi-port adapter that allows you to connect up to 8 serial devices through one expansion slot and up to 32 serial devices per system.

Your Intellicon product consists of the following components:

- Intellicon adapter
- Cable harness or external connector box .

Note: *The Intellicon-2 does not require a cable harness or external connector box.*

- Intellicon device drivers

Intellicon Adapters

Intellicon adapters provide the high speed interfaces between a host computer and multiple external serial devices. Intellicon adapters off load from the host computer the task of managing the serial communication component of an application. Intellicon adapters include features such as:

- Intellicon adapters have an on-board processor to off-load the burden of managing serial communications from the main CPU.
- Intellicon adapters have 64KB of SRAM for data and program storage.
- Users can download specialized protocols to the on-board SRAM for custom applications.
- Rapid data transmission rates up to a maximum of 38.4K baud on each port.
- Memory and I/O port addresses are DIP switch selectable
- IRQ lines are jumper selectable from 2 to 7, 10, 11, 12, 14, 15.
- You can combine different models of Intellicon adapters in a host computer up to a maximum of 32 ports per system.
- System requirements are one full ISA bus compatible expansion slot; one available IRQ line; and one available 64KB segment in the first megabyte of memory address space.

Other features specific to particular models of Intellicon adapters are:

Intellicon-2 Adapters

- Allows you to connect two asynchronous RS-232 or RS-485/422 serial ports to a computer.

Intellicon-4 Adapters

- Allows you to connect four asynchronous RS-232 and/or RS-485/422 serial ports to a computer.

Intellicon-8 Adapters

- Allows you to connect eight asynchronous RS-232 serial ports to a computer.

Intellicon-6+2 Adapters

- Allows you to connect six asynchronous RS-232 and two asynchronous RS-485/422 serial ports to a computer.

Intellicon-Flex8 Adapters

- Allows you to connect eight asynchronous RS-232 and/or RS-485/422 and/or 20mA current loop serial ports to a computer.
- The RS-232, RS-485/422, RS-423 and 20mA current loop electrical line interfaces are on Serial Line Interface Modules (SLIM). This allows you to run different line interfaces on the adapter at the same time.
- The Serial Line Interface Modules (SLIM) are field upgradeable.

Figures 1, **2**, **3**, **4**, and **5** show the locations of hardware components on the various Intellicon adapters.

Figure 1: *Intellicon-2 adapter*

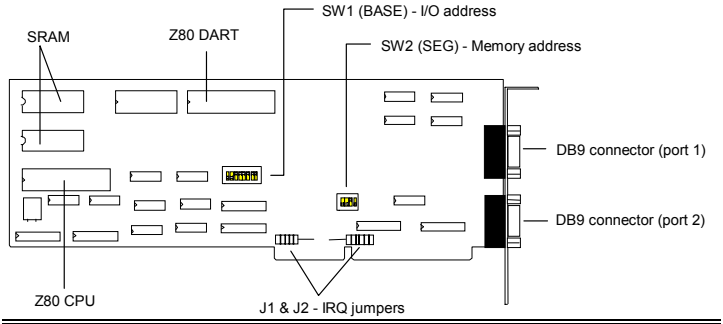


Figure 2: *Intellicon-4 adapter*

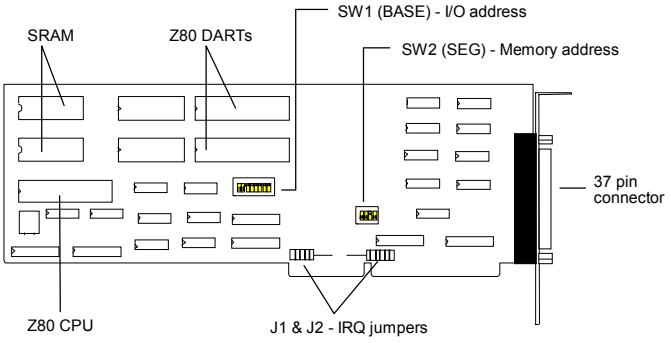


Figure 3: *Intellicon-8 adapter*

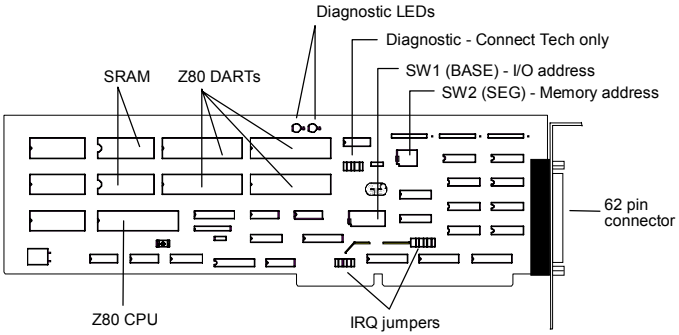


Figure 4: *Intellicon-6+2 adapter*

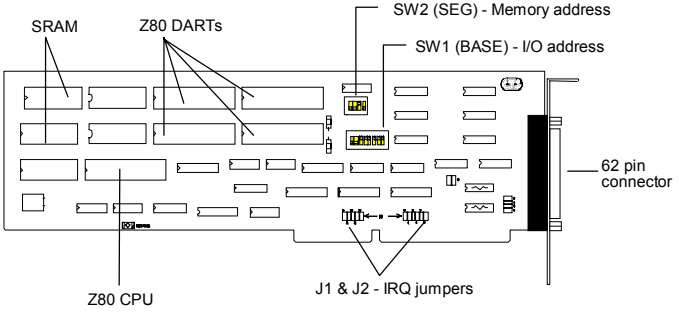
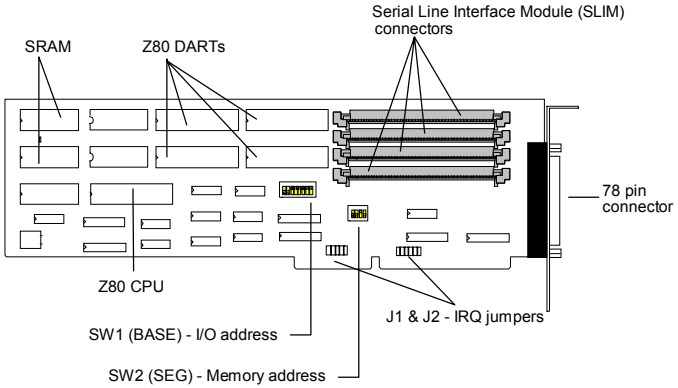
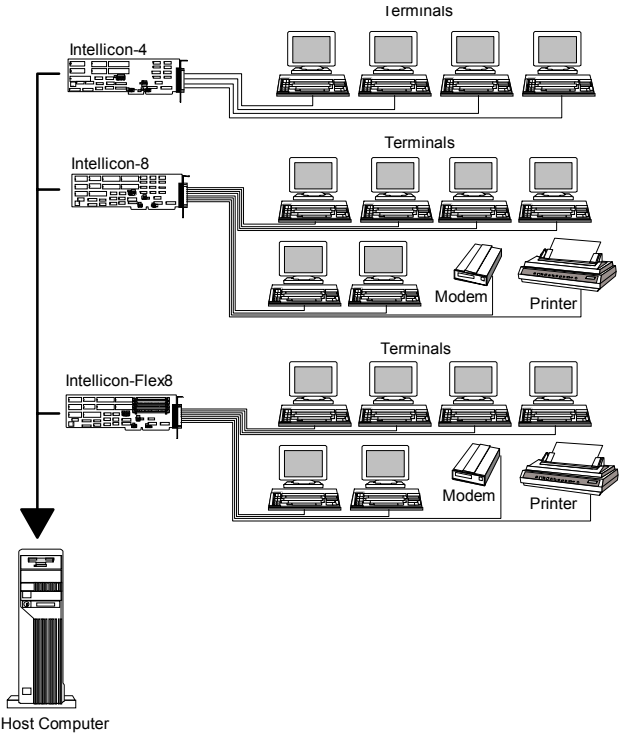


Figure 5: *Intellicon-Flex8 adapter*



The Intellicon adapters can accommodate both small and large multi-channel applications. You can combine different Intellicon adapters in a single computer, up to a maximum of 32 ports per system. **Figure 6** below represents a sample configuration for a system requiring 20 ports.

Figure 6: *Intellicon - 20 port configuration*



Intellicon Software

The Intellicon comes with its own real-time executive and software device drivers. The device drivers provide support for multi-port serial communications with different operating systems without the need for programming. See **Chapter 3: Software Installation** for driver installation instructions for the different operating systems.

Customer Service Overview

If you experience difficulties after reading the manual and using the product, contact the Connect Tech reseller from which you purchased the product. In most cases the reseller can help you with product installation and difficulties.

In the event that the reseller is unable to resolve your problem, our highly qualified support staff can assist you. Please refer to and complete the problem summary sheet found in **Figure 7** before contacting us.

Figure 7: *Problem summary sheet*

Problem Description	
Connect Tech Product Description	
Product:	Revision no.:
Product serial no.:	No. of serial ports:
IRQ selected:	Base address selected:
I/O port address selected:	
Device driver:	Revision no.:
System Description	
Operating system:	Revision no.:
System type & manufacturer:	
Amount of RAM:	CPU type/speed:
Video adapter:	Settings:
Network adapter:	Settings:
Hard disk adapter:	Settings:
Tape adapter:	Settings:
Other serial adapters	Settings:
Other adapters:	Settings:
Devices connected to ports:	

We offer four ways for you to contact us:

Mail/Courier

You may contact us by letter and our mailing address for correspondence is:

Connect Tech Inc.
c/o Customer Service
42 Arrow Road
Guelph, Ontario
Canada N1K 1S6

Email/Internet

You may contact us through the Internet. Our email, FTP and URL addresses on the Internet are:

sales@connecttech.com
support@connecttech.com
ftp.connecttech.com
http://www.connecttech.com

Note:

1. *You can submit your technical support questions to our customer support engineers via our Internet email address*

   OR

2. *You can refer to our knowledge database found in the Customer Support section of our World Wide Web site.*

   OR

3. *You can fill out the problem summary form, found in the Customer Support section of our World Wide Web site and submit it to our customer support engineers via the Web.*

   OR

4. *You can obtain the latest versions of software drivers and manuals from the Customer Support section of our World Wide Web site or from the /pub section on our FTP site.*

RMA Overview

Connect Tech products requiring warranty or non warranty repairs need an RMA number. To obtain a Return Merchandise Authorization (RMA) Number please contact:

Connect Tech Inc.
Technical Support
42 Arrow Road
Guelph, Ontario
Canada N1K 1S6
Phone: (519) 836-1291
1-800-426-8979
Facsimile: (519) 836-4878
Email: support@connecttech.com
FTP: ftp.connecttech.com
HTTP: //www.connecttech.com

Include with the product, proof of purchase (including date of purchase), a description of the problem and the RMA number. Clearly display the RMA number on the external packaging. Please refer to the Limited Warranty for further restrictions or requirements.

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from your computer, you wear an anti-static wrist-band. When you remove the board from your computer, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

Note:

1. *Please pack the item for repair securely and ship it* ***prepaid and insured. Connect Tech is not liable for damage or loss to the product due to shipping.***
2. *Connect Tech will not accept items for repair without an RMA number.*
3. *Connect Tech will not accept items for repair shipped freight collect.*

Chapter 2: Hardware Installation

Introduction

Hardware installation section involves configuration of the following Intellicon products:

- The Intellicon-2/-4/-8 adapters
- The Intellicon-6+2 adapter
- The Intellicon-Flex8 adapter

You must configure various settings and options before installing your Intellicon adapter in your computer. In order to insure a successful installation, please follow the steps in the order specified below.

1. Selection of I/O port addresses.
2. Selection of base memory addresses.
3. Selection of IRQs.
4. Installation of Serial Line Interface Modules (SLIM) **(Intellicon-Flex8 model only)**
5. Configuration of RS-232, RS485/422, Current Loop options **(Intellicon-2, Intellicon-4, Intellicon-6+2, and Intellicon-Flex8 models only**) and/or RS423 **(Intellicon-Flex8 only).**
6. Installing the Intellicon adapter in your computer

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

I/O Port Address Selection

In order for the host computer to communicate with the Intellicon adapter, each adapter requires an unique I/O address block (four consecutive I/O port addresses) in your computer's I/O address space.

Switch block SW1 configures the I/O port addresses for the Intellicon adapter. **Figures 8, 9, 10, 11,** and **12** show the location and orientation of SW1 on the Intellicon-2/-4/-6+2/-8/-Flex8 respectively.

Figure 8: *Intellicon-2: I/O port address(SW1)*

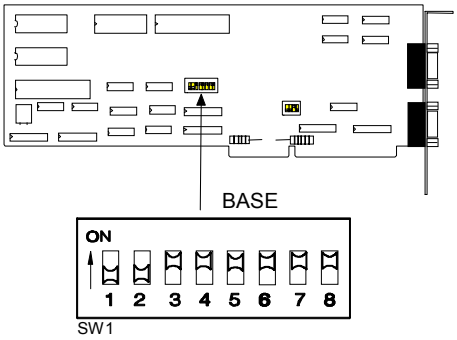


Figure 9: *Intellicon-4: I/O port address(SW1)*

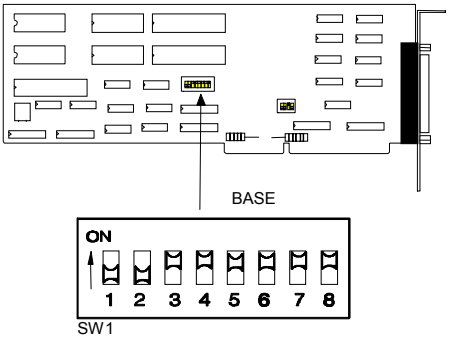


Figure 10: *Intellicon-6+2: I/O port address(SW1)*

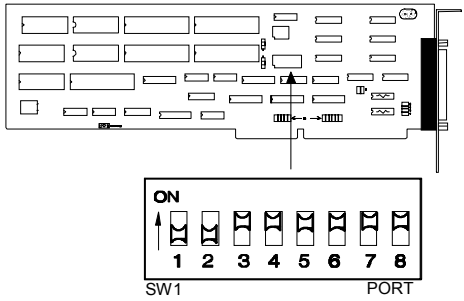


Figure 11: *Intellicon-8: I/O port address(SW1)*

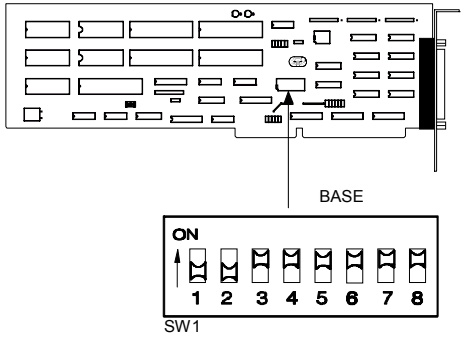
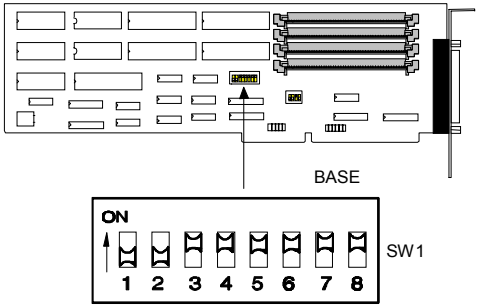


Figure 12: *Intellicon-Flex8: I/O port address(SW1)*



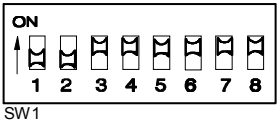
Intellicon adapters ship with SW1 factory set for port addresses 300-303(hex). However many other devices may require port addresses in your computer's I/O address space. In most cases the factory setting is sufficient, but when there is a conflict, SW1 may require a different setting. **Table 1** summarizes typical I/O port address usage and **Table 2** lists all the possible switch settings on SW1 for port addresses.

WARNING

Do not use a pencil to set the DIP switches as the lead graphite may short-circuit the switch.

Note: *If you install more than one Intellicon adapter in a computer, you must choose a different I/O port address for each adapter.*

Examples
The following example illustrates SW1 set for the factory default I/O port address setting of 300 - 303(hex)



The following example illustrates SW1 set for the I/O port address setting of 308 - 30B(hex)

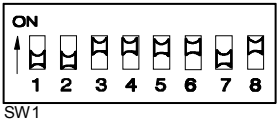


Table 1: *Typical port address usage*

Port Address (Hex)	Usage
000 - 0FF	Reserved
170 - 177	Fixed disk controller (primary)
1F0 - 1F7	Fixed disk controller (secondary)
200 - 207	Game port
238 - 23B	Bus mouse
278 -27F	Parallel port (LPT2)
2C0 - 2CF	EGA video adapter (#2)
2F8 - 2FF	Asynchronous serial port (COM2)
360 - 36F	Parallel port (LPT1)
370 - 377	Floppy disk controller (secondary)
378 - 37F	Parallel port (LPT1)
380 - 38F	SDLC adapter (#2)
3A0 - 3AF	SDLC adapter (#1)
3B0 - 3BF	Video adapters
3F0 - 3F7	Floppy disk controller (primary)
3F8 - 3FF	Asynchronous serial port (COM1)

Table 2: *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
100 - 103	on	off	on	on	on	on	on	on
104 - 107	on	off	on	on	on	on	on	off
108 - 10B	on	off	on	on	on	on	off	on
10C - 10F	on	off	on	on	on	on	off	off
110 - 113	on	off	on	on	on	off	on	on
114 - 117	on	off	on	on	on	off	on	off
118 - 11B	on	off	on	on	on	off	off	on
11C - 11F	on	off	on	on	on	off	off	off
120 - 123	on	off	on	on	off	on	on	on
124 - 127	on	off	on	on	off	on	on	off
128 - 12B	on	off	on	on	off	on	off	on
12C - 12F	on	off	on	on	off	on	off	off
130 - 133	on	off	on	on	off	off	on	on
134 - 137	on	off	on	on	off	off	on	off
138 - 13B	on	off	on	on	off	off	off	on
13C - 13F	on	off	on	on	off	off	off	off
140 - 143	on	off	on	off	on	on	on	on
144 - 147	on	off	on	off	on	on	on	off
148 - 14B	on	off	on	off	on	on	off	on
14C - 14F	on	off	on	off	on	on	off	off
150 - 153	on	off	on	off	on	off	on	on
154 - 157	on	off	on	off	on	off	on	off
158 - 15B	on	off	on	off	on	off	off	on
15C - 15F	on	off	on	off	on	off	off	off
160 -163	on	off	on	off	off	on	on	on
164 - 167	on	off	on	off	off	on	on	off
168 - 16B	on	off	on	off	off	on	off	on
16C - 16F	on	off	on	off	off	on	off	off
170 - 173	on	off	on	off	off	off	on	on
174 - 177	on	off	on	off	off	off	on	off
178 - 17B	on	off	on	off	off	off	off	on
17C - 17F	on	off	on	off	off	off	off	off

Table 2(continued): *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
180 - 183	on	off	off	on	on	on	on	on
184 - 187	on	off	off	on	on	on	on	off
188 - 18B	on	off	off	on	on	on	off	on
18C - 18F	on	off	off	on	on	on	off	off
190 - 193	on	off	off	on	on	off	on	on
194 - 197	on	off	off	on	on	off	on	off
198 - 19B	on	off	off	on	on	off	off	on
19C - 19F	on	off	off	on	on	off	off	off
1A0 - 1A3	on	off	off	on	off	on	on	on
1A4 - 1A7	on	off	off	on	off	on	on	off
1A8 - 1AB	on	off	off	on	off	on	off	on
1AC - 1AF	on	off	off	on	off	on	off	off
1B0 - 1B3	on	off	off	on	off	off	on	on
1B4 - 1B7	on	off	off	on	off	off	on	off
1B8 - 1BB	on	off	off	on	off	off	off	on
1BC - 1BF	on	off	off	on	off	off	off	off
1C0 - 1C3	on	off	off	off	on	on	on	on
1C4 - 1C7	on	off	off	off	on	on	on	off
1C8 - 1CB	on	off	off	off	on	on	off	on
1CC - 1CF	on	off	off	off	on	on	off	off
1D0 - 1D3	on	off	off	off	on	off	on	on
1D4 - 1D7	on	off	off	off	on	off	on	off
1D8 - 1DB	on	off	off	off	on	off	off	on
1DC - 1DF	on	off	off	off	on	off	off	off
1E0 - 1E3	on	off	off	off	off	on	on	on
1E4 - 1E7	on	off	off	off	off	on	on	off
1E8 - 1EB	on	off	off	off	off	on	off	on
1EC - 1EF	on	off	off	off	off	on	off	off
1F0 - 1F3	on	off	off	off	off	off	on	on
1F4 - 1F7	on	off	off	off	off	off	on	off
1F8 - 1FB	on	off	off	off	off	off	off	on
1FC - 1FF	on	off	off	off	off	off	off	off

Table 2(continued): *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
200 - 203	off	on	on	on	on	on	on	on
204 - 207	off	on	on	on	on	on	on	off
208 - 20B	off	on	on	on	on	on	off	on
20C - 20F	off	on	on	on	on	on	off	off
210 - 213	off	on	on	on	on	off	on	on
214 - 217	off	on	on	on	on	off	on	off
218 - 21B	off	on	on	on	on	off	off	on
21C - 21F	off	on	on	on	on	off	off	off
220 - 223	off	on	on	on	off	on	on	on
224 - 227	off	on	on	on	off	on	on	off
228 - 22B	off	on	on	on	off	on	off	on
22C - 22F	off	on	on	on	off	on	off	off
230 - 233	off	on	on	on	off	off	on	on
234 - 237	off	on	on	on	off	off	on	off
238 - 23B	off	on	on	on	off	off	off	on
23C - 23F	off	on	on	on	off	off	off	off
240 - 243	off	on	on	off	on	on	on	on
244 - 247	off	on	on	off	on	on	on	off
248 - 24B	off	on	on	off	on	on	off	on
24C - 24F	off	on	on	off	on	on	off	off
250 - 253	off	on	on	off	on	off	on	on
254 - 257	off	on	on	off	on	off	on	off
258 - 25B	off	on	on	off	on	off	off	on
25C - 25F	off	on	on	off	on	off	off	off
260 - 263	off	on	on	off	off	on	on	on
264 - 267	off	on	on	off	off	on	on	off
268 - 26B	off	on	on	off	off	on	off	on
26C - 26F	off	on	on	off	off	on	off	off
270 - 273	off	on	on	off	off	off	on	on
274 - 277	off	on	on	off	off	off	on	off
278 - 27B	off	on	on	off	off	off	off	on
27C - 27F	off	on	on	off	off	off	off	off

Table 2(continued): *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
280 - 283	off	on	off	on	on	on	on	on
284 - 287	off	on	off	on	on	on	on	off
288 - 28B	off	on	off	on	on	on	off	on
28C - 28F	off	on	off	on	on	on	off	off
290 - 293	off	on	off	on	on	off	on	on
294 - 297	off	on	off	on	on	off	on	off
298 - 29B	off	on	off	on	on	off	off	on
29C - 29F	off	on	off	on	on	off	off	off
2A0 - 2A3	off	on	off	on	off	on	on	on
2A4 - 2A7	off	on	off	on	off	on	on	off
2A8 - 2AB	off	on	off	on	off	on	off	on
2AC - 2AF	off	on	off	on	off	on	off	off
2B0 - 2B3	off	on	off	on	off	off	on	on
2B4 - 2B7	off	on	off	on	off	off	on	off
2B8 - 2BB	off	on	off	on	off	off	off	on
2BC - 2BF	off	on	off	on	off	off	off	off
2C0 - 2C3	off	on	off	off	on	on	on	on
2C4 - 2C7	off	on	off	off	on	on	on	off
2C8 - 2CB	off	on	off	off	on	on	off	on
2CC - 2CF	off	on	off	off	on	on	off	off
2D0 - 2D3	off	on	off	off	on	off	on	on
2D4 - 2D7	off	on	off	off	on	off	on	off
2D8 - 2DB	off	on	off	off	on	off	off	on
2DC - 2DF	off	on	off	off	on	off	off	off
2E0 - 2E3	off	on	off	off	off	on	on	on
2E4 - 2E7	off	on	off	off	off	on	on	off
2E8 - 2EB	off	on	off	off	off	on	off	on
2EC - 2EF	off	on	off	off	off	on	off	off
2F0 - 2F3	off	on	off	off	off	off	on	on
2F4 - 2F7	off	on	off	off	off	off	on	off
2F8 - 2FB	off	on	off	off	off	off	off	on
2FC - 2FF	off	on	off	off	off	off	off	off

Table 2(continued): *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
300 - 303	off	off	on	on	on	on	on	on
304 - 307	off	off	on	on	on	on	on	off
308 - 30B	off	off	on	on	on	on	off	on
30C - 30F	off	off	on	on	on	on	off	off
310 - 313	off	off	on	on	on	off	on	on
314 - 317	off	off	on	on	on	off	on	off
318 - 31B	off	off	on	on	on	off	off	on
31C - 31F	off	off	on	on	on	off	off	off
320 - 323	off	off	on	on	off	on	on	on
324 - 327	off	off	on	on	off	on	on	off
328 - 32B	off	off	on	on	off	on	off	on
32C - 32F	off	off	on	on	off	on	off	off
330 - 333	off	off	on	on	off	off	on	on
334 - 337	off	off	on	on	off	off	on	off
338 - 33B	off	off	on	on	off	off	off	on
33C - 33F	off	off	on	on	off	off	off	off
340 - 343	off	off	on	off	on	on	on	on
344 - 347	off	off	on	off	on	on	on	off
348 - 34B	off	off	on	off	on	on	off	on
34C - 34F	off	off	on	off	on	on	off	off
350 - 353	off	off	on	off	on	off	on	on
354 - 357	off	off	on	off	on	off	on	off
358 - 35B	off	off	on	off	on	off	off	on
35C - 35F	off	off	on	off	on	off	off	off
360 - 363	off	off	on	off	off	on	on	on
364 - 367	off	off	on	off	off	on	on	off
368 - 36B	off	off	on	off	off	on	off	on
36C - 36F	off	off	on	off	off	on	off	off
370 - 373	off	off	on	off	off	off	on	on
374 - 377	off	off	on	off	off	off	on	off
378 - 37B	off	off	on	off	off	off	off	on
37C - 37F	off	off	on	off	off	off	off	off

Table 2(continued): *Port address switch settings(SW1)*

	SW1 switch settings							
Port Address (Hex)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
380 - 383	off	off	off	on	on	on	on	on
384 - 387	off	off	off	on	on	on	on	off
388 - 38B	off	off	off	on	on	on	off	on
38C - 38F	off	off	off	on	on	on	off	off
390 - 393	off	off	off	on	on	off	on	on
394 - 397	off	off	off	on	on	off	on	off
398 - 39B	off	off	off	on	on	off	off	on
39C - 39F	off	off	off	on	on	off	off	off
3A0 - 3A3	off	off	off	on	off	on	on	on
3A4 - 3A7	off	off	off	on	off	on	on	off
3A8 - 3AB	off	off	off	on	off	on	off	on
3AC - 3AF	off	off	off	on	off	on	off	off
3B0 - 3B3	off	off	off	on	off	off	on	on
3B4 - 3B7	off	off	off	on	off	off	on	off
3B8 - 3BB	off	off	off	on	off	off	off	on
3BC - 3BF	off	off	off	on	off	off	off	off
3C0 - 3C3	off	off	off	off	on	on	on	on
3C4 - 3C7	off	off	off	off	on	on	on	off
3C8 - 3CB	off	off	off	off	on	on	off	on
3CC - 3CF	off	off	off	off	on	on	off	off
3D0 - 3D3	off	off	off	off	on	off	on	on
3D4 - 3D7	off	off	off	off	on	off	on	off
3D8 - 3DB	off	off	off	off	on	off	off	on
3DC - 3DF	off	off	off	off	on	off	off	off
3E0 - 3E3	off	off	off	off	off	on	on	on
3E4 - 3E7	off	off	off	off	off	on	on	off
3E8 - 3EB	off	off	off	off	off	on	off	on
3EC - 3EF	off	off	off	off	off	on	off	off
3F0 - 3F3	off	off	off	off	off	off	on	on
3F4 - 3F7	off	off	off	off	off	off	on	off
3F8 - 3FB	off	off	off	off	off	off	off	on
3FC - 3FF	off	off	off	off	off	off	off	off

Memory Address Selection

The Intellicon adapters communicate with the host computer through 64K blocks of shared memory. This block of shared memory lies within the first megabyte of the computer's memory address space.

Technical Tips:

1. *Most system BIOS's can "shadow" the upper memory region (640KB - 1MB) in 8K, 32K or 64K blocks. This can cause some adapters that use shared memory in this area to malfunction.* ***Please ensure that you disable Shadow RAM for the upper memory region used by your Intellicon adapter.***
2. *Most system BIOS's cache the upper memory region (640KB - 1MB). This can cause some adapters that use shared memory in this area to malfunction.* ***Please ensure that you disable Cache for the upper memory region used by your Intellicon adapter.***
3. *Your Intellicon adapter is not a plug and play (PnP) adapter. It requires an IRQ, I/O port addresses and a 64KB segment of shared memory in the region between 640 KB and 1 MB. It is often necessary to go into the CMOS setup (BIOS) and ensure that the resources used by the Intellicon are available to Non-PnP ISA devices. Look in the CMOS setup for PnP or ISA setup options.*

DIP switches 1 through 4 on switch block SW2 select the memory address for this 64K block of shared memory. Please refer to **Figures 13, 14, 15, 16,** and **17,** for the location and orientation of SW2 on the Intellicon-2/-4/-6+2/-8/-Flex8 respectively.

Figure 13: *Intellicon-2: Memory address(SW2)*

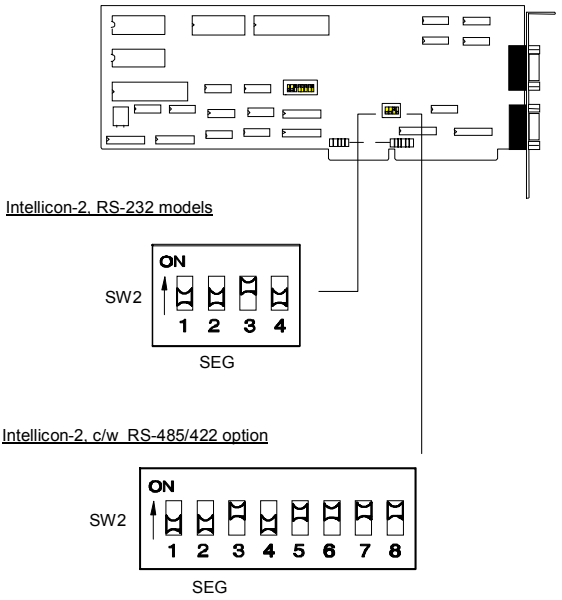


Figure 14: *Intellicon-4: Memory address(SW2)*

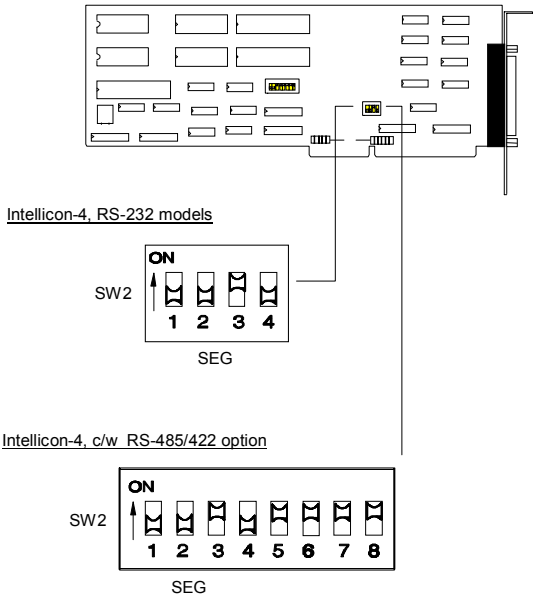


Figure 15: *Intellicon-6+2: Memory address(SW2)*

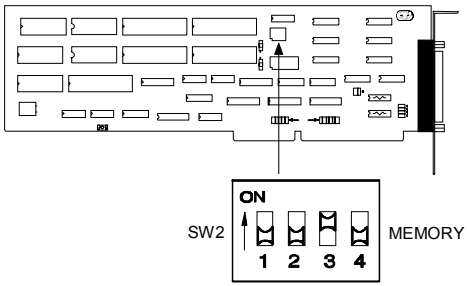


Figure 16: *Intellicon-8: Memory address(SW2)*

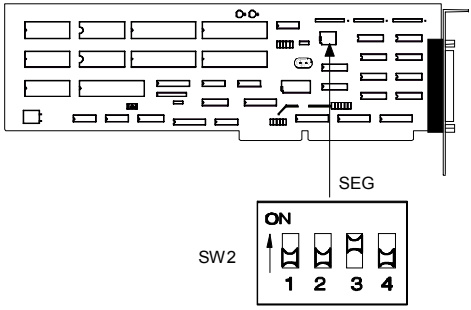
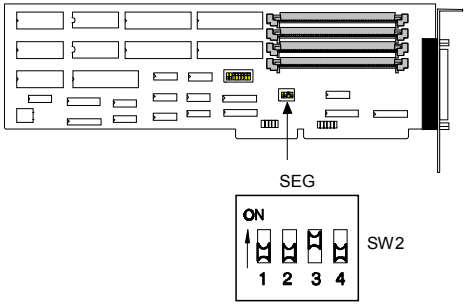


Figure 17: *Intellicon-Flex8: Memory address(SW2)*



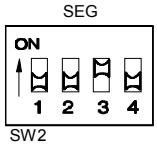
Intellicon adapters ship with SW2 set for a base memory address of D000 hex. In most cases this setting is sufficient, but where there are many different expansion devices installed in the host computer, SW2 may require a different base address setting in order to avoid a conflict.

WARNING

Do not use a pencil to set the DIP switches as the lead graphite may short-circuit the switch.

Note: *If you install more than one Intellicon adapter in a computer, you must choose the same base address for each adapter.*

Example:
The following example illustrates SW2 set for the factory setting of base address D000 hex



Example:
The following example illustrates SW2(SEG) on an Intellicon-2 or Intellicon-4 with the RS-485/422 option, set for the factory setting of base address D000 hex

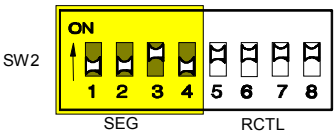


Table 3 outlines some typical expansion devices and the base address ranges they may occupy. **Table 4** shows SW2 switch settings for all possible base memory addresses.

Table 3: *Typical base address usage*

Device	Address Range (Hex)	Size
Upper user RAM	80000 - 9FFFF	128K
Video buffer RAM	A0000 - BFFFF	128K
EGA/VGA video memory	A0000 - BFFFF	128K
MDA video memory	B0000 - B7FFF	32K
CGA video memory	B8000 - BFFFF	32K
ROM expansion	C0000 - DFFFF	128K
EGA BIOS ROM	C0000 - C3FFF	16K
PGA comm area	C6000 - C63FF	1K
Cluster adapter BIOS	D0000 - D7FFF	32K
Adaptec controllers	DC000 - DFFFF	15K
BIOS expansion	E0000 - EFFFF	64K
BIOS	F0000 - FFFFF	64K

Table 4: *Memory address settings(SW2)*

Memory address (8K blocks - Hex)	SW2 switch settings 1	2	3	4
A0000 - AFFFF	off	on	off	on
B0000 - BFFFF	off	on	off	off
C0000 - CFFFF	off	off	on	on
D0000 - DFFFF	off	off	on	off
E0000 - EFFFF	off	off	off	on

IRQ Selection

Each Intellicon adapter requires one interrupt request vector (IRQ) to communicate with the Central Processing Unit (CPU) on the host computer.

The jumper blocks labelled J1, J2 or IRQ program your Intellicon adapter for an IRQ. The corresponding IRQ numbers are beside the block positions. **Figures 18** and **19** show the location and orientation of these jumpers on the Intellicon-2/-4/-6+2/-8/-Flex8.

Technical Tip:
Your Intellicon adapter is not a plug and play (PnP) adapter. It requires an IRQ, I/O port addresses and a 64KB segment of shared memory in the region between 640 KB and 1 MB. It is often necessary to go into the CMOS setup (BIOS) and ensure that the resources used by the Intellicon are available to Non-PnP ISA devices. Look in the CMOS setup for PnP or ISA setup options.

Figure 18: *IRQ jumpers for Intellicon-2/-4/-6+2/-Flex8*

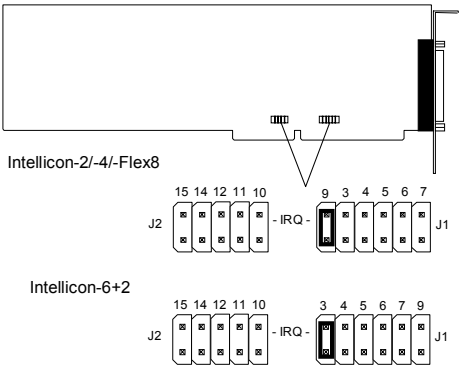
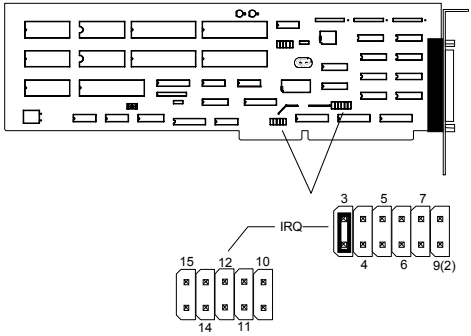


Figure 19: *IRQ jumpers for Intellicon-8*



Many other devices such as the keyboard and disk drives also require an IRQ. **Table 5** shows some typical IRQ assignments.

Table 5: *Typical IRQ assignments*

IRQ	Device
0	Timer output
1	Keyboard
2	Reserved
3	COM2, COM4, SDLC
4	COM1, COM3, SDLC
5	Unassigned (typically)
6	Floppy disk controller
7	LPT1
8	Real-time clock
9	Reserved
10	Unassigned (typically)
11	Unassigned (typically)
12	Unassigned (typically)
13	Co-processor
14	Primary fixed disk controller
15	Secondary fixed disk controller

In most cases the default IRQ setting for your Intellicon adapter is satisfactory. If an IRQ conflict exists between the Intellicon adapter and another expansion device, you must choose another IRQ. To select an IRQ simply install the jumper across the corresponding pins for that IRQ.

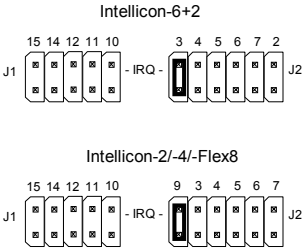
ATTENTION
Contents
Static Sensitive

Your Intellicon adapter is very sensitive to static electricity. When setting the jumpers make sure that you wear an anti-static wrist-band. Handle the adapter by its edges and place it on the anti-static bag or an anti-static mat.

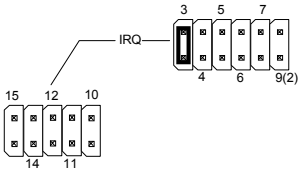
Note: *If you install more than one Intellicon adapter (up to four per system) in a computer you must select an unique IRQ for each adapter.*

Example*:*

1. *The following example shows J1-J2(IRQ) jumpers set for the factory setting of IRQ 3.(Intellicon-2/-4/-6+2/-Flex8)*



2. *The following example shows IRQ jumpers set for the factory setting of IRQ 3.(Intellicon-8)*



Electrical Interface Options

You can order the Intellicon-2 and the Intellicon-4 adapters with RS-485/422 interfaces. If your Intellicon-2 and Intellicon-4 adapter has the RS-485/422 interface installed, please refer to **Appendix F: RS-485/422 Option.**

The Intellicon-6+2 comes with six ports of RS-232 and two ports of RS-485/422. Please refer to **Appendix F: RS-485/422 Option**.

The Intellicon-Flex8 has four SLIM sockets to accept Connect Tech's RS-232 and/or RS-485/422 and/or RS-423 and/or 20mA current loop Serial Line Interface Modules(SLIM). If you order RS-232 modules please refer to **Appendix E: RS-232 Option** for more technical information. If you order RS-485/422 modules please refer to **Appendix F: RS-485/422 Option** for more technical information. If you order RS-423 modules please refer to **Appendix G: RS-423 Option** for more technical information. If you order 20mA current loop modules please refer to **Appendix H: 20mA Current Loop Option** for more technical information

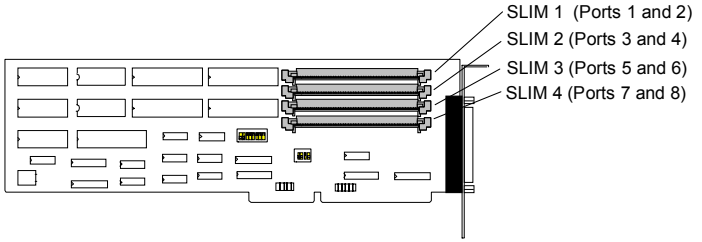
SLIM Installation(Intellicon-Flex8)

The Intellicon Flex-8 communication controller has four 35 pin SLIM sockets which accept line transceiver modules. Each transceiver module provides the RS-232 or RS-485/422 or RS-423 or 20mA current loop receivers and transmitters to condition two serial ports. SLIM 1 interfaces ports 1 and 2, SLIM 2 interfaces ports 3 and 4, SLIM 3 interfaces ports 5 and 6, and SLIM 4 interfaces ports 7 and 8. Refer to **Figure 20** for the location and orientation of the SLIM sockets on the Intellicon-Flex8

WARNING

You must insert Serial Line Interface Modules into all of the SLIM sockets before operating your Intellicon adapter.. Failure to observe this precaution will result in damage to the Intellicon adapter.

Figure 20: *Intellicon-Flex8: SLIM socket locations*



To insert a SLIM into it's connector follow these steps:

1. Place the SLIM into the socket almost vertically (about 75 degree angle), making sure that it is properly oriented and fully inserted into the socket.
2. Press downward and sideways on the SLIM until it latches into the socket. **Do NOT force the SLIM**, the installation process requires a small force and should be very smooth and easy. If you encounter resistance then re-check the orientation and insertion depth. Refer to **Appendix D: SLIM Insertion/Removal** regarding the insertion of a SLIM

To remove a SLIM from it's connector follow these steps:

1. Using both hands, place your thumbs on the clips that hold the SLIM into the socket, and place your index fingers on the edge of the SLIM.
2. Push outwards (with your thumbs) on the clips until the SLIM is released from the clips. Your index fingers can then raise the SLIM past the clips. The SLIM is then free to be removed from the Intellicon-Flex8 board. The force required to open the clips is small and the SLIM naturally springs upwards as soon as the clips are open wide enough. If removal is difficult, then you are probably trying to raise the SLIM before the clips are open. Refer to **Appendix D: SLIM Insertion/Removal** regarding the removal of a SLIM.

The RS-232, RS-485/422, RS-423 and 20mA current loop SLIMs are very sensitive to static electricity. Make sure that before you remove the SLIMs from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

Installing the Intellicon Adapter

To install your Intellicon adapter in your computer follow these steps:

Computer components are very sensitive to static electricity. When installing adapters in your computer make sure that you wear an anti-static wrist-band. Handle the adapter by its edges and place it on the anti-static bag or an anti-static mat.

1. Turn the power off to your computer.
2. Open your computer to expose the expansion slots (consult the system documentation for information on this procedure.)
3. Choose an available full length 8 or 16-bit expansion slot.
4. Remove the screw and the expansion slot cover from the slot you select and save both.

5. Place the Intellicon adapter in the expansion slot and push down gently until the card seats fully in the slot.

Note: *Do not force the card into the expansion slot. If you meet a great deal of resistance, remove the board and try again.*

6. Align the mounting bracket and secure the board with the screw that you saved.
7. Close your computer.

Chapter 3: Software Installation

Introduction

This chapter describes the installation of Intellicon software device drivers for different operating systems. Intellicon software device drivers support:

- BSD/OS
- PC-DOS & MS-DOS
  - Versions 3.XX & up
- QNX
  - Versions 2.XX & up
  - Versions 4.XX & up
- UNIX
  - SCO UNIX
  - SCO XENIX
  - Interactive UNIX
  - Unixware, v.1.1.X
- Windows 3.XX
- Windows NT

Connect Tech also offers custom driver development services. Please contact Connect Tech Sales for further information on custom driver development.

Before installing the Intellicon software device driver, verify and note your switch and jumper settings for:

- The Intellicon adapter's base memory address
- The Intellicon adapter's I/O port addresses
- The Intellicon adapter's IRQ settings.

BSD/OS Device Driver

The Intellicon BSD/OS device driver provides an interface between the BSD/OS operating system and an Intellicon adapter.

Under BSD/OS, the Intellicon driver will link into the kernel to generate a new run time system. The diskette includes an installation program that configures the driver, links it to the kernel and creates the special device files.

> ***Note:***
> *The diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver software utilities*

Installation

Before proceeding with the installation, locate the appropriate Intellicon installation diskette for BSD/OS. To determine the contents of each diskette, carefully check each label and any addendum included with this manual. You will find the BSD/OS Intellicon driver and installation script on the installation diskette in the compressed tar file **cti**.tar.Z**. To install the Intellicon BSD/OS driver please follow these steps:

1. First use **tar** to copy the file cti**.tar.Z from the installation diskette to a temporary directory by typing:

   ```
   cd /tmp
   tar xvf /dev/fd*
   ```

   Where **fd*** is the name of the floppy drive.

2. You then must uncompress the **cti**.tar.Z** file by typing:

   ```
   uncompress cti**.tar.Z
   ```

3. You must now untar this file by typing:

   ```
   tar xvf cti**.tar
   ```

4. You may now install the Intellicon driver into the BSD/OS system. Run the **cti_install** script by typing :

   ```
   /cti_install
   ```

DOS Device Driver

The Intellicon DOS device driver provides an interface between the DOS operating system and an Intellicon adapter. Under DOS you can install a maximum of 32 ports using any combination of Intellicon adapters.

Installation

To install the DOS device driver follow these procedures:

> ***Note:***
> *The diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver and software utilities*

> ***Technical Tip:***
> *Memory management software such as EMM386™, 386MAX™ or QEMM™, must not manage the shared memory region used by your Intellicon adapter. These software packages usually have an "exclude" command to prevent the use of a specified area of memory. Here are a few examples of QEMM™, 386MAX™ and EMM386™ statements using exclude commands:*
>
> **device= c:\qemm\qemm.sys exclude=d000-dfff**
> **device= c:\386max\386max.sys exclude=d000-dfff**
> **device= c:\windows\emm386.exe x=d000-dfff**

1. Create a directory named CTI on your C: drive. Insert the Intellicon DOS device driver diskette into a floppy drive and copy all the files found in the DOS directory of the Intellicon device driver diskette to the CTI directory on your system's hard drive.

> ***Example:***
> *To copy the files from floppy drive "A:" to a directory called "CTI" on hard drive "C:" the syntax is:*
>
> **copy a:\DOS*.* c:\CTI**
>
> ***Note:***
> *Please add* **C:\CTI;** *to your* **PATH=** *statement in your* **AUTOEXEC.BAT** *file*

2. Add the following entry to your **CONFIG.SYS** file and then reboot your computer:

DEVICE=C:\CTI\CTI.DOS A=nnnnn F=file N=nn D=name I=nn P=nn H={N|Xnnmm|C|P}

Where:

A= The memory segment used for the dual ported memory. This will default to D000 (hex) if not specified. Note that the dual ported memory takes up a 64K footprint (i.e.: you use the region D000 to DFFF when you select D000 as the base address).

F= The file name that contains the firmware for the Intellicon board. This will default to **\Z80OS.DOS** if not specified port address used by the Intellicon adapter. This defaults to 300 - 303 (hex).

N= The offset for "COM" ports. When using **CTI.DOS** along with standard serial ports, you must enter an offset where the Intellicon ports begin. For example the setting **N=3** will start the Intellicon ports at COM3.

D= The prefix to use in the device names (maximum of 6 characters). The default is **COM**

I= The interrupt request line (IRQ) used by the Intellicon adapter and specified in decimal. The default is 3 for the first board.

H= The initial handshake mode to use. This will default to **N** (none) if not entered. Valid handshake modes are:

N	none
Xnnmm	XON/XOFF protocol where: **nn=** XON in hex (default is 11 hex) **mm=** XOFF in hex (default is 13 hex)
C	CTS/RTS
P	PCTerm XPC protocol

Example:
When entering the handshaking modes, you specify for all ports on a board in one string. For example the statement:

H=C, C, N, , X, X, P

yields the following:

port 1	*CTS/RTS*
port 2	*CTS/RTS*
port 3	*none*
port 4	*none (not specified)*
port 5	*XON/XOFF (using 11 and 13 hex)*
port 6	*XON/XOFF*
port 7	*XPC*
port 8	*none (not specified)*

/ Indicates the next board. Parameters for multiple boards are separated with a **/**.

Notes:

1. *You require only one* **DEVICE=** *statement in the* **CONFIG.SYS** *for the Intellicon adapter, even when using multiple Intellicon boards.*

2. *When using multiple Intellicon adapters in one system, each adapter requires a unique IRQ setting.*

3. *You use a space as a delimiter between each parameter and a* **"/"** *as a delimiter between each board.*

4. *In some applications, the name* **COMn** *invokes special device handling (i.e.: BASIC). The software will assume a standard serial port rather than a character device. To avoid this, you can use the* **D=** *option to change the device name prefix.*

Example:
Installing a single Intellicon adapter with the following parameters:

memory segment:	*D000 hex*
port address:	*300 hex*
port offset:	*COM1*
interrupt - IRQ:	*10*
device name prefix:	*COM*
firmware file on hard disk:	*C:\CTI*

The **DEVICE=** *statement for the* **CONFIG.SYS** file *should read:*

```
DEVICE=C:\CTI\CTI.DOS a=D000 f=c:\CTI\z80os.dos p=300 n=1 i=10 d=COM
```

Example:
Installing two Intellicon adapters in a system that has two serial ports (installed as COM1 and COM2). The **DEVICE=** *statement for the* **CONFIG.SYS** *file should read:*

```
DEVICE=C:\CTI\CTI.DOS a=D000 f=c:\CTI\z80os.dos p=300 n=3 i=10 d=COM
/ p=304 i=11 d=COM
```

Where:

firmware file on hard disk:	*C:\CTI*

Where the parameters for board #1 are:

memory segment is:	*D000 hex*
port address is:	*300 hex*
port offset is:	*3*
interrupt is:	*10*
device name prefix is:	*COM*

And the parameters for board #2 are:

memory segment is:	*D000 hex*
port address is:	*304 hex*
interrupt is:	*11*
device name prefix is:	*COM*

Note:
The two previously installed serial ports remain as COM1 and COM2, while the Intellicon ports are COM3 onwards. Also note that both boards share the same memory segment and **CTI.DOS** *will arbitrate between the two boards.*

Accessing Ports

There are two methods of accessing the extra serial ports on the Intellicon adapter.

- Character Device Interface
- INT 14h Functions

Character Device Interface

The simplest method of access is through file pointers (FCB or handle). The **CTI.DOS** driver provides a DOS character device interface that allows opening and reading or writing as a file named **COMn**.

> ***Note:***
> *In some applications, the name* **COMn** *invokes special device handling (i.e.: BASIC). The software will assume a standard serial port rather than a character device. To avoid this, you can use the* **d=** *option to change the device name prefix.*

In order to configure the extra serial ports on the Intellicon adapter, please refer to the **CTIMODE** section below.

CTIMODE

The **CTIMODE** command allows you to configure the extra serial ports on the Intellicon adapter. It provides the same functionality and defaults as the DOS **MODE** command with the following exceptions:

- Does not support redirection of LPTs
- Does not support time out retry parameter P
- Supports extra baud rates
- Supports setting of protocols

Type in the following statement or add it to your **AUTOEXEC.BAT** file

CTIMODE COMn[:] baud[,[parity] [,[databits] [,[stopbits][,[protocol]]]]

Where **[protocol]** can be:

N=	none
X[,nn[,mm]]=	Xon-Xoff **nn** = XON, **mm** = XOFF in hex (default is 11 and 13 hex)
C=	CTS/RTS
P=	Xon-Xoff for PC Term

Example:

CTIMODE COM4 9600,ODD,8,1,X

The above is an example of a **CTIMODE** *command statement configuring an Intellicon serial port as:*

COMn = COM4
baud = 9600
parity = odd
databits = 8
stopbit = 1
protocol = XON/XOFF

INT 14h Functions

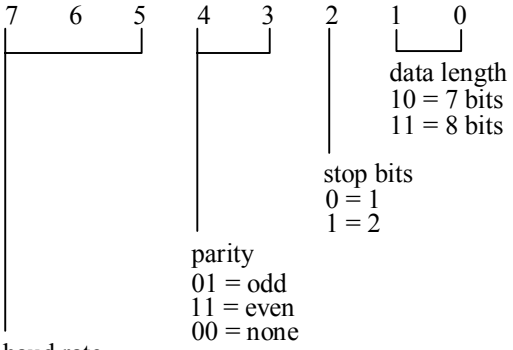
A more powerful and more complicated method of accessing and configuring the extra ports is through the INT 14h function calls. The first four functions are identical to the standard BIOS functions for INT 14h. We provide an additional 13 function calls for increased functionality. The function calls are:

Function 0 - Initialize the port

Input

ah = 0
al = baud, parity, length
dx = communication channel number(i.e. 0 = COM1, 1 = COM2,)

Bit designations for al are:



Return Value

ah = port status (see **Function 3**)
al = modem status

Description

This function allows for the selection of traditional port parameters. For more information on port initialization see **Function 4** (i.e. baud rates above 9600 bps).

Function 1 - Output character

Input
ah = 01h
al = character to send
dx = communication port number

Return Values
ah = port status (bit 7 set if unable to send)
al = character to send

Description
This function attempts to send a character. The Intellicon adapter buffers the data when it is busy. When it is free, the Intellicon transfers the data from the buffer to its queue.

Function 2 - Receive character

Input
ah = 02h
dx = communication port number

Return Values
ah = port status (bit 7 set if timeout has occurred)
al = character received

Description
This function tries to get a character from a port. If an empty input buffer receives no character, then bit 7 of ah is set to indicate a timeout has occurred (182 ticks).

Function 3 - Return port status

Input
ah = 03h
dx = communication port number

Return Values
ah = port status
al = modem status

Description
This function returns the port status. Bit designations are as follows:

ah bit
0 = data ready
1 = data overrun
2 = parity error
3 = framing error
4 = break interrupt
5 = transmitter holding register empty
6 = transmitter shift register empty
7 = timeout

al bit
0 = change in CTS
1 = change in DSR (always 0)
2 = change in RI (always 0)
3 = change in CD
4 = CTS
5 = DSR (always 0)
6 = RI (always 1)
7 = CD

Function 4 - Extended port initialization

Input

ah = 04h
al = parity, stop bits, length (described in **Function 0**)
cx:bx = baud rate (cx contains highword)
dx = communication port number

Return Values

ah = 0 on success

Description

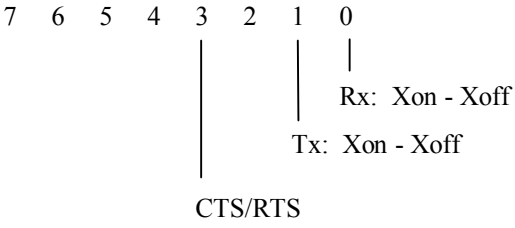
This function allows the setting of all Intellicon supported baud rates, including the ones not supported by **Function 0**.

Example:
19200 bps: cx = 0, bx = 4B00.(19200 decimal = 0004B00 hex)
9600 bps: cx = 0, bx = 2580 (9600 decimal = 0002580 hex)

Function 5 - Set protocol

Input
ah = 05h
al = protocols (0 for none)
bl = XON character
bh = XOFF character
dx = communication port number

Bit designations for al are:



Return Values
ah = FFh on invalid protocol specification

Description
This function allows the user to set protocols for each port. If you specify the Xon-Xoff protocol and bx is non-zero, then bx becomes the specified XON/XOFF characters.

Function 6 - Identification

Input
ah = 06h
dx = communication port number

Return Values
ah = high bit set
al = highest function number supported

Description
This function allows you to identify the Intellicon driver as an intelligent communication board interface.

Function 7 - Break modem connection

Input
ah = 07h
al = when bit 0 is set, bx contains break duration in ticks
when bit 0 is clear, break length will be 9 timer ticks
bx = number of timer ticks to hold break on line if al bit 0 is set
dx = communication port number

Return Values
None

Description
This function allows a program to send a **BREAK** on the communication line.

Function 8 - Nondestructive read

Input
ah = 08h
dx = communication port number

Return Values
al = next character in queue if zero flag is clear, else no characters are waiting

Description
This function allows the user to check the queue for the next character to read. This sets a zero flag if no characters are waiting.

Function 9 - Flush buffers

Input
ah = 09h
dx = communication port number

Return Values
None

Description
This function will flush (throw away the data) all the Intellicon input and output buffers for the port.

Function 10 - Input queue check

Input
ah = 0Ah
dx = communication port number

Return Values
ax = number of characters waiting for input

Description
This function allows a program to determine the number of characters waiting in the input buffer.

Function 11 - Disable a port

Input
ah = 0Bh
dx = communication port number

Return Values
ah = FFh if an invalid port number was specified

Description
This function allows the user to disable a given port, freeing up its resources for use by other ports.

Function 12 - Read current parameters

Input
ah = 0Ch
dx = communication port number

Return Values
ah = FFh if an invalid port number was specified
al = parameters as described in **Functions 0, 4**
cx, bx = baud rate (cx high order)

Description
This function allows the user to query a port for its parameters.

Function 13 - Register a port

Input
ah = 0Dh
dx = communication port number

Return Values
es:bx = will set a pointer to a byte flag to non-zero if there is a character waiting in the input queue.
ah = FFh if an invalid port number was specified

Description
This function provides a byte pointer to a flag indicating that a character is waiting for input. When data is waiting, the memory location pointed to is non zero. By registering a port and testing the memory location pointed to, implementation of polled operation is far faster than a nondestructive read. The reason for this is that you check the flag without the overhead of an INT 14h call.

Function 14 - String output

Input

ah = 0Eh
cx = maximum number of characters to transfer
dx = communication port number
es:bx = pointer to string
si = timeout length

Return Values

ax =number of characters transferred. If the zero flag is clear (0), the transfer was successful, else a timeout occurred during the transfer, i.e. the output buffer is full.

Description

This function allows for the transfer for multiple characters via one "INT" call rather than a single character in one call, thereby reducing software interrupt overhead. If si is zero, the function uses an internal timeout. If si is non-zero, si is the number of timer ticks allowed to send one character. If any one character in the string takes more than si ticks to send, the function returns with the zero flag set and ax will contain the count of characters sent before timing out.

Function 15 - String input

Input

ah = 0Fh
cx = maximum number of characters to read
dx = communication port number
es:bx = pointer to receiving buffer
si = timeout length

Return Values

ax =number of characters transferred. If the zero flag becomes (1), a timeout occurred waiting for a character from the port.

Description

Function 15 allows for more than one character to be read during a single INT call, and therefore decrease the INT overhead significantly. This function tells the driver the maximum number of characters to transfer. The driver will always return in ax the number of characters transferred. If si is zero, the function uses an internal timeout. If si is non-zero, it is the number of timer ticks allowed to receive each character. If more than si timer ticks go by before receiving a character, the function returns with the zero flag set.

Function 16 - Reserved

Function 17 - Lower/raise DTR & RTS

Input

ah = 11h
al = when bit 0 is set, raise DTR
when bit 0 is reset, lower DTR
when bit 1 is set, raise RTS
when bit 1 is reset, lower RTS
dx = communication port number

Return Values

None

Description

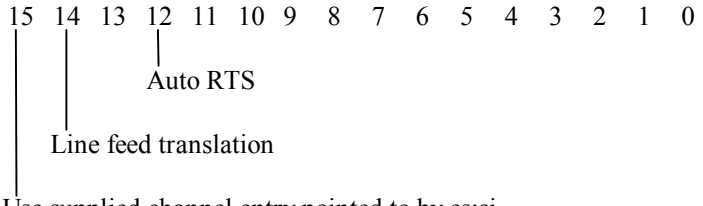
This function allows the user to change the states of the RTS and DTR handshake lines.

Function 18 - Special feature control

Input:

ah = 12h
bx = options to be changed (bit = 1 for change)
cx = set/reset values for options defined by bx

Bit designations for bx and cx are:



dx = communication port number
es:si = pointer to Z80 channel entry structure if high bit of bx is set

Channel entry structure definitions are as follows:

```
reserved        db    dup (?)
noedit_length   db    ?
reserved        db    3 dup (?)
noedit_eot1     db    ?
reserved        db    ?
noedit_eot2     db    ?
reserved        db    8 dup (?)
baud            dw    ?
parity          db    ?
stop_bits       db    ?
data_bits       db    ?
tty_options     dw    ?
reserved        db    6 dup (?)
out_xon_xoff    dw    ?
in_xon_xoff     dw    ?
reserved        db    18 dup (?)
```

Noedit_length is the maximum packet size. Noedit_eot1 and noedit_eot2 are optional characters signifying the end of packet (set to 0 to ignore).

Return Values:
bx = current option settings
es:si = current Z80 channel entry settings if high bit of bx was set

Description:
This function provides control for the special options available in the plus version of the Z80 OS (Z80OSP.DOS). It can also be used to gain access to the Z80 channel entry for a given port. To query the options set bx to 0 and the options will be returned in bx. To query the Z80 channel entry set bx to 8000 hex and cx to 0 and the information will be returned at the address specified by the pointer es:si. Care must be taken to reserve 56 bytes starting at es:si when using channel entries and the structure should be loaded by using this function to query the settings before changing any of them. It is recommended that this function only be used for configuring special features and querying board settings.

Enhanced Features

Special Protocols

An optional version of the Z80 firmware has been provided to allow some additional processing of data on the Intellicon board. This special processing includes:

- RTS - RTS operation for switched lines
- CRLF - linefeed to carriage return input translation
- NOEDIT - data packetization

To accommodate these features, you must specify **Z80OSP.DOS** as the OS file for the Intellicon board using the **F=** option. The command **CTISET** (or INT 14h Function 18) can then be used to turn on or off the required feature on each individual port.

RTS

When the RTS feature is requested, the RTS signal will only be asserted when there is data to be transmitted. After the last stop bit of the last byte has been transmitted, RTS will be removed allowing other devices access to the switched line.

Example: **CTISET COM4: +rts**
RTS is asserted when data is transmitted on COM4.

CRLF

The CRLF feature provides input translation of CR to CR, LF to CR, and CRLF pairs to a single CR.

Example: **CTISET COM5 +crlf**
CR to CR, LF to CR, and CRLF translation on COM5

NOEDIT

When NOEDIT is turned on for a port, data received for that port will be buffered (packetized) until one of three events occur.

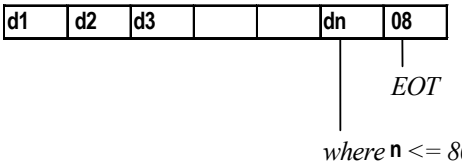
- one of 2 programmable end-of-transmission characters are received.
- a user specified minimum packet size is received.
- 255 characters are received.

Using this feature may substantially reduce the burden on the host computer.

Example:

If the format of the data to be received is known to be as follows:

1. *maximum of 80 characters*
2. *end of packet always marked with 08 hex*



The command syntax is:

CTISET COM6: +noedit eot1=08 eot2=08 len=80

CTISET

The **CTISET** command allows the user to control the extra features of the optional **Z80OSP.DOS**, as well as allowing the reallocation of data buffers on the board. The command syntax is:

CTISET [COM]n[:] [options]

where **[options]** are:

in=dd	% of total buffer space to allocate for input.
+/-rts	turn on/off special rts handling.
+/-crlf	turn on/off special linefeed translation.
+/-noedit	turn on or off packetization.
eot1=hh	specify end of transmission character 1 (0 to ignore)
eot2=hh	specify end of transmission character 2 (0 to ignore)
len=nnn	specify minimum packet length

Example:
If the format of the data to be received is always the following:

1. *byte#1 = character count*
2. *byte#2 = 01 hex - marks start of packet*
   *(**Note:** count is known to be never smaller than 2 bytes)*
3. *byte#3 = data*

The command syntax is:

CTISET COM3: +noedit eot1=01 eot2=01 len=255

The data will flow from the Intellicon adapter to your program as: a count followed by a 01 hex(2 bytes), followed by "count" bytes. One could use this to efficiently read data using string input Int14h calls.

In the Intellicon DOS driver about 90% of the buffers are allocated to output. In some applications it is desirable to allocate more space to input. This can be done by specifying the percentage to use for input with the **in=** parameter. Since character buffers are allocated in blocks, the reported percentage may vary from the requested amount. At least one buffer will be allocated for input and output.

Error Messages

Unable to Open Code file
Unknown symbol in load file
Unknown record type in load file
Missing or corrupt **Z80OS.DOS**

Unable to verify code into coprocessor.
I/O or Memory Address of board improperly specified or memory address conflict.

Interrupt not received from coprocessor.
IRQ conflict or mis-specification or possible hardware failure.

Invalid option -
An unknown option was specified on the command line. Edit the **CONFIG.SYS** file.

Error - device name too long
Optional device name specified must be 6 characters at most.

QNX Device Driver

The Intellicon QNX device driver provides an interface between the QNX operating system (V. 2.XX and V. 4.XX) and an Intellicon communications adapter. The Intellicon QNX driver software is a device administrator that executes after the system powers up, (usually from a **SYS.INIT** file) and runs in background continuously.

QNX Version 2.XX

There are two administrators supplied on the QNX software diskette. The default driver **CTI** provides the standard interface between QNX 2.XX and the serial ports. The optional driver **CTITP** provides the standard interface, as well as an extra set of devices which allow access to printers attached to terminals.

Also on the software diskette are files which contain the code which is downloaded to execute on the Intellicon board. The default, **"/config/z80os.qnx"** once again provides the standard set of features for QNX serial devices. The optional file **"/config/z80os.qnx.plus"** provides enhanced features and transparent print handling.

Note:
The Intellicon driver diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver and/or software utilities

Intellicon QNX 2.XX driver software is compatible with QNX's **Dev** task. Applications designed to use standard serial ports under **Dev** can use Intellicon adapters with little or no modifications

Installation

To install the Intellicon QNX 2.XX device driver follow these procedures:

1. Insert the Intellicon QNX 2.XX device driver diskette into a floppy drive and copy all the files on to your system's boot drive.

> ***Example:***
> *For example, if you mount your hard drive as disk 3 and your floppy drive is disk 1, insert the Intellicon QNX driver diskette into the floppy drive and type:*
>
> **backup 1:/ 3:/ +a**

2. Add the following entry to your **SYS.INIT** file:

cti [irq, port][options] &

Where **[irq, port]** are:

i=interrupt_line Interrupt line to use. When using multiple boards, you specify one interrupt line for each board.
default: **i=3**

> ***Technical Tip:*** *we recommend using either IRQ's 10, 11, or 15*

p=port_address The port address of the board. When using multiple boards you specify one port address for each board.
default: **p=300 (hex)**

and **[options]** are:

a=memory_segment Shared memory segment. When using multiple boards, you only specify one memory segment.
Default is **a= D000**

n=device_name Base name of the adopted devices.
default: **n=$cti**

s=signal_port Signal port to use. When using multiple boards, you specify only one signal port. default: **s=interrupt_line + 1**

> ***Technical Tip:*** *we recommend specifying* **s=19**

f=os_file_name File name that contains the Z80 OS. default: **f=/config/z80os.qnx**

g=pre_grow Pre-grow amount (in K bytes). This tells the administrator how much to pre-grow its data segment. The use of this parameter is usually not required on unprotected operating systems and is NEVER required when using protected operating systems. Default: **g=10+** (4 * number of boards)

Note:

1. *When using multiple boards, a comma is used to separate the parameters for each board.*
2. *Due to certain restrictions with QNX Version 2.XX, you can install a maximum of 40 devices which can include up to 32 Intellicon ports using any combination of Intellicon adapters.*

Example:

cti a=E000 i=7 &

The above is an example of a **SYS.INIT** *statement for one Intellicon adapter in your system.*

Where:

memory_segment:	*E000 hex*
interrupt_line:	*7*

The non-specified parameters will use the following defaults:

device_name:	*$cti*
signal_port:	*s=8*
port_address:	*p=300 hex*
os_file_name:	*/config/z80os.qnx*
pre_grow:	*g=14*

Example:

cti a=E000 i=3 p=300 , i=7 p=310 f=/config/z80os.qnx.plus &

The above is an example of a **SYS.INIT** *statement for two Intellicon adapters in your system*

Where the enhanced version of the firmware code will execute:
os_file_name: /config/z80os.qnx.plus

Where the CTI administrator will run in:
memory_segment is: E000 hex

One Intellicon adapter will use:
interrupt_line: i=3
port_address: p=300 hex

The other Intellicon adapter will use:
interrupt_line: i=7
port_address: p=310 hex

The non-specified parameters will use the following defaults:
signal_port: s=4
device_name: n=$cti
pre_grow: g=18

Initialization

When the **CTI** or **CTITP** administrator begins execution, it initializes the hardware then boots the processor on the adapter. This operation is not lengthy but usually takes a second.

Technical Tip:
For QNX Version 2.XX it is advisable that you place a **sleep 5** *command in your* **SYS.INIT** *file following the* **CTI** or **CTITP** *command. This will allow the administrator time to initialize. If you do not place a* **sleep** *command after the* **CTI** *or* **CTITP** command, **stty** *commands executed from the* **SYS.INIT** *file with the adopted serial ports will probably generate* **Invalid Device** *messages because the administrator has not completed its initialization.*

QNX Line Editing

Intellicon adapters can perform on-board processing of almost all of the **stty** options. When you use the **stty +edit** option the Intellicon adapter processes the line editing options on-board.

> ***Note:*** *The only* **stty** *option not supported by the Intellicon QNX driver software is the* **+/-lock** *option (opening a task or multiple tasks for write)*

Enhanced Features

When the **z80os.qnx.plus** software is executing on the Intellicon board, additional data processing and line control features may be invoked on each individual serial line. These features are usually queried and set by the **ctty** program. Syntax for the **ctty** statement is:

ctty [options]* [>device] [<device]

Where **[options]** are:

+/-pcterm	turn on or off PCTerm key scan code processing (see **PCTerm** section below)
+/-rts	turn on or off the Auto RTS feature (see the **Auto RTS** section below).
+/-crlf	the CRLF feature provides input translation of CR to CR LF to CR, and CRLF pairs to a single CR.
+/-noedit	turn on or off special input handling (see **Noedit Processing** section below)

PCTerm

Many terminals support a mode of emulation known as PCTerm. In this mode, the terminal transmits an unique code (key scan) for the pressing and release of each key. When **+pcterm** has been set on a port, the Intellicon board will translate these key scan codes into QNX key codes. This allows PCTerm terminals to operate like QNX terminals.

Example:
To turn on PCTerm processing for a port, use the following statement:

```
ctty +pcterm </dev/cti1
```

To turn off PCTerm processing for a port, use the following statement:

```
ctty -pcterm </dev/cti1
```

To translate output to a PCTerm terminal, a TCAP entry has been included in the file "**tcap.dbase.2**". If required it can be added to the standard data base with the command:

```
tcap append /config/tcap.dbase.2 pcterm
```

Auto RTS

The Auto RTS feature allows you to assert the RTS signal when there is data to be transmitted. After the last stop bit of the last byte has been transmitted, RTS will be dropped allowing other devices access to the switched line. This feature is very useful when doing half duplex or multi-drop RS-485/422 communications.

Example:
To turn on Auto RTS processing for a port, use the following statement:

```
ctty +rts </dev/cti1
```

To turn off Auto RTS processing for a port, use the following statement:

```
ctty -rts </dev/cti1
```

Noedit Processing

When implementing software to support communication protocols, frequently edit mode must be turned off. This may give rise to a large software overhead as the application may be executed (unblocked) for each character received. To lessen both the interrupt and the task switching overhead, the Intellicon board can be put into a mode where it will buffer data until a packet is received. If the standard QNX edit is off and noedit is on, the data will be buffered until:

- up is received
- down is received
- eot is received
- esc number of characters are received
- 255 characters are received

These parameters can be set with **stty** or **set_attr** to describe a packet. If any are not required, set them to 0.

Transparent Print

When **CTITP** is executed with the option "**f=/config/z80os.qnx.plus**" the administrator allows access to printers attached to the auxiliary ports of the terminals. The device is opened with a "**,p**" appended to the name of the serial port (i.e.: if port 1 is called **$cti1** then the printer is **$cti1,p**).

When there is no data to be sent to the terminal screen and yet there is data to be sent to the printer, the Intellicon board will transmit a start print command followed by the data. If a transmission is generated for the terminal screen, the Intellicon board will send a stop print command before the data. Before accessing the printer, the start and stop print commands must be set for the **",p"** device. This may be done with a **stty** command or with a **set_attr** function call. A start print command of up to 4 characters is programmed into break, esc, rub and can. A stop print command is programmed into eot, up, right and down. Any trailing unused characters should be 0.

Example:
For example, if your terminal starts printing with **"<esc>d#"** *and stops with* **"<cntrl>T"** *then use:*

```
stty break=1b esc=64 rub=23 can=0 eot=14 up=0 right=0 down=0 >$cti1,p
```

Utilities

CTIVIEW

CTIVIEW is a monitor which displays status for a single Intellicon board. It executes at priority 14 and constantly polls the shared memory on the board to update the screen.

For more information on the use of **CTIVIEW**, please refer to the readme file **CTIVIEW.DOC** on the Intellicon QNX 2.XX driver diskette.

DLA - Data Line Analyzer

DLA is a software utility that allows you to monitor and analyze data transmitted through the Intellicon adapter. **DLA** features include:

- you can monitor data as it is captured, and re-examine the captured data.
- You can save captured data to disk, for examination at a future date

- **DLA** gives you the flexibility to choose which ports to monitor, (a maximum of two ports simultaneously) as well as which "stream" of data to monitor for the individual ports (ie. the stream can be set to monitor transmitted or received data for the individual ports).
- You can monitor and examine the data in either character or hexadecimal mode.
- **DLA** allows you to monitor the "status" of the data flow. With the status feature turned on, you can witness transitions in the DCD and CTS lines, as well as the occurrences of parity, framing and overrun errors and breaks on the line.

For more information on the use of **DLA**, please refer to the readme file **DLA.DOC** on the Intellicon QNX 2.XX driver diskette.

Note:
DLA *does not operate with CTI Administrators previous to v2.14Z10*

CTI_BUFF

The **CTI_BUFF** is a software utility to adjust the allocation of Intellicon memory buffers. Buffers on the Intellicon are allocated during startup according to a default algorithm. **CTI_BUFF** allows you to change the allocation of I/O buffers to better suit your application.

For more information on the use of **CTI_BUFF**, please refer to the readme file **CTI_BUFF.DOC** on the Intellicon QNX 2.XX driver diskette.

Note:
CTI_BUFF *does not operate with CTI Administrators previous to v2.14Z10*

CTI_X

The **CTI_X** software utility allows users to change the value of the flow control characters from the default values to their own values.

For more information on the use of **CTI_X**, please refer to the readme file **CTI_X.DOC** on the Intellicon QNX 2.XX driver diskette.

Note:
CTI_X *does not operate with CTI Administrators previous to v2.14Z10*

Error Messages

QNX port dd in use by tid xxxx.
The signal port that **CTI** is attempting to use is already in use by another task. Re-run **CTI** using the **s=** parameter to change the signal port.

Must be a super user.
CTI must be run as a super user. Either login as a super user and run **CTI** or place the **CTI** invocation in your **sys.init** file.

Must be run as a background task
CTI must be run as a background task. Re-run **CTI** with an "**&**" on the command line. The "**&**" will place **CTI** in the background.

Unable to initialize card. Return status xxxx.
This message is always preceded by another error message. Please refer to the other error message.

Unable to adopt devices.
There are no more virtual ttys available. Try decreasing the number and re-run **CTI**.

Segment must fall on a 64K boundary.
The segment you specified using the **a=** parameter does not fall on a 64K boundary. Change the **a=** parameter so the specified segment does fall on a 64K boundary.

Unknown option
You have specified an unknown option on the command line of **CTI**.

Cannot create selector for segment at xxxx.
This means one of two things: the system is out of selectors or the ownership of the **CTI** file is not the super user. On some versions of QNX if the file is not owned by the super user, selector creation will fail. You may **chattr** the file so that it is owned by the super user.

Insufficient memory.
CTI cannot allocate enough memory for its device tables. Add more memory to your system or remove some tasks.

Unknown symbol cc.
Unknown record type xx.
The Z80 OS file (default **/config/z80os.qnx**) has become corrupted.

Cannot verify code into coprocessor.
The verify-after-write of the OS on the Intellicon has failed. Check to make sure that the memory segment and IO port address settings match those on the command line of **CTI**.

Cannot open OS file file_name.
When **CTI** attempted to load the OS onto the Intellicon board, it could not open the OS file. Check that the file exists. If using the default, the file **/config/z80os.qnx** is used for the OS file.

Use i=9 instead of i=2.
When using an AT (or AT compatible) and IRQ 2, it is necessary to specify the interrupt as **i=9** on the command line of **CTI** for proper operation. Note, this error cannot be detected on real-mode operating systems. Be careful.

No interrupt from card x: port=xxx int=x
No response from card x: port=xxx int=x
Interrupt request line not properly set, or hardware problem.

Unable to grow data segment
A pre-grow value has been used which is too large.

Warning Error Messages

Only able to adopt dd of the nn ports.
There was a shortage of virtual ttys available for adoption. **dd** tells you how many of the ports will be available.

Short on memory. Use g=nnK to increase. Current value g=ddk.
There is a shortage of pre-allocated memory available for **CTI**. Increase the amount of pre-allocated memory using the **g=** parameter.

msg=dd mflags=hh mdrive_number=d mdev_no=d marg=hhhh hhhh
An invalid message was received by the **CTI** administrator. Offending task will remain reply blocked. Check program that sent the bad message.

Other Errors

COMM will not autobaud a serial port.
Some versions of **COMM** will not work properly on adopted devices. Download a newer version of **COMM** from Quantum's update service.

MOUNT will not display the adopted serial ports on the Intellicon.
Earlier versions of **MOUNT** do not display adopted devices. Download a newer version of **MOUNT** from Quantum's update service.

stty ioport=hh will not change the I/O port of the Intellicon.
CTI will not allow another task to change its i/o port through an **stty_message**. **CTI** assumes that it was run with the proper i/o port setting on the command line.

The receipt of an **stty** message to change the i/o port will not, however, return an error to the sending task.

Messaging to DEV
On some of versions of the QNX operating system, some functions have been coded to message with **DEV** (task 3) rather than the task which has adopted the device, i.e. **get_tty**, **set_tty**.

Warning to QNX Network users
The Intellicon board defaults match those on the QNX Network board. The memory segment on either board must be reconfigured. Consult the configuration section of the manual. Most of the time this can be fixed by readdressing the network card to address CC000 (hex) or CE000 (hex).

QNX Version 4.XX

There are four administrators supplied on the Intellicon QNX 4.XX driver diskette. The default drivers **Dev.cti**(for QNX 4.22 or earler) and **Dev32.cti**(for QNX 4.23 or later) provide the standard interface between QNX 4.XX and the serial ports. The optional drivers **Dev.cti.tp**(for QNX 4.22 or earler) and **Dev32.cti.tp**(for QNX 4.23 or later) provide the standard interface, as well as an extra set of devices which allow access to printers attached to terminals.

Also on the diskette are files which contain software code for enhanced features such as transparent print control and software utilities such as **chmon** and **chmon32**.

Note:

1. *The Intellicon driver diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver and/or software utilities. You will find the README files in the* **/usr/cti** *directory*
2. **Dev32.cti** *and* **Dev32.cti.tp** *are 32 bit drivers for QNX v.4.23 or later*

Technical Tip: *You* ***must*** *run the QNX 4.XX* **Dosfsys** *utility before installation of the Intellicon driver. This utility allows you to access DOS formatted diskettes. To run* **Dosfsys** *add the following entry to your* **sysinit** *file or type it at the command prompt.*

```
Dosfsys &
```

Intellicon QNX 4.XX driver software is compatible with QNX's device administrator **Dev**. Applications designed to use standard serial ports under **Dev** can use Intellicon adapters with little or no modifications

Installation

To install the QNX 4.XX device driver follow these procedures:

1. Insert the Intellicon QNX 4.XX device driver diskette into floppy drive A and type:

   install -u /dos/a/qnx4/qnx4

   Note:
   *If you want to use the **PCTerm** option, you must insert the Intellicon QNX 4.XX device driver diskette into floppy drive A and type:*

   install -u /dos/a/qnx4/qnx4pct

2. Add the following entry to your **/etc/config/sysinit** file:

   Dev.cti [port],[irq] [options] &
   or
   Dev32.cti [port],[irq] [options] &

   Where:
   [port] are the I/O port addresses (default: 300 - 303)
   [irq] is the Interrupt Request Line (default: 3)
   [options] are:

-a number	Base address (in hex) of Shared Memory (default: D0000).
-b number	Define initial baud rate (default: 9600).
-C number	Size of canonical input buffer (default: 256).
-d	Set DTR on initially.
-D	Set DTR off initially.
-e	Set options to "edit" mode.
-E	Set options to "raw" mode.
-f	Enable hardware flow control.
-F	Disable hardware flow control.
-h number	Define high-water mark for input flow control (default: 3/4 point of input buffer).
-I number	Size of raw input buffer (default: 2048).
-l number	Define low-water mark for input flow control (default: 1/4 point of input buffer).
-m	Disable modem carrier processing (HUP).
-M	Enable modem processing (HUP).
-n	Similiar to the **-N** device prefix parameter, but allows a unique prefix specification for each Intellicon adapter in a system.
-N name	Prefix to register (default: "cti").
-O number	Size of output buffer (default: 2048).
-p number	Set port number for subsequent options.

-q	Run board in polled mode (no hardware IRQ)
-r	Set RTS on initially.
-R	Set RTS off initially.
-t number	Number of kilobytes of Intellicon memory to test (default: 64)

Note:

1. *QNX* **Dev** and **Dev32** *only support 32 devices by default. When using multiple Intellicon boards, increase the number of devices using the* **-n** *flag on the command line of* **Dev/Dev32**. *For example the following* **Dev/Dev32** *commands support 40 devices total:*

   Dev -n 40 & (for QNX 4.22 or earlier)
   Dev32 -n 40 & (for QNX 4.23 or later)

2. *Due to certain restrictions with the Intellicon QNX 4.XX device driver, you can install a maximum of 64 ports using any combination of Intellicon adapters.*

Example 1: **Dev.cti &**

Where:

memory =	*D0000 hex*	*(default)*
I/O port addresses =	*300-303 hex*	*(default)*
IRQ =	*3*	*(default)*

Note: *Other parameters will retain defaults*

Example 2: **Dev.cti -a e0000 &**

Where:

memory = E0000 hex

Note: *Other paramters retain their default values*

All examples listed for **Dev.cti** *or* **Dev.cti.tp** *apply to* **Dev32.cti** *and* **Dev32.cti.tp**, *but you must substitute* **Dev32.cti** *or* **Dev32.cti.tp** *for* **Dev.cti** *or* **Dev.cti.tp** *in the example.*

Example 3: **Dev.cti 280,10 -N ser3 &**

The above is an example of a **sysinit** *statement for an Intellicon adapter in a system where COM1 and COM2 are enabled and the Intellicon serial ports have the name prefix of "ser" starting at number 3*

Where:

memory =	*D0000 hex*
I/O port addresses =	*280-283 hex*
IRQ =	*10*
Register as name =	*ser3*

Note: *Other paramters retain their default values*

Example 4: **Dev.cti -p 1 -b 9600 -E -p 4 -b 1200 -e &**

Where:

memory =	*D0000 hex*
I/O port addresses =	*300-303 hex*
IRQ =	*3*
Register as name =	*ser*
Baud rate (ports 1 through 3) =	*9600*
Set options (ports1 through 3) =	*raw mode*
Baud rate (ports 4 through 8) =	*1200*
Set options (ports 4 through 8) =	*edit mode*

Note: *Other paramters retain their default values*

Example 5*:* **Dev.cti -n red 300,10 -n blue 304,15 -a e0000 &**

The above is an example of a **sysinit** *statement for two Intellicon adapters in your system*

Where:

memory=	*E0000 hex*
IRQ (board 1) =	*10*
IRQ (board2) =	*15*
I/O port address (board 1) =	*300-303 hex*
I/O port address (board 2) =	*304-307 hex*
Device prefix (board 1) =	*red*
Device prefix (board 2) =	*blue*

Note: *Other paramters retain their default values*

All examples listed for **Dev.cti** *or* **Dev.cti.tp** *apply to* **Dev32.cti** *and* **Dev32.cti.tp***, but you must substitute* **Dev32.cti** *or* **Dev32.cti.tp** *for* **Dev.cti** *or* **Dev.cti.tp** *in the example.*

Example 6: **Dev.cti 340 -q &**

This statement configures the Intellicon Adapter for polled mode.

Where:

memory =	*d0000*	*(default)*
I/O port addresses	*340-343*	
IRQ	*no IRQ*	*(in polled mode)*

Note: *Other parameters retain their default values*

All examples listed for **Dev.cti** *or* **Dev.cti.tp** *apply to* **Dev32.cti** *and* **Dev32.cti.tp**, *but you must substitute* **Dev32.cti** *or* **Dev32.cti.tp** *for* **Dev.cti** *or* **Dev.cti.tp** *in the example.*

Initialization

When either the **Dev.cti**, **Dev32.cti**, **Dev.cti.tp**, *or* **Dev32.cti.tp** administrator executes, it initializes the hardware then boots the processor on the adapter. This operation is not lengthy but usually takes a second.

Technical Tip:
For QNX Version 4.XX it is advisable that you place a **sleep 5** *command in your* **sysinit** *file following the* **Dev.cti**, **Dev32.cti**, **Dev.cti.tp**, *or* **Dev32.cti.tp** *command. This will allow the administrator time to initialize. If you do not place a* **sleep** *command after these commands,* **stty** *commands executed from the* **sysinit** *file with the adopted serial ports will probably generate* **Invalid Device** *messages because the administrator has not completed its initialization.*

Enhanced Features

When the firmware is executing on the Intellicon board, you may invoke additional data processing and line control features for each individual serial line. These features are usually queried and set by the **ctty** program. Syntax for the **ctty** statement is:

ctty [options]* [>device] [<device]

Where **[options]** are:

eof=*value*	end -of-file character
eol=*value*	end-of-line character
eol2= *value*	alternate end-of-line character
erase= *value*	delete previous character
intr= *value*	generate **SIGINT** character
kill= *value*	delete entire line character
min= *number*	minimum characters required to satisfy raw

	input
quit= ***value***	generate **SIGQUIT** character
susp= ***value***	generate **SIGSTP** character
time= ***number***	timeout value for raw input
start= ***value***	resume output
stop= ***value***	stop output
login= ***value***	generate login character
+/-pcterm	turn on or off PCTerm key scan code processing (see **PCTerm** section below)
+/-rts	turn on or off the Auto RTS feature (see the **Auto RTS** section below).
+/-polarity	turn on or off **+/-polarity** option. With this you can program the "polarity" of the RTS signal in the Auto RTS feature. The default is **+polarity** (see the **Auto RTS** section below).
+/-sync	turns on or off synchronous modes (if the Intellicon adapter supports it)
+/-usrmde	turns on or off user mode support or custom protocols
+/-rxd	turns on or off the receiver discard. This option is for use with RS-485 two wire half duplex communications The default is receiver on (see the **Auto RTS** section below).

PCTerm

Many terminals support a mode of emulation known as PCTerm. In this mode, the terminal transmits an unique code (key scan) for the pressing and release of each key. When **+pcterm** has been set on a port, the Intellicon board will translate these key scan codes into QNX key codes. This allows PCTerm terminals to operate like QNX terminals.

Note:
If you want to use the ***PCTerm*** *option, you must insert the Intellicon QNX 4.XX device driver diskette into floppy drive A and type:*

install -u /dos/a/qnx4/qnx4pct

To translate output to a PCTerm terminal, a termcap entry has been included in the file **pctrmcap**. To inform QNX 4.XX of the Intellicon adapter's PCTerm capabilities, please do the following:

1. Append the file **pctrmcap** to the QNX file **/etc/termcap** with the following command:

   cat pctrmcap >> /etc/termcap

2. Append the file **pcterm** to the QNX file **/usr/lib/terminfo/terminfo.src** with the following command:

   cat pcterm >> /usr/lib/terminfo/terminfo.src

   In order to update the terminfo database, you must then recompile **terminfo.src** by running **tic terminfo.src**.

> ***Note:***
> **termdef** *will not function properly until* **terminfo.src** *is recompiled using* **tic**

> ***Technical Tip:***
> *Sometimes it is not possible to recompile the* **terminfo.src** *file. An alternative is to create the directory* **/usr/lib/terminfo/p** *and then run* **tic pcterm**. *The program* **tic** *will compile the file* **pcterm** *and place it in the* **/usr/lib/terminfo/p** *directory.*

You use the Connect Tech program **ctty** to tell the Intellicon adapter to set the **+/-pcterm** option on the appropriate ports. You then use the QNX program **tinit** to tell QNX which ports are running in PCTerm mode.

Example:
To turn on PCTerm processing for a port, use the following statement:

```
ctty +pcterm </dev/cti1
```

To turn off PCTerm processing for a port, use the following statement:

```
ctty -pcterm </dev/cti1
```

To inform QNX which ports are running in PCTerm mode, use the following statement:

```
tinit -c termdef -T /dev/cti1 &
```

When the program **termdef** *receives data from the port, it will prompt the user for the terminal type. You simply enter* **pcterm** *and the appropriate terminal definition will load from* **/usr/lib/terminfo/p**

Technical Tip:
Occasionally data errors on the line will cause a make or break code to be lost or garbled. When this happens the pcterm firmware may think a key is depressed when it is not. To clear this situation you can use the following script file:

```
ctty -pcterm </dev/$1
stty baud=9600 </dev/$1
ctty +pcterm </dev/$1
stty baud=19200 </dev/$1
```

Note:
Please substitute the correct baud rate you are using for **baud=19200**

Auto RTS

The Auto RTS feature allows you to assert the RTS signal when there is data to be transmitted. After the last stop bit of the last byte has been transmitted, RTS will be dropped allowing other devices access to the switched line. This feature is very useful when doing half duplex or multi-drop RS-485/422 communications.

Example:
To turn on Auto RTS processing for a port, use the following statement:

```
ctty +rts </dev/cti1
```

To turn off Auto RTS processing for a port, use the following statement:

```
ctty -rts </dev/cti1
```

You can program the "polarity" of the RTS signal in the Auto RTS feature with the **+/-polarity** option. With **+polarity** set you will assert the RTS line during the transmission of data. With **-polarity** set you will assert the RTS line when no data is transmitted. The default is **+polarity**.

Example:
To set the RTS line low during data transmission, use the following statement:

```
ctty +rts -polarity </dev/cti1
```

To set the RTS line high during data transmission, use the following statement:

```
ctty +rts +polarity </dev/cti1
```

You can also turn on or off the "receiver discard" of the RTS signal in the Auto RTS feature with the **+/- rxd** option. This option disables the receiver whenever the transmitter is active. This prevents character echo when using half duplex RS-485 communications.

Example:
To turn on the receiver discard option, use the following statement:

```
ctty +rts +rxd </dev/cti1
```

Packet Tigger Mode

Packet Trigger Mode allows you to define a proxy that will be triggered each time a specific end of packet character is seen in the input data stream. You define the end of packet character by the **c_cc[VLEFT]** member of the **termios** structure for the channel. Using this feature may substantially reduce the burden on the host computer.

Note:
Please see the sample program **ptm.c** *for more information concerning Packet Trigger Mode. This program shows how you can read packets from a channel while at the same time "listen" for messages from other processes without polling. You will find this program in the* **/usr/cti** *directory.*

Super Raw Mode

This mode allows your application to minimize the overhead of QNX 4.XX **tti** function calls. Since **Dev.cti** and **Dev32.cti** call these functions for every character received , this overhead is significant in high performance applications.

Super Raw Mode removes this overhead for applications that do not need input processing(or raw data streams). It does this by placing incoming characters directly into the user's allocated buffer.

Example:
Here is an example of the **superraw** *program where any characters received on* **/dev/cti<port #>** *will print to the screen in hexadecimal form:*

```
Superraw < /dev/cti<port #>
```

Note: *You must compile this program with the* **-T 1** *parameter.*

Transparent Print

The Intellicon QNX 4.XX device driver supports a special feature called transparent print. This feature sends printer data to a terminal via the serial link and then re-directs the data to the printer attached to the auxiliary port on the same terminal. This process occurs "transparently" to the user.

Configuration

The Intellicon QNX 4.XX transparent print feature uses two drivers that work together. The first driver **Dev.cti.tp** or **Dev32.cti.tp** runs like the standard **Dev.cti** or **Dev32.cti** driver but with some changes to allow the driver to communicate with a buffer space reserved for transparent print operations on the Intellicon adapter.

The drivers **Dev.cti.tp** or **Dev32.cti.tp** create new device names called **/dev/tp1** through to the number of Intellicon ports on your system(unless **-N** is used to override the name). The command syntax for this statement is:

Dev.cti.tp [options] & (for QNX 4.22 or earler)
or
Dev32.cti.tp [options] & (for QNX 4.23 or later)

Where **[options]** are:

-N name	Prefix to register (default: "**tp**").
-O number	Size of output buffer (default: 1024).
-p number	Set port number for subsequent options.
-f filename	Set file name to use to lookup external terminal defs from.[default **/etc/config/tp.defs**]
-b cps_rate	Effective cps rate to send data to transparent print printer at(used to pace data so that terminal is not bogged down) [default of 100 cps]
terminal	Type of terminal being used, defines the ton/toff sequence

Example 1: **Dev.cti.tp &**

Where: Device names = /dev/tp1 (default)
Transparent print on/off = kimtron (default)
Output buffer size = 1024 bytes (default)

Example 2: **Dev.cti.tp vt100 -N ctip &**

Where: Device names = /dev/ctip1
Transparent print on/off = vt100
Output buffer size = 1024 bytes (default)

Example 3: **Dev.cti.tp -p 1 kimtron -p 4 pcterm &**

Where: Device names = /dev/tp1
Transparent print on/off(ports 1,2,3) = kimtron
Transparent print on/off(ports 4 & up) = pcterm
Output buffer size = 1024 bytes

Note:

1. *While a* **stty -a </dev/tp1** *command will display a lot of the usual* **stty** *parameters, the user should note that since* **/dev/tp1** *shares the serial connection with the* **/dev/cti1** *port, the settings for baud rate, bits etc. do not have any effect.*
2. *All examples listed for* **Dev.cti.tp** *apply to* **Dev32.cti.tp**, *but you must substitute* **Dev32.cti.tp** *for* **Dev.cti.tp** *in the example.*

Since **Dev.cti.tp** and **Dev32.cti.tp** manage the data sent to the transparent print device, it must know the approximate character per second(CPS) rate of the printer and the terminal type(i.e: Wyse50 etc.)

The character per second (cps) rate defaults to 100. This is a reasonable setting but some printers may be faster or slower than that. By setting the cps rate correctly, the user can use the terminal while printing with very little degradation in response.

Different terminal types use different control sequences to turn on or off re-direction of data to an attached printer. If you do not use a terminal supported within the **Dev.cti.tp** or **Dev32.cti.tp** driver, create a new entry in the **/etc/tp.defs** file (this file contains comments on the command syntax necessary). Please refer to **Table 6** for the supported terminal types and their transparent print control sequences:

Table 6: QNX 4.XX transparent print - terminal types

Terminal	TON Sequence	TOFF Sequence
ampex230	\033\140	\033\141
kimtron	\033\	\033a
pcterm	\033`	\033a
volker craig	\033[5i	\033[4i
vt100	\033[5i	\033[4i
wyse50	\033d#	\024
wyse60	\033d#	\024
wyse99	\033d#	\024
wyse150	\033d#	\024

Note: *you do not need the* **/etc/tp.defs** *file if the terminal type is one supported by* **Dev.cti.tp** *or* **Dev32.cti.tp**.

tp_ctrl

You can set or query the transparent print settings of ports after a driver is started by using the **tp_ctrl** command (stands for Transparent Print ConTRoL). The command syntax for this statement is:

tp_ctrl [options] &

Where **[options]** are:

-f filename	Set the file name to use to lookup external terminal definitions. (default **/etc/config/tp.defs**)
-b cps_rate	Effective cps rate to send data to transparent print printer at(used to pace data so that terminal is not bogged down) (default of 100 cps)
-q	Query the port settings
terminal	Type of terminal used (defines the ton/toff sequence)

Example 1: **tp_ctrl vt100 </dev/cti3**

Set Intellicon port 3 to vt100 terminal type

Example 2: **tp_ctrl -q </dev/cti1**

Query the settings for Intellicon port 1

Utilities

CHMON

CHMON (CHannel MONitor) is a software utility that monitors all Intellicon serial ports. It obtains information about ports from the driver every second and only updates this information when the state of a port changes since the last scan. You can also run **CHMON** from a remote computer.

For more information on the use of **CHMON**, please refer to the readme file **CHMON.DOC** on the Intellicon QNX 4.XX driver diskette.

cti_tcdrain

We provide the function **cti_tcdrain()** because the **tcdrain()** function supplied by QSSL does not work with Intellicon 4.XX device drivers.

The reason that **tcdrain()** does not work is that it expects the UART device to have a transmit FIFO of one byte. Therefore when the buffer between **Dev** and **Dev.cti/Dev32.cti** is empty, **tcdrain()** assumes the transmission of all bytes of data.

However, it is possible that the Intellicon adapter can have several kilobytes of data in its buffers and **tcdrain()** will indicate that all data has transmitted. To avoid this situation **cti_tcdrain()** will operate like **tcdrain()** except that it will not return until all the data in the buffers has transmitted.

You use **cti_tcdrain()** in conjunction with **tcdrain()**, thus ensuring that the operating system's buffers and the Intellicon buffers are empty.

Example:
The following is an example of **cti_tcdrain()**:

```
int             fd;
fd = open("/dev/nt1", O_RDWR);
write( fd, "ATH");
tcdrain( fd );
cti_tcdrain( fd );
```

Note:
When **tcdrain()** *returns, Dev's buffers have emptied to the Intellicon adapter. When* **cti_tcdrain()** *returns, all data buffered on the Intellicon adapter has been transmitted.*

ctiflush

We provide the function **ctiflush()** because the **tcflush()** function supplied by QSSL does not work with Intellicon QNX 4.XX device drivers.

The reason that **tcflush()** does not work is that it expects the UART device to have a transmit or receive FIFO of one byte. Therefore when the buffer between **Dev** and **Dev.cti** or **Dev32.cti** is empty, **tcflush()** assumes that all input/output buffers are clear.

However, it is possible that the Intellicon can have several kilobytes of data in its buffers and **tcflush()** will indicate that all buffers are clear. To avoid this situation **ctiflush()** will operate like **tcflush()** except that it will not return until all the data in the buffers has cleared.

ctiflush() is a replacement for **tcflush().** When you use **ctiflush()** it calls the function **tcflush()** as well , thus ensuring that the operating system's buffers and the Intellicon buffers are empty.

For more information on the use of **ctiflush**, please refer to the **README** file on the Intellicon QNX 4.XX driver diskette.

Example:
The following is an example of **ctiflush()**:

```
int             fd;
fd = open("/dev/cti1", O_RDWR);
write( fd, "ATH");
ctiflush( fd, TCIOFLUSH );
```

Note:
When **ctiflush()** *returns, the Intellicon's buffers and Dev's buffers are cleared.*

Error Messages

Dev.cti: WARNING: No interrupt received from card *n* on IRQ *nn*
During startup of the driver an interrupt was not sensed from board number ***n*** on the specified Interrupt line ***nn***. Check your Interrupt settings for conflicts with other boards and make sure you have set the IRQ correctly.

Dev.cti: No response from card *n*
The Intellicon is not responding to commands in a timely fashion on board ***n***. This can be due to serial lines being incorrectly connected or intermittent memory conflicts.

Dev.cti: Cannot verify code into Intellicon
The firmware being written to the card is not being read back properly. Since a memory test is run before this it is likely a hardware failure of some sort.

Dev.cti: No devices available.
All devices slots are in use under QNX 4.xx. Try changing your **sysinit** so that the **Dev** line has a "**-n 40**" or bigger on it.

Dev.cti: System unable to add new devices.
The device names that were specified to be used by **Dev.cti** already exist. This is usually caused by some abnormal termination of our **Dev.cti** driver, where the **/dev** names did not get cleaned up.

Intellicon Memory Test failed at *mmmmm*(hex) on board using IO Port *nnn*(hex)
Check for Memory Conflicts
Contents of this address are:
|------ H E X -------|--- A S C I I ----|
The startup memory test failed at address ***mmmmm*** on the board that uses io port ***nnn***. Ensure your memory segment and IO port has been set properly both on the board and on the command line (if set wrong this test will usually fail near the start of the segment). If the board is set correctly this error usually means there is some sort of memory conflict. Check the settings on other adapter boards such as Networks etc... The **HEX - A S C I I** output may give you a clue what it is in conflict with.

Notes:

1. *All error messages listed for* **Dev.cti** *apply to* **Dev32.cti** *as well.*

2. *When contacting Connect Tech Customer Support, please have the following information available:*

- *sysinit.# file*
- *output from sin ar*
- *output from sin ve*
- *any relevant script files (ie stty commands)*
- *Dev.cti or Dev32.cti command line*
- *Intellicon adapter serial number*
- *a list of devices connected to the Intellicon adapter*

UNIX Device Driver

The Intellicon UNIX device driver provides an interface between the UNIX operating system(and derivatives) and an Intellicon adapter.

Under UNIX, and its derivatives, the Intellicon UNIX driver will link into the kernel to generate a new run time system. The diskettes include installation programs that will configure the driver, link it into the kernel and create the special device files.

> ***Note:***
> *The diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver software utilities*

Before proceeding with the installation, determine which type of UNIX you have and locate the appropriate Intellicon installation diskette for that type of UNIX. To determine the contents of each diskette, carefully check each label and any addendum included with this manual.

Installation

Once you have matched the installation diskette with your particular brand of UNIX, examine the installation diskette and note the installation command found on the label

1. Login as root and change directories to the root directory. (i.e.: **cd /**).
2. Insert the installation diskette and execute the installation command. Refer to the section **OS Specific Information** for detailed installation instructions.

3. The program **cti8boot** automatically loads into the system startup procedure to initialize the Intellicon boards. It takes as options the parameters chosen in the installation procedure. The command syntax is:

cti8boot [options]

Where **[options]** are:

a=mem_seg	Shared memory segment. Default: **a=d0000** (hex)
i=irq	Interrupt request line. There must be one for each board.
p=port	Port address of the board. There must be one for each board.
/ or ,	Indicates the next board. Parameters for multiple boards are separated with either a **,** or a **/**.

4. You will be asked how many boards you are installing, the total number of ports, and whether to install transparent print software. The number of boards being installed is used to look up the default I/O and IRQ settings for each board. The typical defaults under UNIX are shown in **Table.7**

Table 7*: UNIX IRQ & port address defaults*

Board#	Port	IRQ
1	300	10
2	304	11
3	308	12
4	30C	14

Note: *Intellicon boards are set to IRQ 3 at the factory and will require change. As well, the default memory address D0000 hex, the default tty starting name and the default major number are looked up if appropriate. You will be asked to confirm the defaults and your selection before the installation will complete. If you do not accept the defaults, the installation program will provide details on changing them.*

Vectors/IRQ

Within UNIX, IRQ lines correspond to interrupt vectors. **Table 8** provides the current translation of IRQ lines to UNIX vector numbers.

Table 8: *UNIX IRQ/Vector translation*

Hardware IRQ	XENIX Vector	UNIX Vector
2	25	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
10	26	10
11	27	11
12	28	12
14	30	14
15	31	15

Device File Names

You may install a maximum of 32 Intellicon ports in your computer under UNIX. UNIX requires unique device file names for each one of these ports. The syntax for naming the ports of the Intellicon adapter is:

/dev/ttyXz

Where:

X The offset for ports. When using Intellicon adapters in a system with standard serial ports, it is necessary to enter an offset where the Intellicon ports will begin.

> ***Note:****The default offset is 3 for SCO XENIX and 4 for SCO UNIX.*

z is the port (z ranges from a - af)

> ***Note:*** *The port designation(z) in device file names is always lower case*

Example:
The file names for the devices connected to the Intellicon adapter when there is one Intellicon-8 adapter and 2 standard serial ports(COM1 & COM2) in your system are:

Intellicon, port #1	**tty3a**
Intellicon, port #2	**tty3b**
Intellicon, port #3	**tty3c**
...and so on up to:	
Intellicon, port #8	**tty3h**

Note:
The port designation(z) in device file names is always lower case.

Example:
The names for the devices connected to the Intellicon adapters when there is two Intellicon-8 adapters and two standard serial ports (COM 1 & COM2) in your system are:

Intellicon #1, port #1	**tty3a**
Intellicon #1, port #2	**tty3b**
Intellicon #1, port #3	**tty3c**
...and so on up to:	
Intellicon #2, port #14	**tty3n**
Intellicon #2, port #15	**tty3o**
Intellicon #2, port #16	**tty3p**

Note:
The port designation(z) in special device file names is always lower case.

Example:
The names for the devices connected to the Intellicon adapters when there is four Intellicon-8 adapters and two standard serial ports (COM 1 & COM2) in your system are:

Intellicon #1, port #1	**tty3a**
Intellicon #1, port #2	**tty3b**
Intellicon #1, port #3	**tty3c**

...and so on up to:

Intellicon #4, port #30	**tty3ad**
Intellicon #4, port #31	**tty3ae**
Intellicon #4, port #32	**tty3af**

Note:
The port designation(z) in special device file names is always lower case.

Modems

Along with the device file names mentioned previously, it is necessary to create other device file names as well. If you connect an ACU (Automatic Calling Unit), more commonly referred to as a modem, to the port, then the device file name for the ACU has the following form:

/dev/ttyXZ

Where:

X The offset for ports. When using Intellicon adapters in a system with standard serial ports, it is necessary to enter an offset where the Intellicon ports will begin.

> ***Note:*** *The default offset is 3 for SCO XENIX and 4 for SCO UNIX .*

Z is the port (Z ranges from A - AF)

> ***Note:*** *The port designation(Z) in modem device file names is always upper case.*

Example:
The device file names for modems connected to the Intellicon adapter when there is one Intellicon-8 adapter and 2 standard serial ports(COM1 & COM2) in your system are:

Intellicon, port #1	**tty3A**
Intellicon, port #2	**tty3B**
Intellicon, port #3	**tty3C**
...and so on up to:	
Intellicon, port #8	**tty3H**

Note:
The port designation(Z) in modem device file names is always upper case.

Example:
The device file names for modems connected to the Intellicon adapters when there is two Intellicon-8 adapters and two standard serial ports (COM 1 & COM2) in your system are:

Intellicon #1, port #1	**tty3A**
Intellicon #1, port #2	**tty3B**
Intellicon #1, port #3	**tty3C**
...and so on up to:	
Intellicon #2, port #14	**tty3N**
Intellicon #2, port #15	**tty3O**
Intellicon #2, port #16	**tty3P**

Note:
The port designation(Z) in modem device file names is always upper case.

Example:
The device file names for modems connected to the Intellicon adapters when there is four Intellicon-8 adapters and two standard serial ports (COM 1 & COM2) in your system are:

Intellicon #1, port #1	**tty3A**
Intellicon #1, port #2	**tty3B**
Intellicon #1, port #3	**tty3C**

...and so on up to:

Intellicon #4, port #30	**tty3AD**
Intellicon #4, port #31	**tty3AE**
Intellicon #4, port #32	**tty3AF**

Note:
The port designation(Z) in modem device file names is always upper case.

Printers

If you choose the Transparent Print option during installation, the install script creates a special device file name for the appropriate port. This device name is the standard device name with the letter **p** appended to the name. Refer to the **Transparent Print** section for more details.

Enhanced Features

Transparent Print

Some terminals(i.e.: Wyse 50 or 60) support a feature called transparent print. This means that when a program sends a specific sequence of characters to the terminal's modem port, the terminal will route all received data to its auxiliary port until another specific character sequence turns it off.

When you install the Intellicon device driver with this feature, the installation program creates a special device file name for the appropriate port. This device name is the standard device name with the letter **p** appended to the name. The names for transparent print devices have the form:

/dev/ttyXzp

Where:

- **X** is the offset number
- **z** is the port on the ACM (z ranges from a - p)
- **p** designates the terminal as a transparent print device

> ***Note:*** *The transparent print designation(**p**) in special device file names is always lower case.*

Example:
The device file name of a transparent print device on port #1 of an Intellicon adapter in a system with two standard serial ports is:

tty3ap

The device file name of a transparent print device on port #11 of an Intellicon adapter in a system with two standard serial ports is:

tty3kp

The device file name of a transparent print device on port #32 of an Intellicon adapter in a system with two standard serial ports is:

tty3afp

Note:
*The transparent print designation(**p**) in special device file names is always lower case.*

tp_ctrl

The Intellicon **tp_ctrl** utility allows you to declare the transparent print on and off character sequences for each port. You should add these declarations to the rc file in order to initialize the Intellicon board at start up. The syntax of this function is:

tp_ctrl device_name[option]

Where **[option]** is either:

-term terminal_type The **-term** option allows the user to select one of the terminal types listed in the **/etc/tp_terms** library file. You can edit this file to provide definitions for other terminal personalities. Currently, there are definitions for wyse60, wyse50 and ampex230.

-ton on_sequence
-toff off_sequence The **-ton** and **-toff** options allow the setting of the on and off character strings respectively. You separate these strings by commas (,) and can be in the following formats:

- hexadecimal (i.e. 0x31, 0x42)
- character (i.e. '1', 'B')
- octal (i.e. \061, \102)
- decimal (i.e. 49, 66)

When editing the **tp_terms** file, you should enter the on and off sequences as shown above. However when entering these from the command line, often an extra **** must precede certain special characters (i.e. **'**). This is because the UNIX shell consumes and interprets them.

Example:
For example the following statement is ***incorrect:***

tp_ctrl /dev/tty3a -ton 'd',\066 -toff 0x45

The following is the ***correct*** *way of representing the above:*

tp_ctrl /dev/tty3a -ton \'d\',\\066 -toff 0x45

Example:
You can add the same on and off sequence to the **tp_terms** *file as a definition called* **myterm** *by appending the following to it:*

```
myterm -ton 'd',\066 -toff 0x45
```

Note:
You do not require preceding backslashes since shell will not interpret them.

Note:
It is necessary to remove the **ixany** *option from the terminal device. You can do this with the following command:*

```
stty -ixany </dev/tty??
```

Technical Tip: Keyboard Delay Under Transparent Print
When the printer attached to the terminal is printing a file, the keyboard of the terminal may become hampered. In other words, the terminal may ignore keystrokes because it is too busy sending data to the printer. It may be necessary to add delays to the print stream to avoid keyboard scanning problems by using the **stty** *option* **nl1**. *However, this option cannot be set from the command line, but must be set within the printer's interface file, because options in the interface file override current port settings.*

In Unix, there is a directory called **/usr/spool/lp/admins/lp/interfaces** *which contains an interface file for each printer defined in the system. These interface files contain information about the printer setup including a* **stty** *command for the port settings. Add the* **nl1** *option to the* **stty** *command line within this file. If* **nl0** *is currently present in the* **stty** *command line, remove it.*

The delays added by the **nl1 stty** *option reduces the number of characters per second in the print stream. Since the terminal does not have to process as many characters per second while the printer is printing, it can properly scan the keyboard for keystrokes.*

Error Messages

Fatal Error Messages

CTI8BOOT: Cannot open /dev/cti_tty for initialization.
The special device file cannot be opened to initialize the Intellicon. Probably the wrong kernel was booted.

CTI8BOOT: Unknown option.
An unknown option was specified on the command line. Edit the rc file.

CTI8BOOT: Memory Segment Incorrect.
You have specified a memory address for the Intellicon which is not physically possible.

CTI8BOOT: Only n boards allowed.
The options on the command line have indicated more than 16 boards are installed.

DFUN: function n failed.
An IOCTL function failed during boot. Probably the wrong kernel was booted.

Warning Error Messages

CTI: Unable to load code at xxxx.
IO or Memory Address of board improperly specified or memory address conflict.

CTI: WARNING: No interrupt from card n on vector nn.
IRQ conflict or mis-specification.

CTI: No response from card n.
Possible hardware failure or conflict.

OS Specific Information

SCO UNIX AT386

Under SCO UNIX, you install the Intellicon SCO UNIX driver by using **custom**. To install the SCO UNIX device driver follow these procedures:

1. Login as **root** and change directories to the root directory (i.e. **cd /**).
2. Insert the Intellicon UNIX driver diskette that contains the SCO UNIX driver and type **custom.**
3. Once you are at custom's main menu, make the following menu choices:

   Install
   ⇩
   A New Product
   ⇩
   Packages

4. The installation program will ask you to insert **Distribution Floppy Volume 1**. Insert the Intellicon UNIX driver diskette that contains the SCO UNIX driver, if you have not done so and select **CONTINUE**.
5. The installation program gives you a choice of packages to install. Select **INTELLICON FOR SCO UNIX.** Select **CONTINUE** once more.
6. At this point, the installation program will ask you how many boards and how many ports you wish to install. Enter the appropriate numbers.
7. The installation program displays a list of common parameters and asks you if these are suitable. If they are not, you can make the appropriate changes now.
8. When you finish making your choices, select **YES**. The installation process will proceed and build a new kernel.

9. Once the kernel rebuilds, UNIX will prompt you **"Do you wish the new kernel to boot by default?"** and **"Do you wish the kernel environment to be rebuilt?"**. Answer **YES** to both. Your current kernel will backed to **/unix.old**. In the event that your system does not reboot, perhaps due to a hardware conflict, you can bring up your system by booting **unix.old**.

10. You can now select **QUIT** to leave custom. Do **shutdown** first before you reboot the new kernel.

11. Once your system reboots, you must run the system administrative shell to create entries in the tty database. You do this by executing **sysadmsh** and selecting the following menus:

ACCOUNTS
⇩
TERMINAL
⇩
CREATE

12. You must then enter all your ttys.

SCO UNIX Open Server

Under SCO UNIX Openserver, you install the Intellicon SCO UNIX driver by using the new **custom** installation utility. To install the SCO UNIX device driver follow these procedures:

1. Login as **root** and change directories to the root directory (i.e. **cd /**).
2. Insert the Intellicon UNIX driver diskette that contains the SCO UNIX driver and type **custom.**
3. Once you are at custom's main menu, make the following menu choices:

Software
⇩
Install New
⇩
From (your machine name)
⇩
Select Your Floppy Disk Drive
⇩
Partial

4. The installation program gives you a choice of packages to install. Select **INTELLICON DEVICE DRIVER VER X.XX** . Hit the **ENTER** key
5. A second reference to the **INTELLICON DEVICE DRIVER VER X.XX** will come up below the first one. Please select it by pressing the **ENTER** key
6. A list of items will appear some of which are drivers for different versions of SCO UNIX/XENIX. Use the arrows and select the appropriate driver for your installation (i.e., **Intellicon for SCO UNIX**). Hit the **ENTER** key.
7. Use the **TAB** key to select the **INSTALL** option and hit the **ENTER** key.
8. At this point, the installation program will ask you how many boards and how many ports you wish to install. Enter the appropriate numbers.
9. The installation program displays a list of common parameters and asks you if these are suitable. If they are not, you can make the appropriate changes now.
10. When you finish making your choices, select **YES**. The installation process will proceed and build a new kernel.

11. Once the kernel rebuilds, UNIX will prompt you **"Do you wish the new kernel to boot by default?"** and **"Do you wish the kernel environment to be rebuilt?"**. Answer **YES** to both. Your current kernel is copied to **/unix.old**. In the event that your system does not reboot, perhaps due to a hardware conflict, you can bring up your system by booting **unix.old**.

12. You can now select **QUIT** to leave **custom**. Do **shutdown** first before you reboot the new kernel

13. You must then enter all your ttys.

SCO XENIX AT386

Under SCO XENIX, you install the Intellicon SCO XENIX driver by using **custom**. To install the SCO XENIX device driver follow these procedures:

1. Login as **root** and change directories to the root directory (i.e. **cd /**).
2. Insert the installation diskette and type **custom**.
3. Once you are at custom's main menu, make the following selection:

   4. ADD A SUPPORTED PRODUCT

4. The installation program prompts you to insert **DISTRIBUTION VOLUME 1**. Insert the Intellicon UNIX driver diskette that contains the SCO XENIX driver and press **RETURN**.
5. The installation program gives you another list of options. From this list, select:

   1. INSTALL ONE OR MORE PACKAGES

6. The installation program displays a list of packages. Select **INTELLICON FOR SCO XENIX 386 .**
7. The installation program prompts you to insert the driver diskette. Insert the Intellicon UNIX driver diskette that contains the SCO XENIX driver, if you have not done so already, and press **RETURN**.
8. At this point, the installation program will ask you how many boards and how many ports you wish to install. Enter the appropriate numbers.
9. The installation program displays a list of common parameters and asks you if these are suitable. If they are not, you can make the appropriate changes now.
10. When you finish making your choices select **YES**. The installation process will proceed and build a new kernel.
11. When the installation process is complete, there is a new kernel called **/xenix.new** containing the Intellicon driver. To access this new kernel, you must do **shutdown** first, reboot your system and type **xenix.new** at the boot prompt. Once you test the new kernel, you can then make a backup copy of your old kernel, then move **/xenix.new** to **/xenix**

INTERACTIVE UNIX AT386

You install the Intellicon INTERACTIVE UNIX driver by using **sysadm.** To install the device driver follow these procedures:

1. Login as **root** and change directories to the root directory (i.e. **cd /**).
2. Insert the installation diskette and type **sysadm**.
3. Once you are at the **sysadm** main menu, make the following choices below:

   SOFTWARE
   ⇩
   INSTALL A PACKAGE
   ⇩
   SELECT INSTALLATION MEDIA TYPE

4. The installation program will ask you to insert **DISTRIBUTION FLOPPY VOLUME 1**. Insert the Intellicon UNIX driver diskette that contains the INTERACTIVE UNIX driver, if you have not done so and select **YES** to continue.

4. The installation program gives you a choice of packages to install. Press the space bar to select:

   1) INSTALL INTELLICON DEVICE DRIVER.

6. At this point, the installation program will ask you how many boards and how many ports you wish to install. Enter the appropriate numbers.
7. The installation program displays a list of common parameters and asks you if these are suitable. If they are not, you can make the appropriate changes now.
8. When you finish making your choices, select **YES**. The installation process will proceed and build a new kernel.
9. Once the new kernel rebuilds, the system will automatically do **shutdown**. Reboot the system to access the Intellicon device drivers.

Unixware v.1.1.X

Under Unixware V.1.1.X, you install the Intellicon Unixware V.1.1.X driver by using **custom**. To install the Unixware V.1.1.X device driver follow these procedures:

1. Login as **root** and change directories to the root directory (i.e. **cd /**).
2. Insert the Intellicon UNIX driver diskette that contains the Unixware V.1.1.X driver and type **installpkg.**
3. The installation program will ask you to insert the distribution diskette. Insert the Intellicon UNIX driver diskette that contains the Unixware V.1.1.X driver, if you have not done so and hit the enter key.
4. At this point, the installation program will ask you how many boards and how many ports you wish to install. Enter the appropriate numbers.
5. The installation program displays a list of common parameters and asks you if these are suitable. If they are not, you can make the appropriate changes now.
6. When you finish making your choices, select **YES**. The installation process will proceed and build a new kernel.
7. Once the kernel rebuilds, you must reboot the your system in order to access the Intellicon ports.

Windows 3.XX

The Intellicon Windows 3.XX device driver provides an interface between the Windows 3.XX operating environment and an Intellicon adapter. Under Windows 3.XX you can install a maximum of 9 ports using one Intellicon adapter.

Installation

To install the Intellicon Windows 3.XX device driver follow these procedures:

> ***Technical Tip***
> *Although the installation process provides a large amount of flexibility, it is usually best to select the default options by hitting* **Enter** *in response to the install questions.*

> ***Technical Tip:***
> *Memory management software such as EMM386™, 386MAX™ or QEMM™, must not manage the shared memory region used by your Intellicon adapter. These software packages usually have an "exclude" command to prevent the use of a specified area of memory. Here are a few examples of QEMM™, 386MAX™ and EMM386™ statements using exclude commands:*
>
> **device= c:\qemm\qemm.sys exclude=d000-dfff**
> **device= c:\386max\386max.sys exclude=d000-dfff**
> **device= c:\windows\emm386.exe x=d000-dfff**

> ***Note:***
> *The diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver.*

1. Insert the Intellicon Windows 3.1 device driver disk into your floppy drive.
2. Open this drive using **FILE MANAGER** and double click on **setup.exe** **OR** select **RUN** from the **FILE** menu and type **a:setup.exe** (assuming **a:** is the appropriate disk drive).
3. A screen will appear that states the driver type and version number. Select **OK** for this screen and the other screens that follow.

4. Eventually a screen will appear that asks for the destination drive letter. Select one of the drives and hit **OK**.

5. Choose a directory to install the utilities to and hit **OK**.

6. Verify that the Windows directory displayed is correct and hit **OK**.

7. Now the installation will begin. A progress meter will appear and after awhile the installer will return and asks whether you wish to call the directory **CTI UTILS**. Either choose another name or hit **OK** for the default.

8. Now the configuration program will start. This program will set the necessary parameters to run all the serial ports in your machine. You should ensure that you list your standard serial ports in order from top to bottom, then list any DFlex boards and lastly list the Intellicon board.

   Note: *The Intellicon board must be last on the list.*

   8.1. To add a board hit the **NEW BOARD** button. A board will appear on the list **AFTER** the currently high-lighted entry in the **INSTALLED BOARDS** list.

   8.2. To change the hardware settings of a board, select the board from the **INSTALLED BOARDS** list and the parameters to change in the **HARDWARE SETTINGS** group.

   Notes
   1. *The* **DRIVER VERSION INFORMATION** *box will likely display* **Not Currently Running** *since the driver is not loaded yet.*
   2. *You can usually ignore the* **CONFIGURATION** *button. You use this button to setup serial printer baud rates etc.*

   8.3. When all boards are setup, hit **OK** to proceed to the next step.

9. You are now asked a series of questions. Respond with **OK** and **YES** to all of them. When your computer starts to reboot, shut it off and install the Intellicon board (if not previously installed). Make sure the Intellicon settings are the same as the ones installed for Windows. Turn your computer on when this is done.

10. Enter Windows and run the Connect Tech Control Panel application. If you see the driver's version number in the **Driver Version Information** box, then you have installed the Intellicon adapter and software correctly. The Intellicon's serial ports will now run like standard serial ports for Windows applications.

Uninstallation

To uninstall the Intellicon Windows 3.XX device driver follow these procedures:

1. From DOS, **CD** to the main Windows directory and open the **SYSTEM.INI** file with **EDIT**.

2. In the **[boot]** section locate the **comm.drv=** statement. You can change the driver to the standard Windows serial driver by changing the statement to:

comm.drv=comm.drv

If you installed a serial port driver previous to the Intellicon driver, copy the driver listed in the **OldCOMM=** statement in the **[ConnectTech]** section to the **comm.drv=** statement in the **[boot]** section.

3. In the **[386enh]** section find the **device=ctivcd.386** statement. Right above or below this statement you will see the old line that looks something like **;device=*vcd**. Remove the semi-colon from this line and delete the **device=ctivcd.386** statement.

4. The next time you start Windows, the system will not recognize the Intellicon drivers.

Technical Tip:
After you have uninstalled the Intellicon driver you can remove the exclude option (i.e., **X=D000-DFFF***) from the* **EMM386** *statement in your* **config.sys** *file. This will free up this section of high memory for DOS to use.*

Windows NT

The Intellicon Windows NT device driver provides an interface between the Windows NT operating environment and an Intellicon adapter. Under Windows NT you can install a maximum of 256 ports.

Installation

To install the Intellicon Windows NT device driver follow these procedures:

> ***Technical Tip***
> *Although the installation process provides a large amount of flexibility, it is usually best to select the default options by hitting* **Enter** *in response to the install questions.*

> ***Note:***
> *The diskettes may also include README files. Please examine these files for technical tips or release notes concerning installation and configuration of the device driver.*

1. Copy the Intellicon driver files to a temp directory or to a floppy disk. Ensure that the files all remain together.
2. Run the Connect Tech installation utility **setup.exe**. This will uncompress and copy the driver files to the appropriate locations.
3. The ctisetup CPL applet will automatically launch now. You can enter the Intellicon hardware parameters. For example:

Shared Memory segment	**d0000**
I/O port address	**300**
Hardware Interrupt (IRQ)	**10**

   > ***Technical Tip:*** *The above parameters are recommended.*

4. Click on the **EXIT** button and the setup program will **either** start the Intellicon driver (Windows NT v.3.51) or inform you to start the driver with the **CONTROL PANEL** and **DEVICES CPL** applet or a reboot (Windows NT v.4.00)
5. You can now view the **read.me** file included with driver.

6. Installation is complete. Now you can access the additional Intellicon serial ports.

> ***Note:***
> *The driver installs with the* **AUTOMATIC** *setting. This means it will start each time the system reboots.*

Enhanced Features

You can order various models of Intellicon adapters that support the different electrical interfaces. The Intellicon-2/-4 RS-485; Intellicon-2+2 and the Intellicon-Flex8 offer features such as multi-drop and half duplex RS-485. You can configure these features and more via the **Control Panel** applet **ctisetup.cpl**. The currently supported features are:

Multi-drop RS-485: Full Duplex

This feature enables or tri-states the transmitter when your application uses the RTS signal from the UART. The transmitter tri-states when transmission is complete, allowing multiple transmitters to use the RS-485 bus at the same time.

Multi-drop RS-485: Half Duplex

This feature disables the receiver whenever the transmitter is active. This prevents character echo when using half duplex RS-485.

Timer Enable/Disable

This feature enables and disables the receiver timer. This can minimize receiver latency. You must use this feature with discretion.

RS-422 Support

The Intellicon driver supports the RS-422 electrical interface **without** the above features.

> ***Note:***
> *Please see examples in the file* **winnt_tidbits.c** *if you want to use these settings in an application. You will find this file on the Intellicon Windows NT driver diskette.*

Technical Tips:

1. *The Intellicon is not a plug and play(PnP) adapter. The Intellicon requires an IRQ, I/O port addresses and a 64 KB segment of shared memory in the region between 640 KB and 1 MB. It is often necessary to go into the CMOS setup (BIOS) and ensure that the resources used by the Intellicon are available to Non-PnP ISA devices. Look in the CMOS setup for PnP or ISA setup options.*
2. *You may have a hardware conflict if you experience erratic operation of the Intellicon adapter or some other I/O card in the system (SCSI adapter, network adapter, etc.). You may need to alter the settings of the adapters to resolve the conflict.*
3. *Ensure that you disable Shadow RAM in the memory region where the Intellicon resides. You can do this in your CMOS setup.*
4. *The compressed Intellicon driver file includes the memory test utility* **ctimtest.exe**. *This is a DOS program that you can use to verify that the Intellicon memory is available for use. Please see the file* **ctimtest.zip** *for more information. You should run this program with DOS 5.0 or better. This program will give erroneous results under Windows NT.*

Chapter 4: Appendices

Appendix A: Specifications

Intellicon-2/-4/-8/-6+2/Flex8

Processor

- Zilog Z80B processor, 5 MHz

Memory

- 64K static RAM

PC Bus Interface

- Dual-ported RAM interface
- Memory appears as 64K byte window in PC address space
- 8 bit transfers
- PC interrupt is jumper selectable, (3, 4, 5, 6, 7, 9, 10, 11, 12, 14, or 15)
- Base address for I/O ports is DIP switch selectable, requires 4 addresses
- Memory starting address is DIP switch selectable.

Communications

- Zilog Z80 DARTs communication controllers
- Programmable baud rate generator - up to 38.4K baud on all ports

Control Signals

- RS-232 RTS; DTR; TxD; DCD; CTS; RxD
- RS-485 RTS; TxD; CTS; RxD
- RS-423 RTS; TxD; CTS; RxD
- 20mA Current Loop TxD; RxD

Operating Environment

- Ambient temperature: 5 - 60° C
- Relative humidity: 5 - 90% no condensing
- Air movement: no requirement
- Altitude: 15,000 feet (5000 metres)
- UL Flame Rating: UL 94V-0

Power Requirements

- *Intellicon-2 (RS-232)*

+5 VDC	+/- 5%	@ 0.660 A. typ.	1.200 A. max.
+12 VDC	+/- 10%	@ 0.038 A. typ.	0.050 A. max.
-12 VDC	+/- 10%	@ 0.038 A. typ.	0.046 A. max.

- *Intellicon-2 (RS-485)*

+5 VDC	+/- 5%	@ 0.700 A. typ.	1.200 A. max.

- *Intellicon-4 (RS-232)*

+5 VDC	+/- 5%	@ 0.860 A. typ.	1.500 A. max.
+12 VDC	+/- 10%	@ 0.076 A. typ.	0.100 A. max.
-12 VDC	+/- 10%	@ 0.072 A. typ.	0.092 A. max.

- *Intellicon-8 (RS-232)*

+5 VDC	+/- 5%	@ 1.100 A. typ.	2.000 A. max.
+12 VDC	+/- 10%	@ 0.110 A. typ.	0.150 A. max.
-12 VDC	+/- 10%	@ 0.110 A. typ.	0.140 A. max.

- *Intellicon-6+2*

+5 VDC	+/- 5%	@ 1.000 A. typ.	1.600 A. max.
+12 VDC	+/- 10%	@ 1.000 A. typ.	0.100 A. max.
-12 VDC	+/- 10%	@ 1.000 A. typ.	0.030 A. max.

- *Intellicon-Flex8 (without SLIMs)*

+5 VDC	+/- 5%	@ 2.000 A. typ.	1.600 A. max.
+12 VDC	+/- 10%	@ 0.000 A. typ.	0.000 A. max.
-12 VDC	+/- 10%	@ 0.000 A. typ.	0.000 A. max.

- *RS-232 SLIM (no load)*

+5 VDC	+/- 5%	@ 30 mA. typ.
+12 VDC	+/- 5%	@ 1 mA. typ.

- *RS-485 SLIM (no load)*

+5 VDC	+/- 5%	@ 120 mA. typ.

- *RS-423 SLIM (no load)*

+5 VDC	+/- 5%	@ 70 mA. typ.
+12 VDC	+/- 10%	@ 26 mA. typ.
-12 VDC	+/- 10%	@ 26 mA. typ.

- *Current loop SLIM (no load)*

+5 VDC	+/- 5%	@ 30 mA. typ.
+12 VDC	+/- 5%	@ 4 mA. typ.

Dimensions

- *Intellicon-2/-4*
  Length: 27.78 cm
  Height: 10.60 cm
- *Intellicon-8/-6+2/-Flex8*
  Length: 33.60 cm
  Height: 10.60 cm

Connectors/Interface

Intellicon-2

- RS-232 9-pin male
- RS-485 9-pin female

Intellicon-4

- RS-232 25-pin male configured as DTE (you may optionally order a cable harness as male or female configured as DTE or DCE)
- RS-485 9-pin male
- RS-485 9-pin female (Intellicon-2+2)

Intellicon-6+2

- RS-232 25-pin male
- RS-485 9-pin male

Intellicon-8

- RS-232 25-pin male, configured as DTE (you may optionally order a cable harness as male or female configured as DTE or DCE)

Intellicon-Flex8

- RS-232 9-pin male
- RS-485 9-pin male
- RS-423 9-pin male
- 20mA current loop 9-pin male

I/O Connector Box (Intellicon-8)

- RS-232 25-pin female, configured as DCE and RJ-45

I/O Connector Box (Intellicon-Flex8)

- RS-232 9-pin male
- RS-485 9-pin male
- RS-423 9-pin male
- 20mA current loop 9-pin male

Certification

Intellicon-2/-4/-6+2/-8/-Flex8

FCC

This equipment complies with the requirements in Part 15 of FCC Rules for a Class A computing device. Operation of this equipment in a residential area may cause unacceptable interference to radio and TV reception requiring the operator to take whatever steps are necessary to correct the interference.

DOC/IC

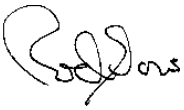
This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe est conforme à la norme NMB-003 du Canada.

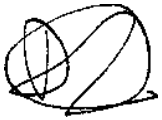
CE

Connect Tech Inc. declares that the product(s) covered by the contents of this manual have been tested and found compliant with the below listed standards as required by the Electromagnetic Compatibility (EMC) Directive for General Immunity Compliance, EN 50 0082.1:1997

EN 55022	Conducted and Radiated emissions
IEC 61000-4-2	Electrostatic Discharge
IEC 61000-4-3	Radiated Immunity
IEC 61000-4-4	Electrical Fast Transients



Rod Doré
Manager of Engineering



Dave Worthen
President

General

The above agency conformances were met by independent laboratory testing of Connect Tech Inc. product(s) with shielded cables, with metal hoods, attached to either the terminating connectors or cable assemblies supplied with the product(s). **Failure to follow good EMC/EMI compliant cabling practices may produce more emissions or less immunity than were obtained in laboratory measurements.**

Operation of this equipment in a residential area may cause unacceptable interference to radio a TV reception, requiring the user to take whatever steps necessary to correct the interference.

Appendix B: Connectors/Pinouts

Cable Harness/Pinouts

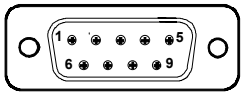
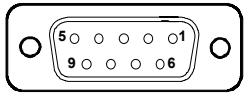
Several cable harness options are available for Intellicon adapters.

> ***Technical Tip:***
> *If you are not using the inputs DCD or CTS on your Intellicon cabling, please ensure that you terminate them properly. The common way to do this is to connect DCD to DTR and/or CTS to RTS.* ***Failure to do so can result in a loss of performance****.*

Intellicon-2

The Intellicon-2 has two DB-9 connectors on the board and therefore does not require a cable harness. The connectors are male when the interface is RS-232 and female when it is RS-485/422. Refer to **Table 9** for the pinouts for these connectors.

Table 9: *Intellicon-2 DB-9 pinouts*

Pin No.	RS-232 Signal	Direction	RS-485 Signal	Direction
1	DCD	input	SR	signal ref.
2	RxD	input	TxD B(+)	output
3	TxD	output	RxD B(+)	input
4	DTR	output	RTS B(+)	output
5	SG	signal gnd.	CTS B(+)	input
6			TxD A(-)	output
7	RTS	output	RxD A(-)	input
8	CTS	input	RTS A(-)	output
9			CTS A(-)	input
	Male DB-9 Connector (RS-232) 		Female DB-9 Connector (RS-485) 	

Intellicon-4

The Intellicon-4 with an RS-232 interface has a cable harness made up of four cables attached to a single DB-37 connector and terminated with DB-25 connectors. The DB-25 connectors on the standard Intellicon-4 cable harness are male, configured as DTE (Data Terminal Equipment). You may order other 25-pin configurations such as male DCE (Data Communications Equipment), female DTE or female DCE. See **Table 10** for the DB-37 connector pinout and **Tables 11** and **12** for the different DB-25 connector pinouts.

You can order an Intellicon-4 with all ports RS-485/422 or with two ports RS-232 and two ports RS-485/422. The cable harness for the Intellicon-4, four ports RS-485/422 has male DB-9 connectors. The cable harness for the Intellicon-4, two ports RS-232, two ports RS-485/422 has male DB-25 DTE connectors(RS-232) and female DB-9 connectors(RS-485/422). Refer to **Tables 11** and **12** for the DB-25 connector pinouts and **Tables 13** and **14** for the DB-9 connector pinouts.

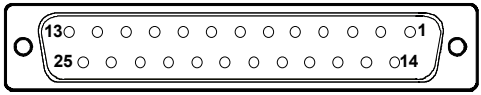
Table 10: *Intellicon-4 DB-37 pinouts*

Pin No.	Port No.	RS-232 Signal	RS-485 Signal
1	1	SG	SR
2	1	DTR	TxD B(+)
3	1	TxD	RxD B(+)
4	1	RxD	RTS B(+)
5	1	DCD	CTS B(+)
6		unused	unused
7	3		TxD A(-)
8	3	CTS	RxD A(-)
9	3	RTS	RTS A(-)
10	3		CTS A(-)
11	4		TxD A(-)
12	4	CTS	RxD A(-)
13	4	RTS	RTS A(-)
14	4		CTS A(-)
15	2	SG	SR
16	2	DTR	TxD B(+)
17	2	TxD	RxD B(+)
18	2	RxD	RTS B(+)
19	2	DCD	CTS B(+)
20	1		TxD A(-)
21	1	CTS	RxD A(-)
22	1	RTS	RTS A(-)
23	1		CTS A(-)
24	3	SG	SR
25	3	DTR	TxD B(+)
26	3	TxD	RxD B(+)
27	3	RxD	RTS B(+)
28	3	DCD	CTS B(+)
29	4	SG	SR
30	4	DTR	TxD B(+)
31	4	TxD	RxD B(+)
32	4	RxD	RTS B(+)
33	4	DCD	CTS B(+)
34	2		TxD A(-)
35	2	CTS	RxD A(-)
36	2	RTS	RTS A(-)
37	2		CTS A(-)

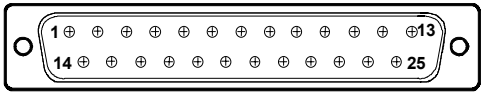
Table 11: *Intellicon-4 DB-25 DTE pinouts*

Pin #	RS-232 Signal	Signal Direction
2	TxD	output
3	RxD	input
4	RTS	output
5	CTS	input
7	SG	signal gnd
8	DCD	input
20	DTR	output

Female DB-25 Connector



Male DB-25 Connector

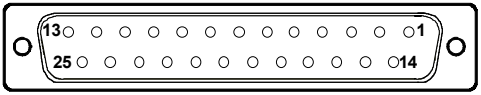


PN: CAB04MT
PN: CAB04FT

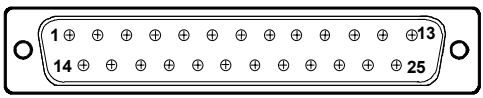
Table 12: *Intellicon-4 DB-25 DCE pinouts*

Pin #	RS-232 Signal	Signal Direction
2	RxD	input
3	TxD	output
4	DCD	input
5	RTS	output
7	SG	signal gnd
8	DTR	output
20	CTS	input

Female DB-25 Connector



Male DB-25 Connector



PN: CAB04MC
PN: CAB04FC

Table 13: *Intellicon-4 DB-9 pinouts*

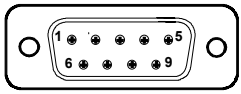
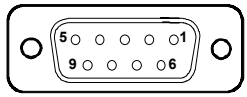
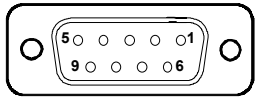
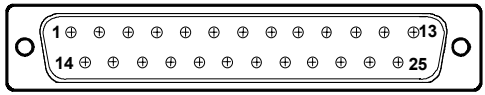
Pin No.	RS-232 Signal	Direction	RS-485 Signal	Direction
1	DCD	input	SR	signal ref.
2	RxD	input	TxD B(+)	output
3	TxD	output	RxD B(+)	input
4	DTR	output	RTS B(+)	output
5	SG	signal gnd.	CTS B(+)	input
6			TxD A(-)	output
7	RTS	output	RxD A(-)	input
8	CTS	input	RTS A(-)	output
9			CTS A(-)	input
	Male DB-9 Connector (RS-232) 		Female DB-9 Connector (RS-485) 	
PN: CAB04DX PN: CAB04485				

Table 14: *Intellicon-2+2 pinouts*

Pin No.	RS-232 Signal	Direction	Pin No.	RS-485 Signal	Direction
2	TxD	output	1	SR	signal ref.
3	RxD	input	2	TxD B(+)	output
4	RTS	output	3	RxD B(+)	input
5	CTS	input	4	RTS B(+)	output
7	SG	signal gnd	5	CTS B(+)	input
8	DCD	input	6	TxD A(-)	output
20	DTR	output	7	RxD A(-)	input
			8	RTS A(-)	output
			9	CTS A(-)	input
	Female DB-9 Connector (RS-485) 				
	Male DB-25 Connector (RS-232) 				
P/N: CAB042+2					

Intellicon-6+2

The Intellicon-6+2 has six ports RS-232 and two ports RS-485/422. The Intellicon-6+2 has a cable harness made up of eight cables attached to a single DB-62 connector and terminated with six DB-25 male connectors(RS-232) and two DB-9 male connectors(RS-485/422). See **Table 15** for the DB-62 connector pinout and **Table 16** for the DB-9 and DB-25 connector pinouts.

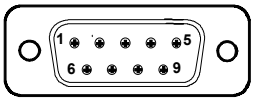
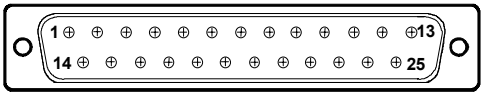
Table 15: *Intellicon-6+2 DB-62 pinouts*

Pin #	Port #	RS-232 Signal	RS-485 Signal	Signal Direction
1	1	DCD		input
2	1	DTR		output
3	1	SG		signal gnd
4	2	DCD		input
5	2	DTR		output
6	3	DCD		input
7	3	DTR		output
8	3	SG		signal gnd
9	4	DCD		input
10	4	DTR		output
11	5	DCD		input
12	5	DTR		output
13	5	SG		signal gnd
14	6	DCD		input
15	6	DTR		output
16	7		TxD A(-)	output
17	7		TxD B(+)	output
18	7		RTS A(-)	output
19	8		TxD A(-)	output
20	8		TxD B(+)	output
21	8		RTS A(-)	output
22	1	RxD		input
23	1	RTS		output
24		unused		
25	2	RxD		input
26	2	RTS		output
27	3	RxD		input
28	3	RTS		output
29		unused		

Table 15(continued): *Intellicon-6+2 DB-62 pinouts*

Pin #	Port #	RS-232 Signal	RS-485 Signal	Signal Direction
30	4	RxD		input
31	4	RTS		output
32	5	RxD		input
33	5	RTS		output
34	8		SG	signal gnd
35	6	RxD		input
36	6	RTS		output
37	7		RxD A(-)	input
38	7		RxD B(+)	input
39	7		RTS B(+)	output
40	8		RxD A(-)	input
41	8		RxD B(+)	input
42	8		RTS B(+)	output
43	1	TxD		output
44	1	CTS		input
45	2	SG		signal gnd
46	2	TxD		output
47	2	CTS		input
48	3	TxD		output
49	3	CTS		input
50	4	SG		signal gnd
51	4	TxD		output
52	4	CTS		input
53	5	TxD		output
54	5	CTS		input
55	6	SG		signal gnd
56	6	TxD		output
57	6	CTS		input
58	7		CTS A(-)	input
59	7		CTS B(+)	input
60	7		SG	signal gnd
61	8		CTS A(-)	input
62	8		CTS B(+)	input

Table 16: *Intellicon-6+2 pinouts*

Pin #	RS-232 Signal	Signal Direction	Pin No.	RS-485 Signal	Signal Direction
2	TxD	output	1	RxD B(+)	input
3	RxD	input	2	TxD B(+)	output
4	RTS	output	3	TxD A(-)	output
5	CTS	input	4	RxD A(-)	input
7	SG	signal gnd	5	SG	signal gnd
8	DCD	input	6	CTS A(-)	input
20	DTR	output	7	RTS A(-)	output
			8	RTS B(+)	output
			9	CTS B(+)	input
	Male DB-9 Connector (RS-485) 				
	Male DB-25 Connector (RS-232) 				
PN: CAB08MA					

Intellicon-8

The Intellicon-8 has a harness made up of eight cables attached to a single DB-62 connector and terminated with eight DB-25 connectors . The DB-25 connectors on the standard Intellicon-8 harness are male, configured as DTE (Data Terminal Equipment). You may order other 25-pin configurations such as male DCE (Data Communications Equipment), female DTE or female DCE. See **Table 17** for the DB-62 connector pinout and **Tables 18** and **19** for the different DB-25 connector pinouts.

Table 17: *Intellicon-8 DB-62 pinouts*

Pin #	Port #	RS-232 Signal	Signal Direction
1	1	DCD	input
2	1	DTR	output
3	1	SG	
4	2	DCD	input
5	2	DTR	output
6	3	DCD	input
7	3	DTR	output
8	3	SG	
9	4	DCD	input
10	4	DTR	output
11	5	DCD	input
12	5	DTR	output
13	5	SG	
14	6	DCD	input
15	6	DTR	output
16	7	DCD	input
17	7	DTR	output
18	7	SG	
19	8	DCD	input
20	8	DTR	output
21		unused	
22	1	RxD	input
23	1	RTS	output
24		unused	
25	2	RxD	input
26	2	RTS	output
27	3	RxD	input
28	3	RTS	output

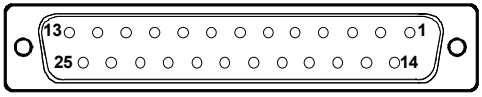
Table 17(continued): *Intellicon-8 DB-62 pinouts*

Pin #	Port #	RS-232 Signal	Signal Direction
29		unused	
30	4	RxD	input
31	4	RTS	output
32	5	RxD	input
33	5	RTS	output
34		unused	
35	6	RxD	input
36	6	RTS	output
37	7	RxD	input
38	7	RTS	output
39		unused	
40	8	RxD	input
41	8	RTS	output
42		unused	
43	1	TxD	output
44	1	CTS	input
45	2	SG	
46	2	TxD	output
47	2	CTS	input
48	3	TxD	output
49	3	CTS	input
50	4	SG	
51	4	TxD	output
52	4	CTS	input
53	5	TxD	output
54	5	CTS	input
55	6	SG	
56	6	TxD	output
57	6	CTS	input
58	7	TxD	output
59	7	CTS	input
60	8	SG	
61	8	TxD	output
62	8	CTS	input

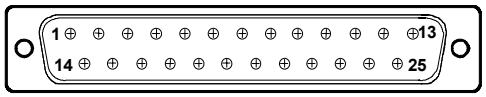
Table 18: *Intellicon-8 DB-25 DTE pinouts*

Pin #	RS-232 Signal	Signal Direction
2	TxD	output
3	RxD	input
4	RTS	output
5	CTS	input
7	SG	signal gnd
8	DCD	input
20	DTR	output

Female DB-25 Connector



Male DB-25 Connector

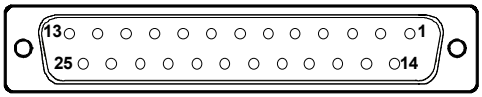


PN: CAB08MT
PN: CAB08FT

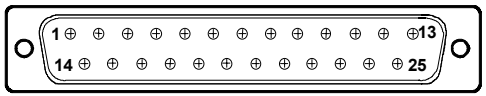
Table 19: *Intellicon-8 DB-25 DCE pinouts*

Pin #	RS-232 Signal	Signal Direction
2	RxD	input
3	TxD	output
4	DCD	input
5	RTS	output
7	SG	signal gnd
8	DTR	output
20	CTS	input

Female DB-25 Connector



Male DB-25 Connector



PN: CAB08MC
PN: CAB08FC

Intellicon-Flex8

The Intellicon-Flex8 has a cable harness made up of eight cables attached to a single DB-78-connector and terminated with eight male DB-9 connectors. See **Table 20** for the DB-78 connector pinout and **Table 21** for the DB-9 connector pinouts.

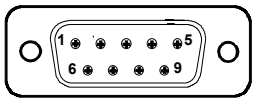
Table 20: *Intellicon-Flex8 DB-78 pinouts*

Pin No.	SLIM/Pin No.	Port No.	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	3-2	5	RTS	RTS A(-)	RTS (-)	TxD(-)
2	3-4	5	CTS	RTS B(+)	RTS Ref	RxD(+)
3	3-6	5	Reserved	CTS A(-)	CTS A(-)	RxD Return
4	3-8	5	Reserved	CTS B(+)	CTS B(+)	RxD Source
5		5	SG	SR	SR	SG
6	3-10	6	RTS	RTS A(-)	RTS (-)	TxD(-)
7	3-12	6	CTS	RTS B(+)	RTS Ref	RxD(+)
8	3-14	6	Reserved	CTS A(-)	CTS A(-)	RxD Return
9	3-16	6		CTS B(+)	CTS B(+)	RxD Source
10	4-2	7	RTS	RTS A(-)	RTS (-)	TxD(-)
11	4-4	7	CTS	RTS B(+)	RTS Ref	RxD(+)
12	4-6	7	Reserved	CTS A(-)	CTS A(-)	RxD Return
13	4-8	7	Reserved	CTS B(+)	CTS B(+)	RxD Source
14		7	SG	SR	SR	SG
15	4-10	8	RTS	RTS A(-)	RTS (-)	TxD(-)
16	4-12	8	CTS	RTS B(+)	RTS Ref	RxD(+)
17	4-14	8	Reserved	CTS A(-)	CTS A(-)	RxD Return
18	4-16	8		CTS B(+)	CTS B(+)	RxD Source
19	NC					
20	NC					
21	3-3	5	TxD	TxD A(-)	TxD (-)	TxD Source
22	3-5	5	RxD	TxD B(+)	TxD Ref	TxD(+)
23	3-7	5	DTR	RxD A(-)	RxD A(-)	TxD Return
24	3-9	5	DCD	RxD B(+)	RxD B(+)	RxD(-)
25		6	SG	SR	SR	SG
26	3-11	6	TxD	TxD A(-)	TxD (-)	TxD Source
27	3-13	6	RxD	TxD B(+)	TxD Ref	TxD(+)
28	3-15	6	DTR	RxD A(-)	RxD A(-)	TxD Return
29	3-17	6	DCD	RxD B(+)	RxD B(+)	RxD(-)
30	4-3	7	TxD	TxD A(-)	TxD (-)	TxD Source
31	4-5	7	RxD	TxD B(+)	TxD Ref	TxD(+)
32	4-7	7	DTR	RxD A(-)	RxD A(-)	TxD Return
33	4-9	7	DCD	RxD B(+)	RxD B(+)	RxD(-)
34	NC					
35	4-11	8	TxD	TxD A(-)	TxD (-)	TxD Source
36	4-13	8	RxD	TxD B(+)	TxD Ref	TxD(+)
37	4-15	8	DTR	RxD A(-)	RxD A(-)	TxD Return
38	4-17	8	DCD	RxD B(+)	RxD B(+)	RxD(-)
39		8	SG	SR	SR	SG

Table 20(continued): *Intellicon-Flex8 DB-78 pinouts*

Pin No.	SLIM/Pin No.	Port No.	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
40	1-2	1	RTS	RTS A(-)	RTS (-)	TxD(-)
41	1-4	1	CTS	RTS B(+)	RTS Ref	RxD(+)
42	1-6	1	Reserved	CTS A(-)	CTS A(-)	RxD Return
43	1-8	1	Reserved	CTS B(+)	CTS B(+)	RxD Source
44		1	SG	SR	SR	SG
45	1-10	2	RTS	RTS A(-)	RTS (-)	TxD(-)
46	1-12	2	CTS	RTS B(+)	RTS Ref	RxD(+)
47	1-14	2	Reserved	CTS A(-)	CTS A(-)	RxD Return
48	1-16	2		CTS B(+)	CTS B(+)	RxD Source
49	2-2	3	RTS	RTS A(-)	RTS (-)	TxD(-)
50	2-4	3	CTS	RTS B(+)	RTS Ref	RxD(+)
51	2-6	3	Reserved	CTS A(-)	CTS A(-)	RxD Return
52	2-8	3	Reserved	CTS B(+)	CTS B(+)	RxD Source
53		3	SG	SR	SR	SG
54	2-10	4	RTS	RTS A(-)	RTS (-)	TxD(-)
55	2-12	4	CTS	RTS B(+)	RTS Ref	RxD(+)
56	2-14	4	Reserved	CTS A(-)	CTS A(-)	RxD Return
57	2-16	4		CTS B(+)	CTS B(+)	RxD Source
58	NC					
59	NC					
60	1-3	1	TxD	TxD A(-)	TxD (-)	TxD Source
61	1-5	1	RxD	TxD B(+)	TxD Ref	TxD(+)
62	1-7	1	DTR	RxD A(-)	RxD A(-)	TxD Return
63	1-9	1	DCD	RxD B(+)	RxD B(+)	RxD(-)
64		2	SG	SR	SR	SG
65	1-11	2	TxD	TxD A(-)	TxD (-)	TxD Source
66	1-13	2	RxD	TxD B(+)	TxD Ref	TxD(+)
67	1-15	2	DTR	RxD A(-)	RxD A(-)	TxD Return
68	1-17	2	DCD	RxD B(+)	RxD B(+)	RxD(-)
69	2-3	3	TxD	TxD A(-)	TxD (-)	TxD Source
70	2-5	3	RxD	TxD B(+)	TxD Ref	TxD(+)
71	2-7	3	DTR	RxD A(-)	RxD A(-)	TxD Return
72	2-9	3	DCD	RxD B(+)	RxD B(+)	RxD(-)
73		4	SG	SR	SR	SG
74	2-11	4	TxD	TxD A(-)	TxD (-)	TxD Source
75	2-13	4	RxD	TxD B(+)	TxD Ref	TxD(+)
76	2-15	4	DTR	RxD A(-)	RxD A(-)	TxD Return
77	2-17	4	DCD	RxD B(+)	RxD B(+)	RxD(-)
78	NC					

Table 21: *Intellicon-Flex8 DB-9 pinouts*

Pin No.	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	DCD	RxD B(+)	RxD(+)	RxD(-)
2	RxD	TxD B(+)	TxD Ref	TxD(+)
3	TxD	TxD A(-)	TxD(-)	TxD Source
4	DTR	RxD A(-)	RxD(-)	TxD Return
5	SG	SR	SR	SG
6	Reserved	CTS A(-)	CTS(-)	RxD Return
7	RTS	RTS A(-)	RTS(-)	TxD(-)
8	CTS	RTS B(+)	RTS Ref	RxD(+)
9	Reserved	CTS B(+)	CTS(+)	RxD Source
	Male DB-9 Connector 			
PN: CAB08FXDX				

WARNING

If an Intellicon-Flex8 adapter has RS232 SLIMs, pins 6 and 9 on the corresponding ports are reserved and you should not make a connection to these pins.

Connector Box /Pinouts

You may order the Intellicon-8 and the Intellicon-Flex8 adapters with an external I/O Box option. The I/O Box option comes with a metal bracket that can be mounted on a wall or other surface. When you receive the I/O Box, this bracket is clipped on to the back of the connector box. If you wish to attach the I/O Box to a wall or other surface, just remove the bracket, fasten it in place, and then re-attach the connector box. If you wish to set the I/O Box on its rubber feet only, just remove the bracket from the back. See **Figures 21** and **22** for the orientation of the I/O Box.

Figure 21: *Intellicon-8 I/O Box*

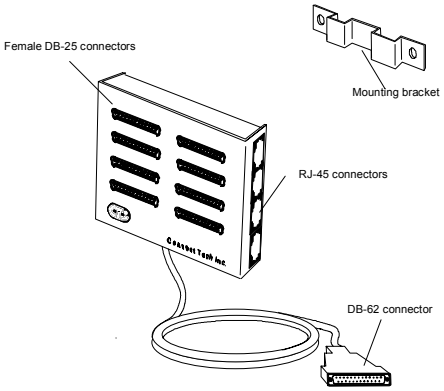
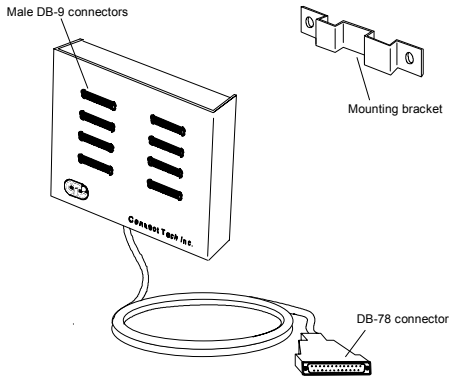


Figure 22: *Intellicon-Flex8 I/O Box*



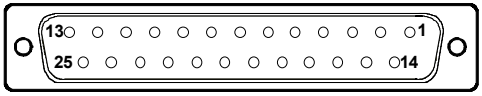
Intellicon-8

You can order the Intellicon-8 with the I/O Box option. The eight port connector box connects to Intellicon-8 adapter via a 62 pin connector. For DB-62 pinout information please refer to the **Table 17**. The I/O Box provides both female DB-25 and RJ-45 connectors for each port. The DB-25 connectors are configured for DCE. For the DB-25 pinout information refer to **Table 22**. For the RJ-45 pinout information refer to **Table 23.**

Table 22: *Intellicon-8 - I/O Box DB-25 DCE pinouts*

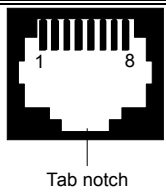
Pin #	RS-232 Signal	Signal Direction
2	RxD	input
3	TxD	output
4	DCD	input
5	RTS	output
7	SG	signal gnd
8	DTR	output
20	CTS	input

Female DB-25 Connector



PN: IOB08

Table 23: *Intellicon-8 - I/O Box RJ-45 pinouts*

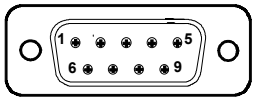
Wire #	RJ modular connector			Signal Direction
	8 wire	6 wire	4 wire	
1	DTR			output
2	RTS	RTS		output
3	SG	SG	SG	signal ground
4	TxD	TxD	TxD	output
5	SG	SG	SG	signal ground
6	RxD	RxD	RxD	input
7	CTS	CTS		input
8	DCD			input
				
PN: IOB08				

Note: *If you wish to use 6 wire or 4 wire modular connectors instead of 8 wire modular connectors, please refer to the above table to determine which signals are available for the different connector types.*

Intellicon-Flex8

You can order the Intellicon-Flex8 with the I/O Box option. The eight port connector box connects to Intellicon-Flex8 adapter via a 78-pin connector. For DB-78 pinout information please refer to the **Table 20**. The I/O Box provides male DB-9 connectors for each port. For the DB-9 pinout information refer to **Table 24.**

Table 24: *Intellicon-Flex8, I/O Box DB-9 pinouts*

Pin No.	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	DCD	RxD B(+)	RxD(+)	RxD(-)
2	RxD	TxD B(+)	TxD Ref	TxD(+)
3	TxD	TxD A(-)	TxD(-)	TxD Source
4	DTR	RxD A(-)	RxD(-)	TxD Return
5	SG	SR	SR	SG
6	Reserved	CTS A(-)	CTS(-)	RxD Return
7	RTS	RTS A(-)	RTS(-)	TxD(-)
8	CTS	RTS B(+)	RTS Ref	RxD(+)
9	Reserved	CTS B(+)	CTS(+)	RxD Source
	Male DB-9 Connector 			
PN: IOB08DB9V1				

Appendix C: Factory Settings

Appendix C summarizes the factory settings for the following Intellicon adapters:

- Intellicon-2 and Intellicon-4
- Intellicon-6+2
- Intellicon-8
- Intellicon-Flex8

WARNING

Do not use a pencil to set the DIP switches as the lead graphite may short-circuit the switch.

ATTENTION
Contents
Static Sensitive

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

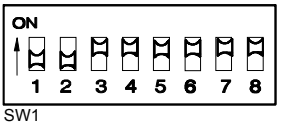
Intellicon-2/Intellicon-4

The Intellicon-2 adapter ships with factory settings for the following:

- DIP switch SW1 - I/O port address
- DIP switch SW2 - base memory address for Intellicon-2, RS-232 and Intellicon-4, RS-232
- DIP switch SW2 - base memory address and TxD/RTS settings for Intellicon-2, RS-485/422 and Intellicon-4, 4 ports RS-485/422; or 2 ports RS-232 and 2 ports RS-485/422.
- Jumpers J1 and J2 - interrupt request line (IRQ)

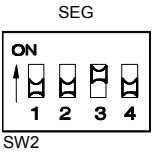
Please refer to **Figures 23, 24, 25** and **26** for the above factory settings. **Chapter 1: Introduction** shows the location on the adapter for the switches and jumpers.

Figure 23: *Intellicon-2/-4 - SW1 factory setting*



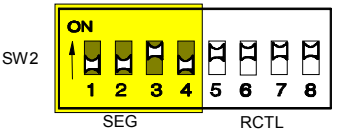
Note: *DIP switch SW1 factory set for I/O port address 300 - 303 hex for Intellicon-2 and Intellicon-4, RS-232 and/or RS-485/422*

Figure 24: *Intellicon-2/-4(RS-232) SW2 factory setting*



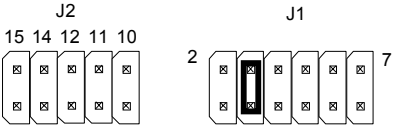
Note*: DIP switch SW2 factory set for base memory address D0000 hex for Intellicon-2, RS-232 and Intellicon-4, RS-232*

Figure 25: *Intellicon-2/-4(RS-485/422) SW2 setting*



Note: *DIP switch SW2 factory set for base memory address D0000 hex and RxD/RTS enabled on ports 1, 2, 3, and 4 for Intellicon-2, RS-485/422 and Intellicon-4, 4 ports RS-485/422 or 2 ports RS-232 and 2 ports RS-485/422*

Figure 26: *Intellicon-2/-4 - J1, J2 factory settings*



Note: *Jumpers J1 and J2 factory set for IRQ 3 for Intellicon-2 and Intellicon-4, RS-232 and/or RS-485/422*

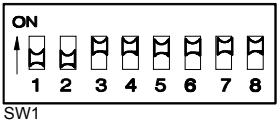
Intellicon-6+2

The Intellicon-6+2 adapter ships with factory settings for the following:

- DIP switch SW1 - I/O port address
- DIP switch SW2 - Base memory address
- IRQ jumper - Interrupt Request line
- Jumpers J7 and J8 - Rxd/CTS enable/disable
- Jumpers J3, J4, J5 and J6 - RxD/CTS line load
- Jumpers J9 and J10 - internal/external CTS

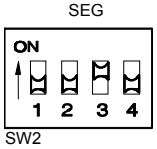
Please refer to **Figures 27, 28, 29, 30, 31** and **32** for the above factory settings. **Chapter 1: Introduction** shows the locations on the adapter for the switches and jumpers

Figure 27: *Intellicon-6+2 - SW1 factory setting*



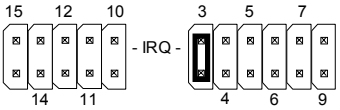
Note: *DIP switch SW1 factory set for I/O port address, 300-303 hex for an Intellicon-6+2*

Figure 28: *Intellicon-6+2 - SW2 factory setting*



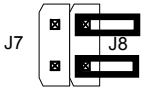
Note: *DIP switch SW2 factory set for base memory address D000 hex for an Intellicon-6+2*

Figure 29: *Intellicon-6+2 - IRQ factory settings*



Note: *IRQ jumper factory set for IRQ 3 for an Intellicon-6+2*

Figure 30: *Intellicon-6+2 - J7, J8 factory settings*

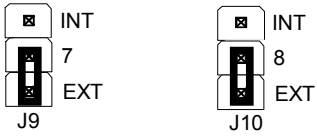


Note: *Jumpers J7 and J8 factory set for RxD/CTS disabled on ports 7 and 8*

Figure 31: *Intellicon-6+2 - J3, J4, J5, J6 factory settings*

Note: *Jumpers J3, J4, J5 and J6 factory set for R/C series loads for RxD and CTS on ports 7 and 8*

Figure 32: *Intellicon-6+2 - J9, J10 factory settings*



Note: *Jumpers J9 and J10 factory set for external CTS on ports 7 and 8*

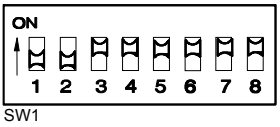
Intellicon-8

The Intellicon-8 adapter ships with factory settings for the following:

- DIP switch SW1 - I/O port address
- DIP switch SW2 - Base memory address
- IRQ jumper - Interrupt Request line

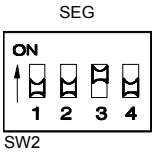
Please refer to **Figures 33, 34, and 35** for the above factory settings. **Chapter 1: Introduction** shows the locations on the adapter of the switches and jumpers

Figure 33: *Intellicon-8 - SW1 factory setting*



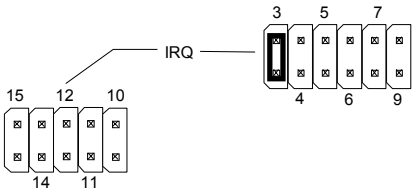
Note: *DIP switch SW1 factory set for I/O port address, 300-303 hex for an Intellicon-8*

Figure 34: *Intellicon-8 - SW2 factory setting*



Note: *DIP switch SW2 factory set for base memory address D0000 hex for an Intellicon-8*

Figure 35: *Intellicon-8 - IRQ factory settings*



Note: *IRQ jumper factory set for IRQ 3 for an Intellicon-8*

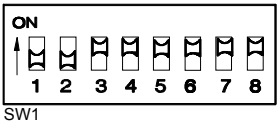
Intellicon-Flex8

The Intellicon-2 adapter ships with factory settings for the following:

- DIP switch SW1 - I/O port address
- DIP switch SW2 - base memory address
- Jumpers J1 and J2 - interrupt request line (IRQ)

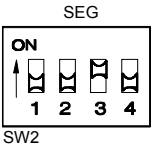
Please refer to **Figures 36, 37, and 38** for the above factory settings. **Chapter 1: Introduction** shows the locations on the adapter of the switches and jumpers

Figure 36: *Intellicon-Flex8 - SW1 factory setting*



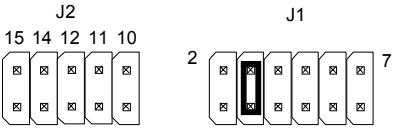
Note: *DIP switch SW1 factory set for I/O port address 300 - 303 hex for Intellicon-Flex8*

Figure 37: *Intellicon-Flex8 - SW2 factory setting*



Note: *DIP switch SW2 factory set for base memory address D0000 hex for Intellicon-Flex8*

Figure 38: *Intellicon-Flex8 - IRQ factory settings*



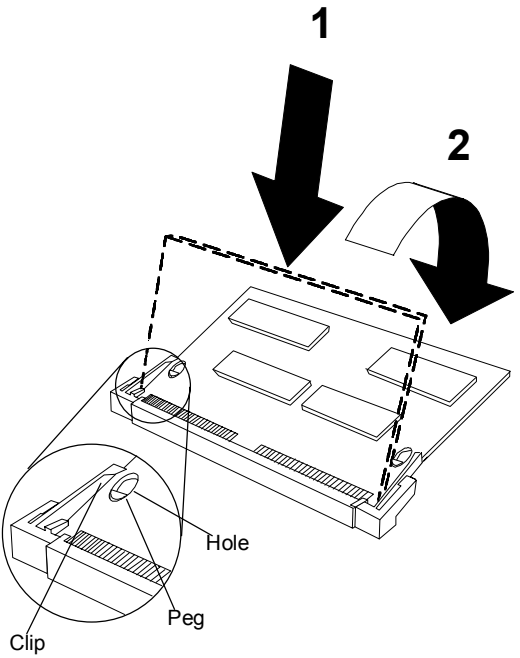
Note: *Jumpers J1 and J2 factory set for IRQ 3 for Intellicon-Flex8*

Appendix D: SLIM Insertion/Removal

SLIM insertion

Figure 39 outlines the correct procedure for inserting RS-232, RS-485/422, RS-423, and 20mA curent loop Serial Line Interface Module (SLIM).

Figure 39: *SLIM insertion*

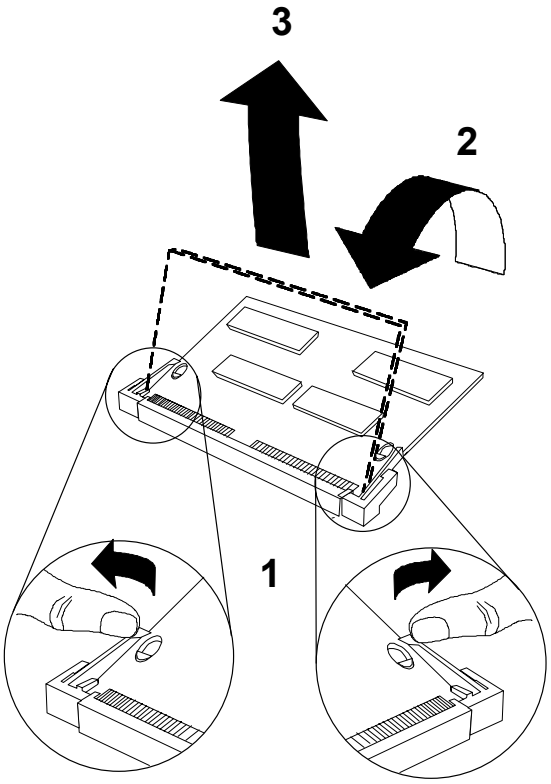


1. Place the SLIM into the socket almost vertically (about 75 degree angle), making sure that it is properly oriented and fully inserted into the socket.
2. Press downward and sideways on the SLIM until it latches into the socket. **Do NOT force the SLIM**, the installation process requires a small force and should be very smooth and easy. If you encounter resistance then re-check the orientation and insertion depth.

SLIM removal

Figure 40 outlines the correct procedure for removing a RS-232, RS-423, RS-485/422, and 20mA current loop Serial Line Interface Module (SLIM).

Figure 40: *SLIM removal*



1. Using both hands, place your thumbs on the clips that hold the SLIM into the socket, and place your index fingers on the edge of the SLIM.
2. Push outwards (with your thumbs) on the clips until the SLIM is released from the clips. Your index fingers can then raise the SLIM past the clips. The SLIM is then free to be removed from the Intellicon-Flex8 board. The force required to open the clips is small and the SLIM naturally springs upwards as soon as the clips are open wide enough. If removal is difficult, then you are probably trying to raise the SLIM before the clips are open.

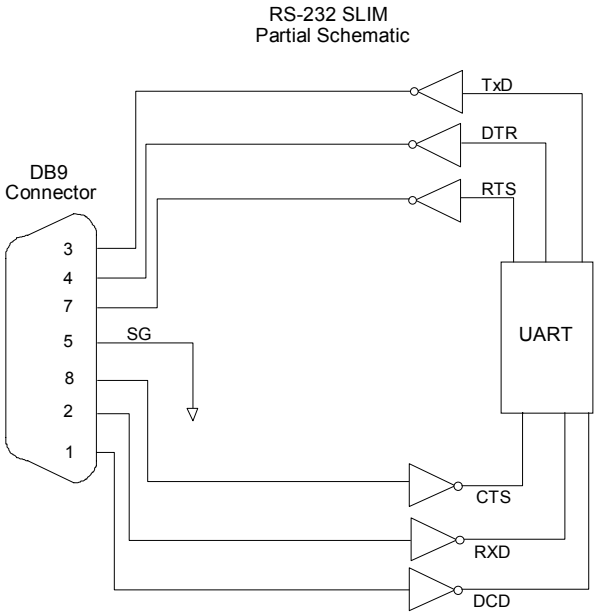
Appendix E: RS-232 Option

You may order various models of Intellicon-Flex8 adapters with RS-232 Serial Line Interfaces Modules. The RS-232 SLIM offers the following features:

- The RS-232 electrical interface, an industry standard that offers connection to a wide range of peripheral devices.
- Each SLIM controls two ports.
- SLIMS are field upgradeable, providing you with more flexibility for your application.

Please refer to **Figure 41** for a partial schematic of the RS-232 SLIM

Figure 41: *RS-232 SLIM: partial schematic*



Note: *The RS-232 Serial Line Interface Module controls two ports, and therefore the schematic shows only a portion of the circuit.*

Installation

The Intellicon-Flex8 adapter has four SLIM sockets which accept Serial Line Interface Modules. SLIM 1 socket (S1) interfaces ports 1and 2. SLIM 2 socket (S2) interfaces ports 3 and 4. SLIM 3 socket (S3) interfaces ports 5 and 6. SLIM 4 socket (S4) interfaces ports 7 and 8. Please refer to **Chapter 2: Hardware Installation** for the location of the SLIM sockets and to **Appendix D: SLIM Insertion/Removal** for the proper procedures to installing and removing SLIMs.

Your Intellicon adapter and Serial Line Interface Module (SLIM) are very sensitive to static electricity. Make sure that before you remove them from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove them from the anti-static bag, handle them only by the edges and place them on the anti-static bag or an anti-static mat.

Please refer to **Table 25** for the pinouts for the SLIM sockets found on the Intellicon-Flex8 adapter.

Table 25: *SLIM socket pinouts*

SLIM Pin No.	TTL Signal	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	Reserved				
2		RTS 1	RTS A(-) 1	RTS (-) 1	TxD(-) 1
3		TxD 1	TxD A(-) 1	TxD (-) 1	TxD Source 1
4		CTS 1	RTS B(+) 1	RTS Ref 1	RxD(+) 1
5		RxD 1	TxD B(+) 1	TxD Ref 1	TxD(+) 1
6		Reserved	CTS A(-) 1	CTS A(-) 1	RxD Return 1
7		DTR 1	RxD A(-) 1	RxD A(-) 1	TxD Return 1
8		Reserved	CTS B(+) 1	CTS B(+) 1	RxD Source 1
9		DCD 1	RxD B(+) 1	RxD B(+) 1	RxD(-) 1
10		RTS 2	RTS A(-) 2	RTS (-) 2	TxD(-) 2
11		TxD 2	TxD A(-) 2	TxD (-) 2	TxD Source 2
12		CTS 2	RTS B(+) 2	RTS Ref 2	RxD(+) 2
13		RxD 2	TxD B(+) 2	TxD Ref 2	TxD(+) 2
14		Reserved	CTS A(-) 2	CTS A(-) 2	RxD Return 2
15		DTR 2	RxD A(-) 2	RxD A(-) 2	TxD Return 2
16		NC	CTS B(+) 2	CTS B(+) 2	RxD Source 2
17		DCD 2	RxD B(+) 2	RxD B(+) 2	RxD(-) 2
18	Ground				
19	+12 V				
20	-12 V				
21	+5 V				
22	DCD 2		Tied high	Tied high	Tied high
23	CTS 2				Tied high
24	RTS 2				
25	DTR 2				
26	TxD 2				
27	RxD 2				
28	RxD 1				
29	TxD 1				
30	DTR 1				
31	RTS 1				
32	CTS 1				Tied high
33	DCD 1		Tied high	Tied high	Tied high
34	TC				
35	RC				

Note: *earlier versions of the current loop SLIM have pins 22, 23, 32, and 33 tied low.*

Appendix F: RS-485/422 Option

You may order various models of Intellicon adapters with RS-485/422 interfaces. The RS-485/422 electrical interface is a reliable high speed serial link that offers superior noise immunity and multi-drop network connectivity. The RS-485/422 electrical interface is also a superset of the RS-422 electrical interface. The Intellicon adapters that offer a RS-485/422 interface option are:

- Intellicon-2
- Intellicon-4
- Intellicon-Flex8
- Intellicon-6+2

RS-485/422(Intellicon-2/-4)

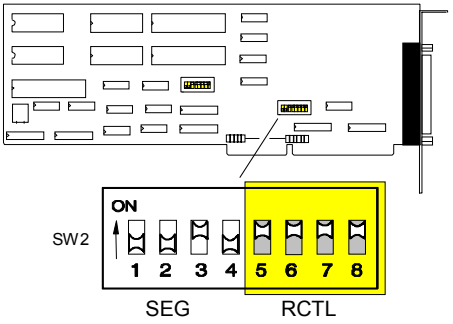
If your Intellicon-2 or 4 port board has RS-485/422 drivers and receivers, you will find SN75175N receivers installed at sites U27 and/or U29 and SN75174N transmitters installed at sites U28 and/or U30 on the board. You must set switches 5 through 8 on DIP Switch 2 (SEG/RCTL) to control the driver enables on ports 1 through 4 respectively. See **Figure 42** for the location of SW2 on the Intellicon-2/-4 adapters.

WARNING

Do not use a pencil to set the DIP switches as the lead graphite may short-circuit the switch.

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

Figure 42: *Intellicon-2/-4: RS-485/422 switch(SW2)*

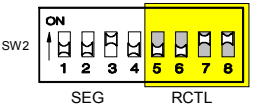


If the DIP switches 5 through 8 are ON, the TxD and RTS drivers will always be enabled. If the switch is off, the drivers will only be enabled when the RTS is true on the port. The RTS can be controlled by the PC to control the state of the line.

Examples:

The following example shows SW2(RCTL) set for TxD and RTS drivers always enabled on ports 1, 2, 3 and 4:

The following example shows SW2(RCTL) set for TxD and RTS drivers enabled only when RTS is true on ports 1 and 2, and always enabled on ports 3 and 4.



The RS-485/422 option is factory set to terminate each receiver with a 120 Ohm load. However differential inputs not driven are susceptible to receiver noise. On the RxD lines, this can generate unwanted input characters and breaks. On the CTS line, this noise can significantly degrade performance as the co-processor becomes busy processing the transitions.

You should follow one of two approaches to alleviate the problem of receiver noise:

1. If you do not require the CTS input, you may tie the CTS input to the RTS output.
2. If, however, you use CTS or RxD but the driver on the remote device is not always enabled, you should bias the inputs with resistors.

Technical Tip
Many RS485/422 devices are designed to work in a multi-drop environment. These devices will enable their transmitters only when they send data. When no device transmits, the receivers on the other end are susceptible to noise, unless you install biasing resistors to maintain the differential voltage. You usually bias signals to the high state so that the line is inactive when not driven. ***Figure 43*** *shows the location of the resistor sites on the Intellicon-2/-4.* ***Table 26*** *lists the resistor locations of default high and default low states for the signals on each port. To default a signal, solder a pair of 1K Ohm resistors into the appropriate locations.*

Figure 43: *Intellicon-2/-4: RS-485/422 resistor sites*

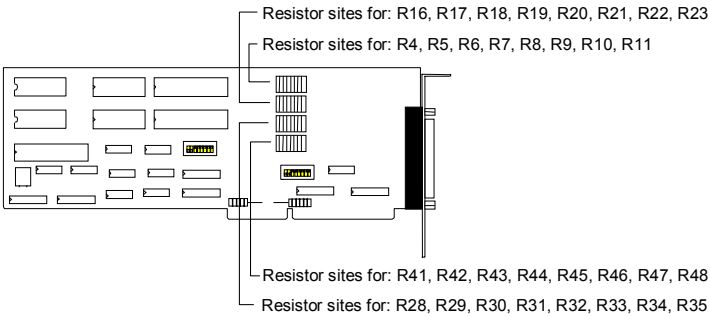


Table 26: *Intellicon-2/-4, RS-485/422 resistor defaults*

Port #	Signal	Resistor	Default(high)	Default(low)
1	RxD	R1	R9, R10	R8, R11
1	CTS	R2	R4, R7	R5, R6
2	RxD	R13	R16, R19	R17, R18
2	CTS	R14	R21, R22	R20, R23
3	RxD	R25	R33, R34	R32, R35
3	CTS	R26	R28, R31	R29, R30
4	RxD	R38	R41, R44	R42, R43
4	CTS	R39	R46, R47	R45, R48

The Intellicon-2 and Intellicon-4 also provides a signal reference (SR) pin with a 100 Ohm current limiting resistor to signal ground. This may be connected to the far end to provide common mode reference.

RS-485/422(Intellicon-Flex8)

You may order Intellicon-Flex8 adapters with RS-485/422 Serial Line Interface Modules (SLIM). The RS-485/422 electrical interface provides a reliable high speed serial link over long distances that offers superior noise immunity and multi-drop network connectivity.

WARNING

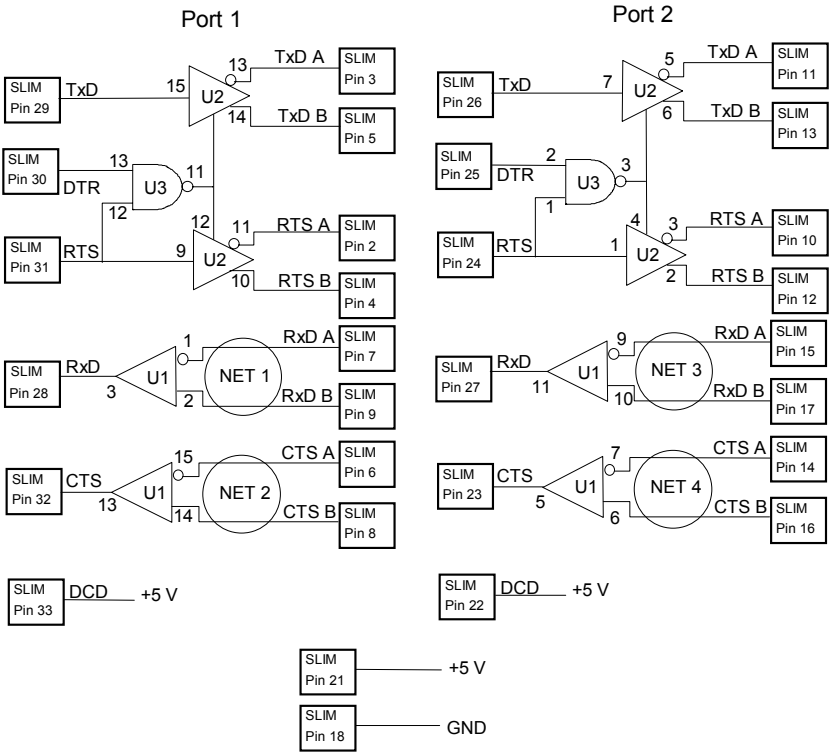
You must insert Serial Line Interface Modules into all of the SLIM sockets before operating your Intellicon adapter. Failure to observe this precaution will result in damage to the Intellicon adapter.

The Intellicon-Flex8 does not provide DIP switches to control the driver enables for the RS-485/422 Serial Line Interface Modules (SLIM). Instead, the DTR signal is used to gate the control. If DTR is asserted, the drivers are always enabled. If DTR is not asserted, then the driver will be enabled only when RTS is asserted.

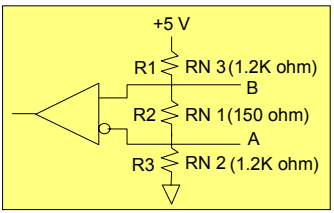
Also, the RS-485/422 SLIM does not provide locations for biasing resistors. Each receiver is biased high through fixed resistors. Please refer to **Figure 44** for a partial schematic of the RS-485/422 SLIM circuit, while **Table 27** shows the pinouts for the SLIM sockets found on the Intellicon-Flex8 adapter.

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

Figure 44: *RS-485/422 SLIM partial schematic*



Network detail



Net #	RN 1 (150 ohm) Pin #	RN 2, RN 3 (1.2k ohm) Pin #
Net 1	7, 8	5
Net 2	5, 6	4
Net 3	3, 4	3
Net 4	1, 2	2

Note: *As a RS-485/422 Serial Line Interface Module controls two ports, the schematic shows the circuit for two ports.*

Table 27: *SLIM socket pinouts*

SLIM Pin No.	TTL Signal	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	Reserved				
2		RTS 1	RTS A(-) 1	RTS (-) 1	TxD(-) 1
3		TxD 1	TxD A(-) 1	TxD (-) 1	TxD Source 1
4		CTS 1	RTS B(+) 1	RTS Ref 1	RxD(+) 1
5		RxD 1	TxD B(+) 1	TxD Ref 1	TxD(+) 1
6		Reserved	CTS A(-) 1	CTS A(-) 1	RxD Return 1
7		DTR 1	RxD A(-) 1	RxD A(-) 1	TxD Return 1
8		Reserved	CTS B(+) 1	CTS B(+) 1	RxD Source 1
9		DCD 1	RxD B(+) 1	RxD B(+) 1	RxD(-) 1
10		RTS 2	RTS A(-) 2	RTS (-) 2	TxD(-) 2
11		TxD 2	TxD A(-) 2	TxD (-) 2	TxD Source 2
12		CTS 2	RTS B(+) 2	RTS Ref 2	RxD(+) 2
13		RxD 2	TxD B(+) 2	TxD Ref 2	TxD(+) 2
14		Reserved	CTS A(-) 2	CTS A(-) 2	RxD Return 2
15		DTR 2	RxD A(-) 2	RxD A(-) 2	TxD Return 2
16		NC	CTS B(+) 2	CTS B(+) 2	RxD Source 2
17		DCD 2	RxD B(+) 2	RxD B(+) 2	RxD(-) 2
18	Ground				
19	+12 V				
20	-12 V				
21	+5 V				
22	DCD 2		Tied high	Tied high	Tied high
23	CTS 2				Tied high
24	RTS 2				
25	DTR 2				
26	TxD 2				
27	RxD 2				
28	RxD 1				
29	TxD 1				
30	DTR 1				
31	RTS 1				
32	CTS 1				Tied high
33	DCD 1		Tied high	Tied high	Tied high
34	TC				
35	RC				

Note: *earlier versions of the current loop SLIM have pins 22, 23, 32, and 33 tied low.*

RS-485/422(Intellicon-6+2)

The Intellicon-6+2 adapter provides 6 ports of RS-232(ports 1 through 6) and 2 ports of RS-485/422 (ports 7 and 8). The RS-485/422 interface on the Intellicon-6+2 gives the user maximum versatility in configuring their RS-485/422 communication line. Features include:

- TxD/RTS enable with DTR
- TxD/RTS enable with RTS
- Rxd/CTS disable
- RxD line load options
- RxD line biasing
- CTS selection

Figure 45 shows a partial schematic of the RS-485/422 interface on the Intellicon-6+2 adapter, while **Figure 46** shows the locations of the jumpers that configure the RS-485/422 interface.

Your Intellicon adapter is very sensitive to static electricity. Make sure that before you remove the card from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove the board from the anti-static bag, handle it only by the edges and place it on the anti-static bag or an anti-static mat.

Figure 45: *Intellicon-6+2: RS-485/422 schematic(partial)*

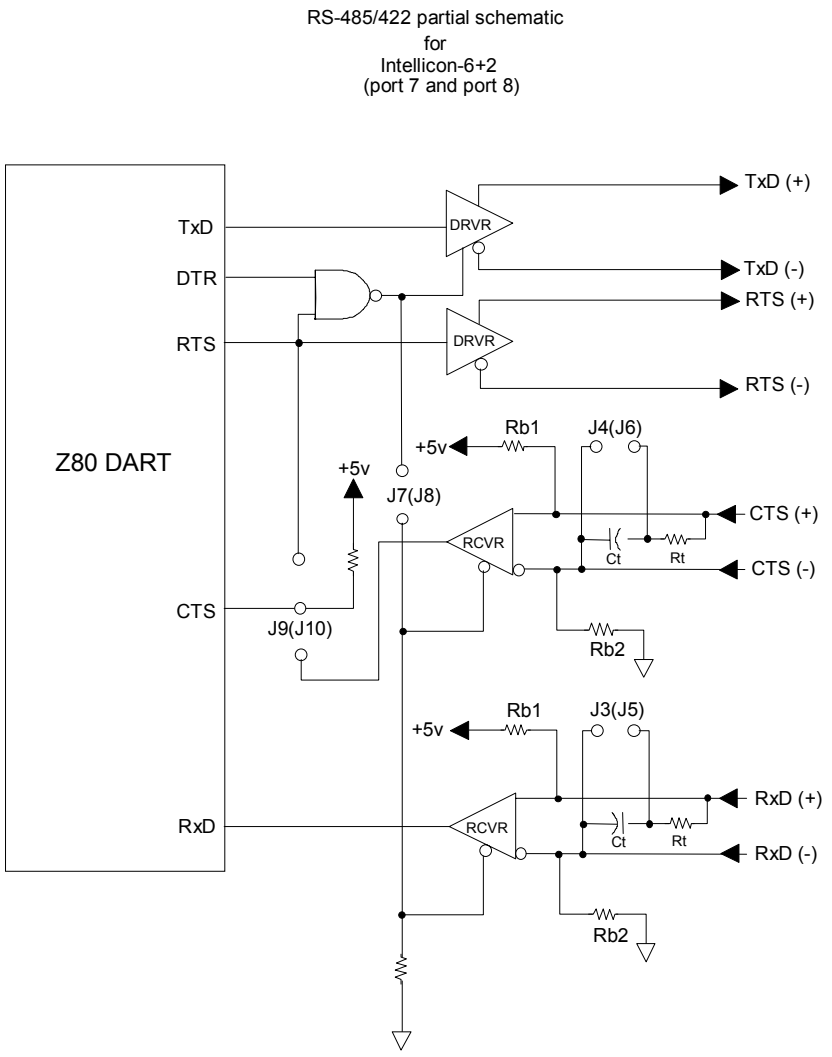
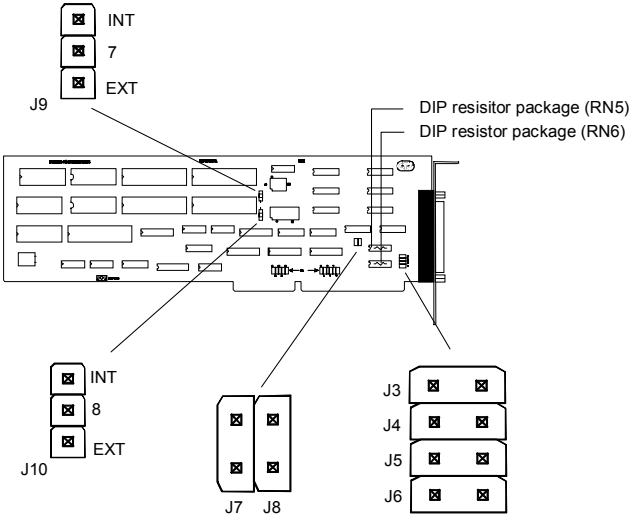


Figure 46: *Intellicon-6+2: Jumper block locations*



TxD/RTS enable with DTR

When the DTR signal is turned "ON" the TxD and the RTS driver circuits are permanently enabled. You use this in simple point to point systems where you do not require these drivers to be "off" the line.

TxD/RTS enable with RTS

This configuration is activated by turning the DTR signal OFF and then enabling the RTS signal to go active during data transmission. This feature is often referred to as "auto-rts". Both the TxD and RTS drivers are active during data transmission and when transmission finishes the drivers disable (tri-state). You use this configuration most often for half-duplex and multi-drop communications links.

RxD/CTS disable

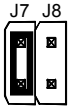
When using either a half-duplex or multi-drop configuration the RxD of a given port is connected to its own TxD. This creates a situation where a port will send data to itself. You can disable this by installing a jumper on J7 and J8. Refer to **Figure 46** for their location and **Table 28** for their function.

Table 28: *Intellicon-6+2 RxD/CTS disable*

Jumper No.	Port No.	Jumper	
		Removed	**Installed**
J7	7	Normal operation	RxD/CTS disabled
J8	8	Normal operation	RxD/CTS disabled

Examples:

1. *The following shows RxD and CTS enabled on port 7 and disabled on port 8*



2. *The following shows RxD and CTS enabled on port 7 and port 8*

RxD/CTS line load options

RS-485/422 lines normally need some type of load to prevent impedance mis-matches from causing signal interference. The Intellicon-6+2 provides two types of load, one is a purely resistive ("R") load and the other is a resistive-capacitive ("RC") load. The RS-485/422 schematic in **Figure 45** shows these as "Rt" and "Ct". You can apply these loads on ports 7 and 8 by installing or not installing jumpers at J3, J4, J5 and J6. Refer to **Figure 46** for their location and **Table 29** for their function.

Table 29: *Intellicon-6+2 RxD/CTS line loads*

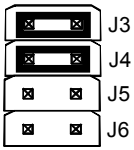
Jumper No.	Port No.	Signal	Jumper	
			Removed	Installed
J3	7	RxD	R/C series load	Resitive load
J4	7	CTS	R/C series load	Resitive load
J5	8	RxD	R/C series load	Resitive load
J6	8	CTS	R/C series load	Resitive load

You should use the "R" load for two situations: a point to point connection that is less than 50 meters (150') long and for a "last station load" on a multi-drop line of less than 50 meters. To enable this load install a jumper on J3, J4, J5, and/or J6.

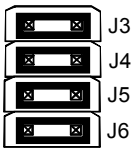
You use the "RC" load for situations described above, where the line length exceeds 50 meters. To enable this load remove the appropriate jumper on J3, J4, J5 and/or J6. The RS-485/422 schematic in **Figure 45** shows these jumpers as "Jc".

Examples:

1. *The following example shows port 7 set for a point to point connection less than 50 metres("R" load) and port 8 set for a point to point connection over 50 metres("RC" load).*



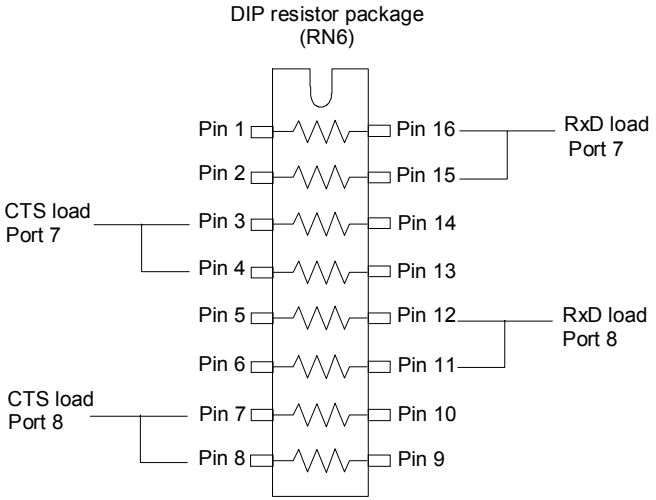
2. *The following example shows port 7 set for a point to point connection less than 50 metres("R" load) and port 8 set for a last station load("R" load) on a multi-drop line of less than 50 metres.*



For situations where you do not need the line load, ie: short lines or intermediate stations on a multi-drop line, you can remove the load from the board by removing the DIP resistor package at RN6 from its socket. Please refer to **Figure 46** for the location of RN6.

If you require a custom load, you can remove the DIP resistor package at RN6 and replace it with either, a DIP resistor package of a different value or fabricate a custom resistor load on a component header and insert it into the DIP resistor socket at RN6. Please refer to **Figure 47** for the pinout information of the DIP resistor socket at RN6

Figure 47: *DIP resistor RN6 pinout*



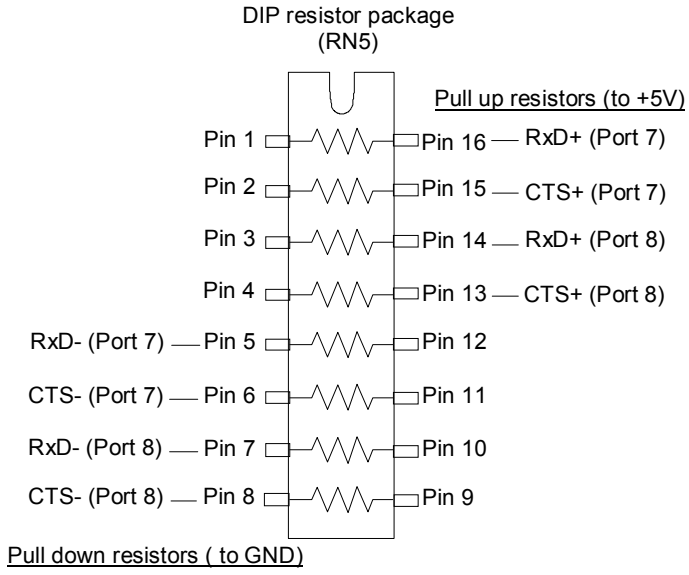
Note:
Each load consists of 2 resistors wired in parallel. This allows you to adjust the load resistance to just about any value.

RxD/CTS line biasing

In half-duplex or some types of multi-drop situations there is the condition where there may be no active driver on the line. This creates a line that is electrically "floating" and the possibility of other electrical signals interfering with the operation of the RS485 communication link.

To help prevent interference, CTI has added line biasing resistors (shown as "Rb1, Rb2" on the schematic in **Figure 45**) in a DIP resistor package RN5. You can remove or replace this resistor package to implement special circumstances. Please refer to **Figure 46** for the location and **Figure 48** for the pinout information of RN5.

Figure 48: *DIP resistor RN5 pinout*



CTS selection

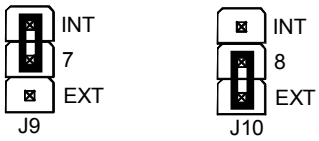
On the Intellicon-6+2 you can program the CTS signal on ports 7 and 8 for three possible modes of operation. One way has CTS connected directly to RTS. Another method connects CTS to a line receiver circuit, therefore allowing the external communication link to supply the CTS signal. Finally you can set CTS permanently false(OFF). This is useful for half duplex communication links and for other special user needs. You program all three modes by installing or not installing jumpers at J9 and J10. Refer to **Figure 46** for their location and **Table 30** for their function.

Table 30: *Intellicon-6+2 CTS selection*

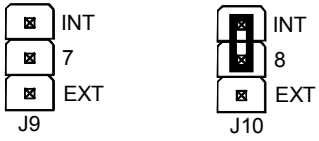
Jumper No.	Jumpers installed	Function
J9	INT/7	Port 7 - Internal CTS
J9	7/EXT	Port 7 - External CTS
J10	INT/8	Port 8 - Internal CTS
J10	EXT/8	Port 8 - External CTS

Example:

1. *The following example shows port 7 set for internal CTS and port 8 for external CTS*



2. *The following example shows port 7 set for CTS permanent false(OFF) and port 8 set for internal CTS*

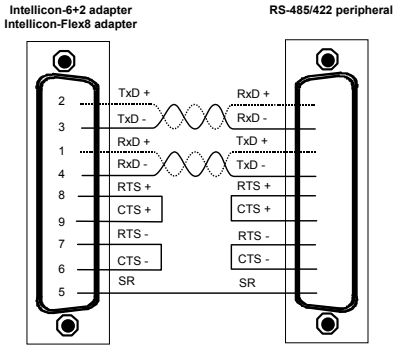


RS-485/422 Cable Wiring

You can wire Intellicon adapters with the RS-485/422 option in various ways to communicate with RS-485/422 peripherals. This section will descibe a few examples of RS-485/422 cabling schemes.

Figures 49 and 51 describe 4 wire and 2 wire cabling schemes between a port on a Intellicon-6+2 or Intellicon-Flex8 adapter to a port on the RS-485/422 peripheral. **Figures 50** and **52** describe 4 wire and 2 wire cabling schemes between a port on a Intellicon-2, Intellicon-4 or Intellicon-2+2 adapter to a port on the RS-485/422 peripheral.

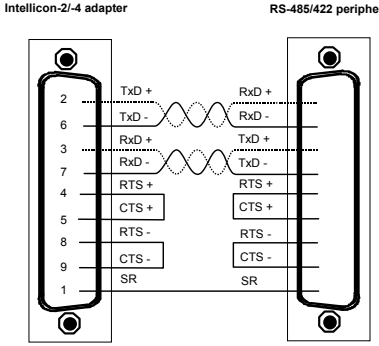
Figure 49: *RS-485/422 wiring diagram(4 wire)*



Note:
The RTS, and CTS connections are not necessary if you jumper the CTS selection (J9 and J10) appropriately on the Intellicon-6+2

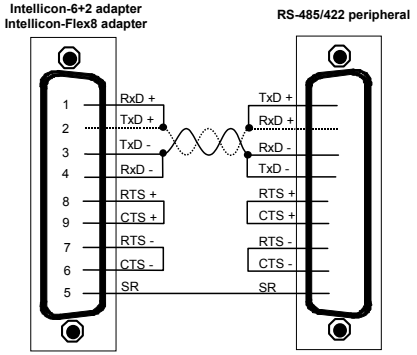
Technical Tip
The RS-485/422 electrical interface consists of a differential signaling scheme. ***You should always connect the signals with twisted pairs***

Figure 50: *Intellicon-2/-4:RS-485/422 wiring diagram(4 wire)*



Technical Tip
The RS-485/422 electrical interface consists of a differential signaling scheme. ***You should always connect the signals with twisted pairs***

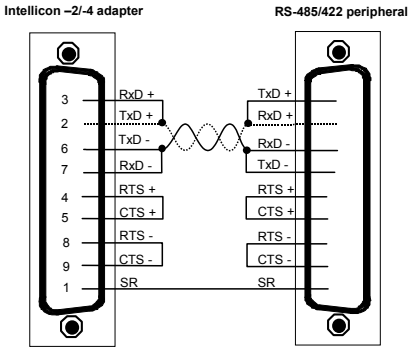
Figure 51: *RS-485/422 wiring diagram(2 wire)*



Note:
The RTS, and CTS connections are not necessary if you jumper the CTS selection (J9 and J10) appropriately on the Intellicon-6+2

Technical Tip
The RS-485/422 electrical interface consists of a differential signaling scheme. ***You should always connect the signals with twisted pairs***

Figure 52: *Intellicon-2/-4: RS-485/422 wiring diagram(2 wire)*



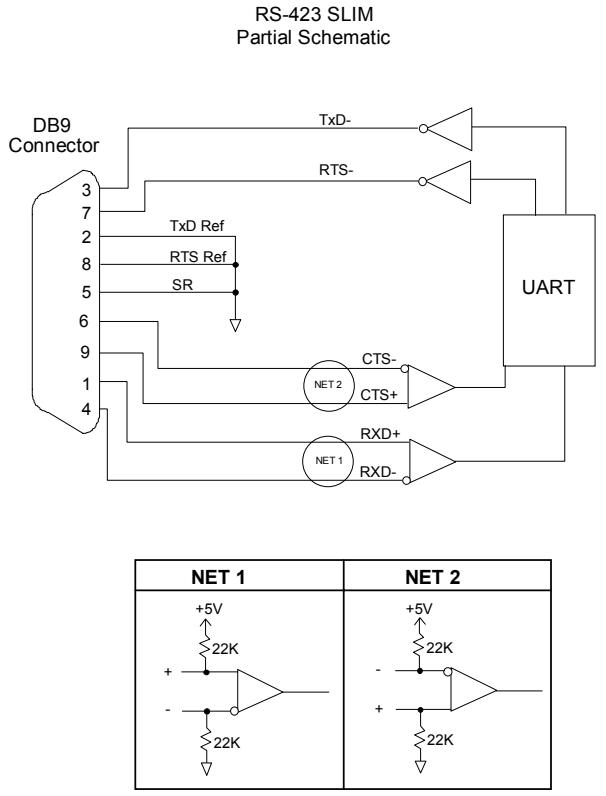
Technical Tip
The RS-485/422 electrical interface consists of a differential signaling scheme. ***You should always connect the signals with twisted pairs***

Appendix G: RS-423 Option

You may order the Intellicon-Flex8 adapter with RS-423 Serial Line Interfaces Module. The RS-423 electrical interface is a reliable high speed serial link that offers superior noise immunity.

Please refer to **Figure 53** for a partial schematic of the RS-423 SLIM circuit

Figure 53: *RS-423 SLIM: partial schematic*



Note: Note: *The RS-423 Serial Line Interface Module controls two ports, and therefore the schematic shows only a portion of the circuit.*

Installation

The Intellicon-Flex8 has four SLIM sockets which accept Serial Line Interface Modules. SLIM 1 socket (S1) interfaces ports 1and 2. SLIM 2 socket (S2) interfaces ports 3 and 4. SLIM 3 socket (S3) interfaces ports 5 and 6. SLIM 4 socket (S4) interfaces ports 7 and 8. Please refer to **Chapter 2: Hardware Installation** for the location of the SLIM sockets and to **Appendix D: SLIM Insertion/Removal** for the proper procedures to installing and removing SLIMs.

Your Intellicon adapter and Serial Line Interface Module (SLIM) are very sensitive to static electricity. Make sure that before you remove them from the anti-static shipping bag, you wear an anti-static wrist-band. When you remove them from the anti-static bag, handle them only by the edges and place them on the anti-static bag or an anti-static mat.

Please refer to **Table 31** for the pinouts for the SLIM sockets found on the Intellicon-Flex8 adapter.

Table 31: *SLIM socket pinouts*

SLIM Pin No.	TTL Signal	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	Reserved				
2		RTS 1	RTS A(-) 1	RTS (-) 1	TxD(-) 1
3		TxD 1	TxD A(-) 1	TxD (-) 1	TxD Source 1
4		CTS 1	RTS B(+) 1	RTS Ref 1	RxD(+) 1
5		RxD 1	TxD B(+) 1	TxD Ref 1	TxD(+) 1
6		Reserved	CTS A(-) 1	CTS A(-) 1	RxD Return 1
7		DTR 1	RxD A(-) 1	RxD A(-) 1	TxD Return 1
8		Reserved	CTS B(+) 1	CTS B(+) 1	RxD Source 1
9		DCD 1	RxD B(+) 1	RxD B(+) 1	RxD(-) 1
10		RTS 2	RTS A(-) 2	RTS (-) 2	TxD(-) 2
11		TxD 2	TxD A(-) 2	TxD (-) 2	TxD Source 2
12		CTS 2	RTS B(+) 2	RTS Ref 2	RxD(+) 2
13		RxD 2	TxD B(+) 2	TxD Ref 2	TxD(+) 2
14		Reserved	CTS A(-) 2	CTS A(-) 2	RxD Return 2
15		DTR 2	RxD A(-) 2	RxD A(-) 2	TxD Return 2
16		NC	CTS B(+) 2	CTS B(+) 2	RxD Source 2
17		DCD 2	RxD B(+) 2	RxD B(+) 2	RxD(-) 2
18	Ground				
19	+12 V				
20	-12 V				
21	+5 V				
22	DCD 2		Tied high	Tied high	Tied high
23	CTS 2				Tied high
24	RTS 2				
25	DTR 2				
26	TxD 2				
27	RxD 2				
28	RxD 1				
29	TxD 1				
30	DTR 1				
31	RTS 1				
32	CTS 1				Tied high
33	DCD 1		Tied high	Tied high	Tied high
34	TC				
35	RC				

Note: *earlier versions of the current loop SLIM have pins 22, 23, 32, and 33 tied low.*

RS-423 Cable Wiring

You can wire Intellicon-Flex8 adapters with the RS-423 option in various ways to communicate with RS-423 peripherals. This section will descibe a few examples of RS-423 cabling schemes.

Figure 54 describes a 4 wire cabling scheme between a port on the Intellicon-Flex8 to a port on the RS-423 peripheral.

Figure 54: *RS-423 wiring diagram(4 wire)*

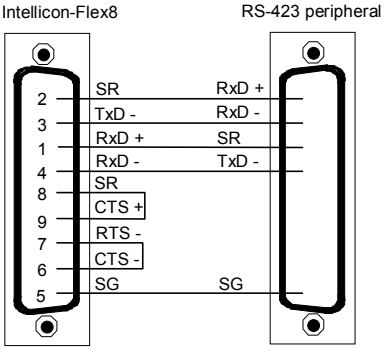
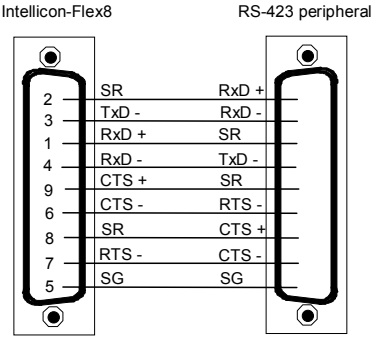


Figure 55 describes an all wire cabling scheme between a port on the Intellicon-Flex8 to a port on the RS-423 peripheral.

Figure 55: *RS-423 wiring diagram (all wire)*



Appendix H: 20mA Current Loop Option

You may order Intellicon-Flex8 adapters with 20mA current loop Serial Line Interface Modules (SLIM). The 20mA current loop electrical interface provides a reliable high speed serial link over long distances that offers superior noise immunity and multi-drop network connectivity.

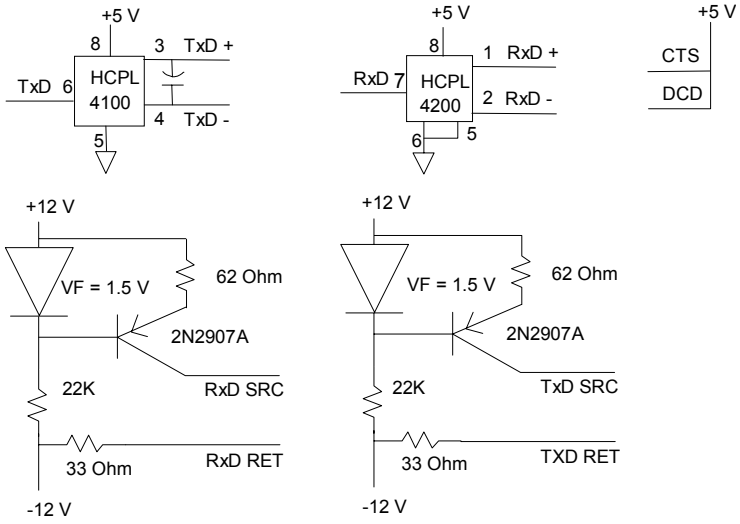
WARNING

You must insert Serial Line Interface Modules into all of the SLIM sockets before operating your Intellicon adapter. Failure to observe this precaution will result in damage to the Intellicon adapter.

***Note:**. The current source uses a compliance voltage of 24 volts, and the transmitter will withstand a 27 volt drop. Please refer to the Hewlett Packard Optoelectronics Manual/Catalogue for a complete description of HPCL 4100 and HPCL 4200 specifications.*

The 20mA current loop module provides optically isolated 20mA active or passive current loop interfaces for two ports. For each port there is an isolated receiver, an optically isolated transmitter and two 20mA current sources. The current source uses a compliance voltage of 24 volts, and the transmitter will withstand a 27 volt drop. See **Figure 56** for a partial schematic of the 20mA current loop module and **Table 32** for the pinouts of the SLIM module.

Figure 56: *20mA Current Loop Schematic*



Where:

SLIM Pin No.	Current Loop Signal	SLIM Pin No.	Current Loop Signal
2	TxD(-) 1	14	RxD Return 2
3	TxD Source 1	15	TxD Return 2
4	RxD(+) 1	16	RxD Source 2
5	TxD(+) 1	17	RxD(-) 2
6	RxD Return 1	18	Gnd
7	TxD Return 1	19	+12 V
8	RxD Source 1	20	-12 V
9	RxD(-) 1	21	+5 V
10	TxD(-) 2	22	DCD 2 (+5 V)
11	TxD Source 2	23	CTS 2 (+5 V)
12	RxD(+) 2	32	CTS 1 (+5 V)
13	TxD(+) 2	33	DCD 1 (+5 V)

To implement a passive interface, wire the transmitter and the receiver, while leaving the current sources unconnected.

To implement an active interface, connect a 20mA current source in series with the transmitter and another 20mA current source in series with the receiver. Although the you can operate the interface with only the receiver or the transmitter active, this could cause the voltage between the two loops to exceed safe limits.

When you use a current loop SLIM as a current source in the loop and the 20mA is flowing, the corresponding LED on the Serial Line Interface Module will light. If the loop is open or the current loop SLIM is not used as a current source the LED will be off.

WARNING

When using the current source in a loop with low resistance, the transistor will run hot. This can be avoided by increasing the loop resistance. The calculation for Maximum Loop Resistance(R_L) is:

$$\text{Max } R_L = \frac{22 - (S \times V_D)}{0.020}$$

Note:

1. *S = the number of stations in the loop (TX's and RX's)*
2. *V_D = the voltage drop for the HP opto couplers on the SLIM. V_D = 2.3 V for the HPCL 4100 and HPCL 4200. When you use other manufacturers'equipment in the loop the value of V_D may differ.*

Table 32: *SLIM socket pinouts*

SLIM Pin No.	TTL Signal	RS-232 Signal	RS-485 Signal	RS-423 Signal	Current Loop Signal
1	Reserved				
2		RTS 1	RTS A(-) 1	RTS (-) 1	TxD(-) 1
3		TxD 1	TxD A(-) 1	TxD (-) 1	TxD Source 1
4		CTS 1	RTS B(+) 1	RTS Ref 1	RxD(+) 1
5		RxD 1	TxD B(+) 1	TxD Ref 1	TxD(+) 1
6		Reserved	CTS A(-) 1	CTS A(-) 1	RxD Return 1
7		DTR 1	RxD A(-) 1	RxD A(-) 1	TxD Return 1
8		Reserved	CTS B(+) 1	CTS B(+) 1	RxD Source 1
9		DCD 1	RxD B(+) 1	RxD B(+) 1	RxD(-) 1
10		RTS 2	RTS A(-) 2	RTS (-) 2	TxD(-) 2
11		TxD 2	TxD A(-) 2	TxD (-) 2	TxD Source 2
12		CTS 2	RTS B(+) 2	RTS Ref 2	RxD(+) 2
13		RxD 2	TxD B(+) 2	TxD Ref 2	TxD(+) 2
14		Reserved	CTS A(-) 2	CTS A(-) 2	RxD Return 2
15		DTR 2	RxD A(-) 2	RxD A(-) 2	TxD Return 2
16		NC	CTS B(+) 2	CTS B(+) 2	RxD Source 2
17		DCD 2	RxD B(+) 2	RxD B(+) 2	RxD(-) 2
18	Ground				
19	+12 V				
20	-12 V				
21	+5 V				
22	DCD 2		Tied high	Tied high	Tied high
23	CTS 2				Tied high
24	RTS 2				
25	DTR 2				
26	TxD 2				
27	RxD 2				
28	RxD 1				
29	TxD 1				
30	DTR 1				
31	RTS 1				
32	CTS 1				Tied high
33	DCD 1		Tied high	Tied high	Tied high
34	TC				
35	RC				

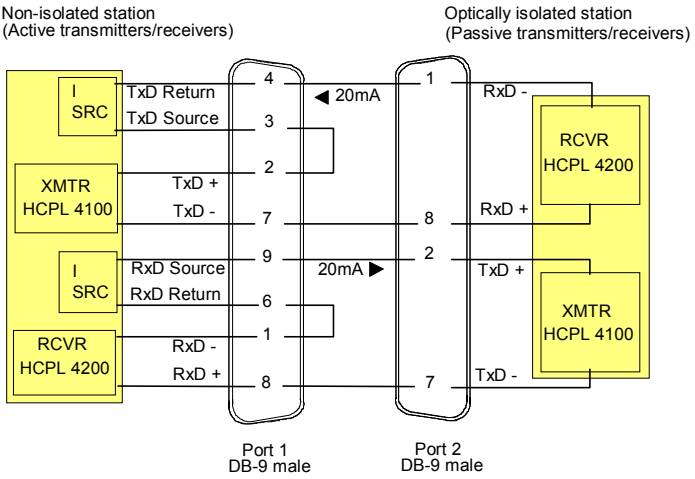
Note: *earlier versions of the current loop SLIM have pins 22, 23, 32, and 33 tied low.*

Current Loop Cable Wiring

You can wire Intellicon-Flex8 adapters with the 20mA Current Loop option in various ways to communicate with 20mA Current Loop peripherals. This section will descibe a few examples of current loop cabling schemes.

Figure 57 describes a 4 wire cabling scheme between a port on the Intellicon-Flex8 adapter to another port on the Intellicon-Flex8 adapter.

Figure 57: *Current Loop wiring diagram(4 wire)*



Note:
The example above illustrates a 20mA current loop cable wiring configuration between one port of an Intellicon adapter and another port of an Intellicon adapter. This example shows wiring for both active and passive modes