Converging Systems Inc.

CS-Bus Controllers

Installation, Programming and Interface Guide

-Intelligent Bus Translator (IBT-100)™

Version 1.5

Revision History

Date	Version	Description	Author
2/2/2009	1.0	Initial Draft	TS
6/15/2009	1.1	ZGN Update	TS
7/24/2009	1.2	Serial Command Update	TS
8/6/2009	1.3	VPad update	TS
9/25/2013	1.4	Update	AD
10/2/2012	1.5	Update baud rate	AD

Referenced Hardware/Firmware

Applicable Firmware	Version
IBT-100SFS	Version 1.3
ILC-100	Version 1.1
BSKP-2000	Version 1.1

CONTENTS

1 OVERVIEW

The Converging Systems' developed and manufactured Intelligent Bus Translator (IBT-100™) is a multi-purpose communication device for CS-Bus devices. This unique piece of hardware combined with its integrated firmware and optional ancillary software can be used to interface CS-Bus Controllers with external RS-232C networks to enable seamless bi-directional control and feedback.

General features of the Hardware are described within Section 1.1 below. General features of the Software embedded within the IBT-100™ are contained in Section 1.2 below.

Generic communication software as well as proprietary communication tools are described in Section 1.3 below.

1.1 SPECIFICATIONS OF THE HARDWARE

The IBT-100™ is designed to connect to a single CS-Bus Controller or an entire CS-Bus network (through a single CS-Bus Controller) using a simple interconnect cable. The IBT-100™ enables standard computers or serial communication controllers with an RS-232-C output port to interface easily with CS-Bus devices or networks. This architecture allows RS-232-C communication to occur over long wiring runs even though the typical limitation on RS-232-C serial runs is just 50 feet. With the IBT-100™, a remote CS-Bus network can be connected to a local serial controller even though (the last) CS-Bus device may be up to 4000 feet away from the controlling computer or serial controller. Generally, the IBT-100™ is then placed within fifty (50) feet of the serial controller and connected to that device with an RS-232-C cable. Then, the CS-Bus can initiate from the IBT-100™ and can continue for up to 4000 feet using standard CAT5 wire without repeaters, routers or other devices to one or more CS-Bus controllers.

Key features are as follows:

- CS-Bus interface port (RJ-25 6P6C female)
- RS-232-C connector (DB-9 female)
- Optically isolated CS-Bus from front-side RS-232-C serial port.
- Self-powered when connected to Powered-Bus Connector on CS-Bus controllers.

1.2 Specifications of the Embedded Software

Embedded firmware enables the IBT-100™ to interface with any CS-Bus Controller using simple ASCII commands.

More information on this command syntax can be found within the Appendix 1.

1.3 Description of Compatible Software

Virtually any serial communication software package can communicate to CS-Bus controllers through the IBT-100™ device. Some of the various packages available that can be used are listed below:

- Microsoft Window's HyperTerminal. This basic communication software package comes packaged with each copy of Microsoft Windows. This package is useful to test general functionality of serial communication. However, beyond initial testing, it is not recommended that this package be used for permanent CS-Bus control purposes because it its many limitations Further information on using HyperTerminal is contained within Section 3 below.
- CS-vPAD Application. This user-friendly Windows application is designed to program CS-Bus hardware over a serial interface for simple testing.

2 Setting Up the Hardware

In this section we will show you how to setup your IBT-100™ device. If you are planning on setting up the IBT-100, it will be necessary to follow the instructions in Section 2.1 in order to setup and connect your single CS-Bus Controller. If you are already familiar with the process skip and need to begin communicating with the product with software, proceed to the Section 2.2.

2.1 Setting up the IBT-100™

This section will take you through all of the steps required to set up an IBT-100™ device. These directions assume that you will choose the default RS-232-C serial communication settings as specified below.

Setting	
Baud Rate	57600
Parity	None
Data	8 bits
Stop	1 bit

First, connect a straight DB-9 (female) to DB-9 (male) serial cable from your computer or automation serial output device to the DB-9 connector on the IBT-100™. No NULL modem cable is required.

Next, connect CS-Bus interconnect wire from a powered CS-Bus port (Port 0, the left port on a CS-Bus device) to the RJ-25 (6P6C) connector on the IBT-100™. The IBT-100™ will not operate unless it receives appropriate external DC power from the controller. The left port on the CS-Bus devices is designed for this purpose.

Note: If you system utilizes another CS-Bus accessory such as a keypad that also receives its DC power over the same CS-Bus powered port (Port 0), it is necessary to find an available CS-Bus powered port on another CS-Bus device. You may also need to swap one or more CS-Bus interconnect wires connecting one CS-Bus device with another CS-Bus device in order to free up a CS-Bus Powered Port. You may connect two non CS-Bus Powered Ports on two separate CS-Bus devices to accomplish this requirement, but **you must not connect** two CS-Bus Powered Ports on two separate devices to each other.

2.2 Interfacing the IBT-100 with CS-Bus devices or a CS-bus network.

This section will take you through all of the steps required to communicate with one or more CS-Bus controllers through the IBT-100™.

1. Power up your CS-Bus Controller as described in the applicable CS-Bus Controller User Manual.

2. As soon as you power-up your computer or automation serial output controller, and provided the IBT-100™ is connected to a CS-Bus powered port, the device will begin operating.

3. You are now ready to launch your own serial communication software or one of the various compatible serial communication software packages available for the CS-Bus controllers. Proceed to one of the following sections related to external serial communication software for more information here.

3 MICROSOFT WINDOWS' HYPERTERMINAL

Minimum Requirements. The following minimum requirements need to be met in order to run the Microsoft Windows' HyperTerminal application:

■ PC platform that runs Windows 98 or later

■ Built-in RS-232c serial port or USB port with external USB-Serial adapter (generally devices with Prolific chipsets are the most compatible with standard serial protocols).

3.1 SIMPLE SETUP

This section will take you through all of the steps it takes to update communicate with the IBT-100™ to a CS-Bus network or device.

1. Launch the HyperTerminal computer program. You can find this program on your Windows system by following the below steps:

2. Select Program, select Accessories, select Communications, Select HyperTerminal.

3. In the Connection Description menu, type in IBT and select an icon (anything) and click OK:

4. In the Connect To menu, you will need to select an alternative other than the default of modem communication. You will need to know what communication port (COM port) to which your IBT-100™ is connected. If your device is connected to COM1 then, scroll through the various options until you find COM1. If your device is connected to COM2 or some other COM port, you must continue to scroll through the various options until you find the particular COM port to which your device is connected.

After you have set the correct COM port (i.e. 1,2,4,5, etc.), click OK

5. In the COM (n) properties menu, select 57600 bits per second, 8 data bits, None Parity, 1 Stop Bit, None Flow Control (important not to leave it set to the default of Hardware) and click OK.

6. In the IBT HyperTerminal window, type in the following command

&TYPE=?

Followed by a carriage return or enter.

You should receive a response back of the proper name of the IBT-100™. This demonstrates that bi-directional communication to the IBT is active and working properly. To further test; type a

#0.0.0.LED=ON; for ILC controllers

Or **#0.0.0.MOTOR=DOWN for Motor controllers.**

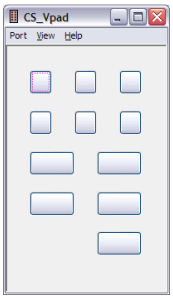
A message should be echoed back to the HyperTerminal indicating the message was correctly interpreted and passed onto the CS Bus. The respective controller should react.

7. You are now ready to begin testing and/or programming. Please refer to Appendix 1 for information regarding application commands and syntax.

4 CS-VPAD TESTER

This section will take you through all of the steps it takes to update communicate with the IBT-100™ to a CS-Bus network or device using the IBT Tester application.

1. Launch the CS-vPAD application. You should immediately see the below view. Basically, CSvPAD is a virtual keypad. Anything that your 11-button BSKP keypad can transmit onto the CS-Bus, so can your CS-vPAD.



2. Select the “Port” tab and set the appropriate COM port. All other communication parameters will have been automatically set so there is no need to make any other communication settings in your Control Panel or elsewhere.
3. Select the “View” tab and select the appropriate user interface. By default, the BSKPkeypad type device is selected. Other interfaces from time to time may be added.
4. Select any button on the display and the action associated with that button will be transmitted onto the CS-Bus. This is a great application to test your wiring, crimps and proper operation of your device.

APPENDIX 1-ZGN COMMANDS

The IBT-100™ supports commands in the format of Z(one).G(roup).N(ode) (hereinafter "ZGN" commands) to control devices on the CS-Bus.

More information on ZGN commands can be found in the associated manual entitled "CS-Bus Messaging."

SYNTAX

A command message is in the dot delimited format as:

Field	Format
Category	#
Address	Zone. Group. Node e.g. 2.1.1
Class	Device type e.g. LED
Item	Option used for configuring
Command	Operation to perform; e.g. ON

As an example a #2.1.1.LED=ON; will turn on the LED's on an ILC controller that has been set up with a zone of 2, a group of 1, and a node of 1.

Commissioning commands (those that start with a #UID address are not supported by the IBT. An e-Node is required.

SIMPLE COMMAND EXAMPLES

Command	Operation
#1.2.3.MOTOR=UP;	Motor at Zone=1, Group=2, Node=3 moves up.
#2.2.1.LED=FADE_UP	LED at Zone=2, Group=2, Node=1 fades up.
#0.0.0.LED=OFF;	All LED controllers turns off.
#2.0.0.LED=ON;	All LED in Zone 2 turn on.
#2.3.0.LED=RECALL,2;	All LED in Zone 2, Group 3 fade to the value of preset 2

SUPPORTED COMMANDS

For a complete list of all commands see the appropriate controller documentation. Below is a list of the most common commands but is not a complete list.

Device		Command	
Motor Control	.MOTOR	=DOWN	Moves motor down
		=UP	Moves motor up
		=STOP	Motor stops
LED Controller	.LED	=ON	LED on
		=OFF	LEDs off
		=FADE_UP	Fades up brightness
		=FADE_DOWN	Fades down brightness
		=SAT_UP	Increases saturation
		=SAT_DOWN	Decreases saturation
		=HUE_UP	Shits hue red->green->blue
		=HUE_DOWN	Shits hue red->blue->green
		=STOP	Stops the above action
		=SET,#	Sets brightness to value # (0-240)
		=SAT,#	Sets saturation to value # (0-240)
		=HUE,#	Sets hue to value # (0-240)
		=STORE,#	Stores current color in preset # (1-24)
		=RECALL,#	Recalls preset # (1-24)
		=EFFECT,#	Activates effect # (1.2)

APPENDIX 2 TECHNICAL INFORMATION-LOW VOLTAGE WIRING DIAGRAMS

The IBT-100™ connects to any RS-232C communication port and one or more CS-Bus controller devices or CS-Bus network. The RS-232-C serial wiring pin outs are defined within section A3.1 below. The CS-Bus interconnection wiring is defined within section A3.2 below. Refer to following diagram to understand location of Pin 1 on all connectors.

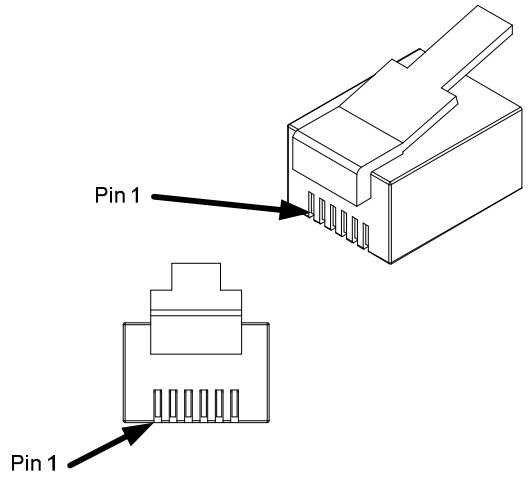


Figure 1

IBT-100SFS (RS-232C) TO REMOTE RS-232C CONTROL DEVICE OR SERIAL CONTROLLER

Typical wiring between a computer or a serial controller's RS-232-C output is described within this section. A ***straight*** DB-9 (male) to DB-9 (female) cable is used. A null modem is **NOT** required

DB9 IBT	DB9 COMPUTER	SIGNAL
1	1	**Not Required**
2	2	**PC Rx <- IBT (Tx)**
3	3	**PC Tx -> IBT (Rx)**
4	4	**Not Required**
5	5	**GND**

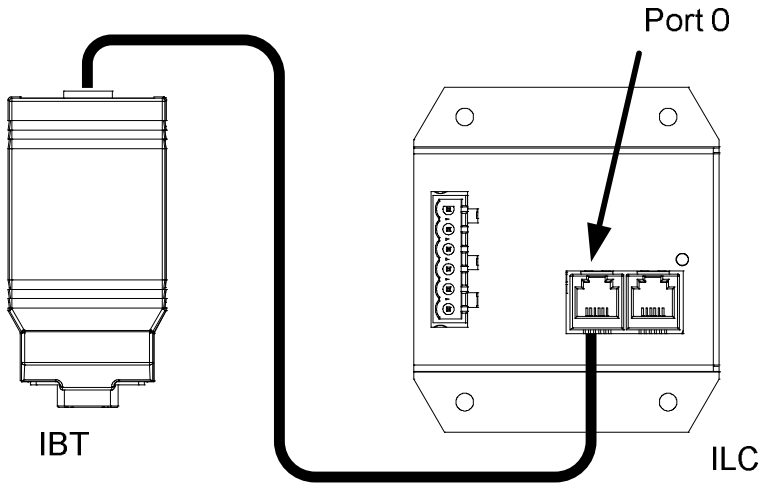
6	6	**Not Required**
7	7	**Not Required**
8	8	**Not Required**
9	9	**Not Required**

FIGURE 14

ILC-100 CONTROLLER TO IBT-100

A ***straight-thru*** 6-conductor wire (RJ-25 to RJ-25) should be used to connect the CS-Bus Powered Port (Port 0 - Left Port) on any one IMC-100 or ILC controller to the IBT-100™. It is typically advised that UTP (unshielded twisted pair) wire be used to insure noise immunity. It has been found the CAT5 wire is the best choice. While CAT5 has eight conductors, the BROWN/BROWN-White pair should be cut prior to the cable being terminated.

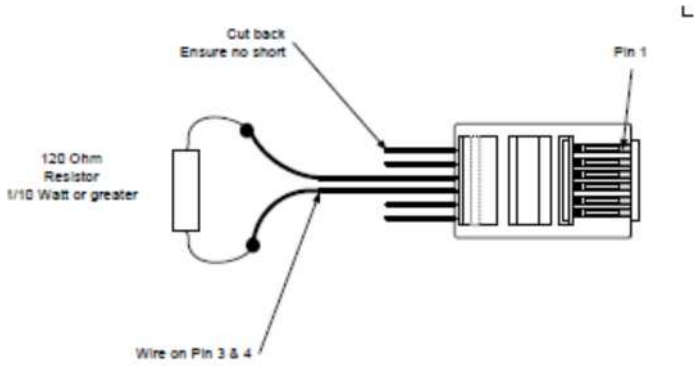
The IBT-100 is remotely powered from a CS-Bus Powered Port (Port 0) on a CS-Bus controller. That Powered Port is referred to as Port 0. See diagram below for location of Port 0.



RS-485 TERMINATORS

Applicability Note: RS-485 communication is designed around end-of-bus resistor termination. This guarantees error-free communication despite external noise and other sources of interference. It is highly advised that when more than one IMC-100x or ILC Controller is interconnected, one terminator is placed on one end of the bus and another terminator is placed on the other end of the bus. It does not matter into which CS-Bus port these terminators are connectors so long as they are at the very beginning of the bus and the very end of the bus. A BSKP Keypad has a built in optional termination. If the keypad is configured at the end of a line, it could be used as the termination by enabling the jumper (default) on the back of the keypad.

Note: Built into the IBT-100sfs is a termination resistor. Therefore, no additional termination is needed on the side of the CS-Bus connecting to the IBT-100.


Cut back
Ensure no short
Pin 1
120 Ohm
Resistor
1/10 Watt or greater
Wire on Pin 3 & 4

APPENDIX 3 SPECIFICATIONS

Dimensions: 3.25” (82.55mm) x 1.635” (41.53mm) x 0.875” (22.23mm)

Connectors:

-DB-9 female for RS-232C communications

-RJ25 (6P6C) female for CS-Bus communications

Power Requirements: 55 ma 5 or 9vdc provided through CS-Bus interconnect to Powered Port (Port 0 or left port) of CS-Bus Controller.