HW300
MDB/USB Interface

User Manual

Version 1.00

Revision History

Issue	Date	Comments
1.00	08/20/02	Preliminary

1.0 INTRODUCTION

The HW300 is a product built with an USB interface (output) and two independent serial ports (inputs) that interface with a PC Host system and different currency acceptors. The HW300 converts the interface of an MDB compatible coin changer/validator, to an easy-to-use USB language for the PC user.

1.1 Disclaimer

Universal Serial Bus (USB) is a trademark of USB Implementer Forum, Inc. Microsoft Windows 98, Windows Me, Windows 2000 are trademarks of Microsoft Corporation.. Multi-Drop Bus (MDB) is a voluntary standard proposed by National Automatic Merchandising Association.

Hemisphere West Technologies reserves the right to change the product or the product specifications at any time. While every effort has been made to ensure that the information in this publication is accurate, Hemisphere West Technologies disclaims any liability for any direct or indirect losses (howsoever caused) arising out of use or reliance on this product or information.

2.0 GENERAL DATA

Host - Refers to the application program or the computer system that handles the USB communication and interpretation of all the messages.

BNA - Refers to the bill acceptor.

MDB - Refers to the Multi-Drop Bus interface coin changer.

Enumeration - When a USB device is attached to the USB Bus, the Host will start an enumeration process. During this process, information and capability of the device is sent. The Host will determine its current resources and decide whether or not to configure the device as an active device and give it a resource handle. The factors that may prohibit a device from being configured include the data bandwidth requested by the device, the CPU usage of the system, the traffic already on the USB bus and the power budget etc. Only upon successful enumeration, the device is given a resource handle for application program to access. In MS Windows 9X/2K/Me, a registered USB device is shown under the *System Properties/Device Manager*.

Suspend mode - According to the USB 1.0 specification, all devices should enter a suspend (power saving) mode upon 3msec of inactivity of the USB bus. This includes the loss in communication with the Host or the Host Operation System disables the device etc. On exiting the Suspend Mode, HW300 will be reset by the inherent USB Bus reset. Any prior data stored in the HW300 will be lost.

2.1 Proprietary Data

Bus Address - According to the USB spec., up to 127 devices can be connected to the same USB bus on a single computer system. The attachment of the device should be done without user settings and be configured automatically by the Host Operation System. But in some situation, it is impractical or illogical to connect more than one device of the same type onto a single system. This limitation includes *fighting* between devices, sharing of resources or interference between them (For example: keyboard, mouse…).

The HW300 has the capability to identify itself among a multi-HW300 system. The HW300 utilities a unique Bus Address Setting to give itself a unique Bus Address for identification and for resources allocation. This Bus Address is a proprietary value in the product and will not interfere the normal USB operation. By using different Bus Address Settings, up to two HW300's can be attached on a single system and work independently. Upon successful enumerations, each HW300 is assigned with different resource handle. The application program should be able to retrieve the different handles from the Operating System and communicate with them. End user can identify each HW300 by observing to the jumper setting.

Sequence No. - In USB development, it is a common issue that the application program cannot *see* a real time message sent from the device. That is to say, all the messages sent from the device are stored to the bottom of the Host's buffer. Only by continual "popping" of the buffer will the latest message be able see by the program. This leads to an unsynchronised situation in the HW300 communications. In MS Windows 98 the buffer size has a fixed storage of 2 messages. In MS Windows 98SE/2K/Me, the default number is 8. User can change this number with *SetNumInputBuffers* register. To cope with this variance of Operating Systems, HW300 is designed to incorporate with a Sequence No. This number is added to the 8th byte of all messages. The Host initiates this number at 00h and increments on each successful message. The HW300 follows the same number in its message when sending data to the Host. In this way, the Host can keep track of the received message and continue popping the buffer until the latest message with the same Sequence No. as of the sent message appears. The Sequence No. rollovers at FFh and starts at 00h again.

WRITE Request - A standard request to send data from Host to HW300. It can be either an IN Packet or an OUT Packet.

IN Packet - A command only message sent from Host to HW300 to tell the HW300 to prepare data for the next READ Request. It contains only the Device ID and the Sequence No. The rest of the message is set to 00h.

OUT Packet - A command and data message sent from Host to HW300. It contains a Control byte, User data and a Sequence No.

User Data - The actual message wants to send to the Serial Device.

Device Data - The actual message gets from the Serial Device.

READ Request - A standard request to get data from HW300. The reception of data is done by hardware. Once DEV Packet is received from the HW300, user can fetch it out from the Host's buffer.

DEV Packet - (Device Packet) A command and data message sent from HW300 to Host. It contains a Control byte, Device data and a Sequence No.

Serial Device - HW300 can interface with two serial devices simultaneously. The serial devices are the BNA and MDB.

Device ID - Device ID exists in the Control byte of each IN/OUT/DEV Packet. It tells which Serial Device the message belongs to.

Central Buffer - Memory space inside HW300 for temporary storage of communication data.

3.0 COMMUNICATIONS

The HW300 conforms to USB 1.0 specification issued by the USB Implementers Forum. It is a low speed device (1.5Mbit/sec) supporting EndPoint 0 (Interrupt IN/OUT and Setup) and EndPoint 1 (Interrupt IN only) reports. When MF300 is attached to the USB bus, an enumeration process takes place automatically by the Host Operation System, usually on a PC. Upon successful enumeration, the HW300 is registered as a HID-class device under the Human Interface Device category specified by the USB specification and a unique resource handle is given.

The Host sends a request to the HW300 for data by using the resource handle. The HW300 answers to the request and conveys messages to/from serial devices. There are three communication types on the HW300 namely, the USB CONNECTION, the MDB CONNECTION and the BNA CONNECTION.

USB CONNECTION Specification: USB low speed, 1.5Mbit/sec.
MDB CONNECTION Specification: Optical Isolated, 9600bps, 9 bits, No Parity, 1 Start, 1 Stop.
BNA CONNECTION Specification: True RS-232, 4800bps, 7 Bits, Even Parity, 1 Start, 1 Stop.

3.1 Inside USB, Message Orientation

The Host sends messages to HW300 via WRITE Request. A WRITE Request can be an IN Packet or an OUT Packet. An IN Packet is a command message to tell the HW300 to prepare data for the next READ Request. Upon receiving an IN Packet, the HW300 will put all received Device data into packet(s) and be ready to send to the Host on the next coming READ Request(s). An OUT Packet is a data message to the HW300. It contains a Device ID to signify which Serial device this message is designating, control codes like MODE Bit, Clear Buffer Bit, More Packets Bit, and Num# of User Data it contains. In case of a long User Data, a multiple OUT Packets (WRITE Requests) are needed. More Packets Bit is set on all OUT Packets except the last one. Be sure not to set the Clear Buffer Bit after the first OUT Packet, otherwise the HW300 will clear all the data get from previous packets. The More Packets Bit is clear when all User Data is sent. This bit also tells the HW300 to start serial transmission and output all the User Data to the designated Serial Device.

The Host should give enough time between an OUT Packet and an IN Packet for the Serial Device to response. Once the time elapse is passed, the Host should send an IN Packet to HW300 followed by a READ Request. A DEV Packet will then be sent from the HW300 to the Host that contains the desired Device Data. User can fetch the data out by using Windows API Call: ReadFile. Each DEV Packet corresponds to each READ Request. If the Device Data is long and cannot be fitted into one DEV packet, multiple READ Requests and DEV Packets are needed.

3.2 Inside USB, Packet Format

WRITE Request, OUT Packet

Format: *1 Control Byte / 6 User Data Bytes / 1 Sequence No.*

Control Byte	Bit 6-7	Device ID	00= Not Used 01= MDB device 10= BNA device 11= Not Used
	Bit 5	Mode Bit	1= set MODE bit on MDB 0= clear MODE bit on MDB
	Bit 4	Clear Buffer	1= clear Central Buffer 0= do not clear Central Buffer
	Bit 3	More Bit	1= more packets coming, do not start serial transmission. 0= last packet, starts serial transmission.
	Bit 2-0	# of User Data	Binary format.
User Data	Byte 1-6		Actual data sending to Serial Device. Each packet limits to 6 User Data. If User data is more than 6 bytes, multiple packets are sent.
Sequence No.	Byte 1		Increments on each new message.

Example:
Sending a single Out Packet to MDB, 5 User Data bytes, set MODE bit, clear Buffer, Sequence No.= 1.

User Data (Hex)	04 56 E8 A4 2C
OUT Packet (Hex)	75 04 56 E8 A4 2C 00 01

Sending multiple Out Packets to BNA, 8 bytes of User Data, clear Buffer on 1st packet, Sequence No.= 5,6.

User Data (Hex)	04 56 E8 A4 2C 67 9A 10
1ST OUT Packet (Hex)	9E 04 56 E8 A4 2C 67 05
2ND OUT Packet (Hex)	82 9A 10 00 00 00 00 06

WRITE Request, IN Packet

Format: *1 Control Byte / 6 Zero Bytes / 1 Sequence No.*

Control Byte	Bit 6-7	Device ID	00= Not Used 01= MDB device 10= BNA device

			11= Not Used
	Bit 0-5	Not Used	Set to 0
Zero Byte	Byte 1-6	Not Used	Set to 00h
Sequence No.	Byte 1		Increments on each new message.

Example:
Sending an IN Packet to MDB, Sequence No.= 18.

IN Packet (Hex)	40 00 00 00 00 00 00 18

READ Request, DEV Packet

Format: 1 Control Byte / 6 Device Data Bytes / 1 Sequence No.

Control Byte	Bit 6-7	Device ID	00= Not Used 01= MDB device 10= BNA device 11= Not Used
	Bit 5	Mode Bit	1= MODE bit is set on MDB 0= MODE bit is not set on MDB
	Bit 4	Msg Corruption	1= message corrupted 0= message okay
	Bit 3	More Bit	1= more packets coming 0= last packet
	Bit 2-0	# of Device Data	Binary format.
Device Data	Byte 1-6		Actual data sending from Serial Device. Each packet limits to 6 Device Data. If Device data is more than 6 bytes, multiple READ Requests are needed.
Sequence No.	Byte 1		Follows the Sequence No. in the most recent IN packet.

Example:
Getting a single DEV Packet from MDB, 2 Device Bytes, MODE bit is set, Sequence No.= 23.

Device Data (Hex)	0B 0B
DEV Packet (Hex)	62 0B 0B 00 00 00 00 23

Getting multiple DEV Packets to BNA, 8 Device Bytes, Sequence No.= 6E, 70.
Note: Two READ Requests are needed for this operation.

Device Data (Hex)	02 31 5B 48 31 3E 03 2E
1st IN Packet (Hex)	80 00 00 00 00 00 00 6E
1st DEV Packet (Hex)	8E 02 31 5B 48 31 3E 6E
2nd IN Packet (Hex)	80 00 00 00 00 00 00 70
2nd DEV Packet (Hex)	82 03 2E 00 00 00 00 70

Refer to FIG. 1 for the illustration of using Requests and Packets.

3.3 Inside the HW300, Central Buffer

The HW300 has a buffer size of 60 bytes for central buffering. This Central Buffer is shared between the USB transmission and the two serial transmissions. If the Clear Buffer Bit is set in a WRITE Request, the HW300 will clear all its buffer content. In case of multi WRITE Requests, User Data from different messages is stored in the Central Buffer. Serial transmission will start upon receiving a More Bit=0.

When the HW300 detects an incoming message from a Serial Device (eg. MDB), HW300 will clear its Central Buffer, block communication from another Serial Device (eg. BNA) and store the Device Data into buffer. This piece of data will be resident in the buffer until the HW300 gets an IN Packet. If the Device ID of the IN Packet matches the Serial Device that sent the data, the HW300 will put the data into packet(s) and ready to send out on the coming READ Request.

4.0 ELECTRICAL HOOKUP

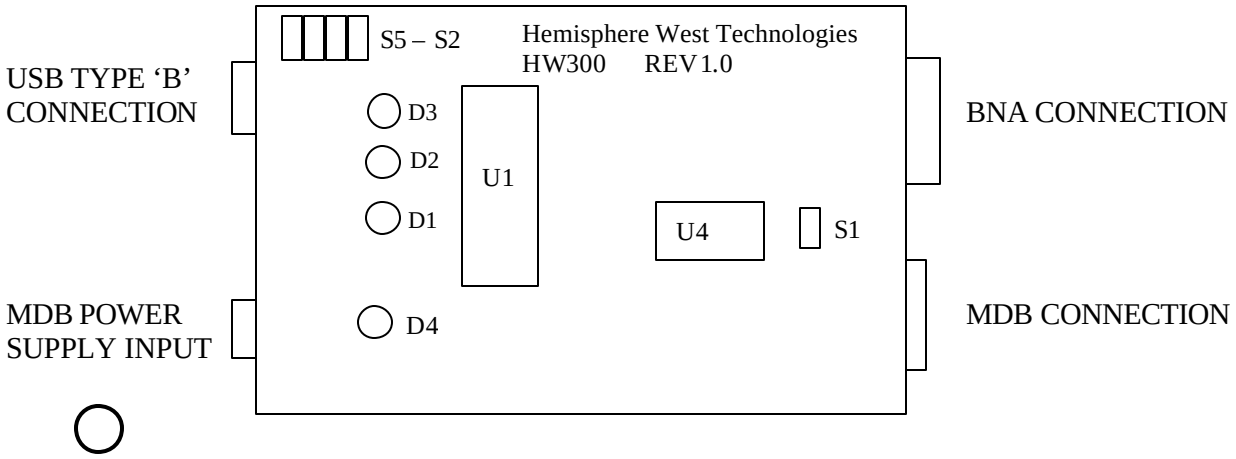

FIG. 2 HARDWARE LAYOUT

USB TYPE 'B' CONNECTION - For connection to computer USB port or USB Hub uplink port.

MDB POWER SUPPLY INPUT - An external power supply input for MDB device.
Power requirement: 34 VDC

BNA CONNECTION - A DB9 Male connector for connection with BNA.

MDB CONNECTION - A 6-pin connector for connection with MDB.

D1 - A Green LED to show communication between Host.

D2 - A Yellow LED to show communication between MDB.

D3 - A Yellow LED to show communication between BNA.

D4 - A Red LED to show presence of MDB Power.

S1 - Not used

S2 - Not used

S3 - Bus Address Jumper. OPEN: Address0, default
CLOSE: Address1

Note: Do not set more than one HW300 with the same Bus Address on a single system.

S4 - USB Port Power. To select power for digital part.
OPEN: AUX Power from MDB Power
CLOSE: USB Port Power, default

S5 - Join Ground. To join Digital GND with MDB GND.
OPEN: Isolated GND, default
CLOSE: Join digital GND with MDB GND

Note: This jumper is set only when S4 is OPEN.

U1 - Main Processor Chip.

U4 - Memory Extension Chip.

4.1 MDB CONNECTION configuration

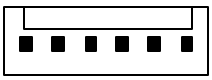

FIG. 3 MDB CONNECTOR, VIEWING TOWARDS

TYCO/AMP 640457-6

Pin 1 34VDC Power
Pin 2 DC Power Return
Pin 3 N/C
Pin 4 Master Receive
Pin 5 Master Transmit
Pin 6 Communication Common

4.2 MDB Bus cable

See FIG. 4 for MDB Bus cable drawing.

4.1 Jumper configuration

There are 5 jumpers on HW300. The default setting is shown in FIG. 5

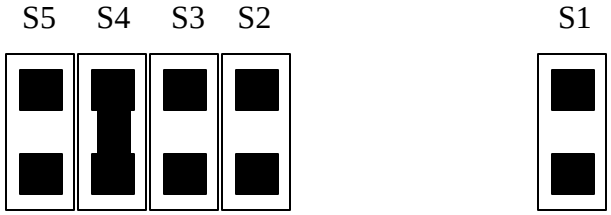

FIG. 5 DEFAULT JUMPER SETTING

Observe the following procedure when doing jumper settings.

1. Stop all USB communication on the Host.
2. Unplug the power cable to HW300.
3. Disconnect HW300 from the Hub.
4. Locate the desired jumpers on the circuit board.
5. Change the jumper setting according to your needs.
6. Reattach the power cable and USB cable.
7. Start the USB communication on the Host.