CVDM-2000

INNOVATIVE SOLUTIONS WIRELESS LINK

Wireless Link's CVDM-2000 mobile communication and tracking unit is the ideal product for vehicle and asset management. Designed to be the most comprehensive and inexpensive solution on the market, the CVDM-2000 integrates cellular (3-watt AMPS or IS-136) voice and data communications (circuit-switched and control channels), global positioning and an intelligent power management system into a compact single board module.

Unique to the CVDM-2000 is its open software architecture and dedicated function control processor (FCP). This gives OEMs the ability to develop and load their own custom or proprietary features adding to the already extensive feature list. Other existing features of the FCP allow for scheduling, logging and reporting of user selectable parameters (e.g. location history, speed, route, alarms, etc.) in addition to programming the abundant digital I/O and A/D converter inputs. The FCP also supports the power control system which provides numerous power modes and the capability to turn on and off specific portions of the module when needed — minimizing power consumption and optimizing battery efficiency.

The CVDM-2000 has a rugged switching power supply for harsh vehicle environments, a built-in battery charger and two optically isolated inputs to provide solid protection from noise and spikes. The CVDM-2000 also has up to 512 Kbytes of non-volatile flash memory for data logging, over-the-air programming for new software updates and a robust over-the-air data protocol for highly reliable message delivery.

DISTINCTIVE FEATURES

- Integrated cellular, GPS, and power management system
- Dedicated microcontroller for OEM custom features
- Real-time clock for scheduling and power management
- Numerous power modes
- Vehicle computer interface (RS-485)
- Data reporting and logging
- Event schedules and triggers
- Over-the-air flash programming
- Industrial switching power supply
- Built-in battery charger
- Optional 2-ampere hour backup battery
- Optional handset and/or headset with microphone for voice communications

WIRELESS LINK corporation

WWW.WIRELESS—LINK.COM

Wireless Link Corporation
CVDM-2000

CVDM-2000 User's Manual

This document describes the interface specifications for using CVDM-2000 in a product application.

Warning:

While this device is in operation, a separation distance of at least 50 centimeters (20 inches) should be maintained between the radiating antennas and the body of the user or nearby persons.

Caution:

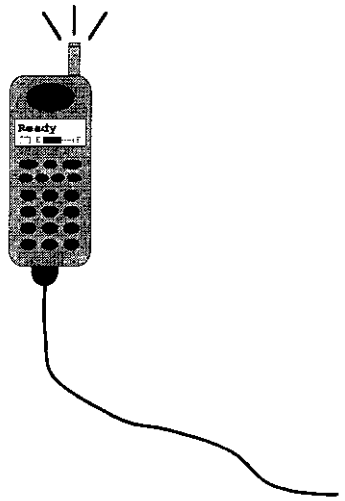
Changes or modifications not expressly approved by Wireless Link Corporation could void the user's authority to operate this equipment.

CVDM-2000 User's Manual

Warning: **While this device is in operation, a separation distance of at least 50 centimeters (20 inches) should be maintained between radiating antenna and the body of the user or nearby persons.**

This manual describes the interface specifications for using CVDM-2000 in a product. A serial packetized data interface (PDI) is defined to communicate with board.

WIRELESS LINK
corporation



WIRELESS LINK SOFTWARE INTERFACE SPECIFICATION

CVDM2000 MODULE "C" VERSION

Doc. Revision: 1.1
Doc. Date: 2/17/98

Revision History

10/21/97 Rev 1.0 J. Olson	Derived from WAP "B" specification. Differences are: - All references to MDCD changed to "CVDM2000 Module". - PDI was expanded to include much of the original WAP, as a result WAP is now a special mode of operation for the Hayes Modem option documented in "**Wireless Link Packet Data Interface Supplement, CVDM Hayes Modem Option**". The PDI protocol and standard messages are documented in "**Wireless Link Packet Data Interface, CVDM - All Versions**". The document "**Wireless Link Packet Data Interface Supplement, CVDM Real Time Clock Option**" describes the supplemental PDI messages that support the RTC option. The document "**Wireless Link Packet Data Interface Supplement, CVDM GPS Receiver Option**" describes the supplemental PDI messages that support the GPS option. This document pertains to the software interface extentions specific to the CVDM2000 Module.
2/16/98 Rev 1.1 J. Olson	Modifications to the following commands:

TABLE OF CONTENTS

1.0 HARDWARE INTERFACE TO CVDM2000 Module

The CVDM2000 Module firmware designed by Wireless Link supports a built-in Hayes compatible modem to allow control and monitoring via the AMPS cellular network. This Modem supports the standard CCITT V.22 bis (2400 bps), Bell 212A and V.22 (1200 bps), and Bell 103 (300 bps) data modes. Calls by the CVDM2000 Module originate and answer at the following settings:

Data Mode:	**Bell 212A (1200 bps)**
Baud Rate:	**1200**
Parity:	**None**
Stop Bits:	**1**

2.0 SOFTWARE INTERFACE TO CVDM2000 Module

The data format used to communicate to the CVDM2000 Module over the cellular network is the **Wireless Link Air Protocol**, or **WAP**.

With WAP, data is formatted in such a way that makes it easy to send and receive binary data to and from the CVDM2000 Module. This data is error checked via a 16-bit CRC-CCITT, frames are sequenced and acknowledged, and byte stuffing is utilized in order to always identify the start and end of a packet, even if bytes are missed. Also, because of the message structure it is a simple task to add custom packets needed for each application.

The next section describes the format of the WAP packet, which is followed by a summary of the standard WAP messages for the CVDM2000 Module.

2.1 INTERFACE DATA FORMAT - WAP

The following describes the Wireless Link Air Protocol (WAP) which is based on Wireless Link Corporation's Packetized Data Interface (PDI). This packetized interface was designed to use a minimum of code and data resources to handle the encoding and decoding, yet provide rugged error detection and reliable data transport.

WAP Data Protocol

Packet Format:

The following shows the format of each data packet that gets sent to and from the CVDM2000 Module. The '**DATA**' field is variable length which is indicated by the '**TYPE**' field. The CRC is a 16-bit cyclic redundancy check which checks all of the data in the packet starting with the SEQ and ending with the last character in the DATA.

1 byte	1 byte	1+ bytes	0+ bytes	2 bytes	1 byte
SOP	SEQ	TYPE	... DATA (may contain byte stuffing) ...	CRC	EOP

where:

SOP	=	Start Of Packet, which is always 'D1' hex
SEQ	=	Send/Ack sequence field
TYPE	=	message type that determines format of 'DATA' field
DATA	=	binary data appropriate for 'TYPE'
CRC	=	CRC-CCITT
EOP	=	End Of Packet, which is always 'DF' hex
STF	=	byte stuffing (DATA and SEQ/CRC fields only), which is always 'DE' hex

Sequence field:

The SEQ field is a single byte containing the two-bit send sequence number and the two-bit acknowledge sequence number. The bit definitions are as follows:

bits 7-4	bits 3-2	bits 1-0
0000	SEND SEQ	ACK SEQ

The **SEND SEQ** field is a two-bit sequence assigned to a send frame. This number is used by the frame recipient for two purposes:

1. Recognize a new frame.

2. Generate a new **ACK SEQ** number to send out as acknowledgment of receiving a valid frame.

The **ACK SEQ** field is a two-bit sequence number equal to the **SEND SEQ** number of the most recent, valid received frame. The **ACK SEQ** number is used to acknowledge successful reception of a frame.

When a connection is first established between the CVDM2000 Module and the data center, the sequence numbers are both reset to zero. This will provide positive acknowledgment when the first message is transmitted from either party. The sequence number for a new message is always incremented, thus the first message transmitted will have a send sequence number of one. A message will be transmitted continuously until a corresponding ack sequence number is received or a time-out occurs. If a time-out occurs, this is handled as a carrier failure and the carrier failure algorithm is invoked.

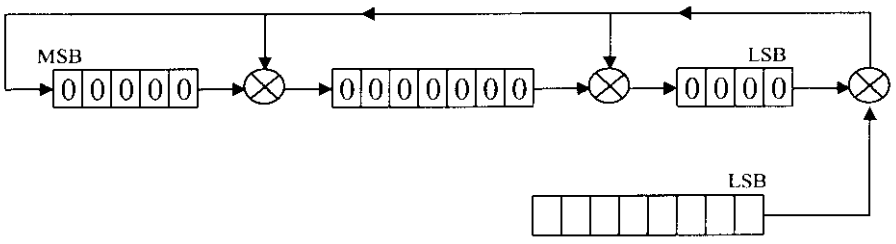
TYPE Field:

The **TYPE** field is a one or two byte field indicating the type of message. The first byte in this field is referred to as the "**primary type**". All messages have a primary type. Most message types also have a second byte in the **TYPE** field; this byte is referred to as the "**secondary type**". Often, a primary type is assigned to a category of message types (such as "Configuration" messages and the secondary type is used to define the which configuration parameters are being set/queried/reported.

CRC Field:

The **CRC** field is the 16-bit CRC-CCITT polynomial $X^{16}+X^{12}+X^5+1$ performed on all bytes in the packet starting with the sequence field and ending with the last DATA byte.

The "reversed" CRC-CCITT hardware circuit is implemented:



A method for computing the "reversed" 16-bit CRC-CCITT is shown below.

```
#define CRC_CITT      0x8408        // Reversed CCITT polynomial
```

```
unsigned short crc_ccitt(unsigned char b, unsigned short crc)
{
    unsigned short s;

    s = (unsigned short)b ^ (crc & 0xff);
    s = s ^ (s << 4);
    s = (crc >> 8) ^ (s << 8) ^ (s << 3) ^ (s >> 4);
    return (s);
}
```

The CRC for a message can then be calculated for a message as follows:

```
crc = crc_ccitt(msg[0], 0);      // initial crc calculation
for (index = 1; index < msg_length; index++)
{
    crc = crc_ccitt(msg[index],crc);
}
```

A faster run-time implementation is to generate a CRC look-up table as follows:

```
unsigned short crc_table[0x100];

for (index = 0; index < 0x100; index++)
{
    crc_table[index] = crc_ccitt(index, 0);
}
```

Then use the following routine instead of crc_ccitt( ):

```
unsigned short crc_ccitt_fast(unsigned char data_byte, unsigned short crc)
{
    unsigned short tbl_index;

    tbl_index = (crc & 0xff)  ^ ((unsigned short)data_byte);
    return ((crc >> 8) ^ crc_table[tbl_index]);
}
```

Byte Stuffing Process:

The **SEQ**, **DATA** and **CRC** fields contain byte stuffing to prevent a premature end of packet detection. The byte stuff code **STF** works as follows:

Packet Encoding: If **SOP**, **EOP**, or **STF** is contained in the data, then precede it with a **STF** code and include it in the checksum calculation.

Packet Decoding: If **STF** is received, do not include as part of **SEQ**, **DATA** or **CRC**. The next byte received, regardless of its value, becomes the actual **SEQ**, **DATA** or **CRC**. Both bytes are included in the checksum calculation. If **EOP** is receive without a preceding **STF**, then this is the end of the packet.