CHAPTER 3
WIRELESS MEASUREMENT INSTRUMENT

This chapter gives a functional description of the WMI hardware and software for implementation in IVDS. A detailed technical description is not given, but is provided in the WMI hardware and software manuals.[11][12]

3.1 Laboratory Configuration
As part of the laboratory configuration shown in Figure 3.1, the WMI receives test messages from the RF test transmitter, and then forwards the messages to the personal computer (PC) so that they can be displayed on the computer monitor. Additionally, the handheld Liquid Crystal Display (LCD) terminal can display test messages if desired. As shown in Figure 3.1, information flow is unidirectional from the transmitter to the WMI.

All the hardware connections are straightforward except for the PC serial communication ports. Communication Port 2 (COM 2) interfaces the WMI to the Microtec Monitor Program. This monitor program provides the capability to download programs into the WMI and to control the execution of the WMI application software. COM 1 interfaces the WMI to the Telemate communications program executing on the PC. Telemate was used configure the computer monitor for dumb terminal emulation so that test messages consisting of ASCII characters could be displayed.

Figure 3.1  Laboratory Configuration for IVDS System
3.2 Hardware
The WMI hardware has a main interface control board (ICB), a global positioning system (GPS) receiver, and a backplane which interfaces the ICB to a maximum of four receiver/decoder pairs. This hardware is contained in a metal housing with one antenna port for all four receivers, a GPS receiver port, a power port for 12V DC, two DB9 serial peripheral ports for RS-232 communications, an RJ-11 port for a handheld LCD terminal, two miniature DIN connector ports, a miniature fit connector port, and a data card slot.

The ICB hardware can be divided into five main sections: central processor unit (CPU), memory, ICB backplane, special purpose devices, and an ICB bus. Figure 3.2 shows the ICB hardware block diagram.

![Interface Control Board Block Diagram](image)

Figure 3.2 Interface Control Board Block Diagram
In the CPU section, the main processor is the Motorola 68306 microprocessor which executes the ICB kernel and application software. The secondary processors are the Motorola 6805s which communicate with the peripheral devices.

Memory is divided into three subsections: Read Only Memory (ROM), Random Access Memory (RAM), and Personal Computer Memory - Card Industry Association (PCM-CIA) card slot. ROM has 256 kilobytes (KB) of memory that is contained in two 128 KB electrically programmable read only memory (EPROM) circuit chips. The ROM contains both the kernel and application software. The even addresses of the software instructions are stored in one chip and the odd addresses are stored in the other. RAM has 1 megabyte (MB) of memory that is contained in eight 128 KB RAM chips. RAM stores all data needed by the kernel and the application software, and it may also be used to contain a RAM version of the kernel and application software. The PCM-CIA card slot provides the 68306 with read/write privileges to a RAM card. The battery backup provides power to the ICB RAM and to the real-time clock whenever power is removed from the ICB.

The ICB backplane has four decoder connectors, four receiver connectors, a GPS connector, and two 24-pin connectors directly connected to the ICB. Each installed decoder/receiver pair is connected to the backplane using a decoder connector and a receiver connector.

Since the ICB was designed to collect and process data from many different devices, various special purpose devices may be connected directly to the ICB via the backplane. The devices currently used are the following: the digital thermostat which turns on/off the WMI internal fans, the real-time clock (RTC) which maintains the current date/time with an accuracy of one second, the reference oscillator which provides a stable 14.4 megahertz (MHz) oscillator to all the receivers, the volume control device which controls the internal speaker volume, and the handheld LCD terminal.

As a multiprocessor bi-directional data bus, the ICB bus allows communication between the 68306 and a maximum of seven 6805 microcontrollers. The 6805s communicate with the special purpose devices and provide buffering between the 68306 and the decoder boards. The ICB bus is capable of transferring one byte at a time between the 68306 and any one of the 6805s.

3.3 Kernel Software
Grayson Electronics wrote the kernel software in the C programming language using the Microtec Research 68K Development System. This development system was used to debug and test the kernel software, and to convert the C code into compatible 68000 assembly language code. This code, once assembled, was burned into EPROM so that the kernel automatically executes upon power-up.
Basically, the kernel software is the WMI operating system software and executes in one of two modes. In mode one, the ROM kernel code accepts data packets from a PC and processes or re-routes the data packets as appropriate. In mode two, the kernel gives control of the 68306 to a separate program located in the ICB RAM.

The WMI kernel performs the following functions. Upon power-up, it reconfigures the WMI to start in the same kernel mode that was executing before power was removed. It restarts the application program in RAM if there was one executing before power was removed. It sends/receives all pending data to/from the peripherals connected to the ICB. It processes all commands received from the PC, and sends all peripheral data over serial port B if a PC is connected and has control of the WMI. It allows recovery from all 68306 exceptions, and lastly, keeps track of the current date and time. Figure 3.3 shows the WMI kernel program flow chart.

3.3.1 Kernel Memory Map
The Motorola 68306 has 4 gigabytes of directly addressable memory space. However, as shown in Figure 3.4, the WMI only uses a fraction of this addressable space. The memory space is divided into three main sections: RAM, ROM, and PCM-CIA card. The RAM section is addressable from 000000 hexadecimal (h) - 0FFFFFh for 1 MB of memory. The ROM section is addressable from 30000 - 33 FFFFh for 256 KB of memory. The PCM-CIA static RAM is addressable from 400000 - 5FFFFFh for 2 MB of memory.

RAM is further divided into eight subsections. The 68306 Vector memory is reserved for the 68306 exception vectors, which are used to set address locations for interrupt handling routines and for error handling routines. The Kernel Variables memory is reserved for the kernel global variables. The Kernel Jump Table memory is reserved for an array of function pointers that an application program can use to call a kernel function. The EPROM Kernel Data Space memory is reserved for the ROM kernel stack, heap, and static variables when the EPROM kernel is active. The RAM Kernel Code Space memory is reserved for the kernel executable code, constants, strings, and initial variable values. The RAM Kernel Data Space memory is reserved for the RAM kernel’s stack, heap, and static variables when the RAM kernel is active. The Application Code Space memory is reserved for the application executable code, constants, literals, strings, and initial variable values. The Application Data Space memory is reserved for the application stack, heap, and static variables when the application program is active.
initialize low level ICB hardware/software

initialize serial port B

disable power to all decoder boards

initialize kernel global variables

initialize kernel global variables

initialize jump table

kernel restart due to exception?

Y

send exception packet over serial port B

disable application (handheld) software

N

compute ROM checksum

RAM kernel last active?

Y

RAM checksum = last checksum?

Y

jump to RAM kernel entry point and run

N

initialize high level ICB hardware/software

process all devices

Y

application (handheld) program last active?

Y

application checksum = last checksum?

Y

jump to application entry point and run

N

N

Figure 3.3  WMI Kernel Program Flow Chart
Figure 3.4 WMI Kernel Memory Map
3.4 Application Software

The application software may be viewed as a top layer to the kernel software. The application program, in simple terms, calls the kernel functions to get decoder board messages, processes the messages, and then calls the kernel functions to transmit the messages to the Host subsystem. Figure 3.5 shows a simplified application software flow chart that executes after the kernel code initializes the WMI and the peripherals.

The application program continuously polls the decoder boards for messages. If a message is available, the application program then processes the message. This processing includes a Cyclic Redundancy Check (CRC) to validate messages, a duplicate message check for discarding repeated messages, and a message priority check for putting them into appropriate inbound queues. After processing, the messages are put into the outbound queue for transmission to the Host subsystem.

The application software routines to forward messages to the Host subsystem had not yet been written at the time the decoder board was being developed. To simulate messages being forwarded to the Host subsystem, the application software sent the messages to the Telemate communications program that was executing in the PC. Telemate configured the computer monitor as a dumb terminal so that the messages consisting of ASCII characters could be displayed on the monitor.

![Figure 3.5 Simplified Flow Chart for WMI Application Program](image-url)
3.5 Summary
In the laboratory configuration, the WMI receives test messages from the RF test transmitter and then sends these messages to a PC. The PC then displays the messages on the computer monitor. The PC simulates the operation of the Host subsystem and provides the means to determine if data from the CR is correct by displaying the data on the computer monitor.

The WMI contains the main processor of the CR and allows a maximum of four decoder boards to communicate with it. The kernel software is the WMI operating system software which provides all the necessary utility routines for which the user can use in the application software. The WMI application software performs three basic functions. The application software polls the decoder boards for available test messages, processes the received messages, and then forwards them to the Host subsystem. The processing includes a CRC check to validate messages, a duplicate message check for discarding repeated messages, and a message priority check for putting them into appropriate inbound queues.