MOTOROLA

Mobile Workstation 520™
Model F5203/F5205

Commercial, Government and
Industrial Solutions Sector

Application Developer's Guide

98-08901C31-A

COMPUTER SOFTWARE COPYRIGHTS

The Motorola products described in this instruction manual may include copyrighted Motorola computer programs stored in semiconductor memories or other media. Laws in the United States and other countries preserve for Motorola certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Motorola computer programs contained In the Motorola products described in this instruction manual may not be copied or reproduced in any manner without the express written permission of Motorola. Furthermore, the purchase of Motorola products shall not be deemed to grant either directly or by implication, estoppel or otherwise. Any license under the copyrights, patents or patent applications of Motorola, except for the normal non-exclusive, royalty-free license to use that arises by operation of law in the sale of a product.

EPS – 34440- B

This warranty applies within the fifty (50) United States, the District of Columbia and Canada.

LIMITED WARRANTY

MOTOROLA COMMUNICATION PRODUCTS

If the affected product is being purchased pursuant to a written Communications System Agreement signed by Motorola, the warranty contained in that written agreement will apply. Otherwise, the following warranty applies.

I. WHAT THIS WARRANTY COVERS AND FOR HOW LONG:

Motorola Inc. or, if applicable, Motorola Canada Limited ("Motorola") warrants the Motorola manufactured radio communications product, including original equipment crystal devices and channel elements ("Product"), against material defects in material and workmanship under normal use and service for a period of One (1) Year from the date of shipment.

Motorola, at its option, will at no charge either repair the Product (with new or reconditioned parts), replace it with the same or equivalent Product (using new or reconditioned Product), or refund the purchase price of the Product during the warranty period provided purchaser notifies Motorola according to the terms of this warranty. Repaired or replaced Product is warranted for the balance of the original applicable warranty period. All replaced parts of the Product shall become the property of Motorola.

This express limited warranty is extended by Motorola to the original end user purchaser purchasing the Product for purposes of leasing or for commercial, industrial, or governmental use only, and is not assignable or transferable to any other party. This is the complete warranty for the Product manufactured by Motorola. Motorola assumes no obligations or liability for additions or modifications to this warranty unless made in writing and signed by an officer of Motorola. Unless made in a separate written agreement between Motorola and the original end user purchaser, Motorola does not warrant the installation, maintenance or service of the Product.

Motorola cannot be responsible in any way for any ancillary equipment not furnished by Motorola which is attached to or used in connection with the Product, or for operation of the Product with any ancillary equipment, and all such equipment is expressly excluded from this warranty. Because each system which may use the Product is unique, Motorola disclaims liability for range, coverage, or operation of the system as a whole under this warranty.

II. GENERAL PROVISIONS:

This warranty sets forth the full extent of Motorola's responsibilities regarding the Product. Repair, replacement or refund of the purchase price, at Motorola's option, is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER EXPRESS WARRANTIES. MOTOROLA DISCLAIMS ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MOTOROLA BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS OR OTHER INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE SUCH PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW.

III. HOW TO GET WARRANTY SERVICE:

Purchaser must notify Motorola's representative or call Motorola's Customer Response Center at 1-800-247-2346 within the applicable warranty period for information regarding warranty service.

IV. WHAT THIS WARRANTY DOES NOT COVER:

A) Defects or damage resulting from use of the Product in other than its normal and customary manner.
B) Defects or damage from misuse, accident, water, or neglect.
C) Defects or damage from improper testing, operation, maintenance, installation, alteration, modification, or adjustment.
D) Breakage or damage to antennas unless caused directly by defects in material workmanship.
E) A Product subjected to unauthorized Product modifications, disassemblies or repairs (including, without limitation, the addition to the Product of non-Motorola supplied equipment) which adversely affect performance of the Product or interfere with Motorola's normal warranty inspection and testing of the Product to verify any warranty claim.
F) Product which has had the serial number removed or made illegible.
G) Batteries (they carry their own separate limited warranty).
H) Freight costs to the repair depot.
I) A Product which, due to illegal or unauthorized alteration of the software/firmware in the Product, does not function in accordance with Motorola's published specifications or with the FCC type acceptance labeling in effect for the Product at the time the Product was initially distributed from Motorola.
J) Scratches or other cosmetic damage to Product surfaces that does not affect the operation of the Product.
K) That the software in the Product will meet the purchaser's requirements or that the operation of the software will be uninterrupted or error-free.
L) Normal and customary wear and tear.
M) Non-Motorola manufactured equipment unless bearing a Motorola Part Number in the form of an alpha numeric number (i.e., TDE6030B).

V. GOVERNING LAW

In the case of a Product sold in the United States and Canada, this Warranty is governed by the laws of the State of Illinois and the Province of Ontario, respectively.

VI. PATENT AND SOFTWARE PROVISIONS:

Motorola will defend, at its own expense, any suit brought against the end user purchaser to the extent that it is based on a claim that the Product or its parts infringe a United States patent, and Motorola will pay those costs and damages finally awarded against the end user purchaser in any such suit which are attributable to any such claim, but such defense and payments are conditioned on the following:

A) that Motorola will be notified promptly in writing by such purchaser of any notice of such claim;
B) that Motorola will have sole control of the defense of such suit and all negotiations for its settlement or compromise; and
C) should the Product or its parts become, or in Motorola's opinion be likely to become, the subject of a claim of infringement of a United States patent, that such purchaser will permit Motorola, at its option and expense, either to procure for such purchaser the right to continue using the Product or its parts or to replace or modify the same so that it becomes non-infringing or to grant such purchaser a credit for the Product or its parts as depreciated and accept its return. The depreciation will be an equal amount per year over the lifetime of the Product or its parts as established by Motorola.

Motorola will have no liability with respect to any claim of patent infringement which is based upon the combination of the Product or its parts furnished hereunder with software, apparatus or devices not furnished by Motorola, nor will Motorola have any liability for the use of ancillary equipment or software not furnished by Motorola which is attached to or used in connection with the Product. The foregoing states the entire liability of Motorola with respect to infringement of patents by the Product or any its parts thereof.

Laws in the United States and other countries preserve for Motorola certain exclusive rights for copyrighted Motorola software such as the exclusive rights to reproduce in copies and distribute copies of such Motorola software. Motorola software may be used in only the Product in which the software was originally embodied and such software in such Product may not be replaced, copied, distributed, modified in any way, or used to produce any derivative thereof. No other use including, without limitation, alteration, modification, reproduction, distribution, or reverse engineering of such Motorola software or exercise of rights in such Motorola software is permitted. No license is granted by implication, estoppel or otherwise under Motorola patent rights or copyrights.

EPS – 48759 – O

FCC INTERFERENCE WARNING

The FCC Requires that manuals pertaining to Class A and Class B computing devices must contain warnings about possible interference with local residential radio and TV reception. This warning reads as follows:
NOTE: This equipment has been tested and found to comply with limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial or residential environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.

Trademarks

Contents

Commercial, Government and Industrial Solutions Sector
1301 E. Algonquin Road, Schaumburg, IL 60196

98-08901C31-A
June 2001

About This Guide

Scope

The Mobile Workstation 520™ (MW-520) is a standard mobile computer with a Pentium II or Pentium III ® processor which runs Microsoft Windows 98®, Windows NT® 4.0, Windows 2000®, or Windows Me (for model F5205 only). The MW-520 can include an embedded wireless network device (optional).

The application programmer can use any Windows programmer workbench. Typical programmer workbenches are:

- Visual C/C++®, Microsoft®
- Microsoft Foundation Class (MFC), Microsoft
- Visual Basic, Microsoft
- COM/DCOM, Microsoft

Disclaimer

The information included in this document is intended for planning application development. Despite our best efforts, some information may change. Motorola will try to keep you informed of any significant changes. This document carries no guarantee that the information contained herein is completely accurate.

Intended Audience

This manual is intended for Original Equipment Manufacturer (OEM) and end user software application designers and integrators who require knowledge of the product programming interface.

Related Documents

The following manuals provide additional information:

- *Mobile Workstation 520™ Model F5205 Owner's Manual*, 68P02962C10
- *Mobile Workstation 520™ Model F5205 Quick Reference Card*, 68P02962C11
- *Mobile Workstation 520™ Model F5205 Vehicle Installation Manual,* 68P02962C15
- *RPM500, Radio Portable Modem, Radio Service Software, User's Guide,* 68P02945C40
- *Software Specification – MultiKey/3434L Technical Reference* V 1.0 Phoenix Technologies, 1996.
- *APM BIOS Interface Specification*, Revision 1.2. Intel and Microsoft, 1993.
- *ACPI Specification* Revision 1.06, February 1999
- *Phoenix BIOS 4.0,* December 1998
- *Writing Windows Device Drivers*, Microsoft.
- *405i Communication Protocol, Native Mode* V 1.1.
- *Power Management Developers Reference*, PM SDR, Phoenix Technologies, 1995.

MW-520 Description

General

This chapter provides an introductory overview of the MW-520 workstation. Characteristics of the mechanical and electrical design and the basics of operation are briefly described.

When designing and optimizing applications for the MW-520, note the following:

- Although the application can be operated as a standard Windows application, it can also take advantage of services provided by the MW-520 enhanced Application Program Interface (API). These services are described in “Using Specific Hardware Features” on page 32.
- The design goal of the MW-520’s system software is to free the application from managing the MW-520 hardware. This allows applications to be developed on a desktop PC with more memory resources and a larger display. It also saves the programmer the effort of studying the hardware and the Basic Input/Output System (BIOS), and of developing the software needed to manage them. See “System Software” on page 9 for more information.

Specifications

Table 1
MW-520 Features & Specifications

Feature		Basic MW-520 Specifications
Processor		Intel® Pentium® II 333 MHz; Pentium III 500 MHz
Internal Memory		64 MB RAM expandable to 256 MB[1]
LCD		VGA or SVGA
	Resolution:	640 x 480 or 600 x 800
	Colors:	256,000 Colors
	Type:	Active Matrix
Communications/Expansion		
	Serial:	1 with 16550 UART support
	Parallel:	1 with ECP/EPP support
	Video:	Analog VGA
	PCMCIA Card Slots[2]:	Two Type II or One Type III Card Bus Slots
	USB Slot:	One
	IDB Slot:	One
Mass Storage		
	Hard Disk:	6 GB or larger
Keyboard		
	Main:	QWERTY, 84 keys total, 12 function keys, spill-resistant
	Pointing Device:	Integrated Touch-Pad
	Display:	6 illuminated function keys

Table 1
MW-520 Features & Specifications

Feature	Basic MW-520 Specifications
Radio Communications	Optional Private DataTAC, iDEN, CDPD
GPS	Optional Trimble II integrated device

1. Model dependent.
2. Supports 16 and 32 bit.

Specifications are subject to change without notice.

Data Exchange Network

The MW-520 typically communicates over a radio data network consisting of a central (host) computer that runs messaging and form-generation software, a radio network control processor, and remote base stations with a general communications controller.

This network configuration enables messages to be transmitted and received quickly and accurately, and ensures the availability of information when and where it is needed. This also means that the MW-520 user can download and edit forms located on the host computer, regardless of the user's location.

Through the use of several base stations, a single frequency can be reused for wide area coverage. This allows different messages to be transmitted simultaneously to different workstations.

Hardware Description

MW-520 is a terminal product for private and shared markets. The mobile device supports data entry via a keyboard, touchpad and touchscreen (optional).
It consists of three separate interconnected components: processor unit, display unit, and keyboard.

The Processor Unit

The Front Panel

The main power switch, the reset button and the PCMCIA card slot are located on the front panel of the processor.

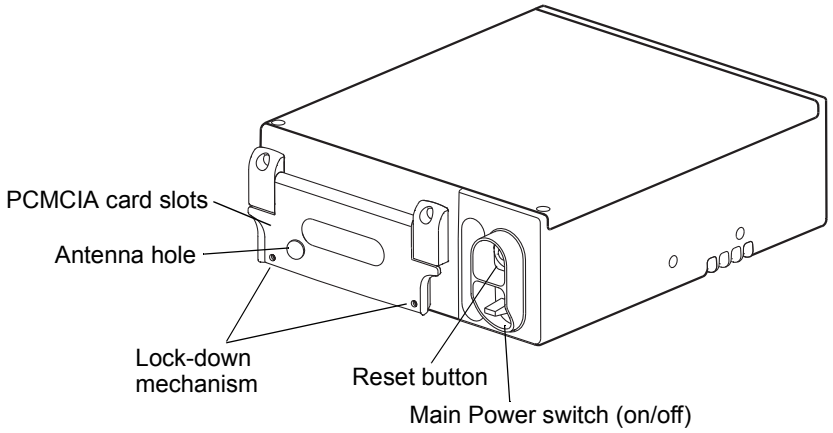


Figure 1
Front View of the Processor

Main Power Switch	Connects/disconnects the terminal power supply from the vehicle's battery.
Reset Button	Generates a hardware reset.
Card Bus slot	The MW-520 features a slot for installing two PC-Card (32-bit) Type II cards or one Type III card. A cover protects the PC-card slot against severe environment conditions. The Card Bus slot is also compatible with older PCMCIA cards (16 bit).
Lock-down mechanism	For security purposes, the door can be locked with two screws. The screws are provided together with an Allen key.

The Back Panel

All device and communication connectors are located on the back panel of the processor.

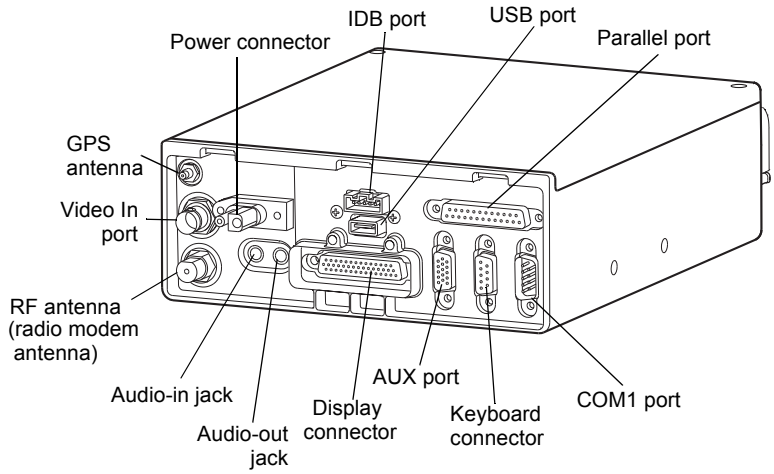


Figure 2
Processor Peripheral Ports

You can attach peripheral devices to these connectors, as follows:

Parallel port	A 25-pin port for parallel devices such as a parallel printer.
Serial port	A 9-pin port for serial devices such as a serial printer. *If the device has a 25-pin connector, use a 25-to-9 pin serial adapter.*
Audio connectors	An audio-in jack for an external microphone. An audio-out jack for an external 5W speaker.
Auxiliary port	A 15-pin connector for an external monitor. *Plug the interface cable of an external monitor into this port and then plug the monitor power cord into a grounded outlet.*
USB port	A port for USB devices such as a USB floppy disk.
IDB port	A port for connection of the in-vehicle bus.
Video In port	A port for connection of an external video source.

The Display Unit

The display unit is a color, active-matrix LCD. The display includes seven pre-defined buttons and six user-defined buttons.

The pre-defined buttons are:

- Power
- Suspend/Resume
- Backlight On/Off
- Brightness
- Volume On/Off (mute)
- Emergency

The Emergency button is located in the lower left corner of the display unit. It can only be used if the application supports its functionality.

The other buttons are located near the LCD, adjacent to the respective LEDs.

The Suspend button is a toggle switch.

In Suspend mode the display is turned off, and the electrical circuits enter a power saving mode. When MW-520 wakes-up from Suspend mode, the previous terminal state is restored (previous display, LED indications, etc.).

The display unit can use a touch screen (optional).

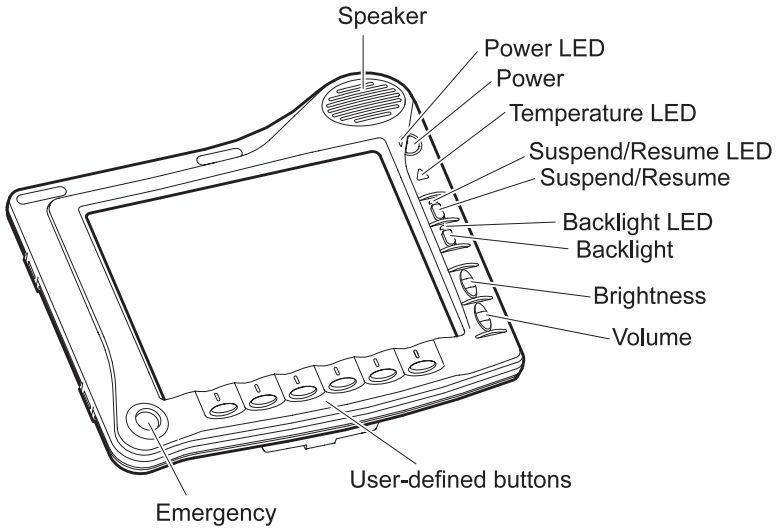


Figure 3
Display Unit

Table 2
Display Buttons Functionality

Control/Indicator	Function
Power	Turns the MW-520 on or off.
Power LED	Power on/off indication; Blinks when the battery is low.
Temperature LED	Extreme temperature conditions indication.
Suspend/Resume	Toggles between suspend and resume.
Suspend/Resume LED	Suspend indication.
Backlight	Backlight on/off. Adjust backlight level, using the Brightness button.
Backlight LED	Backlight on/off or faulty processor indication.
Brightness	Increases/decreases brightness level.
Volume	Increases/decreases speaker's volume level (if both buttons have been pressed, the sound will be muted or audible).
User-defined	Carries out a specific function, according to the specific user application.
Emergency	Sends an emergency message to the RF host computer. (This key's functionality depends on its definition in each user system and the active application.)

Battery

MW-520 is externally powered from the 12V vehicle battery. A Lithium backup battery maintains the system configuration and time.

Microphone, Sound Controller, and Speaker

One of the MW-520's main features is its integrated data and voice communication capability.

The workstation has an internal speaker located on the display unit. The speaker is used for various audio alert signals.

The processor unit includes a Windows compatible sound card and external microphone and speaker jacks.

The microphone and the speaker can also be used for future two-way voice communication, text-to-speech, and voice recognition applications.

System Software

BIOS

The MW-520 system setup program is Phoenix BIOS 4.06, developed by Phoenix Technologies Ltd. for mobile Pentium II/III based computers.

The BIOS enables running Windows® 98/2000/Me and Windows NT® 4.0 on the MW-520 platform. It also provides hardware initialization, system boot, power management, APM 1.2 or ACPI, Plug & Play registry data-base, and other features.

The BIOS setup is a program that configures the MW-520 hardware according to individual needs and saves the configuration into the CMOS memory (see *Mobile Workstation 520, Owner's Manual* for more information).

Operating System

MW-520 may run the following operating systems (depending on the model):

- Windows 98
- Windows NT 4.0
- Windows 2000
- Windows Me

A full set of `.CAB` files and other necessary software packages/drivers for these operating systems is saved under `C:\SETUP`.
Use this directory if you need additional drivers.

Windows 9x

To verify Plug & Play operation in the BIOS setup, enter the Advanced Setup menu, and set the `Installed O/S` option to `Windows` (default). To support special hardware features, Windows 9x includes additional files that were added by Motorola to support the workstation's special abilities.

When adding new applications to the hard drive, please be careful not to remove these files, as this can cause your system to malfunction.

To support I/O devices shared interrupts with Windows 9x, perform the following modification in the `SYSTEM.INI` file:

```
[386enh]
ComIrqSharing=true
```

HWAPP.INI file

This file is contained in the Windows 98 and Windows Me system software configurations. It contains different parameters that are used by the driver that handles the special hardware and software of the MW-520.

Table 3
HWAPP.INI Parameters

Parameter	Options	Description
`Logger_On`	1=Enable 0=Disable	Enables/disables the MW-520 logging utility
`Radio_On`	1=ON 0=OFF	Defines the internal radio modem status

If the application failed to read the `HWAPP.INI` file, the default parameters will be loaded. These parameters are:.

`Logger_On=0` Disable
`Radio_On=1` ON

Windows NT, Windows 2000

HWAPP Registry

Windows NT and Windows 2000 parameters are saved in the Registry data-base. The MW-520 Registry keys are located in the following registry folder:

In Windows NT:
`My Computer\Hkey_Local_Machine\Software\MW-520\HWDRIVER`

In Windows 2000:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mwserv\Parameters`

The HWDRIVER includes the following parameters:

Table 4
HWDRIVER Parameters

Parameter	Data	Description
`BatteryManagement`†	0×01	These values are displayed but cannot be effectively changed.
`TemperatureManagement`†	0×01	
`BatteryOffDelay`†	0×0b4	
`TemperatureOffDelay`†	0×0b4	
`SpeakerBarDelay`†	0×03	
`Radio_On`	0×01	Defines the internal radio modem status
`Logger_On`	0×01	Enables/disables the MW-520 logging utility
`InitOverTempDelay`*	0×014	Maximal acceptable duration of CCFL critical temperature (shutdown if time has elapsed)
`LastBrightness`*	0×0c	Brightness level of the last login
`LimitedLCDTemp1`*	0×03c	First upper limit for CCFL temperature (locking manual access in critical case)
`LimitedLCDTemp2`*	0×04b	Low bound for CCFL temperature (restoring manual access)
`NormalLCDTemp1`*	0×041	Second upper limit for CCFL temperature (shutdown if critical)
`NormalLCDTemp2`*	0×04b	Second upper limit for CCFL temperature (by default the same as LimitedLCDTemp2)
`CCFLTimerDelay`*	0×02	Delay for CCFL temperature check polling

Table 4
HWDRIVER Parameters

Parameter	Data	Description
`ShutdownTime*†`	0×03	Duration of shutdown process (in case of critical CCFL temperature). This value does not exist in Windows 2000.
`CCFLPollingEnable`	0×01	This parameter indicates whether the MWService module must perform MW-520 display temperature polling. It also enables/disables registration for brightness up and brightness down events. If the value of this parameter is 1, display temperature polling and registration for brightness events are enabled, otherwise, they are disabled.
`IgnitionOffEnable`	0×01	This parameter indicates whether registration for ignition power key off event is enabled. If the value of this parameter is 1, registration is enabled, otherwise, it is disabled.
`CurrentStateDelay`	0×078	This parameter is used by the MWDLL module only. When an emergency event occurs, MWDLL saves information about it in its internal variable. CurrentStateDelay is a period of time (in seconds) during which MWDLL keeps the emergency state since the last time emergency event occurred. If, during this period of time, a new client registers for an emergency event, MWDLL sends it a notification message about the emergency event. If CurrentStateDelay expires, MWDLL does not send emergency notification message to new registered clients.

Table 4
HWDRIVER Parameters

Parameter	Data	Description
`IgnitionOffShutdownTime`	0×03	IgnitionOffShutdownTime is a period of time (in minutes) during which MWService waits before shutting down in the case of an ignition power key off event. For example, if the value of this parameter is 4 and MWService receives notification about an ignition power key off event, it prints a message box on the screen, waits for four minutes and then shuts down MW-520.
`DisplayOverheatShutdownTime`	0×03	DisplayOverheatShutdownTime is a period of time (in minutes) during which MWService waits before shutting down in the case of display overheating. For example, if the value of this parameter is 5 and the temperature of the MW-520 display becomes critical, MWService prints a message box on the screen, waits for five minutes and then shuts down MW-520.
`IgnitionExclusiveRegistration`	0×00	This parameter enables clients to cancel handling of an ignition power off event by MWService. In the case that no other clients, except MWService, have registered for an ignition power event, MWService always handles the ignition. Otherwise (if another client has registered for an ignition power event), the value of IgnitionExclusiveRegistration is relevant: if the value is 0, MWService handles the ignition power event; if the value is 1, it does not.

* Relevant only for MW-520s with a 1000 Nit display.
† Values that do not exist in Windows 2000

If the application fails to read the HWDRIVER, the default parameters will be loaded.

PC Configuration

The MW-520 mobile workstation consists of three separate interconnected components: processor, display unit, and keyboard.

Figure 4 describes the MW-520 components and their interfaces.

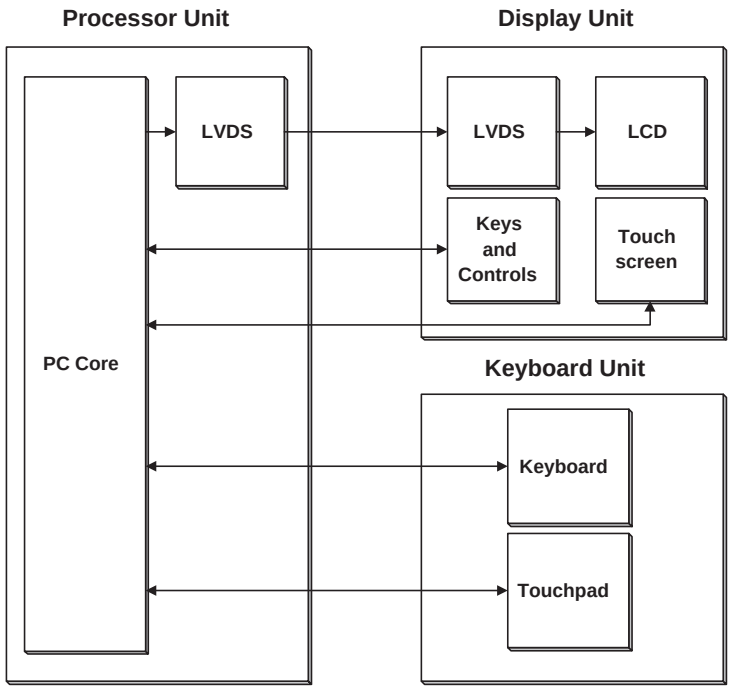


Figure 4
MW-520 Three Unit Model

The MW-520 PC architecture is illustrated in Figure 5 . It contains the devices listed below:

Processor	EMC-2 (Pentium II MMX 333 or Pentium III 500 MHz)
ChipSet	Intel 82371EB (443BX and PIIX4E)
Level 2 cache	512KB
SDRAM	64 MB (1 × 8M × 64 bit) 128 MB (1 × 16M × 64bit) 256 MB (1 × 256M × 64bit)
Keyboard controller	Hitachi H647343416
Display controller	69000 (CHIPS) + 2 MB RAM
PCMCIA controller	PCI1225 (Texas Instruments)
COM1, COM2	Super I/O National PC87338/PC97338
COM3, COM4	National PC16550 (2 items)
BIOS Flash	Intel flash 4 MB - E28F400B5-T60 (Intel)
Sound controller	Solo - 1E (ESS)
USB	(Part of PIIX4E)
Temperature controller	LM77
Touchscreen	Elo™ Resistive Digitizer
Video capturing	Enhanced Video Input Processor (EVIP) SAA7111A

The MW-520 computer is based on the Intel EMC-2 Module that includes a mobile Pentium II or Pentium III CPU chips, and 512KB second level cache.

PCI and ISA buses are used for peripheral device connection.

PCMCIA, VGA, IDE and sound controllers are connected to the PCI bus.

The following devices are connected to the ISA bus:

- the BIOS flash chip
- the keyboard controller
- the super I/O chip (two UARTs, parallel port and floppy disk controller)
- two auxiliary UARTs
- the touchscreen (via UART)
- the processor unit ASIC.

The MW-520 PC architecture complies with Microsoft PC99 requirements.

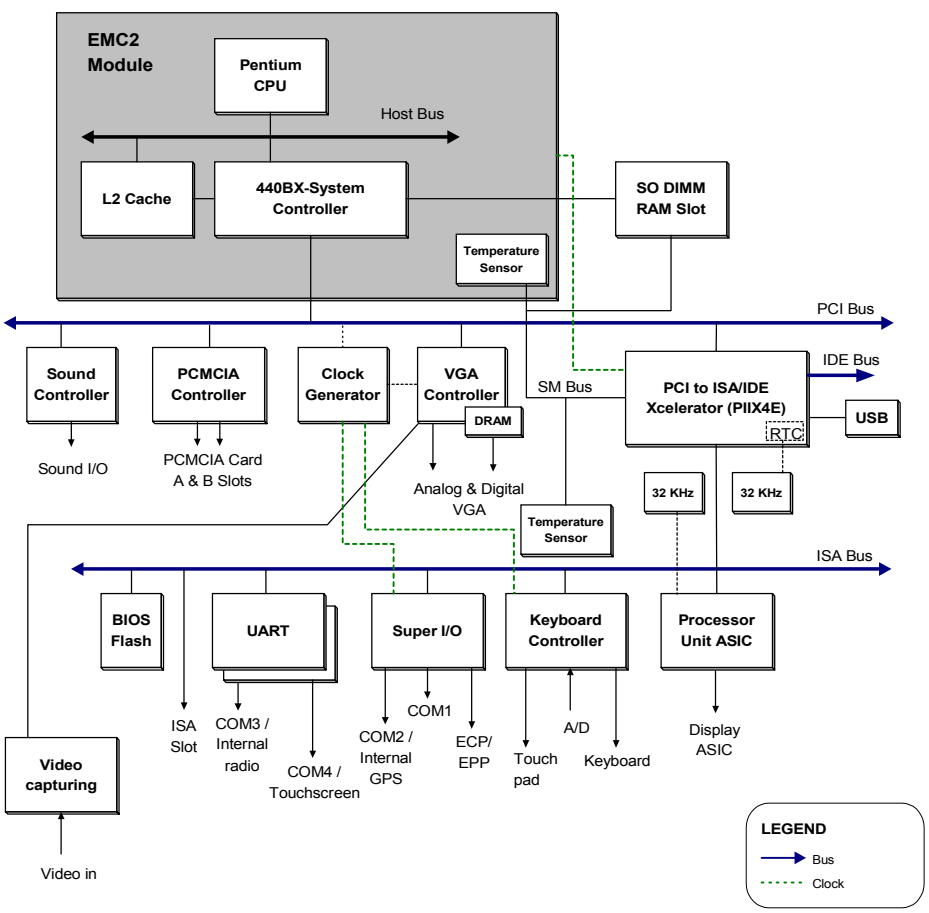


Figure 5
MW-520 PC Architecture

Logic Control and Addressing

This section includes maps of upper memory blocks, I/O addresses, and interrupts.

Upper Memory Blocks

Table 5
UMB Map

Address (hex)	Task	Comments
A0000-BFFFF	Video memory	
C0000-CAFFF	VGA BIOS	
CB000-CFFFF	PCM boot BIOS	
D0000-D7FFF	PCMCIA services	Used for power management
D8000-DFFFF	PCMCIA services	BIOS area in SMM mode
E0000-FFFFF	SYSTEM BIOS areas	
EA000-EBFFF	ESCD (PNP/DMI configuration)	Part of system BIOS area
EC000-EFFFF	BIOS boot block	Part of system BIOS area
F0000-FFFFF	BIOS	Part of system BIOS area

I/O Map

Table 6
I/O Map

Address (Hex)	Width	FUNCTION
0000-000F	8-bit	DMAC-1
0020	8-bit	PM2_CTRL register
0020-0021	8-bit	PIC-1
0040-0043	8-bit	TIMER
0048-004B	8-bit	TIMER
0060	8-bit	Reset X-bus, IRQ12/M and IRQ1
0061	8-bit	NMI status and control, speaker
0064	8-bit	Keyboard Controller
0070	8-bit	CMOS Index and NMI enable
0071	8-bit	RTC data
0078-0079	8-bit	Reserved
0080	8-bit	BIOS Port 80 [debug port]
0081-008F	8-bit	DMA1/DMA2 page register
00A0-00A1	8-bit	PIC-2
00B2-00B3	8-bit	Advanced PM control (PCI bus only)
00C0-00DE	8-bit	DMAC-2
00F0	8-bit	Co-processor error
0100-011F	8-bit	Processor unit ASIC PORTS
0170-0177	8-bit	Channel 2 IDE
01F0-01F7	8-bit	Channel 1 IDE
0200-0207	8-bit	Legacy joystick port (does not exist)
0220-022F	8-bit	Sound controller
0240-024F	8-bit	Sound controller
02E8-02EF	8-bit	COM4 [GPS]
02F8-02FF	8-bit	COM2
0300-0301	8-bit	Sound controller
0308-030F	8-bit	Free (but will still be locked by the BIOS)
0330-0335	8-bit	Sound controller
0378-037F	8-bit	Parallel port
0388-038D	8-bit	Sound controller
0398-0399	8-bit	Super I/O
03B4-03B5	16-bit	Video
03BA	16-bit	Video
03C0-03CA	8-bit	Video

Table 6
I/O Map (Continued)

Address (Hex)	Width	FUNCTION
03CC	8-bit	Video
03CE-03CF	8-bit	Video
03D4-03D5	8-bit	Video
03DA	8-bit	Video
03E0-03E1	8-bit	PCMCIA Controller
03E4-03E5	8-bit	PCMCIA Controller
03E8-03EF	8-bit	COM3 [Radio]
03F0-03F5	8-bit	Primary diskette controller (not used)
03F6	8-bit	Primary IDE channel command port
03F7	8-bit	Primary floppy channel command port
03F8-03FF	8-bit	COM1
04D0-04D1	8-bit	PIC-1/PIC-2 IRQ edge/level control
0CF8-0CFC	16-bit	PCI configuration space
0CF8	32-bit	CONFADD register (PCI Rg)
0CFC	32-bit	CONFDATA register (PCI Rg)
0CF9	8-bit	Reset control (PCI bus only)

For details of the 82431TX System Controller's register map, see [8a], pages 21-22.
For details of the 82371AB PCI-to-ISA/IDE XCelerator register map, see [8b], pages 43-46.

SMBus Interface

Table 7
SMBus Device Address

Device	Address
EMC (Thermal Sensor MMO (Intel))	0x9C
SDRAM Buffer	0000
Temperature Sensor (LM77)	0x90

Interrupts and DMA Channels Map

The devices use the Interrupts as defined in the following tables:.

Table 8
PCI INT Line Definitions

PCI IRQ	Devices
INT A	PCMCIA Controller (1) + Sound Controller (1)
INT B	PCMCIA Controller (2) + VGA Controller
INT C	Free
INT D	USB


Note

The PCI IRQs are allocated dynamically and may differ, depending on the operating system.

Table 9
Interrupt Map

IRQ	Source
IRQ 15	Processor unit ASIC
IRQ 14	IDE hard disk
IRQ 13	Floating point exception
IRQ 12	Mouse
IRQ 11	ISA PCMCIA cards
IRQ 10	PCI IRQ Resource
IRQ 9	Redirected IRQ2 + SMI
IRQ 8	RTC
IRQ 7	LPT 1
IRQ 6	Floppy drive (not used)
IRQ 5	PCI IRQ Resource
IRQ 4	COM1 (external), COM3 (radio)
IRQ 3	COM2 (GPS), COM4 (digitizer)
IRQ 2	Second internal interrupt controller
IRQ 1	Keyboard controller
IRQ 0	Timer


Caution

COM1/COM3 use IRQ4 and COM/2COM4 use IRQ3 in sharing mode. Please take this feature into consideration when making a high speed device connection.


Note

IRQ 5 and 10 will be used for PCI devices, and will be allocated dynamically. Usually, the sound controller and the card bus controller will be allocated to IRQ 5.

Table 10
Channel Map

Channel	Source
DMA 0	Free (can be used by PCI devices)
DMA 1	Parallel Port (ECP mode)
DMA 2	Free (can be used by PCI devices)
DMA 3	Free (can be used by PCI devices)
DMA 4	Direct memory access controller
DMA 5	Free (can be used by PCI devices)
DMA 6	Free (can be used by PCI devices)
DMA 7	Free (can be used by PCI devices)

Power Management

APM 1.2 System State Machine

The APM 1.2 System State Machine defines the states of all components in the system.

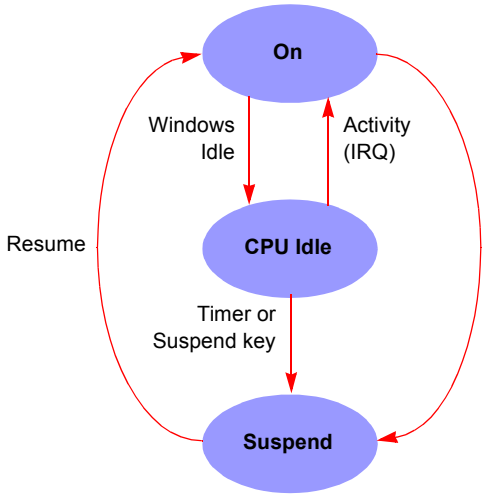


Figure 6
System State Machine

The machine contains three states: On, CPU Idle, and Suspend. All local *Standby* modes are canceled.

Table 11
State Definitions

Phoenix	Functionality
On	All devices on, full clock
CPU idle	All devices On, reduced clock
Suspend (to RAM)	Display and Disk power down Stop Clock, DRAM refresh mode
Suspend (to Disk)	Memory contents written to disk; power off

ACPI

The MW-520 supports ACPI for Windows 98/2000/Me, using a standard ACPI system with a number of extensions:

- At state S4, power is preserved for most devices.
  At state S5 the power is Off.
- Battery handling is similar to standard battery handling, providing a battery low notification. However, in case of a “Low” event, the battery power level is set to 100% or 5%.
- The MW-520 includes three temperature sensors designed to protect the system from overheating and causing the CPU to throttle. If the temperature rises above the determined “Suspend” level, the system is suspended (see below for more details).

Sleeping State

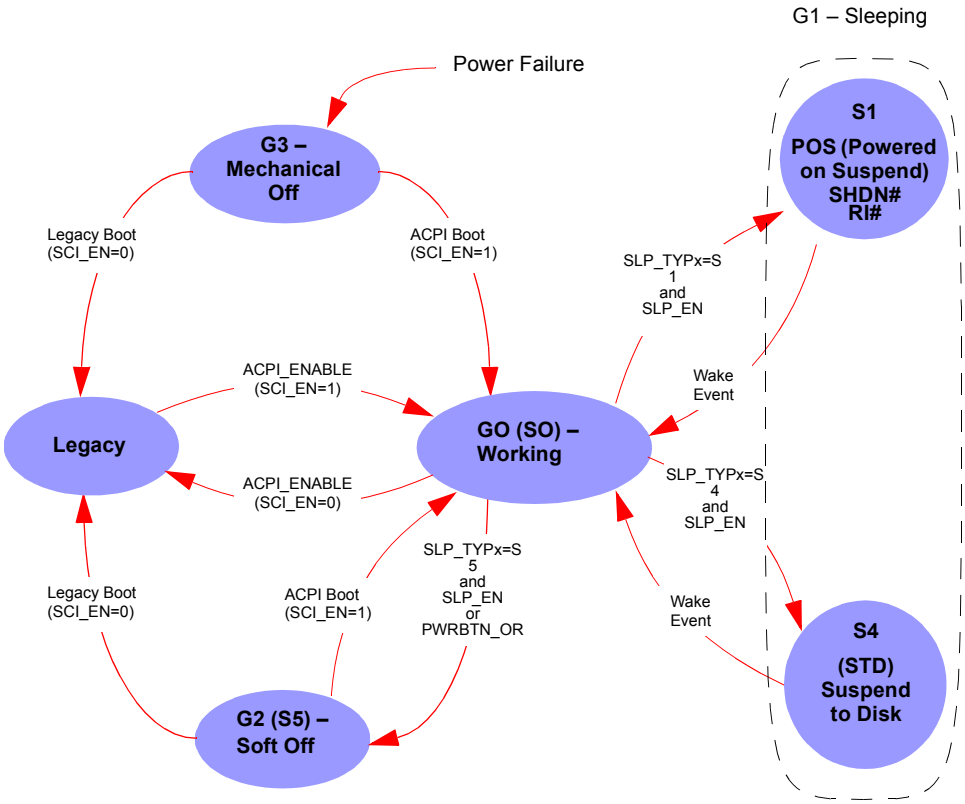


Figure 7
Sleeping State Chart

Table 12
Device States

Device/State	D0	D1	D2	D3	Controlled by
Hard Disk	ON (working state)*	Not supported	Not supported	Standby (hard disk stopped)	PIIX4E (IDE Controller)
VGA Controller	ON (working state)*	Not supported	Not supported	Device off + LVDS closed†	
Sound Controller	ON (working state)*	Not supported	Embedded DSP is halted & analog functions are off	D2 and oscillator off (PCI bus stopped)	
PCMCIA Card	ON (working state)*	Not supported	Device at intermediate state (RI_OUT available)	Off state (PCI bus stopped and no RI_OUT)	Card Bus Controller
Com 1	ON (working state)*	Not supported	Not supported	Closed (buffer 1 closed)	SHDN#1 Super I/O
Com 2	ON (working state)*	Not supported	Not supported	Closed (buffer 2 closed)	SHDN#1 Super I/O
Com 3	ON (working state)*	Not supported	Not supported	Not supported	–
Com 4	ON (working state)*	Not supported	Not supported	Not supported	–
LPT	ON (working state)*	Not supported	Not supported	Closed (Via Super I/O depend on Com 1&2)	Super I/O
USB	ON (working state)*	Not supported	Not supported	USB port suspended	PIIX4E (USB controller)
Radio Modem‡	ON (working state)*	Not supported	Not supported	Radio off	ASIC RPM_OFF bit
GPS	ON (working state)*	Not supported	Not supported	GPS off	ASIC GPS_OFF

* The device is open/closed depending on the OS decision (when not used with a specific device, it can be closed)
† The LVDS is closed by the hardware when the VGA controller is off.
‡ The radio powers on only when the application accesses COM 3.

Device/State	C0	Throttling*	C1	C2	C3	Controlled By
CPU	100% speed	50% Or 25%	Clock stopped	Not supported	Not supported	PIIX4E (stop clock)

* The throttling state is controlled by the PIIX4E and activated in hot conditions

MW-520 Temperature Control Mechanism

Model 5203/5205 contains a new temperature control mechanism, based on CPU internal clock throttling. The new mechanism is valid for all operating systems, and has several advantages:

- The clock-throttling enables reduction of the CPU heat in high temperature conditions. In most applications, the CPU performance degradation is negligible.
- The throttling thresholds are software-adjustable.
- At very high temperature conditions, the MW-520 will usually enter Suspend state, instead of shutting down. This enables the MW-520 to continue receiving messages even if, for instance, the unit was left working in a parked car on a very hot day.

Working Unit:

- At 38°C, the CPU frequency will be reduced to 50%.
- At 45°C, the CPU frequency will be reduced to 25% and the red temperature led will start blinking.
- At 55°C, the MW-520 will enter Suspend State.
- At 65°C, the MW-520 will shut down.

Power ON:

- At up to 55°C, the unit will start up at full speed.
- At up to 60°C, the unit will start up at 50% CPU frequency.
- At up to 65°C, the unit will start up at 25% CPU frequency.
- If powered on at 70°C, the unit will immediately enter Suspend state.
- The unit will not power on at 75°C or above.

Note

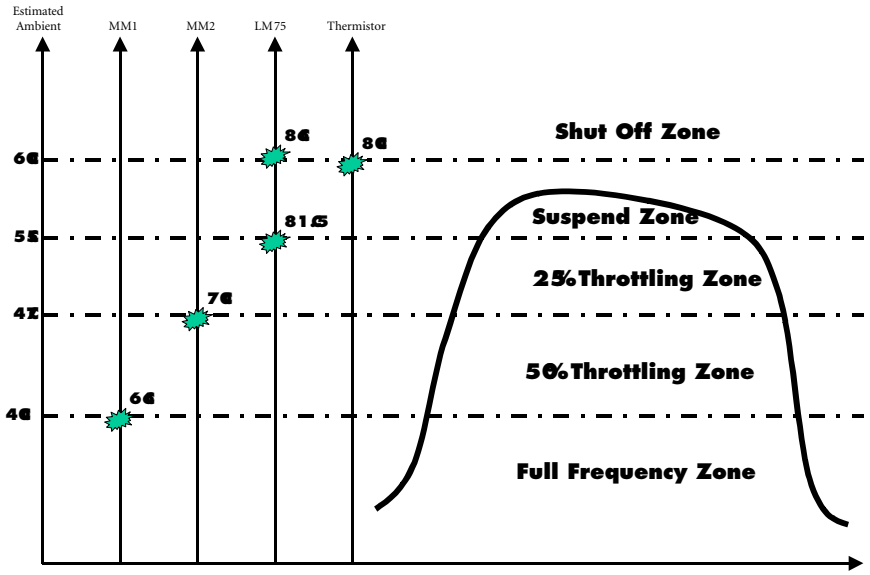
The temperature thresholds are defined according to the internal temperature, therefore the ambient temperature tolerance is ± 2C

Decisions regarding the CPU clock states are based on internal temperatures within the CPU housing, measured by four sensors inside the CPU housing:

- LM75 is a sensor located on the main board. This sensor is responsible for the Suspend threshold and for the software shut off sequence, and its level can be adjusted using the relevant software.
- MM1 is a sensor located on the EMC2 module (Pentium module). This sensor is responsible for the first 50% throttling, and its level can be adjusted using the relevant software.

- MM2 is a sensor located on the EMC2 module too. This sensor is responsible for the second 25% throttling, and its level can be adjusted using the relevant software.
- The thermistor located on the main board (identical to the sensor which was responsible for the temperature mechanism in previous MW-520 phases). This sensor is responsible for hardware shutdown (in case of software hang), and for the red temperature led activity. This sensor is not S/W adjustable.

The temperature control mechanism is described in the following chart:



Windows System Power Management Events

Any of the following events will resume the MW-520 from Suspend mode:

- Resume key pressed
- Key pressed
- COM1, COM3 ring indicator
- Touch screen display
- Emergency key pressed.

The Windows software informs the application about a power management event over a `WM_POWER` message. Use a `WM_POWER BROADCAST` message for 32-bit applications. Use a `WM_POWER` message for 16-bit applications.

WM_POWERBROADCAST

The `WM_POWERBROADCAST` message is sent by Windows to a 32-bit application to notify it of power-management events.

```
WM_PowerEvent = (DWORD) wParam;
dwData = (DWORD) 1Param;
```

Parameters

dwPowerEvent

Event notification message. This parameter can be one of the following values:

Value	Meaning
`PBT_APMBATTERYLOW`	Battery power is low.
`PBT_APMOEMEVENT`	OEM-defined event occurred.*
`PBT_APMPOWERSTATUSCHANGE`	Power status has changed.*
`PBT_APMQUERYSUSPEND`	Request for permission to suspend.
`PBT_APMQUERYSUSPENDFAIL`	Suspension request denied.
`PBT_APMRESUMECRITICAL`	Operation resuming after critical suspension.
`PBT_APMRESUMESUSPEND`	Operation resuming after suspension.
`PBT_APMSUSPEND`	System is suspending operation.

* Not supported by MW-520 BIOS and irrelevant to the application.

dwData

Function-specific data. For most messages, this parameter is reserved and not used. However, if *wParam* is one of the resume notifications (`PBT_RESUME`), the *lParam* parameter can specify the `PBTF_APMRESUMEFROMFAILURE` flag. This flag indicates that a suspend operation failed after the `PBT_APMSUSPEND` message was sent.

Return Values

Return TRUE to grant a request.

Request BROADCAST_QUERY_DENY to deny a request.

WM_POWER

The WM_POWER message is sent when the system is about to enter Suspend mode.

```
WM_POWER
fwPowerEvt = wParam; // power-event notification message
```

Parameters

fwPowerEvt

Value of *wParam*. Specifies a power-event notification message. This parameter can be one of the following values:

Value	Meaning
PWR_CRITICALRESUME	Indicates that the system is resuming operation after entering Suspend mode without first sending a PWR_SUSPENDREQUEST notification message to the application. An application should perform any necessary recovery actions.
PWR_SUSPENDREQUEST	Indicates that the system is about to enter Suspend mode.
PWR_SUSPENDRESUME	Indicates that the system is resuming operation after having entered Suspend mode normally, that is, the system sent a PWR_SUSPENDREQUEST notification message to the application before the system was suspended. An application should perform any necessary recovery actions.

Return Values

The value an application returns depends on the value of the *wParam* parameter. If *wParam* is PWR_SUSPENDREQUEST, the return value is PWR_FAIL to prevent the system from entering the suspended state; otherwise, it is PWR_OK. If *wParam* is PWR_SUSPENDRESUME or PWR_CRITICALRESUME, the return value is zero.

PBT_APMRESUMESUSPEND

The PBT_APMRESUMESUSPEND message is sent as a notification that the system has resumed operation after being suspended. A window receives this message through the WM_POWERBROADCAST message.

```
dwData = (DWORD) lParam;
```

Parameters

dwData

Reserved; must be 0.

Return Values

No return value.

Remarks

Applications may receive this message at any time without a preceding `PBT_APMSUSPEND` message.

PBT_APMSUSPEND

The `PBT_APMSUSPEND` is sent immediately before the computer is suspended. This message is typically sent when all applications and installable drivers have returned `TRUE` to a previous `PBT_APMQUERYSUSPEND` message. A window receives this message through the `WM_POWERBROADCAST` message.

```
dwData = (DWORD) lParam;
```

Parameters

dwData

Reserved; must be 0.

Return Values

No return value.

Remarks

An application should process this message by completing all tasks necessary to save data. This message may also be sent, without a prior `PBT_APMQUERYSUSPEND` message, if an application or device driver uses the **SetSystemPowerState** function to force suspension.

Note

Windows NT includes the APM2.0 utility, which adds APM support to the Windows NT system.

See Microsoft MSDN for additional information about Power Management events.

Suspend

The MW-520 supports two suspend modes: Suspend to RAM and Suspend to Disk.

In Suspend to RAM mode, the memory switches to "self refresh" mode, all system clocks are stopped, and all devices switch to StandBy.

In Suspend to Disk mode, the memory content and essential register values are written to the hard disk, and then the power is switched off.

When the system restarts, the BIOS knows that the last time the MW-520 was powered off this was in the course of a suspend to disk action. The necessary devices are initialized and the saved registers and memory content are restored. Resuming after "Suspend to Disk" takes longer than "Suspend to Ram", but the power consumption is much lower.

Executing the Suspend

From the Windows Start menu, select the Suspend button

or

Exit windows to suspend or Suspend time out

The system initiates the Suspend action. A progress bar indicates that the memory content is being written to the disk, and when completed, the computer is powered off.

PCMCIA Configuration

Configuring PCMCIA Cards in Windows NT

When you have Windows NT as your operating system, the Phoenix Card Executive for NT is automatically loaded onto your hard drive. The Card Executive configures the PCMCIA card properly each time it is inserted in MW-520.

It is recommended to insert Ethernet cards before running Windows NT and remove the cards after Windows shutdown. Card Executive enables Modems, ATA and SRAM cards “hot insertion” mode, i.e., inserting and removing the cards while Windows NT are running.

Whenever you insert a PCMCIA card in your workstation for the first time, the *PCMCIA IDE Driver Parameters* appears on the screen. To configure a PCMCIA card:

1. Set the IRQ Level parameter to `IRQ 11`.
2. Set the I/O Port Address to `Auto`.
3. Press *Continue.*
   A prompt for configuration completed appears on the screen.

Using Specific Hardware Features

Software Development Tools

Microsoft Visual C++ development kit was used by the MW-520 engineering team during product development. No special restrictions apply to the software development tools for its applications.

The Application Program Interface (API) library and *include* files are located on your hard disk as follows:

```
C:\HWDRIVER\HWEVNT.LIB (16-bit)   HWEVNT.H
C:\HWDRIVER\HWEVNT32.LIB (32-bit) HWEVNT32.H
```

Hardware Access API

User Application's Connection Mechanism

The MW-520 hardware access software contains an application, a DLL and a VxD driver. Figure 8 describes the hardware access software modules and their interfaces. `HWAPP.INI` is a system parameter text file.

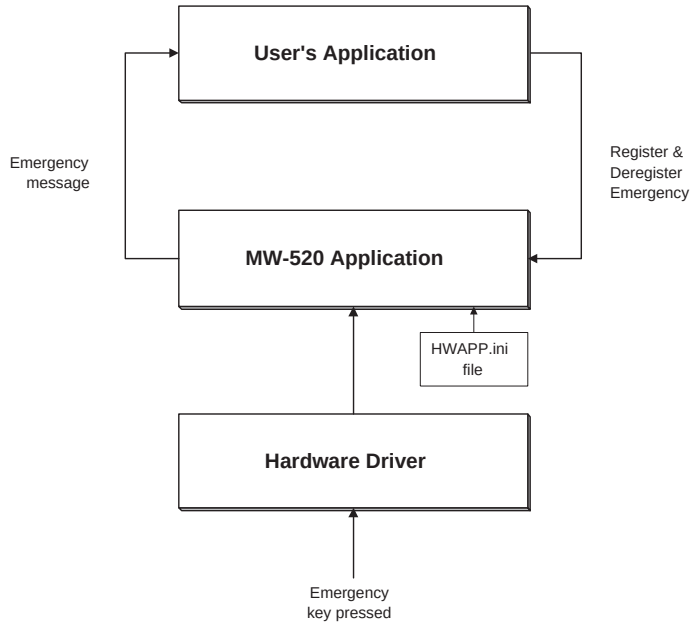


Figure 8
Hardware Access Software

When the Emergency key is pressed, an interrupt that modifies the MW-520 application is generated. The MW-520 application notifies the user's applications registered for an emergency event. The registration mechanism is described in the following section.

Emergency Event Registration

The Windows user application can be registered to receive the MW-520 emergency events. The registration process is as follows:

1. Activate the `Register_Emg` function call from the user's application (see API for details). Its output is the registration status.
2. If an emergency occurred, the message defined by the user's application is sent to it.

If the registered application is not activated, the `DeRegister_Emg` application should be activated.

Following is a C-code example of emergency registration/deregistration:

```
long FAR PASCAL_EXPORT Wnd Proc
    (HWND hwnd,
     UINT message
     UINT wParam
     UINT lParam
{
.
.
.
switch (message)
    {
    case WM_CREATE
         .
         .
         .
         Register_Emg (hwnd, MY_MSG_NUMBER);
         .
         .
         .
         break;
         case WM_DESTROY:
         .
         .
         .
         Deregister_Emg (hwnd);
         Break;
    }
}
```

Power Off Event Registration

The duration until the MW-520 is shut down may be used for several purposes, for example, sending de-registration messages by the messaging application. The registration process of the application is similar to the emergency registration.

1. Activate the `RegisterSysEvent` function call with the event parameter `POWER_OFF` (see API for details). Its output is the registration status.
2. If a `POWER_OFF` event occurs, the message defined by the user's application is sent to the application.

If the registered application is not activated, the `DeRegisterSysEvent` application should be activated before closing the user's application.

Note

With ACPI systems, the application should register to Windows Power Off event and not this event.

Hardware Access API

Register_Emg

`WORD APIENTRY Register_Emg` (HWND hwnd, UINT uiMsg) or
`WORD APIENTRY Register_Emg32` (HWND hwnd, UINT uiMsg)

Purpose Opens the emergency session.

Input hwnd: Window handle of the emergency session.
uiMsg: The message number used for notification.

Output None.

Returned Value 1 – ok, 0 – fail

DeRegister_Emg

`WORD APIENTRY DeRegister_Emg` (HWND hwnd) `Register_Emg` or
`WORD APIENTRY DeRegister_Emg32` (HWND hwnd) `Register_Emg`

Purpose Closes the emergency session.

Input hwnd: Window handle of the emergency session.

Output None.

Returned Value 1 – ok, 0 – fail

RegisterSysEvent

`WORD APIENTRY RegisterSysEvent` (HWND hwnd, UINT uiMsg, WORD event) or
`WORD APIENTRY RegisterSysEvent32` (HWND hwnd, UINT uiMsg, WORD event)

Purpose Registers the application to get notification for the system events.

Input hwnd: Specifies the handle of any user application.
uiMsg: The message number used for notification.
`EMERGENCY 0x0100`
`IGNITION 0x0040` (Windows 2000 only)

Output event, pointer to actual system event status.

Returned Value 1 – ok, 0 – fail

DeRegisterSysEvent

`WORD APIENTRY DeRegisterSysEvent` (HWND hwnd) or
`WORD APIENTRY DeRegisterSysEvent32` (HWND hwnd)

Purpose	De-registers the system event session.
Input	hwnd: specifies the handle of the user application.
Output	None.
Returned Value	1 – ok, 0 – fail

Driver Wide Communication API

ReadLCDTemp

`BOOL APIENTRY ReadLCDTemp` (float *temperature)

Purpose Reads the LCD temperature from LCD ASIC.

Input None

Output float *data. (volts)

Returned Value 1 – ok, 0 – fail

ReadBrightnessLevelStep

`BOOL APIENTRY ReadBrightnessLevelStep` (int *BrighnessLevel)

Purpose Reads the brightness level from LCD ASIC.

Input None

Output UINT *BrighnessLevel. (step)

Returned Value 1 – ok, 0 – fail

SetBrightnessLevelStep

`BOOL APIENTRY SetBrightnessLevelStep` (int delta_step)

Purpose Sets the brightness level from LCD ASIC.

Input int delta (range -32, +32) in steps

Output None

Returned Value 1 – ok, 0 – fail

SetBrightnessLock

`BOOL APIENTRY SetBrightnessLock` (BOOL value)

Purpose Sets the brightness control in LCD ASIC

Input BOOL value : TRUE – lock, FALSE – unlock

Output None

ReadBrightnessLock

`BOOL APIENTRY ReadBrightnessLock` (void)

Purpose	Reads the brightness lock value in LCD ASIC.
Input	BOOL value : TRUE – lock, FALSE – unlock
Output	None
Returned Value	BOOL value : TRUE – locked, FALSE – unlocked

ReadLCDVersion

`BOOL APIENTRY ReadLCDVersion` (DWORD *data)

Purpose	Reads the LCD version LCD ASIC.
Input	None
Output	DWORD *data (version number)
Returned Value	1 – ok, 0 – fail

General Purpose Lines Control API


Note

The following API refers to the MW-520 hardware driver from versions 7.0.0.15 (Windows 2000), and 6.0.5.98 (Windows 98 and Windows Me) only.


Note

GPIO Line 3 is an internal line (it does not connect to the auxiliary port).

ReadGPIOLine

`BOOL APIENTRY ReadGPIOLine` (PBYTE pValue, BYTE Line, BOOL status)

Purpose Reads the state of a single GPIO line.

Input [out] PBYTE pValue - pointer to the byte variable in which to place the read state of the GPIO line. Legal values for this parameter are 0 and 1.

[in] BYTE Line - number of the GPIO line whose state is to be read. Legal values for this parameter are 0, 1, 2 and 3.

[in] BOOL status - set line direction to read if status=0

Output The value level of the requested line.

Returned Value BOOL value: TRUE - success, FALSE - failure

WriteGPIOLine

`BOOL APIENTRY WriteGPIOLine` (BYTE Value, BYTE Line)

Purpose Sets the state of a single GPIO line.

Input [in] BYTE Value - the byte value that indicates a new state for the GPIO line. Legal values for this parameter are 0 and 1.

[in] BYTE Line - number of the GPIO line whose state is to be set. Legal values for this parameter are 0, 1, 2 and 3.

Output None

Returned Value BOOL value: TRUE - success, FALSE - failure

ReadGPIO

`BOOL APIENTRY ReadGPIO` (PBYTE pValue, BYTE status)

Purpose Reads the state of an entire GPIO (all four GPIO lines).

Input [out] PBYTE pValue - pointer to the byte variable in which to place the read value of the GPIO state. The returned value will be placed into the four least significant bits of this variable; the four most significant bits are unused.

[in] BYTE status - the byte value for four-direction (input/output) bits of GPIO register. For each of these bits, the value 0 means input (reading from GPIO line) and the value 1 means output (writing to GPIO line). For each ReadGPIO operation the status bits can be set to the required direction. (For example, the value 0x3 indicates that GPIO lines 0 and 1 will be set to output and GPIO lines 2 and 3 will be set to input.) Legal values for this parameter are 0-15.

Output The value level of all four lines.

Returned Value BOOL value: TRUE - success, FALSE - failure

WriteGPIO

`BOOL APIENTRY WriteGPIO` (BYTE Value)

Purpose Sets the state of an entire GPIO (all four GPIO lines).

Input [in] BYTE Value - the byte value that indicates a new state for the GPIO. Legal values for this parameter are 0-15.

Output None

Returned Value BOOL value: TRUE - success, FALSE - failure

Resuming the Display

This section describes how to interrupt the screen saver or activate the display in the event that an emergency message is received. The message is then displayed on screen.

The following instructions use the API function `keybd_event()` to artificially terminate the screen saver. An MSDN-recommended practice is described in the technical articles Q140723 and Q150785.

```
VOID keybd_event(BYTE, BYTE, DWORD, ULONG_PTR)
```

The `keybd_event` function emulates a keystroke. It can be used to generate a `WM_KEYUP` or `WM_KEYDOWN` message, which imitates key pressing/releasing for Windows, thereby terminating the screen saver.

This function has four parameters, of which only the first and third are used here. The first parameter is a virtual-key code and the third parameter indicates whether the key is pressed or released. By default, the third parameter is zero, which means Key Down.

The following is an example of the use of the `keybd_event ()` function:

```
keybd_event(VK_SHIFT, 0, 0, 0);

keybd_event(VK_SHIFT, 0, KEYEVENTF_KEYUP, 0);
```

Note

The second call to `keybd_event` is required to indicate to the operating system that Shift key is no longer pressed.

User-Defined Buttons

The six user-defined buttons located on the display unit use scan codes `64h` through `69h`, from left to right. These scan codes are not used by any other keys on the keyboard.

Main Peripherals

General

The following peripheral devices may be attached to the MW-520 connectors:

- mobile printers
- external speaker
- serial RS-232 communication devices
- DataTAC radio
- iDEN radio
- CDPD radio
- Standard GPS
- Any standard USB drive
- IDB vehicle bus standard

The MW-520 External Ports

The MW-520 serial external port supports a full set of RS-232 lines. The MW-520 includes one external port (COM1), and additional serial ports can be added by using USB serial converters.

DataTAC Radio

Creating a Radio Application

Using the standard interface for RF communication, you can choose to communicate directly with the MW-520 internal modem which is a standard serial device, running on COM3, with a default of 8 bits, no parity, 9600 baud interface. Options such as handshaking are selectable through the Native Mode interface. The only difference in the software interface is the COM3 base address, which is 03E8H, and the COM3 IRQ, which is IRQ4. By using Native Mode, you take on the responsibility of respecting all conventions of the Native Mode protocol and must personally address any contention between multiple applications (if there are any). For more information, refer to the *Native Mode Interface Reference Manual*, 68P04014C90.

When the MW-520 is in Suspend, the RF modem will usually be powered up in Receive mode. In order to prevent data loss when an outbound message arrives while the system is in Suspend (wake-up from Suspend can take 5 seconds), it is recommended to deassert the DTR line before Suspend.

The radio application/driver can detect the Suspend event by waiting for the Windows `PWR_SUSPENDREQUEST` message. When the Suspend request occurs,

the radio application should notify the modem to stop transferring data (deassert DTR, or switching the modem to transparent mode). If the application receives a `PWR_SUSPENDREQUEST` message during data transferring, it is recommended to reject the request in order to prevent data loss. When Resume occurs, it should notify the modem to continue (assert DTR, or switch the modem to Native mode).

The MW-520 resumes from Suspend mode when an RI signal is received from an external radio-modem connected to COM1, or from the internal radio-modem connected to COM3 (resume occurs on the RI falling edge). The modem sends an RI signal only when the DTR line is deasserted.

Note

Ensure that your application is power management aware and compatible with Microsoft Power Management API.

The RF-modem power-on can be controlled automatically via Windows start-up by an MW-520 HWAPP application (the power-on option is the default option).

Any Modem that Supports DTR Functionality

When the MW-520 enters Suspend state, the DTR is inactivated.
When the MW-520 is resumed from the Suspend state, an application should activate the DTR when it is ready to receive incoming data from the RF-modem.

The required sequence when resuming from Suspend:

1. Get the Windows message `PWR_SUSPENDRESSUME`.
2. Assert DTR.
3. Return with `PWR_OK` from the window procedure.

Loading a New System Image

This section details how to load a new system image on an MW-520 device, Models F5203/F5205. The three different methods for loading a system image on a device are: PCMCIA ATA HARD DRIVE card, Network, and Parallel Port. The selection of which method to use is left to the individual administrator. The following sections will describe how to use each one of these methods.

Materials and Software Needed

Once a preferred method for loading a system image has been selected, the appropriate materials, as described below, should be gathered.

Materials for PCMCIA ATA HARD DRIVE Card Method

- One bootable PCMCIA card Type II. Recommended cards are Calluna 260 MB HD (Type II), SanDisk ATA HARD DRIVE card 440 MB. (For more information on the products, go to the appropriate web site referred to on page 49.)
- Version 6.01 of the Ghost application - available from Symantec Corporation. (This should be installed on the ATA HARD DRIVE card.)
- A copy of the image to be downloaded onto the MW-520 unit. This should be copied onto the ATA HARD DRIVE card.
- A PC with DOS 6.22, to create the bootable ATA HARD DRIVE card.

Materials for Network Method

- A networked host PC (Pentium 166 MHz or above)
- A PC with Microsoft Windows 98 or DOS 6.22 to create the bootable floppy disk.
- An Ethernet PCMCIA card (an example is a 3Com 3CCFE574BT).
- A USB floppy drive (an example is from Y-E DATA, Inc.)
- A bootable floppy prepared with the network cards drivers and the relevant setting to a user's network. (The preparation process is described below.)
- Version 6.01 of the Ghost application - available from Symantec Corporation. (This should be installed on the ATA HARD DRIVE card in a shared directory.)
- A copy of the image to be downloaded onto the MW-520 unit. (This should be in a shared directory on the networked host PC.)

Materials for Parallel Port Method

- A host PC (Pentium 166 MHz or above) with a parallel port and a Microsoft Windows 9x system.
- A USB floppy drive (an example is from Y-E DATA, Inc.)

- One of the following two bootable media:
  - A bootable floppy and a USB floppy drive (an example is from Y-E DATA, Inc.)
  - A bootable ATA HARD DRIVE card
- A standard parallel crossover cable (male - male)
- Version 6.01 of the Ghost application - available from Symantec Corporation. (This should be installed on the host PC and on the bootable media.)
- A copy of the PREGHOST utility. (This is available in the recovery CD or from Motorola at FTP://MW520.MOT.COM). This should be installed on the bootable media.
- A copy of the image to be downloaded onto the MW-520 unit. (This should be on the host PC.)

Image Installation Via PCMCIA ATA HARD DRIVE

ATA HARD DRIVE card Preparation Procedure:

1. Insert the ATA HARD DRIVE card into a computer with DOS 6.22 and type the following command: `format <PC drive letter> /s`
2. Edit the config.sys in the root of the ATA HARD DRIVE card so that it contains the following strings:

   ```
   device=a:\himem.sys
   ```
3. Edit the `autoexec.bat` in the root of the ATA HARD DRIVE card so that it contains the following strings:

   ```
   @echo off
   prompt $p$g
   smartdrv.exe
   ```
4. Make sure that the following files are in the root directory of the ATA HARD DRIVE card: `himem.sys` and `smartdrv.exe`. If they are not, they must be copied to the root directory of the ATA HARD DRIVE card from a PC with MS DOS 6.22.
5. Copy the GHOST.EXE and the MW-520 system image (*.img or *.gho) files to the ATA HARD DRIVE card.

Image Installation Procedure

1. Ensure that the MW-520 unit is turned off.
2. Insert the ATA HARD DRIVE card into the lower PCMCIA card slot of the MW-520 unit.

Note

The orientation of the PCMCIA slot on the MW-520 unit is upside down from the orientation of previous versions of MW-520 units.

3. Power-on the MW-520 unit and launch `GHOST.EXE` from the DOS command prompt.

4. From within GHOST, choose `Local -> Disk -> From Image.`
5. Choose the image file you need.
6. Choose target hard drive (number 2, not ATA HARD DRIVE card).
7. Press `OK` to size, and then `Yes` to proceed.
8. Wait until the system image loading process is completed, then restart the computer.

Image Installation Via Network

Floppy Preparation Procedure

1. Insert a floppy disk into a computer with DOS 6.22 and Microsoft Windows 3.11 for Workgroups. Access a DOS window and type the following command: `format <PC drive letter> /s`
2. Edit the `config.sys` in the root of the floppy disk so that it contains the following strings:

```
device=a:\himem.sys /testmem:off
shell=a:\command.com /p /e:800
DOS=HIGH,umb
FILES=30
buffers=40
STACKS=9,256
LASTDRIVE=Z
device=a:\mswgcn\protman.dos /i:a:\mswgcn
device=a:\mswgcn\workgrp.sys
device=a:\mswgcn\elpc3.dos (or other name of your network
card MS-DOS driver)
```

3. Edit the autoexec.bat in the root of the floppy disk so that it contains the following strings:

```
@echo off
prompt $p$g
PATH=a:\mswgcn
a:\mswgcn\net start
```

4. Make sure that the following file is in the root directory of the floppy disk: `himem.sys`. If it is not, it must be copied from the root directory of the PC used to format the floppy.
5. Using a PC with Microsoft Windows 3.11 for Workgroups, install Microsoft Workgroup Client for Network (MSWGCN) to the following subdirectory of the floppy drive: `\mswgcn\`.
6. Make sure that the DOS PCMCIA network card driver is installed in the following subdirectory of the floppy disk: `\mswgcn\`.

7. Edit the `protocol.ini` in the `\mswgcn\` directory of the floppy disk, so that it contains the following strings:

```
[network.setup]
version=0x3100
netcard=ms$elnk3,1,MS$ELNK3
transport=ms$netbeui,MS$NETBEUI
lana0=ms$elnk3,1,ms$netbeui

[protman]
DriverName=PROTMAN$
PRIORITY=MS$NETBEUI

[MS$ELNK3]
DriverName=ELpc3$
        (or name of your network card driver: elpc3.dos -> elpc$)
IOADDRESS=0x300
MAXTRANSMITS=6
pcmcia_enabler=yes
INTERRUPT=0xB

[MS$NETBEUI]
DriverName=netbeui$
SESSIONS=6
NCBS=12
BINDINGS=MS$ELNK3
LANABASE=0
```

Image Installation Procedure

1. Ensure that the MW-520 unit is turned off, that the PCMCIA network card is installed in the unit, and that it is connected to a network. (Note: The orientation of the PCMCIA slot on the MW-520 unit is upside down from the orientation of previous versions of MW-520 units.)
2. Make sure that you have opened file sharing to the directory with the image file on the networked host PC. Finally, ensure that the Ghost application is on the host PC and that the PC is on the same network with the MW-520 unit.
3. Connect the USB floppy to the MW-520 unit and insert the bootable floppy disk that was created according to the "Floppy Preparation Procedure" section included above.
4. Power-on the MW-520 unit and enter `NET` from the DOS command prompt. You should see a color interface. Use it to map the network location on the networked host PC to the network drive on the MW-520 unit, for example: drive "X" to \\OurFileServer\images.

1. Exit the `NET` program and change directories to the network drive.
2. Launch `GHOST.EXE` from the DOS command prompt.
3. From within GHOST, choose `Local -> Disk -> From Image`.
4. Choose the image file you need from the networked PC's shared directory.

5. Press OK to "Target Drive", press OK to size and then Yes to proceed.
6. Wait until the system image loading process is completed, then restart the computer.

Image Installation Via Parallel Port


Note

When installing an MW-520 unit system image via a parallel crossover cable and the parallel port, it is very important that the MW-520 unit be configured as MASTER.


Note

The MW-520 unit parallel port is by default configured for ECP. Make sure that both the Host PC parallel port and the MW-520 unit parallel port are both configured the same.

As noted above, a bootable media is required. This can be either a bootable PCMCIA ATA HARD DRIVE card, or a bootable floppy and a USB floppy drive. The choice of which media to use is left to the individual administrator.

Floppy Preparation Procedure

Copy Ghost.exe, and Preghost.com into the floppy.

PCMCIA Preparation Procedure

Insert a PCMCIA ATA HARD DRIVE card into the computer:

Copy Ghost.exe and Preghost.com into the PCMCIA ATA HARD DRIVE card.

Image Installation Procedure

1. Ensure that the host PC (with Microsoft Windows 9x) has the GHOST.EXE program and the MW-520 unit system image (*.img or *.gho) file.
2. Before powering-on the MW-520 unit, connect the MW-520 unit to the Microsoft Windows 9x host via the parallel crossover cable.
3. Power-on the MW-520 unit and enter the BIOS Setup by pressing the F2 key when the Motorola logo is displayed.
4. Open the POWER menu item and set the Power Saving setting to Disabled.
5. Press the F10 key and select Yes to confirm changes.
6. Before the MW-520 unit is restarted, either connect the USB floppy drive (with the bootable floppy inserted), or insert the ATA HARD DRIVE card into the MW-520 PCMCIA drive.
7. Execute the PREGHOST.EXE utility on the MW-520 unit from the DOS prompt. Wait until it is completed.
8. Launch the Ghost application on the MW-520 unit. Choose LPT -> Master.
9. Launch the Ghost application on the host PC. Choose LPT -> Slave.

10. Wait until the connection is established (one second).
11. Choose Local -> Disk -> From Image
12. Choose the image file you need.
13. Press OK to the target drive, OK to size, and then Yes to proceed.
14. After the image is loaded, restart the computer.
15. Open Bios Setup (by pressing the F2 key when the Motorola Logo is displayed).
16. Press the F9 key (Setup defaults) and confirm changes (by selecting the Yes item).
17. Press the F10 key and confirm changes (by selecting the Yes item).
18. Restart your system.

Calluna Technology: http://www.calluna.com

SanDisk: http://www.sandisk.com

Symantec Corporation: http://www.symantec.com

Y-E Data: http://www.yedata.com

3Com Corporation: http://www.3com.com

Acronyms and Abbreviations

ACPI	Advanced Configuration and Power management Interface
API	Application Program Interface
ASIC	Application Specific Integrated Circuit
BIOS	Basic Input/Output System
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
DMA	Direct Memory Access
DRAM	Dynamic Random Access Memory
DTR	Data Terminal Ready
IDB	Integrated Data Base
IDE	Integrated Drive Electronics
I/O	Input/Output
IRQ	Interrupt ReQuest
ISA	Industry Standard Architecture
LCD	Liquid Crystal Display
LED	Light Emitting Diode
OEM	Original Equipment Manufacturer
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
RAM	Random Access Memory
RF	Radio Frequency
RI	Ring Indicator
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Signaling Bus
VGA	Video Graphics Array
SVGA	Super Video Graphics Array

Glossary

Application	A computer program used to perform a specific work.
API	The interface between application programs and the network software.
ASIC	A chip which can be readily customized for a given application.
BIOS	Software for transferring information between elements such as memory, screen and disk.
Bus	A communication channel carrying signals from any device used by the system to another device. For example, data being transferred to and from a hard disk travels on a bus.
Card Bus slot	A socket into which the Card Bus is inserted.
CMOS	The memory that stores the configuration you establish by running the computer's setup program. CMOS memory uses very little power and stores the information even when the computer is turned off.
COM Port	COM stands for communication. COM ports are the serial ports of the MW-520.
Device driver	A program that controls how software communicates with a physical device (for example, a mouse, memory, or a printer).
DMA	A method that allows a peripheral device to directly read or write to memory, without the time delay of going through the CPU.
DOS	A software that supervises computer's operation, including handling I/O.
Drive	A hierarchical organization of directories, stored on a disk.
DTR	An RS-232 control signal used by a terminal to tell a modem that it is ready to receive data.
Emergency key	An orange key - although it can be any key - that is configured to send emergency information to Dispatch.
Hard disk	A large-capacity data-storage device that is installed inside the MW-520.
IDE	A hard drive with a built-in controller.

ISA bus	A computer's bus is the hardware system it uses to transfer information between the different hardware elements of the computer. The ISA bus is the most common form of bus.
OEM	Supplier who makes equipment for sale by a third party. The equipment is usually disguised by the third party with his own labels.
Operating System	A program that supervises the computer's operation, including handling I/O.
PCI bus	A 32-bit local bus that provides connections for 32-bit add-in boards. The bus operates at an external clock speed of the microprocessor (up to 33 MHz). PCI devices are configured automatically by the system.
PCMCIA card	A removable media with various uses (such as disk drive, network or fax/modem) for expanding your MW-520. It should be used with the PCMCIA slot.
PCMCIA slot	Either of the two sockets on the processor into which the PCMCIA cards are placed (supports 16 and 32 bit).
RAM	A portion of the system's memory that is designed as a temporary storage area for data and programs. RAM includes conventional and extended memory.
RI	A modem signal which indicates that a remote modem has called (literally, "the phone is ringing").
UART	An integrated circuit which takes a character of data (eight bits in parallel), and transmits each bit serially over an asynchronous communication channel. It also accepts asynchronous data and provides a character to the device in which the UART is installed.

Index