American Megatrends

AMIKey-3 38813 Keyboard Controller

User's Guide

MAN-AMIKey13
6/19/97

Limited Warranty

Buyer agrees if this product proves to be defective, that American Megatrends, Inc. is only obligated to replace or refund the purchase price of this product at American Megatrend's discretion. American Megatrends shall not be liable in tort or contract for any loss or damage, direct, incidental or consequential. Please see the Warranty Registration Card shipped with this product for full warranty details.

Limitations of Liability

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

Trademarks

American Megatrends acknowledges the following trademarks:

Mitsubishi is a registered trademark of Mitsubishi Corporation.
MS-DOS and Microsoft are registered trademarks of Microsoft Corporation. Microsoft Windows, Windows NT, and Windows 95 are trademarks of Microsoft Corporation.
Novell NetWare is a registered trademark of Novell Corporation.
Unix and SCO are registered trademarks of The Santa Cruz Operation, Inc.
3COM is a registered trademark of 3COM Corporation.
IBM, AT, VGA, PS/2, MCA, OS/2, and EGA are registered trademarks of International Business Machines Corporation. PC-DOS, XT and CGA are trademarks of International Business Machines Corporation.
DEC is a registered trademark of Digital Equipment Corporation.

Revision History

6/25/96	Initial release.
12/20/96	Added chapter about smart battery support.
1/7/97	Added additional commands.
6/5/97	Added additional commands
6/19/97	Added additional commands

Table of Contents

Preface

Purpose This manual provides sufficient technical information for the OEM to design system hardware based on the Mitsubishi® 38813 keyboard controller. This manual was written for the OEM to assist in the proper use and operation of the American Megatrends keyboard controller.

If You Need Help

Call American Megatrends technical support at 770-246-8645 if you have problems. The hours are 8:00 AM - 7:00 PM EST.

Tech Support Information Gather the following information before calling technical support:

- the system configuration, including hard disk drive type, floppy drives, amount of memory, type of monitor and type of keyboard (and model number).
- the keyboard controller reference number,
- the operating system/environment (DOS, Windows, OS/2, Unix),
- contents of AUTOEXEC.BAT and CONFIG.SYS,
- all resident programs (loaded or not),
- the system BIOS reference number,
- the CPU type and clock frequency and the keyboard controller clock frequency, and
- a clear description of the problem.

BBS You can access American Megatrends product information on the American Megatrends BBS.

Data Transmission The American Megatrends BBS automatically handles modems with data transmission rates from 1,200 to 28,800 bps.

Cont'd

Phone Numbers The BBS requires no parity, eight data bits, and one stop bit. The characteristics of the American Megatrends technical support, BSS, and other important phone numbers are:

Phone Number/Address	Characteristics
Tech Support Voice	770-246-8645
Tech Support Fax	770- 246-8772
Sales	800-828-9264
Internet Address	support@american.megatrends.com
Web Site	http://www.megatrends.com
770-246-8780	28,800 baud rate. Supports v.34.
770-246-8781	28,800 baud rate. Supports v.34.
770-246-8782	Supports HST and v.42.
770-246-8783	Supports HST and v.42.

1 AMIKey/38813 Keyboard Controller Features

The AMIKey/38813 keyboard controller firmware features include:

- based on the Megakey keyboard controller code,
- supports ISA, EISA, and MCA® architecture,
- provides transparent software Gate A20 support,
- supports an internal keyboard,
- supports auxiliary devices,
- supports an internal scan code controller,
- supports internal pointing devices,
- supports an internal numeric keypad,
- supports direct LED control, and
- supports the smart battery SMBus interface.

External Keyboard and Auxiliary Devices The American Megatrends AMIKey/38813 keyboard controller firmware supports:

- standard IBM PC/AT® computers,
- IBM PS/5S numeric keypads, and
- PS/2® auxiliary devices.

Cont'd

Simultaneous Use of External and Internal Devices The firmware provides: simultaneous use of external and internal keyboards, and simultaneous use of external and internal auxiliary devices.

Hot Pluggability The firmware provides: hot pluggability of external keyboard and PS/2 devices and hot port swapping of external devices.

Power Management Support The firmware supports: WAIT Mode and STOP Mode.

Password and Hot Key Support The firmware supports: a BIOS password and general and function hot keys.

Downloadable Features The firmware supports: downloadable internal keyboard matrix, and downloadable function shifted key table.

Based on Megakey Code AMIKey has a modular set of features. This has come from the previous versions of American Megatrends keyboard controllers. This has been brought to give optimum performance and flexibility to include future enhancements in keyboard controller computer designs.

ISA, EISA, and MCA Support The AMIKey/ 38813 firmware supports a superset of IBM PS/2 keyboard and auxiliary device controller host interface. The system BIOS can configure the AMIKey as an AT 8042 with no PS/2 support. The default is PS/2 support. The IRQ12 line is not normally connected on the AT schematic but is connected on the MCA and EISA schematics. If the IRQ12 line is connected on the AT schematic, mouse support can be achieved through port swapping with only one external keyboard connector.

Cont'd

Transparent Software Gate A20 Support GateA20 commands access system memory above 1 MB. These commands are often used in Windows™ and Novell NetWare® networking applications. As machine speeds and software memory requirements have increased, the number of Gate A20 commands also increased. Most chipsets directly handle Gate A20 commands, speeding system performance.

Internal Keyboard Scan Code Controller This feature provides a fully functional and compatible internal parallel keyboard. The internal scan code controller responds to all keyboard commands and produces 101, 102, 105, and 106 key keyboard-compatible data.

Internal Auxiliary Device Support The AMIKey/ 38813 firmware supports an additional PS/2 port for an internal pointing device such as a Touch Pad, track ball, or glide point.

Internal Numeric Keypad Support The firmware supports a numeric keypad centered around the letter 'I'. The system BIOS can configure the keyboard controller to enable numeric keypad support. If the external numeric keypad or the keyboard is present, the computer can (optionally) choose the internal numeric pad that is supported.

Cont'd

Direct LED Support The firmware uses the LED drive port to show the Scroll Lock, Caps Lock, and Num Lock status by driving the corresponding LEDs directly. When the corresponding bit is set, the LED glows unless it is Off. Up to four LEDs can be driven directly from the keyboard controller, freeing up an additional port pin. This pin can be used for the function bit status.

External Keyboards The firmware supports many types of external keyboards including all AT, PS/2, 84-key, 101-key, 102-key, 105-key, or 106-key keyboard.

External PS/2 Style Auxiliary Devices The firmware is made compatible to any PS/2 mouse or any auxiliary device that has a PS/2 serial interface.

External and Internal Keyboards Simultaneously The firmware allows you to use both the external and internal keyboards at the same time. So, the special features such as Fn hot keys that are available only in the internal keyboard can be used while using the external keyboard. This way it gives more user flexibility and convenience.

Simultaneous External and Internal Auxiliary Devices The firmware supports both external PS/2 mouse and internal auxiliary devices (any kind of pointing device with a PS/2 interface), making them usable at the same time. Users can choose from many types of pointing devices.

Cont'd

External PS/2 Device are Hot Pluggable The firmware tracks the external PS/2 devices status. It knows if the device has been plugged in or unplugged. If the device has been plugged in recently, it first checks for port swapping and later initializes the device that has been plugged in to the same state as the internal device (if the internal device is already present). It maintains the controller configuration to handle both internal and external devices. Hot pluggability is a user convenience.

This feature is useful when hot docking the computer to a docking station with external devices already attached to it.

Hot Port Swapping Usually two ports connect the keyboard and auxiliary devices. With port swapping, any port can be used for the keyboard or the mouse. Ports can be interchanged while the computer is up and running. Only one external keyboard or mouse can be present at one time. Both ports cannot be occupied by two keyboards or mice. Port swapping can be done during power-on or while the computer is up and running.

WAIT Mode AMIKey supports the power saving mode (WAIT) during idle condition. All the keyboard controller events are handled by interrupts, so when there is no work to be done by the keyboard controller, it automatically enters WAIT mode. While WAIT mode is active, all RAM locations, ports, and registers maintain the same status they had before WAIT Mode. The keyboard controller returns to normal operating mode (so the generated interrupt can be processed) when any interrupt event (such as a key press) occurs.

Cont'd

STOP Mode This is a command initiated mode. Generally, when the computer is entering suspend mode, the command CBh makes the keyboard controller enter STOP mode. The keyboard controller leaves STOP mode when:

- a hardware reset occurs,
- a host command or data is received, or
- a key is pressed on the internal keyboard.

This keypress can also be used to wake the computer up by performing the assigned wake up task. The wake-up procedure is controlled by Anykey wakeup and Anykey resume flags.

If the Anykey wakeup and Anykey resume flags are set, the wakeup task is executed automatically when the keyboard controller leaves STOP mode. This task may cause an SMI or another event that, in turn, cause computer operation to resume. If the Anykey resume flag is zero, the computer can be resumed by an hot key event.

Password Support The keyboard controller provides the keyboard password feature. This password is available in addition to the password provided by the system BIOS. Depending on the implementation, the keyboard controller password can be the same as the system BIOS password.

Quick Lock Support The keyboard and mouse are locked until the user enters the correct password. The keyboard controller password is 6 bytes long with a null character at the end. The password must be loaded before this feature can be used. By default, either <Function key> <Backspace> or <Ctrl> <Alt> <Backspace> locks the keyboard and mouse under the Quick Lock enabled state. The keystroke combinations can be changed to any other keystroke combination, if desired.

The keyboard and mouse can be locked without leaving the user application. The faceplate LEDs are flashed when the keyboard controller enters the quick lock state. LED flashing is disabled automatically after a predefined time. LED flashing stops after the user enters the correct keystroke combination.

Cont'd

Hot Key Support AMIKey supports three types of hot keys:

- function hot keys
- pulse hot keys, and
- general hot keys.

Hot keys set, clear, toggle, or pulse the designated port pin. When the hot key is pressed, an external SMI or another interrupt occurs. A variable is set to indicate the hot key that was pressed. The system BIOS recognizes the interrupt and sends a CDh command to the keyboard controller to find the hot key that caused the SMI. The BIOS executes the task assigned to that hot key.

Function Hot Keys Function hot keys are invoked by pressing a <Function key> <Function key> keystroke combination. A function key is any key from F1 to F12. When this key combination is pressed, the function task is set pending and is executed when the function key is released.

Pulse Hot Keys The pulse hot keys are invoked by pressing <Function key> <pulse hot keyX> keystroke combination on the internal keyboard. Any key on the internal keyboard can be configured as a pulse hot key. The keyboard controller sends low-going pulses at the typematic rate to the designated port pin when the specified keystroke combination is pressed. The pulses are used for controlling brightness, contrast and volume.

General Hot Keys General hot keys are invoked by pressing either <Ctrl> <Alt> <hot key> or <Function key> <hot key>. The hot key is any non-extended key. A non-extended key is a key that produces only one scan code per Make/Break). The scan code value of the key used as a general hot key is stored in the RAM location by the system BIOS. The corresponding task is stored in the corresponding Keytask RAM location. For <Ctrl> <Alt> <hot key> keystroke combinations, the corresponding task is set pending and is executed when all hot key keys are released. For all <Function key> <hot key> keystroke combinations, the task is executed when the Hot key is released.

Cont'd

Smart Battery SMBus Interface This feature provides an interface to smart batteries that are compatible with the Intel/Duracell SMBus interface specification. The system BIOS and any application can retrieve battery-related data using this interface.

Downloadable Internal Keyboard Matrix The firmware allows you to download the internal keyboard matrix through the BIOS at power on. Any value from 00h to FFh can be placed in the matrix, depending on the functionality requirement. A numeric keypad centered on the letter "I" is made available by pressing Fn key or with Num Lock on. Any 128 key matrix location can be modified at any time by writing to the corresponding RAM location using extended command BBh and an index value from 80h through FFh with bank 0 selected.

Downloadable Function Key Table The firmware can down load the function modifiable keys. Any value from 00h through BFh can be put in this table. The table stores the scan code value of a specific key when it is pressed alone (unshifted value) and if it is used in conjunction with the function key (fn_shifted value). These key values include standard keys, hot keys, and special handled keys for full functionality.

Speed Independent Operation The AMIKey keyboard controller works independently with either a 4 MHz or 8 MHz clock. AMIKey automatically adjusts all clock-related variables. The default clock speed is 4 MHz. The system BIOS can change the keyboard controller speed to 8 MHz by executing the DFh command.

2 Keyboard Controller Memory Map

The AMIKey/38813 keyboard controller allows the system BIOS or the OEM keyboard utility to read from or write to the 384 byte RAM using extended commands 0B8h through 0BBh. The 38813 memory map is shown below. The bit definitions describe all diagnostics and state saving/restoring information that is needed to understand the M38813 internal states.

Symbol	RAM Location (Range)	Description
SFR area	000h-03Fh	M38813 Special Function Registers
BUFFER	40h	General purpose temporary register
R1	41h	General purpose register r1
Stat8042	42h	Controller status flags **Bit** **Description** 7 Parity error during receive. 6 General timeout error. *Local bit definition for bit 5-0* 5 mouse IRQ pulse type. 4 Reserved. 3 A6 flag and password active. 2 Reserved. 1 Break code flag. 0 PS/2 interface *System bit definition for bit 5-0* 5 1 Mouse data output buffer full 4 1 Keyboard not inhibited. 3 1 Last data output in port 64. 2 1 System flag bit set. 1 1 Input buffer full. 0 1 Output buffer full.
R3-R7	43h-47h	Temporary subroutine scratch registers (6 bytes)
STACK	48h-56h	System Stack (16 bytes)
STACK	57h	Stack Pointer
BLANK	58h	Unused
SWAP_XCHG	59h	Scratch Register
MS_RESP_COUNTER	5Ah	Mouse response counter
KB_RESP_COUNTER	5Bh	Keyboard response counter

Symbol	RAM Location (Range)	Description
PASSWORD_PTR	5Ch	Password Pointer
PORT_60_CHAR	5Dh	Old character buffer
WAKEUP_TASK_REG	5Eh	Wakeup task storage register
MEM_INDEX	5Fh	Internal memory index register
CCB	60h	Keyboard Controller command byte. **Bit Description** 7 Reserved (must be 0). 6 1 IBM Keyboard translate mode. 5 1 Disable mouse interface. 4 1 Disable keyboard interface. 3 1 Keyboard inhibit override. 2 1 Set system flag. 1 1 Enable mouse interrupt. 0 1 Enable keyboard interrupt.
USER_RAM	61h-7Fh	IBM defined User RAM (20-3Fh)
PASSWORD_AREA	80h-86h	Password storage area (6 bytes)
UNUSED	87h-8Bh	Unused Memory Area
Pk_sel_byte	8Ch	Pulse key option byte
debounce_time	8Dh	Make/Break debounce time
pulse_width	8Eh	Pulse width for function hot keys
HK_FLAG_BYTE	8Fh	Hot key Make/Break flags
SCODE_HK1	90h	Hot key1 scan code storage
SCODE_HK2	91h	Hot key2 scan code storage
SCODE_HK3	92h	Hot key3 scan code storage
SCODE_HK4	93h	Hot key4 scan code storage
SCODE_HK5	94h	Hot key5 scan code storage
SCODE_HK6	95h	Hot key6 scan code storage
TASK_HK1	96h	Hot key1 task storage
TASK_HK2	97h	Hot key2 task storage
TASK_HK3	98h	Hot key3 task storage
TASK_HK4	99h	Hot key4 task storage
TASK_HK5	9Ah	Hot key5 task storage
BATTERY_CONFIG	9Bh – A0h	Storage for battery configuration data.
Unused	A1h – B2h	Unused Memory Area
Kbc_State3	B3h	Keyboard controller flag byte Bits 7-2 Reserved Bit 1 Enable IRQ12 tristate Bit 0 Ghost keycheck disable
Ccdata	B4h	Comparator control data

Symbol	RAM Location (Range)	Description
Aux_Config	B5h	Bits 7-5 Reserved Bit 4 Aux. control bit 1 Bit 3 Aux. control bit 0 Bits 2-0 Reserved
Prescale	B6h	Timer prescale value
CFG813	B7h	Keyboard controller configuration register Bits 7-6 Reserved Bit 5 Int. aux. present Bits 4-3 Reserved Bit 2 Controller speed Bit 1 Reserved Bit 0 Scan method
MSTATE4	B8h	Auxiliary status register Bit 7 Station req. hit Bit 6 Aux. tested Bits 5-4 Reserved Bit 3 Expect aux. data bit 0 Bit 2 Expect aux. data bit 1 Bit 1 Last aux. data bit 1 Bit 0 Last aux. data bit 0
KBMSidletimer	B9h	Idle timer
P1DATA	BAh	Emulated Port1 data
MSTATE1	BBh	Mouse status information **Bit Description** 7 ms_data_bit(Status/data) 6 ms_modebit (remote/stream) . 5 ms_flag bit(enable/disable) 4 ms_scale bit(2:1/1:1). 3 Wrap_mode bit. 2 Unused. 1 ms_f3 flag. 0 ms_e8_flag.

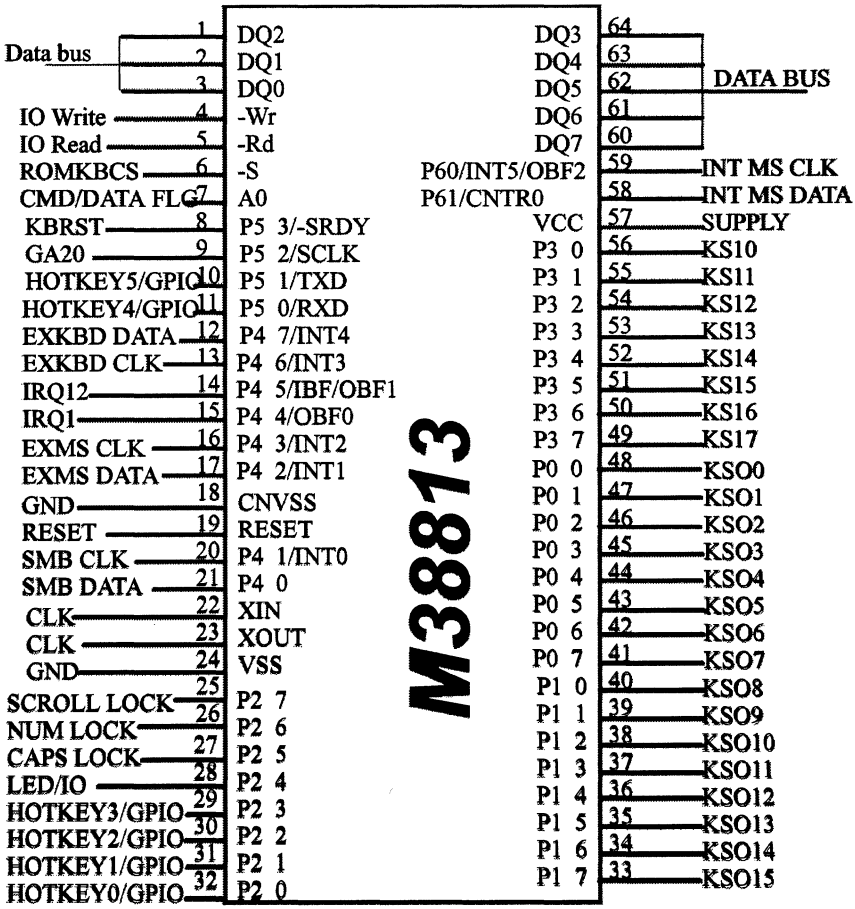
Symbol	RAM Location (Range)	Description
MSTATE2	BCh	Mouse Status information **Bit Description** 7 Int. aux packet count (bit 1) 6 Int. aux packet count (bit 0) 5 External mouse attached 4 Mouse resolution (bit 1)/ 3 Mouse resolution (bit 0). 2 Unused. 1 Unused. 0 Unused.
MSTATE3	BDh	Keyboard/Mouse status information **Bit Description** 7 Delay_flag bit. 6 Set Anykey resume bit. 5 Set Anykey wakeup bit. 4 External keyboard connected. 3 Ext. aux packet count (bit 1). 2 Ext. aux packet count (bit 0). 1 Aux. last data pointer (bit 1). 0 Aux. last data pointer (bit 0).
FUNCTSK_REG	BEh	FN hot key task storage register
INT_FN_REQ_NUM	BFh	Data for get interrupt function number
TYPEMATIC_RATE	C0h	Typematic rate. **Bit Description** 7 Auto_repeat bit. 6 Typematic rate delay (bit 1). 5 Typematic rate delay (bit 0). 4 Typematic rate (bit 4). 3 Typematic rate (bit 3). 2 Typematic rate (bit 2). 1 Typematic rate (bit 1). 0 Typematic rate (bit 0).
LED_FLAG_BYTE	C1h	LED and Control flags. **Bit Description** 7 Alt key status bit. 6 Ctrl key status bit. 5 Left Shift status bit. 4 Right Shift status bit. 3 Function Key LED bit. 2 Caps Lock LED bit. 1 Num Lock LED bit. 0 Scroll Lock LED bit.
CH_BUFF_HEAD	C2h	Character buffer head pointer.
CH_BUFF_TAIL	C3h	Character buffer tail pointer.
SC_BUFF_PTR	C4h	Scanner buffer pointer
LAST_SCAN_INDEX	C5h	Last scanner character found

Symbol	RAM Location (Range)	Description
DELAY_COUNT_BYTE	C6h	Delay Counter **Bit Description** 7 Unused. 6 Unused. 5 Unused. 4 Delay counter (bit 4). 3 Delay counter (bit 3). 2 Delay counter (bit 2). 1 Delay counter (bit 1). 0 Delay counter (bit 0).
KBMS_COMMAND_STATUS	C7h	Keyboard Command Status **Bit Description** 7 Keyboard and mouse port swapped. 6 Scanner disabled/enabled 5 Scan code Set (Bit 1). 4 Scan Code Set (Bit 0). 3 T1_flag bit. 2 Keyboard command flag(bit 2). 1 Keyboard command flag(bit 1). 0 Keyboard command flag(bit 0).
CH_BUFFER	C8h-D7h	Character buffer Character buffer start address is C8h Character buffer end address is D0h
Unused	D8h – DAh	Unused
KBC_FLAGS	DBh	Keyboard Command Status **Bit Description** 7 Memory Page Index (bit 1). 6 Memory Page Index (bit 0). 5 Disable internal auxiliary device. 4 Hot key make flag 3 Numeric pad lock bit 2 Function pad lock bit 1 Disable hot pluggability 0 Error flag
MSSMPL_RATE	DCh	Mouse Sampling Rate 0Ah 10 Packets/sec 14h 20 Packets/sec 28h 40 Packets/sec 3Ch 60 Packets/sec 50h 80 Packets/sec 64h 100 Packets/sec C8h 200 Packets/sec

Symbol	RAM Location (Range)	Description
Kbs_state1	DDh	Keyboard controller flag register Bit 7 Counter Bit 4 Bit 6 Counter Bit 3 Bit 5 Counter Bit 2 Bit 4 Counter Bit 1 Bit 3 Counter Bit 0 Bit 2 LED flash bit Bit 1 LED tmpstp bit Bit 0 Reserved
Unused	DEh - DFh	Unused memory
SC_BUFFER	E0h – E9h	Scanner buffer Scanner buffer start address=E0h Scanner buffer end addres=D8h
SC_FLAG_BUFFER	EAh – F3h	Scanner buffer flag byte storage The scanner buffer flag byte start address is EAh. The scanner buffer flag byte end address is F4h.
Unused	F4h - FFh	Unused memory
FN_SHFT_TABLE	100h-13FH	RAM loaded function shift code table (64bytes)
SCAN_MATRIX	140h - 1BFh	User-defined scan code set_2 table (128 bytes).
BATT_DATA	1C0h – 1FFh	Smart battery handling data registers
Unused	200h – 23Fh	

AMIKey-3 /38813 Pinout for M38813

The recommended pin configuration for implementing AMIKey-3/38813 using a Mitsubishi M38813 keyboard controller in a QFP- type package is:



Note

Pins 8 and 9 can either be used as GA20/KBRST pins or as general purpose I/O or as general hot key task pins.

DQ0 - DQ7 are the data bus for host CPU interface.

Key Scan Output Level Option

A keyboard matrix consists of 16 columns (scan lines) intersected by 8 rows (sense lines). To determine any key press on the rows involves scanning the scan lines one at a time and reading the sense lines. Different keyboard types require scan lines that are not being strobed to be either high impedance or to be driven high. The configuration bits involved in these options are:

Keyboard Type	Ghost Key Function	Scan Method Flag
Resistor network keyboard (default)	1	0
Diode network keyboard	1	1
Plain keyboard	0	1

Auxiliary Device Control Options

The American Megatrends keyboard controller has several options that configure both internal and external auxiliary devices. These options are controlled by aux_config register variable. The default condition makes both internal and external auxiliary devices active simultaneously. This may cause some compatibility errors when both auxiliary devices are not PS/2-compatible. Only one device can be made active by downloading configuration bits. Auxiliary device control should only be configured by the system BIOS before the first PS/2 device command (the keyboard Reset command) is sent during POST/Resume. The Aux_config register contents at B5h are:

Bit	Description
7-6	Reserved
4 -3	Aux_cntrl_bit1 and Aux_cntrl_bit0 00 Both devices are active simultaneously. 01 External Aux. Primary. The external auxiliary device is active if it is attached during POST/Resume. Both devices are active if the external auxiliary is hot plugged after POST/Resume. 10 Reserved 11 Either device support. Only the external auxiliary is active if it is present during POST/Resume. Otherwise, only the internal auxiliary is active.
2-0	Reserved

3 Keyboard Controller Functions

The function of the keyboard controller in an ISA or EISA computer is shown below:

Keyboard

Sends a keyboard
Make/Break scan code

Keyboard Controller

Sends a Make/Break
system scan code

BIOS

INT 09h

Queries shift and toggle state flags
Handles internal function requests.
Converts system scan code to
16-bit character code.
Sends character code to INT 16h.

Receives and Translates Serial Data The keyboard controller receives serial data from the keyboard and/or mouse, checks the parity of the data and translates it to a system scan code, if necessary. The keyboard controller places the received and processed data in the data buffer and interrupt processor.

Executes System Commands The keyboard controller executes system commands through the controller command buffer (CCB) and places the result, if necessary, in the data buffer and then interrupts the CPU.

Cont'd

Transmits Computer Data The keyboard controller transmits computer data, places it in the data buffer, and sends it to the keyboard or mouse in a serial format with the parity bit inserted. It receives responses from the keyboard or mouse and reports to the computer CPU.

Reports Errors The keyboard controller reports errors to the computer through status registers at the time of data communication with the keyboard or mouse.

Receives Data from the Keyboard The keyboard sends data in an 11-bit serial format to the keyboard controller:

Step	By	Action
1	keyboard	The data begins with a start bit (low level) followed by 8 data bits (least significant data bit first), an odd parity bit, and a stop bit (high level).
2	keyboard	Data sent is synchronized with the keyboard clock.
3	keyboard controller	On receiving a byte of data from the keyboard, the keyboard controller places the data in its one-byte receive-data buffer and disables the keyboard interface until that data is picked up by the computer microprocessor. This avoids data overrun.
4	system CPU	Reads the data from the keyboard controller receive-data buffer.
5	keyboard controller	On parity error, the controller requests that the keyboard resend the data. If the error is repeated, the controller sets the parity error bit in its status register. The keyboard controller sets the timeout bit in the status register if all eleven bits are not received within two milliseconds from the start of the transmission. If either error occurs, FFh is placed in the receive-data buffer.

Cont'd

Possible Errors When Sending Data to the Keyboard Data is sent to the keyboard in the same serial format as data received from the keyboard. Some errors that may occur include:

If the time between request to send and start of transmission is greater than 15 milliseconds or the duration of transmission is greater than 2 milliseconds, the transmit timeout error bit is set in the status register.
The keyboard must acknowledge every transmission from the controller. If the acknowledgment has a parity error, the keyboard controller sets both the parity and transmit timeout error status bits.
If the acknowledgment does not arrive within 25 milliseconds, both the receive and transmit timeout error bits are set.
FEh is placed in the data buffer if any of these errors occur.
No retries are made for errors when transmitting to the keyboard.

Keyboard Inhibit The keyboard can sometimes be inhibited by shorting a jumper. See the motherboard manual for additional information. Although all transmissions from the computer to the keyboard are permitted when the keyboard is inhibited, the keyboard controller tests all data received from the keyboard. If it is a response to a command sent to the keyboard, it is placed in the data buffer.

It is ignored otherwise.

Computer and Keyboard Controller Command Interface

The computer issues commands to the keyboard controller through I/O port 64h. The data and subcommands associated with the main command are written to I/O port 60h. The keyboard controller executes these commands and places the response in I/O port 60h.

All data from the keyboard controller to the computer is routed through I/O port 60h. The computer reads all keyboard and auxiliary data from I/O port 60h.

The computer reads the keyboard controller status from I/O port 64h.

The keyboard commands are passed by the computer through I/O port 60h.

Auxiliary device commands are also written to I/O port 60h, but these commands are actually data and subcommands to the keyboard controller D4h command (write the byte to the auxiliary device).

Keyboard Controller Status Register

The keyboard controller status register format is:

Bit	Description
7	Parity Error 0 No parity error. 1 The last byte received from the keyboard had a parity error. The keyboard sends data with odd parity.
6	Timeout Error 0 No timeout error. 1 A data transmission from the keyboard to the keyboard controller was not properly completed within the pre-defined time limit.
5	Auxiliary Output Buffer Full 0 Keyboard data 1 Mouse data
4	Inhibit Switch This bit reflects the state of the keyboard inhibit switch. This bit is updated when the controller writes to the output buffer. 0 Keyboard inhibited 1 Keyboard not inhibited
3	Command/Data Used by the keyboard controller to determine if the input buffer contains a command or data. 0 The system writes to the input buffer through I/O port 60h. 1 The system writes to the input buffer through I/O port 64h.
2	System Flag The keyboard controller can set this bit to 0 or 1 depending on the command from the computer. It is set to 0 after power on reset.
1	Input Buffer Full 0 The keyboard controller input buffer is empty. 1 The system has written to the input buffer. This bit is reset to 0 when the controller reads the input buffer.
0	Output Buffer Full 0 The keyboard controller output buffer has no data. 1 The keyboard controller has written to the output buffer. The keyboard controller set this bit to 0 when the computer reads the output buffer (60h).

Cont'd

The keyboard controller status register, which is read from I/O port 64h, indicates if the keyboard controller is ready to accept another command or if response data is ready from the previous command.

The computer can only send data or commands to the M38813 keyboard controller if the IBF (Input Buffer Full) flag is *cleared*. The data from the M38813 is valid only if the OBF (Output Buffer Full) flag is *Set*.

If the computer sends a command to the keyboard controller that has a response, the computer waits for the OBF flag to be set before reading the data from I/O port 60h.

Keyboard Controller Command Byte

The internal status is defined by the keyboard controller command byte (CCB). The CCB resides in RAM location 20h, defined in the IBM 8042 programming specification. This location is mapped to 60h in the M38813 keyboard controller.

The CCB can be read from or written to using special commands or using the IBM standard commands. In the IBM standard command set, use command 20h to read the CCB and command 60h to write CCB command 60h.

Bit	Description
7	Reserved
6	IBM PC Compatibility Mode 0 Pass untranslated scan codes to the computer. 1 Translate scan codes to IBM PC standard before sending to the computer (default setting).
5	Disable Auxiliary Device 0 Auxiliary device enabled. 1 Auxiliary device disabled (in an AT, use 1 for XT keyboards).(default)
4	Disable keyboard 0 Keyboard enabled. 1 Keyboard disabled (interface inactive). (Default)
3	Inhibit override (in AT). Reserved in PS/2. 1 = disable keyboard inhibit function
2	System Flag 0 = System executing POST as a result of cold boot (Default). 1 = System executing POST as a result of warm boot
1	Enable Auxiliary OBF interrupt 0 (Default) 1 System interrupt generated when a byte is placed in the auxiliary output buffer
0	Enable Keyboard OBF Interrupt 1 System interrupt generated when a byte is placed in the output buffer

4 Keyboard Controller Commands

Standard Keyboard Controller Commands

The standard keyboard controller commands implemented by 38813 keyboard controller for AT and PS/2 environments are:

Command	AT	PS/2	Description
00h-1Fh	X	X	Read the contents of the designated RAM locations (20h - 3Fh) and send it to the computer
20h-3Fh	X	X	Read the contents of the designated RAM locations (20h-3Fh) and send it to the computer
40h-5Fh	X	X	Get a byte of data from the computer and write into one of the RAM locations (20h - 3Fh)
60h-7Fh	X	X	Get a byte of data from the computer and write into one of the RAM locations (20h-3Fh)
A4h	X	X	Test password installed Returns 0FAh if password is loaded Returns 0F1h if password is not installed
A5h	X	X	Load password Loads password until a '0' is received from the computer
A6h	X	X	Enable password Enables the checking of keystrokes for a match with the password
A7h		X	Disable auxiliary device
A8h		X	Enable auxiliary device
A9h		X	Test auxiliary device clock and data (interface test)
AAh	X	X	M38813 keyboard controller self test Returns 055h if successful.
ABh	X	X	Test keyboard clock and data lines (interface test)
ADh	X	X	Disable keyboard
AEh	X	X	Enable keyboard
C0h	X	X	Read input port (emulate port1)
C2h	X	X	Poll input port high and put it in the status register
C3h	X	X	Poll input port low and put it in the status register
D0h	X	X	Read Port 2 (send the Gate A20 status to the computer).

Command	AT	PS/2	Description
D1h	X	X	Write Port 2 (Set or reset the Gate A20 line based on system bit 1).
D2h	X	X	Send data back to the system as if it came from the keyboard.
D3h		X	Send data back to the system as if it came from the auxiliary device.
D4h		X	Output next received byte of data from system to auxiliary device
E0h	X	X	Read test inputs.
FEh	X	X	Pulse the RC (the reset line) low for 6 µs.

Special Keyboard Controller Commands

The special commands supported by the AMIKey keyboard controller are used to:

- access internal RAM for loading an internal keyboard matrix,
- edit function key table,
- set hot key values, and
- set other configuration data.

Command	AT	PS/2	Description
A0h	x	x	Send American Megatrends copyright message to the computer.
A1h	x	x	Send controller BIOS version number to the computer.
B8h	x	x	Configure the memory access index MEM_INDEX*
B9h	x	x	Get current contents of MEM_INDEX
BAh	x	x	Read the contents of the memory location pointed by MEM_INDEX
BBh	x	x	Write the memory location pointed by MEM_INDEX

Command	AT	PS/2	Description
BCh BDh	x	x	Read or write the following controller variables pointed by MEM_INDEX: STAT8042 0 Controller status register Password_ptr 1 Pointer to password storage area Wakeup_Tsk_Reg 2 Wakeup task storage register CCB 3 Controller Command Byte Debounce_time 4 Make/Break debounce time Pulse_Width 5 Pulse width selection for hot keys Pk_sel_byte 6 Pulse hot key control byte Func_Tsk_Reg 7 Function hot key task storage register TypematicRate 8 Keyboard typematic rate storage Led_Flag_Byte 9 LED/Ctrl/Alt/Shift status register Kbms_Command_St A Command status register Delay_Count_Byte B Typematic delay counter value KBC_Flags C Keyboard status flags register SCODE_HK1 D Hot key 1 scan code storage SCODE_HK2 E Hot key 2 scan code storage SCODE_HK3 F Hot key 3 scan code storage SCODE_HK4 10 Hot key 4 scan code storage SCODE_HK5 11 Hot key 5 scan code storage SCODE_HK6 12 Hot key 6 scan code storage TASK_HK1 13 Hot key 1 task storage TASK_HK2 14 Hot key 2 task storage TASK_HK3 15 Hot key 3 task storage TASK_HK4 16 Hot key 4 task storage TASK_HK5 17 Hot key 5 task storage Batt_Poll_delay_Time 18 Battery Polling delay time value Batt_Alarm_Reg1 19 Battery alarm mask register Batt_Alarm_Reg2 1A Battery alarm removed mask register Batt_Alarm_Tsk_Reg 1B Alarm task storage register Kbc_State1 1C Keyboard controller flag register 1 Aux_Config 1D Auxiliary configuration register Kbc_state3 1E Keyboard controller flag register 3
C1h	x	x	Write emulated port 1

Command	AT	PS/2	Description
C5h	x	x	Read, Set, Clear, or Toggle port bits Bit 7 0 Level 1 Pulse Bits 6-5 00 Read 01 Set 10 Clear 11 Toggle Bits 4-0 00h–07h Ports 0.0 – 0.7 08h–0Fh Ports 1.0 – 1.8 10h–17h Ports 2.0 – 2.7 18h–19h Port 4.0, 4.1 1Ah–1Dh Ports 5.0 – 5.3 1Eh–1Fh Ports 6.0, 6.1
C8h	x	x	Tristate IRQ12 Line 00h Tristate IRQ12 01h Default IRQ12
CBh	x	x	Enable password and flash LEDs
CCh	x	x	Enter Power_Down State (STOP mode)
CDh	x	x	Get interrupt function request 00h No keyboard controller function 01h HK1 02h HK2 03h HK3 04h HK4 05h HK5 06h HK6 Quick Lock 07h Unused 08h Wakeup Event Request 09h Unused 0Ah FK1 0Bh FK2 0Ch FK3 0Dh FK4 0Eh FK5 FK1 through FK12 0Fh FK6 Internal keyboard only 10h FK7 11h FK8 12h FK9 13h FK10 14h FK11 15h FK12 16h Pulse P2.0 17h Pulse P2.1 18h Pulse P2.2 19h Pulse P2.3 1Ah Pulse P5.0 1Bh Pulse P5.1 1Ch Pulse P5.2 1Dh Pulse P5.3 1Eh Battery alarm event

Command	AT	PS/2	Description
CFh	x	x	Control Any Key Resume 00 Disable Anykey wakeup. 01 Enable Anykey wakeup and hot key resume. 10 Enable Anykey wakeup and Anykey resume. 11 Unused
D8h	x	x	Auto sense external keyboard and mouse
DAh	x	x	Send data to an SMBus device
DBh	x	x	Get data from an SMBus device
DDh	x	x	Get the battery status word
DFh	x	x	04h Set clock speed to 4 MHz 08h Set clock speed to 8 MHz

Keyboard Commands

Commands written to I/O port 60h are automatically transmitted to the keyboard by the M38813 keyboard controller if the keyboard controller is not waiting for data. Both the command and data are written to I/O port 60h for two-byte commands such as EDh.

Command	Description
EDh	Set the keyboard LEDs.
EEh	Echo
EFh	Invalid command
F0h	Select alternate scan code set
F1h	Invalid command
F2h	Read ID bytes
F3h	Set typematic delay and rate
F4h	Enable keyboard
F5h	Disable keyboard and set default values.
F6h	Set defaults
F7h	Set all keys typematic
F8h	Set all keys make/break.
F9h	Set all keys make only.
FAh	Set all keys typematic make/break.
FBh	Set key type typematic
FCh	Set key type make/break.
FDh	Set key type make only.
FEh	Resend the last transmission.
FFh	BAT. Reset the defaults and buffers.

Note: Commands F7h through FDh are normally used for Character Set 3.

Auxiliary Device Commands

Auxiliary Device Command Sequence The auxiliary device command sequence is:

Step	Action
1	Write an M38813 0D4h command (Write Auxiliary Device) to I/O port 64h.
2	Write command/data to I/O port 60h.

The above sequence is executed twice for 2-byte auxiliary device commands, such as the E8h Set Resolution command.

Auxiliary Device Commands

Command	Description
E6h	Reset scaling
E7h	Set scaling
E8h	Set resolution
E9h	Status request
EAh	Set stream mode
EBh	Read data
ECh	Reset wrap mode
EDh	Invalid command
EEh	Set wrap mode
EFh	Invalid command
F0h	Set remote mode
F1h	Invalid command
F2h	Read device type
F3h	Set sampling rate
F4h	Enable auxiliary device
F5h	Disable auxiliary device
F6h	Set default values
F7h	Invalid command
F8h	Invalid command
F9h	Invalid command
FAh	Invalid command
FBh	Invalid command
FCh	Invalid command
FDh	Invalid command
FEh	Resend
FFh	Reset

Accessing M38813 RAM Locations

The keyboard controller special commands B8h through BDh allow you to read and write to any RAM location from 40h to 23Fh (a total of 512 bytes). Commands BCh and BDh read and write the most commonly used status and configuration bytes, as required. These commands provide additional ways to access highly accessed RAM locations.

Access

The 512 byte RAM area is divided into four 128 byte pages. Each page can be accessed by two variables:

- MEM_INDEX, and
- KBC_FLAGS (Bits 7 and 6).

Location

The location of the pages are

MEM_INDEX value	Page	Accesses RAM
00 to 7Fh	1	40h to BFh
80h to FFh	2, 3, or 4	Depends on the setting of KBC_FLAGS Bits 7-6.

KBC_FLAGS Bits

KBC_FLAGS Bit 7	KBC_FLAGS Bit 6	RAM locations accessed
0	0	140h to 1BFh
0	1	0C0h to 13Fh
1	0	0C0h to 13Fh
1	1	1C0h to 23Fh

Downloadable Features RAM is made accessible to support downloadable features like the keyboard matrix and the function key table.

The most important downloadable feature is the keyboard matrix. The KBC_FLAGS bits 7-6 setting default is 00, so that it accesses the page corresponding to the matrix.

Cont'd

Changing the Function Key Table To change the function key table, set bits 7-6 of KBC_FLAGS to 10 or 01 (preferably 10). Memory index values from C0h – FFh refer to 100h – 13Fh for the function shift key table.

Smart Battery Memory locations above 1C0h are used for smart battery interface handling. This memory can be accessed by setting bits 7 and 6 of KBC_Flags to 11 and setting Mem_index to 80h...FFh.

You should set the page index bits to the default (00) once either the function key table has been modified or the smart battery related data has been accessed.

5 Scan Code Controller

The keyboard controller monitors the status of all internal keyboard key switches. Based on the key pressed, it generates the scan codes according to the standard IBM scan code set specifications.

Internal and External Keyboard Support The M38813 keyboard controller supports external and internal keyboards simultaneously. When no external keyboard is installed, the keyboard controller processes and replies to all keyboard commands.

When both the internal and external keyboards are present, only internal keyboard responses are sent to the computer to avoid duplicate replies. The M38813 scan code controller is fully compatible to the IBM 101/102 key keyboard specification with the exception is that it does not support scan code set 3 and related functions.

38813 RAM The keyboard controller RAM locations from 140h to 1BFh(128 bytes) store the internal keyboard matrix. The function key table is in RAM locations 100h to 13Fh.

The keyboard matrix values correspond to the default scan code set 2 values. This matrix can have values from 00h to FFh, depending on the keys required, such as:

- function modifiable,
- extended keys (IBM-specific keys 75 to 89).

Keyboard controller special commands B8h through BBh load and edit these matrix values.

Default Matrices

Keyboard Matrix	RAM locations	Mem_index
00h, 00h, 76h, 00h, 0ch, 34h, 03h, 33h	140h-147h	80h-87h
0bh, 00h, 52h, 00h, 00h, 00h, 75h, 11h	148h-14fh	88h-8Fh
00h, 12h, 0dh, 58h, 04h, 2ch, 66h, 35h	150h-157h	90h-97h
5bh, 83h, 54h, 00h, 00h, 00h, 89h, 7ch	158h-15Fh	98h-9Fh
14h, 00h, 0eh, 05h, 06h, 2eh, 01h, 36h	160h-167h	A0h-A7h
55h, 0ah, 4eh, 71h 70h, 7dh, 6ch, 00h	168h-16Fh	A8h-AFh
00h, 00h, 16h, 1eh, 26h, 25h, 09h, 3dh	170h-177h	B0h-B7h
3eh, 46h, 45h, 78h, 07h, 7ah, 69h, 00h	178h-17Fh	B8h-BFh
00h, 00h, 15h, 1dh, 24h, 2dh, 00h, 3ch	180h-187h	C0h-C7h
43h, 44h, 4dh, 00h, 00h, 00h, 00h, 00h	188h-18Fh	C8h-CFh
96h, 00h, 1ch, 1Bh, 23h, 2Bh, 5dh, 3bh	190h-197h	D0h-D7h
42h, 4Bh, 4Ch, 00h, 00h, 00h, 00h, 00h	198h-19Fh	D8h-DFh
00h, 59h, 1ah, 22h, 21h, 2ah, 5ah, 3ah	1A0h-1A7h	E0h-E7h
41h, 49h, 00h, 77h, 00h, 00h, 00h, 00h	1A8h-1AFh	E8h-EFh
00h, 00h, 00h, 00h, 00h,32h, 29h, 31h	1B0h-1B7h	F0h-F7h
00h, 00h, 4ah, 72h, 74h, 00h, 6bh, 87h	1B8h-1BFh	F8h-FFh

Default Function-Shifted Key Matrix

Fn-Shifted Keyboard Matrix									RAM locations	Mem_ index
Un	Fn	Un	Fn	Fn	Un	Fn	Un	Values		
05h	78h	06h	07h	91h	8Bh	90h	8Ch	C0-C3h	100h-107h	C0-C7h
92h	8Fh	93h	8Eh	81h	86h	82h	87h	C4-C7h	108h-10Fh	C8-CFh
8Ah	9Ah	8Dh	7Eh	88h	84h	7Eh	08h	C8-CBh	110h-117h	D0-D7h
77h	07h	77h	98h	89h	B2h	00h	00h	CC-CFh	118h-11Fh	D8-DFh
78h	84h	07h	88h	8Ah	7Eh	05h	A6h	D0-D3h	120h-127h	E0-E7h
06h	A7h	04h	A8h	0Ch	A9h	03h	AAh	D4-D7h	128h-12Fh	E8-EFh
0Bh	ABh	83h	ACh	0Ah	ADh	01h	AEh	D8-DBh	130h-137h	F0-F7h
09h	AFh	78h	B0h	07h	B1h	1Ah	61h	DC-DFh	138h-13Fh	F8-FFh

Scan Code Generation

The American Megatrends M38813 keyboard controller supports the IBM/AT standard character set 1 and 2 scan code sequences for all keys, some enhanced keys, and some other additional keys that generate special scan code sequences. The standard IBM listing for scan code set 1 and 2 are listed below.

Scan Code Set 1 and 2

Matrix Value (Standard Character Set 2)	Description	System Scan Code (Standard Character Set 1)
00h	Error (Overrun)	FFh
01h	F9	43h
02h	F7	41h
03h	F5	3Fh
04h	F3	3Dh
05h	F1	3Bh
06h	F2	3Ch
07h	F12	58h
08h	Reserved	64h
09h	F10	44h
0Ah	F8	42h
0Bh	F6	40h
0Ch	F4	3Eh
0Dh	Tab	0Fh
0Eh	~ `	29h
0Fh	Reserved	59h
10h	Reserved	65h
11h	Left Alt	38h
12h	Left Shift	2Ah
13h	Reserved	70h
14h	Left Ctrl	1Dh
15h	Q	10h
16h	! 1	02h
17h	Reserved	5Ah
18h	Reserved	66h
19h	Reserved	71h
1Ah	Z	2Ch
1Bh	S	1Fh
1Ch	A	1Eh
1Dh	W	11h
1Eh	@ 2	03h
1Fh	Reserved	5Bh
20h	Reserved	67h

Matrix Value (Standard Character Set 2)	Description	System Scan Code (Standard Character Set 1)
21h	C	2Eh
22h	X	2Dh
23h	D	20h
24h	E	12h
25h	$ 4	05h
26h	# 3	04h
27h	Reserved	5Ch
28h	Reserved	68h
29h	Space	39h
2Ah	V	2Fh
2Bh	F	21h
2Ch	T	14h
2Dh	R	13h
2Eh	% 5	06h
2Fh	Reserved	5Dh
30h	Reserved	69h
31h	N	31h
32h	B	30h
33h	H	23h
34h	G	22h
35h	Y	15h
36h	^ 6	07h
37h	Reserved	5Eh
38h	Reserved	6Ah
39h	Reserved	72h
3Ah	M	32h
3Bh	J	24h
3Ch	U	16h
3Dh	& 7	08h
3Eh	* 8	09h
3Fh	Reserved	5Fh
40h	Reserved	6Bh
41h	< ,	33h
42h	K	25h
43h	|	17h
44h	0	18h
45h	) 0	0Bh
46h	(9	0Ah
47h	Reserved	60h
48h	Reserved	6Ch
49h	> .	34h
4Ah	? /	35h
4Bh	L	26h
4Ch	: ;	27h
4Dh	P	19h

Matrix Value (Standard Character Set 2)	Description	System Scan Code (Standard Character Set 1)
4Eh	_ -	0Ch
4Fh	Reserved	61h
50h	Reserved	6Dh
51h	Reserved	73h
52h	" '	28h
53h	Reserved	74h
54h	{ [	1Ah
55h	+ =	0Dh
56h	Reserved	62h
57h	Reserved	6Eh
58h	Caps Lock	3Ah
59h	Right Shift	36h
5Ah	Return	1Ch
5Bh	}]	1Bh
5Ch	Reserved	75h
5Dh	| , \ (US only) - # (102-key)	2Bh
5Eh	Reserved	63h
5Fh	Reserved	76h
60h	Reserved	55h
61h	\ | (102-key)	56h
62h	Reserved	77h
63h	Reserved	78h
64h	Reserved	79h
65h	Reserved	7Ah
66h	Backspace	0Eh
67h	Reserved	7Bh
68h	Reserved	7Ch
69h	1 End	4Fh
6Ah	Reserved	7Dh
6Bh	4 ←	4Bh
6Ch	7 Home	47h
6Dh	Reserved	7Eh
6Eh	Reserved	7Fh
6Fh	Reserved	6Fh
70h	0 Ins	52h
71h	, Del	53h
72h	2 ↓	50h
73h	5	4Ch
74h	6 →	4Dh
75h	8 ↑	48h
76h	Esc	01h
77h	Num Lock	45h
78h	F11	57h
79h	+	4Eh

Matrix Value (Standard Character Set 2)	Description	System Scan Code (Standard Character Set 1)
7Ah	3 PgDn	51h
7Bh	-	4Ah
7Ch	*	37h
7Dh	9 PgUp	49h
7Eh	Scroll Lock	46h
7Fh	Sys Req (84-key)	54h

Special Handling Scan Codes and Fn_Shiftable Keys

The scan codes corresponding to the special handling routines for matrix values from 80h – BFh are listed below. The table on the previous pages (beginning on page 37) lists the scan codes that correspond to matrix values 00h – 7Fh.

Set or reset the internal flag within in handling routine and finally the keyboard controller will send the special scan code according to the current shift status (Num Lock, Fn-shift, Shift, control and Alt).

Special Handled Key Codes

Matrix Value	Description	Explanation	Matrix Value	Description	Explanation
080h	Unused		0A0h	Unused	
081h	Unused		0A1h	Unused	
082h	Unused		0A2h	Unused	
083h	F7	Scan code > 80h	0A3h	L_Winkey	Win95 key
084h	SysReq	Scan code > 80h	0A4h	R_Winkey	Win95 key
085h	Unused		0A5h	Unused	
086h	Right Ctrl	Flags and E0seq	0A6h	Func 0Ah	do func task
087h	Right Alt	Flags and E0seq	0A7h	Func 0Bh	do func task
088h	PrtScrn	E0 code seq	0A8h	Func 0Ch	do func task
089h	Pause	E1 code seq	0A9h	Func 0Dh	do func task
08Ah	Insert	E0 code seq	0AAh	Func 0Eh	do func task
08Bh	Home	E0 code seq	0ABh	Func 0Fh	do func task
08Ch	PageUp	E0 code seq	0ACh	Func 10h	do func task
08Dh	Delete	E0 code seq	0ADh	Func 11h	do func task
08Eh	End	E0 code seq	0AEh	Func 12h	do func task
08Fh	Page Down	E0 code seq	0AFh	Func 13h	do func task
090h	Up Arrow	E0 code seq	0B0h	Func 14h	do func task
091h	Left Arrow	E0 code seq	0B1h	Func 15h	do func task
092h	Down Arrow	E0 code seq	0B2h	Break	E0 code seq
093h	Right Arrow	E0 code seq	0B3h	~/@	Emulated key
094h	/	E0 code seq	0B4h	= / -	Emulated key
095h	keypad Enter	E0 code seq	0B5h	+ / :	Emulated key

Matrix Value	Description	Explanation	Matrix Value	Description	Explanation
096h	Fn	Flags	0B6h	R_Win_ MenuKey	Win95 key
097h	Unused		0B7h	Func. 16h	Pulse Low P2.0
098h	Fn_Lock	Flags	0B8h	Func. 17h	Pulse Low P2.1
099h	Unused		0B9h	Func. 18h	Pulse Low P2.2
09Ah	Num Lock	Num lock acts as pad lock	0BAh	Func. 19h	Pulse Low P2.3
09Bh	Unused		0BBh	Func. 1Ah	Pulse Low P5.0
09Ch	Unused		0BCh	Func. 1Bh	Pulse Low P5.1
09Dh	Unused		0BDh	Func. 1Ch	Pulse Low P 5.2
09Eh	Unused		0BEh	Func. 1Dh	Pulse Low P5.3
09Fh	Unused		0BFh	Unused	Unused

Fn Shifted Scan Codes If the matrix value is from C0h to DFh, the current status of the function shift flag is checked and the scan code is chosen between the unshifted and Fn-shifted values as shown below. The selected scan code value goes through the same steps as the standard scan code values (0-BFh) and arrives at the final scan code sequence according to the specification. The scan code values from C0 through DFh, which are not used by default, can be used for any key combination.

Fn_Shifted Scan Codes

Scan Code Value		Matrix Value	Description	
Unshifted	**Fn_shifted**		**Unshifted**	**Fn-shifted**
005h	078h	0C0h	F1	F11
006h	007h	0C1h	F2	F12
091h	08Bh	0C2h	Left Arrow	Home
090h	08Ch	0C3h	Up Arrow	PgUp
092h	08Fh	0C4h	Down Arrow	PgDn
093h	08Eh	0C5h	Right Arrow	End
014h	086h	0C6h	Left Ctrl	Right Ctrl
011h	087h	0C7h	Left Alt	Right Alt
08Ah	09Ah	0C8h	Insert	Num lock
08Dh	07Eh	0C9h	Delete	Scroll lock
088h	084h	0CAh	PrtScrn	SysReq
07Eh	078h	0CBh	Scroll lock	F11
077h	007h	0CCh	Num lock	F12
077h	098h	0CDh	Num lock	Fn lock No
089h	0B2h	0CEh	Pause	Break

Scan Code Value		Matrix Value	Description	
000h	000h	0CFh	Unused	Unused
078h	084h	0D0h	F11	SysReq
007h	088h	0D1h	F12	Prt Scrn
08Ah	07Eh	0D2h	Insert	Scroll Lock
005h	0A6h	0D3h	F1	Func 0Ah
006h	0A7h	0D4h	F2	Func 0Bh
004h	0A8h	0D5h	F3	Func 0Ch
00Ch	0A9h	0D6h	F4	Func 0Dh
003h	0AAh	0D7h	F5	Func 0Eh
00Bh	0ABh	0D8h	F6	Func 0Fh
083h	0ACh	0D9h	F7	Func 10h
00Ah	0ADh	0DAh	F8	Func 11h
001h	0AEh	0DBh	F9	Func 12h
009h	0AFh	0DCh	F10	Func 13h
078h	0B0h	0DDh	F11	Func 14h
007h	0B1h	0DEh	F12	Func 15h
01Ah	061h	0DFh	Z	| \ (102 key)

If the matrix value is from E0h to FFh, by default it is used for a numeric keypad defined by the keys around the letter 'I'.

If the matrix value is above E0h and below FFh, it is checked against Num Lock bit set or the Fn flag set and the scan code value is chosen between the unshifted and Fn-shifted values as shown in the table below. The selected value will pass through the same steps as the standard scan code values (00h through BFh) and arrive at the final scan code sequence. The key combination used for the matrix values from E0h to FFh is preprogrammed and cannot be modified by the user.

The various configurations that are valid for numeric keypad control are explained in "External Numeric Keypad Control".

Scan Code Value		Matrix Value	Description	
Unshifted	**Fn_shifted**		**Unshifted**	**Fn-shifted**
03Bh	069h	0E0h	J	1
042h	072h	0E1h	K	2
04Bh	07Ah	0E2h	L	3
03Ch	06Bh	0E3h	U	4
043h	073h	0E4h	I	5
044h	074h	0E5h	O	6
03Dh	06Ch	0E6h	7	7
03Eh	075h	0E7h	8	8
046h	07Dh	0E8h	9	9
03Ah	070h	0E9h	M	0
049h	071h	0EAh	.	.
04Ah	094h	0EBh	/	/
04Ch	079h	0ECh	;	+
04Ch	07Ch	0EDh	;	*
045h	07Ch	0EEh	0	*
045h	079h	0EFh	0	+
04Dh	07Bh	0F0h	P	-
05Ah	095h	0F1h	Enter	keypad Enter
04Ch	07Bh	0F2h	;	-
045h	07Bh	0F3h	0	-
04Dh	079h	0F4h	P	+
04Dh	07Ch	0F5h	P	*
04Eh	07Bh	0F6h	-	-

Scan Code Value		Matrix Value	Description	
054h	095h	0F7h	{	keypad Enter
052h	07Ch	0F8h	*	*
045h	094h	0F9h	0	/
04Ah	079h	0FAh	/	+
0B3h	095h	0FBh	~/@	Keypad Enter
0B4h	07Bh	0FCh	=/-	–
0B5h	07Ch	0FDh	*/;	*
000h	000h	0FEh	Unused	Unused
55h	079h	0FFh	+/=	+

Internal Numeric Keypad Control

The numeric keypad on the internal keyboard can be configured by the OEM. The default configuration is Numlock = Numeric Pad lock with a dedicated key assigned as the Num Lock key. To reconfigure the numeric keypad, modify the internal keyboard matrix with a different numeric keypad matrix value and a different num_pad_lock flag bit.

Keypad Options The three keypad options are:

- Numeric keypad lock = Num Lock,
- Numeric keypad lock = Fn + Num Lock, and
- Numeric keypad lock = OFF

Numeric keypad lock = Num Lock In this configuration, internal numeric keypad state is same as the system Num Lock state. When the Num Lock LED is On, all keys on the numeric keypad (the keys around the letter 'I') can be used as number keys. To achieve this:

If...	Then...
a dedicated key is available on the internal keyboard,	define the Num Lock key matrix value as 9Ah (Num Lock acts as pad lock). Set the num_pad lock flag (Bit 3, of the KBC_FLAGS register).
a dedicated key is not available on the internal keyboard:	Define the function shifted key value C8h with a shifted value = to 9Ah (Num Lock acts as pad lock). Set the num_pad lock flag (Bit 3, of the KBC_FLAGS register). The function shifted key value = C8h can have any desired unshifted key value

Cont'd

Numeric keypad lock = Fn + Num Lock A dedicated Num Lock key is used in this option. The numeric keypad keys can be used as number keys if the Num lock LED is On and the user presses the Fn + Num lock combination.

To achieve this mode of operation:

Step	Action
1	Define Num lock as the function shifted key (CDh) with shifted value = 98h (fn_pad lock). The unshifted value is 77h (the standard system Num Lock).
2	Clear the num_padlock flag, bit 3, of KBC_FLAGS register.

Numeric keypad lock = OFF In this configuration, the numeric keypad can be used as number keys only by pressing Fn + numeric pad key while the system Num Lock status is On.

To get this mode of operation:

Step	Action
1	Define Num Lock as the standard system Num Lock (77h).
2	Clear the num_padlock flag (Bit 3 of the KBC_FLAGS register).

While downloading the new matrix, care must be taken to have valid numeric keypad configuration.

6 Smart Battery Support

The functions supported by Smart Battery Support include:

- smart battery interface (SMBus) support,
- battery status monitoring for alarm conditions and reporting to the system in case of an alarm event, and
- commands to the system BIOS to access battery-related data.

Battery Interface Commands

The Smart Battery Interface commands supported by Amikey/38813 include:

- DAh - Send data to an SMBus device,
- DBh - Get data from an SMBus device, and
- DDh - Get the battery status word

DAh Send Data to an SMBus Device

SMBus Protocols The keyboard controller supports four SMBus protocols. Each protocol is assigned a fixed code:

Code	Protocol
85h	WRITE_WORD
86h	READ_WORD
87h	WRITE_BLOCK
88h	READ_BLOCK

Data Formats The data format for the protocols for DAh is:

Command/Data	Destination	Description
Protocol: READ_WORD		
DAh	I/O port 64h	
xxh	I/O port 60h	Protocol
YYh	I/O port 60h	Slave Address
ZZh	I/O port 60h	Device Command
Protocol: READ_BLOCK		
DAh	I/O port 64h	
xxh	I/O port 60h	Protocol
YYh	I/O port 60h	Slave Address
ZZh	I/O port 60h	Device Command
Protocol: WRITE_WORD		
DAh	I/O port 64h	
xxh	I/O port 60h	Protocol
YYh	I/O port 60h	Slave Address
ZZh	I/O port 60h	Device Command
Data	I/O port 60h	Byte 0
Data	I/O port 60h	Byte 1
Protocol: WRITE_BLOCK		
DAh	I/O port 64h	
xxh	I/O port 60h	Protocol
YYh	I/O port 60h	Slave Address
ZZh	I/O port 60h	Device Command
Data	I/O port 60h	byte0 (COUNT (n))
	I/O port 60h	Byte 1
...	...	...
...	...	...
Data	I/O port 60h	Byte (n)

DBh Get Data from SMBus Device

BIOS command DBh is written to I/O port 64h. If the previous transaction was successful, the status byte contains 00h.

Data Formats The data format for the protocols for DBh is:

Command/Data	Destination	Description
Protocol: READ_WORD		
DBh	I/O port 64h	
Status Byte	I/O port 60h	
Byte 0	I/O port 60h	
Byte 1	I/O port 60h	
Protocol: READ_BLOCK		
DAh	I/O port 64h	
Status Byte	I/O port 60h	If the previous transaction was erroneous, the status byte contains an error code.
Byte 0	I/O port 60h	Number of bytes
Byte 1	I/O port 60h	
...	...	...
...	...	...
Byte n	I/O port 60h	
Protocol: WRITE_WORD		
DBh	I/O port 64h	
Status Byte	I/O port 60h	
Protocol: WRITE_BLOCK		
DBh	I/O port 64h	
Status Byte	I/O port 60h	

DDh Get Battery Status Word

The keyboard controller returns two bytes of data in response to the DDh command. The low byte is returned first and the high byte is returned next. If polling is disabled or polling fails, the high byte is returned with a general timeout error in the keyboard controller status word (bit 6 of port 64h).

Error Codes for Commands DAh, DBh, and DDh

The valid error codes for commands DAh and DBh are:

Error Code	Description
10h	Slave address not acknowledged
11h	device detected error/data byte not acknowledged
18h	Timeout error
19h	Unsupported protocol specified
1Ah	SMBus busy

Smart Battery Status Monitoring

AMIKey periodically polls for the smart battery status and verifies the status against the alarm conditions that are sent to the system BIOS (if alarm conditions are sent to the BIOS).

The keyboard controller sends the battery status command (16h) to the smart battery once every Batt_Delay_time + 1 second, as long as Batt_Delay_time is a non-zero value. The battery responses are stored in keyboard controller RAM.

The high byte in the status word is checked to see if any alarm has been reported or if the previously reported alarm has been removed.

If either of these conditions exist and (depending on the alarm bits, and the mask status), the battery alarm task is executed.

Memory Map for Battery Management

Symbol	RAM Location	Description
Smb_Prot_addr	9Bh	SMBus protocol storage
Status_byte	9Ch	SMBus operation status storage
Batt_Poll_delay_time	9Dh	Battery alarm polling delay time in seconds
Bat_Alrm_Regstr1	9Eh	Battery Alarm bits mask register Bit 7 Enable Over_charged_Alarm notify Bit 6 Enable Terminate_charge_alarm notify Bit 5 Reserved Bit 4 Enable Over_Temp_alarm notify Bit 3 Enable Terminate discharge alarm notify Bit 2 Reserved Bit 1 Enable Remaining Capacity alarm notify Bit 0 Enable Remaining Time alarm notify
Bat_Alrm_Regstr2	9Fh	Battery Alarm bits removed mask register Bit 7 Enable Over_charged Alarm removed Bit 6 Enable Terminate_charge Alarm removed Bit 5 Reserved Bit 4 Enable Over_Temp_alarm removed Bit 3 Enable Terminate discharge alarm removed Bit 2 Reserved Bit 1 Enable Remaining Capacity alarm removed Bit 0 Enable Remaining Time alarm removed
Bat_Alrm_task_reg	A0h	Task executed in an alarm condition
Smb_Low_timeout	1C2h	SMBus timeout value(default=25ms)
Prev_Bat_Status_high	1C4h	Previous Battery Status high byte
Batt_Poll_timer	1C5h	Battery Polling timer
Bat_Status_low	1C6h	Battery Status Low Byte
Bat_Status_high	1C7h	Battery Status High Byte
Batt_Poll_fail_cnt	1C8h	Battery polling fail counter

Symbol	RAM Location	Description
Batt_Poll_retry_cnt	1C9h	Battery Polling retry counter (default 16)
Poll_finish_flag	1F5h	Flag byte used for status report
Smbus_data_buffer	1D0h – 1D2h	SMBus buffer for data storage (35 bytes)

Index

Q

R

S

T

U

W