Using Game and MIDI Ports in the PC87363 and PC87366

National Semiconductor
Application Note AN-1130
Yuval Margalit
March 1999

National Semiconductor's new advanced SuperI/O™ devices, PC87363 and PC87366, introduce Game and Musical Instrument Digital Interface (MIDI) Port inputs. This document describes suggested external circuits and programming procedures to operate these new modules. It also describes how to exploit these two modules with Microsoft's SideWinder® Force Feedback Pro joystick.

For further information, see the PC87363 or PC87366 Datasheet.

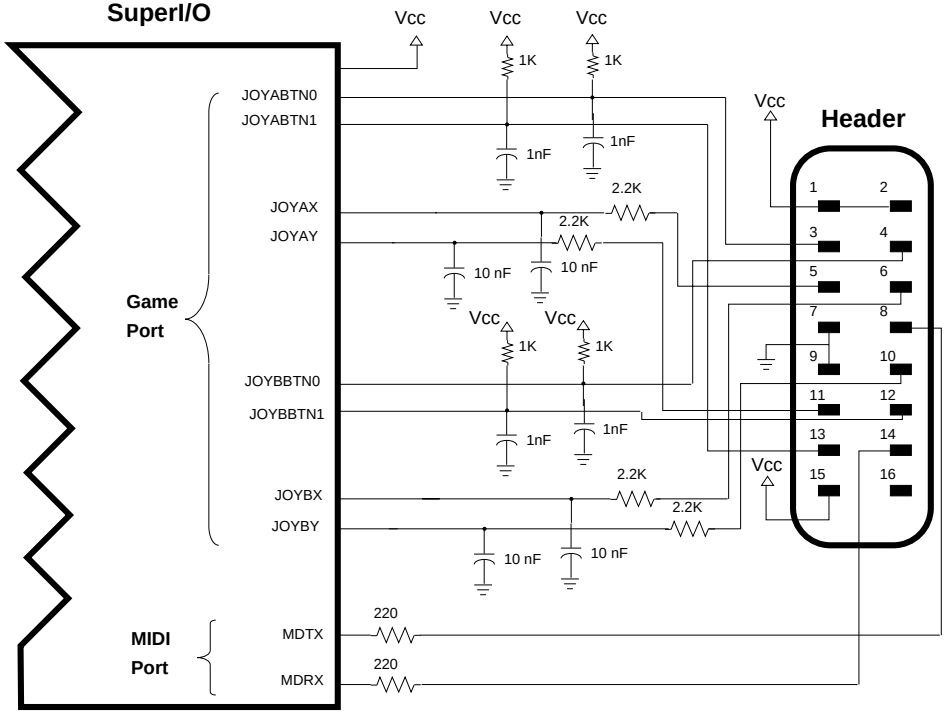
CIRCUIT DESCRIPTION

The figure below illustrates an external circuit that can be used to connect the Game and the MIDI Port signals to the Game and MIDI Port connector, a dual-row 8-pin header. A standard Game Port cable connects this header to an external DB15 connector.

The values shown for the resistors and capacitors connected to the joystick axis signals (JOYAX, JOYAY, JOYBX and JOYBY) comply with standard joystick specifications, such as Microsoft's SideWinder Force Feedback Pro and Game Pad. These components should be within ±5% of the stated values. Other joysticks may require different values.

The values shown for the components connected to the remaining joystick and MIDI signals are not critical, and can be within ±10% of their stated values.

Note: MIDI port pin MDRX should have either an external pull-up resistor of 2.2 Kohm, or use the SuperI/Os internal pull-up. See "Game Port and MIDI Programming Guidelines" on page 5 for configuration information.



LOGICAL DEVICE CONFIGURATION

Access to the Game and MIDI Configuration register is through the SuperI/O Index and Data registers. They are located respectively at either I/O addresses 2Eh, 2Fh (when BADDR strap is set to 0) or 4Eh, 4Fh (when BADDR strap is set to 1).

Logical Device Number

The PC87363 and PC87366 Game and MIDI Ports are assigned with logical device numbers 0Bh and 0Ch respectively. To enable access to the configuration registers of the modules, write the logical device number in the PnP Standard Control register at index 07h.

Base Address

The Game and MIDI I/O Port base addresses must be defined in the Game and MIDI Port Configuration registers at indexes 60h (MSB) and 61h (LSB). The Game Port base address must be 16-byte aligned (four LSBs are 0). The MIDI base address must be 8-byte aligned (three LSBs are 0).

Interrupt

The MIDI Port and the Game Port, when in enhanced mode, may have dedicated interrupts, whose numbers must be defined at index 70h.

Activation

To activate the Game and MIDI Port logical devices, write 01h in the Activate registers at index 30h.

Game and MIDI Port Configuration Registers

The Game and MIDI Port Configuration registers are read/write registers located at index F0h.

Bit 2 of the Game Port Configuration register controls the internal pull-up resistors on the Button pins (GPIO22/JOYABTN0, GPIO23/JOYABTN1, GPIO26/JOYBBTN0 and GPIO27/JOYBBTN1). All other bits are reserved.

Bit 2 of the MIDI Configuration register controls the internal pull-up resistor on pin 126 (GPIO31/MDRX). Bit 0 defines its state (TRI-STATE, or not). All other bits are reserved.

Multiplexed Pins

The PC87363 and PC87366 Game and MIDI Port pins are multiplexed with other pins. Multiplexing control is performed through the SuperI/O Configuration 4 register (SIOCF4) at index 24h. After the PC87363 or PC87366 is reset, all multiplexed pins are configured as GPIO pins, and the SIOCF4 register value is 00h. To select game and MIDI functionality, set the following bits of the SIOCF4 register to 1:

- Bit 4, to enable pins 117-124 as joystick pins.
- Bit 5, to enable pins 125, 126 as MIDI pins.

USING THE GAME PORT

Since there is currently no application that supports the Game Port Enhanced mode, only the Legacy mode of operation and the Legacy Status register are described in this document.

Game Port Functional Operation

The game device indicators are monitored by polling their momentary status via the Game Port Legacy Status register (GMPLST).

Performing a write access to offset 1 in the Game Port address space initiates a read of the game device position. This write access causes the Game Port to release the JOYnX,Y pins, and to set the corresponding bit in the GMPLST register to 1. To capture the position indicated by the game device, the software must poll the GMPLST register and measure the time it takes for the JOYnX,Y to go high. This measurement is performed by measuring the time during which an axis bit is 1.

Reading the status of the buttons of the game device is done by polling the GMPLST register and looking for changes in the status bits of the JOYnBTN0,1 pins.

The Game Port does not debounce the input signals. If necessary, the software must implement debouncing.

Game Port Legacy Status Register (GMPLST)

Reading this register returns the status and the state of Device A and B Button and Axis pins, as defined in the table below. Writing to the offset of this register initiates a game device position reading process by forcing a low pulse to be driven on the axis pins.

Access to the GMPLST register is direct. The address is calculated by adding the register offset (01h) to the Game Port base address.

Location: 01h

Type: Read Only

Bit	7	6	5	4	3	2	1	0
Name	**Device B Button 1 Pin Status**	**Device B Button 0 Pin Status**	**Device A Button 1 Pin Status**	**Device A Button 0 Pin Status**	**Device B Y-Axis Pin Status**	**Device B X-Axis Pin Status**	**Device A Y-Axis Pin Status**	**Device A X-Axis Pin Status**
Reset	X	X	X	X	X	X	X	X

Bit	Description
7	**Device B Button 1 Pin Status.** This bit directly reflects the status of Device B Button 1 input pin. 0: Low 1: High
6	**Device B Button 0 Pin Status.** This bit directly reflects the status of Device B Button 0 input pin. 0: Low 1: High
5	**Device A Button 1 Pin Status.** This bit directly reflects the status of Device A Button 1 input pin. 0: Low 1: High
4	**Device A Button 0 Pin Status.** This bit directly reflects the status of Device A Button 0 input pin. 0: Low 1: High
3	**Device B Y-Axis Pin Status.** This bit reflects the state of Device B Y-axis input pin. 0: JOYBY pin is released for charging 1: JOYBY pin is driven low
2	**Device B X-Axis Pin Status.** This bit reflects the state of Device B X-axis input pin. 0: JOYBX pin is released for charging 1: JOYBX pin is driven low
1	**Device A Y-Axis Pin Status.** This bit reflects the state of Device A Y-axis input pin. 0: JOYAY pin is released for charging 1: JOYAY pin is driven low
0	**Device A X-Axis Pin Status.** This bit reflects the status of Device A X-axis input pin. 0: JOYAX pin is released for charging 1: JOYAX pin is driven low

USING THE MIDI PORT

The MIDI Port is an asynchronous receiver/transmitter that uses a two-wire, bi-directional, relatively slow communication channel to transmit and receive data bytes to or from MIDI-compliant devices, according to a predefined communication protocol. The MIDI Port is compatible with MPU-401 UART mode. The MIDI was originally defined to establish a standard interface between computers and digital musical instruments such as synthesizers, and has become the de facto standard for this purpose. However, the MIDI is also commonly used for other purposes, such as communicating with advanced game devices. The MIDI Port serves as a communication pipe between software and a MIDI device, which must interpret the data they exchange, and act accordingly.

MIDI Data In Register (MDI)

This read register is used for reading data received by the MIDI Port, and status information returned by the MIDI Port in response to a previously issued command. When the FIFOs of the MIDI Port are enabled, reading from this offset returns the next byte from the Receive FIFO.

Location: Offset 00h

Type: Read

Bit	7	6	5	4	3	2	1	0
Name	**Data In**							
Reset	X	X	X	X	X	X	X	X

MIDI Data Out Register (MDO)

This write register is used for writing data to be transmitted by the MIDI Port. When the FIFOs of the MIDI Port are enabled, writing to this offset puts the data byte into the Transmit FIFO.

Location: Offset 00h

Type: Write

Bit	7	6	5	4	3	2	1	0
Name	**Data Out**							
Reset	X	X	X	X	X	X	X	X

MIDI Status Register (MSTAT)

This read register provides status information on the functional blocks of the MIDI Port.

Location: Offset 01h

Type: Read

Bit	7	6	5	4	3	2	1	0
Name	**Rx Buffer Empty**	**Tx Buffer Full**	**Rx FIFO Full**	**MIDI Port Operation Mode**	**Rx Overrun Error**	**Tx FIFO Not Empty**	**Reserved**	
Reset	1	0	0	0	0	0	0	0

Bit	Description
7	**Rx Buffer Empty.** When set to 1, it indicates that the Receive Buffer in Pass-Thru mode, or the FIFO in UART mode, is empty. When set to 0, it indicates that the Receive Buffer or FIFO contain data that can be read via the MDI register. 0: Not empty 1: Empty (default)
6	**Tx Buffer Full.** When set to 1, it indicates that the Transmit Buffer or FIFO cannot accept any more data. When set to 0, it indicates that the Transmit Buffer or FIFO can accept more data written to the MDO register. 0: Not full (default) 1: Full
5	**Rx FIFO Full.** When set to 1, it indicates that the Receive FIFO cannot accept any more received data bytes. When set to 0, it indicates that the Receive FIFO can accept more received data bytes. This bit is forced to 0 when the FIFOs are disabled. 0: Not full or disabled (default) 1: Full
4	**MIDI Port Operation Mode.** When set to 1, it indicates that the MIDI Port is currently operating in UART mode. When set to 0, it indicates that the MIDI Port is currently operating in Pass-Thru (non-UART) mode. 0: Pass-Thru mode (default) 1: UART mode
3	**Rx Overrun Error.** This bit is cleared to 0 when the MSTAT register is read. An overrun error is defined as the state in which one or more data bytes have been received by the MIDI Port while the Receive Buffer, or FIFO, was full. 0: No overrun error (default) 1: Overrun error
2	**Tx FIFO Not Empty.** This bit is forced to 0 when the FIFOs are disabled. 0: Empty or disabled (default) 1: Not empty
1-0	**Reserved**

MIDI Command Register (MCOM)

This write register is a port via which commands are issued by the host to the MIDI Port.

Location: Offset 01h

Type: Write

Bit	7	6	5	4	3	2	1	0
Name	**Command Byte**							
Reset	X	X	X	X	X	X	X	X

MIDI Control Register (MCNTL)

This register controls enhanced MIDI functions.

Location: Offset 02h

Type: Read/Write

Bit	7	6	5	4	3	2	1	0
Name	**Loopback Mode Enable**	**MIDI Thru Enable**	**Reserved**	**Pass-Thru Transmit Enable**	**Rx Data Ready Interrupt Enable**	**MDTX Pin Masking Enable**	**Tx Buffer Empty Interrupt Enable**	**Rx FIFO Enable for Pass-Thru Mode**
Reset	0	0	0	0	1	0	0	1
Required			0					

Bit	Description
7	**Loopback Mode Enable**. When enabled, the MIDI receive signal is internally connected to the MIDI transmit signal. 0: Disabled (default) 1: Enabled
6	**MIDI Thru Enable**. When enabled, the MDRX pin is internally connected to the MDTX pin, which then reflects the MIDI receive signal. When disabled, the MDTX pin is driven with data coming from the MIDI Port transmit engine. 0: Disabled (default) 1: Enabled
5	**Reserved**
4	**Pass-Thru Transmit Enable.** When enabled, data is transmitted in Pass-Thru (non-UART) mode. 0: Disabled (default) 1: Enabled
3	**Rx Data Ready Interrupt Enable.** When enabled, an interrupt request is asserted in response to a Receive Data Ready event. 0: Disabled 1: Enabled (default)
2	**MDTX Pin Masking Enable.** When enabled, the MDTX pin is constantly driven high by the MIDI Port. When disabled, MDTX serves as the MIDI Port transmit line. 0: Disabled (default) 1: Enabled
1	**Tx Buffer Empty Interrupt Enable.** When enabled, an interrupt request is asserted in response to a Transmit Buffer Empty event. 0: Disabled (default) 1: Enabled
0	**Rx FIFO Enable for Pass-Thru Mode**. When this bit is set to 1, the Receive FIFO is enabled in Pass-Thru mode. This bit is ignored in UART mode. 0: Disabled 1: Enabled (default)

GAME PORT AND MIDI PROGRAMMING GUIDELINES

The following code sequence is an example of the BIOS code for Game and MIDI Port configuration. The sample code is written in a pseudo Assembler language, and serves to illustrate Game Port and MIDI programming principles.

In this code, the port base addresses and the MIDI interrupt are defined, the multiplexed pins are selected to route the desired function signals, and the modules are activated.

BADDR refers to the PC87363 or the PC87366 device base address (which is the Index register address, and therefore BADDR + 1 is the Data register address).

```
out BADDR, 24h          ; SIOCF4 register
out BADDR + 1, 30h      ; Select pins 125, 126 as MIDI and pins 117-124 as Game Port
out BADDR, 07h          ; Logical Device selection register
out BADDR + 1, 0Bh      ; Select Game Port
out BADDR, 60h          ; Set Game Port base address to 200h (GMPLST is located at 201h)
out BADDR + 1, 02h
out BADDR, 61h
out BADDR + 1, 00h
out BADDR, 30h          ; Activate Game Port using read-modify-write
in  al, BADDR + 1
and al, FEh
or  al, 01h
out BADDR + 1, al
out BADDR, 07h          ; Logical Device selection register
out BADDR + 1, 0Bh      ; Select MIDI
out BADDR, 60h          ; Set MIDI base address to 330h
out BADDR + 1, 03h
out BADDR, 61h
out BADDR + 1, 30h
out BADDR, 70h          ; Set MIDI interrupt to IRQ9 using read-modify-write
in  al, BADDR + 1
and al, F0h
or  al, 09h
out BADDR + 1, al
out BADDR, F0h          ; Enable internal pull-up resistor on pin 126
out BADDR + 1, 04h
out BADDR, 30h          ; Activate MIDI Port using read-modify-write
in  al, BADDR + 1
and al, FEh
or  al, 01h
out BADDR + 1, al
out SIO_MIDI_BASE + 2, 0B;Enable an interrupt request assertion in response to a Transmit
                          Buffer Empty event
```

USING THE MICROSOFT SIDEWINDER FORCE FEEDBACK PRO WITH GAME AND MIDI PORTS

The Microsoft SideWinder Force Feedback Pro is an enhanced joystick with an output channel to control the joystick's motor. This output channel is implemented by the MIDI Port. This section describes the required steps to control this joystick through the PC87363 or PC87366 SuperI/O.

Installation

1. Locate the 15-pin Game Port on the back of your computer, and insert the joystick's connector into it.
2. Locate the power connector on the left side of the joystick and insert the cable from the AC adapter into it.
3. Turn on your system.
4. Configure and activate the Game and MIDI Port as described in the section "Game Port and MIDI Programming Guidelines".
5. Install SideWinder software, Version 3.0 or higher. You can download this software from Microsoft's website: www.microsoft.com/sidewinder

Configuring Windows for Game and MIDI Port Drivers

The procedure below applies to WIndows 98. Minor modifications may be required for Windows 95. Note that "click" indicates a single click on the left mouse button.

1. To access the Control Panel, click on the following sequence of options:
   `Start` (located on the bottom left side of the screen)
   `Settings`
   `Control Panel`
2. Double click on `Add New Hardware`.
3. The following steps apply to the `Add New Hardware Wizard` window:
   A. Click twice on `Next`. In response to `Do you want Windows to search for new hardware?`, click on `No`. (`Yes` is already selected as the default.) Click on `Next`.
   B. Click on `Sound, video and game controllers` hardware type, and then on `Next`.
   C. Select `Microsoft` from the Manufacturer's list, and `Game port joystick` from the Models list to configure the Game Port. To configure the MIDI Port, begin with step 2 of this procedure and select instead `MPU-401 compatible` from the Models list Click on `Next`.
4. In the `Resources` window, click on `Next`.
5. Click on `Finish`.
6. Click on `No` in response to `Restart`.
7. The following steps perform a resource check to verify if the base address and interrupt request match previously set values.
   A. Return to the Control Panel, and double click on the `System` icon.
   B. In the `Device Manager` window, click on the + symbol to the left of `Sound, video and game controllers.`
   C. Click on `MPU-401 compatible` and then on `Resources`.
   D. Set the identical base address and interrupt request that you set previously.
8. Restart Windows with the new configuration values by clicking on the following sequence of options:
   `Start`
   `Shutdown`
   `Restart`
   `Yes.`

Configuring Windows for the SideWinder Joystick

1. Access the Control Panel, and double click on the `Game Controllers` icon.
2. Click on `Add`.
3. Scroll down, select `Microsoft SideWinder Force Feedback Pro` and click on `OK`.
4. Click on `OK` to close the `Game Controllers` dialog box.

Loading SideWinder Software

Perform the following procedure after completing the SideWinder installation procedure provided with the joystick. You may then run the application.

1. Run the `SideWinder Profile Activator`.
2. In the `Getting Started` window, click on `OK`.
3. Add the desired profile either by selecting if from the list or by clicking on the following sequence of options:
   `File`
   `Add Profile`
   `Add`
4. Activate the profile by clicking on the button in the `Activate` column.
5. Click on `File` and then on `Exit`.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

www.national.com