

Watchdog

Contents

Features

Description

Ordering Code

Check List

Software Installation

Installing Drivers in OS Linux

Installing Drivers in OS FreeBSD

Diagnostic Utility

Registers Description

DATA Register

STS Register

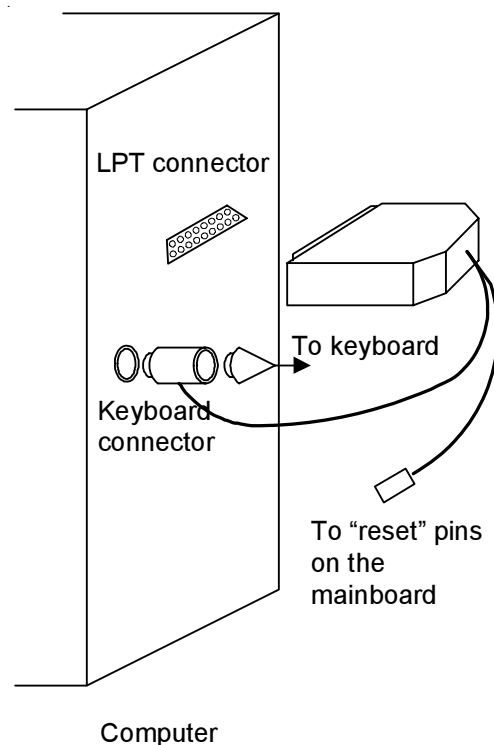
Code Example

Features

- Computer hang-up protection
- Activation time programmed in range 100 msec to 60 sec
- Connects to standard LPT port
- Powered from AT or PS/2 keyboard port
- Drivers for OS Linux and OS FreeBSD
- Open source API

Description

Watchdog is designed for use on servers working in the round-the-clock unattended mode. It restores an operation of the server performing the hardware reset of a computer in case of the operating system hang-up.



The watchdog is fitted into a standard DB-25 connector, with two additional cables. It must be connected to the LPT port of a computer, additional cables are connected to RESET contacts on the motherboard, and to the keyboard connector.

After computer powerup or cold reset, the watchdog is in a passive state and does not affect the operation of a computer. After the operating system startup, the watchdog driver will initialize it and set up the timeout value: 100 msec, 1 sec, 5 sec, 10 sec, 20 sec, 40 sec or 60 sec. During the normal system operation the watchdog software periodically signals the watchdog that the computer is alive. In a case of

operating system hang up, after the timeout elapsed, the watchdog will perform a cold reset of the computer.

Ordering Codes

There are several models of the Watchdog:

- Wdog/AT - powered from AT keyboard port.
- Wdog/PS - powered from PS/2 keyboard port.

Check List

The package should contain the following items:

- Watchdog
- Floppy disk or CD with software
- User's manual

Software Installation

Software disk contains drivers for Linux and FreeBSD operating systems. It also contains diagnostic utility (for MS-DOS, source code included). Please consult readme.txt file for full list of disk contents. You could copy from disk and extract files using utilities mread (mtools package) and tar. For example:

```
mread a:wdog.tgz .
tar xvzf wdog.tgz
```

Installing Drivers in Linux OS

Extract software package with the Watchdog drivers and go in to directory named linux. Use following command to compile and install drivers:

```
make install
```

This command will install following files:

- /lib/modules/lpww.o loadable driver module
- /etc/rc.d/init.d/wdog startup script
- /sbin/hwreset cold reset utility

Restart system. At every system startup it will run background process which will periodically communicate with the Watchdog. If this background process would be stopped, for example in case of

system hang-up, after the timeout elapsed, the Watchdog will perform a cold reset of the computer. Default timeout is 60 sec. The value of timeout interval can be changed in /etc/rc.d/init.d/wdog file.

You can perform cold reset using /sbin/hwreset utility. You need superuser permissions to perform this action.

Note: Watchdog driver and standard LPT port driver are mutually exclusive. To install Watchdog driver you need to disable LPT port driver in OS kernel.

Installing Drivers in FreeBSD OS

Extract software package with the Watchdog drivers and go in to directory named freebsd. Use following command to compile and install drivers:

```
make install
```

This command will install the following files:

- /modules/lpww_mod.ko loadable driver module
- /usr/local/etc/rc.d/wdog.sh startup script
- /sbin/hwreset cold reset utility

Restart the system. At every system startup it will run background process which would periodically communicate with the Watchdog. If this background process would be stopped, for example in case of system hang-up, after the timeout elapsed, the Watchdog will perform a cold reset of the computer. Default timeout is 60 sec. The value of timeout interval can be changed in /usr/local/etc/rc.d/wdog.sh file.

You can perform cold reset using /sbin/hwreset utility. You need superuser permissions to perform this action.

Note: Watchdog driver and standard LPT port driver are mutually exclusive. To install Watchdog driver you need to disable LPT port driver in OS kernel.

Diagnostic Utility

You can find wdog.exe diagnostic utility in directory named msdos. It could run under following operation systems: DOS/Win3x/Win9x. Make sure that you

have no running printer driver. After you run the utility it will find Watchdog and will periodically communicate with it. If you end program by pressing any key, a cold reset would be performed in a while. Source code of wdog.exe utility can be used as an example code of Watchdog programming. The Turbo C 2.0 compiler was used for compiling the wdog.exe utility.

Register Description

Watchdog is controled via LPT port registers. The base address of LPT port depends on mainboard settings:

378h	— LPT1
278h	— LPT2
3BCh	— LPT3

Registers have the following offset addresses:

Register	Access	Offset	Description
DATA	write 8	0	Data register
STS	read 8	1	Status register

DATA register

7	6	5	4	3	2	1	0
x	x	x	x	x	IN	/CLK	/CS

Bits 7-3 — reserved.

Bit 2 — IN — serial data for Watchdog.

Bit 1 — /CLK — inverted data strobe.

Bit 0 — /CS — inverted chip select.

STS register

7	6	5	4	3	2	1	0
/OUT	x	x	x	x	x	x	x

Бит 7 — /OUT — acknowledgment of the next bit of information.

Bits 6-0 — reserved.

Commands for Watchdog are send as a serial code via /CS, /CLK, IN bits of DATA register. The /OUT bit of STS register is used to monitor readiness of Watchdog to receive next bit of information.

Watchdog commands are 8 bit wide. Bits are transmitted starting from high to low. Comman codes are listed below:

Code	Command
C8h	Set 100 msec timeout
C9h	Set 1 sec timeout
CAh	Set 5 sec timeout
CBh	Set 10 sec timeout
CCh	Set 20 sec timeout
CDh	Set 40 sec timeout
CEh	Set 60 sec timeout
CFh	Reset timer (system activity acknowledgement)

While idle, bits /CS and /CLK must be set to 1. Write cycle starts from zeroing the /CS bit. Wath Dog will respond by setting /OUT bit. Bit transmission cycle consists of four steps:

- 1) Set IN to the bit to be transmitted
- 2) Clear /CLK bit
- 3) Set /CLK bit
- 4) Compare inversed value of transmitted bit (IN) with /OUT (echoing)

For command transmission you must repeat these steps eight times. At the end of byte transmission you must set /CS bit.

Code example

The C code below starts Watchdog with 100 ms timeout command C8h).

```
#define DATA 0x378      /* LPT1+0 */
#define STS 0x379       /* LPT1+1 */

outb (DATA, 7);        /* initialization */
outb (DATA, 2);        /* set CS */
/* бит 7 = 1 */
outb (DATA, 2);        /* set data */
outb (DATA, 0);        /* set CLK */
outb (DATA, 2);        /* reset CLK, data */
if (! (inb (STS) & 0x80)) error ();
/* бит 6 = 1 */
outb (DATA, 2);        /* set data */
outb (DATA, 0);        /* set CLK */
outb (DATA, 2);        /* reset CLK, data */
if (! (inb (STS) & 0x80)) error ();
/* бит 5 = 0 */
outb (DATA, 6);        /* set data */
outb (DATA, 4);        /* set CLK */
outb (DATA, 6);        /* reset CLK, data */
if (inb (STS) & 0x80) error ();
/* бит 4 = 0 */
outb (DATA, 6);        /* set data */
outb (DATA, 4);        /* set CLK */
outb (DATA, 6);        /* reset CLK, data */
if (inb (STS) & 0x80) error ();
/* бит 3 = 1 */
outb (DATA, 2);        /* set data */
outb (DATA, 0);        /* set CLK */
outb (DATA, 2);        /* reset CLK, data */
if (! (inb (STS) & 0x80)) error ();
/* бит 2 = 0 */
outb (DATA, 6);        /* set data */
outb (DATA, 4);        /* set CLK */
outb (DATA, 6);        /* reset CLK, data */
if (inb (STS) & 0x80) error ();
/* бит 1 = 0 */
outb (DATA, 6);        /* set data */
outb (DATA, 4);        /* set CLK */
outb (DATA, 6);        /* reset CLK, data */
if (inb (STS) & 0x80) error ();
/* бит 0 = 0 */
outb (DATA, 6);        /* set data */
```

```
outb (DATA, 4);        /* set CLK */
outb (DATA, 6);        /* reset CLK, data */
if (inb (STS) & 0x80) error ();

outb (DATA, 3);        /* reset CS */
```