

# Сторожевой таймер

Руководство пользователя

## Указания по технике безопасности



Восклицательный знак в треугольнике служит для предупреждения пользователя о наличии важных инструкций по эксплуатации и обслуживанию устройства.

При монтаже, эксплуатации и техническом обслуживании устройства следует соблюдать действующие правила техники безопасности. Работы по установке, техническому обслуживанию и ремонту должны выполняться только квалифицированным и уполномоченным персоналом.

# Содержание

<b>Раздел 1. Введение .....</b>	<b>4</b>
1.1. Применение .....	4
1.2. Принцип действия .....	4
1.3. Характеристики .....	4
1.4. Исполнение .....	5
1.5. Коды заказа .....	5
1.6. Комплектность .....	5
<b>Раздел 2. Установка программного обеспечения .....</b>	<b>6</b>
2.1. Установка в ОС Linux .....	6
2.2. Установка в ОС FreeBSD .....	7
2.3. Установка в ОС Microsoft Windows 2000/XP/2003 .....	7
Параметры драйвера .....	7
Установка драйвера .....	9
Управление из программы пользователя .....	18
Технические подробности работы драйвера .....	19
2.4. Утилита тестирования .....	19
<b>Раздел 3. Программный интерфейс .....</b>	<b>20</b>
3.1. Регистр DATA .....	20
3.2. Регистр STS .....	20
3.3. Пример запуска таймера .....	22

# Раздел 1. Введение

## 1.1. Применение

Сторожевой таймер предназначен для использования в серверах, работающих в круглосуточном необслуживаемом режиме. Он восстанавливает работу сервера путем аппаратного сброса компьютера при сбоях или “зависании” операционной системы.

## 1.2. Принцип действия

После включения питания или сброса компьютера сторожевой таймер находится в пассивном состоянии и не оказывает влияния на функционирование компьютера. При загрузке операционной системы программное обеспечение (драйвер) таймера производит запуск сторожевого таймера и задает время срабатывания: 100 мсек, 1 сек, 5 сек, 10 сек, 20 сек, 40 сек либо 60 сек. После запуска драйвер выполняет периодические обращения к таймеру. При отсутствии обращений, например, при “зависании” компьютера или сбое программного обеспечения, через гарантированное время произойдет аппаратный сброс компьютера.

## 1.3. Характеристики

- Защита компьютера от зависания.
- Время срабатывания от 100 мсек до 60 сек.
- Подключение к стандартному параллельному порту компьютера.
- Питание от порта клавиатуры AT или PS/2.
- Драйверы для ОС Linux, FreeBSD, Microsoft Windows 2000/XP/2003.
- Открытый программный интерфейс.

## 1.4. Исполнение

Сторожевой таймер размещается в корпусе стандартного разъема DB-25 и имеет два дополнительных кабеля. Таймер вставляется в разъем параллельного порта компьютера, дополнительные кабели подключаются к контактам RESET на материнской плате и к разъему клавиатуры.

## 1.5. Коды заказа

Сторожевой таймер выпускается в двух вариантах исполнения:

- **WDOG/AT** – с питанием от клавиатурного разъема AT;
- **WDOG/PS** – с питанием от клавиатурного разъема PS/2.

## 1.6. Комплектность

В комплект поставки входят:

- сторожевой таймер;
- дискета с программным обеспечением;
- руководство пользователя.

## Раздел 2. Установка программного обеспечения

В комплект поставки адаптера входят драйверы для операционных систем Linux, FreeBSD и Microsoft Windows, а также утилита тестирования таймера (с исходными текстами). Драйверы для FreeBSD, Linux и Microsoft® Windows® находятся на поставляемой дискете. В файле `readme-r.txt` приведен полный список драйверов. Прочитать файлы с дискеты можно утилитой `mread` (пакет `mtools`). Для распаковки используйте утилиту `tar`. Например:

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

### 2.1. Установка в ОС Linux

Распакуйте дистрибутив программного обеспечения таймера. Для установки войдите в каталог `linux/` и выполните команду:

```
make install
```

В систему будут установлены:

- загружаемый драйвер `/lib/modules/lpwd.o`;
- утилита запуска `/etc/rc.d/init.d/wdog`;
- утилита аппаратной перезагрузки `/sbin/hwreset`.

Перезагрузите компьютер. При старте системы будет автоматически запущен фоновый процесс `wdog`, периодически обращающийся к таймеру. Если произойдет останов фонового процесса, например в случае «зависания» операционной системы, по истечении определенного интервала времени сторожевой таймер произведет аппаратный сброс компьютера. По умолчанию интервал времени составляет около 60 секунд. Значение интервала можно изменить в файле `/etc/rc.d/init.d/wdog`. При необходимости аппаратный сброс компьютера можно также выполнить командой `/sbin/hwreset`. Для этого требуются полномочия суперпользователя.

## 2.2. Установка в ОС FreeBSD

Распакуйте дистрибутив программного обеспечения таймера. Для установки войдите в каталог `freebsd/` и выполните команду:

```
make install
```

В систему будут установлены:

- загружаемый драйвер `/modules/lpwd_mod.ko`;
- утилита запуска `/usr/local/etc/rc.d/wdog.sh`;
- утилита аппаратной перезагрузки `/sbin/hwreset`.

Перезагрузите компьютер. При старте системы будет автоматически запущен фоновый процесс `wdog`, периодически обращающийся к таймеру. Если произойдет останов фоновой процесс, например в случае «зависания» операционной системы, по истечении определенного интервала времени сторожевой таймер произведет аппаратный сброс компьютера. По умолчанию интервал времени составляет около 60 секунд. Значение интервала можно изменить в файле `/usr/local/etc/rc.d/wdog.sh`.

При необходимости аппаратный сброс компьютера можно также выполнить командой `/sbin/hwreset`. Для этого требуются полномочия суперпользователя.

## 2.3. Установка в ОС Microsoft® Windows® 2000/XP/2003

### Параметры драйвера

Параметры, определяющие режимы работы драйвера и сторожевого таймера, располагаются в следующем разделе системного реестра:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CronyxWatchdog\Parameters
```

Драйвер использует следующие параметры:

- **SystemStartupPeriod.milliseconds** – установка сторожевого таймера на заданный интервал при старте системы. При нулевом значении драйвер не будет запускать сторожевой таймер и будет бездействовать до установки интервала из программы пользователя. Значение по умолчанию: 10000 (10 секунд);

- **AutoAwake.boolean** – включает периодический автоматический перезапуск сторожевого таймера средствами драйвера. Значение по умолчанию = включено;
- **SystemPoweroffInterval.milliseconds** – интервал срабатывания при выключении системы. Сторожевой таймер запускается непосредственно перед тем, как ОС отключит питание (средствами ACPI или APM). При нулевом значении драйвер не будет запускать/перезапускать сторожевой таймер в этот момент. Значение по умолчанию: 1000 (1 секунда);
- **SystemRestartInterval.milliseconds** – интервал срабатывания при перезагрузки системы. Сторожевой таймер запускается после размонтирования файловых систем, но перед записью внутреннего каша дисков и их отключением. При нулевом значении драйвер не будет запускать/перезапускать сторожевой таймер в этот момент. Значение по умолчанию: 5000 (5 секунд);
- **SystemShutdownInterval.milliseconds** – интервал срабатывания при остановке системы. Сторожевой таймер запускается после остановки служб, но перед размонтированием файловых систем (перед очисткой файла подкачки). При нулевом значении драйвер не будет запускать/перезапускать сторожевой таймер в этот момент. Значение по умолчанию: 60000 (60 секунд);
- **DisableSounds.boolean** – запрещает драйверу подавать звуковые сигналы для сигнализации об ошибках и проблемах. Значение по умолчанию: выключено (разрешено подавать звуковые сигналы);
- **DontDisplayInUI.boolean** – скрывает сторожевой таймер из дерева отображения устройств в окне «Диспетчера устройств». Значение по умолчанию: выключено (не скрывать сторожевой таймер);
- **PreventSystemSleep.boolean** – запрещает системе переходить в режимы «Sleep», «Standby», «Hibernate» при активизированном сторожевом таймере. Значение по умолчанию: разрешено;
- **StartWithoutDevice.boolean** – допускает запуск драйвера при отключенном сторожевом таймере от LPT-порта. Драйвер сигнализирует об ошибке и будет каждые 5 секунд пытаться запустить сторожевой таймер. Значение по умолчанию: разрешено (допускать запуск без сторожевого таймера).



Если задано ненулевое значение **SystemStartupPeriod.milliseconds** и отключен автоматический перезапуск (**AutoAwake.boolean** равно нулю), то перезапуск таймера должен производиться программой пользователя.

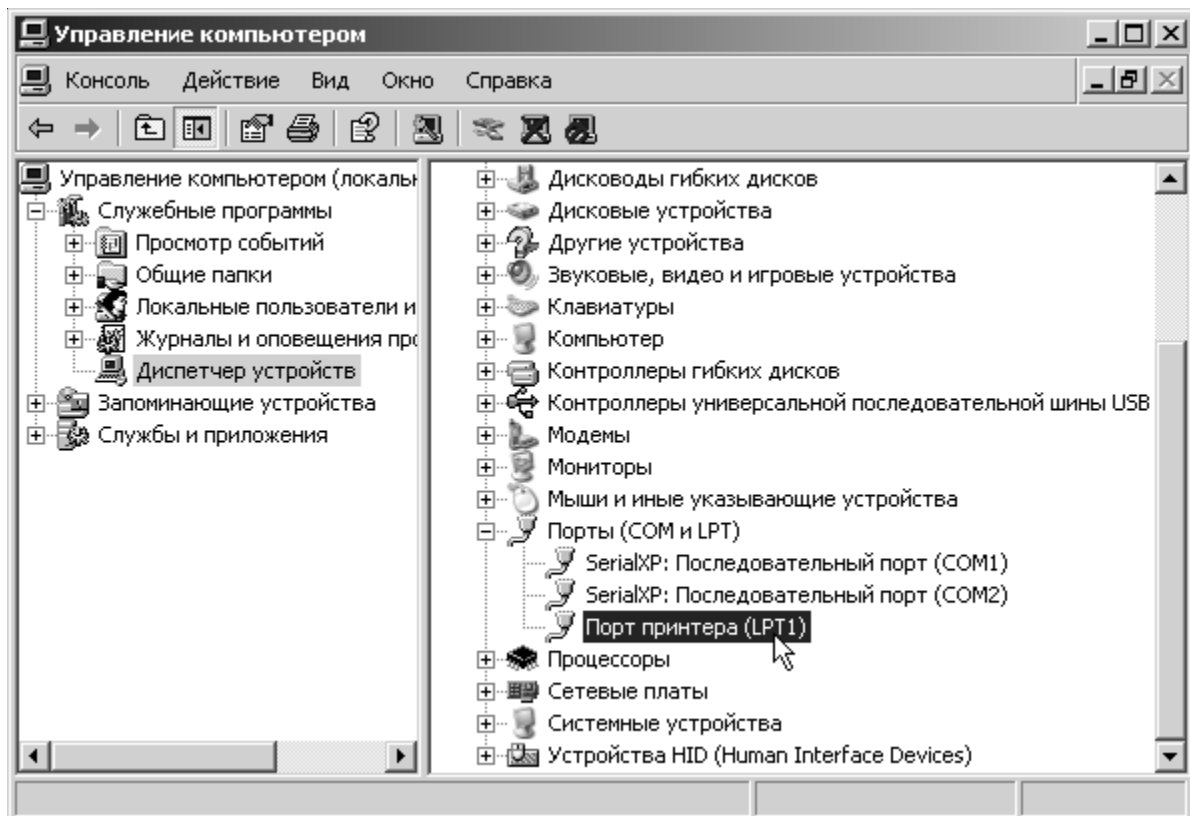
Если этого не сделать в заданный интервал времени с момента запуска драйвера, сторожевой таймер сработает и сбросит систему.



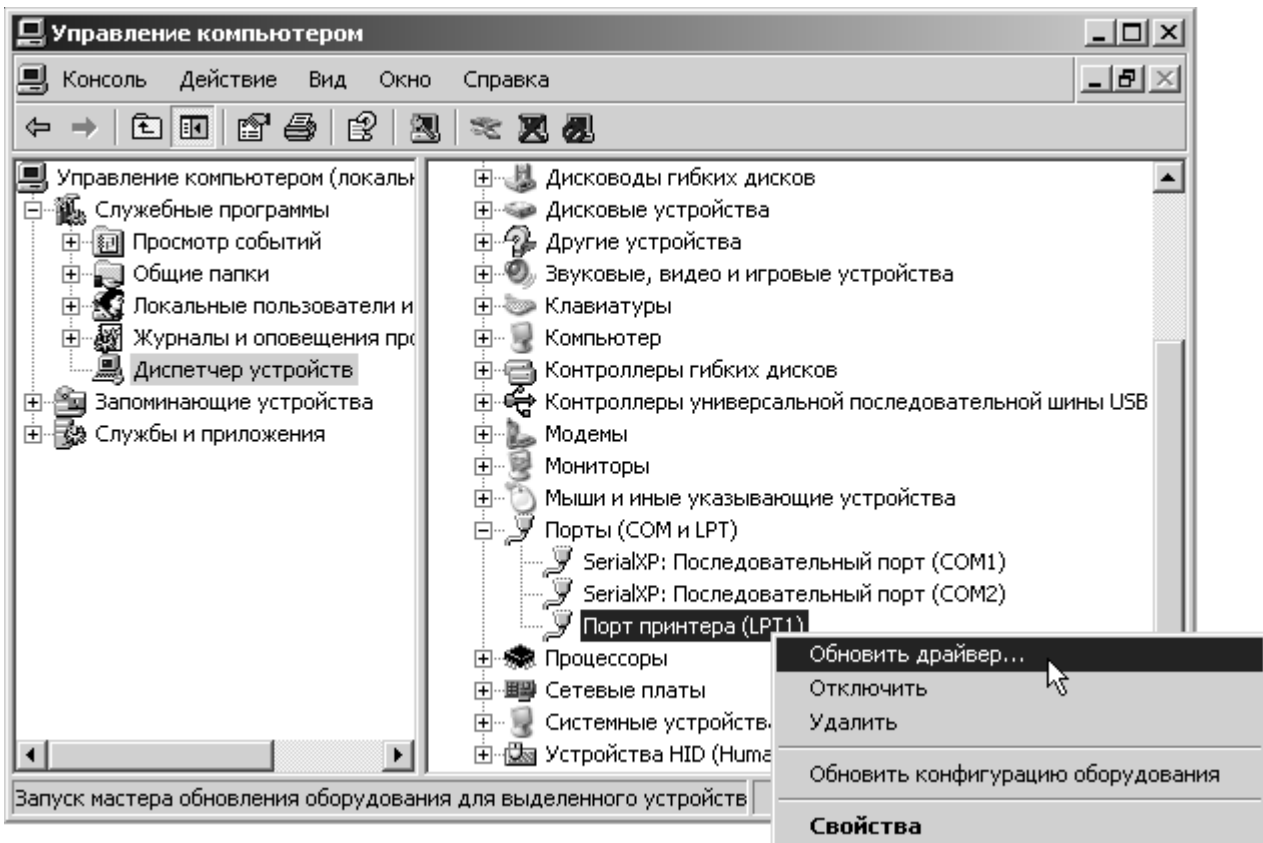
## Установка драйвера

Установка производится посредством обновления установленного драйвера для LPT-порта на драйвер Cronux Watchdog (деинсталляция – путем обновления драйвера Cronux Watchdog на стандартный системный драйвер). Ниже приведена пошаговая инструкция по установке драйвера.

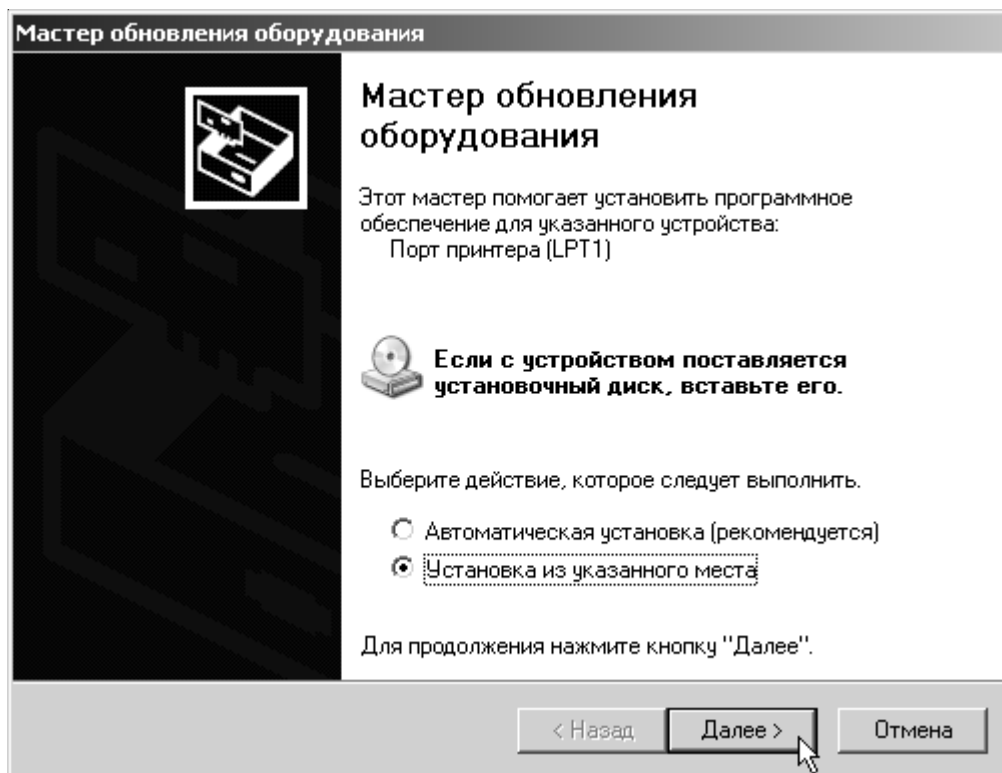
1. Запустите «Менеджер устройств» (Device Manager). Для этого запустите «Панель управления» (Control Panel), из нее запустите «Управление компьютером» (Computer Manager). В открывшейся консоли управления, в списке слева выберите «Управление устройствами» (Device Manager).



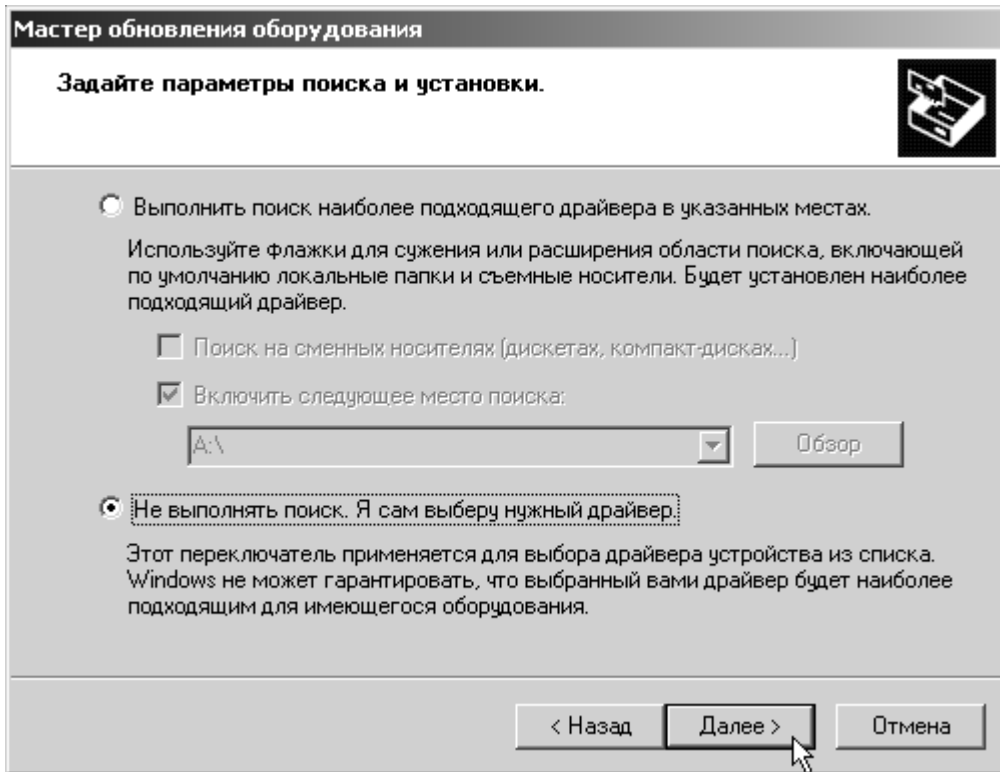
- В списке справа выберите LPT-порт и правой кнопкой мыши вызовите его свойства. Выберите функцию «Обновить драйвер» (Update Driver).



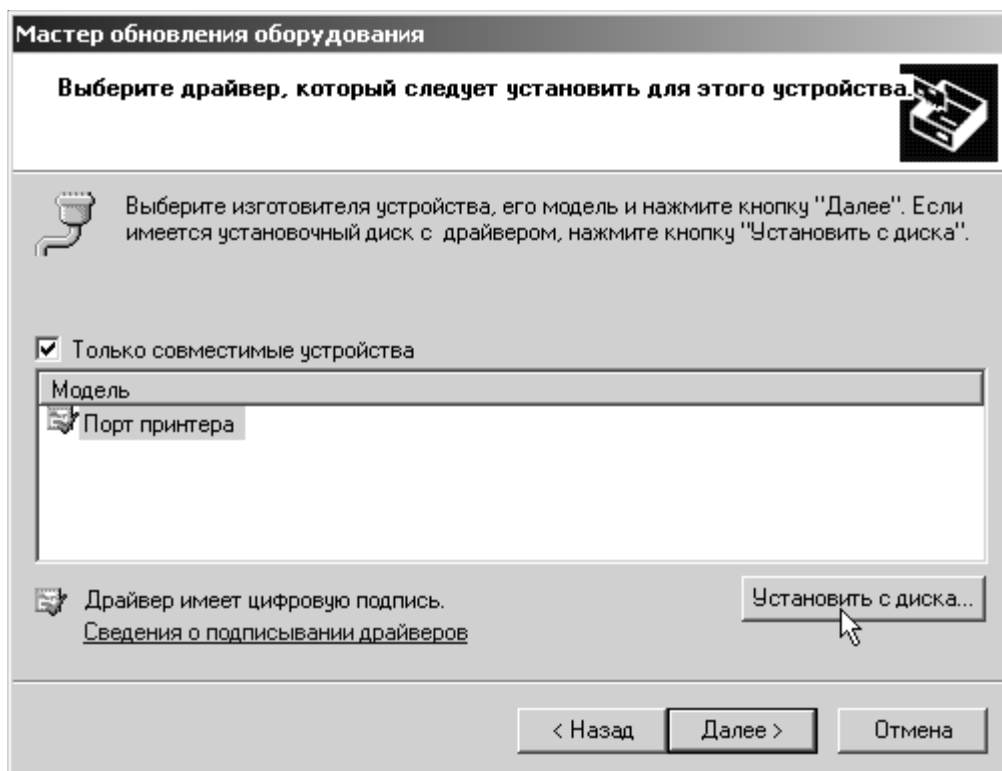
3. Запустится мастер установки и обновления драйверов, в появившемся окне выберите «Установить из списка или указанного источника» (Install from a list or specific location), нажмите кнопку «Далее» (Next).



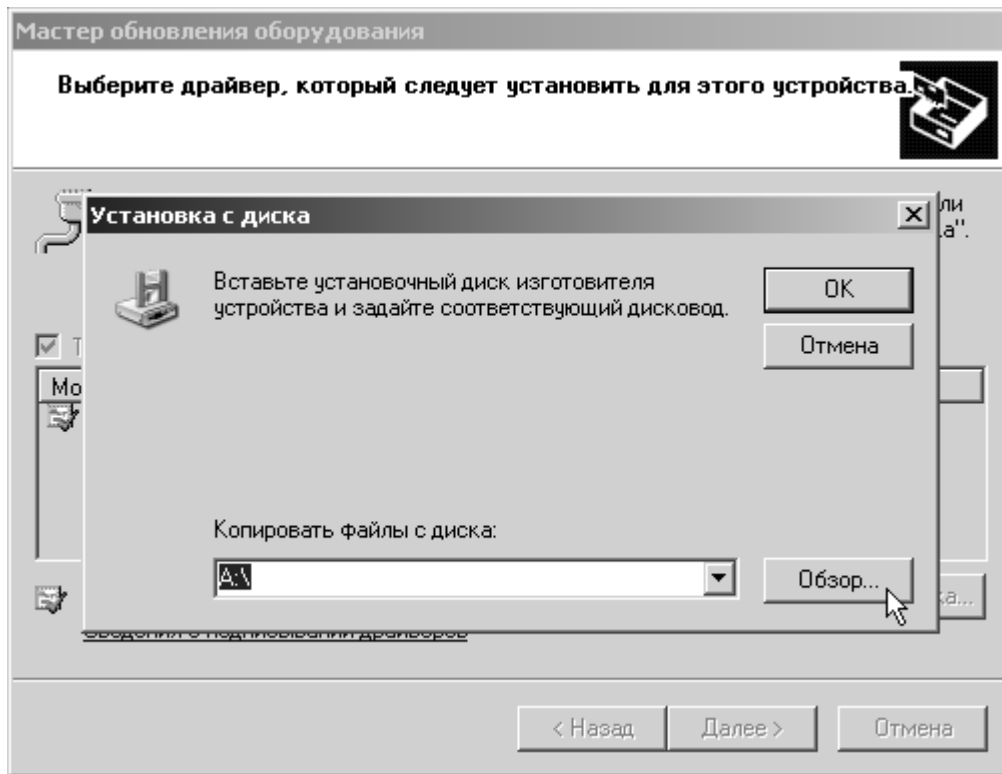
4. В появившемся окне выберите «Не производить поиск» (Don't search. I will choose driver to install), нажмите кнопку «Далее» (Next).



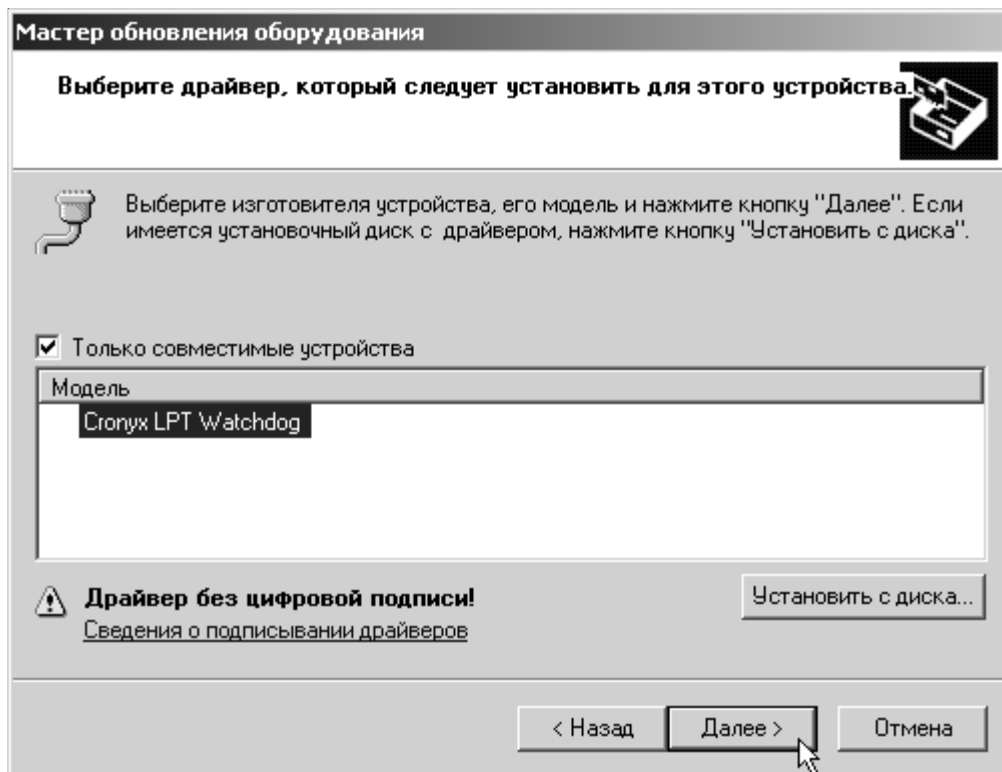
5. В появившемся окне нажмите кнопку «Установить с диска» (Have Disk).



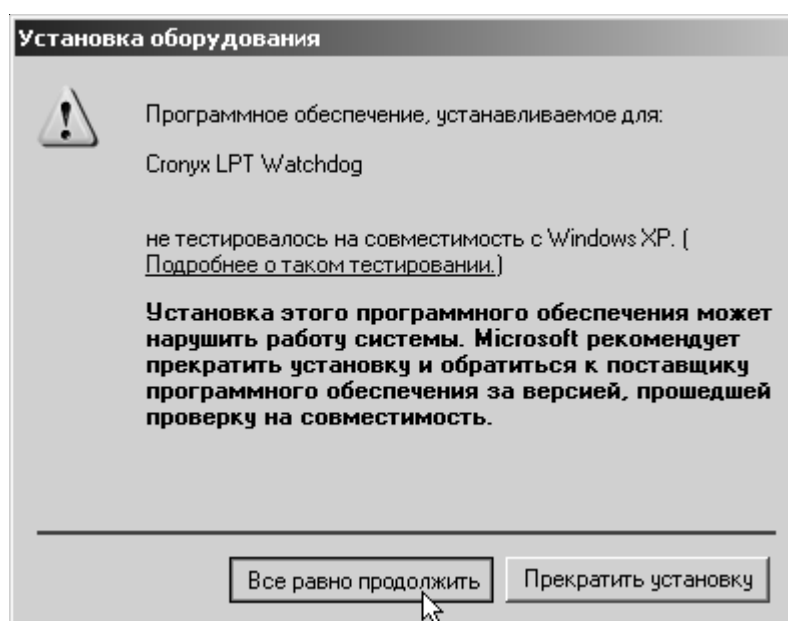
6. Укажите месторасположение извлеченных из архива файлов драйвера, нажмите кнопку «ОК».



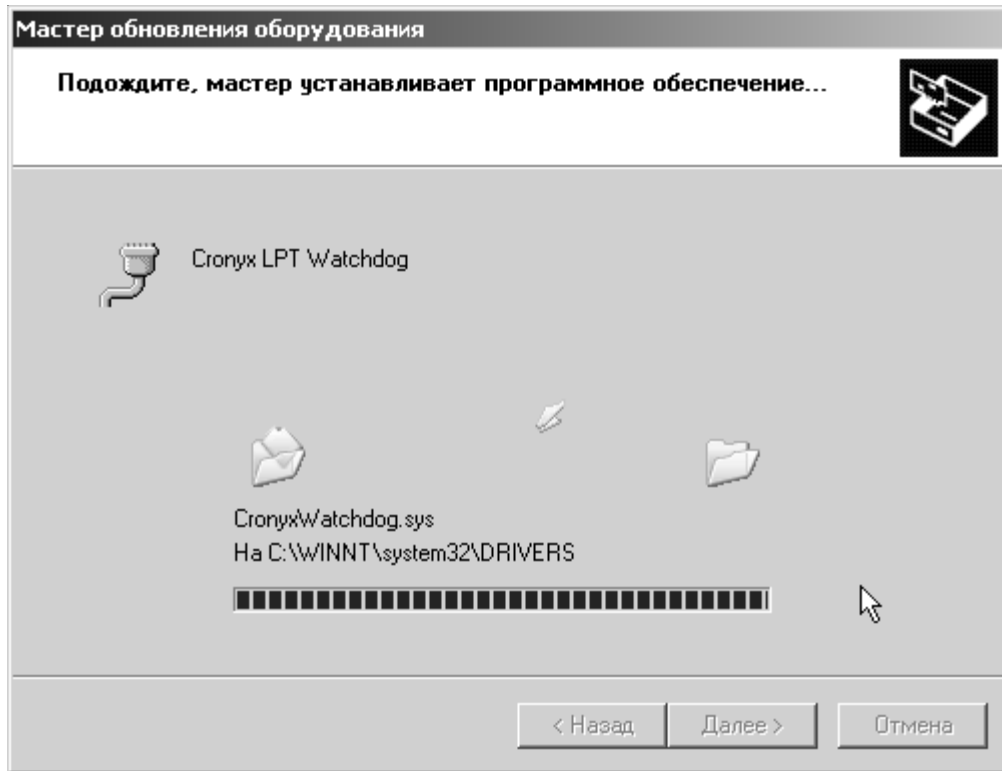
7. Выберите из списка необходимую версию драйвера, нажмите кнопку «ОК».



8. Операционная система может отобразить сообщение об отсутствии цифровой подписи, выберите кнопку «Продолжить установку» (Continue Anyway).

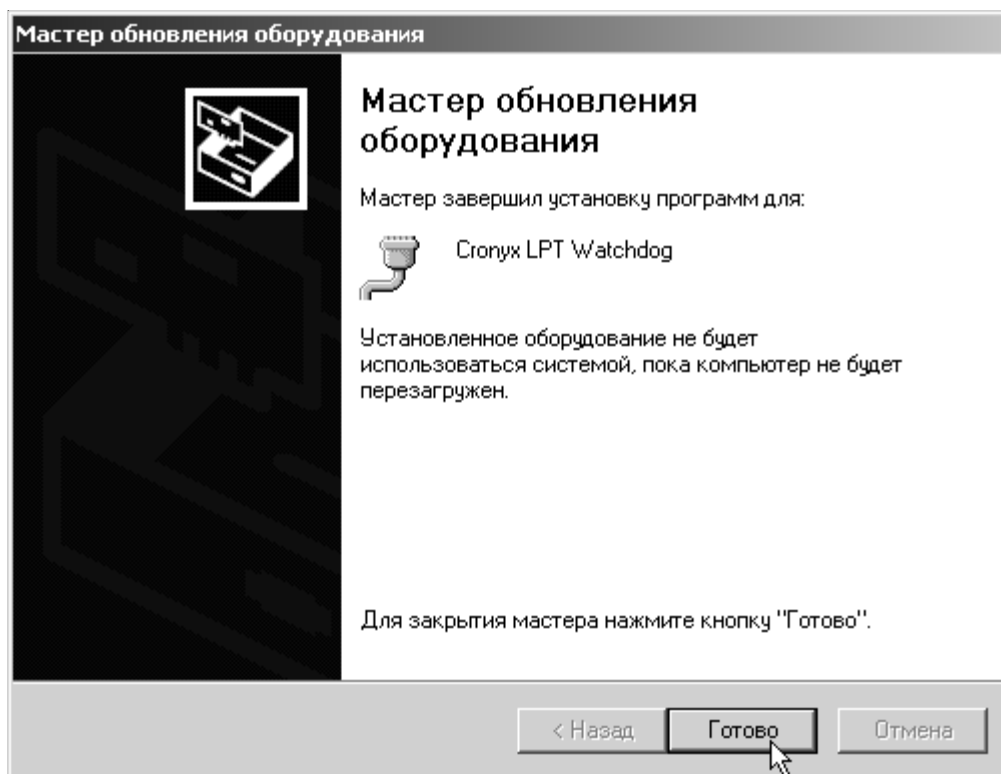


9. Система произведет копирование и установку драйвера.

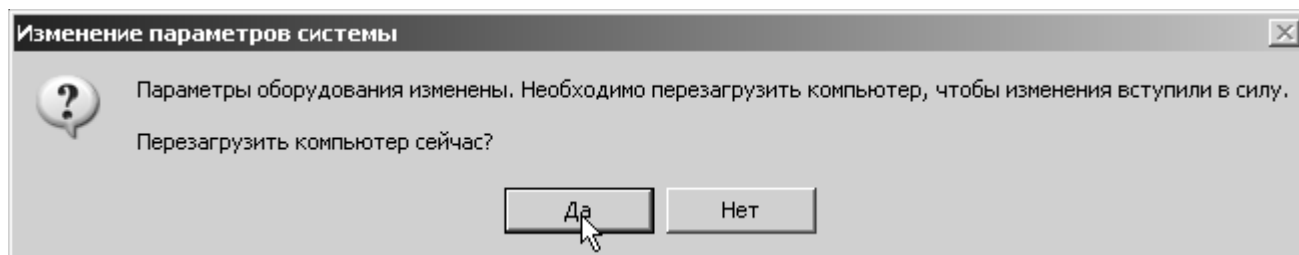




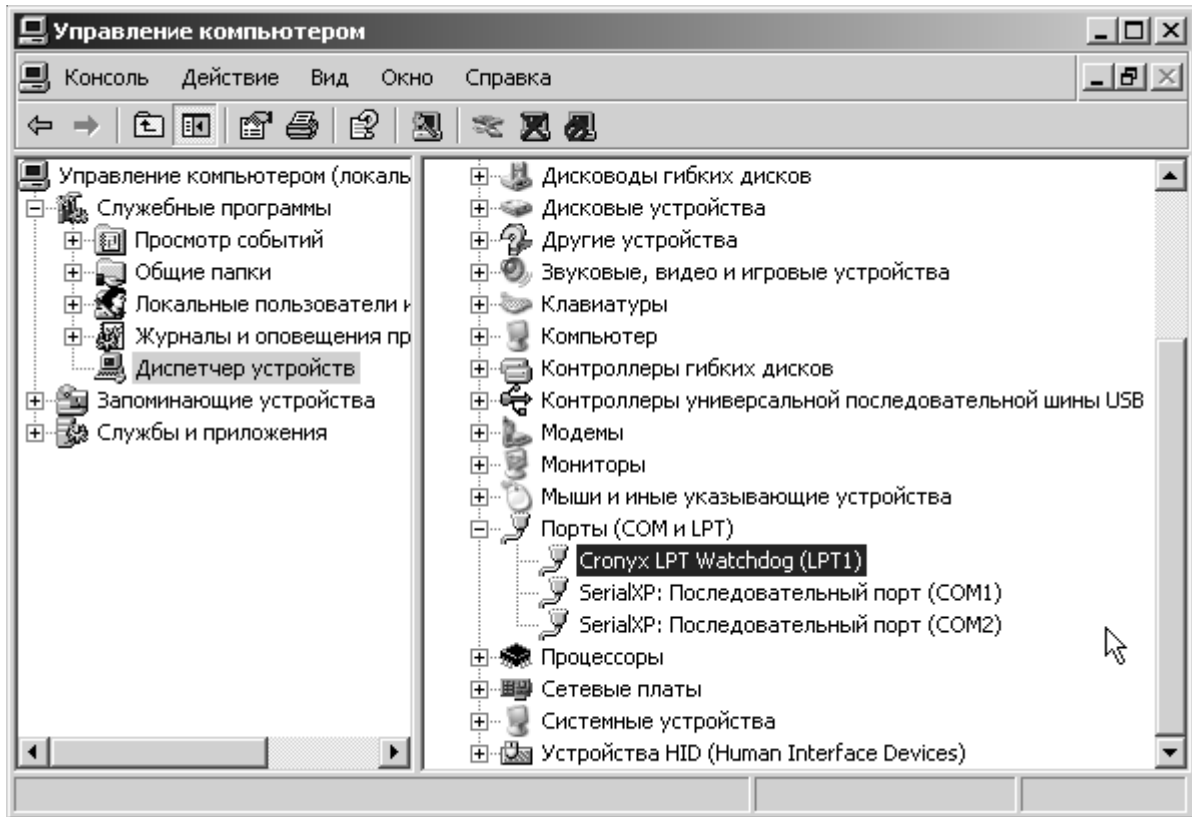
10. Затем система сообщит о результатах в финальном окне установки.



11. Для завершения установки может потребоваться перезагрузка системы.



12. После завершения установки можно заметить, что название устройства (LPT-порта) изменилось.



13. При возникновении ошибок стоит заглянуть в журнал системных событий (System Event Log). При затруднениях обратитесь в службу технической поддержки пользователей.

### Управление из программы пользователя

Программы пользовательского режима могут взаимодействовать с драйвером посредством `DeviceloControl()`. Набор IOCTL-команд позволяет устанавливать период срабатывания таймера, перезапускать сторожевой таймер, форсировать его срабатывание, включать и выключать режим автоматического периодического перезапуска таймера. Для получения спецификации и исходного кода обращайтесь в службу технической поддержки.

## Технические подробности работы драйвера

Обслуживание `SystemShutdownInterval.milliseconds` производится по получению `IRP_MJ_SHUTDOWN`, при регистрации посредством `IoRegisterShutdownNotification()`.

Обслуживание `SystemRestartInterval.milliseconds` и `SystemPoweroffInterval.milliseconds` производится в ветви «Device-Power-Down» кода управления питанием.

Периодический перезапуск сторожевого таймера производится посредством отложенной обработки через `IoQueueWorkItem()`. Это сделано намеренно с целью обеспечить срабатывания сторожевого таймера при включенном «автоматическом перезапуске» в случае «заморозки» обслуживания системных очередей отложенных заданий.

## 2.4. Утилита тестирования

Утилита тестирования `wdog.exe` находится на дискете в каталоге `msdos/`. Она может функционировать под операционными системами DOS/Win3x/Win9x, при условии отсутствия драйвера принтера. При вызове утилита производит поиск сторожевого таймера, после чего производит периодические обращения к нему. При нажатии любой клавиши утилита заканчивает работу, что приводит к аппаратному сбросу.

Поставляемые тексты утилиты `wdog.exe` могут также служить примером программного интерфейса к сторожевому таймеру. Для трансляции использовался компилятор Turbo C 2.0.

## Раздел 3. Программный интерфейс

Управление сторожевым таймером производится через регистры параллельного порта компьютера. Адрес базового регистра определяется конструкцией и настройками материнской платы:

378h	— LPT1
278h	— LPT2
3BCh	— LPT3

Смещения регистров относительно базового адреса указаны в таблице:

Регистр	Доступ	Смещение	Описание
DATA	зп8	0	Регистр данных
STS	чт8	1	Регистр состояния

### 3.1. Регистр DATA

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

Бит 2 — /DOUT — последовательные данные для выдачи в сторожевой таймер.

Бит 1 — /CLK — тактовый импульс для передачи очередного бита данных.

Бит 0 — /CS — сигнал цикла обмена.

### 3.2. Регистр STS

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

Бит 7 — /IN — признак готовности сторожевого таймера к пересылке очередного бита информации.

Передача команд сторожевому таймеру выполняется последовательным кодом, через биты /CS, /CLK, /DOUT регистра DATA. Бит /DIN регистра STS используется для контроля правильности передачи данных.

Команды сторожевого таймера имеют разрядность 8 бит и передаются, начиная со старшего бита. Коды команд перечислены в таблице:

<b>Код</b>	<b>Команда</b>
C8h	Установка таймера на 100 миллисекунд
C9h	Установка таймера на 1 секунду
CAh	Установка таймера на 5 секунд
CBh	Установка таймера на 10 секунд
CCh	Установка таймера на 20 секунд
CDh	Установка таймера на 40 секунд
CEh	Установка таймера на 60 секунд
CFh	Сброс таймера (продление периода)

В холостом состоянии биты /CS и /CLK должны быть установлены в 1. Цикл обмена начинается со сброса бита /CS. Передача бита команды выполняется в четыре этапа:

- 1) установить бит /DOUT по значению передаваемого бита команды;
- 2) сбросить бит /CLK;
- 3) установить бит /CLK;
- 4) проверить, что бит /DIN равен инверсии передаваемого бита (см. пример ниже).

Для передачи всей команды шаги 1-4 необходимо повторить восемь раз. По окончании обмена нужно установить бит /CS.

### 3.3. Пример запуска таймера

Ниже приведен пример запуска таймера с периодом 100 миллисекунд (команда C8h).

```

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

outb (DATA, 7);           /* инициализация*/
outb (DATA, 2);           /* устанавливаем CS */
/* бит 7 = 1 */
outb (DATA, 2);           /* устанавливаем данные */
outb (DATA, 0);           /* устанавливаем CLK */
outb (DATA, 2);           /* сбрасываем CLK, данные */
if (!(inb (STS) & 0x80)) error ();
/* бит 6 = 1 */
outb (DATA, 2);           /* устанавливаем данные */
outb (DATA, 0);           /* устанавливаем CLK */
outb (DATA, 2);           /* сбрасываем CLK, данные */
if (!(inb (STS) & 0x80)) error ();
/* бит 5 = 0 */
outb (DATA, 6);           /* устанавливаем данные */
outb (DATA, 4);           /* устанавливаем CLK */
outb (DATA, 6);           /* сбрасываем CLK, данные */
if (inb (STS) & 0x80) error ();
/* бит 4 = 0 */
outb (DATA, 6);           /* устанавливаем данные */
outb (DATA, 4);           /* устанавливаем CLK */
outb (DATA, 6);           /* сбрасываем CLK, данные */
if (inb (STS) & 0x80) error ();
/* бит 3 = 1 */
outb (DATA, 2);           /* устанавливаем данные */
outb (DATA, 0);           /* устанавливаем CLK */
outb (DATA, 2);           /* сбрасываем CLK, данные */
if (!(inb (STS) & 0x80)) error ();
/* бит 2 = 0 */
outb (DATA, 6);           /* устанавливаем данные */
outb (DATA, 4);           /* устанавливаем CLK */
outb (DATA, 6);           /* сбрасываем CLK, данные */
if (inb (STS) & 0x80) error ();
/* бит 1 = 0 */
outb (DATA, 6);           /* устанавливаем данные */
outb (DATA, 4);           /* устанавливаем CLK */
outb (DATA, 6);           /* сбрасываем CLK, данные */
if (inb (STS) & 0x80) error ();
/* бит 0 = 0 */
outb (DATA, 6);           /* устанавливаем данные */
outb (DATA, 4);           /* устанавливаем CLK */
outb (DATA, 6);           /* сбрасываем CLK, данные */
if (inb (STS) & 0x80) error ();

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



