

# **CAN - bus - MicroPC интерфейс, версия 1.0**

**Руководство пользователя**



## **Замечания о праве на копирование**

Все права на программное обеспечение, аппаратное обеспечение и данное руководство принадлежат фирме Марафон и защищены законодательством Российской Федерации.

Копирование этого руководства возможно только при получении письменного разрешения у фирмы Марафон.

Москва, 117330, Мосфильмовская ул., 17б

Тел.: +7 095 9395659, +7 095 9391324

Факс: +7 095 9395659

<http://www.marathon.ru>



## Содержание

<b>1. Основные характеристики.....</b>	<b>7</b>
<b>2. Установка и тестирование карты.....</b>	<b>9</b>
2.1. Настройка карты .....	9
2.2. Установка карты.....	11
<b>3. Образ карты в памяти .....</b>	<b>14</b>
<b>4. Программирование CAN-контроллера.....</b>	<b>15</b>
4.1. Инициализация CAN-контроллера .....	15
4.2. Чтение регистров CAN-контроллера .....	15
4.3. Запись в регистры CAN-контроллера.....	15
4.4. Аппаратный сброс CAN-контроллера .....	16
<b>5. CAN-bus интерфейс .....</b>	<b>17</b>
<b>6. Приложения .....</b>	<b>19</b>
6.1. Заводские установки.....	19
6.2. Полная таблица соответствия базового адреса положению переключателя SW1 .....	20



## 1. Основные характеристики

- плата выполнена в формате MicroPC
- возможна установка в 8-ми разрядный слот ISA
- два CAN-контроллера Philips SJA1000
- частота внешней синхронизации CAN-контроллеров 16 MHz
- быстрый и эффективный доступ к CAN-контроллерам благодаря отображению внутренних регистров CAN-контроллеров в область памяти центрального процессора
- возможен выбор одной из пяти линий прерываний  
IRQ: 3, 4, 5, 7, 9
- базовый адрес выбирается пользователем с помощью DIP переключателя, начиная с 0xC000
- двухканальный CAN-bus интерфейс с гальванической развязкой. Возможна установка защиты от импульсных помех по специальному заказу.
- диапазон рабочих температур от минус 40 до +85 °C
- напряжение питания 5В ±5%
- потребляемый ток не более 500 мА, средний 300 мА
- программное обеспечение:
  - программа MONITOR для MS-DOS, Windows 9x
  - драйвер и программа NT\_MONITOR для Windows NT 4.0
  - драйвер для ОС Linux. Поставляется по специальному заказу

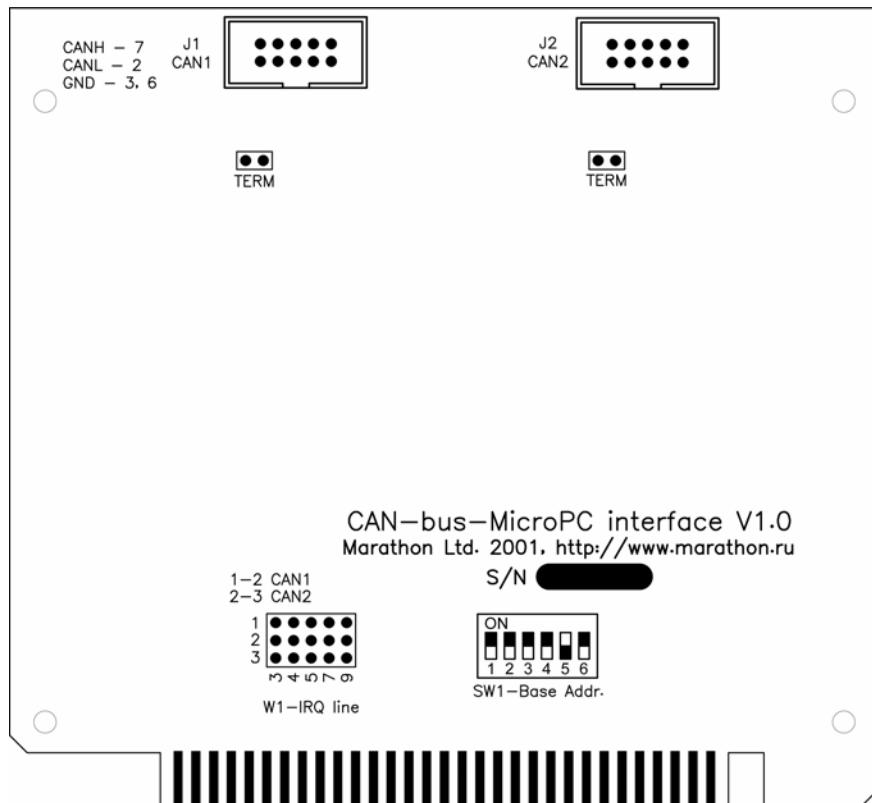




## 2. Установка и тестирование карты

### 2.1. Настройка карты

До того, как Вы установите интерфейсную карту в компьютер, нужно произвести настройку базового адреса и номеров IRQ. DIP переключатели и перемычки, настраиваемые пользователем, показаны на следующем рисунке:



#### Настройка базового адреса

Настройка базового адреса карты производится с помощью DIP переключателя SW1. Приведенная ниже таблица показывает

соответствие переключателей наиболее часто используемым адресам.

Базовый адрес	SW1 1	SW1 2	SW1 3	SW1 4	SW1 5	SW1 6
C800h	ON	ON	ON	OFF	ON	ON
D000h	ON	ON	ON	ON	OFF	ON
D800h	ON	ON	ON	OFF	OFF	ON
E000h	ON	ON	ON	ON	ON	OFF
E800h	ON	ON	ON	OFF	ON	OFF

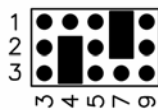
Интерфейсная карта использует следующие за базовым адресом 4 КБайт памяти. CAN-контроллеры доступны в этой области памяти как обыкновенная RAM.

Удостоверьтесь в том, что никакое другое устройство не использует область памяти, предназначенную для интерфейсной карты.

### Настройка IRQ

Две линии прерывания IRQ должны быть назначены карте для обеспечения эффективной работы программного обеспечения. Выбор IRQ производится с помощью замыкания соответствующих перемычек поля W1. Надпись рядом с каждой перемычкой обозначает номер IRQ.

1–2 CAN1  
2–3 CAN2



W1-IRQ line

На рисунке приведен пример настройки IRQ: первому каналу назначено IRQ7, второму каналу назначено прерывание IRQ4. Если по какой-либо причине Вы не используете прерывания вообще, то оставьте все перемычки открытыми.

Удостоверьтесь, что выбранные прерывания не используется ни одним устройством в Вашем компьютере.

### Терминатор CAN линии

Для каждого канала с помощью перемычки TERM можно установить 120Ω терминатор. Шина CAN должна иметь терминаторы на обоих концах.

## 2.2. Установка карты

1. После установки карты и загрузки компьютера Вы должны выделить для интерфейсной карты по крайней мере 4 Кбайт адресного пространства. Следующие примеры демонстрируют эту процедуру для карты с базовым адресом 0xD000.
  - **MS-DOS:** Если на Вашем компьютере используется менеджер памяти (например EMM386.EXE), то необходимо модифицировать команду его запуска так, чтобы он не работал с областью адресов, в которых находится интерфейсная карта. Например, для EMM386.EXE команда запуска из CONFIG.SYS будет выглядеть так:  
DEVICE = EMM386.EXE [Ваши опции] X=D000h-D0FFh.
  - **MS Windows 3.1.x :** Если менеджер памяти не загружается из CONFIG.SYS, то необходимо изменить файл SYSTEM.INI в каталоге Windows. В секции [386ENH] необходимо добавить строку: EMMExclude = D000h-D0FFh. В этой секции может находиться несколько директив EMMEXclude. Пожалуйста, обратите внимание на то, что менеджер памяти Windows распознает как начало диапазона адресов только те адреса, которые кратны 16 КБайт.
  - **Windows 9x:** В Windows 9x можно выделить необходимые ресурсы следующим образом:
    - нажмите кнопку «Пуск» на панели задач
    - выберите «Настройка»
    - выберите «Панель Управления»
    - дважды щелкните по иконке «Система»
    - выберите закладку «Устройства»
    - сделайте строку «Компьютер» активной

- нажмите кнопку «Свойства» в нижней части диалога
- в появившемся диалоге выберите закладку «Резервирование ресурсов»
- активируйте радио-кнопку «Запросы прерывания (IRQ)»
- нажмите кнопку «Добавить»
- введите номер линии прерывания для первого канала, которая соответствует установленному на интерфейсной карте
- нажмите кнопку «ОК»
- зарезервируйте аналогичным образом прерывание для второго канала
- активируйте радио-кнопку «Память»
- нажмите кнопку «Добавить»
- введите базовый адрес карты в поле «Начальное значение» (например, в нашем случае D0000). Обратите внимание на то, что это должен быть линейный адрес, а не сегмент, как было в случае для DOS.
- введите конечный адрес выделяемой области в поле «Конечное значение » (в нашем случае D0FFF)
- закройте диалог кнопкой «ОК»
- закройте окно «Свойства: Компьютер» кнопкой «ОК»
- Windows автоматически перезагрузит Ваш компьютер

Сделанные изменения теперь вступят в силу, и можно будет приступить к запуску тестовых программ.

- **Windows NT 4.0:** Для работы с картой необходимо установить драйвер SJA1000.sys

Войдите в систему с правами администратора. Запустите программу конфигурации сервиса Sja1000 "SjaConf.exe". Программа проверит регистрацию драйвера в системе. Если

драйвер не установлен, то будет выдан запрос на установку. При положительном ответе производится копирование файла "Sja1000.sys" в каталог \winnt\system32\drivers (при установке может потребоваться указать путь к файлу "Sja1000.sys") и вносятся изменения в системный реестр. Ресурсы, используемые адаптерами (на базе контроллера Sja1000) могут быть определены автоматически или установлены вручную. При установке флажка "Автоопределение ресурсов" необходимо задать начальный адрес памяти (по умолчанию 0xC0000), размер области поиска (0x30000), шаг изменения базового адреса (0x1000) и перечислить возможные линии прерываний. При ручной установке необходимо задать количество устройств и назначить ресурсы каждому устройству отдельно (правильность вводимых значений не проверяется). Все изменения вступают в силу после перезагрузки системы. Если был выбран режим "Автоопределение ресурсов" количество обнаруженных устройств и ресурсы, занимаемые ими, будут отображаться при следующем запуске программы "SjaConf.exe".

Вместе с интерфейсной картой поставляется также программа MONITOR, работающая в операционных системах DOS, Windows 3.1.x., Windows 9x и программа NT\_MONITOR, работающая в операционной системе Windows NT 4.0. Эти программы можно использовать для мониторинга CAN сети: отправки и приема сообщений.

### 3. Образ карты в памяти

Следующая таблица показывает смещения внутри области памяти, занимаемой картой, для доступа к CAN-контроллерам и областям аппаратного сброса.

Base: 0000h – Base: 00FFh	1 CAN-контроллер
Base: 0100h – Base: 01FFh	Область аппаратного сброса 1 CAN-контроллера
Base: 0200h – Base: 02FFh	2 CAN-контроллер
Base: 0300h – Base: 03FFh	Область аппаратного сброса 2 CAN-контроллера

Регистры CAN-контроллеров отображаются (memory mapping) в RAM персонального компьютера. Такое конструктивное решение позволяет прикладным программам быстро и эффективно обмениваться данными с CAN-контроллерами. Чтение и запись производится с помощью обычных инструкций read и write процессора.

Аппаратный сброс (hardware reset) CAN-контроллеров осуществляется при записи каких-либо данных в область аппаратного сброса (см. таблицу выше).

## 4. Программирование CAN-контроллера

Следующие примеры демонстрируют способы работы с CAN-контроллером SJA1000. Примеры написаны с использованием Borland C/C++ v.4.5 и должны правильно компилироваться в среде DOS всеми C компиляторами, совместимыми со стандартом ANSI C.

### 4.1. Инициализация CAN-контроллера

Для правильной работы карты необходимо провести инициализацию CAN-контроллера SJA1000 - записать значение 0x5E в регистр Output Control.

### 4.2. Чтение регистров CAN-контроллера

```
#define unsigned char BYTE
/*
// Описание: считывает один байт из CAN-контроллера
// Параметры:
// mem_segment - базовый адрес карты (сегмент в терминах DOS)
// reg_offset - смещение читаемого регистра от базового адреса
//
// Возвращаемое значение: считанный байт
*/

BYTE can_hw_read ( int mem_segment, BYTE reg_offset )
{
    BYTE huge *can_base_ptr;
    can_base_ptr = ( mem_segment * 0x10000L );
    return ( *( can_base_ptr + reg_offset ) );
}

void main(void)
{
    BYTE reg_addr;          /* определяет регистр */
    unsigned int segment; /* определяет базовый адрес */
    BYTE value;            /* будет содержать значение регистра */
    reg_addr = 0x00;
    segment = 0xD000;
    value = can_hw_read ( segment, reg_addr );
}
```

### 4.3. Запись в регистры CAN-контроллера

```
#define unsigned char BYTE
/*
// Описание: записывает один байт в CAN-контроллер
// Параметры:
// mem_segment - базовый адрес карты (сегмент в терминах DOS)
```

```

// reg_offset - смещение читаемого регистра от базового адреса
// value - значение, которое будет записано
//
// Возвращаемое значение: нет
*/

BYTE can_hw_write ( int mem_segment, BYTE reg_offset, BYTE value )
{
    BYTE huge *can_base_ptr;
    can_base_ptr = ( mem_segment * 0x10000L );
    ( *( can_base_ptr + reg_offset ) ) = value;
}

void main(void)
{
    BYTE reg_addr;          /* определяет регистр */
    unsigned int segment;  /* определяет базовый адрес */
    BYTE value;           /* содержит значение регистра */
    reg_addr = 0x00;
    segment = 0xD000;
    value = 1;
    can_hw_write ( segment, reg_addr, value );
}

```

#### 4.4. Аппаратный сброс CAN-контроллера

```

void main(void)
{
    BYTE huge *can_base_ptr;
    BYTE foo = 1; /*значение не важно*/
    unsigned int segment = 0xD000;

    can_base_ptr = (segment * 0x10000L);
    * ( can_base_ptr + 0x100 ) = foo;
}

```



## 5. CAN-bus интерфейс

CAN-bus интерфейс выведен на разъемы IDC10.

Контакты разъемов используются следующим образом:

Номер контакта IDC10	Сигнал
1	Не используется
2	CAN low
3	GND
4	Не используется
5	Не используется
6	GND
7	CAN high
8	Не используется
9	Не используется
10	Не используется



## 6. Приложения

### 6.1. Заводские установки

Интерфейсная карта поставляется со следующими установками:

1. базовый адрес: D000h

DIP SW1-1 ON  
DIP SW1-2 ON  
DIP SW1-3 ON  
DIP SW1-4 ON  
DIP SW1-5 OFF  
DIP SW1-6 ON

2. номера IRQ: CAN1 - IRQ 5, CAN2 - IRQ7

**6.2. Полная таблица соответствия базового адреса положению переключателя SW1**

Базовый адрес	SW1 1	SW1 2	SW1 3	SW1 4	SW1 5	SW1 6
C000h	ON	ON	ON	ON	ON	ON
C100h	OFF	ON	ON	ON	ON	ON
C200h	ON	OFF	ON	ON	ON	ON
C300h	OFF	OFF	ON	ON	ON	ON
C400h	ON	ON	OFF	ON	ON	ON
C500h	OFF	ON	OFF	ON	ON	ON
C600h	ON	OFF	OFF	ON	ON	ON
C700h	OFF	OFF	OFF	ON	ON	ON
C800h	ON	ON	ON	OFF	ON	ON
C900h	OFF	ON	ON	OFF	ON	ON
CA00h	ON	OFF	ON	OFF	ON	ON
CB00h	OFF	OFF	ON	OFF	ON	ON
CC00h	ON	ON	OFF	OFF	ON	ON
CD00h	OFF	ON	OFF	OFF	ON	ON
CE00h	ON	OFF	OFF	OFF	ON	ON
CF00h	OFF	OFF	OFF	OFF	ON	ON
D000h	ON	ON	ON	ON	OFF	ON
D100h	OFF	ON	ON	ON	OFF	ON
D200h	ON	OFF	ON	ON	OFF	ON
D300h	OFF	OFF	ON	ON	OFF	ON
D400h	ON	ON	OFF	ON	OFF	ON
D500h	OFF	ON	OFF	ON	OFF	ON
D600h	ON	OFF	OFF	ON	OFF	ON
D700h	OFF	OFF	OFF	ON	OFF	ON
D800h	ON	ON	ON	OFF	OFF	ON
D900h	OFF	ON	ON	OFF	OFF	ON
DA00h	ON	OFF	ON	OFF	OFF	ON
DB00h	OFF	OFF	ON	OFF	OFF	ON
DC00h	ON	ON	OFF	OFF	OFF	ON
DD00h	OFF	ON	OFF	OFF	OFF	ON
DE00h	ON	OFF	OFF	OFF	OFF	ON
DF00h	OFF	OFF	OFF	OFF	OFF	ON

E000h	ON	ON	ON	ON	ON	OFF
E100h	OFF	ON	ON	ON	ON	OFF
E200h	ON	OFF	ON	ON	ON	OFF
E300h	OFF	OFF	ON	ON	ON	OFF
E400h	ON	ON	OFF	ON	ON	OFF
E500h	OFF	ON	OFF	ON	ON	OFF
E600h	ON	OFF	OFF	ON	ON	OFF
E700h	OFF	OFF	OFF	ON	ON	OFF
E800h	ON	ON	ON	OFF	ON	OFF
E900h	OFF	ON	ON	OFF	ON	OFF
EA00h	ON	OFF	ON	OFF	ON	OFF
EB00h	OFF	OFF	ON	OFF	ON	OFF
EC00h	ON	ON	OFF	OFF	ON	OFF
ED00h	OFF	ON	OFF	OFF	ON	OFF
EE00h	ON	OFF	OFF	OFF	ON	OFF
EF00h	OFF	OFF	OFF	OFF	ON	OFF
F000h	ON	ON	ON	ON	OFF	OFF
F100h	OFF	ON	ON	ON	OFF	OFF
F200h	ON	OFF	ON	ON	OFF	OFF
F300h	OFF	OFF	ON	ON	OFF	OFF
F400h	ON	ON	OFF	ON	OFF	OFF
F500h	OFF	ON	OFF	ON	OFF	OFF
F600h	ON	OFF	OFF	ON	OFF	OFF
F700h	OFF	OFF	OFF	ON	OFF	OFF
F800h	ON	ON	ON	OFF	OFF	OFF
F900h	OFF	ON	ON	OFF	OFF	OFF
FA00h	ON	OFF	ON	OFF	OFF	OFF