

# **R16x Flash Programmer 1.3**

**Руководство пользователя.**

**Версия 1.0**

## Содержание

Введение .....	3
Описание пользовательского интерфейса .....	4
Программирование «шаг за шагом» .....	6
Запуск из командной строки .....	11

## Введение

R16x Flash Programmer версии 1.3 представляет собой программу для операционных систем MS Windows 95/98/NT4, предназначенную для программирования внешней flash-памяти серий 29F010 и 29F040, установленных на контроллерах M164, M167-1, M167-2, 167-104, MF167, E167-3UM, G167F-3U1.

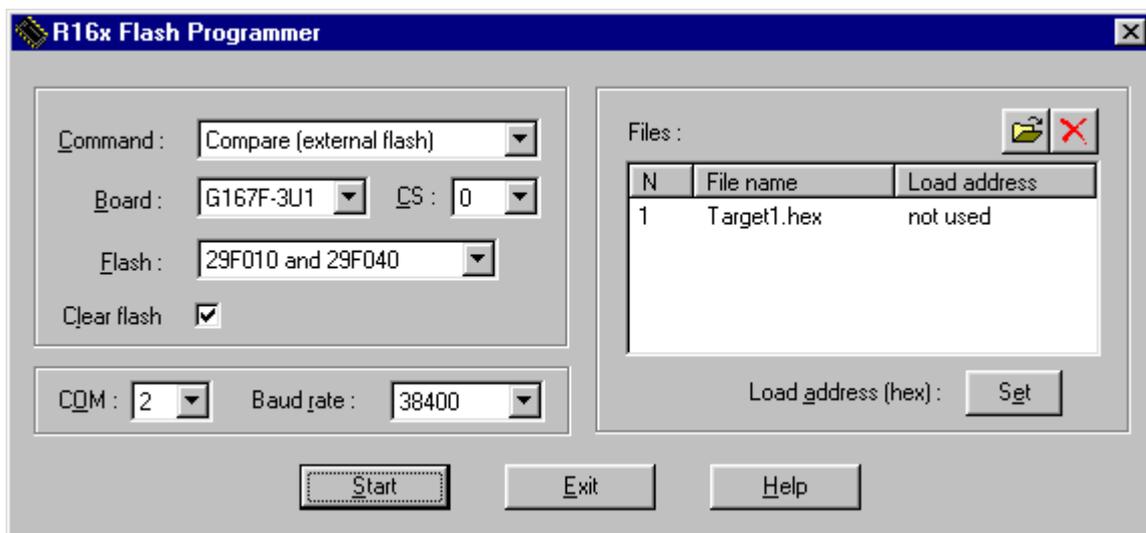


Рис.1 Главное окно программатора R16x Flash Programmer

Программатор R16x Flash Programmer позволяет:

- Программировать flash-память данными из указанного списка файлов в двоичном или шестнадцатеричном Intel Hex 86 форматах;
- Сравнивать данные во flash-памяти с данными файлов, заданных в списке;
- Задавать все параметры из командной строки;
- Автоматически запускать указанную команду на выполнение.

## Описание пользовательского интерфейса

Пользовательский интерфейс программатора R16x Flash Programmer состоит из основного окна, представленного на рис.1., и включает в себя следующие элементы:

Опция	Описание
<b>Command</b>	Список выбора команды программатора для выполнения. Содержит команды программирования внешней flash-памяти и сравнения
<b>Board</b>	Список выбора подключенной платы контроллера
<b>CS</b>	Список выбора используемого сигнала CS (chip select) при программировании
<b>Flash</b>	Список выбора типа flash-памяти, установленной на плате выбранного контроллера
<b>Clear flash</b>	Флажок для указания стирания flash-памяти перед программированием
<b>COM</b>	Список выбора номера последовательного порта PC для работы программатора
<b>Baud rate</b>	Список выбора скорости передачи данных для последовательного порта PC

Правильный выбор представленных параметров программатора является очень важным и влияет на правильность работы программатора в целом.

В программаторе R16x Flash Programmer имеется возможность указания набора файлов для программирования по указанным адресам. Ниже представлен фрагмент окна программатора, элементами которого можно управлять списком файлов:

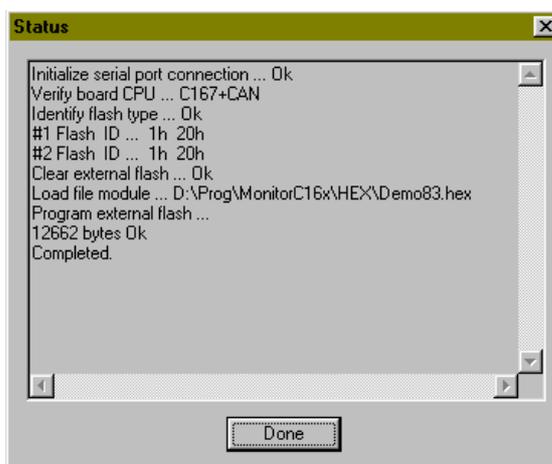
N	File name	Load address
1	Demo83.hex	not used
3	M1671dm.hex	not used
3	Demo.pgm	12 0FDC

Список файлов состоит из имени файла (полный путь к файлу не отображается) и его адреса загрузки (начального адреса программирования). Для добавления файла к списку необходимо нажать на кнопку , при этом появится диалог для выбора файла. При выборе файла по его расширению определяется его тип – двоичный или шестнадцатеричный (см. Раздел «Используемые форматы файлов»). Для удаления файла из списка необходимо его сначала выделить, а затем нажать на кнопку .

Адрес загрузки имеют только двоичные файлы. При добавлении нового файла в список адрес загрузки принимает значение по умолчанию равно 0. Для изменения этого значения необходимо нажать на кнопку **Set**, при этом появится диалог, в котором нужно задать требуемый адрес загрузки в шестнадцатеричном формате. Максимальное значение, которое можно указать, равно 0FFFFFFh, исходя максимального адресного пространства, равно 16 Мбайт, используемого в формате файлов Intel HEX 86 (см. Раздел «Используемые форматы файлов»).

Для запуска команды на выполнение необходимо нажать на кнопку **Start**. После запуска команды на экране появляется статусное окно, отображающее текущее состояние программатора и результаты выполненных действий.

Ниже приведен пример статусного окна при выполнении команды программирования внешней flash-памяти:



## Программирование «шаг за шагом»

Процедура программирования состоит из следующих действий:

1. Снять напряжение питания;
2. Установить перемычку “Bootstrap”;
3. Подать напряжение питания;
4. Произвести сброс контроллера (для контроллеров не имеющих функцию сброса по RS232);
5. Настроить параметры программатора: тип контроллера, тип флэш-памяти, номер подключенного последовательного порта, процедуру программатора и указать список файлов с соответствующими адресами программирования;
6. Нажать кнопку **Start** для начала программирования.

Сброс подключенного контроллера осуществляется после запуска выбранной команды на выполнение автоматически, используя для этого сигналы RS232 – DTR и RTS.

Если такой сброс не предусмотрен для подключенного контроллера, то после нажатия на кнопку **Start** будет выведено напоминающее о сбросе сообщение, после чего необходимо вручную сбросить контроллер.

Если же сброс по RS232 предусмотрен, но он подключен только с использованием сигналов RS232 – RX, TX и GND, то программатор не сможет сбросить контроллер автоматически. В этом случае также необходимо производить сброс вручную.

## Используемые форматы файлов

Программатор R16x Flash Programmer может программировать flash-память данными из файлов, представленных либо в двоичном, либо в шестнадцатеричном форматах.

Шестнадцатеричный формат файла или Intel HEX 86 имеет расширения *hex* или *h86*. Для двоичного формата можно использовать любые другие расширения. Двоичный формат представляет собой непосредственно сами данные для программирования.

### Формат Intel HEX 86

Данный формат представляет собой набор ASCII строк в соответствии со стандартом Intel HEX 86. Каждая строка состоит из одной записи, состоящей из шестнадцатеричных символов, и начинается с символа двоеточия, а заканчивается символами признака конца строки.

#### Формат записи:

Файл может состоять из неограниченного числа записей. Каждая запись состоит из пяти полей, находящихся в следующей последовательности:

```
:11aaaatt [ dd...] cc
```

где :

: С этого символа начинается каждая запись.

11 Размер поля данных (dd) в байтах.

aaaa Адрес первого байта данных.

tt Типа записи:

00 запись с данными

01 признак конца файла

02 запись с номером страницы (параграфа)

04 запись с номером сегмента

- dd Данные, каждый байт которых представлен парой шестнадцатеричных символов. Количество байт данных должно соответствовать значению в поле ll.
- сс Контрольная сумма записи, которая вычисляется как сумма всех полей записи по модулю 256, а затем вычисляется дополнение от полученного значения

**Например:**

**:10246200464C5549442050524F46494C4500464C33**

где:

- 10 Количество байт данных
- 2462 Адрес первого байта поля данных
- 00 Запись с данными
- 46...4C Данные, каждый байт которых представляет собой пару шестнадцатеричных символов
- 33 Значение контрольной суммы

**Запись признака конца файла**

Запись признака конца файла является последней записью в файле, и всегда имеет следующий вид:

**:00000001FF**

где:

- 00 Нет данных
- 0000 Адрес равен 0, но так как данных нет, то адрес обычно просто игнорируется
- 01 Тип записи – конец файла
- FF Контрольная сумма записи, которая вычисляется как  $01h + NOT(00h + 00h + 00h + 01h)$ .

### Запись с номером страницы

Данная запись используется для указания номера используемой страницы, которая имеет размер равный 16 байт. Значение номера страницы является начальным смещением для всех последующих записей с данными. Действие данной записи распространяется до очередной записи с номером страницы или с номером сегмента.

Пример записи:

:020000021000EC

где:

02	Количество байт данных равно двум
0000	Значение адреса игнорируется
02	Тип записи
1000	Номер страницы равный 1000h (адрес будет равен значению номера страницы, сдвинутому на 4 бита влево, в данном случае – 10000h)
EC	Значение контрольной суммы записи

### Запись с номером сегмента

Данная запись используется для указания номера используемого сегмента, который имеет размер равный 64 Кбайтам. Значение номера сегмента является начальным смещением для всех последующих записей с данными. Действие данной записи распространяется до очередной записи с номером страницы или с номером сегмента.

Пример записи:

**:0200000400FFFB**

где:

02	Количество байт данных равное двум
0000	Значение адреса игнорируется
04	Тип записи
00FF	Номер страницы равный 0FFh (адрес будет равен значению номера сегмента, сдвинутому на 16 бит влево, в данном случае – 0FF0000h)
FB	Значение контрольной суммы записи

Пример файла в формате Intel HEX 86.

**:020000021000EC**

**:10C20000E0A5E6F6FDFFE0AEE00FE6FCFDFFE6FD93**

**:10C21000FFFFFF6F50EFE4B66F2FA0CFEF2F40EFE90**

**:10C22000F04EF05FF06CF07DCA0050C2F086F097DF**

**:10C23000F04AF054BCF5204830592D02E018BB03F9**

**:020000020000FC**

**:04000000FA00000200**

**:00000001FF**

## Запуск из командной строки

Программатор R16x Flash Programmer позволяет задавать параметры из командной строки.

### **Внимание!**

*Если программатор был запущен без параметров, то будут использоваться значения, введенные в последней сессии работы или значения по умолчанию, если программатор был запущен в первый раз. Если же программатор был запущен из командной строки с параметрами, то будут использоваться только эти параметры, а для неуказанных параметров будут использоваться значения по умолчанию.*

**Синтаксис задания параметров в командной строке:**

```
Flash16x[.exe] /{b,h}:<file[.ext]> /{h,d,b}:<address> /cmd:<id> /board:<num>  
/f:<type> /com:<port> /br:<baudrate> /nclr /s /? /h
```

Параметры:

- /b:<file[.ext]>** Двоичный файл (ext – любое расширение)
- /h:<file[.ext]>** Шестнадцатеричный файл (ext – расширение *hex* или *h86*).
- /h,d,b}:<address>** Адрес загрузки для файла, указанного ранее в командной строке.

По умолчанию используется значение равное 0.

Адрес может задаваться в трех форматах:

h – префикс для шестнадцатеричного формата

d – префикс для десятичного формата

b – префикс для двоичного формата

**/cmd:<id>** Номер команды

1 – программирование внешней flash-памяти

2 – сравнение данных во внешней flash-памяти с данными из файла

По умолчанию используется значение равное 1

**/board:<num>** Номер используемой платы

1 – M164

2 – M167-1

3 – M167-2

4 – 167-104

5 – MF167

6 – E167-3UM

7 – G167F-3U1

По умолчанию используется значение равное 1

**/f:<type>** Номер установленной на плате Flash-памяти

1 – 29F040 и 29F010

По умолчанию используется значение равное 1

**/com:<port>** Номер последовательного порта (COM)

1 - COM1    2 - COM2

3 - COM3    4 - COM4

По умолчанию используется значение равное 2

**/br:<baudrate>** Скорость последовательного порта

Возможные значения для скоростей:

110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000, 57600, 115200

По умолчанию используется значение равное 9600

**/nclr** Не очищать flash-память перед программированием

По умолчанию flash-память очищается

**/s** Запуск команды на выполнение сразу после загрузки программатора

**/?, /h** Вывести подсказку по использованию командной строки

### Пример:

```
>flash16x.exe /h:data.hex /b:..\code.bin /h:ffe0 /com:1 /cmd:1 /board:1
```

В данном примере:

**/board:1** Используемая плата контроллера - M167-1

**/cmd:1** Команда – программирование внешней flash-памяти

**/h:data.hex /b:..\code.bin** Данные для программирования – файл data.hex и code.bin

**/h:ffe0** Адрес загрузки для второго файла – равен 0FFE0h

Адрес загрузки для первого файла – не требуется