

### Работа с программой ASISP

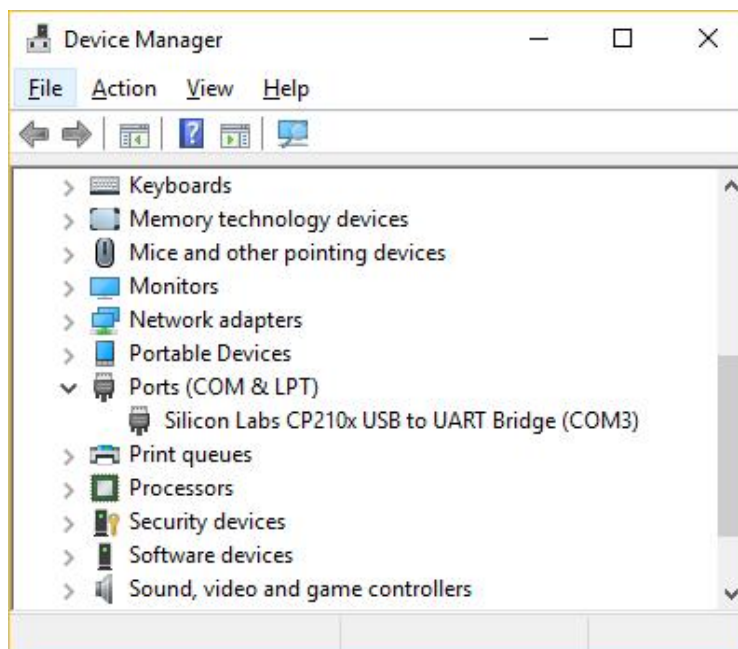
**Внимание!** Необходимо при использовании программаторов AS2M, AS3, AS3E и AS3M соединить общий провод платы с программируемым микроконтроллером и корпус компьютера, в целях защиты USB-порта компьютера. Для программаторов AS-4/AS4E, которые имеют гальваническую развязку, соблюдение этого правила не является необходимым.

**ВАЖНО:** на разъем VH-10 программатора должно подаваться напряжение питания в диапазоне 3,3 – 5,0 В.

Программа ASISP функционирует в среде Windows 7/8.1/10, Linux (32- и 64-разрядные версии) и предназначена для совместной работы с программаторами AS2M, AS3, AS3E, AS3M, AS-4, AS4E. Обеспечивается программирование микроконтроллеров серий ATmega/ATtiny/AT89S/AT90. Программа может запускаться с графическим пользовательским интерфейсом, также предусмотрена возможность вызова программы из командной строки. Текущую версию программы ASISP и текущую версию USB-драйвера можно загрузить отсюда: [http://www.as-kit.ru/asisp\\_current\\_version](http://www.as-kit.ru/asisp_current_version)

Перед началом работы с программаторами, имеющими порт USB (все модели, кроме AS2M), нужно установить USB-драйвер Silicon Lab Virtual COM Port (VCP).

После установки USB-драйвера, при подключении программатора к порту USB, операционная система назначит программатору номер виртуального COM-порта.

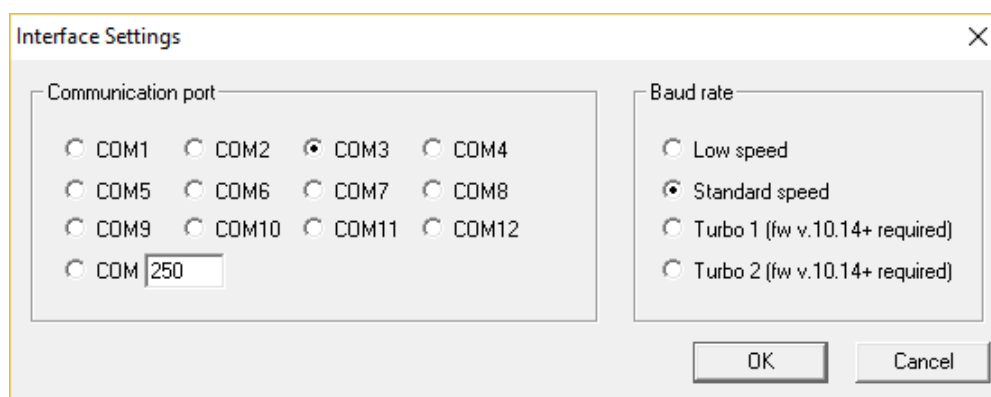


Этот номер COM-порта нужно посмотреть в окне "Device Manager" и внести в окно **Interface Settings** программы ASISP.

## Первичная настройка программы ASISP

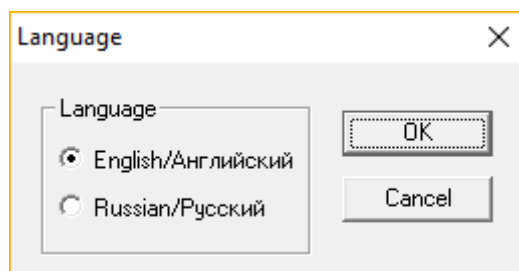
При первом запуске программы ASISP в настройки ее интерфейса нужно внести номер COM-порта, назначенного программатору системой, и стандартную скорость обмена по COM-порту. Это выполняется в окне **Настройки интерфейса** программы ASISP. Поддерживаемый диапазон COM-портов от 0 до 255. Настройки сохраняются после выхода из программы, и при следующих запусках программы ASISP подставляются автоматически.

### Окно настроек интерфейса программы ASISP



На «медленных» компьютерах может не хватить ресурсов для обслуживания COM-порта на стандартной скорости, вследствие чего возможно нарушение синхронизации. В таких случаях рекомендуется выбрать низкую скорость. Повышенные скорости обмена доступны для программаторов, имеющих версию прошивки не ниже, чем 10.14. Окно «Настройки интерфейса» также доступно непосредственно из меню «Настройки» в главном окне программы ASISP.

После выбора русскоязычного интерфейса (главное окно, меню **Settings**, строка **Select language**) все информационные сообщения будут выводиться на русском языке.



Для параллельного программирования нескольких микроконтроллеров возможно одновременное подключение нескольких программаторов к одному компьютеру. При этом, с каждым экземпляром программатора следует запустить отдельную копию программы ASISP. Для запуска разных копий программы на рабочем столе компьютера можно использовать иконки разного цвета — набор иконок для выбора встроен в программу ASISP.

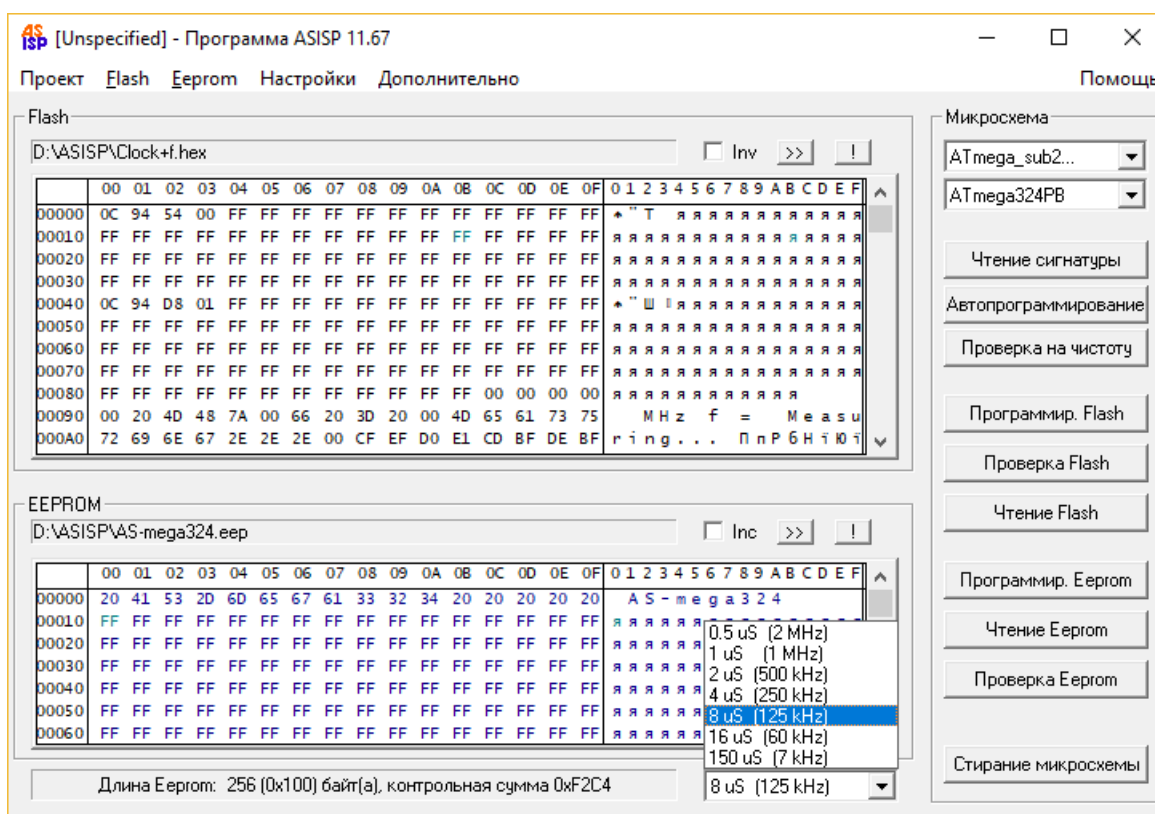
### Описание функций программы ASISP

Программа имеет интуитивно понятный интерфейс. Основные команды вынесены в главное окно, и доступ к ним осуществляется одним нажатием кнопки мыши или комбинацией клавиш на клавиатуре. Размеры главного окна можно изменять для более удобного отображения содержимого буфера памяти Flash. Содержимое обоих буферов можно редактировать, причем это можно делать как в области бинарных кодов, так и в области кодов ASCII. В правой части главного окна размещены кнопки для быстрого выполнения часто используемых команд.

Главное окно содержит следующие элементы:

- панель «ниспадающих» меню для работы с файлами и настройками;
- поле просмотра содержимого файла ПЗУ программы, предназначенного для загрузки в микроконтроллер, либо содержимого ПЗУ, считанного из микроконтроллера;
- поле с указанием имени файла ПЗУ программы;
- поле просмотра содержимого файла ЭСПЗУ, предназначенного для загрузки в микроконтроллер, либо содержимого ЭСПЗУ, считанного из микроконтроллера;
- поле с указанием имени файла ЭСПЗУ;
- поле с названием типа программируемой микросхемы;
- группа клавиш «быстрого доступа»;
- строка сообщений;
- строка с указанием текущих установок.

### Главное окно программы ASISP



Процесс программирования осуществляется нажатием соответствующих программных кнопок. Если



программируемая микросхема ранее содержала какую-либо информацию, или у нее были установлены биты защиты, микросхему необходимо предварительно стереть.

### Панель меню

Меню **"Проект (Project)"** — создание, загрузка, сохранение проекта;  
Меню **"Flash"** — операции с файлами, содержащими информацию о ПЗУ программ;  
Меню **"Eeprom"** — операции с файлами, содержащими информацию о ПЗУ данных;

Меню **"Настройки (Settings)"** — вызов меню настроек;  
Меню **"Дополнительно (Advanced)"** — вызов окна команд;  
Меню **"Помощь (Help)"** — вызов окна помощи ( в формате HTML).

Кнопка **"Inv"** управляет отображением порядка байтов в слове: старший-младший или младший-старший. Кнопка **"Inc"** включает функцию "Автоинкремент" для памяти EEPROM (см. ниже описание опции "Автоинкремент").

По нажатию кнопки **">>>"** вызывается окно загрузки файла, а нажатие кнопки **"!"** сразу загружает буфер Flash или EEPROM информацией из файла, путь к которому указан в строке над полем просмотра Flash и EEPROM. В нижней части основного окна находится служебная строка. Здесь отображается контекстная информация — версия программы, название программируемой микросхемы, контрольная сумма загружаемого файла и т.д. Справа от командной строки находится поле для быстрой установки скорости работы интерфейса SPI.

### Меню «Проект»

Для удобства работы с различными типами микросхем рекомендуется использовать файл проекта. Файлов проектов может быть несколько, что обеспечивает легкий переход к программированию микросхемы другого типа. Текущий тип микросхемы индицируется в соответствующем поле окна программы. Для AVR-микроконтроллеров тип микросхемы может быть установлен автоматически, при выполнении команды чтения сигнатуры. Для работы с остальными микросхемами тип микросхемы необходимо установить вручную. Затем можно производить любые операции с файлами, памятью микроконтроллера и его битами защиты. Все настройки для программирования — имена файлов, пути к файлам и тип микросхемы — сохраняются в файле проекта. Последний проект автоматически открывается при последующем запуске.

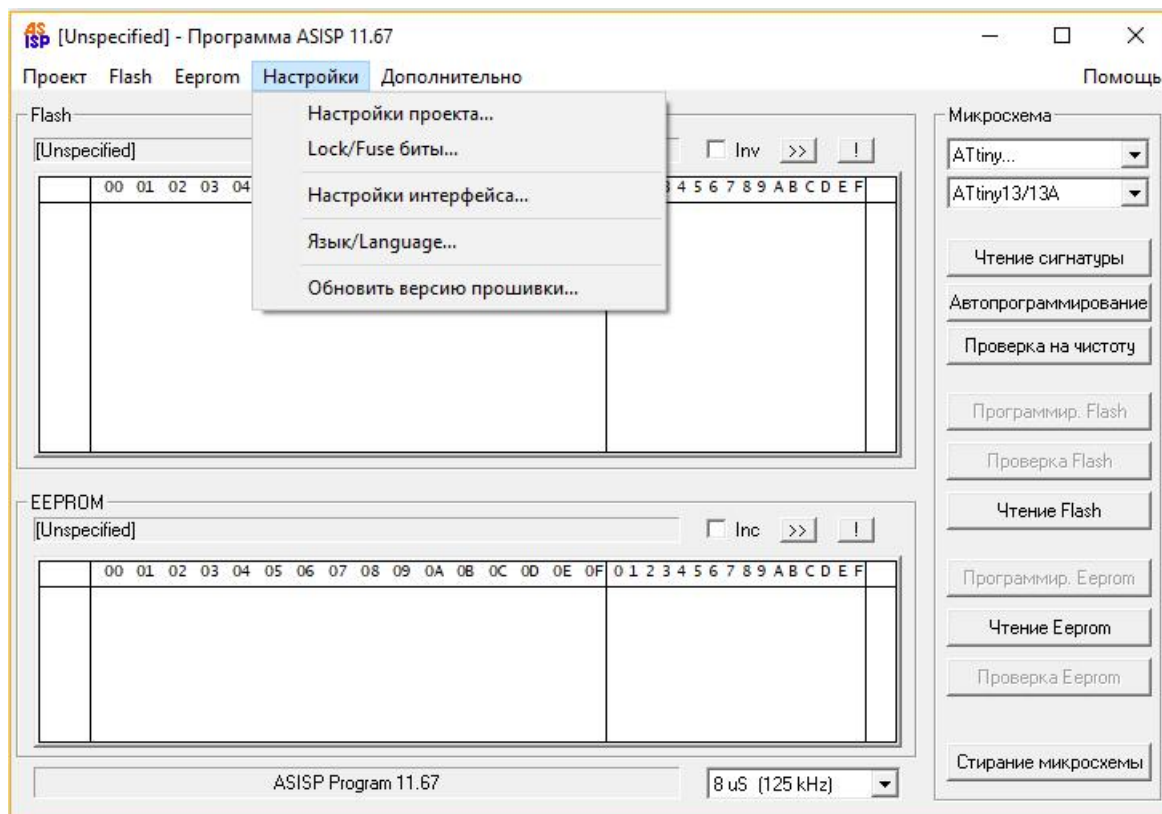
Файл проекта — это текстовый файл с расширением .as2, который доступен для просмотра и редактирования.

### Меню «Flash» и «EEPROM»

В этих меню собраны команды по работе с буферами Flash и EEPROM памяти. Программа ASISP позволяет загружать два файла по разным адресам как в буфер памяти программ, так и в буфер памяти данных. По команде: FLASH/открыть по адресу/ (FLASH/load At) или EEPROM/открыть по адресу/(EEPROM/Load At) можно загружать, например, заранее подготовленные фрагменты кода в заданную область памяти программ без дополнительной компиляции. Эта опция также очень удобна для размещения таблиц или наборов параметров по верхним адресам FLASH или EEPROM.

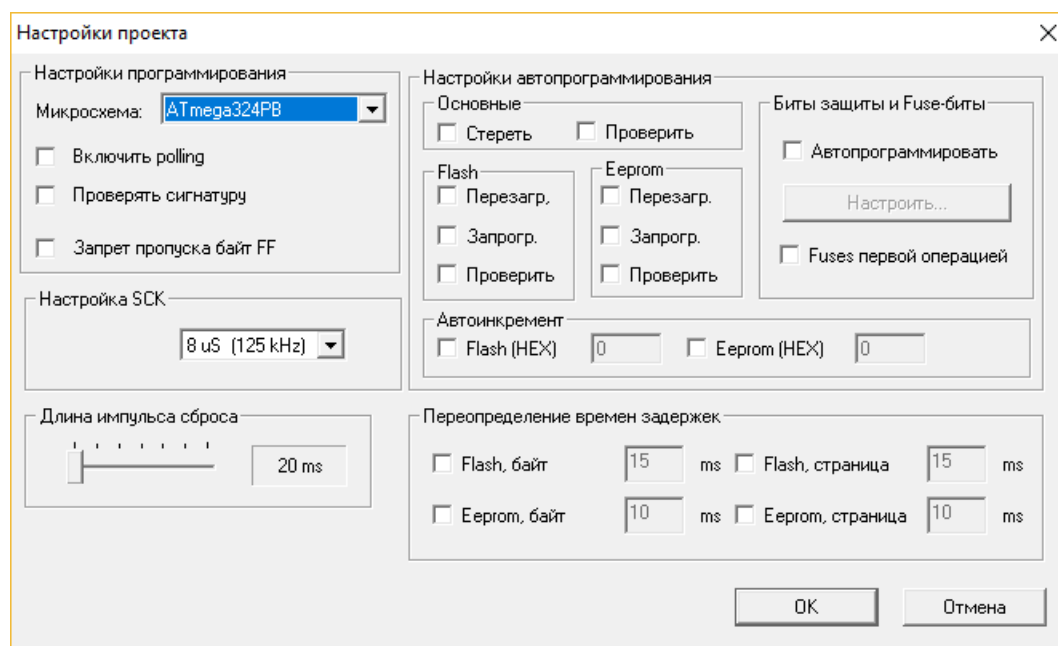
Поддерживается режим загрузки файлов Flash и EEPROM с использованием механизма «drag and drop», то есть простым перетаскиванием имени файла в поле буфера Flash и EEPROM основного окна программы ASISP.

### Меню «Настройки»



Верхняя строка этого меню открывает окно «**Настройки проекта**», содержащее команды для управления процессом программирования микросхемы. Отмеченные галочками команды можно запустить на выполнение в пакетном режиме.

### Окно «Настройки проекта»





### Настройки программирования

Здесь можно установить тип микросхемы для программирования и активировать функцию проверки сигнатуры перед выполнением каждой операции.

Для прошивки микросхем программа ASISP использует один из двух алгоритмов: стандартный и ускоренный (data polling). Стандартный алгоритм подразумевает достаточно большое фиксированное время для прошивки каждого байта, поэтому общая скорость прошивки не очень высокая. При ускоренном алгоритме (опция Enable data polling включена) производится проверка каждого запрограммированного пакета и, в случае подтверждения окончания программирования, программа переходит к следующему адресу. Так как реальное время программирования конкретного байта, как правило, меньше указанного в технических условиях на микросхему, общая скорость программирования повышается.

Опция отмены синхронизации применяется в случае, если программируемая микросхема нестабильно отвечает на запрос чтения сигнатуры. Такая ситуация изредка встречается при работе с микроконтроллерами прошлых лет выпуска. Вероятная причина — использование некондиционного экземпляра микросхемы из «серых» поставок.

Опция выбора частоты задающего генератора управляет длительностью импульса SCK в микросекундах при обмене данными между программируемым микроконтроллером и программатором. Для работы с микроконтроллерами семейства AT90/ATtiny/ATmega частота программирующих импульсов SCK должна быть не менее, чем в четыре раза ниже частоты задающего генератора, а для AT89S51, AT89S52 и AT89S8253 — не менее, чем в 16 раз ниже частоты задающего генератора программируемого микроконтроллера.

Опция выбора длительности импульса сброса помогает в тех случаях, когда в цепи «Reset» микроконтроллера установлен конденсатор повышенной емкости, например, 2,2 мкФ (рекомендованное значение составляет 100 нФ). В таком случае длительность этого импульса можно увеличить. Диапазон настроек: от 20 до 2000 миллисекунд.

### Настройки автопрограммирования

Выбор набор команд, которые будут выполняться при нажатии кнопки «Автопрограммирование» в главном окне программы.

Обычно в процессе отладки программы приходится часто перекомпилировать программу и производить операцию обновления прошивки. Для этого необходимо стереть микросхему, обновить содержимое буфера программы, выполнить собственно перепрограммирование и убедиться, что прошивка «легла» в память микроконтроллера без ошибок. Аналогичные процедуры могут требоваться и для памяти EEPROM. Отметив необходимые операции, можно выполнить их нажатием кнопки «Автопрограммирование» непосредственно из главного окна.

Выбранные опции автопрограммирования (стирание, перезагрузка / программирование / проверка файлов, биты защиты, FUSE-биты, автоинкремент FLASH и EEPROM) также можно сохранить в файле проекта.

### Автоинкремент

Эта опция предназначена для автоматической записи серийного номера изделия в память FLASH или EEPROM. Для этого в соответствующей строке вписывается шестнадцатеричный адрес хранения

серийного номера. При активации режима «Автоинкремент» программа определяет 4-байтную строку, в которой будет размещаться серийный номер. При этом, после окончания каждого цикла программирования, производится инкремент этого 4-байтного числа с учетом переноса.

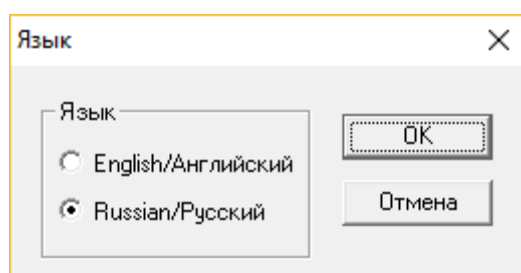
Пример:

- в поле EEPROM основного окна по адресам 04, 05, 06 и 07 записывается число 12 34 56 78 (серийный номер);
- в строке Eeprom (HEX) окна «Настройки проекта» вводится адрес 07 и активируется опция «Автоинкремент»;
- производится программирование микросхемы.

После окончания программирования серийный номер примет следующее значение: 12 34 56 79. При размещении серийного номера в пространстве FLASH памяти микроконтроллера необходимо на этапе создания исходного кода программы зарезервировать необходимое пространство.

### Окно «Язык/Language»

Вызывается из меню «**Настройки**». Окно предназначено для оперативного изменения языка меню и команд программы ASISP.



### Окно «Lock/Fuse биты»

#### Важная информация

Для перевода AVR-контроллера в режим внутрисхемного программирования используется вывод Reset. Некоторые типы AVR-контроллеров допускают использование этого вывода в качестве стандартного порта ввода/вывода или отладочного порта. Переключение режима работы вывода Reset производится установкой fuse-битов RSTDSBL и DWEN соответственно. Следует помнить, что после программирования любого из этих fuse-битов, внутренние цепи, отвечающие за сброс микроконтроллера, отключаются от вывода Reset, и режим внутрисхемного программирования контроллера через порт SPI становится недоступным.

Вызывается из третьей сверху строки в меню «**Настройки**». Окно содержит набор настроек для программирования Lock и Fuse битов AVR-контроллеров. Набор этих битов отличается у разных микросхем. Значения, установленные в данном окне, действуют на протяжении текущего сеанса работы с программатором. Чтобы сохранить выбранные опции для следующих сеансов работы, нужно перед закрытием окна выполнить команду «Сохранить и выйти».

Настройки Lock/Fuse битов для автопрограммирования

**Биты защиты**

Нет

LB1: Защита от записи

LB1+LB2: Защита от Ч&З

LB1+LB2+LB3 (только 895)

**Boot Lock биты**

BLB01     BLB11

BLB02     BLB12

**Fuse биты**

7	CKDIV8	OCDEN	Нет
6	CKOUT	JTAGEN	Нет
5	SUT1	SPIEN	Нет
4	SUT0	WDTON	Нет
3	CKSEL3	EESAVE	CFD
2	CKSEL2	BOOTSZ1	BODLEVEL2
1	CKSEL1	BOOTSZ0	BODLEVEL1
0	CKSEL0	BOOTRST	BODLEVEL0

Low Byte      High Byte      Ext. Byte

Нажатая кнопка обозначает, что fuse бит запрограммирован (0)

Программировать
Прочитать
Проверить

Стереть микросхему
Сохранить и выйти
Выйти без сохранения

### Меню «Дополнительно»

The screenshot shows the ASISP 11.67 software interface. The 'Дополнительно' (Additional) menu is open, displaying the following options:

- Проверка на чистоту...
- Чтение сигнатуры...
- Чтение калибровочных байтов...
- Информация о программаторе...** (highlighted)

The background interface shows the Flash memory table with addresses 00000 to 000A0 and hex values. The EEPROM section is also visible with addresses 00000 to 00060. The status bar at the bottom indicates 'AS4x Programmer 10.15 (Серийный номер A4 D5 43 A7 0B)' and a speed of '8 uS (125 kHz)'.

Здесь собраны команды, позволяющие прочитать информацию о программируемой микросхеме и собственно программаторе.

Проверка на чистоту (Empty check) — проверка программируемого микроконтроллера на «чистоту».

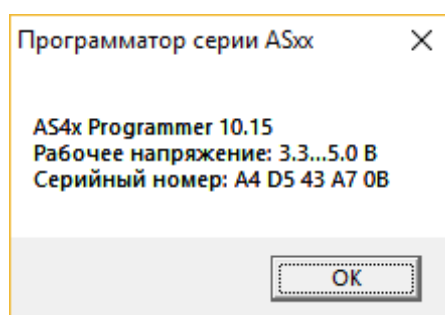


Чтение сигнатуры (Read signature) — команда считывает сигнатуру программируемого микроконтроллера, может использоваться для контроля правильности подключения ISP-разъема к программируемой микросхеме.

Чтение калибровочных байтов (Read Calibration bytes) — чтение калибровочных байтов (в микросхемах, которые имеют эту опцию).

Информация о программаторе — отображение версии программы (firmware), зашитой в микроконтроллер программатора. Может использоваться для проверки правильности подключения программатора к компьютеру.

### Окно «Информация о программаторе»



### Запуск программы ASISP из командной строки

Программа ASISP имеет оконный интерфейс, упрощающий процедуру составления файла проекта, а также позволяющий просматривать и редактировать содержимое окон FLASH (память программ) и EEPROM (память данных). Однако, после того как файл проекта создан и сохранен, можно пользоваться вызовом программы ASISP из командной строки, что в ряде случаев является более удобным.

Формат вызова следующий: `d:\asisp1164 «d:\AVR\clock.as2» -autoprogram`

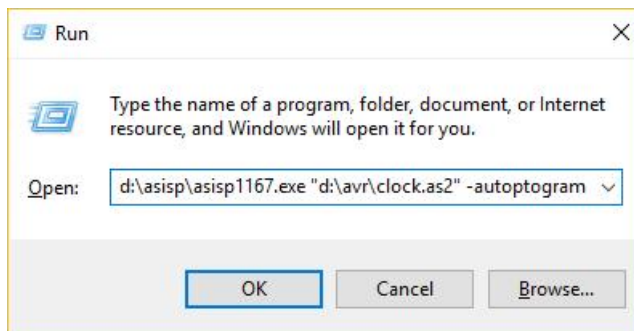
`d:\asisp1164` — имя программы с указанием полного пути

`«d:\AVR\clock.as2»` — имя файла проекта с указанием полного пути (кавычки обязательны!)

`-autoprogram` — необязательный ключ. Если он указан, программа ASISP выполняется без отображения основного окна, причем выполняются команды, отмеченные «галочками» в окне «Настройки проекта».

Файл проекта имеет текстовый формат и расширение `.as2`. Он содержит информацию о настройках и командах, которые будут выполняться в пакетном режиме при нажатии кнопки «Автопрограммирование» в основном окне программы ASISP.EXE. По умолчанию файл создается в папке с программой ASISP при выполнении команды «Проект/Сохранить». Если выбрать команду «Сохранить как», то можно указать желаемое место хранения файла. Файл проекта формируется заранее, при работе с окном «Настройки проекта». Изображение окна «Настройки проекта» приведено выше.

Окно вызова программы ASISP из командной строки:



В программе ASISP, начиная с версии версии 11.58, расширены функции по созданию лог-файла при использовании командной строки. Теперь имя и расположение лог-файла должно быть помещено в первую главу файла проекта (этот файл имеет расширение .as2). При программировании из командной строки лог записывается в этот файл. Вот пример добавления строки:

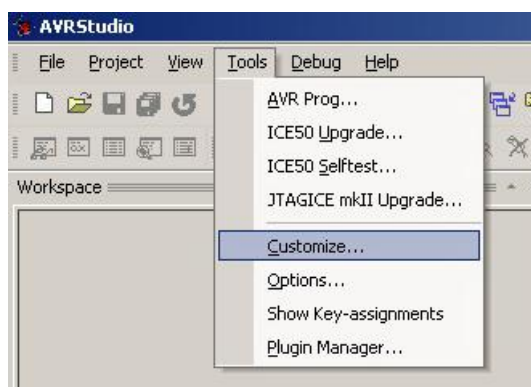
```
[Files]
Flash=D:\programs\Clock+f.hex
Eeprom=D:\programs\AS-megaM.eep
FlashInvert=0
Log=D:\programs\AS-megaM.log
```

Теперь можно создавать различные сценарии программирования через bat-файл. Один из примеров — ниже:

```
@echo off for %i in (odin dva tri) do echo Begin %i >> D:\programs\AS-megaM.log && asisp1158.exe
«d:\programs\clock+f.as2» -autoprogram
&& pause && echo End %i >> D:\programs\AS-megaM.log
```

## Добавление вызова программы ASISP.EXE в меню Atmel AVR Studio 4.18

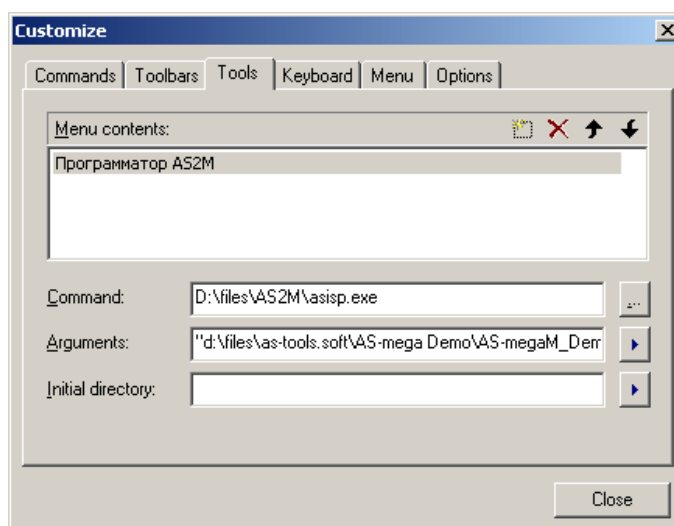
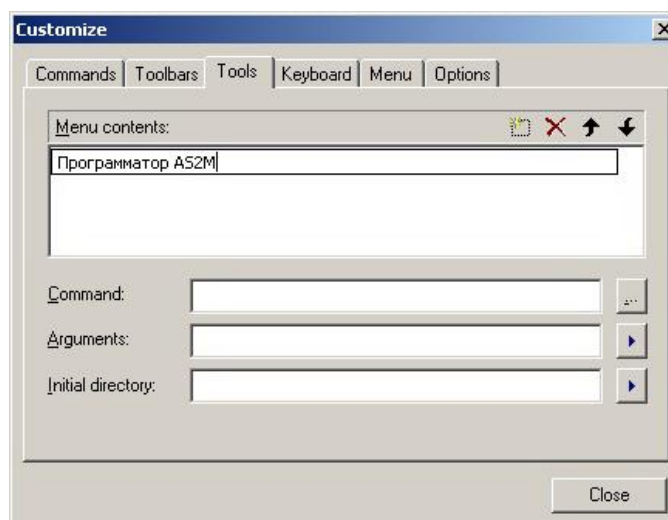
В меню Tools главного окна AVR Studio выбрать пункт Customize:



— Выбрать закладку Tools. Двойным нажатием кнопки мыши или нажав Insert на клавиатуре, добавить новую команду в список и назвать ее, например, “Программатор AS2M”:

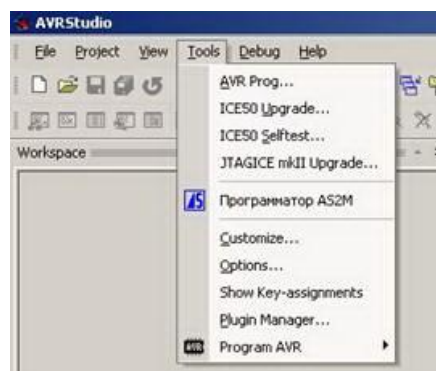
— Указать путь к исполняемому файлу программатора ASISP.EXE, введя его непосредственно в поле для ввода “Command”, или нажав на кнопку «...» справа от этого поля.

— Указать путь к файлу проекта программатора (файл с расширением .as2, введя его непосредственно в поле для ввода “Arguments”, или нажав на кнопку “>” справа от этого поля:



*Примечание.* В случае, если в поле “Arguments” установлен ключ “- autoprogram”, программирование микросхемы будет произведено без вызова оконной оболочки программы ASISP.EXE.

В меню Tools будет добавлен пункт “Программатор AS2M”:

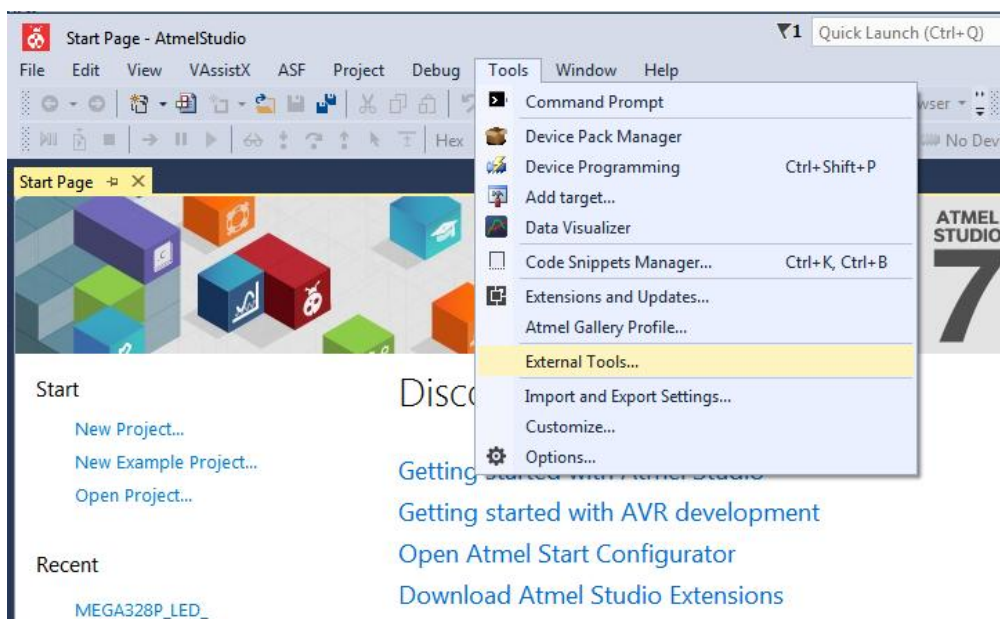


Строка вызова “Программатор AS2M» успешно добавлена в меню AVR Studio.

Файл проекта содержит информацию о настройках и командах, которые будут выполняться в пакетном режиме при нажатии кнопки «Автопрограммирование» в основном окне программы ASISP.EXE. Команды, отмеченные «галочками» также будут выполнены, если установлен ключ “-autoprogram”.

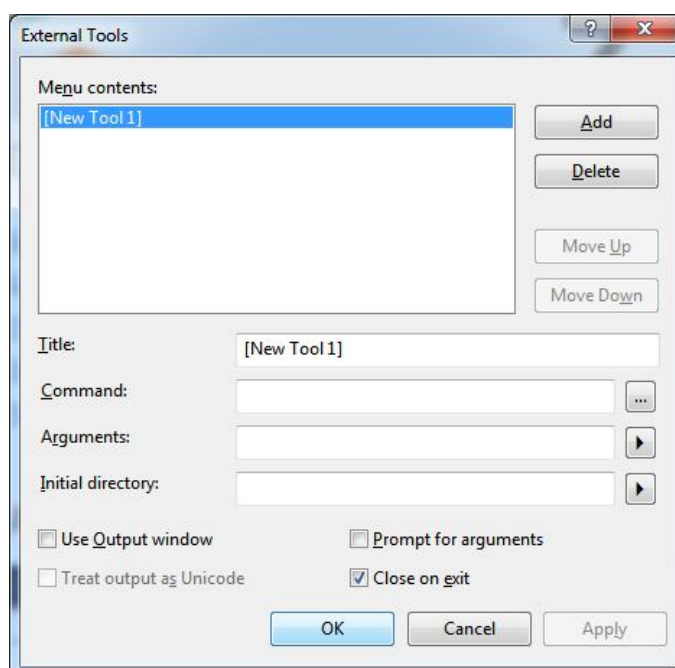
## Добавление вызова программы ASISP.EXE в меню пакета Atmel Studio 7

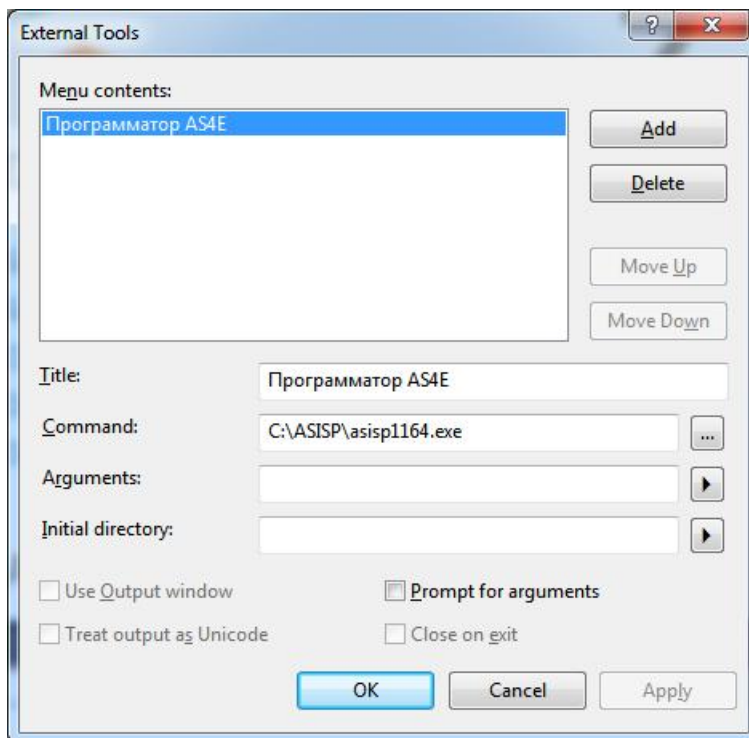
— В меню Tools главного окна Atmel Studio выбрать пункт External tools..:



Отобразится окно External Tools:

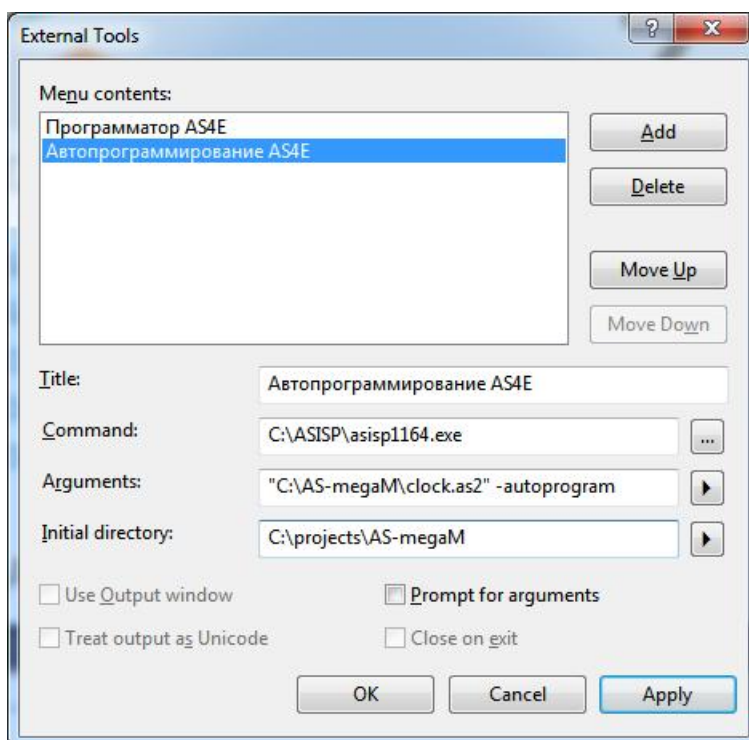
- Заполнить поле «Title» и нажать кнопку «Add»
- В поле «Command» указать путь к программе asisp.exe
- Нажать кнопку OK:



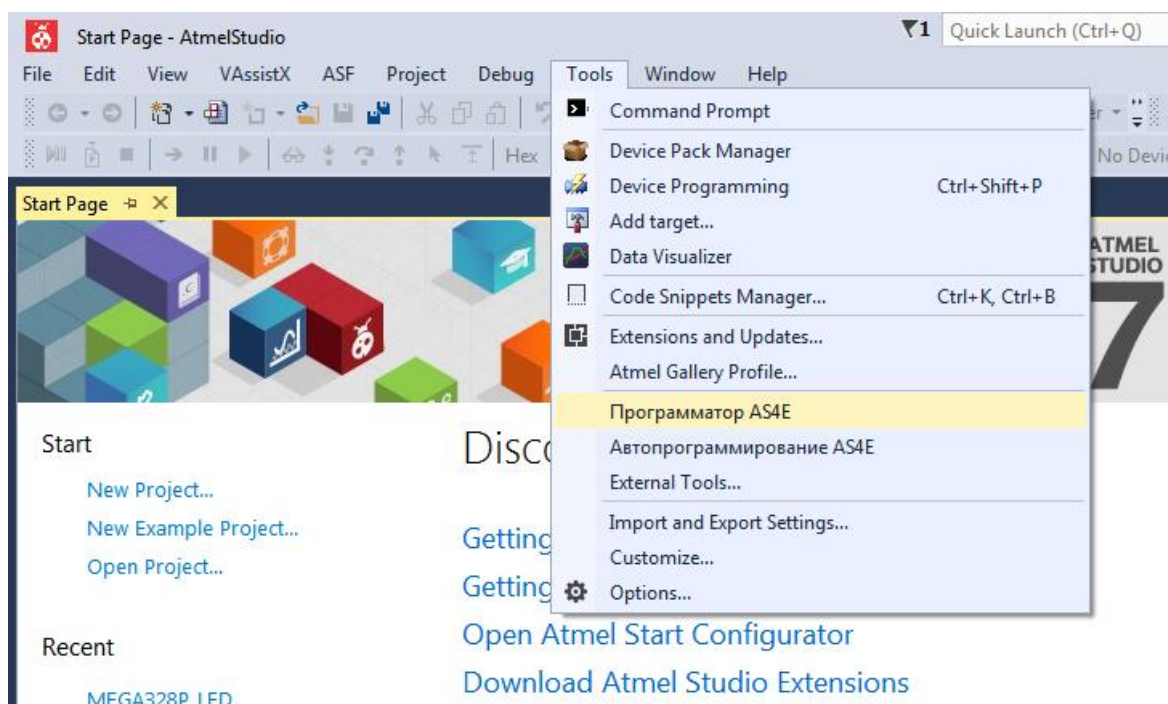


При желании можно добавить еще одну конфигурацию вызова программы ASISP, например, для запуска режима «Автопрограммирование»:

- Заполнить поле «Title» и нажать кнопку «Add»
- В поле «Command» указать путь к программе asisp.exe
- В поле «Initial directory» указать путь к файлу проекта
- Нажать кнопку ОК.



В результате в меню External tools главного окна Atmel Studio будут добавлены две строки:



Дополнительную информацию по программе ASISP можно получить здесь:  
<http://www.as-kit.ru/asisp-program-quick-view>