

Руководство Пользователя

Версия 1.0 (14/04/2013)

Jetlink SuperPro

Эмулятор с функцией автономного программатора



Jetlink SuperPro

ОГЛАВЛЕНИЕ

Комплектация.....	3
Информация для заказа.....	3
Краткое описание эмулятора.....	3
Работа в режиме эмулятора.....	4
IAR EWARM.....	4
Keil MDK.....	8
Программирование Flash памяти микроконтроллеров.....	10
Утилита J-Flash.....	10
Автономное программирование.....	14
Порядок действий.....	14
MSD режим.....	16
Управление по Telnet.....	17
Команды управления через UART.....	18
Ответы от эмулятора.....	19
Назначение контактов разъема JTAG.....	20
Назначения сигналов разъема в режиме SWD.....	21
Управление питанием подключенного микроконтроллера.....	22
Проверка серийного номера эмулятора.....	22
Установка IP адреса эмулятора.....	22
Встроенный WEB сервер.....	23
Обновление внутреннего Firmware эмулятора.....	25
История исправления документа.....	25

КОМПЛЕКТАЦИЯ

Наименование	Количество
Упаковочная коробка	1 шт
Эмулятор Jetlink SuperPro	1 шт
Блок питания 5В	1 шт
USB кабель	1 шт
Ethernet кабель	1 шт
20 контактный шлейф	1 шт

ИНФОРМАЦИЯ ДЛЯ ЗАКАЗА

Код заказа	Описание
Jetlink SuperPro	Эмулятор с функцией автономного программатора микроконтроллеров архитектуры ARM

КРАТКОЕ ОПИСАНИЕ ЭМУЛЯТОРА

Эмулятор – программатор Jetlink SuperPro предназначен для загрузки внутренней Flash/RAM памяти микроконтроллеров и отладки с помощью IDE, таких как:

- IAR EWARM
- KEIL MDK
- Freescale CodeWarrior
- Rowley CrossWorks
- Coocox CoIDE

Эмулятор поддерживает следующие архитектуры:

- ARM7/9/11
- Cortex-M0/M0+/M1/M3/M4
- Cortex-A5/A8/A9
- Renesas RX610/621/62N/630/631/63N

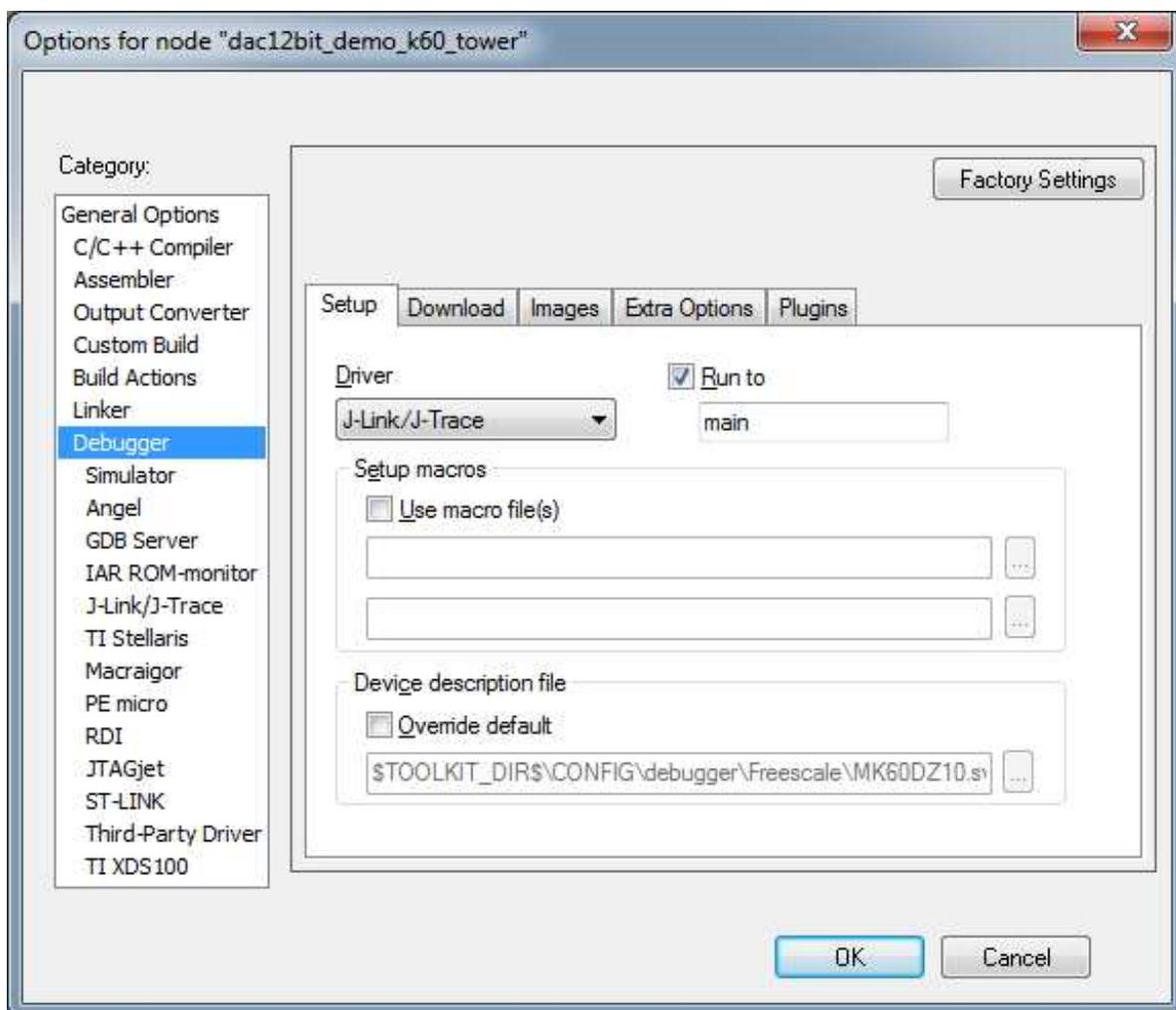
Эмулятор имеют внутреннюю память объемом 128 Мбайт для хранения файлов прошивок, для автономного программирования на производстве.

РАБОТА В РЕЖИМЕ ЭМУЛЯТОРА

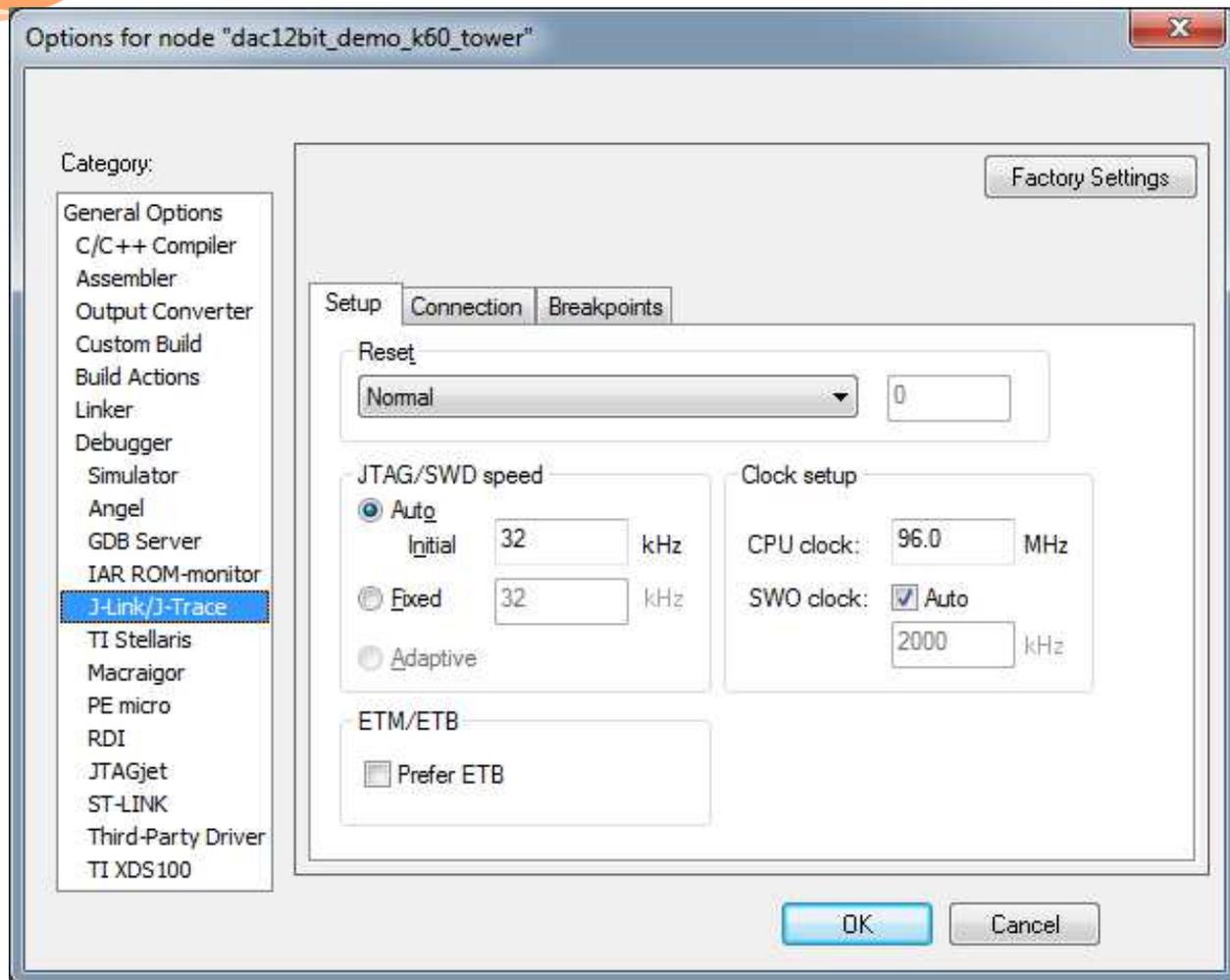
В режиме эмулятора отладчик работает как стандартный J-Link. Для подключения к компьютеру может быть использован как интерфейс USB 2.0 (Full Speed) так и сеть Ethernet (10/100 Mbit). При подключении по сети, эмулятор имеет гальваническую развязку от компьютера, что в некоторых случаях необходимо. Также скорость работы по сети выше чем через интерфейс USB. Для питания эмулятора при сетевом подключении используется блок питания 5В, входящий в комплект поставки. Ниже приведено описания настройки эмулятора в средах Keil и IAR.

IAR EWARM

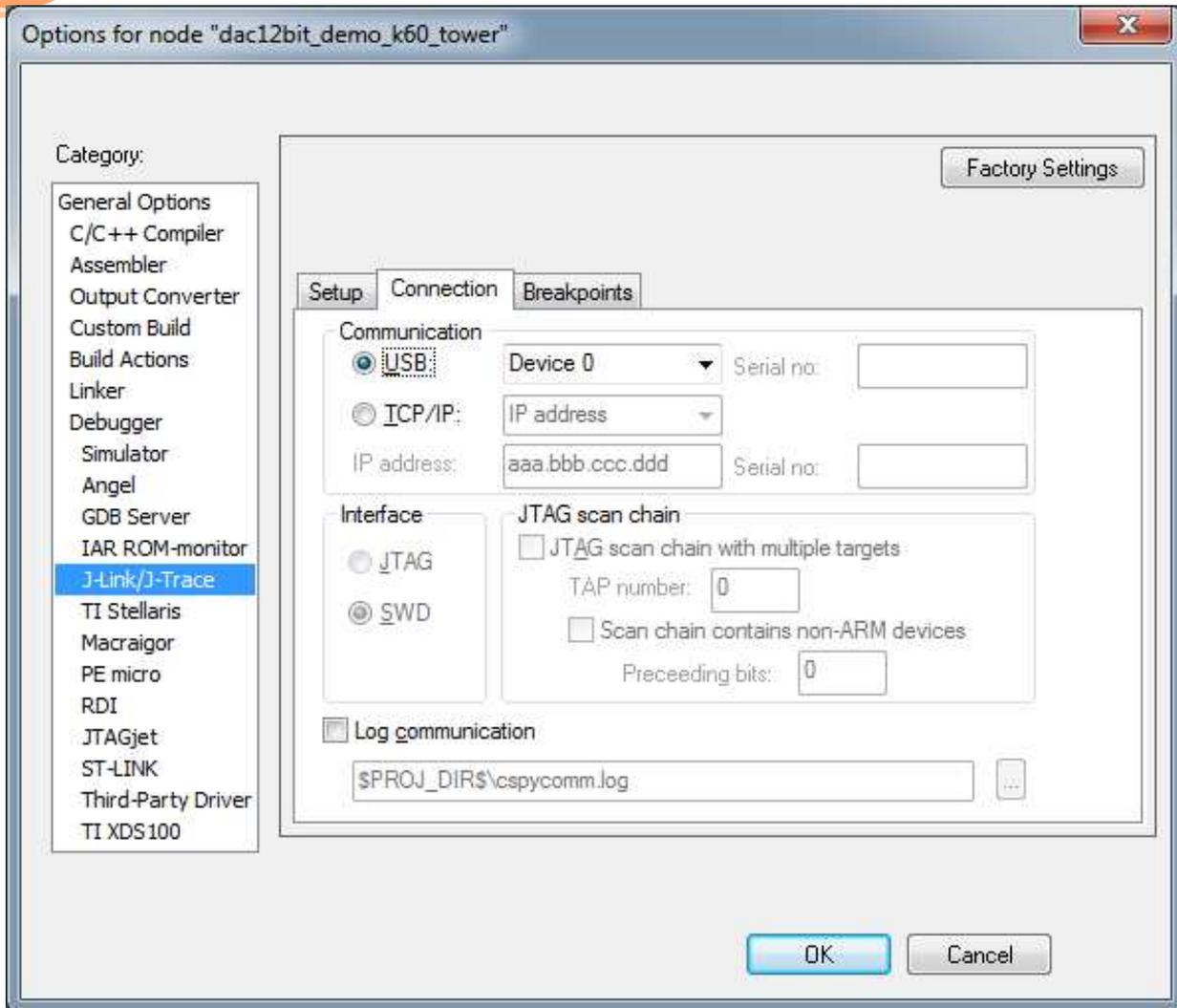
Заходим в Project - Options, переходим на вкладку Debugger



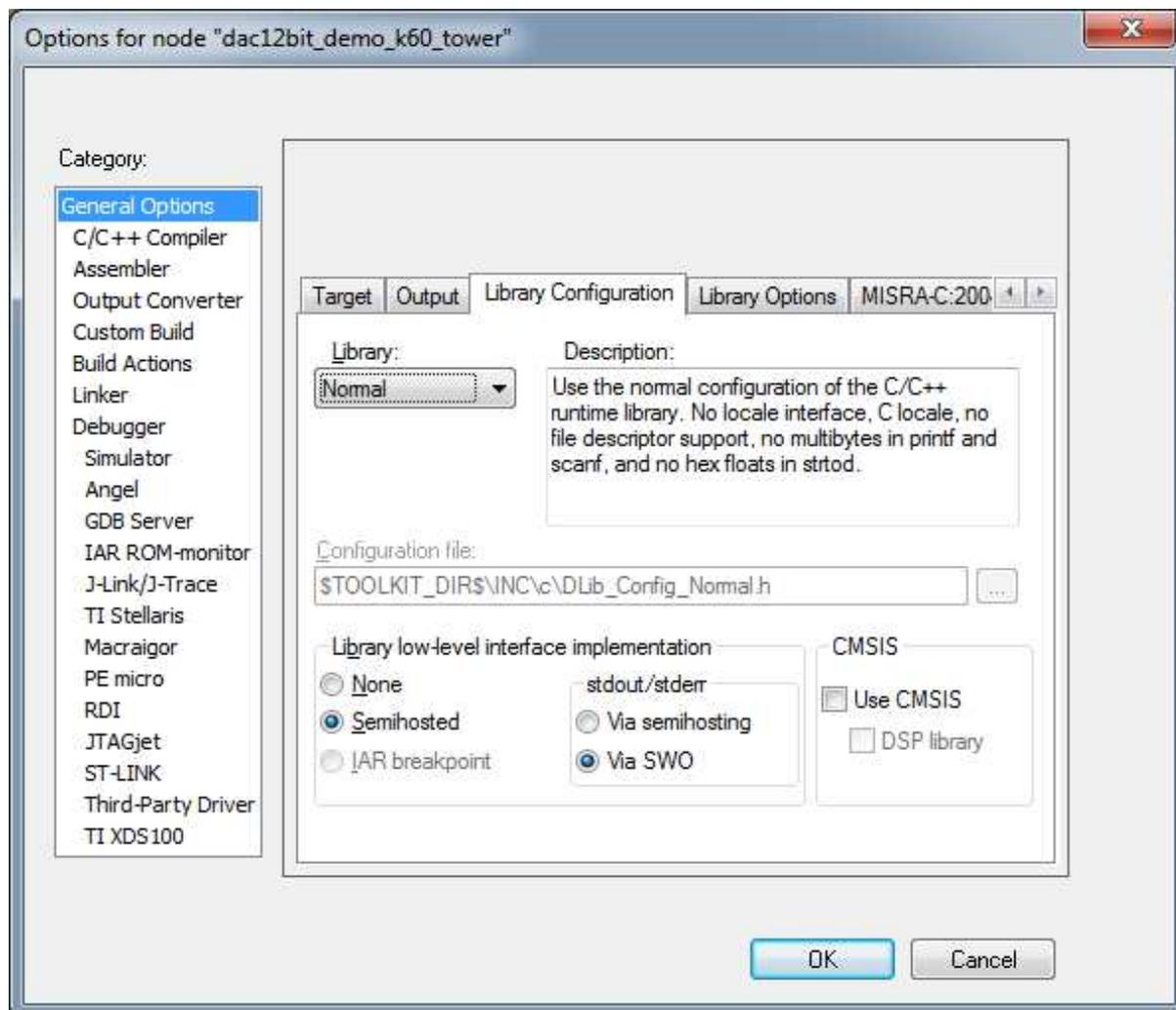
В выпадающем списке Driver выбираем J-Link/J-Trace. Переходим на вкладку J-Link/J-Trace



Устанавливаем режим тактирования (Auto – автоматическая установка скорости, Fixed – заданная пользователем или Adaptive – изменяемая эмулятором в зависимости от скорости работы интерфейса JTAG процессора). Переходим на вкладку Connection, здесь можно задать интерфейс подключения эмулятора (USB или Ethernet), а также режим работы интерфейса JTAG/SWD. Режим SWD может быть использован только с микроконтроллерами Cortex-M0/M0+/M1/M3/M4, в микроконтроллерах с архитектурой ARM7/9/11 SWD отсутствует.

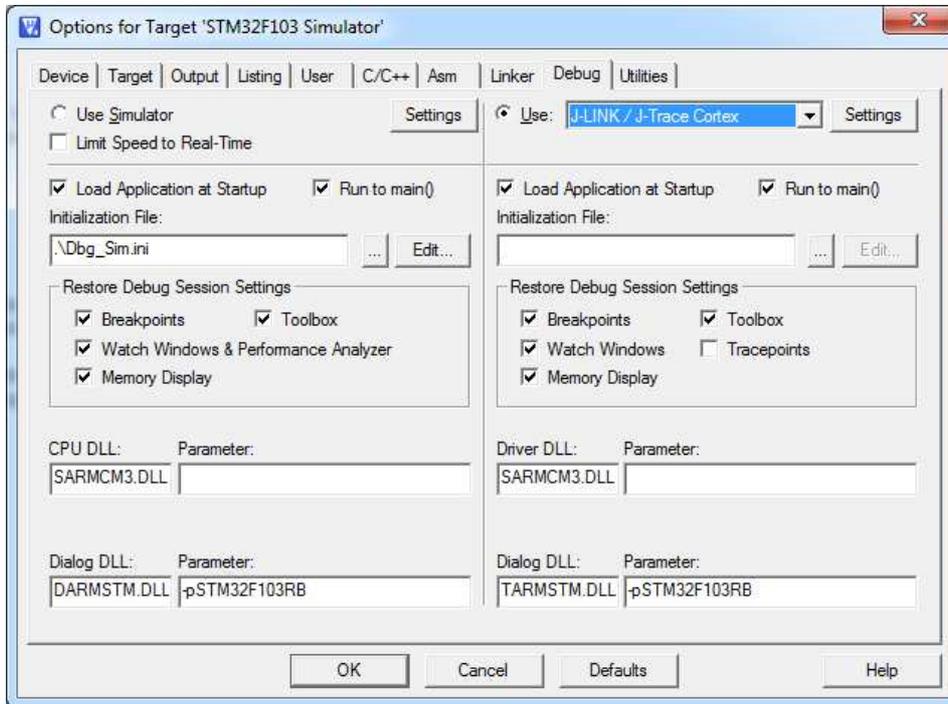


Для использования SWO при отладке микроконтроллеров Cortex-M0/M0+/M1/M3/M4 необходимо выбрать данный режим на вкладке General Options.

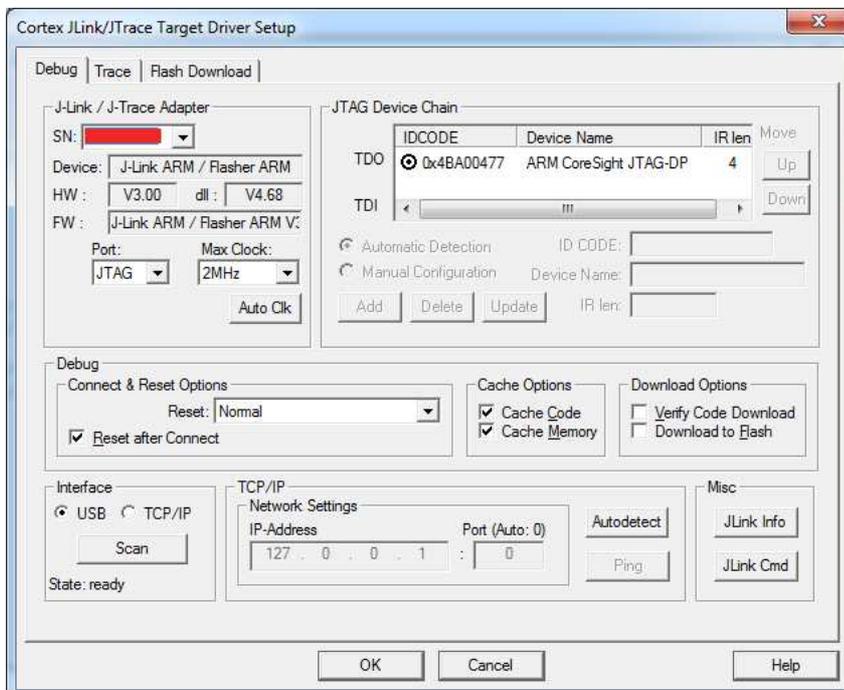


KEIL MDK

Заходим в Project - Options... , переходим на вкладку Debug, в выпадающем списке выбираем J-LINK/ J-Trace Cortex

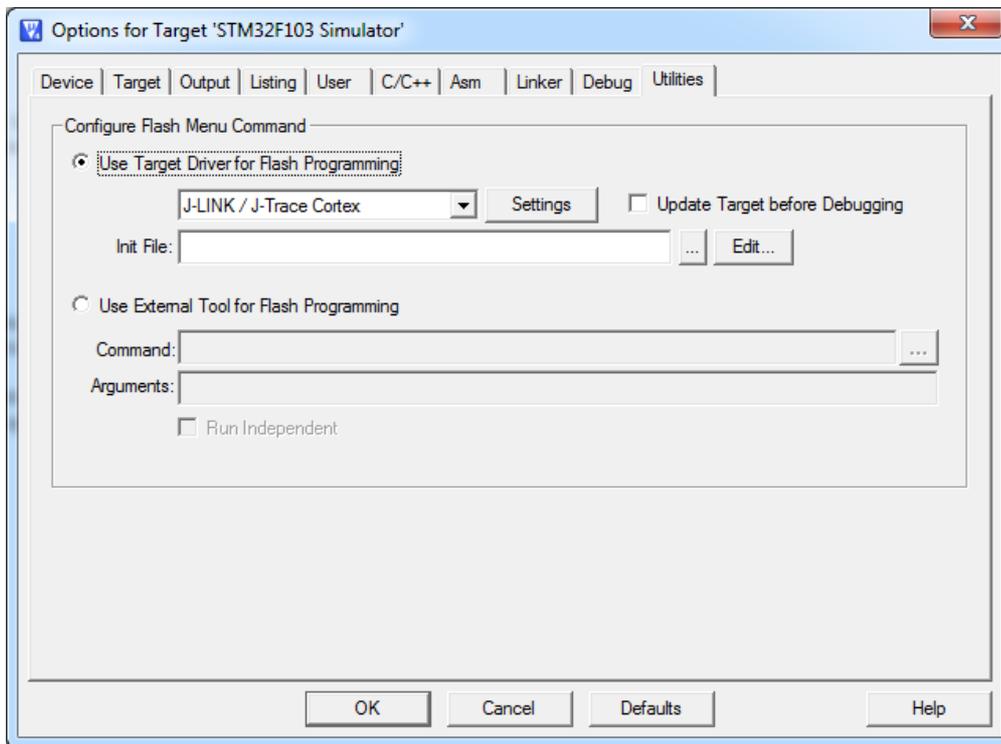


Нажимаем кнопку Settings рядом с выпадающим списком

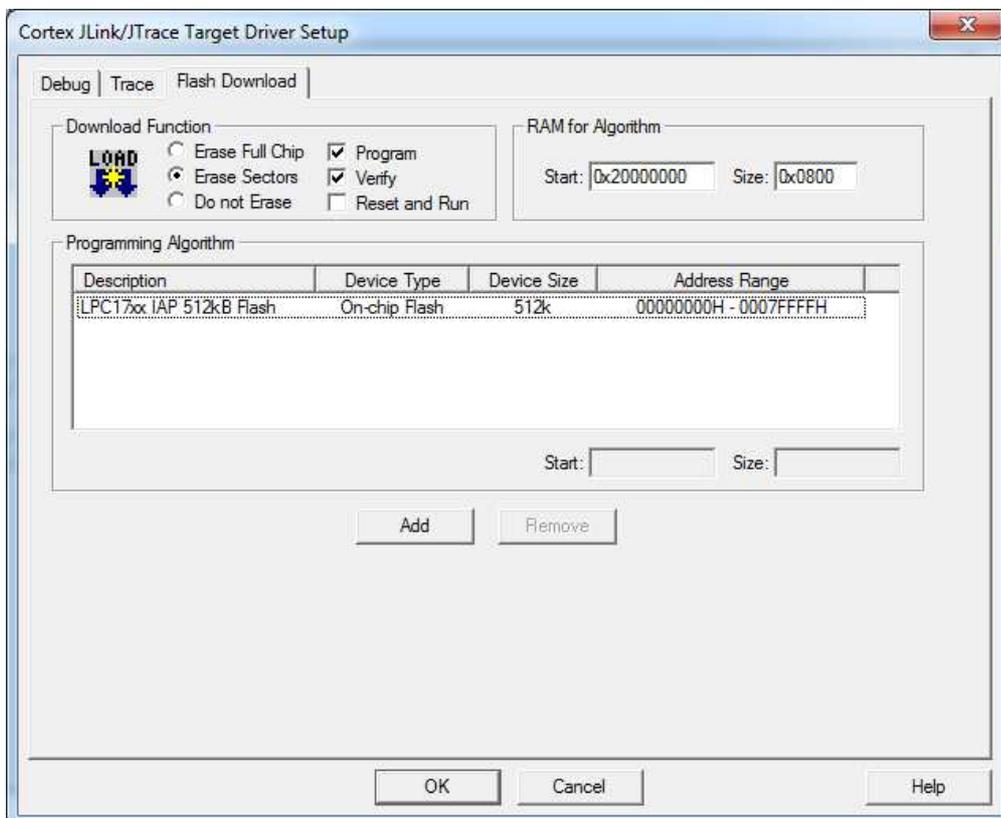


Здесь можно выбрать режим работы интерфейса JTAG/SWD (выпадающий список Port) и интерфейс подключения USB/Ethernet (Поле Interface). Если к эмулятору подключена какая-либо отладочная плата, то в окошке JTAG Device Scan можно

увидеть IDCODE подключенного процессора. Закрываем данное окно настроек и переходим на вкладку Utilities



В выпадающем списке Use Target Driver for Flash Programming выбираем J-LINK/ J-Trace Cortex и нажимаем кнопку Settings

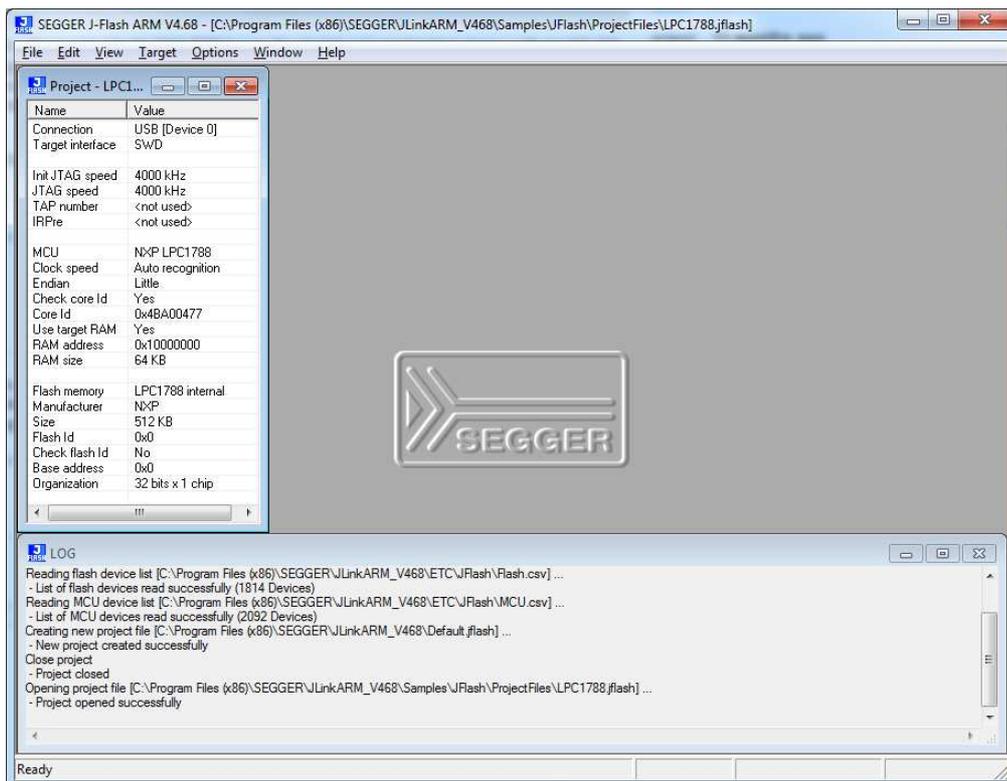


Здесь необходимо нажать кнопку Add и выбрать подключенный к эмулятору процессор (На картинке – NXP LPC1788). Настройка окончена.

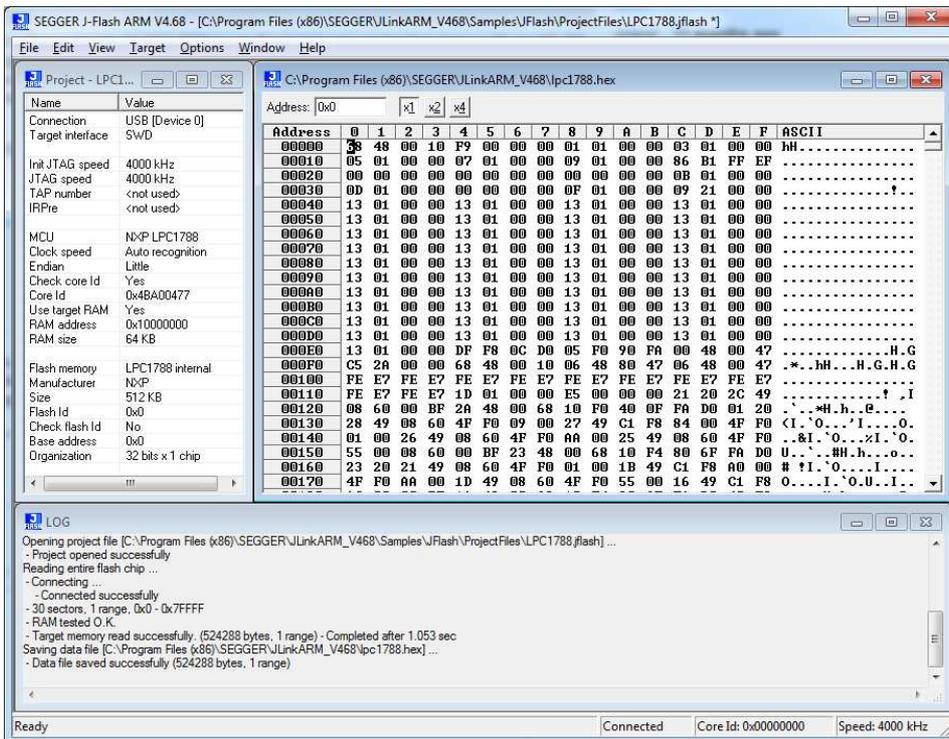
ПРОГРАММИРОВАНИЕ FLASH ПАМЯТИ МИКРОКОНТРОЛЛЕРОВ

УТИЛИТА J-FLASH

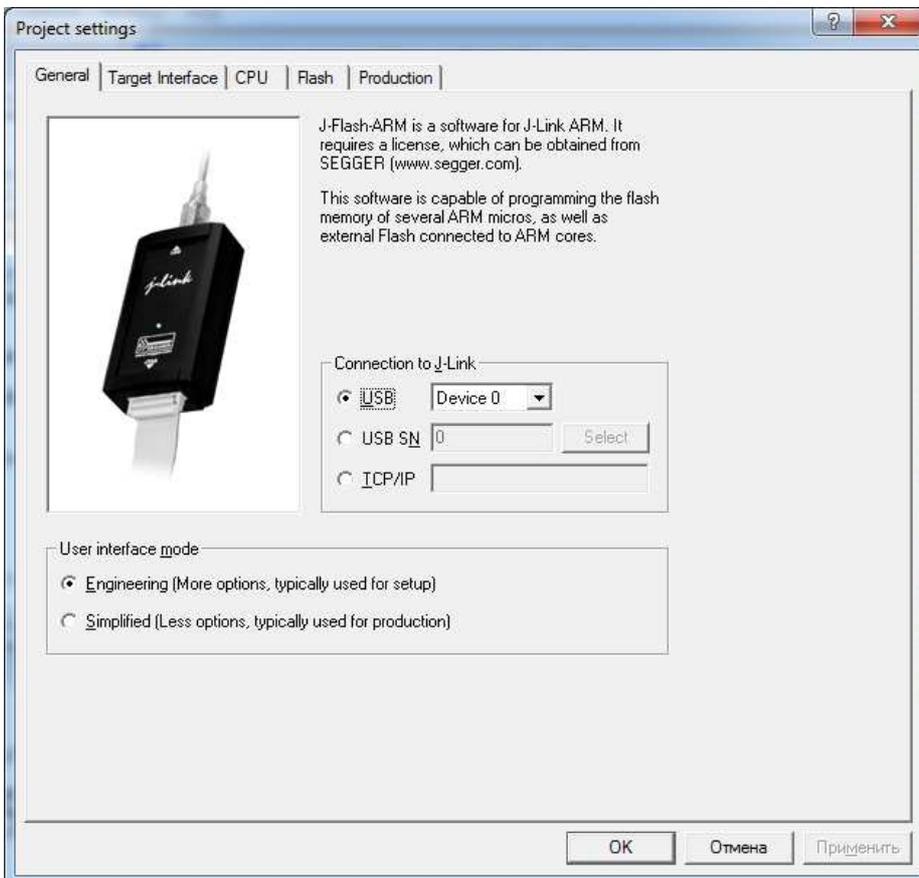
Запускаем утилиту J-Flash (Program Files/Segger/...) Открываем файл проекта соответствующий подключенному процессору File – Open Project



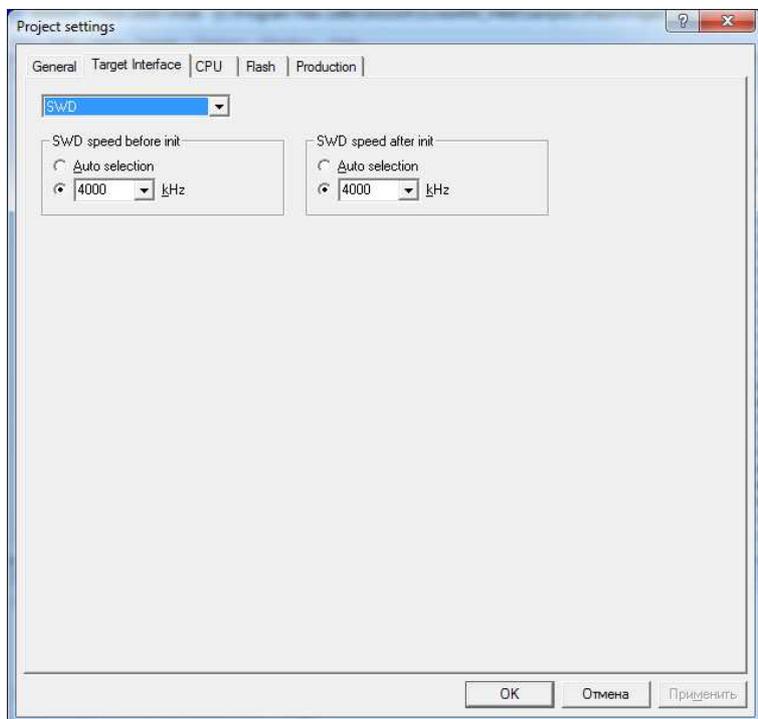
Открываем файл прошивки (*.bin, *.hex, *.sl9, *.mot и т.д.)



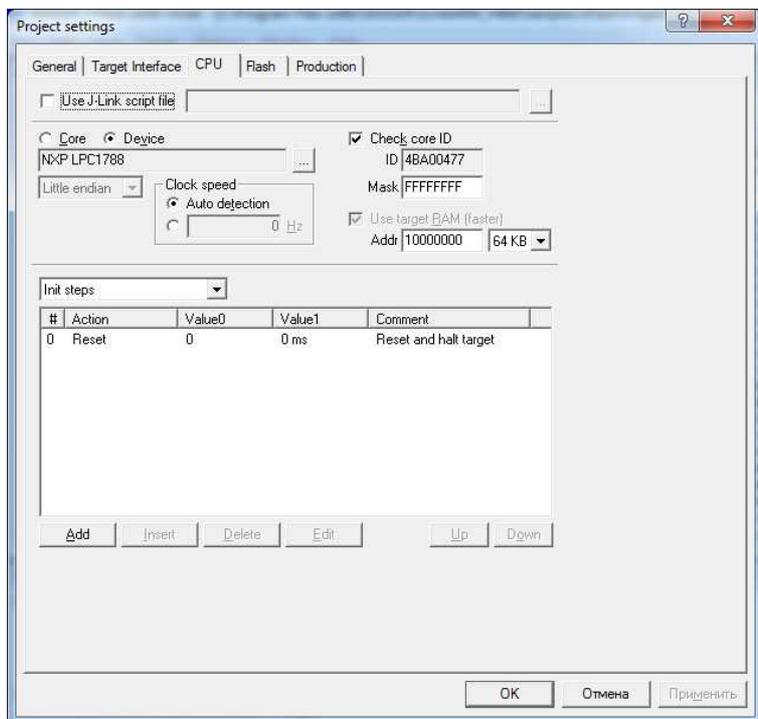
Заходим в Options - Project settings



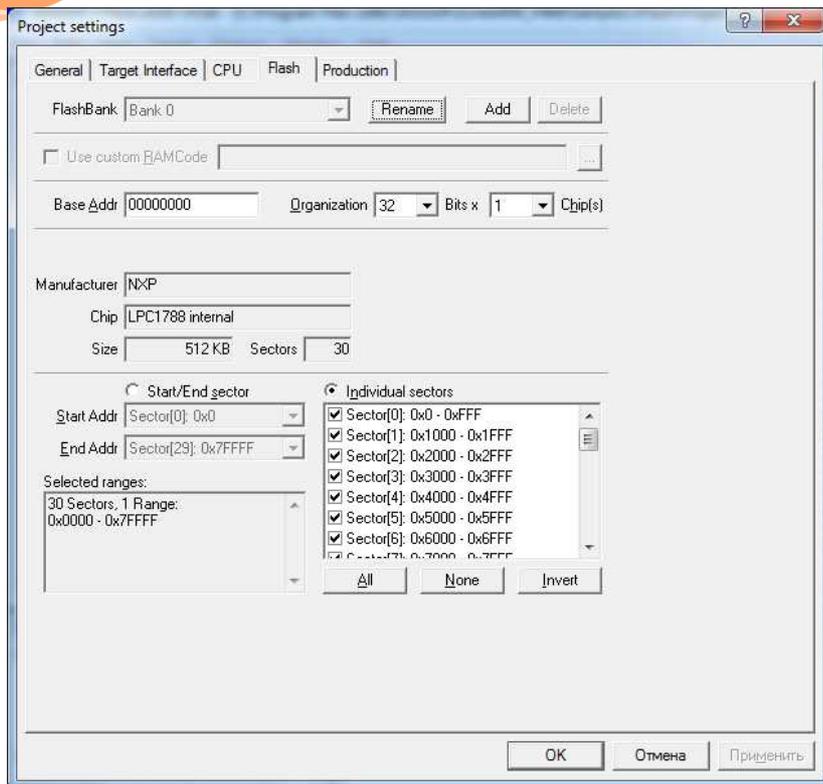
На вкладке General можно настроить интерфейс подключения эмулятора к компьютеру (USB или Ethernet), на вкладке Target Interface можно выбрать интерфейс подключения к микроконтроллеру (JTAG или SWD) и задать скорость интерфейса



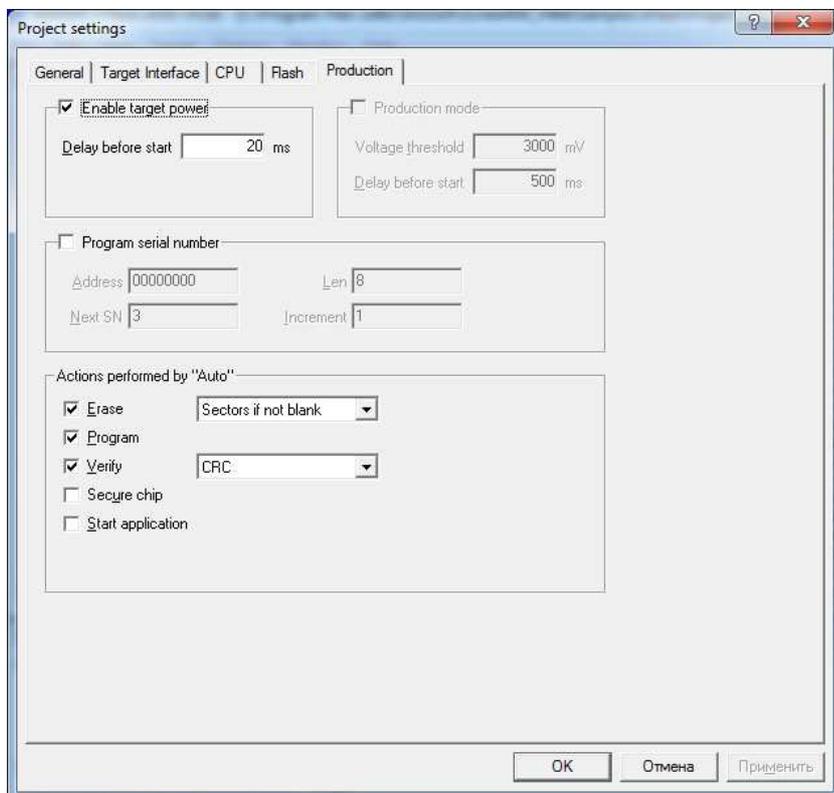
На вкладке CPU можно выбрать процессор, стратегию инициализации процессора и отключить проверку ID core



На вкладке Flash задаются адреса и организация Flash памяти микроконтроллера



На вкладке Production задаются действия которые будут осуществлять по нажатию кнопки F7 (Target – Auto) для программирования микроконтроллера.



Если вам необходимо защитить память микроконтроллера от считывания, не забудьте установить галочку Secure Chip.

После того как все параметры заданы, при нажатии кнопки F7 (Auto) будут произведены действия заданные на вкладке Production (стирание, программирование, верификация, установка защиты).

При необходимости произвести чтение из микроконтроллера (при условии что у него не установлена защита от чтения) можно выбрать Target – Read Back – Entire Chip. После чего сохранить файл с прочитанным содержимым в нужном формате командой File – Save data file as...

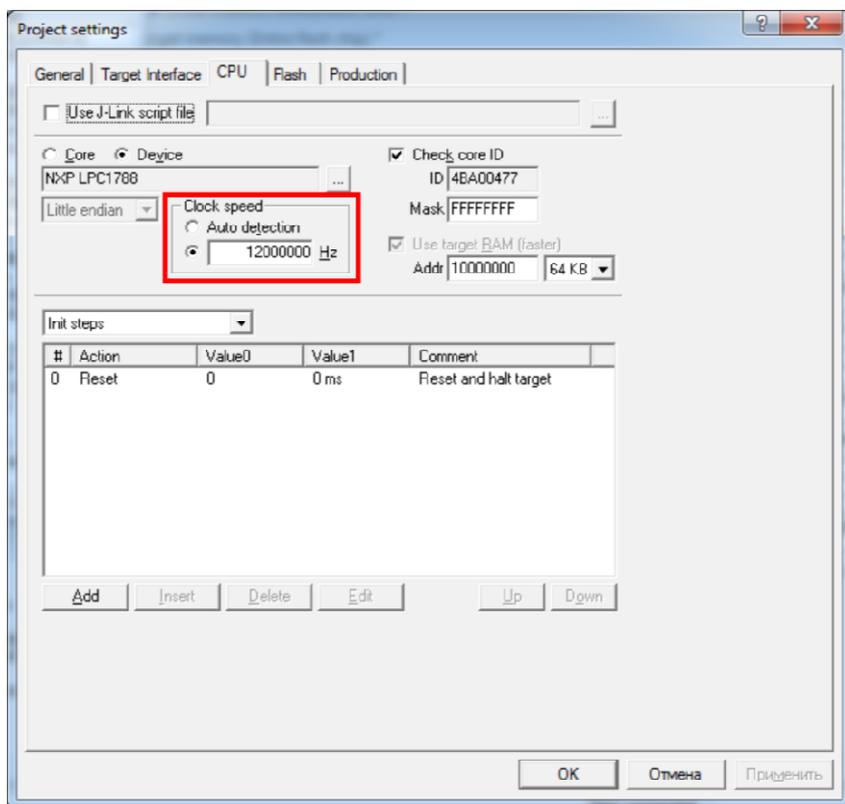
Стереть микроконтроллер можно командой Target – Erase Chip.

АВТОНОМНОЕ ПРОГРАММИРОВАНИЕ

Автономное программирование является полезной функцией для небольших производств. Персоналу необходимо только нажимать кнопку на программаторе и подключать следующую плату для программирования. Программатор имеет возможность записи серийного номера по указанному адресу с автоувеличением.

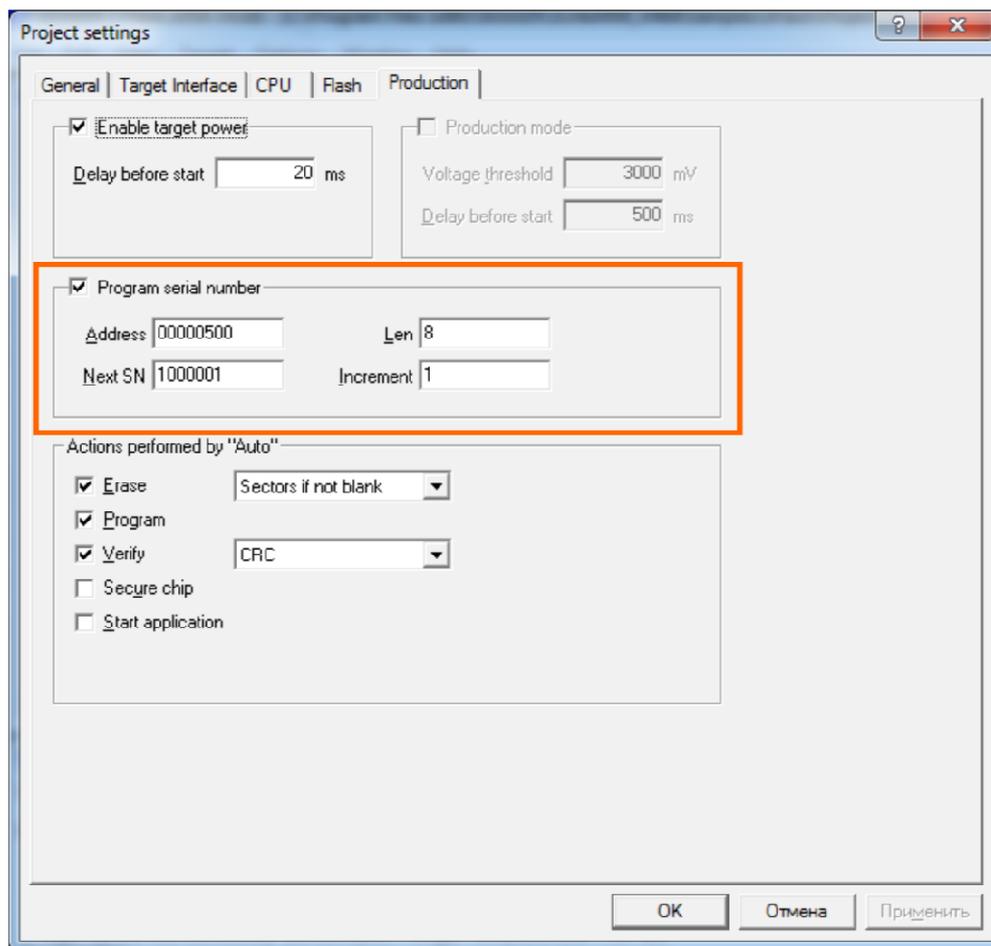
ПОРЯДОК ДЕЙСТВИЙ

1. Подключаем эмулятор по интерфейсу USB к компьютеру.
2. Запускаем утилиты J-Flash
3. Открываем файл проекта соответствующий микроконтроллеру. Очень важно! На вкладке CPU опций проекта необходимо указать частоту кварцевого резонатора микроконтроллера в Герцах (то есть если установлен кварцевый резонатор 12МГц ввести значение 12000000)



Невыполнение этого условия будет приводить к ошибкам ERROR: Auto detection of CPU clock frequency is not supported for this core.

4. Загружаем настроенный проект в эмулятор. File - Download config & data file to Flasher
5. Если имеется необходимость в программировании серийных номеров изделия, предварительно в настройках проекта (вкладка Production) задаем адрес, значение, разрядность и величину инкремента.



В микроконтроллере это будет выглядеть так

По адресу 0x500 в Flash памяти будет находиться число 0x41420f00 которое соответствует номеру 1000001. В следующем запрограммированном микроконтроллере по адресу 0x500 будет находиться число 0x42420f00, которое соответствует номеру 1000002 и т.д.

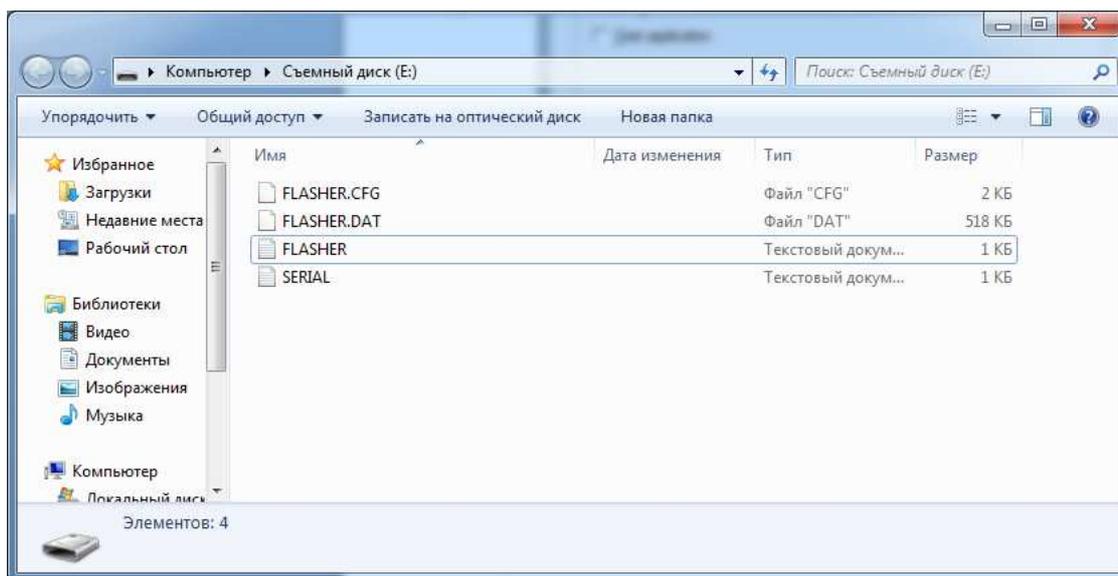
Загружаем значение серийного номера и параметры в эмулятор File - Download serial number file to Flasher.

6. Отключаем эмулятор от компьютера, подключаем только блок питания и программируемую плату с микроконтроллером. Светодиод на эмуляторе быстро мигает зеленым цветом. Нажимаем кнопку START на эмуляторе, светодиод должен медленно промигать 10 раз зеленым светодиодом, а затем опять

замигать быстро. Это означает, что процесс записи и проверки завершился удачно. Если светодиод переливается красным цветом, запись неуспешна.

MSD РЕЖИМ

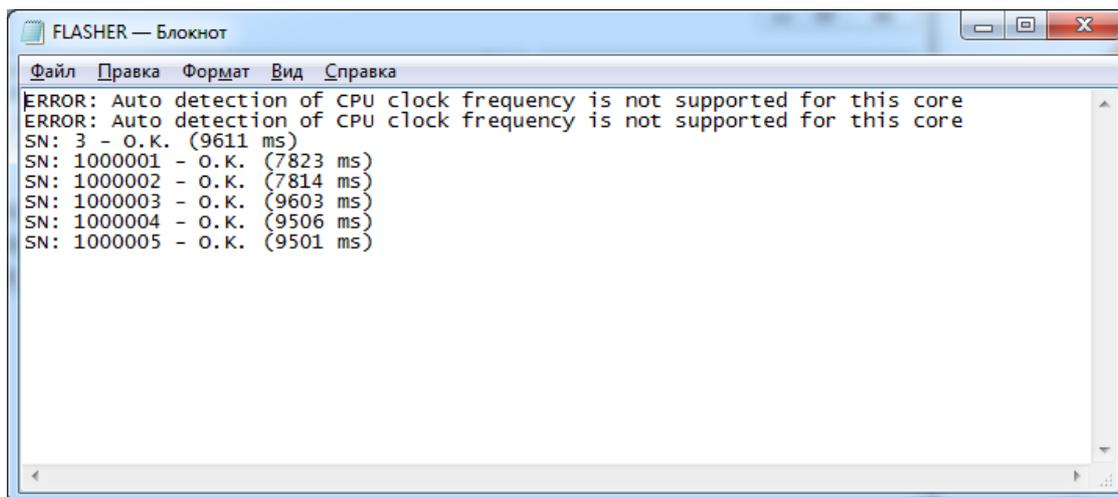
Если нажать кнопку START и не отпуская ее подключить кабель USB к эмулятору, то прибор определится как съемный диск. Обычными средствами Windows на нем можно увидеть следующие файлы (и перезаписать их при необходимости)



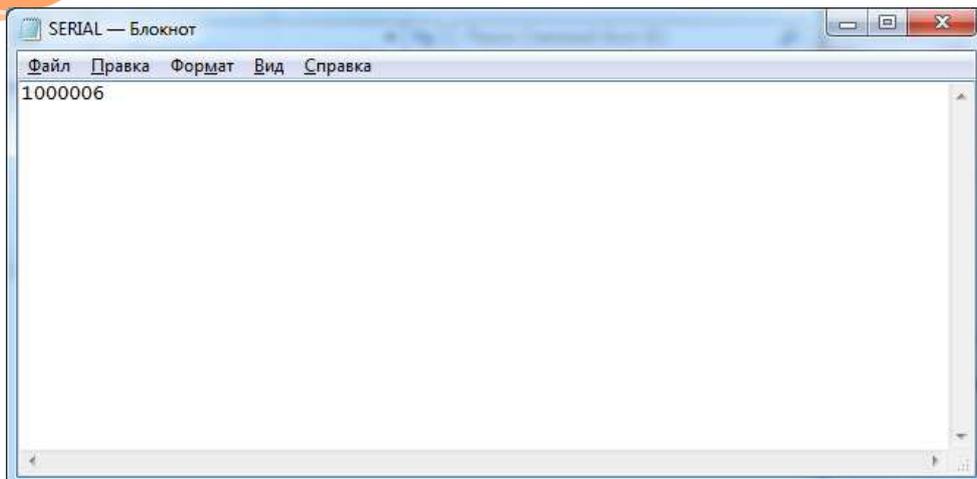
FLASHER.CFG – файл настроек проекта (В программе J-FLASH его можно получить выбрав File – Save Flasher Config file)

FLASHER.DAT – файл прошивки (В программе J-FLASH его можно получить выбрав File – Save Flasher Data file)

FLASHER.TXT – лог файл работы автономного программатора, в нем можно посмотреть результаты работы

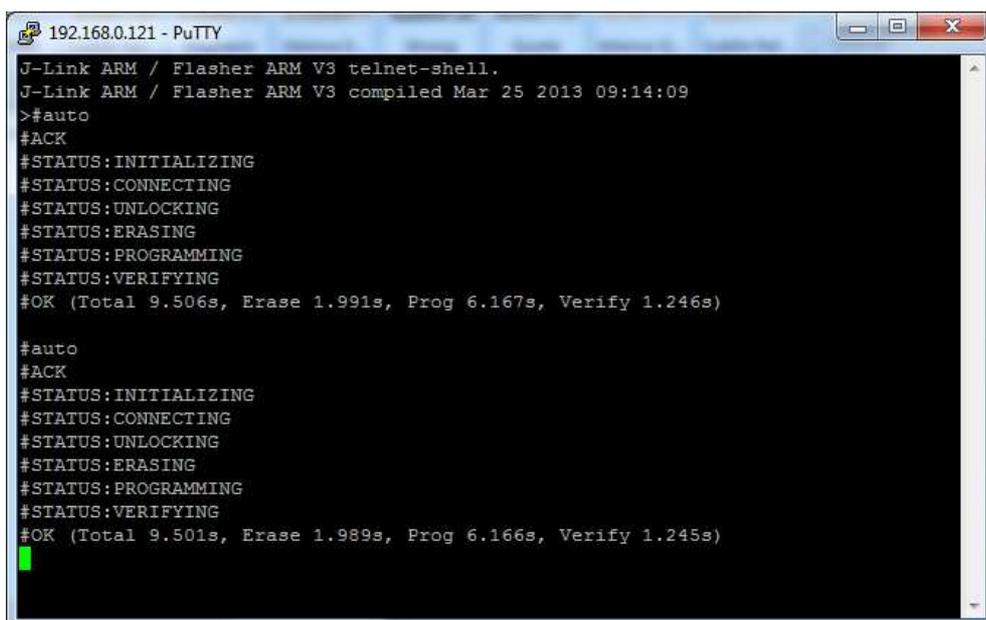


SERIAL.TXT – серийный номер, который будет записан в следующий микроконтроллер



УПРАВЛЕНИЕ ПО TELNET

Если эмулятор подключен к сети Ethernet, то запустив любой Telnet терминал можно подключиться к эмулятору (порт 23) и управлять процессом записи удаленно. Команда `#auto` запускает процесс программирования

A screenshot of a PuTTY terminal window titled "192.168.0.121 - PuTTY". The terminal shows the following output:

```
J-Link ARM / Flasher ARM V3 telnet-shell.  
J-Link ARM / Flasher ARM V3 compiled Mar 25 2013 09:14:09  
>#auto  
#ACK  
#STATUS:INITIALIZING  
#STATUS:CONNECTING  
#STATUS:UNLOCKING  
#STATUS:ERASING  
#STATUS:PROGRAMMING  
#STATUS:VERIFYING  
#OK (Total 9.506s, Erase 1.991s, Prog 6.167s, Verify 1.246s)  
  
#auto  
#ACK  
#STATUS:INITIALIZING  
#STATUS:CONNECTING  
#STATUS:UNLOCKING  
#STATUS:ERASING  
#STATUS:PROGRAMMING  
#STATUS:VERIFYING  
#OK (Total 9.501s, Erase 1.989s, Prog 6.166s, Verify 1.245s)
```

Список команд:

`#auto` - запускает процесс стирания/программирования/верификации (Заданный на вкладке настроек проекта Production) Результат каждой операции отображается в терминале. Статус `#OK` говорит о том что все операции завершились успешно, `#ERRxxx` - произошла ошибка.

`#auto noinfo` - похожа на предыдущую команду, за исключением отсутствия выдачи промежуточных статусов, выдается только `OK` или `ERRxxx`

`#baudrate<baudrate>` задает скорость интерфейса UART используемого для приема команд по интерфейсу UART

#cancel - команда отмены операции

#erase - стирает всю Flash память микроконтроллера

#program - команда записи во Flash память. Стирание и верификация не производятся

#result - команда может быть отправлена в любое время, во время любой операции, возвращает последний результат выполненной предыдущей команды.

#select<filename> выбирает конфигурационный файл и файл с данными для работы

#start - команда переводит все выходы интерфейса JTAG/SWD в Z состояние и сбрасывает микроконтроллер. Это позволит выполнить записанную в микроконтроллер программу.

#verify - команда сравнения содержимого Flash памяти и файла с данными.

КОМАНДЫ УПРАВЛЕНИЯ ЧЕРЕЗ UART

Эмулятор может быть подключен через интерфейс RS-232 к компьютеру или специализированному тестовому оборудованию.

#FCLOSE - команда закрывает файл открытый в эмуляторе командой #FOPEN.

- #FCLOSE
- #ACK
- #OK

При ошибке

- #ACK
- #ERR255:No file opened

#FDELETE<Filename> команда удаляет из памяти эмулятора файл.

- #FDELETE flasher.dat
- #ACK
- #OK

При ошибке

- #ACK
- #ERR255:Failed to delete file

#FOPEN<Filename> - команда открывает указанный файл.

- #FOPEN flasher.dat
- #ACK
- #OK

При ошибке

- #ACK
- #ERR255:A file has already been opened

Minsk 2013

#FREAD<offset>,<NumBytes> - команда используется для чтения из файла со смещением заданным в <offset> числа байт заданного в <NumBytes>

- #FREAD 0,4
- #ACK
- #OK:04:466c6173

При ошибке

- #ACK
- #ERR255:No file opened

#FSIZE - команда возвращает размер файла открытого эмулятором

- #FSIZE
- #ACK
- #OK:10 // открытый файл в эмуляторе имеет размер 16 байт

При ошибке

- #ACK
- #ERR255:No file opened

#FWRITE <Offset>,<NumBytes>:<Data> - команда используется для записи в файл со смещением <offset>, <NumBytes> определяет число байт которые будут отправлены. Количество байт ограничено значением 512, если вы желаете записать 1024 байта, вам придется дважды вызывать команду #FWRITE

- #FWRITE 0,200:<Data>
- #FWRITE 200,200:<Data>

Пример команды передачи слова Hello !

- #FWRITE 0,7:48656C6C6F2021

При ошибке

- #ACK
- #ERR255:No file opened

ОТВЕТЫ ОТ ЭМУЛЯТОРА

#ACK - команда принята и начато выполнение

#NACK - неизвестная команда

#OK - команда выполнена без ошибок

#OK:<NumBytes>:<Data> - если команда #FREAD выполнена передается число реально прочитанных байт

#OK:<Size> - ответ на успешно выполненную команду #FSIZE

#STATUS: - информация о текущем состоянии

#STATUS:READY – эмулятор готов принять следующую команду

#STATUS:CONNECTING – эмулятор инициализирует подключение к программируемому микроконтроллеру

#STATUS:INITIALIZING – эмулятор проводит самотестирование и внутреннюю инициализацию

#STATUS:UNLOCKING – Сектора Flash памяти незаблокированы

#STATUS:ERASING – эмулятор стирает Flash память в подключенном микроконтроллере

#STATUS:PROGRAMMING – эмулятор программирует Flash память в подключенном микроконтроллере

#STATUS:VERIFYING – эмулятор производит проверку на совпадения содержимого Flash памяти и файла

Ошибки

#ERRxxx – если команда завершена неудачно, эмулятор сообщает об ошибке.

#ERR007 – эмулятор принял команду #CANCEL и текущая операция отменена

#ERR008 – эмулятор занят выполнением предыдущей команды

#ERR255 – произошла неизвестная ошибка

НАЗНАЧЕНИЕ КОНТАКТОВ РАЗЪЕМА JTAG

В таблице приведено назначение контактов разъема в режиме JTAG

Номер	Сигнал	Тип	Описание
1	VTREF	Вход	Вход напряжения питания подключенного микроконтроллера.
2	Vsupply	Не используется	Не используется
3	nTRST	Выход	Сброс JTAG
5	TDI	Выход	Выход данных JTAG
7	TMS	Выход	Выход выбора режима JTAG
9	TCK	Выход	Тактирование JTAG
11	RTCK	Вход	Вход тактирования от подключенного микроконтроллера. При отсутствии у подключенного

			микроконтроллера данного сигнала замкните этот вход с сигналом TCK
13	TDO	Вход	Вход данных JTAG
15	RESET	Вход/Выход	Сигнал сброса подключенного микроконтроллера
17	DBGREQ	Не используется	
19	5V TARGET	Выход	Может быть использован для питания подключенного микроконтроллера.
4, 6, 8, 10, 12, 14, 16, 18, 20	GND	Земля	

НАЗНАЧЕНИЯ СИГНАЛОВ РАЗЪЕМА В РЕЖИМЕ SWD

Номер	Сигнал	Тип	Описание
1	VTREF	Вход	Вход напряжения питания подключенного микроконтроллера.
7	SWDIO	Вход/Выход	Двухнаправленная линия данных
9	SWCLK	Выход	Тактирование
13	SWO	Выход	Выход трассировочного порта (подключение необязательно)
15	RESET	Вход/Выход	Сигнал сброса подключенного микроконтроллера
19	5V TARGET	Выход	Может быть использован для питания подключенного микроконтроллера.
4, 6, 8, 10, 12, 14, 16, 18, 20	GND	Земля	

УПРАВЛЕНИЕ ПИТАНИЕМ ПОДКЛЮЧЕННОГО МИКРОКОНТРОЛЛЕРА

С 19 вывода эмулятора можно получить напряжение 5 Вольт, ток до 0,3А, например, для питания подключенной отладочной платы. Запустите утилиту J-Link.exe и с помощью следующих команд вы можете управлять питанием:

power on - включить питание включено только на данную сессию работы эмулятора

power off - выключить питание

power on perm - включить питание постоянно (при подключении эмулятора к компьютеру питание 19 выводе эмуляторе появляется автоматически)

power off perm - питание постоянно отключено

ПРОВЕРКА СЕРИЙНОГО НОМЕРА ЭМУЛЯТОРА

Подключите эмулятор к компьютеру по USB интерфейсу и запустите утилиту J-Link.exe

УСТАНОВКА IP АДРЕСА ЭМУЛЯТОРА

Подключите эмулятор к компьютеру по USB интерфейсу и запустите утилиту J-Link.exe

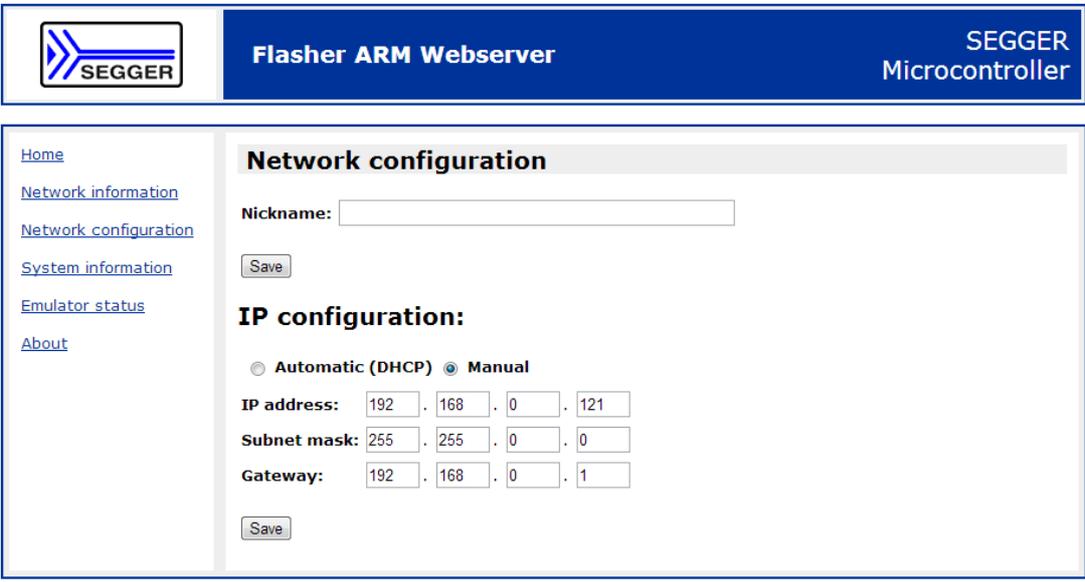
Командой **ipaddr 192.168.0.121** можно назначить IP адрес (в данном случае 192.168.0.121)

Командой **gwaddr** можно назначить шлюз.

Команда **dnsaddr** назначает DNS сервер.

ВСТРОЕННЫЙ WEB СЕРВЕР

Эмулятор имеет встроенный web сервер, зайти на который можно набрав в строке браузера IP адрес эмулятора



The screenshot displays the 'Flasher ARM Webserver' interface for a SEGGER Microcontroller. The page features a navigation menu on the left with links for Home, Network information, Network configuration, System information, Emulator status, and About. The main content area is titled 'Network configuration' and includes a 'Nickname' input field with a 'Save' button. Below this, the 'IP configuration' section offers two options: 'Automatic (DHCP)' and 'Manual' (which is selected). The 'Manual' configuration includes input fields for 'IP address' (192, 168, 0, 121), 'Subnet mask' (255, 255, 0, 0), and 'Gateway' (192, 168, 0, 1), each with a 'Save' button.

Информация и настройка сети

Flasher ARM Webservice
SEGGER Microcontroller

[Home](#)
[Network information](#)
[Network configuration](#)
[System information](#)
[Emulator status](#)
[About](#)

Network information

Configuration:

Configuration type: User assigned
IP address: 192.168.0.121 / 16
MAC address: 00:22:C7:04:07:5E
Nickname:

Memory usage:

Small buffers (256 byte): 8 / 8
 Big buffers (592 byte): 3 / 4

Network connections:

Connection info

List of TCP connections

Socket	Local	Peer	State	MTU/MSS	Retrans. delay	Idle time	Local window	Peer window
1	Any:19020	---	Listen	60/0	20190	4043320	0/0	0
2	Any:23	---	Listen	60/0	20190	4043290	0/0	0
3	Any:80	---	Listen	60/0	20190	4040430	0/0	0
4	192.168.0.121:23	192.168.0.100:18171	Established	576/536	2290	5430	1070/1072	65283
5	192.168.0.121:80	192.168.0.100:18306	Established	576/536	2290	0	1072/1072	65392
44	192.168.0.121:80	192.168.0.100:18348	FIN wait 2	576/536	2290	0	1072/1072	64901
45	192.168.0.121:80	192.168.0.100:18349	Established	576/536	20190	0	641/1072	65392

Сетевая информация

Flasher ARM Webservice
SEGGER Microcontroller

[Home](#)
[Network information](#)
[Network configuration](#)
[System information](#)
[Emulator status](#)
[About](#)

System information

embOS statistics:

System info

Number of tasks : 8
 System time : 4069324
 System stack (size@base) : 512@0x200040

List of tasks

Id	Priority	Task names	Context switches	Task Stack
0x2068BC	11	IP Stack	0	0/0@0x0
0x207AA8	9	Webserver	0	0/0@0x0
0x207AF4	8	Webserver Child	0	0/0@0x0
0x207AD0	8	Webserver Child	0	0/0@0x0
0x206F68	5	Terminal Telnet	0	0/0@0x0
0x207428	4	Terminal UART	0	0/0@0x0
0x204F38	3	J-Link Server	0	0/0@0x0
0x20A550	1	MainTask	0	0/0@0x0

The table above shows some real time information of the RTOS kernel used. Information about your application can easily be displayed in the same way.

Используемые ресурсы

Minsk 2013

24

ОБНОВЛЕНИЕ ВНУТРЕННЕГО FIRMWARE ЭМУЛЯТОРА

Обновление микропрограммы эмулятора происходит автоматически при установке нового программного обеспечения. Рекомендуется всегда обновлять внутреннюю микропрограмму эмулятора, поскольку в новых релизах добавляются новые возможности и исправляются найденные ошибки.

ИСТОРИЯ ИСПРАВЛЕНИЯ ДОКУМЕНТА

14/04/2013 – Начальная ревизия документа 1.0