Оглавление

[0. Необходимые программы. 1](#_Toc473893781)

[1. Программатор для работы с RTL. 1](#_Toc473893782)

[2. подключение ST-link адаптера (программатора) 1](#_Toc473893783)

[3. Переделка ST-link в J-link 5](#_Toc473893784)

[**4.** Установка J-flash. 8](#_Toc473893785)

[**5.**Снятие ограничений. 9](#_Toc473893786)

[**6.** Настройка J-Flash под контроллер. 12](#_Toc473893787)

[**7**. Подключение адаптера к контролеру. 14](#_Toc473893788)

[**8**. Проверка связи. 16](#_Toc473893789)

[9. RTL00 flasher. 20](#_Toc473893790)

[9.1 Базовая работа с прграммой. 26](#_Toc473893791)

[10. Память в контроллере RTL8710 28](#_Toc473893792)

[11. Немного о том, что такое IDE Eclipse. 31](#_Toc473893793)

[12. Установка IDE Eclipse и всего необходимого (под windows). 31](#_Toc473893794)

[12.1. установка IDE Eclipse 31](#_Toc473893795)

[12.2 Установка minGW 31](#_Toc473893796)

# 0. Необходимые программы.

Папка/архив со всеми необходимыми дистрибутивами и программами упоминаемые в описании. <https://yadi.sk/d/8bS4SDgC3CJQf4> Папка по мере написания инструкций дополняется.

# 1. Программатор для работы с RTL.

*Программирование любого чипа на базе архитектуры ARM осуществляется по J-tag или SWD протоколу/интерфейсу. Он не работает через UART (COM) порт. Поэтому шить как ардуину через usb-кабель не получится. Нужен програматор. Это может быть оригинальный SEGGER, оригинальный ST-link, или китайский клон ST-link.*

*Скачайте архив ( это папка на яндекс диске) с дистрибутивами.*

*А. Если есть фирменный адаптера от SEGGER то пропустить пункт 1,2,4*

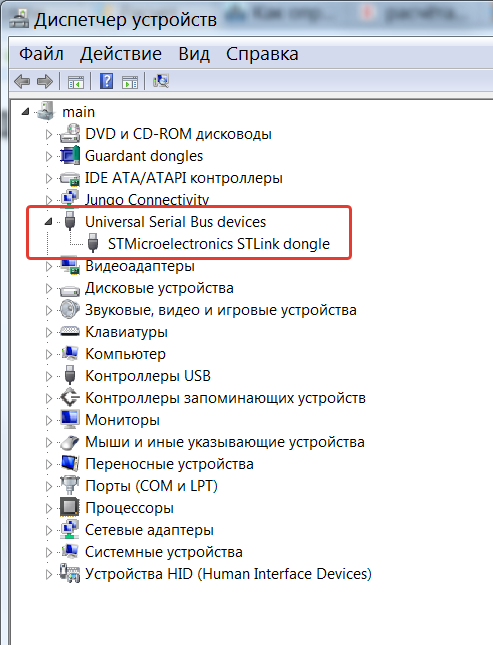
*Б. Если есть китайский или оригинальный ST-Link адаптер то выполниять по порядку.*

*В. Если нет адаптера то купить его.*

# 2. подключение ST-link адаптера (программатора)

После подключения Stlink адаптера необходимо установить под него драйвера. Есть в архиве в папке “Stlink drivers” или списать здесь: <http://www.st.com/content/st_com/en/products/embedded-software/development-tool-software/stsw-link009.html>

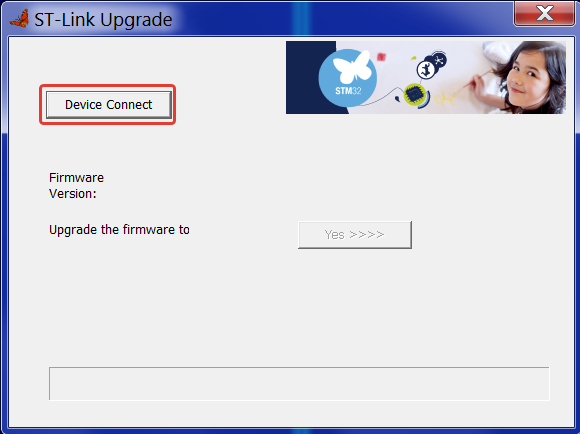
После установки драйвера в диспетчере устройст он выглядит так:

**

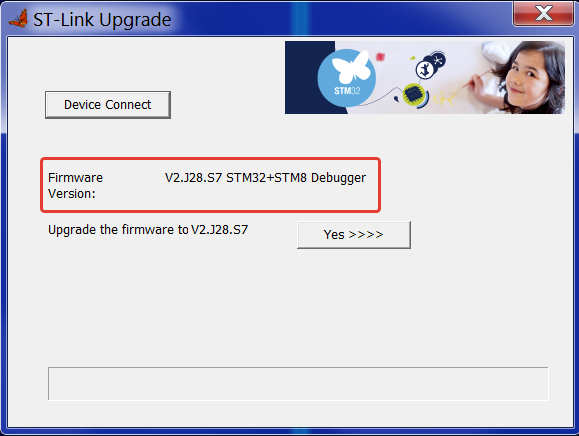
Для проверки, что установлен коррктно запустите ST-linkUpgrade.exe из папки «ST link upgrade software” или возьмите здесь:

<http://www.st.com/en/embedded-software/stsw-link007.html>

После запуска утилита выглядит так, нажмите **Device Connect**

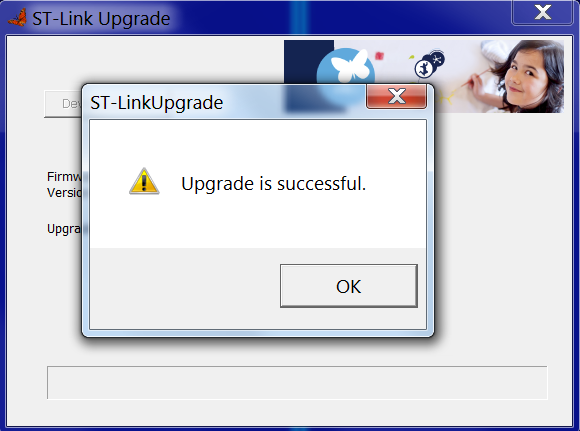
**

Утилита свяжется с адаптером. Окно должно выглядеть так.

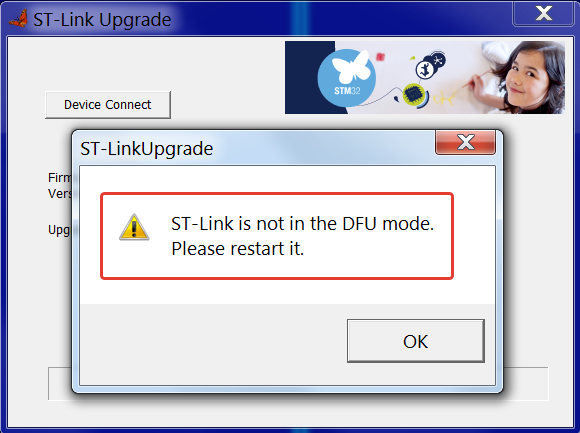


Будет показана установленная версия прошивки адаптера и возможность апгрейда. (Не знаю связывается ли утилита с сервером когда предлагает апгрейд прошивки или он есть в ней)

Можете нажать кнопку YES>>>> и тогда произойдёт обновление. По окончанию оно выглядеть будет так. (занимает 10-15 секунд)



Если выдаст такую ошибку, то просто переподключите адаптер.



# 3. Переделка ST-link в J-link

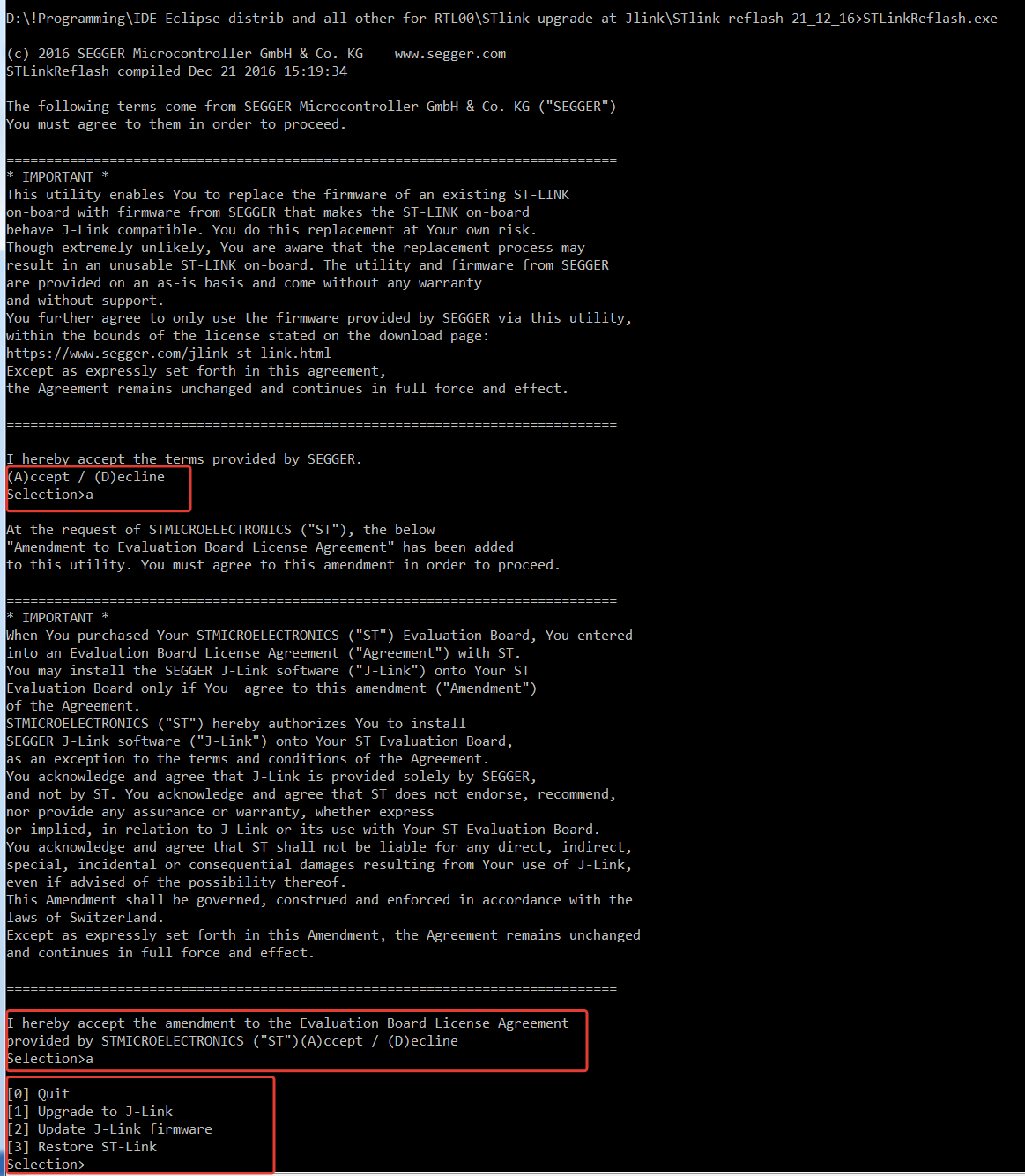
**Переподключите адаптер! Не забудьте.**

Перепрошить ST-link адаптер в J-link для этого запустить программу STlinkreflash.exe

*Здесь у меня была проблема не все версии работают. В папке «****Stlink upgrade at Jlink****» есть две папки с разными версиями программы апдейта та которая работает у меня (более старая) помечена как “****working for me****”. У вас может заработать какая то другая. Или вообще не заработать. Проблема связана китайский клоном ST-Link. Тут как повезёт. С родным ST-link должна работать всегда. (У меня клон)*

Последнюю версию всегда можно взять здесь: <https://www.segger.com/downloads/jlink>

Запустите STlinkreflash.exe и дважды согласитесь с предложением (латинская буква A)

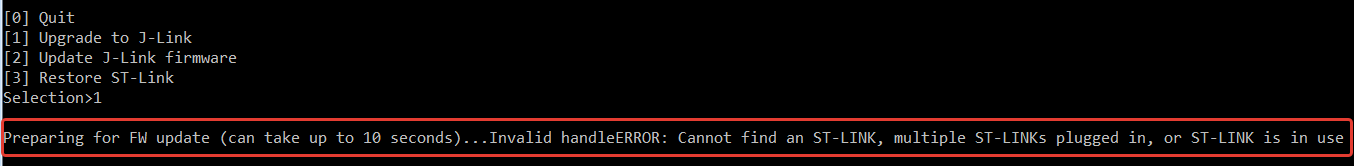


У вас отобразится меню с возможными вариантами.

1. Quit
2. Upgrade to J-link ( Превращает ST-Link в J-link)
3. Upgrade J-link firmware (Обновляет прошивку в J-link)
4. Restore ST-link (Восстанавливает ST-link из J-link)

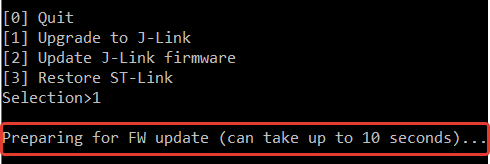
Выбираете пункт **[1] Upgrade to J-link**.

Если после выполнения пункта вы видете что то вроде такого:



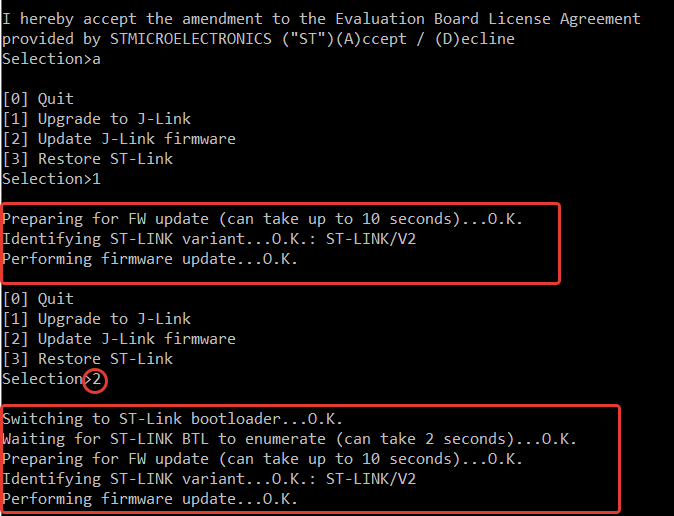
То у вас неподходящая версия STlinkreflash.exe или ваш адаптер не может быть переделан. (Не забудьте переподключить адаптер после работы с ST-link update. Из за этого тоже может не определяться)

Если зависло здесь ( у меня занимает около секунды):



То просто перезапустите утилиту.

Нормальное выполнение программы апдейта выглядит так.



Можно сразу обновить програмное обеспечение, после первого пункта выбрав пункт 2.

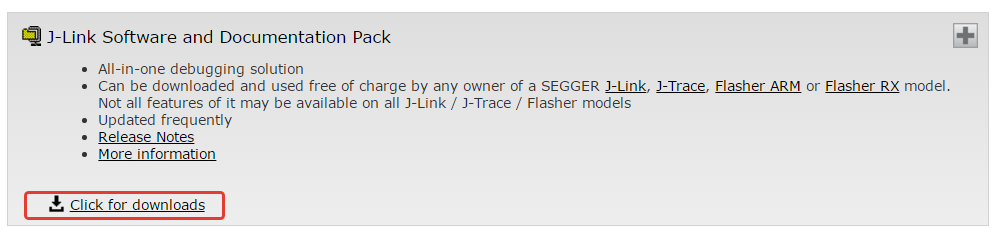
Выйдите из утилиты.

Stlink пропадёт в дереве устройств.Но драйверов под J-link пока не стоит просто появится неизвестное устройство. (У меня цвет светодиода на адаптере изменился с красного на зелёный но это опционально в зависимости что у вас за адаптер)

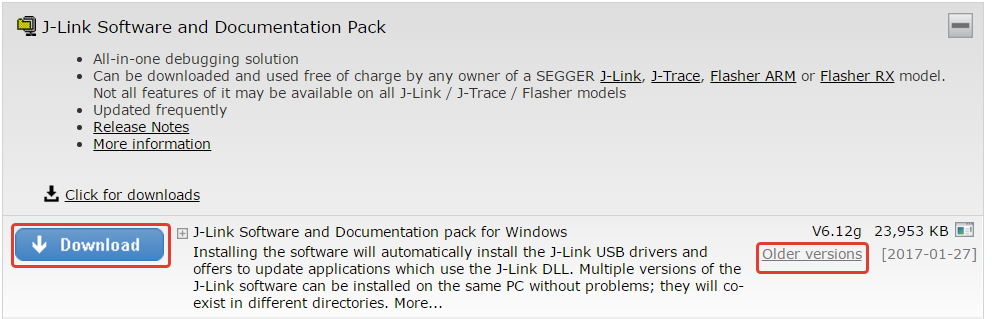
# **4.** Установка J-flash.

Взять версию 6.12g можно в папке «Jlink SDK win 612g +keygen» или взять любую версию здесь:

<https://www.segger.com/downloads/jlink> (J-Link Software and Documentation Pack) Для скачивания нажмите click for download

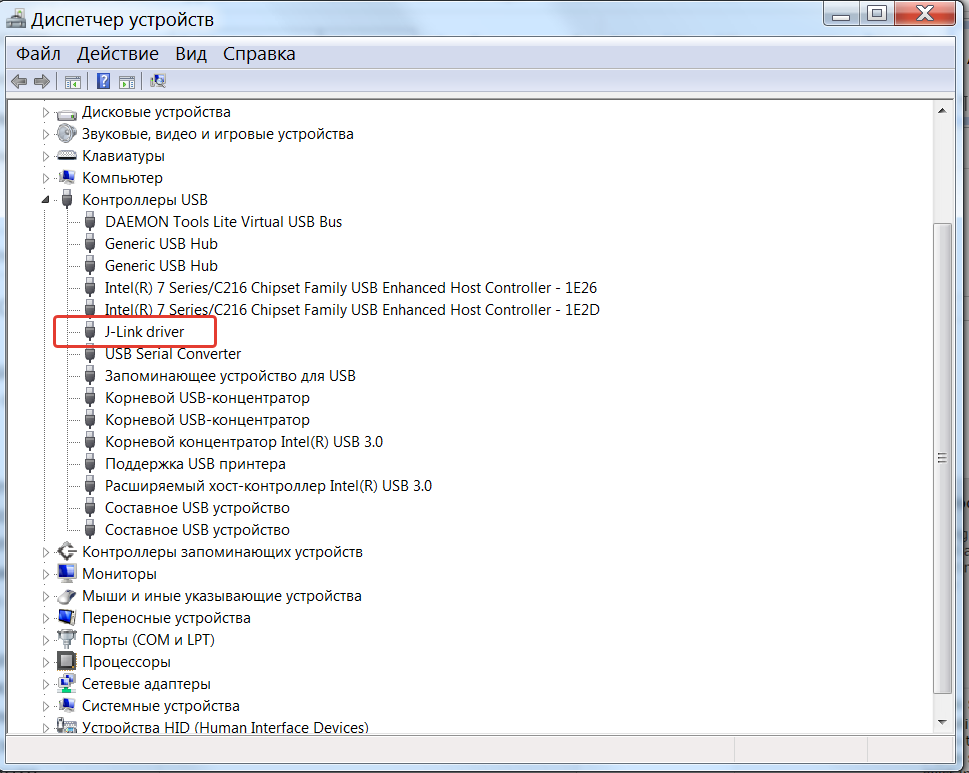


и скачайте последнюю версию нажав кнопку downlod или более старую выбрав older versions



Установите флешер в папку без кирилических символов в пути. (на всякий случай).

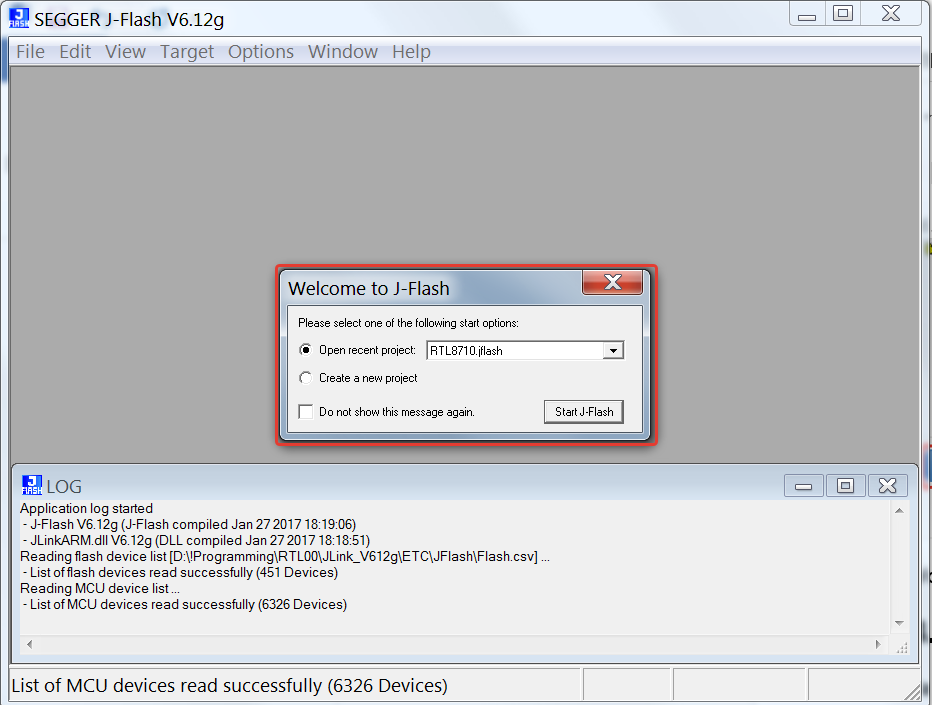
После установки драйвера должны встать автоматически и в диспетчере устройств адаптер должен выглядеть так: (не забудьте переподключить его)



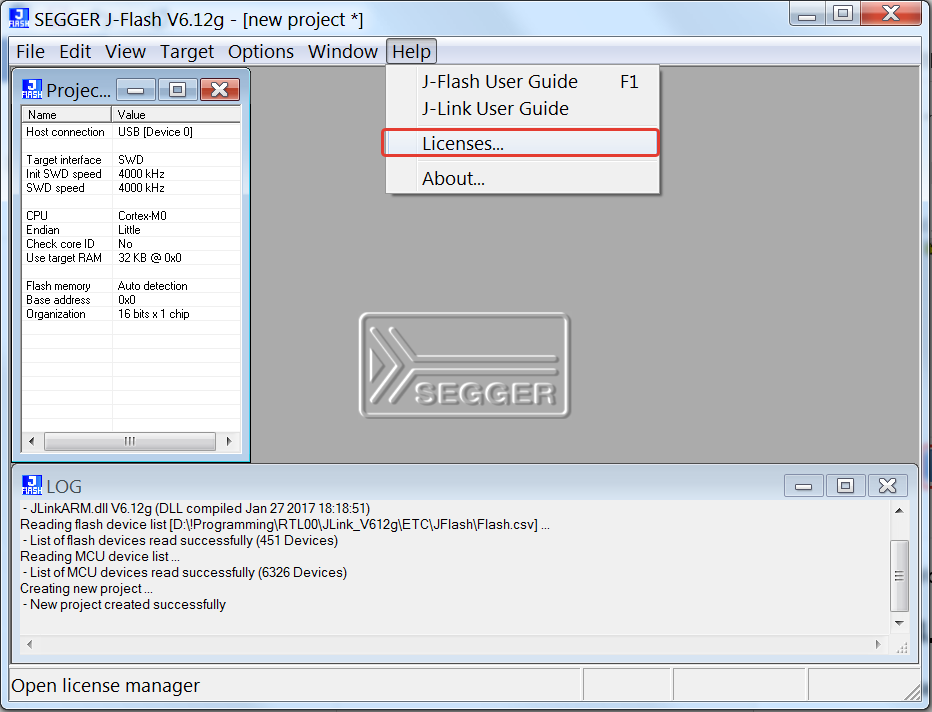
# **5.**Снятие ограничений.

Запустите Jflash.

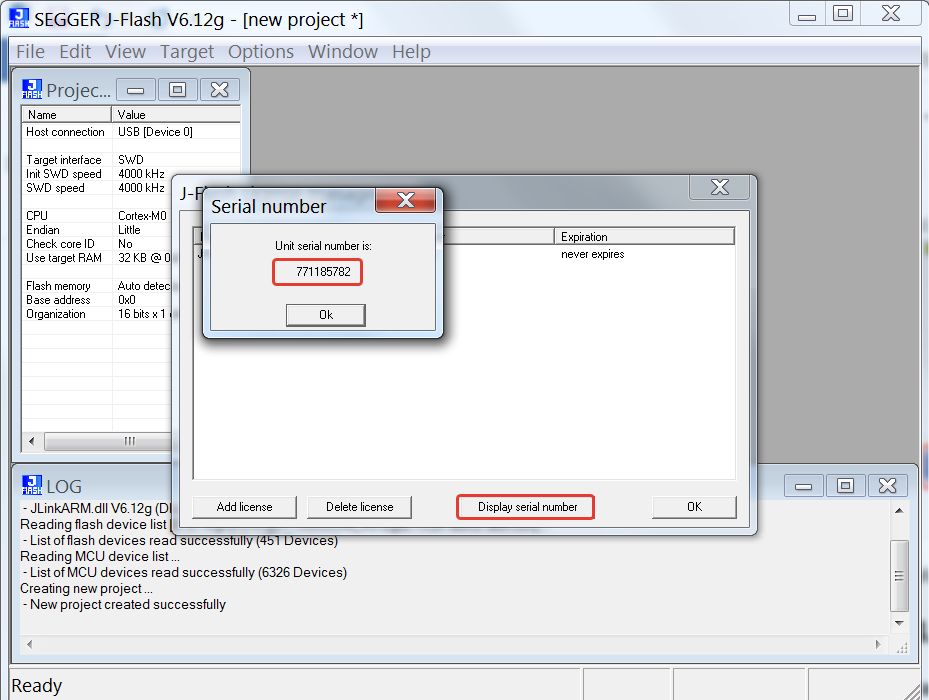
Закройте окошко открытия проекта.



Идите в меню **Help - Licenses**



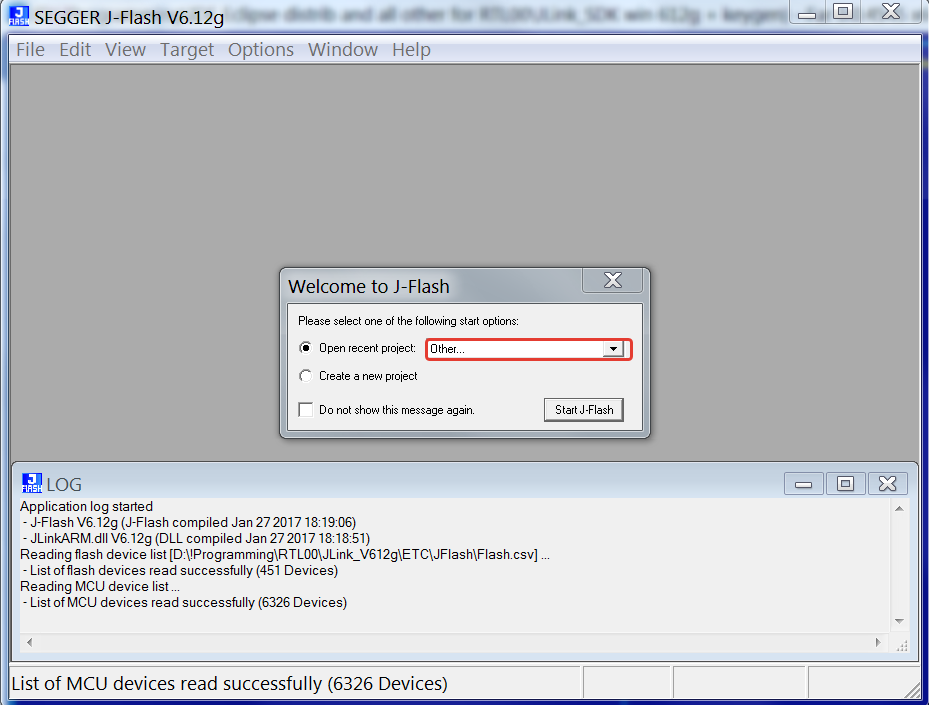
И посмотрите свой серийный номер.



И для снятия ограничений ищите в яндексе по слову rdikeygen. Там всё просто. Лицензию добавите через кнопку Add license.

# **6.** Настройка J-Flash под контроллер.

Перезапустите J-Flash и в окошке выбора проекта. Выберите other.

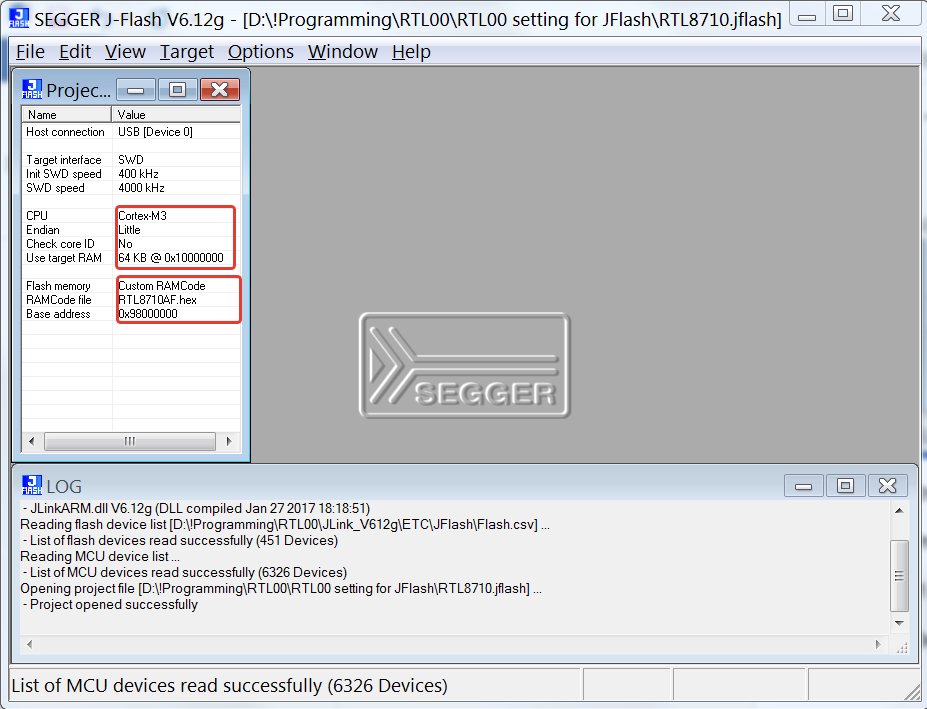


Откроется проводник и выбирите файл RTL8710.jflash есть в папке «RTL00 setting for Jflash»

Это настройки под чип RTL8710.

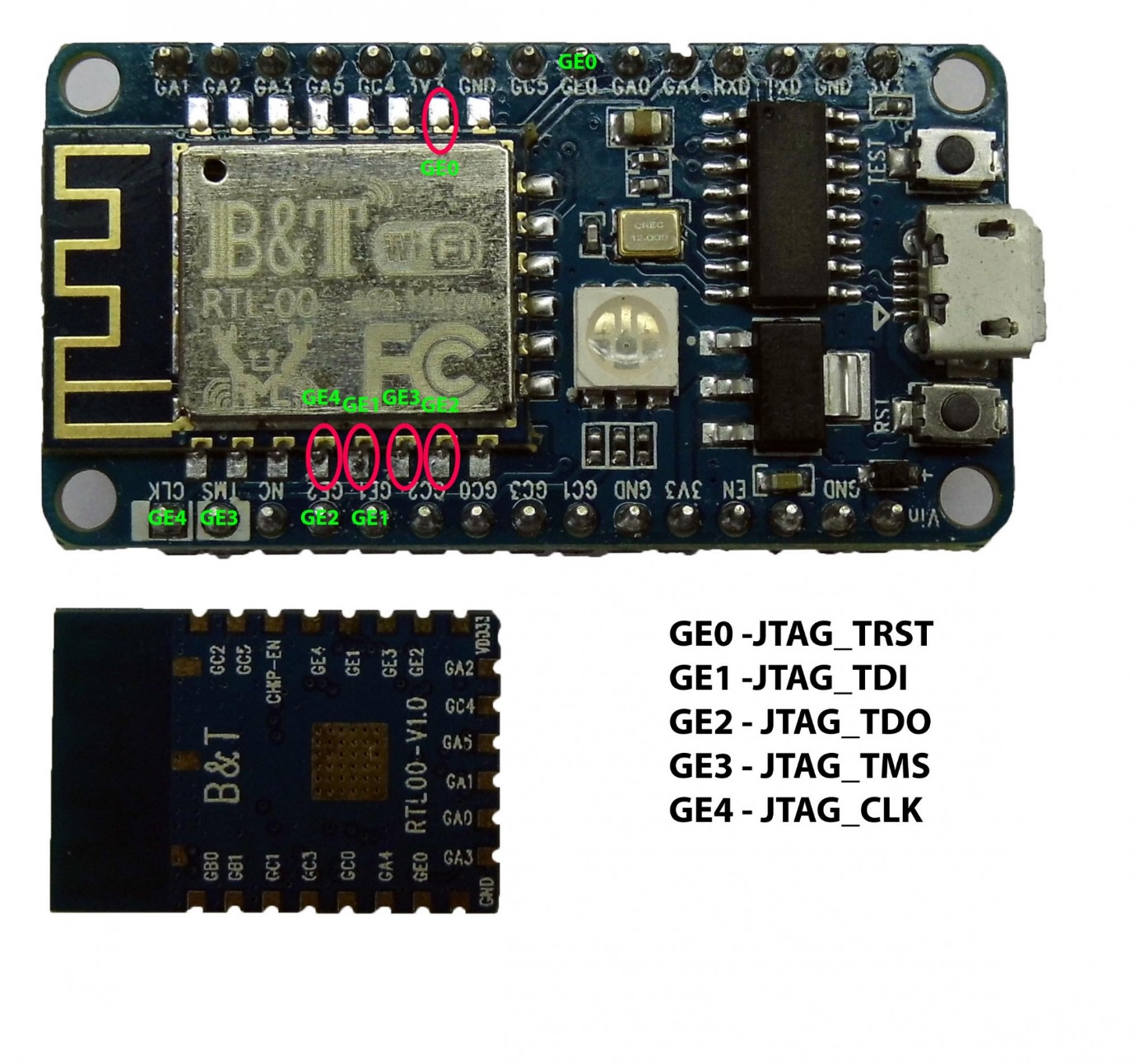
И нажмите кнопку Start J-Flash.

Окно флешера станет выглядеть так.



# **7**. Подключение адаптера к контролеру.

Для проверки связи необходимо подключить адаптер к контролеру. Скорее всего это будет какая то из этих двух плат.



Подключаются они к стандартгому разъёму Jtag 20 pin вот так:



Для работы достаточно двух линий JTAG\_CLK (SWCLK) и JTAG\_TMS(SWDIO) В данном случае J-link работает по SWD протоколу.

Дополнительное питание в случае платы с USB разъёмом никакое не нужно. Просто подключите его к USB. Для маленькой платы подайте на неё 3-3,6V. Можно ли запитать от 5 вольт не проверял, это на свой страх и риск.

*Про USB.*

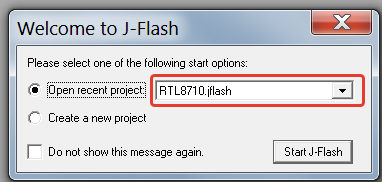
*USB разъём НЕ СЛУЖИТ для прошивки, общение через AT команды –да. Использование как обычного UART порта –да. Прошивка НИКОГДА. Особенность ARM. (Вроде так)*

*О нем, о его драйвере, и как работать с ним будет ниже.*

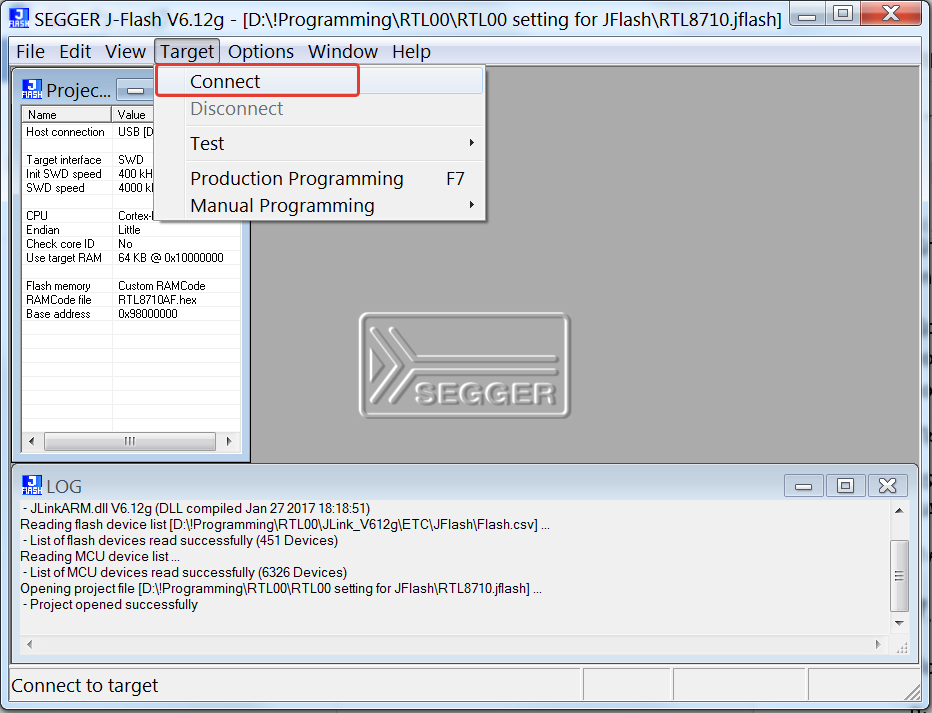
# **8**. Проверка связи.

После подключения контролера к адаптеру и подания на него питания. Запускаем J-flash.

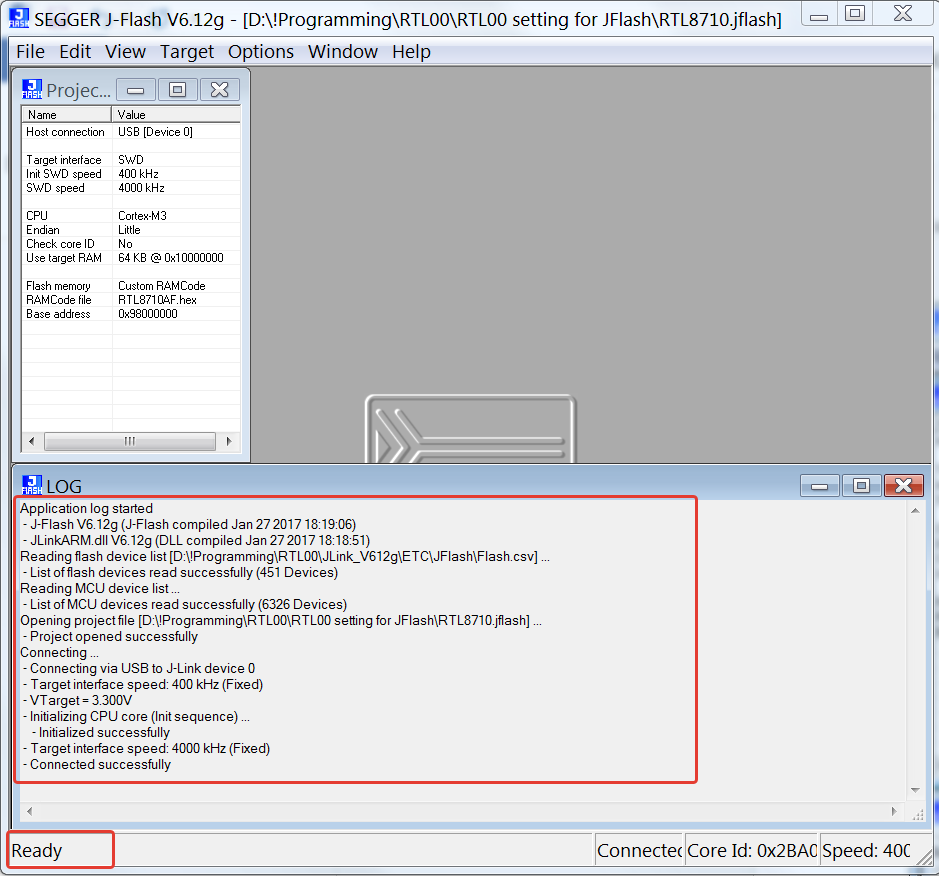
Выбираем в выпдающем меню проект RTL8710 (он уже будет здесь после действий описанных в пункте 5) и жмём кнопку Start Jflash.



Выбираем Target-Connect

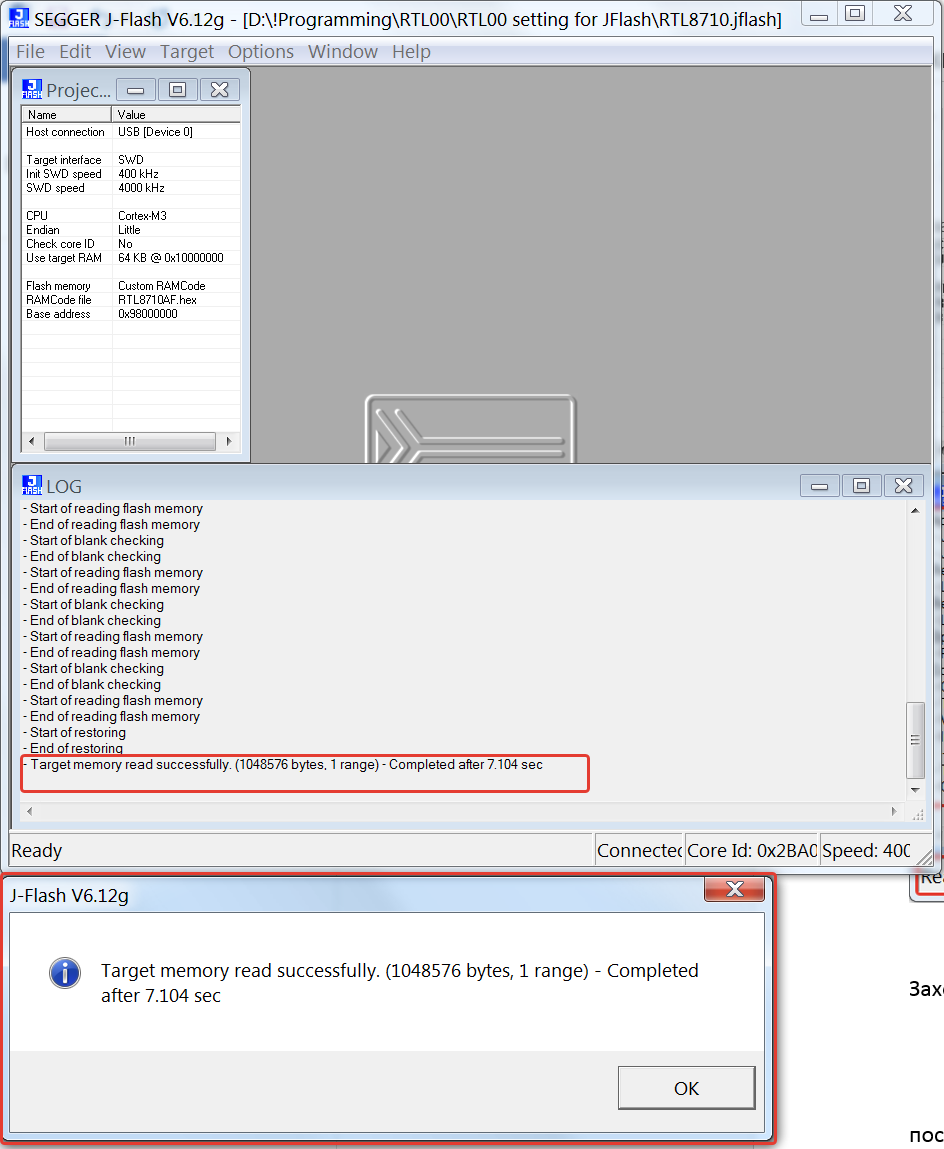


Должно произойти соединение:



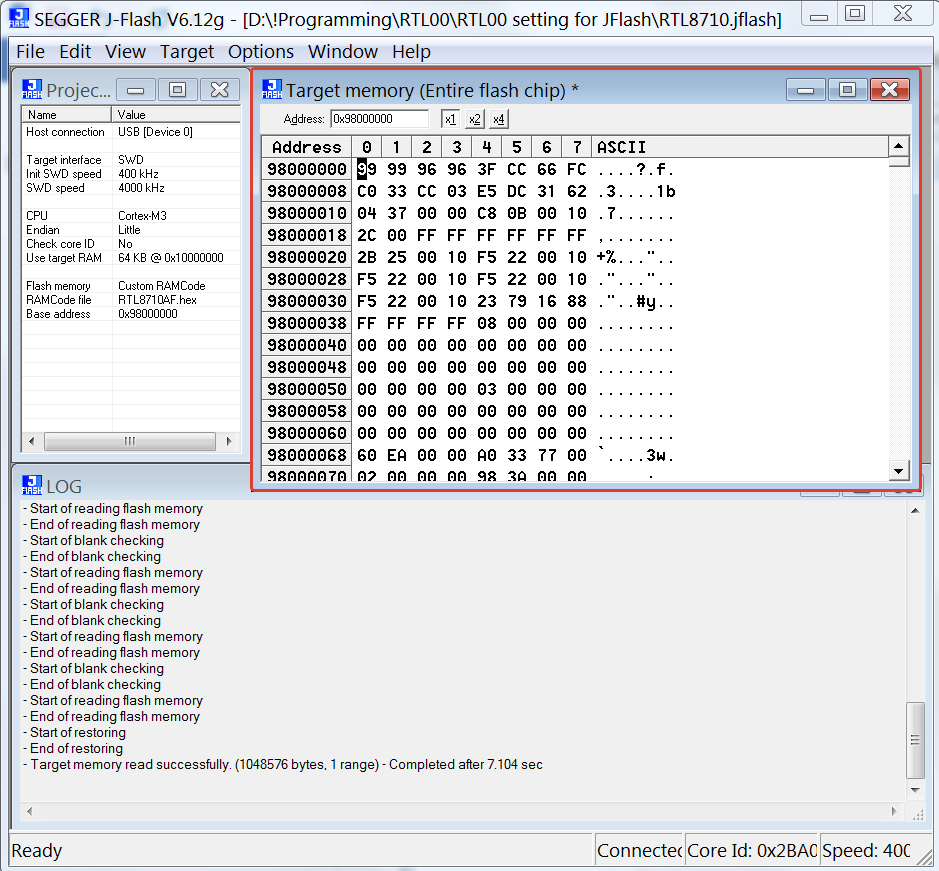
Заходим в **Target – Manual programming – Read back – Entire chip**

Должен начать считывать чип по окончанию выдаст вот такое:

****

После нажатия кнопки ОК:

Откроется окно с содержимым чипа.

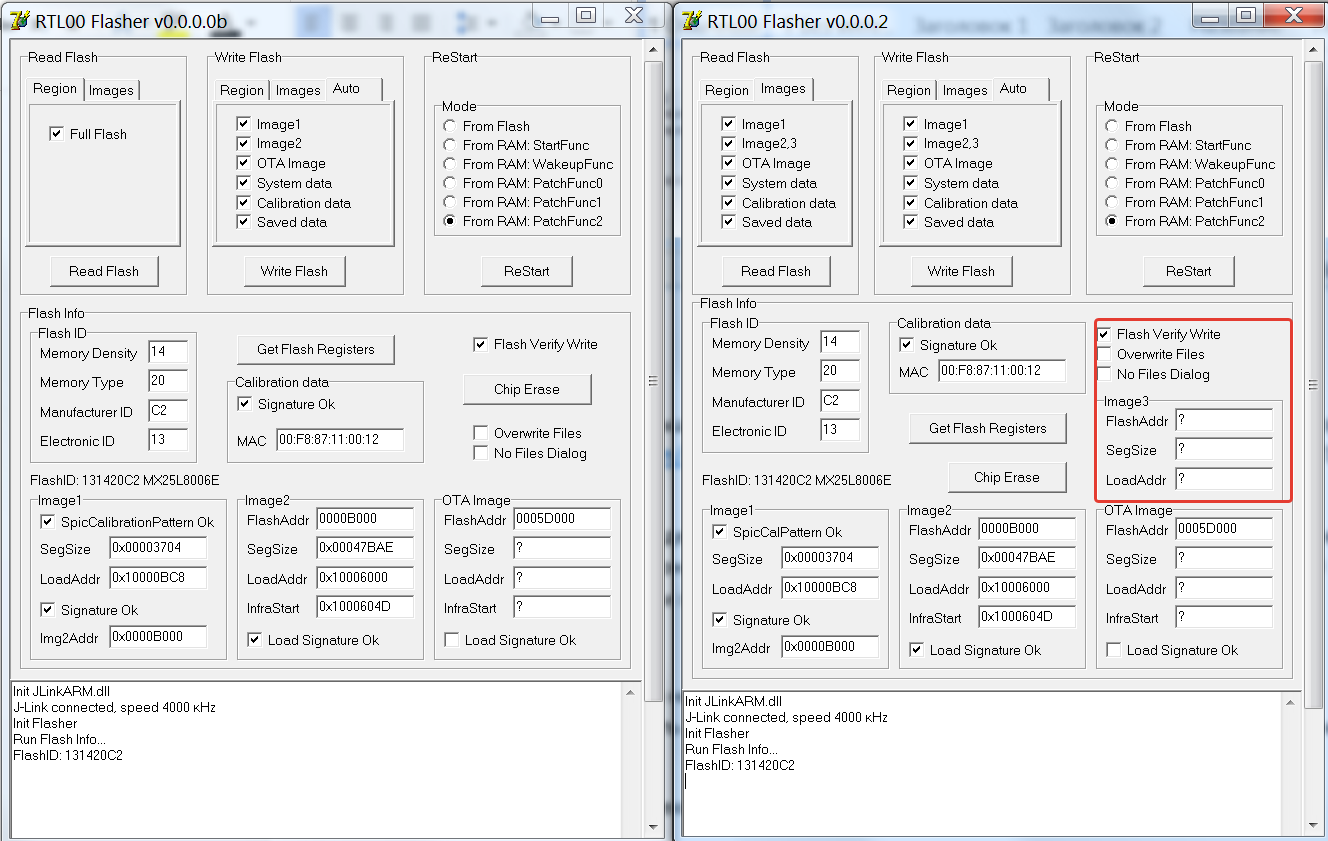


# 9. RTL00 flasher.

Это флешер (прошивальшик) сделанный коллективным усилием некотых участников форума [https://esp8266.ru/forum/](https://esp8266.ru/forum/%20) точно @pvxx и вроде кто то ещё. Работает исключительно c RTL8710 (RTL00 RTL01). Есть исходники на форуме, написаны на delphi.

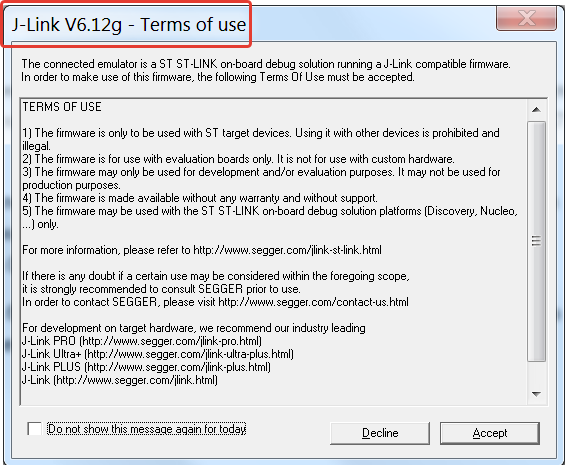
В основном, на начальном этапе, будет нужна для восстановления убитых прошивкой конроллеров. Поскольку пишет всё. Защиты от дурака в ней нет вовсе, поэтому освоение начинайте с умения делать копию фулфлеша, и заливку её обратно.

Поскольку программа сделана на коленке, в ней не всё работает, она может сглючить, чем отличаются версии непонятно и возможно что то ещё , но она работает. В папке «**RTL00 flasher**» есть две версии программы 0.0.0.0b и 0.0.0.2 отличие версий в версии 0.0.0.2 появилась возможность писать image 3 (зачем и что это не скажу).



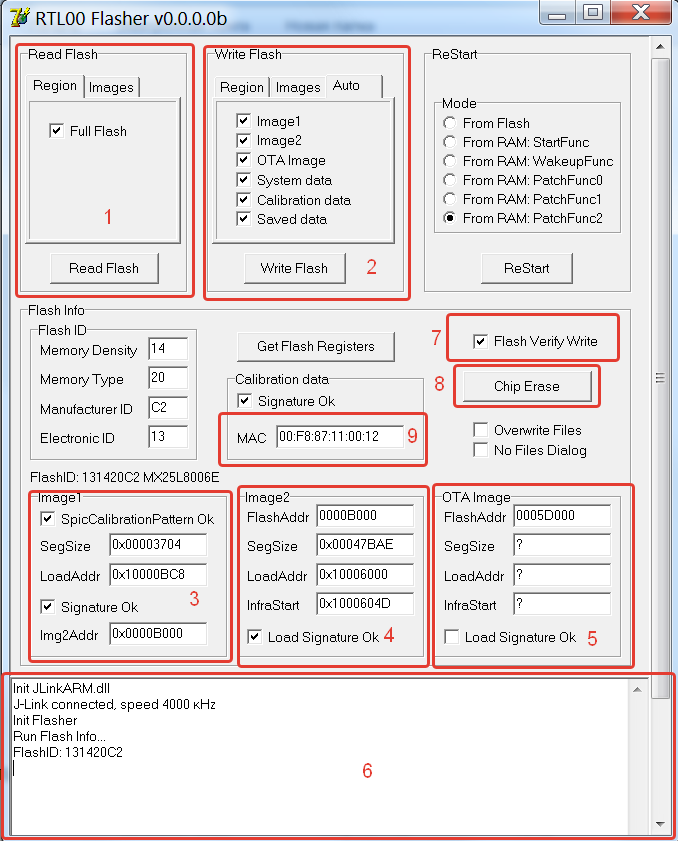
Для работы программы необходима библиотека JlinkARM.dll в той же папке что и RT00flasher.exe (она есть в fh[bdt) библиотека берётся из каталога установки J-flasher и может иметь разные версии.

При запуске RTLflasher.exe увидите версию библиотеки:



Жмёте Accept и попадаете в основное окно программы. Описываю версию 0.0.0.0b

Программа конектится автоматически ! В лог окне программы (область 6) должны появится такие строки. О проблемах ниже.



**1. Read flash**

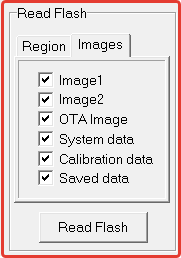
Чтение флеш памяти микроконтролера. Возможны два варианта чтения:

Fullflash с закладки **Region** (см скриншот выше).

Если снять галочку с Full Flash то можно прочитать произвольный кусок из flash.

*Здесь у меня был глюк, первый микроконтроллер с которого я считал и залил в него его же фуллфлеш благополучно «умер» была ошибка в чтении фулл флеша или ещё где сказать не могу. Что обидное это была единственная evolution board на которой мигал плавно мигал светодиод по умолчанию. Потом я её прошил другим фулом и она заработала, но стартовая прошивка пропала*.

И некоторых частей прошивки с закладки **Images**

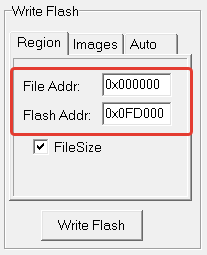


В принципе этот чип (особенно купленный в китае) не имеет индивидуальных настроек и калибровок. Более того не зная модель памяти и что как куда компилируется и линкуется всё это бессмысленно, поэтому информация на текущий момент для справки.

2. Write flesh

Аналог чтения но это запись. Всё просто. Есть три закладки Region/Images/Auto. **Images** не работает – это нормально в данной версии.

**Region** запись флеша.



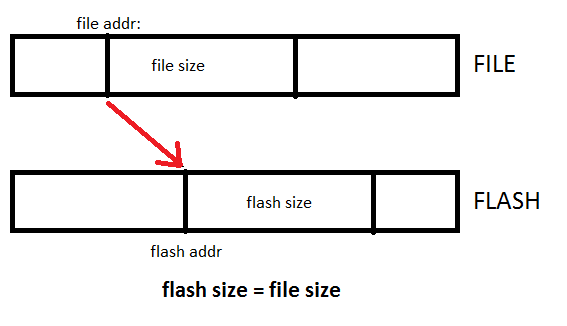
Позволяет записать произвольный кусок флеша в контроллер.

File addr – Указываете с какого байта файла прошивки (его вы выберите когда нажмёте Write Flash)

Flash addr – Указываете с какого адреса будет происходить запись.

File size – Галочка указывает что будет пытаться шить весь файл. Если снять галочку то появится поле размера. В котором нужно указать сколько байт из файла шить.

Иллюстрация выглядит так.



Для записи фулфлеша.

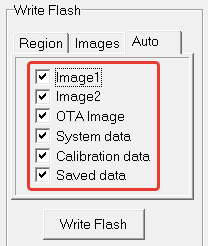
**File addr: 0x00000000**

**Flash addr: 0x00000000**

**File size: галочка.**

Размер фуллфлеша 1 MB.

Закладка auto. При отмеченных всех флажках аналогична записи фулл флеша (ну почти, разница как я понимаю в том что в случае фулл флеша пишется абсолютно всё, а в этом случае только программа. Неиспользуемая память не перезаписывается).



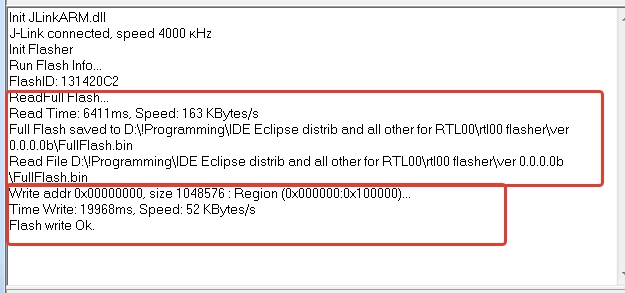
*Если указать не восемь нулей в качестве адреса а например шесть то у меня почему то не шьётся в версии 0.0.0.2 в версии 0.0.0.0b всё работает.*

**3,4,5** Без понимания модели памяти и что куда шьётся не нужны. (Я пока её не знаю).

**6. Лог.**

**Всё понятно.**

Вот так например выглядит чтение и запись фулл флеша.



7. Проверка записи, всегда ставьте галочку хуже не будет.

8**. Erase Chip**. Стереть чип полностью. (пока не представляю зачем)

**9. MAC** Мак адрес. (Как понял, каждый девайс должен иметь свой мак адрес, если у вас он один то всё нормально, если несколько с одинаковыми то будет проблема. Китайцы шьют одну прошивку и мак у плат одинаковый, поэтому нужно его менять. Разные поставщики могут иметь разный мак, но в партии скорее всего будет один)

## 9.1 Базовая работа с прграммой.

(Чтение фулл флеша и заливка его обратно)

1. После запуска программы она должна соединиться с контроллером. (если нет читайте ниже)

2. Область Read flash – region – галочка на фулл флеш – кнопка read flash.

При чтении может подвиснуть на несколько секунд (5-10 сек) это нормально, по окончанию предложит сохранить считанный фулл флеш на диск.

3. Область Write flash – region –

**File addr: 0x00000000**

**Flash addr: 0x00000000**

**File size: галочка.**

Кнопка Write flash.

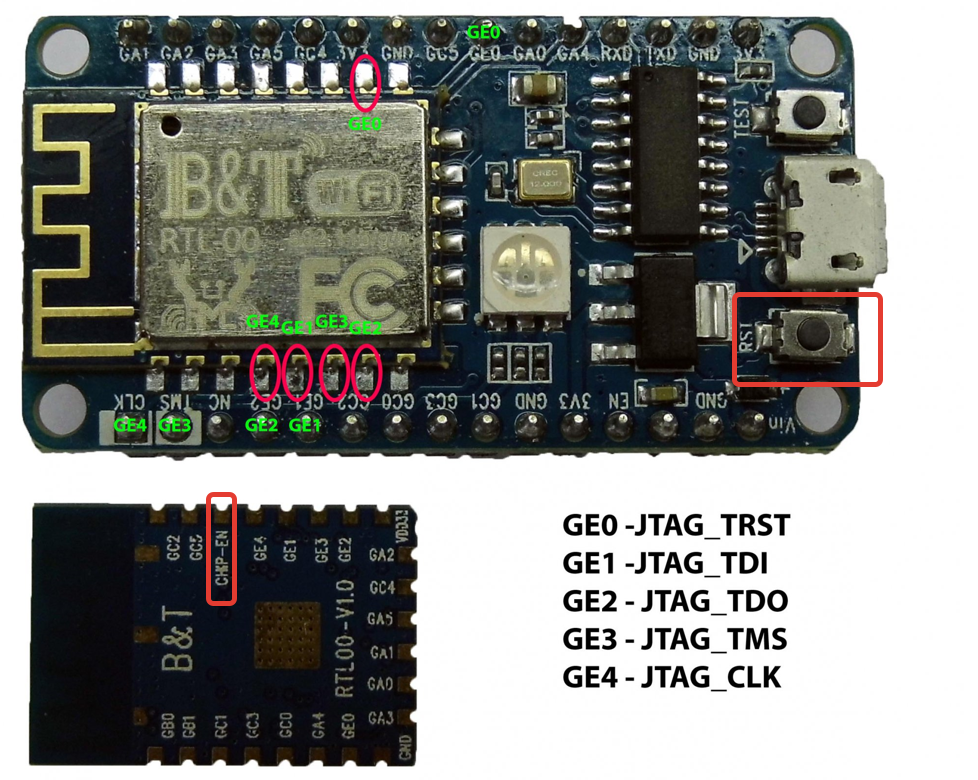
При записи так же может подвиснуть на несколько секунд (5-15 сек).

**Всё вы прочитали и записали свой флеш !!!**

*Если связи нет (если зашили на смерть такое возможно)то необходимо следущее.*

*Запустите J-Flash (выбирите настройки для RTL8710 как указано в 6 пункте) и в меню target наведитесь на пункт connect.*

*Одновременно с нажатием connect жмите кнопку ресет на evolution board или сажайте на землю вывод CHP-EN.*

**

*C какой то попытки всё получится. После чего запускайте RTL00 flasher он автоматически присоеденится*.

# 10. Память в контроллере RTL8710

Важно ! Я сам плохо понимаю как она организована информации мало, многое построил на умозаключениях. По большому счёту, знать как она устроена, где и что хранится, человеку пишущему прикладные программы не нужно. Выполняется и бог с ней. Но для общего развития и понимания пытаюсь разобраться. Если видите ошибки или всё схема некорректна пишите в личку или в других моих темах. Поправлю.

Если посмотреть описание чипа RTL8710 то по памяти есть такая информация.

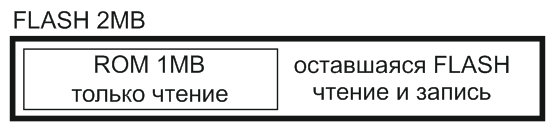
**1 Мбайт ROM, 1 Мбайт Flash-память, 512 Кбайт ОЗУ;**

Как это понимать.

Физически в любом контроллере существует 2 типа памяти FLASH в неё можно писать (очень медленно и ограниченное число раз) и читать (побыстрее и неограниченно число раз), и после пропадания питания в ней всё сохраняется. И RAM (DRAM,SDRAM,DDRRAM) можно читать и писать очень быстро и неограниченное число раз, но при пропадании питания все данные исчезают.

ROM это (Read only memory) память только для чтения. Физически она располагается в FLASH. Поэтому говоря о ROM, надо понимать что это какая то часть FLASH памяти контролера.

Разница между FLASH и ROM в том что флеш память программа изменять может а ROM нет. (ЭТО делается **аппаратным** способом, для разделения облати данных от программы в целях безопасности, что бы данные никогда не стали исполняемым кодом.)



В данном чипе ROM=FLASH.

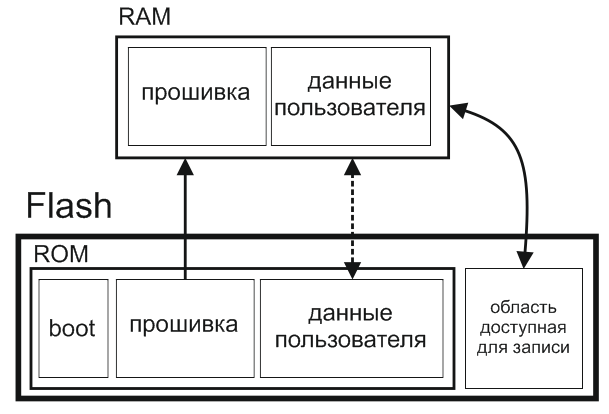


То есть занимает всю Флеш память (если упрощённо) и формально области для чтения записи флеш программой нет. Писать во флеш нужно когда вам надо сохранять состояние программы во время пропадания питания. Хотя обычно токая область есть, но она как правило очень небольшая (от нескольких байт до 2-4 Кбайт и отдельно её не указывают, есть ли она в RTL я не знаю)

**Прошивка=программа пользователя=firmware.**

**System data = «Выбор какую прошивку копировать в память»**

Упрощённо структура памяти обычного контролера выглядит так.



1.Boot выполняется из флеш, выполняет инициализацию устройств контролера (процессор, порты, таймеры, и.т.д.) копирует пользовательскую программу (прошивку) в память и запускает программу из памяти.

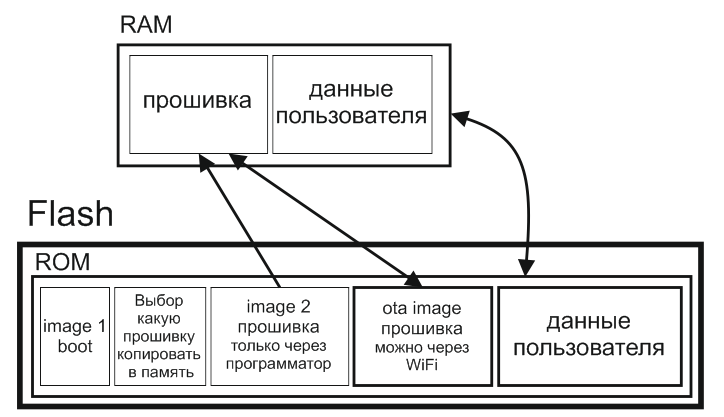
2. Программа выполняясь в RAM подгружает необходимые данные пользователя в оперативную память.

3. Если нужно читает/записывает в доступную для чтения область FLASH.

Область ROM **никогда не изменяется при работе программы**.

Но в RTL сделали такую вещь как обновление прошивки по через вайфай, вещь удобная но из за этого структура памяти изменилась. Появилась необходимость в двух прошивках.

Структура памяти в RTL8710 Выглядит так.



1. Boot выполняется из флеш, выполняет инициализацию устройств и попадает в область «System data» где указывается какую прошивку (firmware) надо загрузить в память. Копирует нужную прошивку в память и запускает программу из памяти.

1.1 Прошивка из из **image2** в состоянии изменить данные в **ota image**, **пользовательских данных**, и в **system data**.

Возможно и саму себя и загрузчик. (нет информации, по идее не должна такого делать)

То есть изменения в ROM доступны не только через програматор но и через wifi.

Есть конечно и другие зоны, но по большому счёту знать что и как не нужно если не собираетесь делать флешер для контролера. Просто информация для общего понимания.

О прошивке через wifi позже.

# 11. Немного о том, что такое IDE Eclipse.

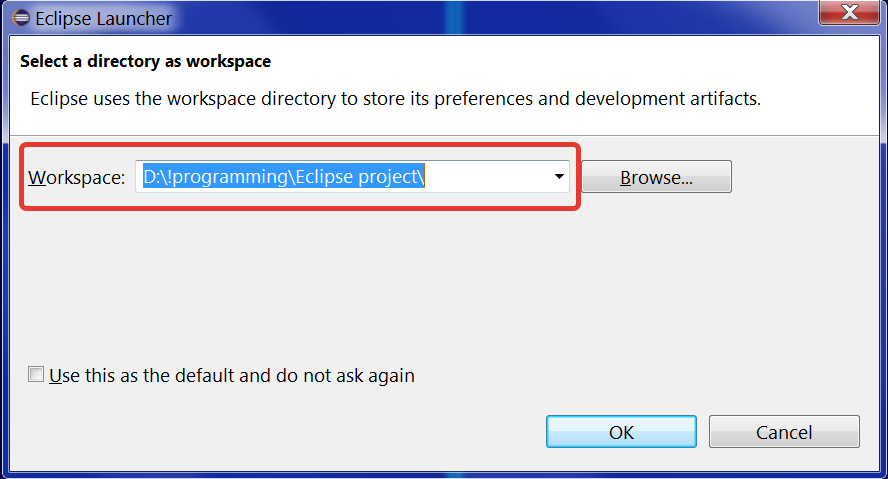
# 12. Установка IDE Eclipse и всего необходимого (под windows).

## 12.1. установка IDE Eclipse

версия Эклипса 4.6 есть в папке ”Eclipse Neon distrib “ или ей всегда можно списать здесь : <http://www.eclipse.org/downloads/eclipse-packages/>

Это обычный архив без установщика просто распакуйте в любую папку. ( киррилических символов в пути лучше избегать)

Запустите Eclipse, при первом запуске он попросит указать каталог где будут хранится все проекты. Задайте его. Если не хотите больше видеть это оконо поставьте галочку.



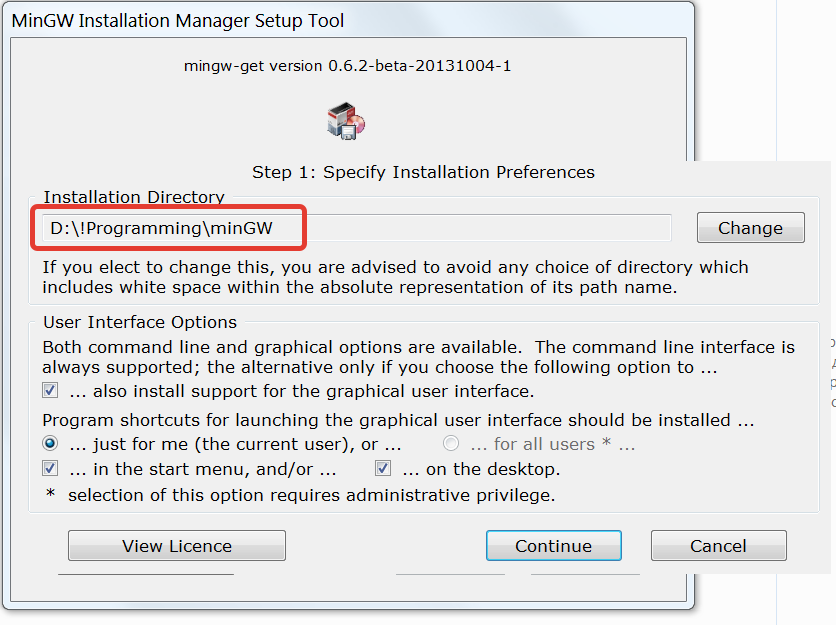
**Вы установили Eclipse !**

## 12.2 Установка minGW

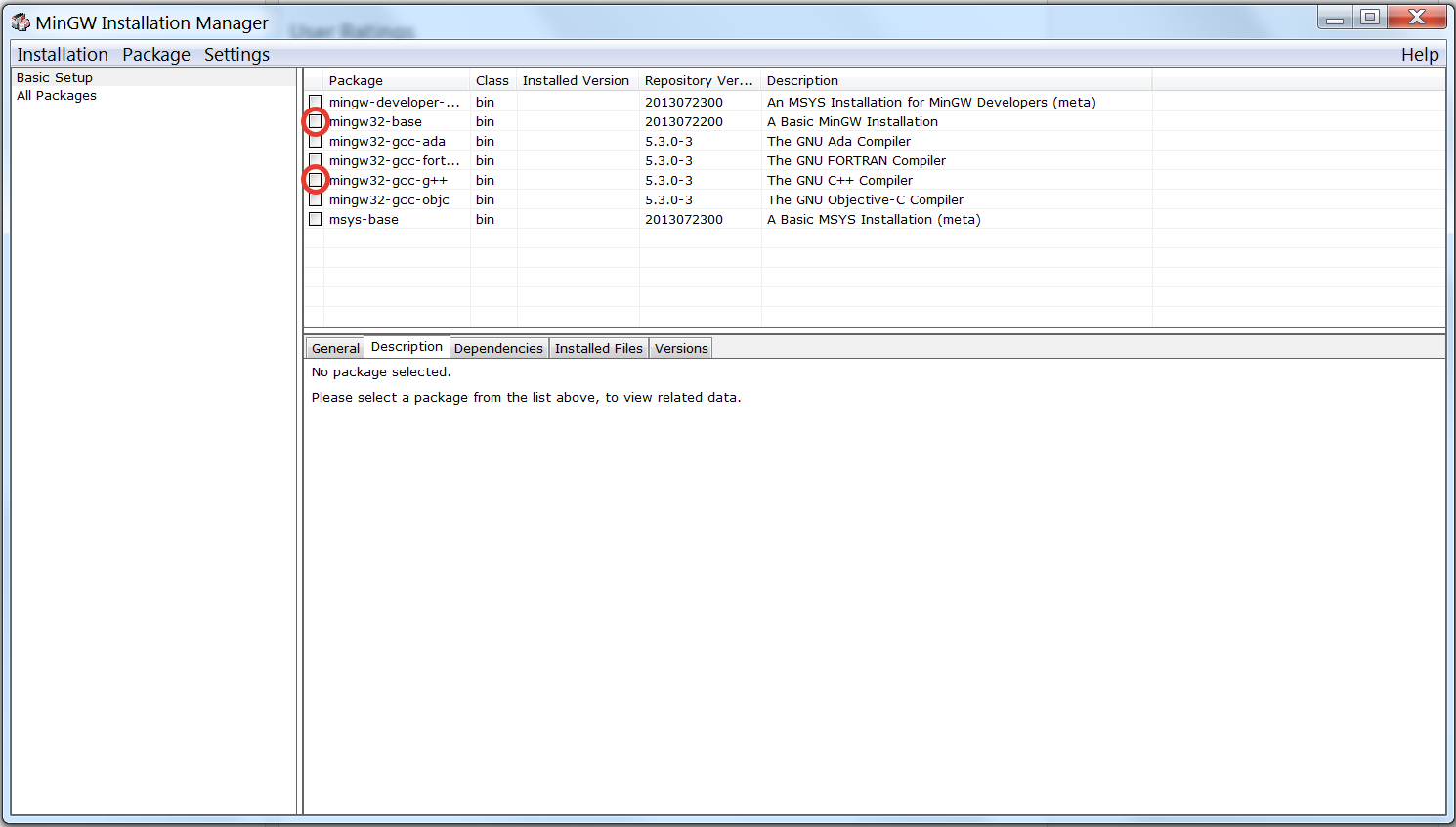
Инсталятор minGW можно найти в папке “minGW” файл mingw-get-setup.exe или взять здесь:

<https://sourceforge.net/projects/mingw/?source=typ_redirect>

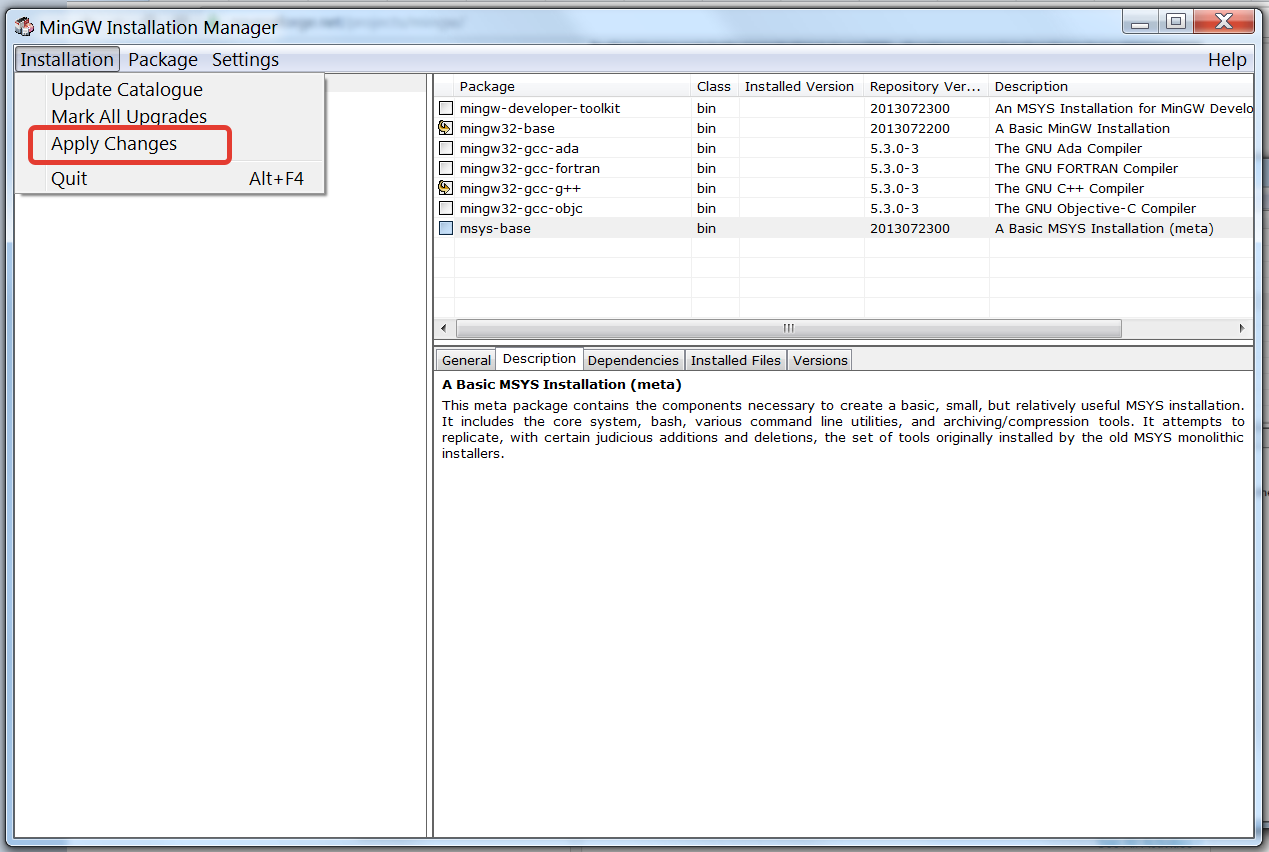
После запуска инсталятора жмите Install и он выдаст такое окошко: в нём укажите каталог установки (напоминаю про отсутствие киррилические символов в пути), галочки у пунктов достаточно понятные добавлять в стартовое меню и на десктоп решайте сами. Первую не снимайте.



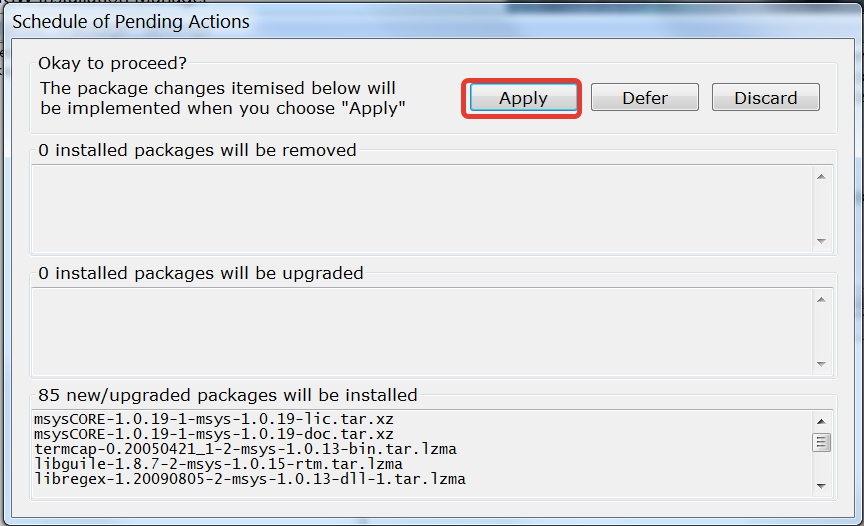
После чего он начнёт установку своих компонетов . По окончанию запустит менеджер инсталяций. В нём отметте две позиции «A basic minGW instolation» «The GNU C++ compiler»



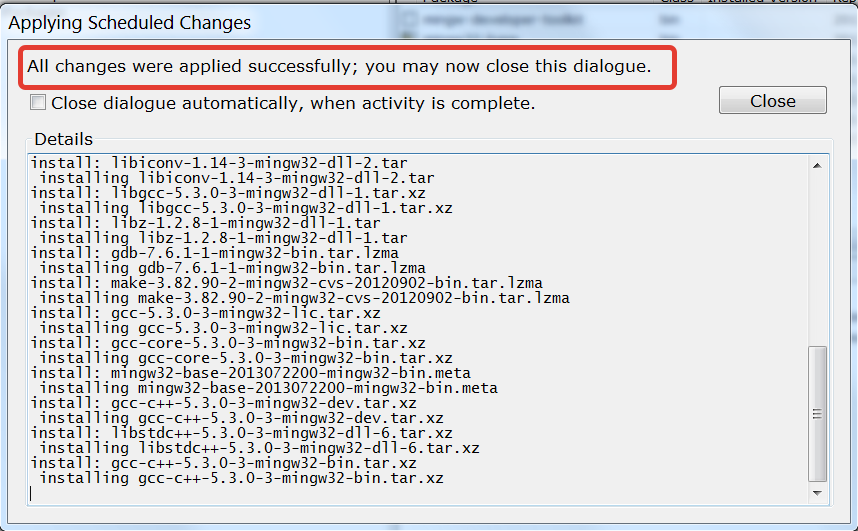
Instalation-Apply Changes



Apply

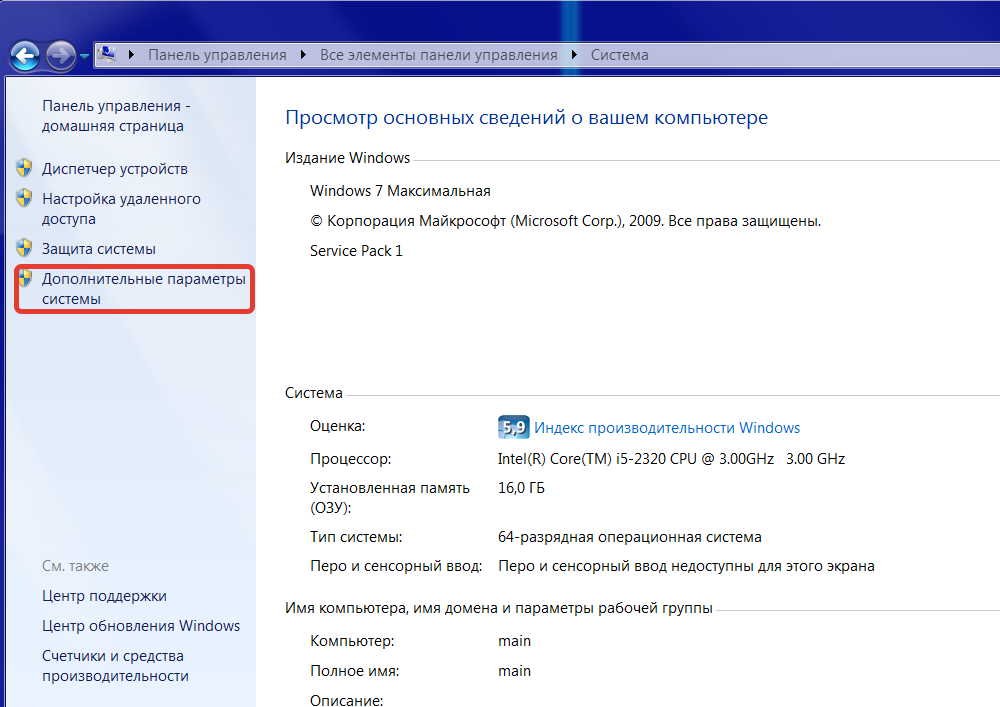


После полной загрузки закройте окно. Close.

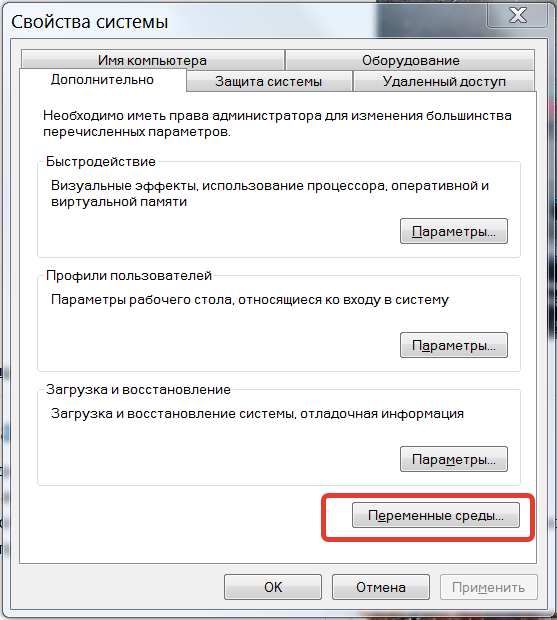


Добавление пути.

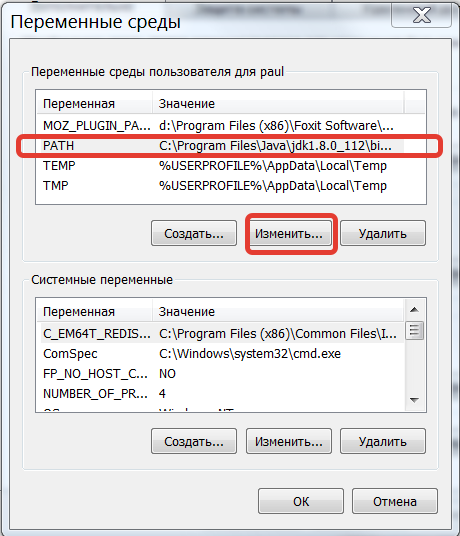
Откройте панель система и выбирите дополнительные параметры системы.



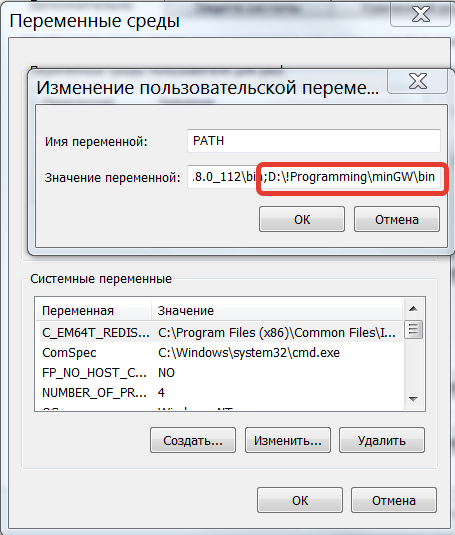
Выбирите переменные среды.



Выбирите переменную PATH и нажмите изменить.



Добавьте в конце через точку с запятой, путь куда был установлен minGW после нажмите ОК.

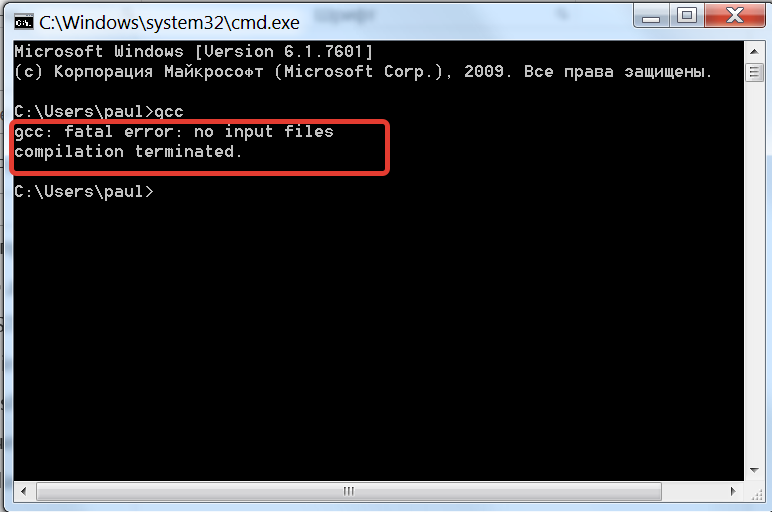


Перезагрузите компьютер !

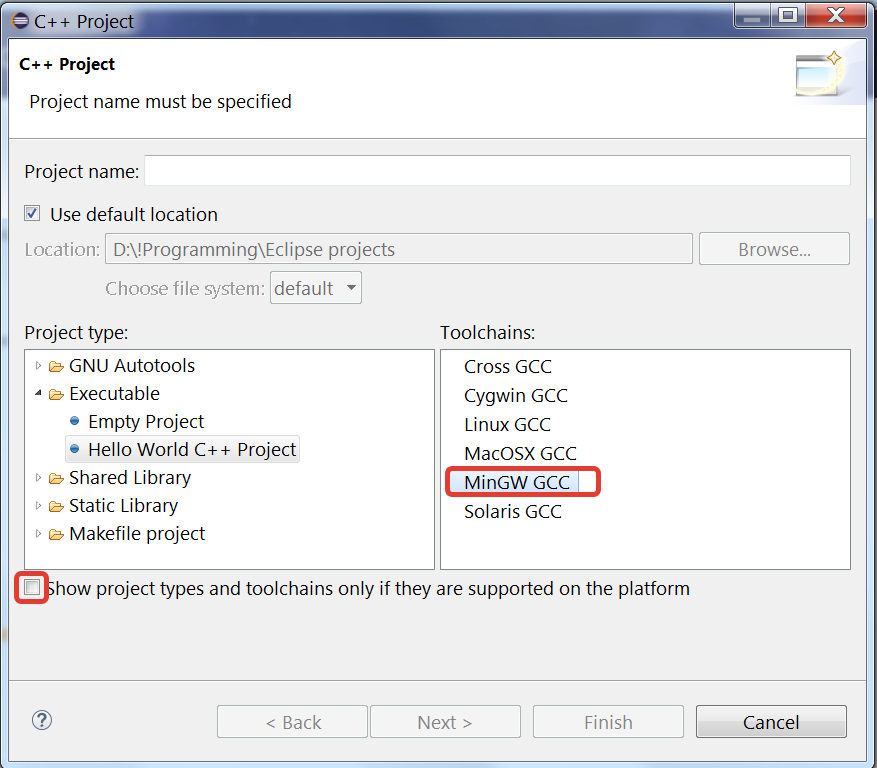
Для проверки что переменная среды прописалась корректно в консоли (**Кнопка:Пуск** - **поле: Найти программы и файлы** - **Набрать:cmd.exe**)

Наберите gcc

Ответ должен выглядеть так:



Запустите Eclipse. Выбирайте **File – New – C++ Project** откроется следующее окно.



Выберите Hello World c++ Project и снимите галочку. minGW должен появится в тулчейнах.

**Если всё так, то вы корректно установили minGW в среду Eclipse !!**