Загрузка работающей программы делается в четыре этапа:

- Средствами процессора при положении перемычек boot[2..0]=001 загружается и выполняется код размером 2048 байт из сектора 0 NAND.
- Этот код зажигает светодиод на выводе GC31 процессора (выдает 0 вольт), загружает 26 секторов начиная с сектора 1 NAND в память с адреса 0x00000800 (это со смещения 2048 байт внутренней статической памяти процессора), гасит ранее зажжённый светодиод, проверяет наличие сигнатуры в загруженных данных и передаёт управление по адресу 0x00000800.
- Данный код инициализирует отладочный вывод в коммуникационный порт UART0 (выводы TX-G19, RX-GA18) с подключение питания на UART-CMOS преобразователь (GC3=1, GC5=0), выполняет инициализацию 1GB DDR3 памяти и выполняет загрузку одного сектора номер 64 из NAND (со смещения 0x20000) в DDR3 по адресу 0x40000000.
- Используя информацию из заголовка, получает информацию о полном размере программы и адресе запуска. Далее выполняет загрузку необходимого количества секторов, проверяет контрольную сумму и при подтверждении целостности файла передает управление на адрес 0x40000100.

Для загрузки в NAND память технологической утилитой подготавливается двоичный образ для прошивки –

Утилитой ***wtitefsbl.exe*** формируется двоичный образ из трех компонент

```
wtitefsbl.exe progimage.bin 0x00000000 tc1_vm14_boot0.bin -w
wtitefsbl.exe progimage.bin 0x00000800 tc1_vm14_boot.bin -w
wtitefsbl.exe progimage.bin 0x00020000 tc1_vm14_app.stm32 -w
```

Полученный файл (необходимо контролировать не превышение размера 2 мегабайта, 2097152 байт) преобразуется в Intel Hex формат с базовым адресом 0x80000000 утилитой ***bin2ihex.exe***.

```
bin2ihex  -l 0x80000000 -s progimage.bin >progimage.hex
```

Результирующий файл ***progimage.hex*** используется для передачи в инструментальную программу, выполняющуюся на целевой плате через терминальную программу.

Файл ***tc1_vm14_app.stm32*** формируется из двоичного образа целевой программы, сформированной для расположения с адреса 0x40000100 утилитой ***stm32image.exe***, формирующей заголовок с сигнатурой, информаций о размере и контрольной суммой обработанного образа.

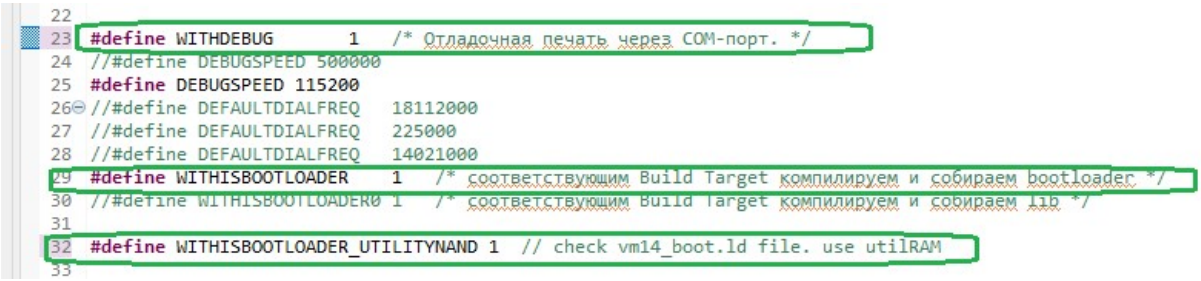
```
stm32image -l 0x40000100 -e 0x40000100 -s tc1_vm14_app.bin -d tc1_vm14_app.stm32
```

Создание tc1_vm14_util.hex

Рекомендуется выполнять до остальных операций формирования компонентов *progimage.bin* , так как при данном действии файл *tc1_vm14_boot.bin* используется как место временного хранения. Формируется выполнением целей ***Clean*** и ***Bootloader*** в IDE с преследующим ручным переименованием файла *tc1_vm14_boot.hex* в *tc1_vm14_util.hex* или выполнением из командной строки

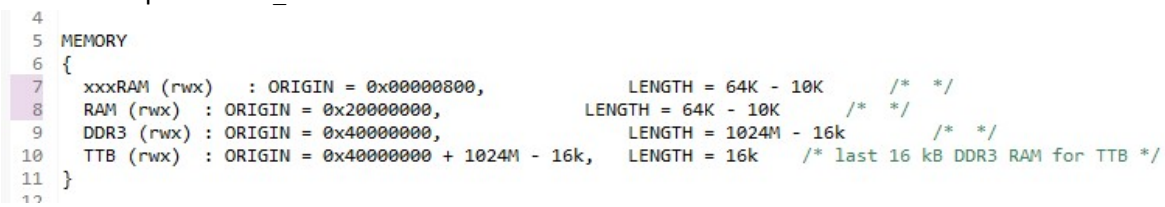
```
make clean boot
copy tc1_vm14_boot.hex tc1_vm14_util.hex
```

Состояние файла *product.h*:



Состояние файла *Makefile*:

Состояние файла *vm14_boot.ld*:



Создание tc1_vm14_boot0.bin

Формируется выполнением целей ***Clean*** и ***Support Library*** в IDE или выполнением из командной строки

```
make clean lib
```

получившийся файл *tc1_vm14_boot0.bin* используется в последовательности формирования *progimage.bin*

Состояние файла *product.h*:

```
22
23 //#define WITHDEBUG     1   /* Отладочная печать через COM-порт. */
24 //#define DEBUGSPEED 500000
25 #define DEBUGSPEED 115200
26 //#define DEFAULTDIALFREQ   18112000
27 //#define DEFAULTDIALFREQ   225000
28 //#define DEFAULTDIALFREQ   14021000
29 #define WITHISBOOTLOADER    1   /* соответствующим Build Target компилируем и собираем bootloader */
30 #define WITHISBOOTLOADER0   1   /* соответствующим Build Target компилируем и собираем lib */
31
32 //#define WITHISBOOTLOADER_UTILITYNAND 1    // check vm14_boot.ld file. use utilRAM
33
```

Состояние файла *Makefile*:

```
57
58 # Define optimisation level here
59 #OPT = -Og
60 #OPT = -Ofast -flto
61 OPT = -Os -flto=4 --specs=nano.specs
62 #OPT = -Os
63
```

Создание tc1_vm14_boot.bin

Формируется выполнением целей ***Clean*** и ***Bootloader*** в IDE или выполнением из командной строки

make clean boot

получившийся файл *tc1_vm14_boot.bin* используется в последовательности формирования *progimage.bin*

Состояние файла *product.h*:

Удаление обозначения комментария в начале строки 23 (два символа «косая черта») допустимо для получения диагностических сообщений в последовательный порт.

```
22
23 //#define WITHDEBUG     1   /* Отладочная печать через COM-порт. */
24 //#define DEBUGSPEED 500000
25 #define DEBUGSPEED 115200
26 //#define DEFAULTDIALFREQ   18112000
27 //#define DEFAULTDIALFREQ   225000
28 //#define DEFAULTDIALFREQ   14021000
29 #define WITHISBOOTLOADER    1   /* соответствующим Build Target компилируем и собираем bootloader */
30 //#define WITHISBOOTLOADER0 1   /* соответствующим Build Target компилируем и собираем lib */
31
32 //#define WITHISBOOTLOADER_UTILITYNAND 1    // check vm14_boot.ld file. use utilRAM
33
```

Состояние файла *Makefile*:

```
58 # Define optimisation level here
59 #OPT = -Og
60 #OPT = -Ofast -flto
61 #OPT = -Os -flto=4 --specs=nano.specs
62 OPT = -Os
```

Состояние файла *vm14_boot.ld*:

```
4
5  MEMORY
6  {
7    RAM (rwx)  : ORIGIN = 0x00000800,             LENGTH = 64K - 10K      /*  */
8    utilRAM (rwx)  : ORIGIN = 0x20000000,             LENGTH = 64K - 10K      /*  */
9    DDR3 (rwx) : ORIGIN = 0x40000000,                 LENGTH = 1024M - 16k        /*  */
10   TTB (rwx)  : ORIGIN = 0x40000000 + 1024M - 16k,   LENGTH = 16k    /* last 16 kB DDR3 RAM for TTB */
11 }
12
```

Создание tc1_vm14_app.stm32

Формируется выполнением целей ***Clean*** и ***Default*** в IDE или выполнением из командной строки

```
make clean all
```

получившийся файл *tc1_vm14_app.stm32* используется в последовательности формирования *progimage.bin*

Состояние product.h:
Удаление обозначения комментария в начале строки 23 (два символа «косая черта») допустимо для получения диагностических сообщений в последовательный порт.

```
22
23 //#define WITHDEBUG     1   /* Отладочная печать через COM-порт. */
24 //#define DEBUGSPEED 500000
25 #define DEBUGSPEED 115200
26 //#define DEFAULTDIALFREQ   18112000
27 //#define DEFAULTDIALFREQ   225000
28 //#define DEFAULTDIALFREQ   14021000
29 //#define WITHISBOOTLOADER  1   /* соответствующим Build Target компилируем и собираем bootloader */
30 //#define WITHISBOOTLOADER0 1   /* соответствующим Build Target компилируем и собираем lib */
31
32 //#define WITHISBOOTLOADER_UTILITYNAND 1    // check vm14_boot.ld file. use utilRAM
33
```

Состояние Makefile:

```
58 # Define optimisation level here
59 #OPT = -Og
60 #OPT = -Ofast -flto
61 #OPT = -Os -flto=4 --specs=nano.specs
62 OPT = -Os
```