



Запуск программы в ОЗУ из Keil.

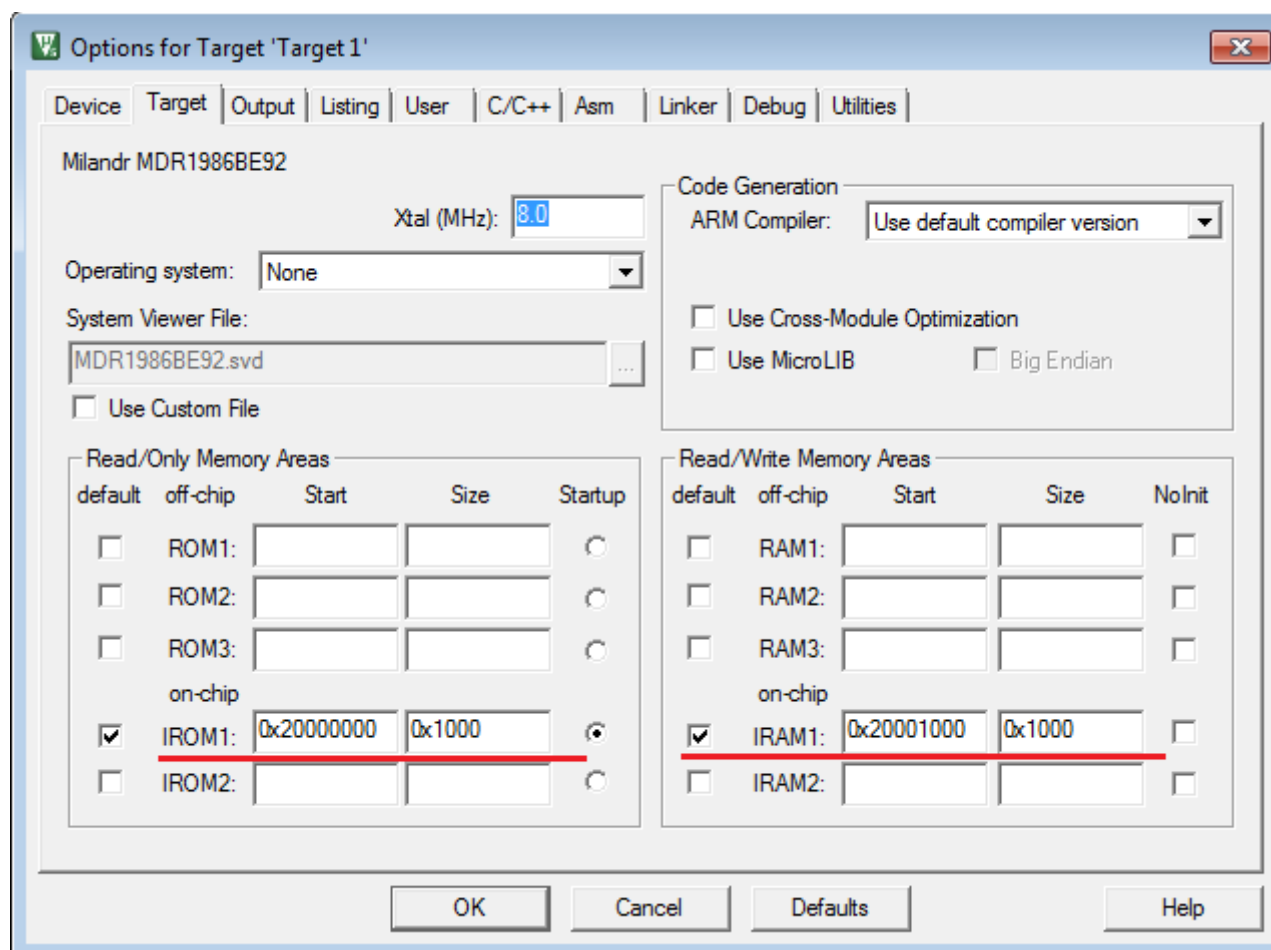
Flash память процессора имеет ограниченный ресурс где-то в 10000 циклов перезаписи. По этой, или иным причинам иногда встает необходимость разместить программу целиком в ОЗУ памяти микропроцессора. Рассмотрим, как это реализовать средствами Кейла, на примере нашей программы мигания светодиодами - "Hello Word" - светодиод.

Давайте прошьем и запустим в МК проект "HelloWorld", убедимся что светодиод мигает.

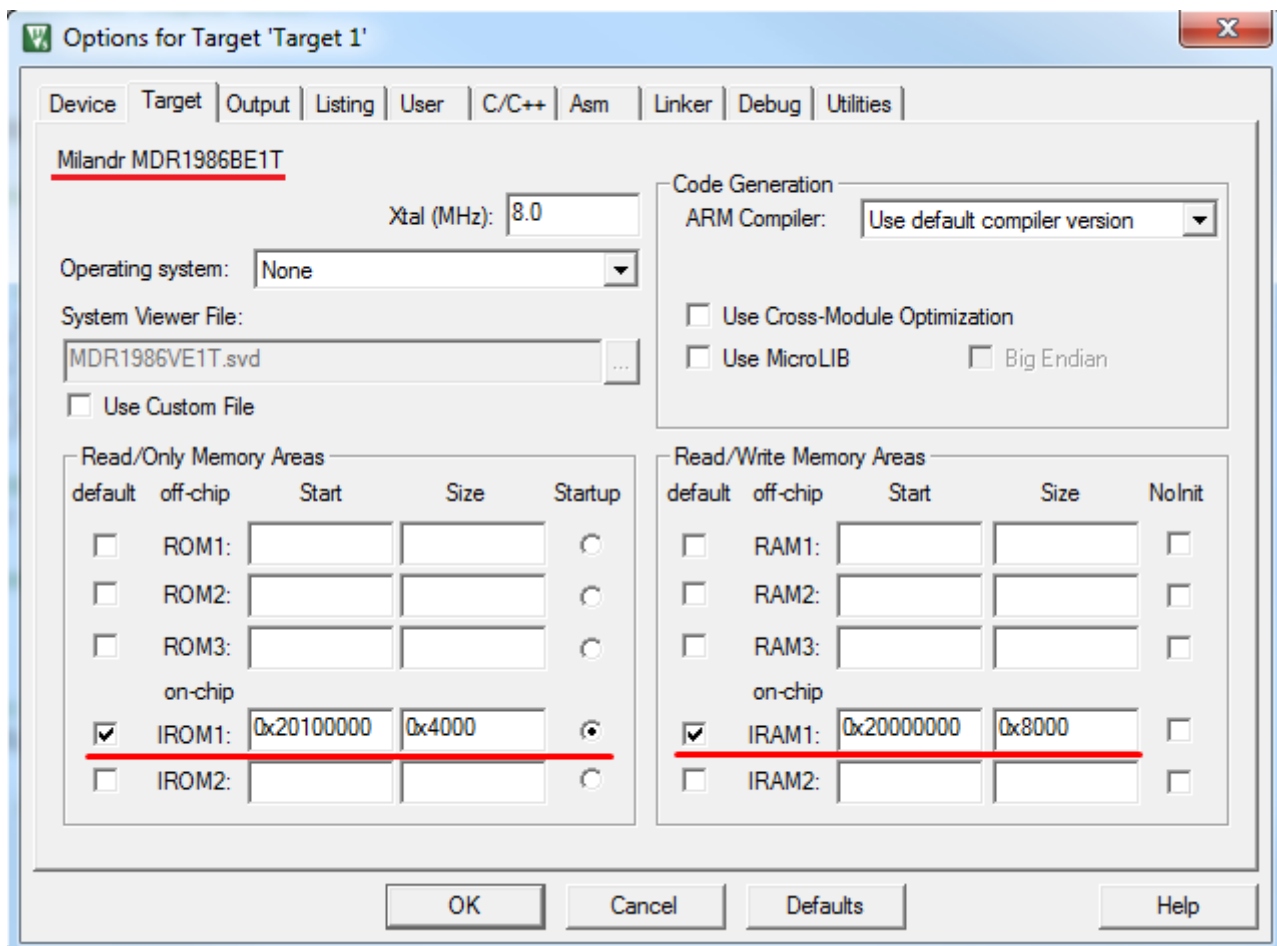
Сделаем копию проекта и назовем HelloWorld_RAM, чтобы не испортить исходный проект. Кроме этого в новом проекте уменьшим в половину значение задержек определяющих период мигания светодиодом. Теперь мигание будет происходить в 2 раза чаще. Таким образом, при загрузке из Flash у нас мигание будет в 2 раза медленней, чем при выполнении программы из ОЗУ. Так мы узнаем, какая программа в текущий момент работает.

Настройка проекта

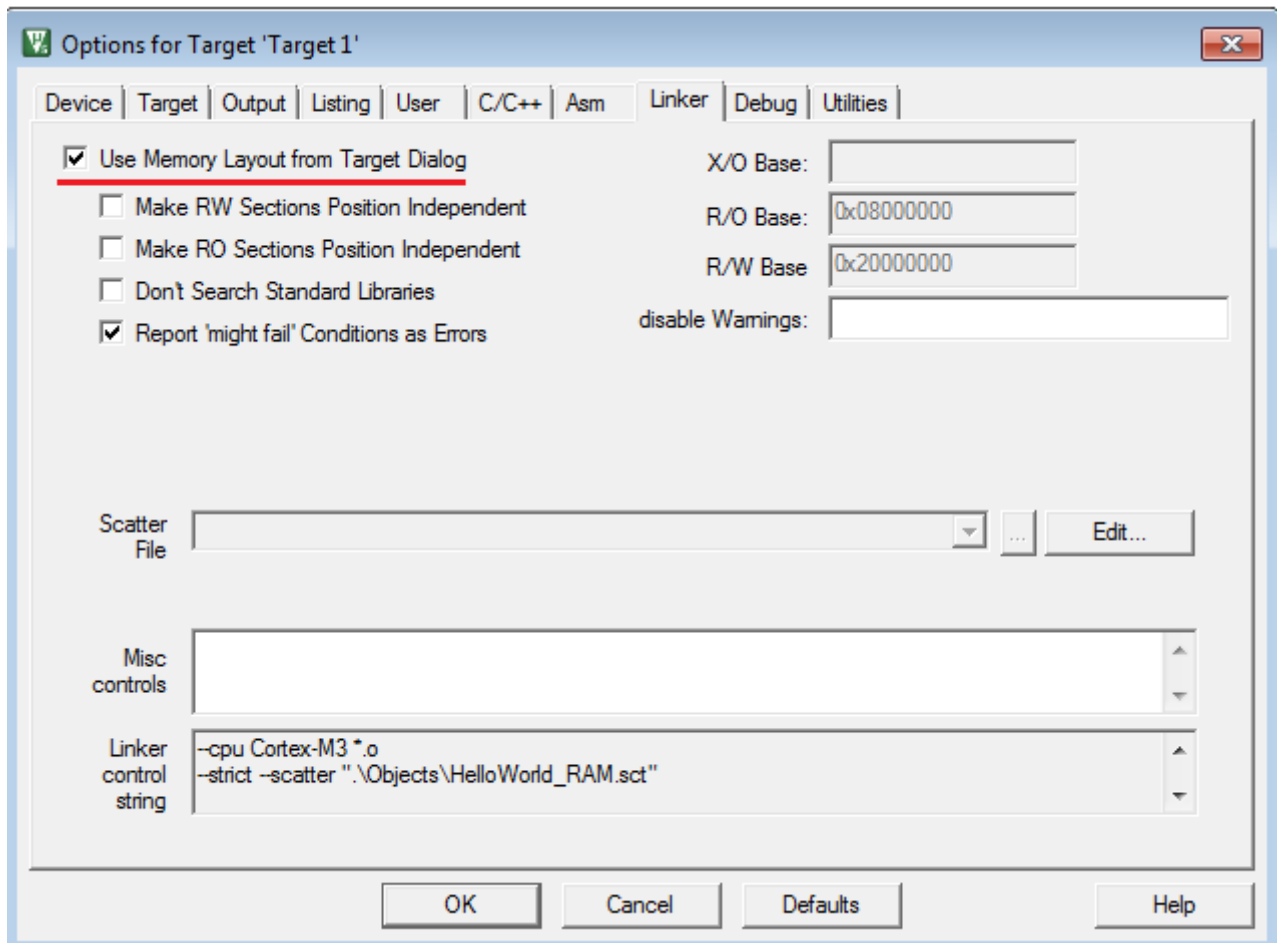
Важно понимать, что все данные, функции, адреса переходов в программе должны находиться по адресам в ОЗУ. Поэтому в закладке *Options - Target* в разделе *IROM1* вместо диапазонов памяти Flash, необходимо указать адреса ОЗУ. Для этого ту память ОЗУ, которая в обычном проекте целиком была определена в *IRAM1* (Internal RAM1) поделим на две части и одну половину отдадим под раздел *IROM1*.



В МК 1986BE1T(3T) уже имеется две области памяти ОЗУ *IRAM1* и *IRAM2*, поэтому можно одну из них целиком выделить под *IROM1*. Однако для этого подходит только память *IRAM2*, адреса которой начинаются с 0x20100000. Только к этой памяти имеет доступ шина АНВ_Lite, поэтому код из этой памяти может исполняться.



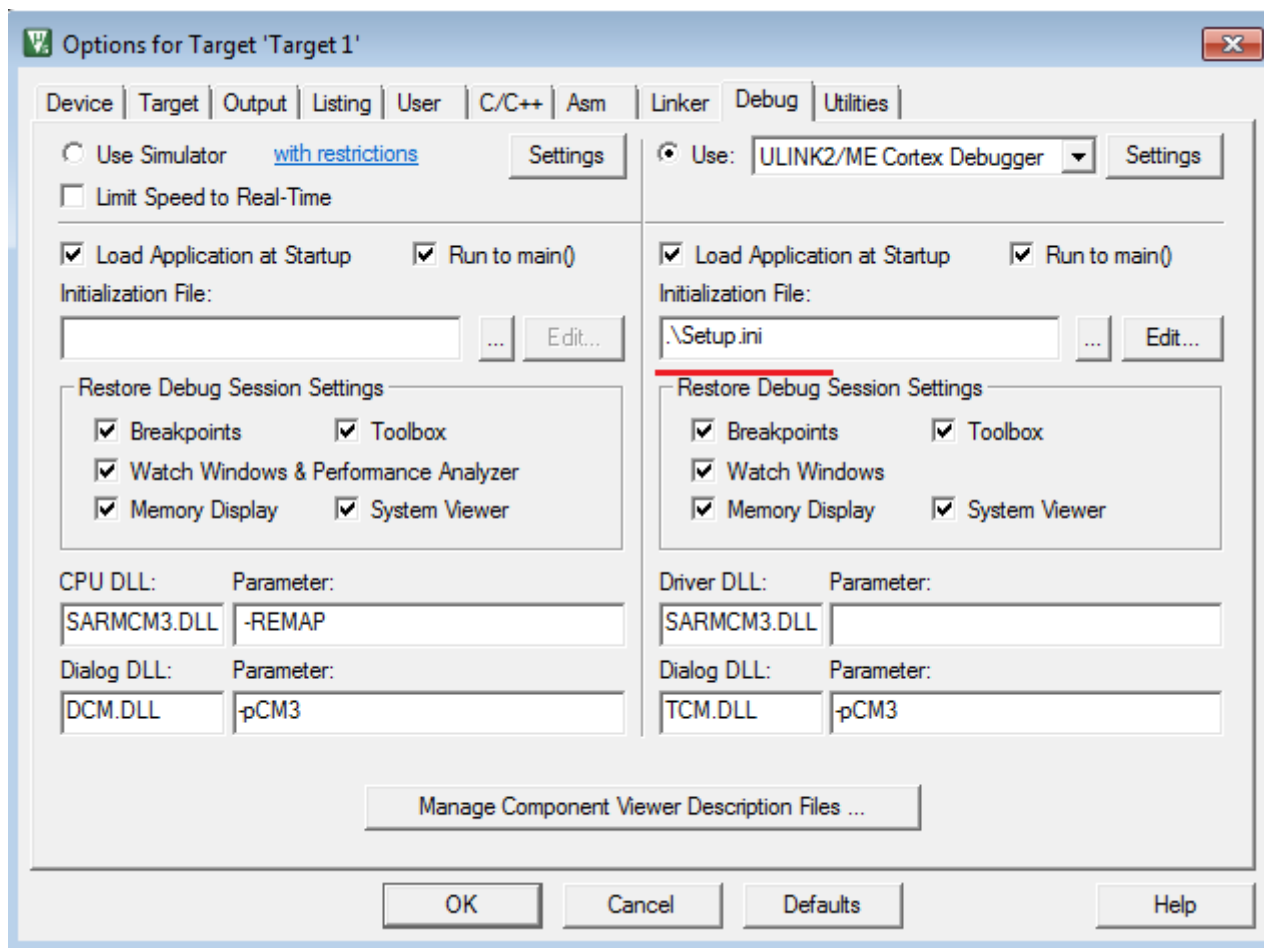
Далее в опциях проекта идем в закладку *Linker*, необходимо выбрать опцию *Use Memory Layout from Target Dialog*.



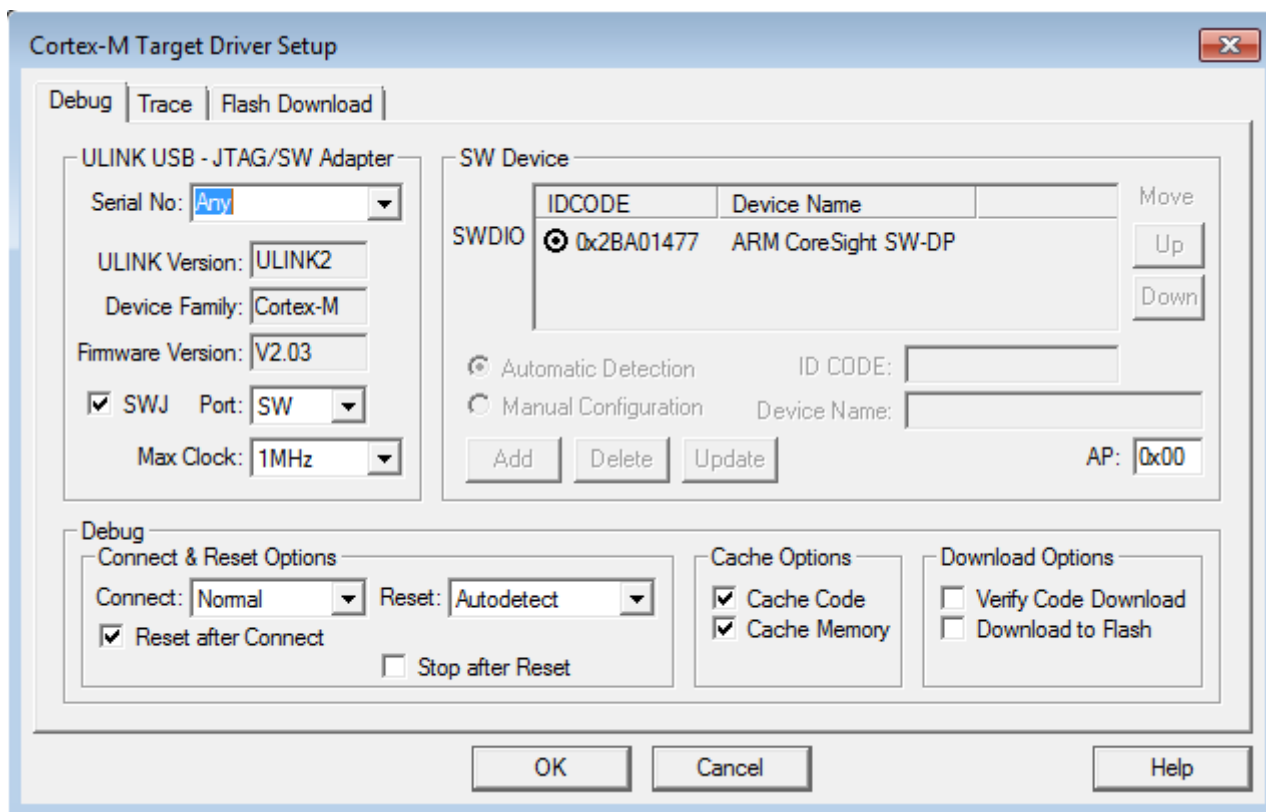
В закладке *Debug* необходимо подключить в опции "*Initialization File*" файл инициализации. Для этого зайдём в папку проекта и создадим текстовый файл *Setup.ini*. Вот листинг этого файла.

```
FUNC void Setup (unsigned int region) {  
    region &= 0xFFFFF000;  
    SP = _RDWORD(region);          // Setup Stack Pointer  
    PC = _RDWORD(region + 4);      // Setup Program Counter  
    _WDWORD(0xE000ED08, region);  // Setup Vector Table Offset Register  
}  
Setup(0x20000000);
```

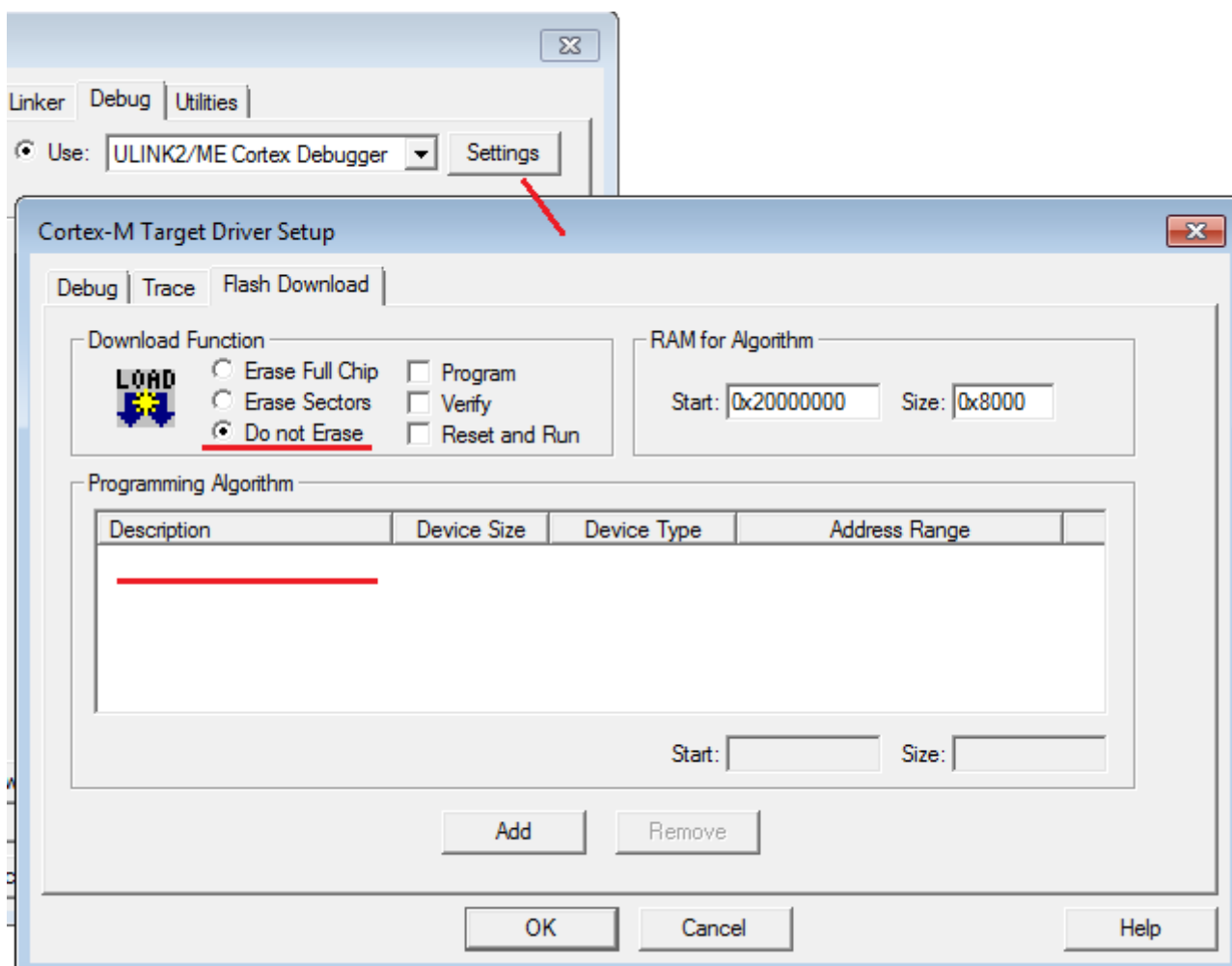
Здесь функция `Setup()` указывает отладчику, как инициализировать необходимые регистры в зависимости от стартового адреса программы. Функция `Setup()` должна быть вызвана с адресом который мы указали в *IROM1*, т.е. `Setup(0x20000000)`. Подключаем созданный файл.



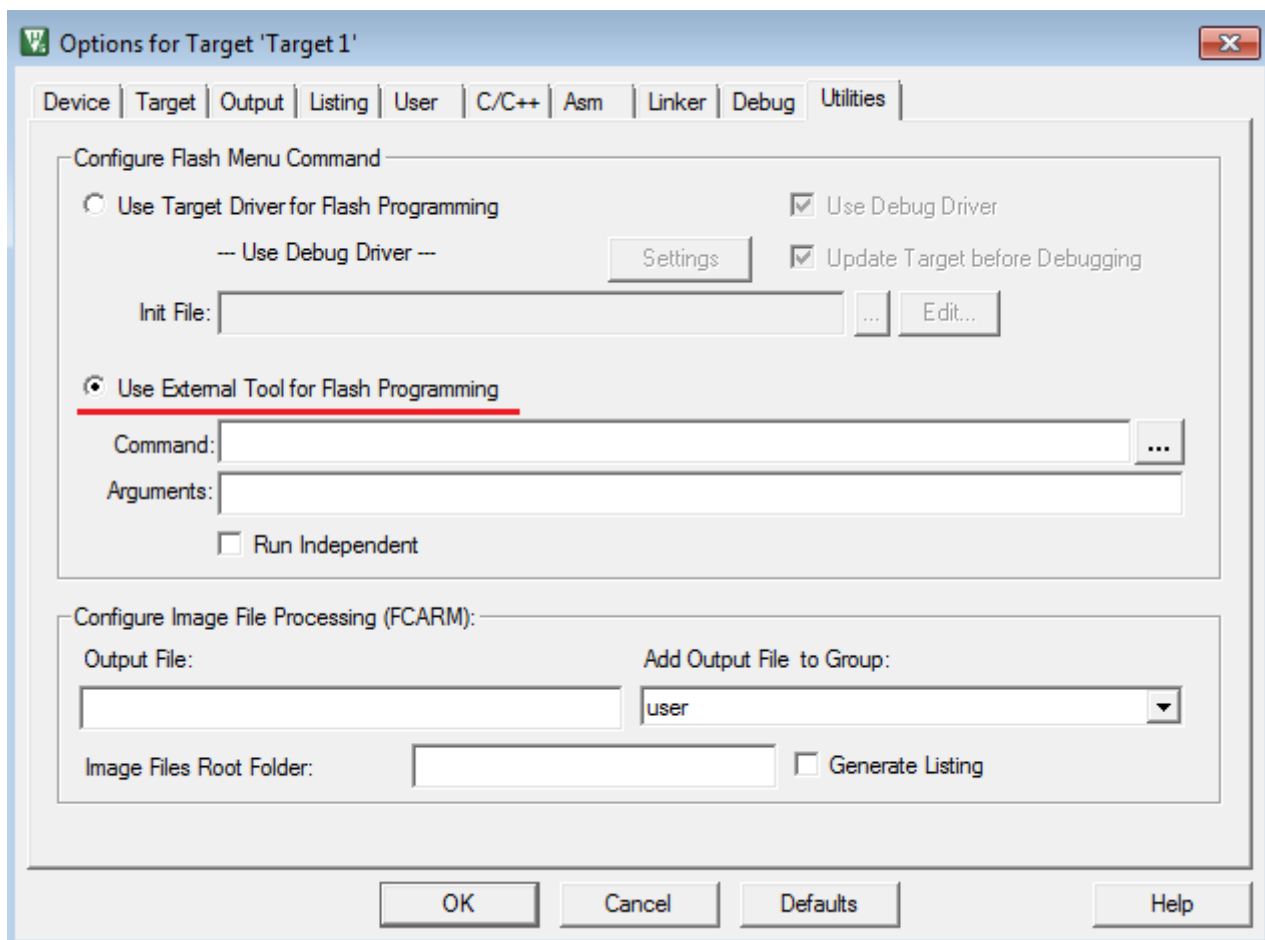
В этой же вкладке *Debug* заходим в *Settings* для программатора. В открывшемся окне, оставляем все как было настроено ранее для работы с Flash. Сейчас у меня подключен программатор ULINK2, стабильно работал он только на таких настройках.



Вносим изменения в закладку *FlashDownload*. Выбираем опцию "Do not Erase" и удаляем алгоритм программирования. Таким образом Keil не сможет изменить записанную в Flash программу при запуске текущего проекта в ОЗУ.



Последняя закладка - *Utilities*. Выключаем использование драйвера для программирования Flash выбором *Use External Tool for Flash Programming*.



Теперь мы готовы к испытаниям.

Запуск программы

Для запуска программы переходим в режим отладки **Ctrl+F5**, в этот момент программа загружается в ОЗУ. После этого осуществляем запуск, **F5**. Наблюдаем мигание светодиода. В окне ассемблера видим, что наша программа располагается в адресах ОЗУ, с **0x20000000** и т.д.

В режиме исполнения из ОЗУ мне не удалось поставить точку останова. Кейл ругнулся "Cannot set breakpoint(s) ..." и отказал в такой возможности. Оказалось, что надо было перед запуском программы поставить первую точку останова. А когда процессор остановится на ней, можно поставить следующую и так далее. Если запуск **F5** уже состоялся, то необходимо в тулбаре нажать кнопку стоп и в остановленном режиме наставить точек останова там, где это необходимо.

Давайте теперь нажмем **Reset**. По изменившемуся периоду мигания светодиода видно, что программа теперь исполняется из Flash. Это объяснимо, ведь загрузку и запуск новой программы в ОЗУ выполнил отладчик через JTAG. А при возникновении **Reset** запуск произошел штатным режимом.