**Реализация printf.**

Вывод printf можно перенаправить в любое устройство ввода/вывода, например в DBGU или на экран LCD. Встроенная в IAR библиотека содержит готовый модуль, который для этого вам нужно заменить на свою собственную реализацию, причем нет необходимости перекомпилировать всю библиотеку. Библиотечные файлы исходного кода, которые пользователь может перезадать на свои, находятся в папке …\IAR Systems\Embedded Workbench 6.4\arm\src\lib. Для изменения низкоуровнего printf нужно определить свою собственную версию модуля write.c, - нужно будет исправить код функции \_\_write и добавить реализацию функции MyLowLevelPutchar.

1. Копируем файл write.c в папку своего проекта IAR, где находятся все модули исходного кода программы.
2. Добавьте скопированный модуль write.c в дерево проекта и откройте его для редактирования. Посмотрите на реализацию функции \_\_write: в ней блоком #if 0 / #else удалена реализация низкоуровневого вывода printf. Функция \_\_write предназначена для вывода size символов из буфера buffer, причем она может делать этот вывод специальным, определяемым вами способом (благодаря вызову функции MyLowLevelPutchar). Удалите директивы #if 0 / #else / #endif и последний ненужный оператор return \_LLIO\_ERROR; (он находится между #else и #endif).
3. Определите в проекте (в любом модуле, или прямо в модуле write.c) код функции MyLowLevelPutchar. Эта функция предназначена для низкоуровневого вывода одного символа на любое устройство вывода. Это может быть порт DBGU, файл, экран LCD или любой выбранный вами интерфейс, поэтому в этой функции вы сами должны определить код для вывода символа. Функция MyLowLevelPutchar должна возвратить тот же символ, который был у нее во входном параметре, либо -1 в случае ошибки.

Пример реализации функции низкоуровневого вывода символа:

int MyLowLevelPutchar(int x)

{

while(!(USART3->SR & 0x80));

USART3->DR = ((uint16\_t)x & (uint16\_t)0x01FF);

return x;

}