

# Совместимость Guardant Code 4 и 5 поколений

Для обеспечения полной совместимости ключей Guardant Code 5 поколения (выпускаемых с июня 2021 года) с предыдущим поколением устройств, требуется однократно пересобрать загружаемый код (далее - ЗК). BIN-файл, подготовленный в соответствии с данной инструкцией, будет универсальным. Его можно будет загружать как в новые, так и старые ключи.

## Процедура сборки

Для сборки универсального ЗК нужно:

1. Скачать Guardant SDK последней версии [по ссылке](#) и установить (требуется Guardant SDK 7 Update 7 или новее)
2. Из установленного SDK потребуются следующие файлы:
  - а. Универсальный **makefile** (можно взять в составе примера [%ProgramFiles(x86)%\Guardant\SDK7\Samples\Loadable Code API\01 - General Sample\Loadable Code\])
  - б. Утилита **elfconv.exe** (в директории [%ProgramFiles(x86)%\Guardant\SDK7\Build\arm\bin])
3. В одну директорию с исходным кодом ЗК нужно скопировать новый makefile
4. Если в makefile через [CFG\_SYS\_DIR = "\$(BUILD\_ARM)\bin"] указана нестандартная директория, то в эту же директорию нужно скопировать утилиту elfconv.exe. Если этот параметр в makefile настроен «по умолчанию» (см. выше пункт 2.б), то elfconv.exe никуда копировать не нужно
5. Если требуется, указать в makefile свои параметры ROM и RAM
6. Выполнить [make template](#), затем [make](#)

## Примечание

Отличие нового makefile от старого: адрес RAM памяти задаётся в относительных значениях, как смещение. Для этого введена константа [CFG\_RAM\_OFFSET]:

```
#
# Makefile for building Guardant Code application.
#
# Aktiv Co. 2009-2021
#
# On command line:
#
# "make all" = Make software.
# "make clean" = Clean out built project files.
# "make filename" = Just compile filename into the assembler code only.
# "make template" = Generate template of blank project.
#
# To rebuild project do "make clean" then "make all".
#
# -----
# Main Configuration
# -----
# If you modified this section, you must call "make template"!
#
# C-code entry point name:
CFG_ENTRYPOINT_NAME      = main

# ROM configuration:
CFG_PROGRAM_ADDR        = 0x00020000
CFG_PROGRAM_SIZE        = 0x00008000

# RAM start address:
CFG_RAM_ADDR            = 0x10003000

# RAM configuration:
CFG_RAM_OFFSET          = 0x00000000
CFG_RAM_SIZE            = 0x00004FE0

# Input-output buffers configuration
```