Advanced key gen for SVI FF R2, R3

# Usage

genkey3w <definition-file> <device-id1> <device-id2> …

E.g.   
genkey3w std-r2.txt 12120118

## Update

The command-line key generator is now built with Visual Studio (2017). See below for the reason. The old version called genkey3 built with DevC++ 5.11 is no longer sufficient and fails to rebuild, and VS now supports designated initializers, so there no blockers in using VS

Output format: Hex keys followed by the generated key. E.g.:

[0]=0x7FFFFFFF

[1]=0x180BFE00

[2]=0xFFFFFFF8

[3]=0x00FFFFFF

[4]=0x00000000

[5]=0x7FFFFFFF

[6]=0x180BFE00

[7]=0xFFFFFFF8

[8]=0x00FFFFFF

[9]=0x00000000

[10]=0x00001FFE

[11]=0x00000000

Id 12120118: 0CEC78EC

# How to create a definition file

In a project folder FD-SW\inc\_FFP, a file named advtemplate.txt is created as a result of command-line build. It is created during Unimal phase of the build, so you can use command line  
FD-SW>gnumake proj=FFP UNIMAL   
to create it; no need for IAR license.

advtemplate.txt has the following format:

[Diagnostics]

DIAG\_NOT\_RUNNING

DIAG\_EXEC\_SIGNATURE

…

[Read]

PTB

ST\_REV

TAG\_DESC

STRATEGY

ALERT\_KEY

…

[Write]

PTB

ST\_REV

TAG\_DESC

STRATEGY

ALERT\_KEY

…

To create a desired definition file, comment out (with a “;” in column 1) any entries you want to disable. For the examples, see std-r2.txt, adv-r2.txt, and adv-r3.txt (which enables everything).

# Programmer’s details

## General design

The bit mapping in key generator must exactly match that of the firmware. So some firmware files are used in keygen3. They are written and/or created for C99, and most notably, use designated initializers. Unfortunately, this rules out Visual Studio C/C++.

## Build environment for genkey3

For simplicity, I chose DevC++ (<http://orwelldevcpp.blogspot.com/> ; download at <https://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/Dev-Cpp%205.11%20TDM-GCC%204.9.2%20Setup.exe/download>).

The criteria for selection are the following:

* Decent IDE
* Trusted compiler (GCC 4.9.2 port)
* Little-to-no learning curve

The IDE is very basic but fits the purpose. It may or may not be used for long

## Build environment for genkey3w

VS solution genkey3w.sln

## Content of genkey3 and genkey3w projects

The project is defined in genkey3.dev.

It contains the following sources:

* advhelpers.c – man-made helpers to manage files
* advlockdiag.c – automatically *generated* mapping of DIAG\_... enumeration to key bits
* advmain.c – man-made main routine and Rev.2-compatible bits calculations
* advmap.c – *generated* mapping of TB parameters to text strings
* mnadvcalc.c – man-made Rev.2-compatible key calculations (straight from firmware)
* Include directories:
  + "../FD-SW/includes"
  + "../FIRMWARE/includes"
  + "../FIRMWARE/mn\_instrum/noinstrum"
  + "../FD-SW/target/inc"
  + "../FD-SW/target/cfg"
  + "../FD-SW/target/inc/ff"
  + "../FD-SW/target/appl/fbif/inc"
  + "../FD-SW/target/inc/segm"