	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	1 of 12	02

Gemini

Software Development Kit

User Guide

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	2 of 12	02


Revision History

Rev.	Date	Author	Changes
00	March 2021	AS	Initial Release
01	3 rd October 2023	MS	Edited with latest SDK changes
02	1 st November 2023	MS	KMS Graphics Inclusion for Raspian AARCH64

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	3 of 12	02

Index

Overview	4
Installation.....	5
Windows	5
Linux	5
Structure of the Gemini SDK	5
Windows Compilation	9
Linux Compilation	11
Raspberry PI 3 (Model B+)	11
OpenGL Setting	12
Raspian AARCH32 (Bullseye)	12
Raspian AARCH64 (Bookworm)	12

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	4 of 12	02

Overview

The intention of this document is to explain the structure of the GeminiSDK, including the compilation of the test applications, so that users can integrate the source code in their application.

The purpose of the source code is to demonstrate the functionality and usage of the libraries provided. Test examples for communicating with Gemini devices are accessed through the Svs5Seq library. The Svs5Seq library allows the user to interface to a Gemini Sonar device and also log and playback data in the Trittech proprietary GLF log format.

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	5 of 12	02

Installation

Windows

On Windows computers, the Gemini SDK will install into the 'C:\Program Files\Tritech\Gemini SDK' folder.

For Windows, the libraries are provided in two forms: **.lib** (*which is used for compiling the application*) and **.dll** used at runtime linking.

Linux

On Linux, the Gemini SDK can be installed into a location of your choice, as long as you have write/execute permissions. We recommend using \$HOME/work/tritechsdk.

To install the SDK, run the **InstallSDK.sh** shell script, this will copy the library files (**.so**) from the bin folder into **/usr/local/lib** which is in most **\$PATH** variables within Linux.

The install script will also copy the **arial.ttf** file into a **fonts** directory within **bin**.

Structure of the Gemini SDK

bin: This directory contains the executables of the source code provided including all dependent dlls.

doc: Contains the Gemini Network Interface Specification (*for low level interfacing to a connected Gemini Sonar Device through GeminiComms*) and Tritech proprietary Genesis Log format (*for reading from a .GLF log file as supported in the Tritech 'Genesis' software and creating GLF log files compatible with 'Genesis'*).

include: This directory contains the interface (**.h**) files for all the libraries provided. The developer needs to include header file(s) based on the library they are including in their project:


Gemini: Include header files, if interfacing to the GeminiComms.

GenesisSerializer: Include header files, if interfacing to the GLF logger.

ECDLogDataTypes: Include header files, if interfacing to the ECD logger.
Supported in older software and now deprecated.

MathsLib: Only accessed in ECDLogDataTypes library.

Svs5Seq: Include header file, if interfacing to the Svs5SeqLib. This library is a wrapper on both GeminiComms and GenesisSerializer library. The application may need to include header files from both Gemini and

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	6 of 12	02

Svs5Seq directories.

Console: ciheader.h provides the common interface header used for GLF logging. This header file is dependent on types.h and DataTypes.h (*located in the include directory*).

DataProcessor: DataProcessorAPI.h provides the common interface header used for AHRs Compass data. This header file is dependent on errorcodes.h (*located in the include directory*) and DataProcessorDefs.h

libs: contains following libs:

GeminiComms: Provides a low level communication interface with Gemini Sonar Devices. Only one instance of the application will be created if using GeminiComms because of the network port binding.


GenesisSerializer: Provides the logging functionality (*Reading and Writing log files*) in GLF file format. The GLF format is Tritech proprietary and part of the Genesis application software. The full SDK is not required to be implemented/followed if just the logging component is of interest, i.e. to record to and/or playback data from a .GLF formatted log file.

Reading log files can be used either synchronously or asynchronously. Test examples for both synchronous and asynchronous have been provided in the “src/GenesisSerializerApp” directory.

Svs5SeqLib: This is the wrapper library built on top of the GeminiComms and GenesisSerializer. Only one instance of this library is allowed. This library provides a very simple interface to the user and hides all the complexity. Please see Svs5SequencerApi.h for all supported features. The GeminiSDKGUIApp and GeminiSDKConsoleApp have been developed based on this library. “GeminiSDKConsole.cpp” has all supported features.

Note: Earlier Gemini SDKs used a lower level interface through the GeminiComms library. It is highly recommended that any earlier Gemini hardware interfaces using this older API be updated to use the SVS5Sequencer. This will facilitate GLF logging and also allow extension in future when new devices, Gemini models and features are added into Tritech’s latest generation Genesis software application.

Compass and GPS data needs to be recorded in their raw ASCII string format and for this SVS5_CONFIG_LOG_RAW_GPS and SVS5_CONFIG_LOG_RAW_COMPASS are the configuration messages that need to be passed into SequencerApi::Svs5SetConfiguration().

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	7 of 12	02

ECDLogDataTypes (Deprecated): Provides the logging functionality (*Reading and Writing log files*) in ECD file format. The ECD file format is deprecated. This library has the dependency of MathsLib (*provided in the package*) and the BOOST libraries.

Note: To add support for the ECD logger in the test application define ECD_LOGGER flag during compile time and in the application also configure SVS5SeqLib to use ECD file format. An application example has been provided in gemininetwork.cpp Please see example below:

```
bool glfLogger = true; //true for GLF, false for ECD. Default is GLF
SequencerApi::Svs5SetConfiguration(
    SequencerApi::SVS5_CONFIG_LOG
    GER, sizeof(bool),
    &glfLogger
);
```

MathsLib: Provides the basic maths functionality and only used in ECDLogDataTypes library.

DataProcessor: Provides the AHRS Compass data functionality. Logging for Compass data is only supported with the GLF file format.


src: Contains the following test examples:

GeminiSDKConsoleApp: This is the console based test application demonstrating the functionality of the Svs5Sequencer library. This application provides the functionality of live Gemini sonar data, sonar configuration (e.g. change gain, range, online/offline etc), record and playback GLF files.

GeminiSDKGuiApp: This is the GUI based test application demonstrating the functionality of the Svs5Sequencer library. This test application connects to any attached Gemini sonar heads and renders the Gemini ping data onto an OpenGL view. Basic controls are included to start the user off such as Gain, Range, record and playback. This test application requires use of the Qt Toolkit (QT V5.13 was used to create this).

Svs5LoggerApp: Svs5LogReader.cpp and Svs5LogWriter.cpp test examples are based on the Svs5Sequencer library. It provides only GLF log reader and writer functionality in an asynchronous manner.

GenesisSerializerApp: There are two test applications in this directory i.e. SynchronousLogReader and AsynchronousLogReader, demonstrating the functionality of both synchronous and asynchronous ways of reading GLF log files. These test applications are dependent on the GenesisSerializer library and interface header files are in the

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	8 of 12	02

GenesisSerializer include directory.

In synchronous mode, GLF::CreateLogFileReader API will only return when it will finish generating the index table while in asynchronous mode GLF::CreateLogFileReader API will return immediately. Further details and comments are in the source code.

Note: Logging of GLF is only supported through Svs5Sequencer library.

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	9 of 12	02

Windows Compilation

Pre-requisite

- cmake 3.15.0 or later version has been installed
- Visual Studio 2017 has been installed
- Qt V5.13 or above has been installed

Note: Please confirm that all the above tools are included in the system path, otherwise specify the full path.

Steps to compile 32-bit Release application

- Setup the Visual Studio 32-bit environment. Run the following command on the cmd prompt...

```
"C:\Program Files (x86)\Microsoft Visual
Studio\2017\Professional\VC\Auxiliary\Build\vcvars32.bat"
```

- Create a directory "mkdir c:\work\tritechsdk"
- cd c:\work\tritechsdk
- Run the following command for compiling 32-bit test application:

```
cmake -G"NMake Makefiles JOM" -DARCH=ARCH32 -
DCMAKE_BUILD_TYPE=Release "C:\Program Files\Tritech\Gemini SDK"
```

To compile in debug mode, please specify -DCMAKE_BUILD_TYPE=Debug

To enable ECD logger specify -DECD_LOGGER=yes on the

command prompt e.g. cmake -G"NMake Makefiles JOM" -

```
DARCH=ARCH32 -DCMAKE_BUILD_TYPE=Release
```

```
-ECD_LOGGER=yes "C:\Program Files\Tritech\Gemini SDK"
```

ECD logger has the dependency of boost libraries. Please update the path in the CMakeList.txt file according to your installation path.

- cmake --build .

Steps to compile 64-bit Release application

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	10 of 12	02

- Setup the Visual Studio 64-bit environment. Run the following command on the cmd prompt...

```
"C:\Program Files (x86)\Microsoft Visual
Studio2017\Professional\VC\Auxiliary\Build\vcvars64.bat"
```

- Create a directory "mkdir c:\work\tritechsdk"
- cd c:\work\tritechsdk
- Run the following command for compiling 64 bit test application:

```
cmake -G"NMake Makefiles JOM" -DARCH=ARCH64 -
DCMAKE_BUILD_TYPE=Release "C:\Program Files\Tritech\Gemini SDK"
```

To compile in debug mode, please specify -

```
DCMAKE_BUILD_TYPE=Debug To enable ECD
```

logger specify -DECD_LOGGER=yes

- cmake --build .

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	11 of 12	02

Linux Compilation

Pre-requisite

- cmake 3.15.0 or later version has been installed
- gcc 5.5.0 or later version has been installed
- Qt V5.13 or above has been installed
- Freetype v2.8.1 or above has been installed

Note: Please confirm that all the above tools are included in the system path, otherwise specify the full path.

Steps

- Create a directory “mkdir ~/work/tritechsdk”
- cd ~/work/tritechsdk
- Run the following command...

```
cmake -DARCH=ARCH64 -DCMAKE_BUILD_TYPE=Release "GeminiSDK
installation path"
```

- make

To enable ECD logger specify the –DECD_LOGGER=yes e.g.

```
cmake -DARCH=ARCH64 -DCMAKE_BUILD_TYPE=Release –
DECD_LOGGER=yes "GeminiSDK installation path"
```

Note: If you are switching between the configuration, a clean build is required. This can be achieved by removing files from ~/work/tritechsdk using “rm –rf ~/work/tritechsdk/*”


For cross compilation, set the working environment first.

Raspberry PI 3 (Model B+)

Compilation is the same as above, with the same pre-requisite requirements.

For a smoother visual within the SDK GUI Application, it is recommended to turn on the H264 compression, this can only be done with the following switch:

```
-DDISABLE_SERIAL=no
```

	Gemini Software Development Kit User Guide	Document	Page	Rev.
		0685-SDS-00006	12 of 12	02

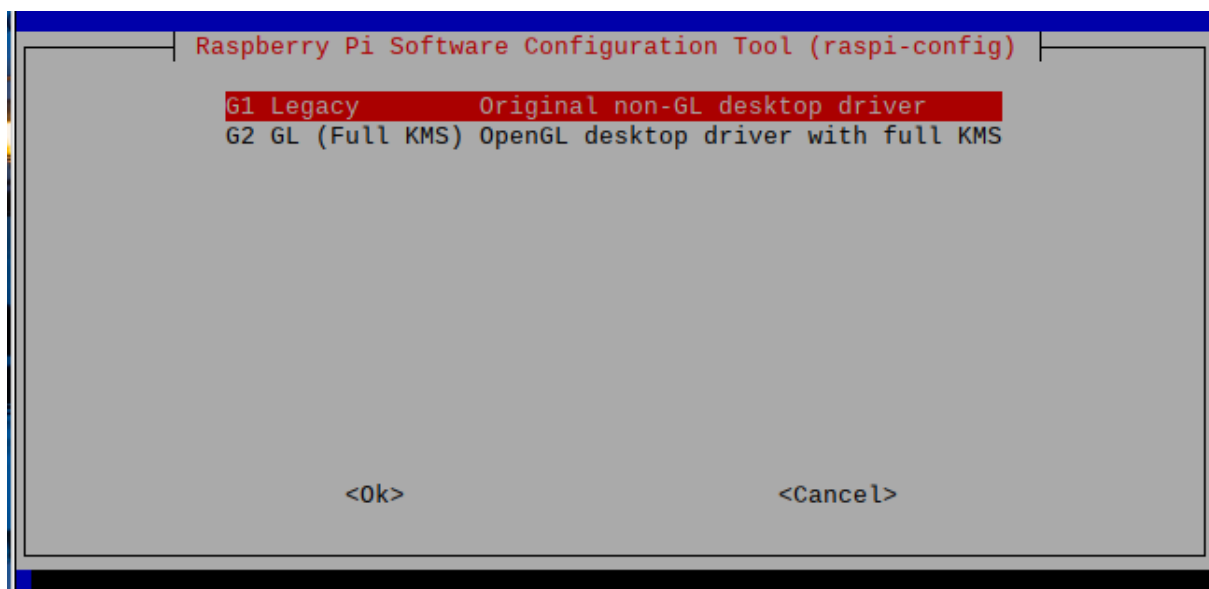
Therefore, to build on the Raspberry PI with H264 Compression available, please run the following:

```
cmake -DARCH=AARCH64 -DCMAKE_BUILD_TYPE=Release -DECD_LOGGER=no -
DDISABLE_SERIAL=no "GeminiSDK installation path"
```

OpenGL Setting

Raspian AARCH32 (Bullseye)

To use the SDK GUI application, the Legacy option is recommended. The Full KMS is untested.



Raspian AARCH64 (Bookworm)

Within the /boot/firmware/config.txt file, set the overlay to FKMS rather than KMS.

```
# Enable DRM VC4 V3D driver
dtoverlay=vc4-fkms-v3d
```