**Generic SCons**

**Design Document**

|  |  |
| --- | --- |
| Document Control: | |
| Project: | ETK: SCT\_Sconstools |
| Revision: | 0.1 |
| Last Change: |  |
| Confidence Level: | 🞏 Public 🗷 Confidential |

CONFIDENTIAL AND PROPRIETARY PROPERTY OF ADAS SIBIU - ALL RIGHTS RESERVED

|  |  |
| --- | --- |
| Document State: | |
| State: | Draft |
| Author: | Lenin Palanisamy |
| Reviewed by: |  |
| Released by: | 31-May-2016 |

Distribution List

When you check-in a new version in MKS you have to inform the persons assigned to the following roles.

|  |  |
| --- | --- |
| Sl No | Name |
| 1 | Jens Petersohn (email: Jens Petersohn/usr/cag) |
| 2 |  |
| 3 |  |

**Change sheet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Processed by** | **Description** | **Chapter(s)** |
| 0.1 | 31-May-2016 | Lenin Palanisamy | Initial draft version | All |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1 Introduction 4](#_Toc452469745)

[1.1 Purpose 4](#_Toc452469746)

[1.2 Scope 4](#_Toc452469747)

[1.3 Overview 4](#_Toc452469748)

[1.4 Reference Material 4](#_Toc452469749)

[1.5 Definitions and Acronyms 5](#_Toc452469750)

[2 GSCons Overview 6](#_Toc452469751)

[3 GSCons Architecture 7](#_Toc452469752)

[3.1 GSCons Architecture Design 7](#_Toc452469753)

[3.2 Functional Descriptions 8](#_Toc452469754)

[4 GSCons Configurations and Data Design 12](#_Toc452469755)

[4.1 GSCons Configuration 12](#_Toc452469756)

[4.2 GSCons Data Dictionary 12](#_Toc452469757)

[5 GSCons Modules Detailed Design 12](#_Toc452469758)

[6 Graphical User Interface Design 12](#_Toc452469759)

[7 Requirement Traceability Matrix 12](#_Toc452469760)

[8 Appendix 12](#_Toc452469761)

Figures

[Figure 1: GSCons Design Document Overview 4](#_Toc452469762)

[Figure 2: GSCons Architecture 7](#_Toc452469763)

[Figure 3: SCons Architecture 8](#_Toc452469764)

[Figure 4: GSCons Configuration 12](#_Toc452469765)

**Tables**

[Table 1 : GSCons Modules 11](#_Toc452469766)

# Introduction

## Purpose

This Generic SCons (GSCons) design document is intended primarily for use by developers and others working on GSCons, although it is also intended to serve as a detailed overview of GSCons for other interested parties in the algorithm components of ADC GmbH.

## Scope

GSCons is a tool that has been implemented and provided by the SCons-developers themselves and that has been published. However, the open source product “SCons” is utilized to implement “Generic SCons” as required by ADC GmbH. Even though “shared scons environment” would be more appropriate name for this tool, the term “Generic SCons” seems to have been established in the algorithm components, we continue to use this phrase.

This document does not describe the design of open source product “SCons”.

## Overview

This design document is generated from GSCons source code using the concept of reverse engineering. Doxygen tool primarily used to generate source code documentation. Firstly, all python modules of GSCons are updated with file header with functional description and function header for all methods in each module. Also this high level design document prepared to give an overview of functional groups and primary function module description of GSCons.

GSCons Source Code (\*.py)

GSCons Source Documentation

(gscons\_code\_documentation.chm)

GSCons Design Documentation

(gscons\_design\_documentation.docx)

Figure 1: GSCons Design Document Overview

## Reference Material

This document was prepared on the basis of the following documents:

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No | Name | Description/Location | Version |
| 1 | SCons design | <http://www.scons.org/doc/production/PDF/scons-design.pdf> |  |
| 2 | GSCons build environment | <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/generic%5fscons/project.pj&selection=SCons%5fbuild%5fenvironment.docx> |  |
|  |  |  |  |

If changes are made to the content of this document, you must check whether those changes affect the content of the following documents and work products:

|  |  |
| --- | --- |
| Sl No | Work products and Location |
| 1 | Release Notes document:  <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/release%5fnotes/project.pj&selection=Release%5fNotes.docx> |
| 2 | One or more modules in scons\_adas\_extensions: <http://ims-adas:7001/si/viewproject?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/scons%5fadas%5fextensions/project.pj> |
| 3 | One or more modules in scons\_common\_scripts: <http://ims-adas:7001/si/viewproject?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/scons%5fcommon%5fscripts/project.pj> |
| 4 | GSCons Source code documentation  <http://ims-adas:7001/si/viewrevision?projectName=/nfs/projekte1/REPOSITORY/Base%5fDevelopment/05%5fAlgorithm/ETK%5fEngineeringToolKit/04%5fEngineering/SCT%5fSconstools/docs/generic%5fscons/design%5fdocumentation/project.pj&selection=gscons%5fcode%5fdocumentation.chm> |

## Definitions and Acronyms

Definitions of all terms, acronyms, and abbreviations that exist to properly interpret this design document:

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Abbreviation** | **Definition** |
| 1. | ETK | Engineering Tool Kit |
| 2. | GSCons | Generic SCons tool in ETK project |
| 3. | VME | VISION MID EVE |
| 4. | VH 28 | VISION HIGH |
| 5. | ADC | Automotive Distance Control Systems |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |
| 10. | NA | Not Applicable |

# GSCons Overview

GSCons is ADC-internal build software based on SCons. This tool contains the following main features which are generic for all algorithm components:

* Build of ECU-libraries for
  + C66xx
  + C674x
  + ARP32
  + ARM Cortex A8
  + ARM Cortex A15
  + ARM Cortex M3
  + ARM Cortex M4
* Build of PC Simulation Targets
* Build of sdl- and cdl-files
* Build of Visual Studio and CCS IDEs
* QAC-analysis for all ECU-libraries
* Cantata unit tests
* Polyspace project generation
* Doxygen document generation
* Build of HiL-Outfiles for Evalboard
  + C66xx
  + C674x
  + ARP32
  + ARM Cortex A8
  + ARM Cortex A15
  + ARM Cortex M3
  + ARM Cortex M4
* VME/VH28 RAM/ROM allocation

# GSCons Architecture

## GSCons Architecture Design

GSCons is strongly associated with SCons tool which is an open source and it has also been extended to create a common build environment for algorithm components of ADC GmbH.

SCons tool

GSCons

scons\_adas\_extensions

scons\_common\_scripts

scons\_common\_config and component specific configurations (\*.scfg)

Figure 2: GSCons Architecture

The below architecture diagram gives the list of objects associated and aggregated in SCons tool. The detailed description of each object would be found in SCons documentation or in source code documentation (.CHM file) referred in the section .

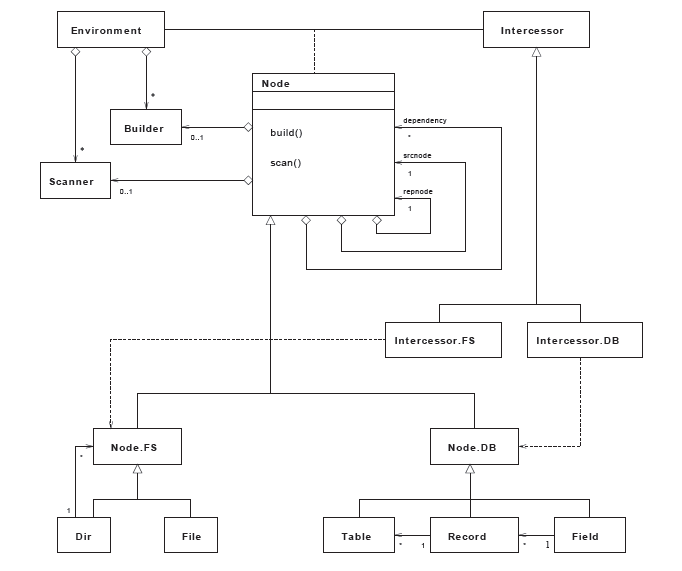


Figure 3: SCons Architecture

## Functional Descriptions

This section just gives a functional description of each module under a functional group and detailed flow, calling graph and collaboration diagram would be found at the section .

The following table gives a brief overview on the GSCons modules.

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Group** | **Module** | **Module Description** | **Module Location** |
| RAM/ROM calculation | areas.py | Script that contains the list of area used for Ram/Rom/shared allocation of Libraries | scons\_adas\_extensions |
| maps.py | Script that contains the list of maps used for Ram/Rom allocation of Libraries |
| libs\_objs.py | Script that contains the list of libraries and objects used for Ram/Rom allocation of Libraries |
| ram\_rom\_algo\_libsize.py | Script that determines RAM/ROM usage information from a library. This script file runs in a python main environment as well as a SCons script. |
| EVM HIL | arm\_gcc.py | SCons builder for generating GCC compiled ARM libraries and out files. |
| tms320dm6000.py | Script used to setup DSP build environment. It specifies the TI DSP compiler used, include directories and library directories. |
| tms470r1x.py | Script used to setup ARM build environment using TI compiler. It specifies the ARM TI compiler used, include directories and library directories. |
| Auto release | autorel\_functions.py | Script that contains functions for the automatic release sequence and to generate code handover script. |
| autorel\_main.py | Script that controls the automatic release sequence and to generate code handover. |
| autorel\_mks\_util.py | Script that contains MKS functions for the Auto Release script. |
| autorel\_word\_util.py | Script that contains utility functions for generation and modification of Microsoft Word files. |
| Doxygen | doxygen.py | Script used to generate HTML documentation from Doxygen doxyfile. |
|  | eclipse\_cdt.py | Script used to generate Eclipse project for unit test. |
| Code Composer Studio | eclipse\_cdt\_ccs5.py | Script used to generate Eclipse projects for ECU-Lib builds and SIM-HiL builds |
| MS Visual studio | msvc-addon.py | Script that do the modifications to the Microsoft Visual Studio builder which is already supplied by SCons. |
| msvs-patched.py | Script that do the modifications to the Microsoft Visual Studio builder which is already supplied by SCons. |
| vsscript.py | Script that contains Additional builder to patch Visual Studio Resource (.rc) file. |
| Finger Print | fingerprint.py | Script used to generate the fingerprint files |
| PDO | pdo.py | Script used to generate sdl- and cdl-files via pdo-tool and sdlcompiler |
| GSCons Profiling | profile.py | Script that allows using a profiling mode for Generic Scons |
| QAC | qac.py | Script that controls the generation of QAC analysis and reports |
| ARP32 | ti\_eve\_arp32.py | Script used to setup ARP32 build environment. It specifies the TI compiler used, include directories and library directories. |
| GSCons startup | help\_menu.py | Script that contains functions for the creation of a help menu |
| sconscript\_setup.py | Script that setup shared scons scripts to a project and copy them into folder structure |
| sconstruct\_helpers.py | Script that contains helpful functions used by Scons build environment |
| Cantata Module tests | unittest.py | Script that setup the environment for unit test |
| Algo Libraries | lib\SConscript.py | SConscript to build sdl-files, cdl-files for pc simulation and ECU- and PC libraries | scons\_common\_scripts |
| lib\_env\_ti\_<core>\SConscript.py | SConscript to set up the build environment for the build of <core>-Libraries. It also generates IDE'S for ECU-Libraries. Where <core> is ARP32, C66xx, C674x, A15, A8, M3/M4 |
| Doxygen | convert\_xml.py | Python script for xml conversion for Doxygen documentation |
| EVM HIL | evm\_hil\_env\generate\_DebugServerScripting.py | Script used to generate \_CopyDSScriptLoadCmdToClipboard.bat and DebugServerScripting.js which are used to run the debugger and load out files to one or more processor cores. |
| evm\_hil\_env\SConscript.py | SConscript to call the build of EVM HIL out builds for EVM platforms VME and VH28 |
| evm\_hil\_env2\SConscript.py | SConscript to call the build of EVM HIL out builds for EVM platforms VME and VH28, where VME requires Cortex\_a8\_Only  WITHOUT C674x |
| evm\_hil\_ti\_<core>\SConscript.py | SConscript to build EVM HIL out file for EVM HIL targets |
| evm\_hil\_ti\_<core>\SConscript\_<core>.py | SConscript to create build env for <core> out file. where <core> is build targets TI\_C674X, TI\_C66XX, TI\_CORTEX\_A8, TI\_CORTEX\_A15, SIM ands TI\_ARP32 |
| Finger Print | SConscript\_fingerprint.py | SConscript to build fingerprint |
| Cantata Module tests | SConscript.py | SConscript to create Cantata build environment |
| SConscript\_unittests.py | SConscript to build Cantata unit tests and generate reports |
| Algo Simulation | sim\_adapt\SConscript.py | SConscript to build sim adapter (Adapter Plugin: dll ) |
| sim\_env\SConscript\_simenv.py | SConscript to create build environment for pc simulation targets |
| sim\_exe\SConscript.py | SConscript to build sim adapter executable (EXE) |
| sim\_lib\SConscript.py | SConscript to build sim adapter Static library (LIB) |
| sim\_sconscript\SConscript\_sim.py | SConscript to build simulation DLLs and Visual studio IDE. It calls all other SConscript related to PC simulation targets. Those SConscript's are defined in sconstruct\_config.scfg. |
| sim\_swc\SConscript.py | SConscript to build Sim HIL wrapper (SWC-Plugin: dll, sdl, cdl, res-file) |
| sim\_swc\_vis\SConscript.py | SConscript to build sim Visualization Plugin: dll, res-file |
| Auto release | autorel\SConscript.py | SConscript to execute automatic release sequence and generate code handover script. |
| GSCons startup | SConstruct | Scons utility script that contains the main entry point for build construction for each component. By default, the SCons utility searches for a file named SConstruct, Sconstruct or sconstruct (in that order) in the current directory, and reads its configuration from the first file found |

Table 1 : GSCons Modules

# GSCons Configurations and Data Design

## GSCons Configuration

SConstruct is the main script that is executed by SCons. This script triggers the build of one or more targets by calling SConscript files. All .scfg-files are text files holding python variables that will be read by SConstruct and SConscript.py before building targets. The content of those .scfg-files is component-specific in the majority of cases, so generally they are non-shared. Lists of source files and include-paths are only two examples for the possible content of a configuration file. The separation of build information into SConstruct, SConscript.py’s, further .py-files, and .scfg files is chosen to make the scripts of Generic SCons as generic.

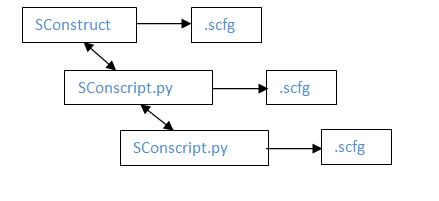


Figure 4: GSCons Configuration

## GSCons Data Dictionary

All major source data of GSCons are defined in common\_config.scfg which provides the common definition of directory paths of all project and default compiler and development tool settings. Those data are would be processed while executing one or more build targets. The generic directory structure should be adopted as given in GSCons build environment (Ref. ) to integrate GSCons for algorithm component.

# GSCons Modules Detailed Design

GSCons Source code documentation is generated from all modules of GSCons and SCons using Doxygen tool which is detailed in the CHM file referenced in section 1.4

# Graphical User Interface Design

NA

# Requirement Traceability Matrix

There is no formal or in-informal requirement document found while generating this document. Hence, the traceability matrix would not be generated.

# Appendix

None