gen_avr8 Documentation

Release https://github.com/vroncevic/gen_avr8/releases

Vladimir Roncevic <elektron.ronca@gmail.com>

Apr 27, 2022

Contents

gen_avr8 is toolset for generation of AVR8 project skeleton for development embedded applications.

Developed in python code.

The README is used to introduce the tool modules and provide instructions on how to install the tool modules, any machine dependencies it may have and any other information that should be provided before the modules are installed.

CHAPTER 1

gen_avr8

1.1 gen_avr8 package

1.1.1 Subpackages

1.1.1.1 gen_avr8.pro package

Submodules

gen_avr8.pro.mcu_selector module

Module mcu_selector.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class MCUSelector with attribute(s) and method(s). Selecting MCU target for generating process of project structure.

class gen_avr8.pro.mcu_selector.**MCUSelector**(*verbose=False*)
Bases: ats_utilities.config_io.base_check.FileChecking

Defined class MCUSelector with attribute(s) and method(s). Selecting MCU target for generating process of project structure. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
MCU_LIST - configuration file with MCU list.

__mcu_list - MCU list.

methods

__init__ - initial constructor.
get_mcu_list - getter for MCU list object.
choose_mcu - select MCU target.
__str__ - dunder method for MCUSelector.

GEN_VERBOSE = 'GEN_AVR8::PRO::MCU_SELECTOR'

MCU_LIST = '/../conf/mcu.yaml'

VERBOSE = 'ATS_UTILITIES'

choose_mcu(*verbose=False*)
Select MCU target.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns MCU name | None.

Return type <str> | <NoneType>

Exceptions None

get_mcu_list()
Getter for MCU list object.

Returns MCU list | None.

Return type <list> | <NoneType>

Exceptions None

gen_avr8.pro.module_type module

Module module_type.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class ModuleType with attribute(s) and method(s). Check module type (it can be source module | build module).

class gen_avr8.pro.module_type.**ModuleType**
Bases: object

Defined class ModuleType with attribute(s) and method(s). Check module type (it can be source module | build module). It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
SOURCE - list of expected source extensions.
BUILD - list of expected build extensions/files.

methods

pre_process_module - process module name.
is_source_module - check is source module.
is_build_module - check is build module.
__str__ - dunder method for ModuleType.

BUILD = ['.mk', 'Makefile']

GEN_VERBOSE = 'GEN_AVR8::PRO::MODULE_TYPE'

SOURCE = ['.c', '.h']

classmethod is_build_module(*module*)
Check is build module.

Parameters **module** (*<str>*) – module name.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

classmethod is_source_module(*module*)
Check is source module.

Parameters **module** (*<str>*) – module name.

Returns boolean status, True | False.

Return type <bool>

Exceptions None

classmethod pre_process_module(*pro_type*, *pro_name*, *module*, *verbose=False*)
Process module name.

Parameters

- **pro_type** (*<str>*) – project type.
- **pro_name** (*<str>*) – project name.
- **module** (*<str>*) – module name.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns processed module name.

Return type <str>

Exceptions None

gen_avr8.pro.osc_selector module

Module osc_selector.py

Info Defined class OSCSelector with attribute(s) and method(s). Selecting FOSC for generating process of project structure.

class gen_avr8.pro.osc_selector.**OSCSelector**(*verbose=False*)
Bases: ats_utilities.config_io.base_check.FileChecking

Defined class OSCSelector with attribute(s) and method(s). Selecting FOSC for generating process of project structure. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
FOSC_LIST - configuration file with FOSC list.
__fosc_list - FOSC list.

methods

__init__ - initial constructor.
get_fosc_list - getter for FOSC list object.
choose_osc - select FOSC for target.
__str__ - dunder method for OSCSelector.

FOSC_LIST = '/../conf/fosc.yaml'

GEN_VERBOSE = 'GEN_AVR8::PRO::OSC_SELECTOR'

VERBOSE = 'ATS_UTILITIES'

choose_osc(*verbose=False*)
Select FOSC for target.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns FOSC | None.

Return type <str> | <NoneType>

Exceptions None

get_fosc_list()
Getter for FOSC list object.

Returns FOSC configuration | None.

Return type <list> | <NoneType>

Exceptions None

gen_avr8.pro.read_template module

Module read_template.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class ReadTemplate with attribute(s) and method(s). Created API for read a template file and return a content.

class gen_avr8.pro.read_template.**ReadTemplate**(*verbose=False*)
Bases: ats_utilities.config_io.base_check.FileChecking

Defined class ReadTemplate with attribute(s) and method(s). Created API for read a template file and return a content. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
FORMAT - file format for template.

methods

__init__ - initial constructor.
read - read template file.
__str__ - dunder method for ReadTemplate.

FORMAT = 'template'

GEN_VERBOSE = 'GEN_AVR8::PRO::READ_TEMPLATE'

VERBOSE = 'ATS_UTILITIES'

read(*template_file*, *verbose=False*)
Read template file.

Parameters

- **template_file** (*<str>*) – template file path.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns template content for module | None.

Return type <str> | <NoneType>

Exceptions ATSTypeError | ATSBadCallError

gen_avr8.pro.template_dir module

Module template_dir.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class TemplateDir with attribute(s) and method(s). Created API for checking project template directory.

class gen_avr8.pro.template_dir.**TemplateDir**
Bases: object

Defined class TemplateDir with attribute(s) and method(s). Created API for checking project template directory. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
CONF_DIR - configuration directory path.
TEMPLATE_DIR - template directory path.

methods

check_dir - check project directory.
setup_conf_dir - setup configuration directory.
setup_template_dir - setup template directory.
__str__ - dunder method for TemplateDir.

CONF_DIR = '/../conf/'

GEN_VERBOSE = 'GEN_AVR8::PRO::TEMPLATE_DIR'

TEMPLATE_DIR = '/../conf/template/'

classmethod check_dir(*target_dir*, *verbose=False*)
Checking project directory.

Parameters

- **target_dir** (*<str>*) – target directory to be checked.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True directory ok | False.

Return type <bool>

Exceptions None

classmethod setup_conf_dir(*verbose=False*)
Setup configuration directory.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns template directory | None.

Return type <str> | <NoneType>

Exceptions None

classmethod setup_template_dir(*verbose=False*)
Setup template directory.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns template directory | None.

Return type <str> | <NoneType>

Exceptions None

gen_avr8.pro.template_type module

Module template_type.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class TemplateType with attribute(s) and method(s). Created API for checking project template type and template structure.

class gen_avr8.pro.template_type.**TemplateType**
Bases: object

Defined class TemplateType with attribute(s) and method(s). Created API for checking project template type and template structure. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
APP_TEMPLATE - application type of project.
LIB_TEMPLATE - library type of project.
TEMPLATE_TYPE - project template structures.

methods

check_template_type - check project template type.
setup_template_type - setup template type (app | lib).
__str__ - dunder method for TemplateType.

APP_TEMPLATE = 'app'

GEN_VERBOSE = 'GEN_AVR8::PRO::TEMPLATE_TYPE'

LIB_TEMPLATE = 'lib'

TEMPLATE_TYPE = {'app': 'project_app.yaml', 'lib': 'project_lib.yaml'}

classmethod check_template_type(*template_type*, *verbose=False*)
Check project template type (App | Lib).

Parameters

- **template_type** (*<str>*) – project template type.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True project type for ok | False.

Return type <bool>

Exceptions ATSTypeError | ATSBadCallError

classmethod setup_template_type(*template_type*, *verbose=False*)
Setup template type (App | Lib).

Parameters

- **template_type** (*<str>*) – project template type.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns project type (app | lib) | None.

Return type <str> | <NoneType>

Exceptions ATSTypeError | ATSBadCallError

gen_avr8.pro.write_template module

Module write_template.py

Copyright Copyright (C) 2018 Vladimir Roncevic <elektron.ronca@gmail.com> gen_avr8 is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. gen_avr8 is

distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Info Defined class WriteTemplate with attribute(s) and method(s). Created API for write template content with parameters.

class gen_avr8.pro.write_template.**WriteTemplate**(*verbose=False*)
Bases: ats_utilities.config_io.base_check.FileChecking

Defined class WriteTemplate with attribute(s) and method(s). Created API for Write template content with parameters. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__pro_dir - current project directory.

methods

__init__ - initial constructor.
pro_dir - property methods for set/get operations.
check_module - check project module.
write - write a template content to a project module.
__str__ - dunder method for WriteTemplate.

GEN_VERBOSE = 'GEN_AVR8::PRO::WRITE_TEMPLATE'

VERBOSE = 'ATS_UTILITIES'

check_module(*module*, *verbose=False*)
Check project module.

Parameters

- **module** (*<str>*) – module name.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns 'build' | 'source'| None (wrong module name).

Return type <str> | <NoneType>

Exceptions ATSTypeError | ATSBadCallError

pro_dir
Property method for getting project dir.

Returns project dir | None.

Return type <str> | <NoneType>

Exceptions None

write(*project_data*, *verbose=False*)
Write a template content to a project module.

Parameters

- **project_data** (*<dict>*) – project data.
- **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type <bool>

Exception ATSTypeError | ATSBadCallError

Module contents

Module __init__.py

Info Defined class AVR8Setup with attribute(s) and method(s). Generate AVR project skeleton.

class gen_avr8.pro.**AVR8Setup**(*verbose=False*)
Bases: object

Defined class AVR8Setup with attribute(s) and method(s). Generate AVR project skeleton. It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
__mcu_sel - MCU selector API.
__fosc_sel - FOSC selector API.
__reader - reader API.
__writer - writer API.
__project_setup - project setup.

methods

__init__ - initial constructor.
project_setup - property methods for set operations.
gen_pro_setup - generate project skeleton.
__str__ - dunder method for AVR8Setup.

GEN_VERBOSE = 'GEN_AVR8::PRO::AVR8SETUP'

gen_pro_setup(*verbose=False*)
Generate AVR8 project setup.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

project_setup(*project_name*, *project_type*, *verbose=False*)
Setter for project setup.

Parameters

- **project_name** (*<str>*) – project name.
- **project_type** (*<str>*) – project type.
- **verbose** (*<bool>*) – enable/disable verbose option.

Exceptions ATSTypeError | ATSBadCallError

1.1.2 Module contents

Module __init__.py

Info Defined class GenAVR8 with attribute(s) and method(s). Load a base info, create an CLI interface and run operation(s).

class gen_avr8.**GenAVR8**(*verbose=False*)

Bases: ats_utilities.cli.cfg_cli.CfgCLI

Defined class GenAVR8 with attribute(s) and method(s). Load a base info, create an CLI interface and run operation(s). It defines:

attributes

GEN_VERBOSE - console text indicator for process-phase.
CONFIG - tool info file path.
LOG - tool log file path.
LOGO - logo for splash screen.
OPS - list of tool options.
logger - logger object API.

methods

__init__ - initial constructor.
process - process and run tool option.
__str__ - dunder method for GenAVR8.

CONFIG = '/conf/gen_avr8.cfg'

GEN_VERBOSE = 'GEN_AVR8'

LOG = '/log/gen_avr8.log'

LOGO = '/conf/gen_avr8.logo'

OPS = ['-g', '--gen', '-t', '--type', '-v', '--verbose', '--version']

VERBOSE = 'ATS_UTILITIES'

process(*verbose=False*)

Process and run operation.

Parameters **verbose** (*<bool>*) – enable/disable verbose option.

Returns boolean status, True (success) | False.

Return type <bool>

Exceptions None

CHAPTER 2

Installation

Navigate to release page download and extract release archive.

To install **gen_avr8** type the following

```
tar xvzf gen_avr8-x.y.z.tar.gz
cd gen_avr8-x.y.z/
#python2
python setup.py install_lib
python setup.py install_data
python setup.py install_egg_info
#python3
python3 setup.py install_lib
python3 setup.py install_data
python3 setup.py install_egg_info
```

You can use Docker to create image/container, or You can use pip to install

```
#python2
pip install gen-avr8
#python3
pip3 install gen-avr8
```

CHAPTER 3

Usage

Create AVR8 Project Blink, MCU/FOSC will be selected during generation process

```
python gen_avr8_run.py -g Blink -t app
```

CHAPTER 4

Dependencies

gen_avr8 tool-module requires other modules and libraries

- ats-utilities - Python App/Tool/Script Utilities

CHAPTER 5

Supported mcus

Current list of supported microcontrollers

```
attiny2313    atmega128      at90s2313
attiny24      atmega1280     at90s2333
attiny25      atmega1281     at90s4414
attiny26      atmega1284p    at90s4433
attiny261     atmega16       at90s4434
attiny44      atmega163      at90s8515
attiny45      atmega164p     at90s8535
attiny461     atmega165
attiny84      atmega165p
attiny85      atmega168
attiny861     atmega169
              atmega169p
              atmega2560
              atmega2561
              atmega32
              atmega324p
              atmega325
              atmega3250
              atmega329
              atmega3290
              atmega32u4
              atmega48
              atmega64
              atmega640
              atmega644
              atmega644p
              atmega645
              atmega6450
              atmega649
              atmega6490
              atmega8
              atmega8515
```

(continues on next page)

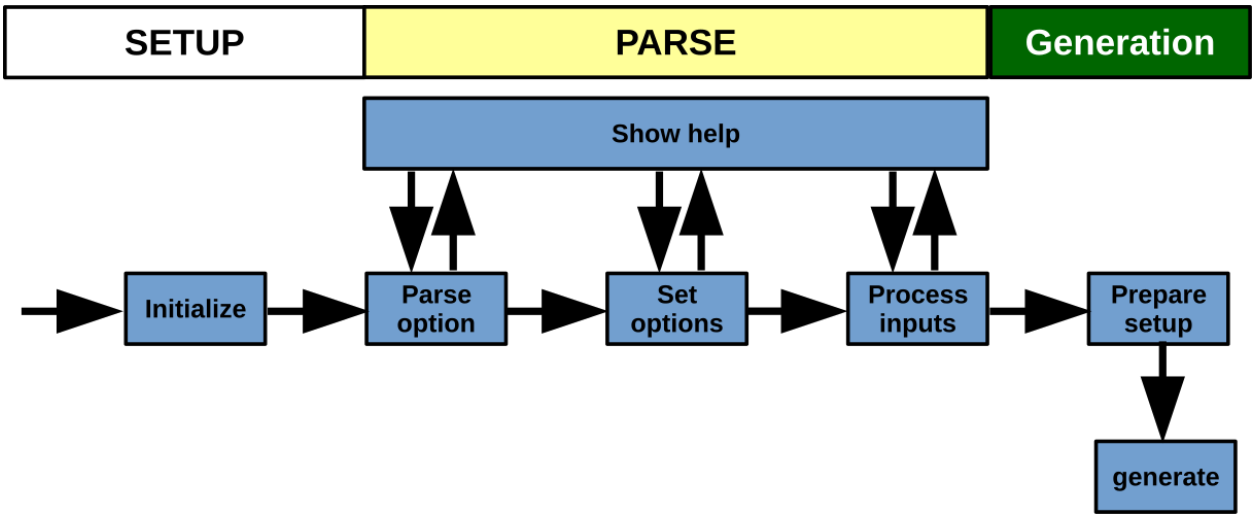
(continued from previous page)

```
atmega8535
atmega88
```

CHAPTER 6

Generation flow of project setup

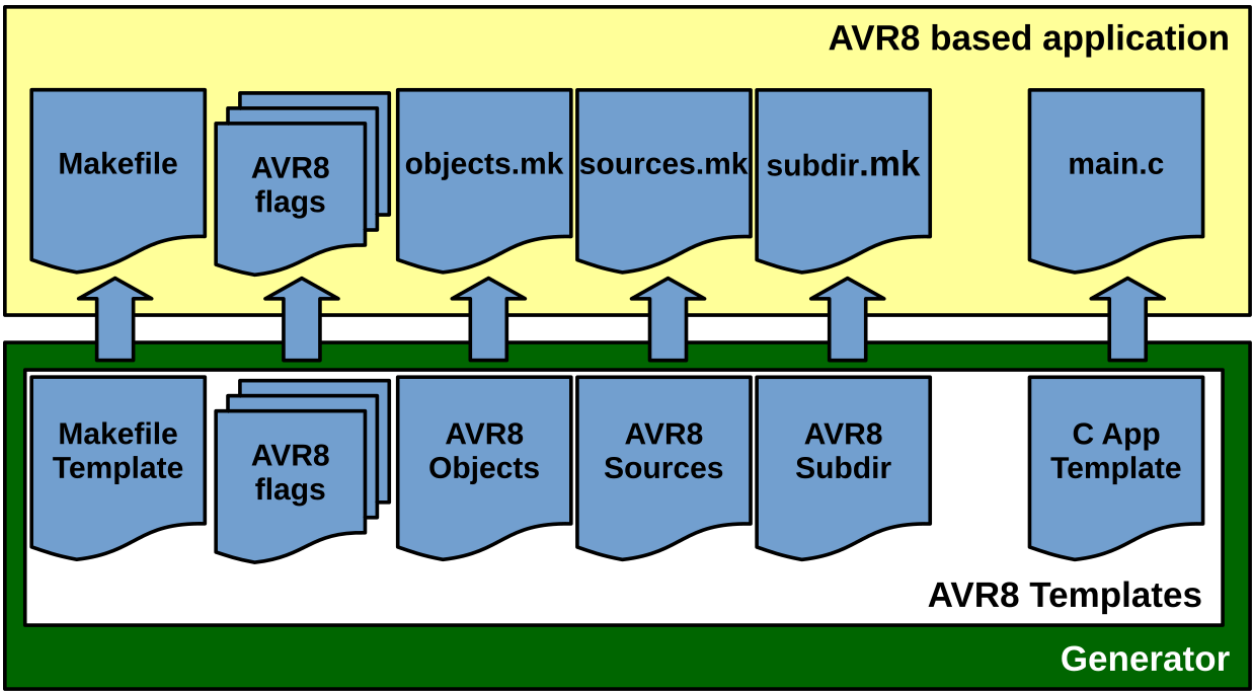
Base flow of generation process



CHAPTER 7

Tool structure

gen_avr8 is based on Template mechanism



Generator structure

```
gen_avr8/
├── conf/
|   ├── gen_avr8.logo
|   ├── fosc.yaml
|   ├── gen_avr8.cfg
|   ├── gen_avr8_util.cfg
```

(continues on next page)

(continued from previous page)

```
│   ├── mcu.yaml
│   ├── project_app.yaml
│   ├── project_lib.yaml
│   └── template/
│       ├── app/
│       │   ├── cflags.template
│       │   ├── csflags.template
│       │   ├── Makefile.template
│       │   ├── module.template
│       │   ├── objects.template
│       │   ├── ocflags.template
│       │   ├── odflags.template
│       │   ├── sources.template
│       │   ├── subdir.template
│       │   └── tools.template
│       └── lib/
│           ├── aflags.template
│           ├── avr_lib_c.template
│           ├── avr_lib_h.template
│           ├── cflags.template
│           ├── csflags.template
│           ├── Makefile.template
│           ├── objects.template
│           ├── ocflags.template
│           ├── odflags.template
│           ├── sources.template
│           ├── subdir.template
│           └── tools.template
├── __init__.py
├── log/
│   └── gen_avr8.log
├── pro/
│   ├── __init__.py
│   ├── mcu_selector.py
│   ├── module_type.py
│   ├── osc_selector.py
│   ├── read_template.py
│   ├── template_dir.py
│   ├── template_type.py
│   └── write_template.py
└── run/
    └── gen_avr8_run.py
```

CHAPTER 8

Copyright and licence

Lets help and support PSF.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

Python Module Index

g

Index

M

O

P

R

S

T

V

W