pyavrutils Documentation

Release 0.1.2

ponty

February 10, 2013

CONTENTS

pyavrutils

Date February 10, 2013

PDF pyavrutils.pdf

Contents:

pyavrutils is a Python library that can build AVR and arduino code at runtime.

Links:

- home: https://github.com/ponty/pyavrutils
- documentation: http://ponty.github.com/pyavrutils

Features:

- python wrapper for avr-gcc, avr-size, arscons
- build files or strings (strings are saved as temp files)
- MCU list
- get code size using avr-size
- avr-gcc default is optimized for size
- supported python versions: 2.6, 2.7

Known problems:

- temp files are not removed
- arscons does not perfectly matches the Arduino build process

Possible usage:

- experimenting with flags
- building from paver
- unit tests
- building arduino code without GUI

CHAPTER

ONE

BASIC USAGE

```
>>> from pyavrutils import AvrGcc
>>> cc = AvrGcc()
>>> cc.build('int main(){}')
>>> cc.size().program_bytes
66
```

```
>>> from pyavrutils import Arduino
>>> cc = Arduino()
>>> cc.mcu = 'atmega8'
>>> cc.build('void setup(){};void loop(){}')
>>> cc.size().program_bytes
1612
```

CHAPTER

TWO

INSTALLATION

2.1 General

- arscons is already included in the library
- install pip
- install gcc-avr
- install scons (only for arscons)
- install arduino (only for arscons)
- install the program:

if you have setuptools installed:

```
# as root
pip install pyavrutils
```

2.2 Ubuntu

```
sudo apt-get install python-pip
sudo apt-get install binutils-avr
sudo apt-get install gcc-avr
sudo apt-get install scons
sudo apt-get install arduino
sudo pip install pyavrutils
# optional for examples:
sudo pip install entrypoint2
```

2.3 Uninstall

using pip:

```
# as root
pip uninstall pyavrutils
```

CHAPTER

THREE

USAGE

3.1 AVR

```
>>> from pyavrutils import AvrGcc
>>> cc = AvrGcc(mcu='atmega48')
>>> cc.targets
[u'avr1', u'avr2', u'avr25', u'avr3', u'avr31', u'avr35', u'avr4', u'avr5', u'avr51', u'avr6', u'a
>>> cc.options_generated()
['avr-gcc', '-Df_cpu=4000000', '-mmcu=atmega48', '--std=gnu99', '-Wl,--relax', '-Wl,--gc-sections
>>> cc.build('int main(){}')
>>> cc.output
'/tmp/pyavrutils_kkuDCB.elf'
>>> cc.size()
AvrSize <prog:80 bytes 2.0% mem:0 bytes 0.0% >
>>> cc.size().program_bytes
80
>>> cc.mcu='atmega168'
>>> cc.options_generated()
['avr-gcc', '-Df_cpu=4000000', '-mmcu=atmega168', '--std=gnu99', '-Wl,--relax', '-Wl,--gc-section
>>> cc.build('int main(){}')
>>> cc.output
'/tmp/pyavrutils_kkuDCB.elf'
>>> cc.size().program_bytes
132
```

3.2 arduino

```
>>> from pyavrutils import Arduino
>>> cc = Arduino(board='mini')
>>> cc.build('void setup(){};void loop(){}')
>>> cc.output
path('/tmp/pyavrutils_bJXXaG/pyavrutils_tSXb4Y/pyavrutils_tSXb4Y.elf')
>>> cc.size()
AvrSize <prog:436 bytes 2.7% mem:9 bytes 0.9% >
>>> cc.size().program_bytes
436
>>> cc.board='pro'
>>> cc.build('void setup(){};void loop(){}')
>>> cc.output
path('/tmp/pyavrutils_brP9G4/pyavrutils_asCNCd/pyavrutils_asCNCd.elf')
>>> cc.size().program_bytes
460
>>> cc.warnings
[u'build/core/IPAddress.h:51:55: warning: dereferencing type-punned pointer will break strict-ali
```

display warnings on console:

```
$ python -m pyavrutils.cli.arduino.build /usr/share/arduino/examples/4.Communication/Dimmer/Dimme
backend: arscons
MCU: atmega168

=============================================
SIZE
=============================================
program: 2406
data: 192

=============================================
WARNINGS
=============================================

core
-------------------
build/core/IPAddress.h:51:55: warning: dereferencing type-punned pointer will break strict-aliasi
build/core/IPAddress.h:52:108: warning: dereferencing type-punned pointer will break strict-alias
build/core/IPAddress.h:52:75: warning: dereferencing type-punned pointer will break strict-aliasi
build/core/Tone.cpp:108:45: warning: only initialized variables can be placed into program memory

lib
-------------------


sketch
-------------------
```

CHAPTER

FOUR

EXAMPLES

4.1 Simple example

Example program:

```
'''
test minimum program size with different optimizations
'''

from pyavrutils import AvrGcc
from entrypoint2 import entrypoint

cc = AvrGcc()
code = 'int main(){}'


def test():
    print '    compiler option:', ' '.join(cc.options_generated())
    cc.build(code)
    print '    program size =', cc.size().program_bytes


@entrypoint
def main():
    print 'compiler version:', cc.version()
    print 'code:', code
    print
    print 'no optimizations::'
    print
    cc.optimize_no()
    test()
    print
    print 'optimize for size::'
    print
    cc.optimize_for_size()
    test()
```

Output:

```
$ python -m pyavrutils.examples.simple
compiler version: 4.5.3
code: int main(){}

no optimizations::

    compiler option: avr-gcc -Df_cpu=4000000 -mmcu=atmega168 --std=gnu99
    program size = 150
```

```
optimize for size::

    compiler option: avr-gcc -Df_cpu=4000000 -mmcu=atmega168 --std=gnu99 -Wl,--relax -Wl,--gc-sect
    program size = 132
```

4.2 Test size with unused code

Example program:

```
from pyavrutils.avrgcc import AvrGcc
from entrypoint2 import entrypoint

cc = AvrGcc()


def test_option(sources, optimization, gc_sections=0, ffunction_sections=0):
    print 'optimization =', optimization,
    print 'gc_sections =', gc_sections,
    print 'ffunction_sections =', ffunction_sections,
    print

    cc.optimization = optimization
    cc.gc_sections = gc_sections
    cc.ffunction_sections = ffunction_sections
    try:
        cc.build(sources)
        size = cc.size()
        print 'program, data =', str(size.program_bytes).rjust(8), ',', str(size.data_bytes).rjust
    except:
        print  'compile error'


def test(sources):
    print 'sources:', sources
    test_option(sources, 0)
    test_option(sources, 's', 0)
    test_option(sources, 's', 1)
    test_option(sources, 's', 1, 1)


@entrypoint
def main():
    cc.optimize_no()
    print  'compiler version:', cc.version()
    print  'compiler options:', ' '.join(cc.options_generated())
    print
    print 'minimum size'
    print 20 * '='
    test(['int main(){}'])

    print
    print 'unused function in separate file'
    print 40 * '='
    test(['int main(){}', 'int f(){return 2;}'])

    print
    print 'unused function in the same file'
    print 40 * '='
    test(['int main(){}; int f(){return 2;}'])
```

Output:

```
$ python -m pyavrutils.examples.deadcode
compiler version: 4.5.3
compiler options: avr-gcc -Df_cpu=4000000 -mmcu=atmega168 --std=gnu99

minimum size
====================
sources: ['int main(){}']
optimization = 0 gc_sections = 0 ffunction_sections = 0
program, data =      150 ,         0
optimization = s gc_sections = 0 ffunction_sections = 0
program, data =      138 ,         0
optimization = s gc_sections = 1 ffunction_sections = 0
program, data =      138 ,         0
optimization = s gc_sections = 1 ffunction_sections = 1
program, data =      138 ,         0

unused function in separate file
========================================
sources: ['int main(){}', 'int f(){return 2;}']
optimization = 0 gc_sections = 0 ffunction_sections = 0
program, data =      168 ,         0
optimization = s gc_sections = 0 ffunction_sections = 0
program, data =      144 ,         0
optimization = s gc_sections = 1 ffunction_sections = 0
program, data =      138 ,         0
optimization = s gc_sections = 1 ffunction_sections = 1
program, data =      138 ,         0

unused function in the same file
========================================
sources: ['int main(){}; int f(){return 2;}']
optimization = 0 gc_sections = 0 ffunction_sections = 0
program, data =      168 ,         0
optimization = s gc_sections = 0 ffunction_sections = 0
program, data =      144 ,         0
optimization = s gc_sections = 1 ffunction_sections = 0
program, data =      144 ,         0
optimization = s gc_sections = 1 ffunction_sections = 1
program, data =      138 ,         0
```

Conclusions:

- both `gc_sections` and `ffunction_sections` should be used

4.3 Test size with delay.h

Example program:

```
from entrypoint2 import entrypoint
from pyavrutils.avrgcc import AvrGcc, AvrGccCompileError

templ = '''
#include <avr/io.h>
#include <util/delay.h>
int main()
{
    %s;
    return 0;
}
'''
```

```
cc = AvrGcc()
cc.optimize_no()
print 'compiler version:', cc.version()
print


def test(snippet, option=''):
    print snippet.ljust(33),
    cc.options_extra = option.split()
    print 'compiler option:', option, '\t',
    try:
        cc.build([templ % snippet])
        size = cc.size()
        print 'program, data =', str(size.program_bytes).rjust(8), ',', str(size.data_bytes).rjus
    except AvrGccCompileError as e:
        print 'compile error'


@entrypoint
def main():
    cc.optimization = 0

    test('_delay_ms(4)', '-O0')
    test('_delay_ms(4)', '-O1')
    test('_delay_ms(4)', '-O2')
    test('_delay_ms(4)', '-O3')
    test('_delay_ms(4)', '-Os')
```

Output:

```
$ python -m pyavrutils.examples.delaysize
compiler version: 4.5.3

_delay_ms(4)                      compiler option: -O0        program, data =     3266 ,
_delay_ms(4)                      compiler option: -O1        program, data =      150 ,
_delay_ms(4)                      compiler option: -O2        program, data =      150 ,
_delay_ms(4)                      compiler option: -O3        program, data =      150 ,
_delay_ms(4)                      compiler option: -Os        program, data =      150 ,
```

Conclusions:

- parameter should be constant
- optimization should be 1, 2, 3 or s

4.4 Test size with program space

Example program:

```
from pyavrutils.avrgcc import AvrGcc
from entrypoint2 import entrypoint

templ = '''
#include <avr/io.h>
#include <avr/pgmspace.h>
int main()
{
    %s;
    return 0;
}
'''
```

```
cc = AvrGcc()
cc.optimization = 0
print 'compiler version:', cc.version()
print 'compiler options:', ' '.join(cc.options_generated())
print


def test(snippet):
    print snippet, '\t\t',
    try:
        cc.build([templ % snippet])
        size = cc.size()
        print 'program, data =', str(size.program_bytes).rjust(8), ',', str(size.data_bytes).rjus
    except:
        print 'compile error'


def test_comb(s):
    words = 'static const PROGMEM'.split()

    def choice(i):
        return [words[i], ' ' * len(words[i])]

    for s0 in choice(0):
        for s1 in choice(1):
            for s2 in choice(2):
#                   for s3 in choice(3):
                    test('%s %s char s[] %s = "%s"' % (s0, s1, s2, s))


@entrypoint
def main():
    test_comb("12345")
    test_comb("1234512345")
```

Output:

```
$ python -m pyavrutils.examples.pgmspace
compiler version: 4.5.3
compiler options: avr-gcc -Df_cpu=4000000 -mmcu=atmega168 --std=gnu99 -Wl,--relax -Wl,--gc-sectio

static const char s[] PROGMEM = "12345"             program, data =      144 ,          0
static const char s[]         = "12345"             program, data =      166 ,          0
static       char s[] PROGMEM = "12345"             program, data =      144 ,          0
static       char s[]         = "12345"             program, data =      166 ,          0
       const char s[] PROGMEM = "12345"             program, data =      220 ,          6
       const char s[]         = "12345"             program, data =      220 ,          6
             char s[] PROGMEM = "12345"             program, data =      220 ,          6
             char s[]         = "12345"             program, data =      220 ,          6
static const char s[] PROGMEM = "1234512345"             program, data =      144 ,          0
static const char s[]         = "1234512345"             program, data =      166 ,          0
static       char s[] PROGMEM = "1234512345"             program, data =      144 ,          0
static       char s[]         = "1234512345"             program, data =      166 ,          0
       const char s[] PROGMEM = "1234512345"             program, data =      232 ,         12
       const char s[]         = "1234512345"             program, data =      232 ,         12
             char s[] PROGMEM = "1234512345"             program, data =      232 ,         12
             char s[]         = "1234512345"             program, data =      232 ,         12
```

Conclusions:

- constant string should be static or global
- `const` has no effect on size

- PROGMEM should be used

4.5 Test minimum size

Example program:

```
'''
test minimum program size with all MCUs
'''

from entrypoint2 import entrypoint
from pyavrutils.avrgcc import AvrGcc, AvrGccCompileError


def test(cc, mcu):
    print 'MCU =', mcu.ljust(20),
    cc.mcu = mcu
    try:
        cc.build(cc.minprog)
        print '    program/data size =', cc.size().program_bytes, ',', cc.size().data_bytes
    except AvrGccCompileError:
        print '    compile error'


@entrypoint
def main():
    cc = AvrGcc()
    print '--------------'
    print 'avr-gcc'
    print '--------------'

    print 'compiler version:', cc.version()
    cc.optimize_for_size()
    print 'compiler options:', ' '.join(cc.options_generated())
    print 'code:', cc.minprog
    print
    for mcu in cc.targets:
        test(cc, mcu)
```

Output:

```
$ python -m pyavrutils.examples.minsize
--------------
avr-gcc
--------------
compiler version: 4.5.3
compiler options: avr-gcc -Df_cpu=4000000 -mmcu=atmega168 --std=gnu99 -Wl,--relax -Wl,--gc-sectio
code: int main(){};

MCU = avr1                          compile error
MCU = avr2                          program/data size = 0 , 0
MCU = avr25                         program/data size = 0 , 0
MCU = avr3                          program/data size = 0 , 0
MCU = avr31                         program/data size = 0 , 0
MCU = avr35                         program/data size = 0 , 0
MCU = avr4                          program/data size = 0 , 0
MCU = avr5                          program/data size = 0 , 0
MCU = avr51                         program/data size = 0 , 0
MCU = avr6                          program/data size = 0 , 0
MCU = avrxmega1                     compile error
MCU = avrxmega2                     program/data size = 0 , 0
MCU = avrxmega3                     compile error
MCU = avrxmega4               program/data size = 0 , 0
MCU = avrxmega5               program/data size = 0 , 0
MCU = avrxmega6               program/data size = 0 , 0
MCU = avrxmega7               program/data size = 0 , 0
MCU = avrtiny10               program/data size = 0 , 0
MCU = at90s1200               compile error
MCU = attiny11                compile error
MCU = attiny12                compile error
MCU = attiny15                compile error
MCU = attiny28                compile error
MCU = at90s2313               program/data size = 46 , 0
MCU = at90s2323               program/data size = 30 , 0
MCU = at90s2333               program/data size = 52 , 0
MCU = at90s2343               program/data size = 30 , 0
MCU = attiny22                program/data size = 30 , 0
MCU = attiny26                program/data size = 48 , 0
MCU = at90s4414               program/data size = 54 , 0
MCU = at90s4433               program/data size = 52 , 0
MCU = at90s4434               program/data size = 62 , 0
MCU = at90s8515               program/data size = 54 , 0
MCU = at90c8534               program/data size = 42 , 0
MCU = at90s8535               program/data size = 62 , 0
MCU = attiny13                program/data size = 44 , 0
MCU = attiny13a               program/data size = 44 , 0
MCU = attiny2313              program/data size = 62 , 0
MCU = attiny2313a             program/data size = 66 , 0
MCU = attiny24                program/data size = 58 , 0
MCU = attiny24a               program/data size = 58 , 0
MCU = attiny4313              program/data size = 70 , 0
MCU = attiny44                program/data size = 62 , 0
MCU = attiny44a               program/data size = 62 , 0
MCU = attiny84                program/data size = 62 , 0
MCU = attiny84a               program/data size = 62 , 0
MCU = attiny25                program/data size = 54 , 0
MCU = attiny45                program/data size = 58 , 0
MCU = attiny85                program/data size = 58 , 0
MCU = attiny261               program/data size = 62 , 0
MCU = attiny261a              program/data size = 62 , 0
MCU = attiny461               program/data size = 66 , 0
MCU = attiny461a              program/data size = 66 , 0
MCU = attiny861               program/data size = 66 , 0
MCU = attiny861a              program/data size = 66 , 0
MCU = attiny87                program/data size = 68 , 0
MCU = attiny43u               program/data size = 60 , 0
MCU = attiny48                program/data size = 68 , 0
MCU = attiny88                program/data size = 68 , 0
MCU = at86rf401               program/data size = 40 , 0
MCU = ata6289                 program/data size = 82 , 0
MCU = at43usb355              program/data size = 80 , 0
MCU = at76c711                program/data size = 88 , 0
MCU = atmega103               program/data size = 124 , 0
MCU = at43usb320              program/data size = 80 , 0
MCU = attiny167               program/data size = 108 , 0
MCU = at90usb82               program/data size = 144 , 0
MCU = at90usb162              program/data size = 144 , 0
MCU = atmega8u2               program/data size = 180 , 0
MCU = atmega16u2              program/data size = 180 , 0
MCU = atmega32u2              program/data size = 180 , 0
MCU = attiny1634              compile error
MCU = atmega8                 program/data size = 66 , 0
MCU = atmega48                program/data size = 80 , 0
MCU = atmega48a               program/data size = 80 , 0
MCU = atmega48pa              compile error
MCU = atmega48p                 program/data size = 80 , 0
MCU = atmega88                  program/data size = 80 , 0
MCU = atmega88a                 program/data size = 80 , 0
MCU = atmega88p                 program/data size = 80 , 0
MCU = atmega88pa                program/data size = 80 , 0
MCU = atmega8515                program/data size = 62 , 0
MCU = atmega8535                program/data size = 70 , 0
MCU = atmega8hva                program/data size = 70 , 0
MCU = at90pwm1                  program/data size = 92 , 0
MCU = at90pwm2                  program/data size = 92 , 0
MCU = at90pwm2b                 program/data size = 92 , 0
MCU = at90pwm3                  program/data size = 92 , 0
MCU = at90pwm3b                 program/data size = 92 , 0
MCU = at90pwm81                 program/data size = 68 , 0
MCU = at90pwm161                compile error
MCU = atmega16                  program/data size = 112 , 0
MCU = atmega16a                 program/data size = 112 , 0
MCU = atmega161                 program/data size = 112 , 0
MCU = atmega162                 program/data size = 140 , 0
MCU = atmega163                 program/data size = 100 , 0
MCU = atmega164a                program/data size = 152 , 0
MCU = atmega164p                program/data size = 152 , 0
MCU = atmega165                 program/data size = 116 , 0
MCU = atmega165a                program/data size = 116 , 0
MCU = atmega165p                program/data size = 116 , 0
MCU = atmega168                 program/data size = 132 , 0
MCU = atmega168a                program/data size = 132 , 0
MCU = atmega168p                program/data size = 132 , 0
MCU = atmega169                 program/data size = 120 , 0
MCU = atmega169a                program/data size = 120 , 0
MCU = atmega169p                program/data size = 120 , 0
MCU = atmega169pa               program/data size = 120 , 0
MCU = atmega32                  program/data size = 112 , 0
MCU = atmega323                 program/data size = 108 , 0
MCU = atmega324a                program/data size = 152 , 0
MCU = atmega324p                program/data size = 152 , 0
MCU = atmega324pa               program/data size = 152 , 0
MCU = atmega325                 program/data size = 120 , 0
MCU = atmega325a                program/data size = 120 , 0
MCU = atmega325p                program/data size = 120 , 0
MCU = atmega325pa               compile error
MCU = atmega3250                program/data size = 128 , 0
MCU = atmega3250a               program/data size = 128 , 0
MCU = atmega3250p               program/data size = 128 , 0
MCU = atmega3250pa              compile error
MCU = atmega328                 program/data size = 132 , 0
MCU = atmega328p                program/data size = 132 , 0
MCU = atmega329                 program/data size = 120 , 0
MCU = atmega329a                program/data size = 120 , 0
MCU = atmega329p                program/data size = 120 , 0
MCU = atmega329pa               program/data size = 120 , 0
MCU = atmega3290                program/data size = 128 , 0
MCU = atmega3290a               program/data size = 128 , 0
MCU = atmega3290p               program/data size = 128 , 0
MCU = atmega3290pa              compile error
MCU = atmega406                 program/data size = 120 , 0
MCU = atmega64                  program/data size = 168 , 0
MCU = atmega640                 program/data size = 256 , 0
MCU = atmega644                 program/data size = 140 , 0
MCU = atmega644a                program/data size = 152 , 0
MCU = atmega644p                program/data size = 152 , 0
MCU = atmega644pa               program/data size = 152 , 0
MCU = atmega645                 program/data size = 120 , 0
MCU = atmega645a              program/data size = 120 , 0
MCU = atmega645p              program/data size = 120 , 0
MCU = atmega649               program/data size = 120 , 0
MCU = atmega649p              program/data size = 120 , 0
MCU = atmega649a              program/data size = 120 , 0
MCU = atmega6450              program/data size = 128 , 0
MCU = atmega6450a             program/data size = 128 , 0
MCU = atmega6450p             program/data size = 128 , 0
MCU = atmega6490              program/data size = 128 , 0
MCU = atmega6490a             program/data size = 128 , 0
MCU = atmega6490p             program/data size = 128 , 0
MCU = atmega64hve             program/data size = 128 , 0
MCU = atmega16hva             program/data size = 112 , 0
MCU = atmega16hva2            program/data size = 116 , 0
MCU = atmega16hvb             program/data size = 144 , 0
MCU = atmega16hvbrevb         program/data size = 144 , 0
MCU = atmega32hvb             program/data size = 144 , 0
MCU = atmega32hvbrevb         program/data size = 144 , 0
MCU = at90can32               program/data size = 176 , 0
MCU = at90can64               program/data size = 176 , 0
MCU = at90pwm216              program/data size = 156 , 0
MCU = at90pwm316              program/data size = 156 , 0
MCU = atmega32c1              program/data size = 152 , 0
MCU = atmega64c1              program/data size = 152 , 0
MCU = atmega16m1              program/data size = 152 , 0
MCU = atmega32m1              program/data size = 152 , 0
MCU = atmega64m1              program/data size = 152 , 0
MCU = atmega16u4              program/data size = 200 , 0
MCU = atmega32u4              program/data size = 200 , 0
MCU = atmega32u6              program/data size = 180 , 0
MCU = at90usb646              program/data size = 180 , 0
MCU = at90usb647              program/data size = 180 , 0
MCU = at90scr100              program/data size = 180 , 0
MCU = at94k                   program/data size = 172 , 0
MCU = m3000                   compile error
MCU = atmega128               program/data size = 168 , 0
MCU = atmega1280              program/data size = 256 , 0
MCU = atmega1281              program/data size = 232 , 0
MCU = atmega1284p             program/data size = 168 , 0
MCU = atmega128rfa1           program/data size = 316 , 0
MCU = at90can128              program/data size = 176 , 0
MCU = at90usb1286             program/data size = 180 , 0
MCU = at90usb1287             program/data size = 180 , 0
MCU = atmega2560              program/data size = 260 , 0
MCU = atmega2561              program/data size = 236 , 0
MCU = atxmega16a4             program/data size = 404 , 0
MCU = atxmega16d4             program/data size = 392 , 0
MCU = atxmega16x1             compile error
MCU = atxmega32a4             program/data size = 404 , 0
MCU = atxmega32d4             program/data size = 392 , 0
MCU = atxmega32x1             compile error
MCU = atxmega64a3             program/data size = 516 , 0
MCU = atxmega64d3             program/data size = 484 , 0
MCU = atxmega64a1             program/data size = 536 , 0
MCU = atxmega64a1u            program/data size = 548 , 0
MCU = atxmega128a3            program/data size = 520 , 0
MCU = atxmega128b1            compile error
MCU = atxmega128d3            program/data size = 488 , 0
MCU = atxmega192a3            program/data size = 520 , 0
MCU = atxmega192d3            program/data size = 488 , 0
MCU = atxmega256a3            program/data size = 520 , 0
MCU = atxmega256a3b           program/data size = 520 , 0
MCU = atxmega256a3bu          compile error
MCU = atxmega256d3             program/data size = 488 , 0
MCU = atxmega128a1             program/data size = 540 , 0
MCU = atxmega128a1u            program/data size = 552 , 0
MCU = attiny4                  program/data size = 48 , 0
MCU = attiny5                  program/data size = 50 , 0
MCU = attiny9                  program/data size = 48 , 0
MCU = attiny10                 program/data size = 50 , 0
MCU = attiny20                 program/data size = 62 , 0
MCU = attiny40                 program/data size = 62 , 0
```

CHAPTER

FIVE

ARDUINO BUILD TESTS

Code:

```
void setup()
{
}

void loop()
{
}
```

5.1 Results

5.1.1 Arduino version 0022

MCU	min
atmega8	OK (P:290 D:9)
atmega48	OK (P:358 D:9)
atmega168	OK (P:420 D:9)
atmega328p	OK (P:420 D:9)
atmega640	OK (P:622 D:9)
atmega1280	OK (P:622 D:9)
atmega2560	OK (P:626 D:9)

5.1.2 Arduino version 1.0

MCU	min
atmega8	OK (P:304 D:9)
atmega48	OK (P:372 D:9)
atmega168	OK (P:436 D:9)
atmega328p	OK (P:436 D:9)
atmega640	OK (P:638 D:9)
atmega1280	OK (P:638 D:9)
atmega2560	OK (P:642 D:9)

CHAPTER

SIX

API

class `pyavrutils.`**`AvrGcc`**(*mcu='atmega168'*)

`build`(*sources=None, headers=None*)
sources can be file name or code: sources=['x.c','int main(){}'] or sources='int main(){}'

`command_list`(*sources, _opt=False*)
command line as list

`error_text`

`minprog` = **'int main(){};'**

`ok`

`optimize_for_size`()
http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=90752

http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&t=69813

`optimize_no`()
all options set to default

`options_generated`()

`size`()

`targets`

`version`()
avr-gcc version

class `pyavrutils.`**`AvrSize`**
wrapper for avr-size

`ok`

`parse_output`(*s*)
Example output:

Device: atmega2561

Program: 4168 bytes (1.6% Full) (.text + .data + .bootloader)

Data: 72 bytes (0.9% Full) (.data + .bss + .noinit)

`run`(*objfile, mcu*)

class `pyavrutils.`**`Arduino`**(*mcu=None, f_cpu=None, board=None, hwpack='arduino', extra_lib=None, ver=None, backend='arscons', arduino_home=None, avr_home=None*)
wrapper for arscons and ino

`build`(*sources=None*)

`build_arscons`(*sources=None*)

`build_ino`(*sources=None*)

`command_list`()

`command_list_arscons`()
command line as list

`command_list_ino`()

`error_text`

`guess_projname`(*allfiles*)

`mcu_compiler`()

`minprog` = 'void setup(){};void loop(){};'

`ok`

`setup_sources`(*tempdir*, *sources*)

`size`()

`stderr`

`warnings`

CHAPTER

SEVEN

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

INDEX