pyavrutils Documentation

Release 0.1.0

ponty

March 11, 2012

CONTENTS

pyavrutils

Date March 11, 2012

PDF pyavrutils.pdf

Contents:

pyavrutils can build AVR and arduino code from python

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

Known problems:

- Python 3 is not supported
- temp files are not removed
- **arscons has some problems:**
  - it builds bigger programs
  - compile error in some cases

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
```

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_odD4ma.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-sections
>>> cc.build('int main(){}')
>>> cc.output
'/tmp/pyavrutils_odD4ma.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_p7OQV3/pyavrutils__RoiAe/pyavrutils__RoiAe.elf')
>>> cc.size()
AvrSize <prog:430 bytes 2.6% mem:9 bytes 0.9% >
>>> cc.size().program_bytes
430
>>> cc.board='pro'
>>> cc.build('void setup(){};void loop(){}')
>>> cc.output
path('/tmp/pyavrutils_oI9M6y/pyavrutils_sz6jqe/pyavrutils_sz6jqe.elf')
>>> cc.size().program_bytes
454
>>> cc.warnings
[u'build/core/IPAddress.h:51:55: warning: dereferencing type-punned pointer will break strict-alia
```

display warnings on console:

```
$ python -m pyavrutils.cli.arduino_warnings /usr/share/arduino/examples/4.Communication/Dimmer/Di
build/core/IPAddress.h:51:55: warning: dereferencing type-punned pointer will break strict-aliasi
build/core/IPAddress.h:52:75: 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/Tone.cpp:108:45: warning: only initialized variables can be placed into program memory
```

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).rjus
    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')

    test('volatile int x=3;_delay_ms(x)', '-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 ,
volatile int x=3;_delay_ms(x)     compiler option: -Os        compile error
```

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

index	board	min
1	atmega8	OK (P:288 D:9)
2	atmega88	OK (P:350 D:9)
3	bt	OK (P:416 D:9)
4	bt328	OK (P:416 D:9)
5	diecimila	OK (P:416 D:9)
6	fio	OK (P:440 D:9)
7	lilypad	OK (P:440 D:9)
8	lilypad328	OK (P:440 D:9)
9	mega	OK (P:618 D:9)
10	mega2560	OK (P:622 D:9)
11	metaboard	OK (P:416 D:9)
12	mini	OK (P:416 D:9)
13	pro	OK (P:440 D:9)
14	pro328	OK (P:440 D:9)
15	pro5v	OK (P:416 D:9)
16	pro5v328	OK (P:416 D:9)
17	uno	OK (P:416 D:9)
18	arduino_OrangutanSVP1284	OK (P:470 D:9)
19	arduino_amber128	OK (P:416 D:9)
20	arduino_android2561	OK (P:620 D:9)
21	arduino_android2561_16	OK (P:596 D:9)
22	arduino_at90can128	OK (P:474 D:9)
23	arduino_at90can32	OK (P:474 D:9)
24	arduino_at90can64	OK (P:474 D:9)
25	arduino_at90usb162	OK (P:374 D:9)

Continued on next page

Table 5.1 – continued from previous page

index	board	min
26	arduino_at90usb646	OK (P:484 D:9)
27	arduino_at90usb647	OK (P:484 D:9)
28	arduino_at90usbkey	OK (P:484 D:9)
29	arduino_atmega16	OK (P:364 D:9)
30	arduino_atmega165	OK (P:412 D:9)
31	arduino_atmega3290p	OK (P:400 D:9)
32	arduino_atmega8515	OK (P:262 D:9)
33	arduino_atmega8535	OK (P:290 D:9)
34	arduino_attiny2313	OK (P:298 D:9)
35	arduino_attiny26	OK (P:96 D:0)
36	arduino_attiny45	OK (P:294 D:9)
37	arduino_attiny85	OK (P:294 D:9)
38	arduino_bahbots1284p	OK (P:470 D:9)
39	arduino_butterfly	OK (P:416 D:9)
40	arduino_cerebot_plus	OK (P:644 D:9)
41	arduino_cerebotii	OK (P:446 D:9)
42	arduino_digilent_explorer	OK (P:412 D:9)
43	arduino_duino644	OK (P:422 D:9)
44	arduino_duino644p	OK (P:434 D:9)
45	arduino_gator	OK (P:428 D:9)
46	arduino_illuminato	OK (P:392 D:9)
47	arduino_penguino_avr	OK (P:340 D:9)
48	arduino_teensy2_ser	OK (P:530 D:9)
49	arduino_teensypp2_ser	OK (P:484 D:9)
50	arduino_wiring1281	OK (P:592 D:9)
51	atmega168	OK (P:414 D:9)
52	atmega328	OK (P:414 D:9)
53	atmega48	OK (P:354 D:9)
54	atmega640	OK (P:616 D:9)
55	atmega8	OK (P:286 D:9)
56	atmega88	OK (P:354 D:9)
57	bt	OK (P:414 D:9)
58	bt328	OK (P:414 D:9)
59	diecimila	OK (P:414 D:9)
60	dvk90can1	OK (P:498 D:9)
61	ecavr_atmega32	OK (P:364 D:9)
62	fio	OK (P:438 D:9)
63	lilypad	OK (P:438 D:9)
64	lilypad328	OK (P:438 D:9)
65	mega	OK (P:616 D:9)
66	mega1280stk500v2	OK (P:616 D:9)
67	mega2560stk500v2	OK (P:620 D:9)
68	mini	OK (P:414 D:9)
69	pro	OK (P:438 D:9)
70	pro328	OK (P:438 D:9)
71	pro5v	OK (P:414 D:9)
72	pro5v328	OK (P:414 D:9)
73	stk502	OK (P:416 D:9)
74	stk525	OK (P:508 D:9)
75	stk525_647	OK (P:508 D:9)

5.1.2 Arduino version 0023

index	board	min
76	atmega168	OK (P:416 D:9)
77	atmega328	OK (P:416 D:9)
78	atmega8	OK (P:288 D:9)
79	bt	OK (P:416 D:9)
80	bt328	OK (P:416 D:9)
81	diecimila	OK (P:416 D:9)
82	fio	OK (P:440 D:9)
83	lilypad	OK (P:440 D:9)
84	lilypad328	OK (P:440 D:9)
85	mega	OK (P:618 D:9)
86	mega2560	OK (P:622 D:9)
87	mini	OK (P:416 D:9)
88	pro	OK (P:440 D:9)
89	pro328	OK (P:440 D:9)
90	pro5v	OK (P:416 D:9)
91	pro5v328	OK (P:416 D:9)
92	uno	OK (P:416 D:9)

5.1.3 Arduino version 1.0

index	board	min
93	atmega168	OK (P:430 D:9)
94	atmega328	OK (P:430 D:9)
95	atmega8	OK (P:300 D:9)
96	bt	OK (P:430 D:9)
97	bt328	OK (P:430 D:9)
98	diecimila	OK (P:430 D:9)
99	ethernet	OK (P:430 D:9)
100	fio	OK (P:454 D:9)
101	lilypad	OK (P:454 D:9)
102	lilypad328	OK (P:454 D:9)
103	mega	OK (P:632 D:9)
104	mega2560	OK (P:636 D:9)
105	mini	OK (P:430 D:9)
106	mini328	OK (P:430 D:9)
107	nano	OK (P:430 D:9)
108	nano328	OK (P:430 D:9)
109	pro	OK (P:454 D:9)
110	pro328	OK (P:454 D:9)
111	pro5v	OK (P:430 D:9)
112	pro5v328	OK (P:430 D:9)
113	uno	OK (P:430 D:9)

5.2 Board configurations

5.2.1 Arduino version 0022

index	package	id	name	MCU
1	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8
				Con

Table 5.2 – continued from previous page

index	package	id	name	MCU
2	arduino	atmega88	atmega88@20000000 programmer:usbasp	atmega8
3	arduino	bt	Arduino BT w/ ATmega168	atmega1
4	arduino	bt328	Arduino BT w/ ATmega328	atmega3
5	arduino	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega1
6	arduino	fio	Arduino Fio	atmega3
7	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega1
8	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega3
9	arduino	mega	Arduino Mega (ATmega1280)	atmega1
10	arduino	mega2560	Arduino Mega 2560	atmega2
11	arduino	metaboard	Metaboard	atmega1
12	arduino	mini	Arduino Mini	atmega1
13	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega1
14	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega3
15	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega1
16	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega3
17	arduino	uno	Arduino Uno	atmega3
18	arduino-extras	arduino_OrangutanSVP1284	Arduino-Orangutan SVP-1284	atmega1
19	arduino-extras	arduino_amber128	Arduino-Amber 128 14.7456 Mhz	atmega1
20	arduino-extras	arduino_android2561	Arduino-Android 2561 8Mhz	atmega2
21	arduino-extras	arduino_android2561_16	Arduino-Android 2561 16Mhz	atmega2
22	arduino-extras	arduino_at90can128	AT90CAN128 development board NHL (arduino core)	at90can
23	arduino-extras	arduino_at90can32	at90can32 (arduino core)	at90can
24	arduino-extras	arduino_at90can64	at90can64 (arduino core)	at90can
25	arduino-extras	arduino_at90usb162	Arduino-at90usb162	at90usb
26	arduino-extras	arduino_at90usb646	Arduino-at90usb646	at90usb
27	arduino-extras	arduino_at90usb647	Arduino-at90usb647	at90usb
28	arduino-extras	arduino_at90usbkey	Arduino-at90usbkey	at90usb
29	arduino-extras	arduino_atmega16	Arduino-Atmega16	atmega1
30	arduino-extras	arduino_atmega165	Arduino-Atmega165	atmega1
31	arduino-extras	arduino_atmega3290p	Arduino-Atmega3290p	atmega3
32	arduino-extras	arduino_atmega8515	Arduino-ATmega8515	atmega8
33	arduino-extras	arduino_atmega8535	Arduino-Test-Atmega8535	atmega8
34	arduino-extras	arduino_attiny2313	Arduino-ATtiny2313	attiny23
35	arduino-extras	arduino_attiny26	Arduino-ATtiny26	attiny26
36	arduino-extras	arduino_attiny45	Arduino-ATtiny45	attiny45
37	arduino-extras	arduino_attiny85	Arduino-ATtiny85	attiny85
38	arduino-extras	arduino_bahbots1284p	Arduino-BahBots 1284p	atmega1
39	arduino-extras	arduino_butterfly	Arduino-Butterfly stk500	atmega1
40	arduino-extras	arduino_cerebot_plus	Arduino-Cerebot Plus	atmega2
41	arduino-extras	arduino_cerebotii	Arduino-Cerebot II atemga64	atmega6
42	arduino-extras	arduino_digilent_explorer	Arduino-Digilent I/O Explorer USB	atmega1
43	arduino-extras	arduino_duino644	Arduino-Duino 644	atmega6
44	arduino-extras	arduino_duino644p	Arduino-Duino 644P	atmega6
45	arduino-extras	arduino_gator	Arduino-Rugged Circuits Gator Board	atmega3
46	arduino-extras	arduino_illuminato	Arduino-illuminato	atmega6
47	arduino-extras	arduino_penguino_avr	Arduino-Penguino AVR	atmega3
48	arduino-extras	arduino_teensy2_ser	Arduino-Teensy 2.0 (USB Serial)	atmega3
49	arduino-extras	arduino_teensypp2_ser	Arduino-Teensy++ 2.0 (USB Serial)	at90usb
50	arduino-extras	arduino_wiring1281	Arduino-Wiring 1281	atmega1
51	arduino-extras	atmega168	Arduino NG or older w/ ATmega168	atmega1
52	arduino-extras	atmega328	Arduino Duemilanove or Nano w/ ATmega328	atmega3
53	arduino-extras	atmega48	Arduino Atmega48	atmega4
54	arduino-extras	atmega640	Arduino atmega640	atmega6

Con

Table 5.2 – continued from previous page

index	package	id	name	MCU
55	arduino-extras	atmega8	Arduino NG or older w/ ATmega8	atmega8
56	arduino-extras	atmega88	Atmega88	atmega8
57	arduino-extras	bt	Arduino BT w/ ATmega168	atmega1
58	arduino-extras	bt328	Arduino BT w/ ATmega328	atmega3
59	arduino-extras	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega1
60	arduino-extras	dvk90can1	STK500 w/DVK90CAN1 - AT90can128 (Arduino Core)	at90can
61	arduino-extras	ecavr_atmega32	Embedded market atmega32	atmega3
62	arduino-extras	fio	Arduino Fio	atmega3
63	arduino-extras	lilypad	LilyPad Arduino w/ ATmega168	atmega1
64	arduino-extras	lilypad328	LilyPad Arduino w/ ATmega328	atmega3
65	arduino-extras	mega	Arduino Mega	atmega1
66	arduino-extras	mega1280stk500v2	Arduino Mega1280 stk500v2	atmega1
67	arduino-extras	mega2560stk500v2	Arduino Mega2560 stk500v2	atmega2
68	arduino-extras	mini	Arduino Mini	atmega1
69	arduino-extras	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega1
70	arduino-extras	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega3
71	arduino-extras	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega1
72	arduino-extras	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega3
73	arduino-extras	stk502	STK500 w/STKk502 - ATmega169 (Arduino Core)	atmega1
74	arduino-extras	stk525	STK500 w/STK525 - at90usb1287 (Arduino Core)	at90usb
75	arduino-extras	stk525_647	STK500 w/STK525 - at90usb647 (Arduino Core)	at90usb

5.2.2 Arduino version 0023

index	package	id	name	MCU	F_CPU
76	arduino	atmega168	Arduino NG or older w/ ATmega168	atmega168	16000000L
77	arduino	atmega328	Arduino Duemilanove or Nano w/ ATmega328	atmega328p	16000000L
78	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8	16000000L
79	arduino	bt	Arduino BT w/ ATmega168	atmega168	16000000L
80	arduino	bt328	Arduino BT w/ ATmega328	atmega328p	16000000L
81	arduino	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega168	16000000L
82	arduino	fio	Arduino Fio	atmega328p	8000000L
83	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega168	8000000L
84	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega328p	8000000L
85	arduino	mega	Arduino Mega (ATmega1280)	atmega1280	16000000L
86	arduino	mega2560	Arduino Mega 2560	atmega2560	16000000L
87	arduino	mini	Arduino Mini	atmega168	16000000L
88	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega168	8000000L
89	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega328p	8000000L
90	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega168	16000000L
91	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega328p	16000000L
92	arduino	uno	Arduino Uno	atmega328p	16000000L

5.2.3 Arduino version 1.0

index	package	id	name	MCU	F_CPU
93	arduino	atmega168	Arduino NG or older w/ ATmega168	atmega168	16000000L
94	arduino	atmega328	Arduino Duemilanove w/ ATmega328	atmega328p	16000000L
95	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8	16000000L
96	arduino	bt	Arduino BT w/ ATmega168	atmega168	16000000L
97	arduino	bt328	Arduino BT w/ ATmega328	atmega328p	16000000L
98	arduino	diecimila	Arduino Diecimila or Duemilanove w/ ATmega168	atmega168	16000000L
99	arduino	ethernet	Arduino Ethernet	atmega328p	16000000L
100	arduino	fio	Arduino Fio	atmega328p	8000000L
101	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega168	8000000L
102	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega328p	8000000L
103	arduino	mega	Arduino Mega (ATmega1280)	atmega1280	16000000L
104	arduino	mega2560	Arduino Mega 2560 or Mega ADK	atmega2560	16000000L
105	arduino	mini	Arduino Mini w/ ATmega168	atmega168	16000000L
106	arduino	mini328	Arduino Mini w/ ATmega328	atmega328p	16000000L
107	arduino	nano	Arduino Nano w/ ATmega168	atmega168	16000000L
108	arduino	nano328	Arduino Nano w/ ATmega328	atmega328p	16000000L
109	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega168	8000000L
110	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega328p	8000000L
111	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega168	16000000L
112	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega328p	16000000L
113	arduino	uno	Arduino Uno	atmega328p	16000000L

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**(*board='pro'*, *hwpack='arduino'*, *mcu=None*, *f_cpu=None*, *extra_lib=None*, *ver=None*, *backend='arscons'*)
wrapper for arscons

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

DEVELOPMENT

7.1 Tools

1. setuptools
2. Paver
3. nose
4. ghp-import
5. pyflakes
6. pychecker
7. paved fork
8. Sphinx
9. sphinxcontrib-programscreenshot
10. sphinxcontrib-paverutils
11. `autorun` from sphinx-contrib (there is no simple method, you have to download/unpack/setup)

7.2 Install on ubuntu

```
sudo apt-get install python-setuptools
sudo apt-get install python-paver
sudo apt-get install python-nose
sudo easy_install ghp-import
sudo apt-get install pyflakes
sudo apt-get install pychecker
sudo easy_install https://github.com/ponty/paved/zipball/master
sudo apt-get install scrot
sudo apt-get install xvfb
sudo apt-get install xserver-xephyr
sudo apt-get install python-imaging
sudo apt-get install python-sphinx
sudo easy_install sphinxcontrib-programscreenshot
sudo easy_install sphinxcontrib-programoutput
sudo easy_install sphinxcontrib-paverutils
```

7.3 Tasks

Paver is used for task management, settings are saved in pavement.py. Sphinx is used to generate documentation.

print paver settings:

```
paver printoptions
```

clean generated files:

```
paver clean
```

generate documentation under *docs/_build/html*:

```
paver cog pdf html
```

upload documentation to github:

```
paver ghpages
```

run unit tests:

```
paver nose
#or
nosetests --verbose
```

check python code:

```
paver pyflakes
paver pychecker
```

generate python distribution:

```
paver sdist
```

upload python distribution to PyPI:

```
paver upload
```

CHAPTER

EIGHT

INDICES AND TABLES

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

INDEX