confduino Documentation

Release 0.0.6

ponty

November 06, 2011

CONTENTS

confduino

Date November 06, 2011

PDF confduino.pdf

Contents:

confduino is an arduino library configurator

Links:

- home: https://github.com/ponty/confduino
- documentation: http://ponty.github.com/confduino
- arduino libraries: http://www.arduino.cc/en/Reference/Libraries

Features:

- list, install, remove arduino libraries
- install libraries from internet or local drive
- fix `examples` directory name before installing
- clean library (.*,_*,..) before installing
- move examples under `examples` directory
- list, install, remove arduino programmers
- list, install, remove arduino boards
- written in python
- crossplatform
- can be used as a python library or as a console program
- unpacker backend: pyunpack
- downloader backend: urllib
- some functionality is based on arscons

Known problems:

- Python 3 is not supported
- tested only on linux
- some libraries with unusual structure can not be installed
- not all commands have console interface

CHAPTER

ONE

BASIC USAGE

install library:

```
>>> from confduino.libinstall import install_lib
>>> install_lib('http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip')
```

or on console:

```
python -m confduino.libinstall http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip
```

install a lot of libraries:

```
python -m confduino.libinstall.examples.upgrademany
```

CHAPTER

TWO

INSTALLATION

2.1 General

- install arduino
- install python
- install setuptools
- install backends for pyunpack (optional)
- install the program:

```
# as root
easy_install confduino
```

2.2 Ubuntu

```
sudo apt-get install arduino
sudo apt-get install python-setuptools
sudo easy_install confduino
# optional
sudo easy_install http://sourceforge.net/projects/patool/files/0.13/patool-0.13.tar.gz/download
sudo apt-get install unzip unrar p7zip-full
```

2.3 Uninstall

first install pip:

```
# as root
pip uninstall confduino
```

CHAPTER

THREE

USAGE WITH LIBRARIES

3.1 Arduino path

If Arduino can not be found at default path, then `ARDUINO_HOME` environment variable should be set.

on Ubuntu: in `~/.profile`:

```
ARDUINO_HOME=~/opt/arduino
export ARDUINO_HOME
```

Default path:

- Mac: /Applications/Arduino.app/Contents/Resources/Java
- Linux: /usr/share/arduino/

3.2 List installed libraries

From python:

```
>>> from confduino.liblist import libraries
>>> libraries()
['AdvButton', 'Bounce', 'CapSense', 'Charlieplex', 'Collection', 'DallasTemperature', 'DateTime', 'Da
```

From console:

```
$ python -m confduino.liblist
AdvButton
Bounce
CapSense
Charlieplex
Collection
DallasTemperature
DateTime
DateTimeStrings
DigitalToggle
EEPROM
Enerlib
EventFuse
Flash
FreeMemory
FreqCounter
LiquidCrystal
```

```
Metro
MsTimer2
NewSoftSerial
OneWire
PID_v1
PS2Keyboard
PString
PinChangeInt
Qtouch1Wire
SSerial2Mobile
SerialDebug
SerialIP
SevenSegment
Streaming
TimedAction
TimerOne
UComms
bassdll
myprojects
```

3.3 Install new library

Existing library will not be changed.

From python:

```
>>> from confduino.libinstall import install_lib
>>> install_lib('http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip')
```

From console:

```
python -m confduino.libinstall http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip
```

3.4 Upgrade existing library

Same as install with *replace_existing* option.

From python:

```
>>> from confduino.libinstall import install_lib
>>> install_lib('http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip', replace_existing=1)
```

From console:

```
python -m confduino.libinstall http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip --replace
```

3.5 Remove existing library

From python:

```
>>> from confduino.libremove import remove_lib
>>> remove_lib('PS2Keyboard')
```

From console:

```
python -m confduino.libremove PS2Keyboard
```

CHAPTER

FOUR

USAGE WITH BOARDS

4.1 List installed boards

From python:

```
>>> from confduino.boardlist import boards
>>> boards()
AutoBunch(atmega8=AutoBunch(bootloader=AutoBunch(file='ATmegaBOOT.hex', high_fuses='0xca', lock_bits=
>>> boards().diecimila.build.f_cpu
'16000000L'
>>> boards()['diecimila']['build']['f_cpu']
'16000000L'
```

From console:

```
$ python -m confduino.boardlist
['atmega8',
 'atmega88',
 'bt',
 'bt328',
 'diecimila',
 'fio',
 'lilypad',
 'lilypad328',
 'mega',
 'mega2560',
 'metaboard',
 'mini',
 'pro',
 'pro328',
 'pro5v',
 'pro5v328',
 'uno']
```

4.2 List installed MCUs

From python:

```
>>> from confduino.mculist import mcus
>>> mcus()
['at90can128', 'at90can32', 'at90can64', 'at90usb1286', 'at90usb1287', 'at90usb162', 'at90usb646', 'a
```

From console:

4.3 Install new board

Existing board will not be changed.

From python:

```
from confduino.boardinstall import install_board
from confduino.util import AutoBunch
from entrypoint2 import entrypoint

@entrypoint
def install(id='atmega88', mcu='atmega88', f_cpu=20000000, upload='usbasp', core='arduino', replace_e
    'install atmega88 board'
    board = AutoBunch()
    board.name = '{mcu}@{f_cpu} programmer:{upload}'.format(mcu=mcu,f_cpu=f_cpu,upload=upload)

    board.upload.using = upload

    board.build.mcu = mcu
    board.build.f_cpu = str(f_cpu) + 'L'
    board.build.core = core

    install_board(id, board, replace_existing=replace_existing)
```

console is not implemented

4.4 Remove existing board

From python:

```
>>> from confduino.boardremove import remove_board
>>> remove_board('diecimila')
```

From console:

```
python -m confduino.boardremove diecimila
```

CHAPTER

FIVE

USAGE WITH PROGRAMMERS

5.1 List installed programmers

From python:

```
>>> from confduino.proglist import programmers
>>> programmers()
AutoBunch(arduinoisp=AutoBunch(communication='serial', name='Arduino as ISP', protocol='stk500v1', sp
>>> programmers().arduinoisp.speed
'19200'
>>> programmers()['arduinoisp']['speed']
'19200'
```

From console:

```
$ python -m confduino.proglist
{'arduinoisp': {'communication': 'serial',
                'name': 'Arduino as ISP',
                'protocol': 'stk500v1',
                'speed': '19200'},
 'avrisp': {'communication': 'serial',
            'name': 'AVR ISP',
            'protocol': 'stk500v1'},
 'avrispmkii': {'communication': 'usb',
                'name': 'AVRISP mkII',
                'protocol': 'stk500v2'},
 'parallel': {'force': 'true',
              'name': 'Parallel Programmer',
              'protocol': 'dapa'},
 'stk200': {'force': 'true', 'name': 'STK200', 'protocol': 'dapa'},
 'usbasp': {'communication': 'usb', 'name': 'USBasp', 'protocol': 'usbasp'},
 'usbtinyisp': {'name': 'USBtinyISP', 'protocol': 'usbtiny'}}
```

5.2 Install new programmer

From python:

```
from confduino.proginstall import install_programmer
from confduino.util import AutoBunch
from entrypoint2 import entrypoint

@entrypoint
```

```
def install(replace_existing=False):
    'install usbasp programmer'
    usbasp = AutoBunch()
    usbasp.name = 'USBasp'
    usbasp.communication = 'usb'
    usbasp.protocol = 'usbasp'

    install_programmer('usbasp', usbasp, replace_existing=replace_existing)
```

console is not implemented

5.3 Remove existing programmer

From python:

```
>>> from confduino.progremove import remove_programmer
>>> remove_programmer('parallel')
```

From console:

```
python -m confduino.progremove parallel
```

CHAPTER

SIX

EXAMPLES

6.1 Install libraries

Many libraries are upgraded in examples/upgrademany.py, this can be started:

```
python -m confduino.examples.upgrademany
```

Code:

```
from confduino.libinstall import install_lib
from entrypoint2 import entrypoint

UPGRADE = True

def upgrade(url):
    print 'upgrading ' + url
    install_lib(url, UPGRADE)

@entrypoint
def upgrade_many():
    'upgrade many libs'

    # you can set your arduino path if it is not default
    #os.environ['ARDUINO_HOME'] = '/home/...'

    ############################
    # arduino.cc
    ############################
    upgrade('http://arduino.cc/playground/uploads/Main/PS2Keyboard002.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/Metro.zip')
    upgrade('http://www.arduino.cc/playground/uploads/Main/MsTimer2.zip')
#    upgrade('http://www.arduino.cc/playground/uploads/Code/Time.zip')
#    upgrade('http://arduino.cc/playground/uploads/Main/LedControl.zip')
#    upgrade('http://www.arduino.cc/playground/uploads/Code/ks0108GLCD.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/Bounce.zip')
    upgrade('http://arduino.cc/playground/uploads/Main/CapacitiveSense003.zip')
    upgrade('http://arduino.cc/playground/uploads/Main/PinChangeInt.zip')
#    upgrade('http://arduino.cc/playground/uploads/Code/TimerThree.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/TimedAction-1_6.zip')
#    upgrade('http://www.arduino.cc/playground/uploads/Code/Time.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/EventFuse.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/Charlieplex.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/DigitalToggle.zip')
    upgrade('http://arduino.cc/playground/uploads/Code/Enerlib.zip')
```

```
    upgrade('http://arduino.cc/playground/uploads/Code/AdvButton_11.zip')
    #upgrade('http://arduino.cc/playground/uploads/Code/AdvButton.zip') # old version

#    upgrade('http://arduino.cc/playground/uploads/Code/SerialDebugger.zip') # can't install

    ############################
    # arduiniana.org
    ############################
    # TODO: how to get latest version??
    upgrade('http://arduiniana.org/PString/PString2.zip')
    upgrade('http://arduiniana.org/Flash/Flash3.zip')
    upgrade('http://arduiniana.org/NewSoftSerial/NewSoftSerial10c.zip')
    upgrade('http://arduiniana.org/Streaming/Streaming4.zip')
#    upgrade('http://arduiniana.org/PWMServo/PWMServo.zip')
#    upgrade('http://arduiniana.org/TinyGPS/TinyGPS10.zip')


    ############################
    # google
    ############################

#    upgrade('http://rogue-code.googlecode.com/files/Arduino-Library-Tone.zip') # already in core!

#    upgrade('http://arduino-playground.googlecode.com/files/LedDisplay03.zip')
    upgrade('http://sserial2mobile.googlecode.com/files/SSerial2Mobile-1.1.0.zip')
#    upgrade('http://webduino.googlecode.com/files/webduino-1.4.1.zip')
    upgrade('http://arduino-pid-library.googlecode.com/files/PID_v1.0.1.zip')
    upgrade('http://ideoarduinolibraries.googlecode.com/files/Qtouch1Wire.zip')
    upgrade('http://arduino-timerone.googlecode.com/files/TimerOne-v2.zip')


    ############################
    # others
    ############################
    upgrade('http://download.milesburton.com/Arduino/MaximTemperature/DallasTemperature_370Beta.zip')
    upgrade('http://www.pjrc.com/teensy/arduino_libraries/OneWire.zip')
    upgrade('http://interface.khm.de/wp-content/uploads/2009/01/FreqCounter1.zip')
#    upgrade('http://github.com/wimleers/flexitimer2/zipball/v1.0')
#    upgrade('http://www.state-machine.com/arduino/qp_arduino.zip')
#    upgrade('ftp://momjian.us/pub/arduino/TButton.zip') # AdvButton is better
    upgrade('http://johnmchilton.com/media/UComms.zip')
    upgrade('http://www.shikadi.net/files/arduino/SerialIP-1.0.zip')
```

6.2 Install USBasp programmer

```
python -m confduino.examples.usbasp
```

Code:

```
from confduino.proginstall import install_programmer
from confduino.util import AutoBunch
from entrypoint2 import entrypoint

@entrypoint
def install(replace_existing=False):
    'install usbasp programmer'
```

```
    usbasp = AutoBunch()
    usbasp.name = 'USBasp'
    usbasp.communication = 'usb'
    usbasp.protocol = 'usbasp'

    install_programmer('usbasp', usbasp, replace_existing=replace_existing)
```

6.3 Install STK200 programmer

```
python -m confduino.examples.stk200
```

Code:

```
from confduino.proginstall import install_programmer
from confduino.util import AutoBunch
from entrypoint2 import entrypoint

@entrypoint
def install(replace_existing=False):
    'install stk200 programmer'
    bunch = AutoBunch()
    bunch.name = 'STK200'
    bunch.protocol = 'stk200'
    #bunch.force = 'true'
    # bunch.delay=200

    install_programmer('stk200', bunch, replace_existing=replace_existing)
```

6.4 Install atmega88 board

```
python -m confduino.examples.atmega88
```

Code:

```
from confduino.boardinstall import install_board
from confduino.util import AutoBunch
from entrypoint2 import entrypoint

@entrypoint
def install(id='atmega88', mcu='atmega88', f_cpu=20000000, upload='usbasp', core='arduino', replace_e
    'install atmega88 board'
    board = AutoBunch()
    board.name = '{mcu}@{f_cpu} programmer:{upload}'.format(mcu=mcu,f_cpu=f_cpu,upload=upload)

    board.upload.using = upload

    board.build.mcu = mcu
    board.build.f_cpu = str(f_cpu) + 'L'
    board.build.core = core

    install_board(id, board, replace_existing=replace_existing)
```

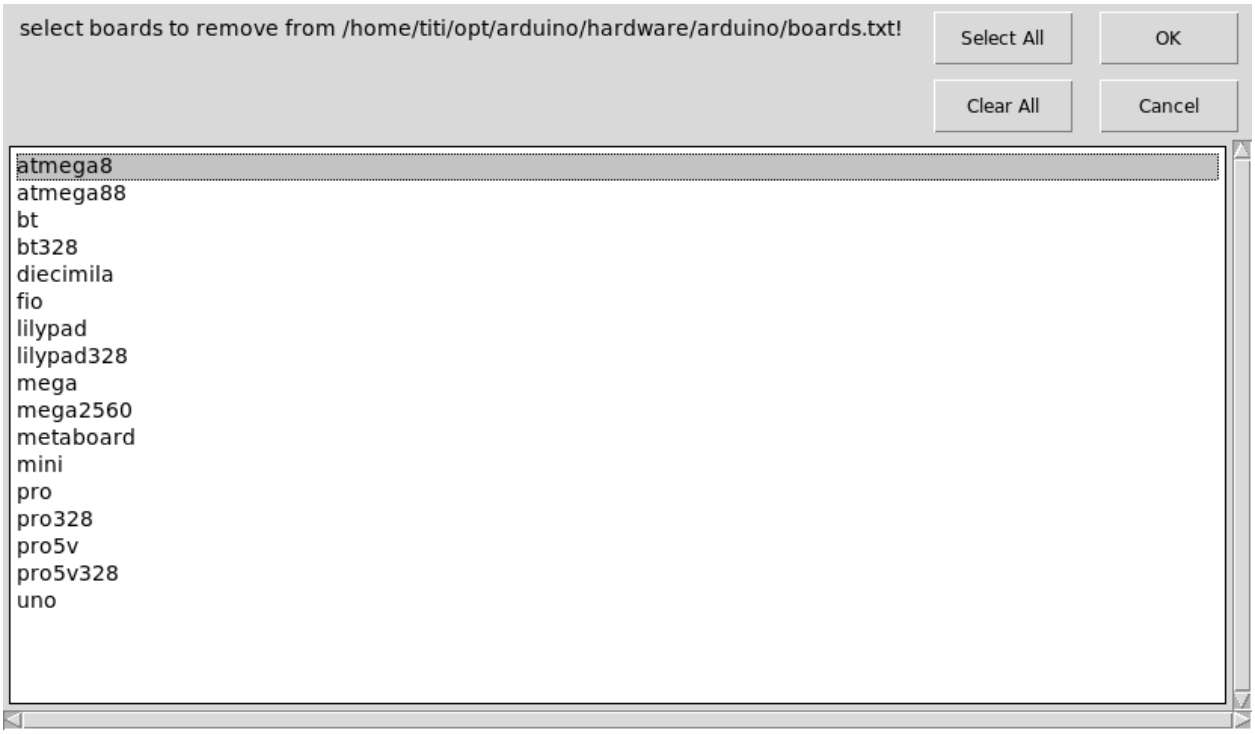
options:

```
$ python -m confduino.examples.atmega88 --help
usage: atmega88.py [-h] [-i ID] [-m MCU] [-f F_CPU] [-u UPLOAD] [-c CORE] [-r]
                   [--debug]

install atmega88 board

optional arguments:
  -h, --help            show this help message and exit
  -i ID, --id ID
  -m MCU, --mcu MCU
  -f F_CPU, --f-cpu F_CPU
  -u UPLOAD, --upload UPLOAD
  -c CORE, --core CORE
  -r, --replace-existing
  --debug               set logging level to DEBUG
```

6.5 remove boards

```
$ python -m confduino.examples.remove_boards
```


```
$ python -m confduino.examples.remove_boards --hwpack arduino
```



Code:

```
from confduino.boardlist import boards, boards_txt, board_names
from confduino.boardremove import remove_board
from confduino.hwpacklist import hwpack_names
from entrypoint2 import entrypoint
import psidialogs


@entrypoint
def remove_boards_gui(hwpack=''):
    'remove boards by GUI'
    if not hwpack:
        if len(hwpack_names()) > 1:
            hwpack = psidialogs.choice(hwpack_names(),
                                       'select hardware package to select board from!',
                                       title='select')
        else:
            hwpack = hwpack_names()[0]
    print hwpack, 'selected'

    if hwpack:
        sel = psidialogs.multi_choice(board_names(hwpack),
                                      'select boards to remove from %s!' % boards_txt(hwpack),
                                      title='remove boards')
        print sel, 'selected'

        if sel:
            for x in sel:
                remove_board(x)
                print x + ' was removed'
```

CHAPTER

SEVEN

COMMAND LINE HELP

7.1 lib

7.1.1 list

```
$ python -m confduino.liblist --help
usage: liblist.py [-h] [--debug]

print installed arduino libraries

optional arguments:
  -h, --help  show this help message and exit
  --debug     set logging level to DEBUG
```

7.1.2 install

```
$ python -m confduino.libinstall --help
usage: libinstall.py [-h] [-r] [--debug] url

install library from web or local files system

positional arguments:
  url                   web address or file path

optional arguments:
  -h, --help            show this help message and exit
  -r, --replace-existing
                        bool
  --debug               set logging level to DEBUG
```

7.1.3 remove

```
$ python -m confduino.libremove --help
usage: libremove.py [-h] [--debug] lib_name

remove library

positional arguments:
  lib_name    library name (e.g. 'PS2Keyboard')
```

```
optional arguments:
  -h, --help  show this help message and exit
  --debug     set logging level to DEBUG
```

7.2 board

7.2.1 list

```
$ python -m confduino.boardlist --help
usage: boardlist.py [-h] [--hwpack HWPACK] [--debug]

print boards from boards.txt

optional arguments:
  -h, --help       show this help message and exit
  --hwpack HWPACK
  --debug          set logging level to DEBUG
```

7.2.2 install

not implemented

7.2.3 remove

```
$ python -m confduino.boardremove --help
usage: boardremove.py [-h] [--debug] board_id

remove board

positional arguments:
  board_id    board id (e.g. 'diecimila')

optional arguments:
  -h, --help  show this help message and exit
  --debug     set logging level to DEBUG
```

7.3 programmer

7.3.1 list

```
$ python -m confduino.proglist --help
usage: proglist.py [-h] [--debug]

print programmers from programmers.txt

optional arguments:
  -h, --help  show this help message and exit
  --debug     set logging level to DEBUG
```

7.3.2 install

not implemented

7.3.3 remove

```
$ python -m confduino.progremove --help
usage: progremove.py [-h] [--debug] programmer_id

remove programmer

positional arguments:
  programmer_id  programmer id (e.g. 'avrisp')

optional arguments:
  -h, --help     show this help message and exit
  --debug        set logging level to DEBUG
```

CHAPTER

EIGHT

API

8.1 lib

confduino.liblist.**libraries**()
return installed library names

confduino.liblist.**libraries_dir**()
return library root path

confduino.liblist.**print_libraries**()
print installed arduino libraries

confduino.libinstall.**find_lib_dir**(*root*)
search for lib dir under root

confduino.libinstall.**fix_examples_dir**(*lib_dir*)
rename examples dir to `examples`

confduino.libinstall.**install_lib**(*url*, *replace_existing=False*)
install library from web or local files system

Parameters

- **url** – web address or file path
- **replace_existing** – bool

Return type None

confduino.libinstall.**move_examples**(*root*, *lib_dir*)
find examples not under lib dir, and move into `examples`

confduino.libremove.**remove_lib**(*lib_name*)
remove library

Parameters **lib_name** – library name (e.g. 'PS2Keyboard')

Return type None

8.2 board

confduino.boardlist.**board_names**(*hwpack='arduino'*)
return installed board names

confduino.boardlist.**boards**(*hwpack='arduino'*)
read boards from boards.txt

Parameters **core_package** – 'all,'arduino',..

confduino.boardlist.**boards_txt**(*hwpack='arduino'*)
path of boards.txt

confduino.boardlist.**print_boards**(*hwpack='arduino'*)
print boards from boards.txt

confduino.boardinstall.**install_board**(*board_id*, *board_options*, *hwpack='arduino'*, *replace_existing=False*)
install board in boards.txt

Parameters

- **board_id** – string identifier
- **board_options** – dict like
- **replace_existing** – bool

Return type None

confduino.boardremove.**remove_board**(*board_id*)
remove board

Parameters **board_id** – board id (e.g. 'diecimila')

Return type None

8.3 programmer

confduino.proglist.**print_programmers**()
print programmers from programmers.txt

confduino.proglist.**programmers**()
read programmers from programmers.txt

confduino.proglist.**programmers_txt**()
path of programmers.txt

confduino.proginstall.**install_programmer**(*programmer_id*, *programmer_options*, *replace_existing=False*)
install programmer in programmers.txt

Parameters

- **programmer_id** – string identifier
- **programmer_options** – dict like
- **replace_existing** – bool

Return type None

confduino.progremove.**remove_programmer**(*programmer_id*)
remove programmer

Parameters **programmer_id** – programmer id (e.g. 'avrisp')

Return type None

CHAPTER

NINE

INDICES AND TABLES

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

PYTHON MODULE INDEX

C

INDEX