FREE eBook

LEARNING robotframework

Free unaffiliated eBook created from
Stack Overflow contributors.

#robotframework

work

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: robotframework

It is an unofficial and free robotframework ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official robotframework.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with robotframework

Remarks

This section provides an overview of what robotframework is, and why a developer might want to use it.

It should also mention any large subjects within robotframework, and link out to the related topics. Since the Documentation for robotframework is new, you may need to create initial versions of those related topics.

Versions

Version	Release date
Robot Framework 3.0.2	2017-02-14
Robot Framework 3.0.1	2017-01-06
Robot Framework 3.0	2015-12-31
Robot Framework 2.9.2	2015-10-09
Robot Framework 2.9.1	2015-08-28
Robot Framework 2.9	2015-07-30

Examples

Installation or Setup

Detailed instructions on getting Robot Framework set up or installed.

Robot framework is a generic test automation framework.This is implemented using Python and is supported on Python 2 and Python 3 Jython (JVM) and IronPython (.NET) and PyPy. For

1. Acceptance testing
2. Acceptance test-driven development (ATDD)

Prerequisites

1. Install a interpreters

2. Configuring PATH
3. Setting https_proxy

Python has the most advanced implementations and it is suggested to use Python, if you do not have exceptional requirements.

Robot Framework Version	Supported interpreter Version
Robot Framework 3.0	Python 2.6
Robot Framework 3.0	Python 2.7
Robot Framework 3.0	Python 3.3
Robot Framework 3.0	Jython 2.7 & Java 7
Robot Framework 3.0	IronPython 2.7
Robot Framework 2.5-2.8	Python 2.5
Robot Framework 2.5-2.8	Jython 2.5
Robot Framework 2.0-2.1	Python 2.3
Robot Framework 2.0-2.1	Python 2.4
Robot Framework 2.0-2.1	Jython 2.2

Python installation

Desired version of python can be downloaded from https://www.python.org/downloads/

Jython installation

An installer can be found at http://jython.org. You can run this executable JAR package from the command line like javaava -jar jython_installer-.jar.

IronPython installation

An installer can be found at http://ironpython.net/download/ for IronPython 2.7.When using IronPython, an additional dependency is installing elementtree module 1.2.7

Configuring PATH & Setting https_proxy

Add Python installation directory (by default C:\Python27, C:\Python27\Scripts, C:\jython2.7.0\bin etc on windows) and Scripts directory to the beginning of your path variable

Value of https_proxy should be the URL of the proxy. This is required when these packages are installed with pip and you are in a proxy network

Installing Robot Framework with pip

Install the latest version of robotframework

```
pip install robotframework
```

Install a specific version

```
pip install robotframework==2.0
```

Installing Robot Framework from source

Source distribution of Robot Framework can be found at https://code.google.com/archive/p/robotframework/downloads.Robot Framework is installed from source using Python's standard setup.py script in the source scripts directory

```
python setup.py install
jython setup.py install
ipy setup.py install
```

Installing Robot Framework 3.0 on a Windows Machine using Python 2.7.11

This is a quick guide to get Robot Framework 3.0 working on a Windows machine using Python 2.7.11 - It does not go into too much depth on the why and how, it simply gets you up and running. First things are first, let't go and install Python!

1. Download Python 2.7.11 for Windows. (Windows x86-64 MSI installer or Windows x86 MSI installer depending on architecture)

2. Run through the install, making sure you install "pip" and that you opt in for the "Add python.exe to Path" (You may have to restart your machine to take advantage of the Python PATH. In this guide, it presumes you don't have that luxury)

3. Once it is installed, let's do a quick check to make sure it installed correctly. Run CMD as admin and navigate to where Python was installed to `cd C:\Python27` and type in `python -V`. It should return "Python 2.7.11"

That is it, Python is now installed to your machine. The next part is getting the Robot Framework Installed on your machine using pip.

1. First, let's make sure we have the latest version of pip, by first navigating to the scripts directory within Python `cd C:\Python27\Scripts` and then entering `python -m pip install -U pip`. It should say that you have the most up to date version installed!

2. Next, lets install Robot Framework by entering `pip install robotframework`

3. Once pip has finished downloading and installing the files, enter `robot --version` to make sure it installed correctly. It should say Robot Framework 3.0 (Python 2.7.11 on win32/64)

4. (Optional) If in the future there is an update for Robot Framework, you can run this command `pip install --upgrade robotframework`

Read Getting started with robotframework online:
https://riptutorial.com/robotframework/topic/5187/getting-started-with-robotframework

Chapter 2: How robot framework is used in Automation testing in Embedded Systems?

Introduction

Robot framework is widely used in Automation testing of Embedded products. We are going to take an Embedded product as an example and see how to automate the test cases using Robot Framework.

Remarks

Abbreviation:

- RPS - Remote power supply
- RF - Robot frame work

Examples

Remote Power Supply Testing

Remote Power supply simulation

Since we don't have a real remote power supply hardware, we are going to simulate it using python program.

Basic idea about RPS

- Actually remote power supply has a http server.
- User can send commands to turn ON/OFF power supply using http request.

We are going to simulate remote power supply using following program rps-server.py.

```
from flask import Flask, request
from flask_httpauth import HTTPBasicAuth


app = Flask(__name__)
auth = HTTPBasicAuth()

users = {
        'admin': '12345678'
}
app.url_map.strict_slashes = False

PINS = ['P60', 'P61', 'P62', 'P63']
```

```
PINS_STATUS = {'P60':'0', 'P61': '0', 'P62':'0', 'P63':'0'}

@auth.get_password
def get_pw(username):
        if username in users:
                return users.get(username)
        return None

@app.route('/')
@auth.login_required
def index():
        return "Hello, %s!" % auth.username()

def get_html_string():
        html_str = '<html>P60={}P61={}P62={}P63={}</html>'.format(PINS_STATUS['P60'],
                                                    PINS_STATUS['P61'],
                                                PINS_STATUS['P62'],
                                                PINS_STATUS['P63'])
        return html_str

def parse_cmd_args(args):
        global current_status
        if str(args['CMD']) == 'SetPower':
                for key in args:
                        if key in PINS:
                                PINS_STATUS[key] = str(args[key])

                return get_html_string()

        if str(args['CMD']) == 'GetPower':
                return get_html_string()


    @app.route('/SetCmd', methods=['GET','POST'])
    def rps():
        if request.method=="GET":
                args=request.args.to_dict()
                ret = parse_cmd_args(args)
                return ret
```

The above code actually simulates http server to control the remote power supply.

How to Run RPS server ?

```
$ export FLASK_APP=rps-server.py
$ flask run
```

How to send commands to rps server ?

Following are the two commands used to control the RPS

1. SetPower
2. GetPower

By default the server will be listening at the port 5000.

The power supply ports are,

1. P60
2. P61
3. P62
4. P64

The states of the ports are,

1. ON - 1
2. OFF - 0

Requirements

Requirements for building a remote power supply are

1. Remote power supply should be able to turn ON/OFF remotely
2. Remote power supply status can be accessed remotely.

Deriving test cases

Test cases derived from requirement

1. Turn on Power supply 2 remotely.
2. Verify power supply 2 is on.
3. Turn off Power supply 2 remotely.
4. Verify power supply 2 is off.

Manual Testing

- Run the rps server.
- To turn on Port 3, open a browser and give following URI

```
http://admin:12345678@localhost:5000/SetCmd?CMD=SetPower&P62=1
```

- To get the status of all the ports

```
http://admin:12345678@localhost:5000/SetCmd?CMD=GetPower
```

Writing test library

We need to write a test library in python for sending http commands using http request. Later we will be using this library as keywords in robot frame work.

commands.py

We are going to use library from commands.py to send SetPower and GetPower.

```
import requests
import re


class commands(object):

    ROBOT_LIBRARY_SCOPE = 'GLOBAL'
    def __init__(self, ip='localhost:5000'):
        self.ip_address = ip
        self.query = {}
        self.user = 'admin'
        self.passw = '12345678'

    def form_query(self, state, cmd, port):
        port = self.get_port_no(port)
        self.query = {port: state}
        return self.query

    def get_port_no(self, port_no):
        port = 'P6' + str(port_no)
        return port

    def clean_html(self, data):
        exp = re.compile('<.*?>')
        text = re.sub(exp, "", data)
        return text.rstrip()

    def send_cmds(self, cmd, port=None, state=None):
        url = 'http://{}:{}@{}/SetCmd?CMD={}'\
              .format(self.user,
                      self.passw,
                      self.ip_address,
                      cmd)
        print url
        if cmd == 'SetPower':
            self.form_query(state, cmd, port)
            self.req = requests.get(url, params=self.query)
            return True
        elif cmd == 'GetPower':
            self.req = requests.get(url)
            data = self.clean_html(self.req.text)
            return data
        else:
            return False

        return self.req.text


# c = commands('localhost:5000')

# c.send_cmds('SetPower', 2, 1)
# c.send_cmds('SetPower', 3, 1)
# print c.send_cmds('GetPower')
```

Python key word documentation

1. `send_cmds(cmd, port=None, state=None)` is the function we are going to use.

2. While using this function in Robot key word, no need to bother about _, or `Lowercaser` or `Uppercase` in function name.

Python function will look like this while using as keyword,

```
Send Cmds       cmd   port  state
```

Writing test Keywords

We are going to use `Send Cmds` as python keyword in test suite.

- RPS send commands uses following four arguments to set power
    - command = SetPower
    - port = 2
    - state = 1 for ON / 0 for off When we call that command it will turn ON/OFF the power supply
- RPS get power will return the status of all the Power supply ports

```
*** Keywords ***
RPS send commands
    [Arguments]    ${command}    ${port}    ${state}
    ${output}=    Send cmds    ${command}  ${port}  ${state}
    [return]    ${output}

RPS get Power
    [Arguments]    ${command}
    ${output}=    Send cmds    ${command}
    [return]    ${output}}
```

Algorithm to test power supply

1. Set power to a port
2. Check the status of cmd
3. Get the status of the port and check whether it is ON/OFF

Writing test cases using the above key words

Now we are ready to write test case using following two keywords

- RPS send commands - To set and unset a power of port
- RPS get power - To get the status of all the port

```
*** Settings ***
Library      commands.py

*** Test Cases ***
Turn on Power supply 2 remotely
     ${out}=    RPS send commands     SetPower  2  1
```

```
     Should be equal    ${out}  ${True}

Verify power supply 2 is on
    ${out}=     RPS get power    GetPower
    should contain    ${out}  P62=1

Turn off Power supply 2 remotely
     ${out}=    RPS send commands     SetPower  2  0
     Should be equal    ${out}  ${True}

Verify power supply 2 is off
    ${out}=     RPS get power    GetPower
    should contain    ${out}  P62=0
```

Create a file name `remote-power-supply.robot`

Copy above key words and test case in to the file.

How to execute RPS server and remote-power-supply.robot ?

- Run remote power supply first
- Run the test suite remote-power-supply.robot

```
$ export FLASK_APP=rps-server.py
$ flask run
$ pybot remote-power-supply.robot
```

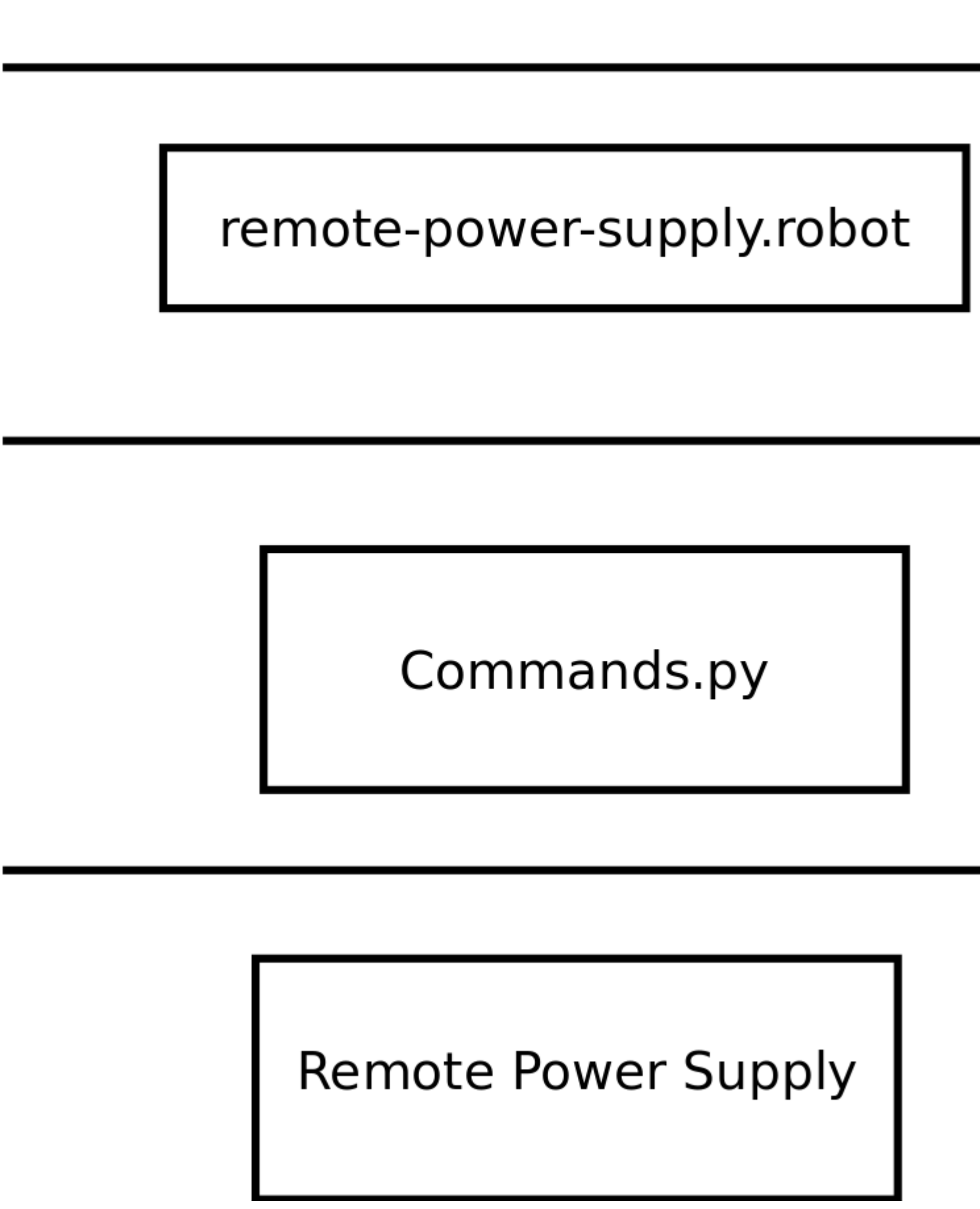
Output

```
$ pybot remote-pwer-supply.robot
==============================================================================
Remote-Pwer-Supply
==============================================================================
Turn on Power supply 2 remotely                                       | PASS |
------------------------------------------------------------------------------
Verify power supply 2 is on                                           | PASS |
------------------------------------------------------------------------------
Turn off Power supply 2 remotely                                      | PASS |
------------------------------------------------------------------------------
Verify power supply 2 is off                                          | PASS |
------------------------------------------------------------------------------
Remote-Pwer-Supply                                                    | PASS |
4 critical tests, 4 passed, 0 failed
4 tests total, 4 passed, 0 failed
==============================================================================
Output:  /tmp/talks/robot-framework-intro/test-cases/output.xml
Log:     /tmp/talks/robot-framework-intro/test-cases/log.html
Report:  /tmp/talks/robot-framework-intro/test-cases/report.html
```

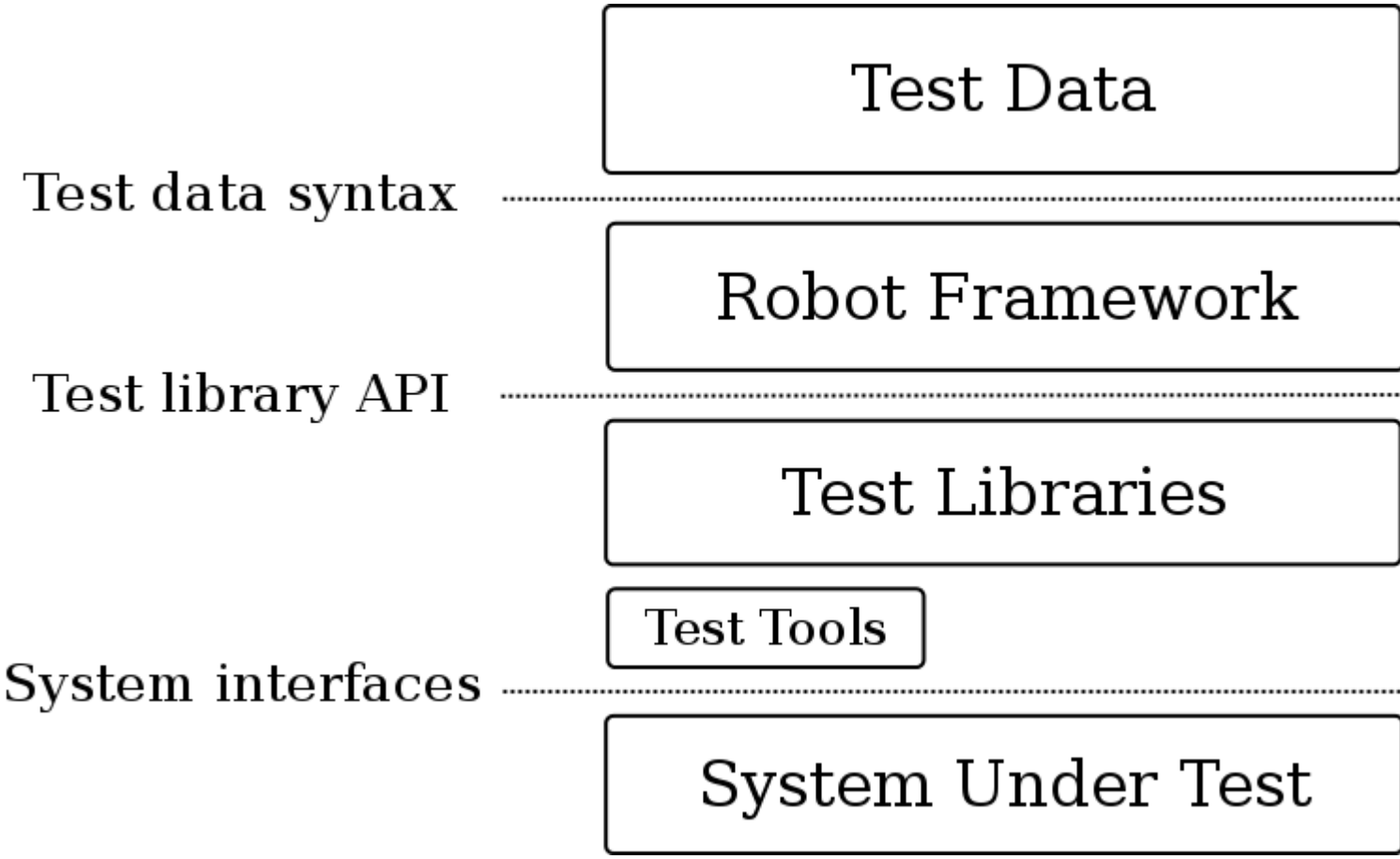
Following two diagrams explains about test

architecture between RPS and RF

Remote Power supply test architecture


commands, state, port no
remote-power-supply.robot
Commands.py
Remote Power Supply

Robot frame work architecture



Credits

Thanks to robot framework for architecture diagram.

The complete code is available here

Remote power supply

commands.py

remote-power-supply.robot

Read How robot framework is used in Automation testing in Embedded Systems? online: https://riptutorial.com/robotframework/topic/10672/how-robot-framework-is-used-in-automation-testing-in-embedded-systems-

Credits

S. No	Chapters	Contributors
1	Getting started with robotframework	Community, Goralight, Hariprasad, Shijo
2	How robot framework is used in Automation testing in Embedded Systems?	kame, sakthirengaraj