txThings Tutorial

In this tutorial, we will describe how to run a CoAP server on Raspberry Pi, and run the CoAP client on a PC. The CoAP server and clients are implemented using txThings, which is a Phyton implementation of CoAP.

# About txThings

txThings - CoAP library for Twisted framework

txThings is a Python implementation of Constrained Application Protocol (CoAP):

<http://tools.ietf.org/html/rfc7252>

txThings is based on Twisted - asynchronous I/O framework and networking engine written in Python.

<http://twistedmatrix.com/>

txThings uses MIT License (like Twisted itself).

<http://opensource.org/licenses/mit-license.php>

Copyright (c) 2012 Maciej Wasilak

<http://sixpinetrees.blogspot.com/>

# txThings Python Library Features

txThings has the following features:

* support for draft-ietf-core-coap-13 - including automatic piggyback/separate response handling. No caching support.
* support for draft-ietf-core-block-12 (no support for server initiative though - waiting for the resolution)
* limited support for RFC6690 (Core Link Format) - server only.

Other nice things:

* txThings works nicely on RaspberryPi
* txThings is compatible with [Kivy](http://kivy.org/) - brilliant new Python GUI library (I'll post some examples soon).
* txThings is fully asynchronous (thanks to twisted framework)

# txThings Installation HowTo

txThings is posted on [Github](https://github.com/siskin/txThings). The easiest way to get it is to clone the repository to your local machine using the command below:

*git clone git://github.com/siskin/txThings.git*

Library contains CoAP code (inside "iot" directory) and three examples:

* server.py - CoAP server that starts on localhost, port 5683 and hosts several resources
* client\_GET.py - example client which performs GET request to localhost, port 5683
* client\_PUT.py - example client which performs PUT request to localhost, port 5683

Client\_GET and client\_PUT both use port 61616 - to use them simultaneously change port number in one of the clients. Server will send blockwise responses for default settings. To use txThings you need Python 2.7 with Twisted installed (I suggest using the latest Twisted version, but older releases also work - tested with 11.1).

# Setting up Raspberry-Pi

1. Install OS on Raspberry Pi (set the appropriate raspi-config for all,including ssh and desktop boot)
2. Install tightvncserver – optional
3. Install setup tools for python (such as pip)

<https://pip.pypa.io/en/stable/installing.html#install-pip>

1. Install Twisted on Raspberry Pi
   1. *sudo apt-get install python-pip*
   2. *pip install twisted==15.1.0*
2. Install txThings on Raspberry Pi
   1. *pip install txThings*

# Setting up PC

1. If using Windows, install Cygwin (with all the necessary packages including git, wget, ssh)
2. Install txThings and all the required dependencies (using git on Cygwin and pip)
3. Install Copper on Firefox as a GUI client – optional

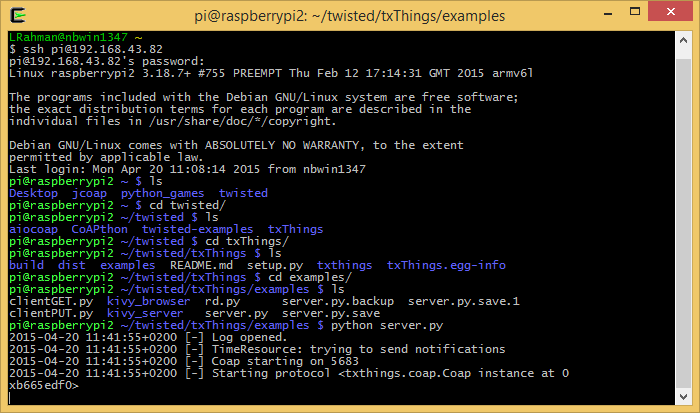
<https://addons.mozilla.org/en-US/firefox/addon/copper-270430/>

1. Install Wireshark – optional

# Run the CoAP Server on Raspberry-Pi

1. From the Raspberry’s console or using *ssh* to the Raspberry-Pi, go to the examples directory at : *Twisted/txThings/examples*
2. Run the CoAP server : *python server.py*

(See **Figure 1**)



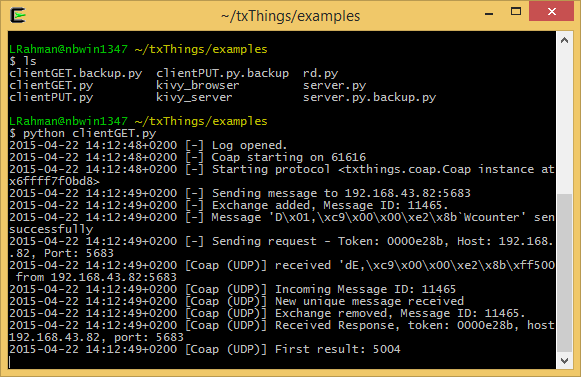
**Figure 1. Running the CoAP Server on Raspberry-Pi using SSH**

# Run CoAP Client on PC

1. If using Windows, open Cygwin
2. Go to the examples directory at : *Twisted/txThings/examples*
3. Change the resource name and remote IP address in the source code of *client\_GET.py* or *client\_PUT.py* (see **Figure 2**)
4. Run the CoAP clients (see **Figure 3**):
   1. For GET operation: *python client\_GET.py*
   2. For PUT operation: *python client\_PUT.py*



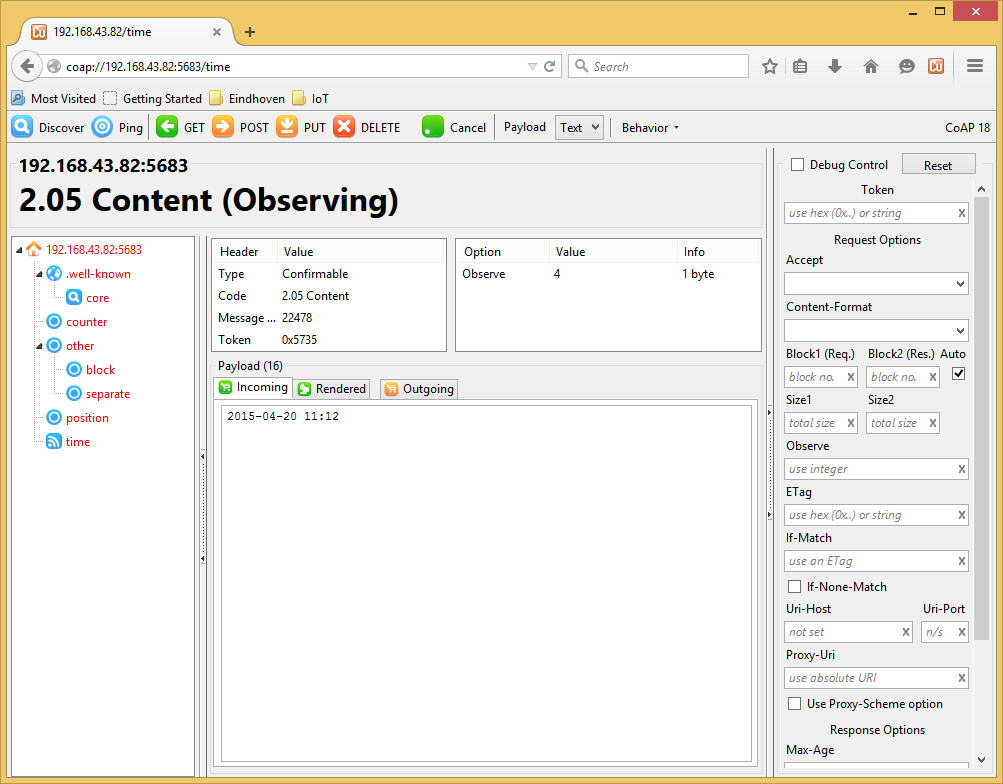
**Figure 2. Change the resource name and remote IP Address in the source code of *client\_GET.py***



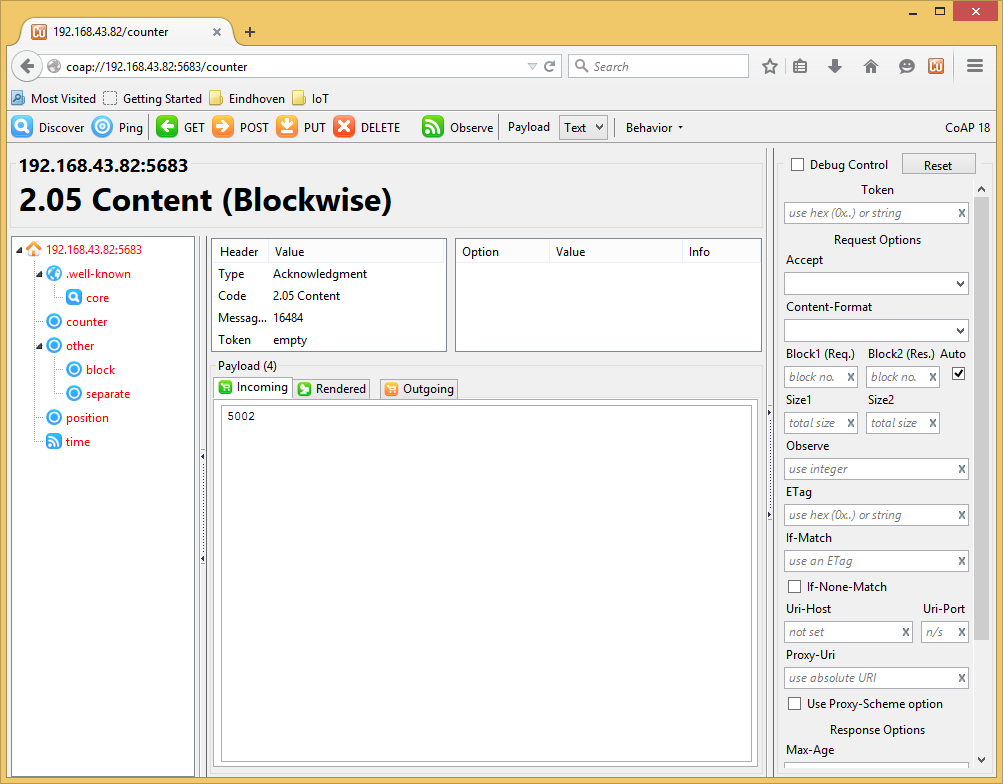
**Figure 3. Run *client\_GET.py* on a Windows PC using Cygwin**

# Run Copper on Firefox

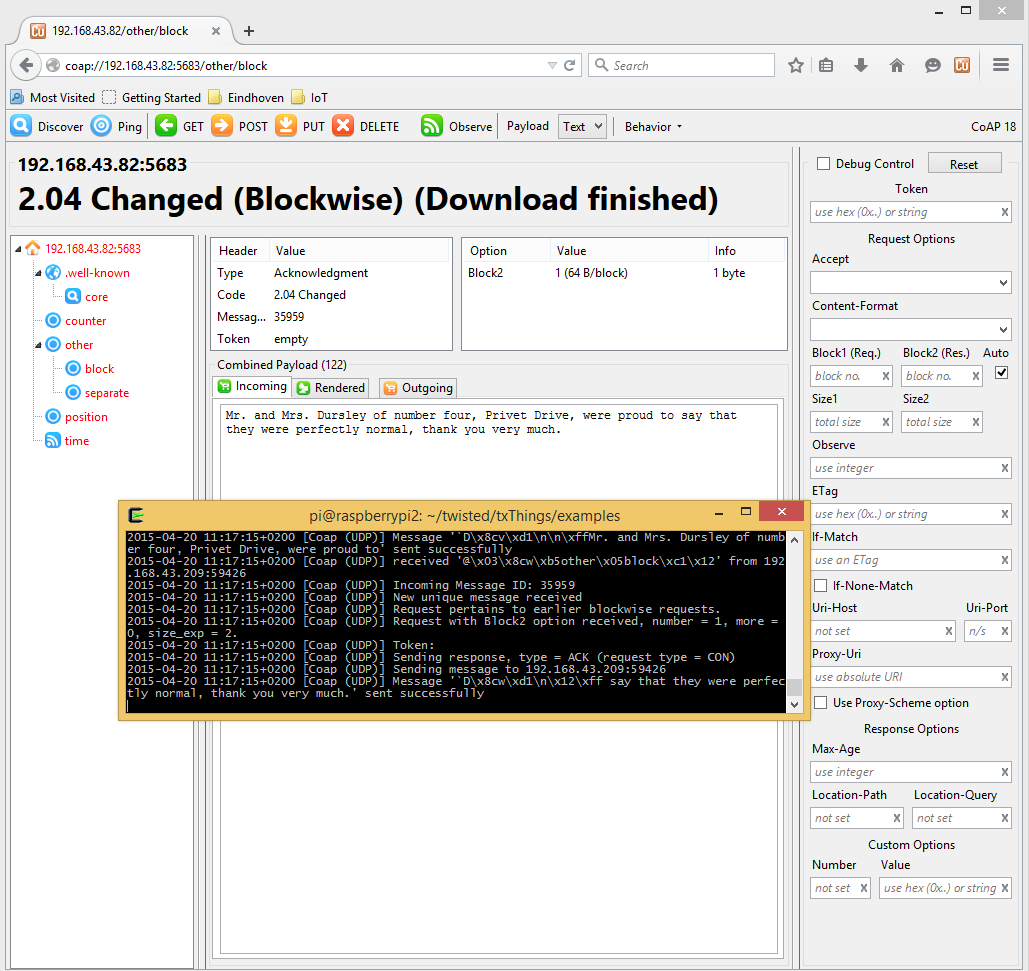
The Copper (Cu) CoAP user-agent is an add-on for the Firefox Web browser. It allows browsing, bookmarking, and direct interaction with CoAP resources. Simply enter a CoAP URI into the address bar (See **Figure 4**, **Figure 5** and **Figure 6**)



**Figure 4. Running the OBSERVE operation on the resource “time” using Copper**



**Figure 5. Running the GET operation on the resource “counter” using Copper**



**Figure 6. Running the PUT operation on the resource “block” using Copper**