Python Code for FD100 Control

Harry Li ‡, Ph.D.
Computer Engineering Department, San Jose State University
San Jose, CA 95192, USA

Email†: harry.li@ctione.com

***Abstract*—This note describes the first python programming for FD100 robot control.**

I. Setup the LAN Network

If you have already set up the LAN connection to the Robot, then you can skip the network setup sections, and go to the python code section. Scan IP addresses for all the devices on the LAN which FD100 and your computer are connected on. To check your computer IP address, use "ifconfig" command. Note your computer is Linux OS based system.

$ifconfig

To check all the devices attached to your LAN, use "arp -a" command.

$arp -a

The above command "arp" may not able to list all the IP devices on LAN, so log into your AP router to search under "status" to verify all IP devices on the LAN.

The AP router usually with IP address ends with "1", for example, 192.168.0.1 for Palo Alto site setting, so you can go with the following steps:

Step 1. pointing your browser to "http://192.168.0.1" user id: admin (by dafault), then enter password;

Step 2. Once logged in, select "Basic Router > DHCP" , then you should be able to see the pop up window with all the IP devices as shown in the figure below.

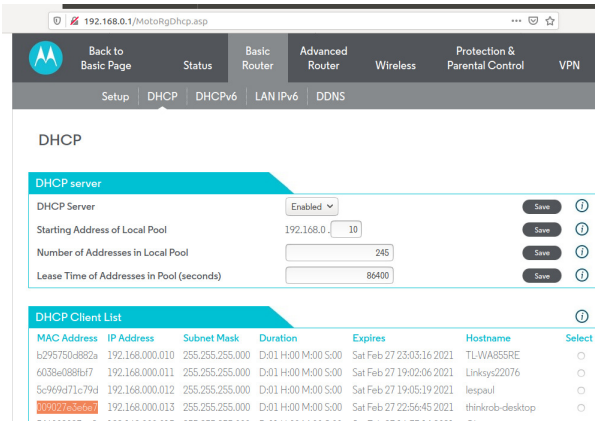

Fig. 1. Once enter the AP, select "Basic Router > DHCP".

The LAN network setup is described in the following figure.

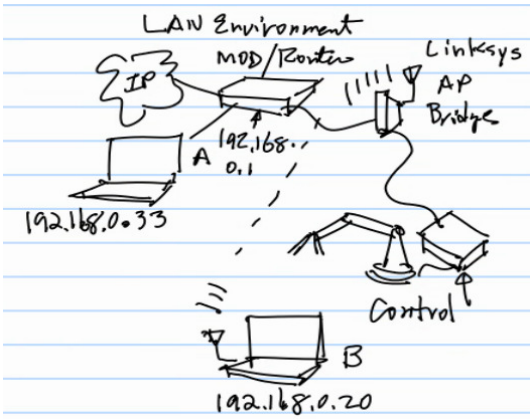

Fig. 2. LAN network architecture for FD100 at Palo Alto Site.

II. Python Testing Program to Run FD100

Fig. 3. Python Programming to Run FD100.

A. Network Connection Verification

To run python code to control FD100 robot, first, you will have to make sure that your computer is connected to FD100. The connection is established via ethernet cable, and the LAN archiecture is given in the previouse section, see Figure 2.

To confirm the robot networking part is working, use telnet to verify:

$telnet 192.168.0.13 8527

where assuming FD IP address is 192.168.0.13, and always use port 8527 [Yunxiao Jia, 2019].

When the telnet connection is successful, you should be able to see message on your console "telnet is connected". Then you can kill the telnet program and start runing the python program as

$python3 turin_telnet.py

At the prompt of the program, you can enter your selection 1 or 2 to run pre-programmed trajectory 1 or 2. You can enter 3 to stop the program.

III. Python Code for Robot Control

We will first give the full listing of a python program turin_telnet.py below.

```
import getpass
import sys
import telnetlib
import time

HOST = '192.168.0.13'
PORT = 8527
tn = telnetlib.Telnet(HOST, PORT, timeout = 1)
print()
print("Connected : ", tn)
print()
#print (tn.read_all())
login_str = str('<Bodys> <Cmd Name="Login"
            Status="Send"><Param
            UserName = "administrator"
            Password="12345678"/></Cmd>
            </Bodys>')
print(login_str)
print("Authentication Successful")
tn.write((login_str+"\n").encode('ascii'))
print("Logged In to the TURIN System")

while True:
    print("Available Tasks:")
    print("1. Trajectory1")
    print("2. Trajectory2")
    print("3. Stop\n")
    print("Enter Task No: ")
    task=int(input())

    if task==1:
        action_str = str('<Bodys><Cmd
                  Name="MotionEnd"
                  Status="Send"/></Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        time.sleep(1)
        action_str = str('<Bodys>
                  <Cmd Name="MotionBegin"
                  Status="Send"> <Param FileName =
                  "test1-05-03-2019.txt"
                  StartLine = "0"/>
                  </Cmd> </Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        print("Task1 sent.\n\n")
        #print (tn.read_until(b'/Bodys'))

    if task==2:
        action_str = str('<Bodys><Cmd Name="MotionEnd"
                  Status="Send"/></Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        time.sleep(1)
        action_str2 = str('<Bodys>
                   <Cmd Name="MotionBegin"
                   Status="Send"> <Param FileName =
                  "test1-05-17-2019.txt"
                  StartLine = "0"/>
                  </Cmd> </Bodys>')
        tn.write((action_str2+"\n").encode('ascii'))
        print("Task2 sent.\n\n")

    if task==3:
        action_str = str('<Bodys><Cmd Name="MotionEnd"
                  Status="Send"/></Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        print("ARM Mobility Stopped.")
```

A. Pseudo Code

Add Pseudo code here.

B. Code Walk Through

Now, we walk through the code.

1) Telnet Connection and Login: First, telnet connection is given below.

```
import getpass
import sys
import telnetlib  #for telnet networking
import time

HOST = '192.168.0.13' #for Palo Alto Site LAN sittings
PORT = 8527  #FD100 desinated port
tn = telnetlib.Telnet(HOST, PORT, timeout = 1)
print()
print("Connected : ", tn)
print()
```

Now the login once the telnet connection is established.

```
login_str = str('<Bodys> <Cmd Name="Login"
            Status="Send"><Param
            UserName = "administrator"
            Password="12345678"/></Cmd>
            </Bodys>')
print(login_str)
print("Authentication Successful")
tn.write((login_str+"\n").encode('ascii'))
print("Logged In to the TURIN System")
```

Note the command syntax in the following table.

TABLE I
Command Syntax

Syntax Description	Note
<Bodys> ... </Bodys>	start-end with pair
<Cmd Name="Login" Status="Send">	
<Param UserName = "xxx" Password="xxx"/>	end </Cmd>

2) Move the Robot:

```
while True:
    print("Available Tasks:")
    print("1. Trajectory1")
    print("2. Trajectory2")
    print("3. Stop\n")
    print("Enter Task No: ")
    task=int(input())

    if task==1:
        action_str = str('<Bodys><Cmd
                  Name="MotionEnd"
                  Status="Send"/></Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        time.sleep(1)
        action_str = str('<Bodys>
                  <Cmd Name="MotionBegin"
                  Status="Send"> <Param FileName =
                  "test1-05-03-2019.txt"
                  StartLine = "0"/>
```

```
            </Cmd> </Bodys>')
        tn.write((action_str+"\n").encode('ascii'))
        print("Task1 sent.\n\n")
        #print (tn.read_until(b'/Bodys'))
```

Let's take a look at the first part of this while-loop, action_str is defined to give "MotionEnd"

TABLE II
MOTIONEND

Syntax
<Bodys> ... </Bodys>
<Cmd Name="MotionBegin"
Status="Send"
<Param FileName = "test1-05-03-2019.txt" StartLine = "0"/>

Then telnet write (send) is given below

`tn.write((action_str+"\n").encode('ascii'))`

Now, let's take a look at MotionBegin which will allows pre-recorded motion trajectory file in txt format to be send to the robot controller for execution.

TABLE III
MOTIONBEGIN

Syntax
<Bodys> ... </Bodys>
<Cmd Name="MotionEnd"
Status="Send"

ACKNOWLEDGMENT

I would like to express my thanks to Allen Lee for his 3D printer fine-tuning for the FD100 robot gribber parts.

REFERENCES

[1] [Yunxiao Jia, 2019] CTI One Technical Note, Lec Turin Robot User Guide-YJ-HL-2019-03-04.ppt-2, Minh Passdown, FD100, Feb, 2021.