Networking with Python

In this resource, you will learn about computer networks and how to send messages to other devices on the network.

Introduction

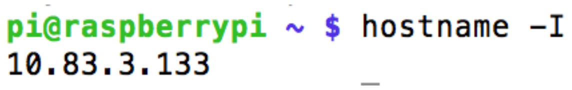
Computer networks are everywhere! Nearly every computer in your school will be connected together via 1 big network, if you have wireless or wired internet at home, that means your devices connecting to it are all networked together. On a bigger scale though, mobile phones are all connected to a big network, or else how would you be able to phone up your friend?

All the machines in the room today are connected to each other, meaning you can send messages to them. Although blindly sending a message to a random computer on the network is unlikely to do anything, unless it is listening on that port for that message.

Talking to other machines

Today we are going to be sending messages between each other and to a central server connected to a very special device.

First though, you need to understand how networks work.
Each device on the network has an address, kind of like an address in real life.
This is called an IP address. An IP address is 4 numbers with a dot in between each number. For example, 192.168.1.42 is an IP address. Usually, the first 3 numbers will be the same for every device on the network, while the 4th number will change.
To get our IP address, open an LXTerminal and type `hostname -I` then hit enter. You should get something like this.

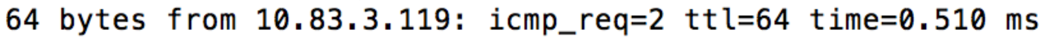


```
pi@raspberrypi ~ $ hostname -I
10.83.3.133
```

If you get 127.0.0.1, this means you aren't connected to a network.
Lets try checking if we can send a message to someone else in the room. To do this, we will use ping! Ping is very simple, it sends a message basically saying "Are you there?". If you get a reply, you know there is. To ping a friend, get their IP address off them, open an LXTerminal and type `ping 1.1.1.1` (replacing 1.1.1.1 with your friends IP address).

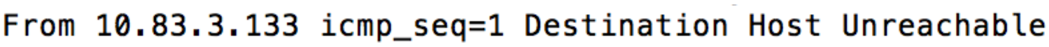
By Andrew Mulholland.
Based off https://www.raspberrypi.org/learning/networking-lessons/
Northern Ireland Raspberry Jam

You should get something like this.



```
64 bytes from 10.83.3.119: icmp_req=2 ttl=64 time=0.510 ms
```

If you get something like this, you typed their IP address in wrong.



```
From 10.83.3.133 icmp_seq=1 Destination Host Unreachable
```

Sending information

Pings are great, but they only check if there is something at the other end, what if we want to send them some information? Perhaps a message?
For that, we need to write Python program.

On the desktop, double click IDLE 3 to launch IDLE (for Python 3).
Next, click file and new window.
We are now ready to create our program. We will need to create 2 programs, one for sending data and one for receiving it.

Receiving

First, lets build our receiver, or server.
In a new window in IDLE 3, add the code in grey below

```
import network
```

This imports the network library so we can use it.

```
def heard(message):
    print("I got something " + message)
```

Creates a new function called heard that will display whatever message s passed to it via the message variable.

```
network.wait(whenHearCall=heard)
```

Network.wait starts a process in the background that waits for new messages, when one is received it runs the heard function.

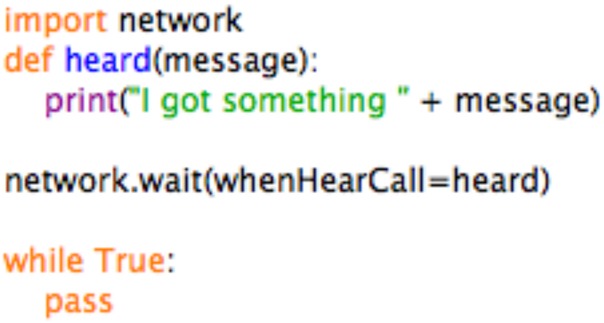
```
while True:
    pass
```

Finally, we add an infinite loop (while True). As loops must have at least 1 command inside them, pass is used as it does nothing.

By Andrew Mulholland.
Based off https://www.raspberrypi.org/learning/networking-lessons/
Northern Ireland Raspberry Jam

The final sending program should look something like this.

When you are done, select File – Save as and give it a useful name. For example, awesomeServer.py. Don’t forget to add .py at the end!



```
import network
def heard(message):
    print("I got something " + message)

network.wait(whenHearCall=heard)

while True:
    pass
```

Sending

Now that we have our awesome server done, we need to make the sender.
Again, select File – New Window to create a new blank Python program.
In the new window, add the code in grey below.

`import network` This imports the network library so we can use it.

`ipAddress=”`Put your friends IP address here in speech marks`”`

`network.call(ipAddress)` Opens a connection to your friends Raspberry Pi.

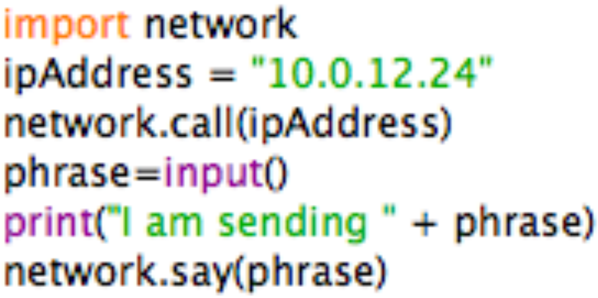
`phrase=input()` Wait for user to type something on keyboard, then stores it into the variable called “phrase” to be used later.

`print(“I am sending “ + phrase)` Outputs “I am sending” followed by what you just typed.

`network.say(phrase)` Sends message to other Raspberry Pi.

The final sending program should look something like this.

When you are done, select File – Save as and give it a useful name. For example, awesomeSender.py. Don’t forget to add .py at the end!



```
import network
ipAddress = "10.0.12.24"
network.call(ipAddress)
phrase=input()
print("I am sending " + phrase)
network.say(phrase)
```

Trying it out

Now its time to try out your code, pick someone else in the room and start your awesomeServer.py program, get them to put in your IP address into their program and give it a go, then swap around after.

Morse code

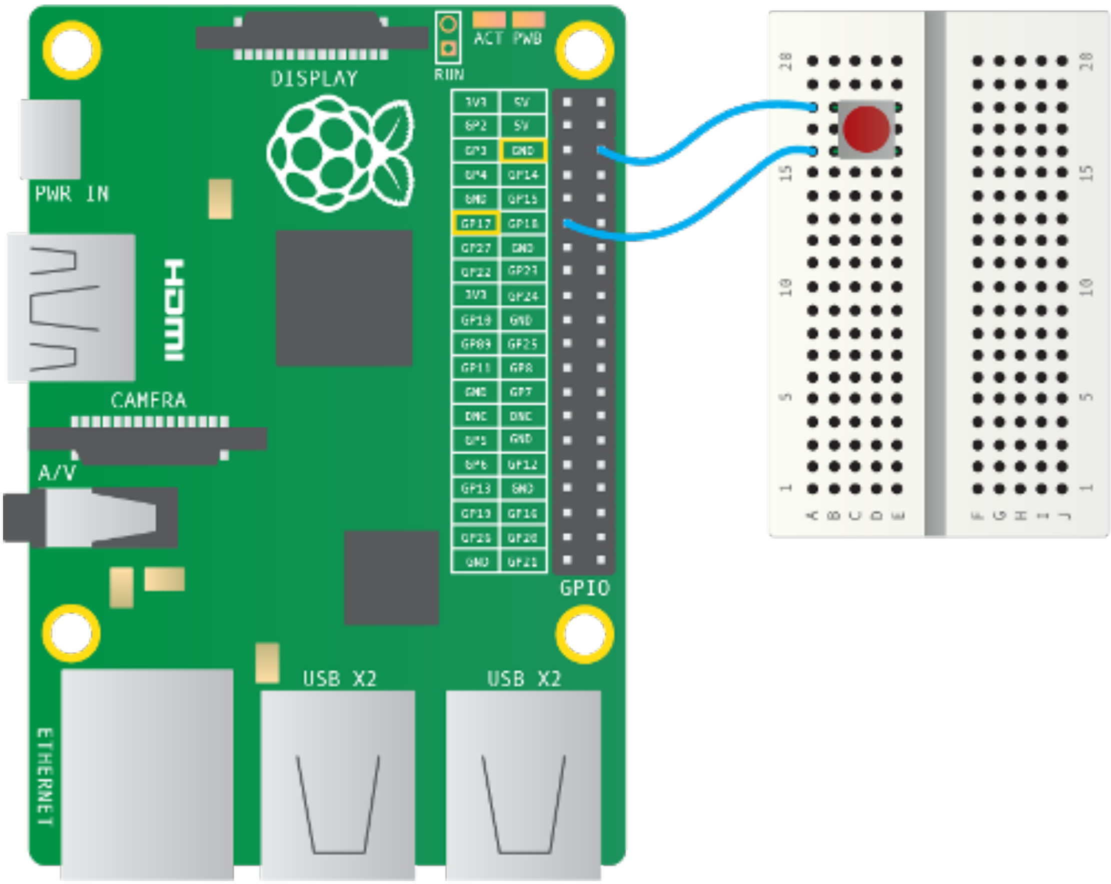
Now that we can send messages between Raspberry Pis, what about building a Morse code like system for sending messages using a single button!

What do you need?

You will need

- A breadboard
- A button
- 2 male to female jumper cables

Wire it up like below

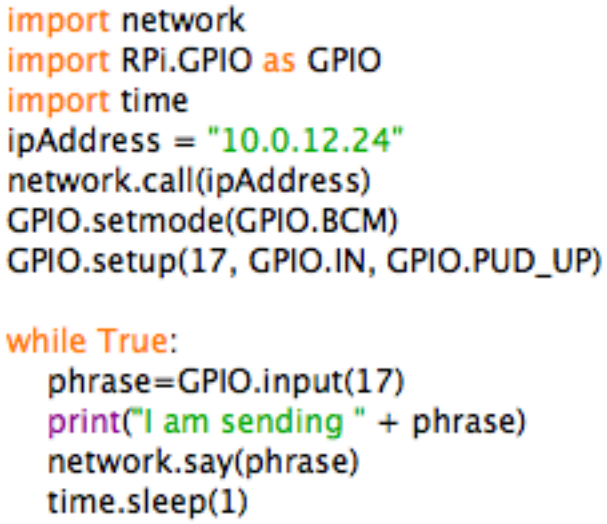


By Andrew Mulholland.
Based off https://www.raspberrypi.org/learning/networking-lessons/
Northern Ireland Raspberry Jam

Code

We don’t need to change our server code, but we will need to change our sender code to check for button presses.
The basic idea is we get the Raspberry Pi to check the current status of the button every second, if it is not pressed, send a 0, it is pressed, send a 1.

Add this new code as in the program below.



```
import network
import RPi.GPIO as GPIO
import time
ipAddress = "10.0.12.24"
network.call(ipAddress)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN, GPIO.PUD_UP)

while True:
    phrase=GPIO.input(17)
    print("I am sending " + phrase)
    network.say(phrase)
    time.sleep(1)
```

The 3 additional GPIO commands (GPIO.setmode, GPIO.setup and GPIO.input) are for setting up GPIO pin 17 as an input, then reading it.
Time.sleep(1) pauses the program for 1 second.

Trying it out

With your friend, try sending messages to each other. One will need to be running the server and the other will need to be running the sender.

Things to try

Can you create a code for certain letters or words? For example, 0001 = A?
Create your own unique code and try sending encoding messages to your friend with it. See if they can figure them out.

By Andrew Mulholland.
Based off https://www.raspberrypi.org/learning/networking-lessons/
Northern Ireland Raspberry Jam

Controlling a big traffic light

Now that you are an expert in sending messages across a network, what about using those messages to control something big?

Traffic light

The traffic light contains a Raspberry Pi inside running similar code to that you have been working on, although it can accept multiple connections at the same time from different people. This means you can all control the same traffic light at the same time!

Code

Using your awesomeSender.py (or whatever you called yours) as a base, all you need to do to send messages to the traffic light is enter the IP address of the traffic light in the `ipAddress = “”` section and then use commands like `network.say(“greenOn”)`

It supports a number of different commands

Command	**What it does**
GreenOn	Turns green light on.
GreenOff	Turns green light off.
AmberOn	Turns amber light on.
AmberOff	Turns amber light off.
RedOn	Turns red light on.
RedOff	Turns red light off.
AllOn	Turns all 3 lights on.
AllOff	Turns all 3 lights off.

Things to try

Can you

- Make the traffic light red light flash a few times?
- Make each coloured light flash a few times one after another?
- Make the traffic light perform a normal stop/start sequence?

By Andrew Mulholland.
Based off https://www.raspberrypi.org/learning/networking-lessons/
Northern Ireland Raspberry Jam