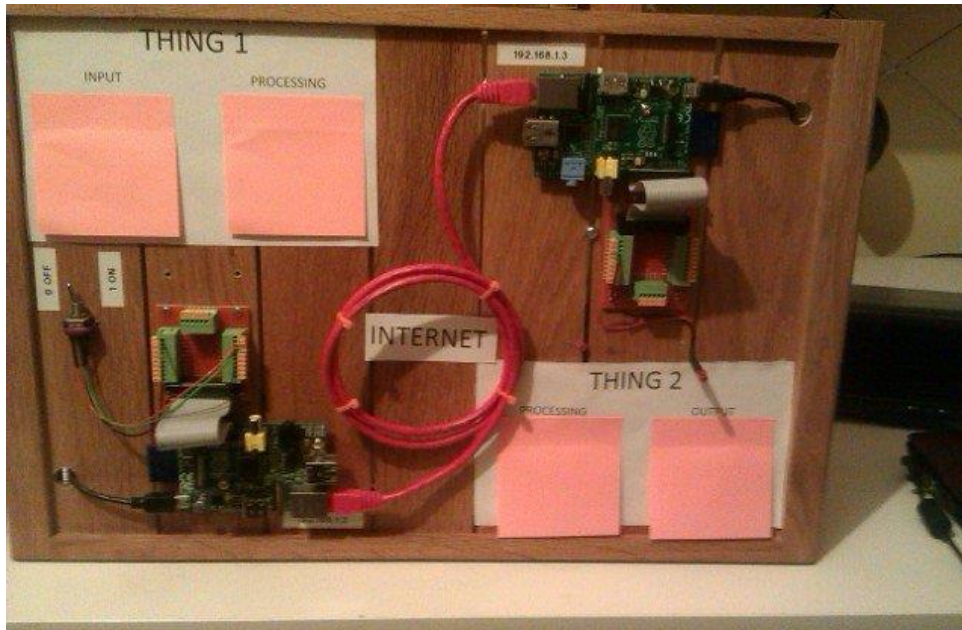


# Raspberry Pi – Network connected hardware

Once you know how to write a network server and a network client on your Raspberry Pi in Python, it is easy to connect up bits of hardware to it. This will allow you to build bigger systems that sense the environment, send a message over the internet, and then control some hardware elsewhere in the world.

## Our “Internet of things” demonstrator



## What it does

The “internet of things” demonstrator shows a switch on the left connected to one Raspberry Pi computer. This connects over our “internet” to a second Raspberry Pi computer with an LED. If you turn the switch on and off, the LED goes on and off.

We used our own personal internet for this demonstrator, but it is just as easy to connect the two Raspberry Pi computers to the real internet either by plugging them into your router, or by using a WiFi adapter. You will have to change the IP addresses used when on the real internet, and if you want the server to be visible outside of your house, you will have to configure a “port forward” on your router to route port 8888 through to your raspberry Pi’s IP address – but then the code will work in exactly the same way, from one side of the planet to the other!

## What could you invent?

Think about what the switch could represent – it could be measuring the heartbeat of a patient as they live their daily life. The computer could be monitoring the heartbeat and if it is too slow or too fast, send a message over the internet.

Think about what the LED could represent – the second Raspberry Pi computer could be a computer at the hospital that receives heartbeat information from thousands of patients living their daily lives. If it receives a message saying there is a problem, it could look up the patient record, contact their doctor, and the LED could represent a message that pops up on the doctors mobile phone saying “this patient needs some help, here is their phone number and their health history from their patient records”.

## The Python Code

Here are the program listings for the net\_led.py and net\_switch.py python code that was used to build this demonstrator. You will recognise a lot of the code from your internet chat.py program.

### The net\_switch.py program

```
import RPi.GPIO as GPIO
import time
import network

SWITCH = 10
GPIO.setmode(GPIO.BCM)
GPIO.setup(SWITCH, GPIO.IN)

def heard(phrase):
    if (GPIO.input(SWITCH)):
        network.say("1")
    else:
        network.say("0")

while True:
    network.wait(whenHearCall=heard)
    while network.isConnected():
        time.sleep(1)
```

### How it works

This is a server program just like server.py. It waits for an incoming connection from the network.

Then, all the while the client and server are connected, if it receives any message from the network, it reads the present state of the switch, and then sends a message back with a “1” for on and a “0” for off.

It is the responsibility of the client program to “poll” the server at regular intervals asking it for the status of the switch.

If the client disconnects from the server, this program will go round the outer while loop again and wait for a new incoming connection.

## The net\_led.py program

```
import RPi.GPIO as GPIO
import time
import sys
import network

SERVER_IP = sys.argv[1]
LED = 11
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED, GPIO.OUT)

def heard(phrase):
    a = phrase[0]
    if a == "0":
        GPIO.output(LED, False)
    else:
        GPIO.output(LED, True)

network.call(SERVER_IP, whenHearCall=heard)
while network.isConnected():
    network.say("?")
    time.sleep(1)
```

## How it works

The led program is a client program, just like client.py.

It accepts an IP address of the server, on the command line using sys.argv, just like our chat.py did. It connects to our server, and then once per second it sends a “?” to the server. This will trigger the server to send us back the status of the switch as a “1” or a “0”.

When the response message comes back, we look at the first character of the message and if it is a “0” we turn the LED off, if it is a “1” we turn the LED on.

You can now see that it is easy to create a “connected product” that senses or controls some hardware and connects over the internet.

Use your imagination – the switch and the LED could represent anything you could imagine, and the two computers could be opposite sides of the planet – what could you invent?

**That’s it! You now know everything you need to invent an awesome internet connected product!**