Difficulty level: Beginner

Approx reading time:

**Components Required:**

**1.** Raspberry Pi 3 model B – 2

**2.** MicroSD card 8 or 16 GB (Class 4 and above) with Raspbian – 2

**3.** Windows PC / Linux PC (Tested on Windows 10, Ubuntu 14.04 LTS)

**4.** Ethernet cable (Category 5 – also called Cat 5) – 2

**5.** Micro USB cable – 2

**6.**Breadboard – 2

**7.**[NRF24L01+ RF Transceiver](http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01) – 2

**8.**Jumper Wires (Male to Female, Male to Male)

**Way to go ->**

1. Login to both your devices using PuTTy or your Ubuntu terminal.

**2.**Login to both your devices’ GUIs using VNC server (Linux / Windows).

**Do the following for both the devices:**

**3.** In the terminal type:

sudo raspi-config

and press **Enter**on your PC keyboard.

**Follow steps 7 – 9 of [RPi 3 Tutorial – 11](http://invent.module143.com/daskal_tutorial/rpi-3-tutorial-11-gpio-analog-sensor-mcp3008-mcp3004/" \t "_blank) for steps 4 – 6.**

**4.**Go to **Advanced Options**.

**5.**Go to **SPI**.

**6.**Enable **SPI Interface**by clicking on **<Yes>**.

**7.**Reboot the Pi. In the terminal, type:

sudo reboot

**8.**In the terminal type:

sudo apt-get update

**9.** After it has rebooted, install Python development tools. In the terminal, type:

sudo apt-get install python-dev. # python2

sudo apt-get install python3-dev. # python3

**10.** Download this package. In the terminal type:

wget https://github.com/Gadgetoid/py-spidev/archive/master.zip

**11.**In the terminal type: **ls**to check that you have **master.zip**in your current working directory.

**12.**To unzip, in the terminal type:

unzip master.zip

**13.**Remove master.zip:

rm master.zip

**14.**Navigate to py-spidev-master directory(folder):

cd py-spidev-master

**15.**Type in the terminal: **ls**, to view the contents of the directory(folder).

**16.**In the terminal run the commands:

sudo python setup.py install # python2

sudo python3 setup.py install # python3

**17.**Navigate to the desktop by running the following commands. In terminal type:

cd ..

Press **Enter**on your PC keyboard.

cd Desktop/

Press **Enter**on your PC keyboard.

**18.**Create the following directory(folder) and navigate to it. In terminal type:

mkdir NRF24L01

Press **Enter**on your PC keyboard.

cd NRF24L01/

Press **Enter**on your PC keyboard.

**19.**In the terminal type:

git clone https://github.com/Blavery/lib\_nrf24

**20.**Navigate to directory just downloaded. In the terminal type:

ls

Press **Enter**on your PC keyboard.

cd libnrf24/

Press **Enter**on your PC keyboard.

ls

Press **Enter**on your PC keyboard.

**21.**Copy lib\_nrf24.py to the NRF24L01 directory. In the terminal type:

cp lib\_nrf24.py ~/Desktop/NRF24L01/

**22.**Navigate to the NRF24L01 directory:

cd..

ls

**23.**In **Python 3 (IDLE)**, create a new file.

**24.**The names could be: **TransmitPi.py**(in one of the devices) and **Receive.py**(in the other device). Save them in the **NRF24L01** directory in both the devices.

**25.**In the file **TransmitPi.py**, write the following code with comments (line starting with **“#”**) for clear understanding and save (press **Cntrl + S** on your PC keyboard) the file.

import RPi.GPIO as GPIO

from lib\_nrf24 import NRF24

import time

import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xe7, 0xe7, 0xe7, 0xe7, 0xe7], [0xc2, 0xc2, 0xc2, 0xc2, 0xc2]]

radio = NRF24(GPIO, spidev.SpiDev())

radio.begin(0, 17)

radio.setPayloadSize(32)

radio.setChannel(0x60)

radio.setDataRate(NRF24.BR\_2MBPS)

radio.setPALevel(NRF24.PA\_MIN)

radio.setAutoAck(True)

radio.enableDynamicPayloads()

radio.enableAckPayload()

# radio.openReadingPipe(1, pipes[1])

radio.openWritingPipe(pipes[1])

radio.printDetails()

# radio.startListening()

# message = list(input("Enter a message to send: "))

while(1):

message = list("Hello World is awesome")

radio.write(message)

print("We sent the message of {}".format(message))

# Check if it returned ackPL

if radio.isAckPayloadAvailable():

returnedPL = []

radio.read(returnedPL, radio.getDynamicPayloadSize())

print("Our returned payload was {}".format(returnedPL))

else:

print("No payload received")

time.sleep(1)

**26.**In the file **ReceivePi.py**, write the following code with comments (line starting with **“#”**) for clear understanding and save (press **Cntrl + S** on your PC keyboard) the file.

import RPi.GPIO as GPIO

from lib\_nrf24 import NRF24

import time

import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xe7, 0xe7, 0xe7, 0xe7, 0xe7], [0xc2, 0xc2, 0xc2, 0xc2, 0xc2]]

radio = NRF24(GPIO, spidev.SpiDev())

radio.begin(0, 17)

radio.setPayloadSize(32)

radio.setChannel(0x60)

radio.setDataRate(NRF24.BR\_2MBPS)

radio.setPALevel(NRF24.PA\_MIN)

radio.setAutoAck(True)

radio.enableDynamicPayloads()

radio.enableAckPayload()

radio.openReadingPipe(0, pipes[1])

radio.printDetails()

radio.startListening()

while(1):

ackPL = [1]

while not radio.available(0):

time.sleep(1 / 100)

receivedMessage = []

radio.read(receivedMessage, radio.getDynamicPayloadSize())

print("Received: {}".format(receivedMessage))

print("Translating the receivedMessage into unicode characters")

string = ""

for n in receivedMessage:

# Decode into standard unicode set

if (n >= 32 and n <= 126):

string += chr(n)

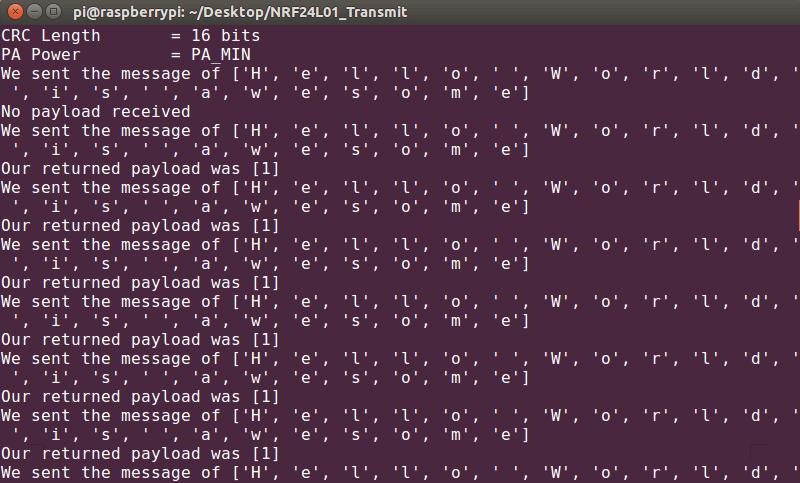
print(string)

radio.writeAckPayload(1, ackPL, len(ackPL))

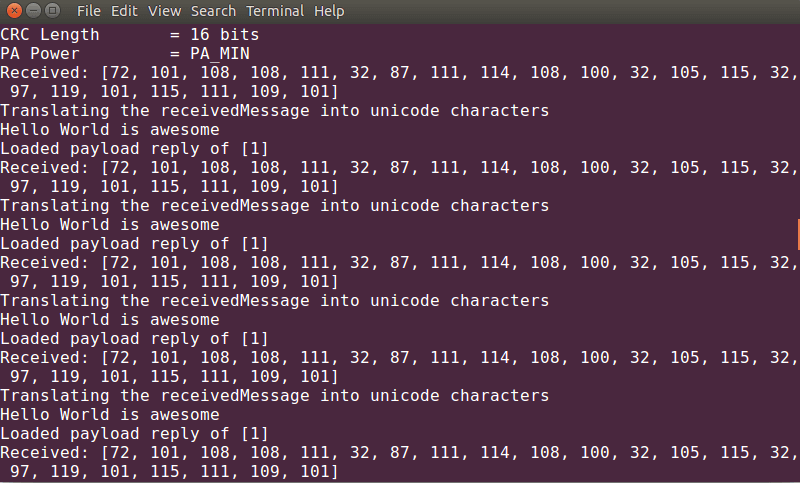
print("Loaded payload reply of {}".format(ackPL))

**27.**Finally, run the script by clicking on **Run -> Run Module**in the menu bar or by pressing **F5**on your PC keyboard.

**TransmitPi.py** output

[](http://invent.module143.com/wp-content/uploads/2016/07/TxOutput-1.png)

**ReceivePi.py** output

[](http://invent.module143.com/wp-content/uploads/2016/07/RxOutput-1.png)

**28.**For a master/slave setup, follow steps 1 – 23 as above and run the following python scripts **Master.py**and **Slave.py**on your respective devices.

**Master.py**

import RPi.GPIO as GPIO

from lib\_nrf24 import NRF24

import time

import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xe7, 0xe7, 0xe7, 0xe7, 0xe7], [0xc2, 0xc2, 0xc2, 0xc2, 0xc2]]

radio = NRF24(GPIO, spidev.SpiDev())

radio.begin(0, 17)

radio.setPayloadSize(32)

radio.setChannel(0x60)

radio.setDataRate(NRF24.BR\_2MBPS)

radio.setPALevel(NRF24.PA\_MIN)

radio.setAutoAck(True)

radio.enableDynamicPayloads()

radio.enableAckPayload()

radio.openReadingPipe(1, pipes[0])

radio.openWritingPipe(pipes[1])

radio.printDetails()

# radio.startListening()

def receiveData():

print("Ready to receive data.")

radio.startListening()

while not radio.available(0):

time.sleep(1 / 100)

receivedMessage = []

radio.read(receivedMessage, radio.getDynamicPayloadSize())

print("Translating receivedMessage into unicode characters...")

string = ""

for n in receivedMessage:

# Decode into standard unicode set

if (n >= 32 and n <= 126):

string += chr(n)

print("Our slave sent us: {}:".format(string))

radio.stopListening()

while(1):

command = "GET\_TEMP"

message = list(command)

# message = list("Hello World")

radio.write(message)

print("We sent the message of {}".format(message))

# Check if it returned ackPL

if radio.isAckPayloadAvailable():

returnedPL = []

radio.read(returnedPL, radio.getDynamicPayloadSize())

print("Our returned payload was {}".format(returnedPL))

receiveData()

else:

print("No payload received")

time.sleep(1)

**Slave.py**

import RPi.GPIO as GPIO

from lib\_nrf24 import NRF24

import time

import spidev

GPIO.setmode(GPIO.BCM)

pipes = [[0xe7, 0xe7, 0xe7, 0xe7, 0xe7], [0xc2, 0xc2, 0xc2, 0xc2, 0xc2]]

radio = NRF24(GPIO, spidev.SpiDev())

radio.begin(0, 17)

radio.setPayloadSize(32)

radio.setChannel(0x60)

radio.setDataRate(NRF24.BR\_2MBPS)

radio.setPALevel(NRF24.PA\_MIN)

radio.setAutoAck(True)

radio.enableDynamicPayloads()

radio.enableAckPayload()

radio.openWritingPipe(pipes[0])

radio.openReadingPipe(1, pipes[1])

radio.printDetails()

radio.startListening()

def getTemp():

temp = 25

return str(temp)

def sendData(ID, value):

radio.stopListening()

time.sleep(0.25)

message = list(ID) + list(value)

print("About to send message.")

radio.write(message)

print("Sent the data")

radio.startListening()

while(1):

ackPL = [1]

radio.writeAckPayload(1, ackPL, len(ackPL))

while not radio.available(0):

time.sleep(1 / 100)

receivedMessage = []

radio.read(receivedMessage, radio.getDynamicPayloadSize())

print("Received: {}".format(receivedMessage))

print("Translating the receivedMessage into unicode characters")

string = ""

for n in receivedMessage:

# Decode into standard unicode set

if (n >= 32 and n <= 126):

string += chr(n)

print(string)

# We want tp react to the command from the master.

command = string

if command == "GET\_TEMP":

print("We should get the temperature!")

tempID = "temp\_"

temp = getTemp()

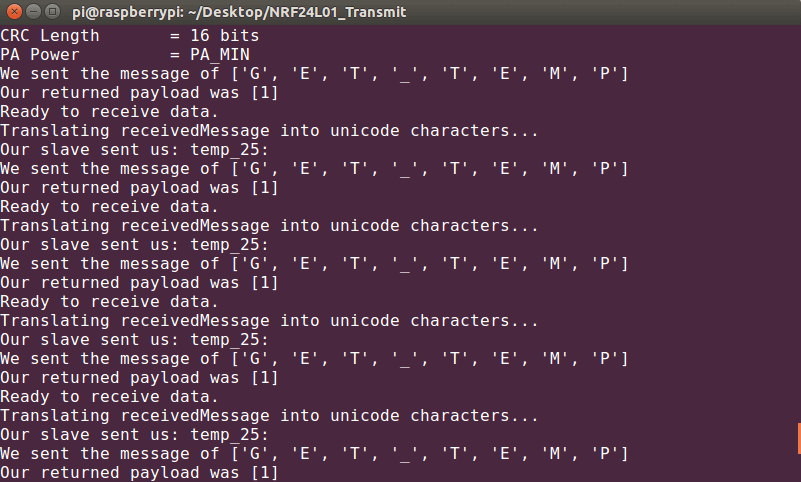
sendData(tempID, temp)

command = ""

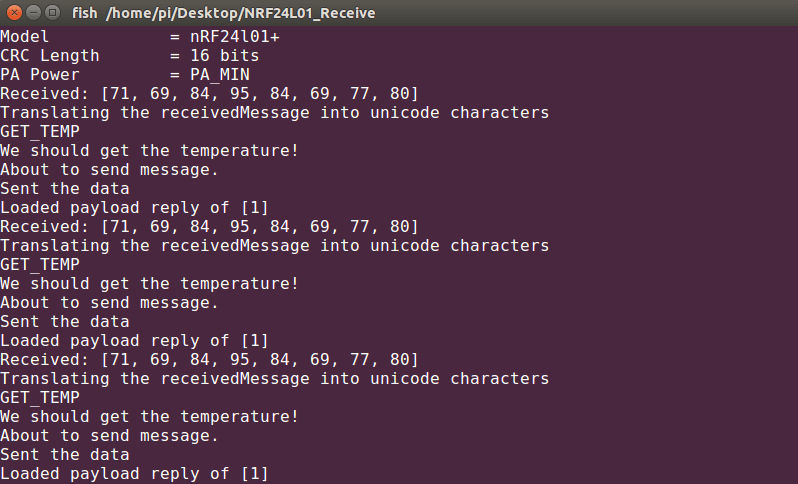
radio.writeAckPayload(1, ackPL, len(ackPL))

print("Loaded payload reply of {}".format(ackPL))

**Master.py**output

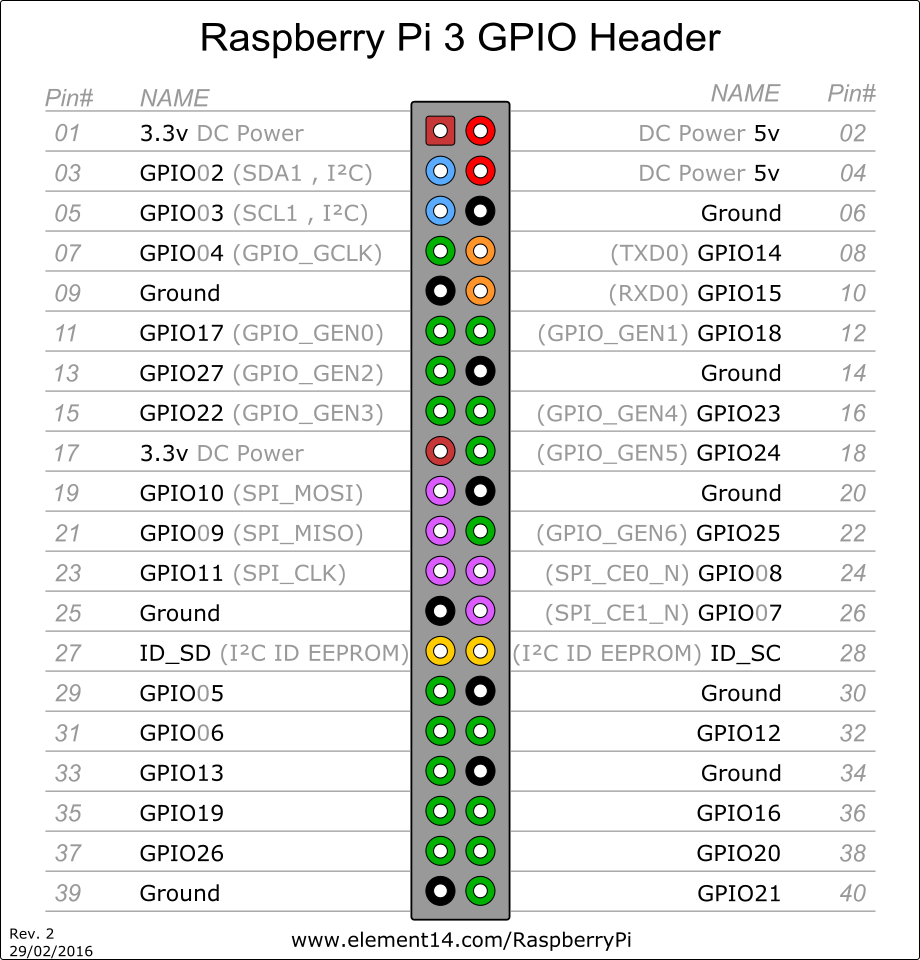
[](http://invent.module143.com/wp-content/uploads/2016/07/MasterOutput-1.png)

**Slave.py**output

[](http://invent.module143.com/wp-content/uploads/2016/07/SlaveOutput-1.png)

**Hardware Connections ->**

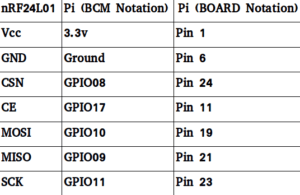
**1.**Raspberry Pi 3 GPIO Header.

[](http://invent.module143.com/wp-content/uploads/2016/06/pi3_gpio-1.png)

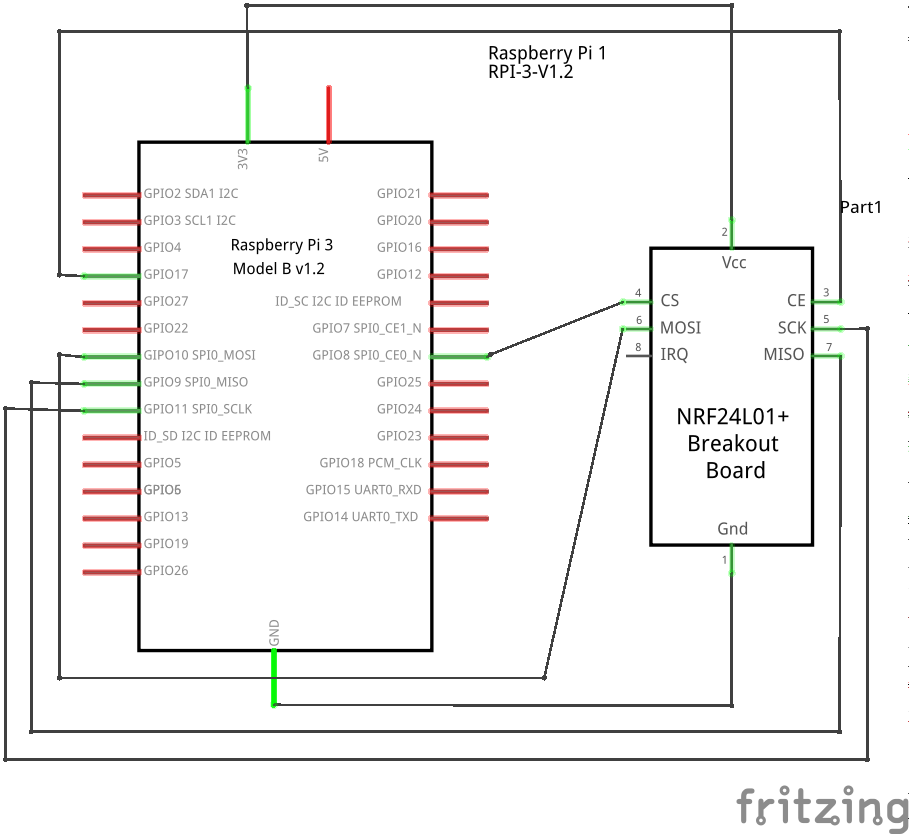
**2.**nRF24L01 pinout

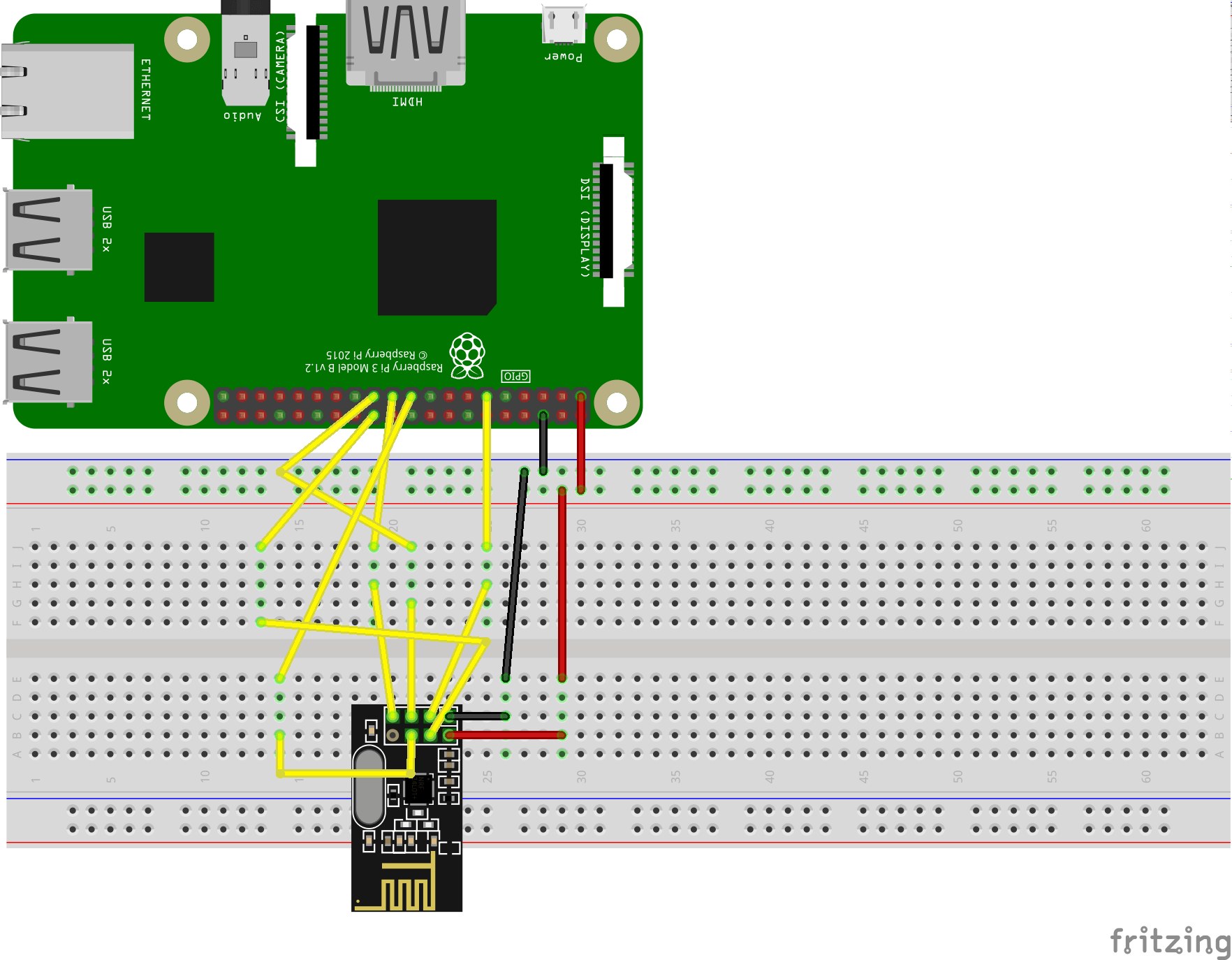


**3.**nRF24L01 to Pi connections.



**4.**Wire up both the devices to the RF modules.

[](http://invent.module143.com/wp-content/uploads/2016/07/RFToPi_schem-1.png)

[](http://invent.module143.com/wp-content/uploads/2016/07/RFToPi_bb-1.png)