Setting up Rasberry Pi

1)On the very first bootup of the Raspberry Pi device you will be prompted to install the OS. Select the option “Raspbian”.(Note: Make sure the memory card containing Rasbian is inserted into the PI kit)

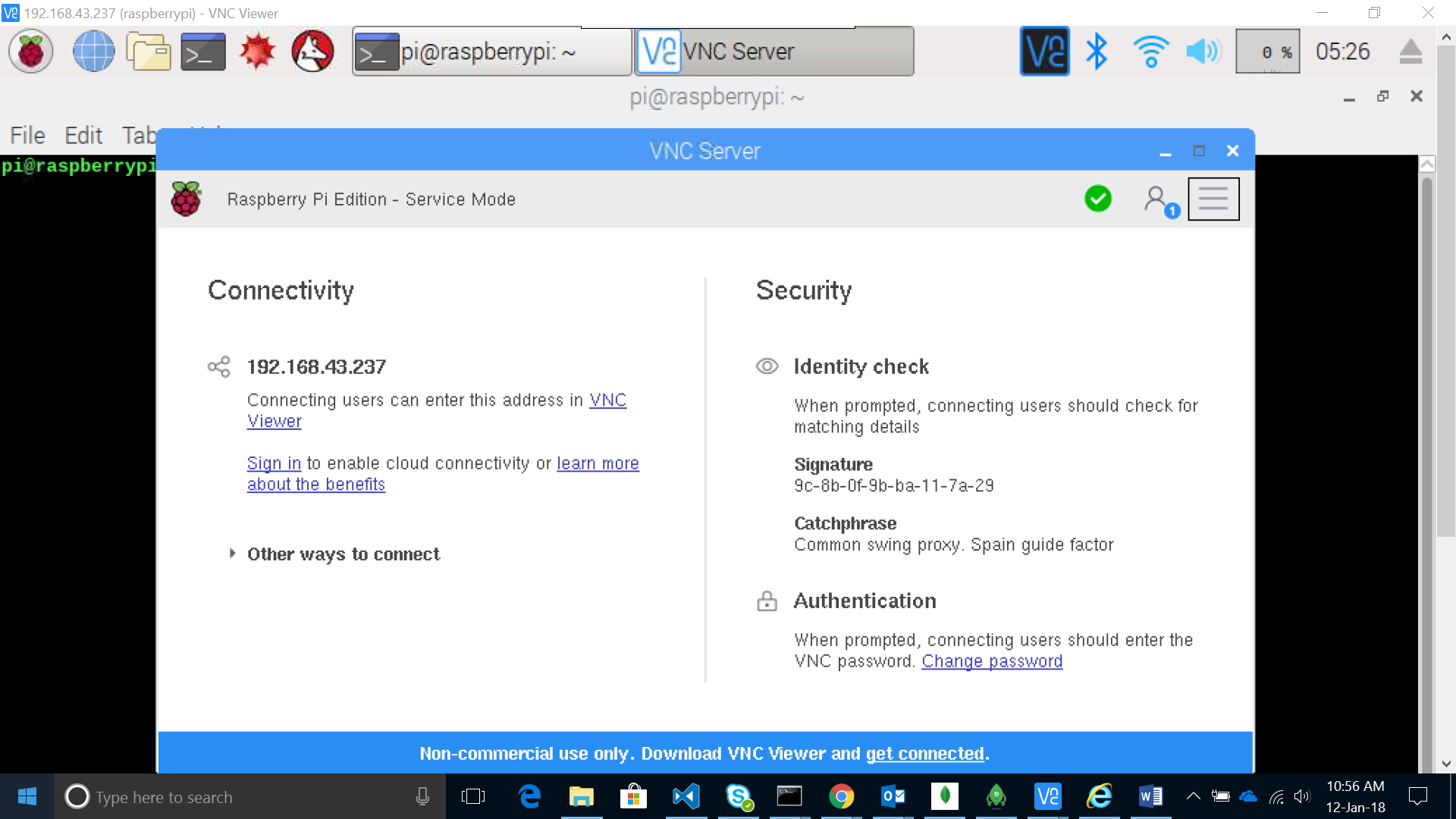
2)Install VNC server on the Raspberry Pi device. Open terminal and then execute the following commands sequentially:

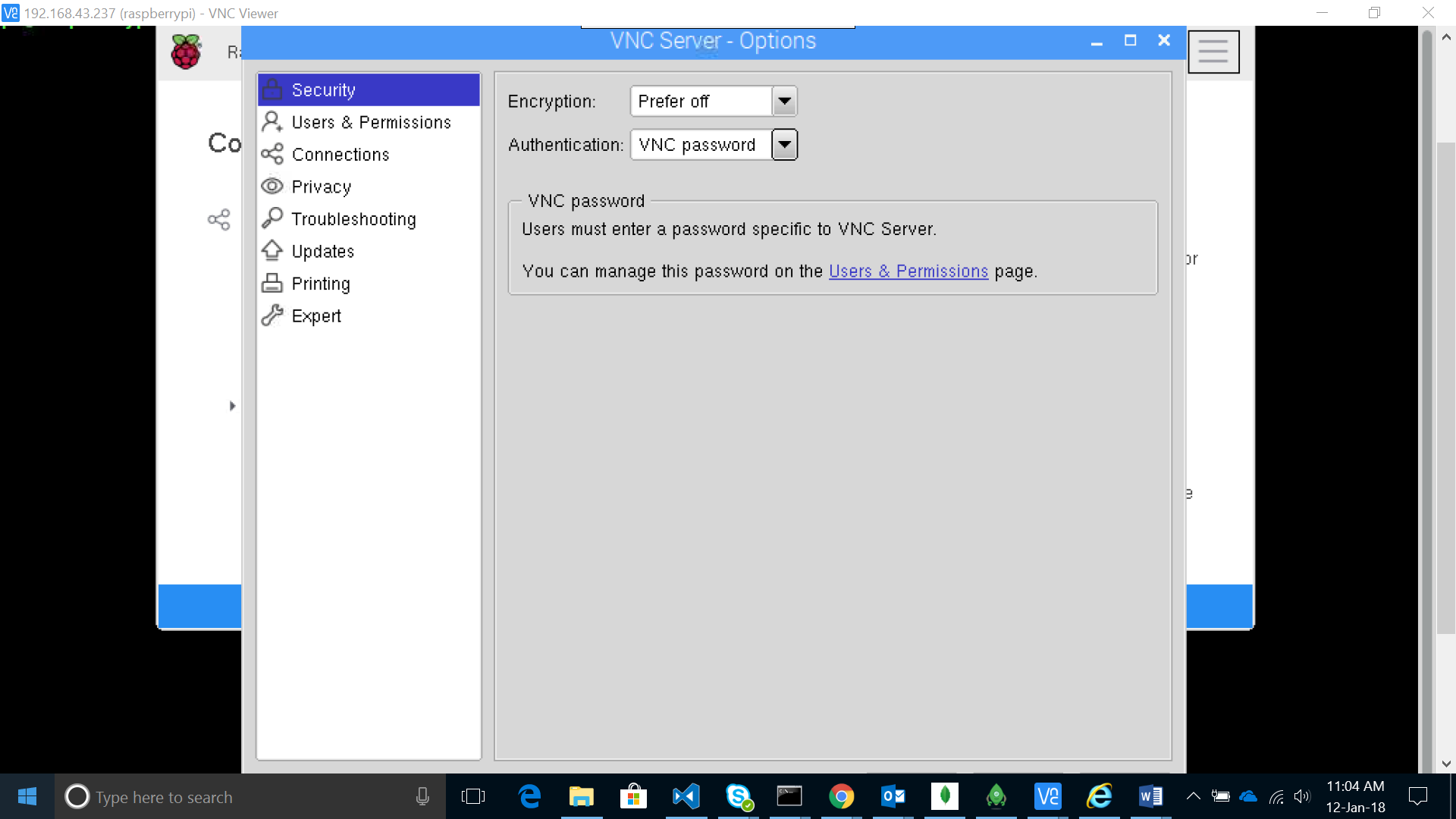
* sudo apt-get update
* sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer

To enable VNC Server type the following command in terminal:

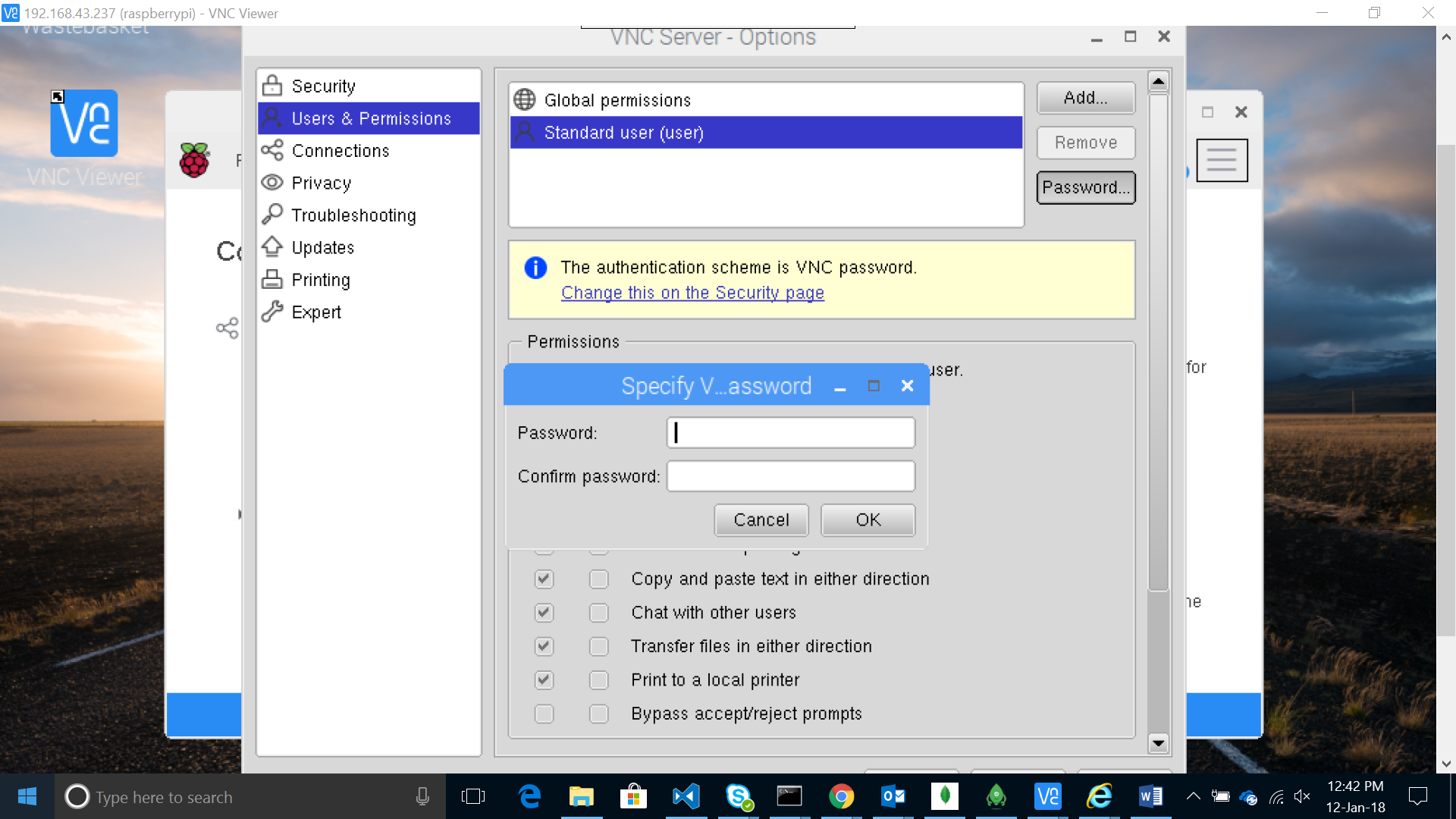
* sudo raspi-config

This will open an interactive menu. Navigate to Interfacing Options, scroll down and select VNC > Yes.

3) Set a password for the VNC server running on the Raspberry Pi. To do this, open VNC server and then click on the menu option() on the top right corner. Select Options->Authentication(VNC Password).

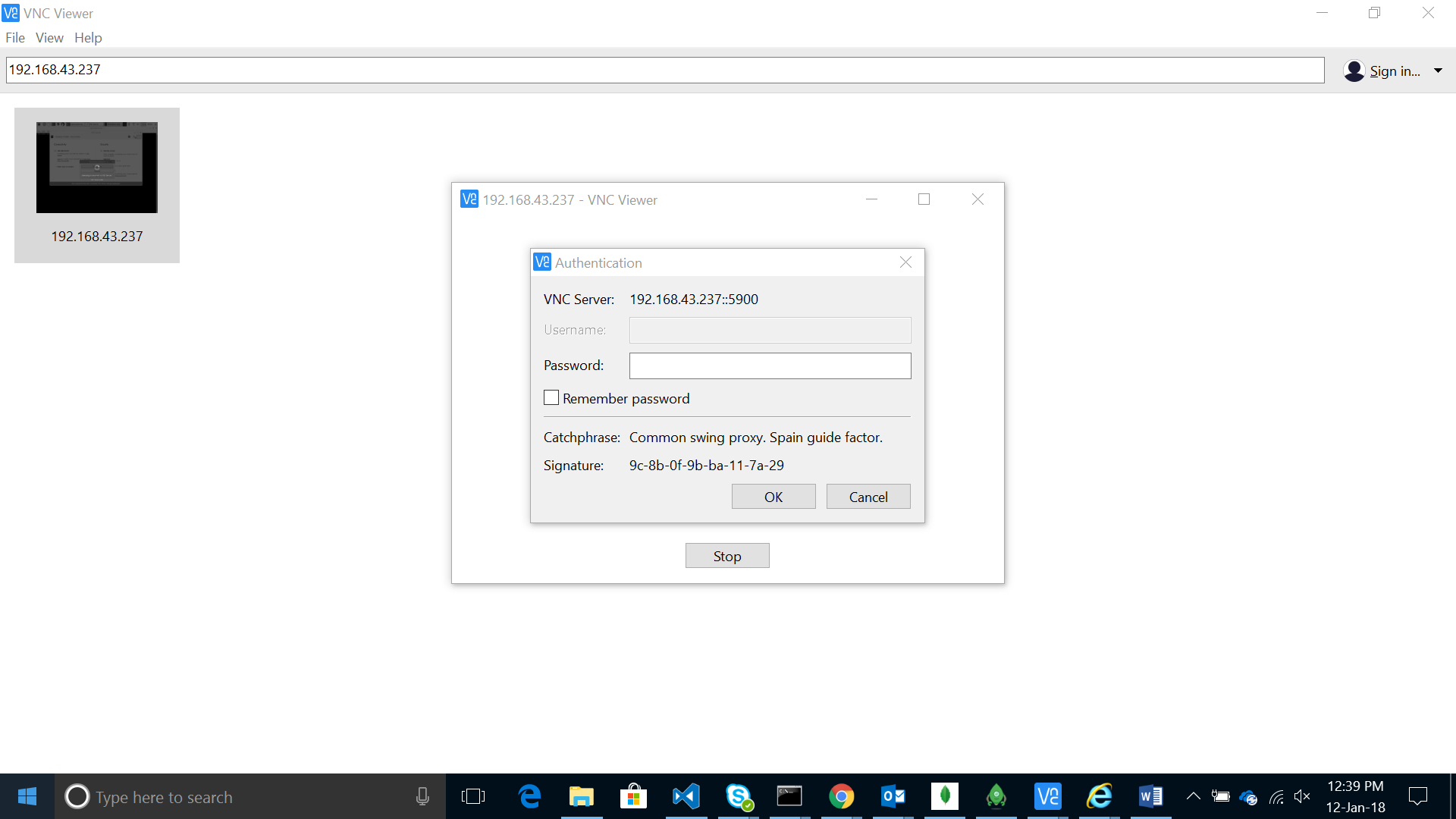


Click on the Users and Permissions tab and then click the Password button. Fill up the desired password in the pop up. This password will be used to connect to the Raspberry PI from the Laptop/Monitor.



4)Install VNC Viewer on your system. Open the viewer and enter the IP address of the Raspberry Pi and you will be prompted for password.(Note: The Raspberry Pi and Laptop should be connected to the same Wi-Fi network.)

Link for download: <https://www.realvnc.com/en/connect/download/viewer/>



5)Now you will be able to view as well as use the keyboard and mouse of your system to control the Raspberry Pi device.

6)On connecting the Raspberry Pi device directly to the laptop/monitor, the VNC viewer can be viewed only on a small window. To solve this issue, follow the steps in the below link.

<https://www.raspberrypi.org/forums/viewtopic.php?t=161813>

Testing the Ultrasound Sensor

1. Connect the sensor to the Raspberry Pi device as per the guide in the following link.

<https://www.raspberrypi-spy.co.uk/2012/12/ultrasonic-distance-measurement-using-python-part-1/#prettyPhoto>

1. The code in the ultrasonic\_1.py file measures distance only once. Replace the entire code with the code below to measure distance twice and compute speed.

#!/usr/bin/python

#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

#|R|a|s|p|b|e|r|r|y|P|i|-|S|p|y|.|c|o|.|u|k|

#+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

#

# ultrasonic\_1.py

# Measure distance using an ultrasonic module

#

# Ultrasonic related posts:

# http://www.raspberrypi-spy.co.uk/tag/ultrasonic/

#

# Author : Matt Hawkins

# Date : 16/10/2016

# -----------------------

# Import required Python libraries

from \_\_future\_\_ import print\_function

import time

import RPi.GPIO as GPIO

# Use BCM GPIO references

# instead of physical pin numbers

GPIO.setmode(GPIO.BCM)

# Define GPIO to use on Pi

GPIO\_TRIGGER = 23

GPIO\_ECHO = 24

# Speed of sound in cm/s at temperature

temperature = 20

speedSound = 33100 + (0.6\*temperature)

print("Ultrasonic Measurement")

print("Speed of sound is",speedSound/100,"m/s at ",temperature,"deg")

duration=1

def measure():

# Set pins as output and input

GPIO.setup(GPIO\_TRIGGER,GPIO.OUT) # Trigger

GPIO.setup(GPIO\_ECHO,GPIO.IN) # Echo

# Set trigger to False (Low)

GPIO.output(GPIO\_TRIGGER, False)

# Allow module to settle

time.sleep(0.5)

# Send 10us pulse to trigger

GPIO.output(GPIO\_TRIGGER, True)

# Wait 10us

time.sleep(0.00001)

GPIO.output(GPIO\_TRIGGER, False)

start = time.time()

while GPIO.input(GPIO\_ECHO)==0:

start = time.time()

while GPIO.input(GPIO\_ECHO)==1:

stop = time.time()

# Calculate pulse length

elapsed = stop-start

# Distance pulse travelled in that time is time

# multiplied by the speed of sound (cm/s)

distance = elapsed \* speedSound

# That was the distance there and back so halve the value

distance = distance / 2

#print("Distance : {0:5.1f}".format(distance))

# Reset GPIO settings

return distance

print("First Measure")

x=measure()

time.sleep(duration)

print("Second Measure")

y=measure()

print("Distance 1:",x)

print("Distance 2:",y)

speed=abs(x-y)/duration

print("Speed of moving object:",speed)

# Reset GPIO settings

GPIO.cleanup()

1. Run the code using the command “python ultrasonic\_1.py” and take the required measurements.

Testing Buzzer Board

1)Make the following connections:

* Connect pin 1(3V) to VCC of sensor.
* Connect pin 6(GND) to GND of sensor.
* Connect pin 11(GPIO 17) to I/P of sensor.

2) Copy the python code for Active sensor from the following link

<https://www.sunfounder.com/learn/sensor-kit-v2-0-for-raspberry-pi-b-plus/lesson-10-buzzer-module-sensor-kit-v2-0-for-b-plus.html>

3)Now execute the python code and the Buzzer should sound.