fMonitor for Smoke  
Raspberry Pi

# Preliminary Discussion

A common function of network management is to monitor the environment in the network equipment rooms. Things to watch are temperature, humidity, smoke, entry into the room, and so on. This can be done by purchasing devices that incorporate various sensors and notification methods. In this lab we will see how these devices do what they do as well as learn how to create such a device.

**Setup**

#### The following equipment is needed for this lab:

Video camera that can stream video to a computer for display by a projector.

The following parts are needed for this lab:

Raspberry Pi Board

Raspberry Pi Case

Raspberry Pi Power Supply

MQ-2 Smoke Sensor

MCP3002 Analog to Digital Converter

Computer Monitor with a HDMI Port or a Converter

Solderless Breadboard

Eight Jumper Wires

An Internet Connection

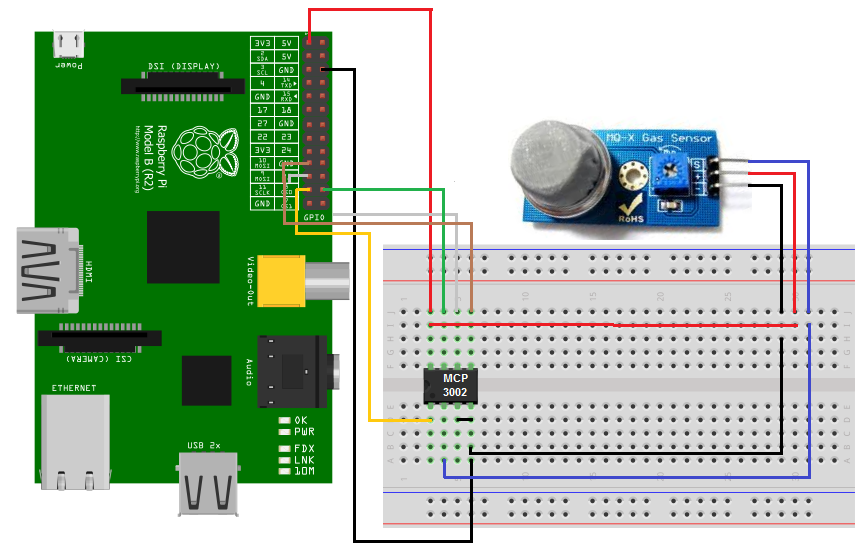
This program is required:

Any method that allows interaction with the Raspberry Pi will work. In this example the following are used:

TeraTerm or some other terminal program attached to the same network as the Raspberry Pi in order to access the operating system’s command line.

**Assemble the Parts**

The parts are laid out in this manner.



The active components used for this lab perform the following functions:

The Raspberry Pi board reads the input from the sensor and then outputs it. In this example we see it as a smoke level reading at the command line of the operating system and as an email message sent to the email address sensor@chipps.com.

The datasheet for the MQ-2 smoke sensor says it is a gas sensor composed of a micro AL2O3 ceramic tube, Tin Dioxide SnO2 sensitive layer, measuring electrode, and heater that are fixed into a crust made by plastic and stainless steel. The board based version used here has a potentiometer that is used to adjust its sensitivity to smoke.

The MCP3002 analog to digital converter chip is used to convert the MQ-2 analog signal to a digital signal as the Raspberry Pi only reads digital signals.

Now that we have all of the pieces of the lab, let’s assemble them. You can see how I do this by watching the camera view projected on the screen.

Here is where the parts go:

MQ-2 GND pin to breadboard pin J26

MQ-2 VCC pin to bread board pin J27

MQ-2 SIG pin to breadboard pin J28

Breadboard pin I26 to breadboard pin B6 which is the MCP3002 pin 4 VSS

Breadboard pin H27 to breadboard pin G3 which is the MCP3002 pin 8 VDD or VREF

Breadboard pin G28 to breadboard pin A4 which is the MCP3002 pin 2 CH0

Breadboard pin D3 which is the MCP3002 pin 1 CD to Raspberry Pi pin left side second up from yellow\

Breadboard pin A6 to the Raspberry Pi pin GND pin left side third down from top

Breadboard pin J3 which is the MCP3002 pin 8 to the Raspberry Pi pin 3.3V right side top

Breadboard pin J4 which is the MCP3002 pin 7 to the Raspberry Pi pin left side second from bottom

Breadboard pin J5 which is the MCP3002 pin 6 to the Raspberry Pi pin left side third from bottom

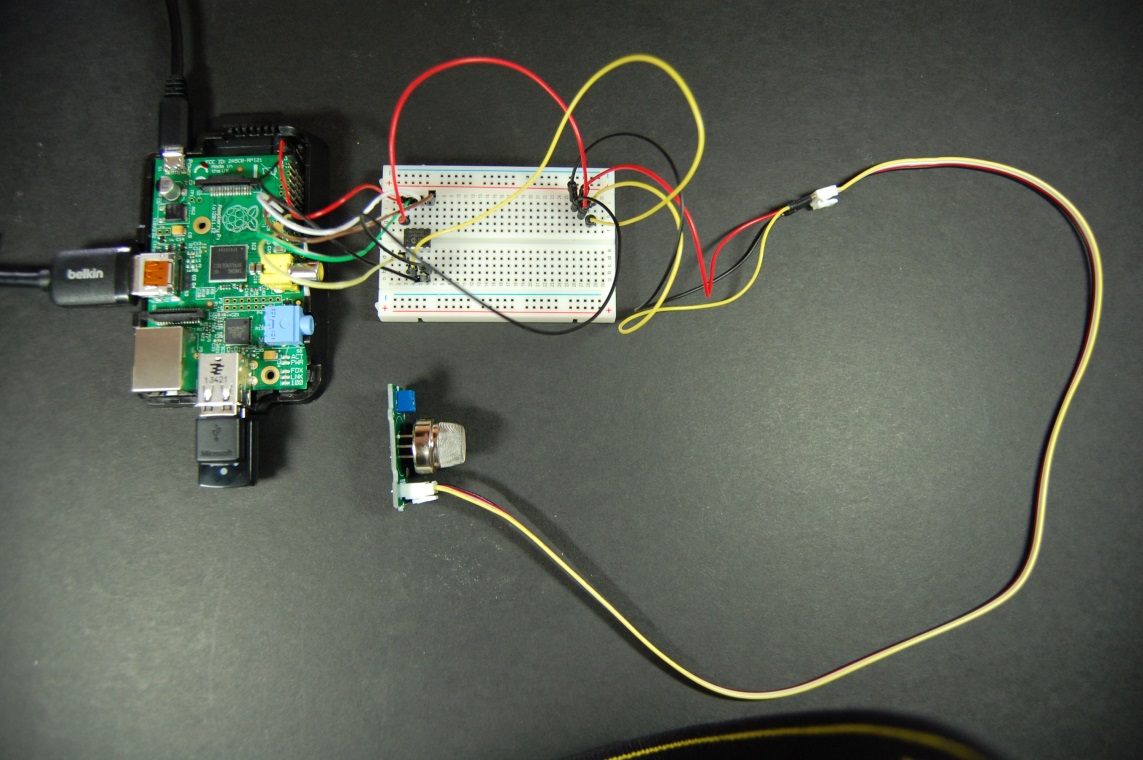
Breadboard pin J6 which is the MCP3002 pin 5 to the Raspberry Pi pin left side fourth from bottom

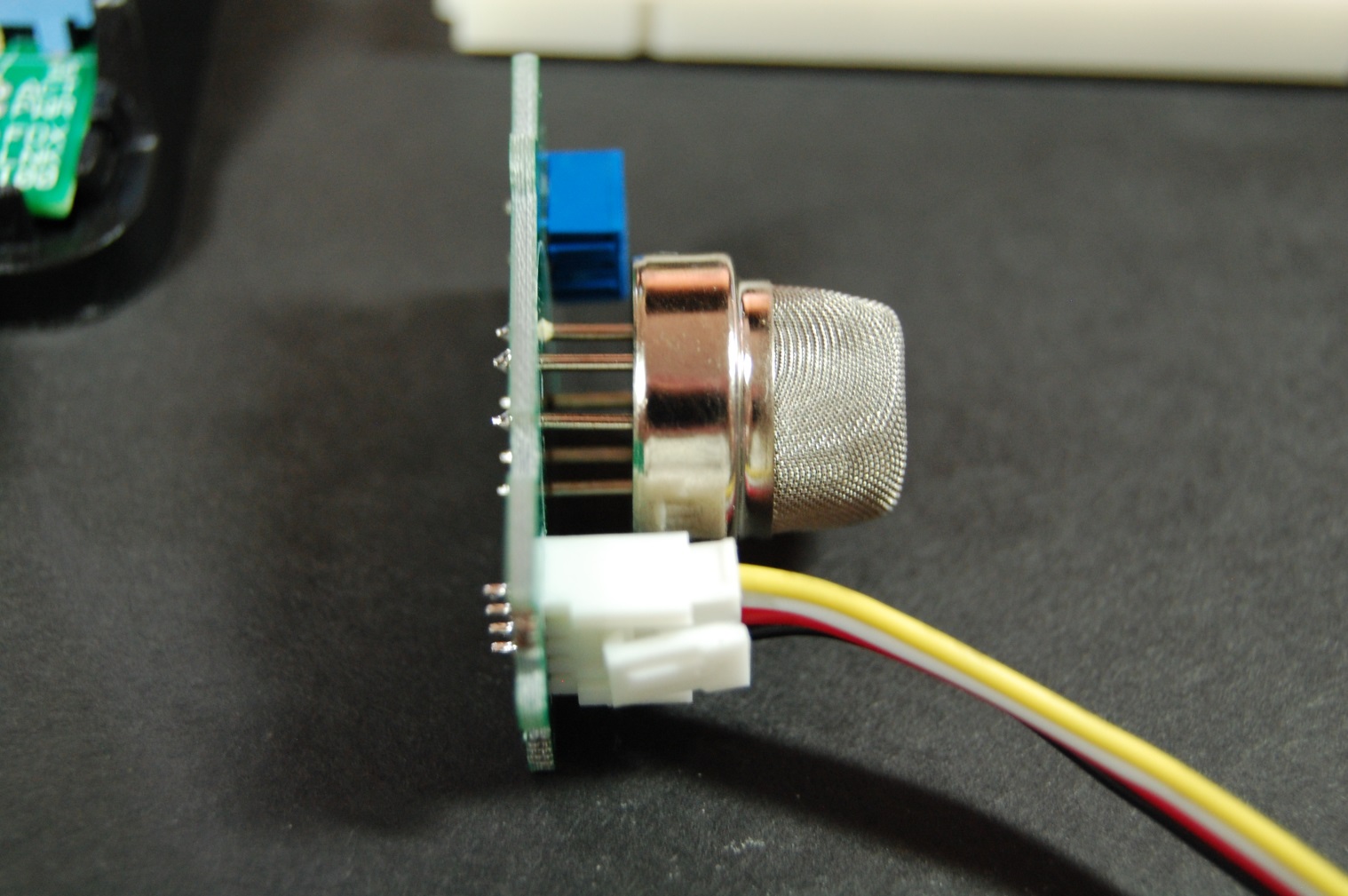
HDMI connector to monitor

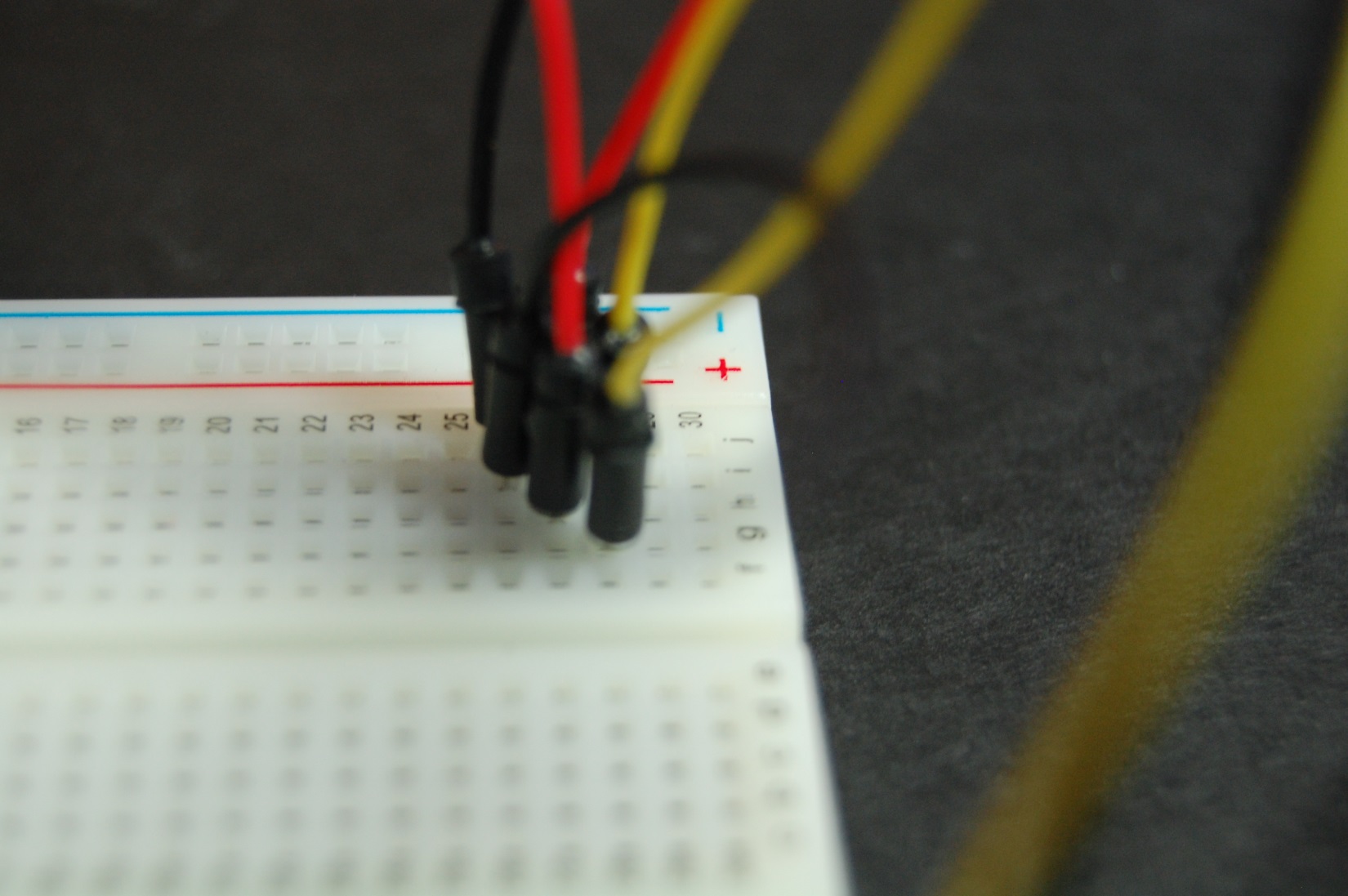
Wired or wireless NIC to a switch to which the Internet is connected

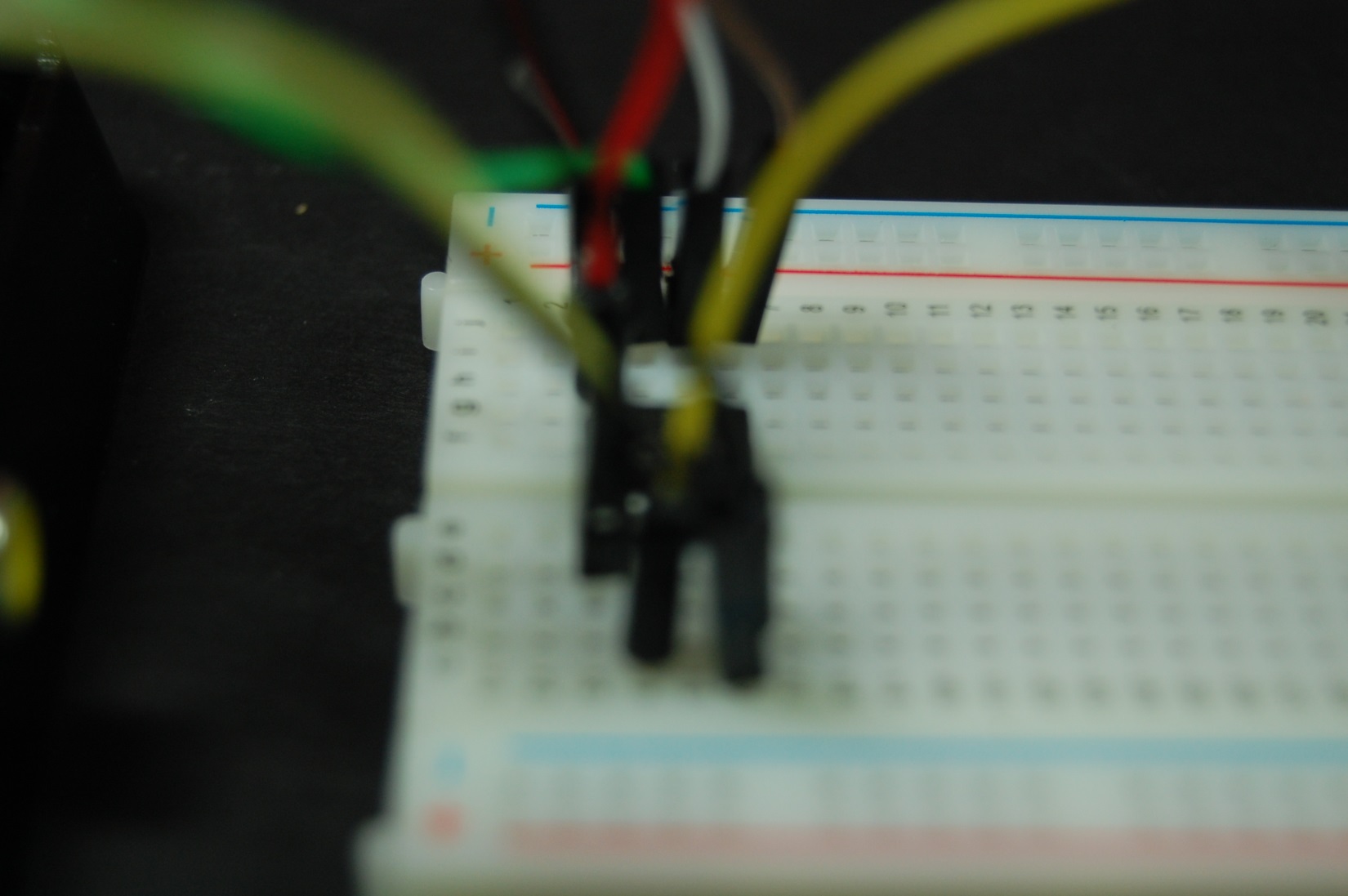
Power supply to Micro USB port

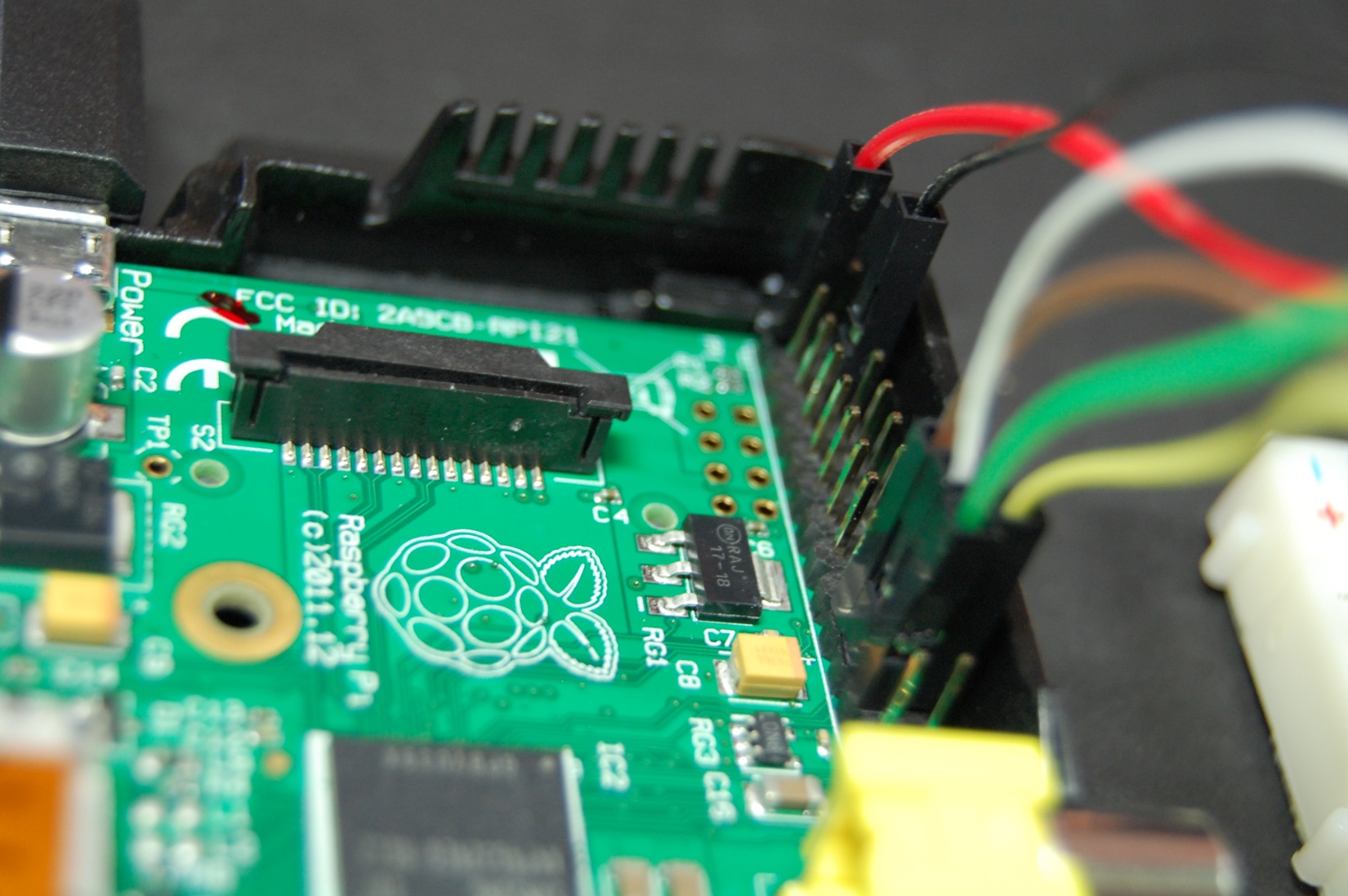
The finished system looks like this:











Let’s apply power to the system by plugging the USB cable into the port on the Raspberry Pi and the other end to a power outlet.

**Load the Code**

So that the parts can talk to each other and to us, we need to run a Python script. Here it is:

import time

import botbook\_mcp3002 as mcp

import smtplib

from email.MIMEText import MIMEText

email\_to = 'chipps@chipps.com'

email\_from = 'chipps@chipps.com'

import time

import botbook\_mcp3002 as mcp

import smtplib

from email.MIMEText import MIMEText

email\_to = 'chipps@chipps.com'

email\_from = 'chipps@chipps.com'

server = 'smtp.chipps.com:2525'

username = 'sensor@chipps.com'

password = '!mail4706'

gracePeriod = 1 \* 10

def sendEmail(subject, msg):

msg = MIMEText(msg)

msg['Subject'] = subject

msg['To'] = email\_to

msg['From'] = email\_from

smtp = smtplib.SMTP(server)

smtp.starttls()

smtp.login(username,password)

smtp.sendmail(email\_from, email\_to, msg.as\_string())

smtp.close()

def main():

while True:

smokeLevel = mcp.readAnalog()

print("Current smoke level is %i " % smokeLevel)

if smokeLevel > 120:

print("Smoke detected")

sendEmail("Smoke","Smoke level was %i" % smokeLevel)

time.sleep(gracePeriod)

time.sleep(0.5)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Using TeraTerm login to the raspberry Pi as

root

raspberry

To load the code into the Raspberry Pi download the code file from the same website this document was accessed from, open it in a text editor, copy it, run the Nano text editor on the Raspberry Pi, and then paste it to the Raspberry Pi Nano text editor through TeraTerm. Name the file:

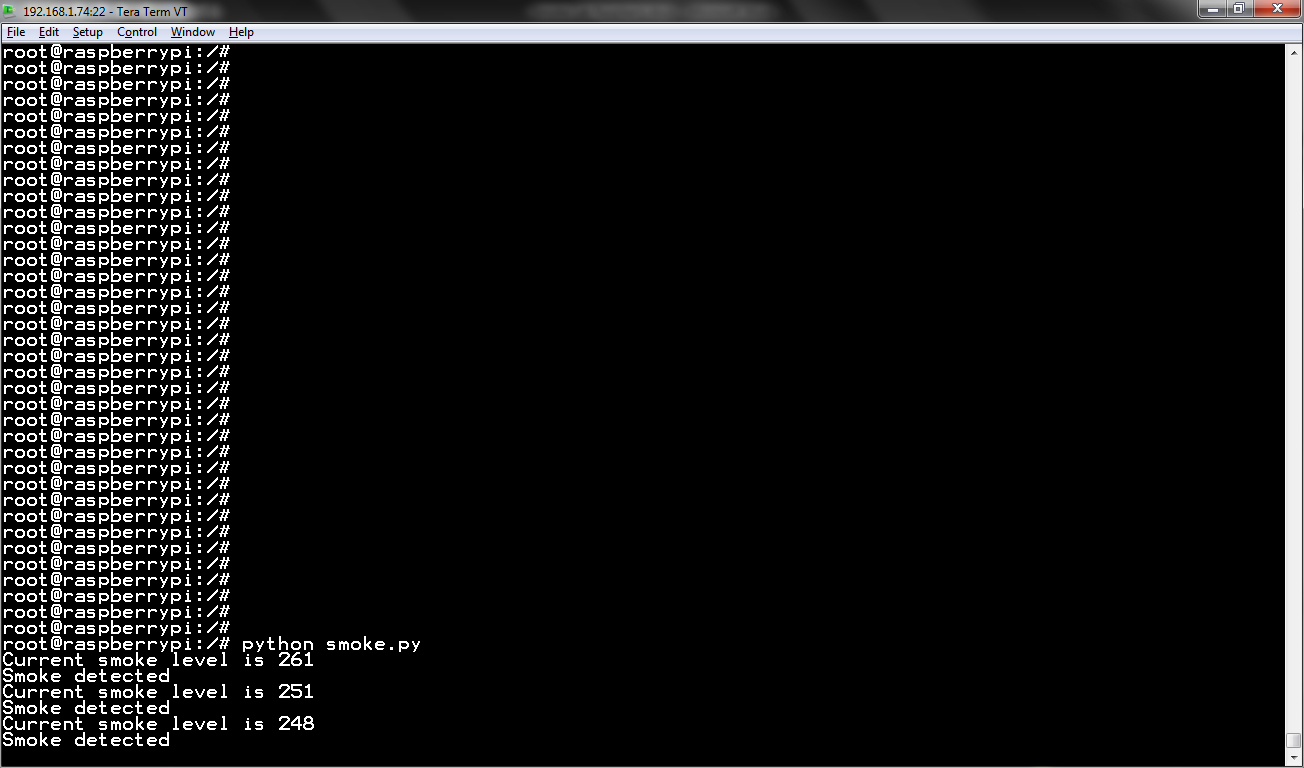
smoke.py

Run the code by entering at the command prompt:

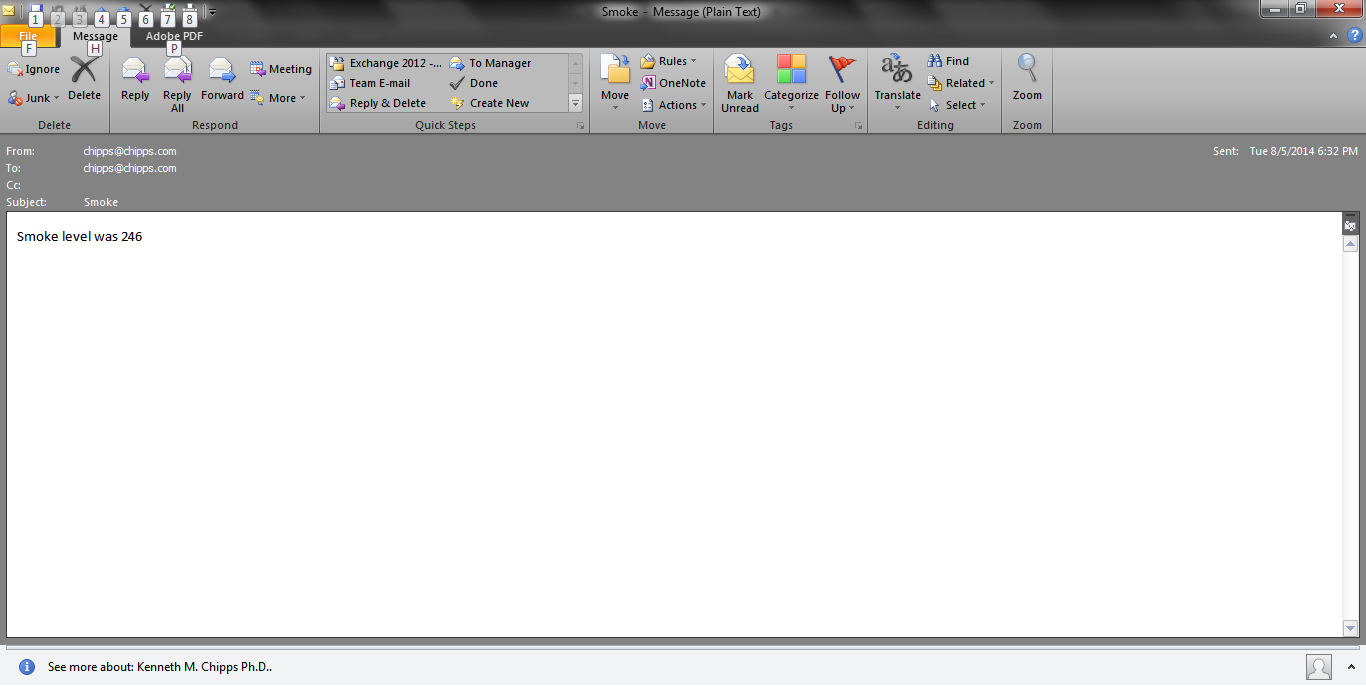
python smoke.py

**Watch the Result**

With the code loaded and the potentiometer set to detect low levels of smoke, output similar to this should appear at the command prompt of the Raspberry Pi.



Access the chipps.com email system to retrieve email messages from the sensor@chipps.com account. A message such as this should appear.



**Source**

The basic procedure used here is from the book Make: Sensors and the article MQ-2 Smoke Sensor Circuit Using Raspberry Pi from http://www.learningaboutelectronics.com/Articles/MQ-2-smoke-sensor-circuit-with-raspberry-pi.php.