

Using GSM Module with Raspberry-Pi

About GSM module:

GSM/GPRS module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. Global Packet Radio Service (GPRS) is an extension of GSM that enables higher data transmission rate. GSM/GPRS module consists of a GSM/GPRS modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. GSM/GPRS MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM and GPRS network. It requires aSIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification.

The MODEM needs **AT commands**, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM and GPRS cellular network.

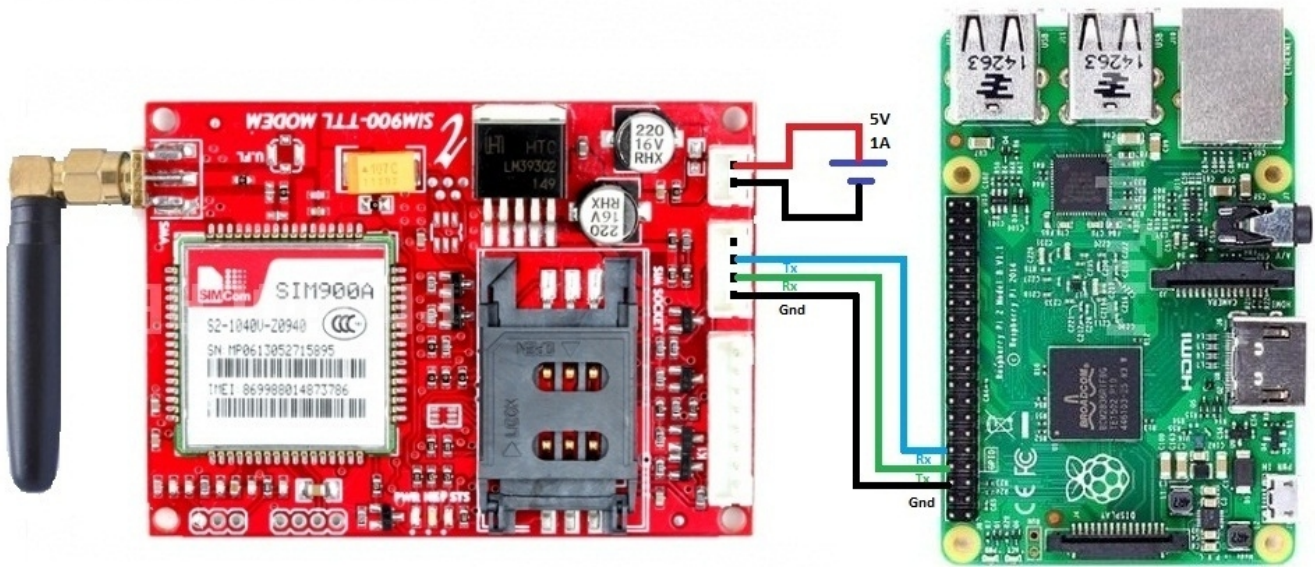
For more about AT commands, refer below links:

<http://www.engineersgarage.com/tutorials/at-commands>

<http://www.developershome.com/sms/atCommandsIntro.asp>

https://www.sparkfun.com/datasheets/Cellular%20Modules/AT_Commands_Reference_Guide_r0.pdf

Pin Connection:



The Network LED indicates the various states of the GSM module i.e. **POWER ON**, **NETWORK REGISTRATION** and **GPRS CONNECTIVITY**. When the modem is powered up, this NETWORK LED will blink every second. After the Modem registers in the network (it takes 10-60 seconds), this LED will blink in step of 3 seconds at slow rate. At this stage we can start using the modem for our application. This shows that the modem is registered with the network.

Connect the pins of the Sensor to raspberry-pi as follows:

Note: You can use the 12V adapter if there is adapter port available (Hence the Vcc pin may not be available in the Module)

Pin of Sensor → Pin of Raspberry-Pi

GND → Pin 6

TX → RX (Pin 10)

RX → TX (Pin 8)

Code To receive the SMS:

```
import serial
import RPi.GPIO as GPIO
import os, time

# Find a suitable character in a text or string and get its position
def find(str, ch):
    for i, ltr in enumerate(str):
        if ltr == ch:
            yield i

GPIO.setmode(GPIO.BOARD)

# Enable Serial Communication
port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)

# Transmitting AT Commands to the Modem
# '\r\n' indicates the Enter key

port.write('AT'+'\r\n')
port.write("\x0D\x0A")
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('ATE0'+'\r\n')                                # Disable the Echo
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('AT+CMGF=1'+'\r\n')                            # Select Message format as Text mode
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('AT+CNMI=2,1,0,0,0'+'\r\n')                    # New SMS Message Indications
rcv = port.read(10)
print rcv
time.sleep(1)

ck=1
while ck==1:
    rcv = port.read(10)
    print rcv
    fd=rcv
    if len(rcv)>3:                                          # check if any data received
        ck=12
        for i in range(5):
            rcv = port.read(10)
            print rcv
            print rcv
            fd=fd+rcv                                    # Extract the complete data

# Extract the message number shown in between the characters "," and '\r'

p=list(find(fd, ","))
q=list(find(fd, '\r'))
```

```

        MsgNo=fd[p[0]+1:q[1]]

# Read the message corresponds to the message number
rd=port.write('AT+CMGR='+MsgNo+'\r\n')
msg=''
for j in range(10):
    rcv = port.read(20)
    msg=msg+rcv
    print msg
time.sleep(0.1)

```

Note: Use `/dev/ttyAMA0` instead of `/dev/ttyS0` (Depending upon your Raspberry Pi version and Configurations), Try to use serial data transferring via UART, so that you will be comfortable using GSM module.

Sample Output:

```

+CMTI: "
SM",4

+CMGR: "REC UNREAD","+917760049480","", "16/12/17,16:44:49+22"
Hi nike

OK

pi@raspberrypi:~ $ █

```

Code To send the SMS:

```

import serial
import RPi.GPIO as GPIO
import os, time

GPIO.setmode(GPIO.BOARD)

# Enable Serial Communication
port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)

# Transmitting AT Commands to the Modem
# '\r\n' indicates the Enter key

port.write('AT'+'\r\n')
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('ATE0'+'\r\n')      # Disable the Echo
rcv = port.read(10)
print rcv

```

```

time.sleep(1)

port.write('AT+CMGF=1'+'\r\n') # Select Message format as Text mode
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('AT+CNMI=2,1,0,0,0'+'\r\n') # New SMS Message Indications
rcv = port.read(10)
print rcv
time.sleep(1)

# Sending a message to a particular Number

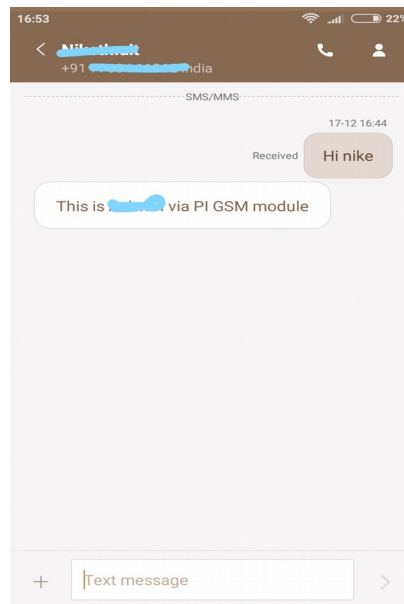
port.write('AT+CMGS="+91<ENTER your NO>"'+'\r\n')
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('This is <Your Name> via PI GSM module'+'\r\n') # Message
rcv = port.read(10)
print rcv

port.write("\x1A") # Enable to send SMS
for i in range(10):
    rcv = port.read(10)
    print rcv

```

Sample Output (Screen Shot From Phone):



Try This Out:

Using the GSM module and Smoke sensor (or Flame Sensor) combine the code such that, when smoke is detected a message is sent to the user (concerned person) about the incident.

Code:

```
import serial
import RPi.GPIO as GPIO
import os, time

GPIO.setmode(GPIO.BOARD)

def send_msg():
    # Enable Serial Communication
    port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)

    # Transmitting AT Commands to the Modem
    # '\r\n' indicates the Enter key

    port.write('AT'+'\r\n')
    rcv = port.read(10)
    print rcv
    time.sleep(1)

    port.write('ATE0'+'\r\n')      # Disable the Echo
    rcv = port.read(10)
    print rcv
    time.sleep(1)

    port.write('AT+CMGF=1'+'\r\n') # Select Message format as Text mode
    rcv = port.read(10)
    print rcv
    time.sleep(1)

    port.write('AT+CNMI=2,1,0,0,0'+'\r\n') # New SMS Message Indications
    rcv = port.read(10)
    print rcv
    time.sleep(1)

# Sending a message to a particular Number

    port.write('AT+CMGS="+917760049480"'+'\r\n')
    rcv = port.read(10)
    print rcv
    time.sleep(1)

    port.write('Smoke Detected!!! Take appropriate action'+'\r\n') # Message
    rcv = port.read(10)
    print rcv

    port.write("\x1A") # Enable to send SMS
    for i in range(10):
        rcv = port.read(10)
        print rcv
    exit(0);

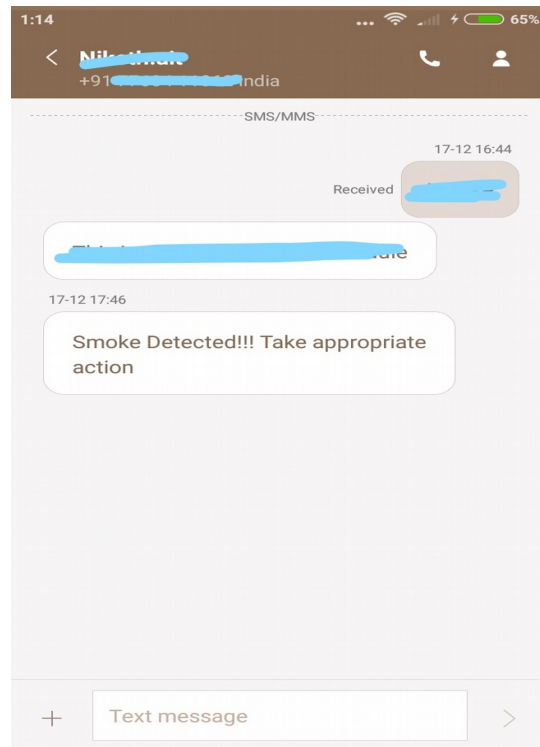
GPIO.setup(7,GPIO.IN)
while True:
    try:
        if (GPIO.input(7)==0):
            print("Smoke Detected! Move to safety")
```

```

        send_msg()
    else:
        print("No Smoke! you are in SAFE zoone")
        time.sleep(1)
except KeyboardInterrupt:
    exit()
GPIO.cleanup()

```

Sample Output (Screen Shot):



References:

- <http://www.rhydolabz.com/wiki/?p=10450>
- http://extremeelectronics.co.in/datasheets/gsm_shield_sim900a.pdf
- <http://electronicsforu.com/resources/what-is-gsm-module>