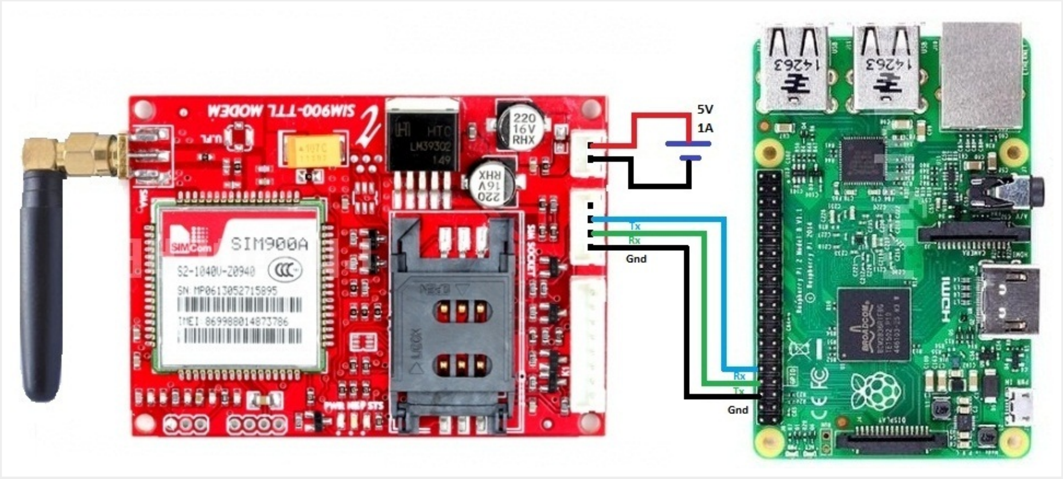
GSM/GPRS TTL UART Modem-(SIM900A, SIM900, SIM800) Interface With Raspberry Pi

Most of the people are familiarized with the dial up modem as it a common device for land line telephone network used in homes and offices. It is a wired network which has got its own limitations in the embedded area. GSM modem is introduced to rectify the main limitation of the dial up modem based on its acceptance of a sim card. It is almost equivalent to a mobile communication system as operates over a subscription to a mobile operator. From the mobile operator perspective, a GSM modem looks just like a mobile phone.Using the transmission and reception pins, a modem can receive and send the messages and it could be interfaced with the PC or to a microcontroller. This property makes the modem to exist in a relevant position on embedded applications.

Raspberry Pi is a high end embedded device, so an interface with a GSM modem will results in the invention of an intelligent system. As the Raspberry Pi finds its own applications in image, video and audio processing, the modem interface could develop a new path in the field of home automation, surveillance etc.

Here we gives the details for how to interface a GSM/GPRS TTL UART Modem-SIM900A with Raspberry Pi2 and to send and receive message through it.



Modules Needed

1. Raspberry Pi 2
2. GSM/GPRS TTL UART Modem-SIM900A
3. A 5V battery or power supply
4. Connecting wires

5. A mobile phone
6. An extra SIM card

Insert a sim card in to the GSM modem and make the suitable connections as shown in the figure. A suitable factor that should keep in mind is that, the sim used in the module should have enough balance to send the sms and it should be kept in a place having appropriate range for that particular network. The transmission and reception pins should be connected in a reverse order and the ground pins must be shorted. Power up the GSM module and wait for few seconds for sim initialization.

Now we need to send and receive messages through this modem with the help of a python script. A sample python code for sending a message through raspberry pi and gsm modem is given below:

Sample Code to send SMS via GSM modem

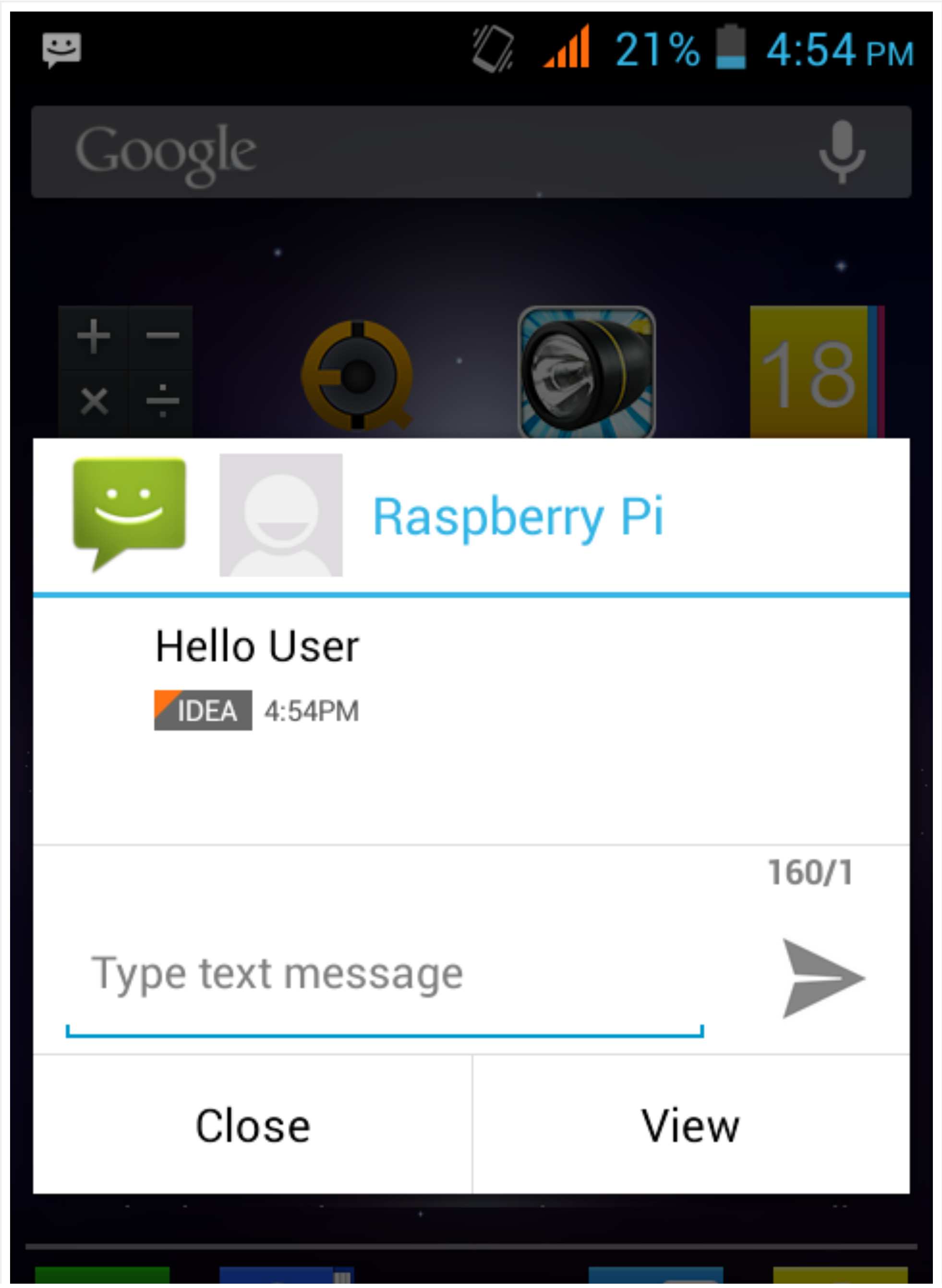
```python
import serial
import RPi.GPIO as GPIO
import os, time

GPIO.setmode(GPIO.BOARD)

# Enable Serial Communication
port = serial.Serial("/dev/ttyAMA0", 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 Indication
rcv = port.read(10)
print rcv
time.sleep(1)

# Sending a message to a particular Number

port.write('AT+CMGS="9495353464"'+'\r\n')
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('Hello User'+'\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
```

In the code, import the proper libraries and enable the serial communication. Modem control is done through AT commands, so suitable commands should be transmitted to the modem for each purpose and the modem will respond to these commands by transmitting suitable messages that should received and display it on the python shell.

Click here to know the details of SIM900A AT command set.

The suitable message will be received by the mobile number given in the code.



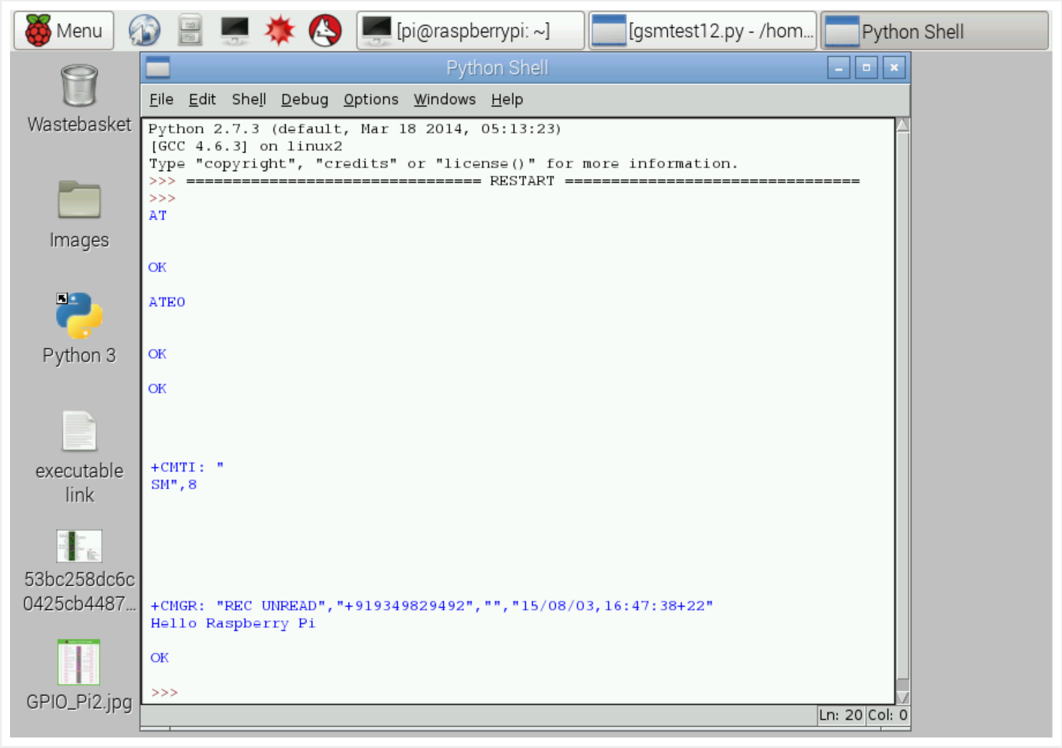
Hello User

Now we can move on to another sample code to receive the sms.

Sample Code to retrieve SMS received via GSM modem

```
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/ttyAMA0", 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
rcv = port.read(10)
print rcv
time.sleep(1)

port.write('AT+CNMI=2,1,0,0,0'+'\r\n')      # New SMS Message Indicat
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)&gt;3:                   # check if any data received
        ck=12
        for i in range(5):
            rcv = port.read(10)
            print rcv
            fd=fd+rcv                 # Extract the complete data

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

        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)
```

The message will be displayed on the python shell as in the figure shown below.



```
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
AT


OK

ATE0


OK

OK




+CMTI: "
SM",8






+CMGR: "REC UNREAD","+919349829492","","15/08/03,16:47:38+22"
Hello Raspberry Pi

OK

>>>
```

Read the SIM900A AT command set for advanced options and transmit the corresponding commands to the modem to get the suitable responses. High end real time applications can be done through this interface in an easy manner which makes a new intelligent embedded world.