Fetching GPS Coordinates using Raspberry Pi 4 Model B and Quectel EC200U via SMS in URL format

Contents

Components Required: 2

Software Requirements: 3

Connections: 3

Python Code: 5

Procedure: 7

Output: 8

Components Required:

7Semi EC200U-CN LTE 4G GPS GNSS Mini Industrial Modem with inserted SIM card

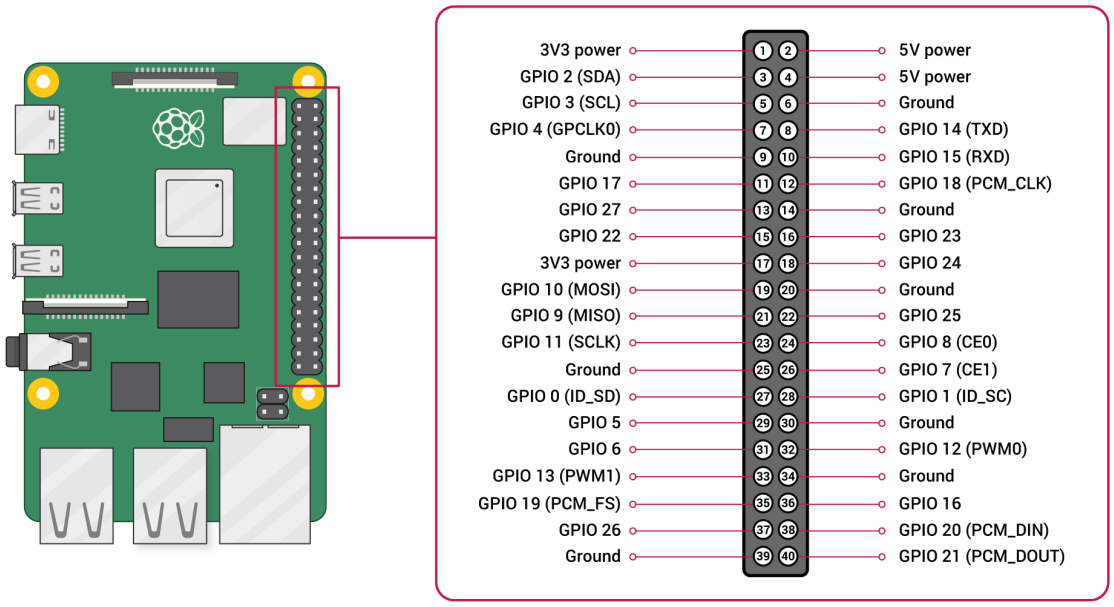
GPS External Active Antenna (3m) - SMA

Raspberry Pi 4 Model-B

Software Requirements:

- VNC Viewer in your system to communicate with Raspberry Pi (Any alternative can be used)
- In Raspberry Pi:
  - Python
  - Pyserial Library (for serial communication)
  - Minicom (To check whether AT commands are working)

Connections:



Raspberry Pi	EC200U
GPIO 15 (RXD)	TX
GPIO 14 (TXD)	RX
GND	GND

- GPS Antenna to GNSS of EC200U
- Antenna to Main of EC200U
- Insert SIM card (If it is not inserted previously)
- Give power supply to your Pi through the switch board.
- Give power supply to EC200U through your System or Raspberry Pi


GPS Antenna
Frequency:1575.42MHz
Voltage:3V~5V
SGS FC CE
15

Python Code:

```
import os
import time
import serial

#Sending AT Commands
def send_at_command(ser, command, expected_response, timeout=2):
    ser.write((command + '\r').encode())
    time.sleep(timeout)
    response = ser.read_all().decode()
    if expected_response not in response:
        print(f"Error: {response}")
        return False, response
    return True, response

#To enable GPS
def enable_gps(ser):
    print("Enabling GPS...")
    success, response = send_at_command(ser, 'AT+QGPS=1', 'OK')
    if not success:
        print(f"Failed to enable GPS: {response}")
    else:
        print("GPS enabled successfully.")
    return success

#To disable GPS
def disable_gps(ser):
    print("Disabling GPS...")
    success, response = send_at_command(ser, 'AT+QGPSEND', 'OK')
    if success:
        print("GPS disabled successfully.")
    else:
        print(f"Failed to disable GPS: {response}")

#To get location coordinates and change it to URL format
def get_location(ser):
    if enable_gps(ser):
        print("Waiting for GPS fix...")
        max_retries = 10
        retry_interval = 10  # seconds
        for attempt in range(max_retries):
            success, response = send_at_command(ser, 'AT+QGPSLOC=2', '+QGPSLOC:',
timeout=5)
            if success:
```

```
                disable_gps(ser)
                lat_long = response.split(',')[1:3]
                if len(lat_long) == 2:
                                                                    return
f"http://maps.google.com/?q={lat_long[0]},{lat_long[1]}"
            else:
                print(f"Attempt {attempt + 1} failed: {response}")
                time.sleep(retry_interval)
        disable_gps(ser)
    return None

#To send SMS to a specific phone number
def send_sms(ser, phone_number, message):
    success, _ = send_at_command(ser, 'AT+CMGF=1', 'OK')
    if success:
        success, _ = send_at_command(ser, f'AT+CMGS="{phone_number}"', '>')
        if success:
            ser.write((message + '\x1A').encode())
            time.sleep(5)
            if 'OK' in ser.read_all().decode():
                return True
    return False

def main():
    phone_number = 'XXXXXXXXXX'  # Replace with the recipient's phone number with
ISD
    try:
        ser = serial.Serial('/dev/ttyS0', baudrate=115200, timeout=1)
        ser.dtr = True
    except serial.SerialException as e:
        print(f"Error opening serial port: {e}")
        return

    location_url = get_location(ser)
    if location_url:
        print(f"Location URL: {location_url}")
        if send_sms(ser, phone_number, location_url):
            print("SMS sent successfully")
        else:
            print("Failed to send SMS")
    else:
        print("Failed to get location")

    ser.close()
```

```
if _name_ == "_main_":
    main()
```

Procedure:

1. Make connections as mentioned above.
2. Display the pi using VNC viewer in your system.
3. Open terminal and type:

   //To update and upgrade your raspberry pi, use

   - `sudo apt update`
   - `sudo apt upgrade`

   //To manually test the AT commands to verify the GPS functionality

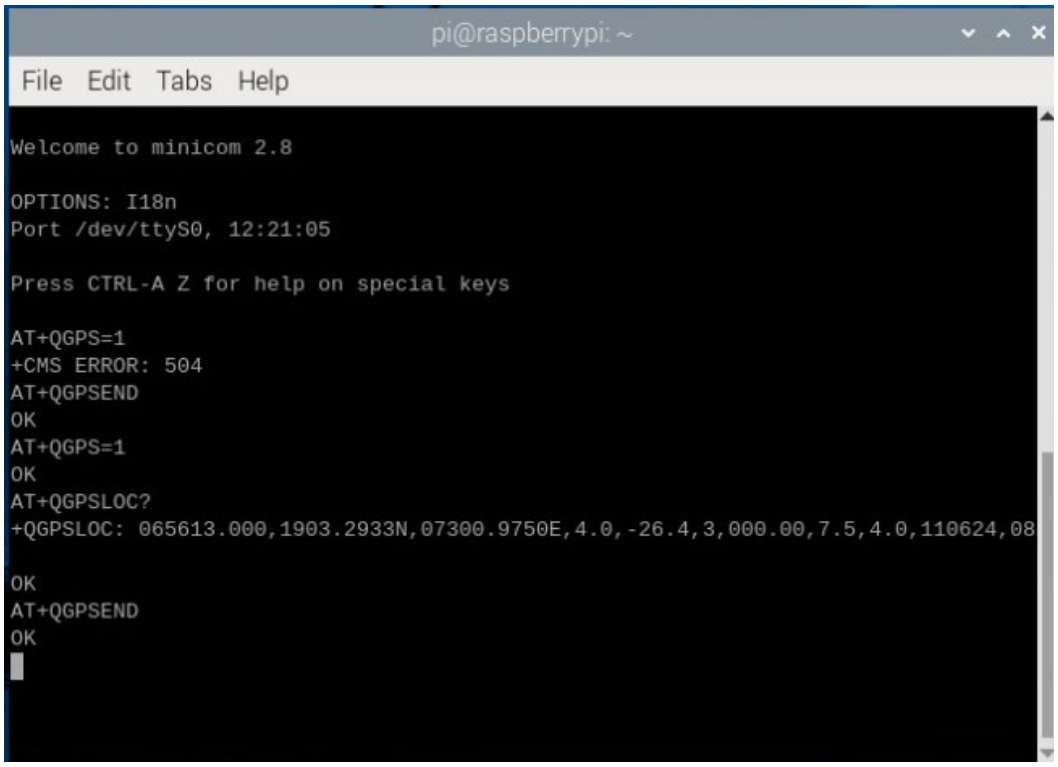
   - `sudo apt-get install minicom`
   - `sudo minicom -D /dev/ttyS0 -b 115200`

   //Send AT commands in minicom:

   - `AT  # Check if module responds with OK`
   - `AT+CPIN?  # Check if SIM is ready (should respond with +CPIN: READY)`
   - `AT+CSQ  # Check signal quality (should respond with +CSQ: <rssi>,<ber>)`
   - `AT+CGATT?  # Check if attached to network (should respond with +CGATT: 1)`
   - `AT+QGPS=1  # Turn on GPS`
   - `AT+QGPSCFG="outport","usbnmea" # Optional: Configure output port if needed`
   - `AT+QGPSLOC?  # Request GPS location`
   - `AT+QGPSEND  # Turn off GPS`
4. Write the python script in Thonny or IDLE.
5. Now, Run the python script and get location in SMS to the recipient's phone number.

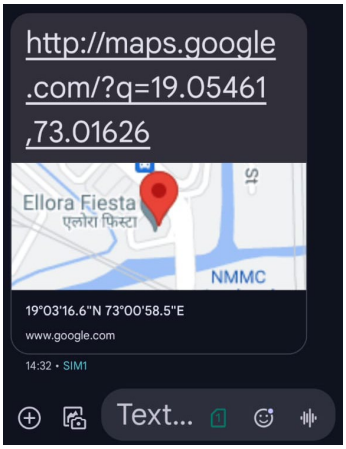
Output:

Verifying AT commands in minicom:



Python Shell Output:

SMS Received:



Redirecting to Google Maps: