Fetching GPS coordinates with the help of EC200U using Python

Contents

Components required:

7Semi EC200U-CN LTE 4G GPS GNSS Mini Industrial Modem

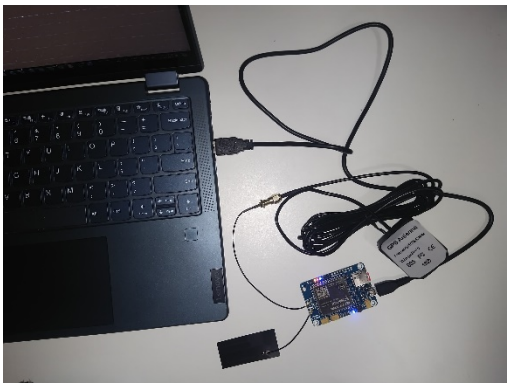
GPS External Active Antenna (3m) - SMA

Software Requirements:

- Python IDLE 3.10 (or later)

Connections:

- Connect the Antenna and GPS Antenna to Main and GNSS of EC200U respectively.
- Insert SIM card.
- Connect to your PC or system using USB cable (it doubles as power supply and for serial communication).



Code:

```
import serial

import time


# Configuration

serial_port = 'COMX'  # Replace ‘X’ with your COM port number

baud_rate = 115200

recipient_phone_number = '+XXXXXXXXXXXX'

# Replace with the recipient's phone number with ISD code

max_retries = 10  # Number of retries for getting the GPS location

retry_delay = 10  # Delay between retries in seconds


def send_at_command(ser, command, expected_response='', timeout=5):

    ser.write((command + '\r').encode())

    time.sleep(timeout)

    response = ser.read_all().decode()

    print(f'Sent: {command}')

    print(f'Received: {response}')

    if expected_response and expected_response not in response:

        raise Exception(f"Expected '{expected_response}' but got '{response}'")
```

```
    return response


def check_network_registration(ser):
    response = send_at_command(ser, 'AT+CREG?', '+CREG:')
    if '+CREG: 0,1' not in response and '+CREG: 0,5' not in response:
        raise Exception("Network registration failed. Please check the SIM card
and signal.")


def enable_gps(ser):
    response = send_at_command(ser, 'AT+QGPS?', '')
    if '+QGPS: 1' in response:  # GPS already enabled
        print("GPS already enabled")
        return True
    elif '+QGPS: 0' in response:  # GPS not enabled, try enabling
        print("Enabling GPS...")
        response = send_at_command(ser, 'AT+QGPS=1', 'OK')
        print("GPS enabled successfully")
        return True
    else:
        print("Unknown response to AT+QGPS?:", response)
        return False


def get_gps_coordinates(ser):
    if not enable_gps(ser):
        return None, None


    # Retry mechanism for GPS fix
    for attempt in range(max_retries):
        print(f"Attempt {attempt + 1} to get GPS coordinates...")
        response = send_at_command(ser, 'AT+QGPSLOC=2', '', 10)
```

```
        if '+CME ERROR' in response or '+CMS ERROR' in response:
            print(f"Failed to retrieve GPS coordinates: {response}")
        else:
            parts = response.split(',')
            if len(parts) >= 4:
                latitude = parts[1]
                longitude = parts[2]
                try:
                    if (-90 <= float(latitude) <= 90) and (-180 <= float(longitude) <= 180):
                        print("GPS coordinates retrieved successfully")
                        return latitude, longitude
                except ValueError:
                    print("Invalid GPS coordinates:", latitude, longitude)

        print(f"Retrying in {retry_delay} seconds...")
        time.sleep(retry_delay)

    print("Failed to obtain GPS coordinates after several attempts.")
    return None, None

def format_location_url(latitude, longitude):
    return f"http://maps.google.com/?q={latitude},{longitude}"

def send_sms(ser, message, recipient):
    try:
        send_at_command(ser, 'AT+CMGF=1', 'OK')  # Set SMS mode to text
        send_at_command(ser, f'AT+CMGS="{recipient}"', '>')  # Prepare to send SMS
        ser.write((message + '\x1A').encode())  # Send message with CTRL+Z
```

```
        time.sleep(2)
        response = ser.read_all().decode()
        print(f'SMS send response: {response}')
        if '+CMGS:' not in response:
            raise Exception(f"Failed to send SMS. Response: {response}")
    except Exception as e:
        raise Exception(f"Failed to send SMS: {e}")

def disable_gps(ser):
    try:
        send_at_command(ser, 'AT+QGPSEND', 'OK')
        print('GPS disabled successfully.')
    except Exception as e:
        print(f"Failed to disable GPS: {e}")

if __name__ == '__main__':
    ser = serial.Serial(serial_port, baud_rate, timeout=5)
    try:
        # Check connection
        send_at_command(ser, 'AT', 'OK')
        # Check network registration
        check_network_registration(ser)
        # Get location
        latitude, longitude = get_gps_coordinates(ser)
        if latitude and longitude:
            # Format location URL
            location_url = format_location_url(latitude, longitude)
            print(f'Location URL: {location_url}')
            # Send SMS
            send_sms(ser, location_url, recipient_phone_number)
```

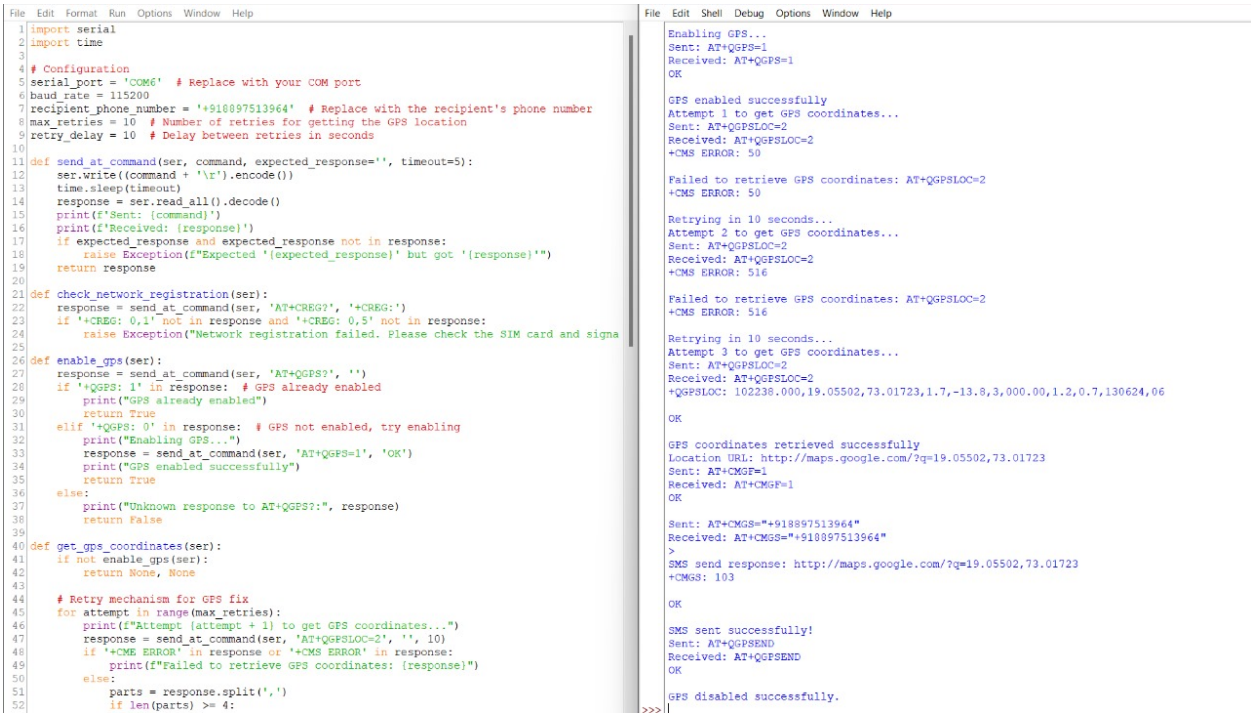
```
            print('SMS sent successfully!')
        else:
            print('Failed to obtain GPS coordinates.')
    except Exception as e:
        print(f'Error: {e}')
    finally:
        # Disable GPS
        disable_gps(ser)
        ser.close()
```

Procedure:

1. Open Command Prompt and install pyserial using:
   ```
   pip install pyserial
   ```
2. Open IDLE
3. Open New File
4. Copy the above code and paste it here, replace the port number and phone number
5. Save the file and Run the program.

Output:

Python File and Shell

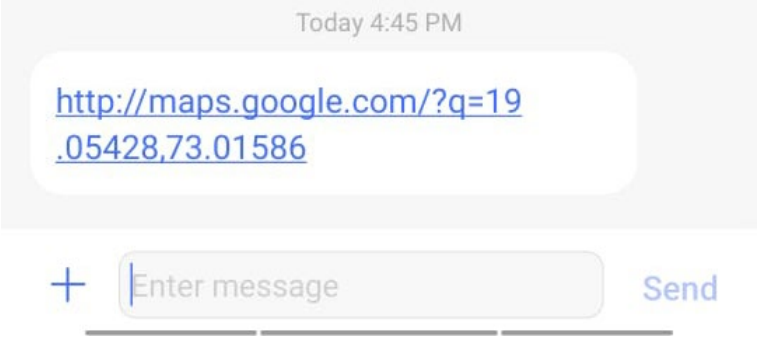


```
File  Edit  Format  Run  Options  Window  Help
import serial
import time

# Configuration
serial_port = 'COM6'  # Replace with your COM port
baud_rate = 115200
recipient_phone_number = '+918897513964'  # Replace with the recipient's phone number
max_retries = 10  # Number of retries for getting the GPS location
retry_delay = 10  # Delay between retries in seconds

def send_at_command(ser, command, expected_response='', timeout=5):
    ser.write((command + '\r').encode())
    time.sleep(timeout)
    response = ser.read_all().decode()
    print(f'Sent: {command}')
    print(f'Received: {response}')
    if expected_response and expected_response not in response:
        raise Exception(f"Expected '{expected_response}' but got '{response}'")
    return response

def check_network_registration(ser):
    response = send_at_command(ser, 'AT+CREG?', '+CREG:')
    if '+CREG: 0,1' not in response and '+CREG: 0,5' not in response:
        raise Exception("Network registration failed. Please check the SIM card and signa

def enable_gps(ser):
    response = send_at_command(ser, 'AT+QGPS?', '')
    if '+QGPS: 1' in response:  # GPS already enabled
        print("GPS already enabled")
        return True
    elif '+QGPS: 0' in response:  # GPS not enabled, try enabling
        print("Enabling GPS...")
        response = send_at_command(ser, 'AT+QGPS=1', 'OK')
        print("GPS enabled successfully")
        return True
    else:
        print("Unknown response to AT+QGPS?:", response)
        return False

def get_gps_coordinates(ser):
    if not enable_gps(ser):
        return None, None

    # Retry mechanism for GPS fix
    for attempt in range(max_retries):
        print(f"Attempt {attempt + 1} to get GPS coordinates...")
        response = send_at_command(ser, 'AT+QGPSLOC=2', '', 10)
        if '+CME ERROR' in response or '+CMS ERROR' in response:
            print(f"Failed to retrieve GPS coordinates: {response}")
        else:
            parts = response.split(',')
            if len(parts) >= 4:
```

```
File  Edit  Shell  Debug  Options  Window  Help
Enabling GPS...
Sent: AT+QGPS=1
Received: AT+QGPS=1
OK

GPS enabled successfully
Attempt 1 to get GPS coordinates...
Sent: AT+QGPSLOC=2
Received: AT+QGPSLOC=2
+CMS ERROR: 50

Failed to retrieve GPS coordinates: AT+QGPSLOC=2
+CMS ERROR: 50

Retrying in 10 seconds...
Attempt 2 to get GPS coordinates...
Sent: AT+QGPSLOC=2
Received: AT+QGPSLOC=2
+CMS ERROR: 516

Failed to retrieve GPS coordinates: AT+QGPSLOC=2
+CMS ERROR: 516

Retrying in 10 seconds...
Attempt 3 to get GPS coordinates...
Sent: AT+QGPSLOC=2
Received: AT+QGPSLOC=2
+QGPSLOC: 102238.000,19.05502,73.01723,1.7,-13.8,3,000.00,1.2,0.7,130624,06

OK

GPS coordinates retrieved successfully
Location URL: http://maps.google.com/?q=19.05502,73.01723
Sent: AT+CMGF=1
Received: AT+CMGF=1
OK

Sent: AT+CMGS="+918897513964"
Received: AT+CMGS="+918897513964"
>
SMS send response: http://maps.google.com/?q=19.05502,73.01723
+CMGS: 103

OK

SMS sent successfully!
Sent: AT+QGPSEND
Received: AT+QGPSEND
OK

GPS disabled successfully.
>>>
```

```
>>>
= RESTART: C:\Users\vinay\AppData\Local\Programs\Python\Python312\GPS_Location_Code.py
Sent: AT
Received: AT
OK

Sent: AT+CREG?
Received: AT+CREG?
+CREG: 0,1

OK

Sent: AT+QGPS?
Received: AT+QGPS?
+QGPS: 0

OK

Enabling GPS...
Sent: AT+QGPS=1
Received: AT+QGPS=1
OK

GPS enabled successfully
Attempt 1 to get GPS coordinates...
Sent: AT+QGPSLOC=2
Received: AT+QGPSLOC=2
+CMS ERROR: 50

Failed to retrieve GPS coordinates: AT+QGPSLOC=2
+CMS ERROR: 50

Retrying in 10 seconds...
Attempt 2 to get GPS coordinates...
Sent: AT+QGPSLOC=2
Received: AT+QGPSLOC=2
+QGPSLOC: 111500.000,19.05428,73.01586,4.3,-61.1,3,000.00,1.0,0.5,130624,08

OK

GPS coordinates retrieved successfully
Location URL: http://maps.google.com/?q=19.05428,73.01586
Sent: AT+CMGF=1
Received: AT+CMGF=1
OK

Sent: AT+CMGS="+918897513964"
Received: AT+CMGS="+918897513964"
>
SMS send response: http://maps.google.com/?q=19.05428,73.01586
+CMGS: 113

OK

SMS sent successfully!
Sent: AT+QGPSEND
Received: AT+QGPSEND
OK

GPS disabled successfully.
```

SMS Recieved



Location in Google Maps