

## MWC Experiment 4

Shivani Bhat

BE EXTC A  
2022200013

### Aim:

To use different AT Commands and find out different parameters for mobile communication

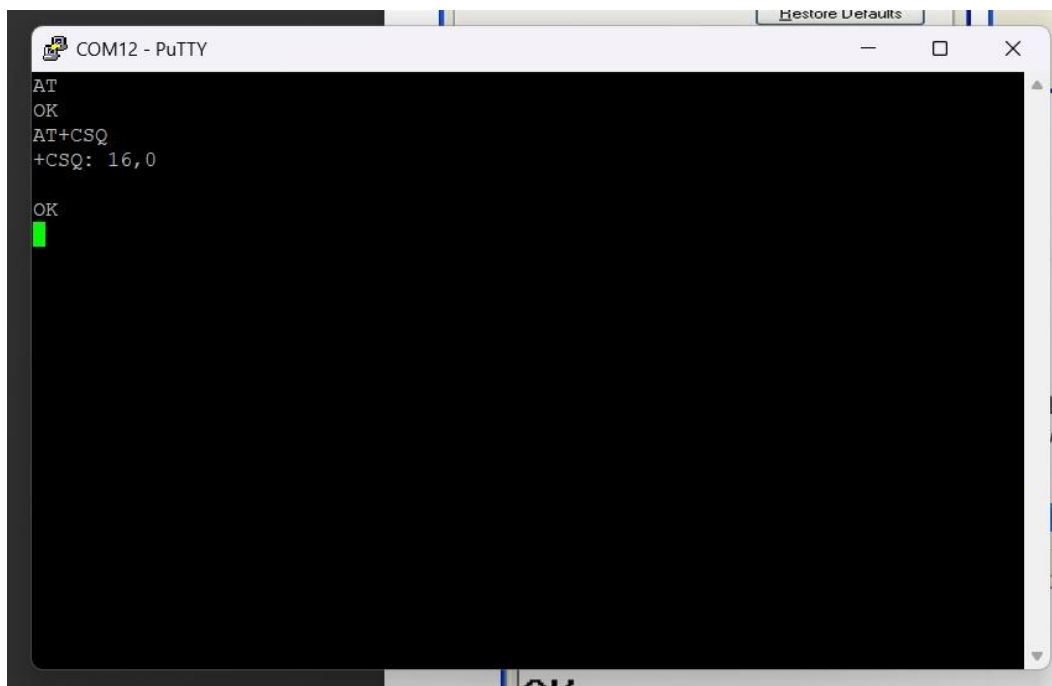
### Components Required:

Hardware: GSM Module, SIM Card (Non-Jio), PC

Software: PuTTY, Hyper Terminal

### Commands:

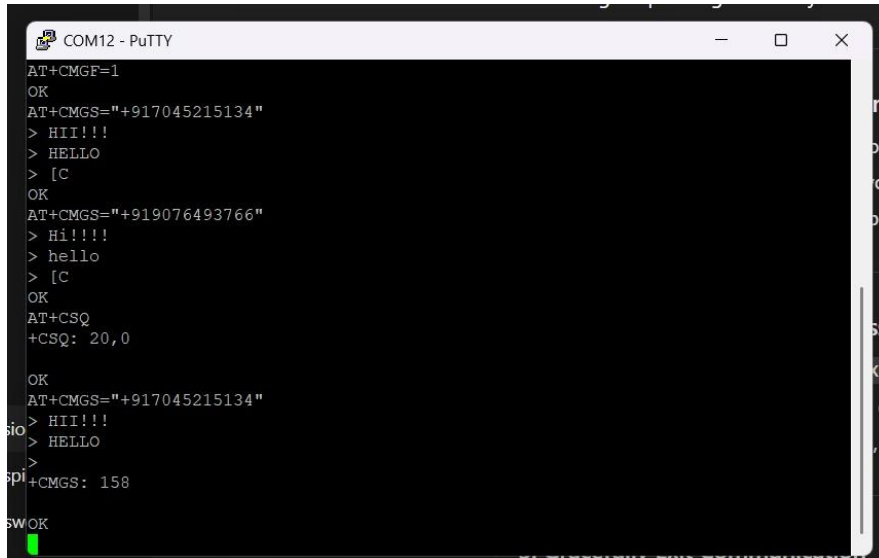
1. **AT + CSQ:** The AT+CSQ command is used to query the signal quality of the GSM network. It returns two values: the Received Signal Strength Indication (RSSI) and the Bit Error Rate (BER). RSSI helps determine how strong the connection is (on a scale of 0–31, where 31 indicates excellent strength), while BER gives an idea about the quality of the received signal. This command is crucial in wireless communication for assessing network coverage, troubleshooting connectivity issues, and ensuring stable communication links.



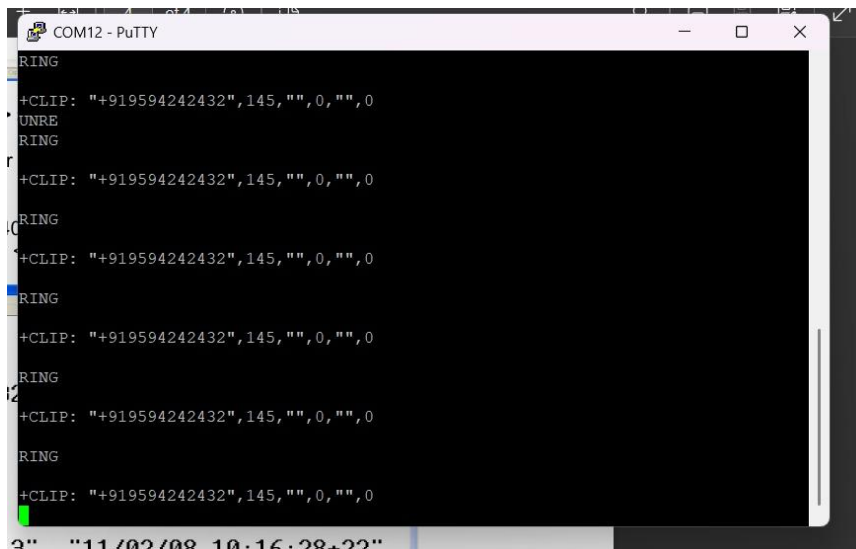
```
COM12 - PuTTY
AT
OK
AT+CSQ
+CSQ: 16,0
OK
```

2. **AT + CMGS:** The AT+CMGS command is used to send SMS messages from a GSM module or mobile device. After entering the recipient's phone number and message content, the

command instructs the device to transmit the SMS through the network. It supports both text mode and PDU (Protocol Data Unit) mode, allowing flexibility depending on the level of control needed. This command is widely used in IoT applications, remote monitoring systems, and messaging services.



3. **+ CLIP:** The +CLIP command enables the Calling Line Identification Presentation feature. When activated, the GSM module displays the caller's number during an incoming call, provided the network supports it and the caller hasn't restricted the number. This feature is important for applications where caller verification is needed, such as security systems or automated response units.



4. **AT + CMGL:** The AT+CMGL command is used to list SMS messages stored on the SIM card or device memory. It allows filtering messages based on status (e.g., unread, read, sent, or all). This command is commonly used in SMS-based applications where a device needs to process or respond to incoming messages, such as in M2M (machine-to-machine) communication.

```
NO CARRIER
AT+CMGL="REC UNREAD"
ERROR
AT+CMGL="REC UNREAD"
OK
```

5. **AT + CGMI:** The AT+CGMI command retrieves the manufacturer's identification of the GSM module. It is useful for developers and diagnostic tools to confirm the hardware vendor or check compatibility during integration. This helps when managing devices from different vendors in large-scale deployments.

```
AT+CGMI
SIMCOM_Ltd
OK
```

6. **+CMTI :** The +CMTI command is an unsolicited result code that notifies the device when a new SMS message has arrived. It provides the memory location where the message is stored, so the user or system can fetch it using read commands. This feature is critical for real-time SMS-triggered applications such as alerts, authentication, and control systems.

```
+CMTI: "SM",1

RING

+CLIP: "+917045215134",145,"",0,"",0

RING

+CLIP: "+917045215134",145,"",0,"",0
ATA
OK
```

7. **AT+CGMM:** The AT+CGMM command returns the model identification of the GSM module. Like AT+CGMI, it helps in device recognition and inventory management, but specifically gives the model rather than the manufacturer. This is useful when multiple models from the same vendor are deployed.

```
AT+CGMM
SIMCOM_SIM800A
OK
```

8. **AT + CGMR:** The AT+CGMR command retrieves the revision number of the firmware running on the GSM module. It is valuable for debugging, ensuring firmware compatibility, and updating modules when new software features or patches are released.

```
AT+CGMR
Revision:1418B05SIM800A24
OK
```

9. **ATD+(number) :** The ATD command followed by a number initiates a voice call to the specified phone number. It works similarly to dialing on a regular phone. In wireless communication applications, it is used in modules that support voice features, enabling devices to automatically place calls in emergency or service-based applications.

```
OK
ATD+917045215134;
OK
ath
OK
```

10. **AT+COPS?:** The AT+COPS? command queries the current network operator selected by the GSM module. It provides information about whether the operator is automatically chosen or manually set. This is essential in roaming scenarios or when forcing a device to connect to a specific network for reliability or cost reasons.

```
AT+COPS?
+COPS: 0,0,"AirTel"
OK
```

11. **AT+CPIN:** The AT+CPIN command is used to enter the PIN code of a SIM card, allowing the GSM module to unlock and register to the network. Without providing the correct PIN (if required), the device cannot access network services. This is important for SIM security and network authentication.

```
AT+CPIN?
+CPIN: READY
OK
```

12. **AT+CGSN:** The AT+CGSN command retrieves the device's International Mobile Equipment Identity (IMEI) number. The IMEI is a unique identifier for GSM devices, used by networks for

authentication, tracking, and blocking stolen devices. Applications that require secure device identification often rely on this command.

```
AT+CGSN
868440031425571
OK
```

13. **AT+CUSD:** The AT+CUSD command is used to send Unstructured Supplementary Service Data (USSD) messages. USSD is widely used for real-time services such as checking balance (\*123#), mobile banking, or activating network services. The command facilitates interactive, session-based communication between users and mobile network operators.

```
AT+CUS=1,"*123#"
ERROR
AT+CUSD=1,"*123#"
OK

+CUSD: 0, "O/G Validity: 11 Oct 2025, Bal:Rs 0.23.", 15
```

14. **AT+CPBS:** The AT+CPBS command selects the phonebook storage memory (e.g., SIM, device, or fixed dial). Different storages have different capacities and access rules. This command is useful in applications involving contact management, automated dialing, or number verification.

15. **AT + CPBR:** The AT+CPBR command reads phonebook entries from the currently selected storage (chosen with AT+CPBS). It allows retrieval of saved contacts, including their phone numbers and names. This is particularly relevant in applications requiring automated access to stored contact information, such as call automation or bulk messaging.

```
AT+CPBS="SM"
OK
AT+CPBR=1
+CPBR: 1, "121", 129, "customer care"
OK
```

## Conclusion:

### 1. AT+CSQ (Signal Quality Test)

- **Observation:** Returned signal strength values (RSSI) from 0–31 and bit error rate (BER). Indoors the values were weaker (~12–18), outdoors stronger (~20–28).
- **Takeaway:** A minimum RSSI of ~12 is needed for stable SMS/call operations. Helps in antenna placement and ensuring reliable network connectivity.

## 2. AT+CMGS (Send SMS Message)

- **Observation:** SMS could be sent in text mode by entering the number and message, ending with Ctrl+Z. Delivery time varied with network traffic.
- **Takeaway:** Very reliable for notifications and IoT alerts, but SMS is not real-time guaranteed. Shouldn't be used for time-critical communication.

## 3. +CLIP (Caller Line Identification Presentation)

- **Observation:** Caller number displayed correctly when enabled, but hidden numbers were shown as "unknown."
- **Takeaway:** Useful for screening and verifying incoming calls, but dependent on network support and caller preferences.

## 4. AT+CMGL (List Messages)

- **Observation:** Could retrieve SMS by status (ALL, UNREAD, READ). Deletion required another command (AT+CMGD).
- **Takeaway:** Suitable for SMS-based automation. But unless paired with +CMTI, it requires polling, which consumes extra processing.

## 5. AT+CGMI (Manufacturer Identification)

- **Observation:** Returned "SIMCOM" (vendor name).
- **Takeaway:** Important for verifying device vendor in mixed-hardware deployments and for debugging compatibility issues.

## 6. +CMTI (New SMS Indication)

- **Observation:** Automatically gave notification +CMTI: "SM", index when a new SMS arrived.
- **Takeaway:** Enables **event-driven systems**—no need to poll for SMS. Very efficient for real-time SMS-triggered actions.

## 7. AT+CGMM (Model Identification)

- **Observation:** Returned model like "SIM800L".

- **Takeaway:** Confirms exact hardware model, especially important when one firmware is used for multiple GSM modules.

#### 8. AT+CGMR (Firmware Revision)

- **Observation:** Gave firmware revision string, e.g., "Revision:1418B05SIM800L24".
- **Takeaway:** Crucial for debugging—some commands or behaviors only work on certain firmware versions. Version tracking ensures consistency.

#### 9. ATD+(number) (Dial a Call)

- **Observation:** Outgoing voice calls were placed successfully; needed ATH to hang up.
- **Takeaway:** Useful for emergency dialers, call automation, or IVR systems. Requires audio hardware for full two-way communication.

#### 10. AT+COPS? (Query Operator Selection)

- **Observation:** Returned current operator like "Airtel" or "Vodafone".
- **Takeaway:** Confirms successful network registration. Helps in roaming scenarios and when forcing specific operator selection.

#### 11. AT+CPIN (Enter SIM PIN)

- **Observation:** Returned "READY" if no PIN was needed. If SIM was locked, "SIM PIN" prompt appeared. Correct PIN unlocked the network.
- **Takeaway:** Provides SIM security and prevents unauthorized usage. Essential for embedded/remote systems with SIMs.

#### 12. AT+CGSN (Request IMEI Number)

- **Observation:** Returned the unique IMEI of the module.
- **Takeaway:** Acts as a unique device ID. Very useful for authentication, device management, and blacklist checks.

#### 13. AT+CUSD (Send USSD Command)

- **Observation:** Commands like \*123# worked, returning balance instantly.
- **Takeaway:** Provides **real-time services** (balance, recharge, banking). Faster than SMS because it is session-based and doesn't store messages.

#### 14. AT+CPBS (Select Phonebook Storage)

- **Observation:** Could switch between SIM memory ("SM") and device memory ("ME").
- **Takeaway:** Offers flexibility in storing contacts. Useful for automated dialers that rely on pre-saved numbers.

### 15. AT+CPBR (Read Phonebook Entries)

- **Observation:** Retrieved contact names and numbers from selected storage.
- **Takeaway:** Enables programmatic access to contacts, ideal for auto-dialer systems, call centers, or bulk messaging setups.

## Part 2:

For part 2 of the experiment, we used the commands above in a serial terminal with python using Thonny.

### Source Code:

```
import serial
```

```
import time
```

```
def send_at_command(command, serial_port, timeout=1):
```

```
    """
```

Sends an AT command to the GSM module and waits for the response.

Parameters:

command (str): The AT command to send.

serial\_port (serial.Serial): The serial connection to the GSM module.

timeout (int): The time to wait for a response (in seconds).

Returns:

str: The response from the GSM module.

```
    """
```

```
    serial_port.write((command + '\r').encode()) # Send command with \r to denote Enter
```

```
    time.sleep(timeout)
```

```
    response = serial_port.read_all().decode() # Read the response
```

```
    return response
```



```

def main():

    # Configure the serial port connection

    port = '/dev/ttyUSB0' # Replace with your serial port name

    baud_rate = 115200 # Adjust according to the GSM modem's baud rate

    timeout = 5 # Timeout for reading from the serial port

    # Open serial port connection

    try:

        gsm_modem = serial.Serial(port, baud_rate, timeout=timeout)

        print(f"Connected to GSM modem on {port}")


    # Example AT commands

    at_commands = [

        'AT', # Basic test command

        'AT+CSQ', # Signal quality

        'AT+CREG?', # Network registration status

        'AT+COPS?', # Operator selection

        'AT+CMGF=1', # Set SMS to text mode

        'AT+CMGS="phone_number"', # Send SMS (you need to enter the phone number)

        'Hello, this is a test message.' + chr(26) # Text message ends with Ctrl+Z (chr(26))

    ]


    # Send each command and print the response

    for command in at_commands:

        response = send_at_command(command, gsm_modem)

        print(f"Command: {command}\nResponse: {response}\n")


    # Close the serial connection

    gsm_modem.close()

    except serial.SerialException as e:

        print(f"Error: {e}")

```

```
if __name__ == '__main__':
```

```
    main()
```

## Outputs:

```
Command: AT+CGATT?
AT+CGATT?
+CGATT: 1
```

```
OK
```

```
Command: AT+CMGF=1
AT+CMGF=1
OK
```

```
Command: AT+CMGL="ALL"
AT+CMGL="ALL"
OK
```

```
Command: AT+CSQ
AT+CSQ
+CSQ: 24,0
```

```
OK
```

```
Command: AT+CREG?
AT+CREG?
+CREG: 0,1
```

```
OK
```

```
Command: AT+COPS?
AT+COPS?
+COPS: 0,0,"Orange"
```

```
OK
```

```
Command: AT+CGATT?
AT+CGATT?
+CGATT: 1
```

```
OK
```

```
Command: AT+CMGF=1
AT+CMGF=1
OK
```

```
Command: AT+CMGL="ALL"
AT+CMGL="ALL"
OK
```

```
Command: AT+CCID
AT+CCID
8991200050572736275f
```

```
OK
```

```
Command: AT+CPAS
AT+CPAS
+CPAS: 0
```

```
OK
```

```
Command: AT+CPBS
AT+CPBS
ERROR
```

```
Command: ATA
ATA
ERROR
```

```
Command: ATH
ATH
OK
```

```
Command: AT+CGMI
AT+CGMI
SIMCOM_Ltd
```

```
Command: AT+CGSN
AT+CGSN
864495038220924
```

```
Command: AT
AT
OK

Command: AT+CSQ
AT+CSQ
+CSQ: 17,0

OK

Command: AT+CREG?
AT+CREG?
+CREG: 0,1

OK

Command: AT+COPS?
AT+COPS?
+COPS: 0,0,"Orange"

OK

Command: AT+CGATT?
AT+CGATT?
+CGATT: 1

OK
```

```
Command: AT+CGSN
AT+CGSN
864495038220924

OK

--- Making a Call ---

Command: ATD+918530837267;
ATD+918530837267;
OK

Command: ATH

--- Sending SMS ---

Command: AT+CMGF=1
AT+CMGF=1
OK

Call Ready

SMS Ready

Command: AT+CMGS="+918530837267"
AT+CMGS="+918530837267"
>
```

## Conclusion:

- **Connectivity and Diagnostics:** Commands like **AT+CSQ** and **AT+COPS?** proved essential for real-time network diagnostics. We observed that signal strength (RSSI) is a critical parameter for reliable communication, with a minimum value necessary for stable operation. This reinforces the importance of proper antenna placement in wireless applications.
- **Automation and Communication:** The experiment highlighted the power of AT commands in automating machine-to-machine (M2M) communication. We successfully sent an SMS using **AT+CMGS** and understood the importance of **+CMTI** for efficient, event-driven message handling, which is far superior to constant polling. The ability to place calls (**ATD+**) and access phonebook data (**AT+CPBR**) also showed the potential for developing automated dialers and contact-based systems.
- **Device Management and Security:** Commands such as **AT+CGSN** (IMEI) and **AT+CGMI** (Manufacturer ID) are crucial for device management and tracking in large-scale deployments. The use of **AT+CPIN** also underscored the role of AT commands in managing SIM security, which is vital for protecting remote devices from unauthorized network access.
- **Programming Interface:** The second part of the experiment effectively demonstrated how AT commands can be integrated into a programming environment like **Python**. By using a serial library, we created a software-based bridge to the hardware, proving that these commands are not just for manual terminal use but serve as a robust, low-level Application Programming Interface (API) for developing custom mobile communication applications.