GSM_TESTER

Abstract

The GSM Connectivity Testing Tool is a Python-based GUI application designed to simplify the testing, management, and debugging of GSM modules, such as SIM7070G, in IoT and embedded applications. Traditional GSM testing requires developers to manually write firmware for microcontrollers to send AT commands, check network registration, configure APN settings, and perform HTTP/HTTPS communication. This process is time-consuming, prone to errors, and requires advanced knowledge of embedded systems and serial communication.

The proposed tool provides a user-friendly interface to select COM ports, set baud rates, and connect to GSM modules via USB-TTL interfaces. It enables both manual and automated AT command transmission, real-time TX/RX logging, signal strength monitoring, SIM status verification, APN configuration, and HTTP/HTTPS connectivity testing. By automating these tasks, the tool reduces setup time, improves reliability, and allows developers to focus on higher-level application development rather than low-level module debugging. This software is essential for IoT engineers, embedded system developers, and network testers, providing comprehensive diagnostics and accelerating GSM-enabled device development.

Existing System

In the existing GSM testing system, developers often rely on microcontrollers or embedded platforms such as Arduino, ESP32, or Raspberry Pi to test GSM modules. The testing process involves manually sending AT commands, writing firmware for network registration checks, configuring APN settings, and handling HTTP/HTTPS communication.

The major limitations of the existing system include:

- Requires extensive programming knowledge.
- Manual command input increases the risk of human error.
- Monitoring signal strength and SIM status is time-consuming.
- Debugging network or module issues is complex.
- No centralized interface for viewing TX/RX communication logs.

These limitations make module testing inefficient, especially for rapid prototyping or large-scale deployment.

Proposed System

The proposed GSM Connectivity Testing Tool introduces a GUI-based solution to simplify GSM module testing. Key features of the system include:

- Selection of COM port and baud rate for flexible communication.
- Manual and automated AT command transmission.
- Real-time TX/RX logging for monitoring all communication.
- Automated GSM network setup, including signal strength verification and APN configuration.
- HTTP/HTTPS connectivity testing for validating network access.
- Prompt-based configuration for APN and URLs for ease of use.

This system eliminates the need for extensive embedded programming, reduces testing time, and provides a centralized interface for monitoring GSM module activity, making it ideal for IoT development and GSM network diagnostics.

Method

The GSM Connectivity Testing Tool is implemented using Python, Tkinter for the GUI, and PySerial for serial communication with GSM modules. The workflow of the tool is as follows:

1. **Hardware Connection**:
    - Connect GSM module to a USB-TTL interface.
    - GSM TX → TTL RX, GSM RX → TTL TX, GND → GND.
2. **Software Setup**:
    - Launch GSM_TESTER.exe.
    - Select the COM port and baud rate (default 9600).
3. **AT Command Interaction**:
    - Send manual AT commands using an input box.
    - Use automatic commands via "Connect GSM" or "Send HTTP ATs" buttons.
4. **GSM Network Connection**:
    - The software checks SIM status and network registration.
    - Signal strength is monitored and displayed in real time.
    - APN settings are configured via prompt.
5. **HTTP/HTTPS Communication**:
    - Configure URL and APN.
    - Establish HTTP connection and send requests.
    - Read responses and display TX/RX logs.
6. **Logging and Diagnostics**:
    - All transmitted and received commands are logged in real time.
    - Errors and responses are displayed for troubleshooting.
7. **Automation & Reliability**:
    - Automatic retries for AT commands.
    - Link restart and error handling ensures robust communication.

This method ensures that GSM modules can be tested efficiently, with minimal manual intervention, providing a reliable and user-friendly interface for developers and network testers.