Electronic Basics #11: Sending SMS with Arduino || TC 35 GSM Module

Introduction to TC35 GSM Module

The **TC35 GSM module** is a compact and powerful device that enables wireless communication through the GSM network. It allows a microcontroller, such as an **Arduino**, to send **SMS messages**, make **voice calls**, and connect to the internet using the GSM network. This module is used in various applications like remote control systems, alarms, and IoT projects.

The **TC35** module can communicate with other devices through **serial communication** (UART) and is compatible with standard **SIM cards** to establish mobile network connections.

Basic Structure of TC35 GSM Module

The **TC35 GSM module** consists of several key components:

- **SIM Card Slot**: A slot to insert a **GSM SIM card**, which connects the module to the mobile network.
- **RS232/TTL Interface**: Serial communication pins for connecting with microcontrollers (e.g., Arduino).
- **Power Supply Pins**: Provides power to the module.
- **Status Indicators (LEDs)**: Indicate the working status of the module, such as network connection, power, and activity.
- **Audio Pins**: Used for voice communication but not needed for SMS-related operations.

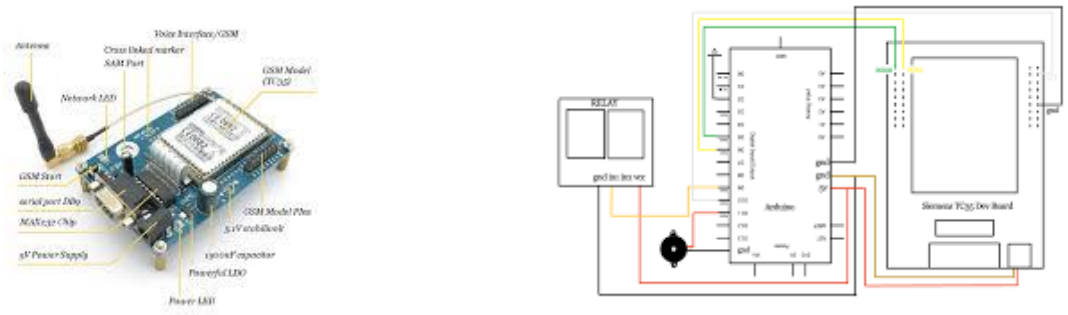

Fig11.1: TC35 GSM Module

Inserting a SIM Card

To use the **TC35 GSM module**:

1. Insert a **GSM SIM card** into the SIM card slot on the module.
2. Make sure that the SIM card is activated and has sufficient balance or data (if required).
3. Ensure that the module is powered through the **VCC** and **GND** pins.

Once the SIM card is inserted and the module is powered on, it will attempt to connect to the GSM network, and the **STATUS LED** will indicate the connection status (blinking for searching network and solid when connected).

Connecting TC35 GSM Module with Arduino

To interface the **TC35 GSM module** with an **Arduino**, you need to connect the following pins:

- **TX (Transmit)** of the GSM module to **RX (Receive)** on Arduino (Pin 0).
- **RX (Receive)** of the GSM module to **TX (Transmit)** on Arduino (Pin 1).
- **VCC** of the GSM module to **5V** on the Arduino.
- **GND** of the GSM module to **GND** on the Arduino.

You may also need to use **voltage level shifters** if the TC35 uses **RS232 levels** and the Arduino uses **TTL levels**.

Sending an SMS using TC35 and Arduino

To send an SMS using the **TC35 GSM module** and **Arduino**, the following steps are involved:

1. **Initialize the GSM Module**: Send AT commands to the module to initialize it.
2. **Send SMS Command**: Use the AT command AT+CMGS to send an SMS message to a recipient's phone number.
3. **Receive Acknowledgment**: The GSM module will respond with an acknowledgment that the message was sent successfully.

Here's an example **Arduino code** to send an SMS using the TC35 GSM module:

```
#include <SoftwareSerial.h>

SoftwareSerial gsmSerial(7, 8); // RX, TX pins for GSM communication

void setup() {
  // Start serial communication with the GSM module
  gsmSerial.begin(9600); // Baud rate for GSM module
  Serial.begin(9600);    // Baud rate for Arduino serial monitor

  // Initialize the GSM module
  Serial.println("Initializing GSM module...");
  delay(1000);
```

```
  gsmSerial.println("AT");          // Check communication
  delay(1000);
  gsmSerial.println("AT+CMGF=1");    // Set SMS format to text mode
  delay(1000);
  gsmSerial.println("AT+CSCS=\"GSM\"");  // Set character set to GSM
  delay(1000);
}

void loop() {
  // Send an SMS
  gsmSerial.println("AT+CMGS=\"+1234567890\""); // Replace with the recipient's phone number
  delay(1000);
  gsmSerial.println("Hello, this is a test message!");  // The message text
  delay(1000);
  gsmSerial.write(26);    // Send Ctrl+Z to indicate the end of the message
  delay(1000);

  Serial.println("Message sent!");
  delay(5000);  // Delay before sending the next SMS
}
```

Explanation of Code:

- **SoftwareSerial** is used to communicate with the GSM module through pins 7 (RX) and 8 (TX).
- gsmSerial.begin(9600) initializes the serial communication with the GSM module at a baud rate of 9600.
- AT+CMGF=1 sets the SMS format to text mode, which is the most common format.
- AT+CSCS="GSM" specifies the GSM character set for SMS.
- AT+CMGS="<PhoneNumber>" is used to set the recipient’s phone number for the SMS.
- gsmSerial.write(26) sends a **Ctrl+Z** (ASCII value 26), signaling the end of the message.
- The Serial.println command displays a message in the Arduino Serial Monitor for feedback.

Key AT Commands for GSM Communication

- **AT**: Check if the module is working (response should be "OK").
- **AT+CMGF=1**: Set the SMS mode to text.
- **AT+CMGS**: Send an SMS message to a phone number.
- **AT+CSCS="GSM"**: Set the character set for the SMS to GSM.
- **AT+CREG?**: Check if the GSM module is registered on the network.

Conclusion

The **TC35 GSM module** is a versatile component for adding **SMS** functionality to your **Arduino projects**. It allows simple text message communication, which can be used for various applications, including remote control, alarms, and IoT devices. By sending AT commands via a serial connection, the module can interact with other devices, and with a few simple steps, you can integrate SMS functionality into your projects.