DATA COMMUNICATION
LAB 3: GSM MODULE

I. Introduction to GSM

GSM stands for **Global System for Mobile communications**, which is a standard developed by the European Telecommunications Standards Institute to describe the protocols for second-generation digital cellular networks used by mobile devices such as tablets.

There are many available modules supporting GSM functions such as SIM900 or SIM800L (see Figure 1 and Figure 2). These modules are designed based on SIM900 Quad-band GSM/GPRS modules (support 850/900/1800/1900MHz network)which can be used throughout the world.

Figure 1:GSM Module using SIM900

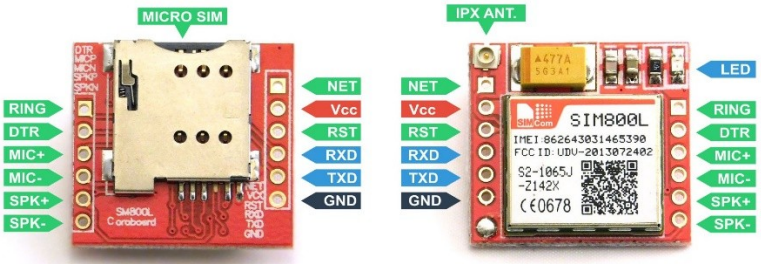


Figure 2: GSM module using SIM800L

In order to control GSM modules, AT commands are sent via serial communication. Therefore, the Tx and Rx of NodeMCU are connected to Rx and Tx of the GSM module, respectively.

II. AT Commands to Send a Message

In this manual, a terminal in PC is used to send AT Commands to the GSM module. This tool is very popular to test the module is working well or not.

Step 1: Press the PKey button on the GSM module, until the NetLight (blue light) is blink

Step 2: Download the terminal from https://www.dropbox.com/s/7xuwege5pjv6fis/Terminal.exe?dl=0. Open Terminal, scan the port, set the speed and then click Connect.

Step 3: Send the first AT command to test the connection. Type **AT**, check on **CR** and the click **Send**. If the connection is good, the AT OK should be received in the Terminal. (See figure bellow)

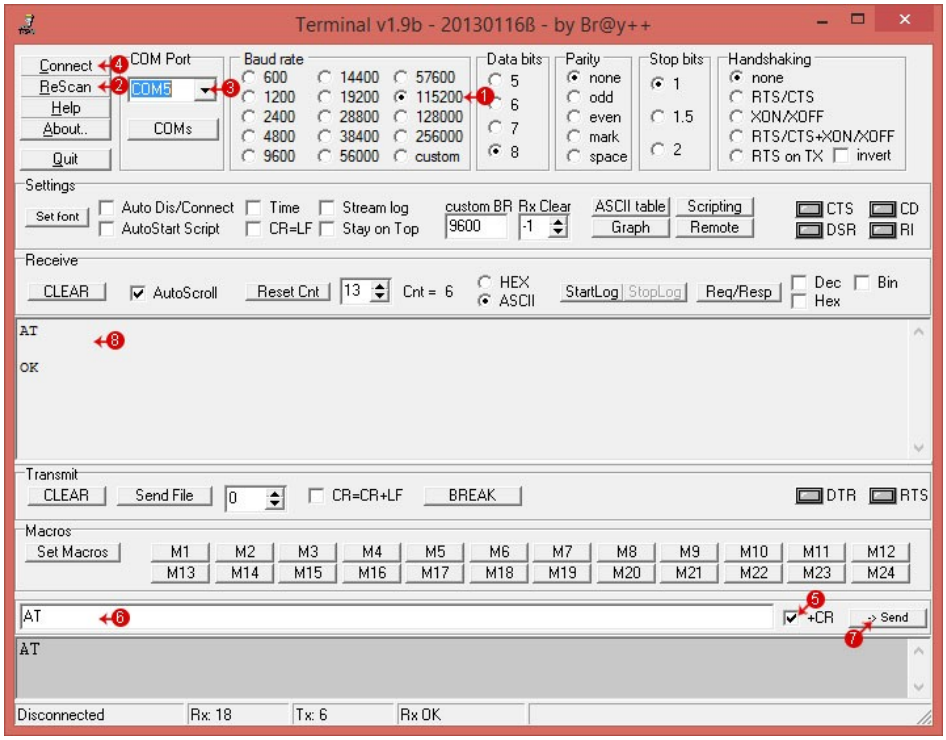


Figure 3: Terminal on PC

After the connection and the pin are correct. Following steps are required to send a message:

1. Send AT+CMGF=1
2. Send AT+CSCS="GSM"
3. Send destination number: AT+CMGS="84906362340"
4. Send text message
5. Send 0x1A

III. Macro Setting in Terminal

For easy testing, you can set some macro on Terminal by clicking the **Set Macros** button, which is in the left corner. Please check the figure bellow for more details.

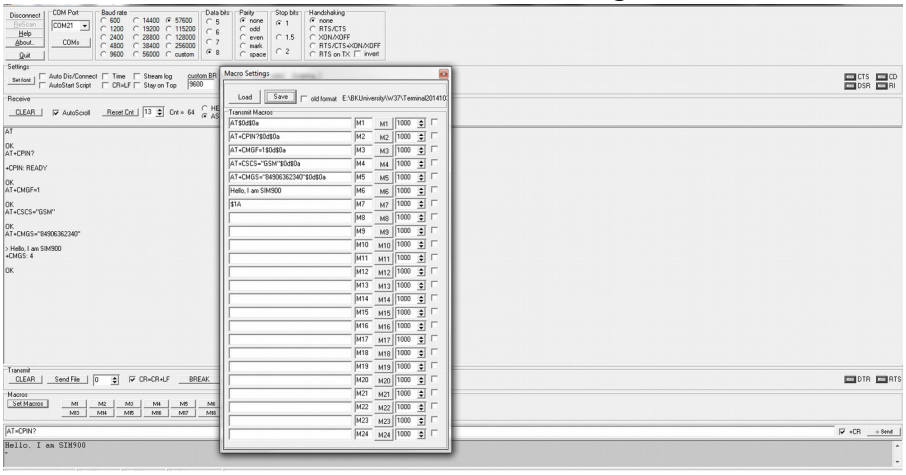


Figure 4: Macro setting in Terminal

For example, the first macro is set to M1 button and the value of this macro is "AT$0d$0a", which is the "AT" plus a carry return character ($0d$0a). After this setting, you just need to click M1 button. The behavior will be the same as sending AT plus a carry return. Following values are used for the macros:

- AT$0d$0a
- AT+CPIN?$0d$0a
- AT+CMGF=1$0d$0a
- AT+CSCS="GSM"$0d$0a
- AT+CMGS="84906362340"$0d$0a
- Hello, I am SIM900
- $1A

Then, by click king M1, M2, M3, M4, M5, M6, a message "Hello, I am SIM900" is sent to the phone which number is 0906362340.

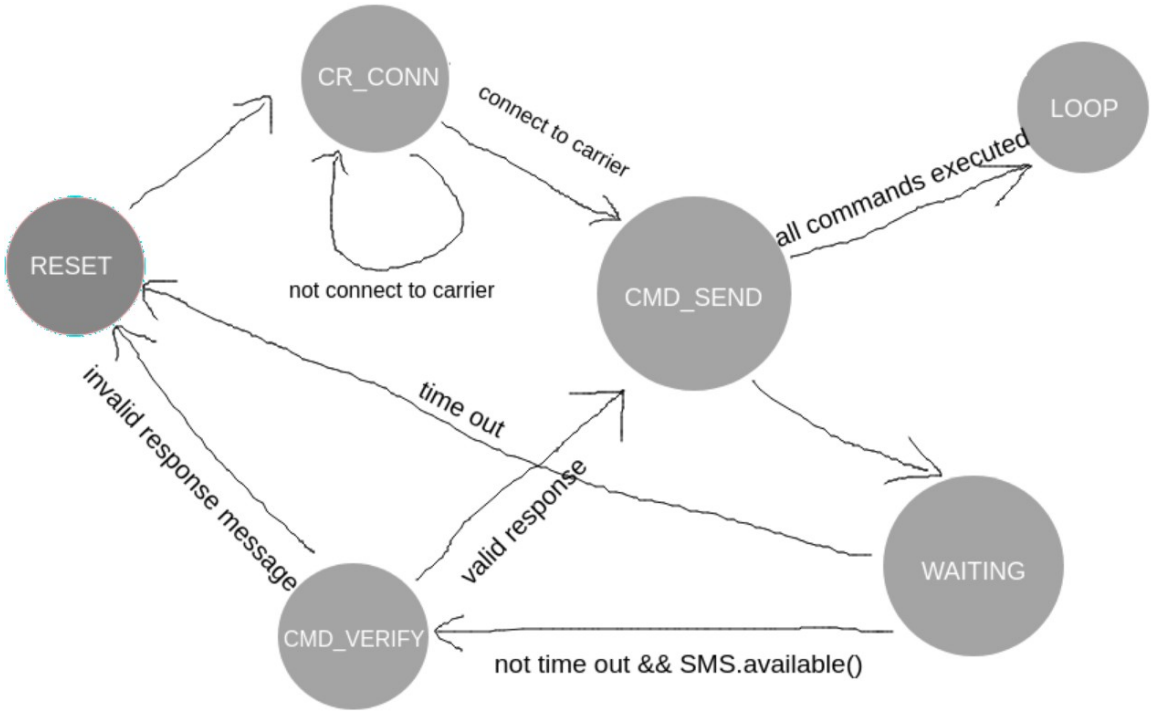
IV. Exercise

1. Implement a program on NodeMCU to communicate with the GSM module using the SoftwareSerial library. The reason is that if the real Serial is used, you don’t have any tool to debug your program. The AT commands sending from NodeMCU to the GSM module is presented in Section II.

```
void setup(){
  Serial.begin(9600);
  while(!Serial);
  SIM.begin(9600);

  resetSIM();
   delay(5000);                          // time needed for the connection's
stablization
```

```
  Serial.println("Reset done");

  GSMCommunicate();
}

void GSMCommunicate(){
  SIM.print("AT+CMGF=1\r");
  Serial.println(_readSerial());  delay(1000);

  SIM.print("AT+CSCS=\"GSM\"\r");
  Serial.println(_readSerial());  delay(1000);

  SIM.print("AT+CMGS=\"");
  SIM.print(number);
  SIM.print("\"\r");              delay(10);
  SIM.print(text);
  SIM.print ("\r");
  SIM.print((char)26);
  Serial.println(_readSerial());  delay(100);

  Serial.println("Sent successfully");
}
```

2. Improve your program by checking the response from GSM module whenever an AT command is sent. If the response is not correct or over a time out (e.g. 5s) and there is no response, the system should come back to initialized state. Design a DFA to illustrate your system and then, implement your program following the DFA.



V. Extra exercise

The GSM module is also support GPRS communication. Using this connection, the module can connect to a webserver and send a request. Following AT commands are used to make a request to a website at http://www.iforce2d.net/test.php:

AT+SAPBR=1,1$0d$0a

AT+HTTPINIT$0d$0a

AT+HTTPPARA="CID",1$0d$0a

AT+HTTPPARA="URL","http://www.iforce2d.net/test.php"$0d$0a

AT+HTTPACTION=0$0d$0a

AT+HTTPREAD$0D$0A

AT+HTTPTERM$0d$0a

Unlike AT commands to send message or make a call, you can continue the whole process even some errors can be occurred.

The AT+HTTPACTION must be delay at least 500ms.

Finally, AT+HTTPTERM is used to close the session, then you can start to request again by looping from AT+HTTPINIT. The first command AT+SAPBR=1,1 is not necessary anymore.

If there is no error, you should receive a response with current date time.

Implement these AT commands in Node MCU. Please note that the response must be checked and there is also a timeout for a response (e.g. 5s).