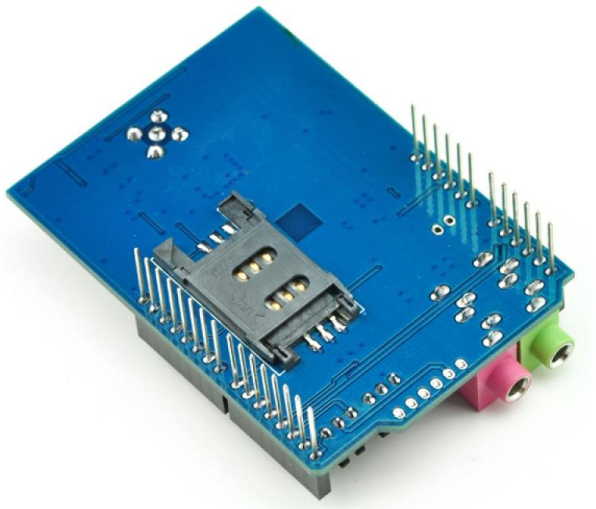
# GPRS Shield

| | |
|---|---|
| **TUJUAN** | A. |
| **DASAR TEORI** |  <br>  <br>GPRS shield menggunakan chip SIM900 quad-band GSM/GPRS. Fitur SIM900, antara lain:<br>o Memiliki 4 tingkat frekuensi jaringan 850/900/1800/1900MHz.<br>o Paket data GPRS kelas 10/8. |

<table>
<tr><td></td><td>
<ul>
<li>Di kontrol dengan AT commands (GSM 07.07, 07.05 dan SIMCOM enhanced AT Commands).</li>
<li>SMS (Short message service)</li>
<li>Power ON/OFF dan fungsi reser di dukung oleh arduino</li>
<li>Embedded TCP/UDP stack</li>
<li>Speaker and Headphone jacks</li>
<li>12 GPIOs, 2 PWMs and an ADC (all 2.8 volt logic)</li>
</ul>
</td></tr>
<tr><td>Praktikum Testing Modul</td><td>
1. The factory default setting untuk GPRS Shield adalah 19200 bps 8-N-1.<br>
2. Hardware Serial - D0/D1 of Arduino/Seeeduino<br>
3. Software serial - D7/D8 of Arduino/Seeeduino.<br>
4. Status LED – menyala jika power SIM900 on<br>
5. Net light - status SIM900 ketika tersambung ke jaringan<br>
6. D9 – digunakan dengan cara software control untuk power up/down SIM900<br>
7. Status LED
<table>
<tr><th>LED</th><th>State</th><th>Function</th></tr>
<tr><td>Status</td><td>Off</td><td>Power Off</td></tr>
<tr><td></td><td>On</td><td>Power On</td></tr>
<tr><td>Netlight</td><td>Off</td><td>SIM900 is not working</td></tr>
<tr><td></td><td>64ms On/800ms Off</td><td>SIM900 does not find the network</td></tr>
<tr><td></td><td>64ms On/3000ms Off</td><td>SIM900 finds the network</td></tr>
<tr><td></td><td>64ms On/300ms Off</td><td>GPRS communication</td></tr>
</table>
8. Power UP/DOWN (Hardware) dengan menekan tombol power 2 detik<br>
9. Power UP/DOWN (Software)dengan mengatur pin D9.<br>
10. Sorce code
<pre>
void powerUpOrDown()
{
  pinMode(9, OUTPUT);
  digitalWrite(9,LOW);
  delay(1000);
  digitalWrite(9,HIGH);
  delay(2000);
  digitalWrite(9,LOW);
  delay(3000);
}
</pre>
11. Testing source code
<pre>
//Serial Relay - Arduino will patch a
//serial link between the computer and the GPRS Shield
//at 19200 bps 8-N-1
//Computer is connected to Hardware UART
//GPRS Shield is connected to the Software UART
</pre>
</td></tr>
</table>

```
#include <SoftwareSerial.h>

SoftwareSerial GPRS(7, 8);
unsigned char buffer[64]; // buffer array for data recieve over serial
port
int count=0;     // counter for buffer array
void setup()
{
   GPRS.begin(19200);         // the GPRS baud rate
   Serial.begin(19200);        // the Serial port of Arduino baud rate.

}

void loop()
{
   if (GPRS.available())              // if date is comming from
softwareserial port ==> data is comming from gprs shield
   {
      while(GPRS.available())         // reading data into char array
      {
         buffer[count++]=GPRS.read();     // writing data into array
         if(count == 64)break;
      }
      Serial.write(buffer,count);
// if no data transmission ends, write buffer to hardware serial port
      clearBufferArray();
// call clearBufferArray function to clear the storaged data from the
array
      count = 0;                     // set counter of while loop to zero

   }
   if (Serial.available())
   GPRS.write(Serial.read());       // write it to the GPRS shield
}
void clearBufferArray()             // function to clear buffer array
{
   for (int i=0; i<count;i++)
   { buffer[i]=NULL;}
 // clear all index of array with command NULL
}
```

12. Testing AT Command
    - AT
    - +CFUN
    - +CPIN
    - Call

<table>
<tr><td></td><td>
<ul><li>AT+IPR=19200</li></ul>
</td></tr>
<tr><td>Praktikum Mengirim SMS</td><td>
1. AT command untuk SMS
<ul>
<li>AT+CMGF=1\r → set GPRS shield ke text mode</li>
<li>AT+CMGS=PHONE_NUMBER → nomer tujuan, international mode</li>
<li>AT+CMGR=ALL\r → membaca SMS dari inbox</li>
</ul>
2. Source Code
<pre>
/*********
  Complete project details at http://randomnerdtutorials.com
*********/

#include <SoftwareSerial.h>

// Configure software serial port
SoftwareSerial SIM900(7, 8);

void setup() {
  // baud rate of 19200
  // Make sure that corresponds to the baud rate of your module
  SIM900.begin(19200);
  // Give time to your GSM shield log on to network
  delay(20000);

  // Send the SMS
  sendSMS();
}

void loop() {
}

void sendSMS() {
  // AT command to set SIM900 to SMS mode
  SIM900.print("AT+CMGF=1\r");
  delay(100);

  /*
REPLACE THE X's WITH THE RECIPIENT'S MOBILE NUMBER
USE INTERNATIONAL FORMAT CODE FOR MOBILE
NUMBERS
  */
  SIM900.println("AT + CMGS = \"+XXXXXXXXXXXX\"");
  delay(100);

  // REPLACE WITH YOUR OWN SMS MESSAGE CONTENT
  SIM900.println("Message example from Arduino Uno.");
  delay(100);
</pre>
</td></tr>
</table>

```
  // End AT command with a ^Z, ASCII code 26
  SIM900.println((char)26);
  delay(100);
  SIM900.println();
  // Give module time to send SMS
  delay(5000);
}
```

**Praktikum Mengirim Data**

1. **AT** return OK, untuk mengecek komunikasi antara sim900 dengan serial komunikasi. apabila return OK maka jalur komunikasi sudah benar.
2. **AT+SAPBR=3,1,"Contype","GPRS",** digunakan untuk setting mode koneksi menjadi GPRS.
3. **AT+SAPBR=1,1,** mengaktifkan GPRS mode.
4. **AT+SAPBR=2,1,** mengaktifkan bearer dari GPRS. Sim900 akan memberikan balasan berupa 1,1,"IP", yang artinya 1 adalah bearer aktif dan selanjutnya adalah ip yang didapatkan.
5. **AT+HTTPINIT**, inisialisasi HTTP service.
6. **AT+HTTPPARA="CID",1**, set HTTP profile identifier.
7. **AT+HTTPPARA="URL","...../getstatus.php?pwabsen=NIM**, setting parameter yang akan dikirim lewat URL kemudian mengakses file dengan nama getstatus.php. file getstatus.php bertugas untuk mengambil nilai "pwabsen=NIM" yang kemudian akan ditulis didatabase dan ditampilkan kembali.
8. **AT+HTTPACTION=0**, digunakan untuk mengirim perintah dengan metode 0=GET sedangkan 1=POST. SIM900 akan memberikan balasan berupa kode "+HTTPACTION=0,200,28" yang artinya 0=metode GET, 200 merupakan kode berhasil mengirimkan request dan 28 merupakan pengembalian sebanyak 28byte. Untuk lebih jelas kode pengembalian bisa dilihat pada datasheet sim900.
9. AT+CIPSHUT :- to close TCP Port Explicitly means disconnect connection if any
10. AT+CGATT? :- Checking SIM card has internet connection or not
11. AT+CSTT = "APN","userName","Pass" :- connect to internet (ex; AT+CSTT="airtelgprs.com","","")
12. AT+CIICR :- bring up with the wireless network. Checking SIM card has data pack or balance
13. AT+CIFSR :- get IP (sometimes without this command GSM do not work so use this command)
14. AT+CIPSTART = ”TCP”,”SERVER IP”,”PORT” :- is used for creating TCP connection with the server that we provide in place of SERVER IP
15. AT+CIPSEND :- this command is used for sending data to the server. After input, this command server asks for data.
16. Source Code

```
#include <SoftwareSerial.h>
```

```
/* Create object named SIM900 of the class SoftwareSerial */
SoftwareSerial SIM900(8, 7);
void setup() {
  SIM900.begin(9600);           /* Define baud rate for software serial
communication */
  Serial.begin(9600);   /* Define baud rate for serial communication */
}

void loop() {
  Serial.println("TCP Send :");
  Serial.print("AT\\r\\n");
  SIM900.println("AT"); /* Check Communication */
  delay(5000);
  ShowSerialData();    /* Print response on the serial monitor */
  delay(5000);
  Serial.print("AT+CIPMODE=0\\r\\n");
  SIM900.println("AT+CIPMODE=0");
/* Non-Transparent (normal) mode for TCP/IP application */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CIPMUX=0\\r\\n");
  SIM900.println("AT+CIPMUX=0");
/* Single TCP/IP connection mode */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CGATT=1\\r\\n");
  SIM900.println("AT+CGATT=1"); /* Attach to GPRS Service */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CREG?\\r\\n");
  SIM900.println("AT+CREG?");      /* Network registration status */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CGATT?\\r\\n");
  SIM900.println("AT+CGATT?");
/* Attached to or detached from GPRS service */
  delay(5000);
  ShowSerialData();
  delay(5000);

 Serial.print("AT+CSTT=\"TATA.DOCOMO.INTERNET\",\"\",\"\"\\r\\n");
```

```
  SIM900.println("AT+CSTT=\"TATA.DOCOMO.INTERNET\",\"\",\"\"");
        /* Start task and set APN */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CIICR\\r\\n");
  SIM900.println("AT+CIICR");
/* Bring up wireless connection with GPRS */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CIFSR\\r\\n");
  SIM900.println("AT+CIFSR");       /* Get local IP address */
  delay(5000);
  ShowSerialData();
  delay(5000);

Serial.print("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"\\r\\n");

SIM900.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");
        /* Start up TCP connection */
  delay(5000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CIPSEND\\r\\n");
  SIM900.println("AT+CIPSEND");
/* Send data through TCP connection */
  delay(2000);
  ShowSerialData();
  delay(2000);
  Serial.print("GET
/update?api_key=C7JFHZY54GLCJY38&field1=1\\r\\n");
  SIM900.print("GET
/update?api_key=C7JFHZY54GLCJY38&field1=1\r\n\x1A");
/* URL for data to be sent to */
  delay(10000);
  ShowSerialData();
  delay(5000);
  Serial.print("AT+CIPSHUT\\r\\n");
  SIM900.println("AT+CIPSHUT");  /* Deactivate GPRS PDP content */
  delay(5000);
  ShowSerialData();
  delay(5000);
}

void ShowSerialData()
```

| | | |
|---|---|---|
| | | {<br>while(SIM900.available()!=0) /* If data is available on serial port */<br>Serial.write(char (SIM900.read())); /* Print character received on to the serial monitor */<br>} |
| **Sumber** | | |