

Willkommen!

Und herzlichen Dank für den Kauf unseres **AZ-Delivery SIM800L GSM GPRS Modul mit Antenne für Arduino!** Die folgenden Seiten zeigen Dir die ersten Schritte, für die Einrichtung dieses Moduls mit einem Arduino.

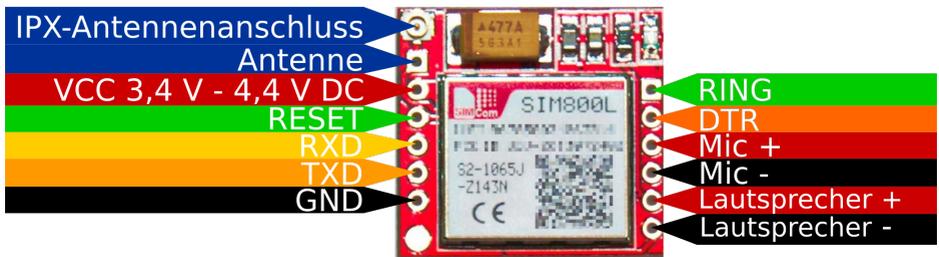


Das **AZ-Delivery SIM800L GSM GPRS Modul** kommt mit einer Stiftleiste, einer Antenne zum Anlöten und einer optionalen Antenne mit IPX-kompatiblen Stecker.

Der SIM800L-Chip kann sich in GSM- und GPRS-Netze einwählen.

Mobilfunkempfang vorausgesetzt, kann das Modul dementsprechend SMS senden und empfangen, Anrufe tätigen und entgegennehmen und sich in das mobile 2G-Internet einwählen.

Durch die variable Eingangsspannung im Bereich von 3,4 V bis 4,4 V DC kannst Du das Modul direkt mit einem Lithium-Ionen-Akku betreiben. Eine direkte Spannungsversorgung vom Arduino 5V-Ausgang ist jedoch nicht ohne Spannungsregulator möglich und, durch die kurzzeitig hohen Ströme von 2 Ampere beim Beginn einer Verbindung, auch nicht ratsam. Bei 3,3 V läuft das Modul nicht stabil.



Die Wichtigsten Informationen in Kürze

- » SIM800L-Chipsatz
- » Micro-SIM Kartenslot
- » externe oder aufgelötete Antenne
- » Anschlussmöglichkeit für einen Lautsprecher und ein Mikrofon
- » Serielle Kommunikation über D0/D1 oder mit der SoftwareSerial Library über viele andere Pins
- » Stromversorgung direkt über einen Lithium-Ionen-Akku oder über Spannungswandler
 - 3,4V bis 4,4V DC

Der SIM800L-Chip benötigt beim Aufbau einer Verbindung bis zu 2A! Achte darauf, dass Dein Akku dies leisten kann. Wenn Dein Akku nicht leistungsstark genug ist, können Verbindungsabbrüche und Beschädigungen am Akku die Folge sein. **Ein beschädigter Lithium-Ionen-Akku birgt Brandgefahr!** Um Spitzenlasten aufzufangen, kann es sinnvoll sein, einen Elektrolytkondensator mit höherer Kapazität zur Spannungsversorgung parallel zu schalten.

Auf den nächsten Seiten findest Du wichtige Links und Informationen zum

- » Aufbau der Schaltung
- » Senden einer SMS
- » Empfangen einer SMS
- » Tätigen eines Anrufes

Diese Anleitung setzt voraus, dass Du Sketche auf Deinen Arduino hochladen kannst, den Bibliotheksverwalter kennst und mit dem Serial-Monitor vertraut bist.

Interessante Links zum Produkt

Arduino Bibliothek:

<https://github.com/erdemarslan/GSMSim>

Übersicht der AT-Befehle für das Modul:

https://simcom.ee/documents/SIM800H/SIM800%20Series_AT%20Command%20Manual_V1.10.pdf

Programmieroberflächen:

Arduino IDE: <https://www.arduino.cc/en/Main/Software>

Web-Editor: <https://create.arduino.cc/editor>

YAT auf sourceforge:

<https://sourceforge.net/p/y-a-terminal/wiki/Home/>

Interessantes von AZ-Delivery

Arduino-kompatible Boards:

<https://az-delivery.de/collections/arduino-kompatible-boards>

Arduino Zubehör:

<https://az-delivery.de/collections/arduino-zubehor>

AZ-Delivery G+ Community:

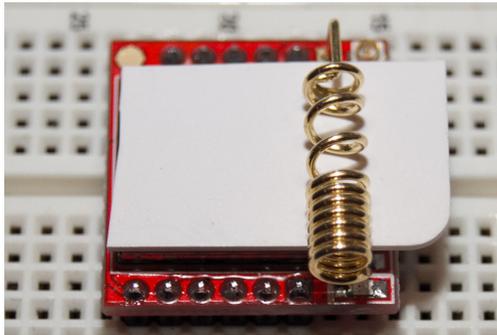
<https://plus.google.com/communities/115110265322509467732>

AZ-Delivery auf Facebook:

<https://www.facebook.com/AZDeliveryShop/>

Einrichtung des Modules

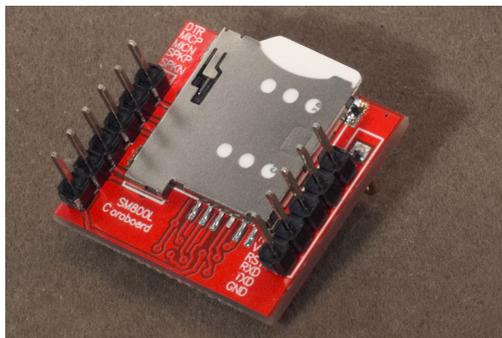
Bevor Du beginnst, musst Du Dich entscheiden, ob du die Ansteckantenne benutzt oder die Antenne zum Einlöten. Solltest Du die Antenne anlöten, benutze unbedingt Flussmittel und Lötzinn mit möglichst niedrigem Schmelzpunkt. **Achte darauf, dass die Antenne keine anderen Bauteile berührt.** Ein Stück Pappe oder Plastik als Abstandhalter beim Löten vereinfacht das Erstellen der Lötverbindung. **Eine schlechte Lötverbindung stört den Empfang!**



Die externe Antenne eignet sich besonders, wenn das Modul in einem Gehäuse aus Metall eingebaut wird. **Solltest Du eine externe Antenne benutzen, muss die Lötstelle für die interne Antenne frei bleiben, um Interferenzen zu vermeiden.**

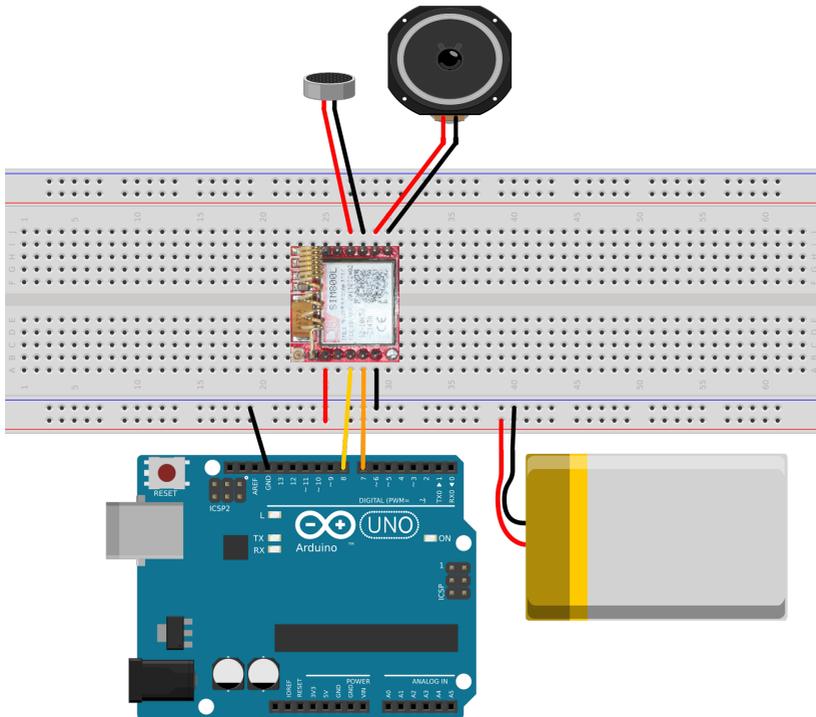
Du kannst die mitgelieferten Stiftleisten verwenden oder direkt Kabel anlöten.

Auf den SIM-Kartensockel geschaut, wird die Micro-SIM mit der Kerbe nach **oben links gerichtet** eingeschoben, bis der Federmechanismus einrastet.



Aufbau der Schaltung

Stelle zum Anfang **alle in der Skizze dargestellten Verbindungen** her. Du kannst die RX- und TX-Pins auch auf andere Pins des Arduino legen, wenn Du dies in Deinem Programm richtig zuordnest. (Die Stromversorgung des Arduino, soll erstmal über den USB-Anschluss erfolgen)



Benötigst Du die Telefonfunktion nicht, kannst Du Mikrofon und Lautsprecher weglassen.

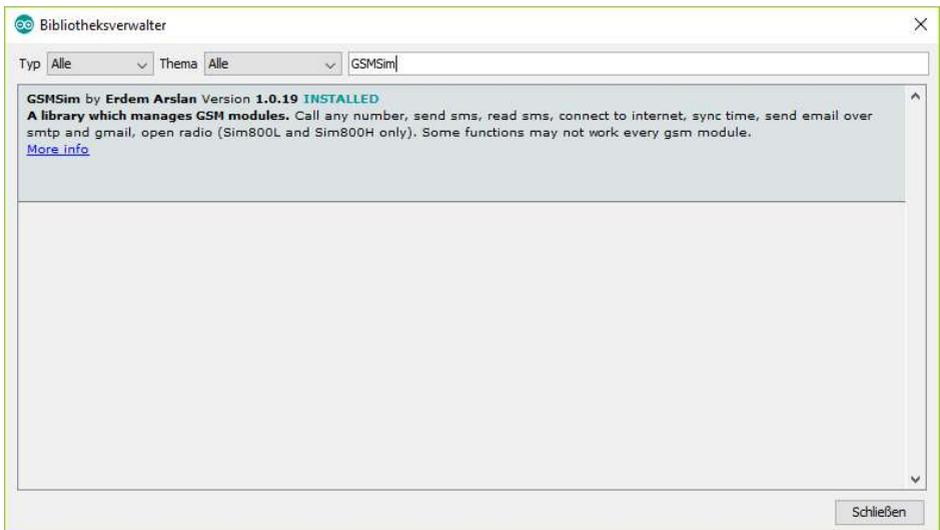
Später kann es durchaus sinnvoll sein, den Reset-Pin des Moduls auch anzuschließen.

Einrichtung in der Arduino IDE

Prinzipiell benötigt das SIM800L-Modul keine Bibliothek, es kann über serielle Kommunikation direkt mit AT-Befehlen gesteuert werden. Eine Übersicht der möglichen AT-Befehle findest Du unter: https://simcom.ee/documents/SIM800H/SIM800%20Series_AT%20Command%20Manual_V1.10.pdf

Das Ansteuern des Moduls mithilfe einer Bibliothek ist jedoch viel komfortabler.

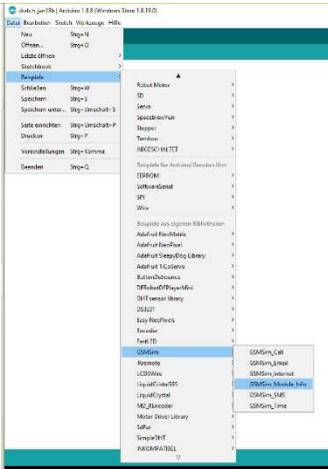
Installiere zunächst die **GSMSim** Bibliothek von Erdem Arslan über den Bibliotheksverwalter. Auf Github gibt es unter dem Link <https://github.com/erdemarslan/GSMSim> eine gute Dokumentation in englischer Sprache zu der Bibliothek.



Alternativ kannst Du GSMSim auch direkt bei Github herunterladen und in das libraries-Verzeichnis deines Sketchbook-Ordners entpacken.

Bei Problemen prüfe außerdem, ob die SoftwareSerial-Bibliothek installiert ist, da diese benötigt wird. (Bei aktuellen Versionen der Arduino IDE ist die SoftwareSerial-Bibliothek standardmäßig installiert)

Funktionsprüfung



Wenn Du alles richtig verbunden hast, kannst Du das Beispiel `GSM_Sim_Module_Info` aufrufen. Gehe dafür in einem neuen Arduinosketch auf: Datei > Beispiele > GSM_Sim > `GSM_Sim_Module_Info`

Da Du den Reset-Pin des Moduls noch nicht verbunden hast, müsst Du zuerst zwei kleine Änderungen am Sketch vornehmen. Zuerst entferne die Zeile:

```
#define RESET 2
```

Dann ändere den Bibliotheksaufruf von

```
GSM_Sim gsm(RX, TX, RESET);
```

auf:

```
GSM_Sim gsm(RX, TX);
```

Um es etwas bequemer zu machen, kannst Du nach den Zeilen:

```
void setup() {  
  // put your setup code here, to run once:  
  
  Serial.begin(9600);
```

Noch folgenden Code eingeben:

```
while (!Serial){ }
```

Damit wartet der Arduino dann, bis Du den seriellen Monitor geöffnet hast, bevor er mit dem Programm fortfährt.

Nachdem Du das Program auf den Arduino geladen hast und Du den seriellen Monitor gestartet hast, solltest Du nach kurzer Zeit dort folgendes sehn:



The screenshot shows the 'COM7 (Arduino/Genuino Uno)' serial monitor window. The title bar includes standard window controls and a 'Senden' button. The main content area displays 'GSMsim Library - Module Information' with the following data:

```
Manufacturer: SIMCOM Ltd
Model: SIMCOM SIM800L
Revision: 1418B05SIM800L24
IMEI: 8688813406214
IMSI: 71457774018767
ICCID: 89430030044800011
Is Connected?: 1
Signal Quality: 23
Operator: 中国移动
Operator From Sim: Things Mobile
```

At the bottom, there are checkboxes for 'Autoscroll' (checked) and 'Zeitstempel anzeigen' (unchecked). On the right, there are dropdown menus for 'Kein Zeilenende' and '9600 Baud', and an 'Ausgabe löschen' button.

(Wenn Du den Wartebefehl nicht zugefügt hast, drücke nach dem öffnen des seriellen Monitors die RESET-Taste des Arduino)

Bitte überprüfe folgende Daten bevor Du weiter machst:

1. Die IMEI sollte mit der IMEI auf deinem Modul übereinstimmen.
2. Hinter „Is Connected:“ sollte eine 1 für true stehen
3. Der Wert von „Signal Quality“ sollte zwischen 1 und 31 liegen (ein Wert von 0 oder 99 deutet auf einen Fehler der Antenne hin)
4. Operator From Sim sollte Deinem Anbieter entsprechen

Sollten keine oder falsche Werte in der zweiten Reihe der Tabelle stehen, kann der Arduino keine gute Verbindung zum Modul aufbauen. Überprüfe dann Deine Verkabelung. Du kannst auch

```
gsm.start(BAUD);
```

in

```
gsm.start();
```

ändern, dann handeln Modul und Arduino eine Baudrate aus.

Wenn hinter „Is Connected:“ eine 0 für false steht, überprüfe ob Deine SIM-Karte ein **PIN** hat, denn diese **muss deaktiviert sein**. (Standart bei ThingsMobile SIM-Karten)

Sollte das Modul trotz guter Signalstärke weiterhin keine Verbindung aufbauen, **überprüfe ob die SIM beim Anbieter aktiviert ist**.

Ein Sketch erstellen

Fange zunächst mit einem leeren Sketch an. Zuerst musst Du die Bibliotheken einbinden und die Pins festlegen. Die folgenden Codezeilen zeigen, wie es geht. (Du kannst die Kommentare natürlich weglassen)

```
//Einbinden der benötigten Bibliotheken:
#include <GSMSim.h>
#include <SoftwareSerial.h>

#define RX 7 //festlegen des RX-Pin am Arduino
#define TX 8 //festlegen des TX-Pin am Arduino
//#define RESET 2 //festlegen des RESET-Pin am Arduino

GSMSim gsm(RX, TX); //Ruft die Bibliothek auf
/*
 * andere Möglichkeiten sind:
 * GSMSim gsm; //RX=D7|TX=D8|Reset=D2|LEDpin=D13 als Standart
 * GSMSim gsm(RX, TX, RESET);
 * GSMSim gsm(RX, TX, RESET, LED_PIN, LED_FLAG);
 */
```

Später kannst Du auch den REST-Pin festlegen. Noch ist die Definition hierfür auskommentiert. Solltest du andere Pins für RX und TX verwenden wollen achte darauf, dass TX mit RX verbunden wird und RX mit TX.

Die erste SMS senden

Um eine SMS zu senden, füge die folgenden Codezeilen deinem Sketch hinzu:

```
void setup() {
  gsm.start(); //Startet die Funktionen der Bibliothek

  gsm.smsTextMode(true); //aktiviert den SMS modus des Moduls

  char* Nummer = "+49170123456789"; //gibt die Nummer an
  char* Nachricht = "Hello World!"; //legt die Nachricht fest

  //Senden der Nachricht an die angegebene Nummer:
  gsm.smsSend(Nummer, Nachricht);
}
void loop() {
  // put your main code here, to run repeatedly:
}
```

Anschließend musst Du das fertige Sketch nur noch auf Deinen Arduino hochladen. Mit jedem Neustart sendet Dein Arduino nun eine SMS an die angegebene Nummer.

Eine empfangene SMS lesen

Das Modul speichert eingegangene Nachrichten auf der SIM-Karte. Je nach Anbieter können dies 10 bis 50 SMS sein. Die folgenden Befehle listen die ungelesene SMS auf und zeigen die erste SMS im Speicher über den seriellen Monitor an.

```
void setup() {
  gsm.start(); //Startet die Funktionen der Bibliothek
  Serial.begin(9600); //Startet die Verbindung zum PC
  while (!Serial){ //auf den Seriellen Monitor warten
  }
  //gibt eine Liste mit ungelesene SMS aus:
  Serial.println("Ungelesene Nachrichten Auflisten:");
  Serial.println(gsm.smsListUnread());

  //Gibt die erste SMS im Speicher aus:
  Serial.println("Die erste SMS im Speicher lautet:");
  Serial.println(gsm.smsRead(1));
  /* ist der Speicher leer, lautet die Ausgabe *INDEX_NO_ERROR
  */
}
```

Einen Anruf tätigen

Um einen Anruf zu tätigen, sollten Mikrofon und Lautsprecher verbunden sein. Die Telefonnummer wird auch hier in einer Variablen des Typs `char` festgelegt. Zum Beispiel so:

```
char* Nummer = "+49170123456789";
```

Der Anruf wird mit

```
gsm.call(Nummer);
```

gestartet.

Beendet wird der Anruf mit dem Befehl:

```
gsm.callHangoff();
```

Nützliche Tipps zum Modul

In der Bibliothek enthalten sind weitere Beispiele zum Umgang mit dem **AZ-Delivery SIM800L GSM GPRS Modul**. Dort findest Du Möglichkeiten, wie Du Daten vom Modul abrufst, eine Internetverbindung aufbaust oder eine E-Mail sendest.

Solltest Du Probleme mit Deinem Modul haben, prüfe immer zuerst, ob alle Verbindungen ohne Wackelkontakt sind und ob die **Eingangsspannung im Bereich von 3,4 V bis 4,4 V DC** ist. Steckbretter sind besonders anfällig für Wackelkontakte. Serielle-Kommunikation kann dann schwierig sein.

Zum Experimentieren kannst Du ein USB-Netzteil und ein DC-DC Step Down Modul verwenden, solltest Du keinen passenden Akku zur Verfügung haben. Ein passendes Modul von AZ-Delivery ist das **LM2596S DC-DC Step Down Modul**. Du benötigst dann jedoch auch ein Multimeter, um die Spannung am Modul einzustellen.

Um Deinen Li-Ion Akku zu laden, kannst Du den **AZ-Delivery Laderegler TP4056 Micro-USB** verwenden.

Mit einem mit einem **FTDI Adapter** und einem Terminalprogramm wie YAT, kannst Du AT-Befehle, auch ohne Arduino, vom PC aus an das Modul senden. Die Bit-Rate des Moduls in YAT ist 115200.

Einen leicht zu installierenden **FTDI Adapter** gibt es auch bei AZ-Delivery, solltest Du das **SIM800L GSM GPRS Modul** direkt am PC, mithilfe eines Terminals, benutzen wollen.

Für Experimente am PC gibt es auch drei Möglichkeiten einen Arduino UNO als FTDI Adapter zu verwenden. Diese sind:

1. Den Rest Pin des Arduino UNO mit GND verbinden
2. Den Atmel Chip zu entfernen
3. Folgendes Sketch auf den Arduino UNO zu laden:

```
void setup(){  
  pinMode(0, INPUT);  
  pinMode(1, INPUT);  
}  
void loop(){ }
```

Der RX-Pin des Arduino ist dann D0, der TX-Pin ist D1.

Herzlichen Glückwunsch!

Du hast dieses Tutorial erfolgreich durchlaufen. Jetzt heißt es für Dich: Versuch macht klug. Schau Dir die verwendeten Codes an, um die einzelnen Befehle besser kennenzulernen. Ein Blick in die Dokumentation der GSMSim Bibliothek auf GitHub kann auch hilfreich sein. Hier findest Du viele weitere Funktionen des SIM800L-Moduls. Solltest Du auch die Radiofunktion ausprobieren, beachte, dass das Modul keine Antenne dafür hat.

Wenn Du ein anderes GSM/GPRS-Modul oder weitere tolle Hardware brauchst, dann schau in unserem Online-Shop auf:

<https://az-delivery.de>

Wir wünschen dir viel Spaß mit diesem tollen Produkt!

Impressum

<https://az-delivery.de/pages/about-us>