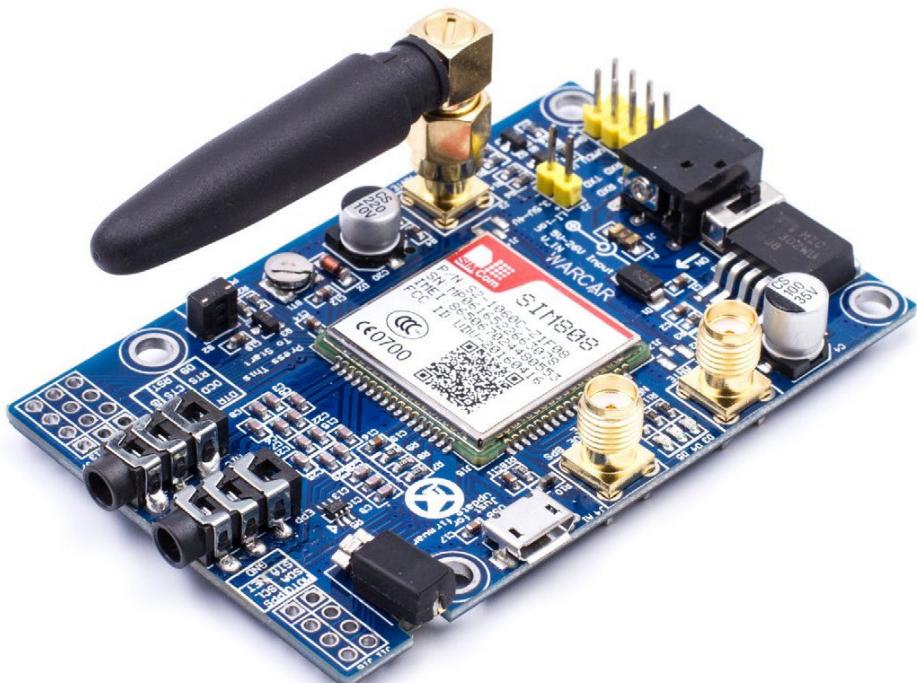


Willkommen!

Und herzlichen Dank für den Kauf unseres **AZ-Delivery SIM-808-Boards mit GSM, GPRS und GPS!** Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zurersten SMS und der ersten GPS-Ortung. Viel Spaß!



<http://flyt.it/SIM808>

Das **AZ-Delivery SIM-808-Board** kommt mit je einer externen Antenne für GSM / GPRS und GPS. Im Gegensatz zum **AZ-Delivery GPRS-Shield** kann es nicht auf einen Arduino UNO aufgesteckt werden, lässt sich aber ebenso leicht mit nur drei Kabeln verbinden.

Die Stromversorgung erfolgt am besten über ein spannungsstabilisiertes 5V-Netzteil mit mindestens 10W Leistung!

Die wichtigsten Informationen in Kürze

- » SIM-808-Chipsatz
- » Standard-SIM-Kartenslot
- » externe Antennen für GSM/GPRS und GPS
- » separate 3,5 mm Klinkenanschlüsse für Mikrofon und Kopfhörer
- » Serielle Kommunikation über Hard- und SoftwareSerial
- » Stromversorgung mit 5V über externen 5/2,5 mm Anschluss (Schalter nach innen zum Einschalten)

Der SIM-808-Chip benötigt teilweise Stromstärken bis zu 2A, die der Arduino nicht leisten kann. Funktionsabbrüche und Beschädigungen am Arduino können die Folge sein. Es empfiehlt sich daher, immer ein spannungsstabilisiertes 5V / 10W-Netz- teil zur externen Versorgung des Shields zu verwenden!

Auf den nächsten Seiten findest du Informationen zum

- » *Aufbau der Schaltung*

und eine Anleitung für

- » *das Senden einer SMS und*
- » *die Abfrage deiner GPS-Koordinaten.*

Diese Anleitung setzt voraus, dass du weißt, wie du Sketche auf einen Arduino hochlädst und den Serial Monitor verwendest!

Alle Links im Überblick

Arduino-Bibliothek:

- » <https://github.com/MarcoMartines/GSM-GPRS-GPS-Shield>

SIM808 Hardware-Dokumentationen:

- » <http://simcom.ee/documents/?dir=SIM808>

Programmieroberflächen:

- » Arduino IDE: <https://www.arduino.cc/en/Main/Software>
- » Web-Editor: <https://create.arduino.cc/editor>
- » Arduino-Erweiterung für SublimeText:
<https://github.com/Robot-Will/Stino>
- » Arduino-Erweiterung "Visual Micro" für Atmel Studio
oder Microsoft Visual Studio:
<http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx>

Interessantes von AZ-Delivery

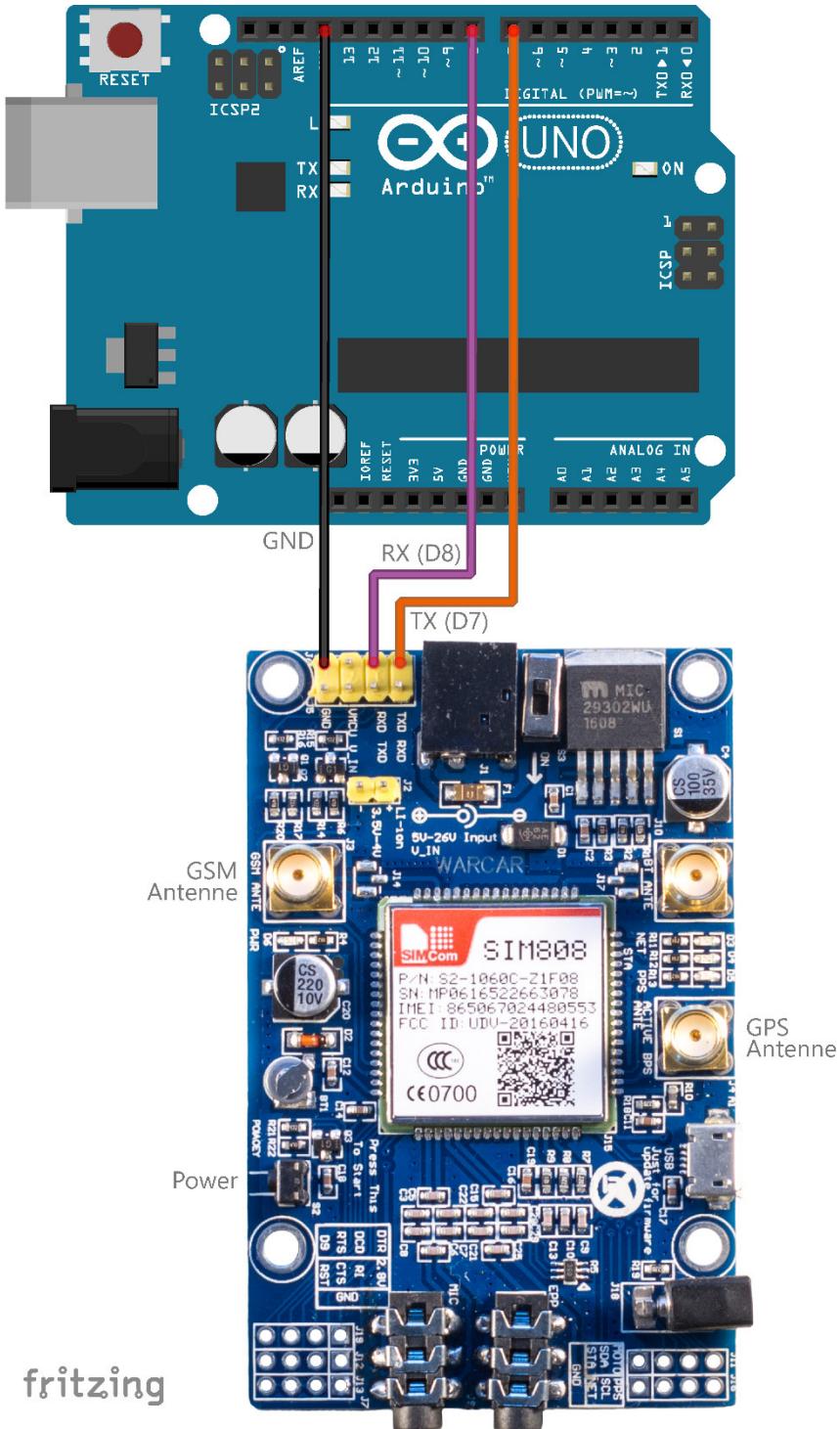
- » Arduino-kompatible Boards:
<https://az-delivery.de/collections/arduino-kompatible-boards>
- » Arduino Zubehör:
<https://az-delivery.de/collections/arduino-zubehor>
- » AZ-Delivery G+Community:
<https://plus.google.com/communities/115110265322509467732>
- » AZ-Delivery auf Facebook:
<https://www.facebook.com/AZDeliveryShop/>

Aufbau der Schaltung

Für die serielle Kommunikation zwischen Board und Microcontroller sind nur drei Verbindungen nötig. Der **TX**-Pin des einen Gerätes wird mit dem **RX**-Pin des anderen verbunden wie auch **RX** mit **TX**. Da Board und Controller unterschiedliche Spannungsquellen besitzen, müssen zudem noch die Massen (**GND**) angeglichen werden.

RX und **TX** liegen beim UNO auf den Pins "**D0**" und "**D1**". Diese "**Hardware Serial**" genannte Verbindung wird allerdings auch für den Serial Monitor der Arduino IDE verwendet. Allerdings ist es möglich, über eine Bibliothek eine sogenannte "**Software Serial**" zu emulieren. Da die hier verwendete Bibliothek auch für das **AZ-Delivery SIM-900-Shield** verwendet wird, passen wir die Pin-Belegung so an, dass sie für beide Module funktioniert: "**D7**" für **TX** und "**D8**" für **RX**.

Wie eingangs erwähnt sollte das Board über ein **spannungsstabilisiertes 5V-Netzteil** mit mindestens **10W Leistung** versorgt werden. Beim Versenden von Signalen kann es kurzzeitig zu einer erhöhten Leistungsaufnahme kommen, die billige Netzteile oft nicht bereitstellen können. Eine Versorgung durch den Arduino ist aus diesem Grund ebenfalls nicht zu empfehlen, da dieser im schlimmsten Fall sogar beschädigt werden kann.



fritzing

Der letzte fehlende Schritt ist das Einsetzen der SIM-Karte. Den Slot findest du auf der Unterseite des Boards. **Vergewissere dich aber vorher, dass deine Karte keinen PIN verlangt!** Ansonsten deaktiviere ihn über die Sicherheitseinstellungen eines Mobiltelefons.

Für den Fall, dass du nur eine Micro- oder Nano-SIM-Karte besitzt, klemme sie in den jeweiligen Sim-Kartenadapter aus dem Set, das dir mit deinem SIM-808 geliefert wurde. Durch den Einschub im Deckel des Kartenslots werden auch solche eher losen Verbindungen stabil auf die Kontakte des Shields gedrückt.

Installation der Bibliothek für das SIM808-Board

Bibliotheken für das **SIM-808** findet man im Vergleich zum **SIM-900**-Chipsatz recht wenige. Da aber beide auf den gleichen AT-Befehlssatz hören, sind diese oft mit beiden kompatibel. Eine der funktional umfangreichsten ist die von dem Entwickler Marco Martines. Lade sie hierüber herunter:

» <https://github.com/MarcoMartines/GSM-GPRS-GPS-Shield/archive/GSMSHIELD.zip>

Entpacke anschließend den Ordner "**GSM-GPRS-GPS-Shield-GSMSHIELD**" in das **libraries**-Verzeichnis deines Sketchbook-Ordners. Dort kannst du den Namen der Lesbarkeit halber einkürzen, z.B. in "**GSMSHIELD**". Anschließend musst du noch die Pin-Belegung für die Software-Serial anpassen. Öffne in dem Verzeichnis die Datei "**GSM.cpp**" und ändere in den Zeilen **27** und **28** die Werte zu:

```
25 //De-comment this two lines below if you have the
26 //second version og GSM GPRS Shield
27 #define _GSM_TXPIN_ 7
28 #define _GSM_RXPIN_ 8
```

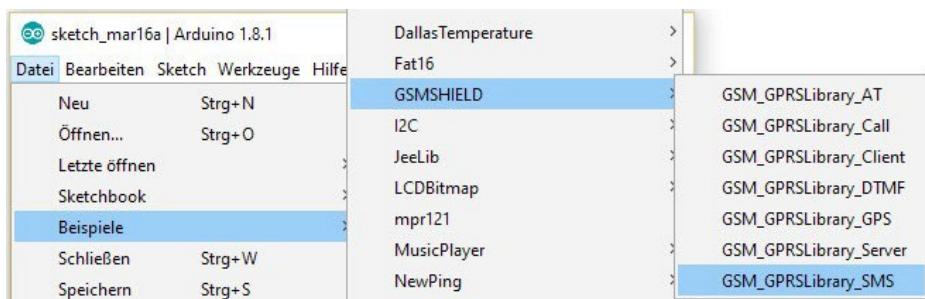
Um für spätere Experimente gerüstet zu sein, definiere auch den Start- und Reset-Pin in der Datei "**GSM.h**" wie folgt:

```
45 #define GSM_ON 9 // connect GSM Module turn ON to pin 77
46 #define GSM_RESET 6 // connect GSM Module RESET to pin 35
```

Danach schließe alle eventuell noch offenen Instanzen deiner Arduino IDE und starte das Programm erneut. Nun solltest du von der Bibliothek unter Anderem ihre mitgelieferten Beispiele finden können.

Die erste SMS

Das klassische "Hello World" soll uns auch bei der ersten SMS unseres Arduinos begrüßen. Starte dazu den Beispiel-Sketch "GSM_GPRS Library SMS".



Damit aber überhaupt etwas bei dir ankommt, musst du zunächst den Befehl zum Senden der SMS aktivieren. Entferne dafür die Kommentarzeichen aus den Zeilen **40** und **41** und ändere gleich auch die Zielnummer sowie bei Bedarf den Text der Mitteilung:

```
38     if(started) {  
39         //Enable this two lines if you want to send an SMS.  
40         if (sms.SendSMS("+49123456789", "Hello World!"))  
41             Serial.println("\nSMS sent OK");  
42     }
```

Lade den Code anschließend auf deinen UNO.

Aktiviere nun das **SIM-808-Board**, indem du es an das Netzteil anschließt, den Schalter daneben nach innen schiebst und den

Taster auf der Seite der GSM-Antenne so lange gedrückt hältst, bis auf der gegenüberliegenden Seite zwei rote LEDs aufleuchten. Das schnelle Blinken der einen bedeutet, dass das Board versucht, sich ins Mobilfunknetz einzuwählen. Sinkt die Blinkfrequenz auf ein kurzes Aufleuchten aller drei Sekunden, ist die Verbindung hergestellt und das Board bereit.

Starte nun den Serial Monitor mit einer Baud-Rate von **9600**.

GSM Shield testing.

ATT: OK

RIC: OK

Mit diesen Zeilen beginnt die Testphase. Ist diese vorbei, sollte sich dein Handy auch schon mit einer neuen SMS bei dir melden: "**Hello World!**"

Der Serial Monitor stellt derweil die Meldung für die erfolgreich gesendete SMS wie folgt dar:

The screenshot shows the Arduino Serial Monitor window titled "COM6". The window has a "Senden" button at the top right. The text area contains the following log:

```
status=READY
DEBUG:SMS TEST
ATT: >
RIC:
OK

>
DEBUG:>
ATT: +CMGS
RIC:
+CMGS: 8

OK

SMS sent OK
```

At the bottom, there are three settings: "Autoscroll" (unchecked), "Kein Zeilenende" (selected), and "9600 Baud".

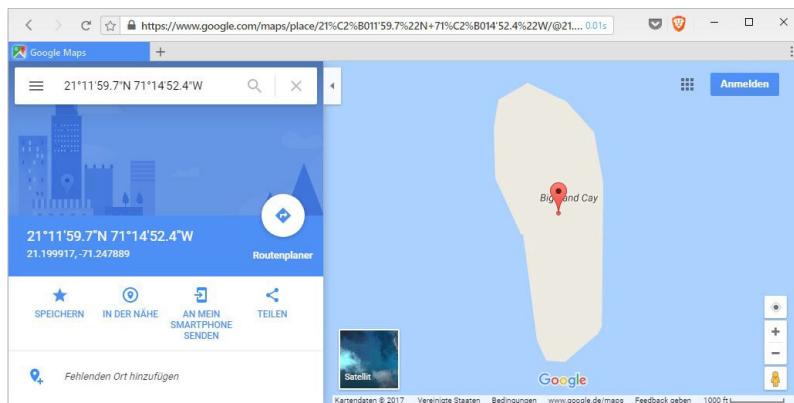
GPS-Koordinaten auslesen

Das Erfassen deiner Position ist fast genauso einfach. Starte dazu aus dem gleichen Verzeichnis den Beispiel-Sketch "**GSM_GPRS-Library_GPS**". Hier musst du nichts ändern und kannst den Code direkt auf den UNO laden. Starte anschließend den Serial Monitor und lass das Programm durchlaufen, bis du folgende letzte Zeilen siehst (Werte werden abweichen):

```
2111.597000  
-7114.524000  
73.600000  
20170722120902.000  
0.074080
```

Die ersten beiden Zeilen sind die Angaben für den Breiten- und für den Längengrad. Diese sind so leider nicht interpretierbar. Wenn du diese Angaben beispielsweise bei Google Maps überprüfen möchtest, schreibe sie wie folgt um:

```
2111.597000 in 21°11'59.70"  
-7114.524000 in -71°14'52.40" also:
```



Herzlichen Glückwunsch!

Du hast das Tutorial erfolgreich durchlaufen, deine erste SMS mit einem Arduino verschickt und auch noch herausgefunden, von wo! Ab jetzt heißt es Lernen und Ausprobieren. Schau dir am besten den verwendeten Code genauer an, um herauszufinden, wie die Sende- und Lese-Befehle eingesetzt werden. Ein Blick in die anderen Beispiele der Bibliothek lohnt sich ebenfalls.

Wenn du dein SIM-808-Board mit einem weiteren kommunizieren lassen möchtest oder einfach andere tolle Hardware suchst, wirst du wie immer fündig bei deinem Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!

Impressum

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

