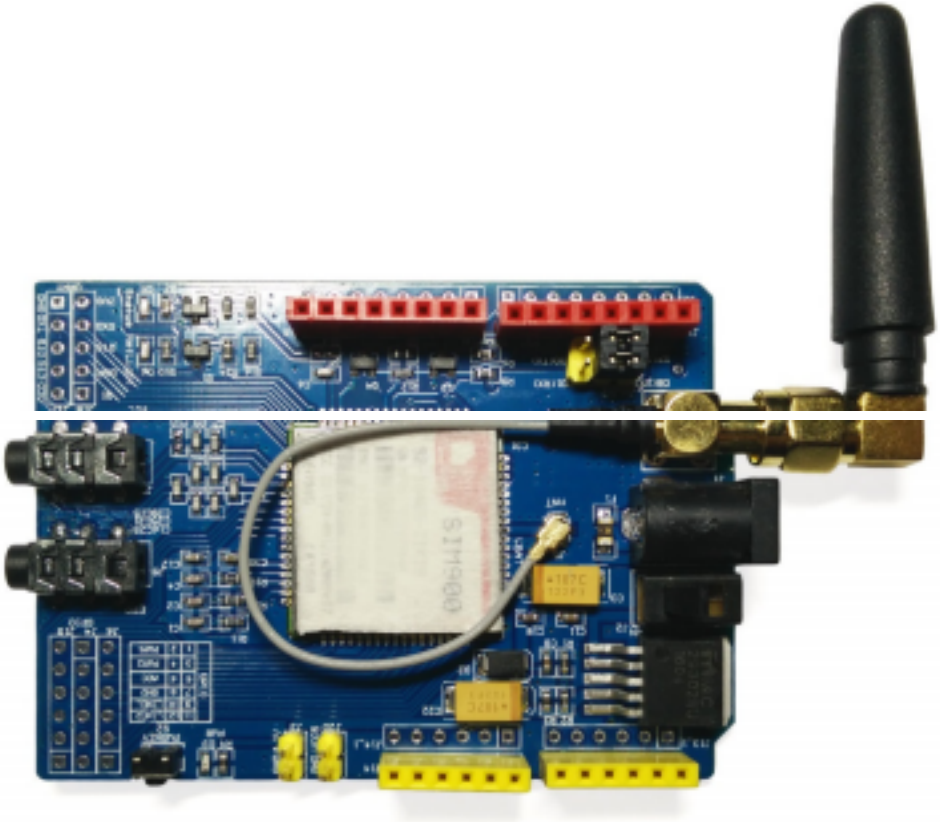
AZ-Delivery

Ihr Experte für Mikroelektronik!



http://flyt.it/GPRS-Shield

Willkommen!

Und herzlichen Dank für den Kauf unseres **AZ Delivery GPRS Shields**! Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zur ersten SMS. Viel Spaß!

Das **AZ-Delivery GPRS Shield** kommt mit einer externen Antenne und ist mit dem beliebten **SIM900**-Chip ausgestattet, der sich in das GSM- und GPRS-Netz einwählen kann. Das Pin-Layout ist mit dem Atmega328p kompatibel, kann also mittels optionaler Pin Leisten direkt aufgesetzt werden. Zur Kommunikation sind allerdings maximal drei Kabel nötig. Das Shield kann direkt über den Atmega328p mit Strom versorgt werden, doch wird eine externe 5V-Stromversorgung empfohlen!

Die wichtigsten Informationen in Kürze

» SIM900-Chipsatz
» Standard-SIM-Kartenslot
» externe Antenne
» separate 3,5 mm Klinkenanschlüsse für Mikrofon und Kopfhörer
» Serielle Kommunikation über D0/D1 oder D7/D8
» Stromversorgung
» 5V über Atmega328p (Schalter nach außen)
» 5V über externen 5/2,5 mm Anschluss (Schalter nach innen)

Der SIM900-Chip benötigt teilweise Stromstärken von bis zu 2A, die der Atmega328p nicht leisten kann. Funktionsabbrüche und Beschädigungen am Atmega328p können die Folge sein. Es empfiehlt sich daher, immer ein spannungsstabilisiertes 5V / 10W-Netzteil 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*.

Diese Anleitung setzt voraus, dass du weißt, wie du Sketche auf einen Atmega328p hochlädst und den Serial Monitor verwendest!

Alle Links im Überblick

Bibliothek:
» *https://github.com/MarcoMartines/GSM-GPRS-GPS-Shield*

SIM900 Pinout-Schema:
» *https://www.open-electronics.org/gsm-remote-control-part-4- sim900/*

Programmieroberflächen:
» Arduino IDE:
https://www.arduino.cc/en/Main/Software »

Web-Editor: *https://create.arduino.cc/editor*

» Erweiterung für SublimeText:
https://github.com/Robot-Will/Stino

» Erweiterung "Visual Micro" für Atmel Studio
oder Microsoft Visual Studio:
http://www.visualmicro.com/page/Arduino-for-Atmel Studio.aspx

Interessantes von AZ-Delivery

» Atmega328p-kompatible Boards:
https://az-delivery.de/collections/arduino-kompatible-boards

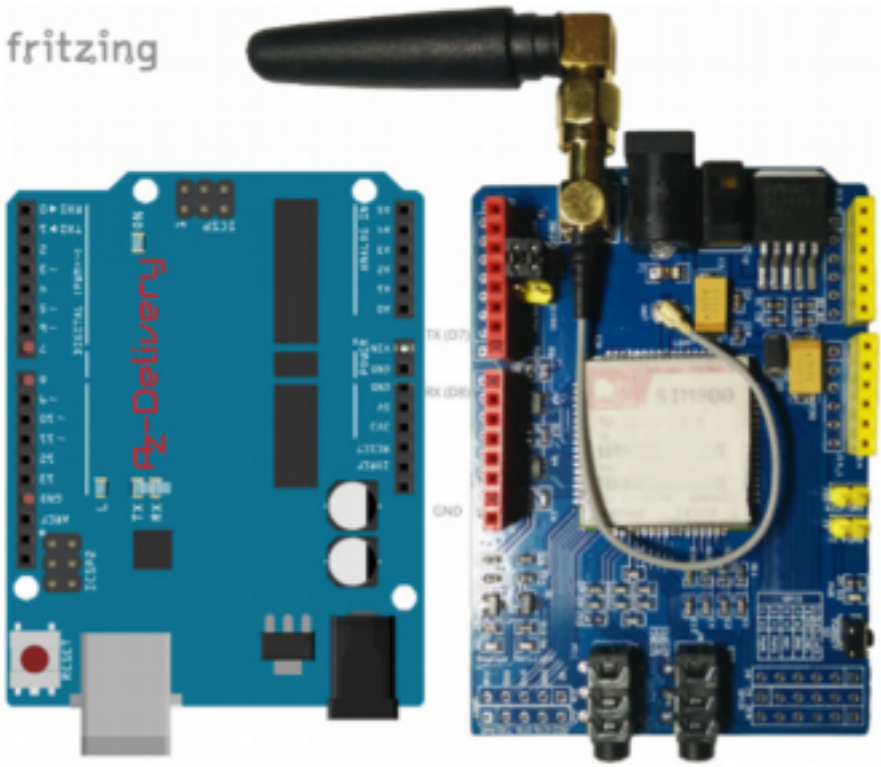
» Zubehör:
https://www.az-delivery.de/collections/weiteres-zubehoer

» AZ-Delivery G+Community:
https://plus.google.com/communities/115110265322509467732 » AZ-Delivery auf Facebook:
https://www.facebook.com/AZDeliveryShop/

Aufbau der Schaltung

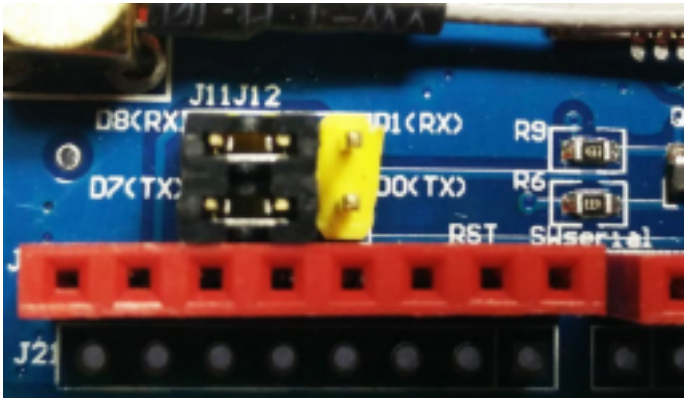
Für die serielle Kommunikation zwischen Shield und Microkontroller sind nur zwei Verbindungen nötig. Der **TX**-Pin des einen Gerätes wird mit dem **RX**-Pin des anderen verbunden wie auch **RX** mit **TX**. Diese Überkreuzschaltung ist bei dem Shield allerdings bereits gegeben.

Die dritte Verbindung dient nur der **Masse**-Angleichung beider Module, sollte sicherheitshalber aber dennoch gelegt werden.



RX und **TX** liegen beim Atmega328p 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, bei der für viele Shield-Layouts die Pins "**D7**" und "**D8**" zur Konvention geworden sind.

Über die Jumper an den roten Pin-Leisten kann zwischen Hard und Software-Serial gewechselt werden. Für das Tutorial verwenden wir letztere.



Wie eingangs erwähnt sollte das Shield über ein spannungsstabilisiertes 5V-Netzteil mit mindestens 10W Leistung versorgt werden. Bei den möglichen Stromspitzen und der Speisung über den Atmega328p unterbricht dieser möglicherweise
seinen Programm- durchlauf und kann im schlimmsten Fall beschädigt werden.

Zum Wechsel auf die externe Versorgung stellst du den Schalter neben der 5/2,5 mm-Buchse auf die Innenposition.

Der letzte fehlende Schritt ist das Einsetzen der SIM-Karte. Den Slot findest du auf der Unterseite des Shields. **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 Rahmen, die dir mit deinem Shield geliefert worden sind. 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 GPRS Shield

Bibliotheken gibt es für GSM / GPRS-Module mit dem **SIM900**-Chipsatz viele im Internet. Eine der funktional umfangreichsten ist die von *gsmlib.org*, welche von dem Entwickler Marco Martines noch einmal aktualisiert wurde. Lade sie hierüber herunter:

» *https://github.com/MarcoMartines/GSM-GPRS-GPS-Shield/ar chive/GSMSHIELD.zip*

Entpacke anschließend den Ordner "**GSM-GPRS-GPS-Shield-GS MSHIELD**" 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:

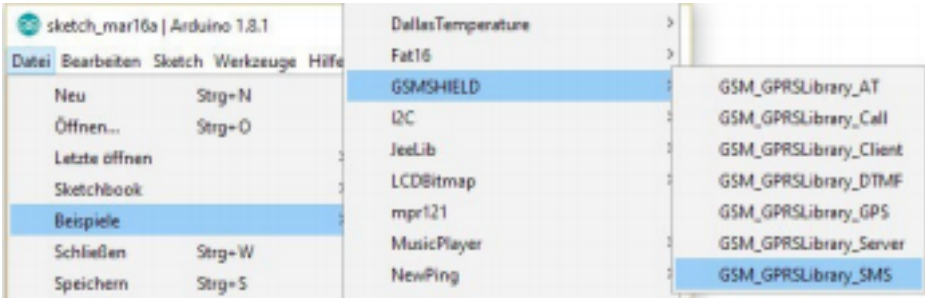
```
45  #define GSM_ON 9 // connect GSM Module turn ON to pin 77
46  #define GSM_RESET 6 // connect GSM Module RESET to pin 35
```

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 Mikrocontrollers begrüßen. Starte dazu den Beispiel-Sketch "**GSM_GPRSLibrary_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 Atmega328p.

Aktiviere nun das GPRS-Shield. Wenn es bereits mit Strom versorgt wird, leuchtet die rote LED neben dem **PWRKEY**-Taster. Letzteren drückst du mindestens eine halbe Sekunde lang, bis auf der gegenüberliegenden Seite zwei ebenfalls rote LEDs

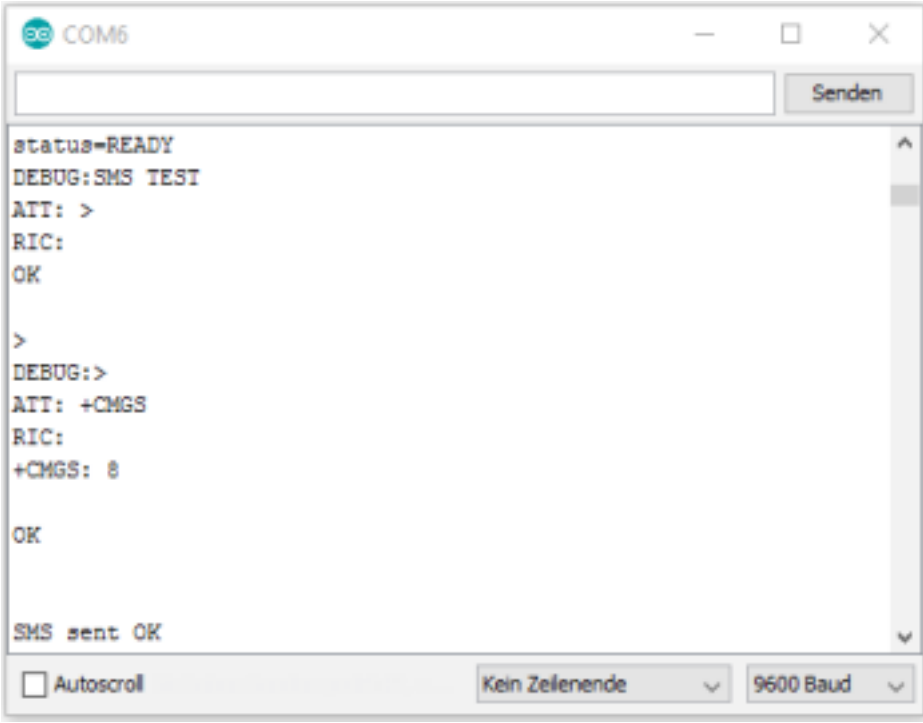
aufleuchten.

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:



Herzlichen Glückwunsch!

Du hast das Tutorial erfolgreich durchlaufen und deine erste SMS mit einem Atmega328p verschickt. 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. Beachte dabei aber, dass der SIM900-Chip nicht über GPS verfügt und die Sketche dafür nicht funktionieren werden.

Wenn Sie ein weiteres GPRS Shield oder andere tolle Hardware suchen, werden Sie wie immer fündig bei dem Online-Shop auf:

https://az-delivery.de

Viel Spaß!
Impressum

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