

## Welcome!

And thank you very much for purchasing our **AZ-Delivery GPRS Shield**! On the following pages we will go together through the first steps from the establishing until the first SMS. Have fun!



<http://flyt.it/GPRS-Shield>

The **AZ-Delivery GPRS Shield** comes with an external antenna and is equipped with the popular **SIM900** chip, which can dial into the GSM and GPRS network. The pin layout is compatible with Arduino UNO and can be attached directly to it with optional pin headers. However just up to three cables are needed to enable the communication between these devices.

The shield can be powered directly from the Arduino, but an external 5V power supply is recommended!

## The most important information in brief

- » SIM900 chipset
- » standard SIM card slot
- » external antenna
- » separate 3.5 mm jacks for microphone and headphones
- » serial communication via D0/D1 or D7/D8
- » **power supply**
  - » 5V via Arduino (switch outside)
  - » 5V external 5/2.5 mm connector (switch inside)

The SIM900 chip partially requires currents of up to 2A, which the Arduino cannot afford. Function errors and damage to the Arduino can be the result. Therefore it is advisable to use always a stabilized voltage 5V / 10W-Netzteil to the external supply of the shields!

On the following pages you will find information to the

» *construction of the circuit*

and a guide to

» *send an SMS.*

This guide assumes that you know how to upload sketches to an Arduino and how to use the serial monitor!

## All links at a glance

### Arduino library:

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

### SIM900 pinout scheme:

- » <https://www.open-electronics.org/gsm-remote-control-part-4-sim900/>

### Programming interfaces:

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

### Interesting stuff by AZ-Delivery

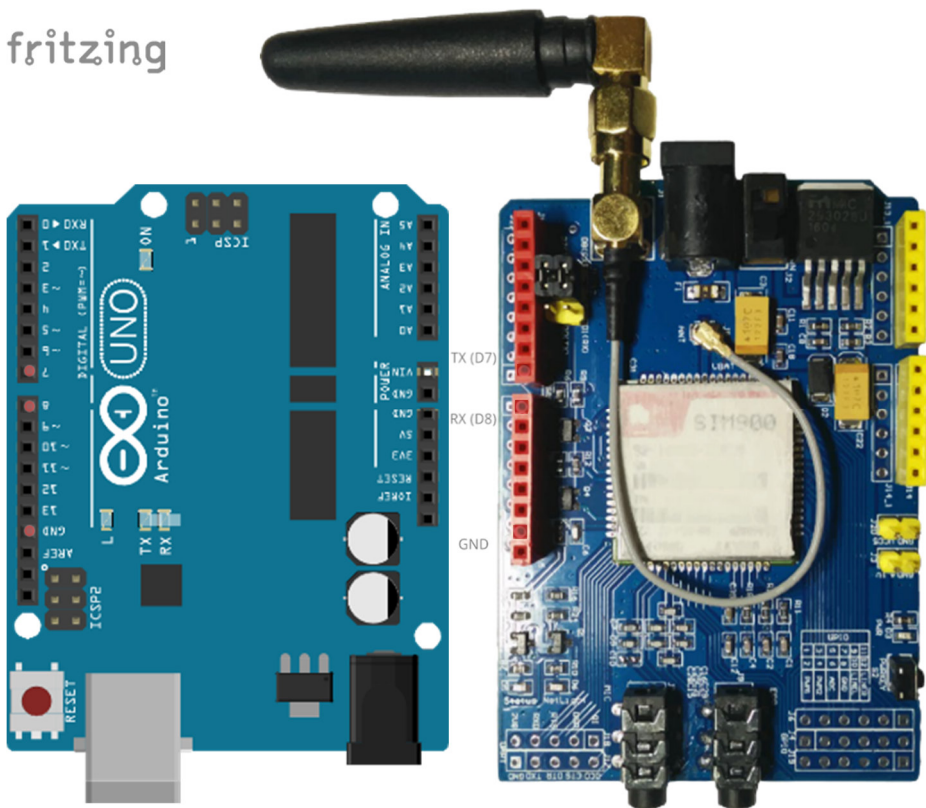
- » Arduino compatible boards:  
<https://az-delivery.de/collections/arduino-kompatible-boards>
- » Arduino accessories:  
<https://az-delivery.de/collections/arduino-zubehor>
- » AZ-Delivery G+Community:  
<https://plus.google.com/communities/115110265322509467732>
- » AZ-Delivery on Facebook:  
<https://www.facebook.com/AZDeliveryShop/>

## Construction of the circuit

Only two connections are needed for serial communication between shield and microcontroller. The **TX** pin of one device is connected to the other's **RX** pin as also **RX** with **TX**. This cross circuit is already given in the shield.

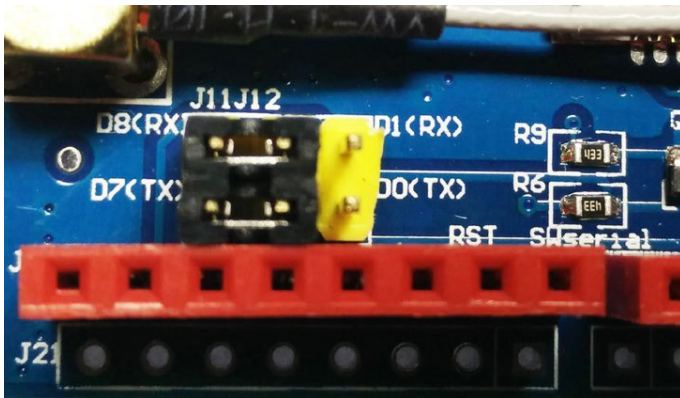
The third connection is used only to equate the grounds of both modules, but nevertheless it should be placed.

fritzing



At the UNO you can find **RX** and **TX** on the pins "**D0**" and "**D1**". This connection, known as "**Hardware Serial**", is used for the Serial Monitor of the Arduino IDE. However, using a library it is possible to emulate a so-called "**Software Serial**", for which many shield layouts the pins "**D7**" and "**D8**" have become a convention.

Via the jumpers at the red pin strips, you can switch between hardware and software serial. For this tutorial, we use the latter.



As mentioned at the beginning the shield should be supplied by a voltage-stabilized **5V** adaptor with at least **10W** power. The possible current peaks and the supply from the Arduino might interrupts its program sequence and can be damaged in the worst case.

To switch to the external supply, put the switch next to the **5/2.5 mm** connector on the inside position.

The last missing step is inserting the SIM card. You can find the slot on the bottom of the shield. But first, make sure that your card doesn't require a PIN! Otherwise disable it by using the security settings of a mobile phone.

If you only have a micro or nano SIM card, clip them into the appropriate frame that has been supplied with your shield. Due to the insert in the cover of the card slot, such rather loose connections are pressed stably on the contacts of the shield, too.

## Installation of the library for the GPRS shield

There are many libraries for GSM / GPRS modules based on the **SIM900** chipset on the Internet. One of the functionally most extensive ones comes from [gsm.lib.org](http://gsm.lib.org), developed by Marco Martines. Download it here:

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

Then unpack the folder "**GSM-GPRS-GPS-Shield-GSMSHIELD**" into the libraries directory of your sketchbook folder. There you can cut off the name for a better readability, e. g. to "**GSMSHIELD**". Now you must adjust the pinout for the software serial. Open the file "**GSM.cpp**" in that folder and change the values in the rows **27** and **28** to:

```
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
```

To be prepared for further experiments, also define the start and reset pin in the file "**GSM.h**" like that:

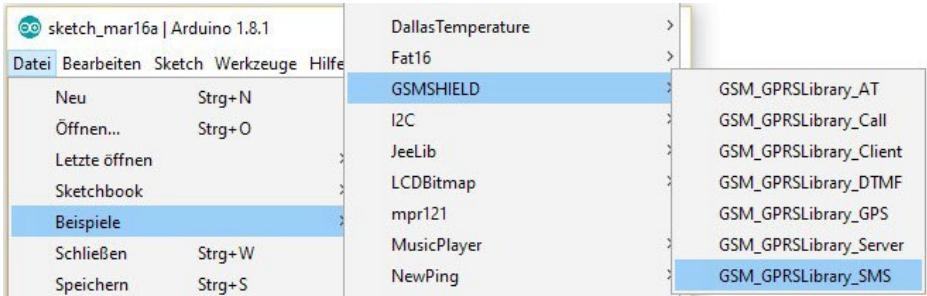
```
45 #define GSM_ON 9 // connect GSM Module turn ON to pin 77
46 #define GSM_RESET 6 // connect GSM Module RESET to pin 35
```

Then close all possibly remaining instances of your Arduino IDE and restart the application. You should now find among other things its included examples from the library.



## The first SMS

The classic "Hello World" should welcome us at the first SMS of our Arduino, too. Therefor run the example sketch "GSM\_GPRSLibrary\_SMS".



In order to receive anything at all, you have to activate the command for send the SMS. Therefor remove the comment signs from lines **40** and **41** and change the target number as well as the text of the message, if desired:

```
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     }
```

Then upload the code to your UNO.

Now activate the GPRS shield. If it is already powered, the red LED next to the **PWRKEY** button lights. Press that button for at least half a second, until two opposite red LEDs light up on the opposite side.

The fast flashing of one of the two LEDs means that the shield tries to dial into the mobile network. If the flashing frequency drops down to a brief flashing every three seconds, the connection is established and the shield is ready.

Now start the serial monitor with a baud rate of 9600.

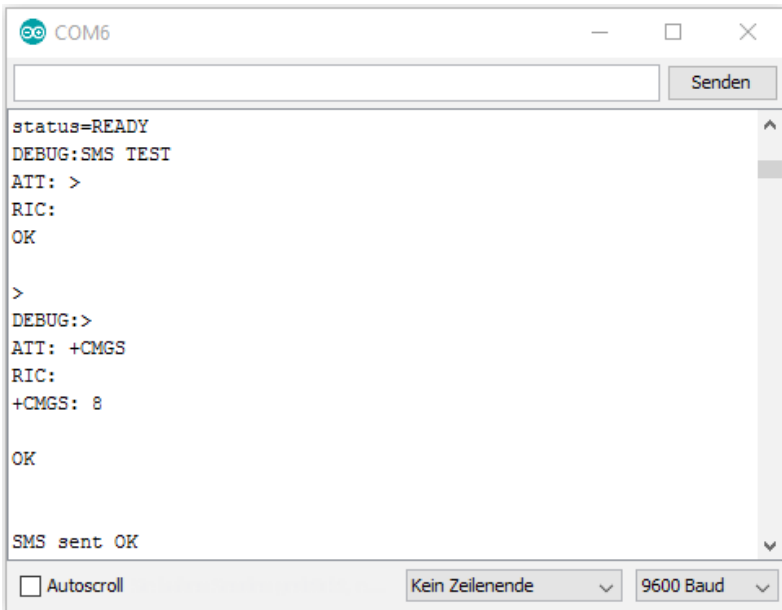
```
GSM Shield testing.
```

```
ATT: OK
```

```
RIC: OK
```

With these lines the test phase starts. If this is over, your mobile phone should have already signed up with a new SMS:  
**"Hello World!"**

The serial monitor now displays the message of the successfully sent SMS as seen on the right side.



## **Congratulations!**

You have successfully completed the tutorial and sent your first SMS with an Arduino. From now on you have to learn and test yourself. Look more closely at the code we have used to find out how to use the send and read commands. A look at the other examples of the library is also worthwhile. Note, however, that the SIM900 chip does not have GPS and the sketches for this function will not work.

If you're looking for a second GPRS shield or other great hardware, you will always find it in your online shop on:

*<https://az-delivery.de>*

Have fun!

**Impressum**

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