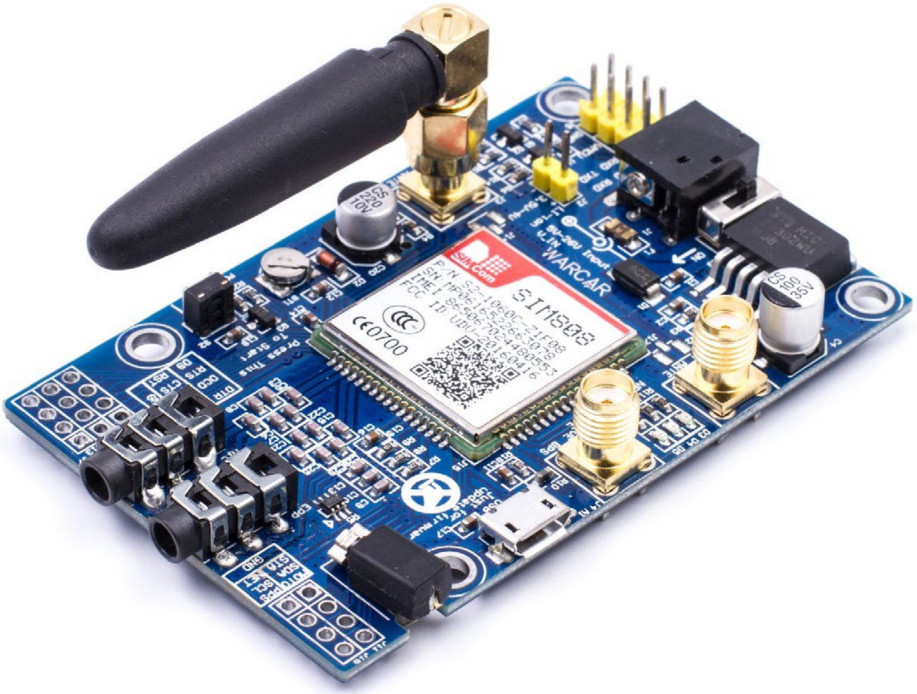
Welcome!

And thank you for purchasing our **AZ-Delivery SIM-808-Boards with GSM, GPRS and GPS**! On the following pages, we will take you through the first steps of the installation process to the first SMS and the first GPS tracking. We wish you a lot of fun!



http://flyt.it/SIM808

The **AZ-Delivery SIM-808-Board** comes with one external antenna for GSM / GPRS and GPS. Unlike the **AZ-Delivery GPRS shield,** it cannot be plugged into an microcontroller, but it can be connected to only with three cables.
The electricity supply is best provided by a 5V voltage-stabilizer with at least 10W power!

Overview of the most important information

» SIM-808 chipset
» Standard SIM card slot
» External antennas for GSM/GPRS and GPS
» Separate 3,5 mm jack connector for microphone and headphones
» Serial communication via hard- and software serial
» Power supply with 5V via external 5/2,5 mm connection (to turn on, press switch inwards)

The SIM-808 chip sometimes requires a current with strength of up to 2A, which the microcontroller cannot afford or manage. The result can be, functional breaks and damages of the microcontroller. It is therefore advised to always use a voltage stabilized 5V / 10W-power supply, as an external care application of the shield.

On the following pages, you will find information about
» *Structure of the circuit*
And a guide for
» *sending an SMS* and
» *the query of your GPS coordinates*.

It is assumed by this tutorial, that you are familiar with uploading sketches to an microcontroller and you know how to use the Serial Monitor!

Overview of all Links

library:

» *https://github.com/DFRobot/DFRobot_SIM808*

SIM808 hardware documentation:

» *http://simcom.ee/documents/?dir=SIM808*

Application programming interfaces:

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

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

» Extension for SublimeText:
https://github.com/Robot-Will/Stino

» Extension "Visual Micro" for Atmel Studio or
Microsoft Visual Studio:
http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx

Interesting information from AZ-Delivery

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

» Accessories:
https://az-delivery.de/collections/arduino-zubehor

» AZ-Delivery G+Community:
https://plus.google.com/communities/115110265322509467732

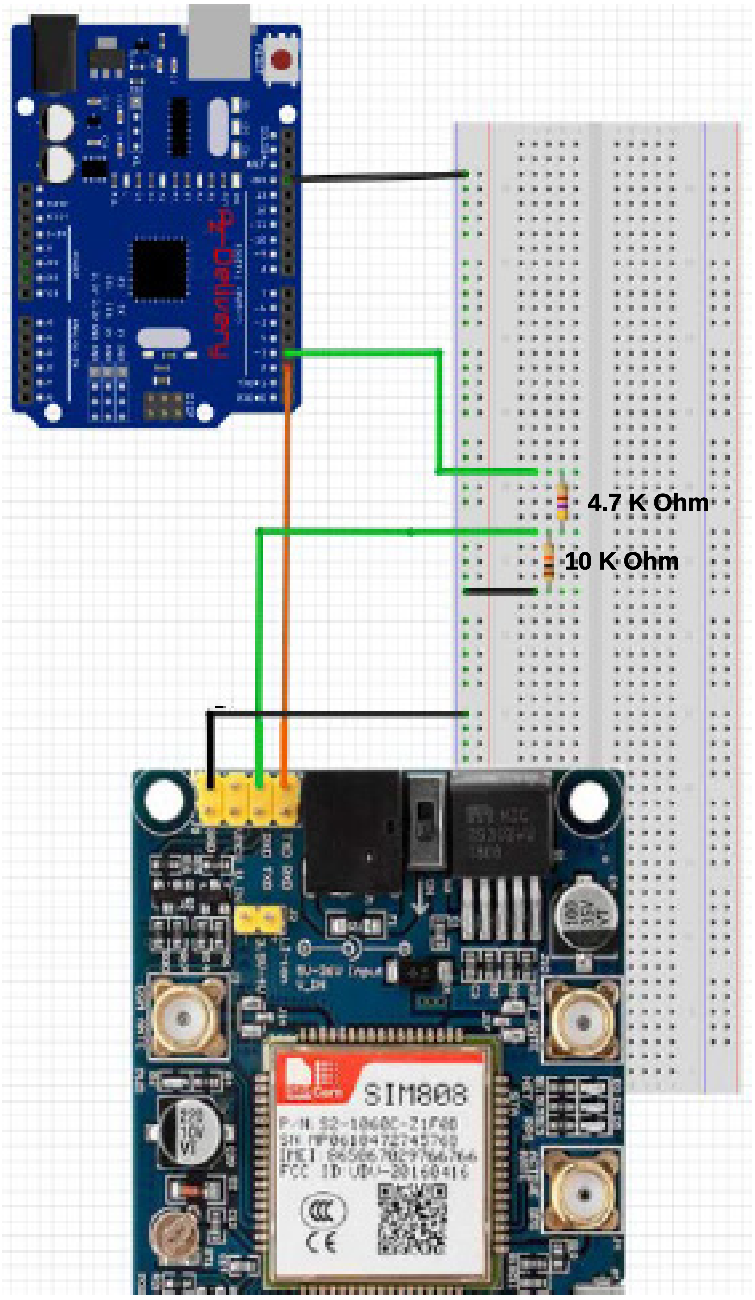
» AZ-Delivery on Facebook:
https://www.facebook.com/AZDeliveryShop/

Structure of the circuit

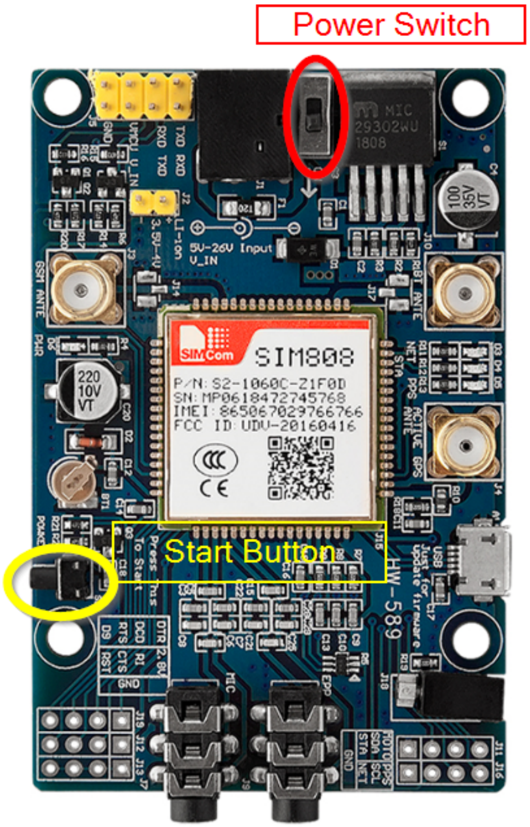
For the serial communication between the board and the microcontroller, only three connections are necessary. The **TX**-pin of one device is connected to the **RX**-pin of the other, like the **RX**-to the **TX** pin. Since the board and the controller have different voltage sources, the masses **(GND)** must also be adjusted. It should be mentioned that the SIM808 module has a logic level of 3.3V, because the microcontroller works with 5V we have to use a voltage divider (see page 6). Alternatively you can also use a logic level converter, which is available in our store.

RX and **TX** are on the UNO’s "**D0**" and "**D1**" pins. This connection, namely "**Hardware Serial**", is also used for the serial monitor of the microcontroller IDE. It is possible, nevertheless, to emulate a so-called "**Software Serial**" via a library. Since the library used here, is also used for the **AZ-Delivery SIM-900-shield**, we should adjust the pin assignment to function for both modules: "**D2**" for **TX** and "**D3**" for **RX**.

As previously mentioned, the board should be powered by a **voltage stabilized 5V power supply unit** with at least **10W power**. When sending signals, a brief increase in the power consumption may occur, which cheap power supply units will not be able to handle. It is also not recommendable to get power supply from the microcontroller as, in the worst case scenario, this could lead to its damage.


4.7 K Ohm
10 K Ohm
SIM808

The last step is to insert the SIM card. The slot for insertion can be found on the bottom side of the board. **But first, make sure that your card does not require a PIN!** Otherwise, you will have to deactivate it via the security settings of a mobile phone. Make sure that the power switch is set to ON. To start the module you have to press the Start button manually (hold to turn it off).



Installing the library for the SIM808-Board

There are fewer libraries for the **SIM-808** compared to the **SIM-900** chipsets. But since both of them listen to the same AT command set, they are both compatible with one another. You can download it here:

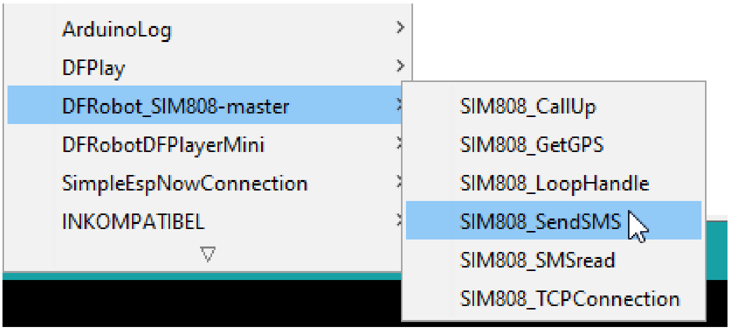
https://github.com/DFRobot/DFRobot_SIM808

Unzip the folder "DFRobot_SIM808-master" into the libraries directory of your Sketchbook folder.

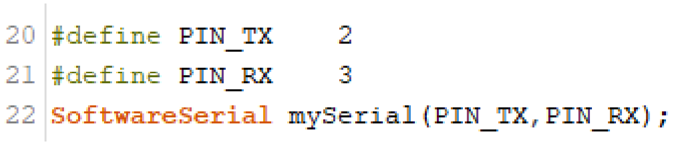
Then close all instances of your Arduino IDE and restart the program. Now you should be able to find the examples provided by the library.

The first SMS

The classic "Hello World" should be included in the first SMS of our microcontroller. Start the example sketch "**SIM808_SendSMS**".

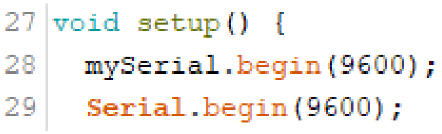


Afterwards you have to adjust the pin assignment for the software serial. To do this, lines 20 and 22 must be adjusted as follows:



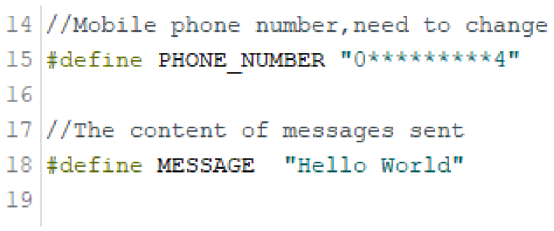
```
20 #define PIN_TX     2
21 #define PIN_RX     3
22 SoftwareSerial mySerial(PIN_TX,PIN_RX);
```

Line 25 must be commented out. Row 28 must be adjusted as follows:



```
27 void setup() {
28   mySerial.begin(9600);
29   Serial.begin(9600);
```

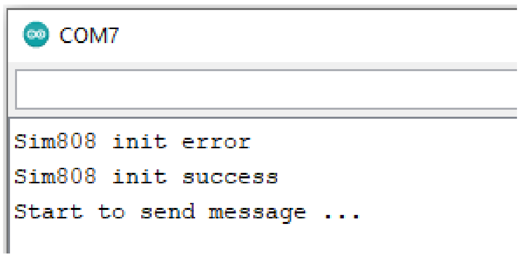
Befor running, you must first enter the destination number and the message. Enter the number in line 15 and change the message to be sent by SMS in line 18. In this example it is called "Hello World".



```
14 //Mobile phone number,need to change
15 #define PHONE_NUMBER "0*********4"
16
17 //The content of messages sent
18 #define MESSAGE  "Hello World"
19
```

Now you can upload the code to your microcontroller. The GSM module can now be started by pressing the Start button until two red LED's on the opposite side light up.

Now start the Serial Monitor with a baud rate of 9600.

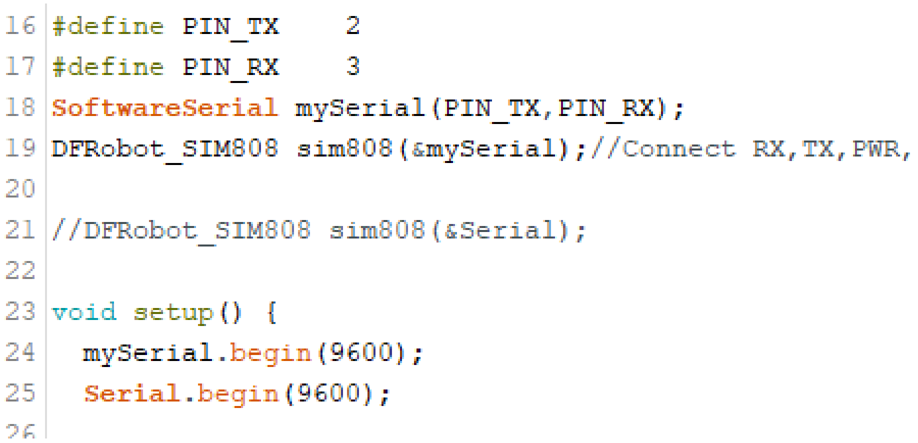


```
Sim808 init error
Sim808 init success
Start to send message ...
```

Now the sent SMS should arrive on the target number

Read GPS coordinates

Capturing your position is almost as easy. To do this, start the example sketch from the same directory "SIM808_getGPS". Here you have to make the same changes as in the SMS example, the sketch should look like this:



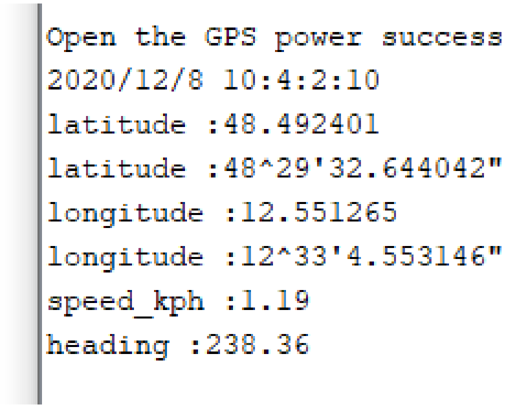
```
#define PIN_TX    2
#define PIN_RX    3
SoftwareSerial mySerial(PIN_TX,PIN_RX);
DFRobot_SIM808 sim808(&mySerial);//Connect RX,TX,PWR,

//DFRobot_SIM808 sim808(&Serial);

void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);
```

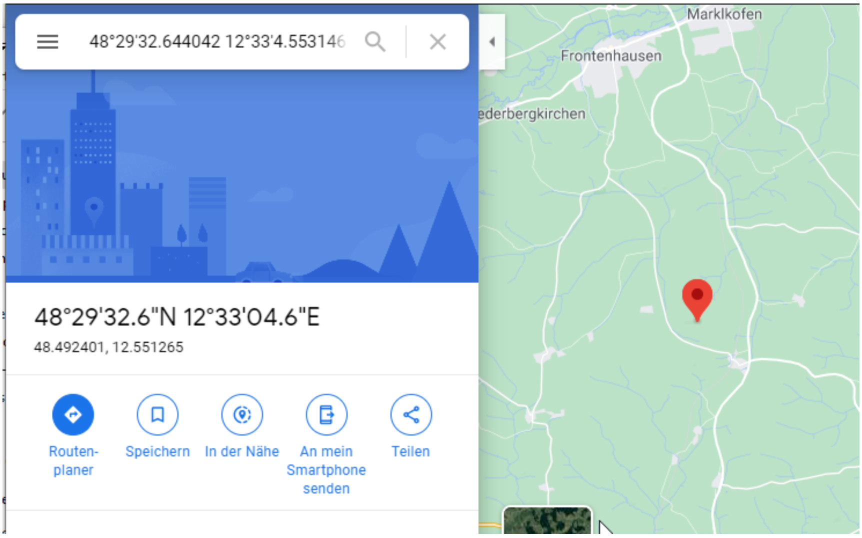
The example can then be uploaded to the microcontroller.

After uploading we open the serial monitor and should get a result of this form:



```
Open the GPS power success
2020/12/8 10:4:2:10
latitude :48.492401
latitude :48^29'32.644042"
longitude :12.551265
longitude :12^33'4.553146"
speed_kph :1.19
heading :238.36
```

These coordinates can be entered into Google Maps to get the GPS position. The " ^ " characters should be replaced by " ° ".



Congratulations!

You have successfully completed this tutorial, sent your first SMS with an microcontroller, and also found out from where the SMS was sent! Now it is time to learn and practice. It is best advised to look at the code you are using, to find out and know how to use the *send* and *read* commands. It is also worthwhile to look at the other examples that are in the library.
If you want your SIM-808-Board to be able to communicate with another, or you simply want to browse for other hardware, you will always find it in your online store at:

https://az-delivery.de

Enjoy!