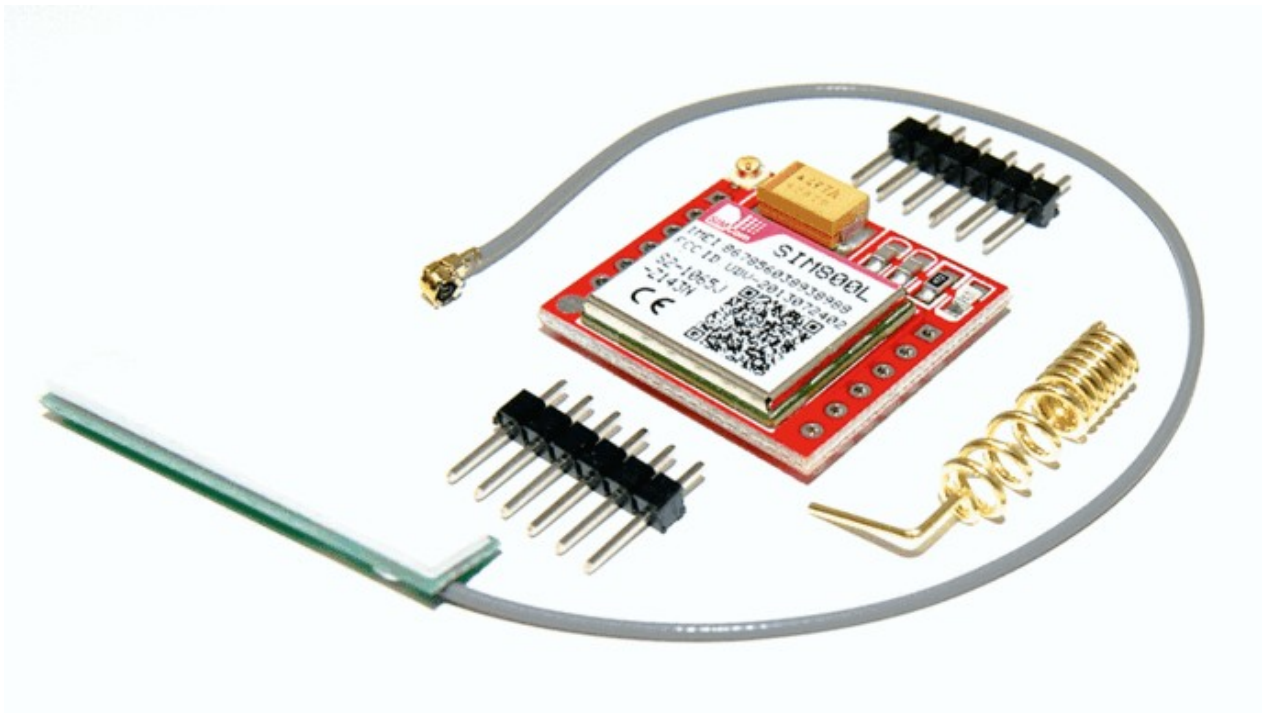


AZ-Delivery

Welcome!

Thank you very much for purchasing our AZ-Delivery SIM800L GSM GPRS module with antenna for Atmega328p. On the following pages, we will introduce you to how to use and setup this handy device.

Have fun!



AZ-Delivery

The AZ-Delivery SIM800L GSM GPRS module comes with a pin header, a solder antenna and an optional antenna with IPX compatible connector. The SIM800L chip can dial into GSM and GPRS networks. Provided it receives mobile phone calls, the module can send and receive SMS messages, make and receive calls and dial into the mobile 2G Internet.

Due to the variable input voltage in the range from 3.4V to 4.4V DC, you can operate the module directly with a lithium-ion battery. However, a direct power supply from the Atmega328p 5V output is not possible without a voltage regulator and, due to the short-term high currents of 2 amps at the start of a connection, not advisable. At 3.3V the module does not run stable.





The most important information in a nutshell

- » SIM800L chipset
- » Micro-SIM card slot
- » external or soldered antenna
- » Connection possibility for a loudspeaker and a microphone
- » Serial communication via D0/D1 or with the SoftwareSerial Library via many other pins
- » Power supply directly via lithium-ion battery or voltage converter:
3.4V ÷ 4.4V DC

NOTE: The SIM800L chip requires up to **2A** when establishing a connection! Make sure that your battery can do this. If your battery is not powerful enough, the connection may be lost and the battery may be damaged. **A damaged Lithium-ion battery represents a fire hazard!** In order to absorb peak loads, it may be useful to connect an electrolytic capacitor with a higher capacity in parallel to the voltage supply.

On the following pages you will find important links and information about:

- » Structure of the formwork
- » Sending and receiving an SMSes
- » Making a calls

This guide assumes that you can upload sketches to your Atmega328p, know the library manager, and are familiar with the serial monitor.



All links at a glance

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

Overview of AT commands for the module:

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

Programming interfaces

- » Arduino IDE: <https://www.arduino.cc/en/Main/Software>
- » Web-Editor: <https://create.arduino.cc/editor>
- » Atmega328p-extension for SublimeText: <https://github.com/Robot-Will/Stino>
- » Atmega328p extension "VisualMicro" for Atmel Studio or Microsoft Visual Studio:
<http://www.visualmicro.com/page/Arduino-for-Atmel-Studio.aspx>

Tutorials, Examples, Reference, Community

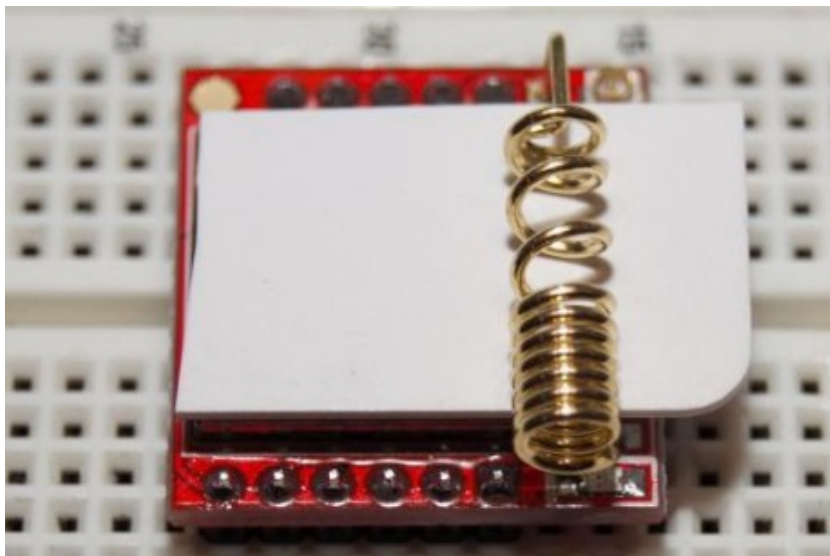
- » <https://www.arduino.cc/en/Tutorial/HomePage>
- » <https://www.arduino.cc/en/Reference/HomePage>

Interesting from AZ-Delivery

- » Atmega328p compatible Boards:
<https://az-delivery.de/collections/arduino-kompatible-boards>
- » Atmega328p accessories:
<https://az-delivery.de/collections/arduino-zubehor>
- » AZ-Delivery on Facebook:
<https://www.facebook.com/AZDeliveryShop/>

Setting up the module

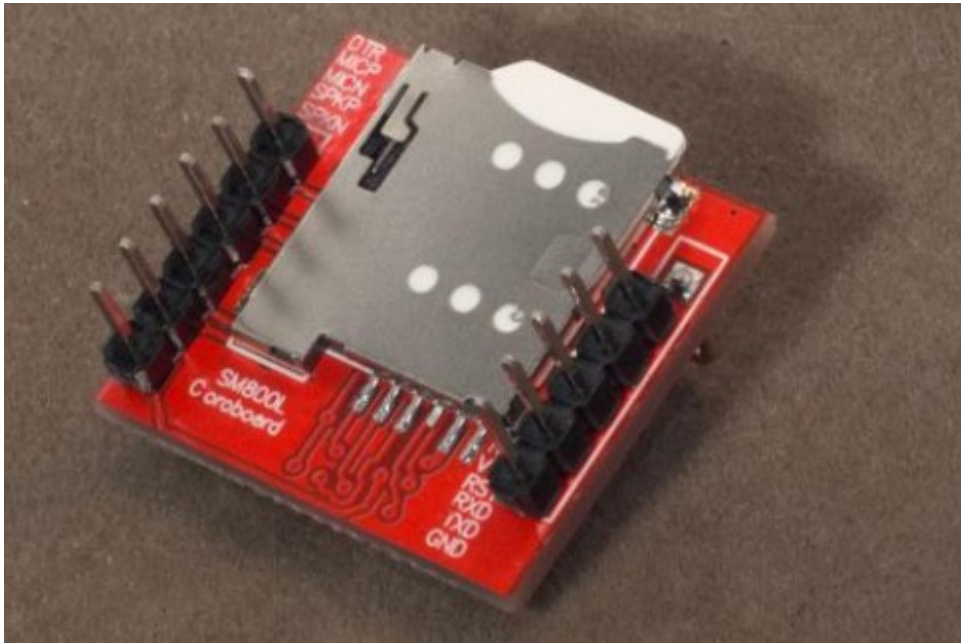
Before you start, you have to decide if you want to use the Plug-in antenna or the antenna for soldering. If you want to solder on the antenna, use flux and solder with the lowest possible melting point. Make sure that the antenna does not touch any other components. A piece of cardboard or plastic as a spacer for soldering simplifies the soldering process. A bad solder connection disturbs the reception!



The external antenna is particularly suitable if the module is installed in a metal housing. If you are using an external antenna, the solder joint for the internal antenna must remain free to avoid interference. You can use the supplied pin headers or solder cables directly.

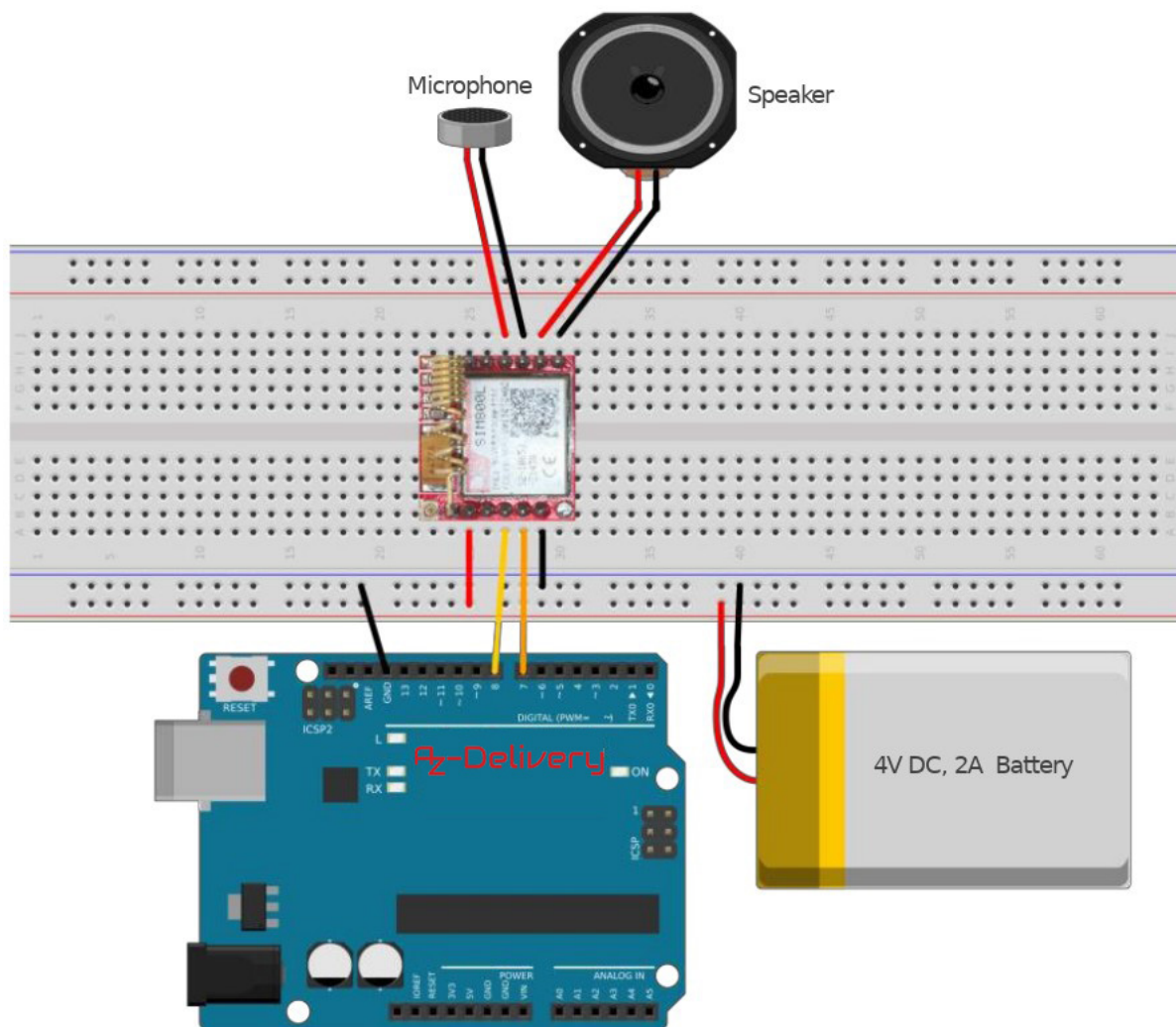
Az-Delivery

When you look at the SIM card socket, the Micro-SIM is marked with the notch after is pushed in at the top left until the spring mechanism engages.



Layout of the circuit

Make all the connections shown on image below. You can also place the RX and TX pins on other pins of the Atmega328p if you assign them correctly in your program. (The power supply of the Atmega328p should first be done via the USB port)



If you don't need the phone function, you can omit the microphone and speakers.

Later it may be useful to connect the reset pin of the module.

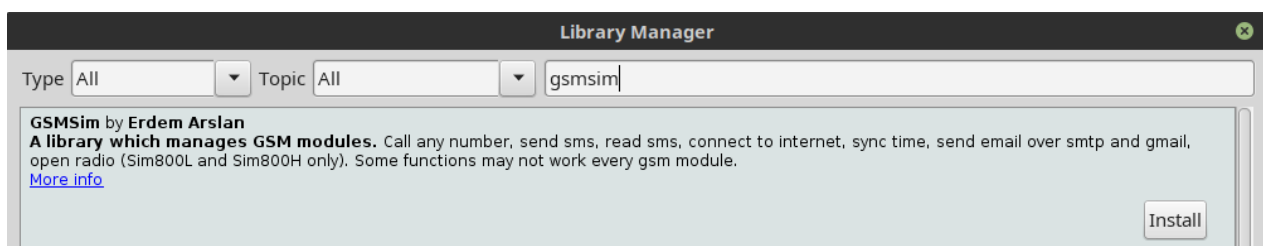
Az-Delivery

Setup in the Arduino IDE

In principle, the SIM800L module does not require a library, it can be controlled directly with AT commands via serial communication. An overview of the possible AT commands can be found below:

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

However, controlling the module using a library is much more convenient. First install the GSMS in Erdem Arslan's library via the library manager. On Github there is a good documentation in English language for the library under the link <https://github.com/erdemarslan/GSMSim>



Alternatively you can download GSMSim directly from Github and extract it into the libraries directory of your Sketchbook folder.

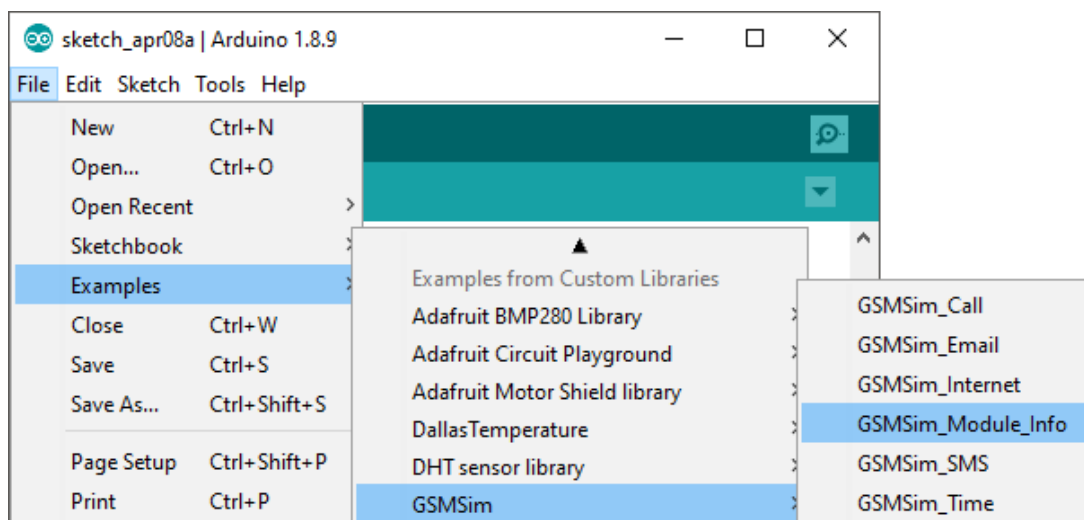
Setup in the Arduino IDE

In case of problems, check if the SoftwareSerial library is installed as it is needed. (Current versions of the Arduino IDE have the SoftwareSerial library installed by default.)

Test sketch for module

If you have connected everything correctly, you can open the example GSMSim_Module_Info. To do this, go to:

File > Examples > GSMSim > GSMSim_Module_Info



Az-Delivery

Since you haven't connected the reset pin of the module yet, you first have to make two small changes to the sketch. First remove the line:

```
#define RESET 2
```

Then change this line:

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

to

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

To make it a little more comfortable, you can add code line after the lines:

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

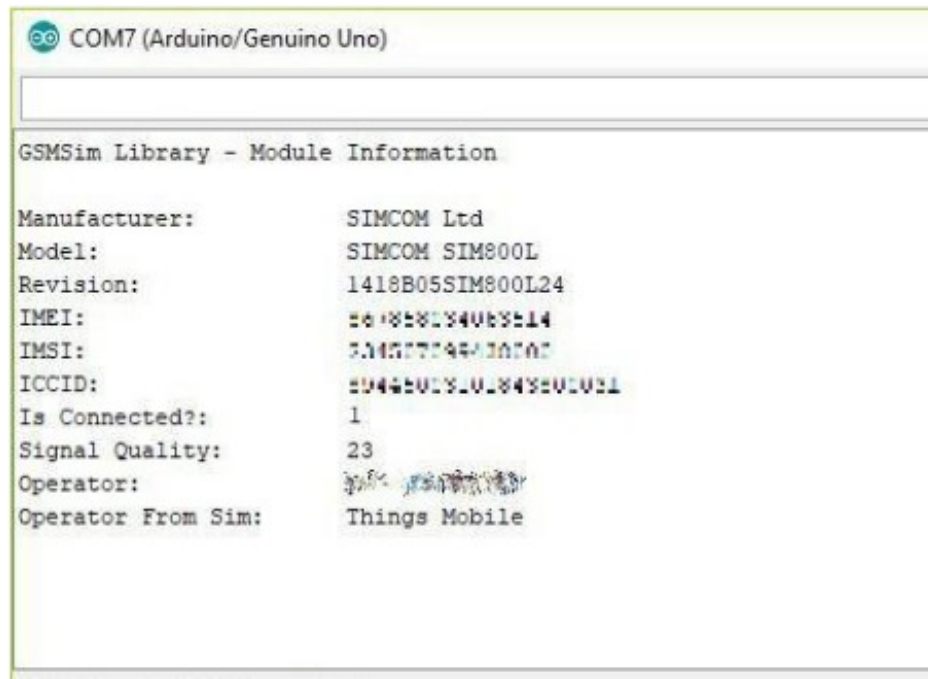
the following code line:

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

The Atmega328p will then wait until you have opened the serial monitor before proceeding with the program.

Az-Delivery

After you have uploaded the sketch on the Atmega328p and started the serial monitor, you should see the following after a short time:



The screenshot shows the serial monitor window for COM7 (Arduino/Genuino Uno). The text displayed is as follows:

```
COM7 (Arduino/Genuino Uno)

GSMSim Library - Module Information

Manufacturer:      SIMCOM Ltd
Model:             SIMCOM SIM800L
Revision:          1418B05SIM800L24
IMEI:              861818408314
IMSI:              31057744010707
ICCID:             8944101810184801011
Is Connected?:     1
Signal Quality:    23
Operator:           Things Mobile
Operator From Sim:  Things Mobile
```

(If you did not add the wait command, press the RESET button on the Atmega328p after opening the serial monitor)

Please check the following data before you proceed:

1. The IMEI should match the IMEI on your module.
2. Behind "*Is Connected:*" there should be a 1 for true
3. The value of "*Signal Quality*" should be between 1 and 31 (a value of 0 or 99 indicates an antenna error)
4. Operator from sim should match your provider

If there are no or wrong values in the second row of the table, the Atmega328p cannot establish a good connection to the module. If so, check your wiring.

Az-Delivery

You can change

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

to

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

then Module and Atmega328p agree on a baud rate.

If "*Is Connected:*" is followed by a 0 for false, check if your SIM card has a PIN, because it must be deactivated. (Standard for ThingsMobile SIM cards)

If the module still does not establish a connection despite good signal strength, check whether the SIM is activated at the provider.



Creating a Sketch

Start with an empty sketch. First you have to include the libraries and set the pins. The following lines of code show you how to do it. (Of course you can omit the comments)

```
//Include the required libraries:
#include <GSMSim.h>
#include <SoftwareSerial.h>
#define RX 7 // fixing the RX-Pin on the Atmega328p
#define TX 8 // fixing the TX-Pin on the Atmega328p
// #define RESET 2 // fixing the RESET-Pin on the Atmega328p
GSMSim gsm(RX, TX); //Calls the library
/*
 * other possibilities are:
 * GSMSim gsm; //RX=D7|TX=D8|Reset=D2|LEDpin=D13 as standard
 * GSMSim gsm(RX, TX, RESET);
 * GSMSim gsm(RX, TX, RESET, LED_PIN, LED_FLAG);
 */
```

Later you can also set the REST pin. The definition is still commented out. If you want to use other pins for RX and TX, make sure that TX is connected to RX and RX to TX.



Send the first SMS

To send an SMS, add the following lines of code to your sketch:

```
void setup() {  
  gsm.start(); // Starts the functions of the library  
  gsm.smsTextMode(true); // activates the SMS module of the module  
  char* Number = "+49170123456789"; // Specifies the number char*  
  Message = "Hello World!"; // Defines the message  
  //Send the message to the specified number:  
  gsm.smsSend(Number, Message);  
}  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Then all you have to do is upload the finished sketch to your Atmega328p. Each time you restart your Atmega328p, it will send an SMS to the specified number.



Read a received SMS message

The module stores incoming messages on the SIM card. Depending on the provider, these can be 10 to 50 SMS. The following commands list the unread SMS and display the first SMS in the memory via the serial monitor.

```
void setup() {  
    gsm.start(); // Starts the functions of the library  
    Serial.begin(9600); // Starts the connection to the PC  
    while (!Serial){ } // wait for the serial monitor  
    // displays a list of unread SMS messages:  
    Serial.println("List unread messages:");  
    Serial.println(gsm.smsListUnread());  
    // Outputs the first SMS in memory:  
    Serial.println("The first SMS in memory is:");  
    Serial.println(gsm.smsRead(1));  
    // if the memory is empty, the output is *INDEX_NO_ERROR  
}
```




Making a call

To make a call, the microphone and speaker should be connected. The telephone number is also defined here in a variable of type char. For example like this:

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

The call is answered with

```
gsm.call(Number);
```

The call is ended with the command:

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



Useful tips for the module

The library contains further examples for handling the AZ-Delivery SIM800L GSM GPRS module. There you will find possibilities how to retrieve data from the module, establish an internet connection or send an e-mail.

If you have problems with your module, always check first whether all connections are good or without loose contact and whether the input voltage is in the range of 3.4V to 4.4V DC. Plug-in boards are particularly susceptible to loose contacts. Serial communication can then be difficult.

For experimenting you can use a USB power supply and a DC-DC Step Down module if you don't have a suitable battery available. A suitable module from AZ-Delivery is the LM2596S DC-DC Step Down Module. However, you will also need a multimeter to adjust the voltage on the module.

To charge your Li-Ion battery, you can use the AZ-Delivery charge controller TP4056 Micro-USB.

With an FTDI adapter and a terminal program like YAT, you can send AT commands, even without Atmega328p, from the PC to the module. The bit rate of the module in YAT is 115200. An easy to install FTDI adapter is also available at AZ-Delivery, if you want to use the SIM800L GSM GPRS module directly on your PC, using a terminal.

Az-Delivery

For experiments with the PC there are also three possibilities to use an Atmega328p as FTDI adapter. These are:

1. Connect the rest of the pins of the Atmega328p to GND
2. Remove the Atmel chip
3. To upload the following sketch to the Atmega328p:

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

The RX pin of the Atmega328p is then D0, the TX pin is D1.

You've done it, you can now use your module for your projects.



Now it is time to learn and make the Projects on your own. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.

If you are looking for the high quality microelectronics and accessories, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

<https://az-delivery.de>

Have Fun!

Impressum

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