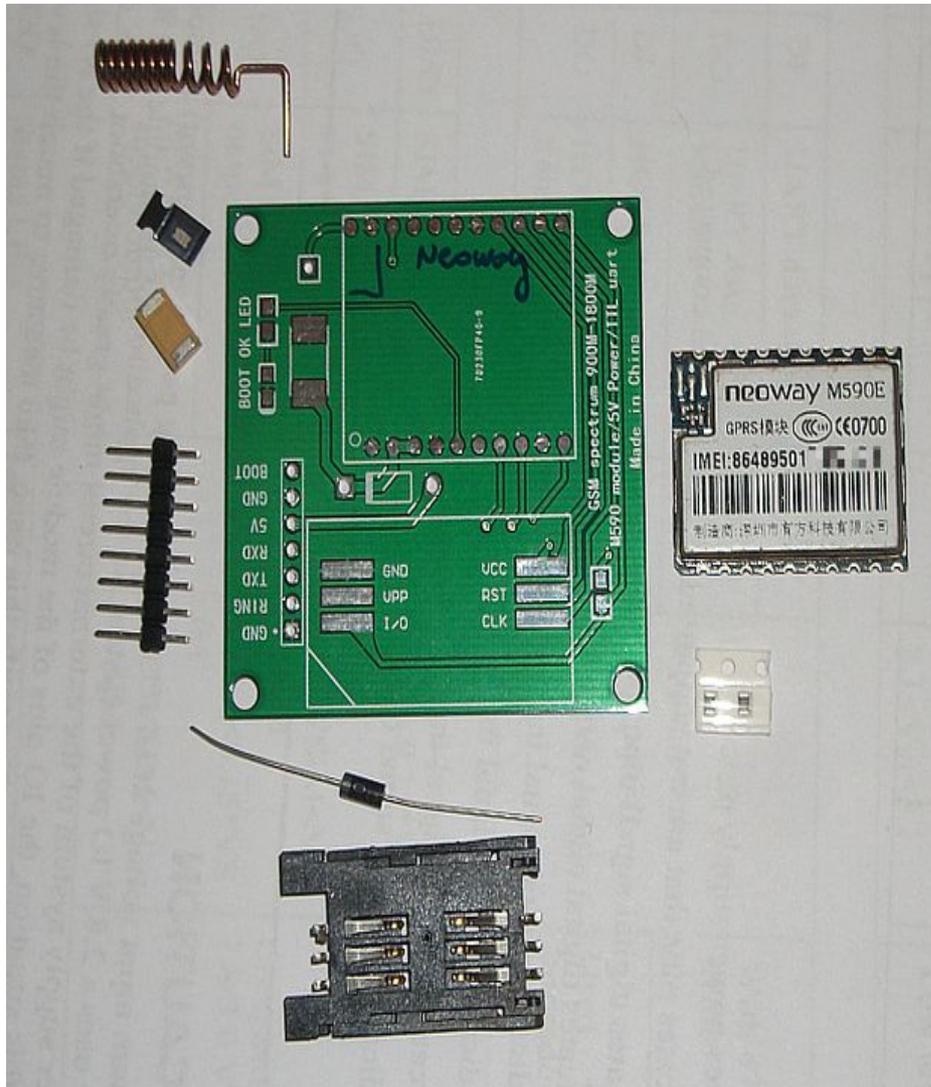


GSM/GPRS Module DIY Kit



This instructable is about an extremely cheap GSM/GPRS module which comes as a do it yourself kit. We are going to assemble the parts and do some basic operations through software terminal. Finally we send out SMS messages.

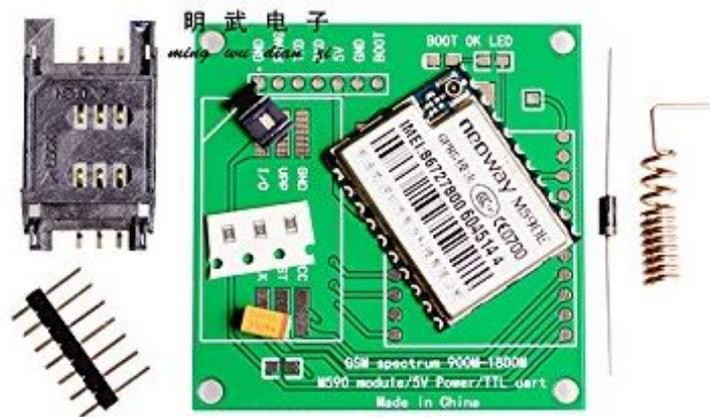
This is the first article of a serie. Later iblees will discuss:

- Controlling the GSM module by a PIC microcontroller (and maybe by arduino).
Update: [click here to read the PIC controlled SMS sending instructable](#).
- Connecting sensors to the microcontroller to send SMS alerts when the sensors are triggered. Sensors can be PIR detectors, ultrasonic sensors, or door opening detectors.

- One step beyond sending SMS: using the GSM/GPRS modul data transmission features to send data to a web server using POST method.
- One step further: sending data through FTP protocol.
- More triggers: using multiplexer to get up to 16 event source with only 3 pins of the microcontroller.

Further ibles will born as I develop my automated off-site security and yard-automation system.

Step 1: Neoway M590E GSM/GPRS Module



one result for "m590e gsm"

Show results for
DIY & Tools >
Building Supplies



MagiDeal M590E GSM GPRS Module Board 900M-1800M SMS Message DIY Kit CPU MCU Test Pack of 1
by MagiDeal
£1.93
More buying choices
£1.93 (2 new offers)

DIY & Tools: See all items

I was looking for the cheapest solution to give mobile communication features to my home automation hobby project. M590E is surprisingly underpriced: the time of writing this ible one can buy full GSM and GPRS functionality for **less than 2 GBP**. That's about 3 USD. Wow!

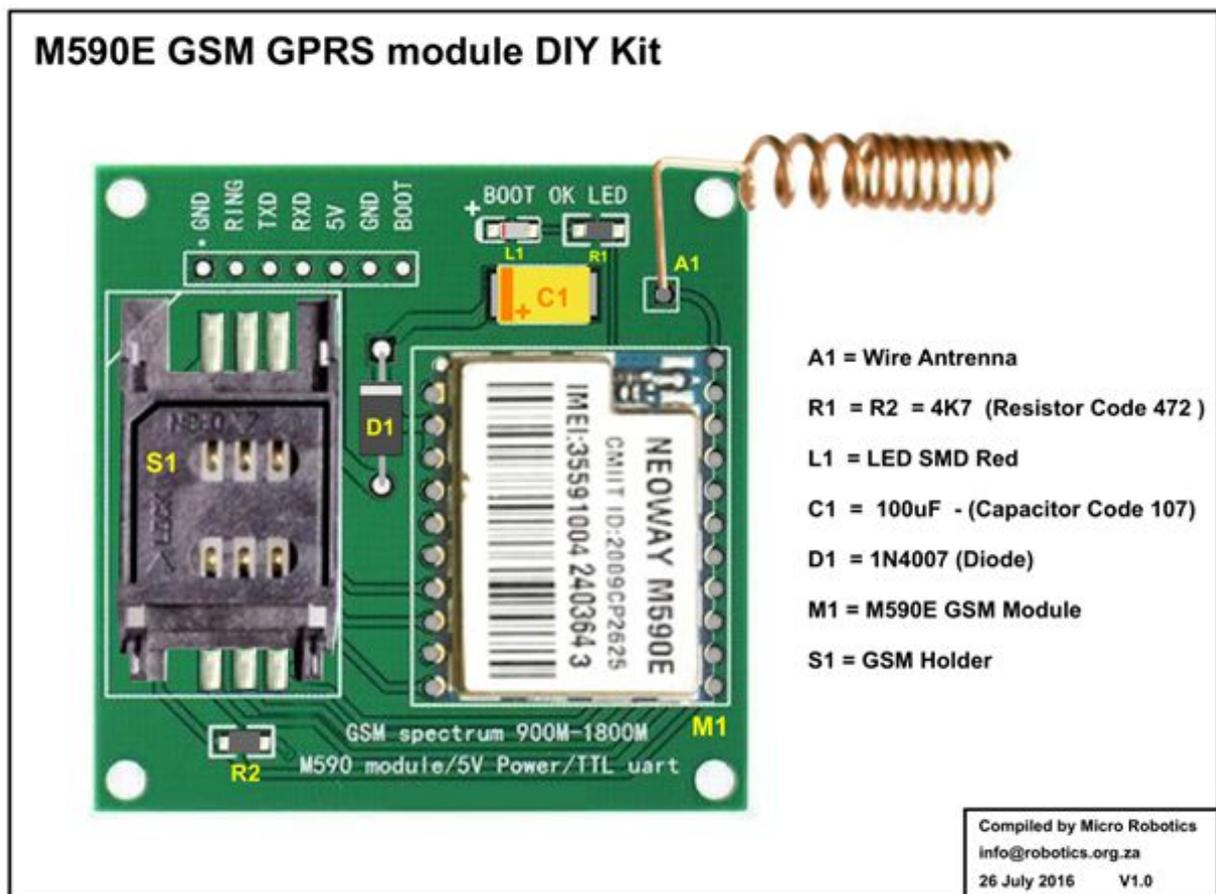
What we get for this price is impressive:

- ultra compact (28 x 22 mm) GSM/GPRS industrial wireless module
- support for both 900MHz and 1800MHz GSM frequencies
- standard AT instruction set (a very easy way to command&control the modem)

- integrated support of TCP/IP, http GET/POST, FTP and UDP data transfer protocols
- low energy consumption (measured virtually zero wattage with 3.2 mA at 3.9 V, and 180 mA peaks during SMS sending)
- accomodate any SIM card size (with a card adapter)

Well why this module is sooo cheap, you may ask. Not sure... but this is a 2G device and as 5G is coming these modems will be obsolete within a few years. But until that let's get good use of it! ;-)

Step 2: The Modified DIY Kit



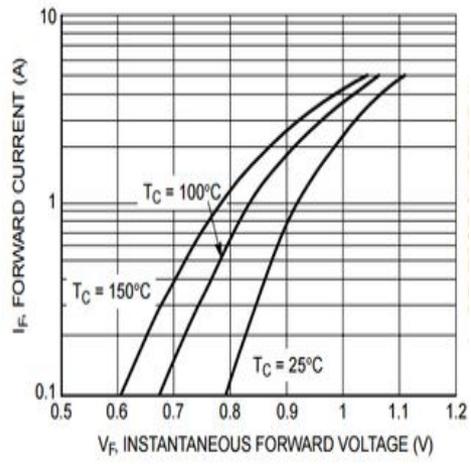


Figure 1. Typical Forward Voltage



Figure 34 Top view of the M590E V1 module

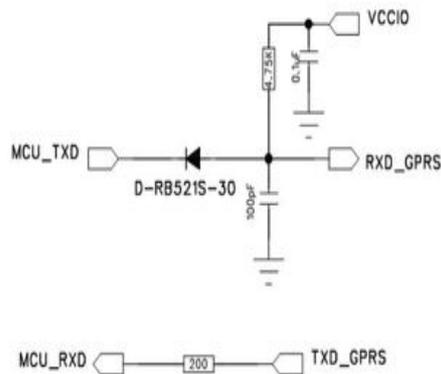
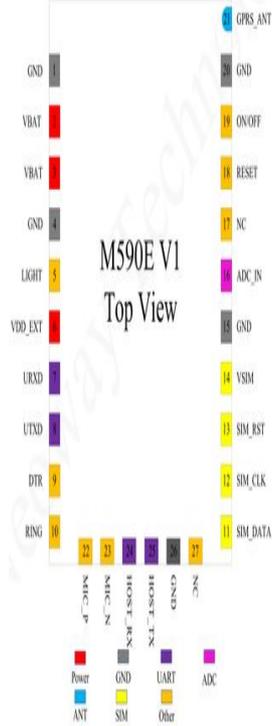


Diagram 7 5.0V MCU and module serial communication recommended circuit



Show All 8 Items

M590E is the module itself and the PCB is an independent product. You required to solder a few surface-mounted parts to the board, and the GSM modul itself of course.

You can relatively easily assemble the kit and get it working. There are a few blog posts on the internet illustrating that this kit can be used as it is. However if you read the manufacturer's specification for M590E you can recognise some design flaws of the DIY kit. These are obviously there to get the kit as cheap as possible and to simplify the process of assembling the board by hobbyists equipped with only a common soldering iron. So the design is all right and all my modifications to fit it to M590E hardware specification is optional.

First of all you need a good power source to feed the modem as it may suck as much as 2 amperes while transmitting. Many wall adapters are unable to provide this amount of current. If the power source is weak the modem will not boot up at all or repeatedly rebooting, or just fail to communicate.

Next, the modem can be operated within the 3.3V - 4.5V range. The recommended voltage is 3.9V. The kit is designed to be used with 5V (probably because arduino also works on 5V). So the DIY kit includes a 1N4007 diode with a 0.7V voltage drop, which results a 4.3V input for the GSM module. Simple but negligent solution. Voltage drop is a side effect of a diode not a feature. It is dependent on temperature and current flowing through the diode as you can see on the diagram copied from the datasheet. Also the diode cause some current drop and has a continous current limit of 1A that together may result inadequate operation (I mean: spontaneous reboot) of the GSM modem.

As I plan to use my GSM module in a rural area where temperature may vary from -25 to +35 °C I discarded the diode and opted to use a LM350T linear voltage regulator which easily provides the required stable 2A at 3.9V.

So I used a wire in place of the diode, as the picture shows.

Next modification is the operation LED. The DIY kit contained a red SMD LED and two 4.7 kOhm SMD resistors. One of them are for the LED. I do not like the red LED as it means warning or failure so I replaced it with a green one. The resistor is more important: with 4.3V (which the board is designed for) the voltage drop on the resistor is 2.3V. The LED eats about 15 mA which would require a 150 Ohm resistor. ($(4.3V - 2.0V) / 0.015A = 153 \text{ Ohm}$) So a 4700 Ohm resistor is waaay to high. I choose a 820 Ohm SMD resistor as replacement because I want the LED to be rather dim.

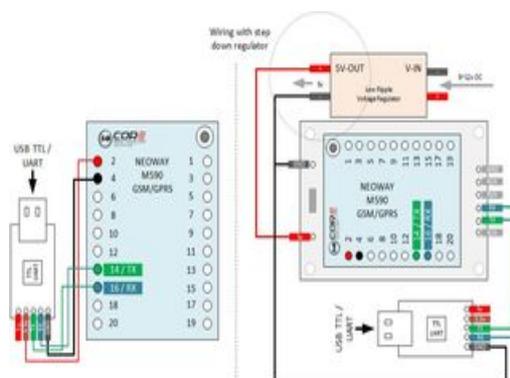
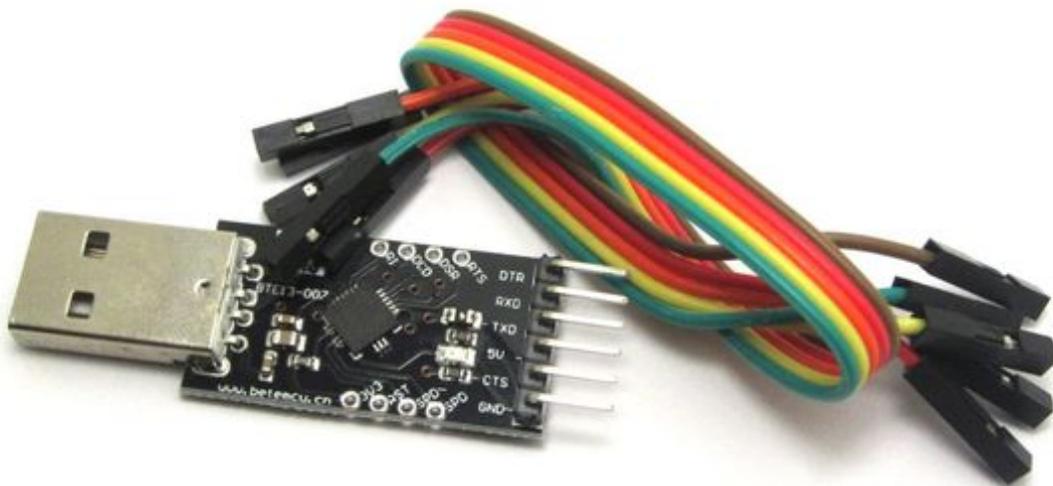
So I had an extra 4.7 kOhm resistor. Good news: the M590E datasheet states that a 4.7k resistor is required between Pin 6 (Vdd EXT) and Pin 7 (URXD). This resistor is required when the data signals of the microcontroller (e.g. arduino) is higher than 2.85V - and arduino operates on 5V so the resistor is needed to prevent damaging the GSM modem input port, yet the DIY kit does not contains one. The printed board also does not have a dedicated place for such a resistor. So I soldered the 4.7K resistor saved from the operation LED onto Pin 6 and Pin 7. (See the picture and the diagrams.)

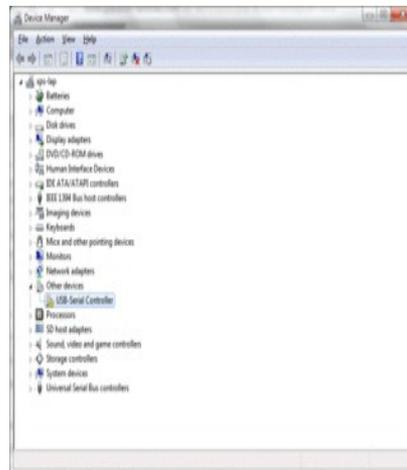
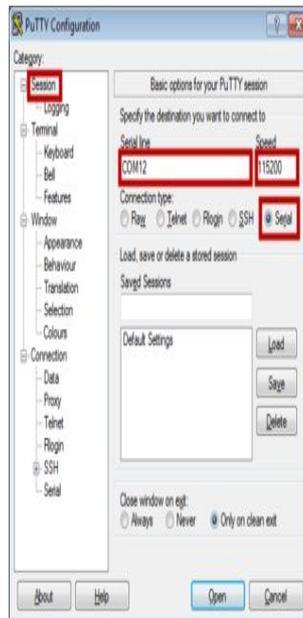
Next modification is the use of a schottky diode as RXD input pin of the board. This is part of the voltage reducing circuit recommended by the datasheet. Also, a 100 pF capacitor is soldered to this pin on the back of the board (see the pictures).

By the way, capacitors. The kit has a tantalum 100 μ F capacitor. The datasheet recommend two more ceramic capacitors: a 0.1 μ F to filter common digital noise and a 100 pF to filter radio frequency noise. I have soldered these onto the tantalum capacitor.

No other modifications was necessary. After I have finished soldering I've tested all and every soldering with a multimeter and fixed some of them. The modem was ready to boot up!

Step 3: Boot and Initialize





Before connecting your power source to your GSM modem you have to prepare your computer to communicate with the modem. This requires a hardware tool and two software tools.

To be able to control your modem from you personal computer's USB port you need a special driver.

Download Prolific USB driver and install it.

Now it is time to get the hardware: you need a TTL-USB (a.k.a. UART-USB) converter.

With this device you can control the GSM modem from your personal computer.

Now take a look at the connection pins on the converter. RXD pin of the converter should be connected to TXD pin of the modem. (TXD means: transfer data while RXD is read data: if one device writes data the other will read it and vica versa.) Similarly the

TXD pin of the converter is to be connected to RXD pin of the GSM modem. Next connect the GND pin of the converter to the same power source which you use to feed your GSM modem. (Again: use at least a 2A power source.)

Leave the 5V pin of the converter unconnected! If you connect the GSM module to 5V (and you omitted the diode when you assembled the kit) you certainly ruin the GSM module.

Next a terminal program is required. That is basically looks like a plain text window waiting to your commands to send them to the modem. I found PuTTY easy to use as it requires only 2 parameter from you: the number of COM port where your GSM modem is connetec to, and the speed of communication (baud rate) your modem is configured for. You can read more about [using PuTTY here](#).

When your TTL-USB converter is plugged in to the USB port In Windows7 the Device Manager window indicates which port number belongs to the converter (see the picture above). This is the port number you have to type in the terminal program.

The other parameter you have to give is the speed of the serial communication. My M590E has a default speed of 38400 bit/second.

After you've installed the USB driver, wired and connected your TTL converter and configured the terminal program you are ready to connect the positive terminal of your power source to the GSM modem and wait it to boot up.

If everything is fine the LED on the GSM modem starts flashing and the terminal window displays:

```
MODEM:STARTUP
```

If you have inserted a valid SIM card into the GSM modem's card holder before, and the SIM is free from PIN code soon you will see:

```
+PBREADY
```

That means phonebook is ready. Also means that your GMS modem is good to go, it can read your SIM card, and your power source can provide adequate amount of current. You made everything right!

However, if your the startup phase repeats endlessly you probably have power source problem. Check your solderings, wire connections and the power source itself.

Even worst if you do not see anything on the terminal window check the wiring of the TTL converter and the modem.

If there are weird characters on the terminal you have to select another speed for the

serial communication. Try these standard values until you get readable text on the terminal: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

Now type an AT command, the most basic one:

```
AT
```

Hit enter, and see as your modem responds:

```
OK
```

That's it! Your GSM modem operates. You did a great work. Only the easy and funny part is left: let's send an SMS!

Step 4: Sending SMS

Now it is time to play with the GSM module. Type to the terminal windows (with an Enter key at the end):

```
AT
```

You should get immediate response from the modem:

```
OK
```

M590E by default does not give any description or error code when a command results in error. Let's change this and enable error descriptions:

```
AT+CMEE=2
```

The response from the modem:

```
OK
```

Get the serial baud rate (the speed of communication):

```
AT+IPR?
```

The modem answers something like this:

```
+IPR:38400
```

Now try to talk to the SIM card. Ask the PIN state:

```
AT+CPIN?
```

If your card has not set a PIN code the answer is:

```
+CPIN: READY
```

Now see if the modem was able to connect to the GSM network:

AT+CREG?

If it is connected to GSM network the response is:

+CREG: 0,1

If you get 0,0 your modem was not able to register itself at your GSM provider probably due to the tiny antenna can not gather strong enough radio frequency signal. Try to change the position of the GSM modem.

Read the RF signal strength:

AT+CSG

Response should be similar to this:

+CSQ: 29,0

According to the data sheet 29 is excellent signal strength. With 17.2 I was still able to send SMS but at 16.3 dB message sending has failed.

Ask the name of the GSM service provider:

AT+COPS?

In my case the response is:

+COPS: 0,0,"vodafone HU"

And now send your first SMS!

AT+CMGF=1

Response is: OK

AT+CSCS="GSM"

Response is: OK

Give the phone number you want the SMS send to:

AT+CMGS="+36301234567"

After you press the Enter key you can start typing the message:

> Instructables rocks!

Then press the Ctrl + Z key pair to finish typing and to start sending the SMS.

If your RF signal strength is OK then in a few seconds you get a response from your modem similar to this:

```
+CMGS: 19
```

OK

That's it! You have built your custom GSM module, powered up, connected to your computer, controlled with AT commands and successfully sent an SMS.

Nice! ;-) Your wireless communication module is now ready to be used in some exciting DIY projects.