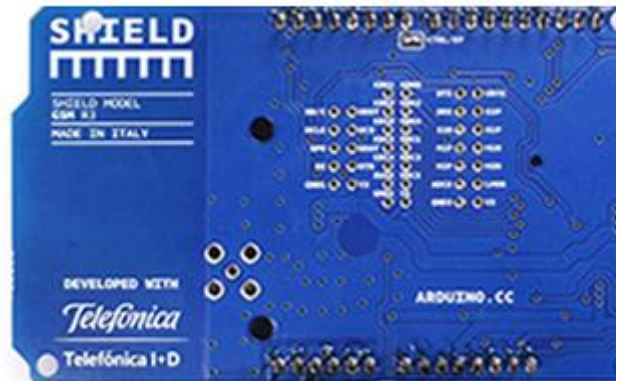


Arduino GSM Shield



Arduino GSM Shield Front

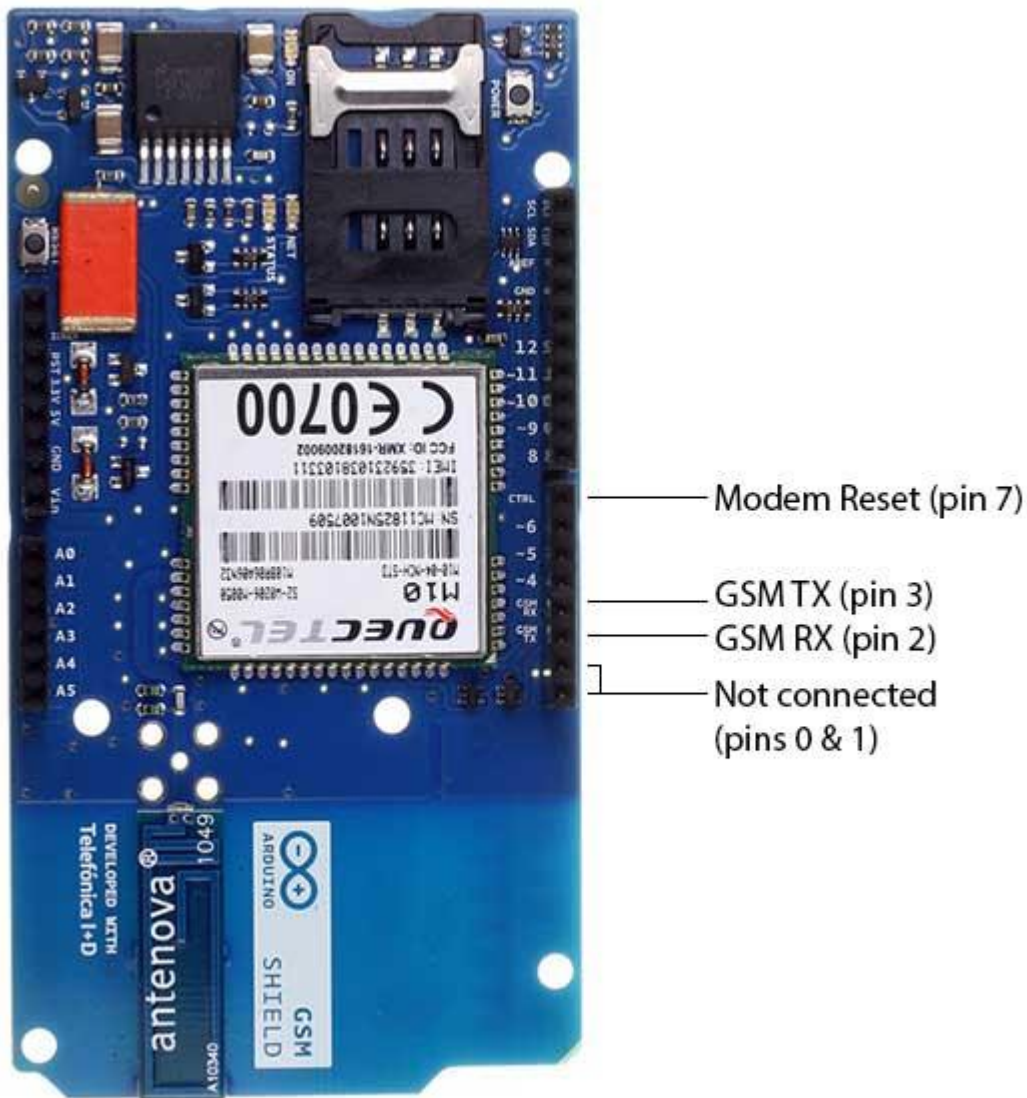
Arduino GSM Shield Back

Download: [PDF of GSM shield schematic](#), [Reference design](#)
The [GSM library](#) is included with [Arduino IDE 1.0.4](#) and later.

Overview

The Arduino GSM Shield connects your Arduino to the internet using the GPRS wireless network. Just plug this module onto your Arduino board, plug in a SIM card from an operator offering GPRS coverage and follow a few simple instructions to start controlling your world through the internet. You can also make/receive voice calls (you will need an external speaker and microphone circuit) and send/receive SMS messages.

As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!



- Requires an Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Connection with Arduino Uno on pins 2, 3 (Software Serial) and 7 (reset). [See these notes](#) for working with a Mega, Mega ADK, or Leonardo.

Description

The Arduino GSM Shield allows an Arduino board to connect to the internet, make/receive voice calls and send/receive SMS messages. The shield uses a radio modem M10 by Quectel ([datasheet](#)). It is possible to communicate with the board using [AT commands](#). The [GSM library](#) has a large number of methods for communication with the shield.

The shield uses digital pins 2 and 3 for software serial communication with the M10. Pin 2 is connected to the M10's TX pin and pin 3 to its RX pin. [See these notes](#) for working with an Arduino Mega, Mega ADK, or Leonardo. The modem's PWRKEY pin is connected to Arduino pin 7.

The M10 is a Quad-band GSM/GPRS modem that works at frequencies GSM850MHz, GSM900MHz, DCS1800MHz and PCS1900MHz. It supports TCP/UDP and HTTP protocols through a GPRS connection. GPRS data downlink and uplink transfer speed maximum is 85.6 kbps.

To interface with the cellular network, the board requires a SIM card provided by a network operator. See the [getting started page](#) for additional information on SIM usage.

The most recent revision of the board uses the 1.0 pinout on rev 3 of the Arduino Uno board.

Notes on the Telefonica/Bluevia SIM included with the shield

The GSM shield comes bundled with a SIM from Telefonica/Bluevia that will work well for developing machine to machine (M2M) applications. It is not necessary to use this specific card with the shield. You may use any SIM that works on a network in your area.

The Bluevia SIM card includes a roaming plan. It can be used on any supported GSM network. There is coverage throughout the Americas and Europe for this SIM, check the [Bluevia service availability page](#) for specific countries that have supported networks.

Activation of the SIM is handled by Bluevia. Detailed instructions on how to register and activate your SIM online and add credit are included on a [small pamphlet](#) that comes with your shield. The SIM must be inserted into a powered GSM shield that is mounted on an Arduino for activation.

These SIM card come without a PIN, but it is possible to set one using the GSM library's [GSMPIN class](#). You cannot use the included SIM to place or receive voice calls.

You can only place and receive SMS with other SIMs on the Bluevia network.

It's not possible to create a server that accepts incoming requests from the public internet. However, the Bluevia SIM will accept incoming requests from other SIM cards on the Bluevia network.

For using the voice, and other functions of the shield, you'll need to find a different network provider and SIM. Operators will have different policies for their SIM cards, check with them directly to determine what types of connections are supported.

Power requirements

It is recommended that the board be powered with an external power supply that can provide between 700mA and 1000mA. Powering an Arduino and the GSM shield from a USB connection is not recommended, as USB cannot provide the required current for when the modem is in heavy use. The modem can pull up to 2A of current at peak usage, which can occur during data transmission. This current is provided through the large orange capacitor on the board's surface.

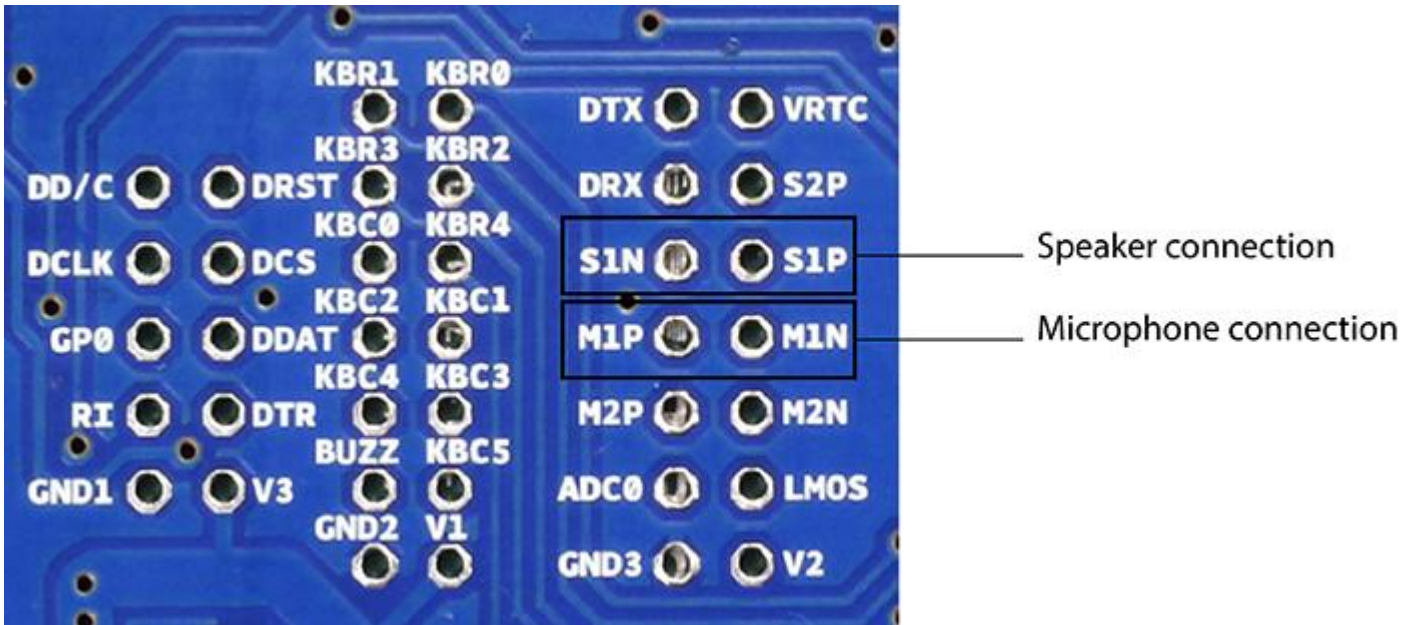
On board indicators

The shield contains a number of status LEDs:

- On: shows the Shield gets power.
- Status: turns on to when the modem is powered and data is being transferred to/from the GSM/GPRS network.
- Net: blinks when the modem is communicating with the radio network.

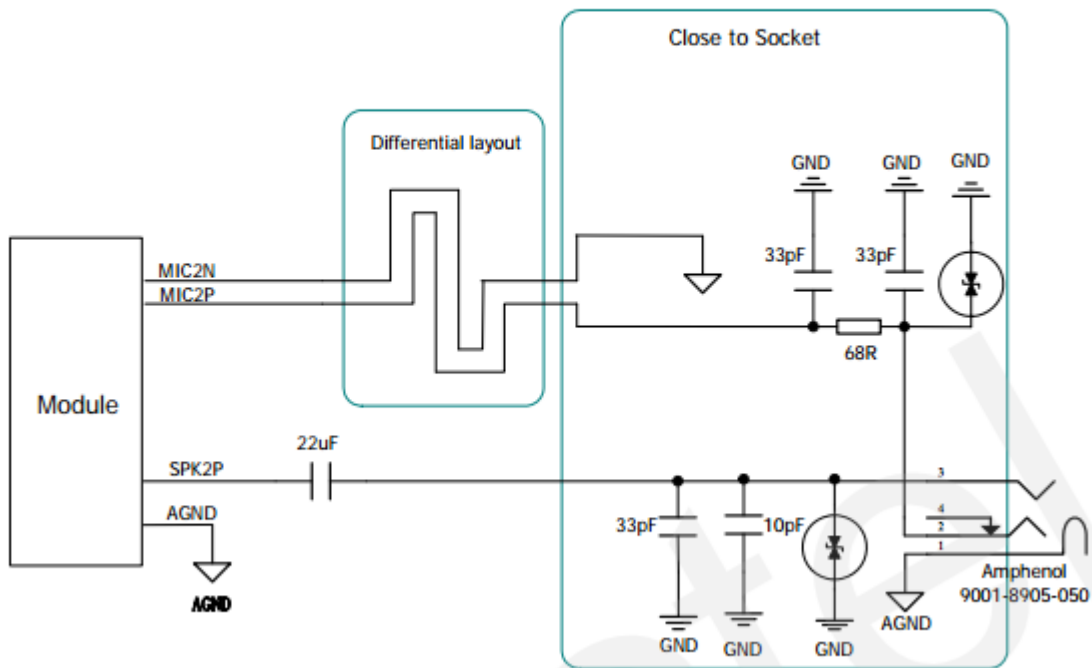
On board interfaces

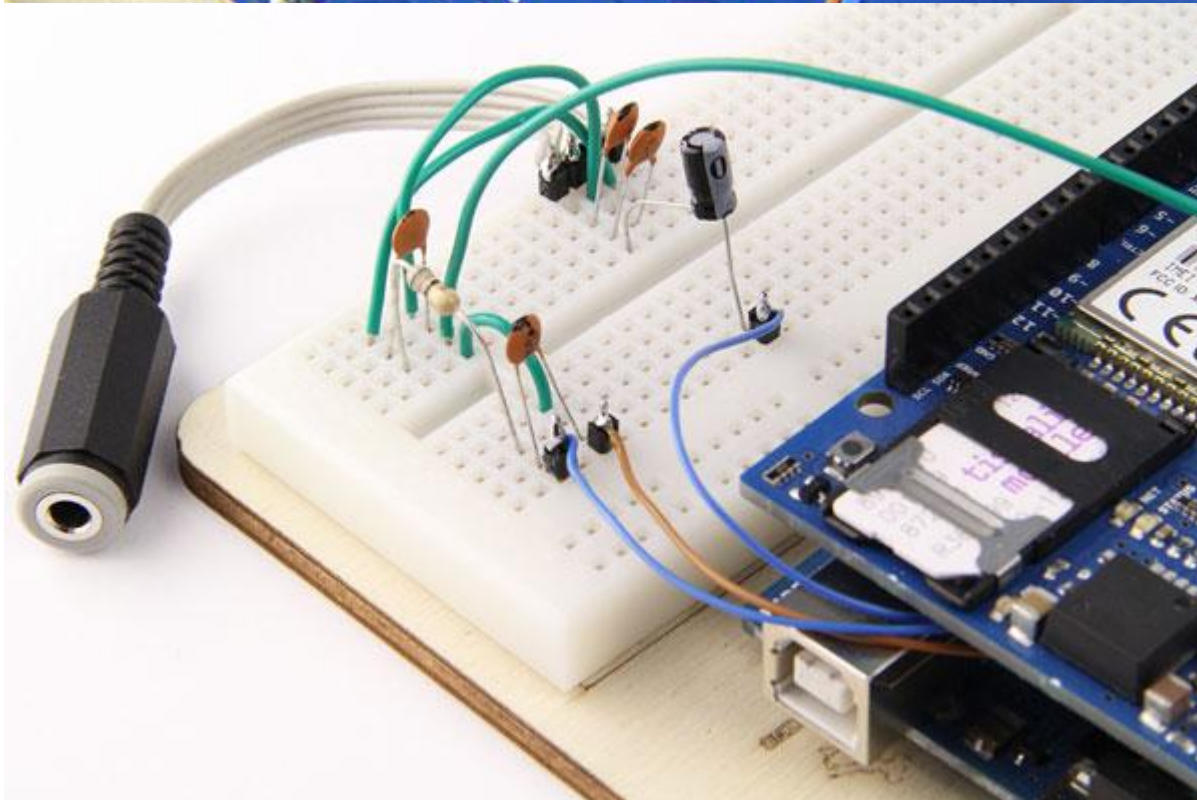
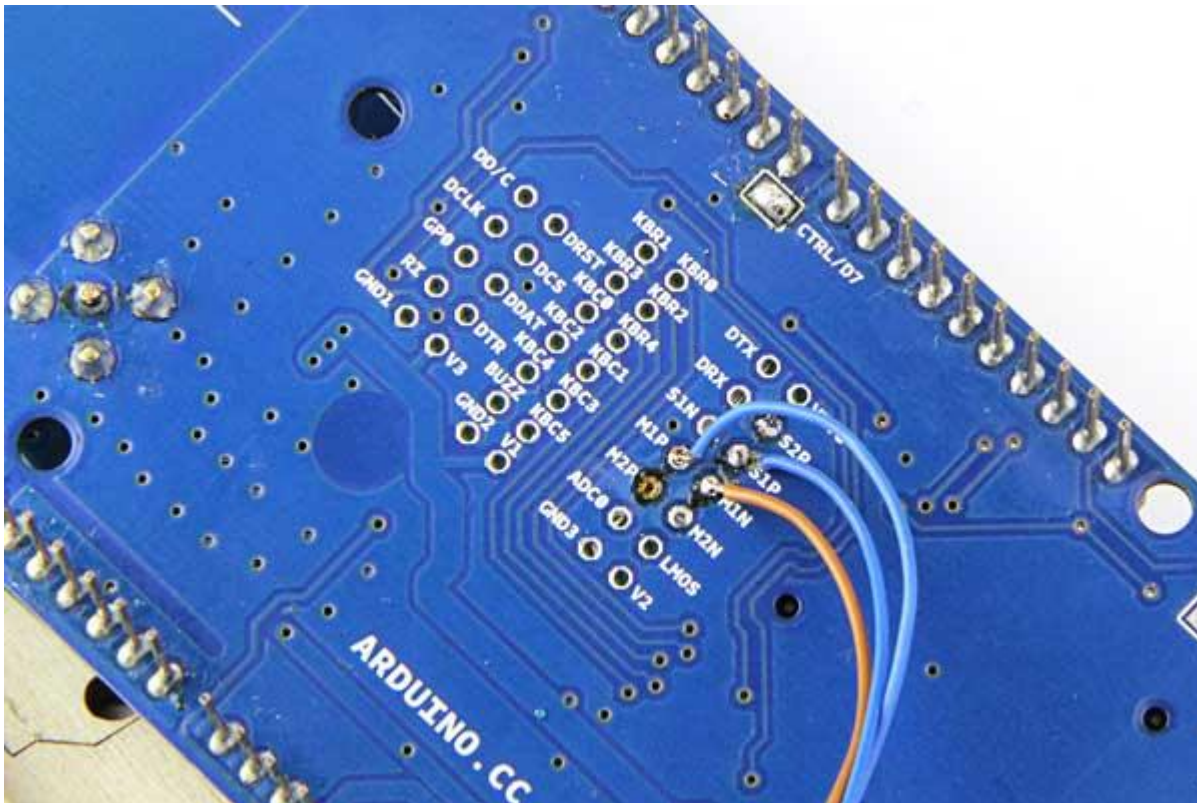
The shield supports AIN1 and AOUT1 as audio interfaces; an analog input channel and an analog output channel. The input, exposed on pins MIC1P/MIC1N, can be used for both microphone and line inputs. An electret microphone can be used for this interface. The output, exposed as lines SPK1P/SPK1N, can be used with either a receiver or speaker. Through the modem, it is possible to make voice calls. In order to speak to and hear the other party, you will need to add a speaker and microphone.



On page 43 of the [modem documentation](#), there is an example voice and sound circuit that will connect to an earphone:

3.9.3 Earphone interface configuration





There are two small buttons on the shield. The button labelled "Reset" is tied to the Arduino reset pin. When pressed, it will restart the sketch. The button labelled "Power" is connected to the modem and will power the modem on and off. For early versions of the shield, it was necessary to press the *power* button to turn on the modem. Newer versions of the board will turn the modem on automatically. If you have an early version of the shield, and it does not turn on automatically, you can solder a jumper to the CTRL/D7 pad on the reverse side of the board, and it will turn on when an attached Arduino receives power.

Bridged pin to automatically power the modem



Several of the modem pins are exposed on the underside of the board. These provide access to the modem for features like speaker output and microphone input. See the datasheet for complete information.

