Iridium-Shield for *Arduino and Clones*

***Arduino* Board Duemilanove (Atmega328)**
***Arduino* Board UNO (Atmega328)**
***Arduino* Board Mega2560 (Atmega2560)**
Clones

Manual

15.01.2015

Contents

General

Arduino is an open-source physical computing platform based on flexible and easy-to-use hardware and software which is used to for the development of prototypes. The *Arduino* board is composed of a micro controller with several digital and analogue inputs and outputs and interfaces. Including JAVA based development environment, based on processing and other open-source software. Developed programs can directly by loaded to the *Arduino* board.

Arduino has a great fan community.. and of course their own website with many example applications as well as an extensive forum → http://www.arduino.cc/

To extend the functionality of the *Arduino* boards and to control other devices, so-called Shields (extension boards) are plugged to the *Arduino* board. With the ***Iridium-Shield*** the *Arduino* board is enabled to use "real satellite communication over the Iridium network". The Iridium system offers worldwide unique network coverage.

The ***Iridium-Shield*** is plugged onto the *Arduino* mainboard and communicated with the micro controller via the serial interface. The only external component required is an Iridium antenna (connected via a SMA connector).

Please read this manual completely and carefully before initial use. It describes correct use and includes important instructions for the installation of the ***Iridium-Shields*** with the ***Arduino*** board. The manufacturer accepts no liability for damage resulting from improper use. All guarantee claims are then void.

Scope of delivery

Hardware

Everything there? The delivery of an ***Iridium-Shield*** contains:

- the ***Iridium-Shield***
- this manual
- an Iridium SBD module, type "Iridium 9602 SBD Transceiver" (optional)
- the *Arduino* board (optional)
- an Iridium antenna (optional)
- a power supply (optional)

Software

Software pack (available for download) containing:

- a library to use the ***Iridium-Shield***
- Example program for using the Iridium module
- Design and schematic of the *Arduino* board / ***Iridium-Shield***

NOTE: The link to the *Arduino* development (incl. driver) is available here **Links**

System requirements

General

- *Arduino* board ''Duemilanove'', ''UNO'' or ''Mega2560'' or suitable clones
- Iridium contract with an airtime provider (if requested we can support you with the assessment of Iridium agreements of various airtime providers)

Programming

- *Arduino* development environment
- USB interface
- Windows / Mac OS X / Linux: 32 or 64 bit

Technical specification

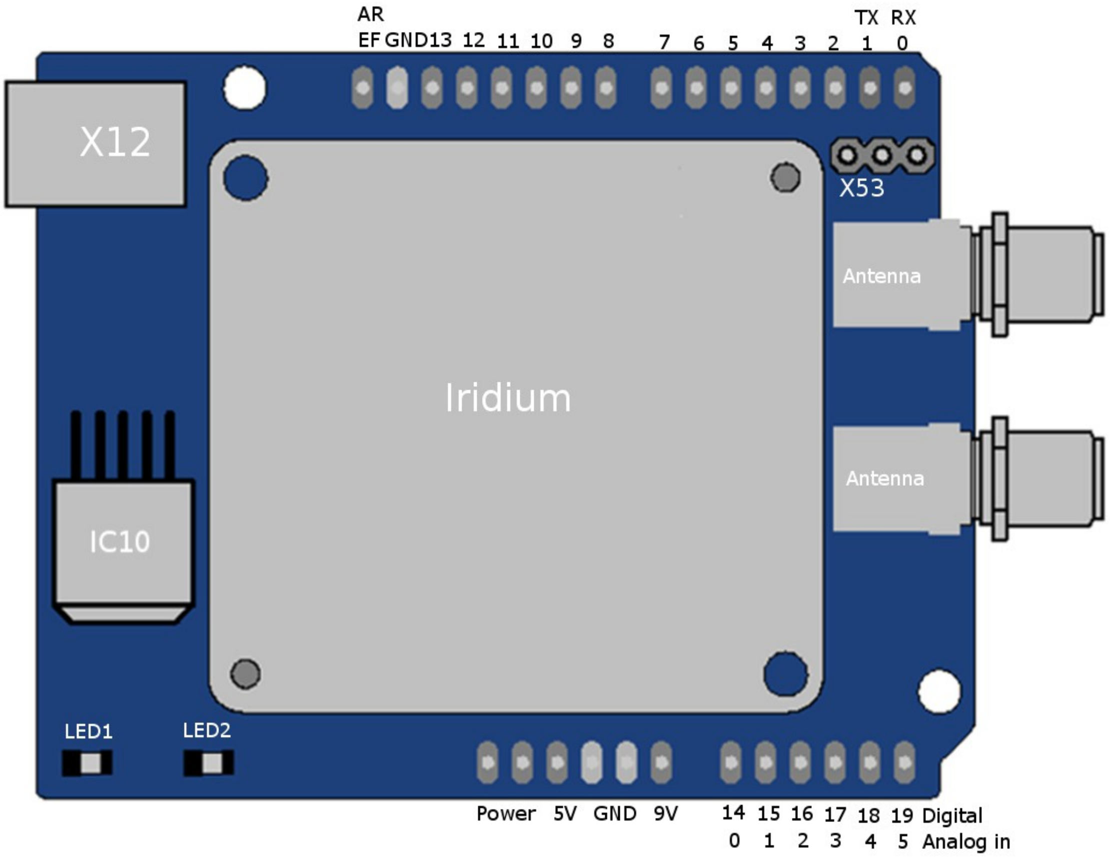
General

- Operating voltage: 5VDC ... 9VDC
- required mainboards: Duemilanove/Uno, Mega2560 or Arduino clones

Iridium

- Slot for an Iridium 9602 SBD Transceiver
  - to be used with a contract of an airtime provider
  - control via the serial interface (TTL level)
- exact description of the Iridium transceiver please see here: http://www.antrax.de/downloads/iridium-shield/quake-datasheets/

Iridium-Shield



Installation

Hardware

- plug the Iridium module on to the ***Iridium-Shield***
- connect the Iridium antenna
- plug the ***Iridium-Shield*** onto the *Arduino* board
- connect the power supply to the *Arduino* board and the *Arduino* board to a PC using an USB cable

NOTE:

It is important to have a good sight to the sky with the Iridium antenna (the label "SKY SIDE" on the antenna housing is to be taken seriously!). A bad satellite connection is quite always a result of unfavourable antenna placement or connection. Only use adequate extension cables, e.g. Aircom Plus or LMR®-400 cable and high quality plug connectors from brand manufacturers like Huber+Suhner, Radiall or Telegärtner. Simple RG-58- or RG-174 connector cables are mostly not suitable!

Iridium antennas and its control are significantly more sophisticated as for example mobile communication antennas.

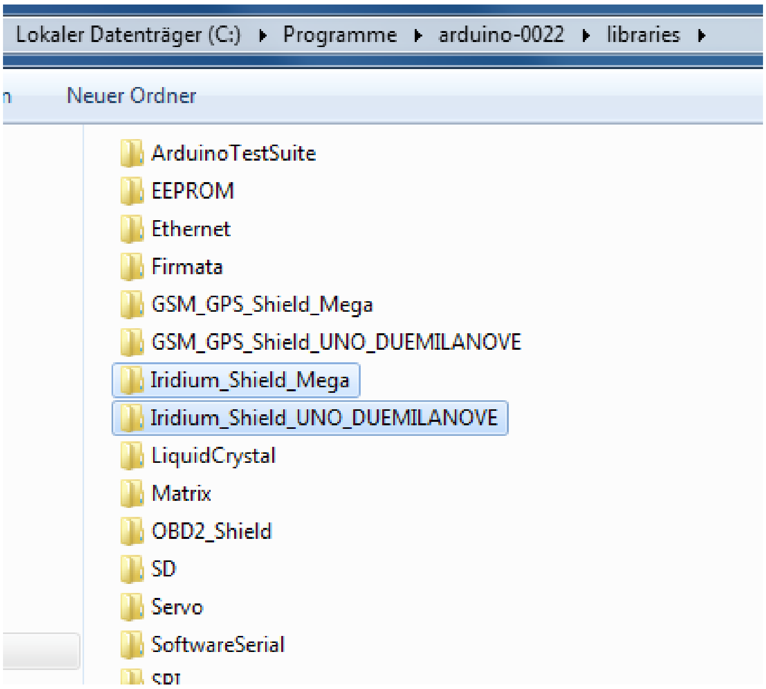
Software

Install drivers to the *Arduino* board from the directory of the *Arduino* development environment (if required).

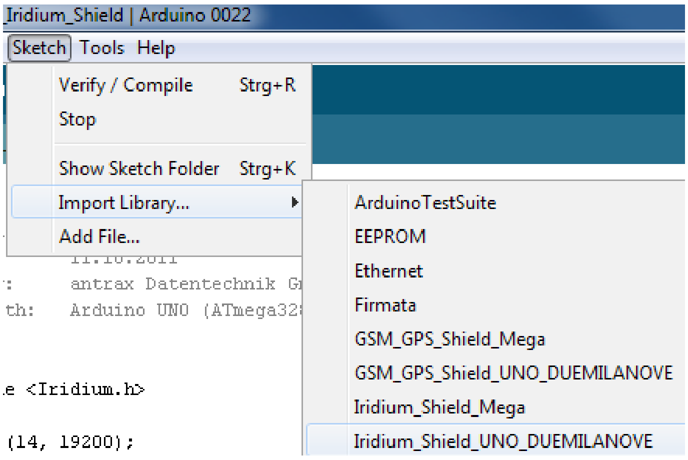
Operation

Library

To make working with the ***Iridium-Shield*** as simple as possible a library for this shield was developed.
Please find this library and two example programs for sending and receiving messages in the download section of our website. After unzipping the file please copy the content to the "libraries" folder of the Arduino directory (e.g. "f:\arduino-1.0.1\libraries\").



The library can be added to the currently open program with just 3 clicks (#include <Iridium.h>). Of course this library can be modified or extended to your own requirements.



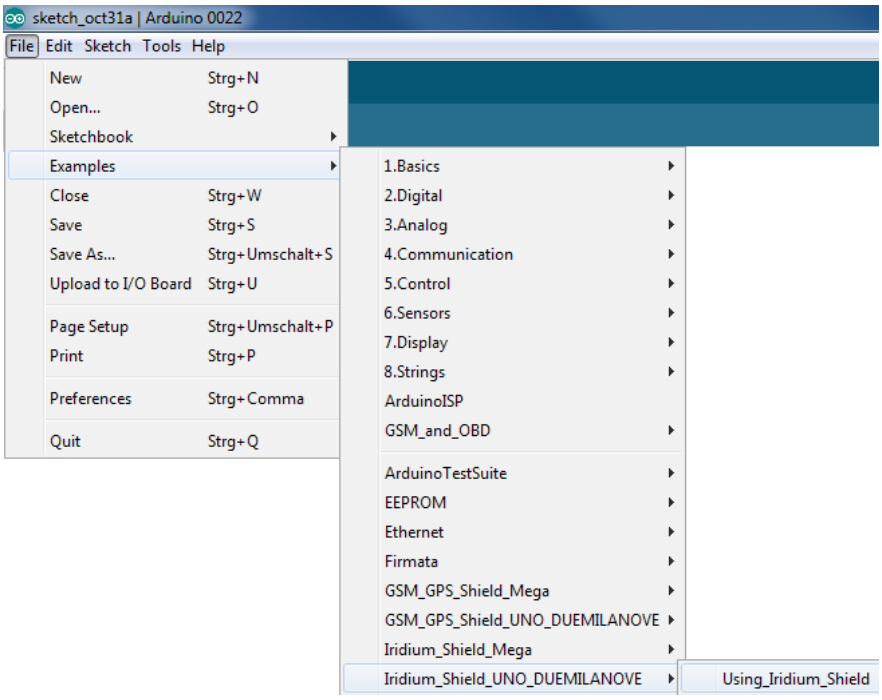
Programming the Arduino-Board

The ***Iridium-Shield*** is always addressed and controlled by the software used on the *Arduino* board. An example program is supplied. It is very easy to transfer a software to the *Arduino* board with the *Arduino* development environment. The required steps are described as follows::

- plug the ***Iridium-Shield*** to the *Arduino* mainboard and connect with a PC via the USB interface (please have the USB drivers installed already)

- open the *Arduino* development environment
  - see: File → Examples
  - Iridium
  - Examples

- since the *Arduino* board (despite USB connection) is shown as serial device to the PC, please choose
  - Tools → Serial Port as used interface

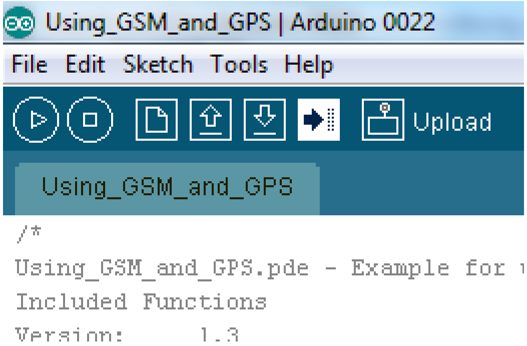


- use the upload button (highlighted in white in this picture) to directly upload the the modified program to the *Arduino* board

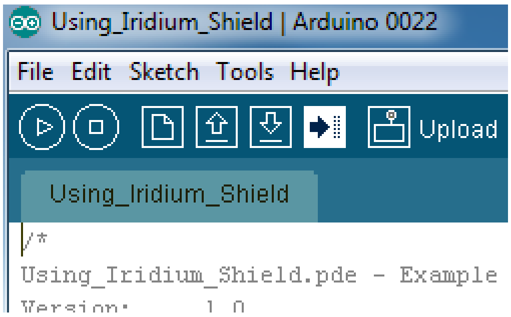
- after the message "done" is shown at the lower end of the window the transfer has been completed successfully

- the program is automatically started

- via the serial port in the Arduino development environment, the communication between the Iridium module and the Arduino board can monitored and where required received messages can be displayed. (ATTENTION: When the monitor is opened the program starts from beginning!)



```
/*
Using_GSM_and_GPS.pde - Example for
Included Functions
Version:      1.3
```


Using_Iridium_Shield | Arduino 0022
File Edit Sketch Tools Help
Upload
Using_Iridium_Shield

```
/*
Using_Iridium_Shield.pde - Example
Version:      1.0
```

<u>Note:</u> The ***Iridium-Shield*** always needs to be operated with an adequate power supply. The current consumption of the Iridium module is up to 1,5A. Using an USB cable to power the Iridium module is insufficient!

Example program "Iridium_Shield_Send"

The example program "Iridium_Shield_Send" shows how an SBD message from the ***Iridium-Shield*** is sent.

The demo software does the following in sequence:

- Initialising the Arduino system
- Turning ON the ***Iridium-Shields*** via Pin 14 (Pin 54 when using Arduino Mega 2560) of the Arduino board. LED2 is on.
- Write text to be transmitted to the sending buffer (Mobile Originated Buffer – MOB). Only one message at a time can be stored in the MOB, i.e. a possibly stored previous message in the send memory is overwritten with the new message.
- Initiate transfer (contains satellite connect and actual sending procedure). Depending on the reception location this process takes a few seconds or up to several minutes. The data exchange between ***Iridium-Shield*** and satellite is called SBD session. Here the message in the send memory is sent to the satellite (Caution: the message itself remains in the send memory, it is only copied and sent).

  During the SBD session it is also checked whether a message for the Iridium module is present on the satellite. If this is the case, the message is transferred from the satellite to the receiving memory (Mobile Terminated Buffer – MTB) of the module. The receiving memory also only stores one message, whereby a previously stored message will always be overwritten.
- Output of the result message (sent/not sent)
- Turn OFF the ***Iridium-Shield***

LED1 indicates whether the Iridium module has radio reception (after Power-On this may take up to several minutes).

The complete communication with the Iridium module is via simple AT commands in clear text. For a more sophisticated communication the demo routines in the libraries can be modified.

The demo routines communicate with he Iridium module as follows (trace on the serial interface):

(blue = Arduino mainboard ---> Iridium module)

(red = Iridium module ---> Arduino mainboard)T

```
IRIDIUM-Shield on!
AT
AT
OK
ATE0
ATE0
OK
AT+CSQ
+CSQ:0
OK
signal quality not good enough! wait...
AT+CSQ
+CSQ:2
OK
AT+SBDREG
+SBDREG:2,32
OK
Connected
AT+SBDWT=this is my important message ...
OK
AT+CSQ
+CSQ:2
OK
```

```
AT+SBDI
+SBDI: 1, 337, 0, 0, 0, 0
OK
AT+SBDD0
0
OK
mo buffer cleared!
Your Message sent successfully!
IRIDIUM-Shield off!
```

Example program "Iridium_Shield_Receive"

The example program "Iridium_Shield_Receive" explains how a SBD message is received by the ***Iridium-Shield***.

The demo software does the following in sequence:

- Initialising the Arduino system
- Turning ON the ***Iridium-Shields*** via Pin 14 (Pin 54 when using Arduino Mega 2560) of the Arduino board. LED2 is on.
- Initiate receiving procedure (contains satellite connect, a status query and the actual sending procedure). Depending on the reception location this process takes a few seconds or up to several minutes. The data transfer between Iridium module and satellite is called SBD session. Hereby the Iridium module queries the satellite whether a receiving message is stored for the own module ID.
  If this is the case, the message is transferred from the satellite to the receiving memory (Mobile Terminated Buffer – MTB) of the module. The receiving memory of the module also only stores one message, so that a possible previous message is always overwritten.
- Output of the result message and the received message
- Turn OFF the ***Iridium-Shield***

LED1 indicates whether the Iridium module has radio reception (after Power-On this may take up to several minutes).

The message to be sent needs to previously be sent as enclosure (i. e. text file) of an e-mail to the address "data@sbd.iridium.com". The reference/subject line of the e-mail needs to contain the ID of the Iridium module, to which the message has then to be sent to.

The demo routines communicate with he Iridium module as follows (trace on the serial interface):

(blue = Arduino mainboard ---> Iridium module)

(red = Iridium module ---> Arduino mainboard)T

```
IRIDIUM-Shield on!
AT
AT
OK
ATE0
ATE0
OK
AT+CSQ
+CSQ:5
OK
AT+SBDREG
+SBDREG:2,0
OK
Connected
AT+CSQ
+CSQ:5
OK
AT+SBDI
+SBDI: 0, 353, 1, 60, 32, 0
OK
The message received successfully!
AT+SBDRT
+SBDRT:
```

```
Hello IRIDIUM-Modul!

OK

AT+SBDD1

0

OK

mt buffer cleared!

Received Message: Hello IRIDIUM-Modul!

OK

IRIDIUM-Shield off!
```

Particularities of satellite communication

Satellite communication is partly comparable to the communication in mobile communication, but has has also striking differences:

- The accounting system of airtime providers is partly different from mobile communication providers. For example the query whether a SBD message is available must be actively effected. In addition, this query is already charged. It should therefore possibly be combined with the sending of SBD messages.
- There are "retention periods" (3 minutes), to prevent from too frequent login to the satellite network.
- The Iridium module can always send a SBD message to only one (single) e-mail address. This e-mail address needs/has to be specified with the initial set-up of the the activation of the connection.

Links

Please find additional information here

- on our website → http://www.antrax.de
- Arduino development environment → http://arduino.cc/en/Main/Software
- Schematic and layout → http://www.antrax.de/downloads/iridium-shield/antrax-hardware/
- Iridium library → http://www.antrax.de/downloads/iridium-shield/antrax-examples/
- exact description of the Iridium transceiver and SBD description see here → http://www.antrax.de/downloads/iridium-shield/quake-datasheets/