kick-start for h-nanoGSM 1.08 by itbrainpower.net
ARDUINO & RASPBERRY PI GSM +BTH 3.0 (VOICE/SMS/GPRS/DTMF) shield (nano)

kick-start for h-nanoGSM 1.08 by itbrainpower.net
ARDUINO & RASPBERRY PI GSM (VOICE/SMS/GPRS/DTMF) shield (nano)

1. HARDWARE

1.1 h-nanoGSM PIN DEFINITION

POWERING, SERIAL and CONTROL INTERFACE
In the left edge of the top PCB side, bottom to top:
1. RX(TXD) - GSM SHIELD SERIAL RX (TXD) - input
2. TX(RXD) - GSM SHIELD SERIAL TX (RXD) - output
3. NC – not connected
4. ON /OFF - POWER ON / OFF - input, active LOW*
5. Vusb - POWER PIN - output +5V (USB +5V)
6. Vin - POWER PIN - input +5V for LiPol charger only
7. Vcc - POWER PIN - input/output +4V**
8. GND - POWER and DIGITAL GROUND
9. RI - GSM SHIELD RING INDICATOR - output
10. STATUS - GSM SHIELD STATUS - output
11. RTS - GSM SHIELD READY TO SEND - output

* min. 200msec. pulse
* read POWERING SECTION, bellow

BATTERY and ANTENNAS
In the right edge of the top PCB side, bottom to top:
1. + LiPol - connect + pole of the LiPol battery
2. + LiPol - connect - pole of the LiPol battery

In the bottom edge of the top PCB side, left to right:
1. BTH 3.0 antenna connector - uFL
2. GSM antenna connector - uFL or SMA F

AUDIO INTERFACE and BACKUP BATTERY**
In the top edge of the top PCB side, left to right:
1. GND - GND connection **
2. AGND - Analog GND for analog audio
3. SPK+/SPKP - loudspeaker output + pole***
4. SPK-/SPKN - loudspeaker output - pole***
5. MIC-/MICN - cap. microphone input - pole
6. MIC+/MICP - cap. microphone input + pole
8. BkBat+ : Backup battery + pole****
9. BkBat- : Backup battery - pole
*** pin1 on AUDIO interface can be used as secondary GND
connection on the board.
**** WARNING! For non rechargeable battery (eg. Silver Oxide coin
cell): insert one diode (1N4148) between the battery plus pole and the
GSM shield "BkBat+" pad. 1N4148 anode must be connected with the
battery.

SWITCHES
In the bottom right corner of the top PCB side:
1. RESET - GSM SHIELD RESET

SIM SOCKET AND USB PORT
In the bottom PCB side, left to right:
1. SIM SOCKET
2. 2. USB PORT - micro USB type A – used for <u>POWERING</u> and as
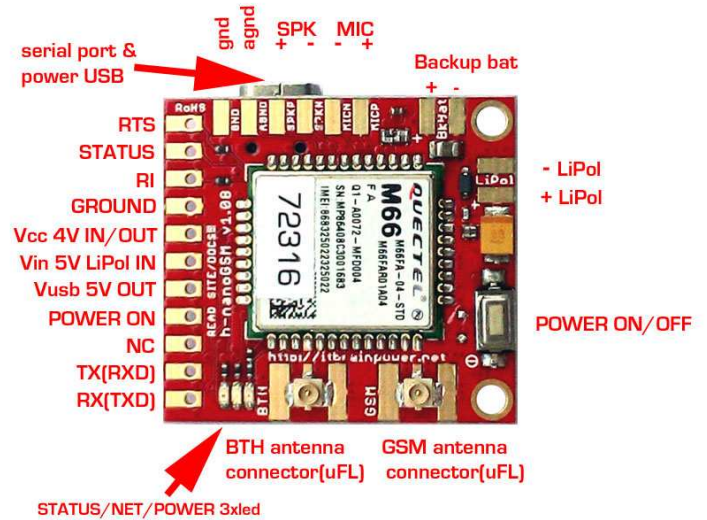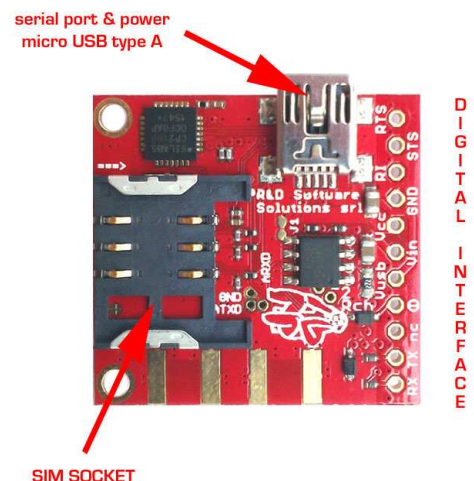   <u>SERIAL to USB bridge</u> adapter

Image 1(TOP)

Image 1(bottom)

1.2 LOGICAL WIRRING h-nanoGSM shield

1.2.1 General information

h-nanoGSM digital interfaces:

PIN1 RX(TXD) – SERIAL RX (TXD) - input
PIN2 TX(RXD) - SERIAL TX (RXD) - output
PIN3 NC – not connected
PIN4 ON /OFF - POWER ON / OFF - input, active LOW
PIN8 GND - POWER and DIGITAL GROUND
PIN9 RI - GSM SHIELD RING INDICATOR - output
PIN10 STATUS - GSM SHIELD STATUS - output
PIN11 RTS - GSM SHIELD READY TO SEND - output

are 5V, 3V and 2.8V compliant (*you can wire/interface them directly to your Arduino micro-controller / board*).
WARNING: WIRING the h-nanoGSM-gsm board with u-controllers/boards must be made with boards UNPOWERED!!

h-nanoGSM POWERING pins:

PIN5 Vusb - POWER PIN - output +5V (USB +5V)*
PIN6 Vin - POWER PIN - input +5V for LiPol charger only
PIN7 Vcc - POWER PIN - input/output +4V**
PIN8 GND - POWER and DIGITAL GROUND
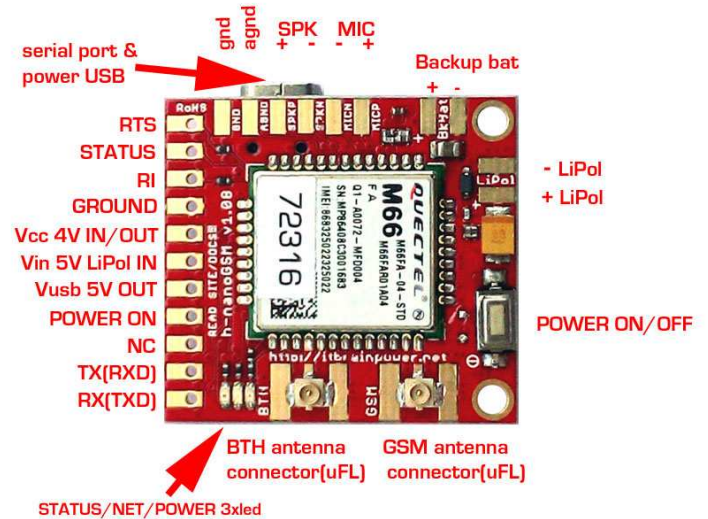
*   WARNING: PIN5 (Vusb) it is directly connected to the 5V USB bus!!!
** READ POWERING SCHEMAS/OPTIONS bellow(chapter 1.3)

Image 1(top)

1.2.2 Digital PINS connection correspondence with Arduino and Raspberry PI

h-nanoGSM shield PIN NAME	UNO/MINI/NANO (Mega328)	MEGA2560 using software serial	DUE/MEGA2560 using hardware serial	Raspberry PI B+, PI ZERO, PI II, PI 3
1. RX(TXD)	D2 rx soft	D10 rx soft	D19(RX1)	PIN10 RXD0 *
2. TX(RXD)	D3 tx soft	D3 tx soft	D18(TX1)	PIN08 TXD0 *
3. RESET	~~D6~~	~~D6~~	~~D6~~	~~PIN18~~
4. POWER ON/OFF	D7	D7	D7	PIN16
8. GND	GND	GND	GND	PIN04 or 14
10. STATUS	D5	D5	D5	PIN 12

* Raspberry PI: do not wire 1 and 2 (serial TX and RX) if USB communication is used!
* Raspberry PI: **i-hat3GGSM** adapter usage it is recommended!

1.2.3 Connecting with UNO, ArduinoPro Mini / Arduino Micro

Image 2

In this case, SoftwareSerial (Arduino environment emulates in software a second serial port) configuration it's the only option available. The Arduino USB connector it is placed in the right side of the Arduino image.

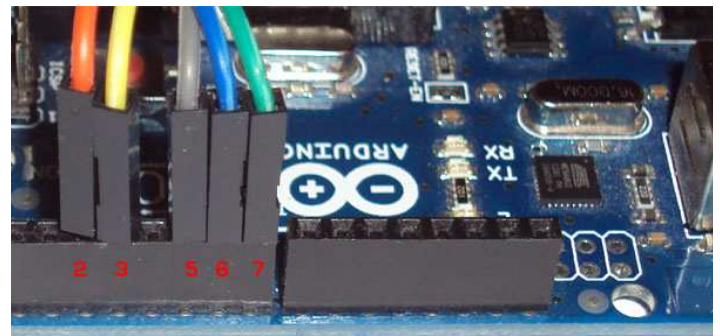h-nanoGSM PIN1(RX[TXD]) ←(ORANGE*) WIRE➡ Arduino PIN D2(RX soft)
h-nanoGSM PIN2(TX[RXD]) ←(YELLOW*) WIRE➡ Arduino PIN D3(TX soft)
h-nanoGSM PIN3(NC) ←(BLUE*) WIRE➡ Arduino PIN D6 (no need to wire/represented just for compatibility!)
h-nanoGSM PIN4(POWER ON/OFF) ←(GREEN*) WIRE➡ Arduino PIN D7
h-nanoGSM PIN8(GND) ←(BLACK* **) WIRE➡ Arduino GND
h-nanoGSM PIN10(STATUS) ←(GREY*) WIRE➡ Arduino PIN D5
*COLORS corresponding with example images.
** GND wire (BLACK) not shown in right image, must be inserted in Arduino header marked as GND
JUMPER placed between PIN5[Vusb] and PIN6[Vin] must NOT BE installed! It is valid ONLY for "WITH Lithium Polymer battery" configuration and POWERING via USB schema!!!

Arduino Micro, Arduino Nano USB, ArduinoPro Mini: **j-328GSM3GLader** adapter usage it is recommended

1.2.4 Connecting with MEGA2560 (SOFTWARE SERIAL mode)
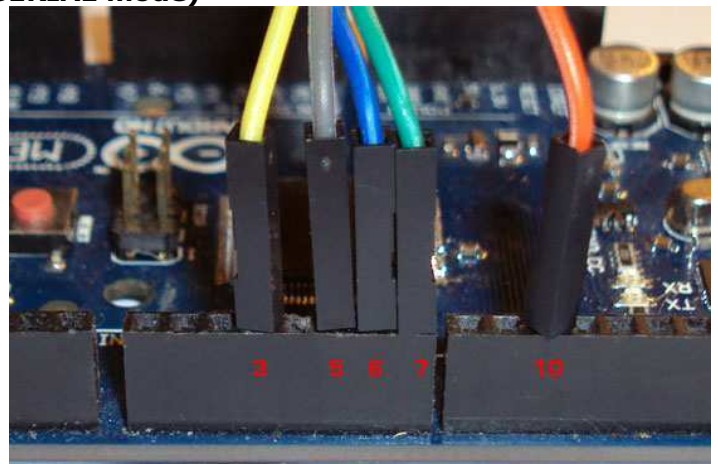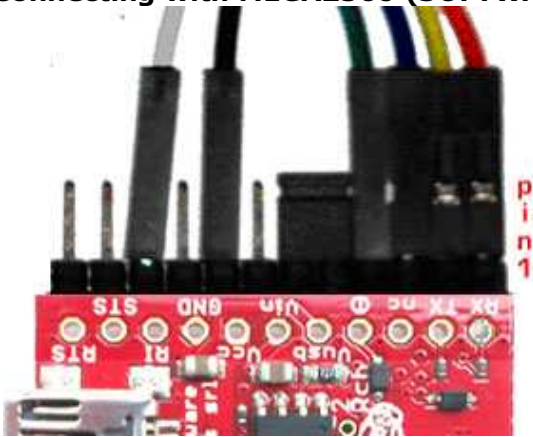
Image 3

SoftwareSerial Mode (RX soft=>10,TX soft=>3) –only ORANGE has changed comparing with UNO(Image2).
The Arduino USB connector it is placed in the right side of the Arduino image.

h-nanoGSM PIN1(RX[TXD]) ←(ORANGE*) WIRE➡ Arduino PIN D10(RX soft)
h-nanoGSM PIN2(TX[RXD]) ←(YELLOW*) WIRE➡ Arduino PIN D3(TX soft)
h-nanoGSM PIN3(NC) ←(BLUE*) WIRE➡ Arduino PIN D6(no need to wire/represented just for compatibility!)
h-nanoGSM PIN4(POWER ON/OFF) ←(GREEN*) WIRE➡ Arduino PIN D7
h-nanoGSM PIN8(GND) ←(BLACK* **) WIRE➡ Arduino GND
h-nanoGSM PIN10(STATUS) ←(GREY*) WIRE➡ Arduino PIN D5
*COLORS corresponding with example images.
** GND wire (BLACK) not shown in right image, must be inserted in Arduino header marked as GND
JUMPER placed between PIN5[Vusb] and PIN6[Vin] must NOT BE installed! It is valid ONLY for "WITH Lithium Polymer battery" configuration and POWERING via USB schema!!!

1.2.5 Connecting with DUE or MEGA2560 (HARDWARE SERIAL mode)
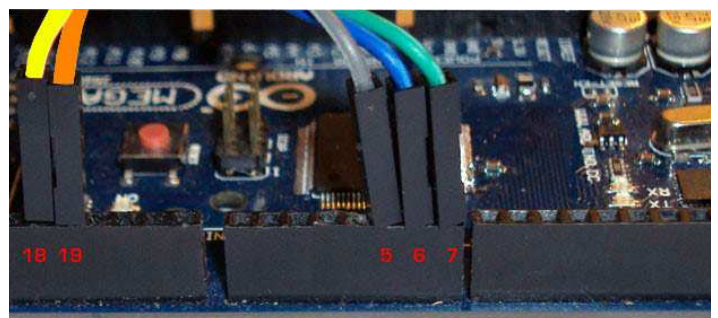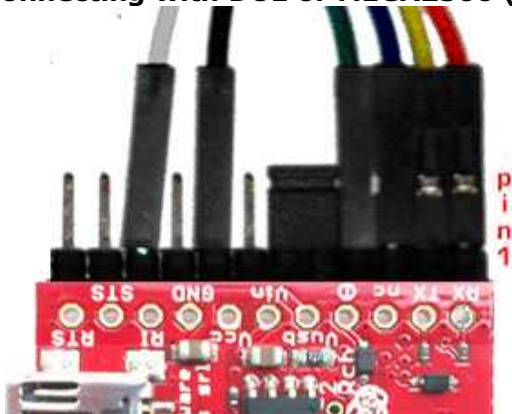

Image 4

HardwareSerial Mode ***(RX1=>19,TX1=>18) –only <u>ORANGE and YELLOW wires has changed</u> comparing with UNO(Image2). The Arduino USB connector it is placed in the right side of the Arduino image.

h-nanoGSM PIN1(RX[TXD]) ←(ORANGE*) WIRE➔ Arduino PIN D19(RX1)

h-nanoGSM PIN2(TX[RXD]) ←(YELLOW*) WIRE➔ Arduino PIN D18(TX1)

h-nanoGSM PIN3(NC) ←(BLUE*) WIRE➔ Arduino PIN D6(<u>no need to wire</u>/represented just for compatibility!)

h-nanoGSM PIN4(POWER ON/OFF) ←(GREEN*) WIRE➔ Arduino PIN D7

h-nanoGSM PIN8(GND) ←(BLACK* **) WIRE➔ Arduino GND

h-nanoGSM PIN10(STATUS) ←(GREY*) WIRE➔ Arduino PIN D5

*COLORS corresponding with example images.

** GND wire (BLACK) not shown in right image, must be inserted in Arduino header marked as GND

*** add/un-comment following preprocessor definition in software examples: #define HARDWARESERIAL

<u>JUMPER placed between PIN5[Vusb] and PIN6[Vin] must NOT BE installed!</u> **It is valid <u>ONLY</u> for "<u>WITH</u> Lithium Polymer battery" configuration and <u>POWERING via USB schema!!!</u>**

1.2.6 Connecting with Raspberry PI

Please see: http://itbrainpower.net/images/GSM-SHIELD-RPI-logical-wiring-h-nanoGSM.png

See also chapter 1.2.2 and chapter 1.4.

1.3 POWERING the h-nanoGSM shield

1.3.1 General powering information

In the right edge of the top PCB side:

PIN5   Vusb - POWER PIN - output +5V (USB +5V)*

PIN6   Vin - POWER PIN - input +5V for LiPol charger only**

PIN7   Vcc - POWER PIN - input/output +4V***

PIN8   GND - POWER and DIGITAL GROUND

In the right edge of the top PCB side, bottom to top:

1. + LiPol - connect + pole of the LiPol battery****

2. + LiPol - connect - pole of the LiPol battery****

*   **PIN5 (Vusb)** it is <u>directly connected</u> to the **5V USB bus**! Use it ONLY if shield it is **used in "<u>WITH</u> Lithium Polymer battery" configuration <u>and VIA USB</u> powering. Else leave it unconnected.**

**   **PIN6 (Vin)** INPUT +5V <u>for LiPol charger</u>. Must be connected to +5V source <u>ONLY</u> if shield it is **used in "<u>WITH</u> Lithium Polymer battery" configuration.**
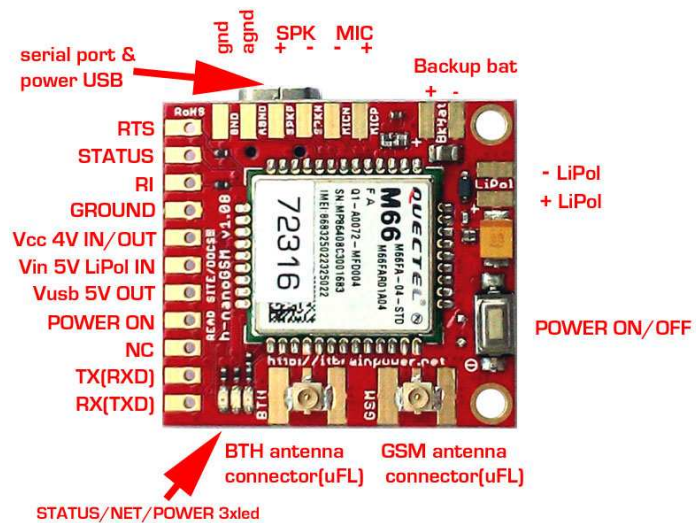

Image 5

*** PIN7 (Vcc) role / behavior:
   a. **INPUT POWER +4V typical** when the shield it is **used in "<u>WITHOUT</u> Lithium Polymer battery" configuration.** Use power supply (switching power supply is recommended) capable of voltage between **3.6V->4.2V and current more than 1.5A -eg.: our gSPS101#4V(DDRV).**
   b. **OUTPUT POWER** when the shield it is **used in "<u>WITH</u> Lithium Polymer battery" configuration.** In this case, Pin7 it is connected to the **+pole** of the **LiPol** battery and it is capable to **provide power (with typical voltage between 3.4->4.1V, but depending on used battery specifications)** to external devices. If you do not plan to power additional devices, **leave it unconnected.**

**** If you want to use a Lithium Polymer battery, we recommend to you to solder a PCB JST 2POLE connector. <u>Take care at battery connector polarity!</u>

SPECIAL SAFETY WARNINGS:
-    **WIRRING MUST BE MADE WITHOUT BOARD/BOARDS POWERED!**
-    **DO NOT PLACE THE POWERING PINS IN SHORT CIRCUIT!**
-    **DOUBLE CHECK THE CONNECTIONS BEFORE POWER THE CIRCUIT!**
-    **DO NOT INVERCE POLARITIES!**
-    **USE ONLY Lithium Polymer battery EQUIPED WITH SHORT-CIRCUIT, OVER-CHARGING AND REVERSE POLARITY PROTECTION!**
-    **SIM CARD INSERION/REMOVAL MUST BE MADE BOARD/BOARDS UNPOWERED!**

BAD or UNPROPER WIRING/HANDLING/USAGE of the hardware can conduct to SERIOUS DAMAGES! You are the ONLY RESPONSIBLE for hardware handling, usage and wiring!!!

1.3.2 "Portable stand-alone Lithium Polymer battery" powering schema

Just plug a fully charged Lithium Polymer battery in to the LiPol battery connector. DO NOT connect/wire **Pin6 [Vin]**, **Pin7 [Vcc]** and **Pin5 [Vusb]** (leave them unconnected)! The shied can be started pressing for more than 200ms on the "POWER ON" switch or by external logic (see chapter **1.2** and software examples).

Observation: the USB communication function (SERIAL to USB bridge) can be used even in this case.

1.3.3 "WITH Lithium Polymer battery" configuration and VIA Vin (+5V) powering schema

Leave unconnected: **Pin7 [Vcc]** and **Pin5 [Vusb]**.
Wire/connect:

> **Pin6[Vin]** ←WIRE→ +5V source (Eg.: UNO +5V header or +5V connector on power supply, other…)
> **Pin8[GND]** ←WIRE→ GND (Eg.: UNO GND header or GND connector on power supply, other…)

Finally, plug a Lithium Polymer battery in to the LiPol battery connector.

Observation:
- the USB logical communication (SERIAL to USB bridge) can be used even in this case.
- the USB 5V power line it is **SEPARATED** from the shield/controller.
- we recommend the usage of g-SPS 5V adapter board v1.01 by itbrainpower.net [P/N gSPS101#5V(LiPOL)]

1.3.4 "WITH Lithium Polymer battery" configuration and VIA USB powering schema

Leave unconnected: **Pin7 [Vcc]**.
Connect/place jumper between:

> **Pin6[Vin]** ←JUMPER→ **Pin5 [Vusb]** (see jumper in **Image4**)

Finally, plug a Lithium Polymer battery in to the LiPol battery connector.

Observation: the USB logical communication (SERIAL to USB bridge) can be used. The USB 5V power line **POWERS** the shield/controller.

1.3.5 "ONLY Lithium Polymer battery charger" configuration

Same as **1.3.3 [VIA Vin +5V powering)]** or **1.3.4 [VIA USB powering]**, but with any other logical interfacing / wire connections (Eg.: SERIAL, RESET, other..) removed. The h-nanoGSM shield acts as regular Lithium polymer battery charger. It is NOT necessary to start the shied from the "POWER ON" switch.

1.3.6 "WITHOUT Lithium Polymer battery" powering schema (DDRV)

Leave unconnected: **Pin6[Vin]** and **Pin5 [Vusb]**.
Wire/connect:

> **Pin7 [Vcc]** ←WIRE→ +4V source (Eg.: +4V connector on power supply, other…)
> **Pin8[GND]** ←WIRE→ GND (Eg.: GND connector on power supply, other…)

DO NOT PLUG (UNPLUG ANY) Lithium Polymer battery in to the LiPol battery connector!

Observation:
- the USB logical communication (SERIAL to USB bridge) can be used even in this case.
- the USB 5V power line it is **SEPARATED** from the shield/controller.
- used power supply (switching power supply recommended) must be supplz voltage between 3.6V->4.2V, 4V recommended, and to drive 0.8A and capable fore more than 1.5A in pulses/spikes.
- we recommend the usage of g-SPS 4V adapter board v1.01 by itbrainpower.net [P/N gSPS101#4V(DDRV)]

1.4 SERIAL (and SERIAL USB bridge) communication

Your h-nanoGSM shield came with special auto-baud feature enabled (the speed of communication is determined at first AT command detection w/o external intervention) and 8N1 settings. The board works off the shelf with your code, without the needs for setting the module communication seed first! This feature performs in same manner for TXD/RXD SERIAL connection over 2.8-3-5V compliant interface and in connection over USB.

Communication speed performed in code examples:
- h-nanoGSM series GSM (VOICE/SMS/GPRS/DTMF) shield (nano) examples list >> 9600bps, 8N1 *
- h-nanoGSM series GSM (VOICE/SMS/GPRS/DTMF) shield (nano) Raspberry PI shield examples>>19200bps, 8N1
- h-nanoGSM series software kickstart for Arduino >> 9600bps, 8N1 **
- h-nanoGSM-raspian-ppp-1.0.tar.gz (Raspian PPP and routing utility) >> 115200bps, 8N1

IMPORTANT! ARDUINO ENVIRONMENT Serial Monitor speed:

*, ** 57600bps

Raspberry PI connectivity support:

RPI 3, Noobs & Jessy 2016 version support >> Read the PROJECTS / HOW TO section, here:
http://itbrainpower.net/a-gsm/gsm-shield-Arduino-RaspberryPI-projects
RPI via SERIAL connection >> Use vi, mcedit or other editor and comment last "/etc/inittab" line:
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
reboot your RPi: "reboot" or "restart now"
RPI via USB connection >> Native supported starting with 2015-02-16-raspbian-wheezy. If something goes wrong, you can still use the 2015-02-16 version, used by us in tests. You can download it from here:
http://itbrainpower.net/a-gsm/downloadables/2015-02-16-raspbian-wheezy.zip

1.5 SIM card support

Standard size 1.8 / 3V GSM SIM [2G] card it is supported [see **Image 1(bottom)**].
WARNING: SIM cards insertion/removal must be made with boards UNPOWERED!!

1.6 FLASHDISK and RAMDISK support

The internal FLASHDISK and RAMDISK is accessed "via AT commands" and does not need the load for additional libraries and is limited to read/write/delete small files that provides support for "on the run" save/access application parameters and variables (For RAMDISK, one suggestion: save local data, when your project needs to sleep-low energy mode-using AT+CFUN=4).

1.7 LEDs

h-nanoGSM shield it is equipped with 3 LEDs (yellow – STATUS, GREEN for GSM MODE and RED for LiPol charging) that signalize the working state of the GSM modem.
STATUS LED (yellow) lights when the M95 modem is ON.
GSM MODE LED (green) blinks twice per second when the M95 it is NOT registered into the GSM network and blinks around 1.5 seconds when the modem is registered.
LiPol CHARGING (red) lights on when LiPol battery is charging.

HINTS:
A. if the STATUS LED blinks (once at 4-10 seconds) or does not lights on shield POWERON.This may happen if:
   a. "WITHOUT Lithium Polymer battery" powering schema (DDRV) as described in chapter 1.3.6 it is used, and your power supply it is not capable to provide the current needed. You may check / change your power supply or connecting wires or solders or you can try to solder one 470->2200uF / 6V Low ESR electrolytic capacitor on LiPol pads (take care on the polarity).
   b. Any "WITH Lithium Polymer battery" powering schema, if the battery it is very / deep discharged, and the +5V source connected to Vin can provide only small amount of current. You may wait some time in order to give to LiPol battery the chance to be charged a little bit more, or you may change the +5V source connected to Vin with one capable to provide more current.
B. if the GSM MODE LED does NOT acquire the "GSM register" mode (blinks around twice per second). This may happen on following reasons:
   a. SIM's PIN code has not been removed. Remove the SIM PIN code checking.
   b. The GSM signal is to low at the working place. Try to relocate your board, or replace the antenna with a better one.
   c. Your GSM provider has no service at the working place. Change with SIM from other GSM provider.
   d. Your SIM has no credit. Add some credit to your SIM.

1.8 Audio interface

Audio output => 32 ohm.

High sensitivity capacitor microphone interface.
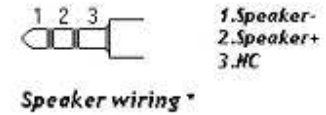
HEALTH AND SAFETY WARNING!!!!!!!!!!
You can damage your years! Use it with care when headsets are connected.
We recommend to use audio setup commands in order to limit the output power. Use AT+CLVL=20 (as maximum value).

1. Mic+
2. NC
3. Mic-

Microphone wiring *

1.Speaker-
2.Speaker+
3.NC

Speaker wiring *

HEALTH AND SAFETY WARNING!
HEADSET USAGE CAN DAMAGE YOUR EARS!
High power audio output!

Image 6

Audio wiring
h-nanoGSM AudioPIN1(GND) ⬅SOLDER➔ h-nanoGSM AudioPIN2(AGND)*
h-nanoGSM AudioPIN3(SPK*) ⬅WIRE➔ Speaker + pole
h-nanoGSM AudioPIN4(SPK+) ⬅WIRE➔ Speaker - pole
h-nanoGSM AudioPIN5(MIC-) ⬅WIRE➔ capacitor microphone - pole
h-nanoGSM AudioPIN5(MIC+) ⬅WIRE➔ capacitor microphone + pole
* you may try to connect here the microphone wire shielding.
* Audio interfacing: f-audioOutputs - audio extender board usage it is recommended

2. Debugging (your) AT CHAT
2.1 Preparation phase
A1. Download USB 2 SERIAL BRIDGE DRIVERS (choose the adapted one to your computer operating system CP2102 driver):
http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpdrivers.aspx
A2. Download serial terminal software. We like TeraTermPro (google it for).
A3. Connect the h-nanoGSM board with USB cable to your computer. Follow install procedure, if needed.
A4. Open the terminal application and connect to the h-nanoGSM com port
A5. Set the proper communication speed: 9600bps, 8N1
A7. Check desired AT commands.
A6. Close the terminal software. Disconnect the cable.

2.2 Via USB program AT chat debugging how-to
1. Connect the USB cable to your computer.
2. Open the serial (com/tty) terminal application
3. Check if proper serial port (described in A4) is chosen in the terminal software
4. Check the communication speed (9600, 8N1)
From this moment, you will be able debug the AT exchange between your prefered u-controller and the h-nanoGSM board. Run your Arduino/RaspberryPI code. The AT exchange will be replicated into the terminal window.
You can also use the precompiled directive
#define atDebug
in the code based on our libraries/examples. In this way, the AT CHAT between the Arduino u-controller and h-nanoGSM shield it is debugged on the Arduino "Serial Monitor". You may like to take a look at sendATCommand and recUARTDATA functions defined in hnanoGSM_basic_lbr.ino.

3. Software examples and code support

3.1 h-nanoGSM series GSM (VOICE/SMS/GPRS/DTMF) shield (nano) kickstart for Arduino

<u>Interactive interface</u> with your h-nanoGSM shield (nano). You can dial, pick up, hang up calls, read, delete or send SMSs, see the signal strength, read/write the RTC(real time clock), enable / disable the synchronization of the RTC, read modem serial(IMEI), SIM serial(IMSI), GSM and GPRS registration status, perform DTMF tasks, GET and POST (<u>with</u> or without <u>SSL encryption</u>) requests and even interact with the modem trough AT commands, directly from the application. More features will be added (your contribution will be welcomed!).

Download from (module's IMEI & your email address needed):
http://itbrainpower.net/downloads#h-nanoGSM
Uncompress the archive file into suggested folder("hnanoGSM_kickstart*version*"). Dive to that folder. Do not remove any file fro the folder. Double click on the "hnanoGSM _kickstart*version*.ino" file. ARDUINO environment will start. Chose your Arduino board, serial communication port. Compile the sketch and upload it to your board.
Open the ARDUINO "Serial Monitor" and chose 57600bps speed.
IMPORTANT!
Before compiling, select in the Arduino environment, for editing, "user_GPRS_HTTP_PARS.h" file. You may want to change the APN related settings in order to fit your GSM provider settings.

Be sure you have made proper setup for your board and h-nanoGSM shield. Used SIM card must have the PIN checking procedure removed.

3.2 h-nanoGSM series GSM (VOICE/SMS/GPRS/DTMF) shield (nano) ARDUINO examples list

Compliable code (IP DATA TRANSFER w or wo SSL, DTMF, SMS, CALL handling, file handling, and other) examples for your h-nanoGSM board and Arduino. Can be used as foundation starter for your GSM projects. Compile and running directives inside the code and associated txt files. Set the ARDUINO "Serial Monitor" at 57600bps. Speed settings for debug via USB (if desired): 9600bps, 8N1.

Download from (module's IMEI & your email address needed):
http://itbrainpower.net/downloads#h-nanoGSM

3.3 h-nanoGSM series GSM (VOICE/SMS/GPRS/DTMF) shield (nano) RPI examples list

Running code (IP DATA TRANSFER w or w/o SSL, DTMF, SMS, CALL handling, file handling, and other) examples for your h-nanoGSM board and Raspberry PI. Can be used as foundation starter for your GSM projects. Compile and running directives inside the code and associated txt files.
Speed settings for debug via USB (if desired): 19200bps, 8N1.
IMPORTANT:
- <u>some py files needs to run under root permission</u>. You can try something like: sudo python xyz.py
- <u>you may chose between SERIAL and USB communication</u>, in order to fit to your hardware interfacing option (see inside py files)

Download from (module's IMEI & your email address needed):
http://itbrainpower.net/downloads#h-nanoGSM

3.4 Arduino GSM class hack

Run (almost) any project written 4 Arduino GSM using the h-nanoGSM shield. Download from:
http://itbrainpower.net/downloads#h-nanoGSM

3.5 Quectel M66F AT command manual

Download from:
http://itbrainpower.net/downloads#h-nanoGSM_documentation

4 General/legal software/hardware considerations

Write us back about your projects and/or software contribution! We will be more than pleased to publish your hardware/software projects on our site or to include your contribution into the support files!

The support SOFTWARE is provides "AS IS" in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Please read carefully and respect the license agreement published in downloaded software.

BAD or UNPROPER WIRING/HANDLING/USAGE of the hardware can conduct to SERIOUS DAMAGES! You are the ONLY RESPONSIBLE for hardware handling, usage and wiring!!!