kick-start for l-LTE 1.07 by itbrainpower.net
ARDUINO, BEAGLEBONE & RASPBERRY PI 4G LTE +GNSS shield

kick-start for l-LTE 1.07 by itbrainpower.net

ARDUINO, BEAGLEBONE & RASPBERRY PI 4G LTE +GNSS shield

1. HARDWARE

1.1 l-LTE PIN DEFINITION

POWERING, SERIAL and CONTROL INTERFACE
In the left edge of the top PCB side, top to bottom:
1. RX(TXD) - 4G SHIELD SERIAL RX (TXD) - input
2. TX(RXD) - 4G SHIELD SERIAL TX (RXD) - output
3. RESET - 4G SHIELD RESET - input, active HIGH*
4. POWER ON - 4G SHIELD POWER ON - input, active HIGH*
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 - 4G SHIELD RING INDICATOR - output
10. STATUS - 4G SHIELD STATUS - output
11. RTS - 4G SHIELD READY TO SEND - output
12. CTS - CLEAR TO SEND - input
13. SLP - SLEAP - input, active HIGH
* min. 200msec. pulse
** read POWERING SECTION, bellow

SHIELD ANTENNAS
In the right edge of the top PCB side, top to bottom:
1. 4G/LTE antenna connector - uFL or SMA F
2. GNSS [GPS+GLONASS] antenna connector - uFL

Image 1(TOP)

DIGITAL AUDIO INTERFACE
In the top edge of the top PCB side, left to right:
1-8. DIGITAL AUDIO interface**
** pin2 on DIGITAL AUDIO interface can be used as secondary GND connection on the board.

SIM and SWITCHES
In the bottom edge of the top PCB side, left to right:
1. RESET SWITCH - 4G SHIELD RESET
2. SIM SOCKET - STANDARD SIZE 1.8-3V, 4G SIM/USIM supported
3. POWER ON/OFF SWITCH - 4G SHIELD POWER ON / OFF

USB PORT, LiPO battery & AUX GNSS POWER
In the bottom PCB side, left to right:
1. USB PORT - mini USB type B - 4G SHIELD POWERING and SERIAL to USB bridge adapter

2. LiPol - connect - pole of the LiPo battery [GND]
3. LiPol - connect + pole of the LiPo battery
4. Auxiliary GNSS antenna power - may connect the positive pole of aux power supply for the GNSS antenna powering.

5. Auxiliary LiPol - connect + pole of the LiPo battery
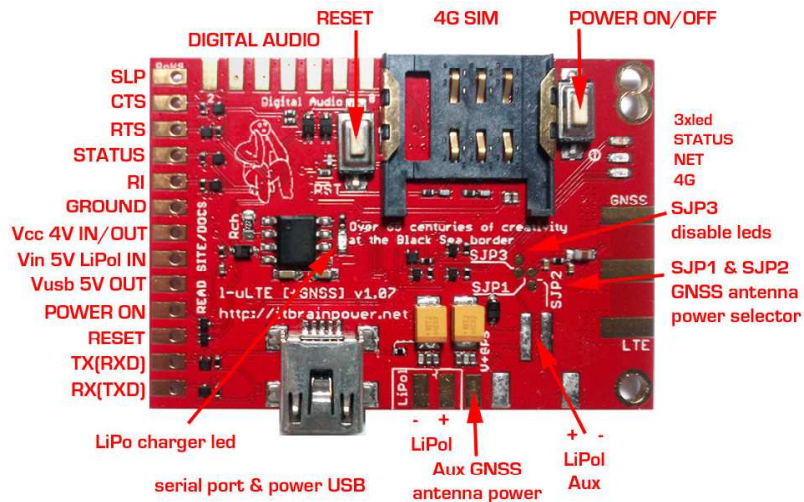6. Auxiliary LiPol - connect - pole of the LiPo battery [GND]

Image 1(bottom)

1.2 LOGICAL WIRRING l-LTE shield
1.2.1 General information
l-LTE digital interfaces:
PIN1   RX(TXD) – UART RX (TXD) - input
PIN2   TX(RXD) – UART TX (RXD) - output
PIN3   RESET - 4G SHIELD RESET - input, active HIGH
PIN4   POWER ON/OF - POWER ON/OFF - input, active HIGH
PIN8   GND - POWER and DIGITAL GROUND
PIN9   RI - 4G SHIELD RING INDICATOR - output
PIN10   STATUS - 4G SHIELD STATUS - output
PIN11   RTS - 4G SHIELD READY TO SEND - output
PIN12   CTS - CLEAR TO SEND - input
PIN13   SLP - SLEAP - input, active HIGH

are 5V and 3Vcompliant (*you can wire/interface them directly to your Arduino micro-controller / board*).

WARNING: WIRING the l-LTE-gsm board with u-controllers/boards must be made with boards UNPOWERED!!

l-LTE 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)

1.2.2 Digital PINS connection correspondence with Arduino, BeagleBone and Raspberry PI

l-LTE shield PIN NAME	UNO/MINI/NANO (Mega328)	MEGA2560 using soft-ware serial	DUE/MEGA2560 using hardware serial	Raspberry PI B+ II, Zero and 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	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!

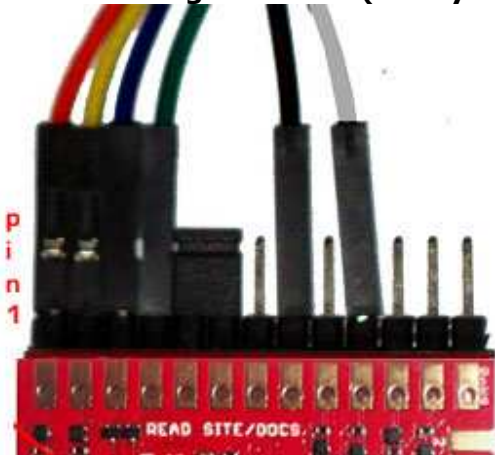### 1.2.3 Connecting with UNO (or any ATMega 328 shield)

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.

l-LTE PIN1(RX[TXD]) ←(ORANGE*) WIRE➔ Arduino PIN D2(RX soft)

l-LTE PIN2(TX[RXD]) ←(YELLOW*) WIRE➔ Arduino PIN D3(TX soft)

l-LTE PIN3(RST) ←(BLUE*) WIRE➔ Arduino PIN D6

l-LTE PIN4(POWER ON) ←(GREEN*) WIRE➔ Arduino PIN D7

l-LTE PIN8(GND) ←(BLACK* **) WIRE➔ Arduino GND

l-LTE 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!!!

1.2.4 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.

l-LTE PIN1(RX[TXD]) ←(ORANGE*) WIRE➔ Arduino PIN D10(RX soft)

l-LTE PIN2(TX[RXD]) ←(YELLOW*) WIRE➔ Arduino PIN D3(TX soft)

l-LTE PIN3(RST) ←(BLUE*) WIRE➔ Arduino PIN D6

l-LTE PIN4(POWER ON) ←(GREEN*) WIRE➔ Arduino PIN D7

l-LTE PIN8(GND) ←(BLACK* **) WIRE➔ Arduino GND

l-LTE 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!!!

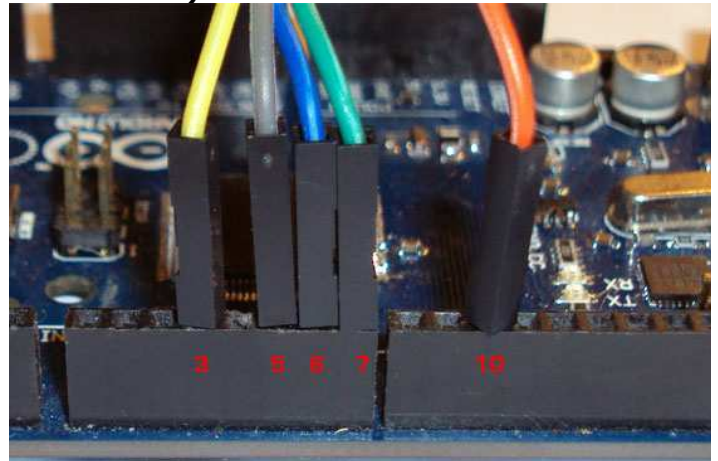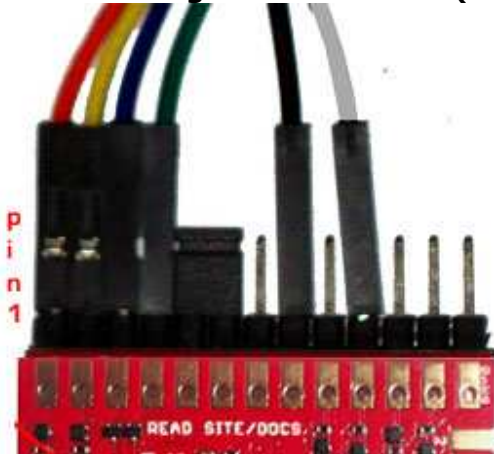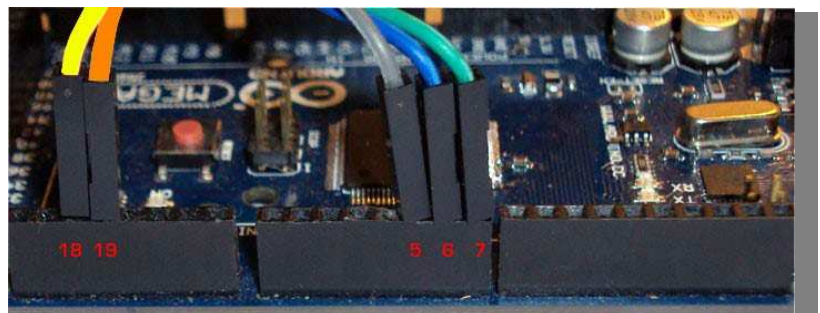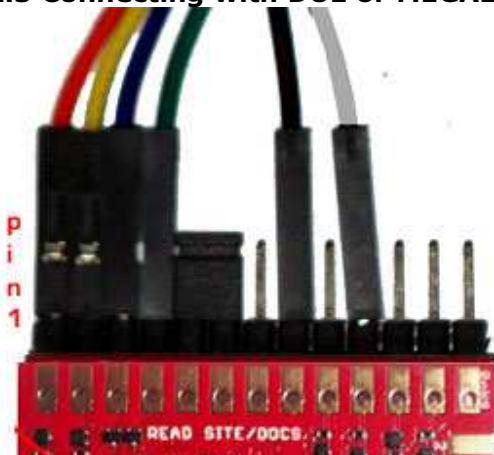## 1.2.5 Connecting with DUE or MEGA2560 (HARDWARE SERIAL mode)

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.

l-LTE PIN1(RX[TXD]) ←(ORANGE*) WIRE→ Arduino PIN D19(RX1)

l-LTE PIN2(TX[RXD]) ←(YELLOW*) WIRE→ Arduino PIN D18(TX1)

l-LTE PIN3(RST) ←(BLUE*) WIRE→ Arduino PIN D6

l-LTE PIN4(POWER ON) ←(GREEN*) WIRE→ Arduino PIN D7

l-LTE PIN8(GND) ←(BLACK* **) WIRE→ Arduino GND

l-LTE 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 and BeagleBone Black

RPIx no wires how to using i-HAT adaptor:
http://itbrainpower.net/a-gsm/i-hatGSM3G_d-u3G_c-uGSM_shield_howto_start_tutorial.php

RPIx connection reference schema using wires:
http://itbrainpower.net/images/4G-SHIELD-RPI-logical-wiring-l-LTE.png

NOOBS RPI 3 HOW TO:
http://itbrainpower.net/a-gsm/NOOBS-RASPBERRY-PI-3-GSM-3G-SERIAL-HOW-TO

DEBIAN JESSY RPI 3 HOW TO:
http://itbrainpower.net/a-gsm/RASPBIAN-JESSIE-RASPBERRY-PI-3-GSM-3G-SERIAL-HOW-TO

BeagleBone Black & itbrainpower modular modems how to:
http://itbrainpower.net/a-gsm/BBB-gsm-how-to

1.3 POWERING the l-LTE 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.**
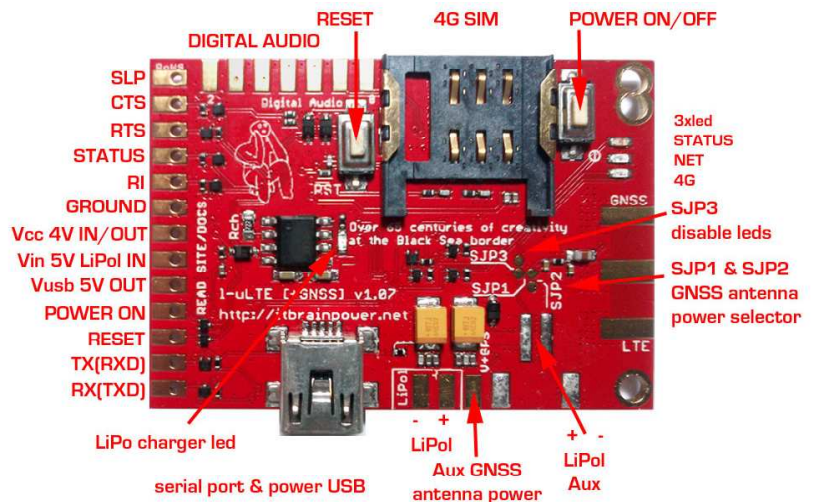** **PIN6 (Vin)** INPUT +5V <u>for LiPol charger</u>. Must be

Image 5

connected to +5V source <u>ONLY</u> if shield it is **used in "<u>WITH</u> Lithium Polymer battery" configuration.**
*** PIN7 (Vcc) role / behavior:
    a. **<u>INPUT POWER +4V typical</u>** 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 <u>more than</u> 1.5A -eg.: our gSPS101#4V(DDRV).**
    b. **<u>OUTPUT POWER</u>** 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!**

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.

Hints:
- 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)]
- the Lithium Polymer battery <u>can be replaced with one super-capacitor</u>, bigger than 1F, rated for more than 5V and having ERS lower than 250mOhm. We've made tests using SCMT22C505MRBA0 from AVX and PM-5R0H105-1 from POWERSTOR/EATON.

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 l-LTE 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 supply 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 l-LTE shield came with 115200bps default UART speed. In our l-LTE series 4G LTE shield kickstart for Arduino software, we've implemented, in code, the speed change. The USB port supports the auto-baud feature.

Communication speed performed in code examples:
- l-LTE series 4G LTE shield examples list >> 9600bps, 8N1 *
- l-LTE series 4G LTE shield Raspberry PI shield examples >> 19200bps, 8N1
- l-LTE series software kickstart for Arduino >> 9600bps, 8N1 **
- l-LTE-raspian-ppp-1.0.tar.gz (Raspian PPP and routing utility) >> 115200bps, 8N1

IMPORTANT! ARDUINO ENVIRONMENT Serial Monitor speed:

* 57600bps

** 57600bps

Raspberry PI connectivity support:

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 >> download the RPI support files and install the l-LTE-usb script

1.5 SIM card support

Standard size 1.8 and 3V SIM/USIM cards usage is supported [see **Image 1(bottom)**].

WARNING: SIM cards insertion/removal must be made with boards UNPOWERED!!

HINT: SIM with 4G/LTE plan enabled must be used for 4G/LTE connectivity. Ask your operator about.

1.6 RAMDISK support

The internal 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 (one suggestion: save local data, when your project needs to sleep-low energy mode-using AT+CFUN=4).

1.7 LEDs

l-LTE shield it is equipped with 3 LEDs, in the right side (yellow – STATUS, GREEN for GSM MODE and RED – 4G/LTE connectivity) that signalize the working state of the GSM modem and one supplementary RED LED for the LiPo charging [reference **Image 1(bottom)**].

STATUS LED (yellow) lights when the UG95 modem is ON.

GSM MODE LED (green) blinks twice per second when the UG95 it is NOT registered into the network and blinks around 1.5 seconds when the modem is registered.

4G/LTE (red) lights on the modem is connected in LTE/4G mode to the network.

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 POWER ON.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 "register" mode (blinks 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 network signal is to low at the working place. Try to relocate your board, or replace the antenna with a better one.
   c. Your network provider has no service at the working place. Change with SIM/USIM from other network provider.
   d. Your SIM has no credit. Add some credit to your SIM.

1.8 Solder jumpers

Observe the 3 solder jumpers grouped in the right side of the board [image 6].

SJP1 [placed between SJP2 and SJP3]– enable active GNSS antenna powering from the Vcc. **CIRCUIT OPEN* default**

SJP2 – enable active GNSS antenna powering from the Aux GNSS power [independent powering]. **CIRCUIT OPEN default**

SJP3 – disable state LED activity. **CIRCUIT CLOSED** default**

*   **CIRCUIT OPEN - not connected**
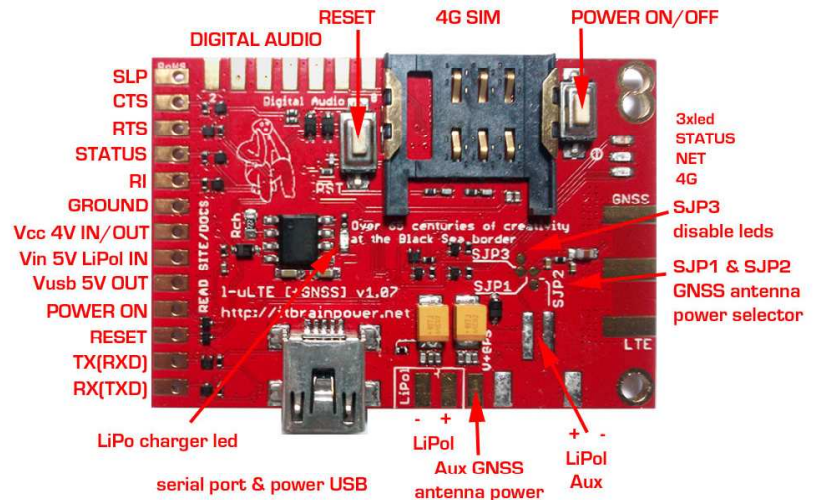** **CIRCUIT CLOSED - connected**

IMAGE 6

WARNINGS:
-   DO NOT MODIFY SOLDER JUMPERS CONFIGURATION UNTIL YOU REALLY KNOW WHAT YOU ARE DOING!
-   SPECIAL WARNING FOR SJP1 and SJP2! DO NOT ACTIVATE TOGETHER SJP1 AND SJP2 [as CIRCUIT CLOSED]! WRONG CONFIGURATION MAY DAMAGE YOUR power supplies, GNSS antenna and/or l-LTE shield!

HINTS:
-   DISABLE STATE LEDs: The l-LTE shield it is equipped with 3 state LEDs. In order to decrease the total current consumption, in certain [production] environments, you may want to disable those LEDs. For this you must interrupt the SJP3. Just insert a cutter between the pads of the SJP3 and press vertically [keep it vertical and stable, do not move horizontally!]. Check SJP3 for "CIRCUIT OPEN" with your millimeter.
-   ENABLE GNSS ANTENNA POWERING FROM THE l-LTE VCC [internal GNSS antenna powering]. First, CHECK your GNSS antenna datasheet, if powering up to 4.3V it is supported! If yes, proceed further. Just make solder over the JSP1 pads [close the circuit]. Make sure JSP2 it is CIRCUIT OPEN / DEFAULT
-   ENABLE GNSS ANTENNA POWERING FROM THE Aux GNSS power pad [external / independent GNSS antenna powering]. First, CHECK your GNSS antenna datasheet, and use proper power voltage! Just make solder over the JSP2 pads [close the circuit]. Make sure JSP1 it is CIRCUIT OPEN / DEFAULT. Connect the GNSS power supply plus pole to the "Aux GNSS antenna power" pad and the GNSS power supply GND to the GND pad [the pad placed in the right side of the Aux GNSS antenna power].

2. Debugging (your) AT CHAT

2.1 Preparation phase
A1. Download USB DRIVERS:
http://itbrainpower.net/4G-shield-modular-modem-l-LTE/resources.php#CODE%20EXAMPLES
A2. Download serial terminal software. We like TeraTermPro (google it for).
A3. Connect the l-LTE board with USB cable to your computer. Follow install procedure, if needed.
A4. Open the terminal application and connect to the l-LTE 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 Program AT chat debugging how-to
Use the precompiled directive:
#define atDebug
found in the code based on our libraries/examples. In this way, the AT CHAT between the Arduino u-controller and l-LTE shield it is debugged on the Arduino "Serial Monitor". You may like to take a look at sendATCommand and recUARTDATA functions defined in lLTE_basic_lbr.ino.

3. Software examples and code support

3.1 l-LTE series 4G LTE shield kickstart for Arduino

Interactive interface with you're l-LTE shield (micro). 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 (with or without SSL encryption) 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/4G-shield-modular-modem-l-LTE/resources.php#CODE%20EXAMPLES

Uncompress the archive file into suggested folder("lLTE_kickstart*version*"). Dive to that folder. Do not remove any file fro the folder. Double click on the "lLTE_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 4G provider settings.

Be sure you're made proper setup for your board and l-LTE shield. Used SIM card must have the PIN checking procedure removed.

3.2 l-LTE series 4G LTE shield ARDUINO examples list

Compliable code (IP DATA TRANSFER w or wo SSL, DTMF, SMS, CALL handling, file handling, and other) examples for you're l-LTE board and Arduino. Can be used as foundation starter for your 4G projects. Compile and running directives inside the code and associated txt files. Set the ARDUINO "Serial Monitor" at 57600bps.

Download from (module's IMEI & your email address needed):
http://itbrainpower.net/4G-shield-modular-modem-l-LTE/resources.php#CODE%20EXAMPLES

3.3 l-LTE series 4G LTE shield Raspberry PI examples list and PPP support files

Running code (IP DATA TRANSFER w or w/o SSL, DTMF, SMS, CALL handling, file handling, and other) examples for your l-LTE board and Raspberry PI. Can be used as foundation starter for your 4G projects. Compile and running directives inside the code and associated txt files.
IMPORTANT:
- some py files needs to run under root permission. You can try something like: sudo python xyz.py
- you may chose between SERIAL and USB communication, 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/4G-shield-modular-modem-l-LTE/resources.php#CODE%20EXAMPLES

3.4 l-LTE series 4G LTE shield and BeagleBone Black

Read: http://itbrainpower.net/a-gsm/BBB-gsm-how-to

3.5 Quectel EC21/EC25 AT command manual
Download from:
http://itbrainpower.net/downloadables/Quectel_EC25_EC21_AT_Commands_Manual_V1.0.pdf

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!!!