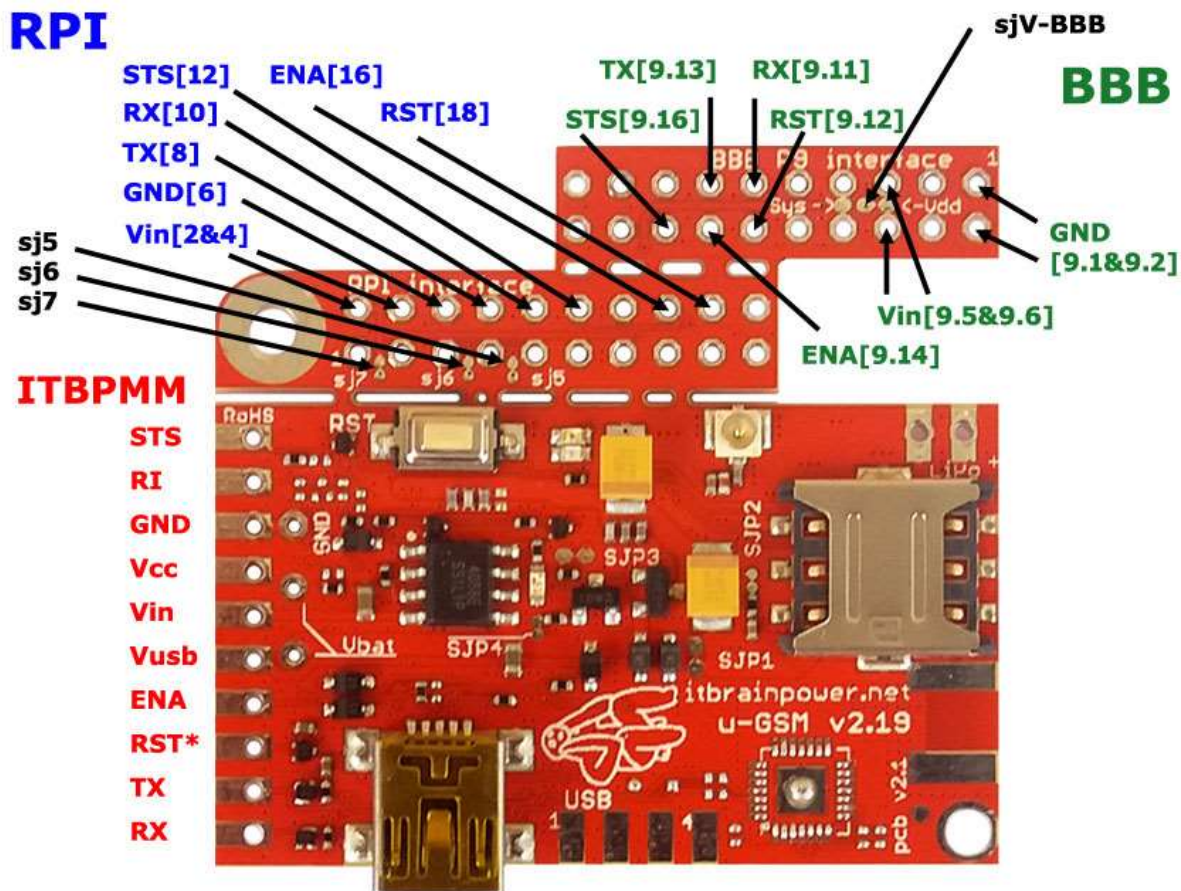


BananaPI M1+ u-GSM modem brief how to.

Reference about u-GSM interfaces:



u-GSM shield signals description:

RX - TXD - to be wired to MCU RX pin
TX - RXD - to be wired to MCU TX pin
RST - RESET modem, active HIGH level **
ENA - Enable modem power, active HIGH level
Vusb - 5V out, export USB 5V power
Vin - 5V input for Lithium Polymer battery charger
Vcc - 4V in/out, internally connected to LiPo + pad
GND - Ground
RI - Ring Indicator
STX - STATUS indicator, HIGH level if the modem is ON
** alternate functionality [ON / OFF / exit PSM mode] for BG96 variants

Solder jumpers description:

sjp5 - disconnect RX from RPI/BBB [default connected]***
sjp6 - disconnect TX from RPI/BBB [default connected]***
sjp7 - disconnect Vin from RPI/BBB 5V [default connected]
*** read about u-GSM to BBB/RPI USB connection !!

RX/TX/RST/ENA/RI/STX signals are 3-5V auto level compliant

u-GSM shield v 2.19 pinout

Raspberry PI interface:

PIN 02 - 5V RPI - wired to Vin
PIN 04 - 5V RPI - wired to Vin
PIN 06 - GND - wired to GND
PIN 08 - GPIO14[UART TX] - wired to TX
PIN 10 - GPIO15[UART RX] - wired to RX
PIN 12 - GPIO18 - wired to STX
PIN 16 - GPIO23 - wired to ENA
PIN 18 - GPIO24 - wired to RST

Beagle Bone Black interface:

P9.01 - GND - wired to GND
P9.02 - GND - wired to GND
P9.05 - VDD 5V - wired to Vin
P9.06 - VDD 5V - wired to Vin
P9.11 - GPIO_30[UART4_RXD] - wired to RX
P9.12 - GPIO_60 - wired to RST
P9.13 - GPIO_31[UART4_TXD] - wired to TX
P9.14 - GPIO_50 - wired to ENA
P9.16 - GPIO_51 - wired to STX

BBB option pins:

*P9.07 - SYS 5V - Vin [config via sjp-VBBB]
*P9.09 - SYS 5V - Vin [config via sjp-VBBB]

More u-GSM info and references:

<https://itbrainpower.net/u-GSM/resources> and <https://itbrainpower.net/u-GSM/u-GSM-part-numbers>

BananaPI M1+ u-GSM modem brief how to.

Modem preparation:

Step1.

[u-GSM modem with uFL connector]: Prepare the u-GSM modem:

- solder the 2x10 connector to u-GSM embedded RaspberryPI interface,
- solder the LiPO/LiION 1CELL battery connector or, as alternative, solder the 1F, >5V, ESR <150mOhm super-capacitor] as described in: <https://itbrainpower.net/a-gsm/u-GSM-Raspberry-PI-BeagleBone-Black-howto>

Optional, you may like to trim-down the BBB embedded interface [as described in above link].

Connect your GSM antenna to the u-GSM uFL connector.

[u-GSM modem with SMA connector]: Prepare the u-GSM modem:

- solder the LiPO/LiION 1CELL battery connector or, as alternative, solder the 1F, >5V, ESR <150mOhm super-capacitor] as described in: <https://itbrainpower.net/a-gsm/u-GSM-Raspberry-PI-BeagleBone-Black-howto>
- solder one 1x10 pin header to the ITBPMM [RX/TX/ENA/RST/Vusb/Vin/Vcc/GND/STS/] interface,

DOUBLE CHECK the soldering, wiring and polarity!!!

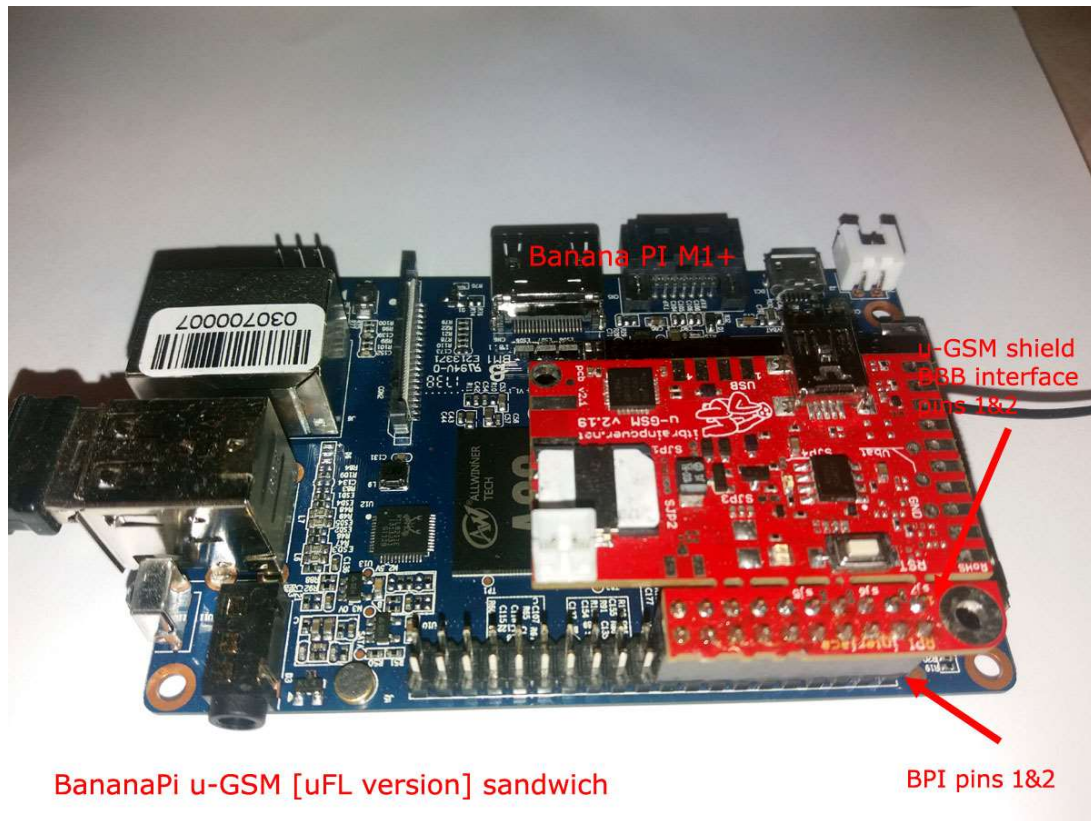
Optional, you may like to trim-down the BBB embedded interface.

Connect your GSM antenna to the u-GSM SMA connector.

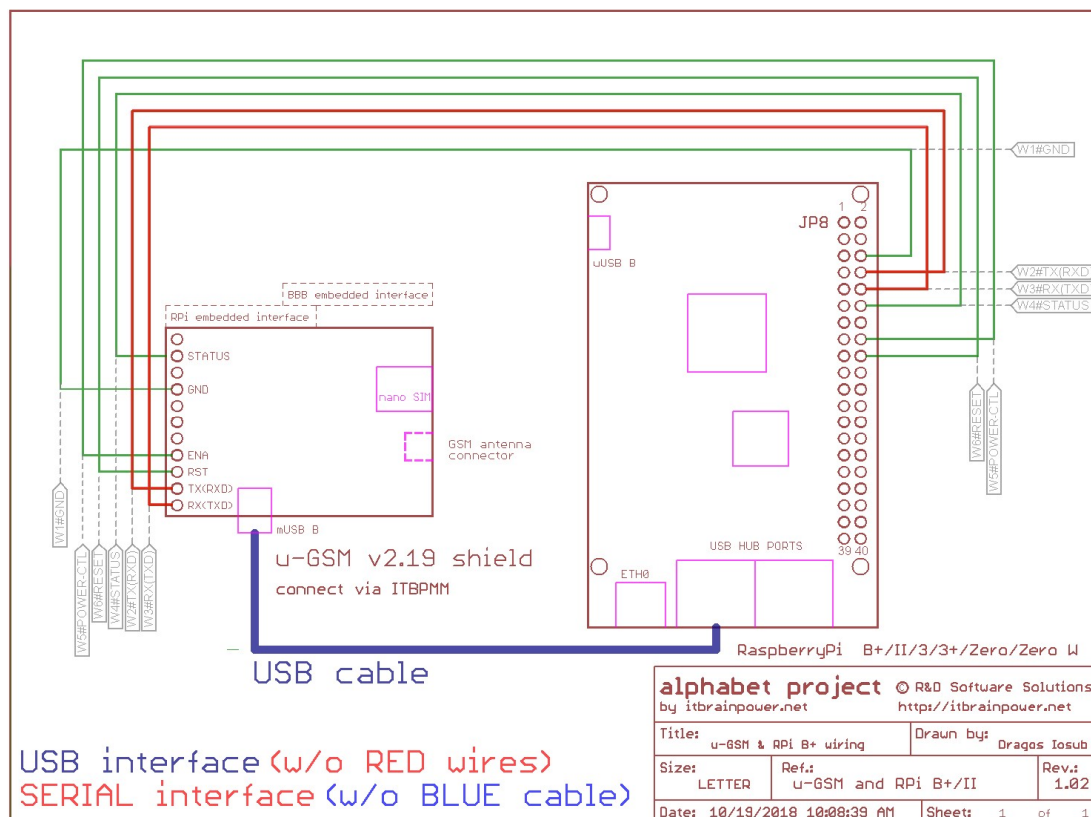
Step2. Insert the nanoSIM card into the u-GSM nanoSIM socket. Make sure the SIM card have the PIN checking procedure removed. You may use one phone to do this, or use the guidelines here: <https://itbrainpower.net/FAQ/SIM-CARD-PIN-CODE-REMOVE>

Step 3: Connect the u-GSM modem to the BPI M1+ using trough BPI GPIO connector, as shown in picture bellow [u-GSM modem with uFL connector]:

BananaPI M1+ u-GSM modem brief how to.



Or [u-GSM modem with SMA connector], wire the modem with BPI M1+ as in schema below:



BananaPI M1+ u-GSM modem brief how to.

Step 4: Connect the LiPO/LiION battery to the u-GSM modem [not required if you use super-capacitor].

Important - [u-GSM modem with *uFL* connector] – In order to avoid RF interferences, make sure you route the antenna feeder *away* from BPI sensitive areas as: Ethernet and USB ports, HDMI and audio ports! Additional info may be read <https://itbrainpower.net/a-gsm/u-GSM-Raspberry-Pi-antenna-notes>

BPI operating system preparation:

Step 1: Download BPI M1+ *Armbian Stretch* image from <https://www.armbian.com/banana-pi-plus/>

Step 2: Install the Armbian Image to your SD card. Utilities / resources that may help you:

- https://docs.armbian.com/User-Guide_Getting-Started/
- <https://sourceforge.net/projects/win32diskimager/>
- <https://www.balena.io/etcher/>
- <https://www.sdcard.org/downloads/formatter/index.html>

Step 3: Insert the SD card into BPI M1+, connect one keyboard, LAN interface, HDMI monitor and power your BPI M1+ and u-GSM bundle. For powering use one power supply that fulfil BPI M1+ specifications!

Step 5 Login to shell interface [initial credentials root - 1234].

5.a. Let's make some updates:

```
sudo apt-get update
sudo apt-get upgrade
```

5.b. Install additional packages required:

```
sudo apt-get install ppp
sudo apt-get install python-serial
sudo apt-get install python-dev
sudo apt-get install python-setuptools
```

and some utilities:

```
sudo apt-get install mc
sudo apt-get install mtr
```


BananaPI M1+ u-GSM modem brief how to.

Remarque: In our tests, the NTP service [in DHCPD configuration], as is delivered default, starts automatically one additional IP V6 interface and one listener. If you would like to disable this behavior, just edit the `/etc/rc3.d/@S01ntp` and comment line 26 [ntp dhcpd related]

5.c Add static ETH0:0 IP address, add USB support for u-GSM modems

Using your preferred editor, edit the `/etc/rc.local` file: `mcedit /etc/rc.local`

Add following lines, just above `exit 0` line:

```
#next 4 lines add USB support for itbrainpower.net u-GSM LTE modems
/sbin/modprobe --first-time usbserial > /dev/null 2>&1
echo 2C7C 0191 > /sys/bus/usb-serial/drivers/generic/new_id #u-GSM equipped with EG91
echo 2C7C 0195 > /sys/bus/usb-serial/drivers/generic/new_id #u-GSM equipped with EG95
echo 2C7C 0296 > /sys/bus/usb-serial/drivers/generic/new_id #u-GSM equipped with BG96

sleep 20
/sbin/ifconfig eth0:0 192.168.122.200 netmask 255.255.255.0 up > /dev/null
```

Save. Reboot your BPI.

5.d Now, let install Python GPIO support for BananaPI M1+, first:

```
git clone https://github.com/BPI-SINOVOIP/BPI-WiringPi.git -b BPI_M1_M1Plus
cd BPI-WiringPi
chmod +x ./build
sudo ./build
then:

git clone https://github.com/BPI-SINOVOIP/BPI-WiringPi2-Python.git -b
BPI_M1_M1Plus
cd BPI-WiringPi2-Python
sudo python setup.py install
```

5.e Check the installation. Test u-GSM power on / power off procedures:

- `gpio readall` will list all GPIO ports mode /values.
- All u-GSM modems but BG96 power on check
 - o `gpio mode 4 out`
 - o `gpio write 4 1`
- All u-GSM modems but BG96 power off check
 - o `gpio write 4 0`
 - o `gpio mode 4 in`
- u-GSM BG96 power on
 - o `gpio mode 4 out`
 - o `gpio write 4 1`
 - o `gpio mode 5 out`
 - o `gpio write 5 1`
 - o wait for about 1 second, then
 - o `gpio write 5 0`

BananaPI M1+ u-GSM modem brief how to.

- u-GSM BG96 power off
 - o `gpio write 4 0`
 - o `gpio mode 5 in`
 - o `gpio mode 4 in`

Setup and using ppp, powerOn and powerOff scripts:

Step1: Download “u-GSM shield RPI [and BBB] UTILITIES” from <https://itbrainpower.net/downloads#u-GSM>

For download, the following information are required: your name, email address and the modem IMEI. The modem IMEI can be found printed on the Quectel module, or via AT+GSN command.

Expand the archive, chdir to expanded folder and copy the etc/ppp/peers/u-GSM file to /etc/ppp/peers/u-GSM, make the destination file accessible by anyone, then edit it:

```
cp startPPP /root/startPPP
cp stopPPP /root/stopPPP
chmod 777 /root/startPPP
chmod 777 /root/stopPPP

cp etc/ppp/peers/u-GSM /etc/ppp/peers/u-GSM
chmod 777 /etc/ppp/peers/u-GSM
mcedit /etc/ppp/peers/u-GSM
```

Update:

- YOUR GSM PROVIDER APN [underlined parameter] in line `'connect "/usr/sbin/chat -v -f /etc/chatscripts/gprs -T net'`
- serial port from `/dev/ttyAMA0` to `/dev/ttyS1` – for UART modem connectivity
- if USB connection it's used, chose the USB port that fit Quectel module USB port [follow the guidelines inside the u-GSM file]
- other various ppp options, in order to fit YOUR GSM PROVIDER ppp settings.

Save.

Step2:

2.1 Create file /root/ ponModem.py.

```
chmod 777 /root/ ponModem.py
mcedit 777 /root/ ponModem.py
```

BananaPI M1+ u-GSM modem brief how to.

Copy the code bellow and paste inside ponModem.py

```
#!/usr/bin/python

#choose your u-GSM modem type bellow
MODEM=3          #select this for BG96 modems!!
#MODEM=2          #select this for EC91 or EC95 modems!!
#MODEM=1          #select this for ANY other u-GSM modem type (M95FA, BC95G,
UG95x or UG96)

POWERPIN=4  #ENABLE PIN is GPIO 23
RESETPIN=5  #RST PIN is GPIO 24

INPUT=0
OUTPUT=1

import sys
import wiringpi2

HIGH = 1
LOW = 0

print("poweron the u-GSM shield")

wiringpi2.wiringPiSetup()
wiringpi2.delay(50)

wiringpi2.pinMode(POWERPIN,OUTPUT) # Set pin as OUTPUT
wiringpi2.digitalWrite(POWERPIN,HIGH)

wiringpi2.pinMode(RESETPIN,OUTPUT) # Set pin as OUTPUT
wiringpi2.digitalWrite(RESETPIN,LOW)

sys.stdout.write('.')
sys.stdout.flush()
wiringpi2.delay(1000)

if (MODEM == 3):
    wiringpi2.digitalWrite(RESETPIN,LOW)
    sys.stdout.write('.')
    sys.stdout.flush()
    wiringpi2.delay(1000)
    wiringpi2.digitalWrite(RESETPIN,HIGH)
    MODEM = 1

#here we wait for modems to boot
MODEM = MODEM * 8
while(MODEM > 0):
    sys.stdout.write('.')
    sys.stdout.flush()
    wiringpi2.delay(1000)
    MODEM = MODEM -1

print("")
exit(0)
```

Update modem type accordingly your u-GSM modem module type [eg.: MODEM=2 for EC95E]. Save.

BananaPI M1+ u-GSM modem brief how to.

2.2 Create file /root/ poffModem.py.

```
chmod 777 /root/ poffModem.py
mcedit 777 /root/ poffModem.py
```

Copy the code bellow and paste inside poffModem.py

```
#!/usr/bin/python
POWERPIN = 4
RESETPIN = 5

HIGH = 1
LOW = 0
INPUT = 0
OUTPUT = 1

import wiringpi2
import os

print("poweroff the u-GSM shield")

os.system("/root/stopPPP >> /dev/null") #shutdown PPP if any

wiringpi2.wiringPiSetup()

wiringpi2.delay(50)

#now release
wiringpi2.pinMode(POWERPIN, INPUT)
wiringpi2.pinMode(RESETPIN, INPUT)

exit(0)
```

Save.

Step3: Final installation check

Bring up the ppp interface:

- `sudo python ponModem.py`
- `./startPPP`

Bring down the ppp interface:

- `./stopPPP`
- `sudo python poffModem.py`

BananaPI M1+ u-GSM modem brief how to.

Good to know:

BananaPI M1+ GPIO reference: http://wiki.banana-pi.org/Banana_Pi_BPI-M1%2B#GPIO_PIN_define

If supercapacitor it's used instead LiPO/LiION battery, avoid to run `ponModem.py` script for 90 seconds after BPI boot end [allow the capacitor to be loaded at full capacity, before power ON the modem].

Default route it's deleted when you run the `startPPP` script. If you want to avoid this, just comment "`sudo route del default`" line inside `startPPP`.

`ifconfig -a`, `route`, `traceroute`, `mtr` utilities may help you to test your TCP status/settings/routes.

`tail -f /var/log/messages` may help you identify and debug the u-GSM USB port enable/disable process and to debug the ppp process.

`iptables` may help you to setup service firewall and to enable port masquerading (service forwarding/internet tethering) for LAN/wireless LAN.

Ported python examples for BPI M1+. are available at: <https://itbrainpower.net/downloads.php#u-GSM>.

For download, the following information are required: your name, email address and the modem IMEI. The modem IMEI can be found printed on the Quectel module, or via AT+GSN command.

THIS TUTORIAL IT'S PROVIDED WITH ABSOLUTELY NO WARRANTY! USE IT AT YOUR OWN RISK!

Original how to written by Dragos Iosub & itbrainpower.net team