

kick-start for a-gsm 2.064 by itbrainpower.net

ARDUINO & RASPBERRY PI GSM / GPRS / DTMF / SMS shield

1. HARDWARE

1.1 a-gsm pin definition

1.2 POWERING a-gsm shield

1.2.1 General information

1.2.2 VOLTAGE SETTINGS RECOMMENDED FOR USE WITH ARDUINO

1.2.3 Powering examples

1.3 WIRING a-gsm shield

1.3.1 General information

1.3.2 Connection information for Arduino UNO

1.3.3 Connection information for Arduino Mega and Mega2560

1.3.3 Connection information for Raspberry PI B+ or Raspberry PI II

1.3.4 USB interface

1.3.5 Audio interfaces information and wiring

1.4 SERIAL communication

1.5 SIM card support

1.6 uTF/uSD card support

1.7 LEDs

2. Debugging (your) running code

2.1 Preparation phase

2.2 AT chat debugging how-to

3. Software examples and code support

3.1 a-gsm series software kickstart for Arduino

3.2 a-gsm series GSM / GPRS / DTMF / SMS ARDUINO shield examples list

3.3 a-gsm series GSM / GPRS / DTMF / SMS Raspberry PI shield examples list

3.4 Quectel M85 AT command manual

4 General/legal software/hardware considerations

kick-start for a-gsm 2.064 by itbrainpower.net

ARDUINO & RASPBERRY PI GSM / GPRS / DTMF / SMS shield

1. HARDWARE

1.1 a-gsm pin definition

PIN definition:

Pin D2 = GSM TXD(RX),
Pin D3 = GSM RXD(TX),
Pin D7 = PWRKEY - POWER-CONTROL-MODEM(ON/OFF),
Pin D5 = MODEM-STATUS,
Pin D6 = RESET-MODEM,
PinRST = Arduino RESET OUT,
Pin5V = Arduino 5V,
PinVin = Arduino Vin,
Pin GND(1&2) = GND

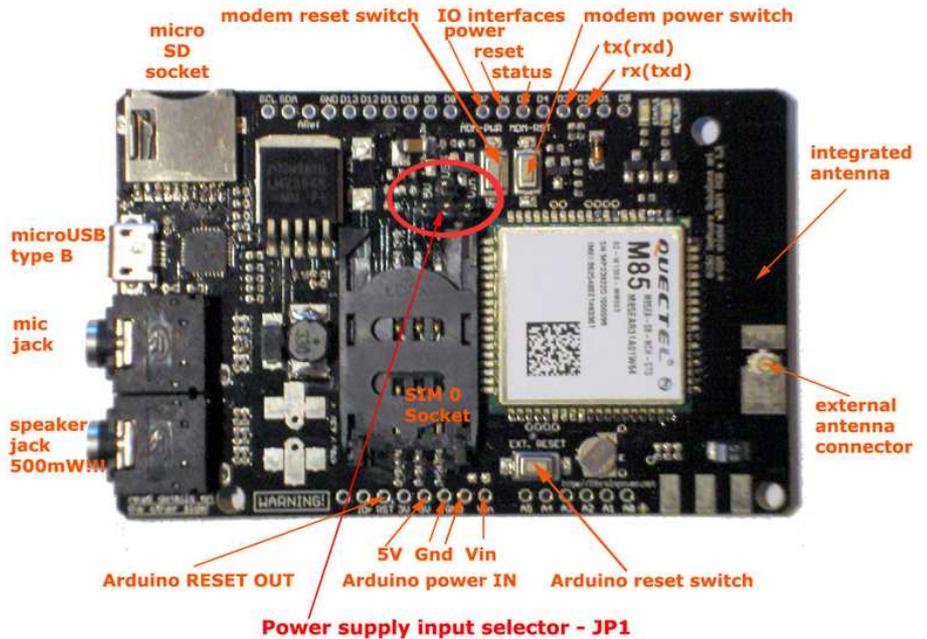


Image 1

Standard Arduino Pin-out

ONE to ONE connection without additional cables for Arduino UNO/LEONARDO and Arduino MEGA ADK/MEGA 2560*

* Arduino LEONARDO & Arduino MEGA ADK/MEGA 2560, additional strap / 1k resistor may be needed



1.2 POWERING a-gsm shield

1.2.1 General information

5V-28V input support, low consumption, 3 way powering profiles: via a-gsm micro USB type B, trough Arduino Vin PIN or trough 5V PIN, with manually selector (JP1) for users convenience.

Besides the high performances of the a-gsm embedded switching power supply, you MUST use ONLY quality power supply source that can provide at least 1A (for 5V sources) / 850mA (for 7-8V sources).

If powering trough Arduino (or other) Power supply connector, take care NOT to exceed the maximum recommended voltage values recommended by manufacturer.

If you choose "via a-gsm USB powering" mode, you can power your a-gsm board, even from your iPhone/ Arduino/tablet phone adapter.

1.2.2 VOLTAGE SETTINGS RECOMMENDED FOR USE WITH ARDUINO (JUMPER SELECTOR)

1*. power the Arduino and a-gsm board trough Arduino USB, from your PC => "use Vin pin"/"use 5V pin"

2***. power the Arduino and a-gsm board trough Arduino POWER CONNECTOR => "use Vin pin"/"use 5V pin" (depending on Arduino board implementation and Arduino Power adapter used)

3. Independent powering the boards. a-gsm board power trough a-gsm USB (from any (phone) power adapter with USB micro type B connector, 5V , >=1000mA) => "use USB as power"

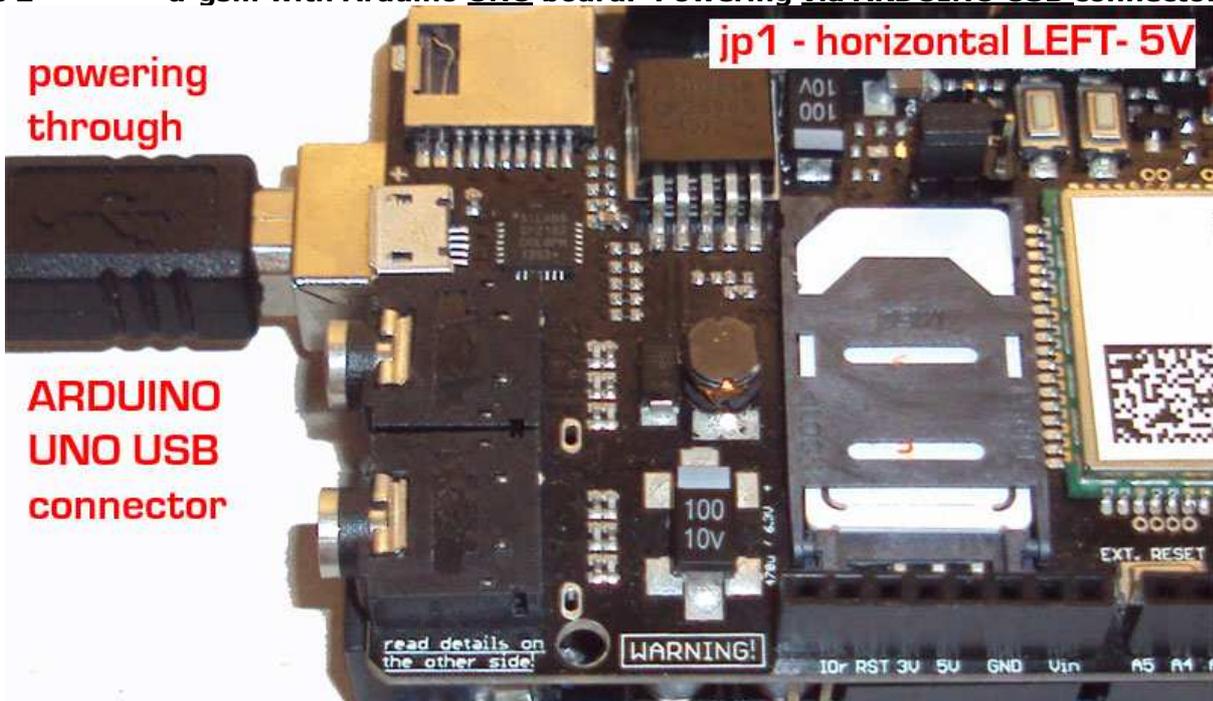
RECOMENDED (BEST first, GOOD, ACCEPTABLE): 3, 2 and 1(Image 2)

* Arduino Mega2560/Mega ADK ==> a-gsm possible NOT to work. Try power trough Arduino external power connector, using one power adapter 5-12V ~1000mA or using variant 3.

** Arduino Mega2560/Mega ADK ==> a-gsm will function if "use Vin pin" is selected.

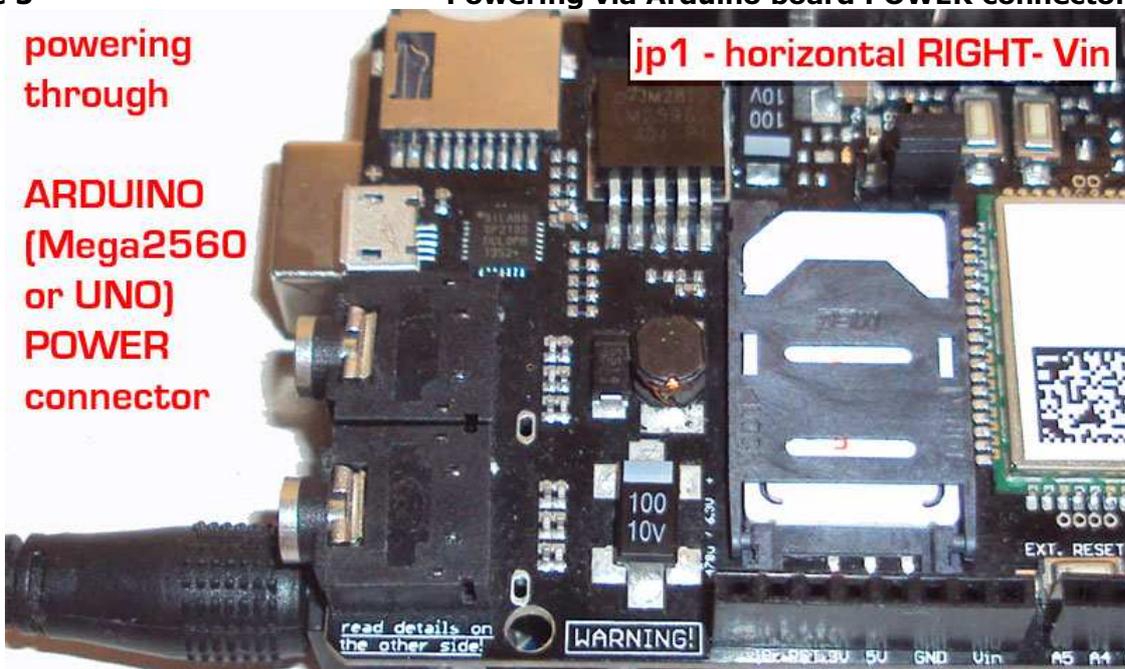
1.2.3 Powering examples

Image 2 a-gsm with Arduino UNO board. Powering via ARDUINO USB connector



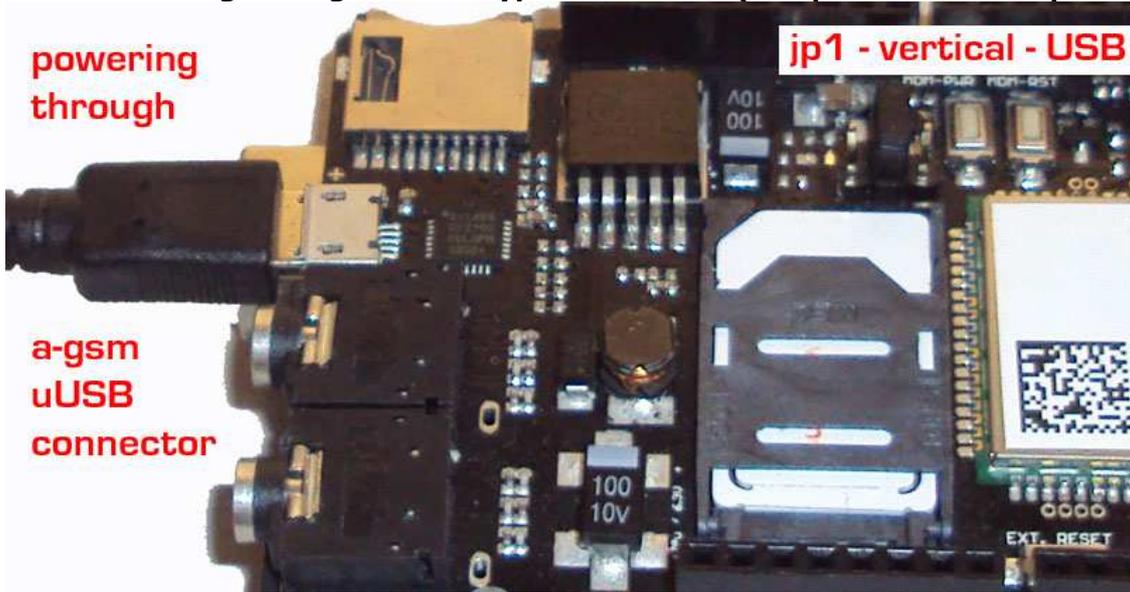
* Does not fit for Mega/Mega2560 boards

Image 3 Powering via Arduino board POWER connector



1.2.3 Powering examples (part2)

Image 4 Powering via a-gsm uUSB typeB connector (independent boards powering)



1.3 WIRING a-gsm shield

1.3.1 General information

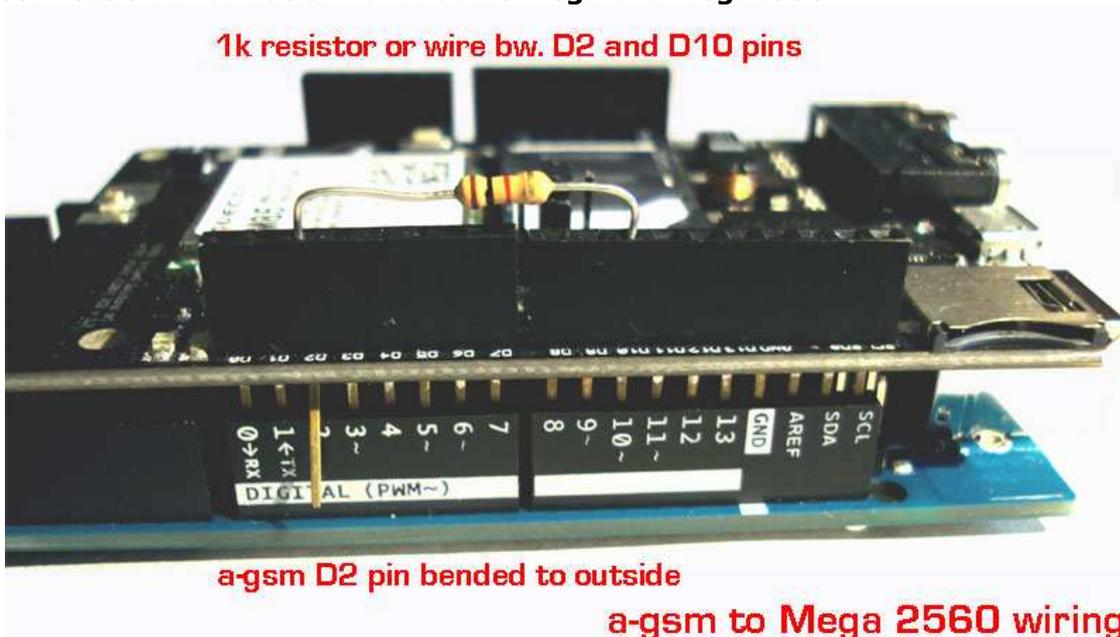
a-gsm digital interfaces [D2 = GSM TXD(RX), D3 = GSM RXD(TX), PWRKEY - POWER-CONTROL-MODEM (ON/OFF), D5 = MODEM-STATUS, D6 = RESET-MODEM, PinRST = Arduino RESET OUT] are available in standard ARDUINO format, are 5V and 3V compliant (you can wire/interface them directly to your micro-controller / board 5V down to 3V compliant). Also, the powering pins (5V, 2xGND and Vin), are available in same ARDUINO standard pin-out. See on Image 1.

WARNING: - WIRING a-gsm with u-controllers/boards must be made with boards UNPOWERED!!

1.3.2 Connection information for Arduino UNO

STANDARD PIN CONNECTION, just plug the a-gsm board into the corresponding Arduino UNO board sockets. **BE CAREFUL AT POSSIBLE SHORT CIRCUITS ON EXTRA PINS!**

1.3.3 Connection information for Arduino Mega and Mega2560



a-gsm to Mega 2560 wiring

IMAGE 5 On the a-gsm board, header on pin2 must be pulled to side. Place a wire jumper (or 1k resistor-recommended) between pin2 and pin 10.

1.3.3 Connection information for Raspberry PI B+ or Raspberry PI II

Please see the "a-gsm & RPI B+ wiring" schema (same for Raspberry PI 2), available at: http://itbrainpower.net/a-gsm/images/RaspberryPI_a-gsm_shield-wiring.png.

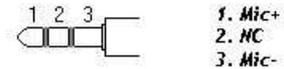
1.3.4 USB interface

USB adapter embedded - SERIAL to USB bridge adapter with micro-USB type B socket (you can use the a-gsm board as wireless modem with your PC, connecting it directly through USB to your PC - Windows and Linux compatible or for easily debug the AT chat). Use the drivers available at: <http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpcdrivers.aspx>

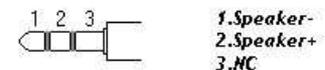
1.3.5 Audio interfaces information and wiring

Audio input and output, 3.5mm stereo jacks, in PC head-set standard.

- HIGH power audio output (700mW RMS), can independently drive speakers!
- High sensitivity capacitor Microphone interface



Microphone wiring *



Speaker wiring *

*PC headset can be connected directly

HEALTH AND SAFETY WARNING!!!!!!!!!!!!

You can damage your ears! Use it with care when headset is connected.

We recommend to use audio setup commands in order to limit the output power. Use AT+CLVL=20 (as maximum value).

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

a-gsm 2.064 audio wiring

Image 6

1.4 SERIAL communication

Your a-gsm board 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 speed first! This feature performs in same manner for TXD/RXD SERIAL connection over 3-5V compliant interface and in connection over USB.

Communication speed performed in code examples:

- a-gsm series GSM / GPRS / DTMF / SMS ARDUINO shield examples list >> 9600bps, 8N1 *
- a-gsm series GSM / GPRS / DTMF / SMS Raspberry PI shield examples >> 19200bps, 8N1
- a-gsm series software kickstart for Arduino >> 9600bps, 8N1 **
- a-gsm-raspian-ppp-1.0.tar.gz (Raspian PPP and routing utility) >> 115200bps, 8N1

IMPORTANT! ARDUINO ENVIRONMENT Serial Monitor speed:

* 9600bps

** 57600bps

1.5 SIM card support

Two SIM cards usage is supported (single SIM active). Depending on you a-gsm P/N, your board came populated with 2SIM sockets – top SIM (default) and bottom SIM (AGSM2064#2S0AP or AGSM2064#2SBAP) or with 1SIM socket – top SIM (default) (AGSM2064#1S0AP or AGSM2064#1SBAP). Switching the active SIM can be done on the fly using series of AT commands.

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

1.6 uTF/uSD card support

The micro SD/TF interface is available as standard and supports cards up to 32Gb. The TF/SD 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: saving local data, when your project goes to deep sleep/ low energy mode).

WARNING: uSD card insertion/removal must be made with boards UNPOWERED!!

1.7 LEDs

a-gsm shield it is equipped with two LEDs (yellow - STATUS & RED/GREEN for GSM MODE) that signalize the working state of the GSM modem.

STATUS LED (yellow) lights when the M85 modem is ON.

GSM MODE LED (red/green) blinks twice per second when the M85 it is NOT registered into the GSM network and blinks around 1.5 seconds when the modem is registered.

HINTS:

- A. if the STATUS LED blinks (once at 4-10 seconds) means that your POWERING schema has NOT been set proper!! Try to change your power source or the powering mode (see on chapter 1.2).
- B. if the GSM MODE LED does NOT acquire the "GSM 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 GSM signal is to low at the working place. Try to relocate your board, or use an external antenna.
 - c. Your GSM provider has no (2G) service at the working place. Change with SIM from other GSM company.
 - d. Your SIM has no credit. Add some credit to your SIM.

2. Debugging (your) running code

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 a-gsm board with USB cable to your computer. Follow install procedure, if needed.

A4. Open your terminal software, connect to the (a-gsm) com port (in hardware is identified as CP2102 port)

A5. Set the proper communication speed: 9600bps, 8N1

A6. Close the terminal software. Disconnect the cable.

Cable it your Arduino/RaspberryPI/other a-gsm project, in respect of a-gsm hardware tips. Compile/upload your code.

2.2 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 preferred u-controller and the a-gsm board. Run your Arduino/RaspberryPI code. The AT exchange will be replicated into the terminal window.

3. Software examples and code support

3.1 a-gsm series software kickstart for Arduino

Interactive interface with you're a-gsm board. You can dial, pick up, hang up calls, read, delete or send SMSs, see the signal strength, modem serial(IMEI), SIM serial(IMSI), GSM and GPRS registration status, and even interact with the modem trough AT commands, directly from the application. More features will be added soon (your contribution will be welcomed!).

Download from (module's IMEI & your email address needed):

<http://itbrainpower.net/a-gsm/gsm-shield-Arduino-RaspberryPI-features-code-examples.php>

Uncompress the archive, be sure both sketch (.ino) files are located in a folder named "kickstart". Double click on the "kickstart.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. You can use, also, the debugging procedure, using the USB connection, in order to have more details about

Be sure: - you're made proper setup for your board and a-gsm. SIM card without PIN checking procedure (remove PIN checking, fist), must be inserted into default SIM holder (on a-gsm's TOP).

3.2 a-gsm series GSM / GPRS / DTMF / SMS ARDUINO shield examples list

Compliable code (GPRS, DTMF, SMS, VOICE, file handling, and other) examples for you're a-gsm board and Arduino. Can be used as foundation for your GSM projects. Compile and running directives inside the code and associated txt files. Set the ARDUINO "Serial Monitor" at 9600bps.

Speed settings for debug via USB (if desired): 9600bps, 8N1.

Download from (module's IMEI & your email address needed):

<http://itbrainpower.net/a-gsm/gsm-shield-Arduino-RaspberryPI-features-code-examples.php>

3.3 a-gsm series GSM / GPRS / DTMF / SMS Raspberry PI shield examples list

Running code (GPRS, DTMF, SMS, VOICE, file handling, and other) examples for you're a-gsm board and Raspberry PI. Can be used as foundation 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: some py files needs to run under root permission. You can try something like:

```
sudo python xyz.py
```

Download from (module's IMEI & your email address needed):

<http://itbrainpower.net/a-gsm/gsm-shield-Arduino-RaspberryPI-features-code-examples.php>

3.4 Quectel M85 AT command manual

Download from:

<http://itbrainpower.net/a-gsm/gsm-shield-Arduino-RaspberryPI-features-code-examples.php>

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!!!