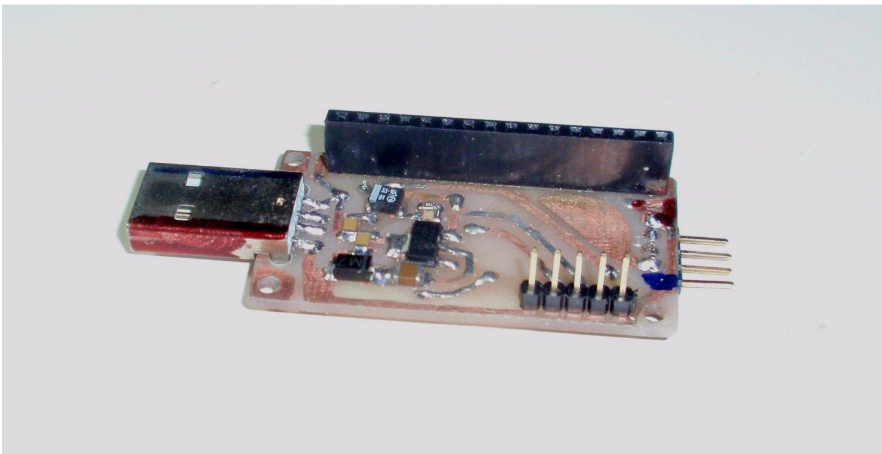
LoRa – PIC – USB system

This LoRa project is the evolution of my previous shield for 3.3 V Arduino and compatibles. It is still an experimental design, using LoRa module mounted on a socket, and not soldered to the board. Since 1.27 mm pitch connectors are quite fragile, the basic idea was to use a solution with two board, connected together using Arduino-like connectors. They are quite high, then RF module and his socket can be located in the interspace between boards, protecting them during handling.

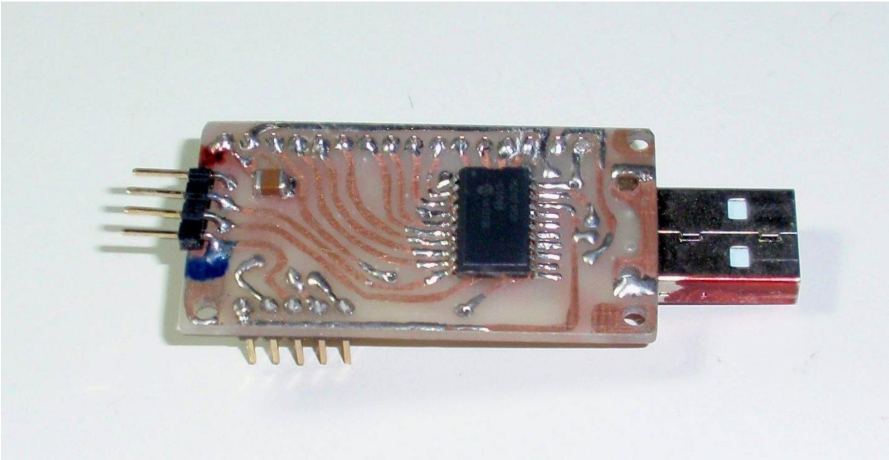
First board is just the PIC board. The model selected is the PIC16F1459, an 8 bit mid range, with 8K words of program and 1 kByte of RAM. The PIC is a Harvard machine, and program word can be variable. In this version it is 14 bits wide, and 8K words are 14 Kbytes. This PIC is packaged in a 20 pin SOIC, with a pin pitch of 1.27 mm (0.05”). easy to solder for an hobbist. The PCB also can be hand drawn by hand using a very sharp felt-tip pen. Of course copper should be previously cleaned very well using emery paper, but the ink of the model used, shown in the picture,

can survive in ferric chloride just the time necessary for copper etching.
Be careful when drawing and with timing ! I have no experience in other etchants like sodium persulfate ot hydrogen peroxide.

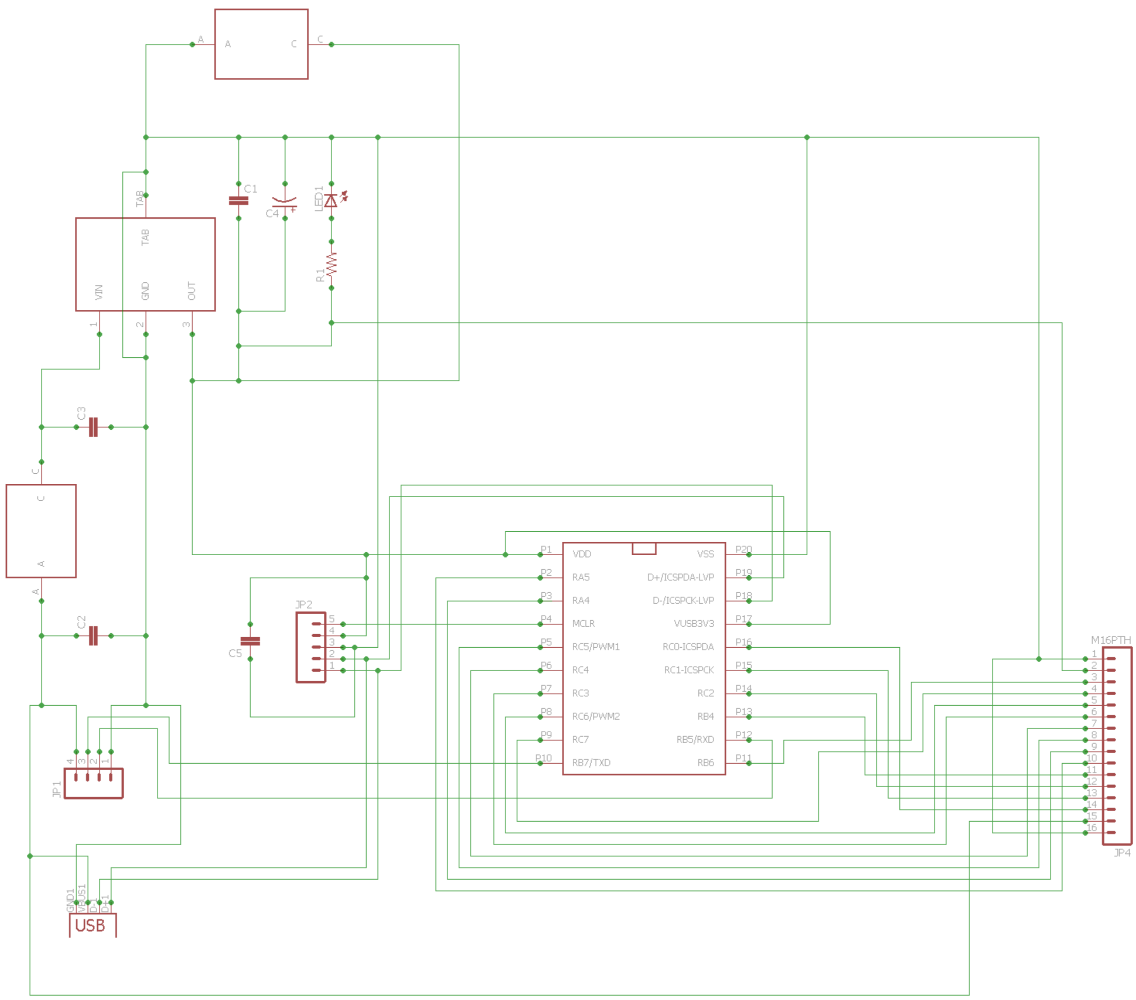
The PIC16F1459 is a recent advance by Microchip. It has the USB module, only in device mode, and can work without crystal, sinchronizing itself on the host timing. Then this two board combination achieves two targets: can work as a USB LoRa peripheral for a PC or a notebook, or work alone as a portable device, like a propagation tester (my original project target) or a long range remote control. The PIC board can be seen in pictures. up,



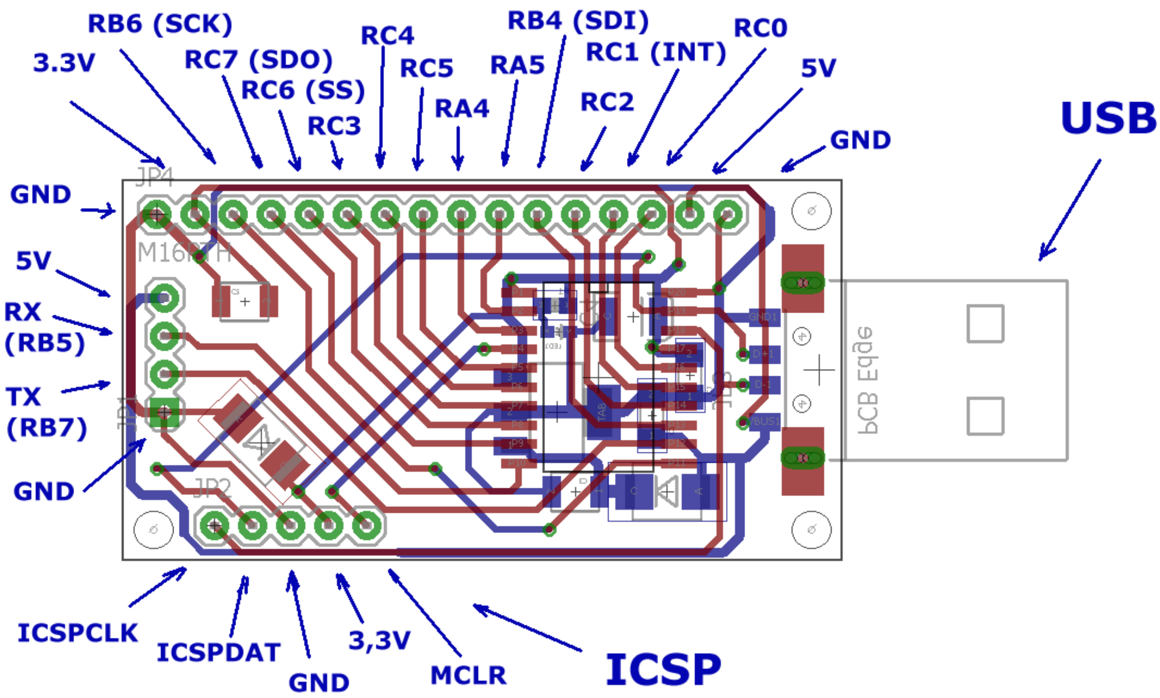
and down.



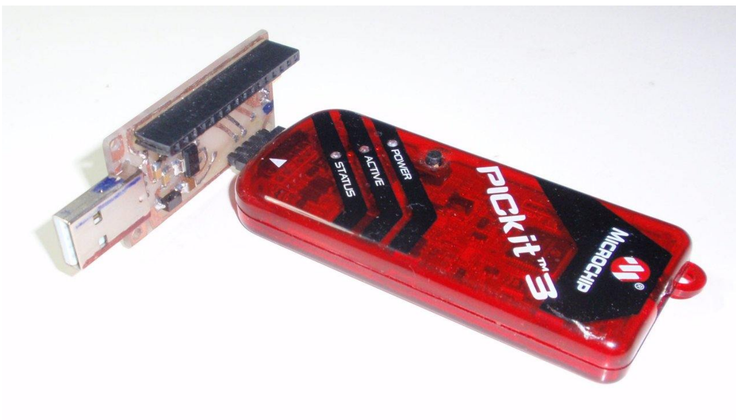
His shape is optimized for the two-board system, but can be used also as a general purpose microcontroller board, like a USB to serial, USB to I2C, or a USB to SPI interface converter. In particular there is plenty of USB to serial converters on the market, but USB to I2C, or USB to SPI converters are missing. The electric schema of the board is shown in the picture.



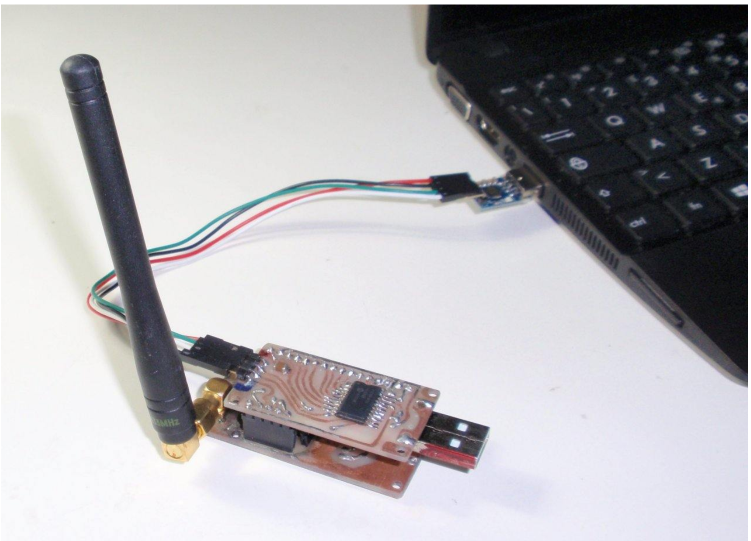
The global pinout of the board is shown in the picture.



Connectors are four: the first is the USB A-male, for PC connection, or, alternatively, to power the board using a power bank in portable applications. The second is a five pin, 0.1” pitch male for the connection to the PICKIT 3 for programming. Original ICSP Microchip connector is 6 ways, but the sixth here is not used. Keep the PICKIT 3 aligned as in the picture ! The ease of connection is shown in the picture.



After programming can be used as a general purpose connector, to power a 3.3 V device, or for alternative use USB D+ and D- pins when USB is not in use. A clever feature of Microchip in fact is the possibility to use also the USB pins for ICSP data and clock. Of course USB should be disconnected during programming, or lines impedance is changed too much. Even MCLR pin can be reused under program control as a general purpose I/O pin. The third connector is a Arduino-like 16 pin female connector. Most of the PIC GPIO pins are found here. It includes also a 3.3V-GND couple and a 5V-GND couple. The fourth connector is a four way including 5V, GND, and TXD and RXD of the serial. Even TXD and RXD can be used under program control as GPIO pins. This connector can be used to connect the board to a host, like in the picture,



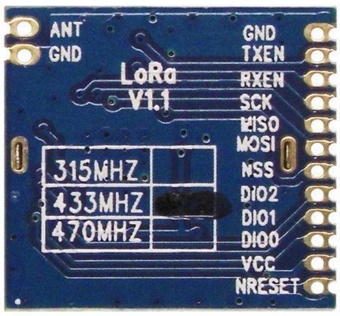
using a USB to serial converter, easy to find in the market.

You could ask: why use an externel USB to serial converter, when USB is already in the PIC ? For many reasons. First: USB programming is quite difficult, on PIC side and on the host side. Every programmer can learn it, but project complexity can or can not match learning complexity. Second: USB handling requires program memory. According to my experience minimal USB software requires 3K of 8K words of flash. Application can fit in the remaining 5K or not. This is valid for XC8 in the free version. Professional, paid optimizer can save up to 40 % of memory, but at present moment i can not afford it. Third: the PIC can run at many different clock speeds, from 31 Khz to 48 Mhz, with processing power and current consumption increasing in parallel. Obviously best choice is minimal frequency enough to cope with application complexity. For USB in low speed mode minimum clock frequency is 8 Mhz. In the application test program clock frequency is 2 Mhz, and i am testing for further reduction. Fourth: any USB device requires two 16 bit numbers, a PID (product ID) and a VID (vendor ID). They are not required in host mode, but the PIC16 can act only as a device. For in-house applications you can use the Microchip VID (0x04D8), but not for commercial applications. You can sublicence Microchip VID for a limited number of parts, or pay to buy, from USB consortium, your own VID. Using an external USB to serial converter you pay the device and the included VID in it, usually FTDI or Silabs.

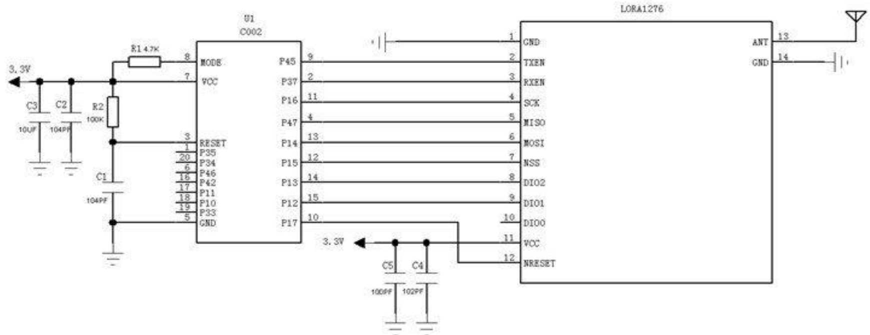
In Total on the four connectors there are three 5V and two 3.3V. The three 5V are input and are in parallel. Any one can be used to power the board, but not two or three. Using more than a power supply is not dangerous for the board, but can be dangerous for power supplies. After the 5V there is a protection diode agains reverse polarity, and then a Microchip TC1262 3.3V voltage regulator. Beware ! TC1262 pinout is not compatible with pinout of L1117 also often used in DIY projects. The two 3.3V are OUTPUTS. They can not be used as inputs to power the board, the PIC is not damaged, but the TC1262 is. There is on the board also a protection diode in case of input REVERSED on the 3.3V. It has been put because room was available, but is not really necessary, it is almost paranoia. On my prototypes is not mounted. In general, being a DIY board, some components can be not soldered when not necessary, like USB connector, this diode, power LED or serial connector, and, i repeat, the board can be used as a general purpose microcontroller board, independently from LoRa.

The second board is the LoRa board, designed to match the PIC board. It acts as a shield to it, like an Arduino shield to Arduino itself. It is powered from the 3.3V, the 5V is not connected and not used. LoRa is mounted on a socket, and there is a SMA female connector for antenna, allowing various different models, long or short, straight or angled, with different efficiencies. The connector itself can be straight or right angle, to fit at best any need.

There are many LoRa breakout boards on the market, wit different pinouts. The model chosen is the same of Arduino shield. Pinout is shown in the picture,

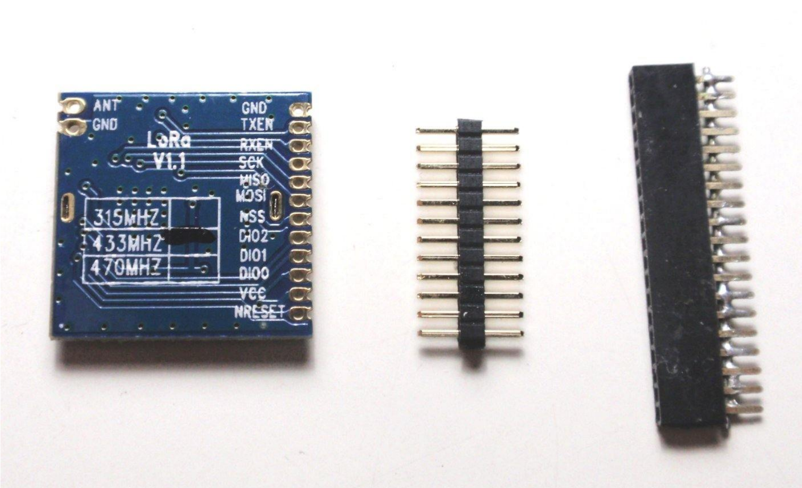


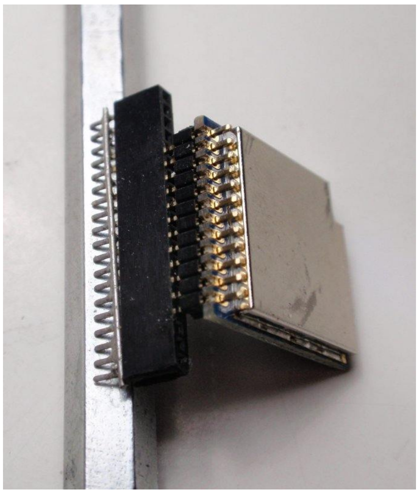
and also shown is the host connection suggested by manufacturer.



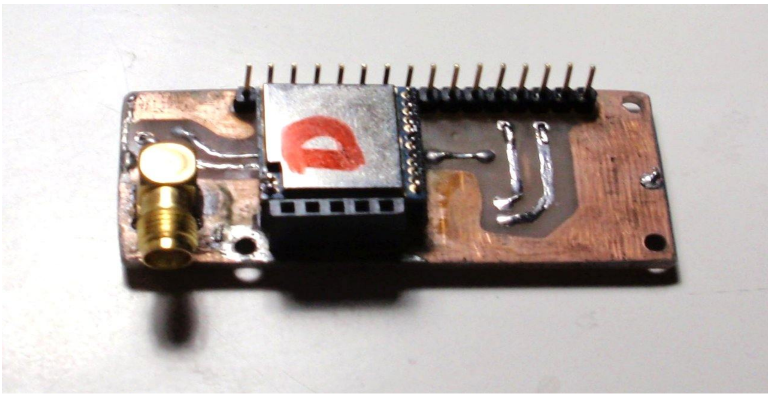
Connection is quite complex, four pins for SPI, one for RESET, one for RXenable, one for TXenable, and one for DIO0. Only four pins are left, and they are connected to a two LEDs and a two buttons, allowing the stand-alone use as a long range remote control. Finally two LEDs are connected to DIO1 and DIO2 pin of the LoRa. The second is used by it in case of faults.

To mount the breakout board on the socket, male pins are soldered. A temporary short-circuit female connector is used during soldering, to prevent voltage-damage, as shown in pictures.

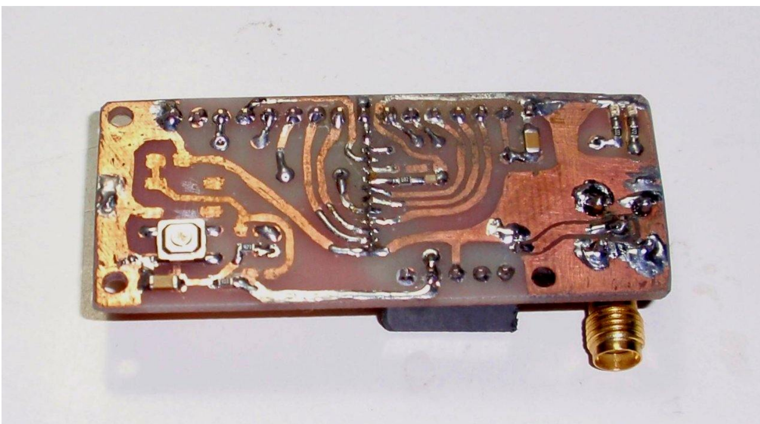




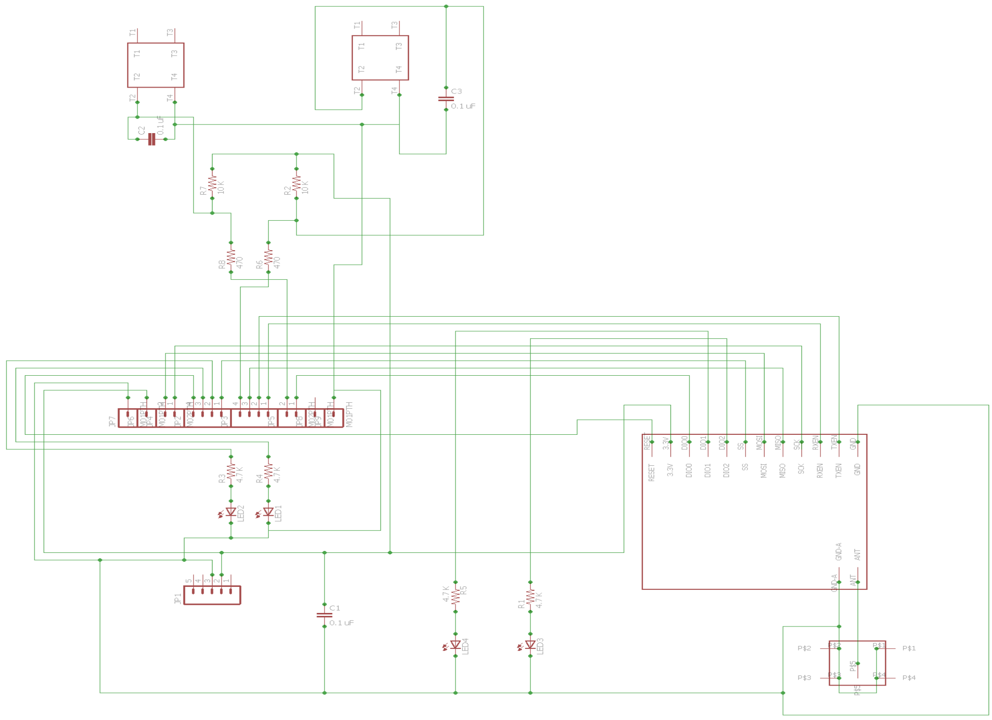
In the following pictures can bee seen the shield board, up,



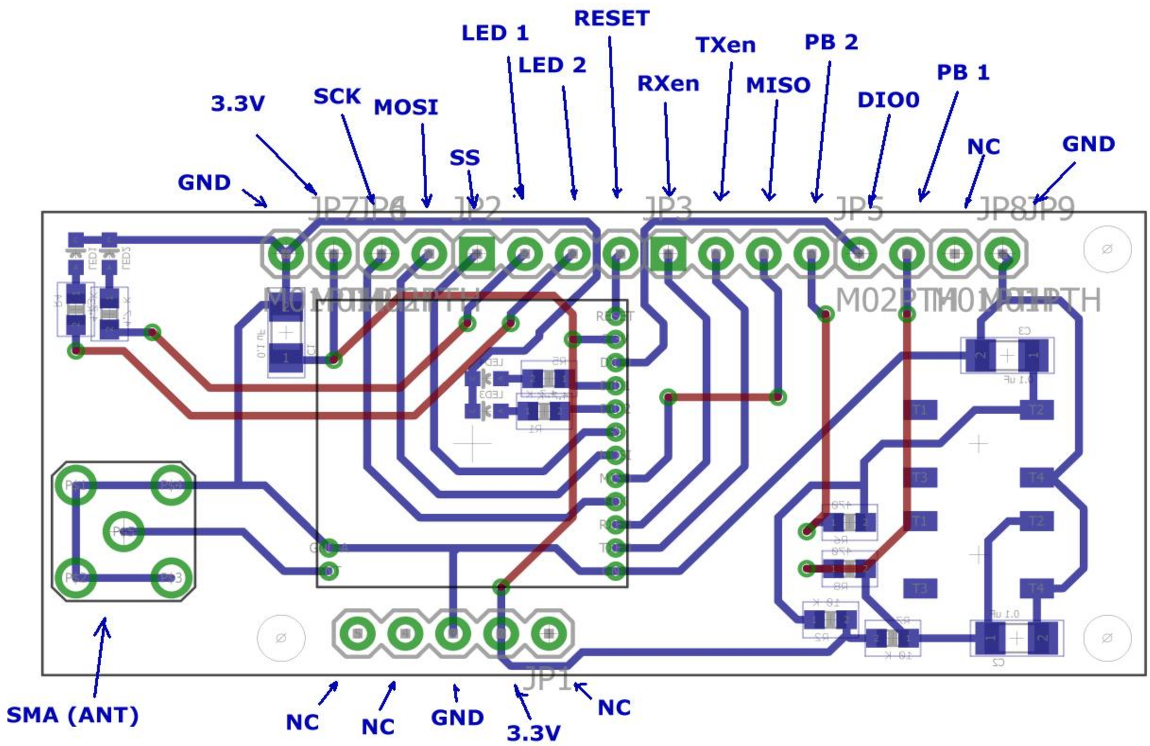
down,



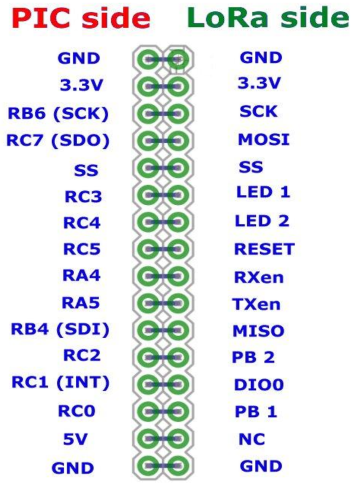
schematics,



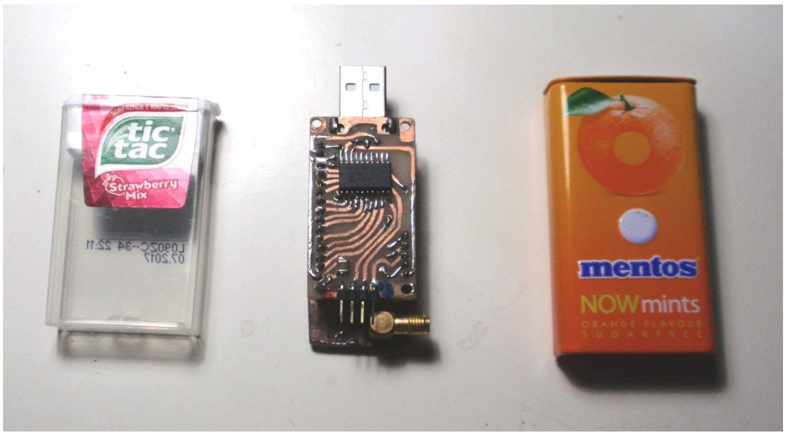
pinout,



and connections between boards on 16 way connector.



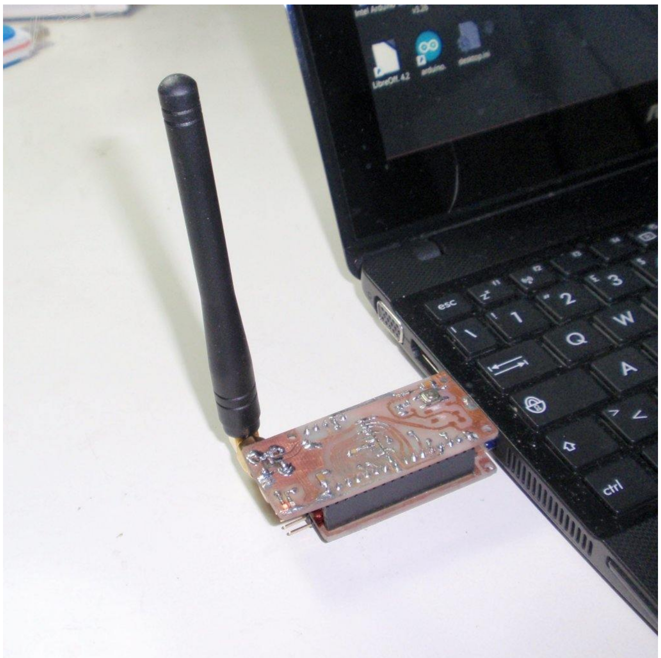
In both boards all connectors are soldered on just one side of the board, avoiding need of hole metallization, difficult for hobbists. This configuration ensures the correct side positioning of connectors. In case of wrong side, assembly the boards is not possible. In the picture can be seen

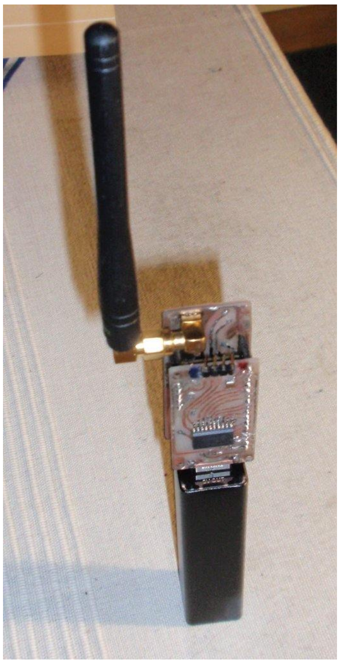


the overall size. Could be even smaller, but as seen compared to some of most popular mint pocket box, surely is not too big.

Test software is same of Arduino shield software, including only necessary changes from Arduino pseudo-C language to Microchip C language. In particular pin names and the instructions to set and read them. SPI is again used in software mode, but pinouts allow hardware mode. To be done in next step ! Files .c and .h included in .zip files are the standard of MPLAB X template. Create an empty project and overwrite just generated files. It is done ! Just a further warning: the PIC can operate from 2.3 to 5.5 Volts, but in this board it is powered to 3.3V. Accordingly signal levels on any interface should not exceed 3.3V, or the PIC can be damaged. Furthermore in MPLAB X project properties are to be set to LOW VOLTAGE PROGRAMMING. MPLAB warns you, but better two warnigs than one. Compiled size of connection handling is less than 1K words of the PIC, and using only the free version of compiler. 7K are free for application software, and even adding 3K for USB, something useful can still be done.

Concluding, the set of two boards can be used as a PC USB LoRa peripheral or a stand-alone system for various purposes,





according to the program uploaded in the PIC.

As a final warning, remember: the LoRa chip is very flexible, and power, frequency and bandwidth can be set in many different ways. Every country has his own laws and regulations. Values proposed in the sample software are just a template. It is your responsibility to be compliant to the law of the country where you are operating.

Be careful and have fun !

Torino (ITALY), july 4, 2016

Gianfranco PONCINI