

# ProMicroLoRa

## Rilevare e trasmettere dati in ambiente esterno

Versione 5.2

**Paolo Bonelli**

paolobo87@gmail.com

Il presente documento è distribuito con licenza Creative Commons BY-NC-SA  
This document is distributed with licence Creative Commons BY-NC-SA

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

**05/12/2019**



René Magritte - Bruxelles

## Sommario

Introduzione .....	2
La scheda .....	3
I due componenti principali.....	4
Piedinatura della scheda Promicrolora .....	5
Il collegamento tra i due moduli .....	6
Alimentazione .....	6
Consumi di RFM95.....	7
Consumi della scheda ProMicroLoRa .....	7
Antenne e portata .....	8
Programmazione di ProMicroLoRa con la IDE di Arduino.....	11
Librerie necessarie.....	11
Impiego della scheda ProMicroLoRa nell'outdoor sensing.....	12
Cosa si intende per Outdoor Sensing? .....	12
A quale scopo?.....	12
Glossario .....	13
Come possono funzionare i nodi trasmettitori di un sensore.....	14
Nodo Trasmettitore con invio dati A Tempo.....	14
Nodo trasmettitore con invio dati ad evento .....	15
Mini rete di nodi LoRa .....	17
Mini ponti radio .....	17
Inserimento dei dati in una rete WiFi.....	17
Scheda WiFi alternativa .....	19
Server Exosite .....	19
Alcune note sui parametri di trasmissione .....	20
Il KIT per iniziare (Natale 2019) .....	21
Il software per il KIT .....	22
Come collegare le varie parti DEL KIT .....	23

## Introduzione

In questo documento l'autore riporta la sua esperienza nell'uso di una particolare combinazione di tecnologie che gli ha permesso di trasmettere piccoli pacchetti di dati via radio a lunga distanza in condizioni di bassi consumi energetici. La banda di frequenza utilizzata è quella di 868 MHz, che può essere usata liberamente in tutta Europa per trasmettere messaggi brevi.

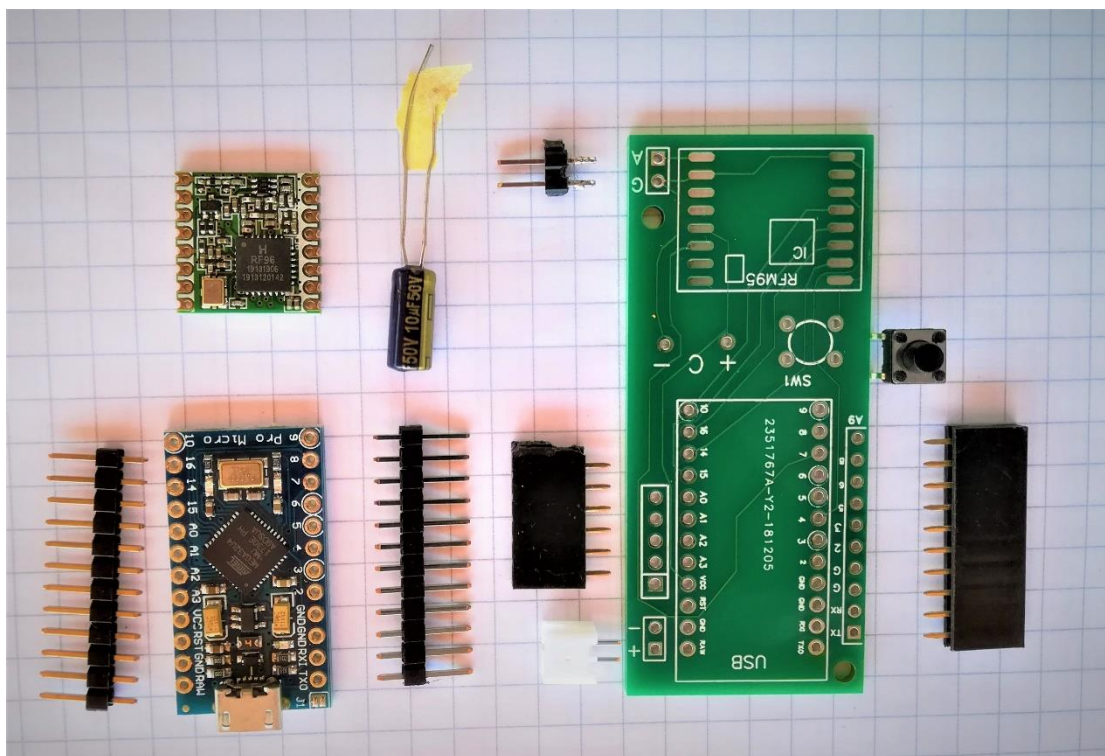
L'autore ha creato una scheda PCB, che chiama ProMicroLoRa, per utilizzare al meglio queste tecnologie, senza scopi commerciali ma seguendo la filosofia open-source. Questa scheda è stata impiegata in numerose applicazioni di monitoraggio in ambiente esterno. Applicazioni in cui la scheda ProMicroLoRa acquisisce i segnali da uno o più sensori formando un pacchetto di byte che viene trasmesso via radio. La stessa scheda è usata come ricevitore a qualche chilometro di distanza. Accoppiando poi il ricevitore con una scheda dotata di connessione WiFi, è possibile trasferire i dati su Internet.

Tutto quanto scritto e il software relativo, sono rilasciati così come sono, quindi non privi di errori o possibili miglioramenti. Tutti possono usare queste conoscenze secondo quanto riportato nella licenza Creative Commons, citata sopra. L'autore ha cercato di citare la fonte di tutto il materiale di terzi, nonostante ciò potrebbero esserci eventuali dimenticanze.

L'autore sarà grato a tutti coloro che vorranno comunicargli eventuali errori o imperfezioni trovati nei contenuti di questo documento.

Un particolare ringraziamento va alla Community di Wemake s.r.l. di Milano (wemake.cc) che ha supportato con consigli ed esperienza questo progetto.

## La scheda



ProMicroLora è una scheda PCB sulla quale sono stati montati due componenti principali:

un Arduino ProMicro con ATmega32U4 dotato di interfaccia USB (progetto Sparkfun) funzionante con clock a 8 MHz e 3.3 V di alimentazione;

un tranciver RFM95 della HOPERF adatto alle frequenze 868/915 MHz con modulazione LoRa (in Europa la frequenza 868 è libera per la trasmissione di piccoli pacchetti di dati).

Pochi altri componenti completano la scheda: un pulsante di reset e un elettrolitico per stabilizzare l'alimentazione del modulo radio. La scheda assomiglia alla Feather 32u4 LoRa della Adafruit, ma più economica.

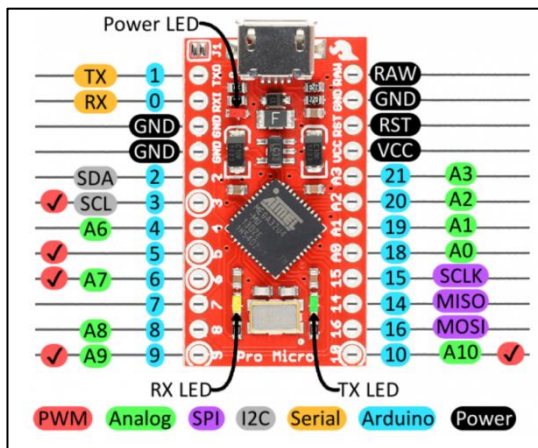
Lo schema dei collegamenti è riportato più avanti.

Le caratteristiche di Arduino ProMicro sono fornite dalla stessa Sparkfun e non sono molto diverse da quelle di Arduino Leonardo. Arduino Pro Micro può essere programmata con la IDE di Arduino scegliendo opportunamente la scheda "Sparkfun pro micro" dal menù "Gestione schede".

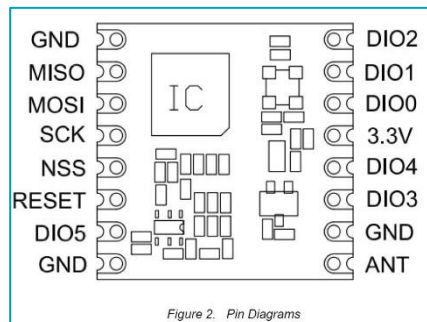
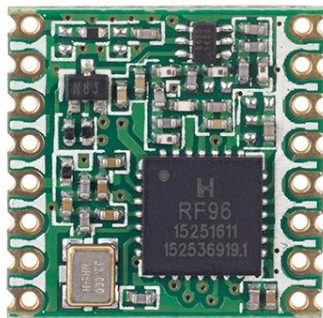
Tramite l'interfaccia SPI, ProMicro è in grado di dialogare con il tranciever RFM95 e inviare o ricevere pacchetti di dati via radio LoRa.

La modulazione LoRa (Long Range) è una tecnologia che si sta affermando per le sue caratteristiche di lunga distanza e bassi consumi, è stata brevettata dalla Semtech. Il sito ufficiale è: <https://www.semtech.com/lora>

## I DUE COMPONENTI PRINCIPALI



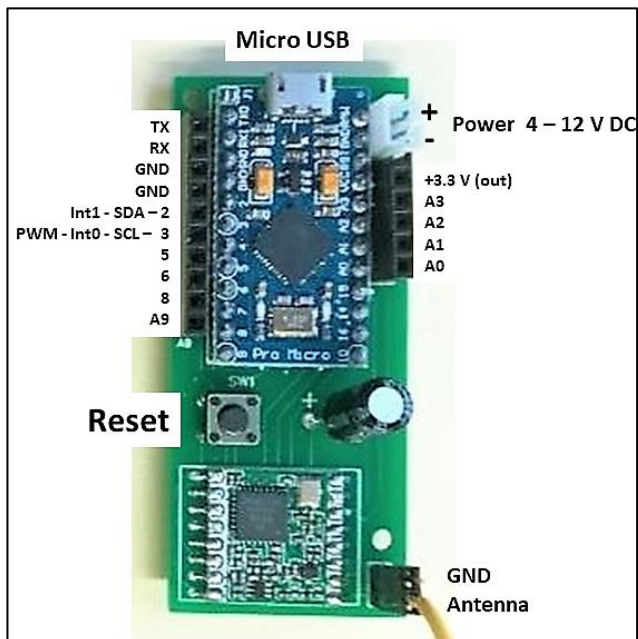
*ProMicro della Sparkfun. Nella scheda ProMicroLoRa alcuni piedini sono impegnati per il collegamento con il modulo radio (fonte Sparkfun)*



*Modulo Radio LoRa RFM95, 868 MHz*

*(fonte HopeRF)*

## PIEDINATURA DELLA SCHEDA PROMICROLORA



I piedini utilizzabili per i collegamenti con il mondo esterno sono:

A0, A1, A2, A3, A9: ingressi analogici con ADC a 10 bit, che possono essere usati anche come I/O digitali.

D2, D3 per l'interfaccia I<sup>2</sup>C (SDA, SCL) oppure come I/O digitali, il D3 è anche un output PWM. I piedini possono essere usati anche come INT<sub>1</sub> e INT<sub>0</sub> rispettivamente (Interrupt).

D5, D6, D8 I/O digitali. D5 e D6 possono essere anche output PWM. TX0, RX1 Interfaccia seriale e INT<sub>3</sub>, INT<sub>2</sub> (Interrupt).

VCC eroga 3.3 V per alimentare circuiti esterni, fino a 500 mA di assorbimento.

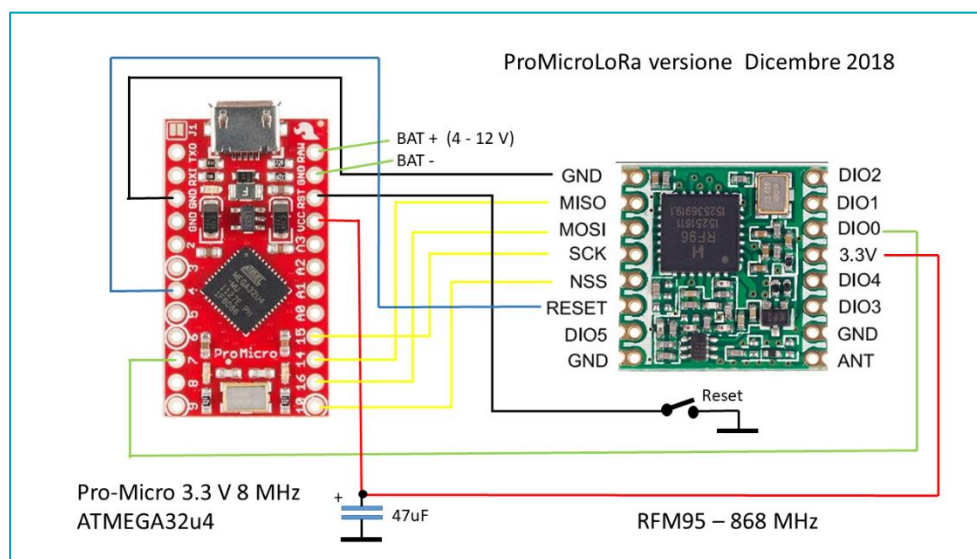
Per altri dettagli, fare riferimento alla documentazione su ProMicro 8 MHz, 3.3 V, della Sparkfun.

### LED

Nel software si può usare il piedino 17 per accendere o spegnere il LED incorporato e usato dal chip come RX nelle trasmissioni seriali. Da notare che funziona al contrario: Piedino HIGH >>> LED spento.

## IL COLLEGAMENTO TRA I DUE MODULI

Il modulo radio si interfaccia con l'Arduino ProMicro per mezzo della SPI (Serial Peripheral Interface) con i piedini: MISO, MOSI, SCK, NSS. Lo sketch quindi userà la libreria SPI.h. Altri collegamenti sono l'alimentazione a 3.3 V presa dall'uscita Vcc di Arduino Pro Micro, il GND, il DIO0 che è collegato al D7, il RESET che è collegato al D4. Tutti questi collegamenti sono compresi nel PCB. Inoltre è presente un pulsante che collega il piedino RST del ProMicro al GND.



## ALIMENTAZIONE

La scheda può essere alimentata con una tensione continua da +4 a +12 V applicata al piedino RAW (+) del ProMicro. Possono essere utilizzate anche 3 o 4 batterie da 1.5 V in serie. Può essere utile usare anche power bank a 5 V, ormai diffusi e poco costosi. Se invece si vuole usare una LiPo da 3.7 V bisogna inserire un DC/DC converter che aumenti questa tensione e la porti nel range suddetto.

## CONSUMI DI RFM95

Si riporta una tabella originale presa dal Data Sheet del RFM95

### 2.4.1. Power Consumption

Table 51 Power Consumption Specification

Symbol	Description	Conditions	Min	Typ	Max	Unit
IDDSL	Supply current in Sleep mode		-	0.2	1	uA
IDDIDLE	Supply current in Idle mode	RC oscillator enabled	-	1.5	-	uA
IDDST	Supply current in Standby mode	Crystal oscillator enabled	-	1.6	1.8	mA
IDDFS	Supply current in Synthesizer mode	FSRx	-	5.8	-	mA
IDDR	Supply current in Receive mode	<i>LnaBoost</i> Off, higher bands <i>LnaBoost</i> On, higher bands Lower bands	- - -	10.8 11.5 12.1	- - -	mA
IDDT	Supply current in Transmit mode with impedance matching	RFOP = +20 dBm, on PA_BOOST RFOP = +17 dBm, on PA_BOOST RFOP = +13 dBm, on RFO_LF/HF pin RFOP = + 7 dBm, on RFO_LF/HF pin	- - - -	120 87 29 20	- - - -	mA mA mA mA

Per quanto riguarda il ProMicro, i consumi dipendono molto dal tipo di impiego e dal software. Nello stato dormiente (sleep) i consumi dovrebbero scendere a 0.3 mA.

## CONSUMI DELLA SCHEDA PROMICROLORA

I consumi di una configurazione TX composta da:

- una scheda ProMicroLoRa con il LED rosso acceso e il verde che lampeggia solo nel momento della trasmissione;
- un sensore DHT22;
- un pacco batterie da 4 x AAA;

sono 1.8 mA a riposo (sleep), mentre durante la trasmissione, essendo molto rapida, il consumo di picco è di circa 130 mA per 70 ms (dati Adafruit per la scheda Feather).

Disattivando il LED rosso, indicazione di Power on, si raggiungono i 0.38 mA a riposo (durante lo sleep).

Per quanto riguarda una scheda usata come RX nella configurazione minima (KIT):

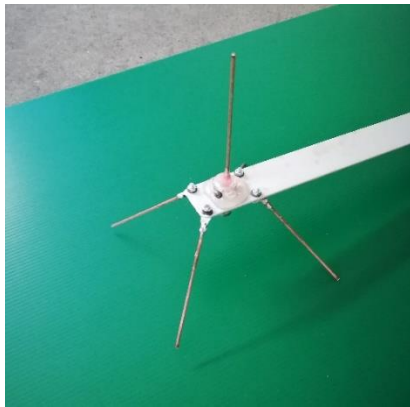
- una scheda ProMicroLoRa con il LED rosso acceso e il verde che lampeggia solo nel momento della ricezione;
- un display OLED 128 x 32;
- un pacco batterie da 4 x AAA;



i consumi sono pari a 27.8 mA a riposo.

## Antenne e portata

Per il funzionamento della radio, è necessario collegare un'antenna sul piedino apposito. Le dimensioni dell'antenna dipendono dalla lunghezza d'onda della portante. Un'antenna minimale, che consente già portate di un paio di km, può essere fatta con un filo di rame lungo 86 mm, collegato direttamente al pin della scheda (86 mm è in realtà la lunghezza del filo + quella del connettore). Antenne più performanti sono quelle a stilo o Ground Plane.



*Antenna Ground Plane autocostruita dall'autore. Diversi siti spiegano come fare. Ad esempio:*

<https://izoups.jimdo.com/antenne/vhf-gplane/>



*Antenna Ground Plane commerciale (fonte Futurashop)*



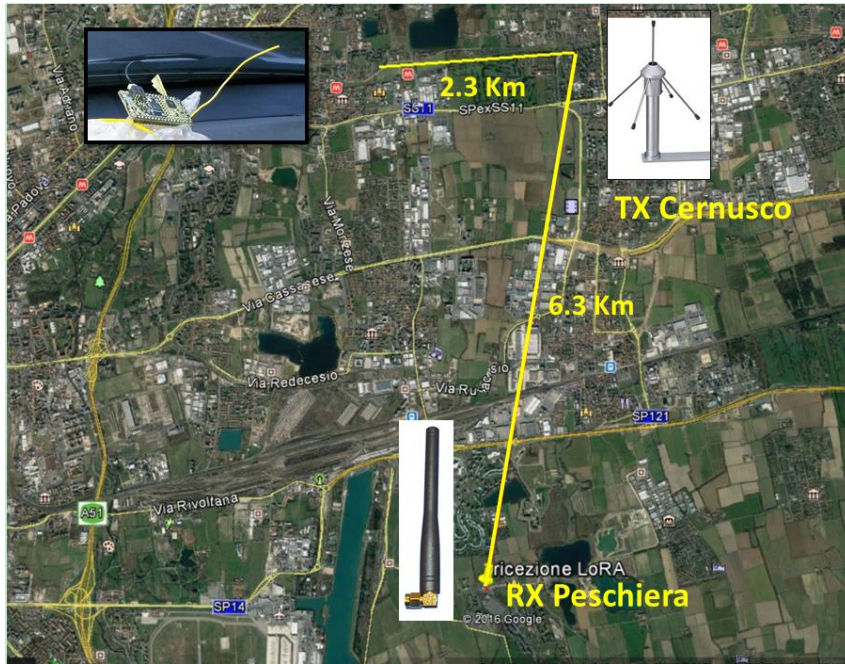
*Antenna a stilo pieghevole collegata ad una scheda Adafruit Feather LoRa con batteria LiPo.*



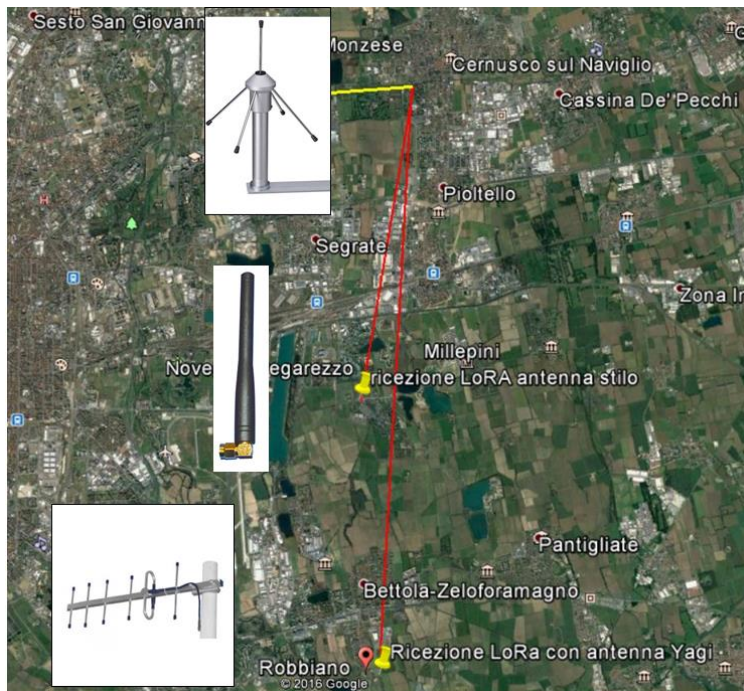
*Se si vogliono ottenere portate maggiori si possono usare antenne direzionali come la Yagi, sempre però dimensionate per la frequenza 868 MHz. Nella foto, quella usata dall'autore per le prove di portata.*

L'autore ha eseguito alcune prove di portata usando 2 schede Feather della Adafruit che, ai fini di queste prove, hanno le stesse prestazioni delle ProMicroLoRa. Sono infatti composte da un microcontrollore ATMEL 32u4 e un modulo radio HopeRF RFM95.

Durante le prove è stata sistemata un'antenna Ground Plane sul proprio balcone al settimo piano, mentre con un'altra scheda ci si è allontanati in diverse direzioni in modo da non avere ostacoli sulla linea retta congiungente le due antenne. I risultati sono rappresentati nelle seguenti figure.



*Prove di portata massima con antenna a filo(2.3 km), a stilo (6.3 km)*



*Prove di portata massima con antenna Yagi (11 km)*

## Programmazione di ProMicroLoRa con la IDE di Arduino

Lo sketch per far funzionare la scheda può essere scritto, compilato e caricato con l'interfaccia IDE, appositamente rilasciata da Arduino sul sito [arduino.cc](http://arduino.cc). Tutte le prove fatte dall'autore alla data di questo documento hanno usato la versione 1.8.5, ma quelle superiori dovrebbero funzionare altrettanto bene.

E' importante che la configurazione della IDE (versione in Italiano) abbia le seguenti impostazioni:

- Menù "**Impostazioni**", mettere un check mark sulle voci "**compilazione**" e "**caricamento**" a destra della voce "**Mostra un output dettagliato durante:**"
- Menù "**Strumenti**", scegliere il tipo di scheda "**Spark Fun Pro Micro**". Nel caso in cui questa scheda non dovesse essere presente nell'elenco, cliccare sulla voce "**Gestore schede**" e cercare "**SparkFun**", se si trova SparkFun AVR boards, click su installa, altrimenti bisogna ritornare al menù impostazioni e inserire il link: [https://raw.githubusercontent.com/sparkfun/Arduino\\_Boards/master/IDE\\_Board\\_Manager/package\\_sparkfun\\_index.json](https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json) o altro link consigliato sul sito della Sparkfun.
- Assicurarsi che la scheda sia quella a 8 MHz e 3.3 V

La scelta della porta Seriale nel menù "**Strumenti**", "**Porta**" è un pochino macchinosa le prime volte, ma ci si fa l'abitudine:

Se non vedete comparire alcuna porta, scollegate e ri-collegate il cavo USB, cliccate due volte di fila il tasto reset sulla scheda e aspettate qualche secondo.

Può capitare che, nella fase di caricamento dello sketch sulla scheda, compaiano nella finestra di dialogo della IDE ripetute richieste della Porta, in questo caso resettate la scheda durante queste richieste.

(Queste operazioni per l'impostazione della porta seriale sono valide per PC windows 7 o 10, l'autore non ha provato su altri sistemi operativi).

### LIBRERIE NECESSARIE

Per far funzionare il tranciever RFM95, occorre includere nel proprio sketch la libreria

<RH\_RF95.h> che si trova nel pacchetto RadioHead. Conviene caricarsi tutto il pacchetto disponibile al sito:

(<https://www.airspayce.com/mikem/arduino/RadioHead/>)

Questo pacchetto contiene le librerie per tutti i tranciever della HopeRF; sebbene a noi interessi solo il modulo per la RFM95, conviene lasciare gli altri moduli nella propria cartella Libraries.

Inoltre è necessaria la libreria <SPI.h> che si occupa del dialogo tra ProMicro e il tranciever. Questa libreria è compresa tra quelle fornite con la IDE di Arduino.

Altra libreria utile ma non necessaria è quella che si occupa di “mettere a dormire” la ProMicro al fine di consumare solo un minimo di corrente. Questa libreria è fornita dalla Adafruit:

<Adafruit\_SleepyDog.h>

[https://github.com/adafruit/Adafruit\\_SleepyDog](https://github.com/adafruit/Adafruit_SleepyDog)

## Impiego della scheda ProMicroLoRa nell'outdoor sensing

### COSA SI INTENDE PER OUTDOOR SENSING?

“Outdoor Sensing” è un termine coniato come titolo di un workshop, tenuto presso il maker-space Wemake di Milano (wemake.cc) in ottobre 2017. Il suo significato riguarda le tecnologie per misurare in continuo l'ambiente intorno a noi, sia i fenomeni naturali, come ad esempio i parametri meteo, sia la qualità dell'aria, dell'acqua e della terra. Queste tecnologie possono rientrare nel campo più vasto dell'IoT (Internet of Things), tenendo conto però che vanno risolti i problemi tipici degli ambienti aperti e dell'assenza di reti WiFi.

### A QUALE SCOPO?

Alcuni fenomeni d'inquinamento antropico possono essere scoperti solo con un campionamento continuo nel tempo e la condivisione on-line dei dati. Infatti spesso le analisi anche più precise effettuate dalle autorità ambientali preposte non sono sufficienti a coprire nel tempo e nello spazio la vastità dei territori dove viviamo. Cittadini, opportunamente addestrati e dotati di strumentazione possono sopperire a questa mancanza. Esempi in tal senso si trovano in diverse parti del mondo.

L'idea di base è ben spiegata sul sito [www.coscienzambientale.com](http://www.coscienzambientale.com).

La tecnologia elettronica mette oggi a disposizione di qualsiasi cittadino materiali a basso costo, facilmente reperibili in Rete; inoltre molte persone condividono su Internet le loro conoscenze ed esperienze.

CoScienza Ambientale ([www.coscienzambientale.com](http://www.coscienzambientale.com) e su FB:) è un sito dove sono pubblicati documenti sulle esperienze di costruzione e sperimentazione di strumenti. E' aperto a tutti, secondo la prassi open source, come è giusto per tutti quei progetti dove ci si è avvalsi della conoscenza gratuita di molti maker sparsi per il mondo.

Se si vuole misurare la natura in modo continuo per accorgersi di eventi pericolosi come una frana o una esondazione o deprecabili come l'inquinamento, è necessario dotarsi di sensori e sistemi di trasmissione dati a distanza che ci allertino di quanto accade dove sta il sensore. Siccome stiamo in campo aperto, spesso l'unico sistema pubblico di trasmissione dati è costituito dalle reti GSM 2,5 G fino a 4 G. Certe volte neanche queste reti hanno copertura sufficiente. L'uso di queste reti comporta poi un canone mensile e acquisto di SIM per ogni sensore trasmettitore. Inoltre i modem GSM assorbono molta energia e quindi richiedono alimentazioni di rete o pannelli solari.

La tecnologia radio LoRa ci viene in aiuto, permettendoci trasmettitori piccoli, che possono funzionare per mesi a batteria, nessun costo di servizio, e soprattutto non hanno bisogno di reti esterne di terzi.

Quindi con la tecnologia LoRa possiamo costruirci la nostra rete e, dove è possibile avere una connessione Internet, collegarsi ad un Server per depositare i dati e renderli disponibili a tutti.

## GLOSSARIO

Spiegazione di alcuni termini che saranno adottati successivamente

**Microcontrollore:** è una scheda elettronica in grado di acquisire segnali elettrici dall'esterno e generarne altri verso l'esterno secondo una logica scritta in un programma caricato sulla sua memoria. Il programma si chiama anche firmware.

**Sensore:** E' un dispositivo che trasforma segnali fisici esterni in segnali elettrici che possono essere elaborati da un microcontrollore.

**Nodo trasmettitore:** Il nodo trasmettitore o nodo sensore è composto da uno o più sensori di cui sopra e da un microcontrollore che raccoglie questi segnali, li trasforma in numeri e li trasmette via radio.

**Nodo ricevitore:** E' composto da un microcontrollore in grado di ricevere i dati inviati dal nodo trasmettitore e renderli visibili all'utente, tramite un display o inviandoli sulla Rete Internet.

## Come possono funzionare i nodi trasmettitori di un sensore

In un contesto di “Outdoor Sensing” il nodo trasmettitore può essere programmato per funzionare in due modi che qui descriveremo.

### NODO TRASMETTITORE CON INVIO DATI A TEMPO

Il nodo trasmettitore di solito è posto nel luogo dove deve fare le misurazioni, spesso privo di rete elettrica, quindi deve essere alimentato da una batteria che duri il più possibile. A tal fine si cerca di attivare il nodo solo per il tempo necessario alla gestione dei sensori e alla trasmissione dei loro valori, cioè per pochi secondi o frazioni. Il tempo tra due misurazioni è invece più lungo, dell'ordine dei minuti. Questo si decide in base alla presunta variabilità della grandezza da misurare. Per esempio la temperatura dell'acqua di un torrente cambierà di poco nel corso del tempo e quindi si potrà misurare anche ogni ora; mentre la temperatura dell'aria dovrà essere misurata ogni 5, 10 minuti.

Durante il periodo di “riposo” tra due misure si può mettere tutto il nodo in condizioni di minimo assorbimento energetico, stato che si dice in “sleep”. Per far questo esistono alcune funzioni che cambiano a seconda del microcontrollore usato. Nel caso dell'Arduino Pro-Micro abbinato al modulo radio RFM95, o della scheda Adafruit Feather 32u4 LoRa, la libreria `Adafruit_SleepyDog.h` dispone della funzione `Watchdog.sleep(time)`; che permette di far dormire il processore per un tempo pari a `time` millisecondi. Purtroppo il massimo valore di `time` è solo 8000.

Si può ovviare a questo inconveniente ripetendo la chiamata alla funzione più volte, ma non con un loop di tipo `for`, infatti il processore, al risveglio, esce dal loop `for` e porta il controllo alla prima istruzione del ciclo `Loop`. E' quindi necessario scrivere la funzione `Watchdog.sleep(time)` un po' di volte in sequenza e, se servono tempi lunghi, creare un contatore di cicli `Loop` che possa essere confrontato ogni volta con un suo valore massimo, raggiunto il quale, si procede alla lettura del sensore e alla trasmissione dei suoi dati. Il metodo è descritto dallo schema a blocchi sottostante.

Assieme alla funzione di `sleep` del microcontrollore è bene chiamare quella analoga per il modulo radio `rf95.sleep()`; il modulo si risveglierà alla prima chiamata della funzione di trasmissione `rf95.send(...)`.

Nello schema si notano alcune istruzioni come leggere la batteria e incrementare il numero del messaggio. Si è infatti ipotizzato che il messaggio inizi con i seguenti parametri, uguali per tutte le applicazioni.

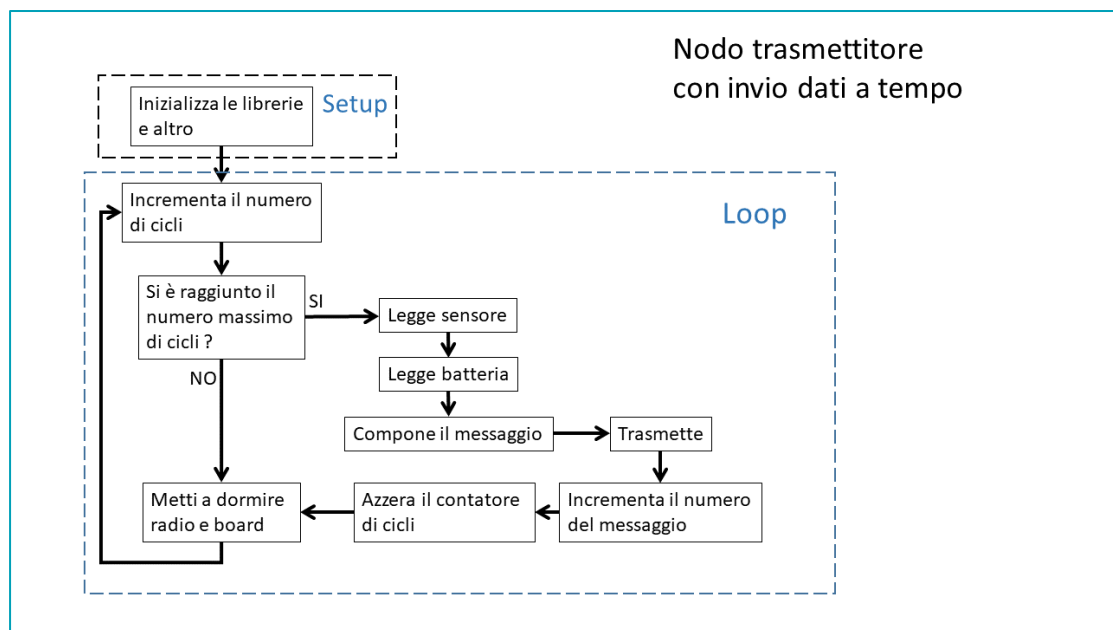
ID\_TX : numero intero senza segno(2 byte), di 5 cifre, identificativo del trasmettitore

Len: numero intero senza segno (2 byte) che indica il numero totale di byte del messaggio (payload)

vBatTx: numero intero senza segno (2 byte) con la tensione della batteria in mV

npacket: numero intero positivo (2 byte) con il numero progressivo del messaggio

Seguono poi le variabili tipiche dell'applicazione.



## NODO TRASMETTITORE CON INVIO DATI AD EVENTO

Può capitare di dover costruire un nodo che trasmetta il valore del sensore solo quando si presenta un evento particolare. Ad esempio un sensore, sensibile alla vibrazione prodotta da uno smottamento di terreno, dovrà attivare una trasmissione di allerta appena percepita la vibrazione, mentre nella maggior parte del tempo tutto il nodo sarà in stato di riposo.

In questo caso abbiamo bisogno di usare la funzione di `Interrupt` del nostro microcontrollore. Tale funzione, presente in quasi tutti i microcontrollori, consiste nel “sentire” se un piedino particolare (il 2 o il 3 nella ProMicro) ha cambiato il suo stato tramite un segnale elettrico esterno. In quel caso, istantaneamente, il microcontrollore sospende l’esecuzione delle istruzioni che stava processando e trasferisce il controllo ad una funzione appositamente predisposta e dichiarata nell’assegnazione dell’`Interrupt` (vedere in particolare la spiegazione dell’`Interrupt` su [Arduino.cc](http://Arduino.cc)). Finita l’esecuzione di

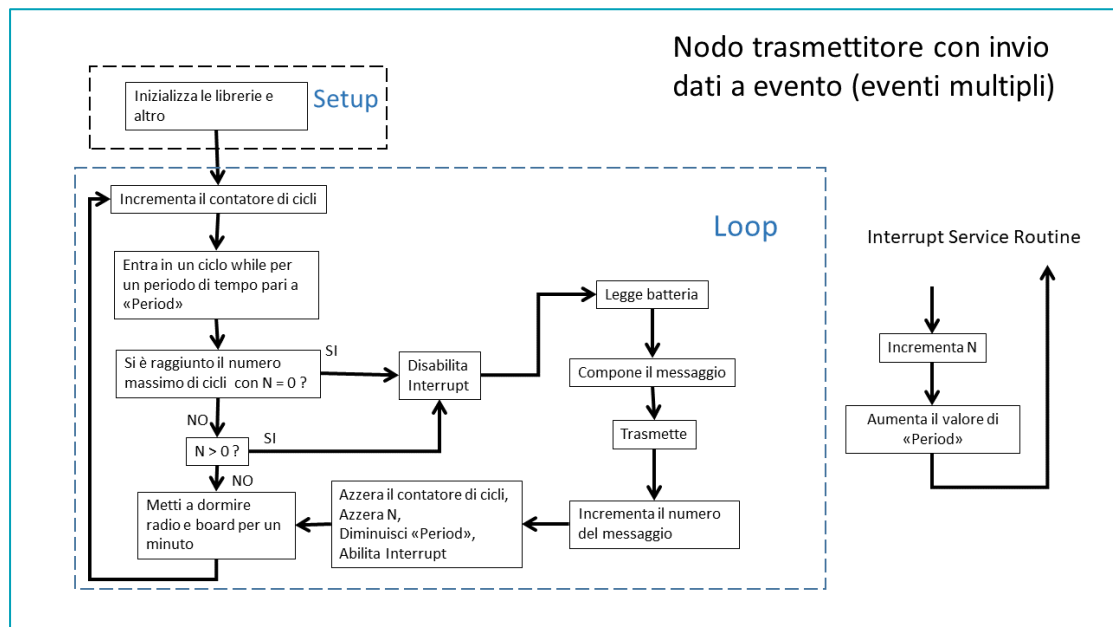


questa funzione, il controllo riprende dalle istruzioni che stava eseguendo prima dell'interrupt. Questa funzione particolare si chiama ISR (Interrupt Service Routine).

Che succede però se l'Interrupt avviene mentre il microcontrollore è dormiente? Nel caso particolare delle schede con chip 32u4 e della libreria `Adafruit_SleepyDog.h`, il microcontrollore viene svegliato dall'interrupt, viene eseguita la ISR indicata e poi il controllo passa alla prima istruzione del ciclo `Loop`. In altri microcontrollori invece succede che la CPU si azzeri (reset), quindi il controllo passa alla prima istruzione del setup.

Questo succede ad esempio con la scheda Adafruit Feather 32u4 e la ProMicroLora che hanno la stessa CPU 32u4. Un'applicazione di questa tecnica è stata implementata nel nodo trasmettitore sensibile alle frane.

Lo schema equivalente del nodo Frane è il seguente:



Lo sketch in figura mostra un ciclo principale che prevede un periodo di ascolto di impulsi di "Period" millisecondi (10000) e uno di "sleep" di un minuto. Quando arriva il primo impulso, si attiva la ISR che incrementa il contatore; finita la ISR, il controllo passa all'inizio del loop e per un periodo di 10 s aspetta altri impulsi contandoli. Finito il periodo, la CPU disattiva l'interrupt e trasmette il pacchetto contenente il numero di impulsi contati. Dopo la trasmissione l'interrupt viene riattivato. Un pacchetto viene comunque inviato dopo un certo numero di cicli "N" per segnalare che il nodo è sempre in vita.

## MINI RETE DI NODI LORA

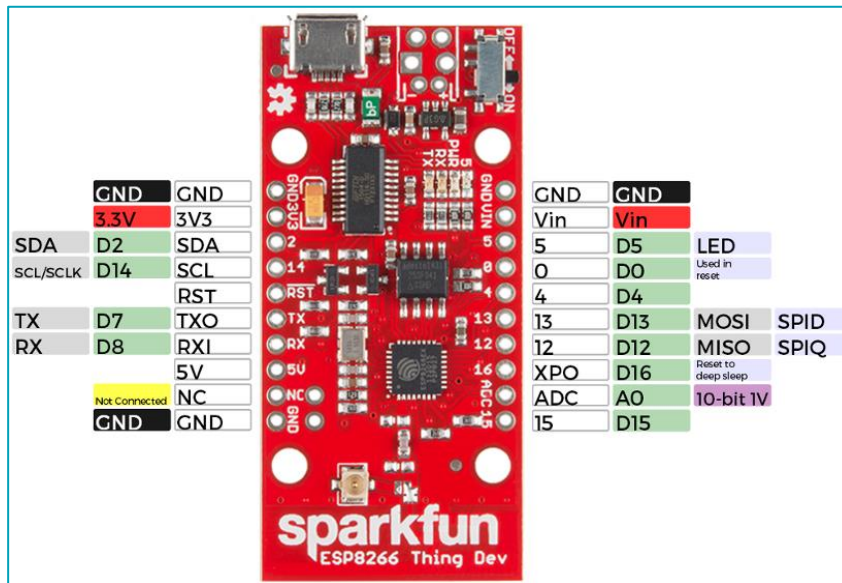
E' possibile creare una mini rete LoRa usando un certo numero di trasmettitori, tutti accordati sulla stessa frequenza, ed un unico ricevitore. Infatti se la periodicità di invio dati è lenta, p.e. alcuni minuti, è poco probabile che i pacchetti inviati dai trasmettitori interferiscano tra loro. Lo sketch del ricevitore riconoscerà i vari pacchetti arrivati dal numero del trasmettitore, codificato all'inizio del pacchetto, come è stato riportato in un paragrafo precedente. La lunghezza in byte del pacchetto è riportata nell'intestazione dello stesso così il ricevitore non deve nemmeno sapere quanto deve essere lungo.

## MINI PONTI RADIO

Se si necessita di allungare la portata della trasmissione, è possibile in linea teorica, visto che l'autore non lo ha ancor provato, costruire delle schede RX-TX con la funzione di ponte radio. Queste schede verrebbero poste lungo la linea tra TX e RX. Ogni scheda dovrà ricevere su una frequenza e trasmettere su un'altra (basta cambiare anche un decimo di MHz).

## Inserimento dei dati in una rete WiFi

Una trasmissione di dati via radio tra due schede ProMicroLoRa, risolve il problema dell'assenza di reti GSM/GPRS o WiFi dove è sistemato il sensore-trasmettitore. Ma se si vuole inviare i dati sulla Rete, per esempio per mandarli su un server, consultabile ovunque, è necessario che il nodo ricevitore sia collocato in un luogo dove è presente una connettività WiFi o una presa ethernet. Il nodo ProMicroLora non è in grado di connettersi ad una rete WiFi, ma ci sono schede in grado di farlo con le dovute librerie. Una di queste, provata dall'autore con successo, è la ESP8266 Thing Dev della Sparkfun, vedi figura in basso.



Il ricevitore, provato con successo dall'autore, è costituito da una scheda ProMicroLoRa, e una scheda ESP8266 collegate da tre fili, 2 per il protocollo seriale e uno per il ground.

La ESP8266 funziona con la libreria:

```
#include <ESP8266WiFi.h>
```

Tale libreria si trova assieme alle librerie della IDE di Arduino e viene presa dal compilatore dopo che si è selezionata la scheda ESP8266 Things Dev sul menù Strumenti/Schede. Per avere questa scheda sul menù, bisogna inserire nel menù File/impostazioni l'URL:

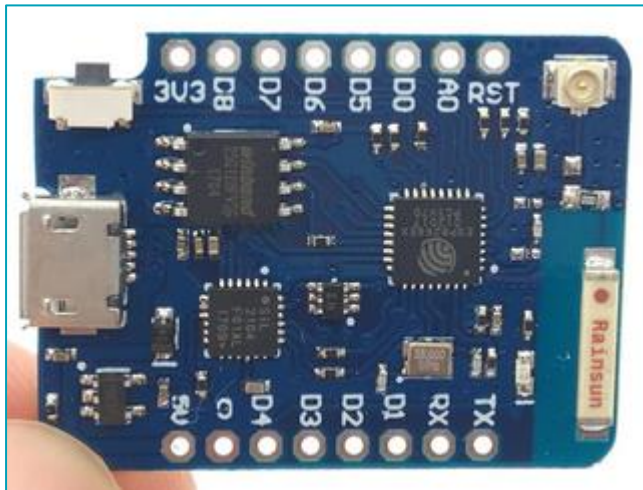
```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

Che provvede al collegamento alla Rete mediante la rete WiFi presente.

E' necessario poi dare istruzioni alla ESP per collegarsi ad un server dove riversare i dati. In generale ogni server richiede di ricevere i dati con un protocollo diverso. La funzione del server nel nostro caso è quella di archiviare i dati e presentarli graficamente. Dopo un esame non esaustivo tra i server disponibili, la scelta è caduta su [exosite.com](http://exosite.com) che offre gratuitamente alcuni servizi tra i quali l'archiviazione, la grafica, il retrieval e la generazione di mail di allarme. La figura mostra una schermata di questo server con i dati meteorologici inviati da una centralina realizzata con ProMicroLora. Lo stesso server fornisce una libreria ([Exosite.h](http://Exosite.h)) che facilita l'invio dei dati dalla scheda ESP8266.

## SCHEDA WIFI ALTERNATIVA

In alternativa alla Sparkfun descritta sopra, è possibile usare la scheda Wemos D1 Mini Pro 16 Mbytes che contiene lo stesso chip ESP8266. Lo sketch sviluppato per la scheda Sparkfun può essere caricato su questa scheda a patto di chiamare i pin digitali con la “D” davanti al numero. Per la compilazione con la IDE usare la board “LOLIN(WEMOS) D1 R2 & mini”

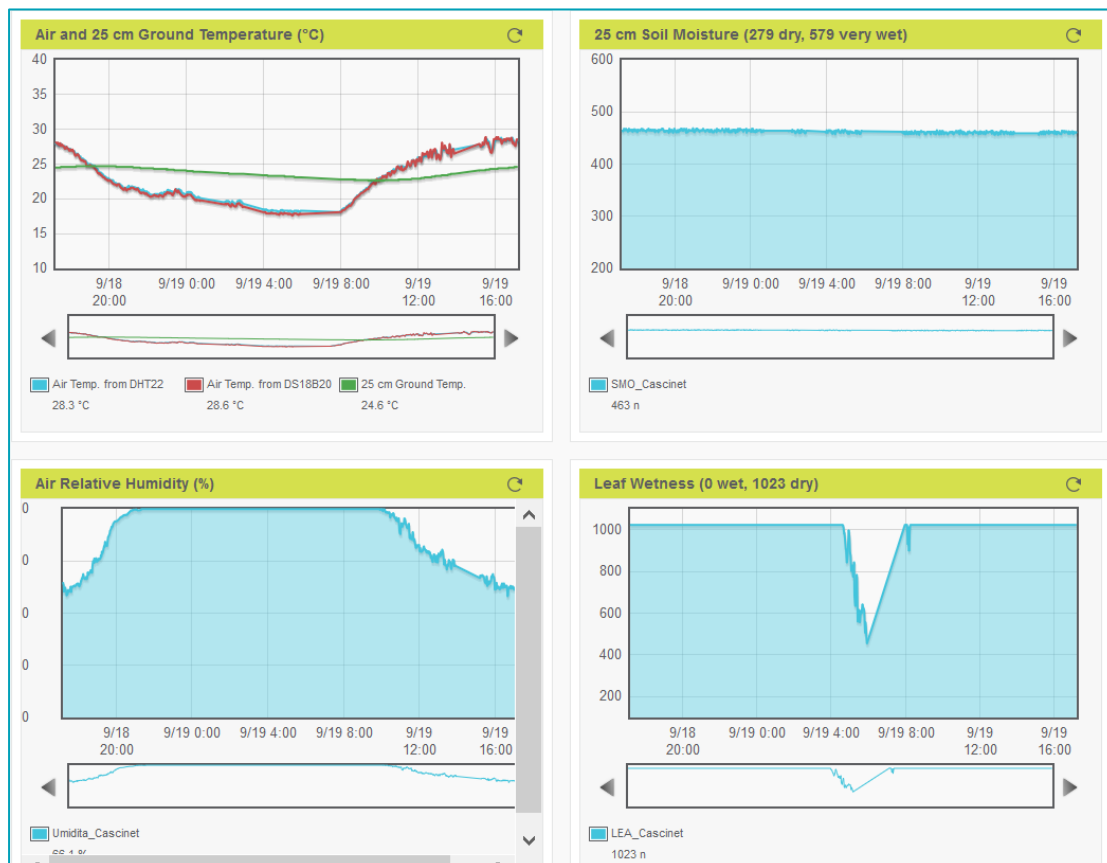


## SERVER EXOSITE

Il server exosite <https://portals.exosite.com/login> consente un accesso gratuito a qualsiasi utente registrato. Il server svolge la funzione di semplice database e di presentazione grafica dei dati. Consente l'estrazione dei dati in un paio di modalità. La figura sotto mostra un esempio di restituzione grafica di alcune serie temporali di dati.

Tutte le impostazioni che seguono possono essere fatte da un comune browser, per esempio l'autore ha usato Firefox.

Una volta collegati, le semplici istruzioni che vengono presentate consentono di configurare il server per ricevere i propri dati. Prima cosa da fare è definire un “Device” generico, questo corrisponde alla nostra scheda ESP8266. Una volta definito, il server genera una “CIK”, una stringa di numeri esadecimali, che costituirà la password, che permetterà alle stinghe http emesse dalla ESP8266 di entrare nel server ed essere interpretate. La CIK deve essere inserita nello sketch sulla scheda ESP8266.



Definito il “Device” si può passare a definire i “data-source” che contengono tutte le informazioni sul tipo di dati che arriveranno al server. Il data-source avrà un nome, ad esempio “Temperatura\_acqua” che deve corrispondere alla variabile portatrice del dato nelle stringa `http: Temperatura_acqua = 10.5`. In questo modo il server caricherà il valore 10.5, assieme alla data e ora di ricezione, nel suo database sotto la variabile Temperatura\_acqua. Si possono definire centinaia di data-source sotto una stesso Device. E’ bene invece non inviare dati da una stessa scheda ESP8266 a più Device perché il sistema probabilmente blocca i dati che non arrivano dallo stesso Mac Address.

**AVVERTENZA:**

La società EXOSITE dismetterà il portale <https://portals.exosite.com/login> A gennaio 2020, sostituendolo con il prodotto “Murano”. Consultare il sito [www.exosite.com](http://www.exosite.com) per ulteriori dettagli.

## Alcune note sui parametri di trasmissione

La trasmissione LoRa con il modulo RF95 prevede la definizione di alcuni parametri:

Bw: bandwidth; Cr: coding rate; Sf: spreading factor

E' possibile cambiare questi parametri per ottenere prestazioni diverse nella trasmissione. Ogni cambiamento sul TX dovrà essere riportato uguale sul RX. Nella tabella seguente sono riportate alcune combinazioni di questi parametri con il loro effetto sulla portata e velocità di trasmissione.

Enumerator	
Bw125Cr45Sf128	Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on. Default medium range.
Bw500Cr45Sf128	Bw = 500 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on. Fast+short range.
Bw31_25Cr48Sf512	Bw = 31.25 kHz, Cr = 4/8, Sf = 512chips/symbol, CRC on. Slow+long range.
Bw125Cr48Sf4096	Bw = 125 kHz, Cr = 4/8, Sf = 4096chips/symbol, CRC on. Slow+long range.

(Fonte: documentazione della libreria RadioHead)

Output power: potenza di uscita del segnale, va da 5 a 23 dBm

Frequenza: può essere variata intorno agli 868 MHz, prove sono state fatte positivamente cambiando alcuni decimali.

Una spiegazione sul Data Rate e Spreading Factor si trova a questo link.

[https://docs.exploratory.engineering/lora/dr\\_sf/](https://docs.exploratory.engineering/lora/dr_sf/)

## Il KIT per iniziare (Natale 2019)

Il KIT ha lo scopo di fare pratica sulla trasmissione dei dati mediante la tecnologia LoRa. Il KIT permette di assemblare senza la necessità di saldature:

- un nodo TX in grado di trasmettere temperatura e umidità dell'aria,
- un nodo RX in grado di ricevere i dati da uno o più TX e mostrarli su un display OLED, oltre che sul PC, se quest'ultimo è collegato tramite USB alla scheda e con il monitor seriale della IDE.

L'hardware disponibile nel KIT è il seguente:

2 x ProMicroLoRa pre-programmate, una per il nodo TX e l'altra per il nodo RX

2 x portabatteria 4 x AAA con connettore adatto alle schede (uno di questi possiede un partitore resistivo con un terminale che va collegato all'ingresso A9 del TX)

1 x sensore di temperatura ed umidità DHT22

- 1 x mini display OLED 128 x 32 punti
- 2 x antenne a filo
- 2 x mini breadboard
- 7 x jumper
- 1 x handbook “Outdoor Sensing with LoRa” (Italian)

## IL SOFTWARE PER IL KIT

I due sketch necessari al TX e RX sono stati pre-caricati sulle due schede, ma si possono vedere e scaricare al link:

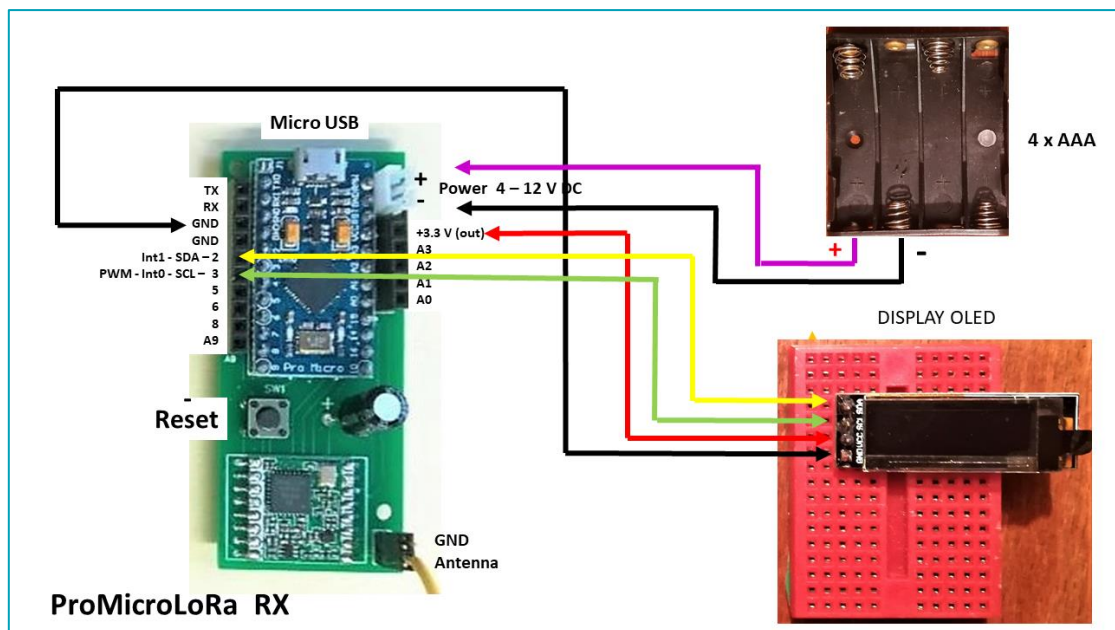
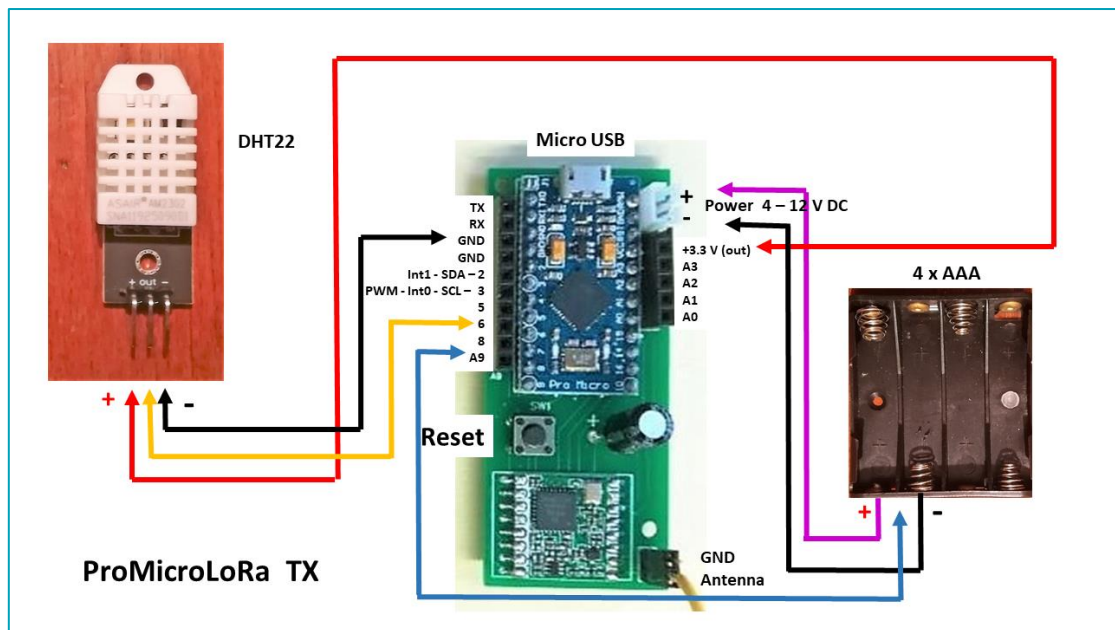
<https://github.com/paolometeo/Outdoor-Sensing>

oppure copiarli dal manuale “Mini Rete LoRa - il Software”.

Seguendo le istruzioni del manuale e usando la IDE di Arduino è possibile adattare gli sketch del KIT a nuovi progetti, come altri sensori sullo stesso nodo o più nodi TX. Inoltre è possibile, aggiungendo una scheda ESP8266, collegata alla ProMicroLoRa, di inviare i dati su un server internet.

Il software del KIT è spiegato nella pubblicazione “Mini Rete LoRa - Il Software”

## COME COLLEGARE LE VARIE PARTI DEL KIT



L'antenna in dotazione è saldata a due pin femmina, prima di inserirli nei due pin maschio della scheda, assicurarsi che il punto bianco sull'antenna sia dalla parte più esterna alla scheda (Antenna).