

Capire e usare

2



di PIER CALDERAN

Entriamo nel vivo imparando a utilizzare la tecnologia LoRaWAN in ambito IoT, attraverso la realizzazione di un nostro gateway a canale singolo ed uno multicanale.

Come abbiamo avuto modo di esporre nella scorsa puntata, la sola comunicazione LoRa consente una semplice ricetrasmisione punto-punto. Oggi parliamo di LoRaWAN, ovvero della tecnologia LoRa applicata al mondo IoT. Lo scopo principale sarà quello di poter gestire i dati provenienti dai nostri end device, visualizzandoli su un cloud. Successivamente, impareremo a creare delle interfacce grafiche per poter visualizzare i dati in modo piacevole, attraverso un browser di un qualsiasi computer o smartphone.

LPWAN

La tecnologia LoRaWAN rientra nell'ambito LPWAN (Low Power Wide Area Network), traducibile come rete estesa a lungo raggio a basso consumo. Si tratta quindi di un tipo di rete per telecomunicazioni senza fili progettata per consentire comunicazioni a basso bit-rate, senza un eccessivo consumo di energia, il che la rende ideale

per dispositivi alimentati a batteria. La bassa potenza, la ridotta velocità di comunicazione e l'uso non costante della trasmissione dei dati, consentono un notevole risparmio energetico. Una rete LPWAN può essere utilizzata per creare un network di sensori wireless privati, ma può anche essere un servizio o infrastruttura offerto da terze parti, che consente ai proprietari di sensori di implementarli sul campo senza eccessivi investimenti. Lo dimostrano le attività delle compagnie telefoniche come Swisscom, in Svizzera. Semplificando al massimo, possiamo riassumere i vantaggi di una rete LPWAN nei seguenti punti:

- lungo raggio: il raggio operativo della tecnologia LPWAN varia da qualche chilometro nelle aree urbane a oltre 10 km in contesti open-air; può consentire anche un'efficace comunicazione in luoghi interni e sotterranei;
- basso consumo: ottimizzati per un consumo energetico limitato, i ricetrasmittitori possono funzionare fino a 20 anni con batterie piccole ed economiche;

- basso costo: i protocolli semplificati e leggeri riducono la complessità nella progettazione dell'hardware e riducono i costi dei dispositivi;
- infrastruttura semplice: la lunga portata combinata con una tipologia di collegamento "a stella" riducono i costi di infrastruttura;
- bande di frequenza libere: con una rete LPWAN su piattaforma LoRaWAN, a differenza di altri competitor (ad esempio, SigFox), i costi di rete sono azzerati.

Nella prima puntata del corso abbiamo accennato al concetto di LoRaWAN come applicazione lato server, progettata per la raccolta e la gestione di dati provenienti da uno o più gateway, collegati a un numero indefinito di end device (dispositivi finali).

Per ridurre al minimo le collisioni rese possibili da un gran numero di dispositivi è necessaria, come vedremo, un'applicazione lato server in grado di eseguire funzioni MAC (Media Access Control) su una connessione Internet.

Una rete LoRaWAN è basata tipicamente su collegamenti a stella in cui i gateway fanno da tramite ai messaggi dei dispositivi finali verso un server di rete. Come vedremo, potremo utilizzare un cloud che mette a disposizione backend per la gestione dei dati attraverso applicazioni MAC. In altre parole, tutto quello che un gateway trasmette alla rete è incorporato e gestito nel cloud.

Quello che è importante tenere presente è che la tecnologia LoRaWAN è stata progettata principalmente per applicazioni in cui sono richiesti solo pochi messaggi da e verso i dispositivi. La comunicazione tra dispositivi finali e gateway è distribuita su diversi canali nella stessa banda di frequenze, utilizzando diverse velocità di trasmissione e di quantità dati. La scelta della velocità di trasmissione è un compromesso tra duty cycle (tempo di trasmissione) e durata del messaggio stesso.

Differenti velocità di trasmissione dei dati non interferiscono tra loro e creano un insieme di canali "virtuali" che aumentano la capacità dei gateway.

PIATTAFORME BASATE SU LORAWAN

Per fare qualche esempio, qui di seguito forniamo un elenco di piattaforme LoRaWAN disponibili in forma gratuita o a pagamento. In questo corso faremo riferimento alla piattaforma The Things Network e ciò per due validi motivi: il primo è che l'accesso ad essa è completamente gratuito e il secondo è che si tratta della piattaforma più diffusa fra gli sviluppatori di tutto il mondo.

- The Things Network: è un provider di rete LoRaWAN open source supportato da una comunità mondiale di oltre 100.000 sviluppatori;
- Globalsat: è un fornitore pubblico e privato di soluzioni per il mercato in Europa, Stati Uniti, Asia e Giappone;
- ThingsConnected; è una piattaforma gratuita fornita da UK Digital Catapult;

- ResIOT: è una piattaforma in parte gratuita per reti private, pubbliche e IoT;
- Everynet: fornisce una piattaforma e gateway per l'utilizzo nelle Americhe, in Europa, in Cina;
- ThingPark Wireless: è una piattaforma di Actility basata su LoRaWAN;
- LORIoT: operatore pubblico globale di LoRaWAN per reti private e pubbliche.

Altre parti interessate al mercato della LPWAN sono operatori di telefonia come Swisscom (Svizzera), Vodafone (Regno Unito) e Orange (Francia).

Per ragioni di spazio e di dinamica del mercato, in questo corso non verranno trattati i numerosi prodotti commerciali in ambito LoRa/LoRaWAN. Per un elenco aggiornato si veda la pagina di LoRa Alliance:

<https://lora-alliance.org/lorawan-certified-products>.

SICUREZZA LORAWAN

Per quanto riguarda la protezione e la sicurezza, sono stati introdotti livelli di crittografia con una chiave per il collegamento di rete, una chiave per l'applicazione e un ID MAC specifico del dispositivo.

LoRaWAN specifica un numero di chiavi di sicurezza:

- NwkSKey (chiavi di sessione);
- AppSKey (chiavi di sessione dell'applicazione);
- AppKey (chiavi dell'applicazione).

Tutte le chiavi hanno una lunghezza di 128 bit. L'algoritmo utilizzato allo scopo è l' AES-128, simile all'algoritmo di cifratura utilizzato nello standard 802.15.4.

Chiavi di sessione (NwkSKey e AppSKey)

Quando un dispositivo finale effettua una "unione" (join) o una "attivazione" (activation) alla rete, vengono generate dal cloud una chiave di sessione di rete (NwkSKey) e una chiave di sessione dell'applicazione (AppSKey).

La chiave NwkSKey è condivisa con la rete, mentre la chiave AppSKey è privata. Queste chiavi di sessione verranno utilizzate per la durata della sessione, ovvero per il tempo di collegamento ai dispositivi finali che gestiscono queste chiavi.

La chiave NwkSKey viene utilizzata per l'interazione tra il dispositivo finale e il server di rete. Serve a convalidare l'integrità di ciascun pacchetto mediante un codice di integrità del messaggio MIC (Message Integrity Control). Questo MIC è simile a un checksum, tranne per il fatto che impedisce la manomissione intenzionale di un messaggio. Per questo, LoRaWAN utilizza algoritmi AES-CMAC, che calcolano un codice MAC attraverso l'algoritmo di crittografia AES 128.

La chiave AppSKey viene utilizzata per la crittografia del payload. Il payload è il pacchetto LoRaWAN completamente crittografato tra il dispositivo finale e l'applicazione sul



cloud. Ciò significa che nessuno può leggere o decrittare il contenuto dei messaggi.

Queste due chiavi di sessione, NwkSKey e AppSKey sono uniche per ogni dispositivo finale e per ogni sessione. Più avanti spiegheremo come attivare dinamicamente il dispositivo con una configurazione OTAA (Over The Air Activation) in cui queste chiavi vengono rigenerate dal cloud ad ogni sessione e come attivare il dispositivo staticamente con una configurazione ABP (Activation By Personalization), che utilizza queste chiavi fino a quando non si decide di cambiarle nel proprio dispositivo finale.

La chiave dell'applicazione AppKey è nota solo al dispositivo e dall'applicazione. I dispositivi attivati dinamicamente con OTAA utilizzano la chiave dell'applicazione per derivare le due chiavi di sessione durante la procedura di attivazione. Con The Things Network si ottiene un valore predefinito AppKey che verrà utilizzato per attivare tutti i dispositivi o personalizzare la chiave AppKey del dispositivo.

Contatori di frame

Poiché stiamo lavorando con un protocollo radio, chiunque potrebbe intercettare i messaggi; in realtà questo non è possibile senza la chiave AppSKey, né è possibile manometterli senza la chiave NwkSKey, in quanto questo farebbe fallire il controllo di integrità MIC. È tuttavia possibile subire un cyber attacco che può ritrasmettere i messaggi (reply). Ma questi eventuali cyber attacchi possono essere rilevati e bloccati usando i cosiddetti Frame Counter (contatori di frame).

Quando un dispositivo è attivato, questi contatori di frame (FCntUp e FCntDown) sono entrambi impostati su 0; ogni volta che il dispositivo trasmette un messaggio di uplink (dal dispositivo finale al cloud), FCntUp viene incrementato e ogni volta che la rete invia un messaggio di downlink (dal cloud al dispositivo finale), FCntDown viene anch'esso incrementato. Se il dispositivo o la rete riceve un messaggio con un contatore dei frame inferiore a quello precedente, il messaggio viene scartato.

Questa misura di sicurezza ha conseguenze per i dispositivi di sviluppo, che spesso sono attivati staticamente in configurazione ABP. Quando si esegue questa operazione, è necessario rendersi conto che questi contatori di frame si reimpostano ogni volta che si riavvia il dispositivo, quando si esegue un nuovo firmware sul dispositivo o quando lo si scollega e ricollega al cloud. Di conseguenza, The Things Network bloccherà tutti i messaggi dal dispositivo fino a quando non FCntUp diventa più alto del precedente FCntUp. Pertanto, è necessario registrare nuovamente il dispositivo nel back-end ogni volta che lo si reimposta.

In altre parole, bisogna azzerare il Frame Counter.

Affidabilità del segnale

Molti potrebbero chiedersi quanto sia affidabile il segnale LoRa in termini di robustezza. Come abbiamo avuto modo di

vedere, la tecnologia utilizzata in LoRa è a spettro diffuso di tipo CHIRP (Compressed High Intensity Radar Pulse), usato nei radar. L'acronimo assomiglia molto alla nostra traduzione di "cinguettio" radio che tratta l'impulso ad alta intensità comprimendolo su uno spettro di frequenze. In questo modo il segnale radio viene impiegato, tra le altre cose, per l'elaborazione del guadagno del segnale, l'immunità alle interferenze, la condivisione dei canali virtuali e la resistenza alle riflessioni radio. In parole povere, è un impulso con tanta energia elaborata in pochissimo tempo per la trasmissione di pochi dati.

CANALI LORAWAN

Per la rete LoRaWAN in Europa si usa la banda libera sub-gigahertz ISM (Industrial Scientific Medical) compresa fra 863 e 870 MHz e anche la banda (meno usata) sui 433 MHz. Questa banda è regolamentata dalla normativa Europea che ne limita la potenza di trasmissione ed il tempo di trasmissione o duty-cycle.

Nella banda da 868,00 a 868,60 abbiamo il primo canale di 125 KHz di LoRaWAN centrato su 868,1. Poi abbiamo altri due ulteriori canali predefiniti centrati su 868,30 e 868,50 MHz. Questi canali predefiniti vengono utilizzati globalmente dalla piattaforma LoRaWAN.

Mentre per altre piattaforme LPWAN il duty-cycle complessivo della banda può limitare la trasmissione di dati, per LoRaWAN, la situazione cambia:

- il duty-cycle viene gestito per ogni canale e il dispositivo finale usa i canali uno dopo l'altro;
- il duty-cycle viene diviso per il numero di canali nella stessa banda di frequenza.

Poiché la configurazione standard ha tre canali nella stessa sotto banda, il duty-cycle di ciascun canale è ovviamente pari a 0,33% del totale. Tuttavia, se vengono assegnati alcuni canali a un'altra banda, per esempio da 867,10 MHz

 **Tabella 1 - Canali LoRaWAN.**

CANALE	FREQUENZA (HZ)	BANDA
0	868.100.000	g
1	868.300.000	g
2	868.500.000	g
3	867.100.000	g
4	867.300.000	g
5	867.500.000	g
6	867.700.000	g
7	867.900.000	g
8 (FSK)	868.800.000	g2



Fig. 1 - Il gateway iC880A-SPI di Wireless Solutions.

a 867,90 MHz, si può impostare un duty cycle del 1% su questi canali. Quindi il dispositivo finale può comunicare effettivamente con un duty cycle di $3 \times 0,33\%$ su una banda più 1% sull'altra banda. In pratica è possibile comunicare con un duty-cycle fino al 3%, utilizzando 8 canali LoRaWAN. La **Tabella 1** mostra gli 8 canali (da 0 a 7) comunemente usati in banda g per la comunicazione LoRaWAN e un canale FSK per la banda g2.

IL GATEWAY LORAWAN

Il cuore di una rete LoRaWAN è il cosiddetto gateway o concentratore (concentrator, in inglese). Si tratta di un dispositivo in grado di "concentrare" tutti i pacchetti LoRaWAN inviati dai vari dispositivi finali. A seconda dei modelli, un gateway LoRaWAN può essere a canale singolo (Single Channel) oppure multicanale (Multi Channel), fino a 8 canali. Per facilitare la vostra scelta nell'acquisto, faremo riferimento a due prodotti specifici.



Fig. 2 - Gateway a canale singolo LoRaWAN della Futura Elettronica.

1. Gateway a 8 canali iC880A di Wireless Solutions
 2. Gateway a canale singolo LoRaWAN di Futura Elettronica
- La differenza sostanziale fra un gateway a 8 canali e uno a canale singolo sta nel chip che viene usato: il gateway a 8 canali usa il chip Semtech SX1301, il gateway a canali singolo usa il chip Semtech SX1276.

A questo punto è facile capire che potete costruirvi il vostro gateway utilizzando lo stesso breakout del dispositivo finale che abbiamo già conosciuto nella scorsa puntata, ovvero il modulo Hoperf RFM95W, che monta il chip Semtech SX1276, esattamente come abbiamo fatto noi per il nostro gateway a canale singolo.

Il Gateway a 8 canali che useremo è siglato iC880A-SPI (**Fig. 1**) ed è costruito dall'azienda tedesca Wireless Solutions (<https://wireless-solutions.de>); è equipaggiato con il chip Semtech SX1301 e il suo costo, completo di accessori e spedizione si aggira, al momento della pubblicazione di questo corso, intorno ai 160 euro.

Il gateway a singolo canale che useremo è costruito dalla Futura Elettronica ed è disponibile anche on-line su www.futurashop.it (**Fig. 2**); questo dispositivo è basato sul chip Semtech SX1276.

GATEWAY MULTICANALE IC880A

Iniziamo con questo gateway per illustrare tutte le funzionalità fornite dagli 8 canali LoRaWAN.

Diamo uno sguardo alle sue caratteristiche principali.

- Tecnologia di modulazione LoRa.
- Banda di frequenza 868MHz.
- Fattori di diffusione ortogonali.
- Sensibilità fino a -138 dBm.
- Interfaccia SPI.
- Processore Semtech SX1301.

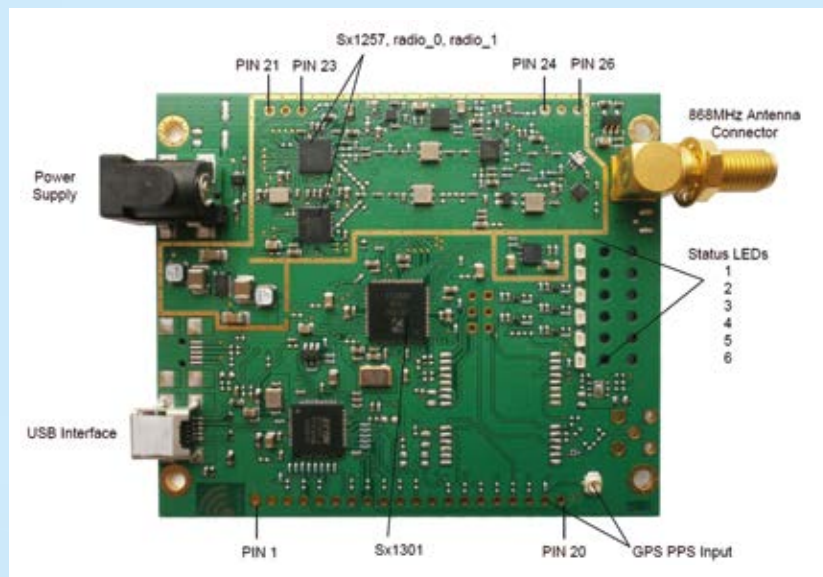


Fig. 3 - La piedinatura del modulo iC880A.

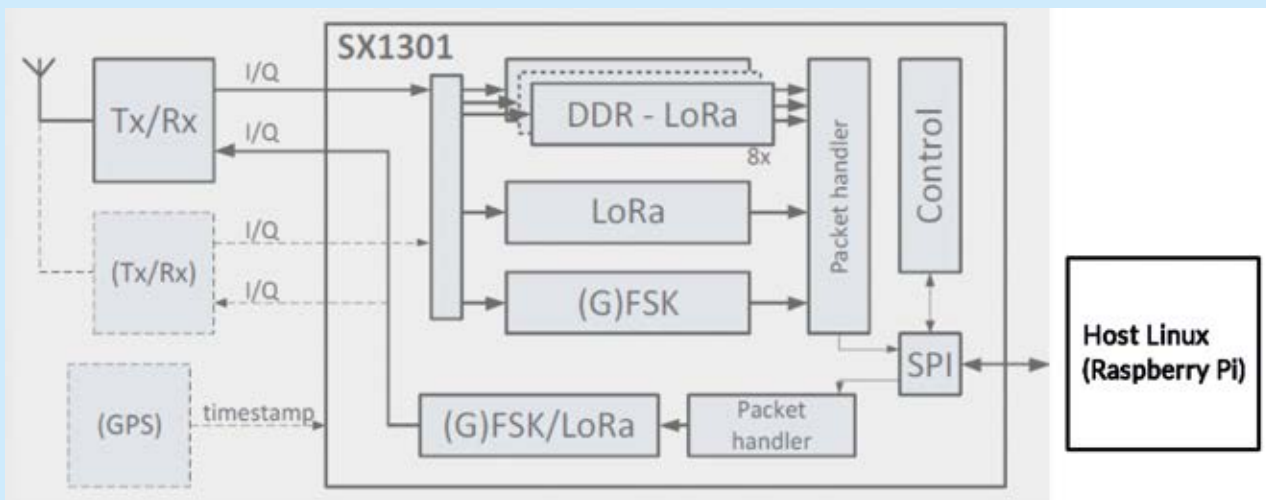


Fig. 4 - Schema a blocchi del chip SX1301.

- Emula 49 demodulatori LoRa.
- 10 percorsi di demodulazione paralleli.
- 1 demodulatore (G)FSK.
- 2 x front-end SX1257 Tx/Rx.
- Tensione di alimentazione 5 V.
- Interfaccia RF ottimizzata a 50 ohm.
- Livello di potenza in uscita fino a 20 dBm.
- Ricevitore GPS (opzionale).
- Portata fino a 15 km (LOS, Line Of Sight).
- Portata di diversi km in ambiente urbano.
- LED di stato.
- HAL disponibile da https://github.com/Lorantet/lora_gateway.
- Dimensioni: 79,8 x 67,3 mm.

La Fig. 3 illustra il circuito stampato dell'iC880A, lato componenti. Per l'interfaccia principale ci sono i pin header da 1 a 20 che andranno collegati ad un host Linux (per noi sarà una scheda Raspberry Pi).

Sono presenti i connettore SMA e uFL per collegare un'antenna tarata per gli 868 MHz.

La scheda offre alcuni LED per fornire un riscontro visivo dello stato operativo corrente. Il significato reale dei LED dipende comunque dal software di gestione. Come impostazione predefinita nella versione 1.7 l'impostazione dei LED indicati nella Fig. 3 è:

- LED 1: pacchetto di Backhaul;
- LED 2: pacchetto TX;
- LED 3: pacchetto sensore RX;
- LED 4: pacchetto RX FSK;
- LED 5: buffer RX;
- LED 6: alimentazione.

IL CHIP SEMTECH SX1301

Diamo uno sguardo a come funziona il chip Semtech SX1301.

Guardando lo schema di Fig. 4, nella parte ricevente, riceve i flussi di bit digitalizzati I/Q (Inphase/Quadrature) per uno o due ricevitori (chip SX1257), demodula questi segnali usando i demodulatori, adattando le impostazioni dei demodulatori al segnale ricevuto e memorizza i pacchetti demodulati ricevuti in un FIFO per essere recuperati da un sistema host (per esempio, Raspberry Pi).

L'SX1301 ha un blocco di controllo interno che riceve microcodice dal sistema host. Il microcodice viene fornito da Semtech come file binario che viene caricato nell'SX1301 all'accensione.

Il controllo dell'SX1301 da parte del sistema host viene effettuato utilizzando un HAL (Hardware Abstraction Layer). Il codice sorgente HAL viene fornito da Semtech e può essere adattato dallo sviluppatore. Consigliamo vivamente di utilizzare l'ultimo HAL fornito da Semtech a questo link <https://github.com/Lora-net>.

HOST LINUX

Il codice fornito da Semtech è stato progettato per host Linux, quindi per sviluppare il nostro progetto utilizzeremo quale host una scheda Raspberry Pi; può andare bene uno dei seguenti modelli: 2B/3B/3B+/4B con sistema operativo Raspbian o Raspberry Pi OS.

COLLEGAMENTO CON RASPBERRY PI

L'iC880A possiede un'interfaccia SPI da collegare alla medesima interfaccia di Raspberry Pi. Per stabilire una connessione SPI tra iC880A e Raspberry Pi, devono essere utilizzati i pin descritti nella Tabella 2.

NOTA: per un avvio stabile del chip SX1301, il pin di reset deve essere portato HIGH per 100 ns (minimo), una volta che la tensione di alimentazione è stabile. Più avanti spieghiamo come si effettua facilmente questa operazione.

I pin SPI (da 14 a 17) dell'iC880A devono essere collegati

Tabella 2 - Collegamento dell'iC880A con Raspberry Pi.

PIN IC880A	NOME DEL PIN	FUNZIONE DEL PIN	DESCRIZIONE
21	VDD	Power	Alimentazione +5 V (700 mA o più)
22	GND	Power	Massa
14	CLK	Input	SX1301 SPI-Clock
15	MISO	Output	SX1301 SPI-MISO
16	MOSI	Input	SX1301 SPI-MOSI
17	NSS	Input	SX1301 SPI-NSS (o CS)
13	Reset	Reset	resetta il chip

all'interfaccia SPI di Raspberry Pi, cioè ai pin 19, 21, 23 e 24. I pin di alimentazione 21 e 22 dell'iC880A devono essere collegati a una sorgente di alimentazione che sia in grado di fornire almeno 700 mA.

Si raccomanda pertanto di scegliere un alimentatore adeguato per fornire energia a sufficienza sia al modulo iC880A che a Raspberry Pi.

Può essere usato un qualsiasi pin per il reset. Nel nostro caso abbiamo usato il pin 11 di Raspberry Pi, contrassegnato come GPIO17.

La **Fig. 5** illustra il collegamento fisico dei pin della porta GPIO di Raspberry Pi all'iC880A.

Per il collegamento dell'antenna si può usare un'antenna con attacco SMA e un pigtail da u.fl a SMA, come illustrato in **Fig. 6**. Per comodità si consiglia di inserire tutto in una scatola di plastica.

DRIVER/HAL

Dopo aver assemblato il gateway con Raspberry Pi e iC880A,



Fig. 5 - Il collegamento tra Raspberry Pi e iC880A.

è giunto il momento di farlo funzionare.

Semtech fornisce un driver/HAL open source per soluzioni LoRa basate su SX1301. Il repository Semtech è ospitato sul GitHub al seguente indirizzo: <https://github.com/Lora-net>. In questo repository si troverà un'implementazione HAL denominata `lora_gateway` (https://github.com/Lora-net/lora_gateway).

Una volta qui, bisogna effettuare il download tramite il seguente comando git dal terminale di Raspbian:

```
git clone https://github.com/Lora-net/lora_gateway.git
```

Attenzione: il driver della Semtech richiede un'interfaccia SPI "nativa" di Raspberry Pi per cui bisogna assicurarsi che sia abilitata l'interfaccia SPI nella finestra di configurazione di Raspbian.

PIN DI RESET

Come già spiegato, è necessario impostare la logica di controllo per il pin di reset dell'iC880A. Il metodo più semplice è scrivere un piccolo script.

Lo script deve resettare l'iC880A prima che il driver LoRa possa accedere all'hardware.

A titolo di esempio forniamo lo script sul pin GPIO 17 (pin fisico 11) di Raspberry Pi collegato al pin di reset (13) dell'iC880A:

```
#!/bin/bash
echo 17 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio17/direction
echo 1 > /sys/class/gpio/gpio17/value
sleep 5
echo 0 > /sys/class/gpio/gpio17/value
sleep 1
echo 17 > /sys/class/gpio/unexport
echo "reset complete!"
```

Si fa notare che nello script abbiamo inserito dei ritardi notevoli fra un'operazione e l'altra per dare il tempo sufficiente a un reset efficace.

Chiameremo questo file di testo `reset.sh`. Si dovrà eseguire questo script a ogni avvio del sistema.

Per rendere eseguibile lo script basta digitare dal terminale il seguente comando:

```
sudo chmod +x reset.sh
```

Fig. 6 - L'antenna 868 MHz e il pigtail (adattatore) da U.FL a SMA.



Per eseguire lo script dal terminale basta digitare il comando:

```
sudo ./reset.sh
```

Per rendere automatica l'esecuzione dello script a ogni avvio, si può modificare il file `rc.local` in questo modo:

```
sudo nano /etc/rc.local
```

... aggiungere le righe seguenti (supponendo che lo script si trovi in `/home/pi`):

```
cd /home/pi
sudo ./reset.sh &
exit 0
```

COMPILAZIONE DELLA LIBRERIA LORAWAN

Per poter utilizzare il driver/HAL è necessario compilare la libreria `libloragw` che si trova nella cartella `lora_gateway` che abbiamo clonato nella directory `home/pi` di Raspbian. Per fare questo, si userà il compilatore `gcc/g++` e l'utilità `make`, già presenti in Raspbian. Da `home/pi/`, entrare nella cartella `lora_gateway` ed eseguire il comando `make`:

```
cd lora_gateway
make
```

Questo compilerà la libreria e tutte delle utilità di test, situate nelle sottocartelle della cartella `lora_gateway`.

Esempi nella cartella `lora_gateway`

Nella cartella `lora_gateway` vengono compilate alcune utility di esempio per illustrare l'uso della libreria `libloragw` con il gateway. Diamo una breve descrizione delle utility senza entrare nel dettaglio:

`util_pkt_logger`

Questa utility mostra come utilizzare il gateway come data logger. In pratica, tutti i pacchetti ricevuti dai dispositivi finali vengono salvati in un file locale in formato CSV (si veda più avanti).

`util_spi_stress`

Questa utility può essere utilizzata per verificare l'affidabilità del collegamento tra l'host e il file di registro del gateway.

`util_tx_test`

Questa utility può essere utilizzata per inviare pacchetti di prova con il gateway. I pacchetti contengono poche informazioni, ma possono essere utilizzati per valutare la funzionalità di un downlink del gateway.

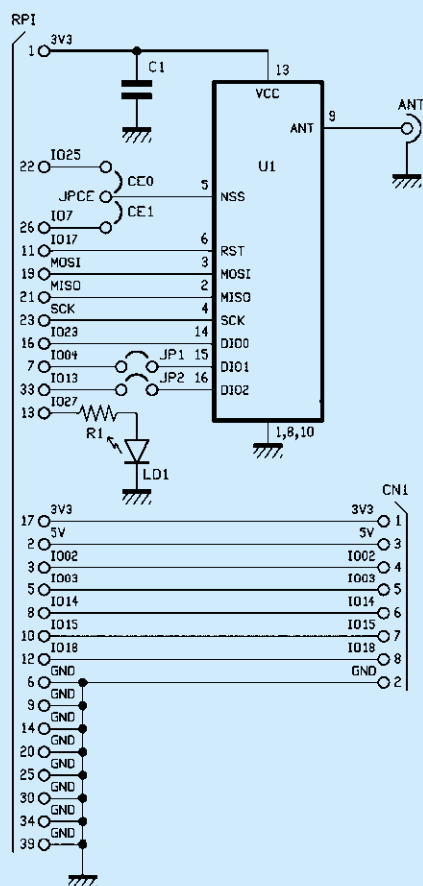
`util_tx_continuous`

Questa utility può essere utilizzata per impostare il

Listato 1

```
{
  "SX1301_conf": {
    "lorawan_public": true,
    "clksrc": 1, /* radio_1 provides clock to concentrator */
    "radio_0": {
      "enable": true,
      "type": "SX1257",
      "freq": 867500000,
      "rssi_offset": -166.0,
      "tx_enable": false
    },
    "radio_1": {
      "enable": true,
      "type": "SX1257",
      "freq": 868500000,
      "rssi_offset": -166.0,
      "tx_enable": false
    },
    "chan_multiSF_0": {
      /* Lora MAC channel, 125kHz, all SF, 868.1 MHz */
      "enable": true,
      "radio": 1,
      "if": -400000
    },
    "chan_multiSF_1": {
      /* Lora MAC channel, 125kHz, all SF, 868.3 MHz */
      "enable": true,
      "radio": 1,
      "if": -200000
    },
    "chan_multiSF_2": {
      /* Lora MAC channel, 125kHz, all SF, 868.5 MHz */
      "enable": true,
      "radio": 1,
      "if": 0
    },
    "chan_multiSF_3": {
      /* Lora MAC channel, 125kHz, all SF, 867.1 MHz */
      "enable": true,
      "radio": 0,
      "if": -400000
    },
    "chan_multiSF_4": {
      /* Lora MAC channel, 125kHz, all SF, 867.3 MHz */
      "enable": true,
      "radio": 0,
      "if": -200000
    },
    "chan_multiSF_5": {
      /* Lora MAC channel, 125kHz, all SF, 867.5 MHz */
      "enable": true,
      "radio": 0,
      "if": 0
    },
    "chan_multiSF_6": {
      /* Lora MAC channel, 125kHz, all SF, 867.7 MHz */
      "enable": true,
      "radio": 0,
      "if": 200000
    },
    "chan_multiSF_7": {
      /* Lora MAC channel, 125kHz, all SF, 867.9 MHz */
      "enable": true,
      "radio": 0,
      "if": 400000
    },
    "chan_Lora_std": {
      /* Lora MAC channel, 250kHz, SF7, 868.3 MHz */
      "enable": true,
      "radio": 1,
      "if": -200000,
      "bandwidth": 250000,
      "spread_factor": 7
    },
    "chan_FSK": {
      /* FSK 50kbps channel, 868.8 MHz */
      "enable": true,
      "radio": 1,
      "if": 300000,
      "bandwidth": 125000,
      "datarate": 50000
    }
  },
  "gateway_conf": {
    "gateway_ID": "AA555A0000000000"
  }
}
```

schema ELETTRICO



concentratore in modalità TX continua.

util_spectral_scan

Questa utility può essere utilizzata per eseguire la scansione della banda spettrale in background, dove opera il gateway.

util_lbt_test

Questa utility può essere utilizzata per testare i timestamp lbt (Listen Before Talk).

Come utilizzare l'utility util_pkt_logger

Fra i vari test proposti da Semtech abbiamo scelto questa utility che serve a dimostrare la capacità di ricezione dei pacchetti LoRa e di salvarli in un file CSV. Ad ogni avvio dell'utility, viene dato un nome univoco al file CSV contenente il MAC address del gateway e la data. I file possono essere aperti e controllati in seguito con un editor di testo o convertiti in un foglio di calcolo. Questa utility ha bisogno di un file json di configurazione con i parametri relativi all'operazione RX del concentratore. Si consiglia di dare un'occhiata al file `global_conf.json`, nella stessa cartella. Un esempio di come impostare il file è illustrato nel **Listato 1**.

All'inizio del file ci sono alcune impostazioni generali riguardanti la configurazione dell'SX1301, nel blocco `SX1301_conf`. L'impostazione del clock interno non deve essere modificata. Dopo c'è un blocco di configurazione per i due chip radio SX1257 del gateway. Ogni canale radio può essere attivato/disattivato e funziona su una certa frequenza di base (si vedano i blocchi `radio_0` e `radio_1`). Le impostazioni "type" dei due blocchi radio devono essere mantenute invariate.

I blocchi che seguono definiscono i parametri utilizzati dagli 8 canali virtuali `chan_multi-SF` con Spreading Factor variabile. Ogni configurazione logica del canale è composta dal chip radio da utilizzare e dalla frequenza RF. La frequenza RF è definita come differenza rispetto alla frequenza di base definita nel corrispondente blocco di configurazione radio. Il canale `chan_Lora_std` è fissato su SF7.

Il blocco `gateway_conf` contiene il `gateway_ID` impostato come MAC address `AA555A0000000000`. Per ulteriori dettagli si consiglia di consultare la documentazione dell'iC880A, presso il sito web della Wireless Solutions. Dopo aver controllato la configurazione, l'utility può essere avviata tramite questi comandi:

```
cd lora_gateway/util_pkt_logger
./util_pkt_logger
```

All'avvio dell'utility, ogni pacchetto LoRa che viene ricevuto dal concentratore verrà salvato in un file CSV, nella stessa cartella. La **Fig. 7** illustra la finestra del terminale con i dati della configurazione del gateway e i dati relativi al pacchetto che viene scritto nel file CSV (ultima riga della figura). Il nome del file contiene l'ID del gateway, ovvero `AA555A0000000000` seguito dalla data di ricezione del pacchetto, in formato iso `yyyymmddThhmmssZ`, ... ovvero: `20200710 (10 luglio 2020) 211610 (ore 21:16:10)`.

`pktlog_AA555A00000000101_20200710T211610Z.csv`

GATEWAY A SINGOLO CANALE LORAWAN

In alternativa al gateway a otto canali iC880A appena descritto, si può iniziare la sperimentazione con il gateway a singolo canale prodotto dalla Futura Elettronica.

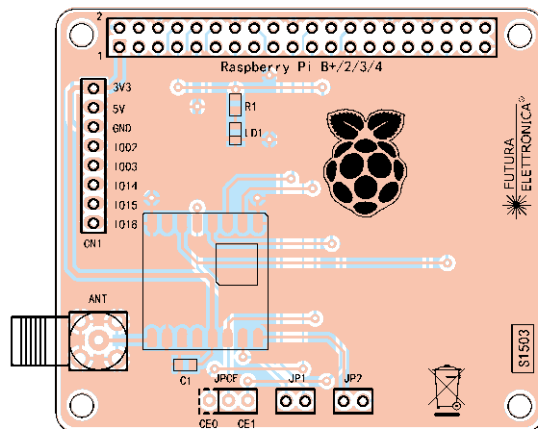
È sicuramente meno performante, ma costa decisamente molto meno e pertanto lo consigliamo a chi vuole iniziare a sperimentare nel mondo LoRaWAN.

Il gateway a singolo canale è di tipo HAT (Hardware Attached on Top) per Raspberry Pi ed è meglio descritto dallo schema elettrico che trovate in questa pagina.

Come vedete, il circuito è essenziale perché ospita un LED di segnalazione con relativa resistenza di limitazione della corrente e l'U1 altro non è che il modulo (che si trova in commercio già pronta) basato sull'SX1276.

La stessa hat ripete una serie di linee della Raspberry Pi. Questo gateway è basato sostanzialmente sulla breakout

piano di MONTAGGIO



R1: 330 ohm 0805
C1: 100 nF multistrato 0805
LD1: Led verde 0805
U1: Modulo LoRa RFM95 868 MHz

Varie:
- Strip M/F lungo 20 + 20
- Strip M 8 poli
- Strip M 3 poli

- Jumper
- Connettore SMA da CS
- Circuito stampato S1503 (65x55 mm)

Hoperf RFM95W, che nello schema è appunto U1, la quale è incentrata sul chip Semtech SX1276, lo stesso che abbiamo usato per i dispositivi finali. Le caratteristiche di tale chip lo rendono perfettamente compatibile con le specifiche LoRaWAN. Per quanto riguarda la realizzazione, il gateway si

costruisce a partire dal circuito stampato, del quale trovate le tracce sul nostro sito www.elettronica.in insieme ai file di questo numero della rivista; la basetta è a doppia faccia, quindi se non ve la sentite di realizzarla sapete che potete comunque acquistare il tutto in kit. In queste pagine vedete il

```
pi@raspberrypi: ~/lora_gateway/util_pkt_logger
File Modifica Schede Aiuto
pi@raspberrypi:~/lora_gateway/util_pkt_logger $ ./util_pkt_logger
loragw_pkt_logger: INFO: found global configuration file global_conf.json, trying to parse it
loragw_pkt_logger: INFO: global_conf.json does contain a JSON object named SX1301_conf, parsing SX1301 parameters
loragw_pkt_logger: INFO: lorawan_public 1, clksrc 1
loragw_pkt_logger: INFO: radio 0 enabled (type SX1257), center frequency 867500000, RSSI offset -166.000000, tx enabled 0, tx_notch_fr
eq 0
loragw_pkt_logger: INFO: radio 1 enabled (type SX1257), center frequency 868500000, RSSI offset -166.000000, tx enabled 0, tx_notch_fr
eq 0
loragw_pkt_logger: INFO: LoRa multi-SF channel 0 enabled, radio 1 selected, IF -400000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 1 enabled, radio 1 selected, IF -200000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 2 enabled, radio 1 selected, IF 0 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 3 enabled, radio 0 selected, IF -400000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 4 enabled, radio 0 selected, IF -200000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 5 enabled, radio 0 selected, IF 0 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 6 enabled, radio 0 selected, IF 200000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa multi-SF channel 7 enabled, radio 0 selected, IF 400000 Hz, 125 kHz bandwidth, SF 7 to 12
loragw_pkt_logger: INFO: LoRa standard channel enabled, radio 1 selected, IF -200000 Hz, 250000 Hz bandwidth, SF 7
loragw_pkt_logger: INFO: FSK channel enabled, radio 1 selected, IF 300000 Hz, 125000 Hz bandwidth, 50000 bps datarate
loragw_pkt_logger: INFO: global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
loragw_pkt_logger: INFO: gateway MAC address is configured to AA555A0000000000
loragw_pkt_logger: INFO: found local configuration file local_conf.json, trying to parse it
loragw_pkt_logger: INFO: local_conf.json does not contain a JSON object named SX1301_conf
loragw_pkt_logger: INFO: local_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
loragw_pkt_logger: INFO: gateway MAC address is configured to AA555A0000000000
loragw_pkt_logger: INFO: concentrator started, packet can now be received
loragw_pkt_logger: INFO: Now writing to log file pktlog_AA555A0000000000_20200710T212018Z.csv
```

Fig. 7 - Il terminale con l'output dell'utility util_pkt_logger.

```

pi@raspberrypi: ~/Desktop/single_chan_pkt_fwd
File Modifica Schede Aiuto
pi@raspberrypi:~/Desktop/single_chan_pkt_fwd $ ./single_chan_pkt_fwd
SX1276 detected, starting.
Gateway ID: b8:27:eb:ff:ff:8a:19:e7
Listening at SF7 on 868.100000 Mhz.
-----
stat update: {"stat":{"time":"2020-07-11 14:59:17 GMT","lati":0.00000,"long":0.00000,"alti":0,"rxnb":0,"rxok":0,"rxfw":0,"ackr":0.0,"dwnb":0,"txnb":0,"pfrm":"Single Channel Gateway","mail":"","desc":""}}

```

Fig. 8 - L'output sul terminale del Single Channel Packet Forwarder.

relativo piano di montaggio con lista dei componenti. L'uso di un solo canale predefinito può essere limitante, ma questo non inficia la possibilità di comunicare con decine di dispositivi finali, se non addirittura centinaia. L'uso di un solo canale LoRaWAN, influirà, per esempio, sul contatore dei frame (Frame Counter).

A differenza del gateway iC880A, qui non è possibile usare la libreria *libloragw* di Semtech, ma solo una parte del codice elaborato da Thomas Telkamp (con autorizzazione di Semtech) e reso disponibile nel suo repository GitHub: https://github.com/tftelkamp/single_chan_pkt_fwd. Si tratta di un Single Channel Packet Forwarder molto semplice ma efficace, in grado di dialogare con i dispositivi finali e The Things Network. Questo software viene rilasciato in forma libera a solo scopo di test, come spiegato nel README del repository.

SINGLE CHANNEL PACKET FORWARDER

Passiamo adesso all'utilizzo. Da terminale di Raspbian digitare il seguente comando per clonare la cartella del Single Channel Packet Forwarder in home/pi:

```
cd /home/pi
git clone https://github.com/tftelkamp/single_chan_pkt_fwd
```

All'interno della cartella si trova il codice sorgente che va compilato per produrre un eseguibile.

Basta entrare nella cartella e compilare con make:

```
cd single_chan_pkt_fwd
make
```

Dopo alcuni secondi verrà compilato il file eseguibile `single_chan_pkt_fwd` che potrà essere lanciato da terminale con:

```
./single_chan_pkt_fwd
```

Se vedete qualcosa di simile alla **Fig. 8**, significa che il vostro gateway a singolo canale funziona. Quello che si vede stampato sulle prime righe del terminale mostra che è stato rilevato un chip Semtech SX1276 e il gateway ID. Il canale in ascolto è fissato su 868.100000 MHz con SF7:

SX1276 detected, starting.
Gateway ID: b8:27:eb:ff:ff:8a:19:e7
Listening at SF7 on 868.100000 Mhz

CONCLUSIONI

Per il momento ci fermiamo qui. Con la speranza di essere stati sufficientemente chiari, vi diamo appuntamento alla prossima puntata, nella quale vi spiegheremo come programmare i vostri ricetrasmittitori LoRa come dispositivi finali LoRaWAN in grado di comunicare con il gateway e, soprattutto, con The Things Network.

