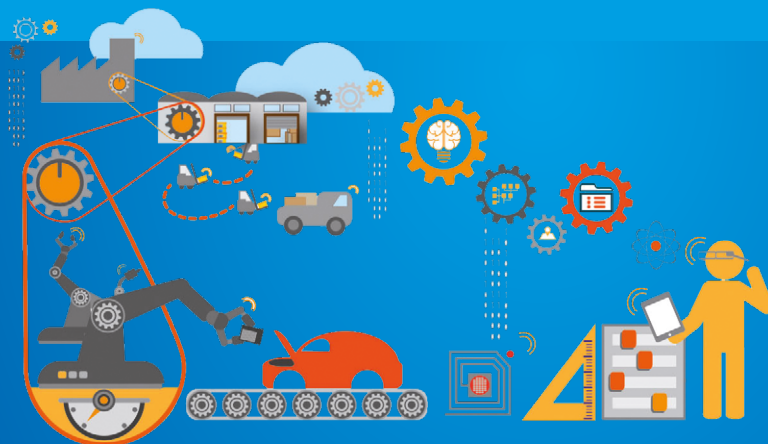


Il mondo dell'



dell'Ing. MIRCO SEGATELLO



**Sperimentiamo l'IoT
con l'impiego della rete a
basso consumo LoRa, valida
alternativa alle reti WiFi
e al GSM.**

Nella prima lezione di questo corso abbiamo parlato di dispositivi connessi nel senso più generico del termine, ma anche di quali possano essere le reali applicazioni dell'IOT (internet delle cose) con le tecnologie attualmente disponibili. Abbiamo anche dipinto un quadro dei possibili sviluppi di questa tecnologia e di quale sia la sua importanza nel mercato globale.

Nella seconda puntata abbiamo presentato una semplice applicazione basata sulla scheda Arduino MKR1000 che utilizzava una connessione WiFi per accedere alla rete Internet. Disporre di una rete WiFi nella propria abitazione è oggi

non abbastanza comune, così come in numerosi luoghi pubblici o aperti al pubblico, ma cosa succede se non si dispone di un accesso alla rete con il WiFi? Il primo pensiero va all'utilizzo di un modem GSM (GSM shield, se si utilizza Arduino) munito di SIM dati e relativo contratto con operatore che, pur essendo sempre più a buon mercato, non è certamente gratuito. Inoltre, utilizzare una tecnologia pensata per far dialogare smartphone, sarebbe ridondante per la trasmissione di alcuni byte di un semplice sensore di parametri ambientali e in generale per l'IoT. Per questo alcune aziende molto lungimiranti hanno pensato di sviluppare nuove tecnologie

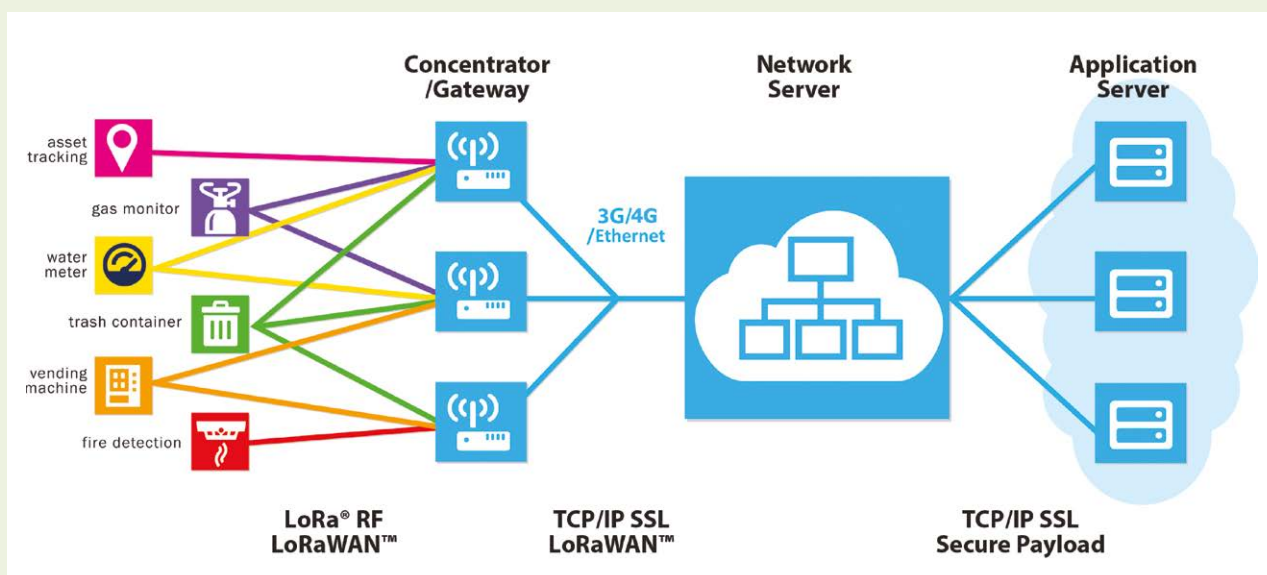


Fig. 1 - Schema della rete LoRaWAN.

specifiche, con l'unico fine di ottimizzare l'accesso alla rete, da parte di un specifico tipo di oggetti che non necessitano di uno streaming ad alto bit-rate né di una connessione continua alla rete. Queste tecnologie sono più d'una ma le più rilevanti di esse risultano certamente LoRa e SigFox; la tecnologia di cui andiamo a occuparci in questa terza puntata del corso è la prima, ossia LoRa (contrazione di Long Range).

LoRa: POCHI DATI, MOLTA COPERTURA

Questa è proposta dalla LoRa Alliance (<https://loro-alliance.org>) ovvero un'associazione senza scopo di lucro, impegnata a consentire l'implementazione su vasta scala di IoT in reti a bassa potenza (LPWAN) attraverso lo sviluppo e la promozione dello standard aperto LoRaWAN (Long Range Wide Area Network). La rete LoRaWAN (schematizzata nella Fig. 1) è in continua espansione ed attualmente conta oltre 100 operatori con una copertura che al momento comprende il Nord America, parte del Sud America, l'Europa e l'Australia; non c'è ovviamente copertura in mare aperto.

Questa tecnologia permette di usare sensori wireless LoRaWAN IoT (Internet of Things) in grado di comunicare a distanze di 100 km in campi aperti, 15 km in ambienti semi-rurali e fino a 2 km in ambienti densamente popolati con la capacità

di gestire sino a 1 milione di nodi. LoRaWAN è parte di una categoria di tecnologie chiamate LPWAN (Low Power Wide Area Network), sviluppata per poter permettere a sensori alimentati a batteria di inviare e ricevere messaggi, utilizzando pochissima energia in modo da poter funzionare per anni con una singola batteria stilo, questo a patto di sacrificare la velocità di trasmissione (non superiore a 50 kbps), e la quantità di dati inviabili.

LoRaWAN utilizza due tipi di chiavi simmetriche per la sicurezza della comunicazioni, che sono uniche per ciascun device; questo accorgimento rende impossibile l'intercettazione e la manomissione delle comunicazioni.

Tipiche applicazioni si possono ricercare in dispositivi IoT e IoE (Internet of Everything), ovvero piccoli sensori alimentati a batteria che inviano solo pochi byte poche volte al giorno, come ad esempio contatori di gas o acqua, lampioni per illuminazione pubblica, monitoraggio ambientale, tracciamento di veicoli ecc. Lo schema generale di una rete LoRa vede la presenza di una serie di dispositivi IOT che accedono tramite



Fig. 2
The Things Gateway
per LoRaWAN.

dei Gateway alla rete internet potendo così trasferire i dati al server centrale (Network Server) il quale a suo volta può dialogare con gli Application Server.

Il network server più adatto per i nostri scopi si chiama The Things Network (TTN) e nasce da una startup olandese che ha raccolto quasi 300 mila euro in una campagna di crowdfunding per lanciare una rete di Internet of Things aperta, libera, posseduta e gestita dagli utenti. The Things Network (che vedete nella Fig. 1) utilizza il protocollo LoRa-WAN (in futuro è previsto anche l'utilizzo del Bluetooth 4.2 per il corto raggio) e mette a disposizione tutta l'infrastruttura server necessaria al corretto funzionamento della rete libera, insieme ad una serie di servizi utilizzabili dai propri membri per interfacciarsi con i propri dispositivi. Per realizzare tutto ciò i codici, i firmware, i progetti e le conoscenze per la produzione dei dispositivi fisici necessari alla rete sono open source.

L'iniziativa è partita ad agosto 2015 con una prima sperimentazione ad Amsterdam in un progetto per la cura e la gestione delle imbarcazioni nei canali, e si è poi consolidata in una serie di altre esperienze. Il progetto è cresciuto inoltre a livello internazionale e ad oggi annovera 300 città nel mondo, fra cui varie città italiane.

L'accesso alla rete LoRaWAN avviene tramite appositi The Things Gateway (Fig. 2), spesso installati da privati nella loro abitazione, e connessi alla propria connessione WiFi o Ethernet. La portata è di circa 5-10km e permette di

espandere la comunità globale di sviluppatori IoT, così, ad esempio, l'area di Milano potrebbe essere coperta interamente con meno di 20 gateway (Fig. 3). Per sapere se nella vostra zona sono presenti Gateway potete recarvi nel sito <https://www.thethingsnetwork.org/> e consultare la relativa mappa oppure scaricate l'apposita app di nome LoRa dal Play Store. Nulla vieta, anzi è auspicabile, installare un Gateway nella propria abitazione, magari autocostruito utilizzando una scheda a basso costo come Raspberry Pi.

UN HARDWARE PER SPERIMENTARE

Volendo sperimentare anche noi questa nuova tecnologia, abbiamo approfittato di una nuovissima scheda Arduino denominata MKR WAN 1300 (distribuita da Futura Elettronica - www.futurashop.it - con il codice MKRWAN1300) che implementa all'interno tutte le funzionalità di una scheda MKR (MCU ATMEL SAMD21) con in più un modulo radio per la connessione alla rete LoRa, per la precisione un Murata CMWX1ZZABZ.

Riassumiamo brevemente le caratteristiche salienti della scheda MKR WAN 1300:

- microcontrollore ATMEL SAMD21 core a 32bit;
- alimentazione da USB a 5V (anche per programmazione);
- alimentazione tramite pacco batteria (2x AA o AAA) con connettore a vite;
- il microcontrollore e quindi tutti i pin funzionano a 3V3 e non sono tolleranti nei confronti dei 5V;

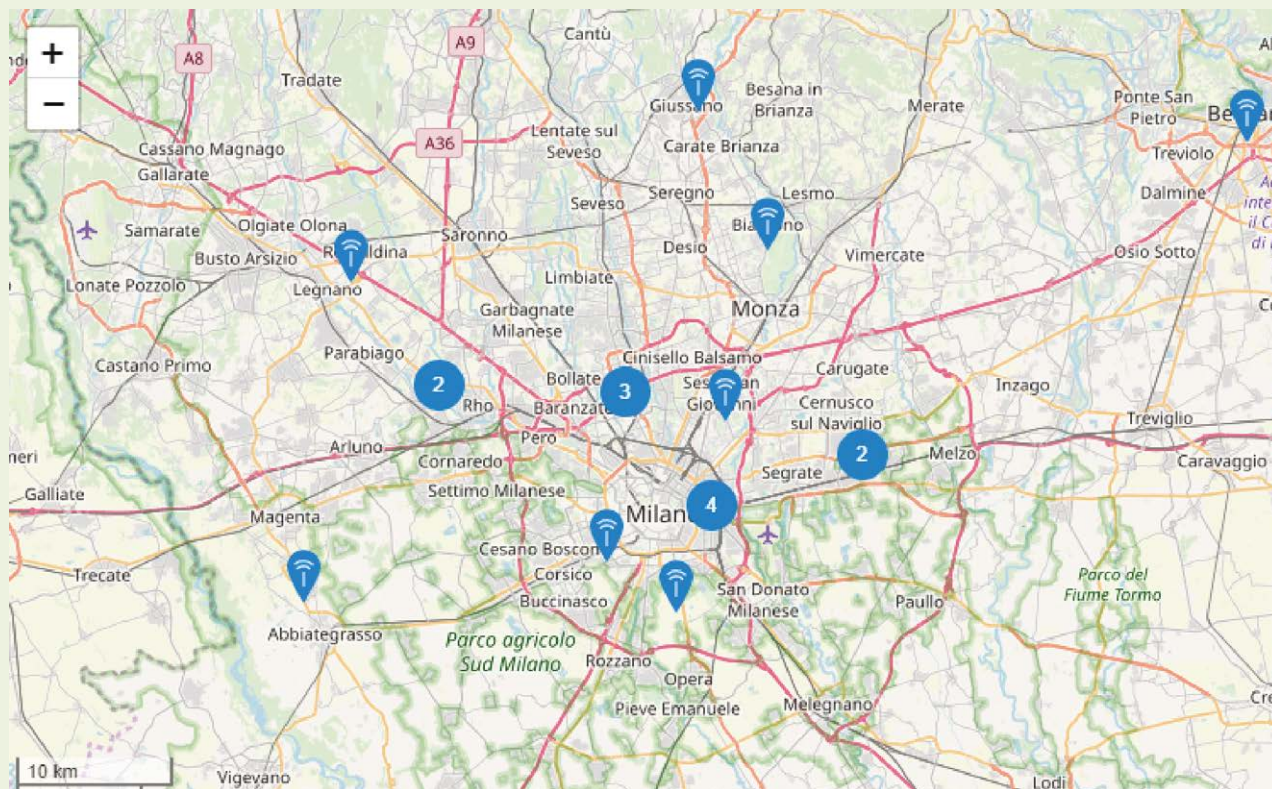


Fig. 3 - Mappa dei Gateway LoRaWAN nella zona di Milano.

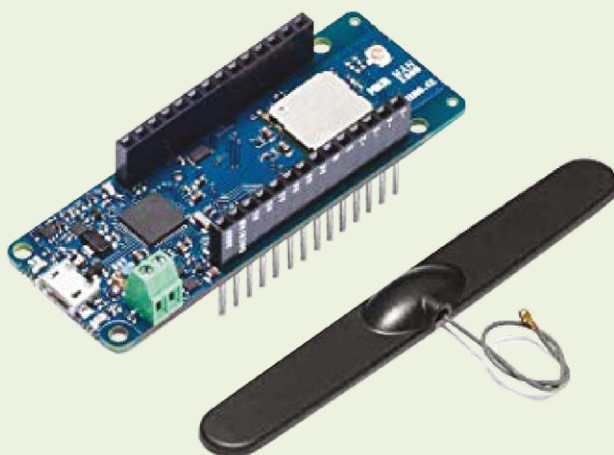


Fig. 4 - Configurazione Hardware per la nostra applicazione basata su LoRa.

- LED integrato collegato al pin D6 (non D13 come su Arduino UNO);
- LED di alimentazione collegato ai 5V acceso se viene fornita l'alimentazione tramite il connettore micro USB o VIN, mentre rimane spento quando la scheda è alimentata a batteria;
- il pin Vin può essere utilizzato per alimentare la scheda con una tensione di 5 VDC (in questo caso la fonte di alimentazione USB viene disconnessa);

- il pin 5V è un'uscita a 5 V quando la scheda viene alimentata attraverso il connettore micro USB o dal pin VIN;
- il pin VCC è un'uscita a 3,3 V. Se l'alimentazione viene fornita tramite il connettore micro USB o il pin VIN ed è uguale alla serie delle due batterie (quando vengono utilizzate).

Per il corretto funzionamento deve essere utilizzata un'antenna esterna connessa al connettore micro UFL e adatta a ricevere la gamma di frequenza LoRa nel paese di utilizzo: nel caso dell'Italia le frequenze assegnate fanno da 863 a 870 MHz; nel nostro caso abbiamo utilizzato l'antenna contraddistinta dal codice 7300-ANTENNADIPOLO (Fig. 4) fornita da Futura Elettronica (www.futurashop.it).

La nostra configurazione hardware per la prima prova consiste quindi una scheda MKR WAN 1300 connessa alla porta USB del PC. All'ingresso A0 della scheda MKR WAN 1300 collegheremo un potenziometro a simulare un segnale analogico di un sensore connesso alla scheda. Per la buona riuscita della connessione è necessario essere sufficientemente vicini ad un Gateway, in zone abitate già 2 km possono rappresentare un limite, l'antenna deve essere posizionata all'esterno meglio se in posizione sopraelevata. A livello software la prima cosa da fare è installare il supporto alle SAM Boards (Fig. 5) e successivamente scaricare la libreria per l'utilizzo delle schede MKR WAN (Fig. 6). Facoltativamente potete aggiornare il firmware del modem LoRa caricando il file `MKRWWANFWUpdate_standalone.ino`;

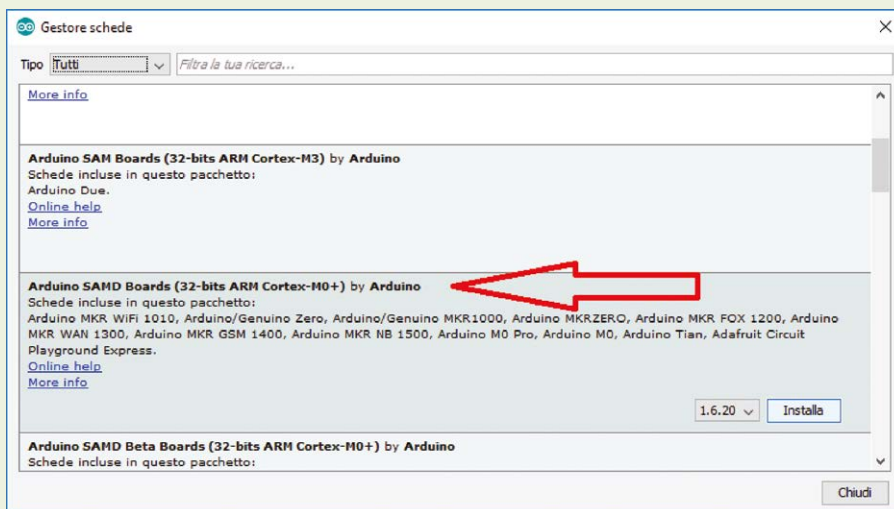
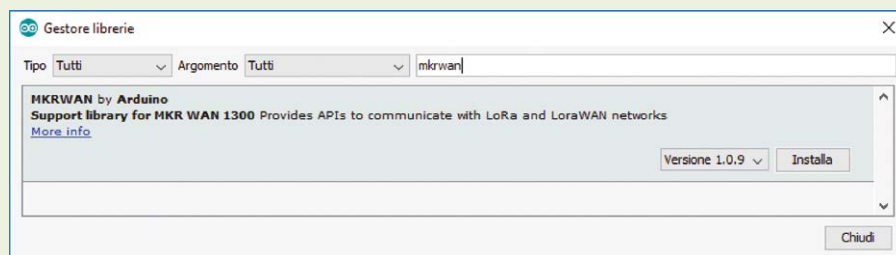


Fig. 5
Installazione supporto alle SAM boards.

Fig. 6
Installazione libreria MKR WAN.



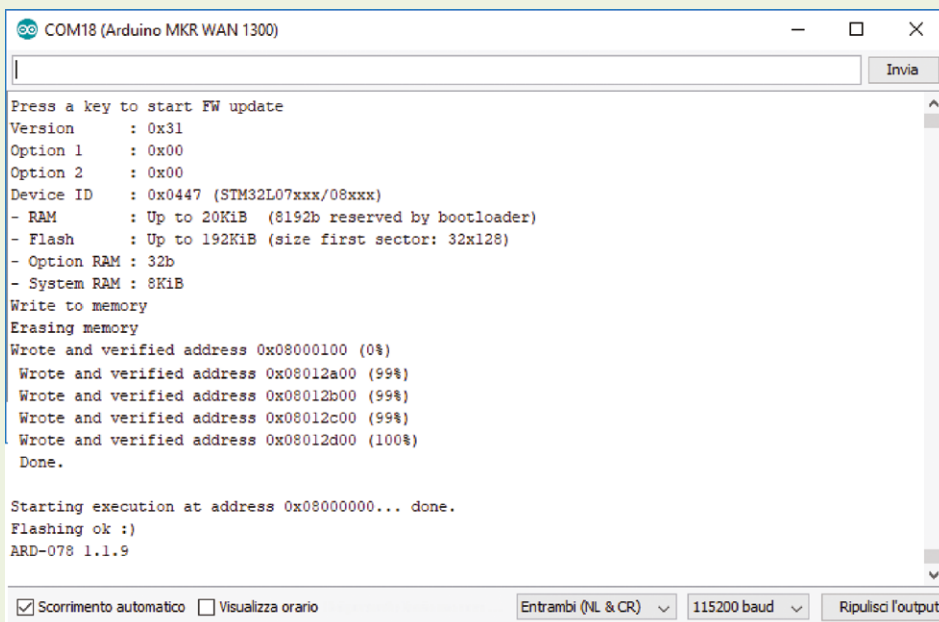


Fig. 7
Aggiornamento
firmware modem
LoRa.

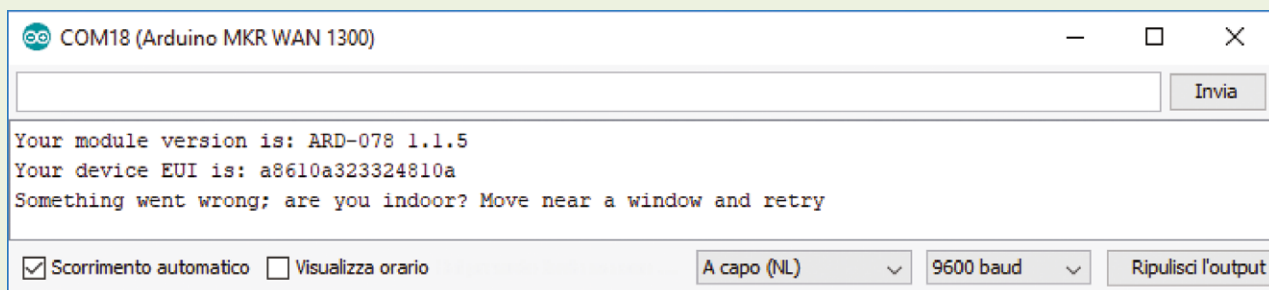


Fig. 8
Serial Monitor
con l'esempio
LoraSendAndReceive.

nel nostro caso il firmware è stato aggiornato dalla versione 1.1.5 alla versione 1.1.9 risolvendo alcuni bug (Fig. 7).

Aprirete e caricate l'esempio *LoraSendAndReceive.ino* (fornito con la libreria MKRWAN) e leggerete l'output tramite Serial Mo-

nitor; la finestra del monitor contiene la versione del modulo LoRa e il device EUI univoco per ciascun dispositivo, come visibile nella Fig. 8.

Non avendo ancora specificato le credenziali di accesso, il

ADD APPLICATION

Application ID
The unique identifier of your application on the network

Description
A human readable description of your new app

Application EUI
An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.

Handler registration
Select the handler you want to register this application to

Fig. 9
Aggiunta di un
nuovo dispositivo.

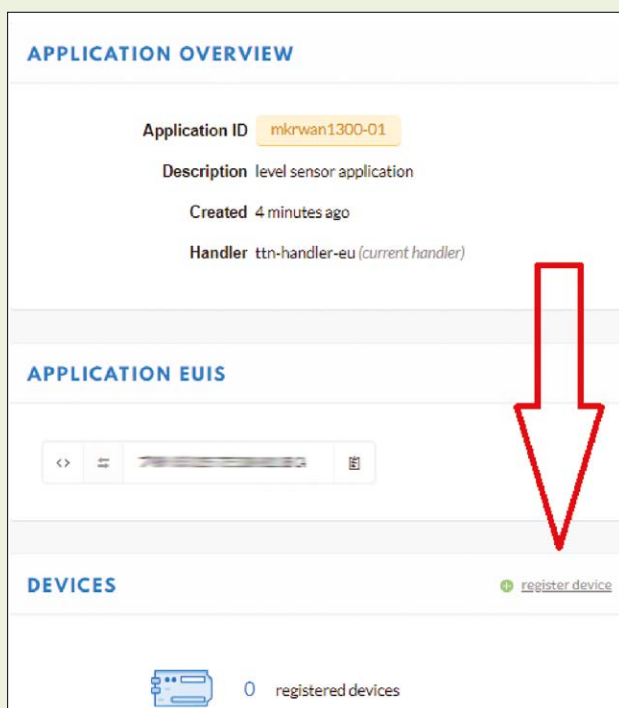


Fig. 10 - Configurazione di un nuovo dispositivo.

modulo non potrà connettersi alla rete e comparirà un messaggio di errore; analogamente avverrà se il modulo, pur essendo correttamente configurato, non troverà disponibile alcun accesso alla rete. In Europa, LoRaWAN opera nelle frequenze 863-870 MHz, ovvero la banda ISM (Industrial, Scientific & Medical) assegnata all'Europa come banda di libero utilizzo, assicuratevi che la riga **modem.begin(EU868)** contenga effettivamente la sigla EU868.

Il protocollo LoRaWAN necessita di valori numerici identificatori dei dispositivi, delle applicazioni e dei gateway, per cui per avviare una comunicazione è necessario disporre dei seguenti valori:

- Device EUI - Identificatore del dispositivo, unico per ciascun dispositivo, assegnato dal produttore del modulo;
- Device Address - Indirizzo del dispositivo (non univoco);
- Application EUI - Identificatore dell'applicazione (univoco);
- Gateway EUI - Identificatore del gateway (univoco).

I device LoRaWAN possono registrarsi sulla rete in due modi differenti: il primo (quello che useremo nel nostro esempio applicativo descritto in queste pagine) si chiama OTAA, Over-the-Air-Activation e prevede che il device e il network si scambino dinamicamente il Device Address e una security Key (Application Key).

La seconda modalità per effettuare la registrazione si chiama ABP (Activation By Personalization) e prevede che le chiavi delle sessioni vengano inserite manualmente dall'utente; tuttavia questo potrebbe causare problemi di sicurezza. A questo punto occorre registrarsi sulla rete creando un account sul sito <https://www.thethingsnetwork.org>. Dalla console principale avete la possibilità di registrare un gateway oppure, quello che interessa a noi, un dispositivo. Inserite il nome dell'applicazione (application ID) a piacimento e una semplice descrizione, quindi cliccate su **Aggiungi applicazione** (Fig. 9). Nella nuova schermata inserite un Device ID a piacimento e il Device EUI letto su Serial monitor di Arduino, quindi cliccate su **Register devices** (Fig. 10); si aprirà la schermata di riepilogo del vostro dispositivo, della quale ci interessano i campi Application EUI e Application KEY generati dal sistema (Fig. 11); per comodità, in basso troverete i due codici pronti per essere copiati (Fig. 12).

Fig. 11
Registrazione nuovo dispositivo.

REGISTER DEVICE

Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.

mkrwan1300-01

Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.

A8 61 0A 32 33 24 81 0A

App Key
The App Key will be used to secure the communication between you device and the network.

this field will be generated

App EUI

Fig. 12
Riepilogo della
configurazione del
nuovo dispositivo.

DEVICE OVERVIEW

Application ID

mkrwan1300-01

Device ID

mkrwan1300-01

Activation Method

OTAA

Device EUI

<> A8 61 0A 32 33 24 81 0A

Application EUI

<>

App Key

<>

Status

never seen

Frames up

0

[reset frame counters](#)

Frames down

0

COM5 (Arduino MKR WAN 1300)

Invia

Your module version is: ARD-078 1.1.5
Your device EUI is: a8610a323324810a

Enter a message to send to network
(make sure that end-of-line 'NL' is enabled)

Sending: ciao - 63 69 61 6F
Message sent correctly!
No downlink message received at this time.

Enter a message to send to network
(make sure that end-of-line 'NL' is enabled)

☒ Scorrimento automatico
☐ Visualizza orario

A capo (NL)

115200 baud

Ripulisci l'output

Fig. 13
Avvio sketch e
connessione alla rete
LoRa.

Ritornate sull'IDE di arduino e nel tab `arduino_secrets` inserite i due codici appena copiati, quindi caricate lo sketch su Arduino MKR ed attendete che il dispositivo si connetta, aprendo Serial Monitor potrete vedere lo stato della connes-

data rate	SF	BW	bitRate
0	SF12	125 Khz	250 bps
1	SF11	125 Khz	440 bps
2	SF10	125 Khz	980 bps
3	SF9	125 Khz	1760 bps
4	SF8	125 Khz	3125 bps
5	SF7	125 Khz	5470 bps

Tabella 1 - Valori del parametro `dr` (data rate).

sione. Per verificare che tutto funzioni correttamente inserite nell'apposito campo una piccola stringa (nel nostro caso "ciao") e cliccate sul pulsante **Invia**. Il debug su Serial Monitor visualizzerà lo stato dell'invio e la desiderata stringa "Message sent correctly!" che ci assicura che il messaggio è stato correttamente spedito e ricevuto (**Fig. 13**).

Dal portale di `thethingsnetwork` (TTN) aprite la tab `Application Data`, dove visualizzerete delle righe contenenti il traffico dati con la vostra scheda, cliccandoci sopra potrete visualizzare la stringa appena ricevuta codificata come codice ASCII in esadecimale. Scorrendo verso il basso potrete visualizzare anche i metatada contenenti tutti i dettagli della connessione come il gateway al quale si è connessi e l'intensità del segnale. Lo sketch di esempio fa un unico tentativo di connessione alla rete (dopodiché è necessario riavviare la scheda); dalle

APPLICATION DATA

uplink
downlink
activation
ack
error

time

counter

port

15:13:50

0

2

retry confirmed

payload: 63 69 61 6F

Uplink

Payload

63 69 61 6F

Fig. 14 - Stato della connessione.

nostre prove abbiamo notato che l'intensità del segnale (rssi) deve essere superiore a -120dB, pena la disconnessione dalla rete. Nella tab Downlink avete la possibilità di inviare messaggi verso la scheda MKR, ricordate che i messaggi sono codificati in esadecimale, quindi la scritta TEST diventa 54 45 53 54, dal codice ASCII di ogni singolo carattere. I byte inseriti risultano programmati per l'invio che avverrà solamente quando sarà ricevuto un qualche dato dalla scheda, se programmate più invii solo l'ultimo sarà effettivamente inviato (Fig.16 e Fig.17).

Listato 1

```
int sensorValue;
byte payload[2];
payload[0]=lowByte(sensorValue);
payload[1]=highByte(sensorValue);
modem.beginPacket();
modem.write(payload,2);
```

Metadata

```
{
  "time": "2019-03-05T13:53:03.326342273Z",
  "frequency": 868.5,
  "modulation": "LORA",
  "data_rate": "SF7BW125",
  "coding_rate": "4/5",
  "gateways": [
    {
      "gtw_id": "XXXXXXXXXX",
      "timestamp": 1075220163,
      "time": "2019-03-05T13:53:03Z",
      "channel": 2,
      "rssi": -115,
      "snr": 1.75,
      "rf_chain": 1
    }
  ]
}
```

Fig. 15 - Metadata della comunicazione.

TTN ha una politica di accesso che limita il tempo di trasmissione a 30 secondi in uplink e 10 messaggi di downlink al giorno, poiché i gateway devono rispettare questi limiti, la risposta dal gateway non è garantita, inoltre i tempi tra l'invio e la ricezione di conferma ricezione possono essere molto lunghi. La classica applicazione è quindi l'invio ad intervalli regolari (di minuti o ore) di piccolissimi pacchetti dati, meglio se codificati in byte.

Un classico esempio potrebbe essere l'invio del valore analogico acquisito da Arduino ed inviato al server TTN ad intervalli di alcuni minuti.

L'impostazione dello sketch rimane sostanzialmente lo stesso, con l'inizializzazione del modem e della connessione che però possiamo migliorare aggiungendo la possibilità di reiterare il tentativo di connessione qualora dovesse fallire. Un altro particolare su cui vogliamo soffermarci riguarda il baud rate di trasmissione, che in questo tipo di comunicazione è molto importante perché legato in modo inverso alla massima distanza raggiungibile e comunque non superiore a 50 kbps. L'istruzione si chiama *modem.dataRate(dr)* nella quale il

DOWNLINK

Scheduling

replace

first

last

FPort

1

Confirmed

Payload

bytes
fields

54 45 53 54

4 bytes

Send

Fig. 16 - Invio di una stringa verso MKRWAN.

time	counter	port			
17:00:29	1		payload: 81 82	Invio dei bytes programmati	
17:00:27	3	2	confirmed payload: 67	Ricezione	
16:58:08	1		payload: 81 82	Bytes programmati per l'invio	
16:57:59	1		payload: 80 81		
16:57:18	0				

Fig. 17 - Stato della connessione durante un invio.

parametro **dr** può assumere i valori riportati nella **Tabella 1**. Valori più alti di **dr** assicurano velocità di trasmissioni maggiori e consumi minori ma si riduce la portata coperta. Per massimizzare sia la durata della batteria che la portata e la capacità complessiva della rete, l'infrastruttura di rete LoRa può gestire la velocità dei dati e la potenza di uscita utilizzati per la comunicazione mediante uno schema ADR (Adaptive Data Rate). Significa che migliore è la copertura del segnale più basso sarà il Spreading Factor (vedi **Tabella 1**) ma è comunque sconsigliato usare la modalità ADR per oggetti in movimento continuo. Potremmo pertanto utilizzare l'istruzione *modem.setADR(true)* per attivare la modalità ADR. Nell'esempio *LoRa_Level_Analog.ino* ad intervalli regolari di un minuto acquisiamo il valore analogico dal pin A0, codifichiamo il valore in un solo byte (0-255 in decimale) e lo inviamo al server TTN. Nella console TTN, per evitare di dover leggere valori espressi in byte (di non immediata lettura), è possibile specificare come saranno codificati i dati ricevuti; nel tab payload formats è sufficiente aggiungere la riga *decoded* (**Fig. 19**).

```
level=bytes[0];
```

Se invece interessa inviare un dato di tipo integer codificato con 2 bytes è necessario modificare le righe di codice come riportato nel **Listato 1**.

APPLICATION DATA						
Filters						
uplink downlink activation ack error						
time	counter	port				
16:09:07	16	2	dev id: mkrwan1300-01 payload: 5C level: 92			
16:08:07	15	2	dev id: mkrwan1300-01 payload: 75 level: 117			
16:07:07	14	2	dev id: mkrwan1300-01 payload: 75 level: 117			
16:06:07	13	2	dev id: mkrwan1300-01 payload: 75 level: 117			
16:05:06	12	2	dev id: mkrwan1300-01 payload: 8D level: 141			

PAYLOAD FORMATS

Payload Format

The payload format sent by your devices

Custom

decoder

converter

validator

encoder

```
1 function Decoder(bytes, port) {
2   // Decode an uplink message from a buffer
3   // (array) of bytes to an object of fields.
4   var decoded = {};
5   decoded.level=bytes[0];
6
7
8   return decoded;
9 }
```

Fig. 19 - Decodifica dei dati ricevuti in formato decimale.

Mentre per la decodifica la riga sarà:

```
decoded.level = (bytes[1] << 8) + bytes[0];
```

Ricordiamo anche che è anche disponibile l'app TTN Mobile per il controllo dei propri dispositivi direttamente dal proprio smartphone.

CONCLUSIONI

Per il momento ci fermiamo qui; l'applicazione proposta è un esempio per spiegare l'utilizzo della connessione radio LoRa. Volendo realizzare un'applicazione più avanzata in cui è anche necessario prevedere un'alimentazione esterna a batteria, vi rimandiamo ad un articolo dedicato che pubblicheremo a breve. Appuntamento al prossimo fascicolo di Elettronica In, dove concluderemo questo corso!

Fig. 18 - Visualizzazione del tab application data di TTN con l'esempio *LoRa_Level_Analog*.