

Capire e usare

3

LoRa

di PIER CALDERAN

Impariamo a utilizzare le librerie LoRaWAN per poter connettere i nostri dispositivi finali IoT e i nostri gateway al cloud di The Things Network.

Giunti a questo punto, dopo aver spiegato che cos'è LoRa e quali sono e come funzionano le reti LoRaWAN, nonché dopo aver visto come realizzare i primi dispositivi da connettere ad essa, riteniamo che con le informazioni fornite e gli schemi degli hardware proposti nelle scorse puntate, dovrete avere costruito almeno un dispositivo finale (end-device) ed un gateway. A questo punto possiamo ritenere che siate pronti a lavorare con essi e a realizzare insieme alcune applicazioni di esercitazione. In questo corso, per la visualizzazione dei dati ricevuti dal gateway, utilizzeremo esclusivamente il portale **The Things Network** (www.thethingsnetwork.org), che abbrevieremo spesso con la sigla **TTN**, come viene convenzionalmente chiamato dalla community LoRa.

REGISTRAZIONE SU TTN

Dalla home page di The Things Network è possibile registrare i nostri gateway e i dispositivi finali in pochissimo tempo con

dei semplici passaggi. Nella home page del sito sono visibili anche il numero aggiornato di gateway attivi nel mondo, compreso il nostro (riferitevi alla **Fig. 1**) su una mappa geografica e quindi geolocalizzati.

Per lavorare con The Things Network, la prima cosa da fare è creare un account gratuito; allo scopo, cliccando sul pulsante *Get Started* si accede alla sezione *Create a free account*. Seguire le istruzioni per la creazione di un account e quindi accedere nuovamente a TTN effettuando il login. Una volta effettuato il login si potrà accedere alla propria console, come illustrato in **Fig. 2**.

APPLICATIONS E GATEWAY

La console consente di registrare **applications** e **gateway**. Si tratta di dispositivi virtuali che andranno a riflettere il nostro hardware. Ogni application riceverà i dati dai dispositivi finali e ogni gateway virtuale riceverà i dati dai gateway fisici. La prima volta che si accede alla console è necessario regi-

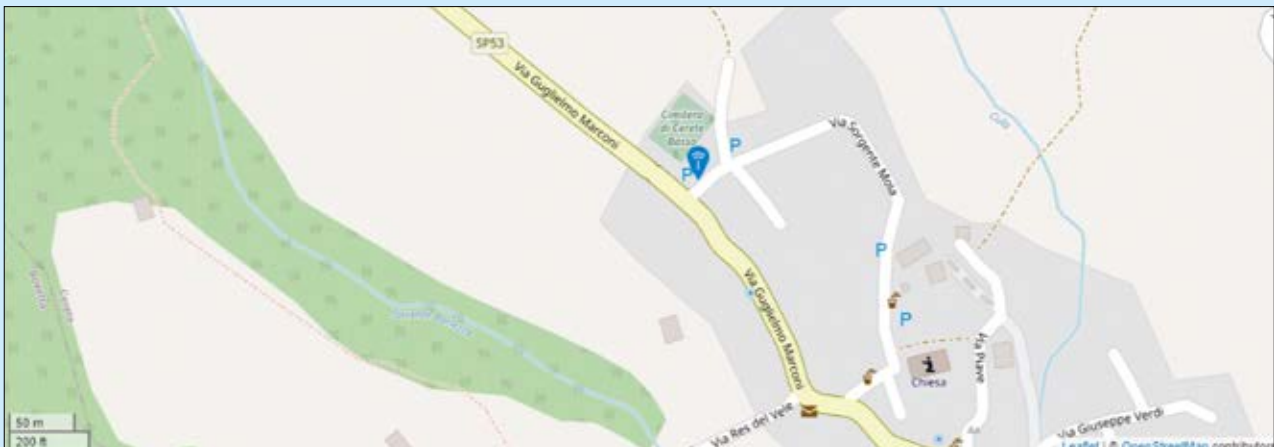


Fig. 1 - La localizzazione del nostro gateway.

strare almeno un gateway. I gateway registrati in precedenza verranno elencati nella pagina (**Fig. 3**) dove ogni gateway riporta la scritta *connected* quando viene connesso, altrimenti riporterà la scritta *not connected*.
Il gateway che è stato registrato per la dimostrazione fatta in questo corso si chiama **FUTURA RASPBERRY HAT** e lo trovate nella predetta figura.



Fig. 2 - La console personale su TTN.

Gli altri gateway che abbiamo registrato sono:

- **Dragino LG02**: un gateway a 2 canali dell'azienda cinese Dragino;
- **iC880A + Raspberry Pi NEW**: un gateway a 8 canali composto da un iC880A e un Raspberry Pi;
- **Dragino Corso LoRa/LoRaWAN**: un gateway a 1 canale Dragino con HAT per Raspberry Pi, che abbiamo usato nel webinar LoRa;
- **ttn packet forwarder**: ancora un iC880A, ma funzionante con "ttn packet forwarder".

Si fa notare che è possibile registrare un numero indefinito di gateway e connetterli tutti contemporaneamente.

REGISTRARE UN GATEWAY

Per registrare un nuovo gateway o per aggiungerne uno a quelli già esistenti, basta fare clic su *register gateway*, in alto a destra.

Nella pagina di registrazione, nella sezione *Register Gateway*

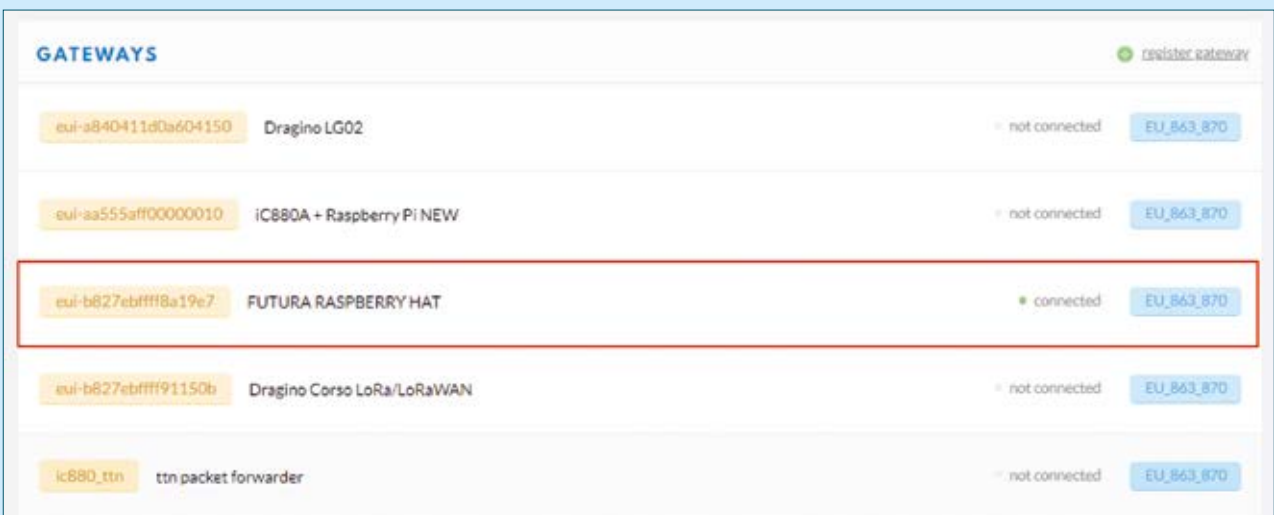


Fig. 3 - L'elenco dei nostri gateway registrati.

(Fig. 4) è necessario compilare il modulo con le seguenti informazioni:

Gateway ID: il testo riporta *"A unique, human-readable identifier for your gateway. It can be anything so be creative!"* ovvero "Un identificatore univoco leggibile dall'uomo per il vostro gateway. Può essere qualsiasi cosa, quindi siate creativi!". Attenzione! Se si sceglie questa opzione, è possibile usare solo il software **ttn_packet_forwarder** fornito da TTN sotto forma di eseguibile senza sorgenti e quindi senza la possibilità di modificarlo in alcun modo. Spiegheremo più avanti nel

corso l'uso di questo software. Le istruzioni per l'uso con un iC880A e Raspberry Pi, sono comunque disponibili nel repository https://github.com/TheThingsNetwork/packet_forwarder/blob/master/docs/INSTALL_INSTRUCTIONS/IMST_RPI.md. Noi, invece, useremo il packet forwarder Semtech open source e disponibile presso il repository https://github.com/TheThingsNetwork/packet_forwarder/tree/legacy. Per rendere attiva questa opzione è necessario vistare la casella "I'm using the legacy packet forwarder". In questo modo, invece del *Gateway ID* bisognerà inserire il

Gateways

Register

REGISTER GATEWAY

Gateway ID

A unique, human-readable identifier for your gateway. It can be anything so be creative!

☐ I'm using the legacy packet forwarder

Select this if you are using the legacy [Semtech packet forwarder](#)

Description

A human-readable description of the gateway

Frequency Plan

The [frequency plan](#) this gateway will use

no selection

Router

The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.

Location

The exact location of your gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.

+

-

Gessate

SP125

SP126

SP127

SP128

SP129

SP130

SP131

SP132

SP133

SP134

SP135

SP136

SP137

SP138

SP139

SP140

SP141

SP142

SP143

SP144

SP145

SP146

SP147

SP148

SP149

SP150

SP151

SP152

SP153

SP154

SP155

SP156

SP157

SP158

SP159

SP160

SP161

SP162

SP163

SP164

SP165

SP166

SP167

SP168

SP169

SP170

SP171

SP172

SP173

SP174

SP175

SP176

SP177

SP178

SP179

SP180

SP181

SP182

SP183

SP184

SP185

SP186

SP187

SP188

SP189

SP190

SP191

SP192

SP193

SP194

SP195

SP196

SP197

SP198

SP199

SP200

SP201

SP202

SP203

SP204

SP205

SP206

SP207

SP208

SP209

SP210

SP211

SP212

SP213

SP214

SP215

SP216

SP217

SP218

SP219

SP220

SP221

SP222

SP223

SP224

SP225

SP226

SP227

SP228

SP229

SP230

SP231

SP232

SP233

SP234

SP235

SP236

SP237

SP238

SP239

SP240

SP241

SP242

SP243

SP244

SP245

SP246

SP247

SP248

SP249

SP250

SP251

SP252

SP253

SP254

SP255

SP256

SP257

SP258

SP259

SP260

SP261

SP262

SP263

SP264

SP265

SP266

SP267

SP268

SP269

SP270

SP271

SP272

SP273

SP274

SP275

SP276

SP277

SP278

SP279

SP280

SP281

SP282

SP283

SP284

SP285

SP286

SP287

SP288

SP289

SP290

SP291

SP292

SP293

SP294

SP295

SP296

SP297

SP298

SP299

SP300

SP301

SP302

SP303

SP304

SP305

SP306

SP307

SP308

SP309

SP310

SP311

SP312

SP313

SP314

SP315

SP316

SP317

SP318

SP319

SP320

SP321

SP322

SP323

SP324

SP325

SP326

SP327

SP328

SP329

SP330

SP331

SP332

SP333

SP334

SP335

SP336

SP337

SP338

SP339

SP340

SP341

SP342

SP343

SP344

SP345

SP346

SP347

SP348

SP349

SP350

SP351

SP352

SP353

SP354

SP355

SP356

SP357

SP358

SP359

SP360

SP361

SP362

SP363

SP364

SP365

SP366

SP367

SP368

SP369

SP370

SP371

SP372

SP373

SP374

SP375

SP376

SP377

SP378

SP379

SP380

SP381

SP382

SP383

SP384

SP385

SP386

SP387

SP388

SP389

SP390

SP391

SP392

SP393

SP394

SP395

SP396

SP397

SP398

SP399

SP400

SP401

SP402

SP403

SP404

SP405

SP406

SP407

SP408

SP409

SP410

SP411

SP412

SP413

SP414

SP415

SP416

SP417

SP418

SP419

SP420

SP421

SP422

SP423

SP424

SP425

SP426

SP427

SP428

SP429

SP430

SP431

SP432

SP433

SP434

SP435

SP436

SP437

SP438

SP439

SP440

SP441

SP442

SP443

SP444

SP445

SP446

SP447

SP448

SP449

SP450

SP451

SP452

SP453

SP454

SP455

SP456

SP457

SP458

SP459

SP460

SP461

SP462

SP463

SP464

SP465

SP466

SP467

SP468

SP469

SP470

SP471

SP472

SP473

SP474

SP475

SP476

SP477

SP478

SP479

SP480

SP481

SP482

SP483

SP484

SP485

SP486

SP487

SP488

SP489

SP490

SP491

SP492

SP493

SP494

SP495

SP496

SP497

SP498

SP499

SP500

SP501

SP502

SP503

SP504

SP505

SP506

SP507

SP508

SP509

SP510

SP511

SP512

SP513

SP514

SP515

SP516

SP517

SP518

SP519

SP520

SP521

SP522

SP523

SP524

SP525

SP526

SP527

SP528

SP529

SP530

SP531

SP532

SP533

SP534

SP535

SP536

SP537

SP538

SP539

SP540

SP541

 Fig. 4 - La pagina Register Gateway.

Fig. 5 - La casella Gateway EUI per registrare il gateway con il packet forwarder di Semtech.

Gateway EUI (Extended Unique Identifier), come illustrato in Fig. 5.

Questo identificativo corrisponde al DevEUI, un identificatore univoco esteso a 64 bit (EUI-64) assegnato dal produttore o dal proprietario del dispositivo finale o del gateway. Quando si attiva il gateway con il packet forwarder di Semtech o il packet forwarder di Sandeep Mistry, che abbiamo usato nel numero scorso per verificarne il funzionamento, viene creato un gateway ID all'inizio del log sul terminale (si veda il riquadro rosso della Fig. 6) del tipo:

Gateway ID: b8:27:eb:ff:ff:8a:19:e7

Questo corrisponde al Mac Address del chip SX1276 montato sulla breakout board della HopeRF, con due byte FF aggiunti prima degli ultimi tre byte.

Per copiare questo gateway ID senza sbagliare consigliamo di evidenziarlo con il mouse e di copiarlo con SHIFT+CTRL+C oppure impartendo il comando Copy dal menu Edit del terminale. Una volta copiato il gateway ID, basta incollarlo nella casella *Gateway EUI* della pagina *Register Gateway*.

Dopo aver compiuto tale operazione dovete inserire quanto segue negli altri campi del modulo di registrazione.

- **Description:** è possibile inserire una descrizione per il gateway; per esempio, noi abbiamo usato la descrizione FUTURA RASPBERRY HAT.
- **Frequency Plan:** qui è importante scegliere un piano di frequenza dall'elenco a discesa; nel nostro caso deve essere Europe 868 MHz.
- **Router:** il router di The Things Network da scegliere dall'elenco è quello europeo, ovvero ttn-router-eu.
- **Location:** anche se non è strettamente necessario, possiamo impostare la location del nostro gateway. Facendo clic sulla mappa di Google Maps, si può contrassegnare la dislocazione del gateway.
- **Antenna Placement:** se si si vuole informare che il gateway si trova all'interno di un'abitazione scegliere *Indoor* altrimenti, scegliere *Outdoor*.

Al termine, fare clic sul pulsante *Register Gateway* per completare la registrazione e venire indirizzati alla pagina *Gateway Overview* (Fig. 7).

PAGINA GATEWAY OVERVIEW

Nella pagina Gateway Overview è possibile visualizzare lo stato del gateway, ovvero *connected* quando il packet forwarder

Fig. 6 - Il gateway ID nella finestra del terminale.

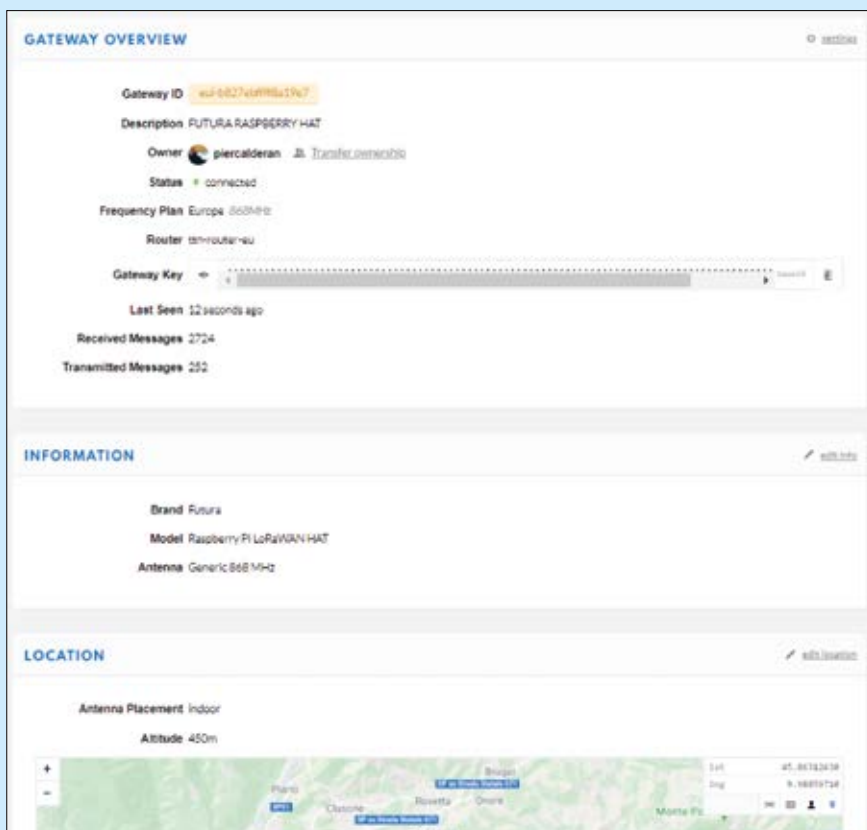


Fig. 7 - La pagina Gateway Overview.

der è in esecuzione oppure *not connected*.

A volte bisogna attendere qualche minuto prima che l'ID del gateway si propaghi in rete.

Le altre informazioni relative al gateway sono le seguenti.

- **Frequency Plan:** Europe868MHz ovvero il piano di frequenza scelto.
- **Router:** ttn-router-eu ovvero il router ttn.

- **Gateway Key:** è la chiave assegnata al gateway da TTN. Si tratta di una chiave univoca codificata in base64 che deve essere usata soltanto con il ttn packet forwarder.
- **Last Seen:** 24 seconds ago, ovvero il tempo trascorso dall'ultima connessione.
- **Transmitted Messages:** 2728 ovvero il numero di messaggi trasmessi.

Fig. 8 - Pagina che riporta il traffico di dati.

uplink		downlink		join		0 bytes		X		pause		clear	
time	frequency	mod.	CR	data rate	airtime (ms)	cnt	dev addr	payload size	bytes	bytes	bytes	bytes	bytes
▲ 14:32:56	868.1	loro	4/5	SF 7 BW 125	61.7	297	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:31:20	868.1	loro	4/5	SF 7 BW 125	61.7	294	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:29:43	868.1	loro	4/5	SF 7 BW 125	61.7	291	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:28:07	868.1	loro	4/5	SF 7 BW 125	61.7	288	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:26:31	868.1	loro	4/5	SF 7 BW 125	61.7	285	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:21:42	868.1	loro	4/5	SF 7 BW 125	61.7	276	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:20:05	868.1	loro	4/5	SF 7 BW 125	61.7	273	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:18:29	868.1	loro	4/5	SF 7 BW 125	61.7	270	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:13:40	868.1	loro	4/5	SF 7 BW 125	61.7	261	dev addr: 26 01 13 A4	payload size: 25 bytes					
▲ 14:12:04	868.1	loro	4/5	SF 7 BW 125	61.7	258	dev addr: 26 01 13 A4	payload size: 25 bytes					

↓ Listato 1

```
{
  "gw_id": "eui-b827ebffff8a19e7",
  "payload": "QcGvASaAVwABVDudyZdeVE+akvL5ptDB0w=",
  "f_cnt": 87,
  "lora": {
    "spreading_factor": 7,
    "bandwidth": 125,
    "air_time": 61696000
  },
  "coding_rate": "4/5",
  "timestamp": "2019-07-12T10:47:21.710Z",
  "rssi": -24,
  "snr": 9,
  "dev_addr": "26011528",
  "frequency": 868100000
}
```

- **Received Messages:** 252 ovvero il numero di messaggi ricevuti.
- **INFORMATION:** qui si possono inserire opzionalmente alcune informazioni del gateway, come marca, modello e tipo di antenna.
- **LOCATION:** qui si può visualizzare e/o modificare l'ubicazione del gateway.

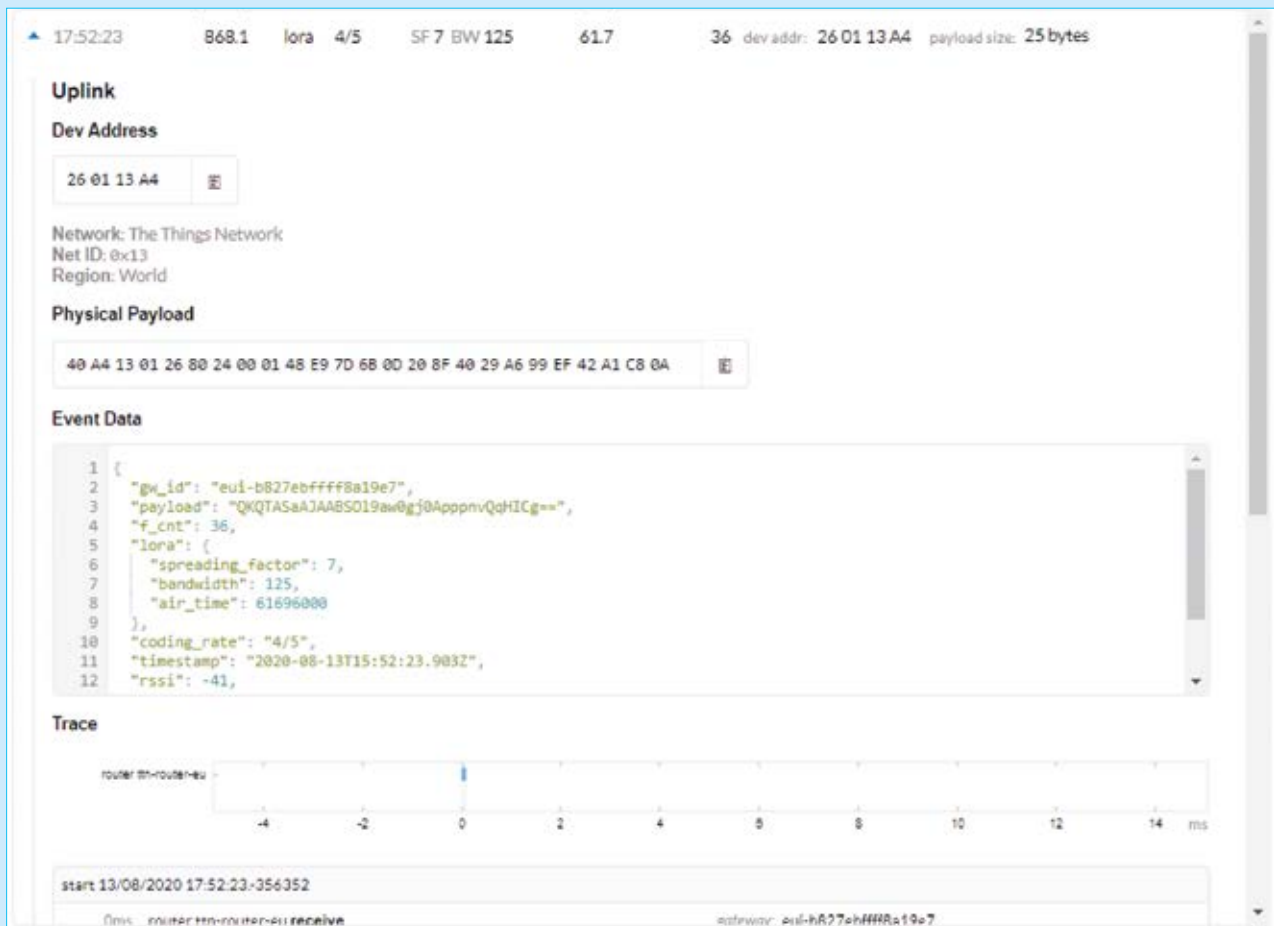
- **PRIVACY:** qui si può visualizzare e/o modificare lo stato della privacy ovvero si può rendere pubblico lo stato per consentire ad altri di ottenere informazioni sui tempi di attività del gateway, rendere pubblica la posizione e consentire ad altri di conoscere la posizione esatta del gateway e rendere pubblico il proprietario per consentire ad altri di conoscere il proprietario del gateway.

PAGINA TRAFFIC

Il pulsante Traffic apre la pagina omonima in cui è possibile monitorare il traffico di dati ricevuti o inviati dei vari dispositivi finali (**Fig. 8**).

In questa pagina si possono filtrare le informazioni dei pacchetti in *downlink*, *uplink*, *join* oppure tutti insieme. Nella forma compressa delle righe si possono visualizzare i dati suddivisi nelle seguenti colonne:

- **time**, ossia l'orario di ricezione;
- **frequency**, ovvero la frequenza di trasmissione;
- **mod.**, che è il tipo di modulazione (ad esempio LoRa);
- **CR**, ossia il Coding Rate;
- **data rate**, che è il Data Rate;
- **airtime (ms)** ovvero il tempo di trasmissione (ms);



↑ Fig. 9 - Sezione Uplink del pacchetto ricevuto.

GATEWAY SETTINGS

- General
- Owner
- Location
- Privacy
- Information
- Collaborators

GENERAL

Description
A human-readable description of the gateway
FUTURA RASPBERRY HAT

Frequency Plan
The [frequency plan](#) this gateway will use
Europe 868MHz

Router
The id of the router your gateway will connect to.
ttn-router-eu

☒ **Automatically update gateway**
If enabled the gateway will periodically check if updates are available and perform them.
Enabling auto updates may cause your gateway to have unexpected downtime when updating.

☐ **Beta Updates**
Turn this on to receive firmware from the beta release channel.

Fig. 10 - La tipica pagina Settings.

- **cnt**, ossia il contatore (*frame counter*);
- **dev. addr.** che è l'indirizzo del dispositivo finale;
- **payload size**, cioè la dimensione del payload.

SEZIONE UPLINK

Con un clic su una riga di un messaggio qualsiasi, possiamo espandere il dettaglio di ogni pacchetto ricevuto in *Uplink* (Fig. 9). È importante notare che i dettagli del messaggio si riferiscono ai dati "fisici" di trasmissione del messaggio e non al suo "contenuto".

In altre parole, questi dati riportano non "cosa contiene" un messaggio ma "come è stato ricevuto o inviato".

Sotto la voce *Uplink* vengono elencate le informazioni seguenti:

- **Dev Address:** è l'indirizzo fisico di 4 byte assegnato al nostro dispositivo finale (vedi più avanti);
- **Network, Net ID e Region** sono informazioni di The Things Network;
- **Physical Payload:** è il payload fisico ricevuto, ovvero il messaggio codificato in base64. Può essere decodificato grazie un'applicazione che si chiama *LoRaWAN packet decoder* disponibile a questo indirizzo: <https://lorawan-packet-decoder-Ota6puiniaut.runkit.sh>.
- **Event Data:** in questa finestra vengono mostrati in modo esplicito tutti i dati del pacchetto ricevuto in formato json (**Listato 1**);
- **Trace:** in questo grafico viene mostrato il tracciamento del pacchetto ricevuto e tutte le informazioni tecniche relative al router TTN, di cui non ci occuperemo in questa sede.

PAGINA SETTINGS

Il pulsante *Settings* apre la pagina omonima (Fig. 10) in cui si possono modificare le impostazioni del gateway.

- Nella sezione *General*, si possono modificare i campi *Description*, *Frequency Plan*, *Router*, *Automatically Update Router* e *Beta Updates*. Queste opzioni, se abilitate, controlleranno periodicamente gli aggiornamenti e li installerà.
- Nella sezione *Owner* si può trasferire la proprietà a un altro utente. Il proprietario è il nome visualizzato nelle pagine della community;
- Nella sezione *Location* si può modificare la location del gateway;
- Nella sezione *Privacy* è possibile modificare lo stato della privacy;
- Nella sezione *Information* si possono aggiornare le informazioni del gateway come *Brand*, *Model*, *Antenna*;
- Nella sezione *Collaborators* si possono aggiungere collaboratori esterni, oltre al proprietario.

In fondo alla pagina *Settings* è presente anche l'opzione *Delete gateway...* che consente di eliminare in modo definitivo il gateway. Per confermare l'eliminazione si aprirà una nuova finestra in cui inserire l'ID del gateway.

Notate che fino a questo momento il gateway virtuale che abbiamo registrato su TTN sta ricevendo pacchetti di dati dal gateway fisico (il nostro FUTURA RASPBERRY HAT, per intenderci). Questi pacchetti includono il payload e altre informazioni relative al trasmettitore, ma non cosa viene effettivamente trasmesso da un dispositivo finale.

Prima di creare un'applicazione in grado di fare questo, vediamo i passi necessari per programmare il nostro dispositivo finale in modo che possa inviare i dati al gateway fisico e quindi all'applicazione di TTN.

MODALITÀ LORAWAN

Finora abbiamo utilizzato i nostri dispositivi finali come semplici ricetrasmittitori LoRa. Per renderli compatibili con un gateway e con TTN, è necessario che vengano programmati in modalità LoRaWAN. Per fare questo abbiamo a disposizione una libreria appositamente studiata per Arduino che si chiama **arduino-lmic**, disponibile come open source nel seguente repository <https://github.com/matthijskooijman/arduino-lmic>. Questo contiene la libreria IBM LMIC (LoraMAC-in-C), leggermente modificata per essere eseguita nell'ambiente Arduino, per consentire l'utilizzo dei ricetrasmittitori Semtech SX1272, SX1276 e moduli compatibili, come gli RFM96 della HopeRF.

Questa libreria espone principalmente le funzioni definite dalla libreria originale LMIC. Per scoprire come utilizzare la libreria stessa, vedere gli esempi o vedere il file PDF nella sottodirectory doc. Questa libreria richiede Arduino IDE versione 1.6.6 o successiva, poiché richiede che la modalità C99 sia abilitata per impostazione predefinita. Dalle informazioni del repository si legge che è stata sviluppata dall'IBM Zurich Research Laboratory (IBM Research GmbH), 8803 Rüschlikon, Svizzera. Per ulteriori informazioni suggeriamo di contattare lo stesso via e-mail, all'indirizzo di posta elettronica lrsc@zurich.ibm.com. Per installare questa libreria si può utilizzare il gestore delle librerie Arduino attraverso il comando di menu **Sketch>Includi libreria>Gestisci librerie...** oppure scaricare il file zip dal GitHub utilizzando il pulsante "Scarica ZIP" e installare la libreria usando il comando di menu **Sketch > Includi libreria > Aggiungi libreria da file zip...** oppure

pure clonare direttamente questo repository nella cartella *libraries*. Per maggiori informazioni su come installare le librerie, vedere il sito ufficiale di Arduino <https://www.arduino.cc/it/guide/libraries>.

Fate attenzione a un dettaglio: dalla versione 1.8.x dell'IDE Arduino, la libreria LMIC non è più compatibile ed è necessario apportare la modifica descritta da questa procedura:

1. aprire la cartella `...\Arduino\libraries\IBM_LMIC_framework\src\lmic`;
2. aprire con un editor tipo notepad++, Programmer's notepad o simili il file `oslmic.h`;
3. cercare la riga `inline type table_get ## postfix(const type *table, size_t index)`;
4. aggiungere "static" prima di "inline":
`static inline type table_get ## postfix(const type *table, size_t index)`;
5. salvare e riavviare l'IDE di Arduino.

CARATTERISTICHE DELLA LIBRERIA LMIC

La libreria fornisce un'implementazione LoRaWAN in Classe A e Classe B abbastanza completa, supportando le bande EU-868 e US-915. Solo un numero limitato di funzionalità è stato testato utilizzando l'hardware Arduino, quindi si prega di fare attenzione quando si utilizzano delle funzionalità non testate. Ciò che sicuramente funziona:

- Invio di uplink di pacchetti, tenendo conto del duty-cycle.
- Crittografia e controllo dell'integrità dei messaggi (MIC).
- Ricezione di pacchetti di downlink nella finestra RX2.
- Frequenze personalizzate e impostazioni del data-rate.
- Attivazione OTAA e ABP.

ESEMPI DELLA LIBRERIA LMIC

La libreria LMIC fornisce attualmente tre esempi, che sono quelli descritti qui di seguito.

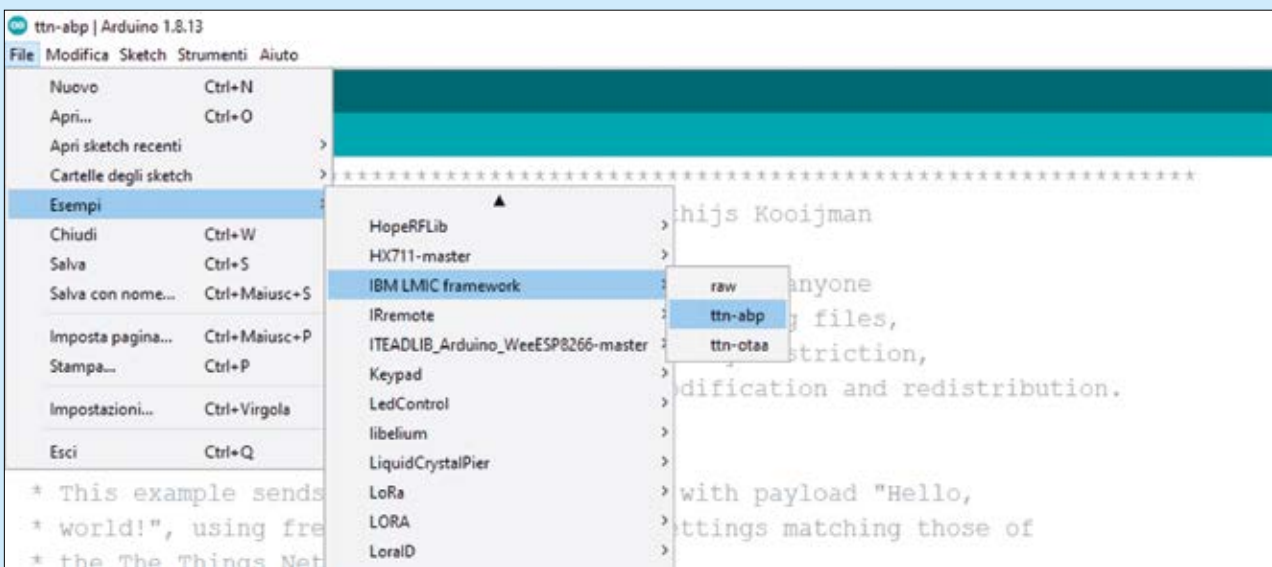


Fig. 11 - Apertura del file di esempio ttn-abp.



Fig. 12 - Pagina Application.

- **ttn-abp**: effettua una trasmissione del messaggio "Hello, world!" a TTN utilizzando l'attivazione ABP (*Activation By Personalization*), per cui bisogna preconfigurare un indirizzo del dispositivo e le chiavi di sessione e network.
- **ttn-otaa**: effettua una trasmissione del messaggio "Hello, world!" utilizzando l'attivazione OTAA (*Over The Air Activation*) con indirizzo preconfigurato da TTN.
- **raw**: mostra come accedere alla radio a un livello piuttosto basso e consente di inviare pacchetti raw (non LoRaWAN) tra i nodi direttamente. Questo esempio è utile solo per verificare la connettività di base e quando non è disponibile un gateway. Questo esempio ignora anche i controlli del ciclo di lavoro.

L'ESEMPIO TTN-ABP

Per capire come far funzionare il tutto, partiamo con l'esempio più semplice: dal comando di menu **File>Esempi>IBM LMIC framework** andiamo ad aprire l'esempio **ttn-abp**, come illustrato in **Fig. 11**. Una volta caricato l'esempio, i commenti all'inizio spiegano quello che fa lo sketch. Questo esempio invia un pacchetto LoRaWAN con il payload "Hello, world!", utilizzando l'attivazione ABP per cui il **DevAddr** e le chiavi **NWKSKEY** (*Network Session Key*) e **APPSKEY** (*Application Session Key*) devono essere preconfigurati nell'applicazione che creeremo su TTN (come vedremo più avanti).

Per motivi di spazio riportiamo nel **Listato 2** solo la parte dello sketch che ci interessa; più avanti avremo modo di approfondire l'argomento.

Quello che si nota subito all'inizio dello sketch è la presenza delle variabili statiche **NWKSKEY**, **APPSKEY** e **DEVADDR** che dovremmo necessariamente impostare per ogni dispositivo finale che andrà a connettersi in modalità ABP, ovvero con l'attivazione personalizzata.

Nel blocco *Pin mapping* relativo alla mappatura dei pin del nostro dispositivo finale bisognerà modificare i pin in questo modo per farli corrispondere esattamente ai collega-

menti del PCB e del modulo HopeRF:

```
const lmic_pinmap lmic_pins = {
  .nss = 10,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = 9,
  .dio = {2, 8, LMIC_UNUSED_PIN},
};
```

CREARE UN'APPLICAZIONE

Prima di caricare lo sketch nel nostro dispositivo finale, bisogna ottenere da TTN le chiavi **NWKSKEY**, **APPSKEY** e l'indirizzo **DEVADDR**. Per fare questo è necessario creare un'applicazione su TTN contenente almeno un dispositivo. Vediamo dunque come si esegue tale creazione.

Pagina Applications

Dalla console personale, con un clic su **Applications**, possiamo vedere tutte le applicazioni registrate o aggiungerne una nuova (**Fig. 12**). Nella predetta figura vedete che sono presenti già alcune applicazioni create in precedenza ed una creata appositamente per questo corso, che abbiamo chiamato **futura_devices** (Dispositivi Futura).

Facendo clic su **add application** in alto a destra si aprirà la pagina **Add Application**, che sarà simile a quella illustrata nella **Fig. 13**. In questa pagina bisognerà compilare tutti i campi del modulo come segue.

- **Application ID**: inserire qui un identificativo univoco dell'applicazione da creare; nel nostro caso abbiamo messo il nome **futura_devices**. Notate che tutti i caratteri devono essere minuscoli e senza spazi.
- **Description**: in questo campo occorre scrivere una descrizione dell'applicazione, che nel nostro esempio è Dispositivi Futura.
- **Application EUI**: numero di identificazione EUI che verrà emesso dal gestore The Things Network.
- **Handler registration**: selezionare il gestore con cui operare, per esempio **ttn-handler-eu**.

Listato 2

```
#include <lmic.h> //libreria LMIC
#include <hal/hal.h> //libreria HAL
#include <SPI.h> //libreria SPI

static const PROGMEM u1_t NWKSKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F,
0x3C }; //chiave da modificare

static const u1_t PROGMEM APPSKEY[16] = { 0x2B, 0x7E, 0x15, 0x16, 0x28, 0xAE, 0xD2, 0xA6, 0xAB, 0xF7, 0x15, 0x88, 0x09, 0xCF, 0x4F,
0x3C }; //chiave da modificare

static const u4_t DEVADDR = 0x03FF0001; //indirizzo da modificare

static uint8_t mydata[] = "Hello, world!"; //payload con il messaggio Hello, world!
static osjob_t sendjob; //chiamata alla funziona sendjob

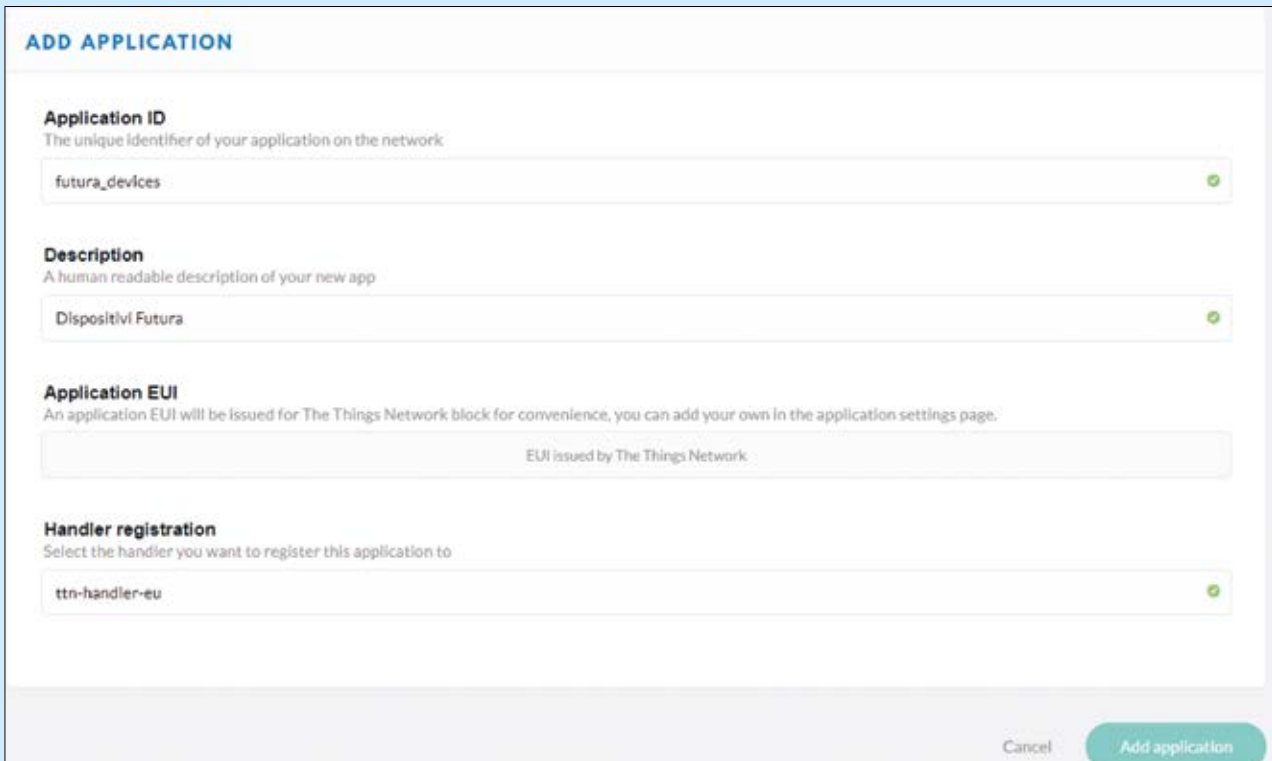
const unsigned TX_INTERVAL = 60; //intervallo di trasmissione fra un messaggio e il successivo

//Pin mapping (mappatura dei pin del dispositivo finale da modificare)
const lmic_pinmap lmic_pins = {
    .nss = 6,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 5,
    .dio = {2, 3, 4},
};
```

Al termine, fare clic sul pulsante **Add Application** per creare l'applicazione. Si verrà indirizzati alla pagina **Application Overview** ovvero all'anteprima dell'applicazione che riassume tutti i dati dell'applicazione (**Fig. 14**) e l'identificativo EUI. Questo EUI dovrà essere usato per collegare i dispositivi finali usando l'attivazione OTAA.

Pagina Application Overview

In questa pagina oltre alle informazioni immesse nella pagina Add Application, si possono individuare le sezioni **Devices**, **Collaborators** e **Access Key**. Per il momento concentriamoci sulla sezione **Devices** perché avremo modo di tornare a parlare della sezione Collaborators e Access Key.



 **Fig. 13 - Pagina Add Application.**

APPLICATION OVERVIEW

Application ID: **futura_devices** [documentation](#)

Description: Dispositivi Futura

Created: 3 hours ago

Handler: **tttn-handler-eu** (current handler)

APPLICATION EUIs

[manage euis](#)

70 83 05 7E D0 03 34 3F

DEVICES

[create device](#) [manage devices](#)

0 registered devices

COLLABORATORS

[manage collaborators](#)

piercaldern [collaborators](#) [delete](#) [devices](#) [settings](#)

ACCESS KEYS

[manage keys](#)

default key [delete](#) [messages](#)

..... [delete](#)

← Fig. 14 - Pagina Application Overview.

REGISTER DEVICE

[bulk import devices](#)

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

device_1

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

AA AA AA AA AA AA AA AA 8 Bytes

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

70 83 05 7E D0 03 34 3F

Cancel Register

↑ Fig. 15 - Pagina Register Device.

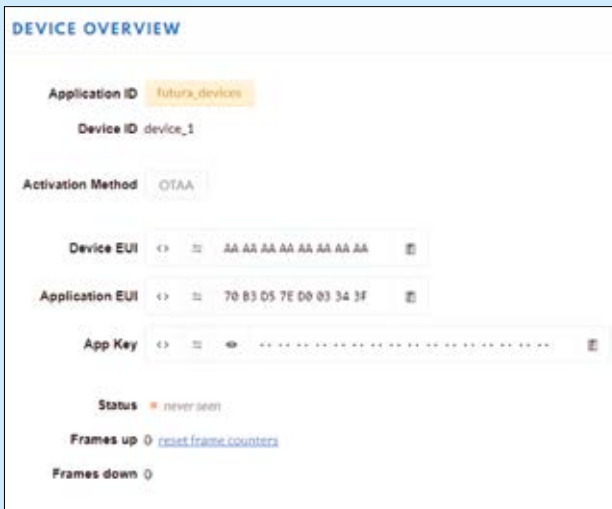


Fig. 16 - Pagina Device Overview.

Qui si vedrà l'indicazione *0 registered device*, la quale ci comunica che attualmente non esiste alcun dispositivo registrato. Per registrarne uno, bisogna fare clic sul collegamento *register device*, operazione che aprirà la pagina *Register Device*. In questa pagina si potranno registrare tutti i dispositivi finali (end-device) che apparterranno all'Application da noi appena creata.

Pagina Register Device

Passiamo adesso alla pagina **Register Device** (Fig. 16), nella quale bisogna compilare i relativi campi come segue.

- **Device ID:** questo è l'identificativo univoco per il dispositivo dell'applicazione. Nel nostro caso è stato scelto il nome **device_1**. L'ID del dispositivo può essere composto solo

da caratteri alfanumerici minuscoli senza spazi.

- **Device EUI:** questo è il numero EUI del dispositivo ovvero l'identificativo univoco per questo dispositivo sulla rete. L'EUI del dispositivo deve contenere esattamente 8 byte, per esempio, **AA AA AA AA AA AA AA AA** o qualsiasi altro numero esadecimale.
- **App Key:** questa chiave verrà utilizzata per proteggere la comunicazione tra il dispositivo e la rete; si tratta di una chiave autogenerata da TTN che viene utilizzata in modalità OTAA.
- **App EUI:** è l'identificativo EUI dell'applicazione generata in precedenza.

A questo punto basta fare clic sul pulsante *Register* per registrare il dispositivo; con tale operazione si verrà indirizzati alla pagina **Device Overview**, ovvero all'anteprima del dispositivo, in cui vengono riassunti tutti i dettagli del dispositivo finale (ne vedete l'esempio in Fig. 16).

Pagina Device Overview

Come si può notare, il metodo di attivazione predefinito (*Activation Method*) è OTAA. Siccome l'esempio **ttn-abp** che caricheremo nel nostro dispositivo finale si basa sul metodo di attivazione ABP, bisognerà cambiare questa impostazione. Allo scopo, dalla stessa pagina **Device Overview** bisogna fare clic sul pulsante **Settings**, allorché si aprirà una pagina di impostazioni del dispositivo simile a quella proposta nella Fig. 17. Nella sezione *Activation Method*, andate a selezionare **ABP** come metodo di attivazione, quindi fare clic sul pulsante **Save** per memorizzare l'impostazione.

Tornando alla pagina **Device Overview** si potranno copiare l'indirizzo *Device Address* assegnato al dispositivo finale e le due chiavi *Network Session Key* e *App Session Key*, che use-

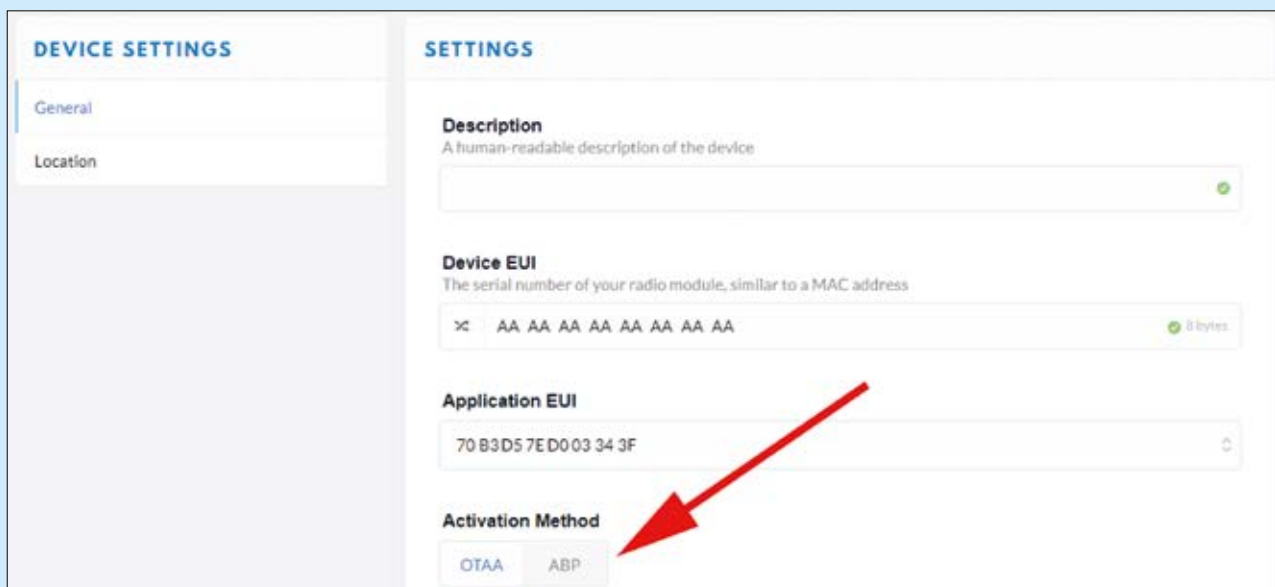


Fig. 17 - Impostazione del metodo di attivazione ABP.

DEVICE OVERVIEW

Application ID futura_devices

Device ID device_1

Activation Method ABP

Device EUI <> == AA AA AA AA AA AA AA AA

Application EUI <> == 70 B3 D5 7E D0 03 34 3F

Device Address <> == 26 01 16 48


Network Session Key <> == msb { 0xB3, 0xDE, 0x56, 0xEE, 0x2D, 0x40, 0xE7, 0x17, 0x06, 0x31, 0x2C, 0x16, 0x18, 0x0A, 0x0B, 0x0C, 0x0D }

App Session Key <> == msb { 0x3E, 0x25, 0x8D, 0x11, 0x77, 0x7C, 0x16, 0x95, 0x8F, 0x25, 0x5E, 0x94, 0x3E, 0x0A, 0x0B, 0x0C, 0x0D }

Status never seen

Frames up 0 [reset frame counters](#)

Frames down 0

 Fig. 18 - Pagina Device Overview con le indicazioni dell'indirizzo e delle chiavi da usare nel nostro sketch.

remo nel nostro sketch. Per rendere visibili le chiavi dovete cliccare sull'icona "occhio" e per visualizzarle in formato array cliccare sull'icona "<>", come illustrato dalle frecce di **Fig 19**. Per copiare queste informazioni basterà fare clic sull'icona "copia" e incollarle nel codice di esempio **ttn-abp** in corrispon-

denza delle variabili:

```
static const u4_t DEVADDR = 0x 26011648; //mettere il prefisso
0x davanti al Device Address
static const PROGMEM u1_t NWKKEY[16] =
```

Overview

Data

Settings

APPLICATION DATA

Filters

uplink

downlink

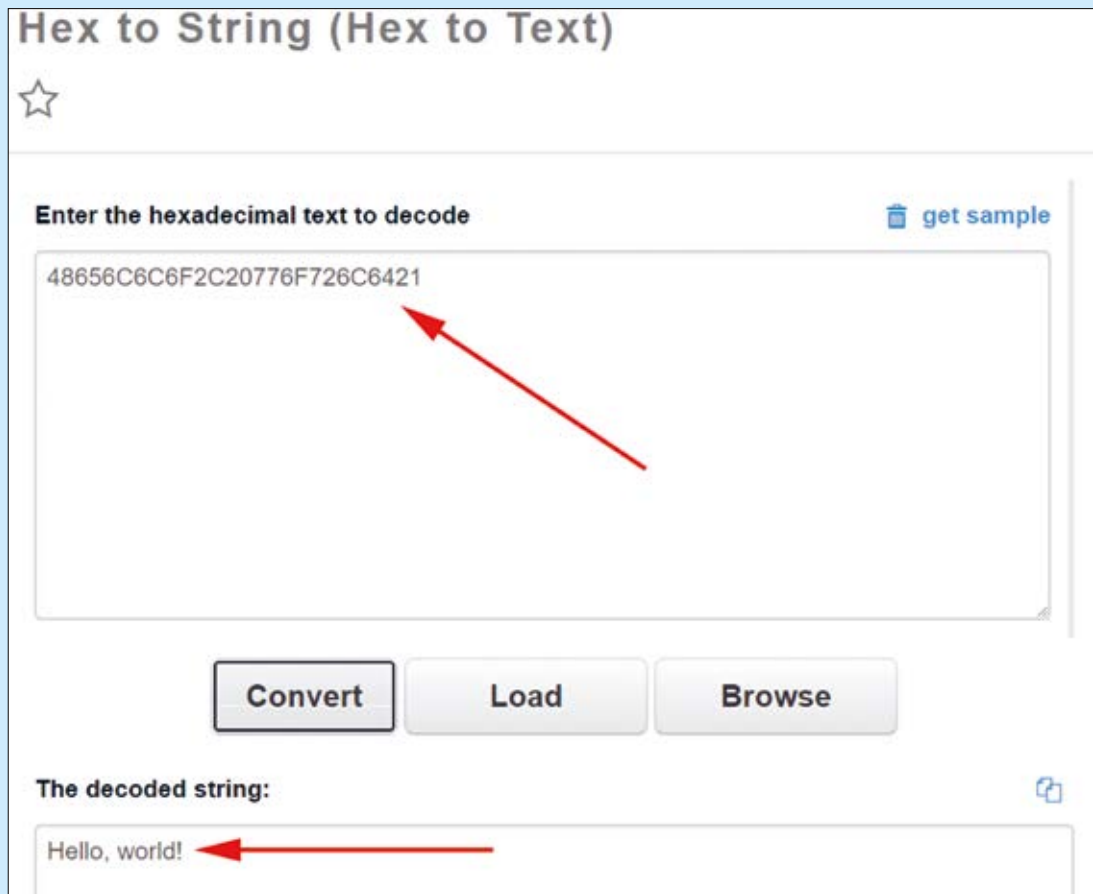
activation

ack

error

time	counter	port	payload
▲ 14:01:33	40	1	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▲ 13:53:16	32	1	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▲ 13:44:59	24	1	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▲ 13:36:42	16	1	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▲ 13:28:25	8	1	payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21
▲ 13:20:08	0	1	retry payload: 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

 Fig. 19 - Pagina Application Data con i messaggi ricevuti e il relativo payload.



→ **Fig. 20**
Verifica del
messaggio "Hello,
world!" ricevuto.

```
{ 0xB3, 0xDE, 0x56, 0xEE, 0x2D, 0x40, 0xE7, 0x17, 0x06, 0x31,  
0x2C, 0x16, 0x18, 0x0E, 0x13, 0x9E };
```

```
static const u1_t PROGMEM APPSKEY[16] =  
{0x3E, 0x25, 0x8D, 0x11, 0x77, 0x7C, 0x16, 0x95, 0xBF, 0x25,  
0x5E, 0x94, 0x3E, 0x0F, 0xBE, 0xBC };
```

In questo modo il nostro dispositivo finale viene configurato correttamente con l'indirizzo del dispositivo e le due chiavi di accesso. A questo punto si può procedere al caricamento dello sketch ttn-abp nel nostro dispositivo finale, seguendo la stessa procedura vista nella prima puntata del corso, ovvero impostando una scheda Arduino come programmatore ISP e collegando il chip ATmega328 8 MHz su una breadboard.

HELLO WORLD!

Se tutto è stato fatto correttamente, quello che dovremo vedere su TTN sarà il messaggio "Hello, world!", come previsto dall'esempio.

Per visualizzare i dati del device_1 all'interno dell'applicazione, basta andare su **Applications>futura_devices>Devices>device_1 > Data**.

Osservando la **Fig. 19**, noterete che sono stati ricevuti sei messaggi contenenti lo stesso payload, che è:

48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21

Il messaggio ricevuto è composto da numeri esadecimali corrispondenti ai caratteri ASCII del testo "Hello, world!". Usando un convertitore da *Hex* a *String* (per es. <https://codebeautify.org/hex-string-converter>) si può verificare che si tratta proprio del messaggio "Hello, world!" (**Fig. 20**).

CONCLUSIONI

Bene, con questo abbiamo terminato la nostra applicazione basata su The Things Network; in questa puntata del corso abbiamo creato il nostro account su tale piattaforma, registrato in esso i nostri device, impostato il gateway e sviluppato, testandola, la nostra applicazione, fino ad ottenere il classico Hello World! A questo punto non ci resta che darvi appuntamento alla prossima puntata, nella quale approfondiremo quanto spiegato sommariamente in queste pagine e spiegheremo come programmare più efficacemente i dispositivi finali, in modo da sfruttarli al meglio con le potenzialità e le funzionalità offerte da The Things Network. □

