

Workshop Trasformatorio 2018

Parte pratica



Preliminari

Connettersi alla cartella condivisa di Drive

Qui sono contenuti tutti gli sketch, i documenti e le librerie

Per la condivisione serve la mail di ogni partecipante

Configurazione della IDE

Versione IDE Arduino: 1.8.5 (consigliata. Non ho provato quelle successive!)

Occorre far riconoscere alla IDE le schede che useremo:

Adafruit Feather 32u4

Sparkfun ESP8266

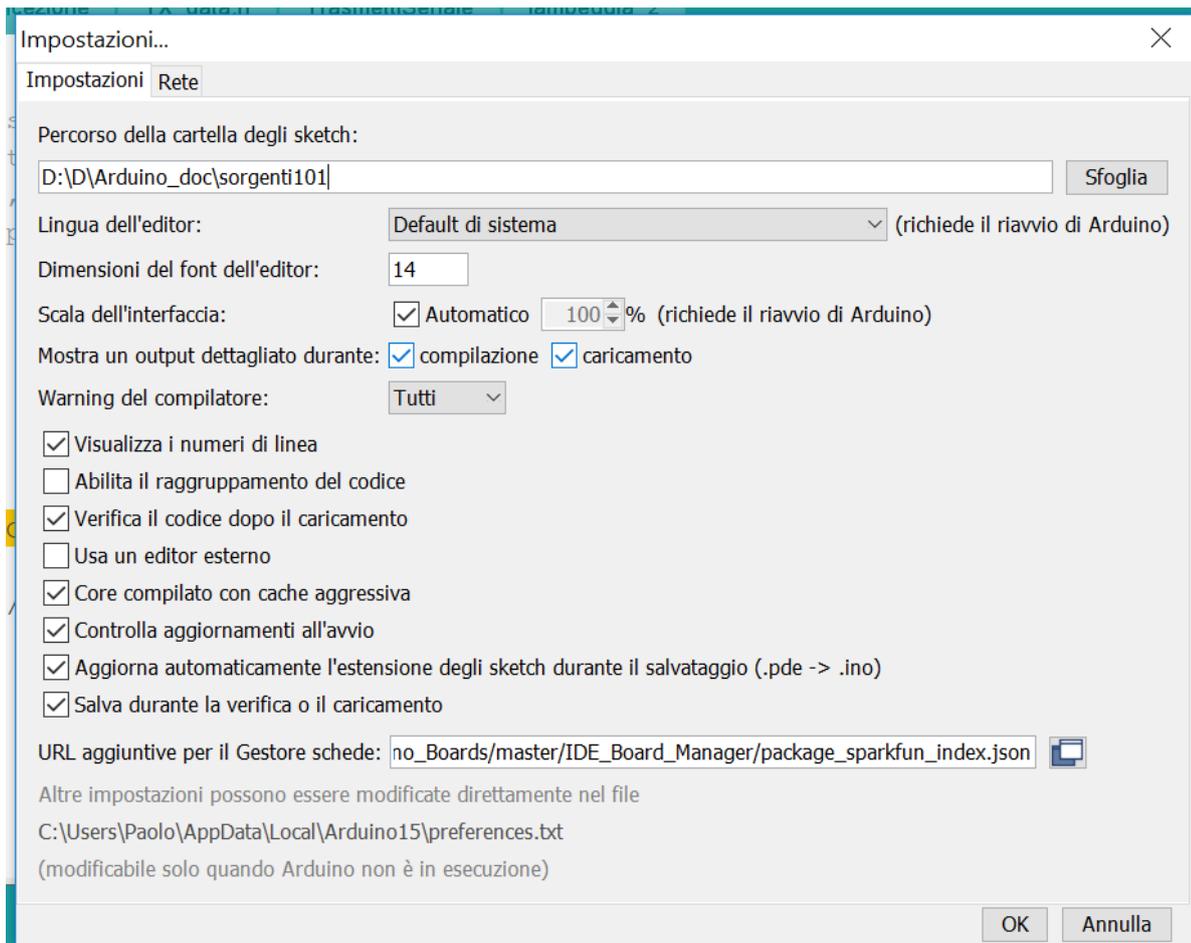
Link per i compilatori (Additional Board Manager)

Adafruit: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

Sparkfun (ESP Community):

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

Settaggi sul menu "Impostazioni" della IDE



Librerie:

Per Adafruit Feather TX

#include <SPI.h> (E' compresa in Arduino. Serve alla comunicazione tra modulo radio e microcontrollore. Per RX e TX)

#include <RH_RF95.h> (E' dentro la RadioHead. Serve a mandare comandi al modulo radio. Per RX e TX)

#include "HX711.h" (Serve ad amplificare e convertire in digitale il segnale della cella di carico della bilancia. Solo per TX)

#include <Adafruit_SSD1306.h> (Serve per pilotare il display OLED. TX e RX)

#include "Adafruit_SleepyDog.h" (Serve a risparmiare energia al TX)

Per Feather RX

#include <SPI.h> (E' compresa in Arduino. Serve alla comunicazione tra modulo radio e microcontrollore.)

#include <RH_RF95.h> (E' dentro la RadioHead. Serve a mandare comandi al modulo radio.)

#include <Adafruit_SSD1306.h> (Serve per pilotare il display OLED.)

#include <SoftwareSerial.h> (E' compresa in Arduino. Serve alla comunicazione seriale tra Scheda Feather e Sparkfun.)

Per Sparkfun ESP8266

#include <SoftwareSerial.h> > (E' compresa in Arduino. Serve alla comunicazione seriale tra Scheda Feather e Sparkfun.)

#include <ESP8266WiFi.h> (Serve a connettere la scheda Sparkfun al WiFi)

#include <Exosite.h> (Serve a inviare i dati al server exosite)

Copiare le librerie nella cartella "Libraries" all'interno della cartella degli sketch

Prove preliminari

Caricare sulle schede uno sketch tratto dagli esempi (blink)

Soluzioni di problemi di caricamento sulla scheda Adafruit

Se sulla IDE non compare la porta seriale, una volta collegato il cavo USB, provare a schiacciare il tasto reset. Altrimenti caricare lo stesso e, quando arrivano i messaggi della ricerca di una COM, dare un reset.

Prove di compilazione degli sketch

Esperimento 1

Molti trasmettitori e un ricevitore

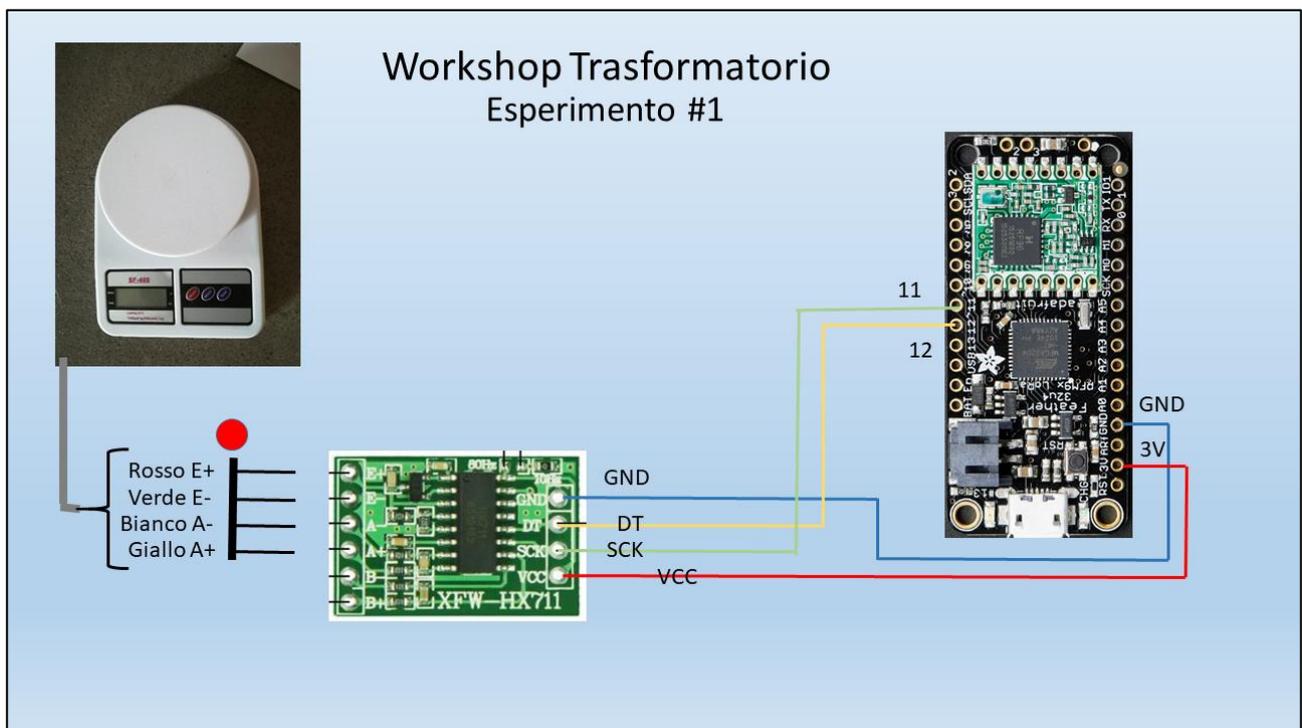
Via radio non si può decidere a chi inviare un messaggio, un ricevitore sulla stessa frequenza dei trasmettitori riceve tutto. Sta a noi provvedere a filtrare i messaggi ricevuti e selezionare solo quello che ci interessano. Inoltre un ricevitore può ricevere solo un pacchetto alla volta, quindi se arrivano due pacchetti insieme saranno persi dei dati.

I gruppi preparano alcuni pesi “campione”, pesando oggetti vari sulla bilancia, avendo preventivamente collegato il cavo di uscita della cella di carico al circuito della bilancia.

I gruppi collegano il trasmettitore alla bilancia e alla batteria e incominciano a trasmettere le pesate mettendo pesi diversi sulla bilancia. Gli sketch sono già pre-caricati e non occorre compilare e caricare alcunché.

Il prof. riceve su un modulo RX feather i pacchetti trasmessi e li fa vedere sul suo PC proiettato. I tempi tra un invio e l'altro devono essere sufficientemente lunghi.

Software usato: su TX: `Trasmetti_peso_1...4.ino`; su RX_1: `feather_RX_ESP_OLED.ino`



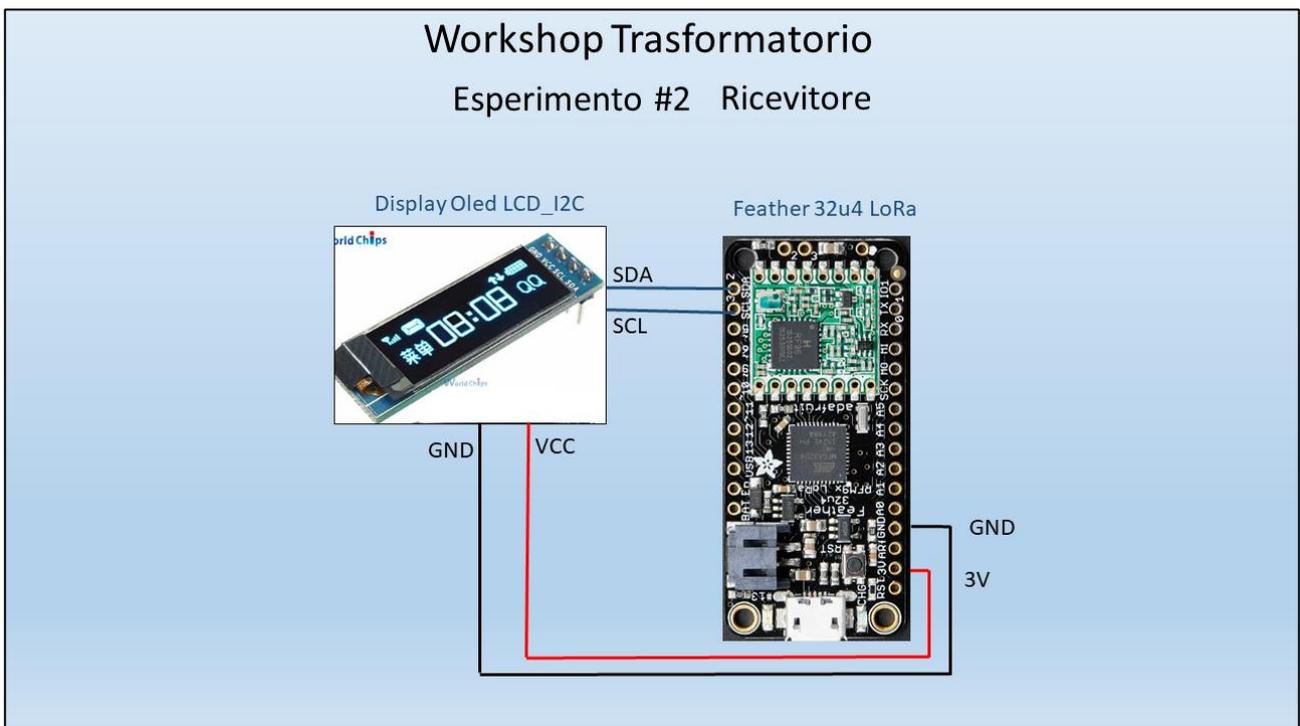
Esperimento 2

A ciascuno il suo ricevitore

Il ricevitore può filtrare i pacchetti in arrivo sulla base di un identificativo del trasmettitore. Bisogna quindi che RX e TX “si mettano d’accordo” sull’identificativo.

I gruppi usano lo stesso TX dell’esperimento 1 e collegano RX al display OLED. Sul display vedranno solo i loro dati, perché sullo sketch del RX c’è il filtro sulla stazione.

Software: su TX: `Trasmetti_peso_1...4.ino`; su RX: `feather_RX_ESP_OLED_1...4.ino`



Legenda su OLED

#xxxxx identificativo del nodo trasmittente

Vxxxx tensione in mV della batteria

Nxxxx numero progressivo del pacchetto trasmesso

Pxxxx pesata in g

Rxxxx RSSI (received signal strength indicator) è la forza del segnale ricevuto (viene dato in negativo ma visualizzato in valore assoluto).

Esperimento 3

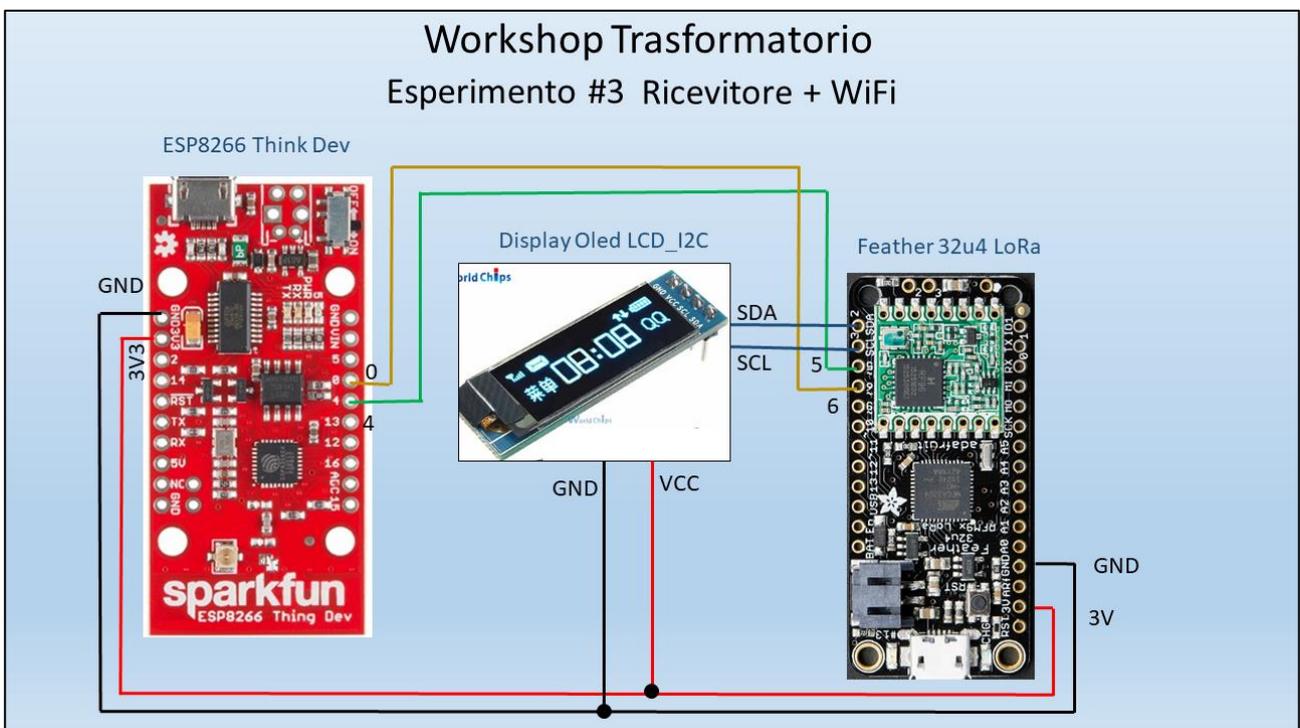
Inviemo i nostri dati su Internet

I gruppi usano lo stesso TX dell'esperimento 1, collegano RX alla scheda ESP8266.
Il server accetterà solo i dati che arrivano da una specifica scheda ESP8266 e quelli che sono stati configurati sul server stesso.

Software: su TX: `Trasmetti_peso_1...4.ino;`

su RX: `feather_RX_ESP_OLED_1...4.ino`

su ESP8266: `ESP_exosite_General_V7_Messina_1...4.ino`



Server Exosite.com

Al server ci si connette con l'account:

Username: dac@fablabmessina.it

Password: Giampilieri01

Si possono consultare i dati numerici e costruire grafici da visualizzare sulla Home page

Device e data source

I dati sul server sono divisi in gruppi, chiamati "Device". In ogni Device ci sono diversi "Data source" ognuno con un nome diverso all'interno nello stesso Device. I data source corrispondono alle variabili trasmesse dal

nodo trasmettitore. Ad esempio, i valori di peso in grammi della bilancia B1 vanno a finire nel data source chiamato "pesog_1" del Device "MessinaB1".

Il Device è logicamente legato alla scheda ESP8266 che trasmette su Internet. Due schede diverse non possono mandare dati allo stesso Device. Un Device può raccogliere dati provenienti da trasmettitori diversi se i dati di questi convergono su uno stesso ESP8266.

Per vedere i dati cronologici di un data source, cliccare sul suo nome nella tabella che compare selezionando la voce "Data" del menù in alto a sinistra.

Nella Home page è possibile creare rappresentazioni grafiche diverse dei dati usando il tasto "Add Widget".

Device presenti:

I DataSource per ognuno dei 4 Device **MessinaBx** sono:

vBatTX_x	tensione della batteria in mV
npacket_x	numero progressivo del pacchetto di dati
pesog_x	peso sulla bilancia in g
npulse_x	numero di impulsi del sensore di vibrazione
sPeriod_x	periodo in secondi in cui vengono contati gli impulsi

I DataSource del 5° Device "**MessinaSensori**" sono:

vBatTX_pluvio	tensione della batteria in mV del pluviometro
vBatTX_frane	tensione della batteria in mV del sensore frane
npacket_pluvio	numero progressivo del pacchetto di dati del pluviometro
npacket_frane	numero progressivo del pacchetto di dati del pluviometro
mm5m	mm pioggia in 5 minuti
mm1h	mm pioggia nell'ultima ora
mm3h	mm pioggia nelle ultime 3 ore
npulse	numero impulsi contati dal sensore frane nel tempo sPeriod
sPeriod	periodo di conteggio impulsi in s

Ulteriori esperimenti

Si osservano conflitti? Visto che stiamo usando tutti la stessa frequenza.

Proviamo a cambiare la frequenza portante, ogni gruppo imposta un frequenza diversa, usare decimali di 868, su TX e RX. Es: 868.1; 868.2; 867.9; 867.8

Informazioni aggiuntive

Indicativi dei trasmettitori del peso: 48001, 48002, 48003, 48004

Indicativi dei trasmettitori delle vibrazioni: 49001, 49002, 49003, 49004

Fattore di calibrazione per la bilance

B1: 247.68

B2: 240.70

B3: 238.33

B4: 241.82

Configurazione della scheda ESP8266

Ogni gruppo ha una scheda ESP8266 che viene riconosciuta dal server exosite come un "Device".

I Device sono: **MessinaB1**, **MessinaB2**, **MessinaB3**, **MessinaB4**

Un ulteriore Device: **MessinaSensori** è riservato al ricevitore del pluviometro e sensore frane.

Ad ogni device viene assegnata dal server una stringa CIK (come una password) che è inserita nella stringa http inviata al server.

Ogni variabile ha il suo nome e il suo valore. Il nome della variabile deve essere lo stesso che è stato configurato sul server, sotto la voce del Device.

Il CIK va inserito nella scheda ConfigWiFi.h degli sketch tipo

```
ESP_exosite_General_V7_Messina.ino
```

Nella stessa scheda vanno inserite le credenziali per il collegamento alla rete WiFi.

Il messaggio ricevuto da una stazione, viene confezionato come stringa http da mandare al server; a questo provvede la scheda dello sketch corrispondente all'ID della stazione. Quando la stringa è arrivata al server i dati contenuti vengono smistati nel Device relativo al CIK contenuto nella stringa e inseriti nelle rispettive variabili.

Esempio

Stringa: http....CIK...var1=100...var2=345...

Server Device: MessinaB2; Datasource: vBatTX...

Uno stesso Device può ospitare variabili di più stazioni.

Configurazione server Exosite

Sono create 5 "Device" dal nome: MessinaB1; MessinaB2; MessinaB3; MessinaB4; MessinaSensori (5 è il massimo numero di Device per un account free).

Le prime 4 sono riservate agli esperimenti condotti durante il workshop, la quinta serve per i sensori pluviometro e frane installati a Giampilierì.

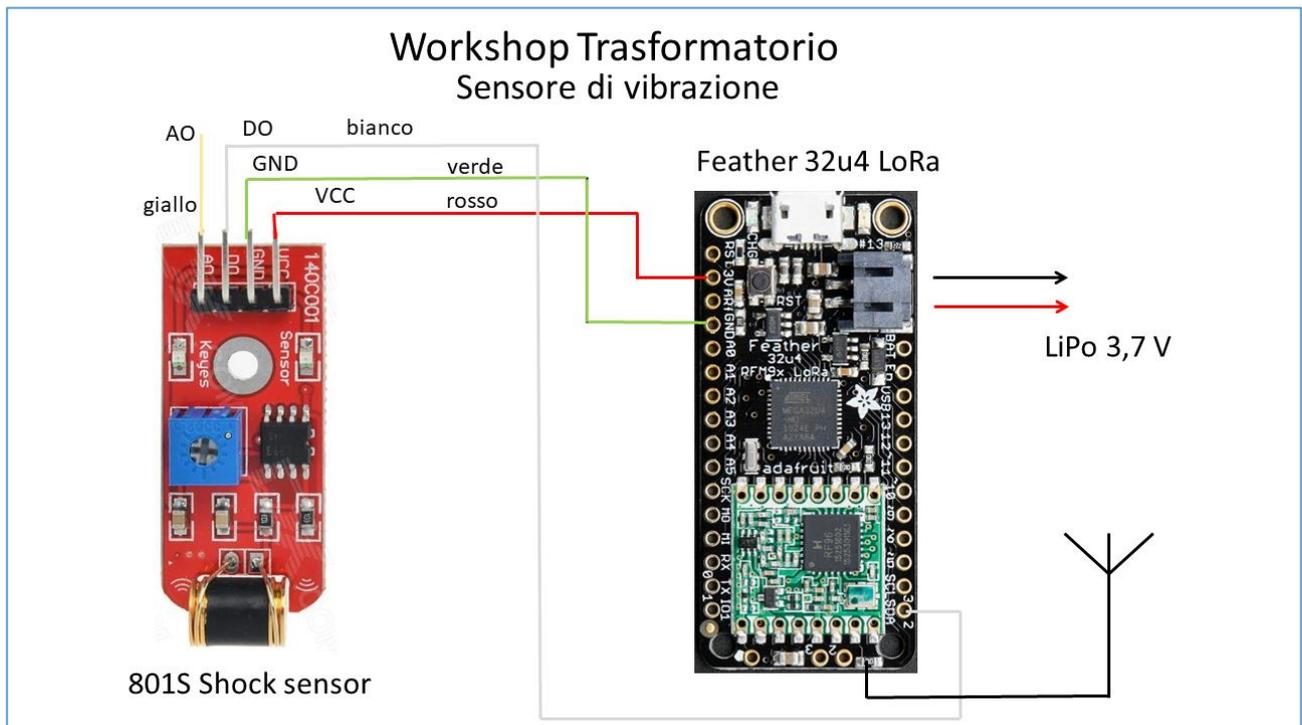
Il server riconosce il Device attraverso il CIK e il Macaddress della scheda hardware di provenienza. Se tento di accedere allo stesso Device con due schede diverse si pianta il server.

All'interno di ogni Device posso creare un certo numero di DataSource (100 in totale per un account free), che devono avere lo stesso nome di quelle che lo sketch su ESP8266 invierà nella stringa http.

Esperimento #4

Cambiamo sensore e modalità di acquisizione

Il sensore di vibrazione manda un certo numero di impulsi quando subisce uno shock. Nel programma voglio “svegliare” la CPU e contare il numero di impulsi in arrivo in un certo tempo prefissato. Dopo questo tempo invio il messaggio e la CPU torna a “dormire”.



Materiale: Sensore vibrazione 801S, Feather TX, cavo USB, Feather RX con display OLED, ESP8266

Rete WiFi con credenziali di accesso

Software:

su TX: `Trasmetti_vibrazioni.ino`; Da personalizzare sul gruppo, bisogna cambiare l'identificativo di stazione

su RX: `feather_RX_ESP_OLED.ino`

su ESP8266: `ESP_exosite_General_V7_Messina.ino`

Il numero del trasmettitore cambia da 48xxx a 49xxx. Cambia anche la lunghezza del messaggio da 10 a 12 byte. Cambiano anche le schede ID e le impostazioni sul server Exosite

Lo sketch su TX sarà fatto con Interrupt e trasmetterà il numero di impulsi. Farà uso anche dello sleep.

Cambiamo insieme lo sketch di RX e ESP8266