



TOOLKIT
by wemake.cc

Toolkit per il cambiamento

WeMake LoRaKit Trasmissione dati open source

maggio 2021



TOOLKIT
by wemake.cc

Sommario

Introduzione	2
Citizen Science	3
Il futuro richiede una visione d'insieme	4
Metti In Circolo il Cambiamento	5
Toolkit per il cambiamento	6
Autocostruire un dispositivo per la trasmissione di dati a distanza	7
IL KIT	9
Principali caratteristiche di WeMake LoRaKIT	10
Componenti contenuti nel WeMake LoRaKIT	11
Il nodo TX	15
Nodo RX	16
LE ISTRUZIONI	17
Funzionamento	18
Accesso al server adafruit IO	20
Adattare il KIT a sensori diversi	21
I FILE CONDIVISI	27
Librerie e software	28



TOOLKIT
by wemake.cc

Introduzione

WeMake è un fablab di Milano attivo dal 2014 che ha come obiettivo la divulgazione di processi e tecnologie digitali. WeMake è anche un luogo fisico, un laboratorio urbano che mette a disposizione strumenti e macchine di prototipazione avanzati grazie ai quali è possibile apprendere e condividere conoscenze legate alle tecnologie, al design e, più in generale, al “saper fare”.

In ambito educativo interviene nei contesti scolastici e tra i giovani per favorire la crescita di cittadini consapevoli e responsabili in una società sempre più globale e interdipendente. A questo scopo promuove percorsi educativi innovativi volti a rafforzare l’uso consapevole degli strumenti digitali per lo sviluppo di competenze di cittadinanza globale (interscambio, inclusione e comprensione critica, ascolto attivo, cooperazione) e per approfondire e rispondere alle sfide evidenziate dagli Obiettivi di Sviluppo Sostenibile e dall’Agenda 2030.

WeMake è promotore di progetti di social impact italiani ed europei, che coinvolgono comunità che condividono conoscenze, competenze, esperienze relative alla *digital fabrication*, alle tecnologie digitali, all’elettronica, domotica, robotica e alla *citizen science*.



WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Citizen Science

La *citizen science* è la scienza nelle mani di cittadini e cittadine che non solo sono diventati più sensibili ai problemi legati all'ambiente, ma possono attivarsi per costruire un uso della tecnologia utile al miglioramento delle nostre vite.

Come *makers* a noi piace quando la *citizen science* permette a più persone possibili di porre domande e fornire risposte su questioni scientifiche e ad indirizzare l'attenzione pubblica su questioni ambientali e sulla salute pubblica favorendo la collaborazione tra comunità di cittadini e istituzioni scientifiche attraverso l'uso di strumenti a basso costo.

Negli ultimi tre anni a **WeMake** si è aperta una riflessione e diverse sperimentazioni sull'utilizzo dell' IoT per progetti per la tutela ambientale.

Il monitoraggio dell'ambiente e applicato all'agricoltura ha infatti enormi potenzialità e porta benefici non solo agli abitanti di quelle zone ma al territorio nel suo complesso.

Paolo Bonelli è uno dei maker più attivi e ha contribuito da un punto di vista scientifico a sviluppare molti dei nostri progetti. Questo Toolkit è frutto del suo lavoro di condivisione. Paolo ha un suo sito www.coscienzambientale.com.



"Misurare la natura è come ammirarla con altri occhi, quelli dei nostri strumenti e della nostra intelligenza nell'interpretarne i dati"

Paolo Bonelli



TOOLKIT
by wemake.cc

Il futuro richiede una visione d'insieme

L'Agenda globale per lo Sviluppo Sostenibile, entrata in vigore il 1 gennaio 2016, contiene i 17 Obiettivi di Sviluppo Sostenibile (Sustainable Development Goals o SDGs) da raggiungere entro il 2030 per il futuro del pianeta e dell'umanità. I 17 SDGs, interconnessi e indivisibili, sono articolati in 169 target (traguardi) tesi a bilanciare le tre dimensioni dello sviluppo sostenibile - crescita economica, inclusione sociale, tutela dell'ambiente. Le iniziative per porre fine alla povertà estrema devono andare di pari passo con le strategie a sostegno dell'economia e rispondere anche alle sfide epocali rappresentate dalla crisi climatica, le migrazioni ambientali e la protezione dal degrado dell'ambiente marino e terrestre.





TOOLKIT
by wemake.cc

Metti In Circolo il Cambiamento

“Metti in circolo il cambiamento!” è un progetto che promuove il necessario cambiamento culturale verso i principi dell’ECONOMIA CIRCOLARE. Attualmente la maggior parte della produzione e del consumo segue il modello economico lineare prendi-usa-getta. L’economia circolare invece permette di riutilizzare i prodotti destinati alla discarica, di ri-immetterli nel ciclo di produzione e di estenderne il ciclo di vita. Il modello diventa **riduci-riutilizza-ricicla**.

“[Metti in circolo il cambiamento](#)” è un progetto educativo e culturale ma anche di formazione professionale, per promuovere idee e attività d’impresa “green” che si svolge in quattro regioni italiane (Piemonte, Emilia Romagna, Toscana, Sicilia).

Il progetto vuole anche essere un ponte tra esperienze realizzate in Italia e in Africa (tramite l’esperienza di cooperazione internazionale di LVIA) che si ispirano all’Economia Circolare per tutelare l’ambiente e creare opportunità di lavoro sostenibile. WeMake è partner del progetto. L’Agenzia Italiana per la Cooperazione allo Sviluppo è cofinanziatore del progetto (AID 011793). I contenuti di questa comunicazione rientrano sotto la sola responsabilità dei promotori e non rispecchiano necessariamente il punto di vista dell’ AICS.

L’Agenzia Italiana per la Cooperazione allo Sviluppo nata nel 2016 dalla legge di riforma della cooperazione (Legge n. 125/2014). Ha l’obiettivo di allineare l’Italia ai principali partner europei e internazionali nell’impegno per lo sviluppo. Il compito dell’Agenzia è quello di svolgere le attività di carattere tecnico-operativo connesse alle fasi di istruttoria, formulazione, finanziamento, gestione e controllo delle iniziative di cooperazione internazionale. Linea di finanziamento sull’educazione alla cittadinanza globale.



WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Toolkit per il cambiamento

L'obiettivo di questo toolkit è quello di rendere cittadine e cittadini autonomi nella rilevazione e nel monitoraggio ambientale. Ogni Toolkit è composto da tre parti che consentono di procedere in autonomia nello sviluppo SW e HW del progetto .

CITIZEN SCIENCE

La citizen science, o scienza partecipata, è il coinvolgimento attivo dei cittadini nella raccolta, analisi e interpretazione di dati a fini scientifici.

CAPABILITIES

Per rispondere alle sfide del futuro è necessaria una partecipazione attiva e collaborativa dei cittadini attraverso la promozione e il rafforzamento delle capacità e delle competenze di tutti.

COMMUNITY

Far parte di un gruppo che condivide gli stessi valori, creare connessioni con altre persone è fondamentale per ognuno di noi, così come sentirci parte di una Comunità. Quella dei maker è una grande comunità di condivisione che è accomunata dal valore collaborativo e aperto delle pratiche e della progettazione.



TOOLKIT
by wemake.cc

Autocostruire un dispositivo per la trasmissione di dati a distanza

LoRaKIT è uno strumento didattico finalizzato a far conoscere le potenzialità della tecnologia LoRa (Long Range) applicata alla trasmissione di piccole quantità di dati a grande distanza con bassi consumi energetici.

Questa tecnologia è stata applicata con successo al monitoraggio di grandezze ambientali in aree dove non sono presenti o non utilizzabili altre reti di comunicazione dati, come il WiFi, GSM/GPRS, 4G.

Il LoRa KIT è un dispositivo IOT con costi bassi e con un approccio a difficoltà graduale: il kit è pensato come piattaforma educativa utilizzabile in una prima fase così come pensato e programmato e che successivamente può essere modificato, ampliato per adattare il KIT a sensori diversi o modalità di funzionamento diverse.

Chi si è già avvicinato al microcontrollore Arduino ed ha potuto apprezzare la facilità con cui si possono creare moltissime applicazioni con questo oggetto, potrà estendere con questo KIT le sue esperienze al campo del Outdoor Monitoring, che comprende la misura e la rilevazione di tutto ciò che avviene all'aperto, fuori dalle mura di casa, come la variabilità meteorologica, l'inquinamento dell'aria e dell'acqua e molto altro.



WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Il Toolkit è composto da:



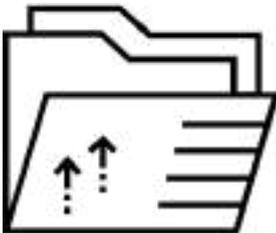
Il Kit

Include la lista dei materiali, dei software e dell'hardware necessario per la realizzazione del dispositivo



Le istruzioni

E' il manuale di istruzioni da seguire per la realizzazione del progetto. Una documentazione essenziale.



I file

Sono i file condivisi in ottica collaborativa per eseguire i progetti: stl per la stampa, dxf per il taglio laser, ma sono presenti anche stringhe di codice o scketch per la Ide



TOOLKIT
by wemake.cc



IL KIT

WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Principali caratteristiche di WeMake LoRaKIT

LoRaKIT si compone di 3 nodi trasmettitori (TX) e un nodo ricevitore (RX). Tutti i nodi sono montati su tavolette di legno indipendenti. Gli esperimenti sono concepiti per un gruppo di persone per ogni TX, mentre lo RX è in comune a tutti i gruppi. Le eventuali modifiche sul nodo RX andranno concordate tra tutti. Nel WeMake LoRa Kit ogni TX trasmette periodicamente un messaggio via radio(LoRa) (valori della temperatura e umidità dell'aria). RX riceve i messaggi dei nodi, li presenta sul display e li invia su Internet per mezzo della rete WiFi.

Open

E' modificabile e riproducibile da chiunque sappia utilizzare le risorse dell'elettronica open source.

Personalizzabile

I nodi TX e RX sono già forniti con il loro firmware ma possono essere riprogrammati caricando il programma con il cavo USB-FTDI. Il nodo RX offre la possibilità di collegarsi ad una rete WiFi esterna e trasmettere i dati rilevati ad un server su Internet. I nodi TX possono essere modificati in modo da poter trasmettere i dati di altri sensori. Oltre la modifica hardware è necessario un aggiornamento del firmware che sarà facilitato grazie alla particolare struttura con cui è scritto il codice. Anche il codice di RX può essere facilmente adattato a ricevere messaggi da altri nodi TX e/o da altri sensori.

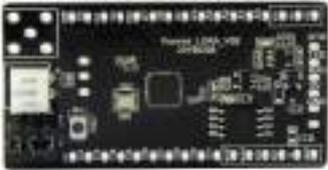
Portatile e di facile utilizzo

I TX hanno una batteria ricaricabile, quindi possono essere usati anche all'aperto. Gli RX, hanno un display che consente una lettura immediata dei messaggi inviati dai TX.



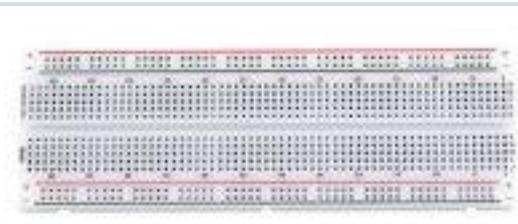
TOOLKIT
by wemake.cc

Componenti contenuti nel WeMake LoRaKIT

quantità	figura	descrizione
NODO TX		
1		TTGO T-Deer Pro Mini LoRa_V02 868 MHz
1		Batteria piatta 3,7 V 2500 mA con connettore
1		Connettori per batteria LiPo
1		Resistenza 220
1		LED
1		Sensore di Temperatura e umidità SHT21

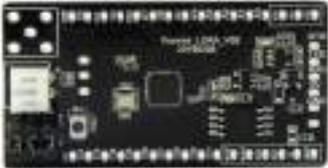


TOOLKIT
by wemake.cc

quantità	figura	descrizione
1		Antenna stilo per 868 MHz (di solito compresa nella scheda TTGO)
1		Breadboard da 830 buchi
q.b.		ponticelli
1		Basetta in legno



TOOLKIT
by wemake.cc

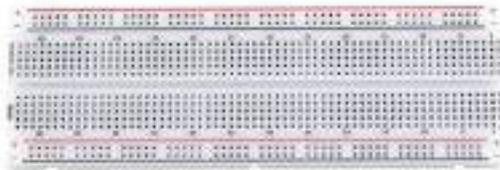
NODO RX		
quantità	figura	descrizione
1		TTGO T-Deer Pro Mini LoRa_V02 868 MHz
1		Wemos D1 mini con antenna WiFi interna
1		Display 16 x 2 con interfaccia I2C
1		Resistenza 220
1		LED
1		Antenna stilo per 868 MHz (di solito compresa nella scheda TTGO)

WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

1



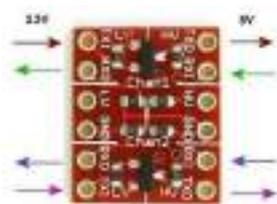
Breadboard da 830
buchi

q.b.



ponticelli

1



Level Converter a
due canali

1

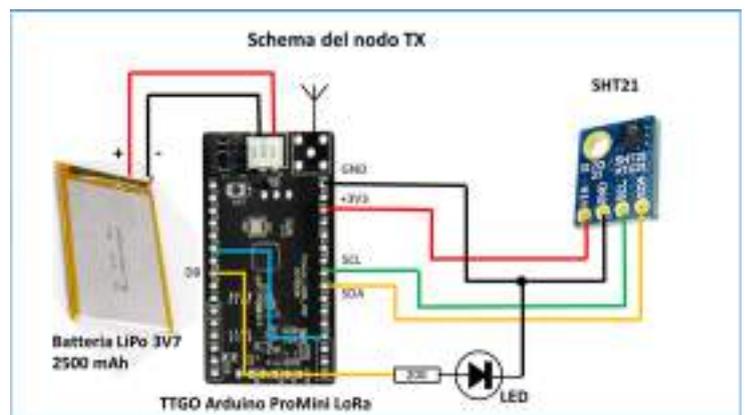
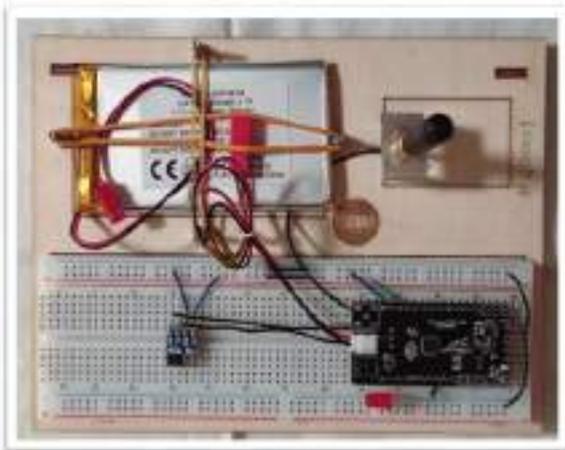


Basetta in legno



TOOLKIT
by wemake.cc

Il nodo TX



Ogni nodo TX è composto da:

- una scheda Arduino Pro-Mini LoRa della TTGO;
- un sensore di temperatura e umidità SHT21 collegato alla scheda TTGO via protocollo I2C;
- una batteria LiPo da 3.7 V e 2500 mAh che alimenta tutto il nodo tramite la TTGO;
- un LED collegato alla scheda TTGO che lampeggia ogni volta che viene inviato un messaggio;
- un' antenna adatta alla frequenza di 868 MHz usata da LoRa;
- un cavo USB-FTDI per la programmazione della scheda TTGO e la ricarica della batteria.

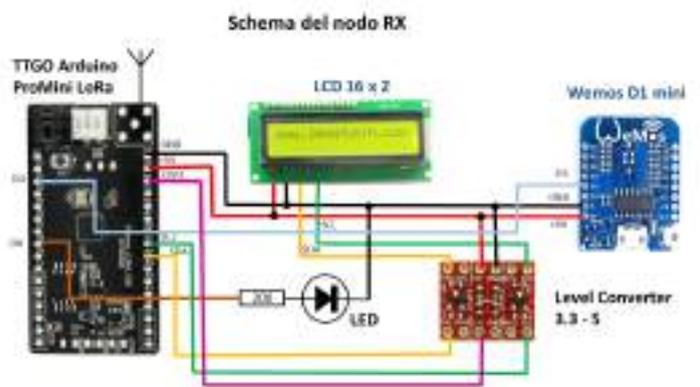
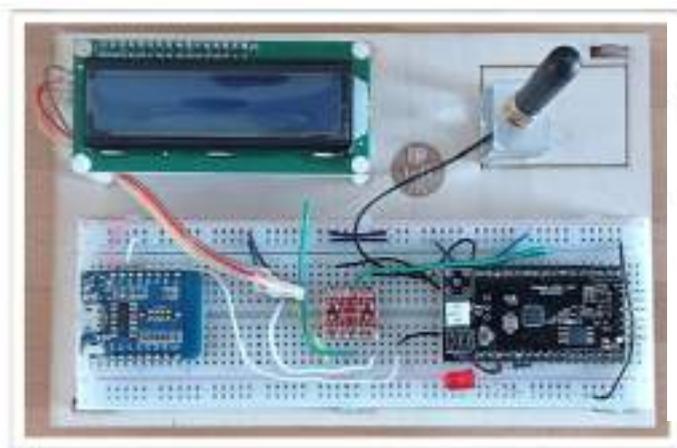


Il nodo TX è alimentato dalla batteria ricaricabile LiPo ricaricabile con il cavo USB-FTDI collegato al PC o ad un alimentatore plug-in da 5V. Il cavo FTDI va inserito nei sei pin maschi della scheda TTGO in modo che il filo nero (GND) stia a sinistra guardando la scheda con i 6 pin verso il basso.



TOOLKIT
by wemake.cc

Nodo RX



Il nodo RX è composto da:

- una scheda Arduino Pro-Mini LoRa della TTGO;
- una scheda Wemos D1 mini;
- un display LCD 16 x 2 collegato alla scheda TTGO via protocollo I²C;
- un LED collegato alla scheda TTGO che lampeggia da ogni messaggio ricevuto;
- un' antenna adatta alla frequenza di 868 MHz usata da LoRa.
- un cavo USB-FTDI per la programmazione della scheda e l'alimentazione del nodo.

Il nodo RX può essere alimentato con il cavo USB-FTDI tra la TTGO e un PC o un alimentatore plug-in da 5V con presa USB; in alternativa può essere alimentato da un cavo micro USB tra la Wemos e PC o plug-in; non è prevista l'alimentazione a batteria visto il consumo di corrente più sostenuto rispetto al nodo TX.



TOOLKIT
by wemake.cc



LE ISTRUZIONI

WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Istruzioni per l'uso

Il KIT è pronto all'uso. Una volta accesi i nodi TX e RX, i dati sono rilevati e trasmessi via radio LoRa dal TX al RX e inviati su Internet al server Adafruit.

Accensione dei TX: collegare la scheda TTGO alla batteria, unendo le due parti del connettore rosso; collegare il cavo USB-FTDI tra la scheda e il PC per visualizzare sul monitor seriale della IDE i messaggi che sono trasmessi.

Accensione del RX: collegare il cavo USB-FTDI alla scheda TTGO e ad un PC con installata la IDE di Arduino e aprire il monitor seriale.

Funzionamento

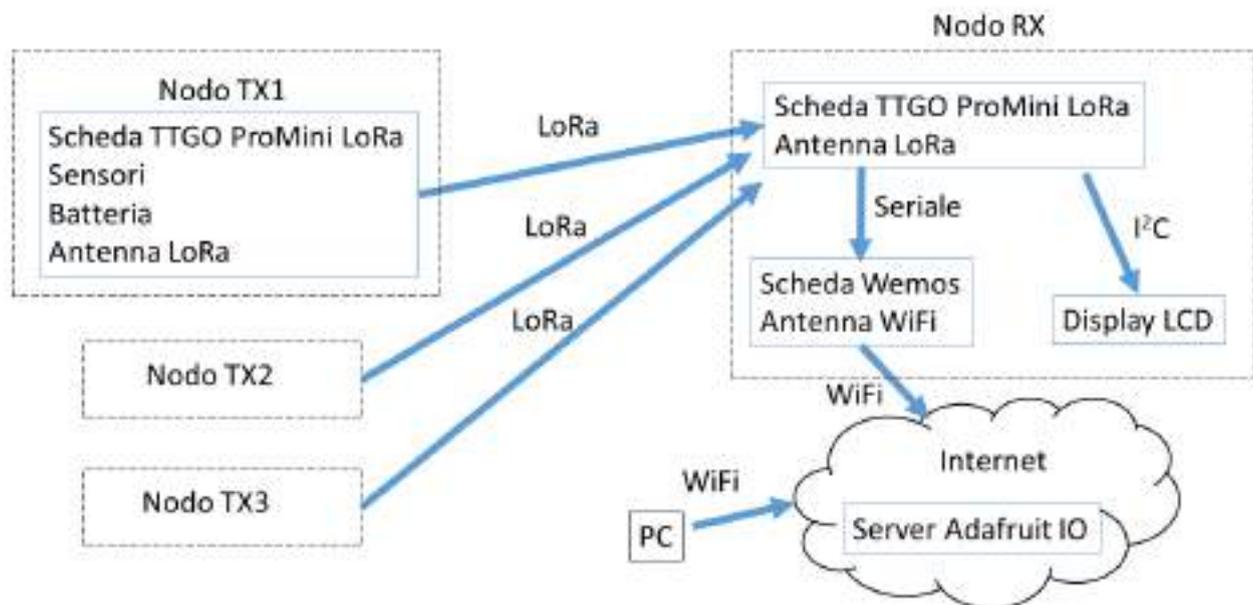
1. Ogni nodo TX, in modo indipendente dagli altri, trasmette periodicamente la temperatura e l'umidità rilevate dal sensore SHT21. Il periodo è di circa 25 s. I dati sono trasmessi all'interno di un messaggio che contiene anche altre informazioni sul nodo.
2. La trasmissione è segnalata dal lampeggio del LED rosso.
3. Ogni messaggio TX porta al suo interno l' identificativo del TX: 40001, 40002, 40003. L'identificativo è anche scritto sulla tavoletta di legno del nodo TX.
4. RX riceve tutti e soli i messaggi dei 3 TX, li presenta sul display LCD e li invia alla scheda Wemos per la loro trasmissione ad un server Internet. La Wemos è infatti dotata di antenna interna e modulo radio WiFi. Affinchè la scheda Wemos possa ri-trasmettere i dati su Internet è necessario che sia disponibile una rete WiFi e le sue credenziali di accesso siano inserite nel firmware della Wemos.
5. La comunicazione tra le due schede TTGO e Wemos del RX avviene via protocollo seriale asincrono realizzato con un filo che collega un pin digitale su ogni scheda. Un protocollo seriale completo richiede due fili, ma in questo caso è prevista solo la trasmissione dei dati in un senso, dalla TTGO alla Wemos.



TOOLKIT
by wemake.cc

6. La Wemos è stata programmata per accedere e trasmettere i dati al server su Internet AdafruitIO (<https://io.adafruit.com/>) che ne permette una visualizzazione grafica; occorre però creare un account su questo server per ottenere una chiave di accesso. Più avanti si riportano le spiegazioni per fare questo passaggio.

ARCHITETTURA DELLA MINI RETE LORA DEL KIT





TOOLKIT
by wemake.cc

Accesso al server adafruit IO

Collegarsi con un browser al server <https://io.adafruit.com/>

- cliccare su: **Get Started for free**
- mettere i propri dati oltre ad una email e password e cliccare su **create an account**
- aspettare la mail che abilita ad entrare nel server
- entrare nel server con username e password precedentemente inserite
- cliccare in alto su **IO**
- cliccare su **My Key**
- copiare quello che sta scritto nella finestra Arduino, qui sotto un esempio:

```
#define IO_USERNAME  "outdoorsensing"  
#define IO_KEY       "aio_JPct47rppv6ct8IgwW3AjsJD8"
```

- Queste due righe di istruzioni devono essere poi incollate nel programma Wemos nel folder `Config.h` e commentare eventuali righe preesistenti (durante il workshop questa operazione è già fatta)
- Nello stesso folder andranno scritte le credenziali di accesso alla rete WiFi:

```
#define WIFI_SSID    "xxxxxxxx"  
#define WIFI_PASS   "xxxxxxxx"
```

- Dopo queste operazioni tutto il programma va ricompilato e caricato sulla scheda Wemos
- Per verificare se i dati arrivano al server, cliccare su **Feeds** nella finestra di **IO**, apparirà un elenco delle variabili arrivate sul server, queste sono:

```
Temp1, Humi1    per il nodo 40001  
Temp2, Humi2    per il nodo 40002  
Temp3, Humi3    per il nodo 40003
```

Oltre alla tensione delle tre batterie: vBat1, vBat2, vBat3



TOOLKIT
by wemake.cc

Cliccando su una di queste apparirà l'elenco dei valori arrivati con la data e l'ora di arrivo al server

Grafica sul server Adafruit

- Cliccare su **Dashboards** sul menù **IO**,
- Cliccare su **+ New Dashboard** e dare un nome alla stessa,
- Creata la Dashboard cliccare sul suo nome,
- Nella finestra che si apre è possibile creare dei grafici, cliccare il simbolo dell'**ingranaggio** in alto a destra e cliccare **+ Create New Block**,
- Scelto il tipo di grafico, si apre una finestra dove bisogna scegliere le variabili presenti in Feeds che si vuole graficare e le opzioni sugli assi. Si possono creare più **block** e sistemarli come si vuole nella pagina della Dashboard

Adattare il KIT a sensori diversi

Per adattare il KIT a rilevare e trasmettere grandezze rilevate da sensori diversi, occorre fare modifiche all'hardware e al codice del TX e RX.

Hardware

Un nodo può gestire diversi sensori compatibilmente con la capacità hardware della scheda TTGO. In particolare la scheda TTGO Pro mini, nella configurazione usata nel KIT, ha i seguenti pin di I/O liberi:

D3, D5, D7, A1, A2, A3, A6, A7

Mentre i seguenti sono impegnati in varie funzioni:

D0 (RX), D1(TX) servono alla comunicazione con il PC durante la fase di up-loading del programma e per il monitor seriale.



TOOLKIT
by wemake.cc

D13, D12, D11, D10, D9, D2 destinati alla comunicazione con il modulo radio,

D8 può essere liberato se si rinuncia al LED che segnala una trasmissione o ricezione di un messaggio,

A0 è collegato al D6 per monitorare la tensione della batteria.

Come sulla scheda Arduino UNO, i pin A4, A5 sono anche usati per le comunicazioni I²C e rappresentano rispettivamente il segnale SDA e SCL. Sullo stesso “bus” I²C possono essere collegati più dispositivi che usano questo protocollo a patto che i loro indirizzi siano diversi.

Da tener presente che i pin di ingresso analogico Ax, sono utilizzabili anche come pin di output digitale.

I segnali digitali in entrata e uscita dalla scheda TTGO sono a 3.3 V, anche la massima tensione ammessa sui pin analogici è 3.3V, pertanto bisogna stare attenti che i dispositivi collegati alla scheda non inviino segnali con tensioni superiori, p.e. 5 V. In caso contrario bisogna ricorrere ai “Level Converter”, piccoli circuiti che adattano i segnali digitali alla tensione che vogliamo.

Software

Eseguiti i collegamenti hardware del/dei nuovi sensori alla scheda TTGO del TX, occorre modificare lo sketch per generare il messaggio che contiene le nuove variabili.

Per prima cosa si spiega qui di seguito come è strutturato il messaggio generato da TX.

Ogni messaggio è composto da una testata lunga sempre 8 byte e una parte che contiene i valori delle variabili misurate dal sensore o dai sensori. Nel caso del sensore in dotazione SHT21, la parte delle variabili è fatta da altri 8 byte



TOOLKIT
by wemake.cc

Testata

Byte

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

ID_TX (int)	Length (int)	vBatTX (int)	npacket (int)
-------------	--------------	--------------	---------------

Variabili misurate

Byte

8	9	10	11	12	13	14	15
---	---	----	----	----	----	----	----

Temp (float)	Humi (float)
--------------	--------------

Significati:

- ID_TX: (intero senza segno) identificativo del nodo trasmittente
- Length: (intero senza segno) lunghezza in byte di tutto il messaggio (in questo caso 16 byte)
- vBatTX: (intero senza segno) tensione della batteria in mV
- npacket: (intero senza segno) numero progressivo del messaggio (si azzera al reset o quando maggiore di 60000)

- Temp: (numero float) temperatura in °C con decimali
- Humi: (numero float) umidità relativa in % con decimali

Se ad esempio dovessi trasmettere anche la concentrazione di polveri sottili PM10 e PM2.5, espresse in $\mu\text{g}/\text{cm}^3$, la parte variabile del messaggio, sarebbe composta da altri 8 byte, per due variabili di tipo float da 4 byte ciascuna. Quindi la lunghezza totale del messaggio diventerebbe di 24 byte e il valore della variabile `Length` nella testata dovrà essere aggiornato a 24.



TOOLKIT
by wemake.cc

Tutto il software dei nodi TX e RX è diviso in moduli (Folder nella IDE) e solo alcuni vanno modificati nel caso si cambino o aggiungano sensori.

Sketch TX da caricare sulla scheda TTGO del TX

folder	Cosa contiene	Da modificare
TX	Main program	Si Aggiornare se necessario il valore della variabile <code>ncicliMax</code> a seconda del periodo di invio messaggi: ogni ciclo sono 8 secondi. <code>RF95_FREQ, RF95_POW</code> , se si vogliono modificare frequenza portante e potenza radio di uscita; tener presente che la frequenza portante deve essere uguale a quella impostata sul nodo RX <code>ECHO = 1</code> , se si vuole usare il monitor seriale in fase di test, 0 altrimenti.
Packet.h	Dichiarative delle variabili da trasmettere nel messaggio	Si Inserire le variabili lette dai vari sensori, che si vuole trasmettere, identificativo del TX e lunghezza del payload in byte
Sens.h	Dichiarative riguardanti il sensore	Si Inserire le dichiarative necessarie agli eventuali sensori diversi da quello del KIT
SensInit	Funzione di inizializzazione sensore	Si Inserire le inizializzazioni necessarie agli eventuali sensori diversi da quelle del KIT
ReadSensor	Funzione per la lettura sensore	Si Inserire le istruzioni di lettura degli altri eventuali sensori.
Battery	Funzione per leggere la tensione della batteria	Non modificare
Blinking	Funzione lampeggio del LED	Non modificare
Nword	Funzione per approssimazione all'intero più vicino	Non modificare
Transmit	Funzione per trasmette il pacchetto di dati	Non modificare
altDelay	Funzione di ritardo	Non modificare
Radio.h	Dichiarative riguardanti la radio	Non modificare Solo se si usa una scheda diversa dalla TTGO, commentare e scommentare i vari <code>#define</code>



TOOLKIT
by wemake.cc

Chi ha già un minimo di esperienza di programmazione nel C++ di Arduino, può facilmente modificare le parti del software che sono riportate nella tabella. Per quanto riguarda le istruzioni C++ che servono ad interrogare i sensori collegati, possono essere usati gli sketch di esempio contenuti nelle librerie o i tutorial relativi agli stessi sensori facilmente trovabili su Internet.

E' bene non superare il limite di 50 byte per la lunghezza totale del messaggio (Intestazione + variabili misurate).

Il formato delle variabili da trasmettere potrà essere di qualsiasi tipo: int, float, byte, char... e andrà dichiarato nell'istruzione `typedef struct` del folder `Packet.h`

Sketch `RX_LCD_Wemos` da caricare sulla scheda TTGO del RX

folder	Cosa contiene	Da Modificare se si aggiungono altri TX da ricevere o altri sensori
<code>RX_LCD_Wemos</code>	Main program	No Settare la variabile ECHO secondo necessità
<code>Packet.h</code>	Dichiarative delle variabili misurate dai vari sensori TX che vogliamo ricevere. Le strutture sono uguali a quelle dei singoli sketch dei TX.	Si
<code>RadioRX.h</code>	Dichiarative per la radio	No
<code>Action</code>	Funzione che decide le azioni da fare in quando è ricevuto un messaggio da un TX	Si
<code>Blinking</code>	Funzione per lampeggio	No
<code>Nword</code>	Funzione per la trasformazione di una variabile float in un intero troncato senza segno	No
<code>Print40001,</code> <code>Print40002,</code> <code>Print40003</code>	Funzioni per stampare su PC e su Display. Ce n'è una per ogni TX.	Creare per ogni nuovo TX se si vuole visualizzare ciò che è ricevuto su Display e PC
<code>RadioRX_Init</code>	Funzione per l'inizializzazione della radio	No
<code>Receive</code>	Funzione per la ricezione dati via radio	No
<code>SerialTransmit2</code>	Funzione per l'invio dei dati alla scheda Wemos	No



TOOLKIT
by wemake.cc

Sketch `Wemos` da caricare sulla scheda Wemos del RX

folder	Cosa contiene	Da Modificare se si aggiungono altri TX da ricevere o altri sensori
<code>Wemos</code>	Main program	No Settare la variabile ECHO secondo necessità
<code>Packet.h</code>	Dichiarative delle variabili misurate dai vari sensori TX che vogliamo ricevere. Le strutture sono uguali a quelle dei singoli sketch dei TX.	Si
<code>SerialReceive</code>	Funzione per la ricezione dei dati via seriale	No
<code>Action</code>	Funzione che decide le azioni da fare in quando è ricevuto un messaggio da un TX	Si
<code>Config.h</code>	Dichiarazioni per il collegamento alla rete WiFi e al server	Si
<code>ID40001, ID40002, ID40003</code>	Funzioni per stampare su monitor seriale ed inviare i dati al server Ce n'è una per ogni TX.	Da creare per ogni nuovo TX
<code>Constant.h</code>	Dichiarazioni delle librerie e delle variabili globali	No



TOOLKIT
by wemake.cc



I FILE CONDIVISI

WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc



TOOLKIT
by wemake.cc

Librerie e software

WeMake LoRaKIT ha una sua repository di file all'interno dell'account di [WeMake di Github](#).

Potrete trovare:

- il file svg per taglio laser contenitore
- gli sketch caricati sulle schede del Kit

I programmi che sono caricati sulle schede del KIT sono:

- Sulla scheda TTGO del nodo TX: TX
- Sulla scheda TTGO del nodo RX: RX_LCD_Wemos
- Sulla scheda Wemos del nodo RX: Wemos

I programmi citati sopra richiedono alcune **librerie**, oltre a quelle standard di Arduino, che sono:

Librerie per il nodo TX

```
#include <RH_RF95.h>
```

questa libreria fa parte del pacchetto RadioHead e serve a far funzionare il modulo radio.

```
#include "Adafruit_SleepyDog.h"
```

contiene la funzione di sleep utilizzata per diminuire i consumi di energia tra una trasmissione e l'altra.

Ambedue le istruzioni `#include` si trovano nel folder `Radio.h` del programma TX

```
#include <Wire.h>
```



TOOLKIT
by wemake.cc

```
#include <SHT2x.h>
```

La prima serve a gestire le comunicazioni con protocollo I²C, la seconda serve a leggere il sensore SHT21, con quel protocollo, e a ricavare i valori di temperatura e umidità.

Queste istruzioni si trovano nel folder `Sens.h` del programma `TX`.

Librerie per il nodo RX

```
#include <RH_RF95.h>
```

Come per il nodo TX, gestisce la ricezione radio

```
#include <SoftwareSerial.h>
```

gestisce la comunicazione seriale tra le schede TTGO e Wemos, si trovano nel folder `Radio.h` del programma `RX_LCD_Wemos`

```
#include <LiquidCrystal_I2C.h>
```

gestisce il display LCD 16 x 2 via protocollo I²C (la libreria `Wire.h` è chiamata da questa); si trova nel folder `Radio.h`



TOOLKIT
by wemake.cc

Librerie per la scheda Wemos

```
#include <AdafruitIO_WiFi.h>
```

Gestisce le comunicazioni con il server AdafruitIO e si trova nel folder Config.h

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

```
#include <ESP8266WiFiMulti.h>
```

Servono alla comunicazione tra la scheda e la rete WiFi

```
#include <SoftwareSerial.h>
```

gestisce la comunicazione seriale tra le schede TTGO e Wemos; tutti e tre gli

```
#include
```

 si trovano nel folder Constant.h

Una trattazione completa sulla libreria RadioHead con molte informazioni sui parametri radio e loro significato si trova qui:

<https://www.airspayce.com/mikem/arduino/RadioHead/index.html>



TOOLKIT
by wemake.cc

WeMake LoRaKit è stato redatto da WeMake sulla base del lavoro di sperimentazione e prototipazione di Paolo Bonelli.

E' stato rilasciato nell'ambito di progetto "[Metti in circolo il cambiamento](#)".

Referente di progetto: Cristina Martellosio: cristina@wemake.cc



[CC BY NC](#)



WeMake Fablab - Via Guerzoni 23 - 20158 Milano
hello@wemake.cc - www.wemake.cc