

## IoT con Tuya (<https://www.tuya.com>)

Lo scopo di questa è utilizzare dispositivi IoT Tuya-compatibili Wi-Fi (lampadine, prese, strisce led, ecc) senza appoggiarsi ai servizi cloud forniti da Tuya.

La procedura standard (dal punto di vista dell'utilizzatore finale) per l'utilizzo di un dispositivo Tuya-compatibile WiFi è la seguente:

- Si resetta il dispositivo
- Si lancia da Android l'applicazione SmartLife e ci si registra
- Si sceglie l'aggiunta manuale di un dispositivo
- Il dispositivo viene trovato ed aggiunto al repository (su cloud)
- A questo punto il dispositivo è controllabile da remoto con una qualsiasi delle applicazioni Tuya, ad esempio SmartLife o dal sito di Tuya

L'attività di reverse engineering ha permesso di individuare i protocolli e permettere quindi il controllo dei dispositivi da applicazioni Python, senza usare le applicazioni né il repository su Tuya

Una sintetica descrizione di tali protocolli è la seguente:



**Bill Sobel (Zaphod)**, lives in San Diego, CA (2006-present)

Answered 2 years ago · Author has 126 answers and 19.6K answer views

Originally Answered: What protocols do the tuya smart devices use?

The vast majority of their devices use UDP, TCP, HTTP, and MQTT. UDP is used for local network discovery and to help enable the device to receive the wifi credentials without being on the wifi network to begin with. TCP is used for local control. MQTT is used for remote control (when TCP is not available). Devices use HTTP to talk to the Tuya cloud for registration, firmware updates, etc.

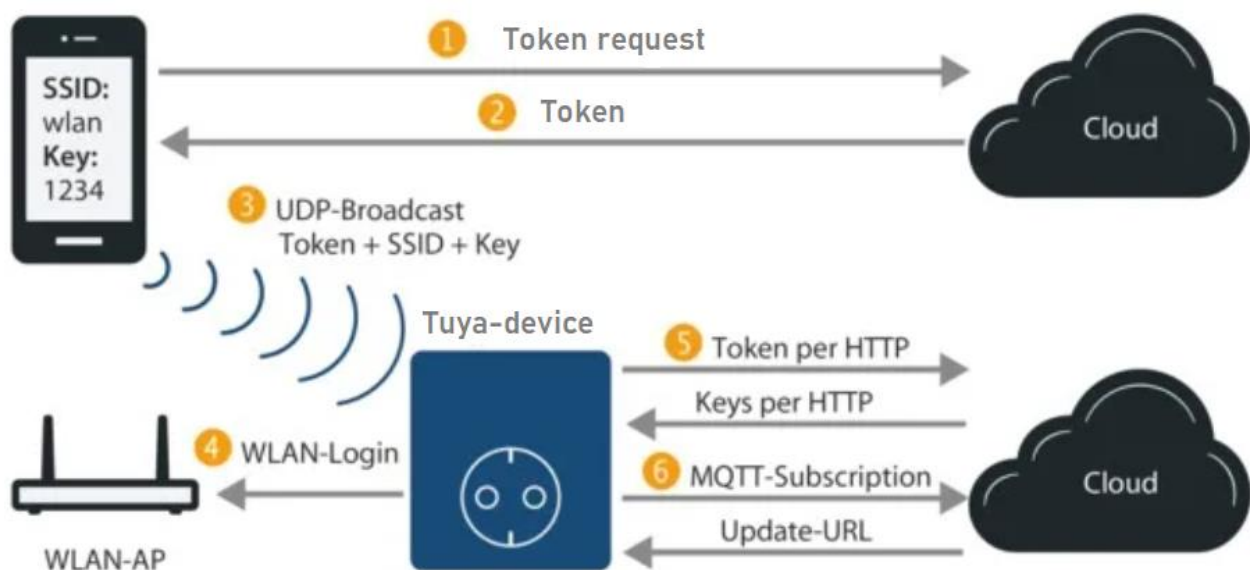
3.9K views · View upvotes

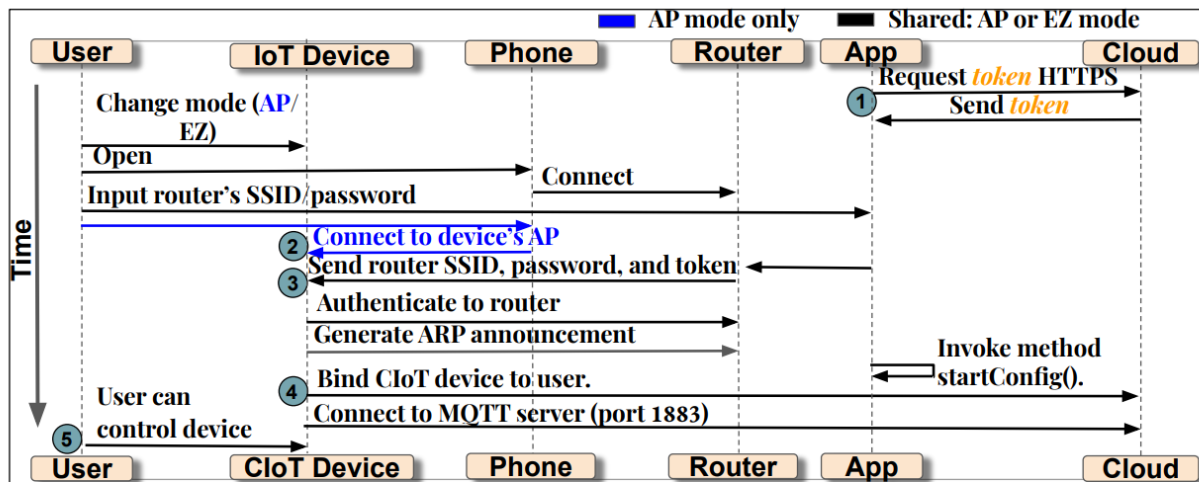


2



1





**Figure 1: A summary of the data exchanged involved during the EZ and AP Modes device setup process.**

(<https://conferences.sigcomm.org/imc/2021/pdf/15.pdf>)

L'obiettivo è controllare il dispositivo localmente (quindi via TCP) da un'applicazione Python che gira nella stessa rete locale del dispositivo stesso, utilizzando la libreria **tinytuya**.

La comunicazione è criptata ma è possibile acquisire la chiave di cifratura in due modi differenti:

1. Nel modo ufficiale, cioè registrandosi su Tuya come sviluppatore e seguendo le indicazioni nell'help di tinytuya (<https://pypi.org/project/tinytuya/1.0.4>)
2. Accedendo al file di configurazione di SmartLife su Android, dopo aver aggiunto il dispositivo interessato, e copiare dal file di configurazione la chiave.

Di seguito è illustrato come procedere in questo secondo modo.

## 1. Installazione del dispositivo Tuya-compatibile

La prima operazione da compiere è l'installazione e la configurazione del dispositivo, acquisendolo all'interno dell'applicazione SmartLife. I passi da compiere sono i seguenti:

- a. Collegamento del dispositivo Android sulla stessa rete WiFi dove si conatterà successivamente il dispositivo Tuya-compatibile. La rete WiFi deve avere un accesso diretto ad Internet, senza proxy.
- b. Installazione applicazione SmartLive su Android.
- c. Registrazione di un utente sulla piattaforma cloud di Tuya
- d. Aggiunta del dispositivo WiFi, seguendo le indicazioni dell'applicazione e la tipologia del dispositivo. Consultare a tale scopo le informazioni fornite col dispositivo Tuya specifico.

- e. Provare il corretto funzionamento del dispositivo utilizzando le funzionalità di gestione presenti in SmartLife.

## 2. Accesso ad Android come root

*Per accedere ai file di configurazione è necessario accedere ad Android come utente root, in quanto i file di configurazione sono in directory normalmente protette.*

*Questa operazione risulta essere piuttosto complessa se effettuata su un dispositivo Android reale, mentre risulta più rapida se effettuata con un telefono Android simulato.*

*A tal scopo è necessario installare un emulatore, ad esempio Bluestacks, seguendo le indicazioni seguenti:*

- a. Loggarsi come amministratore della macchina (eventualmente) sulla quale si installerà l'emulatore; non è necessario che la rete WiFi sia la stessa usata nelle fasi 1, 4 e 5.
- b. Scaricare Bluestacks a [questo link](#) e procedere all'installazione
- c. Accedere alla cartella `C:\ProgramData\BlueStacks_nxt`
- d. Nel file di configurazione `bluestacks.conf` modificare la seguente riga:

```
bst.instance.Nougat64.enable_root_access="1"
```

- e. Lanciare Bluestacks

## 3. Installazione applicazioni

*Per accedere ai file di configurazione è necessario accedere ad Android come utente root, in quanto i file di configurazioni sono in directory normalmente protette.*

*Questa operazione risulta essere piuttosto complessa se effettuata su un dispositivo Android reale, mentre risulta più rapida se effettuata con un telefono Android simulato.*

- a. Scaricare (su Windows e non da PlayStore) la versione SmartLife 3.6.1 a [questo link](#)
- b. Cliccare sul file apk scaricato: si installerà in Bluestacks
- c. Effettuare la stessa operazione per il file manager a [questo link](#)

## 4. Reperimento chiave di cifratura

- a. Accedere in SmartLife con le stesse credenziali utilizzare per la registrazione del dispositivo Tuya-compatibile: dopo essersi loggati si potrà operare sul dispositivo e nel file di configurazione di SmartLife saranno riportate le informazioni per l'accesso fra cui la chiave di cifratura.
- b. Lanciare il file manager installato precedentemente e nelle impostazioni attivare l'opzione "Sfoglia root" per avere l'accesso a tutto il file system
- c. Accedere alla cartella `/data/data/com.tuya/smartlife/shared_prefs`
- d. Sarà presente un file con un nome del tipo:

```
preferences_global_keyeu16374240184271K6E9.xml
```

- e. Copiarlo nella cartella *Download*
- f. Con l'applicazione preinstallata *Media Manager* presente nel gruppo *App di sistema* effettuare l'esportazione del file su Windows
- g. Cercare nel file di configurazione la chiave *devId* che riporta il codice del dispositivo (ad esempio 3081138394b97e08c0f2) e la chiave *localKey* che riporta la chiave di cifratura (ad esempio 2aec91701bb2b652)

## 5. Applicazione Python

- a. Installare la libreria *tinytuya* (<https://pypi.org/project/tinytuya/1.0.4>)
- b. Rintracciare l'indirizzo ip del dispositivo Tuya-compatible con l'utility fornita:

```
py -m tinytuya -nocolor
```

- c. Arrestare l'applicazione SmartLife dai vari dispositivi Android (alcuni dispositivi Tuya non sono in grado di accettare più connessioni)
- d. Utilizzare il seguente programma di prova:

```
import tinytuya
import time

DEV_ID='bf6803f790ab9fc509ddlz'
LOCAL_KEY='9e7aa8adc9ff56e3'
IP_ADDR='192.168.43.80'

#per le lampade o le strisce
d = tinytuya.BulbDevice(DEV_ID, IP_ADDR, LOCAL_KEY)
#per le prese
#d = tinytuya.OutletDevice(DEV_ID, IP_ADDR, LOCAL_KEY)

d.set_version(3.3)

d.turn_on()
time.sleep(2)
d.turn_off()
```