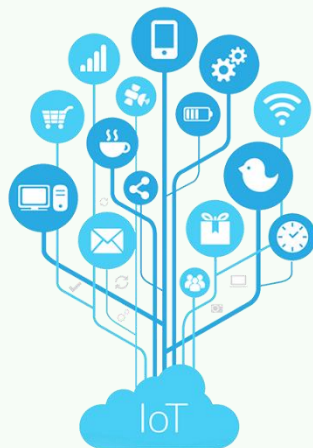


TP n°2 : Réseaux

MASTER RÉSEAUX ET TÉLÉCOM



VERSION : 10 juin 2021

Florent NOLOT
UNIVERSITÉ DE REIMS CHAMPAGNE ARDENNE

Table des matières

I.	Première communication.....	2
A.	SmartRF Studio	2
1.	Téléchargement et installation	2
2.	Configuration et test de communication	2
B.	Configuration <i>rfPacket TX</i>	3
C.	Configuration <i>rfPacketRX</i>	4
II.	API EasyLink.....	5
III.	AT commandes.....	5
IV.	Réseau sans fils de capteur	6
V.	Exercices.....	6
A.	Exercice 1 : Transfert de données	6
B.	Exercice 2 : Mise en forme des données.....	7
VI.	Annexe.....	8
A.	Ressources.....	8
1.	RF User's Guide	8
2.	CC1350 Technical Reference Manual.....	8
3.	CC1350 Datasheet	8
4.	CC1352 Technical Reference Manual.....	8
5.	CC1352 Datasheet	8
6.	RF Driver APIs	8
7.	Kernel APIs.....	8
8.	AT commandes.....	8
9.	JSON API	8



I. Première communication

Pour ce tutoriel on utilise deux LaunchPads. Le LaunchPad CC1350 est utilisé pour l'émission de paquets (TX) et le LaunchPad CC1352 est utilisé pour la réception de paquets (RX).

A. SmartRF Studio

1. Téléchargement et installation

Smart RF Studio est un logiciel permettant d'évaluer et de tester les paramètres réseaux des appareils (capteurs /cartes développement) de Texas Instrument, il peut aussi recevoir et envoyer des paquets quand il est connecté à un appareil. De plus, après avoir paramétré la partie réseaux des appareils, on peut exporter ces paramètres en code C dans CCS.

Pour obtenir le logiciel SmartRF Studio, Cliquez sur « View » dans la barre d'outils, puis sur « CCS App center », recherchez SmartRF Studio et installez la dernière version.

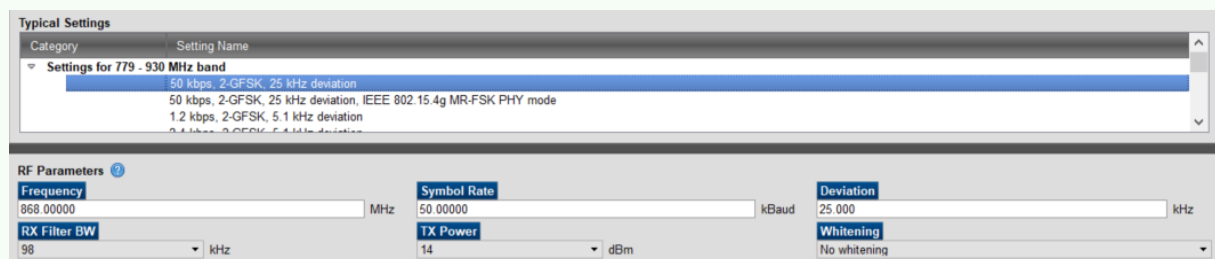
2. Configuration et test de communication

Dans un premier temps connectez les deux LaunchPads au PC puis lancez SmartRF Studio. Vous devriez voir la liste des deux appareils connectés, si ce n'est pas le cas cliquez sur la flèche de rafraîchissement.



Double cliquez sur un des LaunchPads connectés et choisissez « Proprietary mode », puis répétez l'opération avec l'autre LaunchPad. Ainsi, il y a deux fenêtre de contrôle, un pour chaque LaunchPad.

Vérifiez que les paramètres réseaux des deux LaunchPad correspondent ceux de l'image ci-dessous.



Puis positionnez l'interface du LaunchPad CC1352 sur « Packet RX » et cliquez sur « Start », puis positionner celle du LaunchPad CC1350 sur « Packet TX » et cliquez sur « Start ».

Désormais vous pouvez visualiser dans la fenêtre du LaunchPad CC1352 les paquets reçus ainsi que l'indication de la puissance du signal reçu (RSSI), le nombre de paquet que bien reçu ou non, et les taux d'erreurs.



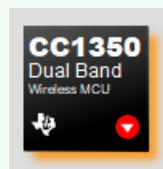
B. Configuration *rfPacket TX*

Importez l'exemple *rfPacketTX* depuis le Ressource Explorer : Software → SimpleLink Academy → Exemples → Development Tools → CC1350 LaunchPad → TI Drivers → *rfPacketTx* → TI-RTOS → CCS Compiler → *rfPacketTx*

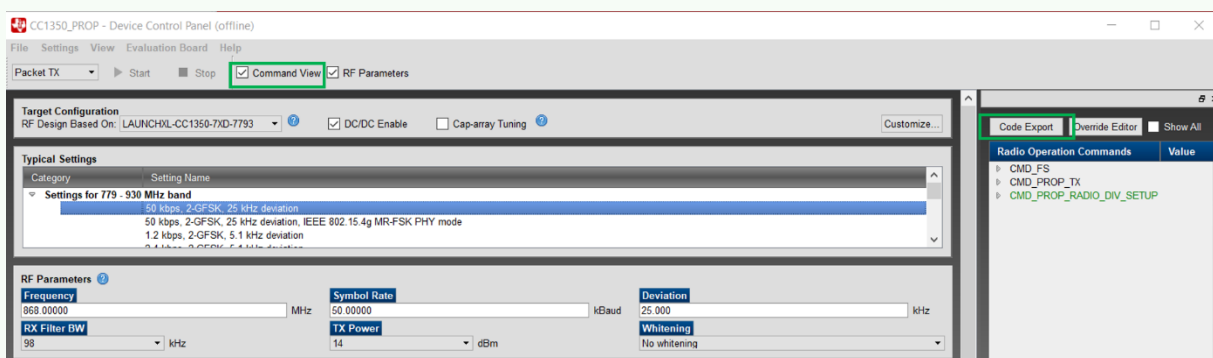
A l'aide du Readme, on remarque que la fréquence sélectionnée par défaut pour le LaunchPad CC1350 est 433.92 MHz, on doit changer cette fréquence et choisir 868 Mhz.

Etant donné que le fichier SysConfig n'est pas présent dans le menu de Project Explorer, il est préférable d'utiliser SmartRF Studio pour configurer le LaunchPad et importer les paramètres en code C.

Fermez la fenêtre de SmartRF Studio si elle est ouverte. Si le LaunchPad CC1350 est déjà connecté avec CCS, il peut y avoir occasionnellement des erreurs, il est donc nécessaire d'ouvrir une nouvelle fenêtre SmartRF Studio en mode offline. Sélectionnez le bouton rouge du CC1350 dans le fenêtre d'accueil de SmartRF Studio, puis cliquez sur « Open RF Device in Offline Mode »

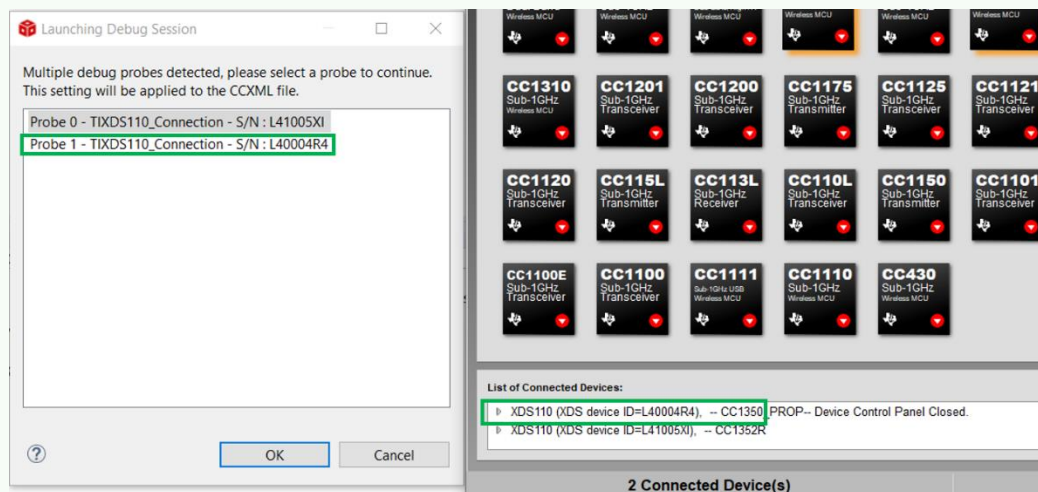


Vérifiez les RF paramètres, afin d'obtenir les mêmes paramètres que ceux présentés précédemment. Cliquez sur l'onglet « Commande View » puis sur « Code Export ».



A présent une nouvelle fenêtre apparaît contenant un fichier .c et un fichier .h représentant les informations de réseaux paramétrées précédemment.

Retournez dans CCS puis cliquez droit sur le nom du projet puis sur « Properties » puis dans l'onglet « Ressource », se trouve l'attribut « Location » correspondant au chemin où se situe les fichiers de configuration de réseaux, supprimez ces fichiers (« .c » et « .h ») et sauvegardez le chemin de la location. Enfin retournez dans la fenêtre de SmartRF Studio, celle où sont apparues les fichiers « .c » et « .h » et sauvegardez ces deux fichiers à l'emplacement des anciens grâce à File → Save Smartrf_setting... As. Désormais, les paramètres réseaux du projet sont modifiés. Compilez le projet dans le CC1350 et lancez le programme. Il est nécessaire de sélectionner le bon Probe si une fenêtre d'erreur apparaît, il est possible de vérifier cette information avec SmartRF Studio.

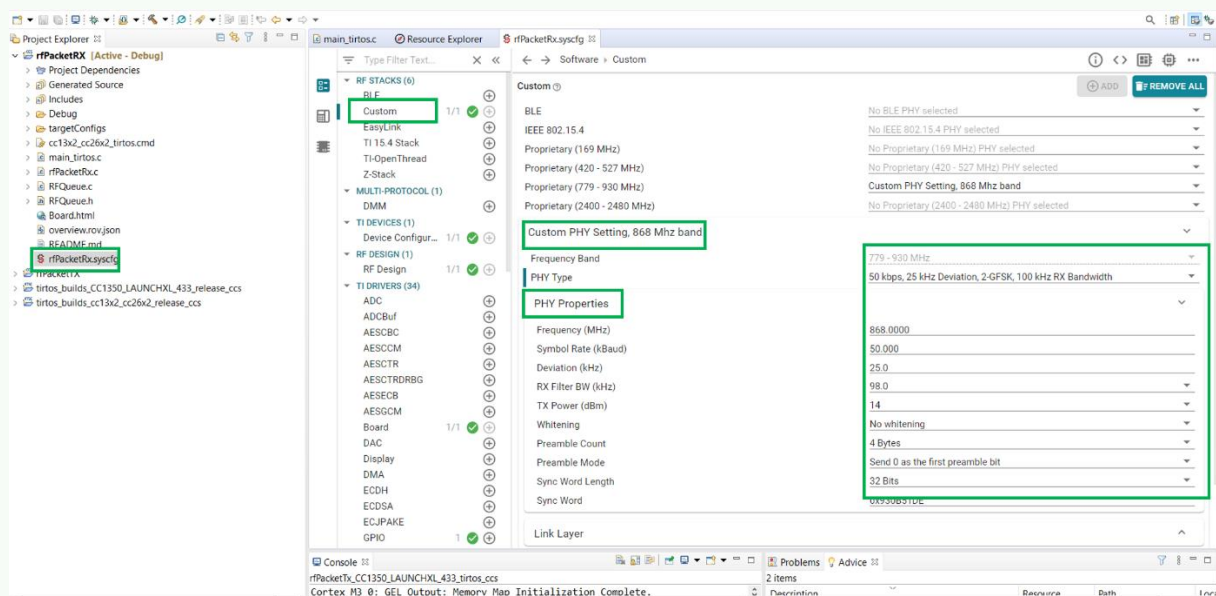


Si le CC1352 est toujours connecté à SmartRF Studio, il est possible de cliquer sur « Start » et visualiser les paquets reçus.

C. Configuration *rfPacketRX*

Pour le second LaunchPad (CC1352), importez l'exemple *rfPacketTX* depuis le Ressource Explorateur : Software → SimpleLink CC13x2... → Exemples → Development Tools → CC1352R LaunchPad → TI Drivers → *rfPacketRx* → TI-RTOS → CCS Compiler → *rfPacketRx*

Le fichier *SysConfig* est présent dans ce projet, il est donc possible de configurer les paramètres sans passer par SmartRF Studio. Cliquez sur *rfPacketRx.syscfg* (dans Project Explorer), puis dans la nouvelle fenêtre : Custom → Custom PHY Setting... → PHY Properties et vérifiez que les informations de réseaux coïncident avec celles du CC1350.



Désormais compilez le projet en choisissant le bon probe (vérifiez que le CC1352 est déconnecté de SmartRF Studio), puis lancez le programme.

Si tout à bien été configuré et que les deux programmes sont en fonctionnement, la LED verte du LaunchPad CC1350 s'allume quand un paquet est envoyé et la LED rouge du LaunchPad CC1352 s'allume quand un paquet est reçu.

II. API EasyLink

Le TI-RTOS RF Driver est le driver est utilisé pour toutes interaction avec le RF cœur. L'API EasyLink est une couche d'abstraction au-dessus du RF Driver permettant aux développeurs de créer un protocole propriétaire ou des applications utilisant la bande de fréquence Sub1-GHz (en dessous de 1 GHz) ou 2.4 GHz. (Cf : Annexe, connaître l'ensemble des APIs EasyLink).

Par ailleurs, il est possible de ne pas utiliser l'API EasyLink pour ces types d'application, par exemple dans les programmes de la partie précédente seulement le Ti-RTOS RF Driver est utilisé.

III. AT commandes

Pour ce tutoriel on utilise deux LaunchPads. Le LaunchPad CC1350 est utilisé pour l'émission de paquets (TX) et le LaunchPad CC1352 est utilisé pour la réception de paquets (RX).

Importez depuis le Ressource Explorer l'exemple rfEasyLinkNp Software → SimpleLink Academy → Exemples → Development Tools → CC1350 LaunchPad → EasyLink → rfEasyLinkNp → TI-RTOS → CCS Compiler → rfEasyLinkNp

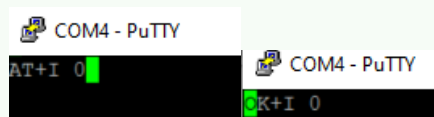
Modifiez les paramètres réseaux avec SmartRF Studio comme expliqué dans la partie précédente afin de positionner la fréquence à 868 MHz et écraser les fichiers « .c » et « .h » portant le nom smartf_settings, si d'autres fichiers sont présents il n'est pas nécessaire de les supprimer, compilez et lancez le programme. A l'aide du logiciel PuTTY visualisez la communication série (Baud : 115200), une fenêtre similaire devrait apparaître.



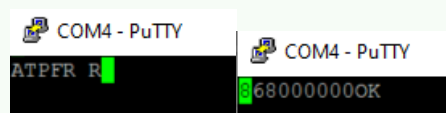
Il est possible d'effacer le contenu en positionnant le curseur à la fin de la phrase et d'appuyer sur la touche « retour arrière », puis sur « retour chariot ». En appuyant sur « entrée » le MCU comprend qu'il s'agit de la fin de la requête.

Tapez la commande suivantes pour choisir les bon paramètres (ceux contenu dans le fichier smartf_settings), si la requête est correctement formatée « OK » apparaît.

Attention : la touche retour arrière est comprise dans la requête car chaque caractère est envoyé un par un au programme.



Tapez la commande suivante pour vérifier si la fréquence est correcte, dans le cas contraire vérifiez le bon choix des paramètres avec SmartRF Studio ou en tapant la commande précédente. (En tapant AT+I 1, d'autres réglages sont choisis et la fréquence est désormais de 433 MHz).

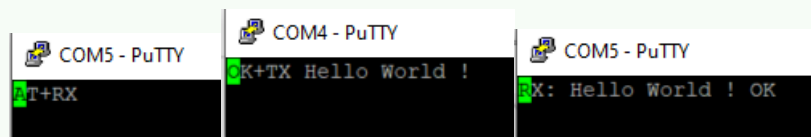


Il est possible de modifier cette fréquence avec la requête `AT+PFR=868000000`, toutefois de nombreuses erreurs apparaissent couramment.

Importez depuis le Ressource Explorer l'exemple `rfEasyLinkNp` : Software → SimpleLink CC13x2... → Exemples → Development Tools → CC1352R LaunchPad → EasyLink → `rfEasyLinkNp` → TI-RTOS → CCS Compiler → `rfEasyLinkNp`

Reproduisez les mêmes requêtes pour vérifier la configuration du CC1352, normalement il n'est pas nécessaire de modifier les paramètres de fréquences car elle est déjà à 868 MHz.

Positionnez le CC1352 en réception avec la commande `AT+RX` et envoyez un message avec le CC1350 comme illustré ci-dessous.



Il est également possible d'envoyer et de recevoir des messages à partir de SmartRF Studio. (Cf annexe : pour connaître les autres commandes AT)

IV. Réseau sans fils de capteur

Pour ce tutoriel on utilise deux LaunchPads. Le LaunchPad CC1350 est utilisé pour la réception de paquets (RX) et le LaunchPad CC1352 est utilisé pour l'émission de paquets (TX).

Importez depuis le Ressource Explorer l'exemple `rfWsnConcentrator` Software → SimpleLink CC13x0... → Exemples → Development Tools → CC1350 LaunchPad → EasyLink → `rfWsnConcentrator` → TI-RTOS → CCS Compiler → `rfWsnConcentrator`

Compilez ce programme, lancez-le sur le LaunchPad CC1350 et ouvrez PuTTY sur le port série correspondant au CC1350, il devrait apparaître le message « waiting nodes ».

Importez depuis le Ressource Explorer l'exemple `rfWsnNode` Software → SimpleLink CC13x2... → Exemples → Development Tools → CC1352R LaunchPad → EasyLink → `rfWsnNode` → TI-RTOS → CCS Compiler → `rfWsnNode`

Compilez ce programme, lancez-le sur le LaunchPad CC1352 et ouvrez PuTTY sur le port série correspondant au CC1352. Les informations visibles sur les deux émulateur devraient changer régulièrement. Dès que le CC1352 détecte un changement d'une certaine quantité relatif à la PIN DIO26 (ADC), un événement est actif et le CC1352 envoie au concentrateur la donnée mesurée. En rapprochant le doigt de cette broche, on remarque que la valeur mesurée diminue.



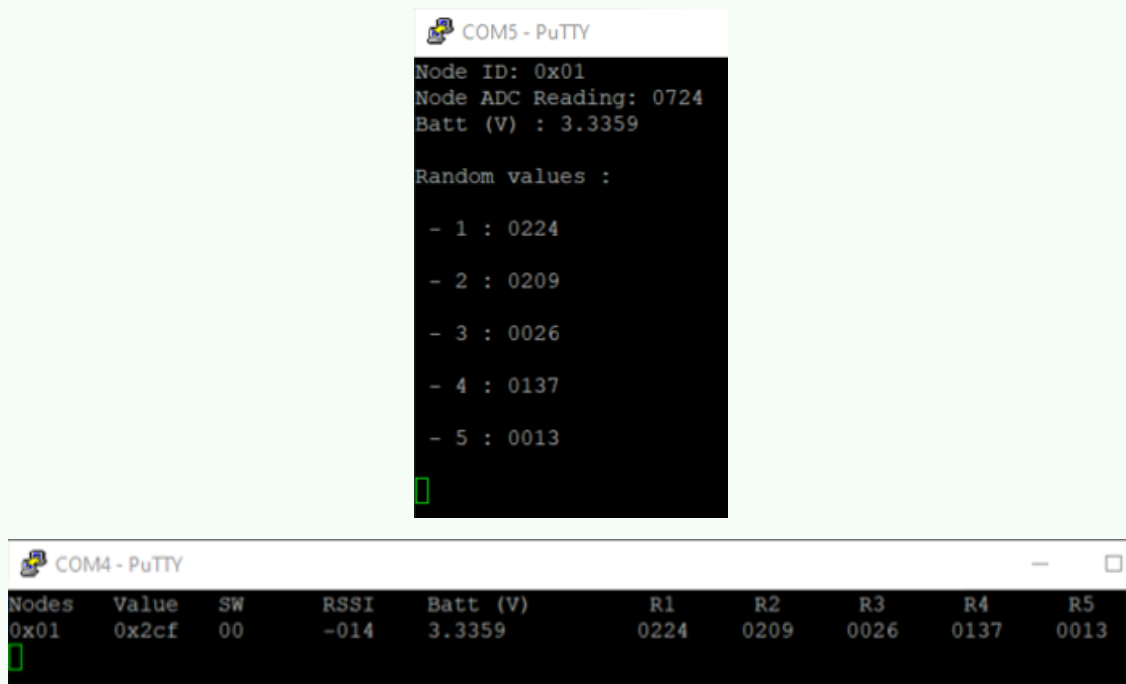
V. Exercices

A. Exercice 1 : Transfert de données

L'objectif de cet exercice est de comprendre l'exemple importé précédemment. Modifiez le programme afin d'envoyer des données aléatoires de l'émetteur au récepteur, et afficher ces données grâce à l'émulateur série.



Astuce : Pour obtenir la valeur de voltage de la batterie, il faut diviser l'entier obtenu par 256.



```

COM5 - PuTTY
Node ID: 0x01
Node ADC Reading: 0724
Batt (V) : 3.3359

Random values :

- 1 : 0224
- 2 : 0209
- 3 : 0026
- 4 : 0137
- 5 : 0013

COM4 - PuTTY
Nodes Value SW RSSI Batt (V) R1 R2 R3 R4 R5
0x01 0x2cf 00 -014 3.3359 0224 0209 0026 0137 0013

```

B. Exercice 2 : Mise en forme des données

Modifiez le programme précédent en y incorporant une fonction permettant de formater les données reçues par le concentrateur en utilisant l'API JSON (cf : annexe). Le résultat peut être affiché grâce à l'émulateur de série.



```

COM5 - PuTTY
Node ID: 0x01
Node ADC Reading: 0722
Batt (V) : 3.3359

Random values :

- 1 : 0210
- 2 : 0039
- 3 : 0018
- 4 : 0049
- 5 : 0169

COM4 - PuTTY
JSON :
{"nodeId" : 1 ,
  "adc" : 722 ,
  "sw" : 0 ,
  "rssi" : -14 ,
  "batt" : 3 ,
  "random" : [210 , 39 , 18 , 49 , 169]
}

```

Astuces :

- En plus d'importer le fichier d'entête, il est nécessaire d'ajouter la bibliothèque JSON au linker.

Clic droit sur le nom du projet -> Properties -> Build -> Arm Linker -> File Search Option -> add -> Browse. Puis dans SimpleLink CC13x0 sdk -> source -> ti -> utils -> json -> lib -> ccs -> m4f -> json_release.a

- La version de l'API JSON utilisée dans cet exemple ne permet pas de gérer les nombres décimaux.



VI. Annexe

A. Ressources

1. RF User's Guide

https://software-dl.ti.com/simplelink/esd/simplelink_cc13x2_26x2_sdk/3.30.00.03/exports/docs/proprietary-rf/proprietary-rf-users-guide/proprietary-rf-guide/index-cc13x2_26x2.html

2. CC1350 Technical Reference Manual

https://dev.ti.com/tirex/explore/node?devtools=LAUNCHXL-CC1350&node=AB2J8A8QJsHG2yuI9-dDEg_coGQ502_LATEST

3. CC1350 Datasheet

https://dev.ti.com/tirex/explore/node?devtools=LAUNCHXL-CC1350&node=ALVtnkqusYSLJ2kRV-la2w_coGQ502_LATEST

4. CC1352 Technical Reference Manual

https://dev.ti.com/tirex/explore/node?devtools=LAUNCHXL-CC1352R1&node=AFAIDSad-dUrZQZxqwVljg_coGQ502_LATEST

5. CC1352 Datasheet

https://dev.ti.com/tirex/explore/node?devtools=LAUNCHXL-CC1352R1&node=ALzDah9okU.vN18I0UTg6A_coGQ502_LATEST

6. RF Driver APIs

https://dev.ti.com/tirex/explore/node?node=AOng5xFsavzvQ16.KytQHg_eCfARaV_LATEST

7. Kernel APIs

https://dev.ti.com/tirex/explore/node?node=ALK7.RCEsk55Rp9wysRkGw_eCfARaV_LATEST

8. AT commandes

http://software-dl.ti.com/simplelink/esd/simplelink_cc13x0_sdk/2.10.00.36/exports/docs/proprietary-rf/proprietary-rf-users-guide/proprietary-rf-guide/rfEasyLinkNetworkProcessor_README.html

9. JSON API

https://dev.ti.com/tirex/content/simplelink_msp432e4_sdk_2_20_00_20/docs/ns/html/group_ti_utils_json_JSON.html

