

Ecole Publique d'Ingénieurs en 3 ans

Rapport

TUTO LORAWAN SUR STM32WL55

le 26 septembre 2022,
version 1.1

el-maazouzi,
Etudiant
[mohcine.el-
maazouzi@ensicaen.fr](mailto:mohcine.el-maazouzi@ensicaen.fr)

Tuteur école : Prénom NOM
Tuteur entreprise : Prénom
NOM



www.ensicaen.fr

TABLE DES MATIÈRES

TITRE PARTIE / CHAPITRE	3
1. Création de projet LoRaWan End Node	3
2. Configuration de loRaWAN et les périphériques	7
3. Compilation de projet	14

TABLE DES FIGURES

Figure 1 Légende de l'image	5
-----------------------------	---

TABLE DES TABLEAUX

Tableau 1 tableau	3
-------------------	---

TITRE PARTIE / CHAPITRE

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.


1. Création de projet LoRaWan End Node

En premier, il faut l'environnement de développement STM32cubeIDE et le firmware de la carte STM32WL55 vous pouvez le télécharger de ce lien :

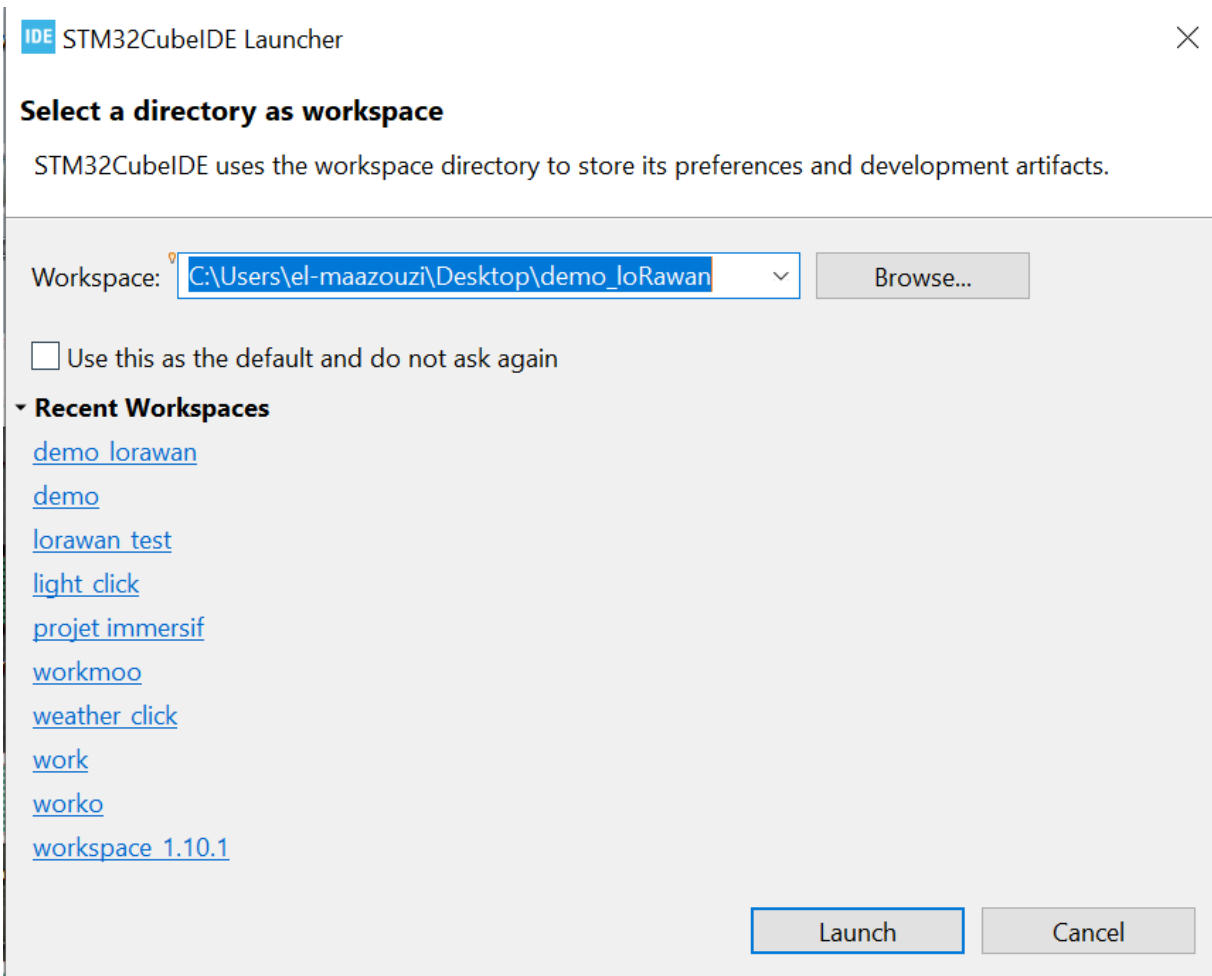
https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-mcu-mpu-packages/stm32cubewl.html

The screenshot shows the product page for STM32CubeWL. At the top left, it says "STM32CubeWL" with a green "ACTIVE" status. At the top right, there is a "Save to MyST" button. The main heading is "STM32Cube MCU Package for STM32WL series (HAL, Low-Layer APIs and CMSIS, File system, RTOS, KMS, Secure Engine, Sub-GHz Phy, LoRaWAN and Sigfox stacks - and examples running on ST boards)". Below the heading, there are two buttons: "Get Software" (highlighted with a red box) and "Download databrief" (with a document icon).

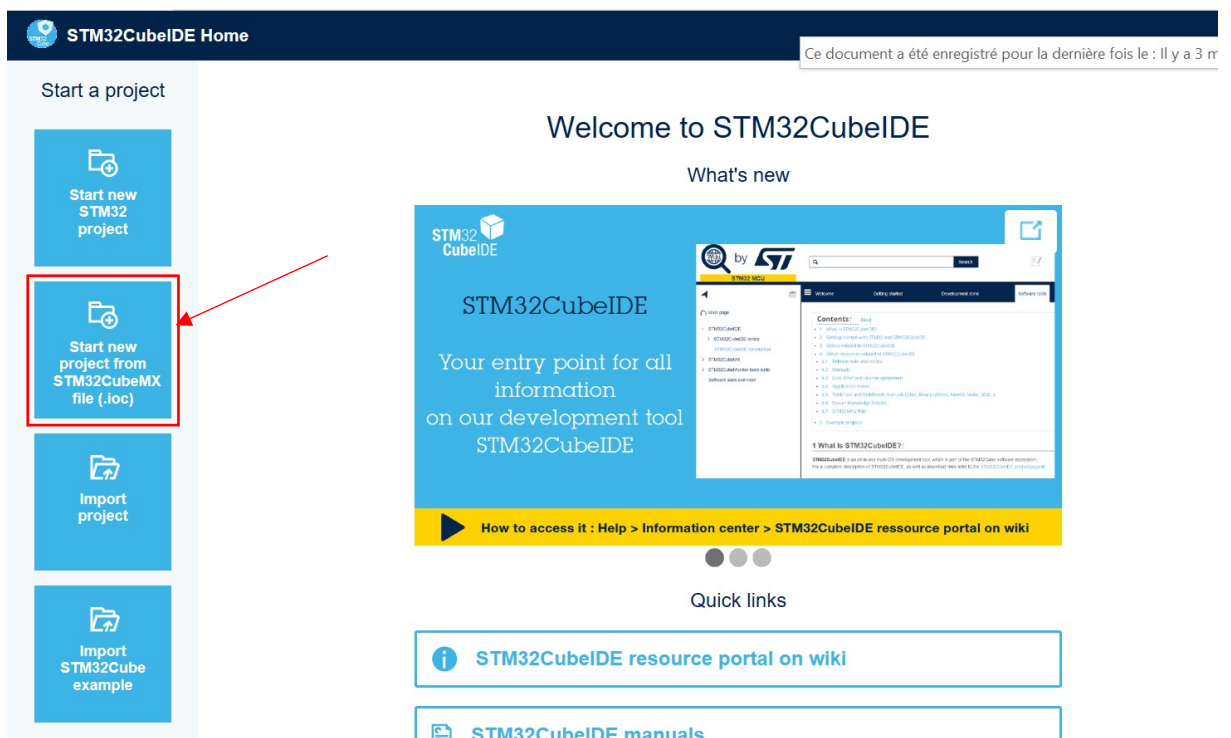
Après le téléchargement, extraire le fichier zip dans le répertoire de travail .

 STM32Cube_FW_WL_V1.2.0	12/09/2022 16:16	Dossier de fichiers
--	------------------	---------------------

Lancer le logiciel STM32cubeIDE et créer un nouveau workspace

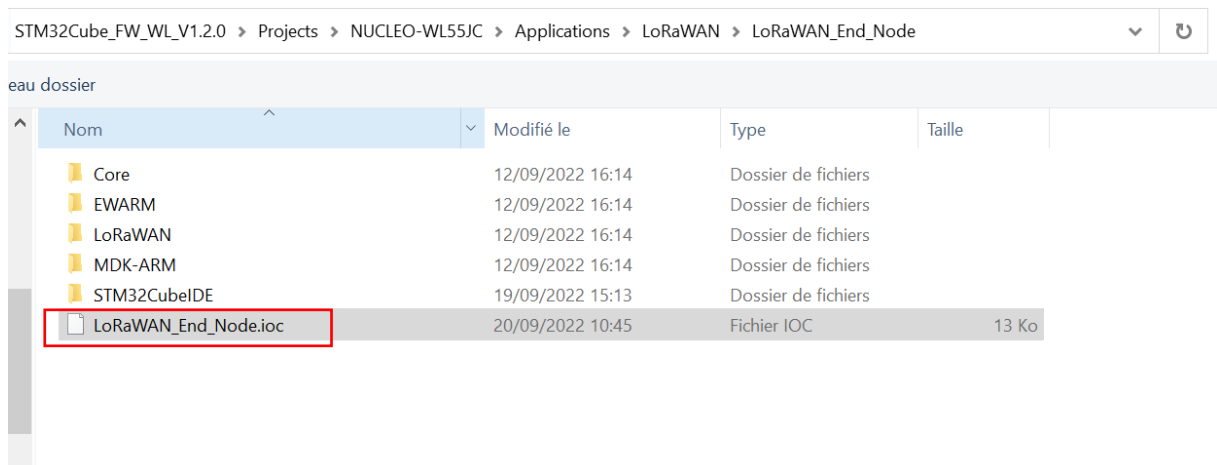
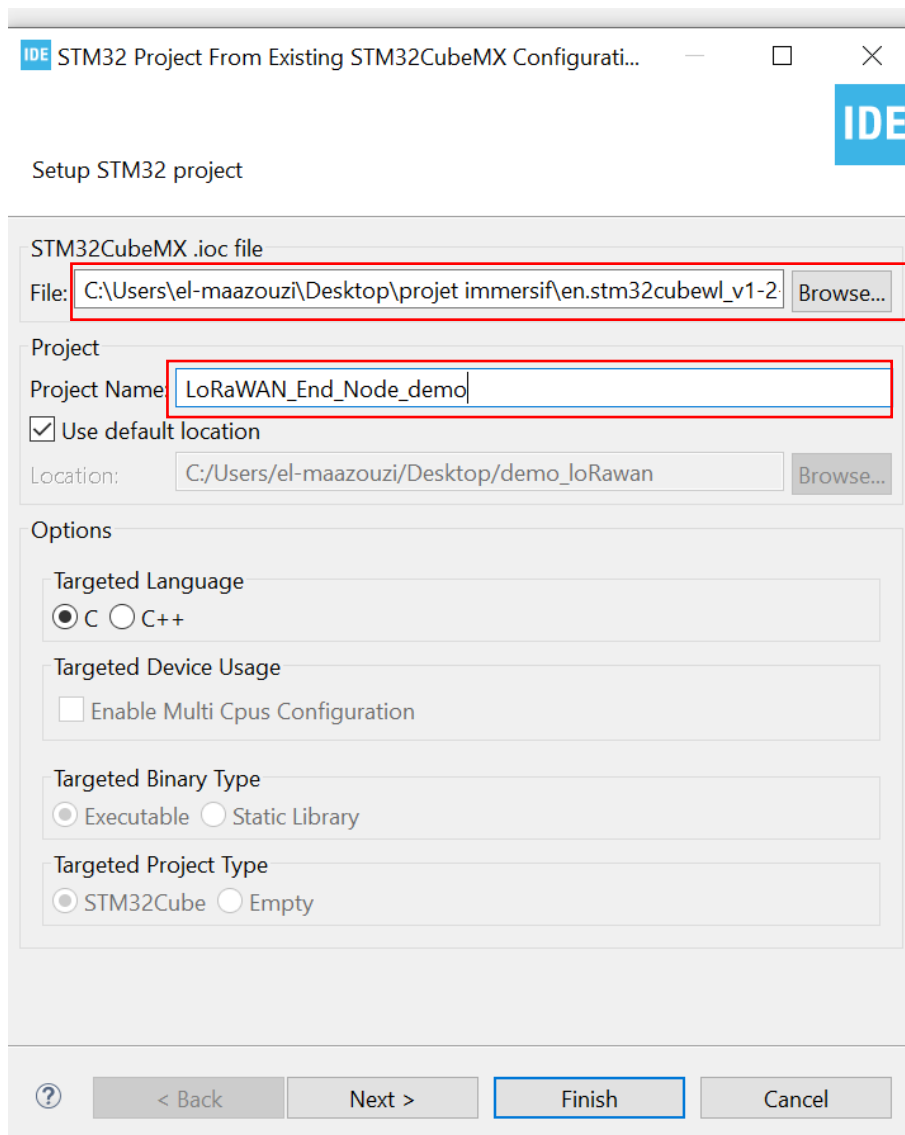


Créer un nouveau projet mais en choisissant l'option de créer un projet à partir d'un projet existant



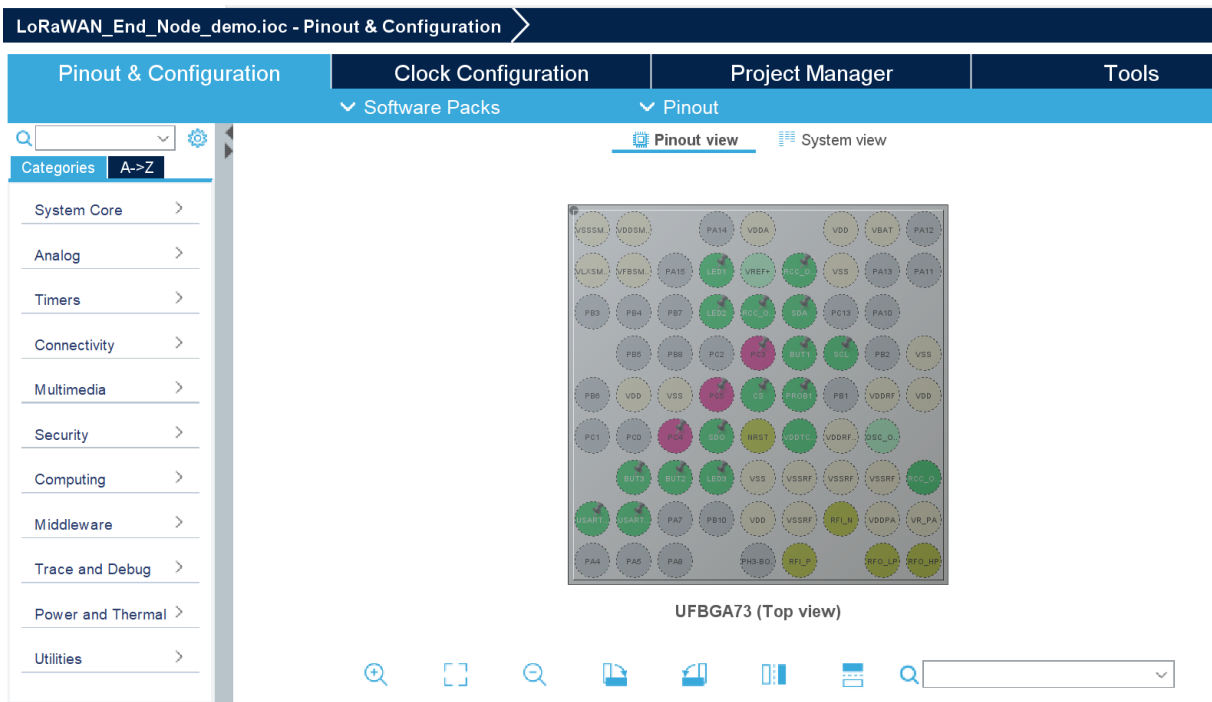
On choisit le projet LoRaWAN_End_Node dans le répertoire :

STM32Cube_FW_WL_V1.2.0\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\LoRaWAN_End_Node

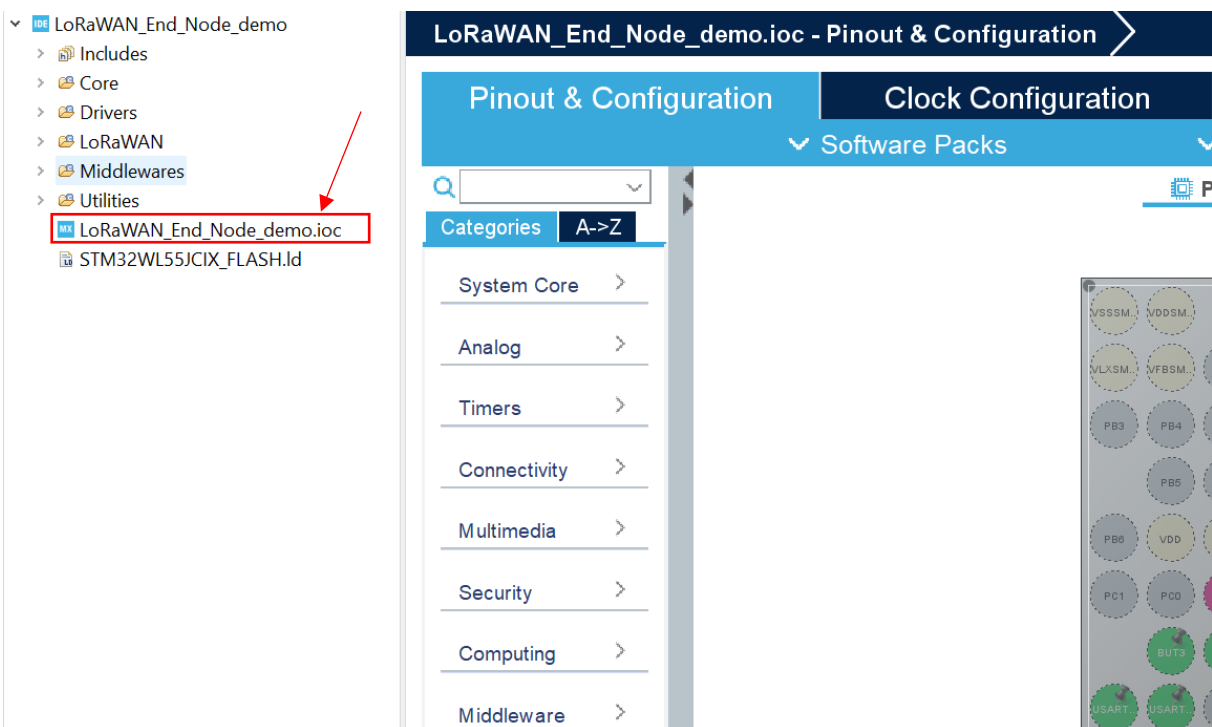


Après on renomme le projet et en clique sur Finish

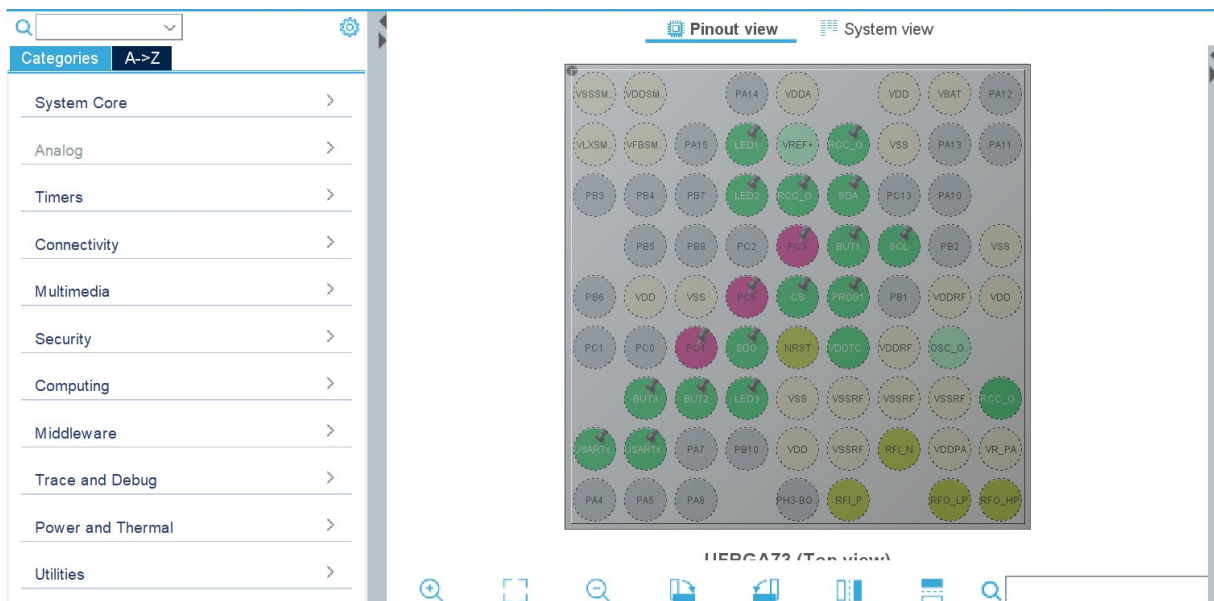
Normalement on aura la fenêtre de configuration ci-dessous qui s'ouvre ci-dessous



Si ce n'était pas le cas ,vous pouvez cliquer sur le fichier .ioc pour la réouvrir .

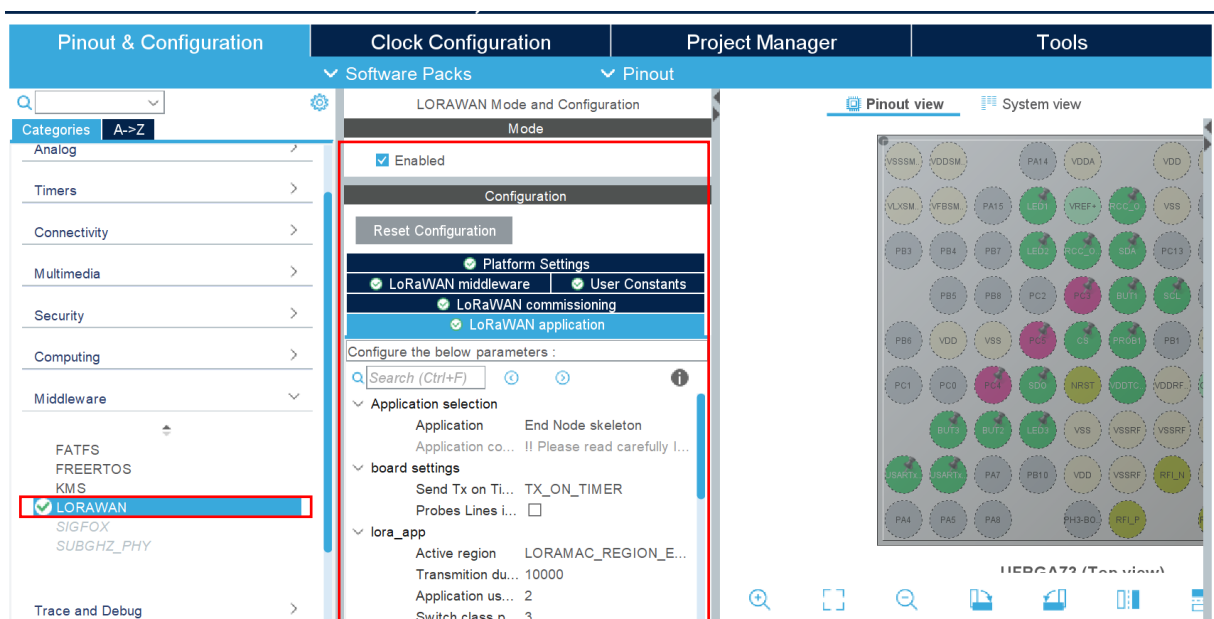


2. Configuration de LoRaWAN et les périphériques



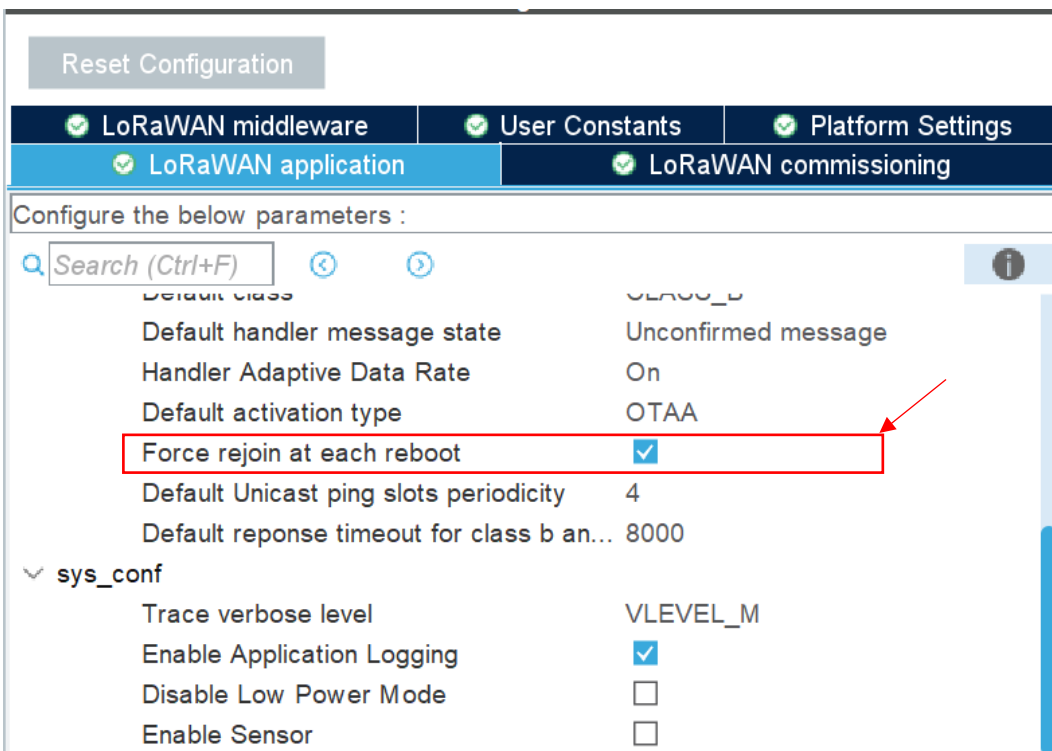
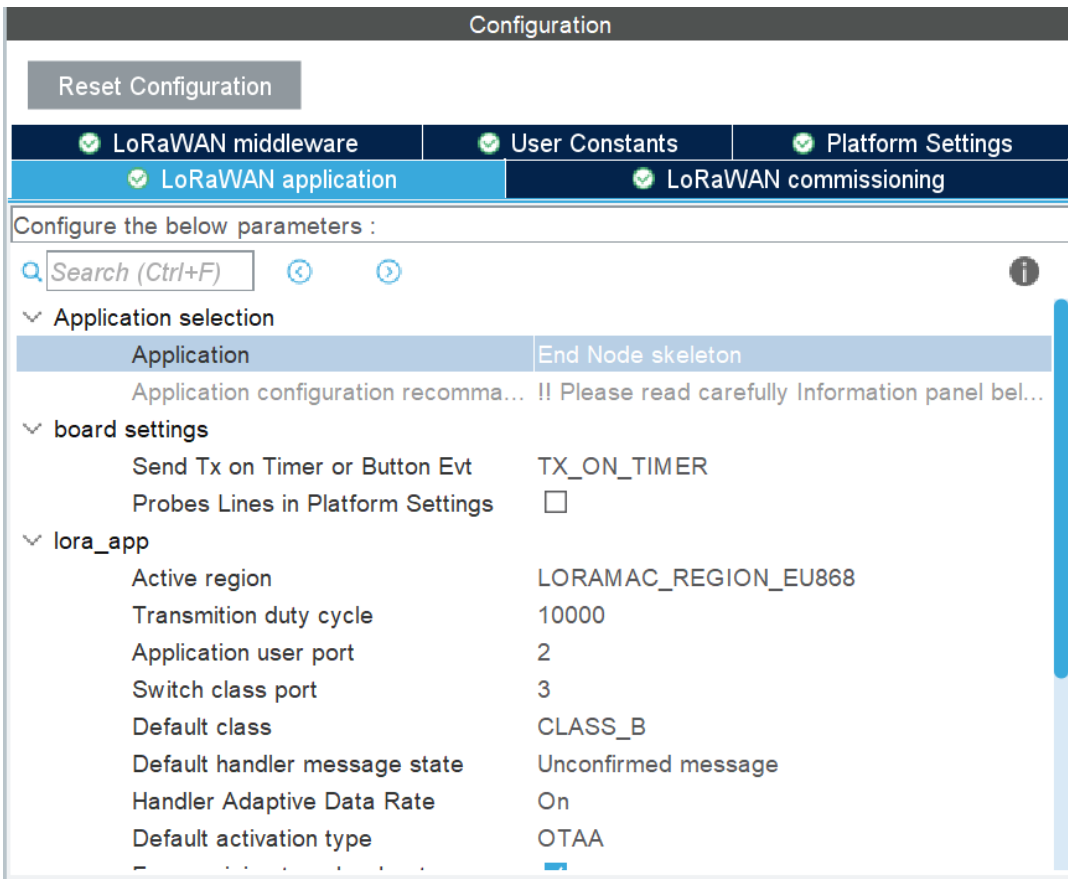
Dans la capture en dessus on voit une partie gauche qui sert à configurer plusieurs fonctionnalités ,et à droite on associe les broches avec les fonctions voulus.

Pour configurer le loRaWAN il faut aller dans middleware et après choisir lorawan.



Quand on appuie sur LORAWAN ,on aura un menu qui s'ouvre à coté qui permet de configuer les différentes couches de loRaWAN .

L'exemple de configuration de La couche application :



Pour le paramètre FORCE REJOIN AT EACH REBOOT ,par défaut c'est décocher et au début de notre développement on l'a gardé décocher .après on a trouvé que les cartes se connectent rapidement (au bout de 10s-15s) la première fois avec la Gateway lorawan ,


```

APPLICATION VERSION: V1.2.0
MH_LORAWAN_VERSION: V2.4.0
MH_RADIO_VERSION: V1.2.0
L2_SPEC_VERSION: V1.0.3
RP_SPEC_VERSION: V1-1.0.3
##### OTAA #####
##### AppKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3D
##### NwkKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3D
##### ABP #####
##### AppSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3D
##### NwkSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3D
##### IDs #####
##### DevEui: 00:80:E1:15:00:0A:05:73
##### AppEui: 01:01:01:01:01:01:01:06
##### DevAddr: 00:0A:05:73
0s041:TX on freq 868500000 Hz at DR 0
1s528:MAC txDone
6s561:RX 1 on freq 868500000 Hz at DR 0
8s369:MAC rxDone

##### = JOINED = OTAA =====
##### MCRotKey: 97:8E:4F:8F:5A:59:C2:58:AA:01:22:1F:8B:91:9D:85
##### MCKEKey: 65:EC:62:5D:6F:7A:51:94:F6:2E:C1:64:BB:BC:57:E3
##### NwkSKey: 36:73:C9:BE:F9:8D:04:36:43:89:C4:AF:45:01:F1:2D
##### AppSKey: 65:1D:A3:D1:87:CE:FD:03:05:A7:DA:CD:91:2E:04:FF
##### D/L FRAME:0000 | PORT:0 | DR:0 | SLOT:1 | RSSI:-91 | SNR:6
10s046:VDDA: 254
10s046:temp: 23
10s201:TX on freq 868500000 Hz at DR 0
10s205:SEND REQUEST
11s687:MAC txDone

```

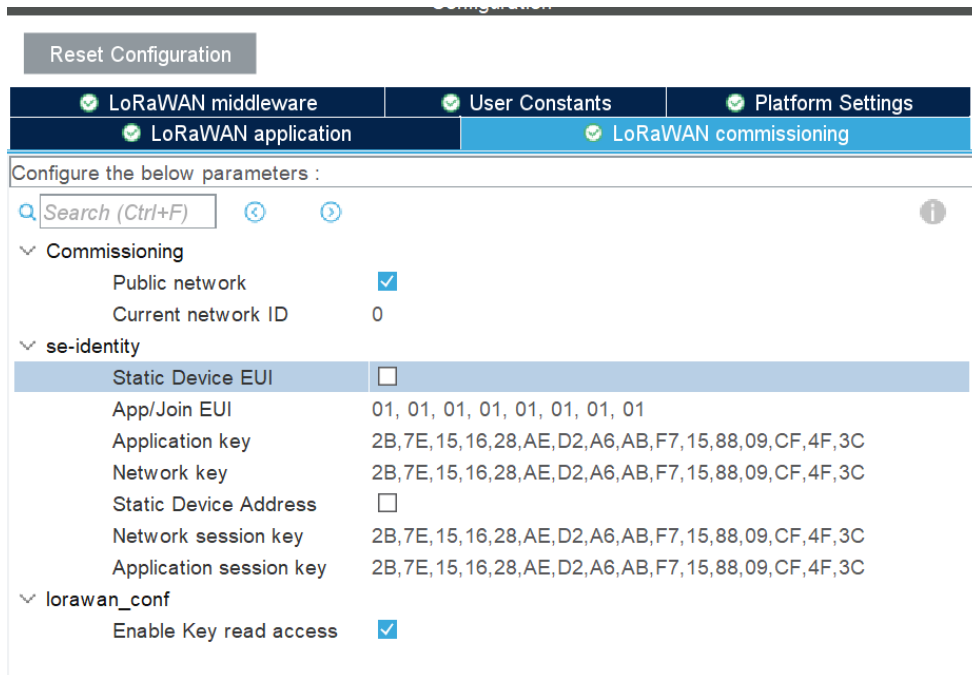
mais à chaque reconnexion ca prend plus de temps ,parfois on n'arrive jamais à se connecter, et **donc la solution trouvée** c'est d'activer FORCE REJOIN AT EACH REBOOT en la cochant sur l'interface ou la changer dans le code de False à True (il se trouve à lora_app.h) .

```

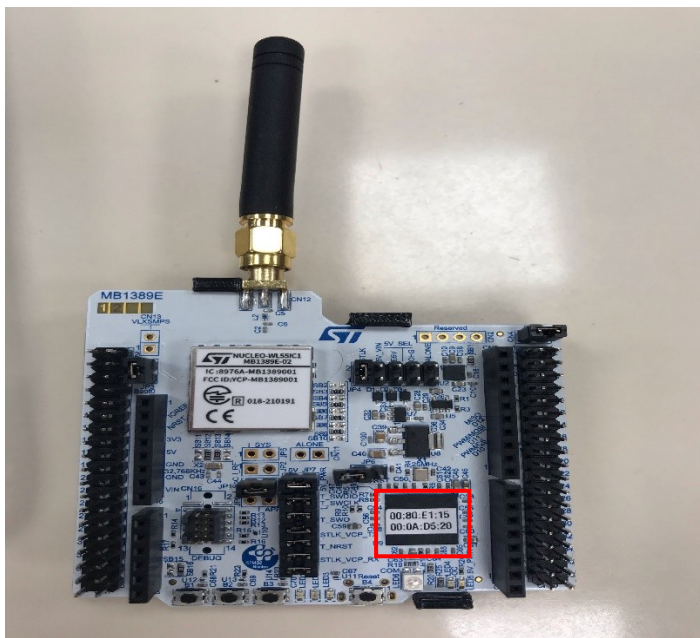
94 //
95 #define LORAWAN_DEFAULT_ACTIVATION_TYPE          ACTIVATION_TYPE_OTAA
96
97 /*!
98 * LoRaWAN force rejoin even if the NVM context is restored
99 * @note useful only when context management is enabled by CONTEXT_MANAGEMENT_ENABLED
100 */
101 #define LORAWAN_FORCE_REJOIN_AT_BOOT            true
102
103 /*!
104 * User application data buffer size
105 */
106 #define LORAWAN_APP_DATA_BUFFER_MAX_SIZE      242
107
108 /*!

```

L'exemple de configuration la couche LoRaWAN commissioning (la couche de cryptage et l'identification de device)



Le Static Device EUI c'est l'adresse MAC de la carte (voir la photo savoir où ça se trouve sur la carte), si vous la laissez comme par défaut le logiciel va mettre automatique l'adresse MAC de la carte branché pendant la compilation de projet. Si vous la cochez il va vous proposer de le rentrer manuellement.

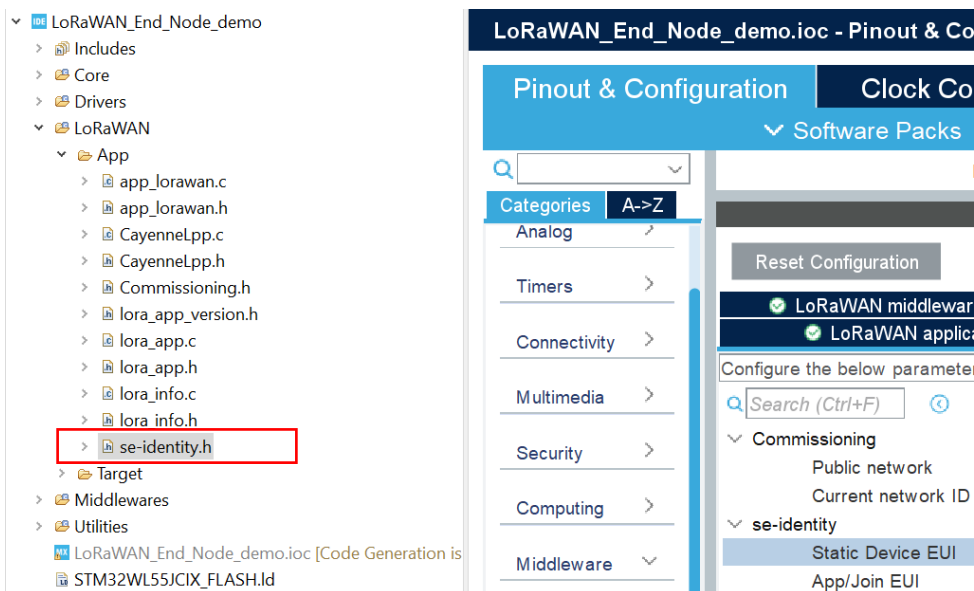


Le APP/JOIN EUI on peut le changer si on a plusieurs device mais il sert plutôt à les identifier.

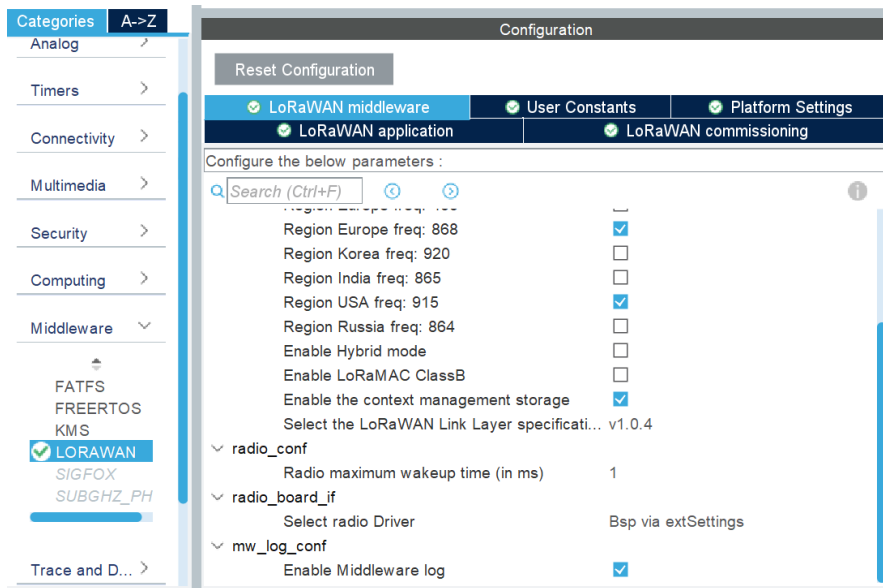
Le Application key et le network key c'est les clés de chiffrement, donc s'ils ne sont pas les mêmes utilisés sur the things network, vous aurez un message d'erreur 'MIC mismatch'

```
Live data See all activity →  
↑ 15:58:46 Join-request to cluster-local Join Server failed MIC mismatch  
↑ 15:57:35 Join-request to cluster-local Join Server failed MIC mismatch  
↑ 15:56:27 Join-request to cluster-local Join Server failed MIC mismatch  
+ 15:55:30 Create end device  
+ 15:55:30 Create end device  
+ 15:55:30 Create end device
```

Pour apporter des modifications sur LoRaWAN commissioning sans régénérer le code et le fichier .ioc (ceci prend un peu plus de temps et normalement on le fait qu'une fois) on peut changer tout à partir de code, pour LoRaWAN commissioning on le trouve dans le fichier header « se-identity.h ».

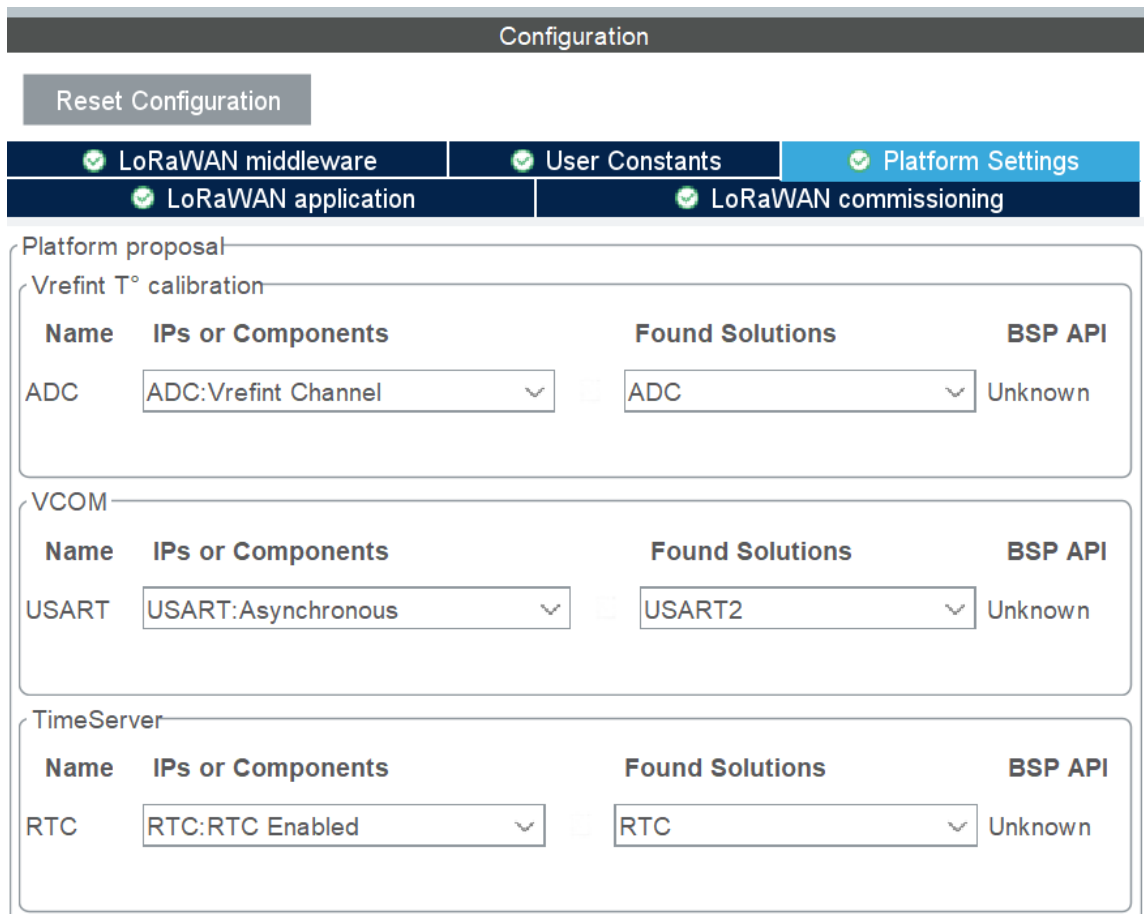


Exemple de configuration de loRaWAN middlewar, en général il n'y a rien à changer, on définit la région et c'est bon.

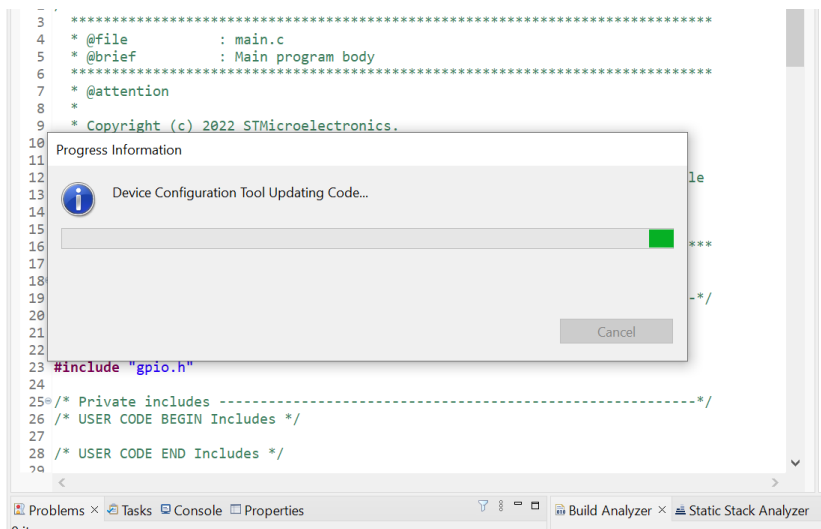
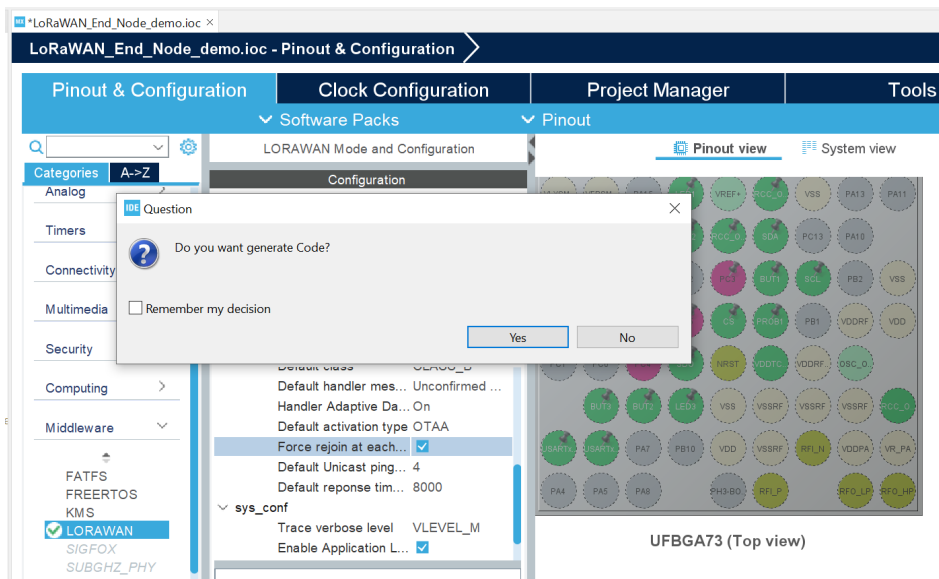


Les User constants, on l'a pas utilisé.

Exemple de Platform settings, si les probes lines était configuré (ceci est dans le lorawan application board settings) il va vous proposer de choisir des broches pour les avoir acces.



Après configurer le loRAWAN et les autres périphériques (SPI,uart,I2C,ADC)on peut enregistrer le fichier .ioc (raccourci (Ctrl +s))ou faire un build de projet .



Ensuite le IDE va générer le code suite à la configuration choisie ou il va faire une mise à jour si ce n'était pas la première fois ,dans ce cas il faut faire attention ,car à chaque fois vous régénérez le code il va effacer tous les modification apportés. Pour éviter ceci il faut écrire votre code (les Macros ,les includes, et code USER) entre les commentaires générés par IDE qui commence par USER CODE BEGIN et finit par USER CODE END .

```

/* Private includes -----
/* USER CODE BEGIN Includes */
int declarer_vos_includes_ici ;
/* USER CODE END Includes */

/* Private typedef -----
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----
/* USER CODE BEGIN PV */
int declarer_vos_variable_global_ici ;
/* USER CODE END PV */

```

3. Compilation de projet

Normalement, après le génération de projet ,si on essaye de faire un build de projet ou le compiler ,vous aurez une erreur en disant qu'il ne trouve pas des fichier voir capture en dessous.

```

In file included from ../LoRaWAN/Target/radio_conf.h:30,
                 from ../Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/radio_driver.c:34:
../Core/Inc/platform.h:45:10: fatal error: stm32wlxx_nucleo_radio.h: No such file or directory
 45 | #include "stm32wlxx_nucleo_radio.h"
    | ^~~~~~
In file included from ../LoRaWAN/Target/radio_conf.h:30,
                 from ../Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/radio.c:39:
../Core/Inc/platform.h:45:10: fatal error: stm32wlxx_nucleo_radio.h: No such file or directory
 45 | #include "stm32wlxx_nucleo_radio.h"
    | ^~~~~~
compilation terminated.
compilation terminated.
make: *** [Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/subdir.mk:25: Middlewares/Third_Party/SubGHz_P
make: *** Waiting for unfinished jobs....
make: *** [Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/subdir.mk:25: Middlewares/Third_Party/SubGHz_P
In file included from ../LoRaWAN/Target/radio_conf.h:30,
                 from ../Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/radio_fw.c:25:
../Core/Inc/platform.h:45:10: fatal error: stm32wlxx_nucleo_radio.h: No such file or directory
 45 | #include "stm32wlxx_nucleo_radio.h"
    | ^~~~~~
compilation terminated.
make: *** [Middlewares/Third_Party/SubGHz_Phys/stm32_radio_driver/subdir.mk:25: Middlewares/Third_Party/SubGHz_P
"make -j8 all" terminated with exit code 2. Build might be incomplete.

16:40:25 Build Failed. 7 errors. 0 warnings. (took 2s.542ms)

```

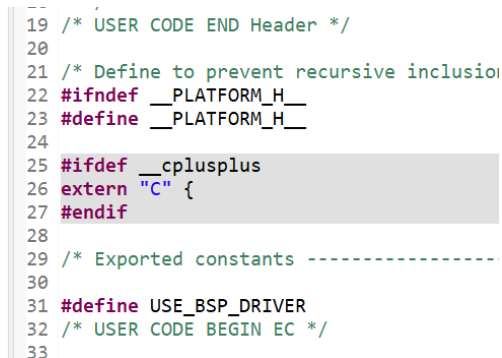
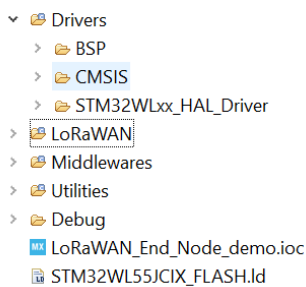
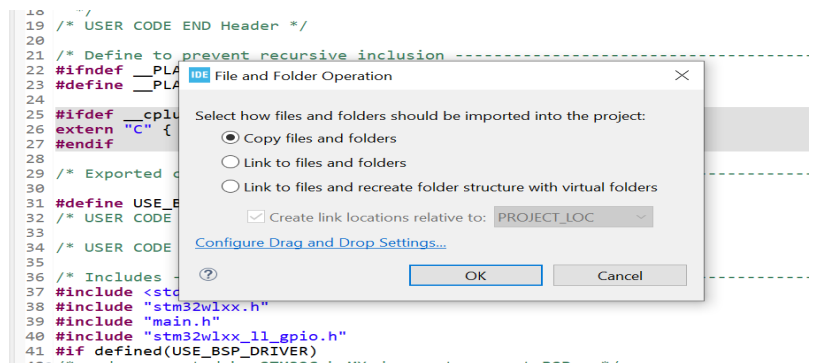
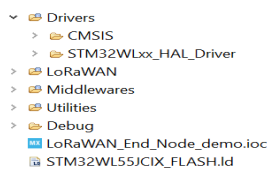
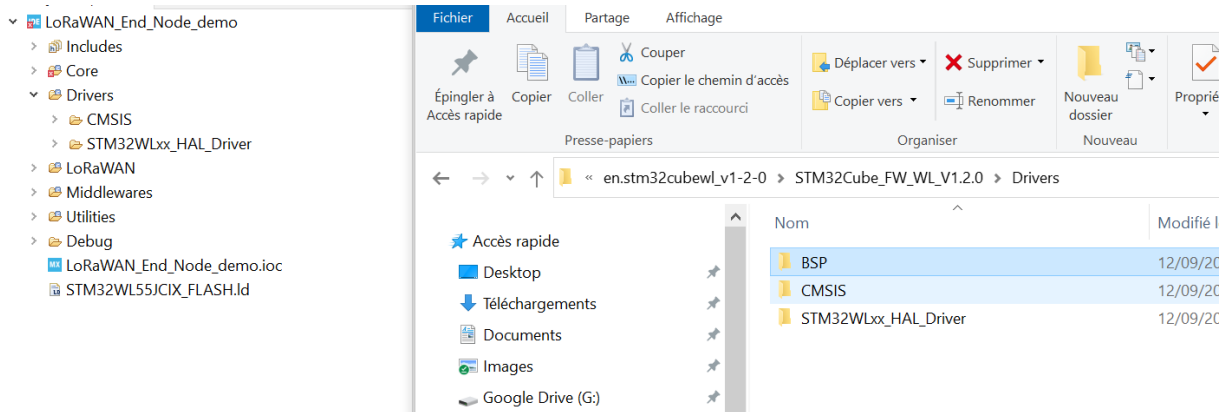
Ceci est normal car l'IDE n'ajoute pas le BSP (BOARD SUPPORT PACKAGE) automatiquement, c'est noté dans le fichier « platform.h »,donc il faut l'ajouter manuellement.

```

#if defined(USE_BSP_DRIVER)
/* code generated by STM32CubeMX does not support BSP. */
/* In order to use BSP, users can add the BSP files in the IDE project space
/* and define USE_BSP_DRIVER in the preprocessor definitions */
#include "stm32wlxx_nucleo_radio.h"
#include "stm32wlxx_nucleo.h" /* not used by this project*/
#endif /* defined(USE_BSP_DRIVER) */

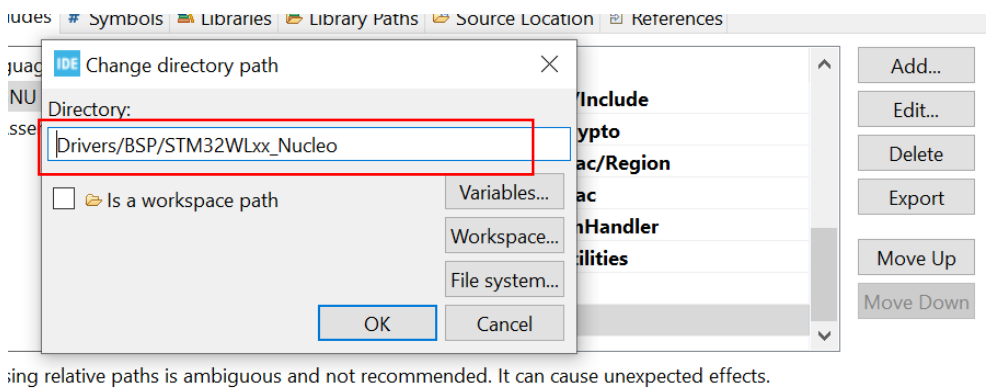
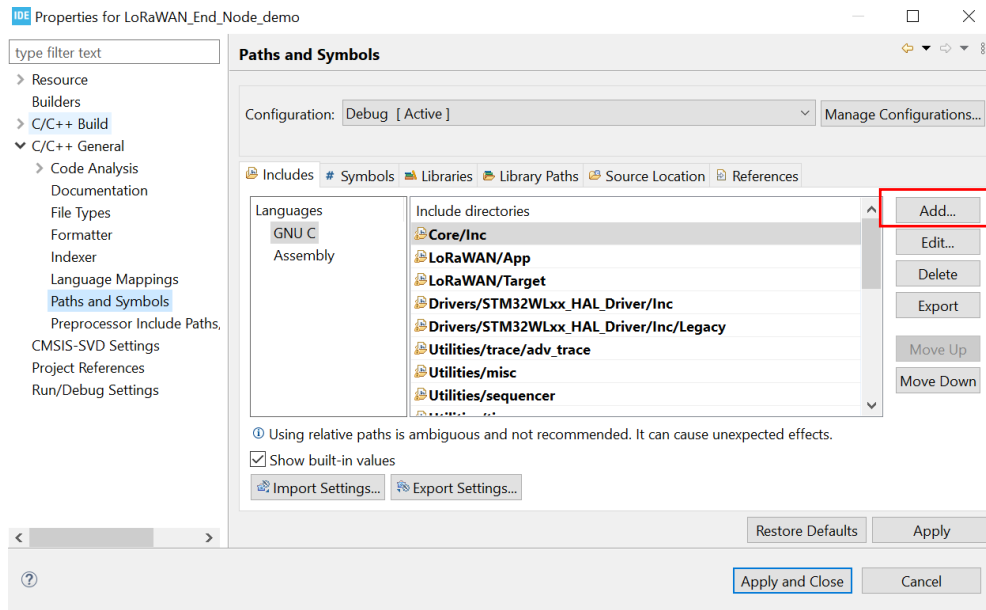
```

Pour le résoudre, il faut copier le dossier BSP qui se trouve dans le dossier de firmware STM32Cube_FW_WL_V1.2.0\Drivers vers le dossier Drivers de projet, donc on peut glisser le dossier BSP dans l'explorateur de fichiers vers le pavé des arborescences au répertoire Drivers.

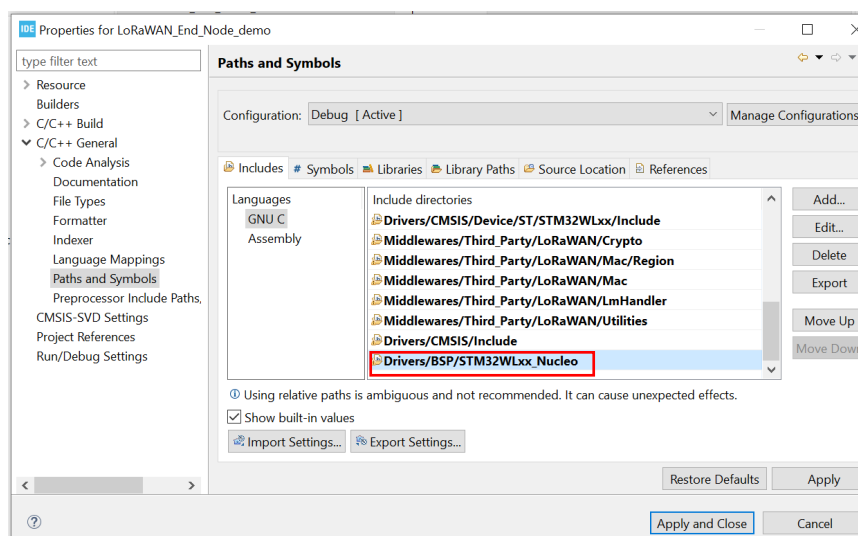


Mais ce n'est pas fini, car il faut indiquer au compilateur que vous avez ajouté les fichiers .C et .H .

Donc il faut aller aux propriétés de projet et ajouter l'arborescence Drivers/BSP/STM32WLxx_Nucleo dans les includes.



Using relative paths is ambiguous and not recommended. It can cause unexpected effects.

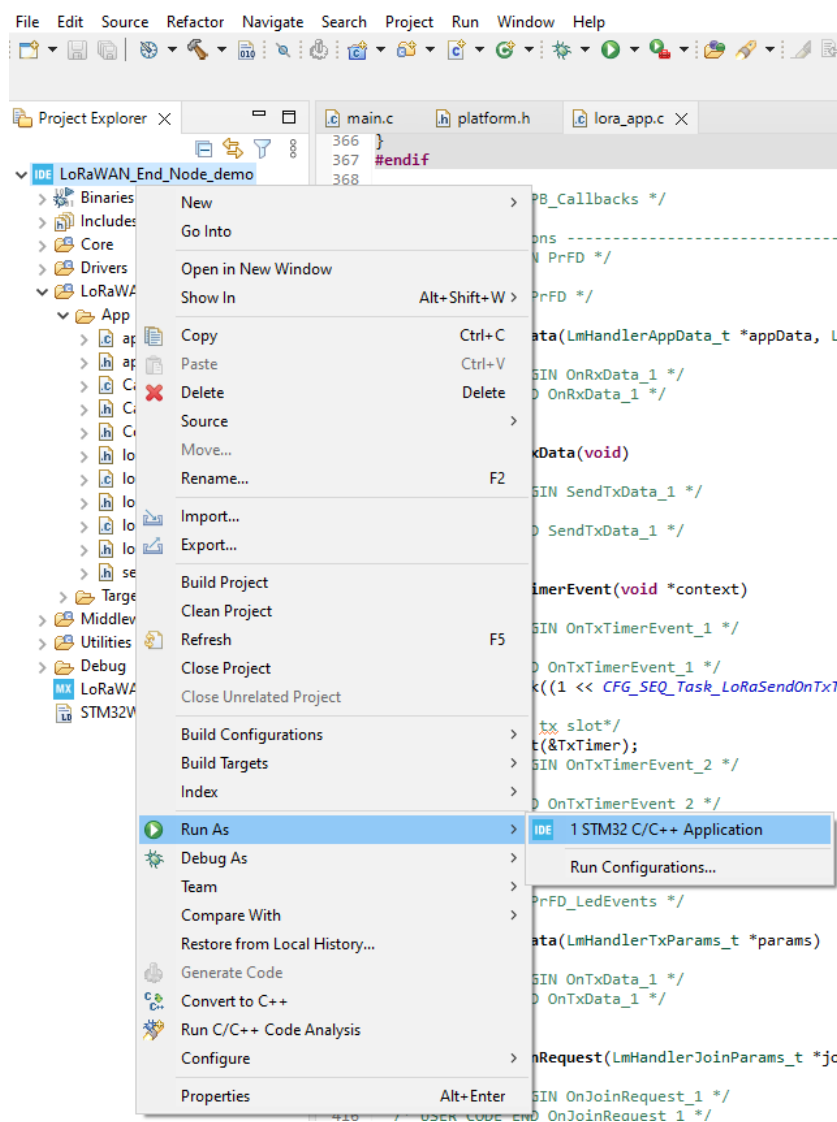


Après on recompile le projet et ça devrait marcher sans problème.

```
17:45:12 *** Incremental Build of configuration Debug for project LoRaWAN_End_Node_demo ***
make -j8 all
arm-none-eabi-size  LoRaWAN_End_Node_demo.elf
  text  data  bss  dec  hex filename
120236  320  10560  13116  2002c LoRaWAN_End_Node_demo.elf
Finished building: default.size.stdout

17:45:13 Build Finished. 0 errors, 0 warnings. (took 1s.408ms)
```

Ensuite on téléverse le programme vers la carte, pour faire ceci on appuie sur le bouton play vert ou en cliquant sur le projet (voir figure en dessous).



On peut utiliser un HyperTerminal par exemple Teraterm pour voir les informations et l'état de la communication LoRawan envoié par le microcontrôleur sur le périphérique UART, il faut configurer le débit de communication à 115200 bits/s.

```
##### OTAA #####
##### AppKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NwkKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### ABP #####
##### AppSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NwkSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### IDs #####
##### DevEui: 00:80:E1:15:00:0A:C7:8B
##### AppEui: 01:01:01:01:01:01:01:01
##### DevAddr: 00:0A:C7:8B
0s040:TX on freq 868300000 Hz at DR 0
1s525:MAC txDone
6s559:RX_1 on freq 868300000 Hz at DR 0
6s757:IRQ_RX_TX_TIMEOUT
6s757:MAC rxTimeOut
7s559:RX_2 on freq 869525000 Hz at DR 0
7s757:IRQ_RX_TX_TIMEOUT
7s757:MAC rxTimeOut
```

En dessus c'est ce qu'on reçoit sur Teraterm ,mais on remarque que le programme au bout de 7 second ,car si on voit le fichier « lora_app.c » on trouve que plusieurs définitions des fonctions sont vides(voir figure en dessous) ,par exemple la fonction « SendTxData(void) » c'est la fonction qui on envoié les données .



```

LoRaWAN_End_Node_demo
├── Binaries
├── Includes
├── Core
├── Drivers
├── LoRaWAN
│   └── App
│       ├── app_lorawan.c
│       ├── app_lorawan.h
│       ├── CayenneLpp.c
│       ├── CayenneLpp.h
│       ├── Commissioning.h
│       ├── lora_app_version.h
│       └── lora_app.c
│           ├── lora_app.h
│           ├── lora_info.c
│           ├── lora_info.h
│           └── se-identity.h

```

```

300 }
367 #endif
368
369 /* USER CODE END PB_Callsbacks */
370
371@ /* Private functions -----*/
372 /* USER CODE BEGIN PrFD */
373
374 /* USER CODE END PrFD */
375
376 static void OnRxData(LmHandlerAppData_t *appData, LmHandlerRxParams_t *params)
377 {
378     /* USER CODE BEGIN OnRxData_1 */
379     /* USER CODE END OnRxData_1 */
380 }
381
382 static void SendTxData(void)
383 {
384     /* USER CODE BEGIN SendTxData_1 */
385
386     /* USER CODE END SendTxData_1 */
387 }

```

Ce problème est dû au logiciel car quand on crée un projet à partir de projet d'exemple il ne copie que la structure. Donc il faut aller copier ou remplacer le fichier « lora_app.c » qui se trouve le projet actuel par celui de projet d'exemple LoRaWAN_End_Node(perso j'ai ouvert les deux fichiers lora_app.c de chaque

projet ,j'ai fait un Ctrl+A et un Ctrl+C de fichier lora_app.c de projet de exemple vers le projet courent avec un Ctrl+A et un Ctrl+V).

Extrait de code de la fonction SendTxData(void) après faire le copier coller.

```
537  /* USER CODE END OnRxData_1 */
538  }
539
540 static void SendTxData(void)
541 {
542  /* USER CODE BEGIN SendTxData_1 */
543  LmHandlerErrorStatus_t status = LORAMAC_HANDLER_ERROR;
544  uint8_t batteryLevel = GetBatteryLevel();
545  sensor_t sensor_data;
546  UTIL_TIMER_Time_t nextTxIn = 0;
547
548 #ifndef CAYENNE_LPP
549  uint8_t channel = 0;
550 #else
551  uint16_t pressure = 0;
552  int16_t temperature = 0;
553  uint16_t humidity = 0;
554  uint32_t i = 0;
555  int32_t latitude = 0;
556  int32_t longitude = 0;
557  uint16_t altitudeGps = 0;
558 #endif /* CAYENNE_LPP */
559
560  EnvSensors_Read(&sensor_data);
561
562  APP_LOG(TS_ON, VLEVEL_M, "VDDA: %d\r\n", batteryLevel);
563  APP_LOG(TS_ON, VLEVEL_M, "temp: %d\r\n", (int16_t)sensor_data.temperature);
564
565  AppData.Port = LORAWAN_USER_APP_PORT;
566
567 #ifndef CAYENNE_LPP
568  CayenneLppReset();
569  CayenneLppAddBarometricPressure(channel++, sensor_data.pressure);
570  CayenneLppAddTemperature(channel++, sensor_data.temperature);
571  CayenneLppAddRelativeHumidity(channel++, (uint16_t)sensor_data.humidity);
572
573  if ((LmHandlerParams.ActiveRegion != LORAMAC_REGION_US915) && (LmHandlerPar
574      && (LmHandlerParams.ActiveRegion != LORAMAC_REGION_AS923))
575      {
576      CayenneLppAddDigitalInput(channel++, GetBatteryLevel());
577      CayenneLppAddDigitalOutput(channel++, AppLedStateOn);
578      }
579
580  CayenneLppCopy(AppData.Buffer);
581  AppData.BufferSize = CayenneLppGetSize();
```

Après faire ceci on recompile et on fait un RUN et normalement le programme ne doit pas s'arrêter au bout de 7s ,mais on aura un join failed et si le on n'arrive pas à se connecter avec la Gateway après 120s le programme ne fait plus de join request il affiche les données récupérées.on y aura une led rouge clignote sur la carte tant qu'on n'arrive pas à se connecter.

Quand le device arrive à se connecter , la led rouge arrete de clignoter et on y aura la led blue qui s'allume et apres s'etteind ,ceci signifie que la connexion est bien etablie ,apres il y aura la led vert ou la led blue parfois qui clignote .

```
##### OTAA #####
##### AppKey:      2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NukKey:      2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### ABP #####
##### AppSKey:     2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NukSKey:     2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### IDs #####
##### DevEui:      00:80:E1:15:00:0A:C7:8B
##### AppEui:      01:01:01:01:01:01:01:02
##### DevAddr:     00:0A:C7:8B
0s039:TX on freq 868100000 Hz at DR 0
1s525:MAC txDone
6s558:RX_1 on freq 868100000 Hz at DR 0
8s366:MAC rxDone

##### = JOINED = OTAA =====
##### MCRoortKey:  7D:F7:6B:DC:1A:B8:99:B3:3E:42:FD:47:B9:1B:54:6F
##### MCKEKey:     8C:B8:66:5E:0C:0E:0B:64:5B:2E:D9:E4:8A:19:27:7C
##### NukSKey:     EA:A5:1E:1A:60:83:CD:CE:FB:88:6A:ED:7B:D2:80:0B
##### AppSKey:     22:D1:AB:17:10:66:52:BC:08:F1:A4:C9:A7:6D:0D:47
10s042:VDDA: 254
10s042:temp: 26
10s047:TX on freq 867900000 Hz at DR 0
10s050:SEND REQUEST
11s698:MAC txDone
16s731:RX_1 on freq 867900000 Hz at DR 0
18s046:MAC rxDone

##### ===== MCPS-Confirm =====
20s051:VDDA: 254
20s051:temp: 26
20s056:TX on freq 867500000 Hz at DR 4
20s059:SEND REQUEST
20s183:MAC txDone
25s167:RX_1 on freq 867500000 Hz at DR 4
25s218:IRQ_RX_TX_TIMEOUT
25s218:MAC rxTimeOut
26s171:RX_2 on freq 869525000 Hz at DR 3
26s231:IRQ_RX_TX_TIMEOUT
26s231:MAC rxTimeOut
```

4. Envoie des donnes de capteur

Lors de développement on a fait fonctionner les capteurs sur STM32WL55, en premier temps sans LoRaWan ,les 4 capteurs :weather_click, air_quality ,pollution_click et light_click utilisent des protocoles de communication différentes ,par exemple weather et air_quality utilisent I2C ,light utilise SPI et pollution donne une valeur analogique .en second temps on a essayé d'interger les capteurs dans le projet loRaWan et donc les capteurs qu'on a pu interger c'est weather_click et air_quality. Le light_click ne fonctionne pas car le microcontrôleur utilise le périphérique SPI pour communiquer avec le module loRaWan mais le light_click aussi utilise le même périphérique SPI ,du coup il y avait un conflit et les donnes reçues de light_click était fausses.

Pour injecter les donnes de capteurs il faut aller donc le fichier « lora_app.c » plus précisément la fonction SendTxData(void) .

```

AppData.BufferSize = CayenneLppGetSize();
#else /* not CAYENNE_LPP */
humidity    = (uint16_t)(sensor_data.humidity * 10);          /* in %*10 */
temperature = (int16_t)(sensor_data.temperature);
pressure    = (uint16_t)(sensor_data.pressure * 100 / 10); /* in hPa / 10 */

AppData.Buffer[i++] = AppLedStateOn;
AppData.Buffer[i++] = (uint8_t)((pressure >> 8) & 0xFF);
AppData.Buffer[i++] = (uint8_t)(pressure & 0xFF);
AppData.Buffer[i++] = (uint8_t)(temperature & 0xFF);
AppData.Buffer[i++] = (uint8_t)((humidity >> 8) & 0xFF);
AppData.Buffer[i++] = (uint8_t)(humidity & 0xFF);

if ((LmHandlerParams.ActiveRegion == LORAMAC_REGION_US915) || (LmHandlerParams.ActiveReg
    || (LmHandlerParams.ActiveRegion == LORAMAC_REGION_AS923))
{
    AppData.Buffer[i++] = 0;
    AppData.Buffer[i++] = 0;
    AppData.Buffer[i++] = 0;
    AppData.Buffer[i++] = 0;
}
else
}

```

Pour envoyer les données on a utilisé la structure AppData.Buffer déjà définie, ceci est uint_8 et donc il faut découper les données en paquet de 8 bits.

Puisqu'il avait plusieurs types de capteurs on a décidé que le premier paquet serait un identifiant car sur le things network il faut un formatage de données pour l'envoi par les webhooks sur thingspeak de mathworks et donc on définit une macro comme celle de l'exemple (voir figure en dessous) quand on active ou désactive en commentant la macro **#define THINGSPEAK** donc lora_app.h

```

-
9 AppData.BufferSize = CayenneLppGetSize();
0 #elif defined THINGSPEAK /* used THINGSPEAK ENSICAEN */
1
2 temperature = (int16_t) (BME280_GET_TEMPERATURE(CALIBRATION)*100);
3 humidity    = BME280_GET_HUMIDITY(CALIBRATION)*100;
4 pressure    = (BME280_GET_PRESSURE(CALIBRATION));
5
6
7 AppData.Buffer[i++] = 0x02;
8 AppData.Buffer[i++] = (uint8_t)((temperature >> 8) & 0xFF);
9 AppData.Buffer[i++] = (uint8_t)(temperature & 0xFF);
0 AppData.Buffer[i++] = (uint8_t)((pressure >> 16) & 0xFF);
1 AppData.Buffer[i++] = (uint8_t)((pressure >> 8) & 0xFF);
2 AppData.Buffer[i++] = (uint8_t)(pressure & 0xFF);
3
4 AppData.Buffer[i++] = (uint8_t)((humidity >> 8) & 0xFF);
5 AppData.Buffer[i++] = (uint8_t)(humidity & 0xFF);
6
7
8 AppData.BufferSize = i;
9
0 #endif /* not CAYENNE_LPP */

```

Formatter type ^

Custom Javascript formatter



Formatter code *

```
1 function Decoder(b, port) {
2   var identity = b[0];
3   if (identity == 2) {
4     var temperature = ((b[1] << 8) + b[2])/100;
5     var humidity = ((b[6] << 8) + b[7])/100;
6     var pressure = ((b[3] << 16) + (b[4] << 8) + b[5])/100;
7     return {
8       field1: temperature,
9       field2: humidity,
10      field3: pressure
11    }
12  }
13  if (identity == 1) {
14    var CO2 = ((b[1] << 8) + b[2]);
15    var TVOC = ((b[3] << 8) + b[4]);
16    return {
17      field4: CO2,
18      field5: TVOC,
19    }
20  }
21 }
```



Ecole Publique d'Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04

