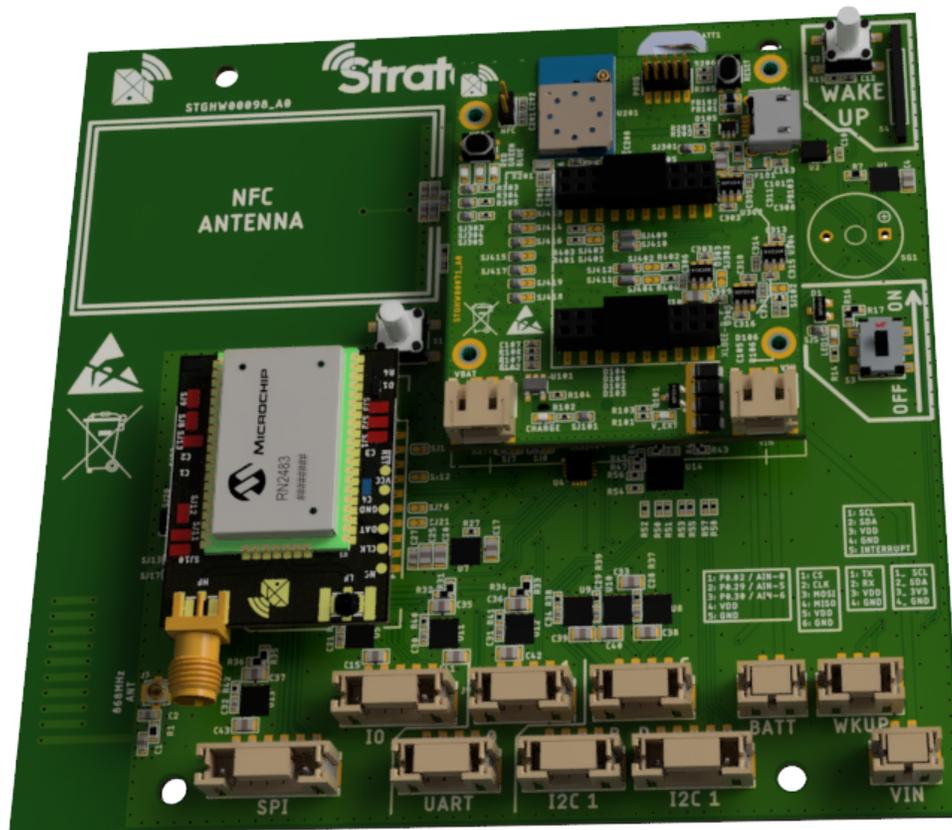


Manuel d'Utilisateur



Produit: **CAMEL v1**

Identifiant de produit: **STGHW-00098**



Information concernant le document

Nom du fichier	CAMELv1_user_manual.docx
Date de création	2020-09-23
Nombre de pages	12

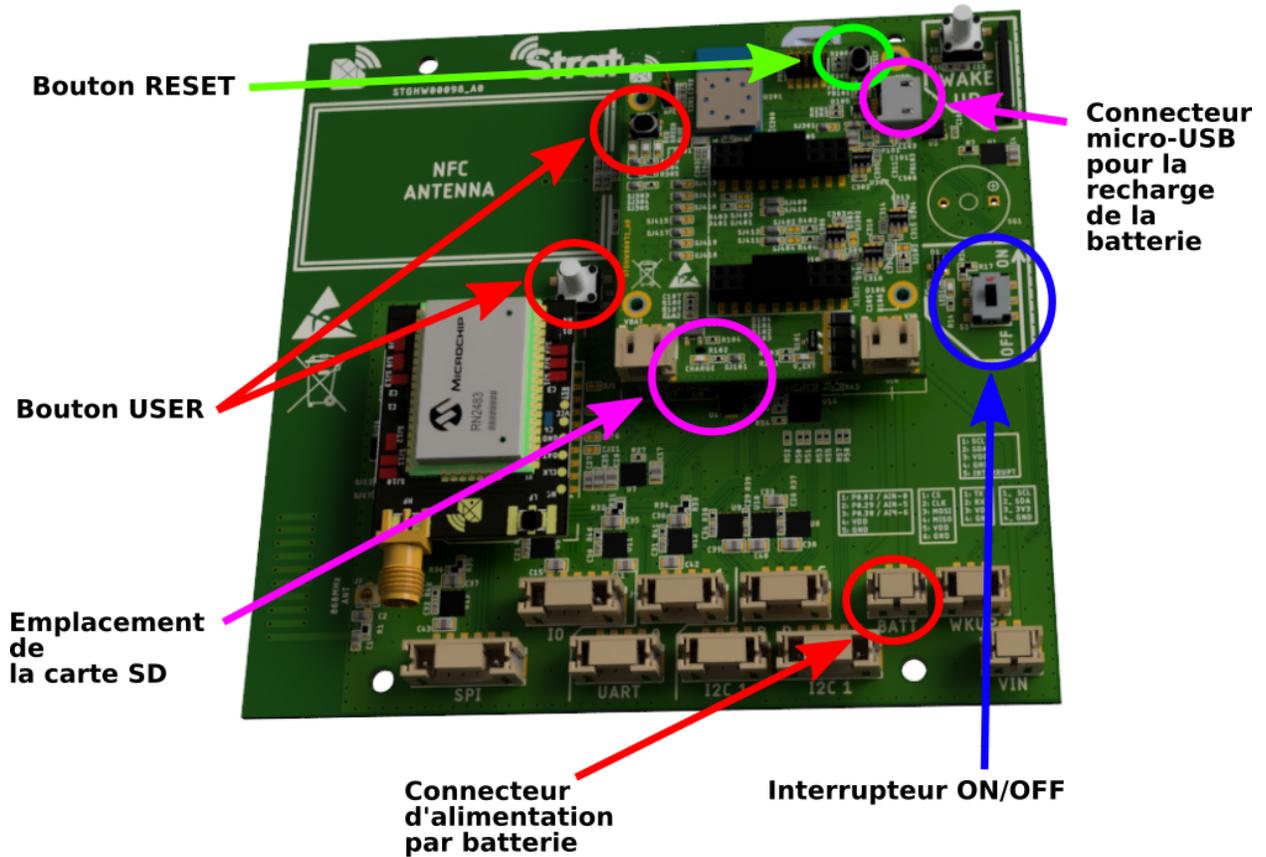
Historique des révisions

Version	Note
0.1	Création

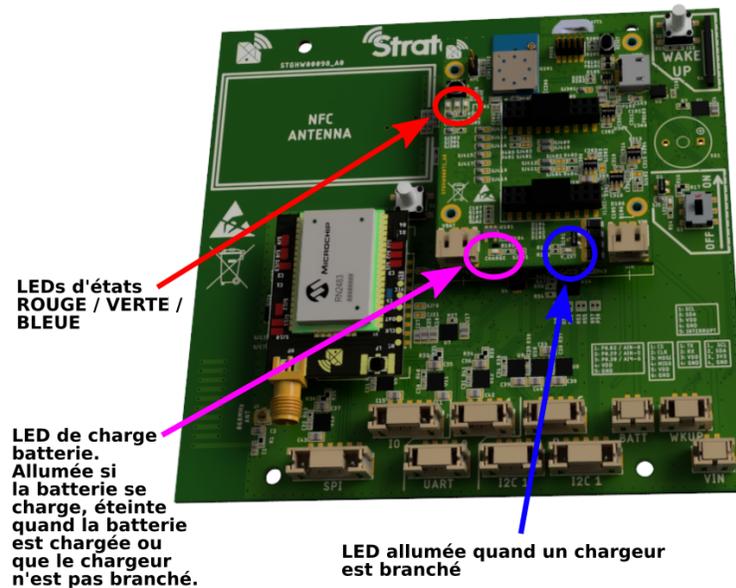
1 Guide d'utilisation	3
1.1 Schéma des principaux éléments	3
1.2 LEDs d'états	4
1.3 Guide rapide	4
1.4 Guide détaillé	4
1.4.1 À l'allumage	5
1.4.2 Pour changer la carte SD	5
2 Format du fichier de configuration	6
2.1 Explications des paramètres du fichier de configuration	6
2.1.1 time_between_measures_s	6
2.1.2 number_measures_before_write_sd	7
2.1.3 number_writes_in_interval_file	7
2.1.4 lora_active	8
2.1.5 rtc_type	8
3 Logigramme de fonctionnement	9
4 Tableau des LEDs de status	10
5 Codec Javascript pour décoder une trame radio	11

1 Guide d'utilisation

1.1 Schéma des principaux éléments



1.2 LEDs d'états



1.3 Guide rapide

1. Prenez une carte SD formatée en fat32 (vfat), le format standard.
2. Copiez-collez le fichier de configuration dans la carte SD (fichier présenté section [2](#) [Format du fichier de configuration](#)) dont un exemple est accessible avec ce [lien](#).
3. Brancher la batterie (3.3V) sur le connecteur *BATT*
4. Mettez l'interrupteur sur *ON*

Si vous voyez les LEDs qui clignotent et que la séquence se termine par la LED verte clignotant 3x, c'est bon le système enregistre périodiquement la température et l'humidité.

Si en plus vous avez vu une fois la LED bleu s'allumer 3 secondes alors le système transmet correctement les données en radio en temps réel.

1.4 Guide détaillé

De manière générale: quand la **LED rouge** est allumée il ne faut pas enlever la carte SD au risque de la corrompre.

Si la **LED rouge** clignote, alors il y a un problème avec la carte SD (absente, impossible à lire ou le fichier de configuration est absent ou impossible à lire).

1.4.1 À l'allumage

le système

1. fait un check, puis une mesure test et écrit sur la carte SD : la **LED ROUGE** est allumée pendant la durée de l'écriture.
2. si la communication radio est activée: établi la connexion radio
 - a. succès: **LED BLEUE** allumée pendant 3 secondes
 - b. erreur: passe à la suite
3. OK: la **LED VERTE** clignote **3x** puis s'éteint

S'il n'y a pas de carte SD : la **LED ROUGE** clignote.

Il suffit d'ajouter une carte SD et la séquence d'allumage se fera normalement.

1.4.2 Pour changer la carte SD

1. Appuyer 1x sur le bouton USER (à gauche à côté des LEDs, en dessous du logo Stratagem)
2. Attendre que la **LED VERTE** clignote
3. Enlever la carte SD
4. Remettre une carte SD
5. Appuyer 1x sur le bouton USER
6. Si tout va bien:
 1. le système écrit sur la carte SD (**LED ROUGE**): NE PAS ENLEVER LA CARTE SD.
 2. OK: clignote **3x LED VERTE** puis s'éteint

2 Format du fichier de configuration

Le fichier de configuration est un fichier json. Il doit être déposé à la racine de la carte SD et doit être nommé:

- *flipper_config.json*

vous pouvez trouver des exemples, dont celui ci-dessous, à télécharger sur le site de documentation de Stratagem qui se trouve à l'adresse suivante:

<https://wiki.stratagem.com/doku.php?id=products:camel:camelv1>

```
{
  "rtc_type": "am1805",
  "time_between_measures_s": 600,
  "number_measures_before_write_sd": 18,
  "number_writes_in_interval_file": 8,
  "lora_active": true
}
```

La configuration ci-dessus peut-être téléchargée [ici](#).

La taille du fichier doit être inférieure ou égale à:

- 1024 octets

Les paramètres peuvent être écrits dans n'importe quel ordre dans le fichier du moment que ce dernier respecte le format [json](#).

2.1 Explications des paramètres du fichier de configuration

2.1.1 time_between_measures_s

Ce paramètre est le délai approximatif entre 2 mesures de température et d'hygrométrie en secondes.

- Le paramètre est un entier positif codé sur 32bits. Il est donc possible d'avoir des délais extrêmement longs entre deux mesures (le maximum est plus grand que 136 ans ...).
- Le délai est respecté approximativement avec une précision inférieure à 20s.

Le tableau ci-après donne quelques valeurs pour des durées types souvent utilisées:

Délai	Valeurs du paramètre time_between_measures_s
10 minutes	600
30 minutes	1800
1 heure	3600
2 heures	7200
24 heures	86400
2 jours	172800
une semaine	604800
1 mois (30 jours)	2592000

2.1.2 number_measures_before_write_sd

Ce paramètre est le nombre de mesures stockées en mémoire avant qu'elles ne soient écrites sur la carte SD. En effet l'écriture sur une carte SD est une opération relativement coûteuse en énergie. Il est donc avantageux de faire plusieurs écritures d'un seul coup.

Dans l'exemple de configuration donné par le fichier de configuration présenté ci-dessus, la valeur est de 18 pour un intervalle de mesure de 10 minutes, donc les données sont écrites sur la carte SD toutes les 3 heures.

- Le paramètre est un entier positif codé sur 32bits.
- Si c'est nécessaire d'avoir une sauvegarde à toutes les mesures, le paramètre doit être mis à zéro (0).

2.1.3 number_writes_in_interval_file

Les données sont sauvegardées sur la carte SD à 2 endroits à chaque écriture :

1. dans un fichier global qui est créé à chaque fois que la carte SD est remise et dont le nom commence par la date et l'heure de création,
2. dans un fichier intermédiaire, à intervalle régulier (par exemple un fichier journalier), dont le nom comporte la date et l'heure de création.

Cela permet que si la carte SD a un problème et que le système de fichier commence à être corrompu, la probabilité reste grande que les données soient toujours présentes dans les fichiers d'intervalles et que seul le fichier global ait un problème.

Ce paramètre est le nombre de fois que des sauvegardes sont faites dans chaque fichier d'intervalle. Une fois que ce nombre est atteint, le fichier est fermé et un nouveau fichier d'intervalle est créé. Pour savoir combien de mesures sont stockées dans chaque fichier d'intervalle il faut multiplier

- $number_measures_before_write_sd \times number_writes_in_interval_file$
- Le paramètre est un entier positif codé sur 32bits.

Dans l'exemple de configuration donné par le fichier de configuration présenté ci-dessus, la valeur est de 8 et la valeur de `number_measures_before_write_sd` est de 18, donc le fichier d'intervalle stocke 144 mesures.

Comme les mesures sont effectuées toutes les 10 minutes un fichier d'intervalle stocke les données d'une journée entière (24h = 144 x 10 min).

2.1.4 lora_active

Ce paramètre permet d'activer ou désactiver la communication radio des mesures à chaque mesure effectuée. Cela peut être intéressant pour économiser la batterie ou si vous n'avez pas installé de module de communication LoRa.

Pour avoir la communication radio:

```
"lora_active": true
```

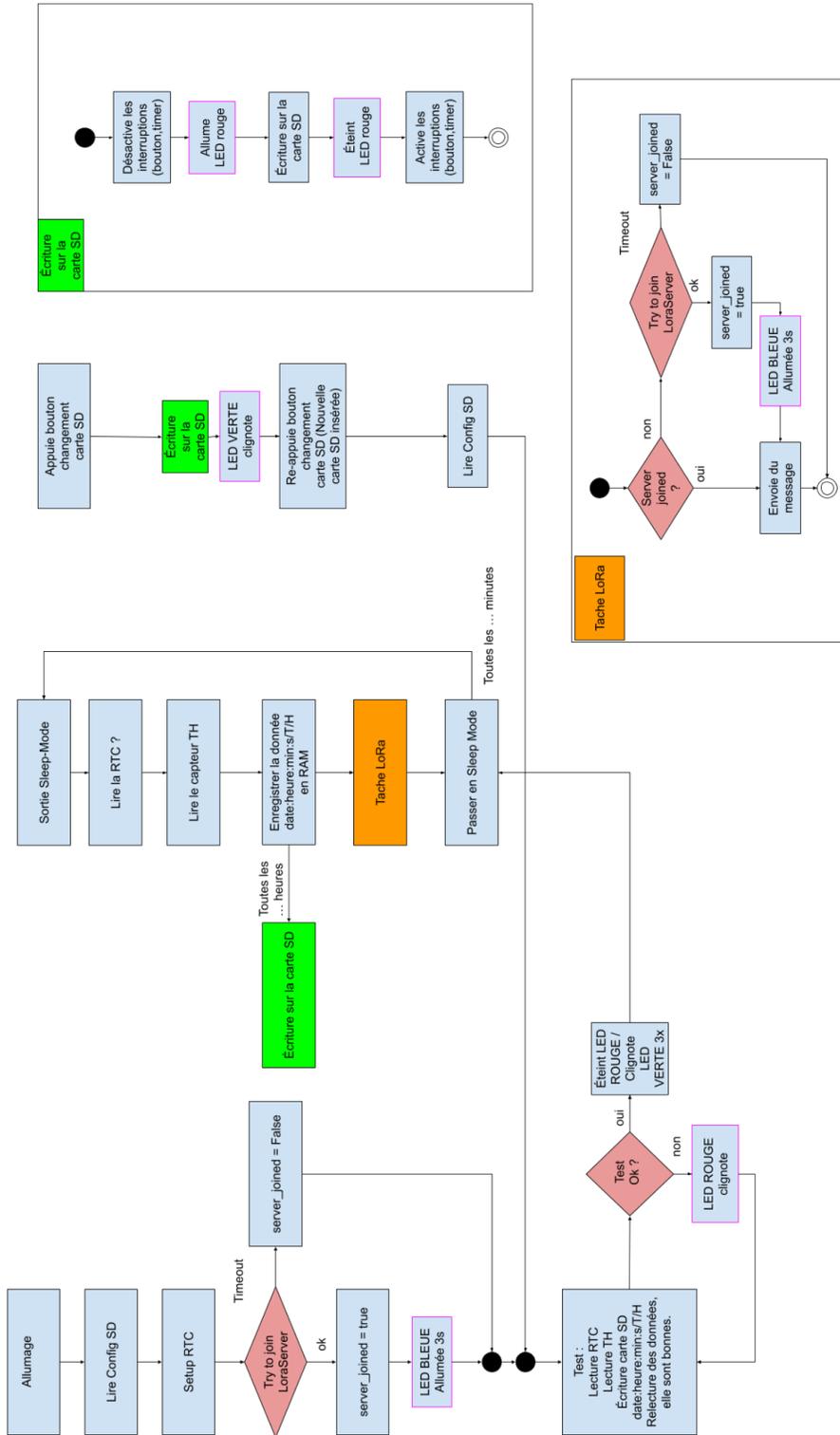
Pour supprimer la communication radio:

```
"lora_active": false
```

2.1.5 rtc_type

Ce paramètre donne le type d'horloge RTC (Real Time Clock) qui est utilisé par le système. Ce paramètre est présent pour des raisons de rétrocompatibilité, mais peut être omis pour tous les CAMELv1 acquis après 2021.

3 Logigramme de fonctionnement



4 Tableau des LEDs de status

État des LEDs	Signification	Opérations à effectuer
<p>LED Rouge : clignote toutes les secondes</p> <p>LED Bleue : éteinte</p> <p>LED Verte : éteinte</p>	<p>Erreur Sur la carte SD (Impossible d'écrire, de lire la configuration, erreur(s) dans le fichier de configuration, pas de carte SD présente)</p>	<p>1. Vérifier l'état de la carte SD, mettre une carte SD si elle est absente, mettre une carte formatée (vFAT), reformater si nécessaire si une carte est présente mais pas reconnue.</p> <p>2. Appuyer sur le bouton USER (relance la configuration du système).</p>
<p>LED Rouge : allumée fixe</p> <p>LED Bleue : éteinte</p> <p>LED Verte : éteinte</p>	<p>Opération sur la carte SD (Lecture de la config / Ecriture des données). L'opération dure en moyenne 1s.</p>	<p>Rien.</p> <p>Ne surtout pas enlever la carte SD à ce moment-là : risque de corruption de la carte SD.</p> <p>L'attente dure en moyenne 1s. Si vous voulez récupérer la carte SD, attendre que la LED rouge s'éteigne avant d'appuyer sur le bouton USER.</p>
<p>LED Rouge : éteinte</p> <p>LED Bleue : éteinte</p> <p>LED Verte : clignote toutes les secondes</p>	<p>La carte SD peut être retirée, en attente de confirmation que l'on peut relancer le système (par appuie sur le bouton USER).</p>	<p>1. enlever la carte SD,</p> <p>2. récupérer les données,</p> <p>3. remettre la carte SD en place,</p> <p>4. ré-appuyer sur le bouton USER pour relancer le fonctionnement normal (prise de mesures et log sur sur carte SD).</p>
<p>LED Rouge : éteinte</p> <p>LED Bleue : allumée pendant 3 secondes</p> <p>LED Verte : éteinte</p>	<p>Le système a correctement réalisé sa liaison avec le serveur LoRaWAN (join request: ok).</p>	<p>Rien.</p>
<p>LED Rouge : éteinte</p> <p>LED Bleue : éteinte</p> <p>LED Verte : clignote 3 fois</p>	<p>Le système est correctement configuré et a démarré normalement: toutes les opérations d'initialisations ont été effectuées.</p>	<p>Rien.</p>

5 Codec Javascript pour décoder une trame radio

```
// Decode decodes an array of bytes into an object.
// - fPort contains the LoRaWAN fPort number
// - bytes is an array of bytes, e.g. [225, 230, 255, 0]
// The function must return an object, e.g. {"temperature": 22.5}
var TEMP_BYTE_0_IDX = 0;
var TEMP_BYTE_1_IDX = 1;
var RH_BYTE_0_IDX = 2;
var RH_BYTE_1_IDX = 3;
var VOLT_BYTE_0_IDX = 4;
var LORA_MIN_BATT_VAL = 1;
var LORA_MAX_BATT_VAL = 254;
var LORA_BATT_RANGE = LORA_MAX_BATT_VAL - LORA_MIN_BATT_VAL;
var LORA_BATT_ERROR = 255;
var BATT_VOLTAGE_LOW = 3.0;
var BATT_VOLTAGE_HIGH = 4.2;

function Decode(fPort, bytes) {

    var temperature_code = 0;
    var humidity_code = 0;
    var temperature = 0.0;
    var humidity = 0.0;
    var voltage_code = 0;
    var voltage = 0.0
    temperature_code = (bytes[TEMP_BYTE_0_IDX] << 8 ) + bytes[TEMP_BYTE_1_IDX];
    humidity_code = (bytes[RH_BYTE_0_IDX] << 8 ) + bytes[RH_BYTE_1_IDX];
    voltage_code = bytes[VOLT_BYTE_0_IDX];

    temperature = ((175.72 * temperature_code) / 65536.0) - 46.85;
    humidity = ((125.0 * humidity_code) / 65536.0) - 6.0;
    var data = {};
    if (voltage_code != LORA_BATT_ERROR) {
        voltage = (((voltage_code - LORA_MIN_BATT_VAL) * (BATT_VOLTAGE_HIGH - BATT_VOLTAGE_LOW))/LORA_BATT_RANGE) +
        BATT_VOLTAGE_LOW;
        data =
        {
            "temperature": temperature,
            "humidity": humidity,
            "voltage" : voltage,
            "raw": bytes,
        };
    }
    else {
        data =
        {
            "temperature": temperature,
            "humidity": humidity,
            "raw": bytes,
        };
    }
}

return data;
}
```

Ce code peut être utilisé tel quel pour configurer le CODEC du Device-profiles de l'objet dans un serveur LoRaWAN Chirpstack.

Les données transmises sont:

1. la température sur 16 bits,
2. l'humidité sur 16 bits,
3. le voltage de la batterie sur 8 bits.

Le code est configuré pour une batterie Lithium-Ion avec un voltage haut à 4.2V et un voltage bas (cut-off) à 3.0V.