

## Bilan d'avancement du projet InitCube – Segment Vol

### TABLE DES MATIÈRES

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
1.1	Maquette 3D de la solution actuelle.....	2
<b>2</b>	<b>EXPRESSION DU BESOIN .....</b>	<b>2</b>
2.1	Besoin.....	3
2.2	Diagramme d'états/transitions .....	3
<b>3</b>	<b>RÉALISATION .....</b>	<b>4</b>
3.1	Descriptions matérielles.....	4
3.2	Contexte de test .....	5
3.3	Exemple de résultats obtenus .....	6
3.3.1	Zone 1 : Exemple de mesure de Pixels.....	6
3.3.2	Zone 2 : Exemple de mission .....	7
3.3.3	Zone 3 : Exemple d'état du système (ou données de servitude) .....	7

## 1 INTRODUCTION

### 1.1 Maquette 3D de la solution actuelle

La figure 1 ci-dessous est une capture de la maquette 3D de la solution développée lors de l'incrément qui s'est terminé au mois de Juin 2019.

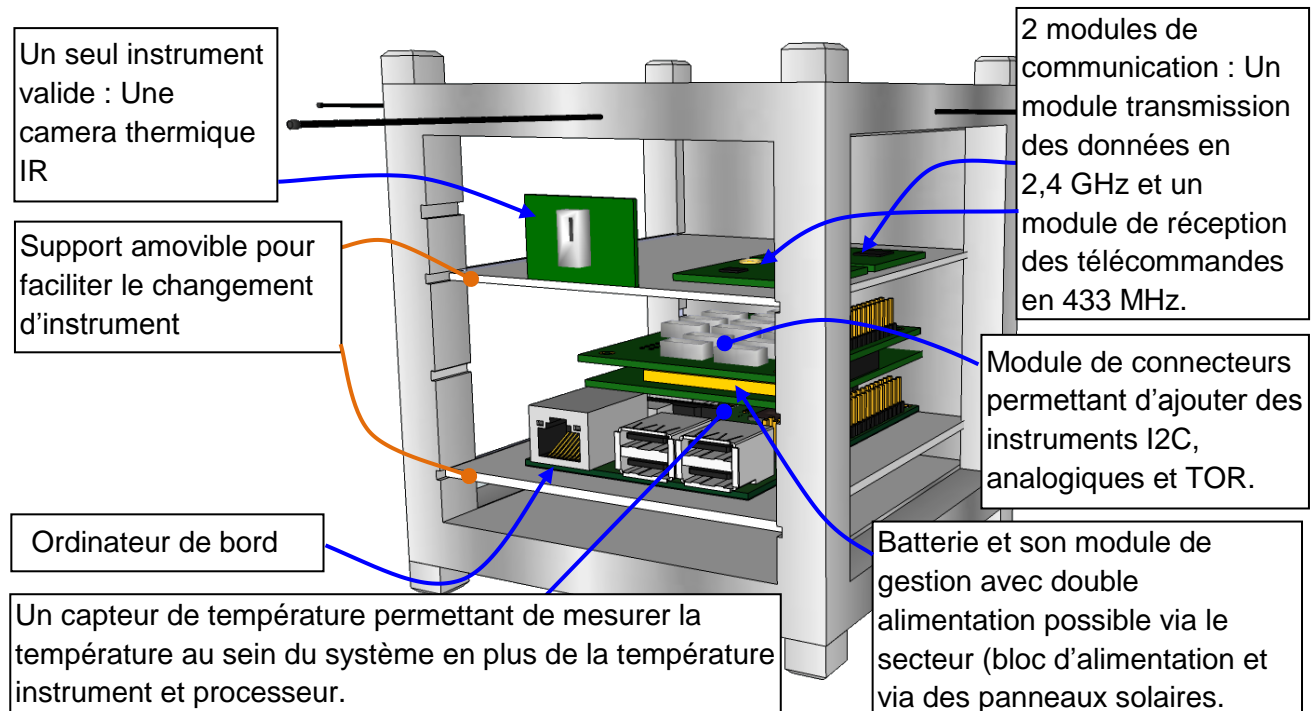


Figure 1 : Maquette 3D du segment Vol tel que validé actuellement

## 2 EXPRESSION DU BESOIN

L'analyse SysML est UML a été réalisée. On peut en extraire les diagrammes des figures 2 (Expression du besoin) et 3 (Diagramme des états envisagés pour le segment vol). D'autres diagrammes d'analyse sont disponibles sur demande.

## 2.1 Besoin

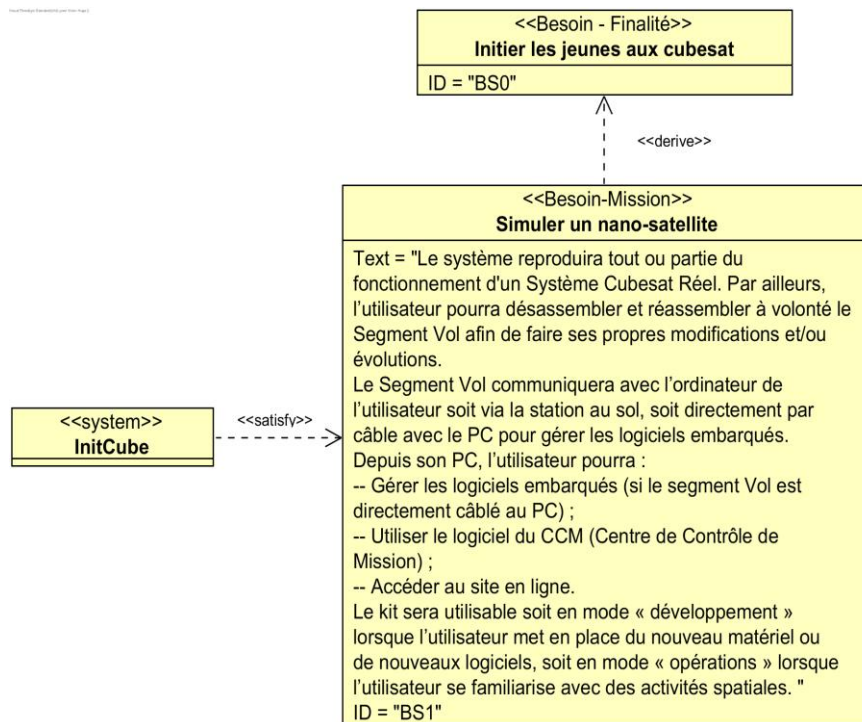


Figure 2 : diagramme du besoin du client

Le diagramme ci-dessus présente les besoins finaux du système sur lequel nous avons travaillé cette année. Le besoin principal auquel nous répondons est la simulation d'un nano-satellite.

## 2.2 Diagramme d'états/transitions

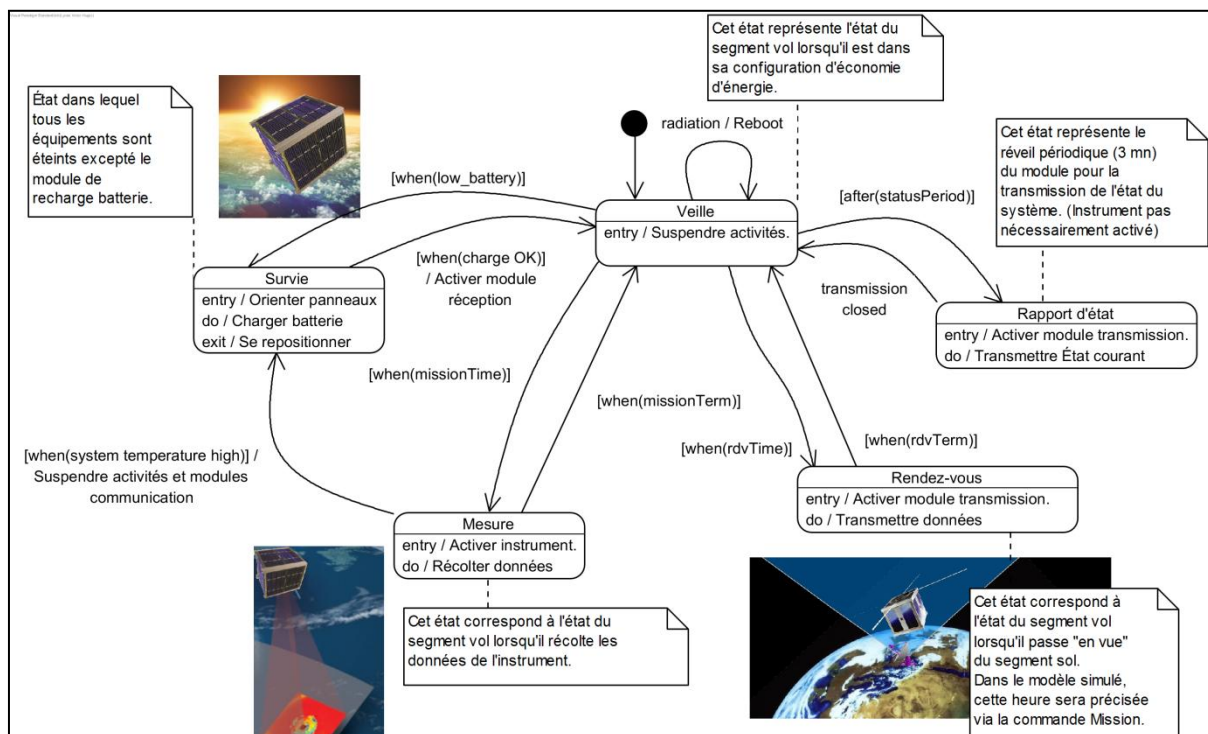


Figure 3 : Diagramme d'états/transitions

Le segment Vol d'Initcube simule d'ors et déjà les différents états d'un satellite qui sont :

- la veille pour lequel tous les appareils sont en mode **veille** pour des raisons d'économies d'énergie,
  - la mesure pour lequel l'instrument est en fonctionnement normal,
  - le rendez-vous pour lequel le satellite se trouve dans la zone du segment sol et transmet donc toutes ses données de mission,
  - le rapport d'état qui correspond à la transmission périodique de son état,
- Les trois derniers états (mesure, rendez-vous et rapport d'état) sont nommés « mission » dans le spatial.

Le mode survie dans lequel le satellite se met en cas de surchauffe (tous les appareils sont éteints) ou de charge batterie trop faible n'a pas encore été simulé.

Les différents modes de fonctionnement ne sont, pour le moment, pas configurables. Ils sont codés en dur dans le programme. Les étudiants n'ont pas eu le temps de :

- faire la lecture de la configuration dans un fichier XML (de configuration),
- transmettre et traiter les commandes en provenance du segment sol.

```
segmentVol->getHorloge()->setRdvTime("2019/04/04 15:25:00");  
segmentVol->creerMission(1, 4, "2019/07/04 15:25:00", "-TC");  
thread t1 = segmentVol->tLancerMission();  
thread t2 = segmentVol->tArretMission();
```

### 3 RÉALISATION

#### 3.1 Descriptions matérielles

La figure 4 montre l'apparence réelle du segment vol.

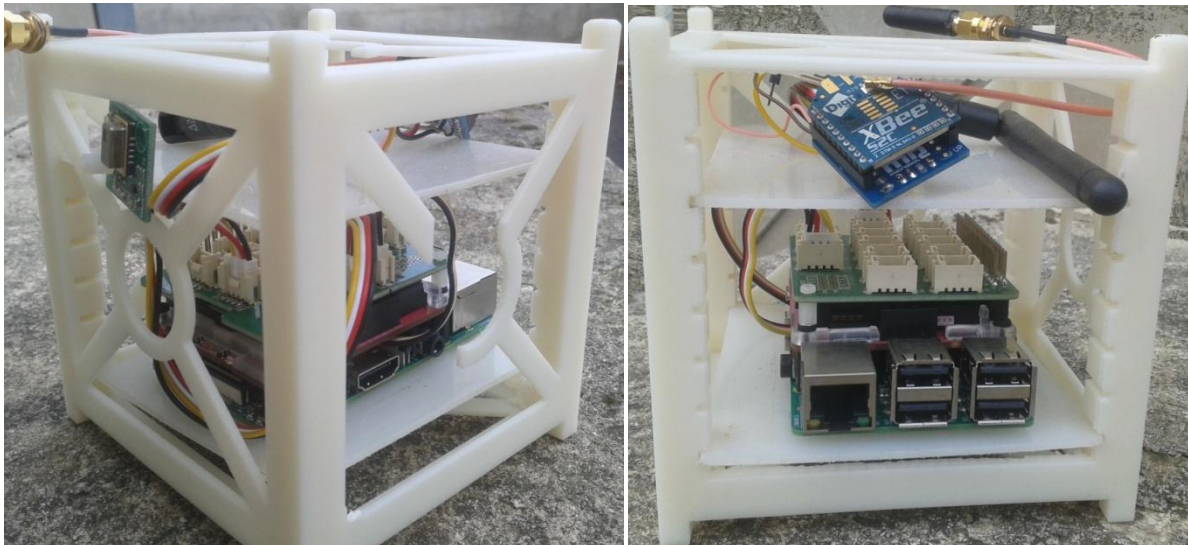
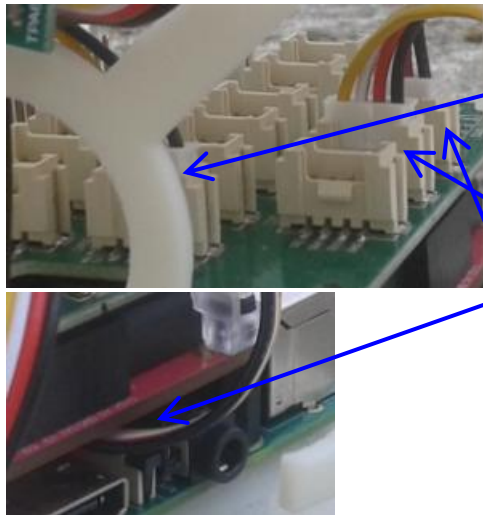


Figure 4 : apparence réelle



- 3 connecteurs sont utilisés sur le CAPE :
- Le connecteur UART pour les communications série avec les modules de transmission/réception.
  - 1 connecteur I2C pour l'instrument (Camera thermique)
  - 1 connecteur I2C pour la température du système (État de l'InitCube ou donnée de servitude)

### 3.2 Contexte de test

Comme illustré à la figure 5, le test du segment vol mettait en oeuvre un PC sur lequel était installé un logiciel de simulation de station sol (Lecture des données transmises par le segment Vol).

Des modules émetteurs/récepteurs étaient branchés à ce PC via une carte de développement qui assurait la conversion USB/série TTL et l'alimentation des modules.

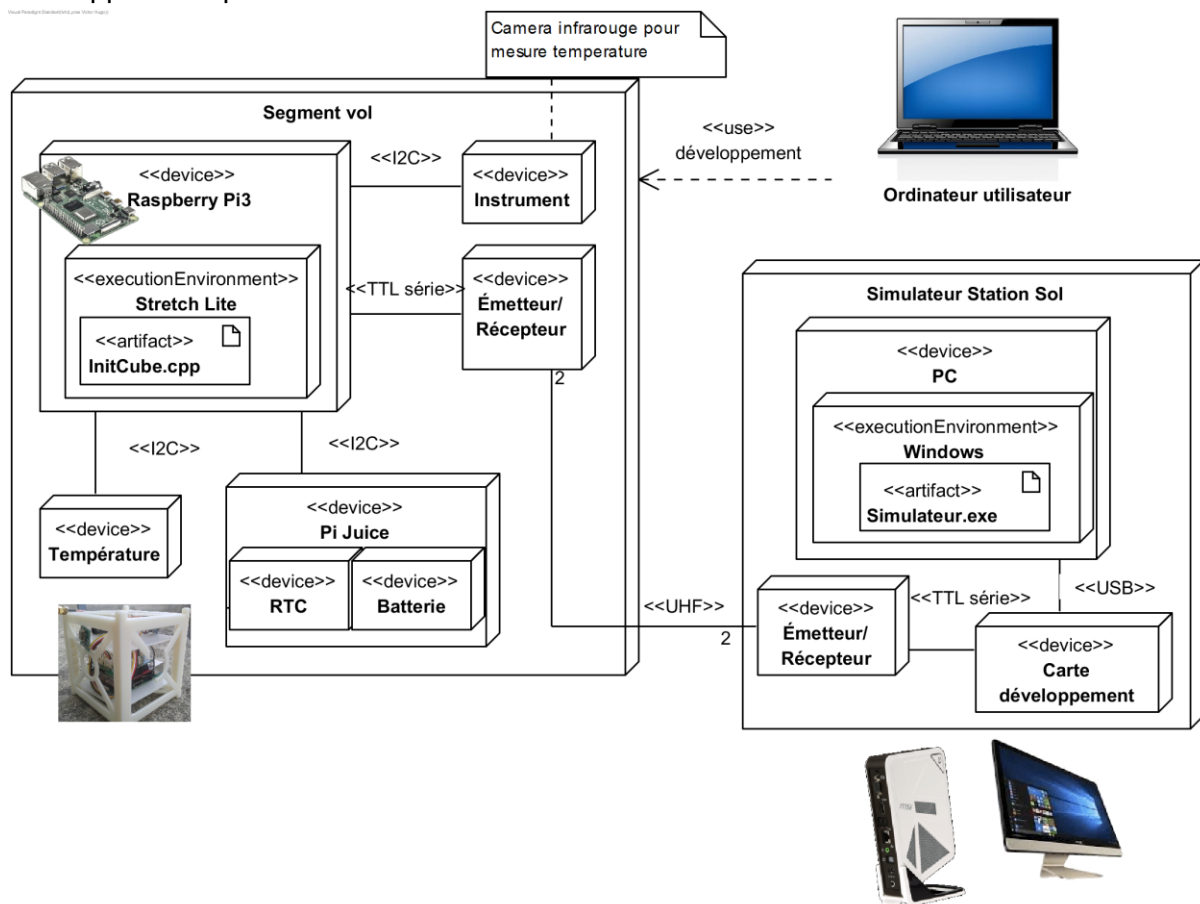


Figure 5 : diagramme de déploiement de test



### 3.3 Exemple de résultats obtenus

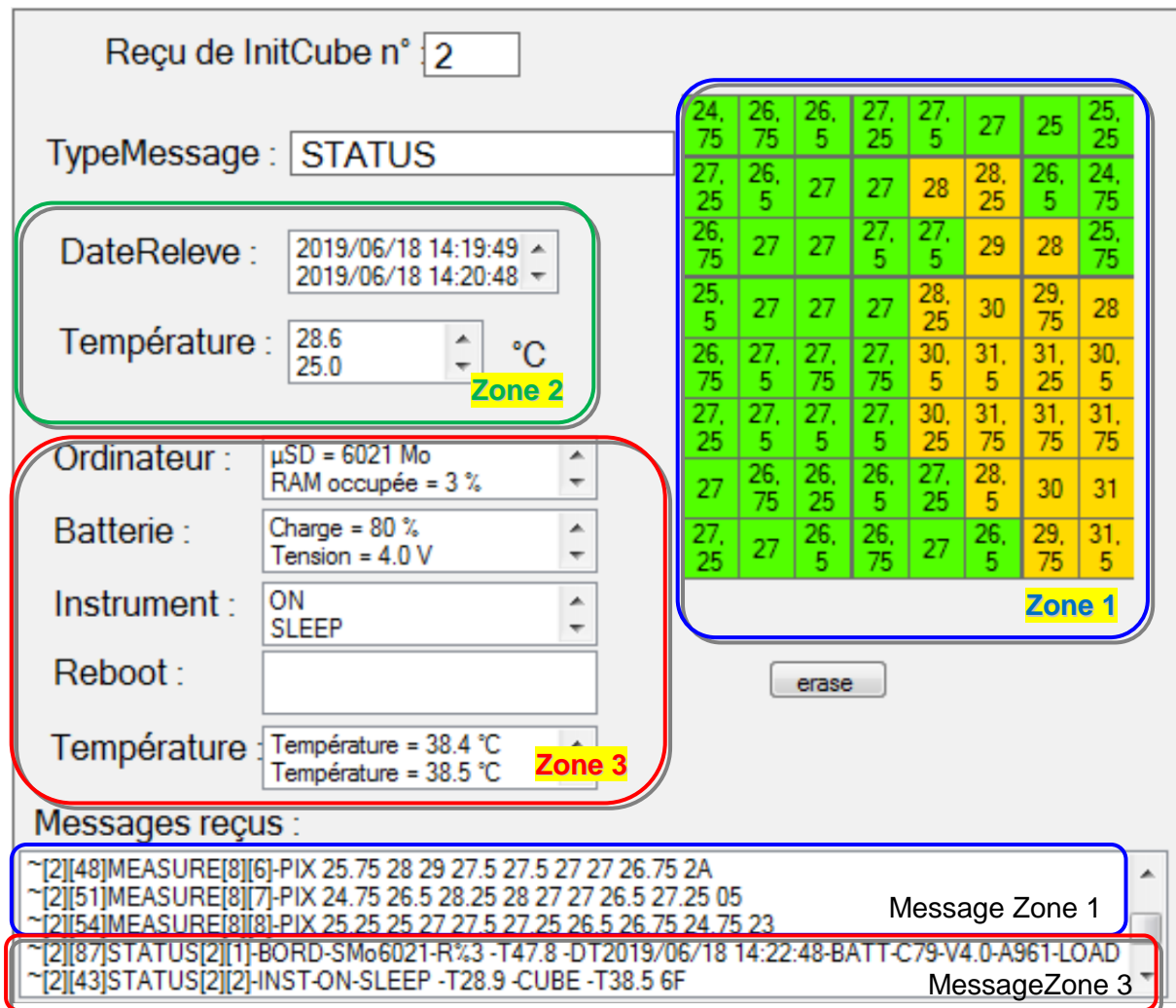


Figure 6 : affichages de résultats sur un simulateur

La figure 6 est une capture d'écran de l'IHM du simulateur du Segment Sol utilisé sur un PC Windows. Il permet de visualiser ce qui a été transmis par le segmentVol. Dans ce cas, il s'agissait d'une mission (Zone 2), d'une mesure ponctuelle (Zone 1) de l'état du système (Zone3).

#### 3.3.1 Zone 1 : Exemple de mesure de Pixels

La **zone 1** de la figure 6 représente un exemple de transmission d'une image de la camera thermique.

Le **message Zone 1** représente un exemple de trames transmises par le segment Vol pour fournir le résultat de la zone 1.

Compte tenu des possibilités du module d'émission, le protocole est construit autour de 8 paquets contenant les éléments suivants :

~1	56	MEASURE	8	1	-	P	I	X	_29.6	_28.6	_27.6	_26.6	_25.6	_24.6	_10.6	_11.6	5	A
2 octets (début et numéro d'InitCube)	Nbre octets sur 1 octet	7 octets (Code commande à laquelle on répond)	Nombre paquets sur 1 octet	numéro paquet 1 octet	Type mesure et unité 4 octets	pixel 1 en °C 5 octets	pixel 2 en °C 5 octets	pixel 3 en °C 5 octets	pixel 4 en °C 5 octets	pixel 5 en °C 5 octets	pixel 6 en °C 5 octets	pixel 7 en °C 5 octets	pixel 8 en °C 7 octets	2 octets Checksum				

Remarque : il est possible de réduire le nombre de paquets.

### 3.3.2 Zone 2 : Exemple de mission

La **zone 2** de la figure 6 représente des résultats de mission. Une mission est un ensemble de mesures effectuées périodiquement, pendant une durée et à partir d'une date qui seront configurables (Pour le moment, ce n'est pas le cas.)

Sur cette capture, on voit que le 18 juin 2019 à 14h19mn49s, la valeur moyenne des températures mesurées par la caméra thermique était de 28,6 °C.

La mesure suivante a eu lieu 1 mn plus tard et il faisait 25 °C.

Compte tenu des possibilités du module d'émission, le protocole est construit de telle façon qu'il y ai autant de paquets que de paires de mesures :

~1	73	MISSION	2	1	-	T	C	~T12.5	~DT2019/11/12_12:12:12	~T14.5	~DT2019/11/12_12:22:12	1	6
2 octets (début et numéro d'InitCube)	Nbre octets sur 1 octet	7 octets (Code commande à laquelle on répond)	Nombre paquets sur 1 octet	numéro paquet 1 octet	Type mesure et unité 3 octets			Date relevé 1		Date relevé 2		2 octets Checksum	

~ : Espace (0x20)

~1	72	MISSION	2	2	-	T	C	~T16.5	~DT2019/11/12_12:32:12	~T18.5	~DT2019/11/12_12:42:12	3	5
2 octets	Nbre octets sur 1 octet	7 octets	Nombre paquets sur 1 octet	numéro paquet 1 octet	Type mesure et unité 3 octets			Date relevé 3		Date relevé 4		2 octets	

~ : Espace (0x20)

### 3.3.3 Zone 3 : Exemple d'état du système (ou données de servitude)

Dans la zone 3 de la figure 6, on trouve les différentes données en lien avec l'état du système. Il s'agit :

- des données de l'ordinateur de bord : espace libre en  $\mu$ SD = 6021 Mo, espace occupé en RAM = 3%, température processeur = 47,8 °C et heure système au moment du scan= 18 juin 2019 à 14h22mn48s ; (Rem : ces 2 dernières valeurs apparaissent plus bas dans le menu déroulant du simulateur mais on les voit dans le message)

- des données de la batterie : la charge est à 80 % (79 dans la trame), tension = 4 V, courant de charge = 961 mA et son état est « en charge » (LOAD) ; (Rem : ces 2 dernières valeurs apparaissent plus bas dans le menu déroulant du simulateur mais on les voit dans le message)

- de l'état de l'instrument : ON, SLEEP et température interne = 28,9 C ; (Rem : cette dernière valeur apparait plus bas dans le menu déroulant du simulateur mais on la voit dans le message)

- de la température du système (entre ordinateur de bord et module de gestion de charge batterie) = 38,5 °C au moment du dernier scan.

Compte tenu des possibilités du module d'émission, le protocole est construit de telle façon qu'il y ai autant de paquets que nécessaires pour transmettre l'état de l'InitCube :

~1	88	STAT US	2	1	-BORD	- SMo17 60	-R%25	~ T41.7	~ DT2019/11/02_12:1 2:12	- BATT	-C75	-V4.5	- A750	- LOA D	~ T50.4_	0	C
2 octets	Nbre octets sur 1 octet	7 octets	nbre Paquets	num paquet												2 octets checksum	

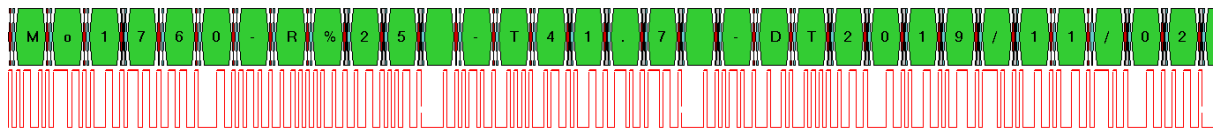
~ : Espace (0x20)

~1	74	STATUS	2	2	-INST	-ON	- NORM	~ T37.8	~ CUBE	~ T36.2_	-REBOOT	- DT2019/05/20_1 0:00:12	-N1	3	5
2 octets	Nbre octets sur 1 octet	7 octets	nbre Paquets	num paquet								date du dernier reboot	nbre reboot	2 octets checksum	

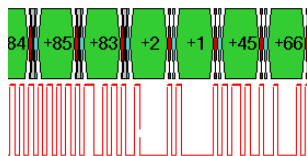
~ : Espace (0x20)

Rem : le Reboot n'était pas géré en juin 2019.

L'information est codée en ASCII :



Excepté le numéro d'InitCube, le nombre d'octets, le nombre et le numéro de paquets.



Cela permettra la transmission d'images avec un nombre de paquets pouvant atteindre 255.