

# The French Pratham ground station software system

Équipe Pratham

Université Paris Diderot – Institut de Physique du Globe de Paris,  
4 avenue de Neptune, 94100 Saint-Maur-des-Fossés, France

(Dated: August 25, 2012)

The copyright to this document and its source code are held by Institut de Physique du Globe de Paris. All rights reserved. This file is distributed under the license CeCILL Free Software License Agreement [1].

## I. INTRODUCTION

The following application note introduces technical issues related to the control-data-acquisition software system of the ground station at IPGP for Pratham satellite [2]. It aims to give scientific answers to the choice of implemented structures, including: antenna rotor control, time-based acquisition scheduler, FSK (FREQUENCY-SHIFT KEYING) and OOK (ON-OFF KEYING) demodulation, AX.25 frame decoding, MORSE code decoding and data recording.

**Warning.** Section marks 🚨 give scientific arguments but not necessarily relevant information for the understanding of this document. Some algorithm implementation details have been omitted for more abstraction towards specific programming languages.

### A. Schéma d'acquisition général

The NI-6353 [16] [15] Acquisition Box (considered as an ADC) acquires twelve voltage measurements in volt (V).

TABLE I: Set of voltage measurements to acquire

|                     |             |                           |         |
|---------------------|-------------|---------------------------|---------|
| 145,980 MHz antenna | Plane 1 ... | VR5000 →                  | NI-6353 |
|                     |             | AD8302 <sub>mag</sub> →   |         |
|                     |             | AD8302 <sub>phase</sub> → |         |
|                     | Plane 2 ... | VR5000 →                  |         |
|                     |             | AD8302 <sub>mag</sub> →   |         |
|                     |             | AD8302 <sub>phase</sub> → |         |
| 437,455 MHz antenna | Plane 1 ... | VR5000 →                  |         |
|                     |             | AD8302 <sub>mag</sub> →   |         |
|                     |             | AD8302 <sub>phase</sub> → |         |
|                     | Plane 2 ... | VR5000 →                  |         |
|                     |             | AD8302 <sub>mag</sub> →   |         |
|                     |             | AD8302 <sub>phase</sub> → |         |

**Remarque.** We introduce on TABLE I a flowchart of acquisition channels at the input of ADC. **Blue** arrows (→) match with channels from which the ADC acquire voltage measurements (in Volts). In the same way, **green** (resp. **red**) arrows match with as well voltage measurements acquisition as OOK demodulation and MORSE

decoding (resp. FSK demodulation and AX.25 frame decoding).

### B. Digital signal at 437,455 MHz : Health monitoring data

The following process is applied to the two channels specified by **red** arrows of the previous flowchart. At the end of the process we then have two output files for the same information – as a matter of fact – more accuracy.

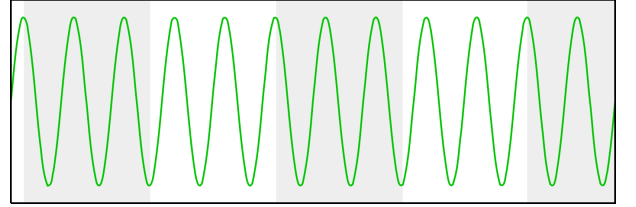


FIG. 1: Carrier signal (central frequency  $F_c$ )

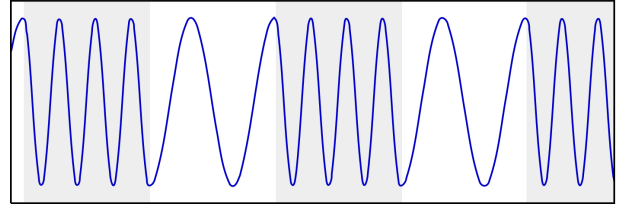


FIG. 2: Modulated signal in FSK at  $F_c \pm \frac{\Delta f}{2}$  frequencies received by the VR5000

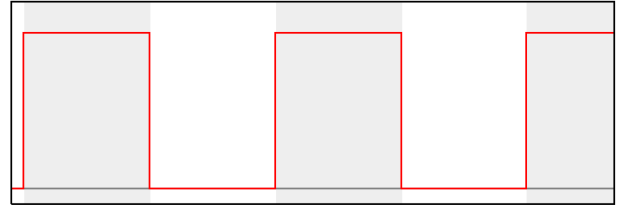


FIG. 3: Modulating signal at  $F_b = 1,2 \text{ kbit.s}^{-1}$  bit rate (retrieved by FSK demodulation)

The satellite sends a signal with a declared carrier signal at  $F_c = 437,455 \text{ MHz}$  (FIGURE 1) and a passband

bandwidth of  $\Delta f = 9,90$  kHz (ref ?). At the end of the acquisition chain the VR5000 transceiver acquires a signal whose central frequency is  $F_c$  and whose passband bandwidth is  $\Delta f$  (FIGURE 2).

The transceiver returns an audio signal. It finally describes – through a Schmitt trigger (ref ?) – the modulating signal (FIGURE 3). The digital baseband transmission method is of the kind non-return-to-zero (NRZ) [40] (ref ?). The data bits are then saved in a file in such a way that (more info on peak sync ? method not sure...)

La démodulation effectuée par le transceiver renvoie un signal audio, correspondant directement au *signal modulant* (FIGURE 3). Les changements d'états codent les bits de données en NRZ [40]. Ces bits de données sont par la suite enregistrés dans un fichier de telle façon que – sur une période  $T_b$  – la présence d'une amplitude positive non-négligeable (moyenne du signal sur une durée  $T_b$ ) corresponde au 1 logique et son l'amplitude nulle au 0. Ces bits de données sont par la suite enregistrés dans un fichier (FIGURE 4).

... 01111110 ... 10101010 ... 01111110 ...  
debut de trame information fin de trame

FIG. 4: Bits de données correspondant au signal modulant

On ne s'intéresse cependant qu'à une partie des bits de données. L'information issue du satellite – codée en AX.25 – assure une fiabilité dans la transmission de l'information. On découpe alors les bits en données en trames (FIGURE 5).

... 10101010 ...

FIG. 5: Bits de données décodées des trames AX.25

Les bits restants du découpage sont alors les bits de données au format définies par l'ITB[3].

### C. Signal numérique à 145,980 MHz : Beacon

De manière analogue, le transceiver démodule un signal à fréquence centrale  $F_c = 145,980$  MHz (FIGURE 6). Avec un débit binaire de  $F_b = 10$  bit.s<sup>-1</sup>, la présence de la composante fréquentielle  $F_c$  indique une impulsion et, celle de la composante 0, aucune impulsion (On-Off Keying) (FIGURE 7).

On déchiffre ensuite l'information codée en morse suivant la longueur des impulsions (trait long ou trait court) suivant le système de codage international (FIGURE 8).

On applique alors un tel traitement aux deux voies (flèches vertes sur le schéma précédent) afin d'obtenir deux informations et donc plus de précision.

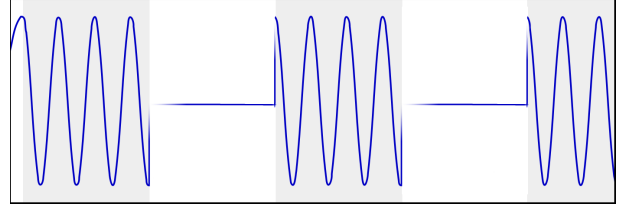


FIG. 6: Signal modulé en OOK à fréquence  $F_c$  reçu par le VR5000



FIG. 7: Signal numérisé correspondant à des impulsions

### D. Rapports de gain et de phase : information sur le TEC

Huit voies (flèches bleues du schéma précédent) sont dédiés à l'acquisition de rapports de gain et de phase, issus des AD8302. On enregistre directement en fichier les mesures de tensions des huit voies restantes. Un post-traitement de ces valeurs sera effectué avec Matlab afin de récupérer l'information sur le TEC.

## II. THÉORIES ET IMPLÉMENTATIONS

### A. Contrôle motorisé de l'antenne

Le contrôle de la position des axes de l'antenne est entièrement gérée par le logiciel NLSA Nova [9]. Le contrôleur de l'antenne étant relié à l'ordinateur par RS-232, Nova gère à la fois :

- la gestion et la mise à jour de la base de données sur les positions des satellites (Space-Track);
- la génération d'informations sur les élévations minimales et maximales de satellites ;
- le contrôle moteur de l'antenne suivant la trajectoire du satellite désiré dans le référentiel de l'antenne.

**Remarque.** Pour les détails techniques sur la mise en route de Nova, cf. APPENDIX II.

### B. Gestion du temps et minuterie

**Technique.** Les données de temps nécessaires à la mise en route de la minuterie sont fournies par

... VU2MDQ ...

FIG. 8: Information décodée du code morse

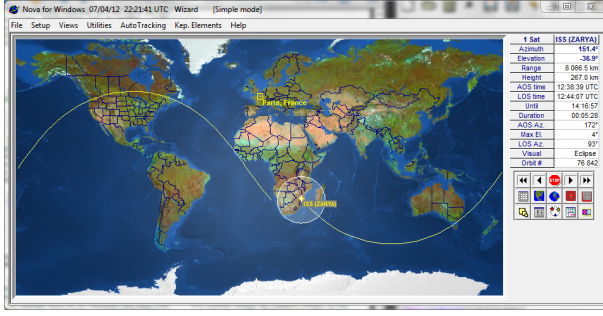


FIG. 9: Logiciel Nova de NLSA

Nova (fichier Nova listing data.TXT) en suivant la procédure suivante :

1. Choisir le satellite et le référentiel de préférence (Configure current view)
2. Utilities → Listing
3. One Observer AOS/LOS → ReCalc
4. Capture Listing → ASCII Text File (Entire listing)

Par les étapes d'analyses lexicales et syntaxiques sur ce fichier, le logiciel-mre renvoie un ensemble de séquences d'acquisition.

**Remarque.** ❄️ (SYNTAXE DES FICHIERS DE DESCRIPTION D'ÉCLIPSE DE NOVA) Les informations d'élévation min/max d'un satellite renvoyées par le logiciel Nova sont enregistrées par ce dernier dans une syntaxe dont on donne la grammaire en forme Backus–Naur étendue [12] (FIGURE 18).

**Remarque.** (FUSEAU HORAIRE) Pour des raisons de commodités d'implémentation, on suppose que la machine est configurée au fuseau horaire UTC ou GMT+0.

**Définition.** ❄️ (SÉQUENCE D'ACQUISITION) Une *séquence d'acquisition* est un couple de temps correspondant respectivement à la date de début d'acquisition et à la date de fin d'acquisition. On note  $\mathcal{S}$  un ensemble de séquences d'acquisition ordonné.

L'algorithme suivant (ALGORITHME 1, IMPLÉMENTATION 10) est proposé afin d'implémenter une minuterie qui déclenche une séquence d'acquisition.

**Remarque.** (PRÉCISION DES DÉCLENCHEMENTS ET ARRÊTS DE SÉQUENCES D'ACQUISITION) Soit  $\varepsilon > 0$ . Si l'on note  $F_r$  la fréquence de rafraîchissement du logiciel, on risque d'avoir une erreur de retard d'acquisition de l'ordre de  $\frac{1}{F_r} - \varepsilon$ . La boucle d'acquisition vérifie le temps initial à période  $\Delta t$  (largeur de fenêtre de calcul). On risque alors d'avoir une erreur de dépassement de temps d'acquisition à  $\Delta t - \varepsilon$  près.

### Algorithme 1 Minuterie

---

**Entrées:**  $\mathcal{S}$  non-vidé,  $t_i()$  : temps à l'appel de  $t_i$ ,  $F_s$  : fichier de la séquence  $s$ ,  $F_r$  : fréquence de rafraîchissement

- 1: **pour**  $s \in \mathcal{S}$  par ordre  $R$  **faire**
- 2: **tantque**  $\neg(t_d(s) \leq t_i()) < t_f(s)) \wedge (t_i() < t_f(s))$  **faire**
- 3:  $\text{sleep}(\frac{1}{F_r})$  // rafraîchissement
- 4: **fin tantque**
- 5:  $Q$  : file
- 6:  $Q \leftarrow \text{Boucle d'acquisition}(t_i(), t_d(s), t_f(s))$
- 7:  $F_s \leftarrow \text{Boucle d'enregistrement}(Q)$
- 8: **fin pour**
- 9: **retourner**  $F_{s_1}, \dots, F_{s_{|\mathcal{S}|}}$

---

### C. Boucle d'acquisition

**Technique.** Le temps d'échauffement recommandé du boîtier d'acquisition avant usage est de 15 minutes[16].

**Définition.** (BOUCLE D'ACQUISITION) Une *boucle d'acquisition* est une structure de contrôle permettant l'acquisition répétée et cadencée des valeurs retournées par le boîtier d'acquisition à un instant  $t$  durant un temps fini  $\Delta t$  à une fréquence  $F_e$ .

Une séquence d'acquisition est réalisée par une boucle d'acquisition (ALGORITHME 2) qui appelle la fonction DAQ Assistant 11 [18] [23] et stocke les valeurs retournées en mémoire morte.

**Remarque.** Une minimisation des temps de retards imposés par les limitations physiques du disque dur est rendue optimale par la mémoire tampon du boîtier d'acquisition[16] sans intervention de l'utilisateur.

### Algorithme 2 Boucle d'acquisition

---

**Entrées:**  $t_i()$  : temps initial,  $t_d$  : temps de début d'acquisition,  $t_f$  : temps de fin

- 1:  $F$  : fichier
- 2: **tantque**  $t_d \leq t_i() < t_f$  **faire**
- 3:  $F \leftarrow \text{acquérir}()$
- 4: **fin tantque**
- 5: **retourner**  $F$

---

### D. Échantillonnage

On se propose d'introduire quelques notations permettant de clarifier les prochaines configurations.

**Notation.** Soit  $x : I \rightarrow \mathbb{R}, t \mapsto x(t)$  un signal à  $N$  échantillons indexés par  $I$ . On note :

$F_c$  : la fréquence centrale [Hz]

$\Delta f$  : la largeur de bande passante [Hz]

$F_e$  : la fréquence d'échantillonnage [Hz]

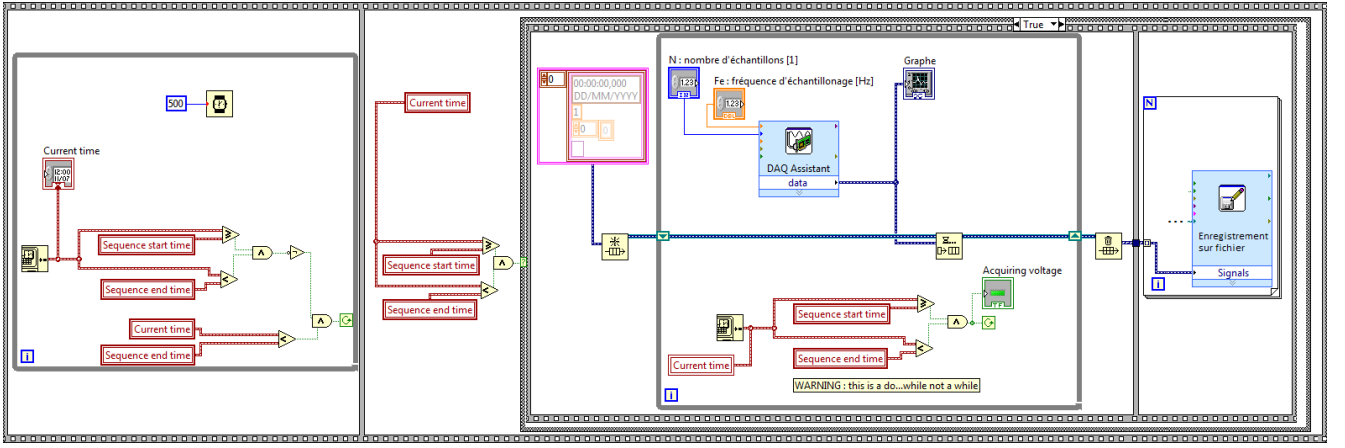


FIG. 10: Implémentation de la minuterie en LabVIEW

$F_b$  : le débit binaire du signal modulé [ $\text{Hz} = \text{bits} \cdot \text{s}^{-1}$ ]

$N$  : le nombre d'échantillons [1]

$\Delta t$  : la largeur de fenêtre de calcul [s]



FIG. 11: Fonction DAQ Assistant

**Configuration.** Lors de la définition de la fonction DAQ Assistant, les paramètres suivants correspondent aux valeurs définies plus haut :

- Acquisition mode : choisir Continuous Samples [19]
- Samples to Read :  $N$
- Rate (Hz) :  $F_e$

**Remarque.** (MÉMOIRE TAMPON FIFO DU BOÎTIER) Le boîtier possède une mémoire tampon FIFO de 4095 échantillons [16] (csq ?).

**Remarque.** (COMPLEXITÉ) Le coût en temps de la fonction DAQ Assistant s'exprime en  $\Omega(N)$ [37] et prend un temps  $\Delta t = \frac{N}{F_e}$  entre l'appel de fonction et le retour des valeurs.

La largeur de fenêtre choisie est de  $\frac{1}{F_b}$  (pk?).

On justifie dans les paragraphes suivants les choix de configurations d'échantillonnage optimaux pour l'acquisition.

**Théorème. II.1** (NYQUIST-SHANNON [34] [35])

$$F_e \geq 2 \times \max \{F_c; F_b\} \quad (1)$$

**Remarque.** (ÉCHANTILLONAGE OPTIMAL) Le théorème énonce que, pour un signal numérique, la fréquence d'acquisition de base théorique est de 0,5 Hz par baud. En pratique, il faudrait au moins entre 0,7 et 0,8 Hz par baud[11].

Le boîtier d'acquisition NI-6353 peut fournir une fréquence d'échantillonnage de 1 Méch./s sur plusieurs voies [16]. L'acquisition de toutes les voies devant être quasi-simultanée, il paraît trivial que toutes les voies soient au moins échantillonnées à  $2 \times 2,4 \times 10^3 = 4,8 \times 10^3$  Hz (par application du thémème précédent). On note cependant que certains DSP ne se limitent qu'à  $4 \times F_c$  pour des raisons de surcharge de calculs [27].

En considérant un post-traitement des opérations de nuérisation des signaux, on peut alors, pour des raisons de précision de mesures, échantillonner chacune des voies à 10 kHz.

#### E. Détection de composante fréquentielle et transformée de Fourier

❖ Cette partie n'entre en aucune considération dans le plan logiciel. Il n'est qu'une justification quant à l'usage des démodulateurs de fréquence pourvus par les quatre transceivers VR-5000. Cette dernière assure les opérations de calcul d'amplitudes associées aux composantes fréquentielles définies plus haut, incluant nativement des tranformations de Fourier.

**Définition.** (SPECTRE FRÉQUENTIEL) Soit  $\mathcal{F}$  la transformée de Fourier d'une fonction intégrable sur  $\mathbb{R}$  et  $\hat{x} = \mathcal{F}(x) = [k \mapsto \hat{x}(k)]$  le spectre fréquentiel de  $x$  au  $k$ -ième échantillon. On a[6] :

$$\begin{aligned}\hat{x}(k) &= \sum_{n=0}^{N-1} x(n) e^{\frac{-2ik\pi n}{N}} \\ &= \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2k\pi n}{N}\right) - i \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2k\pi n}{N}\right)\end{aligned}$$

On note  $A(f)$  l'amplitude de la composante fréquentielle  $f$ .

**Proposition. II.2** (AMPLITUDE) *La transformée de Fourier d'un signal à  $N$  échantillons est une application qui à un indice  $k$  ( $k \in [0; N-1]$ ) associe le spectre de la fréquence  $f$  tel que  $f = \frac{k.F_e}{N}$ .*

On a alors que[6] :

$$\begin{aligned}A(f) &= \left\| \hat{x}\left(\frac{f.N}{F_e}\right) \right\| \\ &= \sqrt{\Re\left(\hat{x}\left(\frac{f.N}{F_e}\right)\right)^2 + \Im\left(\hat{x}\left(\frac{f.N}{F_e}\right)\right)^2}\end{aligned}$$

## F. Démodulation FSK

En considérant que les opérations de démodulation FSK sont réalisées par le transceiver VR5000, on propose l'algorithme suivant afin de numériser les amplitudes associées aux composantes fréquentielles d'un signal à partir d'une liste de  $N$  mesures de tensions.

**Proposition. II.3** (ÉCHANTILLONS) *Sachant que l'on souhaite détecter la présence du 1 ou du 0 logique sur un temps  $\Delta t = \frac{1}{F_b}$ . On a :*

$$N = \frac{F_e}{F_b} \quad (2)$$

L'algorithme suivant doit alors être exécuté sur un signal à  $\frac{F_e}{F_b}$  valeurs.

**Idée.** (NUMÉRISATION FSK) On cherche, à partir d'un signal composé de deux amplitudes, à distinguer celle ayant l'occurrence la plus régulière. On peut alors calculer la moyenne du signal et vérifier si cette valeur "se rapproche plus" d'une amplitude ou d'une autre.

On propose un algorithme (ALGORITHME 3, IMPLÉMENTATION 12) qui ne détecte qu'un seul bit d'information par signal  $x$  reçu et qui, à partir de deux amplitudes  $A(f_0)$  et  $A(f_1)$ , renvoie la numérisation de  $x$  comme suit:

$$\text{numerisation\_FSK}(x) = \begin{cases} 0 & \text{si } |\text{moyenne}(x) - A(f_0)| > |\text{moyenne}(x) - A(f_1)| \\ 1 & \text{sinon} \end{cases}$$

**Proposition. II.4** (FORMULE DE LA MOYENNE) *Soient  $N \in \mathbb{N}$  et  $f : \{0; \dots; N-1\} \rightarrow \mathbb{R}$  un signal à  $N$  échantillons. On définit la moyenne du signal  $f$  par la fonction :*

$$\begin{aligned}\mathbb{R}^{\{0; \dots; N-1\}} &\rightarrow \mathbb{R} \\ \text{moyenne} : f &\mapsto \frac{1}{N} \sum_{i=0}^{N-1} f(i)\end{aligned}$$

**Spécification.** Lors de l'écriture des bits de données, la détection de l'amplitude associé à la composante  $f_0$  (combien?) indique le 0 logique tandis que celle associée à  $f_1$  (combien?) indique le 1 [36].

---

### Algorithme 3 Numérisation FSK

---

**Entrées:** signal  $x$  à  $N$  valeurs, amplitude de  $f_0 : A(f_0)$  et amplitude de  $f_1 : A(f_1)$   
1: **moyenne**  $\leftarrow \frac{1}{N} \sum_{i=0}^{N-1} x(i)$   
2: **si**  $|\text{moyenne} - A(f_0)| > |\text{moyenne} - A(f_1)|$  **alors**  
3: **retourner** 0  
4: **sinon**  
5: **retourner** 1  
6: **fini**

---

## G. Démodulation OOK (CW)

La modulation en *On-Off Keying* (OOK) est un cas particulier de la modulation en ASK (Amplitude Shift-Keying). Les données numériques sont modulées par la présence ou bien l'absence du signal porteur  $F_c$ . On note  $A(F_c)$  l'amplitude associée à la composante fréquentielle  $F_c$ . De manière analogue, en considérant que les opérations de démodulation sont effectuées par le VR5000, on propose un algorithme (ALGORITHME 4) similaire de numérisation de données modulées en OOK.

---

### Algorithme 4 Numérisation OOK

---

**Entrées:** signal  $x$  à  $N$  valeurs, amplitude de  $F_c : A(F_c)$   
1: **si**  $\frac{1}{N} \sum_{i=0}^{N-1} x(i) < \frac{A(F_c)}{2}$  **alors**  
2: **retourner** 0  
3: **sinon**  
4: **retourner** 1  
5: **fini**

---

## H. Décodage de trames AX.25 et automates finis

**Définition.** ♣ (LSB–MSB) Soit  $x$  un nombre binaire. Il existe alors une suite finie  $(b_n)_{0 \leq n \leq n_0}$  d'entiers égaux à 0 ou 1 tel que

$$x = \sum_{n=0}^{n_0} b_n \times 2^n$$

L'entier  $b_0$  est appelé *bit de poids faible* (LSB) et l'entier  $b_{n_0}$  le *bit de poids fort* (MSB).



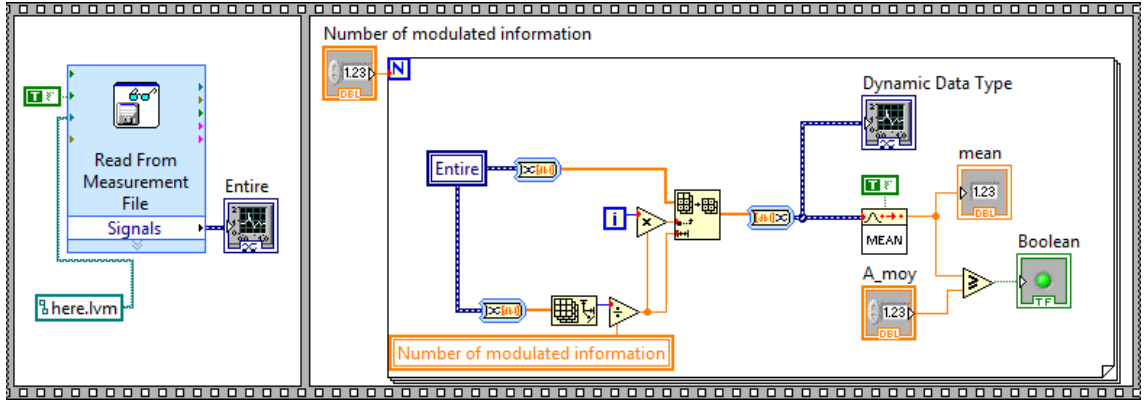


FIG. 12: Implémentation du numériseur d'amplitudes en LabVIEW

**Exemple.** On pose  $x$  le nombre 42 en représentation décimale, on note  $(x)_{\text{bin}}$  la représentation binaire de  $x$  dans la notation positionnelle habituelle, on a :

$$(x)_{\text{bin}} = \underbrace{1}_{\text{MSB}} 0101 \underbrace{0}_{\text{LSB}}$$

**Remarque.** (BOUTISME DE BITS) Tous les octets (mots de longueur 8) d'une trame AX.25 (excepté pour le facteur FCS) sont reçus avec le *bit de poids faible* en premier[3]. Le facteur FCS d'une telle trame est la concaténation d'octets de poids forts en premier.

On note l'alphabet  $\mathbb{B} = \{0, 1\}$ .

**Définition.** (TRAME AX.25 DE PRATHAM) Un mot  $m$  sur l'alphabet  $\mathbb{B}$  est une *trame AX.25 de Pratham* [3] [8] s'il est la concaténation des facteurs suivants (à un facteur *fanion* près):

| $m$ : 114 octets |         |          |          |             |          |          |
|------------------|---------|----------|----------|-------------|----------|----------|
| Fanion           | Adresse | Contrôle | PID      | Information | FCS      | Fanion   |
| 01111110         | ...     | 00000011 | 11110000 | ...         | 2 octets | 01111110 |

| Adresse : 21 octets       |                         |                           |                           |          |                                    |
|---------------------------|-------------------------|---------------------------|---------------------------|----------|------------------------------------|
| Sous-adresse <sub>1</sub> |                         | Sous-adresse <sub>2</sub> | Sous-adresse <sub>3</sub> |          |                                    |
| 7 octets                  |                         | 7 octets                  | 7 octets                  |          |                                    |
| CQ                        | $((20)_{\text{hex}})^4$ | 01100000                  | VU2DMQ                    | 01100000 | RELAY $(20)_{\text{hex}}$ 01100000 |

| Information : 87 octets   |
|---------------------------|
| Health monitoring data[3] |

**Théorème. II.5** ♣ (KLEENE) Soit un alphabet fini  $\Sigma$ .

$$\text{Rat } \Sigma^* = \text{Rec } \Sigma^*$$

**Proposition. II.6** ♣ Soit  $e$  une expression rationnelle dénotant un langage  $L(e)$  sur l'alphabet  $\mathbb{B}$  tel que :

$$\exists u \in \mathbb{B}^* \forall w \subset u \begin{cases} e = (01111110).u.(01111110)^* \\ w \neq 01111110 \end{cases} \quad (3)$$

Il existe un automate fini  $A$  tel que  $A$  reconnaisse  $L(e)$ .

**Remarque.** (SYNTAXE DES TRAMES PRATHAM EN AX.25) La syntaxe des trames du protocole AX.25 est présentée en FIGURE 13 afin d'implémenter un analyseur syntaxique.

**Implémentation.** (ALGORITHME 5) ♣ On procède à une analyse lexicale sur les bits de données, d'où l'intérêt et l'usage d'automates finis, afin de découper les trames séparées par des fanions. On implémente un analyseur lexical munie d'une file[?] [29] avec *ocamllex* [30].

#### Algorithme 5 Découpage de trames AX.25

**Entrées:** mot  $m = m_0 \dots m_{p-1}$  sur l'alphabet  $\mathbb{B}$  de longueur  $p$

- 1: mot vide  $a$  //accumulateur indiquant l'ouverture ou non d'une nouvelle trame
- 2: liste vide  $l$
- 3: **analyse lexicale** de  $m$  **faire**
- 4: **parse** 01111110 :
- 5: **si** une trame n'est pas ouverte **alors**
- 6: ouvrir une nouvelle
- 7: **sinon**
- 8: **si** une trame est déjà ouverte, **alors**
- 9: **si** elle n'est pas vide **alors**
- 10: la refermer et l'enfiler dans  $l$
- 11: **finsi**
- 12: **finsi**
- 13: **finsi**
- 14: **parse** \_ :
- 15: **si** une trame est déjà ouverte **alors**
- 16: concaténer le bit \_ dans la trame
- 17: **sinon**
- 18: ne rien faire //bits perdus car aucune trame ouverte
- 19: **finsi**
- 20: **fin analyse lexicale**
- 21: **retourner** liste  $l$  de mots binaires dont chaque élément est une trame

**Remarque.** (BOURRAGE DE BIT) Le fanion 01111110 (7E hexadécimal) ne peut donc, en aucun cas, avoir d'occurrence dans de trames complètes[3]. Le procédé

## AX.25 packets protocol

```

-----
packets      ::= { frame }
frame        ::= flag+ address control pid information fcs flag
flag         ::= "01111110"
address      ::= "CQ" ((20)_hex)^4 "01100000" "VU2DMQ" "01100000" "RELAY" (20)_hex "01100000" (* retranscrire *)
control      ::= "00000011"
pid          ::= "11110000"
information  ::= (* pratham *)
fcs          ::= (* attention au boutisme *)

```

FIG. 13: Syntaxe des trames Pratham en AX.25 en notation EBNF

de *bouillage de bit* (*bit stuffing*) permet alors d'éviter la confusion entre les données et les fanions. Durant la réception des données, on ignore le bit 0 à chaque occurrence de cinq bits 1 consécutifs, après les opérations de découpages de trames.

**Implémentation.** (ALGORITHME 6) 🚀 Après le découpage des trames, on extrait, de chaque trame, les bits de bouillage.

---

**Algorithme 6** Extraction du bit de bouillage

---

**Entrées:** mot  $m = m_0 \dots m_{p-1}$  sur l'alphabet  $\mathbb{B}$  de longueur  $p$ ,  $\cdot\cdot$  la concaténation de deux mots

- 1: mot vide  $m'$
- 2: compteur  $c \leftarrow 0$
- 3: **pour**  $i$  de 0 à  $p$  **faire**
- 4: **si**  $m_i = 0$  **alors**
- 5: **si**  $c = 5$  **alors**
- 6:  $c \leftarrow 0$
- 7: **sinon**
- 8:  $c \leftarrow 0$
- 9:  $m' \leftarrow m'.0$
- 10: **finsi**
- 11: **sinon**
- 12:  $c \leftarrow c + 1$
- 13:  $m' \leftarrow m'.1$
- 14: **finsi**
- 15: **fin pour**
- 16: **retourner**  $m'$

---

**Implémentation.** (ALGORITHME 7) 🚀 En considérant que chaque unité de trame, n'a pas le même boutisme de bits, on peut alors découper les unités **Adresse**, **Contrôle**, **PID**, **Information**, **FCS** et calculer leur représentation suivant la spécification établie.

---

**Algorithme 7** Pattern-matching on AX.25 packets

---

**Entrées:** liste de mots  $l$

---

## I. Décodage du Morse

**Liaison.** Simplex (définition ANSI)

**Spécification.** La spécification du code Morse en vigueur dans le cadre du projet est celle décrite par l'International Telecommunication Union [13]. On en rappelle ci-dessous la description :

1. La lettre DAH (–) dure trois fois plus longtemps que la lettre DIT (·) ;
2. L'écart entre deux éléments (DIT ou DAH) d'une même lettre dure un DIT ;
3. L'écart entre deux lettres d'alphabet dure un DAH ;
4. L'écart entre deux mots dure sept DIT.

**Définition.** On note  $\mathbb{M}$  l'alphabet Morse. On a :

$$\mathbb{M} = \{\varepsilon_1, \varepsilon_3, \varepsilon_7, \cdot, -\}$$

**Définition.** On définit par cas sur  $\mathbb{M}$  l'application :

$$h : \mathbb{M} \rightarrow \mathbb{B}^*$$

$$m \mapsto \begin{cases} 0 & \text{si } m = \varepsilon_1 \\ 000 & \text{si } m = \varepsilon_3 \\ 0000000 & \text{si } m = \varepsilon_7 \\ 1 & \text{si } m = \cdot \\ 111 & \text{si } m = - \end{cases}$$

**Définition.** On définit sur  $\mathbb{M}$  la longueur d'une lettre  $x \in \mathbb{M}$  comme étant l'application :

$$L : \mathbb{M} \rightarrow \mathbb{N}$$

$$x \mapsto \begin{cases} 1 & \text{si } x \in \{., \varepsilon_1\} \\ 3 & \text{si } x \in \{-, \varepsilon_3\} \\ 7 & \text{si } x = \varepsilon_7 \end{cases}$$

**Proposition. II.7** (NORME SUR  $\mathbb{M}^*$ ) Soit la fermeture de Kleene [39] de l'alphabet  $\mathbb{M}$ , notée  $\mathbb{M}^*$ . L'application

$$\tilde{N} : \mathbb{M}^* \rightarrow \mathbb{N}$$

$$x = (x_0 \dots x_{n-1}) \mapsto \sum_{i=0}^{n-1} N(x_i)$$

est une norme sur  $\mathbb{M}^*$ .

**Démonstration.** Soit un mot  $x \in \mathbb{M}^*$ . Il existe une suite de lettres de  $\mathbb{M}$  ( $x_1 \dots x_n$ ) telle que

$$x = (x_1 \dots x_n)$$

## 1. (SÉPARATION)

$$\forall x \in \mathbb{M}^*, \tilde{N}(x) = 0 \Rightarrow x = \varepsilon$$

## 2. (HOMOGÉNÉITÉ)

$$\forall(\lambda, x) \in \mathbb{N} \times \mathbb{M}^*, \tilde{N}(x^\lambda) = \lambda \tilde{N}(x)$$

## 3. (INÉGALITÉ TRIANGULAIRE)

$$\forall(x, x') \in \mathbb{M}^*, \tilde{N}(x.x') \leq \tilde{N}(x) + \tilde{N}(x')$$

**Implémentation.** On implémente tout d'abord un transducteur fini qui sur l'alphabet  $\mathbb{B} = \cdot, -$  reconnaît un mot sur  $\mathbb{B}$  et retourne la lettre associée à son codage en Morse. Le transducteur est donné en FIGURE 16.

**Implémentation.** On termine l'implémentation avec une analyse lexicale sur les bits issus de la numérisation du signal à 145,980 MHz (ALGORITHME 8).

---

**Algorithme 8** Décodage Morse à partir du signal modulant (numérique)

---

**Entrées:** mot  $m = m_0 \dots m_{p-1}$  sur l'alphabet  $\mathbb{B}$  de longueur  $p$ , transducteur  $T_{\text{Morse}}$  muni de **eval()**  
 1: chaîne de caractères  $acc$ , et  $res$   
 2: **analyse lexicale** de  $m$  **faire**  
 3: **parse** 0000000 :  $(\ast \varepsilon_7 \ast)$   
 4:  $res \leftarrow res. \text{' ' ' '}$   
 5:  $acc \leftarrow \varepsilon$   
 6: **parse** 000 :  $(\ast \varepsilon_3 \ast)$   
 7:  $res \rightarrow res. \text{eval}(T_{\text{Morse}}, acc)$   
 8:  $acc \leftarrow \varepsilon$   
 9: **parse** 0 :  $(\ast \varepsilon_1 \ast)$   
 10:  $acc \leftarrow \varepsilon$   
 11: **parse** 111 :  $(\ast - \ast)$   
 12:  $acc \leftarrow acc.-$   
 13: **parse** 1 :  $(\ast . \ast)$   
 14:  $acc \leftarrow acc.$   
 15: **fin analyse lexicale**  
 16: **retourner**  $res$

---

## J. Précision de l'acquisition

Le boîtier d'acquisition dispose d'un convertisseur analogique-numérique d'une résolution de 16 bits. La résolution  $Q$  du CAN ou *tension du bit de poids faible* (LSB) est donnée par la relation

$$Q = \frac{E_{\text{FSR}}}{N} = \frac{V_{\text{RefHi}} - V_{\text{RefLow}}}{2^M}$$

où :

$E_{\text{FSR}}$  : intervalle de tensions calibrés [V]

$N$  : nombre d'intervalles de tensions [1]

$V_{\text{RefHi}}$  : calibre maximal de mesure de tension [V]

$V_{\text{RefLow}}$  : calibre minimal de mesure de tension [V]

$M$  : résolution du CAN [bit]

La spécification du boîtier d'acquisition renseigne sur la précision absolue [16] en page "AI Absolute Accuracy Table".

On a alors déterminé que nombre de chiffres significatifs de cet appareil de mesure s'élève à 12 digits

## K. Enregistrement et sauvegarde des données

On réalise une estimation des coûts en espace que peuvent engendrer l'enregistrement des valeurs de mesures au format de fichier Pratham en convention avec l'IITB[4]. La durée d'acquisition lors d'un passage de satellite au-dessus de Paris dépend de plusieurs paramètres, tels que :

- l'altitude du satellite ;
- le système de tracking (NOVA) ;
- l'ouverture de l'antenne ;
- le multitrajet à basse élévation.

On se propose d'effectuer une première estimation sur les fichiers de données brutes par intervalles de données en portant à considération que 12 digits suffiront à la représentation des nombres à virgule flottante.

**Proposition. II.8** (COMPLEXITÉ EN ESPACE D'UN FLOTTANT) *En considérant que la représentation d'un nombre à virgule flottante en ASCII s'effectue sur 12 caractères et qu'un caractère en encodage ASCII a un coût en espace d'un octet. Alors un nombre à virgule flottante a un coût en espace de 12 octets.*

**Remarque.** La représentation en machine 32-bits[31] suivant les conventions ANSI-C d'un nombre signé à virgule flottante en précision simple ne coûte que 4 octets[33], soit trois fois moins que la représentation précédente.

**Proposition. II.9** (COMPLEXITÉ EN ESPACE DES ENREGISTREMENTS DE DONNÉES) *Soit  $(\mathbb{B}^*, l.)$  un espace normé. On a alors l'approximation suivante de la longueur d'un enregistrement d'une séquence d'acquisition :*

$$l_{\text{fichier}} = \underbrace{\Delta t}_{\in [7;13] \text{ min}} \times \underbrace{f_e}_{10 \text{ kHz}} \times \underbrace{\left( \underbrace{l_{\text{date}}}_{23 \text{ octets}} + \underbrace{n_{\text{voies}}}_{12} \times \left( \underbrace{l_{\text{flottant}}}_{12 \text{ octets}} + \underbrace{l_{\text{separateur}}}_{1 \text{ octet}} \right) \right)}_{\text{une ligne}} \quad (4)$$

$$\in [7,518 \times 10^8; 1,3962 \times 10^9] \quad (5)$$

car  $\Delta t \mapsto l_{\text{fichier}}(\Delta t)$  croissante



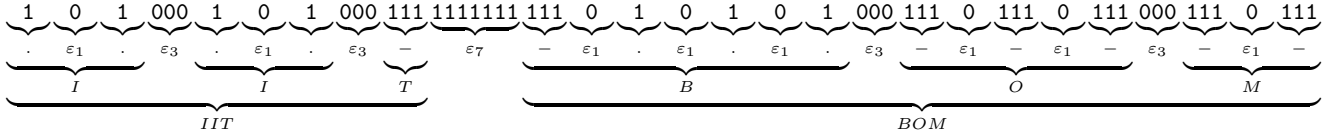


FIG. 14: Exemple des couches d'abstraction d'un signal Morse codant l'information ‘‘IIT Bombay’’

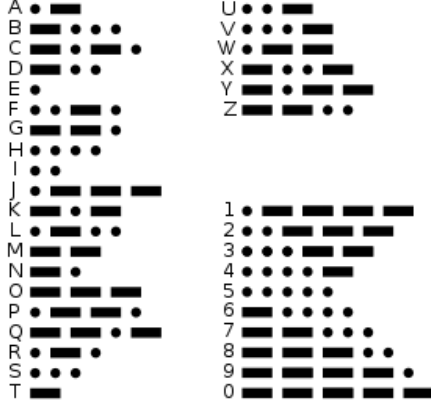


FIG. 15: Alphabet du code Morse

En se basant sur une estimation du nombre de passages du satellite au dessus de Paris[5] :

$$n_{\text{passages/jour}} \in [4; 6] \text{ passages} \quad (6)$$

On a alors par jour :

$$l_{\text{jour}} = n_{\text{passages/jour}} \times l_{\text{fichier}} \quad (7)$$

$$\in [3,0072 \times 10^9; 8,3772 \times 10^9] \quad (8)$$

car  $n_{\text{passages/jour}} \mapsto l_{\text{jour}}(n_{\text{passages/jour}})$  croissante

Sur un fonctionnement supposé de 4 mois[2] du satellite, il est nécessaire de disposer d'un espace de stockage d'au moins :

$$l_{\text{total}} = n_{\text{passages/jour}} \times 30,5 \times 4 \times l_{\text{fichier}} \quad (9)$$

$$\in [3,668784 \times 10^{11}; 1,0220184 \times 10^{12}] \quad (10)$$

On présente en TABLE II, les estimations de taille de fichier concernant les données brutes la sortie de l'acquisition.

En considérant que la fréquence d'échantillonnage  $F_e$  est de 10 kHz, on obtient un instant de mesure

$$T_e \stackrel{\text{def}}{=} \frac{1}{F_e}$$

Par passage de satellite, on n'obtient que 7200000 valeurs pour les slanTEC (calcul à justifier).

TABLE II: Estimations minimales et maximales de l'espace de stockage nécessaire aux données brutes

| Durée de temps (s) | Taille des fichiers |          |
|--------------------|---------------------|----------|
|                    | MIN                 | MAX      |
| 1 min              | 107,4 Mo            |          |
| 1 seq. d'acq.      | 751,8 Mo            | 1,4 Go   |
| 1 jour             | 3,0 Go              | 8,37 Go  |
| 1 mois             | 91,7 Go             | 255,5 Go |
| 4 mois             | 366,8 Go            | 1,02 To  |

Dans le cadre d'une solution provisoire, on se permet d'utiliser le format de fichier de ‘‘LabVIEW Measurement File’’ [24] (solution à problème car enregistre avec la date d'enregistrement)

#### L. Tolérance aux fautes

Le système de tolérance aux fautes se limite qu'à une gestion des exceptions par LabVIEW qui lors d'une exception poursuit l'acquisition et effectue un rapport.

#### M. Logiciel final

Le logiciel final sera compilé avec NI Application Builder [25] [26].

### III. FORMATS DE FICHIER

#### A. Élévations minimales/maximales fournies par Nova

Nous donnons ci-dessous la syntaxe en EBNF des fichiers de données sur les élévations minimales et maximales d'un satellite par rapport à un référentiel géographique.

LabVIEW, étant langage de programmation peu adapté à certains paradigmes, ne dispose pas de générateur d'analyseur syntaxique, ni lexical mais d'un analyseur par formats. Nous implémentons provisoirement une analyse formatée avec la fonction `Scan From String` [22] (*analyseur par formats d'une chaîne de*

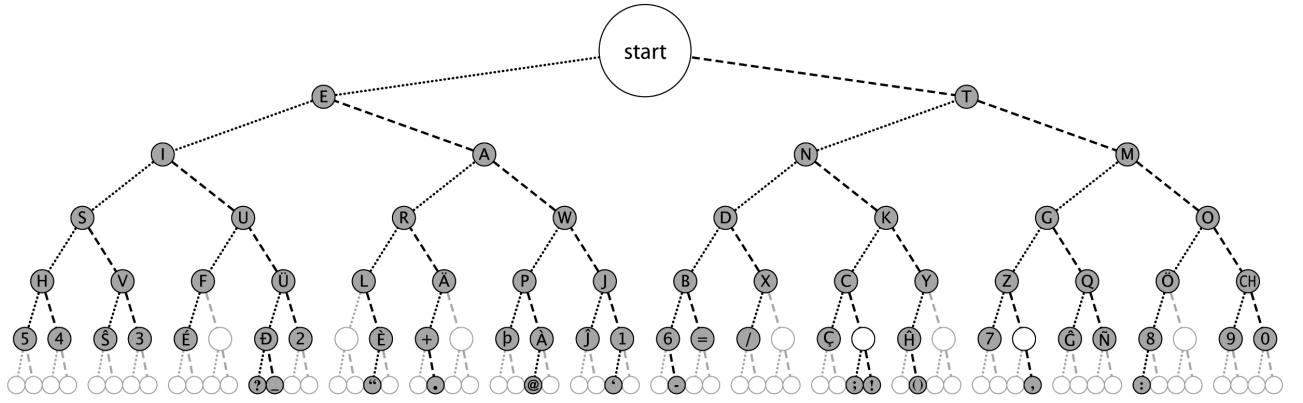


FIG. 16: Transducteur fini associé au décodage MORSE [38]

caractères à l'aide d'indicateurs de conversion). On se propose de donner une preuve de correction de l'algorithme utilisant cet analyseur (ALGORITHME 9, IMPLÉMENTATION 17) [29].

---

**Algorithme 9** Analyse formatée des fichiers d'éphémérides de Nova

---

**Entrées:**  $F$  : fichier  
 1: hello world  
 2: **retourner**  $s$

---

$\text{FOLLOW}(F) = \emptyset$   
 $\text{FOLLOW}(S) = \{f\}$   
 $\text{FOLLOW}(D) = \{\$ \}$   
 $\subseteq \text{FOLLOW}(D')$  car  $\epsilon \in \text{FIRST}(D')$   
 $\text{FOLLOW}(D') = \{\$ \}$

Comme montré en 21, la grammaire ne présente aucun conflit FIRST/FIRST dans le cas d'une analyse  $LL(1)$ .  
 Donc :

### B. Enregistrements Pratham

Les formats de fichier qui seront utilisées pour l'enregistrement des données de TEC vertical et slant sont spécifiées dans le document ??? dont on donne une syntaxe en notation EBNF en 20.

Par souci de rigueur, on s'intéresse à la classe de grammaire à laquelle appartient la grammaire  $G$  représentant cette syntaxe.

#### Proposition. III.1

$$G \in LL(1)$$

**Démonstration.** Montrons que la grammaire est  $LL(1)$ . Soit la grammaire  $G$  dont le symbole de départ est  $F$  et dont on donne les ensembles de lexèmes en figure 21.  $D'$  est annulable.

On calcule  $\text{FIRST}()$  et  $\text{FOLLOW}()$  :

$$\begin{aligned} \text{FIRST}(F) &= \{SI\} \\ \text{FIRST}(D) &= \{n\} \\ \text{FIRST}(L) &= \{n\} \\ \text{FIRST}(L') &= \{n, \epsilon\} \end{aligned}$$

$$G \in LL(1)$$

□

## IV. BANC D'ESSAI

### V. PREUVES DE PROGRAMMES

♣ Cette partie donne les preuves de correction des algorithmes mentionnés dans l'article. La méthode formelle utilisée sera la *logique de Hoare*[10] (1969) et son cas particulier la preuve par *invariant de boucle*.

**Définition.** ♣ (TRIPLÉT DE HOARE) Soient  $P, Q$  des prédicats et **prog** un programme. Un triplet  $T$  est un *triplet de Hoare* si et seulement si

$$T = \{P\} \text{ prog } \{Q\}$$

**Définition.** ♣ (CORRECTION D'UN PROGRAMME) Un triplet de Hoare  $\{P\} \text{ prog } \{Q\}$  est *vrai* si pour tout état initial vérifiant  $P$ , si l'exécution de **prog** termine, alors  $Q$  est vraie après l'exécution de **prog**. Le programme **prog** est dit *correct* par rapport à  $P$  et  $Q$ .

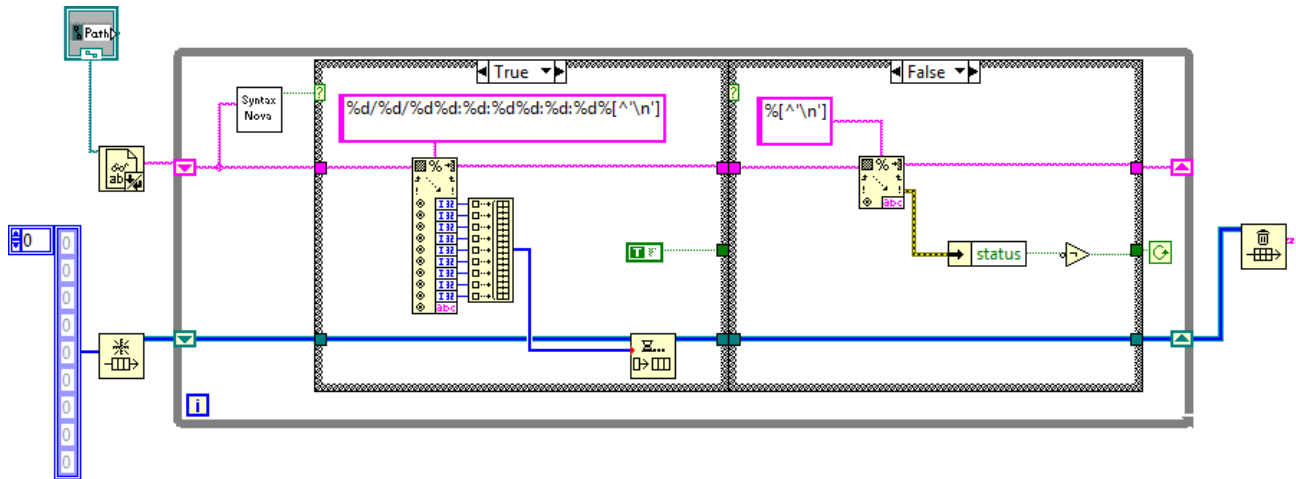


FIG. 17: Implémentation d'une analyse formatée en LabVIEW

Nova satellite AOS/LOS file

-----

```
file      ::= columns_desc { day_eclipses }
```

```
columns_desc ::= "Date (Z)" "AOS (Z)" "LOS (Z)" "Duration" "Between Az" '@' "AOS Max El Az" '@' "LOS Height" "km" n
```

```
day_eclipses ::= day_header { day_entry }
```

```
day_header  ::= { character } "at" { character } ',' { character } newline
```

```
day_entry   ::= date time time time time int_number ' ' int_number ' ' int_number ' ' float_number newline
```

```
date        ::= int_number '/' int_number '/' int_number
```

```
time        ::= int_number ':' int_number ':' int_number
```

```
newline     ::= \\n
```

```
character   ::= 'a'|'b'|...|'A'|'B'|...|'0'|'1'|..
```

```
int_number  ::= [ '-' | '+' ] { '0' | .. | '9' }
```

```
float_number ::= [ '-' | '+' ] { '0' | ... | '9' } '.' { '0' | ... | '9' }
```

FIG. 18: Syntaxe des fichiers d'élévations minimales/maximales de Nova en notation EBNF

#### A. Algorithme 1 : Minuterie

**Proposition. V.1** *L'algorithme déclenche une suite de séquences d'acquisition  $s_i$  à des temps pour chaque  $s_i$ .*

**Démonstration.**

#### B. Algorithme 2 : Boucle d'acquisition

**Proposition. V.2** *L'algorithme boucle sur des acquisitions sur un temps fini  $\Delta t$ .*

**Démonstration.** Clair.

#### C. Algorithme 3 : Numérisation FSK

#### D. Algorithme 4 : Numérisation OOK

#### E. Algorithme 5 : Découpage de trames AX.25

#### F. Algorithme 6 : Extraction du bit de bourrage

#### G. Algorithme 7 : Pattern-matching on AX.25 packets

#### H. Informations sur la machine d'acquisition

#### I. Caractérisation du boîtier d'acquisition

### VI. INTÉGRITÉ ET MISE EN ROUTE MACHINE

#### A. Informations sur la machine

Les logiciels suivants doivent être installés et éventuellement correctement configurés dans l'ordre[14]

|           | Date (Z)         | AOS (Z)  | LOS (Z)  | Duration | Between Az | @ AOS | Max El | Az @ LOS | Height km |
|-----------|------------------|----------|----------|----------|------------|-------|--------|----------|-----------|
| ITUPSAT,1 | at Paris, France |          |          |          |            |       |        |          |           |
|           | 16/04/12         | 10:55:23 | 11:05:03 | 00:09:40 | 01:01:01   | 38    | 7      | 126      | 714.6     |
|           | 16/04/12         | 12:32:01 | 12:46:13 | 00:14:12 | 01:26:57   | 17    | 58     | 185      | 714.6     |
|           | 16/04/12         | 14:10:15 | 14:23:00 | 00:12:43 | 01:24:02   | 4     | 22     | 236      | 714.7     |

FIG. 19: Exemple de fichier de prévisions de passages de satellite produit par Nova

Raw output file from ground station acquisition (RAW\_PRAT\_SSSS\_yyyy\_ddd\_hh\_mm\_ss.txt)

```

-----
file ::= "Station_ID"          word
      "Location"              int_number ':' int_number ':' float_number int_number ':' int_number ':' float_number int_number
      "Satellite_Tracking_ID" word
      "Start_time_UT"         time_stamp
      "End_time_UT"           time_stamp
      "Sampling_rate"         int_number
      "Data_points"           float_number
      "Acquisition_type"      int_number
      "Time" "145_VMAG1" "145_VPHS1" "145_VMAG2" "145_VPHS2" "437_VMAG1" "437_VPHS1" "437_VMAG2" "437_VPHS2"
      { time_stamp float_number float_number float_number float_number float_number float_number float_number float_number }

time_stamp ::= int_number ':' int_number ':' int_number ':' int_number ':' int_number

word ::= { character }
character ::= 'a' | 'b' | ... | 'z' | 'A' | 'B' | ... | 'Z' | '0' | '1' | ... | '9'
int_number ::= [ '-' | '+' ] { '0' | ... | '9' }
float_number ::= [ '-' | '+' ] { '0' | ... | '9' } '.' { '0' | ... | '9' }

```

FIG. 20: Syntaxe des enregistrements au format Pratham en notation EBNF pour les données brutes

TABLE III: Force, area, and pressure data for the experiment shown in Fig. 9 and described by Eq. ???. Agreement is typically within five percent.

|                            | Piston 1 | Piston 2 |
|----------------------------|----------|----------|
| Avg. Force (N)             | 4.40     | 2.25     |
| Area (cm <sup>2</sup> )    | 6.16     | 2.25     |
| $F/A$ (N/cm <sup>2</sup> ) | 0.714    | 0.717    |

suivant:

1. LabVIEW 8.6 (National Instruments)
2. NI DAQ-mx 9.2.3 (pilote du boîtier d'acquisition)
3. Nova for Windows 2.2c (NLSA)
4. OCaml 3.12 pour Windows

Il sera prévu que la machine d'acquisition – prévue pour fonctionner 24h/24 durant 4 mois – sera de type :

- Modèle : Dell Latitude ATG D630
- Processeur : Intel Core 2 Duo T7500 (2,2 GHz, 2,19 GHz)

- Système d'exploitation : Microsoft Windows XP
- Disque dur : Toshiba MK8046GSX (74,3 Go dont 71,5 Go occupés) (rapidité d'écriture ?)
- Carte réseau :
  - LAN: Broadcom NetXtreme 57xx Gigabit Controller
  - Wifi: Intel PRO/Wireless 3945 ABG Network Connection

DELL LATITUDE ATG D630.

Elle sera constamment connectée sur le réseau afin de permettre à l'équipe de surveiller et de rapatrier les données acquises.

## B. Système d'avertissement et de gestion des erreurs

## VII. CONCLUSION

[1] Commissariat l'Energie Atomique, Centre National de la Recherche Scientifique, Institut National de Recherche en

$$\begin{aligned} \text{Term} &= \{f, n, m, \$, SI, L, STID, STU, ETU, SR, DP, AT, T, 145_0, 145_1, 145_2, 145_3, 437_0, 437_1, 437_2, 437_3\} \\ \text{NonTerm} &= \{F, S, D, D'\} \end{aligned}$$

$$\begin{aligned} F &\rightarrow SI\ m\ L\ n : n : f\ n : n : n : f\ n\ STI\ m\ STU\ n : n : n : n : n\ ETU\ n : n : n : n : n\ SR\ n\ DP\ f\ AT\ n \\ &\quad T\ 145_0\ 145_1\ 145_2\ 145_3\ 437_0\ 437_1\ 437_2\ 437_3\ D\ \$ \\ S &\rightarrow n : n : n : n : n \\ D &\rightarrow S\ f\ f\ f\ f\ f\ f\ f\ f\ D' \\ D' &\rightarrow \epsilon | S\ f\ f\ f\ f\ f\ f\ f\ f\ D' \end{aligned}$$

|    | SI        | L | STI | STU | SR | DP | AT | T | 145 <sub>0</sub> | 145 <sub>1</sub> | 145 <sub>2</sub> | 145 <sub>3</sub> | 437 <sub>0</sub> | 437 <sub>1</sub> | 437 <sub>2</sub> | 437 <sub>3</sub> | f | m | n            | : | \$       |
|----|-----------|---|-----|-----|----|----|----|---|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|---|---|--------------|---|----------|
| F  | {F → ...} | ∅ | ∅   | ∅   | ∅  | ∅  | ∅  | ∅ | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅ | ∅ | ∅            | ∅ | ∅        |
| S  | ∅         | ∅ | ∅   | ∅   | ∅  | ∅  | ∅  | ∅ | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅ | ∅ | {S → ...}    | ∅ | ∅        |
| D  | ∅         | ∅ | ∅   | ∅   | ∅  | ∅  | ∅  | ∅ | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅ | ∅ | {L → ...}    | ∅ | ∅        |
| D' | ∅         | ∅ | ∅   | ∅   | ∅  | ∅  | ∅  | ∅ | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅                | ∅ | ∅ | {L' → n ...} | ∅ | {L' → ε} |

FIG. 21: Table d'analyse LL(1) de la grammaire

- License Agreement* (September, 2006)
- [2] Saptarshi Bandyopadhyay, Jhonny Jha, Haripriya, Ameya Damle, Deepika Thakur, Sanyam Mulay, Prashant Sachdeva, Jaideep Joshi, Vaibhav Unhelkar, Yashovardhan Chati, Mayank Chaturvedi, Niranjan Parab, Manas Rachh, Shashank Tamaskar, Mallesh Bommanahal, Ashish Goel, Kartavya Neema, Subhasis Das, Vishnu Sresht, Ramnath Pai, Ankit Chiplunkar, *Introduction to Pratham, IIT Bombays Student Satellite Project* (IIT Bombay, June 2010)
  - [3] J. Jha, S. Bandyopadhyay, C. Talnikar, *Pratham - Telemetry and Telecommand Document* (IIT Bombay, November 2010)
  - [4] H. Nguyen Van, P. Coisson, P. Godbole, J. Jha, *Pratham satellite : File Formats Specification* (IPGP, May 2011)
  - [5] L. Viens, P. Coisson, *Pratham satellite : Paris Ground Station Simulations* (IPGP, November 2010)
  - [6] J. Senlis, *Formation DSP sur 320F28335* (INSSET, Université de Picardie, 2008)
  - [7] T. Capitaine, M. Hamzaoui, A. Lorthois, J. Senlis, *Démodulation et décodage de trames AX.25 par DSPIC pour la localisation d'un ballon sonde météo dans le cadre d'une action "Planète Sciences"* (CETSIS, Bruxelles, 2008)
  - [8] William A. Beech, Douglas E. Nielsen, Jack Taylor, *AX.25 Link Access Protocol for Amateur Packet Radio, version 2.2* (Tucson Amateur Packet Radio Corporation, July 2008)
  - [9] Michael R. Owen, *Nova for Windows – User's Manual* (Northern Lights Software Associates, February 2000)
  - [10] C. A. R. Hoare, *An axiomatic basis for computer programming* (Communications of the ACM 12 (10): 576-580)
  - [11] Agnès Foucher, Christian Valade, *Modulation, Démodulation, FSK. Étude structurelle* (Université de Toulouse 1, Février 1999)
  - [12] International Organization for Standardization, *Information technology – Syntactic metalanguage – Extended BNF* (ISO/IEC 141977, December 1996)
  - [13] International Telecommunication Union, *International Morse code – Recommendation ITU-R M.1677-1* (ITU, 2009)
  - [14] National Instruments, *Install NI LabVIEW and NI-DAQmx Driver* (2010)
  - [15] National Instruments, *B/E/M/S/X Series Calibration Procedure* (370937K-01, August 2010)
  - [16] National Instruments, *NI 6351/6353 Specifications* (370787B-01, August 2010)
  - [17] National Instruments, *Creating a Typical DAQ Application* (371361D-01, August 2007)
  - [18] National Instruments, *Data Acquisition Reference Design for LabVIEW* (July 2010)
  - [19] National Instruments, *When Should I Use Continuous or Finite Sampling Modes?* (3L8BGMXL, May 2005)
  - [20] National Instruments, *Extending Virtual Memory Usage for 32-bit Windows* (371361D-01, August 2007)
  - [21] National Instruments, *Request Deallocation* (371361D-01, August 2007)
  - [22] National Instruments, *Scan From String* (371361D-01, August 2007)
  - [23] National Instruments, *Express VIs, Dynamic Data Type* (371361D-01, August 2007)
  - [24] National Instruments, *Specification for the LabVIEW Measurement File (.lvm)* (July 2010)
  - [25] National Instruments, *NI LabVIEW Compiler: Under the Hood* (July 2010)
  - [26] National Instruments, *Creating Executables with the LabVIEW Application Builder* (March 2010)
  - [27] Texas Instruments, *FSK Modulation and Demodulation With the MSP430 Microcontroller* (SLAA037, December 1998)
  - [28] Windows Developer Center, *Memory Limits for Windows Releases* (Microsoft Developer Network, 2009)
  - [29] Donald Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms, Third Edition*, Section 2.2.1: Stacks, Queues, and Deques, pp. 238–243. (Addison-Wesley, 1997)
  - [30] Xavier Leroy, Damien Doligez, Alain Frisch, Jacques

- Garrigue, Didier Rémy and Jérôme Vouillon, *The OCaml system, release 3.12*, Chapter 12: Lexer and parser generators (ocamllex, ocaml yacc) (INRIA, 2011)
- [31] “A microprocessor with a “wordlength equal to  $n$ ” is also referred to as an  $n$ -bit microprocessor defined as a processor with  $n$ -bit wide internal data registers and  $n$ -bit wide ALU (arithmetic logic unit) which carries out operations on  $n$ -bit input operands.”, N. Alexandridis in *Computer Systems Architecture: Microprocessor-Based Designs* (The George Washington University, May 1999)
- [32] Virgínio de Oliveira Sannibale, *Measurements and Significant Figures* (California Institute of Technology, October 2011)
- [33] D. Marshall, *C Basics* (Cardiff University, May 1999)
- [34] H. Nyquist, *Certain topics in telegraph transmission theory*, Trans. AIEE, vol. 47, pp. 617644, Apr. 1928. Reprint as classic paper in: Proc. IEEE, Vol. 90, No. 2, Feb 2002.
- [35] C. E. Shannon, *Communication in the presence of noise*, Proc. Institute of Radio Engineers, vol. 37, no.1, pp. 1021, Jan. 1949. Reprint as classic paper in: Proc. IEEE, Vol. 86, No. 2, (Feb 1998).
- [36] Jhonny Jha : “f0 would mean 1 and f1 would mean 0 is what I intended to say”
- [37] “ $f(n) \in \Omega(g(n)) \stackrel{\text{Landau}}{\Leftrightarrow} \exists k > 0, n_0 \forall n > n_0 g(n) \cdot k \leq |f(n)|$ ”, P. E. Black in *Dictionary of Algorithms and Data Structures* (U.S. National Institute of Standards and Technology, December 2004)
- [38] Aris00, *A binary tree of the Morse Code adapted from the dichotomic search table in the Morse code Wikipedia entry* (Wikimedia Commons, CC-BY-SA-2.5,2.0,1.0; GFDL-WITH-DISCLAIMERS)
- [39] Jean-Michel Autebert, *Théorie des langages et des automates* (Dunod, December 1997)
- [40] Godred Fairhurst, *Internet Communications Engineering - A Tutorial* (University of Aberdeen, October 2001)