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: September 1, 2012)

I. INTRODUCTION

The following application note introduces technical issues related to the control-data-acquistion 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. General overview

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
			$AD8302_{mag}$ →	
			$AD8302_{phase}$ →	
	Plane 2	...	VR5000 →	
			$AD8302_{mag}$ →	
			$AD8302_{phase}$ →	
437,455 MHz antenna	Plane 1	...	VR5000 →	
			$AD8302_{mag}$ →	
			$AD8302_{phase}$ →	
	Plane 2	...	VR5000 →	
			$AD8302_{mag}$ →	
			$AD8302_{phase}$ →	

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.

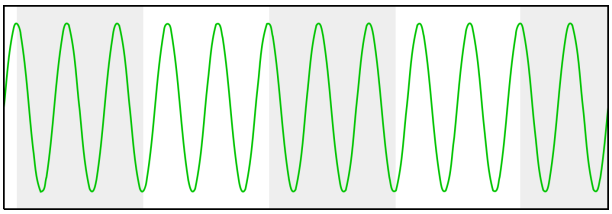

FIG. 1: Carrier signal (central frequency F_c)

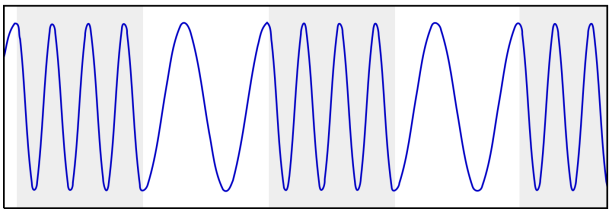

FIG. 2: Modulated signal in FSK at $F_c \pm \frac{\Delta f}{2}$ frequencies received by the `VR5000`

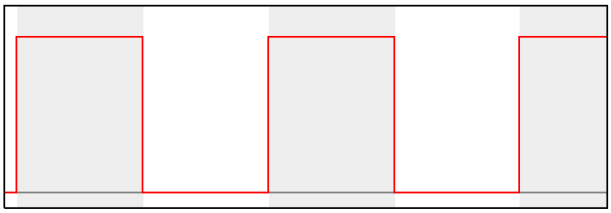

FIG. 3: Modulating signal at $F_b = 1,2$ kbit.s^{-1} bit rate (retrieved by FSK demodulation)

The satellite sends a signal with a declared carrier signal at $F_c = 437,455$ 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 Δf (FIGURE 2).

The transceiver returns an audio signal. It finally decribes – 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 (FIGURE 4) in such a way that (more info on peak sync ? method not sure...) on a time length T_b, the highest mean amplitude of a frequency would match its logical `0` or `1`.

FIG. 4: Data bits associated to the modulating signal

We decode data bits from the AX.25 data link layer protocol (FIGURE 5).

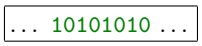

FIG. 5: Data bits decoded from `AX.25` packets

At the end of the process we evaluate data bits as defined by IITB's specification[3].

C. Digital signal at 145,980 MHz : Beacon

In the same way the transceiver demodulates a signal of central frequency $F_c = 145,980$ MHz (FIGURE 6). Considering a bit rate of $F_b = 10$ bit.s^{-1}, the presence of the amplitude associated to the frequency component F_c matches with a pulse and the abscence with no pulse as specified by ON-OFF KEYING (FIGURE 7).

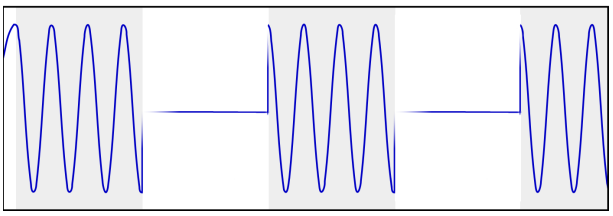

FIG. 6: Modulated signal in OOK of fréquency F_c received by the transceiver

We demodulate the digital baseband transmission method considering that a logical `1` matches with a pulse of a time length that equals to a DIT with DITS, DAH and ε (no impulse) (FIGURE 7).

We finally decode MORSE to retrieve the information on two channels (green arrows on the previous flowchart). We have two files with the same information.

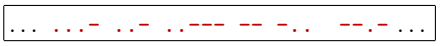

FIG. 7: Digitized signal associated to pulses

FIG. 8: MORSE decoded information

D. Gain and phase detection : TEC information

Eight channels (blue arrows on the previous flowchart) will allow us to acquire the AD8302 Gain and Phase Detectors. We directly save voltage measurements in a file. TEC retrieving will be done as post-processing with an external Matlab program.

II. THEORY AND IMPLEMENTATION

A. Antenna rotor control

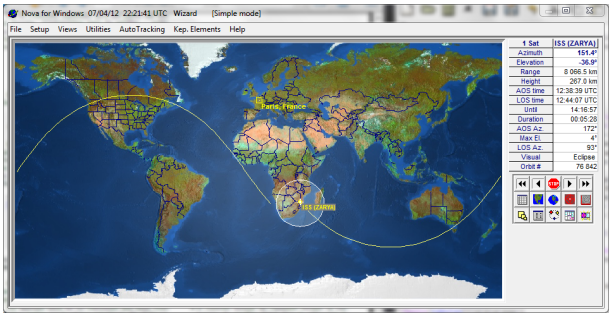

FIG. 9: NLSA Nova software

The antenna rotor will be entirely managed by NLSA Nova software [9] will send commands through RS-232 communication port to the GS-232B (ref ?) microcontroller (wired to the GS ? (ref ?) antenna rotor). This software manages:

- orbital elements database and automatic updates from Space-Track ;
- AOS and LOS time information generation ;
- antenna rotor control (AZIMUTH and ELEVATION parameters) following satellite trajectory.

Remarque. Technical details on the software usage can be found at APPENDIX II.

B. Earth orbit model

The Earth orbit model chosen is SGP4/SDP4 model for the following reasons: (ref ?)

C. Time-based job scheduling

Method. AOS and LOS time information files can be generated by Nova (usual file name `Nova listing data.TXT`) as specified below:

1. Choose the satellite and the observer (`Configure current view`)
2. Utilities → Listing
3. One Observer AOS/LOS → ReCalc
4. Capture Listing → ASCII Text File (Entire listing)

Remarque. ☢ (AOS/LOS TIME INFORMATION SYNTAX) AOS and LOS time informations returned by Nova are saved in a syntax whose grammar is given in Extended Backus−Naur Form (EBNF) [12] (FIGURE 17).

Remarque. (TIME ZONE) For software implementation convenience, we assume the machine time zone is configured at `UTC` or `GMT+0`.

Définition. ☢ (ACQUISITION SEQUENCE) An *acquisition sequence* is specified by a couple of times which respectively matches with the AOS and the LOS. We denote $\mathcal{S}$ a set of ordered acquisition sequences specification.

We suggest the following algorithm (ALGORITHM 1, IMPLEMENTATION 10) to implement a time-based scheduler for continuous acquisitions from AOS to LOS.

Algorithm 1 Scheduler

Require: non-empty time-ordered $\mathcal{S}$, $t_i()$: time at call of t_i, F_s : file associated to sequence s, F_r : refresh rate

for $s \in \mathcal{S}$ **do**
  while $\neg(t_d(s) \leq t_i() < t_f(s)) \wedge (t_i() < t_f(s))$ **do**
    `sleep`$(\frac{1}{F_r})$ // *refresh*
  end while
  $F_s \leftarrow$ `Logging acquisition loop`$(t_i(), t_d(s), t_f(s))$
end for
return $F_{s_1}, \ldots, F_{s_{|\mathcal{S}|}}$

Remarque. (ACCURACY OF ACQUISITION SEQUENCE RELEASE AND STOP) If we assume $\varepsilon > 0$ we label F_r the scheduler refresh rate. We may have an acquisition start delay of $\frac{1}{F_r} - \varepsilon$. Moreover if we assume the logging acquisition loop is called with F_e and N, we may have a acquisition overflow of $\frac{N}{F_e} - \varepsilon$ (acquisition window length).

D. Acquisition loop

Method. For the NI USB-6353 Acquisition Box, the recommended warm-up time is 15 minutes[16].

Définition. (ACQUISITION WINDOW LENGTH) The *acquisition window length* is the time duration of one acquisition of N samples at a sampling rate of F_e given by the formula

$$\Delta t \triangleq \frac{N}{F_e}$$

Définition. (ACQUISITION LOOP) An *acquisition loop* in a loop control structure for repeted and regular acquisitions.

A sequence of acquisitions is implemented by an acquisition loop (ALGORITHM 2) which calls a set of functions provided by NI-DAQmx libraries [18] with a return type of Dynamic Type [23].

Remarque. Delays between data logging and acquiring function recall are assumed to be handled by the NI LabVIEW API through the usage of FIFO memories in the NI USB-6353 Acquisition Box[16].

Algorithm 2 Logging acquisition loop

Require: $t_i()$: initial time, t_d : acquisition start time, t_f : acquisition end time, F_e : sampling rate, N : number of samples a window

F : fichier
while $t_d \leq t_i() < t_f$ **do**
  $F \Leftarrow$ `acquire`(F_e, N)
end while
return F

E. Sampling

We suggest to introduce some denotations for further configurations.

Notation. Let x be a discrete signal of N samples indexed by a set I in the form $x : I \to \mathbb{R}, t \mapsto x(t)$. We denote the following physical quantities

F_c : central frequency [Hz]

Δf : bandwidth length [Hz]

F_e : sampling rate [Hz]

F_b : bit rate of the modulated signal [Hz = bits . s^{-1}]

N : number of samples [1]

Δt : acquisition window length [s]

Configuration. The usage of `DAQ Assistant` function is deprecated but still remains possible with the following configuration:

- `Acquisition mode` : choose `Continuous Samples` [19]

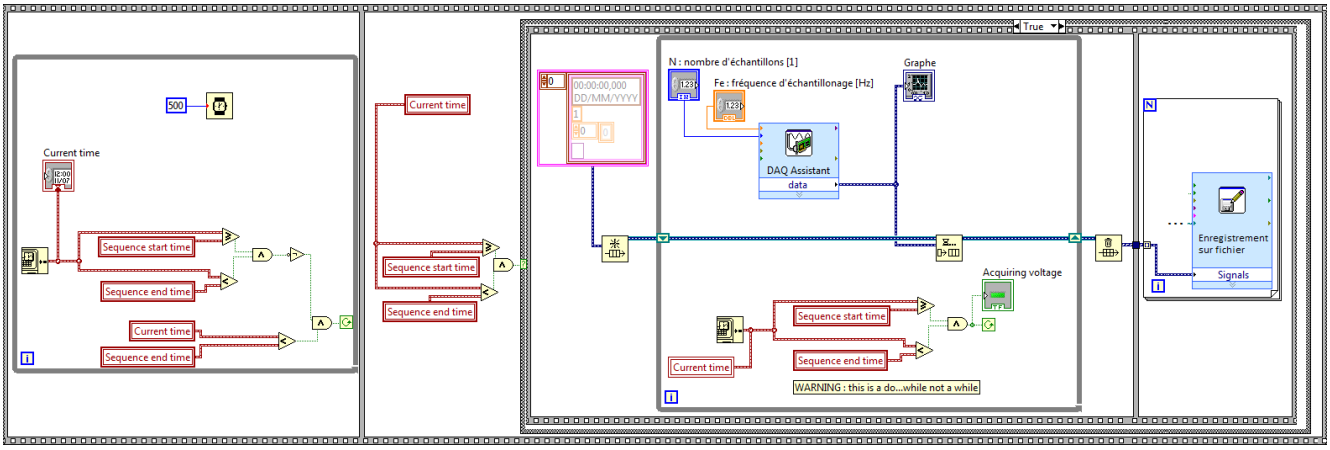

FIG. 10: Scheduler implemented in LabVIEW

- `Samples to Read` : N
- `Rate (Hz)` : F_e

Remarque. (FIFO MEMORY) The NI USB-6353 Acquisition Box has a FIFO memory size of 4095 samples [16] (consequence ?).

Remarque. (COMPLEXITY) 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.

Signal post-processing allows us to choose a specific WINDOW LENGTH for further digitizing. For a signal of bit rate F_b the window length is defined as

$$\frac{1}{F_b}$$

Assuming the FM demodulation process done by the transceiver, the sampling rate can be focused only on the bit rates in the signal.

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

$$F_e \geq 2 \times \max\{F_{b_1}; F_{b_2}\} \tag{1}$$

Remarque. (OPTIMAL SAMPLING) 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].

The NI USB-6353 Acquisition Box on multi-channel is able to deliver a maximum sampling rate (aggregate) on multi-channel [16] of

$$\begin{aligned} F_{e_{\text{MAX}}} &= 1.00 \text{ MS.s}^{-1} \\ &= 1.00 \times 10^6 \text{ Hz} \end{aligned}$$

L'acquisition de toutes les voies devant être quasi-simulaltanée, il paraît trivial que toutes les voies soient au moins échantillonées à $2 \times 2,4 \times 10^3 = 4,8 \times 10^3$ Hz (par application du thórème 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, échantilloner chacune des voies à 10 kHz.

F. Frequency component detection

☢ This section may be omitted from the understanding of the software system. It is only suggested as a reminder for the use of FFT-based SPECTRUM ANALYZERS.

Définition. (FREQUENCY SPECTRUM) Let $\mathcal{F}$ be the discrete Fourier transform of an integrable fonction on $\mathbb{R}$ and $\hat{x} = \mathcal{F}(x) = [k \mapsto \hat{x}(k)]$ the *frequency spectrum* of x (at the kth sample). We have[6] :

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

We denote $A(f)$ the amplitude of the frequency component f.

Proposition. II.2 (AMPLITUDE) *The Fourier transform of a signal of N samples is a map which for an index $k \in \{0; N-1\}$ associates the amplitude of the frequency f such that $f = \frac{k.F_e}{N}$.*

We then have[6] :

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

G. FSK demodulation

We assume FSK demodulation to be made by the VR5000 Transceiver. We only suggest a simple algorithm for signal digitizing.

Proposition. II.3 (SAMPLES PER WINDOW) *We denote N_w the number of samples per calculation window. Assume one logic bit on a time duration $\Delta t = \frac{1}{F_b}$. We have:*

$$N_w = \frac{F_e}{F_b} \tag{2}$$

The calculation window length (expressed in number of samples) of the digitizer is $\frac{F_e}{F_b}$.

Idea. (NUMÉRISATION FSK) On cherche, à partir d'un signal composé de deux amplitudes, à distinguer celle ayant l'occurence 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 autre.

On propose un algorithme (ALGORITHME 3, IMPLÉMENTATION 11) 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:

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

Proposition. II.4 (MEAN FORMULA) *Let $N \in \mathbb{N}$ and $x : \{0; \ldots; N-1\} \to \mathbb{R}$ a signal of N samples. We denote the mean of the signal x with:*

$$\begin{aligned} \mathbb{R}^{\{0;\ldots;N-1\}} &\to \mathbb{R} \\ \text{mean} : x &\mapsto \frac{1}{N}\sum_{i=0}^{N-1} f(i) \end{aligned}$$

Specification. The digital baseband transmission method is of the kind non-return-to-zero (NRZ). In other words, logical `1` matches with one significant condition and logical `0` with another.

H. OOK demodulation (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'abscence du signal porteur F_c. On note $A(F_c)$ l'amplitude associée à la composante fréquentielle F_c. De manière analogue, en consiérant que les opŕations de démodulation sont effectuées par le `VR5000`, on propose un algorithme (ALGORITHME 4) similaire de numérisation de données modulées en `OOK`.

Algorithm 3 Numérisation FSK

Require: signal x à N valeurs, amplitude de f_0 : $A(f_0)$ et amplitude de f_1 : $A(f_1)$
1: `moyenne` $\leftarrow \frac{1}{N}\sum_{i=0}^{N} x(i)$
2: **if** $|\texttt{moyenne} - A(f_0)| > |\texttt{moyenne} - A(f_1)|$ **then**
3: **return** `0`
4: **else**
5: **return** `1`
6: **end if**

Algorithm 4 Numérisation OOK

Require: signal x à N valeurs, amplitude de F_c : $A(F_c)$
1: **if** $\frac{1}{N}\sum_{i=0}^{N} x(i) < \frac{A(F_c)}{2}$ **then**
2: **return** `0`
3: **else**
4: **return** `1`
5: **end if**

I. AX.25 packets decoding

Définition. ☢ (LSB−MSB) Let x be a number in binary representation. There exists a finite sequence $(b_n)_{0 \le n \le n_0}$ of whole numbers that equals to 0 or 1 such that

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

The whole number b_0 is called the *least significant bit* (LSB) and the number b_{n_0} the *most significant bit* (MSB).

Exemple. Let $x = 42$ in decimal representation, we denote $(x)_{\text{bin}}$ the binary representation of x in the usual positional notation. We have:

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

Remarque. (BOUTISME DE BITS) Tous les octets (mots de longeur 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 $\mathbb{B} = \{\texttt{0}, \texttt{1}\}$.

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

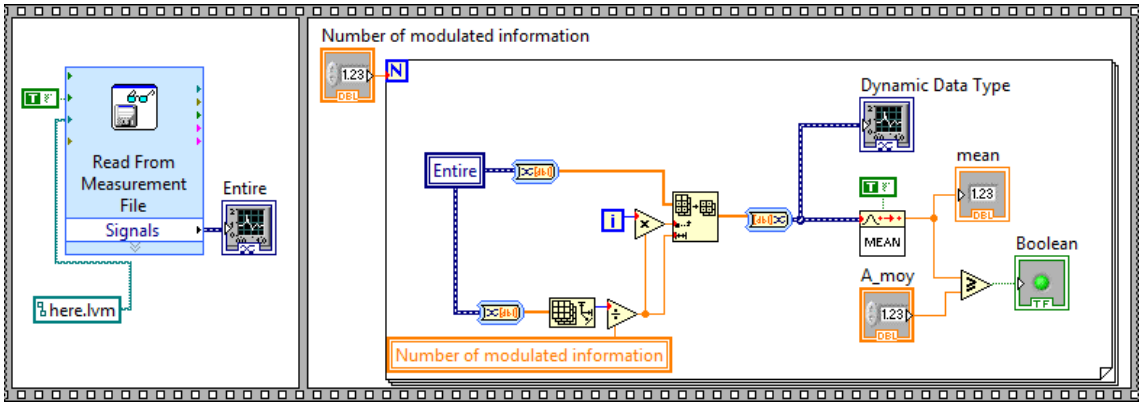

FIG. 11: Implementation of a signal digitizer in LabVIEW

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$_1$			Sous-adresse$_2$		Sous-adresse$_3$	
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* Σ.

$$\operatorname{Rat}\Sigma^\star = \operatorname{Rec}\Sigma^\star$$

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

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

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 12 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].

Remarque. (BOURRAGE DE BIT) Le fanion `01111110` (`7E` hexadécimal) ne peut donc, en aucun cas, avoir d'occurence dans de trames complètes[3]. Le procédé de *bourrage de bit* (*bit stuffing*) permet alors d'eviter la confusion entre les données et les fanions. Durant la réception des données, on ignore le bit `0` à chaque occurence de cinq bits 1 consécutifs, après les opérations de découpages de trames.

Algorithm 5 Découpage de trames AX.25

Require: 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: **if** une trame n'est pas ouverte **then**
6: ouvrir une nouvelle
7: **else**
8: **if** une trame est déjà ouverte, **then**
9: **if** elle n'est pas vide **then**
10: la refermer et l'enfiler dans l
11: **end if**
12: **end if**
13: **end if**
14: **parse** _ :
15: **if** une trame est déjà ouverte **then**
16: concaténer le bit _ dans la trame
17: **else**
18: ne rien faire *//bits perdus car aucune trame ouverte*
19: **end if**
20: **fin analyse lexicale**
21: **return** liste l de mots binaires dont chaque élément est une trame

Implémentation. (ALGORITHME 6) ☢ Après le découpage des trames, on extrait, de chaque trame, les bits de bourrage.

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.

J. Décodage du Morse

Liaison. Simplex (définition ANSI)

```
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. 12: Syntaxe des trames Pratham en AX.25 en notation EBNF

Algorithm 6 Extraction du bit de bourrage

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

```
mot vide m'
compteur c ← 0
for i de 0 à p do
if m_i = 0 then
if c = 5 then
c ← 0
else
c ← 0
m' ← m'.0
end if
else
c ← c + 1
m' ← m'.1
end if
end for
return m'
```

Algorithm 7 Pattern-matching on AX.25 packets

Require: liste de mots l

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 $(\cdot)$;
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 :

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

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

$$h : \mathbb{M} \to \mathbb{B}^*$$
$$m \mapsto \begin{cases} \texttt{0} & \text{si } m = \varepsilon_1 \\ \texttt{000} & \text{si } m = \varepsilon_3 \\ \texttt{0000000} & \text{si } m = \varepsilon_7 \\ \texttt{1} & \text{si } m = \cdot \\ \texttt{111} & \text{si } m = - \end{cases}$$

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

$$L : \mathbb{M} \to \mathbb{N}$$
$$x \mapsto \begin{cases} 1 & \text{si } x \in \{., \varepsilon_1\} \\ 3 & \text{si } x \in \{-, \varepsilon_3\} \\ 7 & \text{si } x = \varepsilon_7 \end{cases}$$

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

$$\widetilde{N} : \mathbb{M}^\star \to \mathbb{N}$$
$$x = (x_0 \dots x_{n-1}) \mapsto \sum_{i=0}^{n-1} N(x_i)$$

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

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

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

1. (SÉPARATION)
$$\forall x \in \mathbb{M}^\star, \widetilde{N}(x) = 0 \Rightarrow x = \varepsilon$$
2. (HOMOGÉNÉITÉ)
$$\forall (\lambda, x) \in \mathbb{N} \times \mathbb{M}^\star, \widetilde{N}(x^\lambda) = \lambda \widetilde{N}(x)$$
3. (INÉGALITÉ TRIANGULAIRE)
$$\forall (x, x') \in \mathbb{M}^\star, \widetilde{N}(x.x') \leq \widetilde{N}(x) + \widetilde{N}(x')$$

Implémentation. On implémente tout d'abord un transducteur fini qui sur l'alphabet $\mathbb{B} = \cdot, -$ reconnait un mot sur $\mathbb{B}$ et retourne la lettre associee a son codage en Morse. Le transducteur est donné en FIGURE 15.

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

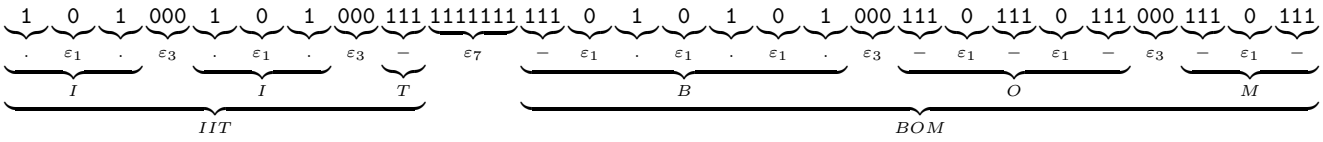

FIG. 13: Exemple des couches d'abstraction d'un signal Morse codant l'information ‘‘IIT Bombay’’

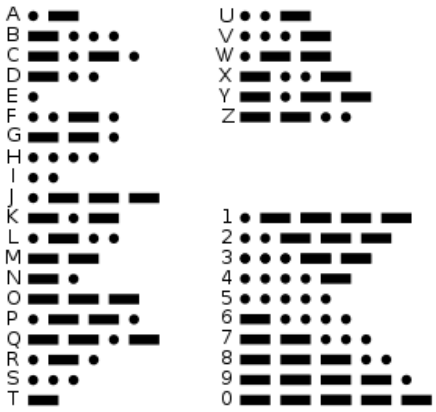

FIG. 14: Alphabet du code Morse

Algorithm 8 Décodage Morse à partir du signal modulant (numérique)

Require: mot $m = m_0 \dots m_{p-1}$ sur l'alphabet $\mathbb{B}$ de longueur p, transducteur T_{Morse} muni de `eval()`

chaîne de caractrères *acc*, et *res*
analyse lexicale de m **faire**
parse `0000000` : *(* ε_7 *)*
$res \leftarrow res.$‘‘ ’’
$acc \leftarrow \varepsilon$
parse `000` : *(* ε_3 *)*
$res \rightarrow res.$ `eval`(T_{Morse}, acc)
$acc \leftarrow \varepsilon$
parse `0` : *(* ε_1 *)*
$acc \leftarrow \varepsilon$
parse `111` : *(* − *)*
$acc \leftarrow acc.-$
parse `1` : *(* · *)*
$acc \leftarrow acc.\cdot$
fin analyse lexicale
return *res*

K. 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

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

où :

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

N : nombre d'intervalles de tensions [1]

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

$V_{\mathbf{RefLow}}$: 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

L. 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'effecuter 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.*

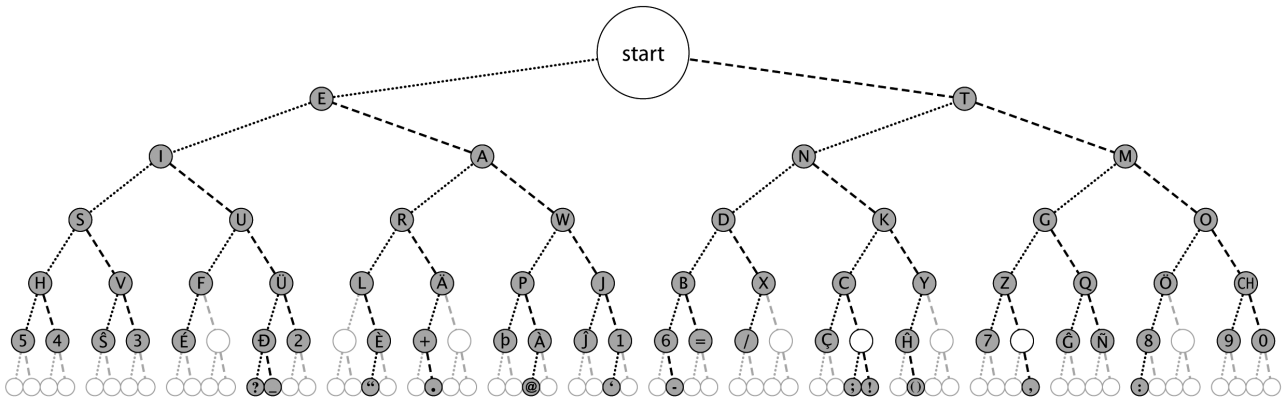

FIG. 15: Transducteur fini associé au décodage MORSE [38]

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* $(\mathbb{B}^\star, l.)$ *un espace normé. On a alors l'approximation suivante de la longueur d'un enregistrement d'une séquence d'acquisition :*

$$\begin{aligned} l_{\text{fichier}} &= \underbrace{\Delta t}_{\in[7;13]\ \text{min}} \times \underbrace{f_e}_{10\ \text{kHz}} \\ &\times \underbrace{\left(\underbrace{l_{\text{date}}}_{23\ \text{octets}} + \underbrace{n_{\text{voies}}}_{12} \times \left(\underbrace{l_{\text{flottant}}}_{12\ \text{octets}} + \underbrace{l_{\text{separateur}}}_{1\ \text{octet}} \right) \right)}_{\text{une ligne}} \quad (4) \\ &\in \left[7,518 \times 10^8; 1,3962 \times 10^9\right] \quad (5) \\ &\quad \text{car } \Delta t \mapsto l_{\text{fichier}}(\Delta t) \text{ croissante} \end{aligned}$$

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 :

$$\begin{aligned} l_{\text{jour}} &= n_{\text{passages/jour}} \times l_{\text{fichier}} \quad (7) \\ &\in \left[3,0072 \times 10^9; 8,3772 \times 10^9\right] \quad (8) \\ &\quad \text{car } n_{\text{passages/jour}} \mapsto l_{\text{jour}}(n_{\text{passages/jour}}) \text{ croissante} \end{aligned}$$

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

$$\begin{aligned} l_{\text{total}} &= n_{\text{passages/jour}} \times 30,5 \times 4 \times l_{\text{fichier}} \quad (9) \\ &\in \left[3,668784 \times 10^{11}; 1,0220184 \times 10^{12}\right] \quad (10) \end{aligned}$$

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

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

En considérant que la fréquence d'échantillonage 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).

Dans le cadre d'une solution provisoire, on se permet d'utiliser le format de fichier de "LabVIEW Measurement File" [24] (solution à probleme car enregistre avec la date d'enregistremen

M. 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.

N. 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 á un référentiel géographique.

LabVIEW, étant langage de programmation peu adapé à certains paradigmes, ne dispose pas de générateur d'analyseur syntaxique, ni lexical mais d'un analyseur par formats. Nous implémentons provisoirement une analyse formattée avec la fonction `Scan From String` [22] (*analyseur par formats d'une chaîne de 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 16) [29].

Algorithm 9 Analyse formattée des fichiers d'éphémérides de Nova

Require: F : fichier
1: hello world
2: **return** s

B. Enregistrements Pratham

Les formats de fichier qui seront utilisées pour l'enregsitrement des données de TEC vertical et slant sont spécifiées dans le document ??? dont on donne une syntaxe en notation EBNF en 19.

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 léxèmes en figure 20. D' est annulable.

On calcule FIRST() et FOLLOW() :

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

$$\begin{aligned} \text{FOLLOW}(F) &= \varnothing \\ \text{FOLLOW}(S) &= \{f\} \\ \text{FOLLOW}(D) &= \{\$\} \\ &\subseteq \text{FOLLOW}(D') \text{ car } \dot{\epsilon} \in \text{FIRST}(D') \\ \text{FOLLOW}(D') &= \{\$\} \end{aligned}$$

Comme montré en 20, la grammaire ne présente aucun conflit FIRST/FIRST dans le cas d'une analyse $LL(1)$. Donc :

$$G \in LL(1)$$

□

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. ☢ (TRIPLET DE HOARE) Soient P, Q des prédicats et `prog` un programme. Un triplet T est un *triplet de Hoare* si et seulement si

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

Définition. ☢ (CORRECTION D'UN PROGRAMME) Un triplet de Hoare $\{P\}$ `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.

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 Δt.*

Démonstration. Clair.

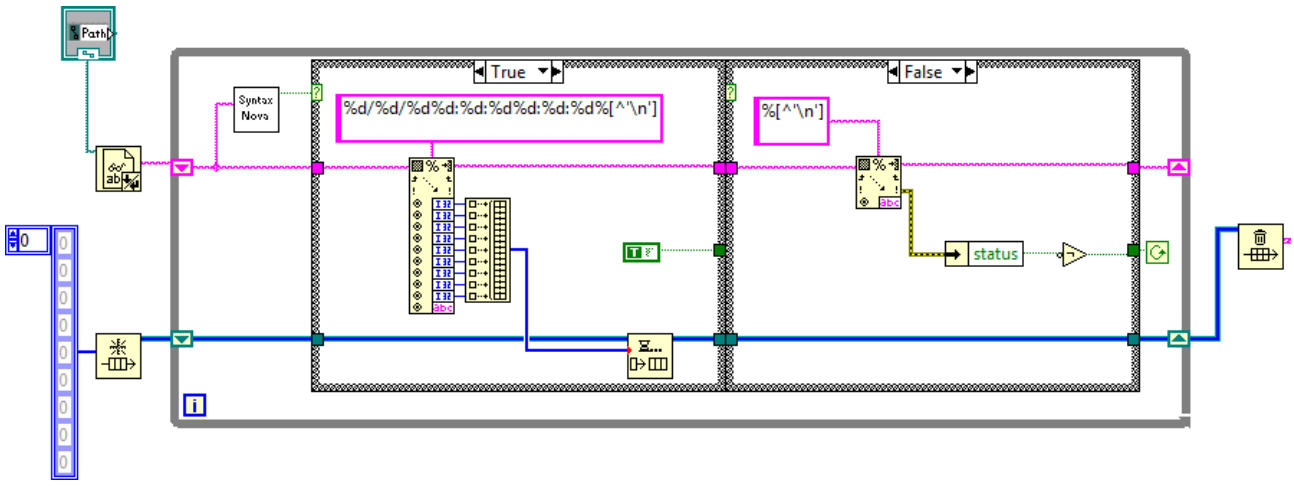

FIG. 16: Implémentation d'une analyse formattée en LabVIEW

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

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 éventullement correctement configurés dans l'ordre[14]

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

	Piston 1	Piston 2
Avg. Force (N)	4.40	2.25
Area (cm^2)	6.16	2.25
F/A (N/cm^2)	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

```
         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. 18: 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. 19: Syntaxe des enregistrements au format Pratham en notation EBNF pour les données brutes

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 constemment 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 Informatique et en Automatique, *CeCILL Free Software 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. Coïsson, P. Godbole, J. Jha, *Pratham satellite : File Formats Specification* (IPGP, May 2011)

[5] L. Viens, P. Coïsson, *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,

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

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

	SI	L	STI	STU	SR	DP	AT	T	145_0	145_1	145_2	145_3	437_0	437_1	437_2	437_3	f	m	n	:	\$
F	$\{F \rightarrow \dots\}$	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅
S	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	$\{S \rightarrow \dots\}$	∅	∅
D	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	$\{L \rightarrow \dots\}$	∅	∅
D'	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	$\{L' \rightarrow n \dots\}$	∅	$\{L' \rightarrow \epsilon\}$

FIG. 20: Table d'analyse LL(1) de la grammaire

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, ocamlyacc) (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 Washigton 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] "$f(n) \in \Omega\left(g(n)\right) \overset{\text{Landau}}{\Leftrightarrow} \exists k > 0, n_0\ \forall n > n_0\ g(n) \cdot k \leq |f(n)|$", 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 Fairhust, *Internet Communications Engineering - A Tutorial* (University of Aberdeen, October 2001)