The French Pratham ground station software system

Équipe Pratham, Université Paris Diderot, Institut de Physique du Globe de Paris

April 7, 2013

Contents

Chapter 1

Foreword

The following reference manual 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.

1.1 Licensing

The set of softwares provided as follows is copyright (c) 2010-2013 Institut de Physique du Globe de Paris (IPGP). IPGP holds all ownership rights to the software system. The software system is open source and can be freely redistributed. See the file `LICENSE` in the distribution for licensing information. The present documentation is copyright (c) 2010-2013 Institut de Physique du Globe de Paris (IPGP). This reference manual may be reproduced and distributed in whole or in part, subject to the following conditions:

- The copyright notice above and this permission notice must be preserved complete on all complete or partial copies.
- Any translation or derivative work of this reference manual must be approved by the authors in writing before distribution.
- If you distribute this reference manual in part, instructions for obtaining the complete version of this manual must be included, and a means for obtaining a complete version provided.
- Small portions may be reproduced as illustrations for reviews or quotes in other works without this permission notice if proper citation is given.

1.2 Availability

The complete set of softwares can be accessed via the Web site `https://github.com/EmptyStackExn`.

Chapter 2

Introduction

2.1 Overview

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

Table 2.1: 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	→	
			$AD8302_{mag}$	→	
			$AD8302_{phase}$	→	

Remark. We introduce on TABLE 2.1 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).

2.2 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.

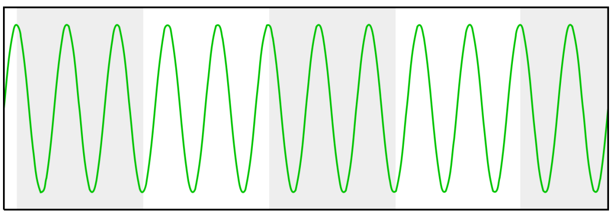

Figure 2.1: Carrier signal (central frequency F_c)

The satellite sends a signal with a declared carrier signal at $F_c = 437,455$ MHz (FIGURE 2.1) 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 2.2).

The transceiver returns an audio signal. It finally decribes – through a Schmitt trigger (ref ?) – the modulating signal (FIGURE 2.3). The digital baseband transmission method is of the kind non-return-to-zero

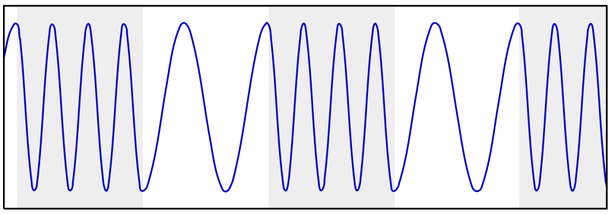

Figure 2.2: Modulated signal in FSK at $F_c \pm \frac{\Delta f}{2}$ frequencies received by the VR5000

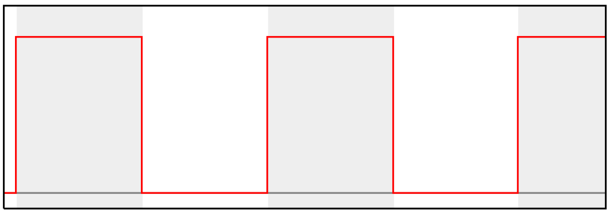

Figure 2.3: Modulating signal at $F_b = 1,2$ kbit . s^{-1} bit rate (retrieved by FSK demodulation)

(NRZ) [39] (ref ?). The data bits are then saved in a file (FIGURE 2.4) 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.

$$\ldots \underbrace{\texttt{01111110}}_{\text{packet start}} \ldots \underbrace{\texttt{10101010}}_{\text{information}} \ldots \underbrace{\texttt{01111110}}_{\text{packet end}} \ldots$$

Figure 2.4: Data bits associated to the modulating signal

We decode data bits from the AX.25 data link layer protocol (FIGURE 2.5).
At the end of the process we evaluate data bits as defined by IITB's specification[3].

```
... 10101010 ...
```

Figure 2.5: Data bits decoded from `AX.25` packets

2.3 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 2.6). 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 2.7).

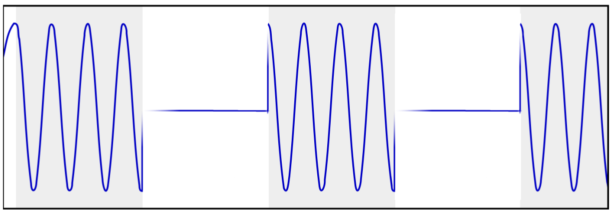

Figure 2.6: Modulated signal in OOK of fréquency F_c received by the transceiver

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 2.7).

```
... ...- ..- ..--- -- -..  --.- ...
```

Figure 2.7: Digitized signal associated to pulses

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.

2.4 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.

```
... VU2MDQ ...
```

Figure 2.8: Morse decoded information

Chapter 3

Acquiring and decoding

3.1 First steps

Repository. You can find the final software en environment in the `ipgp-pratham-gs-essential` repository.

3.2 Antenna rotor control

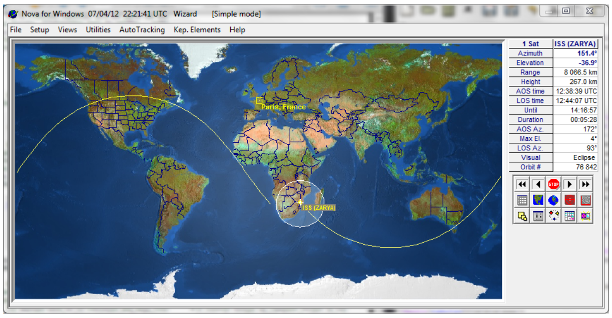

Figure 3.1: NLSA Nova software

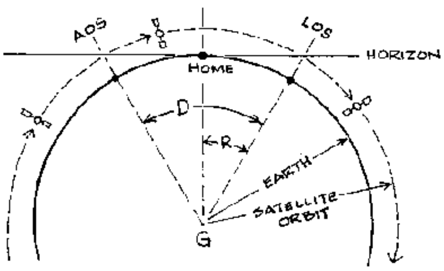

Figure 3.2: Earth orbit representation

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 ?) micro-controller (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.

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

3.3 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)

Remark. (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 4.1).

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

Definition. (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 3.3) 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 $F_{s_1}, \ldots, F_{s_{|\mathcal{S}|}}$

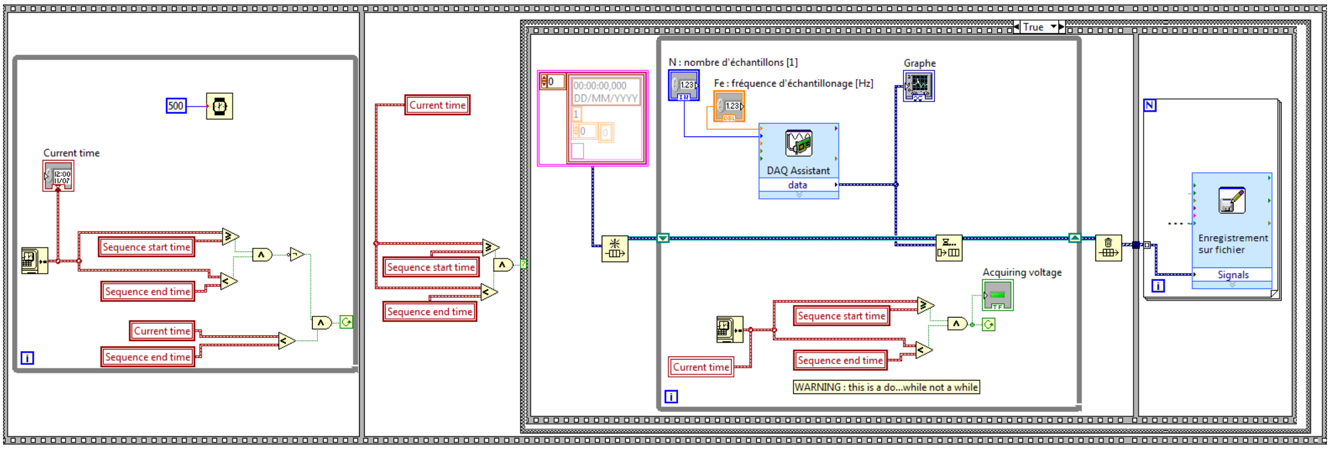

Figure 3.3: Scheduler implemented in LabVIEW

Remark. (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).

3.4 Acquisition loop

Repository. A set of tools for acquisition matters are available in the `ipgp-pratham-daq-collection` repository. Final sources are supposed to be compiled with NI Application Builder [25] [26].

Definition. (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

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

Definition. (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].

Remark. 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

3.5 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.

Repository. We provide however an implementation in C of a FFT−based sprectum analyzer in the `spectra` repository.

Definition. (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.

Lemma. 3.5.1 (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] :

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

3.6 FM demodulation

We assume FSK and OOK demodulation to be made by the VR5000 Transceiver. Retrieving bits of data refers to the NRZ decoding section.

3.7 NRZ decoding

Repository. The NRZ decoder is available in the `ipgp-nrz-decoder` repository.
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.

We assume we have a FM demodulated signal and that we will only focus on shape issue. We suggest different algorithms for same purposes.

Lemma. 3.7.1 (SAMPLES PER WINDOW) *We denote N_w as the number of samples per calculation window. Assume one logic bit to be on a time duration $\Delta t = \frac{1}{F_b}$. We have:*

$$N_w = \frac{F_e}{F_b} \tag{3.1}$$

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

Algorithm 3 NRZ decoder based on mean comparison

Require: signal x of N values, amplitude of f_0 : $A(f_0)$ and amplitude of f_1 : $A(f_1)$

`mean` $\leftarrow \frac{1}{N}\sum_{i=0}^{N} x(i)$
if $|$`mean` $- A(f_0)| > |$`mean` $- A(f_1)|$ **then**
  return `0`
else
  return `1`
end if

3.8 AX.25 packets decoding

Repository. The AX.25 decoder is available in the `ipgp-ax25-decoder` repository.

Definition. (LSB−MSB) Let x be a number in binary representation. There exists a finite sequence $(b_n)_{0\leq n\leq n_0}$ 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).

Remark. (BIT ENDIANNESS) Every bytes (words on bits of length 8) of a AX.25 packet (except for the FCS unit) are received with the LEAST SIGNIFICANT BIT first[3].

We define the alphabet $\mathbb{B}$ as $\{0; 1\}$ and give the following definition for a standard Pratham AX.25 packet.

Definition. (AX.25 PACKET OF PRATHAM) A word m on $\mathbb{B}^\star$ is an AX.25 PACKET OF PRATHAM [3] [8] if there exists a sequence of the following words (or UNITS) (at one flag at least):

m : 114 bytes						
Flag	Address	Control	PID	Information	FCS	Fanion
`01111110`	...	`00000011`	`11110000`	...	2 bytes	`01111110`

Address : 21 octets							
Subaddress$_1$			Subaddress$_2$		Subaddress$_3$		
7 bytes			7 bytes		7 bytes		
`CQ`	$((20)_{\text{hex}})^4$	`01100000`	`VU2DMQ`	`01100000`	`RELAY`	$(20)_{\text{hex}}$	`01100000`

Information : 87 bytes
Health monitoring data[3]

The information unit description is defined in [3].

Lemma. 3.8.1 *(*SYNTAX GRAMMAR OF AX.25 PACKET BITS*) The grammar for AX.25 packets given in 3.4 is linear. Thus, there exists a finite state transducer which parses this grammar.*

Remark. (BIT STUFFING) Bit stuffing coding is able to seperate flags `01111110` (hexadécimal `7E`) from real information given in AX.25 packets[3]. During data reception, we ignore `0` at each successive occurence of `1` after packets seperation operations.

Implementation. (ALGORITHME ??) We use a finite state transducer in order to parse sequences of AX.25 packets.

```
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 *)
```

Figure 3.4: Grammar of Pratham packets en AX.25

$$\underbrace{\underbrace{\underbrace{1}_{.}\underbrace{0}_{\varepsilon_1}\underbrace{1}_{.}}_{I}\underbrace{000}_{\varepsilon_3}\underbrace{\underbrace{1}_{.}\underbrace{0}_{\varepsilon_1}\underbrace{1}_{.}}_{I}\underbrace{000}_{\varepsilon_3}\underbrace{\underbrace{111}_{-}}_{T}}_{IIT}\underbrace{1111111}_{\varepsilon_7}\underbrace{\underbrace{\underbrace{111}_{-}\underbrace{0}_{\varepsilon_1}\underbrace{1}_{.}\underbrace{0}_{\varepsilon_1}\underbrace{1}_{.}\underbrace{0}_{\varepsilon_1}\underbrace{1}_{.}}_{B}\underbrace{000}_{\varepsilon_3}\underbrace{\underbrace{111}_{-}\underbrace{0}_{\varepsilon_1}\underbrace{111}_{-}\underbrace{0}_{\varepsilon_1}\underbrace{111}_{-}}_{O}\underbrace{000}_{\varepsilon_3}\underbrace{\underbrace{111}_{-}\underbrace{0}_{\varepsilon_1}\underbrace{11}_{-}}_{M}}_{BOM}$$

Figure 3.5: Example of token generation steps for the Morse coded beacon signal : "`IIT Bombay`"

3.9 Morse decoding

Repository. The Morse decoder is available in the `ipgp-morse-decoder` repository.
Link. Simplex (ANSI definition)

Specification. The Morse code specification for the satellite beacon signal is the same as provided by the International Telecommunication Union [13]. We give a short description:

1. The DAH letter ($-$) lasts three times longer than the DIT letter ($\cdot$) ;
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.

Definition. We denote $\mathbb{B}$ as the set of bits, $\mathbb{M}$ the Morse alphabet and $\mathbb{N}_{255}$ the ASCII representable characters as follows:

$$\begin{aligned} \mathbb{B} &= \{\mathtt{0};\mathtt{1}\} \\ \mathbb{M} &= \{\varepsilon_1, \varepsilon_3, \varepsilon_7, ., -\} \\ \mathbb{N}_{255} &= \{\ldots; \mathrm{a}; \mathrm{b}; \ldots\} \end{aligned}$$

Definition. We define on $\mathbb{M}$ the coding function for one token (??? inexact):

$$\begin{aligned} \text{morse\_code} : \mathbb{M} &\rightarrow \mathbb{B}^* \\ m &\mapsto \begin{cases} \mathtt{0} & \text{if} \quad m = \varepsilon_1 \\ \mathtt{000} & \text{if} \quad m = \varepsilon_3 \\ \mathtt{0000000} & \text{if} \quad m = \varepsilon_7 \\ \mathtt{1} & \text{if} \quad m = \cdot \\ \mathtt{111} & \text{if} \quad m = - \end{cases} \end{aligned}$$

Definition. We define on $\mathbb{M}$ the length of a letter $x \in \mathbb{M}$ as:

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

Implementation. We implement a transducer with two entrypoints (or two transducers with one unique entrypoint) with a lexical analyzer generator from the alphabet $\mathbb{B}^\star = \{\mathtt{0};\mathtt{1}\}^\star$ to $\mathbb{M}^\star$. The second transducer processes from $\mathbb{M}^\star$ to $\mathbb{N}_{255}^\star$ (FIGURE 3.7).

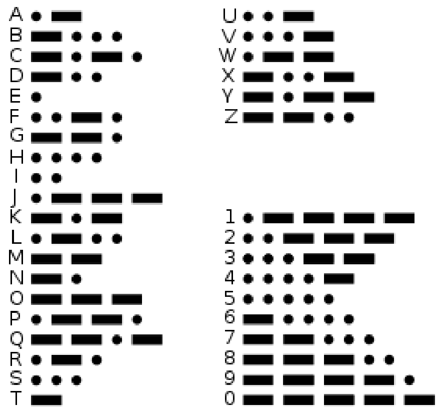

Figure 3.6: Morse code alphabet

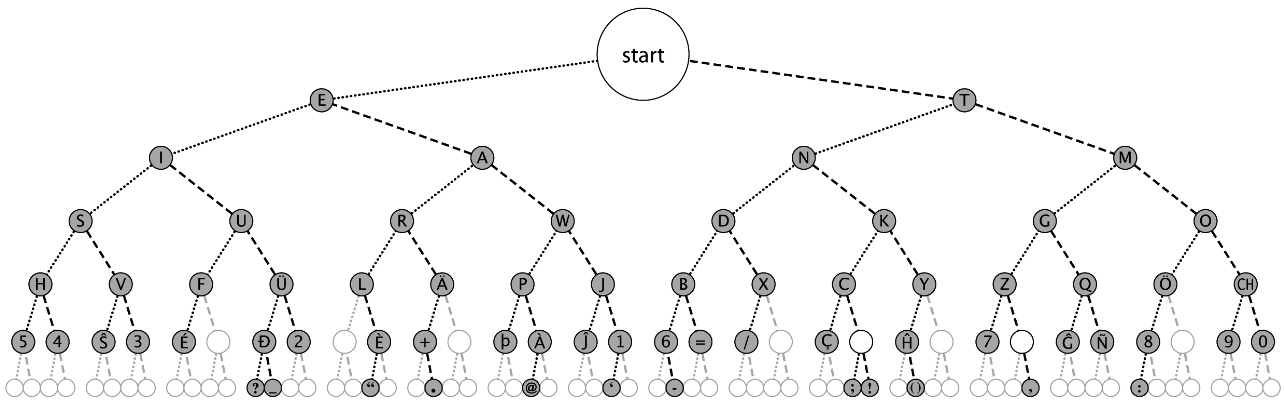

Figure 3.7: Transducteur fini associé au décodage MORSE [37]

3.10 Data logging

We remind that data logging rules are specified in *Pratham satellite : File format specification document*[4]. Duration of satellite AOS depends on the following parameters:

- the satellite altitude ;
- the Earth orbit model used in the tracking softwware ;
- the antenna aperture (???) ;
- the low elevation multi-targeting (???)

We give a preliminary estimation of total logging file sizes considering that a measure may be represented in decimal ASCII character representation with 6 digits.

Lemma. 3.10.1 *(*SPACE COMPLEXITY OF A FLOATING-POINT NUMBER*) Assuming that a floating-point number will be represented on* 6 *digits and that an encoding character costs* 1 *byte. A floating-point number then costs* 7 *bytes.*

Remark. A floating-point number represented in IEEE ??? on 32 bits only costs 4 bytes − that is to say threes times cheaper that the previous representation −

Lemma. 3.10.2 *(*SPACE COMPLEXITY OF DATA 0- CALIBRATION INFO, STATION CHARACTERISTICS*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.3 *(*SPACE COMPLEXITY OF DATA 1- RAW OUTPUT FROM GROUND STATION ACQUISITION*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.4 *(*SPACE COMPLEXITY OF DATA 2- ORIENTATION OF THE REFERENCE*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.5 *(*SPACE COMPLEXITY OF DATA 3- AUXILIARY DATA*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.6 *(*SPACE COMPLEXITY OF DATA 4- PROCESSED DATA*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.7 *(*SPACE COMPLEXITY OF DATA 4- PROCESSED DATA*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

Lemma. 3.10.8 *(*SPACE COMPLEXITY OF DATA 5- PROCESSED DATA FOR PRATHART*) Let* $(\mathbb{B}^\star, l.)$ *be a normed space (???). We then have the following approximation for one acquisition sequence:*

$$\begin{aligned} l_{\text{file}} &= \underbrace{\Delta t}_{\in[7;13]\ \text{min}} \times \underbrace{f_e}_{10\ \text{kHz}} \\ &\times \underbrace{\left(\underbrace{l_{\text{date}}}_{23\ \text{octets}} + \underbrace{n_{\text{voies}}}_{12} \times (\underbrace{l_{\text{flottant}}}_{12\ \text{octets}} + \underbrace{l_{\text{separateur}}}_{1\ \text{octet}}) \right)}_{\text{une ligne}} \qquad (3.2) \\ &\in \left[7,518 \times 10^8 ; 1,3962 \times 10^9\right] \qquad (3.3) \\ &\quad \text{as } \Delta t \mapsto l_{\text{fichier}}(\Delta t) \text{ increases} \end{aligned}$$

Considering a estimation model for the number of satellite AOS over Paris[5] :

$$n_{\text{AOS/day}} \in [4;6]\ \text{AOS} \qquad (3.4)$$

We then have, per day :

$$\begin{aligned} l_{\text{jour}} &= n_{\text{passages/jour}} \times l_{\text{fichier}} \qquad (3.5) \\ &\in \left[3,0072 \times 10^9 ; 8,3772 \times 10^9\right] \qquad (3.6) \\ &\quad \text{car } n_{\text{passages/jour}} \mapsto l_{\text{jour}}(n_{\text{passages/jour}}) \text{ croissante} \end{aligned}$$

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

$$\begin{aligned} l_{\text{total}} &= n_{\text{passages/jour}} \times 30,5 \times 4 \times l_{\text{fichier}} \qquad (3.7) \\ &\in \left[3,668784 \times 10^{11} ; 1,0220184 \times 10^{12}\right] \qquad (3.8) \end{aligned}$$

On présente en TABLE 3.1, les estimations de taille de fichier concernant les données brutes la sortie de l'acquisition.

Table 3.1: 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

$$T_e \triangleq \frac{1}{F_e}$$

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'enregistrement et non la date d'acquisition).

Chapter 4

File formats

4.1 Nova AOS/LOS timestamps file format

We give in EBNF the syntax grammar for AOS and LOS information returned by Nova. We show that a part of this grammar is linear. Thus, the formed language is regular and there exists a regular expression which denotes it. We then parse a file using the `Scan From String` [22] function.

```
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" newline

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' }
```

Figure 4.1: Grammar of AOS/LOS information returned by Nova

Lemma. 4.1.1 *The restriction of 4.1 whose start symbol is* `day_eclipses` *is a linear grammar.*

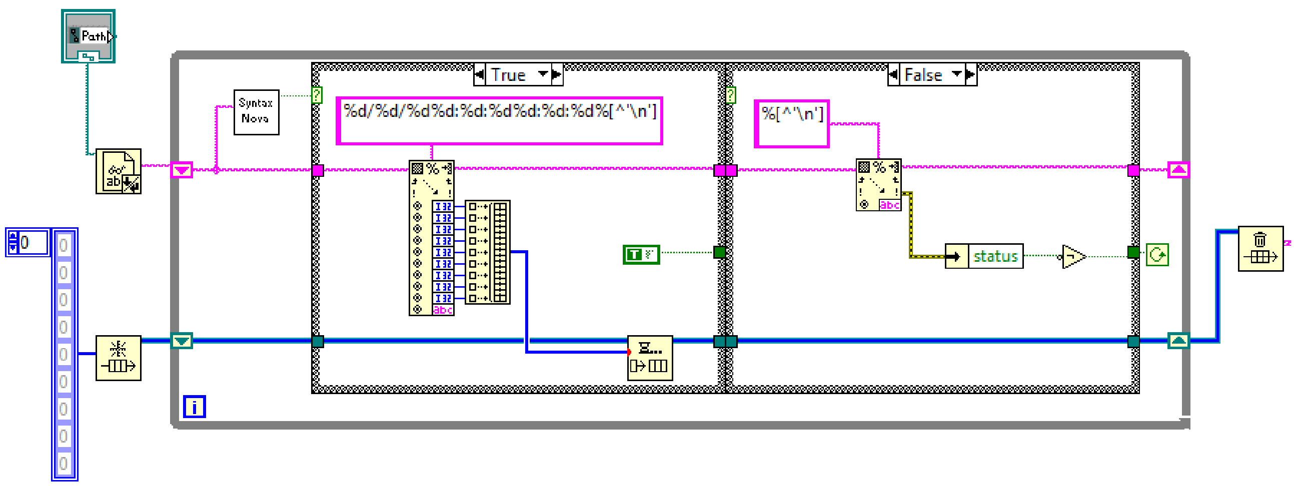

Figure 4.2: Regular expression which denotes the restriction whose start symbol is `day_eclipses`

Lemma. 4.1.2 *There exists a regular expression which denotes the language formed by the restriction of 4.1 and is given in 4.2.*

4.2 Pratham data log

Pratham file formats decided betwwen IPGP and IITB for vertical and slant TEC are specified in ??? whose we give a syntax grammar in EBNF in 4.3.

```
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' }
```

Figure 4.3: Pratham raw data logs grammar

Lemma. 4.2.1 *The grammar 4.3 is linear and forms a regular language.*

Chapter 5

Measurement and computation accuracy

5.1 Earth orbit model

The Earth orbit model chosen is SGP4/SDP4 model.

5.2 Measurement accuracy

The ADC has a resolution of 16 bits. We denote Q as the RESOLUTION or LEAST SIGNIFICANT BIT VOLTAGE (in volts). The maximum error that the instrument gives under the usual conditions are given by the ADC specifications (Section *AI Absolute Accuracy Table*, p.5) [16] and are confirmed by the following calculation for a SPAN of 20 V:

$$\begin{aligned} Q &= \frac{E_{\mathrm{FSR}}}{N} \\ &= \frac{V_{\mathrm{RefHi}} - V_{\mathrm{RefLow}}}{2^M - 1} \\ &= \frac{10 - (-10)}{2^{16} - 1} \\ &\simeq 0.0001520\,\mathrm{V} \end{aligned}$$

where :

$E_{\mathbf{FSR}}$: full scale voltage range (V)

N : number of voltage intervals [1]

$V_{\mathbf{RefHi}}$: upper extreme coding voltage (V)

$V_{\mathbf{RefLow}}$: lower extreme coding voltage (V)

M : ADC resolution (bits)

Considering that $V_{\mathrm{RefHi}} = 10\,\mathrm{V}$, for all acquired voltage U_a, U_a will be expressed in scientific notation as follows:

$$U_a = (d_1.d_2d_3d_4d_5d_6d_7d_8 \pm 0.0001520)\,\mathrm{V}$$

We then decide to keep 8 digits for a voltage measured with this ADC. However, further calculations have to be based on the number of SIGNIFICANT DIGITS which is 6.

5.3 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]
- `Samples to Read` : N
- `Rate (Hz)` : F_e

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

Remark. (COMPLEXITY) Le coût en temps de la fonction `DAQ Assistant` s'exprime en $\Omega(N)$[36] 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.

Theorem. 5.3.1 *(*NYQUIST-SHANNON *[33] [34])*

$$F_e \geq 2 \times \max\{F_{b_1}; F_{b_2}\} \tag{5.1}$$

Remark. (OPTIMAL SAMPLING) The previous theorem suggests that - for a digital signal - the base theoretical acquisition frequency is $0,5$ Hz per baud. In fact, we would need at least between $0,7$ and $0,8$ Hz per 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

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

Considering that the acquisition will be done simultaneously, it is trivial that all channels have a sampling rate of $2 \times 2,4 \times 10^3 = 4,8 \times 10^3$ Hz (see previous theorem). We may notice however that other DSPs have a limit of $4 \times F_c$ due to computation overflow [27].

Finally using a sample rate of 10 kHz is the most convenient choice we made.

Chapter 6

Hardware used at IPGP

6.1 Measurement machine

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

The acquisition machine used at IPGP is suggested as the kind of

- Model: Dell Latitude ATG D630
- Processor: Intel Core 2 Duo T7500 (2,2 GHz, 2,19 GHz)
- Operating system : Microsoft Windows XP
- Hard disk: Toshiba MK8046GSX (74,3 Go dont 71,5 Go occupés) (rapidité d'écriture ?)
- Network card:
  - LAN: Broadcom NetXtreme 57xx Gigabit Controller
  - Wifi: Intel PRO/Wireless 3945 ABG Network Connection

6.2 Monitoring issues

This section deals with additional features that aim to monitor the software/hardware system.

6.2.1 Rotor-based antenna

The antenna is usually out of order due to hard weather conditions. It is suggested to install a remote CCTV. The simplest way is to install a software like Skype and configure it in "Auto answer" mode with camera. The administrator will be able to monitor the state of the camera.

6.2.2 Acquisition software failure

The acquisition software is provided as experimental release and may contain bugs although all tests that have been made. When the system fails to run, an email is sent in order to aware the administrator.

6.2.3 Desktop remote control

It is highly suggested to give a remote control of the acquisition machine to the administrator in order to give him the possible of relauching the software in any case it would fail. For such purposes, it is possible to enable in Windows the remote control feature or use any other desktop remote software.

We notice however that such remote control systems are dangerous and may be unsafe. The administrator in charge of the software system is in charge of network and security issues.

Bibliography

[1] Commissariat á 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, *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] 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.

[34] 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).

[35] Jhonny Jha : "f0 would mean 1 and f1 would mean 0 is what I intended to say"

[36] "$f(n) \in \Omega\left(g(n)\right) \overset{\text{Landau}}{\Leftrightarrow} \exists k > 0, n_0\ \forall n > n_0\ g(n) \cdot k \leq |f(n)|$", P. E. Black in *Dictionary of Algorithms and Data Structures* (U.S. National Institute of Standards and Technology, December 2004)

[37] 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)

[38] Jean-Michel Autebert, *Théorie des langages et des automates* (Dunod, December 1997)

[39] Godred Fairhust, *Internet Communications Engineering - A Tutorial* (University of Aberdeen, October 2001)