# Phase D

# Beacon format

Prepared by:
Ted Choueiri

Checked by:
Florian George

Approved by:
Muriel Noca

EPFL
Lausanne
Switzerland

08/09/2009

Space Center
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Université Neuchâtel unine Hes·so Haute Ecole Spécialisée de Suisse occidentale EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

# RECORD OF REVISIONS

| ISS/REV | Date | Modifications | Created/modified by |
|---|---|---|---|
| 1/0 | 23/06/2008 | | Ted Choueiri |
| 1/1 | 23/07/2008 | Octal coding instead of decimal | Florian George |
| 1/2 | 18/10/2008 | Removed mode from part 1. Added more error flags and states. More precise current limits for solar cells in part 3. | Florian George |
| 1/3 | 08/09/2009 | Added octal conversion table and examples of received values signification.<br>Removed spacecraft mode | Florian George |

# 1 INTRODUCTION

This document describes the beacon message.

# 2 REQUIREMENTS

The requirements for the Beacon subsystem are described in: **S3-D-1-0-Level5_Beacon_specifications.**

# 3 TERMS, DEFINITIONS AND ABBREVIATED TERMS

## 3.1 Abbreviated terms

| **Abbreviation** | **Meaning** |
|---|---|
| ADCS | Attitude Determination and Control System |
| ADS | Antenna Deployment System |
| BSS | Beacon Subsystem |
| CDMS | Command and Data Management System |
| COM | Communication |
| EPS | Electrical Power System |
| LSB | Least Significant Bit. Bit on the far right in binary number representation |
| HBG | Hardware Beacon Generator |
| HBM | Hardware Beacon Message |
| MC | Microcontroller |
| MSB | Most Significant Bit. Bit on the far left in binary number representation |
| Rx | Receiver |
| SBG | Software Beacon Generator (i.e. EPS microcontroller) |
| SBM | Software Beacon Message |
| Tx | Transmitter |

# 4 BEACON MESSAGE

The BSS sends regularly a simple message to earth. It is used to identify the satellite and to send some basic information as to the state of the satellite. It is generated either by the Hardware Beacon Generator or by the Software Beacon Generator.

## 4.1 Basic requirements

The requirements are listed in the **S3-D-1-0-Level5_Beacon_specifications**. However, here is an overview of some requirements:

- The BSS will use at most at input 300mW in peak power, and 50mW in mean power.
- The BSS shall transmit at 14bps.
- Signal will be modulated in Morse code.

## 4.2 Modulation

The signal will be modulated using Morse Code. This is an overview of the Morse alphabet. Other characters, letters and signs are also possible.

| Char | Code | Char | Code | Char | Code | Char | Code |
|---|---|---|---|---|---|---|---|
| A | • – | J | • – – – | S | • • • | 1 | • – – – – |
| B | – • • • | K | – • – | T | – | 2 | • • – – – |
| C | – • – • | L | • – • • | U | • • – | 3 | • • • – – |
| D | – • • | M | – – | V | • • • – | 4 | • • • • – |
| E | • | N | – • | W | • – – | 5 | • • • • • |
| F | • • – • | O | – – – | X | – • • – | 6 | – • • • • |
| G | – – • | P | • – – • | Y | – • – – | 7 | – – • • • |
| H | • • • • | Q | – – • – | Z | – – • • | 8 | – – – • • |
| I | • • | R | • – • | 0 | – – – – – | 9 | – – – – • |

**Table 1 : Morse alphabet**

The symbol (•) is called a **dit,** and the symbol (–) is called a **dah**.

There are four rules to observe to generate correct and understandable Morse code:

- The dah is 3 times longer than the dit.
- The space between a dit and a dah in a character is the length of 1 dit.
- The space between two characters in a word is the length of a dah (= 3 dits).
- The space between two words is the length of 7 dits.

## 4.3 Hardware Beacon Message

The HBG shall emit the callsign HB9EG/1 once every 30 seconds. The HBM will be transmitted only if the EPS is in Recovery mode, or if the EPS microcontroller is no longer functioning.

The desired bit rate is 10 bits/s. Using PARIS as the standard word (it gives us 43 bits/word when we consider a dit as being a bit), this gives us a "speed" of 13.9 wpm.

## 4.4 Software Beacon Message

The software beacon message is divided in 4 parts.

The parts are sent one after the other with an interval of 30 seconds between each sending. Then a complete software beacon message is sent every two minutes.

Before every sent of an entire software beacon message, the information must be updated and the Morse code of the message must be created. These actions are made by the function "updateSBM". Therefore all parts of the message refer to the state of the spacecraft at a same point in time.

As decimal representation is not practical on microcontrollers (the binary to decimal conversion requires not base-2 divisions), numbers are instead transmitted using the octal representation (base-8, digits 0 to 7) which is trivial to implement. A conversion table is available at the end of this document.

#### *4.4.1.1.1 Part 0*

The part 0 of the beacon message contains the SSID of the spacecraft. This identification code is the same code as the hardware beacon message and has for value "HB9EG/1". This code is fixed and every the same all the time of the mission.

##### 4.4.1.1.2 Part 1

The part 1 of the beacon message contains three types of information. The structure of the Morse code is:

header-space-ErrorFlag-space-PowerON

The values are:

- The header of the part
    - Fixed to “1”
- The state of the error flag
    - 5 bits representing the error bit of the five subsystems in following order from MSB to LSB:
        - Payload
        - ADCS
        - CDMS
        - COM
        - EPS
    - These bits are grouped and the value is sent in octal representation
        - For example: receiving '20' is '10000' in binary → payload error bit only.
- The state of the power on flag (which subsystems are turned on or off)
    - 6 bits representing the power on bit of the subsystems in following order from MSB to LSB:
        - ADS
        - Payload
        - ADCS
        - CDMS
        - Beacon
        - COM
    - These bits are grouped and the value is sent in octal representation
        - For example: receiving '23' is '010011' in binary which means Payload, Beacon and COM powered on.

#### *4.4.1.1.3 Part 2*

The part 2 of the beacon message contains two types of information. The structure of the Morse code is:

header-space-BATTERY1VOLTAGE-space-BATTERY2VOLTAGE

The values are:

- The header of the part
  - Fixed to "2"
- The voltage level of the batteries 1 and 2
  - The values are 8-bit and sent in octal representation.
  - They are the non-calibrated raw values of the onboard analog-to-digital converter.
    - The following formula is used to get the value in V: $U = 80x/4095$

#### *4.4.1.1.4 Part 3*

The part 3 of the beacon message contains two types of information. The structure of the Morse code is:

header-space-SolarCells-space-Battery1Temperature

The values are:

- The header of the part
    - Fixed to "3"
- The solar cells produced current
    - The solar cells are in the following order: -X, +X, -Y, +Y, -Z, +Z
    - The produced current is divided in ranges and each range has a value:
        - 0 : 0 ≤ current < 125 mA
        - 1: 125 ≤ current < 250 mA
        - 2: 250 ≤ current < 375 mA
        - 3: 375 ≤ current < 500 mA
        - 4: 500 ≤ current < 625 mA
        - 5: 625 ≤ current < 750 mA
        - 6: 750 ≤ current < 875 mA
        - 7: 875 ≤ current ≤ 1000 mA
    - The six values are sent as one number
        - For example: '203070' means:
            - -X production between 250 and 375 mA
            - -Y production between 375 and 500 mA
            - -Z production between 875 and 1000 mA
            - +X, +Y and +Z production between 0 and 125 mA
- The temperature of the battery 1
        - The value is transformed to fit in 6 bits and to transmit it unsigned as an octal number. The following formula is used to get the correct temperature in °C:
            - $T = 4x - 128$

#### *4.4.1.1.5 Conversion from numerical value to Morse value*

From the moment we know the values to send in the software beacon message; these values must be converted in Morse code.

**<u>Coding each unit in Morse code</u>**

Once the separation is made, the units must be converted in Morse code.

**Warning:** for the software beacon message, we used the abridged format of Morse code for the number. This format is the following:

| Value | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Morse | o_ | oo_ | ooo_ | oooo_ | o | _oooo | _ooo | _oo | _o | _ |

**Figure 1 - Abridged Morse code**

Between two units of the same number, the space is represented by three values set to “0”. Between two different numbers, the space is represented by seven values set to “0”.

# 5 OCTAL REPRESENTATION CONVERSION TABLE

The following table provides the conversion between numbers in octal (Oct), decimal (Dec) and binary (Bin) representations for all number using up to 2 digits in octal (up to 6 bits numbers):

| Oct | Dec | Bin | Oct | Dec | Bin | Oct | Dec | Bin | Oct | Dec | Bin |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 20 | 16 | 10000 | 40 | 32 | 100000 | 60 | 48 | 110000 |
| 1 | 1 | 1 | 21 | 17 | 10001 | 41 | 33 | 100001 | 61 | 49 | 110001 |
| 2 | 2 | 10 | 22 | 18 | 10010 | 42 | 34 | 100010 | 62 | 50 | 110010 |
| 3 | 3 | 11 | 23 | 19 | 10011 | 43 | 35 | 100011 | 63 | 51 | 110011 |
| 4 | 4 | 100 | 24 | 20 | 10100 | 44 | 36 | 100100 | 64 | 52 | 110100 |
| 5 | 5 | 101 | 25 | 21 | 10101 | 45 | 37 | 100101 | 65 | 53 | 110101 |
| 6 | 6 | 110 | 26 | 22 | 10110 | 46 | 38 | 100110 | 66 | 54 | 110110 |
| 7 | 7 | 111 | 27 | 23 | 10111 | 47 | 39 | 100111 | 67 | 55 | 110111 |
| 10 | 8 | 1000 | 30 | 24 | 11000 | 50 | 40 | 101000 | 70 | 56 | 111000 |
| 11 | 9 | 1001 | 31 | 25 | 11001 | 51 | 41 | 101001 | 71 | 57 | 111001 |
| 12 | 10 | 1010 | 32 | 26 | 11010 | 52 | 42 | 101010 | 72 | 58 | 111010 |
| 13 | 11 | 1011 | 33 | 27 | 11011 | 53 | 43 | 101011 | 73 | 59 | 111011 |
| 14 | 12 | 1100 | 34 | 28 | 11100 | 54 | 44 | 101100 | 74 | 60 | 111100 |
| 15 | 13 | 1101 | 35 | 29 | 11101 | 55 | 45 | 101101 | 75 | 61 | 111101 |
| 16 | 14 | 1110 | 36 | 30 | 11110 | 56 | 46 | 101110 | 76 | 62 | 111110 |
| 17 | 15 | 1111 | 37 | 31 | 11111 | 57 | 47 | 101111 | 77 | 63 | 111111 |