

GLOSARIO

Amplificador de alta potencia (HPA, High Power Amplifier): Es el último elemento de una transmisión. Su única función es incrementar la potencia de la portadora proveniente de la salida del *convertidor de subida* al nivel de potencia necesaria para lograr un funcionamiento satisfactorio en el enlace de subida.

Amplificador de bajo ruido (LNA, Low Noise Amplifier): Es utilizado en la recepción como un pre-amplificador. Se encarga de elevar la potencia a un nivel necesario para el procesamiento. La contribución del ruido interno del LNA debe ser lo menor posible de lo contrario la señal de entrada puede ser confundida con el ruido.

Ángulo de inclinación: Es el ángulo entre el plano de referencia (ecuatorial) y otro plano, principalmente el plano orbital del satélite.

Apogeo: Es el punto de la órbita del satélite más alejado del centro de la Tierra.

Azimut: Es el ángulo sobre el cual una antena se mueve sobre el horizonte.

Banda base: Es la señal que no ha sufrido ningún proceso de modulación, es decir, la señal en su frecuencia original.

Cinturón de Van Allen: Son ciertas zonas o anillos toroidales de la magnetósfera terrestre en la que se concentran las partículas cargadas (protones y electrones), las cuales se mueven en forma de espiral entre los polos magnéticos del planeta. Los cinturones de Van Allen son 2, el inferior, de 1 000 km a 5 000 km sobre la superficie de la Tierra, y el cinturón exterior, de 15 000 km a 20 000 km. Estos cinturones se originan debido al intenso campo magnético de la Tierra, originado por su rotación, que atrapa las partículas cargadas (plasma) proveniente del sol (viento solar).

Comunicaciones full dúplex: Comunicación en 2 direcciones, se puede enviar y recibir a través del mismo canal al mismo tiempo.

Día sideral: Periodo que tarda la Tierra en dar una vuelta en su propio eje (aproximadamente 23 h 56 min 4 s).

Dispositivo activo: Es un dispositivo que requiere consumo de energía para que opere.

Dispositivo pasivo: Es un dispositivo que no requiere consumo de energía para que opere.

Elevación: Es el ángulo sobre el cual la antena realiza movimientos perpendiculares al horizonte.

Excentricidad: Es la razón de la distancia entre el centro de la Tierra y el centro de la elipse al semi-eje mayor de la misma, es decir, es el cociente entre los dos semi-ejes de la elipse.

Filtro FIR: Finite Impulse Response, Respuesta finita al impulso. Es un filtro digital en el que si su entrada es una señal impulso, la salida será un número finito de términos no nulos. Para obtener su salida se emplean valores de la entrada actual y de anteriores.

Filtro IIR: Infinite Response Response, Respuesta infinita al impulso. Es un filtro digital en el que si su entrada es una señal impulso, la salida será un número infinito de términos no nulos, es decir nunca vuelve a su estado de reposo. Para obtener su salida se emplean valores de la entrada actual y de anteriores, además de valores de salida anteriores que son realimentados a la entrada.

Frecuencia Intermedia: Es la diferencia constante de frecuencias que se obtiene de la mezcla de la señal recibida en la antena con una frecuencia variable generada por un oscilador local y que guarda con ella una diferencia constante.

Línea de nodos: Es la línea que se proyecta desde el centro de la Tierra hasta el satélite cuando éste corta al Ecuador en su órbita ascendente, es decir, es la recta de intersección del plano de la órbita satelital con el plano del ecuatorial.

Longitud de onda: Es la distancia que existe entre 2 puntos consecutivos que poseen la misma fase, es decir, la distancia a la que se repite la forma de onda.

Nodo ascendente: Es el punto donde la órbita cruza con el plano ecuatorial, yendo de sur a norte.

Nodo descendente: Es el punto donde la órbita cruza con el plano ecuatorial, yendo de norte a sur.

Pérdidas por espacio libre (*Free Space Loss, FSL*): Son aquellas pérdidas de potencia que se miden en el espacio libre sin considerar cualquier tipo de obstáculos.

Perigeo: Es el punto de la órbita del satélite más cercano del centro de la Tierra.

Periodo orbital: Es el tiempo que tarda un satélite en dar una vuelta a su órbita.

PIRE: Potencia Isotrópica Radiada Equivalente. Para satélites, es el producto de la potencia que alimenta la antena transmisora por la ganancia de dicha antena.

Plano ecuatorial: Es el plano que contiene al eje ecuatorial y es perpendicular al eje polar de tal modo que divide a la Tierra en 2 partes iguales denominadas *hemisferios*.

Portadora: Es una forma de onda, generalmente sinusoidal, que es modulada para que la información se pueda transmitir. La portadora es de frecuencia mucho mayor a la de la información (moduladora).

Radiopaquete: Es una tecnología digital utilizada por radioaficionados, que permite la transmisión de información proveniente de una computadora, la cual es transformada a paquetes y enviada a otra estación a través de ondas de radio.

Rango satelital: Distancia entre la estación terrena y el satélite.

S/N o C/N: Relación Señal (Portadora) a Ruido. Es una medida de la cantidad de señal que se tiene con respecto al nivel de ruido.

Rotor: Es el dispositivo que proporciona el movimiento de las antenas para ser apuntadas hacia un satélite determinado.

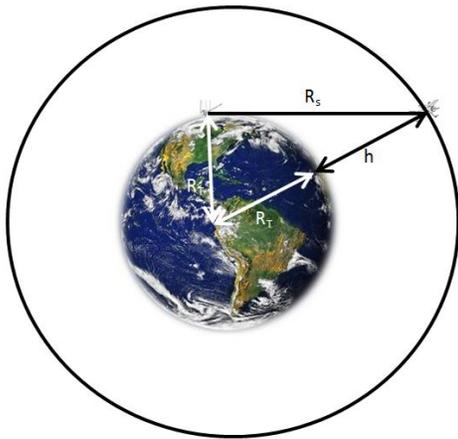
Transpondedor: Un transpondedor es considerado como un retransmisor de microondas, que amplifica y traslada la frecuencia del enlace de subida a una frecuencia menor para el enlace de bajada.

APÉNDICES

APÉNDICE A. CÁLCULO DEL ENLACE SATELITAL

Formulas necesarias

Rango satelital:



$$(R_T + h)^2 = R_S^2 + R_T^2$$

$$R_T^2 + 2R_T h + h^2 = R_S^2 + R_T^2$$

$$2R_T h + h^2 = R_S^2$$

$$R_S = \sqrt{h^2 + 2R_T h}$$

Dónde:

R_S = Rango satelital, en km.

R_T = Radio de la Tierra, en km.

h = Altura del satélite, en km.

Perdidas por espacio libre (FSL):

$$L_{FSL} = \left(\frac{4\pi R_S}{\lambda}\right)^2 = \left(\frac{4\pi R_S f}{c}\right)^2$$

$$L_{FSL}(dB) = 10 \log(L_{FSL})$$

Dónde:

L_{FSL} = Perdidas por espacio libre, en dB.

R_S = Rango satelital, en m.

λ = Longitud de onda, en m.

f = Frecuencia utilizada, en Hz.

c = Velocidad de la luz = 3×10^8 m/s².

PIRE:

$$PIRE = P_{Tx} + G_{Tx} - L_{Tx}$$

Dónde:

$PIRE$ = Potencia Isotrópica Radiada Equivalente, en dBW.

P_{Tx} = Potencia de transmisión, en dBW.

G_{Tx} = Ganancia de la antena transmisora, en dB.

L_{Tx} = Pérdidas en la transmisión, en dB.

Potencia en la recepción:

$$P_{Rx} = PIRE - L_{FSL} - L_{Rx} + G_{Rx} = P_{Tx} + G_{Tx} - \sum L + G_{Rx}$$

Dónde:

P_{Rx} = Potencia en la recepción, en dBW.

$PIRE$ = Potencia Isotrópica Radiada Equivalente, en dBW.

L_{FSL} = Pérdidas por espacio libre, en dB.

L_{Rx} = Perdidas en la recepción, en dB.

G_{Rx} = Ganancia de la antena receptora, en dB.

$\sum L$ = Sumatoria de todas las pérdidas presentadas en el enlace satelital, en dB.

G_{Tx} = Ganancia de la antena transmisora, en dB.

Potencia de ruido:

$$N = kTB$$

Dónde:

N = Potencia de ruido, en W.

k = Constante de Boltzman = $1.38 \times 10^{-23} \text{ J/K}$.

T = Temperatura de ruido del sistema, en K.

B = Ancho de banda del ruido, en Hz.

Ancho de banda de ruido:

$$B = \frac{R_{Tx}}{N} ; N = \log_2 M$$

Dónde:

B = Ancho de banda del ruido, en Hz.

R_{Tx} = Tasa de transmisión del satélite, en bps.

N = Cantidad de bits codificados.

M = Cantidad de condiciones posibles a la salida con N bits.

Relación de portadora a ruido:

$$\frac{C}{N} (\text{dB}) = P_{Rx} - N = P_{Tx} + G_{Tx} - \sum L + G_{Rx} - k - T - B$$

Dónde:

$\frac{C}{N}$ = Relación de portadora a ruido, en dB

P_{Rx} = Potencia en la recepción, en dBW.

N = Potencia de ruido, en W.

P_{Tx} = Potencia de transmisión, en dBW.

G_{Tx} = Ganancia de la antena transmisora, en dB.

$\sum L$ = Sumatoria de todas las pérdidas presentadas en el enlace satelital, en dB.

G_{Rx} = Ganancia de la antena receptora, en dB.

k = Constante de Boltzman = $1.38 \times 10^{-23} \text{ J/K}$.

T = Temperatura de ruido del sistema, en K.

B = Ancho de banda del ruido, en Hz.

Como en esta tesis se trabajó con satélites de órbita baja, el cálculo se realiza para la altura promedio de este tipo de satélites ($h = 1\ 000\ km$).

Obteniendo el rango satelital:

$$R_s = \sqrt{(1\ 000\ km)^2 + 2(6\ 378\ km)(1\ 000\ km)}$$

$$\mathbf{R_s = 3\ 707.02\ km}$$

El cálculo se realiza solamente para una de las bandas de frecuencias en que trabaja la estación terrena (banda de 70cm (430 MHz)). El cálculo para la otra banda (banda de 2m (145 MHz)) se realiza de la misma forma.

Las pérdidas por espacio libre en esta banda son:

$$L_{FSL} = \left(\frac{4\pi R}{\lambda}\right)^2 = \left(\frac{4\pi Rf}{c}\right)^2 = \left(\frac{4\pi(3\ 707\ 020\ m)(430 \times 10^6\ Hz)}{3 \times 10^8 \frac{m}{s}}\right)^2 = 14.45 \times 10^{15}$$

$$L_{FSL}(dB) = 10 \log(14.45 \times 10^{15})$$

$$\mathbf{L_{FSL}(dB) = 156.49\ dB}$$

Cálculo del enlace ascendente en la banda de 70 cm

La potencia mínima que el transceptor de la estación terrena permite para esta banda, es de 5W (6.98 dBW), sabiendo que la ganancia de la antena UHF es de 14 dB y considerando pérdidas en la línea de transmisión de 2 dB:

$$PIRE = 6.98\ dBW + 14\ dB - 2\ dB$$

$$\mathbf{PIRE = 18.98\ dBW}$$

Calculando la Potencia de recepción del satélite, considerando las pérdidas en la recepción de 2 dB y sabiendo que la ganancia de la antena para este tipo de satélites es de 2.14 dB (comúnmente dipolos):

$$P_{Rx} = 6.98\ dBW + 14\ dB - (2\ dB + 156.49\ dB + 2\ dB) + 2.14\ dB$$

$$\mathbf{P_{Rx} = -137.37\ dBW}$$

Obteniendo la potencia de ruido respecto a la tasa de transmisión del satélite (comúnmente 1200 bps para satélites de radioaficionados), modulación BFSK (N=1 y M=2) y suponiendo una temperatura de ruido equivalente del satélite de 273 K:

$$B = \left(\frac{1\,200\text{ bps}}{(\log_2 2)\text{ bits}} \right) = 1\,200\text{ Hz}$$

$$N = \left(1.38 \times 10^{-23} \frac{\text{J}}{\text{K}} \right) (273\text{ K})(1\,200\text{ Hz}) = 4.52 \times 10^{-18}\text{ W}$$

$$N\text{ (dB)} = 10 \log(4.52 \times 10^{-18}) = -173.44\text{ dBW}$$

Finalmente se calcula la relación de portadora a ruido para el enlace descendente:

$$\left(\frac{C}{N} \right)_{asc}\text{ (dB)} = -137.37\text{ dB} - (-173.44\text{ dBW}) = 36.07\text{ dBW}$$

Cálculo del enlace descendente en la banda de 70 cm

Suponiendo una potencia de transmisión del satélite de 1W, una ganancia de la antena del mismo de 2.14 dB por ser dipolo (comúnmente para este tipo de satélites) y pérdidas en la línea de transmisión de 2 dB:

$$PIRE = 0\text{ dBW} + 2.14\text{ dB} - 2\text{ dB}$$

$$PIRE = 0.14\text{ dBW}$$

Calculando la Potencia de recepción en la estación terrena, considerando las pérdidas en la recepción de 2 dB y sabiendo que la ganancia de la antena de UHF de la estación terrena es de 14 dB:

$$P_{Rx} = 0\text{ dBW} + 2.14\text{ dB} - (2\text{ dB} + 156.49\text{ dB} + 2\text{ dB}) + 14\text{ dB}$$

$$P_{Rx} = -144.35\text{ dBW}$$

Obteniendo la potencia de ruido en el ancho de banda respecto a la tasa de transmisión del satélite (comúnmente 1200 bps para satélites de radioaficionados) y suponiendo una temperatura de ruido equivalente de la estación terrena de 300 K:

$$B = \left(\frac{1\,200\text{ bps}}{(\log_2 2)\text{ bits}} \right) = 1\,200\text{ Hz}$$

$$N = \left(1.38 \times 10^{-23} \frac{\text{J}}{\text{K}} \right) (300\text{ K})(1\,200\text{ Hz}) = 4.96 \times 10^{-18}\text{ W}$$

$$N\text{ (dB)} = 10 \log(4.96 \times 10^{-18}) = -173.03\text{ dBW}$$

Finalmente se calcula la relación de portadora a ruido para el enlace descendente:

$$\left(\frac{C}{N}\right)_{desc} (dB) = -144.35 \text{ dB} - (-173.03 \text{ dBW}) = 28.68 \text{ dBW}$$

Después de los resultados obtenidos, se concluye que para el enlace ascendente al obtener que la $P_{Rx} = -137.37 \text{ dBW}$ (-107.37 dBm), el receptor del satélite debe tener como mínimo una sensibilidad de este valor para que pueda recibir la señal transmitida.

Por otro lado, para el enlace descendente, de acuerdo a las especificaciones de la estación terrena utilizada para esta tesis ($C/N > 10 \text{ dB}$), se tiene una relación de portadora a ruido muy aceptable:

$$\left(\frac{C}{N}\right)_{desc} > \left(\frac{C}{N}\right)_{ET}$$
$$27.89 \text{ dBW} > 10 \text{ dBW}$$

Mientras se tenga una mayor relación de portadora a ruido, se puede transmitir a una distancia mayor, por lo que al tener una gran diferencia de C/N respecto a la mínima necesaria para permitir la comunicación entre la estación y el satélite, la información del satélite se recibe con mucha mayor fiabilidad.

Así mismo al obtener que la $P_{Rx} = -144.35 \text{ dBW}$ (-114.35 dBm) y que de acuerdo a las especificaciones del transceptor utilizado en la estación terrena tiene una potencia mínima de recepción de -126.16 dBm, por lo que al requerir -114.35 dBm siendo un valor mayor al que el receptor es capaz de recibir, se puede concluir que el transceptor es capaz de recibir perfectamente la señal sin necesidad de un preamplificador.

APÉNDICE B. CÓDIGO MORSE

Características del Código Morse

- ◆ Cada punto representa 1 unidad y cada raya representa 3 unidades.
- ◆ En una misma letra, la separación que deben tener sus símbolos es de aproximadamente la duración de 1 punto.
- ◆ La separación entre las letras de una misma palabra es de aproximadamente 3 puntos.
- ◆ La separación entre palabras, es de aproximadamente 3 rayas.

| <i>LETRA</i> | <i>CÓDIGO</i> | <i>NÚMERO</i> | <i>CÓDIGO</i> | <i>SÍMBOLO</i> | <i>CÓDIGO</i> |
|--------------|---------------|---------------|---------------|----------------|----------------------------|
| A | .- | 0 | ----- | . | .-.-.- |
| B | -... | 1 | .---- | , | --..-- |
| C | -.-. | 2 | ..--- | ? | ..--.. |
| D | -.. | 3 | ...-- | = | -...- |
| E | . | 4 |- | - | -....- |
| F | ..-. | 5 | | / | -...- |
| G | --. | 6 | -.... | " | .-..- |
| H | | 7 | --... | Error | (Más de 5 puntos) |
| I | .. | 8 | ---.. | | |
| J | .---- | 9 | ----. | | |
| K | -.- | | | | |
| L | .-.. | | | | |
| M | -- | | | | |
| N | -. | | | | |
| O | --- | | | | |
| P | .-. | | | | |
| Q | --.- | | | | |
| R | .-. | | | | |
| S | ... | | | | |
| T | - | | | | |
| U | ..- | | | | |
| V | ...- | | | | |
| W | .--- | | | | |
| X | -..- | | | | |
| Y | -.-- | | | | |
| Z | --.. | | | | |

APÉNDICE C. FÓRMULAS DE CONVERSIÓN DE TELEMETRÍA

SEEDS II

Formato: Telemetría en CW.

| HK data long mode | | | | |
|---|---------------|----------|--------------------------------|--|
| Data | Digits number | Notation | Designation | Conversion equation |
| JQ1YGU | 6 | Alphabet | Call sign | — |
| SEEDS | 5 | Alphabet | Satellite name | — |
| G4 | 2 | Alphabet | Mode | — |
| 0 ₇ 0 ₆ 0 ₅ 0 ₄ 0 ₃ 0 ₂ 0 ₁ 0 ₀ | 8 | HEX | Satellite time | $(0_7 \times 16^7 + 0_6 \times 16^6 + 0_5 \times 16^5 + 0_4 \times 16^4 + 0_3 \times 16^3 + 0_2 \times 16^2 + 0_1 \times 16^1 + 0_0 \times 16^0) / 2$ [s] |
| 1 ₂ 1 ₁ 1 ₀ | 3 | HEX | Li-ion batteries voltage | $5 \times (1_2 \times 16^2 + 1_1 \times 16^1 + 1_0 \times 16^0) / 4096$ [V] |
| 2 ₂ 2 ₁ 2 ₀ | 3 | HEX | Bus voltage | $5 \times (2_2 \times 16^2 + 2_1 \times 16^1 + 2_0 \times 16^0) / 4096$ [V] |
| 3 ₂ 3 ₁ 3 ₀ | 3 | HEX | Solar cell 1 current | $5 \times (3_2 \times 16^2 + 3_1 \times 16^1 + 3_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 4 ₂ 4 ₁ 4 ₀ | 3 | HEX | Solar cell 2 current | $5 \times (4_2 \times 16^2 + 4_1 \times 16^1 + 4_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 5 ₂ 5 ₁ 5 ₀ | 3 | HEX | Solar cell 3 current | $5 \times (5_2 \times 16^2 + 5_1 \times 16^1 + 5_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 6 ₂ 6 ₁ 6 ₀ | 3 | HEX | Solar cell 4 current | $5 \times (6_2 \times 16^2 + 6_1 \times 16^1 + 6_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 7 ₂ 7 ₁ 7 ₀ | 3 | HEX | Solar cell 5 current | $5 \times (7_2 \times 16^2 + 7_1 \times 16^1 + 7_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 8 ₂ 8 ₁ 8 ₀ | 3 | HEX | Solar cell 6 current | $5 \times (8_2 \times 16^2 + 8_1 \times 16^1 + 8_0 \times 16^0) / 4096 \times 90.90909$ [mA] |
| 9 ₂ 9 ₁ 9 ₀ | 3 | HEX | Temperature (Li-ion battery 1) | $0.15797 \times (5 \times (9_2 \times 16^2 + 9_1 \times 16^1 + 9_0 \times 16^0) / 4096) - 39.553 \times (5 \times (9_2 \times 16^2 + 9_1 \times 16^1 + 9_0 \times 16^0) / 4096) + 129.59$ [deg. C] |
| A ₂ A ₁ A ₀ | 3 | HEX | Temperature (Li-ion battery 2) | $0.18923 \times (5 \times (A_2 \times 16^2 + A_1 \times 16^1 + A_0 \times 16^0) / 4096) - 39.27 \times (5 \times (A_2 \times 16^2 + A_1 \times 16^1 + A_0 \times 16^0) / 4096) + 128.33$ [deg. C] |
| B ₂ B ₁ B ₀ | 3 | HEX | Temperature (Transmitter) | $-0.38082 \times (5 \times (B_2 \times 16^2 + B_1 \times 16^1 + B_0 \times 16^0) / 4096) - 36.125 \times (5 \times (B_2 \times 16^2 + B_1 \times 16^1 + B_0 \times 16^0) / 4096) + 121.31$ [deg. C] |
| C ₂ C ₁ C ₀ | 3 | HEX | Temperature (Receiver) | $-0.062626 \times (5 \times (C_2 \times 16^2 + C_1 \times 16^1 + C_0 \times 16^0) / 4096) - 38.305 \times (5 \times (C_2 \times 16^2 + C_1 \times 16^1 + C_0 \times 16^0) / 4096) + 126.89$ [deg. C] |
| D ₀ | 1 | HEX | CW transmission interval | $D_0 \times 3$ [s] |
| E ₀ | 1 | HEX | Switch status | Convert to binary. Its least 3 digits from LSB to 3 rd bit represent the status of Switch1, Switch2 and Switch3 respectively. MSB should be ignored. [EXAMPLE] When E ₀ =1(HEX)=0001(BIN). S1(ON), S2(OFF), S3(OFF). |
| F ₃ F ₂ F ₁ F ₀ | 4 | HEX | MPU reset times (EPS) | $F_3 \times 16^3 + F_2 \times 16^2 + F_1 \times 16^1 + F_0 \times 16^0$ [times] |
| G ₃ G ₂ G ₁ G ₀ | 4 | HEX | MPU reset times (FMR) | $G_3 \times 16^3 + G_2 \times 16^2 + G_1 \times 16^1 + G_0 \times 16^0$ [times] |
| H ₃ H ₂ H ₁ H ₀ | 4 | HEX | MPU reset times (C&DH) | $H_3 \times 16^3 + H_2 \times 16^2 + H_1 \times 16^1 + H_0 \times 16^0$ [times] |

| Data | Digits number | Notation | Designation | Conversion equation |
|---|---------------|----------|-----------------------|---|
| I ₃ I ₂ I ₁ I ₀ | 4 | HEX | MPU reset times (CW) | $I_3 \times 16^3 + I_2 \times 16^2 + I_1 \times 16^1 + I_0 \times 16^0$ [times] |
| J ₃ J ₂ J ₁ J ₀ | 4 | HEX | CW transmission count | $J_3 \times 16^3 + J_2 \times 16^2 + J_1 \times 16^1 + J_0 \times 16^0$ [times] |
| K ₁ K ₀ | 2 | HEX | Uplink count | $K_1 \times 16^1 + K_0 \times 16^0$ [times] |
| M ₁ M ₀ | 2 | HEX | Command status | $M_1 \times 16^1 + M_0 \times 16^0$ [-] |
| N ₀ | 1 | HEX | Battery status | <p>Convert to binary. Its 4 digits from LSB represent the status of "larger than 3.0V", "larger than 4.0V", "larger than 4.2V", and "forced no-charge mode" respectively.</p> <p>[EXAMPLE] When N₀=3(HEX)=0011(BIN). Battery voltage is larger than 4.0V and smaller than 4.2V, and forced no-charge mode is OFF.</p> |
| O ₀ | 1 | HEX | Shunt circuit status | <p>Convert to binary. Its 2 digits from LSB represent the mode of shunt circuit, and next 1 digit represents the status of shunt circuit. 4th digit should be ignored.</p> <p>The mode of shunt circuit is classified as follows.</p> <p>00 : Auto shunt mode 01 : Forced shunt mode 10 : Forced no-shunt mode</p> <p>[EXAMPLE] When O₀=5(HEX)=0101(BIN). The mode of shunt circuit is forced shunt mode, and shunt circuit is performing now.</p> |

SwissCube

Formato: Telemetría en CW.

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: $0 \leq \text{current} < 125 \text{ mA}$
 - 1: $125 \leq \text{current} < 250 \text{ mA}$
 - 2: $250 \leq \text{current} < 375 \text{ mA}$
 - 3: $375 \leq \text{current} < 500 \text{ mA}$
 - 4: $500 \leq \text{current} < 625 \text{ mA}$
 - 5: $625 \leq \text{current} < 750 \text{ mA}$
 - 6: $750 \leq \text{current} < 875 \text{ mA}$
 - 7: $875 \leq \text{current} \leq 1000 \text{ mA}$
 - 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:
 - $T = 4x - 128$

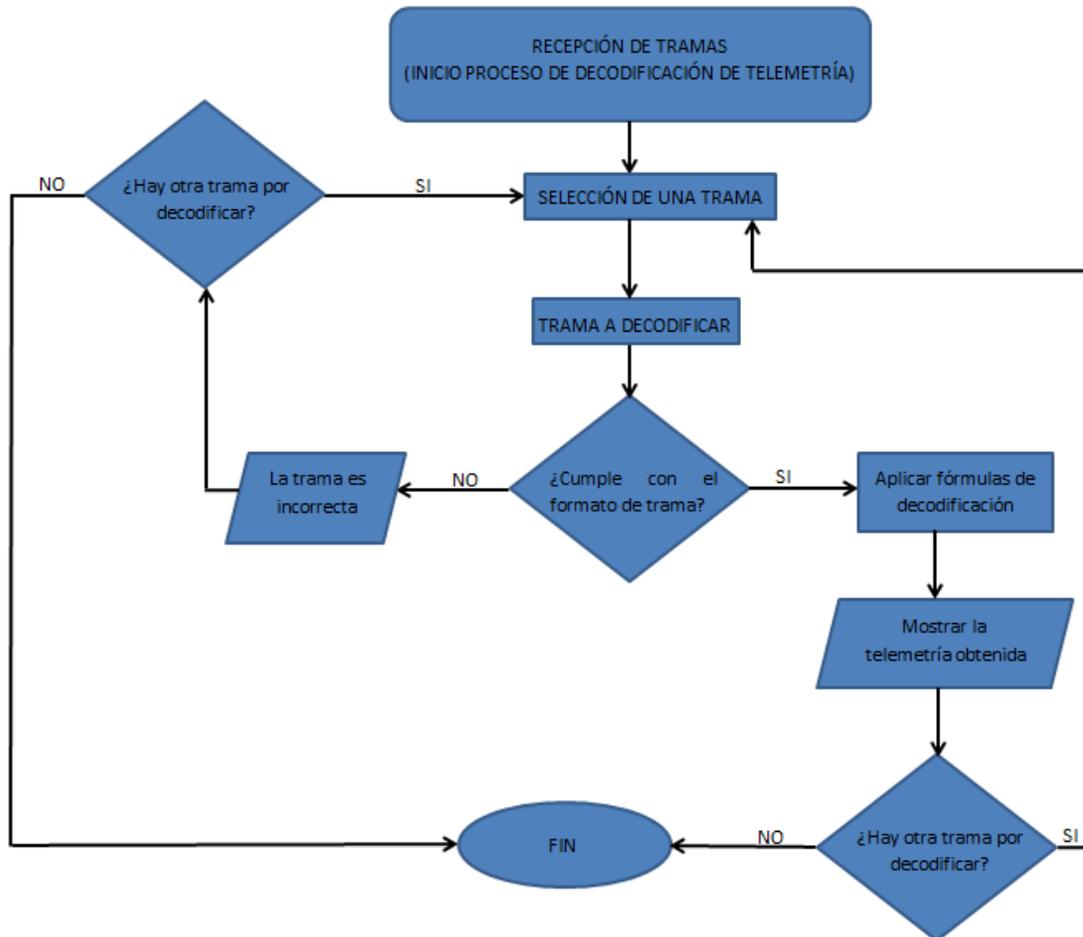
HO-68

Formato: Telemetría en CW.

| Channel | Parameter | Type | Data Format | | Description and Equation | Unit |
|---------|---|--------|-------------|--------|--|------|
| | | | N(min) | N(max) | | |
| CH1 | PA Output RF Switch Status | Status | 000 | 111 | 111 =PA2 Works (Beacon only) 000 = PA1 Works (Transponder and Beacon) | |
| CH2 | Transponder Working Status | Status | 000 | 111 | 000= Beacon only 001= Beacon and Linear Transponder 010= Beacon and FM Transponder 100= Upload Software | |
| CH3 | Transponder Temperature | Data | 099 | 199 | First character =0, T= -N First character=1, T= +Last Two character | °C |
| CH4 | Beacon RF Output Power | Data | 000 | 999 | $P=N$ | mW |
| CH5 | Beacon Power Supply Voltage | Data | 000 | 999 | $V=N/100$ | V |
| CH6 | Receiver Power Supply Current | Data | 000 | 999 | $I=N$ | mA |
| CH7 | Linear Transponder AGC Voltage | Data | 000 | 999 | $V=N/100$ | V |
| CH8 | Transponder RF Output Power | Data | 000 | 999 | $P = N \times 3$ | mW |
| CH9 | Transponder PA Power Supply Current | Data | 000 | 999 | $I=N$ | mA |
| CH10 | Linear Transponder Up converter Power Supply Current | Data | 000 | 999 | $I=N$ | mA |
| CH11 | Linear Transponder Power Supply Voltage | Data | 000 | 999 | $V=N/100$ | V |
| CH12 | FM/Digital Store-forward Transponder Digital Power Supply Current | Data | 000 | 999 | $I=N$ | mA |
| CH13 | FM/Digital Store-forward Transponder Power Supply Voltage | Data | 000 | 999 | $V=N/100$ | V |

APÉNDICE D. CÓDIGO FUENTE DEL SOFTWARE DE MATLAB

Diagrama de flujo general de los programas elaborados en Matlab.



***Nota:** Se utilizó la versión de Matlab 7.6.0.324 (R2008a). Únicamente se muestra la parte medular del código fuente de cada programa.

SEEDS II. CÓDIGO FUENTE

```
function pushbutton1_Callback(hObject, eventdata, handles)

telemetria_CO66=get(handles.edit1,'String');
espacios_telemetria_CO66=findstr(telemetria_CO66,' ');
telemetria(1,1:6)=telemetria_CO66(1:espacios_telemetria_CO66(1)-1);

for i=1:(length(espacios_telemetria_CO66)-1)
telemetria(i+1,1:length(telemetria_CO66(espacios_telemetria_CO66(i)+1:espacios_telemetria_CO66(i+1)-1)))=telemetria_CO66(espacios_telemetria_CO66(i)+1:espacios_telemetria_CO66(i+1)-1);
end

telemetria(length(espacios_telemetria_CO66)+1,1:length(telemetria_CO66(espacios_telemetria_CO66(length(espacios_telemetria_CO66))+1:length(telemetria_CO66))))=telemetria_CO66(espacios_telemetria_CO66(length(espacios_telemetria_CO66))+1:length(telemetria_CO66)));
a=1;
call_sign=telemetria_CO66(1:espacios_telemetria_CO66(a)-1)
a=a+1;
satellite_name=telemetria_CO66(espacios_telemetria_CO66(a-1)+1:espacios_telemetria_CO66(a)-1)
a=a+1;
mode=telemetria_CO66(espacios_telemetria_CO66(a-1)+1:espacios_telemetria_CO66(a)-1)

%CONVERSIONES
b=8;
c=3;
d=2;
e=4;
satellite_time=(hex2dec(telemetria(4,1:b)))
bateries_voltage=5*(hex2dec(telemetria(5,1:c)))/4096
bus_voltage=5*(hex2dec(telemetria(6,1:c)))/4096
solar_cell1=(5*(hex2dec(telemetria(7,1:c)))/4096)*90.90909
solar_cell2=(5*(hex2dec(telemetria(8,1:c)))/4096)*90.90909
solar_cell3=(5*(hex2dec(telemetria(9,1:c)))/4096)*90.90909
solar_cell4=(5*(hex2dec(telemetria(10,1:c)))/4096)*90.90909
solar_cell5=(5*(hex2dec(telemetria(11,1:c)))/4096)*90.90909
solar_cell6=(5*(hex2dec(telemetria(12,1:c)))/4096)*90.90909
Temperature_battery1=0.15797*(5*(hex2dec(telemetria(13,1:c)))/4096)^2-39.553*(5*(hex2dec(telemetria(13,1:c)))/4096)+129.59
Temperature_battery2=0.18923*(5*(hex2dec(telemetria(14,1:c)))/4096)^2-39.27*(5*(hex2dec(telemetria(14,1:c)))/4096)+128.33
Temperature_transmitter=-0.38082*(5*(hex2dec(telemetria(15,1:c)))/4096)^2-36.125*(5*(hex2dec(telemetria(15,1:c)))/4096)+121.31
Temperature_receiver=-0.062626*(5*(hex2dec(telemetria(16,1:c)))/4096)^2-38.305*(5*(hex2dec(telemetria(16,1:c)))/4096)+126.89
CW_transmission_interval=hex2dec(telemetria(17,1))*3

switch_status=dec2bin(hex2dec(telemetria(17,2)))
switch_status_1=zeros(1,4);
restal=length(switch_status_1)-length(switch_status);
```

```

if length(switch_status)< 4
    for i=1: length(switch_status)
        switch_status_1(restal+i)= str2num(switch_status(i));
    end
else
    switch_status_1= switch_status;
end

switch_status_1=num2str(switch_status_1);
MPU_EPS=hex2dec(telemetry(18,1:e))
MPU_FMR=hex2dec(telemetry(19,1:e))
MPU_CDH=hex2dec(telemetry(20,1:e))
MPU_CW=hex2dec(telemetry(21,1:e))
CW_transmission_count=hex2dec(telemetry(22,1:e))
uplink_count=hex2dec(telemetry(23,1:d))
command_status=hex2dec(telemetry(24,1:d))

Battery_status=dec2bin(hex2dec(telemetry(25,1)));
Battery_status_1=zeros(1,4);
restal=length(Battery_status_1)-length(Battery_status);

if length(Battery_status)< 4
    for i=1: length(Battery_status)
        Battery_status_1(restal+i)= str2num(Battery_status(i));
    end
else
    Battery_status_1= Battery_status;
End

Battery_status_1=num2str(Battery_status_1);

Shunt_circuit_status=dec2bin(hex2dec(telemetry(25,2)))
Shunt_circuit_status_1=zeros(1,4);
restal=length(Shunt_circuit_status_1)-length(Shunt_circuit_status);

if length(Shunt_circuit_status)< 4
    for i=1: length(Shunt_circuit_status)
        Shunt_circuit_status_1(restal+i)=
str2num(Shunt_circuit_status(i));
    end
else
    Shunt_circuit_status_1= Shunt_circuit_status;
end

Shunt_circuit_status_1=num2str(Shunt_circuit_status_1);

%%
handles.var1=telemetry_CO66;
set(handles.edit2, 'string', call_sign);
set(handles.edit3, 'string', satellite_name);
set(handles.edit4, 'string', mode);
set(handles.edit5, 'string', satellite_time);
set(handles.edit6, 'string', bateries_voltage);
set(handles.edit7, 'string', bus_voltage);
set(handles.edit8, 'string', solar_cell1);
set(handles.edit9, 'string', solar_cell2);
set(handles.edit10, 'string', solar_cell3);

```

```

set(handles.edit11, 'string', solar_cell4);
set(handles.edit12, 'string', solar_cell5);
set(handles.edit13, 'string', solar_cell6);
set(handles.edit14, 'string', Temperature_batery1);
set(handles.edit15, 'string', Temperature_batery2);
set(handles.edit16, 'string', Temperature_transmitter);
set(handles.edit17, 'string', Temperature_receiver);
set(handles.edit18, 'string', CW_transmission_interval);
set(handles.edit19, 'string', switch_status_1);
set(handles.edit20, 'string', MPU_EPS);
set(handles.edit21, 'string', MPU_FMR);
set(handles.edit22, 'string', MPU_CDH);
set(handles.edit23, 'string', MPU_CW);
set(handles.edit24, 'string', CW_transmission_count);
set(handles.edit25, 'string', uplink_count);
set(handles.edit26, 'string', command_status);
set(handles.edit27, 'string', Battery_status_1);
set(handles.edit28, 'string', Shunt_circuit_status_1);

%set(handles.edit1, 'string', '1');
guidata(hObject, handles);

```

SwissCube. CÓDIGO FUENTE

```

function pushbutton1_Callback(hObject, eventdata, handles)
Parte0=get(handles.edit1, 'String');
Parte1=get(handles.edit63, 'String');
Parte2=get(handles.edit64, 'String');
Parte3=get(handles.edit65, 'String');
Parte1=str2num(Parte1);
Parte2=str2num(Parte2);
Parte3=str2num(Parte3);

%% DECODIFICACIÓN PARTE1
Parte1_dec1=oct2dec(Parte1(2));
Parte1_bin1=dec2bin(Parte1_dec1);
Parte1_dec2=oct2dec(Parte1(3));
Parte1_bin2=dec2bin(Parte1_dec2);

Error_Flag1_1= Parte1_bin1;
Power_ON1_1= Parte1_bin2;

% ERROR FLAG
Error_Flag1_1_2=zeros(1,5);
restal=length(Error_Flag1_1_2)-length(Error_Flag1_1);
if length(Error_Flag1_1)< 5
    for i=1: length(Error_Flag1_1)
        Error_Flag1_1_2(restal+i)= str2num(Error_Flag1_1(i));
    end
else
    Error_Flag1_1_2= Error_Flag1_1;
end
encuentra=find(Error_Flag1_1_2);
a=0;

```

```

for i=1:length(encuentra)
if encuentra(i)==1
error_en_subsistema(i,:)='Payload';
disp('error_en_subsistema: Payload');
a=a+1;
end
if encuentra(i)==2
error_en_subsistema(i,:)='ADCS  ';
disp('error_en_subsistema: ADCS');
a=a+1;
end
if encuentra(i)==3
error_en_subsistema(i,:)='CDMS  ';
disp('error_en_subsistema: CDMS');
a=a+1;
end
if encuentra(i)==4
error_en_subsistema(i,:)='COM   ';
disp('error_en_subsistema: COM');
a=a+1;
end
if encuentra(i)==5
error_en_subsistema(i,:)='EPS   ';
disp('error_en_subsistema: EPS');
a=a+1;
end
end
if a==0
for i=1:5
error_en_subsistema(i,:)='no errores';
end
disp('no hay error flag');
end
error_en_subsistema

%POWER ON
Power_ON1_1_2=zeros(1,6);
restal_2=length(Power_ON1_1_2)-length(Power_ON1_1);

if length(Power_ON1_1)< 6
for i=1: length(Power_ON1_1)
Power_ON1_1_2(restal_2+i)= str2num(Power_ON1_1(i));
end
else
Power_ON1_1_2 = Power_ON1_1_2;
end
encuental=find(Power_ON1_1_2);
state_power=['OFF'; 'OFF'; 'OFF' ;'OFF' ;'OFF' ;'OFF'];
for i=1:length(encuental)
if encuental(i)==1
state_power(1,:)='ON ';
disp('state power ADS ON')
end
if encuental(i)==2
state_power(2,:)='ON ';
disp('state power PAYLOAD ON')
end
end

```

```

if encuentral(i)==3
    state_power(3,:)= 'ON ';
    disp('state power ADCS ON')
end
if encuentral(i)==4
    state_power(4,:)= 'ON ';
    disp('state power CDMS ON')
end
if encuentral(i)==5
    state_power(5,:)= 'ON ';
    disp('state power BEACON ON')
end
if encuentral(i)==6
    state_power(6,:)= 'ON ';
    disp('state power COM ON')
end
end

%% DECODIFICACIÓN PARTE2
Parte2_dec1=oct2dec(Parte2(2));
Parte2_dec2=oct2dec(Parte2(3));

Battery1Voltage_2= 80*(Parte2_dec1)/4095
Battery2Voltage_2= 80*(Parte2_dec2)/4095

%% DECODIFICACIÓN PARTE3
Parte3_dec2=oct2dec(Parte3(3));
Parte3_bin2=dec2bin(Parte3_dec2);

SolarCells_3= num2str(Parte3(2));

for i=1:length(SolarCells_3)
    if SolarCells_3(i) == '0'
        sollar_cells_current(i,:)= 'de 0 a 125mA ';
    else
    if SolarCells_3(i) == '1'
        sollar_cells_current(i,:)= 'de 125 a 250mA ';
    else
    if SolarCells_3(i) == '2'
        sollar_cells_current(i,:)= 'de 250 a 375mA ';
    else
    if SolarCells_3(i) == '3'
        sollar_cells_current(i,:)= 'de 375 a 500mA ';
    else
    if SolarCells_3(i) == '4'
        sollar_cells_current(i,:)= 'de 500 a 625mA ';
    else
    if SolarCells_3(i) == '5'
        sollar_cells_current(i,:)= 'de 625 a 750mA ';
    else
    if SolarCells_3(i) == '6'
        sollar_cells_current(i,:)= 'de 750 a 875mA ';
    else
    if SolarCells_3(i) == '7'
        sollar_cells_current(i,:)= 'de 875 a 1000mA';
    else
end
end

```

```

end
end
end
end
end
end
end
end
end
sollar_cells_current
Battery1Temperature_3= 4*(Parte3_dec2)-128

%%
%handles.var1=telemetria_CO66;
set(handles.edit13, 'string', Parte0);
%partel
set(handles.edit2, 'string', error_en_subsistema(1,:));
set(handles.edit3, 'string', error_en_subsistema(2,:));
set(handles.edit4, 'string', error_en_subsistema(3,:));
set(handles.edit5, 'string', error_en_subsistema(4,:));
set(handles.edit6, 'string', error_en_subsistema(5,:));
set(handles.edit7, 'string', state_power(1,:));
set(handles.edit8, 'string', state_power(2,:));
set(handles.edit9, 'string', state_power(3,:));
set(handles.edit10, 'string', state_power(4,:));
set(handles.edit11, 'string', state_power(5,:));
set(handles.edit12, 'string', state_power(6,:));
%parte2
set(handles.edit14, 'string', Battery1Voltage_2);
set(handles.edit15, 'string', Battery2Voltage_2);
%parte3
set(handles.edit16, 'string', sollar_cells_current(1,:));
set(handles.edit17, 'string', sollar_cells_current(2,:));
set(handles.edit18, 'string', sollar_cells_current(3,:));
set(handles.edit19, 'string', sollar_cells_current(4,:));
set(handles.edit20, 'string', sollar_cells_current(5,:));
set(handles.edit21, 'string', sollar_cells_current(6,:));
set(handles.edit22, 'string', Battery1Temperature_3);

%set(handles.edit1, 'string', '1');
guidata(hObject, handles);

```

HO-68. CÓDIGO FUENTE

```

function pushbutton1_Callback(hObject, eventdata, handles)
telemetriaHO68=get(handles.edit1, 'String');
espacios_telemetriaHO68=findstr(telemetriaHO68, ' ');
telemetria(1,1:5)=telemetriaHO68(1:espacios_telemetriaHO68(1)-1);

for i=1:(length(espacios_telemetriaHO68)-1)
telemetria(i+1,1:length(telemetriaHO68(espacios_telemetriaHO68(i)+1:espacios_telemetriaHO68(i+1)-1)))=telemetriaHO68(espacios_telemetriaHO68(i)+1:espacios_telemetriaHO68(i+1)-1);
end

```



```

disp('CH7=Voltaje del AGC del transpondedor lineal')
disp('CH8=Potencia RF de salida del transpondedor')
disp('CH9=Corriente del transpondedor del Amplificador de Potencia')
disp('CH10=Corriente del transpondedor lineal ascendente')
disp('CH11=Voltaje del transpondedor lineal')
disp('CH12=Corriente del transpondedor FM/Digital')
disp('CH13=Voltaje del transpondedor FM/Digital')

%CH1
if telemetrias(4,1:3) == '111'
CH1='PA2 works (only beacon)';
else
    if telemetrias(4,1:3) == '000'
CH1='PA1 works (Transponder and beacon)';
    else
        CH1=str2num(telemetrias(4,1:3));
    end
end
CH1

%CH2
if telemetrias(5,1:3) == '000'
CH2='Beacon only';
else
    if telemetrias(5,1:3) == '001'
CH2='Beacon and Linear Transponder';
    else
        if telemetrias(5,1:3) == '010'
CH2='Beacon and FM Transponder';
        else
            if telemetrias(5,1:3) == '100'
CH2='Upload Software';
            else
                CH2=str2num(telemetrias(5,1:3))
            end
        end
    end
end
CH2

%CH3
if telemetrias(6,1)=='0'
    CH3=str2num(telemetrias(6,1:3));
    CH3=-CH3;
else
    if telemetrias(6,1)=='1'
CH3=str2num(telemetrias(6,2:3))
    end
end

CH4=str2num(telemetrias(7,1:3))
CH5=str2num(telemetrias(8,1:3))/100
CH6=str2num(telemetrias(9,1:3))
CH7=str2num(telemetrias(10,1:3))/100
CH8=str2num(telemetrias(11,1:3))*3
CH9=str2num(telemetrias(12,1:3))
CH10=str2num(telemetrias(13,1:3))

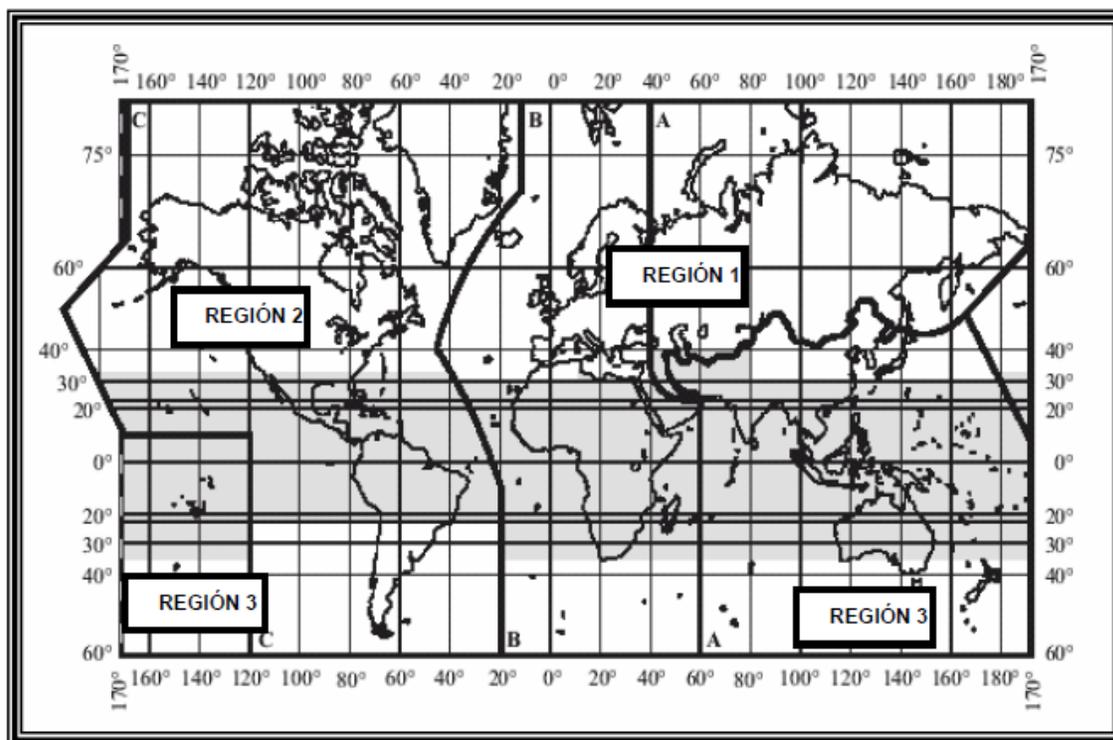
```

```
CH11=str2num(telemetrias(14,1:3))/100
CH12=str2num(telemetrias(15,1:3))
CH13=str2num(telemetrias(16,1:3))/100
```

```
%%
%handles.var1=telemetria_CO66;
set(handles.edit2,'string',CH1);
set(handles.edit3,'string',CH2);
set(handles.edit5,'string',CH3);
set(handles.edit6,'string',CH4);
set(handles.edit7,'string',CH5);
set(handles.edit8,'string',CH6);
set(handles.edit9,'string',CH7);
set(handles.edit10,'string',CH8);
set(handles.edit11,'string',CH9);
set(handles.edit12,'string',CH10);
set(handles.edit13,'string',CH11);
set(handles.edit14,'string',CH12);
set(handles.edit15,'string',CH13);

%set(handles.edit1,'string','1');
guidata(hObject, handles);
```

APÉNDICE E. MAPA MUNDIAL DE REGIONES DE FRECUENCIAS PARA RADIOCOMUNICACIONES



Región 1: Europa y África.

Región 2: América.

Región 3: Asia y Oceanía.