

TRABAJO PRÁCTICO TÓPICOS DE PROGRAMACIÓN

Ivan - Luca - Flor

Consigna completa

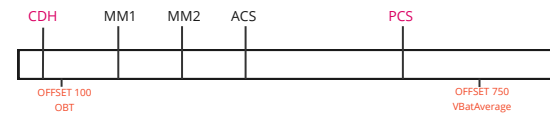
Subsistemas

- La tabla 1 contiene la posición en bytes donde inicia el volcado de información cada subsistema, observe que **PCS en el byte 1604** y **CDH (Command and data handling) en el byte 8**.

 Lo uso en la parte 1

 Lo uso en la parte 2

2.1 TELEMETRY FRAME BREAKDOWN			
Mx	Name	Description	Size [B]
		...	
8	CDH1	CDH1	272
280	MM1	MM1	150
430	MM2	MM2	150
580	ACS	ACS	1024
1604	PCS	PCS	1024



PARTE 1

PARTE 2

A

- El archivo contiene una cantidad desconocida de registros o paquetes, cada registro contiene 4000 bytes, controle que la cantidad de bytes del archivo dividido 4000 de resto 0. Caso contrario aborte el proceso

2.1 TELEMETRY FRAME BREAKDOWN			
Id	Name	Description	Size [B]

8	CDM	CDM	272
200	IMBT	IMBT	1581
479	IMSG	IMSG	158
580	ACB	ACB	1504
1024	PCB	PCB	1024



32398 Registros

B

- El valor de voltaje es informado por el subsistema de potencia (PCS), según documentación del fabricante el "offset" o desplazamiento dentro del área del subsistema para encontrar vBatAverage es de 750 bytes, la Tabla 2 muestra un extracto de la documentación.

ID	OFF_TSL_ID/TSL_Vari_aid	DESCRIPTION	TYPE	FRAM	RANGE / EXPECTED VALUE	ALARM

001	TV	TV	TV	TV	TV	TV
002	TV	TV	TV	TV	TV	TV
003	TV	TV	TV	TV	TV	TV
004	TV	TV	TV	TV	TV	TV
005	TV	TV	TV	TV	TV	TV
006	TV	TV	TV	TV	TV	TV
007	TV	TV	TV	TV	TV	TV
008	TV	TV	TV	TV	TV	TV
009	TV	TV	TV	TV	TV	TV
010	TV	TV	TV	TV	TV	TV
011	TV	TV	TV	TV	TV	TV
012	TV	TV	TV	TV	TV	TV
013	TV	TV	TV	TV	TV	TV
014	TV	TV	TV	TV	TV	TV
015	TV	TV	TV	TV	TV	TV
016	TV	TV	TV	TV	TV	TV
017	TV	TV	TV	TV	TV	TV
018	TV	TV	TV	TV	TV	TV
019	TV	TV	TV	TV	TV	TV
020	TV	TV	TV	TV	TV	TV
021	TV	TV	TV	TV	TV	TV
022	TV	TV	TV	TV	TV	TV
023	TV	TV	TV	TV	TV	TV
024	TV	TV	TV	TV	TV	TV
025	TV	TV	TV	TV	TV	TV
026	TV	TV	TV	TV	TV	TV
027	TV	TV	TV	TV	TV	TV
028	TV	TV	TV	TV	TV	TV
029	TV	TV	TV	TV	TV	TV
030	TV	TV	TV	TV	TV	TV
031	TV	TV	TV	TV	TV	TV
032	TV	TV	TV	TV	TV	TV
033	TV	TV	TV	TV	TV	TV
034	TV	TV	TV	TV	TV	TV
035	TV	TV	TV	TV	TV	TV
036	TV	TV	TV	TV	TV	TV
037	TV	TV	TV	TV	TV	TV
038	TV	TV	TV	TV	TV	TV
039	TV	TV	TV	TV	TV	TV
040	TV	TV	TV	TV	TV	TV
041	TV	TV	TV	TV	TV	TV
042	TV	TV	TV	TV	TV	TV
043	TV	TV	TV	TV	TV	TV
044	TV	TV	TV	TV	TV	TV
045	TV	TV	TV	TV	TV	TV
046	TV	TV	TV	TV	TV	TV
047	TV	TV	TV	TV	TV	TV
048	TV	TV	TV	TV	TV	TV
049	TV	TV	TV	TV	TV	TV
050	TV	TV	TV	TV	TV	TV
051	TV	TV	TV	TV	TV	TV
052	TV	TV	TV	TV	TV	TV
053	TV	TV	TV	TV	TV	TV
054	TV	TV	TV	TV	TV	TV
055	TV	TV	TV	TV	TV	TV
056	TV	TV	TV	TV	TV	TV
057	TV	TV	TV	TV	TV	TV
058	TV	TV	TV	TV	TV	TV
059	TV	TV	TV	TV	TV	TV
060	TV	TV	TV	TV	TV	TV
061	TV	TV	TV	TV	TV	TV
062	TV	TV	TV	TV	TV	TV
063	TV	TV	TV	TV	TV	TV
064	TV	TV	TV	TV	TV	TV
065	TV	TV	TV	TV	TV	TV
066	TV	TV	TV	TV	TV	TV
067	TV	TV	TV	TV	TV	TV
068	TV	TV	TV	TV	TV	TV
069	TV	TV	TV	TV	TV	TV

offset: 750 vBatAverage: 2.354



C

- El valor de voltaje está en formato "crudo o raw", es un valor entero entregado por el sensor directamente al que se le debe aplicar un cálculo para obtener el valor de ingeniería final (flotante) ($\text{raw} \times 0.01873128 + (-38.628956)$). La tabla 3 muestra la información del fabricante que aclara este punto.

[illegible]

Castro, raw tiene que estar en Little Endian

Primer valor de Voltaje d: 62990 x: F60E

Hexa

F	6	0	E
---	---	---	---

D

- Dado que el subtema de potencia está desarrollado sobre equipos no Intel/AMD el encoding del valor crudo es **"big-endian"**. ¿Es compatible con su hardware? ¿Cómo podría deducir (por código C) que tipo de encoding tiene el hardware con el que está trabajando ahora? Ver figura 1. En caso de no ser compatible deberá resolver el problema. **Implemente una solución que reconozca (usando simplemente C) el tipo de encoding tiene el hardware donde está corriendo y efectúe el ajuste sobre el dato solo si corresponde.** Complemente con conocimientos ad-

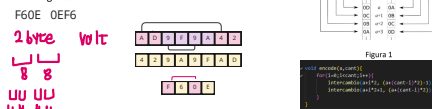


Figura 1

F

Para su control los valores obtenidos deben ser de aproximadamente 33v con bajadas entre 31.5v y 32v (Debido a los periodos de eclipse donde la tierra oculta al sol y por lo tanto los paneles no pueden cargar). Guarde los valores obtenidos en un archivo de texto o CSV, grafique con Excel o herramienta de su preferencia para verificar el comportamiento (Ver Figura 2)

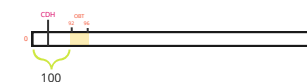


Figura 1

- La **fecha/hora de cada paquete** (OBT, On board time) es brindada por el **subsistema CDH** (Command and data handling). Siguiendo la ayuda de la documentación **obtenga también la fecha hora para cada medición.**

23 CERN TELEMETRY							
REF: 135_VTU		DESCRIPTION	HOW	TYPE	RANGE / EXPECTED VALUE	ALARM	
SD	135_VTU	Very Cold CH1	records since 30-01-1982 00:00:00	4	Status & Configuration	Complete range 12-125000VDC CH1 is a status and not a control. It is not operative. The value range of CH1 is 0VDC to 125000VDC. If CH1 is operative, in any moment, the value should be the 0VDC but, with less resolution (this time it is not exactly the 0VDC line)	Low level anomaly. CH1 is not operative. The value range is 0VDC to 125000VDC. If CH1 is operative, in any moment, the value should be the 0VDC but, with less resolution (this time it is not exactly the 0VDC line)

offset: 92 OBT: 100



51 Sinopsis

Las utilidades de la LE que tiene relación con el tiempo (fecha y hora) están definidas en el fichero de cabecera <ctime> (archivo <time.h>) que contiene las funciones que se relacionan en la tabla adjunta.

Nota: En las descripciones que siguen **timestamp** es un valor numérico que indica una fecha y hora expresada en segundos después del inicio de la época UNIX (1 de Enero de 1970 00:00:00 GMT).

Función	Uso
<code>clock()</code>	Obtener una cadena alfanumérica del tipo <code>time_t</code> . Ej: <code>time_t t; printf("%s", ctime(&t));</code> a partir de una estructura <code>time_t</code> .
<code>clock()</code>	Obtener el tiempo transcurrido desde el comienzo de la ejecución del programa.
<code>ctime()</code>	Obtener una cadena alfanumérica del tipo señalado para <code>asctime</code> a partir de un <code>time_t</code> .
<code>difftime()</code>	Obtener el tiempo en segundos entre dos <code>time_t</code> .
<code>gmtime()</code>	Obtener una fecha GMT en formato <code>tm</code> a partir de un <code>time_t</code> con la hora local.
<code>localtime()</code>	Obtener una fecha en formato <code>tm</code> a partir de un <code>time_t</code> .
<code>asctime()</code>	Obtener un <code>time_t</code> numérico a partir de una fecha <code>tm</code> .
<code>asctime_r()</code>	Obtener una representación alfanumérica a partir de una fecha <code>tm</code> (acepta un formato similar a <code>printf</code>).
<code>time_r()</code>	Obtener un <code>time_t</code> con la fecha y hora del Sistema.

Además de las funciones anteriores, la STL define tres tipos simples: `size_t`, `clock_t`, `time_t` y una estructura `tm`. En realidad se trata de typedefs dependientes de la implementación. Por ejemplo, en el fichero `<time.h>` del compilador Borland C++ encontramos: