

# Desarrollo de una plataforma de pruebas en tiempo real en tierra para CubeSats utilizando un microcontrolador STM32L476RG

Francescolli Anticona

[gustavo.anticona@utec.edu.pe](mailto:gustavo.anticona@utec.edu.pe)

Frank Bustamante

[frank.bustamante@utec.edu.pe](mailto:frank.bustamante@utec.edu.pe)

Universidad de Ingeniería y Tecnología  
Lima, Perú

**Abstract - In the following report will be presented the project for the implementation of a ground real-time test platform for CubeSats, using an STM32L476RG microcontroller. This platform focuses on temperature monitoring and energy management, with the aim of selecting the power supply based on the required power.**

**Resumen - En el siguiente informe se presentará el proyecto de implementación de una plataforma de pruebas en tiempo real en tierra para CubeSats, utilizando un microcontrolador STM32L476RG. Esta plataforma se centra en el monitoreo de temperatura y la gestión de energía, con el objetivo de seleccionar la fuente de alimentación en función de la potencia requerida.**

## I. INTRODUCCIÓN

En el último año, se calculó que alrededor de 8000 satélites se encuentran orbitando el planeta tierra. Esto refleja un aumento del 11.84% con respecto a los satélites orbitando en el año 2021. Estos son enviados al espacio por los gobiernos de países y por empresas privadas con el objetivo de realizar investigaciones sobre el espacio<sup>[1]</sup>. A causa de esta popularidad, la tecnología en este rubro sigue avanzando exponencialmente, siendo la NASA una de las principales empresas pioneras en este ámbito<sup>[2]</sup>. A medida que las tecnologías para explorar el espacio van avanzando, han surgido oportunidades económicas y comerciales. La economía del espacio comprende actividades y recursos vinculados a la investigación del espacio, el cual involucra al gobierno y a empresas privadas. Este aspecto no solo tiene un impacto en la exploración espacial sino también en la economía de los países, pues existen servicios y productos relacionados al espacio cuya evolución cala en diversos sectores y en la vida cotidiana de los ciudadanos<sup>[3]</sup>. Ante esta premisa, la universidad politécnica estatal de california, en busca de incrementar las facilidades para investigación espacial, diseñó un estándar de nanosatélite con la intención de proporcionar un acceso asequible al espacio para los investigadores de las universidades. Este fue llamado CubeSat, se caracteriza por tener una estructura de cubo cuyas caras miden 10 cm y tener una masa aproximada de 1 a 1.3 kg. Estas características minimizan el costo de materiales y lanzamiento para la operación de este en la órbita del planeta tierra<sup>[4]</sup>.

En los últimos años, diferentes universidades peruanas han desarrollado CubeSats con el fin de adquirir datos e imágenes en el espacio<sup>[5]</sup>. Hasta 2016, se registró el lanzamiento de cuatro cubesat PUCP-SAT 1, Pocket-PUCP (técnicamente no es un satélite con forma de cubo), UAPSAT-1 y Chasqui-1. Estos satélites se lanzaron en diferentes periodos de tiempo. Los dos primeros fueron creados por estudiantes de la Universidad Católica del Perú y tenían como objetivo recolectar información sobre la temperatura para la comunicación y utilidad en otros satélites. El tercero, originado en la universidad Alas Peruanas, tuvo como propósito capturar información sobre el clima en el transcurso de 8 meses. Por último, el cuarto Cubesat de nombre Chasqui fue lanzado por la Universidad Nacional de Ingeniería, este tenía la finalidad de enviar imágenes visibles e infrarrojas cercanas<sup>[6]</sup>.

En 1999 por primera vez se definió lo que es un cubesat. Este es un satélite en forma de cubo, el cual además de componerse de un cuerpo rígido, es un mecanismo compuesto por dos sistemas. El primero es el mismo satélite y el segundo es una interfaz mecánica entre el cubesat y un vehículo lanzador denominado P-POD(Poly-Picosatellite Orbital Deployer)<sup>[4]</sup>.

Un aspecto de gran importancia en un CubeSat es el manejo eficiente de la energía. Pues se busca prolongar al máximo la vida de estos satélites para que puedan cumplir su misión sin ningún inconveniente. Para ello, es crucial que los componentes del sistema mantengan un funcionamiento apropiado. Para cumplir con este requerimiento es necesario que se implemente un algoritmo para garantizar una gestión de energía confiable y eficiente<sup>[5]</sup>.

El análisis térmico en el CubeSat es fundamental para identificar si la temperatura es aceptable y no excede las limitaciones de los componentes que conforman el circuito del sistema. Pues de ser este el caso los subsistemas como la recopilación de datos o envío de imágenes pueden verse afectados. En caso el monitoreo de temperatura identifique que el dispositivo no está operando en una temperatura ideal es

necesario agregar protección adicional para mantener una temperatura adecuada<sub>[6]</sub>.

El presente trabajo propone el diseño e implementación de una plataforma de pruebas en tierra adaptada para simular un CubeSat. Para poner a prueba los diversos casos a los que el CubeSat puede estar expuesto en el espacio y comprobar que el monitoreo de temperatura y gestión de energía propuestos responden de forma correcta.

## II. METODOLOGÍA

En esta investigación se simula un CubeSat usando el microcontrolador STM32L476RG, la gestión de energía corresponde a dos fuentes de energía, una batería y un panel solar.

### Convertidor Análogo a Digital (ADC):

Un convertidor ADC se usa para convertir una señal analógica en digital para hacer más accesible su uso posterior. Esto lo hace más inmune al ruido como a otras interferencias donde la señal analógica es mucho más sensible<sub>[7]</sub>. Para el presente proyecto, se usan estos convertidores para los sensores como el fotodiodo, el potenciómetro y el sensor de temperatura, donde se transformará la información de datos en bits para pasarla por los demás procesos descritos en la metodología.

### Batería:

El estado de la batería será medida por medio de dos variables, la temperatura y la carga.

La carga mostrará el porcentaje de batería que se encuentre en ese momento, debido a que no puede descargarse. En caso que se descargue perdería la capacidad de suministrarle energía al satélite y eso desencadena que todos los sistemas que estén funcionando con la batería se queden inoperativos. Por esa razón, se debe estar en constante monitoreo con respecto al nivel de carga<sub>[3]</sub>. Para el presente proyecto, se usará un potenciómetro para simular el nivel de carga de la batería.

### Sensor de Temperatura LM35DZ:

El sensor de temperatura es el LM35DZ, este sensor te brinda de forma directa la temperatura a través de un pin analógico. Asimismo, permite medir un rango extenso de temperatura, desde -55 °C hasta 150 °C. Por este motivo, han sido utilizados en monitoreo de temperatura para garantizar el correcto funcionamiento del sistema [8]. Siendo la temperatura ambiente aproximadamente 20°C, se decidió establecer el límite de temperatura en 22°C para visualizar el cambio y la alarma de forma rápida. En el presente proyecto se verifica si la temperatura de la batería excede o no el límite máximo especificado a modo de prevención.

### Fotodiodo de polarización inversa:

Este componente absorbe los fotones de luz. Esto provoca que la resistencia interna incremente [5]. En este trabajo se utilizará el fotodiodo semiconductor de modelo OP999 como reemplazo de los paneles solares para simular la fase oscura y luminosa a la que se exponen los CubeSat en el espacio. El fotodiodo será conectado en el circuito con el objetivo de que este permita o no el paso del voltaje. En caso de que se encuentre recibiendo luz permitirá el flujo de corriente y en caso contrario lo impedirá.

### Algoritmo MPPT:

MPPT: Es un algoritmo cuyo fin es calcular y monitorear el punto máximo de potencia de un sistema de energía (EPS) [5]. Este proyecto utiliza el cálculo del MPPT de la batería y el fotodiodo para determinar cual es más óptimo y usarlo como fuente de energía de los subsistemas del sistema, los cuales poseen un requerimiento mínimo de 2W para funcionar de manera adecuada. Esta forma de uso se debe a que el cubeSat se encontrará por momentos en fase oscura o luminosa. Dependiendo de esta situación, el panel solar (simulado por el fotodiodo) debe de conmutar cuando sea posible con la batería para optimizar el rendimiento energético del sistema.

Casos	Descripción	Resultado
1	En fase luminosa, la potencia brindada por el fotodiodo es <b>mayor</b> que la potencia de la batería y la potencia requerida	Se enciende el led azul y en el UART se visualiza "Potencia eficiente, rendimiento óptimo"
2	En fase luminosa, la potencia brindada por el fotodiodo es <b>mayor</b> que la potencia de la batería pero no a la potencia requerida	Se apaga el led azul, se enciende el led rojo. En el UART se visualiza "Potencia insuficiente, peligro de mal funcionamiento"
3	En fase oscura, la potencia brindada por la batería es <b>mayor</b> que la potencia del fotodiodo y a la potencia requerida	Se enciende el led azul y en el UART se visualiza "Potencia eficiente, rendimiento óptimo"
4	En fase oscura, la potencia brindada por la batería es <b>mayor</b> que la	Se apaga el led azul, se enciende el led rojo. En el UART se visualiza "Potencia insuficiente,

	potencia del fotodiodo pero no a la potencia requerida	peligro de mal funcionamiento”
--	--------------------------------------------------------	--------------------------------

#### Transmisor y Receptor Asíncrono Universal (UART):

Este sistema contará con un monitoreo de las señales usando RTOS(Real-Time Operating System). Dado que el sistema se encontrará bajo condiciones del espacio exterior se deben tomar ciertas consideraciones.

#### Sistema Operativo a Tiempo Real (RTOS):

Con el monitoreo en el RTOS se tomará muestras del nivel de carga de la batería y la cantidad de potencia recibida por los paneles solares. Además, se deben enviar datos en tiempo real a la consola para el monitoreo. Como el proyecto se está llevando a cabo en FREERTOS, esta metodología se dividirá en 4 tareas como se muestra en la Figura 1. Se utilizó uno clásico ya que el mutex recursivo se centra en manejar múltiples adquisiciones de parte de una misma tarea y para el contexto del funcionamiento de nuestro sistema este no era necesario. El mutex clásico permite que cuando una tarea adquiere el mutex se bloquea y se vuelve prioritaria hasta que lo libere. Algo importante respecto a la jerarquía es que esta es necesaria para la correcta ejecución y sincronización del mutex. Se puede ver un diagrama temporal de esta configuración en la Figura 1. Debido a esto, se configuró las tareas con prioridades de forma descendente al inicio del programa pero estas son editadas conforme se ejecuta el programa. A continuación se explicará el desarrollo de cada una de las tareas:

- Tarea 1:

Esta tarea inicia el desarrollo de la metodología. Se mide la tensión de la batería y del sensor de temperatura a través de ADC1 y ADC2 respectivamente y los valores son almacenados en dos buffers. Una vez completen los buffers de medición se calcula el promedio y se obtienen las variables necesarias para las próximas tareas. Adicionalmente a ello, se diseñarán los cálculos matemáticos para el % de carga (batería) y la temperatura (sensor LM35DZ).

- Tarea 2:

Se medirá la tensión del fotodiodo con un convertidor llamado ADC3 tomando en cuenta la estructura de las otras mediciones. También se levantará un flag que nos avisará si nos encontramos en fase luminosa o fase oscura.

- Tarea 3:

Para esta tarea se diseñará el proceso del algoritmo MPPT. En caso que el fotodiodo esté suministrando energía al programa, la comparación se hace entre la potencia del sistema que debería ser suministrado con la potencia del fotodiodo. Si la potencia del fotodiodo supera a la del sistema, entonces el fotodiodo estará alimentando la energía al sistema; sin embargo, si la potencia del sistema es mayor, entonces la batería estaría siendo usada como fuente de energía al sistema. Además, se usan flags para el fotodiodo como la batería que me muestren ‘1’ si se están usando de forma correcta. En caso que no se estén usando, los flags se muestran como ‘0’.

- Tarea 4:

El monitoreo del sistema en marcha se corroborará a través de una comunicación UART a la PC. Se mostrarán 3 indicadores. El primero es el uso del fotodiodo o el potenciómetro. El segundo monitoreará el nivel de carga. El tercero imprimirá el valor de temperatura.

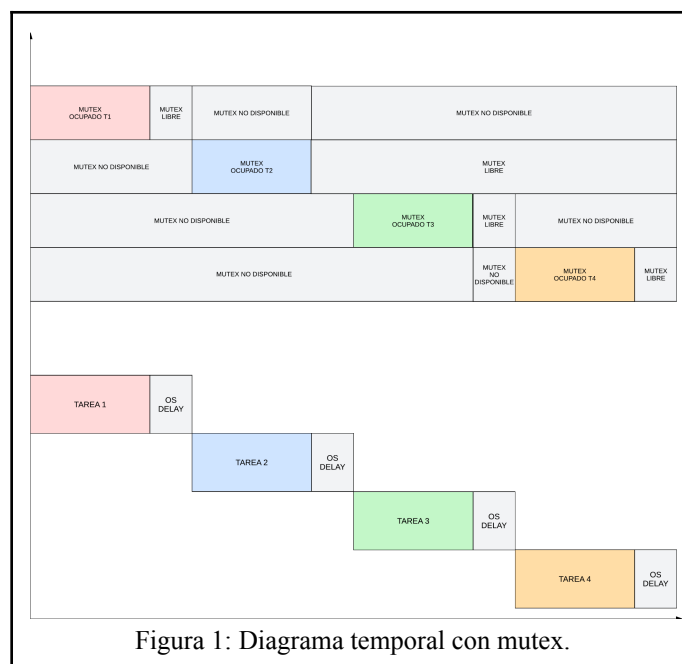
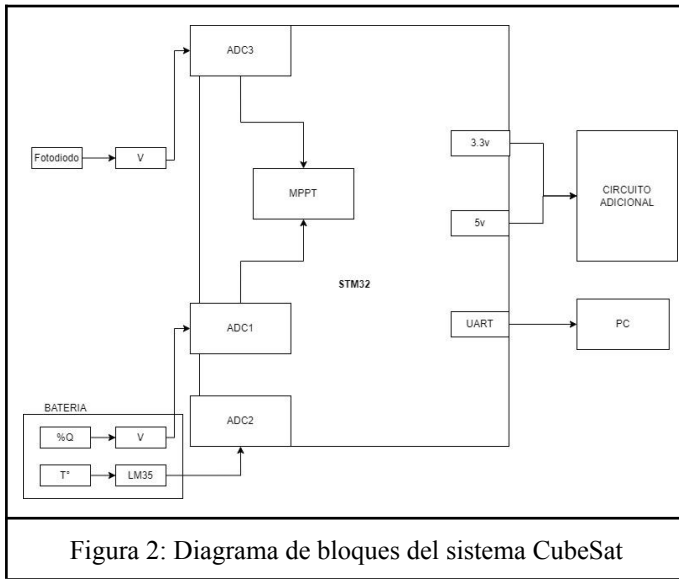


Figura 1: Diagrama temporal con mutex.



### III. RESULTADOS

#### Convertidor Análogo a Digital (ADC):

De acuerdo a la metodología, el ADC funciona correctamente pues convierte los datos del voltaje de las fuentes en valores digitales. Esto permite que el algoritmo trabaje de forma más ordenada con valores discretos. Adicionalmente, se estableció que los tipos de datos serían enteros y no flotantes por el ahorro de consumo de recursos computacionales. Como resultado se puede observar que los buffers de cada ADC recolecta de forma correcta los valores discretizados de voltaje del fotodiodo, voltaje de la batería y del voltaje del sensor de temperatura.

Expression	Type	Value
✓ BADCFOTO	uint32_t [256]	[256]
✓ [0..99]		[256]
0= BADCFOTO[0]	uint32_t	3739
0= BADCFOTO[1]	uint32_t	3734
0= BADCFOTO[2]	uint32_t	3739
0= BADCFOTO[3]	uint32_t	3731
0= BADCFOTO[4]	uint32_t	3729
0= BADCFOTO[5]	uint32_t	3740
0= BADCFOTO[6]	uint32_t	3738
0= BADCFOTO[7]	uint32_t	3733
0= BADCFOTO[8]	uint32_t	3739
0= BADCFOTO[9]	uint32_t	3734

Figura 3: Buffer ADC del Fotodiodo

Expression	Type	Value
✓ BADCBAT	uint32_t [256]	[256]
✓ [0..99]		[256]
0= BADCBAT[0]	uint32_t	1691
0= BADCBAT[1]	uint32_t	1690
0= BADCBAT[2]	uint32_t	1690
0= BADCBAT[3]	uint32_t	1691
0= BADCBAT[4]	uint32_t	1689
0= BADCBAT[5]	uint32_t	1690
0= BADCBAT[6]	uint32_t	1689
0= BADCBAT[7]	uint32_t	1690
0= BADCBAT[8]	uint32_t	1692
0= BADCBAT[9]	uint32_t	1692

Figura 4: Buffer ADC de la Batería

Expression	Type	Value
✓ BADCTER	uint32_t [256]	[256]
✓ [0..99]		[256]
0= BADCTER[0]	uint32_t	282
0= BADCTER[1]	uint32_t	282
0= BADCTER[2]	uint32_t	281
0= BADCTER[3]	uint32_t	278
0= BADCTER[4]	uint32_t	282
0= BADCTER[5]	uint32_t	279
0= BADCTER[6]	uint32_t	285
0= BADCTER[7]	uint32_t	278
0= BADCTER[8]	uint32_t	278
0= BADCTER[9]	uint32_t	280

Figura 5: Buffer ADC del sensor

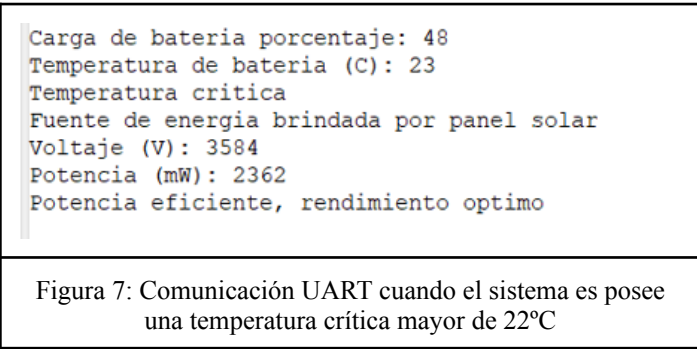
#### Sensor de Temperatura LM35DZ:

En las pruebas de funcionamiento de este sensor, se evaluaron dos escenarios. El primero es mantener una temperatura estable, es decir a un valor menor a 22°C. En este caso al no sobrepasar el límite de temperatura establecido el led amarillo se mantendrá apagado. El resultado se puede apreciar en la figura 6.

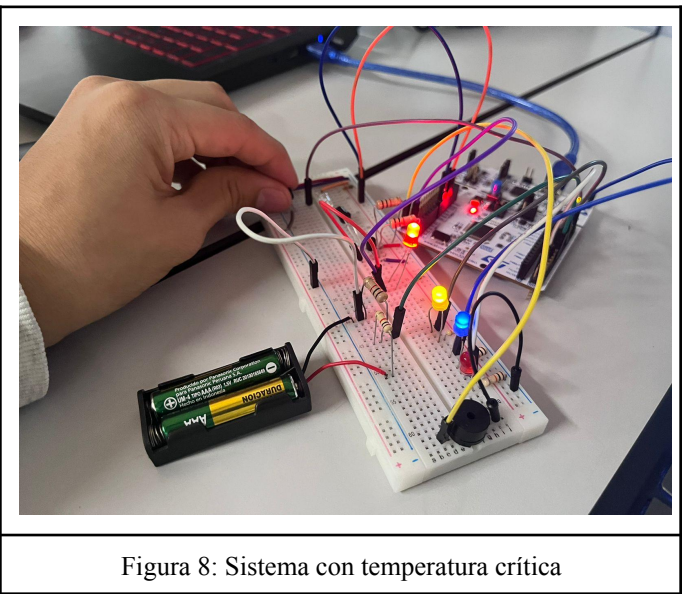
Carga de bateria porcentaje: 47
Temperatura de bateria (C): 22
Temperatura estable
Fuente de energia brindada por panel solar
Voltaje (V): 3602
Potencia (mW): 2374
Potencia eficiente, rendimiento optimo

Figura 6: Comunicación UART cuando el sistema es posee una temperatura estable de 22°C

El segundo es que la temperatura se encuentre crítica a un valor mayor a 22°C y su resultado se puede apreciar en la figura 7.

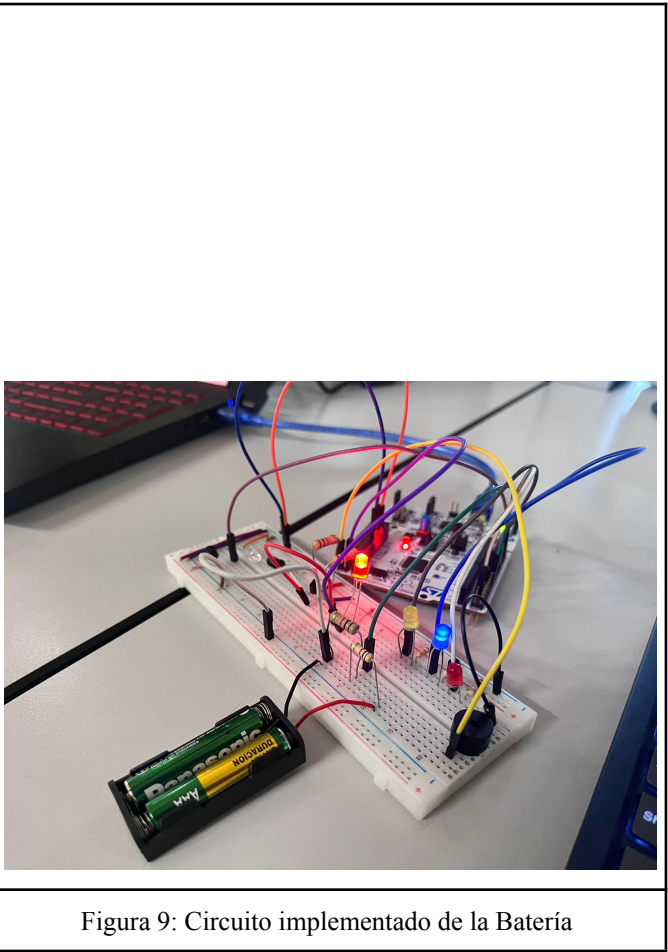


Cuando la temperatura de la batería supera el valor máximo establecido se enciende una alerta por medio de una salida GPIO hacia un led. Alertando así, de la temperatura crítica.

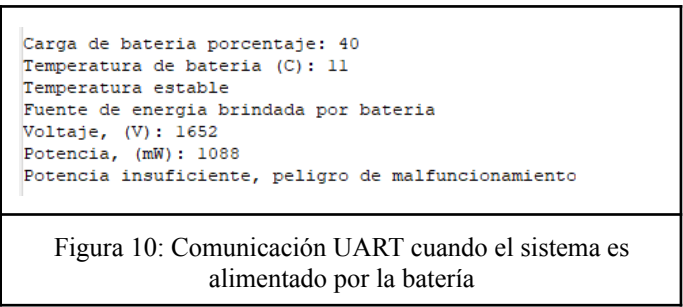


Algoritmo MPPT:  
1. Batería:

La batería tiene el propósito de actuar como fuente de energía en caso la potencia de la batería sea mayor a la brindada por el fotodiodo o el sistema se encuentra en fase oscura como se puede observar el circuito en la figura 9.



En este caso la batería es representada con un socket compuesto de dos pilas AAA de 1.5V de la marca Panasonic. Estas baterías se encuentran descargadas. Por ello, en el caso de utilizarse cómo fuente de energía, la potencia que brinda no supera el valor mínimo declarado. En la figura 10 se puede observar un resumen de la batería como suministro de energía principal.





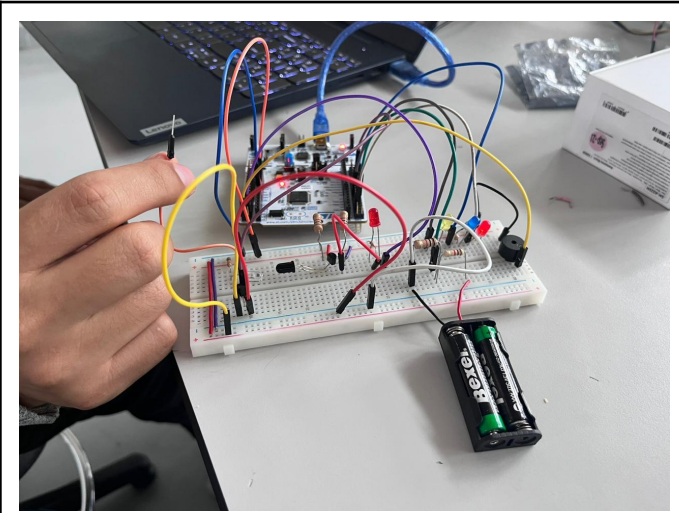


Figura 11: Sistema cuando es alimentado por la batería

2. Fotodiodo de polarización inversa:

Para el panel fotovoltaico se simuló el experimento con un fotodiodo y un led infrarrojo. Cuando el led infrarrojo se encuentra conectado a voltaje, este simula la fase luminosa. Se puede apreciar este caso en la figura 12.

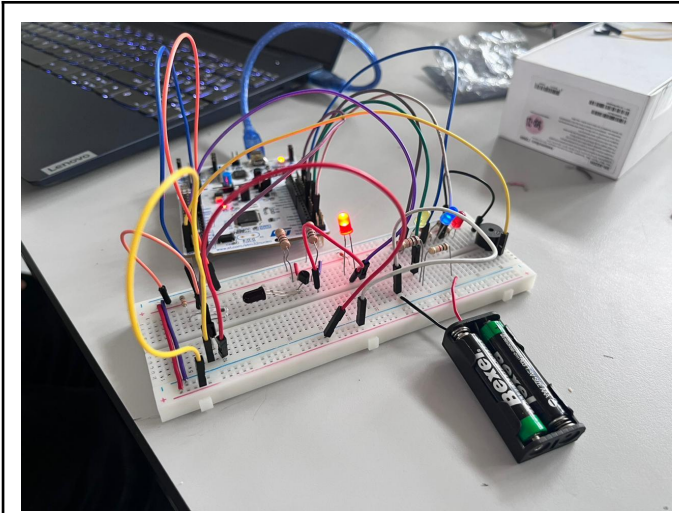


Figura 12: Implementación del fotodiodo en fase luminosa

```
Carga de bateria porcentaje: 48
Temperatura de bateria (C): 23
Temperatura critica
Fuente de energia brindada por panel solar
Voltaje (V): 3584
Potencia (mW): 2362
Potencia eficiente, rendimiento optimo
```

Figura 13: Comunicación UART cuando el fotodiodo se

encuentra en fase luminosa

Al desconectar la alimentación del led infrarrojo, el envío de señal se detiene y el fotodiodo se encuentra en fase oscura, eso quiere decir que la energía es suministrada por la batería como se puede apreciar en la figura 14.

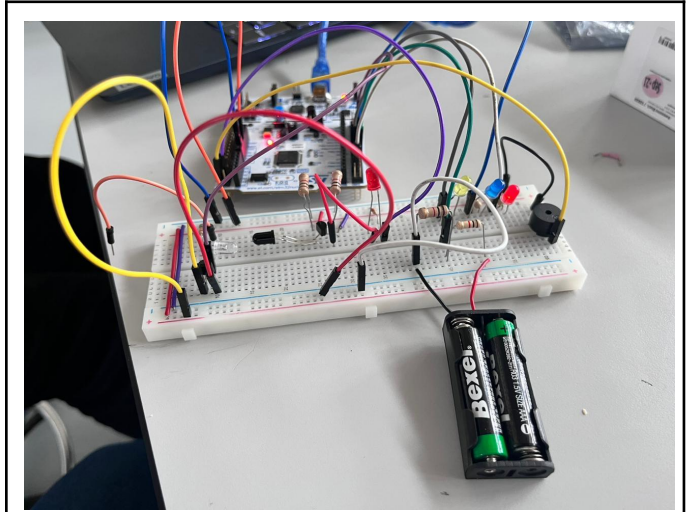


Figura 14: Implementación del fotodiodo en fase oscura

```
Carga de bateria porcentaje: 40
Temperatura de bateria (C): 8
Temperatura estable
Fuente de energia brindada por bateria
Voltaje, (V): 1658
Potencia, (mW): 1092
Potencia insuficiente, peligro de malfuncionamiento
```

Figura 15: Comunicación UART cuando el fotodiodo se encuentra en fase oscura

Transmisor y Receptor Asíncrono Universal (UART):

De acuerdo con la metodología, el UART se encuentra funcionando de la manera correcta para mostrar en la PC los resultados obtenidos por el programa. Se puede observar que después de pasar por las tareas anteriores, toda esta información se puede ver reflejada en la PC así como se observa en la figura 16.

```
Carga de bateria porcentaje: 47
Temperatura de bateria (C): 22
Temperatura estable
Fuente de energia brindada por panel solar
Voltaje (V): 3602
Potencia (mW): 2374
Potencia eficiente, rendimiento optimo
```

Figura 16: Mensaje enviado por UART a la Consola

Sistema Operativo a Tiempo Real (RTOS):

Para el caso del FREERTOS, se gestionó en 4 tareas el proyecto tal como se describe en la metodología. Asimismo, para el correcto funcionamiento de las tareas se implementó un Mutex. Debido a que el Mutex considera la jerarquía en base a la prioridad que se configura a cada tarea. El programa edita la prioridad de las tareas conforme se van ejecutando para asegurarse de que las 4 tareas mencionadas se desarrollen de forma correcta. Las prioridades iniciales se muestran en la figura 17

Task Name	Priority	Stack	Entry Function	Code G.	Parameter	Allocation	Buffer Name	Control Block
TareaMediB...	osPriorityHigh	128	StartTareaBAT	Default	NULL	Dynamic	NULL	NULL
TareaMediF...	osPriorityAboveNormal	128	StartTareaFOTO	Default	NULL	Dynamic	NULL	NULL
TareaMPPT	osPriorityNormal	128	StartTareaMPPT	Default	NULL	Dynamic	NULL	NULL
TareaUART	osPriorityBelowNormal	128	StartTareaUART	Default	NULL	Dynamic	NULL	NULL

Figura 17: Configuración de tareas en FreeRTOS

## IV. OBSERVACIONES Y RECOMENDACIONES

- Se observó que al desconectar el led infrarrojo para cortar la fase luminosa el sensor de temperatura LM35 sufre una interferencia que altera el valor de la lectura ADC3. Por ende se recomienda utilizar un bloqueo de la señal infrarroja con un material como el cartón o un plástico oscuro.
- Para el caso de FREERTOS, se observó que los valores del buffer en preventivo solo midieron los datos hasta la posición 32. Así que, se cambió la configuración a cooperativo y es en esta configuración donde se logró hacer la medición completa.
- Según las mediciones de la memoria RAM, cada tarea configurada en el FREERTOS equivale a un 0,1%.
- Se puede descargar en github el proyecto para cualquier persona que esté interesada en agregar nuevas ideas al proyecto. Este es el enlace: <https://github.com/Frangus/CubeSat-TSC.git>

## V. CONCLUSIONES

- Se desarrolló con éxito una plataforma de pruebas en tiempo real en tierra para CubeSats utilizando un microcontrolador STM32L476RG. Esta plataforma permite simular y evaluar la gestión de energía y el

monitoreo de temperatura de los CubeSats antes de su lanzamiento al espacio.

- Se logró calcular las potencias de las fuentes de energía a través de la lectura de tensiones de la batería y el circuito del fotodiodo por medio de 2 ADCs. De esta forma se garantizó el funcionamiento del algoritmo MPPT
- Se estableció la comunicación UART los valores de la temperatura y el porcentaje de carga de la batería para el monitoreo de esta.

## REFERENCIAS

- [1] ts, “Search OSOidx,” *Unoosa.org*, 2023. [https://www.unoosa.org/oosa/osoindex/search-ng.jsp?lf\\_id=#?c=%7B%22filters%22:%5B%5D,%22sortings%22:%5B%7B%22fieldName%22:%22object.launch.dateOfLaunch\\_s1%22,%22dir%22:%22desc%22%7D%5D,%22match%22:null%7D](https://www.unoosa.org/oosa/osoindex/search-ng.jsp?lf_id=#?c=%7B%22filters%22:%5B%5D,%22sortings%22:%5B%7B%22fieldName%22:%22object.launch.dateOfLaunch_s1%22,%22dir%22:%22desc%22%7D%5D,%22match%22:null%7D) (Accedido Jun. 23, 2023).
- [2] C. de, “6 Tecnologías que la NASA está desarrollando para enviar humanos a Marte. – Madrid Deep Space Communications Complex,” *Nasa.gov*, Jul. 22, 2020. <https://www.mdsc.nasa.gov/index.php/2020/07/23/6-tecnologias-que-la-nasa-esta-desarrollando-para-enviar-humanos-a-marte/> (accedido Jun. 23, 2023).
- [3] *OECD Handbook on Measuring the Space Economy*. 2012. doi: <https://doi.org/10.1787/9789264169166-en>.
- [4] A. Space, “Guía básica de nanosatélites | Alén Space,” *Alén Space*, Apr. 04, 2023. <https://alen.space/es/guia-basica-nanosatelites/> (Accedido Jul. 13, 2023).
- [5] “Simulation of the proposed combined Fuzzy Logic Control for Maximum Power Point Tracking and Battery Charge Regulation used in CubeSat,” Jul. 2023, doi: <https://doi.org/10.24425/ae.2020.133916>.
- [6] “UNIVERSIDAD DEL VALLE DE GUATEMALA Facultad de Ingeniería.” Accedido: Jul. 13, 2023. [Online]. Available: <https://repositorio.uvg.edu.gt/bitstream/handle/123456789/4239/An%c3%a1lisis%20t%c3%a9rmico%20para%20un%20CubeSat%20U%20Estrada%20FINAL.pdf?sequence=1&isAllowed=y>

- [7] K. D. Vega, A. Lopez, J. E. Pedroza, L. G. De Santos and R. R. Saucedo, "ADC y DAC", Investigación, Universidad Politécnica de Aguascalientes, México.
- [8] D. A. Lamb, Stuart, M. Baker, C. Underwood, and Simran Mardhani, "Thin film cadmium telluride solar cells on ultra-thin glass in low earth orbit—3 years of performance data on the AlSat-1N CubeSat mission," vol. 29, no. 9, pp. 1000–1007, May 2021, doi: <https://doi.org/10.1002/pip.3423>.