

## ARQUITETURA DE SOFTWARE EMBARCADO DE UM CUBESAT DE COLETA DE DADOS DE RADIAÇÃO IONIZANTE ATMOSFÉRICA

JUAN MARCO DE JESUS LIBONATTI<sup>1</sup>, FELIPE DAL CIN RODRIGUES COSTA<sup>2</sup>, LEONARDO XAVIER DANTAS<sup>3</sup>, NARAYANE RIBEIRO MEDEIROS<sup>4</sup> e RUAN FERNANDES CAMPOS<sup>5</sup>

<sup>1</sup>Aluno de Graduação em Engenharia, ITA, São José dos Campos-SP, libonatti@hotmail.com;

<sup>2</sup>Aluno de Graduação em Engenharia, ITA, São José dos Campos-SP, lxddantas@gmail.com;

<sup>3</sup>Aluno do Graduação em Engenharia, ITA, São José dos Campos-SP, felipedalcin2006@gmail.com

<sup>4</sup>Aluna do Graduação em Engenharia, ITA, São José dos Campos-SP, narayane.rm@gmail.com;

<sup>5</sup>Aluno de Graduação em Engenharia, ITA, São José dos Campos-SP, ruan.campos.9020@ga.ita.br.

Apresentado no  
Congresso Técnico Científico da Engenharia e da Agronomia – CONTECC  
15 a 17 de setembro de 2024

**RESUMO:** O detalhamento da arquitetura dos códigos embarcados de um cubesat para lançamento via balão atmosférico e suas medições de radiação ionizante foram o alvo desse trabalho. O satélite, denominado CurieSat V2, tinha como missão a medição de radiação atmosférica, atingindo uma altitude de 31,93 quilômetros em seu lançamento. O software foi desenvolvido para um microcontrolador ESP32, utilizando como base a framework ESP-IDF, e implementava um gerenciamento de tarefas baseado em RTOS, dividindo entre os dois núcleos de processamento do dispositivo as tarefas de coleta de dados de sensores e de envio e registro de dados. Também foi discutida a implementação da padronização de pacotes utilizada para a transmissão via rádio LoRa, bem como o rastreamento feito pela estação de solo.

**PALAVRAS-CHAVE:** balão estratosférico, RTOS, sonda atmosférica, radiação ionizante.

### EMBEDDED SOFTWARE AND MEASUREMENTS OF A CUBESAT COLLECTING IONIZING RADIATION DATA IN THE ATMOSPHERE

**ABSTRACT:** This work details the architecture of the embedded codes of a cubesat launched via atmospheric balloon and its measurements of ionizing radiation. The satellite, named CurieSat V2, had the mission of measuring atmospheric radiation and reached an altitude of up to 31.93 kilometers in its launch. The software was developed for an ESP32 microcontroller, using the ESP-IDF framework as a base, and implemented a task management system based on RTOS, dividing between the two processing cores of the device the sensor data collection tasks and the data transmission and logging tasks. The implementation of the packet used for LoRa radio transmission, as well as the ground station tracking, was also discussed.

**KEYWORDS:** weather balloon, RTOS, atmospheric probe, ionizing radiation.

### INTRODUÇÃO

Sistemas embarcados são sistemas de computação focados em uma tarefa específica, caracterizados também pela operação com pouca ou nenhuma interface humana. Sistemas desse tipo são muito comuns em diversas áreas de atuação, mas encontram uma importância ainda maior na indústria espacial: Satélites e veículos lançadores são sistemas extremamente especializados, com acesso e recursos limitados após o lançamento, sendo necessário o desenvolvimento de softwares específicos que otimizem a alocação dos recursos disponíveis para a execução das tarefas necessárias, de acordo com a prioridade de execução.

Além disso, é comum que, nesse tipo de sistema, o tempo também seja um recurso a ser administrado. Ou seja, é necessário controlar o tempo de resposta do sistema, como, por exemplo, no controle de sistemas mecânicos ou na recepção de sinal de rádio, onde não são admitidos atrasos na realização dessas tarefas. Dessa forma, é comum o uso de sistemas operacionais de tempo real (RTOS),

com foco em determinismo, isto é, previsibilidade da cronologia de execução, permitindo o gerenciamento das tarefas de acordo com prioridades, garantindo os tempos de resposta do sistema.

Nano satélites, plataformas de dimensões em 1U a 12U e massa inferior a 10kg, onde 1U é a medida de um cubo de 100x100x100 mm, são ainda mais restritivos pois devido as suas dimensões reduzidas são incapazes tanto de acomodar como prover energia com seus painéis solares para circuitos com grande poder de processamento. Considerando esse tipo de plataforma, uma das missões relevantes que pode ser executada por meio do lançamento em balões atmosféricos é a medição da curva de radiação ionizante, composta por radiações que vão desde UV até raios gama, em relação à altitude. Essa medição impacta diversos aspectos da vida na superfície terrestre e serve como um parâmetro importante para a previsão de desastres naturais e para determinar a dose de radiação recebida por tripulações de aeronaves, que não pode ultrapassar certos níveis de exposição (IAEA, 1996). Essa medição pode ser obtida por meio do uso de sensores comercialmente disponíveis a preços acessíveis (Malmonge et al., 2019).

Nesse contexto, a equipe de desenvolvimento de nano satélites do ITA, a ITACube, desenvolveu os softwares tanto para a estação de solo quanto para o lançamento da plataforma CurieSat V2, equipada com um contador Geiger, cujo objetivo era a medição de radiação ionizante com lançamento via balão. O sistema também participou na segunda edição da olimpíada brasileira de satélites (OBSat). Neste trabalho, são discutidos a arquitetura de software e o link de comunicação, bem como o desempenho desses sistemas em voo e os resultados coletados pelo *payload*.

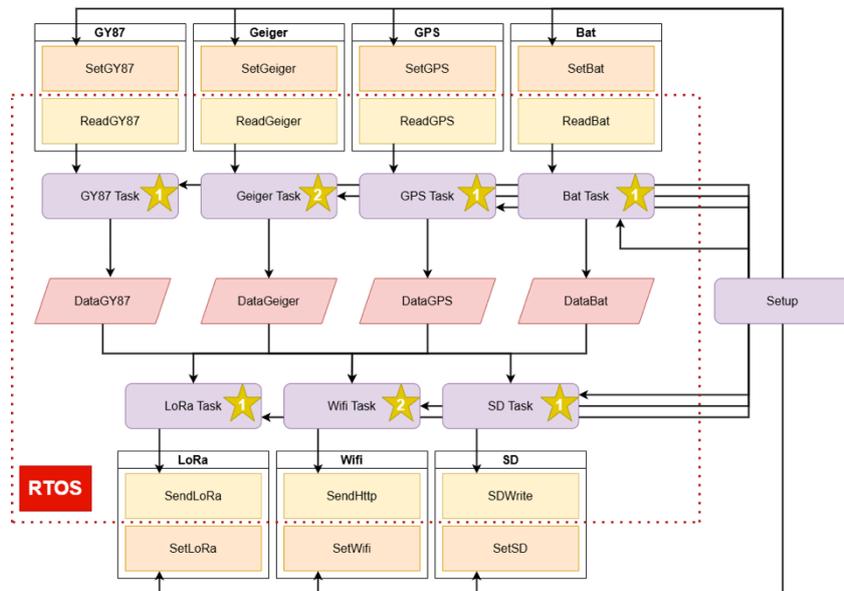
## MATERIAL E MÉTODOS

O cubesat possuía uma arquitetura relativamente simples de 1U que englobava um rádio LoRa, de 100mW e frequência de 900MHz, e um microcontrolador ESP32, de dois núcleos e operando a 240 MHz. Além disso, havia sensores inerciais e um barômetro conectados ao microcontrolador via interface I2C. O sistema contava também com um GPS cuja comunicação era feita por meio de UART/RS232 e um cartão microSD, para registro de dados, utilizando interface SPI. Por fim, havia também um sensor de bateria e o *payload*, um contador Geiger para se realizar medições de radiação ionizante, captando especialmente radiação gama e beta.

Dessa forma, o satélite possuía seis entradas de dados de diferentes prioridades e diferentes frequências de amostragem. O contador Geiger, por exemplo, tinha alta prioridade e deveria ser lido precisamente a cada 10s para cumprir a missão, já o acelerômetro dos sensores inerciais tinha frequência de aquisição de cerca de 1KHz e dados que, apesar de relevantes, não impactavam o cumprimento da missão. As informações coletadas deixavam o sistema por meio de até três vias, sendo duas delas a transmissão via rádio e o armazenamento em SD. O último meio de transmissão ocorria apenas no veículo lançador da OBSat e era uma transmissão via WiFi/HTTP para a sonda da organização da competição embarcada no veículo, que posteriormente retransmitia ao solo. Como essa forma de envio de dados era exigida pela competição em intervalos bem definidos, ela também foi considerada de alta prioridade durante o desenvolvimento do sistema.

Para cumprir executar essas tarefas, o código foi criado utilizando a framework ESP-IDF (ESPRESSIF,2020) e executando o FreeRTOS (BARRY, 2010), um sistema operacional RTOS de código aberto incluído na ESP-IDF. O sistema funcionava por meio da execução de oito tarefas principais, sendo quatro de coleta de dados de sensores, três de consumo de dados e uma de setup. Conforme pode ser conferido na Figura 1, a tarefa de setup executa as funções de inicialização dos componentes. Essas funções inicializam cada componente e retornam quaisquer erros encontrados. A tarefa de setup então sinaliza os erros por meio de buzzer e LED, procedendo para a inicialização das outras sete tarefas e, após isso, a tarefa de setup é deletada.

Figura 1: Esquemático da arquitetura do software embarcado, com as tarefas representadas em roxo, as funções de inicialização em laranja, as de execução em amarelo e os campos de memória de uso compartilhado em vermelho. As prioridades principais das tarefas são indicadas pelos números na estrela.

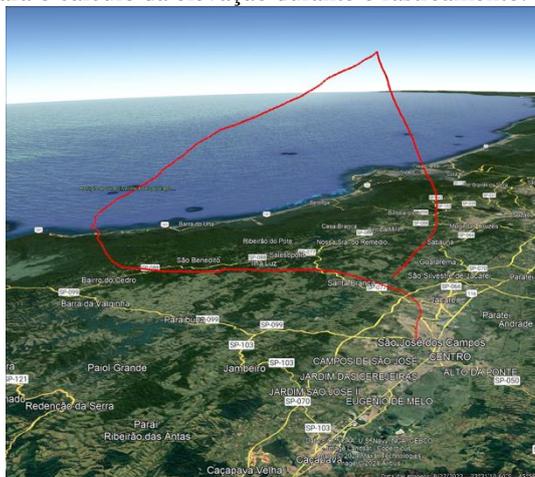


As quatro tarefas de coleta de dados foram separadas de acordo com a interface a ser utilizada, a fim de evitar o compartilhamento de recursos de hardware entre tarefas. Dessa forma, os sensores inerciais e o barômetro, que dividiam a mesma interface I2C, são lidos por uma única tarefa, a GY-87, nomeada pela placa que contém esses três sensores. Todas as tarefas de coleta são executadas no núcleo 1. Para a tarefa do Geiger foi atribuída uma maior prioridade de execução, prioridade 2, de forma a ser sempre executada exatamente em seu tempo programado, salvo o erro ocasional do próprio cristal oscilador do microcontrolador. Como o GPS e a bateria têm baixa frequência de leitura, isto é, estão indisponíveis para coleta de dados durante maior parte do tempo, e dado que o FreeRTOS funciona de forma a priorizar a execução alternada para que todas as tarefas disponíveis de mesma prioridade sejam executadas menos uma vez, optou-se por deixar as outras tarefas com a prioridade 1. Dessa forma, os sensores do GY-87 são lidos na maior frequência possível dentro da capacidade do sistema, mas sem comprometer a eventual leitura do GPS e da bateria.

De forma similar, as três tarefas de transmissão/armazenamento eram executadas no núcleo 2 do microcontrolador. Para realizar a troca de informações entre essas e as tarefas de coleta eram usados endereços de memória compartilhados protegidos por semáforos. Novamente, foi designada prioridade 2 para a tarefa prioritária de transmissão wifi. Como a transmissão LoRa também possui baixa frequência de transmissão (Devalal & Karthikeyan, 2018), foi mantida sua prioridade no mesmo nível daquela de escrita no cartão microSD, operando em dinâmica similar àquela do parágrafo anterior. Os arquivos foram salvos no cartão SD em formato CSV para fácil leitura e tratamento após recuperação.

Outro fator importante foi a escolha do formato de mensagem a ser transmitida pelo rádio LoRa. Tendo em vista a baixa taxa de transmissão dos dados, optou-se por um pacote de 54 bytes contendo principalmente um identificador inteiro, informações das coordenadas e tempo do GPS, em formato *float* e, por fim, um *checksum* de 8 bits. O sistema contava também com outras informações dos sensores do satélite, como barômetro e bateria, e também do *payload*. O pacote era transmitido à estação de solo que utilizava o modelo de formulação geométrica considerando o modelo de Terra geoide da biblioteca python *geographiclib* (Karney, 2022) para calcular o azimute, isto é, o apontamento da antena, e, por meio da geometria da Figura 2, calcula-se também a elevação  $\eta$ , ou seja, a inclinação necessária da antena em relação ao plano do solo, a partir do ângulo  $\theta$  entre os dois pontos e de suas altitudes. Assim é possível realizar com precisão o rastreamento da plataforma no ar, permitindo o uso adequado de uma antena direcional de 17dBi.

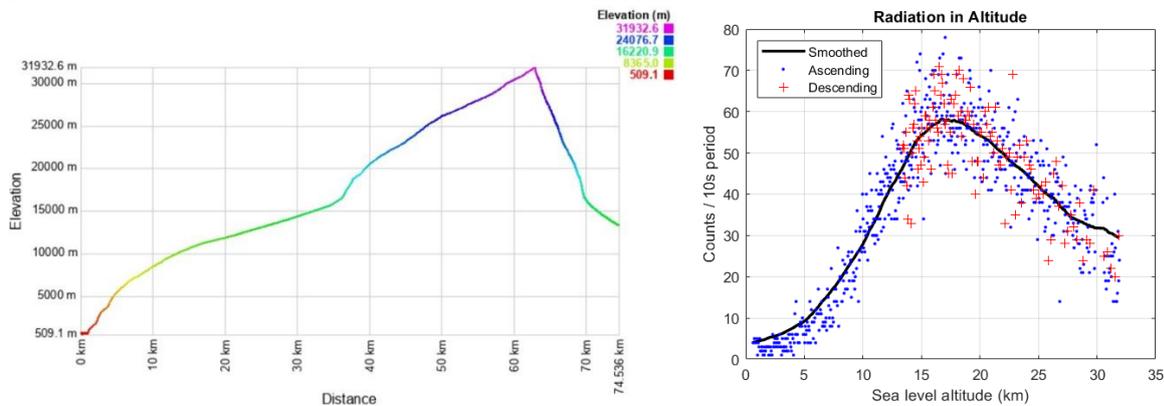
Figura 2: À esquerda, trajetória feita pelo balão em lançamento no dia 29/04/2024. À direita, geometria utilizada para o cálculo da elevação durante o rastreamento.



## RESULTADOS E DISCUSSÃO

O satélite foi lançado em 29 de abril de 2024 a partir da torre da Divisão de Ciências Atmosféricas (ACA), localizada em São José dos Campos, SP, seguindo a trajetória conforme visto na Figura 2. Como foi um lançamento independente, a telemetria foi feita somente com a utilização do rádio LoRa. A plataforma atingiu a altitude máxima de cerca de 31,9 km.

Figura 3: À esquerda, gráfico da distância em solo percorrida pelo balão versus altitude. À direita, gráfico das medições do contador Geiger registradas pelo sistema.



O sistema em RTOS funcionou em plena capacidade até os últimos 14 km de descida, nos quais houve um desligamento da bateria por conta da baixa temperatura. O sistema foi capaz de salvar em SD os dados dos sensores, bem como transmiti-los via rádio LoRa. A coleta de dados do Geiger seguiu o intervalo de 10s, havendo um atraso máximo registrado entre as medições recuperadas de 1 ms em relação à essa referência.

Foi possível receber pacotes do satélite mesmo a uma distância de mais de 74 km, tendo uma taxa de falha de apenas 3 em 237 pacotes recebidos, que eram enviados a cada 30 segundos. Os pacotes corrompidos foram detectados tanto pela formulação dos dados como também pela falha no *checksum*. Por fim, a alta taxa de recebimento de pacotes, de 237 contra 241 pacotes esperados, indica o correto funcionamento da formulação realizada para o rastreamento da plataforma. A figura 3 mostra parte dos resultados obtidos, incluindo as medições do contador Geiger em altitude.

Observa-se na curva de radiação o comportamento similar ao registrado em outros países, ocorrendo a subida usual da radiação atmosférica até o máximo de Pfozter, em torno de 17 km, seguida

da queda dos níveis de radiação (TOBIN, 2017). Foram omitidos do gráfico picos de detecção de radiação na altitude de 30km na atmosfera, encontrados tanto na subida como na descida.

## CONCLUSÃO

O código desenvolvido para a plataforma funcionou de acordo com o previsto, permitindo a execução multitarefa de leitura de sensores, respeitando uma prioridade de execução clara e, dessa forma, controlando o tempo de resposta do sistema. O sistema de envio de dados e rastreamento durante solo também funcionou conforme esperado, permitindo o recebimento de telemetria em todo o voo.

Além disso, o experimento obteve os resultados a que se propôs, levantando a curva do nível de radiação ionizante em função da altitude em São José dos Campos, em território brasileiro.

## AGRADECIMENTOS

Ao Prof. Dr. Christopher Shneider Cerqueira, por sua atuação como professor orientador do grupo de extensão ITACube; Ao CNPq pela concessão de recursos fundamentais ao projeto; Ao Prof. Dr. Tertuliano Ribeiro Pinto e à Divisão de Ciências Atmosféricas (ACA) do Instituto de Aeronáutica e Espaço (IAE) por seu apoio no lançamento da plataforma; Ao Centro Espacial ITA (CEI) pelo seu apoio irrestrito a este e a todos os outros projetos praticados pela ITACube.

## REFERÊNCIAS

- Barry, R. Using the FreeRTOS Real Time Kernel: A Practical Guide. 2. ed. Real Time Engineers Ltd., 2010.
- Devalal, S.; Karthikeyan, A. LoRa Technology - An Overview. 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 284-290, doi: 10.1109/ICECA.2018.8474715.
- Espressif Systems. ESP-IDF Programming Guide. Disponível em: <https://docs.espressif.com>. Acesso em: 10 jul. 2024.
- International Atomic Energy Agency (IAEA). International Basic Safety Standards for Protection against Ionizing Radiation and for the Safety of Radiation Sources. Vienna: IAEA, 1996. 448 p. (Safety Series, n. 115).
- Malmonge, I. M.; Gomes, M. P.; Carvalho, R. R. F.; Gomes, R. Study of a Portable Experimental Set for the Monitoring of Ionizing Radiation in the Tropical Region of Brazil. In: International Nuclear Atlantic Conference (INAC), 2019, Santos, SP. Proceedings of the INAC 2019. Santos: Associação Brasileira de Energia Nuclear (ABEN), 2019. p. 1-10.
- Tobin, R. J.; Burr, A. W.; Gilman, S. W.; Hainsworth, W. J.; Howell, A. S.; Martin, K. M. Measuring Radiation in the Stratosphere: A High Altitude Balloon Journey. In: AIAA Balloon Systems Conference, 5-9 Jun 2017. doi: 10.2514/6.2017-3788.