

DESENVOLVIMENTO DE UM DATA LOGGER PARA BATERIAS COM CÉLULAS DE LÍTIO

SPECK, Thiago Beninca
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
thiago.b14@aluno.ifsc.edu.br

DA SILVA, Eder Andrade
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
eder.silva@ifsc.edu.br

SANTOS, Fernando Rodrigues
Engenharia Mecatrônica
Instituto Federal de Santa Catarina
Criciúma, Santa Catarina, Brasil
fernando.rodrigues@ifsc.edu.br

Resumo – Este trabalho apresenta a elaboração de um *Data Logger* de baixo custo a fim de captar dados de temperatura, tensão e corrente provenientes de células de baterias de lítio. O conceito de Internet das Coisas foi utilizado para desenvolver uma solução embarcada que utiliza um circuito integrado LTC6811 capaz de coletar as informações de tensão de doze células distintas e realizar a leitura das grandezas de temperatura e corrente coletadas por sensores externos. A coleta dos dados de corrente é realizada através de um sensor de efeito hall HCS-LTR5 e a coleta dos dados de temperatura é realizada por meio de três sensores NTC B3435 10k Ohms. O processamento dos dados ocorre em um microcontrolador ESP32 e a transmissão destes dados é feita através do protocolo de comunicação Wi-Fi até um banco de dados localizado em nuvem. Além de armazenar os dados via WEB, o dispositivo também conta com um módulo de cartão SD que se comunica com o módulo ESP32 através de um protocolo SPI e realiza o *back-up* das informações coletadas. Ao término do desenvolvimento do projeto, pode-se testar e validar o *Data Logger*, que apresentou resultados satisfatórios na coleta, processamento e transmissão de dados. Pode-se observar o projeto, desenvolvimento e resultados da pesquisa ao longo deste trabalho, juntamente com sugestões para implementação de projetos futuros.

Palavras-Chaves — Bateria de Lítio, Sistema embarcado, ESP32, Firebase e LTC6811.

I. INTRODUÇÃO

As baterias que utilizam como base o elemento lítio tiveram sua primeira aplicação industrial no ano de 1991, pela empresa Sony. Sua rápida ascensão no mercado se deu pelas características de alta densidade de energia e potência, elevada segurança, baixa taxa de autodescarga e baixo custo [1].

A bateria de lítio é composta por diversas células, que podem possuir formato cilíndrico, prismático ou de filmes laminados [2]. Cada célula possui uma baixa tensão individual, todavia, as células podem ser conectadas entre si, com o objetivo de criar uma bateria de alta tensão [3].

Baterias que esgotam sua vida útil são, muitas vezes, descartadas, acarretando o vazamento de substâncias tóxicas e contaminando o solo e a água do ambiente. Além disso, baterias de lítio podem gerar incêndios residuais ao longo dos anos, poluindo o ar atmosférico e contribuindo para o aquecimento global [4].

Para evitar que o descarte indevido das baterias ocorra, é possível reutilizá-las em uma outra aplicação através do processo de reciclagem dos seus componentes. Todavia, o reaproveitamento dos materiais presentes na bateria é economicamente inviável uma vez que o custo da reciclagem é maior que o valor do material obtido [5].

A reutilização da bateria em uma segunda aplicação seria a opção mais adequada. Porém, para realizar a análise do estado e potencial uso da bateria, é necessário coletar uma

grande quantidade de dados sobre o seu funcionamento e características que posteriormente são utilizados para estimar o estado de saúde da bateria através de métodos matemáticos.

O dispositivo amplamente utilizado em testes de baterias, capaz de coletar dados para avaliar a performance e rendimento é denominado de *Data Logger*. [6]. Seu uso pode determinar as condições de funcionamento da bateria para realocá-la em um segundo uso.

Com os dados coletados através do dispositivo, é possível determinar características de funcionamento da bateria, como o *State of Charge* (SOC), que representa uma razão entre a capacidade atual sobre a capacidade total de carga da bateria, [7], o *State of Health* (SOH), definido como a habilidade de armazenar e fornecer energia [8] e o *Rest of Useful Life* (RUL) que se refere ao número de ciclos para que a capacidade máxima disponível da bateria decaia a um valor limite de falha [9].

Apesar de possuir dispositivos comerciais que atendem à demanda abordada, o seu custo elevado faz com que seja inviável sua aplicação em projetos com baixo orçamento, sendo necessário buscar por uma alternativa que possa desempenhar o mesmo propósito com um custo reduzido. Percebeu-se então que é possível desenvolver uma solução através da aplicação de conceitos de Internet das Coisas e Sistemas Embarcados.

Os dispositivos IoT (*Internet of Things*) permitem a comunicação entre o mundo físico e digital através do uso de microcontroladores e diferentes sensores e atuadores. A crescente aplicação dessas tecnologias se deu por conta do desenvolvimento dos protocolos de comunicação [10].

A aplicação dos conceitos IoT se dá através de um sistema embarcado, que consiste em um sistema eletrônico composto por diversos componentes de *hardware* e *software* que podem executar juntos uma função dedicada, independentemente ou como parte de um sistema maior [11].

Através de investigações relacionadas à integração de sistemas embarcados no contexto do monitoramento e coleta de dados de bateria em tempo real, foi possível constatar a existência de trabalhos prévios documentados, os quais foram empregados como fontes de referência para o desenvolvimento deste trabalho.

No trabalho de Çeven [12], é realizado o monitoramento das células de baterias de lítio através de um módulo ESP32. Nele foram utilizados vinte e quatro pacotes de baterias com oito células cada. Os dados de temperatura, corrente e tensão coletados foram enviados via protocolo MQTT até um dispositivo Raspberry Pi, onde era armazenado em uma base de dados.

Na pesquisa de Wang [13], foi desenvolvido um sistema de monitoramento de bateria com interface para diagnóstico. Foi utilizada uma plataforma Arduino Nano juntamente com o Circuito Integrado (CI) de aquisição de dados LTC6811.

Com base no exposto, este trabalho tem como objetivo desenvolver um *Data Logger* de baixo custo, utilizando sensores para captação de dados de tensão, corrente e temperatura das células de bateria de lítio, que serão processados por um microcontrolador e enviados a um banco de dados localizado na nuvem, através de um protocolo de comunicação Wi-Fi, a fim de contribuir para futuros estudos e soluções de segunda vida da bateria.

II. METODOLOGIA

O desenvolvimento deste projeto se dá a partir de uma metodologia elaborada por Wolf [14], baseada em níveis de abstração. Esta metodologia prevê 5 etapas, sendo elas:

- Requisitos: Podem ser divididos em funcionais, os quais são relativos às funções do sistema; e não funcionais, os quais estão relacionadas as restrições do sistema;
- Especificação: Detalhamento dos requisitos do projeto, contendo informações precisas e compreensíveis o suficiente para atender aos requisitos do sistema.
- Arquitetura: Descreve como o sistema é implementado a fim de atender as especificações previamente definidas. A arquitetura é normalmente representada a partir de um diagrama de blocos e pode ser dividida em *hardware* e *software*.
- Componentes: Define quais são os componentes, tanto de *hardware* quanto de *software* que atendem as especificações projetadas na arquitetura.
- Integração: Consiste em um plano de ação para a junção dos diversos módulos desenvolvidos ao longo do projeto.

A. Requisitos

Os requisitos foram definidos com base em uma análise do projeto.

Como requisitos funcionais, o sistema deve coletar dados de baterias de lítio e armazená-los, tanto localmente quanto em nuvem, através de rede de comunicação Wi-Fi, além de ter a capacidade para coletar dados de diversos pacotes de baterias simultaneamente.

E como requisitos não funcionais, o sistema deve ser capaz de realizar o *back-up* local das informações coletadas mesmo com a ausência de rede de comunicação Wi-Fi, além de assegurar o envio dessas informações para a nuvem após o reestabelecimento do sinal da rede.

B. Especificações

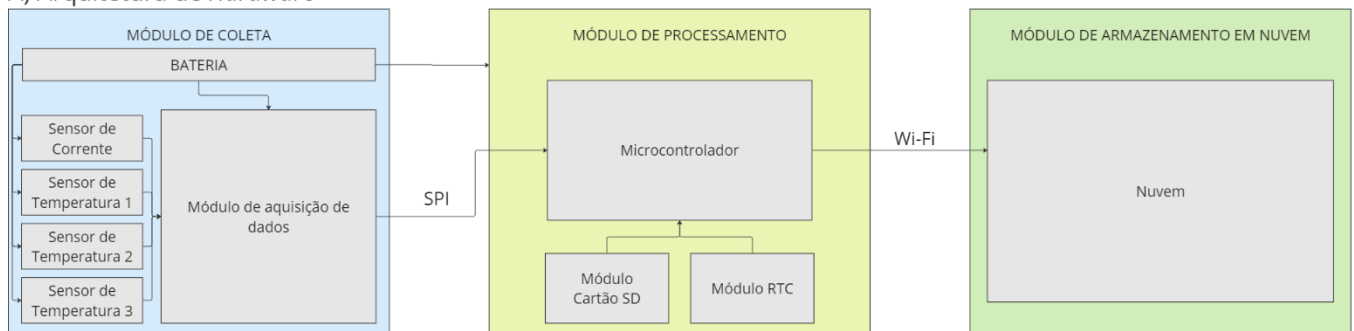
Com base nos requisitos definidos, realizou-se a especificação das tecnologias utilizadas. Os dados a serem monitorados são de tensão, corrente e temperatura. A coleta de dados deve ser realizada por sensores independentes através de um circuito integrado de coleta de dados. Este CI deve se comunicar com um microcontrolador que realizará o processamento das informações. Este microcontrolador deve possuir capacidade de armazenamento local.

O protocolo de comunicação MQTT será implementado para garantir o envio dos dados à nuvem. É necessário também um banco de dados para armazenar essas informações na nuvem.

C. Arquitetura

Foi definida as arquiteturas de *hardware* e *software* do sistema, conforme a figura 1.

A) Arquitetura de Hardware



B) Arquitetura de Software

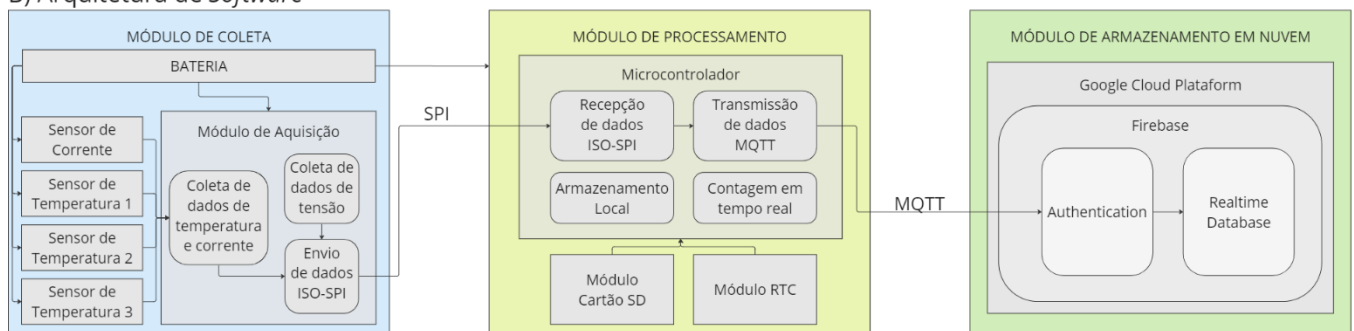


Figura 1 – Arquiteturas de *Hardware* e *Software*

A arquitetura foi concebida de forma modular contendo: Módulo de Coleta de Dados, Módulo de Processamento, Módulo de Armazenamento em Nuvem.

Na arquitetura de *hardware* o Módulo de Coleta de dados contém os dispositivos integrados diretamente à bateria. Ele é responsável por fazer a captura de dados dos sensores, além de realizar o envio para o Módulo de Processamento.

O Módulo de Processamento, por sua vez, é composto por um microcontrolador, um módulo RTC (*Real Time Clock*) e um módulo de cartão SD (*Secure Digital*). O Módulo de Processamento é responsável por realizar o processamento e envio de dados para nuvem, bem como efetuar o armazenamento local das informações.

O Módulo de Armazenamento em Nuvem é responsável pelo armazenamento externo de dados utilizando tecnologia de computação em nuvem.

Na arquitetura de *software*, o Módulo de Coleta de Dados possui três componentes: o primeiro responsável pela coleta de dados de temperatura e corrente dos respectivos sensores; o segundo responsável pela coleta de dados de tensão; e o terceiro é responsável pelo envio dos dados destes sensores para o Módulo de Processamento de Dados.

Já o Módulo de Processamento possui quatro componentes de *software*: o primeiro responsável pela recepção dos dados e pelo tratamento dessas informações; o segundo é responsável pela contagem em tempo real; o terceiro é responsável pelo envio dos dados dos sensores, bem como a data e a hora; e o quarto é responsável pelo armazenamento local.

D. Componentes

O projeto pode ser dividido em componentes de *hardware* e componentes de *software*. Os *hardwares* selecionados foram três sensores de temperatura do tipo NTC B3435 10k Ohms e um sensor de corrente HCS-LTR5 de 500A baseado em efeito hall. Além disso, também foi definido um circuito integrado para coleta de dados de tensão LTC6811, o qual possui capacidade para coleta de até doze células de bateria, podendo operar de maneira modular.

Foi selecionado o módulo ESP32, que possui o microcontrolador ESP32, com Wi-Fi integrado ao chip. Além disso, são necessários um módulo RTC DS3231 e um módulo de leitura e escrita no cartão SD.

Foram definidos os protocolos de comunicação serial SPI (*Serial Peripheral Interface*) e I2C (*Inter-Integrated Circuit*) para comunicação entre os dispositivos com o módulo ESP32.

Como componentes de *software*, o protocolo de mensagens MQTT (*Message Queuing Telemetry Transport*) foi selecionado para realizar o envio de dados a nuvem. Definiu-se por utilizar o padrão de mensagens no formato JSON (*JavaScript Object Notation*) para troca de dados e armazenamento.

Foi selecionada a nuvem da Google Cloud Platform e optou-se pelo conjunto de serviços Firebase, utilizando as aplicações disponibilizadas de Authentication e Realtime Database (RTDB).

E. Integração

Para a integração do sistema, foi elaborado um plano de ação de seis passos, conforme ilustrado na figura 2.

Este plano consiste na implementação dos três módulos apresentados de forma separada, a fim de criar o sistema completo.

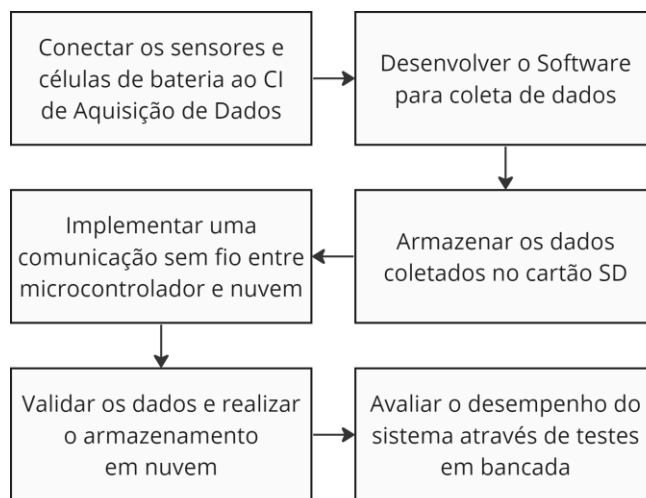


Figura 2 – Fluxograma de Implementação

III. DESENVOLVIMENTO

O projeto foi desenvolvido utilizando um módulo de bateria de oito células, conforme pode ser observado na figura 3. Cada célula possui cerca de 3,2V de tensão, totalizando aproximadamente 25,6V e o módulo possui uma corrente nominal de 5,5A.



Figura 3 – Módulo para o desenvolvimento do projeto

1) Módulo de Coleta de Dados

Para o desenvolvimento deste módulo foi utilizado uma placa de prototipação LTC6811-12S-AL, da marca MAX KGO, a qual possui um chip LTC6811 integrado com as entradas e saídas necessárias para esta aplicação. Na figura 4 é possível observar o CI LTC6811, juntamente com as conexões para até doze células de bateria, cinco GPIO's (*General Purpose In/Out*) e dois canais de comunicação ISO-SPI.

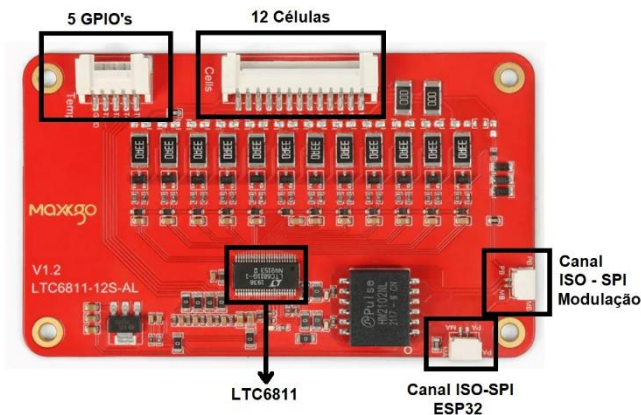


Figura 4 – Placa de prototipação LTC6811

Nela foram realizadas as conexões de três sensores de temperatura do tipo NTC de 10 k Ω e um sensor de corrente baseado em efeito hall HCS-LTR5 de 500A, utilizando quatro das cinco GPIO's disponibilizadas pelo CI. A quinta GPIO é reservada para um sensor NTC aplicado para monitoramento da temperatura da própria placa de prototipação o qual não foi utilizado durante o desenvolvimento desse projeto.

Dois sensores de temperatura foram posicionados nos polos positivo e negativo da bateria. O terceiro sensor foi alocado na lateral do módulo. O sensor de corrente ficou posicionado na saída da bateria. A figura 5 demonstra o posicionamento dos sensores de temperatura no módulo de bateria.

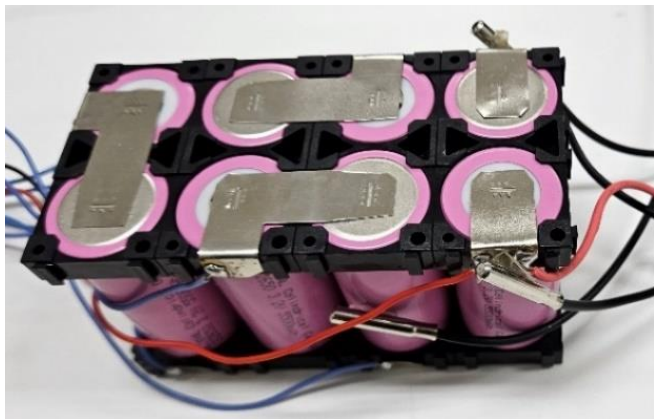


Figura 5 – Sensores de Temperatura

A placa LTC6811-12S-AL foi conectada à todas as células da bateria, utilizando oito dos doze canais de comunicação. Para coletar os dados de tensão das células da bateria foi utilizado o sensor interno desta mesma placa.

Através do protocolo de comunicação serial ISO-SPI, foi estabelecido um canal de comunicação entre a placa e o módulo ESP32.

Para realizar a conversão entre a comunicação ISO-SPI da placa de prototipação e a comunicação SPI do módulo ESP32, foi utilizada uma placa conversora ISO-SPI para SPI. O conversor possui uma conexão ISO-SPI com o LTC6811, bem como outra conexão SPI com o módulo ESP32 e funciona a partir do transformador de sinal CI LTC6820.

A figura 6 retrata a placa de prototipação com as conexões da bateria, os sensores de temperatura o sensor de corrente.

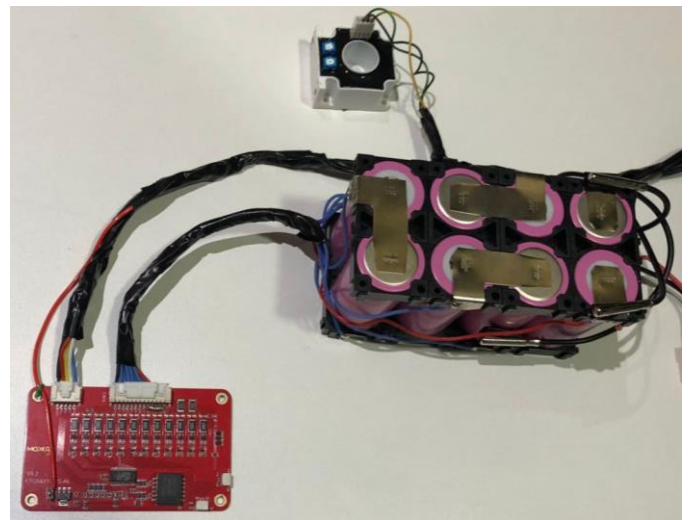


Figura 6 – Integração entre Sensores e Circuito Integrado

2) Módulo de Processamento

Este módulo desempenha o papel de executar o processamento e encaminhar dados para a nuvem, ao mesmo tempo que efetua o armazenamento local das informações.

Para o seu desenvolvimento, primeiramente foi utilizado um software genérico com uma biblioteca específica para o uso do CI LTC6811 desenvolvida pela empresa fabricante Analog Devices. Esta biblioteca em questão funciona de forma manual e foi desenvolvida para ser aplicada em dispositivos da plataforma Arduino. Através de comandos enviados via serial, ela é capaz de realizar a leitura da tensão das células de bateria além de outras funções, como leitura de tensão das GPIO's.

Os primeiros testes do *software* e da biblioteca foram realizados utilizando a plataforma Arduino UNO R3. Os dados de tensão das oito células da bateria bem como a leitura de tensão das cinco GPIO's foram realizados de maneira satisfatória. Em seguida, realizou-se um estudo do programa a fim de manter somente as funções de leitura das células e das GPIO's necessárias para coleta dos dados dos sensores.

O *software* teve seu desenvolvimento baseado na biblioteca e foi programado em etapas, utilizando a IDE (*Integrated Development Environment*) Arduino. A primeira etapa foi a coleta das informações de leitura dos sensores. O programa foi adaptado para o ESP32, alterando parâmetros da comunicação SPI para que funcionasse de forma satisfatória.

A próxima etapa do desenvolvimento foi o tratamento de sinal dos sensores, uma vez que as entradas de propósito geral realizavam a leitura da tensão dos sensores.

Para os sensores de temperatura NTC, foi desenvolvida uma biblioteca para tratamento e conversão dos valores lidos de tensão (V) para graus Celsius ($^{\circ}\text{C}$). Essa biblioteca utiliza princípios da equação de Steinhart-Hart e recebe como parâmetros a resistência em série (R), a resistência nominal (R_0), o coeficiente β (B) e a temperatura nominal (T_0) do sensor para efetuar os cálculos de conversão.

A conversão da tensão em temperatura é realizada através de um divisor de tensão utilizando a resistência serial e a

resistência do sensor. A partir da variação da resistência do sensor, é possível calcular a temperatura.

Também através de cálculos matemáticos, realizou-se a conversão da tensão (V) lida pelo conversor ADC (*Analog-to-digital converter*) do sensor de corrente, em amperes (A).

O sensor HCS-LTR5 de 500A é alimentado por 5V e possui uma saída que varia $\pm 2V$, com a referência fixada em 2,5V, fazendo com que seja capaz de realizar a leitura de correntes negativas e positivas.

Para realizar o tratamento de sinal, foi efetuada a leitura da tensão do valor da corrente. Em seguida, foi diminuído do valor de *offset* para encontrar o valor real de leitura do sensor.

Utilizando a resolução do ADC, foi possível encontrar um valor da corrente em Amperes (A). Por fim, foi efetuada uma amostragem teste com o intuito de realizar um ajuste na fórmula para diminuir o ruído da leitura do sensor.

Após a confirmação de que os dados foram coletados no ESP32, foi adicionado um módulo RTC a fim de armazenar a data e a hora em que as informações foram coletadas.

O uso de um módulo externo se dá por conta do RTC interno do ESP32 não possuir bateria para manter a contagem de tempo em caso de desligamento, característica essa crucial para o bom funcionamento do *Data Logger*.

O *software* utiliza o protocolo NTP (*Network Time Protocol*) para realizar a sincronização do RTC com a data e a hora atualizada utilizando um servidor. Dessa forma é possível manter a manutenção da sincronização precisa do tempo no programa.

Para a próxima etapa do desenvolvimento, foi utilizado a biblioteca ESP32-Firebase para configurar um canal de comunicação sem fio entre o banco de dados e o módulo ESP32.

A biblioteca utiliza um protocolo de comunicação baseado em MQTT, empregando tópicos para envio de informações. A escolha do canal de comunicação é realizada visando a modulação futura dos dispositivos de coleta de dados, os quais podem ser ampliados de maneira que, diversos Módulos de Coleta possam trabalhar com um único Módulo de Armazenamento em nuvem.

A biblioteca estabelece um canal de comunicação a partir da *API_KEY* do banco de dados bem como o *URL* da aplicação. A biblioteca ainda define um usuário e senha para autenticar os dados enviados ao Firebase.

O formato da mensagem enviada é do tipo JSON, o qual é configurado usando funções da biblioteca ESP32-Firebase.

Um arquivo JSON é um formato de texto utilizado para estruturar dados e facilitar a comunicação entre servidores e clientes. Ele consiste em pares de chave-valor organizados hierarquicamente, tornando eficiente a comunicação entre diferentes plataformas e linguagens de programação. A figura 7 retrata a estrutura de um arquivo do tipo JSON, com as chaves representadas em azul e os valores representados entre aspas.

```
celula_1: "3.29"  
celula_2: "3.29"  
celula_3: "3.29"  
celula_4: "3.29"  
celula_5: "3.29"  
celula_6: "3.29"  
celula_7: "3.29"  
celula_8: "3.29"  
temperatura_1: "22.30"  
temperatura_2: "22.30"  
temperatura_3: "22.46"  
corrente: "59.85"
```

Figura 7 – Estrutura de Arquivo JSON

O arquivo JSON é então enviado à nuvem e, ao mesmo tempo, salvo no *back-up* SD. A Figura 8 retrata a lógica de funcionamento do *software* e sua sequência, desde a inicialização, coleta, processamento e envio ao banco de dados na nuvem.

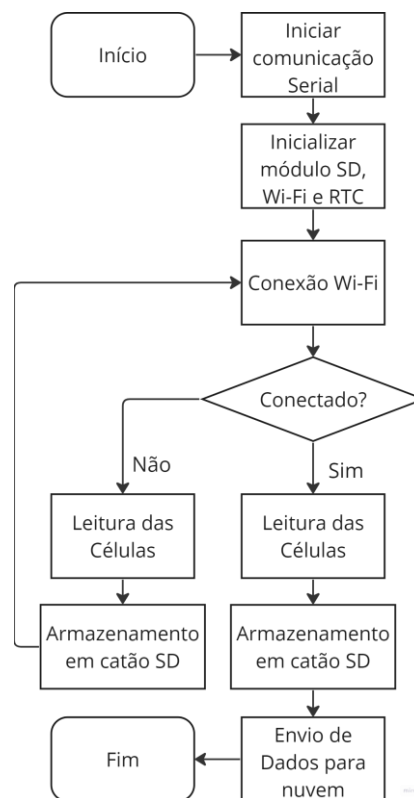


Figura 8 – Lógica de Funcionamento do Programa

Ao fim do processamento dos dados coletados, foi necessário fazer o *back-up* das informações. O objetivo principal de um *back-up* é evitar a perda de dados, sendo necessário um local físico para armazenamento dos dados.

Para o caso do projeto em questão, pelo fato da plataforma microcontrolada selecionada não possuir um módulo de leitura/escrita de cartão SD interno, foi necessário realizar a adição de um módulo externo.

Durante o desenvolvimento do *software*, foi incluído um segundo canal de comunicação SPI para transmissão de dados entre o módulo ESP32 e o cartão SD. Para isso, foi utilizado o canal HSPI (*High-Speed Peripheral Interface*) para comunicação com o cartão SD.

A lógica desenvolvida para a programação de *back-up* consiste em dois arquivos no mesmo cartão SD. O arquivo “*back-up*” possui todos os JSON lidos ao decorrer do programa enquanto o arquivo “*back-up offline*” é utilizado somente quando não é possível enviar dados a nuvem.

Quando o canal de comunicação entre o módulo ESP32 e a nuvem é estabelecido de maneira satisfatória, é feita uma verificação no arquivo de “*back-up offline*” a fim de garantir que todos os dados lidos, mesmo quando não foi possível estabelecer um canal de comunicação com a nuvem, fossem registrados no banco de dados. Uma vez enviados, os dados salvos no arquivo “*back-up offline*” são excluídos do cartão SD.

Um diagrama contendo a lógica de funcionamento do cartão SD é descrito através da figura 9.

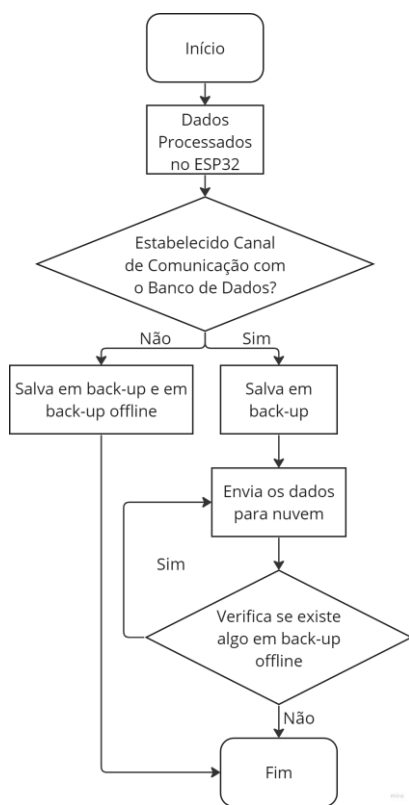


Figura 9 – Lógica do Módulo SD

Além de realizar o armazenamento em uma unidade sólida, o trabalho propõe o envio e armazenamento dos dados coletados em nuvem.

3) Módulo de Armazenamento em Nuvem

O armazenamento em uma unidade remota visa proteger os dados contra perdas ou danos, além de facilitar o compartilhamento ou acesso a informação através da internet.

O banco de dados selecionado foi a plataforma do Google Firebase, por se tratar de uma aplicação de fácil utilização e implementação.

A transferência dos dados capturados pelo módulo ESP32 ocorre por meio da integração com a biblioteca Firebase-ESP32. Esta ferramenta possibilita a criação de arquivos JSON a partir das informações coletadas, permitindo o encaminhamento direto desses dados para o banco de dados.

A fim de efetuar a transmissão para o Google Cloud na aplicação Firebase, é essencial empregar a autenticação para validar o acesso na plataforma. Essa funcionalidade possibilita a legitimação do usuário através de métodos de inscrição, como SMS, e-mail ou de maneira anônima.

No contexto desta aplicação em particular, um endereço de e-mail e uma senha foram registrados para autenticar o acesso ao banco de dados.

O banco de dados em tempo real, *Realtime Database* é empregado como um meio de armazenamento de dados na nuvem. Após o procedimento de validação, o RTDB utiliza regras específicas para operações de leitura e escrita, acrescentando uma camada adicional de segurança ao acesso às informações.

IV. RESULTADOS E DISCUSSÕES

Ao fim do desenvolvimento do projeto, foi realizada a montagem do protótipo em uma placa de circuito impresso com todos os componentes de *hardware* previamente validados. A placa projetada foi alocada em uma caixa de plástico. A figura 10 retrata a montagem da versão final da placa contendo o módulo ESP32, módulo cartão SD, módulo RTC DS3231, placa de conversão ISO-SPI para SPI e módulo regulador de tensão de 24V para 5V.



Figura 10 – Circuito de Processamento

O *Data Logger* desenvolvido pode ser alimentado pela mesma bateria a qual os dados estão sendo coletados, utilizando 24V. A conversão de tensão é realizada através de um módulo regulador 24V/12V para 5V. Essa abordagem simplifica o transporte e a disposição do equipamento durante a sua montagem.

Entretanto, caso o módulo seja empregado em um conjunto de bateria com tensão superior a 24V, é possível alimentá-lo por meio de uma fonte externa energia.

Após o desenvolvimento dos Módulos de Coleta, Processamento e Armazenamento do *Data Logger*, foi realizada a validação dos dados coletados.

Esta operação se deu através de testes em bancada para comparação dos resultados obtidos com os valores coletados por instrumentos de medição certificados.

Para realizar a validação dos dados de temperatura, foi efetuada a aferição do módulo à temperatura ambiente através de um multímetro ET-2042F. Foram realizadas três medições do equipamento para cada sensor de temperatura posicionado.

Os valores lidos apresentaram um erro percentual de 1,5%, os quais ficaram dentro da faixa de erro de operação indicados pelo fabricante do dispositivo ($\pm 2\% + 4$)°C [15], conforme pode ser observado na tabela 1.

Posição Sensor	<i>Data Logger</i> (°C)	ET-2042F (°C)
Polo Negativo	24,76 °C	24 °C
Polo Negativo	24,48 °C	24 °C
Polo Negativo	24,35 °C	25 °C
Polo Positivo	24,82 °C	24 °C
Polo Positivo	24,28 °C	25 °C
Polo Positivo	24,33 °C	25 °C
Lateral	25,01 °C	24 °C
Lateral	24,12 °C	23 °C
Lateral	24,14 °C	23 °C

Tabela 1 – Medidas de Temperatura

Os valores de tensão e corrente, por sua vez, foram aferidos utilizando um multímetro UNI-T UT210E PRO. Os dados de tensão coletados podem ser observados na tabela 2, enquanto os dados de corrente estão destacados na tabela 3.

Tensão (V)	<i>Data Logger</i> (V)	UNI-T UT210E PRO (V)
Célula 1	3,27 V	3,27 V
Célula 2	3,28 V	3,28 V
Célula 3	3,28 V	3,27 V
Célula 4	3,28 V	3,28 V
Célula 5	3,28 V	3,28 V
Célula 6	3,28 V	3,28 V
Célula 7	3,28 V	3,28 V
Célula 8	3,28 V	3,28 V

Tabela 2 – Medidas de Tensão

Percebe-se que os valores de tensão coletados pelo *Data Logger* foram precisos, apresentando um erro padrão de 0,3%, valor praticamente nulo quando comparado com a medição realizada pelo multímetro UNI-T UT210E PRO.

Para coleta dos valores de corrente, foi conectada uma carga resistiva na saída da bateria a fim de analisar a corrente

de descarga. Também foi aferido os valores de corrente da bateria sem carga.

Carga	<i>Data Logger</i>	UNI-T UT210E PRO
Conectada	3,05 A	3,14 A
Conectada	3,10 A	3,16 A
Desconectada	0,20 A	0,08 A
Desconectada	-0,07 A	0,08 A

Tabela 3 – Medidas de Corrente

Observa-se que o sensor de corrente possui um erro aleatório proveniente de ruídos do sensor. Os valores estão de acordo uma vez que o erro padrão de 2,3% ficou próximo da margem indicada pelo fabricante do instrumento de medição $\pm(2\%+3)A$ [16]. A figura 11 apresenta o teste realizado com a carga resistiva.

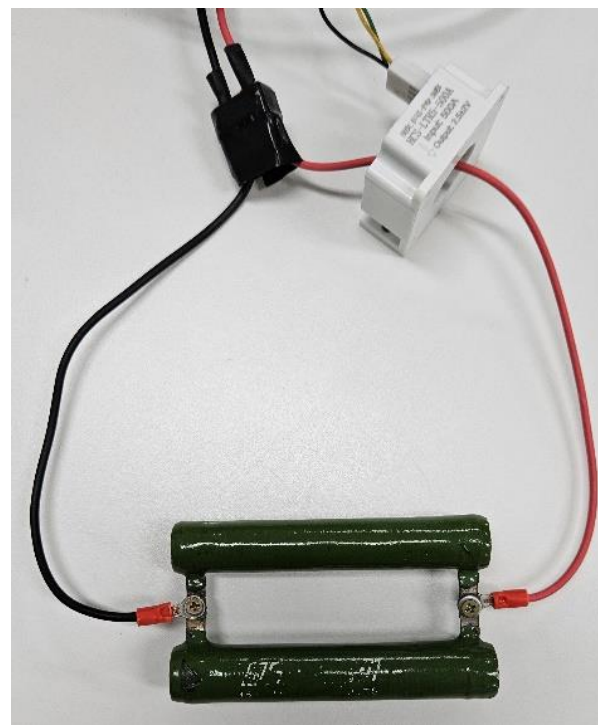


Figura 11 – Carga Resistiva e Sensor de Corrente

Ainda com a carga resistiva conectada, foram realizadas medições de temperatura e constatou-se que a resposta do sensor NTC foi mais lenta em relação ao sensor de temperatura do multímetro, todavia, as aferições realizadas estavam de acordo com a temperatura do equipamento.

A estrutura de armazenamento do banco de dados possui o formato:

- Usuários
 - ID do Usuário
 - Leituras
 - Horário da Coleta
 - Tensão
 - Corrente
 - Temperatura

Dessa forma é possível armazenar as informações de forma organizada, além de possibilitar que outros módulos de

coleta sejam incluídos no mesmo banco de dados, alterando o *User ID*. A figura 12 apresenta uma captura de tela, da aplicação RTDB, demonstrando a estrutura dos dados coletados.

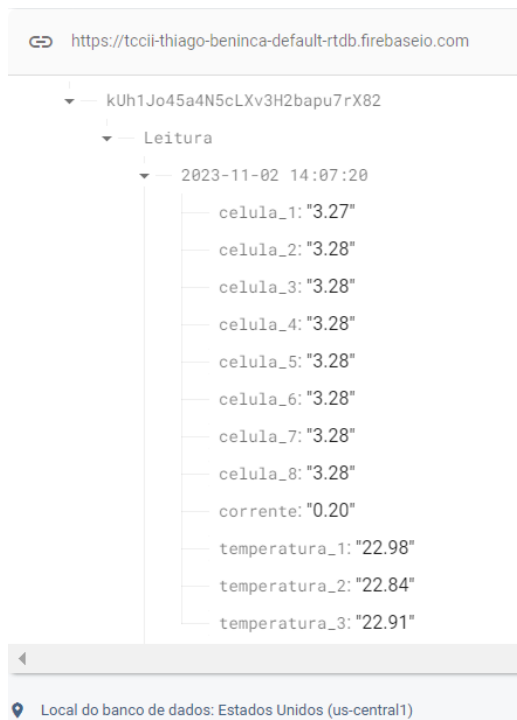


Figura 12 – Estrutura de Armazenamento

Os dados armazenados em nuvem foram comparados com os dados salvos no cartão SD. Primeiramente, o arquivo “back-up” foi analisado e constatou-se que todos os dados enviados para o banco de dados também foram salvos no arquivo de texto. A figura 13 demonstra a estrutura dos dados salvos no cartão SD. É possível observar a estrutura do arquivo do tipo JSON salvo no formato de texto em um arquivo de extensão .txt.

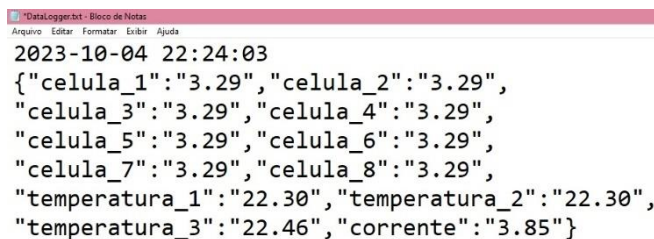


Figura 13 – Arquivo back-up do módulo SD

Por fim, foi realizado um teste para validar se os dados coletados seriam salvos caso a conexão com a internet fosse interrompida.

Foram realizadas 20 gravações de dados no cartão SD com o Wi-Fi desconectado. Percebeu-se que, durante a queda do sinal de internet, todos os dados coletados foram salvos com êxito no arquivo de texto “back-up offline”.

Após o reestabelecimento da conexão Wi-Fi, 19 das 20 gravações, representando 95% dos dados no cartão SD, foram enviados para o banco de dados presente na nuvem. Os 5% das gravações restantes retornaram um erro ao serem enviadas ao banco de dados e foram excluídas após a tentativa de envio.

A figura 14 retrata o sistema completo, com os módulos de coleta e processamento de dados integrados, bem como, uma carga resistiva conectada à bateria.



Figura 14 – Módulo de Coleta, Aquisição e Carga Resistiva

Esse sistema é modular, podendo incorporar diversos Módulos de Coleta para trabalhar com a aquisição de dados de diferentes pacotes de baterias. A sua escalabilidade é praticamente ilimitada em relação à quantidade de Módulos de Coleta, entretanto, é importante observar que o processamento das informações atinge um ponto de saturação, no qual a aquisição dos dados se torna tão lenta que a adição de novos Módulos de Coleta fica inviável [17].

É importante destacar que o *Data Logger* desenvolvido foi validado apenas para um módulo de baterias. No caso de adaptar novos módulos, formando um pacote de baterias, seria necessário realizar uma revalidação dos dados para assegurar o pleno funcionamento do sistema.

Além disso, seria necessário fazer adaptações no código do *software* para reconhecer diferentes sensores, garantindo o processamento preciso dos dados.

Quanto ao limite de armazenamento em nuvem, o *Data Logger* dispõe de espaço adequado nos servidores da Google Cloud Platform. Caso seja necessário um aumento da capacidade de armazenamento, é possível adquirir através da plataforma Firebase.

Em relação à escalabilidade na nuvem, é possível estabelecer diferentes usuários por meio de novas autenticações. Esses usuários podem enviar informações para o mesmo banco de dados, requerendo apenas pequenas adaptações no *software* para permitir que diferentes usuários possam ler/escrever dados, não impondo limitações à escalabilidade do projeto.

A figura 15 demonstra um diagrama de implementação de três Módulos de Coleta atuando em um conjunto de baterias. No diagrama, os dados dos três Módulos são processados por somente um Módulo de Processamento. Os dados são enviados para a aplicação Firebase através de três serviços de Authentication, o qual valida o envio à plataforma e autoriza a escrita da informação no RTDB.

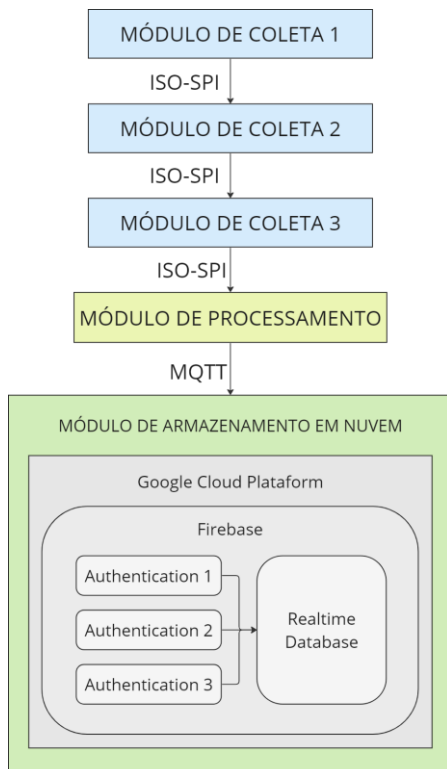


Figura 15 – Estrutura com Três Módulos de Coleta

V. CONCLUSÕES

Com base nos resultados observados, é possível inferir que os objetivos do presente trabalho foram alcançados, mediante o desenvolvimento de um *Data Logger* de baixo custo, capaz de adquirir dados provenientes de uma bateria e armazená-los em um banco de dados na nuvem.

O protótipo demonstrou sua funcionalidade, requerendo apenas intervenções mínimas por parte do usuário para operar de forma satisfatória.

Percebeu-se que os dados salvos no cartão SD ocupam cerca de 25 KB a cada 100 gravações. Considerando que a frequência de coleta de dados seja 1 gravação por hora, um cartão de 16 GB pode durar cerca de 7,6 mil anos operando de forma satisfatória.

No entanto, é perceptível que, para adaptações em diversos ambientes, torna-se necessário efetuar o ajuste da conexão Wi-Fi correspondente ao local em questão via programação. Outro aspecto relevante a ser destacado refere-se à necessidade de posicionar o dispositivo em um local onde a rede apresente um sinal de qualidade.

Além disso, apesar de algumas perdas durante o processo de envio dos dados para a nuvem após o reestabelecimento do sinal Wi-Fi, todas as leituras realizadas foram salvas no cartão SD de forma satisfatória e os sensores demonstraram uma notável precisão.

VI. TRABALHOS FUTUROS

Com o *Data Logger* desenvolvido, tem-se o objetivo de utilizar o dispositivo para realizar a coleta de dados do funcionamento de bateria.

A partir dos dados coletados, será possível realizar uma análise de saúde e do estado atual das células da bateria a fim de estudar as possíveis falhas recorrentes em seu funcionamento, bem como facilitar o emprego dessas baterias em um segundo uso.

Além disso, com uma grande quantidade de dados coletados, será necessário empregar técnicas de análise de dados, bem como uma dashboard para visualização dos principais parâmetros.

Para que seja possível coletar dados de diferentes dispositivos, pode-se realizar testes e ajustes no *software* para leitura de diferentes módulos de bateria de forma automatizada, além da adição da biblioteca Wi-Fi Manager, a qual permite realizar a conexão Wi-Fi em diferentes locais sem alteração do código.

Por fim, pode-se também aprimorar a interface de comunicação com o usuário através da implementação de sinalização para garantir o pleno funcionamento do dispositivo.

VII. REFERÊNCIAS

- [1] WANG, Jiwei et al. **Lifetime prognostics of lithium-ion battery pack based on its early cycling data and complete degradation information of battery cells**, [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 961–965.
- [2] HANNAN, M. A. et al. **State-of-the-Art and Energy Management System of LithiumIon Batteries in Electric Vehicle Applications: Issues and Recommendations**. IEEE Access. [S.l.]: Institute of Electrical and Electronics Engineers Inc. , 20 mar. 2018
- [3] FARZAN MOGHADDAM, A.; VAN DEN BOSSCHE, A. **Flyback Converter Balancing Technique for Lithium Based Batteries**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 1 maio 2019.
- [4] KAMALANATHAN, A. et al. **Impact of Used Battery Disposal in the Environment**. Linguistics and Culture Review, 2 dez. 2021 v. 5, n. s1, p. 1276–1286.
- [5] FRONTERA, N.; CAPPUCCINO, G. **Optimal Traction Battery Module Sizing for Cost-effective Second Life Reuse**. [S.l.: s.n.], 2020.
- [6] SOMAKETTARIN, N. et al. **An evaluation of battery energy efficiency with multi-step sampling rate recording for DC data loggers**. Energy Reports, 1 mar. 2023, v. 45 9, p. 866–872.
- [7] JEEWANDARA, J. M. D. S.; KARUNADASA, J. P.; HEMAPALA, K. T. M. U. **SOC Level Estimation of Lithium-ion Battery Based on Time Series Forecasting Algorithms for Battery Management System**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 43–49.
- [8] VENNAM, G.; SAHOO, A.; AHMED, S. **A Novel Coupled Electro-thermal-aging Model for Simultaneous SOC, SOH, and Parameter Estimation of Lithium-ion Batteries**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2022. p. 5259–5264.
- [9] ZHAO, J. et al. **Review on RUL Prediction Methods for Lithium-ion Battery**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2022. p. 1501–1505.
- [10] MALEKI, N. et al. **DynaSens: Dynamic Scheduling for IoT Devices Sustainability**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2022.
- [11] PASRICHA, S. **Embedded Systems Education: Experiences with Application-Driven Pedagogy**. IEEE Embedded Systems Letters, 1 dez. 2022. v. 14, n. 4, p. 167–170.
- [12] ÇEVEN, S. et al. **Development of IoT Based Battery Management System**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021.
- [13] WANG, X. et al. **Design of Lithium-ion Battery Monitoring System with Diagnostic Interface**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 1 out. 2020. p. 186–191.

- [14] WOLF, Wayne. **Computers as Components: principles of embedded computing system design**. 2. ed. Burlington, MA: Elsevier, 2008.
- [15] MINIPA DO BRASIL LTDA. **MULTÍMETRO DIGITAL Digital Multimeter ET-2042F**. 19 de novembro de 2019. 28 páginas.
- [16] UNI-TREND TECHNOLOGY. **UT210E PRO MANUAL DO USUARIO**. Maio, 2018. 3 páginas.
- [17] DEVICES INC, A. **LTC6811-1/LTC6811-2 (Rev. C)**. [S.l: s.n.], 2019. Disponível em: <www.analog.com>.