

DESENVOLVIMENTO DE CAMADAS DE ABSTRAÇÃO DE HARDWARE PARA O FUNCIONAMENTO DE UM PROTÓTIPO PARA IDENTIFICAÇÃO DE FRAUDES EMBUTIDAS POR SOBREPOSIÇÃO DE IMAGENS

Juliana Maria S. de Santana¹; Ricardo Cerqueira Medrado² Gessé Justiniano de O. Junior³;

¹Bolsista; Pesquisa e Desenvolvimento e Inovação - PD&I; julianamaria.s.santana@gmail.com

²Especialista III; Centro Universitário SENAI CIMATEC; Salvador-BA; ricardo.medrado@fieb.org.br

³Consultor II; Centro Universitário SENAI CIMATEC; Salvador-BA; gesseoliveira@fieb.org.br

RESUMO

O desvio de energia impacta na qualidade do fornecimento de energia e ocasionam perdas financeiras para as concessionárias. Tendo isto em vista, o projeto em desenvolvimento consiste em um protótipo de equipamento destinado a detectar desvios de energia elétrica em circuitos embutidos em paredes, exibindo os resultados através de imagens. É parte do seu escopo o desenvolvimento de firmwares embarcados, através de camadas de abstração de hardware, divididas em *drivers*, *devices*, *libraries* e *applications*. Estas estruturas visam o desenvolvimento de uma arquitetura limpa e são projetadas a partir das funcionalidades do protótipo. Assim, no presente artigo, a relação entre essas camadas é exemplificada através do funcionamento de um dos componentes do dispositivo em desenvolvimento, a sua comunicação via bluetooth.

PALAVRAS-CHAVE: Detecção; desvios; microcontrolador; firmware

1. INTRODUÇÃO

Para as concessionárias de energia as perdas não técnicas são um grande problema visto que, estas impactam na qualidade do fornecimento e ocasionam grandes perdas financeiras. Estas perdas estão relacionadas a adulterações no medidor ou ao desvio de energia¹.

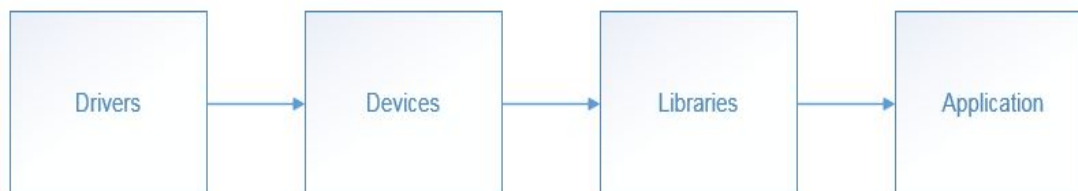
Considerando a parte de desenvolvimento de firmwares embarcados, nesta etapa é importante aplicar alguns princípios arquiteturais através da divisão em camadas, de forma que seja consolidado um limite entre estas, permitindo a criação de uma arquitetura limpa que será reaproveitável e facilitando as manutenções e melhorias².

Portanto, o projeto consiste no desenvolvimento de um protótipo de equipamento destinado a detectar desvios de energia elétrica em circuitos embutidos em paredes, exibindo os resultados através de imagens. O equipamento deverá rastrear os condutores (energizados ou não) do interior da parede e demonstrá-los como camadas de informação sobre a fotografia da parede inspecionada. Assim, fazem parte do escopo do projeto testes de integração em kits de desenvolvimento e o desenvolvimento de firmwares embarcados, através de camadas de abstração de hardware, divididas em *drivers*, *devices*, *libraries* e *applications*.

2. METODOLOGIA

Inicialmente, dedicou-se uma etapa do projeto para o estudo e a análise das plataformas de desenvolvimento do STM32 e das camadas de firmware necessárias para integração dos diversos dispositivos eletrônicos do protótipo. Este estudo foi realizado por meio de um curso no SENAI CIMATEC e teve duração de 3 meses.

Figura 1. Camadas de abstração de hardware



Fonte: Própria, 2020

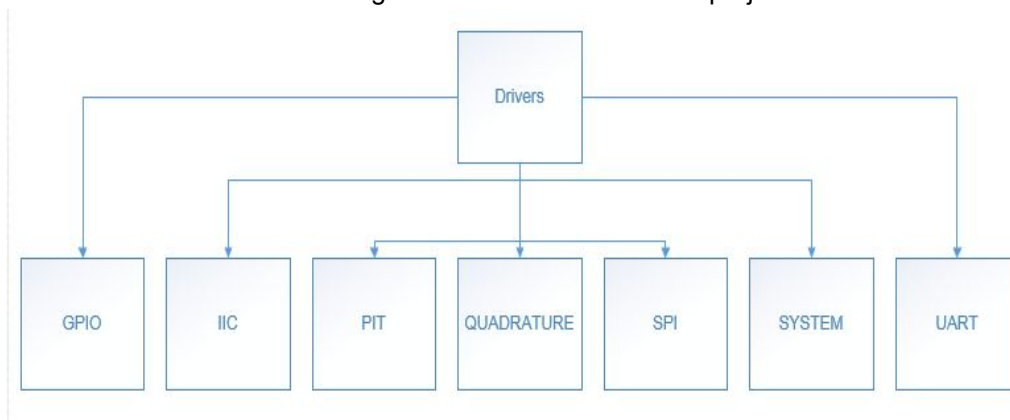
Após esta etapa foram especificadas e estão sendo desenvolvidas as camadas de firmwares do protótipo estruturadas conforme a Figura 1. Esta divisão corresponde a uma ordem hierárquica a qual é aplicada na área de desenvolvimento de firmware do SENAI CIMATEC.

Portanto, as atividades executadas visam o estudo em plataformas de processamento baseados em STM32 (ARM). A partir das quais serão implementados firmwares embarcados, através de camadas de abstração de hardware, divididas em *drivers*, *devices*, *libraries* e *applications*.

3. RESULTADOS E DISCUSSÃO

A camada de driver permite a configuração e o acesso a métodos para gerenciar as funcionalidades dos periféricos internos do microcontrolador. Para o projeto estão sendo utilizados os drivers listados na Figura 2.

Figura 2. *Drivers* utilizados no projeto

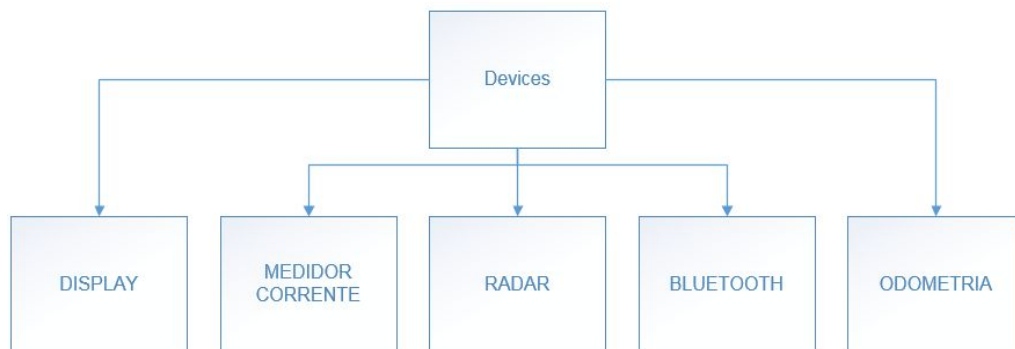


Fonte: Própria, 2020

Estes drivers são responsáveis por manipular os níveis lógicos das entradas e saídas digitais (GPIO - *General Purpose Input/Output*), fornecer métodos, permitir as configurações do periférico e para enviar e receber dados (IIC - *Inter-Integrated Circuit*, SPI - *Serial Peripheral Interface*, UART - *Universal Asynchronous Receiver/Transmitter*), além de configurar o sistema de *clock* do microcontrolador para a criação de *timers* e leitura de frequências (SYSTEM, PIT, QUADRATURE).

A camada *device* apresenta métodos relacionados às funcionalidades dos dispositivos pertencentes ao projeto. Estabelecendo a comunicação destes circuitos externos com o microcontrolador por meio dos drivers citados anteriormente.

Figura 3. *Devices* utilizados no projeto



Fonte: Própria, 2020

Portanto, o protótipo em desenvolvimento é composto pelas estruturas listados na Figura 3 onde os seguintes dispositivos compõem o protótipo: um display é utilizado para exibir informações para o usuário, um magnetômetro é aplicado para obter a intensidade da corrente elétrica, um radar foi inserido no sistema para detectar condutores através das paredes, um módulo *bluetooth* é responsável por enviar os dados da inspeção para um dispositivo celular e os *encoders* que são utilizados no sistema de odometria.

Assim, com relação ao módulo *bluetooth*, por exemplo, na camada *device* uma das rotinas implementada é a de inicialização interna, a qual irá configurar os pinos conforme os drivers que serão necessários para o seu funcionamento. Neste caso, é realizada a configuração do driver GPIO (*General Purpose Input/Output*) com o objetivo de estabelecer uma interface entre o *device* e o microcontrolador e, o driver da UART (*Universal Asynchronous Receiver/Transmitter*) para a recepção e transmissão de dados³.

Já a camada *library* (biblioteca) é projetada visando a execução de uma funcionalidade específica dos dispositivos. Dessa forma, a biblioteca desenvolvida para o *bluetooth* visa gerenciar as funções implementadas na camada *device* estabelecendo um protocolo de comunicação entre o módulo *bluetooth* e o aplicativo. O protocolo escolhido, denominado *XModem*, estabelece a divisão dos dados em pacotes com uma determinada quantidade de bytes, respostas ACK/NACK (se os comandos ou dados recebidos estão corretos/preparados ou não) e, a verificação de erros por meio do método CRC (*Cyclic Redundancy Check*)⁴.

Assim, a rotina desenvolvida apresenta os processos onde, são utilizados alguns bytes para controle do fluxo de dados e, caso seja detectado algum erro através do CRC, o pacote de dados é enviado novamente.

E, por último, a aplicação implementada visa o desenvolvimento do IHM (Interface Homem Máquina) do protótipo e a sua principal responsabilidade é desenvolver as funcionalidades do produto através da manipulação direta de todos os pacotes criados na camada *Library* de modo que, o sistema opere realizando o rastreamento dos condutores armazenando as distâncias percorridas, a presença de materiais e o nível de corrente e estes dados sejam enviados para um aplicativo o qual é responsável por plotar uma imagem por meio da qual será possível detectar a existência de um desvio ou não.

4. CONSIDERAÇÕES FINAIS

O presente artigo abordou as principais camadas de abstração de hardware desenvolvidas no projeto. A partir das quais é possível visualizar as principais funcionalidades do protótipo para que o mesmo seja capaz de identificar fraudes embutidas em paredes.

A organização da arquitetura em camadas permite que alterações, ajustes e correções possam ser realizadas mais rapidamente e, que o fluxo do firmware seja mais eficiente. Com isto, se espera a redução de problemas ao adicionar novas funcionalidades ao protótipo e também permite que as estruturas implementadas possam ser utilizadas em projetos futuros.

Agradecimentos

À Light Serviços de Eletricidade S/A pelo financiamento do projeto de pesquisa e desenvolvimento PD-00382-121/2018, vinculado ao Programa de P&D do Setor Elétrico da Agência Nacional de Energia Elétrica (ANEEL).

5. REFERÊNCIAS

- [1] Perdas de Energia. **ANEEL**,2019. Disponível em: <https://www.aneel.gov.br/metodologia-distribuicao/-/asset_publisher/e2INtBH4EC4e/content/perdas/654800?inheritRedirect=false>. Acesso em: 23 jul. de 2019.
- [2] R. C. Martin, Arquitetura Limpa: O guia do artesão para estrutura e design de software. **Alta Books**, 2019.
- [3] RN4678 Bluetooth® Dual Mode Module Command Reference User's Guide. **Microchip**,2016. Disponível em:< <http://ww1.microchip.com/downloads/en/DeviceDoc/50002506B.pdf>>. Acesso em: 18 Set. de 2019.
- [4] XModem Protocol with CRC. **A. Technologies**. Disponível em: <<http://web.mit.edu/6.115/www/amulet/xmodem.htm>>. Acesso em: 4 Nov. de 2019.

