**Proposta de Projeto Integrador**

**Data: 05/08/2021**

1. **Nome Projeto:**Osciloscópio Arduino

1. **Nome Usuário no GitHub:**vinicresende

1. **Grupo de Alunos:**

|  |  |  |
| --- | --- | --- |
| **RA** | **Nome** | **e-mail** |
| 0030481923037 | Gabriel Jesus Porfirio Marques | gabriel.marques16@fatec.sp.gov.br |
| 0030481923043 | Krisângela do Nascimento | krisangela.nascimento@fatec.sp.gov.br |
| 0030481923047 | Paula Pascuin Mendes | paula.mendes2@fatec.sp.gov.br |
| 0030481923024 | Vinicius de Souza Resende da Silva | vinicius.silva353@fatec.sp.gov.br |

1. **Compreensão do Problema:**

Escolas Técnicas e Faculdades e institutos públicos, que disponibilizam cursos na área de eletrônica, precisam de um equipamento que possa servir de analisador dos sinais analógicos e digitais de circuito experimentais, chamado osciloscópio, porém este equipamento é comercializado com resoluções para pesquisas mais profundas e laboratórios, portanto tem um custo elevado. As aplicações didáticas que utilizam osciloscópios não necessitam da resolução dos produtos comerciais, diminuindo o custo de produção, além disto, os atuais equipamentos são baseados em circuitos proprietários, e softwares também proprietários, tornando-os ainda mais caros, caso fossem baseados em um hardware open source (Arduino) e software open source, poderiam ser adquiridos com preços mais baratos e inserido a programação diretamente pelas instituições, viabilizando o ensino laboratorial e ampliando as possibilidades de ensino.

Existe, em alguns repositórios e sites, experimentos com Arduino para função osciloscópio, como na imagem 2, porém não há hoje nenhuma versão final, open source, ou comunidade voltada a este projeto, que poderia ampliar o ensino de eletrônica pelo país, e tornar acessível a aquisição de um equipamento, em uma escala de custo de até 1:10 em relação ao custo do equipamento convencional

Imagem 1 – Osciloscópio Digital Convencional

Interface gráfica do usuário

Descrição gerada automaticamente

Imagem 2 – Osciloscópio com Arduino

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

1. **Proposta de Solução de Software e Viabilidade**

A proposta se diferencia, principalmente, de algumas soluções encontradas na internet, pois a ideia é fazer um sistema de aquisição Wireless, que ainda não é explorado em repositórios na web.

Para proposta de solução, envolve duas etapas distintas:

- Solução de Hardware.

- Solução de Software.

Na solução de Hardware, a proposta é basear-se em Arduino / ESP32, juntamente com um circuito de isolação e amplificação de sinais de tensão, para a entrada e conversão destes sinais analógicos em um sinal digital, através de um conversor Analógico / Digital (A/D) de precisão.

Na solução de Software, haveria ainda duas etapas, a primeira etapa consiste no software embarcado baseado em Sistema Operacional de Tempo Real (Real Time Operational System), no Arduino / ESP32, que seria responsável pela aquisição dos sinais, conversão dos dados lidos, e envio através de wifi para um dispositivo móvel, notebook, ou outro aparelho através de uma aplicação WEB. A segunda solução seria o aplicativo WEB, responsivo, multiplataforma, que compilasse os dados enviados pelo Arduino / ESP32 em um gráfico.

1. **Visão Geral dos Pré-Requisitos**

R1 – Converter uma grandeza analógica de tensão elétrica, em um sinal digital que possa ser aquisitado pelo Arduino / ESP32.

R2 – Converter o sinal digital em informação periódica de tempo real.

R3 – Enviar as informações de forma wireless para um dispositivo.

R4 – Receber as informações wireless de um dispositivo Arduino / ESP32.

R5 – Exibir graficamente as informações recebidas

R6 – Armazenar as informações nos formatos .csv e CONTRADE (padrão IEEE)

A1 – O software Web deve ser responsivo.

A2 – Não possuir gerenciador de banco de dados, para torná-lo leve e portável, salvar arquivos.

A3 – Ter segurança, cada dispositivo Arduino deve ter uma senha padrão que possa ser alterado pelo usuário.

A4 – Ser um aplicativo WEB.

1. **Conceitos e Tecnologias Envolvidos**

**Conceitos:**

- Eletrônica.

- Aquisição de Sinais

- Sistemas Operacionais de Tempo Real

- Programação WEB.

- Comunicação HTTP / Servidor HTTP Embarcado

- Programação de sistemas embarcados (Arduino / ESP32)

**Tecnologias**

- Sistemas embarcados de hardware Livre (Arduino / ESP32)

- Sistemas de aquisição de sinais

- JavaScript e Node.js

7.1 Osciloscópio

O osciloscópio é um aparelho que recebe um sinal elétrico e converte para a forma de onda, sendo esta conversão a variação da tensão do sinal em função do tempo, apresentada no display do aparelho em questão. Pode ser utilizado em inúmeras aplicações, como reparação de eletrodomésticos, manutenção de veículos automotivos, análise de vibrações ou análise de rede de dados.

Além disso o osciloscópio pode, com o uso de um transdutor específico, medir outras grandezas como temperatura, pressão ou luminosidade, porém o osciloscópio é adequado para medir ondas com foco nos sinais periódicos.

Alguns exemplos de ondas:

Texto

Descrição gerada automaticamente

*Imagem 3: Exemplo de ondas. parte da imagem foi obtida em*[*https://www.raisa.com.br/osciloscopios-introducao*](https://www.raisa.com.br/osciloscopios-introducao)

Com um osciloscópio é possível medir grandezas elétricas como:

* Frequência

A frequência é a medida em hertz, Hz, o número de vezes que determinado sinal repete em um segundo

* Amplitude (de tensão)

A intensidade do sinal

* Desfasamento

Aplicado apenas as ondas sinusoidais, o desfasamento representa a diferença entre dois sinais de mesma frequência.

De maneira usual o osciloscópio, podendo ser portátil ou de bancada, tende a exibir graficamente características de determinado sinal elétrico como amplitude, tempo, existência de interferências (ruídos) continuadas, perturbações transitórias e a comparação de entrada e saída de sistemas, permitindo projetar e depurar os mesmos sistemas.

7.2 Arduino

Arduino é um dispositivo que ajuda projetos de eletrônica e programação. Baseado em uma plaquinha com um microcontrolador capaz de processar e armazenar coisas, possui uma linguagem de programação que faz com que seja possível criar um sistema e o dispositivo funcione. É um hardware livre, que quer dizer que qualquer um pode dar sua contribuição para montar, melhorar, modificar e personalizar um. Foi desenvolvido para ser barato, funcional e fácil de programar. Também existem simuladores gratuitos de Arduino virtuais. Com o hardware é necessário ter o software Arduino IDE (Integrated Development Environment, ou em português, Ambiente de Desenvolvimento Integrado) que serve basicamente para programar o hardware em C/C++, um cabo USB também é necessário para conectar o dispositivo a um computador, quando programado o Arduino poderá ser usado de forma independente.

Abaixo está o IDE com um programa já carregado:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Existem diversos modelos de placas de Arduino, para decidir qual utilizar dependerá do projeto a ser desenvolvido, elas possuem pinos que servem como entrada e como saída de informações, a quantidade de pinos varia dependendo do modelo da placa. Os pinos ou portas podem ser digitais, que só entendem valores binários, ou seja, on e off, e podem ser analógicos, que entendem também valores intermediários, não simplesmente ligado e desligado.

Circuito eletrônico com fios

Descrição gerada automaticamente

Com o Arduino é possível fazer uma infinidade de coisas como, criar um brinquedo novo, acender luzes com um sensor de movimento, ligar equipamentos de acordo com o horário, automatizar um carro ou uma casa, regar o solo de acordo com sua umidade, mandar sinais por Bluetooth ou wi-fi para fazer algo solicitado pelo usuário, movimentar um robô, melhorar algo já existente, qualquer coisa é possível, basta ter criatividade.

7.3 Sistemas Operacionais de Tempo Real

Um sistema operacional é um programa ou conjunto de programas cuja função é gerenciar os recursos do sistema (definir qual programa recebe atenção do processador, gerenciar memória, criar um sistema de arquivos etc.), fornecendo uma interface entre o computador e o usuário. Embora possa ser executado imediatamente após a máquina ser ligada, a maioria dos computadores pessoais de hoje o executa através de outro programa armazenado em uma memória não-volátil do tipo ROM (Read Only Memory) chamado BIOS (Basic Input Output System) num processo chamado bootstrapping, conceito em inglês usado para designar processos autossustentáveis, ou seja, capazes de prosseguirem sem ajuda externa. Após executar testes e iniciar os componentes da máquina (monitores, discos, etc), o BIOS procura pelo sistema operacional em alguma unidade de armazenamento, geralmente o Disco Rígido, e a partir daí, o sistema operacional toma o controle da máquina. O sistema operacional reveza sua execução com a de outros programas, como se estivesse vigiando, controlando e orquestrando todo o processo computacional.

Sistemas operacionais de tempo real ou RTOS (Real Time Operating Systems) são uma categoria especial de sistemas operacionais. Eles são voltados para aplicações onde é essencial a confiabilidade e a execução de tarefas em prazos compatíveis com a ocorrência de eventos externos.

Para a aplicação proposta o sistema precisa medir a grandeza analógica, armazenar em um buffer, esperar uma ocorrência externa (trigger) e então enviar os dados em tempo real para o software que irá gerir os dados convertidos para grandezas digitais.

1. **Situação atual (estado-da-arte)**

Atualmente, existem poucos projetos e repositórios que tratam de aquisição de sinais com Arduino, e muitos projetos de comunicação Wifi e servidores HTTP embarcados, portanto a ideia é unir os dois mundos e criar um projeto completo de aquisição de sinais e envio de dados Wifi.

Existem alguns equipamentos mais baratos que um osciloscópio profissional, porém, não de fabricação brasileira, não open source, desta forma este produto seria uma inovação de mercado.

8.2Tabela de Requisitos

Tabela 1 de requisitos não funcionais.

|  |  |  |
| --- | --- | --- |
| **REQUISITOS ORGANIZACIONAIS** | | |
| **Cd.** | **Nome** | **Tipo** |
| RNF01 | Aplicação web. | Implementação. |
| RNF02 | Programação orientada a objetos. | Padrão. |
| **REQUISITOS DE PRODUTOS** | | |
| **Cd.** | **Nome** | **Tipo** |
| RNF03 | Documentação para ajuda. | Usabilidade. |
| RNF04 | Disponibilidade de uso em 98% do tempo. | Confiabilidade. |
| RNF05 | Adaptação para diversas plataformas. | Portabilidade. |
| RNF06 | Customização de layout. | Usabilidade. |
| RNF07 | Cores diferentes para variadas oscilações. | Usabilidade. |
| RNF08 | Opção de zoom. | Usabilidade. |
| **REQUISITOS EXTERNOS** | | |
| **Cd.** | **Nome** | **Tipo** |
| RNF09 | Opção para cadastro de usuários com senha. | Segurança. |

1. **Glossário**

Arduino, ESP32, Osciloscópio, Digital, Servido HTTP, Programação WEB, Sistemas Embarcados, Sistemas Operacionais de Tempo Real, Responsividade, Wireless, Conversão de Sinais Analógicos, Eletrônica.