**CENTRO PAULA SOUZA**

**RUBENS DE FARIA E SOUZA**

**Eletrônica**

**Antonio Zanini de Farias**

**Caio César Corrá**

**Gustavo Teixeira Morais**

**Hebert Prestes Lima**

**CONTROLE DE ALIMENTAÇÃO ANIMAL**

**SOROCABA**

**2020**

**Antonio Zanini de Farias**

**Caio César Corrá**

**Gustavo Teixeira Morais**

**Hebert Prestes Lima**

**CONTROLE DE ALIMENTAÇÃO ANIMAL**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Eletrônica da Etec Rubens de Faria e Souza, orientado pelos Professores Carlos Lourenço Magoga e Fausto de Barros, como requisito parcial para obtenção do título de técnico em Eletrônica.

**SOROCABA**

**2020**

BANCA DE AVALIAÇÃO

Examinador 1: Nome: Assinatura:

Titulação:

Examinador 2: Nome: Assinatura:

Titulação:

Examinador 3: Nome: Assinatura:

Titulação:

Data: Menção:

RESUMO

O projeto descrito nesse trabalho de conclusão de curso foi idealizado para atender à proposta de criar um dispositivo que venha a facilitar a vida das pessoas, para isso, escolhemos o desenvolvimento de um alimentador de animais automatizado.

Um dispositivo como esse pode ser de grande valia para uso de pessoas com animais domésticos, assim como, pessoas que possuem grandes criações de animais. Tendo isso em vista, utilizamos uma metodologia de pesquisa-ação para solucionar este desafio.

Como consequência deste projeto, conseguimos unir vários tipos de tecnologias distintas para construir a nossa solução planejada, nos levando a adquirir valiosos conhecimentos na área de estudo.

Palavras-Chave: Alimentação. Animais. Criação de Animais. Pessoas.

ABSTRACT

The project described in this course conclusion work was idealized to meet the proposal to create a device that will facilitate people's lives, for this, we chose the development of an automated animal feeder.

A device like this can be of great value for use by people with pets, as well as people who have large livestock. With this in mind, we use an action research methodology to solve this challenge.

As a result of this project, we managed to unite several types of different technologies to build our planned solution, leading us to acquire valuable knowledge in the study area.

Keywords: Feeder. Livestock. People. Pets.

LISTA DE ILUSTAÇÕES

[Figura 1. Exemplo de programa Node-RED, uma aplicação de alerta climático. 14](#_Toc55046341)

[Figura 2. Barramento I2C. 15](#_Toc55046342)

[Figura 3 Circuito Regulador de Tensão para 3.3V. 17](#_Toc55046343)

[Figura 4. Motor DC GA12-N20-200. 18](#_Toc55046344)

[Figura 5. Módulo Relé 5V 1 Canal 19](#_Toc55046345)

[Figura 6 - Fluxograma do Processo de Alimentação 23](#_Toc55046346)

[Figura 7 - Fluxograma da Manutenção de Tempo 26](#_Toc55046347)

[Figura 8. Interface de Reservatório 27](#_Toc55046348)

[Figura 9. Fluxo da interface de reservatório 28](#_Toc55046349)

LISTA DE ABREVIATURAS E SIGLAS

Direct Current (DC)

Inter-Integrated Circuit (I2C)

Internet of Things (IoT)

Internet Protocol (IP)

Liquid Crystal Display (LCD)

Megabit (Mbit)

Message Queue Telemetry Transport (MQTT)

Microfarad (µF)

Quilobit (Kbit)

Rotações por Minuto (RPM)

Secure Sockets Layer (SSL)

Service Set Identifier (SSID)

Transmission Control Protocol (TCP)

Transport Layer Security (TSL)

Nanofarad (nF)

SUMÁRIO

[1 INTRODUÇÃO 10](#_Toc55046460)

[2 DESENVOLVIMENTO 11](#_Toc55046461)

[2.1 Fundamentação Teórica 11](#_Toc55046462)

[2.1.1 IOT 11](#_Toc55046463)

[2.1.1.1 IoT Industrial 12](#_Toc55046464)

[2.1.2 WI-FI 12](#_Toc55046465)

[2.1.3 MQTT 12](#_Toc55046466)

[2.1.3.1 Características 13](#_Toc55046467)

[2.1.3.2 Funcionamento do MQTT 13](#_Toc55046468)

[2.1.4 Node-RED 14](#_Toc55046469)

[2.1.5 I2C 15](#_Toc55046470)

[2.2 Projeto 16](#_Toc55046471)

[2.2.1 Hardware 16](#_Toc55046472)

[2.2.1.1 Circuito de Alimentação 16](#_Toc55046473)

[2.2.1.1.1 Alimentação Chaveada 17](#_Toc55046474)

[2.2.1.1.2 Circuito Regulador de Tensão 17](#_Toc55046475)

[2.2.1.2 Abastecimento 18](#_Toc55046476)

[2.2.1.2.1 Motor de Abastecimento 18](#_Toc55046477)

[2.2.1.2.2 Relé de Acionamento do Motor 19](#_Toc55046478)

[2.2.1.2.3 Rosca helicoidal 19](#_Toc55046479)

[2.2.1.3 Sensores e Atuadores Secundários 20](#_Toc55046480)

[2.2.1.3.1 Sensores Abastecimento 20](#_Toc55046481)

[2.2.1.3.1.1 Características 20](#_Toc55046482)

[2.2.1.3.2 Display informativo 20](#_Toc55046483)

[2.2.1.3.2.1 Características 21](#_Toc55046484)

[2.2.1.3.3 Módulo I2C 21](#_Toc55046485)

[2.2.1.3.3.1 Características 21](#_Toc55046486)

[2.2.1.3.4 Buzzer Sinalizador 21](#_Toc55046487)

[2.2.1.3.4.1 Características 22](#_Toc55046488)

[2.2.2 Software 22](#_Toc55046489)

[2.2.2.1 Linguagem C (ESP32) 22](#_Toc55046490)

[2.2.2.1.1 Alimentador 22](#_Toc55046491)

[2.2.2.1.2 WI-FI 24](#_Toc55046492)

[2.2.2.1.3 MQTT 24](#_Toc55046493)

[2.2.2.1.4 Horário 25](#_Toc55046494)

[2.2.2.1.5 Buzzer 26](#_Toc55046495)

[2.2.2.2 Node-RED 27](#_Toc55046496)

[2.2.2.2.1 Interface de Reservatório 27](#_Toc55046497)

[3 CONCLUSÃO 30](#_Toc55046498)

[REFERÊNCIAS 31](#_Toc55046499)

[ANEXO 1 – Programa em linguagem C (ESP32) 33](#_Toc55046500)

[ANEXO 2 – Programa da Interface Node-RED 34](#_Toc55046501)

1. INTRODUÇÃO

Este trabalho visa experimentar e solidificar conceitos de eletrônica aplicada a automatização de tarefas, circuitos eletrônicos são a base de todos os dispositivos de automação modernos e ao projetar uma nova solução se torna natural o estudo deles.

Este projeto desenvolve o uso de uma placa de prototipagem, que manipula os sinais elétricos de entradas para acionar um motor que abastecerá, de forma programável, o recipiente de um animal.

Uma placa microcontroladora baseada em arduino será responsável por controlar a alimentação, acionando o motor que impulsiona a ração no reservatório em direção à saída alimento. O intervalo de tempo em que este processo ocorre é definido pelas configurações inseridas no painel de entrada e armazenadas na memória do arduino, e utilizará um módulo RTC para o controle de horário. Quando o reservatório estiver em desabastecido, um LED de alerta será ativado, e uma mensagem será exibida no display do painel. O dispositivo também contará com uma interface wireless para a entrada de configurações, sendo disponíveis através da conexão do dispositivo a uma rede e a utilização de um navegador para a exibição e edição de informações

1. DESENVOLVIMENTO

t.

* 1. Fundamentação Teórica

t.

* + 1. IOT

IoT (*Internet of Things*), ou Internet das Coisas em português, é um conceito que envolve utilização de uma rede que conecta pequenos computadores, dispositivos, sensores e atuadores com o intuito de compartilhar e manipular dados para a criação de processos automatizados e monitoramento de atividades.

Apesar de tecnologias que compõe o IoT já existirem há décadas, a grande expansão recente do IoT se deve a alguns fatores:

* Popularização de sensores de baixo custo e baixa potência;
* Desenvolvimento da internet com protocolos desenvolvidos para comunicação de pequenos dispositivos;
* Armazenamento de dados na nuvem e processamento/análise de dados avançados através de *Machine Learning*; e
* Avanço de tecnologias baseada em processamento de linguagem natural.
  + - 1. IoT Industrial

Aplicado a indústria, o IoT assume o papel de dar suporte à instrumentação, ao controle de sensores com interação com tecnologias da nuvem e comunicação máquina a máquina. As seguintes áreas são as principais aplicações do IoT Industrial:

* Cidades inteligentes;
* Manufatura Inteligente;
* Cadeia de suprimentos digitais inteligentes;
* Manutenção preventiva e preditiva;
* Logística conectada e inteligente; e
* Redes elétricas inteligentes.
  + 1. WI-FI

Wi-Fi é uma tecnologia de comunicação de rede sem fio baseada em transmissões de radiofrequência na faixa entre 2.4Ghz e 5Ghz.

Os padrões da família IEEE 802.11 definem os protocolos que envolvem a conexão entre computadores, dispositivos móveis e outros equipamentos. Cada padrão é estabelecido e ratificado, sucedendo o padrão vigente e estabelecendo uma versão com mais funcionalidades.

* + 1. MQTT

Este protocolo foi criado pela empresa IBM, no final dos anos 90, como o objetivo vincular suas estruturas de sensores em oleodutos prolíferos à rede de satélites. Como o desenvolvimento do IoT, o protocolo MQTT se tornou o protocolo mais utilizado na comunicação entre pequenos dispositivos.

* + - 1. Características

O MQTT possui a seguintes características que o qualificam como uma excelente opção de comunicação entre sistemas embarcados:

* Protocolo leve: Este protocolo foi desenvolvido para suportar bom desempenho hardware com capacidades limitadas, larguras de banda pequenas e transmissões de alta latência;
* Flexibilidade: O modo de transmissão e a organização de tópicos confere uma alta maleabilidade sobre qual a informação será trafegada o como ela o será;
* Suporte à Segurança: Apesar de leve e flexível, o MQTT ainda é capaz de fornecer funcionalidades de segurança como autenticação, restrição a conteúdos e suporte a conexões seguras (SSL/TSL).
  + - 1. Funcionamento do MQTT

O MQTT funcionará através da rede sem fio, sendo executado em cima do protocolo TCP/IP. Em sua organização, este protocolo possui duas entidades (broker e cliente), dois papéis (assinante e publicador) e uma estrutura de organização de mensagens (tópicos).

O **broker**, é um centralizador e redistribuidor de mensagens, mantendo registro dos destinatários das mensagens e redirecionando as mensagens tão logo elas são recebidas. Apenas um dispositivo pode ser o broker de uma conexão MQTT.

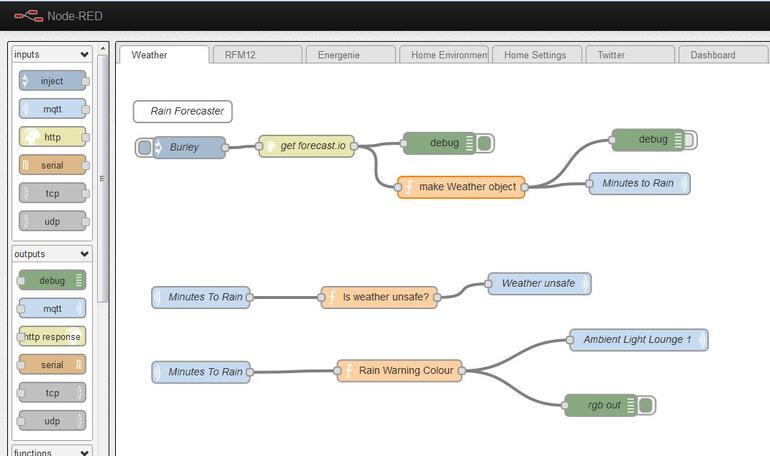
O **cliente** é qualquer dispositivo que mande ou receba mensagens através do broker. Quando está enviando, o cliente está desempenhando o papel de **publicador**, onde ele enviar uma mensagem ao broker para ser aplicada em um determinado tópico de mensagens. Ao receber mensagens, o cliente está no papel do **assinante**, onde ele define, geralmente no momento do estabelecimento de sua conexão MQTT, os tópicos de mensagens aos quais as publicações devem ser redirecionadas a ele. Múltiplos dispositivos podem ser clientes, publicadores e assinantes em uma conexão MQTT, além disso, o cliente pode desempenhar os papeis de Publisher e Subscriber simultaneamente.

Os **tópicos** são uma estrutura de armazenamento de mensagens primariamente temporário, para reter a mensagem apenas durante o período entre o broker receber a mensagem do publicador e o momento em que terminar a entrega para o último assinante de seu tópico.

* + 1. Node-RED

Node-RED é uma ferramenta de desenvolvimento de software criada pela IBM, sua operação é dada através da programação gráfica e organizada em nós e fluxos de dados. Esta tecnologia foi construída em cima de javascript e NodeJS, sendo possível criar trechos de programação textual em javascript dentro de alguns nós e gerenciar pacotes de bibliotecas através do NodeJS.

Figura 1. Exemplo de programa Node-RED, uma aplicação de alerta climático.



Fonte: TechRepublic, 2014.

Um dos grandes poderes do Node-RED é a possibilidade de adicionar novas bibliotecas de nós, e se integrar ao mais diversos tipos de dispositivos ou serviços da internet, sendo uma ferramenta muito poderosa em aplicações IoT que dependem de recursos da nuvem.

* + 1. I2C

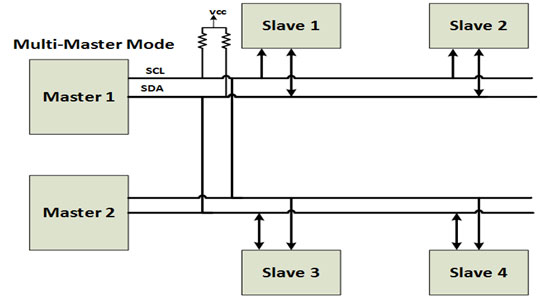
O I2C (Inter-integrated circuit) é um protocolo de comunicação serial que é utilizado para simplificar a transmissão de dados entre um dispositivo principal e seus periféricos.

A disposição de um circuito I2C é dada através de um barramento duplo que possui as vias SCL (Serial Clock Line) e SDA (Serial Data Line). Os dispositivos podem ser inseridos no circuito com o papel de mestre ou escravo.

O mestre em um circuito I2C não recebe um endereçamento no barramento e, além disso, é responsável pelo início de cada comunicação, dirigindo requisições de mensagens, assim como, definindo o clock para estas.

O escravo tem o papel de receber e mandar mensagens aos mestres. Cada escravo deve ser endereçado de forma única no barramento, dessa forma será identificado nas requisições no mestre.

Figura 2. Barramento I2C.



Fonte: Engineers Garage, 2020.

A taxa de transmissão do protocolo varia conforme o modo de operação implementado no circuito I2C em questão. Os modos de operação são os seguintes:

Tabela 1. Modos de Operação I2C

|  |  |
| --- | --- |
| **Modo de Operação** | **Taxa de Transmissão** |
| Standard Mode | 100 Kbit/s |
| Fast Mode | 400 Kbit/s |
| Fast Mode Plus | 1 Mbit/s |
| High-Speed Mode | 3,4 Mbit/s |

Fonte: Engineers Garage, 2020.

Os modos de operação mais utilizados são Standard Mode e Fast Mode.

* 1. Projeto

**t.**

* + 1. Hardware

**t.**

* + - 1. Circuito de Alimentação

O projeto tem como fator de grande importância o seu circuito de alimentação, pois o funcionamento do equipamento na proposta de suprir as necessidades alimentares de um animal não pode ser cerceado por um mal funcionamento momentâneo na rede elétrica. Para este fim foi desenvolvido um circuito de alimentação que mantém o dispositivo funcionando por intermédio de uma bateria quando há falta de energia elétrica na rede de distribuição.

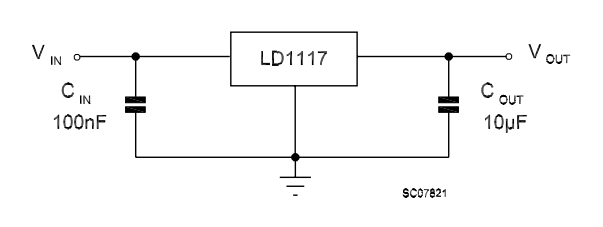
Além disso, o circuito de alimentação é responsável por regular a tensão do circuito (12V), para aquela que supre as necessidades do microcontrolador (3,3V).

* + - * 1. Alimentação Chaveada
* Fonte 12V/1,5A;
* Relé 12V; e
* Bateria.

A fonte é utilizada para alimentação geral do circuito, enquanto a bateria será acionada em caso de uma queda de energia para que o equipamento não fique inutilizável e o animal fique impossibilitado de alimentar-se. Para que isso aconteça, ambos conectados à um relé normalmente fechado com a fonte,

* + - * 1. Circuito Regulador de Tensão

Figura 3 Circuito Regulador de Tensão para 3.3V.



Fonte: STMicroelectronics, 2005.

* Regulador de Tensão 3.3V LD1117;
* Capacitor cerâmico 10µF; e
* Capacitor cerâmico 100nF.

Após o relé da fonte, em paralelo com relé do alimentador, se encontra o circuito regulador de tensão (Figura 3). Composto pelo LD1117 juntamente com dois capacitores (100nF e 10µF), ele realiza a conversão da tensão de alimentação do circuito de entrada para alimentar o microcontrolador com a tensão de 3,3V.

* + - 1. Abastecimento

Para poder alimentar o PET, deve-se transportar a ração do armazenador até a tigela. Com isso em mente, utilizamos uma rosca helicoidal que transportará o alimento ao girar em seu próprio eixo. O movimento da rosca é proporcionado por um motor DC que, por sua vez, é acionado por um relé, conectado diretamente ao microcontrolador.

* + - * 1. Motor de Abastecimento

Figura 4. Motor DC GA12-N20-200.



Fonte: Handson Technology, 2019.

O motor escolhido é o Micro Motor DC N20 com Caixa de Redução 12V 200 RPM, modelo GA12-N20-200. Este motor é equipado com caixa de redução, com taxa de Redução: 1:10, composta por engrenagens de metal que aumentam o torque e mantém sua rotação estável. Tem uma faixa de operação entre 6V a 12V DC e corrente de 0.07A, tem torque máximo de 2 kgf.cm e, quando carregado, apresenta uma rotação máxima de 80 RPM

* + - * 1. Relé de Acionamento do Motor

Para o acionamento do motor foi empregado um módulo relé com bobina de 5V DC e 75mA, e com capacidade de até 12A em 125V AC ou 7A em 250VAC. Em seu contato normalmente aberto será conectado o positivo da bateria 12V e o terminal positivo de alimentação de motor, de modo que, quando o ESP32 enviar o sinal de ativação a bobina será energizada, o contato se fechará e o motor será acionado.

Figura 5. Módulo Relé 5V 1 Canal

Uma imagem contendo eletrônico, circuito

Descrição gerada automaticamente

Fonte: OLIVEIRA Euler, 2018.

* + - * 1. Rosca helicoidal

A rosca helicoidal é um mecanismo mecânico que transforma movimento rotacional em movimento linear. O movimento de rotação do helicoide faz com que um ponto de material, depositado no fundo da carcaça, seja arrastado para frente. Após ser deslocado, o movimento angular fará arrastar linearmente de onde se encontrava inicialmente até o próximo ponto. Se continuar rotacionando o helicoide, o ponto seguinte do helicoide fará o papel de empurrar linearmente o material e isso se repetirá até chegar a um local que o material não esteja mais em contato com o fundo da Rosca transportadora e com o anel do helicoide, ou seja, no bocal de saída do equipamento.

A rosca necessária para o projeto se estende por 500 mm e é feita a partir de uma placa de MDF com 3 mm de espessura cortada num formato afim de se tornar flexível e possibilitar que seja fixada num cano de PVC de 25 mm de diâmetro que assume a função de eixo. Para sua realização serão necessárias 5 peças de MDF que se encaixam ao longo do eixo.

**<<ADICIONAR IMAGENS DA ROSCA (+- 3 IMAGENS) >>**

* + - 1. Sensores e Atuadores Secundários
         1. Sensores Abastecimento

Utilizaremos dois Sensores Capacitivos de Proximidade que serão responsáveis por nos informar a quantidade de mantimentos contidos no reservatório no qual se localiza o alimento do animal desejado. Um dos sensores ficará no meio do reservatório e o outro próximo ao fundo.

Características

Sensor Capacitivo de Proximidade do tipo PNP é um dispositivo eletrônico capaz de detectar a aproximação / presença de objetos metálicos (ferro, aço, alumínio e outros) e objetos não metálicos.

* + - * 1. Display informativo

Um Display LCD 16x2 será o responsável por nos informar as situações presentes no reservatório referente ao estado dos sensores.

Características

Display LCD 16×2 Backlight Verde possui 2 linhas com 16 colunas e controlador HD44780.

* + - * 1. Módulo I2C

Um Circuito Integrado responsável pela comunicação entre a ESP32 e o display de LCD, pois o ESP32 possui um número limitado de portas e não há a necessidade de usar todas.

Características

O CI PCF8574 é um expansor de portas I2C de 8 bits que permite o controle de até 8 portas utilizando apenas 2 pinos do microcontrolador. Funciona com tensões entre 2.5 e 6V, e os pinos A0, A1 e A2 definem o endereço utilizado para comunicação I2C. As 8 portas podem ser configuradas como entrada ou saída e os pinos da interface I2C (SDA e SCL) necessitam de resistores pull-up.

* + - * 1. Buzzer Sinalizador

Um Buzzer Passivo será acionado quando o reservatório estiver quase vazio, ou seja, quando ambos os sensores estiverem enviando sinal baixo para o ESP32. Irá emitir sons sincronizados em forma de uma música.

Características

O buzzer 5V passivo permite que você tenha mais controle sobre a melodia, já que as formas de onda dos sons são enviadas diretamente para o pino de I/O, resultando em um som mais limpo.

* + 1. Software

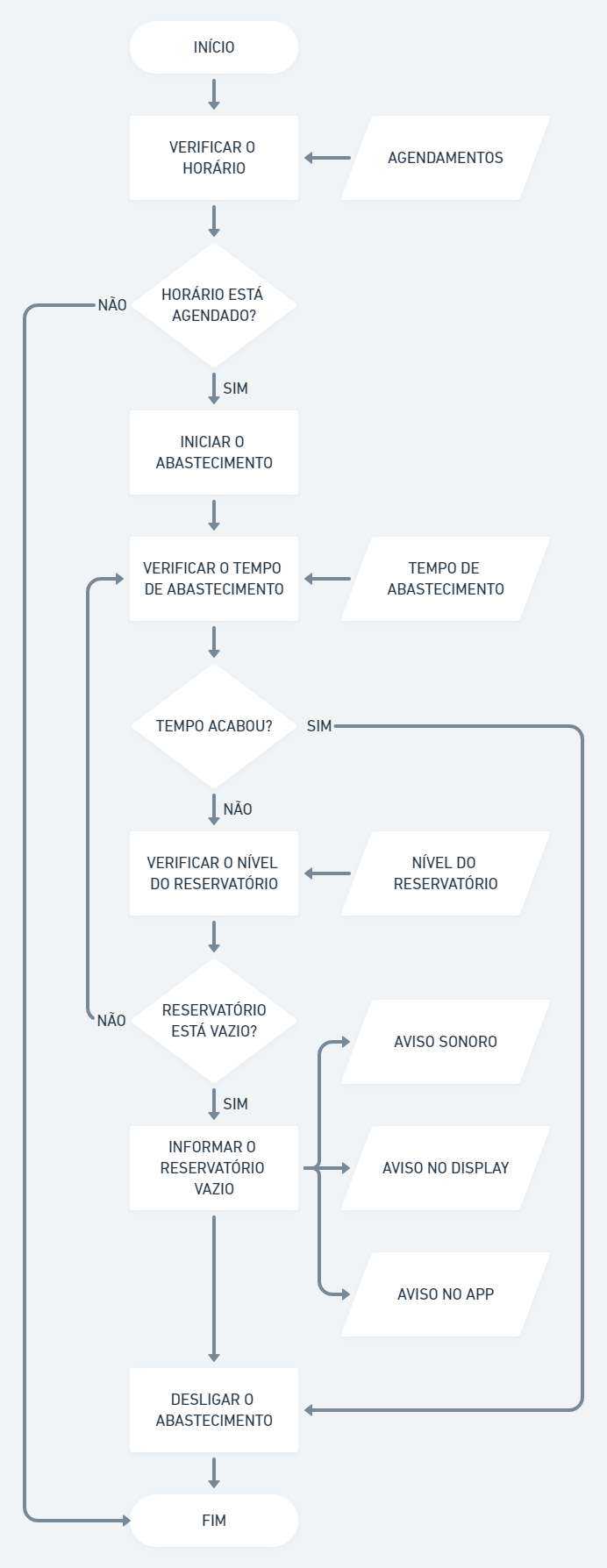
O software de nosso projeto possui várias camadas de aplicação, onde o principal processamento é concentrado no ESP32, implementado através da linguagem C, porém, a comunicação MQTT e a plataforma de Node-RED tornam-se partes essenciais por serem responsáveis pela interação e configuração do sistema.

* + - 1. Linguagem C (ESP32)
         1. Alimentador

Este é o módulo do programa responsável pela ativação do relé de alimentação, o processo principal do projeto. Para executar com sucesso esta função, o módulo de alimentação interage, direta ou indiretamente, com todos os outros módulos do projeto.

Seu processo, como descrito no fluxograma do processo de alimentação (Figura 6), realiza uma verificação nos horários agendados e, caso o horário atual esteja agendado, ele ativa o relé de abastecimento durante o período programado. O processo ainda terá o cuidado de não ativar o relé caso os sensores indiquem que o reservatório está vazio, assim interrompendo a alimentação ou nem mesmo a iniciando e, em vez disso, enviando alertas nos meios disponíveis.

Figura 6 - Fluxograma do Processo de Alimentação



Fonte: Dos próprios autores, 2020.

* + - * 1. WI-FI

Para a utilização de WI-FI no ESP32 utilizamos a biblioteca WiFi.h, ela traz funcionalidades que permitem o manuseio simplificado de uma conexão de rede sem fio.

Na implementação se faz necessária a definição do identificador da rede sem fio (SSID) e da sua respectiva senha de acesso, tendo estas informações um comando de inicialização da conexão WI-FI deve ser executado até a conexão seja estabelecida.

Uma vez tendo a conexão estabelecida, o ESP32 terá que verificar periodicamente o estado da conexão para se informar se a conexão ainda está ativa ou foi perdida e, nesse caso, reiniciar o processo de reconexão.

* + - * 1. MQTT

Como o protocolo principal de comunicação entre os clientes WI-FI, o projeto utilizará o protocolo MQTT.

Utilizamos um broker Eclipse Mosquitto com configurações iniciais instalado em rede local, como um projeto conceitual esta configuração será efetiva para os propósitos desejados. Entretanto, o broker poderia ser utilizado em um serviço hospedado em algum servidor web, adicionando a possibilidade de utilizar as funcionalidades de o projeto de forma remota através da internet.

Na utilização do MQTT pelo ESP32 precisamos da conexão WI-FI, previamente definida, assim como uma biblioteca para fornecer-nos o conjunto de comandos para sua manipulação, nesse caso, utilizaremos a biblioteca PubSubClient.h.

Para estabelecer a conexão MQTT, é necessário o endereço IP (ou web) do computador do broker e a porta configurada para o MQTT. Após realizada a conexão, deve-se configurar nesta o método de comunicação (callback), assim o ESP32 poderá receber as mensagens publicadas nos tópicos aos quais assinar.

Assim como uma conexão WI-FI, os comandos da conexão MQTT devem ser iterados até a conexão seja efetivamente estabelecida e deve ser monitorada periodicamente para tratar possíveis falhas de conexão.

Sempre após realizar a conexão (ou reconexão) o ESP32 deverá assinar os tópicos aos quais deseja receber as mensagens do broker, esses tópicos são:

* “alimentador/horariosAlimentacao”;
* “alimentador/tempoAbastecimento”;
* “alimentador/nivelReservatorio”;
* “alimentador/tipoAlertaSonoro”;

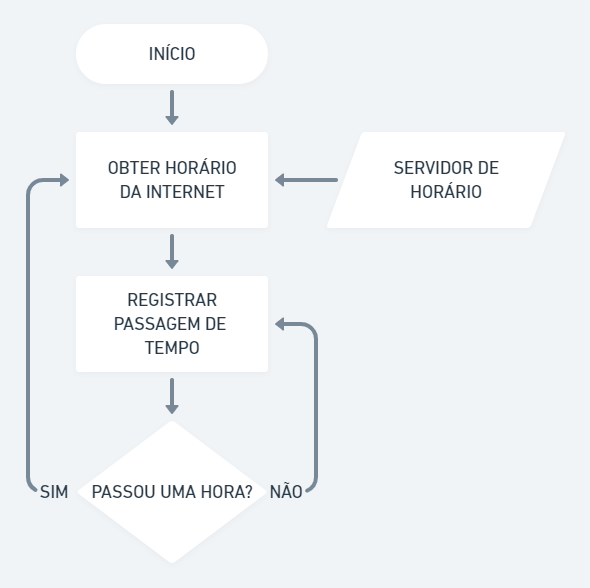
Nos casos de publicação, o ESP32 apenas terá de enviar para o broker uma mensagem contendo o tópico no qual ele deseja publicar e o conteúdo publicado.

* + - * 1. Horário

Ao não possui um hardware RTC adicionado ao nosso projeto, se tornou necessária a adição de um módulo do programa direcionado à manutenção de um relógio interno.

A biblioteca NTPClient foi a base para o desenvolvimento, como a interface principal do projeto depende de uma conexão Wi-Fi, não se torna excessivo o requisito de conexão à internet. Porém, como precaução para que eventuais faltas de internet não atrapalhem o funcionamento do sistema, o processo que mantém o horário apenas busca a hora atual na internet a cada uma hora passada, nos outros momentos o ESP32 calcula a passagem de tempo de forma a atualizar o relógio programado a cada segundo. Processo descrito no fluxograma de manutenção de tempo (Figura 7).

Figura 7 - Fluxograma da Manutenção de Tempo



Fonte: Dos próprios autores, 2020.

Devido a precisão demandada por esta funcionalidade, ela foi separada das outras, sendo processada em modo exclusivo pelo núcleo de processamento zero do ESP32, enquanto os outros processos permanecem sendo executados em no núcleo de processamento um.

* + - * 1. Buzzer

O projeto possui um módulo do programa dedicado à execução de alerta sonoros, estes que serão executados pelo buzzer passivo.

Os alertas são programados para serem executados em momentos nos quais o microcontrolador detecta a falta de algum recurso essencial para o funcionamento pleno do sistema, sendo eles: conteúdo no reservatório e acesso à internet.

O programa envia uma variação de frequências durante intervalos de tempo predeterminados de forma a executar uma melodia que servirá como o alerta para o usuário do sistema voltar suas atenções ao elemento faltante.

A implementação disso no ESP32 utiliza a biblioteca EasyBuzzer, a qual disponibiliza uma série de facilitadores para a execução de notas musicais sequenciais. A principal característica utilizada desta biblioteca é a função singleBeep, esta função permite a execução deu uma frequência durante um tempo determinado e, quando o tempo acabar, esta mesma função chama uma função para executar as próximas notas.

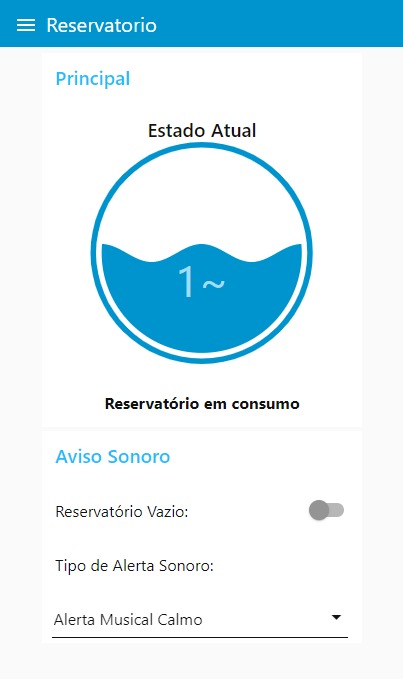
* + - 1. Node-RED

A ferramenta Node-RED está sendo aplicada como a principal responsável pelo gerenciamento de configurações do projeto. Ela implementa interfaces web que, ao interagir como broker mqtt, exibe as configurações atuais do microcontrolador, assim como, disponibiliza novas opções de configurações.

Para a criação da interface web, foi adicionado ao projeto básico do Node-RED os pacotes de ferramentas: “node-red-dashboard” e “node-red-node-ui-list”. Esses pacotes concedem nós para serem utilizados no fluxo e geram uma interface HTML com variadas opções de controles e componentes visuais.

* + - * 1. Interface de Reservatório

Figura 8. Interface de Reservatório

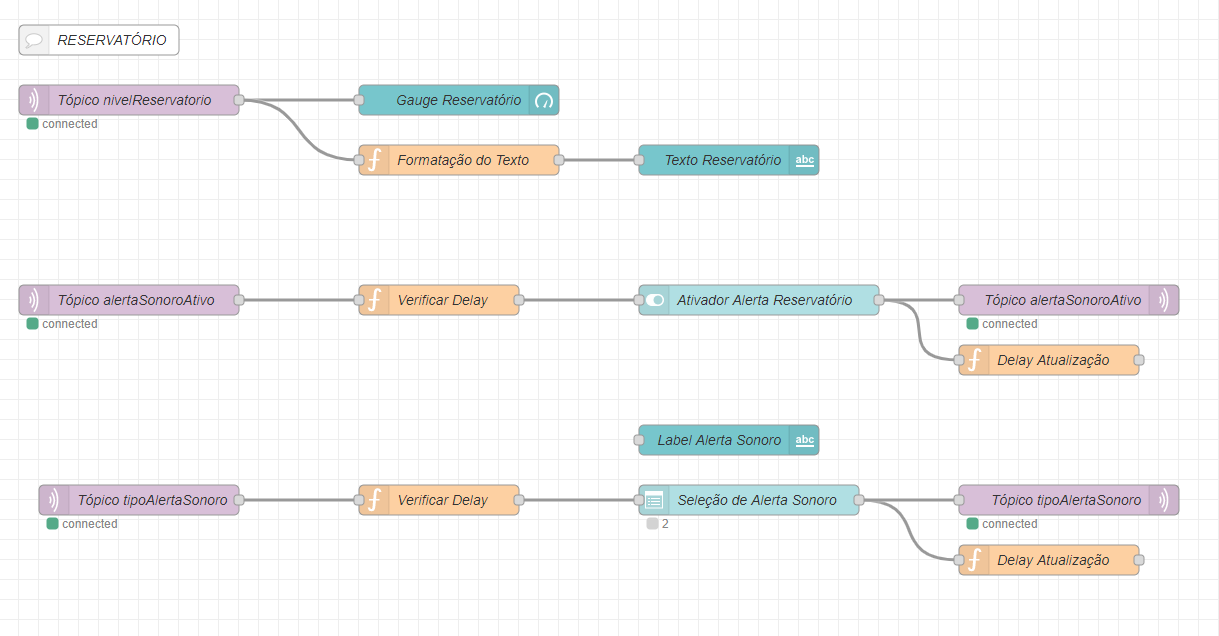


Fonte: Dos próprios autores, 2020.

Nesta tela temos duas seções, a principal e a avisos sonoros.

Na seção principal utilizamos um nó do tipo gauge que lê a quantidade de sensores de alimentação ativos e exibe uma representação gráfica do estado do reservatório, de modo a deixar informação intuitiva do ponto do usuário.

Figura 9. Fluxo da interface de reservatório



Fonte: Dos próprios autores, 2020.

Na seção aviso sonoro possuímos um nó do tipo switch para definir se o alimentador irá executar avisos sonoros quando for impedido de executar uma alimentação por causa do reservatório estar vazio. Logo abaixo, temos um nó do tipo dropdown, que exibirá as opções disponíveis de alerta sonoro a serem executados pelo alimentador.

.

1. CONCLUSÃO

t.

REFERÊNCIAS

BERTOLETI, Pedro. ESP32 e MQTT DASH: controle e monitoramento através de um dashboard MQTT para Android. **Filipeflop**, 2019. Disponível em: https://www.filipeflop.com/blog/esp32-e-mqtt-dashboard-android/. Acesso em: 05 out. 2020.

BHATT, Ashutosh. Understanding the I2C Protocol. **Engineers Garage**, 2020. Disponível em: https://www.engineersgarage.com/tutorials/understanding-the-i2c-protocol/. Acesso em: 08 out. 2020.

CANOVA, Henrique. Sensores Capacitivos. **Mecaweb Education SITE**, 2014. disponível em: <http://www.mecaweb.com.br/eletronica/content/e_sensor_capacitivo>. Acesso em: 12 out. 2020.

ESP32 Series Datasheet. **Espressif**, 2020. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf. Acesso em: 05 out. 2020.

GA12-N20 Geared Mini DC Motor Datasheet. **Handson Technology**, 2019. Disponível em: https://www.handsontec.com/dataspecs/motor\_fan/GA12-N20.pdf. Acesso em: 12 out. 2020.

HEATH, Nick. **TechRepublic**, 2014. Disponível em: https://www.techrepublic.com/article/node-red/. Acesso em: 08 out. 2020.

LIMA, Izabelle. Aprenda a Reproduzir Músicas com o Módulo Buzzer. **Auto Core Robótica blog**, 2018. Disponível em: https://autocorerobotica.blog.br/aprenda-a-reproduzir-musicas-com-o-modulo-buzzer-2/. Acesso em: 05 out. 2020.

MOREIRA, Diego. Módulo I2C para LCD 16x2 e Arduino. **Auto Core Robótica blog**, 2019. Disponível em: <https://autocorerobotica.blog.br/modulo-i2c-para-lcd-16x2-e-arduino/>. Acesso em: 12 out. 2020.

O'LEARY Nick. ARDUINO Client for MQTT. **Knolleary**, 2020. Disponível em: https://pubsubclient.knolleary.net. Acesso em: 05 out. 2020.

OLIVEIRA, Euler. Como usar com Arduino – Módulo Relé 5V 1 Canal. **Blog Masterwalker Shop**, 2018. Disponível em: https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-modulo-rele-5v-1-canal/. Acesso em: 12 out. 2020.

ORACLE, **ORACLE**, 2020. Disponível em: https://www.oracle.com/br/internet-of-things/what-is-iot.html. Acesso em: 07 out. 2020.

Corbari, **Corbari**, 2018. Disponível em: http://www.corbari.com.br/rosca-transportadora. Acesso em: 12 out. 2020.

YUAN, Michael. Conhecendo o MQTT. **IBM**, 2017. Disponível em: https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/. Acesso em: 05 out. 2020.

WIFI library. **Arduino**, 2019. Disponível em: https://www.arduino.cc/en/Reference/WiFi. Acesso em: 05 out. 2020.

LD1117 SERIES Datasheet. **STMicroelectronics**, 2005. Disponível em: https://www.sparkfun.com/datasheets/Components/LD1117V33.pdf. Acesso em: 17 out. 2020.

ANEXO 1 – Programa em linguagem C (ESP32)

ANEXO 2 – Programa da Interface Node-RED