**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 pelo Prof. Magoga, 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 consiste no desenvolvimento de um dispositivo capaz de automatizar a alimentação animal controlando a frequência de abastecimento, que é configurada no mesmo.

Utilizando uma placa de prototipagem da plataforma Arduino para controlar as funções do equipamento.

Para o transporte da ração pela passagem entre o container e o pote de ração foi empregado um motor.

ABSTRACT

This project consist in the development about a device capable of automate an animal feeding by controlling the supply frequency.

Resorting to an Arduino single-board microcontroller to control the equipment features.

A DC motor has the function of transporting the animal food on container down to the food pot.

SUMÁRIO

[**1 INTRODUÇÃO 7**](#_Toc52698709)

[**2 DESENVOLVIMENTO 8**](#_Toc52698710)

[**2.1 Revisão Bibliográfica 8**](#_Toc52698711)

[**2.1.1 IOT 8**](#_Toc52698712)

[**2.1.2 WI-FI 8**](#_Toc52698713)

[**2.1.3 MQTT 8**](#_Toc52698714)

[**2.1.4 NodeRED 8**](#_Toc52698715)

[**2.2 Metodologia 9**](#_Toc52698716)

[**2.3 Projeto 9**](#_Toc52698717)

[**2.3.1 Hardware 9**](#_Toc52698718)

[**2.3.1.1 Sensores e Atuadores Secundários 9**](#_Toc52698719)

[**2.3.1.1.1 Sensores Abastecimento 9**](#_Toc52698720)

[**2.3.1.1.2 Display informativo 9**](#_Toc52698721)

[**2.3.1.1.3 Buzzer Sinalizador 10**](#_Toc52698722)

[**2.3.2 Software 10**](#_Toc52698723)

[**2.3.2.1 Linguagem C (Arduino) 10**](#_Toc52698724)

[**2.3.2.2 NodeRed 10**](#_Toc52698725)

[**3 CONCLUSÃO 11**](#_Toc52698726)

[**REFERÊNCIAS BIBLIOGRÁFICAS 12**](#_Toc52698727)

[**ANEXO 1 – Programa em linguagem C (Arduino) 13**](#_Toc52698728)

1. INTRODUÇÃO

t.

1. DESENVOLVIMENTO

t.

* 1. Revisão Bibliográfica

t.

* + 1. IOT

t.

* + 1. WI-FI

t.

* + 1. MQTT

t.

* + 1. NodeRED

t.

* 1. Metodologia

t.

* 1. Projeto

t.

* + 1. Hardware

t.

* + - 1. Sensores e Atuadores Secundários

t.

* + - * 1. Sensores Abastecimento

t.

* + - * 1. Display informativo

t.

* + - * 1. Buzzer Sinalizador

t.

* + 1. Software

t.

* + - 1. Linguagem C (Arduino)

t.

* + - 1. NodeRed

t.

1. CONCLUSÃO

t.

REFERÊNCIAS BIBLIOGRÁFICAS

t.

ANEXO 1 – Programa em linguagem C (Arduino)

WI-FI

Este módulo do projeto é desenvolvido utilizando a funcionalidade WI-FI embutida no microcontrolador ESP32, a comunicação WI-FI e suas características, ou seja, transmissão e recepção de informações através de radiofrequência utilizando protocolos da família IEEE 802, oferecem os recursos necessários para a criação de uma interface entre o microcontrolador, que será responsável pela coordenação dos sensores e atuadores de forma a implementar a lógica de negócio, e outras tecnologias, como NodeRED, que será responsável por manter e atualizar uma interface entre o usuário e as configuração e informações do microcontrolador.

WI-FI no ESP32

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.

MQTT

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

Este protocolo 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.

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, o broker e o cliente.

O broker, é um tipo de 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 Publisher (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 Subscriber (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, publishers e subscribers em uma conexão MQTT, além disso, o cliente pode desempenhar os papeis de Publisher e Subscriber simultaneamente.

MQTT no ESP32

Neste projeto 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 seriam: “alimentador/horariosAlimentacao” e “alimentador/tempoAlimentacao”.

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.