**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**](#_Toc52700671)

[**2 DESENVOLVIMENTO 8**](#_Toc52700672)

[**2.1 Revisão Bibliográfica 8**](#_Toc52700673)

[**2.1.1 IOT 8**](#_Toc52700674)

[**2.1.2 WI-FI 8**](#_Toc52700675)

[**2.1.3 MQTT 8**](#_Toc52700676)

[**2.1.4 NodeRED 8**](#_Toc52700677)

[**2.1.5 I2C 9**](#_Toc52700678)

[**2.2 Metodologia 9**](#_Toc52700679)

[**2.3 Projeto 9**](#_Toc52700680)

[**2.3.1 Hardware 9**](#_Toc52700681)

[**2.3.1.1 Sensores e Atuadores Secundários 9**](#_Toc52700682)

[**2.3.1.1.1 Sensores Abastecimento 9**](#_Toc52700683)

[**2.3.1.1.2 Display informativo 10**](#_Toc52700684)

[**2.3.1.1.3 Buzzer Sinalizador 10**](#_Toc52700685)

[**2.3.2 Software 10**](#_Toc52700686)

[**2.3.2.1 Linguagem C (Arduino) 10**](#_Toc52700687)

[**2.3.2.2 NodeRed 10**](#_Toc52700688)

[**3 CONCLUSÃO 11**](#_Toc52700689)

[**REFERÊNCIAS BIBLIOGRÁFICAS 12**](#_Toc52700690)

[**ANEXO 1 – Programa em linguagem C (Arduino) 13**](#_Toc52700691)

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. Revisão Bibliográfica

t.

* + 1. IOT

t.

* + 1. WI-FI

t.

* + 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. NodeRED

t.

* + 1. I2C

t.

* 1. Metodologia

t.

* 1. Projeto

t.

* + 1. Hardware

t.

* + - 1. Abastecimento

Para transportar a ração do armazenador para a tigela do pet foi empregado uma rosca helicoidal que, ao rotacionar em seu eixo, movimenta a ração até a saída, estrategicamente posicionada.

O giro da rosca helicoidal é proporcionado por um motor DC 12 V, corrente de até 0,07 A, com caixa de redução 1:10, 200 RPM e torque de 2 kgf.cm. O motor entra em ação quando o Arduino, no momento certo, envia um sinal de 5 V, que energiza a bobina de um módulo relé de 1 Canal 5 V, com capacidade de 30 V DC e 10A ou 250V AC e 10A, chaveando o contato NA com 12 V <>

t.

* + - 1. Sensores e Atuadores Secundários

Atuadores

Iremos utilizar um Sensor Capacitivo de Proximidade do tipo PNP que será responsável por nos informar a quantidade de mantimentos contidos no reservatório no qual se localiza o alimento do animal desejado.

Um Buzzer será colocado para emitir sons sincronizados em formatos de músicas predefinidas escolhidas pelo cliente.

Um LCD para descrever as situações presentes no circuito de reabastecimento.

Funcionamento

Enquanto houver algo a frente do sensor, ele enviará sinal alto para a placa microcontroladora que está responsável pelo processamento de informações presentes no momento e, está placa enviara para o LCD em seu momento de ação, a informação de que está abastecido o reservatório, mas, porém, todavia, entretanto, quando o sensor enviar sinal baixo, signficará que o reservátorio necessita de reabastecimento, irá acionar o Buzzer que servirá como um alerta para o guardião do animal, para que aja antes de estar completamente vazio e, logicamente, este aviso também estará por escrito no LCD.

t.

* + - * 1. Sensores Abastecimento

t.

* + - * 1. Display informativo

t.

* + - * 1. Buzzer Sinalizador

t.

* + 1. Software

t.

* + - 1. Linguagem C (Arduino)
         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 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.

* + - 1. NodeRed

t.

1. CONCLUSÃO

t.

REFERÊNCIAS BIBLIOGRÁFICAS

t.

ANEXO 1 – Programa em linguagem C (Arduino)