**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 8](#_Toc53423426)

[2 DESENVOLVIMENTO 9](#_Toc53423427)

[2.1 Fundamentação Teórica 9](#_Toc53423428)

[2.1.1 IOT 9](#_Toc53423429)

[2.1.1.1 IoT Industrial 10](#_Toc53423430)

[2.1.2 WI-FI 10](#_Toc53423431)

[2.1.3 MQTT 10](#_Toc53423432)

[2.1.3.1 Características 11](#_Toc53423433)

[2.1.3.2 Funcionamento do MQTT 11](#_Toc53423434)

[2.1.4 Node-RED 12](#_Toc53423435)

[2.1.5 I2C 13](#_Toc53423436)

[2.2 Metodologia 14](#_Toc53423437)

[2.3 Projeto 14](#_Toc53423438)

[2.3.1 Hardware 14](#_Toc53423439)

[2.3.1.1 Abastecimento 15](#_Toc53423440)

[2.3.1.2 Sensores e Atuadores Secundários 15](#_Toc53423441)

[2.3.1.2.1 Sensores Abastecimento 15](#_Toc53423442)

[2.3.1.2.1.1 Características: 15](#_Toc53423443)

[2.3.1.2.2 Display informativo 16](#_Toc53423444)

[2.3.1.2.2.1 Características: 16](#_Toc53423445)

[2.3.1.2.3 Módulo I2C 16](#_Toc53423446)

[2.3.1.2.3.1 Características: 16](#_Toc53423447)

[2.3.1.2.4 Buzzer Sinalizador 17](#_Toc53423448)

[2.3.1.2.4.1 Características: 17](#_Toc53423449)

[2.3.2 Software 17](#_Toc53423450)

[2.3.2.1 Linguagem C (Arduino) 17](#_Toc53423451)

[2.3.2.1.1 WI-FI 17](#_Toc53423452)

[2.3.2.1.2 MQTT 18](#_Toc53423453)

[2.3.2.2 Node-RED 19](#_Toc53423454)

[2.3.2.2.1 Interface de Reservatório 19](#_Toc53423455)

[3 CONCLUSÃO 21](#_Toc53423456)

[REFERÊNCIAS 22](#_Toc53423457)

[ANEXO 1 – Programa em linguagem C (Arduino) 23](#_Toc53423458)

[ANEXO 2 – Programa da Interface Node-RED 24](#_Toc53423459)

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.

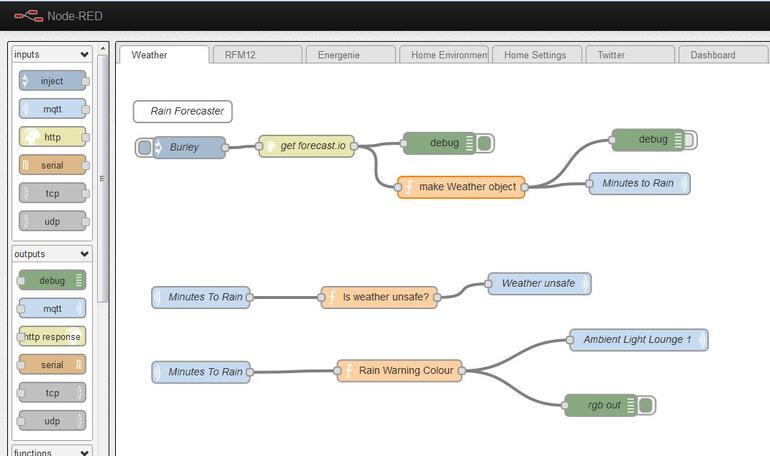


Figura X. Programa de alerta climático. “How IBM's Node-RED is hacking together the internet of Things” – TechRepublic, 2014.

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.

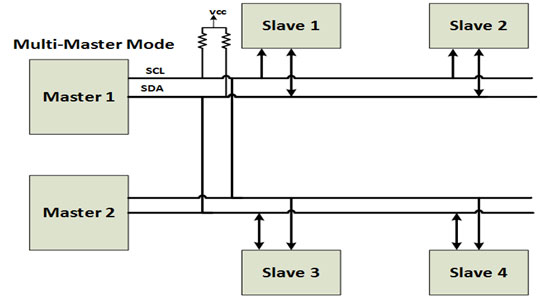


Figura X. Barramento I2C. Fonte: “Understanding the I2C Protocol” - Engineers Garage, 2020.

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:

|  |  |
| --- | --- |
| **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 |

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

* 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
         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

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 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. 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

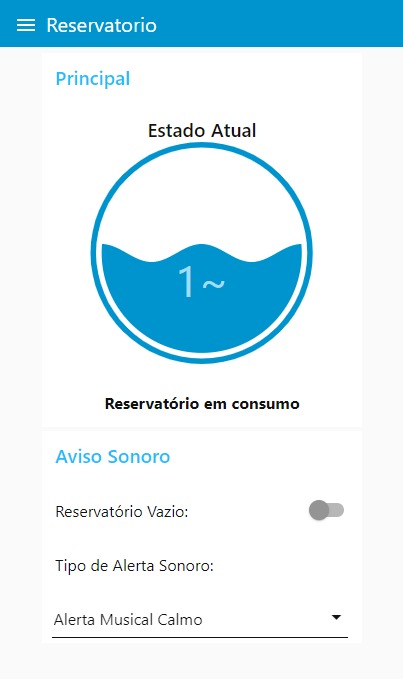


Figura X. Interface de Reservatório. Autores via Node-RED, 2020.

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.

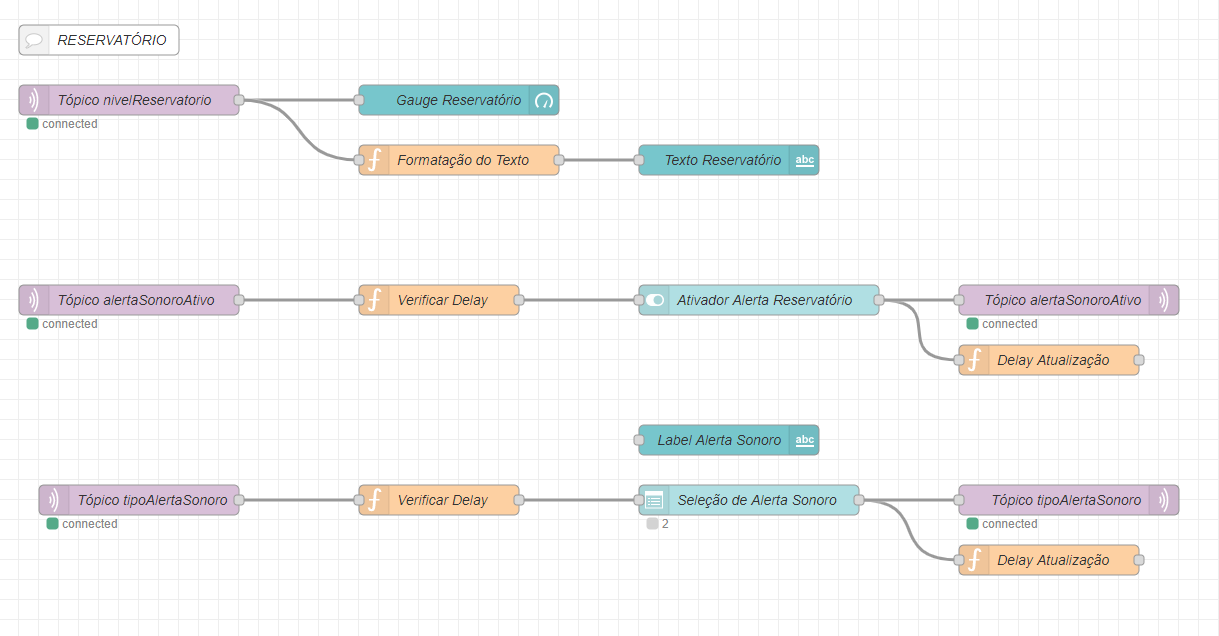


Figura X. Fluxo da interface de reservatório. Autores via Node-RED, 2020.

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.

ESP32 Series Datasheet. **Espressif**, 2020. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32\_datasheet\_en.pdf. Acesso em: 05 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. **Autocorerobotica**, 2018. Disponível em: https://autocorerobotica.blog.br/aprenda-a-reproduzir-musicas-com-o-modulo-buzzer-2/. Acesso em: 05 out. 2020.

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

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

ANEXO 1 – Programa em linguagem C (Arduino)

ANEXO 2 – Programa da Interface Node-RED