|  |  |  |
| --- | --- | --- |
| Ilha Digital | Revista Ilha Digital  Endereço eletrônico:  **http://ilhadigital.florianopolis.ifsc.edu.br/** | Marca IFSC com texto |

**Desenvolvimento de um Alimentador Automático para Gatos com Controle via Aplicativo**

**Resumo** Este artigo apresenta o desenvolvimento de um alimentador automático para gatos, que permite o controle dos horários de alimentação por meio de um aplicativo. O sistema utiliza um ESP32 para processamento e conectividade WiFi, além de um display LCD e uma interface touch para interação direta. O alimentador foi projetado para oferecer praticidade aos tutores, garantindo que os gatos sejam alimentados nos horários programados, mesmo na ausência dos donos. O desenvolvimento incluiu a integração de hardware e software.

**Palavras-chave:** **Alimentador automático. ESP32. WiFi. Aplicativo móvel. Interface touch.**

***Abstract:*** *This paper presents the development of an automatic cat feeder, which allows feeding schedule control through a mobile application. The system uses an ESP32 for processing and WiFi connectivity, along with an LCD display and a touch interface for direct interaction. The feeder was designed to provide convenience to pet owners, ensuring that cats are fed at scheduled times, even in the absence of their owners. The development included hardware and software integration.*

***Keywords:*** *Automatic feeder. ESP32. WiFi. Mobile application. Touch interface.*

# Introdução

A alimentação regular e adequada dos gatos é essencial para sua saúde e bem-estar. No entanto, muitos tutores enfrentam dificuldades em manter uma rotina alimentar devido a compromissos diários. Para solucionar esse problema, os alimentadores automáticos se tornaram uma alternativa viável, permitindo que os animais sejam alimentados mesmo na ausência de seus donos.

Neste contexto, este artigo apresenta o desenvolvimento de um alimentador automático para gatos, que permite o controle dos horários de alimentação por meio de um aplicativo. Diferente de alguns modelos disponíveis no mercado, este projeto foca exclusivamente na programação dos horários das refeições, sem a necessidade de controle de porção. O objetivo é proporcionar praticidade ao tutor e garantir que o gato receba a alimentação nos momentos programados, contribuindo para uma rotina mais organizada.

# Materiais e métodos

A tabela a seguir apresenta os principais componentes utilizados no desenvolvimento do sistema:

|  |  |
| --- | --- |
| Módulos | Quantidade |
| Display LCD | 1 |
| ESP32 | 1 |
| Servo Motor | 1 |
| Fonte 5V/2A | 1 |
| Botão touch | 3 |

O desenvolvimento do projeto seguiu as seguintes etapas:

1. **Testes Individuais dos Componentes**: Foram realizados testes iniciais dos módulos individuais, como display LCD, servo motor, botões capacitivos e RTC interno do ESP32.
2. **Desenvolvimento do Hardware**: O circuito foi projetado e montado, incluindo a criação de uma PCB personalizada no software Altium Designer.
3. **Desenvolvimento do Firmware**: Foi implementado o código no ESP32 utilizando Espressif IDE, garantindo a comunicação WiFi e o controle dos periféricos.
4. **Desenvolvimento do Aplicativo**: A interface do aplicativo foi criada no App Inventor 2, permitindo a configuração remota dos horários de alimentação.
5. **Testes e Validação**: O sistema foi testado para verificar a sincronização entre aplicativo e alimentador, bem como a resposta dos componentes eletrônicos.

**Testes dos Componentes**

Os testes iniciais foram realizados para validar o funcionamento dos módulos antes da integração ao sistema. A imagem a seguir mostra a conexão utilizada para testar o LCD 16x2 com ESP32:

Tela de jogo de vídeo game

Descrição gerada automaticamente com confiança média

Figura 1 - Esp 32 com LCD

No teste do LCD foi utilizado a EspressifIDE e o módulo I2c. Para teste do RTC, utilizamos a RTC interno do ESP32. A seguir temos a parte do código utilizado com as informações do terminal.

Texto

Descrição gerada automaticamente

Figura 2 - RTC

Para validar o funcionamento, utilizamos o terminal do próprio Espressif IDE para debugar o código e observar a alteração automática da data. Alem disso, habilitamos o LED da placa para ficar piscando e conseguirmos observar que o firmware está rodando.

Iniciamos utilizando o projeto de exemplo existente na própia Espressif Ide, para configurar o servo na Esp Idf é utilizado o periférico LEDC para gerar o sinal PWM. Foi feito o teste em bancada utilizando o ESP32 e o servo motor selecionando anteriormente. No arquivo servo.h temos as definições dos pinos e a escolha de qual timer utilizar.

Texto

Descrição gerada automaticamente

Figura 3 - TIMER

Para fazer os testes do touch, iniciamos um projeto de exemplo do próprio Espressif Ide chamado "Touch Pad Read".

Texto

Descrição gerada automaticamente

Figura 4 - Teste do Touch

**Desenvolvimento do Aplicativo**

Para configurar as 24 programações de alimentação por dia vai ser desenvolvido um aplicativo com conectividade via WiFi no APP Inventor2. Inicialmente começamos desenvolvendo a interface de botões e a troca de tela no aplicativo, a lógica de horário também foi iniciada. A tela incial do aplicativo foi desenvolvida, na próxima figura ela é apresentada em:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 5 - Desenvolvimento do Aplicativo

Na próxima imagem temos a figura que mostra uma parte dos blocos do aplicativo desenvolvido. Nessa configuração é realizada a comunicação wifi.

Linha do tempo

Descrição gerada automaticamente

Figura 6 - Desenvolvimento do Aplicativo

Na figura 7 temos a tela inicial do aplicativo desenvolvida, para inicial a operação é necessário clicar no botão conectar. Na figura 8 temos a tela de configuração do aplicativo, nela, temos o botão para sincronizar o horário , voltar, programar alimentação, atualizar alarme na tela e remover os alarmes salvos.



Figura 7 - Tela Principal do Aplicativo

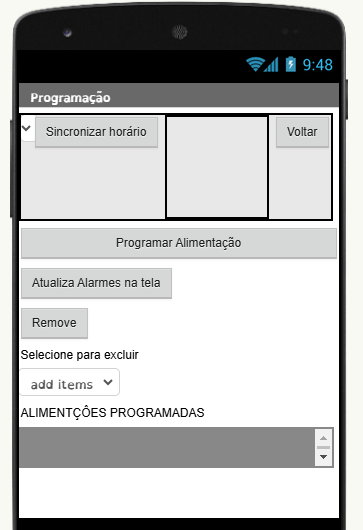


Figura 8 - Tela de Conexão do Aplicativo

**Desenvolvimento da PCB**

A PCB foi projetada no EasyEDA, contendo 3 botões touch, regulador de tensão, conexão para o servo motor e interface I2C para o LCD.

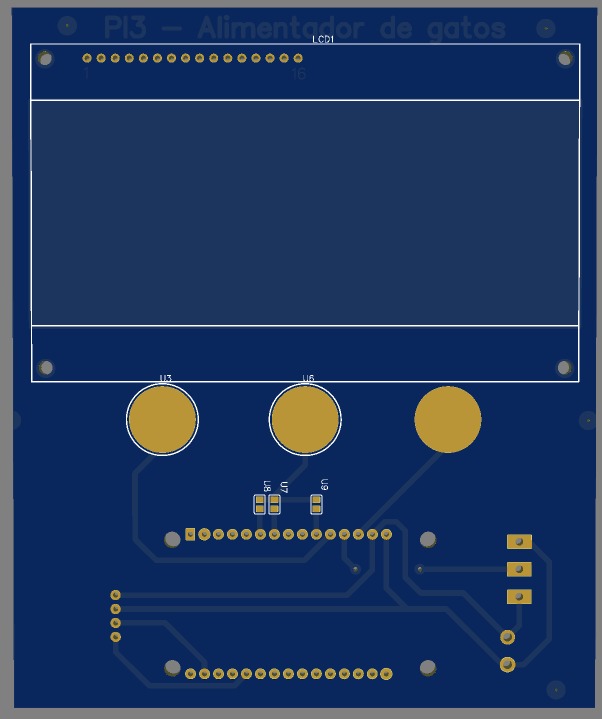


Figura 9 - EasyEDA

Na figura 10 e na figura 11 temos a placa montada e não montada respectivamente, essa placa vai ser fixada na parte superior da estrutura final do alimentador.

Imagem de vídeo game

Descrição gerada automaticamente com confiança média

Figura 10 - Placa Desenvolvida

Mesa com livros em cima de uma superfície de madeira

Descrição gerada automaticamente com confiança baixa

Figura 11 - Placa Desenvolvida

**CONCLUSÃO**

A implementação de um sistema de alarmes para um alimentador automático de gatos no App Inventor, utilizando uma tabela para exibir horários e doses, permite ao usuário visualizar e gerenciar facilmente as programações de alimentação.

Com a abordagem de armazenar alarmes em uma lista e atualizar dinamicamente a interface, garantimos uma experiência intuitiva e eficiente. Além disso, ao integrar TimePicker e Notifier, o usuário pode definir horários e doses sem complicações.

# REFERÊNCIAS

A2ROBOTICS. **Display LCD 16x2**. [S.d]. Disponível em: <https://www.a2robotics.com.br/display-lcd-16x2-azul-com-i2c-acoplado?utm_source=Site&utm_medium>. Acesso em: 16 mar. 2024.

MAKERHERO. **Servo motor 6221mg**. [S.d]. Disponível em: <https://www.makerhero.com/produto/servo-jx-pdi-6221mg-alto-torque-20kg/>. Acesso em: 16 mar. 2024.

FRANZININHO. **Exemplos ESPIDF**. [S.d]. Disponível em: <https://docs.franzininho.com.br/docs/franzininho-wifi/exemplos-espidf/primeiros-passos>. Acesso em: 21 mar. 2024. <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>