

Relatório Técnico

PetFeeder

Sistema remoto de alimentação e monitoramento de animais

Bruno Gabriel T. B. Silva – mrbrunogt@gmail.com

Lucas Ferreira – lheroso@gmail.com

Maurício Matsumoto – mauricio.matsp@gmail.com

Vinícius Terres – viniusterres@alunos.utfpr.edu.br

Maio de 2018

Resumo

Em tempos onde a rotina e a falta de tempo são comuns no dia a dia, tarefas simples como cuidar o animal de estimação acabam cada vez mais difíceis de encaixar na agenda. Com o intuito de solucionar este problema, este projeto traz a funcionalidade de alimentar e monitorar o cachorro de estimação enquanto o dono não puder estar em casa. Utilizando os recursos do PetFeeder, o usuário será capaz de alimentar o seu pet de forma automática, configurando o intervalo de tempo entre refeições pelo aplicativo, assim como fazer o monitoramento através da câmera que acompanha o sistema, além disso, o usuário também será notificado caso haja algum distúrbio sonoro no ambiente que possa incomodar seu animal de estimação. Com PetFeeder o animal de estimação recebe toda a atenção necessária enquanto o dono não está em casa.

1 Introdução

Este projeto tem como objetivo principal desenvolver um sistema remoto de alimentação e monitoramento de animais de estimação, para usuários que ficam pouco tempo em sua casa e geralmente não possuem tempo para realizar tais tarefas.

A seguir, pode ser observado o diagrama de visão geral do projeto (Figura 1), onde podem ser vistos os principais componentes envolvidos no desenvolvimento do PetFeeder.

Além de ser um sistema automático de alimentação para o cachorro, o projeto também consta com um sistema de monitoramento, composto de uma câmera e um microfone. A câmera, para ter um campo de visão maior, é movida

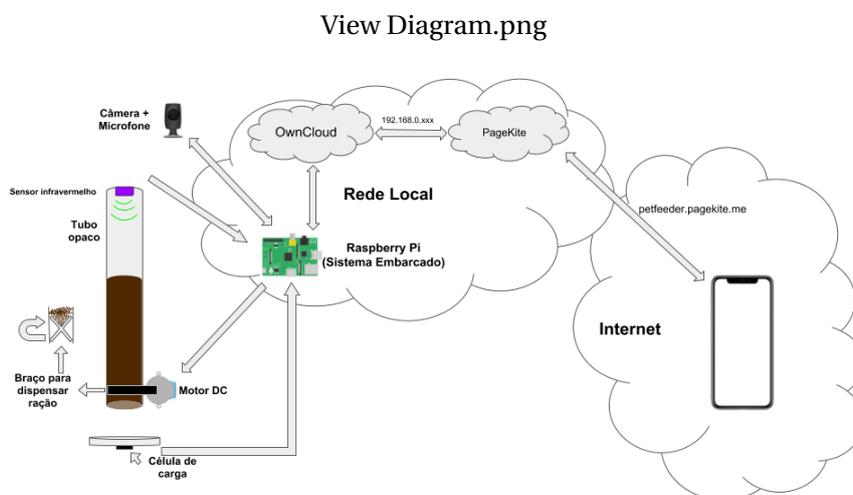


Figura 1: Diagrama de visão geral

verticalmente e horizontalmente por dois servo motores, obtendo imagens que podem ser visualizadas em um aplicativo diretamente de um smartphone.

O microfone é um importante componente para tornar possível o desenvolvimento do sistema de detector de distúrbios, uma vez que o microfone detecta um ruído alto e constante por um certo tempo, a câmera automaticamente gera imagens do ambiente e envia ao usuário, com a notificação de que há algum tipo de distúrbio sonoro no ambiente.

A ideia base do projeto, servir ração automaticamente ao cachorro a cada intervalo de tempo, será implementada da seguinte forma.

- O usuário configura o intervalo de tempo para que seja servida a ração;
- Após a primeira configuração, o sistema já passa a funcionar de forma automática, servindo as refeições mesmo que haja a perda de conexão com a internet;
- A parte mecânica para liberar a ração no pote consta com, uma 'hélice' em formato de 'X' que é rotacionada por um motor DC, de forma que cada um quarto do 'X' represente uma unidade para mensurar a quantidade servida (que pode ser configurada pelo usuário). Além disso, uma célula de carga posicionada abaixo do pote de ração funcionará como um 'feedback' para o sistema verificar se a ração realmente foi servida ou não;
- Um sensor de infravermelho é utilizado para mensurar a quantidade restante de ração no tubo, de forma que o usuário possa monitorar e identificar quando é o momento de complementar o reservatório de ração.

A estação base do sistema é um aplicativo para smartphones Android, onde o usuário pode programar os horários de refeição ou então acionar a liberação de ração a qualquer momento, assim como acompanhar imagens da câmera de monitoramento. Além disso, o usuário será notificado em diferentes eventos como, distúrbio sonoro no ambiente, nível baixo de ração no reservatório, sistema sem conexão, etc.

A comunicação e a troca de dados entre o sistema embarcado e a estação base é realizada por meio da internet.

O sistema embarcado neste projeto é responsável por realizar a captação do estado do ambiente em que o aparelho será instalado e a organizar a entrega da ração de acordo com a programação definida. Além disso, ele possui um servidor com os dados coletados da câmera.

Abaixo encontram-se os requisitos do projeto PetFeeder.

Requisitos Funcionais:

- O aplicativo deve permitir ao usuário duas opções de uso, manual ou automático;
- O aplicativo deve permitir o usuário acionar a liberação de de uma a 9 porções de ração qualquer momento (desde que haja conexão com a internet);
- O aplicativo deve permitir que o usuário visualize um vídeo (ou fotos), que permitam o monitoramento do ambiente;
- O aplicativo deve gerar uma notificação para o usuário quando acabar a ração armazenada no reservatório;
- O aplicativo deve possibilitar a consulta da quantidade de ração disponível no reservatório;
- No modo automático do aplicativo, o usuário deve ser capaz de escolher o intervalo de tempo e em que quantidade a ração deve ser dispensada na vasilha;
- No modo automático do aplicativo, o usuário deve ser capaz de escolher o intervalo de tempo para o vídeo (ou fotos) de monitoramento;
- O aplicativo deve gerar uma notificação para o usuário caso sejam detectados distúrbios sonoros prolongados no ambiente;
- O sistema embarcado deverá medir a quantidade de ração na vasilha;
- O sistema embarcado deverá medir a quantidade de ração armazenada no reservatório;
- O sistema embarcado deverá colocar a ração na vasilha nos momentos definidos pelo usuário;

- O sistema automático, se selecionado, deve funcionar mesmo se ocorrer perda de acesso a rede;

Requisitos não Funcionais:

- A comunicação entre o sistema embarcado e a estação base será feito por WI-FI;
- O sistema embarcado utiliza um circuito elétrico que carrega constantemente a bateria, que, após uma queda de energia, deverá capaz de manter a operação do sistema por uma hora;
- O aplicativo funcionará somente em smartphones Android;
- O banco de dados armazenará mídia por trinta dias;
- A estrutura utilizada para armazenar a ração possuirá um formato cilíndrico com, no máximo, 60 cm de altura e 10 cm de diâmetro;
- O material para armazenar a ração deverá ser opaco.

O presente documento tem por finalidade a documentação sob um aspecto técnico do trabalho realizado pela equipe durante a concepção e realização do projeto PetFeeder. Esse relatório está estruturado de maneira que as próximas seções trazem uma visão mais detalhada da estrutura eletromecânica, da utilização de softwares e dos resultados obtidos, bem como das referências utilizadas.

2 Base Teórica

Nesta seção serão apresentados os principais componentes e ferramentas utilizadas no desenvolvimento deste projeto.

2.1 Raspberry Pi

O Raspberry Pi 3 Modelo B [1] é o microcontrolador selecionado para realização deste projeto, responsável pela integração de todos os componentes do PetFeeder, assim como armazenamento dos dados gerados pelo sistema.

A plataforma foi escolhida pois possui todos os recursos necessários para a realização deste projeto, como Wi-fi integrado, que é necessário para a comunicação com o aplicativo do usuário por exemplo. Estes e outros recursos tornaram o Raspberry a plataforma mais viável para a realização do projeto.

2.2 ownCloud

O servidor web onde os dados serão armazenados e posteriormente transferidos para o usuário, será implementado com a ferramenta ownCloud [2] que, após a instalação do cliente no Raspberry Pi, torna o microcontrolador em um servidor web acessível localmente.

Uma vez instalado e configurado, é possível realizar a transmissão de dados entre a estação base e o sistema embarcado, tais dados incluem fotos e vídeos do ambiente gravados pela câmera de monitoramento do PetFeeder, notificações na hora das refeições, etc.

2.3 PageKite

A ferramenta open-source PageKite[3] foi utilizada para realizar a comunicação entre a estação base e o sistema embarcado.

Utilizando uma técnica conhecida como tunelamento dinâmico de proxy reverso[4], com isto, é possível que o servidor ownCloud configurado localmente no Raspberry Pi, seja visível na internet a partir de um endereço criado na configuração do serviço PageKite.

2.4 Microfone

No contexto do projeto, o microfone foi responsável por captar o som ambiente. O microfone escolhido é o ECM-ds70p da marca Sony [5]. A sua principal aplicação é a captura de sons externos. As frequências adquiridas por ele estão entre 100 Hz e 10100 Hz. A sua sensibilidade é de -38dB com uma variação de 5dB. O nível de ruído de 34dB SPL. O peso do modelo é de 24g. Como Raspberry PI não possui entradas de áudio com o conector Jack, o adaptador de Placa de Som 7.1 Usb Knup Hb-T64 será utilizado.

2.5 Sensor infravermelho

O princípio de funcionamento do sensor infravermelho baseia-se na emissão de radiação infravermelha para medir as distâncias. Ele é composto principalmente por um LED emissor de luz infravermelha e um fotodiodo. O modo de operação do sistema é o seguinte: o LED infravermelho emite a sua radiação. Ela refletirá no objeto e retorna para o sensor. Em seguida, o receptor detecta o sinal. Com base na sua intensidade, um sinal analógico é gerado. A sua função no sistema é a medição da quantidade de ração no reservatório. O modelo escolhido foi o GP2Y0A21 [6]. Ele é capaz de medir distâncias entre 10 cm e 80 cm.

2.6 Célula de carga

Um dos transdutores de força conhecidos atualmente é a célula de carga.

Ela é um dispositivo que altera a resistência elétrica de acordo com a sua deformação. Considerando que são obtidas pequenas variações de resistência, uma ponte de Wheatstone é utilizada para amplificar os resultados obtidos com a alteração da resistência. Na maioria dos casos ela é utilizada no modo ponte completa. Com base nos modelos disponíveis para a aplicação, a célula de carga escolhida foi a do modelo com medição máxima de 1 kg.

2.7 Conversores analógico-digitais

Os conversores analógico digital escolhidos para este projeto foram HX711 [7] e o ADS1115[8]. O primeiro foi responsável por converter os sinal analógico proveniente da célula de carga em um valor digital. O primeiro componente integrado é um conversor específico para células de carga. O selecionado possui dois canais de entrada. Os resultados das conversões têm 24 bits. A interface de saída com o microcontrolador utiliza o protocolo SPI (Serial Protocol Interface) [9].

A segunda peça possui a finalidade de converter o sinal obtido pelo sensor infravermelho. Ele é um conversor de uso genérico, utilizado primariamente em casos que exigem uma resolução maior e não necessitam de uma taxa de amostragem elevada. O ADS1115 possui 4 canais para a entrada de dados analógicos. A sua taxa de amostragem máxima é de 880 amostras por segundo. A sua interface de saída implementa o protocolo I2C(Inter-Integrated Circuit) [9]

2.8 Motor DC

Os motores DC também são importantes dispositivos que operam aproveitando todas as forças de atração e repulsão geradas por eletroímãs e imãs permanentes. Esse tipo de produto possui vários tamanhos e tensões de trabalho, que são indispensáveis para os projetistas do ramo da automação industrial e mecatrônica, principalmente em robótica, equipamentos bancários, entretenimento, esteiras alimentadoras, entre outros.

Para o projeto foi utilizada uma caixa de redução para se obter um maior torque, já que dessa forma se cumpriram os requisitos para movimentar a hélice dispensadora de alimento. O modelo do motor DC com a caixa de redução é o AK510/15PF12R3SE [10]. Ele é alimentado com a tensão de 12V. O seu torque e velocidade de rotação máximos são de 15kgf/cm e 3 RPM.

2.9 Ponte H

A ponte H é um circuito com a finalidade de acionar motores DC através de microcontroladores. Esses atuadores exigem uma potência maior do que o microcontrolador consegue oferecer. A partir do nível lógico de tensão nas suas entradas, ele realiza o acionamento do motor. O modelo escolhido foi o L298N [11]. Ele possibilita o acionamento simultâneo de dois motores DC com tensão e corrente máximas de 46V e 4A, respectivamente.

2.10 Servo Motor

O servo motor é um dispositivo eletromecânico, de movimento proporcional e de malha fechada. Desta forma, o dispositivo recebe um sinal de controle, verifica a posição atual e se movimenta para a posição desejada. Assim, através de um sinal PWM(Pulse Wide Modulation, Modulação por Largura de Pulso)[12] em sua entrada, é possível detectar qual a angulação desejada e assim realizar a rotação necessária para alcançar esta. No projeto são utilizados dois servo motores, possibilitando uma maior angulação de monitoramento ao usuário, podendo variar o ângulo de visualização da câmera de monitoramento na horizontal ou na vertical. O modelo dos servo motores utilizados é o SG90.[13]

2.11 Android Studio

Android Studio é uma IDE (Integrated Development Environment) para o desenvolvimento para a plataforma Android.

Sua primeira compilação estável foi lançada em dezembro de 2014. Baseado no IntelliJ IDEA, que é uma IDE que também oferece suporte ao Android, mas possui um custo elevado. Está disponível para download em Windows, Mac OS X e Linux, e substituiu o Eclipse Android Development Tools (ADT) como a IDE primária do Google de desenvolvimento nativo para Android.

Como características principais o Android Studio tem suporte para compilações baseadas em Gradle, suporte nativo para a Google Cloud Platform e suporte para a criação de apps para Android Wear.

2.12 Câmera

A câmera utilizada é uma específica para o Raspberry PI de 5 MegaPixel [1]. O módulo possui um sensor de 5 MegaPixel OV5647, uma lente foco fixo (1m ao infinito) capaz de fornecer resolução de 2592 x 1944 pixels para imagens estáticas e suportar 1080p30, 720p60 ou 640x480p60/90 para vídeo com codec H.264. O tamanho da placa é de 25 x 24 mm e o peso de aproximadamente 3g.

2.13 Bateria

Dois baterias de 12V 2200mAh, ligadas em série foram utilizadas para alimentar o sistema embarcado e seus periféricos quando houver queda ou falta de energia elétrica. Com o auxílio de um circuito carregador de bateria, o sistema recarrega automaticamente a bateria quando conectado a uma fonte de 19,5V e 3,3A. Caso contrário, a bateria continuará a fornecer a alimentação dos demais componentes. A saída do sistema de bateria terá um pino com 5V e outro com 12V. O diagrama esquemático do sistema está na Figura 2.

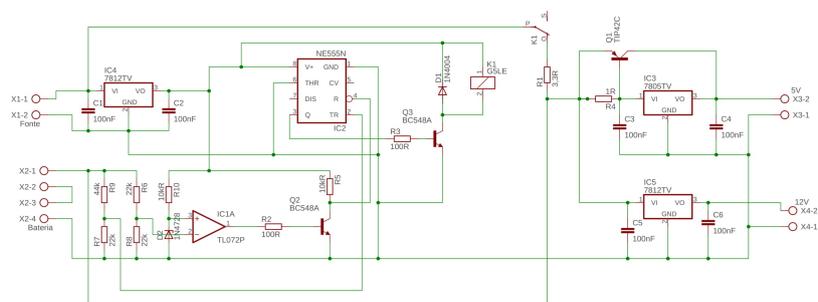


Figura 2: Diagrama esquemático do sistema de Alimentação

3 Desenvolvimento

Nesta seção, será abordado o desenvolvimento dos diversos tópicos do projeto, com uma descrição detalhada de cada um dos temas desenvolvidos para criação do PetFeeder.

3.1 WebServer e comunicação remota

O sistema de monitoramento do PetFeeder tem como objetivo o acesso aos vídeos e imagens gerados pela câmera, a partir de qualquer lugar, uma vez que a ideia é que o usuário consiga monitorar seu pet mesmo estando fora de casa.

Para isso, inicialmente era preciso descobrir uma maneira de transferir estes vídeos e fotos entre o sistema embarcado e a estação base, assim, a ferramenta ownCloud foi selecionada pela praticidade de trabalhar com tais dados. Como descrito anteriormente, ownCloud é uma ferramenta que funciona como os serviços de armazenamento em nuvem, porém neste caso, os dados ficam armazenados no Raspberry Pi.

Após a instalação da ferramenta no Raspberry, é possível acessar a página inicial de configuração acessando o endereço IP local do Pi a partir de um navegador, esta configuração inicial consiste em cadastrar um usuário e senha para que o acesso seja feito posteriormente.

Uma vez autenticado, o usuário consegue acessar os arquivos do diretório

definido no Raspberry Pi, assim como carregar arquivos para o microcontrolador.

O usuário final não necessita compreender o uso desta ferramenta, e portanto, uma API no Android Studio foi utilizada no desenvolvimento do aplicativo para fazer esta "ponte" entre o usuário e os arquivos de forma invisível ao usuário. Mais informação sobre esta API serão fornecidas na secção do desenvolvimento do aplicativo.

O problema de transferência de fotos e vídeos entre o sistema embarcado e a estação base é resolvido utilizando a ferramenta ownCloud, porém, apenas com a instalação da ferramenta, o usuário só consegue acessar os arquivos na mesma rede em que o PetFeeder está conectado, ou seja, quando está em sua residência. Para resolver este problema, foi utilizada a ferramenta PageKite.

Como descrito anteriormente, PageKite é um serviço open source e gratuito que permite a comunicação remota entre a estação base e o sistema embarcado utilizando uma técnica conhecida como tunelamento. Com isto, é possível que o servidor ownCloud configurado localmente no Raspberry Pi, seja visível na internet a partir de um endereço criado na configuração do PageKite.

Após a configuração do PageKite ser concluída, o servidor implementado no Raspberry Pi já pode ser acessado de uma rede externa a partir do endereço previamente configurado, como PetFeeder será instalado numa rede Wi-Fi comum, o endereço de IP público do usuário não é fixo, porém, o serviço PageKite também já cuida deste problema, caso haja a mudança de IP, o endereço criado na configuração é automaticamente vinculado ao serviço. Todo este processo é transparente ao usuário final.

3.2 Detector de Distúrbios

A implementação do detector de distúrbios tem como objetivo fazer com que o sistema, logo após detectar um distúrbio, ative a câmera de monitoramento para fazer um vídeo do ambiente.

Nesta implementação foram utilizados os seguintes componentes:

- Microfone estéreo de alta captação;
- Placa de som USB.

O microfone foi um tópico de debate no início do projeto, pois é necessário que este seja capaz de captar sons de todo o ambiente em que o sistema está instalado (uma sala de apartamento por exemplo), e muitos microfones não tinham a capacidade de captar sons a mais de 2m de distância, e mesmo que amplificados, não seriam precisos.

O uso do microfone estéreo de alta captação permitiu que o sistema de detecção de distúrbios funcionasse da maneira mais precisa possível.

Como o microfone possui uma saída P2, e o Raspberry Pi não possui uma entrada P2, uma placa de som USB foi utilizada para fazer o interfaceamento entre o microfone e o micro-controlador.

Para a captação do som ambiente, um script em Python baseado na ferramenta SoundMeter foi utilizado. SoundMeter é uma ferramenta que retorna a RMS (root-mean-square) de fragmentos de sons captados pelo microfone, com estes valores, é possível determinar a partir de qual valor o sistema deverá considerar como distúrbio.

Utilizando isto, o script verifica se este valor configurado é captado mais de 10 vezes pela ferramenta, caso sim, o sistema considera como sendo um distúrbio e ativa a câmera de monitoramento para gravar um vídeo do ambiente e também notifica o usuário.

3.3 Aplicativo

Para o desenvolvimento do aplicativo foi utilizado o Android Studio em sua versão estável mais recente (3.1). O aplicativo tem como objetivo disponibilizar ao usuário de forma simples o acesso e controle das funcionalidades do sistema de monitoramento PetFeeder.

Para a comunicação com o serviço do ownCloud foi utilizada uma biblioteca disponibilizada pelos próprios criadores do ownCloud desenvolvida pela comunidade em parceria com eles. Esta biblioteca fornece métodos para autenticação de usuário, explorar quais arquivos e pastas estão em um determinado diretório, criar um novo diretório, download, upload e exclusão de arquivos.

O aplicativo foi dividido nas seguintes telas:

- Autenticação do usuário: A Figura 3 mostra a tela onde foram colocados dois campos para login e senha, onde as credencias utilizadas são as mesmas para acessar o serviço do ownCloud direto pelo seu endereço. No topo direito foi colocado o acesso a uma tela de configurações.

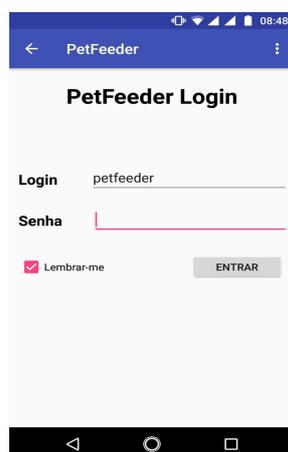


Figura 3: Tela de autenticação de usuário

- Configurações: A figura 4 mostra a tela de configurações gerais do sistema

foi colocado um campo para configurar o endereço do PetFeeder. Este endereço é então armazenado no próprio celular evitando a necessidade do usuário ter de digitá-lo sempre que acessar o sistema e permitindo uma flexibilidade de se ter mais de um PetFeeder.



Figura 4: Tela de configurações



Figura 5: Tela do menu principal

- **Menu principal:** A Figura 5 mostra a tela do menu principal onde foram colocados três botões na parte inferior para acessar as três funções principais do aplicativo: Modo manual, modo automático e um explorador de Mídia. No meio da tela foi colocado uma lista que carrega em tempo real as últimas fotos e vídeos tiradas pelo Petfeeder. No canto superior direito um ícone de bateria informa a quantidade de ração disponível para ser servida.
- **Modo Manual:** A Figura 6 mostra a tela do modo manual onde foram colocados quatro botões, o primeiro para fazer um vídeo do ambiente, o segundo para tirar uma foto do pote, o terceiro para servir uma quantidade específica de ração e o quarto para conferir a quantidade de ração no pote.



Figura 6: Tela do modo manual

- Modo automático: A Figura 7 mostra a tela do modo automático onde foram colocados três botões, o primeiro para acessar a lista de horários e quantidades de porções programadas para serem servidos, o segundo para acessar a lista de horários programados para gravar um vídeo do ambiente e o terceiro para acessar a lista de horários para tirar uma foto do pote. Abaixo dos botões foi colocado uma opção para ligar ou desligar o detector de distúrbios, quando ativado e um distúrbio for detectado o usuário receberá uma notificação e então o detector será desativado para esperar o usuário ligar novamente.

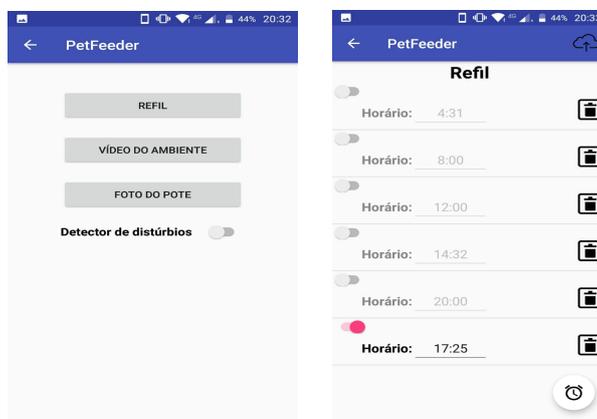


Figura 7: Tela do modo automático

- Mídia: A Figura 8 mostra a tela de mídia onde foram colocados dois botões, o primeiro para vídeos e o segundo para fotos. Ao selecionar qualquer um deles uma nova tela se abrirá e uma lista com as datas que contenham aquele tipo de arquivo (vídeo ou foto) será mostrada. Nesta lista de datas ao selecionar um item, uma nova tela se abrirá com todos os arquivos daquela data em forma de lista para ser selecionado e exibido.

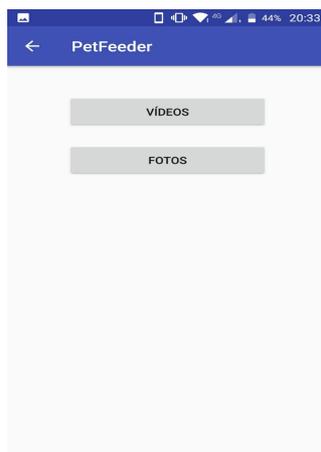


Figura 8: Tela de mídia

3.4 Estrutura

O desenvolvimento da estrutura física foi dividido em duas partes: construção da estrutura em madeira para a sustentação do sistema embarcado e desenvolvimento da estrutura responsável pela regulação da saída de ração.

A estrutura mecânica, criada para a sustentação do sistema embarcado e para possibilitar a mobilidade deste, é composta por:

- Um suporte de madeira em formato de L, medindo 95 cm de altura por 36 cm de largura na "parede" e uma base com 43 cm de altura por 24 cm de largura, com uma prateleira acoplada à parte vertical para sustentar o microcontrolador; e um cano de PVC, com 50 cm de altura e 10 cm de diâmetro, disposto na posição vertical usado para armazenamento de ração. A Figura 9 mostra uma imagem da estrutura base montada.



Figura 9: Estrutura Base

- Um pote de ração associado à uma célula de carga, usada para medir a quantidade de ração nesse. A Figura 10 mostra uma imagem da localização da célula de carga em relação ao pote e de sua fixação



Figura 10: Célula de Carga e Pote de Ração

- Na tampa do recipiente de ração foi acoplado um sensor infravermelho de distância, para medir a quantidade de ração armazenada nele. A Figura 11 mostra imagens da implementação



Figura 11: Interior e exterior da tampa do recipiente

Internamente ao cano, há um módulo para acionar a queda da ração no pote. A peça utilizada para o controle da quantidade de ração no pote é uma estrutura com uma base cilíndrica com 4 pás e uma entrada para o eixo chanfrado. As suas dimensões são semelhantes ao do cano. A Figura 12 mostra uma imagem do peça construída. Ela foi modelada no programa SketchUp [14] e impressa na impressora 3D Wanhao Duplicator [15] por meio do software RepliatorG.

A hélice é acoplada com um eixo de 12mm. Por sua vez, o eixo está acoplado com um motor DC com uma caixa de redução. Utilizando o mesmo procedimento da hélice, a peça de junção foi impressa para a conexão entre o motor DC e o eixo. A fixação do motor DC com a caixa de redução na base do protótipo foi implementada por meio de uma braçadeira. A Figura 12 apresenta uma imagem do sistema mecânico descrito.

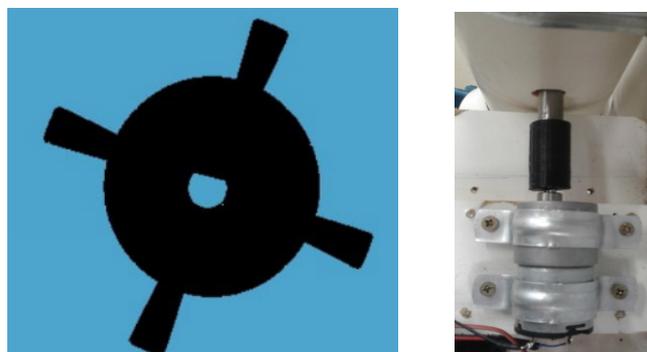


Figura 12: Estrutura para a liberação de ração

3.5 Sistema Embarcado

O sistema embarcado é responsável por gerenciar todos os componentes de hardware do PetFeeder, ativando-os de acordo com as configurações do usuário no aplicativo (estação base).

Foram desenvolvidos diversos scripts em Python para realizar tais tarefas, basicamente, o script faz a leitura dos arquivos de configuração gerados pela estação base e então, após analisar, realiza a tarefa.

A implementação do modo automático, tanto de retil, gravação do vídeo ambiente e foto, se deu da seguinte forma.

- Após a configuração dos horários de uma das três opções do modo automático, a estação base gera um arquivo .xml contendo as informações de horários e envia este para a o sistema embarcado.
- Um script em Python no sistema embarcado verifica a cada 30 minutos mudanças nestes arquivos .xml atualizando as informações do modo automático.
- Com as informações atualizadas, outro script verifica constantemente a cada minuto se o horário contido nas informações bate com o horário atual, caso sim, a tarefa é executada.

Testes de performance foram realizados para verificar se as constantes leituras aos arquivos prejudicariam o desempenho, e, mesmo com uma taxa de leitura de uma por segundo, o uso do processador não ultrapassa os 5% tornando esta implementação viável.

O modo automático já vem com informações padrão de horários para cada uma das 3 opções, quando o usuário configura pela primeira vez, estes dados são sobrescritos.

A implementação do modo manual foi feito de maneira semelhante, porém, como o modo manual deve ter resposta imediata do sistema, a taxa de leitura do arquivo gerado pela estação base é de 5 segundos.

3.6 Sistema de alimentação

O sistema de alimentação é responsável por fornecer a tensão necessária para a operação do projeto. Ele é formado por uma fonte de 19,5V e 3,3A, duas baterias de 12V, um circuito para a recarga das baterias e os reguladores de tensão com saídas de 5V e 12V.

Inicialmente, uma primeira solução foi proposta e montada em uma proto-board para teste do funcionamento efetivo do circuito. Após alguns ajustes e a verificação da eficácia, elaborou-se a versão descrita na seção 2.13. Em sequência, o sistema proposto foi soldado em uma placa de fenolite perfurada.

A carga requerida para o funcionamento de todo sistema é de 2,5A. O cálculo realizado considerou o pior caso de operação. Devido ao baixo valor em relação

ao demais, alguns componentes foram considerados em um item somente. A Equação 1 apresenta as contas realizadas.

$$Carga = I_{Raspberry} + I_{MotorDC} + I_{Restante} = 2,1A + 0,3A + 0,1A = 2,5A \quad (1)$$

A partir disso, é possível estimar a autonomia do sistema. O seu resultado foi de 53 minutos, como demonstrado pela Equação 2.

$$Autonomia = \frac{2200mAh}{2500mA} \approx 53minutos \quad (2)$$

4 Conclusão

O desenvolvimento do projeto resultou em um sistema remoto de monitoramento e alimentação para animais domésticos, de pequeno a médio porte. A função de monitoramento gerou resultados satisfatórios, alertando o usuário quando ocorrem distúrbios indesejados, se este assim o desejar; possibilitando a visualização do ambiente em que se encontra o sistema e do pote de ração a qualquer momento, somente necessitando de um dispositivo *Android* conectado à internet. Além dele poder configurar horários fixos, quando o sistema deverá gerar um vídeo do ambiente ou uma foto do pote automaticamente.

Durante o progresso do projeto foi possível notar a importância de ter tempo de folga no cronograma, uma vez que vários problemas aconteceram: queima de motor de passo, inviabilidade deste para o sistema dispensador de ração, dificuldades na montagem da estrutura mecânica e pouca disponibilidade dos integrantes da equipe se reunirem, visto que três desses fazem estágio e um trabalha.

Além disso, outra questão importante é a autonomia do produto apresentado. Considerando a finalidade do produto, a autonomia da bateria não está adequada para o uso prático. Uma solução possível para esse problema é a utilização de uma bateria com maior capacidade de armazenamento. Contudo, a melhor resposta para a questão é a ativação temporária do Raspberry Pi, deixando-o em modo de espera o restante do tempo.

O projeto PetFeeder desenvolvido engloba o que já existia no mercado (outros produtos com a mesma finalidade), porém trás recursos de monitoramento em tempo real e controle da ração servida que não é encontrado dessa maneira em outros produtos.

Referências

- [1] Raspberry Pi Foundation. Raspberry pi. <https://www.raspberrypi.org>, 2015. Acesso em: 28/06/2018.
- [2] Site oficial do owncloud. <https://owncloud.org/>. Acesso em: 04/05/2018.

-
- [3] Site oficial do pagekite. <https://pagekite.net/>. Acesso em: 04/05/2018.
 - [4] Luis Paulo Gonçalves Pires et al. Alta disponibilidade: uma abordagem com dns e proxy reverso em multi-cloud. 2016.
 - [5] Sony Corporation. *Electret Condenser Stereo Microphone*, 3 2001.
 - [6] Sharp Inc. *GP2Y0A21 datasheet*, 6 2013.
 - [7] Avia Semiconductor. 24-bit analog-to-digital converter (adc) for weigh scales. *Saatavissa: http://www.dfrobot.com/image/data/SEN0160/hx711_english.pdf Viitattu*, 4, 2016.
 - [8] Texas Instruments. *Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference*, 5 2009. Rev. October 2009.
 - [9] Frédéric Leens. An introduction to i2c and spi protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1):8–13, 2009.
 - [10] Sparkfun. Micro motor dc com caixa de redução. <https://www.baudaeletronica.com.br/Documentos/micro-motor-dc-12-vdc-rpm-3-rpm.pdf>, 2015. Acesso em: 28/06/2018.
 - [11] ST Eletronics. L298n - dual full-bridge driver. <https://www.st.com/en/motor-drivers/l298.html>, 2000. Acesso em: 28/06/2018.
 - [12] Michael Barr. Pulse width modulation. *Embedded Systems Programming*, 14(10):103–104, 2001.
 - [13] Tower Pro. Sg90 datasheet. <http://www.towerpro.com.tw/product/sg90-7/>. Acesso em: 28/06/2018.
 - [14] Site oficial do sketchup. <https://www.sketchup.com/pt-BR>. Acesso em: 04/05/2018.
 - [15] Site oficial da impressora 3d. <http://replicat.org/>. Acesso em: 04/05/2018.