

Monitoramento de fluxo de animais de médio e grande porte no Jardim Botânico de Brasília por câmera trap

Gabriel Santos Silva Araújo
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
gabrielsantos_s96@hotmail.com

Yasmine Silveira Andrade
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
andradeyasmine96@gmail.com

Resumo — O presente documento consiste em uma proposta de projeto que visa a partir do uso da raspberry pi 3 identificar animais de médio e grande porte da reserva do Jardim Botânico de Brasília e fotografá-los, e enviar estes dados para um servidor para realização do controle de fluxo dos animais e avaliação de seus hábitos.

Palavras-Chave — *Raspberry; Reserva ambiental; Câmera trap; Fluxo de animais.*

I. INTRODUÇÃO

Existem várias espécies de mamíferos ameaçadas de extinção, especialmente pela perda de habitat. Por isso, é imprescindível que estudos científicos sejam feitos o mais rápido possível com o objetivo de proporcionar diretrizes para planos de manejo que visem a sua conservação [1].

As armadilhas fotográficas, também conhecidas como câmeras trap, são equipamentos eletrônicos amplamente utilizados para fins conservacionistas, em especial para estudos populacionais ou de comunidades de mamíferos de médio e grande porte terrestres[2]. Como este método é muito eficaz e não invasivo, pode ser usado para realização do monitoramento do fluxo de animais de médio e grande porte pelas áreas de reserva do Jardim Botânico de Brasília (JBB).

Conhecer a movimentação desses animais pela reserva é importante definir suas rotas utilizadas, servindo como base para decisões importantes do JBB, como a necessidade de

modificar as áreas de pesquisa ou de acesso ao público e de criar novas áreas de preservação e implantar vias de circulação subterrânea para travessia da rodovia.

Um dos problemas enfrentados hoje é a dificuldade de acesso à algumas áreas e a ocorrência de furtos e perdas das câmeras já instaladas.

Além de serem em quantidade insuficiente e de custo alto, as armadilhas fotográficas utilizadas armazenam as imagens registradas em cartões de memória removíveis e, quando é perdido ou há danos no equipamento, não há como resgatar as fotos, sendo também, necessário disponibilizar mão de obra para recolher o cartão de armazenamento periodicamente.

II. OBJETIVO

Desenvolver um sistema embarcado para auxiliar no monitoramento e fluxo de mamíferos de médio e grande porte no Jardim Botânico de Brasília a partir da coleta e envio de dados aos funcionários.

III. Requisitos e Benefícios

Partindo do problema mencionado, foi verificado a necessidade da construção de uma armadilha fotográfica capaz de capturar imagens de animais de médio e grande porte. Devido a necessidade de locomoção até as áreas a serem monitoradas, o equipamento precisa ter um baixo consumo de

bateria, sendo requerido no mínimo cinco dias no intervalo entre uma troca e outra da mesma.

O equipamento deve ser resistente a chuvas e exposição ao sol, possuindo desempenho em temperaturas entre 8 e 40°C.

Por estar em contato direto com animais selvagens, o formato e coloração, não podem chamar atenção e deve influenciar o mínimo possível no habitat e na rotina dos animais.

A câmera deve ser capaz de registrar imagens de animais em até 8 metros de distância.

O modelo a ser proposto irá auxiliar no monitoramento do fluxo e na detecção de novas espécies de animais de grande e médio porte presentes no local como mencionado, e, como diferencial, é proposto o envio das imagens capturadas por rede wi-fi que será instalada no JBB.

IV. DESCRIÇÃO DE HARDWARE

Visando atender os requisitos e sanar as dificuldades existentes, está sendo construído um protótipo a partir do microcontrolador Raspberry Pi 3, associado a um sensor infravermelho e uma câmera modelo Sleh-0048 da marca Sony, alimentados por uma bateria de lítio, com 2200mAh e 11.1V. Esse sistema irá realizar a captura de imagens dos animais que passarem no local. A utilização desse microcontrolador se dá devido a necessidade de permitir o envio de dados remotamente por rede wi-fi.

A tabela abaixo apresenta a lista de materiais utilizados e suas descrições.

Tabela de materiais	
Item	Descrição
Raspberry Pi 3	- Modelo B, rede sem fio BMC43438.
Câmera Sony modelo Sleh-0048	- 720p - 120 quadros por segundo
Caixa de proteção	- Material: Papel Panamá
Cartão de memória	- 32 gigas.
Sensor de presença	- Ângulo de detecção: 140° - Alcance: até 7m.
Bateria	- Lítio: 2200mAh 11.1v
Regulador de tensão	- 5V

A ligação entre a raspberry, o sensor de presença, a câmera e a bateria é mostrado no esquemático da Fig.1.

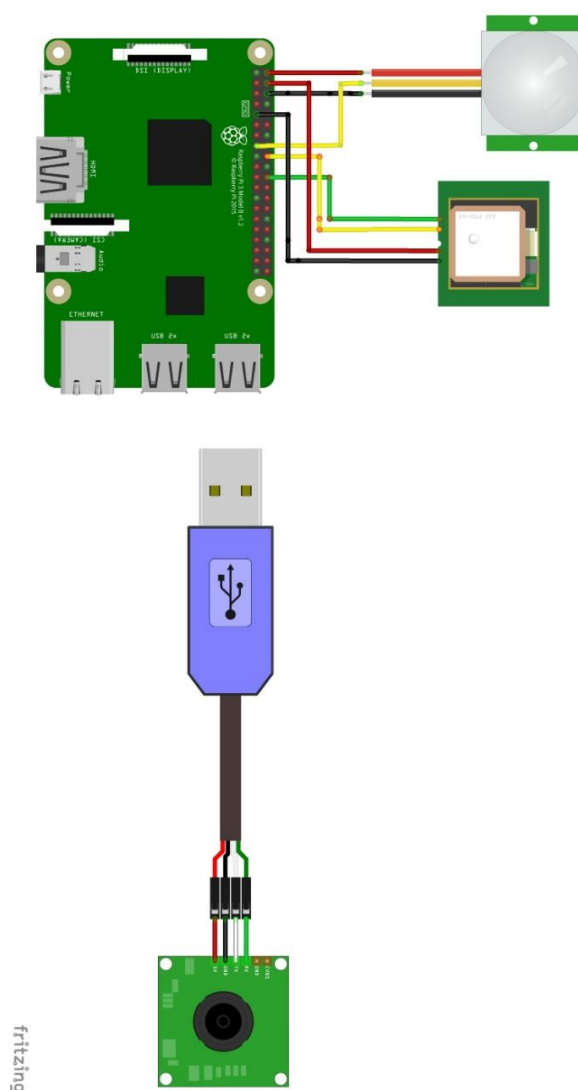


Fig.1. Esquemático de ligação de hardware.

Como pode ser observado, a câmera é conectada à raspberry via cabo USB, enquanto os demais componentes, são conectados às portas digitais da mesma.

Devido a possibilidade de contato com água da chuva, está sendo impressa em impressora 3D uma caixa tamanho 21x15x7 [cm], que posteriormente será vedada com silicone a fim de proteger o hardware. O desenho da caixa foi realizado no software CATIA. (Fig.2, 3 e 4).

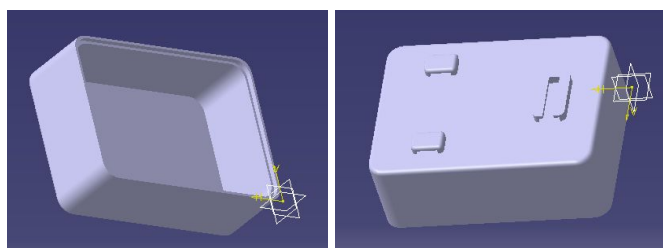


Fig.2 Fundo caixa

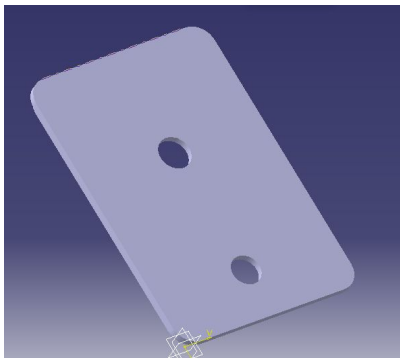


Fig.3 Tampa da caixa

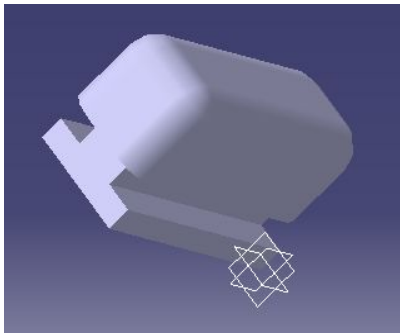


Fig.4 Peça para ajuste de inclinação

Em sua tampa (Fig. 3) foram colocados dois orifícios para passagem da lente da câmera e do sensor de presença. Além disso, a caixa conta com duas alças em seu fundo (Fig. 2), para facilitar a sua acoplagens às árvores onde será colocada, e uma peça removível para regulagem de inclinação apresentada na Fig. 4.

V. DESCRIÇÃO DE SOFTWARE

No desenvolvimento deste projeto utilizou-se a linguagem C para descrever seu funcionamento, onde foram utilizados os pinos GPIO da raspberry PI3 a partir da biblioteca <wiringPi.h>. Com essa biblioteca foi possível realizar a leitura do sensor de presença, e detectar os movimentos. Como a bateria de lítio utilizada tem duração aproximada de 8 horas e não pode ser descarregada completamente a fim de prolongar sua vida útil foi necessário desligar a raspberry após 6 horas de funcionamento, deixando uma margem grande de segurança. Pode-se ver o fragmento de código responsável por desligar a raspberry abaixo.

```
while(1){
    sleep(1);
    desl++;
    if(desl==21600){
        printf("\ndesligando...\n");
        system("sudo shutdown -h now");
    }
}
```

A captura e envio são realizadas por dois processos separados que rodam simultaneamente.

Os processos descritos abaixo foram criados a partir da biblioteca <pthread.h>.

```
pthread_create (&thread_1, NULL, &verif_captura, NULL);
pthread_create (&thread_2, NULL, &send_data, NULL);
```

O processo de captura é feito quando ocorre a detecção de presença que dispara um contador, *number*, e a cada imagem capturada esse contador é incrementado, sendo esse incremento o identificador de cada imagem obtida.

Assim, após a captura, o comando *fswebcam* “nome da imagem a ser criada” é enviado para o terminal a partir do comando *system*, que realiza a captura:

```
number++;
strcpy(captura,"fswebcam ");
sprintf(num,"imagem%d.jpg",number);
strcat(captura,num);
system(captura);
```

No envio das imagens para nuvem, foi utilizada a biblioteca *Dropbox-Uploader* que faz o *upload* das imagens para o Dropbox.

O envio ocorre quando há o acumulo de 5 imagens no repositório */home/pi/gabriel/Drop/Dropbox-Uploader/*, sendo que após a captura dessas imagens o comando *popen* é acionado e começa o envio. Essa função insere o comando no terminal da raspberry assim como a função *system*, no entanto com a função *popen* é possível obter o resultado do comando.

```
while(1){
    if (next_number<=number){
        for(int indi=before_number;indi<=next_number;indi++){

            strcpy(upload,"./dropbox_uploader.sh upload
/home/pi/gabriel/Drop/Dropbox-Uploader/");
            sprintf(num,"imagem%d.jpg imagem%d.jpg",indi,indi);
            strcat(upload,num);

            //enviando imagem
            ptr1 = popen (upload,"r");
            sleep(2);
            fgets (sucess, 300 , ptr1);
            pclose (ptr1);

            //verificando se envio foi bem sucedido
            verifica_envio(indi);
        }
        before_number = next_number +1;
        next_number +=10;
    }
}
```

Ao solicitar o envio do arquivo para o dropbox, a partir da mensagem de retorno da função *popen* é possível verificar se o envio foi realizado ou não.

Essa verificação é feita a partir da função *verifica_envio()*, que compara se a string de retorno do envio da imagem possui a palavra DONE, retornada quando o envio é concluído.

Se o envio for realizado com sucesso a imagem que foi enviada é excluída do repositório da raspberry.

VI. RESULTADOS

O tempo entre a captura de uma foto e outra, que no início do projeto era de 5 segundos, foi diminuído para 2,5 segundos, enquanto o sensor de presença está acionado.

As imagens são nomeadas como “ImagemX.jpg”, onde X um número sequencial determinado pela ordem de captura das mesmas. A nomeação foi feita neste modelo para evitar que uma imagem sobrescreva a outra.

Durante os testes realizados, observou-se que o programa implementado gerava conflito entre a numeração de captura e de envio das imagens, pois eventualmente eram tiradas mais de uma foto enquanto o upload para o Dropbox era realizado. Assim, quando inicializado o envio da imagem seguinte, o programa contava apenas com a última capturada, não enviando as que estavam no intervalo entre a última enviada e a última capturada. Este problema foi solucionado verificando o envio de cada imagem em ordem numérica.

Além disso, o sistema espera ter 5 capturas para só então enviá-las.

A câmera foi instalada no JBB com o auxílio de um dos responsáveis pela reserva, a fim de fazer testes acerca da distância necessária para detecção pelo sensor e a qualidade das imagens sobre as condições de iluminação do local. Por precaução, foi programado para a raspberry desligar em 7,5 horas para evitar a descarga total da bateria de lítio.

O teste foi realizado no dia 5 de junho de 2019 entre 16h e 23:40h, aproximadamente, e, durante esse período, foram realizados disparos continuamente.

A causa atribuída foi a existência falhas na estrutura na continuidade entre as ligações de hardware. Tal falha foi resolvida com a migração das conexões entre a bateria e a raspberry para uma placa de circuito impresso perfurada (Fig. 5). Os testes realizados testes novamente foram satisfatórios e não apresentaram mais o problema mencionado.



Fig.5. Placa de circuito impresso para ligação de hardware.

A possibilidade de interferência por raios solares é totalmente descartada, uma vez que, mesmo o sensor sendo o infravermelho, o problema se manteve no período noturno. Além disso, a copa das árvores impedia que tais raios incidirem diretamente no sensor.

Os testes foram úteis para conhecer o ambiente a ser trabalhado e as reais dimensões e formas necessária para a nova estrutura para armazenamento do hardware, o que gerou a necessidade da construção da caixa com as especificações mencionadas anteriormente em plástico ABS com uma impressora 3D. Foi possível estimar também o horário em que é viável utilizar este modelo de câmera, uma vez que, a partir das 19h já ficou impossível visualizar as imagens sem que seja feito um processamento de imagens.

As imagens tiradas durante o teste foram corrompidas devido a manipulações inadequadas ao passar os dados do cartão SD para um dispositivo USB. No entanto, a partir de um software de recuperação foi possível obter as figuras 6, 7 e 8.

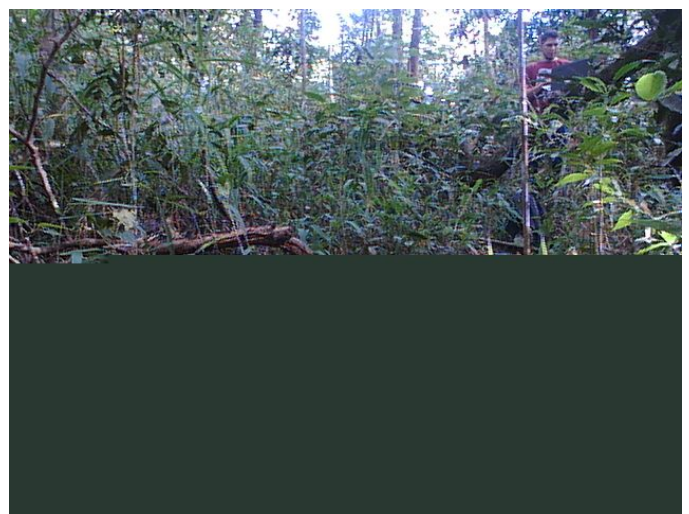


Fig.6. Fig.7. Imagem capturada durante instalação da câmera trap no JBB (1)

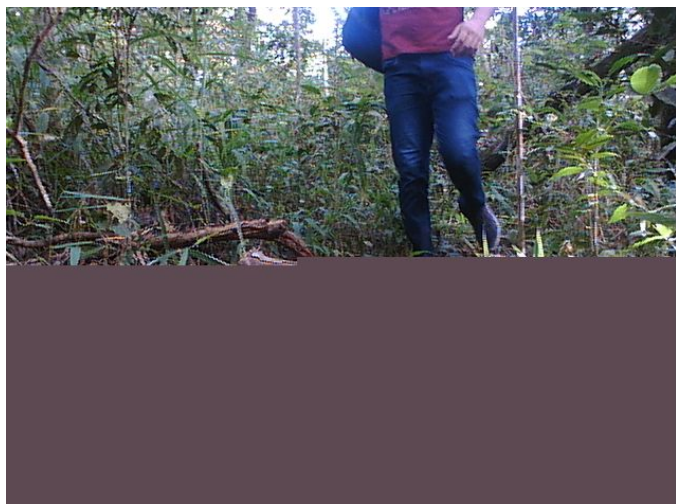


Fig.7. Imagem capturada durante instalação da câmera trap no JBB (2).



Fig.8. Imagem capturada durante instalação da câmera trap no JBB (3).

Nas imagens, apesar de não ser visto, são gravadas a data e hora em que foram tiradas, como na imagem 9, que mostra a imagem noturna às 22:24h.

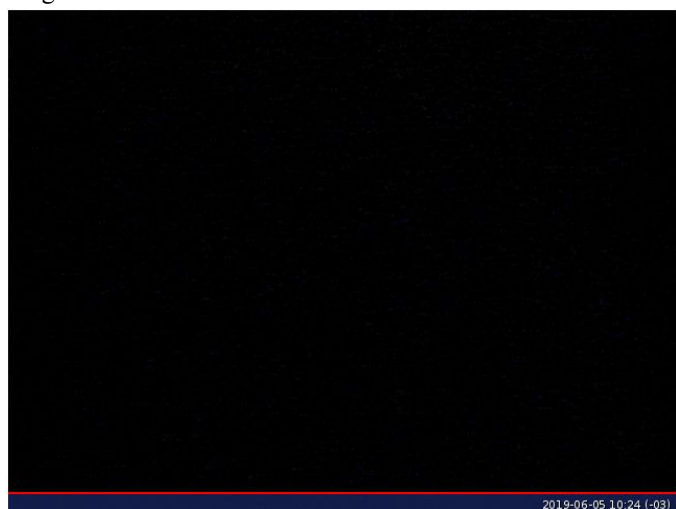


Fig.9. Testando sensor de presença.

CONCLUSÃO

O projeto aqui descrito visa diminuir os custos de aquisição da câmera trap evitar a perda de imagens capturadas e armazenadas em cartão usb por câmeras trap, seja por danos no material ou por extravio do mesmo, um problema frequentemente enfrentado pelo Jardim Botânico de Brasília. A solução proposta foi criar uma câmera trap capaz de enviar as fotografias via wi-fi para a nuvem evitando assim a perda desse material.

Apesar de ainda está em andamento, o projeto já tem resultados satisfatórios quanto a captura e envio de imagens para o dropbox, sendo esta captura realizada a cada 2,5 segundos, sempre que o detector de presença estiver acionado. Os testes de autonomia da bateria realizados após retirar alguns aplicativos e interface gráfica da raspberry Pi3, foram de cerca de 12 horas, e o teste em campo (no JBB) teve duração de 7:40h. E, apesar de não ter sido possível realizar o registro de nenhum animal, foi de extrema importância para que possam ser feitos ajuste no tempo de inicialização de disparos e na estrutura física da câmera trap.

Novos testes serão realizados após as melhorias realizadas. Como continuidade do projeto, pretende-se fazer a substituição da raspberry Pi3 por uma raspberry Pi0, a fim de diminuir o consumo de energia e aumentar o intervalo entre a reposição da bateria em campo. Também será incluída uma câmera noturna para viabilizar fotos noturnas e um sensor de luminosidade para fazer a troca automática das câmeras.

REFERÊNCIAS

- [1] V. M. Rosana, D. M. Fábio. "A UTILIZAÇÃO DE ARMADILHAS FOTOGRÁFICAS PARA O ESTUDO DE MAMÍFEROS DE MÉDIO E GRANDE PORTE". USPRS
- [2] P. C. Fabrício. "MONITORAMENTO DE MAMÍFEROS TERRESTRES DE MÉDIO E GRANDE PORTE". CENAP/ICMBio. Atibaia-São Paulo, Fevereiro 2013

ANEXO 1 - CÓDIGO PARA RASPBERRY

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <wiringPi.h>
#include <pthread.h>

#define sensor 3
#define LED 0

int number =0;
FILE *ptr1;
char sucess[300];

void verifica_envio(int indi){

char envio[5]="DONE";
int tam_envio,tam_sucess,tam,aux=0,x=0;

tam_envio=strlen(envio);
tam_sucess=strlen(sucess);

for(tam=0;tam<tam_sucess;tam++){
    if(envio[0]==sucess[tam])
    {
        aux=tam;
        x=0;
        for(int j=0;j<tam_envio;j++){
            if(envio[j]==sucess[aux]){x++;}
            aux++;
        }
        if(x>=tam_envio){
            printf("\nSucesso no envio!\nApagando
            imagem%d.jpg\n",indi);

            //apagar imagem que foram enviadas
            char image[50];
            sprintf(image,"imagem%d.jpg",indi);
            char remove[200]="sudo rm
/home/pi/gabriel/Drop/Dropbox-Uploader/";
            strcat(remove,image);
            printf("\n%s\n",remove);
        }
        else{printf("\nAinda não foi enviada!!!\n");
            verifica_envio(indi);
        }
    }
}

void* verif_captura(){
char num[30],captura[50];
int quant_cap=3;
delay(3000);
```

```
while(1){
    if(digitalRead(sensor)==1){

        number++;
        strcpy(captura,"fswebcam ");

        sprintf(num,"imagem%d.jpg",number);

        strcat(captura,num);
        printf("%s",captura);
        system(captura);
        printf("\n\n O valor dasd de n é:
        %d\n\n",number);
        delay(2000);}}
return NULL;
}

void* send_data(){
    char upload[300],num[30];
    int next_number = 3, before_number=1;

    while(1){
        if (next_number<=number){
            for(int
            indi=before_number;indi<=next_number;indi++){

                printf("\n\n O valor de indi é: %d\n\n",indi);
                strcpy(upload,"./dropbox_uploader.sh
upload /home/pi/gabriel/Drop/Dropbox-Uploader/");
                sprintf(num,"imagem%d.jpg
imagem%d.jpg",indi,indi);
                printf("\n%s\n",num);
                strcat(upload,num);
                printf("\n%s\n",upload);

                //enviando imagem
                ptr1 = popen (upload,"r");
                sleep(2);
                fgets (sucess, 300 , ptr1);
                printf("\nchegou aqui\n");
                pclose (ptr1);

                //verificando se envio foi bem sucedido
                verifica_envio(indi);

            }

            before_number = next_number +1;
            next_number +=3;

        }
    }
return NULL;
}

int main(void){
    int num2;
```

```

wiringPiSetup(); // inicia a biblioteca WiringPi

pthread_t thread_1;
pthread_t thread_2;

pinMode(sensor,INPUT); // configura o pino 3 como
entrada
pullUpDnControl(sensor, PUD_DOWN);
pinMode(LED,OUTPUT);

//esperando tempo de calibração do sensor PIR
for(int i=0;i<40;i++){
    puts(".");
    delay(100);
}

puts("\nsensor calibrado\n");
delay(500);

//Criando processos para capturar e enciar dados
pthread_create (&thread_1, NULL, &verif_captura,
NULL);
pthread_create (&thread_2, NULL, &send_data, NULL);

int desl=0;
while(1){
    if((digitalRead(sensor))==1){
        ptr2 = popen ("date +%r", "r");
        fgets (horario, 30 , ptr2);
        pclose (ptr2);
        desl++;
        sleep(1);
        printf("\n O horário da imagem é:
%s\n",horario);
    }

    desl++;
    sleep(1);

    if(desl==21600){ //desliga após
aproximadamente 6 horas
        printf("\ndesligando...\n");
        system("sudo shutdown -h now");
    }
}
return 0;
}

```