

Comedouro eletrônico de dietas restritivas para animais com sobrepeso

Danie Auler
Universidade de Brasília
Faculdade Gama
Brasília, Brasil
danielauler7@gmail.com

Paulo Franklin Gomes
Universidade de Brasília
Faculdade Gama
Brasília, Brasil
pgomespereiradeoliveira@gmail.com

I. JUSTIFICATIVA

Em se tratando do século XXI, muitas pessoas que moram em apartamentos na zona urbana, possuem animais de estimação como gatos e cachorros. Muitos destes animais sofrem por ter sobrepeso, pois eles acabam se alimentando de maneira muito rápida, o que corrobora para o estresse desses animais, além do peso exacerbado[1].

A ideia de construir um comedouro eletrônico de dieta restritiva para animais com sobrepeso surgiu a partir de um problema vivenciado por um cachorro que mora perto da minha casa. Este possui um problema intestinal e, devido a isso, ele deve usufruir de uma dieta balanceada e cadenciada em horários previamente determinados.



Figura 1. Bichinho de estimação se alimentando rapidamente

Contudo, uma solução encontrada para contornar esse problema é projetar um sistema que é capaz, em horários previamente determinados, enviar uma foto da tigela do animal, via o aplicativo do telegram, que terá como função verificar o quanto de ração este comeu, bem como ter a funcionalidade de cancelar o bastecimento de ração na tigela do bichinho, caso esta esteja cheia no momento. Caso contrário, o sistema liberará aos poucos a quantidade de comida necessária para alimentar o animalzinho.

II. OBJETIVOS

Criação de um comedouro eletrônico que na hora específica irá enviar para o usuário via telegram uma foto com a tigela de comida para que o mesmo saiba o quando da ração o animal comeu. Caso o usuário não cancele o acionamento da tigela, a mesma irá liberar aos poucos ou de uma vez (escolha

do usuário) a quantidade de ração necessária nos horários especificados.

III. REQUISITOS

O comedouro será equipado com um raspberry pi zero conectado a internet e conectado a uma câmera apontada para a tigela do animal. Além disso, o comedouro contará com um sensor de presença ultrassônico para saber se o animal comeu a porção parcial. O rasp fará requisições http para o bot para avisar o usuário sobre as ações que aconteceram e estará com um server aberto para ouvir as requisições do bot quando o usuário quiser inserir um comando de intervenção ou ajuste nos horários de alimentação.

IV. BENEFÍCIOS

O projeto tem como principal beneficiário o dono de animais com transtorno de ansiedade ou alguma doença que necessite de restrição alimentar com horários especificados. Em consequência disso, temos como beneficiário também pessoas que deixam o animal de estimação sozinho em casa e não querem deixar um pote muito grande com muita ração parada para que o animal consiga se alimentar ao longo do dia.

V. SERVO MOTOR

O servo motor é muito utilizado em projetos de automação industrial, onde de fato é necessário um controle de precisão maior em relação ao sistema que está sendo projetado. Outrora, imaginava-se que este só serviria para aplicações que envolvessem apenas projetos relacionados ao controle preciso da direção do torque, além da velocidade e da posição. Entretanto, este dispositivo vem sendo uma alternativa muito boa para substituir motores convencionais tais como, indução e atuadores pneumáticos e hidráulicos.

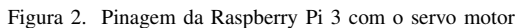
Todavia de fato esta é uma máquina, eletromecânica, que apresenta movimento proporcional a um comando, como dispositivo de malha fechada, ou seja: recebe um sinal de controle que verifica a posição atual para controlar o seu movimento indo para a posição desejada com velocidade monitorada externamente sob feedback de um dispositivo denominado taco ou sensor de efeito Hall ou encoder ou resolver, ou tachsin, dependendo do tipo de servomotor e aplicação.

Para o ponto de controle 2 foi desenvolvido um código em C, que pode ser observado no apêndice A, com o objetivo

Logo, abaixo é mostrado como foi feito a conexão entre os pinos do servo motor e da Raspberry Pi 3.

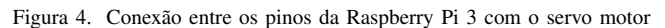


Pin No.	
1	3.3V
2	5V



Com objetivo de conectar o usuário ao comedouro foi feita uma plataforma do telegram com sua API aberta e bem documentada, bem como um fluxograma demonstrando como é feita essa conexão entre o usuário, o bot do telegram e o comedouro (figura 3). Para testarmos essa conexão foi realizado um teste com foco de enviar um comando via o bot do telegram, no qual este acende um led com uso da raspberry, além de verificar o recebimento desse comando. Tal código pode ser visualizado no apêndice B em anexo.

- [1] dicas para auxiliar o seu cachorro comer-mais devagar., 2018. Disponível em: <<https://www.portaldodog.com.br/cachorros/adultos-cachorros/alimentacao-adulto/dicas-para-auxiliar-o-seu-cachorro-comer-mais-devagar>>. Acesso em: 29 set. 2019.
- [2] SERVO Motor SG90 Datasheet. [S. l.], 2014. Disponível em: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf. Acesso em: 6 out. 2019.
- [3] OS BOTS do Telegram estão mais inteligentes. [S. l.], 2016. Disponível em: <https://tecnoblog.net/194163/telegram-bots-2/>. Acesso em: 6 out. 2019.



- Apêndice A: Código em C para controlar o servo motor:

```
int main(void)
{
    int pin = 7, i;
    wiringPiSetup();
    pinMode(pin, OUTPUT);
    for(i=0; i<5; i++)
    {
```

```

motor(pin, 500, 20000-500, 25);
motor(pin, 2500, 20000-2500, 25);
}
return 0;
}

```

Apêndice B: Código em python para controlar a Raspberry Pi 3 com o bot do telegram:

```

import time, datetime
import RPi.GPIO as GPIO
import telepot
from telepot.loop import MessageLoop

led = 26
now = datetime.datetime.now()
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
#LED
GPIO.setup(led, GPIO.OUT)
GPIO.output(led, 0) #Off initially

def action(msg):
    chat_id = msg['chat']['id']
    command = msg['text']

    print 'Received: %s' % command
    if 'on' in command:
        message = "Turned on "
        if 'led' in command:
            message = message + "led"
            GPIO.output(led, 1)
            telegram_bot.sendMessage
            (chat_id, message)

    if 'off' in command:
        message = "Turned off "
        if 'led' in command:
            message = message + "led "
            GPIO.output(led, 0)
            telegram_bot.sendMessage
            (chat_id, message)

telegram_bot = telepot.Bot('Meu token')
print (telegram_bot.getMe())

MessageLoop(telegram_bot, action)
.run_as_thread()
print 'Up and Running....'

while 1:
    time.sleep(10)

```