Alimentador automático para animais utilizando Raspberry PI

Bruno Deivid Mendes Costa - 15/0120214

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
email: costa.eeunb@gmail.com

Lucas Luan Araújo Barbosa - 13/0122173

Programa de Engenharia Eletrônica
Faculdade Gama - Universidade de Brasília
email: lucasluan.fga@gmail.com

1. JUSTIFICATIVA

Mesmo em tempos de tecnologia avançada e "virtualização" das relações, nota-se que a busca do ser humano por algum animal de estimação continua crescente, seja como companhia, auxílio a pessoas com alguma deficiência ou com intuito terapêutico como no caso de tratamento de crianças. [2]
A crescente de tal relação é prejudicada pelo ritmo intenso do cotidiano, principalmente em grandes cidades, sendo que os donos não dispõe de todo o tempo que seria necessário para cuidar de seus animais. Tal situação, aliada ao desenvolvimento tecnológico constante, faz surgir o mercado de produtos específicos para animais de estimação.
A proposta em questão é de um alimentador para cães e gatos (que comporta água ou ração) utilizando uma *Raspberry Pi*, em que, mesmo a distância o dono pode definir o horário de alimentação de seu "amigo" e até mesmo observar como está o mesmo através de uma câmera acessada de qualquer navegador (característica que diferencia o projeto de qualquer outro que exista no mercado), proporcionando assim, maior comodidade para os donos e maior conforto para os animais.

2. OBJETIVOS

O objetivo do projeto é realizar o desenvolvimento de um alimentador automático para animais de estimação domésticos, sendo que por meio deste protótipo, haverá a possibilidade de controle dos horários para liberação da comida assim como o monitoramento remoto por meio de uma câmera para ver se o animal está se alimentando adequadamente.

3. REQUISITOS

O protótipo do alimentador automático terá um reservatório onde será armazenada a ração a ser disponibilizada para o animal e também com o intuito de não precisar repor frequentemente tal recipiente possibilitando que o dono do animal fique ausente por um bom tempo sem se preocupar se há ou não ração suficiente pro seu pet. Haverá um recipiente menor onde será fornecido diretamente para o animal uma quantia considerável de ração setada pelo usuário.

O protótipo realizará as seguintes funcionalidades básicas:

- Programação de horários para disponibilizar automaticamente a ração para o animal;
- Monitoramento remoto do protótipo por meio de uma câmera;

4. BENEFÍCIOS

Esse projeto traz diversos benefícios relacionados à sua funcionalidade ou mesmo em seu modo de operação.
Vale ressaltar que o controle operacional do equipamento por parte do usuário dar-se-á por meio de uma interface web, fator facilitador para grande parte da população, que está acostumada a usar sistemas semelhantes em redes sociais e outros sites, não havendo portanto estranheza por parte dos mesmos.
A utilização da Raspberry Pi é outro fator que chama a atenção para o projeto, já que o uso da mesma proporciona um aumento considerável de performance na velocidade de leitura e armazenamento dos dados quando comparado ao desempenho que teria um micro controlador. [1]

5. Desenvolvimento

5.1 A execução do projeto se iniciou com a elaboração do código para o controle do motor de passo que irá fazer o deslocamento do reservatório de ração a fim de que seja depositada a porção para o animal. O desenvolvimento do código começou em linguagem C, entretanto, não foi concluído a tempo para a apresentação do projeto. Com isso, foi desenvolvido um código em Python, visto que é uma linguagem mais prática de se programar e no caso desse projeto o fato de tratar-se de uma linguagem interpretada não seria problema.
O código em Python utilizado foi:

C:\Users\Bruno Costa\Desktop\Lalala\Motor_passo.py - Sublime Text (UNR

File Edit Selection Find View Goto Tools Project Preferences

Motor_passo.py

```
#Carrega bibliotecas
import sys
import time
import RPi.GPIO as GPIO

#Utiliza numeros da GPIO ao inves
#da numeracao dos pinos
GPIO.setmode(GPIO.BCM)

#Pinos de conexao ao motor
#Pinos 40, 38, 36, 32
#GPIO21,GPIO20,GPIO16,GPIO12
StepPins = [21,20,16,12]

#Define os pinos como saida
for pin in StepPins:
  print "Setup pins"
  GPIO.setup(pin,GPIO.OUT)
  GPIO.output(pin, False)

#Sequencia de ativacao
Seq = [[1,0,0,0],
       [1,1,0,0],
       [0,1,0,0],
       [0,1,1,0],
       [0,0,1,0],
       [0,0,1,1],
       [0,0,0,1],
       [1,0,0,1]]

StepCount = len(Seq)-1

#Configura sentido de giro
StepDir = 2 # 1 ou 2 para sentido horario
            # -1 ou-2 para sentido anti-horario

#Tempo de espera
WaitTime = 1/float(1000)

#Inicializa variaveis
StepCounter = 0

while True:
  #Movimenta o motor e envia os dados de ativacao
  #para o display
  for pin in range(0, 4):
    xpin = StepPins[pin]
    print StepCounter
    print pin
    if Seq[StepCounter][pin]!=0:
```

Figura 1- Código para controle de motor de passo.

```
while True:
  #Movimenta o motor e envia os dados de ativacao
  #para o display
  for pin in range(0, 4):
    xpin = StepPins[pin]
    print StepCounter
    print pin
    if Seq[StepCounter][pin]!=0:
      print " Step %i Enable %i" %(StepCounter,xpin)
      GPIO.output(xpin, True)
    else:
      GPIO.output(xpin, False)

  StepCounter += StepDir

  #Ao final da sequencia, reinicia o processo
  if (StepCounter>=StepCount):
    StepCounter = 0
  if (StepCounter<0):
    StepCounter = StepCount

  #Delay para movimentar o motor
  time.sleep(WaitTime)
```

Line 65, Column 23

Figura 2-Cont.Cdigo para controle de motor de passo.

5.2 Após isso, foi feita a parte de configuração da câmera por meio do Motion. Editando o arquivo “config” do Motion foi possível utilizar a raspberry pi como um servidor de imagem, bastando apenas acessar seu ip.

5.3 Tendo as duas principais partes já prontas (câmera e motor),foi feita então a parte de agendamento de tarefas. Para isso foi elaborado um código em linguagem c baseado no código abaixo:

C:\Users\Bruno Costa\Desktop\Lalala\agenda.ino • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

agenda.ino

```
void sincTempo(relogio_t relogio)
{
    setTime( relogio.hora,
             relogio.minuto,
             relogio.segundo,
             relogio.dia,
             relogio.mes,
             relogio.ano );

    tempoSinc = true;
}

void agendarDespejo(diretriz_t d)
{
  AlarmId id = Alarm.alarmRepeat(d.inicio_hora,d.inicio_minuto,0, alimentapets);

  if(id == 255)
    Serial.println(F("Agendamento não-registrado"));
  else {
    if(id == dtNBR_ALARMS-1)
      Serial.println(F("Limite de agendamentos estourado!"));
    qtdRacao[id] = d.qtd;
    Serial.print(F("Despejo agendado!   alarmID: "));
    Serial.println(id);
  }
  Serial.println("");
}

void printDiretriz(diretriz_t diretriz)
{
  Serial.print(F("  RX:"));
  Serial.print(F("  A-ID:"));
  Serial.print(diretriz.alimentID);
  Serial.print(F("  iniH:"));
  Serial.print(diretriz.inicio_hora);
  Serial.print(F("  iniM:"));
  Serial.print(diretriz.inicio_minuto);
  //Serial.print("  Freq:");
  //Serial.print(diretriz.frequencia);
  Serial.print(F("  qtd:"));
  Serial.println(diretriz.qtd);
}

void printTempo(relogio_t relogio)
{
  Serial.print(F("  RX - Tempo: "));
  Serial.print(relogio.hora);
  Serial.print(F(":"));
  Serial.print(relogio.minuto);
  Serial.print(F(":"));
  Serial.print(relogio.segundo);
  Serial.print(F("      dia"));
  Serial.print(relogio.dia);
  Serial.print(F("/"));
  Serial.print(relogio.mes);
  Serial.print(F("/"));
  Serial.println(relogio.ano);
}
```

Line 25, Column 54

Figura 3- Código para agendamento de tarefas

Entretanto,ao verificar que a raspberry já possui um sistema mais simples para este fim, foi decidido que o utilizaríamos , trata-se do recurso crontab, que permite agendar várias tarefas em dias e horários desejados.

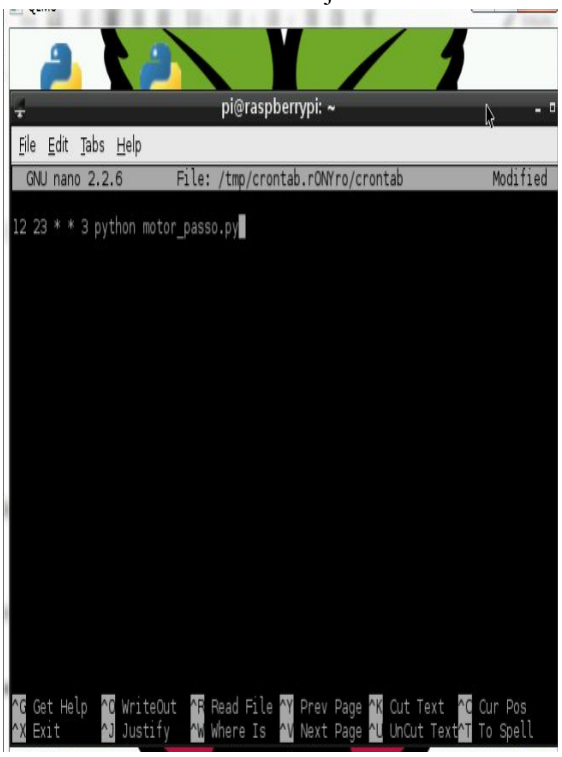

Figura 4-Edição do arquivo crontab.

5.4 Com o agendamento feito,passou-se para os detalhes finais,entre eles o acesso remoto que foi feito via VNC com o aplicativo VNC Viwer. E finalmente, a construção da maquete do projeto para apresentação.

6. VISÃO GERAL DO SISTEMA

5.1 Diagrama de Blocos

O diagrama mostrado abaixo descreve a comunicação dos periféricos e o processador Raspberry Pi.

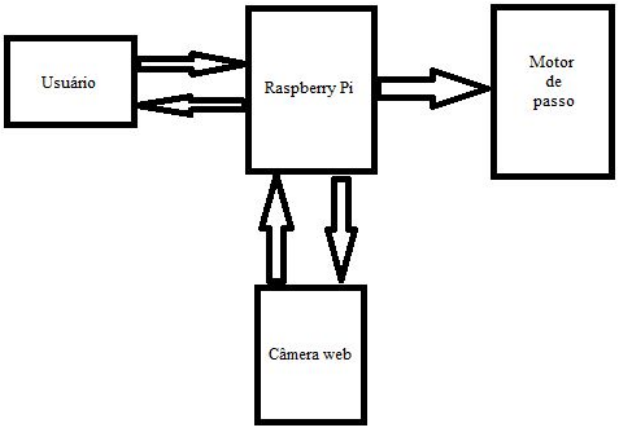

Figura 5: Diagrama de blocos.

O Raspberry Pi programa certos horários do dia para que o motor de passo funcione a um certo passo de forma a liberar a quantia de ração correta referente a uma alimentação por uma comporta ao recipiente pelo qual o pet irá se alimentar. O monitoramento remoto funciona de forma que o usuário solicita acesso à câmera pela web por meio do raspberry, assim, o mesmo poderá verificar se o pet está se alimentando corretamente.

5.2 Fluxograma

O fluxograma mostrado abaixo descreve as etapas de funcionamento do projeto.

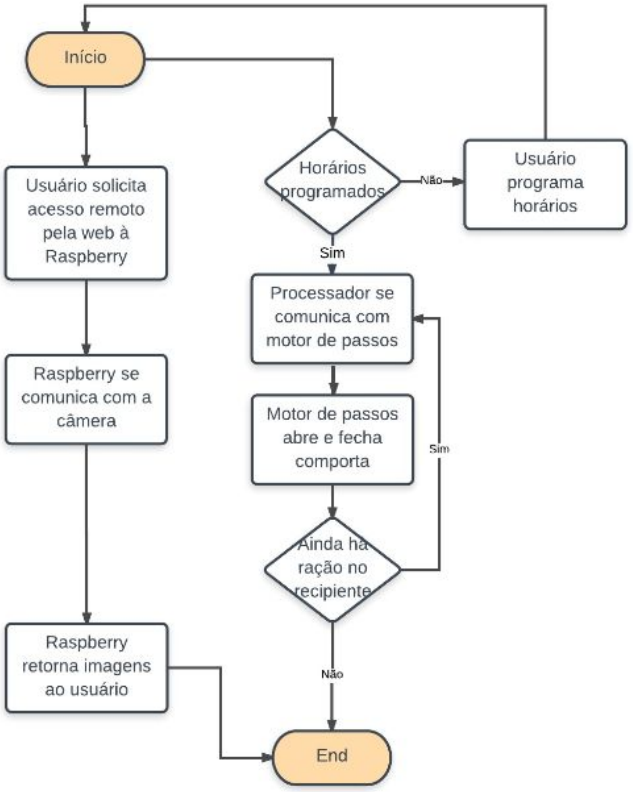

Figura 6: Diagrama de funcionamento.

O acesso é feito pelo usuário através do aplicativo VNC Viewr, em que ele terá na tela de seu celular a visualização gráfica da raspberry pi. Nesse, ele poderá editar o arquivo crontab, adicionando novos horários , retirando, ou ligando e desligando a câmera.

5.3 Periféricos

-Raspberry PI 2 Model B

Figura 7- Raspberry pi 3.

A placa oferece um conjunto de portas e interfaces igual ao apresentado na placa antecessora, a Raspberry Pi 1 Model B+: GPIO de 40 pinos, um total de quatro portas USB, interface Ethernet, HDMI, saída de som, interfaces de câmera e tela, além de slot para cartão de memória microSD.

-Webcam C3 Tech WB2102

Figura 8-Câmera usb.

- Botão para foto instantânea
- Suporte para fixação em monitor LCD ou notebook
- Botão Snapshot
- Foco: 5.0 cm a infinito
- Molduras para fotos e efeitos visuais
- Taxa de transmissão: até 30 fps em resolução VGA (640 x 480)
- Resolução de imagem: de 640 x 480 até 6400 x 4800 (máximo)

7. Resultados e discussões

Após a montagem final do projeto , foi notado o correto funcionamento do mesmo, porém não com a praticidade esperada devido ao fato da interface não ser tão intuitiva para o usuário. Segue imagens do alimentador montado:

Figura 9- Maquete do equipamento.

Figura 10- Maquete com raspberry já conectada.

8. Sugestões de melhorias a serem implementadas

Após a conclusão do projeto, pode-se notar alguns aspectos que poderiam ser melhorados. São eles:

- Conclusão e utilização dos códigos em C para controle do motor.
- Criação de um aplicativo que proporcione uma interface mais amigável ao usuário.
- Inclusão de sensores para fornecer informações sobre o ambiente (temperatura,umidade,etc.).

9. REFERÊNCIAS BIBLIOGRÁFICA

[1] OCHAKOWSKI, Nádia. Protótipo de um alimentador automático para animais de estimação. 2007.50 f, il. Trabalho de Conclusão de Curso - (Graduação em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2007.
Disponível em:
< http://www.bc.furb.br/docs/MO/2008/329255_1_1.pdf>
Acesso em: 05 set. 2017

[2] ORSINI, S. Mercado aposta em animais de estimação. [S.l.], 2004.
Disponível em:
<http://financas.cidadeinternet.com.br/article.asp?878~196264>.
Acesso em: 05 set. 2017

[3] ABINPET. População de pets cresce 5% ao ano e Brasil é quarto no ranking mundial. 2012. Disponível em: <http://abinpet.org.br/imprensa/noticias/populacao-de-pets-cresce-5-ao-ano-e-brasil-e-quarto-no-ranking-mundial/> . Acesso em: 01 set. 2017.