

π Quarium

Aquário autônomo controlado por bot de Telegram utilizando Raspberry Pi

Anderson Sales Rodrigues Pinto
Universidade de Brasília
Faculdade Gama
Brasília, Brasil
aandersonsales@gmail.com

Gabriel de Matos Souza
Universidade de Brasília
Faculdade Gama
Brasília, Brasil
gabriel.matos.s@hotmail.com

Resumo—Este trabalho consiste em construir um aquário automatizado, que será responsável, através de vários sensores e atuadores, controlar o ambiente do aquário para que o peixe fique bem acomodado e bem alimentado. O aquário será interfaceado por um bot do Telegram, onde pode-se obter informações relevantes a respeito do aquário, como PH, temperatura, dentre outros.

Index Terms— π Quarium, Smart, Raspberry pi, microcontrolador

I. JUSTIFICATIVA

Embora possa parecer uma tarefa simples e trivial, a observação regular do seu aquário fornece importantes pistas visuais, indicando alterações na qualidade da água e alerta-o de que algo pode estar fora de equilíbrio. Os parâmetros da água fundamentais que influenciam a qualidade da água são o pH, a amônia, o nitrito e o nitrato. No entanto, muitas condições associadas com a má qualidade da água desenvolvem-se gradualmente. Os primeiros sinais podem passar despercebidos, levando a condições mais graves ou mais persistentes.

É preciso que o aquário atenda algumas necessidades como temperatura adequada, iluminação adaptada, pH balanceado, etc. Outro fator correiro é o esquecimento na hora certa de alimentar o(s) peixe(s) e isso pode acarretar no falecimento do animal.

Sabendo destes empecilhos para a criação de peixes foi-se pensado um aquário automatizado, onde será possível cuidar dos peixes de forma mais independente do dono, com cronogramas estabelecidos pelo usuário através de um bot no Telegram.

II. OBJETIVOS

Os objetivos deste projeto estão separados de forma que atendam os seguintes tópicos:

A. Integração de um Bot do Telegram para controle e obtenção de dados do aquário

O π Quarium será desenvolvido para poder verificar algumas configurações remotas de um aquário comum como temperatura e iluminação, e executar certos tipos de rotinas da alimentação. No entanto terá também um modo manual que permitirá tais ações presencialmente.

B. Temperatura

Cada espécie de peixe necessita de uma temperatura da água ideal para ter uma vida saudável, mas, no geral, eles toleram bem a faixa de 21° a 25°C. Conseguindo mantê-la com essa temperatura, é possível se ter um único aquário com várias espécies de peixes. Por ser algo essencial será mensurada através de um sensor e caso haja uma temperatura indesejada, trocar a água pode ser uma solução.

C. pH

A maioria dos peixes de aquário vivem num pH compreendido na faixa de 5,5 a 8,5. Algumas espécies preferem águas ácidas. Outras preferem alcalinas. Outras preferem águas bem perto da neutralidade. Manter o pH dentro da faixa desejada para os peixes nem sempre é uma tarefa muito fácil pois dependemos da água de nossas torneiras para repor a água de nossos aquários e nem sempre seu pH é apropriado. Outra questão a ser levada em consideração é que o aquário, embora seja um sistema fechado, é um local onde diversas reações químicas estão acontecendo e, portanto, novas substâncias estão sendo formadas sendo que algumas delas podem alterar o pH. Medir o pH seria uma forma preventiva para controlar esse problema.

D. Alimentação

Seria possível alimentar o(s) peixe(s) com certa quantidade de comida, de acordo com uma programação diária(rotina), onde o usuário poderá ativa-las através de um Bot do aplicativo.

E. Monitoramento visual a partir de uma câmera

O sistema terá uma câmera para acompanhamento visual do usuário.

III. REQUISITOS

Para que este projeto seja desenvolvido precisa-se de:

- Uma Raspberry Pi3 que será a unidade central de processamento dos dados enviados pelo microcontrolador e responsável pela integração do bot do Telegram e o usuário;
- Um servo motor responsável pela dispensa do alimento;
- Um sensor de temperatura DS18B20 para aferir a temperatura da água;
- Uma fita tipo LED que será utilizado para a iluminação;

- Um sensor de pH que fornecerá os dados de pH;
- Um conversor AD PCF8591, que converterá os dados recebidos pelo sensor de PH em dados digitais.
- Duas bombas de sucção para efetuar a limpeza do aquário.
- Bot de Telegram, que se responsabilizará pela interface entre o proprietário do peixe e o aquário, fornecendo as informações requisitadas pelo proprietário.
- Um aquário que comporte tanto a quantidade de peixes do usuário quanto os equipamentos necessários para deixá-lo autônomo.

Todo o circuito eletrônico deve ficar protegido da umidade do aquário, para isso será construído um case para essa necessidade.

Além disso, o aquário é designado para peixes de água doce e de pequeno porte. Não recomenda-se utilizar o sistema em um aquário pequeno com muitos peixes, pois pode alterar os valores de leitura dos sensores e/ou comprometer o funcionamento dos atuadores.

IV. BENEFÍCIOS

Com este projeto espera-se que o proprietário do(s) peixe(s) terá uma maior autonomia em seu cotidiano, economizando tempo ao cuidar dos peixes e ainda assim poder acompanhar o crescimento deles.

A interface usuário-máquina sendo feita pelo Telegram deixa o projeto mais versátil, pois boa parte das pessoas possuem um celular e/ou um notebook com acesso a internet, assim fazendo com que a interface seja acessível para qualquer pessoa que utilizar o projeto.

V. DESENVOLVIMENTO

A princípio o projeto será devidamente instalado em um aquário que é bastante similar ao da figura 1.



Figura 1. Formato do aquário a ser utilizado

O projeto funciona com a utilização de uma RaspberryPi 3b (figura 2), onde é criado um Bot para monitoramento do animal e para interface com o cliente. Para se conectar com o Bot o usuário deve adicionar o Bot no telegram, que possui o nome de "Peixoso_bot". Ao se conectar com o Bot, o usuário deverá mandar o comando /start para o início da operação. O Bot oferece uma lista de comandos para o usuário escolher,

estas rotinas servem para cada serviço do projeto e estão listadas abaixo.



Figura 2. Raspberry Pi 3B

A. Informações gerais

São obtidas a partir do comando /info, que mostra o potencial de hidrogênio do aquário a partir de um sensor E201C com módulo BNC (figura 3) e a temperatura com o sensor DS18B20 (figura 5).

1) *pH*: O sensor de pH que será utilizado consegue realizar medições de pH na escala de 0 a 14. O sensor leva cerca de 60 segundos para obter uma resposta completa do valor de pH medido. Para realizar a limpeza do eletrodo inerte, basta lava-lo em água corrente, tornando-o apto a novas medições. Para o cálculo do valor do pH, basta colocar o valor analógico obtido na saída PO do módulo na equação 1. O módulo possui as seguintes especificações:

- Corrente: de 5 a 10mA;
- Tensão de aquecimento de 5 ± 0.2 V (AC DC);
- Temperatura de funcionamento de -10°C a 50°C (temperatura nominal de 20°C); e
- Possui um sinal analógico em sua saída.



Figura 3. Sensor de pH

$$PH = (-5.7 \cdot V_{sensor}) + 16.34 \quad (1)$$

Por apresentar dados analógicos em sua saída, foi preciso utilizar um conversor ad PCF8591 (figura 4).

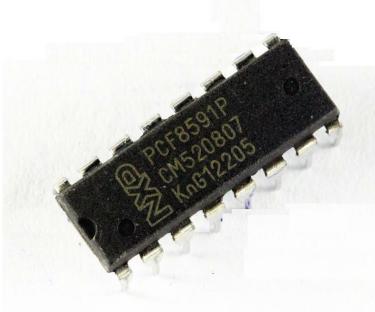


Figura 4. Conversor AD

2) Temperatura: O sensor de temperatura foi escolhido por conta de sua precisão e de sua faixa de operação para controlar a temperatura do aquário. É um termômetro digital que fornece medidas de 9 a 12 bits de resolução. A comunicação com a Raspberry utiliza o protocolo One Wire (Um fio). O sensor possui as seguintes especificações:

- Possui tensão de alimentação na faixa de 3V a 5,5V;
- Corrente de 4mA;
- Faz medidas na faixa de -55°C à 125°C.



Figura 5. Sensor de temperatura

B. Troca de água

É acionada a partir do comando /agua, onde as bombas de sucção (figura 6) irão desempenhar o papel de tirar a água velha e colocar uma água nova previamente disposta em um recipiente pelo usuário. A tarefa é realizada "simultaneamente", onde um processo pai e um processo filho se encarregam de ligar as bombas.



Figura 6. Bomba de sucção

A Raspberry não possui tensão suficiente em seus pinos de saída para acionar as duas bombas, por isso será utilizado um circuito chaveado com MOSFET e protegido por um opto acoplador (figura 7).

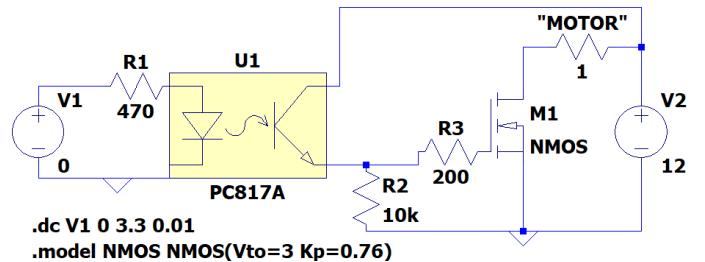


Figura 7. Circuito MOSFET controlado por tensão

O circuito permite que a bomba (que necessita de 12V) seja acionado somente com a tensão de 3.3V proveniente da Raspberry ao se chavear o MOSFET. O opto acoplador serve para isolar o circuito de controle do circuito de potência. A simulação demonstrando a passagem de corrente pode ser vista na figura 8.

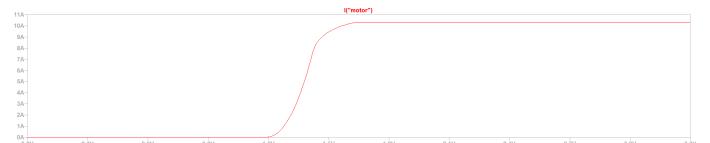


Figura 8. Simulação demonstrando a passagem de corrente

A partir do gráfico nota-se que ao se mandar 0V para o MOSFET, o motor não recebe nenhuma corrente, assim, a partir de 1.3V (nível alto do microcontrolador) a corrente começa a ser liberada, chegando perto de 10.5A quando se manda 3.3V. Isso não diz que o circuito irá utilizar essa corrente, mas afirma que o circuito está apto para a passagem deste valor.

C. Foto

É obtida a partir do comando /foto, que manda uma foto que tirada por uma webcam assim que solicitada.

D. Alimentação

É acionada a partir do comando /comida, onde o servo motor (figura 9) será ativado durante um período determinado para servir a comida para o animal.



Figura 9. Servo motor

O dispositivo de alimentação se encontra na figura 10.

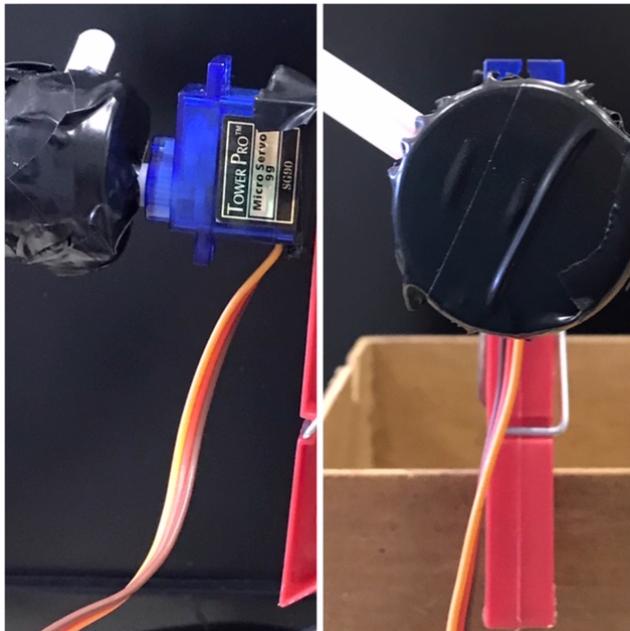


Figura 10. Dispositivo para alimentação

VI. RESULTADOS

Na figura 11 podemos ver os textos enviados pelo Bot ao receber o comando de /info. Neste comando recebe-se os valores de temperatura e de PH do aquário.



Figura 11. Valores de temperatura e PH enviados pelo bot.

Aqui na figura 12 vemos a foto tirada do aquário utilizando o comando /foto.



Figura 12. Foto do aquário capturada pela webcam.

Outro comando que funcionou muito bem foi o de alimentação, que pode ser visto na figura 13

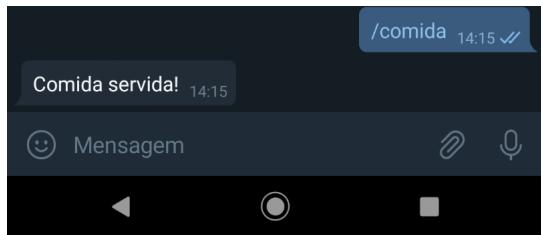


Figura 13. Comando para a alimentação funcionando.

Já nesta figura (14) temos o comando de trocar a água do aquário, que aciona as duas bombas "ao mesmo tempo" por meio de multi-processos.



Figura 14. Comando para troca de água do bot.

Seguem algumas fotos do projeto final, com no que diz respeito a parte de montagem e estrutura.

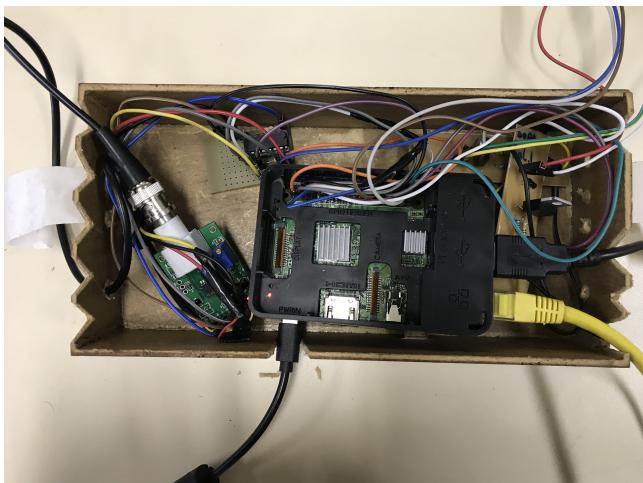


Figura 15. Caixa que guarda a Raspberry e outros circuitos

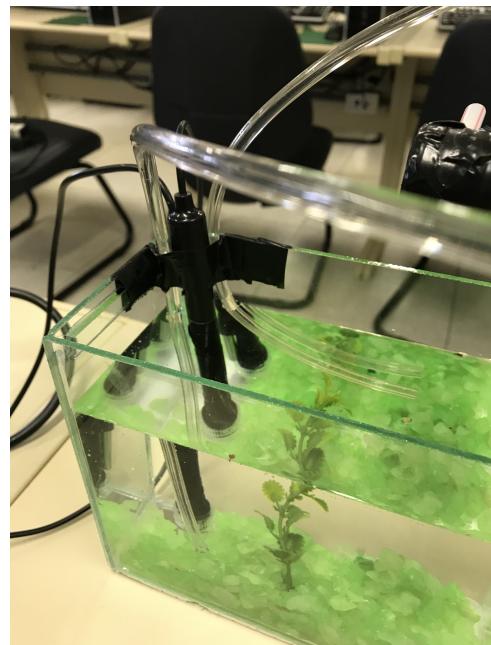


Figura 16. Sensores de temperatura DS18B20 e sensor de pH



Figura 17. Dispositivo de alimentação

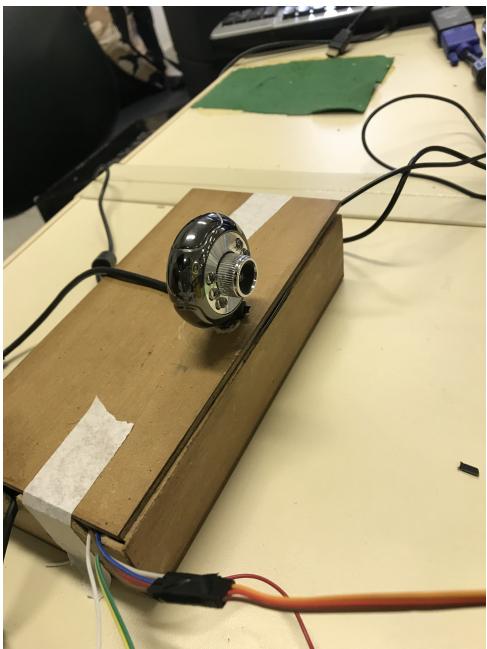


Figura 18. Webcam responsável pela captação das fotos

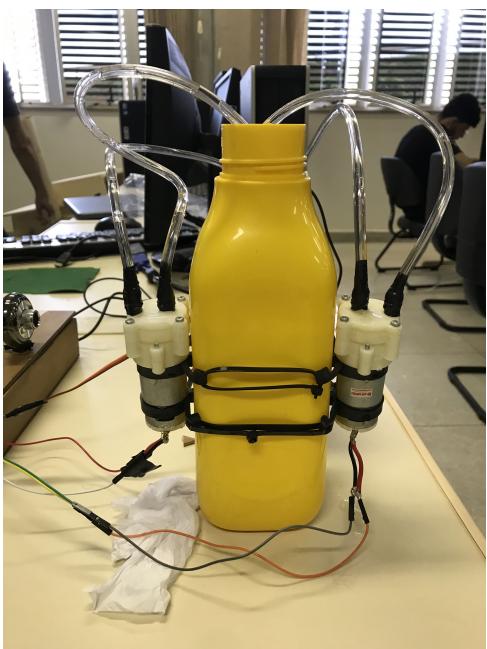


Figura 19. Bombas responsáveis pela troca de água

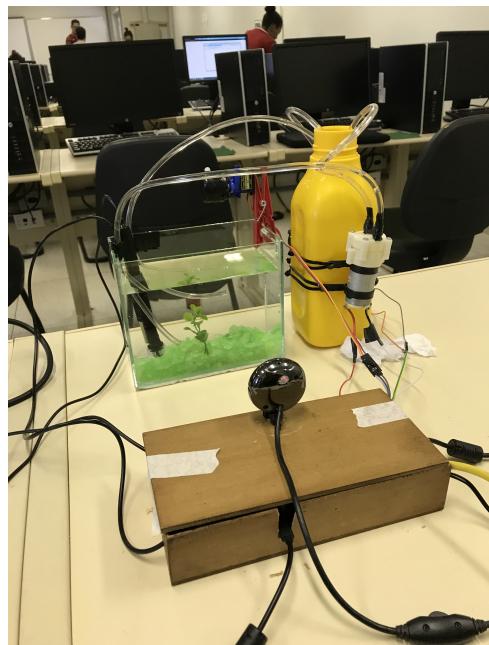


Figura 20. Sistema completo do π Quarium

VII. ANEXOS

Algoritmo 1
CÓDIGO PRINCIPAL

```
#include <csignal>
#include <cstdio>
#include <cstdlib>
#include <exception>
#include <stdio.h>
#include <stdlib.h>
#include "tgbot/tgbot.h"
#include <dirent.h>
#include <string.h>
#include <wiringPi.h>
#include <linux/i2c-dev.h>
#include <pcf8591.h>
#define TOKEN "870108773:
AAG75TI6yFPTl3zSEWxZAp6M-F-sDVHpjPQ"
#define SAIDA 25
#define MOTO1 0
#define MOTO2 2
using namespace std;
using namespace TgBot;
#define BASE 64
#define Address 0x48
#define A0 BASE

void sqwv( int pin , int degree , int N)
{
    int t1 = (100*degree+4)/9+1500;
    int t2 = 20000-t1;
    int i;
    for(i=0; i<N; i++)
    {
        digitalWrite(pin , HIGH);
        usleep(t1);
        digitalWrite(pin , LOW);
        usleep(t2);
    }
}

int main ( int argc , const char **argv )
{
    FILE *fp;
    char teste[20] = "oi";
    char *final;
    wiringPiSetup();
    pcf8591Setup(BASE, Address);
    pinMode(MOTO1, OUTPUT);
    pinMode(MOTO2, OUTPUT);
}
```

```
pinMode(SAIDA, OUTPUT);
char rom[20];
// char* temp_final;

TgBot::Bot bot(TOKEN);
bot.getEvents().onCommand("start", [&bot]
)(TgBot::Message::Ptr message) {
    bot.getApi().sendMessage(
        message->chat->id, "Ol !
Eu sou o bot do Aqu rio!
Segue lista de comandos;\n
/temperatura: mostra a
temperatura do aqu rio. \
n /foto: envia uma foto do
seu peixe.\n /
colocarcomida: alimenta
seu peixe.");
}

bot.getEvents().onCommand("help"
,[&bot](TgBot::Message::Ptr
message ) {
    bot.getApi().sendMessage(message
->chat->id, "Segue lista de
comandos;\n /info: mostra a
temperatura do aqu rio e o ph
. \n /foto: envia uma foto do
seu peixe.\n /comida: alimenta
seu peixe. \n /agua: troca a
agua do aqu rio.");
}

bot.getEvents().onCommand("foto"
,[&bot , &photoFilePath , &
photoMimeType](Message::Ptr
message){
    int fd1;
    if ((fd1 = open("/dev/video0" ,
O_RDONLY))<0)
    {
        bot.getApi().sendMessage(
            message->chat->id, "Camera
desconectada ou n o
compativel.");
        return 1;
    }
    system("sudo fswebcam -d /dev/
video0 640x480 --jpeg 85 -D 1
teste.jpg");
    bot.getApi().sendPhoto(message->
chat->id, InputFile::fromFile(
    "teste.jpg" , photoMimeType));
    system("sudo rm teste.jpg");
    bot.getApi().sendMessage(
        message->chat->id, "Foto
enviada!");
});
```

```

bot.getEvents().onCommand("comida",
    [&bot](TgBot::Message::Ptr message) {
        int N = 40;
        sqwv(SAIDA, -90,N);
        usleep(500000);
        sqwv(SAIDA, 90, N);
        bot.getApi().sendMessage(message
            ->chat->id, "Comida servida!")
            ;
    });

bot.getEvents().onCommand("agua",
    [&bot](TgBot::Message::Ptr message) {
        system("./moto");
        bot.getApi().sendMessage(
            message->chat->id, "Agua trocada!");
    });

bot.getEvents().onCommand("info",
    [&bot](TgBot::Message::Ptr message)
{
    system("sudo modprobe wl-gpio");
    system("sudo modprobe wl-therm");
    const char arquivo_temp[50] = "/sys/bus/wl/devices
/28-0207917754c6/wl_slave";
    float temperatura, ph, ph_dado;
    char ola[10], oi[10];
    char frase[20] = "Temperatura = ";
    ;
    char frase2[20] = "PH = ";
    char *temp;
    char buf[100];
    int fd;
    if((fd = open(arquivo_temp,
        O_RDONLY)) < 0)
    {
        printf("erro na abertura do
arquivo\n");
        return 1;
    }
    if(read(fd,buf,sizeof(buf)) < 0)
    {
        printf("erro na leitura do
arquivo !\n");
        return 1;
    }
    temp = strchr(buf,'t');
    sscanf(temp,"t=%s",temp);
    temperatura = atoi(temp)
});

```

```

    /1000;
    gcvt(temperatura,10,ola);
    strcat(frase,ola);
    strcat(frase, " C ");
    // final = strcat("Temperatura =
",temp_final);
    // message->text = temp_final;
    // read(i2c_fd, &ph_ret, 1);
    int l = 0;
    for (l = 0; l<9; l++)
    {
        ph_dado+=analogRead(A0)/10;
    }
    printf("%f",ph_dado);
    ph=(ph_dado*3.3/255)*(-5.7)
        +16.34;
    gcvt(ph,3,oi);
    strcat(frase2,oi);
    bot.getApi().sendMessage(message
        ->chat->id, frase);
    bot.getApi().sendMessage(message->
        chat->id, frase2);
    // close(i2c_fd);
}
try {
    printf("Nome do bot: %s\n", bot.
        getApi().getMe()->username.
        c_str());
    TgBot::TgLongPoll longPoll(bot);
    while (true) {
        printf("poll started\n");

        // message->text = temp_final;
        longPoll.start();
    }
} catch (TgBot::TgException& e) {
    printf("erro: %s\n", e.what());
}
return 0;
}

```

Algoritmo 2
CÓDIGO DAS BOMBAS

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
#include <dirent.h>
#include <string.h>
#include <wiringPi.h>
#define MOTO1 0
#define MOTO2 2

```

```

int main (int argc , const char **argv
)
{
wiringPiSetup();
pinMode(MOTO1, OUTPUT);
    pinMode(MOTO2, OUTPUT);
pid_t filhote;

filhote = fork();
if (filhote == 0)
{

digitalWrite(MOTO1, HIGH);
    usleep(8000000);
    digitalWrite(MOTO1, LOW);
usleep(8000000);
} else
{
digitalWrite(MOTO2, HIGH);
    usleep(8000000);
    digitalWrite(MOTO2, LOW);
{ return -1;
}

}
return 0;
}

```

VIII. REFERÊNCIAS

- [1] SMART Aquarium. , 2018. Disponível em: <[http://https://www.instructables.com/id/Smart_Aquarium/](https://www.instructables.com/id/Smart_Aquarium/)>. Acesso em: 30 aug. 2019.
- [2] DEMANDA por aquarismo cresce no brasil. , 2016. Disponível em: <[http://https://www.grupoaguasclaras.com.br/demanda-por-aquarismo-cresce-no-brasil](https://www.grupoaguasclaras.com.br/demanda-por-aquarismo-cresce-no-brasil)>. Acesso em: 30 aug. 2019.
- [3] 10 PROBLEMAS em aquários e suas soluções!. , 2018. Disponível em: <[http://https://www.estimacao.com.br/10-problemas-aquarios-suas-solucoes](https://www.estimacao.com.br/10-problemas-aquarios-suas-solucoes)>. Acesso em: 30 aug. 2019.
- [4] Introduction to Bluetooth Low Energy, Disponível em: <<https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>>. Acesso em: 29 set. 2019