Casa inteligente para cachorro

Bruno Alves Ferreira
Matrícula:15/0120117
Universidade de Brasília
bruno.ferreirasg@hotmail.com

Gabriela Cristina Cardoso
Matrícula: 15/0127065
Universidade de Brasília
gabccardoso@gmail.com

I. Resumo

O projeto consiste em uma casinha automatizada para cachorros que oferece ao dono a possibilidade de vigiar, alimentar e hidratar o seu animal de estimação, mesmo não estando em casa, de uma forma rápida e prática que é através de um bot de telegram. Além da comodidade para o dono, garante que o cachorro sempre tenha suas necessidades básicas atendidas.

II. Introdução

Devido a correria do dia a dia e tantas tecnologias disponíveis, as pessoas estão sempre tentanto delegar o máximo de tarefas diárias possiveis à máquinas, robôs etc. A praticidade, agilidade, organização, qualidade e segurança são fatores indispensáveis para gerarem resultados satisfatórios em vários setores na sociedade em que nos encontramos, e o conjunto desses elementos é chamado de automação.

Visto que cachorro não possuem a capacidade de colocar ração, água e controlar a temperatura do ambiente, e muitas vezes os donos são impossibilitados de estar em casa para cumprir essa tarefa e ficam preocupados com seus animais devido a isso, é proposto um projeto onde a casa do pet será automatizada para que o dono fique tranquilo caso não consiga cuidar do cachorro pessoalmente e também para trazer mais conforto e facilitar a vida do cachorro com a garantia de que não lhe faltará nada essencial à sua sobrevivência.

Para a realização do projeto será utilizada o Raspberry Pi 3. A placa é um computador barato, portátil e versátil, usado principalmente em projetos de programação, robótica e em iniciativas em geral com software e hardware livre. A mesma foi escolhida por ser capaz de realizar um bom processamento enquanto trabalha com dados de diversos sensores do projeto. Além disso, por se tratar de um projeto que será utilizado em casa, o consumo de energia gerado pela transmissão e recebimento de dados via wi-fi será suprido pela sua alimentação de 5v via USB.

III. Desenvolvimento

A. Materiais utilizados e custos

Custo do projeto	
COMPONENTES	CUSTO
Raspbeery Pi3	319,00
Válvula solenóide	33,75
Ponte H	19,10
WebCam	19,30
Vasílha de água	6,75
Vasilha de ração com estrutura do alimentador	54,75
Motor DC 12V	10,35
Sensor de temperatura DS18B20	16,90
Estrutura da casinha	70,00
Ventilador	14,99
Luz	5,50
4X Módulo relé	9,90
Preço total	604,08

B. Estrutura

A estrutura do projeto foi feita de madeira e para cachorros de pequeno porte, pois a casinha tem espaço suficiente para que a parte eletrônica fique de difícil acesso para o cachorro, evitando os riscos do contato com a mesma. .Os componentes foram colocados na parte de cima da casinha, pois o acesso à eles pelos projetistas seria de mais fácil acesso e ficou totalmente isolada do lugar que o cachorro fica.

O alimentador de ração foi feito com tubos de PVC, onde dentro do tudo horizontal terá outro tubo com espirais que está encaixado em um eixo feito com o CAP do tubo PVC. Quando acionado o comando de comida, o motor acoplabo ao tubo menor com espirais irá girar, fazendo com que os espirais empurrem a ração até chegar na vasilha do cachorro. O tempo que esse processo irá acontecer será determinado no código.

Para que o tubo tivesse mais força para girar o tubo com as espiras adicionado o peso da ração, foi colocada uma caixa de redução. como mostra na figura 4, assim o motor obteve mais força e foi possível girar o tubo sem grandes problemas.

C. Bot Telegram e Arquitetura da progamção em C

Com o objetivo de realizar a comunicação do dono do cachorro com a raspberry pi, um bot para telegram foi desenvolvido. A criação do Bot consistiu em utilizar o Bot

Figura 1. Estrutura da casinha.

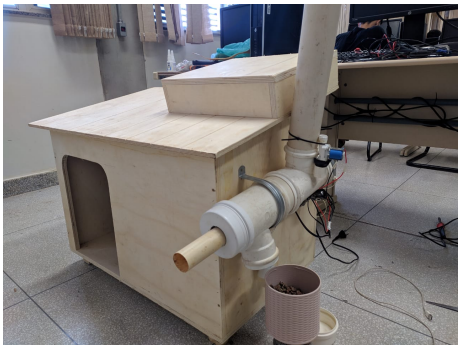

Figura 2. Estrutura da casinha com todos os sistemas acoplados.

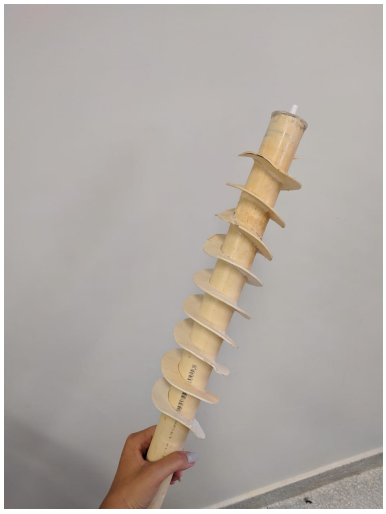

Figura 3. Espirais e tudo interno do alimentador.

Father[1], o qual possui comandos os quais geram um novo bot, e também permite fazer alterações no mesmo.

Com o bot gerado, teremos acesso ao seu token, o qual permite que a partir da API do telegram[2] possamos criar uma interface, utilizando a raspberry pi para enviar e receber dados. A interface com a API do telegram foi feita utilizando o método Getupdates, o qual consiste em utilizar uma página WEB, na qual será apresentada toda a informação enviada pelo bot.

O desenvolvimento do BOT com a raspberry pi consistiu em fazer o download da pagina contendo a informação do chat do bot e partir deste arquivo obter os comandos enviados pelo bot. O arquivo baixado consiste em um arquivo do tipo JSON, de forma que para obter acesso as variáveis deste arquivo uma biblioteca do JSON traduzida para C foi utilizada chamada jasmine. A partir do uso da biblioteca son-c/json.h, o arquivo foi decifrado, de forma a obter-se as informações úteis do bot.

Desta forma, com o JSON, obtemos acesso a todos os objetos da pagina, o que é importante para encontrar o update id, para desta forma sempre que uma nova menssagem chegar, a obtenção da pagina pelo método getUpdates utiliza-se de um offset, o qual consiste do ultimo update ID acrescentado de 1. Com isto, a pagina não ira acumular os comandos antigos, o que evita que a raspberry fique acumulando informação que ja foi utilizada.

Com estas informações, o código em c foi desenvolvido para raspberry pi, de forma a tratar essas informações e gerando os comandos do bot em si. Todo o código foi desenvolvido em linguagem C, e constistiu basicamente de 2 processos paralelos.

Um processo é responsável por obter os dados dos sensores, como o de temperatura. Este processo salva os valoes em uma variável global. Outro processo é responsável por identificar os comandos do bot do telegram, decodificalos, e por conseguinte aplicar as ações necessárias. Os 2 processos em paralelo utilizaram de 2 threads. A seguir temos o diagrama de blocos do sistema

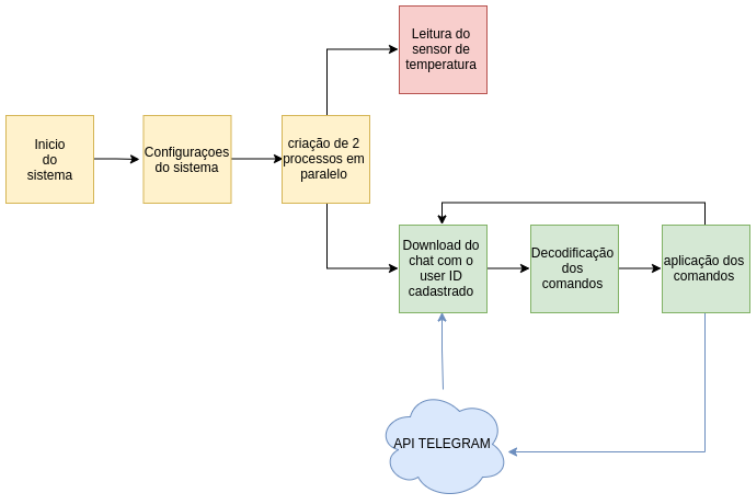

Figura 4. Diagrama de Blocos do sistema

Comandos do bot

- /comandos: Retorna todos os comandos ao usuário
- /temp: Retorna a temperatura da casinha ao usuário.
- /retrato: Fotográfa a agua e comida do cachorro, e liga uma luz no instante da fotografia.
- /agua : Coloca agua para o cachorro.
- /comida : Coloca a comida para o cachorro.
- /VentiladorON: Liga o ventilador do interior a casinha.

- /ventiladorOFF: Desliga o ventilador do interior a casinha

Figura 5. Comandos do Bot do telegram

D. Sensoriamento e atuação

O sensoriamento foi feito utilizando o próprio Raspberry Pi, pois só será usado um sensor, e como este é digital, só precisa conectar o mesmo no Raspberry e fazer a programação necessária. O único sensor que foi utilizado é o DS18B20 que servirá para monitorar a temperatura de modo a evitar que a casinha fique muito quente. o sensor DS18B20 possui sua própria inteligência. Ele é capaz de ler a temperatura, interpreta-la e enviar a informação do valor de temperatura em graus Celsius para o microcontrolador usando um barramento de apenar um fio (protocolo de comunicação One wire ou 1-wire). O sensor foi conectado no wPI-7 e o código feito apenas lê os valores da temperatura obtidos pelo sensor dentro de um arquivo.

O dono poderá acompanhar a temperatura da casinha através do comando ""no bot do telegram, e quando achar necessário, selecionará o comando para ligar o ventilador que estará no interior da casinnha, e assim o relè conectado no wPI-9 será acionado e o ventilador ficará ligado até que o dono desligue através do bot.

Figura 6. Sensor de Temperatura DS18B20

Figura 7. Controle da temperatura através do bot do telegram

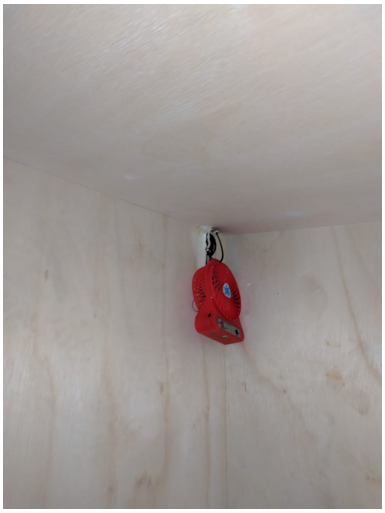

Figura 8. Ventilador e sensor de temperatura dentro da casinha.

E. Abastecimento de água e comida

O nível de água na vasilha do cachorro será monitorado pelo dono através da câmera que enviará fotos para o telegram, e quando necessária será aberta a válvula solenóide para o reabastecimento. A Válvula Solenoide para Água (1/2 x 1/2) possui internamente uma bobina em formato de cilíndrico. No momento que uma determinada corrente elétrica é conduzida pelos fios da bobina ela acaba gerando uma força no seu centro, a qual é responsável pelo acionamento do embolo que encontra-se na parte interna responsável pela abertura e fechamento do sistema.A válvula será ligada à um cano de

água, e quando o comando for selecionado, o relé conectado no Wpi8 é acionado e a válvula abrirá, enchendo a vasilha d'água por um tempo determinado no código de 5 segundos. O abastecimento de comida, como foi citado mais acima será feito por sistema constituido por canos de pvc e um motor DC. O monitoramento e acionamento será feito pelo próprio dono através do bot do telegram, que poderá acompanhar pela câmera o nível de ração da vasilha, e quando achar necessário o reabastecimento selecionará o comando, que irá acionar o relé conectado no wPI6, ligando o motor que fará os espirais girarem e empurrarem a ração até a vasilha.

Figura 9. Válvula solenóide

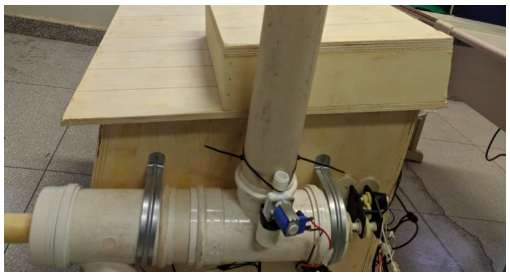

Figura 10. Sistema de alimentação e água para o cachorro.

O abastecimento de comida poderia ser desligado pelo próprio dono, mas tendo em vista que pode ocorrer algum imprevisto com o celular e o dono por algum motivo não conseguir desligar a válvula ou o motor, foi decidido que seria melhor ter um tempo limitado que os relés ficam ligados.

IV. Resultados

Os comandos funcionaram perfeitamente e todo o acionamento foi feito de forma correta. O dimensionamento para os fios foi o suficiente para o projeto ser seguro e eficaz, e a alimentação das fontes e da raspberry, assim como a manipulção das portas da mesma foram eficientes.

A estrutura da casinha comportou tudo que precisava, e o espaço foi perfeito para comportar os componentes evitando o acesso do cachorro à eles, além de proporcionar um ambiente confortável e espaçoso, caso o cachorro seja de pequeno porte, pois a casinha foi projetada focando nesse tipo de raça.

Houve um mau dimensionamento de motor, pois o que foi usado parecia suficiente para girar o sistema de alimentação, e com pouca ração ele é realmente muito eficaz, mas quando enche muito o recipiente o motor não tem torque suficiente para girar o espiral. Por isso, afim de melhoria de projeto, é recomendado um motor mais potente com uma caixa de redução já interna, pois a redução da marcha e aumento da força com certeza serão suficientes para que o alimentador funcione perfeitamente.

V. Conclusão

O projeto atende as expectativas de pessoas que não podem estar sempre presentes para cuidar de seus animais, e traz a segurança ao cliente de que ele sempre saberá se seu cachorro está bem alimentado e hidratado. Além disso, dá mais garantia ao pet do atendimento às suas necessidades para sobrevivência.

Para melhoria de projeto seria importante a troca do motor por um mais pontente e com redução maior, e seria interessante adicionar uma câmera no interior da casinha para que o dono pudesse ver seu cachorro quando quisesse, se assim ele desejasse.

O conteúdo apresentado em sala foi essencial para que o projeto tivesse sucesso, além de permitir um conhecimento mais profundo em programação e em RaspberryPi 3, e contribuiu para um melhor manuseamento da mesma.

Referências

[1] Consumo de energia do Raspberry Pi. [S.l.], 2017. Dispon[ivel em: http://blog.everpi.net/2017/03/raspberry-pi-3-consumo-de-energia.html. Acesso em 30 Ago. 2019.

[2] Truques para manter cães aquecidos. [S.l.]. Disponível em: http://www.biovet.com.br/imprensa/frio-para-cachorro-8-truques-para-manter-caes-aquecidos-no-inverno/20180726-141710-u579. Acesso em 30 Ago. 2019.

Raspberry Pi – Servidor Web com HTTPS e Acesso pela Internet. [S.l.]. Disponível em: https://daniel.scota.com.br/?p=828. Acesso em 30 Ago. 2019.

JSON PARSE – Biblioteca JSON traduzida para C. [S.l.]. Disponível em: https://json-c.github.io/json-c/json-c-0.10/doc/html/json

API TELEGRAM – Telegram APIs. [S.l.]. Disponível em: https://core.telegram.org/. Acesso em 1 Out. 2019.

Anexo 1 - Código do projeto

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <semaphore.h>
#include<json-c/json.h>
#include <string.h>
#include <stdint.h>
#include <wiringPi.h>
#include <wiringSerial.h>
#include <errno.h>
#include <signal.h>
#include <dirent.h>
#include <fcntl.h>


volatile int varCompartilhada=0;
volatile int temp_1,umid,ldr;

static pthread_mutex_t mutexLock;




struct json_struct
{
  int updat_id;
  int valid;
  char msg[50];        // Declaring an
     array will allocate the specified

};


struct json_struct dados_json()
{
    struct json_struct js;
    FILE *fp;
    char buffer[2048];
    struct json_object *parsed_json;
    struct json_object *result;
    struct json_object *result_n;
    struct json_object *update_id;
    struct json_object *text;
    struct json_object *message;
    size_t i;

    fp = fopen("/home/pi/
       Embarcados_codigos/api.telegram.
       org/bot961579878:
       AAG4cVhRDowO13ZJ1vRm-3
       DHSiukfzQN2U8/getUpdates","r");
    fread(buffer, 2048, 1, fp);
    fclose(fp);

    parsed_json = json_tokener_parse(
       buffer);
    json_object_object_get_ex(parsed_json
       , "result", &result);
    i=json_object_array_length(result);
                //Num de indices do
       array result



    if(i>0)
    {
        result_n=
           json_object_array_get_idx(
           result,i-1); //pegamos o
           ultimo array

        json_object_object_get_ex(
           result_n, "update_id", &
           update_id);
        json_object_object_get_ex(
           result_n, "message", &message)
           ;
        json_object_object_get_ex(message
           , "text", &text);
        strcpy(js.msg,
           json_object_get_string(text));
        js.updat_id=json_object_get_int(
           update_id);
    }
    js.valid=i;
return js;
}
// Fun  o que incrementa o Contador
void* bot_data_RT (void *arg)
{



    system("wget -r -P /home/pi/
       Embarcados_codigos https://api.
       telegram.org/bot961579878:
       AAG4cVhRDowO13ZJ1vRm-3
       DHSiukfzQN2U8/getUpdates > log.txt
        ");

    struct json_struct json_dados;
    char linha2[200],linha3[200],off_set
       [200],off_set_rm[200],temp[]="/
       temp",starts[]="/start",foto[]="/
       retrato",bomba[]="/agua",comandos
```

```
        []="/comandos",vent_on[]="/ventON"
        ,vent_off[]="/ventOFF",comidas[]="
        /comida";
    int i=8000,updat_id1=0,updat_id2=1;

    json_dados=dados_json();
    updat_id2=json_dados.updat_id;
while(1)
    {
    printf("Baixando Site\n");
    system("wget -r -P /home/pi/
        Embarcados_codigos https://api.
        telegram.org/bot961579878:
        AAG4cVhRDowO13ZJ1vRm-3
        DHSiukfzQN2U8/getUpdates > log2.
        txt");

    json_dados=dados_json();
    updat_id1=json_dados.updat_id;
    if((updat_id1!=updat_id2) && (
        json_dados.valid>0))
    {
            printf("Comando recebido do
                bot\n");
            if (strcmp(temp,json_dados.
                msg)==0)
            {

                printf("Mande a
                    temperatura\n");
                sprintf (linha2,"curl -s
                    -X POST https://api.
                    telegram.org/
                    bot961579878:
                    AAG4cVhRDowO13ZJ1vRm-3
                    DHSiukfzQN2U8/
                    sendMessage -d chat_id
                    =627732924 -d text='
                    Temperatura: %d C '",
                     temp_1);
                system(linha2);
            }

            if (strcmp(comidas,json_dados
                .msg)==0)
            {

                sprintf (linha3,"curl -s
                    -X POST https://api.
                    telegram.org/
                    bot961579878:
                    AAG4cVhRDowO13ZJ1vRm-3
                    DHSiukfzQN2U8/
                    sendMessage -d chat_id
                    =627732924 -d text='
                    Colocando Comida'");
                system(linha3);
                digitalWrite(6, HIGH);
                usleep(5000000);

                digitalWrite(6,LOW);
}


            if(strcmp(bomba,json_dados.
                msg)==0)
            {
                digitalWrite (8, HIGH) ;
                usleep (3000000) ;
                    digitalWrite (8, LOW
                        ) ;

                    system("curl -s -X
                        POST https://api.
                        telegram.org/
                        bot961579878:
                        AAG4cVhRDowO13ZJ1vRm
                        -3DHSiukfzQN2U8/
                        sendMessage -d
                        chat_id=627732924
                        -d text='
                        Recipiente Cheio'"
                        );

            }

            if(strcmp(vent_on,json_dados.
                msg)==0)
            {
                digitalWrite (9, HIGH) ;
                system("curl -s -X POST
                    https://api.telegram.
                    org/bot961579878:
                    AAG4cVhRDowO13ZJ1vRm-3
                    DHSiukfzQN2U8/
                    sendMessage -d chat_id
                    =627732924 -d text='
                    Ventilador Ligado'");


            }
            if(strcmp(vent_off,json_dados
                .msg)==0)
            {
                digitalWrite (9, LOW) ;
                system("curl -s -X POST
                    https://api.telegram.
                    org/bot961579878:
                    AAG4cVhRDowO13ZJ1vRm-3
                    DHSiukfzQN2U8/
                    sendMessage -d chat_id
```

```
            =627732924 -d text='
            Ventilador Desligado'"
            );


    }

    if((strcmp(comandos,
        json_dados.msg)==0) || (
        strcmp(starts,json_dados.
        msg)==0) )
    {

         system("curl -s -X POST
             https://api.telegram.
             org/bot961579878:
             AAG4cVhRDowO13ZJ1vRm-3
             DHSiukfzQN2U8/
             sendMessage -d chat_id
             =627732924 -d text='
             Ola :) Os comandos da
             sua DOGhouse s o \n /
             temp: Informa a
             temperatura da casinha
              \n /comida: Informa a
              Umidade relativa \n /
             retrato: Tira uma foto
              do interior da
             casinha \n /agua:
             Enche o recipiente de
             agua do cachorro \n /
             ventON: Liga o
             ventilador \n /ventOFF
             : Desliga o ventilador
             \n Em caso de d vidas
             , mande uma mensagem
             para (61) 98145-4643 '
             ");
    }
    if(strcmp(foto,json_dados.msg
        )==0)
    {

         digitalWrite (0, HIGH) ;
         usleep(2000000);
         system("fswebcam -d /dev/
             video0 foto.jpg");
          digitalWrite (0, LOW) ;
         system("curl -X POST
             https://api.telegram.
             org/bot961579878:
             AAG4cVhRDowO13ZJ1vRm-3
             DHSiukfzQN2U8/
             sendPhoto? -F chat_id
             =627732924 -F photo=@/
             home/pi/
             Embarcados_codigos/
             foto.jpg");
                                }

            if(json_dados.valid>4)
            {
                 sprintf(off_set,"wget -r
                     -P /home/pi/
                     Embarcados_codigos/
                     https://api.telegram.
                     org/bot961579878:
                     AAG4cVhRDowO13ZJ1vRm-3
                     DHSiukfzQN2U8/
                     getUpdates?offset=%d",
                     updat_id1+1);
                 system(off_set);
                 sprintf(off_set_rm,"rm /
                     home/pi/
                     Embarcados_codigos/api
                     .telegram.org/
                     bot961579878:
                     AAG4cVhRDowO13ZJ1vRm-3
                     DHSiukfzQN2U8/'
                     getUpdates?offset=%d'"
                     ,updat_id1+1);
                 system(off_set_rm);

            }

     }

     updat_id2=updat_id1;
     usleep(100000);
    }
return NULL;
}

// Fun   o que decrementa o Contador
void* DATA_GET (void *arg)
{

 DIR *dir;
 struct dirent *dirent;
 char dev[16];       // Dev ID
 char devPath[128]; // Path to device
 char buf[256];      // Data from device
 char tmpData[6];    // Temp C * 1000
     reported by device
 char path[] = "/sys/bus/w1/devices";
 ssize_t numRead;

 dir = opendir (path);
 if (dir != NULL)
 {
  while ((dirent = readdir (dir)))
   // 1-wire devices are links beginning
```

```
        with 28-
    if (dirent->d_type == DT_LNK &&
        strstr(dirent->d_name, "28-") !=
            NULL) {
     strcpy(dev, dirent->d_name);
     printf("\nDevice: %s\n", dev);
    }
          (void) closedir (dir);
          }
 else
 {
  perror ("Couldn't open the w1 devices
      directory");
 }

          // Assemble path to OneWire
             device
 sprintf(devPath, "%s/%s/w1_slave", path,
      dev);
 // Read temp continuously
 // Opening the device's file triggers
     new reading
 while(1) {
  int fd = open(devPath, O_RDONLY);
  if(fd == -1)
  {
   perror ("Couldn't open the w1 device."
       );

  }
  while((numRead = read(fd, buf, 256)) >
      0)
  {
   strncpy(tmpData, strstr(buf, "t=") +
       2, 5);
   temp_1 =( strtof(tmpData, NULL))/1000;


 }
  close(fd);
 }
          /* return 1; --never called due
             to loop */

    return NULL;
}

int main (int argc, char** argv)
{



          wiringPiSetup () ;
          pinMode (8, OUTPUT) ;
          pinMode (0, OUTPUT) ;
          pinMode (9, OUTPUT) ;
     pinMode(6, OUTPUT);

     pthread_t t0;
     pthread_t t1;
     pthread_mutex_init(&mutexLock, NULL);
     pthread_create(&t0, NULL, bot_data_RT
         , NULL);
     pthread_create(&t1, NULL, DATA_GET,
        NULL);
     pthread_join(t0, NULL);
     pthread_join(t1, NULL);

     pthread_mutex_destroy(&mutexLock);

     printf("Valor final: %d\n",
        varCompartilhada);
     return 0;
}
```