

DRUG CARE

Controle de medicamentos

Paulo Henrique Moreira de de Carvalho Faria
Universidade de Brasília - FGA
oluap.ph@gmail.com

Rodrigo Bonfácio de Medeiros
Universidade de Brasília - FGA
rodrigo_medeiros92@hotmail.com

I. RESUMO

O presente ponto de controle descreve os avanços e soluções implementadas no projeto Drugcare, que consiste no gerenciamento inteligente de medicamentos, utilizando Raspberry Pi. Está segunda etapa de desenvolvimento visou dar os primeiros passos no direcionamento do software de controle e confecção do mecanismo de extração.

II. INTRODUÇÃO

Se tratando de um sistema que torna dinâmico, a ingestão e armazenamento de medicamentos, o sistema Drugcare demandou a criação de um sistema que automatizasse o modo de aquisição e estoque de medicamentos, bem como uma diretriz inicial para o protocolo de catálogo, controle de fluxo e horários de ingestão.

A automação para estoque de determinado artigo de utilidades é um processo que se torna vital na dinamização de sistemas de distribuição. Assim, todo o maquinário empregado para esse processo deve ser projetado de forma sincronizada com o sistema de controle, seguindo um fluxograma de funcionamento, com o intuito de evitar atrasos de fluxo e falhas repentinas. [1]

No sistema Drugcare, o medicamento vai está sujeito a três processos: catálogo e rotinas, estoque e extração. O processo de catálogo e rotinas recebe dados do usuário e armazena em arquivos do sistema para controle dos medicamentos em estoque e os coloca à disposição para criação de rotinas de ingestão. Adjacente a isso, o processo de extração utiliza um mecanismo controlado para disponibilizar os medicamentos em estoque de acordo com as rotinas em vigor.

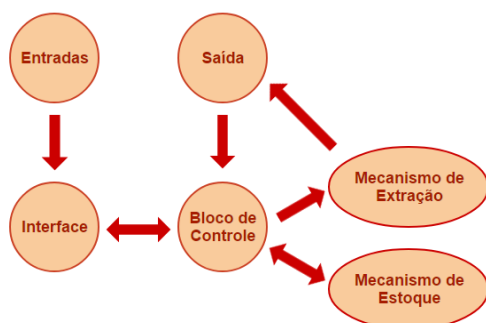


Figura 1 – Fluxograma Drugcare

III. DESENVOLVIMENTO

Para o desenvolvimento inicial do bloco de controle, que a princípio simula a interface de catálogo e rotina, foi necessária a utilização de ideias de escrita em arquivo e conexão GPIO para formar o protocolo de acesso do usuário a máquina. Quanto ao mecanismo de extração, foi necessária a idealização de um sistema de alçapões, uma para cada um dos quatro compartimentos de estoque do protótipo. Cada alçapão é independente e controlado por um servo-motor que é acionado quando solicitado um medicamento.

1. Descrição do hardware e estrutura

1.1 Bloco de interface

O bloco de interface é composto por um aparelho *Smartphone* com um aplicativo dedicado que se comunica em tempo real com a máquina- sendo a máquina um servidor e o usuário cliente - enviando e recebendo parâmetros. Este bloco ainda está em desenvolvimento.



Figura 2 – Smartphone.

1.2 Bloco de controle

O bloco de controle é composto por uma Raspberry Pi 3 B+, optoacoplador e um drive de potência para os servo-motores. A *Raspberry* é responsável por todo o controle e comunicação da máquina, ao passo que o drive de potência e o optoacoplador são componentes para manutenção e segurança do sistema como um todo.



Figura 3 – Raspberry Pi 3 B+.

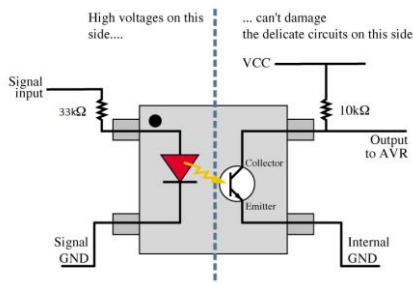


Figura 4 – Sistema optoacoplador

1.3 Bloco de extração

O bloco de extração é composto por um mecanismo de alavancas espelhadas que realizam o mesmo movimento quando solicitadas por apenas um ponto de tensão. Cada compartimento de estoque possui um alçapão de porta dupla que é aberta pelo sistema de alavancas e cada par é acionado por um servo-motor. Todos os alçapões compartilham uma mesma rampa de saída que possui acesso manual para o usuário. O bloco é todo em madeira, exceto os motores.

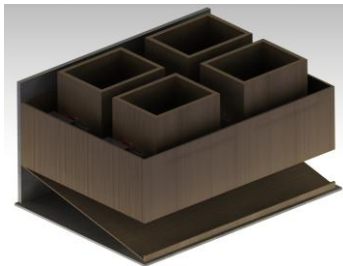


Figura 5 – Bloco de extração completo.

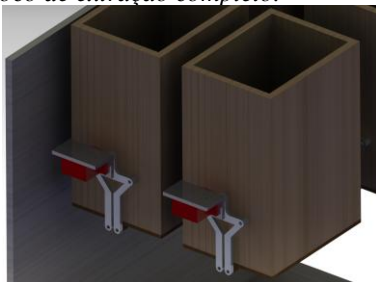


Figura 6 – Sistema de alavancas e alçapão (vista lateral).

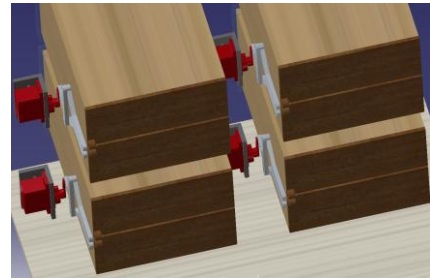


Figura 7 – Sistema de alavancas e alçapão (vista inferior).

1.4 Bloco de estoque

O bloco de estoque é composto por quatro compartimentos que tem dimensões que varrem todos os tamanhos de embalagens. Cada compartimento possui um par de LEDs infravermelho, receptor e emissor que estabelecem um feixe entre eles. Toda vez que o feixe é interrompido o sistema confere que o medicamento foi adicionado ou devolvido. Além disso cada compartimento também contém LEDs RGB indicativos, que dependendo da cor orientam o usuário: amarelo, para aviso de devolução e inserção; verde, para indicar o compartimento que está escalado na rotina no alarme de ingestão; vermelho, indica o compartimento que já foi ingerido. Este bloco ainda está em construção.

2. Descrição do Software

Uma versão inicial de teste (v 1.0) - com a simulação de apenas um motor - do software foi escrito em linguagem C para Linux e compilado com o auxílio da ferramenta de desenvolvimento GCC. A figura 8 mostra o fluxograma do programa. O Bloco “início” indica o início do processo, após isso é requerida as entradas do usuário (Nome do remédio, horário e o recipiente a ser guardado). Os losangos representam uma condição, se a entrada for inválida, como recipiente ou horário não existente, ele retorna a pedir a entrada ao usuário até a entrada ser válida.

Após os dados serem guardados, o código verifica se o horário indicado pelo usuário é igual ao tempo real do computador, ou seja, se é a hora do paciente tomar o remédio, caso a condição seja falsa, o programa verifica novamente até a condição ser verdadeira. Quando a condição for verdadeira é ativo um nível lógico alto que aciona o motor do recipiente para liberar o remédio, é impresso na tela qual remédio a ser tomado e um relatório é criado com dos dados do processo (nome do remédio e horário). Uma réplica do código se encontra no apêndice com a especificação das funções e bibliotecas usadas.

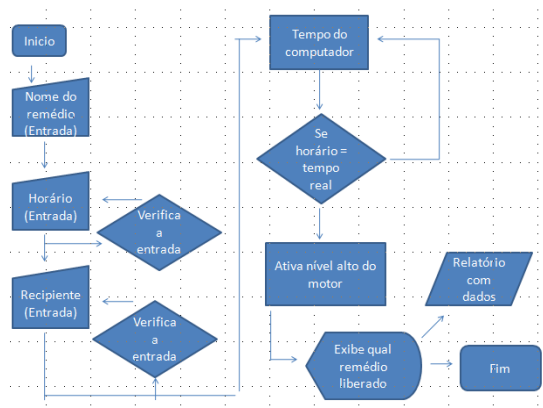


Figura 8 – Fluxograma VI.0

IV. RESULTADOS

Para a validação do projeto proposto, foi montado um circuito usando um led para simular a saída de nível alto do motor, o software de teste funcionou perfeitamente quando as condições eram satisfeitas, ou seja, quando a hora de tomar remédio era igual ao tempo real do computador o led acendia por um período de tempo e depois apagava mostrando o compartimento sendo aberto, após isso o sistema criava o relatório para posterior análise, assim a validação inicial foi satisfatória. Porém há muitas variáveis a serem introduzidas, como a adição dos quatro motores do recipiente, com seus circuitos de proteção e driver de potência, a interface entre o usuário e a máquina via celular e a otimização do código para essas novas variáveis, portanto a validação completa do sistema ainda está em desenvolvimento.

VI. APÊNDICE

```

#include<stdio.h>           //Biblioteca padrão de entrada /saída
#include<time.h>            //Biblioteca para os uso do tempo do
                           sistema
#include<stdlib.h>          //Biblioteca de propósito geral
#include<wiringPi.h>        //Biblioteca para os pinos GPIO
#include<string.h>          // Biblioteca para manipulação de strings

int main(void)
{
    struct tm *local; // Struct para determinar a hora real do
                      computador

    int hora;         // Variável para a hora real do computador
    int minutos;      // Variável para o minuto real do computador
    int h_des=0;       // Variável para a hora desejada de tomar o
                      remédio
    int m_des=0;       // Variável para o minuto desejado de tomar
                      remédio
    char nome[30];     // Variável para o nome do remédio
    int rec;           // Variável para dado recipiente

    FILE *fp;          // Abertura do arquivo de relatório
    fp=fopen("Relatorio.txt","w"); //
  
```

V. CONCLUSÃO

No presente ponto de controle alguns blocos propostos na introdução do problema estão bem direcionados para serem solucionados, como o mecanismo de extração, de estoque e o bloco de controle, salvo algumas modificações já discutidas anteriormente. A descrição do hardware e estrutura estão completa, com exceção da interface do usuário com a máquina que ainda está sendo desenvolvido. A descrição e validação do software, que foi apenas um teste, ainda precisa ser otimizada e completada de acordo com as novas variáveis que ainda serão introduzidas. Em suma, o ponto de controle foi satisfatório, pois apresentou bastantes soluções e clareou os problemas que devem ser resolvidos.

REFERÊNCIAS

- [1] HARADA, J.B; FERNANDA. SCHOR; PAULO. O Problema da autoadministração de medicamentosor idosos com baixa visão e cegueira sob a ótica do design centrado no humanoJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

```

wiringPiSetup () ; // Inicialização da biblioteca de GPIO

pinMode(0 ,OUTPUT); // Definindo o Pino 18 como saída

time_t t; //
t=time(NULL); // Definindo o tempo real do computador
local=localtime(&t); //

digitalWrite (0, LOW) ; // Desligado a saída, caso esteja ativa

printf("Digite o nome do remedio: "); // Entrada do nome do
remédio
scanf("%s",nome); //

// Entrada da hora de tomar o remédio com a condição de hora
// inválida
do
{
printf("Digite a hora desejada de tomar o remédio (00-23):");
scanf("%d",&h_des);
if (h_des > 23 || h_des< 0)
printf("*Hora Inválida*\n\n");
} while (h_des > 23 || h_des< 0);

printf("*Hora inserida*\n\n");

// Entrada do minuto de tomar o remédio com a condição de minuto
// inválida
do
{
printf("Digite o minuto desejado(1-59):");
scanf("%d",&m_des);
if (m_des<1 || m_des>59)
printf("*Minuto invalido*\n\n");
}while (m_des<1 || m_des>59);

printf("*Minuto inserido*\n\n");

// Entrada do recipiente do remédio com a condição de recipiente
// inválido
do
{
printf("Digite o recipiente na qual o remedio foi inserido (1-4):");
scanf("%d",&rec);
if (rec<1 || rec>4)
printf("*Recipiente invalido*\n\n");
}while (rec<1 || rec>4);

printf("*Recipiente inserido*\n\n");

hora=local->tm_hour; // Determinando a hora real
minutos=local->tm_min; // Determinado o minuto real

```

```
// Verifica se a hora e minuto desejado são iguais a hora real
if(hora == h_des && minutos == m_des)
{
    digitalWrite (0, HIGH) ; delay (3000) ; // ativa nível alto
                                     // por 3000ms
    digitalWrite (0, LOW) ; delay (500) ; //Ativa nível baixo
    printf("'%s Liberado*\n\n",nome); // Saída indicando o
                                     //remédio liberado
    fprintf(fp,"Remedio tomado: %s\n",nome); // Escrita no
                                     //arquivo
    fprintf(fp,"Horario tomado: %d:%d\n",h_des,m_des);
}

fclose(fp); // Fecha o arquivo

return 0;
}
```