

Despertador inteligente

Sistema de alarme que gera disposição ao usuário

Vitor Jacinto Sulzbach
Faculdade do Gama - FGA
Universidade de Brasília - UnB
Gama, DF
vjsulzbach@gmail.com

I. JUSTIFICATIVA

O capitalismo moderno traz consigo uma necessidade crescente de máxima eficiência em qualquer processo realizado. Empresas de todos os portes estão buscando profissionais de alto rendimento. Está pressão social amplifica problemas graves como stress crônico, depressão e até suicídio, além de problemas menores que afetam diretamente a qualidade de vida da população como a falta de tempo para a família ou hobbies e sedentarismo.

Em meio a tantos contras é difícil não ver esta cruzada moderna como vilã, porém ela se faz necessária para o progresso que sempre guiou a sociedade a tempos melhores. Nos resta então aprender a lidar com o stress da vida do novo milênio e descobrir maneiras de restaurar o equilíbrio do cotidiano com saúde e qualidade de vida.

Uma via saudável para alcançar esta eficiência é ter um bom ciclo diário de sono e acordar disposto para encarar os desafios diários, ter níveis controlados de cortisol, o hormônio do stress que nos ajuda a acordar e iniciar o dia, e melatonina, considerado o hormônio do sono que tem sua produção associada a menor incidência luminosa e diz ao nosso corpo a hora de começar a parte noturna do ciclo circadiano. Desta forma é possível aproveitar melhor a vida, com mais qualidade e saúde, menos esforço e sem deixar de lado o trabalho ou estudo. Tendo em vista as pessoas que tentam manter este equilíbrio nasce a proposta deste projeto.

II. OBJETIVOS

Arquitetar e construir um sistema despertador otimizado para um acordar suave, porém efetivo, trazendo disposição ao usuário. O sistema também deve trazer dicas e funcionalidades adicionais para manter o ciclo diário de cortisol/melatonina estável.

III. PROJETOS

Com o objetivo de realizar o despertador que minimiza o cansaço do usuário e o deixa disposto, segundo a pesquisa realizada, surge três ideias principais listadas abaixo.

1. Despertar suave

Para muitos é um esforço muito grande levantar ao acordar, aperta-se o botão soneca do despertador seguidamente e quando consegue sair da cama não há disposição alguma. Ao passo que quando se acorda ao natural, como quando se acorda com a luz do dia atravessando a janela e o barulho que advém da mesma, não há resistência do corpo a iniciar o dia. A grande diferença entre estes dois casos é a suavidade em se acordar, em um caso o usuário acorda com um barulho alto que o tira muitas vezes de um sono REM, o sono mais pesado, essencial para a manutenção do corpo, enquanto o outro caso leva a um despertar suave, que dá tempo ao organismo para passar por estágios menores do sono e recuperar o vigor, o usuário acorda com o sentimento de que o tempo de dormir acabou e é hora de levantar. Espera-se então incluir no sistema uma forma de replicar a luz com espectro parecido com o solar, e replicar sons do cotidiano como o som de um trabalho pesado, de motores ou algum outro som da preferência do usuário, optando por sons com graves e fortes. Todos estes fatores devem surgir lentamente, levando em conta que alguns usuários irão acordar antes que outros, cada um terá de regular o horário de início do toque e o horário limite onde o volume fica tão alto quanto o de um despertador comum.

2. Levantar forçado

É sempre um incomodo sair da cama ao acordar, porém se manter em pé após levantar não é difícil, o ato de se levantar age como uma barreira de potencial que precisa ser quebrada para que o cotidiano se inicie. Com o objetivo de quebrar esta barreira o usuário deverá se levantar e se dirigir ao switch que desativa o sistema de alarme, como o despertar será lento, como proposto anteriormente, o usuário já estará bem desperto e de pé.

3. Sistema de assistência

Após acordar é recomendado ler algumas notícias, tomar uma xícara de café e exercitar um pouco a mente, portanto o sistema de assistência deve sugerir tais atividades para o usuário. Um adicional que é cogitado para o sistema é a

automatização de uma cafeteira para preparar o café junto com o desativamento do alarme, além da troca do switch para cessar o despertador por um minijogo, rápido, mas que exige um pouco de raciocínio, como algumas somas e multiplicações. Estes adicionais exigem um gasto adicional considerável que pode não se sustentar ao longo do desenvolvimento.

IV. REQUISITOS

Para atingir o objetivo o projeto deve seguir as etapas descritas abaixo com seus respectivos riscos.

1. Planejamento

Deve-se planejar a partir da revisão bibliográfica todas as funcionalidades do sistema a ser montado, levando em consideração artigos e publicações científicas para justificar tais funcionalidade, mostrando seus mecanismos de atuação e benefícios. A seguir é montado de maneira técnica o esquemático do hardware a ser utilizado com base em pesquisa menos formal em busca de data-sheets de componentes que possam realizar a atuação desejada, sempre buscando o melhor custo benefício. Nesta etapa existe o risco de descobrir problemas graves nos métodos de atuação e mudanças abruptas de hardware o que pode elevar os custos à um patamar insustentável, resultando na troca de projeto.

2. Execução

Com fim do planejamento se faz o pedido dos componentes enquanto é iniciado o desenvolvimento do software. O sistema irá ser montado utilizando uma linguagem de programação mais simples, python, simplesmente para teste de conceito, demonstrando o funcionamento do sistema como um todo. Então módulo a módulo a programação deve ser refeita em C para otimização de projeto. Nesta etapa existe novamente de encontrar problemas não identificados na etapa anterior que acarretariam na elevação do preço e possível troca de projeto, também há o risco de a programação dos módulos em C ser complicada demais para ser realizada dentro do tempo estipulado, resultando em redução do projeto ou aumento de preço devido a troca de componentes.

V. BENEFÍCIOS

O usuário acordara com mais disposição. Muitas pessoas dormem menos tempo que o recomendado(8hrs), se forem acordadas durante o sono REM podem acabar tendo dores de cabeça, enjoo, entre outros problemas. O sistema proposto desperta lentamente, dando tempo para o organismo transitar entre os estados do sono, o que minimiza estes efeitos.

VI. PONTO DE CONTROLE 2

O ponto de controle 2 foi dedicado a projetar de maneira técnica o protótipo do projeto, provando que cada bloco concebido no ponto de controle 1 poderia se encaixar e gerar o

produto final. Foi realizado então diretamente em C dentro da raspberry com auxílio da IDE Geany a programação do software embarcado que controla a parte de alarme.

VII. CIRCUITO

O circuito a ser utilizado se divide em 3 partes, um realiza a iluminação gradual do ambiente, a outra é a circuitaria para desligar o alarme a última é a própria raspberry que junta os blocos com o software.

A. Iluminação gradual

A iluminação gradual pode ser realizada através de um relé que recebe na entrada uma GPIO da raspberry que pode emitir pulsos de largura específica controlando a luminosidade da lâmpada.

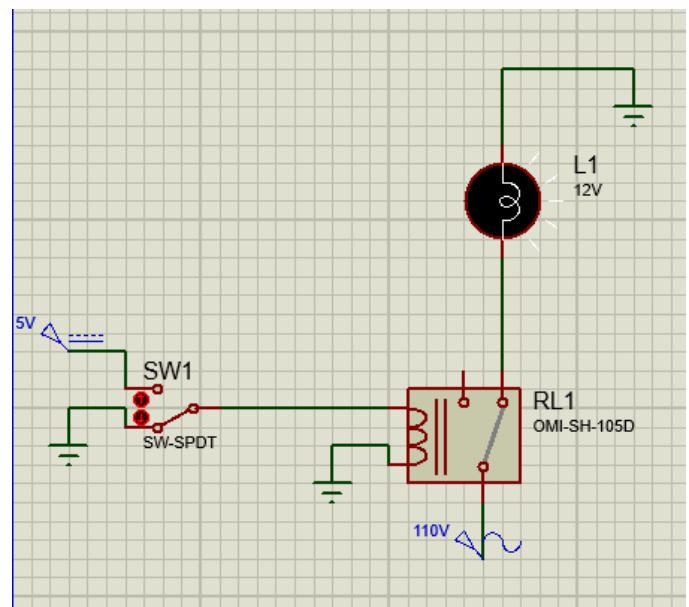


Figura 1- Relé

Utiliza-se um circuito com um transistor NPN darlington (TIP 131) com o intuito de controlar 5V de tensão na entrada do relé também fornecendo a corrente que se necessita (56 mA).

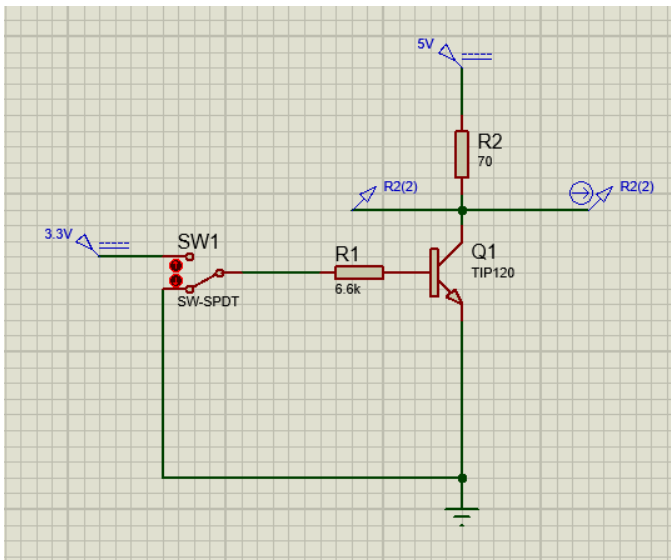


Figura 2 - Darlington

B. Desligar alarme

O alarme é desligado por um jogo simples, composto por 5 LEDs. Gera-se um ciclo onde uma LED é acesa de cada vez, o usuário precisa pressionar o botão quando a LED do meio estiver acesa 3 vezes seguidas, então o alarme para e se inicia um vídeo pré-determinado para auxílio de produtividade pela manhã.

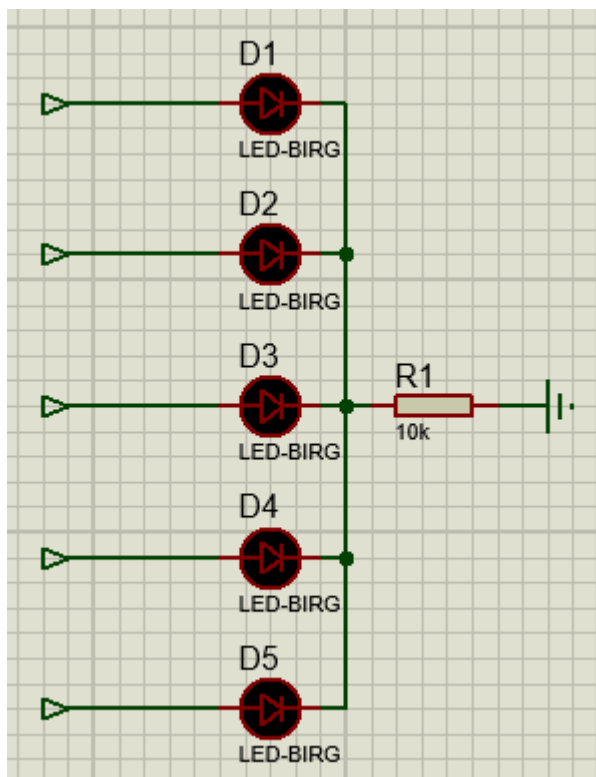


Figura 3 -LEDs

C. Controle do despertador

O despertador será controlado via teclado e LCD, seu funcionamento não depende de circuito exterior além do seu próprio módulo.

VIII. SOFTWARE

O software construído é composto por dois arquivos, um que executa o toque de alarme e recebe um sinal para parar, e outro que controla quando este toque deve ocorrer.

A. Execução do Toque.

Como mostra a figura abaixo o main é apenas a especificação de um sinal e o toque infinito de uma música.

```
int main(int argc, char *argv[]){
    signal(SIGUSR1, tratamento_SIGUSR1);
    while(1)
        system("aplay -q Music.wav");
    return 0;
}
```

A parte de tratamento do sinal recebido é onde o alarme para, como a música toca através do comando system(aplay) é necessário obter o valor do pid do processo aplay através do comando pidof, então este valor é gravado em um arquivo, que é lido para então parar a música, depois basta finalizar a execução com um exit(1).

```
void tratamento_SIGUSR1() {
    char *b;
    char dt[40] = "(kill -9 ";
    char f[] = ") &>/dev/null";
    system("rm -f a.txt");
    system("pidof aplay | tee a.txt");
    b = le_arq_texto("a.txt");
    strcat(dt,b);
    strcat(dt,f);
    system(dt);
    exit(1);
}
```

B. Controle

O controle é composto por várias funções, a main é responsável por determinar a hora atual através da biblioteca time, através de alocação dinâmica uma variável tipo int é preparada para passar por uma função que pergunta ao usuário a hora de despertar e armazena nesta variável, para então entrar em um loop de onde só sai quando a hora certa chega, onde se pode sair do programa ou programar um novo alarme.

```

void loop(){
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    printf("Agora: %d-%d-%d %d:%d:%d\n", tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday, tm.tm_hour, tm.tm_min, tm.tm_sec);
    int* data;
    data = malloc(5);
    set_data(data);
    printf("Despertar em : %i/%i %i:%i\n", data[1], data[0], data[2], data[3]);
    int i = 1;
    while(1) {
        t = time(NULL);
        tm = *localtime(&t);
        if(((i==1)&&(tm.tm_mon+1)==data[0])&&(tm.tm_mday==data[1])&&(tm.tm_min==data[2])&&(tm.tm_hour==data[3])) {
            tocar_despertador();
            i=0;
            sleep(1);
            printf("digite s para ajustar um novo alarme ou n para sair: ");
            char a;
            scanf("%s",&a);
            if(a == 's') {
                set_data(data);
                i=1;
            } else if (a=='n') {
                free(data);
                exit(1);
            }
        }
        usleep(50);
    }
}

```

O código completo segue em anexo no fim do documento.

IX. RESULTADO DO PONTO DE CONTROLE 3

Infelizmente não foi possível realizar a iluminação gradual do ambiente devido a incompatibilidade do relé utilizado e a geração de pulsos da raspberry, o tempo de subida e descida dos da polarização da bobina interna ao relé é grande demais. o que pode ser solucionada com a troca do módulo por um dimmer. Devido ao tempo de projeto não foi possível implementar o display LCD ainda, apesar do teclado estar funcionando bem. O resto do projeto foi testado e implementado como esperado, a partir deste ponto pode-se então trabalhar no aprimoramento do despertador, melhorando a parte estética e implementado novas funções.

X. REVISÃO BIBLIOGRÁFICA

Existem diversos tipos de despertadores com funções diferentes por aí, alguns deles estão listados abaixo.

1. Clocky



Figura 4 – Clocky

O clocky tem em seu diferencial “fugir” do usuário quando desperta, algo semelhante a ter que levantar e resolver um minigame ao acordar, como proposto no projeto, porém de uma maneira mais chamativa e criativa.

2. Wake-up light alarm



Figura 5 - Wake-up light alarm

Este relógio acorda o usuário com rádio FM e uma luz que simula o nascer do sol, algo semelhante ao proposto neste projeto.

Além destes existem outros com funções menos criativas como controle remoto, utilização por aplicativo, entrada para pendrives, etc...

XI. ANEXO

A. Código PC3-1.c

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <string.h>

char* le_arq_texto(char *nome_arquivo);
int tam_arq_texto(char *nome_arquivo);
void tratamento_SIGUSR1();

int main(int argc, char *argv[]){
    signal(SIGUSR1, tratamento_SIGUSR1);
    while(1)
        system("aplay -q Music.wav");
    return 0;
}

void tratamento_SIGUSR1() {
    char *b;
    char dt[40] = "(kill -9 ";
    char f[] = ") &>/dev/null";
    system("rm -f a.txt");
    system("pidof aplay | tee a.txt");
    b = le_arq_texto("a.txt");
    strcat(dt,b);
    strcat(dt,f);
    system(dt);
    exit(1);
}

char* le_arq_texto(char *nome_arquivo){
    FILE *arq = fopen(nome_arquivo, "r");
    int long g = tam_arq_texto(nome_arquivo);
    char *a;
    a = malloc(g+3);
    fread(a, 1, g, arq);
    fclose(arq);
    return a;
}
```

```

    }

    int tam_arq_texto(char *nome_arquivo){
        FILE *arq = fopen(nome_arquivo, "r");
        fseek(arq, 0, SEEK_END);
        int size = ftell(arq);
        return size;
    }

```

B. PC3-2

```

#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#include <errno.h>
#include <signal.h>
#include <string.h>
#include <wiringPi.h>
#include <time.h>

void tocar_despertador();
void setup_io();
void loop();
int* set_data(int *data);
int t = 400000;
int a=0;
void win_animation();
void lose_animation();
void troca_led(int signum);
int n=0;

int main(int argc, char *argv[]){
    setup_io();
    loop();
    return 0;
}

void loop(){
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);
    printf("Agora: %d-%d-%d %d:%d:%d\n", tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday,
tm.tm_hour, tm.tm_min, tm.tm_sec);

```

```

int* data;
data = malloc(5);
set_data(data);
printf("Despertar em : %i/%i %i:%i\n", data[1], data[0], data[2], data[3]);
fflush(stdout);
int i = 1;
while(1) {
    t = time(NULL);
    tm = *localtime(&t);

    if(((i==1)&&(tm.tm_mon+1)==data[0])&&(tm.tm_mday==data[1])&&(tm.tm_min==data[3])&&(tm.
tm_hour==data[2])) {
        tocar_despertador();
        i=0;
        sleep(1);
        printf("digite s para ajustar um novo alarme ou n para sair: ");
        char b;
        scanf("%s",&b);
        if(b == 's') {
            set_data(data);
            i=1;
        } else if (b=='n') {
            free(data);
            exit(1);
        }
    }
    usleep(50);
}

int* set_data(int *data){
    printf("Digite o mês para despertar: ");
    scanf("%d", &data[0]);
    if(data[0]==55)
        tocar_despertador();
    while(data[0]>12) {
        printf("Digite um mês válido: ");
        scanf("%d", &data[0]);
    }
    printf("Digite o dia para despertar: ");
    scanf("%d", &data[1]);
    while(data[1]>31) {
        printf("Digite um dia válido: ");
        scanf("%d", &data[1]);
    }
}

```

```

    }
    printf("Digite a hora para despertar: ");
    scanf("%d", &data[2]);
    while(data[2]>24) {
        printf("Digite uma hora válida: ");
        scanf("%d", &data[2]);
    }
    printf("Digite o minuto para despertar: ");
    scanf("%d", &data[3]);
    while(data[3]>60) {
        printf("Digite um minuto válido: ");
        scanf("%d", &data[3]);
    }
    return data;
}

void setup_io() {
    if(wiringPiSetup() == -1){
        printf("deu ruim no setup!\n");
        exit(1);
    }
    pinMode(0,OUTPUT);
    pinMode(1,OUTPUT);
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(5,INPUT);
    digitalWrite(6,HIGH);
    signal(SIGALRM, troca_led);
}

void tocar_despertador() {
    printf("tocando alarme!\n");
    digitalWrite(6,LOW);
    int chlld;
    int i = 0;
    chlld = fork();
    if(chlld==0){
        char *arg[] = {".emb1",NULL};
        execvp(arg[0], arg);
        exit(1);
    }
    ualarm(t,0);
}

```



```

while(i==0) {
    int d = 0;
    while(d==0){
        if(digitalRead(5)==1){
            ualarm(0,0);
            if(a==2){
                win_animation();
                n++;
                if(n==1)
                    t=200000;
                if(n==2)
                    t=100000;
                if(n==3){
                    kill(chilld, SIGUSR1);
                    digitalWrite(a,LOW);
                    digitalWrite(6,HIGH);
                    d=1;
                }
            } else {
                lose_animation();
            }
            digitalWrite(a,LOW);
            a=4;
            ualarm(t,0);
        }
    }

    ualarm(0,0);
    int c;
    ac=fork();
    if (c==0) {
        system("python -m webbrowser
        \"https://www.youtube.com/watch?v=k7BhL96l-jA\" &> /dev/null/a.txt");
        exit(1);
    }
    sleep(15);
    i++;
}

}

void win_animation(){
    for(int i=0;i<25;i++){
        usleep(20000);
        digitalWrite(a,LOW);
        usleep(20000);
    }
}

```

```
        digitalWrite(a,HIGH);
    }
}

void lose_animation(){
    for(int i=0;i<5;i++){
        usleep(40000);
        digitalWrite(a,LOW);
        usleep(40000);
        digitalWrite(a,HIGH);
    }
}

void troca_led(int signum){
    digitalWrite(a,LOW);
    a=(a+1)%5;
    digitalWrite(a,HIGH);
    ualarm(t,0);
}
```