

# ANEXO I

Versão final do código apresentado como projeto final da disciplina de Sistemas Embarcados  
(Julho de 2017)

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>
#include<pthread.h>

//função para realizar o upload dos arquivos para a pasta do dropbox
void *upload(){

    system("/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/Desktop/Embarcados/*.jpg Fotos");
    system("/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/Desktop/Embarcados/*.gif Fotos");

}

//função para mostrar o gif de aguarde enquanto o gif é montado
void *mostraGif_a(){

    system("animate -size 1280x960 -loop 10
/home/pi/Desktop/Embarcados/fixos/aguarde.gif"); //mostrar gif formado

}

//função para mostrar o gif formado com as fotos tiradas e o gif
explicativo de como baixar as fotos.
void *mostraGif(){

    system("animate -size 1280x960 -loop 25
/home/pi/Desktop/Embarcados/*.gif"); //mostrar gif formado
    system("animate -size 1280x960 -loop 5
/home/pi/Desktop/Embarcados/fixos/upload.gif");

}

//função para montar o gif usando o programa graphicsmagick
void *montaGif(){

    system("gm convert -delay 50 *.jpg gif$(date
+'%d%m%Y_%H%M%S').gif");

}

int main(){

    int code;
    FILE *decode;
    pthread_t t1, t2, t3, t4;
    int rc;
    int buffer;
    int button;
    FILE *button_f;
```

```

        button_f=fopen("buf.txt", "rw");
        system("echo 12 > /sys/class/gpio/export");
        system("sudo echo in > /sys/class/gpio/gpio12/direction");
        system("cat /sys/class/gpio/gpio12/value >
/home/pi/Desktop/Embarcados/buf.txt");
        fscanf(button_f,"%d", &button);
        //printf("%d\n", button);
        buffer=button;
        fclose(button_f);

        //loop para mostrar o gif inicial enquanto o botão não for acionado
        while(button==buffer)
        {
            system("animate -loop 1
/home/pi/Desktop/Embarcados/fixos/inicio.gif");
            button_f=fopen("buf.txt", "rw");
            system("cat /sys/class/gpio/gpio12/value >
/home/pi/Desktop/Embarcados/buf.txt");
            fscanf(button_f,"%d", &buffer);
            //printf("%d\n", buffer);
            fclose(button_f);
        }

        decode=fopen("qrcode.txt", "rw");
        while(1){ //loop principal

            system("raspistill -o foto.jpg");//tirar a foto que será
enviada para o ZBar

            system("zbarimg --raw foto.jpg > qrcode.txt"); //enviar a
leitura do ZBar para um arquivo

            fscanf(decode,"%d", &code); //armazenar o valor do arquivo em
uma variável
            printf("%d\n", code);

            //evitar que o programa tire fotos sem a apresentação do QR
Code correto
            if(code==0 || code>8){
                printf("Falha na leitura do QRCODE\n");
                break;
            }

            //Efeito P&B
            if (code==1){
                system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -cfx 128:128 -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
            }

            //Efeito Azul
            if (code==2){
                system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -cfx -58:0 -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
            }

            //Efeito shade

```

```

        if (code==3){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -awb shade -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
        }

        //Efeito sun
        if (code==4){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -awb sun -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
        }

        //Efeito sketch
        if (code==5){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -ifx sketch -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
        }

        //Efeito colourswap
        if (code==6){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -ifx colourswap -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
        }

        //Efeito cartoon
        if (code==7){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -ifx cartoon -o imagem$(date
+'%d%m%Y_%H%M%S')%01d.jpg");
        }

        //Efeito normal
        if (code==8){
            system("raspistill -t 4000 -tl 1000 -w 1280 -h 960 -q
60 -ex antishake -ifx blur -o imagem$(date +'%d%m%Y_%H%M%S')%01d.jpg");
        }

```

```

system("rm foto.jpg"); //apagar a imagem enviada para o ZBar

```

```

//inicia as threads de montar o gif e mostrar o gif enquanto
espera o gif ser montado

```

```

rc=pthread_create(&t1, NULL, mostraGif_a, NULL);
rc=pthread_create(&t2, NULL, montaGif,NULL);

```

```

pthread_join(t1, NULL);
pthread_join(t2, NULL);

```

```

//inicia as threads para subir as fotos e o gif enquanto
mostra o gif montado

```

```

rc=pthread_create(&t3, NULL, upload, NULL);
rc=pthread_create(&t4, NULL, mostraGif,NULL);

```

```

pthread_join(t3, NULL);
pthread_join(t4, NULL);

```

```

//move as imagens que já foram armazenadas no dropbox
system("mv *.jpg home/pi/Desktop/Embarcados/Fotos");
system("mv *.gif /home/pi/Desktop/Embarcados/GIFs");

//lopp para mostrar gif enquanto botao nao for apertado
button_f=fopen("buf.txt", "rw");
system("echo 12 > /sys/class/gpio/export");
system("sudo echo in > /sys/class/gpio/gpio12/direction");
system("cat /sys/class/gpio/gpio12/value >
/home/pi/Desktop/Embarcados/buf.txt");
fscanf(button_f,"%d", &button);
//printf("%d\n", button);
buffer=button;
fclose(button_f);

//lopp para mostrar gif enquanto botao nao for apertado
button=buffer;
while(button==buffer)
{
    system("animate -loop 1
/home/pi/Desktop/Embarcados/fixos/inicio.gif");
    button_f=fopen("buf.txt", "rw");
    system("cat /sys/class/gpio/gpio12/value >
/home/pi/Desktop/Embarcados/buf.txt");
    fscanf(button_f,"%d", &buffer);
    //printf("%d\n", buffer);
    fclose(button_f);
}

//evita que a raspberry utilize a leitura do qrcode anterior
code=9;
fprintf(decode,"%d", &code);
}

fclose(decode);
system("echo 12 > /sys/class/gpio/unexport");

return 0;
}

```