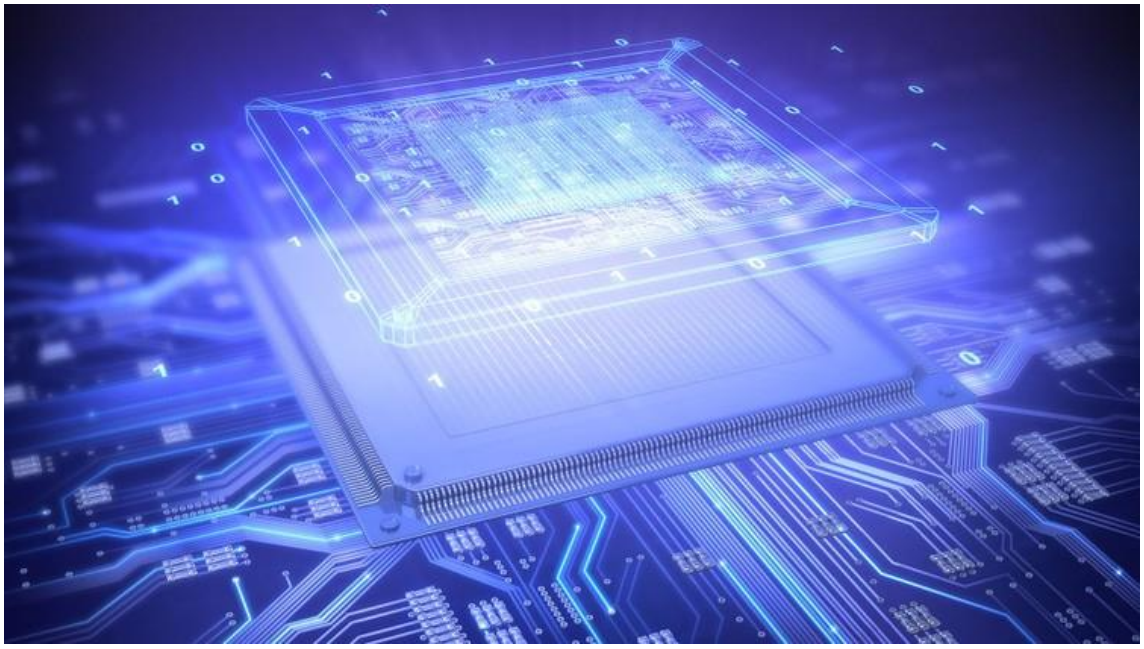
UNIVERSIDADE da MADEIRA

Arquitetura de Computadores

2018/19

Candeeiro Inteligente – 3º Trabalho Prático



Aluno: Ricardo David da Silva Briceño (2032917)

Docentes: Dionísio Barros, Pedro Camacho, Sofia Inácio e Nuno Ferreira

Índice

Introdução / Objetivos

Foi proposto pelos docentes da disciplina desenvolver um programa que controlasse a luz de um candeeiro em linguagem Assembly e C utilizando um microcontrolador 8051. Para simulações do programa utilizou-se o software Keil uVision5.

Como estrutura base, foi reservado o **pino P1.0** para a saída do programa, que indica se a luz está ligada ou desligada. O **pino P1.1** tem como função ligar ou desligar a luz do candeeiro ao ser pressionado, alterando o estado da porta P1.0.

Os **pinos P3.2** e **P3.3**, têm como objetivo alterar a intensidade luminosa da luz do candeeiro e ligar ou desligar a luz do candeeiro através de uma sequência de palmas, respetivamente.

Tinha também como objetivo a realização dos fluxogramas do programa principal e das respetivas interrupções, demonstrando assim a "ideia" em que o programa se baseia e ordem de implementação.

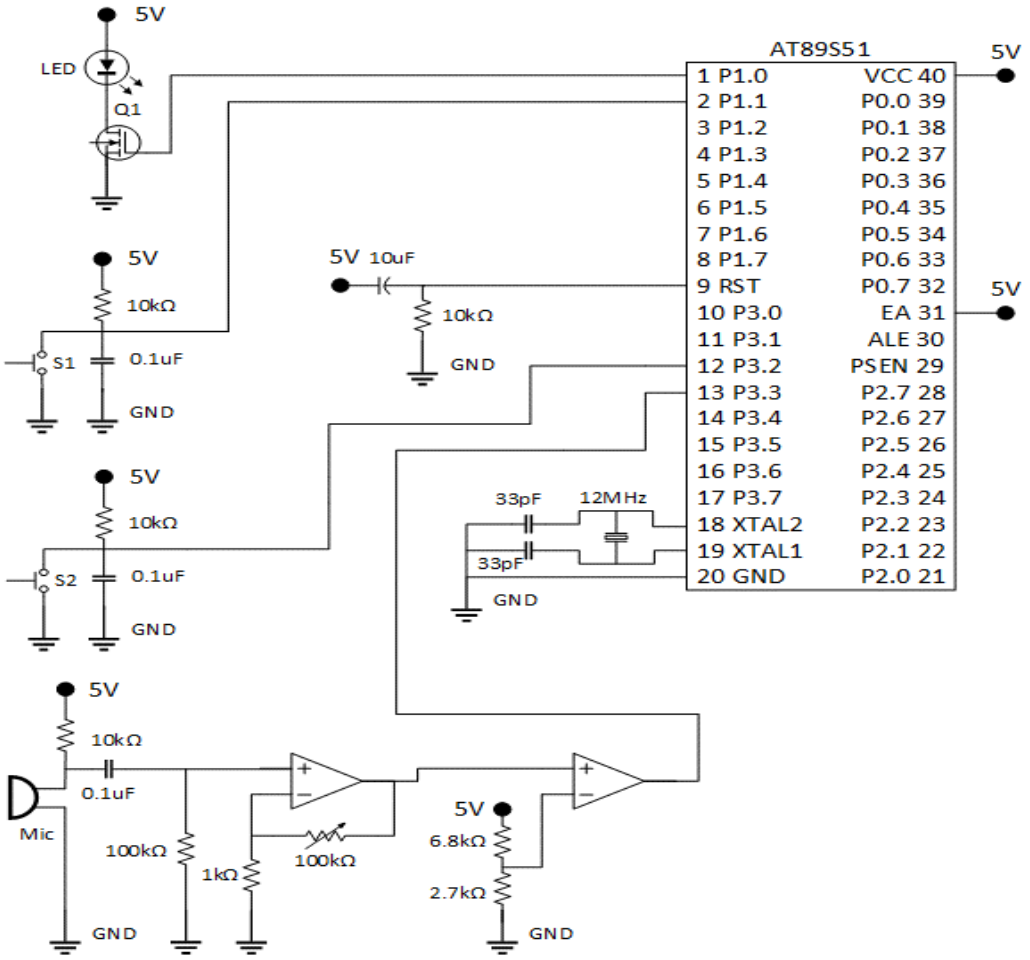


Figura 1 - Esquema das ligações do controlador de um candeeiro.

Discussão

Após perceber de forma simplificada como funcionava o microcontrolador 8051 começou-se por implementar o programa em linguagem C, pois é mais intuitivo e oferece mais clareza no código.

A implementação em C serviu de apoio para a implementação em linguagem Assembly, a programação neste tipo de linguagem deve ser feita sempre que possível pois reduz significativamente o tempo de execução do programa otimizando assim o mesmo. Passando para a implementação do programa em linguagem C, foram implementadas ***5* funções**:

Init: Esta função é responsável por ativar todas as interrupções que foram utilizadas no programa, determinar o modo utilizado no Timer e determinar como o Timer irá contar (Foi apenas utilizado um Timer (Timer0)).

Liga: Esta função é responsável por ligar ou desligar a luz do candeeiro dependendo do seu estado atual de intensidade luminosa**(*)**. Caso o botão **on_off** (P1.1) seja pressionado (**Pressionado = 0** / Não Pressionado = 1) a função deteta qual a sua intensidade luminosa atual, ligando se a intensidade luminosa for 0% e desligando se a sua intensidade luminosa estiver entre 25%, 50%, 75% ou 100%. Ao ligar alterasse a intensidade luminosa a 100% e ao desligar alterasse para 0%.

Intensidade: Está função é responsável por mudar a intensidade luminosa da luz do candeeiro. A intensidade luminosa irá ser mudada dependendo da sua intensidade luminosa atual (25% -> 50% -> 75% -> 100% -> 25% …). Esta função só é executada quando é pressionado o botão da intensidade luminosa **(P*3.2*)**. Ao pressionar o botão o programa "pará" e vai tratar a interrupção externa (interrupção 0) colocando o botão com o valor 0 (indica que o botão foi pressionado). Após a indicação que o botão foi pressionado, a função tem autorização de mudar a intensidade luminosa.

Duty: Esta função é responsável por criar o efeito de Fade (intensidade luminosa) em Loop. A ideia foi comparar o tempo "ativo" da

intensidade com o contador do Timer, quando este fosse alcançado, o candeeiro "desliga" e quando o contador alcançasse o período final voltava a "ligar" criando o efeito de intensidade luminosa. Quando o candeeiro se encontra nestas situações o que realmente esta acontecer é que liga e desliga constantemente a uma velocidade tão rápida que o olho humano não consegue detetar. Este tipo de implementação cria o efeito pretendido.

Palmas: Esta função é responsável por ligar ou desligar a luz do candeeiro com uma sequência de **2 palmas**, caso o microfone não detete 2 palmas num intervalo de 1.5s após ter detetado a primeira o candeeiro é capaz de reconhecer que pode não ter sido intencional e não contabiliza as palmas. Esta função é executada quando o microfone deteta uma palma, o programa "pará" e trata a interrupção externa (interrupção 1) incrementando o contador de palmas, o contador específico para o microfone começa a contar a partir desse momento e a interrupção externa é desativada. Este contador irá passar por 2 fases. A **fase de ruido** e a **fase de detenção** da segunda palma.

Quando é detetada 1 palma, o microfone entra em estado de "silêncio" (fase de ruido), isto é, a interrupção é desativada para garantir que não é detetado ruídos nesse período de tempo (50ms). Após passarem os 50ms e a fase de ruido tiver concluído voltamos a ativar a interrupção pronta para detetar mais uma palma.

Quando são detetadas as 2 palmas**(**)**, o programa tem autorização para ligar ou desligar a luz do candeeiro. Para isso irá, primeiramente, verificar qual é a sua intensidade luminosa atual e dependendo disso ligar ou desligar.

(*) A maneira mais eficiente que se conseguiu implementar para detetar se o candeeiro esta ligado ou desligado foi comparar a sua intensidade luminosa e não o estado atual do candeeiro propriamente dito pois iria dar problemas caso a intensidade estivesse entre 25% e 75% já que quando a intensidade se encontra nestes valores, o programa estará a executar a função Duty,

fazendo fade do candeeiro (liga e desliga constantemente para fazer o efeito de intensidade luminosa).

Se por acaso o programa detetasse que o candeeiro se encontra na parte negativa do Duty, iria dar falsa informação (desligado) comprometendo assim a funcionalidade correta do programa e por sua vez do funcionamento do candeeiro.

(**) As duas palmas só são detetadas se a interrupção for tratada 1 segundo e meio após ter sido tratada 1 vez. Isto é, se for detetada uma palma, o programa espera 1.5s que seja detetada a segunda palma, caso neste intervalo de tempo sejam detetadas 2 palmas, a função liga ou desliga a luz do candeeiro como explicado.

Caso contrário, o candeeiro faz reset do contador de palmas e do respetivo contador do microfone.

A implementação em linguagem Assembly foi mais complexa, notou-se pela quantidade de instruções utilizadas. Para implementar o programa foram implementadas ***5* rotinas**:

As rotinas implementadas em linguagem Assembly **Init, F_on_off, Muda_intensidade_luminosa, DutyCycle** e **Palmas,** têm a mesma funcionalidade que Init, Liga, Intensidade, Duty e Palmas na linguagem C, respetivamente. Tentou-se mapear de forma quase idêntica o programa em linguagem C, contudo, a complexidade da linguagem Assembly e da sua limitação de instruções levaram a extensão de código, o que não foi, de todo, um problema, pois como já foi referido, esta linguagem tem a vantagem de reduzir o tempo de execução do programa.

Conclusão

Conseguiu-se implementar todas as funcionalidades exigidas em ambas as linguagens de programação. A ferramenta Keil uVision5 serviu de auxílio e ajudou a compreender a manipulação que é feita tanto nos registos como nas interrupções, contribuindo assim para o meu domínio pessoal de microcontroladores.

A realização deste trabalho foi bastante complexa no que toca ao domínio de todas as características do microcontrolador, na minha opinião é bastante detalhado e requer amplo conhecimento para uma boa implementação de programas. Reconheço que se tivesse um domínio mais aprofundado deste tema conseguiria reconstruir o meu programa para outro mais simples e eficiente, tudo o que os programadores mais prezam.

O mapeamento para a linguagem Assembly deu alguns problemas, como esperado, teve-se de recorrer aos docentes e a informação online para resolver alguns dos problemas que foram surgindo.

Mesmo tendo o programa em linguagem C como base é bastante complexo ter um bom raciocino em Assembly, tentou-se ao máximo mapear tal e qual, mas nem sempre foi possível, daí a justificação da extensão de código em linguagem Assembly.

Para uma melhor perceção da minha ideia comentei cada instrução em ambas as linguagens, contudo, é sabido que é um bocado abstrato trabalhar com este tipo de linguagem, preferindo por esta razão trabalhar em linguagem C.

Bibliografia

- http://www.keil.com/support/man/docs/c51/c51_le_datatypes.htm
- http://www.keil.com/support/docs/814.htm
- Arquitectura do 8051 – Dionísio Barros (Apontamentos baseados no livro: Philips, "80C51- Based 8-Bit Microcontrollers, Data Handbook IC20", Philips Semiconductors, 1995)

Anexo A

Decidiu-se apresentar os fluxogramas do programa relativamente a linguagem de alto nível C de modo a ser mais claro e simples

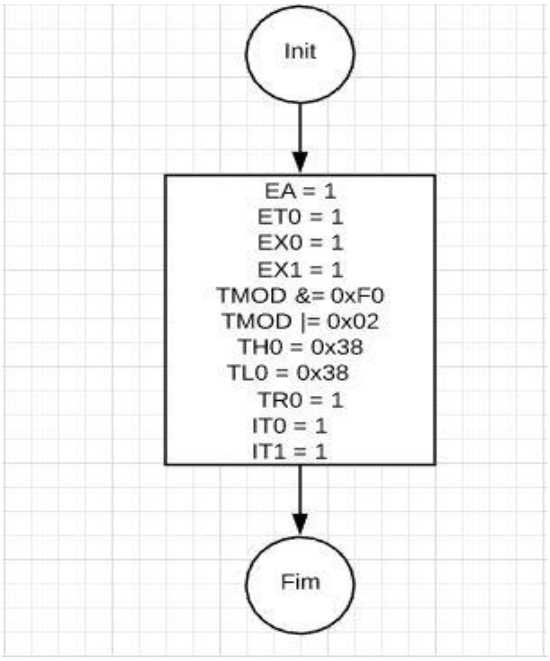


Figura 2- Função Init

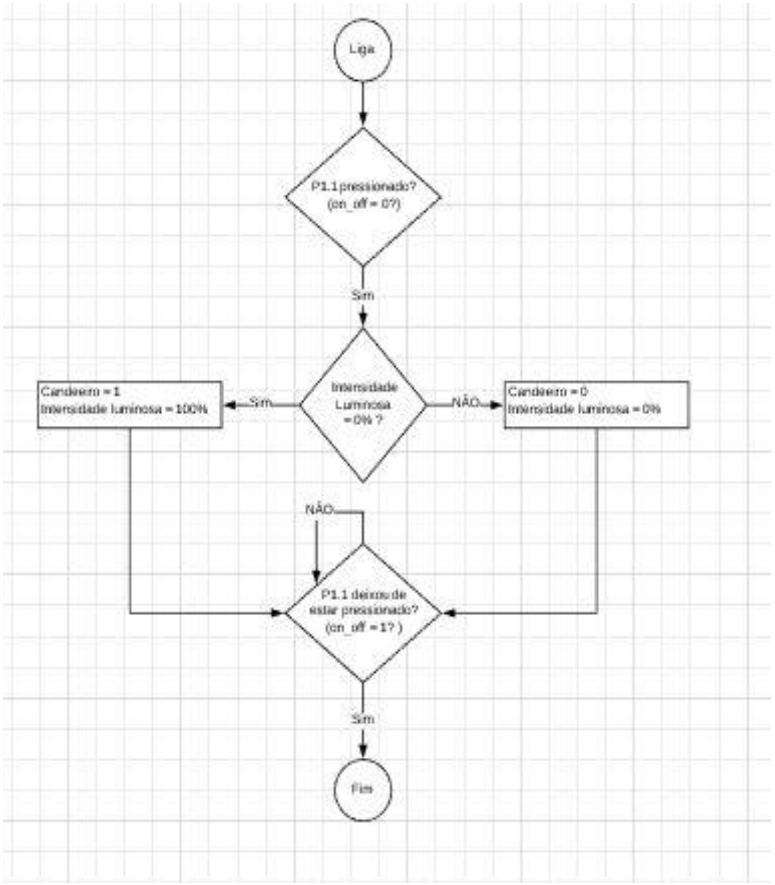


Figura 3 - Função Liga

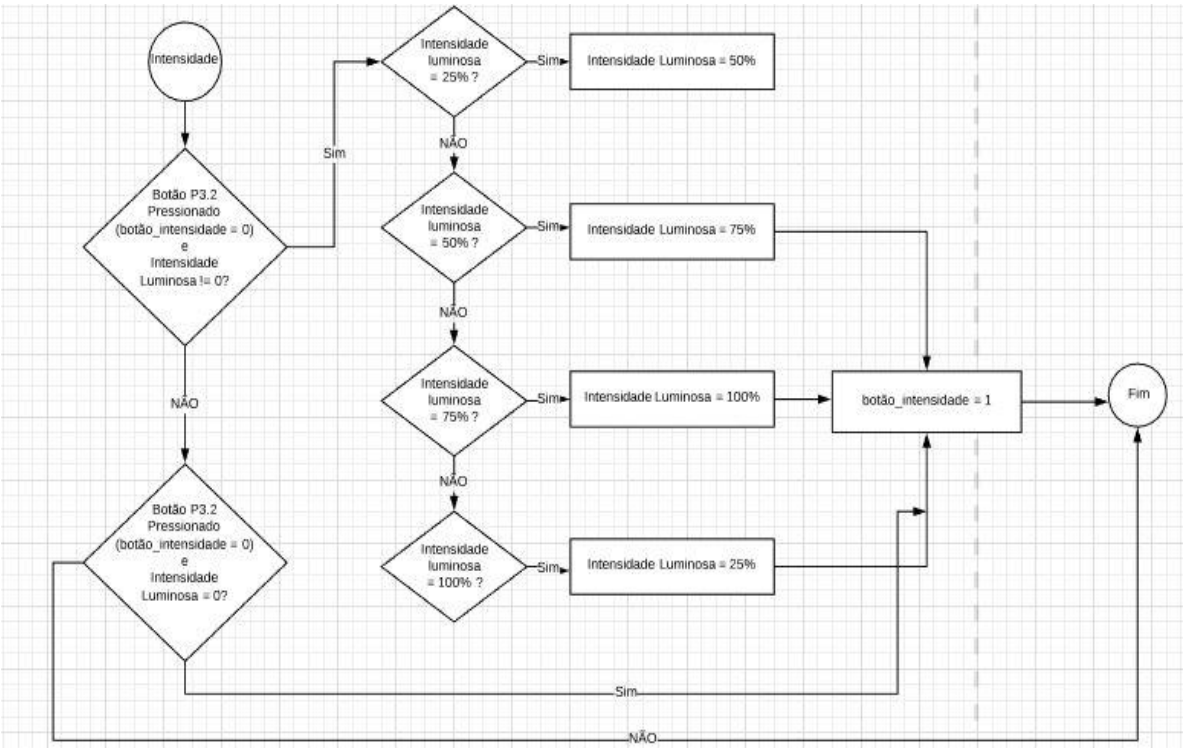


Figura 4 - Função Intensidade

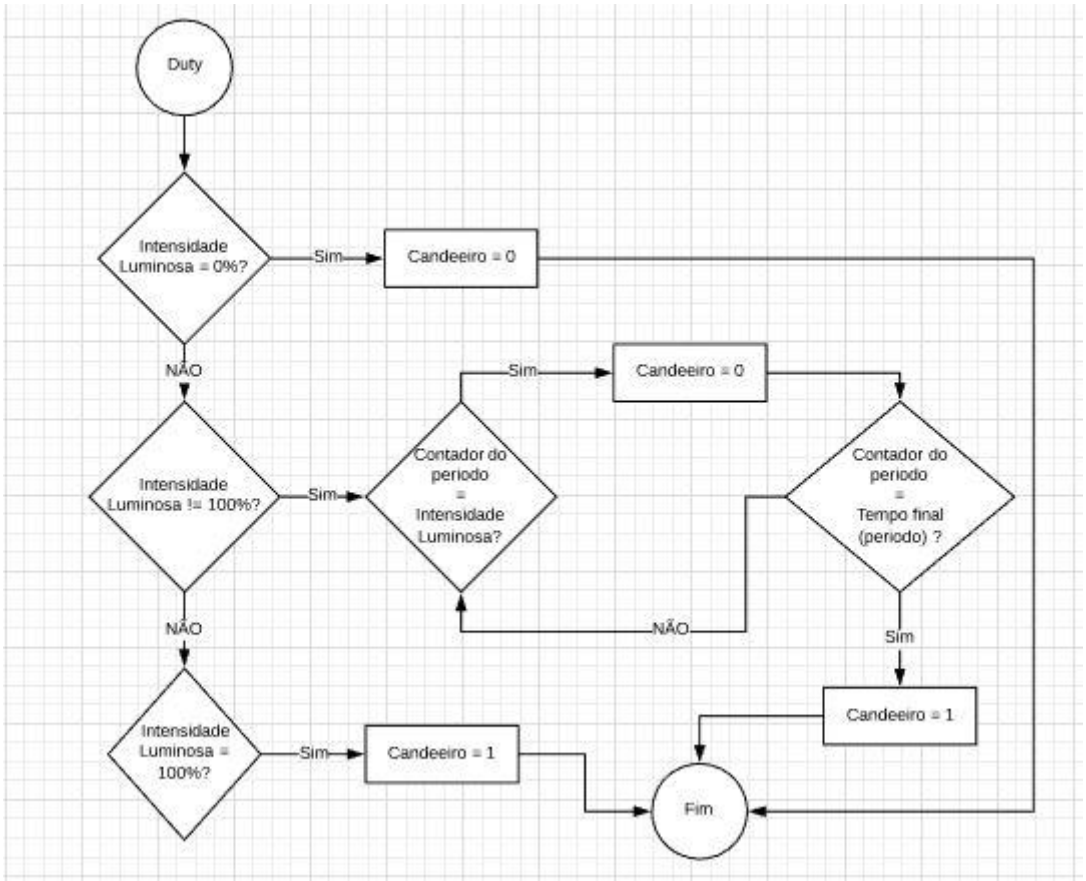


Figura 5 - Função Duty

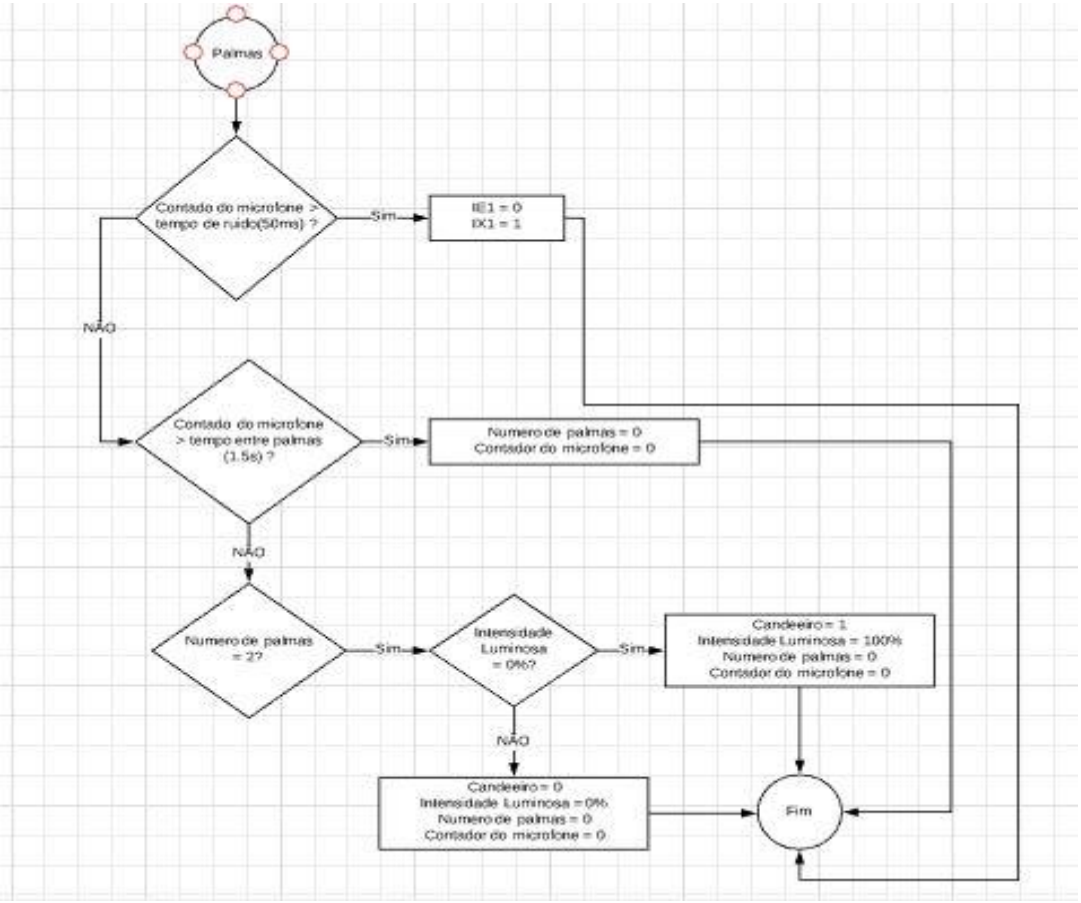


Figura 6 - Função Palmas

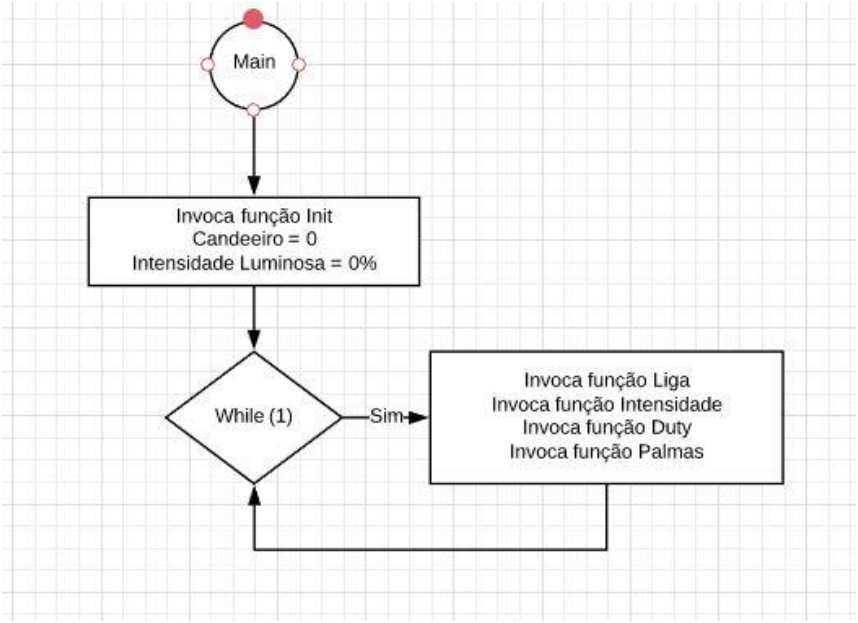


Figura 7 - Main

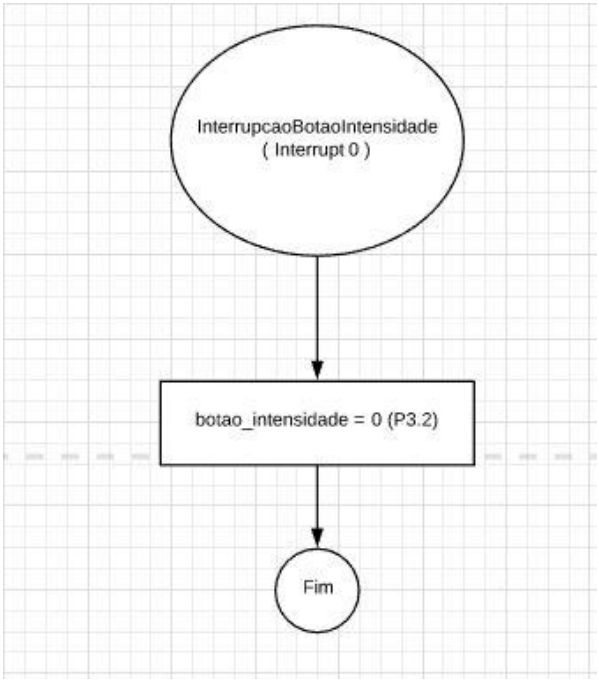


Figura 8 - Interrupção 0

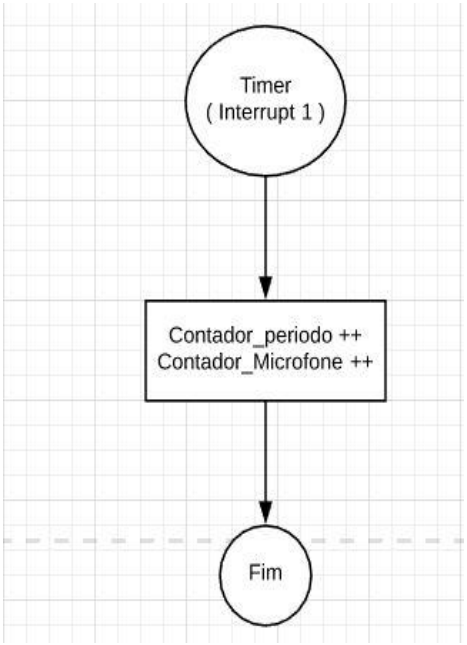


Figura 9 - Interrupção 1

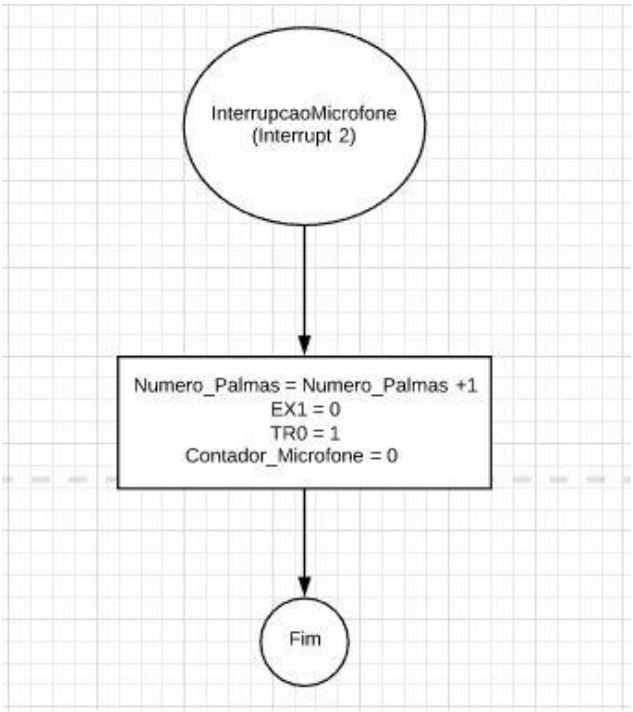


Figura 10 - Interrupção 2

Anexo B

Linguagem C:

```
#include <reg51.h>

#define TEMPO_FINAL 100                                  //periodo 20ms (0.2ms x 100 = 20
ms)

#define ZERO 0                                           //0% do periodo (0.2ms x
0 = 0ms)                    <----

#define VINTE_CINCO 25                                   //25% do periodo (0.2ms x 25 =
5ms)                                |

#define CINQUENTA 50                                     //50% do periodo (0.2ms x 50 =
10ms)              |--->  D u t y  C y c l e

#define SETENTA_CINCO 75                                 //75% do periodo (0.2ms x 75 =
15ms)          |

#define CEM 100                                          //100% do periodo (0.2ms
x 100 = 20ms)     <----

#define TEMPO_RUIDO  250                                 //50ms = (0.2ms x 250) --> tempo
de ruido (vibrações da palma)

#define TEMPO_ENTRE_PALMAS 7500                          //1.5s = (7500 x 0,2ms) --> Se
passarem mais de 1.5s segundos depois da primeira palma, o candeeiro reconhece que não bateu a
palma (deve bater 2 palmas num periodo até 1.5segundos)


sbit candeeiro = P1^0;                                   //Porta reservada para a
saida --> Candeeiro

sbit on_off = P1^1;                                              //Porta reservada
para ligar / desligar o Candeeiro


bit botao_intensidade = 1;                               //Botão que se for pressionado
altera a intensidade luminosa (P3^2)


unsigned char intensidade_luminosa = CEM;        //Referencia BASE da intensidade

unsigned char contador_periodo = 0;                      //contador do periodo

unsigned char numero_palmas = 0;                 //Contador de numero de palmas

unsigned char contador_microfone = 0;            //Contador para o microfone


void Init(void);

void liga(void);

void intensidade (void);

void duty (void);
```

```
void palmas (void);



/*
-------------------------------------------------
-------------------------------------------------
-------------------------------------------------
---C A N D E E I R O  I N T E L I G E N T E ---
-------------------------------------------------
-------------------------------------------------
-------------------------------------------------
*/

void Init(void){    // ativação das interrupções

        EA = 1;             //enable das interrupcoes globais
        ET0 = 1;                    // ativa interrupcao timer 0
        EX0 = 1;                    // ativa interrupcao externa 0
        EX1 = 1;                    //ativa interrupcao externa 1

        TMOD &= 0xF0;  //limpa os 4 bits do timer 0 (8 bits – auto reload)
        TMOD |= 0x02;   //modo 2 do timer 0

        TH0 = 0x38;       //Timer 0 - 0.2ms
        TL0 = 0x38;

        TR0 = 1;                    //indica que o timer 0 começa a contar
        IT0 = 1;            //interrupcao externa 0 activa a falling edge (1 -> 0)
        IT1 = 1;            //interrupcao externa 1 activa a falling edge (1 -> 0)
}

void InterrupcaoBotaoIntensidade(void) interrupt 0{
        // Tratamento da interrupcao ao pressionarmos o botão da intensidade
        botao_intensidade = 0;
                                        // botão foi pressionado
}

void Timer (void) interrupt 1{
                                        // Tratamento da interrupcao do Timer 0
```

```
	contador_periodo++;
                                        // Contador do periodo é incrementado a cada 0.2ms
	contador_microfone++;
                                        // Contador do microfone é incrementado a cada 0.2ms

}


void InterrupcaoMicrofone(void) interrupt 2{
	// Tratamento da interrupcao se o microfone detetar uma palma
	numero_palmas++;
                                        //deteta a palma (contador de palmas incrementa)
	EX1 = 0;
                                        //Desligamos a interrupção do microfone para nao
detetar ruido
	TR0 = 1;
                                        //Indicação que o timer começa a contar quando é
ativada esta interrupção
	contador_microfone = 0;
                                        //O contador do microfone é reseteado para garantir uma
contagem correta
}


void liga (void){
	/*-------------------------------------------
	----------LIGAR / DESLIGAR CANDEEIRO-----------
	-------------------------------------------------*/


	if(on_off == 0){
	//se o botão on_off foi pressionado, dependendo da luminosidade atual, vai ligar ou desligar e
regular a sua intensidade luminosa
		if(intensidade_luminosa == 0){
			candeeiro = 1;
			intensidade_luminosa = CEM;
		}
		else{
			candeeiro = 0;
			intensidade_luminosa = ZERO;
		}
		while(on_off == 0){}
	//Espera que o botão on_off deixe de ser pressionado
	}
}


void intensidade (void){
	/*-------------------------------------------
```

```
	----------INTENSIDADE CANDEEIRO----------

	----------------------------------------------*/

	if(intensidade_luminosa != 0 && botao_intensidade == 0){	// Caso o candeeiro esteja ligado e pressionarmos o botão para aumentar a intensidade(esta muda conforme a sua intensidade atual (25% -> 50% -> 75%  -> 100% -> 25% ...)

		switch(intensidade_luminosa){

			case VINTE_CINCO:

				intensidade_luminosa = CINQUENTA;
	//intensidade 25% -> 50%

				break;

			case CINQUENTA:

				intensidade_luminosa = SETENTA_CINCO;
	//intensidade 50% -> 75%

				break;

			case SETENTA_CINCO:

				intensidade_luminosa = CEM;
	//intensidade 75% -> 100%

				break;

			case CEM:

				intensidade_luminosa= VINTE_CINCO;
	//intensidade 100% -> 25%

				break;

			default:

				break;

		}

		botao_intensidade = 1;
	//Botao da intensidade é reiniciado

	}

	if(intensidade_luminosa == 0 && botao_intensidade == 0){	//Caso o candeeiro esteja desligado e for pressionado o botão para aumentar a intensidade, não faz nada, pois para mudar deve estar ligado

		botao_intensidade = 1;
	//Botao da intensidade é reiniciado

	}

}


void duty (void){

	/*----------------------------------------------

	------------------DUTY CYCLE ---------------------

	-----------------------------------------------*/


	if(intensidade_luminosa == ZERO){					//Se a intensidade for 0 significa que o Candeeiro esta OFF, e por tanto não alteramos o seu estado (So podemos alterar a intensidade luminosa se estiver ligado)
```

```
                candeeiro = 0;
        //Candeeiro OFF

        }

        else if(intensidade_luminosa != CEM){                                   //Se a
intensidade estiver entre 25 e 75 **(já passou a condição de ser 0 e rejeita ser 100)**

                if(contador_periodo == intensidade_luminosa){

                        candeeiro = 0;
        // o candeeiro desliga quando iguala o valor ATIVO do duty cycle

                }

                if(contador_periodo == TEMPO_FINAL){

                                contador_periodo = 0;

                                candeeiro =1;
        //Quando atinge o periodo final (20ms) o contador reinicia para nova contagem de periodo e o
candeeiro liga para restabelecer o duty cycle

                }

        }

        else{

                candeeiro = 1;
        // Se a intensidade for 100 , candeeiro liga

        }

}


void palmas (void){

        /*-----------------------------------------------

        ----------LIGAR / DESLIGAR COM PALMAS -----------

        ----------------------------------------------*/



        if(contador_microfone == TEMPO_RUIDO){                                  //Se o
contador do microfone ultrapassar o tempo de ruido predefinido (50ms)

                IE1 = 0;
                //Limpamos a flag para não guardar o valor anterior

                EX1 = 1;
                //Reativamos a interrupção externa (microfone)

        }

        if(contador_microfone == TEMPO_ENTRE_PALMAS){                           //Se o
contador do microfone ultrapassar o tempo entre palmas (1.5s)

                numero_palmas = 0;
                //O microfone reconhece que não foi dada nenhuma palma

                contador_microfone = 0;
        //O contador é reiniciado, pronto para uma nova interrupção

        }

        else{
```

```
		if(numero_palmas == 2){
	//Se forem detetadas 2 palmas

			if(intensidade_luminosa == ZERO){ //Se a
intensidade luminosa for 0 (Candeeiro desligado) -> o candeeiro liga com intensidade luminosa 100%

				candeeiro = 1;

				intensidade_luminosa = CEM;

				contador_microfone = 0;
	//Reiniciamos o contador para nova interrupção

				numero_palmas = 0;
		//Reiniciamos o contador de palmas

			}

			else{

				candeeiro = 0;
		//Se a intensidade luminosa for diferente de 0 (Candeeiro ligado) -> o candeeiro é
desligado com intensidade luminosa 0%

				intensidade_luminosa = ZERO;

				contador_microfone = 0;
	//Reiniciamos o contador para nova interrupção

				numero_palmas = 0;
		//Reiniciamos o contador de palmas

			}

		}

	}


void main(void){

	Init();

	intensidade_luminosa = ZERO;
	//intensidade luminosa começa com valor 0 (desligada)

	candeeiro = 0;
		//Candeeiro começa off (desligado)

	while(1){

		liga();

		intensidade();

		duty();

		palmas();

	}

}
```

Linguagem Assembly:

```
TEMPO_FINAL                 EQU 100                     ;periodo 20ms (0.2ms x
100 = 20 ms)

ZERO                              EQU 0                       ;Referencia ao
valor '0'

UM                                      EQU 1
        ;Referencia ao valor '1'

DOIS                              EQU 2                       ;Referencia ao
valor '2'

VINTE_CINCO                 EQU 25                      ;25% do periodo (0.2ms x
25 = 5ms)                   <----|

CINQUENTA                         EQU 50                      ;50% do periodo
(0.2ms x 50 = 10ms)          |---> D u t y  C y c l e

SETENTA_CINCO                     EQU 75                      ;75% do periodo
(0.2ms x 75 = 15ms)         |

CEM                               EQU 100                           ;100%
do periodo (0.2ms x 100 = 20ms)    <----|

TEMPO_RUIDO                 EQU 250                           ;50ms = (0.2ms x
250) --> tempo de ruido (vibrações da palma)

numero_palmas               EQU 0                       ;Contador de numero de
palmas


contador_aux                EQU 100                           ;Como o tempo
entre palmas é 1.5s (7500 x 0.2ms) --> A constante 7500 passa do byte, logo, decidi implementar um
algoritmo que contasse 7500 sem ultrapassar o byte


candeeiro                         EQU P1.0                    ;Porta reservada
para a saida --> Candeeiro

on_off                            EQU P1.1                    ;Porta reservada
para ligar / desligar o Candeeiro


; ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

; ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

; ||||||||||| E X P L I C A Ç Ã O   D O   C O N T A D O R _ A U X |||||||||||||||||

; ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

; ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

; ||||| Contou 100 -> Incrementa Registo --> Valor no Registo: 1 --> Limpa contador ||

; ||||| Contou 100 -> Incrementa Registo --> Valor no Registo: 2 --> Limpa contador ||

; ||||| ETC ... ... ... ... ... ... ... ...  ... ... ... ...  ... ... ... ...  ...  ||

; ||||| Quando o Valor no Registo chegar a 75 significa que contou 7500 ( 75 * 100) ||

; ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

```
CSEG AT 0000h

JMP Main

CSEG AT 0003h

JMP InterrupcaoBotaoIntensidade                    ;Caso seja ativada a interrupção
externa 0, salta para InterrupcaoBotaoIntensidade

CSEG AT 000Bh

JMP Timer                                                                    ;Salta
para Timer para tratar a interrupção do Tomer -> contar periodo de 20ms e contador de microfone

CSEG AT 0013h

JMP InterrupcaoMicrofone                           ;Caso seja ativada a interrupção
externa 1, salta para InterrupcaoMicrofone



;-----------------------------------

;-----------------------------------

;--- I N I C I A L I Z A Ç Õ E S ---

;-----------------------------------

;-----------------------------------

Init:

        SETB EA                                                ;EA = 1  ->
enable das interrupcoes globais

        SETB ET0                                       ;ET0 = 1 -> ativa
interrupcao timer 0

        SETB EX0                                       ;EX0 = 1 -> ativa
interrupcao externa 0

        SETB EX1                                       ;EX1 = 1 -> ativa
interrupcao externa 1


        ANL TMOD, #0F0H                        ;limpa os 4 bits do timer 0
(8 bits – auto reload)

        ORL TMOD, #002H                        ;modo 2 do timer 0


        MOV TH0, #038H                         ;Timer 0 - 0.2ms

        MOV TL0, #038H


        SETB TR0                               ;indica que o timer 0
começa a contar

        SETB IT0                               ;interrupcao externa 0
activa a falling edge (1 -> 0)

        SETB IT1                               ;interrupcao externa 1
activa a falling edge (1 -> 0)

        RET
```

```
;-------------------------------------

;-------------------------------------

;----- I N T E R R U P Ç Õ E S -----

;-------------------------------------

;-------------------------------------


InterrupcaoBotaoIntensidade:                    ;Tratamento da interrupcao ao pressionarmos o
botão da intensidade

        MOV R3, #ZERO                                   ;botão foi pressionado

        RETI                                            ;Retorna


Timer:                                                  ;Tratamento da
interrupcao do Timer 0

        INC R4                                          ;Contador do periodo é
incrementado a cada 0.2ms

        INC R6                                          ;Contador do microfone é
incrementado a cada 0.2ms

        RETI                                            ;Retorna


InterrupcaoMicrofone:                           ;Tratamento da interrupcao se o microfone
detetar uma palma

        INC R5                                          ;deteta a palma (contador
de palmas incrementa)

        CLR EX1                                                 ;Desligamos a
interrupção do microfone para nao detetar ruido

        SETB TR0                                        ;Indicação que o timer
começa a contar quando é ativada esta interrupção

        MOV R6, #ZERO                                   ;Começa a contar o contador
auxiliar do microfone

        MOV R7, #ZERO                                   ;Começa a contar o contador
auxiliar 2 do microfone

        RETI                                            ;Retorna


;************************************************
;-------> R0 - candeeiro                  <-------
;-------> R1 - on_off                             <-------
;-------> R2 - intensidade_luminosa       <-------
;-------> R3 - botao_intensidade          <-------
;-------> R4 - contador_periodo           <-------
;-------> R5 - numero_palmas                <-------
;-------> R6 - contador_aux                 <-------
;-------> R7 - contador_aux2                <-------
```

```
;***********************************************

CSEG AT 0050h

Main:

        MOV R0, #ZERO                                   ;Candeeiro inicialmente off

        MOV R2, #ZERO                                   ;Intensidade luminosa começa com
valor 0 (desligada)

        MOV R3, #UM                                     ;Botão intensidade NÃO
PRESSIONADO

        MOV R4, #ZERO                                   ;Contador do Periodo é 0
inicialmente

        MOV R5, #ZERO                                   ;Numero de palmas é 0
inicialmente

        MOV R6, #ZERO                                   ;Contador do microfone é 0
inicialmente (contador_Aux)

        MOV R7, #ZERO                                   ;Contador do microfone é 0
inicialmente (contador_Aux2)

        CLR Candeeiro                                   ;Candeeiro começa off (desligado)

        MOV SP, #7                                      ;Endereço inicial da stack
point

        CALL Init                                       ;Chamada a rotina Init

Ciclo:

        Call F_on_off                                   ;Chamamos a rotina F_on_off

        Call Muda_intensidade_luminosa          ;Chamamos a rotina Muda_intensidade_luminosa

        Call DutyCycle                                  ;Chamamos a rotina DutyCycle

        Call Palmas                                     ;Chamamos a rotina Palmas

        JMP Ciclo                                       ;Loop Infinito



;-----------------------------------------------

;----------LIGAR / DESLIGAR CANDEEIRO-----------

;-----------------------------------------------


F_on_off:

        JNB on_off, Liga

        RET

Liga:

        MOV R1, #ZERO                                   ;Botão on_off está PRESSIONADO

        CJNE R2, #ZERO, Desliga                         ;Se o candeeiro esta ligado (
intensidade != 0% ), então desligamos com intensidade = 0%

        MOV R0, #UM                                     ;Candeeiro ON

        SETB candeeiro                                  ;Candeeiro ON
```

```
        MOV R2, #CEM                                    ;Intensidade luminosa = 100%
        JMP Pressionado_0
Desliga:
        MOV R0, #ZERO                                   ;Candeeiro OFF
        CLR candeeiro                                   ;Candeeiro OFF
        MOV R2, #ZERO                                   ;Intensidade luminosa = 0%
        JMP Pressionado_0
Pressionado_0:
        JNB on_off, Pressionado_0                       ;Espera que o botão on_off deixe de ser
pressionado
        MOV R1, #UM                                     ;Botão on_off já não esta
PRESSIONADO
        RET
;-----------------------------------------------
;------------INTENSIDADE CANDEEIRO----------------
;-----------------------------------------------
Muda_intensidade_luminosa:
        CJNE R0, #UM, Nada                              ;Caso o
candeeiro esteja desligado, não se pode alterar a intensidade luminosa
        CJNE R3, #UM, CaseVinteCinco                    ;Caso pressionarmos o botão para
aumentar a intensidade(esta muda conforme a sua intensidade atual (25% -> 50% -> 75%  -> 100% ->
25% ...)
        RET
CaseVinteCinco:
        CJNE R2, #VINTE_CINCO, CaseCinquenta            ;Verificamos se a intensidade atual é 25 ou
não
        MOV R2, #CINQUENTA                              ;intensidade 25%
-> 50%
        MOV     R3, #UM
        ;"Limpa" o botão da intensidade luminosa --> " Não pressionado"
        RET
        ;Após mudar a intensidade luminosa, fazemos o DutyCycle
CaseCinquenta:
        CJNE R2, #CINQUENTA, CaseSetentaCinco           ;Verificamos se a intensidade atual
é 50 ou não
        MOV R2, #SETENTA_CINCO                          ;intensidade 50%
-> 75%
        MOV     R3, #UM
        ;"Limpa" o botão da intensidade luminosa --> " Não pressionado"
        RET
        ;Após mudar a intensidade luminosa, fazemos o DutyCycle
CaseSetentaCinco:
```

```
        CJNE R2, #SETENTA_CINCO, CaseCem                ;Verificamos se a intensidade atual
é 75 ou não

        MOV R2, #CEM                                    ;intensidade 75%
-> 100%

        MOV     R3, #UM
        ;"Limpa" o botão da intensidade luminosa --> " Não pressionado"

        RET
        ;Após mudar a intensidade luminosa, fazemos o DutyCycle

CaseCem:

        CJNE R2, #CEM, CaseVinteCinco                   ;Verificamos se a intensidade atual
é 100 ou não

        MOV R2, #VINTE_CINCO                            ;intensidade 100% -> 25%

        MOV     R3, #UM
        ;"Limpa" o botão da intensidade luminosa --> " Não pressionado"

        RET
        ;Após mudar a intensidade luminosa, fazemos o DutyCycle

Nada:
                ;

        MOV R2, #ZERO                                   ;Não muda a
intensidade pois deve estar ligado. Intensidade luminosa = 0%

        MOV     R3, #UM
        ;"Limpa" o botão da intensidade luminosa --> " Não pressionado"

        RET




;-------------------------------------------------

;-----------------DUTY CYCLE--------------------

;-------------------------------------------------


DutyCycle:

        CJNE R2, #ZERO, Else_if_0                       ;Se a intensidade for 0
significa que o Candeeiro esta OFF, e por tanto não alteramos o seu estado (So podemos alterar a
intensidade luminosa se estiver ligado)

        MOV R0, #ZERO                                   ;Candeeiro OFF

        CLR candeeiro                                   ;Candeeiro OFF

        RET

Else_if_0:

        CJNE R2, #CEM, If_1_25                          ;Se a intensidade estiver
entre 25 e 75 **(já passou a condição de ser 0 e rejeita ser 100)**

        MOV R0, #UM                                             ;Se
intensidade luminosa = 100% --> Candeeiro ON

        SETB candeeiro                                  ;Candeeiro ON

        RET

If_1_25:
```

```
        CJNE R2, #VINTE_CINCO, If_1_50                  ;Testamos se a
intensidade é 25%

        CJNE R4, #VINTE_CINCO, If_2                     ;o candeeiro desliga
quando iguala o valor ATIVO do duty cycle

        CLR candeeiro

        JMP     If_2                                    ;Parte
negativa da onda Quadrada

If_1_50:

        CJNE R2, #CINQUENTA, If_1_75                    ;Testamos se a intensidade é 50%

        CJNE R4, #CINQUENTA, If_2                       ;o candeeiro desliga
quando iguala o valor ATIVO do duty cycle

        CLR candeeiro

        JMP     If_2                                    ;Parte
negativa da onda Quadrada

If_1_75:

        CJNE R2, #SETENTA_CINCO, If_1_25                ;Testamos se a intensidade é 75%

        CJNE R4, #SETENTA_CINCO, If_2                   ;o candeeiro desliga
quando iguala o valor ATIVO do duty cycle

        CLR candeeiro

        JMP     If_2                                    ;Parte
negativa da onda Quadrada

If_2:

        CJNE R4, #TEMPO_FINAL, Fim1                     ;Quando atinge o periodo
final (20ms) o contador reinicia para nova contagem de periodo e o candeeiro liga para restabelecer o
duty cycle

        MOV R4, #ZERO

        SETB candeeiro

Fim1:

        RET



;-----------------------------------------------

;--------- LIGAR / DESLIGAR COM PALMAS ---------

;-----------------------------------------------



Palmas:

        CJNE R6, #TEMPO_RUIDO, Fim2                     ;Se o contador
auxiliar ultrapassar o tempo de ruido predefinido (50ms)

        CLR IE1
        ;Limpamos a flag para não guardar o valor anterior

        SETB EX1
        ;Reativamos a interrupção externa (microfone)

If_zero:
```

```
        CJNE R6, #contador_aux, if_zero_zero                ;Se o contador auxiliar for igual ao tempo de referencia (100ms)

        INC R7
        ;Incrementamos o contador auxiliar 2 até chegar a 75

        MOV R6, #ZERO
        ;Limpamos a contagem do contador auxiliar

if_zero_zero:

        CJNE R7, #SETENTA_CINCO, else_zero                  ;Se o contador auxiliar 2 é 75, significa que o contador auxiliar contou 7500 (1.5s)

        MOV R5, #ZERO                                       ;Caso hajam sido ultrapassados os 1.5s , o numero de palmas é reseteado

        MOV R6, #ZERO                                       ;Caso hajam sido ultrapassados os 1.5s , o contador auxiliar é reseteado

        MOV R7, #ZERO                                       ;Caso hajam sido ultrapassados os 1.5s , o contador auxiliar 2 é reseteado

        RET

else_zero:

        CJNE R5, #DOIS, Fim2                                ;Testamos se foram registadas 2 palmas

        CJNE R2, #ZERO, else_um                             ;Se registamos 2 palmas e a intensidade for 0 significa que o candeeiro esta desligado, logo vamos liga-lo, caso contrario, desligamos

        SETB candeeiro
        ;Candeeiro ON

        MOV R2, #CEM
        ;Intensidade Luminosa 100%

        JMP Reset

else_um:
        ;Desligar

        CLR candeeiro
        ;Candeeiro OFF

        MOV R2, #ZERO
        ;Intensidade Luminosa 0%

Reset:

        MOV R5, #ZERO                                       ;Numero de palmas reseteado

        MOV R6, #ZERO
        ;Contador auxiliar reseteado

        MOV R7, #ZERO
        ;Contador auxiliar 2 reseteado

        RET

Fim2:

        RET

END
```