

PROJETO 3

Candeeiro Inteligente

**Docentes:**

• Dionísio Barros

• Pedro Camacho

• Nuno Ferreira

• Sofia Isabel Silva Inácio

**Alunos:**

• Joel Camacho, nº 2083516

•Wilson Agrela, nº 2083316

5 De junho de 2019

INDICE

Conteúdo

[Objetivos 3](#_Toc10669303)

[Resultados 3](#_Toc10669304)

[Conclusão 6](#_Toc10669305)

[Anexo A – Fluxogramas 7](#_Toc10669306)

[Anexo B 10](#_Toc10669307)

# Objetivos

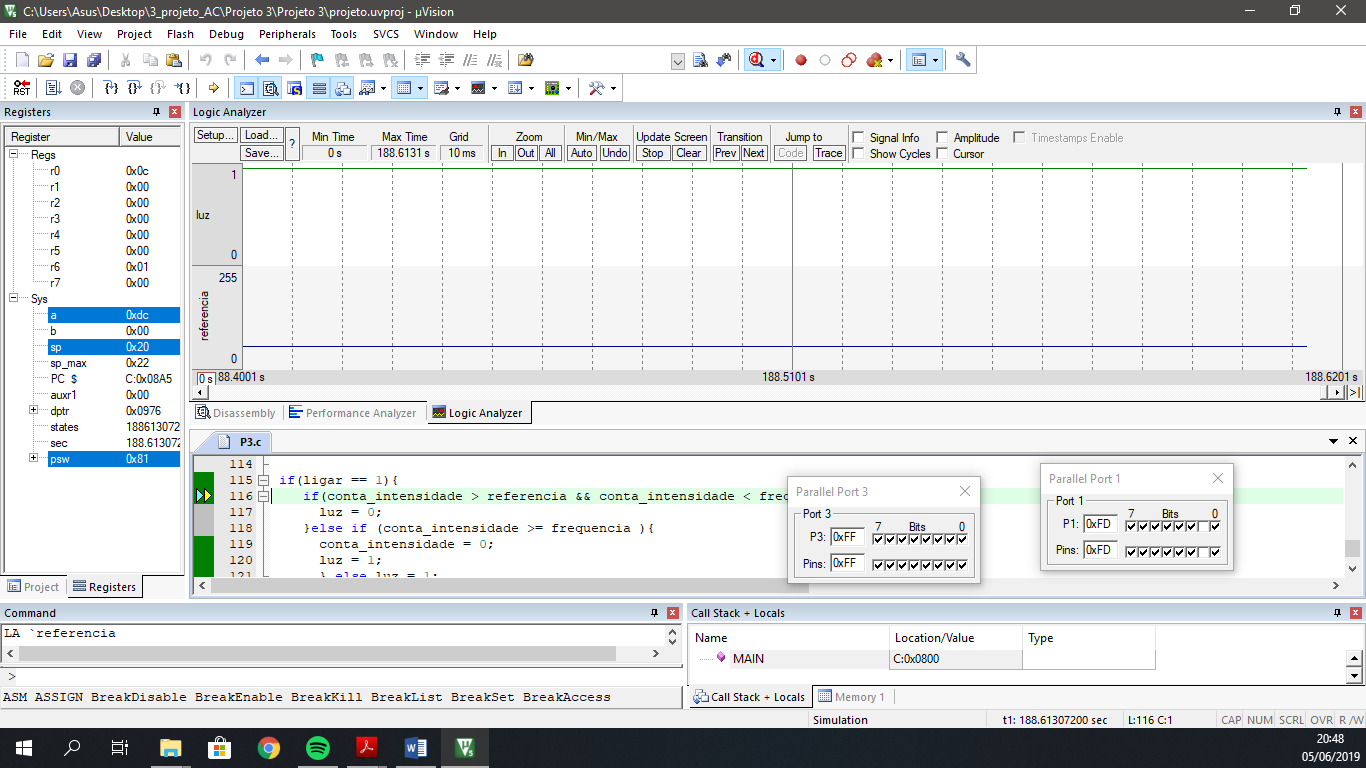
O nosso objetivo neste terceiro projeto é desenvolver, em linguagem c e em assembly, um programa que efetua o controlo da luz de um candeeiro “inteligente”, usando o microcontrolador 8051, um programa que ligue e desligue a luz, que consiga alterar a intensidade da luz e que reconhecendo o som de palmas ligue e desligue a luz.

# Resultados

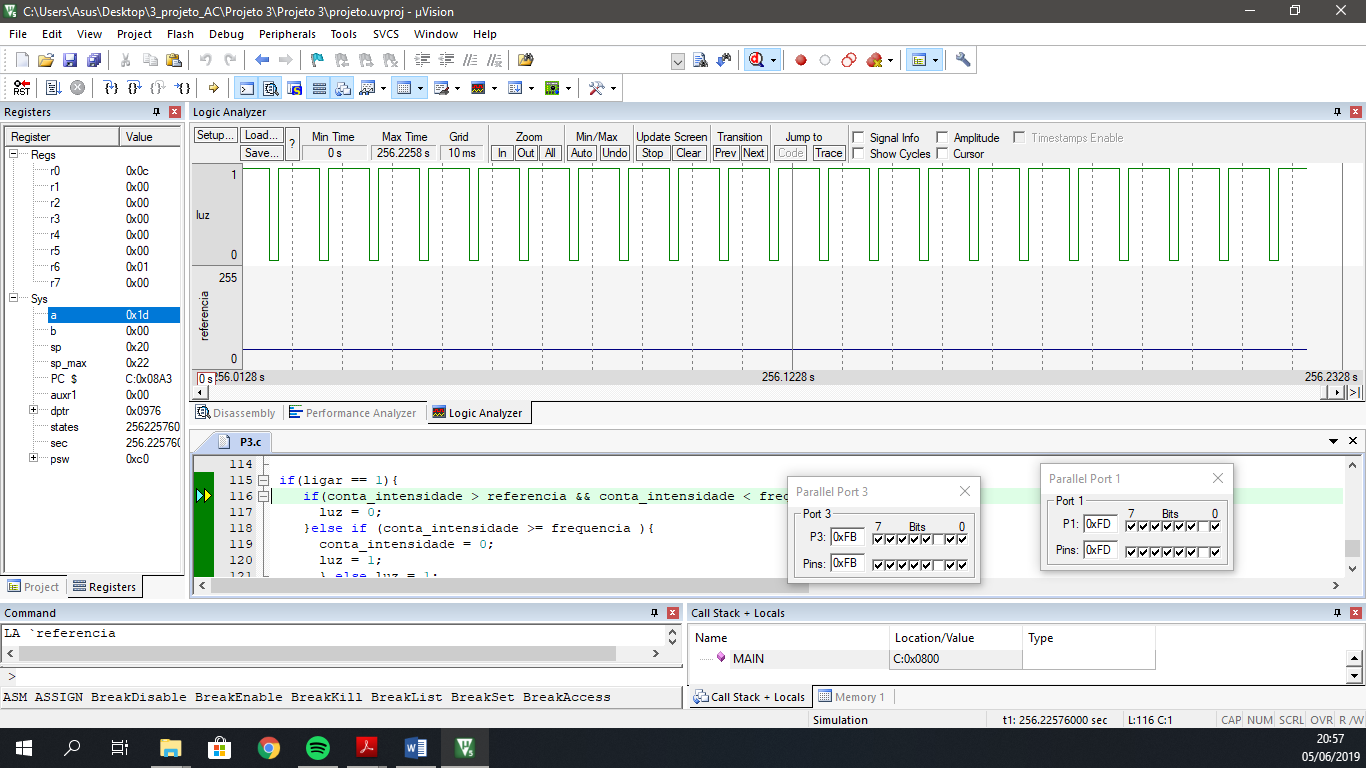
Neste projeto iremos ter 4 pinos:

* P1.0, que será um pino de saída, se o valor logico presente na saída for ‘1’, ira ligar a luz, caso seja ‘0’, a luz ira ser deligada;
* P1.1, este pino tem como objetivo ligar e desligar a luz do candeeiro e estará ligado a um botão S1;
* P3.2, neste pino estará ligado um botão S2, que ira regular a intensidade da luz do candeeiro;
* P3.3, este pino de entrada, estará ligado um microfone, que ao detetar palmas, ao detetar as palmas dadas, tem como objetivo ligar e desligar a luz do candeeiro.

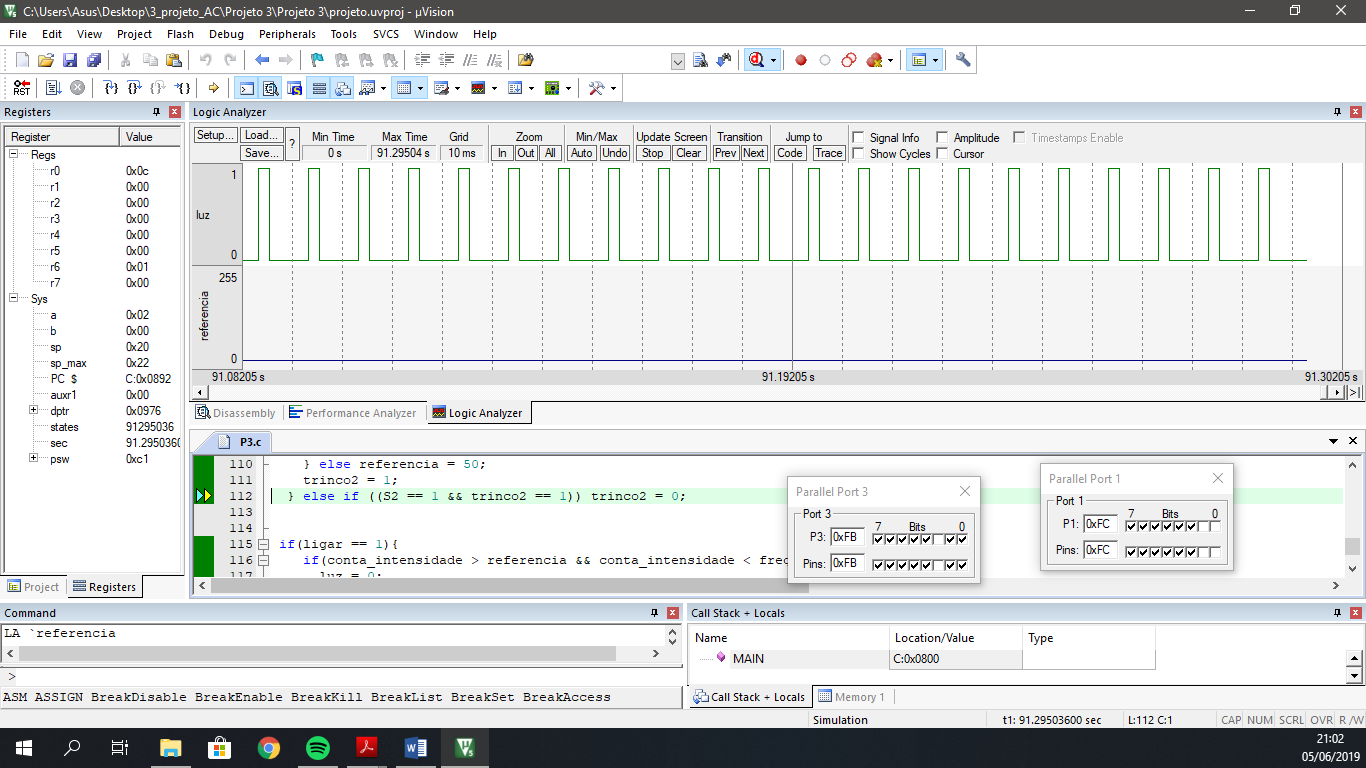
Ao carregar no botão S1, que está ligado ao pino P1.1, liga a luz se estiver desligada, e desliga a luz se esta tiver ligada. Para verificar o funcionamento simulamos no programa “keil uVision”, na figura 1, e verificou-se que funciona como pretendido.

*Figura 1*

No desenvolvimento do projeto, usou-se o timer 1 e a interrupção externa 2, esta interrupção vai estar ligada ao pino P3.2 e serve para variar a intensidade luminosa do candeeiro, nas figuras seguintes verifica-se que funciona como pretendido.



*Figura 2*



*Figura 3*

# Conclusão

Através deste projeto percebeu-se mais detalhadamente como fazer uso das interrupções e as suas vantagens, o facto de conseguirmos controlar a intensidade de uma luz. Em assembly, não se conseguiu atingir 100% do objetivo, em c o objetivo foi totalmente cumprido.

# Anexo A – Fluxogramas

# 

# 

# Anexo B

Linguagem c

#include <reg51.h>

#define fimTempo 100 //timer = 0.2ms -> fimTempo=100\*0.2ms=20ms

#define frequencia 50 //timer = 0.2ms -> frequencia=50\*0.2ms=10ms

#define atraso 25 //50\*20ms = 1s

bit botao = 0; //pino do botão

bit ligar = 0;

sbit luz = P1^0; //pino de controlo do luz motor

sbit S1 = P1^1;

sbit S2 = P3^2;

bit trinco = 0;

bit trinco2 = 0;

//sbit MICRO = P3^3;

bit palma = 0;

unsigned char palmas = 0;

unsigned char conta2 = 0;

unsigned char conta3 = 0;

unsigned char conta = 0; //contador que incrementa a cada 200us

unsigned char conta\_intensidade = 0;

unsigned char referencia = 50; //o luz motor começa nos 0º

//declaração de funções

void Init(void);

void Init(void)

{

//Configuracao Registo IE

EA = 1; //ativa interrupcoes globais

ET0 = 1; // ativa interrupcao timer 0

EX1 = 1; // ativa interrupcao externa 0

//Configuracao Registo TMOD

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

TMOD |= 0x02; //modo 2 do timer 0

//Configuracao Timer 0

TH0 = 0x38; //Timer 0 - 200us

TL0 = 0x38;

//Configuracao Registo TCON

TR0 = 1; //comeca o timer 0

IT1 = 1; //interrupcao externa activa a falling edge

}

//interrupcao externa

void External0\_ISR(void) interrupt 2

{

palma = 1;

}

//interrupcao tempo

void Timer0\_ISR(void) interrupt 1

{

conta++; //incrementa a cada contagem de 200us

conta3++;

if(ligar == 1) conta\_intensidade++;

}

void main(void) {

//inicializações

Init();

luz = 0;

while(1) //loop infinito

{

if(S1 == 0 && trinco == 0){ // btn carrega a 0

luz = ~luz;

trinco = 1;

if (luz == 1) {

conta\_intensidade = 0;

ligar = 1;

} else ligar = 0;

}else{

if(S1 == 1 && trinco == 1){

trinco = 0;

}}

if(conta == fimTempo){

conta = 0;

conta2++;

}

if(palma == 1) {

if((conta3 >= 25 && conta2 <= atraso) || palmas == 0){

palmas++;

conta3 = 0;

}else{

conta2 = 0;

conta3 = 0;

palmas = 0;

}

palma = 0;

}

if(palmas == 2){

if(luz == 0){

luz = 1;

ligar = 1;

}

else {

luz = 0;

ligar = 0;

}

palmas = 0;

}

if(S2 == 0 && trinco2 == 0) {

if(referencia > 10){

referencia = referencia - 10;

} else referencia = 50;

trinco2 = 1;

} else if ((S2 == 1 && trinco2 == 1)) trinco2 = 0;

if(ligar == 1){

if(conta\_intensidade > referencia && conta\_intensidade < frequencia){

luz = 0;

}else if (conta\_intensidade >= frequencia ){

conta\_intensidade = 0;

luz = 1;

} else luz = 1;

}

}

}

Assembly

TempoL EQU 0xEF ; Valor do byte menos significativo para o temporizador

TempoH EQU 0xD8 ; Valor do byte mais significativo para o temporizador

fimTempo EQU 0x64 ;timer = 0.2ms -> fimTempoEQU100\*0.2msEQU20ms

frequencia EQU 0x32 ;timer = 0.2ms -> frequenciaEQU50\*0.2msEQU10ms

atraso EQU 0x19 ;50\*20ms = 1s

botao EQU 0; ;pino do botão

ligar EQU 0;

luz EQU P1.0; ;pino de controlo do luz motor

S1 EQU P1.1;

S2 EQU P3.2;

trinco EQU 0;

trinco2 EQU 0;

palma EQU 0;

palmas EQU 0;

conta2 EQU 0;

conta3 EQU 0;

conta EQU 0; ;contador que incrementa a cada 200us

conta\_intensidade EQU 0;

referencia EQU 0x32;

CSEG AT 0000h

JMP main

CSEG AT 0013h

LJMP External0\_ISR

CSEG AT 000Bh

LJMP Timer0\_ISR

CSEG AT 0050h

main:

LCALL INIT

CLR luz

\_WHILE:

JB S1,\_01

JB trinco,\_01

CPL luz

SETB trinco

JNB luz,\_02

CLR A

MOV conta\_intensidade,A

SETB ligar

SJMP \_03

\_02:

CLR ligar

SJMP \_03

\_01:

JNB S1,\_03

JNB trinco,\_03

CLR trinco

\_03:

MOV A,conta

XRL A,#0x64

JNZ \_04

MOV conta,A

INC conta2

\_04:

JNB palma,\_05

CLR C

MOV A,conta3

SUBB A,#0x19

MOV A,#P0

SUBB A,#P0

JC \_06

SETB C

MOV A,conta2

SUBB A,#0x19

MOV A,#0x80

SUBB A,#0x80

JC \_07

\_06:

MOV A,palmas

JNZ \_08

\_07:

INC palmas

CLR A

MOV conta3,A

SJMP \_09

\_08:

CLR A

MOV conta2,A

MOV conta3,A

MOV palmas,A

\_09:

CLR palma

\_05:

MOV A,palmas

XRL A,#0x02

JNZ \_10

JB luz,\_11

SETB luz

SETB ligar

SJMP \_12

\_11:

CLR luz

CLR ligar

\_12:

CLR A

MOV palmas,A

\_10:

JB S2,\_13

JB trinco2,\_13

SETB C

MOV A,referencia

SUBB A,#conta

MOV A,#0x80

SUBB A,#0x80

JC \_14

MOV A,#0xF6

ADD A,referencia

MOV referencia,A

SJMP \_15

\_14:

MOV referencia,#0x32

\_15:

SETB trinco2

SJMP \_16

\_13:

JNB S2,\_16

JNB trinco2,\_16

CLR trinco2

\_16:

JB ligar,\_17

LJMP \_WHILE

\_17:

MOV A,conta\_intensidade

SETB C

SUBB A,referencia

JC \_19

MOV A,conta\_intensidade

SUBB A,#0x32

MOV A,#0x80

SUBB A,#0x80

JNC \_19

CLR luz

LJMP \_WHILE

\_19:

CLR C

MOV A,conta\_intensidade

SUBB A,#0x32

MOV A,#0x80

SUBB A,#0x80

JC \_20

CLR A

MOV conta\_intensidade,A

SETB luz

LJMP \_WHILE

\_20:

SETB luz

LJMP \_WHILE

INIT:

MOV IP, #00000010b

;MOV IE, #10000011b ; activa as interrupções globais, timer 0 e externa 0

SETB EA

SETB ET0

SETB EX1

MOV TMOD, #00000001b

MOV TH0,#0x38

MOV TL0,#0x38

SETB TR0

SETB IT1

RET

Timer0\_ISR:

INC conta

INC conta3

JNB ligar,Timer0\_ISR\_FIM

INC conta\_intensidade

Timer0\_ISR\_FIM:

RETI

External0\_ISR:

SETB palma

RETI

END