

PROJETO 3 LIGHT TRACKER

Docentes:

- Dionísio Barros
- Pedro Camacho
- Nuno Ferreira
- Sofia Isabel Silva Inácio

Alunos:

- Juan Silva, nº 2082516
- Joel Camacho, nº 2083516

Índice

Introdução	3
Resultados	4
Bibliografia	6
Conclusão	7
Anexo A – Fluxogramas	8
Anexo B	
Assembly	11
Linguagem C	14

Introdução

Programou-se, em linguagem c e em assembly, o microcontrolador AT89S51 com um algoritmo que procura e deteta uma fonte luminosa. Ao ser detetada uma fonte luminosa o servo motor mantém a sua posição e ativa um LED, quando não for detetada uma fonte luminosa o servo motor volta a realizar um varrimento a sua procura.

Resultados

No desenvolvimento do *Light Tracker*, utilizou-se o timer 0 e a interrupção externa 0, esta interrupção externa, ligada ao pin P3.2, é "chamada" quando o sensor deteta luz e é colocada uma variável a "1" para indicar que detetou luz e manter o servo motor na mesma posição.

O varrimento do servo motor é feito incrementando sequencialmente a variável "referencia", que é inicializada com o valor "3", até atingir o valor "12", isto faz com que o servo motor realize o varrimento de, aproximadamente, 18º em 18º graus.

Para verificar o funcionamento, realizou-se a simulação de onda quadrada no programa "Keil uVision", figura 1, e verificou-se que funciona como o pretendido.

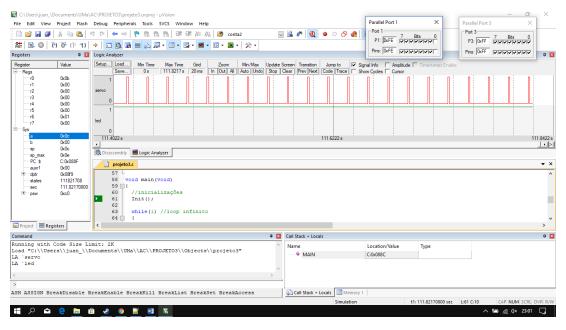


Figura 1 – funcionamento normal

Na figura 2, verifica-se que quando é detetada uma fonte luminosa a onda quadrada mantém a sua largura, o servo motor mantém a mesma posição, e o LED, PIN P1.1 passa a "0", juntamente com a onda correspondente, confirmando que o LED acende enquanto o servo motor está detetando luz.

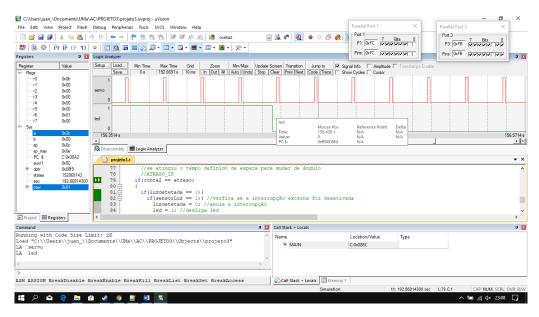


Figura 2- Após detetar uma fonte luminosa

Infelizmente, na parte em assembly, o objetivo não foi alcançado, o algoritmo aumenta a largura de onda, como em c, mas ao chegar aos 180º não diminui. Este algoritmo deteta a interrupção externa, a fonte luminosa, mantém a posição e acende o LED. Foi testada apenas na simulação, figura 3.

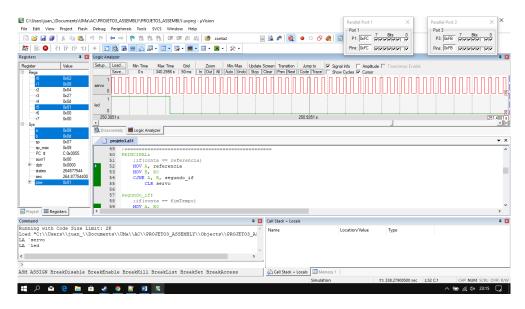


Figura 3- Simulação do assembly

Bibliografia

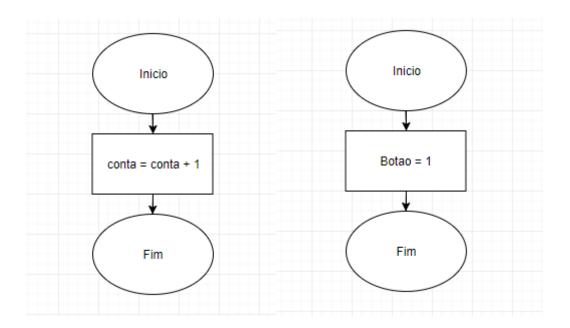
http://www.keil.com/forum/

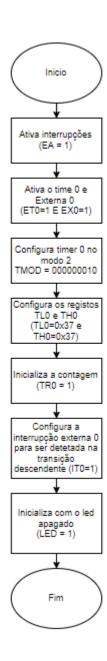
http://moodle.cee.uma.pt/course/view.php?id=54

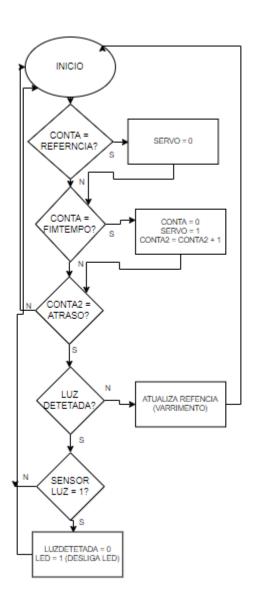
Conclusão

Através deste projeto percebeu-se mais detalhadamente como fazer uso das interrupções e as suas vantagens. Neste projeto, uma vantagem fundamental, é que não é necessário estar sempre a verificar se está, ou não, a detetar uma fonte luminosa. Em assembly, como referido anteriormente, não se conseguiu atingir 100% do objetivo, já em linguagem c, verificou-se que funcionou como o pretendido.

Anexo A – Fluxogramas







Anexo B

Assembly

```
fimTempo EQU 0x64 //timer = 0.2ms -> fimTempo=100*0.2ms=20ms
zero EQU 3 //3*0.2ms=0.6ms -> 0°
centoOitenta EQU 12 //12*0.2=2.4ms \rightarrow 180°
atraso EQU 0 \times 0 D //13 \times 20 ms = 0.260 s \sim 1/4 de segundo
sensorLuz EQU P3.2; //pino de controlo do sensor
servo EQU P1.0; //pino de controlo do servo motor
led EQU P1.1; //pino do led
conta EQU 0; //contador que incrementa a cada 200us
conta2 EQU 0; //tempo de espera entre mudança de angulos
referencia EQU 3; //o servo motor começa nos 0°
atualiza EQU 1;
luzdetetada EQU 0; //o servo motor começa nos 0°
TempoL EQU 0x37 ; Valor do byte menos significativo para o
temporizador
TempoH EQU 0x37; Valor do byte mais significativo para o
temporizador
;«« primeira instução após reset do microcontrolador
CSEG AT 0000H
    JMP INICIO
; «« se ocorrer interrupção externa
CSEG AT 0003H
    JMP INTERRUP EXTO
;«« tratamento da interrupção de temporização 0, para contar 20ms
CSEG AT 000BH
    JMP INTERRUP_TEMP0
; «« inicio do programa
CSEG AT 0050H ; coloca as instruções a partir do endereço 50h
    MOV SP, #7 ;endereço da stackpointer
    CALL INIT ; chama rotina de inicializações
```

```
; «« programa principal
PRINCIPAL:
     ;if(conta == referencia)
     MOV A, referencia
     MOV B, R0
     CJNE A, B, segundo if
          CLR servo
segundo if:
     ;if(conta == fimTempo)
     MOV A, RO
     MOV B, R2
     CJNE A, B, terceiro_if
          CLR A
          MOV RO, A
          SETB servo
          MOV A, R1
          INC A
          MOV R1, A
     terceiro if:
                          ;if(conta2 == atraso)
          MOV A, R1
          MOV B, R4
          CJNE A, B, PRINCIPAL
                MOV A, R5
                CJNE A, #01H, sub else
                     JNB sensorluz, fim sub
                           MOV R5, #00H
                           SETB led
                           JMP fim sub
          sub else:
                MOV A, referencia
                MOV B, centoOitenta
                CJNE A, B, compZero
                     MOV atualiza, #02H
          compZero:
                MOV A, zero
                MOV B, referencia
                CJNE A, B, atual
                     MOV atualiza, #01H
          atual:
                MOV A, atualiza
                CJNE A, #01H, sub_atual
                     MOV A, referencia
                     INC A
                     MOV referencia, A
                JMP fim_sub
          sub_atual:
                MOV A, referencia
                DEC A
                MOV referencia, A
fim sub:
     MOV A, R1
     CLR A
     MOV R1, A
```

JMP PRINCIPAL

END

```
; «« Rotina de inicializações
INIT:
    MOV IE, #10000011b; activa as interrupções globais, timer 0
e externa 0
    MOV TMOD, #00000010b; Activa temporizador no modo 2,
contador com registo de 8-bit (TLO) com auto reload
    ; configuração timer 200us
    MOV TLO, #TempoL ; Valor do byte menos significativo
    MOV THO, #TempoH; Valor do byte mais significativo
    ;Configuracao Registo TCON
    SETB TRO ; Activa o temporizador 0
    SETB ITO ; A interrupção externa vai ser detectada na
transição descendente (0->1)
    MOV RO, #conta ; RO - timer
    MOV R1, #conta2; R1 - conta2
    MOV R2, #fimTempo; R1 - conta2
    MOV R4, #atraso
    MOV R5, #luzdetetada
    SETB led ; led desligado
    RET
; «« tratamento interrupção externa
INTERRUP EXT0:
    MOV R5, #01H
    CLR led ; liga led
    RETI
    ; «« tratamento interrupção do timer0
INTERRUP TEMP0:
    INC R0 ;incrementa a cada contagem de 200us
    RETI
```

Linguagem C

```
#include <reg51.h>
#define fimTempo 100 //timer = 0.2ms -> fimTempo=100*0.2ms=20ms
#define zero 3 //3*0.2ms=0.6ms \rightarrow 0^{\circ}
#define centoOitenta 12 //12*0.2=2.4ms \rightarrow 180^{\circ}
\#define atraso 13 //100*20ms = 2s
sbit sensorLuz = P3^2; //pino de controlo do sensor
sbit servo = P1^0; //pino de controlo do servo motor
sbit led = P1^1; //pino do led
unsigned char conta = 0; //contador que incrementa a cada 200us
unsigned char conta2 = 0; //tempo de espera entre mudança de
angulos
unsigned char referencia = zero; //o servo motor começa nos 0°
unsigned char atualiza = 1;
unsigned char luzdetetada = 0; //o servo motor começa nos 0°
//declaração de funções
void Init(void);
void Init(void)
     //Configuração Registo IE
     EA = 1; //ativa interrupcoes globais
     ETO = 1; // ativa interrupcao timer 0
     EXO = 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 = 0x37; //Timer 0 - 200us
     TL0 = 0x37;
     //Configuracao Registo TCON
     TR0 = 1; //comeca o timer 0
     IT0 = 1; //interrupcao externa activa a falling edge
     led = 1; //led desligado
}
//interrupcao externa
void External0 ISR(void) interrupt 0
     luzdetetada = 1; //assinala que o sensor detetou luz
     led = 0; //liga led
//interrupcao tempo
```

```
void Timer0 ISR(void) interrupt 1
     conta++; //incrementa a cada contagem de 200us
void main(void)
     //inicializações
     Init();
     while(1) //loop infinito
                 //atingiu o valor de referencia (0.6ms, 1.4ms ou
2.4ms)
           if(conta == referencia) {
                 servo = 0; //coloca a saída a 0 até atingir os
20ms
           }
                 //atingiu os 20ms
           if(conta == fimTempo) {
                      conta = 0; //reinicia a contagem
                       servo = 1; //impulso positivo até atingir o
valor de referencia
                       conta2++; //incrementa 20ms na contagem do
tempo de espera entre mudança de angulos
           }
                 //se atingiu o tempo definido de espera para mudar
de ângulo
                 //ATRASO IF
           if(conta2 == atraso)
                 if(luzdetetada == 1) {
                       if(sensorLuz == 1) { //verfica se a
interrupção externa foi desativada
                            luzdetetada = 0; //anula a interrupção
                            led = 1; //desliga led
                 }
                 else {
                       if(referencia == centoOitenta)
                            atualiza = -1;
                       if(referencia == zero)
                            atualiza = 1;
                       referencia = referencia + atualiza;
           conta2 = 0; //reinicia a contagem do tempo
      }
}
```