



LICENCIATURA EM ENGENHARIA INFORMÁTICA

Sistemas Embebidos

TP1: Assembly

a44896 - João Martins

a48542- José Carvalho

A48538 - João Martins

1. Resumo:

Este trabalho consistiu na escrita em linguagem assembly de um programa que gerisse um

sistema de controlo de tráfego numa zona onde existe apenas uma faixa de rodagem disponível, que obriga a circulação de cada sentido à vez. A concepção do programa passou por estruturá-lo em quatro momentos: um primeiro momento em que um sentido circula, um segundo momento em que ambos os sentidos esperam, um terceiro momento em que o sentido oposto circula e por fim um quarto momento em que, novamente, ambos os sentidos esperam. Ao fim do quarto momento, o ciclo recomeça no primeiro momento e assim sucessivamente. Esta foi a forma que entendeu-se ser a melhor para cumprir com os objetivos propostos.

2. Contexto:

Este projeto teve início na formulação de um problema concreto que fosse plausível para aplicar conhecimentos em relação a: I/O, stack, external interrupts e timers. Depois de escolher um de entre os temas que constam na ficha de trabalho, seguiu-se um exercício de imaginação de um contexto suficientemente rico em eventos que justifiquem o uso de interrupções e temporizadores.

Imaginou-se uma estrada nacional com 2 vias de circulação, uma num sentido, outra no sentido oposto, onde circulam veículos ligeiros e pesados. Por causa do mau estado do piso é necessário interromper uma das faixas, para efetuar a substituição do piso danificado. Este evento obriga que a circulação de veículos neste troço da estrada passe a ser realizada numa faixa apenas. Uma vez que a estrada é de dois sentidos, será necessário a instalação de um sistema de controlo do tráfego.

3. Objetivos:

Criar um sistema de controlo de tráfego que faça a gestão do trânsito de veículos nos dois sentidos usando apenas uma faixa, de acordo com os seguintes requisitos:

- i. a circulação na via livre é realizada alternadamente num sentido e depois no outro;
- ii. cada sentido terá um determinado período de tempo para utilizar a faixa livre. Após o fim do período de tempo disponível para o uso da faixa livre:
 - 1. o semáforo deve mudar de verde para vermelho para o sentido que está a circular;
 - 2. o semáforo deve permanecer vermelho para o sentido que está a aguardar até que todos os veículos que estão a circular façam a travessia completa da faixa livre.
 - 3. o semáforo deve mudar de vermelho para verde para o sentido que está parado após a saída do último veículo da faixa livre;
- iii. o período de tempo destinado a cada sentido pode-se prolongar se não houver nenhum veículo à espera no sentido contrário;

4. Solução:

4.1. Dispositivos do sistema de controlo de tráfego:

- 2 unidades de semáforo (com 2 sinais luminosos: 1 verde e 1 vermelho);

função no sistema: dar indicação aos condutores para avançar ou parar;

- 2 unidades sensores (instalado junto a cada semáforo);

função no sistema: controlar se há veículos a circular na faixa livre;

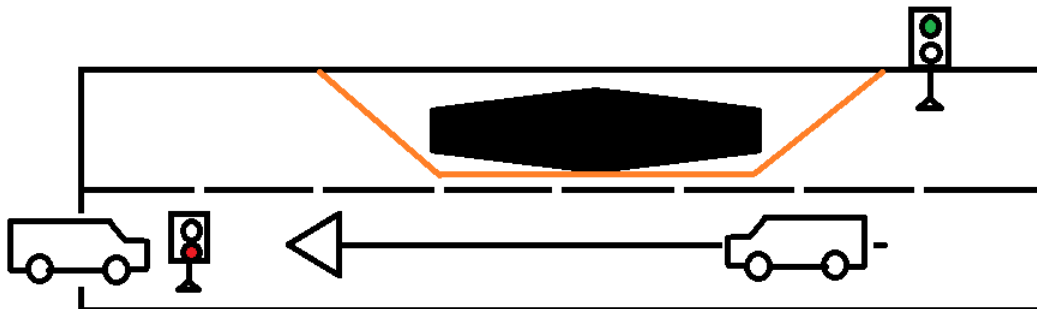
- 1 microprocessador;

função no sistema: controlar o sistema;

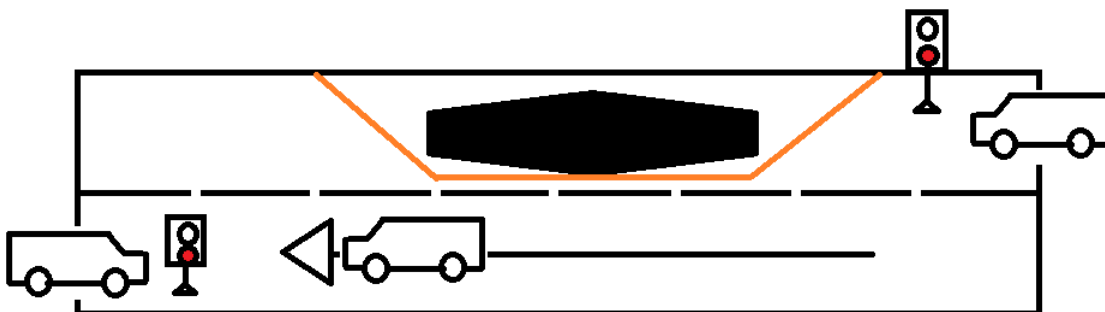
4.2. Método:

Perante o problema apresentado optou-se por uma estratégia que no nosso entender é a melhor solução, pois assegura a segurança de todos os intervenientes garantindo ao mesmo tempo a fluidez do trânsito automóvel. A estratégia passou por criar quatro momentos distintos, que acontecem uns a seguir aos outros e que, chegado ao fim, repete tudo de novo:

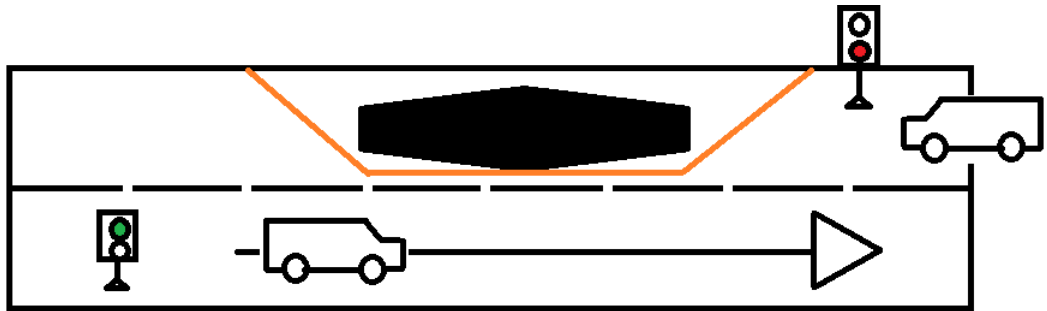
momento 1 -> momento 2 -> momento 3 -> momento 4 -> momento 1 -> momento 2 -> ...



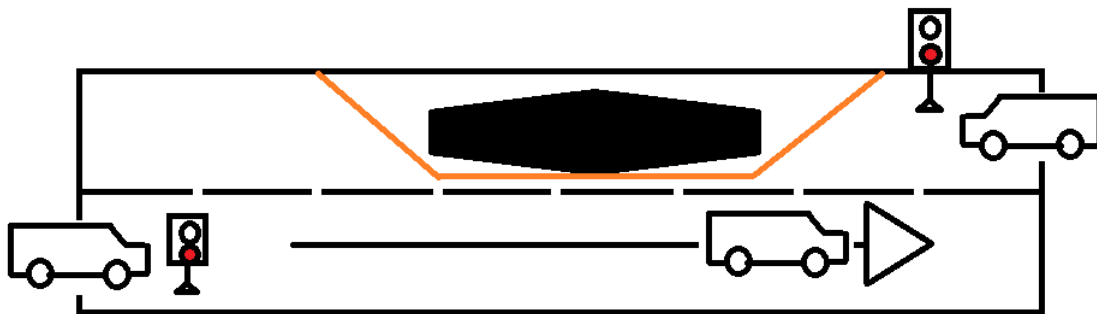
momento 1: os veículos que vêm da direita podem circular na faixa livre enquanto o outro sentido deve esperar.



momento 2: ambos os sentidos (da esquerda e da direita) devem esperar até que os veículos que iniciaram a marcha no momento anterior terminem de atravessar a faixa livre.



momento 3: os veículos que vêm da esquerda podem circular na faixa livre enquanto o outro sentido deve esperar.



momento 4: ambos os sentidos (da esquerda e da direita) devem esperar até que os veículos que iniciaram a marcha no momento anterior terminem de atravessar a faixa livre.

5. Resultado:

```
momento1:
ldi r19,0x01
; semaforo A -> sinal luminoso verde
; semaforo B -> sinal luminoso vermelho
; temporizador -> ON
sbi portA,1
sbi portB,6
ldi r16,0x01
out TCCR0,r16 ; (clk I/O)
rjmp loop0
```

momento1

registra em r19 o número do momento, neste caso 1
 acende o sinal luminoso verde do semaforo A
 acende o sinal luminoso vermelho do semaforo B
 ativa o temporizador
 salta para loop0

```
loop0:
rjmp loop0
```

loop0

aguarda uma interrupção do timer

interrupção acontece e salta para momento0

```
momento0:
cpi r19,0x01
breq momento2
cpi r19,0x02
breq momento3
cpi r19,0x03
breq momento4
cpi r19,0x04
breq momento1
```

momento0

verifica o valor em r19 e salta para o momento correspondente, neste caso 2

```
momento2:
ldi r19,0x02
; semaforo A -> sinal luminoso vermelho
; semaforo B -> sinal luminoso vermelho
; temporizador -> OFF
cbi portA,1
sbi portA,6
ldi r16,0x00
out TCCR0,r16 ; (no clk)
out TCNT0,r16
sei ; enable interrupt
rjmp loop1
```

momento2

registra em r19 o número do momento, neste caso 2

apaga o sinal luminoso verde do semaforo A

acende o sinal luminoso vermelho do semaforo A

desativa o temporizador

salta para loop1

```
loop1:
cpi r20,0x00
breq momento0
rjmp loop1
```

loop1

verifica o valor em r20, se for zero salta para o momento 0

momento0

verifica o valor em r19 e salta para o momento correspondente, neste caso 3

```

momento3:
ldi r19,0x03
; semaforo A -> sinal luminoso vermelho
; semaforo B -> sinal luminoso verde
; temporizador -> ON
sbi portB,1
cbi portB,6
ldi r16,0x01
out TCCR0,r16 ; (clk I/O)
rjmp loop0

```

momento3

registra em r19 o número do momento, neste caso 3

apaga o sinal luminoso vermelho do semaforo B

acende o sinal luminoso verde do semaforo B

ativa o temporizador

salta para loop0

loop0

aguarda uma interrupção do timer

interrupção acontece e salta para momento0

momento0

verifica o valor em r19 e salta para o momento correspondente, neste caso 4

```

momento4:
ldi r19,0x04
; semaforo A -> sinal luminoso vermelho
; semaforo B -> sinal luminoso vermelho
; temporizador -> OFF
cbi portB,1
sbi portB,6
ldi r16,0x00
out TCCR0,r16 ; (no clk)
out TCNT0,r16
sei ; enable interrupt
rjmp loop1

```

momento4

registra em r19 o número do momento, neste caso 2

apaga o sinal luminoso verde do semaforo A

acende o sinal luminoso vermelho do semaforo A

desativa o temporizador

salta para loop1

loop1

verifica o valor em r20, se for zero salta para o momento 0

momento0

verifica o valor em r19 e salta para o momento correspondente, neste caso 0

6. Trabalho de futuro:

O último objetivo foi onde surgiram dúvidas relativamente ao bom funcionamento na prática de um sistema que visasse cumprir o mesmo. Em termos de código, à partida, a implementação não parece difícil, mas quando se imagina o funcionamento na realidade, logo aparecem precalços que iriam afetar o bom desempenho do sistema. Por exemplo: o uso de sensor de deteção de movimento em campo aberto, ou seja, onde várias coisas podem interferir com o sensor para além do que é esperado é um aspeto que nos coloca dúvidas. Uma vez que o tempo era pouco, optamos por não concretizar este objetivo e por isso ele consta aqui.

No decorrer do trabalho apareceu uma ideia que melhoraria o sistema de controlo de tráfego porque introduz nele uma nova funcionalidade tornando-o por isso mais completo. Uma vez que estariam a trabalhar na substituição do piso danificado máquinas de grande porte, e que, determinadas manobras exigem mais espaço para além da largura de uma via, apareceu como uma eventual necessidade, interromper o trânsito nos dois sentidos, ou seja na estrada toda. Sem adiantar muito, esta funcionalidade teria que obedecer aos seguintes requisitos:

- i. por meio de um comando interromper o ciclo (momento1 -> momento2 -> momento3 -> momento4 -> momento1 -> ...) por tempo indeterminado
- ii. o ciclo recomeça no ponto onde parou apenas após um sinal do comando;
- iii. a ação de interrupção do comando só seria executada nos periodos de tempo em que ambos os semáforos estão a vermelho.

ANEXO 1

Código

```
.include "m8515def.inc"

.org 0x0000
rjmp inicio
.org 0x0001 ; interrupt0 (INT0)
rjmp interrupt0A
.org 0x0002 ; interrupt1 (INT1)
rjmp interrupt1B
.org 0x0007 ; timer0 (TC0) PBO
rjmp momento0
```

inicio:

```

ldi r16,low(RAMEND)
out SPL,r16
ldi r16,high(RAMEND)
out SPH,r16

; config. interrupt
ldi r16,0xC0 ; enable INT0 + INT1
out GICR,r16
ldi r16,0x0F ; rising edges
out MCUCR,r16

; config. timer
ldi r16,0x02
out TIMSK,r16 ; TCO overflow interrupt enable

sei ; enable interrupt

clr r20
clr r19

momento1:
ldi r19,0x01
; semaforo A -> sinal luminoso verde
; semaforo B -> sinal luminoso vermelho
; temporizador -> ON
sbi portA,1
sbi portB,6
ldi r16,0x01
out TCCR0,r16 ; (clk I/O)
rjmp loop0

momento2:
ldi r19,0x02
; semaforo A -> sinal luminoso vermelho
; semaforo B -> sinal luminoso vermelho
; temporizador -> OFF
cbi portA,1
sbi portA,6
ldi r16,0x00
out TCCR0,r16 ; (no clk)
out TCNT0,r16
sei ; enable interrupt

```


rjmp loop1

momento3:

ldi r19,0x03

; semaforo A -> sinal luminoso vermelho

; semaforo B -> sinal luminoso verde

; temporizador -> ON

sbi portB,1

cbi portB,6

ldi r16,0x01

out TCCR0,r16 ; (clk I/O)

rjmp loop0

momento4:

ldi r19,0x04

; semaforo A -> sinal luminoso vermelho

; semaforo B -> sinal luminoso vermelho

; temporizador -> OFF

cbi portB,1

sbi portB,6

ldi r16,0x00

out TCCR0,r16 ; (no clk)

out TCNT0,r16

sei ; enable interrupt

rjmp loop1

momento0:

cpi r19,0x01

breq momento2

cpi r19,0x02

breq momento3

cpi r19,0x03

breq momento4

cpi r19,0x04

breq momento1

loop0:

rjmp loop0

loop1:

cpi r20,0x00

breq momento0

```
rjmp loop1
```

```
interrupt0A:
```

```
sbic portA,6
```

```
dec r20
```

```
sbic portA,1
```

```
inc r20
```

```
reti
```

```
interrupt1B:
```

```
sbic portB,6
```

```
dec r20
```

```
sbic portB,1
```

```
inc r20
```

```
reti
```