

Chave Liga Desliga  
(Um toque Liga outro Desliga)

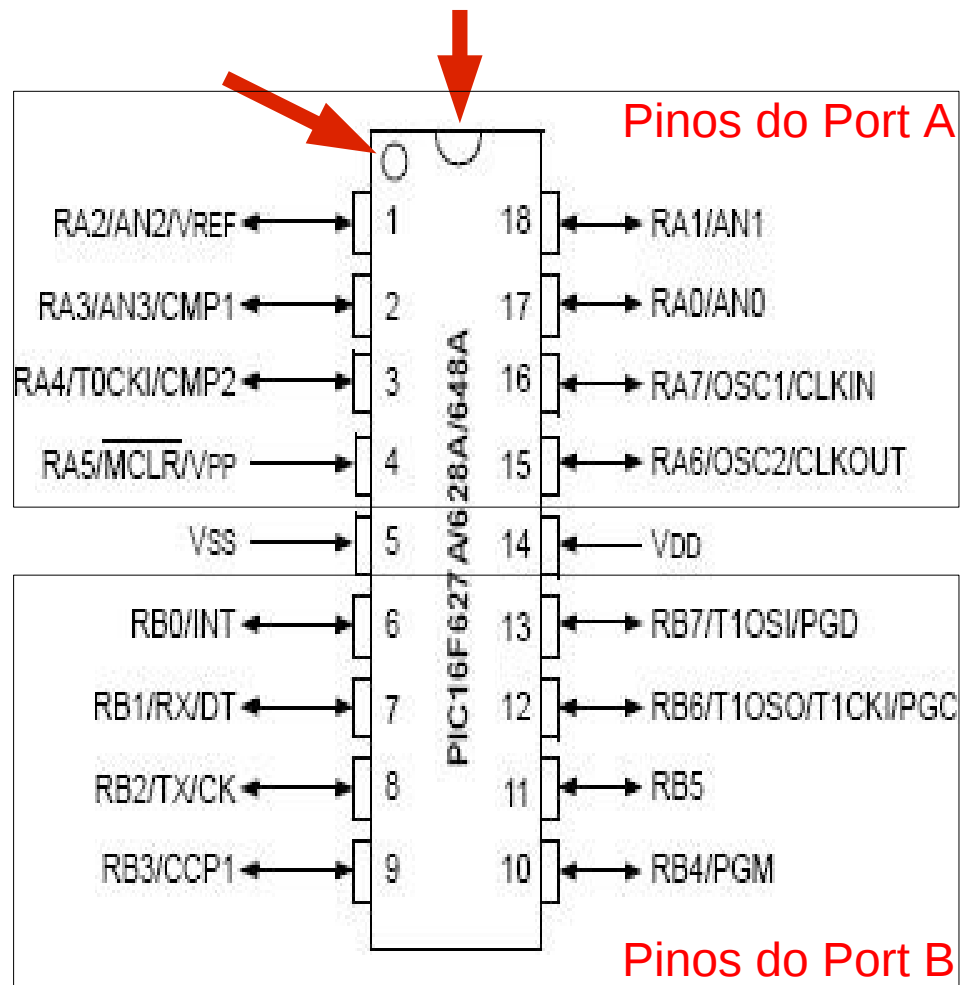
# Roteiro

- Materiais
- Pinos do PIC
- Fluxograma
- E/S com PIC
- Circuito no protoboard
- Melhoramentos
- Sugestão de Atividades

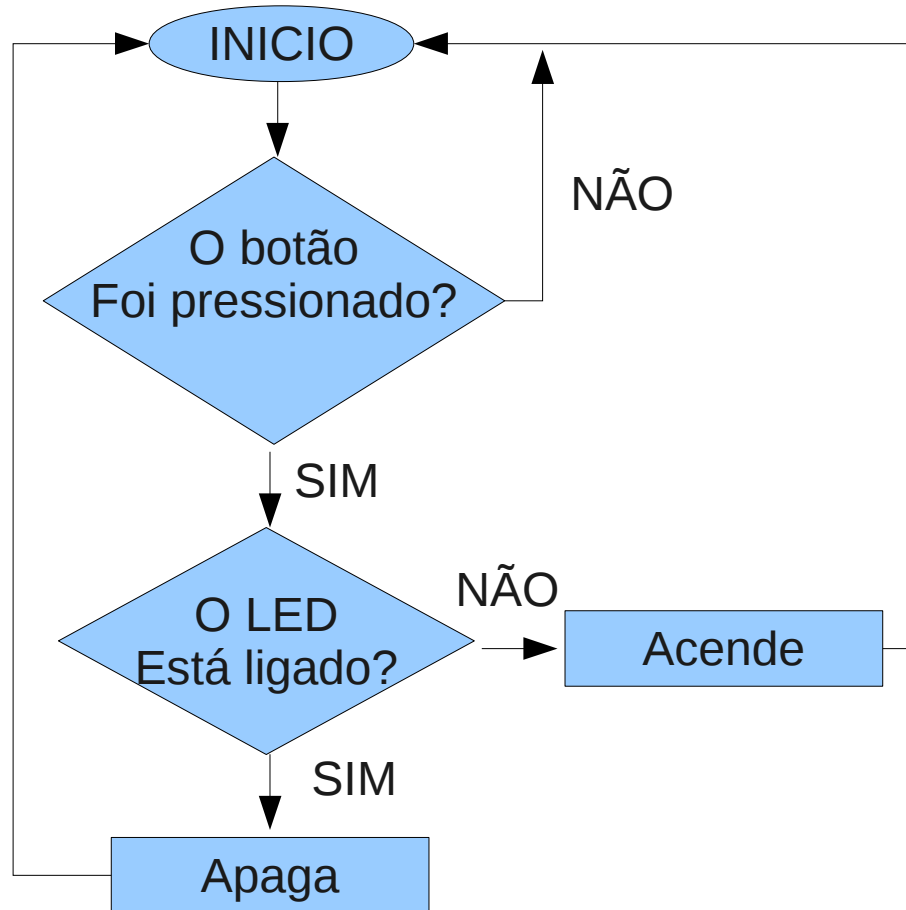
# Materiais

- Gravador
- PIC 16F628A
- 1 Led
- Resistores: 1 x 1K e 1 x 10k
- 2 Suportes para Pilhas AA
- 4 Pilhas AA
- Push button
- Protoboard e Fios

# Pinos do PIC



# Fluxograma

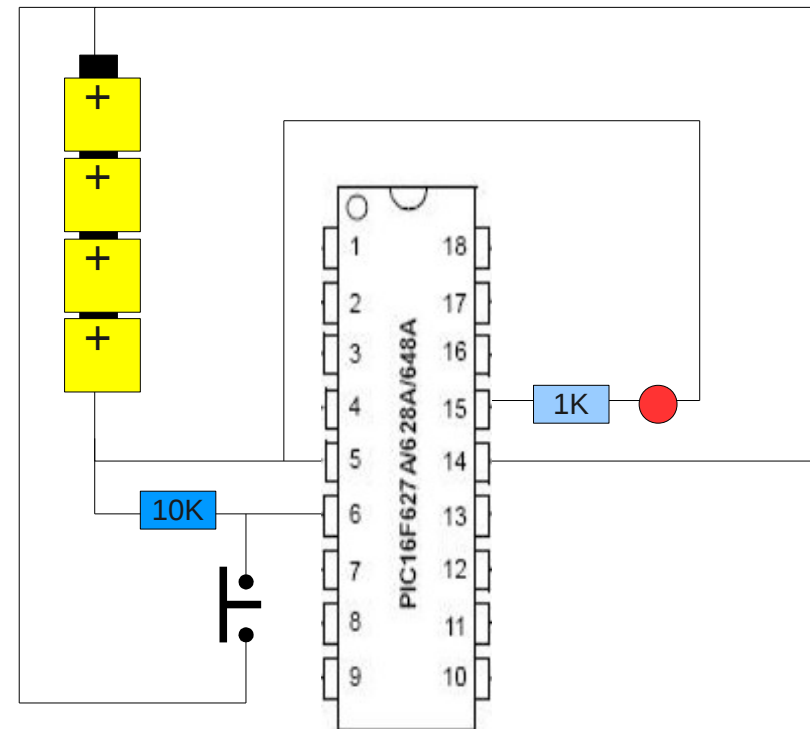


# Entrada e Saída

- Em geral os sistemas computacionais precisam comunicar-se com o mundo externo, no contexto de computadores essas comunicações são conhecidas como operações de E/S (Entrada/Saída).
- **Saída:** Toda vez que o sistema precisar “afetar” de alguma forma o mundo externo, essa operação será considerada saída. Ex: ligar um led, rotacionar um motor.
- **Entrada:** Toda vez que o sistema precisar responder a um estímulo vindo do mundo externo isso será considerado entrada. Ex: Um botão foi pressionado, o sistema acende um led.

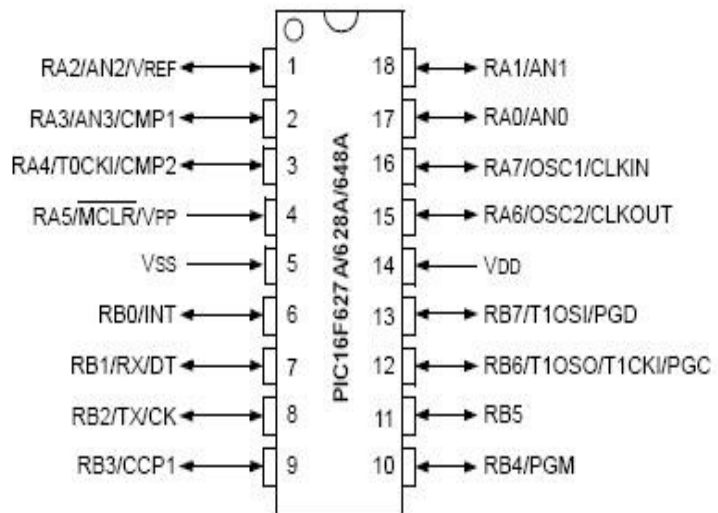
# Fazendo E/S com o PIC

- O controlador PIC tem 2 Portas para Entrada e Saída uma conhecida por **Port A** e a outra **Port B**. Usaremos neste exemplo, usaremos **port B, bit 0** (Pino 6 do CI) como entrada, onde será ligado um botão. E **port A bit 6** (Pino 15 do CI) como saída onde será ligado um led.



# Fazendo E/S com o PIC

- O Port A do PIC tem o endereço 05H e o Port B endereço 06H, Cada um destes endereços tem os seu bits “ligados” aos pinos externos do PIC, de acordo com a tabela a seguir:



BIT PORTA	Pino
0	17
1	18
2	1
3	2
4	3
5	4
6	15
7	16

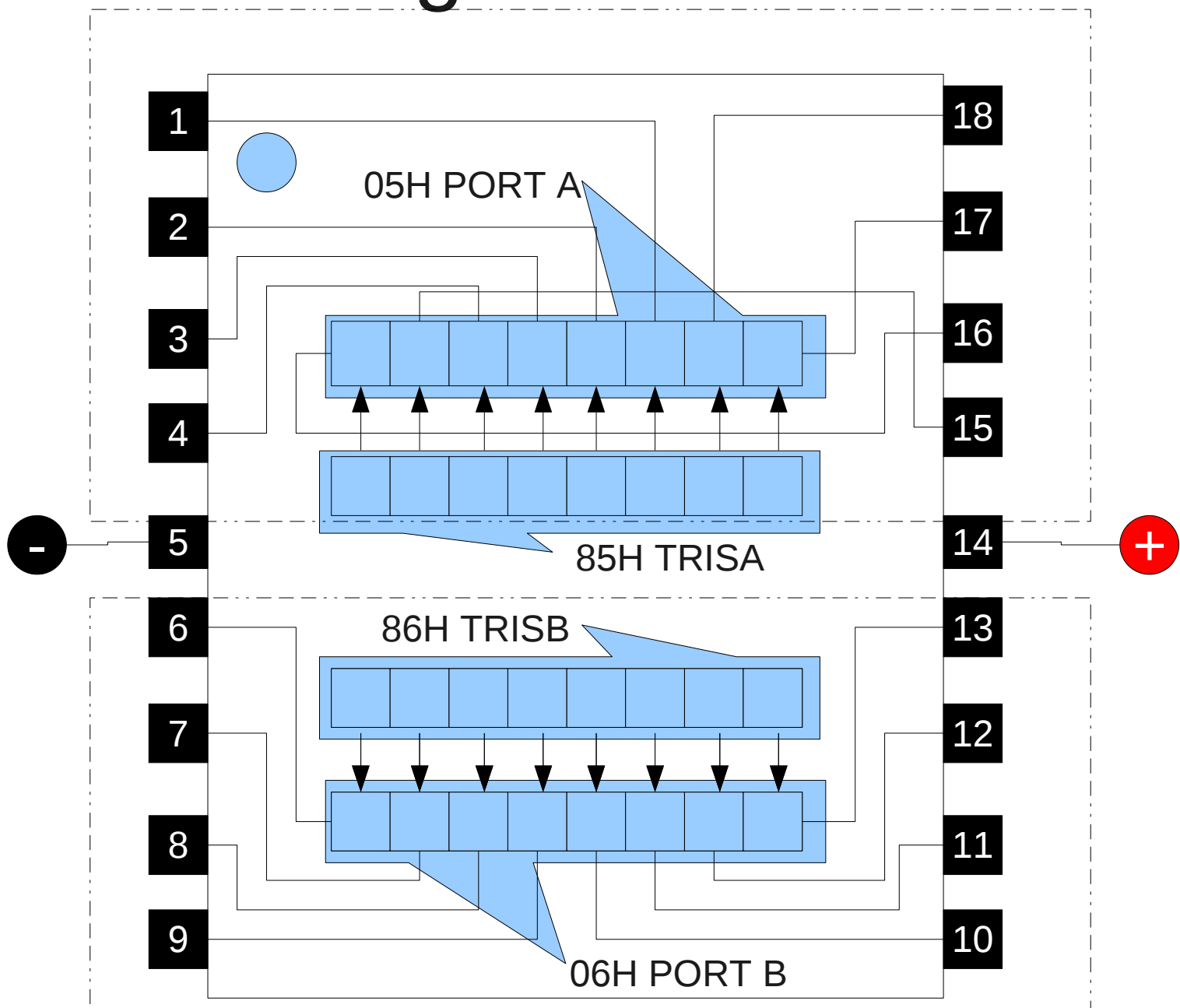
BIT PORTB	Pino
0	6
1	7
2	8
3	9
4	10
5	11
6	12
7	13



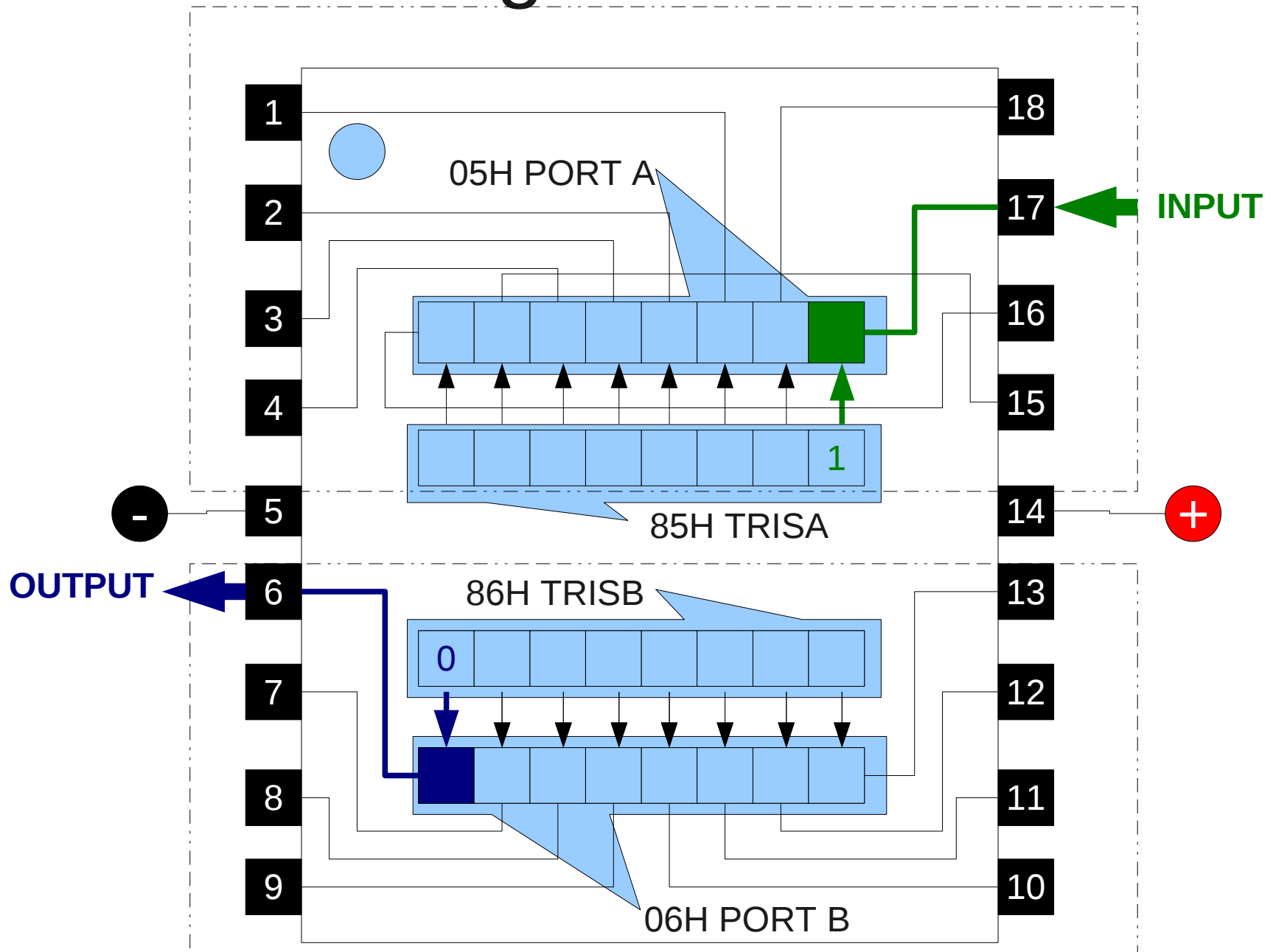
# Configurando E/S

- As configurações de E/S para cada Port são feitas usando os registradores TRISA e TRISB onde TRISA configura o PortA e TRISB o PortB, os endereços são 85H e 86H respectivamente.
- Para configurar algum pino de Port B como saída devemos desligar o bit correspondente em TRISB. Ou seja se queremos que o pino 6 de PortB seja saída (output) configuramos o Bit 6 de TRISB como 0. Para configurar como entrada configuramos o bit 6 como 1;
- **É fácil lembrar desta regra 0 para Output e 1 para Input**

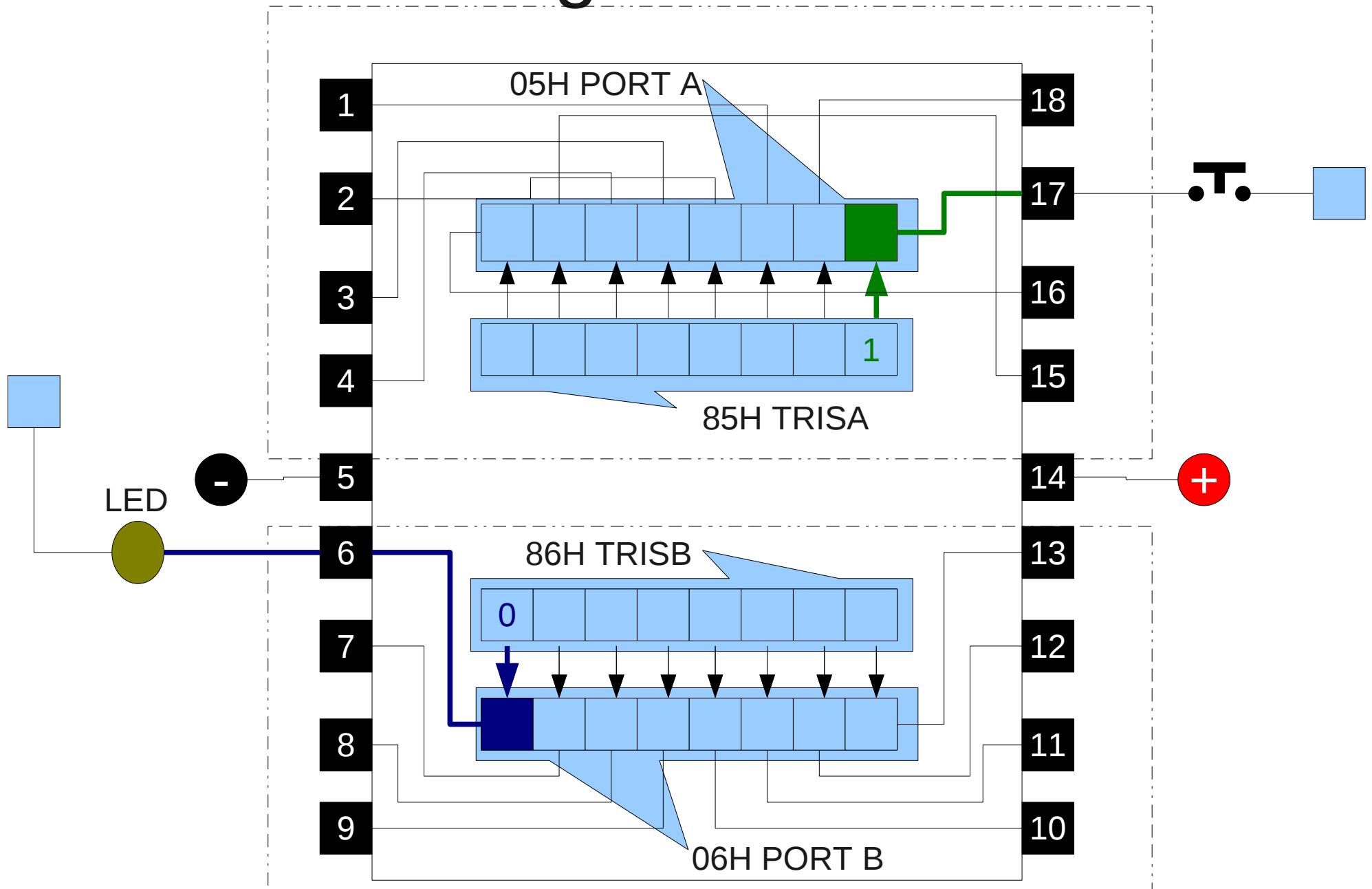
# Configurando E/S



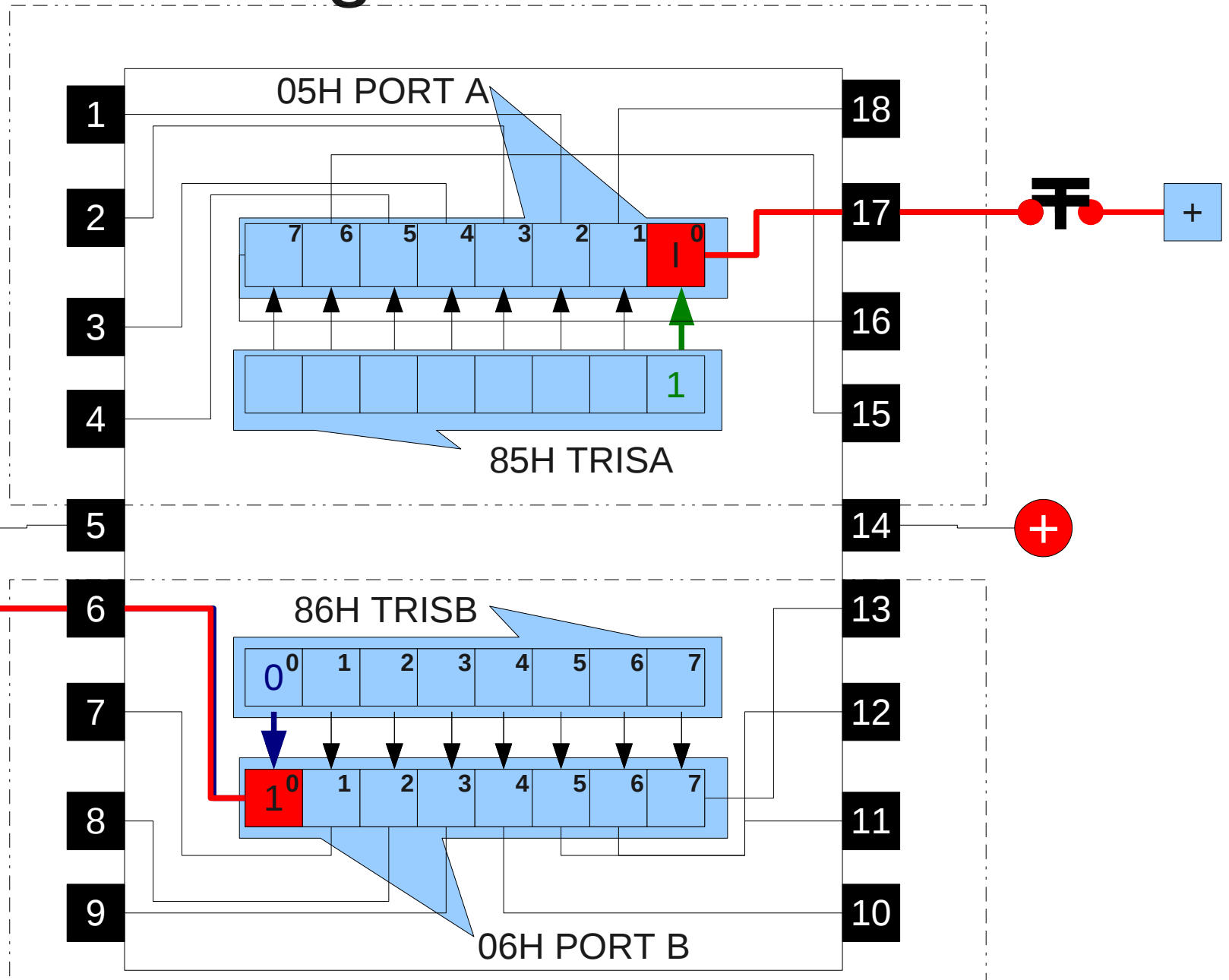
# Configurando E/S



# Configurando E/S

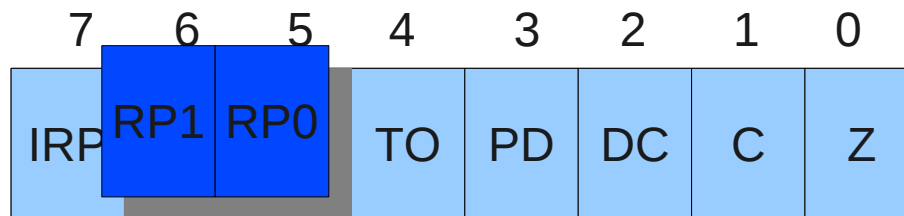
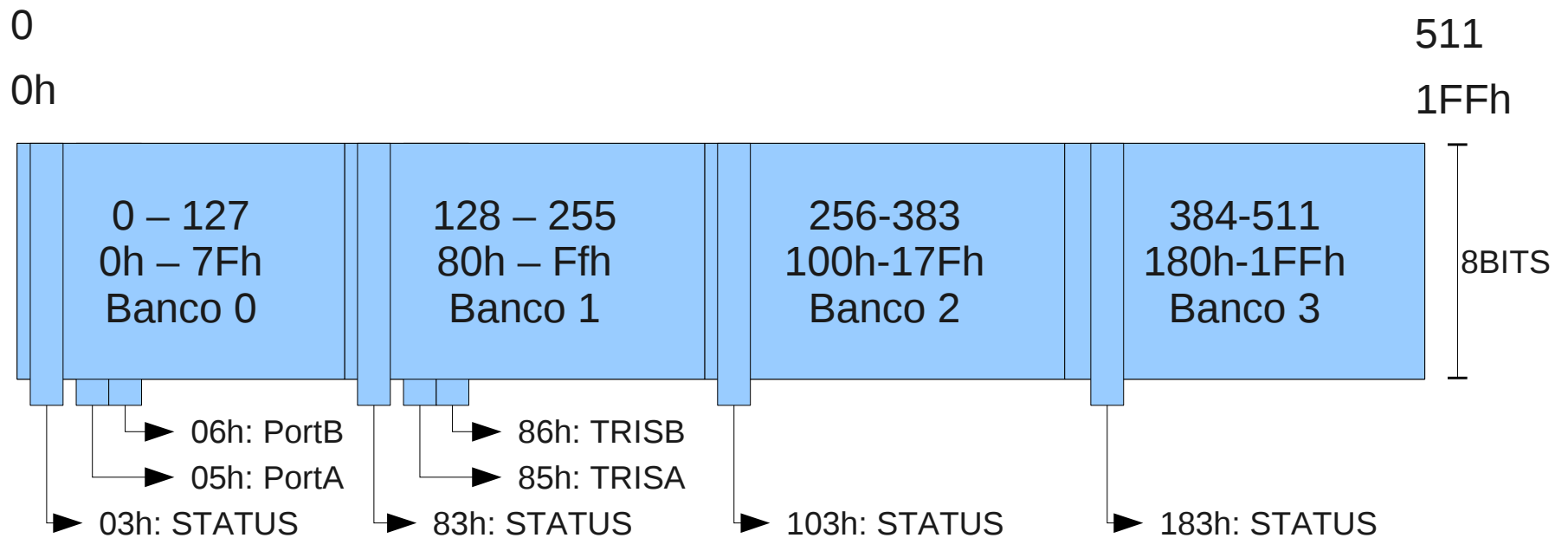


# Configurando E/S



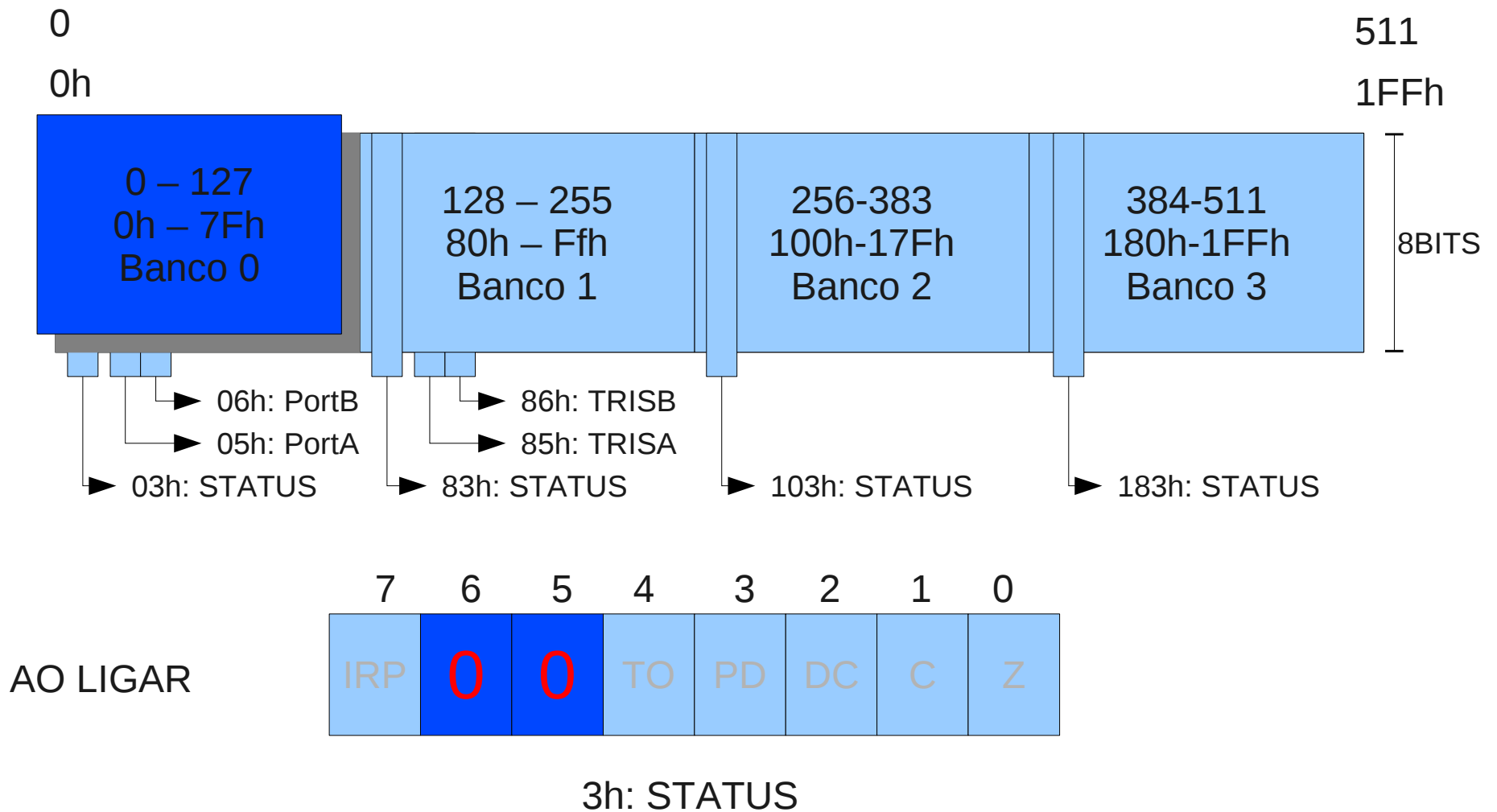
BSF 06h,0

# Memória de dados do PIC

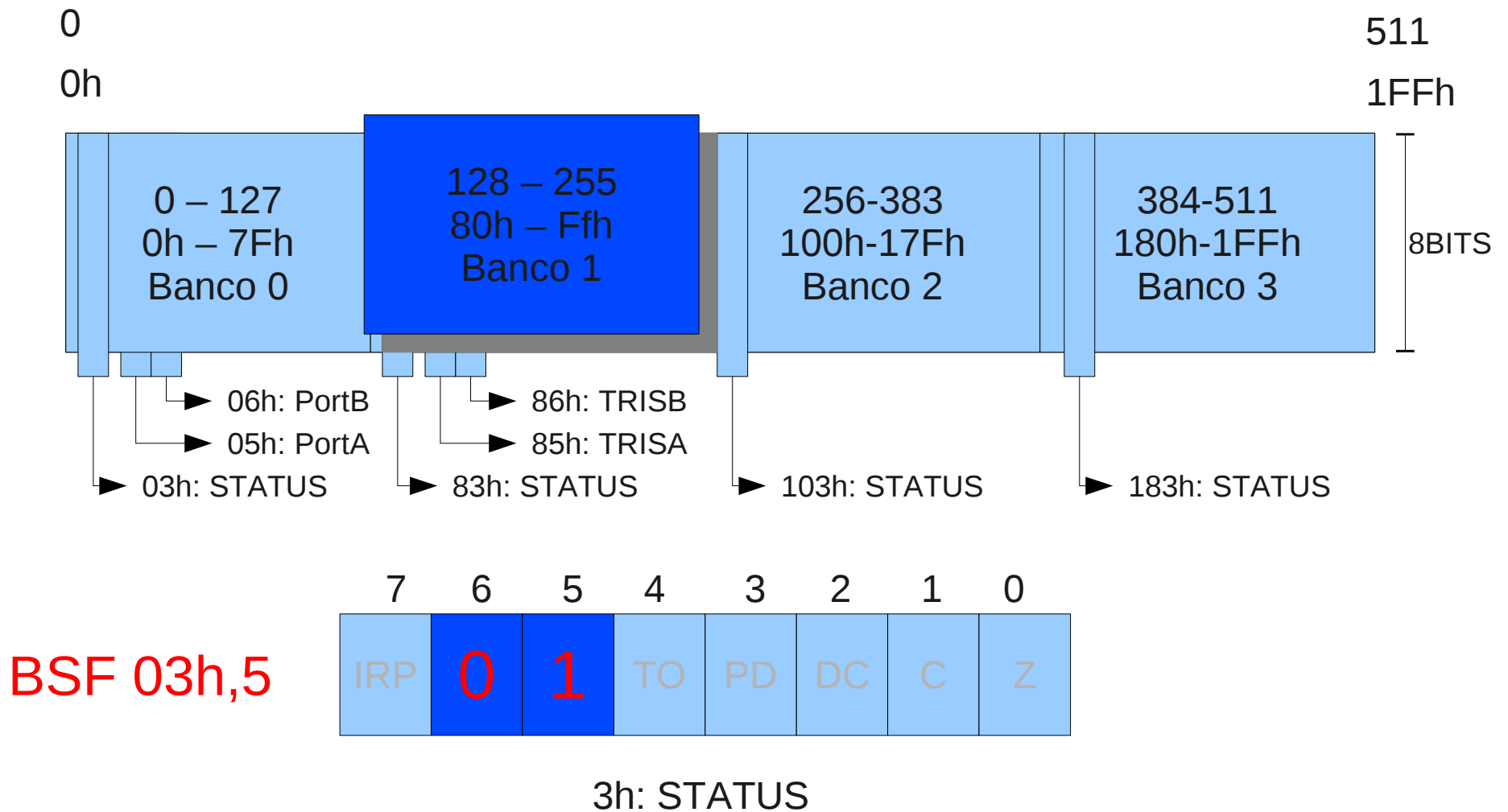


3h: STATUS

# Memória de dados do PIC

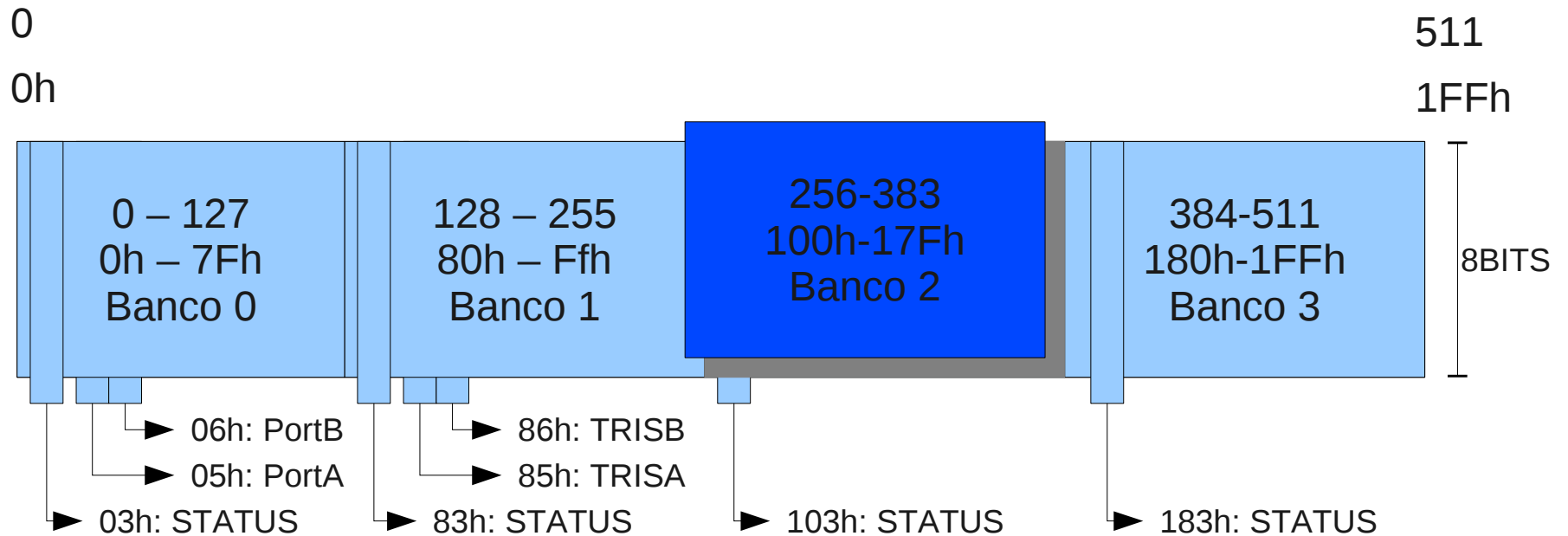


# Memória de dados do PIC





# Memória de dados do PIC

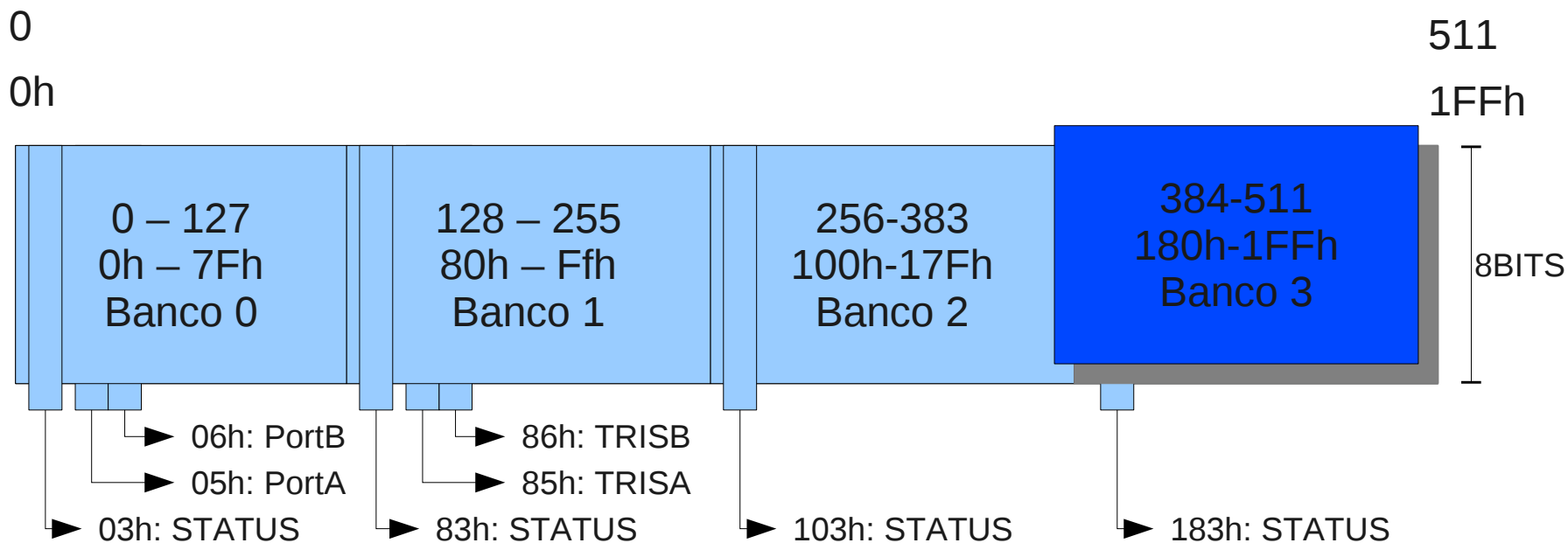


**BCF 03h,5**  
**BSF 03h,6**



3h: STATUS

# Memória de dados do PIC



BSF 03h,5  
BSF 03h,6



3h: STATUS

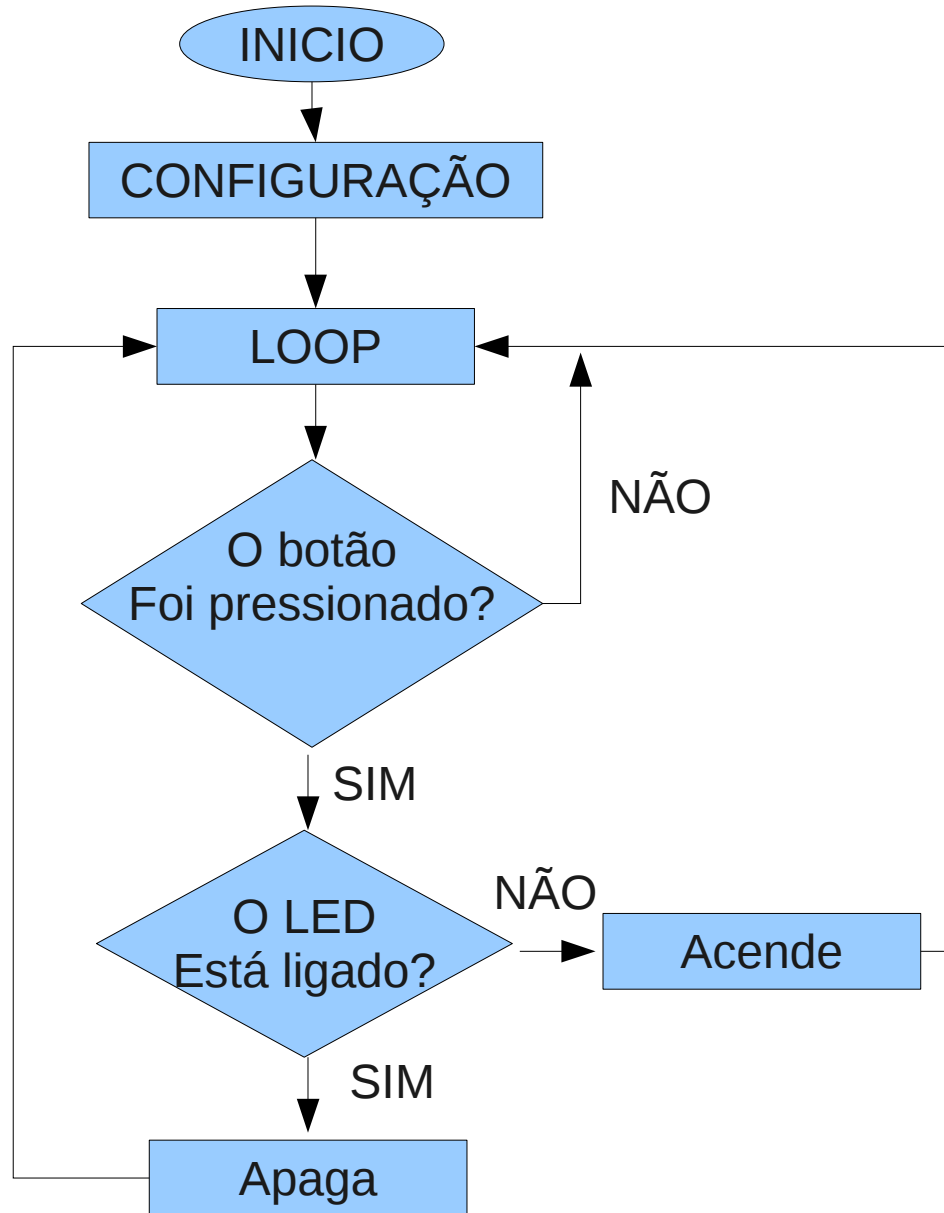
# Configuração

Código: Configuração

## CONFIGURACAO

BSF 03H,5 ; Selecciona banco 1 (seta RP0)  
BCF 85H,6 ; Reseta RA6 (RA6 como saída)  
BSF 86H,0 ; Seta RB0 (RB0 como entrada)  
BCF 03H,5 ; Selecciona banco 0 (reseta RP0)

# Fluxograma



# Teste se o Botão foi pressionado (Testando um Bit)

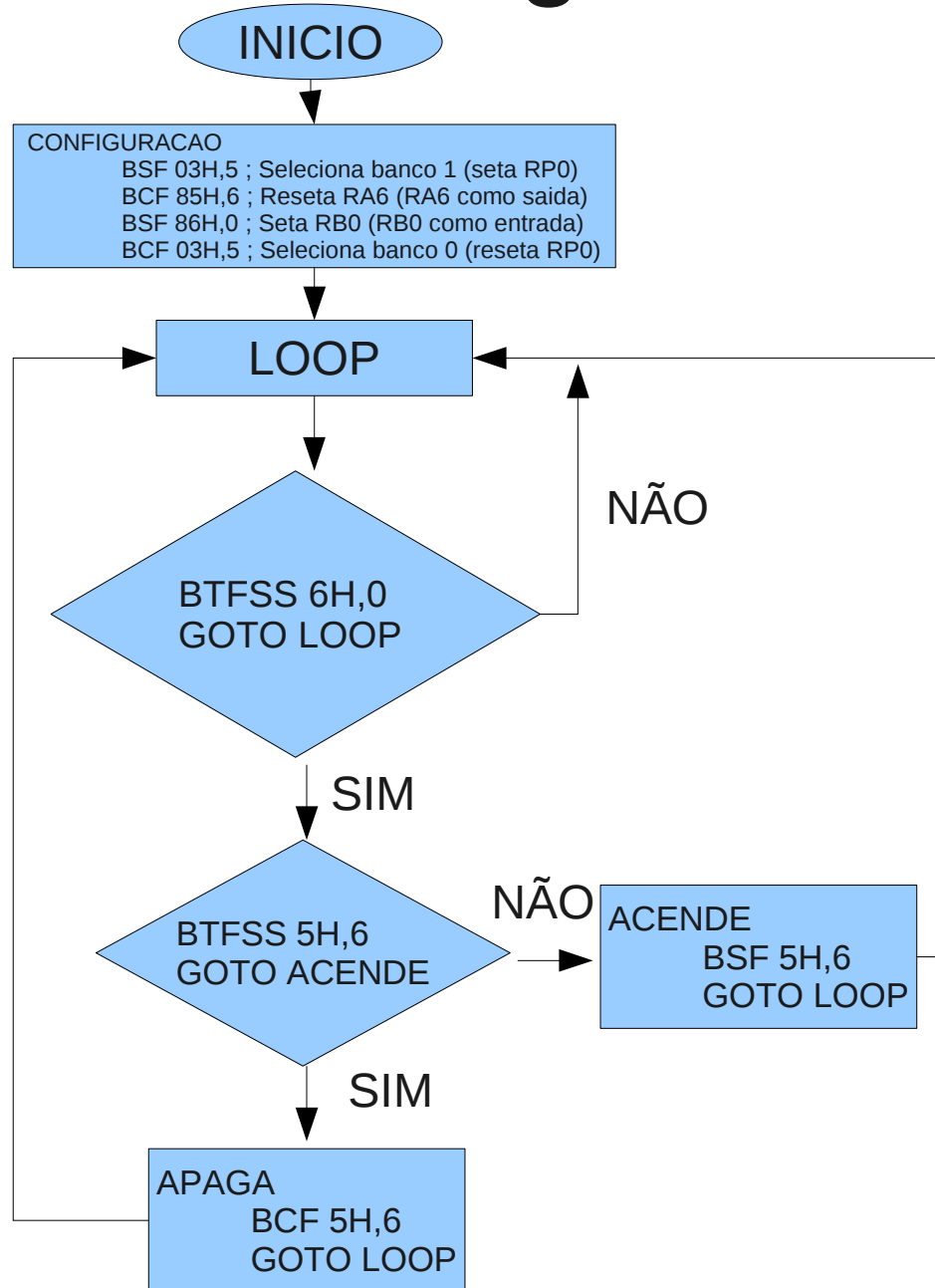
- Se o teste for **Verdadeiro** salta a próxima instrução.
  - Para Testar se um bit está ligado:
    - BTFSS Endereço, bit
  - Para testar se um bit está desligado:
    - BTFSZ Endereço, bit
- EX: (Testa se o **bit 6** de **Port A** está ligado )

```
BTFSS 05h,6  
GOTO DESLIGADO  
GOTO LIGADO
```

# Instruções BCF e BSF

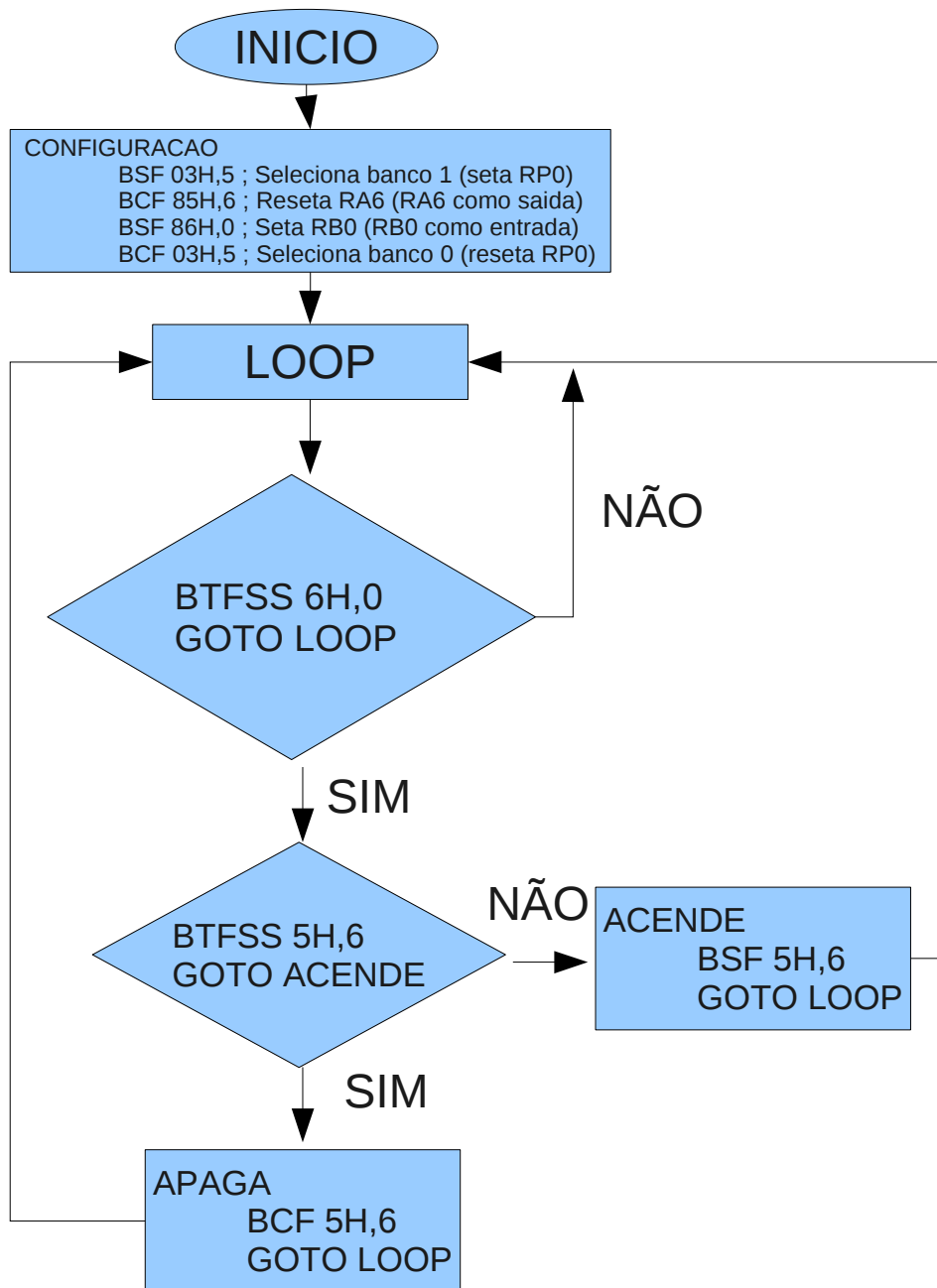
- As instruções BCF (Bit Clear File) e BSF (Bit Set File) são usadas para desligar e ligar bits específicos de um registrador.
- Ex
  - BCF 06H,0 ; desliga o bit 0 de 6H
  - BSF 06H,0 ; Liga o bit 0 de 6H
- A sintaxe destes comandos é
  - BCF Endereço, Bit
  - BSF Endereço, Bit

# Fluxograma



# Código (ex2-0.asm)

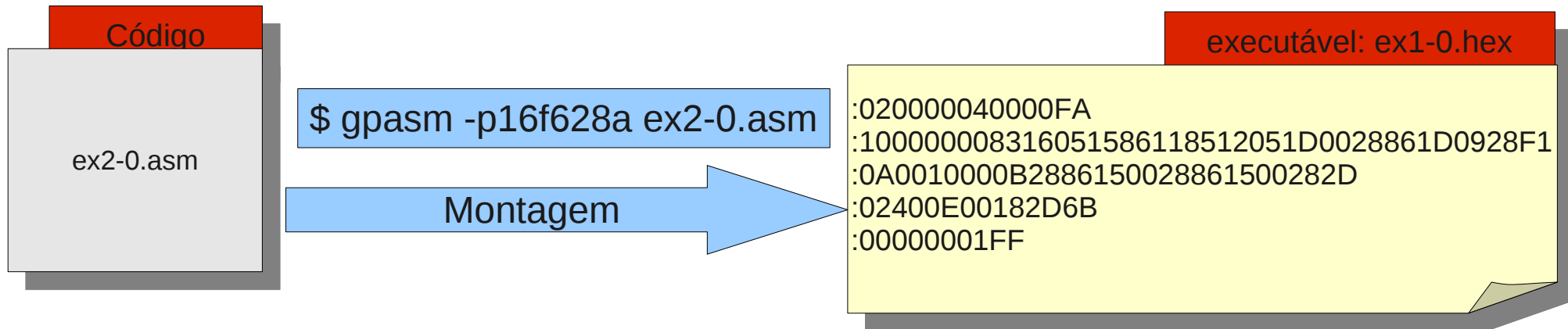
Código: ex2-0.asm



```
__CONFIG 0x3D18
ORG 0x00
INICIO
CONFIGURACAO
    BSF 03H,5 ; Selecciona banco 1 (seta RP0)
    BCF 85H,6 ; Reseta RA6 (RA6 como saída)
    BSF 86H,0 ; Seta RB0 (RB0 como entrada)
    BCF 03H,5 ; Selecciona banco 0 (reseta RP0)
    CLRF 05H
LOOP
    BTFSS 6H,0 ; Pula se botão Pressionado
    GOTO LOOP
    BTFSS 5H,6 ; Pula se Luz acessa
    GOTO ACENDE
    GOTO APAGA
ACENDE
    BSF 5H,6
    GOTO LOOP
APAGA
    BCF 5H,6
    GOTO LOOP
END
```



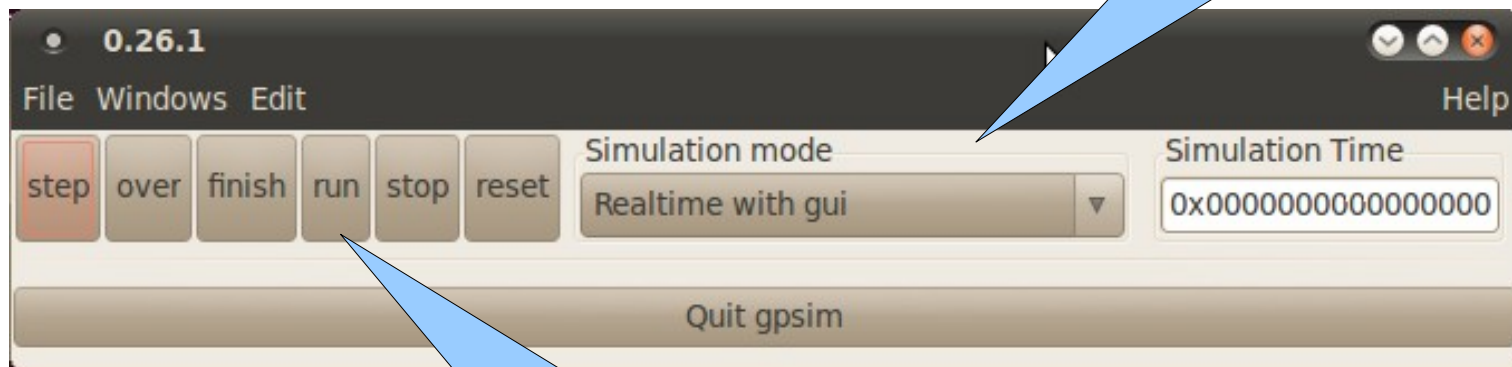
# Processo de Montagem



# Simulação

```
$ gpsim ex2-0.cod
```

Escolha Realtime with GUI  
(simulação em tempo real)



Inicie a simulação  
Verifique a Breadboard

Clique no pino correspondente a RB0 para que ele fique vermelho (1), Verifique o que o pino RA6 fica piscando. Clique novamente em RB0 e verifique que RA6 parou de piscar, mas pode tanto estar ligado como desligado. Na aplicação real, enquanto o botão estiver pressionado o led estará sendo aceso e apagado 1 milhão de vezes por segundo. Desta forma o Led ficará aceso ou apagado dependendo do tempo que o botão ficar pressionado.

Observe que não é isso que queremos.

Então como resolver esse problema? Resp: Proximo slide

**Resposta: Interrupção**

# Interrupções

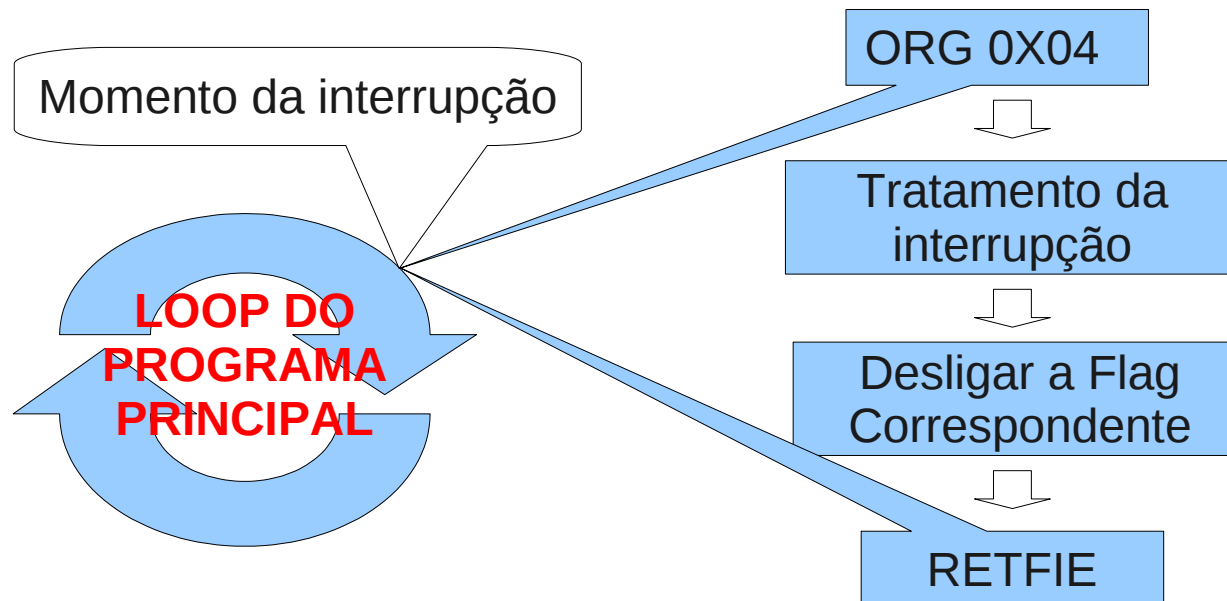
- A interrupção é um recurso importante nos sistemas computacionais, com ela é possível avisar a CPU a ocorrência de algum evento externo que precisa de tratamento imediato. A rotina que vai cuidar dessa interrupção é conhecida como **tratador de interrupção** e deve ficar posicionada em uma área da memória conhecida como vetor de interrupção. No controlador PIC16F628A, quando ocorre uma interrupção o PC é desviado para o **endereço 0x04 da memória de programa**, este endereço é conhecido como **vetor de interrupção** e é nele que deve está o tratador de interrupção.

# Tipos de Interrupção

- O PIC dispõe 10 Tipos de interrupções:
  - Interrupção externa:
    - Mudança de nível na porta RB0
  - Mudança de Estado
    - Mudança de valor em um dos pinos (RB4,RB5,RB6,RB7)
  - 3 Timers (TDVI)
  - 2 Comunicação Serial
  - 1 de comparador
  - 1 de Final de Escrita em ROM
  - 1 de CPP

# O que é uma interrupção?

- Quando uma interrupção ocorre, o fluxo normal de execução é desviado e o contador de programa passa a apontar para o vetor de interrupção 0x04. A sequência de instrução que estiver neste endereço será executada. Para retornar ao ponto de parada do programa usa-se a instrução **RETFIE**



# Configurando as interrupções

- As interrupções são habilitadas e configuradas usando o SFR INTCON (endereço 0BH)
- Abaixo o esquema dos bits do Registrador INTCON
  - Acesso: Se pode ser escrito e lido (RW) ou somente escrito
  - POR: Valor assumido na inicialização

INTCON (0Bh,8Bh,10Bh,18Bh)								
Bit	7	6	5	4	3	2	1	0
Função	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
Acesso	RW	RW	R	RW	RW	RW	RW	RW
POR	0	0	0	0	0	0	0	X





# Passos para configurar interrupções

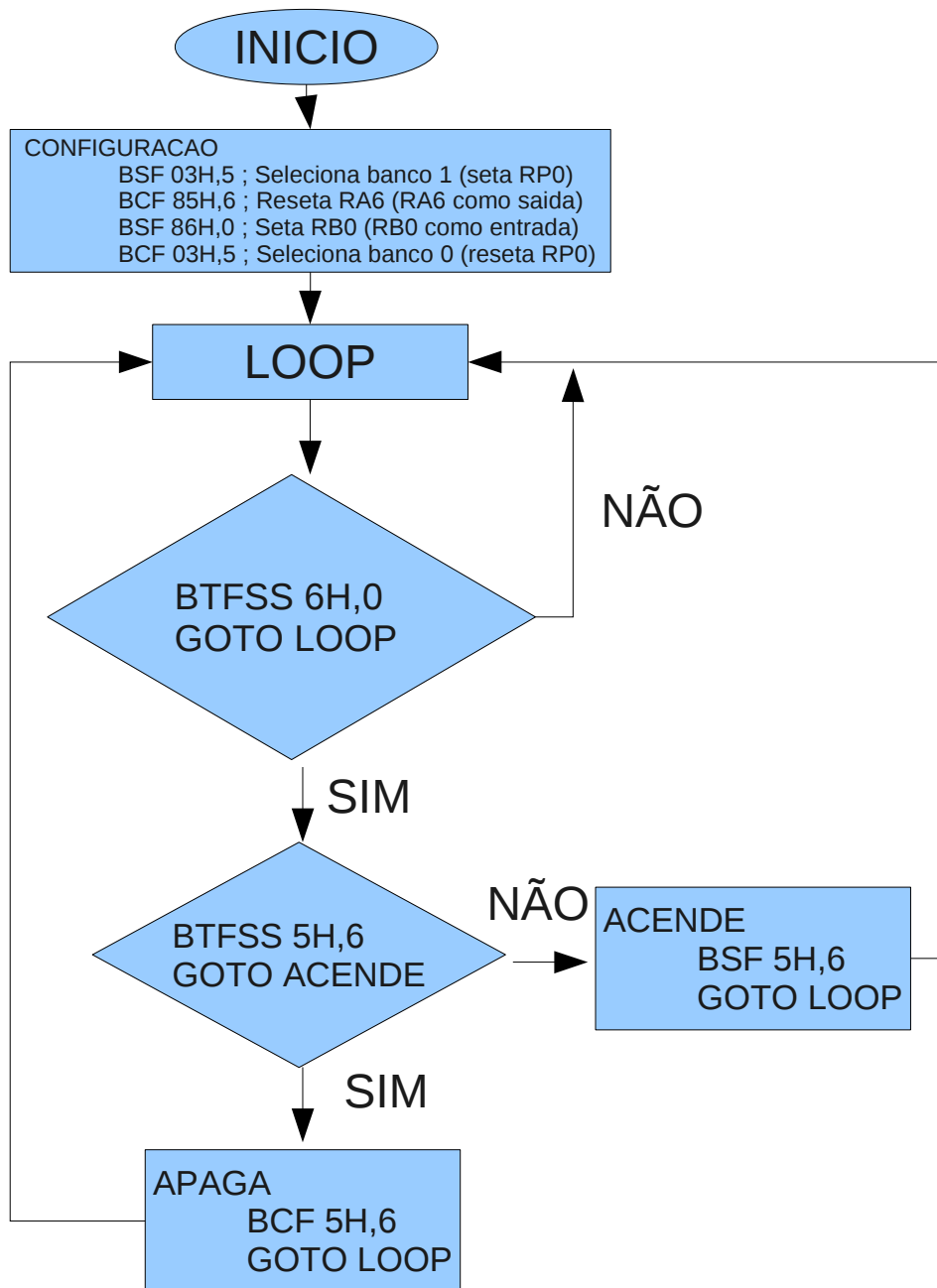
- Ex: Configurar interrupção em RB0

Código

```
BSF 0BH, 7; GIE=1  
BSF 0BH, 4; INTE=1
```

# Código (ex2-0.asm)

Código: ex2-0.asm

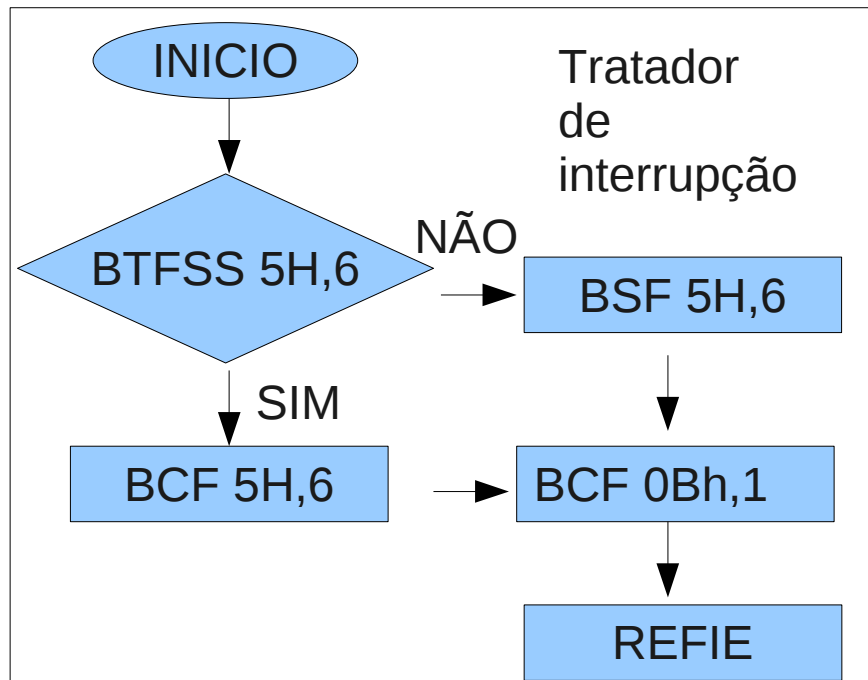
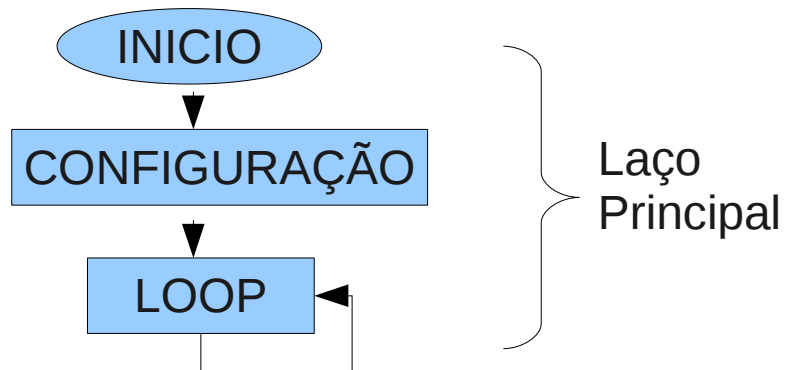


```
__CONFIG 0x3D18
ORG 0x00
INICIO
CONFIGURACAO
    BSF 03H,5 ; Selecciona banco 1 (seta RP0)
    BCF 85H,6 ; Reseta RA6 (RA6 como saida)
    BSF 86H,0 ; Seta RB0 (RB0 como entrada)
    BCF 03H,5 ; Selecciona banco 0 (reseta RP0)
    CLRF 05H
LOOP
    BTFSS 6H,0 ; Pula se botão Pressionado
    GOTO LOOP
    BTFSS 5H,6 ; Pula se Luz acessa
    GOTO ACENDE
    GOTO APAGA
ACENDE
    BSF 5H,6
    GOTO LOOP
APAGA
    BCF 5H,6
    GOTO LOOP
END
```

# Programando interrupções

- Em `ex2-0.asm` é possível trocar a checagem continua por interrupção em RB0

Código: `ex2-1.asm`

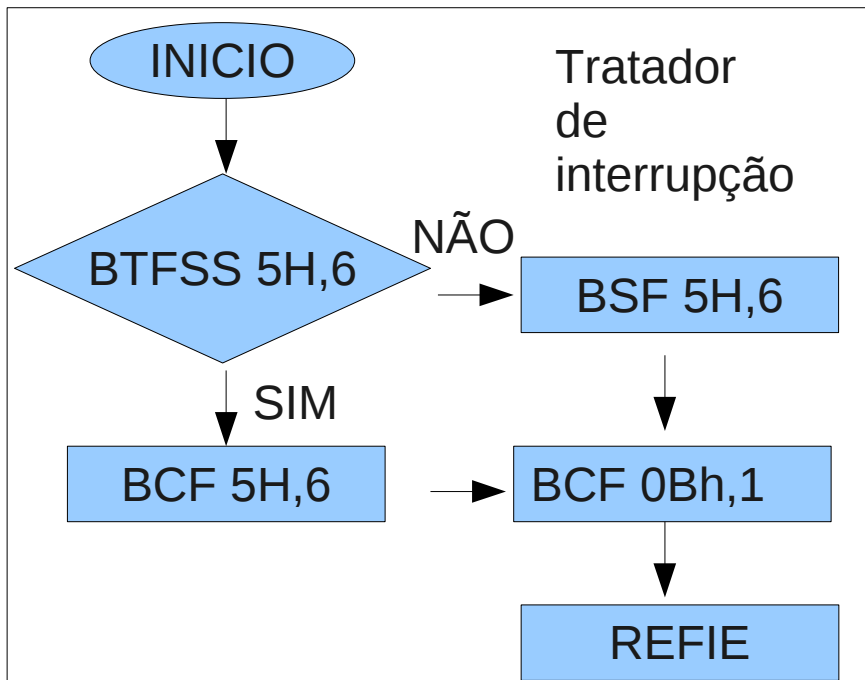
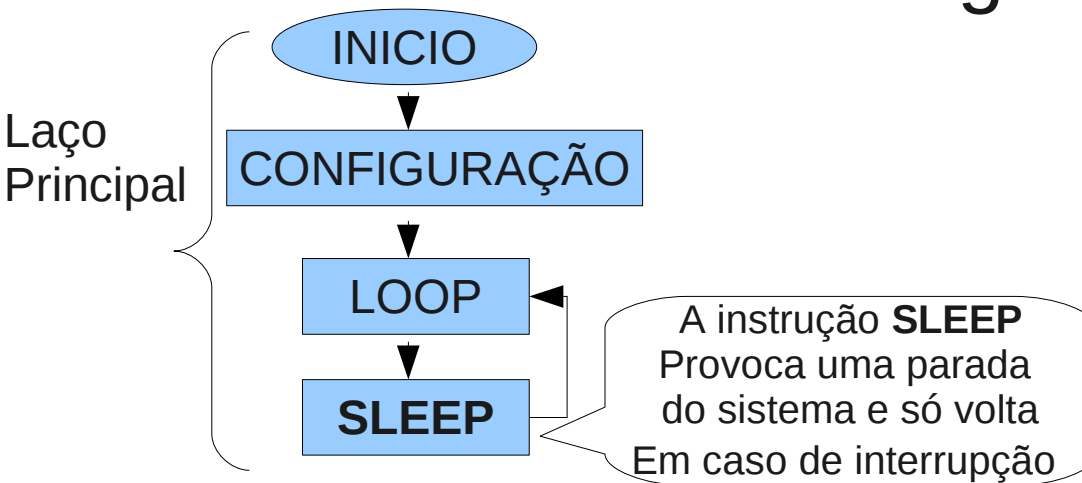


```
__CONFIG 0x2D18
ORG 0x00
GOTO CONFIGURACAO
ORG 0x04
    BTFSS 5H,6 ; Pula se Luz acesa
    GOTO ACENDE
    GOTO APAGA
ACENDE
    BSF 5H,6      ; Acende Led
    GOTO SAIINT
APAGA
    BCF 5H,6      ; Apaga o Led
SAIINT
    BCF 0Bh,1     ; Desliga a Flag INTF
    RETFIE
CONFIGURACAO
    BSF 0BH, 7; GIE=1 (Habilita int geral)
    BSF 0BH, 4; INTE=1 (habilita int externa)
    BSF 03H,5 ; Seleciona banco 1 (seta RP0)
    BCF 85H,6 ; Reseta RA6 (RA6 como saída)
    BSF 86H,0 ; Seta RB0 (RB0 como entrada)
    BCF 03H,5 ; Seleciona banco 0 (reseta RP0)
LOOP
    GOTO LOOP
END
```

# Economizando energia

- Economizando energia

Código: ex2-2.asm



```
__CONFIG 0x2D18
ORG 0x00
GOTO CONFIGURACAO
ORG 0x04
    BTFSS 5H,6 ; Pula se Luz acessa
    GOTO ACENDE
    GOTO APAGA
ACENDE
    BSF 5H,6
    GOTO SAIINT
APAGA
    BCF 5H,6
SAIINT
    BCF 0Bh,1
    RETFIE
CONFIGURACAO
    BSF 0BH, 7; GIE=1 (Habilita int geral)
    BSF 0BH, 4; INTE=1 (habilita int externa)
    BSF 03H,5 ; Selecciona banco 1 (seta RP0)
    BCF 85H,6 ; Reseta RA3 (RA3 como saida)
    BSF 86H,0 ; Seta RB0 (RB0 como entrada)
    BCF 03H,5 ; Selecciona banco 0 (reseta RP0)
LOOP
    SLEEP ; Poe o sistema em standby
    GOTO LOOP
END
```

# Processo de gravação

Linha de comando para Montagem no Live CD.  
Arquivos gerados:  
ex2-2.lst, ex2-2.cod, ex2-2.hex

**Código**  
ex2-2.asm

```
$ gpasm -p16f628a ex2-2.asm
```

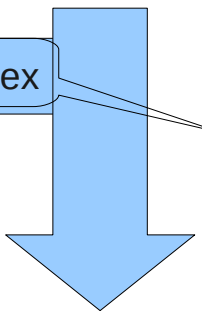


**executável: ex2-2.hex**

```
:020000040000FA  
:020000000C28CA  
:08000800051F07280928051750  
:100010000A2805138B1009008B170B16831605137E  
:0800200006148312630012288C  
:02400E00182D6B  
:00000001FF
```

Porta Serial

```
$ picprog --device=pic16f628a --erase --burn --pic-serial-port=/dev/ttyS0 --input-hexfile ex1.hex
```



Arquivo  
HEX

Linha de comando para Gravação no Live CD.  
O gravador deve está Conectado na porta serial



# Atividades Sugeridas

- 1) Modifique o programa de forma que fique mais precisa a atividade de acender e apagar do botão. (inserir verificação para se o botão foi solto)
- 2) Modifique o programa para que com um toque acenda um led verde (em RA3), com outro toque acenda o vermelho (em RA2) e apague o verde.
- 2) Depois você poderá modificar o programa para que com um toque acenda o verde, um segundo toque acenda o vermelho e o terceiro toque apague os dois.