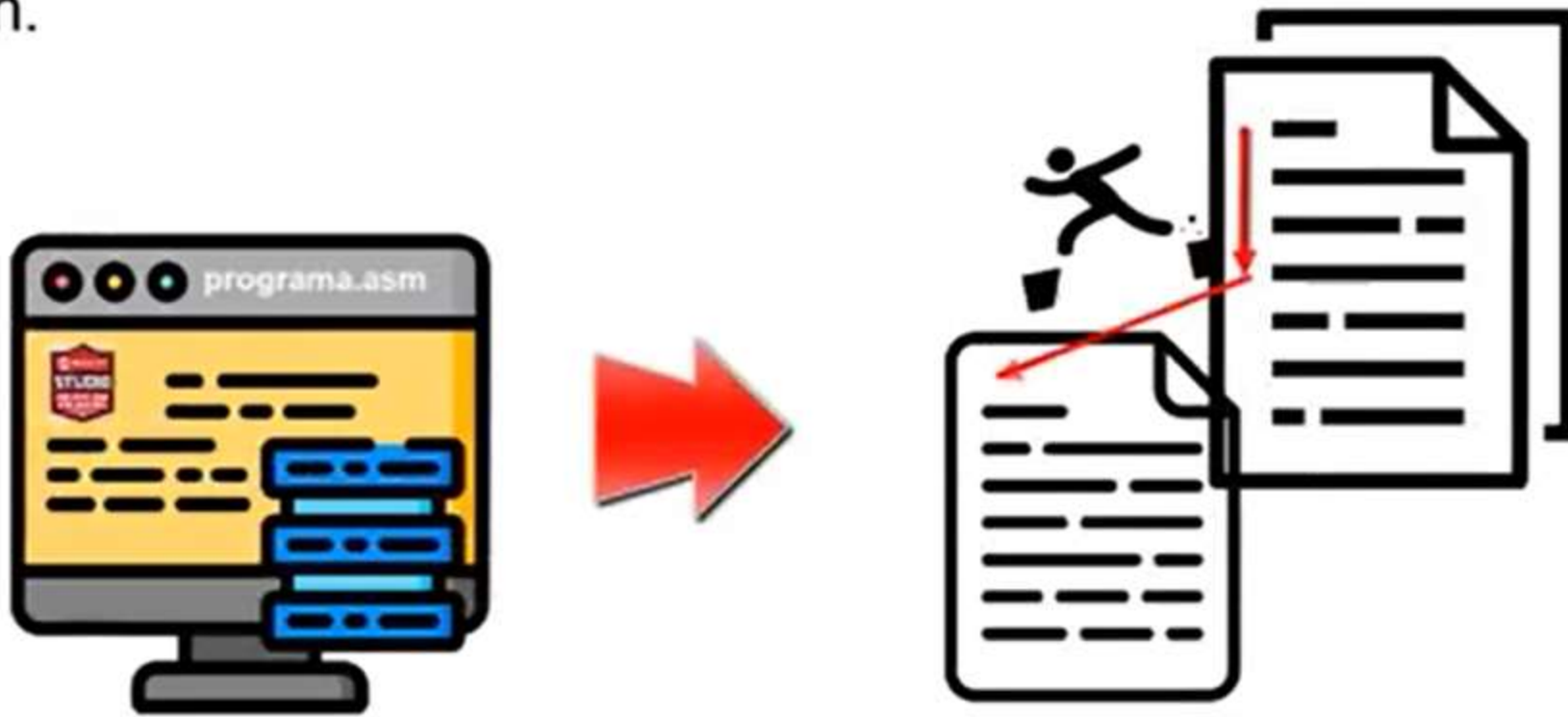


Instituto Tecnológico de Zitácuaro

Tema 4: **Salto Condicionales**

Salto condicional

Permiten controlar el flujo del programa, haciendo una toma de decisión a partir de una condición.





ATmega328p



Instrucciones
ensamblador

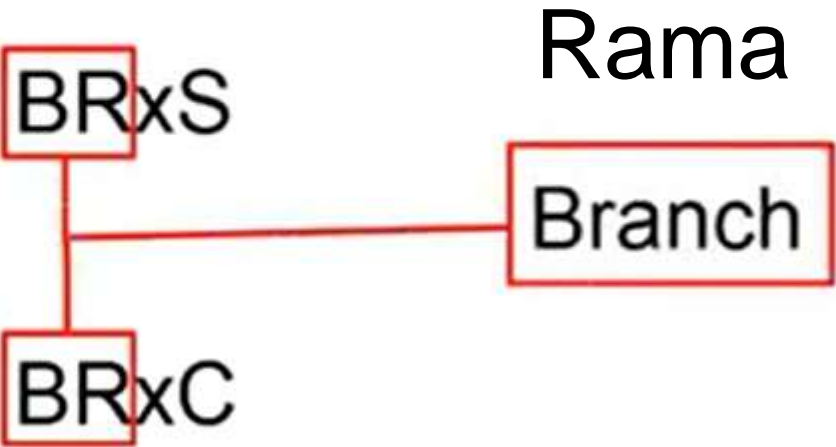


Registro de Estado

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
|-------|-------|-------|-------|-------|-------|-------|-------|
| I | T | H | S | V | N | Z | C |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

| Registro de Estado | | | | | | | |
|--------------------|----------|----------|----------|----------|----------|----------|----------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| I | T | H | S | V | N | Z | C |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

Mnemónicos:

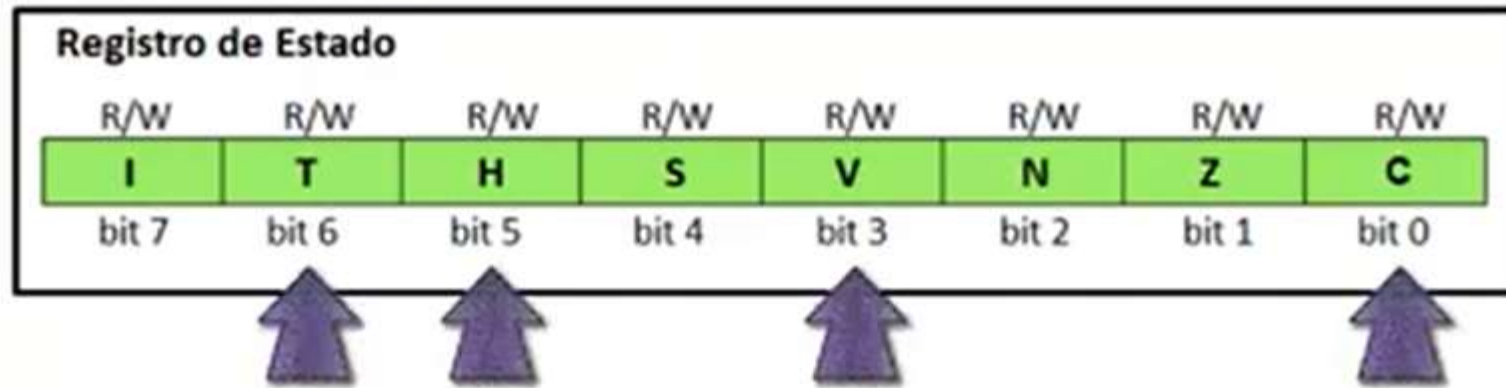


| Registro de Estado | | | | | | | |
|--------------------|----------|----------|----------|----------|----------|----------|----------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| I | T | H | S | V | N | Z | C |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

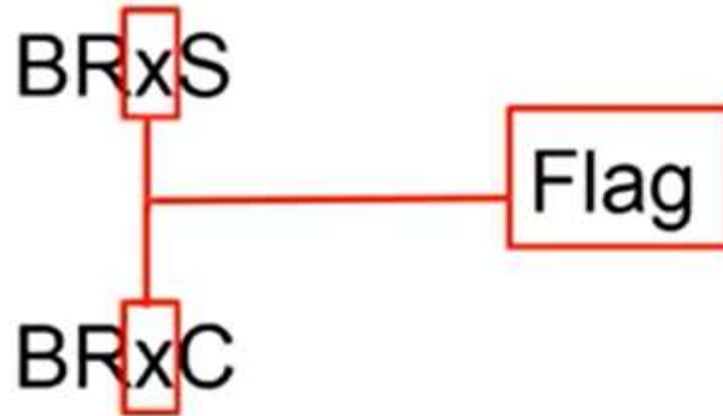
Mnemónicos:

BRx**S** ————— Set 1

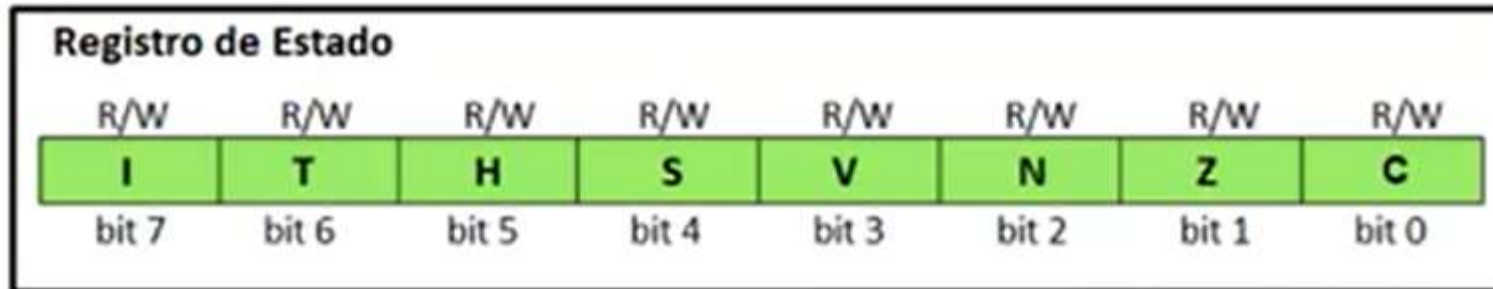
BRx**C** ————— Clear 0



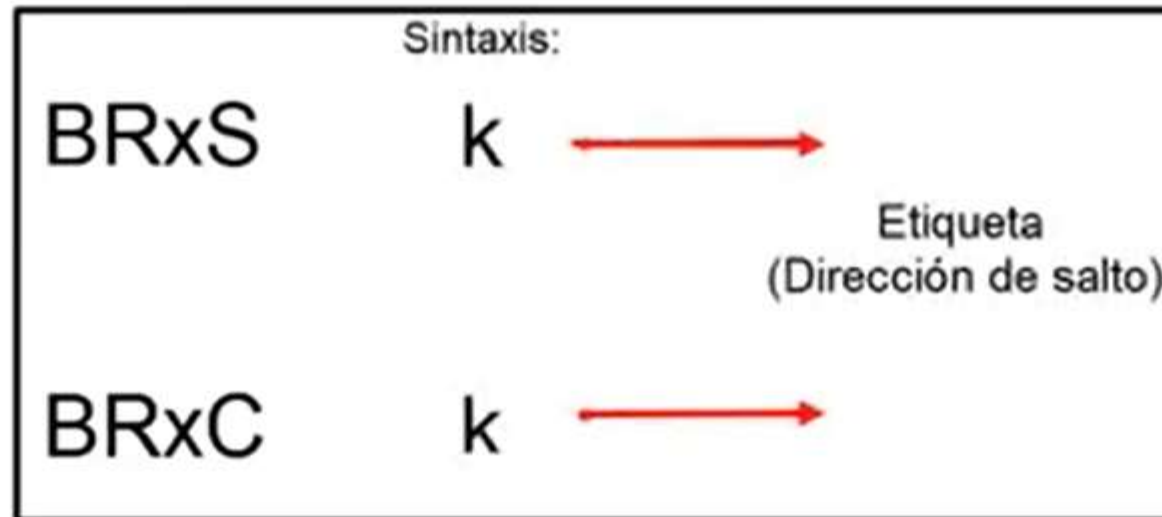
Mnemónicos:



**Uso de
Banderas**



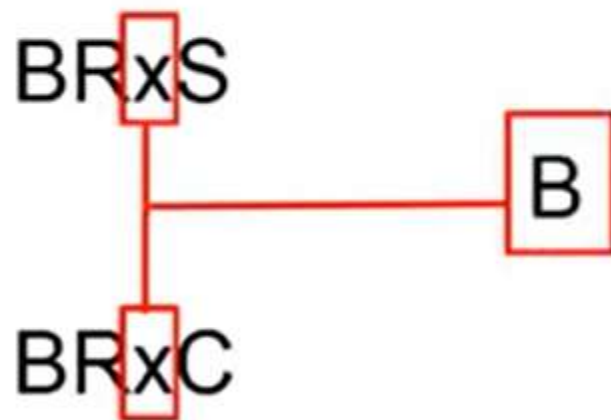
Mnemónicos:



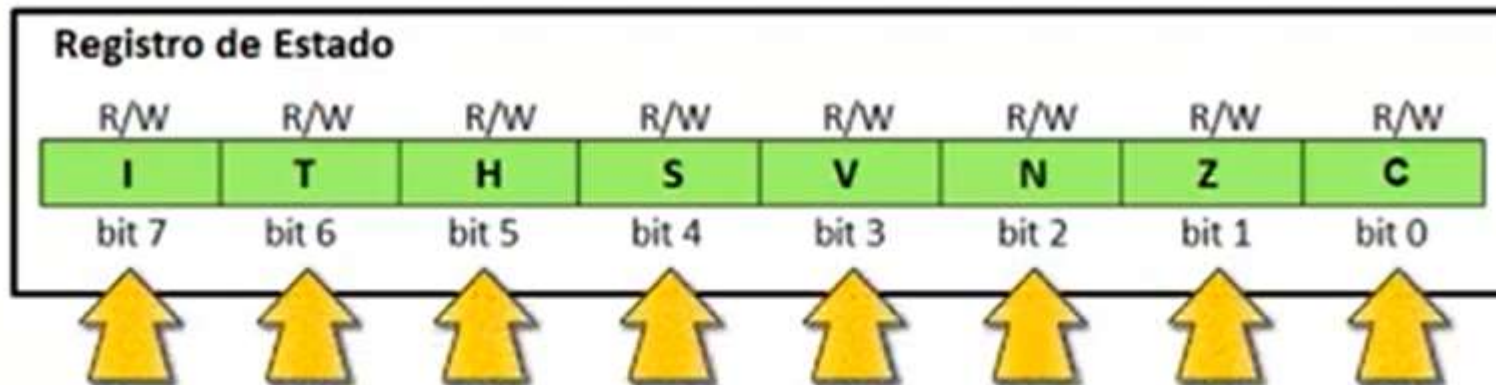
Uso de sintaxis con una etiqueta

| Registro de Estado | | | | | | | |
|--------------------|----------|----------|----------|----------|----------|----------|----------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| I | T | H | S | V | N | Z | C |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

Mnemónicos:



Para hacer uso de cualquier Registro de Estado se coloca la letra B



Mnemónicos:

Sintaxis:

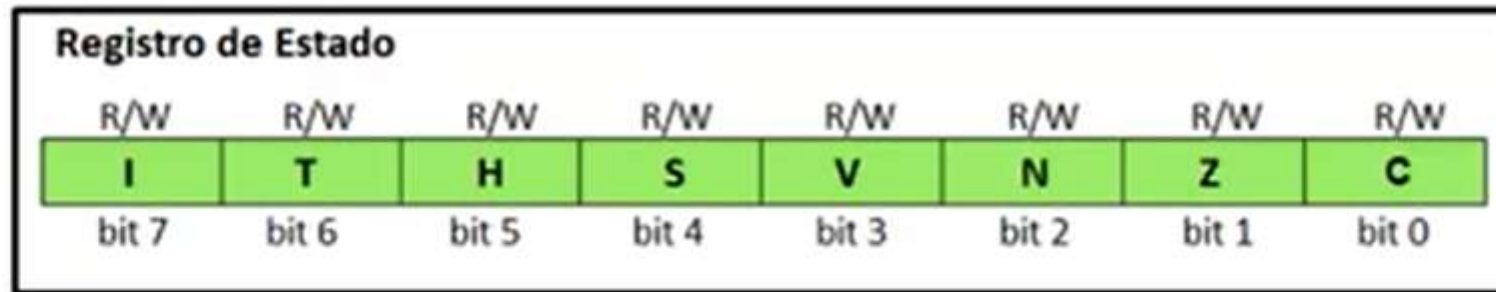
BRxS

S, k

No. Bit de SREG
7:0

BRxC

S, k



Mnemónicos:

BREQ → Igual

Hace un salto cuando se comparan dos valores iguales

| Registro de Estado | | | | | | | |
|--------------------|-------|-------|-------|-------|-------|-------|-------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| I | T | H | S | V | N | Z | C |
| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |

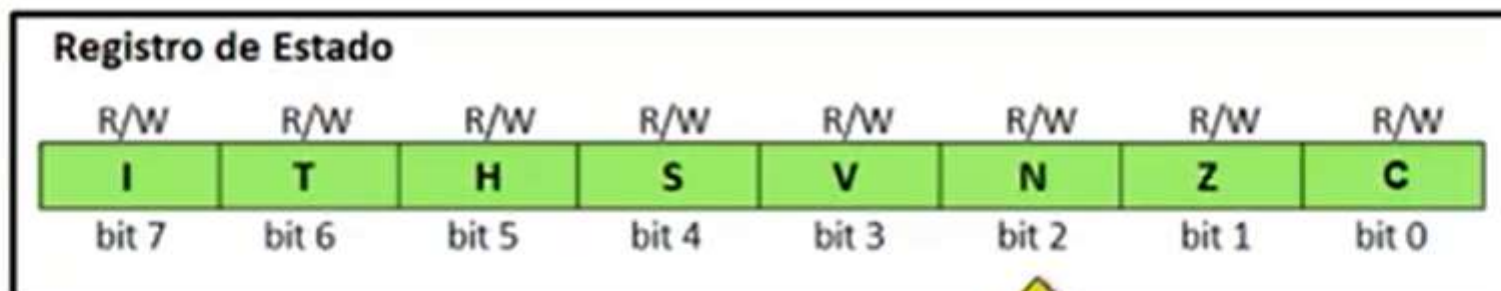
Mnemónicos:



$$a - b = 0$$

$$a = b$$





Mnemónicos:

BRMI

Minus (Negativo)

1

N

BRPL

Plus (Positivo)

0

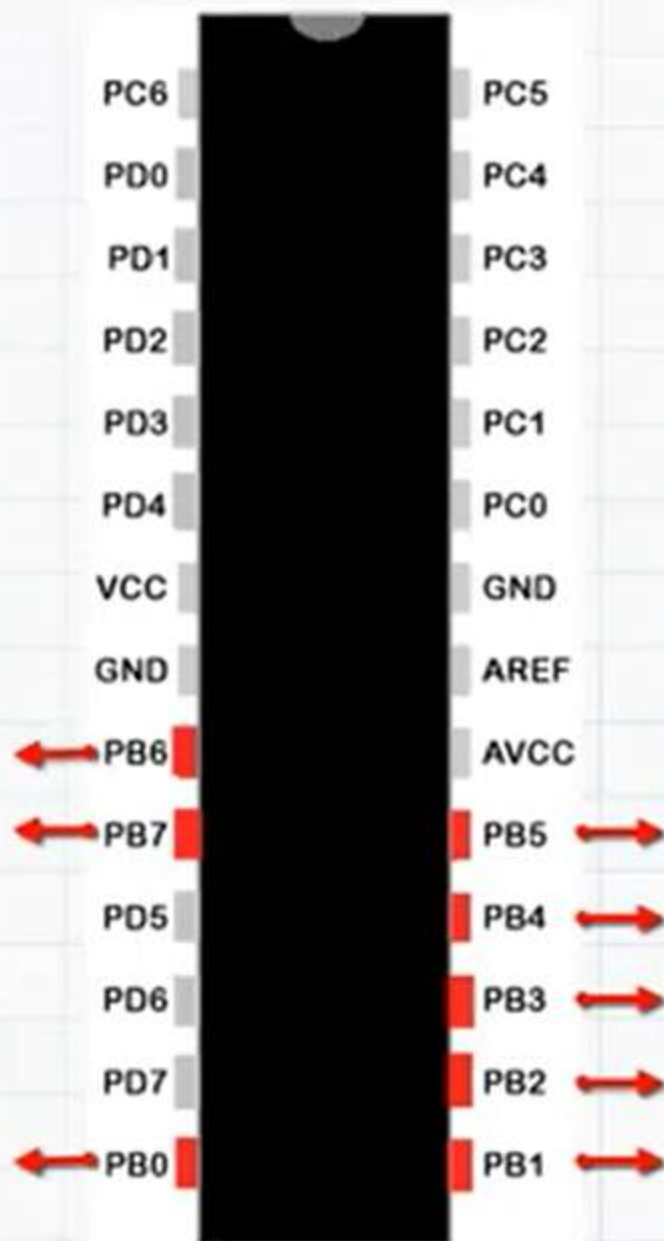
N

Ejemplo de aplicación

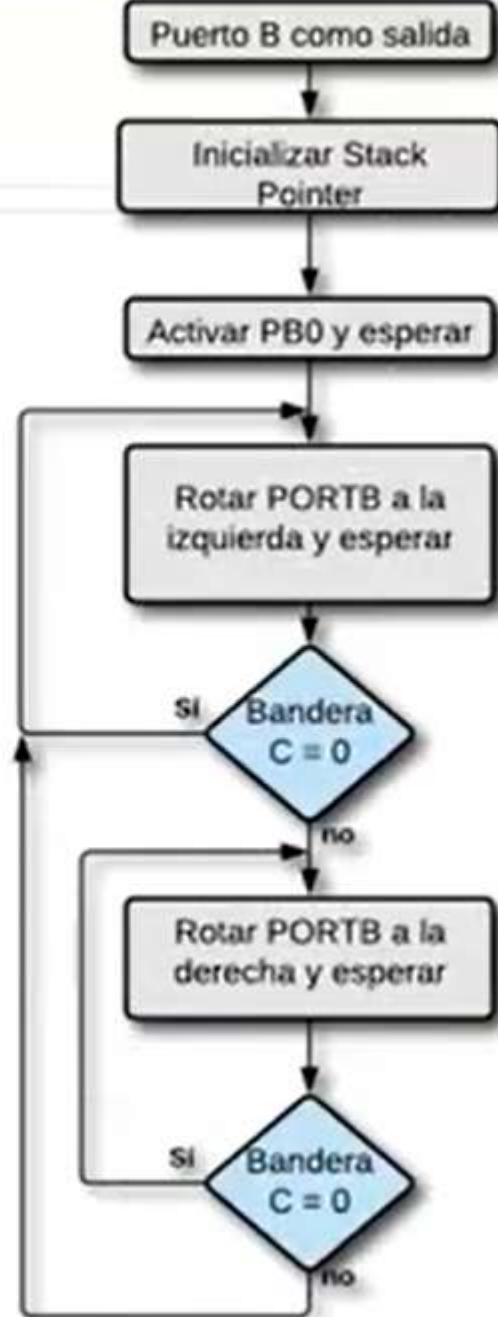
Mostrar por el puerto B la rotación de un bit activo hacia la izquierda y hacia la derecha de forma secuencial.



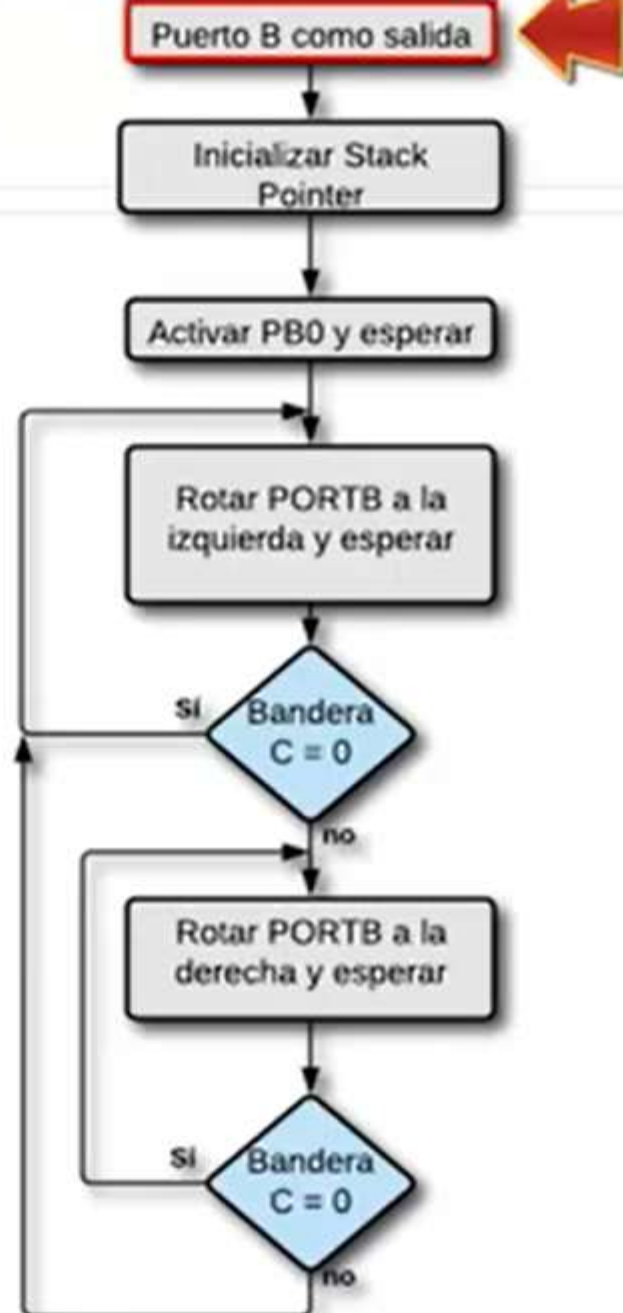
Barra de LED



```
; Saltos.asm  
;  
; Created: 08/03/2021 04:06:43 p.m.  
; Author : USUARIO0  
;
```



```
; Saltos.asm  
;  
; Created: 08/03/2021 04:06:43 p.m.  
; Author : USUARIO0  
;
```




```

; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

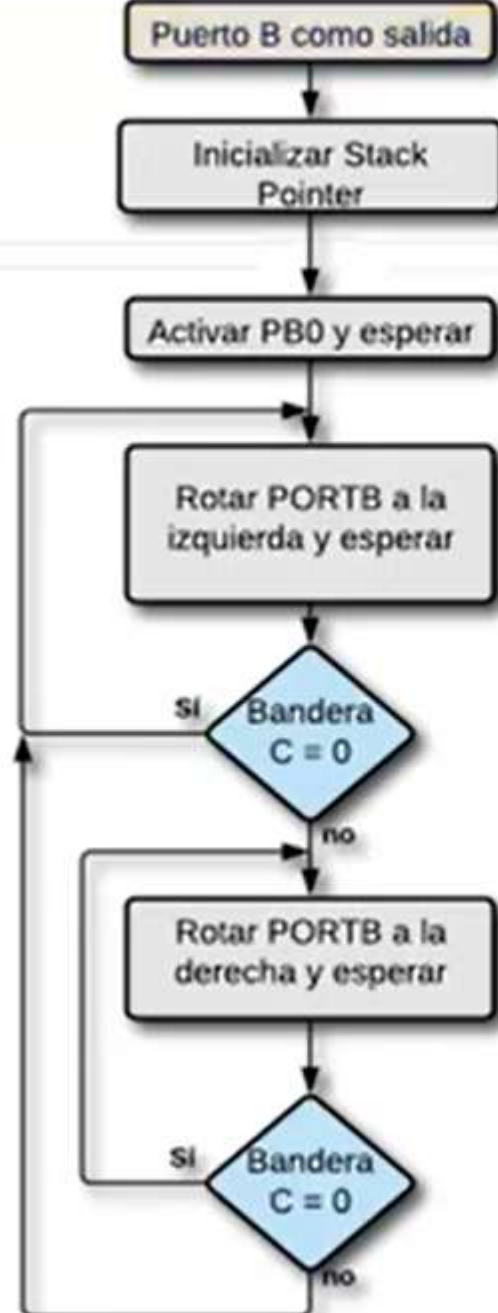
```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.

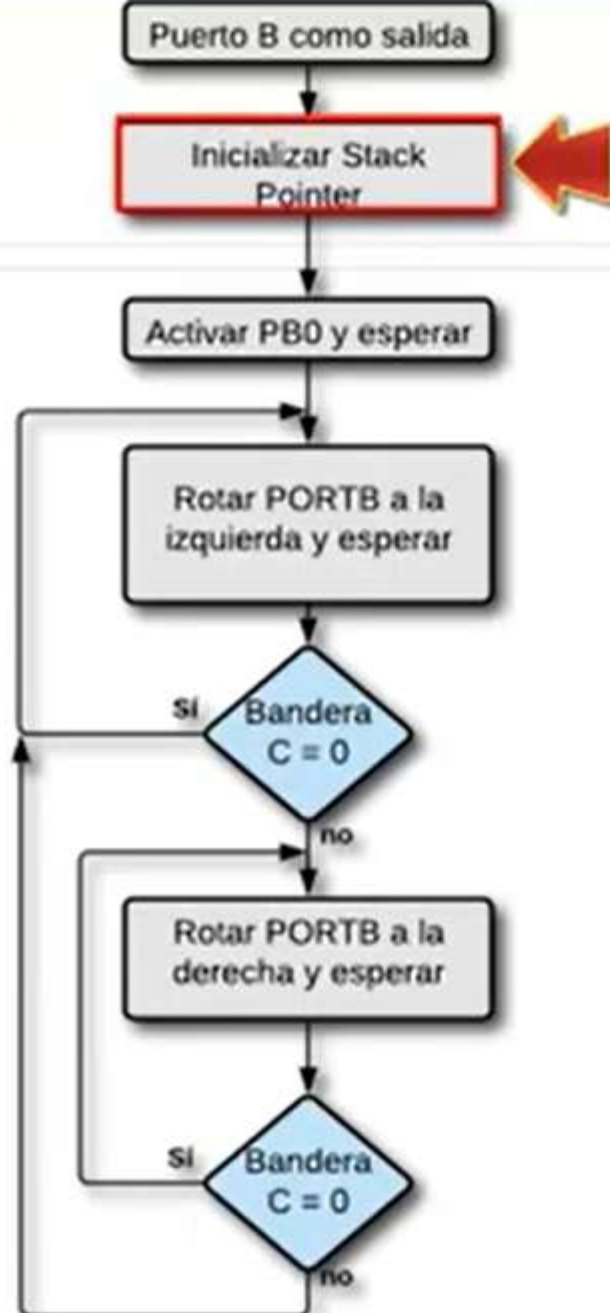
```

| | | |
|-----|--|------|
| PC6 | | PC6 |
| PD0 | | PC4 |
| PD1 | | PC3 |
| PD2 | | PC2 |
| PD3 | | PC1 |
| PD4 | | PC0 |
| VCC | | GND |
| GND | | AREF |
| PB6 | | AVCC |
| PB7 | | PB5 |
| PD5 | | PB4 |
| PD6 | | PB3 |
| PD7 | | PB2 |
| PB0 | | PB1 |



```
; Saltos.asm  
;  
; Created: 08/03/2021 04:06:43 p.m.  
; Author : USUARIO0  
;
```

```
LDI R16, 0xFF  
OUT DDOR, R16 ; Configura Puerto B como salida.
```



```

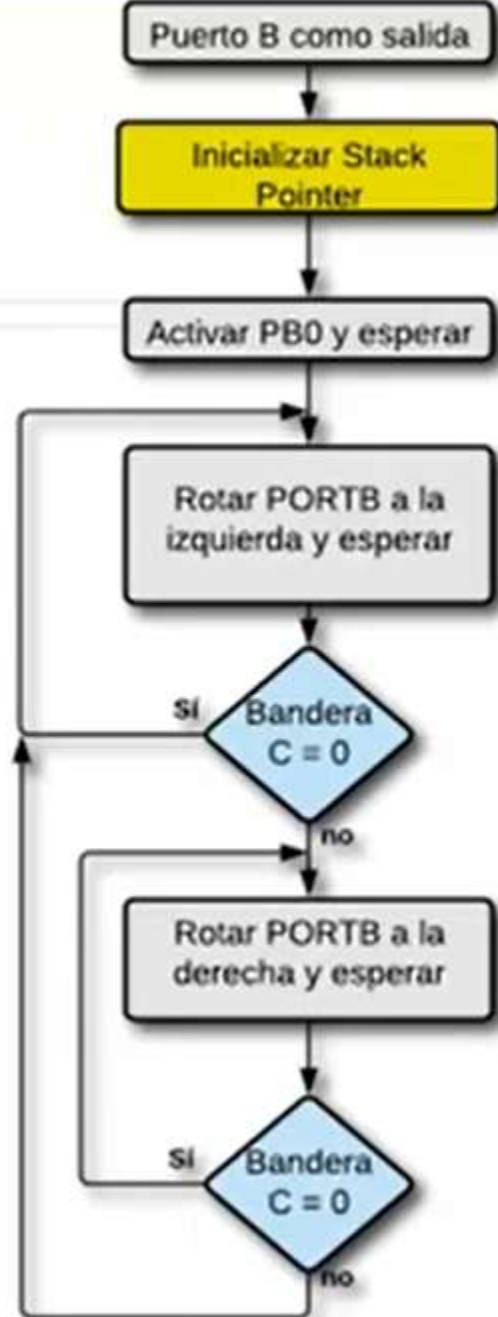
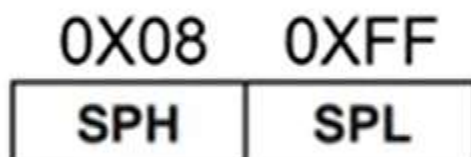
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x08
OUT SPL, R16
OUT SPH, R17

```



```

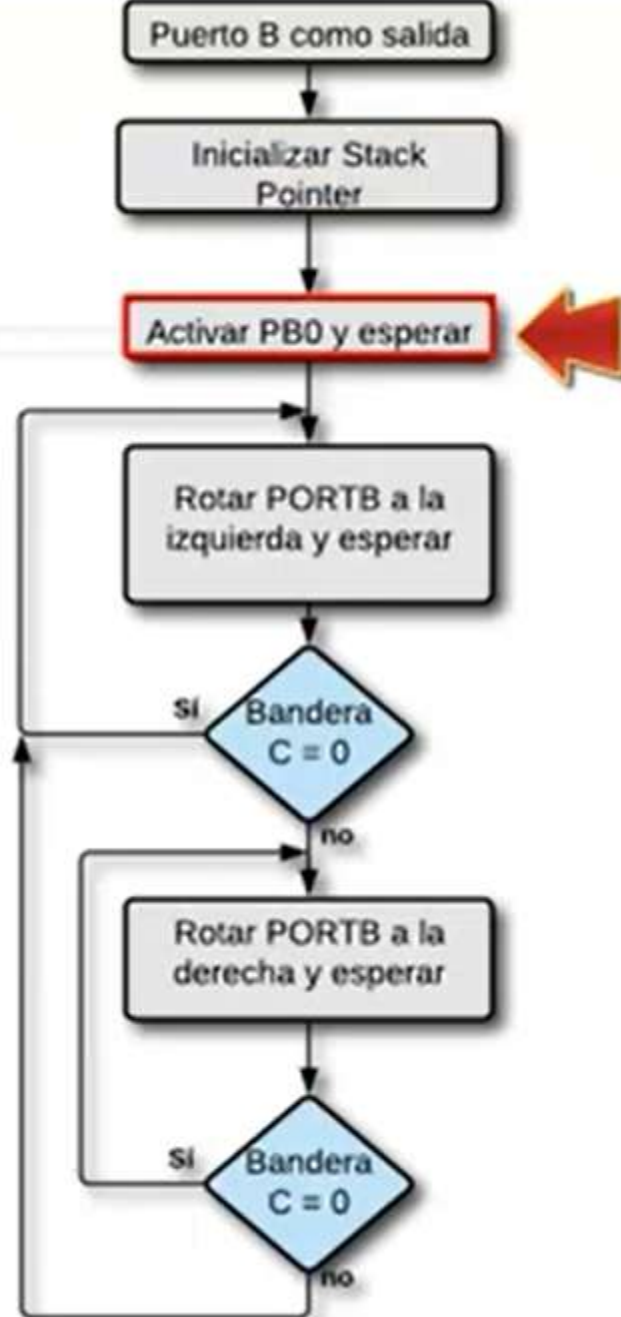
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.

```



```

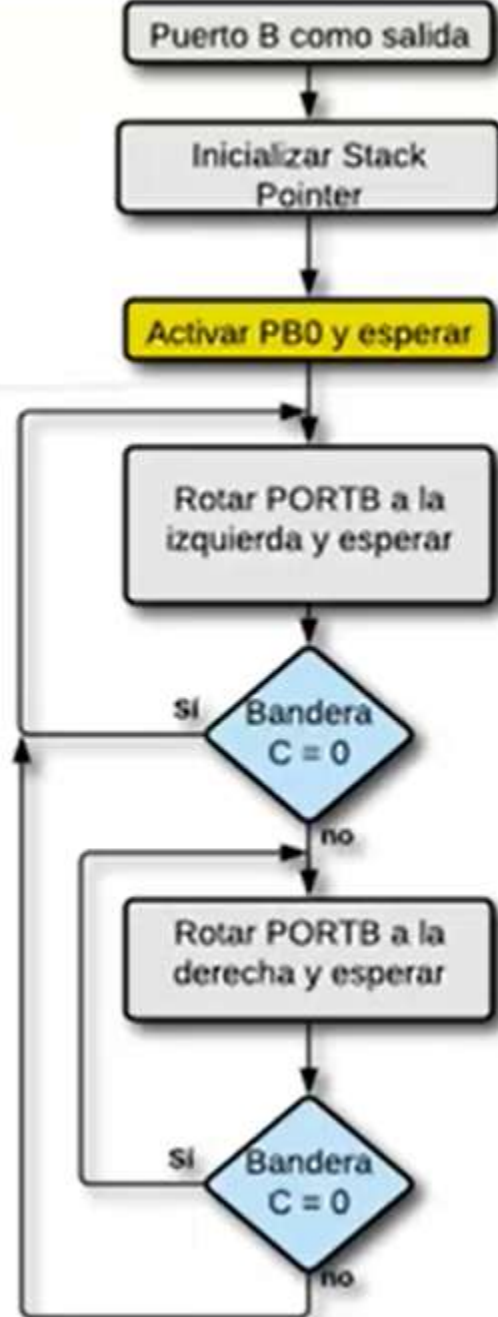
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x01
OUT PORTB, R18 ; Activa PB0.

```



```

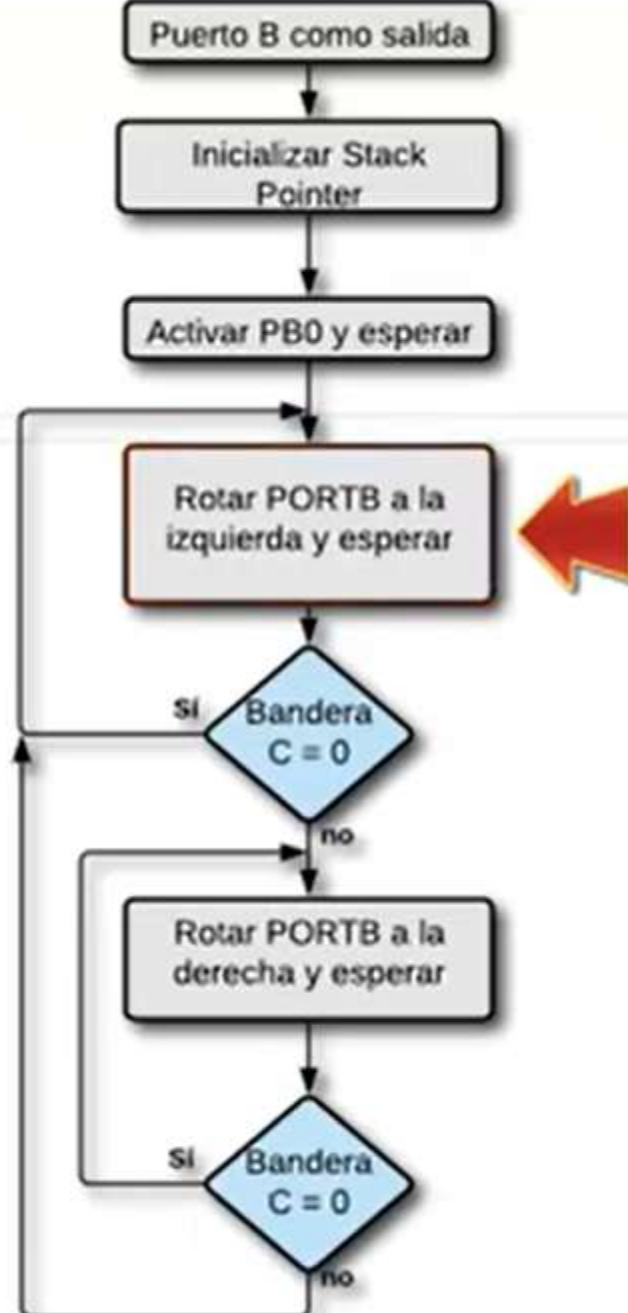
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI    R16, 0xFF
OUT    DDOR, R16    ; Configura Puerto B como salida.
LDI    R17, 0x0088
OUT    SPL, R16
OUT    SPH, R17    ; Inicializa Stack Pointer.
LDI    R18, 0x001
OUT    PORTB, R18  ; Activa PB0.
RCALL  Wait

```



```

; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x0001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL

```

| Mnemónico | Operandos | Descripción |
|-----------|-----------|---|
| ROL | Rd | Rotación hacia la izquierda con acarreo |




```

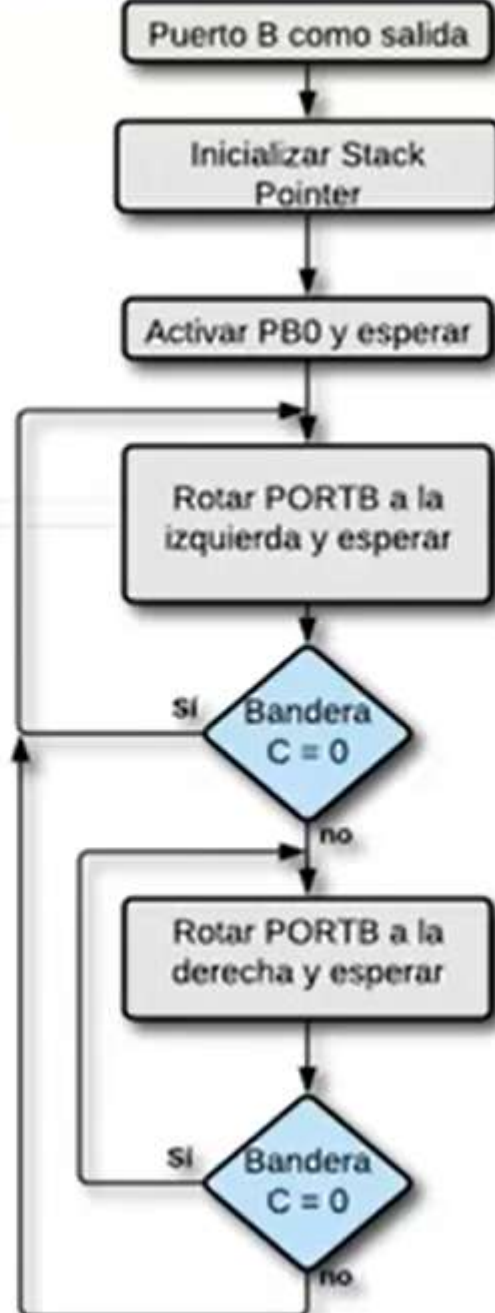
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x0001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait

```




```

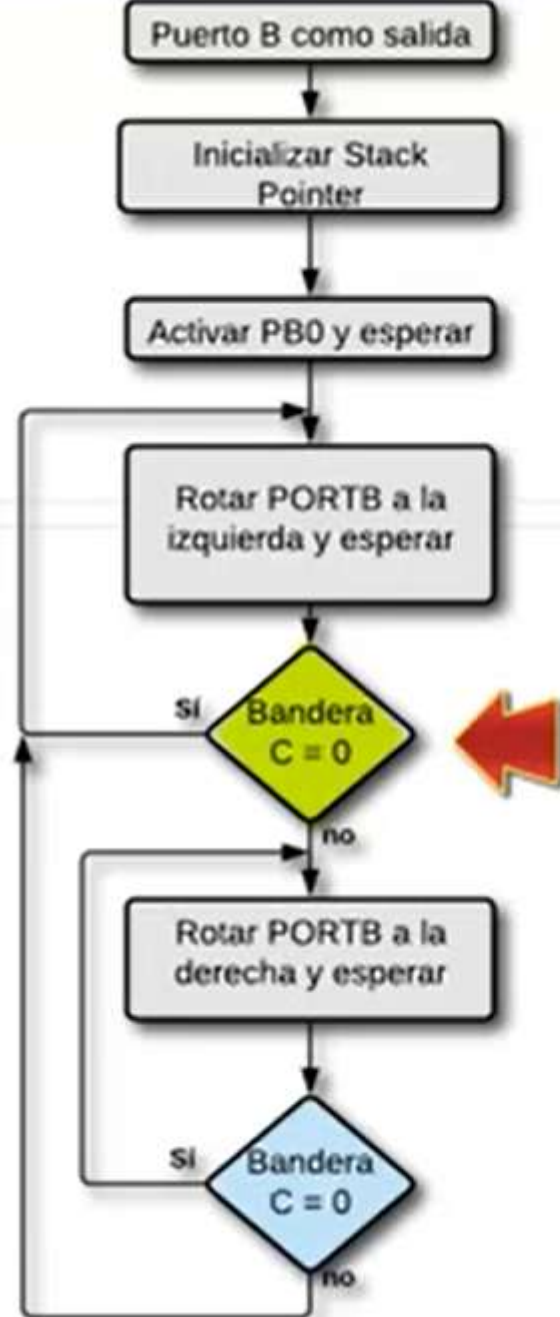
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq

```



```

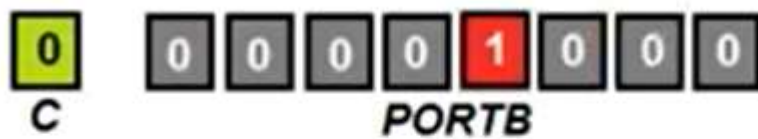
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x08
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x01
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq

```



```

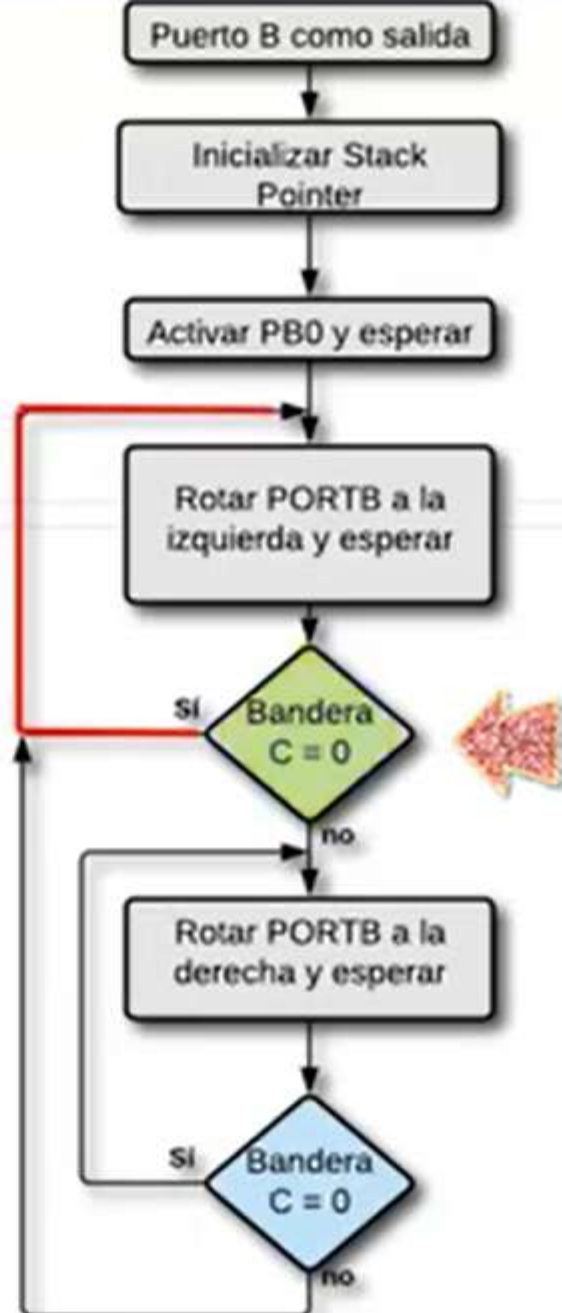
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq

```



```

; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x08
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x01
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq

```



```

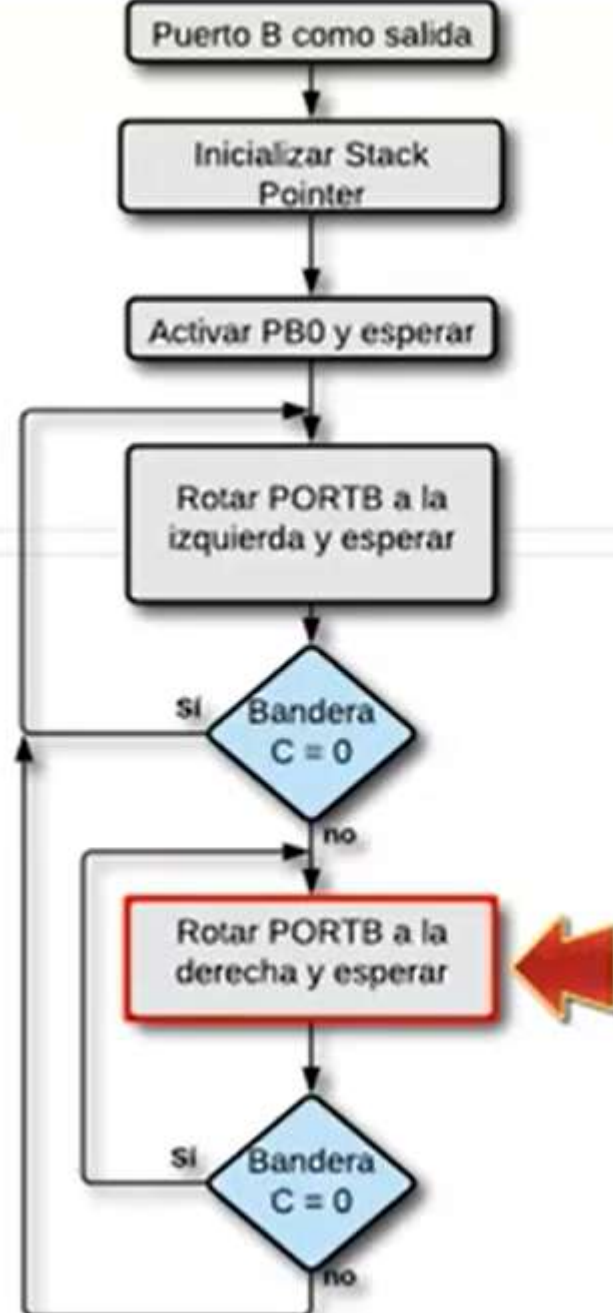
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PBO.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq ; Salta si la bandera C es igual a cero.

```



```

; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

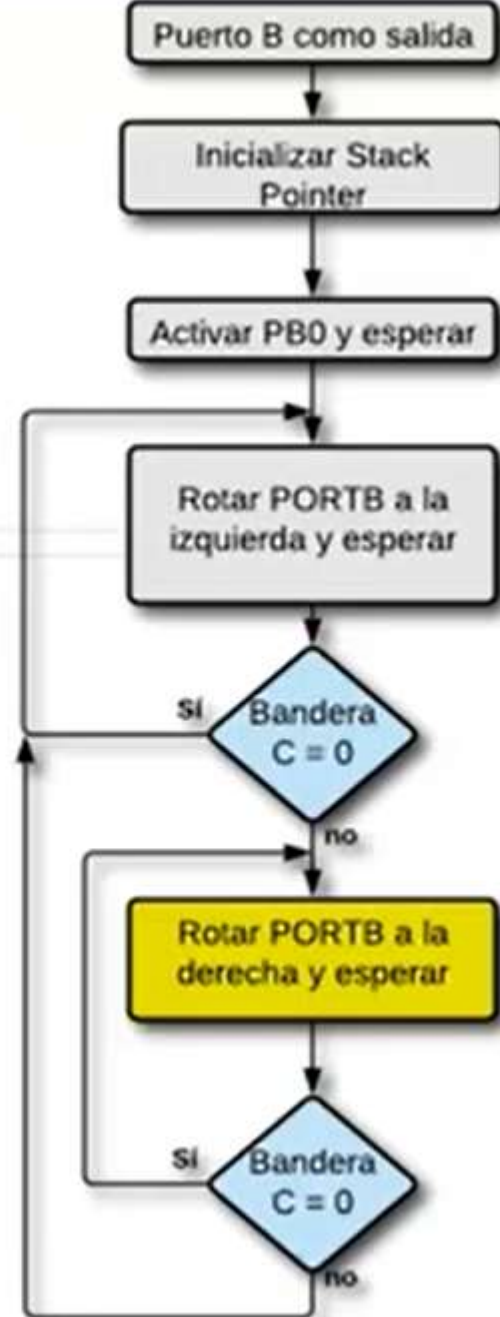
```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq ; Salta si la bandera C es igual a cero.
Der: RO

```

| Mnemónico | Operandos | Descripción |
|-----------|-----------|---------------------------------------|
| ROR | Rd | Rotación hacia la derecha con acarreo |




```

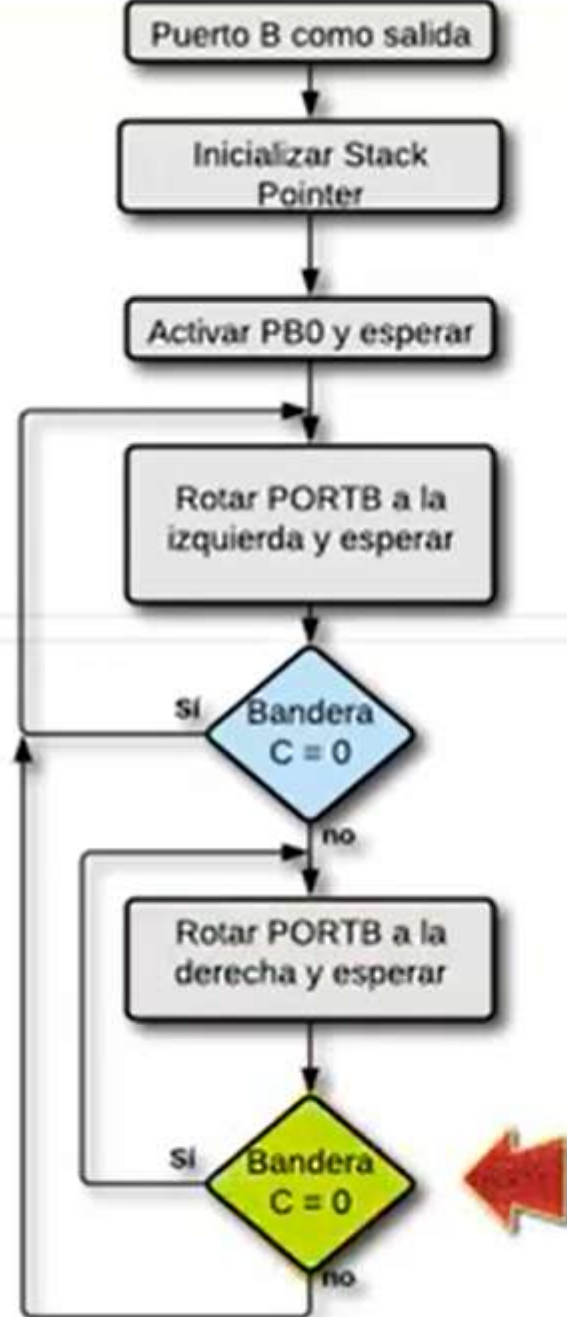
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPM, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq ; Salta si la bandera C es igual a cero.
Der: ROR R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la derecha.
RCALL Wait
BRCC Der

```



```

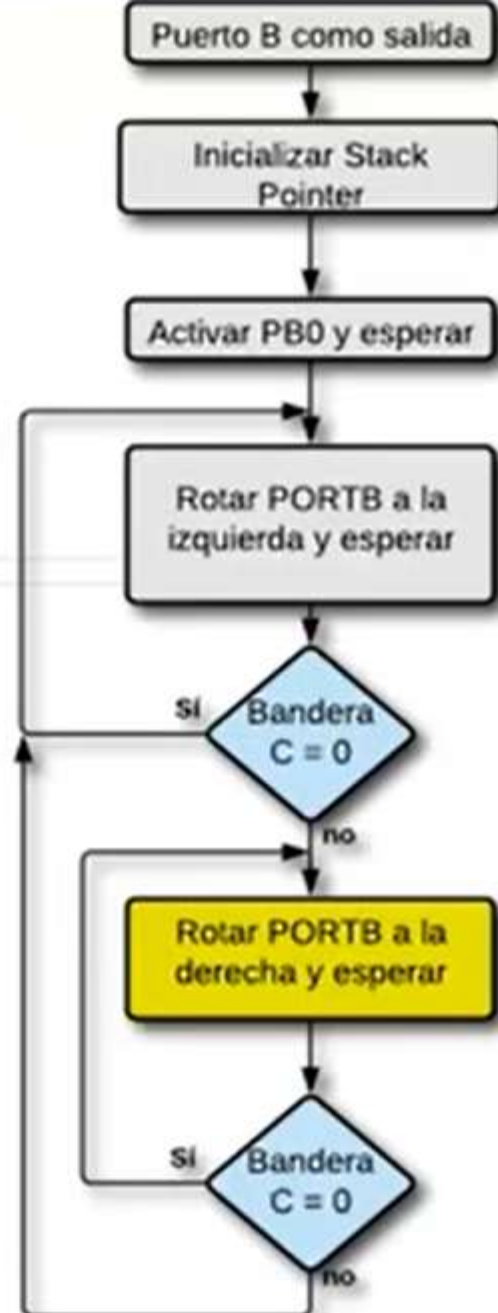
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Irq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Irq ; Salta si la bandera C es igual a cero.
Der: ROR R18
OUT PORTB, R18

```




```

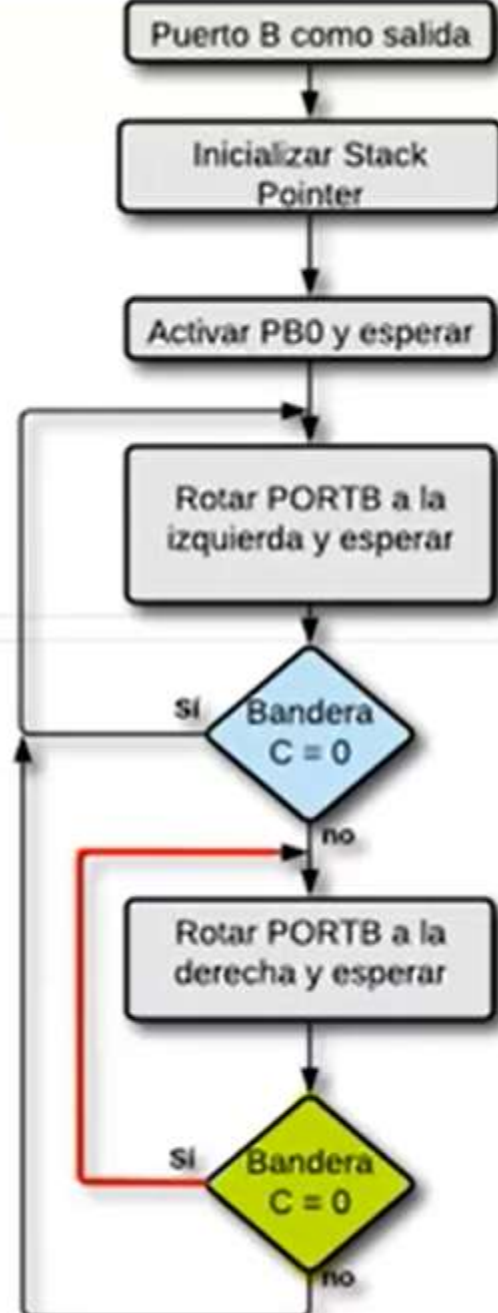
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x0008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x0001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq ; Salta si la bandera C es igual a cero.
Der: ROR R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la derecha.
RCALL Wait
BRCC Der

```



```

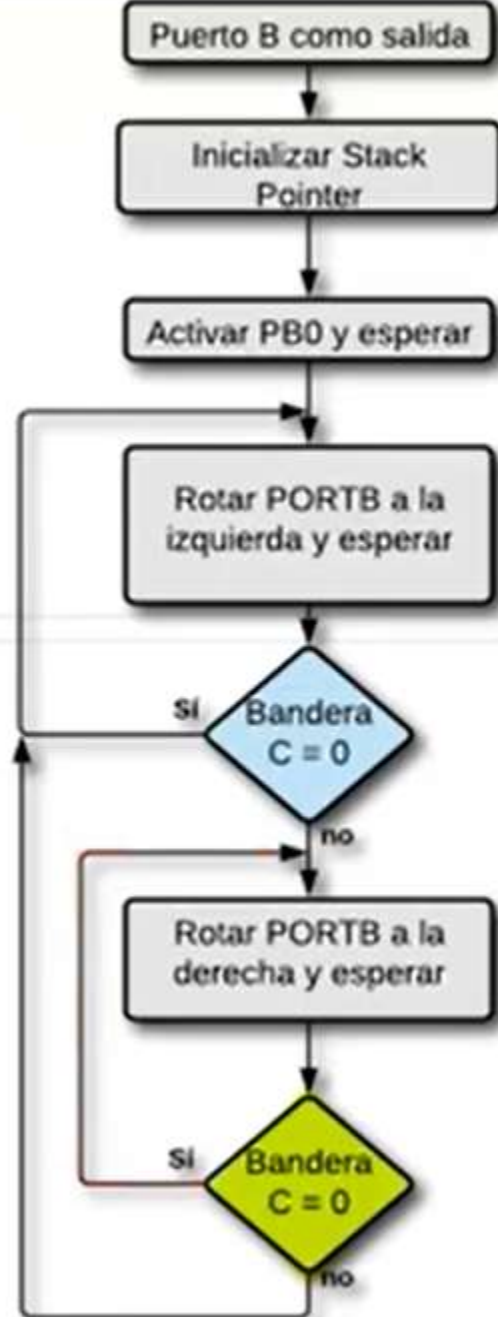
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

```

```

LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x001
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq ; Salta si la bandera C es igual a cero.
Der: ROR R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la derecha.
RCALL Wait
BRCC Der

```



```

; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

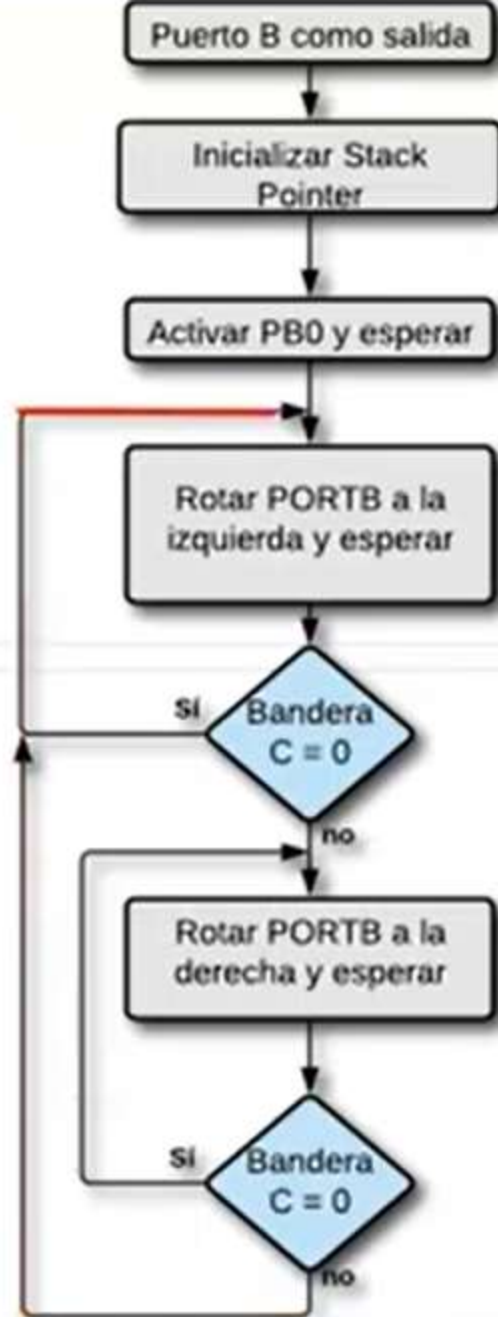
```

```

; Izq:
LDI R16, 0xFF
OUT DDOR, R16 ; Configura Puerto B como salida.
LDI R17, 0x008
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x01
OUT PORTB, R18 ; Activa PB0.
RCALL Wait
; Der:
ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq ; Salta si la bandera C es igual a cero.
ROR R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la derecha.
RCALL Wait
BRCC Der
Rjmp Izq

```

I



```
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;
```

```
    LDI    R16, 0xFF
    OUT    DDOR8, R16    ; Configura Puerto B como salida.
    LDI    R17, 0x0008
    OUT    SPL, R16
    OUT    SPH, R17      ; Inicializa Stack Pointer.
    LDI    R18, 0x0001
    OUT    PORTB, R18    ; Activa PDB.
    RCALL  Wait
Izq: ROL    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la izquierda.
    RCALL  Wait
    BRCC   Izq           ; Salta si la bandera C es igual a cero.
Der: ROR    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la derecha.
    RCALL  Wait
    BRCC   Der
    RJMP   Izq
```

Wait:

Escribir la rutina de retardo



avr delay loop calculator



Buscar con Google

Me siento con suerte

Ofrecido por Google en [Español \(Latinoamérica\)](#)



Cerca de 576,000 resultados (0.57 segundos)

darcy.rsgc.on.ca > ACES > AVRdel... Traducir esta página

[AVR Delay Loop Calculator - RSGC ACES](#)

AVR Delay Loop Calculator. Developed originally by Bret Mulvey. Register enhancement by T. Morland. (ACES '16). MHz microcontroller clock frequency.

www.avrfreaks.net > forum > Impr... Traducir esta página

[Improved delay loop calculator | AVR Freaks](#)

2 ene. 2011 16 publicaciones 7 autores

url=http://www.avrprojects.net/index.php?option=com_content&view=article&id=83:avr-delay-calculator&catid=39:avr-tools&Itemid=59] ...

| | | |
|--|------------------|--------------------|
| Delay in Assembler AVR Freaks | 39 publicaciones | 26 de sep. de 2015 |
| AVR Delay Functions AVR Freaks | 34 publicaciones | 20 de oct. de 2006 |
| Calculate the maximum delay AVR Freaks | 17 publicaciones | 9 de abr. de 2010 |
| Calculate the maximum delay possible ... | 12 publicaciones | 25 de feb. de 2016 |

Más resultados de www.avrfreaks.net

web.csulb.edu > ~hill > Lectures > 06 AVR Looping Traducir esta página

[AVR Looping](#)

500 μ s. DELAY CALCULATION FOR AVR. • We begin by designing a simple loop, wait. ldi r16, ...

www.softpedia.com > Science-CAD Traducir esta página

[Download AVR delay loop generator 1.2 - Softpedia](#)

23 may. 2012 — A delay loops calculator. AVR delay loop generator is a compact program designed to generate delay loops for the ATMEL AVR controllers.

Preguntas relacionadas

How are AVR delays calculated?

How do I add a delay in assembly language?

AVR Delay Loop Calculator

Developed originally by [Bret Hulvey](#). Register enhancement by T. Morland. (ACES '18)

MHz microcontroller clock frequency

cycles for `rcall`/`ret` or other overhead

first register to be used by delay loop

ns us ms s mins hrs days

cycles

☒ assembler ☐ avr-gcc

```
; Assembly code auto-generated  
; by utility from Bret Hulvey  
; Delay 300 000 cycles  
; 300ms at 1.0 MHz
```

```
    ldi r19, 2  
    ldi r20, 134  
    ldi r21, 154  
L1: dec r21  
    brne L1  
    dec r20  
    brne L1  
    dec r19  
    brne L1
```

Se configura a 1 MHZ.

Registro utilizado R19

Ajustar a 0.3 ms

AVR Delay Loop Calculator

Developed originally by [Bret Mulvey](#). Register enhancement by T. Morland. (ACES '18)

MHz microcontroller clock frequency

cycles for `rcall/ret` or other overhead

first register to be used by delay loop

ns us ms s mins hrs days

cycles

☒ assembler ☐ avr-gcc

```
; Assembly code auto-generated  
; by utility from Bret Mulvey  
; Delay 300 000 cycles  
; 300ms at 1.0 MHz
```

```
ldi r19, 2  
ldi r20, 134  
ldi r21, 154  
L1: dec r21  
brne L1  
dec r20  
brne L1  
dec r19  
brne L1
```

Copiar y pegar


```
;
; Saltos.asm
;
; Created: 03/03/2021 @4:06:43 p.m.
; Author : USUARIO
;

    LDI    R16, 0xFF
    OUT    DDRC, R16    ; Configura Puerto D como salida.
    LDI    R17, 0x08
    OUT    SPC, R16
    OUT    SPH, R17    ; Inicializa Stack Pointer.
    LDI    R18, 0x01
    OUT    PORTD, R18    ; Activa PDD.
    RCALL  Wait
Irq:   ROL    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la izquierda.
    RCALL  Wait
    BRCC   Irq           ; Salta si la bandera C es igual a cero.
Der:   ROR    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la derecha.
    RCALL  Wait
    BRCC   Der
    RJMP   Irq

Wait:  ldi    r19, 2
    ldi    r20, 134
    ldi    r21, 154
L1:    dec    r21
    brne   L1
    dec    r20
    brne   L1
    dec    r19
    brne   L1
```

```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO
;

    LDI    R16, 0xFF
    OUT    DDRC, R16    ; Configura Puerto D como salida.
    LDI    R17, 0x08
    OUT    SPL, R16
    OUT    SPH, R17    ; Inicializa Stack Pointer.
    LDI    R18, 0x01
    OUT    PORTB, R18    ; Activa PDB.
    RCALL  Wait
Irq:  ROL    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la izquierda.
    RCALL  Wait
    BRCC   Irq            ; Salta si la bandera C es igual a cero.
Der:  ROR    R18
    OUT    PORTB, R18    ; Desplaza el bit activo hacia la derecha.
    RCALL  Wait
    BRCC   Der
    RJMP   Irq

Wait: ldi    r19, 2
    ldi    r20, 134
    ldi    r21, 154
    L1:  dec    r21
    brne   L1
    dec    r20
    brne   L1
    dec    r19
    brne   L1
    RET
```

Agregar RET

```

; Configura el puerto B como salida.
LDI R17, 0x00
OUT SPL, R16
OUT SPH, R17 ; Inicializa Stack Pointer.
LDI R18, 0x01
OUT PORTB, R18 ; Activa PDB.
RCALL Wait
Izq: ROL R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la izquierda.
RCALL Wait
BRCC Izq ; Salta si la bandera C es igual a cero.
Der: ROR R18
OUT PORTB, R18 ; Desplaza el bit activo hacia la derecha.
RCALL Wait
BRCC Der
RJMP Izq

Wait: ldi r19, 2
      ldi r20, 134
      ldi r21, 154
L1: dec r21
    brne L1
    dec r20
    brne L1
    dec r19
    brne L1
    RET

```

Output

Show output from: Build

Build

```

[.cseg] 0x000000 0x000036 54 0 54 32768 0.2%
[.dseg] 0x000100 0x000100 0 0 0 2048 0.0%
[.eseg] 0x000000 0x000000 0 0 0 1024 0.0%

```

Assembly complete, 0 errors, 0 warnings

Done executing task "RunAssemblerTask".

Done building target "CoreBuild" in project "Saltos.asmproj".

Target "PostBuildEvent" skipped, due to false condition; ('\$(PostBuildEvent)' != '') was evaluated as ('' != '').

Target "Build" in file "C:\Program Files\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Saltos\Saltos\Saltos.asmproj" (entry point):

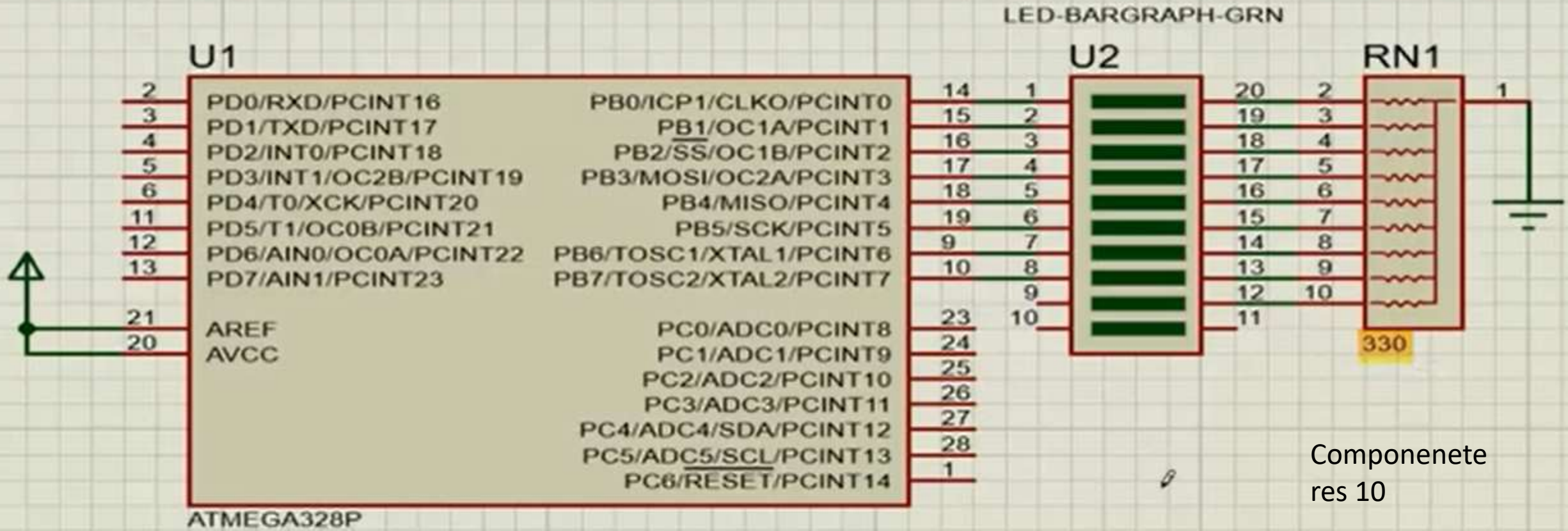
Done building target "Build" in project "Saltos.asmproj".

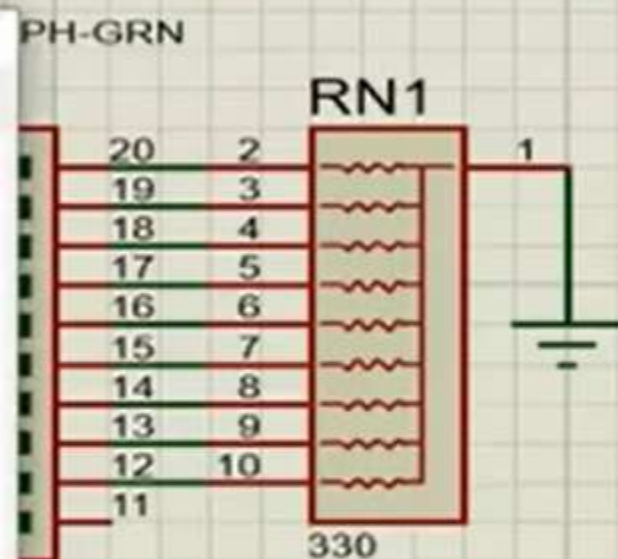
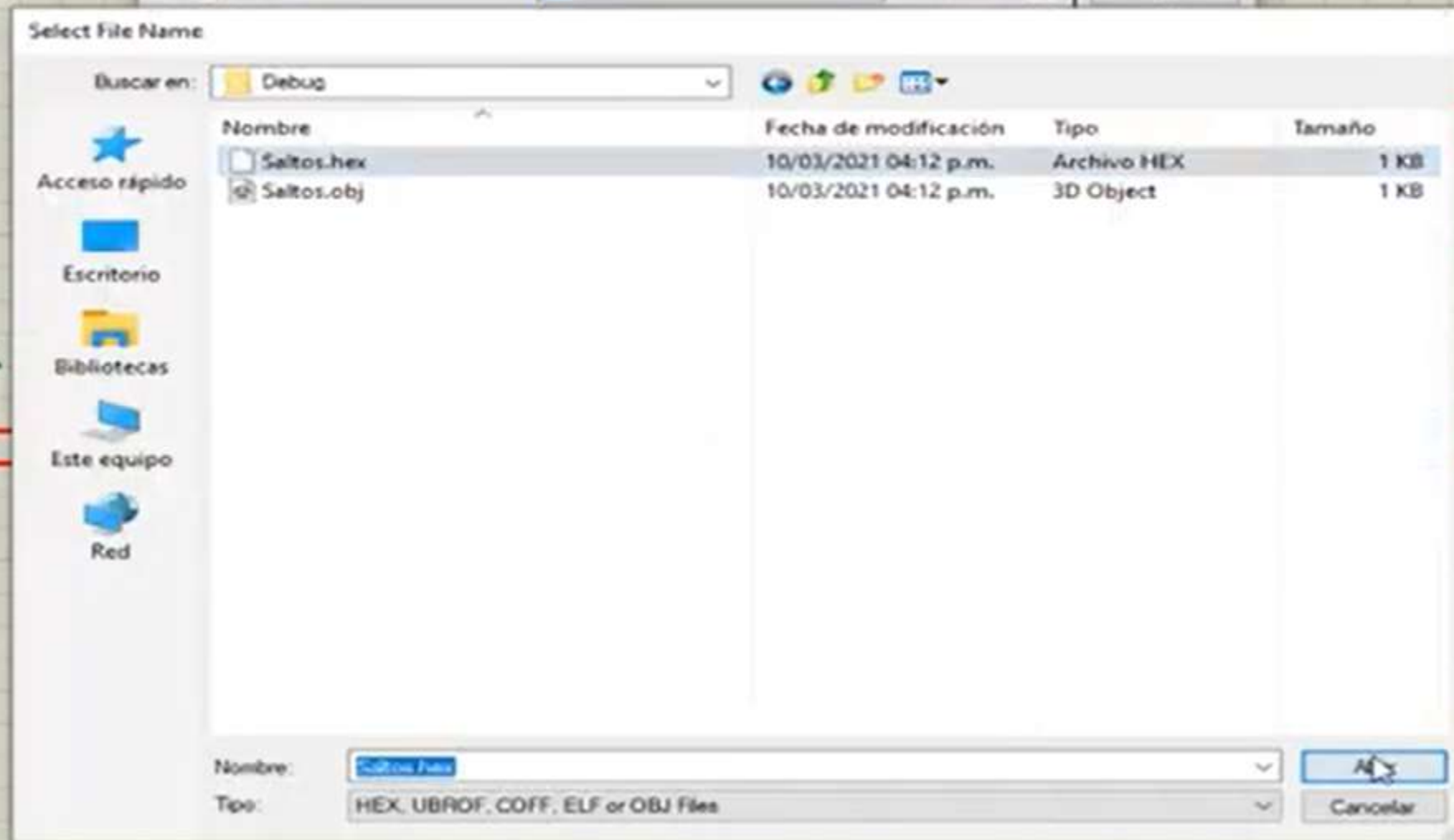
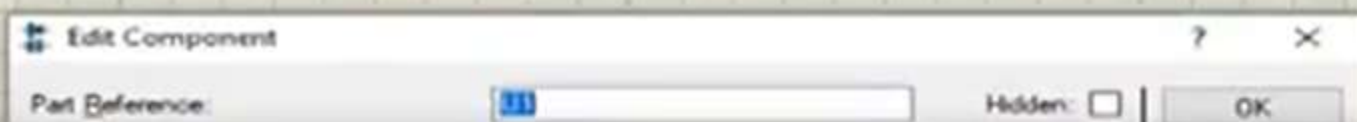
Done building project "Saltos.asmproj".

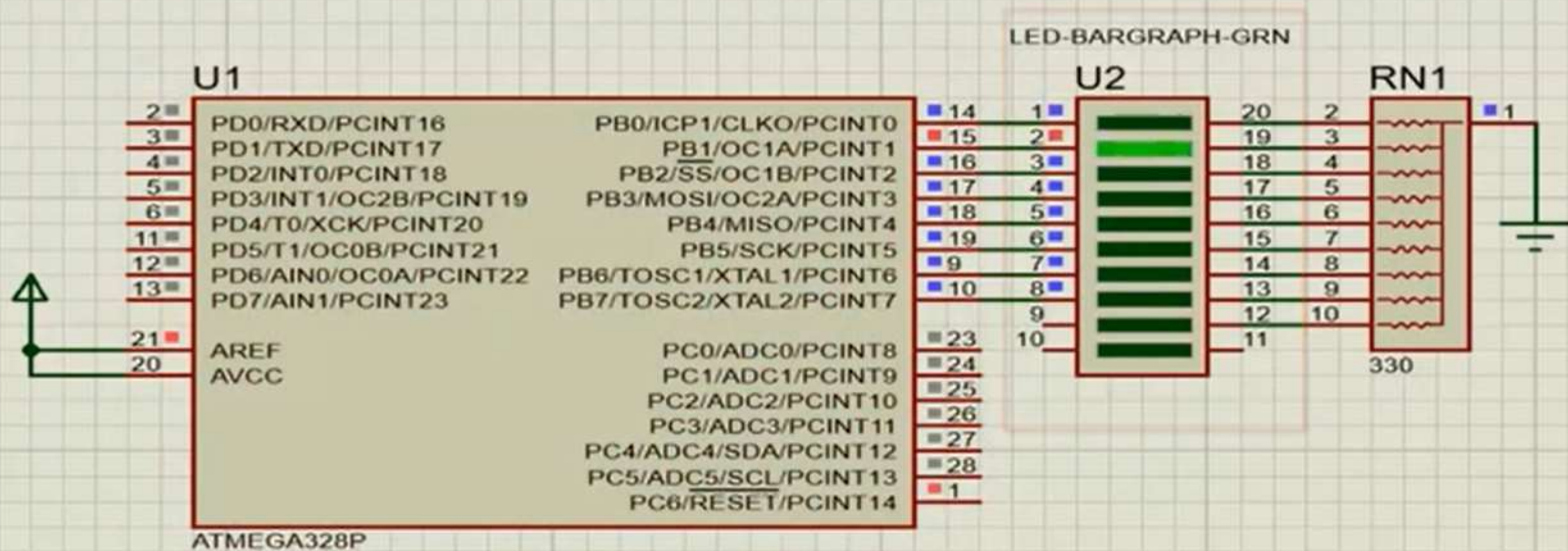
Build succeeded

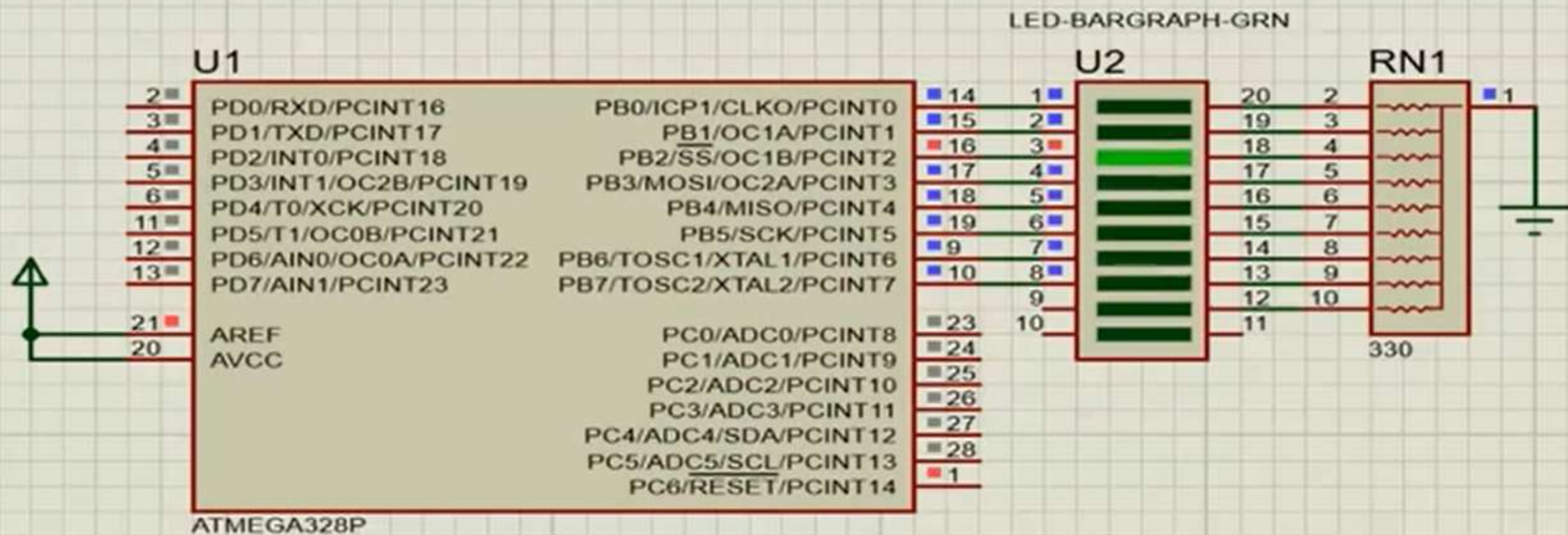
Build: 1 succeeded or up-to-date, 0 failed, 0 skipped

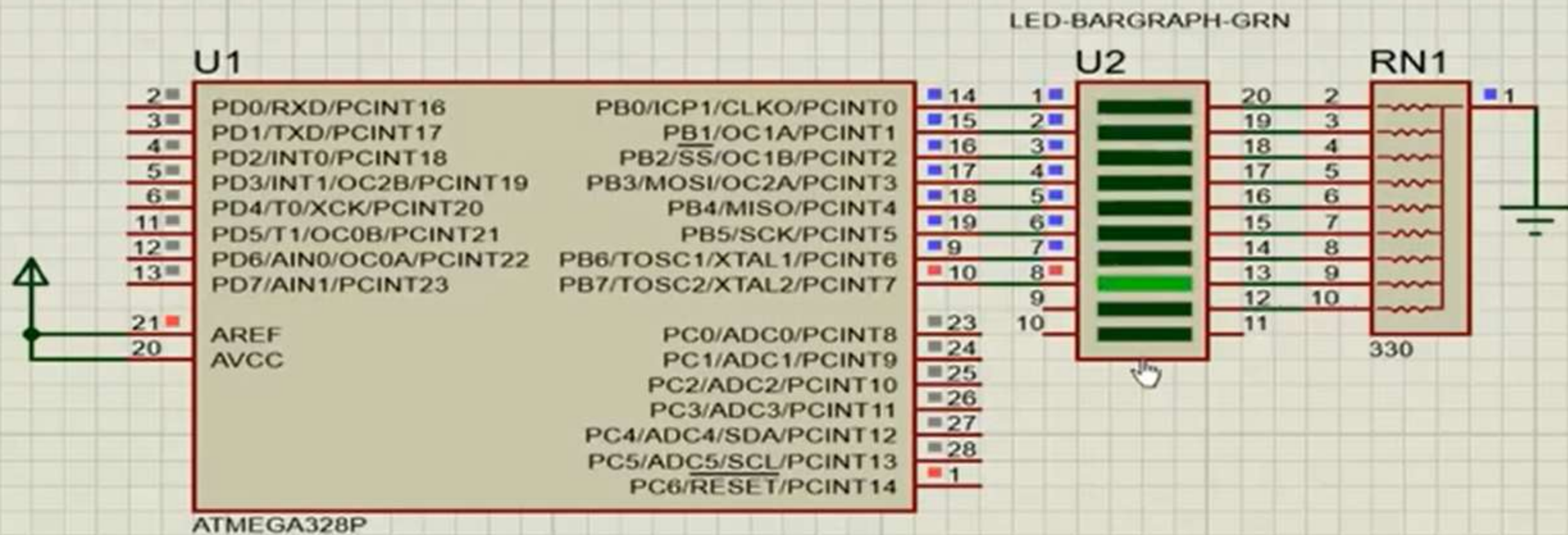
Compilar





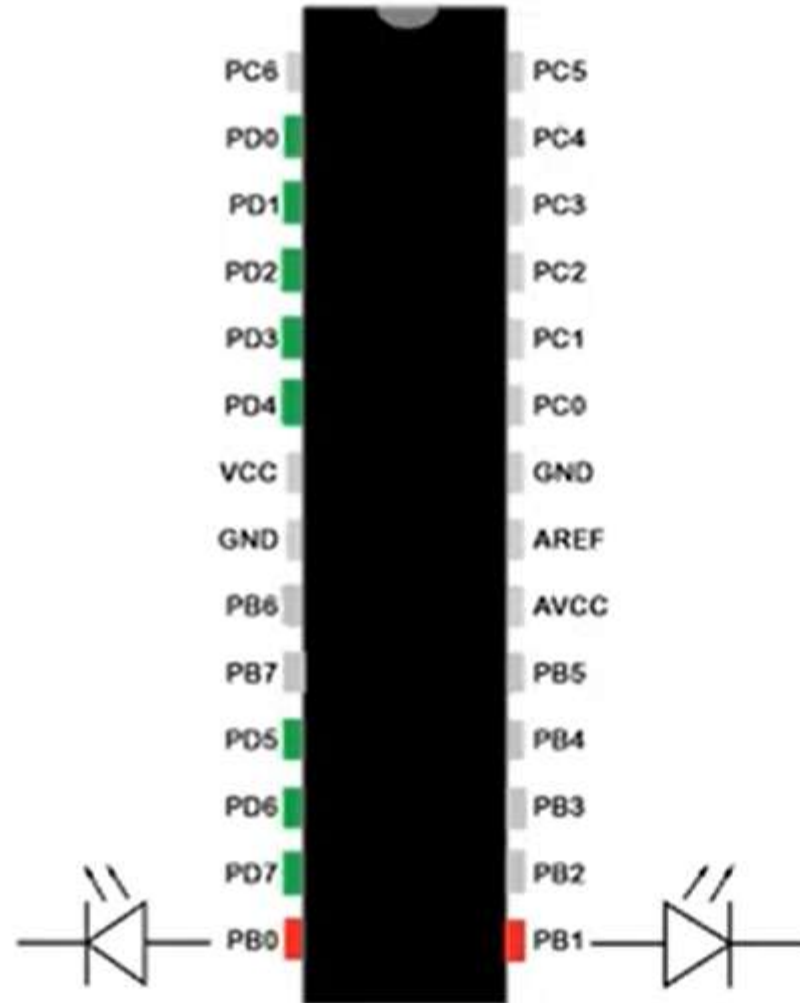






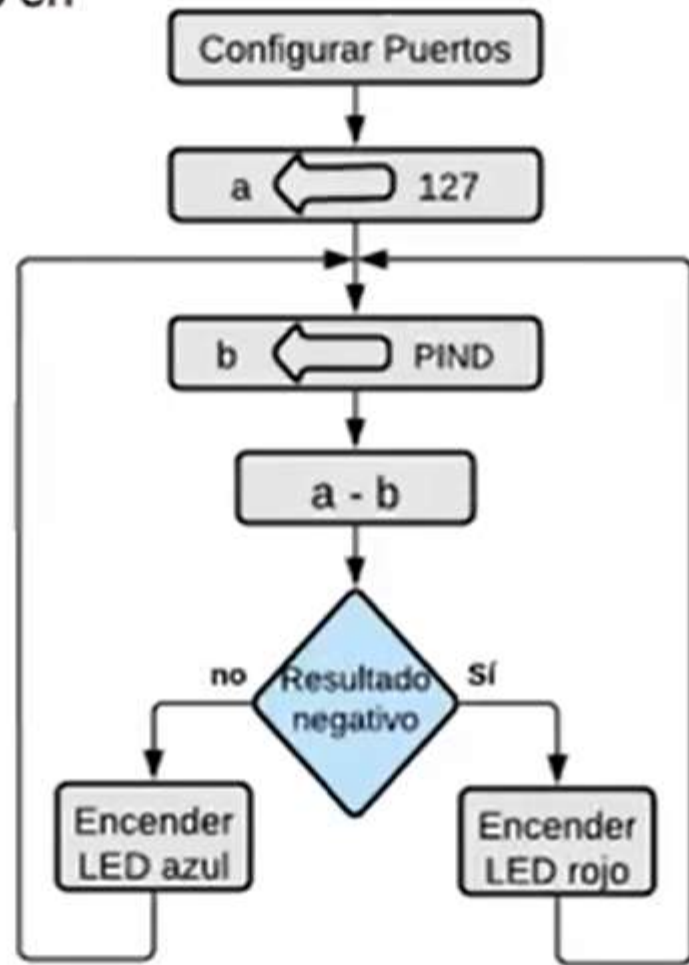
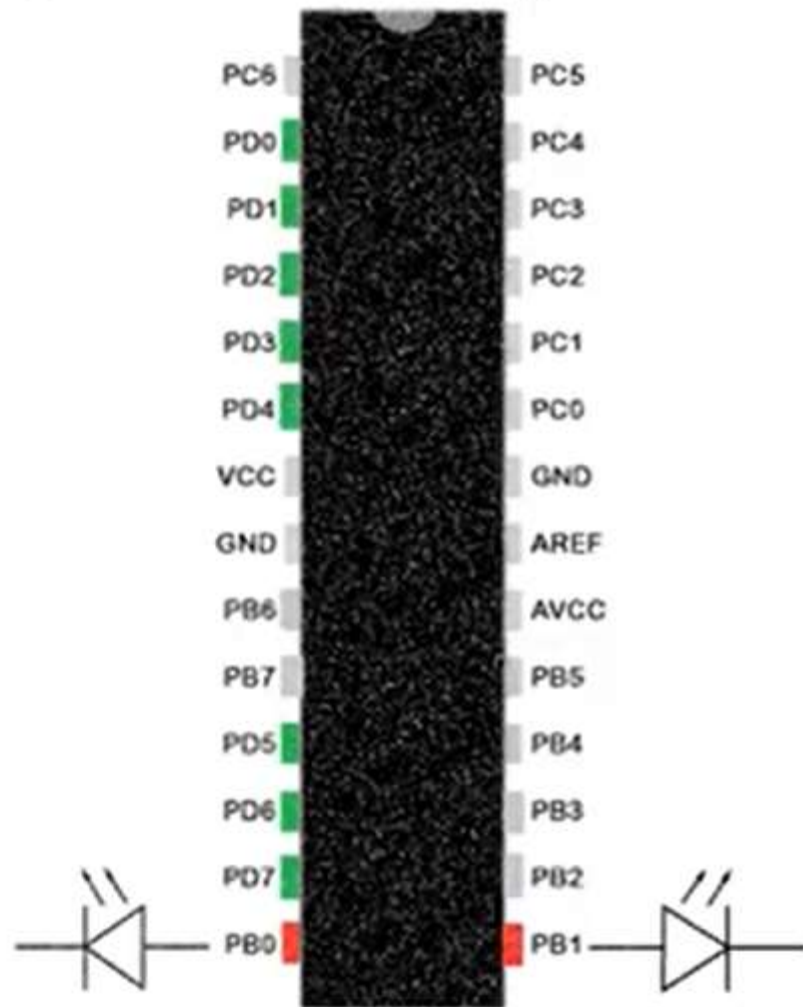
Ejercicio:

Diseñar un programa que realice la resta de dos valores. El primer valor será la constante 127, el segundo valor será dado por la lectura de pines en PORTD.



Ejercicio:

Diseñar un programa que realice la resta de dos valores. El primer valor será la constante 127, el segundo valor será dado por la lectura de pines en PORTD.





¡Muchas gracias!