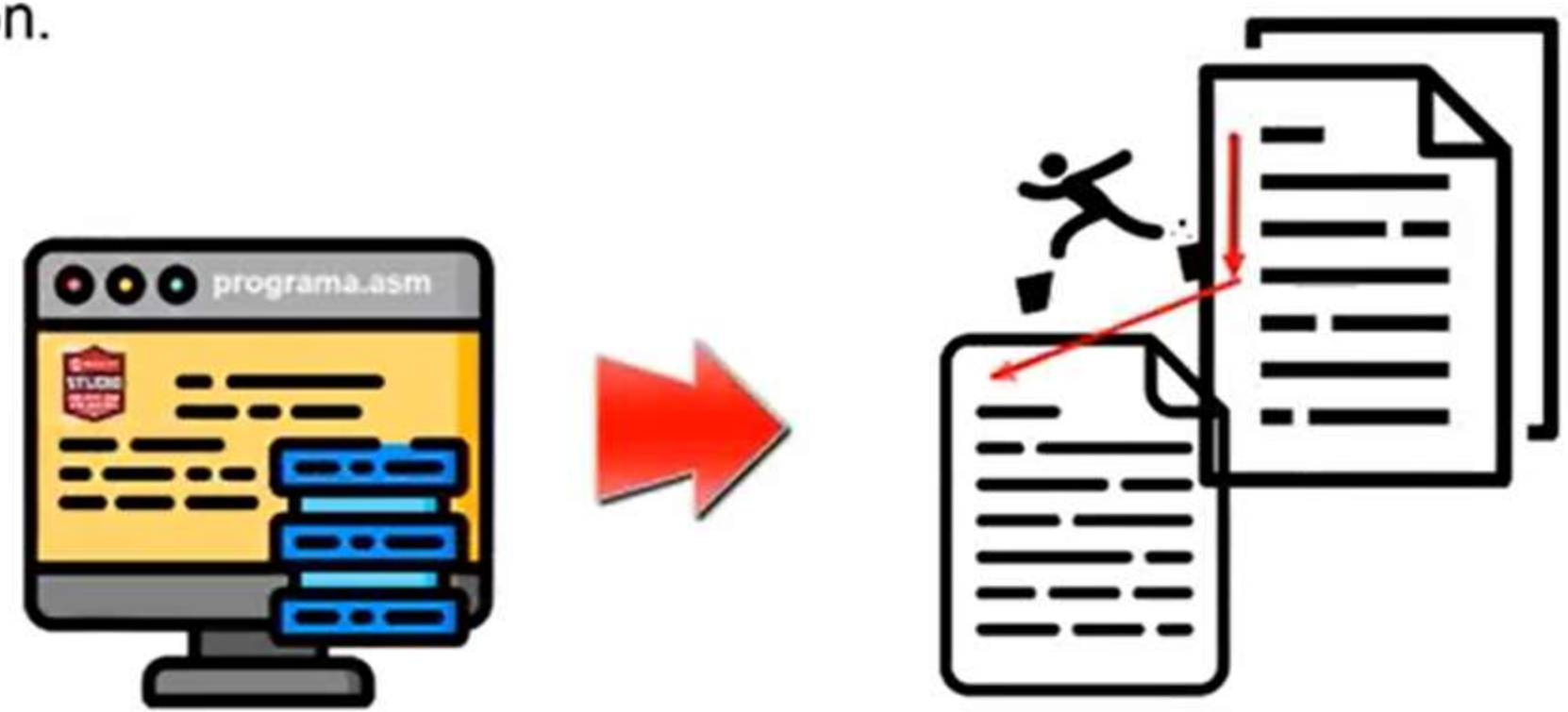
Instituto Tecnológico de Zitácuaro

Tema 4:
Saltos Condicionales

Salto condicional

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



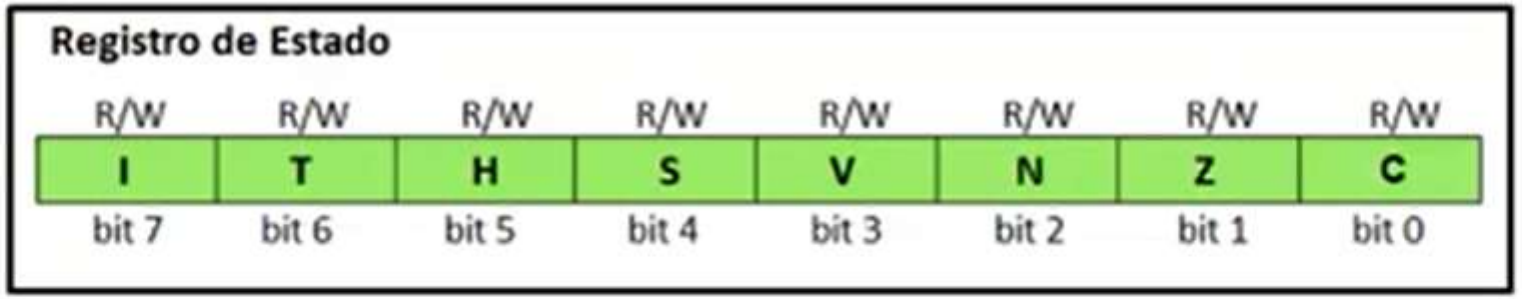
ATmega328p

Instrucciones ensamblador

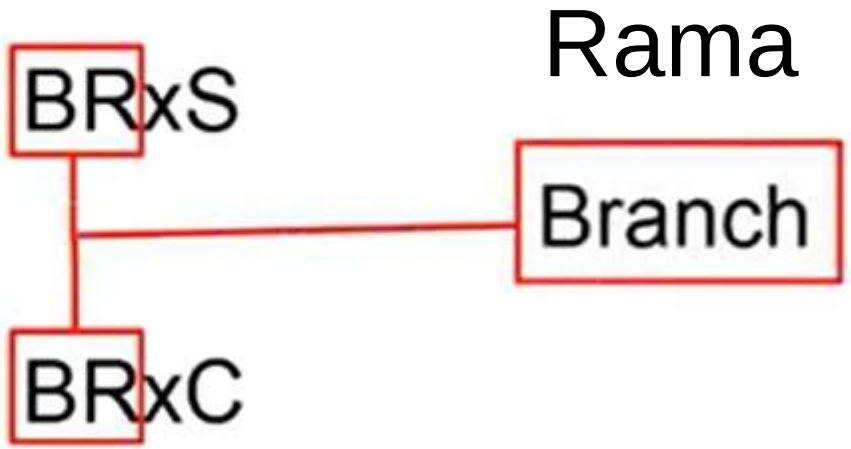
Saltos

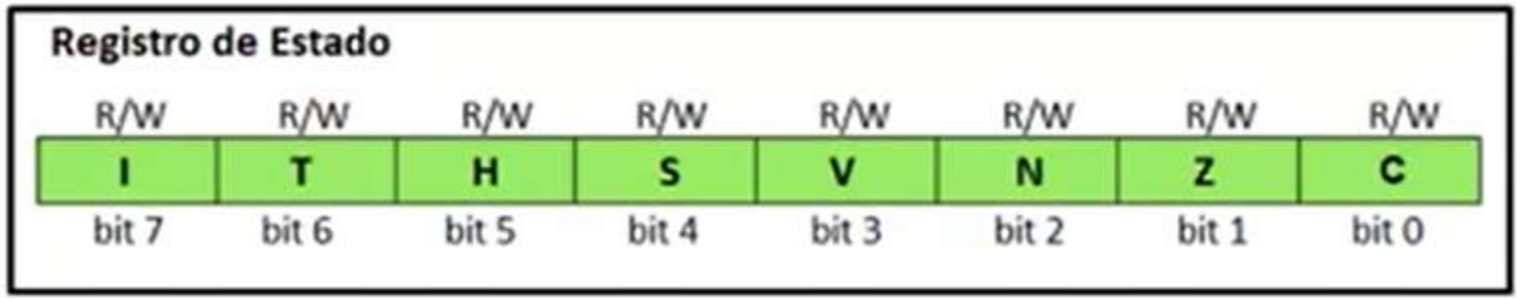
Registro de Estado

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:

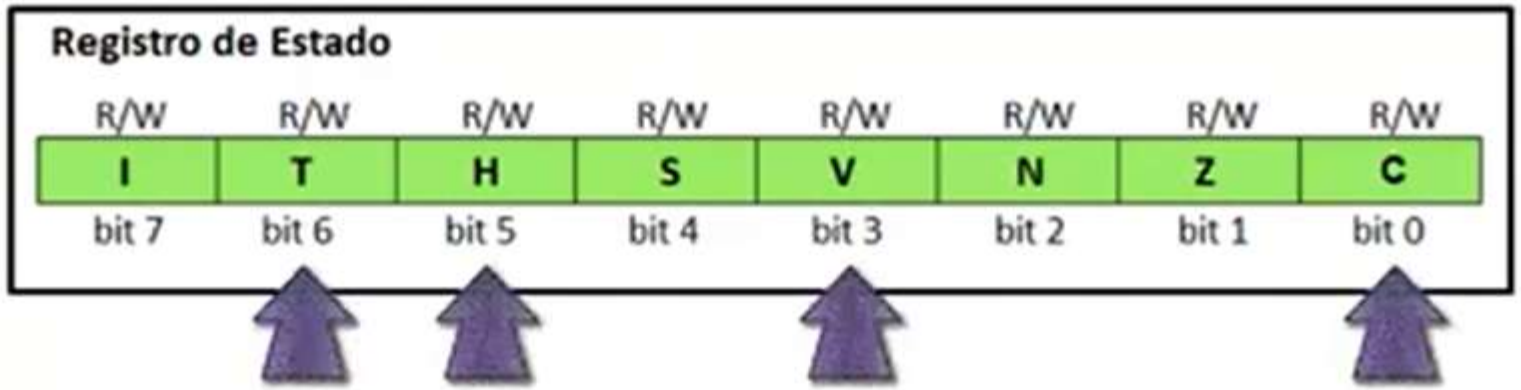

Rama
BRxS
Branch
BRxC



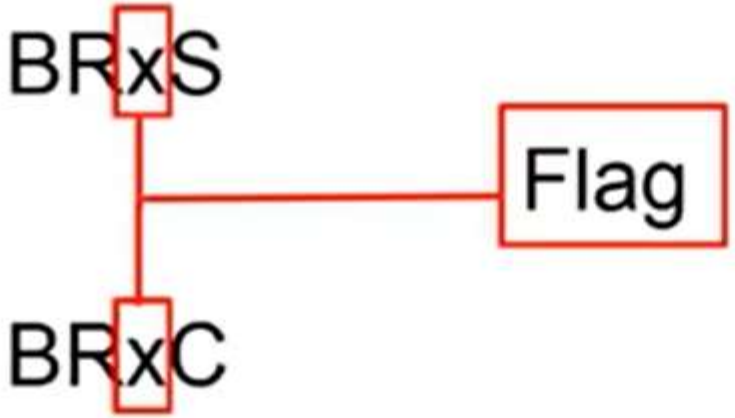
Mnemónicos:

BRxS — Set — 1

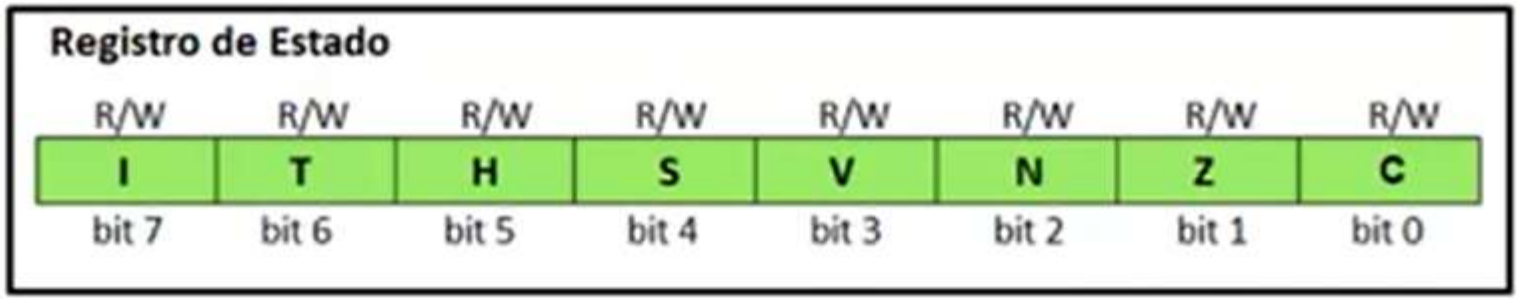
BRxC — Clear — 0



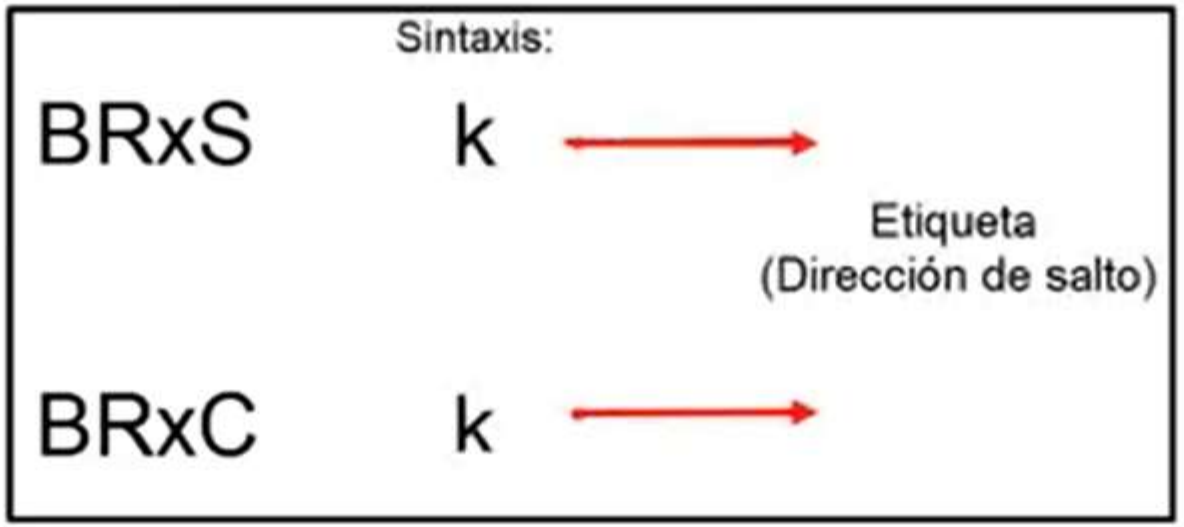
Mnemónicos:



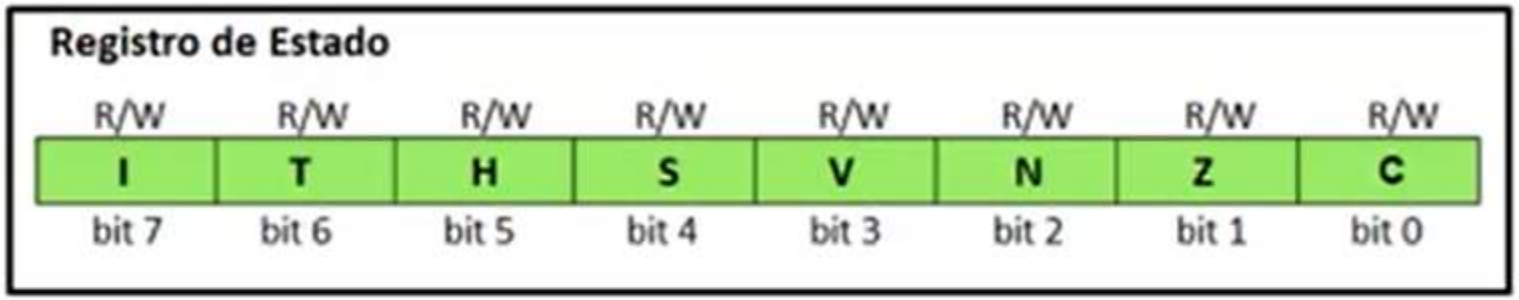
Uso de Banderas



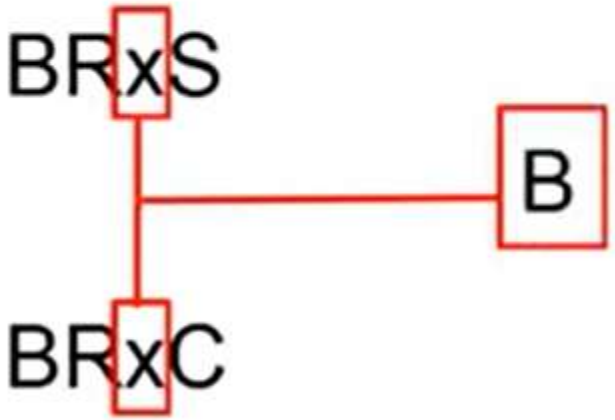
Mnemónicos:



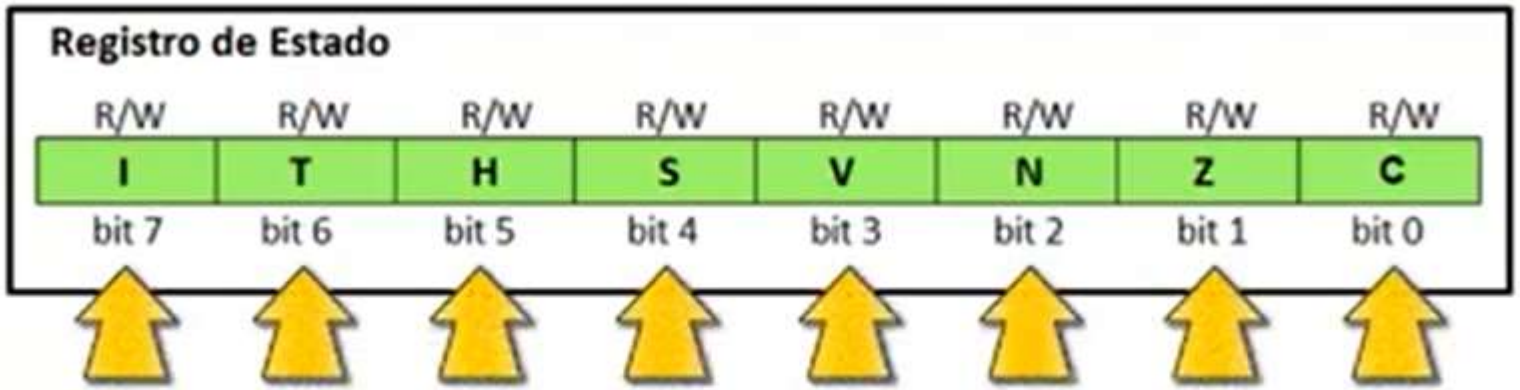
Uso de sintaxis con una etiqueta



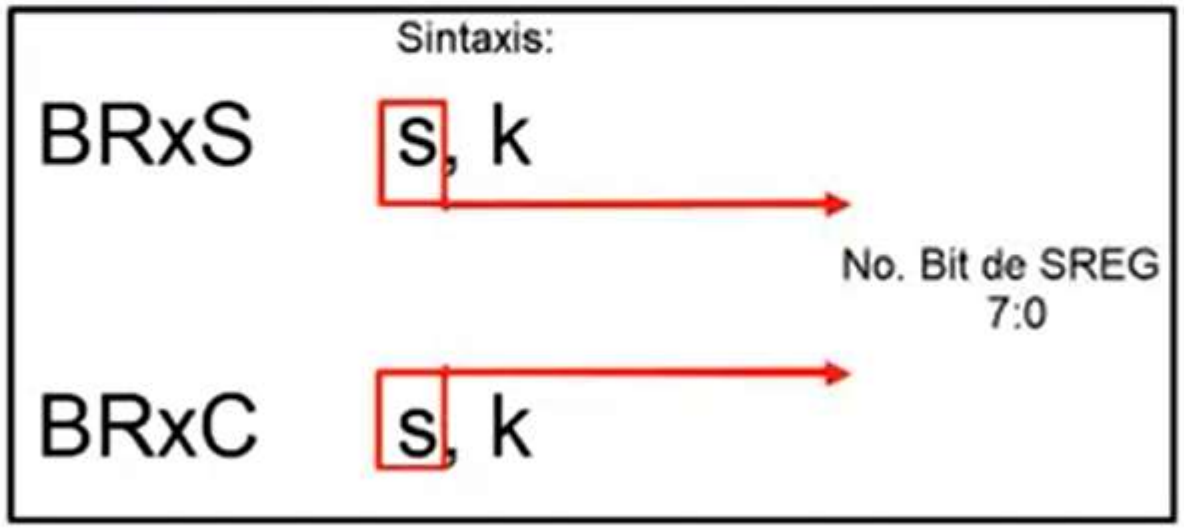
Mnemónicos:

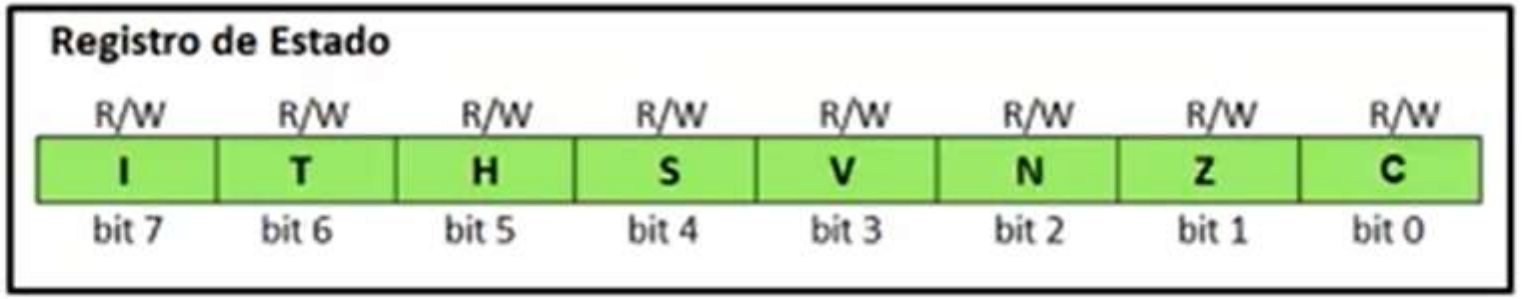


Para hacer uso de cualquier Registro de Estado se coloca la letra B


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:


Sintaxis:
BRxS s, k
No. Bit de SREG
7:0
BRxC s, k



Mnemónicos:

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

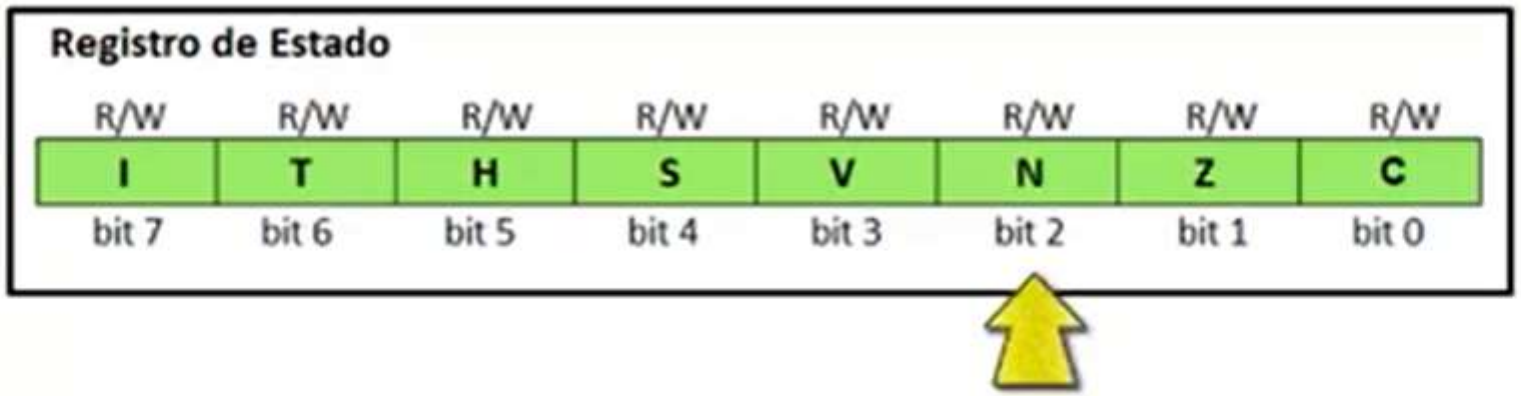
Mnemónicos:

BREQ → 1

z

a - b = 0

a = b



Mnemónicos:

BRMI — Minus (Negativo)

1

N

BRPL — Plus (Positivo)

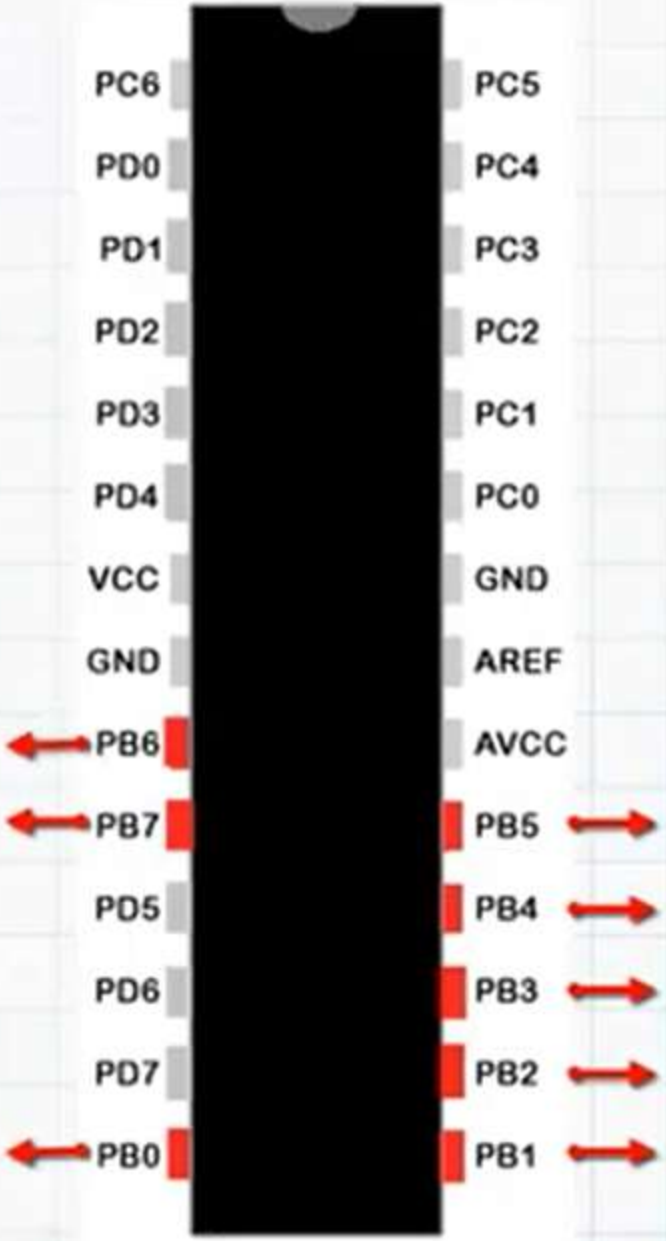
0

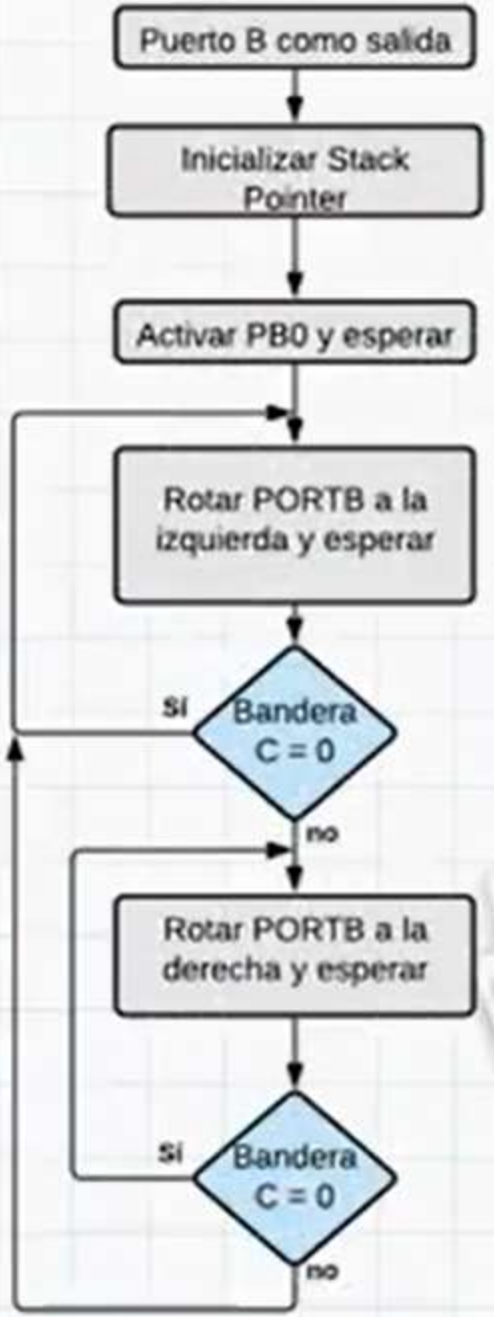
N

Ejemplo de aplicación

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

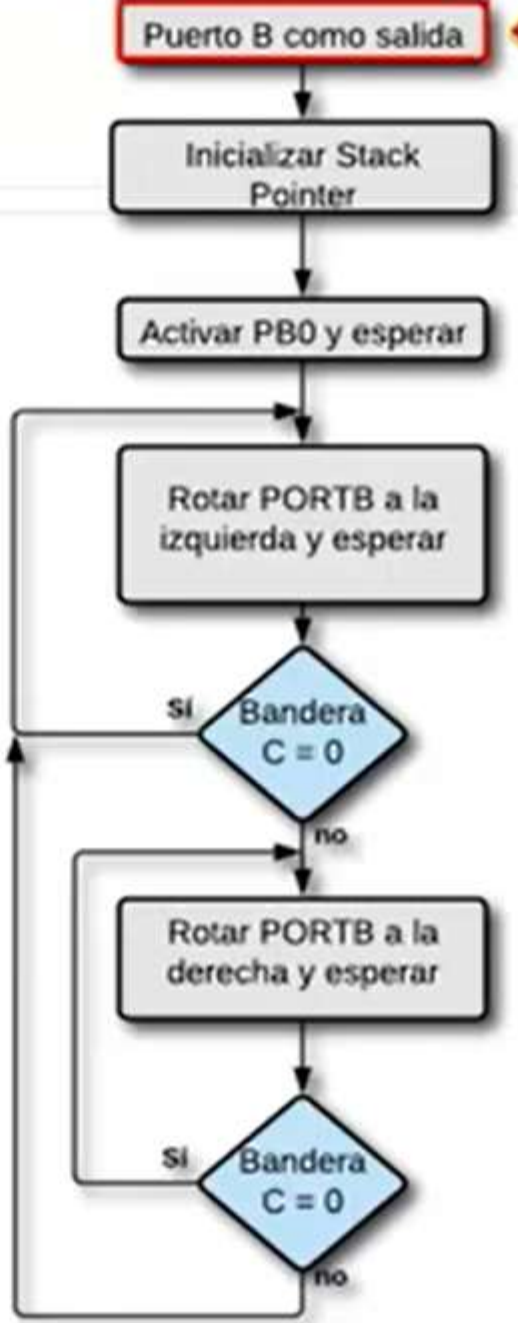
Barra de LED



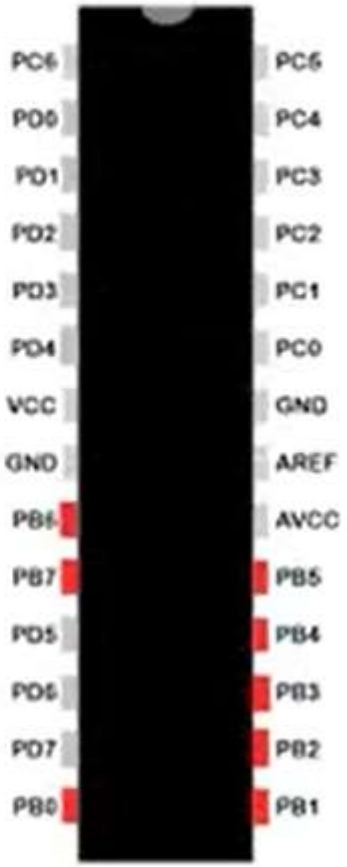


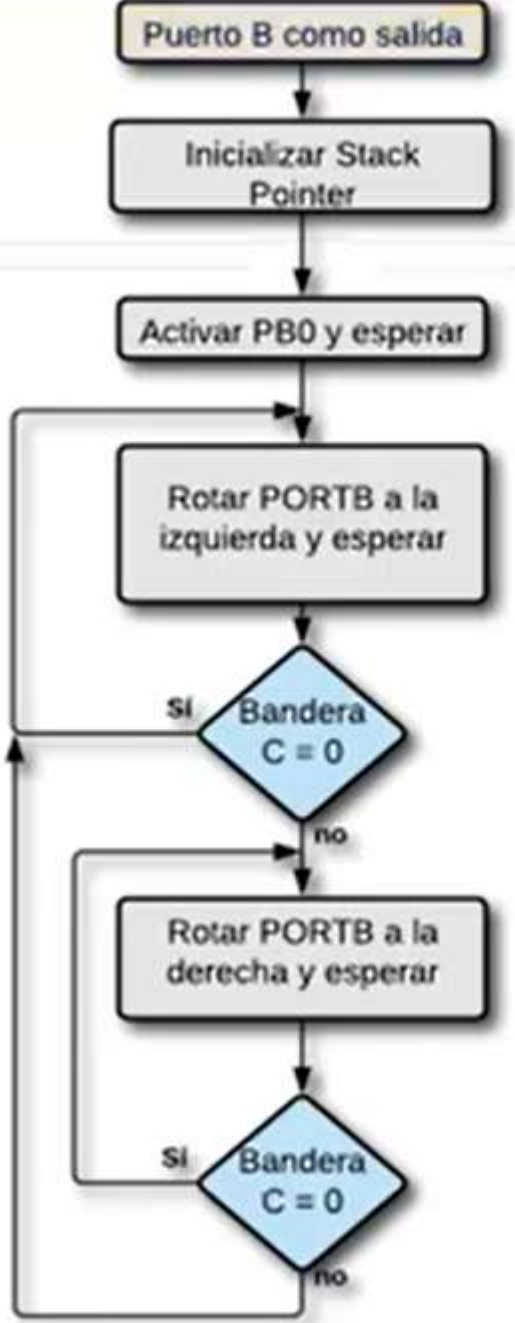
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;
```

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



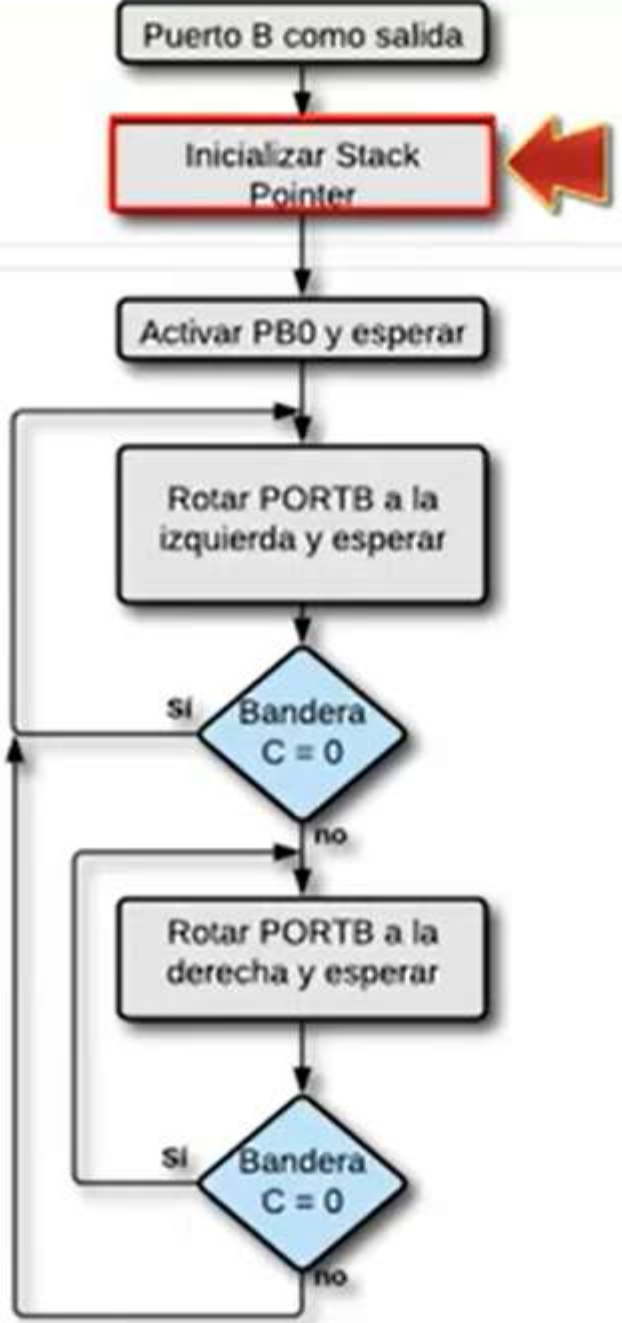
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, R16    ; Configura Puerto B como salida.
```





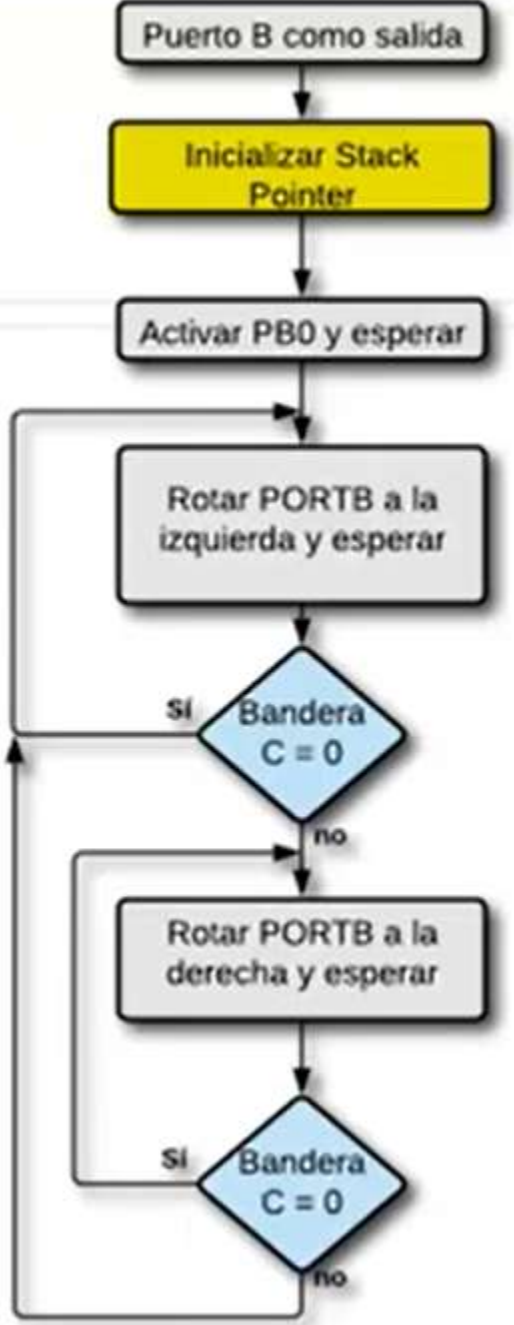
200 %

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

        LDI     R16, 0XFF
        OUT     DDRB, R16    ; Configura Puerto B como salida.
```

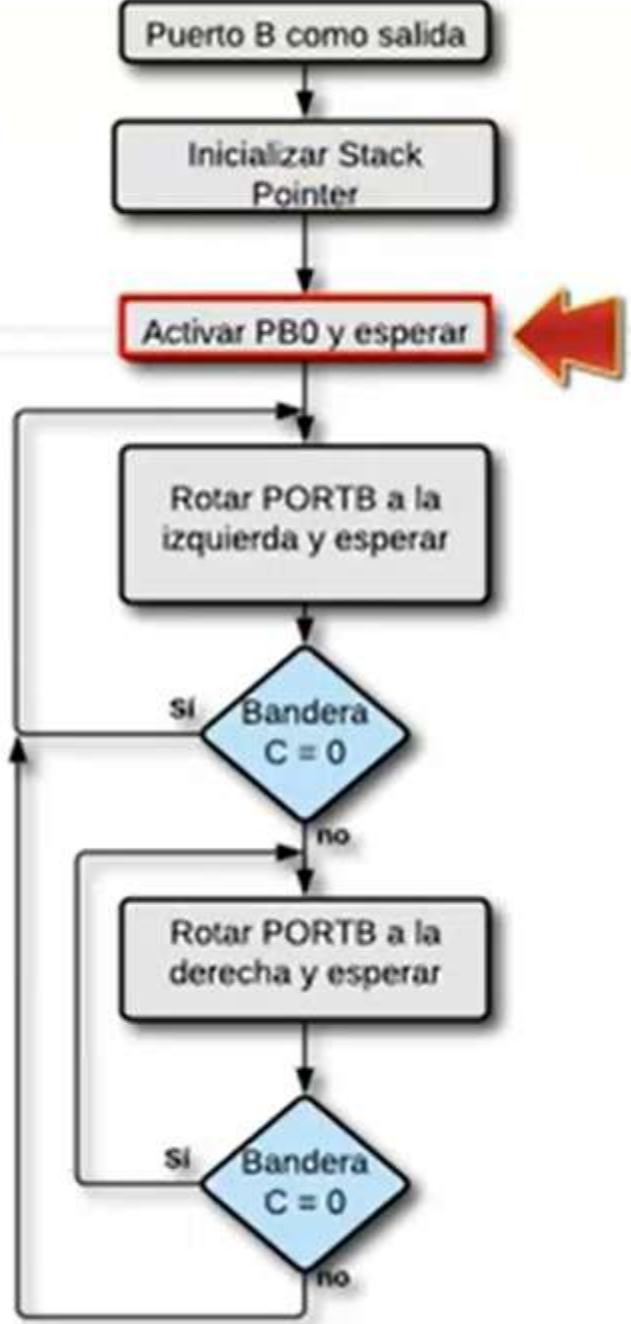


```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, R16    ; Configura Puerto B como salida.
        LDI     R17, 0X08
        OUT     SPL, R16
        OUT     SPH, R17
```

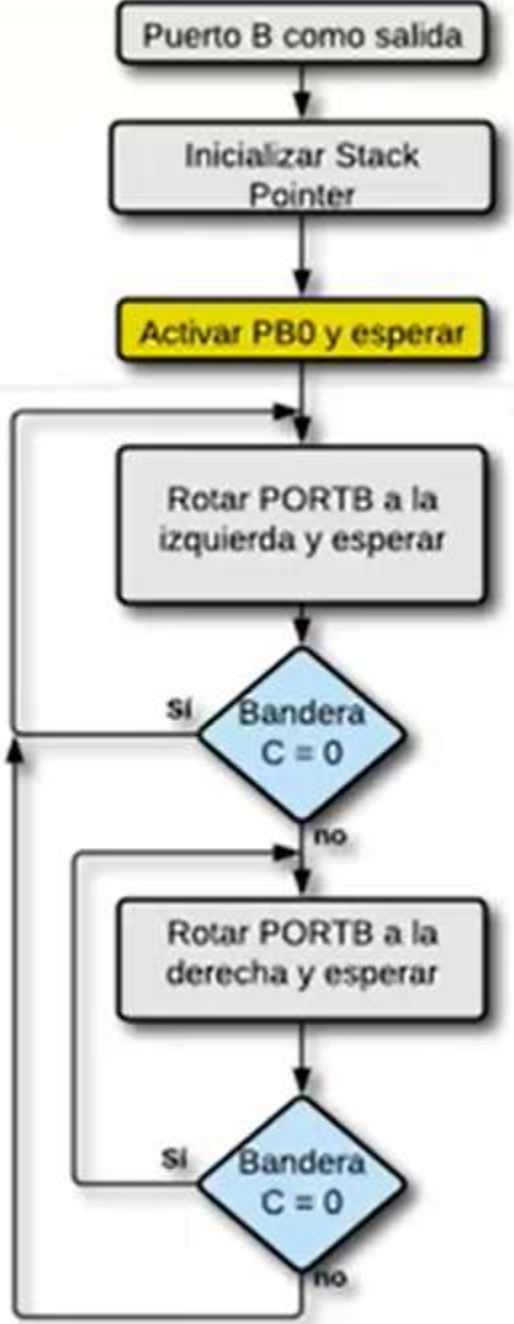
0X08	0XFF
SPH	SPL



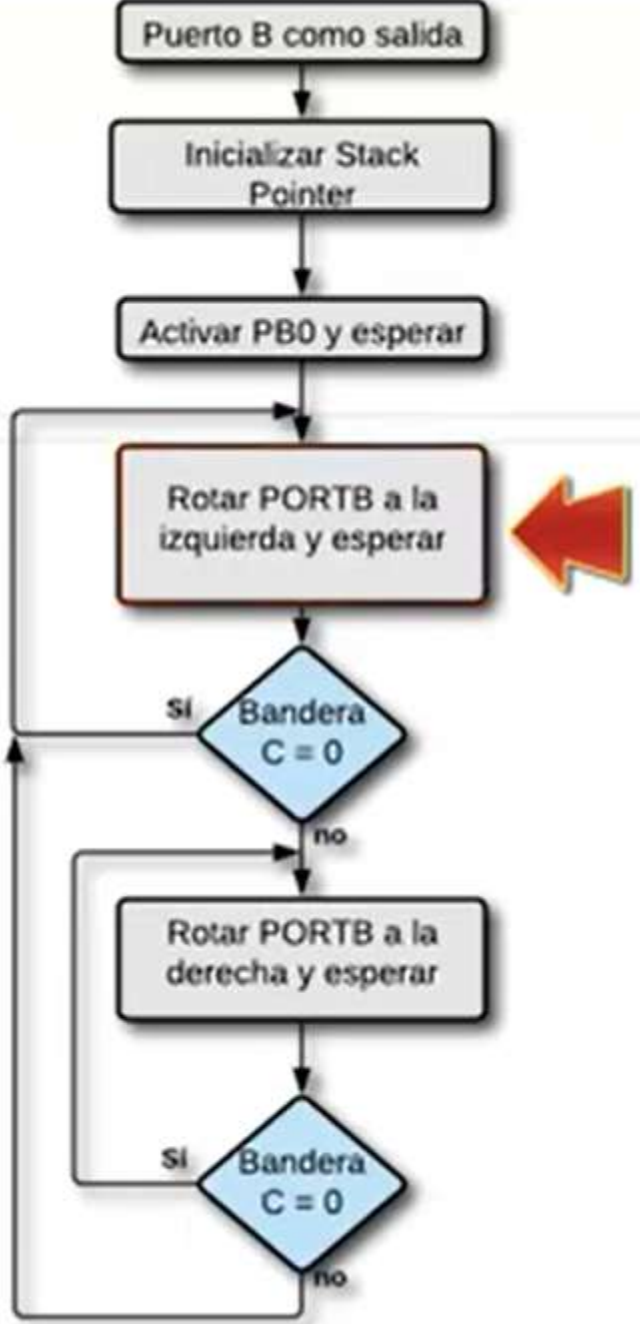
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, R16    ; Configura Puerto B como salida.
        LDI     R17, 0X08
        OUT     SPL, R16
        OUT     SPH, R17     ; Inicializa Stack Pointer.
```



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

        LDI     R16, 0XFF
        OUT     DDRB, 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.
```



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

        LDI     R16, 0xFF
        OUT     DDRB, 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
```

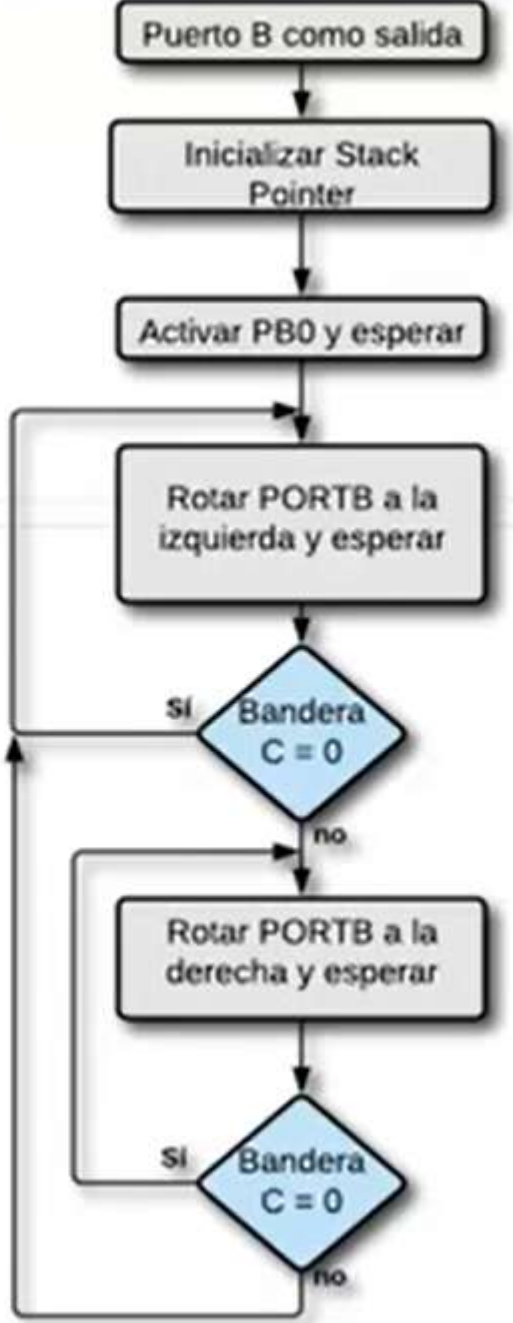


```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, 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
Izq:    ROL
```

Mnemónico	Operandos	Descripción
ROL	Rd	Rotación hacia la izquierda con acarreo

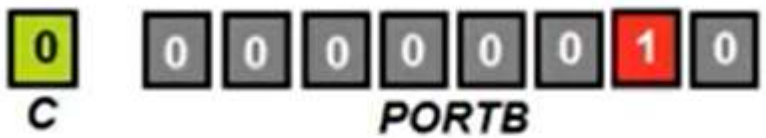
200 %

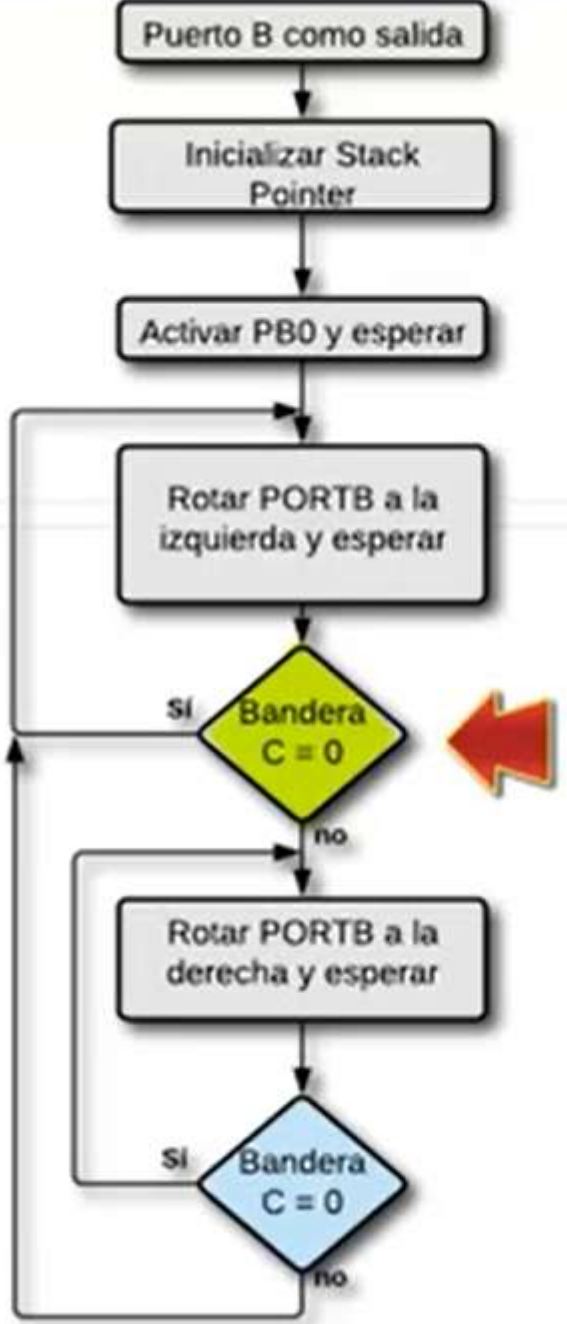
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
Izq:    ROL     R18
        OUT     PORTB, R18   ; Desplaza el bit activo hacia la izquierda.
        RCALL   Wait
```



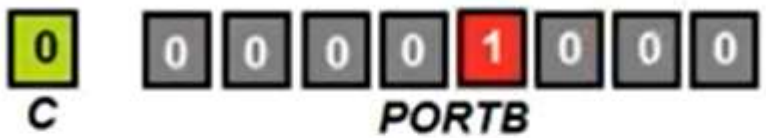
200 %

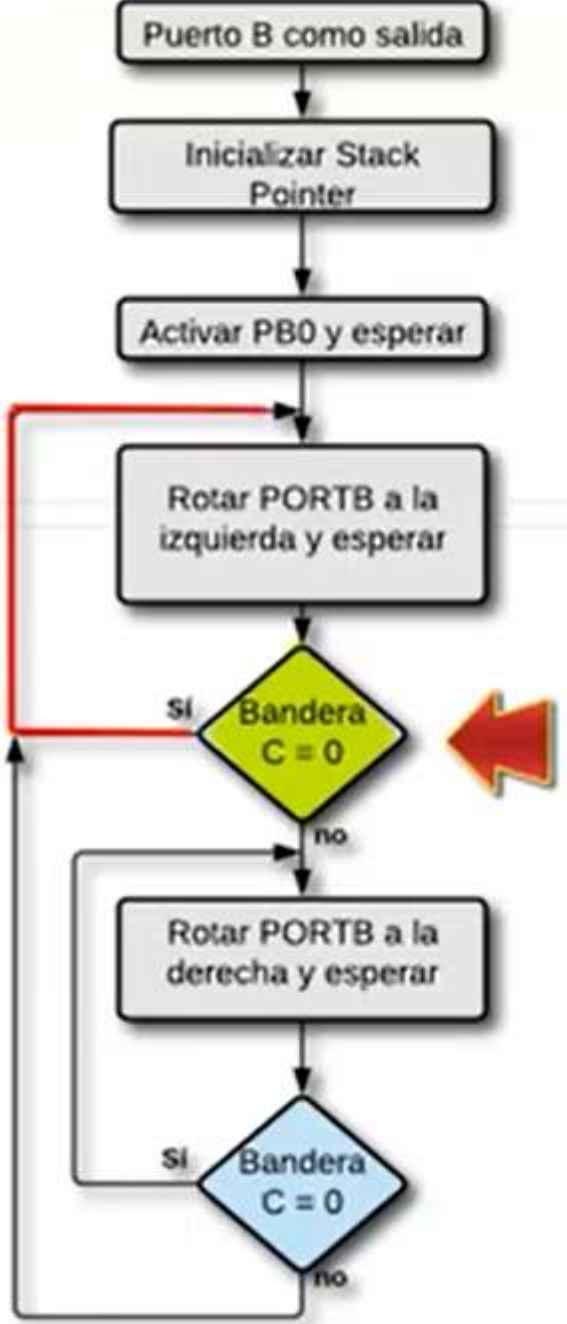
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
Izq:    ROL     R18
        OUT     PORTB, R18   ; Desplaza el bit activo hacia la izquierda.
        RCALL   Wait
        BRCC    Izq
```





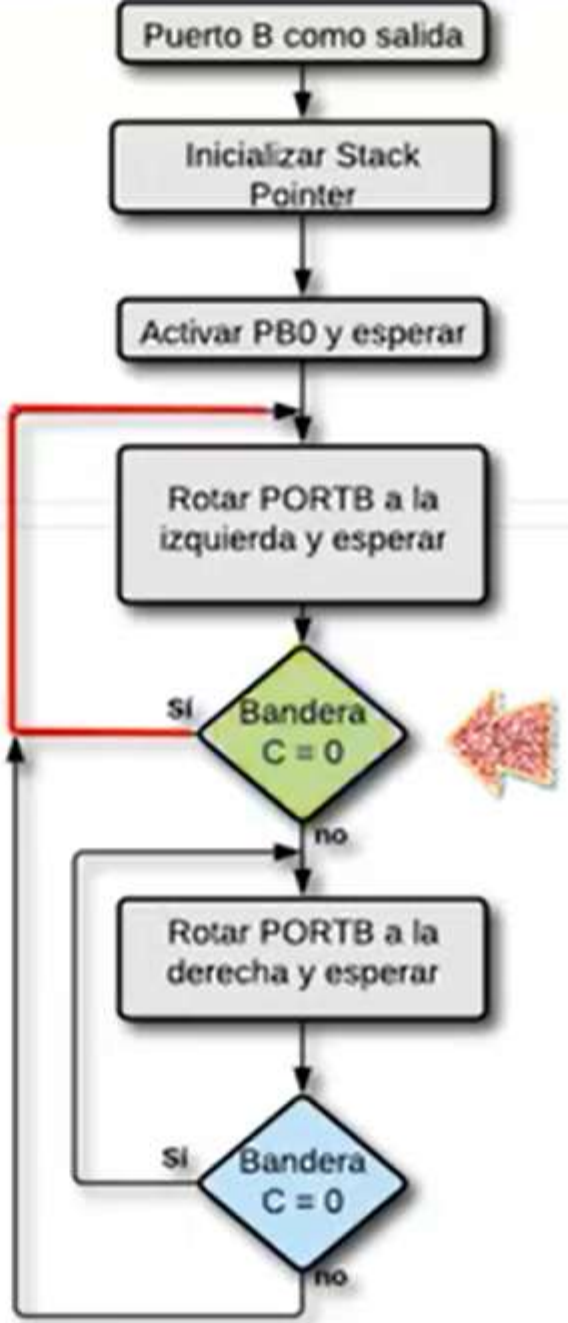
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
Izq:    ROL     R18
        OUT     PORTB, R18   ; Desplaza el bit activo hacia la izquierda.
        RCALL   Wait
        BRCC    Izq
```



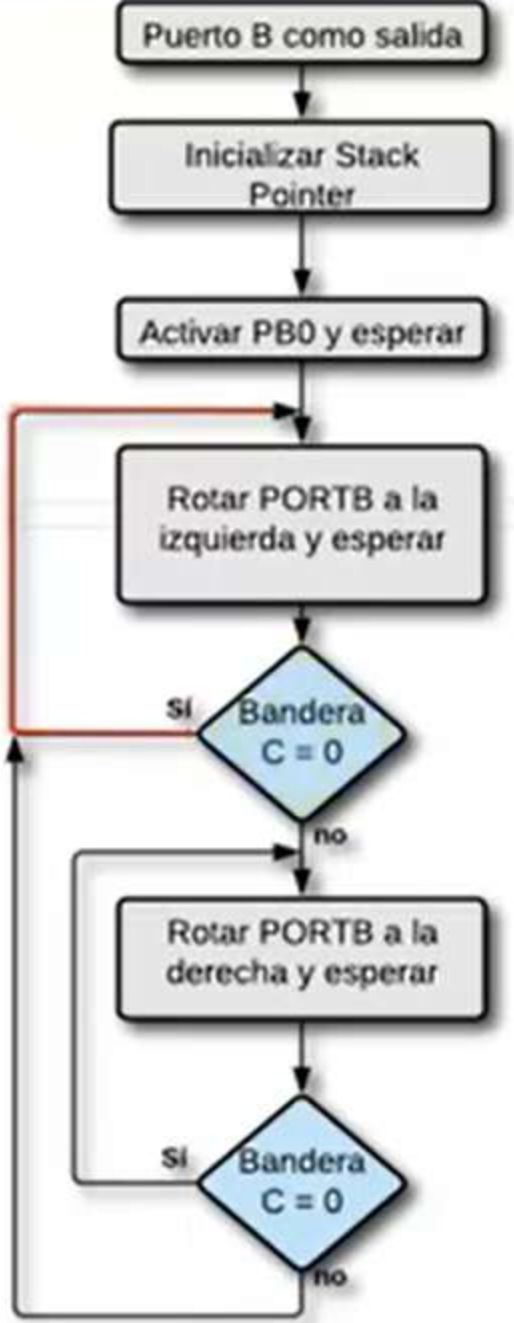


200 %

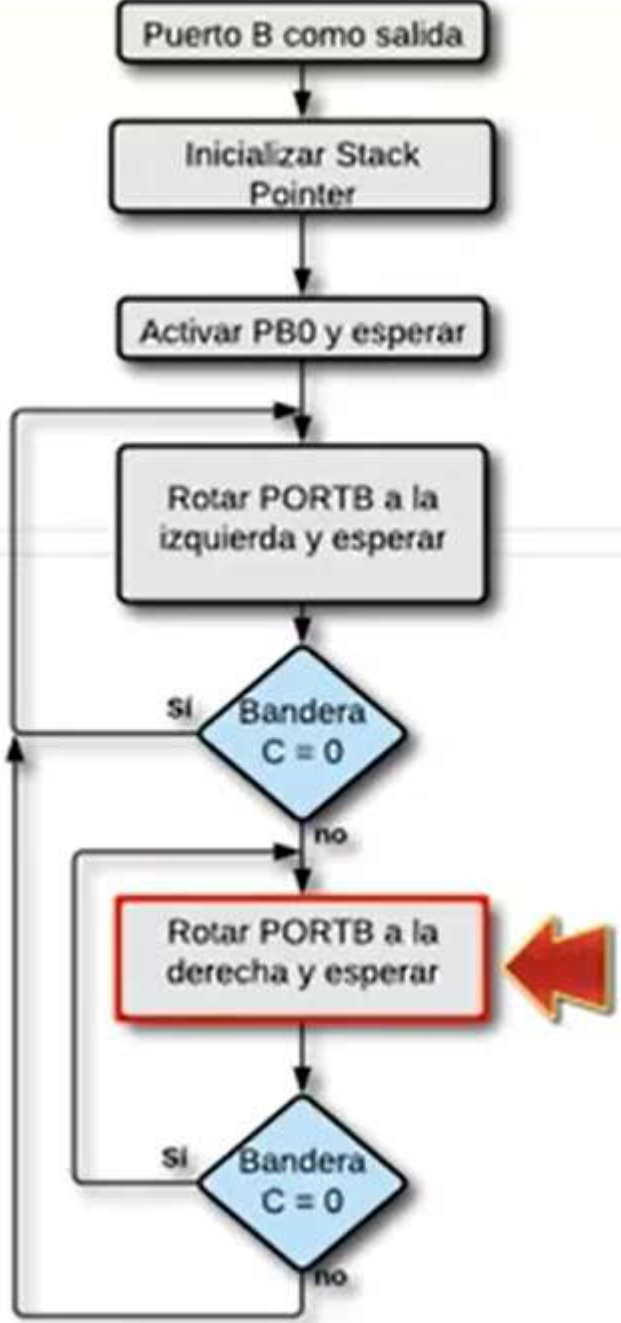
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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 : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, 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
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 : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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.
```

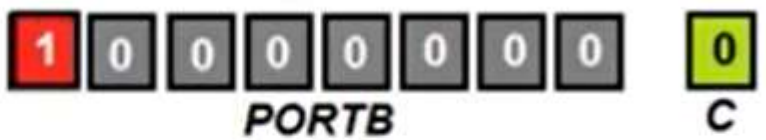


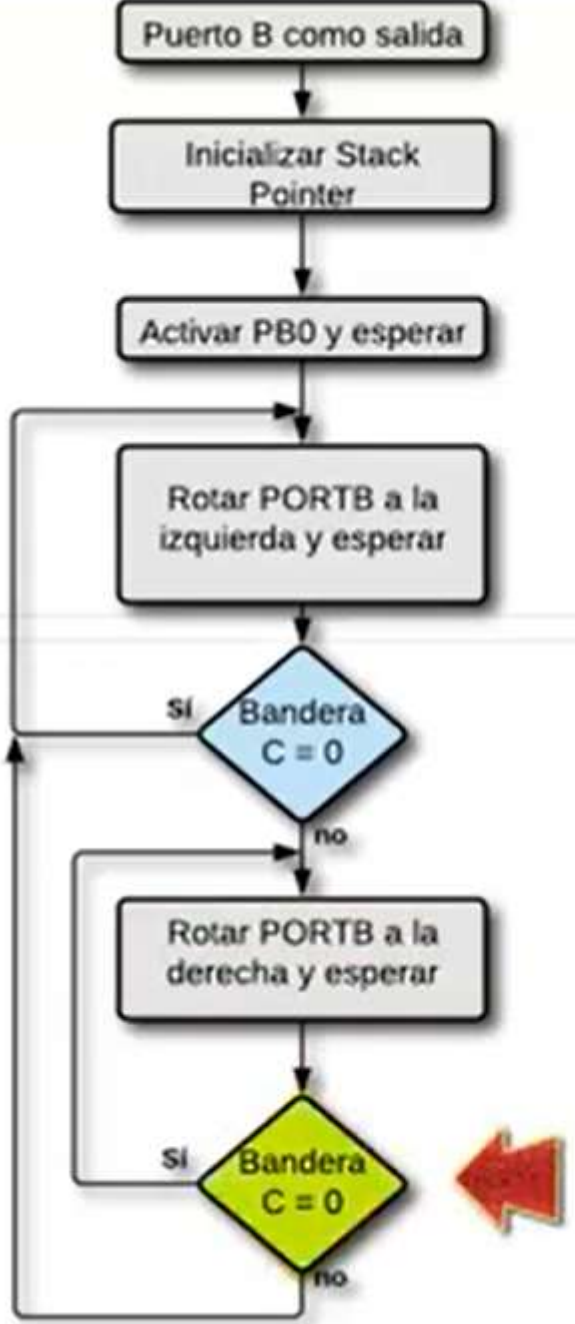
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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:    RO
```

Mnemónico	Operandos	Descripción
ROR	Rd	Rotación hacia la derecha con acarreo

200 %

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

        LDI     R16, 0xFF
        OUT     DDRB, 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
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
```





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

        LDI     R16, 0xFF
        OUT     DDRB, 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
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
```

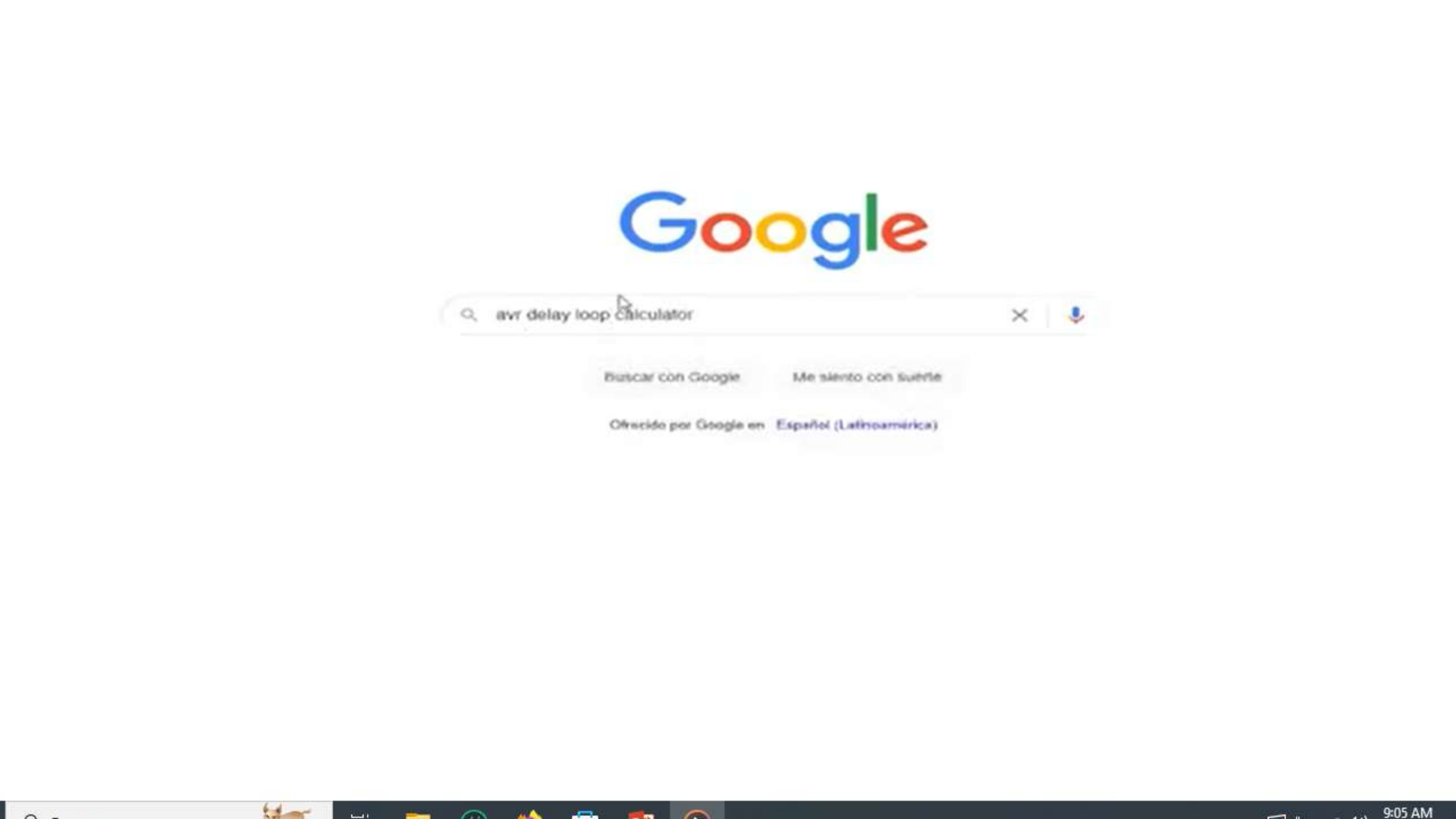
```
;
; Saltos.asm
;
; Created: 08/03/2021 04:06:43 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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 : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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 : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, 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
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
```

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

        LDI     R16, 0XFF
        OUT     DDRB, 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
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


Google
avr delay loop calculator
Buscar con Google
Me siento con suerte
Ofrecido por Google en Español (Latinoamérica)
9:05 AM

avr delay loop calculator

Todo | Imágenes | Shopping | Videos | Noticias | Más | Preferencias | Herramientas

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 ▾ PDF

AVR Looping

500 µs. **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 Mulvey. Register enhancement by T. Morland. (ACES '18)

1.0 MHz microcontroller clock frequency

0 cycles for rcall/ret or other overhead

19 first register to be used by delay loop

0.3 ns us ms s mins hrs days

300000 cycles go

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
```

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)

1.0 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

19 first register to be used by delay loop

0.3 ns us ms s mins hrs days

300000 cycles go

◉ 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, 154
    ldi  r21, 154
L1: dec  r21
    brne L1
    dec  r20
    brne L1
    dec  r19
    brne L1
```

Copiar y pegar

main.asm*

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

        LDI     R16, 0XFF
        OUT     DDRB, R16    ; Configura Puerto B como salida.
        LDI     R17, 0X08
        OUT     SPL, R16
        OUT     SPH, R17     ; Inicializa Stack Pointer.
        LDI     R18, 0X01
        OUT     PORTD, 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
        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
```

100 %

Error List

9:12 AM

main.asm*

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

        LDI     R16, 0XFF
        OUT     DDRB, 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
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
```

Agregar RET

100 %

Error List

9:13 AM

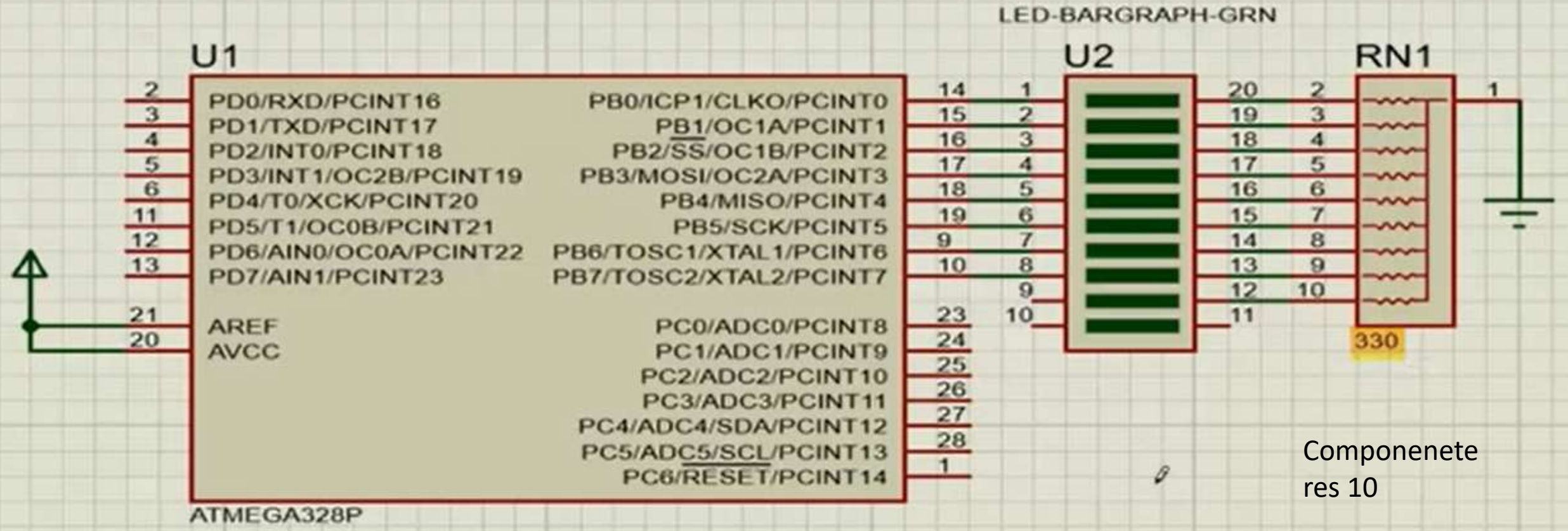
```
        LDI     R17, 0x08
        OUT     SPL, R16
        OUT     SPH, R17    ; Inicializa Stack Pointer.
        LDI     R18, 0x01
        OUT     PORTB, R18  ; Activa PB0.
        RCALL   Wait
Izq:    ROL     R18
        OUT     PORTB, R18  ; Desplaza el bit activo hacia la izquierda.
        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
        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

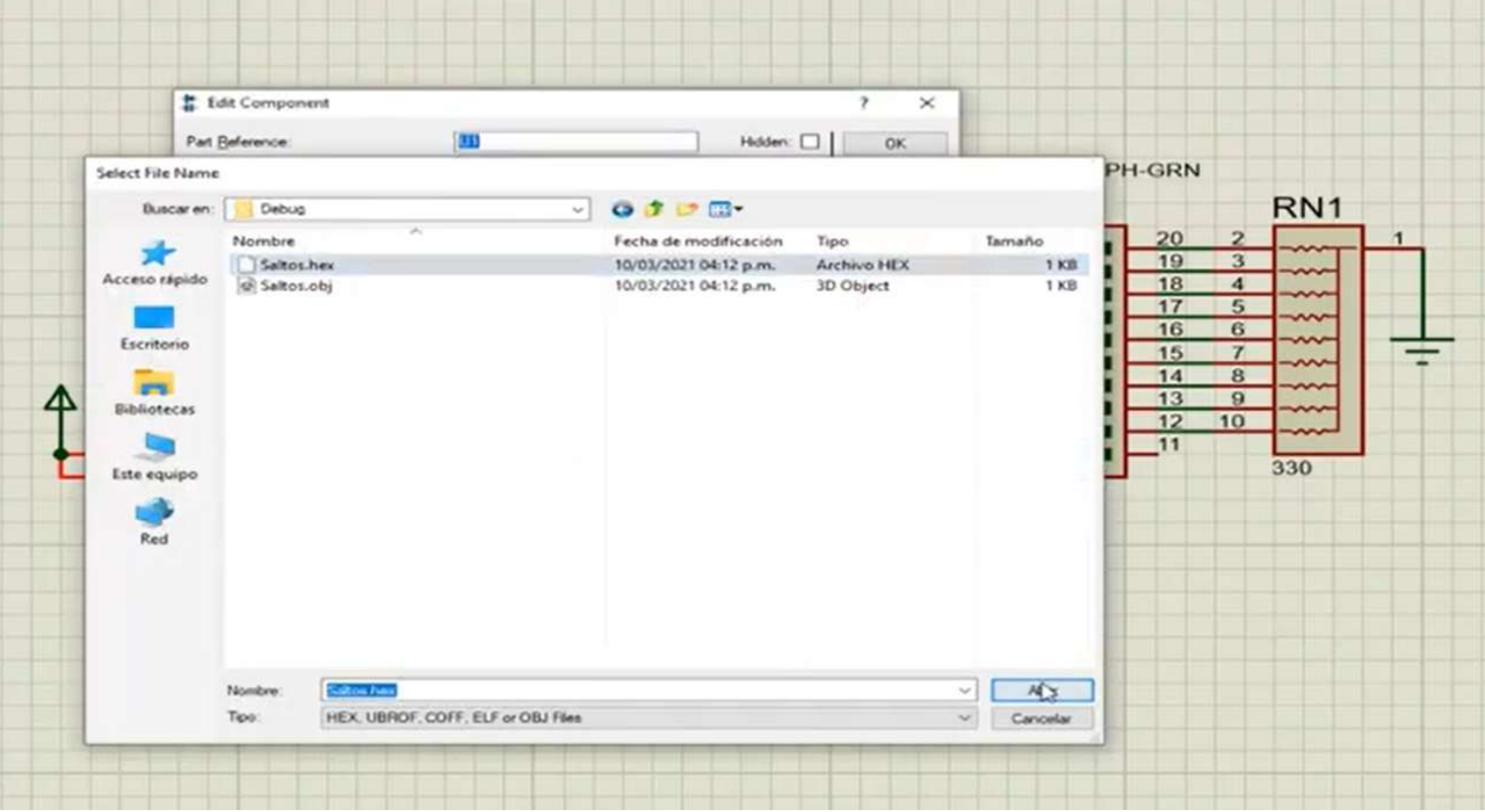
Show output from: Build

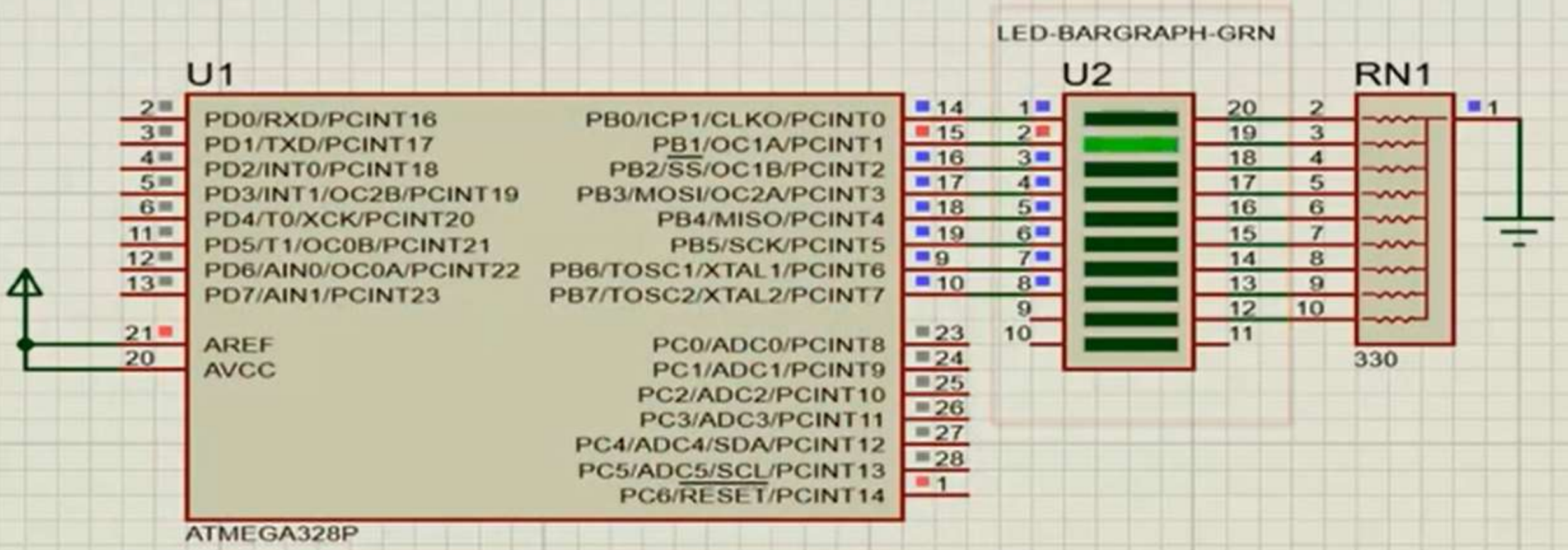
```
        [.cseg] 0x000000 0x000036     54      0     54   32768   0.2%
        [.dseg] 0x000100 0x000100      0      0      0    2048   0.0%
        [.eseg] 0x000000 0x000000      0      0      0    1024   0.0%
        Assembly complete, 0 errors. 0 warnings
    Done executing task "RunAssemblerTask".
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 ==========
```

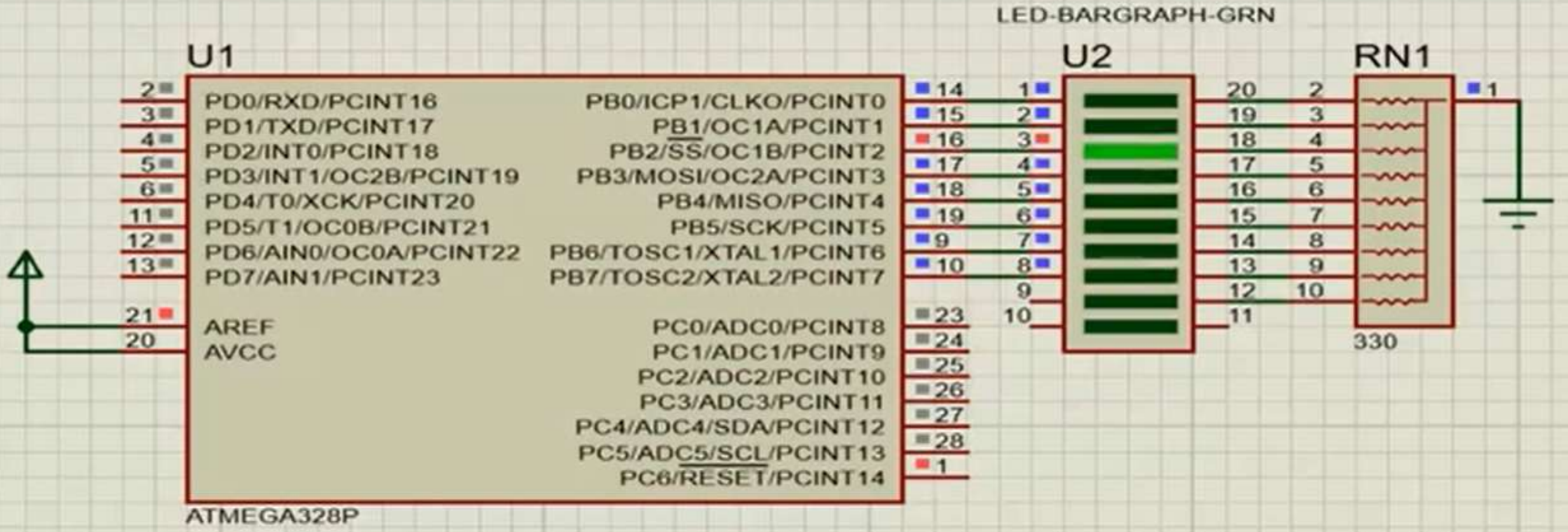
Compilar

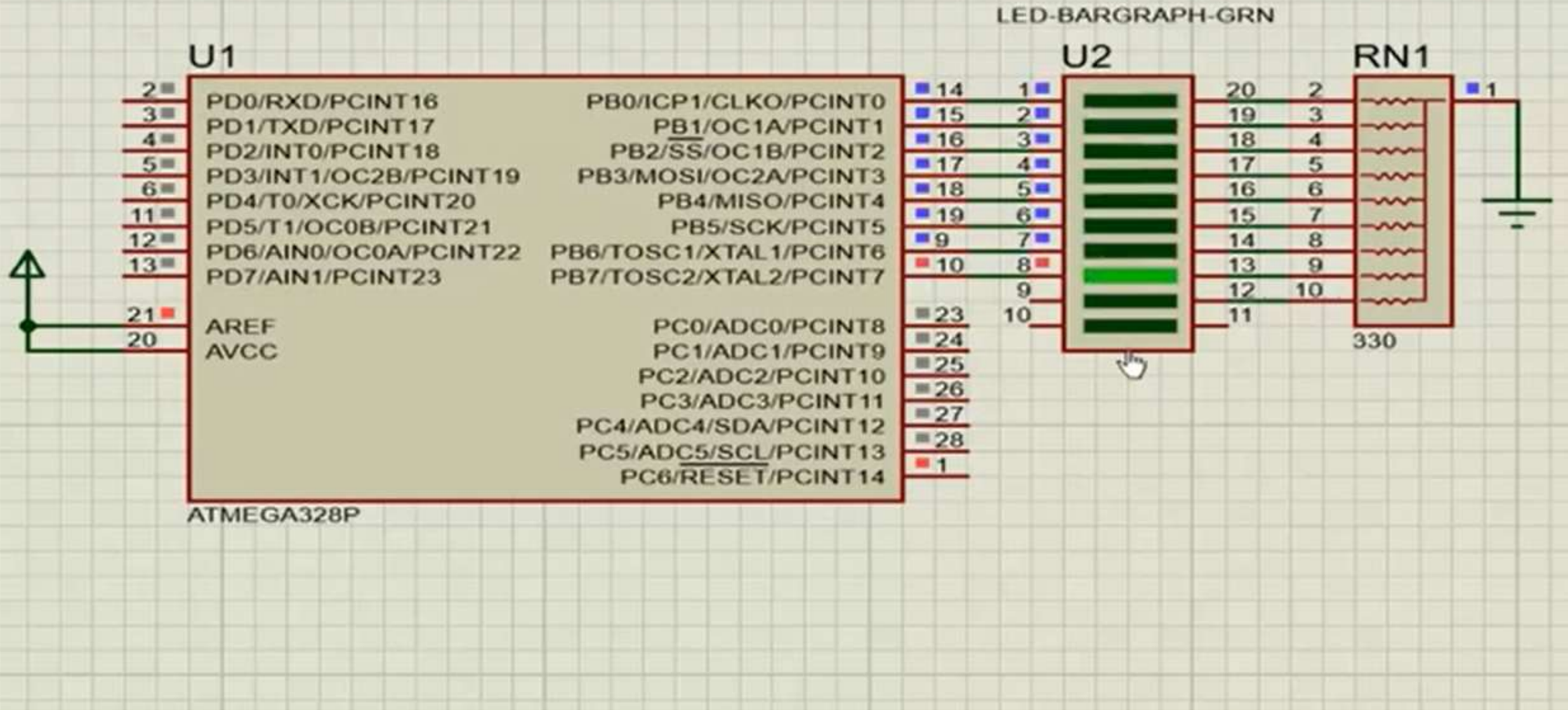


Componenete res 10


Edit Component
Part Reference:
Hidden:
OK
Select File Name
Buscar en:
Debug
Nombre
Fecha de modificación
Tipo
Tamaño
Saltos.hex
10/03/2021 04:12 p.m.
Archivo HEX
1 KB
Saltos.obj
10/03/2021 04:12 p.m.
3D Object
1 KB
Acceso rápido
Escritorio
Bibliotecas
Este equipo
Red
Nombre:
Tipo:
HEX, UBROF, COFF, ELF or OBJ Files
Cancelar
PH-GRN
RN1
330

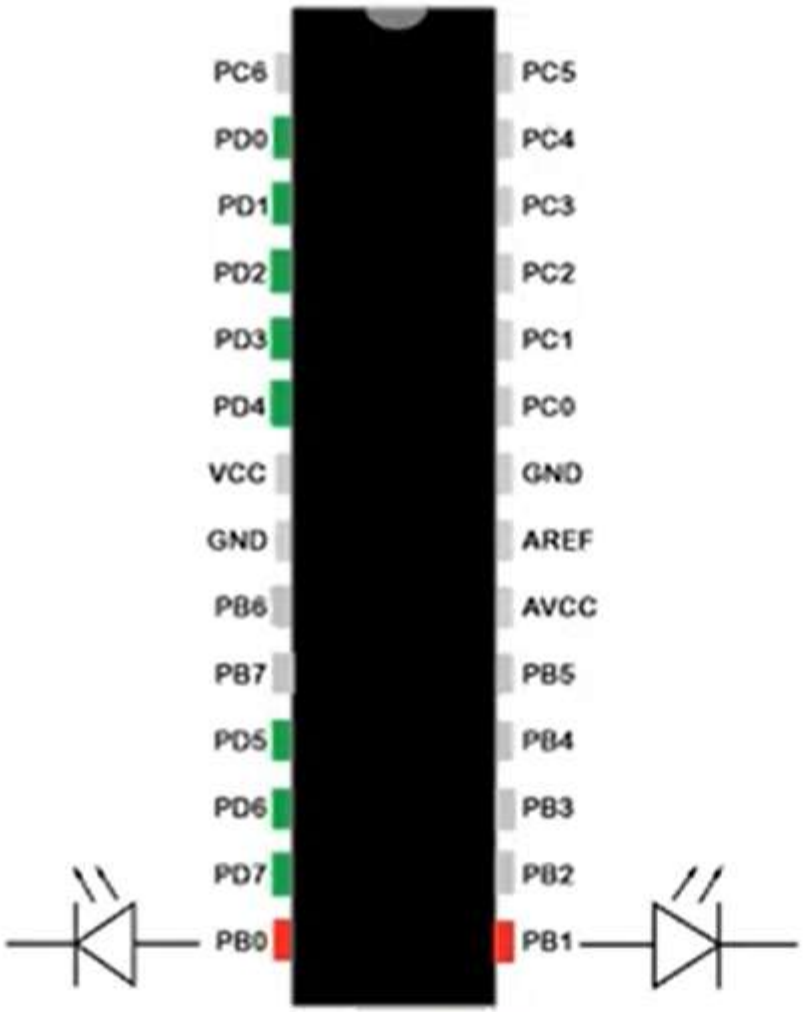

U1
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
AREF
AVCC
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
ATMEGA328P
LED-BARGRAPH-GRN
U2
RN1
330


U1
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
AREF
AVCC
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
ATMEGA328P
LED-BARGRAPH-GRN
U2
RN1
330


LED-BARGRAPH-GRN
U1
U2
RN1
PD0/RXD/PCINT16
PD1/TXD/PCINT17
PD2/INT0/PCINT18
PD3/INT1/OC2B/PCINT19
PD4/T0/XCK/PCINT20
PD5/T1/OC0B/PCINT21
PD6/AIN0/OC0A/PCINT22
PD7/AIN1/PCINT23
AREF
AVCC
PB0/ICP1/CLKO/PCINT0
PB1/OC1A/PCINT1
PB2/SS/OC1B/PCINT2
PB3/MOSI/OC2A/PCINT3
PB4/MISO/PCINT4
PB5/SCK/PCINT5
PB6/TOSC1/XTAL1/PCINT6
PB7/TOSC2/XTAL2/PCINT7
PC0/ADC0/PCINT8
PC1/ADC1/PCINT9
PC2/ADC2/PCINT10
PC3/ADC3/PCINT11
PC4/ADC4/SDA/PCINT12
PC5/ADC5/SCL/PCINT13
PC6/RESET/PCINT14
ATMEGA328P
330

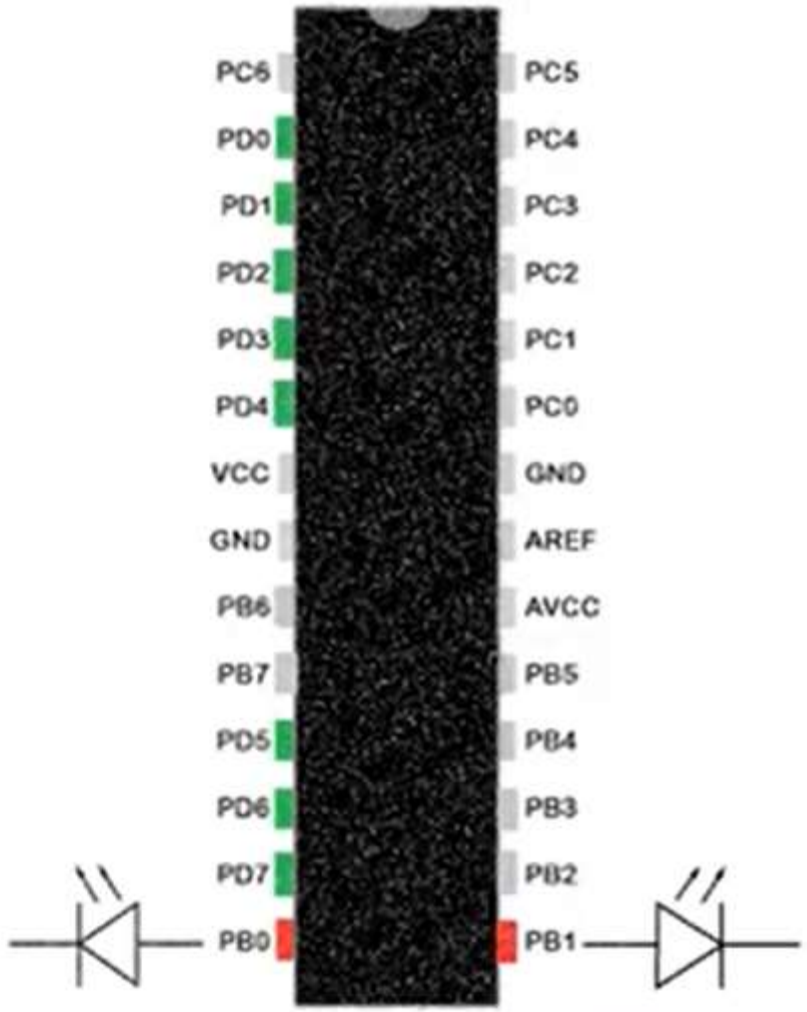
Ejercicio:

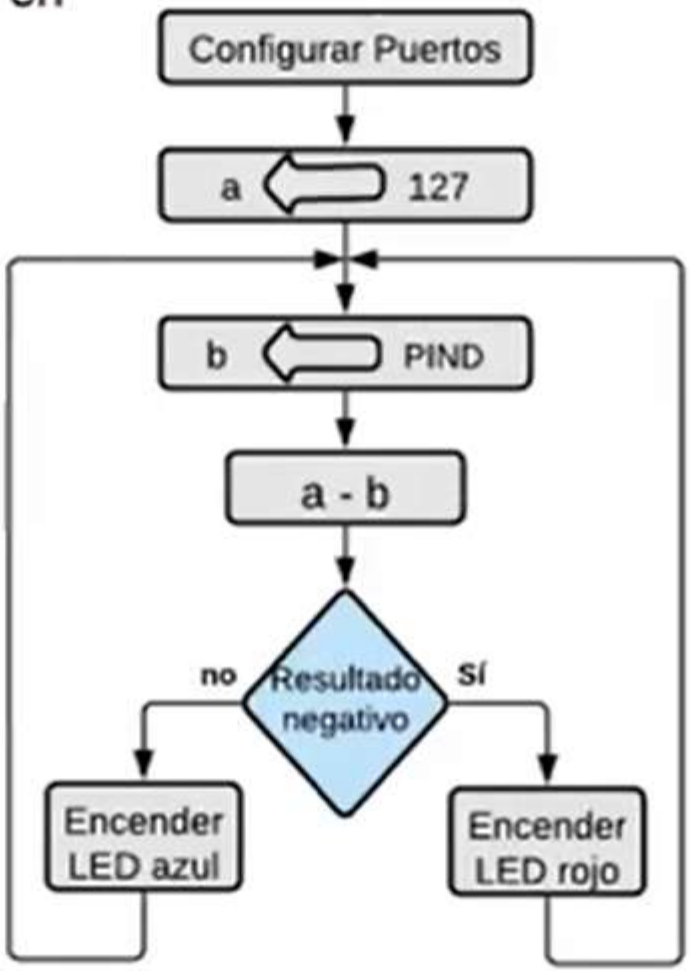
Diseñar un programa que realice la resta de dos valores. El primer valor será la constante 127, el segundo valor será dado por la lectura de pines en PORTD.



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!