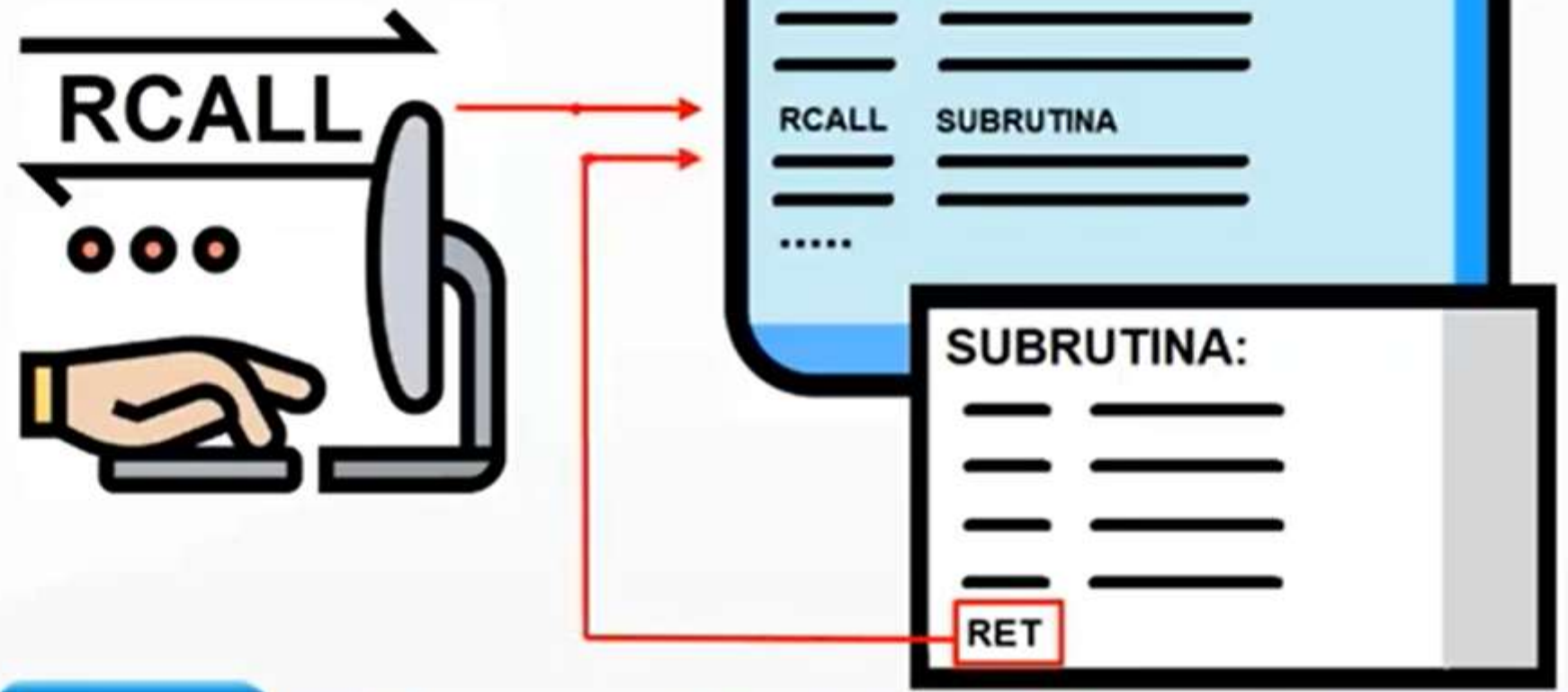
Instituto Tecnológico de Zitácuaro

Tema 2:
Sub Rutinas

¿Qué es una subrutina?

Porción de código que realiza una operación de forma independiente al resto del programa.



Ventajas de las subrutinas:

- Optimización de programa.

De preferencia al final.

Llamadas a subrutina

Algunos mnemónicos son:

CALL

llamada larga a subrutina.

Sintaxis: CALL k

Pila: Stack <- PC + 2

RCALL

llamada relativa a subrutina.

Sintaxis: RCALL k

Pila: Stack <- PC + 1

ATmega328p

Llamadas a subrutina

Algunos mnemónicos son:

RCALL

llamada relativa a subrutina.

Sintaxis: RCALL k

Pila: Stack <- PC + 1

RET

Retorno de subrutina.

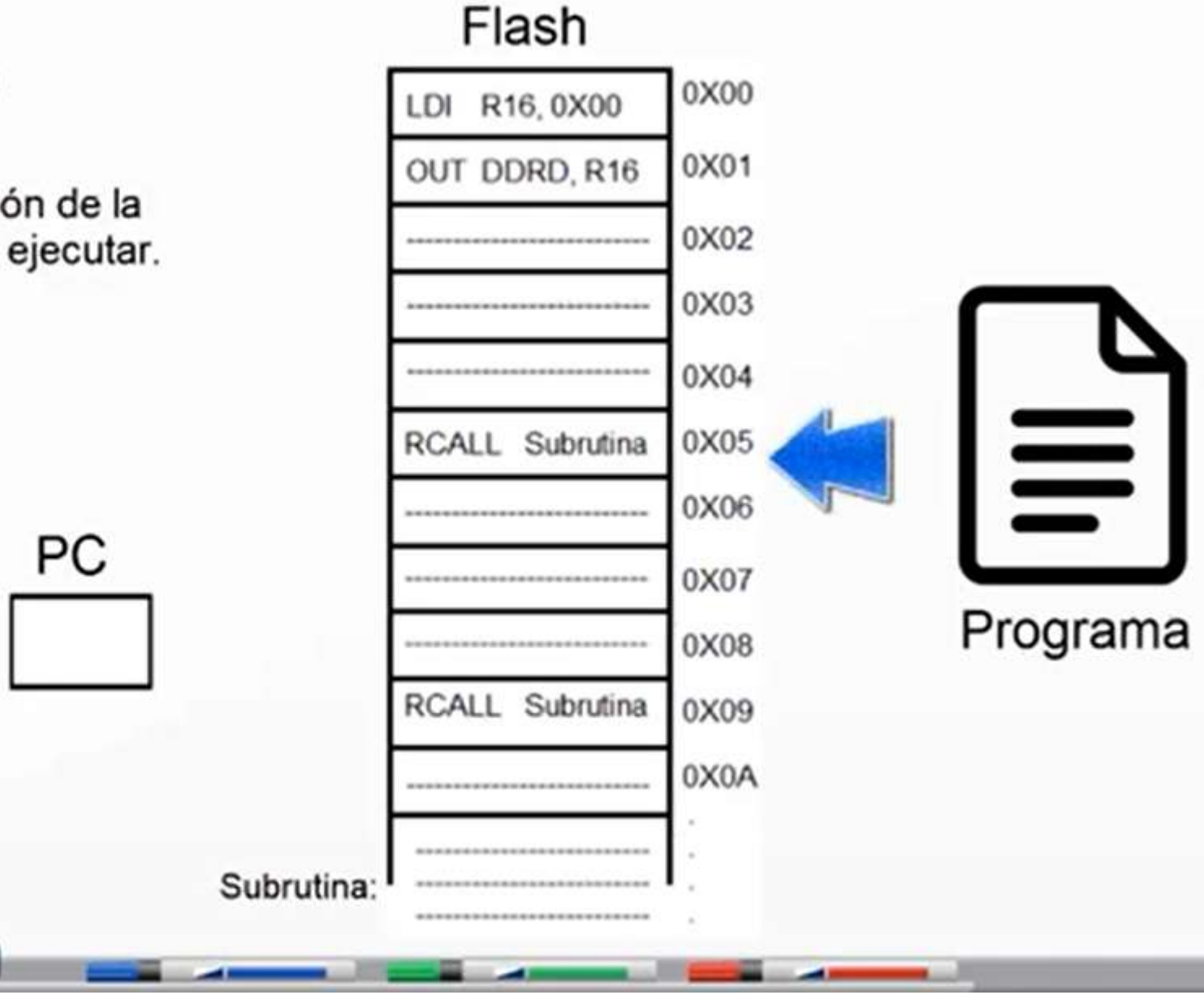
Sintaxis: RET

Operación: PC <- Stack

ATmega328p

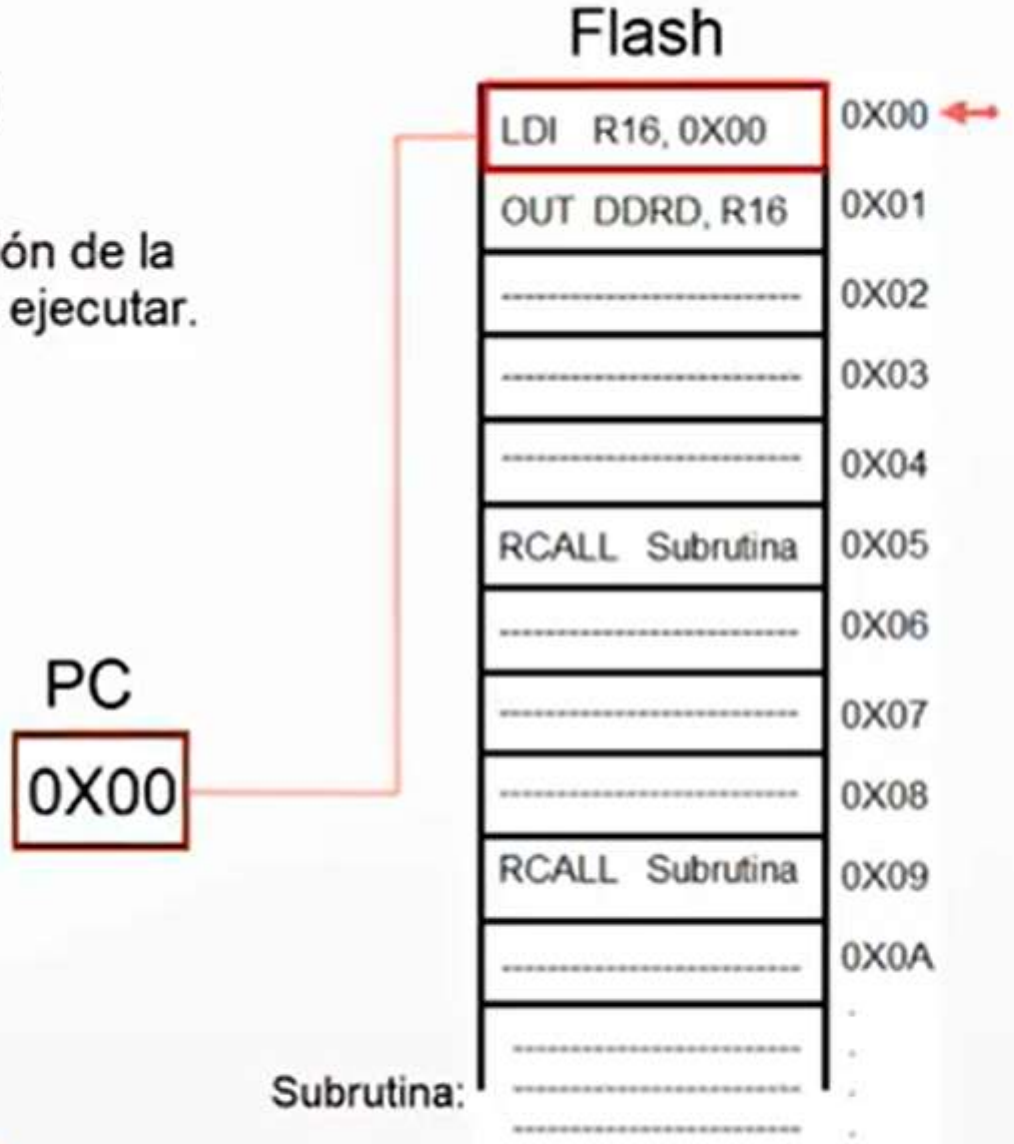
Program Counter (PC)

Su valor corresponde a la dirección de la siguiente instrucción que se va a ejecutar.



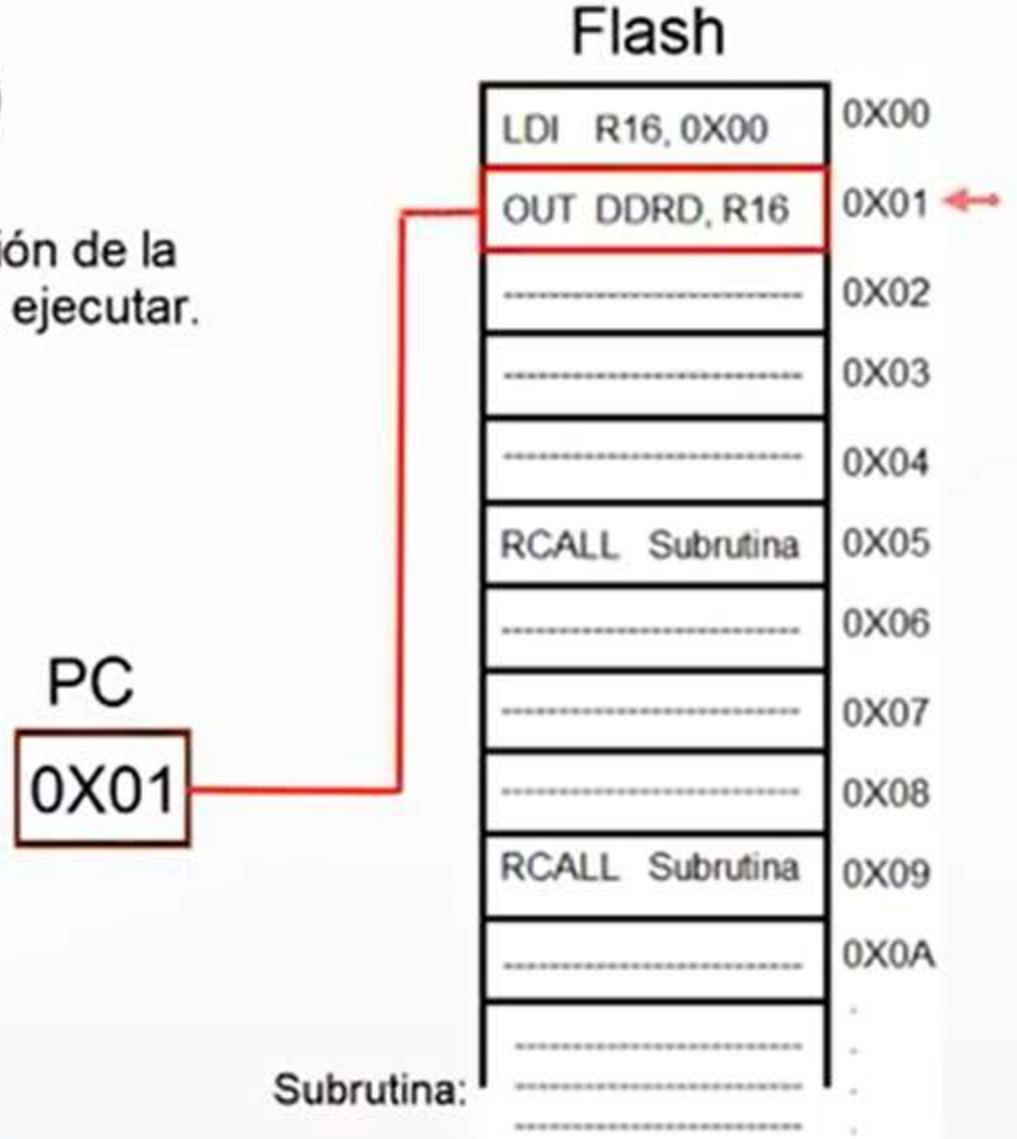
Program Counter (PC)

Su valor corresponde a la dirección de la siguiente instrucción que se va a ejecutar.



Program Counter (PC)

Su valor corresponde a la dirección de la siguiente instrucción que se va a ejecutar.



Program Counter (PC)

Su valor corresponde a la dirección de la siguiente instrucción que se va a ejecutar.

PC

0X0C

Flash

LDI R16, 0X00	0X00
OUT DDRD, R16	0X01
----------	0X02
----------	0X03
----------	0X04
RCALL Subrutina	0X05
----------	0X06
----------	0X07
----------	0X08
RCALL Subrutina	0X09
----------	0X0A

Subrutina: 0X0C

Stack (Pila)

Es memoria donde se guardan esas direcciones de retorno. Está situada en la memoria RAM.

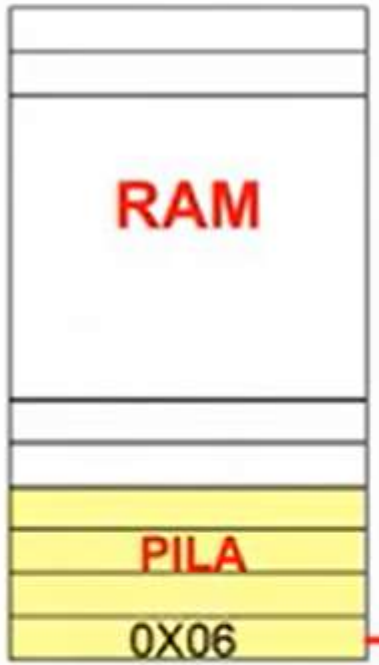
Flash

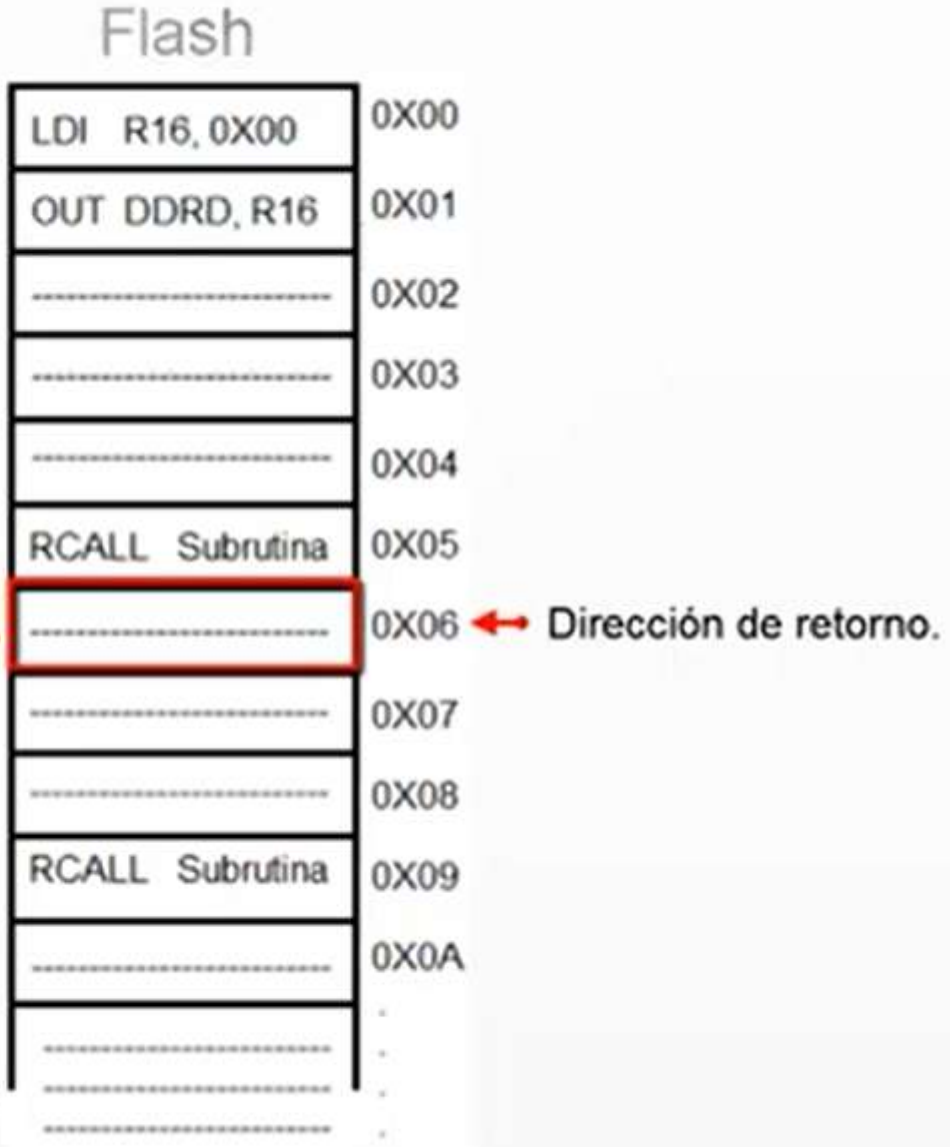
LDI R16, 0X00	0X00
OUT DDRD, R16	0X01
----------	0X02
----------	0X03
----------	0X04
RCALL Subrutina	0X05
----------	0X06 ← Dirección de retorno.
----------	0X07
----------	0X08
RCALL Subrutina	0X09
----------	0X0A
----------	.
----------	.
----------	.

Subrutina:

Stack (Pila)

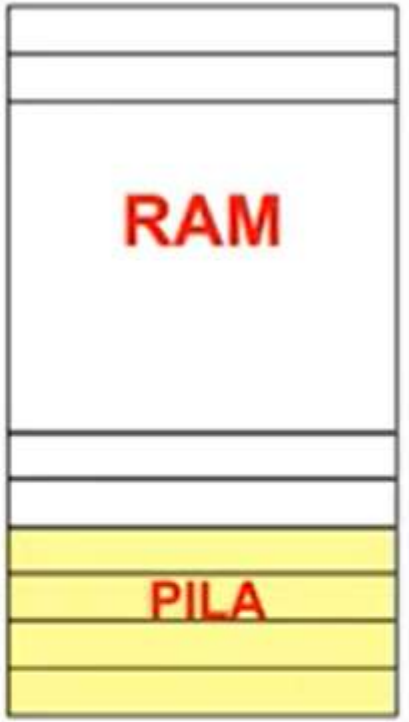
Es memoria donde se guardan esas direcciones de retorno. Está situada en la memoria RAM.





Stack Pointer (Puntero de pila)

Habilita la celda de la pila donde se guardarán las direcciones de retorno para una interrupción o una subrutina.



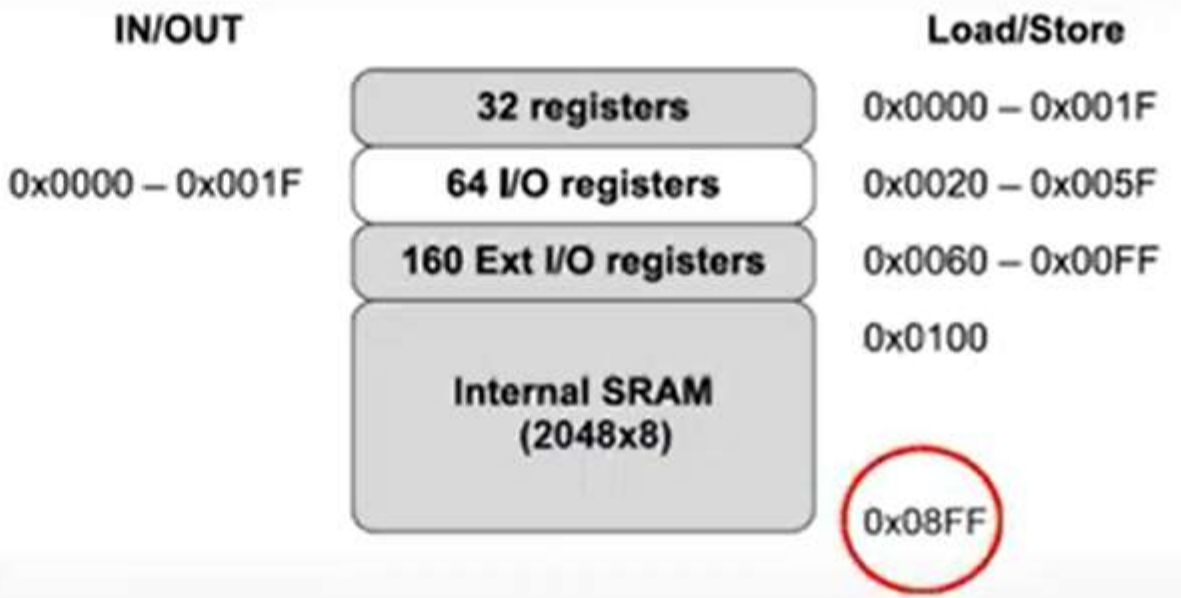
Stack Pointer (Puntero de pila)

AVR -> 16 BITS

Habilita la celda de la pila donde se guardarán las direcciones de retorno para una interrupción o una subrutina.

2 registros

SPH	SPL

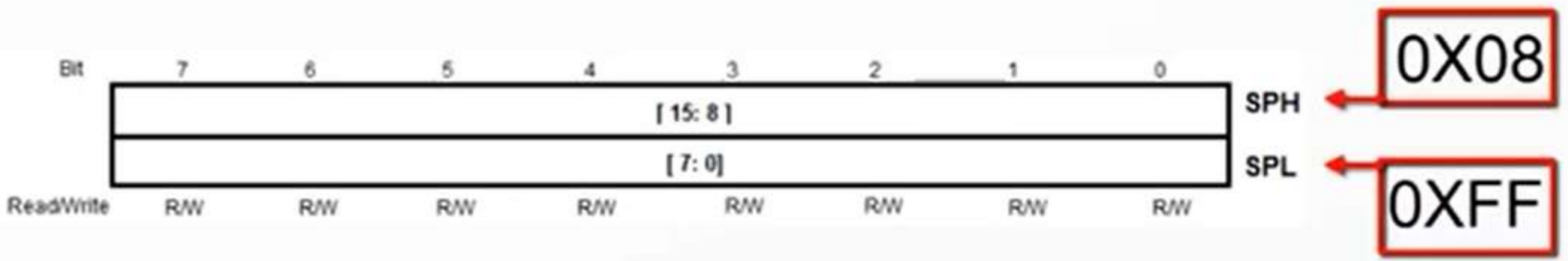


Stack Pointer (Puntero de pila)

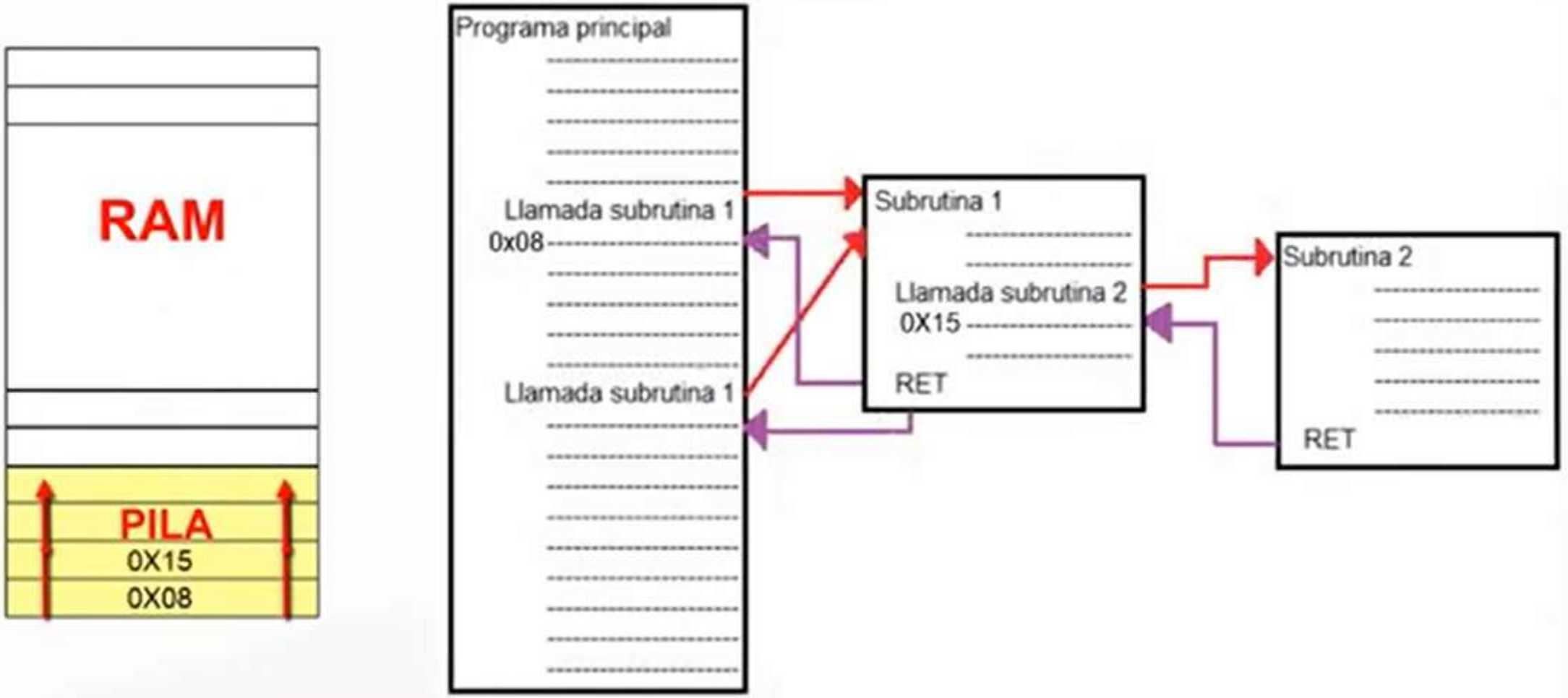
AVR -> 16 BITS

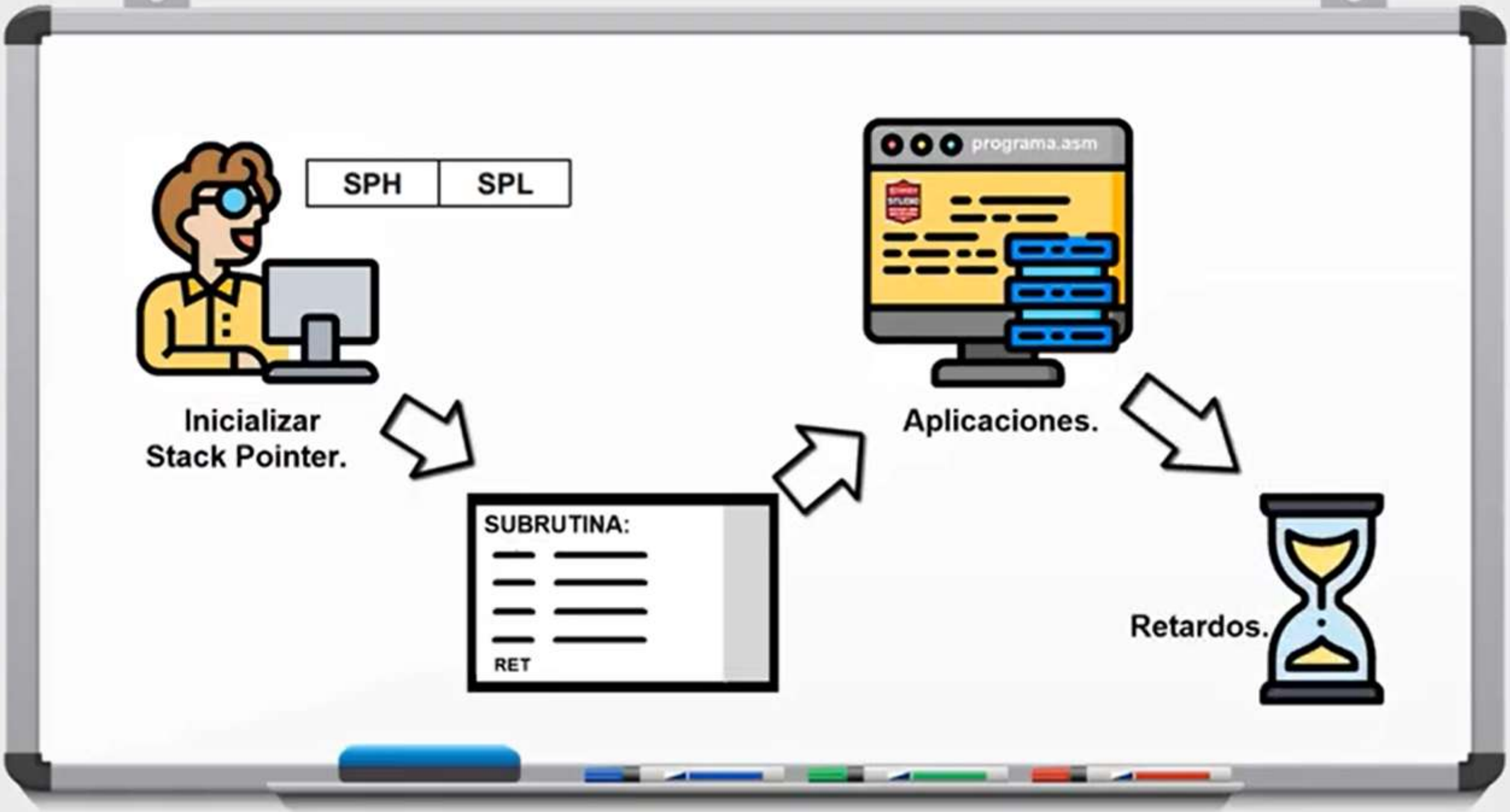
Habilita la celda de la pila donde se guardarán las direcciones de retorno para una interrupción o una subrutina.

2 registros



Stack Pointer (Puntero de pila)




SPH
SPL
Inicializar
Stack Pointer.
SUBRUTINA:
RET
programa.asm
Aplicaciones.
Retardos.

- ¿En qué consiste un retardo?

- Consideraciones.

- Métodos para realizar retardos:

  - Ciclos anidados.
  - AVR delay loop calculator.

- Ejemplo de aplicación.

ATmega328p

Proteus

Retardo

Rutina cuya función es consumir ciclos de reloj durante cierta cantidad de tiempo definida.

Programación

Microcontrolador

Consideraciones al realizar una rutina de retardo:

- Datos del microcontrolador (Frecuencias).

1 segundo ⟶ **1 millón de operaciones**

Consideraciones al realizar una rutina de retardo:

- Datos del microcontrolador (Frecuencias).

- Lenguaje de programación.

Consideraciones al realizar una rutina de retardo:

- Datos del microcontrolador (Frecuencias).

- Lenguaje de programación.

- Ciclos de reloj por operación.

Consideraciones al realizar una rutina de retardo:

- Datos del microcontrolador (Frecuencias).

- Lenguaje de programación.

- Ciclos de reloj por operación.

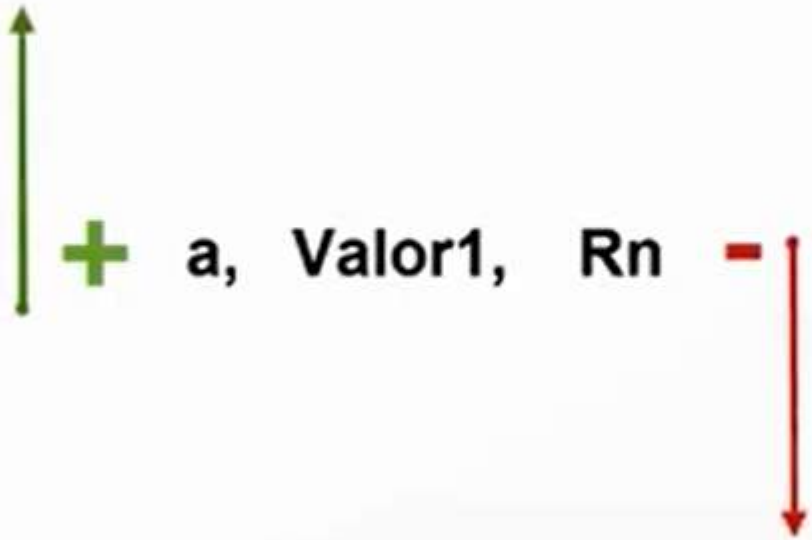
- Requerimientos del problema.

Ciclo anidado

Bucle que se encuentra incluido o enlazado
dentro de otro bucle.

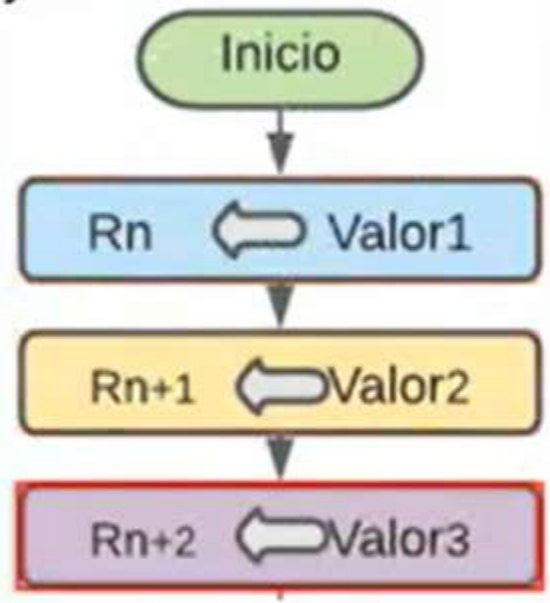
Ciclo anidado

Bucle que se encuentra incluido o enlazado dentro de otro bucle.



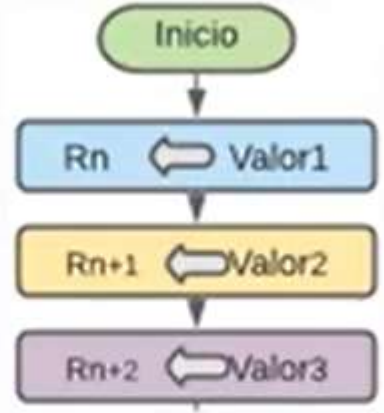
Método de ciclos anidados

- Cargar constantes a registros de trabajo.

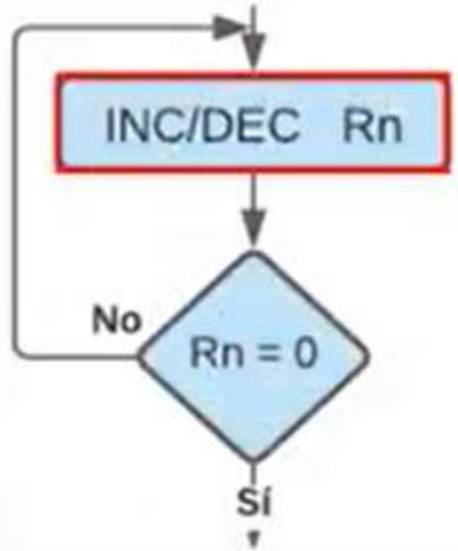


Método de ciclos anidados

- Cargar constantes a registros de trabajo.



- Incrementar o decrementar la primera variable en bucle.

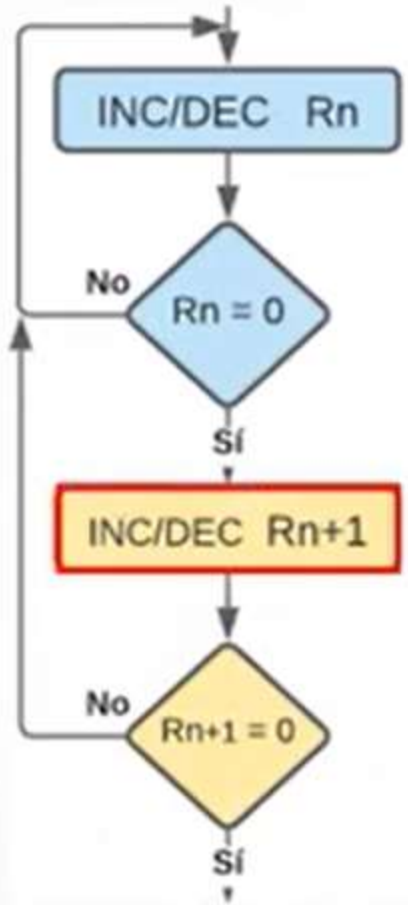


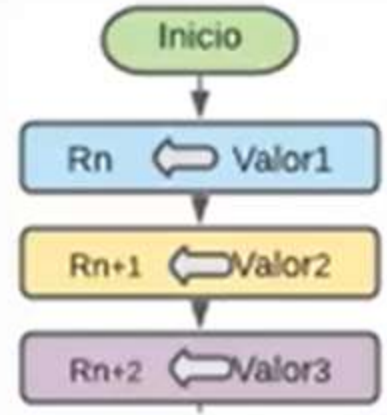
Método de ciclos anidados

- Cargar constantes a registros de trabajo.

- Incrementar o decrementar la primera variable en bucle.

- Incrementar o decrementar la segunda variable, con salto al primer bucle.





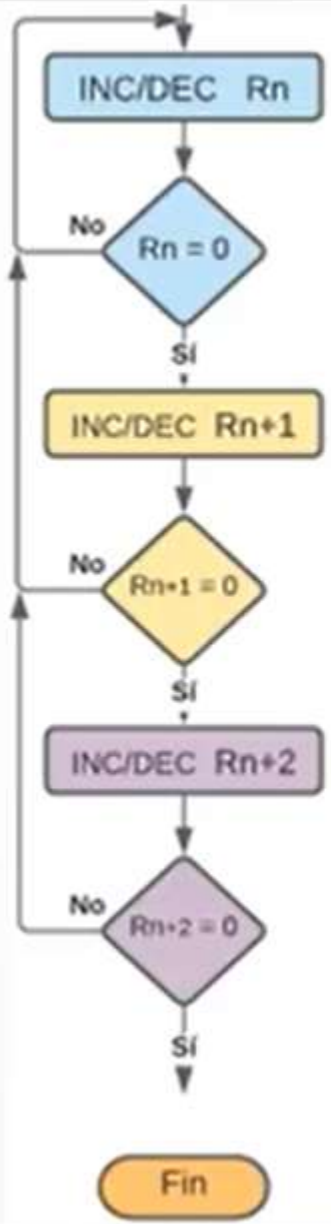
Método de ciclos anidados

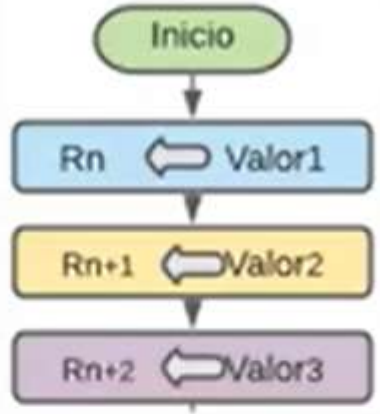
- Cargar constantes a registros de trabajo.

- Incrementar o decrementar la primera variable en bucle.

- Incrementar o decrementar la segunda variable, con salto al primer bucle.

- Repetir con todas las variables a utilizar.

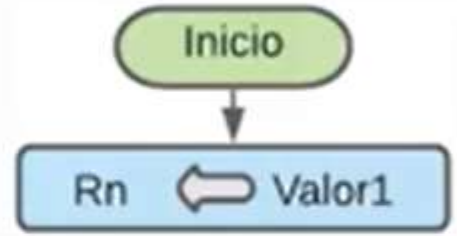


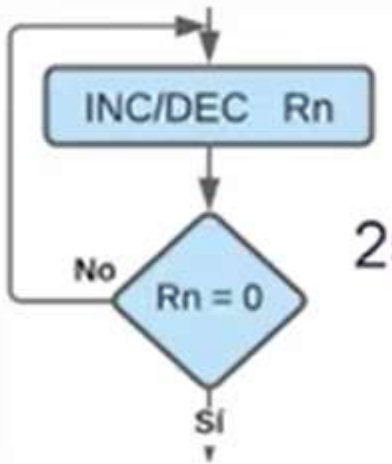


Método de ciclos anidados

Retardo de 1/2 segundo (500,000 operaciones)

R19 ⟵ 0

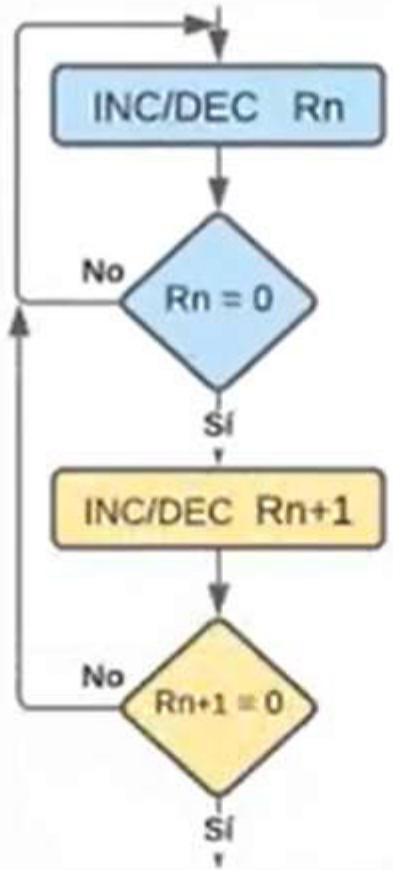




256 x 3 = 768

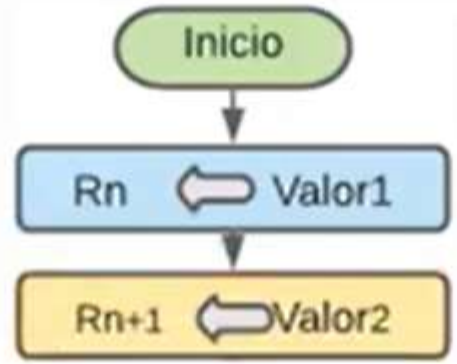
Método de ciclos anidados

Retardo de 1/2 segundo (500,000 operaciones)



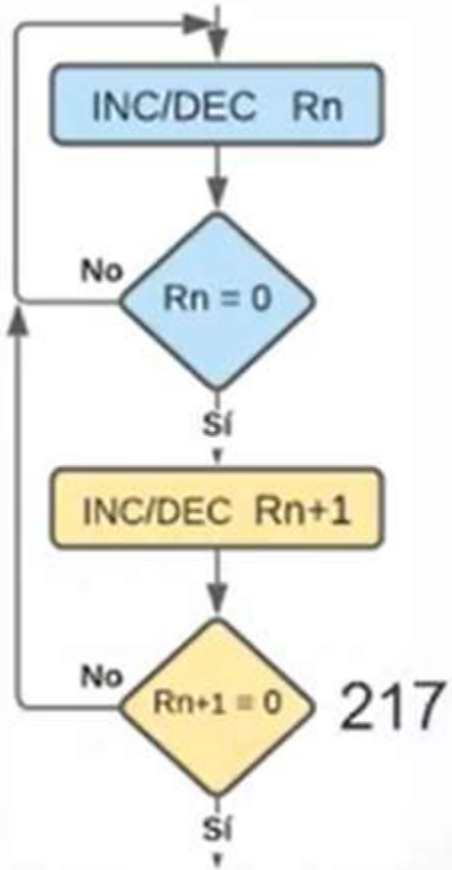
R19 ⟵ 0

R20 ⟵ 39



Método de ciclos anidados

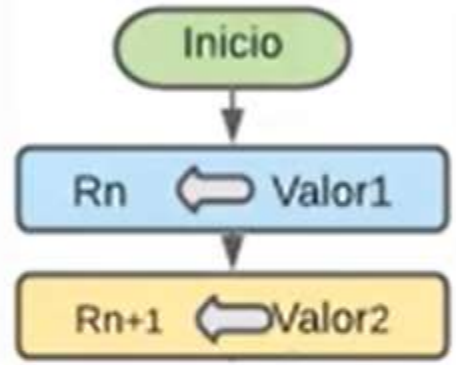
Retardo de 1/2 segundo (500,000 operaciones)



217

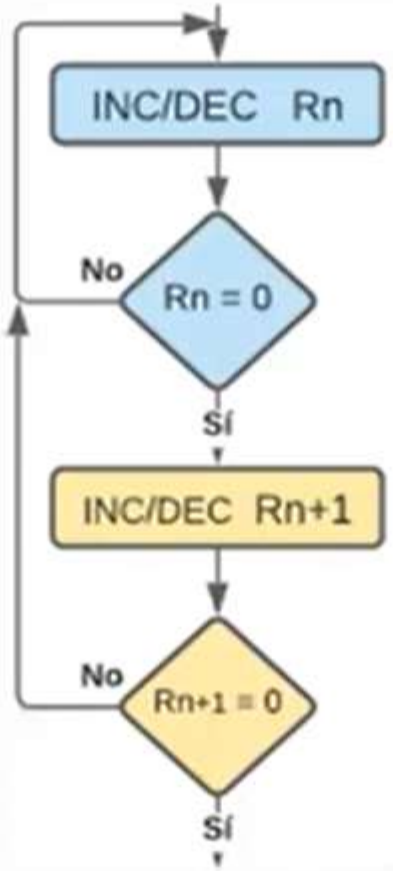
R19 ⟵ 0

R20 ⟵ 39



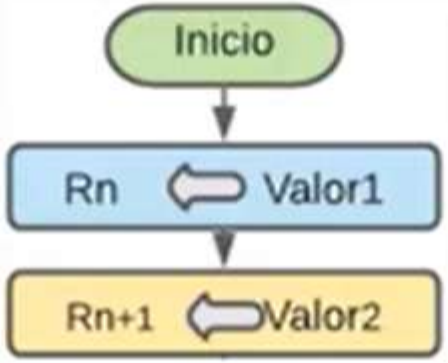
Método de ciclos anidados

Retardo de 1/2 segundo (500,000 operaciones)



R19 ⟵ 0

R20 ⟵ 39

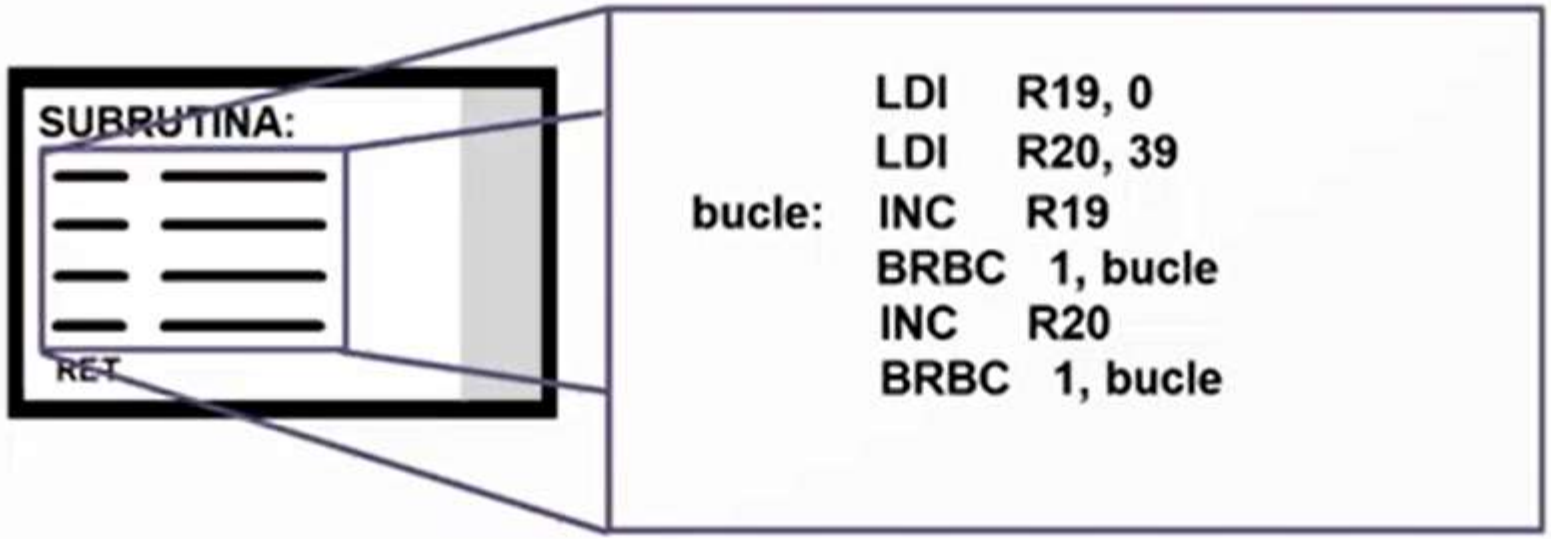


(256x3) x (217x3) = 499,968

1er. bucle 2do. bucle

Método de ciclos anidados

Retardo de 1/2 segundo (500,000 operaciones)




```
        LDI   R19, 0
        LDI   R20, 39
bucle:  INC   R19
        BRBC  1, bucle
        INC   R20
        BRBC  1, bucle
```


Google
avr delay loop cal
avr delay loop calculator
Buscar con Google
Me siento con suerte
Denunciar predicciones ofensivas

Google avr delay loop calculator

Todo Imágenes Shopping Videos Noticias 1 Más Preferencias Herramientas

Cerca de 1,100,000 resultados (0.54 segundos)

http://darcy.rsgc.on.ca › ACES › A... ▾ 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.

https://www.avrfreaks.net › forum ▾ Traducir esta página

Improved delay loop calculator | AVR Freaks

2 ene. 2011 — Here's a one-day **loop** for a 20MHz clock. ; **Delay** 1 728 000 000 000 cycles ldi r18, 134 ldi r19, 195 ldi r20, 193 ldi r21, 122 ldi r22, 166 L1: dec ...

16 publicaciones · Hi Hi Bret. Nice java applet. one thing to make it a bit more sequential is yo...

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** possible ...	12 publicaciones	25 de feb. de 2016
Calculate the maximum **delay** | **AVR** Freaks	17 publicaciones	9 de abr. de 2010

Más resultados de www.avrfreaks.net

http://web.csulb.edu › ~hill › Lectures › 06 AVR ... ▾ por

AVR Looping

500 µs. **DELAY CALCULATION** FOR **AVR**. • We begin by designing a simple **loop**. wait: ldi r16, ...

https://www.softpedia.com › get ▾ 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.

http://www.avr-asm-tutorial.net › d... ▾ Traducir esta página

Delay loops in AVR assembler - AVR-Assembler-Tutorial

AVR Delay Loop Calculator

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

8.0 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Poner velocidad en MHZ del microprocesador

Atmega328p es de 1 Mhz

AVR Delay Loop Calculator

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

8.0 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Número de ciclos a considerar

AVR Delay Loop Calculator

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

8.0 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s min hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Indicar el registro en el cual se iniciarán las variables a decrementar

AVR Delay Loop Calculator

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

8.0 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

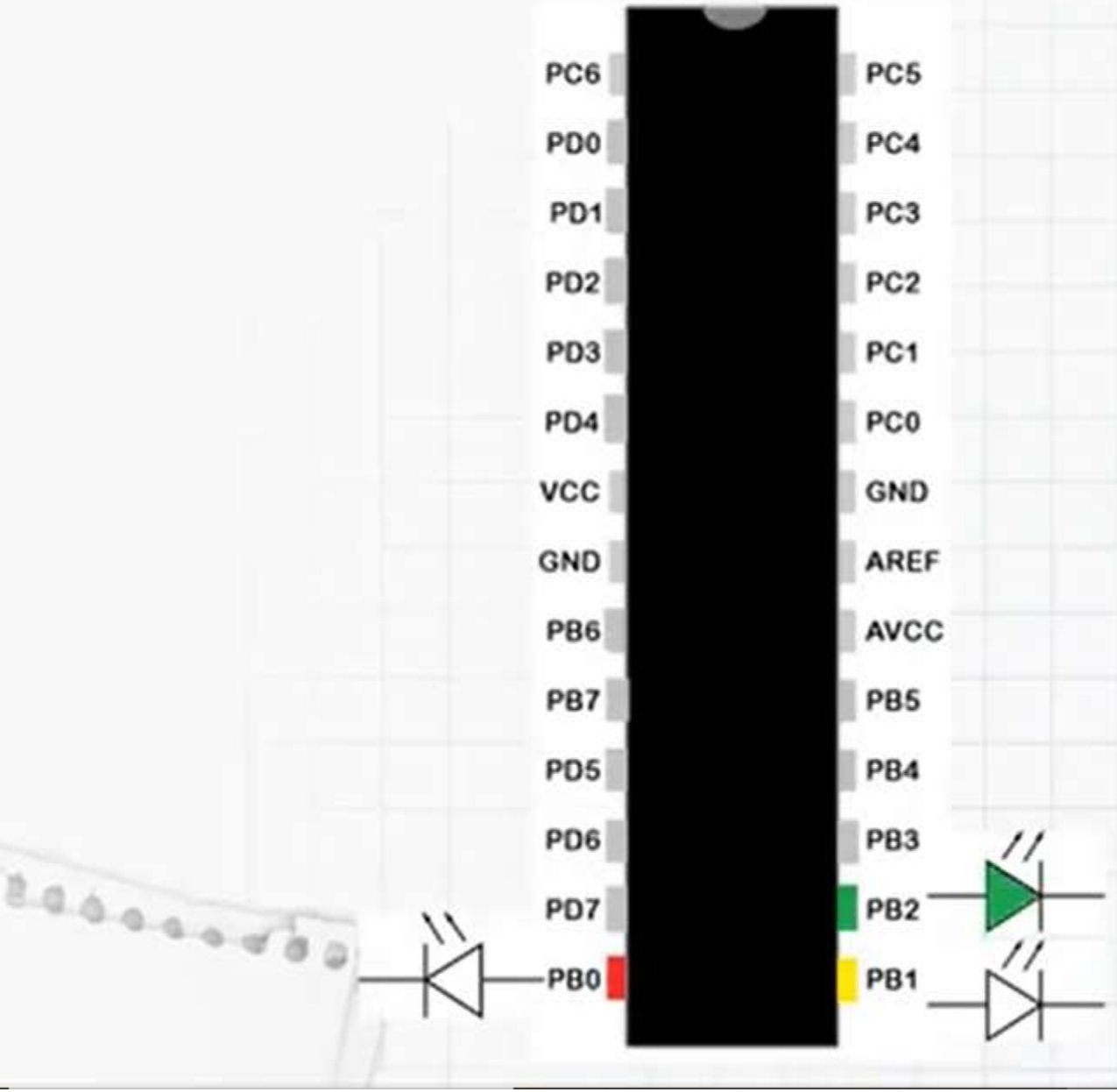
```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Seleccionar el tiempo deseado con su correspondiente unidad

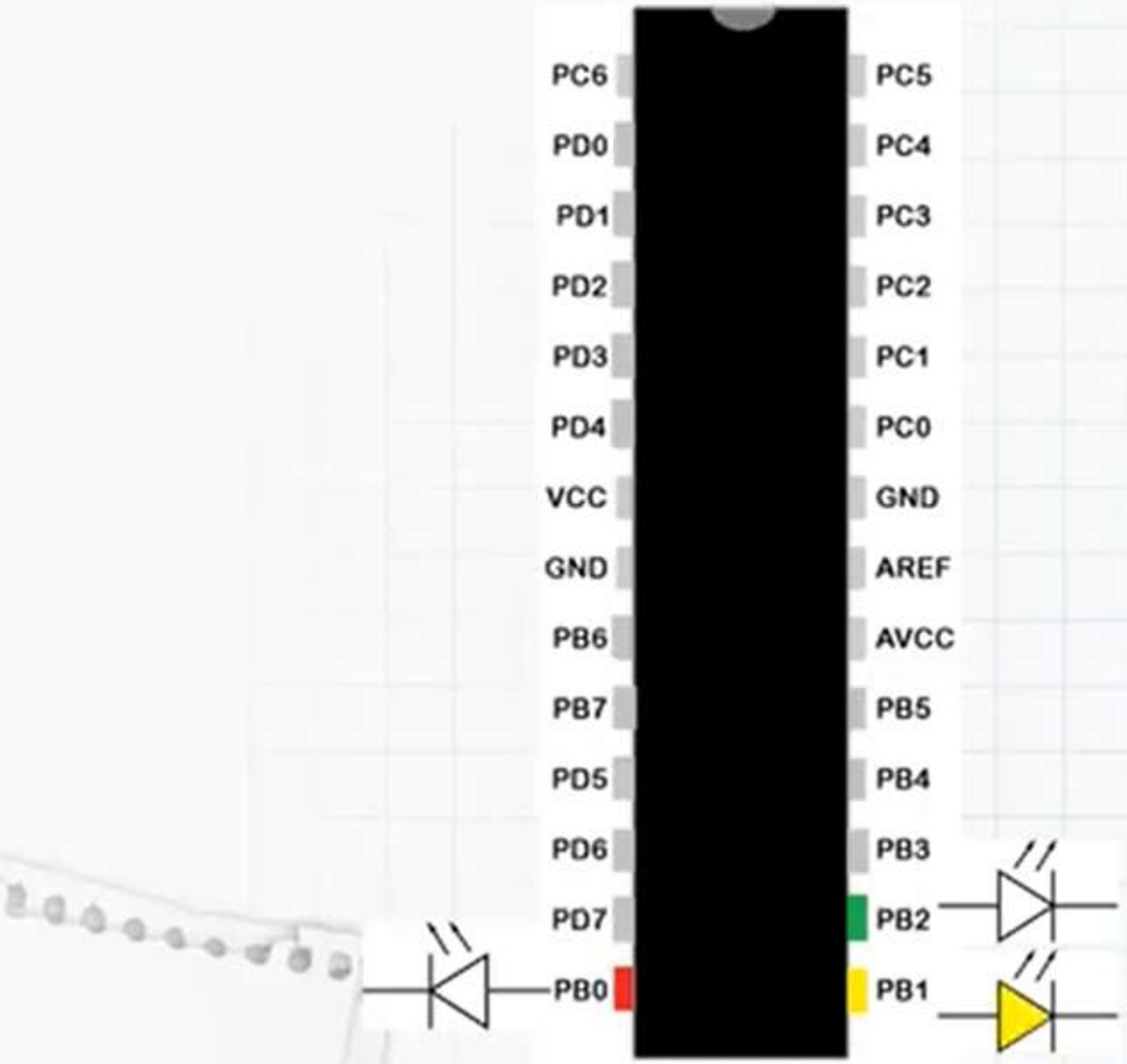
O número de ciclos

Ejemplo

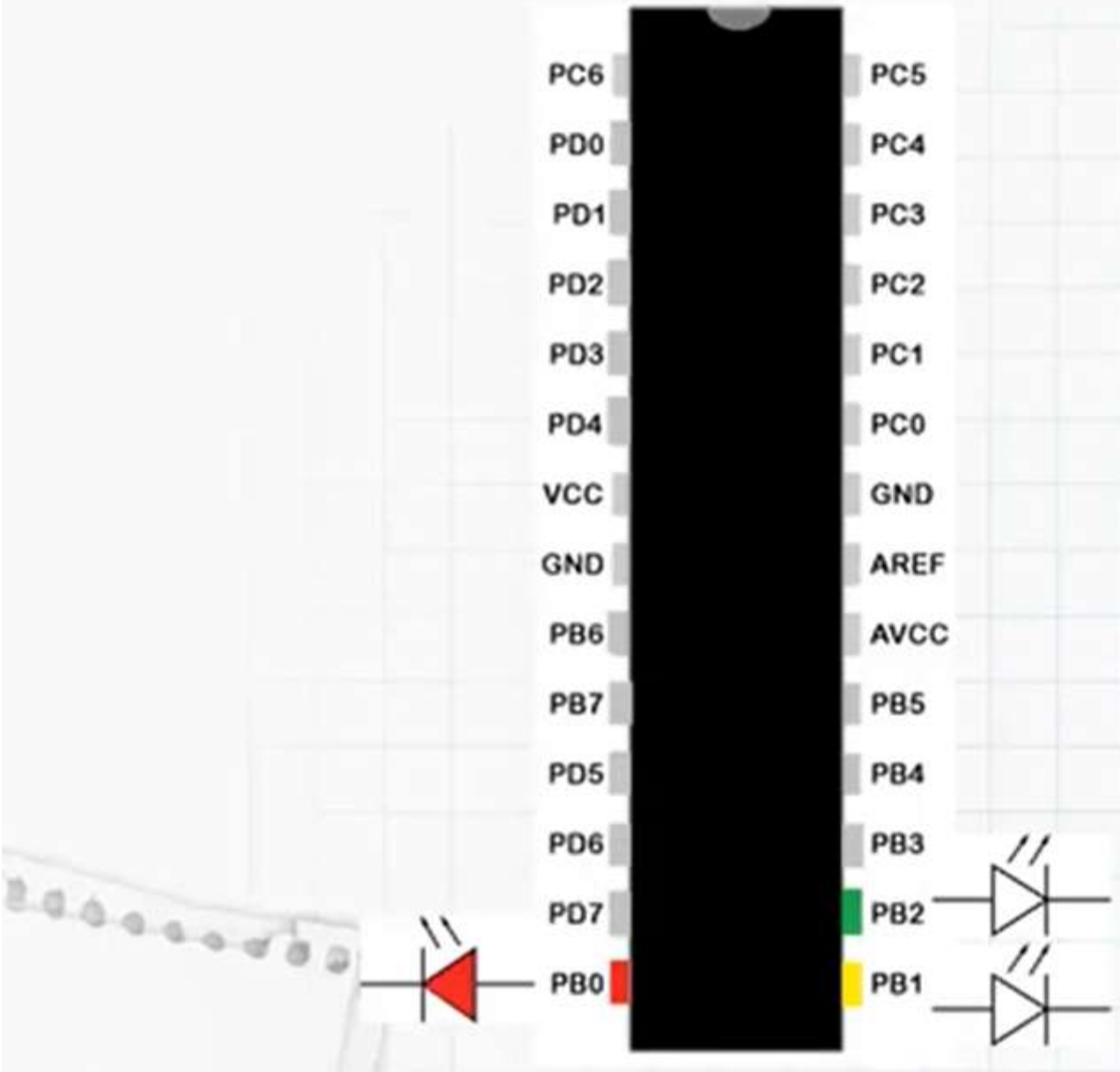
Encender LED verde 3 segundos, hacer que parpadee los siguientes 3 para anunciar el cambio, encender LED amarillo 3 segundos y encender LED rojo 6 segundos. Repetir ciclicamente.



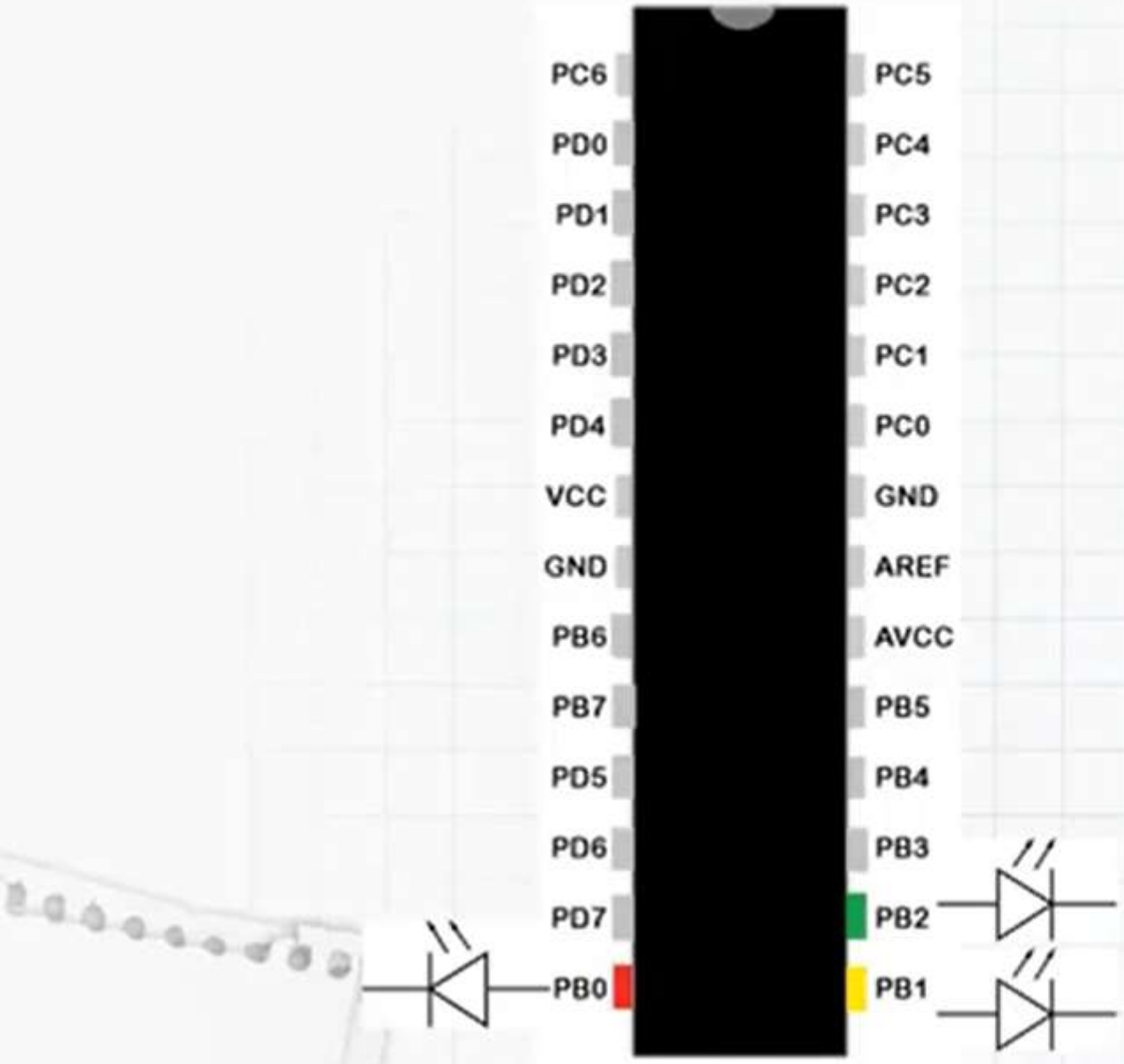
Encender LED verde 3 segundos, hacer que parpadee los siguientes 3 para anunciar el cambio, encender LED amarillo 3 segundos y encender LED rojo 6 segundos. Repetir ciclicamente.

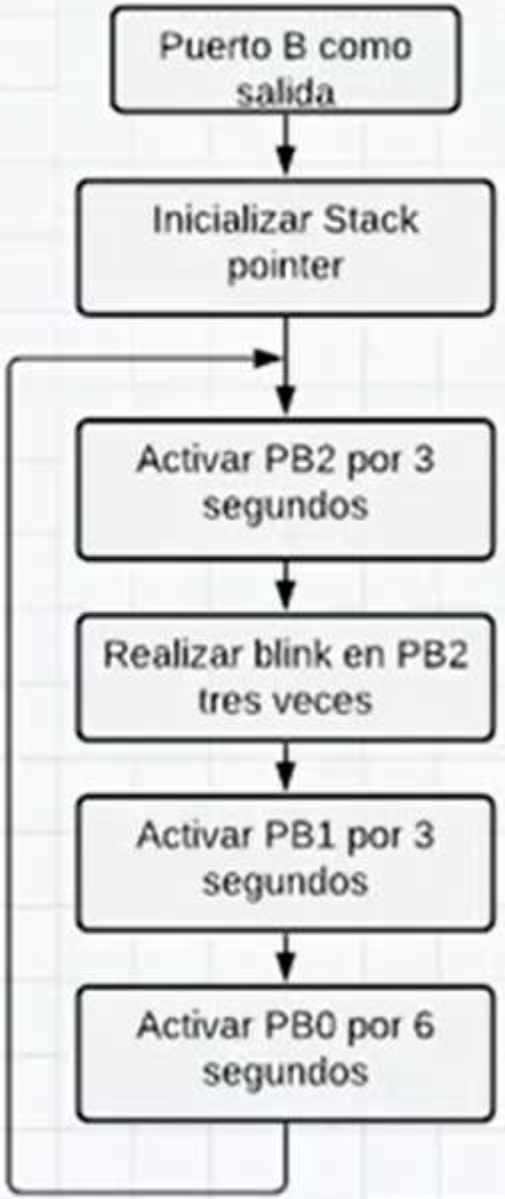


Encender LED verde 3 segundos, hacer que parpadee los siguientes 3 para anunciar el cambio, encender LED amarillo 3 segundos y encender LED rojo 6 segundos. Repetir ciclicamente.

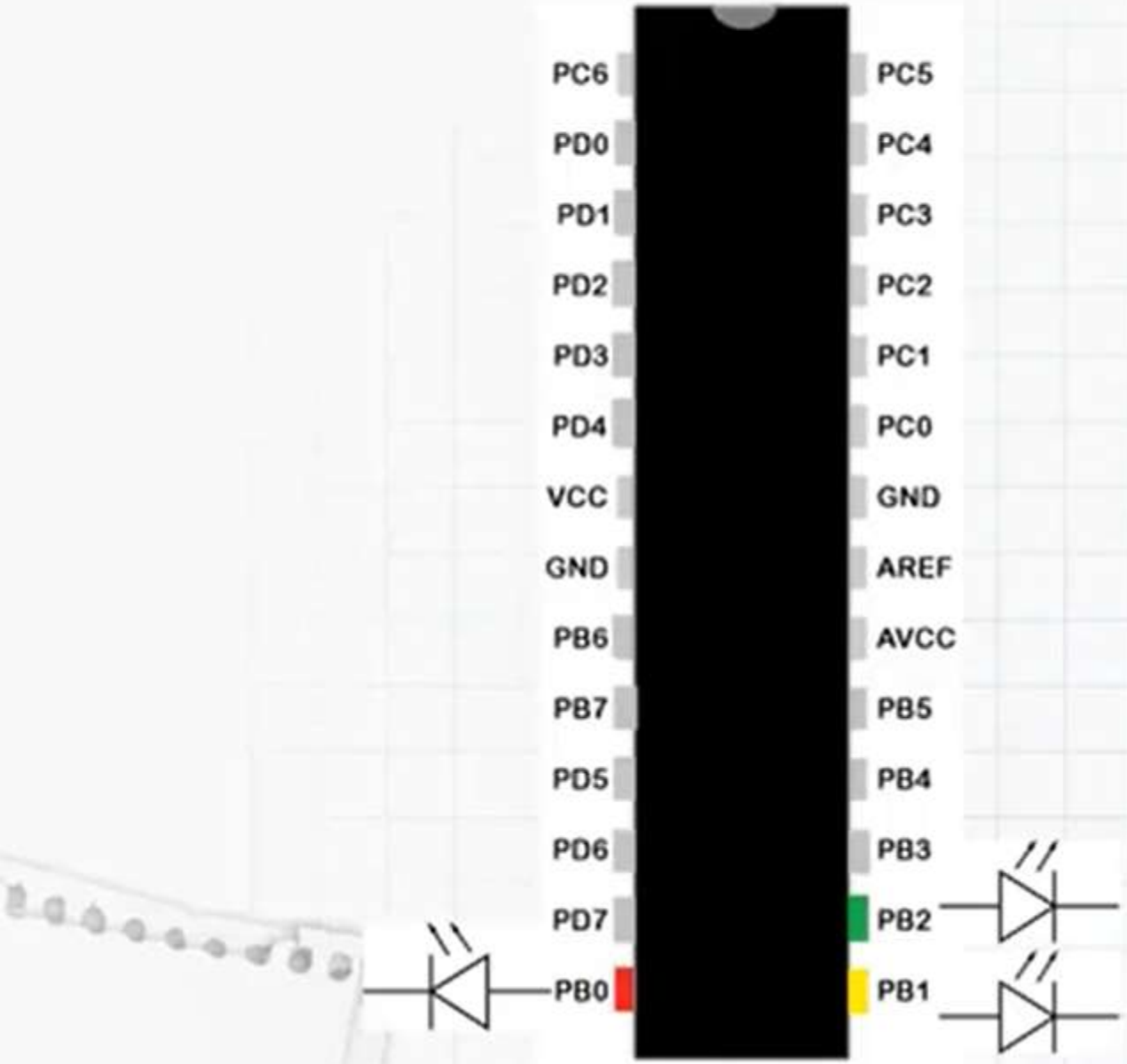


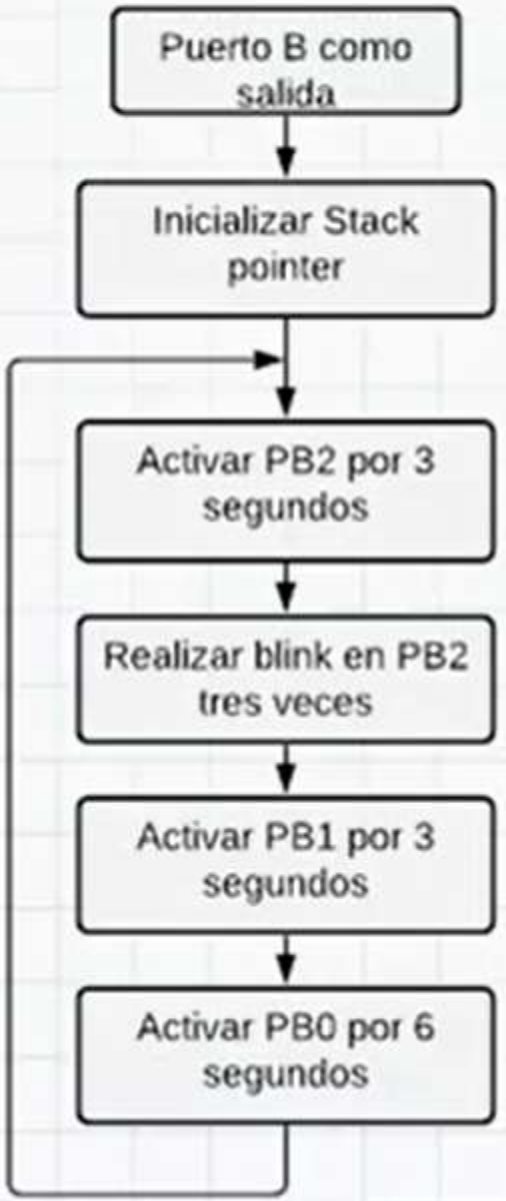
Encender LED verde 3 segundos, hacer que parpadee los siguientes 3 para anunciar el cambio, encender LED amarillo 3 segundos y encender LED rojo 6 segundos. Repetir ciclicamente.





Encender LED verde 3 segundos, hacer que parpadee los siguientes 3 para anunciar el cambio, encender LED amarillo 3 segundos y encender LED rojo 6 segundos. Repetir ciclicamente.





main.asm*

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

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

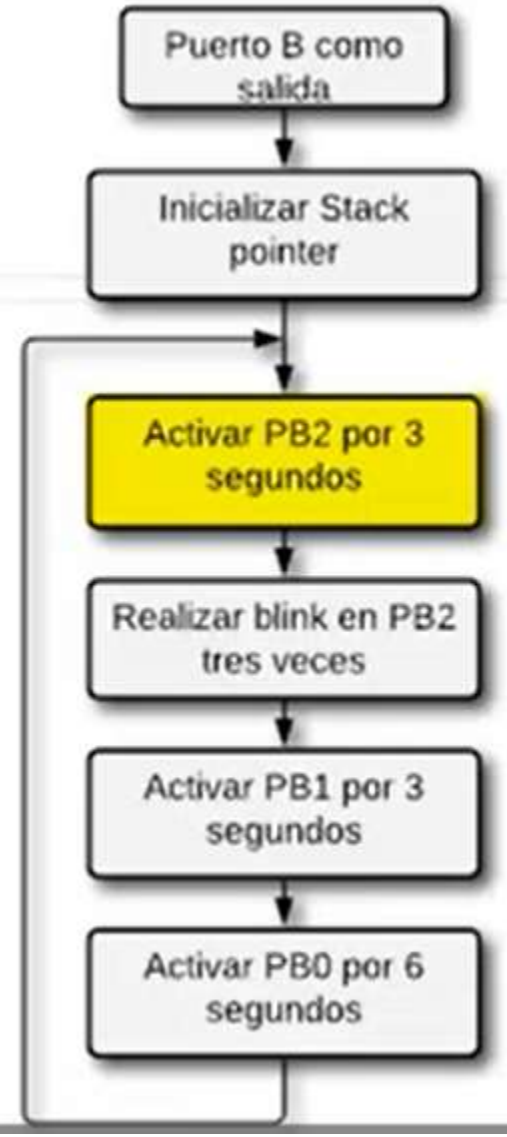
```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

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

Configurar el puntero de pila

0X08	0XFF
SPH	SPL

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORT
```

Mnemónico	Operandos	Descripción
SBI	P, b	Poner a uno el bit de un registro I/O
CBI	P, b	Poner a cero el bit de un registro I/O



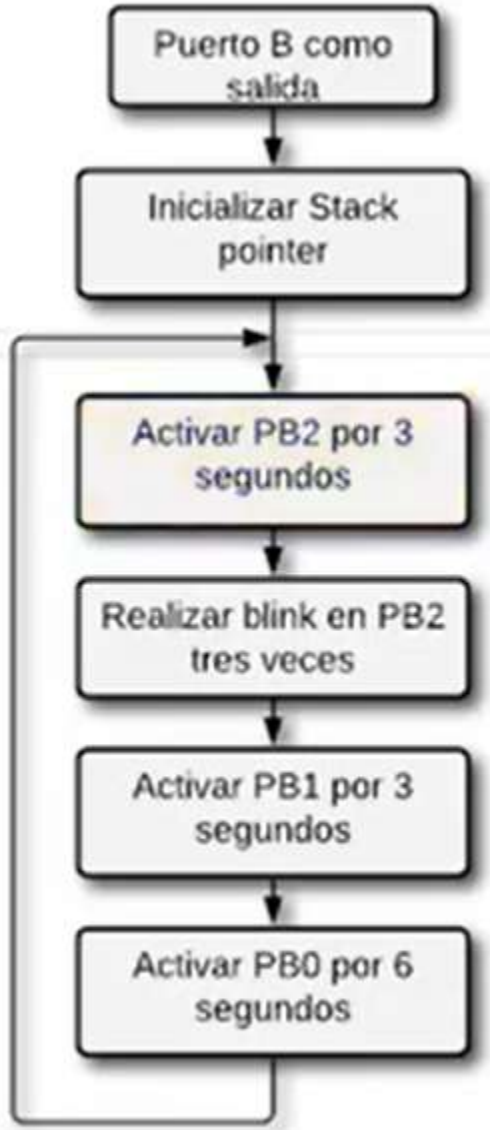
```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF
        OUT     DDRB, R16       ; Configura puerto B como salida.
        LDI     R17, 0x08
        OUT     SPL, R16
        OUT     SPH, R17        ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL
```

En el pin 2

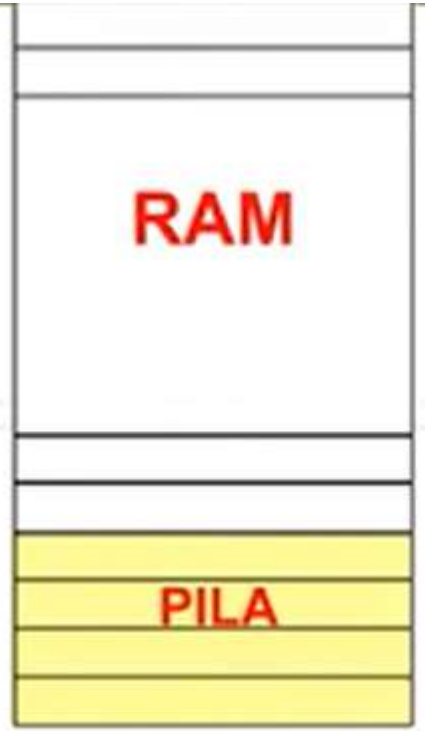
```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, R16       ; Configura puerto B como salida.
        LDI     R17, 0X08
        OUT     SPL, R16
        OUT     SPH, R17        ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2
```

Mnemónico	Operandos	Descripción
SBI	P, b	Poner a uno el bit de un registro I/O
CBI	P, b	Poner a cero el bit de un registro I/O

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
```

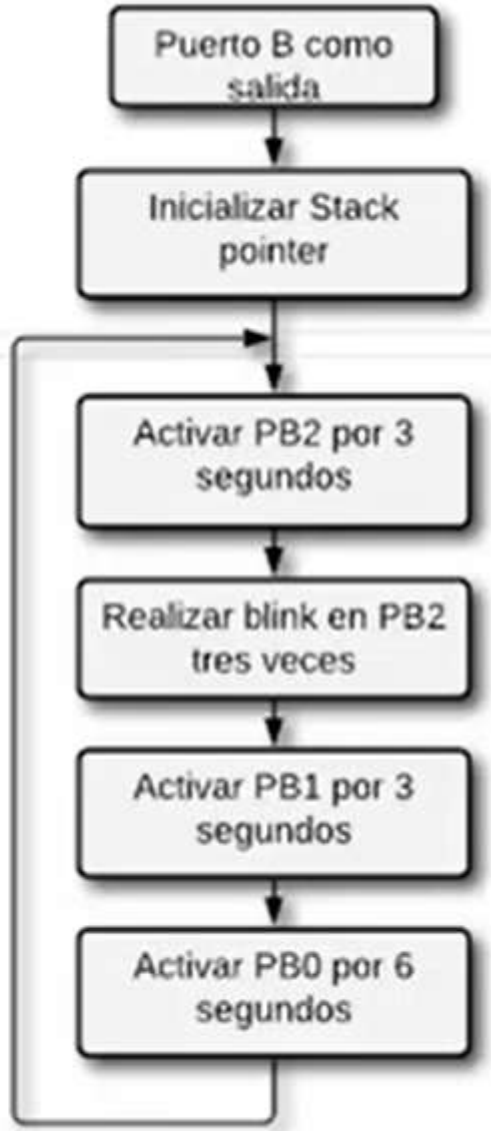


```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF   0X00
        OUT     DDRB, R16   0x01  ; Configura puerto B como salida.
        LDI     R17, 0x08   0x02
        OUT     SPL, R16    0x03
        OUT     SPH, R17    0x04  ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2    0x05  ; Activa LED verde.
        RCALL   Tseg        0x06
        CBI     PORTB, 2    0x07  ; Desactiva LED verde.
```

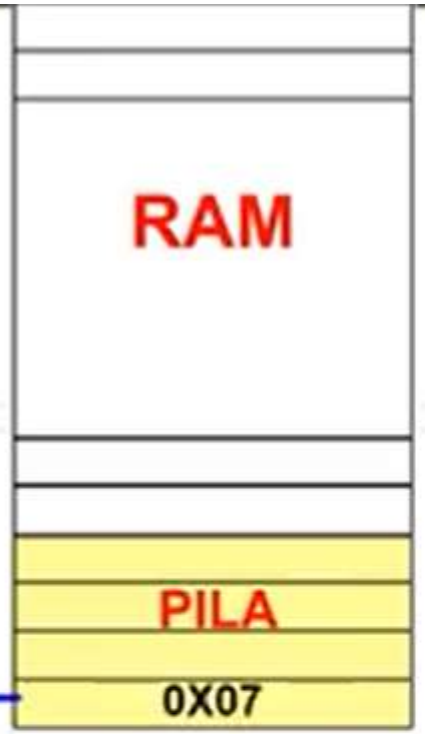


0X06

PC

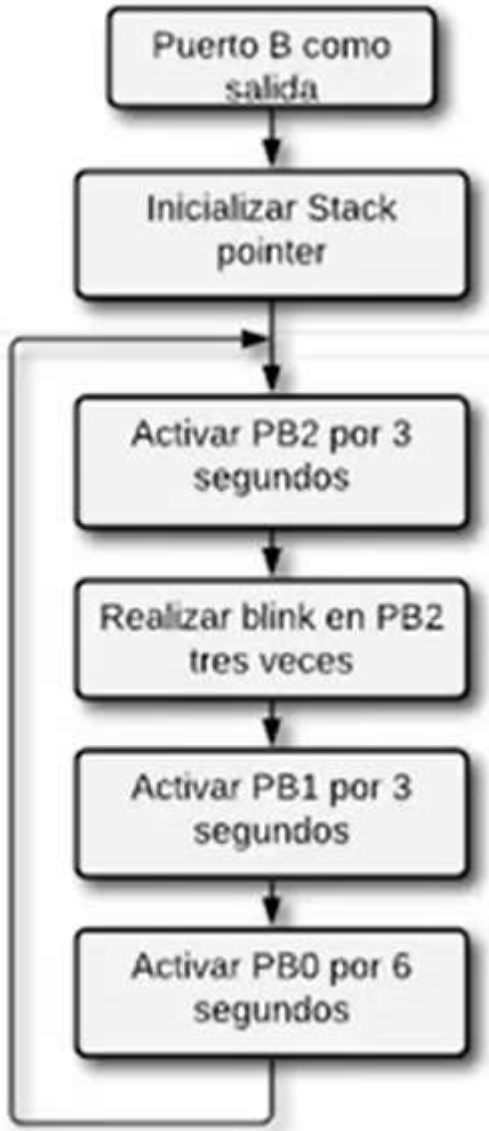


```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF   0X00
        OUT     DDRB, R16   0x01  ; Configura puerto B como salida.
        LDI     R17, 0x08   0x02
        OUT     SPL, R16    0x03
        OUT     SPH, R17    0x04  ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2    0x05  ; Activa LED verde.
        RCALL   Tseg        0x06
        CBI     PORTB, 2    0x07  ; Desactiva LED verde.
```

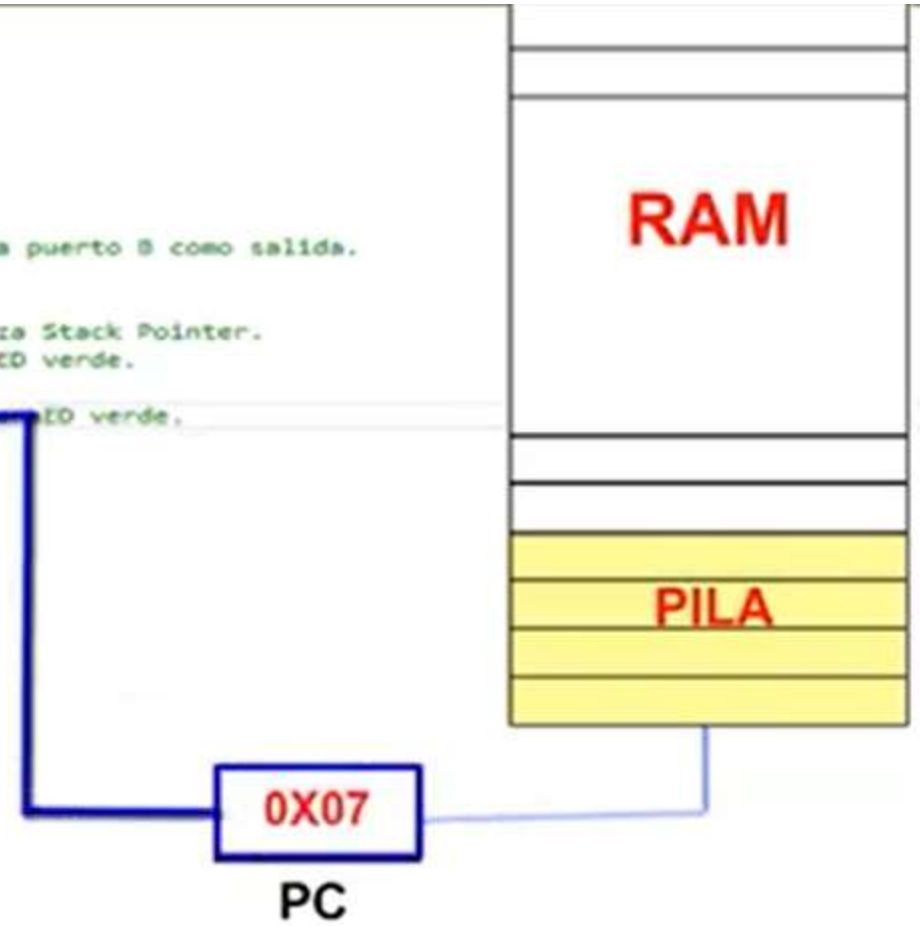


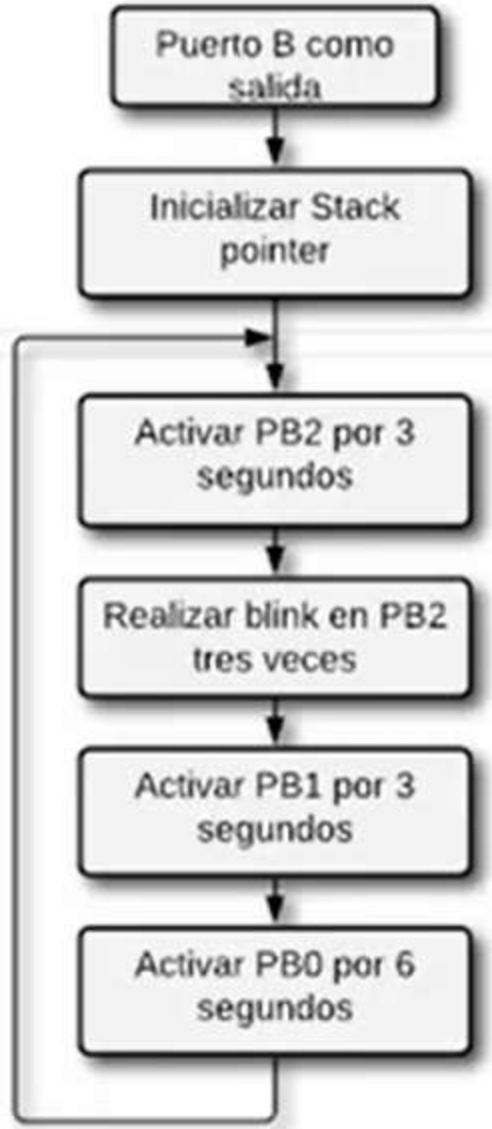
Dir. sub

PC



```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0xFF   0X00
        OUT     DDRB, R16   0x01  ; Configura puerto B como salida.
        LDI     R17, 0x08   0x02
        OUT     SPL, R16    0x03
        OUT     SPH, R17    0x04  ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2    0x05  ; Activa LED verde.
        RCALL   Tseg        0x06
        CBI     PORTB, 2    0x07  ; Desactiva LED verde.
```





```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ;
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 p.m.
; Author : USUARIO0
;

        LDI     R16, 0XFF
        OUT     DDRB, R16       ; Configura puerto B como salida.
        LDI     R17, 0X00
        OUT     SPL, R16
        OUT     SPH, R17        ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET
```

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET

Mseg:
```

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

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

El microcontrolador Atmega328p trabaja 1 Mhz.

AVR Delay Loop Calculator

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

10 MHz microcontroller clock frequency

0 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Stack Pointer	16 bits
RCALL	3 ciclos
RET	4 ciclos

AVR Delay Loop Calculator

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

10 MHz microcontroller clock frequency

7 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

AVR Delay Loop Calculator

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

10 MHz microcontroller clock frequency

7 cycles for rcall/ret or other overhead

18 first register to be used by delay loop

El registro a utilizar es r18

1 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

AVR Delay Loop Calculator

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

1.0 MHz microcontroller clock frequency

7 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

0.5 ns us ms s mins hrs days

8000000 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 8 000 000 cycles
; 1s at 8.0 MHz

    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

Tiempo de 0.5 segundos

AVR Delay Loop Calculator

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

1.0 MHz microcontroller clock frequency

7 cycles for `rcall/ret` or other overhead

18 first register to be used by delay loop

0.5 ns us ms s mins hrs days

499993 cycles go

assembler avr-gcc

```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 499 993 cycles
; 499ms 993us at 1.0 MHz

    ldi  r18, 3
    ldi  r19, 138
    ldi  r20, 84
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    nop
```

Copiar las instrucciones generadas

```
;
; Semáforo.asm
;
; Created: 16/03/2021 06:09:54 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.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET

Mseg:   ldi     r18, 3
        ldi     r19, 138
        ldi     r20, 84
    L1: dec     r20
        brne    L1
        dec     r19
        brne    L1
        dec     r18
        brne    L1
        nop
        RET
```

El código se pega en la etiqueta Mseg

AVR Delay Loop Calculator

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

10 MHz microcontroller clock frequency

7 cycles for rcall/ret or other overhead

18 first register to be used by delay loop

3 ns us ms s mins hrs days

2999993 cycles go

assembler avr-gcc

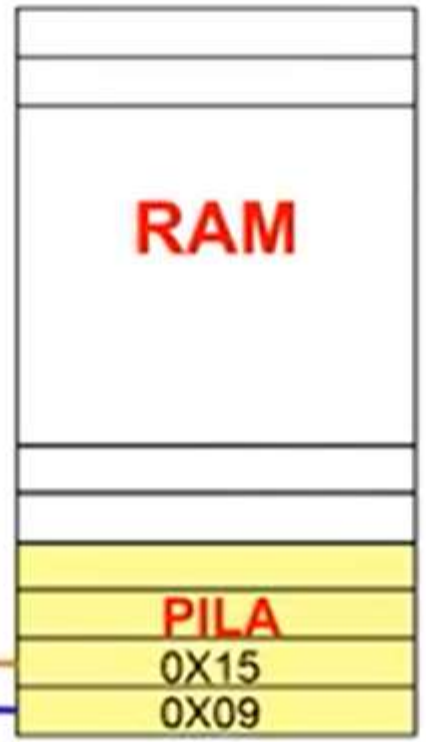
```
; Assembly code auto-generated
; by utility from Bret Mulvey
; Delay 2 999 993 cycles
; 2s 999ms 993us at 1.0 MHz

    ldi  r18, 16
    ldi  r19, 57
    ldi  r20, 12
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    nop
```

Cambiar el tiempo a 3 segundos

```
        OUT     SPH, R17        ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET

Mseg:   ldi     r18, 3
        ldi     r19, 138
        ldi     r20, 84
    L1: dec     r20
        brne    L1
        dec     r19
        brne    L1
        dec     r18
        brne    L1
        nop
        RET

Tseg:   ldi     r18, 16
        ldi     r19, 57
        ldi     r20, 12
    L2: dec     r20
        brne    L2
        dec     r19
        brne    L2
        dec     r18
        brne    L2
        nop
        RET
```

Agregar L2 en lugar de L1 y al final poner RET

```
        OUT     SPH, R17        ; Inicializa Stack Pointer.
Inicio: SBI     PORTB, 2        ; Activa LED verde.
        RCALL   Tseg
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET

Mseg:   ldi     r18, 3
        ldi     r19, 138
        ldi     r20, 84
    L1: dec     r20
        brne    L1
        dec     r19
        brne    L1
        dec     r18
        brne    L1
        nop
        RET

Tseg:   ldi     r18, 16
        ldi     r19, 57
        ldi     r20, 12
    L2: dec     r20
        brne    L2
        dec     r19
        brne    L2
        dec     r18
        brne    L2
        nop
        RET
```



```
        CBI     PORTB, 2        ; Desactiva LED verde.
        RCALL   Blink
        RCALL   Blink
        RCALL   Blink
        SBI     PORTB, 1        ; Activa LED amarillo.
        RCALL   Tseg
        CBI     PORTB, 1        ; Desactiva LED amarillo.
        SBI     PORTB, 0        ; Activa LED rojo.
        RCALL   Tseg
        RCALL   Tseg
        CBI     PORTB, 0
        RJMP    Inicio

Blink:  SBI     PORTB, 2
        RCALL   Mseg
        CBI     PORTB, 2
        RCALL   Mseg
        RET

Mseg:   ldi     r18, 3
        ldi     r19, 138
        ldi     r20, 84
    L1: dec     r20
        brne    L1
        dec     r19
        brne    L1
        dec     r18
        brne    L1
```

Output

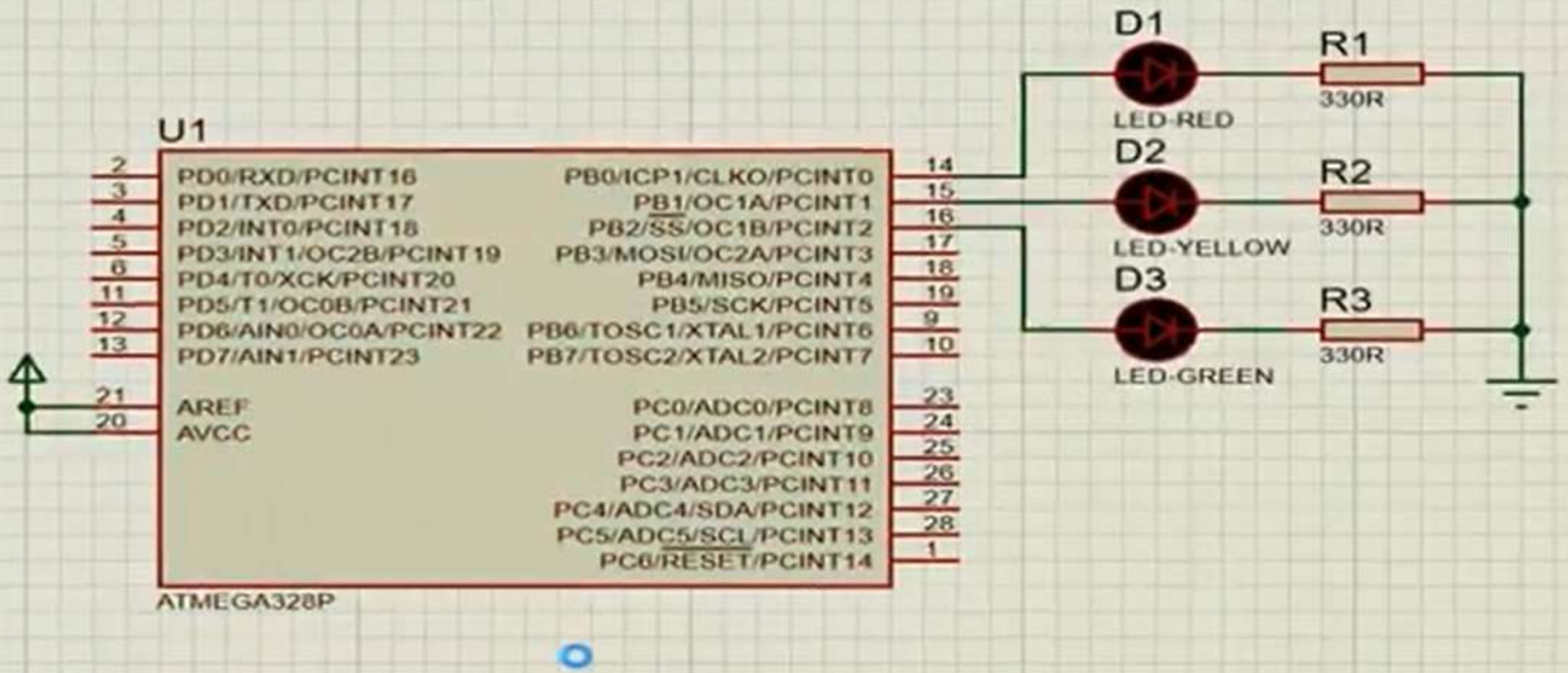
Show output from: Build

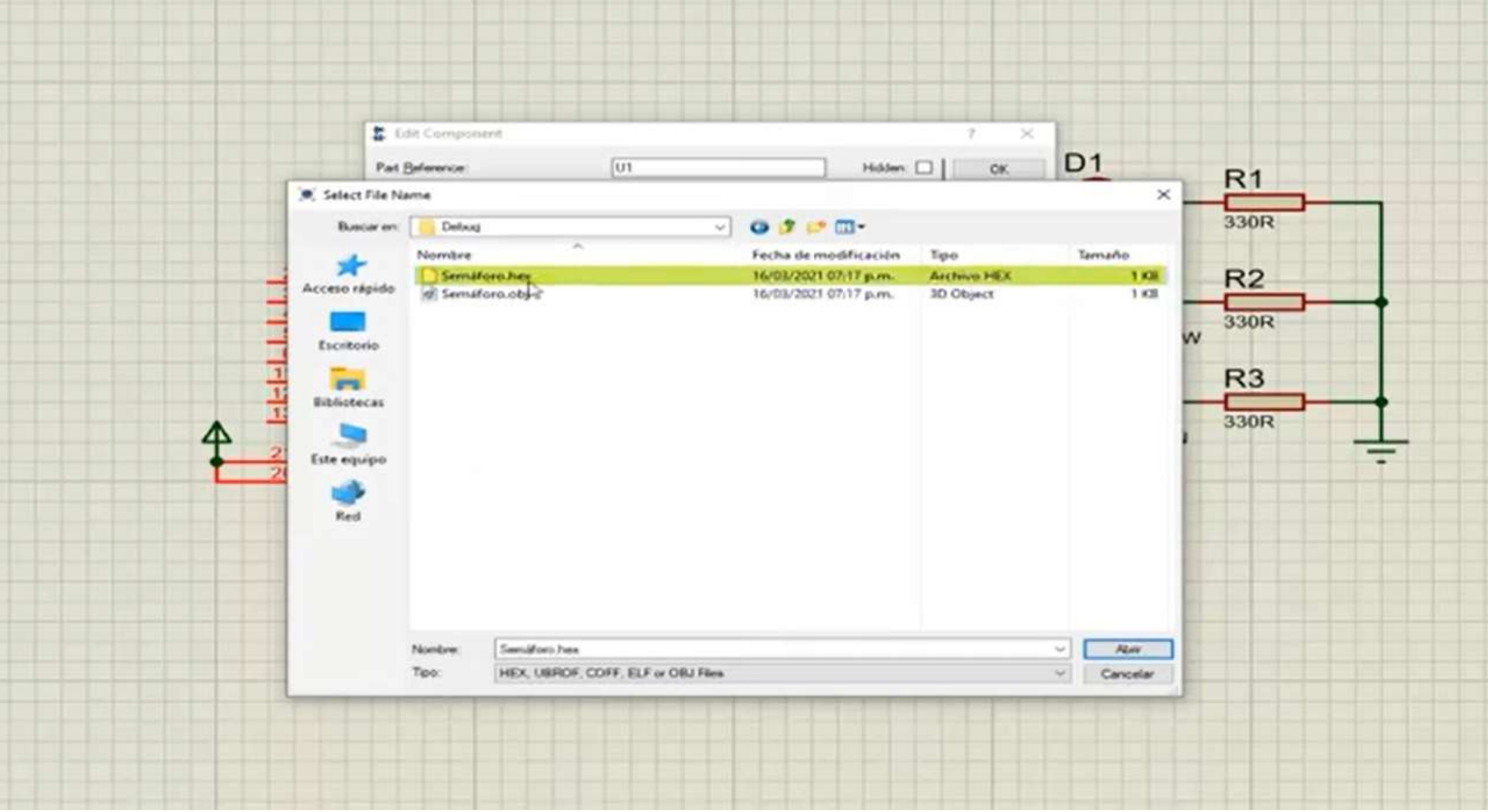
```
        [.cseg] 0x000000 0x00005c      92      0     92   32768   0.3%
        [.dseg] 0x000100 0x000100       0      0      0    2048   0.0%
        [.eseg] 0x000000 0x000000       0      0      0    1024   0.0%
        Assembly complete, 0 errors. 0 warnings
    Done executing task "RunAssemblerTask".
Done building target "CoreBuild" in project "Semáforo.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:\Semáforo\Semáforo\Semáforo.asmproj" (entry point):
Done building target "Build" in project "Semáforo.asmproj".
Done building project "Semáforo.asmproj".

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

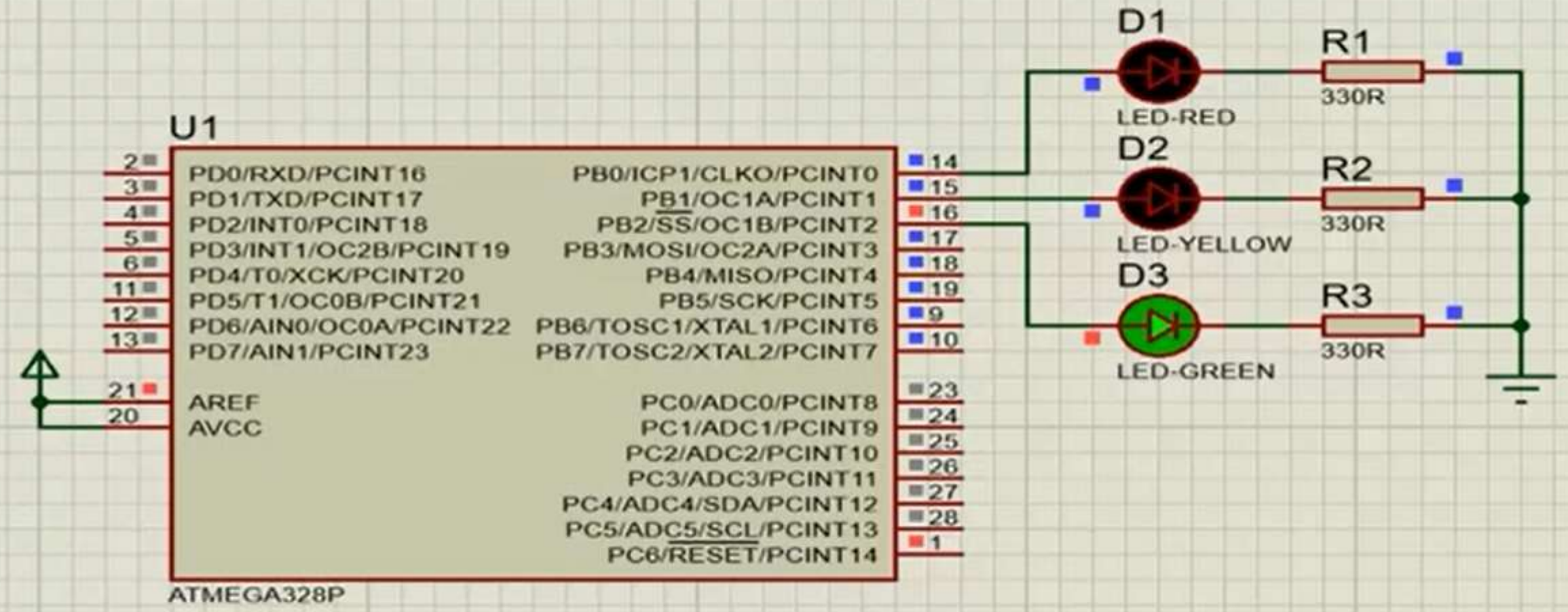
Output

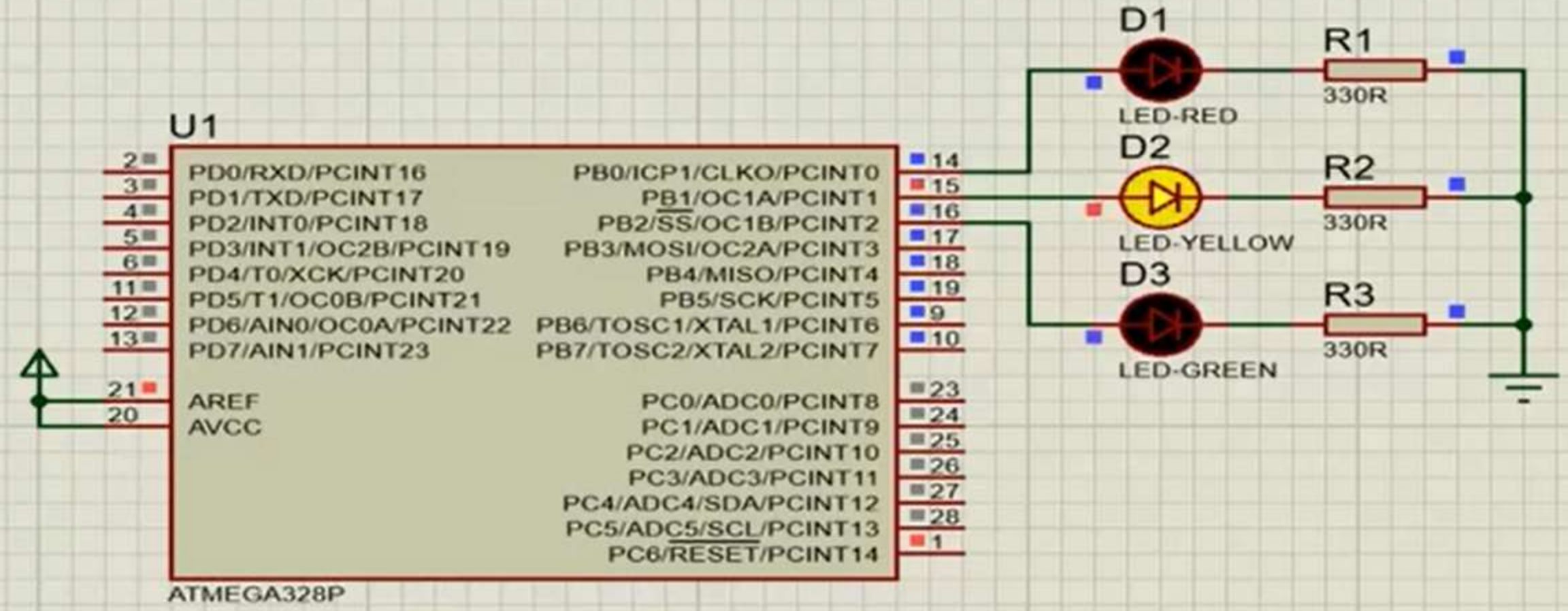
uild succeeded

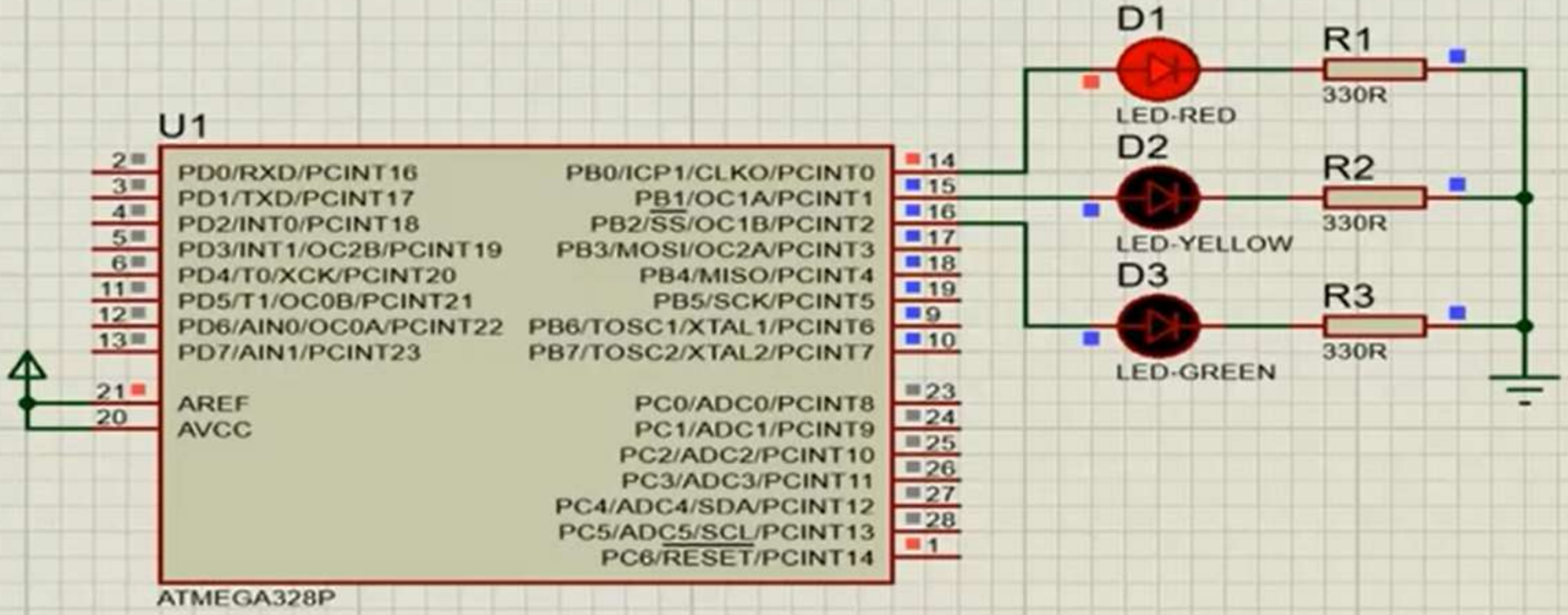
1:51 PM


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
D1
LED-RED
R1
330R
D2
LED-YELLOW
R2
330R
D3
LED-GREEN
R3
330R


Edit Component
Part Reference:
U1
Hidden:
OK
Select File Name
Buscar en:
Debug
Acceso rápido
Escritorio
Bibliotecas
Este equipo
Red
Nombre
Fecha de modificación
Tipo
Tamaño
Semáforo.hex
16/03/2021 07:17 p.m.
Archivo HEX
1 KB
Semáforo.obj
16/03/2021 07:17 p.m.
3D Object
1 KB
Nombre:
Semáforo.hex
Abrir
Tipo:
HEX, IBRDF, COFF, ELF or OBJ Files
Cancelar
D1
R1
330R
R2
330R
R3
330R


U1
2
3
4
5
6
11
12
13
21
20
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
14
15
16
17
18
19
9
10
23
24
25
26
27
28
1
ATMEGA328P
D1
LED-RED
R1
330R
D2
LED-YELLOW
R2
330R
D3
LED-GREEN
R3
330R


U1
2
3
4
5
6
11
12
13
21
20
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
14
15
16
17
18
19
9
10
23
24
25
26
27
28
1
ATMEGA328P
D1
LED-RED
R1
330R
D2
LED-YELLOW
R2
330R
D3
LED-GREEN
R3
330R


U1
2
3
4
5
6
11
12
13
21
20
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
14
15
16
17
18
19
9
10
23
24
25
26
27
28
1
ATMEGA328P
D1
LED-RED
R1
330R
D2
LED-YELLOW
R2
330R
D3
LED-GREEN
R3
330R

Lo que vimos:

- Concepto general de retardo.

- Consideraciones para realizar una rutina de retardo

- Método ciclos anidados y herramienta.

- Ejemplo de aplicación.

¡Muchas gracias!