

12^{AVR}

```
LDI R17,60      ;Se carga un registro con 60 que es el
                 ;valor de vueltas deseado
LDI R18,0        ;Se carga un registro para comparar con
                 ;cero

UN_MINUTO:
RCALL UN_SEGUNDO

INC R18
CP R18,R17       ;Esta instrucción CP compara el valor
                 ;obtenido con el deseado
BRNE UN_MINUTO   ;Esta línea ejecuta una etiqueta según
                 ;el resultado de la comparación anterior
                 ;CP (que es de Branch-If-Not-Equal "BRNE")
```

```
;ESTE PROGRAMA GENERA UN RETARDO DE 1-MINUTO USANDO
;UN CICLO-FOR QUE LLAMA 60 VECES A OTRA SUBROUTINA
;DE 1-SEGUNDO
```

```
.INCLUDE "TN2313DEF.INC"
.CSEG
.ORG 0
```

```
LDI R16, LOW(RAMEND)
OUT SPL, R16
```

```
LDI R17,60           ;Se carga un registro con 60 (para llamar
                    ;a UN_SEGUNDO)
LDI R18,0            ;Se carga un registro para comparar con
                    ;cero
```

```
UN_MINUTO:
RCALL UN_SEGUNDO
```

```
INC R18
CP R18,R17
BRNE UN_MINUTO
```

El registro R18 es incrementado en 1 y comparado con R17 que vale 60, al no ser igual en las primeras 59 vueltas, el cursor regresa a UN_MINUTO (el cual llama a UN_SEGUNDO 60 veces), una vez que se alcanza la igualdad R17=R18, entonces el cursor salta a FIN

```
FIN: RJMP FIN
```

También puede ser usada la instrucción Decrementar DEC en lugar de Incrementar INC, para tal caso R17 debería ser cero y R18 = 60

```
UN_SEGUNDO:
LDI R19,$FF
LDI R20,$3F
LDI R21,$10
LDI R22,0
```

```
CICLO1:
DEC R19
CP R19,R22
BRNE CICLO1
```

```
DEC R20
CP R20,R22
BRNE CICLO1
```

```
DEC R21
CP R21,R22
BRNE CICLO1
RET
```

```

UNA_HORA:
LDI R17,60      ;Se carga un registro con 60
LDI R18,0       ;Se carga un registro para comparar
                ;con cero

```

```

CICLO_1:
RCALL UN_MINUTO
DEC R17
CP R17,R18
BRNE CICLO_1

```

Ahora, el registro R17 es decrementado en 1 y comparado con R18 que vale 0, al no ser igual en las primeras 59 vueltas, el cursor regresa a CICLO_1 el cual llama a UN_MINUTO 60 veces, el cual a su vez llama a UN_SEGUNDO 60 veces

```

FIN: RJMP FIN

```

```

UN_MINUTO:
LDI R19,60      ;Se carga un registro con 60
LDI R20,0       ;Se carga un registro para comparar
                ;con cero

```

```

CICLO_2:
RCALL UN_SEGUNDO

```

```

DEC R19
CP R19,R20
BRNE CICLO_2
RET

```

Convertimos ahora a UN_SEGUNDO en una subrutina terminada con RET para que opere con la instrucción RCALL

```

UN_SEGUNDO:
LDI R20,$FF
LDI R21,$3F
LDI R22,$10
LDI R23,0

```

```

CICLO3:
DEC R20
CP R20,R23
BRNE CICLO3

```

```

DEC R21
CP R21,R23
BRNE CICLO3

```

```

DEC R22
CP R22,R23
BRNE CICLO3
RET

```

Esta subrutina es la nativa usada en el programa anterior, sin embargo, se le han hecho cambios a los registros, ya que de haber usado los registros R17, R18, y R19 se cruzarían con los valores de los registros cargados inicialmente con 60