

Programa:

```

;PROGRAMA PARA GENERAR EL DATO 0b1001_0100
;QUE SERÁ ENVIADO A UN RECUPERDOR DE RELOJ

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

LDI R16,LOW(RAMEND)
OUT SPL,R16

LDI R16,$01
OUT DDRB,R16      ;PARA ENVIAR EL DATO AL RECUPERADOR

;0b1001_0100
LDI R16,$01      ;DATO 1 a ENVIAR
LDI R17,$00      ;DATO 0 a ENVIAR

```

```

CICLO:
OUT PORTB,R16      ;Para Bit-7
RCALL DELAY

OUT PORTB,R17      ;Para Bit-6
RCALL DELAY

OUT PORTB,R17      ;Para Bit-5
RCALL DELAY

OUT PORTB,R16      ;Para Bit-4
RCALL DELAY

OUT PORTB,R17      ;Para Bit-3
RCALL DELAY

OUT PORTB,R16      ;Para Bit-2
RCALL DELAY

OUT PORTB,R17      ;Para Bit-1
RCALL DELAY

OUT PORTB,R17      ;Para Bit-0
RCALL DELAY
RJMP CICLO
    
```

```

DELAY:
LDI R20,$1C
LDI R22,0
    
```

```

CICLO2:
DEC R20
CP R20,R22
BRNE CICLO2
RET
    
```

Este DELAY se debe ajustar según el consumo de ciclos de reloj de la subrutina de detección en el "Recuperador de reloj" que es 30.25 µseg

Para el recuperador de reloj:

```

;PROGRAMA QUE EXTRAE Bit A Bit DEL DATO
;ENVIADO PARA GENERAR EL RELOJ Y GENERAR
;NUEVAMENTE EL DATO RECIBIDO

.INCLUDE "TN2313DEF.INC"
.CSEG
.ORG 0000

LDI R16,LOW(RAMEND)
OUT SPL,R16

LDI R16,0b0000_0011
OUT DDRB,R16

LDI R16,$00
OUT DDRD,R16      ;Sólo usaremos un pin para recibir los
                   ;datos

```

```

LEYENDO_BIT_A_BIT:
IN R16,PIND        ;R16 contiene el DATO a procesar

MOV R20,R16        ;COPIAMOS el DATO a R20 para evitar
                   ;CORRUPCIÓN

LDI R18,1
LDI R17,1

AND R17,R20
CP R17,R18
BREQ DATO_1_BIS
RJMP DATO_0_BIS

DATO_1_BIS:
LDI R18,0b0000_00_1_0

;0bXXXXX_XX D CLK
;0b0000_00 1 0
;0b0000_00 1 1

;CLOCK BAJA
OUT PORTB,R18      ;DATO SUBE

```

Este bloque debe ser revisado en el STOP WATCH para verificar el tiempo consumido para ajustar al Delay y cerrarlo a un tiempo conveniente. En nuestro caso es de 30.25 µseg

```

RCALL DELAY
LDI R18,0b0000_00_1_1
                                ;CLOCK SUBE
OUT PORTB,R18                  ;DATO SUBE

RCALL DELAY
RJMP LEYENDO_BIT_A_BIT

DATO_0_BIS:
LDI R18,0b0000_00_0_0

;0bXXXX_XX D CLK
;0b0000_00 0 0
;0b0000_00 0 1

                                ;CLOCK BAJA
OUT PORTB,R18                  ;DATO BAJO

RCALL DELAY

LDI R18,0b0000_00_0_1
                                ;CLOCK SUBE
OUT PORTB,R18                  ;DATO BAJO

RCALL DELAY

RJMP LEYENDO_BIT_A_BIT

```

```

DELAY:
LDI R20,$0B
LDI R22,0
CICLO:
DEC R20
CP R20,R22
BRNE CICLO
NOP
RET

```

Este DELAY se debe ajustar para ser sumado al bloque rojo del DELAY del "Generador".

```
;PROGRAMA QUE GENERA UN DATO DE 8-Bits CON UN ATtiny13  
;PARA SER CODIFICADOS EN MANCHESTER.  
;SE AJUSTÓ LA FRECUENCIA DE OSCILACIÓN CON EL  
;MÓDULO-PROGRAMADOR AVRISP
```

```
.INCLUDE "TN13DEF.INC"  
.CSEG  
.ORG 0
```

Observe cómo se escribe el .INCLUDE
para la librería del ATtiny13

314 CAPÍTULO 36 • GENERADOR DE PULSO DE RELOJ PARA SINCRONÍA (CLOCK RECOVERY)

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

```
LDI R16,$03
OUT DDRB,R16
```

```
;DELAY PARA EMPEZAR TODO
RCALL DELAY_256
```

```
INFINITO:
```

```
LDI R16,$01
OUT PORTB,R16      ;Bit DE ARRANQUE (START)

RCALL DELAY_256

LDI R16,$00
OUT PORTB,R16      ;SEPARACIÓN

RCALL DELAY_256    ;HASTA AQUÍ SE HAN CONSUMIDO 512 MICROSEG.
```

```
;EMPIEZA EL FRAME 1001_1010
```

```
LDI R16,$01
OUT PORTB,R16      ;DATO 1
```

```
RCALL DELAY_256
```

```
LDI R16,$00
OUT PORTB,R16      ;DATO 0
```

```
RCALL DELAY_256
```

```
LDI R16,$00
OUT PORTB,R16      ;DATO 0
```

```
RCALL DELAY_256
```

```
LDI R16,$01
OUT PORTB,R16      ;DATO 1
```

```
RCALL DELAY_256
```

```
LDI R16,$01
OUT PORTB,R16      ;DATO 1
```

```
RCALL DELAY_256
```

```
LDI R16,$00
OUT PORTB,R16      ;DATO 0
```

```
RCALL DELAY_256
```

```
LDI R16,$01
OUT PORTB,R16      ;DATO 1
```

```
RCALL DELAY_256
```

```
LDI R16,$00
OUT PORTB,R16      ;DATO 0
```

```
RCALL DELAY_256
```

```
;WAIT
RCALL DELAY_256
RCALL DELAY_256      ;PARA ESPERAR UN POCO ANTES DE LA NUEVA
RCALL DELAY_256      ;DETECCIÓN DE START EN EL CODIFICADOR
```

```
RJMP INFINITO
```

```
DELAY_256:
LDI R17,$23
LDI R18,0

SALTO1:
DEC R17
CP R17,R18
BRNE SALTO1
RET
```

Para el codificador Manchester:

```
;PROGRAMA PARA CODIFICAR MANCHESTER EN UN ATTINY2313
;SE GENERA CLOCK_RECOVERY
```

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

```
RJMP RESET
RJMP DETECTA_START
```

```

RESET:
LDI R16,LOW(RAMEND)
OUT SPL,R16

RCALL RETARDO_128_MICROS

LDI R16,$00
OUT DDRD,R16

LDI R16,$03                                ;Saca MANCHESTER (PB1) y
                                           ;CLK_ ;RECOVERY (PB0)
OUT DDRB,R16

LDI R20,0                                ;Para contar # de CONVERSIONES
                                           ;por frame
LDI R21,9                                ;Para comparar # de Bits
LDI R31,1                                ;Para comparar con DATO de
                                           ;entrada

DETECTA_START:
IN R30,PIND
AND R30,R31
CP R30,R31
BRNE DETECTA_START

RCALL RETARDO_256_MICROS                ;Tiempo de START
RCALL RETARDO_128_MICROS                ;Espacio para PRIMERA LECTURA

SEPARADOR_DE_FRAME:
RCALL RETARDO_256_MICROS                ;Tiempo del SEPARADOR

LEYENDO:
INC R20                                ;# DE LECTURAS
CP R20,R21                             ;R21=9, para comparar # de Bits
                                           ;a 8
BR EQ WAIT

IN R30,PIND
AND R30,R31
CP R30,R31                                ;R31=1
BR EQ SACA_1_MANCHESTER_Y_CLOCK
RJMP SACA_0_MANCHESTER_Y_CLOCK

```



```

SACA_1_MANCHESTER_Y_CLOCK:
;0bXXXX_XX D CLK
;0b0000_00 1 1
;0b0000_00 0 0

LDI R19,0b0000_00_1_1

                                ;CLOCK SUBE
OUT PORTB,R19                  ;DATO SUBE
RCALL RETARDO_128_MICROS

LDI R19,0b0000_00_0_0

                                ;CLOCK BAJA
OUT PORTB,R19                  ;DATO BAJA

RCALL RETARDO_128_MICROS

RJMP LEYENDO

SACA_0_MANCHESTER_Y_CLOCK:
;0bXXXX_XX D CLK
;0b0000_00 0 1
;0b0000_00 1 0

LDI R19,0b0000_00_0_1

                                ;CLOCK SUBE
OUT PORTB,R19                  ;DATO BAJA

RCALL RETARDO_128_MICROS

LDI R19,0b0000_00_1_0

                                ;CLOCK BAJA
OUT PORTB,R19                  ;DATO SUBE

RCALL RETARDO_128_MICROS

RJMP LEYENDO

WAIT:
LDI R20,0                      ;RESETEA el # de conversiones

```

```

;ÚLTIMA SEÑAL DE RELOJ:

;0bXXXX_XX D CLK
;0b0000_00 X 1
;0b0000_00 X 0

LDI R19,0b0000_000_1
                                ;CLOCK SUBE
OUT PORTB,R19                  ;DATO BAJA
RCALL RETARDO_128_MICROS

LDI R19,0b0000_000_0
                                ;CLOCK BAJA
OUT PORTB,R19                  ;DATO SUBE
RCALL RETARDO_128_MICROS

LDI R19,0b0000_0000
                                ;Pone a "0" la salida
OUT PORTB,R19                  ;tanto CLK como MANCHESTER

RCALL RETARDO_256_MICROS

RJMP DETECTA_START

;*****
;*****
RETARDO_128_MICROS:
LDI R17,$7D ;78..7E
LDI R18,0

SALTO2:
DEC R17
CP R17,R18
BRNE SALTO2
RET

RETARDO_256_MICROS:
LDI R17,$FD;FE
LDI R18,0

SALTO3:
DEC R17
CP R17,R18
BRNE SALTO3
RET

```

Esta sección es *opcional*.
Se colocó para obtener
un ciclo de reloj "extra" en
CLK_RECOVERY. Pero
puede ser eliminado