**COMUNICACIÓN PARALELA**

Puertos A, B, E y K.

Características comunes:

* Bidireccionales
* Programables pin a pin como entrada o salida.
* Tienen un registro para habilitar o deshabilitar “pull-resistor”[[1]](#footnote-1)
* Registro para habilitar o deshabilitar salida en baja potencia.
* Registro para programar modos de operación.

A continuación se presenta un diagrama a bloques de lo que es un pin de un puerto en el HC12:

Las direcciones de algunos puertos se presentan a continuación:

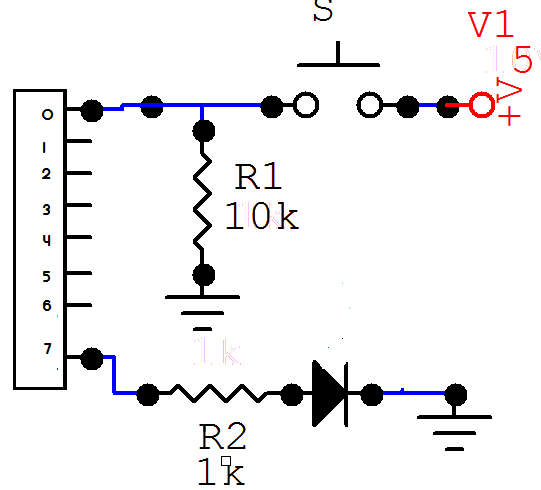
PORTA $0000

PORTB $0001

DDRA $0002

DDRB $0003

**EJEMPLO**: encendido de un led vía interruptor de contacto momentáneo.

Función and.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| \* | \* | \* | \* | \* | \* | \* | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Resultado = 0 => interruptor abierto (Z=1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| \* | \* | \* | \* | \* | \* | \* | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Resultado = 1 => interruptor cerrado (Z=0)

INICIO

Leer estado del interruptor

¿S cerrado? (Z=0)

SI

NO

Prende led

Apaga led

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 |  | ; carga el índice X de manera inmediata con 0000 |
|  |  | LDAA |  | #$80 |  | ; carga el acumulador A con 1000 0000h, bit7 a ‘1’ |
|  |  | STAA |  | DDRA, X |  | ; almacénalo en el registro DDRA |
|  |  |  |  |  |  |  |
| **REV** |  | LDAA |  | PORTA, X |  | ; carga el acumulador A con lo que lee en el puerto A |
|  |  | ANDA |  | #$01 |  | ; operación and, A con 01 |
|  |  | BNE |  | **PRENDE** |  | ; salta si Z=0 (a la etiqueta PRENDE |
|  |  | CLRA |  |  |  |  |
|  |  | STAA |  | PORTA, X |  |  |
|  |  | BRA |  | **REV** |  |  |
|  |  |  |  |  |  |  |
| **PRENDE** |  | LDAA |  | #$80 |  |  |
|  |  | STAA |  | PORTA, X |  |  |
|  |  | BRA |  | **REV** |  |  |

Otra forma seria programándolo con la instrucción BSET

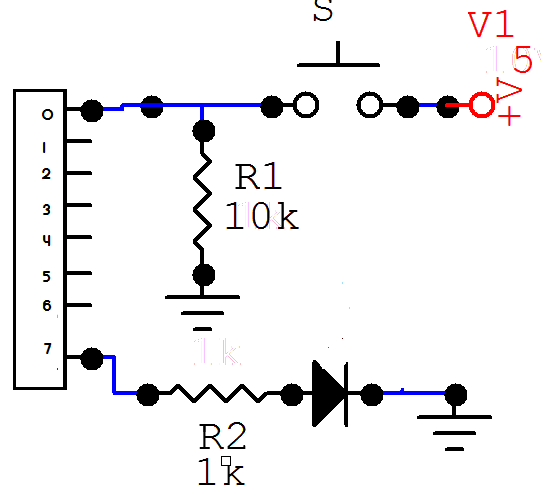
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 |  | ; carga el índice X de manera inmediata con 0000 |
|  |  |  |  |  |  |  |
|  |  | BSET |  | DDRA,X,$80 |  | ; bit set pone un ‘1’ en el bit7 del registro DDRA de manera indexada. **OJO, ya que esta instrucción solo pone un ‘1’ en el bit que le indiquemos, los demás bits quedan sin cambios** |
|  |  |  |  |  |  |  |
| **REV** |  | LDAA |  | PORTA, X |  | ; carga el acumulador A con lo que lee en el puerto A |
|  |  | ANDA |  | #$01 |  | ; operación and, A con 01 |
|  |  | BNE |  | **PRENDE** |  | ; salta si Z=0 a la etiqueta PRENDE |
|  |  | CLRA |  |  |  |  |
|  |  | STAA |  | PORTA, X |  |  |
|  |  | BRA |  | **REV** |  |  |
|  |  |  |  |  |  |  |
| PRENDE |  | LDAA |  | #$80 |  |  |
|  |  | STAA |  | PORTA, X |  |  |
|  |  | BRA |  | **REV** |  |  |

Otra manera seria como sigue, las instrucciones se reducen:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 |  | ; carga el índice X de manera inmediata con 0000 |
|  |  |  |  |  |  |  |
|  |  | BSET |  | DDRA,X,$80 |  | ; bit set pone un ‘1’ en el bit7 del registro DDRA de manera indexada. **OJO, ya que esta instrucción solo pone un ‘1’ en el bit que le indiquemos, los demás bits quedan sin cambios** |
|  |  |  |  |  |  |  |
|  |  | BRSET |  | PORTA,X,$01,**PRENDE** |  | ; Salta a la etiqueta PRENDE si hay un ‘1’ en el bit0 del puerto A, de manera indexada. |
|  |  | BCLR |  | PORTA,X,$80 |  | ; pone un ‘0’ en el bit7 del puerto A de manera indexada. |
|  |  | BRA |  | REV |  | ; salta a la etiqueta REV |
|  |  |  |  |  |  |  |
| **PRENDE** |  | BSET |  | PORTA,X,$80 |  | ; pone un ‘1’ en el bit7 del puerto A de manera indexada. |
|  |  | BRA |  | REV |  |  |

**PROGRAMA (TOGGLE)**

Encender el led y apagar el led cada vez que se presione una vez el interruptor. El puerto A es el que se muestra:



INICIO

Programar puerto A

Leer el estado del interruptor

¿‘s’ cerrado?

Encender led

¿‘s’ aún cerrado?

NO

SI

SI

NO

¿‘s’ cerrado?

NO

SI

apagar led

NO

SI

¿‘s’ aún cerrado?

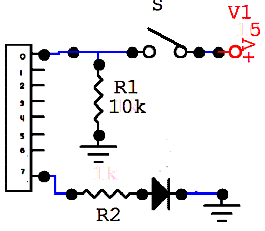
i

**CODIGO[[2]](#footnote-2)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 |  | ; carga el índice X de manera inmediata con 0000 |
|  |  |  |  |  |  |  |
|  |  | BSET |  | DDRA,X,$80 |  | ; bit set pone un ‘1’ (entrada) en el bit7 del registro DDRA de manera indexada. **OJO, ya que esta instrucción solo pone un ‘1’ en el bit que le indiquemos, los demás bits quedan sin cambios** |
|  |  | BCLR |  | DDRA,X,$01 |  | ; bit clear pone un ‘0’ (salida) en el bit0 del registro DDRA de manera indexada. |
|  |  |  |  |  |  |  |
| **LEER** |  | BRCLR |  | PORTA,X,$01,**LEER** |  | ; Salta si hay un ‘0’ a la etiqueta LEER en el bit0 del puerto A, de manera indexada. |
|  |  | LDAA |  | #$80 |  |  |
|  |  | STAA |  | PORTA,X |  |  |
|  |  |  |  |  |  |  |
| **AUN** |  | BRSET |  | PORTA,X,$01,**AUN** |  | ; Salta si hay un ‘1’ a la etiqueta AUN en el bit0 del puerto A, de manera indexada. |
| **CERRADO** |  | BRCLR |  | PORTA,X,$01,**CERRADO** |  | ; Salta si hay un ‘0’ a la etiqueta CERRADO en el bit0 del puerto A, de manera indexada. |
|  |  | LDAA |  | #$00 |  |  |
|  |  | STAA |  | PORTA,X |  |  |
| **AUN2** |  | BRSET |  | PORTA,X,$01, **AUN2** |  | ; Salta si hay un ‘0’ a la etiqueta AUN2 en el bit0 del puerto A, de manera indexada. |
|  |  | BRA |  | **LEER** |  | ; salta a la etiqueta leer. |

**PROBLEMA**

Desarrollar un programa que haga parpadear un led cuando el interruptor ‘S’ de contacto sostenido se cierre.



INICIO

Programar puerto A

Iniciar con led off

Leer estado de interruptor

‘s’ cerrado?

NO

SI

Encender led

Tiempo 250ms

Apagar led

Tiempo 250ms

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 | |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 | |  | ;define pseudo operador |
|  |  | LDX |  | #$0000 | |  |  |
|  |  | LDAA |  | #$80 | |  |  |
|  |  | STAA |  | DDRA,X | |  |  |
|  |  |  |  |  | |  |  |
| **INICIA** |  | BCLR |  | PORTA,X,$80 | |  |  |
| **LEER** |  | BRCLR |  | PORTA,X$01,**INICIA** | |  |  |
|  |  | BSET |  | PORTA,X,$80 | |  |  |
|  |  | JSR |  | **RETARDO** | | (4) | Para programar el retardo observe:  FFFFh=65535d  Por 5 ciclos veces que se realiza el lazo, se  Realizan un LDY, un DECA y BNE, el cual suman  6 ciclos de instrucción.  **4JSR+1LDAA+[6+5(65535)]∆+5RTS**  **esto debemos igualarlo A:**  **250 [ms]=250 OOO [µs]**  **250 000 [µs]={4JSR+1LDAA+[6+5(65535)]∆+5RTS}0.125[µs]**  **despejando de lA Ecuacion**  **∆={(250000/0.125)-4-1-5}/(6+5(65535))**  **∆=6.10347 en decimal**  **por lo tanto:**  **∆=6h** |
|  |  | BCLR |  | PORTA,X,$80 | |  |
|  |  | JSR |  | RETARDO | |  |
|  |  | BRA |  | **LEER** | |  |
|  |  |  |  |  | |  |
| **RETARDO** |  | LDAA |  | #$**∆** | | **(1)** |
| **LAZO** |  | LDY |  | #$FFFF | | **(2)** |
| **SUBLAZO** |  | NOP |  |  | **(5)** | **(1)** |
|  |  | DEY |  |  | **(1)** |
|  |  | BNE |  | **SUBLAZO** | **(3)** |
|  |  | DECA |  |  | | **(1)** |
|  |  | BNE |  | **LAZO** | | **(3)** |
|  |  | RTS |  |  | | **(5)** |

Si deseamos 50[ms]

**∆={(50000/0.125)-4-1-5}/(6+5(65535))**

**∆=1.220**

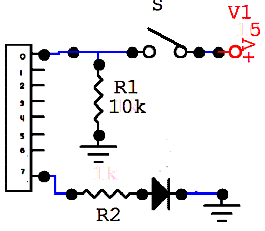
**∆=1H**

¿Qué pasa si deseamos tener un retardo de 50 [ms]?, hay diferentes formas de hacerlo, como nos damos cuenta es la quinta parte del tiempo de retardo del ejemplo anterior, por lo que el cálculo quedaría de la siguiente manera:

**[µs]={4JSR+1LDAA+[6+(65535)]∆+5RTS}0.125[µs]**

**PROBLEMA**

Desarrollar un programa que encienda el led durante 5 segundos cada vez que se cierra el interruptor ‘s’ de contacto sostenido.



º

INICIO

Programar puerto A

Iniciar con led off

Leer estado de interruptor

‘s’ cerrado?

NO

SI

Encender led

Tiempo 5s

Apagar led

‘s’ aún cerrado?

NO

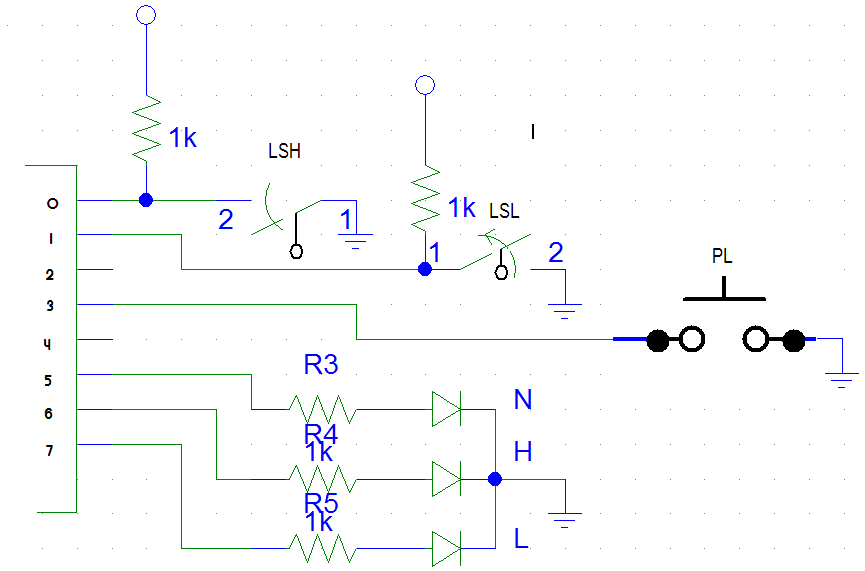
SI

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 | |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 | |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 | |  |  |
|  |  | LDAA |  | #$80 | |  |  |
|  |  | STAA |  | DDRA,X | |  |  |
|  |  |  |  |  | |  |  |
|  |  | BCLR |  | PORTA,X,$80 | |  |  |
| **LEER** |  | BRCLR |  | PORTA,X,$01,**LEER** | |  |  |
|  |  | BSET |  | PORTA,X,$80 | |  |  |
|  |  | JSR |  | **RETARDO5S** | | (4) | Para programar el retardo observe:  FFFFh=65535d  Por 5 ciclos veces que se realiza el lazo, se  Realizan un LDY, un DECA y BNE, el cual suman  6 ciclos de instrucción.  **4JSR+1LDAA+[6+5(65535)]∆+5RTS**  **esto debemos igualarlo A:**  **5000[ms]=5 000 OOO [µs]**  **5 000 000 [µs]={4JSR+1LDAA+[6+5(65535)]∆+5RTS}0.125[µs]**  **despejando de lA Ecuacion**  **∆={(5000000/0.125)-4-1-5}/(6+5(65535))**  **∆=122.07 en decimal**  **por lo tanto:**  **∆=7Ah** |
|  |  | BCLR |  | PORTA,X,$80 | |  |
| **AUN** |  | BRSET |  | PORTA,X,$01,**AUN** | |  |
|  |  | BRA |  | **LEER** | |  |
|  |  |  |  |  | |  |
| **RETARDO5S** |  | LDAA |  | #$**∆** | | **(1)** |
| **LAZO** |  | LDY |  | #$FFFF | | **(2)** |
| **SUBLAZO** |  | NOP |  |  | **5** | **(1)** |
|  |  | DEY |  |  | **(1)** |
|  |  | BNE |  | **SUBLAZO** | **(3)** |
|  |  | DECA |  |  | | **(1)** |
|  |  | BNE |  | **LAZO** | | **(3)** |
|  |  | RTS |  |  | | **(5)** |

En caso de que no nos alcanzara el conteo del acumulador A, utilizaríamos alternativamente el acumulador B, pero esto a su vez, implicaría ocupar todos los registros disponibles para programar un solo retardo, estos ejercicios se realizan para practicar con las instrucciones del hc12, más sin embargo, el microcontrolador cuenta con contadores específicos, que se verán más adelante.

**PROBLEMA**

Realizar un programa que genere señales de alarma independientes por nivel alto o nivel bajo en el tanque, mediante el parpadeo de los leds correspondientes, si el nivel es normal, el led N debe prender, en condiciones de alarma se debe apagar,(prueba de leds), se cierra en cualquier tiempo y condición, los tres leds deben prender y permanecer así, hasta que el interruptor PL se abra para regresar al estado anterior.



INICIO

Programar puerto A

Leer estado de interruptores

PL cerrado ?

Prender los tres leds

SI

NO

Prender N, y apagar H y L

Nivel normal ?

Nivel alto

?

SI

NO

SI

Prender H, y apagar N y L

Apagamos N y H

Prende L

Tiempo

Apaga L

Tiempo

Tiempo

Apaga H

Tiempo

NO

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PORTA |  | EQU |  | $00 | |  | ;define pseudo operadores |
| DDRA |  | EQU |  | $02 | |  | ; define pseudo operador |
|  |  | LDX |  | #$0000 | |  |  |
|  |  | LDAA |  | #$E0 | |  | ;programa puerto A |
|  |  | STAA |  | DDRA,X | |  | ;programa puerto A |
|  |  |  |  |  | |  |  |
| **LEER** |  | LDAA |  | PORTA,X | |  | ;carga el acum, con el contenido de A |
|  |  |  |  |  | |  |  |
| **REG** |  | BRSET |  | PORTA,X,$08,**PRUEBA** | |  | ; |
|  |  | ANDA |  | #$03 | |  | ; AND con 03h y modifica A |
|  |  | LDAA |  | PORTA,X | |  | ; volvemos a cargar A, debido a que ANDA lo modifica |
|  |  | CMPA |  | #$03 | |  | ; |
|  |  | BEQ |  | **NORMAL** | |  | ; salta a NORMAL, si (A)-03=0; Z=1 |
|  |  | BRCLR |  | PORTA,X,$01,ALTO | |  |  |
|  |  | BCLR |  | PORTA,X,$60 | |  | ; apaga N y H |
|  |  | BSET |  | PORTA,X,$80 | |  |  |
|  |  | JSR |  | **RETRASO** | |  |  |
|  |  | BCLR |  | PORTA,X,$80 | |  |  |
|  |  | JSR |  | **RETRASO** | | ;(4) |  |
|  |  | BRA |  | **LEER** | |  |  |
|  |  |  |  |  | |  |  |
| **PRUEBA** |  | BSET |  | PORTA,X,$E0 | |  |  |
|  |  | BRA |  | **REG** | |  |  |
|  |  |  |  |  | |  |  |
| **NORMAL** |  | BCLR |  | PORTA,X,$C0 | |  |  |
|  |  | BSET |  | PORTA,X,$20 | |  |  |
|  |  | BRA |  | **LEER** | |  |  |
|  |  |  |  |  | |  |  |
| **ALTO** |  | BCLR |  | PORTA,X,$CO | |  |  |
|  |  | BSET |  | PORTA,X,$A0 | |  |  |
|  |  | JSR |  | **RETRASO** | |  |  |
|  |  | BCLR |  | PORTA,X,$40 | |  |  |
|  |  | JSR |  | **RETRASO** | |  |  |
|  |  | BRA |  | **LEER** | |  |  |
|  |  |  |  |  | |  | Para programar el retardo observe:  FFFFh=65535d  Por 5 ciclos veces que se realiza el lazo, se  Realizan un LDY, un DECA y BNE, el cual suman  6 ciclos de instrucción.  **4JSR+1LDAA+[6+5(65535)]∆+5RTS**  **esto debemos igualarlo A:**  **250[ms]=250 OOO [µs]**  **250 000 [µs]={4JSR+1LDAA+[6+5(65535)]∆+5RTS}0.125[µs]**  **despejando de lA Ecuacion**  **∆={(250000/0.125)-4-1-5}/(6+5(65535))**  **∆=6.10347 en decimal**  **por lo tanto:**  **∆=6h** |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
| **RETRASO** |  | LDAA |  | #$**∆** | | **(1)** |
| **LAZO** |  | LDY |  | #$FFFF | | **(2)** |
| **SUBLAZO** |  | NOP |  |  | **5** | **(1)** |
|  |  | DEY |  |  | **(1)** |
|  |  | BNE |  | **SUBLAZO** | **(3)** |
|  |  | DECA |  |  | | **(1)** |
|  |  | BNE |  | **LAZO** | | **(3)** |
|  |  | RTS |  |  | | **(5)** |

1. Pull-up resistor: resistencia conectada a Vcc

   pull-down resistor: resistencia conectada a tierra. [↑](#footnote-ref-1)
2. Cada ciclo (~) es típicamente de 125[ns] para un bus de 8[MHz] (oscilador de 16[MHz]) [↑](#footnote-ref-2)