COMENTARIO TÉCNICO

Buceando en el HC908.....

Por Ing. Daniel Di Lella
Dedicated Field Application Engineer
www.edudevices.com.ar
dilella@arnet.com.ar

www.edudevices.com.ar

Como implementar un control remoto por infrarrojo en los microcontroladores HC908.

2da. Parte.

Continuando con el proyecto del control remoto por infrarrojo abordaremos el programa decodificador de la sección del receptor que correrá en nuestro microcontrolador HC908.

El programa decodificador de los pulsos infrarrojos modulados en tiempo (ancho de pulso) está basado en la función "Input Capture" que presenta el módulo del Timer en los HC908. La función Input Capture nos ayudará a hacer más sencillas las rutinas de decodificación de "1" (unos) o "0" (ceros) y el string de datos así conformado, ya que se podrá medir con precisión los distintos anchos de pulsos recibidos por el módulo receptor de infrarrojos.

La rutina implementada podrá formar parte del programa de aplicación del usuario, no requiriendo un trabajo constante de atención del CPU, ya que la misma solo se activará cuando se detecte actividad en el pin de salida del módulo receptor de infrarrojo IRM – 8601S.

Podemos resumir las tareas de la rutina de la siguiente forma:

- Habilitación de interrupción por IRQ (cuando en el pin de IRQ aparezcan pulsos con flancos negativos, el 1er pulso disparará todas las otras actividades)
- Seteo del módulo de Timer para un periodo de 100 ms e interrupción por Timer Overflow (Time Out).
- Seteo inicial del Canal cero (TCH0) con la función "Input Capture" y flanco descendente.
- Medición del ancho del pulso detectado.

- Determinación de “1”, “0”, o pulso erróneo detectado.
- Almacenamiento de “bits” recibidos.
- Detección de “Header” y Datos en el String.
- Confirmación de los datos por sucesivas comparaciones.
- Fin de la decodificación por “Time Out”.

A continuación detallaremos la rutina a implementar.

```
*********************************************************************
* Rutina de decodificación de pulsos infrarrojos por PWM utilizados para            *
* sistema de control remoto por I.R.                                                *
*                                                                                   *
*  En este proyecto se utilizará un MCU MC68HC908JL3CP (DIP 28 PINES)               *
*  Pero se podrá aplicar a cualquier MCU de la familia HC908 !!                     *
*                                                                                   *
*  DEFINICIONES:                                                                    *
*                                                                                   *
*  PUERTOS:                                                                         *
*                                                                                   *
*  PORTB ---> A disposición del usuario                                             *
*                                                                                   *
*  PORTA -->  A disposición del usuario                                             *
*                                                                                   *
*  PORTD --> PTD0 a PTD3 y de PTD5 a PTD7 a disposición del usuario                 *
*                                                                                   *
*   PTD4 --> INPUT -> INPUT CAPTURE -> DATOS MODULO INFRARROJO                      *
*                        MODULO RECEPTOR I.R. IRM8601S (EVERLIGHT)                  *
*                                                                                   *
*                                                                                   *
*  INTERRUPCIONES :                                                                 *
*                                                                                   *
*  EXTERNAS ---->  Por TIMER CHANNEL 0 cuando utilizo el modo ICAP                  *
*                              para capturar el ancho de los pulsos del modulo I.R  *
*                                                                                   *
*         ---> Por PIN IRQ EXTERNO,para ACTIVAR la funcion ICAP cuando              *
*              se recibe una señal a la salida del modulo de I.R.                   *
*                                                                                   *
*    Frecuencia de operacion --> Fbus = 2,4576 Mhz .                                *
*    Frecuencia Oscilador XTAL --> FXTAL = 9,8304 Mhz                               *
*                                                                                   *
* El usuario puede utilizar cualquier frecuencia de Bus, para ello deberá adaptar los *
*  tiempos del timer a la frecuencia elegida.                                       *
*********************************************************************
```

```
***************
* EQUATES     *
***************


          base 10T               ; BASE DECIMAL POR DEFAULT

          include 'jl3regs.inc'    ; equates registros JL3/JK3/JK1

RAM           EQU  $0080       ; Comienzo de la RAM para el JL3
RomStart      EQU  $ED00       ; Comienzo de la ROM/FLASH para el programa
VectorStart   EQU  $FFDE       ; Comienzo tabla de vectores varios
T100msL       EQU  $00         ; Valor hexa LOW base de tiempo del TIMER 100 ms
T100msH       EQU  $1E         ; Valor hexa HIGH base de tiempo del TIMER 100 ms

*************************
* EQUATES A TOCAR POR  *
* el usuario !!!       *
*************************

NUremoto      EQU $55             ; Valor Equivalente del número de remoto
HEAD          EQU $0F             ; Valor Equivalente "encabezado" del string I.R.
UNO_MAX    EQU 130                ; Valor Equivalente de Pulso "1" Máximo (1700 uS)
UNO_MIN     EQU 84                ; Valor Equivalente de Pulso "1" Mínimo (1100 uS)
CERO_MAX     EQU 69               ; Valor Equivalente de Pulso "0" Máximo (900 uS)
CERO_MIN    EQU 23                ; Valor Equivalente de Pulso "0" Mínimo (300 uS)
N_BITS      EQU 12                ; Valor Equivalente de Nº de bits

*******************
* VARIABLES EN RAM *
*******************
   ORG RAM                       ; RAM para uso del programa gral.

TEMPA         RMB 1               ; M. temporal del ACC en RAM
TEMPXH        RMB 2               ; M. temporal de los reg. XH en RAM
AUX           RMB 1               ; Variable auxiliar multipropósito
DISPARO       RMB 1               ; Flag indic de tecla DISP / disp. act
IR_SHOOT      RMB 1               ; Flag indic de comando I.R. aceptado
OKIR          RMB 1               ; Flag O.K RX DATO Trans. x REMOTO
EFRAME        RMB 1               ; Flag de ERROR en Frame de I.R.
NFLANCO       RMB 1               ; Cdor. de Flancos pares / impares
NPULSO        RMB 1               ; Cdor. de PULSOS del I.R.
WpulseL       RMB 1               ; Valor de ancho de pulso LOW
WpulseH       RMB 1               ; Valor de ancho de pulso HIGH
DATOS         RMB 1               ; Valor de datos recibidos x I.R.
HEADER        RMB 1               ; HEADER recibido por I.R.
TunoL         RMB 1               ; Valor "T1" del pulso -->LOW
TunoH         RMB 1               ; Valor "T1" del pulso -->HIGH
TdosL         RMB 1               ; Valor "T2" del pulso -->LOW
TdosH         RMB 1               ; Valor "T2" del pulso -->HIGH
```

```
************************
* FLASH DE PROGRAMA  *
************************

   ORG RomStart                ;Czo del programa en FLASH

********************
* Config. del MCU      *
********************

START       MOV #$01,CONFIG1    ;Reg. CONFIG1->LVID=0,STOP=0,COPD=1
            MOV #$00,CONFIG2    ;Reg. CONFIG2->IRQPUD,LVIT0=0,LVIT1=0
            NOP
            NOP                  ; NOPs de delay aseguran config.
            NOP
            NOP
            NOP
            MOV #$3F,PTAPUE      ;R. PULL UP en PTA0/5 y PTA6EN = 0
            MOV #$FF,PORTA       ;PTA0 a PTA6  -> INPUTS
            MOV #$00,DDRA
            MOV #$FF,PORTA       ;Fuerzo unos al PORTA
            MOV #$00,PORTB       ;PORTB todo como salida
            MOV #$FF,DDRB        ;
            MOV #$00,PORTB       ;Fuerzo ceros al PORTB
            MOV #$00,PDCR        ;PORTD Control register
            MOV #$10,PORTD       ;PORTD->PTD4 ->IN, PTD0/7 ->OUT
            MOV #$EF,DDRD
            MOV #$10,PORTD       ;
            NOP             ;
            BSET 1,KBSCR         ;IMASKK=1 -> MASK KBI, NO GLICHS
            NOP

********************
* HABILITO           *
* CONV. A/D          *
* DESHABILITO        *
* INTERRUPCION       *
* POR KBI            *
* HAB. IRQ EXT       *
********************

            MOV #$20,ADCLK   ;ADC Input Clock Reg. -> ADC clock / 2 (1mhz aprox.)
            MOV #$1F,ADSCR   ;ADC --> OFF
            NOP
            MOV #$06,INTSCR  ;ACK1/IMASK1=1 , MODE1=0 ->IRQ x fcos
            BCLR 1,INTSCR    ;HAB IRQ1->IRQ p/ MOD I.R.

            CLI              ;I MASK=0 --> INT's habilitadas
```

```
*******************
* SETEO INICIAL DE *
* VARIABLES        *
*******************


          CLR DISPARO         ;LIMPIO VARIABLES
          CLR IR_SHOOT         ;
          CLR AUX              ;
          CLR EFRAME          ;
          CLR OKIR             ;
          CLR NFLANCO          ;
          CLR NPULSO           ;
          CLR WpulseL          ;
          CLR WpulseH          ;
          CLR DATOS            ;
          CLR HEADER           ;
          CLR TunoL            ;
          CLR TunoH            ;
          CLR TdosL            ;
          CLR TdosH            ;


MAIN      WAIT                 ; Aquí el usuario pondrá su programa ppal.
          NOP                  ; el "wait" es solo a efectos ilustrativos....
          BRA MAIN             ;


*********************************************************************
*  INTERRUPCIONES :                                                 *
*                                                                   *
*  EXTERNAS ----->  Por TIMER CHANNEL 0 cuando utilizo el modo ICAP *
*              para capturar el ancho de los pulsos del modulo I.R  *
*                                                                   *
*         ----->  Por IRQ (IRQ + PTD4) por efecto de pulsos negativos*
*              provenientes del módulo receptor de Infrarrojos      *
*                                                                   *
*********************************************************************


*********************************************************************
* EXT_IRQ1 - SUB-RUTINA ATENCION DE INTERRUPCIONES EXTERNAS         *
*          IRQ1 + PTD4 --> x SENSOR INFRARROJO (MODULO)             *
*                                                                   *
*                                                                   *
* Si IRQ1 + PTD4 están en nivel low, entonces estoy en plena        *
* recepcion de señales del modulo INFRARROJO !!!                    *
*                                                                   *
* Si confirmo IRQ1 + PTD4 PARO el Timer(si estuviera corriendo)     *
* otras interrupciones en curso y la propia señal de IRQ1           *
* ya que solo se usará la funcion de INPUT CAPTURE CH0              *
* cuya rutina se llamará PULSO_CH0.... y en la presente rutina solo se seteará y *
* activará un TIMER de 100 mSeg. para generar un TIMER OVERFLOW SIN *
* INTERRUPCIONES (X POOLING) que me saque de la sub-rutina.....     *
*********************************************************************
```

```
EXT_IRQ1        BIL IRQ_OK      ;IRQ1 = 0 ? SI --> IRQ PRESENTE !!
EXTIRQ_VUEL     RTI             ;NO -> * RETORNO A PPAL. *


IRQ_OK          BSET 1,KBSCR    ;IMASKK=1 -> MASK KBI, NO GLICHS
                BSET 5,TSC      ;TSTOP=1 -> PARO TIMER !!
                BRCLR 4,PORTD,IRM_ACT1  ;PTD4=0? -> MODULO I.R. ACTIVO !!

                BRA EXTIRQ_VUEL     ;NO -> * RETORNO A PPAL. *


*********************************************
* PREPARO EL TIMER y el CHANNEL 0 para ICAP  *
* con un periodo de timer de 100 mS (Time Out)  *
*********************************************

IRM_ACT1        BSET 1,INTSCR       ;IMASK1=1 ->DESHABILITO IRQ1 !!
                CLI                 ;LPIO MASCARA GRAL INT'S
                                    ;Preparo Timer p/ Time Out = 100 ms
                                    ;e INPUT CAPTURE
                MOV #$35,TSC        ;TSTOP,TRST,PS2/PS0=1,TOIE=0 Psc=32
                MOV #$48,TSC0       ;MS0A/B=0,ELS0B=1,ELS0A=0,CHOIE=1
                NOP                 ;TCH0->INPUT CAPTURE / FALLING EDGE
                MOV #$00,TSC1       ;NO uso pines externos en el TIMER
                MOV #T100msH,TMODH  ;CARGO el Timer Module High
                MOV #T100msL,TMODL  ;CARGO el Timer Module Low ->$1E00
                LDA TSC             ;Leo el TSC para borrar el TOF...
                BCLR 7,TSC          ;LIMPIO TOF en el TSC...
                BCLR 5,TSC          ;TSTOP=0 -> ARRANCO TIMER !!

T_OVERFLOW      BRSET 7,TSC,ESCAPE! ;En loop hasta TOF ->TIMER OVERFLOW
                NOP                 ;esperando recibir el "frame" de datos
                NOP                 ;dentro de los 100 ms (Time Out)
                BRA T_OVERFLOW      ;Y luego salgo de esta SUB-RUTINA

ESCAPE!         CLR OKIR            ;Lpio Flags de sub-rutina TIMER CH0
                CLR NPULSO          ;
                CLR NFLANCO         ;
                CLR HEADER          ;
                CLR DATOS           ;
                BSET 1,KBSCR        ;IMASKK=1 -> MASK KBI, NO GLICHS
                MOV #$2F,KBIER      ;HAB KBIE5,3/0 ->SW1/4 Y E-D
                MOV #$06,KBSCR      ;LPIO INT. PEND. Y DISP. X FLANCOS
                BCLR 1,KBSCR        ;IMASK=0 -> UNMASK KBI-> HAB KBI
                MOV #$06,INTSCR     ;ACK1/IMASK1=1 , MODE1=0 ->IRQ x fcos
                BCLR 1,INTSCR       ;HAB IRQ1->IRQ p/ AJUSTE y MOD I.R.
                MOV #$76,TSC        ;TSTOP,TRST,PS2/PS0,TOIE=1
                MOV #$00,TSC0       ;NO uso pines externos en el TIMER
                MOV #$00,TSC1       ;

                CLI             ;LPIO MASCARA GRAL. INT'S
                JMP EXTIRQ_VUEL     ;Vuelvo a PPAL. !!!
```

```
*************************************************************************
* PULSO_CH0 - RUTINA DE DECODIFICACION DE PULSOS DESDE MODULO I.R.  *
*                                                                       *
* Esta rutina entra en funcionamiento, cuando hay actividad en IRQ1     *
* y en PTD4 al mismo tiempo. En ese caso luego de disparar un TIMER     *
* de OVERFLOW (100ms) seteo el TIMER CHANNEL 0 para INPUT CAPTURE       *
* y voy alternando entre FALLING EDGE y RAISING EDGE para medir el ancho de pulso *
* y determinar si es un cero o un uno lo detectado                      *
* Resolución del TIMER = 13 uS x cuenta                                 *
*                                                                       *
* VARIABLES UTILIZADAS:                                                 *
*                                                                       *
* EFRAME --> Flag de ERROR en el FRAME de I.R. recibido (TEST)          *
* OKIR ----> Flag de O.K en la recepcion DATO transmitido x Remoto      *
* NFLANCO -> Cdor. de Flancos pares / impares (Raising/Falling)         *
* NPULSO --> Cdor. de pulsos recibidos                                  *
* WpulseL -> Valor ancho de pulso LOW                                   *
* WpulseH -> Valor ancho de pulso HIGH                                  *
* DATOS ---> Valor en RAM de DATOS recibidos por I.R.                   *
* HEADER --> Valor en RAM del HEADER recibido por I.R.                  *
* NUremoto -> Valor Equate del Numero de Remoto                         *
* TunoL --> Valor "T1" del pulso --> LOW                                *
* TunoH --> Valor "T1" del pulso --> HIGH                               *
* TdosL --> Valor "T2" del pulso --> LOW                                *
* TdosH --> Valor "T2" del pulso --> HIGH                               *
*                                                                       *
* FLAGS DE SALIDA DE LA RUTINA:                                         *
*                                                                       *
* DISPARO --> FLAG de DISPARO HABILITADO x decodificación O.K. del      *
*          string de datos x infrarrojo    (flag utilizable por el usuario)  *
*                                                                       *
* IR_SHOOT --> Flag de decodificación O.K. del string de datos          *
*           (flag utilizable por el usuario)                            *
*                                                                       *
*************************************************************************


PULSO_CH0   INC NFLANCO                 ;Se incre. cada vez que se ingresa
            BRSET 0,NFLANCO,IMPAR_F ;SI NFLANCO=1,3,5.. Flanco impar
PAR_F       LDA TSC0                    ;Leo TSC0 para limpiar CH0F
            BCLR 7,TSC0                 ;Limpio CH0F
            MOV TCH0H,TdosH             ;TCH0H --> TdosH
            MOV TCH0L,TdosL             ;TCH0L --> TdosL
            MOV #$48,TSC0               ;TSC0 seteado para FALLING EDGE
            LDA TdosL                   ;Calculo el ancho del pulso
            SUB TunoL                   ;WpulseH/L = T2 - T1
            STA WpulseL                 ;WpulseL -> ancho del pulso LOW
            LDA TdosH                   ;
            SBC TunoH                   ;
            STA WpulseH                 ;WpulseH -> ancho del pulso HIGH
            LDA WpulseH                 ;Si Wpulso > 1700 uS -> ERROR !!
            CMP #$00                    ;sino sigo preg. x otros valores
            BEQ WPULSO?                 ;
            JMP Ferror                  ;FRAME ERROR !!

IMPAR_F     JMP IMPAR_F1                ;Salto largo a IMPAR_F1
```

```
WPULSO?     LDA WpulseL                 ;Wpulso < o = 1700 uS y > o = 1100
            CMP #UNO_MAX                ;SI -> el pulso es un "UNO"
            BLS P_uno?                  ;
            JMP Ferror                  ;FRAME ERROR !!

P_uno?      LDA WpulseL             ;si es UNO inc cdor de pulsos y
            CMP #UNO_MIN            ;GUARDO el UNO haciendo ROTATEs
            BHS P_uno!              ;Sino puede ser un CERO....
            CMP #CERO_MAX           ;Wpulso < o = 900 uS y > o = 300
            BLS P_cero?             ;
            JMP Ferror              ;FRAME ERROR !!

P_cero?     LDA WpulseL             ;si es CERO inc. cdor de pulsos y
            CMP #CERO_MIN           ;GUARDO el CERO haciendo ROTATEs
            BHS P_cero!             ;
            JMP Ferror              ;FRAME ERROR !!

P_uno!      INC NPULSO              ;Un pulso más ...
            SEC                     ;SETEO EL CARRY....p/ rotar 1's
            ROL DATOS               ;
            ROL HEADER              ;Voy desplazando 1s x DATOS y HEADER
P_FIN?      LDA NPULSO              ;NPULSO=#N_BITS -> FIN DE CAPTURA !
            CMP #N_BITS             ;
            BEQ ST_PULSO            ;NO -> ME VOY DE LA INTERRUPCION !!
            JMP PULSO_VUEL          ;Vuelvo a PPAL. !!!

P_cero!     INC NPULSO              ;Un pulso más ....
            CLC                     ;LPIO EL CARRY ...p/ rotar 0's
            ROL DATOS               ;
            ROL HEADER              ;Voy despalzando 0s x datos y HEADER
            BRA P_FIN?              ;NPULSO=#N_BITS -> FIN DE CAPTURA !

IMPAR_F1    LDA TSC0                ;Leo TSC0 para limpiar CH0F
            BCLR 7,TSC0             ;Limpio CH0F
            MOV TCH0H,TunoH         ;TCH0H --> TunoH
            MOV TCH0L,TunoL         ;TCH0L --> TunoL
            MOV #$44,TSC0           ;TSC0 seteado para RAISING EDGE
            JMP PULSO_VUEL          ;Vuelvo a PPAL. !!!


ST_PULSO    CLR NPULSO              ;LPIO Cdor de pulsos p/ nueva Rx
            LDA HEADER              ;HEADER=#HEAD ? -> HEADER CORRECTO !
            CMP #HEAD               ;sino lei cualquier cosa y me voy
            BEQ N_REMOTO?           ;
            BRA LPIO_HD             ;Voy a limpiar HEADER y DATOS

N_REMOTO?   LDA DATOS               ;DATOS = NUremoto ? ->COINCIDENCIA!
            CMP #NUremoto           ;sino NO es el numero correcto
            BEQ OK_IRx              ;
            BRA LPIO_HD             ;VOy a limpiar HEADER y DATOS

OK_IRx      INC OKIR                ;Cdor. de aciertos
            LDA OKIR                ;OKIR > o = 2 -> DISPARO x I.R !!
            CMP #$02                ;sino LPIO flags DISPARO, IR_SHOOT
            BHS DISP_XIR            ;
            CLR DISPARO             ;
            CLR IR_SHOOT            ;
LPIO_HD     CLR HEADER              ;
            CLR DATOS               ;
```

```
PULSO_VUEL    RTI               ;

DISP_XIR      BSET 0,DISPARO    ;DISPARO=1 -> DISPARO X I.R. !!
              BSET 0,IR_SHOOT   ;IR_SHOOT=1
              BRA LPIO_HD       ;Voy a limpiar HEADER y DATOS

Ferror        BSET 0,EFRAME     ;FRAME ERROR -> PARA TEST !!
              BRA PULSO_VUEL    ;Vuelvo a PPAL. !!



************************************************************
* INTERRUPT VECTORS TABLE -                                *
*                                                          *
* RESET / SWI / IRQ1 / ADC / Y NO ASIGNADOS ---> START     *
* KEYBOARD INTERRUPT VECTOR ---> START                     *
* TIM1 OVERFLOW VECTOR ---> TIMER_IRQ                      *
* TIM1 CHANNEL 0 VECTOR --> PULSO_CH0                      *
*                                                          *
************************************************************

   ORG  VectorStart

     dw  START         ; ADC Conversion Complete Vector
     dw  START         ; Keyboard Vector
     dw  START         ; (No Vector Assigned $FFE2-$FFE3)
     dw  START         ; (No Vector Assigned $FFE4-$FFE5)
     dw  START         ; (No Vector Assigned $FFE6-$FFE7)
     dw  START         ; (No Vector Assigned $FFE8-$FFE9)
     dw  START         ; (No Vector Assigned $FFEA-$FFEB)
     dw  START         ; (No Vector Assigned $FFEC-$FFED)
     dw  START         ; (No Vector Assigned $FFEE-$FFEF)
     dw  START         ; (No Vector Assigned $FFF0-$FFF1)
     dw  START         ; TIM1 Overflow Vector
     dw  START         ; TIM1 Channel 1 Vector
     dw  PULSO_CH0     ; TIM1 Channel 0 Vector
     dw  START         ; (No Vector Assigned $FFF8-$FFF9)
     dw  EXT_IRQ1      ; ~IRQ1
     dw  START         ; SWI Vector
     dw  START         ; Reset Vector
```

Continuará.....

www.edudevices.com.ar