

## COMENTARIO TÉCNICO

# *Buceando en el HC908.....*



Por Ing. Daniel Di Lella

Dedicated Distributor Field Application Engineer

For Freescale Semiconductors Products & Technical Consult

Dto. Técnico Electrocomponentes S.A.

[fae@electrocomponentes.com](mailto:fae@electrocomponentes.com)

[dilella@arnet.com.ar](mailto:dilella@arnet.com.ar)

## *Detector Óptico de Humo de Bajo consumo utilizando un HC908Q .... ..... “Los chiquitos poderosos de Freescale”*

### 3ra. Parte.

En los artículos anteriores, se vieron en detalle aspectos teóricos, diagramas de circuitos parciales y generales que nos dieron una visión global del proyecto. Ahora damos paso al “firmware” o programa que le da vida al sistema de detector de humo.

La figura 2 nos muestra el diagrama de flujo general del programa del detector de humo. La operación del MC68HC908QT4 es relativamente sencilla en este proyecto y los comentarios del programa “fuente” (smoke1.asm) nos aclararán aun más el funcionamiento. Sin embargo, comentaremos brevemente detalles del programa.

Luego de superar el estado de “power on” se configuran adecuadamente los registros CONFIG1 y CONFIG2, el modulo de Auto Wakeup , el ADC y los puertos I/O. A continuación, el microcontrolador entra en una cadena de modos stop (20 STOPS de 500 mS cada uno). Esto posibilita que el circuito de manejo de energía “cargue” al resto del detector para que este funcione correctamente.

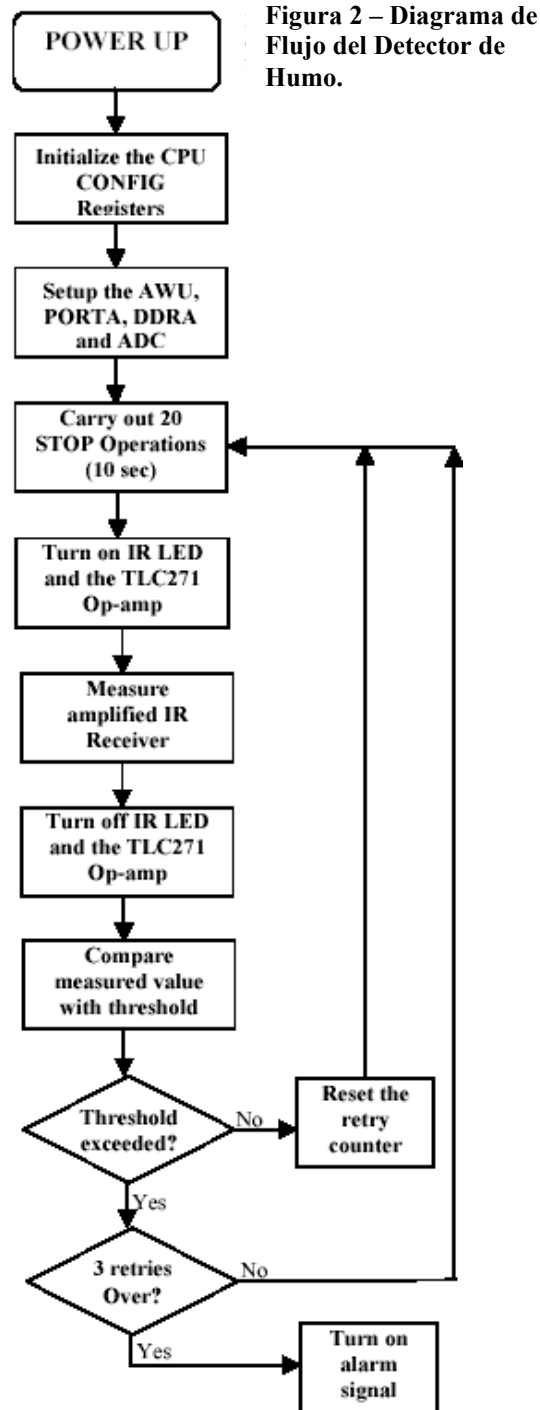
Lo próximo es la fase de medición. El LED IR y el amplificador operacional son encendidos y luego de 1 mS de demora luego de alguna posible reflexión de alguna partícula de humo, la señal amplificada proveniente del receptor es recibida y digitalizada.

Una vez que la digitalización de la medición se completa, el LED IR y el amplificador se apagan.

El valor obtenido del ADC luego es comparado con un valor de “disparo” prefijado.

Este valor de disparo podrá variar según el tipo de cámara de humo utilizada y la construcción de la misma. Para la cámara utilizada en el proyecto descrito, el valor de disparo es prefijado en 1V. Si el valor del ADC supera el umbral es una señal de posible humo y posterior peligro de incendio. Sin embargo, para prevenir las falsas alarmas, la señal de alarma solo se activa si 3 mediciones o más exceden el umbral en forma continua.

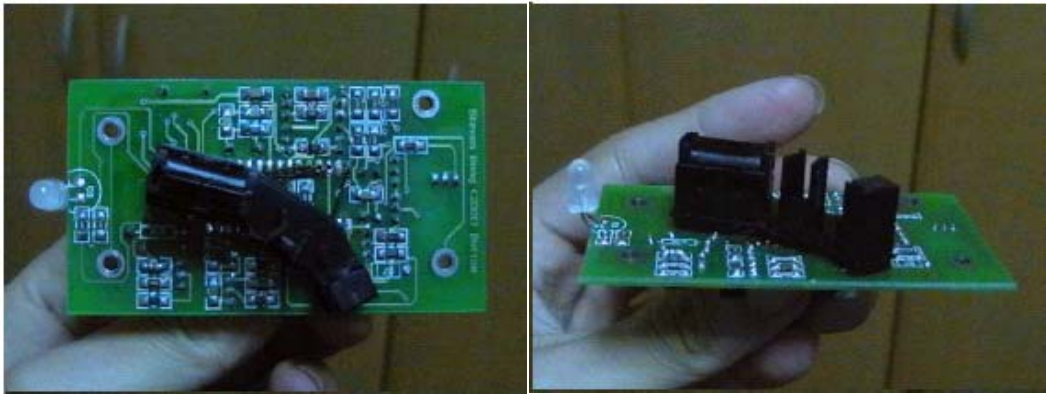
Después de cada fase de medición, una cadena de 20 operaciones STOP se ejecuta para permitir que la bomba de corriente se cargue.



## **Conclusión.**

Con el uso del microcontrolador de Freescale MC68HC908QT4, podemos construir un detector de humo de muy bajo consumo que puede usarse con paneles de incendio estandar. Las características de bajo consumo habilitan el uso de hasta 30 detectores de humo en cascada por canal de un panel de incendio típico.

## **Fotos del proyecto.**



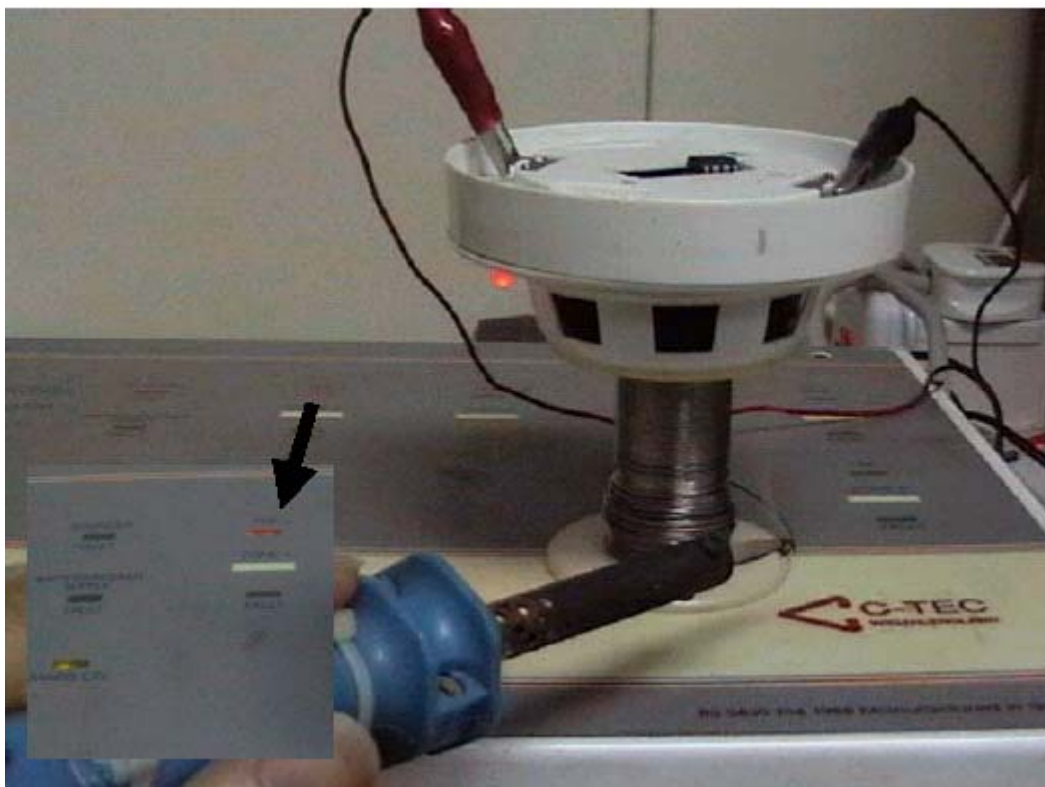
**Detalles de la cámara de humo en la placa.**



**Partes constitutivas del detector de humo.**



**Prueba del detector de humo por medio del humo emitido por un soldador al fundirse alambre de estaño en su proximidad.**



**En la foto se puede apreciar como se activa el LED indicador luego de 1 minuto o menos de iniciada la prueba y además como se activa la señal de alarma en el panel de incendio (flecha indicadora)**

```

;*****
;* DETECTOR OPTICO DE HUMO DE BAJO CONSUMO
;*
;*
;*AUTOR: WONG KAI JUAN, STEVEN
;*****

```

```

        Include 'qtqy_registers.inc' ; Para los QT1, QT2, QT4, QY1, QY2, QY4

```

```

RAMStart    EQU    $0080 ; Czo de las variables en RAM
ROMStart    EQU    $EE00 ; Czo de memoria de programa / constantes.

```

```

STOP_TIME      EQU    $14 ; La duración del stop = STOP_TIME * 500ms
ALARM_VALUE    EQU    $32 ; Tensión de Disparo

```

```

FIRE           EQU    %00010000 ; PTA4 para el control de la señal de alarma.
OP_AMP         EQU    %00000010 ; PTA1 para el control on/off amplif. OP.
LED            EQU    %00000001 ; PTA0 para el control on/off del IR LED

```

```

        ORG RAMStart

```

```

RETRY         DS 1

```

```

        ORG ROMStart

```

```

;*****
;* spurious – Rutina de Servicio de interrupción no esperada
;* (unwanted interrupt)
;*****

```

```

spurious:    ;
            NOP ;
            RTI

```

```

MAIN:

```

```

;****CONFIGURE CONFIG REGISTERS****
LDA          #%00111011
; COP disabled
; STOP enabled
;
; Disable all LVI functions
; Long Auto-wake up period (500ms)
STA          CONFIG1
LDA          #%10000000
; RST pin disabled
; Internal clock
; IRQ disabled
; No pull-up for IRQ pin
STA          CONFIG2

```

```

;****CONFIGURE AUTO-WAKEUP MODULE****

LDA  #00
STA  KBIER ; Solo se habilita AWUIE, NO otras KBI
LDA  #00 ; Clears IMASKK in KBSCR
STA  KBSCR

;****CONFIGURE I/O & ADC PORTS****

MOV  #00,PTAPUE ; No R de pull-ups
MOV  #00,DDRA ; Port A todo como entrada
MOV  #00,PORTA ;
MOV  #30,ADICK ; Bus Clock / 2

;****CONFIGURE INTERRUPTS****

MOV  #06,INTSCR ; Disable IRQ interrupt
CLI ; Enable interrupts

;****CONFIGURE VARIABLES****

LDA  #03
STA  RETRY ; 3 intentos antes de activar la alarma

TEST: JSR  STOP_ME ; Espero x 10secs.
; Permiso q' la bomba se cargue

LDA  DDRA
EOR  #OP_AMP ; Enciendo el Amplif. Op.
STA  DDRA

lda  DDRA ; Enciendo IR LED
eor  #LED
sta  DDRA

JSR  DELAY ; Delay x 1ms

ADC_LOOP: MOV  #03,ADSCR ; ADC ON, Canal 3
RUN_ADC: BRCLR 7,ADSCR,RUN_ADC ; Espero que se complete el ADC

LDA  DDRA ;Apago LED y Amplif. Op.
EOR  #LED
EOR  #OP_AMP
STA  DDRA

LDA  ADR ; Tomo el valor del ADC
CMP  #ALARM_VALUE ; Valor es < que ALARM_VALUE
BLO  NO_ALARM

LDA  RETRY ; Si el valor indic. Supuesto incendio
DECA ; Decremento valor RETRY
BNE  TEST ; 3 intentos hechos?, sino sigo...

ALARM: LDA  DDRA ; Si se hicieron, disparo ALARMA.
ORA  #FIRE
STA  DDRA
BRA  TEST

```

```

NO_ALARM:      LDA    #$00                ; Reseteo el valor RETRY
                STA    RETRY
                BRA    TEST

```

```

ENDING:        BRA    ENDING

```

```

;*****
;*          Delay Procedure          *
;*****

```

```

DELAY:         LDA    #$03    ;2 cycle
DELAY1:        LDX    #$00    ;2 cycle = (7*250ns) => (1.75us+320us) * 3 = 965.25us
DELAY2:        DECX           ;1 cycle-|
                TSTX           ;1 cycle ==> 5*250ns = 1.25us * 256 = 320us
                BNE    DELAY2    ;3 cycle-|
                DECA           ;1 cycle
                TSTA           ;1 cycle
                BNE    DELAY1    ;3 cycle
                RTS            ;4 cycle

```

```

;*****
;* Power Saving Module. Suspends CPU by 500ms * STOP_TIME *
;*****

```

```

STOP_ME:      LDA    #STOP_TIME
STOP_ME1:     STOP           ;Detengo el CPU por unos 500ms
                CLI
                DECA
                BNE    STOP_ME1
                RTS

```

```

;*****
;*          Auto-Wakeup Interrupt Service Handler          *
;*****

```

```

wake_up:      BSET 2,KBSCR    ; Escribo flag ACCK en KBSCR
                RTI

```

```

;*****
;*          Interrupt Vectors          *
;*****

```

```

                ORG    $FFDE
VADC:         fdb    spurious
VKBD:         fdb    wake_up

```

```

;* Hay un agujero (gap) en $FFE2-FFF1
                ORG    $FFF2    ;Salto localizaciones de vectores no usados.

```

```

VTOF:         fdb    spurious
VTCH1:        fdb    spurious
VTCH0:        fdb    spurious
                fdb    spurious
VIRQ:         fdb    spurious
VSWI:         fdb    spurious
VRESET:       fdb    main

```

*Fin del Artículo.....*