

$$\begin{array}{r}
 \times \quad \$4040 \\
 \$8340 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 \times \quad \$4040 \\
 \$1A \\
 \hline
 \end{array}$$

**Procedimiento:**

1. Vamos a separar el número de 24-bits (\$1A8340) en dos partes (\$1A y \$8340):

```
.EQU NUM_SEPARADO_1=$8340
.EQU NUM_SEPARADO_2=$1A
```

2. Vamos a cargar el número de 16-bits \$4040:

```
.EQU NUM_16=$4040
```

3. Vamos a extraer los bytes correspondientes y a multiplicar la primera parte:

```
;PARA EL NÚMERO DE 24-Bits:
LDI R16,LOW(NUM_SEPARADO_1)
LDI R17,HIGH(NUM_SEPARADO_1)

;PARA EL NÚMERO DE 16-Bits:
LDI R21,LOW(NUM_16)
LDI R22,HIGH(NUM_16)

;RESULTADOS EN:
.DEF RES1=R3
.DEF RES2=R4
.DEF RES3=R5
.DEF RES4=R6

;HAGAMOS LA MULTIPLICACIÓN DE 16-Bits x 16-Bits:
MUL R16,R21          ;LOW1*LOW2
MOV RES1,R0          ;PRIMERO SIN TOCAR
MOV RES2,R1          ;PARA SUMA

MUL R17,R22          ;HIGH1*HIGH2
MOV RES3,R0          ;PARA SUMA
MOV RES4,R1          ;ÚLTIMO MÁS CARRY

MUL R16,R22          ;LOW1*HIGH2
ADD RES2,R0
ADC RES3,R1
```

```

MUL R17,R21      ;LOW2*HIGH1
ADD RES2,R0
ADC RES3,R1
ADC RES4,R2      ;R2 no contiene valor sólo si hay CARRY
                  ; (DUMMY)

```

#### 4. Ahora multiplicamos el número de 16-bits por el de 8-bits:

```

; $R22 R21
;x $      1A

;RECORDEMOS QUE PARA EL NÚMERO DE 16-Bits:
LDI R21,LOW(NUM_16)
LDI R22,HIGH(NUM_16)

LDI R18,$1A

```

;RESULTADOS EN:

```

.DEF RES5=R7
.DEF RES6=R8
.DEF RES7=R9

```

```

MUL R18,R21      ;LOW1*LOW2
MOV RES5,R0      ;PRIMERO SIN TOCAR
MOV RES6,R1      ;PARA SUMA

```

```

MUL R18,R22      ;HIGH1*HIGH2
ADD RES6,R0      ;PARA SUMA
ADC RES7,R1      ;ÚLTIMO MÁS CARRY

```

#### 5. Finalmente sumamos los resultados, excluyendo la posición de RES1 y RES2:

```

;
;   +
;
;   RES4 RES3 RES2 RES1
;   ↓   ↓   ↓   ↓
; RES7 RES6 RES5 -----
;   ↓   ↓   ↓   ↓   ↓

```

```

ADD RES5,RES3
ADC RES6,RES4
ADC RES7,R2      ;POR POSIBLE CARRY. R2 es Dummy

```

#### 6. El resultado final está en: RES7 RES6 RES5 RES2 RES1

481	0.027772870	27.773	1736
482	0.027830425	27.830	1739
483	0.027887981	27.888	1743
484	0.027945536	27.946	1747
485	0.028003092	28.003	1750
486	0.028060647	28.061	1754
487	0.028118203	28.118	1757
488	0.028175759	28.176	1761
489	0.028233314	28.233	1765
490	0.028290870	28.291	1768
491	0.028348425	28.348	1772
492	0.028405981	28.406	1775
493	0.028463537	28.464	1779
494	0.028521092	28.521	1783
495	0.028578648	28.579	1786
496	0.028636203	28.636	1790
497	0.028693759	28.694	1793
498	0.028751314	28.751	1797
499	0.028808870	28.809	1801
500	0.028866426	28.866	1804

$$\begin{array}{llll}
 \text{CLK (SIN)} & = & 4\text{MHz} & \Rightarrow T_{\text{CICLO}} = 0.25 \mu\text{seg} * \text{ciclo} \\
 \text{CLK}/8 & = & 4\text{MHz}/8 = 500\text{KHz} & \Rightarrow T_{\text{CICLO}} = 2 \mu\text{seg} * \text{ciclo} \\
 \text{CLK}/64 & = & 4\text{MHz}/64 = 62.5\text{KHz} & \Rightarrow T_{\text{CICLO}} = 16 \mu\text{seg} * \text{ciclo} \\
 \text{CLK}/256 & = & 4\text{MHz}/256 = 15.625\text{KHz} & \Rightarrow T_{\text{CICLO}} = 64 \mu\text{seg} * \text{ciclo} \\
 \text{CLK}/1024 & = & 4\text{MHz}/1024 = 3.906\text{KHz} & \Rightarrow T_{\text{CICLO}} = 256 \mu\text{seg} * \text{ciclo}
 \end{array}$$

**Programa:**

```
; PROGRAMA QUE CONFIGURA EL SENSOR DE PROXIMIDAD
; ULTRASÓNICO HC-SR04
```

Encabezado para ATmega8515

Stack Pointer para ATmega8515

```
LDI R16,$FF
OUT DDRA,R16
```

La ecuación " $Y = 1737(X) - 154000$ " proviene de recorrer a la izquierda el punto decimal de " $m$ " y " $b$ " quedando como:  $Y = 1737.451(X) - 0.154$ , luego eliminamos 0.451 de " $m$ " y multiplicamos por 1 000 000 a " $b$ ". " $X$ " toma valores enteros de TCNT1L y TCNT1H

Siendo el resultado de esta operación de 32-bits, el artificio consiste en considerar las tres últimas cifras más significativas de la cantidad en decimal (que son las que se muestran en el LCD)

```

LDI R16,0b0000_0001      ;PB0-TRIGGER
OUT DDRB,R16              ;PB1-LECTURA de ECHO

LDI R16,$FF
OUT DDRC,R16

LDI R16,$FF
OUT DDRE,R16

;INCIALIZANDO DISPLAY
RCALL DISPLAY_CONTROL_ON
RCALL CLEAR_DISPLAY
RCALL HOME
RCALL ESCRIBIR_EN_LINEA_UNO

TRIGGER:                  ;Dispara al sensor ultrasónico
                          ;y muestra mensaje en el LCD

RCALL D
RCALL I
RCALL S
RCALL T
RCALL ESPACIO

LDI R16,0                  ;Inicializa contadores
OUT TCNT1L,R16
OUT TCNT1H,R16

LDI R16,0b0000_0001
OUT PORTB,R16
RCALL DELAY_10_MICROSEG
LDI R16,0b0000_0000
OUT PORTB,R16

;ESPERAMOS A QUE SALGAN LOS 8 PULSOS DEL BURST
;226 MICRO-SEGS
RCALL DELAY_226_MICROSEG

LDI R16,0b0000_0010
MASCARA:                  ;Detecta filo de subida de ECHO
IN R17,PINB
AND R17,R16
CP R16,R17
BRNE MASCARA

```

```

LDI R16,0<<CS12|1<<CS11|1<<CS10      ;Inicia CONTADOR_1 para
                                         ;contar desde 204useg
OUT TCCR1B,R16                          ;hasta 28.11mseg.

;0<<CS12|0<<CS11|0<<CS10----STOP
;0<<CS12|0<<CS11|1<<CS10----CK
;0<<CS12|1<<CS11|0<<CS10----CK/8
;0<<CS12|1<<CS11|1<<CS10----CK/64 <-----
;1<<CS12|0<<CS11|0<<CS10----CK/256
;1<<CS12|0<<CS11|1<<CS10----CK/1024

LDI R16,0b0000_0010
LDI R31,0b0000_0000
DETECTA_FIN_DE_ECO:      ;Detecta filo de bajada de ECHO
IN R17,PINB
AND R17,R16
CP R31,R17
BRNE DETECTA_FIN_DE_ECO

CLR R18
CLR R19

IN R18,TCNT1L      ;Lee el valor del TIMER1 TCNT1L
IN R19,TCNT1H      ;Lee el calor del TIMER1 TCNT1H

LDI R16,0<<CS12|0<<CS11|0<<CS10      ;STOP
OUT TCCR1B,R16

OUT PORTA,R18      ;Para prueba de LED's (OPCIONAL)

;Multiplicar los contadores por 16micro-segs
;para adaptarlo en MICRO-SEGUNDOS, para ajustar la ecuación
;de la recta que luego serán convertidos a DISTANCIA

;Multiplicación de un número de 16-Bits por uno de 8-Bits
;R18 es el registro LOW = TCNT1L
;R19 es el registro HIGH= TCNT1H

LDI R20,16          ;$10 equivalente a 16 MICRO-SEGUNDOS

MUL R18,R20          ;R18 contiene el valor de TCNT1L
MOV R21,R0            ;PRIMERO
MOV R22,R1           ;+

```

```

MUL R19,R20                ;R19 contiene el valor de TCNT1H
ADD R22,R0
ADC R23,R1                  ;ÚLTIMO
;Debido a que el resultado mayor en nuestro proyecto
;abarca R21 y R22, desechamos R23

;Como vamos a adaptar la ecuación del comportamiento del
;sensor necesitamos ajustar el valor de R21 y R22, por lo
;tanto:

SUBI R21,3
SBC R22,R2                  ;si hay CARRY se resta a R22. R2 es
                           ;Dummy para CARRY

CONVIRTIENDO_MICROSEGUNDOS_A_CENTIMETROS:
;"X" Es el número en MICRO-SEGUNDOS del CONTADOR de 16-Bits
;"Y" Es el resultado en CENTÍMETROS

;Esta es la ecuación original
;Y=17374.51(X)-1.54

;ADAPTANDO para reducir # de Bits
;Y=1737(X)-154000

;***MULTIPLICACIÓN de 16-Bits por uno de 16-Bits***

;Para el número (1737)d=$06C9
;Multiplicado por el número de 16-Bits =$R22 R21

.EQU NUM_1=$06C9

.DEF RES1=R3
.DEF RES2=R4
.DEF RES3=R5
.DEF RES4=R6

LDI R16,LOW(NUM_1)
LDI R17,HIGH(NUM_1)

MUL R16,R21                 ;LOW1*LOW2
MOV RES1,R0                 ;PRIMERO
MOV RES2,R1                 ;+

MUL R17,R22                 ;HIGH1*HIGH2
MOV RES3,R0                 ;+

```

```

MOV RES4,R1                ;ÚLTIMO MÁS CARRY

MUL R16,R22                ;LOW1*HIGH2
ADD RES2,R0
ADC RES3,R1

MUL R17,R21                ;LOW2*HIGH1
ADD RES2,R0
ADC RES3,R1
ADC RES4,R2                ;R2 no contiene valor pero si hay
                           ;CARRY SUMA

;RESULTADO FINAL está en: RES4 RES3 RES2 RES1

;Ahora restamos 154000 según la ecuación
;Y=1737(X)-154000

.EQU NUM_2=$025990        ;=154000

LDI R16,BYTE1(NUM_2)      ;$90
LDI R17,BYTE2(NUM_2)      ;$59
LDI R18,BYTE3(NUM_2)      ;$02

;      RES4 RES3 RES2 RES1
;-      R18  R17  R16
;-----

SUB RES1,R16
SBC RES2,R17
SBC RES3,R18
SBC RES4,R2                ;CARRY

;Ya está convertido en CENTÍMETROS.
;Como el resultado tiene 10,000,000 CIFRAS (32-Bits)
;al final sólo vamos a desplegar en LCD las ÚLTIMAS 3

;CONVERTIDOR DE CENTÍMETROS A DISPLAY LCD
CONVIERTE_DATO_A_LCD:

CLC                        ;CLEAR CARRY
CLR R22                    ;BORRA CONTADOR DE 1ª CIFRA
CLR R23                    ;BORRA CONTADOR DE 2ª CIFRA
CLR R24                    ;BORRA CONTADOR DE 3ª CIFRA
CLR R25                    ;BORRA CONTADOR DE 4ª CIFRA

```



```

;DIEZ MILLONES:
.EQU NUMERO1=10_000_000 ;$989680

LDI R16,BYTE1(NUMERO1) ;$80
LDI R17,BYTE2(NUMERO1) ;$96
LDI R18,BYTE3(NUMERO1) ;$98

LDI R21,0 ;CONTADOR DE DIEZ MILLONES

MOV R10,RES1
MOV R11,RES2
MOV R12,RES3
MOV R13,RES4

CICLO10:
SUB R10,R16
SBC R11,R17
SBC R12,R18
SBC R13,R2

BRCS MILLONES

MOV RES1,R10
MOV RES2,R11
MOV RES3,R12
MOV RES4,R13

INC R21 ;CONTADOR DE DIEZ MILLONES

MOV R22,R21

BRCC CICLO10

MILLONES:
CLC ;CLEAR CARRY

.EQU NUMERO2=1_000_000 ;$0F4240
LDI R16,BYTE1(NUMERO2) ;$40
LDI R17,BYTE2(NUMERO2) ;$42
LDI R18,BYTE3(NUMERO2) ;$0F
.
.
.

```

```
CIEN_MILES:
CLC                                ;CLEAR CARRY
```

```
.EQU NUMERO3=100_000             ;$0186A0
LDI R16,BYTE1(NUMERO3)           ;$A0
LDI R17,BYTE2(NUMERO3)           ;$86
LDI R18,BYTE3(NUMERO3)           ;$01
.
.
.
```

```
DIEZ_MILES:
CLC                                ;CLEAR CARRY
```

```
.EQU NUMERO4=10_000              ;$002710
LDI R16,BYTE1(NUMERO4)           ;$10
LDI R17,BYTE2(NUMERO4)           ;$27
LDI R18,BYTE3(NUMERO4)           ;$00
.
.
.
```

```
SACANDO_ASCII:
LDI R21,$30
ADD R22,R21
ADD R23,R21
ADD R24,R21
ADD R25,R21
```

```
OUT PORTC,R22                     ;1a CIFRA
```

```
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16
```

```
OUT PORTC,R23                     ;2a CIFRA
```

```
.
.
.
```

```
OUT PORTC,R24                     ;3a CIFRA
```

```

.
.
.

```

```
RCALL PUNTO
```

```
OUT PORTC,R25 ;4a CIFRA
```

```

.
.
.

```

```
RCALL ESPACIO
```

```
RCALL C_MINUS
```

```
RCALL M_MINUS
```

```
RCALL PUNTO
```

```
RCALL DELAY_200_MILISEG ;ESPERA UN TIEMPO PARA NUEVO
                          ;TRIGGER
```

```
RCALL HOME ;PARA REGRESAR EL CURSOR DEL
            ;LCD AL ORIGEN
```

```
RJMP TRIGGER
```

```
.INCLUDE "SUBROUTINAS_ASCII_Y_CONTROL_LCD.TXT"
```