



**INSTITUTO POLITÉCNICO
NACIONAL**
ESCUELA SUPERIOR DE CÓMPUTO



“Temporizadores”

Integrantes del Equipo:

Contreras Cardoso Adolfo

Martínez Alvarado Bryan Alexis

Maya Martínez Alonso Rubén

Pérez Gómez Santiago

Grupo:

3CM15

Profesor:

Ing. José Juan Pérez Pérez

Asignatura:

Introducción a los Microcontroladores

Planteamiento del Problema

En un supermercado se ha determinado premiar a cada cliente múltiplo de 6.

- Desarrolle un sistema basado en un AVR, el cual debe detectar al cliente número 6 y generar un tono de 440 Hz (aproximadamente), por 5 segundos, cuando eso ocurra.
- Los clientes deben presionar un botón para ser considerados (flancos de bajada).
- Se trata de una aplicación para los temporizadores, utilice el temporizador 0, manejado con eventos externos, para llevar el conteo de clientes.
- Al temporizador 2 para generar el tono de 440 Hz y al temporizador 1 para el conteo de 5 segundos, considerar la frecuencia del microcontrolador de 1 MHz.
- Mostrar en un display BCD la cuenta descendente (5-0)

Desarrollo

Para realizar la implementación se utilizó como base el programa propuesto por el profesor durante la sesión de clase anterior, añadiendo el uso del timer 2, además, para conseguir lo planteado en el problema se modificaron los valores de reinicio de los timers, así mismo se añadió el conteo descendente para el display dentro del ciclo infinito.

El diagrama de flujo correspondiente al resultado es el siguiente:

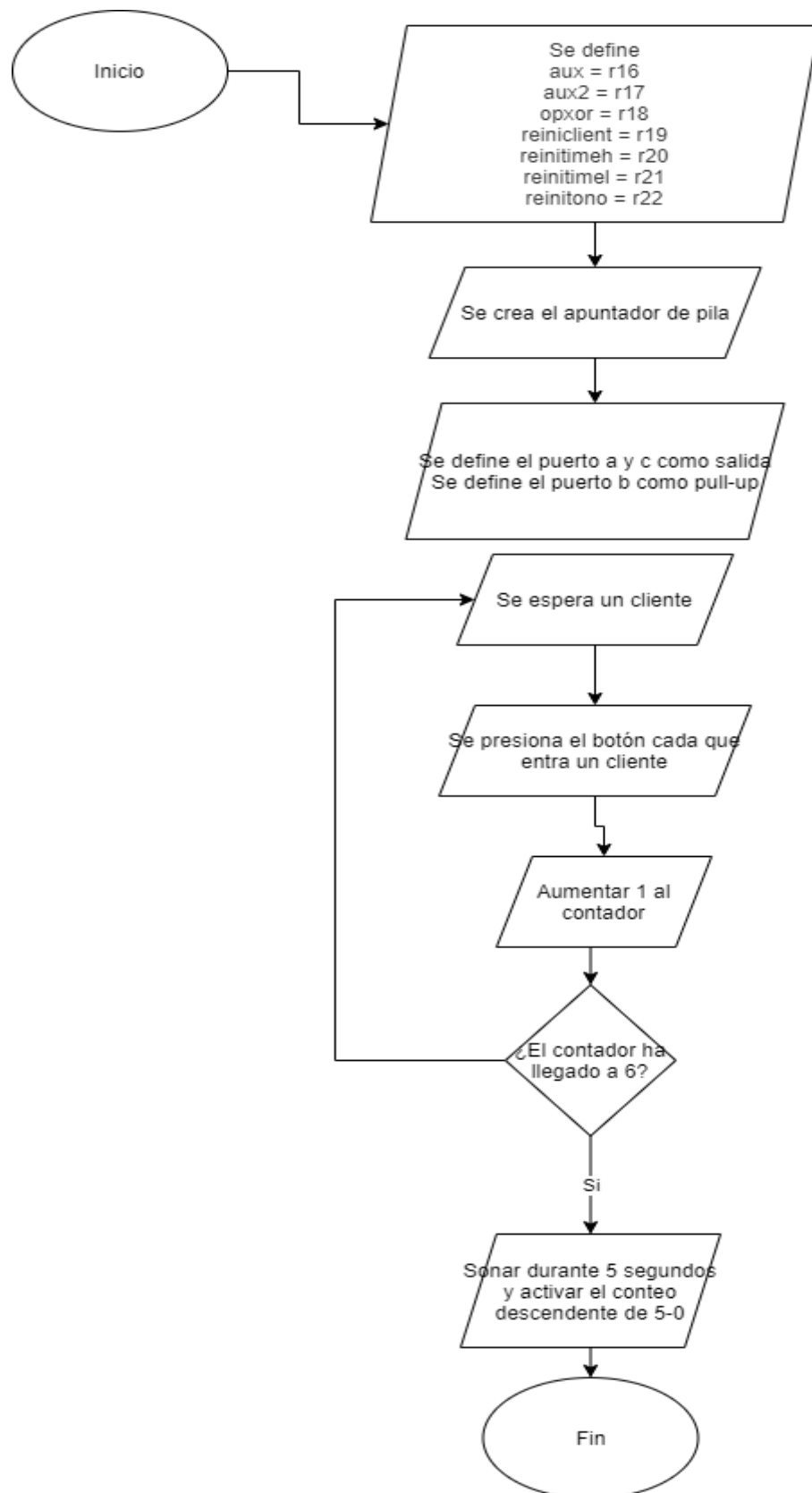


Figura 1. Diagrama de Flujo

El respectivo código en ensamblador es el siguiente:

```
;programa para conteo de clientes con timers

;timer0 = conteo de clientes
;timer1 = generacion de onda 440hz (2.27ms)
;timer2 = duracion de 5 segundos

;valor a cargar en timer1
;19531 x 256prescala = 4,999,936ciclos = 4.99ms
;65536 - 19531 = 46005 = $B3B5

;valor a cargar en timer2 1.1363 (mitad de 2.27ms)
;142 x 8prescala = 1136ciclos = 1.136ms
;256 - 142 = 114

.include "m8535def.inc"

.def aux = r16
.def aux2 = r17
.def opxor = r18

;para timer0
.def reiniclient = r19

;para timer1
.def reinitimeh = r20
.def reinitimel = r21

;para timer2
.def reinitono = r22
```

Figura 2. Primera Sección de Código

```
inicio:
    rjmp main

;timer 2
.org $004
rjmp tono

;timer1 y timer0
.org $008
rjmp segundos
rjmp clientes

main:
    ;apuntador de pila
    ldi aux,low(ramend)
    out spl,aux
    ldi aux,high(ramend)
    out sph,aux

    ;puerto a y c como salida
    ser aux
    out ddra,aux
    out ddrc,aux

    ;puerto b como pull-up
    out ddrb,aux

    ;eligiendo flanco de bajada
    ;para activar interrupcion int0
    ldi aux,2
```

Figura 3. Segunda Sección del Código

```

out mcucr,aux

;para activar int0
ldi aux,$40
out gicr,aux

;habilitar interrupciones por
;desborde de timer counter
;0,1 y 2, pero estan detenidos
ldi aux,$45
out tmsk,aux

;activando el conteo de timer0
ldi aux,6
out tccr0,aux

;habilitando globalmente
;las interrupciones
sei

;para conteo del display
ldi aux2,255

;reiniciando conteo de clientes
ldi reiniclient,250
out tcnt0,reiniclient

;para generar señal cuadrada
ldi opxor,1

```

Figura 4. Tercera Sección del Código

```

;valor inicial timer1
ldi reinitimeh,$B3
ldi reinitimel,$B5

;valor inicial timer2
ldi reinitono,114

ciclar:
    sbic pinb,0
    rcall display

    rjmp ciclar

display:
    in aux,tcnt0
    sub aux2,aux
    out portc,aux2

    ;para conteo del display
    ldi aux2,255

    ret

clientes:
    ;reiniciando timer 0
    out tcnt0,reiniclient

    ;activando el conteo de timer1
    ;(256 veces 65.54ms = 16.77seg)
    ldi aux,4

```

Figura 5. Cuarta Sección del Código

```

        out tccr1b,aux

        ;activando el conteo de timer2
        ;(8 veces 0.26ms = 2.08ms)
        ldi aux,2
        out tccr2,aux

        ;reiniciando timer1
        out tcnt1h,reinitimeh
        out tcnt1l,reinitimel

        ;reiniciando timer2
        out tcnt2,reinitono

        reti

tono:
        ;reiniciando timer2
        out tcnt2,reinitono

        ;or exclusiva para generar señal
        in aux,pina
        eor aux,opxor
        out porta,aux

        reti

segundos:
        ;desactivando el timer1 y timer2
        ldi aux,0
        out tccr1b,aux
        out tccr2,aux

        reti

```

Figura 6. Quinta Sección del Código

The diagram illustrates the hardware setup for an ATMEGA8535 microcontroller. The microcontroller's pins are configured as follows:

- Buttons:** BTN1 and BTN2 are connected to pins 12 (XTAL2/RESET) and 1 (PB0/T0/XCK) respectively. A 1k5 resistor (R1) is connected between pins 1 and 12.
- Display and Speaker:** The 7-segment display (LS1) is connected to pins 22 (PC0/SCL), 23 (PC1/SDA), 24 (PC2), 25 (PC3), 26 (PC4), 27 (PC5), 28 (PC6/TOSC1), and 29 (PC7/TOSC2). The speaker (SPEAKER) is connected to pins 39 (PA0/ADC0) and 40 (PA1/ADC1).
- Logic Analyzer:** The logic analyzer is connected to pins 39, 40, 22, and 23, showing four waveforms (A, B, C, D) representing digital signals.

Figura 7. Valor 5 en el display

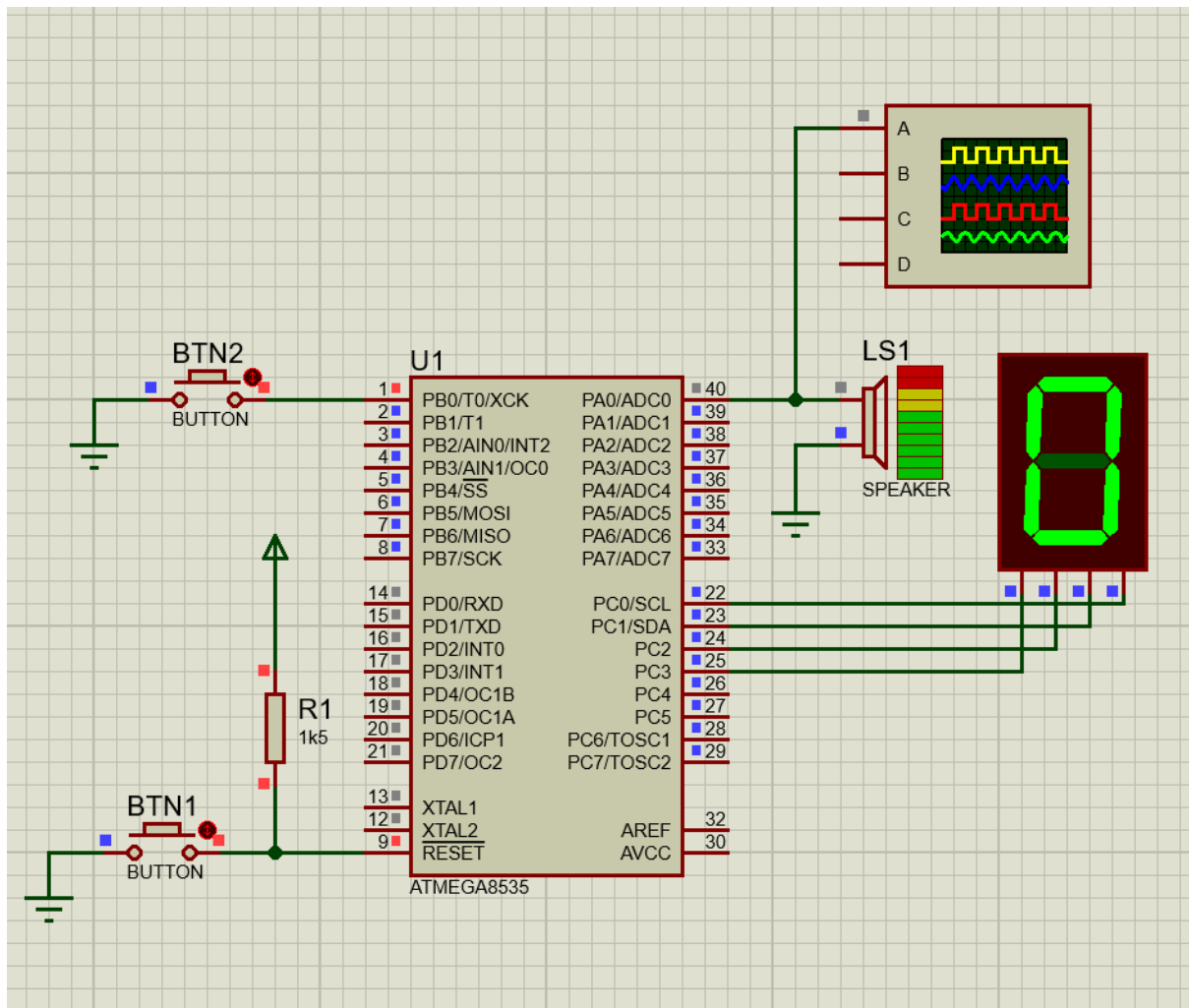


Figura 8. Valor 0 en el display

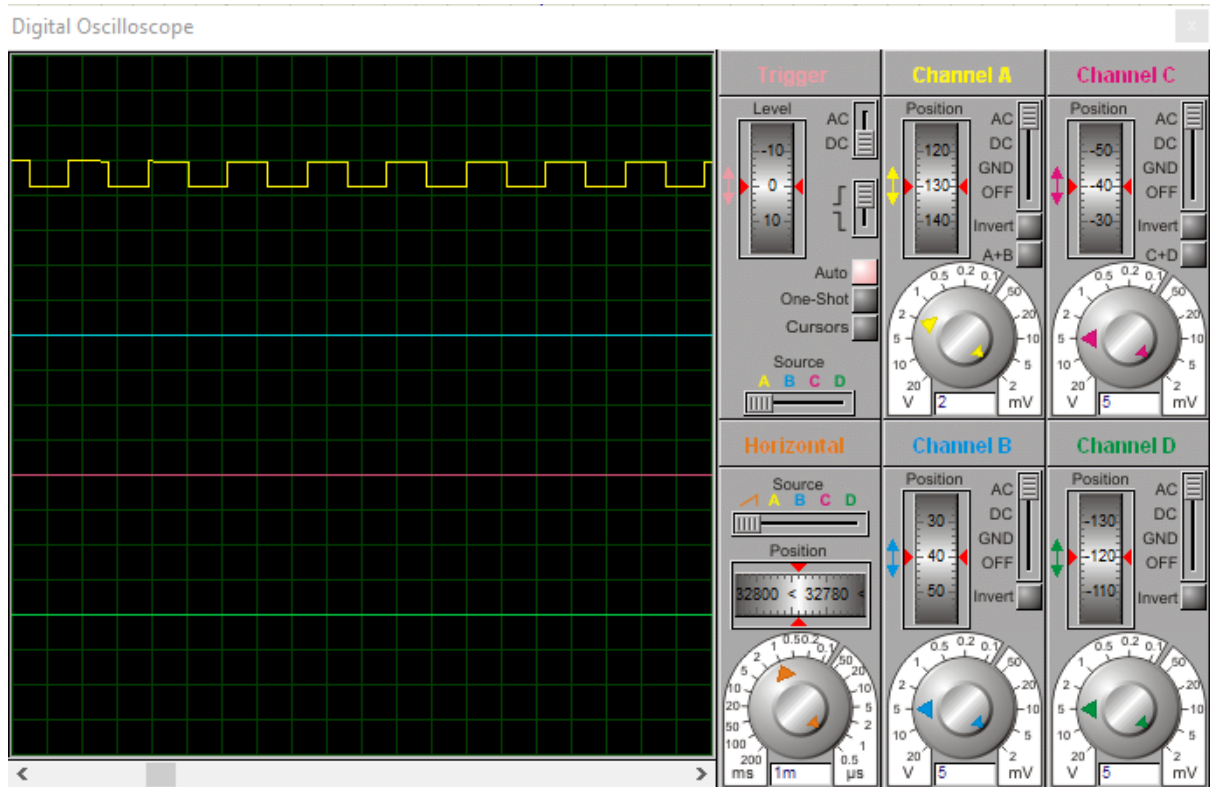


Figura 9. Sonido y onda Generada en el Osciloscopio

Conclusiones Individuales

Contreras Cardoso Adolfo

La elaboración de esta práctica fue interesante y algo compleja, ya que no se había utilizado timers en cursos anteriores, con apoyo de nuestro profesor y el análisis de ejemplos anteriores se pudo realizar la práctica sin problema, en lo personal me cuesta un poco ensamblador pero gracias al trabajo en equipo se pudo entender algo mejor sobre el tema y la implementación del mismo.

Martínez Alvarado Bryan Alexis

Durante la realización de esta práctica se ha observado cómo con el uso de timers, que hemos visto anteriormente, pudimos implementarlos de manera satisfactoria, haciendo uso del conocimiento adquirido en la parte teórica de la clase, y obteniendo los resultados esperados.

Maya Martínez Alonso Rubén

El uso de timers es una herramienta muy poderosa pero es muy complejo de usar, para este tipo de usos es un poco complicado pero quizá para proyectos más complejos sea más útil y más fácil de implementar. A lo largo de las sesiones el manejo de los timers se ha hecho más sencillo de entender pero mientras más practiquemos con ellos más fácil vamos a poder usarlos.

Pérez Gómez Santiago

El manejo de timers resulta bastante complejo, pues aún en este punto desconocemos algunas de sus funcionalidades, sin embargo, resultó más sencilla su implementación al realizar modificaciones a programas realizados con anterioridad, además, el modelo de aprendizaje que se ha empleado durante las sesiones de clase facilitó su comprensión, por lo que se alcanzaron los objetivos propuestos sin contratiempos.