



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Departamento de Ciencias e Ingeniería
de la Computación



Sistemas en Chip

Practica 08

“Cronómetro de 59.9 segundos”

Profesor: Fernando Aguilar Sánchez

Grupo: 6CM1

Equipo 3

Alumnos:

Ocampo Téllez Rodolfo

Patlani Mauricio Adriana

Ruvalcaba Flores Martha Catalina

Sandoval Hernández Eduardo

Fecha de entrega: 11/12/2022

Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad de realizar un cronómetro de 59.9 segundos.

Introducción Teórica

El microcontrolador ATMEGA8535 fabricante ATMEL. En la figura 1 se muestra el microcontrolador ATMEGA8535 la cual maneja datos de 8 bits es decir su bus de datos de 8 bits. Aunque este contiene tres registros los cuales son el x, y y z, los cuales manejan datos de 16 bits.

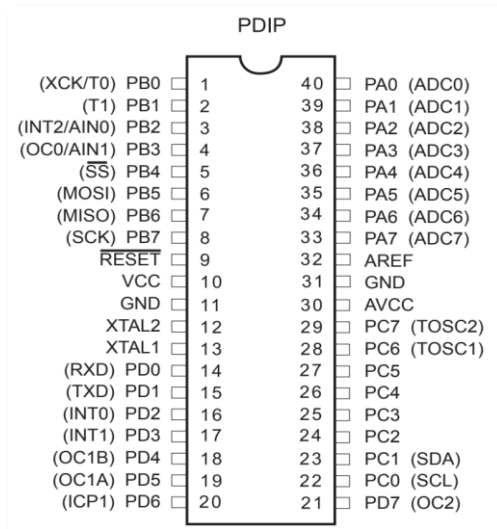


Figura 1. Configuración de pines ATmega8535

El microcontrolador utiliza una arquitectura cerrada, es decir, aquel que es inmodificable por los programadores ajenos a la compañía propietaria del código fuente. Por lo tanto, a este sistema no se le pueden colocar dispositivos periféricos, solo se usa el hardware de la compañía propietaria ya que los dispositivos ajenos a dicha compañía no son compatibles.

El microcontrolador ATMEGA8535 utiliza un encapsulado DIP-40, común en la construcción de circuitos integrados que consiste en un bloque con dos hileras paralelas de pines, observar la figura 2.

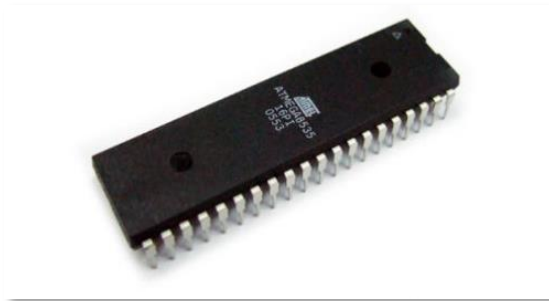


Figura 2. Microcontrolador atmega8535

La comunicación interna del microcontrolador se categoriza en 4 puertos, en la figura 1 se puede analizar la etiquetación de los puertos PA0 al PA7, PB0 al PB7, PC0 al PC7 y PD0 al PD7. Cabe recordar que maneja datos de 8 bits.

Posee un oscilador interno de 1MHz, sin embargo, como es un oscilador RC, es susceptible a variar la frecuencia con respecto a las variaciones de temperatura. Por otro lado, puede conectarse un oscilador de 0 a 8 MHz o de 0 a 16 MHz.

Crónometro

El funcionamiento usual de un cronómetro consiste en empezar a contar desde cero al pulsarse el mismo botón. Permite medir varios tiempos con el mismo comienzo y distinto final. Para ello se congela los sucesivos tiempos con un botón distinto, normalmente con el de reinicio, mientras sigue contando en segundo plano hasta que se pulsa el botón de comienzo.

Los cronómetros pueden activarse con métodos automáticos, con menor margen de error y sin necesidad de un actor. La unidad del intervalo de tiempo es el segundo (s).

Contador

Un contador electrónico digital es muy útil por ello en la actualidad estamos rodeados de dispositivos que disponen de algún tipo de contador digital, incluso en la mayoría de los electrodomésticos vienen equipados con uno. Nuestro contador digital tiene un campo muy amplio para su aplicación de forma rotacional (cuantas vueltas efectúa un objeto) o de forma secuencial (Ej. en una empresa para el conteo de cajas de producto, etc.).

El contador como componente ha colaborado en la evolución de diversos dispositivos tecnológicos. Este dispositivo permite visualizar la formación numérica de manera ascendiente desde 0 hasta 9 en un display.

Display de 7 segmentos

El display de 7 segmentos es un dispositivo optoelectrónico que permite visualizar números del 0 al 9 y o algunas letras. El dispositivo cuenta con 7 leds, uno por cada segmento, a cada uno de estos se le asigna una letra que va de la “a” a la “g” y se encuentran organizados como indica la figura 3.

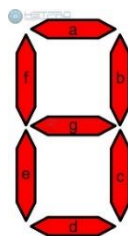


Figura 3. Disposición de los leds en un display de 7 segmentos.

Bandera

Una bandera o interruptor (flag) es una variable lógica que se utiliza para conservar el estado (verdadero o falso) de una condición.

Se denomina bandera o interruptor por asociarse a un interruptor (encendido/apagado) o a una bandera (arriba/abajo). El valor del interruptor debe inicializarse antes de comenzar el bucle y debe cambiar su estado (valor) dentro del cuerpo del bucle para preparar la siguiente iteración.

Las banderas o interruptores se representan con variables lógicas. Estas variables se inicializan a uno de los dos posibles valores (true o false) y toma el otro valor cuando sucede el evento que se está verificando.

Materiales y Equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 3 Display cátodo común
- 21 Resistores de $330\ \Omega$ a $\frac{1}{4}\text{ W}$
- 3 Push Button

Desarrollo Experimental

1.- Diseñe un programa en el que coloque dos Displays, uno en el Puerto A y el otro en el Puerto B y con una terminal del Puerto D detecte la cuenta a través de un par infrarrojo.

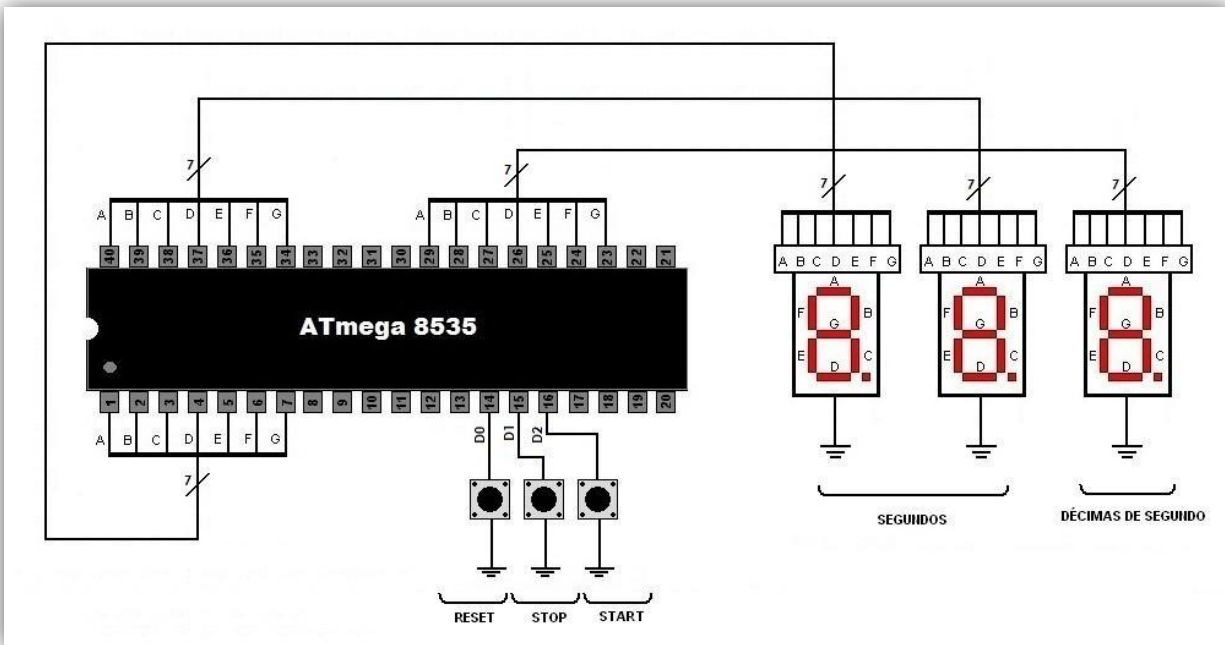


Figura 4. Circuito para el cronómetro.

Estructura del programa

Código de la configuración de los periféricos utilizados y código del programa principal en C, proporcionados por el IDE de CodeVision.

```
#include <mega8535.h>
#include <delay.h>
#define reset PIND.0
#define stop PIND.1
#define start PIND.2
bit bandera;
unsigned char unidades;
unsigned char decenas;
unsigned char decimas;
const char
tabla7segmentos[16]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,
0x7c,0x58,0x5e,0x79,0x71};
void main(void)
{
// Input/Output Ports initialization
// Port A initialization
DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) |
(1<<DDA1) | (1<<DDA0);
PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) |
(0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
// Port B initialization
DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) |
(1<<DDB1) | (1<<DDB0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) |
(0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
// Port C initialization
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) |
(1<<DDC1) | (1<<DDC0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) |
(0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
// Port D initialization
DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) |
(0<<DDD1) | (0<<DDD0);
// State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) |
(1<<PORTD2) | (1<<PORTD1) | (1<<PORTD0);
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) |
(0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;
```

```
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) |
(0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0<<AS2;
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) |
(0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
(0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
(0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) |
(0<<SPR1) | (0<<SPR0);
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
bandera = 0;
unidades = 0;
decenas = 0;
decimas = 0;
while (1)
{
    if(start == 0){
        if(decenas == 6){
            bandera = 1;
            unidades = 0;
            decenas = 0;
            decimas = 0;
        }
    }
}
```

```
    }
    else
        bandera = 1;
}
if(stop == 0)
    bandera = 0;
if(reset == 0){
    bandera = 0;
    unidades = 0;
    decenas = 0;
    decimas = 0;
}
if(bandera){
    decimas++;
    if(decimas == 10){
        decimas = 0;
        unidades++;
    }
    if(unidades == 10){
        unidades = 0;
        decenas++;
    }

    if(decenas == 6){
        bandera = 0;
    }
}
PORTB = tabla7segmentos[decenas];
PORTA = ~tabla7segmentos[unidades];
PORTC = ~tabla7segmentos[decimas];
delay_ms(100);
}
```

Simulación

La simulación se realizó en el software Proteus Design Suite, previamente cargado el programa en formato “.hex” obtenido del software CodeVision AVR, el cual es un software para el microcontrolador Microchip AVR y sus contrapartes XMEGA.

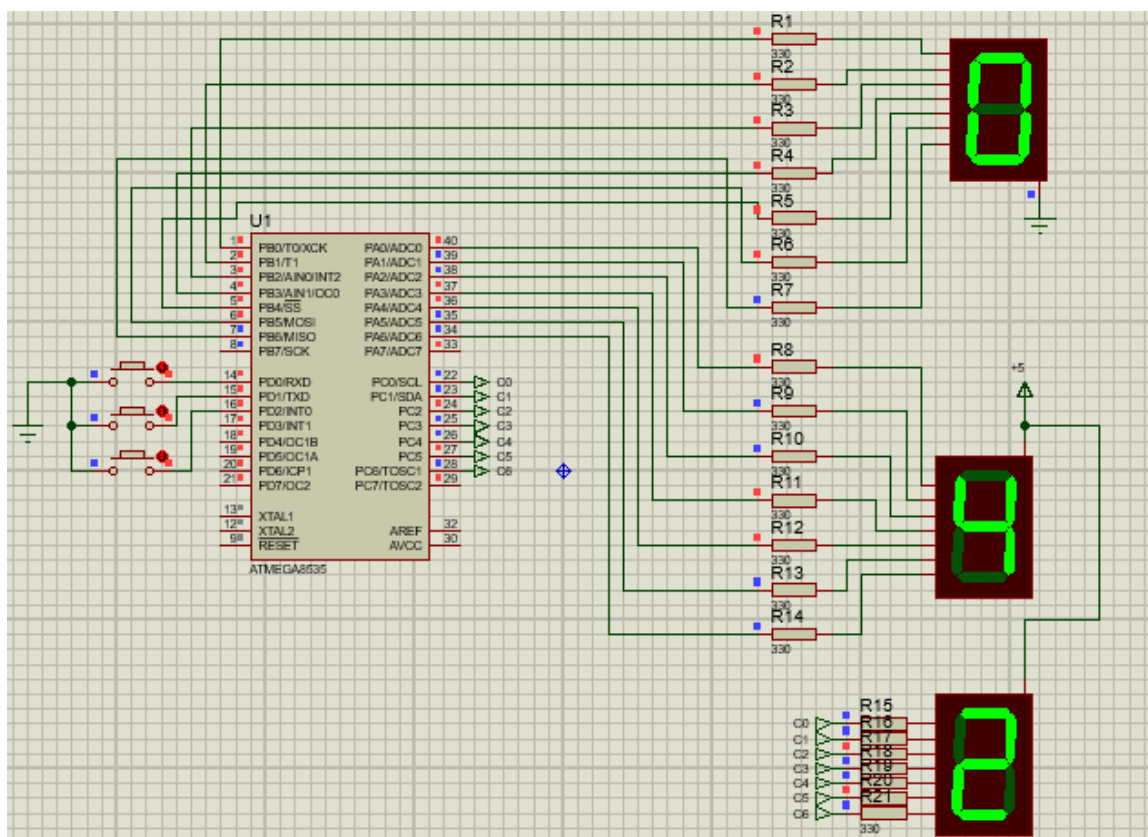


Figura 5. Simulación de la práctica 8.

Fotografía del circuito armado

A continuación, se muestra la figura 6 la evidencia sobre la realización y prueba de la practica 8.

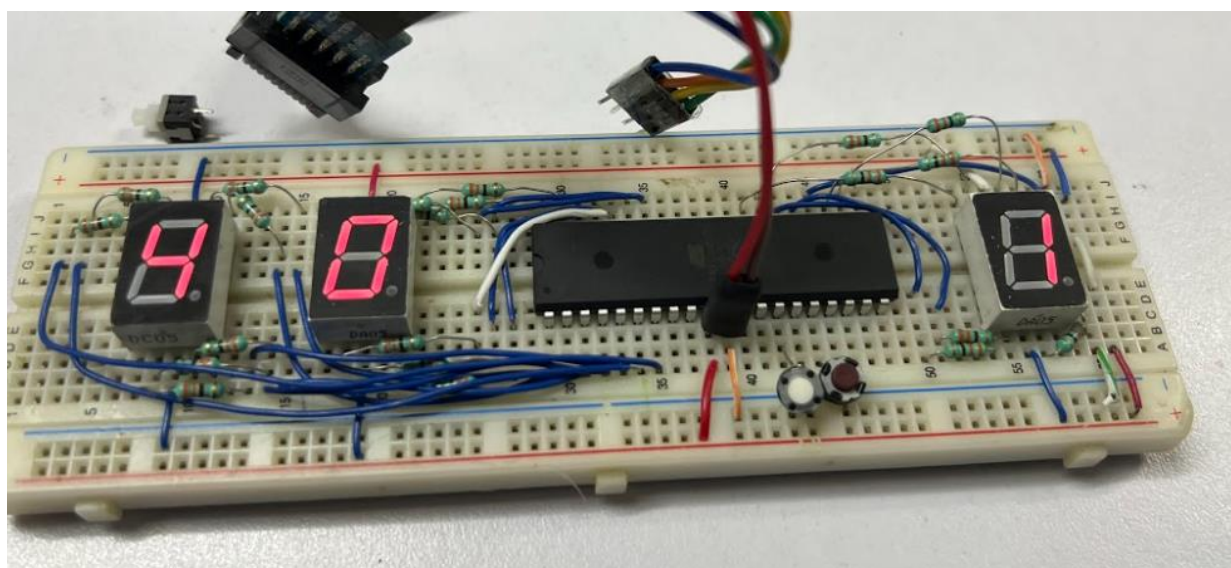


Figura 6. Circuito armado y funcionamiento de la práctica 8.

Observaciones y conclusiones Individuales

Ocampo Téllez Rodolfo

Esta práctica se pudo realizar sin mayor inconveniente, tomando en cuenta únicamente que el puerto C se debía conectar en orden diferente al armar el hardware. En anteriores se manejaron contadores, y limitaciones en el conteo para los dígitos del 0 al 9, en esta ocasión además se añadió una limitación en el conteo del display conectado al puerto B, pues en la práctica se requirió que solo se llegará a 60 segundos, y por tanto el máximo dígito mostrado fue 6. La implementación de organizar el conteo como un cronómetro fue sencilla haciendo uso de condicionales if, así, una vez que el botón de inicio era presionado se procedía a la ejecución de 3 if anidados para décimas, unidades y decenas de segundo, asimismo el uso de una bandera fue necesario para indicar si debía estar activo o no el conteo, de modo que en el botón de paro únicamente requirió ponerla en 0, y finalmente el reseteo manual se implementó, deteniendo el conteo e igualando las variables 0.

Patlani Mauricio Adriana

En esta práctica, comparada a las anteriores, fue sencilla, al tener los contadores, el cronómetro no tuvo mayores complicaciones ya que solo tenía que mostrar de manera consecutiva en los displays los números. Para realizarlo, solo se usaron sentencias de iteración (if-else) anidados donde se manejaron únicamente los índices para así mostrar en cada par de displays las decenas y unidades de segundos y decimas de segundo respectivamente.

Ruvalcaba Flores Martha Catalina

Para la práctica 8, no tuvo mayor complicación, puesto que solo requirió que funcionara de manera consecutiva los displays, por lo que, requirió que se tuviera un control en los índices ya que se manejaron las décimas, unidades y decenas de segundos. Los primeros dos displays se usaron para las decenas y unidades de segundos, implementando el conteo del 0 al 9 visto en las practicas anteriores, y el tercer display para las décimas de segundos, se implementó un conteo limitándolo del 0 al 6.

Sandoval Hernández Eduardo

Esta práctica se realizó de manera exitosa gracias a que se pudo reutilizar el circuito de la práctica 7 y solo anexar el display y los botones necesarios, además de que el código para el cronómetro fue sencillo de realizar pues es muy parecido al contador de 00 a 99 con la diferencia de que se incluyen las décimas de segundo, se limita a 60 y el conteo lo realiza el mismo microcontrolador sin necesidad de pulsar un boton por cada conteo como lo fue en las prácticas anteriores, es seguro que nos servirá para las prácticas futuras pues ya sabemos cómo sincronizar mejor los tiempos para el microcontrolador.

Bibliografía

- F. Aguilar. (2020, Octubre). 08 Práctica de Cronómetro de 59.9 segundos. Introducción a los Microcontroladores. [Online]. Disponible en: <https://youtu.be/1008C88iLdA>

- Atmel Corporation. (2006). 8-bit Microcontroller with 8K Bytes In-System Programmable Flash. [Online]. Disponible en: <https://drive.google.com/file/d/1acpQaDlsyLHr3w3ReSgrlXRbshZ5Z-lb/view>
- UNICEN (s.f.). Cronómetro digital de mano. [Online]. Disponible en: https://kitlab.exa.unicen.edu.ar/cronmetro_digital_de_mano.html#:~:text=Instrumento%20digital%20a%20pila%20que,precisi%C3%B3n%20de%20cent%C3%A9simas%20de%20segundos.
- HetPro (2019). Display 7 Segmentos ánodo y cátodo común: [En línea]. Disponible en <https://hetpro-store.com/TUTORIALES/display-7-segmentos-anodo-catodo-comun/>
- Creative Commons. (s.f.) Bandera - Sección Informática. [Online]. Disponible en: <https://www.glosarioit.com/Bandera>
- A. Solano. (2014). Contador BCD de 0-9 con temporizador 555 (Automatización): [En línea]. Disponible en: <https://www.monografias.com/trabajos102/contador-bcd-0-9-temporizador-555-automatizacion/contador-bcd-0-9-temporizador-555-automatizacion>