



**INSTITUTO POLITÉCNICO
NACIONAL**



ESCUELA SUPERIOR DE CÓMPUTO

Introducción a los microcontroladores

Profesor: Aguilar Sánchez Fernando

Práctica 01 "Puertos de Entrada/Salida"

Integrantes del Equipo:

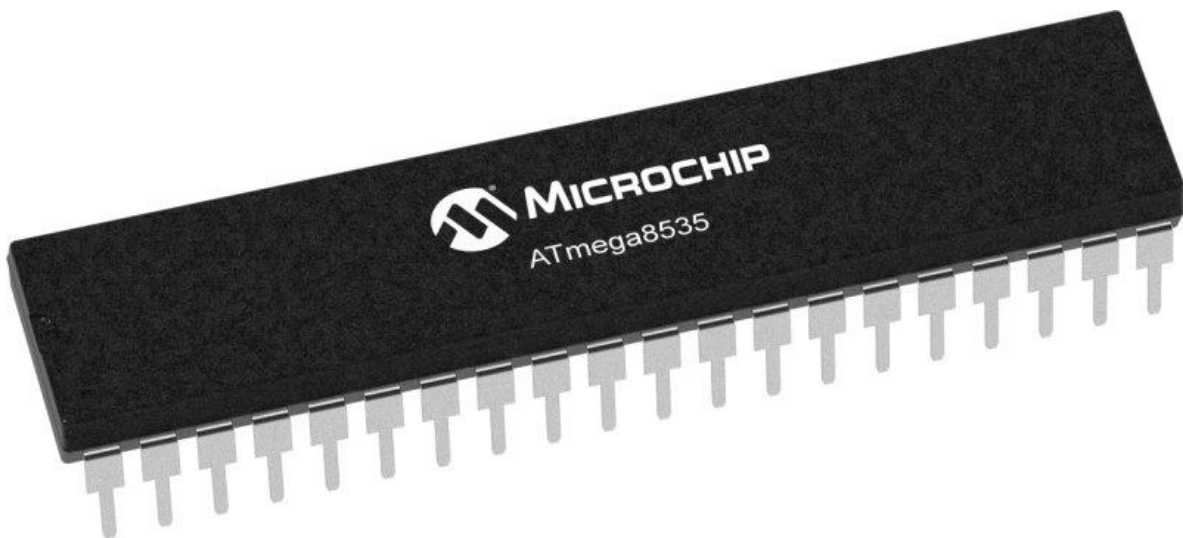
- **Arizmendi Alvarado Brian**
- **Rodríguez López Ricardo**

Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad de programar los puertos de entrada y salida del Microcontrolador ATmega8535 usando las herramientas “Code Vision AVR” y “AVR Studio 4”.

Introducción Teórica

El microcontrolador ATmega8535 es uno de los dispositivos electrónicos más populares en el mundo de la electrónica y la programación de microcontroladores. Es un microcontrolador de 8 bits basado en la arquitectura RISC avanzada, fabricado por Atmel (ahora propiedad de Microchip), y es ampliamente utilizado en aplicaciones de control industrial, automatización, sistemas de seguridad, dispositivos de medición y electrónica de consumo. Debido a su alta capacidad de procesamiento y su facilidad de programación, el ATmega8535 es una opción popular entre los programadores de microcontroladores.



Por otro lado, Code Vision AVR es un software de desarrollo de aplicaciones de microcontroladores AVR. Incluye un entorno de desarrollo integrado (IDE) y un compilador C que permite a los programadores de microcontroladores desarrollar aplicaciones de manera rápida y eficiente. Code Vision AVR ofrece una amplia biblioteca de funciones predefinidas y una interfaz gráfica fácil de usar que permite

a los usuarios programar el microcontrolador ATMega8535 de manera sencilla y precisa.

En conjunto, el microcontrolador ATMega8535 y Codevision AVR son una combinación poderosa que permite a los programadores de microcontroladores desarrollar aplicaciones avanzadas y personalizadas de manera eficiente. Con estas herramientas, los usuarios pueden controlar y programar dispositivos electrónicos con precisión y eficacia. Además, el ATMega8535 cuenta con características tales como 32 pines de entrada/salida, 8 canales ADC, y 8K bytes de memoria flash programable, lo que lo hace ideal para aplicaciones de control en tiempo real. En resumen, el microcontrolador ATMega8535 y Codevision AVR son herramientas esenciales para aquellos que buscan desarrollar aplicaciones avanzadas y controlar dispositivos electrónicos de manera eficiente.

Materiales y Equipo empleado

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATMega 8535
- 8 LED's
- 8 Resistores de 330 Ω a 1/4 W
- 1 Dip switch u ocho Push Botón

Código

```
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

// Port D initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1) | (1<<DDD0);
// State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
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;
```

```
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization
// USART disabled
UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
```

```

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

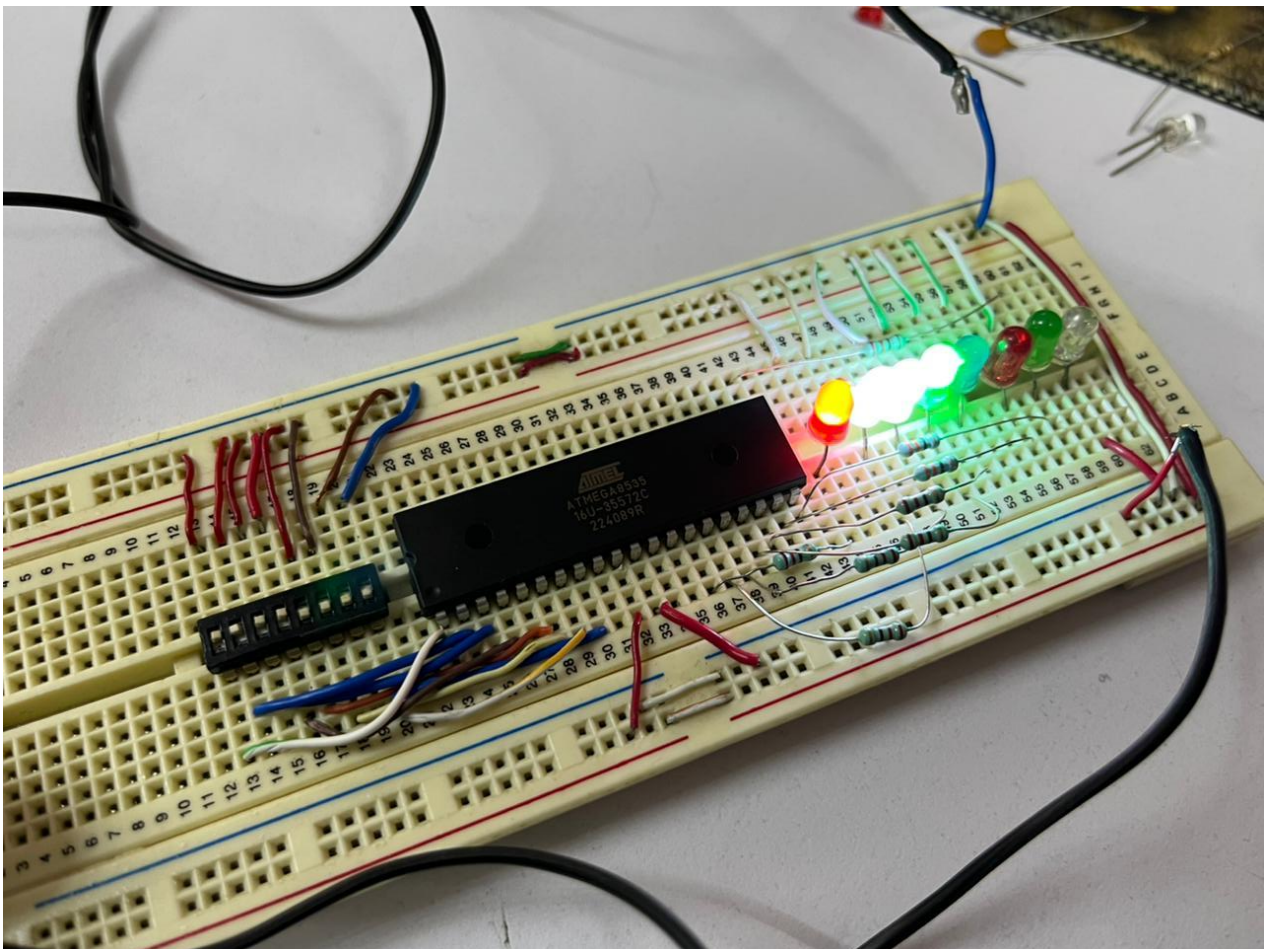
// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

while (1)
{
    // Place your code here

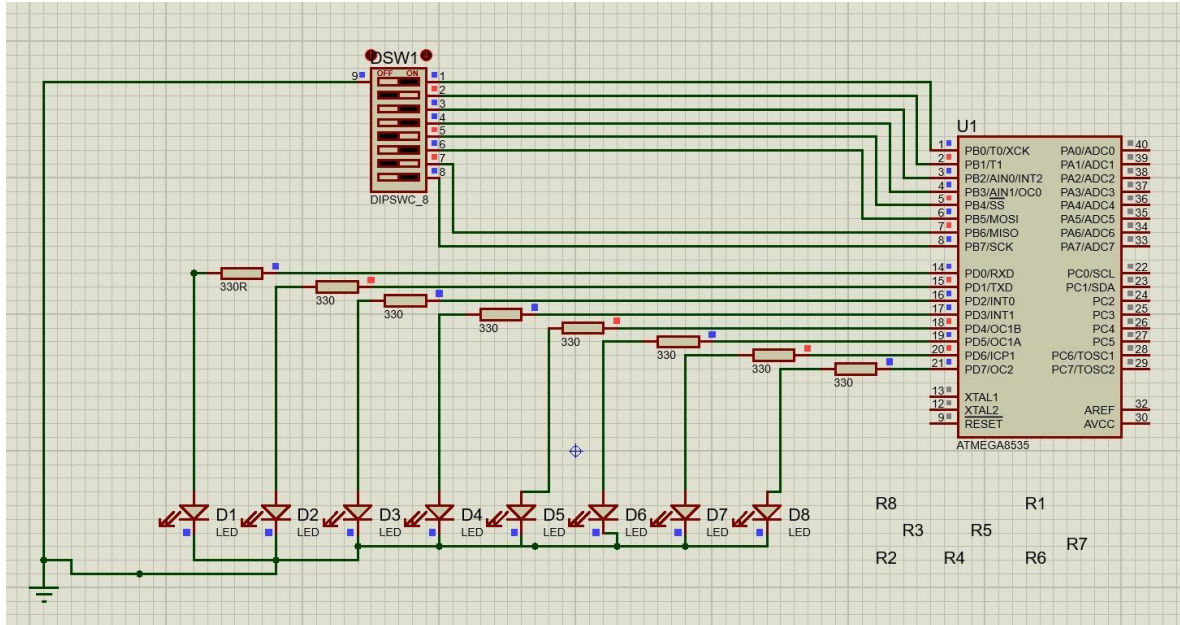
    PORTD=PINB;
}

```

Pruebas



Simulación



Conclusiones

Arizmendi Alvarado Brian

Podemos concluir que el microcontrolador ATmega8535, en efecto, tiene un sin fin de aplicaciones. En este caso, usando 8 pines como entrada, pudimos controlar 8 LEDs a la salida.

También podemos recalcar el uso de la herramienta CodeVision, programando en C, somos capaces de definir el comportamiento del microcontrolador.

Es necesario el uso de un programador para hacer llegar nuestras instrucciones al controlador.

Rodríguez López Ricardo

En conclusión puedo decir que esta práctica sirvió como un acercamiento a lo que es Visual Studio AVR y poder retomar práctica en lo que se refiere a la implementación de circuitos. También puedo decir ver la eficiencia que tiene el ATmega es bastante interesante, pues es un dispositivo que es muy fácil de programar y nos permite hacer bastantes cosas, la activación de los puertos en Visual Studio es bastante sencilla también.

Referencias Bibliográficas:

Microchip. (s.f.). ATmega8535. <https://www.microchip.com/en-us/product/ATmega8535#>

CodevisionAVR. (s.f.). High Performance ANSI C Compiler for Microchip AVR Microcontrollers. <https://www.codevision.be/>