



**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE  
CÓMPUTO.**



## **INTRODUCCIÓN A LOS MICROCONTROLADORES.**

**REPORTE PRÁCTICA 1.**

**PROFESOR: AGUILAR SANCHEZ FERNANDO**

**Grupo: 3CM6**

**Alumno: Rojas Alvarado Luis Enrique**

# Práctica 1: Uso de puertos E/S

## Introducción Teórica

*¿Qué es un microcontrolador?*

El microcontrolador es un circuito integrado de muy alta escala de integración que contiene las partes funcionales de un computador:

- CPU (Central Processor Unit o Unidad de Procesamiento Central)
- Memorias volátiles (RAM), para datos
- Memorias no volátiles( ROM, PROM, EPROM) para escribir el programa
- Líneas de entrada y salida para comunicarse con el mundo exterior.
- Algunos periféricos (comunicación serial, temporizador, convertidor A/D, etc)

Es decir, el microcontrolador es un computador integrado en un solo chip [1].

*¿Qué es CodeVision?*

Es un Entorno de Desarrollo Integrado (IDE) para la serie microcontroladores de Atmel AVR. Sus principales características son:

- Un compilador C compatible con todas las funciones ANSI.
- Editor flexible y completo con resaltado de sintaxis.
- Asistente de generación automática de código con soporte periférico.
- Interfaz de programación para programadores estándar.
- Bibliotecas y ejemplos para una serie de dispositivos periféricos externos.
- Programa de terminal integrado para comunicaciones seriales.
- Navegador de código, Árbol de llamadas, pestaña Información de código y muchas otras funciones [2].

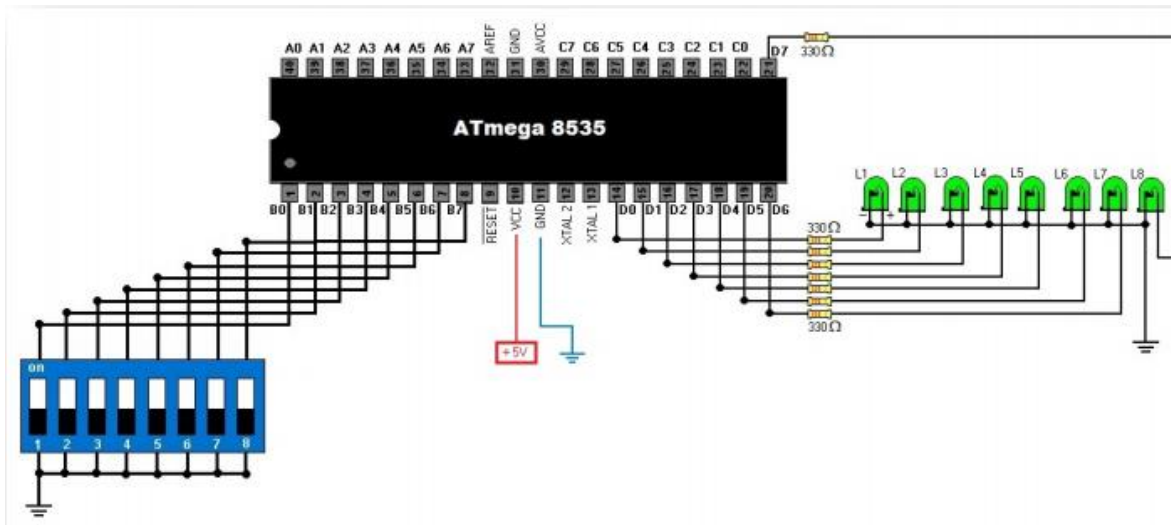
*Características generales del ATmega8535/8535L*

- Microcontrolador AVR de 8 bit de alto rendimiento y bajo consumo.
- Arquitectura RISC avanzada.
  - 130 instrucciones. La mayoría de un simple ciclo de clock de ejecución.
  - 32 x 8 registros de trabajo de propósito general.
  - Capacidad de procesamiento de unos 16 MIPS a 16 MHz.
  - Funcionamiento estático total.
  - Multiplicador On-Chip de 2 ciclos
- Memorias de programa y de datos no volátiles.
  - 8K bytes de FLASH autoprogramable en sistema.
  - Resistencia: 1.000 ciclos de escritura / borrado.

- Sección de código añadida opcional con bits de bloqueo independientes.
- Programación en sistema con el programa añadido On-Chip.
- Operación de lectura durante la escritura.
- 512 bytes de EEPROM.
- Resistencia: 100.000 ciclos de escritura / borrado.
- 512 bytes de SRAM interna.
- Bloqueo (cerradura) programable para la seguridad del software.
- Características de los periféricos.
  - Dos Timer/Contadores de 8 bits con prescaler separado y modo comparación.
  - Un Timer/Contador de 16 bits con prescaler separado, modo comparación y modo de captura.
  - Comparador analógico On-Chip.
  - Timer watchdog programable con oscilador separado On-Chip.
  - Interface serie SPI maestro/esclavo.
  - USART serie programable.
  - Contador en tiempo real con oscilador separado.
  - ADC de 10 bit y 8 canales.
  - 8 canales de terminación simple
  - 7 canales diferenciales sólo en el encapsulado TQFP.
  - 2 canales diferenciales con ganancia programable a 1x, 10x o 200 sólo en el encapsulado TQFP.
  - 4 canales de PWM.
  - Interface serie de dos hilos orientada a byte.
- Características especiales del microcontrolador.
  - Reset de Power-on y detección de Brown-out programable.
  - Oscilador RC interno calibrado.
  - Fuentes de interrupción externas e internas.
  - 6 modos de descanso: Idle, reducción de ruido ADC, Power-save, Power-down, Standby y Standby extendido.
- I/O y encapsulados.
  - 32 líneas de I/O programables.
  - PDIP de 40 pines, TQFP y MLF de 44 pines.
- Tensiones de funcionamiento.
  - 2.7 - 5.5V (ATmega8535L).
  - 4.5 - 5.5V (ATmega8535).
- Niveles de velocidad.
  - 0 - 8 MHz (ATmega8535L).
  - 0 - 16 MHz (ATmega8535). [3]

## Desarrollo Experimental

1. Realiza un programa para programar el Puerto B como entrada y escribir la información en el Puerto D activándolo como salida, recuerde activar las resistencias de Pull-up del puerto B para colocar solo el Dipswitch.



## Estructura del programa

```

1.  /*****
2.  This program was created by the
3.  CodeWizardAVR V2.60 Evaluation
4.  Automatic Program Generator
5.  © Copyright 1998-2012 Pavel Haiduc, HP InfoTech s.r.l.
6.  http://www.hpinfotech.com
7.
8.  Project :
9.  Version :
10. Date   : 10/10/2020
11. Author :
12. Company :
13. Comments:
14.
15.
16. Chip type           : ATmega8535
17. Program type        : Application
18. AVR Core Clock frequency: 1.000000 MHz
19. Memory model        : Small
20. External RAM size    : 0
21. Data Stack size     : 128
22. *****/
23.
24. #include <mega8535.h>
25.
26. // Declare your global variables here
27.
28. void main(void)
29. {
30. // Declare your local variables here
31.
32. // Input/Output Ports initialization
33. // Port A initialization
34. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
35. DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) | (0<<DDA1) | (0<<DDA0);
36. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T

```

```

37. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
38.
39. // Port B initialization
40. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
41. DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
42. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
43. PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) | (1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);
44.
45. // Port C initialization
46. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
47. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
48. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
49. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
50.
51. // Port D initialization
52. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
53. DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) | (1<<DDD1) | (1<<DDD0);
54. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
55. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);
56.
57. // Timer/Counter 0 initialization
58. // Clock source: System Clock
59. // Clock value: Timer 0 Stopped
60. // Mode: Normal top=0xFF
61. // OC0 output: Disconnected
62. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
63. TCNT0=0x00;
64. OCR0=0x00;
65.
66. // Timer/Counter 1 initialization
67. // Clock source: System Clock
68. // Clock value: Timer1 Stopped
69. // Mode: Normal top=0xFFFF
70. // OC1A output: Disconnected
71. // OC1B output: Disconnected
72. // Noise Canceler: Off
73. // Input Capture on Falling Edge
74. // Timer1 Overflow Interrupt: Off
75. // Input Capture Interrupt: Off
76. // Compare A Match Interrupt: Off
77. // Compare B Match Interrupt: Off
78. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
79. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
80. TCNT1H=0x00;
81. TCNT1L=0x00;
82. ICR1H=0x00;
83. ICR1L=0x00;
84. OCR1AH=0x00;
85. OCR1AL=0x00;
86. OCR1BH=0x00;

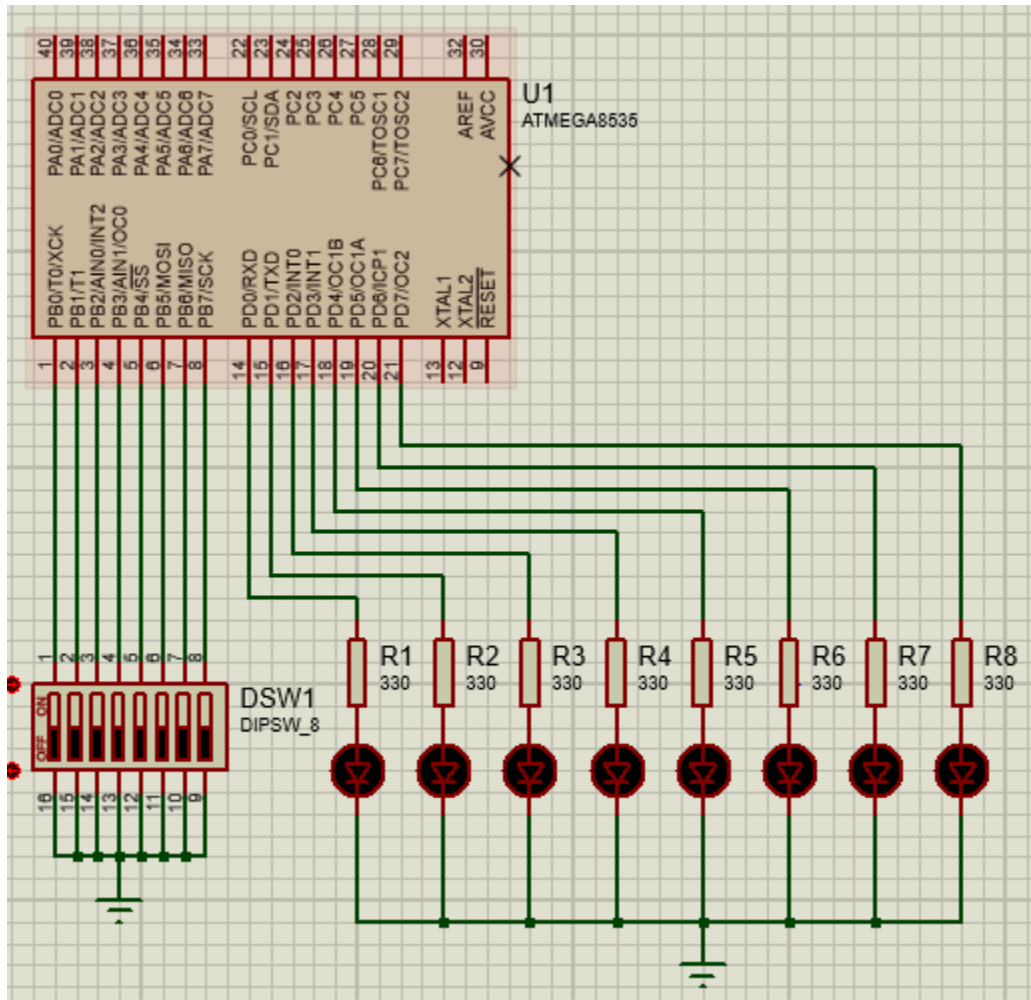
```

```

87. OCR1BL=0x00;
88.
89. // Timer/Counter 2 initialization
90. // Clock source: System Clock
91. // Clock value: Timer2 Stopped
92. // Mode: Normal top=0xFF
93. // OC2 output: Disconnected
94. ASSR=0<<AS2;
95. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) |
    (0<<CS20);
96. TCNT2=0x00;
97. OCR2=0x00;
98.
99. // Timer(s)/Counter(s) Interrupt(s) initialization
100. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
    (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
101.
102. // External Interrupt(s) initialization
103. // INT0: Off
104. // INT1: Off
105. // INT2: Off
106. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
107. MCUCSR=(0<<ISC2);
108.
109. // USART initialization
110. // USART disabled
111. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<U
    CS22) | (0<<RXB8) | (0<<TXB8);
112.
113. // Analog Comparator initialization
114. // Analog Comparator: Off
115. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) |
    (0<<ACIS1) | (0<<ACIS0);
116. SFIOR=(0<<ACME);
117.
118. // ADC initialization
119. // ADC disabled
120. ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<AD
    PS2) | (0<<ADPS1) | (0<<ADPS0);
121.
122. // SPI initialization
123. // SPI disabled
124. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA)
    | (0<<SPR1) | (0<<SPR0);
125.
126. // TWI initialization
127. // TWI disabled
128. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
129.
130. while (1)
131. {
132.     // Place your code here
133.     PORTD=PINB;
134. }
135.

```

## Simulación



## Conclusiones Individuales

En esta práctica podemos hacer uso del ATMEGA 8535, el cual es un microcontrolador que contiene todos los aditamentos mencionados en el apartado teórico y nos permite ejecutar un programa que cumpla con ciertas necesidades. También hacemos uso de la herramienta de codeVision para facilitarnos de una manera bastante robusta la asignación de pines y evitar configurar registros a mano, el programa ya lo hace solo y es cuestión de agregar el programa principal que en este caso es la asignación de lo que sea de entrada se vea reflejado en un puerto de salida configurado previamente. En este caso se trata de una configuración de puertos de entrada y salida sencillo.

## Bibliografía

- [1] Z. Cucho, F. Orihuela, R. Sánchez y R. Laureano, «unrobotica,» 2007. [En línea]. Available: [http://www.unrobotica.com/manuales/SESSION\\_1\\_ATMEGA8.pdf](http://www.unrobotica.com/manuales/SESSION_1_ATMEGA8.pdf). [Último acceso: 10/10/2020].
- [2] Arctan Systems Ltd., «arctan,» 2009. [En línea]. Available: <http://www.arctan.ca/product/cvavr.html>. [Último acceso: 10/10/2020].
- [3] Ehu., «Características generales del ATmega8535/8535L» [En línea]. Available: [http://www.sc.ehu.es/sbweb/webcentro/automatica/web\\_avr/archivos/Otros%20AVRs/ATmega/ATmega8535.htm](http://www.sc.ehu.es/sbweb/webcentro/automatica/web_avr/archivos/Otros%20AVRs/ATmega/ATmega8535.htm). [Último acceso: 10/10/2020].