



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO

**Práctica 2**  
**Contador Ascendente/Descendente**

***Alumno:***

Amador Nava Miguel Ángel

***Profesor:***

Aguilar Sanchez Fernando

***Grupo:***

3CM5

***Asignatura:***

Introducción a los Microcontroladores



# Índice general

Introducción . . . . .	3
Objetivo . . . . .	3
Material y equipo . . . . .	3
Planteamiento del problema . . . . .	3
Análisis teórico . . . . .	4
Diseño del circuito . . . . .	9
Análisis práctico . . . . .	10
Comparaciones . . . . .	10
Conclusiones . . . . .	11
Código en lenguaje C . . . . .	12
Referencias . . . . .	15

# Introducción

Un contador es un circuito en el que sus salidas siguen una secuencia fija que cuando acaba vuelve a empezar, o circuitos que reciben sus datos en forma serial ordenados en distintos intervalos de tiempo.

Los pulsos de entrada pueden ser pulsos de reloj u originarse en una fuente externa y pueden ocurrir a intervalos de tiempo fijos o aleatorios.

El número de salidas limita el máximo número que se puede contar.[1] El Atmega8535 tiene 4 puertos de 8 bits lo que nos permite contar de 0 a 255.

# Objetivo

Al término de la sesión, los integrantes del equipo contarán con la habilidad de realizar un contador descendente y ascendente utilizando funciones de retardo.

# Material y equipo

- CodeVision AVR
- AVR Studio 4
- Microcontrolador ATmega 8535
- 16 LED's
- 16 Resistores de  $330\ \Omega$  a  $\frac{1}{4}\ W$

# Planteamiento del problema

Diseñar un contador binario ascendente en el Puerto B (0-255), y un contador descendente en el Puerto D (255-0). Use un retardo de un segundo para visualizar la información.

# Análisis teórico

Los contadores serán de un 1 byte sin signo por lo que el número máximo es  $2^n - 1$  en este caso  $n = 8$  por lo que la cuenta será de 0 a 255 y viceversa. Los puertos utilizados en el Atmega 8535 serán B y D, donde B es el contador ascendente y D el descente, la representación será en binario, para que le ojo humano pueda visualizar las transiciones se necesita un retardo.

Número	8	7	6	5	4	3	2	1
255	1	1	1	1	1	1	1	1
130	1	0	0	0	0	0	1	0
44	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0

Cuadro 1: Ejemplo de representación binaria

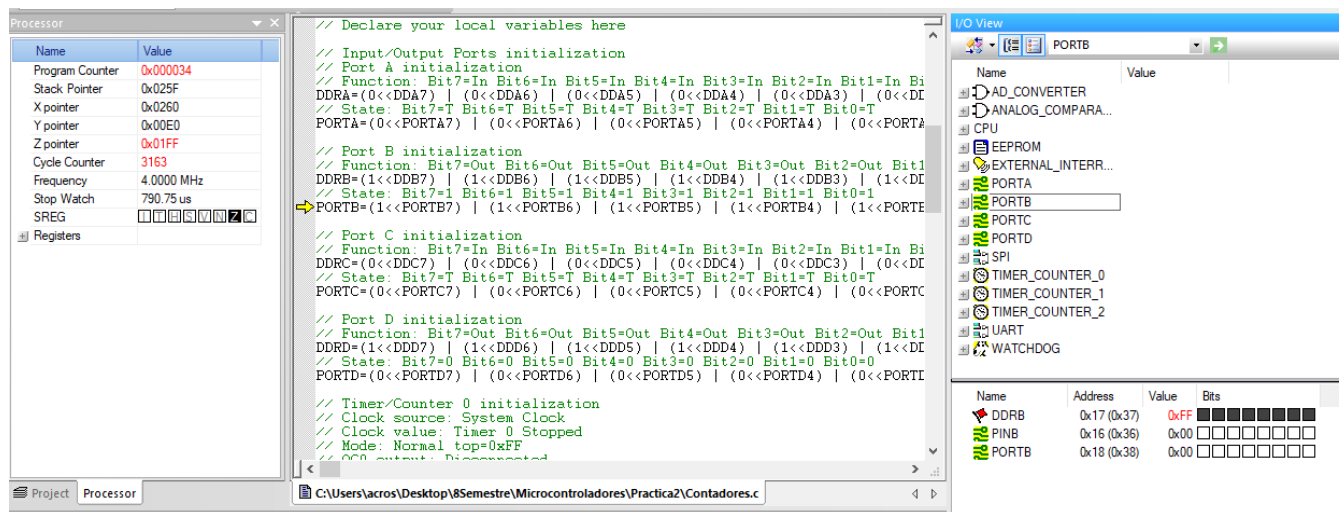


Figura 1: Inicialización del puerto B

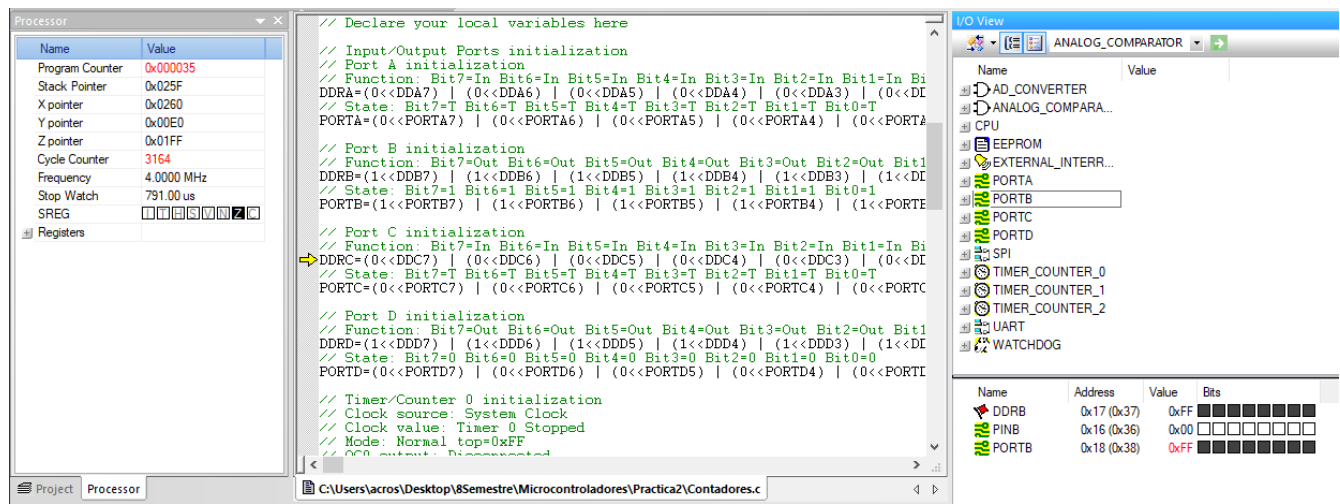


Figura 2: Inicialización del puerto B

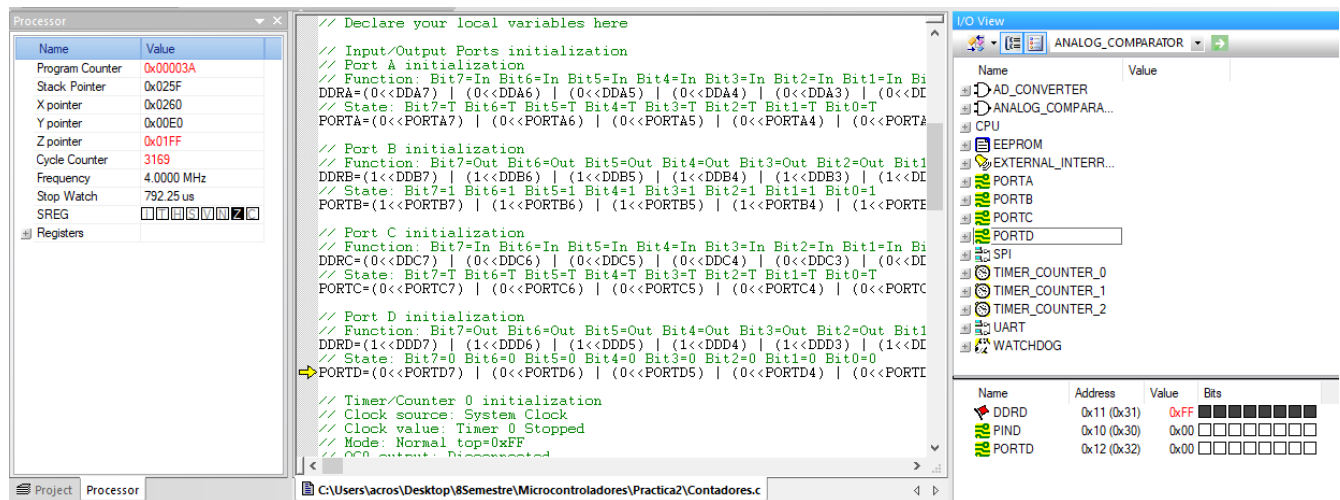


Figura 3: Inicialización del puerto D

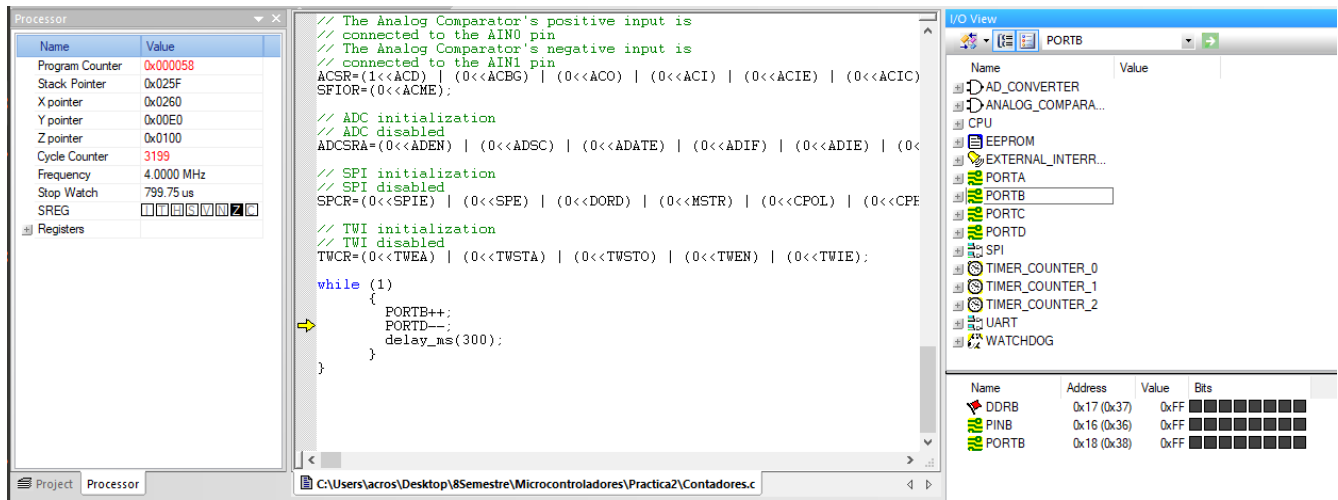


Figura 4: Funcionamiento del programa en el puerto B

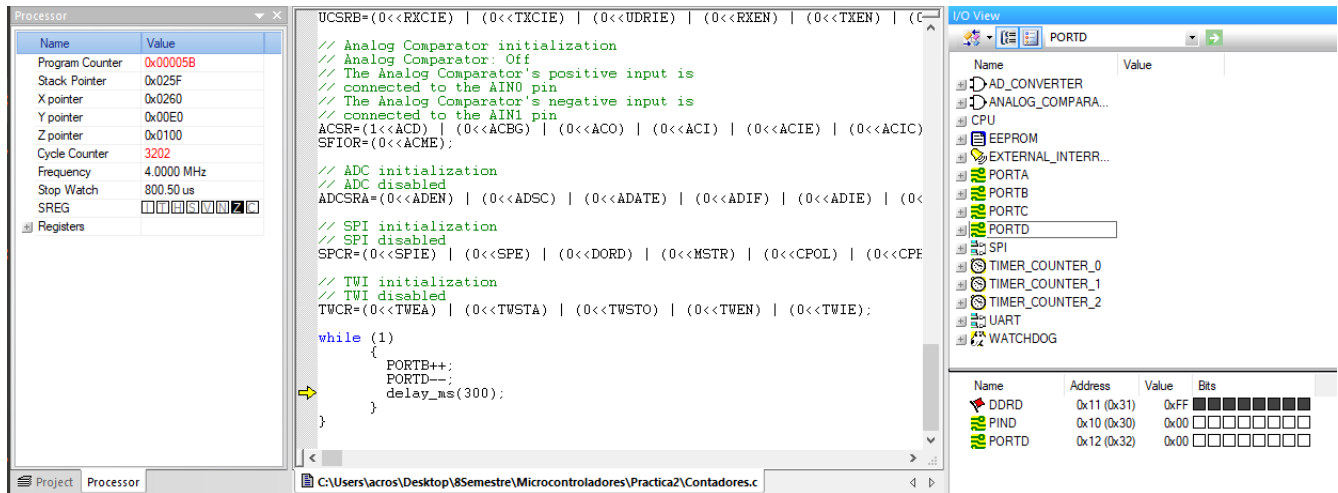


Figura 5: Funcionamiento del programa en el puerto D

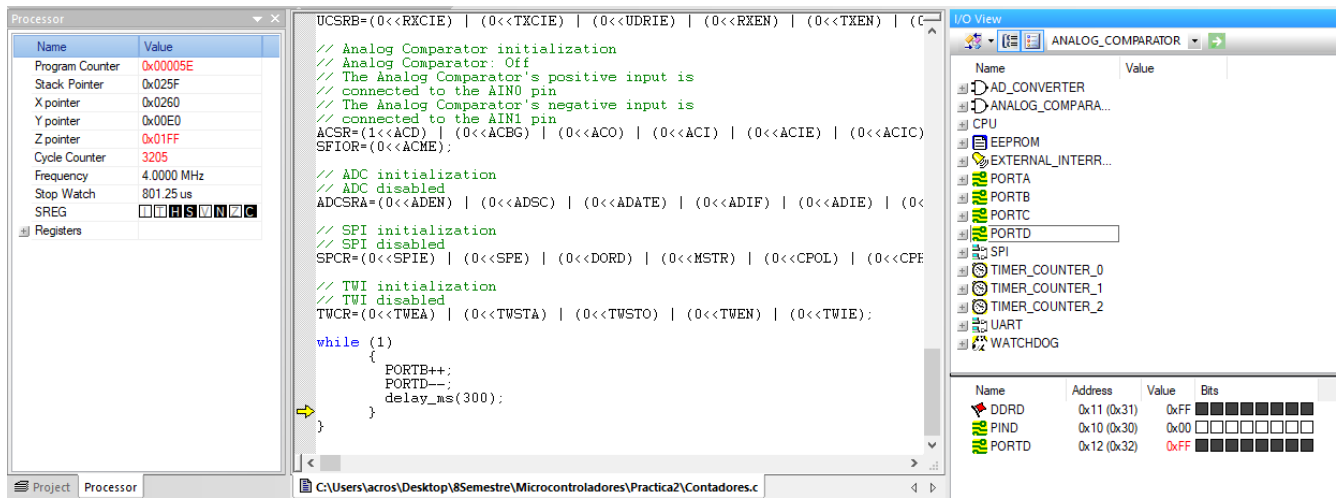


Figura 6: Funcionamiento del programa en el puerto D

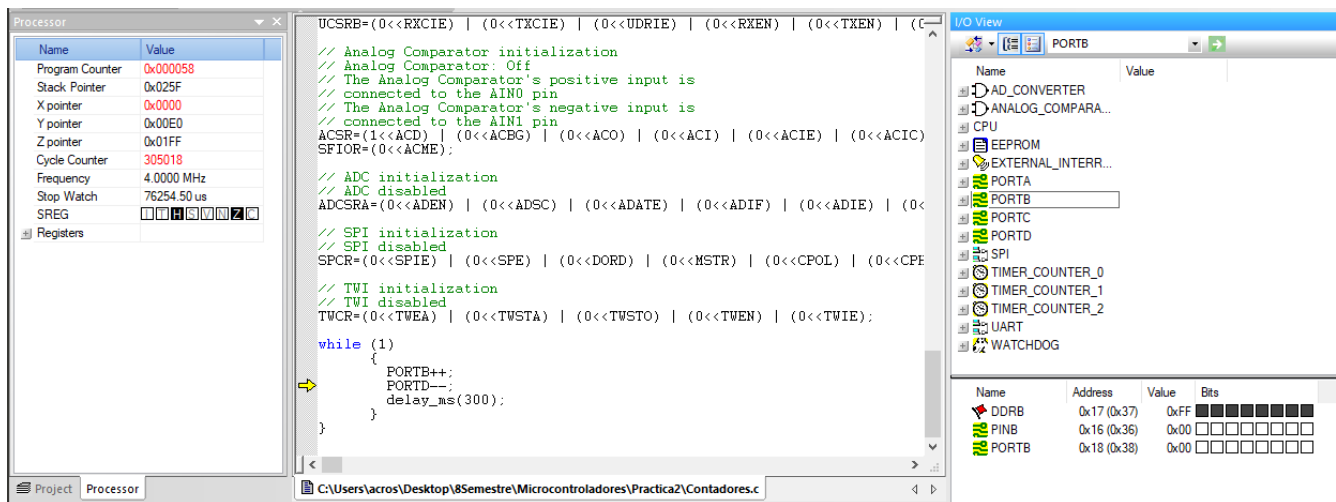


Figura 7: Funcionamiento del programa en el puerto B

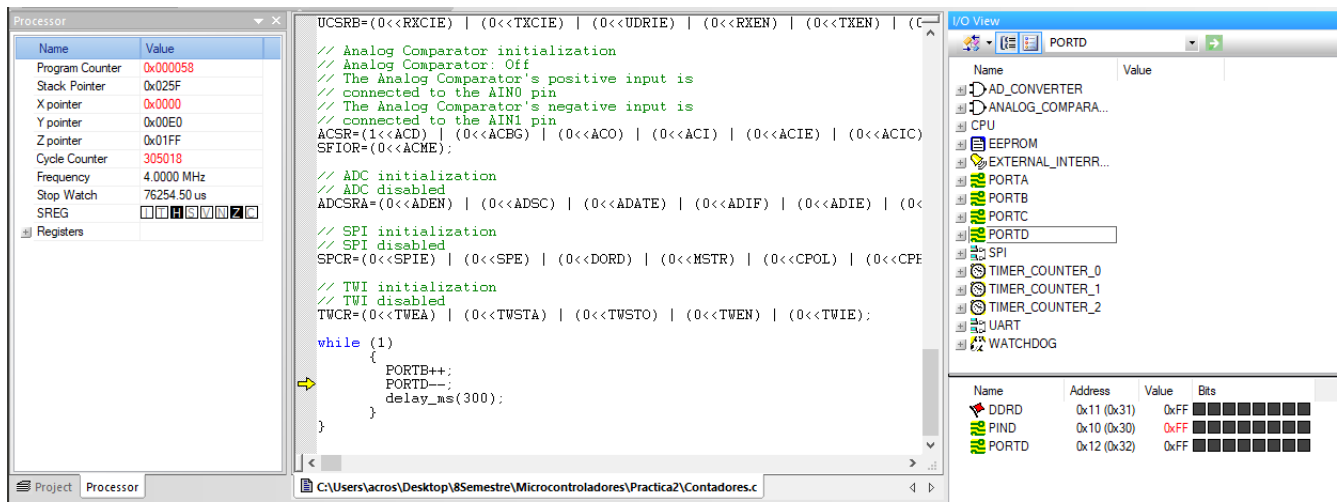


Figura 8: Funcionamiento del programa en el puerto D

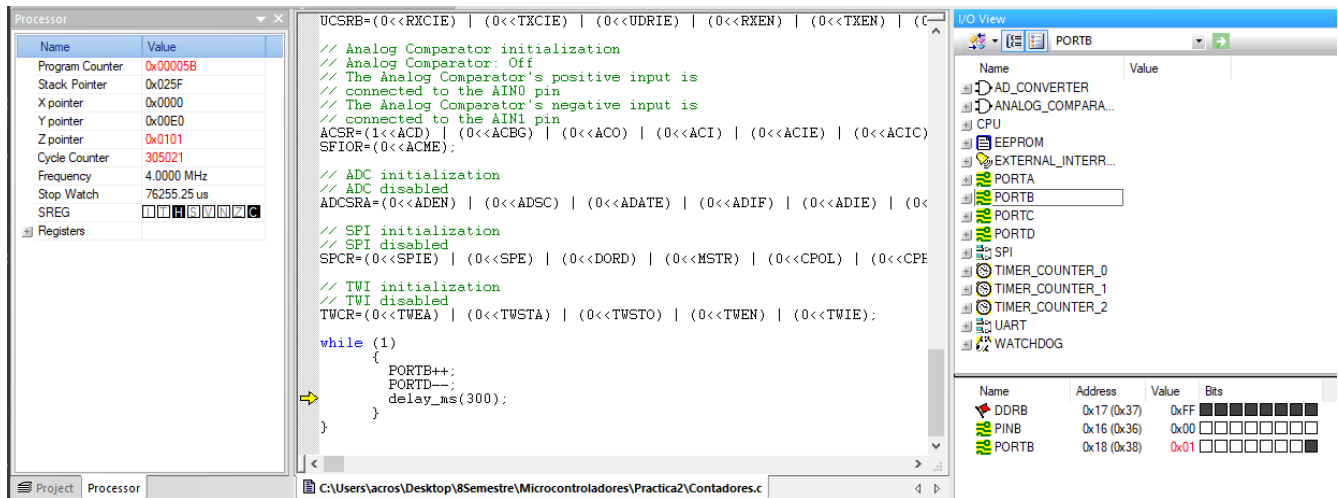


Figura 9: Funcionamiento del programa en el puerto B



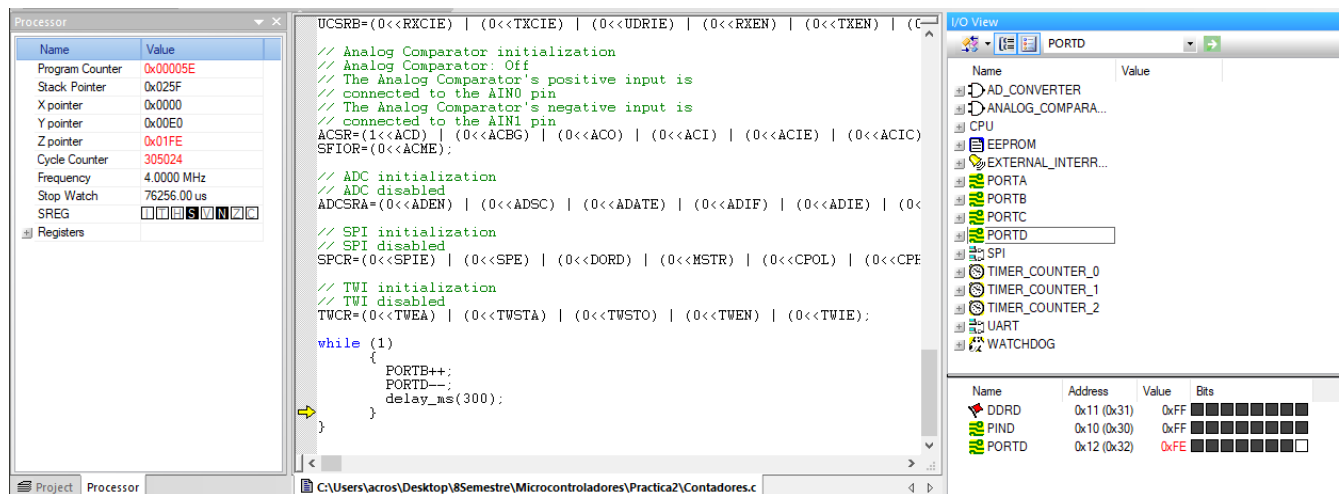


Figura 10: Funcionamiento del programa en el puerto B

## Diseño del circuito

En la Figura 11 se muestra el circuito para la implementación de la práctica.

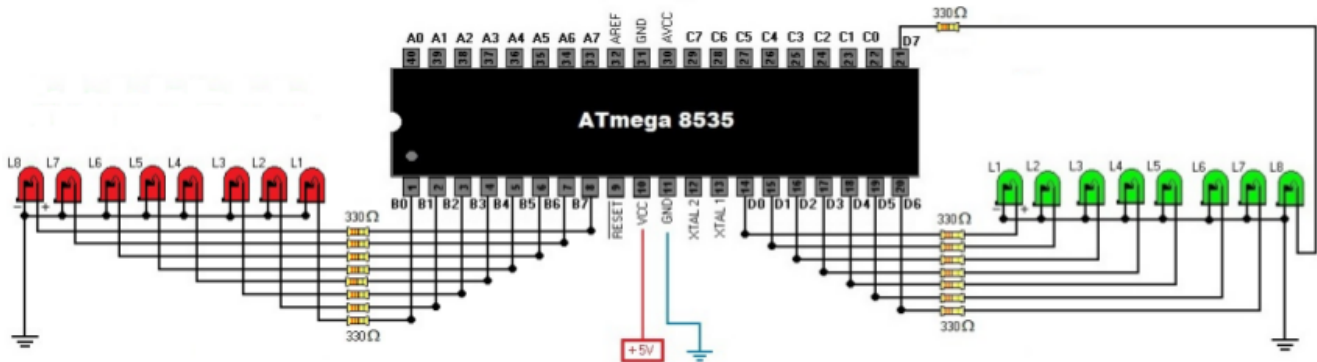


Figura 11: Circuito para el contador descendente y ascendente

# Análisis práctico

En la Figura 12 se puede ver el circuito implementado, siendo alimentado con el grabador de AVR's, la figura muestra el contador ascendente (LED's rojos) y descendente(LED's verdes).

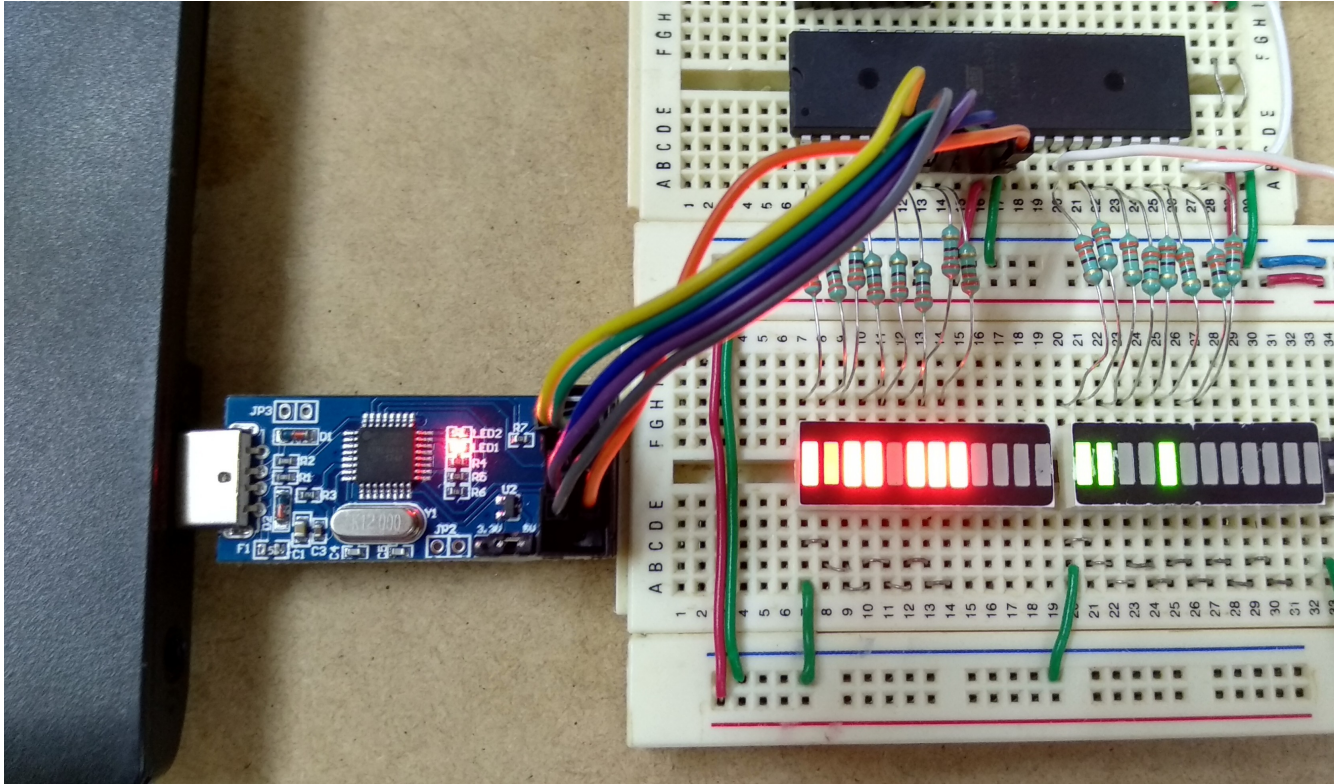


Figura 12: Contador Ascendente y Descendente implementado

## Comparaciones

Los resultados en lo práctico son los esperados en lo teórico, la velocidad puede verse comprometida con los resultados pues la velocidad no depende de un cristal sino de un circuito RC que con la variación de la temperatura hace variar la velocidad del reloj interno del microcontrolador.

# Conclusiones

Al poder manejar los puertos con lenguaje C, la programación se facilita y permite incrementar y decrementar el valor de un puerto como cualquier variable en el lenguaje, lo que nos permite visualizar como los puertos cambian de valor de 0 a 255, poniendo en alto o en bajo los pines de los puertos.

# Código en lenguaje C

```
1  /*****
2  This program was created by the CodeWizardAVR V3.34
3  Automatic Program Generator
4  Copyright 1998-2018 Pavel Haiduc, HP InfoTech s.r.l.
5  http://www.hpinfotech.com
6
7  Project : Introduccion a los Microcontroladores
8  Version : 1.0
9  Date    : 25/01/2019
10 Author  : Amador Nava Miguel Angel
11 Company : Escuela Superior de Computo
12 Comments: Prctica 2 Contador Ascendente Descendente
13
14
15 Chip type           : ATmega8535
16 Program type        : Application
17 AVR Core Clock frequency: 1.000000 MHz
18 Memory model        : Small
19 External RAM size    : 0
20 Data Stack size     : 128
21 *****/
22
23 #include <mega8535.h>
24 #include <delay.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)
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) |
38
39 // Port B initialization
40 // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
41 DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2)
```

```
42 // State: Bit7=1 Bit6=1 Bit5=1 Bit4=1 Bit3=1 Bit2=1 Bit1=1 Bit0=1
43 PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) |
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) |
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) |
50
51 // Port D initialization
52 // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out
53 DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) |
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) |
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) |
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) |
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<<OCIE0A) | (0<<OCIE0B) | (0<<TOIE1);
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<<UCSZA) | (0<<UCSZB);
112
113 // Analog Comparator initialization
114 // Analog Comparator: Off
115 // The Analog Comparator's positive input is
116 // connected to the AIN0 pin
117 // The Analog Comparator's negative input is
118 // connected to the AIN1 pin
119 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIF);
120 SFIOR=(0<<ACME);
121
122 // ADC initialization
123 // ADC disabled
124 ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPR) | (0<<ADIFR) | (0<<ADIF0);
125
126 // SPI initialization
127 // SPI disabled
128 SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR0) | (0<<SPR1) | (0<<SPR2);
129

```

```
130 // TWI initialization
131 // TWI disabled
132 TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
133
134 while (1)
135     {
136         PORTB++;
137         PORTD--;
138         delay_ms(300);
139     }
140 }
```

# Bibliografía

- [1] R. González. Contadores. [Online]. Available: <http://personales.unican.es/manzanom/Planantiguo/EDigitalI/CONTG5.pdf>