



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



INTRODUCCIÓN A LOS MICROCONTROLADORES

PROFESOR:

Aguilar Sanchez Fernando

ALUMNOS:

Arruti Sánchez Alondra
Carrillo Soto Cristian Eduardo

Grupo: 3CM14

Fecha de Entrega: 07/10/22

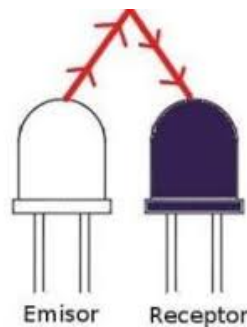
PRÁCTICA 7

Contador de 00 a 99 activado por IR

Introducción

En electrónica digital, el sensor infrarrojo es un dispositivo optoelectrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión. Todos los cuerpos emiten una cierta cantidad de radiación, esta resulta invisible para nuestros ojos, pero no para estos aparatos electrónicos, ya que se encuentran en el rango del espectro justo por debajo de la luz visible.

Los rayos infrarrojos (IR) entran en el fototransistor donde encontramos un material piroeléctrico, natural o artificial, normalmente formando una lámina delgada dentro del nitrato de galio [$\text{Ga}(\text{NO}_3)_3$], nitrato de Cesio (CsNO_3), derivados de la fenilhidrazina, y ftalocianina de cobalto. Normalmente están integrados en diversas configuraciones (1,2,4 píxeles de material piroeléctrico).



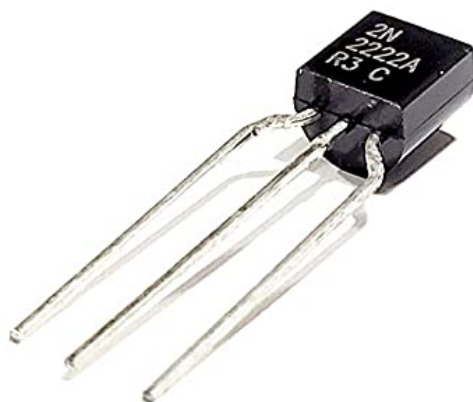
Sensor IR

En el caso de parejas se acostumbra a dar polaridades opuestas para trabajar con un amplificador diferencial, provocando la auto cancelación de los incrementos de energía de IR y el desacoplamiento del equipo.

Los sensores infrarrojos están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

Para aplicaciones domésticas, los sensores infrarrojos se utilizan en electrodomésticos de línea blanca tales como hornos microondas, por ejemplo, para permitir la medición de la distribución de la temperatura en el interior. Estos dispositivos se usan también en el control climático de la casa para detectar oscilaciones de la temperatura en un local. Este planteamiento permite que el sistema de climatización reaccione antes que la temperatura del local varíe. Los sensores infrarrojos también se pueden utilizar como sensores de gas.

El 2N2222, también identificado como PN2222, es un transistor bipolar NPN de baja potencia de uso general.



Transistor 2N2222

Sirve tanto para aplicaciones de amplificación como de conmutación. Puede amplificar pequeñas corrientes a tensiones pequeñas o medias; por lo tanto, sólo puede tratar potencias bajas (no mayores de medio vatio). Puede trabajar a frecuencias medianamente altas.

Por todas esas razones, es un transistor de uso general, frecuentemente utilizados en aplicaciones de radio por los constructores aficionados de radios. Es uno de los transistores oficiales utilizados en el BITX. Su versatilidad ha permitido incluso al club de radioaficionados Norcal lanzar en 1999 un desafío de construir un transceptor de radio utilizando únicamente hasta 22 ejemplares de este transistor y ningún circuito integrado.

Códigos

1. Código de la configuración de los periféricos

```
1. // I/O Registers definitions
2. #include <mega8535.h>
3. #include <delay.h>
4. #define boton PIND.0
5. const char tabCatodo [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7c,0x07,0x7f,0x6f};
6. unsigned char cont, cont2;
7. bit Bx;
8. bit By;
9.
10. // Declare your global variables here
11.
12. void main(void)
13. {
14. // Declare your local variables here
15.
16. // Input/Output Ports initialization
17. // Port A initialization
18. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
19. DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) |
    (1<<DDA0);
20. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
21. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) |
    (0<<PORTA1) | (0<<PORTA0);
22.
23. // Port B initialization
24. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
25. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) |
    (1<<DDB0);
26. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
27. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) |
    (0<<PORTB1) | (0<<PORTB0);
28.
29. // Port C initialization
30. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
31. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) |
    (0<<DDC0);
32. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
33. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) |
    (0<<PORTC1) | (0<<PORTC0);
34.
```

```

35. // Port D initialization
36. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
37. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) |
    (0<<DDD0);
38. // State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
39. PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) | (1<<PORTD2) |
    (1<<PORTD1) | (1<<PORTD0);
40.
41. // Timer/Counter 0 initialization
42. // Clock source: System Clock
43. // Clock value: Timer 0 Stopped
44. // Mode: Normal top=0xFF
45. // OCO output: Disconnected
46. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
47. TCNT0=0x00;
48. OCR0=0x00;
49.
50. // Timer/Counter 1 initialization
51. // Clock source: System Clock
52. // Clock value: Timer1 Stopped
53. // Mode: Normal top=0xFFFF
54. // OC1A output: Disconnected
55. // OC1B output: Disconnected
56. // Noise Canceler: Off
57. // Input Capture on Falling Edge
58. // Timer1 Overflow Interrupt: Off
59. // Input Capture Interrupt: Off
60. // Compare A Match Interrupt: Off
61. // Compare B Match Interrupt: Off
62. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
63. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
64. TCNT1H=0x00;
65. TCNT1L=0x00;
66. ICR1H=0x00;
67. ICR1L=0x00;
68. OCR1AH=0x00;
69. OCR1AL=0x00;
70. OCR1BH=0x00;
71. OCR1BL=0x00;
72.
73. // Timer/Counter 2 initialization
74. // Clock source: System Clock
75. // Clock value: Timer2 Stopped
76. // Mode: Normal top=0xFF
77. // OC2 output: Disconnected

```

```

78. ASSR=0<<AS2;
79. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
80. TCNT2=0x00;
81. OCR2=0x00;
82.
83. // Timer(s)/Counter(s) Interrupt(s) initialization
84. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0)
    | (0<<TOIE0);
85.
86. // External Interrupt(s) initialization
87. // INT0: Off
88. // INT1: Off
89. // INT2: Off
90. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
91. MCUCSR=(0<<ISC2);
92.
93. // USART initialization
94. // USART disabled
95. UCSRB=(0<<RXCIEN) | (0<<TXCIEN) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) |
    (0<<TXB8);
96.
97. // Analog Comparator initialization
98. // Analog Comparator: Off
99. // The Analog Comparator's positive input is
100.    // connected to the AIN0 pin
101.    // The Analog Comparator's negative input is
102.    // connected to the AIN1 pin
103.    ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) |
    (0<<ACIS0);
104.    SFIOR=(0<<ACME);
105.
106.    // ADC initialization
107.    // ADC disabled
108.    ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1)
    | (0<<ADPS0);
109.
110.    // SPI initialization
111.    // SPI disabled
112.    SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) |
    (0<<SPR0);
113.
114.    // TWI initialization
115.    // TWI disabled
116.    TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
117.

```

2. Código del programa principal en C

```
while (1)
{
    if (boton==0)
        Bx=1;
    else
        Bx=0;

    if ((Bx==0) && (By==1)) //Cambio de 0 a 1
    {
        cont++;
        if (cont==10)
        {
            cont=0;
            cont2++;
            if (cont2==10)
                cont2=0;
        }
        delay_ms(40);
    }

    PORTA=tabCatodo [cont];
    PORTB=tabCatodo [cont2];

    if (Bx==1) { //Cambio de 1 a 0
        By=1;
        delay_ms(40);
    }

    if ((Bx==0) && (By==1))
        By=0;

    // Place your code here
};
```

Circuitos

1. Circuito simulado en Proteus

En la figura 1 se puede observar el circuito dentro del simulador, donde se utiliza el sensor infrarrojo PC817 que ofrece proteus, así como un transistor 2N2222. Cabe mencionar que la forma en la que se agruparon los displays no son como en el diagrama propuesto por el profesor, puesto que al tenerlos juntos se visualiza de mejor manera la cifra que generan dichos displays.

Un punto importante que se debe tomar en cuenta es que ahora los cambios en los valores se presentan en cuanto se detecta la baja del push button para el caso de la simulación puesto que la implementación se diseñó tomando como base que se tienen cambios de 0 a 1, sin embargo, esto no ocurre debido a la forma en que se armó el circuito dentro de la simulación, generando que el proceso se invirtiera y por ende ahora se incrementan en las bajadas y no en las subidas.

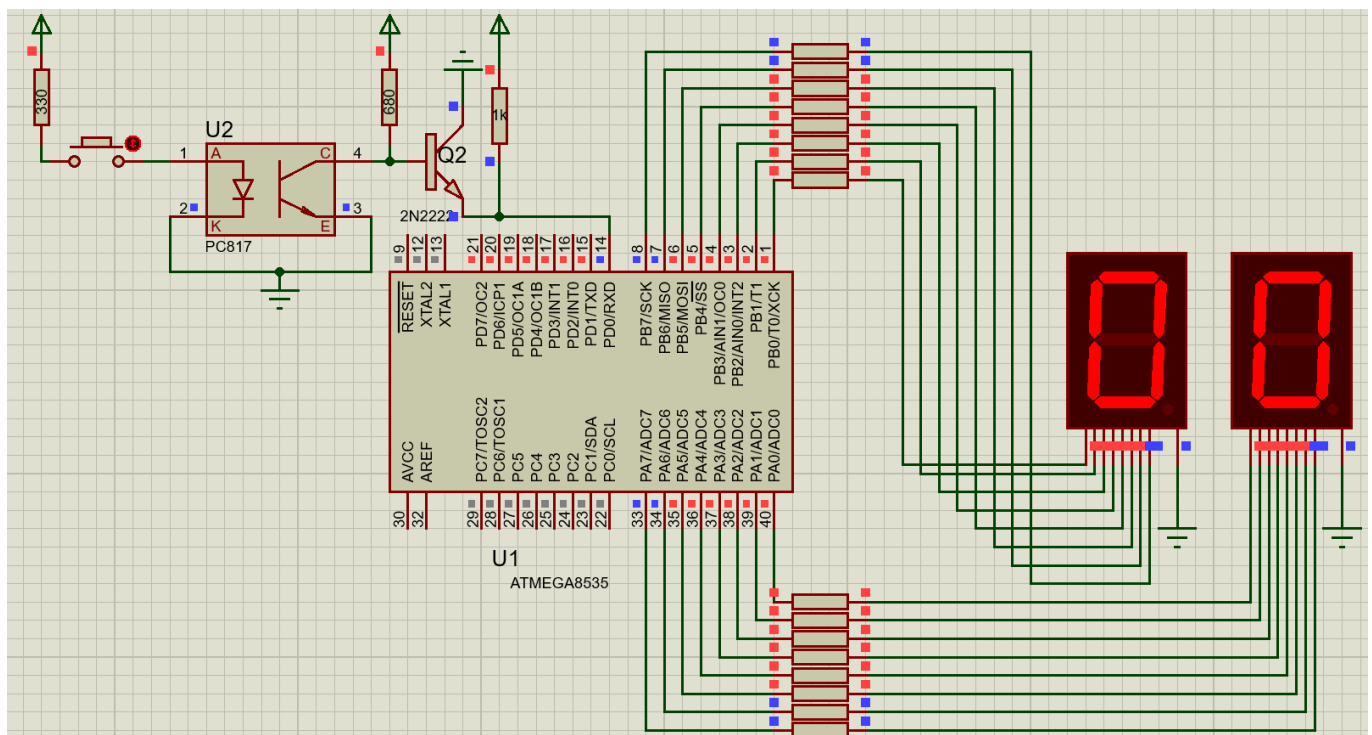


Figura. 1 Inicio del contador

En la figura 2 se muestra el último valor que el contador es capaz de mostrar, para posteriormente reiniciar el conteo a partir de 0.

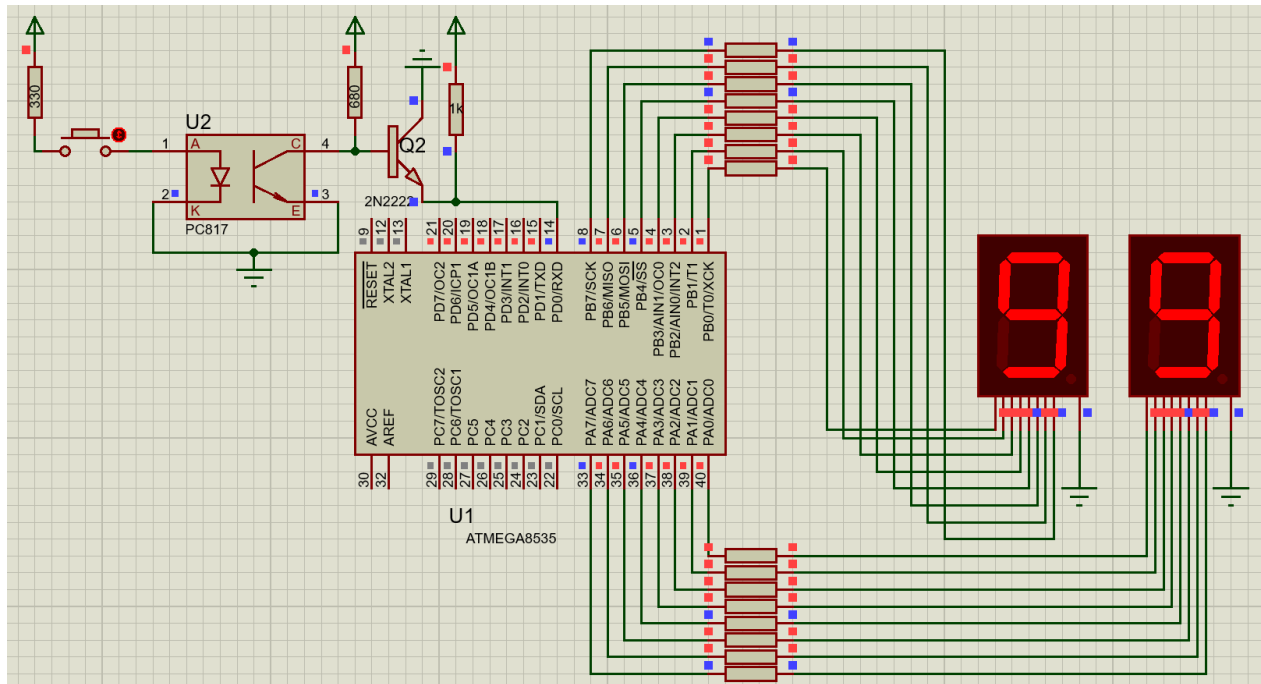


Figura. 2 Último valor del contador

1. Circuito armado en la Proto

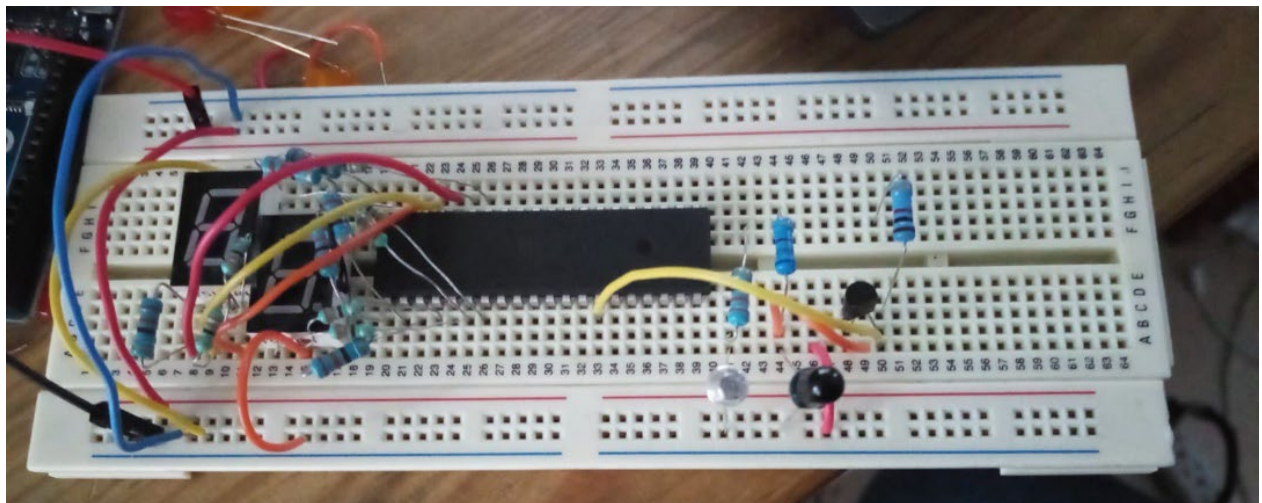


Figura. 3. Sensor IR Proto

Conclusiones

Para la realización de esta práctica simplemente se reutilizo el código que se había trabajado en las últimas tres prácticas puesto que en el se resuelven los problemas de ruido y rebotes, además de que realiza el trabajo de un contador por lo que únicamente se adapto una variable más que se incrementa conforme el primer contador llega al valor máximo soportado y dicha variable se muestra en un display generando el resultado deseado.

Por otro lado, al armar el circuito en el simulador se presentaron algunas complicaciones en cuanto al sensor IR puesto que se desconocía su nombre dentro del simulador, tras una corta investigación se optó por el componente PC817 que es un optoacoplador formado por dos componentes, un diodo emisor de luz infrarrojo y un fototransistor que es capaz de recibir la señal infrarroja, este no puede operar de la misma forma que en el circuito físico donde se espera una interrupción con un objeto (spinner) por lo que se adapta un botón para visualizar el funcionamiento.

Finalmente, tras la realización de la práctica fue posible comprender porque el circuito paso de trabajar con un estado alto a un estado bajo; puesto que el programa detecta los flancos de subida, el cambio radica en que el transistor es NPN por lo que se invierte la configuración o en forma simple el valor que entra en el pinD pasa por un not antes de entrar.

Arruti Sánchez Alondra

Esta práctica se me hizo muy entretenida de hacer ya que cuando iba en la secundaria realice un carro a control remoto con un sensor infrarrojo, sin embargo, nunca sirvió y no entendía porque, hasta que hice esta práctica ya que mi problema es que no estaban bien calibrados los sensores y por lo mismo el control no lo reconoció, también me gusto observar como el transistor captaba la señal del spinner al momento que lo giramos, por lo que me gusto poder analizarlo a detalle este tipo de problemas y soluciones.

Aunque mi parte de armar el hardware es mi favorita, también me resulta muy entretenido la parte del software ya que cuando era mas joven también pensaba que en algún momento podría programar un circuito y es muy satisfactorio poder hacerlo ya que se entiende mas el como funcionan los aparatos que tenemos en casa y/o en nuestra vida cotidiana.

Carrillo Soto Cristian Eduardo

Bibliografía

- [1] ¿Qué es y para qué se usa un contador digital? (s. f.). Netatmo. Recuperado 6 de octubre de 2022, de <https://www.netatmo.com/es-es/guides/energy/heating/consumption/linky-meter>
- [2] colaboradores de Wikipedia. (2022, 15 marzo). Sensor infrarrojo. Wikipedia, la enciclopedia libre. Recuperado 6 de octubre de 2022, de https://es.wikipedia.org/wiki/Sensor_infrarrojo