

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO.**

**INTRODUCCIÓN A LOS MICROCONTROLADORES.**

**REPORTE PRÁCTICA 7.**

**PROFESOR: AGUILAR SANCHEZ FERNANDO**

**Grupo: 3CM6**

**Alumno: Rojas Alvarado Luis Enrique**

**Boleta: 2014010995**

**Link de vídeo explicativo:** [**https://youtu.be/HwHffGVubUE**](https://youtu.be/HwHffGVubUE)

**Contador de 00 a 99**

**Activado con infrarrojo**

**Introducción Teórica**

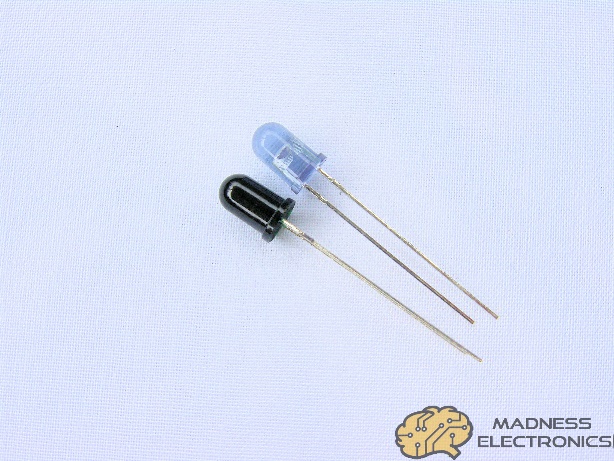
LED de infrarrojos (IRLED):

El diodo IRLED (del inglés Infrared light Emiting Diode), cuentan una radiación electromagnética situada en el espectro electromagnético, en el intervalo que va desde la luz visible a las microondas. Estos diodos se diferencian de los LED por el color de la cápsula que los envuelve que es de color azul o gris. El diámetro de ésta es generalmente de 5 mm. Los rayos infrarrojos se caracterizan por ser portadores de calor radiante. Estos rayos son producidos en mayor o menor intensidad por cualquier objeto a temperatura superior al cero absoluto. Fototransistor: En estos transistores la base está reemplazada por un cristal fotosensible que cuando recibe luz, produce una corriente y desbloquea el transistor. En el fototransistor la corriente circula sólo en un sentido y el bloqueo del transistor depende de la luz; cuanta más luz hay más conduce.

Funciona convirtiendo la corriente eléctrica en luz infrarroja; mientras que los detectores infrarrojos hacen lo opuesto al detectar luz infrarroja y convertirla en una corriente eléctrica. La corriente generada por un detector infrarrojo es una señal que indica que existe ese tipo de luz. El infrarrojo es una longitud de onda de luz que está más allá del rango de la visión humana. Esto hace al infrarrojo una herramienta excelente para aplicaciones donde se requiere la luz, pero donde la luz visible podría ser una distracción o de otra forma no deseada. El uso de diodos infrarrojos emisores de luz, o LEDs, hace posibles a los sistemas de control remoto en varios proyectos. [1]

Diodos emisores de infrarrojos y detectores de infrarrojos. Estos dispositivos simples funcionan a 940 nm y funcionan bien para sistemas IR genéricos incluyendo el control remoto y detección de objetos al menor tacto. Usando un simple ADC en cualquier microcontrolador permitirá lecturas variables que deben recogerse desde el detector. El emisor es accionado hasta 50 mA con una resistencia de limitación de corriente al igual que con cualquier dispositivo LED. La detección es un transistor NPN que está sesgada por la luz IR entrante. [2]

El fotodiodo es un dispositivo que conduce una cantidad de corriente eléctrica proporcional a la cantidad de luz que lo ilumina o incide, este por lo general es usado en conjunto con un led emisor de infrarrojos en aplicaciones como detector de objetos, barreras fotoeléctricas o controles infrarrojos.

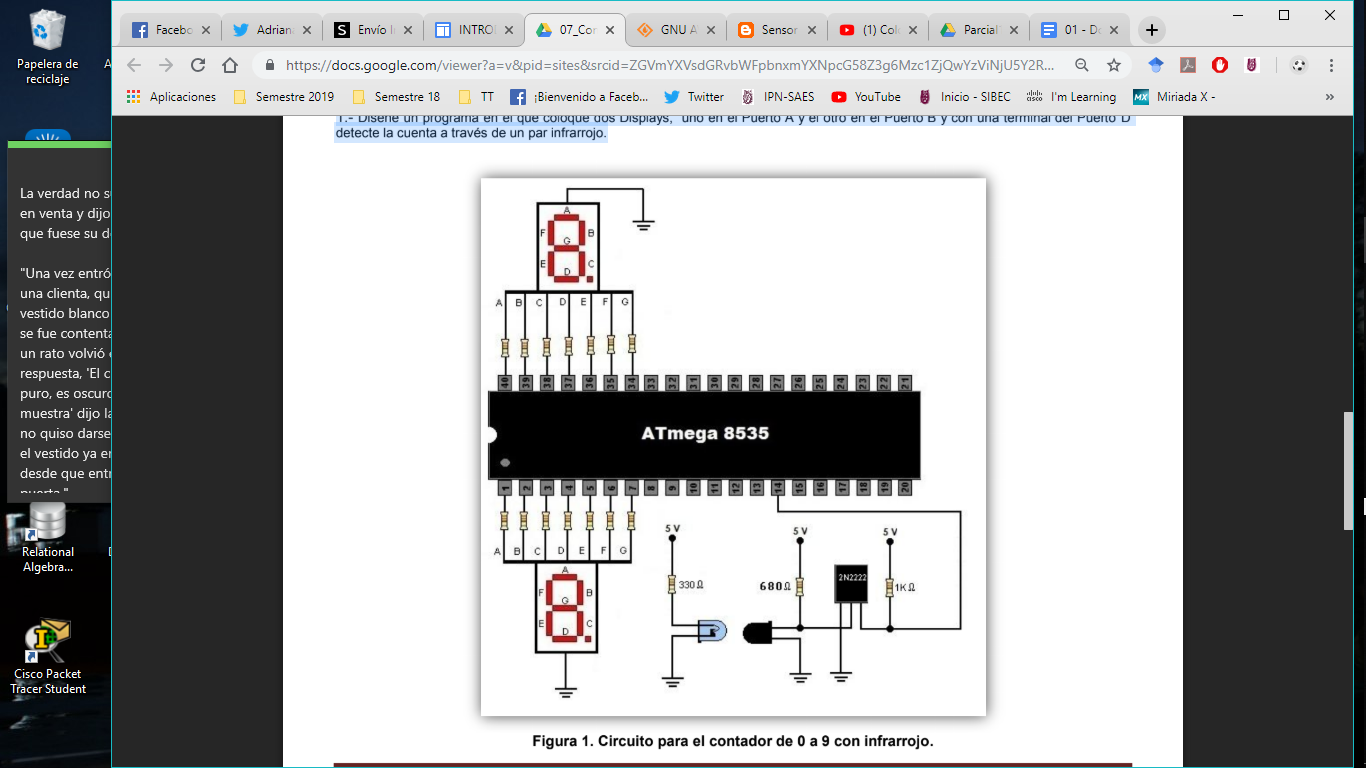
Caractarísticas:

* Emisor:
  + Voltaje de operación: 1,7V
  + Corriente máxima: 100mA
  + Longitud de onda: 940nm
* Receptor:
  + Voltaje maximo inverso: 1,3V
  + Angulo de visión: 20°
  + Longitud de onda: 940nm

Emisor y receptor infrarrojos [3]

**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.



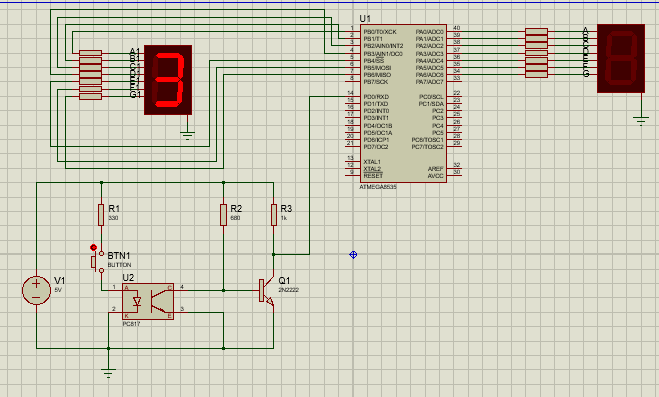
**Código:**

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
8. Project :
9. Version :
10. Date    : 09/11/2020
11. Author  :
12. Company :
13. Comments:

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. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
24. #include <mega8535.h>
25. #include <delay.h>
27. #define boton PIND.0
28. bit botonp;
29. bit botona;
30. unsigned **char** var=0,var2=0;
31. **const** **char** tabla7segmentos [10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

34. **void** main(**void**)
35. {
36. // Declare your local variables here
38. // Input/Output Ports initialization
39. // Port A initialization
40. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
41. DDRA=(1<<DDA7) | (1<<DDA6) | (1<<DDA5) | (1<<DDA4) | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
42. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
43. PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) | (0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);
45. // Port B initialization
46. // Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
47. DDRB=(1<<DDB7) | (1<<DDB6) | (1<<DDB5) | (1<<DDB4) | (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
48. // State: Bit7=0 Bit6=0 Bit5=0 Bit4=0 Bit3=0 Bit2=0 Bit1=0 Bit0=0
49. PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);
51. // Port C initialization
52. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
53. DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) | (0<<DDC2) | (0<<DDC1) | (0<<DDC0);
54. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
55. PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);
57. // Port D initialization
58. // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
59. DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4) | (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);
60. // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=P
61. PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (1<<PORTD0);
63. // Timer/Counter 0 initialization
64. // Clock source: System Clock
65. // Clock value: Timer 0 Stopped
66. // Mode: Normal top=0xFF
67. // OC0 output: Disconnected
68. TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);
69. TCNT0=0x00;
70. OCR0=0x00;
72. // Timer/Counter 1 initialization
73. // Clock source: System Clock
74. // Clock value: Timer1 Stopped
75. // Mode: Normal top=0xFFFF
76. // OC1A output: Disconnected
77. // OC1B output: Disconnected
78. // Noise Canceler: Off
79. // Input Capture on Falling Edge
80. // Timer1 Overflow Interrupt: Off
81. // Input Capture Interrupt: Off
82. // Compare A Match Interrupt: Off
83. // Compare B Match Interrupt: Off
84. TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
85. TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
86. TCNT1H=0x00;
87. TCNT1L=0x00;
88. ICR1H=0x00;
89. ICR1L=0x00;
90. OCR1AH=0x00;
91. OCR1AL=0x00;
92. OCR1BH=0x00;
93. OCR1BL=0x00;
95. // Timer/Counter 2 initialization
96. // Clock source: System Clock
97. // Clock value: Timer2 Stopped
98. // Mode: Normal top=0xFF
99. // OC2 output: Disconnected
100. ASSR=0<<AS2;
101. TCCR2=(0<<WGM20) | (0<<COM21) | (0<<COM20) | (0<<WGM21) | (0<<CS22) | (0<<CS21) | (0<<CS20);
102. TCNT2=0x00;
103. OCR2=0x00;
105. // Timer(s)/Counter(s) Interrupt(s) initialization
106. TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
108. // External Interrupt(s) initialization
109. // INT0: Off
110. // INT1: Off
111. // INT2: Off
112. MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
113. MCUCSR=(0<<ISC2);
115. // USART initialization
116. // USART disabled
117. UCSRB=(0<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (0<<RXEN) | (0<<TXEN) | (0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
119. // Analog Comparator initialization
120. // Analog Comparator: Off
121. ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
122. SFIOR=(0<<ACME);
124. // ADC initialization
125. // ADC disabled
126. ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
128. // SPI initialization
129. // SPI disabled
130. SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);
132. // TWI initialization
133. // TWI disabled
134. TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);
136. **while** (1)
137. {
138. **if** (boton==0)
139. botona=0;
140. **else**
141. botona=1;
143. **if** ((botonp==1)&&(botona==0)){ //hubo cambio de flanco de 1 a 0
144. var2++; //Se incrementa la variable
145. **if** (var2==10){
146. var2=0;
147. var++;
148. }
149. **if**(var==10)
150. var=0;
152. delay\_ms(40); //Se coloca retardo de 40mS para eliminar rebotes
153. }
154. **if** ((botonp==0)&&(botona==1)) //hubo cambio de flanco de 0 a 1
155. delay\_ms(40); //Se coloca retardo de 40mS para eliminar rebotes
157. PORTA=tabla7segmentos[var2];
158. PORTB=tabla7segmentos[var];
159. botonp=botona;
161. }
162. }

**Circuito en Proteus:**

****

**Conclusiones:**

En esta práctica pudimos observar el comportamiento del sensor infrarrojo simulándolo en Proteus, cuanto más cerca esté el infrarojo más estable será la comunicación con el receptor, por lo que se verá reflejado en las salidas, reutilizando la plantilla del código de la práctica anterior solo agregamos un display para hacer la simulación de las decenas con la misma lógica, solo agregando una variable y hacer que gire entorno a esa variable, logando un contador doble, que cuando llegan a 10 se reinician.

# **Referencias**

|  |  |
| --- | --- |
| [1] | G. Electronics, «LED infrarojo,» Geekbot Electronics, [En línea]. Available: http://www.geekbotelectronics.com/producto/led-infrarrojo-ir/. [Último acceso: 9 Noviembre 2020]. |
| [2] | SANDOROBOTICS, «Diodos Infrarrojos – Emisor y Receptor (Par),» Vleeko, 2019. [En línea]. Available: https://sandorobotics.com/producto/sen-00241/. [Último acceso: 9 Noviembre 2020]. |
| [3] | M. Electronics, «Emisor y Receptor Infrarrojo,» Madness Electronics, 2012. [En línea]. Available: http://www.madnesselectronics.com/producto/emisor-y-receptor-infrarrojo/. [Último acceso: 9 Noviembre 2020]. |