ITESHU

**sensor de proximidad**

**PROFEOSR**

* Carlos Arturo Espinoza Galicia

**MATERIA:**

* Sistemas Programables

**INTEGRANTES**

* Rubicela Badillo Martínez
* Alma Brenda Callejas Pérez
* Carolina Hernández Martínez
* Elizabeth Quintanar Martínez
* Misael Zenil García

**DESCRIPCIÓN DE LA PRÁCTICA:**

En esta práctica se realizó un circuito en arduino con el fin de conectar un sensor de proximidad y conocer sus funciones.

**OBJETIVO:**

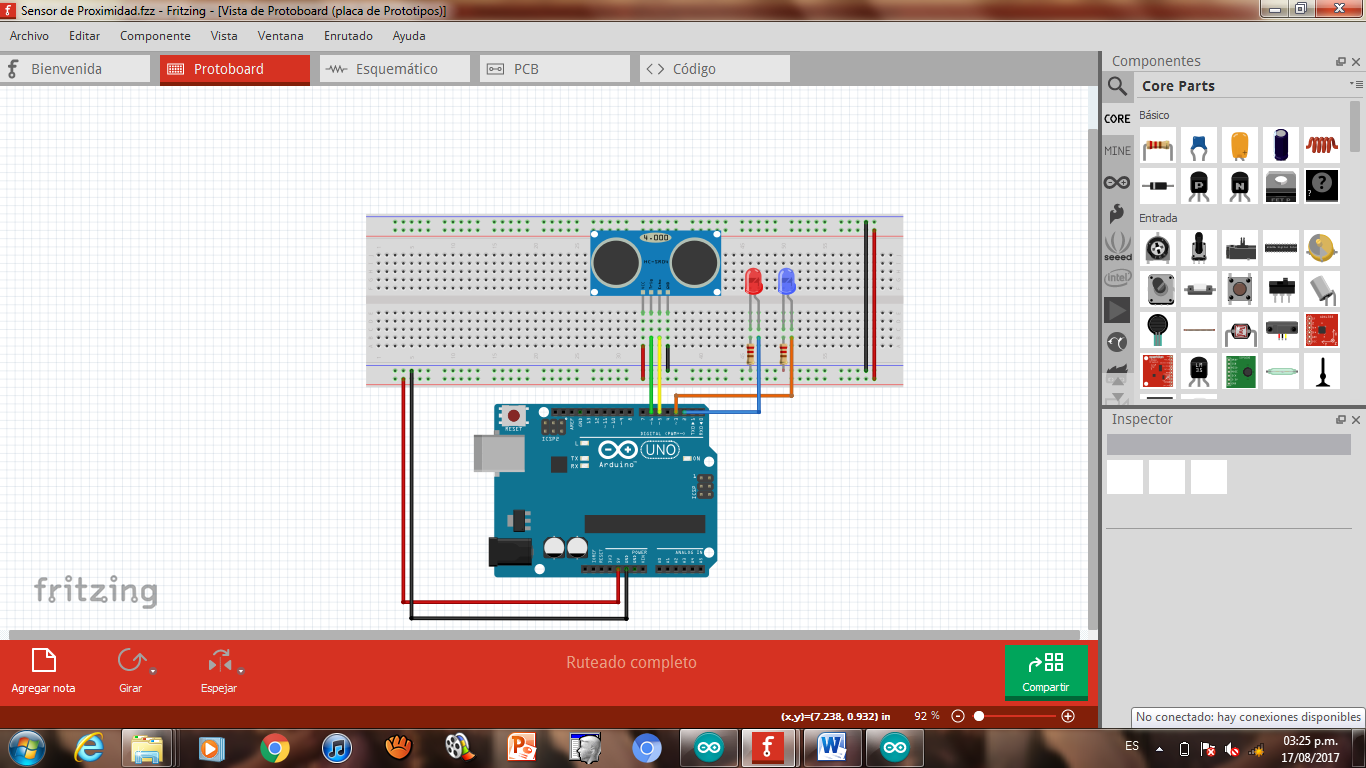
Conocer el funcionamiento del sensor de proximidad y usarlo para detectar que tan lejos o cerca se encuentra un objeto.

**MATERIALES:**

* Una protoboard
* Cable para conexión
* Resistencias
* Leds
* Tarjeta de Arduino Uno
* Cable USB para arduino
* Software de Arduino instalado
* Fritzing
* Sensor de Proximidad

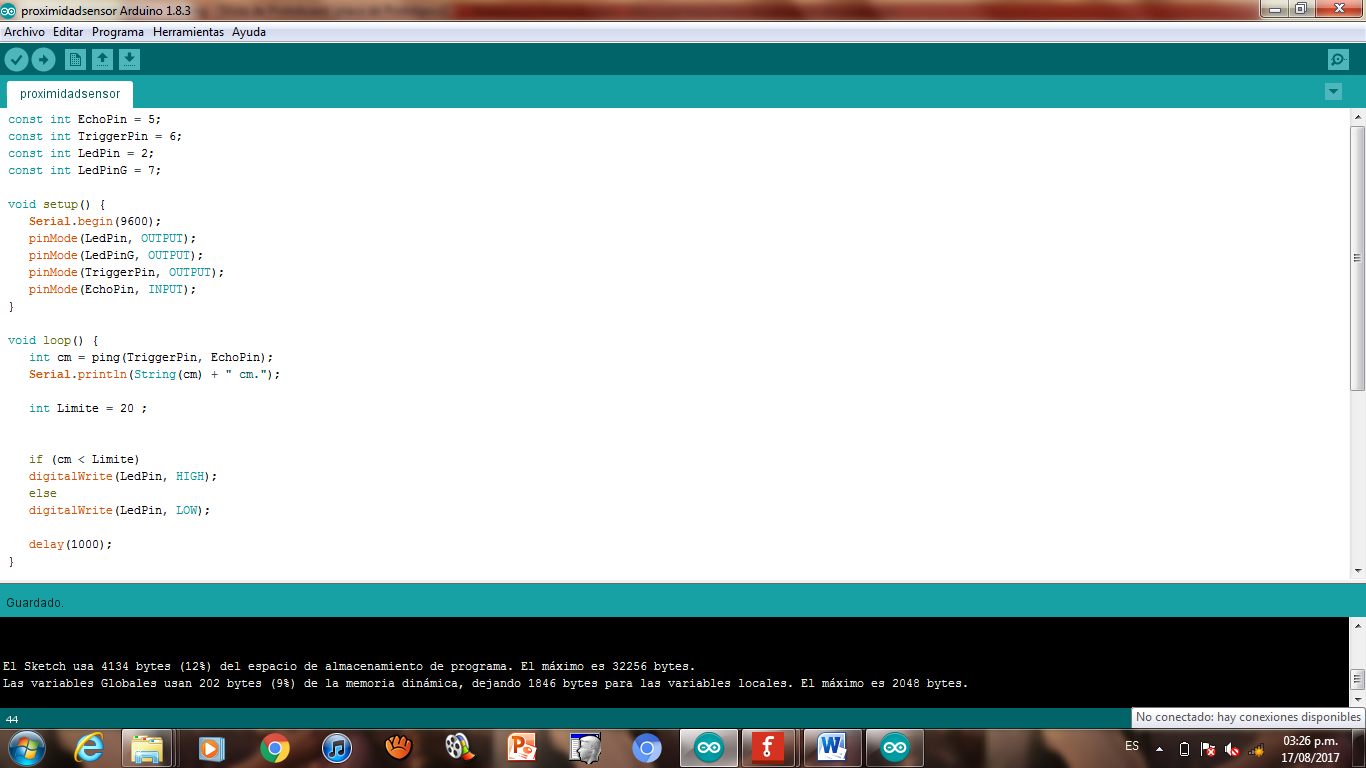
**PROCEDIMIENTO:**

1. Conectamos el sensor de proximidad a nuestra protoboard
2. El sensor trae marcado el nombre de los 4 pines con los que cuenta, realizamos la conexión:
3. El pin GND lo conectamos a la tierra
4. El pin Vcc lo conectamos al pin de 5v del arduino
5. El pin trig lo conectamos al pin 6 digital del arduino
6. El pin Echo lo conectamos al pin 5 digital del arduino
7. Conectamos dos led a la protoboard, el extremo plano lo conectamos a tierra mediante una resistencia de 220 ohms.
8. Conectamos los cables de tierra y corriente del arduino a la protoboard.

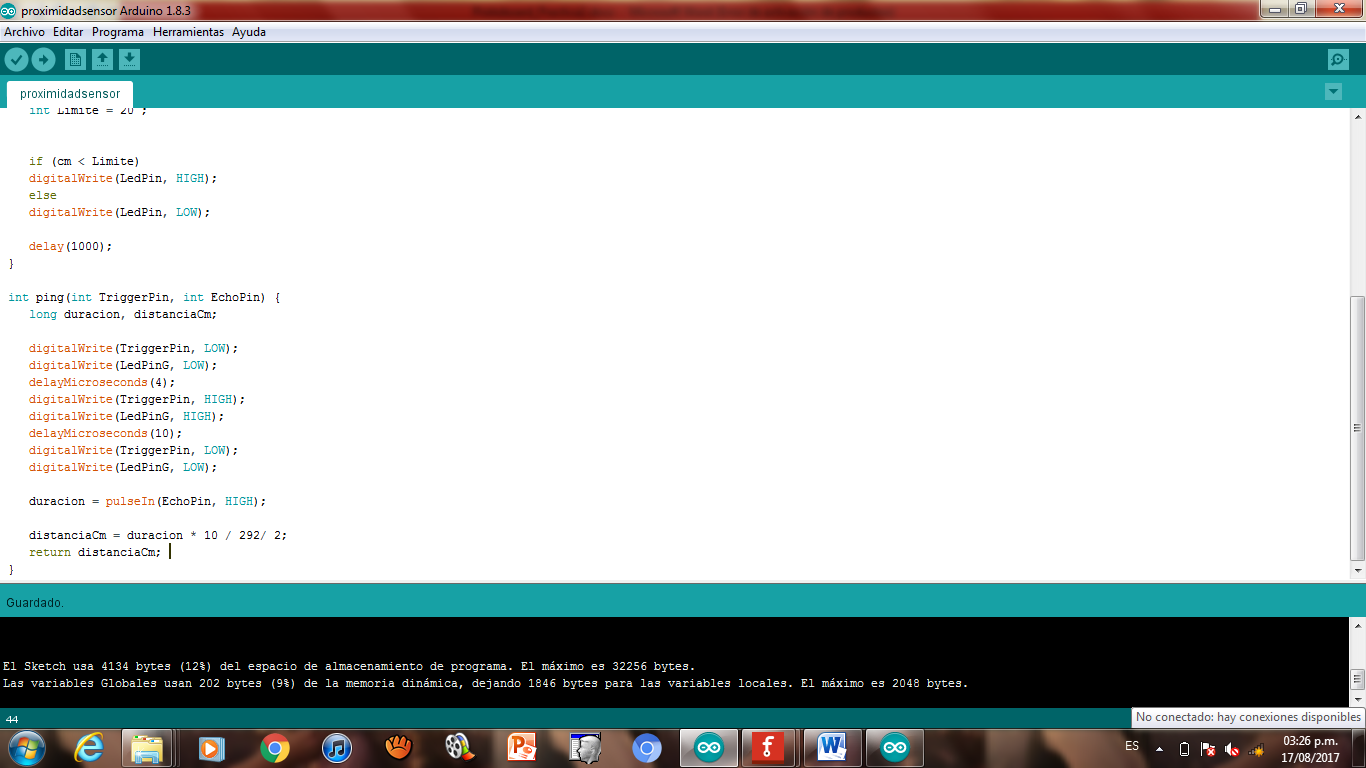


*Diagrama de conexión realizado en Fritzing*

1. Compilamos el código y lo cargamos para que la conexión realizada funcione



*Código en arduino*



*Codigo en arduino*

**Codigo**

const int EchoPin = 5;

const int TriggerPin = 6;

const int LedPin = 2;

const int LedPinG = 7;

void setup() {

Serial.begin(9600);

pinMode(LedPin, OUTPUT); //El Pin del Led de Comparación se establece en salida de datos

pinMode(LedPinG, OUTPUT); //El Pin del Led de Pulso se establece en salida de datos

pinMode(TriggerPin, OUTPUT); //El Pin del Trigger del Sensor se establece en salida de datos

pinMode(EchoPin, INPUT); //El Pin del Echo del Sensor se establece en entrada de datos

}

void loop() {

int cm = ping(TriggerPin, EchoPin); //Se manda llamar a la función PING con los valores de TriggerPin y EchoPin y se le asigna a la variable entera CM

Serial.println(String(cm) + " cm."); //Se imprimen los valores que regresan por la función PING

int Limite = 20 ; //Limita la medición del sensor

//Si el valor que regresa por la función PING dentro de la variable CM es menor al límite establecido, el Led de Comparación se encenderá o de lo contrario se apagara.

if (cm < Limite)

digitalWrite(LedPin, HIGH);

else

digitalWrite(LedPin, LOW);

delay(1000); //Se espera 1 segundo para volver a repetir la operación

}

int ping(int TriggerPin, int EchoPin) {

long duracion, distanciaCm;

digitalWrite(TriggerPin, LOW); //Para generar un pulso limpio

digitalWrite(LedPinG, LOW);

delayMicroseconds(4);

digitalWrite(TriggerPin, HIGH); //Generamos Trigger

digitalWrite(LedPinG, HIGH); //El Led de Pulso se enciende con el disparador por 10 us

delayMicroseconds(10);

digitalWrite(TriggerPin, LOW); //El disparador deja de funcionar para esperar el rebote de la señal emitida

digitalWrite(LedPinG, LOW); //El Led de Pulso se apaga con el disparador

duracion = pulseIn(EchoPin, HIGH); //Medimos el tiempo entre pulsos, en microsegundos

distanciaCm = duracion \* 10 / 292/ 2; //Se convierte la distancia, en cm

return distanciaCm; //Retornamos el valor obtenido en DISTANCIACM para que pueda ser utilizado dentro de la función LOOP, y el proceso se vuelve a repetir

}