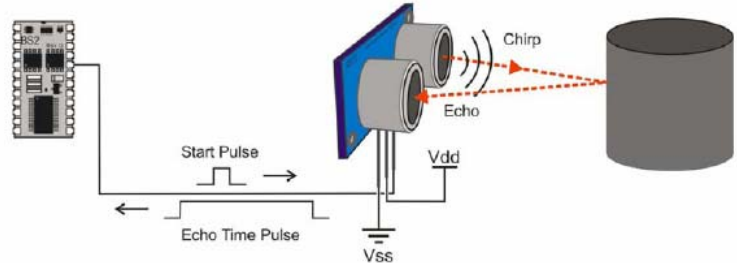


# Arduino Módulo ultrasonidos

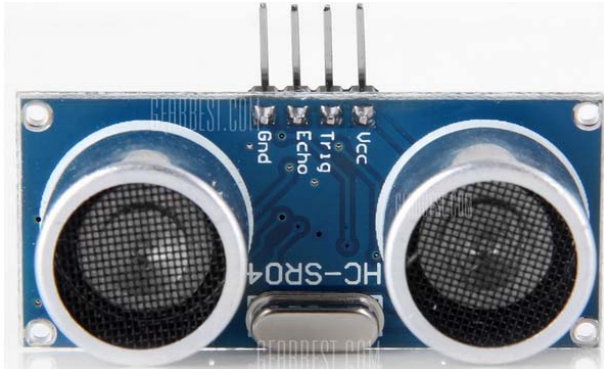
Con este sensor (Parallax , US-100 o HC-SR04) podemos detectar objetos y calcular la distancia.



El sensor de parallax tiene 3 pines, dos para la alimentación y uno para mandar y recibir el pulso de la señal. Puede detectar objetos entre 3cm y 3,3m.



El sensor US-100 o HC-SR04 tiene 4 pines, dos para la alimentación, uno para enviar el pulso (Trig) y otro (Echo) para recibirlo.

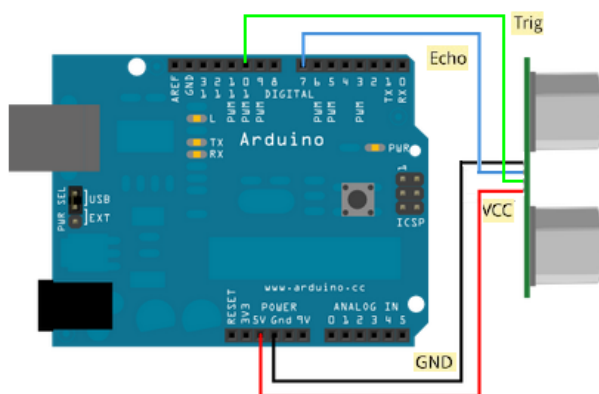


## Características:

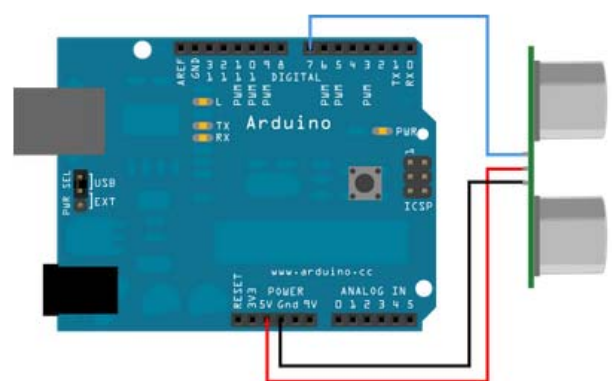
- Static Current: menor de 2mA
- Output Signal: Electric frequency signal, high level 5V, low level 0V
- Sensor Angle: menor de 15 grados
- Detection Distance: 2cm-450cm
- High Precision: mayor de 0.3cm
- Input Trigger Signal: 10us TTL impulse
- Echo Signal: Salida señal TTL PWL.

## Conexiones:

### US-100 o HC-SR04



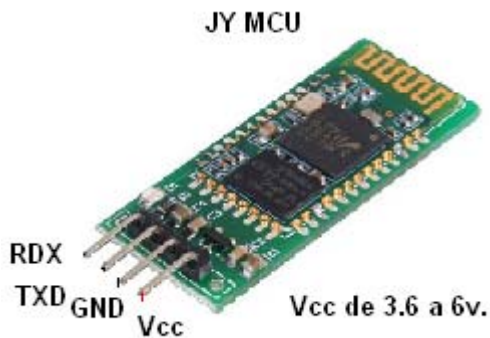
### Parallax



## Módulo Bluetooth:

El módulo JY-MCU podemos utilizarlo para comunicarse con arduino mediante bluetooth.

Conectar a la placa arduino RDX a D1(TX), TXD a D0(RX), GND y Vcc a los pines correspondientes. La información se recibe vía serie.

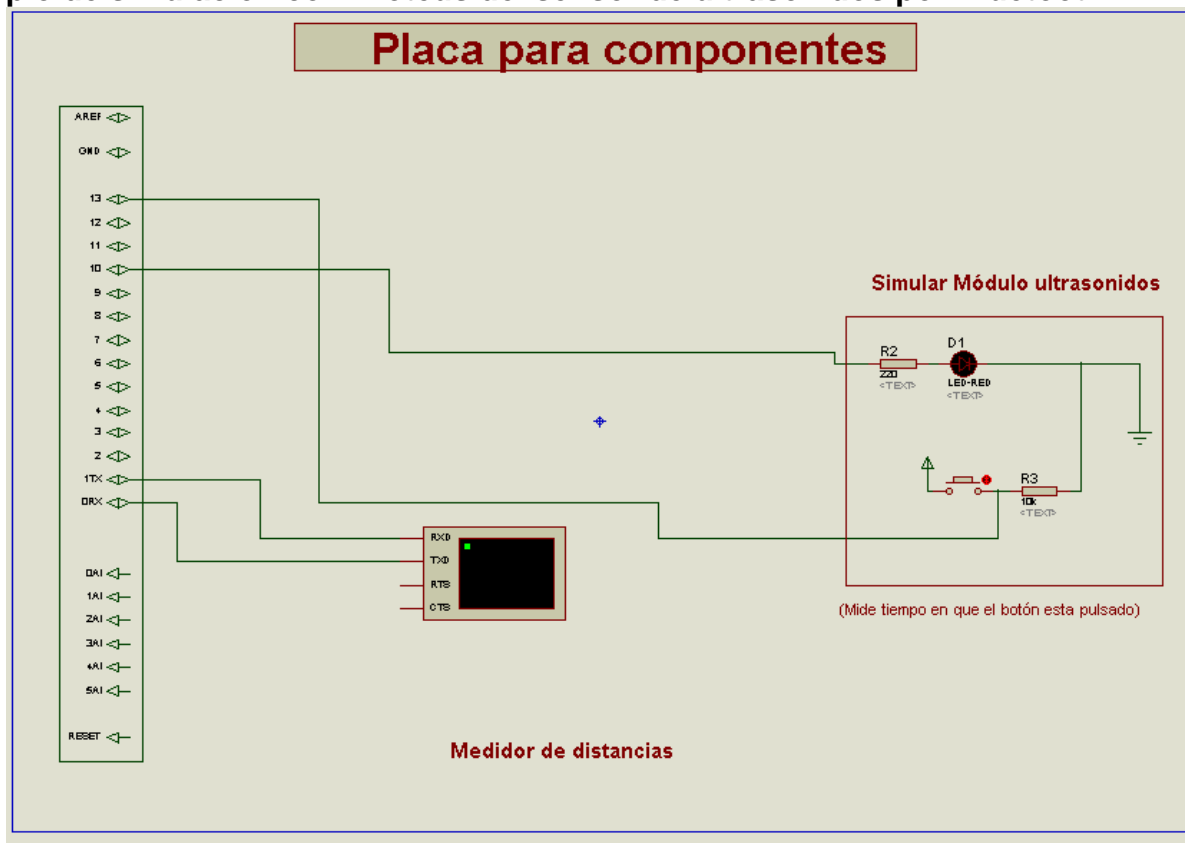


Para utilizar el teléfono móvil como elemento de control, vincular el módulo como **livor** (contraseña **1234**).

Hay aplicaciones en android para descargar como Motbot (<http://www.mobot.es/>) o realizarla nosotros con ayuda de APP Inventor (<http://ai2.appinventor.mit.edu/>).

Para simular el módulo en **proteus** utilizamos el **VIRTUAL TERMINAL** (configurar velocidad del simulador a 1200). Simulamos la información recibida por bluetooth con ayuda del teclado.

### Ejemplo de simulación con Proteus del sensor de ultrasonidos por Bluetooth:



## Programas arduino:

```
/* Ping))) Sensor (Parallax)
   Calcula la distancia del objeto
   http://www.arduino.cc/en/Tutorial/Ping
*/
const int pingPin = 7;                                //pin del sensor

void setup() {
  Serial.begin(9600);
} //setup

void loop(){
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))) a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
} //loop

long microsecondsToInches(long microseconds){
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second). This gives the distance travelled by the ping, outbound
```

```
// and return, so we divide by 2 to get the distance of the obstacle.
// Ver: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
// The speed of sound is 340 m/s or 29 microseconds per centimeter.
// The ping travels out and back, so to find the distance of the
// object we take half of the distance travelled.
return microseconds / 29 / 2;
}
```

```
//Ping))) Sensor (US-100 o HC-SR04)

//http://elcajondeardu.blogspot.com.es/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html

long distancia;
long tiempo;

// Da problemas si se cambia los pines
const int trigger=9; //activación del pin 9 como salida: para el pulso ultrasónico. Da problemas si se cambia
const int echo=8; //activación del pin 8 como entrada: tiempo del rebote del ultrasonido.

void setup(){
  Serial.begin(9600);
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
}

void loop(){
  digitalWrite(trigger,LOW); // Por cuestión de estabilización del sensor
  delayMicroseconds(5);
  digitalWrite(trigger, HIGH); // envío del pulso ultrasónico
  delayMicroseconds(10);
  tiempo=pulseIn(echo, HIGH);
  /* Función para medir la longitud del pulso entrante. Mide el tiempo que transcurrido entre el envío del pulso
  ultrasónico y cuando el sensor recibe el rebote, es decir: desde que el pin 12 empieza a recibir el rebote, HIGH,
  hasta que deja de hacerlo, LOW, la longitud del pulso entrante*/
  distancia= int(0.017*tiempo); //fórmula para calcular la distancia obteniendo un valor entero
  /*Monitorización en centímetros por el monitor serial*/
  Serial.println("Distancia ");
  Serial.println(distancia);
  Serial.println(" cm");
  delay(1000);
}
```