Curso de Sensores en Plataforma Arduino

Departamento de Electrónica

Sensores de Ultrasonidos

© 2013 Departamento de Electrónica. Universidad de Alcalá

D. Julio Pastor Mendoza (pastor@depeca.uah.es)
D. Pedro Revenga de Toro (revenga@depeca.uah.es)
Profesores del Departamento de Electrónica (UAH)

Universidad de Alcalá

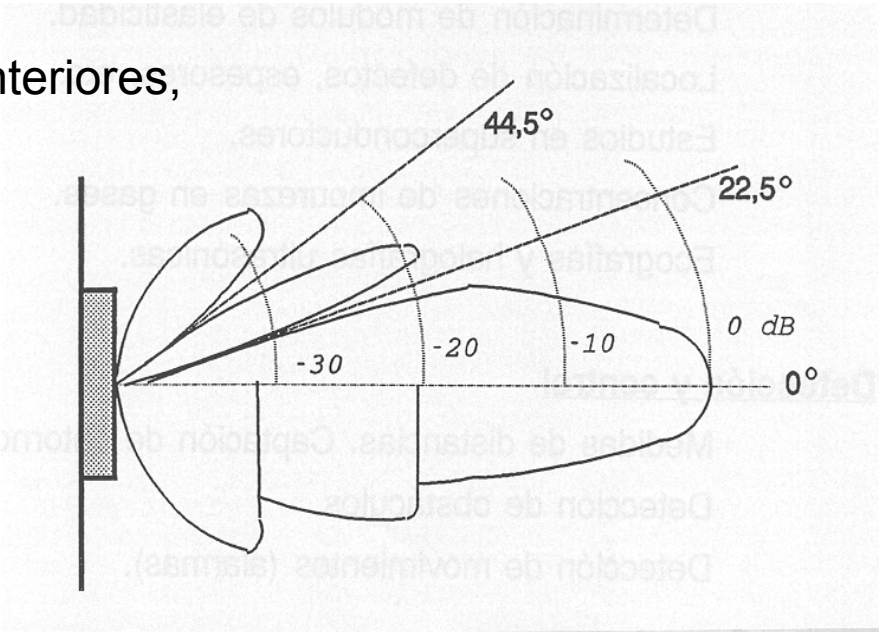
Medida de Distancia: Ultrasonidos (1)

Características de los Ultrasonidos

- ✓**Ondas sonoras** de frecuencia superior a la audible por el humano (20KHz)
- ✓Se generan por fenómenos mecánicos o piezoeléctricos de vibración
- ✓**Resultado:** vibraciones periódicas en el tiempo y en el espacio
- ✓Velocidad de transmisión del **sonido: 0.3m/ms** (en el aire)
  - ✓Velocidad de la luz en el vacío: 0.3m/ns (en el vacío)

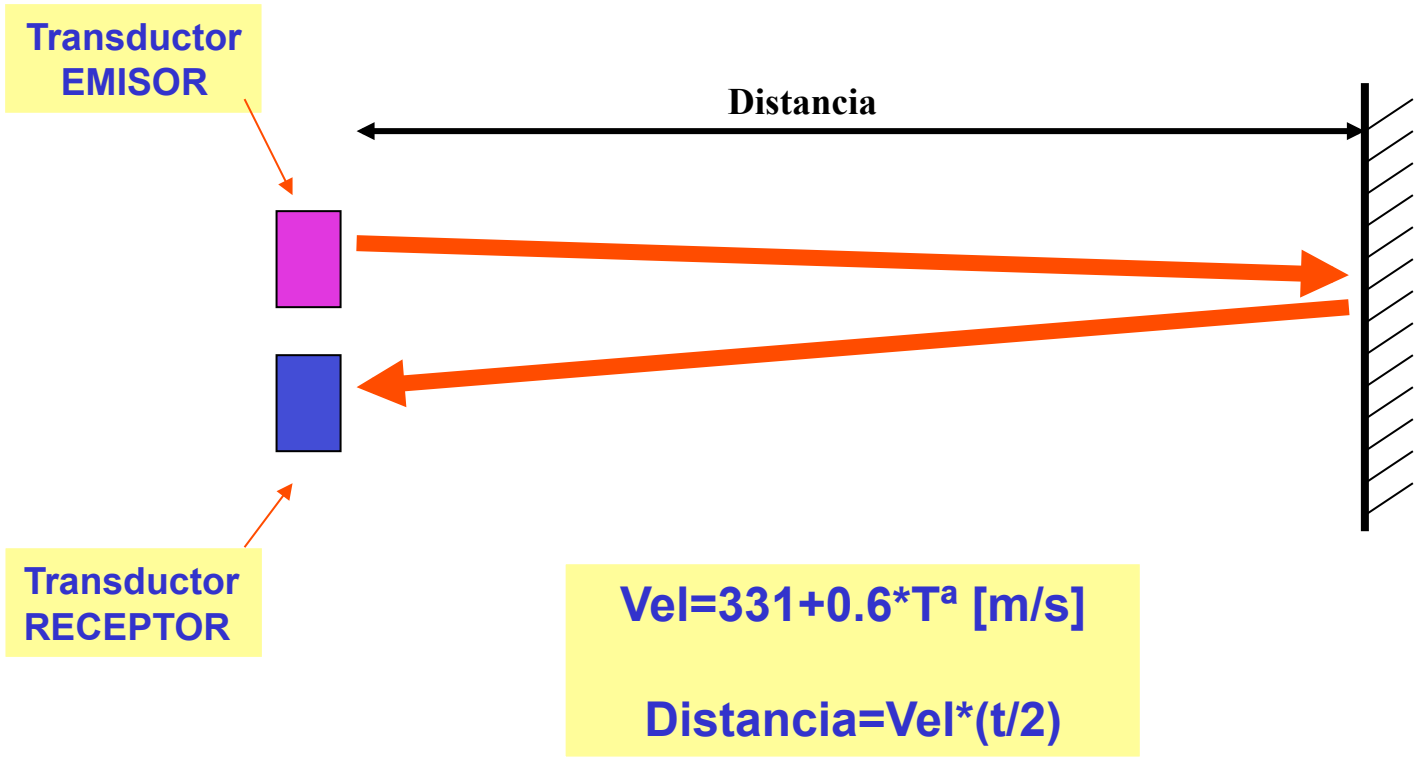
Sensor: Superficie plana oscilando a frecuencia constante

Muy utilizados en robótica, en interiores, por requerir interfaz electrónico sencillo y de bajo coste.



Medida de Distancia: Ultrasonidos

Medidores de distancia por ultrasonidos



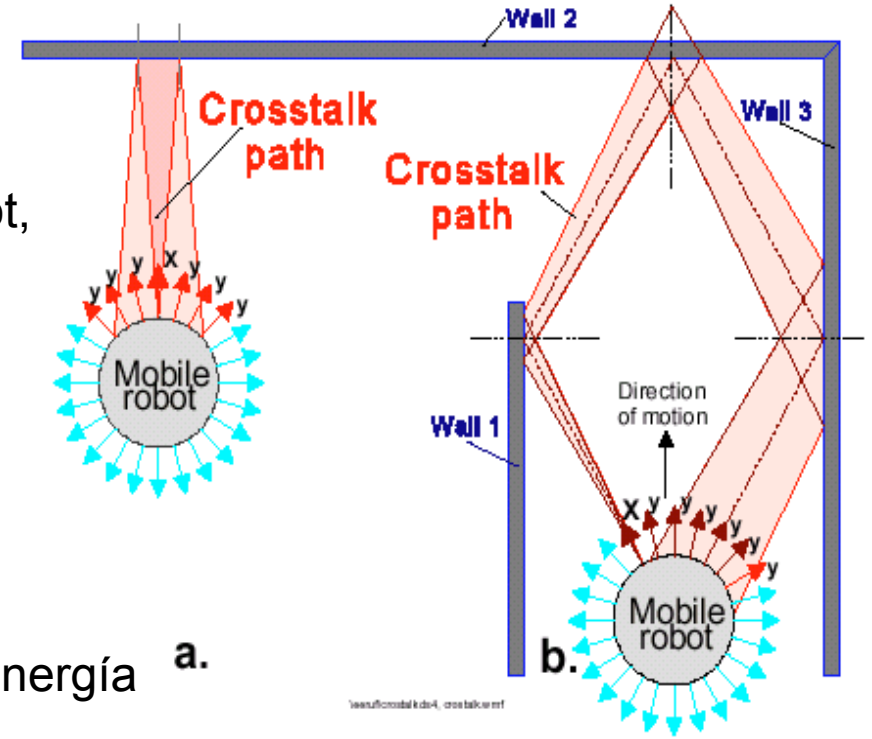
Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

¿Sensado con Ultrasonidos?:

- ✓Basado en el reflejo de la onda en los elementos del entorno del robot,
- ✓pero....

Reflexiones, Atenuaciones, Doppler, Interferencias!

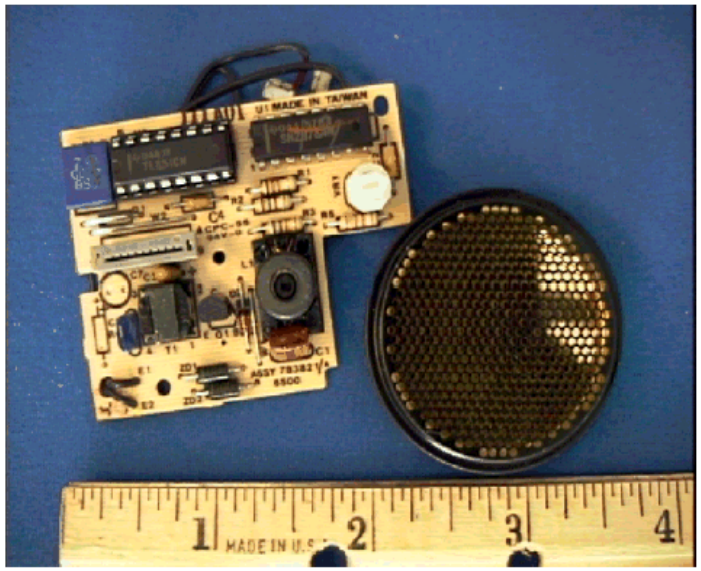


- ✓Se basa en transformación de la energía **transductor → sensor**
  - Emisor: **energía eléctrica => sonora (ultrasonido)**
  - Receptor: **energía sonora (ultrasonido) => eléctrica**

Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

Tipos de transductores :



- ✓**Piezoeléctricos**: Se deforman ante campo eléctrico
  - **Naturales:** cuarzo
  - **Artificiales:** cerámicas

- ✓**Magnetoestrictivos**: Se deforman ante campo magnético

- ✓**Mecánicos y Electromecánicos**: Se deforman al aplicar fuerza

- ✓**Electroestáticos**: Se deforman por efecto electroestático. condensador de placas paralelas

Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

Tipos de transductores:

TRANSDUCTOR		RANGO DE FRECUENCIA	POTENCIAS
PIEZOELÉCTRICOS		KHz a MHz	PEQUEÑAS MEDIAS
MAGNETOESTRICTIVOS		< 100 KHz	MEDIAS
MECÁNICOS		< 50 KHz	ALTAS
ELECTROMECÁNICOS		< 50 KHz	ALTAS
ELECTROSTÁTICOS		< 200 KHz	BAJAS

Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

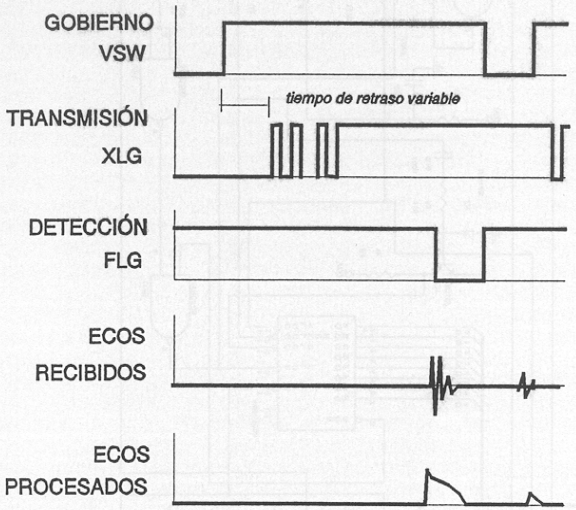
Aplicaciones de los ultrasonidos según potencia de emisión:

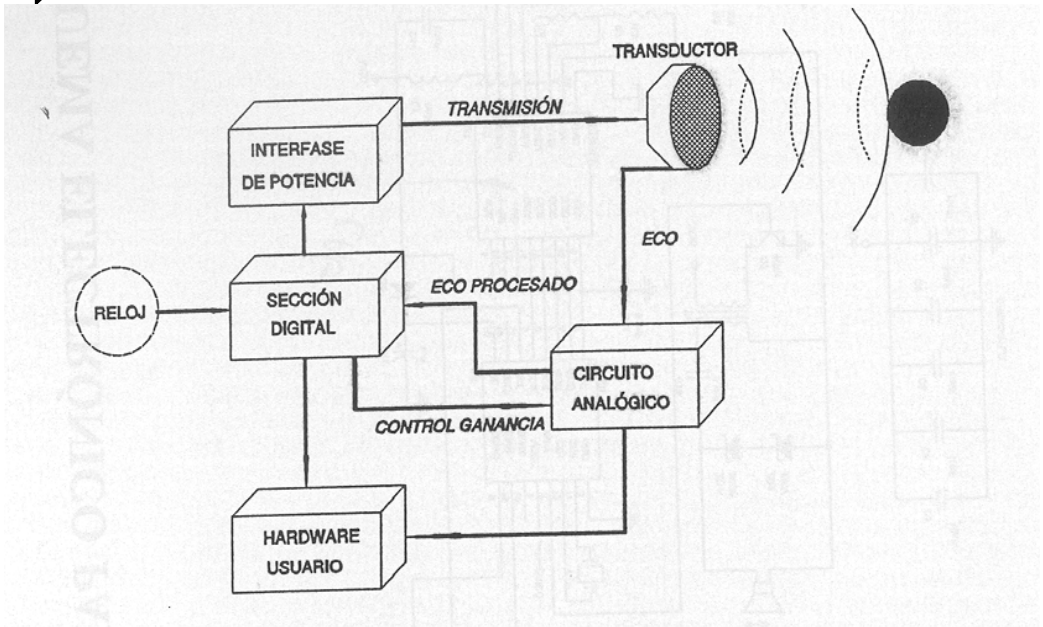
- Alta potencia (mW a KW): **Sonar, limpieza catalización, soldadura, erosión, cirugía**
- Baja potencia(uW a mW): **Ecografía, localización de defectos en objetos, robótica**

medida de distancias (0.5– 10m ±10%)

captación y mapeado de entornos

detección de obstáculos





Medida de Distancia: Ultrasonidos

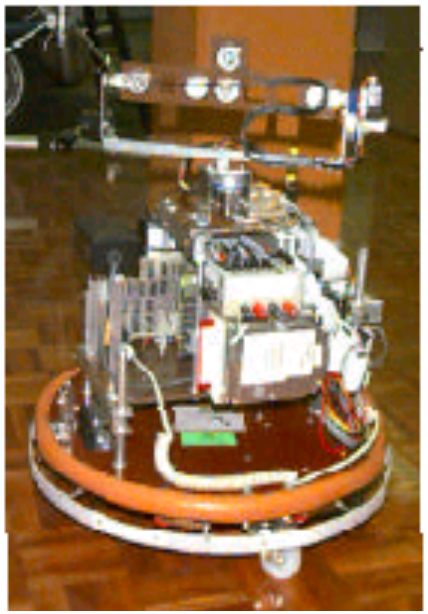
PLATAFORMA ARDUINO

Consideraciones sobre el uso de ultrasonidos según aplicación:

- **Alcance:** Sistema de enfoque acústico (4 a 6m), receptor y emisor enfrentados
- **Resolución**: Altas frecuencias de TX, alta frecuencia para el contador de tiempos de vuelo
- **Exactitud**: Control de Tª, humedad, limpieza del aire
  330 m/s a 0º y 340 m/s a 25

An ambient temperature shift of just 30 ºC can cause a 0.3 meter (1 ft) error at a measured distance of 10 meters. [Everett, 1985].

- **Anchura del haz**: Con lentes acústicas
  y con arrays de sensores



Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

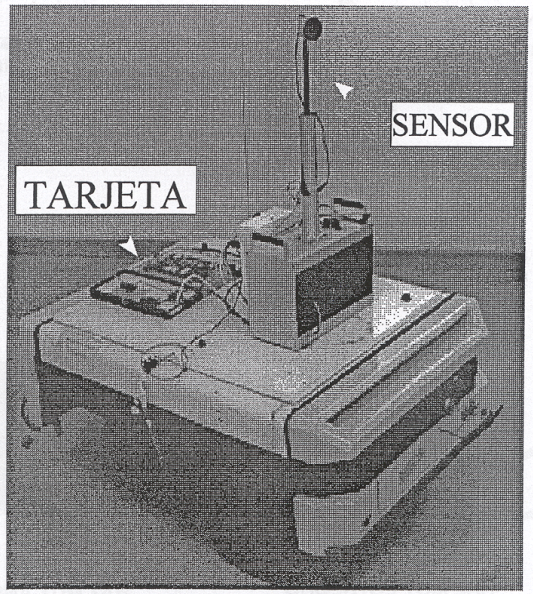
Medidas de Distancias con Ultrasonidos:

- **Tiempo de vuelo** permite obtener distancia al objeto más próximo alrededor del robot
- A partir de esa medida: **mapas del entorno, caracterización de obstáculos**

Disposición del sensor:

- **Anillo de sensores:** Sensado en paralelo
- **Sonar (sensor giratorio):** No existen zonas muertas

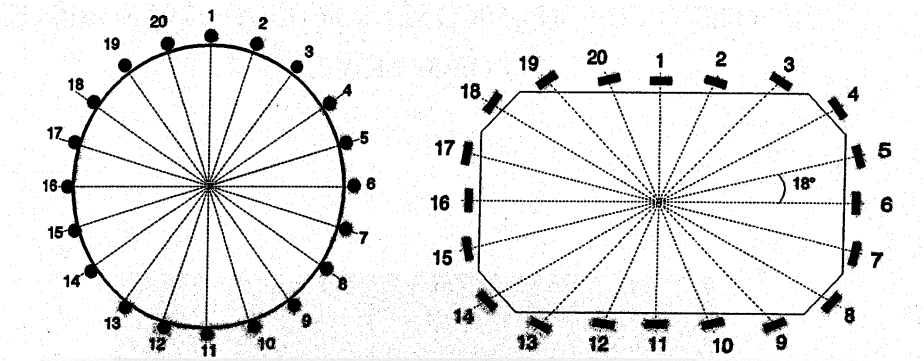
Sensor giratorio →

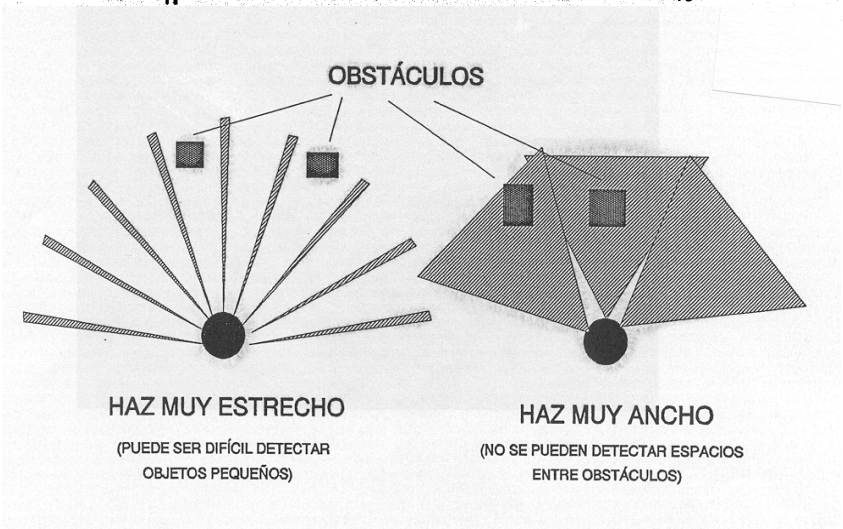


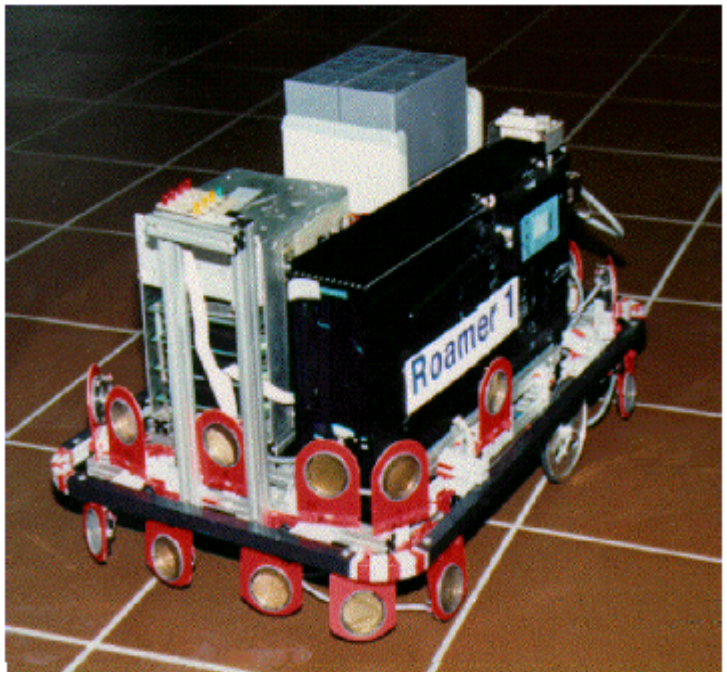
Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

Anillo de sensores →



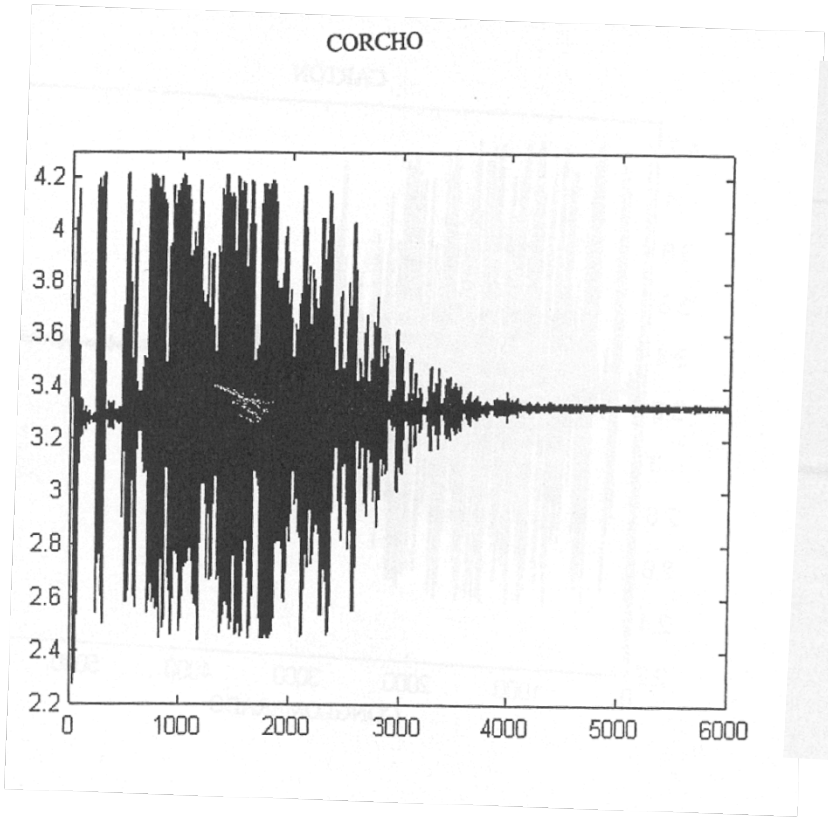


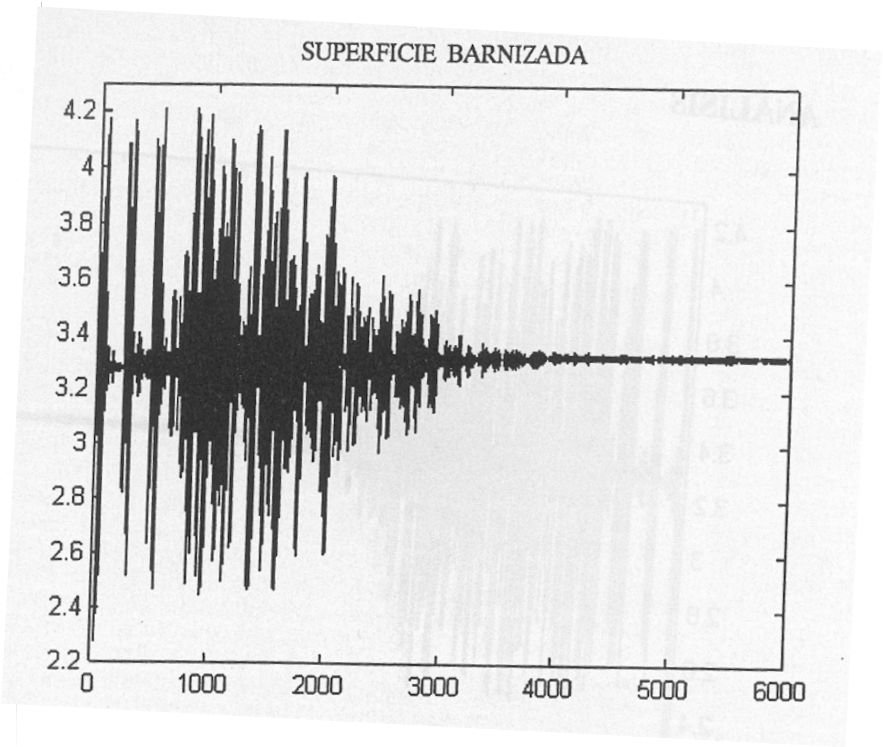


Medida de Distancia: Ultrasonidos

PLATAFORMA ARDUINO

¿cuándo interesa la forma de onda? Reconocimiento de materiales

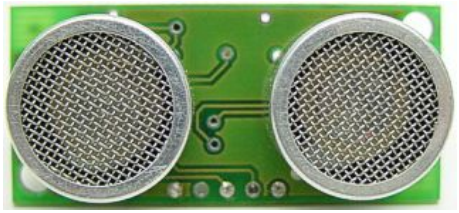




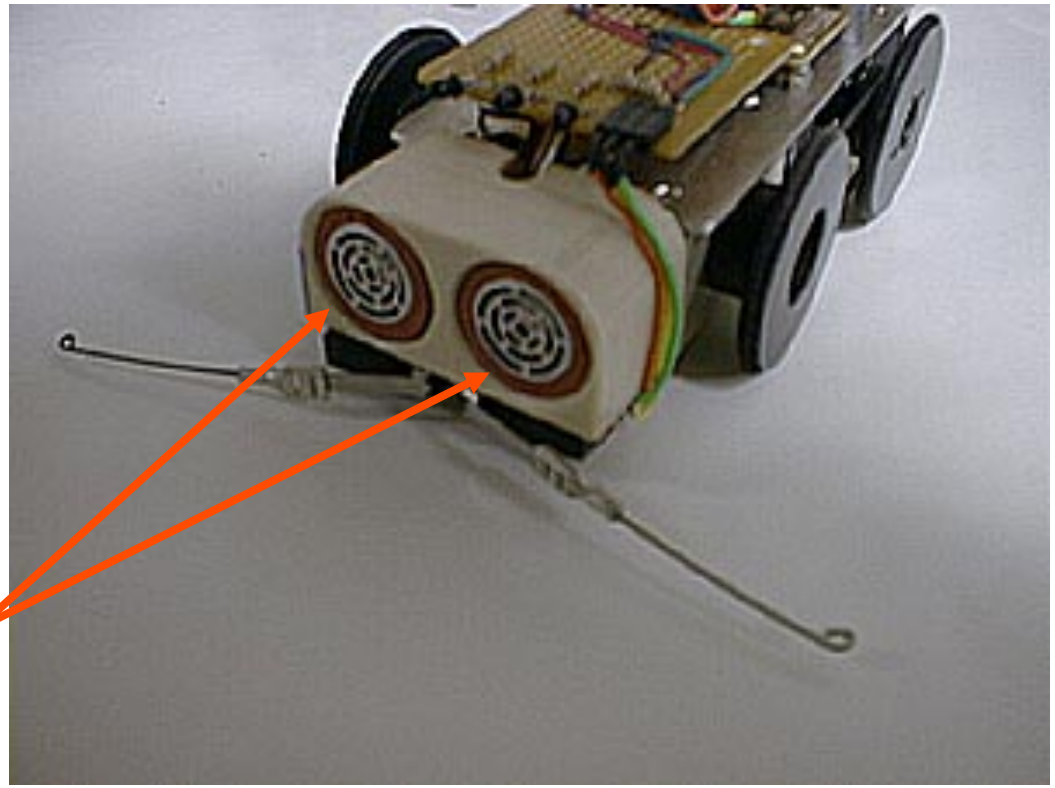
Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

SENSORES: Medidores de distancia por ultrasonidos



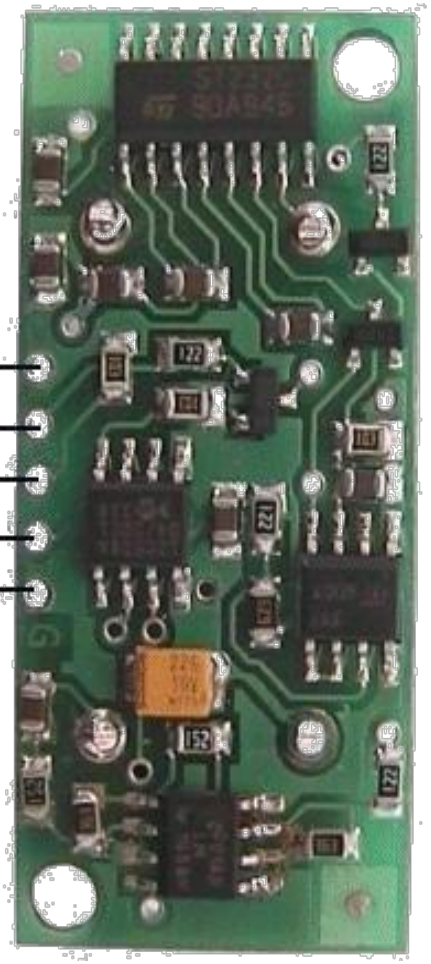
SRF04

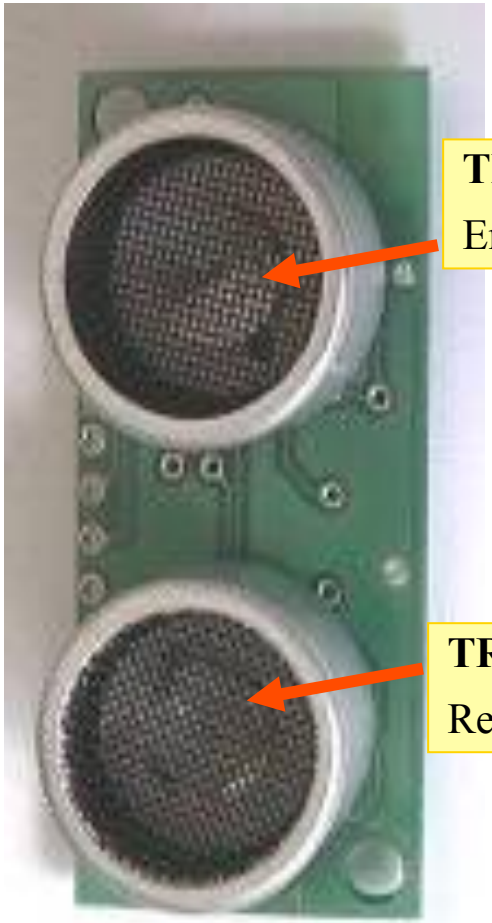


Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

Conexiones del SRF04





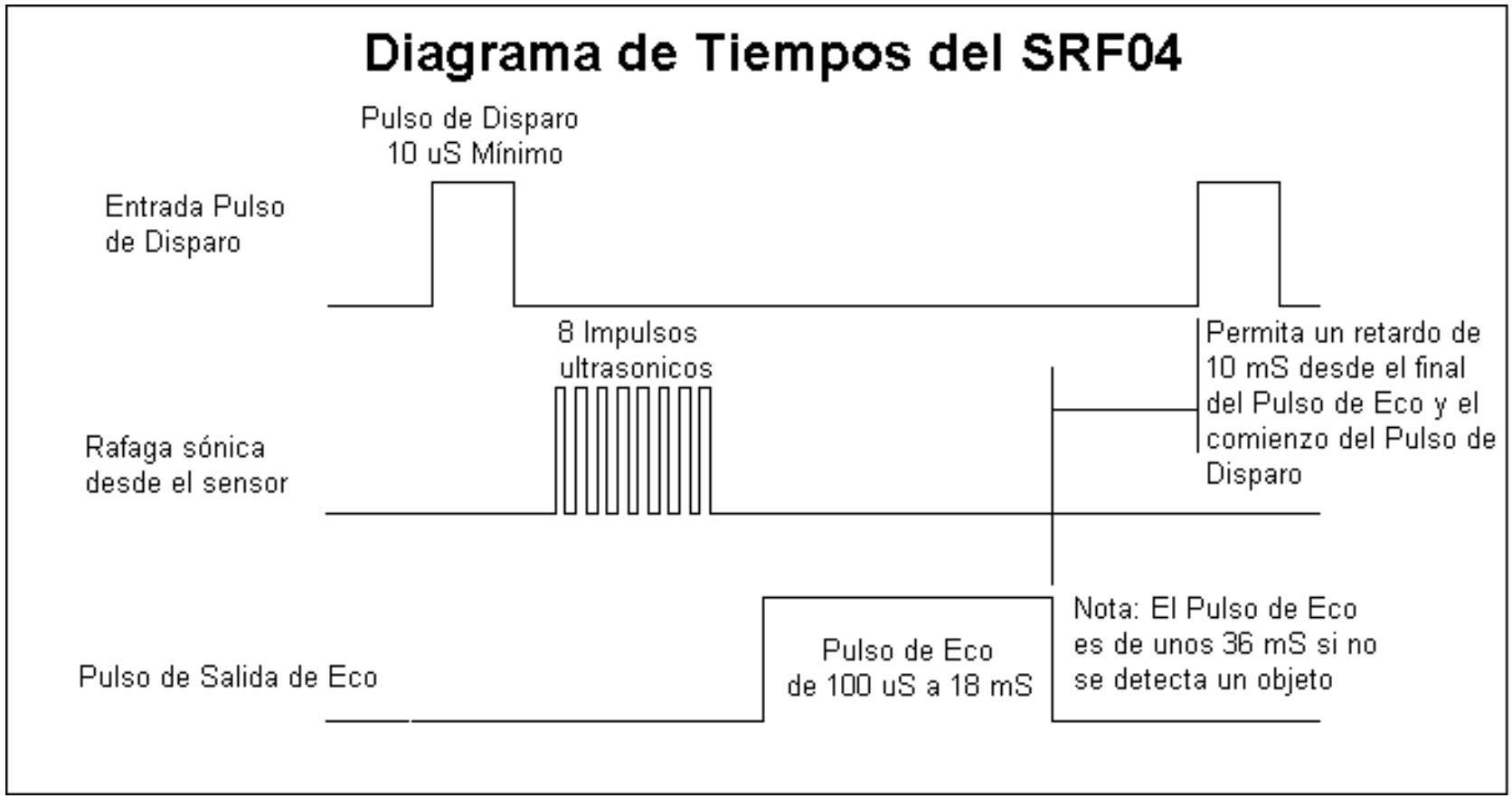
TRANSDUCTOR
Emisor

TRANSDUCTOR
Receptor

Mide de 30cm a 3m

Medida de distancia: Ultrasonidos

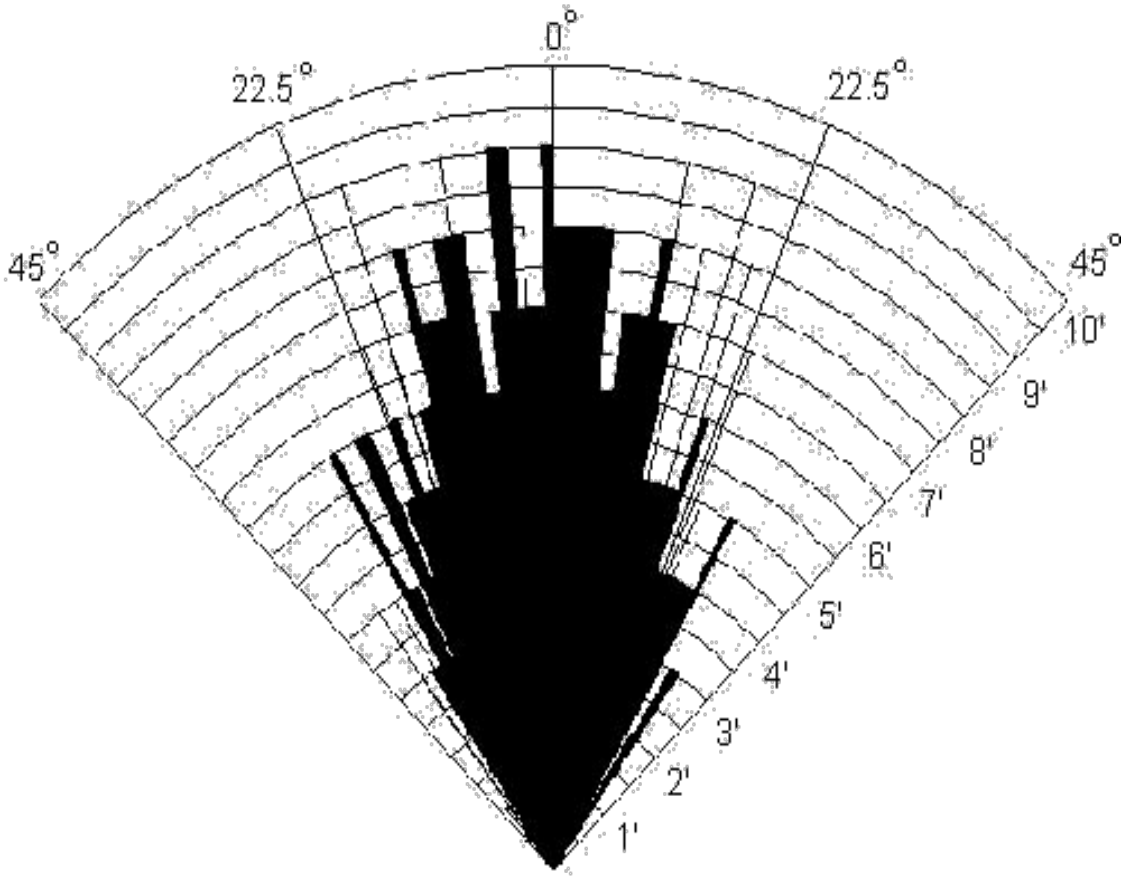
PLATAFORMA ARDUINO



Medida de distancia: Ultrasonidos

Medidor de distancia por ultrasonidos : SRF04

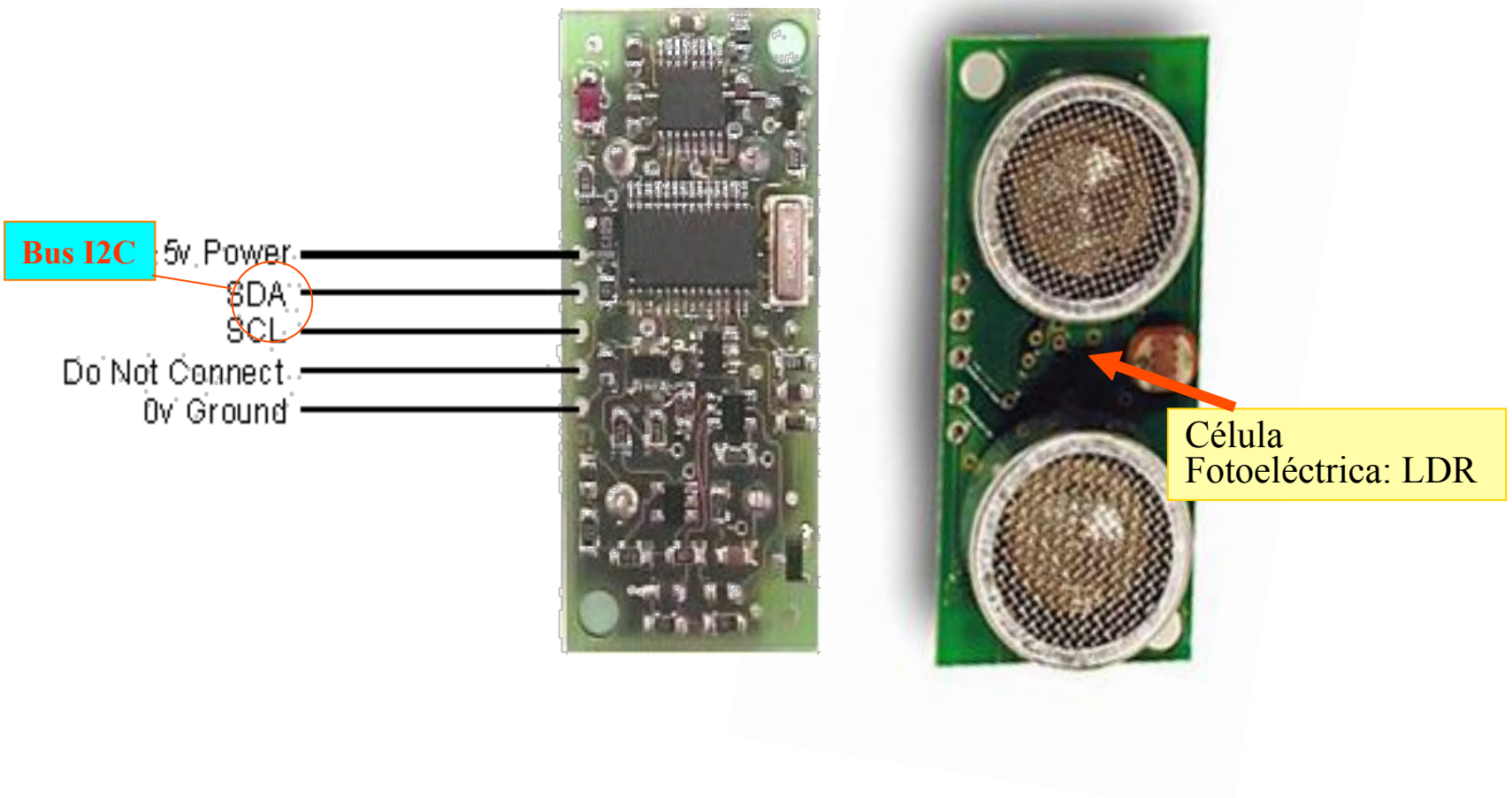
Patrón del haz recibido



Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

Medidor de distancia por ultrasonidos : SRF08

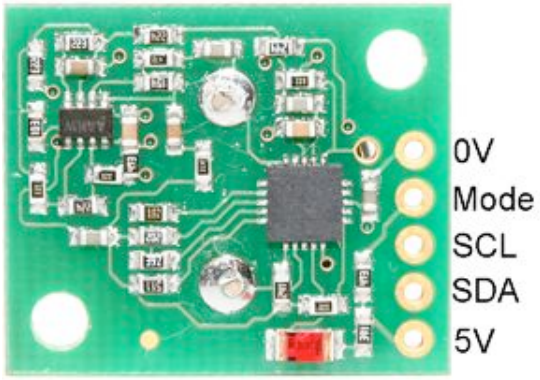


Permite configurar la ganancia de recepción y detectar múltiples ecos

Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

Medidor de distancia por ultrasonidos : SRF02



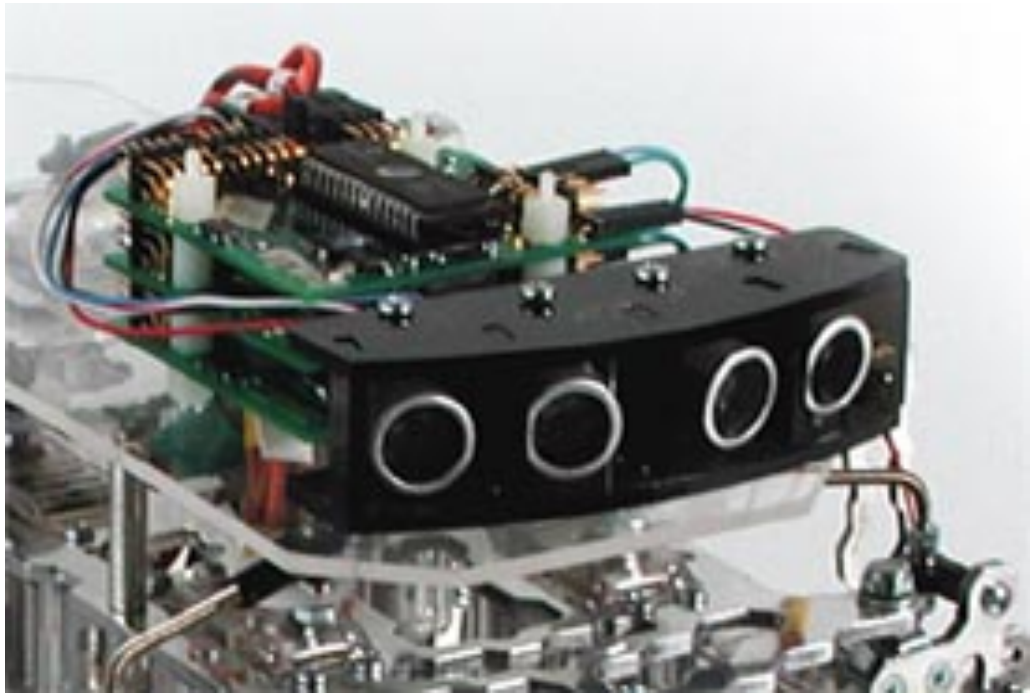
- Señal de ultrasonidos a 40KHz
- Rango: 15cm a 6 m
- Comunicación por I2C

Permite configurar la ganancia de recepción y detectar múltiples ecos

Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

Medidor de distancia por ultrasonidos : SRF08



Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO

Robots con Sensores de Ultrasonidos

Robot Pioneer 2

Nomad

Sensores de Ultrasonidos

Ejemplo de librería de Arduino para uso del SRF04

- **Archivos que componen la librería**
  - DistanceSRF04.h
    - Definición de la clase DistanceSRF04
  - DistanceSRF04.cpp
    - Implementación de las funciones

```
class DistanceSRF04
{
    public:
        DistanceSRF04();
        void begin();
        void begin(int echoPin, int trigPin);

        int getDistanceTime();
        int getDistanceCentimeter();
        int getDistanceInch();

        boolean isCloser(int threshold);
        boolean isFarther(int threshold);

        void setAveraging(int avg);

    private:
        int _trigPin;
        int _echoPin;
        int _average;
        long _duration;
};
```

http://code.google.com/p/srf04-library/
http://jeroendoggen.wordpress.com/2011/12/26/arduino-library-for-the-srf04-ultrasonic-ranger/

Sensores de Ultrasonidos

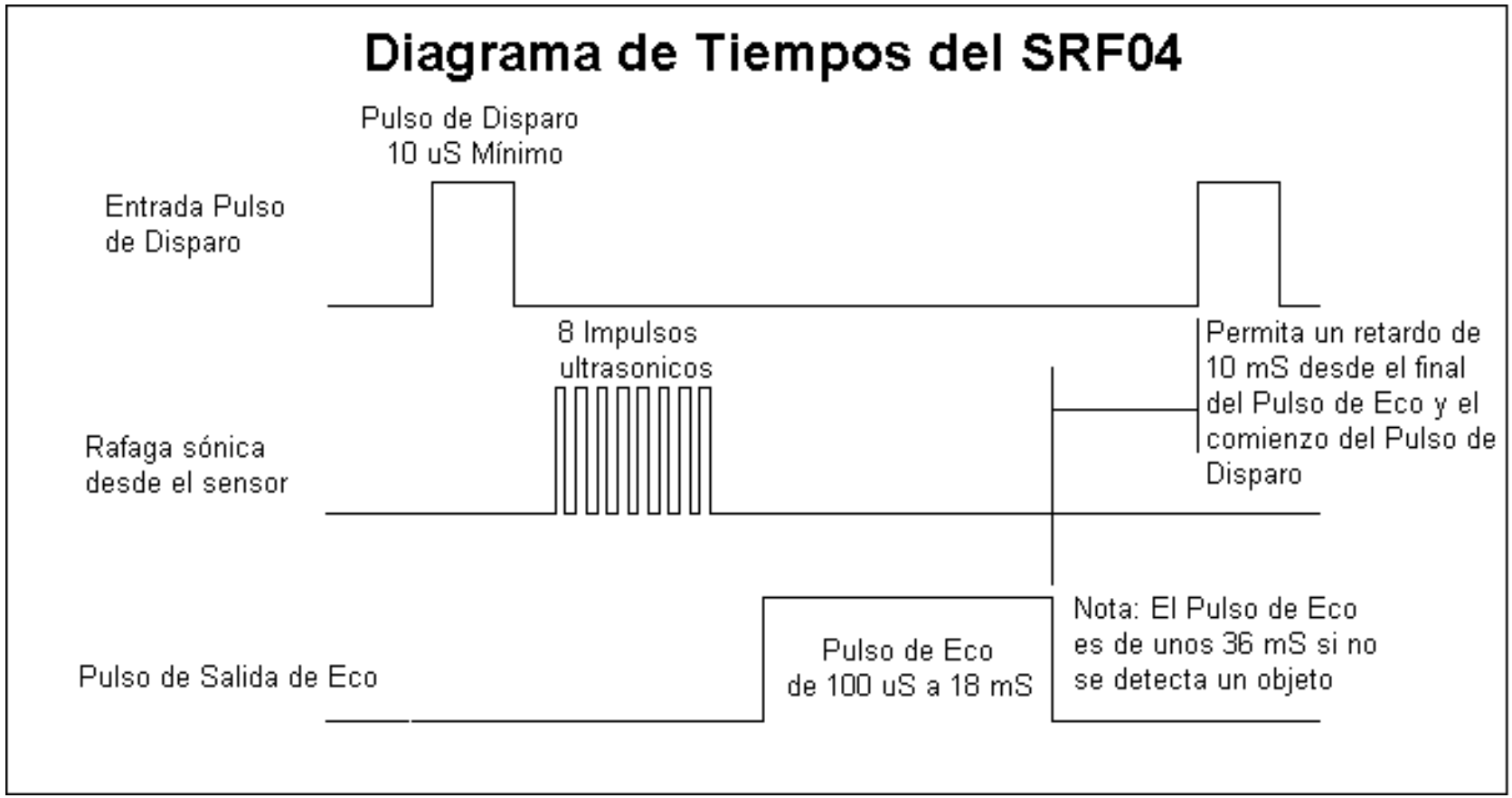
Ejemplo de librería de Arduino para uso del SRF04

- **Funciones de la librería**
  - **void begin(int echoPin, int trigPin);**
    - Configura los pines a los que está conectado el sensor (Echo y Trigger)
    - Si no se especifica presupone Echo en el 2 y Trigger en el 3 (void begin())
  - **int getDistanceTime();**
    - Retorna el tiempo de vuelo medio en ms de varias medidas (*average* medidas).
  - **int getDistanceCentimeter();**
    - Con la velocidad del sonido calcula la distancia en centímetros
  - **boolean isCloser(int threshold);**
    - Indica si está más cerca que un valor umbral (*threshold*) en cm
  - **boolean isFarther(int threshold);**
    - Indica si está más lejos que un valor umbral (*threshold*) en cm
  - **void setAveraging(int avg);**
    - Fija el número de medidas con las que calcular la media de distancia. Por defecto es 1.

http://code.google.com/p/srf04-library/
http://jeroendoggen.wordpress.com/2011/12/26/arduino-library-for-the-srf04-ultrasonic-ranger/

Medida de distancia: Ultrasonidos

PLATAFORMA ARDUINO



Sensores de Ultrasonidos

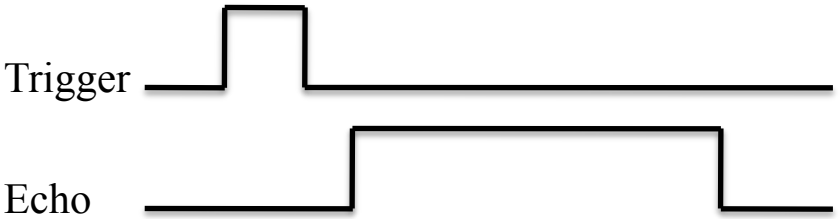
PLATAFORMA ARDUINO

Ejemplo de librería de Arduino para uso del SRF04

- **int getDistanceTime();**

```
int DistanceSRF04::getDistanceTime()
{
    long sum = 0;

    for (int i=0;i<_average;i++)
    {
        digitalWrite(_trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(_trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(_trigPin, LOW);
        _duration = pulseIn(_echoPin, HIGH);
        sum=sum+_duration;
    }
    return(int(sum/_average));
}
```



http://code.google.com/p/srf04-library/
http://jeroendoggen.wordpress.com/2011/12/26/arduino-library-for-the-srf04-ultrasonic-ranger/

Sensores de Ultrasonidos

Función de Arduino para medir el ancho de un pulso

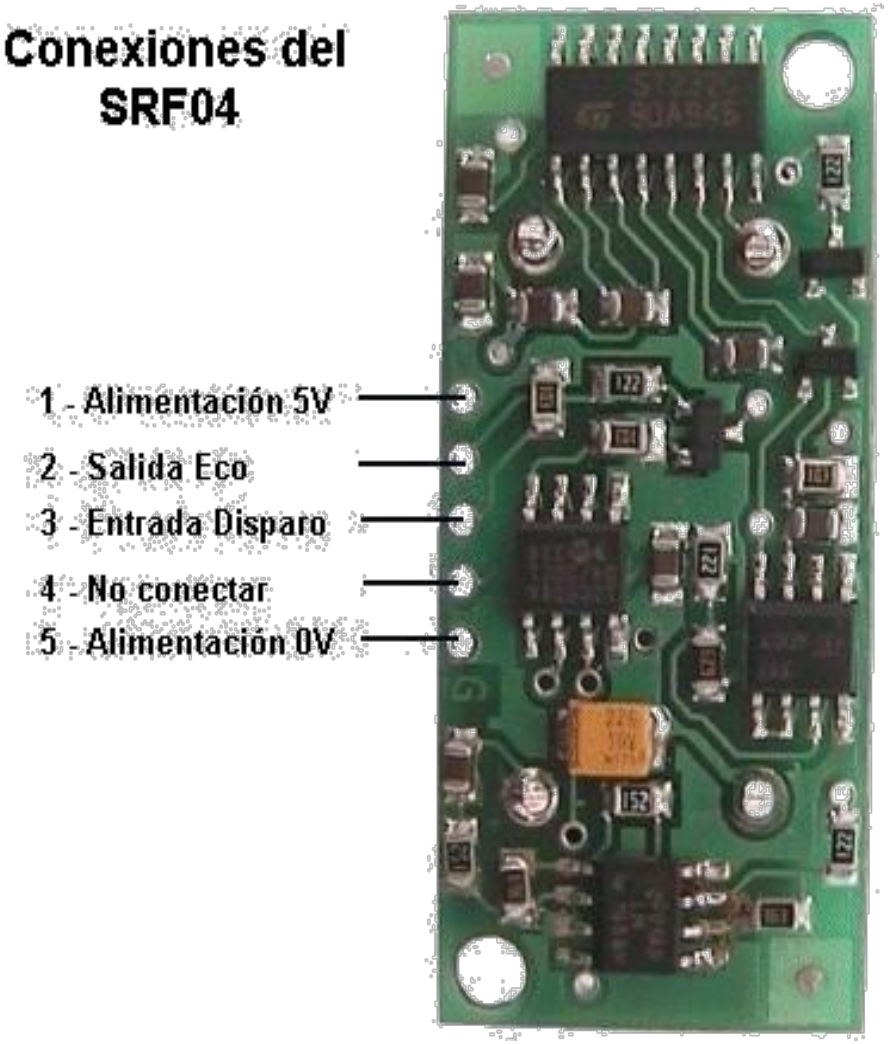
- **unsigned long pulseIn(pin, value)**
- **unsigned long pulseIn(pin, value, timeout)**
  - Mide la duración de un pulso en milisegundos.
  - Si *value* es HIGH
    - Espera hasta que el pin pasa de LOW a HIGH
    - Inicia la medida de tiempo y espera hasta que el pin pasa de HIGH a LOW
    - Devuelve la duración del pulso en milisegundos.
    - Devuelve 0 si el pulso no ha empezado en un cierto tiempo (*timeout*). Si no se especifica *timeout* espera un segundo.
  - Si *value* es LOW
    - Realiza lo mismo pero midiendo el ancho de un pulso a nivel bajo.

http://arduino.cc/en/Reference/PulseIn

Actividad 5.1: Medida de ultrasonidos con el SRF04

- **Objetivo**
  - Utilizar un sensor de ultrasonidos
  - Comprobar cómo se puede medir el ancho de un pulso con Arduino
- **Desarrollo**
  1. Conecta el sensor SRF04 al Arduino UNO. Echo en el 2 y Trigger en el 3.
  2. Realiza un programa que transmita por el puerto serie el valor de una medida cada medio segundo.
  3. Analiza si todas las medidas son coherentes o si hay alguna medida errónea
  4. Modifica el programa para que tome 5 medidas y que envíe por el puerto serie:
     1. El valor medio de las 5 medidas
     2. Y el valor medio de 3 medidas tras descartar la máxima y la mínima
  5. Compara los resultados

Actividad 5.1: Medida de ultrasonidos con el SRF04



Sensores de Ultrasonidos

Ejemplo de librería de Arduino para uso del SRF02

- **Archivos que componen la librería**
  - SRF02.h
    - Definición de la clase SRF02 e implementación de algunas funciones sencillas
  - SRF02.cpp
    - Implementación de las funciones

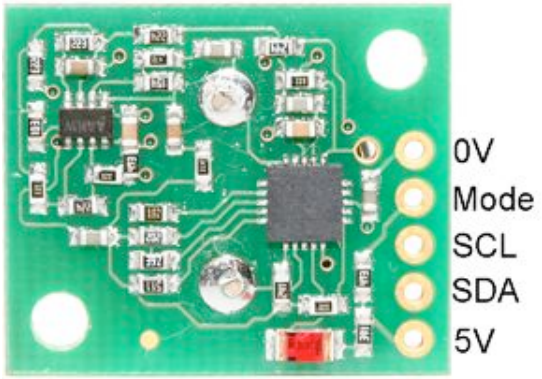
```
class SRF02
{
    public:
       SRF02(uint8_t deviceId, uint8_t mode = SRF02_CENTIMETERS);
       unsigned int read();
       static void update();
       static void setInterval(unsigned int interval);
       static void configureDeviceId(uint8_t deviceId, uint8_t newDeviceId);

    protected:
       void add();
       uint8_t deviceId;                  /** I2C device id */
       uint8_t mode;                      /** Measurement mode */
       unsigned int value;                /** The distance values */
       static unsigned int interval;      /** Measurement interval (milli seconds)*/
       static unsigned long nextRead;     /** When to read the distance sensor values (milli seconds) */
       static unsigned long nextRequest;/** Next time to request distance sensor values (milli seconds) */
       static SRF02 *first;               /** Pointer to the first sensor */
              SRF02 *next;                /** Pointer to the next sensor */
       static bool rangingTriggered;      /** If true, a range measurement was initiated. */
};
```

http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Ejemplo de librería de Arduino para uso del SRF02

- **Funcionamiento del SRF02**
  - Utiliza comunicación I2C permitiendo conectar varios sensores, cada uno con una dirección. La dirección por defecto es 0xE0
    - Direcciones válidas: 0xE0, 0xE2, 0xE4, 0xE6, 0xE8, 0xEA, 0xEC, 0xEE, 0xF0, 0xF2, 0xF4, 0xF6, 0xF8, 0xFA, 0xFC, 0xFE
  - Una medida tarde unos 65ms en llevarse a cabo



http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Ejemplo de librería de Arduino para uso del SRF02

- **Funciones de la librería**
  - SRF02(uint8_t deviceId, uint8_t mode)
    - Configura un nuevo sensor SRF08 con el identificador *deviceId*
    - También se configura el modo de medida:
      - » SRF02_INCHES, SRF02_CENTIMETERS o SRF02_MICROSECONDS

```
SRF02::SRF02(uint8_t deviceId, uint8_t mode)
{
    this->deviceId = deviceId;
    this->mode = mode;
    this->value = 0;
    add();
}

void SRF02::add()
{
    this->next = first;
    first = this;
}
```

http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Sensores de Ultrasonidos

Ejemplo de librería de Arduino para uso del SRF02

- **Funciones de la librería**
  - unsigned int read()
    - Devuelve el valor de medida de distancia
  - void uptate()
    - Realiza la medida de todos los sensores SRF02 configurados
    - Esta función debe ejecutarse continuamente, cada *interval* milisegundos inicializa una lectura de todos los sensores.
  - void setInterval(unsigned int interval)
    - Configura el periodo con el que se realiza la medida de los ultrasonidos en milisegundos.
    - Debe ser superior a 65 que es lo que tarda una medida (*READ_DURATION*)
    - Si es 0 se deshabilita la medida.
  - void configureDeviceId(uint8_t deviceId, uint8_t newDeviceId);
    - Cambia la dirección del dispositivo *deviceId* por *newDeviceId*

http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Funciones básicas de Arduino

- **unsigned long millis()**
  - Devuelve el número de milisegundos desde que la tarjeta Arduino empezó a funcionar. Se desborda cada 50 días.

  - void uptate()
    - Realiza la medida de todos los sensores SRF02 configurados
    - Esta función debe ejecutarse continuamente, cada *interval* milisegundos inicializa una lectura de todos los sensores.
  - void setInterval(unsigned int interval)
    - Configura el periodo con el que se realiza la medida de los ultrasonidos en milisegundos.
    - Debe ser superior a 65 que es lo que tarda una medida (*READ_DURATION*)
    - Si es 0 se deshabilita la medida.
  - void configureDeviceId(uint8_t deviceId, uint8_t newDeviceId);
    - Cambia la dirección del dispositivo *deviceId* por *newDeviceId*

http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Sensores de Ultrasonidos

PLATAFORMA ARDUINO

Ejemplo de librería de Arduino para uso del SRF02

```
void SRF02::update()
{
    if (interval == 0)
    {
        return;
    }
```

Si interval = 0 no realiza medidas

```
    if (rangingTriggered && millis() > nextRead)
    {
        for (SRF02 *i = first; i != 0; i = i->next)
        {
            Wire.beginTransmission(i->deviceId);
            WIRE_WRITE(2);
            Wire.endTransmission();
            Wire.requestFrom(i->deviceId, (uint8_t) 2);
            i->value = ((unsigned long) WIRE_READ()) << 8;
            i->value += (unsigned long) WIRE_READ();
        }
        rangingTriggered = false;
    }
```

Lee la última medida de todos los sensores configurados.
No inicia la medida hasta que no pasen *READ_DURATION* ms desde el disparo de las medidas

```
    if (millis() > nextRequest)
    {
        for (SRF02 *i = first; i != 0; i = i->next)
        {
            Wire.beginTransmission(i->deviceId);
            WIRE_WRITE((uint8_t) 0);
            WIRE_WRITE(i->mode);
            Wire.endTransmission();
        }
        nextRead = millis() + READ_DURATION;
        nextRequest = millis() + interval;
        rangingTriggered = true;
    }
}
```

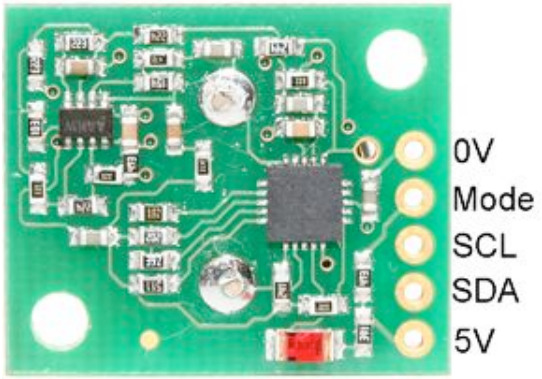
Cada *interval* ms dispara una nueva medida en todos los sensores

Sensores de Ultrasonidos

PLATAFORMA ARDUINO

Funcionamiento del SRF02

- Cada dispositivo tiene un identificador diferente
- Una medida tarde unos 65ms en llevarse a cabo



http://www.superrobotica.com/S320122.htm
http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Sensores de Ultrasonidos

Funcionamiento del SRF02

http://www.superrobotica.com/S320122.htm
http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

- **Envío de un comando**
  - Se envía un cero seguido del comando

Comandos		Descripción
Decimal	Hexadecimal	
80	0x50	Iniciar una nueva medición real. Resultado en pulgadas
81	0x51	Iniciar una nueva medición real. Resultado en centímetros
82	0x52	Iniciar una nueva medición real. Resultado en microsegundos
86	0x56	Iniciar una nueva medida falsa. Resultado en pulgadas
87	0x57	Iniciar una nueva medida falsa. Resultado en centímetros
88	0x58	Iniciar una nueva medida falsa. Resultado en microsegundos
92	0x5C	Transmite una ráfaga de 8 ciclos de 40khz- no hace cálculos de medición
96	0x60	Fuerza un reinicio del sonar SRF02 realizando un ciclo de autoajuste.
160	0xA0	1º comando de la secuencia para cambiar la dirección I2C
165	0xA5	3º comando de la secuencia para cambiar la dirección I2C
170	0xAA	2º comando de la secuencia para cambiar la dirección I2C

```
Wire.beginTransmission(i->deviceId);
WIRE_WRITE((uint8_t) 0);
WIRE_WRITE(i->mode);
Wire.endTransmission();
```

Se inicia una nueva medición en el modo seleccionado (microsegundos, centímetros o pulgadas)

Sensores de Ultrasonidos

Funcionamiento del SRF02

- **Internamente tiene 6 registros que se pueden leer**
  - Primero se indica el registro a partir del que se desea leer y con cada lectura se incrementa el valor del registro a leer.

Registros Nº	Modo de lectura	Modo de Escritura
0	Revisión de software interno	Registros de comandos
1	No usado (se lee 0x80)	No disponible
2	Byte alto de la medidad realizada	No disponible
3	Byte bajo de la medidad realizada	No disponible
4	Byte alto del valor mínimo de distancia	No disponible
5	Byte bajo del valor mínimo de distancia	No disponible

```
Wire.beginTransmission(i->deviceId);
WIRE_WRITE(2);
Wire.endTransmission();
Wire.requestFrom(i->deviceId, (uint8_t)
i->value = ((unsigned long) WIRE_READ()
i->value += (unsigned long) WIRE_READ()
```

Se lee el valor de la medida realizada en 16 bits

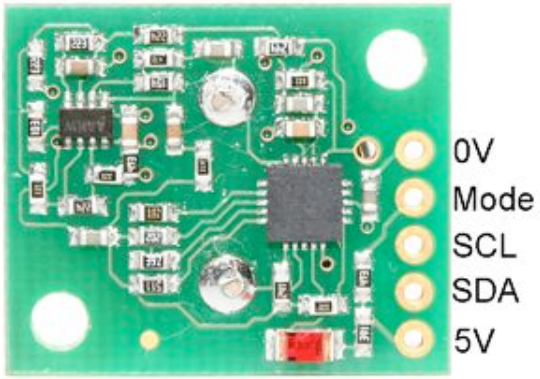
http://www.superrobotica.com/S320122.htm
http://www.grapelabs.org/arduino/libraries/srf02-ultrasonic-distance-sensor

Sensores de Ultrasonidos

Funcionamiento del SRF02

- **Para modificar la dirección**
  - Es necesario que tenga conectado solamente un circuito en el bus.
  - Se debe escribir una secuencia de 3 comandos en el orden correcto y sin interrupción:
    - 0xA0, 0xAA, 0xA5
  - Seguido de la nueva dirección

```
void SRF02::configureDeviceId(uint8_t currentDeviceId, uint8_t newDeviceId)
{
    Wire.beginTransmission(currentDeviceId);
    WIRE_WRITE((uint8_t) 0);
    WIRE_WRITE((uint8_t) 0xA0);
    Wire.endTransmission();
    Wire.beginTransmission(currentDeviceId);
    WIRE_WRITE((uint8_t) 0);
    WIRE_WRITE((uint8_t) 0xAA);
    Wire.endTransmission();
    Wire.beginTransmission(currentDeviceId);
    WIRE_WRITE((uint8_t) 0);
    WIRE_WRITE((uint8_t) 0xA5);
    Wire.endTransmission();
    Wire.beginTransmission(currentDeviceId);
    WIRE_WRITE((uint8_t) 0);
    WIRE_WRITE(newDeviceId << 1);
    Wire.endTransmission();
}
```

Actividad 5.2: Medida de ultrasonidos con el SRF02

- **Objetivo**
  - Utilizar un sensor de ultrasonidos
  - Comprobar cómo se puede medir de un sensor de I2C
- **Desarrollo**
  1. Conecta el sensor SRF02 al Arduino UNO usando los pines SDA (data line) en la entrada analógica 4 y SCL (clock line) en la entrada analógica 5.
  2. Realiza un programa que transmita por el puerto serie el valor de una medida cada medio segundo.