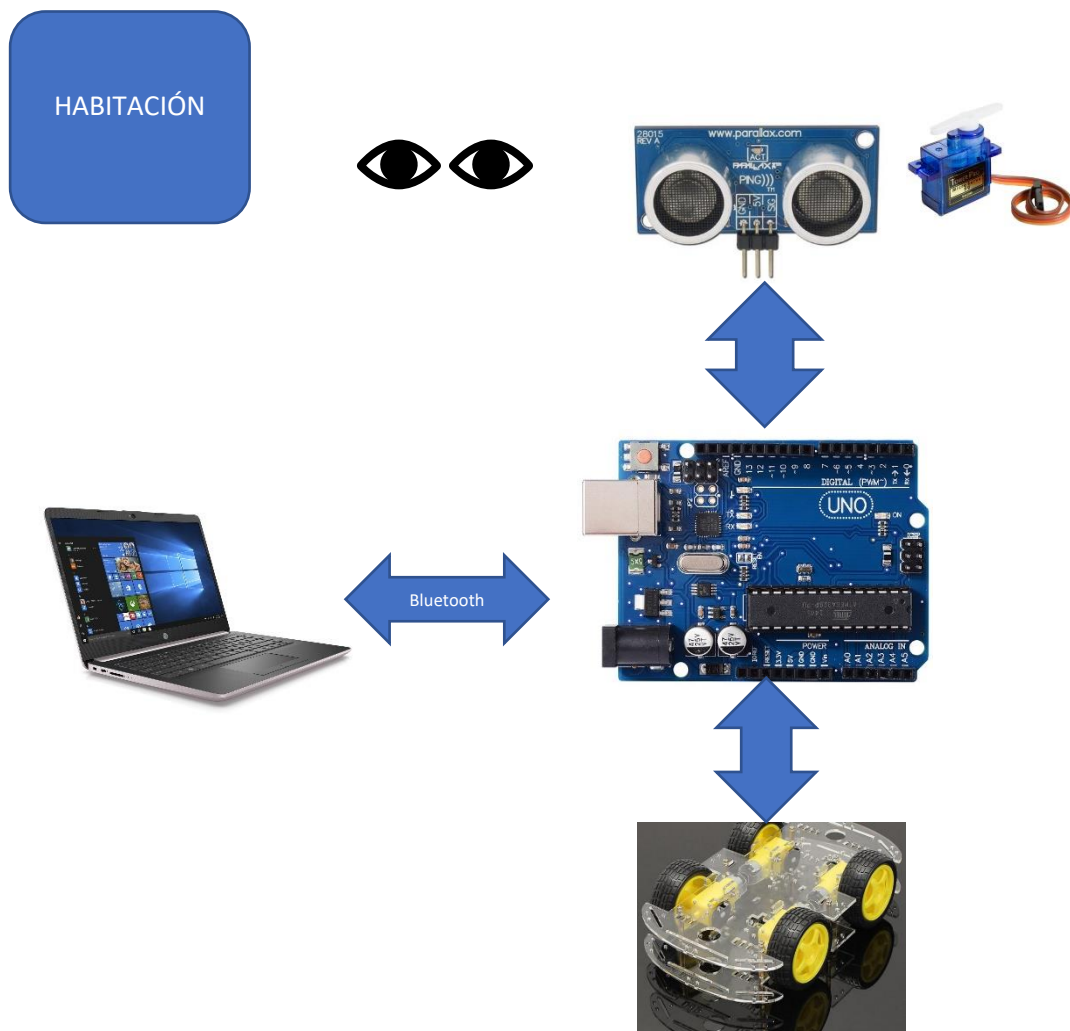


NOMBRE	APELLIDOS	Nº MATRICULA	EMAIL
Pablo	Martínez-Conde Albizu	55352	p.martinez-conde@alumnos.upm.es

TÍTULO Y RESUMEN

Robot de mapeo automático con Processing y Arduino

El robot tendrá integrado un sistema de lectura de distancias mediante ultrasonidos con varios sensores el cual irá recibiendo información y la enviará a un ordenador que procesará esta creando un mapa y su correspondiente representación gráfica en pantalla para que el usuario elija un punto de este mapa y el robot se mueva hasta ese punto intentando reducir la distancia de ruta sin colisionar con ningún obstáculo.



Requisitos funcionales

1º: La aplicación posee una interfaz gráfica para la interacción con el usuario y la representación del mapa, así como el estado del robot.

2º: Al iniciar el proceso, el robot empieza a moverse, y al encontrarse un obstáculo al frente a un límite de distancia mínima de seguridad girará el sensor 360 mediante un servomotor, para que hagan un chequeo total de la distancia de los obstáculos a su alrededor y tras eso, gira 90 grados hacia un lado para seguir con el mapeo siguiendo las paredes, usando otro sensor para mantener una distancia de seguridad a la pared que sigue.

3º: Los sensores captan así los distintos obstáculos que hay a su alrededor, midiendo la distancia y el ángulo.

4º: El ordenador recibe esta información y la procesa para hacer una representación gráfica de esta información para la interpretación por parte del usuario.

5º: El ordenador es capaz de dejar al usuario elegir un punto de este mapa, calcular una ruta que evite paredes con una distancia de seguridad y mandarle las órdenes de movimiento al robot para que se desplace hasta esta posición sin problemas.

6º: El robot debe de evitar situaciones de estancamiento en las que no pueda moverse, y en caso de estar en una situación así (que tenga las mismas lecturas tras 4 giros, por ejemplo, estando en un bucle), que el robot cambie la dirección de giro, o gire una cantidad distinta de grados hasta llegar a otro obstáculo y vuelva a su rutina de giro anterior.

HARDWARE- FUNDAMENTOS TÉCNICOS

Medida de la distancia - Sensor ultrasonidos HC-SR04

Fuente: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>

Un sensor de ultrasonidos es un dispositivo que permite detectar obstáculos y medir distancias. Su funcionamiento se basa en el envío de un pulso de alta frecuencia no audible por el ser humano. Este pulso rebota en los objetos cercanos y es reflejado hacia el sensor, el cual dispone de un micrófono adecuado para esa frecuencia. Midiendo el tiempo entre pulsos y conociendo la velocidad del sonido podemos estimar la distancia del objeto contra cuya superficie impactó el impulso de ultrasonidos.

Los sensores de ultrasonidos son sensores baratos, y sencillos de usar. El rango de medición teórico del sensor HC-SR04 es de 2cm a 400 cm, con una resolución de 0.3cm. En la práctica, sin embargo, el rango de medición real es mucho más limitado, en torno de 20cm a 2 metros. Los sensores de ultrasonidos son sensores de baja precisión. La orientación de la superficie a medir puede provocar que la onda se refleje, falseando la medición. Además, no resultan adecuados en entornos con gran número de objetos, dado que el sonido rebota en las superficies generando ecos y falsas mediciones. Tampoco son apropiados para el funcionamiento en el exterior y al aire libre. Pese a esta baja precisión, que impide conocer con precisión la distancia a un objeto, los sensores de ultrasonidos son ampliamente empleados. En robótica es habitual montar uno o varios de estos sensores, por ejemplo, para detección de obstáculos, determinar la posición del robot, crear mapas de entorno, o resolver laberintos. En aplicaciones en que se requiera una precisión superior en la medición de la distancia, suelen acompañarse de medidores de distancia por infrarrojos y sensores ópticos.

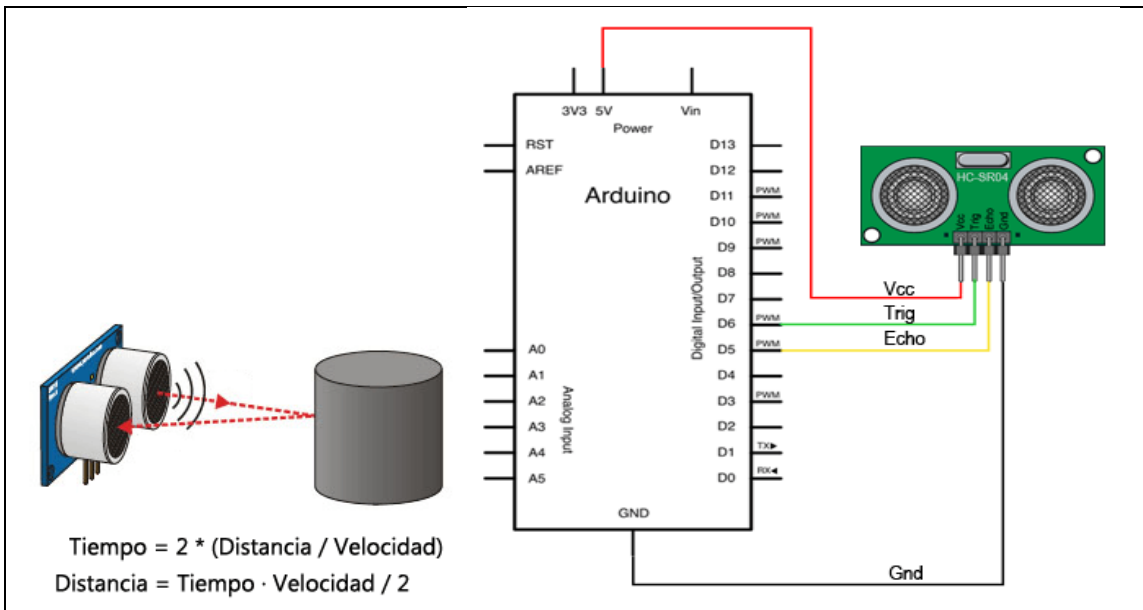
El sensor se basa simplemente en medir el tiempo entre el envío y la recepción de un pulso sonoro. Sabemos que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar.

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s}$$

Es decir, el sonido tarda 29,2 microsegundos en recorrer un centímetro. Por tanto, podemos obtener la distancia a partir del tiempo entre la emisión y recepción del pulso mediante la siguiente ecuación.

$$Distancia(cm) = \frac{Tiempo(\mu s)}{29.2 \cdot 2}$$

NOTA: En el denominador se multiplica por dos ya que el tiempo medido corresponde al empleado por el pulso para ir y volver, de manera que la distancia recorrida es el doble.



Plataforma de giro - Servomotor

Fuente: <https://www.luisllamas.es/controlar-un-servo-con-arduino/>

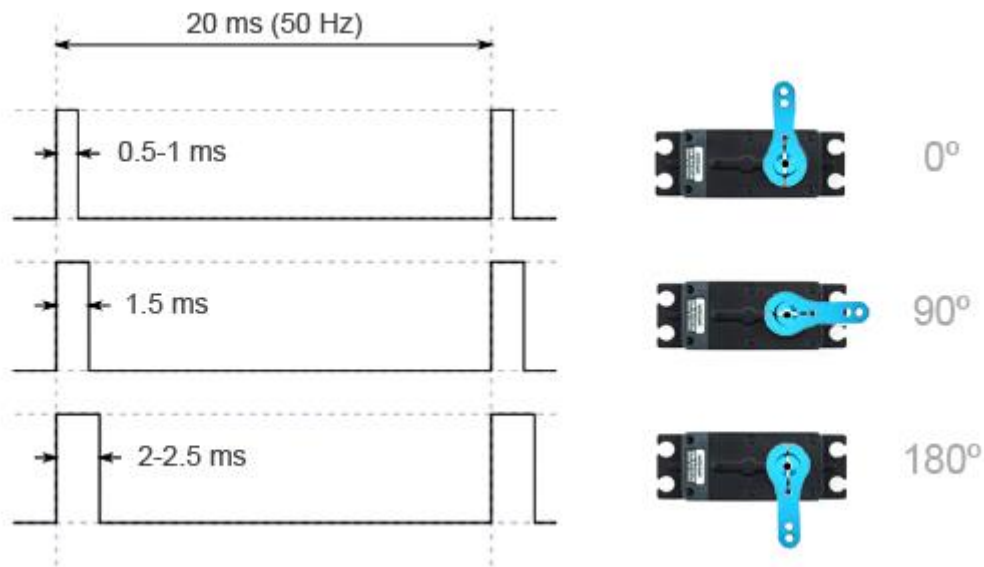
Un servomotor es un accionador ampliamente empleado en electrónica. A diferencia de otros tipos de motores en los que se controla la velocidad de giro, en un servo se indica el ángulo deseado y el servo se encarga de posicionarse en este ángulo.

Típicamente los servos disponen de un rango de movimiento de entre 0 a 180°. Es decir, no son capaces de dar la vuelta por completo (de hecho disponen de topes internos que limitan el rango de movimiento). Internamente un servo está constituido por un motor de corriente continua, acoplado a un reductor para reducir la velocidad de giro, junto con la electrónica necesaria para controlar su posición. Frecuentemente se dispone de un potenciómetro unido al eje del servo que permite al servo conocer la posición del eje. Esta información es tratada por un controlador integrado que se encarga de actuar sobre el motor para alcanzar la posición deseada.

La comunicación de la posición deseada se realiza mediante la transmisión de una señal pulsada con periodo de 20ms. El ancho del pulso determina la posición del servo. La relación entre el ancho del pulso y el ángulo depende del modelo del motor. Por ejemplo, algunos modelos responden con 0° a un pulso de 500 ms, y otros a un pulso de 1000 ms. En general, en todos los modelos un pulso entre 500-1000µs corresponde con 0°, un pulso de 1500µs corresponde con 90° (punto neutro) y un pulso entre 2000-2500µs corresponde con 180°. Por tanto, variando la señal en microsegundos podemos disponer de una precisión teórica de 0.18°-0.36°, siempre que la mecánica del servo acompañe.

Conectar un servo a Arduino es sencillo. El servo dispone de tres cables, dos de alimentación (GND y Vcc) y uno de señal (Sig). En general, la alimentación a los servos se realiza desde una fuente de tensión externa (una batería o una fuente de alimentación) a una tensión de 5V-6.5V, siendo 6V la tensión idónea. Arduino puede llegar a proporcionar corriente suficiente para encender un servo pequeño (SG90), suficiente para hacer unos cuantos proyectos de prueba, sin embargo, no dispone de corriente suficiente para actuar un servo grande (MG996R) ni varios servos pequeños. Además hacer excesiva fuerza con ellos puede exceder la capacidad de corriente de Arduino provocando su reinicio.

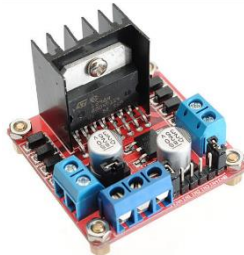
Para el control conectamos el cable de señal (naranja / blanco) a cualquier pin digital de Arduino



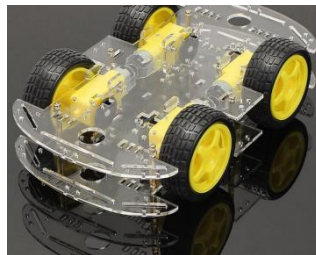
Sistema de movimiento – Conjunto de motores

Fuente: <https://blog.bricogeek.com/noticias/tutoriales/tutorial-robot-4x4-con-arduino/#chasis>

Un motor es un accionador simple que según la orden que reciba gira un componente a más o menos velocidad, usándolo en este caso para girar unas ruedas que permitirán el movimiento del propio robot, en este caso, controlando el movimiento del robot haciendo que el vehículo gire a un lado u otro según convenga, creando en este caso un sistema 4x4 mediante un controlador de motores, para poder darles la suficiente potencia a los 4 y manejarlo todo, ya que la placa Arduino por sí sola no tendría suficiente capacidad para esto.



Este componente permite también la comunicación de la placa Arduino con los motores, para poder darles las órdenes según se requiera.



Para que gire, el sistema hace moverse a las ruedas de un lado en el sentido contrario al otro, y según quiera girar más rápido o menos se ajusta la diferencia de velocidad.

