PRÁCTICA 3 – SENSORES DE DISTANCIA

Un sensor de distancia permite realizar la medida de distancia o desplazamiento entre el propio sensor y un objeto en frente de él una forma automatizada, y proporciona una señal eléctrica según la variación física de distancia. Dependiendo de lo sofisticado que sea el sensor, la señal eléctrica puede ser analógica con una variación de tensión o corriente, digital en forma de pulsos de un tiempo de terminado, o de datos serie.

Esta práctica se trabajará con dos tipos de sensores de distancia muy utilizados en robótica: los que utilizan ultrasonidos y los basados en la reflexión de la luz. En concreto, se experimentará con dos sensores de ultrasonidos, uno de categoría industrial y otro económico, y con un sensor óptico de reflexión económico.

Para estudiar el funcionamiento de los sensores, cada uno de ellos se conectará a un Arduino Uno que será programada para aplicar la curva de calibración del sensor y proporcionar un valor de distancia en metros. La curva de calibración es la función que relaciona la distancia medida por el señor con la señal eléctrica que proporciona. Después, se realizará un conjunto de medidas de la distancia entre cada sensor y dos objetos determinados, de materiales diferentes, para comparar la exactitud y repetitividad de los tres sensores entre sí. Las medidas serán enviadas por Arduino al ordenador mediante la comunicación serie-USB, en formato CSV, de forma que puedan ser fácilmente guardadas en archivos y analizadas con una hoja de cálculo. Finalmente, analizando el funcionamiento de los sensores y la comparación de sus características antes distintos objetos, se concluirá para que tipo de aplicación es más adecuado cada sensor.

1. Conceptos básicos sobre los sensores usados en la rpáctica

1.1. Sensores de distancia mediante ultrasonidos

En la medida de distancias es común el uso de sensores de ultrasonidos. Su funcionamiento se basa en el envío de un pulso de sonido de alta frecuencia (ultrasonidos), no audible por el ser humano, normalmente usan un transductor piezoeléctrico. El pulso de sonido rebota en los objetos cercanos y es reflejado en forma de eco hacia el sensor, que dispone de un micrófono adecuado para captar ultrasonidos, que normalmente se basa en otro transductor piezoeléctrico. Midiendo el tiempo entre el envío del pulso y la recepción del eco en el sensor, y conociendo la velocidad del sonido, se puede estimar la distancia del objeto contra cuya superficie ha impactado el pulso de ultrasonidos. Cabe mencionar que algunos modelos de sensores usan el mismo transductor piezoeléctrico como emisor y receptor, lo que disminuye su tamaño.

La Figura 1 muestra imágenes de los dos sensores con los que se experimentará en esta práctica: El sensor **UK6C/H2-0EUL** de la marca Micro Detectors (Figura 1A), y el sensor estándar **HC-SR04**, proporcionado por múltiples fabricantes (Figura 1B). El sensor UK6C/H2-0EUL es un modelo industrial con forma de tornillo con métrica M18 que proporciona una salida analógica en forma de intensidad de corriente. En cambio, el sensor HC-SR04 es un modelo económico y sencillo, muy utilizado en robótica de pequeños vehículos móviles, que proporciona una salida digital de pulsos. Los siguientes apartados describen las características, principios de funcionamiento, y circuitos de conexión de los dos sensores.

A B

Figura 1. Sensores de distancia por ultrasonidos: (A) UK6C/H2-0EUL, (B) HC-SR04.

Cabe mencionar que, en general, los sensores de distancia basados en ultrasonidos y en la medida de tiempo de vuelo suelen proporcionar poca exactitud en comparación con otras opciones de sensores de distancia. Esto se debe principalmente a que la forma y la orientación de la superficie del objeto de interés pueden provocar que la onda se refleje de forma extraña, 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. Por eso, no son muy apropiados para aplicaciones que requieren medida de distancias en el exterior y al aire libre, por la variación de las condiciones de temperatura y humedad. También hay que considerar que, si se usan múltiples señores de este tipo en un aplicación, y todos se activasen a la vez, las ondas de sonido que emite un sensor pueden falsear la medida de lo os otros. Sin embargo, si se suelen usar mucho como detectores de presencia en exteriores, como por ejemplo en el caso de los sistemas de ayuda al aparcamiento que se incorporan en los vehículos actuales, en las que se compara la distancia aproximada con umbrales preestablecidos, en lugar de hacer medidas de distancia exactas.

1.1.1. Sensor de ultrasonidos HC-SR04

El sensor HC-SR04 es producido por múltiples fabricantes de componentes electrónicos, y por ello las características de este sensor pueden variar dependiendo del fabricante. Pero, en general, los fabricantes suelen indicar unas especificaciones como las siguientes para el sensor HC-SR04:

- Distancia de medida de 2cm a 4m.
- Ángulo de apertura de 15°.
- Velocidad de trabajo de hasta 20 medidas por segundo (20Hz).
- Tensión de alimentación de 5V de corriente continua y consumo de 15mA.
- Salida digital (*echo*) por pulsos con duración proporcional a la distancia.
- Medida de la distancia al recibir un pulso en la entrada de disparo (*trigger*).
- Conexión directa al microcontrolador mediante 4 pines: V_{CC}, GND, *trigger* y *echo*.
- Suministrado como circuito impreso de sin protección especial de tamaño 43 x 20 x 15mm.
- Precio sobre 6,5€ con IVA.

Este sensor funciona midiendo el tiempo entre el envío del pulso de ultrasonidos y la recepción de su eco, que habitualmente se denomina "**tiempo de vuelo**" o **ToF** (*Time of Flight*). El sensor proporciona el tiempo de vuelo medido mediante un pulso a nivel alto en su salida digital "echo", de forma que la duración del pulso corresponde con el tiempo de vuelo. El programa del microprocesador al que se conecta el sensor puede determinar la distancia sabiendo que la velocidad del sonido es aproximadamente 343 m/s en condiciones de un ambiente estándar (temperatura de 20ºC, 50% de humedad, y presión atmosférica a altitud 0m):

$$d = t \cdot v/2$$

La fórmula anterior es la curva de calibración del sensor de ultrasonidos. En ella, *v* es la velocidad del sonido, *t* el tiempo entre la emisión del pulso de ultrasonidos y la llegada de su eco al sensor, y *d* es la distancia entre el sensor y el objeto en donde rebotó el pulso de ultrasonidos. En la fórmula se divide por dos porque el pulso recorre la distancia entre el sensor y el objeto dos veces: en el trayecto desde el sensor al objeto, y en la vuelta desde el objeto al sensor como eco, como muestra la Figura 2.

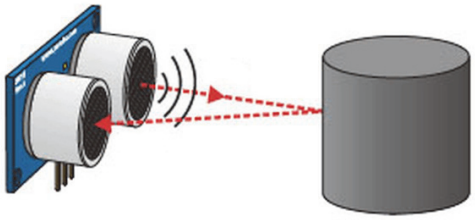

Figura 2. Recorrido del pulso de ultrasonidos.

La velocidad del sonido también se puede expresar en unidades de medida más pequeñas, lo que resulta más cómodo para calcular distancias pequeñas a partir de la medida de tiempos pequeños:

$$343\frac{m}{s}\cdot 100\frac{cm}{m}\cdot\frac{1}{1.000.000}\frac{s}{\mu s}=\frac{1}{29{,}2}\frac{cm}{\mu s}$$

Es decir, el sonido tarda 29,2 microsegundos en recorrer un centímetro. Así, considerando el tiempo t expresado en microsegundos, se tiene la distancia d en centímetros de esta forma:

$$d=\frac{t}{2\cdot 29{,}2}=\frac{t}{58{,}4}$$

La medida de tiempo de vuelo proporcionada por el sensor está limitada, normalmente en función de su rango de medida de distancia. Así, el pulso que proporciona el sensor tiene una duración máxima, pudiéndose tomar este valor como indicativo de que no se detecta un objeto en el rango de medida. En el caso del HC-SR04, aunque el rango fiable de medida es de unos 2cm a 4m, el pulso proporcionado puede tener una duración de hasta 38ms, correspondiendo este tiempo al caso de que el sonido no rebote en ningún objeto. De todas formas, es mejor aceptar como válidas solo las medidas para las que se obtiene una **duración de pulso entre 117µs y 23,324ms**, tiempos que corresponde con el **rango de medida de 2cm a 4m**.

Con sensores como el HC-SR04, es habitual que el microprocesador genere la orden de iniciar una media al sensor mediante una entrada digital de disparo (*trigger*). En la Figura 3 se puede ver esta entrada entre las conexiones del sensor HC-SR04. De esta forma, el sensor espera a recibir la señal de disparo para realizar una medición, y proporcionar el pulso con el tiempo de vuelo medido.

Vcc: Alimentación 5V DC.

Trig: Entrada de disparo (*trigger*).

Echo: Salida del pulso con la medida de tiempo.

GND: Referencia de 0V de la alimenatción.

Figura 3. Conexiones del sensor HC-SR04.

Es importante tener en cuenta que, para enviar la orden de disparo, el microprocesador normalmente debe esperar a que acabe la medida anterior, lo que puede suponer el tiempo correspondiente a una medida de la distancia máxima de su rango de medida. Además de ese tiempo, los sensores de ultrasonidos suelen requerir que el microprocesador espere siempre un tiempo mínimo adicional entre cada orden de disparo. Para el HC-SR04, se suele recomendar un **tiempo mínimo de 50ms entre medidas**, que corresponde con **20 medidas por segundo o 20Hz**.

La conexión del sensor HC-SR04 a un Arduino Uno es directa, y basta con **alimentar el sensor a 5V** y emplear dos pines digitales del Arduino. La Figura 4 muestra un ejemplo de conexión.

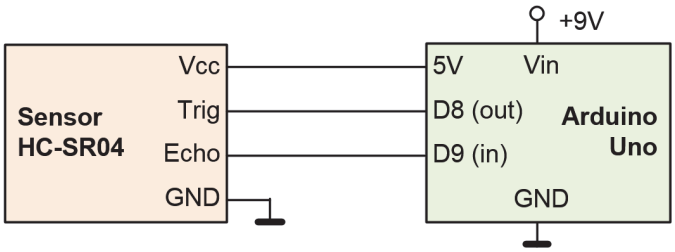

Figura 4. Ejemplo de conexión entre el sensor HC-SR04 y el Arduino Uno.

En un programa de Arduino es fácil determinar la duración de un pulso en una entrada digital usando la función se puede utilizar la función "**pulseIn(pin, value, timeout)**". La función espera a que la señal conectada al pin número "pin" (configurado como entrada digital) cambie al nivel indicado por "valor" (que puede ser HIGH o LOW), y después cronometra el tiempo que permanece la entrada a ese nivel, devolviendo el tiempo como un valor "unsigned long" en microsegundos. Con el parámetro "timeout" se indica el tiempo máximo que la función espera a que el pulso acabe; si se alcanza este tiempo máximo, la función devuelve 0.

Cabe destacar que el sensor HC-SR04, así como el modelo similar Ping, están disponibles como componentes del **simulador Tinkercad – Circuits**. De este modo, los programas para este sensor se pueden simular y depurar con Tinkercad. Como muestra la Figura 5, durante la simulación del circuito se muestra la zona de visibilidad del sensor y un objeto circular delante del sensor, el cual se puede desplazar con el puntero del ratón, además de un indicador de la distancia a la que se encuentra el objeto del sensor. El sensor devolverá una medida de tiempo en su pin de Echo en función de donde esté el objeto. Pero hay que tener en cuenta que la simulación considera propiedades del objeto como su forma 3D o el material de que está hecho.

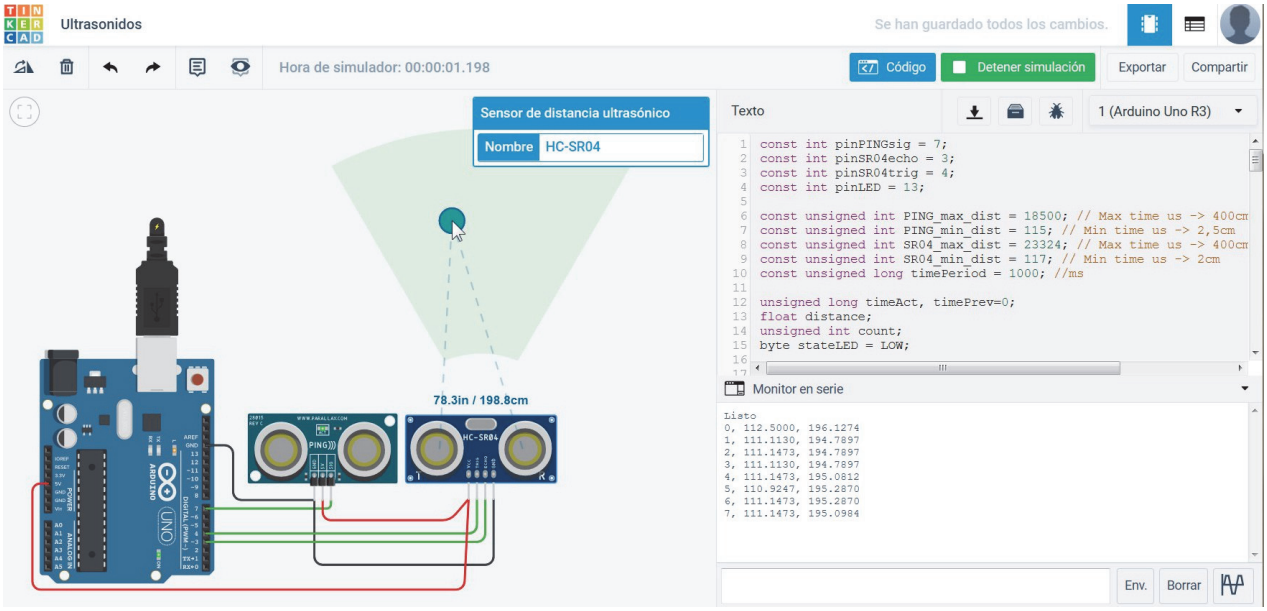

Figura 5. Simulación de los sensores HC-SR04 y Ping en Tinkercad - Circuits.

Para obtener información más detallada sobre cómo utilizar el sensor HC-SR04 con Arduino se pueden consultar estas referencias:

- Datasheet del sensor HC-SR04, disponible entre en los materiales de la asignatura.
- Medir distancia con Arduino y sensor de ultrasonidos HC-SR04, Luis Llamas.
  https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/
- Using the HC-SR04 Ultrasonic Distance Sensor with Arduino, DroneBot Workshop.
  https://dronebotworkshop.com/hc-sr04-ultrasonic-distance-sensor-arduino/

1.1.2. Sensor de ultrasonidos UK6C/H2-0EUL

Aunque el principio de funcionamiento de un sensor de distancia por ultrasonidos en formato industrial, como el UK6C/H2-0EUL, es igual al descrito para el sensor económico HC-SR04 en el apartado anterior, el modelo industrial suele presentar las siguientes ventajas con respecto al económico: es más robusto y soporta entornos más duros, ofrece mejor repetitividad, proporciona la medida de forma simple como un valor de tensión o corriente con un rango estándar, y es más fácil de poner en marcha y calibrar. Como inconveniente principal, su precio es mayor.

Las principales características del sensor UK6C/H2-0EUL de la marca Microdetectors son las siguientes:

- Distancia de medida de 6cm a 80cm (para objeto metálico de 100mm x 100mm).
- Ángulo de apertura de 7°±2°.
- Tensión de alimentación de 10V a 30V de corriente continua. Consumo de 35mA sin contar la salida.
- Salida analógica de corriente (*output*), con el rango estándar de 4mA a 20mA.
- Medición constante de la distancia con una velocidad de hasta 5 medidas por segundo (5Hz).
- Conector M12 para el cableado, con 4 conexiones: V_{CC}, GND, *output* y *teach*.
- Curva de calibración lineal: la corriente de salida depende linealmente de la distancia.

- Forma de tornillo de métrica M18, con carcasa de plástico y protección a IP67 (no se ve afectado por el polvo, y puede sumergirse en agua durante 30 minutos).
- Modo de programación para adecuar un rango de medida al rango de corriente de la salida.
- Funcionamiento de -20°C a 70°C, y compensación ante variaciones de temperatura.
- Precio sobre 130€ con IVA.

La Figura 6 detalla las conexiones disponibles en sensor UK6C/H2-0EUL a través de un conector M12, así como los colores típicos del cableado utilizado con sensores industriales de cuatro conexiones. Este sensor no dispone de una entrada de inicio o disparo como el HC-SR04, y está constantemente realizando mediciones.

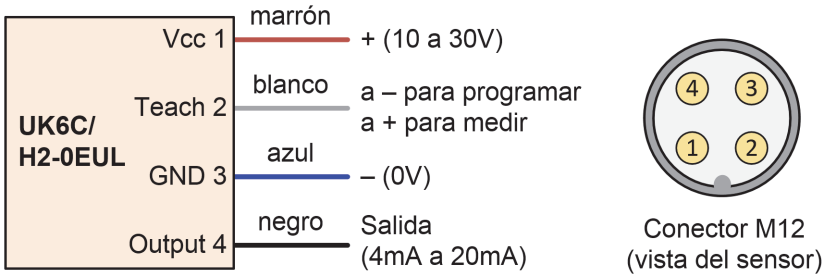

Figura 6. Conexiones del sensor UK6C/H2-0EUL.

La línea "**teach**" (pin 2 o cable blanco) se utiliza **para programar el sensor cuando se conecta al negativo o 0V** de la fuente de alimentación. En funcionamiento normal, **para realizar medidas de distancia, la línea "teach" debe conectarse al positivo o V_{cc}**, o dejarse sin conectar.

El sensor UK6C/H2-0EUL también incluye dos indicadores LED:

- LED naranja. En funcionamiento normal, este LED indica que se detecta un objeto en el rango de medida programado. En el modo de programación, indica los pasos de programación o posibles errores.
- LED verde, que indica que el sensor recibe el eco de ultrasonidos desde un objeto. Esto es, el sensor detecta que tiene un objeto delante.

El sensor UK6C/H2-0EUL proporciona en su **salida un valor de intensidad de corriente, en el rango de 4mA a 20mA**, que es proporcional a la distancia medida a un objeto de interés delante del sensor, cuando el objeto se encuentra dentro de un rango de distancias programado. Así, este sensor tiene una **curva de calibración lineal**, conforme a una de las gráficas mostradas en la Figura 7. Los valores de distancia P1 (objeto lejano) y P2 (objeto cercano) definen el rango de distancias de medida, y se pueden programar con valores dentro del rango de medida del sensor (6cm a 80cm), siguiendo el proceso que se describe en el apartado 1.1.3. También se puede programar si el sensor ofrece una curva con **pendiente positiva (valor por defecto) o negativa.**

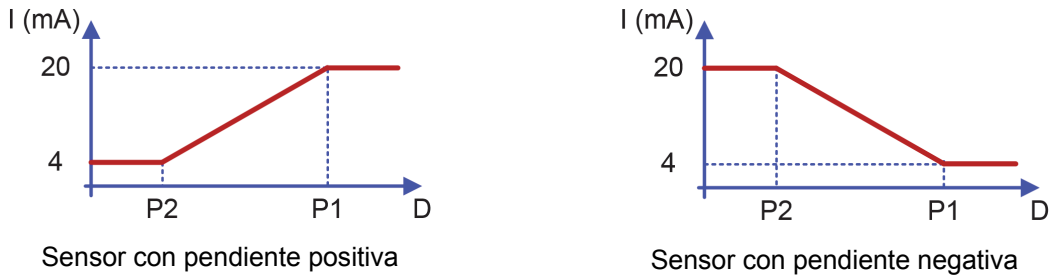

Figura 7. Curva de calibración del sensor UK6C/H2-0EUL.

El rango de intensidades de corriente de 4 a 20mA se usa mucho con sensores que proporcionan una salida analógica, porque permite distinguir entre un valor de salida válido de fallos en el sensor, el cableado o los circuitos de acondicionamiento. De este modo, si se produce un fallo y no llega suficiente corriente al conversor ADC, el *software* debe interpretar que no se ha podido medir, en vez de que la medida es cero.

Para conectar un sensor industrial que se alimenta a más de 5V y que proporciona una salida analógica de corriente, como es el caso del sensor UK6C/H2-0EUL, a una entrada analógica de un Arduino Uno que admite tensiones de 0V a 5V, se puede utilizar un sencillo circuito de acondicionamiento como el de la Figura 8. La resistencia **R_E se debe calcular para que la máxima salida de corriente del sensor produzca la máxima tensión de entrada analógica**, y así se aproveche el rango de la entrada analógica sin sobrepasar su máximo.

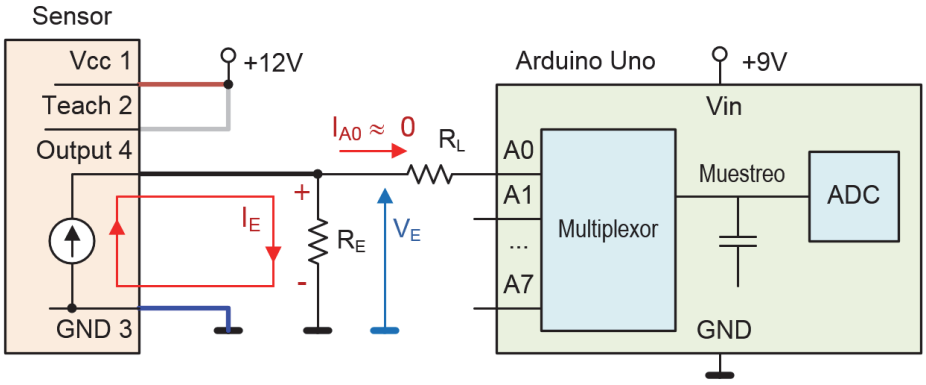

Figura 8. Ejemplo de conexión entre el sensor UK6C/H2-0EUL y el Arduino Uno.

Teniendo en cuenta que una entrada analógica del Arduino Uno admite tensiones V_E de 0V a 5V, y que el sensor proporciona una intensidad de corriente I_E de 4mA a 20mA, el valor de R_E debe ser:

$$R_E = \frac{V_{Emax}}{I_{Emax}} = \frac{5}{0{,}02} = 250\Omega$$

Como el valor de 250Ω no está considerado en las series estándar de resistencias más comunes (E24 y E48), puede ser necesario utilizar más de una resistencia en serie o en paralelo. Por ejemplo, con dos resistencias en serie de 220Ω y 33Ω se tiene el valor de 253Ω, o con dos resistencias en paralelo de 330Ω y 1KΩ se tiene el valor de 248, que son valores próximos al necesario.

En el caso que el sensor proporcione el valor mínimo de corriente de I_E=4mA, se tendrá una tensión V_E de 1V:

$$V_{Emin} = R_E \cdot I_{Emin} = 250 \cdot 0{,}004 = 1\text{V}$$

De este modo, para el rango de distancias que tiene configurado el sensor, se tendrá un **rango de medida de tensión de 1V a 5V en la entrada analógica** del Arduino Uno, que el ADC trasladará a un valor entero de 205 a 1023 en el programa. Y dado que el sensor tiene una curva de calibración lineal, el valor leído dependerá de la distancia de forma lineal, conforme a una de las curvas de la Figura 7. Además, en el programa se puede interpretar que, si el valor de tensión medida es inferior a 1V (correspondiente a 4mA), hay un problema en el sensor, el circuito de acondicionamiento o en la entrada analógica del Arduino Uno.

Un aspecto que hay tener en mente es que, en el cálculo anterior de R_E, se considera que la intensidad de corriente I_{A0} que entra o sale por entrada analógica del Arduino es nula. Esto es cierto si la impedancia o resistencia de la entrada es lo suficientemente alta frente a la impedancia del circuito exterior, que en este caso es R_E. Afortunadamente es así, porque un ADC suele tener un amplificador operacional como primer elemento en sus entradas. En el *datasheet* del controlador ATmega328P que incluye un Arduino Uno, el fabricante explica que la impedancia de un circuito que se conecte a una entrada analógica debe ser de 10KΩ o menos para garantizar el funcionamiento correcto de los circuitos internos de multiplexación y muestro dentro de periodo de muestreo mínimo. Como R_E = 250Ω es 40 veces menor que 10KΩ, no debe haber problemas en la medida de la corriente.

Finalmente, también hay que tener en cuenta que conviene incluir un circuito de protección que evite tensiones superiores a 5V en una entrada del Arduino Uno cuando esa entrada se conecta directamente a un circuito que está alimentado a tensiones mayores, como es el caso del sensor de ultrasonidos utilizado. Esto es aconsejable aun en el caso de que, según el diseño teórico anterior, en la entrada no debería haber más de 5V cuando todo funciona bien. El circuito de protección más sencillo es una simple resistencia en serie que absorba el exceso de tensiones que pueden llegar a la entrada, como R_L en el circuito de la Figura 8. El valor de

esta resistencia no es crítico, ya que la corriente en la entrada es muy pequeña, pero si conviene que sea bastante inferior a la impedancia de la entrada, como se explica en el párrafo anterior. En esta práctica, se utilizará una resistencia R_L de 220Ω.

Para obtener información más detallada sobre el sensor UK6C/H2-0EUL se puede consultar estas referencias:

- Datasheet del sensor UK6C/H2-0EUL disponible en los materiales de la asignatura.
- Catálogo de los sensores de la serie UK6 de Microdetectors: http://microdetectors.rs/pdf/201701_MD_product_catalogue_UK6%20UKR6_ENG.pdf

1.1.3. Programación del sensor de ultrasonidos UK6C/H2-0EUL

Para aprovechar la resolución de la salida analógica, el fabricante recomienda programar el sensor usando el siguiente procedimiento, el cual define un rango de medida de distancias entre un punto P1 lejano del sensor, y otro punto P2 cercano al sensor conforme a la Figura 7:

6. **Ajuste de la posición del punto P1**:
   - Fijar el sensor en la posición de trabajo, y colocar un objeto fijo, plano, y sólido de forma perpendicular al eje del sensor, a la distancia máxima de trabajo P1. El LED verde debe estar encendido.
   - Conectar el cable blanco (*teach*) al cable azul (GND) durante al menos 2 segundos, y menos de 6 segundos, y después desconectar el cable blanco.
   - El LED verde se apagará, y el LED naranja se apagará (si estaba encendido) y se volverá a encender después de unos 2 segundos, parpadeando después rápidamente (con frecuencia de 5 Hz).
   - El LED naranja continuará parpadeando hasta que se adquiera el punto P2.
7. Ajuste de la posición del punto P2:
   - Colocar el objeto a detectar a la distancia mínima de trabajo P2. El LED verde está apagado.
   - Conectar el cable blanco (*teach*) al cable azul (GND) durante al menos 2 segundos, y después desconectar el cable blanco.
   - El LED verde se encenderá, y el LED naranja parpadeará 5 veces a una frecuencia baja, lo que indica que la programación ha finalizado. Si el LED naranja no parpadea 5 veces, ha habido un fallo en la programación, y hay que desconectar la alimentación del sensor y volver a empezar la programación.
   - Conectar el cable blanco (*teach*) al cable marrón (V_{CC}).

Todos los sensores ultrasónicos están configurados por defecto con pendiente positiva. Pero también es posible cambiar entre pendiente positiva y negativa con los siguientes pasos:

- Conectar el cable blanco (*teach*) al cable azul (GND) durante al menos 6 segundos, hasta que el LED naranja comience a parpadear muy rápidamente (con frecuencia de 13 Hz).
- Desconectar el cable blanco (*teach*). El LED naranja parpadeará 5 veces.

Cuando se da una de las siguientes circunstancias, se produce un error de programación:

- Primero se ajusta el punto P2 (punto más cercano) y después de P1 (punto más lejano).
- Se ajusta el punto P1 (punto más lejano) dentro del rango, y el punto P2 en el infinito.

En estos dos casos, el LED naranja emite dos destellos, y el sensor mantiene en la memoria la ventana de distancias programada anteriormente. Para volver a programar el sensor se requiere desconectar la alimentación del sensor, y comenzar de nuevo su programación.

1.2. Sensores de distancia ópticos por reflexión

En aplicaciones en que se requiera medir distancia con un sensor de bajo coste, pero con una precisión mayor que la ofrecida por un sensor de ultrasonidos sencillo, se requiere usar otro tipo sensor, como por ejemplo los sensores ópticos por reflexión que utilizan un emisor de luz y un fotoreceptor de infrarrojos.

Un tipo de sensores ópticos de distancia con una buena relación de exactitud/precio es el que utiliza como fotoreceptor un PSD (*Position Sensitive Detector*) lineal. Básicamente un PSD es dispositivo semiconductor capaz de medir la posición de un punto de luz de un haz que incide sobre su superficie, dando una corriente o tensión proporcional a la posición del punto. Este tipo de sensores incluye además un LED como emisor de un haz de luz infrarroja espontánea (no LASER), y las lentes adecuadas para dirigir el haz de luz en el rango de distancias de trabajo. La medición de distancia se efectúa dirigiendo el haz de luz infrarroja hacia el objeto de interés, y analizando el haz de luz reflejado cuando incide sobre el PSD, como representa la Figura 9.

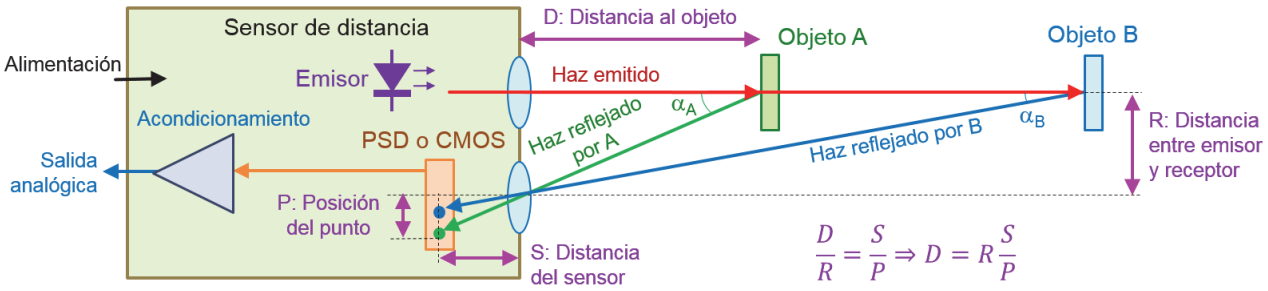

Figura 9. Esquema del funcionamiento de un sensor de distancia basado en un PSD.

Más concretamente, el sensor determina la **medida de distancia por triangulación**: la distancia se obtiene en función de la posición del punto del punto de luz del haz reflejado en el PSD, y esa posición depende del ángulo con que el haz se refleja en el objeto de interés, que a su vez depende de la distancia de dicho objeto al sensor. Dependiendo del modelo del señor, la medida se proporciona a un microcontrolador en forma de una señal digital o analógica.

Además lo sensores basados en PSD son pequeños y realizan las medidas rápidamente, pudiendo operar a frecuencias de medida de hasta 100KHz. Los principales inconvenientes de este tipo de sensores son los siguientes: tiene un valor máximo del rango de medida que no suele sobrepasar el medio metro, la medida puede ser errónea si el objeto es transparente o tiene superficies reflectantes no perpendiculares al objeto, y además son sensibles a la luz ambiente debido a que los rayos de sol también emiten en el espectro de luz infrarroja. Por este motivo, son sensores que se utilizan habitualmente en entornos interiores, con iluminación artificial o controlada.

1.2.1. Sensor de distancia GP2Y0A21YK0F

Entre los sensores ópticos de distancia por reflexión más compactos, económicos y utilizados en robos pequeños, se encuentran varios modelos fabricados por la marca Sharp. Por ejemplo, el sensor GP2Y0A21YK0F que se utilizará en esta práctica, cuyo aspecto y conexiones se muestran en la Figura 10, y cuyas características principales se listan a continuación:

- Distancia de medida de 10cm a 80cm.
- Velocidad de trabajo en torno a 26 medidas por segundo (26Hz).
- Tensión de alimentación de 4,5V a 5,5V de corriente continua. Consumo de 30mA.
- Salida analógica de tensión (V_O), de 0,4V a 2,3V para el rango de medida.
- Conector de tres pines: V_{CC}, GND, y salida V_O.
- Paquete de plástico con tamaño de 40 x 19 x 13,5mm, sin protección especial.
- Precio sobre 10€ con IVA.

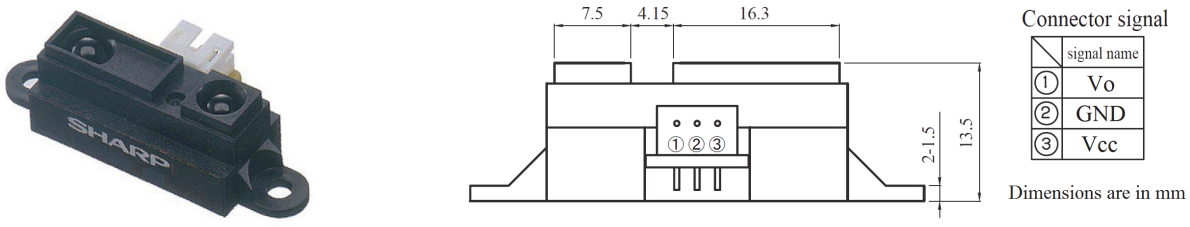

Figura 10. Sensor de distancia por reflexión GP2Y0A21YK0F de Sharp.

La conexión del sensor GP2Y0A21YK0F a un Arduino Uno se puede hacer directamente utilizando una de las entradas analógicas, y alimentando el sensor con la tensión de 5V proporcionada por el Arduino, según muestra la Figura 11.

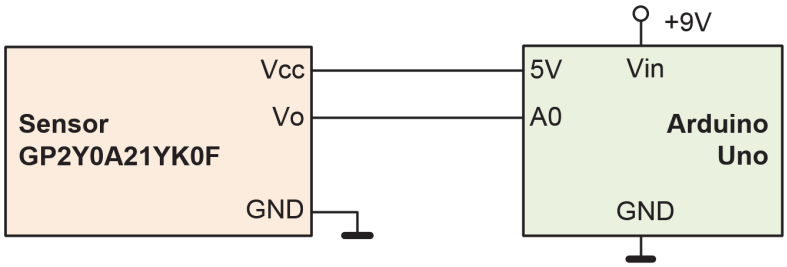

Figura 11. Ejemplo de conexión entre el sensor GP2Y0A21YK0F y el Arduino Uno.

Es importante tener en cuenta que **la medida analógica de tensión que proporciona el sensor GP2Y0A21YK0F no es lineal respecto a la distancia**, y obedece a una curva como la mostrada en la Figura 12, que es la que el fabricante proporciona como ejemplo en el *datasheet* del sensor. Como la curva de ejemplo tiene una tolerancia, es aconsejable calibrar el sensor para obtener la mejor aproximación entre el voltaje que ofrece el sensor y la distancia real medida. En el apartado siguiente se explica cómo se puede llevar a cabo la calibración. Además, como la tensión de salida del sensor varía entre 0V y 3V aproximadamente, no se aprovecha bien el rango de entrada de 0V a 5V del ADC del Arduino Uno, el cual viene establecido por la tensión de referencia por defecto de 5V. Concretamente, según la Figura 12, el sensor proporciona unos valores de tensión entre **aproximadamente 2,25V y 0,4V para el rango de distancias de 10cm a 80cm**.

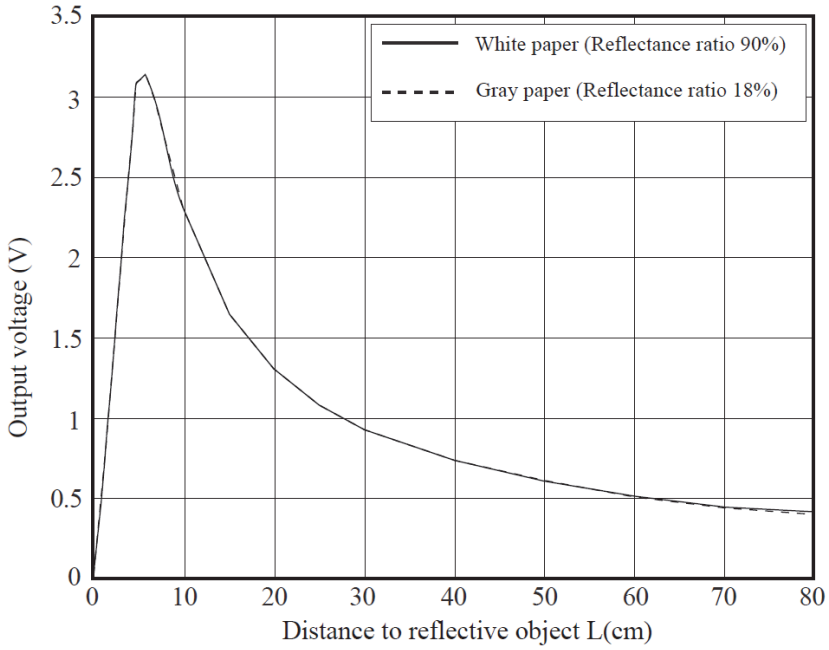

Figura 12. Curva de calibración de ejemplo para el sensor GP2Y0A21YK0F.

En las siguientes referencias se puede encontrar información sobre el funcionamiento y uso del sensor GP2Y0A21YK0F, u otros modelos muy similares.

- Datasheet del sensor GP2Y0A21YK0F de Sharp, disponible en los materiales de la asignatura.
- Distance Measuring Sensors, Sharp. Lista de sensores de distancia PSD de Sharp, con sus *datasheets*. http://www.sharp-world.com/products/device/lineup/selection/opto/haca/diagram.html
- Towards a more Accurate Infrared Distance Sensor Model, Paulo Malheiros, José Gonçalves and Paulo Costa. Artículo que aborda el modelado del sensor GP2Y0A21YK0F de forma detallada. https://bibliotecadigital.ipb.pt/bitstream/10198/4119/1/iscies09_sharp_model.pdf

1.2.2. Calibración del sensor GP2Y0A21YK0F

La calibración de un sensor consiste básicamente en obtener la curva de calibración que da los valores de voltaje en función de la distancia para el sensor concreto utilizado. Lo mejor es disponer de dicha curva como una función antes que una gráfica, de forma que se pueda obtener su función inversa para poder calcular los valores de distancia a partir de los de voltaje dentro del microcontrolador al que se conecta el sensor. Esa función matemática se puede conseguir de dos formas:

- **Mediante aproximación de la curva de calibración** gráfica obtenida del manual del *datasheet* fabricante a la curva de una función matemática usando algún método de ajuste. O mejor aún, obteniendo experimentalmente una tabla de valores de tensión para distancias conocidas, y buscando una función cuya curva se ajuste a esos valores. Se consigue así una función que se puede implementar directamente en un programa. El inconveniente es que la función puede suponer cálculos complejos que requieren un tiempo de proceso destacable.
- **Mediante una tabla con un conjunto suficientemente amplio de pares de distancias y tensiones** obtenidos de una gráfica del manual del *datasheet*, o mejor partir de mediciones reales, con los pares ordenados según el valor de tensión. Esta tabla se la conoce como **tabla *look-up***, y se puede utilizar directamente en el programa del microcontrolador para obtener un valor de distancia a partir de la tensión asociada a un dato leído del ADC. El programa simplemente debe buscar los dos pares consecutivos de la tabla que tienen los valores de tensión entre los que se encuentra la tensión capturada, y calcular la distancia mediante una interpolación lineal entre los valores de distancia de esos dos pares. La principal ventaja de la tabla es que permite una conversión muy rápida de los valores de tensión a distancia, aunque a costa de un mayor uso de memoria.

Se puede escoger entre uno de los dos métodos anteriores en función de los requerimientos de la aplicación donde se use el sensor. En las siguientes referencias se describen ejemplos prácticos de la calibración del sensor GP2Y0A21YK0F basados en los dos métodos descritos:

- Medir distancias con Arduino y el sensor SHARP GP2Y0A02YK0F, Luis Llamas 2016.
  https://www.luisllamas.es/arduino-sharp-gp2y0a02yk0f/
- Using the IR Sensor GP2Y0A21YK…, WoodUino.ca. Ejemplo de uso de un sensor GP2Y0A21 de Sharp.
  http://woodsgood.ca/projects/2015/02/02/using-the-ir-sensor-gp2y0a21yk/
- Linearizing Sharp Ranger Data, Acroname. Ejemplo de cómo obtener la función inversa de la curva de calibración de un sensor GP2Y0A21 de Sharp.
  https://acroname.com/articles/linearizing-sharp-ranger-data/

2. Proceso de medición usando Arduino

2.1. Medidas a realizar

Para evaluar los tres sensores descritos en la sección anterior, en esta práctica se realizará un conjunto de mediciones de distancia para cada sensor, situando un mismo objeto a distancias de referencia o conocidas del sensor. En concreto se considerarán distancias de referencia entre el sensor y el **objeto cada 10cm el rango de 10cm a 70cm, más unas distancias adicionales fuera del rango para 5cm y 80cm**, como representa la Figura 13. Así, se tienen un total de **9 distancias de referencia**.

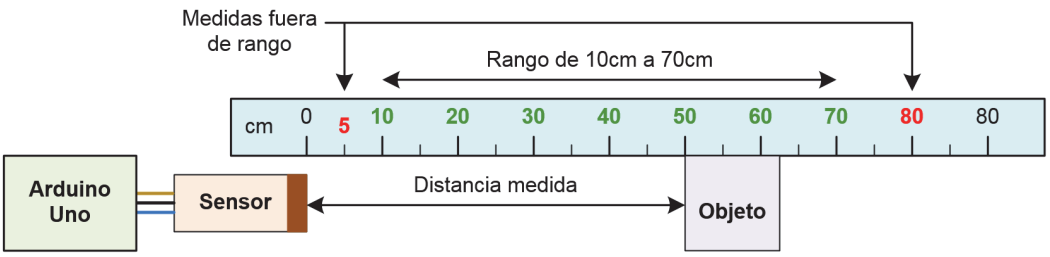

Figura 13. Medidas de distancia de referencia.

Para cada sensor y distancia de referencia se llevarán a cabo 5 medidas, de forma que se pueda extraer información sobre la repetitividad del sensor. Así, se dispondrá de **series de 45 medidas para cada sensor**. Además, con el objetivo de comparar bien las diferencias de funcionamiento entre los sensores basados en ultrasonidos y el basado en luz, se tomará una serie de 45 medidas para cada sensor para dos objetos muy distintos: **una caja de cartón marrón estándar y una esponja**.

En resumen, **se dispondrá de 6 series de medidas**, puesto que se empleará cada uno de los 3 tres sensores para medir distancias a 2 objetos, y cada serie tendrá 45 medidas, porque se tomarán 5 medidas para cada una de las 9 distancias de referencia.

2.2. Programación del Arduino Uno

Para facilitar el proceso de medición y el análisis posterior de las medidas, se realizará un programa de Arduino para cada sensor que automatice en lo posible el proceso. Realmente el programa de Arduino puede ser prácticamente igual para los tres sensores, y diferencias dependientes del sensor pueden limitarse a las siguientes: pines utilizados, forma de leer la señal que entrega el sensor (mediante una entrada analógica y el ADC, o determinado la duración de un pulso en una entrada digital), y función de calibración usada para calcular la distancia.

Además del sensor, **se conectará un pulsador al pin 4 del Arduino Uno para solicitar el inicio de una medición**. Y durante el proceso de medición, **el Arduino Uno enviará información a un ordenador mediante la conexión serie-USB**. Esta información consistirá **en líneas de texto con formato CSV** (*Comma- Separated Values*), que se podrá visualizar en un monitor serie (por ejemplo, el del IDE de Arduino), y posteriormente copiar y guardar en un archivo de texto CSV.

El programa de Arduino debe funcionar conforme al siguiente proceso:

1. En el programa se debe declarar un vector constante con las 9 distancias de referencia ordenadas y expresadas en milímetros.
2. Al arrancar, el Arduino Uno debe enviar una cadena de texto con una línea de cabecera que indica los nombres de los datos que enviará después, según el formato que se describe en el siguiente apartado. Además, iniciará un índice para el vector de distancias de referencia al primer valor del mismo, correspondiente a 5cm.
3. El Arduino quedará esperando a que se actúe sobre el pulsador. Antes de actuar sobre el pulsador, el equipo de alumnos deberá colocar el objeto a la distancia de referencia actual (inicialmente 5cm) usando la cinta métrica, como se indica en la Figura 13.
4. Cuando se actúe en el pulsador, el Arduino realizará una 5 medidas para la distancia de referencia actual, con un tiempo entre medidas de 500ms. Justo después de cada medida, el Arduino calculará el valor que proporciona el sensor (tiempo, corriente o tensión), bien con la función “analogRead” o bien con la función “pulseIn”, y la distancia medida usando la función de calibración o una tabla de *look-up*, según corresponda. Además, con cada medida calculada, el Arduino enviará una cadena de texto con una línea que contendrá la información sobre la medida, según el formato que se describe en el siguiente apartado.
5. Tras las 5 medidas, el Arduino incrementará el índice para el vector de distancias de referencia, y el proceso se repetirá desde el paso 3, hasta completar las 9 distancias de referencia.
6. Después de completar las medidas para las 9 distancias de referencia, el Arduino enviará de nuevo la línea de cabecera, establecerá el índice para el vector de distancias de referencia al primer valor, y repetirá el proceso desde el paso 3.

Con el programa anterior, el Arduino irá capturando las medidas de una serie, para un sensor y objeto determinados, en sincronía con el posicionamiento del objeto a las distancias de referencia por parte de los alumnos, y además el Arduino irá enviando las medidas con el formato adecuado al ordenador.

2.3. Formato de los datos proporcionados por el Arduino Uno

Como se describe en el apartado anterior, el programa de Arduino irá enviando al monitor serie la información de las medidas durante el proceso de medición, con un formato determinado adecuado para ser copiado y guardado como un archivo CSV.

Cada serie de medidas comenzará con una **línea de encabezado con los nombres de los valores** que tendrán las líneas a continuación, y que dependerá del sensor con que se obtienen las medidas:

- Sensor de ultrasonidos HC-SR04: "Num, DReferencia, Pulso, DMedida".
- Sensor industrial de ultrasonidos UK6C/H2-0EUL: "Num, DReferencia, Corriente, DMedida".
- Sensor óptico GP2Y0A21YK0F: "Num, DReferencia, Tension, DMedida".

Después de la línea de encabezado, se dispondrá de **45 líneas con información de las 45 medidas de la serie**. Cada línea debe contener cuatro valores numéricos enteros separados por comas:

1. **Num**: Número de línea, de 1 a 90.
2. **DReferencia**: Distancia de referencia a la que se coloca el objeto desde el sensor, usando la cinta métrica disponible, expresada en milímetros sin decimales.
3. El tercer valor depende del sensor, y puede ser uno de los tres siguientes:
   - **Pulso**, para el sensor de ultrasonidos HC-SR04: Duración del pulso de la salida "echo" del sensor, que corresponde con el tiempo de vuelo de la onda de ultrasonidos, expresado en microsegundos, sin decimales. Debe ser 0 si el pulso está fuera del rango de 117µs a 23,324ms que corresponde al rango de distancias del sensor: 2cm a 40cm
   - **Corriente**, para el sensor industrial de ultrasonidos UK6C/H2-0EUL: Valor de corriente que devuelve proporciona el sensor, expresado en microamperios, y que se obtiene a partir del dato devuelto por la función "analogRead()" de Arduino. Debe ser 0 si el valor de corriente está fuera del rango de 4mA a 20mA que corresponde al rango de distancias configurado para el sensor: 10cm a 70cm.
   - **Tensión**, con el sensor óptico GP2Y0A21YK0F: Valor de tensión obtenido del dato devuelto por la función "analogRead()" de Arduino, que corresponde con la salida tensión del sensor, expresado en milivoltios sin decimales. Debe ser 0 si el valor de tensión está fuera del rango de 2,25V a 0,4V que corresponde al rango de distancias del sensor: 10cm a 80cm.
7. **DMedida**: Distancia calculada a partir del tercer valor (tiempo, corriente o tensión, según el sensor) usando la función de calibración del sensor (u opcionalmente una tabla *look-up* para el sensor óptico), expresada en milímetros sin decimales. Debe ser 0 si el tercer valor es cero.

Después de que se haya acabado un proceso de medición con un sensor y un objeto dados, todas las líneas de texto que ha envía el Arduino al monitor serie se copiaran y guardarán en un archivo de datos CSV con un nombre de acuerdo a la siguiente tabla:

	Ultrasonidos HC-SC04	Ultrasonidos UK6C/H2-0EUL	Óptico GP2Y0A21YK0F
Caja de cartón	E*-HCSR04-caja-E*.csv	E*-UK6C-caja.csv	E*-GP2YA0-caja.csv
Esponja	E*-HCSR04-esponja-E*.csv	E*-UK6C-esponja.csv	E*-GP2YA0-esponja.csv

El asterisco en los nombres de los archivos es un **número asociado al equipo de alumnos**.

El Archivo CVS se puede abrir fácilmente en una aplicación de hojas de cálculo, como puede ser Microsoft Excel, para elaborar información estadística de las medidas de distancia con objetivo de determinar propiedades del sensor como su repetitividad o exactitud, y así poder comparar el comportamiento de los sensores entre sí, para los dos objetos considerados.

3. Experimentos a realizar

3.1. Análisis del sensor HC-SR04

Hay que llevar a cabo las siguientes tareas utilizando el sensor económico de ultrasonidos HC-SR04.

1. Representar en un esquema eléctrico el circuito que incluye la conexión del sensor con el Arduino Uno, así como las fuentes de alimentación necesarias. Después realizar el montaje práctico, y verificar bien que es correcto antes de alimentar el circuito y el Arduino. Hacer una fotografía del montaje.
2. Implementar un programa para el Arduino Uno que ayude en el proceso de adquisición de las dos series de 45 medidas de distancia para cada uno de los dos objetos considerados (caja de cartón y esponja). Hay que seguir las instrucciones de la sección 2 de este manual, considerando que el sensor HC-SR04 proporciona un **pulso cuya duración es el tiempo de vuelo** de la onda de sonido en microsegundos.
3. Obtener cada serie de medidas de distancia en una aplicación de monitor serie en el ordenador, y guardarla como un archivo CSV. Importar estos archivos en una en una aplicación de hojas de cálculo, como por ejemplo Microsoft Excel. Cada archivo se puede importar en una hoja de la aplicación.
4. Para cada una de las mediciones en el rango de 10cm a 70cm de cada serie (35 medidas), hay que calcular el error entre la distancia de referencia y la distancia medida por el sensor (error = DReferencia – DMedida). Después, hay que considerar los 35 valores de error de cada serie para calcular los valores máximo, mínimo y de media aritmética (o promedio) del error de esa serie. Estos valores informan de la exactitud del sensor. Se puede trabajar en milímetros, que es la unidad usada por el programa de Arduino formatear los datos.
5. Para cada grupo de 5 distancias medidas por el sensor en cada una de las 7 distancias de referencia en el rango 10cm a 70cm, hay que determinar el valor de desviación estándar o típica. En este cálculo hay que descartar las distancias medidas por el sensor que no puedan considerarse válidas. Después, para cada serie, hay que calcular el valor promedio de las desviaciones calculadas antes, el cual representa la repetitividad del sensor.

   Para calcular las desviaciones estándar se debe usar la siguiente expresión, en la cual N=5, X_i son las 5 distancias medidas por el sensor para una misma distancia de referencia, y X_M es la media aritmética (o promedio) de esas 5 medidas.

$$DE = \sqrt{\frac{\sum_{i=1}^{i=N}(X_i - X_M)^2}{N}}$$

6. Analizar las series de medidas de distancia para comprobar si el sensor ofrece algún comportamiento especial para distancias inferiores a 10cm y distancias superiores a 70cm, de forma que se pudiera descartar fácilmente las medidas fuera de ese rango.

3.2. Análisis del sensor UK6C

Hay que repetir las 5 tareas de la actividad 1 con el sensor industrial de ultrasonidos UK6C/H2-0EUL, teniendo en cuenta estas dos particularidades:

- Antes de abordar el punto 2, hay que **programar el sensor para que su salida analógica se adapte al rango de medida de 10 a 70 cm**, siguiendo el proceso descripto en el apartado 1.1.3.
- Hay que implementar la **función inversa de la curva de calibración específica del sensor** UK6C/H2-0EUL en el programa de Arduino, y tener en cuenta que este sensor proporciona una corriente que se convierte a tensión, según se explica en el apartado 1.1.2.

3.3. Análisis del sensor GP2Y0A21YK0F

Hay que repetir las 5 tareas de la actividad 1 con el sensor óptico GP2Y0A21YK0F, con la particularidad de que, en el programa de Arduino, hay que implementar la **función inversa de la curva de calibración específica del sensor**, siendo necesario elegir **una de las siguientes opciones** para programar dicha función, conforme se describe en el apartado 1.2.2:

- Una función matemática que se ajuste bien a la curva de medidas reales.
- Una tabla *look-up* y una función para interpolar valores intermedios. En este caso hay definir una tabla que tenga entre 12 y 16 entradas para las distancias entre 10cm y 80 cm.

3.4. Comparación de resultados

Se plantean dos tareas para comparar entre sí los tres sensores estudiados:

7. Resumir los errores máximo, mínimo y medio para las distancias (en mm), así como la desviación estándar media, para los tres sensores y para los dos objetos (caja de cartón y esponja). Se puede usar una tabla con 6 filas (3 sensores x 2 objetos) y 4 columnas (errores y desviación estándar). En base a los datos de la tabla anterior, indicar qué sensor tiene un comportamiento mejor para cada uno de los dos objetos, y por lo tanto sería más adecuado para medir distancias a objetos de ese tipo.
8. Para cada sensor, describir al menos dos ventajas y dos inconvenientes en su uso.

3.5. Tareas Adionales

Se plantean las tareas opcionales, que permitirán obtener una puntuación adicional en la evaluación de la práctica que sirva para compensar puntuaciones bajas en otras tareas.

9. Para cada sensor, determinar experimentalmente la distancia más pequeña a un objeto en frente del sensor a la que se pueden realizar medidas con una exactitud razonable.
10. Añadir un filtro digital en los programas de Arduino diseñados en las actividades para eliminar ruidos en las medidas.

4. Evaluación

Cada equipo entregará a través de UACloud un archivo comprimido en formato ZIP que contendrá los siguientes archivos y documentos:

- Para cada uno de los tres sensores estudiados se incluirá una carpeta que incluya:
  - **Tarea 2**: **Código fuente del programa de Arduino Uno** correctamente comentado y previamente probado. Para que el programa se evalúe, debe compilar sin errores. Si se ha incluido un filtro digital en el programa de Arduino, según se propone en las tareas adicionales, hay que indicar claramente donde se encuentra el filtro dentro del código fuente.
  - **Tarea 3**: **Archivos CSV para las dos series** de medidas de distancias correspondientes a los dos objetos (caja de cartón y esponja).
  - **Tareas 4 y 5**: **Archivo de hojas de cálculo** en formato “XLS” o “XLSX”, que contendrá dos hojas, cada una con la serie de 45 medidas de cada objeto obtenidas del archivo CSV (Num, DReferencia, Pulso/Corriente/Tension, DMedida) y los cálculos estadísticos del error para el rango de 10cm a 70cm (errores de las distancias medidas; máximos, mínimos y promedios del error de cada serie, y desviaciones estándar).
- Un breve documento, en formato PDF, con 6 páginas o caras de máximo, que incluya estos apartados:
  - **Esquemas de la conexión (tarea 1)**. Hay que incluir el esquema eléctrico completo del circuito con el sensor y el Arduino Uno, y aportar una fotografía del montaje práctico.

- **Comportamientos especiales (tarea 6)**. Para cada sensor, hay que describir si el sensor ofrece algún comportamiento especial para distancias fuera del rango de 10cm a 70cm.
- **Tabla comparativa (tareas 7)**. Tabla que compara los errores máximo, mínimo y medio, y la desviación estándar media, de los tres sensores para el rango de 10 a 70cm. Indicar que sensor tiene un comportamiento mejor.
- **Ventajas e inconvenientes (tarea 8)**. Dos ventajas y dos inconvenientes de cada sensor.
- **Distancia mínima (tarea opcional 9)**. Indicar la distancia más pequeña a la que cada uno de los sensores da medidas con una exactitud razonable.
- **Filtros digitales (tarea opcional 10)**. Describir brevemente el tipo de filtro digital empleado en cada programa (puede ser el mismo o diferente.

Para la evaluación se considerarán estas puntuaciones para las distintas tareas:

- Tareas 2 y 3 (programas de Arduino para los 3 sensores): hasta 4,5 puntos.
- Tareas 4 y 5 (hojas de cálculo para los 3 sensores): hasta 2,4 puntos.
- Tareas 1, 6, 7 y 8 (en el informe): hasta 3 puntos.
- Tareas opcionales 8 a 10 (en el informe): Pueden añadir hasta 2 puntos a la nota obtenida con las tareas anteriores, hasta sumar 10 puntos en el total de la práctica.