

Experiment #6 Ultrasonic Sensors

In experiment #6 we will deviate from motion development to introduce the ultrasonic sensor interface. The SR04 sensors we use have four electrical connections, one for power, one for ground, one for a trigger to transmit the ultrasonic pulse, and one to detect the echo or ultrasonic return signal. The power and ground lines are hardwired to the robot power system. The “trigger” and “echo” pins are wired to the Arduino pins. The club robot uses pins 22 through 29 to support four sensors. Variables for these pins are defined globally.

There are several ways to interface to the sensors. The method we use is the “NewPing” library. First a NewPing object is created which links the pin numbers of the sensor to the object. Then the object can call the various ping and conversion methods in the library. If your Arduino IDE does not contain the NewPing.h library it can be downloaded from the site listed below.

Global Declarations

1. `#include <NewPing.h>`
2. `#define Trigger_Sensor1 22`
3. `#define Echo_Sensor1 23`
4. `#define Trigger_Sensor2 24`
5. `#define Echo_Sensor2 25`
6. `#define Trigger_Sensor3 26`
7. `#define Echo_Sensor3 27`
8. `#define Trigger_Sensor4 28`
9. `#define Echo_Sensor4 29`
10. `#define max_distance 200 // max operating distance in cm`

The Trigger definitions set the pin numbers for the triggers of the Ultrasonic sensor. The Echo definitions set the pin numbers for the echo pins which interface to the sensors

Methods – NewPing.h

1. Constructor - the constructor creates an object that can call methods.
 - a. `NewPing sonar(trigger_pin, echo_pin, max_distance) // should be global object`
2. Calls – the object calls the methods
 - a. `sonar.ping(); // returns echo time in microseconds`
 - b. `sonar.ping_in(); // returns echo distance in inches`
 - c. `sonar.ping_cm(); // returns echo distance in centimeters`
 - d. `sonar.ping_median(count); // returns median value of count number of samples in microseconds`
 - e. Other calls available see documentation
 - i. Google search “newping.h”
 - ii. <http://playground.arduino.cc/Code/NewPing>

Serial Monitor

The Arduino interface provides a serial monitor which can be used to display data and text in real-time. To initiate the serial monitor you must begin the serial process in the setup function before using it. To use the serial monitor you must run the robot while it is still connected to your computer.

Setup{

Serial.begin(9600); // 9600 is the baud rate of the serial monitor interface

Function calls

1. Serial.print(" literal" or variable); // sends printable string to monitor
2. Serial.println("literal" or variable); // send string and CR/LF to monitor "newline"

Calibration

The speed of sound at sea level is 13,511 inches per second. Maximum resolution from the sensor is 1 microsecond. The time returned from the sensor is converted to distance in inches by multiplying by 6.756×10^{-3} . i.e. Distance in inches = Distance in microseconds $\times 6.756 \times 10^{-3}$

Procedure

1. Download the Ultrasonic_Sensor.ino file and open it in the Arduino IDE
2. Compile and download the program onto the robot.
3. The file should begin running as soon as it is loaded so click on the serial monitor box
4. Observe the output and place your hands in front of each sensor
 - a. Identify which sensors are attached to which sets of pins.
5. Place an object in front of one of the sensors
6. Edit the sample code to switch from centimeter readings to microsecond readings
7. Compile, load and run.
8. Observe and note the individual variations (noise) at the microsecond level
9. Edit the sample code to use the ping_median(5) methods
10. Compile, load, and run
11. Observe and note the individual variations in the reading
 - a. Is the noise reduced?
12. With the program running
 - a. Slowly move the object closer to the robot
 - b. Observe what happens when the object is closer than minimum distance
 - c. Determine the minimum detectable distance
 - d. Remove the object and the point the robot in an open direction
 - e. Determine the sensor response when no object is present
13. Edit the code to perform the microsecond to distance conversion
 - a. Display the distance measured by each sensor in inches
 - b. Place the robot on the calibration board next to a wall section
 - c. Verify the displayed values with a ruler.