Abstract: The purpose this application note is to educate individuals on the procedure to choose, set up, and implement object avoidance technology. The Devantech SRF05 Ultrasonic range finder can be implemented in conjunction with an Arduino UNO to assist in obstacle avoidance and tracking. Multiple sensors can be used in combination to create the desired range of detection.
# Table of Contents

- **Introduction** .................................................................................................................. 3  
- **Key Terms** .......................................................................................................................... 3  
- **Implementation** .................................................................................................................. 4  
  - Sensor Selection .................................................................................................................. 4  
  - Sensor Specification ........................................................................................................... 4  
  - Cost ...................................................................................................................................... 4  
- **Technical Specifications** .................................................................................................. 5  
- **Installation** ......................................................................................................................... 6  
  - Mode 1 ................................................................................................................................. 6  
  - Mode 2 ................................................................................................................................. 7  
- **Programming** ..................................................................................................................... 8  
- **References** .......................................................................................................................... 13
Introduction

When creating an autonomous target tracking robot, one of the most crucial aspects is obstacle detection and avoidance. Every so often, a target may move in such a way that there is an object between the target and the robot. A sensor must be able to detect the object, with enough of a range to allow the robot to respond and move accordingly. Ideally, the sensor must be small, low in cost, and easy to manufacture and use on a large scale. A readily available sensor that fits all of these requirements is the Devantech SRF05 Ultrasonic sensor. (Figure 1)

![Devantech SRF05 Ultrasonic Sensor](http://www.robot-electronics.co.uk/htm/srf05tech.htm)

Figure 1: Devantech SRF05 Ultrasonic Sensor
http://www.robot-electronics.co.uk/htm/srf05tech.htm

Key Terms

**Ultrasonic**: utilizing waves that have a frequency above the human ear's audibility limit (20,000Hz).

**Arduino UNO**: microcontroller produced by Arduino that contains 14 I/O pins, and has 5V and 3.3V output ports.
Implementation

Sensor Selection

When determining the distance between two objects, there are a variety of sensors to choose from. The most common are laser, radar, ultrasonic and infrared range finding sensors. Design teams must carefully analyze the situation at hand, and figure out what specifications must be met to complete the task. These sensors will all work in one form or another, but most of them deal with situations where the target is in a known location or at a predefined distance. In this case however, the object could be anywhere within the 360 degrees of vision. This means that there needs to be a wide detection range, which can be implemented by using an ultrasonic sensor. Another important aspect of any design is cost. The goal is to buy an efficient, low cost sensor that is capable of being used with the desired system. In this instance, the microcontroller being used is an Arduino UNO. (Figure 2) Research must be done to find the best sensor that meets all of the design and cost specifications. In this case, the ultrasonic sensor chosen was the Devantech SRF05. It has a relatively low cost of $27.95 each.

Figure 2: Arduino UNO microcontroller
http://arduino.cc/en/Main/arduinoBoardUno
**Technical Specifications**

The SRF05 has an object detection range of 3cm to 4m. The ultrasonic sensor sends out an initial 10µs pulse and measures the time until the return signal is detected after it bounces off of an object. The signal waveform path is shown below in **Figure 3**. This provides between 30 and 45 degrees of detection. The sensor consists of 5 connection pins and 5 programming pins. The connection pins are a 5V supply, trigger input, echo return output, ground, and one mode pin. The mode pin will change depending on how the sensor has been wired up to the microcontroller.

![Figure 3: Sensor Wave-path](http://www.robot-electronics.co.uk/htm/srf05tech.htm)
Installation

There are two different ways to connect a SRF05 sensor to a microcontroller. Both will work for all applications, it is up to the user to decide which one to use depending on how many sensors and other devices will be used together.

Mode 1

The first way is known as Mode 1, and the connection pins are shown below in Figure 4, and should be connected to the microcontroller accordingly. In this mode, the trigger and echo pins are separate, and will take up 2 I/O pins on the microcontroller. This is correct, but availability issues may arise when many sensors are used together. In this mode, a total of 7 sensors could be implemented on the Arduino UNO. When the sensor is connected correctly, a yellow LED will flash to indicate that the wire connections are correct.

![Mode 1 pin connections](http://www.robot-electronics.co.uk/htm/srf05tech.htm)

**Figure 4: Mode 1 pin connections**
http://www.robot-electronics.co.uk/htm/srf05tech.htm
Mode 2

Mode 2 is different from Mode 1 in that it combines the trigger and echo pins into 1 I/O pin. This is a space saving measure, and will allow for up to 14 sensors to be used on one microcontroller. In this mode, the trigger pin is the one that is hooked into the microcontroller, and the echo pin is connected in parallel with GND. The final wire diagram should be similar to that shown in Figure 5 below. Note that any of the 14 I/O pins can be used to connect the trigger pin to the microcontroller. Figure 6 shows the difference between the pins of Mode 1 found in Figure 4.

![Figure 5: Mode 2 Wire Diagram](http://communityofrobots.com/tutorial/kawal/srf05-ultrasonic-sensor-and-arduino)
Programming

Programming the microcontroller is the most important aspect of this process. Without the correct code, the sensors will not work. The Arduino Programming language is based on C/C++. The first step is to download the Arduino 1.0.4 software that will allow for communication between the computer and microcontroller. The ultrasonic sensor is considered to be a PING sensor, and the Arduino software actually has a demo program that can be altered to implement multiple sensors. This program will specify that a signal is sent out and returned, and convert the time it took into a physical length in both centimeters and inches. These values are then displayed on a serial monitor screen. **Figures 7 and 8** show the different timing diagrams for both Modes of operation, and how the signal is sent out and received.
Figure 7: SRF05 Timing Diagram Mode 1
http://www.robot-electronics.co.uk/htm/srf05tech.htm

Figure 8: SRF05 Timing Diagram Mode 2
http://www.robot-electronics.co.uk/htm/srf05tech.htm
Here is the basic Arduino Code that can be used to implement one ultrasonic sensor:

```cpp
/* Ping Sensor

This sketch reads a PING ultrasonic rangefinder and returns the
distance to the closest object in range. To do this, it sends a pulse
to the sensor to initiate a reading, then listens for a pulse
to return. The length of the returning pulse is proportional to
the distance of the object from the sensor.

The circuit:
* +V connection of the PING attached to +5V
* GND connection of the PING attached to ground
* SIG8 connection of the PING attached to digital pin 4
*/

// this constant won't change. It's the pin number of the sensor's output:
const int pingPin4 = 4;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration4, inches4, cm4;

  //Pin 4 Ultrasonic Sensor

  // 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(pingPin4, OUTPUT);
  digitalWrite(pingPin4, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin4, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin4, 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(pingPin4, INPUT);
  duration4 = pulseIn(pingPin4, HIGH);

  // convert the time into a distance
  inches4 = microsecondsToInches(duration4);
  cm4 = microsecondsToCentimeters(duration4);
```

```
Serial.print("Pin Sensor 4: ");
Serial.print(inches4);
Serial.print(" in, ");
Serial.print(cm4);
Serial.println();

delay(200);

// 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.
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;
}

This code initialized the variables, and then measures the time that the signal takes to return. This time
distance can then be converted into physical lengths in both inches and centimeters. When the serial
monitor is run, the results are displayed. These measurements are the distance from the robot to the
object. See Figure 9.
Figure 9: Serial Monitor Displaying Measurements
References


   <http://www.robot-electronics.co.uk/htm/srf05tech.htm>.