Contents

ROV’s empty

Classification-Pursposes empty

Essential Parts empty

Sonar empty

Sound properties missing figure

Ultrasonic Sensors empty

HC-SR04 vs JSN-SR04T empty

JSN-SR04T Two Modules’ Integration on Arduino Uno  
missing figure

Submersible vs Immersible Motors

GUI – Processing IDE

ROV’s empty

Classification-Pursposes empty

Essential Parts empty

Sonar empty

Sound properties missing figure  
luanthesis $13/78pdf

Why use sonar for underwater operations and not radar? Unlike sound, light

and radio waves are types of electromagnetic waves. They can propagate

very well in a vacuum, but in mediums with high conductivity like seawater,

these signals experience severe attenuation and scattering. Acoustic waves,

on the other hand, require a medium and cannot exist in a vacuum. This is

because sound is actually a pressure wave that is created through the vibration

of material in a medium. In water, acoustic signals travel as compressional

waves that propagate longitudinally at the local speed of sound, independent of

the speci¯c characteristics of the signal such as frequency and waveform. The

local speed of sound, however, can vary depending on the water temperature,

pressure, and salinity.

A simplified formula for the speed of sound relating these quantities

was proposed in 1969 by CC Leroy [15].

c = 1492.9 + 3\*(T - 10) - 6 \* [10^(-3)]\*(T - 10)^2 -4 \* [10^(-2)]\*(T - 18)^2

+1.2\*(S - 35) – [10^(-2)]\*(T - 18)\*(S - 35) + (Z/61) (2.1)

Here, pressure is expressed as depth Z in meters. This equation yields a speed

accurate to 0.1m/s for a temperature less than 20\*C and in depths less than

8000m. Figure 2.2 displays how the speed of sound changes with temperature

and salinity. The values of salinity shown in the graph range from 29-45 ppt.

In Figure 2.3 depth and temperature are the variables, with depth ranging

from 0-8000 m. missing figure

also check :

<https://en.wikipedia.org/wiki/Speed_of_sound>

<http://hyperphysics.phy-astr.gsu.edu/hbase/tables/soundv.html#c1>

<http://hyperphysics.phy-astr.gsu.edu/hbase/sound/souspe2.html>

<http://www.thermaxxjackets.com/newton-laplace-equation-sound-velocity/>

<https://www.herrmannultrasonics.com/en-us/what-is-ultrasonic/>

Ultrasonic Sensors empty

HC-SR04 vs JSN-SR04T empty

JSN-SR04T Two Modules’ Integration on Arduino Uno  
missing figure

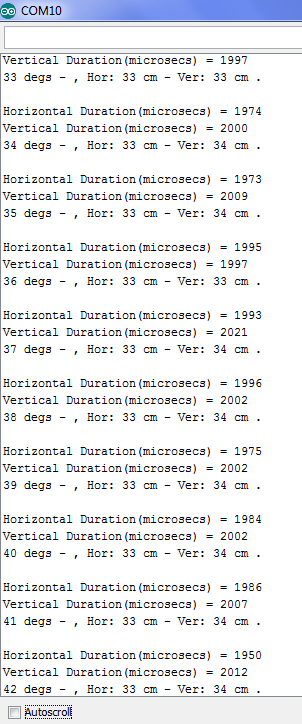
Integration of two JSN-SR04T modules was accomplished successfully on the Arduino Uno Board for scanning in two axes.

They are designed to operate fixed on servo motors, one scanning horizontally and the second one vertically.

NOTE: *For horizontal scanning the trigger pin is connected in digital pin 10 in Arduino while the echo pin in 11.  
For vertical scanning the trigger pin is connected in digital pin 5 in Arduino while the echo pin in 6.*  
  
It is needed to be taken under consideration that they have to be close enough to each other in order to achieve that the starting scanning coordinates are the same.

The result is totally satisfactory as the maximum proximity which allows accurate measurements is 3cm. This is acceptable because the scanning range is 18-20 meters  
which means that there is a 0.15% error.

The next screenshot (from Arduino Serial Monitor) shows measurements taken from these two , positioned side by side in 3 cm distance , targeting directly on a flat surface .



Submersible vs Immersible Motors

<http://www.pumpsandsystems.com/motors/august-2014-differences-between-submersible-immersible-motors?page=2>

Many submerged motors rely on the effluent to stay cool and run continuously while submerged.

Immersible motors are specifically designed to be operated in a dry environment that might flood under unusual circumstances.

Conclusion

Application conditions should be the deciding factor between immersible and submersible motors. Immersible designs are more cost effective, but the submersible design is a better fit in an application that requires continual submergence.

GUI – Processing IDE

https://processing.org/reference/environment/

The Processing Integrated Development Environment (IDE) includes a text editor, a compiler, and a display window. It enables the creation of software within a carefully designed set of constraints.

Overview

The Processing Development Environment (PDE) makes it easy to write Processing programs. Programs are written in the Text Editor and started by pressing the Run button. In Processing, a computer program is called a sketch. Sketches are stored in the Sketchbook, which is a folder on your computer.

Sketches can draw two- and three-dimensional graphics. The default renderer is for drawing two-dimensional graphics. The P3D renderer makes it possible to draw three-dimensional graphics, which includes controlling the camera, lighting, and materials. The P2D renderer is a fast, but less accurate renderer for drawing two-dimensional graphics. Both the P2D and P3D renderers are accelerated if your computer has an OpenGL compatible graphics card.

The capabilities of Processing are extended with Libraries and Tools. Libraries make it possible for sketches to do things beyond the core Processing code. There are hundreds of libraries contributed by the Processing community that can be added to your sketches to enable new things like playing sounds, doing computer vision, and working with advanced 3D geometry. Tools extend the PDE to help make creating sketches easier by providing interfaces for tasks like selecting colors.

Processing has different programming modes to make it possible to deploy sketches on different platforms and program in different ways. The Java mode is the default. Other programming modes may be downloaded by selecting "Add Mode..." from the menu in the upper-right corner of the PDE.