



## Section 5: Underwater Robots for Volume 2: How Do We Explore?



NOAA Ship *Okeanos Explorer*: America's Ship for Ocean Exploration.  
Image credit: NOAA. For more information, see the following  
Web site:  
<http://oceanexplorer.noaa.gov/okeanos/welcome.html>

### Through Robot Eyes

(adapted from the 2010 Gulf of Mexico Deep-sea Ecosystems Education Materials Collection)

#### Focus

Image analysis

#### Grade Level

9-12 (Physical Science/Technology)

#### Focus Question

How can images obtained by underwater robots be used to assist ocean exploration aboard the *Okeanos Explorer*?

#### Learning Objectives

- Students will describe typical applications and limitations of imagery obtained with remotely operated vehicles (ROVs).
- Students will demonstrate how lasers may be used to calibrate images for size and distance measurements.
- Students will analyze ROV imagery from the *Okeanos Explorer* to make inferences about deep-ocean habitats and organisms.

#### Materials

- Copies of the *Robot Imagery Inquiry Guide*, one for each student group
- Copies of *ImageJ Basics*, one for each student group; see Learning Procedure, Step 1c
- Ruler, one for each student group

#### Audio Visual Materials

- (Optional) Video projector or other equipment to display downloaded images (see Learning Procedure, Step 1d)

#### Teaching Time

One or two 45-minute class periods, plus time for students to complete *Inquiry Guide*

#### Seating Arrangement

Groups of two to four students

#### Maximum Number of Students

30

#### Key Words and Concepts

Ocean Exploration  
*Okeanos Explorer*



Robot  
Imagery  
Video  
ImageJ

## Background Information

*NOTE: Explanations and procedures in this lesson are written at a level appropriate to professional educators. In presenting and discussing this material with students, educators may need to adapt the language and instructional approach to styles that are best suited to specific student groups.*

On August 13, 2008, the NOAA Ship *Okeanos Explorer* was commissioned as “America’s Ship for Ocean Exploration;” the only U.S. ship whose sole assignment is to systematically explore Earth’s largely unknown ocean. The strategy for accomplishing this mission is to use state-of-the-art technologies to search the ocean for anomalies; things that are unusual and unexpected. When an anomaly is found, the exploration strategy shifts to obtaining more detailed information about the anomaly and the surrounding area. An important concept underlying this strategy is the distinction between exploration and research. As a ship of discovery, the role of *Okeanos Explorer* is to locate new features in the deep ocean, and conduct preliminary investigations that provide enough data to justify follow-up by future expeditions.

The *Okeanos Explorer* strategy involves three major activities:

- Underway reconnaissance;
- Water column exploration; and
- Site characterization.

Underway reconnaissance involves mapping the ocean floor and water column while the ship is underway, and using other sensors to measure chemical and physical properties of seawater. Water column exploration involves making measurements of chemical and physical properties “from top to bottom” while the ship is stopped. In some cases these measurements may be made routinely at pre-selected locations, while in other cases they may be made to decide whether an area with suspected anomalies should be more thoroughly investigated. Site characterization involves more detailed exploration of a specific region, including obtaining high quality imagery, making measurements of chemical and physical seawater properties, and obtaining appropriate samples.

In addition to state-of-the-art navigation and ship operation equipment, this strategy depends upon four types of technology:

- Telepresence;
- Multibeam sonar mapping;
- CTD (an instrument that measures conductivity, temperature, and depth) and other electronic sensors to measure chemical and physical seawater properties; and
- A Remotely Operated Vehicle (ROV) capable of obtaining high-quality imagery and samples in depths as great as 4,000 meters.

In the summer of 2010, years of planning, field trials, and state-of-the-art technology came together for the first time on the ship’s maiden voyage as part of the INDEX-SATAL 2010 Expedition. This expedition was an international collaboration between



NOAA Ship *Okeanos Explorer*: America’s Ship for Ocean Exploration.  
Image credit: NOAA. For more information, see the following Web site:  
<http://oceanexplorer.noaa.gov/okeanos/welcome.html>

### **Okeanos Explorer Vital Statistics:**

Commissioned: August 13, 2008; Seattle, Washington  
Length: 224 feet  
Breadth: 43 feet  
Draft: 15 feet  
Displacement: 2,298.3 metric tons  
Berthing: 46, including crew and mission support  
Operations: Ship crewed by NOAA Commissioned Officer Corps and civilians through NOAA’s Office of Marine and Aviation Operations (OMAO); Mission equipment operated by NOAA’s Office of Ocean Exploration and Research

For more information, visit <http://oceanexplorer.noaa.gov/okeanos/welcome.html>.

Follow voyages of America’s ship for ocean exploration with the *Okeanos Explorer* Atlas at [http://www.ncddc.noaa.gov/website/google\\_maps/OkeanosExplorer/mapsOkeanos.htm](http://www.ncddc.noaa.gov/website/google_maps/OkeanosExplorer/mapsOkeanos.htm)

scientists from the United States and Indonesia to explore the deep ocean in the Sangihe Talaud Region. This region is located in the ‘Coral Triangle’, which is the global heart of shallow-water marine biodiversity. A major objective of the expedition was to advance our understanding of undersea ecosystems, particularly those associated with submarine volcanoes and hydrothermal vents. A major objective of the INDEX SATAL 2010 expedition was to locate submarine volcanoes, hydrothermal vents, chemosynthetic ecosystems, and seamounts associated with active geologic processes in Indonesia’s deep sea. A key component in the expedition’s quest for anomalies was to look for changes in chemical properties of seawater that can indicate the presence of these features. [For more information about the INDEX-SATAL 2010 Expedition, see <http://oceanexplorer.noaa.gov/okeanos/explorations/10index/welcome.html>.



The site characterization component of the *Okeanos Explorer* exploration strategy depends heavily upon remotely operated vehicles (ROVs). These are unoccupied robots usually linked to an operator aboard a surface ship by a group of cables. Most ROVs are equipped with one or more video cameras and lights, and may also carry other equipment such as a manipulator or cutting arm, water samplers, equipment for collecting samples, and measuring instruments to expand the vehicle’s capabilities for gathering data about the deep-ocean environment.

For the INDEX-SATAL 2010 Expedition, the NOAA Ship *Okeanos Explorer* carried *Little Hercules*, an ROV originally developed by a team of engineers at Dr. Robert Ballard’s Institute for Exploration (IFE) at the University of Rhode Island for the primary purpose of gathering high quality video imagery. Nicknamed “*Little Herc*,” the ROV proved to be well-suited to this purpose on a variety of successful missions for IFE, including providing the first and only images of John Kennedy’s PT Boat, *PT-109*. Eventually, a much larger ROV named “*Hercules*” took over these tasks, and *Little Herc* became part of an exhibit at the Mystic Aquarium. This shore duty came to an end, however, when it became clear that *Okeanos Explorer*’s primary ROV would not be ready in time for the INDEX-SATAL 2010 Expedition. Through a collaboration between IFE and NOAA’s Office of Ocean Exploration and Research, *Little Herc* was brought out of retirement and refitted specifically to meet the expedition’s needs.



*Little Herc* is operated in tandem with a camera platform that carries 2,400 watts of lighting provided by HMI (hydrargyrum medium-arc iodide) arc lamps. This lighting illuminates the total darkness of the deep ocean, helps guide *Little Hercules*, and provides lighting for the high-definition video images of the ROV at work. The camera platform is named *Seirios*, after the name of the brightest star in the night sky (also called the Dog Star, sometimes spelled “Sirius”). *Little Herc* is attached to *Seirios* by a 30-m cable called the Remotely Operated Vehicle Tether, while the camera platform is attached to the *Okeanos Explorer*’s traction winch by a 17 mm Oceanographic Instrumentation and Control Cable which has an armored outer jacket with 3 power conductors and 3 optical fibers for transmitting data and control signals.

A variety of sensors are aboard the ROV for navigation and data collection. These include depth and altitude sensors, an Ultra Short Baseline Tracking System, full color imaging sonar, and a Seabird SBE 49 FastCAT CTD. Video equipment includes two Insite Pacific single CCD (charge-coupled device) high-resolution miniature color video cameras, one Insite Pacific triple CCD high-definition Zeus Plus video camera, two Deep Sea Power and Light 250-watt LED matrix lights, and two Deep Sea Power and Light 400-watt HMI arc lamps. For additional details about *Little Herc*, see the sidebar, The *Little Hercules* Remotely Operated Vehicle. For more information about other ROVs, visit <http://oceanexplorer.noaa.gov/technology/subs/rov/rov.html>.

This lesson introduces students to remotely operated vehicles and video imagery as they are used for ocean exploration aboard the *Okeanos Explorer*. In the future, students will have access to additional video imagery collected by ROVs as the *Okeanos Explorer* continues its voyages of discovery in Earth’s deep ocean.

## Learning Procedure

1. To prepare for this lesson:
  - a) Review:

- Introductory essays for the INDEX-SATAL 2010 Expedition (<http://oceanexplorer.noaa.gov/okeanos/explorations/10index/welcome.html>); including *Little Hercules* ROV (<http://oceanexplorer.noaa.gov/okeanos/explorations/10index/background/rov/rov.html>);
- Daily log entries for
  - July 7 (<http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/july07/july07.html>);
  - July 24 (<http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/july24/july24.html>); and
  - August 6 (<http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/aug06/aug06.html>).

You may want to assign one or more of these essays as background reading prior to beginning the rest of the lesson.

- (b) Review background information about the *Okeanos Explorer* exploration strategy and technologies.
- (c) Review procedures on the *Robot Imagery Inquiry Guide*. If students will be using classroom computers, download ImageJ and the *ImageJ Basics* handout from <http://rsbweb.nih.gov/ij/download.html> and <http://rsbweb.nih.gov/ij/docs/pdfs/ImageJ.pdf>, respectively.

## The *Little Hercules* Remotely Operated Vehicle

*Little Hercules* was developed by a team of engineers at Dr. Robert Ballard’s Institute for Exploration (IFE) at the University of Rhode Island. Its primary purpose is to gather high quality video imagery in support of scientific research and ocean exploration. Major systems include:

- **Power** – 2,800 volts (AC) supplied from the surface; stepped down to 120 VAC by a transformer aboard *Little Herc*; further converted in the electronics pressure housing to 24 VDC (8A maximum load) and 12 VDC (0.6A maximum load)
- **Propulsion** – Four Technadyne 1020 electric thrusters; two oriented horizontally to provide forward, backward, and rotational motion, and two mounted to form a V when viewed from the front (vertran configuration) which provides up, down, and lateral movement
- **Onboard Control** – PC104 computer
- **Imaging** – Main Camera: One Insite Pacific triple CCD high-definition Zeus Plus HDTV camera with zoom and macro; Utility Cameras: Two Insite Pacific single CCD high-resolution miniature color video cameras
- **Lighting** – Two Deep Sea Power and Light 250-watt-equivalent LED matrix lights; Two Deep Sea Power and Light 400-watt HMI arc lamps
- **Navigation** – Ultrashort Baseline acoustic transponder (works in concert with ship-based system that calculates the ROV’s underwater position)
- **Sensors** – Paroscientific 8B7000 pressure/depth sensor; Seabird SBE 49 FastCAT CTD; Tritech PA500 altimeter; Tritech Super SeaKing scanning sonar

These components are integrated within an aluminum frame that is supported in water by a flotation package of syntactic foam, which provides slightly positive buoyancy that is trimmed to neutral by the ROV’s vertical thrusters. Most electronics are contained in a 10-inch diameter titanium pressure housing. The ROV is rated to a depth of 4,000 meters, and in air weighs 1,200 pounds.

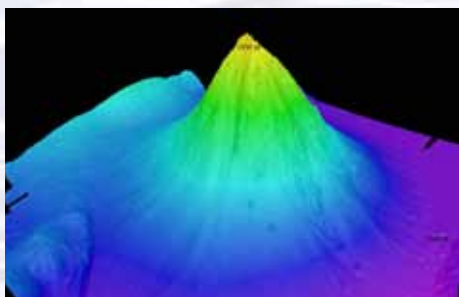
*Little Hercules* is operated in tandem with a camera platform named *Seirios* that is equipped with a HD video camera identical to that on the ROV, as well as six HMI (hydrargyrum medium-arc iodide) arc lamps that provide a total of 2,400 watts of lighting. *Seirios* has no buoyancy module, and is intentionally much heavier than water to provide a buffer between the ROV and surface motion of the ship. *Little Hercules* is attached to *Seirios* by a 30-m cable called the Remotely Operated Vehicle Tether. *Seirios* is attached to the *Okeanos Explorer*’s traction winch by a 17 mm Oceanographic Instrumentation and Control Cable which has an armored outer jacket with 3 power conductors and 3 optical fibers for transmitting data and control signals. A traction winch has large diameter grooved drums that are designed to protect cables from excessive friction and bending under heavy load conditions.

Prior to every dive, the ROV crew reviews multibeam sonar maps of the proposed dive area, and develops a track-line that is the initial path that the ROV will follow. During a dive, the ROV pilots may modify the track-line as they receive requests from scientists aboard the ship and in Exploration Command Centers to obtain video images of certain features and organisms that the ROV encounters during its exploration.





The ROV *Little Hercules* descends through deep water to an undersea volcano in the Celebes Sea to search for hydrothermal vents and associated ecosystems. Image courtesy of NOAA *Okeanos Explorer* Program, INDEX-SATAL 2010  
[http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/hires/1june29\\_hires.jpg](http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/hires/1june29_hires.jpg)



*Okeanos Explorer's* EM302 multibeam sonar mapping system produced this detailed image of the Kawio Barat seamount, which rises around 3800 meters from the seafloor. Image courtesy of NOAA *Okeanos Explorer* Program, INDEX-SATAL 2010  
[http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/hires/june26fig1\\_hires.jpg](http://oceanexplorer.noaa.gov/okeanos/explorations/10index/logs/hires/june26fig1_hires.jpg)



Scientists in the Exploration Command Center at NOAA's Pacific Marine Environmental Laboratory in Seattle view live video from the *Okeanos Explorer's* ROV. Image courtesy NOAA  
<http://www.pmel.noaa.gov/images/headlines/ecc.jpg>



Senior Survey Technician Elaine Stuart holds onto the CTD as it comes aboard the *Okeanos Explorer*. Image courtesy NOAA  
<http://www.moc.noaa.gov/oe/visitor/photos/photospage-b/CAP%20015.jpg>

Also, download the image files EX2010.07.12\_ROV\_01.jpg and EX2010.07.12\_ROV\_02.jpg from

[http://oceanexplorer.noaa.gov/okeanos/edu/resources/media/EX2010.07.12\\_ROV\\_01.jpg](http://oceanexplorer.noaa.gov/okeanos/edu/resources/media/EX2010.07.12_ROV_01.jpg) and [http://oceanexplorer.noaa.gov/okeanos/edu/resources/media/EX2010.07.12\\_ROV\\_02.jpg](http://oceanexplorer.noaa.gov/okeanos/edu/resources/media/EX2010.07.12_ROV_02.jpg), respectively.

If students will not be using classroom computers, ensure that they will have access to these files.

- (d) Download images referenced in Step 2 and 3. You may also want to download examples of imagery from underwater robots from [http://oceanexplorer.noaa.gov/okeanos/media/slideshow/flash\\_slideshow.html](http://oceanexplorer.noaa.gov/okeanos/media/slideshow/flash_slideshow.html) and [http://oceanexplorer.noaa.gov/okeanos/media/slideshow/video\\_playlist.html](http://oceanexplorer.noaa.gov/okeanos/media/slideshow/video_playlist.html).
- (e) (Optional) Review the *Hands-On Activity Guides* included with this lesson and decide whether you wish to have students complete one or more of these in addition to the image analysis activity described below. *Hands-On Activity Guides* are provided to introduce some basic systems used in many underwater robots to gather information for ocean exploration:
- *Getting Control with Microcontrollers*;
  - *Making Things Happen with Servos*; and
  - *Exploring with Sensors*.

The purpose of these activities is to introduce students and educators to materials, methods, and technologies that they can use for a wide variety of inquiries, activities, and projects that integrate skills in science, technology, engineering, and mathematics. These activities may be used at any grade level, depending upon available resources, curriculum content, and individual student needs. Unless students are already familiar with microcontrollers, the *Getting Control with Microcontrollers* activity should precede the other two.

2. Briefly introduce the NOAA Ship *Okeanos Explorer* and the INDEX-SATAL 2010 Expedition, and discuss why this kind of exploration is important (for background information, please see the lesson, *Earth's Ocean is 95% Unexplored: So What?*; [http://oceanexplorer.noaa.gov/okeanos/explorations/10index/background/edu/media/so\\_what.pdf](http://oceanexplorer.noaa.gov/okeanos/explorations/10index/background/edu/media/so_what.pdf)). Highlight the overall exploration strategy used by *Okeanos Explorer*, including the following points:

- The overall strategy is based on finding anomalies;
- This strategy involves
  - Underway reconnaissance;
  - Water column exploration; and
  - Site characterization;
- This strategy relies on four key technologies:
  - Telepresence technologies that allow people to observe and interact with events at a remote location;
  - Multibeam sonar mapping system;
  - CTD and other electronic sensors to measure chemical and physical seawater properties; and
  - A Remotely Operated Vehicle (ROV) capable of obtaining high-quality imagery and samples in depths as great as 4,000 meters.

You may want to show some or all of the images in the adjacent sidebar to accompany this review.

3. Show students images of several ROVs (see <http://oceanexplorer.noaa.gov/technology/subs/rov/rov.html>), and briefly describe their capabilities. Conclude with one or more images of the *Little Hercules* ROV, and explain that the primary purpose of this underwater robot is to gather high quality video images as part of site characterization activities. At this point you may want to show the video clip compilation referenced in Step 1d. Lead a discussion of how ocean explorers might use these images. Students should realize that in many cases these images provide the only indication of living organisms in sites being characterized. In addition, these images can provide data about geological formations and other environmental conditions that cannot be obtained with other data gathering instruments. Students should also realize that video imagery also has limitations, including:

- coverage is limited to a relatively small area;
- mobile organisms may avoid the moving ROV and/or lights used for video imaging; and
- many organisms cannot be accurately identified from photographs alone.

Show the image **EX2010.07.12\_ROV\_02.jpg**, call students' attention to the red dots near the center of the image, and ask for ideas about why these dots are present. Students should realize (possibly with your help) that these dots provide a scale that makes it possible to estimate the size of objects in the image.

Tell students that their assignment is to learn about some of the ways scientists use imagery from underwater robots to investigate deep-sea ecosystems. Provide each student group with a copy of the *Robot Imagery Inquiry Guide* and the *ImageJ Basics* handout, as well as any instructions about operating the ImageJ software on classroom computer systems. Point out that the primary purpose of the *Inquiry Guide* exercises is to become familiar with techniques that can be used for a wide variety of investigations and projects that use digital imagery.

4. When students have answered all questions on the *Inquiry Guide*, lead a discussion of their results. The following points should be included:

- The approximate height of the portion of the soft coral visible in Figure 1 is 25.4 cm.
- The span of the two brittle star arms that are closest to horizontal is about 10 cm.
- The approximate length of the shrimp in the lower left of Figure 2 is 9.2 cm.
- The approximate length of the violet sea cucumber in the upper center of Figure 2 is 10.9 cm; the sea cucumber may be somewhat larger than this, since it is farther away than the substrate where the laser dots are located.
- The fish in Figure 4 is 6 meters from the robot, since the beam from the third laser converges with the beam from the parallel laser on the right, giving the appearance of only two dots on the fish.
- The total length of the fish in Figure 4 is approximately 30 cm.
- The fish in Figure 5 is 3 meters from the robot, since the beam from the third laser forms a dot that is 5.0 cm from the dot formed by the parallel laser on the right. The hint in Figure 6 is that that the angle  $\theta$  is constant, so the tangent of  $\theta$  is constant and equal to  $d / D$ , and also equal to  $b / B$ .





### STEM Connections

Ocean exploration aboard the *Okeanos Explorer* is a real-world example of STEM concepts in action:

**Science** provides the overall objective – to better understand Earth’s ocean – as well as a methodology for systematically acquiring this understanding;

**Technology** includes the tools, systems and processes that have been made to make deep-ocean exploration possible;

**Engineering** designs the technologies that can function in the deep-ocean environment;

**Mathematics** provides the basis for measurements, data analysis, and engineering design.

With increasing attention to developing integrated approaches to STEM education and technological literacy, the *How Do We Explore?* theme offers an exciting context for educators who wish to bring more STEM content to their classrooms.

To assist with such efforts, most lessons developed for the *How Do We Explore?* theme identify opportunities to include specific benchmarks and standards for technological literacy that have been developed by the International Technology and Engineering Education Association (ITEA, 2007). While these standards have not been widely adopted, they provide useful guidance for efforts to enhance STEM content in advance of its inclusion in formal curricula.

In addition, the *How Do We Explore?* suite of lessons includes activities that are intended to provide opportunities to apply design processes, build technological devices, and develop some of the hands-on abilities that are an integral part of most concepts about STEM education. These activities are directly tied to the technologies and scientific methodologies used for ocean exploration aboard the *Okeanos Explorer*.

For more information, see: [http://www.iteaconnect.org/TAA/Publications/TAA\\_Publications.html](http://www.iteaconnect.org/TAA/Publications/TAA_Publications.html)

- The total length of the fish in Figure 5 is approximately 30 cm.
- Errors involved with video fish surveys include avoidance of the robot by some fish species, attraction to the robot by other species which may also circle the robot and be counted more than once, and the exclusion of cryptic species that hide in crevices that are out of the camera’s view. In some cases, particularly when areas are being explored for the first time, it may not be possible to accurately identify every fish seen. This problem may arise when the characteristics used for identification require close inspection (such as the number of rays in a fin). Accurate identification also depends upon the experience of scientists watching the video; but occasionally, no one can identify a strange organism because it is a species new to science and has never been seen before!
- The width (W) of a video transect made with a camera whose horizontal angle of view (A) is 90 degrees and an effective distance (D) of 5 m is :  

$$W = 2 \cdot (\tan(0.5 \cdot A)) \cdot D = 2 \cdot (\tan 45) \cdot 5 = 2 \cdot 1 \cdot 5 = 10 \text{ m}$$
- The length of a transect that begins at 2.523456° North latitude, 125.069212° East longitude, and ends at 2.525255 North latitude, 125.069146 East longitude is 0.2002 km.
- Using values from questions 8 and 9, the area of the transect is  

$$10 \text{ m} \cdot 200.2 \text{ m} = 2002 \text{ square meters}$$

To correct for the space that is outside the camera’s field of view, it is necessary to subtract

$$0.5 \cdot (W \cdot D) = 25 \text{ square meters}$$

which is the shaded area shown in Figure 6

- The distance between the eyes of the crustacean is about 2.15 cm; the length of the longest segment in the first right cheliped is about 7.80 cm.
- The area occupied by the large white sponge in the lower left of the image is about 353 cm<sup>2</sup>.
- The area of the branching coral in the upper right of the image is about 253 cm<sup>2</sup>.

Note that students’ results will vary depending upon how precisely they outline objects whose areas are to be estimated.

5. Discussion of underwater robots and image technology may also include the following components of technological literacy (ITEA, 2007):
  - **Core concepts of technology:** Underwater robots are composed of closely inter-related systems designed to meet specific requirements and constraints, which often involve trade-offs.
  - **Relationships between technologies and other fields of study:** Underwater robotics technology is closely linked to advancements in science and mathematics, and specific information requirements in these fields drive further innovation and invention.
  - **Effects of technology on the environment:** Underwater robotics technology provides new ways to monitor various aspects of the environment to provide information for decision-making.



6. If desired, have students complete the *Getting Control with Microcontrollers* activity (Step 1e). This activity addresses additional components of technological literacy, including:
- Attributes of design;
  - Understanding of engineering design;
  - Problem-solving approaches;
  - Abilities to apply design processes; and
  - Abilities to use and maintain technological products.

### The BRIDGE Connection

[www.vims.edu/bridge/](http://www.vims.edu/bridge/) – Scroll over “Ocean Science Topics” in the menu on the left side of the page, then “Human Activities,” then “Technology” for links to resources about submersibles, ROVs, and other technologies used in underwater exploration.

### The “Me” Connection

Have students write a brief essay about how robotic imagery might be of personal benefit.

### Connections to Other Subjects

English/Language Arts, Life Science, Mathematics, Physics

### Assessment

Students’ answers to *Inquiry Guide* questions and class discussions provide opportunities for assessment.

### Extensions

1. Visit the *Okeanos Explorer* Digital Atlas ([http://www.ncddc.noaa.gov/website/google\\_maps/OkeanosExplorer/mapsOkeanos.htm](http://www.ncddc.noaa.gov/website/google_maps/OkeanosExplorer/mapsOkeanos.htm)) and Web page (<http://oceanexplorer.noaa.gov/okeanos/welcome.html>) for reports, images, and other products from *Okeanos Explorer* cruises.
2. Visit [http://www.marinetech.org/rov\\_competition/rov\\_video\\_2007.php](http://www.marinetech.org/rov_competition/rov_video_2007.php) for a video from the Marine Technology Society’s student ROV competition, and links to other sites about underwater robots.
3. For ideas about building your own underwater robots, see Bohm and Jensen (1998), Bohm (1997), and the Sea Perch Program (see Other Resources).

### Multimedia Discovery Missions

<http://oceanexplorer.noaa.gov/edu/learning/welcome.html> – Click on the links to Lessons 1, 5, and 6 for interactive multimedia presentations and Learning Activities on Plate Tectonics, Chemosynthesis and Hydrothermal Vent Life, and Deep-Sea Benthos.

### Other Relevant Lesson Plans from NOAA’s Ocean Exploration Program

#### Where’s My ‘Bot?

(from the Bonaire 2008: Exploring Coral Reef Sustainability with New Technologies Expedition)

<http://oceanexplorer.noaa.gov/explorations/08bonaire/background/edu/media/wheresbot.pdf>

Focus: Marine Navigation (Grades 9-12; Earth Science/Mathematics)

Students estimate geographic position based on speed and direction of travel, and integrate these calculations with GPS data to estimate the set and drift of currents.







### Hydrothermal Vent Challenge

(from the Submarine Ring of Fire 2004 Expedition)

[http://oceanexplorer.noaa.gov/explorations/04fire/background/edu/media/RoF\\_ventchall.pdf](http://oceanexplorer.noaa.gov/explorations/04fire/background/edu/media/RoF_ventchall.pdf)

Focus: Chemistry of hydrothermal vents (Grades 9-12; Chemistry)

Students define hydrothermal vents and explain the overall processes that lead to their formation, explain the origin of mineral-rich fluids associated with hydrothermal vents, explain how black smokers and white smokers are formed, and hypothesize how properties of hydrothermal fluids might be used to locate undiscovered hydrothermal vents.

### Thar She Blows!

(from the 2002 Galapagos Rift Expedition)

[http://oceanexplorer.noaa.gov/explorations/02galapagos/background/education/media/gal\\_gr9\\_12\\_13.pdf](http://oceanexplorer.noaa.gov/explorations/02galapagos/background/education/media/gal_gr9_12_13.pdf)

Focus: Hydrothermal vents (Grades 9-12; Chemistry/Biology)

Students demonstrate an understanding of how the processes that result in the formation of hydrothermal vents create new ocean floor, and demonstrate an understanding of how the transfer of energy affects solids and liquids.

### This Life Stinks

(from the 2003 Windows to the Deep Expedition)

[http://oceanexplorer.noaa.gov/explorations/03windows/background/education/media/03win\\_lifestinks.pdf](http://oceanexplorer.noaa.gov/explorations/03windows/background/education/media/03win_lifestinks.pdf)

Focus: Methane-based chemosynthetic processes (Grades 9-12; Chemistry/Biology)

Students define the process of chemosynthesis, and contrast this process with photosynthesis, explain the process of methane-based chemosynthesis, and explain the relevance of chemosynthesis to biological communities in the vicinity of cold seeps.

### The Tell-Tale Plume

(from the INSPIRE: Chile Margin 2010 Expedition)

<http://oceanexplorer.noaa.gov/explorations/10chile/background/edu/media/plume.pdf>

Focus: Hydrothermal Vent Chemistry (Grades 9-12; Chemistry/Biology)

Students describe hydrothermal vents, identify changes that they cause to the physical and chemical properties of seawater, and use oceanographic data to recognize a probable plume from hydrothermal activity.

### The Roving Robotic Chemist

(from the PHAEDRA 2006 Expedition)

[http://oceanexplorer.noaa.gov/explorations/06greece/background/edu/media/robot\\_chemist.pdf](http://oceanexplorer.noaa.gov/explorations/06greece/background/edu/media/robot_chemist.pdf)

Focus: Mass Spectrometry

Students explain the basic principles underlying mass spectrometry, discuss the advantages of in-situ mass spectrometry, explain the concept of dynamic re-tasking as it applies to an autonomous underwater vehicle, and develop and justify a sampling strategy that could be incorporated into a program to guide an AUV searching for chemical clues to specific geologic features.



## Other Resources

*The Web links below are provided for informational purposes only. Links outside of Ocean Explorer have been checked at the time of this page's publication, but the linking sites may become outdated or non-operational over time.*

Anonymous. 2010. Web site for the INDEX-SATAL 2010 Expedition [Internet]. Office of Ocean Exploration and Research, NOAA [cited January 7, 2011]. Available from <http://oceanexplorer.noaa.gov/okeanos/explorations/10index/welcome.html> – Includes links to lesson plans, career connections, and other resources

Anonymous. Ocean Explorer [Internet]. NOAA Office of Ocean Exploration and Research [cited January 4, 2011]. Available from: <http://oceanexplorer.noaa.gov>.

Anonymous. 2011. *Okeanos Explorer* Education Materials Collection [Internet]. NOAA Office of Ocean Exploration and Research [cited January 4, 2011]. Available from: <http://oceanexplorer.noaa.gov>

Anonymous. *Okeanos Explorer* America's Ship For Ocean Exploration [Internet]. NOAA Office of Ocean Exploration and Research [cited January 24, 2011]. Available from: [http://explore.noaa.gov/special-projects/indonesia-u-s-scientific-and-technical-cooperation-in-ocean-exploration/files/Okeanos\\_Explorer\\_for\\_WOC\\_-\\_FINAL.pdf](http://explore.noaa.gov/special-projects/indonesia-u-s-scientific-and-technical-cooperation-in-ocean-exploration/files/Okeanos_Explorer_for_WOC_-_FINAL.pdf); NOAA Fact Sheet about *Okeanos Explorer*

Anonymous. Sea Perch Program [Internet]. Massachusetts Institute of Technology Sea Grant Program. [cited January 12, 2011]. Available from <http://seaperch.mit.edu/> – Includes detailed instructions for building a simple remotely operated underwater vehicle; based on designs from “Build Your Own Under Water Robot and Other Wet Projects” by Harry Bohm and Vickie Jensen

Bohm, H. and V. Jensen. 1998. Build Your Own Programmable Lego Submersible: Project: Sea Angel AUV (Autonomous Underwater Vehicle). Westcoast Words. 39 pages.

Bohm, H. 1997. Build Your Own Underwater Robot and Other Wet Projects. Westcoast Words. 148 pages.

International Technology Education Association. 2007. Standards for Technological Literacy: Content for the Study of Technology. Reston, VA. 260 pages.

## National Science Education Standards

### Content Standard A: Science As Inquiry

- Abilities necessary to do scientific inquiry
- Understandings about scientific inquiry

### Content Standard E: Science and Technology

- Abilities of technological design
- Understandings about science and technology

### Content Standard F: Science in Personal and Social Perspectives

- Science and technology in local, national, and global challenges

### Content Standard G: History and Nature of Science

- Science as a human endeavor
- Nature of scientific knowledge





## Ocean Literacy Essential Principles and Fundamental Concepts

*Because most Fundamental Concepts are broad in scope, some aspects of some Concepts may not be explicitly addressed in this lesson. Such aspects, however, can be easily included at the discretion of the individual educator.*

### Essential Principle 1.

#### The Earth has one big ocean with many features.

**Fundamental Concept b.** An ocean basin's size, shape and features (such as islands, trenches, mid-ocean ridges, rift valleys) vary due to the movement of Earth's lithospheric plates. Earth's highest peaks, deepest valleys and flattest vast plains are all in the ocean.

### Essential Principle 5.

#### The ocean supports a great diversity of life and ecosystems.

**Fundamental Concept e.** The ocean is three-dimensional, offering vast living space and diverse habitats from the surface through the water column to the seafloor. Most of the living space on Earth is in the ocean.

**Fundamental Concept g.** There are deep-ocean ecosystems that are independent of energy from sunlight and photosynthetic organisms. Hydrothermal vents, submarine hot springs, and methane cold seeps rely only on chemical energy and chemosynthetic organisms to support life.

### Essential Principle 6.

#### The ocean and humans are inextricably interconnected.

**Fundamental Concept b.** From the ocean we get foods, medicines, and mineral and energy resources. In addition, it provides jobs, supports our nation's economy, serves as a highway for transportation of goods and people, and plays a role in national security.

**Fundamental Concept g.** Everyone is responsible for caring for the ocean. The ocean sustains life on Earth and humans must live in ways that sustain the ocean. Individual and collective actions are needed to effectively manage ocean resources for all.

### Essential Principle 7.

#### The ocean is largely unexplored.

**Fundamental Concept a.** The ocean is the last and largest unexplored place on Earth—less than 5% of it has been explored. This is the great frontier for the next generation's explorers and researchers, where they will find great opportunities for inquiry and investigation.

**Fundamental Concept b.** Understanding the ocean is more than a matter of curiosity. Exploration, inquiry and study are required to better understand ocean systems and processes.

**Fundamental Concept c.** Over the last 40 years, use of ocean resources has increased significantly, therefore the future sustainability of ocean resources depends on our understanding of those resources and their potential and limitations.

**Fundamental Concept d.** New technologies, sensors and tools are expanding our ability to explore the ocean. Ocean scientists are relying more and more on satellites, drifters, buoys, subsea observatories and unmanned submersibles.

**Fundamental Concept f.** Ocean exploration is truly interdisciplinary. It requires close collaboration among biologists, chemists, climatologists, computer programmers, engineers, geologists, meteorologists, and physicists, and new ways of thinking.

### Send Us Your Feedback

We value your feedback on this lesson, including how you use it in your formal/informal education settings.

Please send your comments to:

[oceaneducation@noaa.gov](mailto:oceaneducation@noaa.gov)

### For More Information

Paula Keener, Director, Education Programs  
NOAA Office of Ocean Exploration and Research  
Hollings Marine Laboratory  
331 Fort Johnson Road, Charleston SC 29412  
843.762.8818 843.762.8737 (fax)  
[paula.keener-chavis@noaa.gov](mailto:paula.keener-chavis@noaa.gov)

### Acknowledgments

Produced by Mel Goodwin, PhD, Marine Biologist and Science Writer, Charleston, SC for NOAA. Design/layout: Coastal Images Graphic Design, Charleston, SC. If reproducing this lesson, please cite NOAA as the source, and provide the following URL: <http://oceanexplorer.noaa.gov>





## Robot Imagery Inquiry Guide

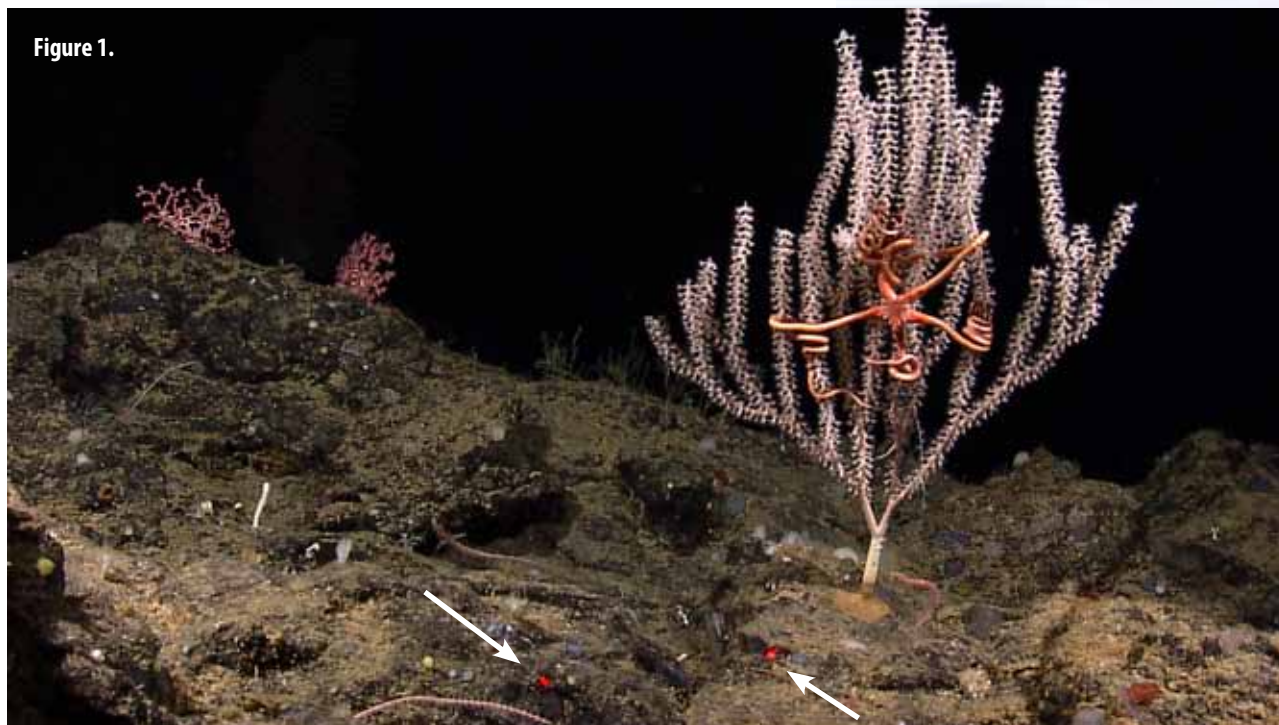
Underwater robots are our eyes into Earth's deep ocean. As digital imaging technology continues to improve, views through these robot eyes are beautiful and fascinating (see, for example, [http://oceanexplorer.noaa.gov/okeanos/media/slideshow/flash\\_slideshow.html](http://oceanexplorer.noaa.gov/okeanos/media/slideshow/flash_slideshow.html) and [http://oceanexplorer.noaa.gov/okeanos/media/slideshow/video\\_playlist.html](http://oceanexplorer.noaa.gov/okeanos/media/slideshow/video_playlist.html)). Site characterization is a key element of the *Okeanos Explorer* strategy for making discoveries, and this characterization depends heavily upon the ability of underwater robots to obtain images that can be used to identify living organisms and geological features. The following activities are designed to acquaint you with some techniques for analyzing images obtained with underwater robots. Once you are familiar with these techniques, you will be able to use them for a wide variety of investigations about ocean organisms and ecosystems.

Many underwater robots carry digital cameras that can look ahead of the robot as well as directly below it. Often, the forward-looking camera records video while the downward-looking camera may record still or video imagery depending upon specific mission needs. For some surveys, a series of overlapping still images is obtained from the downward-looking camera. These images are spliced together to produce a mosaic image that can provide complete coverage of a large area. The essay by Liz Podowski (<http://oceanexplorer.noaa.gov/explorations/09lophelia/logs/aug30/aug30.html>) provides more details about this technique.

One problem with most images from underwater robots is that they are two-dimensional, so it can be very difficult to accurately judge the size of organisms and ecosystem features. To overcome this problem, underwater photography systems use a system of lasers. Downward-looking cameras typically have two lasers whose beams are parallel and a known distance apart. This places two bright dots on each image that establish the scale of the image. In Figures 1 and 2, the dots on the image are 10 cm apart. Using this information, estimate:



Figure 1.



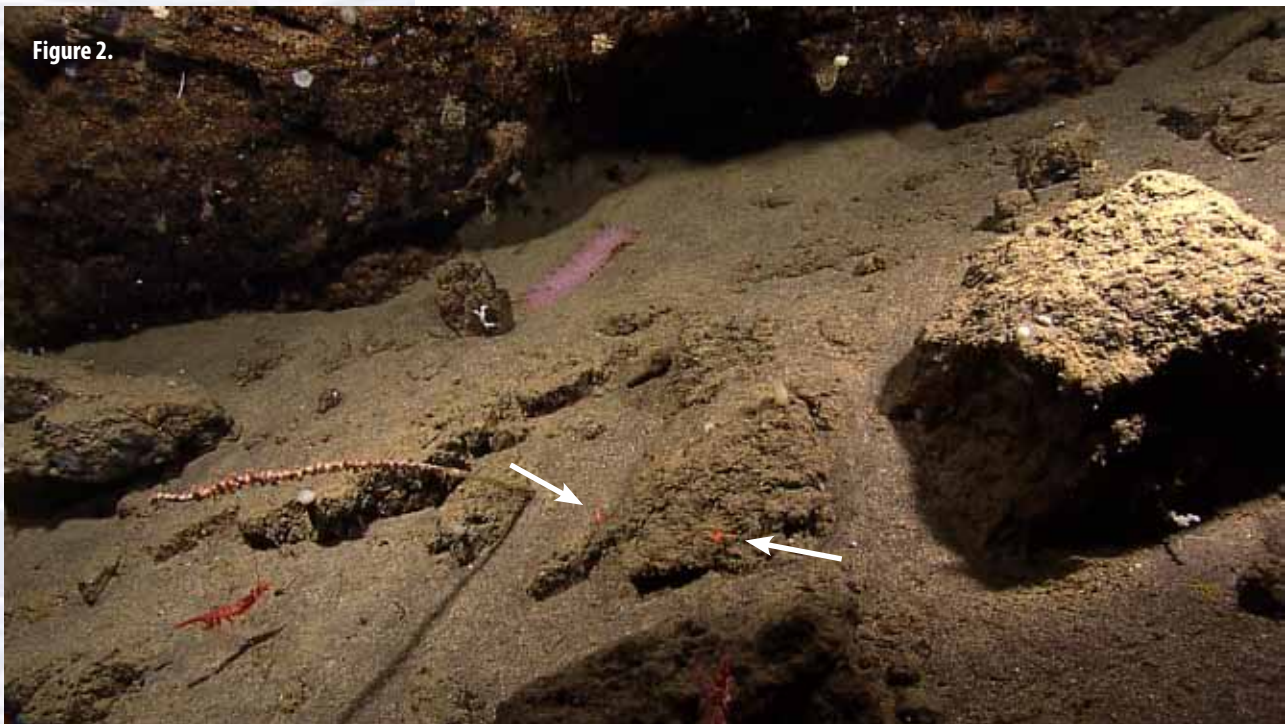
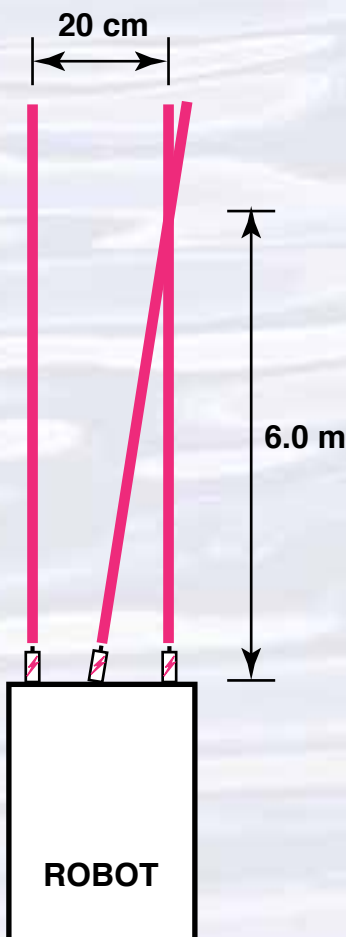


Figure 2.

Figure 3.



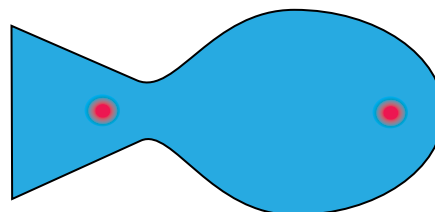
1. The approximate height of the portion of the large branching soft coral that is visible in Figure 1:  
\_\_\_\_\_
2. The span of the two brittle star arms that are closest to horizontal, not including the portions that are twisted around the coral in Figure 1:  
\_\_\_\_\_
3. The approximate length of the shrimp in the lower left of Figure 2:  
\_\_\_\_\_
4. The approximate length of the violet sea cucumber in the upper center of Figure 2:  
\_\_\_\_\_

*Note: the sea cucumber may be somewhat larger than this, since it is farther away than the substrate where the laser dots are located.*

A similar system can be used in forward-looking video, but it is often important to be able to estimate distance as well. To provide this capability, a third laser is added to the system, in between the two parallel lasers. This additional laser is angled so that it converges with one of the parallel laser beams at a known distance in front of the robot (see Figure 3).

Suppose a video system using the lasers diagrammed in Figure 3 shows the image in Figure 4:

Figure 4.



5. How far away is the fish?

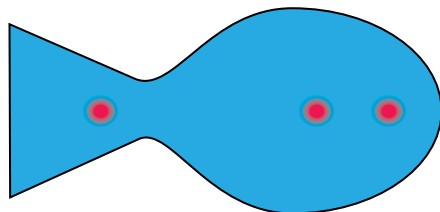
---

6. What is the total length of the fish?

---

Now suppose the same video system shows the image in Figure 5:

**Figure 5.**



7. How far away is the fish?

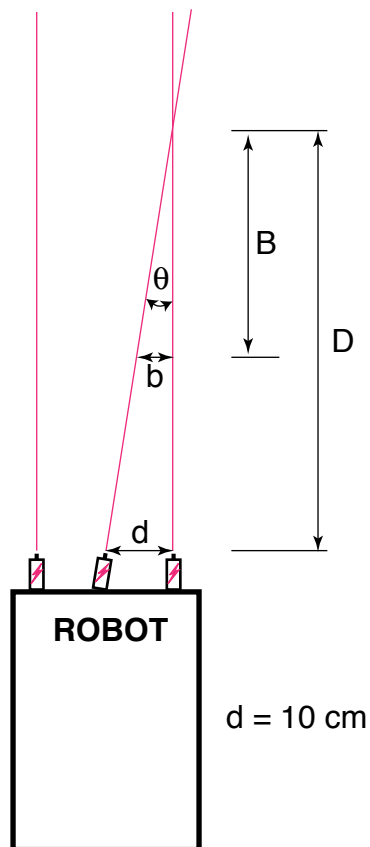
---

8. What is the total length of the fish?

---

HINT:

**Figure 6.**





Fishes are a conspicuous part of many marine ecosystems, and ocean explorers often have questions about what kinds of fish inhabit certain areas, and how many fishes are there. Koenig (2001) describes a way to obtain this information that involves navigating an underwater robot along a series of transects (a transect is a path along which data are collected) while recording video from a forward-looking camera. When the transects are completed, the video recording is reviewed and imaged fishes are identified and counted. The total counts for each species are divided by the total area of the transect to obtain an estimate of the density of each species. For example, if 50 fish belonging to a single species were counted in a video transect that covered 100 square meters, the density of that species would be  $50 \text{ fish} / 100 \text{ square meters} = 0.5 \text{ fish/square meter}$ .

9. What might be some errors involved with surveying fishes in this way?

---

To find the area of a transect, we need to know the width (W) and length (L) of the transect (Figure 7). The width of the transect can be calculated from:

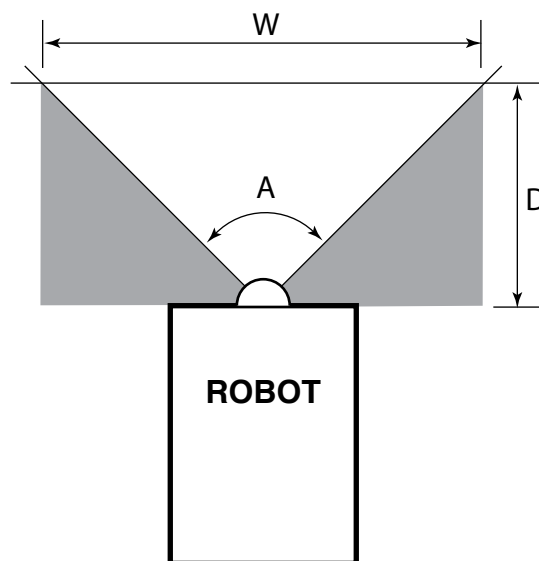
- The effective distance for identifying fish species (D); and
- The video camera's horizontal angle of view (A)

The effective distance for identifying fish species (D) is the limit at which the fish can be identified with a high degree of certainty. Koenig (2001) used a distance of 5 m. The horizontal angle of view (A) depends on the camera used and the position of the zoom.

10. Suppose we have a camera whose horizontal angle of view is 90 degrees. Using 5 m as the effective distance, what is the width (W) of a video transect?

---

**Figure 7.**



The length of a transect (L) can be calculated from the geographic coordinates of the robot at the beginning and end points of the transect. The distance between two points whose latitude and longitude are known can be calculated using the spherical law of cosines formula:

$$d = \text{acos}(\sin(\text{lat1}) \cdot \sin(\text{lat2}) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\text{long2} - \text{long1})) \cdot R$$

where lat1, long1, lat2, and long2 are the latitude and longitude of points 1 and 2, respectively in radians, d is the distance between the points in km, and R is Earth's mean radius (6,371 km); latitude and longitude measurements in degrees may be converted to radians by dividing by 180.

11. If our transect began at 2.523456° North latitude, 125.069212° East longitude, and ended at 2.525255 North latitude, 125.069146° East longitude, what was the length (L) of the transect (Hint: 1. Be sure to use a calculator or spreadsheet program accurate to at least six decimal places.)
- 

12. Now that you know the width (W) and length (L) of the transect, how much area did the transect include? (Hint: be sure to allow for the space that is outside the camera's horizontal field of view.)
- 

Basic analyses of video and single-frame images can be done by identifying and counting individual organisms, and by using laser scale dots to estimate the area covered by the image. Much more sophisticated analyses can be done using image analysis software. ImageJ is a very powerful image analysis program developed by the National Institutes of Health, and is free to the public. Links for downloading, tutorials, and full documentation are available at: <http://rsbweb.nih.gov/ij/index.html>.

Launch the ImageJ program, and open the file provided by your teacher. Review the *ImageJ Basics* handout, then try the following:

- Draw a line between the two laser dots, and set the measurement scale as described in the handout (remember the dots are 10 cm apart).

13. A decapod crustacean is clearly visible near the bottom center of the image. Measure the distance between the eyes, and the length of the longest segment in the first right cheliped (the leg with a claw near the front of the crab), and record your results. (Hint: Use the magnifying glass to enlarge the image for greater accuracy.)
- 

14. What is the approximate area occupied by the large white sponge in the lower left of the image?
- 

15. Notice the branching coral in the upper right of the image. What is the approximate area of this coral? (Outline the entire coral shape, not the individual branches.)
- 





**Figure 1.** A microcontroller



**Figure 2.** A microcontroller placed onto an Arduino board.



**Figure 3.** Lego® NXT brick



## Hands-On Activity Guide

### Underwater Robots:

#### Getting Control with Microcontrollers

This activity is the first in a series of three activities that introduce basic systems used in many underwater robots to gather information for ocean exploration. Other activities in this series are:

- *Exploring with Sensors*; and
- *Making Things Happen with Servos*.

The purpose of these activities is to introduce students and educators to materials, methods, and technologies that they can use for a wide variety of inquiries, activities, and projects that integrate skills in science, technology, engineering, and mathematics.

What is a robot? Most definitions involve the concept of a mechanical device performing human or near-human tasks, and/or behaving in a human-like manner. The key ideas are that a robot has a purpose, and mimics certain human or animal functions.

The earliest robots were operated entirely by humans who were connected by a cable or wireless radio control system. Today, even the simplest robots contain an electronic device called a microcontroller. This device is not only for robots; microcontrollers are found in home appliances, automobiles, marine engines, televisions, media players, interactive games, toys and many other products. A microcontroller is a tiny computer that includes:

- A central processing unit (CPU) that receives input, processes data, and provides output;
- Memory that stores operating instructions, input data, and results of data processing;
- Input and output ports; and
- One or more timers.

All of these functions are contained in a plastic package (a “chip”) that is about the size of a piece of macaroni (Figure 1). Microcontrollers need additional hardware to connect them to other devices, and this hardware is often placed on a circuit board along with the microcontroller (Figure 2).

Many different microcontrollers are used in robots. Two of the best-known and most popular are the Arduino board and the Lego® NXT brick (Figures 2 and 3). Some of their characteristics are compared in Table 1. Igoe (2011) reviews these along with six other microcontroller systems, and offers general guidance on how to choose an entry-level microcontroller. Both of these systems are easy to work with, and absolute beginners can get results in less than 30 minutes. This does not mean, however, that these systems are too simple for “serious” robotics projects. Arduino and NXT microcontrollers have been used in projects ranging from satellites to underwater remotely operated vehicles.

This activity programs a microcontroller to carry out a set of instructions. The other two activities in this series use microcontrollers to operate sensors that collect environmental information, and to operate servos that provide robots with controlled movements. Future activities will use these basic processes to create robots that can perform a variety of tasks related to ocean exploration.



The objective of this activity is to program a microcontroller so that it will flash a light in a specific pattern. This will provide experience with the programming process, as well as with connecting components in a microcontroller-based system.

**Table 1**  
**Some Characteristics of Lego® NXT and Arduino Microcontrollers**

	<b>Lego®</b>	<b>Arduino</b>
<b>Cost</b>	NXT brick alone – \$145 Software – \$80 Base Set – \$280 (includes software; NXT brick; cables; 3 servos; light, touch, ultrasonic sensors; building parts)	Microprocessor Board alone – \$30 Software – free download Starter Kit – \$60 (includes USB cable, prototyping board, power supply adapter, light sensor, electronic components)
<b>Inputs/Outputs</b>	3 motor ports, 4 sensor ports, USB port	14 digital input/output ports, 6 analog input ports, 6 analog output ports, USB port, power supply jack
<b>Programming Language</b>	Mindstorms visual language based on LabVIEW; Robot C software available (\$80)	C-based Arduino language; expandable using C/C++ language
<b>Accessories</b>	Compatible with Lego® construction kits; many projects documented in books and on the Internet; wide variety of accessories and curricula available	Wide variety of add-on kits available; many projects documented in books and on the Internet; open-source construction allows inexpensive electronic components and sensors to be used
<b>Knowledge Base &amp; Support</b>	Many tutorials and examples on the Internet; FIRST Lego League is a team-based robotics competition for 9 – 16 year-olds	Many tutorials and examples on the Internet; world-wide community of users who share ideas and information
<b>Comments</b>	The graphical interface makes it easy to learn programming logic and Lego® construction methods are familiar to many students. The price is high for basic microcontroller projects, and the closed-construction obscures the actual components. Adapting this system to other components can be difficult.	Inexpensive, easily adaptable to many different projects. C/C++ language syntax can be confusing for beginners, but researchers at the MIT Media Lab have recently developed a graphical interface for Arduino programming that provides an alternative to the text-based C/C++ language (Baafi, 2011).

## Materials

### For a Lego® NXT brick system:

- NXT brick microcontroller
- Lego® Color Sensor
- USB cable
- Computer with Lego Mindstorms NXT software installed

### For an Arduino system:

- Arduino Board (many versions are available; the Duemilanove is one of the  
simplest and easiest to learn)
- Light emitting diode (LED) (*e.g.*, Radio Shack 276-0017; exact specifications are  
not critical)
- USB cable
- Computer with Arduinos software installed (download from [www.arduino.cc/en/  
Main/Software](http://www.arduino.cc/en/Main/Software); installation instructions are linked from the same page)

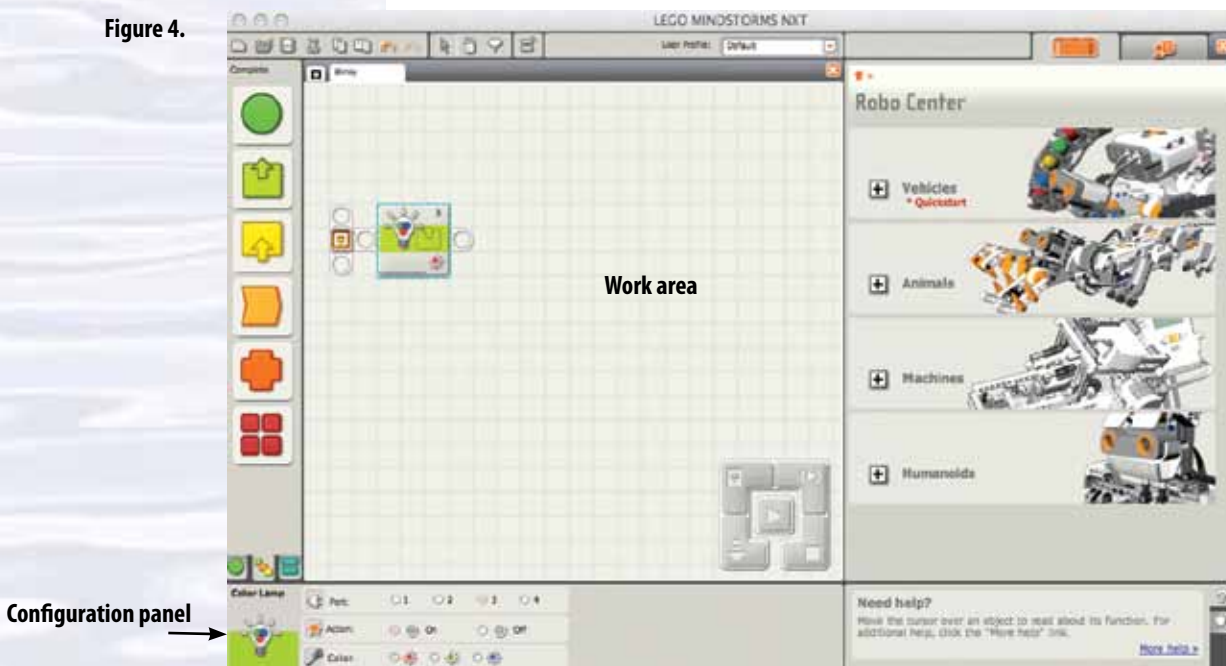
## Procedure

### For a Lego® NXT brick system:

The Lego® NXT brick (Figure 3) has a liquid crystal (LCD) display, four control buttons, three output ports, four input ports, and a USB port. The brick is powered by six size AA batteries.

1. Launch the MINDSTORMS NXT software.
2. In the box beside “Create a new program” enter “Blinky” or whatever name you choose, then click on the “Go >>” button.
3. The programming palette on the left contains all of the building blocks for creating programs for the NXT microcontroller. For this program, click on the middle tab at the bottom (with the three colored squares), then scroll your mouse over on the light green icon (second from the top), and click on the Color Lamp programming block.
4. The area with the light gray grid is the Work Area where programming is done. Drag the Color Lamp block so that it is on top of the square outlined with a blue dashed line, and click to drop the block in this location. Now you have a program with two steps: Start, and Do-Something-With-the-Color-Lamp. See Figure 4.

Figure 4.



5. Many programming blocks can do several things. The Configuration panel on the lower left side of the window is where you specify what a programming block is supposed to do. For the Color Lamp block, the buttons next to “Port:” tell the microcontroller where the Color Lamp is connected. The buttons next to “Action:” tell the microcontroller what to do with the Color Lamp. The buttons next to “Color:” tell the microcontroller the color to which the action applies. For now, be sure these buttons are set to “Port: 3”, “On”, and “Red”.

6. Connect the Color Sensor to Port 3 of the NXT brick with a connector cable.

7. Turn on the NXT brick, and connect it to your computer’s USB port with a USB cable.

8. In the NXT software, click the “Download and run” button on the Controller. When the program is finished compiling and downloading, the Color Lamp should be shining a red light.
9. Notice the two grid boxes to the left and right of the Color Lamp block that resemble pull tabs on soda cans. These are part of the sequence beam, which controls the flow of your program. Blocks must be connected to the sequence beam to be downloaded to the NXT microcontroller. You can find out more about the sequence beam and any other object in the Work Area by placing your cursor on top of the object and reading information in the Help box on the lower right.
10. Now let’s make the Color Lamp turn on for one second, then turn off for one second, then turn back on, and repeat this cycle. In the programming palette, scroll your mouse over the green circle at the top, and click on the Wait programming block (it looks like an hourglass).
11. Drag the Wait block so that it is to the right of the Color Lamp block, and click to drop the block in this location. In the Configuration panel, select “Time” in the window to the right of “Control:”. Be sure the “Seconds” window to the right of “Until:” is set to “1”.
12. Scroll your mouse over the light green icon in the programming palette, and click on the Color Lamp programming block. Drag and drop this block to the right of the Wait block. In the Configuration panel, set “Port:” to “3”, “Action:” to “off”, and “Color:” to “red”.
13. Put another Wait programming block to the right of the Color Lamp block that was added in Step 12. In the Configuration panel, be sure that “Time” is set in the window to the right of “Control:”, and that the “Seconds” window to the right of “Until:” is set to “1”.



Figure 5.

14. In the programming palette, scroll your mouse over the green circle at the top, and click on the Loop programming block (two curved arrows). Drag and drop this block to the right of the Wait block that was added in Step 13. Click on the button in the window to the right of “Control:” This window specifies how long the loop will continue to run. For now, set this window to “Forever” which means that programming instructions inside the loop will repeat until the microcontroller is turned off.
15. Now we have to put all of the programming steps that we want to repeat inside the loop. To do this, click the first step (the first Color Lamp block), then hold down the shift key and click on the other three steps. Click and hold on the selected blocks, and drag your cursor to the middle of the Loop block. The Loop block will expand to enclose all four steps. See Figure 5.

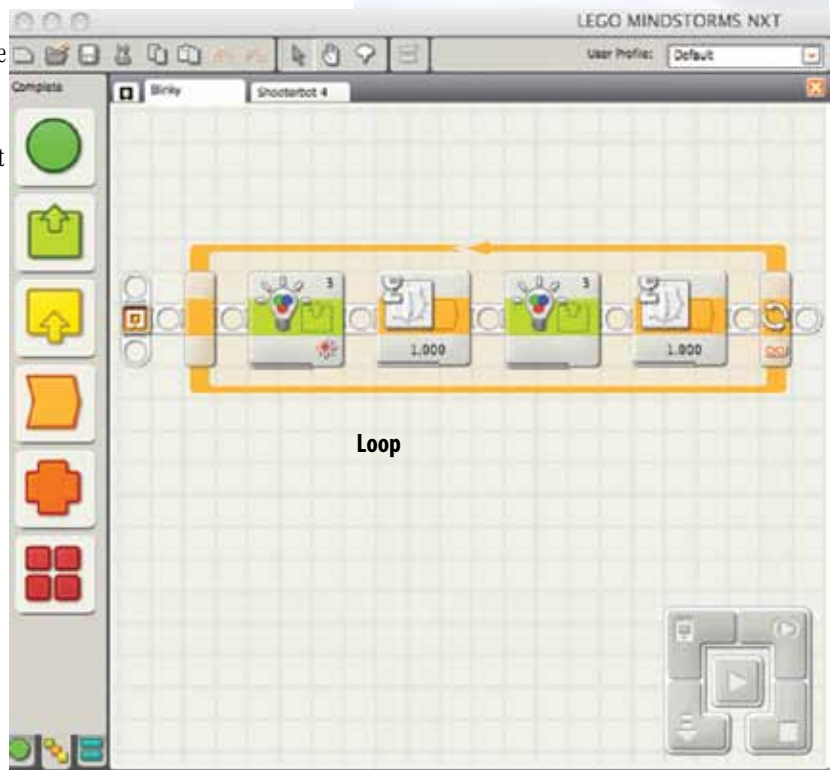
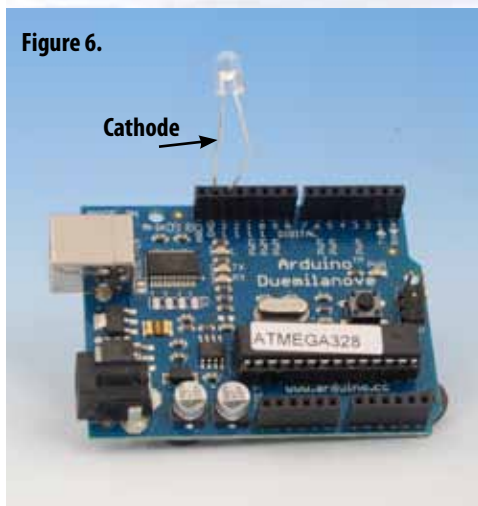




Figure 6.



16. Click the “Download and run” button on the Controller. When the program is finished compiling and downloading, the Color Lamp should be blinking on for one second, then off for one second.
17. Experiment with settings in the Configuration panel for some of your program blocks. Try different wait times, and see how they affect the Color Lamp. Try using “Count” instead of “Forever” for the Loop block to make the Color Lamp blink a few times and then stop. Once the program stops, you can make it run again by pressing the orange button on the NXT brick.
18. **Challenge:** Create a program that will flash SOS in Morse code. In Morse code, S is three short flashes, and O is three long flashes (a long flash lasts three times longer than a short flash).

#### For an Arduino system:

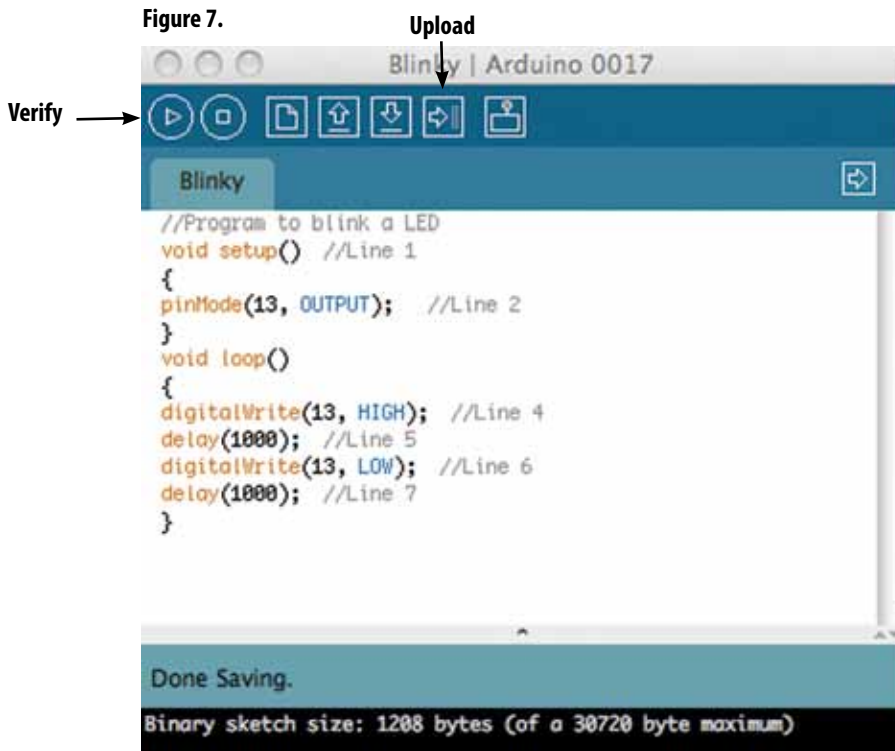
The Arduino board (Figure 2) has 14 digital input/output pins (you use your programs to specify whether they are input or output) numbered 0 – 13; six analog input pins numbered 0 – 5; six analog output pins numbered 3, 5, 6, 9, 10, and 11; a USB port; and power supply jack (the board can use the USB port for power, but will automatically use a power supply if one is connected to the power supply jack).

1. Notice that one side of your LED is flattened. This is the cathode side, which is connected to the negative side of a power source.
2. Insert the cathode lead from your LED into the socket marked “GND” and the other (anode) lead into socket 13. See Figure 6.
3. Connect the Arduino board to your computer with a USB cable.
4. Launch the Arduino software. Select “New” from the “File” menu. Select the folder in which you want your program to be stored, and name your program “Blinky” or whatever name you choose.
5. In the Arduino world, programs are called “sketches”. Type the following program into the Arduino Sketch Editor—the main window of the Arduino software (see Figure 7):

```
//Program to blink a LED
void setup()                //Line 1
{
  pinMode(13, OUTPUT);     //Line 2
}
void loop()                 //Line 3
{
  digitalWrite(13, HIGH);  //Line 4
  delay(1000);              //Line 5
  digitalWrite(13, LOW);   //Line 6
  delay(1000);              //Line 7
}
```

6. Click on the “Verify” button. If everything is correct, the message “Done compiling” will appear at the bottom of the window. If an error message appears, carefully check your typing and be sure your program is identical to the one listed above.





7. Click on the “Upload” button (second from right) to send your program to the Arduino microcontroller. Two small LEDs built into the board will flash, and in a few seconds, the “Done uploading” message will appear in the bottom window. Your program should now be running and the LED should be blinking on for one second, then off for one second.

8. Experiment with the values in the delay functions to see what effect they have on the pattern of blinking.

9. **Challenge:** Create a program that will flash SOS in Morse code. In Morse code, S is three short flashes, and O is three long flashes (a long flash lasts three times longer than a short flash).

10. Here’s what the lines in the program mean:

Any text that begins with // is ignored by Arduino. This kind of text is used for notes that say what is happening in various parts of the program. This is very useful for other people who may use your program, and also for you if you haven’t looked at the program for a while and are trying to remember what each step does.

Any text that doesn’t begin with // is called “code,” and provides specific directions to the microcontroller about what it is supposed to do and when it is supposed to do it. Each line of code must end with a semicolon (;)

Curly braces {} are used to enclose blocks of code that do certain things. Arduino expects that two blocks of code will be present in every program: a block named setup(), and a block named loop(). Curly braces must always be used in pairs.

Line 1 says that the next block of code will be a function named setup(). The word “void” means that the function named setup() will not return any



information once it has done what it is supposed to do. Other functions such as a math function like `tan` will return information; in this case, the tangent of a number.

Line 2 sets digital pin number 13 to be an output.

Line 3 says that the next block of code will be a function named `loop()`. The word `void` means the same thing as in Line 1.

Line 4 sets digital pin number 13 to HIGH, which basically means that pin is turned on and receives a voltage between three and five volts. The word `digitalWrite` is an instruction used to set a pin HIGH or LOW (low means that the pin receives one volt or less).

Line 5 tells the microcontroller to wait 1,000 milliseconds (one second) before doing anything else.

Line 6 sets digital pin number 13 to LOW, which means that pin is turned off.

Line 7 tells the microcontroller to wait another 1,000 milliseconds (one second) before doing anything else.

The curly brace following Line 7 is the end of the loop that began with the curly brace just before Line 4. The microcontroller will continue repeating Steps 4 through 7 until power is turned off or another program is loaded.

### Resources

Baafi, E. 2011. Drag-n-Drop Arduino Programming. *Make* 25:52-56; describes Modkit, a free, web-based editor that uses graphical programming blocks to program Arduino boards

Banzi, M., D. Cuartielles, T. Igoe, G. Martino, and D. Mellis. Arduino [Internet]. Available from: <http://arduino.cc/en/>; Web site of the Arduino Team

Banzi, M. 2008. Getting Started with Arduino. O'Reilly Media, Inc. Sebastopol, CA. 118 pages.

Harrison, T. 2010. Links for Beginners in Electronics [Internet] Available from <http://www.toddfun.com/2010/02/09/beginners-in-electronics/>

Igoe, T. 2011. Getting Started with Microcontrollers. *Make* 25:42-45.

Kelly, J. R. 2010. Lego Mindstorms Nxt-g Programming Guide. Apress. New York. 336 pages.

Palmisano, J. S. Society of Robots [Internet]. Available from <http://www.societyofrobots.com/>; tutorials and basic information for beginning robotics enthusiasts



### Note for Educators

This activity is intended to familiarize students with very basic steps in using two popular types of microcontrollers. Students are not expected to be proficient programmers following this activity; only to know how to follow instructions for creating programs and how to upload such programs to the microcontroller they are using. Further experience with programming and using microcontrollers is provided in other activities cited above, as well as in future activities that will be developed as part of the *Okeanos Explorer* Education Materials Collection. Additional tutorials and applications can be found in the sources listed under Resources.

Solutions to the SOS challenge usually involve:

- creating a loop to generate three short flashes;
- creating another loop to generate three flashes that are three times longer than the short flashes; and
- creating a third loop that alternates three short flashes with three long flashes.

Slightly more sophisticated solutions will insert additional time delays to separate letters, and a longer delay between SOS groups.



## Hands-On Activity Guide

### Underwater Robots: Making Things Happen with Servos

This activity is the second in a series of three activities that introduce basic systems used in many underwater robots to gather information for ocean exploration. Other activities in this series are:

- *Getting Control with Microcontrollers*; and
- *Exploring with Sensors*.

The purpose of these activities is to introduce students and educators to materials, methods, and technologies that they can use for a wide variety of inquiries, activities, and projects that integrate skills in science, technology, engineering, and mathematics.

Microcontrollers, sensors, and servos are systems that allow robots to interact with their environment. Sensors provide information about the environment, microcontrollers process the information and send signals that cause the robot to take actions according to instructions that are included in the microcontroller's program. Often, those "action signals" are sent to servos.

Servos are basically electric motors. But unlike familiar motors that spin continuously when they receive power, the motors in servos are combined with gears and electronic controls that cause them to rotate a certain number of degrees and then stop. This makes servos very useful when precise movements are needed, such as might be desired for a robotic arm. Servos are used in many familiar devices such as copy machines; computer controlled lathes and mills; and radio-controlled model cars, aircraft, and boats. Figure 1 shows a typical servo. The shaft of the servo motor is attached to a mechanical linkage called a servo horn that connects the servo to whatever is supposed to move when the shaft rotates. Servo horns come in many different sizes and shapes.

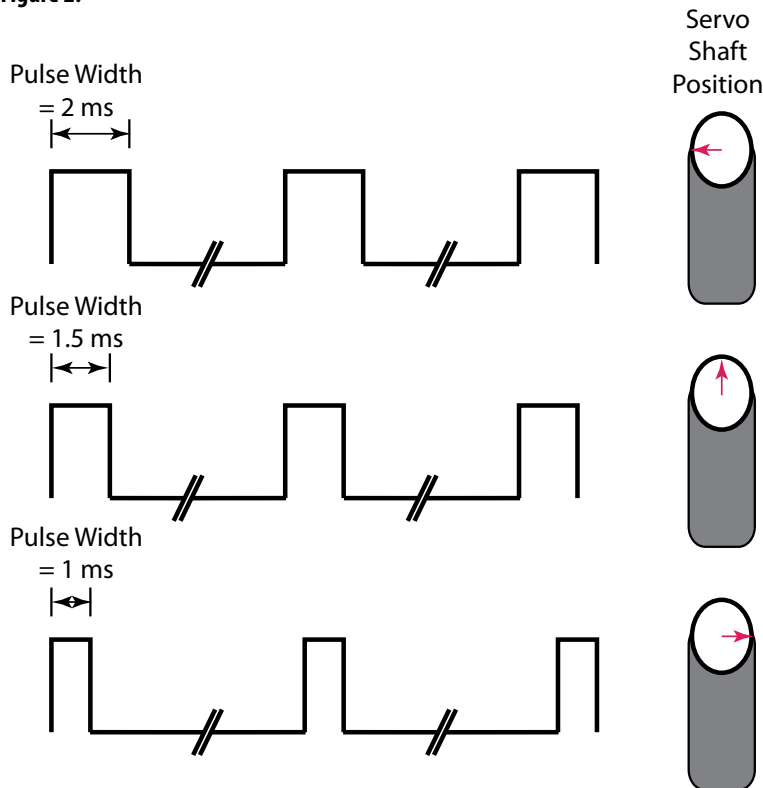
**Figure 1.** A typical servo.





Servos are controlled by pulsing digital signals, such as may be produced by a microcontroller. An example of this kind of signal is shown in Figure 2. The duration of each pulse tells the servo how much to rotate. In many servos, pulses 1 millisecond long tell the servo to rotate as far as it can go in one direction (either clockwise or counterclockwise). Pulses 2 milliseconds long instruct the servo to rotate as far as it can go in the opposite direction. Pulses between 1 and 2 milliseconds tell the servo to rotate to positions in between these extremes. It is important to note that more than one pulse is needed. For a digital signal to cause a servo to rotate, there must be a series of pulses, separated by a certain time interval (usually between 20 and 50 milliseconds). This is an example of pulse-width modulation (PWM), which is a very important technique for controlling a variety of robotic systems.

**Figure 2.**



Typical servos are not able to rotate continuously through 360°; but it is possible to modify many servos so that they can rotate continuously. Why would anyone want to do that? Because servos are more than just electric motors; they also contain gears and electronic control circuits. A servo that is modified to provide continuous rotation is a high-quality, variable speed motor that can be controlled by simple signals from a microcontroller. Pulse width controls the rotation speed of a modified servo.

This activity uses a microcontroller to operate a servo. Future activities will use these basic processes to create robots that can perform a variety of tasks related to ocean exploration. The objective of this activity is to program a microcontroller so that it will cause a servo to move in a controlled way. This will provide experience with programming, as well as an introduction to some of the capabilities of servos.





**Figure 3.** Lego® Servo Motor with tire attached.



## Materials

### For a Lego® NXT brick system:

- NXT brick microcontroller
- Lego® Servo Motor
- Wheel and axle to fit Servo Motor hub
- USB cable
- Computer with Lego® Mindstorms NXT software installed

### For an Arduino system:

- Arduino Board (many versions are available; the Duemilanove is one of the simplest and easiest to learn)
- Standard servo (*e.g.*, Futaba S3003; available from hobby stores and the Internet)
- 3 - Jumper wires to connect the servo cable to the Arduino Board
- USB cable
- Computer with Arduino software installed (download from [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software); installation instructions are linked from the same page)

## Procedure

*Note: These instructions assume that you have completed the “Getting Control with Microcontrollers” activity.*

### For a Lego® NXT brick system:

Lego® Servo Motors have built-in rotation sensors that allow very precise control. The rotation sensors measure motor rotation in units of one degree, or in units of one full rotation.

1. Attach a wheel and rubber tire to the Servo Motor (Figure 3). Connect the Servo Motor to Port A of the NXT brick with a connector cable. Turn on the NXT brick, and connect it to your computer's USB port with a USB cable.
2. Launch the MINDSTORMS NXT software.
3. From the “File” menu, select “New” then “Save As,” enter “Servo” or whatever name you choose, then click “Save.”
4. In the programming palette (on the left), click on the middle tab at the bottom (with the three colored squares), then scroll your mouse over on the green circle (at the top), and click on the Move programming block (gears).
5. Drag the Move block so that it is on top of the square outlined with a blue dashed line, and click to drop the block in this location.
6. The Configuration Menu (lower left) sets the Servo Motor's speed and direction of rotation. If you are controlling two Servo Motors, you can also specify whether they should run at the same speed (so a robot would move in a straight line) or at different speeds (so the robot would turn). If you have Servo Motors on the left and right side of a robot, you can also set the amount of turn and turn direction. Set “Port” to “A,” and “Direction” to forward (arrow pointing up). In the windows to the right of “Duration,” enter 5 and select “Rotations.”
7. Click the “Download and run” button on the Controller. When the program is finished compiling and downloading, the wheel attached to the Servo Motor

should rotate five times and then stop. You can press the orange button on the NXT brick if you want to run the program again.

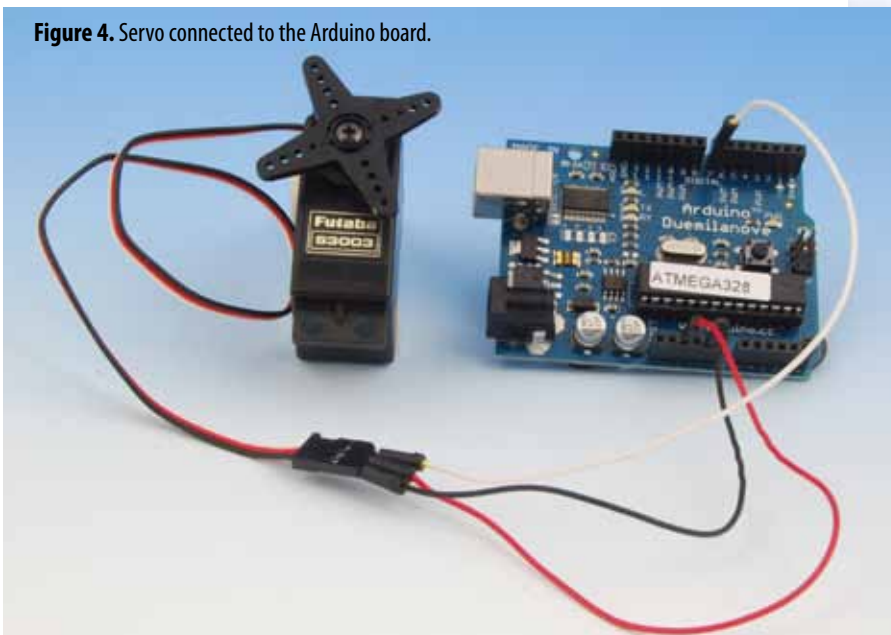
8. Change your program so that the Servo Motor will rotate one-half of one revolution: In the windows to the right of “Duration,” enter 180 and select “Degrees.” Select the “Brake” button to the right of “Next Action.”
9. Click the “Download and run” button on the Controller. When the program runs, the wheel attached to the Servo Motor should rotate one-half turn and then stop. Press the orange button on the NXT brick to run the program again, while you watch the wheel closely to confirm that it rotated one-half turn.
10. Change the “Duration” settings to rotate the wheel 90 degrees, and run the program to confirm that each rotation is one-fourth of a complete revolution.
11. **Challenge:** Use a servo instead of hydraulics to control the simple actuator or part of the robotic arm constructed in the *Invent a Robot* lesson. You will have to:
  - Devise a way to mount the servo;
  - Connect the servo horn to the part to be moved;
  - Modify the program to provide the correct amount of movement.

#### For an Arduino system:

Servos have three wires that must be connected to the control system. The black wire connects to the negative side of the power supply; the red wire connects to the positive side of the power supply; and the white, yellow, or orange wire connects to the source of the digital signal that will control the servo.

1. Use jumper wires to connect the red wire from the servo to the “5V” socket on the Arduino board; the black wire from the servo to the “Gnd” socket on the Arduino board; and the white, yellow, or orange wire to digital pin 7 on the Arduino board (Figure 4).

**Figure 4.** Servo connected to the Arduino board.



2. Connect the Arduino board to your computer with a USB cable.

3. Launch the Arduino software. Select “New” from the “File” menu. Select the folder in which you want your program to be stored, and name your program “Servo” or or whatever name you choose.
4. Type the following program into the Arduino Sketch Editor (the main window of the Arduino software):

```
//Program to Test Servo Rotation
void setup() //Line 1
{
  pinMode(7,OUTPUT); //Line 2
}
void loop() //Line 3
{
  int counter; //Line 4
  for (counter= 0; counter<= 75; counter++) //Line 5
  {
    digitalWrite(7,HIGH); //Line 6
    delayMicroseconds(1000); //Line 7
    digitalWrite(7,LOW); //Line 8
    delay(20); //Line 9
  }
  for (counter= 0; counter<= 75; counter++) //Line 10
  {
    digitalWrite(7,HIGH); //Line 11
    delayMicroseconds(1500); //Line 12
    digitalWrite(7,LOW); //Line 13
    delay(20); //Line 14
  }
  for (counter= 0; counter<= 75; counter++) //Line 15
  {
    digitalWrite(7,HIGH); //Line 16
    delayMicroseconds(2000); //Line 17
    digitalWrite(7,LOW); //Line 18
    delay(20); //Line 19
  }
}
```

5. Click on the “Verify” button. If everything is correct, the message “Done compiling” will appear at the bottom of the window. If an error message appears, carefully check your typing and be sure your program is identical to the one listed above.
6. Click on the “Upload” button (second from right) to send your program to the Arduino microcontroller. Two small LEDs built into the board will flash, and in a few seconds, the “Done uploading” message will appear in the bottom window. Your program should now be running, and the servo should be rotating from its full clockwise position, to its middle position, to its full counterclockwise position, then back to its full clockwise position. If the direction of the rotation sequence is opposite to this, it’s OK; the important thing is that your servo is rotating between two positions with a stop in between.
7. Experiment with the delay time in Line 7. If the delay time is smaller, the pulses from this part of the program will be shorter. Does the servo rotate farther if the





pulses are shorter? If your servo vibrates after it has rotated, it means that it has reached the limits of its rotation. **Don't leave your servo in a vibrating condition for very long, since it could damage the gear train.**

8. Experiment with the delay time in Line 17. If the delay time is larger, the pulses from this part of the program will be longer. Does the servo rotate farther if the pulses are longer? See the warning above about vibration.
9. Once you have determined the limits of rotation for your servo, try changing the delay time in Line 12. Can you position the servo horns exactly midway between the full clockwise and counterclockwise position?
10. Modify the program so that the servo starts at its full counterclockwise position, then travels in three equal steps to its full clockwise position, and then rotates back to the full counterclockwise position.
11. **Challenge:** Use a servo instead of hydraulics to control the simple actuator or part of the robotic arm constructed in the *Invent a Robot* lesson. You will have to:
  - Devise a way to mount the servo;
  - Connect the servo horn to the part to be moved;
  - Modify the program to provide the correct amount of movement.

12. Here's what the lines in the program mean:

Any text that begins with // is ignored by Arduino. This kind of text is used for notes that say what is happening in various parts of the program. This is very useful for other people who may use your program, and also for you if you haven't looked at the program for a while and are trying to remember what each step does.

Any text that doesn't begin with // is called "code," and provides specific directions to the microcontroller about what it is supposed to do and when it is supposed to do it. Each line of code must end with a semicolon (;), except "void" and "for" statements.

Curly braces {} are used to enclose blocks of code that do certain things. Arduino expects that two blocks of code will be present in every program: a block named setup(), and a block named loop(). Curly braces must always be used in pairs.


Line 1 says that the next block of code will be a function named setup(). The word "void" means that the function named setup() will not return any information once it has done what it is supposed to do. Other functions such as a math function like tan will return information; in this case, the tangent of a number.

Line 2 sets digital pin number 7 to be an output.

Line 3 says that the next block of code will be a function named loop(). The word void means the same thing as in Line 1.

Line 4 says that the program will use a variable named "counter" and that it is an integer.





Line 5 is a “for” statement which says that the program is supposed to repeat the next block of code enclosed by curly braces, and to keep on repeating it until a certain condition is met. The “for” statement includes three pieces of information in parentheses, separated by semicolons: (1) the starting condition of some variable, in this case “counter” which is given a starting value of 0; (2) the condition that must be met for the code to repeat again, in this case that “counter” is less than or equal to 75; and (3) how much the variable increases or decreases with each repetition, in this case “++” means that the value of “counter” increases by 1 with each repetition (if “++” were replaced with “--” the value of “counter” would be decreased by 1 with each repetition). The result of this statement is that the following block of code (Lines 6 through 9) will repeat 75 times.

Line 6 sets digital pin number 7 to HIGH, which basically means that pin is turned on and receives a voltage between three and five volts. The word digitalWrite is an instruction used to set a pin HIGH or LOW (low means that the pin receives one volt or less).

Line 7 tells the microcontroller to wait 1,000 microseconds (one millisecond) before doing anything else. Since digital pin number 7 is HIGH during this time, the pulse width will be 1 millisecond. This instruction sets the width of the digital pulses.

Line 8 sets digital pin number 7 to LOW, which means that pin is turned off.

Line 9 tells the microcontroller to wait another 20 milliseconds before doing anything else. This sets the interval between pulses.

Lines 10 and 15 are similar to Line 5.

Lines 11 and 16 are similar to Line 6.

Lines 12 and 17 are similar to Line 7, and set the width of the digital pulses that are sent by this part of the program.

Lines 13 and 18 are similar to Line 8.

Lines 14 and 19 are similar to Line 9, and set the interval between pulses that are sent by this part of the program.

## Resources

Kurt, T.E. 2010. Servomotors. *Make* 19:140-147.

Palmisano, J. S. Actuators - How to Modify a Servo [Internet]. Available from: [http://www.societyofrobots.com/actuators\\_modifyservo.shtml](http://www.societyofrobots.com/actuators_modifyservo.shtml); directions for modifying a servo so that it rotates continuously

Palmisano, J. S. Actuators - Mechanics - Robot Chassis Construction [Internet]. Available from: [http://www.societyofrobots.com/mechanics\\_chassisconstruction.shtml#wheels](http://www.societyofrobots.com/mechanics_chassisconstruction.shtml#wheels); how to mount a wheel on a servo; and [http://www.societyofrobots.com/mechanics\\_chassisconstruction.shtml#servos](http://www.societyofrobots.com/mechanics_chassisconstruction.shtml#servos); how to attach a servo to a robot chassis

## Hands-On Activity Guide

### Underwater Robots: Exploring with Sensors

This activity is the third in a series of three activities that introduce basic systems used in many underwater robots to gather information for ocean exploration. Other activities in this series are:

- *Getting Control with Microcontrollers*; and
- *Making Things Happen with Servos*.

The purpose of these activities is to introduce students and educators to materials, methods, and technologies that they can use for a wide variety of inquiries, activities, and projects that integrate skills in science, technology, engineering, and mathematics.

Microcontrollers, sensors, and servos are systems that allow robots to interact with their environment. Microcontrollers send signals to servos and other actuators that cause the robot to take actions according to instructions that are included in the microcontroller's program. Often, those "action signals" are sent as a response to environmental information that comes from sensors.

Ocean explorers use many types of sensors that may be attached to ships, CTDs, underwater robots, or used independently. These include sensors that measure pressure (often to determine depth), temperature, salinity, pH, dissolved oxygen, redox potential (an indicator of hydrothermal vent activity), optical backscatter (an indicator of methane oxidation), and colored dissolved organic matter (an indicator of petroleum products). Some of these sensors are also available for use in classrooms. Many others are suitable for use with microcontrollers, including sensors that measure acceleration, force, bending, magnetic fields, light, infrared, motion, humidity, and distance using ultrasonic waves (see the *PING*))<sup>TM</sup> *Sonar Simulation* activity for an example of the latter sensor).

The majority of sensors measure various environmental conditions with electric circuits whose current flow changes in response to changes in the environment. Environmental conditions are related to changes in current flow by calibration. This involves graphing changes in current flow against known levels of the environmental condition that caused these changes. Information from calibration graphs is often built into modern instruments, so the instruments appear to provide direct readings of the condition they are supposed to measure; but these readings are based on current flow in an electric circuit.

In addition to sensors that can be used with microcontrollers, Vernier Software and Technology produces a wide variety of data collection systems specifically designed for classroom educators. These systems are based on sensors that are connected to handheld calculators or computers, and include more than 40 different types of sensors. Adapters are available to connect Vernier sensors to NXT bricks, and Vernier offers cables, connectors, and adapters that allow similar connection to other microcontrollers.

This activity uses a microcontroller to read a photocell (also called a photoresistor), which is a sensor whose electrical resistance changes with changing light levels. Future activities will use a similar process to make various environmental measurements that are useful for ocean exploration, and will show how you can program a robot to take actions based on what it senses about its environment.





## Materials

### For a Lego® NXT brick system:

- NXT brick microcontroller
- Lego® Servo Motor
- USB cable
- Computer with Lego Mindstorms NXT software installed
- Colored filters, approximately 2" x 2"; red, green, and blue

### For an Arduino system:

- Arduino Board (many versions are available; the Duemilanove is one of the simplest and easiest to learn)
- Prototyping board (most types will work; the Adafruit Protoshield connects directly to the Arduino board, and has other nice features ([http://www.adafruit.com/index.php?main\\_page=product\\_info&cPath=17\\_21&products\\_id=51](http://www.adafruit.com/index.php?main_page=product_info&cPath=17_21&products_id=51)))
- 3 - Jumper wires to connect the servo cable to the Arduino Board
- Cadmium sulfide photocells (also called photoresistors; often available in “variety packs;” e.g., Radio Shack 276-1657)
- Resistor - see Notes to Educators
- Colored filters, approximately 2" x 2"; red, green, and blue
- USB cable
- Computer with Arduino software installed (download from [www.arduino.cc/en/Main/Software](http://www.arduino.cc/en/Main/Software); installation instructions are linked from the same page)

## Procedure

*Note: These instructions assume that you have completed the “Getting Control with Microcontrollers” activity.*

### For a Lego® System

1. Connect the Color Sensor to Port 3 of the NXT brick with a connector cable. Turn the brick on by pushing the orange button.
2. Launch the MINDSTORMS NXT software.
3. In the box beside “Create a new program” enter “ColorSensor” or whatever name you choose, then click on the “Go >>” button.
4. In the programming palette (on the left), click on the middle tab at the bottom (with the three colored squares), then scroll your mouse over on the yellow square with an arrow, and click on the Color Sensor programming block (with the red, green, and blue circles).
5. Drag the Color Sensor block so that it is on top of the square outlined with a blue dashed line, and click to drop the block in this location.
6. In the Configuration Menu (lower left) Set “Port” to “3,” and “Action” to “Light Sensor.” The “Light” box to the right of “Function” should be unchecked.
7. The feedback box on the left side of the configuration panel shows the current light value (0-100) being measured by the sensor. Hold your hand over the photocell, and the reading should decrease. Increase the light falling on the photocell and the reading should increase.



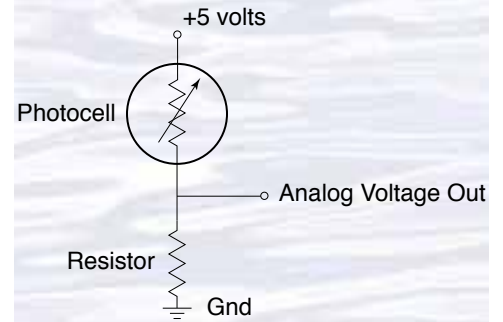
8. **Challenge:** Are photocells equally sensitive to all colors of light? Use the colored filters to find out.

**For an Arduino System:**

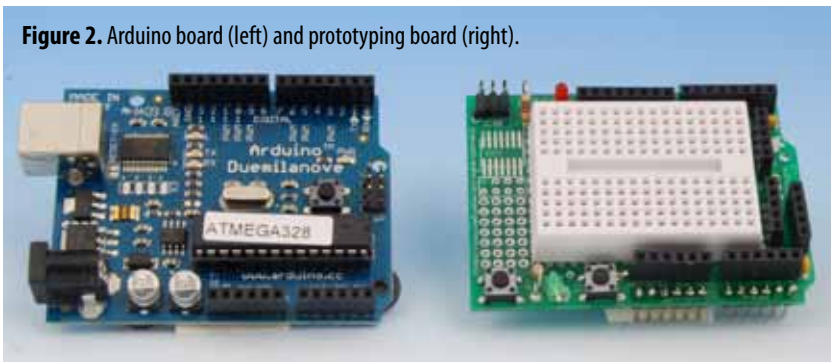
This activity requires a simple circuit (Figure 1). In this circuit, the photocell is connected to a resistor, and power leads are connected so that current flows through the resistor and photocell. An analog input port of the microcontroller is connected to read the voltage at the junction between the resistor and photocell. As more light strikes the photocell, its resistance decreases, so the voltage increases.

The easiest way to build this circuit is with a prototyping board. Figure 2 shows a prototyping board next to an Arduino board, and Figure 3 shows the two boards attached. When the two boards are joined, sockets from the Arduino board are connected to sockets on the prototyping board to simplify circuit building.

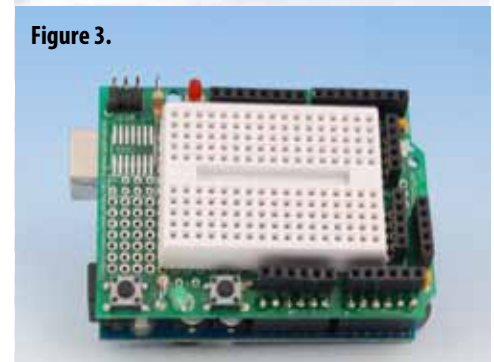
**Figure 1. A simple circuit.**



**Figure 2. Arduino board (left) and prototyping board (right).**

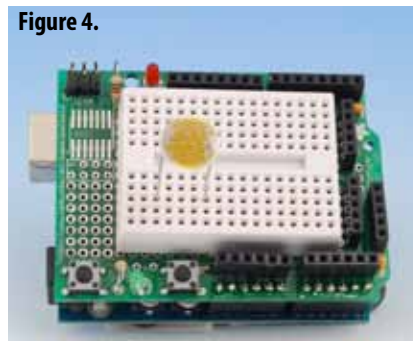


**Figure 3.**

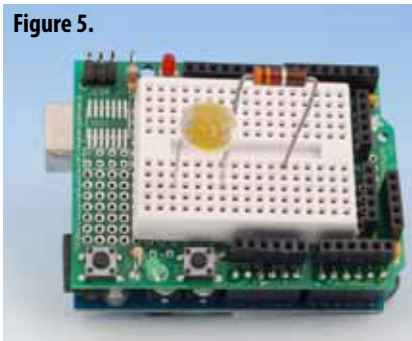


1. Install the cadmium sulfide photocell so that its two leads are in separate sets of holes (Figure 4).
2. Install the resistor so that one of its leads is in the same set as one of the leads from the photocell, and the other lead is in a set of holes by itself (Figure 5).

**Figure 4.**

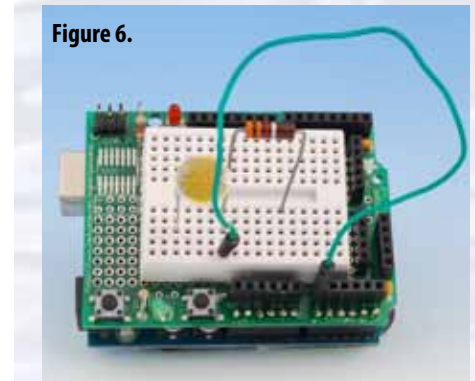


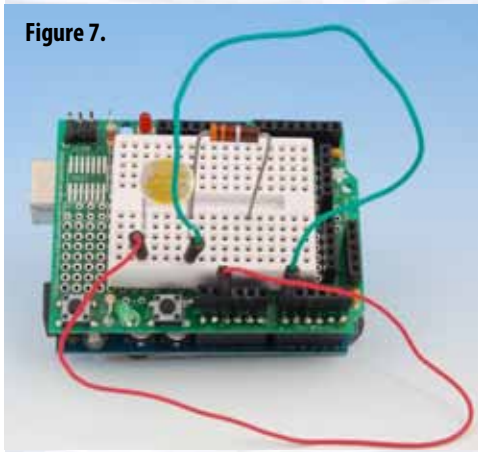
**Figure 5.**



3. Connect the set of holes that has a lead from the photocell AND a lead from the resistor to the socket for Analog pin 0 from the Arduino board. (Figure 6).

**Figure 6.**





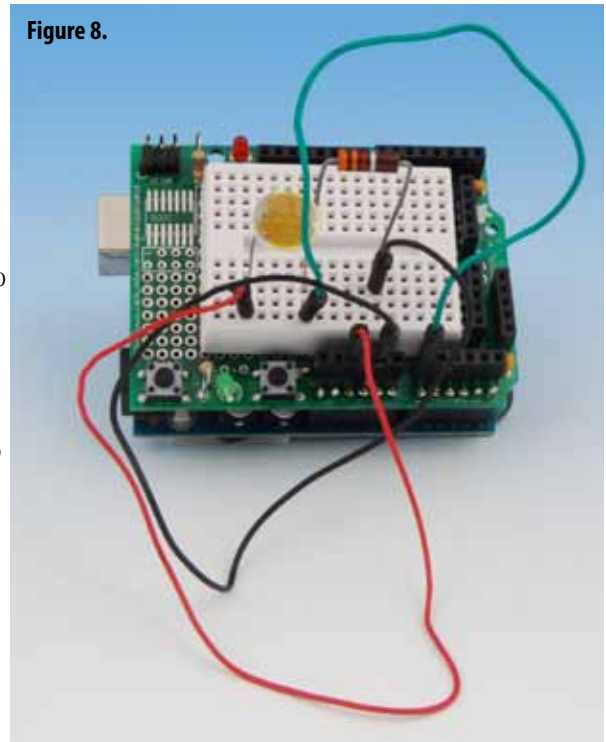
**Figure 7.**

4. Connect the other lead from the photocell to the 5V socket on the prototyping board. (Figure 7)

5. Connect the other lead from the resistor to the the Gnd socket on the prototyping board. (Figure 8).

6. Connect the Arduino board to your computer with a USB cable.

7. Launch the Arduino software. Select “New” from the “File” menu. Select the folder in which you want your program to be stored, and name your program “Servo” or or whatever name you choose.



**Figure 8.**

8. Type the following program into the Arduino Sketch Editor (the main window of the Arduino software):

```
//Program to read photocell //Line 1
void setup() //Line 1
{
  Serial.begin(9600); //Line 2
}
void loop() //Line 3
{
  Serial.print('Analog reading = '); //Line 4
  Serial.println(analogRead(0)); //Line 5
  delay(2000); //Line 6
}
```

9. Click on the “Verify” button. If everything is correct, the message “Done compiling” will appear at the bottom of the window. If an error message appears, carefully check your typing and be sure your program is identical to the one listed above.

10. Click on the “Upload” button (second from right) to send your program to the Arduino microcontroller. Two small LEDs built into the board will flash, and in a few seconds, the “Done uploading” message will appear in the bottom window. Your program should now be running, reading the voltage at the junction of the photocell and resistor, and printing the reading to the serial monitor every two seconds. To see the serial monitor, click on the serial monitor button (on the

right) at the top of the Sketch Editor window. You should see a series of readings. Hold your hand over the photocell, and the reading should decrease. Increase the light falling on the photocell and the reading should increase.

11. **Challenge:** Are photocells equally sensitive to all colors of light? Use the colored filters to find out.
12. For more things you can do with photocells and microcontrollers, see <http://www.ladyada.net/learn/sensors/cds.html>.
13. Here's what the lines in the program mean:  
You should already know about //, semicolons, and curly braces. If not, review the *Getting Control with Microcontrollers* activity.

Line 1 says that the next block of code will be a function named `setup()`.

Line 2 sets the data rate in bits per second (baud) for serial data transmission. In this case it is set to 9600 baud. Some computers may need lower rates, or may accept higher rates. You can check the documentation for your computer's USB port to find out.

Line 3 says that the next block of code will be a function named `loop()`.

Line 4 sends a stream of text to the serial port. The content of the text is defined by the characters between the single quote marks in parentheses.

Line 5 sends the value read at analog pin 0 to the serial port, followed by a carriage return.

Line 6 tells the microcontroller to wait 2,000 milliseconds (2 seconds) before repeating the loop.

## Resources

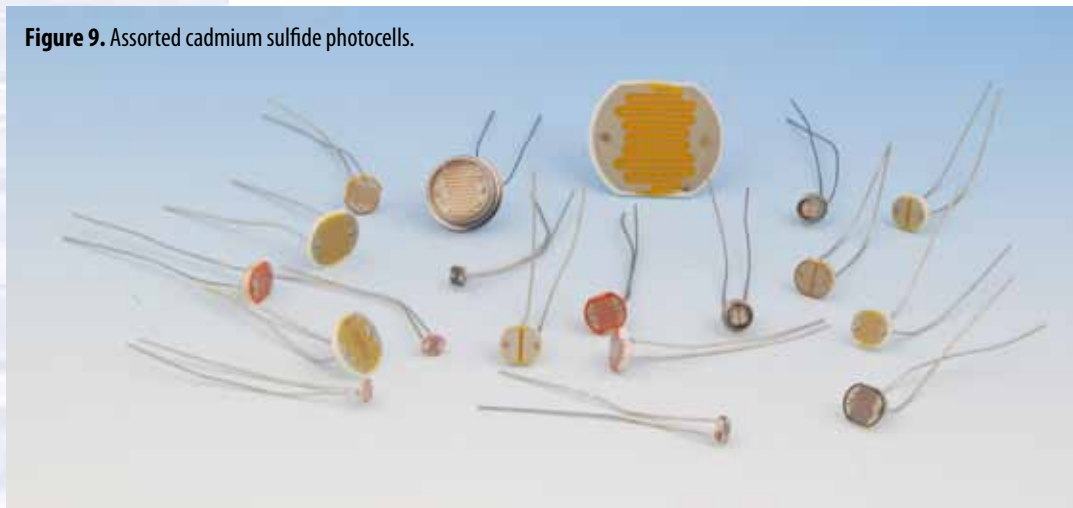
- Anonymous. Vernier Software and Technology [Internet]. Available from: <http://www.vernier.com/>; Web site describing sensors and interfaces for measuring a wide variety of environmental parameters; includes lesson plans and activities
- Fried, L. Photocells [Internet]. Adafruit Industries. Available from: <http://www.ladyada.net/learn/sensors/cds.html>; one of several tutorials on microcontrollers and electronics
- Palmisano, J. S. Schematics - Photoresistor [Internet]. Available from: [http://www.societyofrobots.com/schematics\\_photoresistor.shtml](http://www.societyofrobots.com/schematics_photoresistor.shtml); directions for working with photocells



### Notes to Educators

1. Most photocells are not equally sensitive to all wavelengths of light. Students should find that photocells are more sensitive to red and green light than to blue light.
2. For Arduino users: Cadmium sulfide photocells come in many shapes and sizes, and most will work for this activity. Figure 9 shows an assortment that was purchased for a few dollars from an electronic supply store. The optimum size for the resistor in the circuit shown in Figure 1 depends upon the characteristics of the specific photocell used for the circuit. There are two ways to find the resistor size:
  - (a) If you have a multimeter, set it to measure resistance (ohms) and measure the resistance of the photocell when it is fully illuminated (exposed to the brightest light available), and measure the resistance again when the photocell is completely shielded from light. Multiply the two resistance measurements, and take the square root of the result. This is the correct value for the resistor.
  - (b) If you do not have a multimeter, buy resistors in the following sizes: 1,000 ohm; 2,200 ohm; 3,300 ohm; 4,700 ohm, 6,800 ohm, and 10,000 ohm. Starting with the 10,000 ohm resistor, try different resistors in the circuit until a satisfactory result is obtained. “Satisfactory” means that the photocell does

**Figure 9.** Assorted cadmium sulfide photocells.



not saturate (readings stop changing) at high or low light levels. What you want is for every different light intensity to produce a unique number in the microcontroller output.