# Arduino Robots

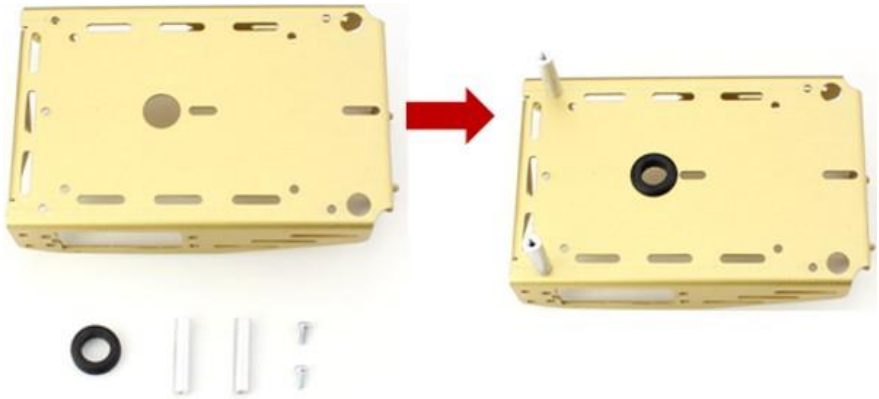
| Robot Kit Parts List | |
|---|---|
| (1) Metal Chassis | (2) Push Button Activators |
| (2) Servo Motors w/ Cross Wheels | (2) IR Receivers |
| (1) Control Board | (1) Piezo Speaker |
| (1) Dual-Sided Screwdriver | (1) Cotter Pin |
| (2) Large Plastic Wheels | (1) Rubber O-Ring/Grommet |
| (1) Plastic Ball | (2) LED Holders *(2 pieces of black plastic)* |
| (4) Rubber Bands | (2) 3 Prong Plug |
| (2) Wire Sensor Arms | (1) Battery Holder |
| Jumper Wire Set:<br>6 – Blue<br>6 – Black<br>4 – Yellow<br>4 – Red | Screws:<br>5 – Long Phillips Screws<br>8 – Medium Phillips Screws<br>8 – Short Phillips Screws<br>2 – Flat Head Phillips Screws |
| Resistors: Color Code:<br>8 – 220 Ω R,R,B,G<br>4 – 470 Ω Y,P,B,G<br>2 – 1K Ω B,BL,R,G<br>2 – 2K Ω R,BL,R,G<br>2 – 4.7K Ω Y,P,R,G<br>4 – 10K Ω B,BL,O,G | Rods & Spacers:<br>4 – Long Aluminum Extension Rods<br>2 – Short Aluminum Extension Rods<br>3 – White Spaces<br><br>Nuts & Washers:<br>10 – Aluminum Nuts |

# Robot

## ★ On the Topside of the Chassis

**Parts List:**

(1) robot chassis
(2) 1" standoffs (removed from BOE Shield)
(2) pan-head screws, 1/4" 4-40 (removed from BOE Shield)
(1) rubber grommet, 13/32"



**Instructions:**

- Remove the 1" aluminum standoffs from the BOE Shield, and save the standoffs and screws.
- Insert the 13/32" rubber grommet into the hole in the center of the chassis.
- Make sure the groove in the outer edge of the rubber grommet is seated on the metal edge of the hole.
- Use two 1/4" of the 4-40 screws to attach two of the standoffs to the top front of the chassis as shown.
- Save the other two standoffs and screws for a later step.

## ★ Mount the Servos on the Chassis

**Parts List:**

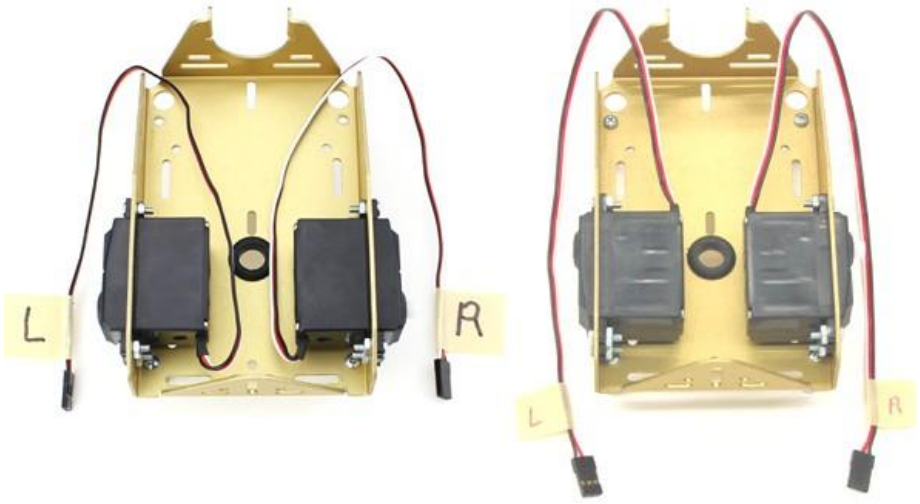
(2) BOE Shield-Bot Chassis, partially assembled.
(2) Parallax continuous rotation servos
(8) pan Head Screws, 3/8" 4-40
(8) nuts, 4-40
masking tape
pen

**Instructions:**

- Decide how you want to mount your servos from the two options described and pictured below.

1. **Outside-forward (left) —** the servos' mounting tabs seat outside the chassis, with their potentiometer access ports facing toward the front of the chassis. This allows easy access to adjust the potentiometer on an assembled robot, and also makes servo replacement quick. However, this gives the BOE Shield-Bot a longer, wider wheel base, so it will be a little less nimble on maneuvers and may need more pulses to make turns.

2. **Inside-backward (right) —** the servos' mounting tabs seat inside the chassis, with their potentiometer access ports facing towards the battery pack. This positions the axles close to the center of the BOE Shield-Bot, for maximum agility. If you are diligent about centering your servos before building your BOE Shield-Bot, this causes no problems.

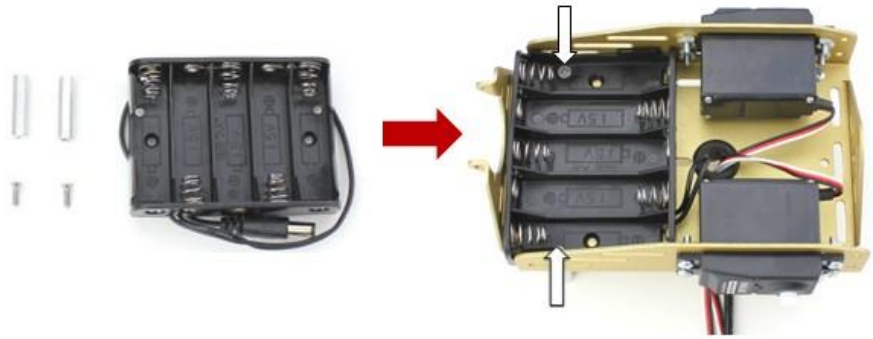



- Attach the servos to the chassis using the Phillips screws and nuts.
- Use pieces of masking tape to label the servos left (L) and right (R), as shown.

## 5-Cell Battery Pack

**Parts List:**

(2) flat-head Phillips screws, 3/8" 4-40
(2) 1" standoffs (removed from BOE Shield previously)
(1) 5-cell battery pack with 2.1 mm center-positive plug



**Instructions:**

- Place the empty battery pack inside the chassis positioned as shown above.
- Insert the two flat-head screws through the inside of the battery pack. Use the smaller set of holes that line up with the chassis mounting holes for the front standoffs, shown by the arrows.
- From the top of the chassis, thread a 1" standoff on each screw and tighten.

## Mounting the Wheels

**Parts List:**

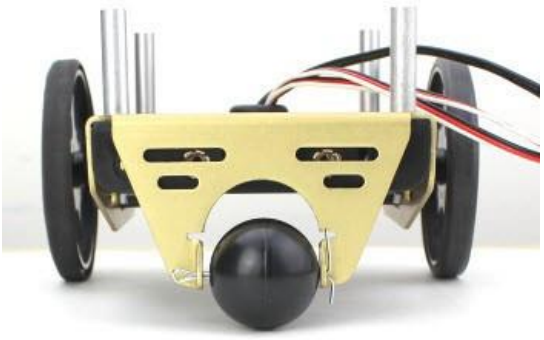
(1) 1/16" cotter pin
(1) tail wheel ball
(2) rubber band tires
(2) plastic machined wheels
(2) screws saved when removing the servo horns

**Your wheels and tires may look different - that's okay!**

The style shown above is being phased out. The new ones have finer spokes and O-ring tires. Both styles work just fine.

**Instructions:**

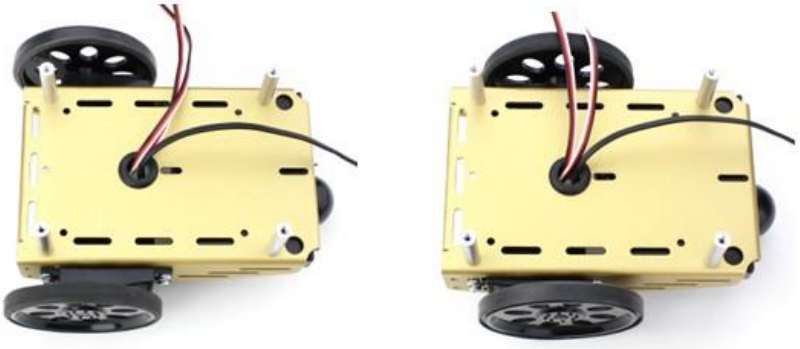
The robot's tail wheel is merely a plastic ball with a hole through the center. A cotter pin holds it to the chassis and functions as an axle for the wheel.



- Line up the hole in the tail wheel with the holes in the tail portion of the chassis.
- Run the cotter pin through all three holes (chassis left tail wheel, chassis right).
- Bend the ends of the cotter pin apart so that it can't slide back out of the hole.
- Press each plastic wheel onto a servo output shaft, making sure the shaft lines up with, and sinks into, the wheel's recess, then secure with the saved servo screws.
- Stretch each rubber band tire and seat it on the outer edge of each wheel.

When you are done, your completed chassis will look like one of the pictures below.

**Left: "Outside-forward" servos** **Right: "Inside-backward" servos**



## Mounting the Shield

The Board of Education Shield makes it easy to build circuits and connect servos to the Arduino module. In this chapter, you will use it to test servos and indicator lights. Next chapter, you'll mount the BOE Shield and servos on a robot chassis to build a robot we'll call the BOE Shield-Bot.

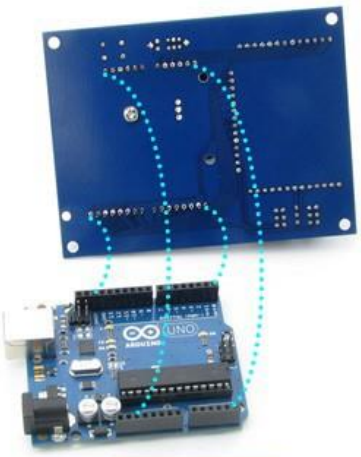
**Parts List:**

(1) Arduino module
(1) Board of Education Shield
(4) 1" round aluminum standoffs
(4) pan head screws, 1/4" 4-40
(3) 1/2" round nylon standoffs
(3) nylon nuts, 4-40
(3) pan head screws, 7/8", 4-40

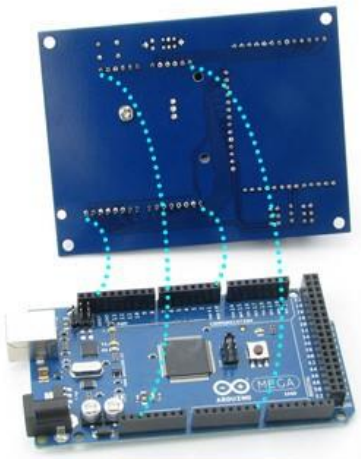
**Instructions:**

The four groups of pins under the Board of Education Shield plug into the four Arduino socket headers. There are also three board connection holes in the shield that line up with holes in the Arduino module, designed to connect the two boards together with screws and nylon standoffs.

If you have a revision 3 Arduino, it will be labeled UNO R3 or MEGA R3 on the back. R3 boards will have two empty pairs of sockets, closest to the USB and power connectors, after socketing the shield. Earlier versions, such as 2, 1, and Duemilanove, have the same number of sockets as the shield has pins, so there will be no empty sockets left over. If you have an Arduino Mega, the four pin groups will fit into the four headers closest to the USB and power connectors, as shown in the box below.



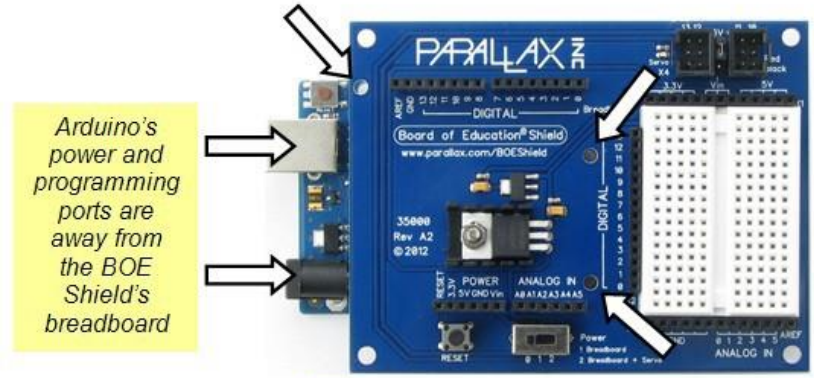
Uno R3



Mega R3

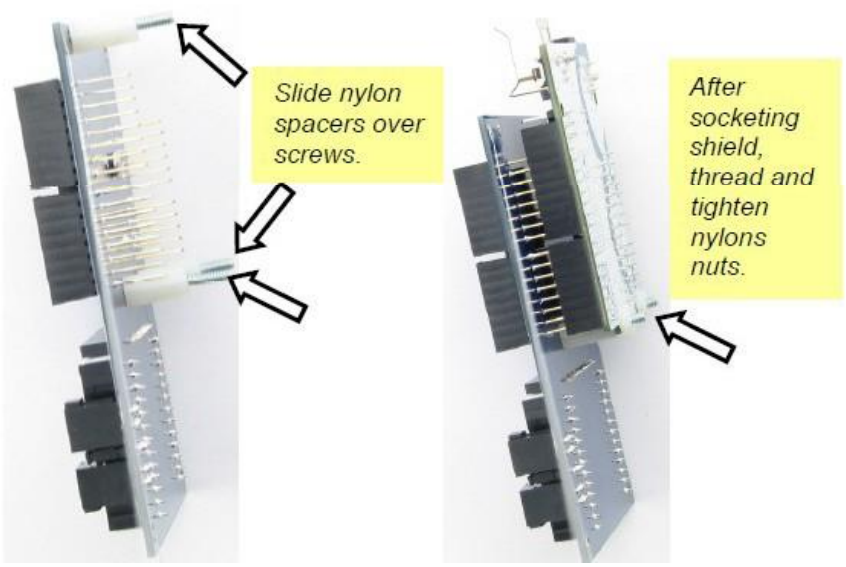
- Disconnect the programming cable from your Arduino module.
- Look closely at your Arduino module and the pins on the Board of Education Shield to see how the sockets and pins will line up for your particular boards. Note that if you have an Arduino Mega, its USB port and power jack will be close to the edge of the shield, like the image on the right.

Component placement varies a little bit for the different Arduino models; some can only fit one or two nylon standoffs for holding the boards together. This is okay, but you need to find out which holes you can use before socketing the Board of Education Shield.

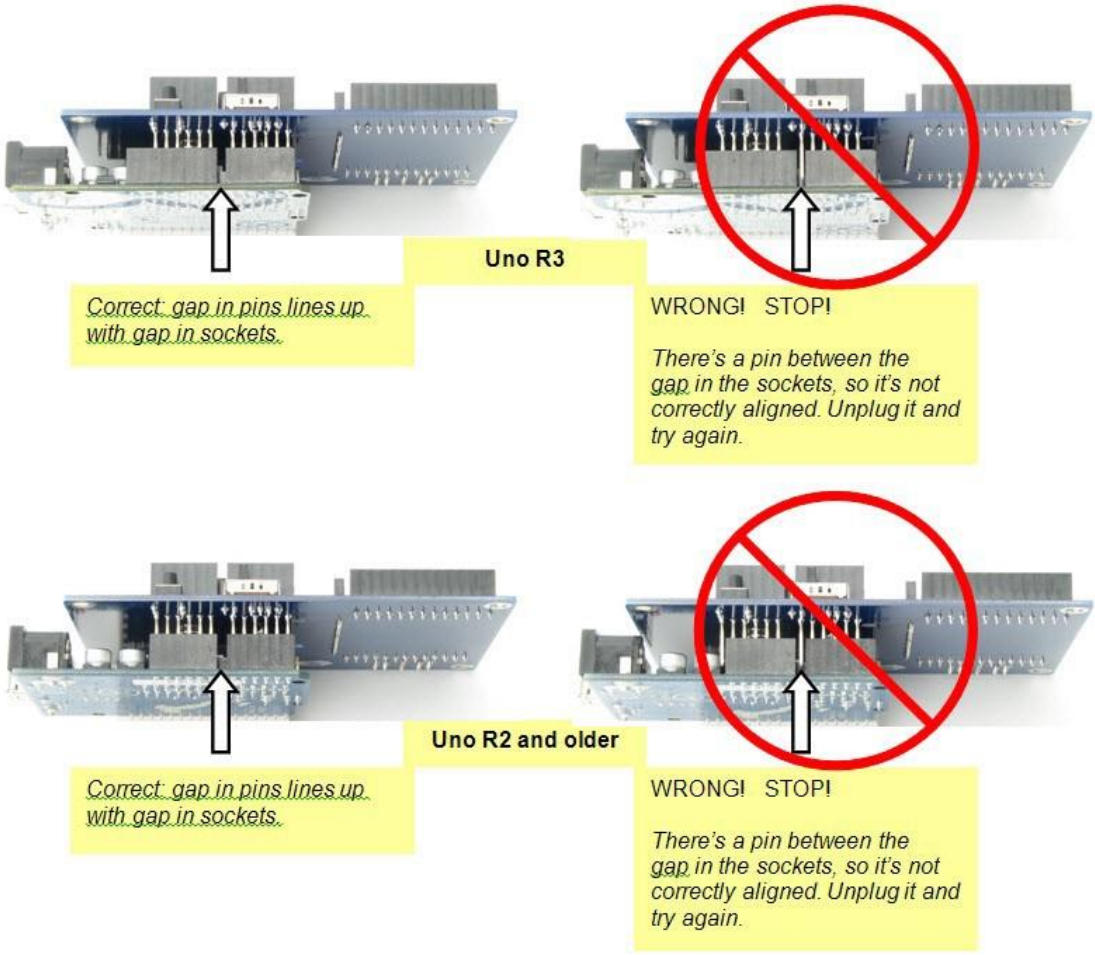



Make sure the BOE Shield mounting holes align with the Arduino's.

- Hold a nylon spacer over each mounting hole on your Arduino module, and look through it to see if the spacer can line up with the hole completely.
- For each hole that works on your Arduino module, insert a 7/8" screw through the corresponding board-connection hole in your Board of Education Shield.



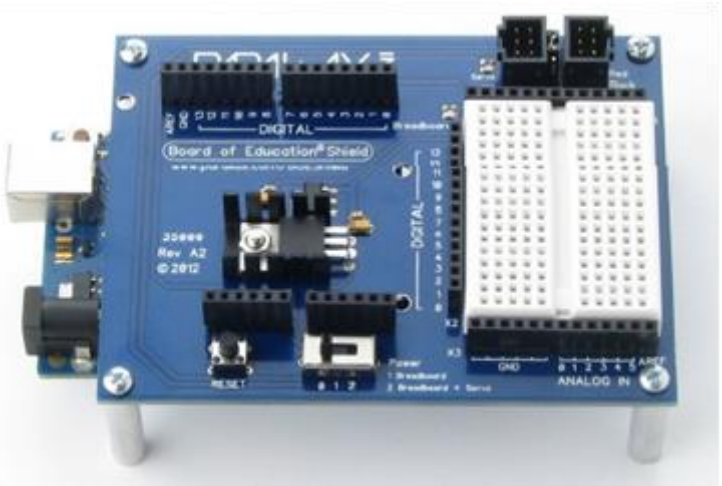

- Slide a nylon spacer over each screw you used.
- Line up the Arduino module's sockets with the Board of Education Shield's pins.
- Also line up the 7/8" screws with the mounting holes in the Arduino board.
- Gently press the two boards together until the pins are firmly seated in their sockets. The sockets will not cover the pins completely; there will be about 3/8" (~5 mm) of the pins still exposed between the bottom of the shield and the top of the sockets.
- Check to make ABSOLUTELY SURE your pins are seated in the sockets correctly. It is possible to misalign the pins, which can damage your board when it is powered.




- Thread a nylon nut over each screw, and tighten gently.

To keep the connected boards up off of the table, we'll mount tabletop standoffs to each corner of the Board of Education Shield.

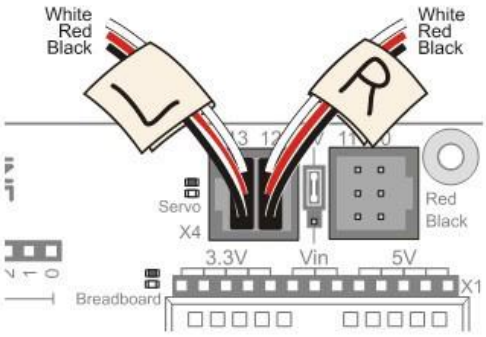
- Thread a 1/4" screw through a corner hole on the Board of Education Shield from the top side.
- Thread a 1" aluminum standoff onto the screw and tighten gently.
- Repeat until all four standoffs are installed.



## Attach the BOE Shield to the Chassis

**Parts List**
(4) pan-head screws, 1/4" 4-40
(1) Board of Education Shield (mounted in next step)

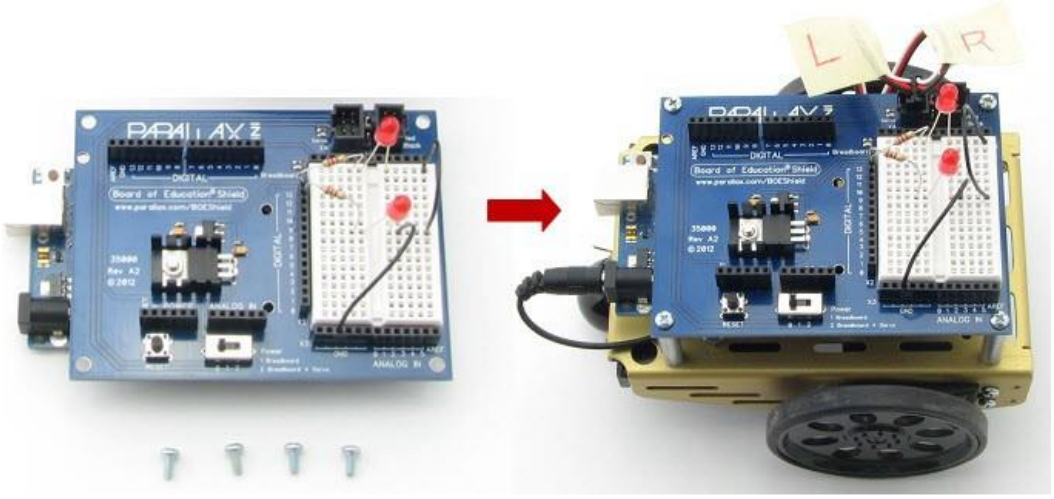
- Set the BOE Shield on the four standoffs, lining them up with the four mounting holes on the outer corner of the board.
- Make sure the white breadboard is closer to the drive wheels, not the tail wheel.
- Attach the board to the standoffs with the pan head screws.
- Reconnect the servos to the servo headers.




**Using Different Pins for the Servos**

The Arduino toggles Pin 13 briefly upon startup or reset. If this causes problems for a particular application, you can use Pins 11 and 12 instead of 12 and 13. Be sure to adjust your code accordingly.

If you are building the BOE Shield-Bot to use with ROBOTC instead of for this tutorial, follow these instructions for using different servo ports.



- From the underside of the chassis, pull any excess servo and battery cable through the rubber grommet hole, and tuck the excess cable lengths between the servos and the chassis.

From <http://learn.parallax.com/node/199>

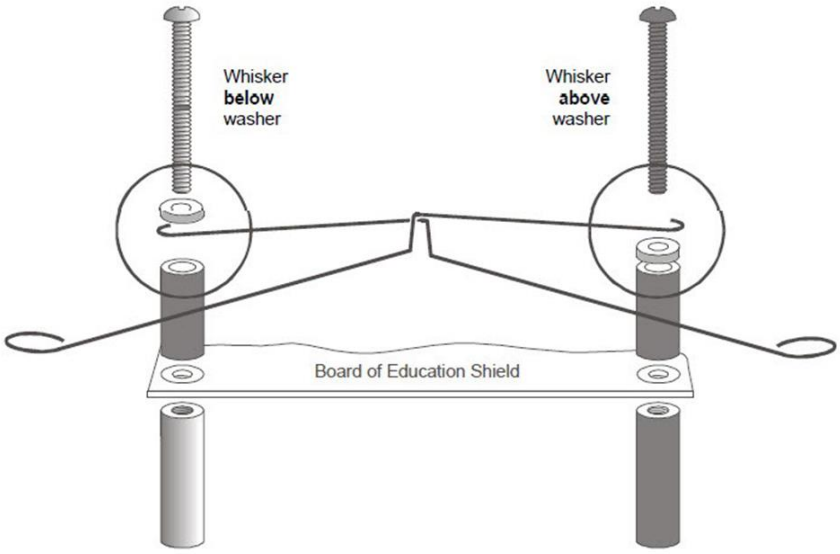
# "Whiskers" Mechanical Switch

- ***Disconnect power from your board.***

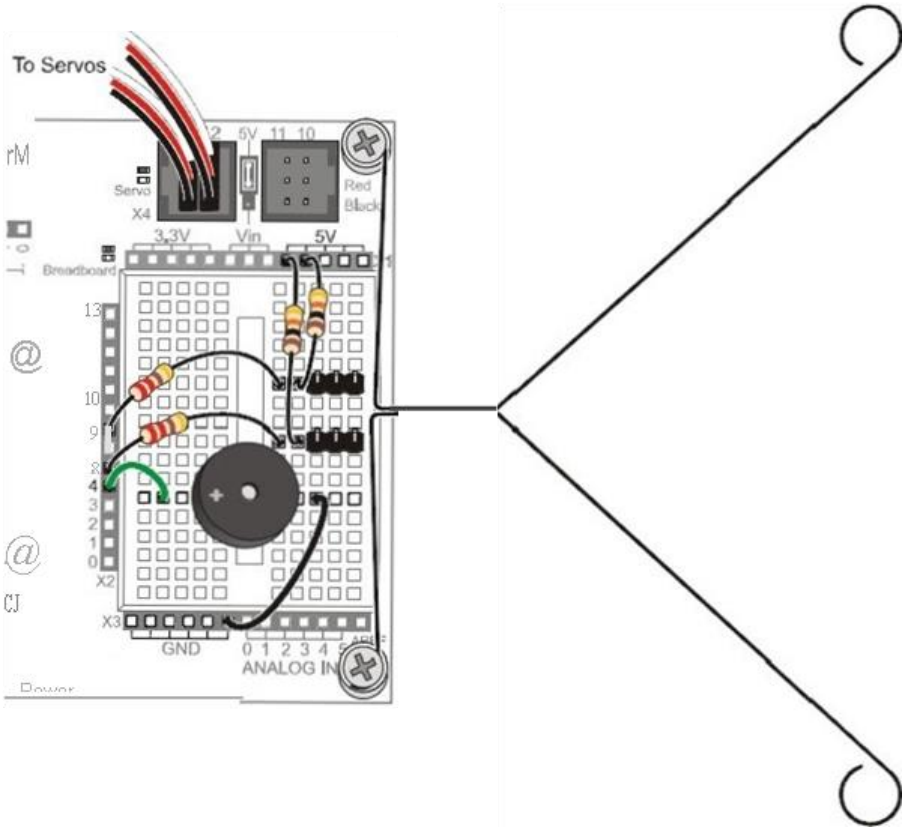
**Parts List:**
(2) whisker wires
(2) 7/8" pan head 4-40 Phillips screws
(2) ½" round spacer
(2) nylon washers, size #4
(2) 3-pin m/m headers
(2) resistors, 220 Ω (red-red-brown)
(2) resistors, 10 kΩ (brown-black-orange)

**Building the Whiskers**

- Remove the two front screws that hold your board to the front standoffs.
- Thread a nylon washer and then a ½" round spacer on each of the 7/8" screws.
- Attach the screws through the holes in your board and into the standoffs below, but do not tighten them all the way yet.
- Slip the hooked ends of the whisker wires around the screws, one above a washer and the other below a washer, positioning them so they cross over each other without touching.
- Tighten the screws into the standoffs.



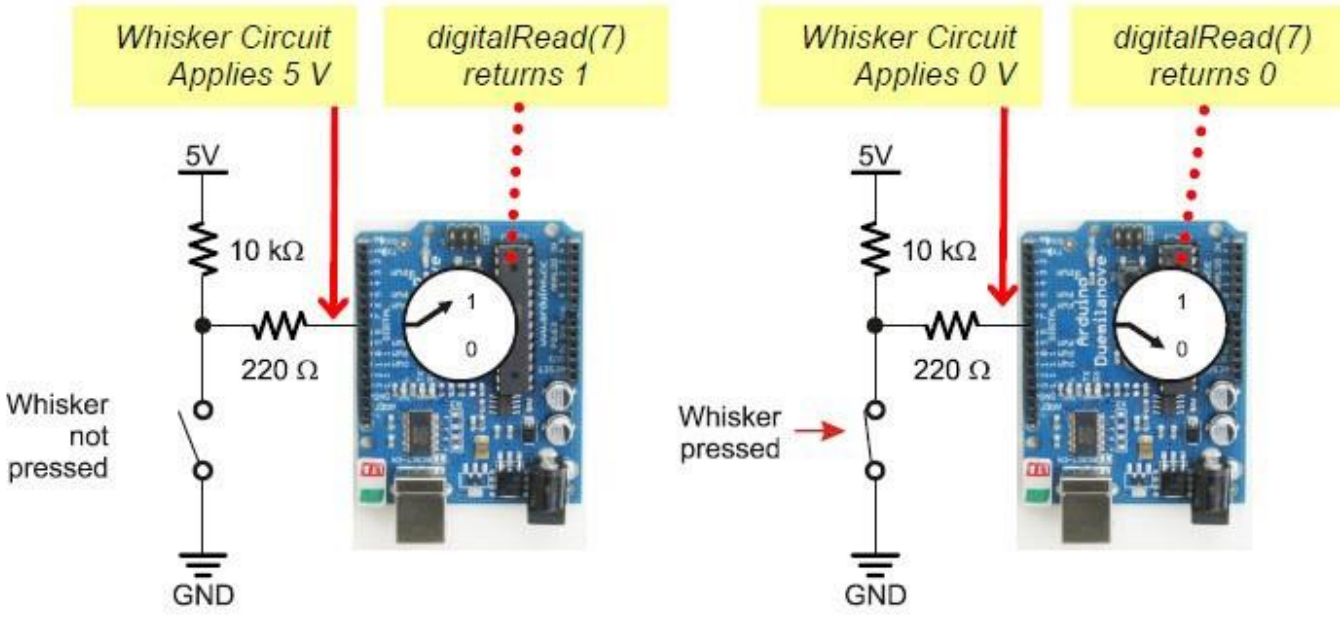

- Use the 220 Ω resistors (red-red-brown) to connect digital pins 5 and 7 to their corresponding 3-pin headers.
- Use the 10 kΩ resistors (brown-black-orange) to connect 5 V to each 3-pin header.
- Make sure to adjust each whisker so that it is close to, but not touching, the 3-pin header on the breadboard. A distance of about 1/8" (3 mm) is about right.
- Make note of your sensor inputs.



**How they Work**

The whiskers are connected to ground because the plated holes at the outer edge of the board are all connected to ground. The metal standoffs and screws provide the electrical connection to each whisker.

Since each whisker is connected to digital I/O, the Arduino can be programmed to detect which voltage is applied to each circuit, 5 V or 0 V. Take a look at the figure below. On the left, the circuit applies 5 V when the whisker is not pressed, so a digital read returns 1 (**HIGH**). On the right, the circuit applies 0 V when the whisker is pressed, so a digital read returns 0 (**LOW**).




**Switch Lingo:** Each whisker is both the mechanical extension and the ground electrical connection of a *normally open* (off until pressed) *momentary* (on only while pressed) *single-pole* (one set of electrical contact points), *single-throw* (only one position conducts) switch.

From <http://learn.parallax.com/node/237>

# Robot Movement

## Basic Movement

In this activity, you are going to test the basic maneuvering of your robot.

1. Build the basic robot setup without sensors, see Robot instructions at the beginning of this document.
2. Upload the Arduino program code called **Test Moving Robot** from Canvas Week 6 module.
3. Disconnect your robot from your laptop, and turn the switch on to start it moving. Be sure you have batteries.

## Calibrate Servos

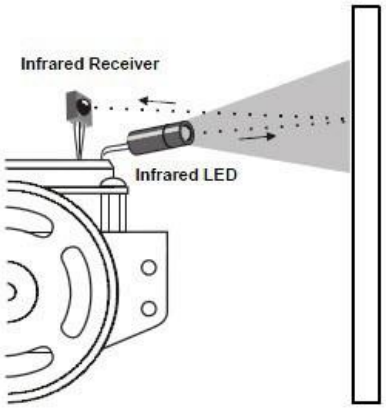
Now, let's calibrate your servos.

1. Modify the movement program from above and write a stop to the servos. Do they move or quiver? If so, adjust the servo until they stop by gently turning the screw on the back.

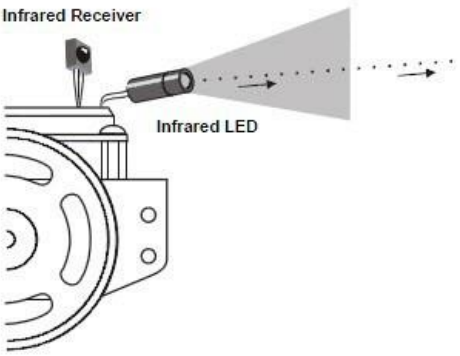

2. Now modify your program to drive the robot forward for a few seconds and then backward.
3. Measure and note your robots movements.
4. Try using an array(s) for movements.

# IR Sensors

Each headlight is an infrared LED inside a tube that directs the light forward, just like a flashlight. Each eye is an infrared receiver that sends the Arduino high/low signals to indicate whether it detects the infrared LED's light reflected off an object.




*Infrared reflected, obstacle detected.*




*Infrared not reflected, no obstacle detected.*

From <http://learn.parallax.com/node/299>

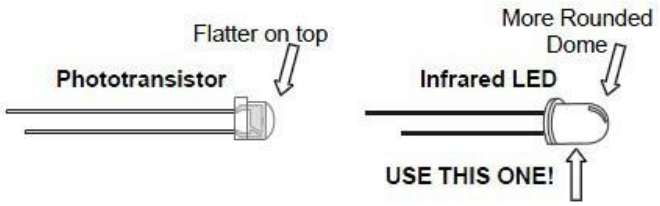
# Building the IR Sensors

**Parts List**

(2) IR receivers
(2) IR LEDs (clear case)
(2) IR LED shield assemblies
(2) Resistors, 220 Ω (red-red-brown)
(2) Resistors, 2 kΩ (red-black-red)

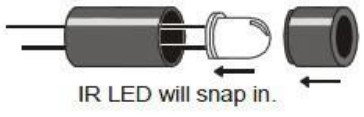
- Gather the parts in the Parts List, using the drawings below to help identify the infrared receivers, LEDs, and shield assembly parts.

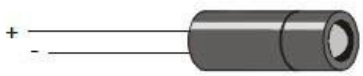
- Check the figure below to make sure you have selected infrared LEDs and not phototransistors. The infrared LED has a taller and more rounded plastic dome, and is shown on the right side of this drawing.




**Assemble the IR Headlights**

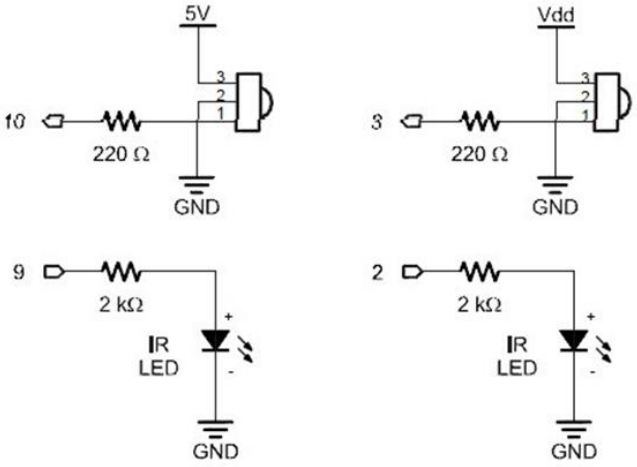
- Insert the infrared LED into the LED standoff (the larger of the two pieces) as shown below.
- Make sure the IR LED snaps into the LED standoff.
- Slip the short tube over the IR LED's clear plastic case. The ring on one end of the tube should fit right into the LED standoff.
- Use a small piece of clear tape to secure the two pieces so the shield assembly doesn't separate during use.

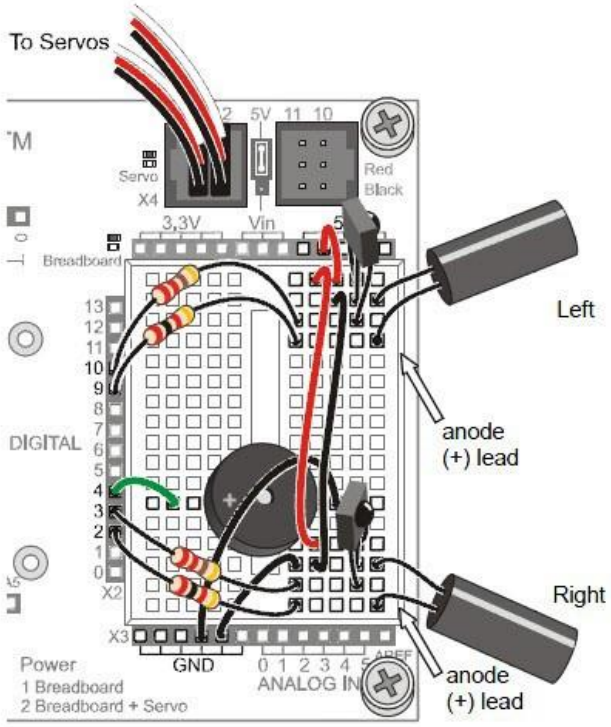


**IR Object Detection Circuit**
The next figures show the IR object detection schematic wiring diagram. One IR object detector (IR LED and receiver pair) is mounted on each corner of the breadboard closest to the very front of the BOE Shield-Bot.

- Disconnect the power and programming cables.
- Build the circuit in the schematic below, using the wiring diagram as a reference for parts placement.
- Note that the anode lead of each IR LED connects to a 2 kΩ resistor. The cathode lead plugs into the same breadboard row as an IR detector's center pin, and that row is connected to GND with a jumper wire.




**Watch your IR LED anodes and cathodes!**
The anode lead is the longer lead on an IR LED by convention. The cathode lead is shorter and mounted in the plastic case closer to its flat spot. These are the same conventions as the red LEDs we have been using.

From <http://learn.parallax.com/node/301>

# Testing the IR Sensors

The infrared receivers are designed to detect infrared light (in the 980 nanometer range) flashing at a rate near 38 kHz.

Test the detection range with different colored objects. What colors and surfaces can it detect only at closest range? What colors and surfaces can it detect from farther away? Also, be sure to check your area for interference. Sunlight and florescent lights can cause false detections.

By making the infrared lights brighter, you can also increase its detection range. A smaller resistor allows more current to flow through an LED. More current through an LED is what causes it to glow more brightly. Examine the effect of different resistance values with both the red and infrared LEDs.

**Remember to disconnect power and the programming cable before you make changes to a circuit.**

Replace the 2 kΩ resistor that connects the IR LED anodes with 4.7 kΩ resistors. Determine the furthest distance at which the same sheet of paper (or something similar) is detected, and record your data. Repeat with 1 kΩ resistors, 470 Ω resistors, and 220 Ω resistors. (For the smaller resistor values, they may end up making the detectors so sensitive that they see the table surface in front of your robot. If that happens, set the robot on the edge of the table with the IR detectors pointing off the end and try the distance measurements again.)

| IR Series Resistance (Ω) | Maximum Detection Distance |
|---|---|
| 4700 | |
| 2000 | |
| 1000 | |
| 470 | |
| 220 | |

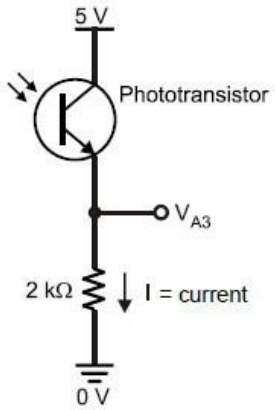
From <http://learn.parallax.com/node/302>

# Phototransistor

A resistor "resists" the flow of current. Voltage in a circuit with a resistor can be likened to water pressure. For a given amount of electric current, more voltage (pressure) is lost across a larger resistor than a smaller resistor that has the same amount of current passing through it. If you instead keep the resistance constant and vary the current, you can measure a larger voltage (pressure drop) across the same resistor with more current, or less voltage with less current.

The Arduino's analog inputs are invisible to the phototransistor circuit. So, A3 monitors the circuit but has no effect on it. Take a look at the circuit below. With 5 volts (5 V) at the top and GND (0 V) at the bottom of the circuit, 5 V of electrical pressure (voltage) makes the supply of electrons in the BOE Shield-Bot's batteries want to flow through it.

The reason the voltage at A3 (VA3) changes with light is because the phototransistor lets more current pass when more light shines on it, or less current pass with less light. That current, which is labeled I in the circuit below, also has to pass through the resistor. When more current passes through a resistor, the voltage across it will be higher. When less current passes, the voltage will be lower. Since one end of the resistor is tied to 0 V, the voltage at the VA3 end goes up with more current and down with less current.




If you replace the 2 kΩ resistor with a 1 kΩ resistor, VA3 will see smaller values for the same currents. In fact, it will take twice as much current to get VA3 to the same voltage level, which means the light will have to be twice as bright to reach the 3.5 V level.

So, a smaller resistor in series with the phototransistor makes the circuit less sensitive to light. If you instead replace the 2 kΩ resistor with a 10 kΩ resistor, VA3 will be 5 times larger with the same current, and it'll only take $1/5^{th}$ the light to generate $1/5^{th}$ the current to get VA3 past the 3.5 V level. So, a larger resistor makes the circuit more sensitive to light.

**Connected in Series**
When two or more elements are connected end-to-end, they are connected in series. The phototransistor and resistor in this circuit are connected in series.

From <http://learn.parallax.com/node/258>

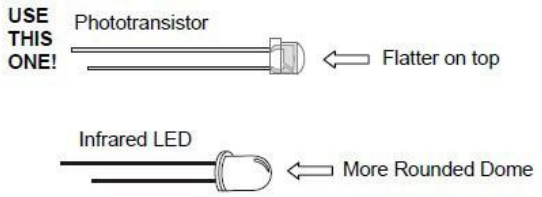
# Building the Phototransistor

**Parts List**
(1) phototransistor
(1) jumper wires
(1) resistor, 2 kΩ (red-black-red)
(1) incandescent or fluorescent flashlight or desk lamp

After some testing, and depending on the light condition in your robotics area, you might end up replacing the 2 kΩ resistor with one of these resistors, so keep them handy:

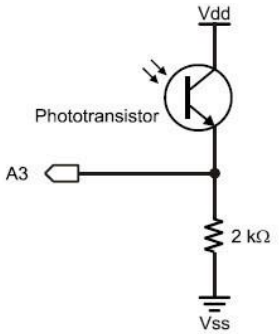
(1) resistor, 220 Ω (red-red-brown)
(1) resistor, 470 Ω (yellow-violet-brown)
(1) resistor, 1 kΩ (brown-black-red)
(1) resistor, 4.7 kΩ (yellow-violet-red)
(1) resistor, 10 kΩ (brown-black-orange)

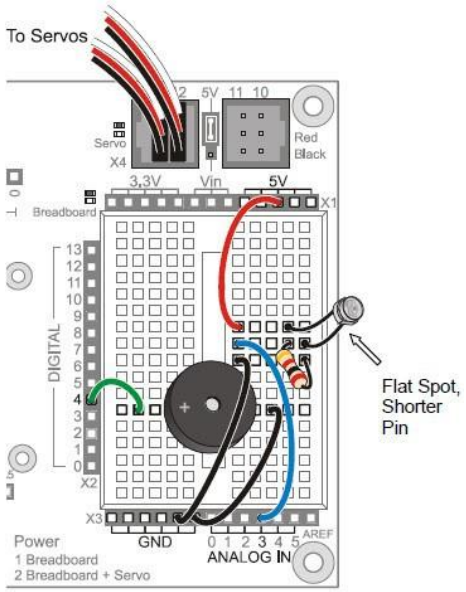
The drawing below will help you tell apart the phototransistor and infrared LED, since they look similar.




**Building the Bright Light Detector**
The schematic and wiring diagram below show the schematic and wiring diagram of a circuit very similar to the ones in streetlights that turn on automatically at night. The circuit outputs a voltage that varies depending on how much light shines on the phototransistor. The Arduino will monitor the voltage level with one of its analog input pins.

- Disconnect the battery pack and programming cable from your Arduino, and set the BOE Shield's switch to 0.
- Remove the whisker circuits, but leave the piezo speaker circuit in place.
- Build the circuit shown, using the 2 kΩ resistor.
- Double-check to make sure you connect the phototransistor's emitter lead (by the flat spot) to the resistor, and its collector to 5V.
- Also double-check that the phototransistor's leads are not touching each other.

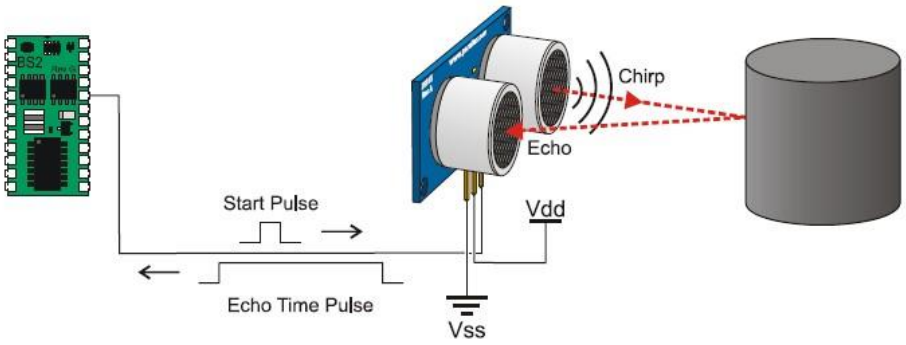



From <http://learn.parallax.com/node/255>

# Ultrasonic Sensor

http://www.parallax.com/product/28015



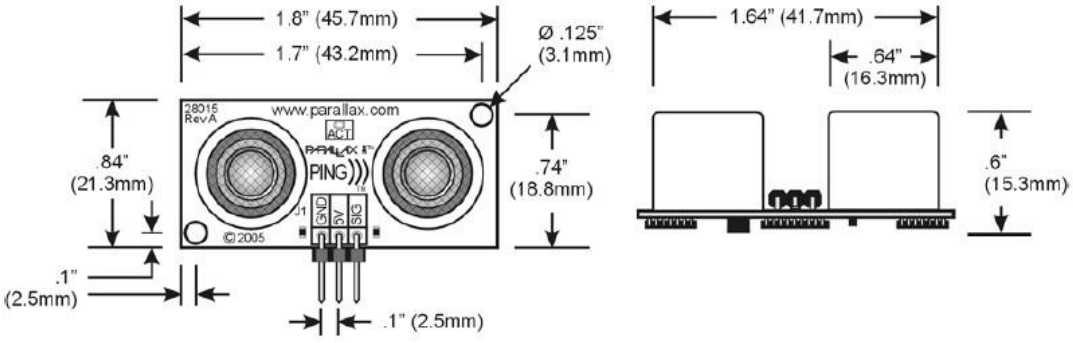

## Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse: positive TTL pulse, 115 µs minimum to 18.5 ms maximum.
- RoHS Compliant

## Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

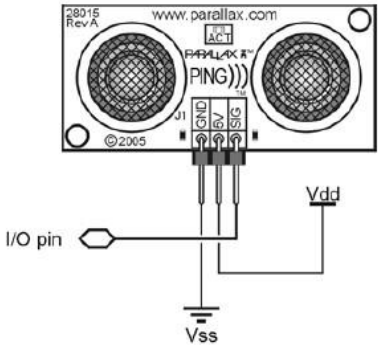
## Dimensions



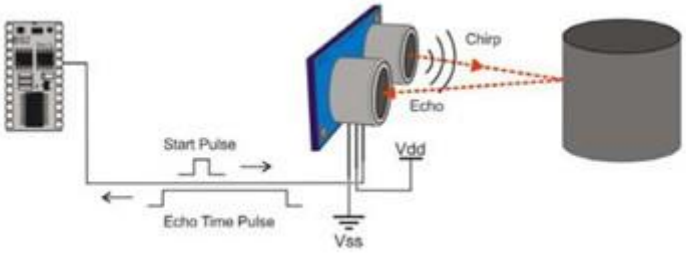
## Pin Definitions

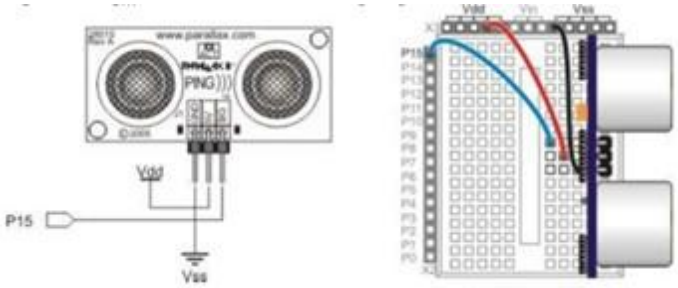
| | |
|---|---|
| GND | Ground (Vss) |
| 5 V | 5 VDC (Vdd) |
| SIG | Signal (I/O pin) |

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).




**Setting up the sensor:**

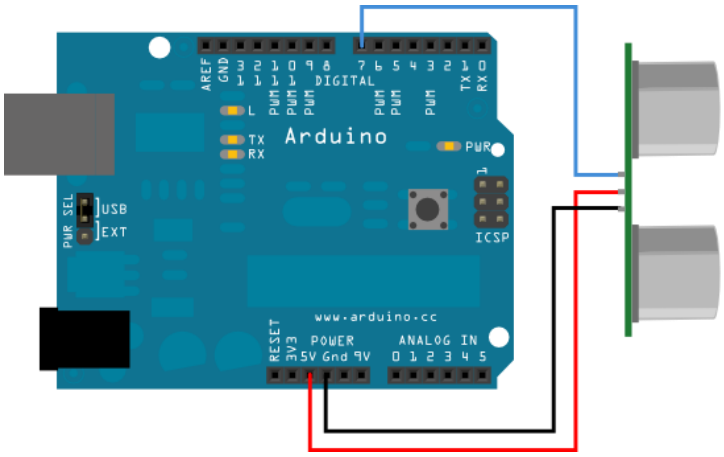


# Setup of the Ultrasonic Sensor

## Circuit

The 5V pin of the PING))) is connected to the 5V pin on the Arduino, the GND pin is connected to the GND pin, and the SIG (signal) pin is connected to digital pin 7 on the Arduino.




From <http://arduino.cc/en/Tutorial/Ping?from=Tutorial.UltrasoundSensor>