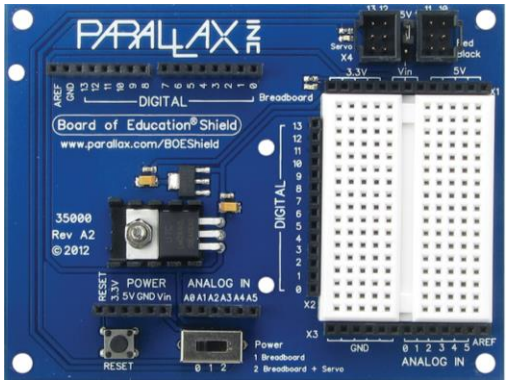
Arduino BOE kit and Raspibot Board

Class 3: Arduino BOE Shield Bot
Basic Navigation Techniques



August 9, 2017

Don Wilcher

Presented by:

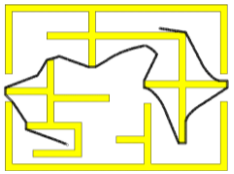
DesignNews CEC CONTINUING EDUCATION CENTER Digi-Key ELECTRONICS ROHDE&SCHWARZ

Class 3: Arduino BOE Shield Bot Basic Navigation Techniques

Agenda

- Mobile Robot Navigation
- Arduino BOE Shield bot sensors
- Start/Reset Indicator Circuit
- Hands-On Labs: Navigation Code Examples

Mobile Robot Navigation



Why is Mobile Robot Navigation important?

a) Avoiding collision

b) Detecting severe environmental conditions prior to moving

   i. temperature

   ii. radiation

   iii. weather conditions

c) Knowing present location.

Question 1

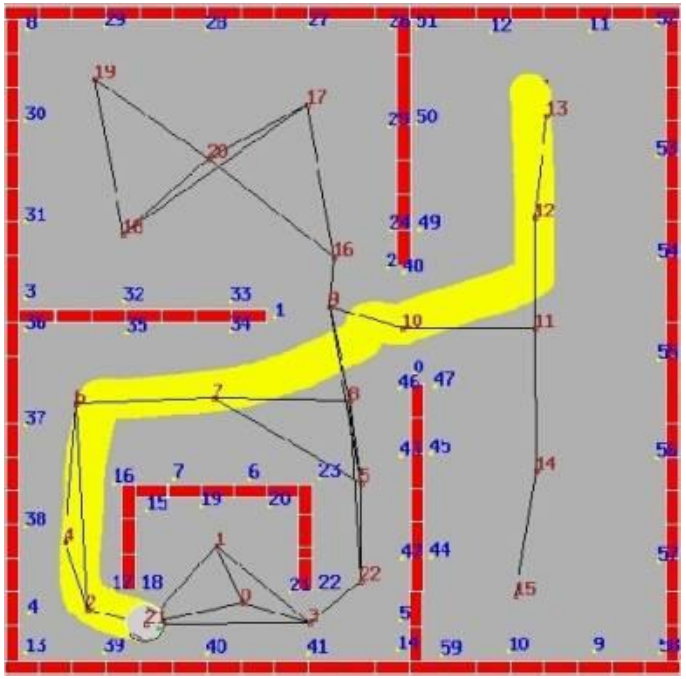
Why is a Mobile Robot Navigation important?

Mobile Robot Navigation…

What is Robot Navigation?

The robot's ability to determine its own position using sensors, software and establishing a reference so as to plan a path to a designated location.

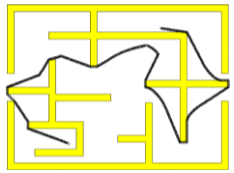
Mobile robot navigation based on panoramic cameras



Source:

http://www.ics.forth.gr/cvrl/index_main.php?l=e&c=32

Mobile Robot Navigation…



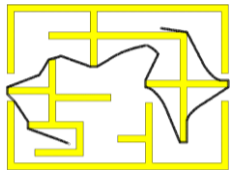
Navigation can be defined as the combination of three fundamental competences:

a) Self-localization – robot establishing its own position and orientation using a reference.

b) Path planning – robot determining its current position and organizing a new designated location through software.

c) Map-building and map interpretation – a map or visual representation for displaying spatial location.

Mobile Robot Navigation…



Types of navigation

a) Vision-based navigation – The use of optical sensors and computer vision algorithms. Also, Vision based navigation uses:

   i. Laser based range finder

   ii. Photometric cameras using CCD(Charged Coupled Device) arrays

Question 2

What are the three fundamental competencies of Navigation?

a) Self-localization, Path Planning, Map Building and Map Interpretation

b) Self-mobilization, Path Planning, Map Building and Map Interpretation

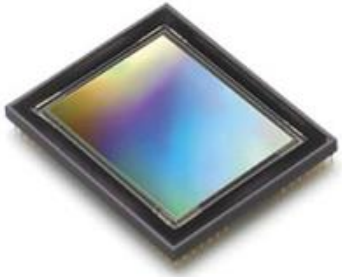
c) Self-localization, Path Manipulation, Map Building and Map Interpretation

d) None of the above

Mobile Robot Navigation…

Examples of optical components used in Vision based navigation

Photometric Camera

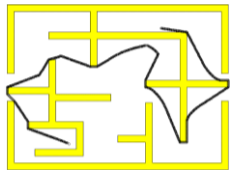


CCD Array

Laser Range Finder

Optical Sensor (Retroreflective)

Mobile Robot Navigation…



Types of navigation…

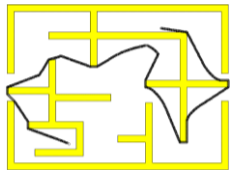
Goals of Vision-based navigation consist of:

a) representation of the environment -mapping

b) sensing models – Abstraction (concept) **models** trying to quantify how well **sensors** can sense physical phenomena at designated locations.

c) localization algorithms – Restriction of spatial location for navigation and mapping using a set of procedural steps.

Mobile Robot Navigation…



Types of navigation…

Indoor Navigation – method of navigating the mobile robot's path using direct guidance to a designated location.

a) inductive loops or magnets in the floor

b) paint or reflective material (lines) on the floor.

c) Beacons

d) Bar codes

e) Tactile or optical object detection (motor drive reversal control)

Question 3

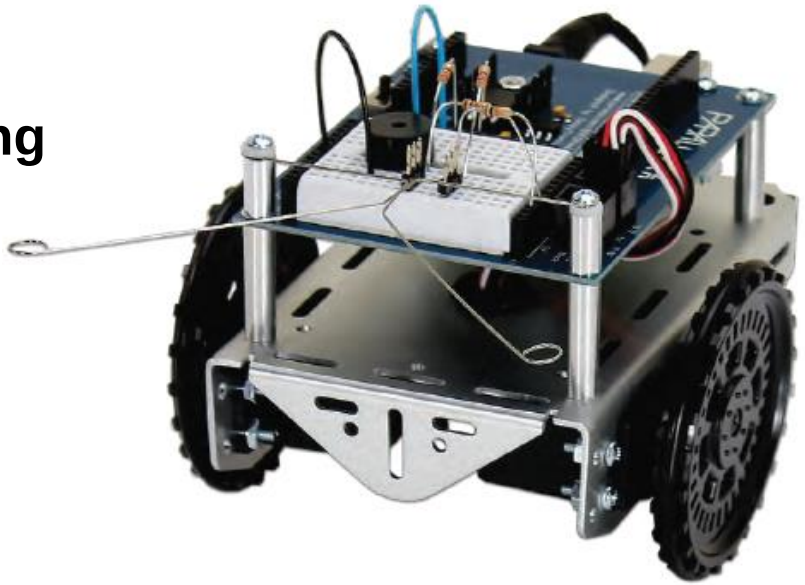
What are the Goals for Vision based Navigation?

Mobile Robot Navigation…

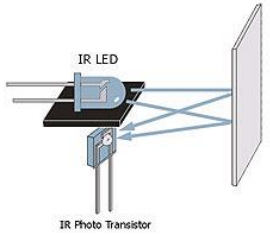
Types of navigation…

The Arduino BOE Shield bot uses Indoor navigation: tactile or object detection method of mobile robot navigating in a designated environment.

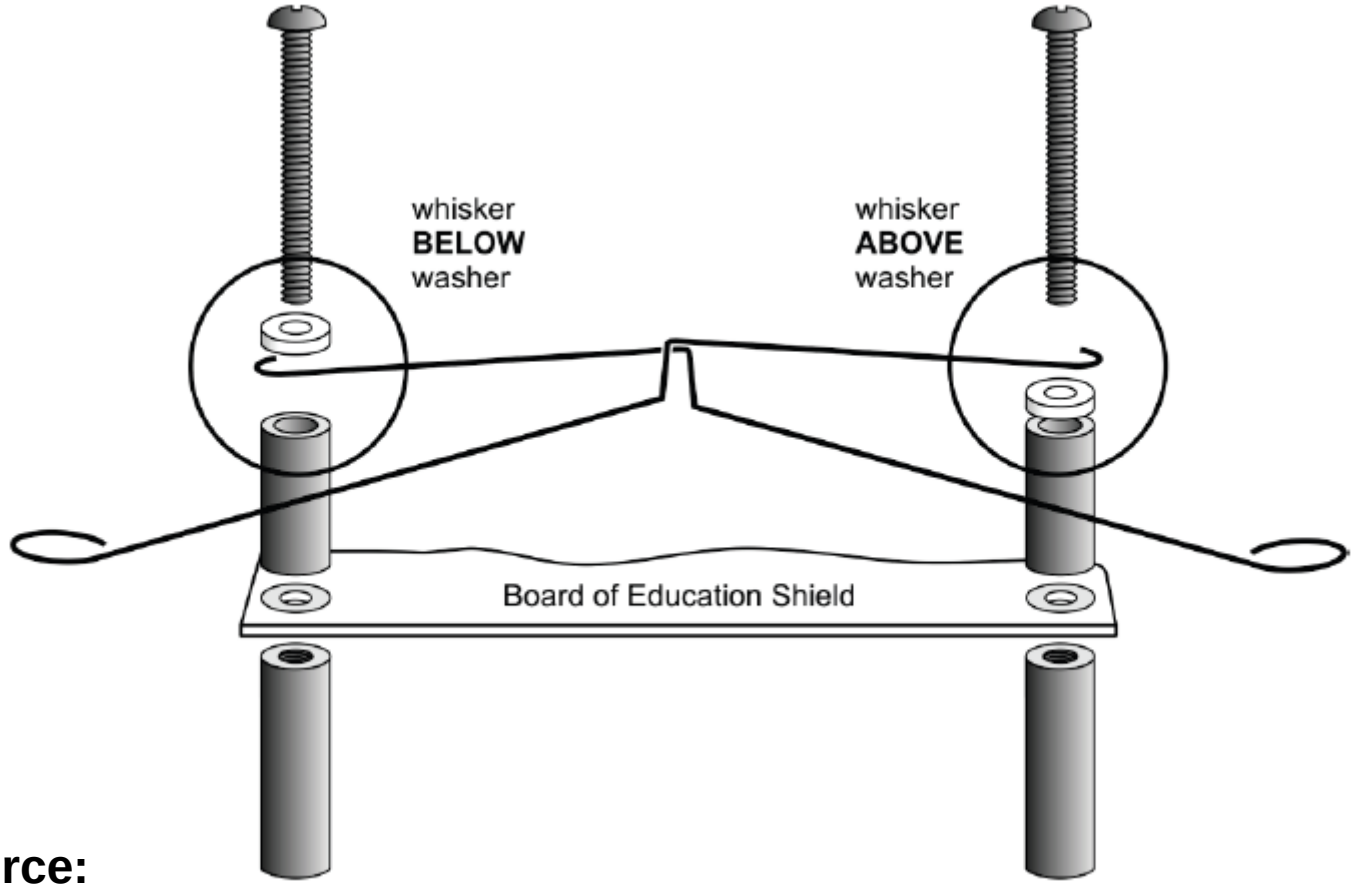
Tactile object detection using "whiskers"



Arduino BOE Shield bot Sensors…



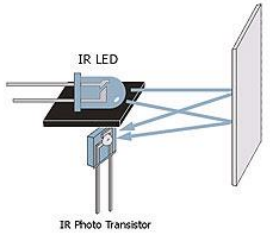
Tactile or Whiskers based Sensor (Assembly)



Book Source:

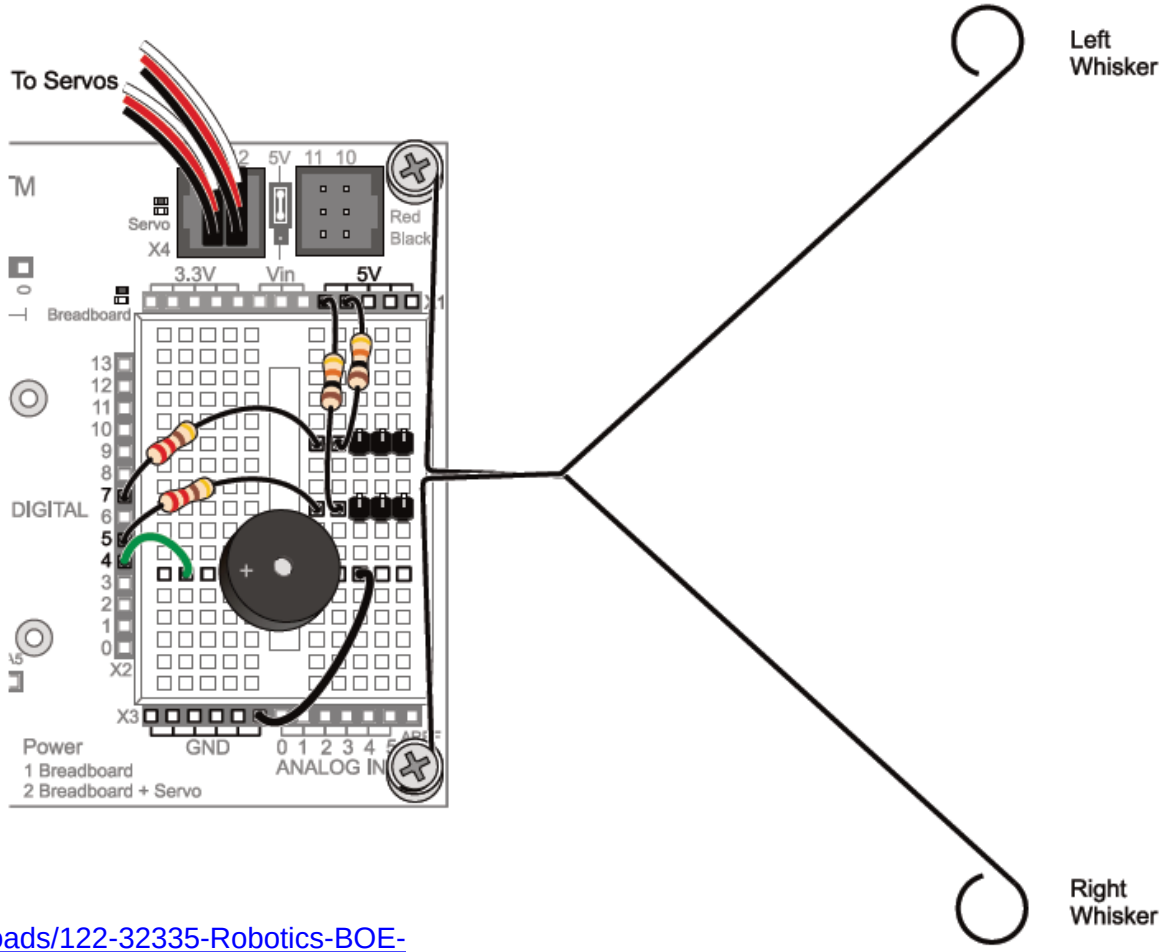
https://www.parallax.com/sites/default/files/downloads/122-32335-Robotics-BOE-Shield-Bot-Arduino-v1.0.pdf

Arduino BOE Shield bot Sensors…



Tactile or Whiskers based Sensor (Assembly)…

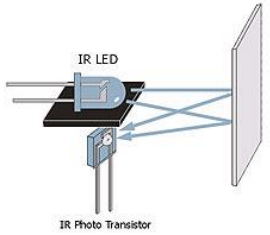
Sensor built on a solderless breadboard



Book Source:

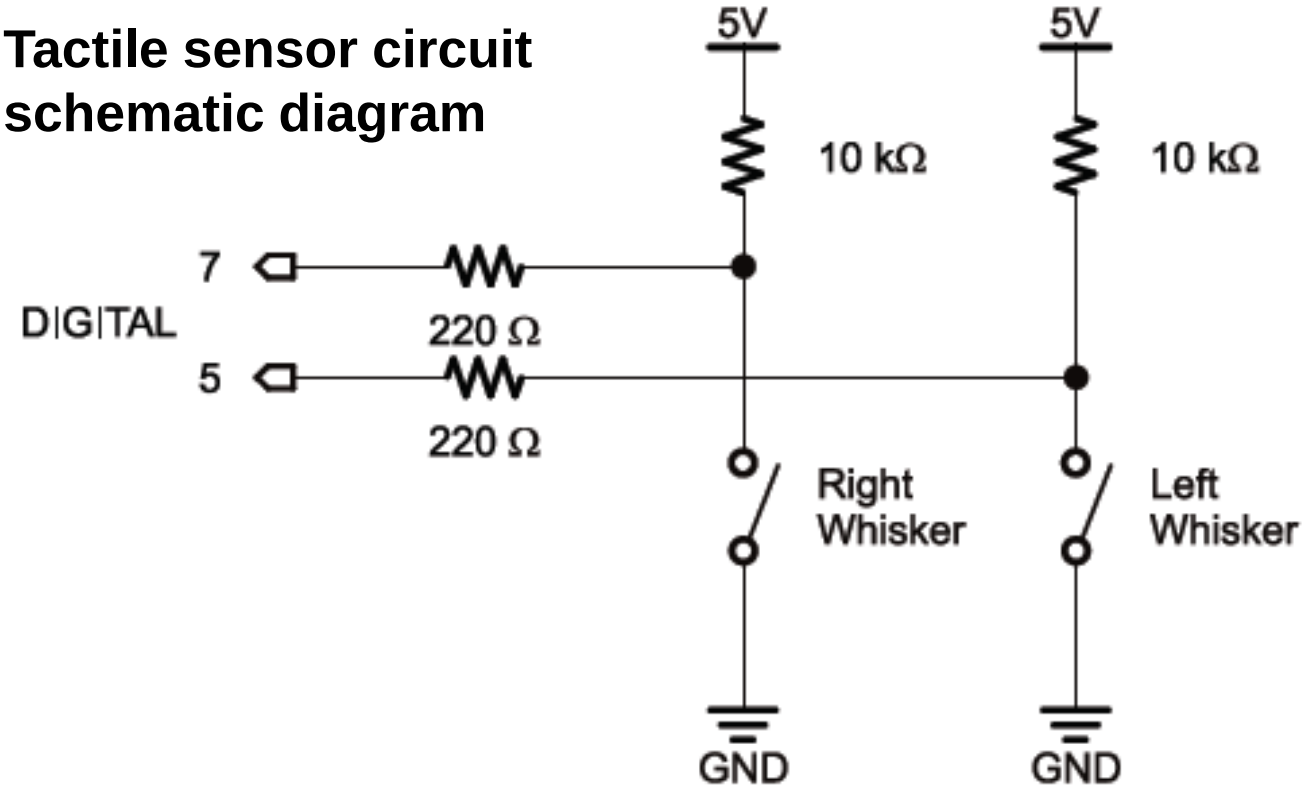
https://www.parallax.com/sites/default/files/downloads/122-32335-Robotics-BOE-Shield-Bot-Arduino-v1.0.pdf

Arduino BOE Shield bot Sensors…

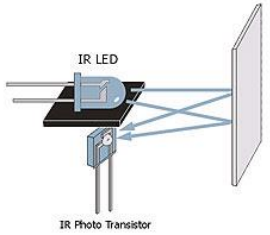


Tactile or Whiskers based Sensor (Assembly)…

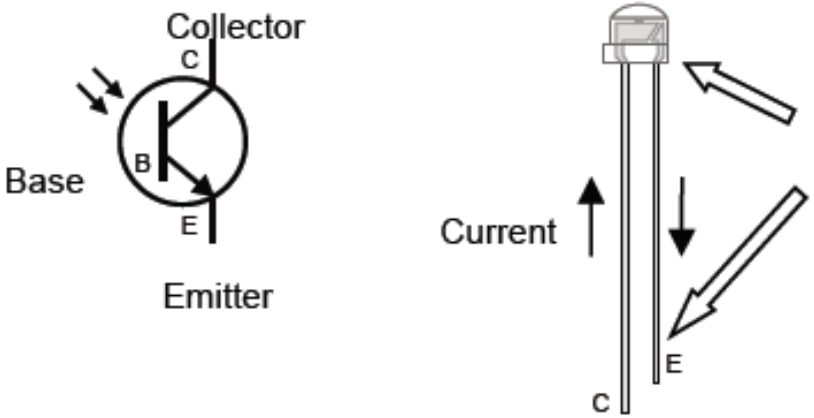
Tactile sensor circuit schematic diagram

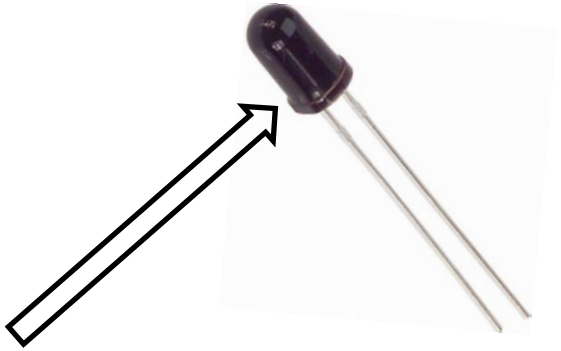


Arduino BOE Shield bot Sensors…



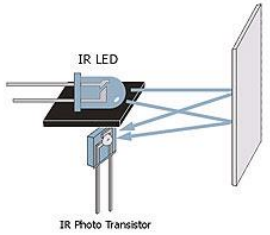
An IR (Infrared) Photo-transistor: Electronic Symbol (L) and Pictorial (R)



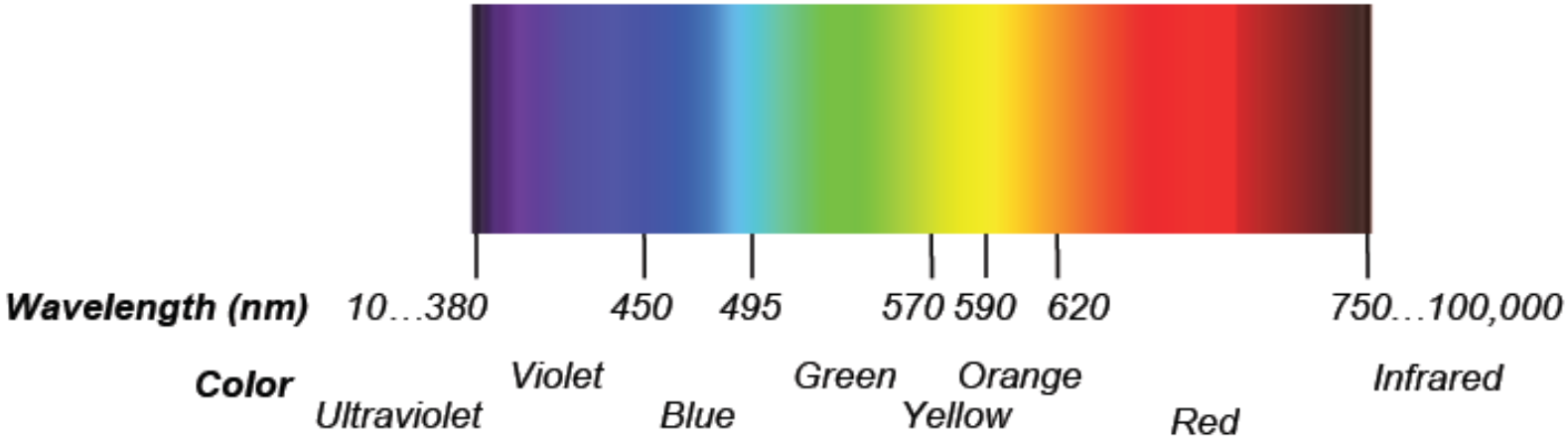


Flat spot and shorter pin indicate the emitter (E) terminal

Arduino BOE Shield bot Sensors…

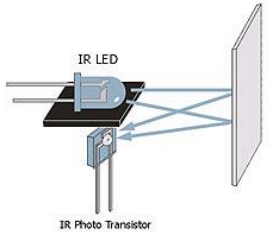


IR Phototransistor:



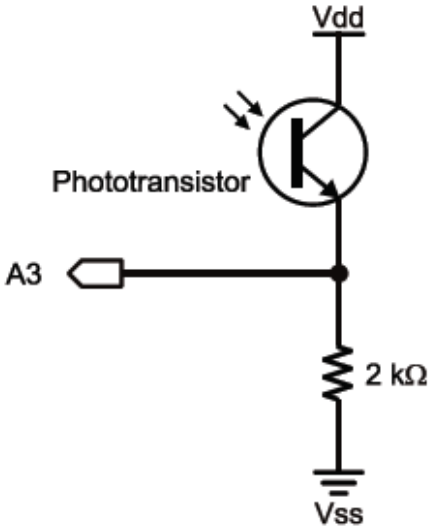
Distance between adjacent peaks is measured in nanometers (nm).

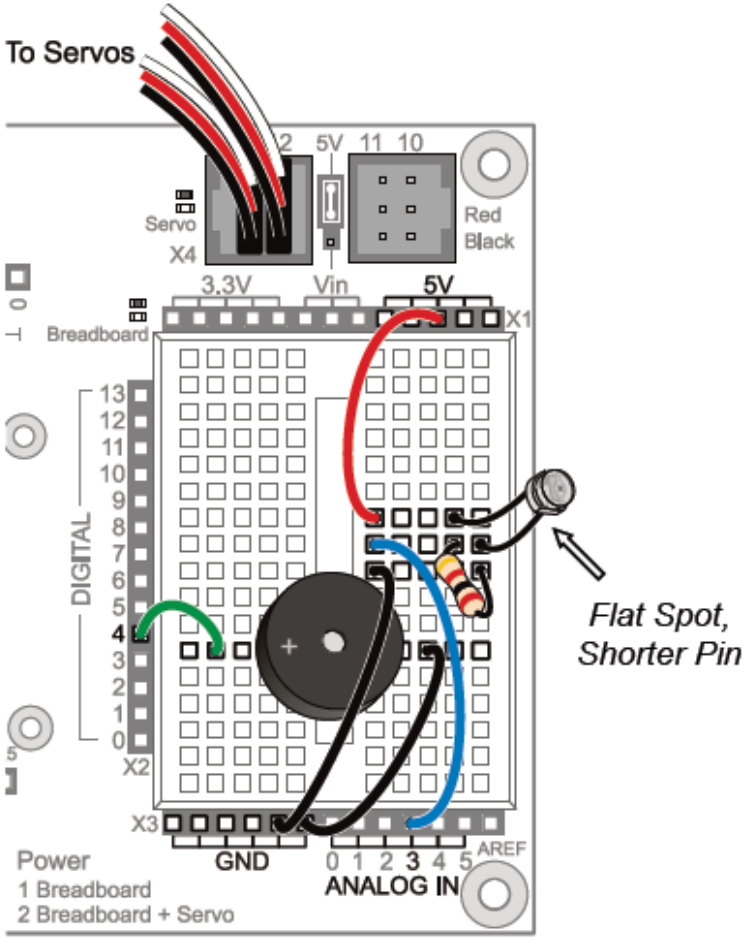
Arduino BOE Shield bot Sensors…



IR Phototransistor Sensor Circuit Assembly on a solderless breadboard:

IR Phototransistor sensor circuit schematic diagram





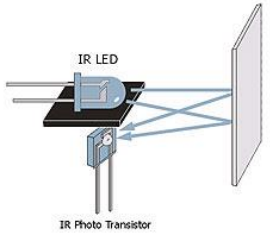
Presented by:

Question 4

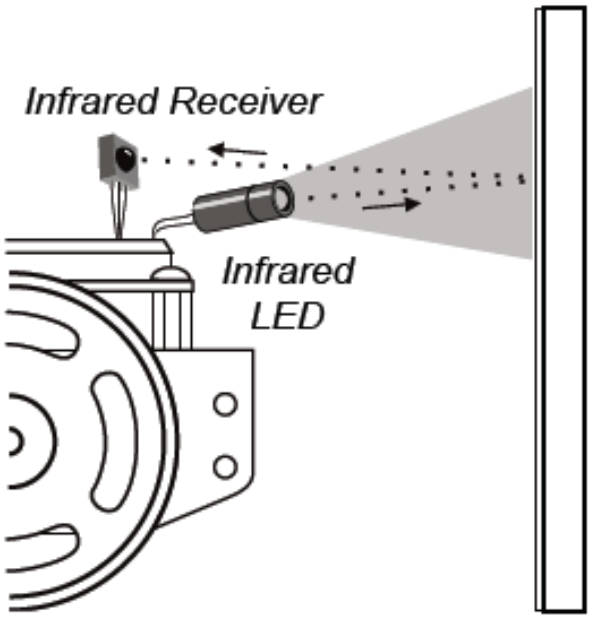
Distance between parallel peaks is measured in nanometers (nm).

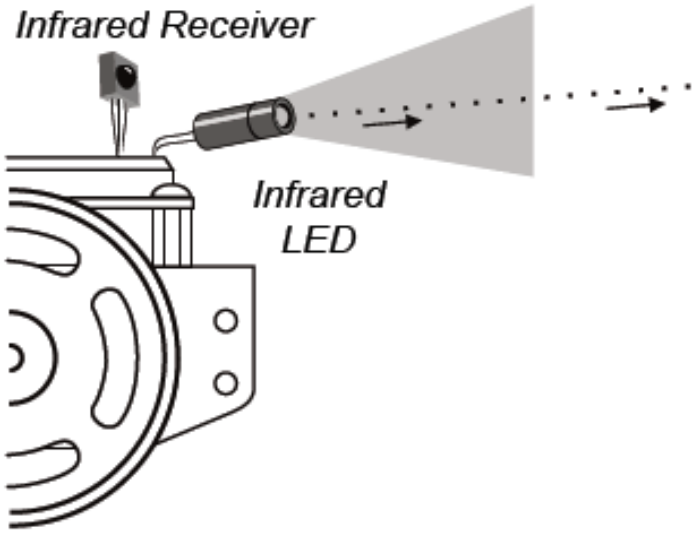
a) True
b) False

Arduino BOE Shield bot Sensors…

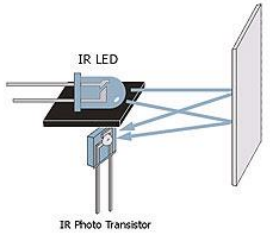


IR Receiver and IR LED: Object Detection Concept…

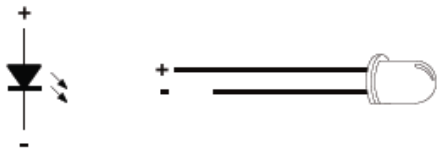




Arduino BOE Shield bot Sensors…



IR Receiver and IR LED: Object Detection Concept…

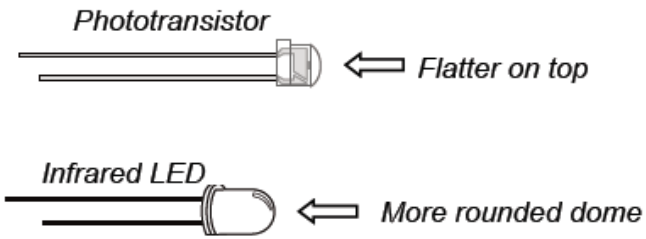


Infrared LED

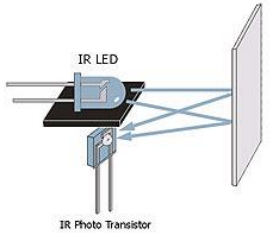
Infrared Receiver

Infrared LED Shield Assembly

✓ 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.



Arduino BOE Shield bot Sensors…

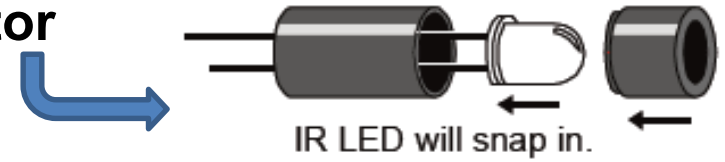


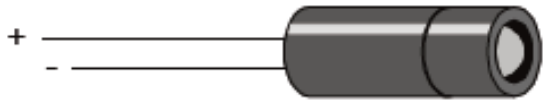
IR Receiver and IR LED: Object Detection Concept…

Assemble the IR Headlights

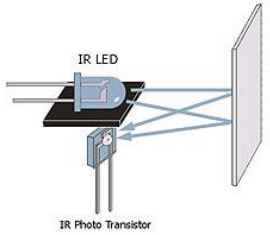
- ✓ Insert the infrared LED into the LED standoff base (the larger of the two pieces) as shown below. The standoff is shaped to fit the flat side of the LED.
- ✓ Make sure the IR LED snaps into the LED standoff.
- ✓ Slip the short tube over the IR LED's clear plastic case. The ring on one end of the tube should fit right into the LED standoff with a little twist.

Constructing a "mini" collimator



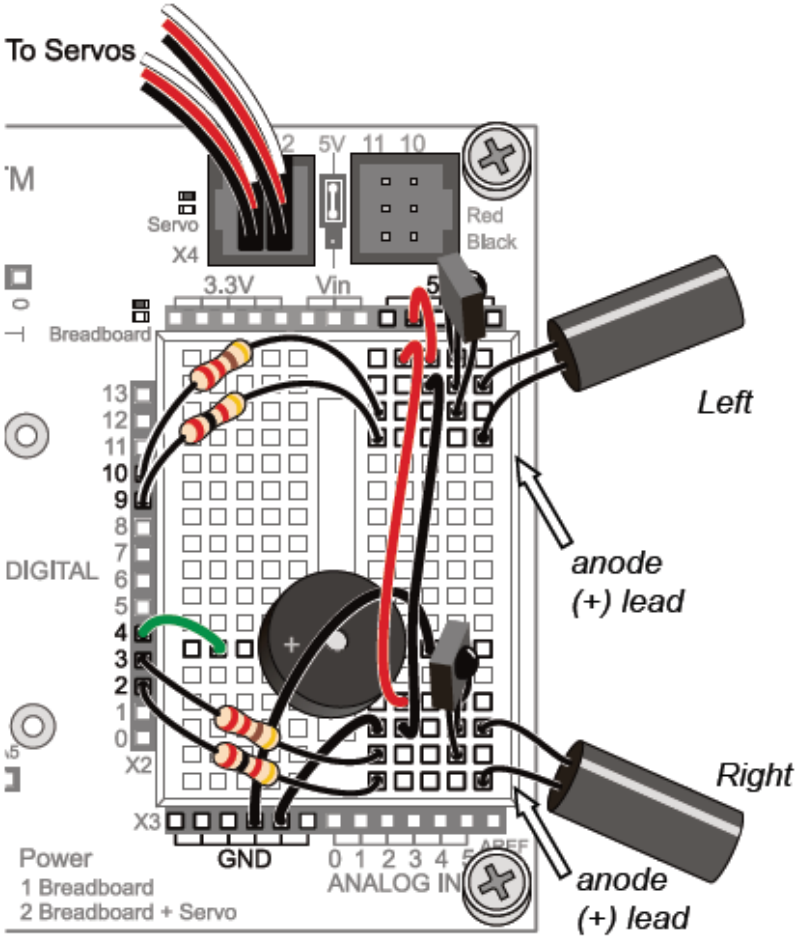


Arduino BOE Shield bot Sensors…

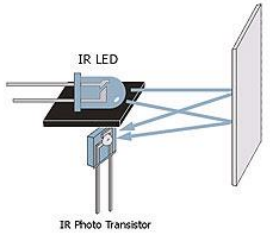


IR Receiver and LED: Remote Control Concept

IR Receiver and IR LED Sensor Detection Circuit Assembled on a solderless breadboard.

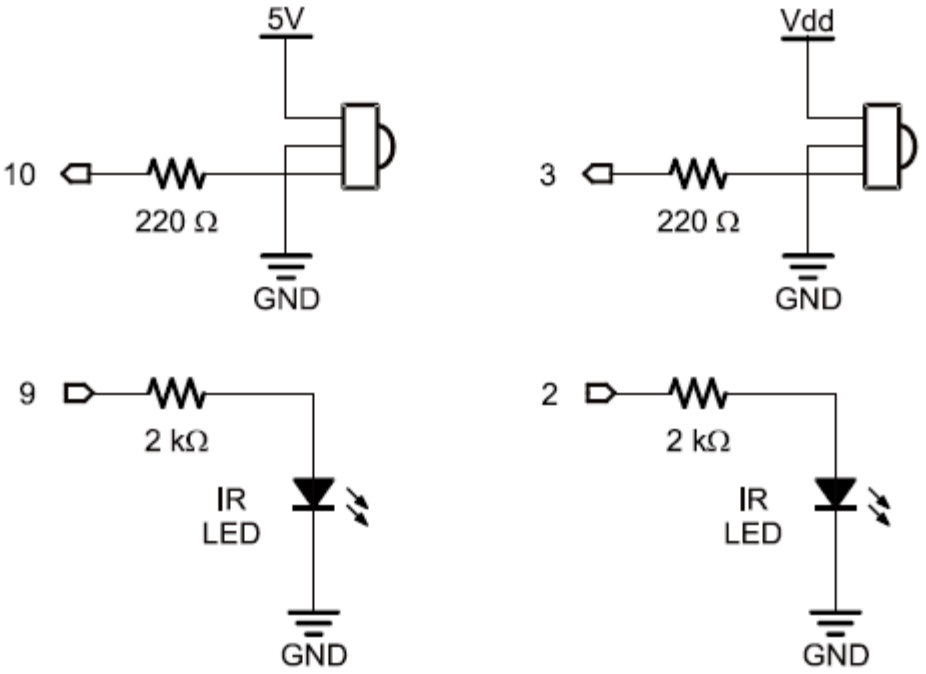


Arduino BOE Shield bot Sensors…



IR Receiver and IR LED: Object Detection Concept…

IR Receiver and IR LED sensor circuit schematic diagram

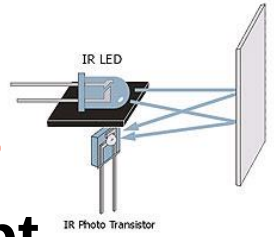


Note: Vdd = 5V

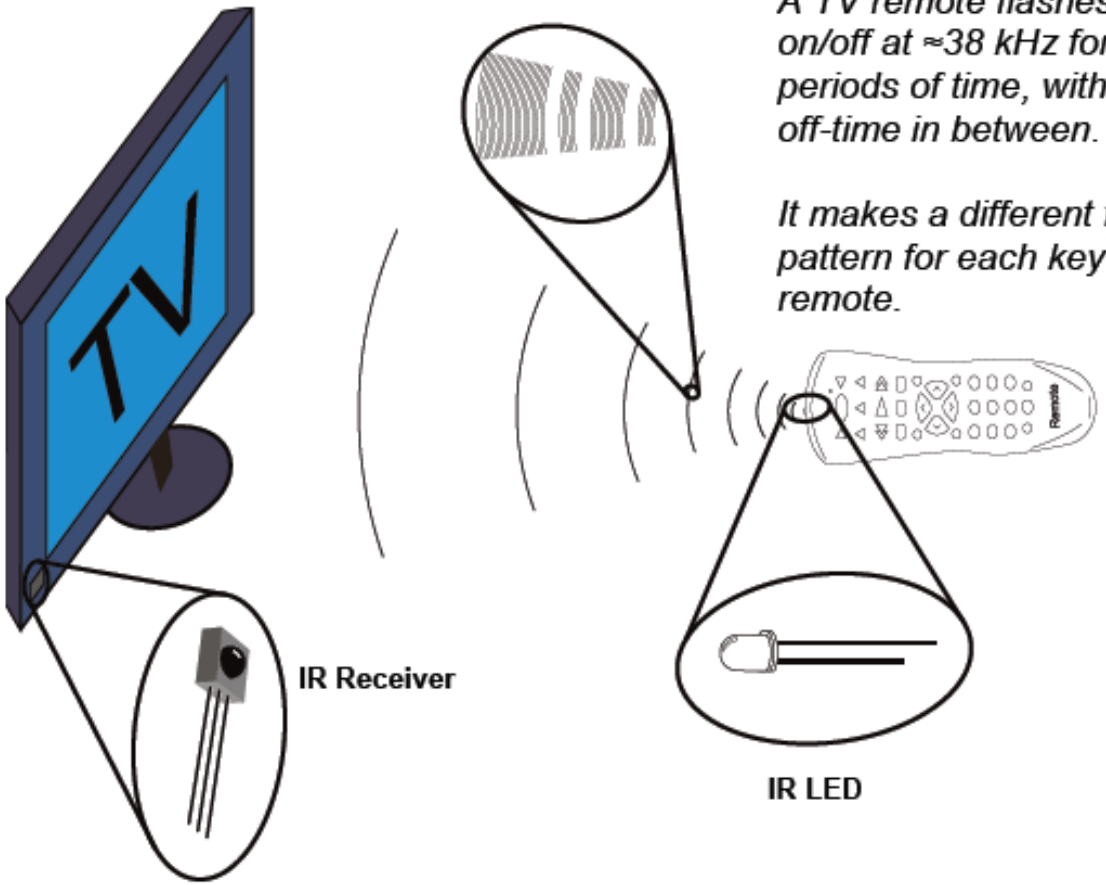
Watch your IR LED anodes and cathodes!

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.

Arduino BOE Shield bot Sensors…



IR Receiver and IR LED: Remote Control Concept



A TV remote flashes the IR LED on/off at ≈38 kHz for certain periods of time, with periods of off-time in between.

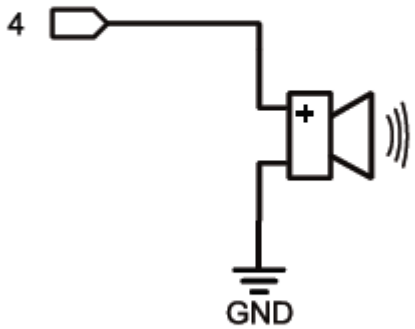
It makes a different flashing/off pattern for each key on the remote.

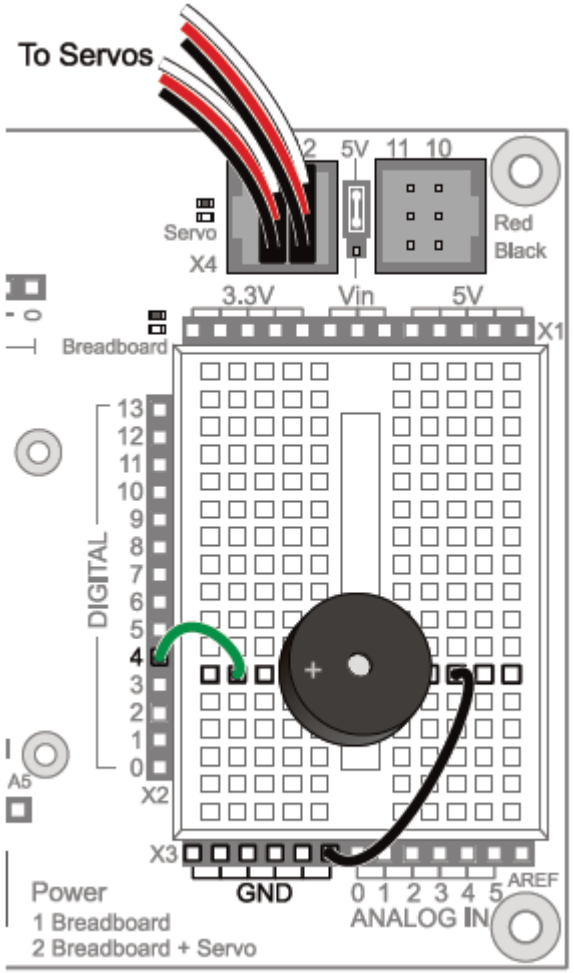
Question 5

In testing IR remotes, what smartphone may not allow you to see the IR LED signal?

Start/Reset Indicator Circuit

A tone is heard through the Piezo Buzzer upon starting the code or resetting the Arduino BOE shield board.



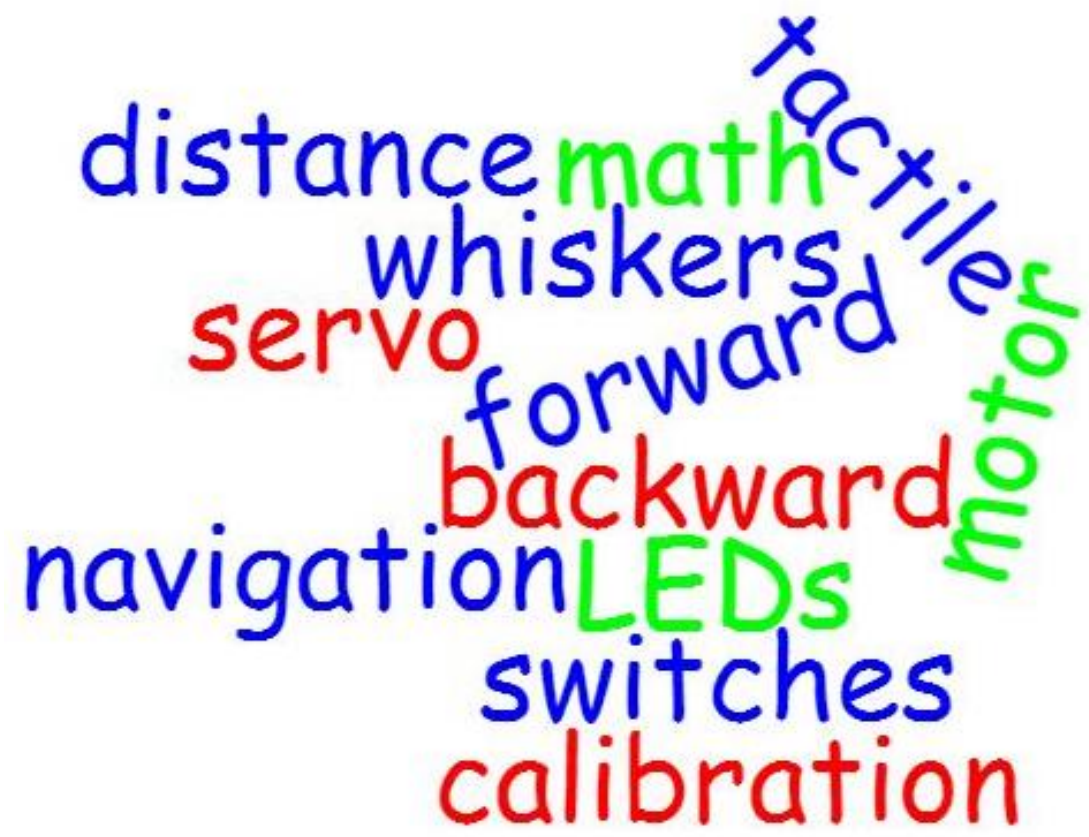


Start/Reset Indicator Circuit…

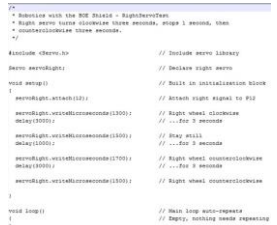
Start/Reset Indicator Code

```
/*
 * Robotics with the BOE Shield - StartResetIndicator
 * Test the piezospeaker circuit.
 */

void setup()                                    // Built in initialization block
{
  Serial.begin(9600);
  Serial.println("Beep!");

  tone(4, 3000, 1000);                          // Play tone for 1 second
  delay(1000);                                  // Delay to finish tone
}

void loop()                                     // Main loop auto-repeats
{
  Serial.println("Waiting for reset...");
  delay(1000);
}
```

Hands-On Labs: Navigation Code Examples



Hands-On Labs: Navigation Code Examples…



Objectives of Coding Labs

- To insure the Arduino IDE is installed correctly.
- To explore the Arduino IDE's programming environment.
- To explore navigation techniques with the Arduino BOE Shield bot.

Note: We'll be using examples from Robotics with Board of Education Shield for Arduino Book by Andy Lindsay, version 1.0

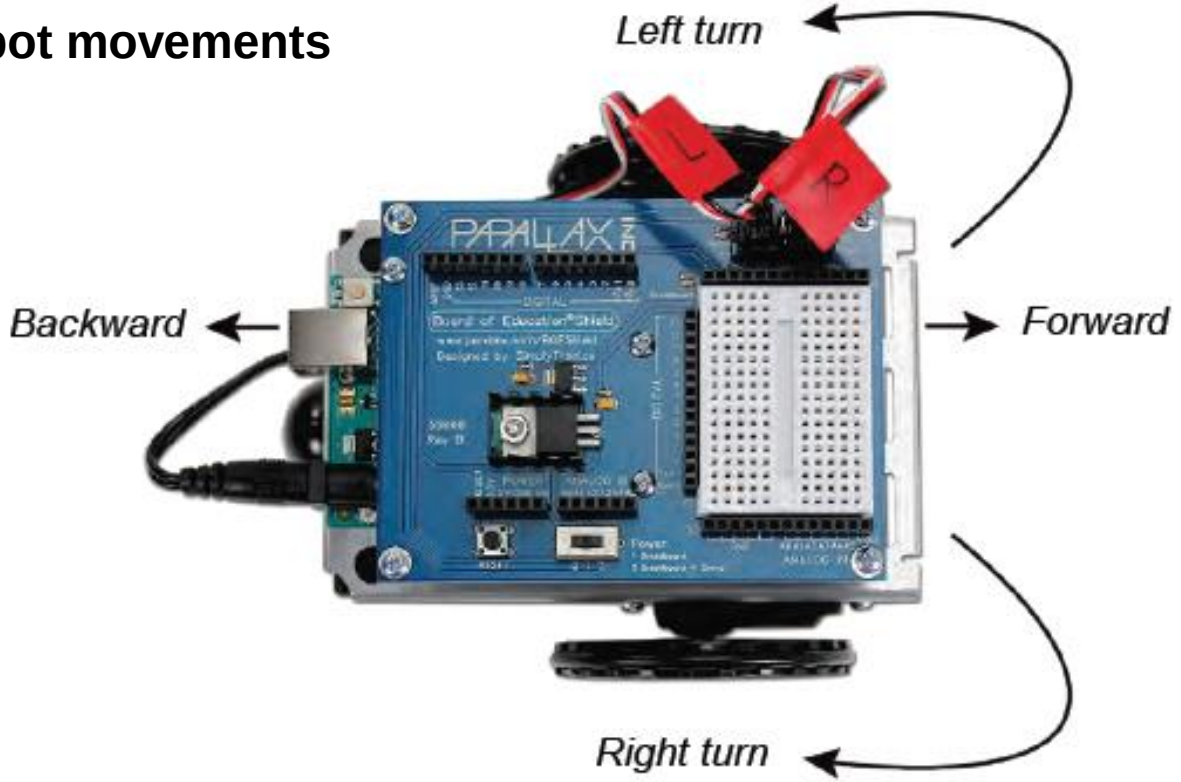
Book Source:

https://www.parallax.com/sites/default/files/downloads/122-32335-Robotics-BOE-Shield-Bot-Arduino-v1.0.pdf

Hands-On Labs: Navigation Code Examples

Arduino BOE Shield bot Navigation

Arduino BOE Shield bot movements



Hands-On Labs: Navigation Code Examples…

Arduino BOE Shield bot Navigation

Forward Three Seconds…

```
// Robotics with the BOE Shield - ForwardThreeSeconds
// Make the BOE Shield-Bot roll forward for three seconds, then stop.

#include <Servo.h>                          // Include servo library

Servo servoLeft;                            // Declare left and right servos
Servo servoRight;


void setup()                                // Built-in initialization block
{
  tone(4, 3000, 1000);                      // Play tone for 1 second
  delay(1000);                              // Delay to finish tone

  servoLeft.attach(13);                     // Attach left signal to pin 13
  servoRight.attach(12);                    // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700);        // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300);       // Right wheel clockwise
  delay(3000);                              // ...for 3 seconds

  servoLeft.detach();                       // Stop sending servo signals
  servoRight.detach();
}

void loop()                                 // Main loop auto-repeats
{                                           // Empty, nothing needs repeating
}
```

Hands-On Labs: Navigation Code Examples…

Arduino BOE Shield bot Navigation

Forward-Left Right-Backward…

```
// Robotics with the BOE Shield - ForwardLeftRightBackward
// Move forward, left, right, then backward for testing and tuning.

#include <Servo.h>                          // Include servo library

Servo servoLeft;                            // Declare left and right servos

Servo servoRight;

void setup()                                // Built-in initialization block
{
  tone(4, 3000, 1000);                      // Play tone for 1 second
  delay(1000);                              // Delay to finish tone

  servoLeft.attach(13);                     // Attach left signal to pin 13
  servoRight.attach(12);                    // Attach right signal to pin 12

  // Full speed forward
  servoLeft.writeMicroseconds(1700);        // Left wheel counterclockwise
  servoRight.writeMicroseconds(1300);       // Right wheel clockwise
  delay(2000);                              // ...for 2 seconds
```

Hands-On Labs: Navigation Code Examples…

Arduino BOE Shield bot Navigation

Forward-Left Right-Backward…

```
  // Turn left in place
  servoLeft.writeMicroseconds(1300);    // Left wheel clockwise
  servoRight.writeMicroseconds(1300);   // Right wheel clockwise
  delay(600);                           // ...for 0.6 seconds

  // Turn right in place
  servoLeft.writeMicroseconds(1700);    // Left wheel counterclockwise
  servoRight.writeMicroseconds(1700);   // Right wheel counterclockwise
  delay(600);                           // ...for 0.6 seconds

  // Full speed backward
  servoLeft.writeMicroseconds(1300);    // Left wheel clockwise
  servoRight.writeMicroseconds(1700);   // Right wheel counterclockwise
  delay(2000);                          // ...for 2 seconds

  servoLeft.detach();                   // Stop sending servo signals
  servoRight.detach();
}

void loop()                             // Main loop auto-repeats
{                                       // Empty, nothing needs repeating
}
```

Hands-On Labs: Navigation Code Examples…

Arduino BOE Shield bot Navigation

Your Turn – Pivoting

You can make the BOE Shield-Bot turn by pivoting around one wheel. The trick is to keep one wheel still while the other rotates. Here are the four routines for forward and backward pivot turns:

```
// Pivot forward-left
servoLeft.writeMicroseconds(1500);    // Left wheel stop
servoRight.writeMicroseconds(1300);   // Right wheel clockwise


// Pivot forward-right
servoLeft.writeMicroseconds(1700);    // Left wheel counterclockwise
servoRight.writeMicroseconds(1500);   // Right wheel stop

// Pivot backward-left
servoLeft.writeMicroseconds(1500);    // Left wheel stop
servoRight.writeMicroseconds(1700);   // Right wheel counterclockwise

// Pivot backward-right
servoLeft.writeMicroseconds(1300);    // Left wheel clockwise
servoRight.writeMicroseconds(1500);   // Right wheel stop
```