



Sensors and Auton

University Liggett School Middle School Robotics

Tony Pan

10/27/2017

Review on Servos

```
Servo testServo = hardwareMap.get(Servo.class,
    "TestServo");

    if (gamepad1.y) {
        testServo.setPosition(0);
    } else if (gamepad1.x || gamepad1.b) {
        testServo.setPosition(0.5);
    } else if (gamepad1.a) {
        testServo.setPosition(1);
    }
}
```

Color Sensor



- A sensor is a device that lets the Robot Controller get information about the robot's environment.
- The Modern Robotics Color Sensor detects the color of an object using reflected light or the color of light from an external light source. Great for detecting and tracking colored lines and determining the color of a lit object.

Passive/Active Modes

In *Active* mode, the internal LED provides the light source for detecting the target color. Best results in *Active* mode are obtained when the target is within about 7cm of the color sensor.

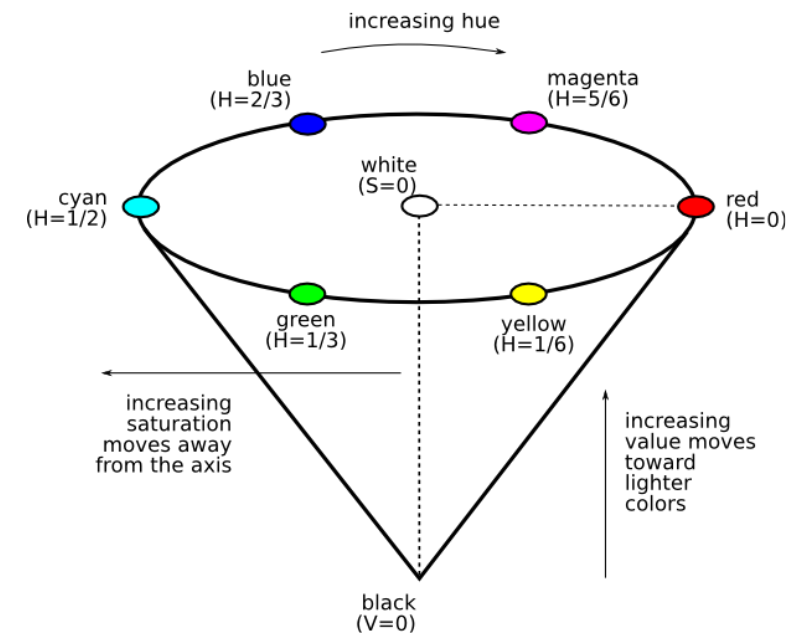
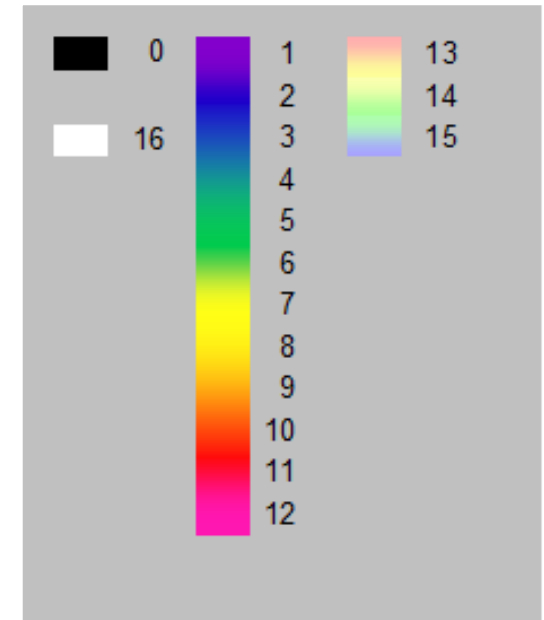
In *Passive* mode, the internal LED is switched off and the sensor detects the color of the light from an external light source such as a colored LED or lighted device.

Color

HSV: The hue (H) of a color are described by a number that specifies the position of the corresponding pure color on the color wheel, as a fraction between 0 and 1. The saturation (S) of a color describes how white the color is. The value (V) of a color, also called its lightness, describes how dark the color is.

RGB: Red, green, blue values from 0 (lightest) to 255 (darkest).

The Color Number returned is in the range 0 – 16 as shown this chart. The sensor also returns values for red, green, blue and white.



Arrays

- Think about matrix, vectors, grids etc.
- Maybe simply a box with grids inside it
- Ex. `int[] arrayName = new int[](size);`
- `float hsvValues[] = {0F,0F,0F};`

Device Interface

- `deviceInterfaceModule DIM = hardwareMap.
deviceInterfaceModule.get("DeviceInterfaceModule");`
- Controls the sensors and gives the output back to the robot controller
- How to check color?
- `if (colorSensor.blue() > colorSensor.red() &&
colorSensor.blue() > colorSensor.green()) // it's blue`
- `if (colorSensor.red() > colorSensor.blue() &&
colorSensor.red() > colorSensor.green()) // it's red`
- `DIM.setLED(1, true); // red on`
- `DIM.setLED(0, false); // blue off`

In Code

```
ColorSensor colorSensor = null;  
    colorSensor =  
hardwareMap.get(ColorSensor.class,  
    "sensor_color");  
colorSensor.enableLed(true);  
    colorSensor.red();  
    colorSensor.blue();
```


Try/Catch

- To execute code that might throw an exception, you must enclose it in a try/catch statement.
- `public void runOpMode() throws InterruptedException`

```
try {  
    statements; // code that might throw an exception  
} catch (ExceptionType name) {  
    statements; // code to handle the error  
}
```

Hardware Map

- This class can be used to define all the specific hardware for a single robot.
- Declare all the variables (motors, servos, sensors)
- `public class Hardware7140 //7140's robot (in the hardware map class)`
- `Hardware7140 robot = new HardwarePushbot(); //using 7140's robot's hardware (in the op mode class)`
- `robot.init(hardwareMap); //pass the hardware map in as a parameter to the hardware class from the op mode`
- `public void init(HardwareMap 7140) //the hardware map class initializes the robot and returns the initialized variables`

Auton

- `@Autonomous`
- `extends LinearOpMode`

```
public void runOpMode() {  
    robot.init(hardwareMap);  
    waitForStart();  
    // set motor power...  
    sleep(time in milliseconds);  
}
```