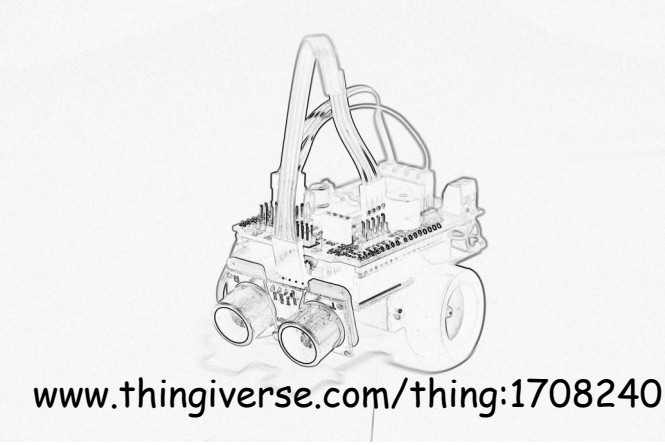

Sense, Think and Move!

Lesson 2: Using Sensors> rtheiss.com

Name:

Date:

School/Subject:

In this activity you will use your Ultra Sonic Sensor to add to your motor movement, ad allow your robot to sense, think and move based on how far an object is away from it.

1. Step One – Add "Head" Code: Open up your MotorCode from lesson 1 and delete the loop portion, then save a file as "UltraSonicCode."

Code that is located before the actual program is often called the head code. This will tell the program which pins you are using for the sensors and motors and what there nicknames will be if you want to assign any. The Ultra Sonic Sensor you are using relies on Echo-Location just like a bat to "see" things. It will send out a signal from one "eye" and then sense when the signal comes back from the other. If you are able to time how long it takes for the signal to bounce back, you can judge the distance.

To start you must add these lines before all other codes:

```
#define trigPin 11
#define echoPin 12
```

.

2. Step Two – Setup Code: At the start of your setup code you will have to tell the program if signals are going out or coming in. You will be sending an "Trigger" sound out and looking for the "Echo" signal coming in.

After the starting bracket in your setup code, add these lines:

```
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

3. Step Three – Time the Echo and Set Your Distance: This code will look for the distance so you can compare it in the loop to allow your bot to make a decision of whether or not to do something. Add this after the setup code for the beginning of your loop code:

```
void loop ()
{
// Ping Utrasonic Sensor to find obstacles within distance in cm
        int duration, distance;
        digitalWrite(trigPin, HIGH);
        delay(1);
        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
        distance = (duration/2) / 29.1;

                if (distance < 15)

                        {

                                backward ();
                                delay(100);
                                brake ();

                        }

}
```

4. Step Four– Make a Shortcut Program for Moving the Motors Backward and One to Brake: All the way at the end of the code (below everything else, add two of your motor movement programs so we can create a shortcut for what the programs in the loop code will do. Its easier to have these as separate programs in case we need to use them multiple times, or make changes in the future.

```
void backward ()
{
  Lservo.write(0);
  Rservo.write(180);
}

void brake ()
{
  Lservo.write(90);
  Rservo.write(90);
}
```

5. Step Five–Upload: Before you upload, guess what the code will do based on your prior knowledge. After the upload does it do what you expected? If you have any trouble with the syntax check the full code in the thingiverse thing files.