```
/*
StalkerBot Code

Purpose:
The code below is written to make a robot to follow a moving object.
It uses two ultrasonic sensors to detect where an object is and
reacts to the position of the object in order to continue following
it. First, an ultrasonic signal is emitted from the sensors, then
when the signal is again recieved, the time it took to come back is
measured and converted to inches. Using the inch value, several if
loops control the position of the robot based on the instantenous
inch value.

Robot components:
 - 2 3 wire VEX Motor controllers (servos)
 - 2 VEX Ultrasonic sensors
 - 1 Arduino UNO
 - 1 6 AA battery holder
 - 1 9 volt battery

The Circuit:
 - The ultrasonic sensors each have two three wire pins coming off (one for
 input and one for output), each 3 wire pin has a pin for Power, Ground, and Digital
 - The 3-wire VEX motor controllers each have three pins, one for ground, power,
 and digital PWM. Instead of recieving power from the Arduino, the servos recieve
 power from the 6 AA batteries

Pin Connections:
Ultrasonic Sensor 1:
          Input - Ground, 5V, Digital Pin 7
         Output - Ground, 5V, Digital Pin 8
Ultrasonic Sensor 2:
          Input - Ground, 5V, Digital Pin 12
         Output - Ground, 5V, Digital Pin 13
Motor controller 1 - Ground, Power supply, Digital PWM Pin 9
Motor controller 2 - Ground, Power supply, Digital PWM Pin 10

Code by Tzofi Klinghoffer
Originally Created November 25, 2012
Final Version Saved December 3, 2012
Credits to David A. Mellis for his Ping Code found in the Arduino Examples Library
and Dave Otten for feedback and advice
*/

#include <Servo.h>  //communicates to program to refer to the Servo library in the Arduino files

//These servos control the movement of the robot
Servo myservo; // create servo object named myservo to control a servo
Servo myservo1; //create servo object named myservo1 to control a servo

int pos = 0;    // variable to store the servo position

//Sets the Arduino Digital Ports for the Ultrasonic Sensor
const int pingPin = 7; //sets output pin for ultrasonic sensor 1 to digital port 7
const int receiverPin = 8; //sets input pin for ultrasonic sensor 1 to digital port 7
const int pingPin1 = 12; //sets output pin for ultrasonic sensor 2 to digital port 7
const int receiverPin1=13; //sets input pin for ultrasonic sensor 2 to digital port 7
```

```
void setup() {
  Serial.begin(9600); //establishes a communication rate and BAUD number
  pinMode(pingPin, OUTPUT); //initializes the digital port 7 pingPin as an output port
  pinMode(pingPin1, OUTPUT); //initializes the digital port 12 pingPin1 as an output port
  pinMode(receiverPin, INPUT); //initializes the digital port 8 recieverPin as an input port
  pinMode(receiverPin1, INPUT); //initializes the digital port 13 recieverPin1 as an input port
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  myservo1.attach(10); // attaches the servo on pin 10 to the servo object
}

void loop()
{
  long duration, inches, cm; //establishes three objects that can hold large numbers called duration, inches, and cm
  long duration1, inches1, cm1; //establishes three objects that can hold large numbers called duration1, inches1, and cm1

//Triggers the first ultrasonic sensor to gather distances
  digitalWrite(pingPin, LOW); //turns on the output to a low power (not as many ultrasonic waves)
  delayMicroseconds(2); //waits for two milliseconds
  digitalWrite(pingPin, HIGH); //moves the output ultrasonic to high power
  delayMicroseconds(5); //waits five milliseconds
  digitalWrite(pingPin, LOW); //drops ultrasonic output back to low
  duration = pulseIn(receiverPin, HIGH); //turns the input on ultrasonic sensor 1 to high so it can recieve the ultrasonic
waves that were emitted and sets duration to this time value

  // convert the time from the first ultrasonic sensor into a distance
  inches = microsecondsToInches(duration); //sets the inches object to the value of the duration converted to inches
  cm = microsecondsToCentimeters(duration); //sets the cm object to the value of the duration converted to centimeters

  //Triggers the second ultrasonic sensor to gather distances
  digitalWrite(pingPin1, LOW); //turns on the output to a low power (not as many ultrasonic waves)
  delayMicroseconds(2); //waits for two milliseconds
  digitalWrite(pingPin1, HIGH); //moves the output ultrasonic to high power
  delayMicroseconds(5); //waits five milliseconds
  digitalWrite(pingPin1, LOW); //drops ultrasonic output back to low
  duration1 = pulseIn(receiverPin1, HIGH); //turns the input on ultrasonic sensor 2 on to high so it can recieve the ultrasonic
waves that were emitted and sets duration1 to this time value

   // convert the time from the second ultrasonic sensor into a distance
  inches1 = microsecondsToInches(duration1); //sets the inches1 object to the value of the duration converted to inches
  cm1 = microsecondsToCentimeters(duration1); //sets the cm1 object to the value of the duration converted to centimeters

 //Serial Moniter Control
  Serial.print(inches); //prints the number of inches read by the first ultrasonic sensor
  Serial.print("in, "); //prints "in" after the number of inches to give a measurement to the value
  Serial.print(cm); //prints the number of centimeters read by the first ultrasonic sensor
  Serial.print("cm"); //prints "cm" after the number of centimeters to give a measurement to the value
  Serial.println(); //goes to next line in serial moniter

  Serial.print(inches1); //prints the number of inches read by the second ultrasonic sensor
  Serial.print("in1, "); //prints "in" after the number of inches to give a measurement to the value
  Serial.print(cm1); //prints the number of centimeters read by the second ultrasonic sensor
  Serial.print("cm1"); //prints "cm" after the number of centimeters to give a measurement to the value
  Serial.println(); //goes to next line in serial moniter

  delay(100); //waits 100 milliseconds before continuing

//Robot Control Code based on distance read from ultrasonic sensors
//Motors operate on a range of 0 to 180 where 90 is braking
```

```
if(inches1>18 && inches>18) //if both sensors read more than 18 inches, brake both motors
  {
   myservo.write(90); //brake the left motor
   myservo1.write(90); //brake the right motor
  }
if(inches1<18 && inches<18) //if both sensors read more than 18 inches, go forward
  {
   myservo.write(110); //set the left motor forward
   myservo1.write(70); //set the right motor forward
  }
if(inches<18 & inches1>18) //if only something is in front of the left ultrasonic sensor, turn left
  {
   myservo1.write(50); //set right motor forward
  }
if(inches>18 & inches1<18) //if only something is in front of the right ultrasonic sensor, turn right
  {
   myservo.write(140); //set left motor forward
  }
}
long microsecondsToInches(long microseconds) //store value for microseconds converted to inches
{
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2; //divide microseconds to get inches
}

long microsecondsToCentimeters(long microseconds) //store value for microseconds converted to centimeters
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2; //divide microseconds to get centimeters
}
```