```
/*
 * Code to control two stepper motors simultaneously using the stepper library
function calls
 * This code has functions which make calls to initiate one step from each of the
two motors at a time
 * therefore each motor takes it in turn to step with the end result of them both
running together
 * the variable 'lastCall' allows for a continuous input to keep the steppers
running if there is no serial input
 * Written by Seven Vinton - March 2016 - available and free to use and
distribute in the public domain.
 */
#include <Stepper.h>

//definitions for each command to be recieved via serial port
#define COMMAND_LEFT 'a'
#define COMMAND_RIGHT 'd'
#define COMMAND_FORWARD 'w'
#define COMMAND_BACK 's'
#define COMMAND_STOP ' '

//enter the steps per rev for your motors here
int stepsInRev = 200;

//this sets the value for the for loops and therefore sets the amount of steps in
each call
int num_of_steps = 1;
// setup pins for each driver motor1 ~ IN1, IN2, IN3, IN4; motor2 ~ IN1, IN2,
IN3, IN4
Stepper myStepper1(stepsInRev, 10, 11, 12, 13);
Stepper myStepper2(stepsInRev, 9, 8, 7, 6);
// variable to store the last call to the serial port
char lastCall = ' ';

//I had prototyped all of the functions here, but for some reason the Arduino IDE
didn't like it
//so they are coded out until I an figure out why
//void forwardStep();
//void backwardStep();
//void leftStep();
//void rightStep();
//void allStop();

//to move the motors forward
void forwardStep(int steps){
  Serial.println("forward");
  // step one step each forward
  myStepper1.step(1);
  myStepper2.step(1);
  delay(10);
}

// to move the motors back
void backwardStep(int steps){
  Serial.println("backward");
  // step one step each backward
  myStepper1.step(-1);
  myStepper2.step(-1);
  delay(10);
}


// to move the motors in opposite directions (left)
void leftStep(int steps){
  Serial.println("left");
 // step one step each left
  myStepper1.step(1);
  myStepper2.step(-1);
  delay(10);
}


// to move the motors in opposite directions (right)
void rightStep(int steps){
  Serial.println("right");
  // step one step each right
  myStepper1.step(-1);
  myStepper2.step(1);
  delay(10);
}

// to power down the motor drivers and stop the motors
void allStop(){
  Serial.println("stop");
  // steppers stop
  PORTD = B00000000; //sets all of the pins 0 to 7 as LOW to power off stepper1
  PORTB = B00000000; //sets all of the pins 8 to 13 as LOW to power off stepper2
}
void setup() {
  Serial.begin(9600);//start the bluetooth serial port - send and recieve at 9600
```

```
baud
  // set the speed at 60 rpm:
  myStepper1.setSpeed(60);
  myStepper2.setSpeed(60);
}

void loop() {
//check to see if there is serial communication and if so read the data
if(Serial.available()) {
char data = (char)Serial.read();
// switch to set the char via serial to a command
switch(data) {
  case COMMAND_FORWARD:
    forwardStep(num_of_steps);
    break;
  case COMMAND_BACK:
    backwardStep(num_of_steps);
    break;
  case COMMAND_LEFT:
    leftStep(num_of_steps);
    break;
  case COMMAND_RIGHT:
    rightStep(num_of_steps);
    break;
  case COMMAND_STOP:
    allStop();
    break;
}
// set the 'lastCall' variable to the last call from the serial
lastCall = data;
}
else{
char data = lastCall;
switch(data) {
  case COMMAND_FORWARD:
    forwardStep(num_of_steps);
    break;
  case COMMAND_BACK:
    backwardStep(num_of_steps);
    break;
  case COMMAND_LEFT:
    leftStep(num_of_steps);
    break;
  case COMMAND_RIGHT:
    rightStep(num_of_steps);
    break;
  case COMMAND_STOP:
    allStop();
    break;
}
lastCall = data;
}
}
```