```
// The code contained at gobotics.com does not hold gobotics.com liable for any damage or injury
// nor does gobotics.com provide any warranty for damage caused by code. It is at your own risk to use
// any and all examples provided.
// Copyright (c) 2012 Gobotics Inc. All Right Reserved, gobotics.com
// Example Line Follower Program for the Rover5 Line Follower Kit
// To Be Used With Arduino Duemilanove Microcontrollers
//
// By; Michael Wolfe



// Assignement of sensors to port pins
const int linesenseleft = 1;
const int linesensemiddle = 2;
const int linesenseright = 3;

// Assignment of H-Bridge inputs to port pins
const int moten = 4;
const int motc = 5;
const int motd = 6;

const int moten2 = 8;
const int motc2 = 9;
const int motd2 = 10;

// Variable to store the number of the last sensor to be active
int lastState = 0;  // 0 = none, 1 = middle, 2 = left, 3 = right
```

```
//int buttonState2 = 0;


void setup()
{
// Defines inputs and outputs
  pinMode(linesensemiddle, INPUT);
  pinMode(linesenseleft, INPUT);
  pinMode(linesenseright, INPUT);
  pinMode(revbutton, INPUT);
  pinMode(collisionswitch, INPUT);

  pinMode(moten, OUTPUT);
  pinMode(motc, OUTPUT);
  pinMode(motd, OUTPUT);
  //pinMode(buttonPin2, INPUT);
  pinMode(moten2, OUTPUT);
  pinMode(motc2, OUTPUT);
  pinMode(motd2, OUTPUT);

}



// THIS IS EFFECTIVELY MAIN, BUT IT LOOPS
void loop()
{
// Reads the states of linessensemiddle, linesenseleft, linesenseright and inverts their states
    boolean lineSenseMiddleState = !digitalRead(linesensemiddle);    // read the button, and store in variable
```

```
boolean lineSenseLeftState = !digitalRead(linesenseleft);
boolean lineSenseRightState = !digitalRead(linesenseright);

delay(10);                    // debounce


if (lineSenseRightState == 1 || (lineSenseLeftState == 0 && lineSenseMiddleState == 0 && lastState == 3)){
   digitalWrite(motc, LOW);
   digitalWrite(motd, HIGH);

   digitalWrite(motc2, LOW);
   digitalWrite(motd2, HIGH);

   digitalWrite(moten, HIGH);
   digitalWrite(moten2, HIGH);

   lastState = 3;
}
else
{
  if (lineSenseLeftState == 1 || (lineSenseRightState == 0 && lineSenseMiddleState == 0 && lastState == 2)){
   digitalWrite(motc, HIGH);
   digitalWrite(motd, LOW);

   digitalWrite(motc2, HIGH);
   digitalWrite(motd2, LOW);

   digitalWrite(moten, HIGH);
```

```
      digitalWrite(moten2, HIGH);


      lastState = 2;
    }
    else
    {
     if (lineSenseMiddleState == 1 || (lineSenseLeftState == 0 && lineSenseRightState == 0 && lastState == 1)){
       digitalWrite(moten, HIGH);
       digitalWrite(moten2, HIGH);


       digitalWrite(motc, HIGH);
       digitalWrite(motd, LOW);


       digitalWrite(motc2, LOW);
       digitalWrite(motd2, HIGH);


       lastState = 1;
      }
      else{
       digitalWrite(moten, LOW);
       digitalWrite(moten2, LOW);
      }
    }
  }


}
```

```
// NOTE:LINE SENSOR GOES HIGH ON WHITE, LOW ON BLACK
```