Line Detection

/*
 * This is a test sketch to see if your IR sensors work.
 */
int pin = 3;

void setup()
{
  // Start the serial monitor at a baud rate of 9600
  Serial.begin(9600);
}

void loop()
{
  /*
   * read the value of the timer in readValue()
   * function. Print that value to the terminal.
   * delay 200 milliseconds between reads.
   */
  int val = readValue();
  Serial.println(val);
  delay(200);
}

int readValue()
{
  /*
   * For these sensors to work you need
   * to set the output pin to high to charge
   * the onboard capacitor. This process
   * takes about 20us. The next step is
   * to change the pin to an input and read
   * from the capacitor. The while loop increments
   * as long as the value of the capacitor is > 0.
   * The longer it takes to discharge the capacitor,
   * the less light reflected. So, when on a black line
   * you should have a larger value than when on a
   * white line.
   */
  pinMode(pin, OUTPUT);
  digitalWrite(pin, HIGH);
  delayMicroseconds(20);
  pinMode(pin, INPUT);
  long timer = 0;
  while(digitalRead(pin) > 0)
  {
    timer++;
  }
  return timer;
}
**Motor Driver**

// Adafruit 2448 Motor Driver Carrier:
int d4 = 4;
int d5 = 5; // Digital Write
int d3 = 3; // Analog Write

int i = 0;

// the setup routine runs once when you press reset:
void setup() {
  Serial.begin(9600); // Used for debugging
  pinMode(d3, OUTPUT);
  pinMode(d4, OUTPUT);
  pinMode(d5, OUTPUT);
}

// the loop routine runs over and over again forever
// Adjust speed of motors
void loop() {
  digitalWrite(d4, HIGH);
  digitalWrite(d5, LOW);
  for (i = 0; i < 255; i++) {
    analogWrite(d3, i);
    Serial.println(i);
    delay(500);
  }
}

**Object Detection**

/*
This snippet explains the basic functionality of a photoresistor and how
the arduino can be configured to work with it.

One of the photoresistor pins can be connected to PIN_0 and the other pin can be
connected to +5[V].
Connect a one leg/pin of the resistor to PIN_0. Resistor should be around 10 KOhm.
A higher resistor value gives higher readings.
Connect the other leg of the resistor to ground.
*/

/*
The setup() function is when the sketch starts.
It is run one time
Variables used:
lightPin: Defines the pin for photoresistor
ledPin: Defines the pin for LED

Serial communication is set to 9600 bits per second
ledPin set as an output pin (Default: All pins set to input)

```cpp
void setup()
{
    int lightPin = 3;
    int ledPin = 11;
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}
```

This value is also sent to the ledPin.
Depending on value of resistor you have to divide the value.
For example, with a 10k resistor divide the value by 2, for 100k resistor divide by 4.

Add a delay to observe changes
You can add a shorter delay for faster response to light

```cpp
void loop()
{
    Serial.println(analogRead(lightPin) * 10);
    analogWrite(ledPin, analogRead(lightPin) * 10);
    delay(100);
}
```

**Servo Control**

```cpp
#include <Servo.h>
// in this example there are 2 servos in use.
Servo servo1; // create servo object to control a servo
// twelve servo objects can be created on most boards
Servo servo2;

int pos = 0; // variable to store the servo position
int pos2;

void setup()
{
    Serial.begin(9600); //Begin serial communication
    servo1.attach(9); // attaches the servo on pin 9 to the servo object
    servo2.attach(10);
}
```
void loop() {
    pos2 = 80;
    for (pos = 70; pos <= 100; pos += 1) { // servos can go from 180 degrees to 0 degrees
        // in steps of 1 degree
        pos2--;
        servo1.write(pos); // tell servo to go to position in variable 'pos'
        servo2.write(pos2);
        Serial.println(pos);
        Serial.println(pos2);
        delay(50); // waits 50ms for the servo to reach the position
    }

    pos2 = 50;
    for (pos = 100; pos >= 70; pos -= 1) { // servos can go from 180 degrees to 0 degrees
        pos2++;
        servo1.write(pos); // tell servo to go to position in variable 'pos'
        servo2.write(pos2);
        Serial.println(pos);
        Serial.println(pos2);
        delay(50); // waits 50ms for the servo to reach the position
    }
}