**Light Seeking Robot**

Olaudo Victor & Anthony Hancock

Department of Electrical and Computer Engineering

Georgia Southern University

Georgia

**ABSTRACT**

An light following robot can be used for Industrial purpose & Military operation. The major components of our project include an ArduinoSparkfun Inventor’s Kit, Servo motor, BO motor(i.e gearmotors), motor driver, photo sensor, Chassis body, Battery & battery holder. The system is controlled by the Arduino Sparkfun Inventor’s Kit which is an advanced version of a microcontroller and a part of an embedded system. In this work, we have designed a robot, which is compact, autonomous and fully functional. It is a proposed model which can be used in such an environment, which may be vulnerable and risky to human beings. It has two types of functions. The functions are light following, obstacle detection. Obstacle avoider light follower robot detects the light (such as the light of flashlight) and follows light on traveling path. Also it can detect the obstacles while it is moving and make the passes by the obstacles. The robot has two light detection sensors which are prepared with LDRs and an infrared obstacle detecting sensor. The sensitivity of the light sensor can be set by using the trim pots.

**1. INTRODUCTION:**

In present times, robotics is one of the important fields in the modern and technological world for the design, construction, operation and application of robots. Before the 20th century this sector was not developed successfully. Today the research, design, and building of new robots used in different domestic, commercial, military sectors are developing the applications of robotics day by day. A Robot, in general an electromechanical & computer programming using power and control machinery device that can perform tasks automatically depending on sensors. This project is designed to build obstacle avoidance at the same time light following a robotic vehicle using an ultrasonic sensor for its movement & LDR module to decide the path to follow according to the light falling on it. An Arduino Uno is used to achieve the desired operation. LDR is used such that the amount of light falling on it from every direction will be calculated and then it will find out the direction from which the light with maximum intensity falls on it. Then it moves in that direction only. The L298N motor driver is used to increase the current by which we can drive the DC motors.

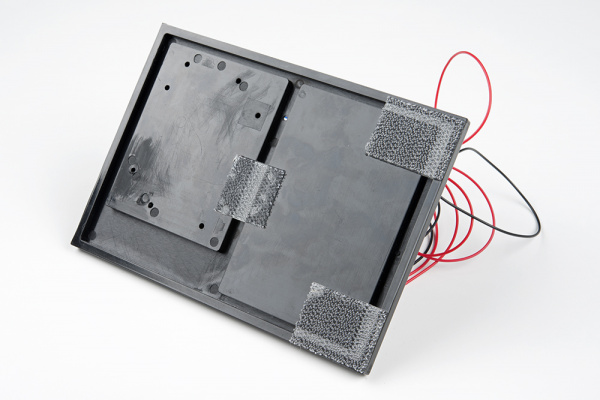
**2. DESIGN OF THE SYSTEM:**

The entire circuit diagram has one major sections which is Light following with the help of LDR module.

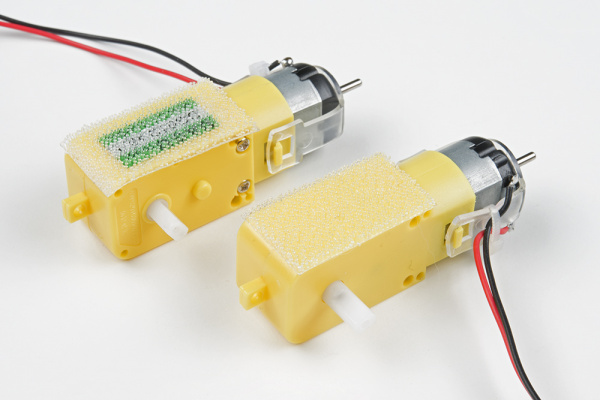
At first the robot gets the power of 1/4W. Then after getting the light source the robot starts to work. Where the LDR sensor gets the highest intensity of light the robot will start to follow that path.

**3. HOW TO SET IT UP THE HARDWARE:**

Using scissors, we cut three strips of Dual Lock that are 1.25 inches (3.2cm) long and 1 inch (2.5cm) wide. Remove the adhesive backing, and attach two pieces to the corners under the baseplate and a third in the center.

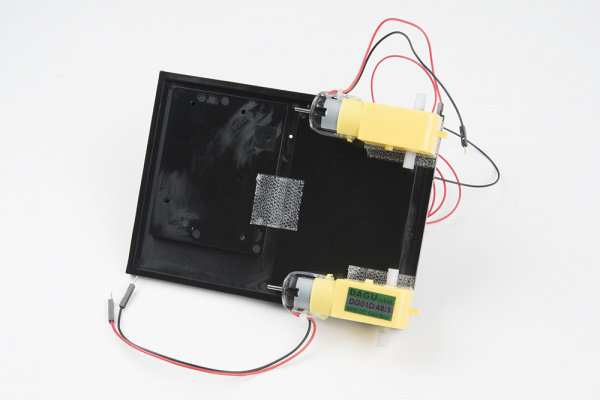


Be sure that your motors are **mirror images** of each other when you attach the Dual Lock.



Press the motors to the baseplate, connecting the two Dual Lock surfaces. Try to get the motors as straight as possible so your robot will drive straight.

The bottom of your baseplate should look like the image below. Remember that the two motors should be mirror images of each other.

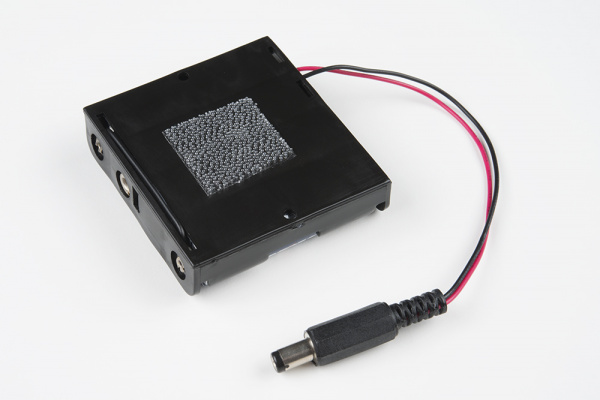


We attached the wheels by sliding them onto the plastic shafts on the gearmotor. The shaft is flat on one side, as is the wheel coupler. Align the two, and then press to fit the wheel onto the shaft.

We clipped the binder clip onto the back end of the robot. This will act as a caster as the robot drives around.



We cut a piece of Dual Lock that is about 1.25 inch x 1 inch (3.2cm x 2.5cm), removed the adhesive backing and attached it to the back of the battery holder.

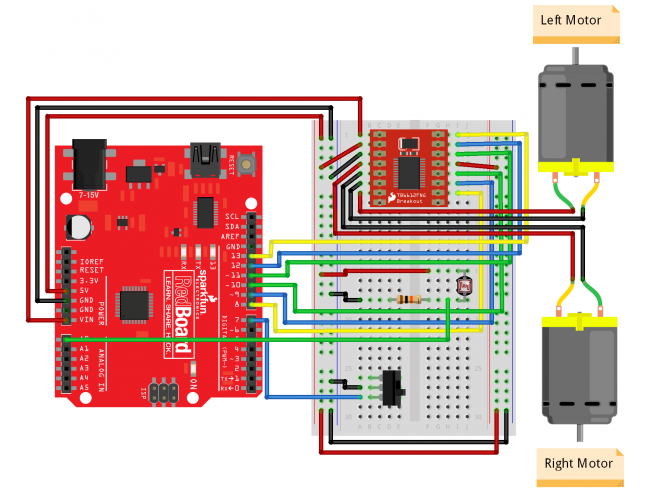


We pressed the battery holder to the baseplate so that the two pieces of Dual Lock snapped together. (Insert the batteries into the holder if you have not done so already.)

**4. CIRCUIT DESIGN AND OPERATION:**

The heart of our module Arduino SparkFun is triggered by, 1500 MAh battery socket. The microcontroller ATMEGA 328p-pu is programmed to control HCSR-04 (Ultrasonic module) & LDR module. As per program HCSR-04 module is connected to the digital pin of Arduino for trig (11) & ECHO (12) & LDR module is connected to A0 ( Analog pins) of Arduino.

The output is sensed by the Arduino and depending on the programming statutes the specific digital data is sent to the motor driver module for different types of rotation using and control motor. Additionally the rotation of the servo motor is a digital pin of an Arduino controlled by the HCSR-04 depending on the obstacle detection. By condition of this sensor & program in Arduino controls the r o t a t i o n & m o v e m e n t of the ro bo t. This circuit works in the monostable mode. In the monostable mode, the 555 timer acts as a "one shot" pulse generator. The pulse begins when the 555 timer receives a signal at the trigger input that falls below a third of the voltage supply. The width of the output pulse is determined by the time constant of an RC network, which consists of a capacitor (C) and a resistor (R). The output pulse ends when the capacitor equals 2/3 of the supply voltage. The output pulse width can be lengthened or shortened to the need of the specific application by adjusting the values of R and C. The output pulse width of time t, which is the time it takes to charge C to 2/3 of the supply voltage.



**5. CODE:**

**Tester Code**

**clear all;**

**clc;**

**close all;**

**a = arduino("COM7","Uno");**

**%% PIN MODES**

**configurePin(a,'D12','DigitalOutput');**

**AIN2 = configurePin(a,'D12');**

**configurePin(a,'D11','DigitalOutput');**

**AIN1 = configurePin(a,'D11');**

**configurePin(a,'D9','DigitalOutput');**

**BIN2 = configurePin(a,'D9');**

**configurePin(a,'D10','DigitalOutput');**

**BIN1 = configurePin(a,'D10');**

**configurePin(a,'D8','DigitalOutput');**

**PWMB = configurePin(a,'D8');**

**configurePin(a,'D13','DigitalOutput');**

**PWMA = configurePin(a,'D13');**

**configurePin(a,'D7','Pullup');**

**SW\_PIN = configurePin(a,'D7');**

**configurePin(a,'A0','AnalogInput');**

**LIGHT\_PIN = configurePin(a,'A0');**

**%% CONSTANTS**

**SEARCH\_DRIVE\_TIME = 200;**

**TURN\_DRIVE\_TIME = 200;**

**MOVE\_DRIVE\_TIME = 300;**

**STOP\_DRIVE\_TIME = 200;**

**NUM\_LIGHT\_LEVELS = 3;**

**%% MAIN CODE**

**light\_levels=NUM\_LIGHT\_LEVELS;**

**if readDigitalPin(a,'D7') == 0**

**drive(0,255)**

**pause(SEARCH\_DRIVE\_TIME)**

**drive(0,0);**

**pause(STOP\_DRIVE\_TIME)**

**light\_levels{0}= readVoltage(LIGHT\_PIN);**

**drive(0,-255)**

**pause(SEARCH\_DRIVE\_TIME)**

**drive(0,0)**

**pause(STOP\_DRIVE\_TIME)**

**drive(255, 0);**

**pause(SEARCH\_DRIVE\_TIME);**

**drive(0, 0);**

**pause(STOP\_DRIVE\_TIME);**

**light\_levels(2) = readVoltage(LIGHT\_PIN);**

**drive(-255, 0);**

**pause(SEARCH\_DRIVE\_TIME);**

**drive(0, 0);**

**pause(STOP\_DRIVE\_TIME);**

**light\_levels(1) = readVoltage(LIGHT\_PIN);**

**max\_light = 0;**

**max\_light\_index = 0;**

**for i=0:1:2**

**if light\_levels(i) > max\_light**

**max\_light = light\_levels(i);**

**max\_light\_index = i;**

**end**

**fprintf("");**

**fprintf("Max light: ");**

**fprintf(max\_light\_index);**

**end**

**if max\_light\_index == 0**

**fprintf("Chasing light to the left");**

**drive(-100, 255);**

**pause(TURN\_DRIVE\_TIME);**

**drive(255, 255);**

**pause(MOVE\_DRIVE\_TIME);**

**drive(0, 0);**

**delay(STOP\_DRIVE\_TIME);**

**elseif ( max\_light\_index == 1 )**

**fprintf("Chasing light straight ahead");**

**drive(255, 255);**

**pause(MOVE\_DRIVE\_TIME);**

**drive(0, 0);**

**pause(STOP\_DRIVE\_TIME);**

**else**

**fprintf("Chasing light to the right");**

**drive(255, -100);**

**pause(TURN\_DRIVE\_TIME);**

**drive(255, 255);**

**pause(MOVE\_DRIVE\_TIME);**

**drive(0, 0);**

**pause(STOP\_DRIVE\_TIME);**

**end**

**else**

**drive(0, 0);**

**end**

**%% If switch is not flipped, do nothing**

**Left Motor Function**

**function l\_m(motorspeed)**

**a = arduino("COM7","Uno");**

**if (motorspeed > 0)**

**writeDigitalPin(a,'D10',1);**

**writeDigitalPin(a,'D9',0);**

**elseif (motorspeed < 0)**

**writeDigitalPin(a,'D10',0);**

**writeDigitalPin(a,'D9',1);**

**else**

**writeDigitalPin(a,'D10',0)**

**writeDigitalPin(a,'D9',0)**

**end**

**writePWMVoltage(a,'D8',abs(motorspeed));**

**end**

**Right Motor Function**

**function r\_m(motorspeed)**

**a = arduino("COM7","Uno");**

**if (motorspeed > 0)**

**writeDigitalPin(a,'D11',1);**

**writeDigitalPin(a,'D12',0);**

**elseif (motorspeed < 0)**

**writeDigitalPin(a,'D11',0);**

**writeDigitalPin(a,'D12',1);**

**else**

**writeDigitalPin(a,'D11',0)**

**writeDigitalPin(a,'D12',0)**

**end**

**writePWMVoltage(a,'D13',abs(motorspeed));**

**end**

**Drive Function**

**function drive (leftSpeed, rightSpeed)**

**l\_m(leftSpeed);**

**r\_m(rightSpeed);**

**end**

**HOW IT WORKS:**

When the switch is OFF, the robot will not move. When you turn the switch ON, the robot will turn left and right, taking light measurements at each extreme. It will also take a light measurement from the center.The robot turns to the direction with the most light and moves forward a small amount. It then repeats the pattern of looking for light and moves toward the direction of brightest light.

**6. THE PROBLEM WE HAD:**

We realized that trying to direct the robot with a flashlight or other light source can be difficult. Reflected light from the wheels can sometimes be brighter, for instance, than reflected light on the ground. It can take some patience to get the robot to move the way you want.

**7. COMPONENTS USED:**

**Twin Wheels**

**BreadBoard**

**RedBoard**

**Jumper Wires**

**Battery Holder**

**Batteries**

**Motor Driver**

**Photo Resistor**

**10k ohms Resistor**

**Switch**

**GearMotors**

**8. APPLICATION:**

1. Automatic Street lights follower.
2. Alarm devices for security purposes.
3. This technology can also be used to measure light intensity for applications that require greater precision.

**9. CONCLUSION:**

We made the robot named Light Following Robot. In the modern era of science and technology it is necessary to reduce the manpower and increase the use of instruments. Through this hardware project we have learnt about many new projects and we have also developed our skill in programming. It was great scope for us to take the technology a little bit far.