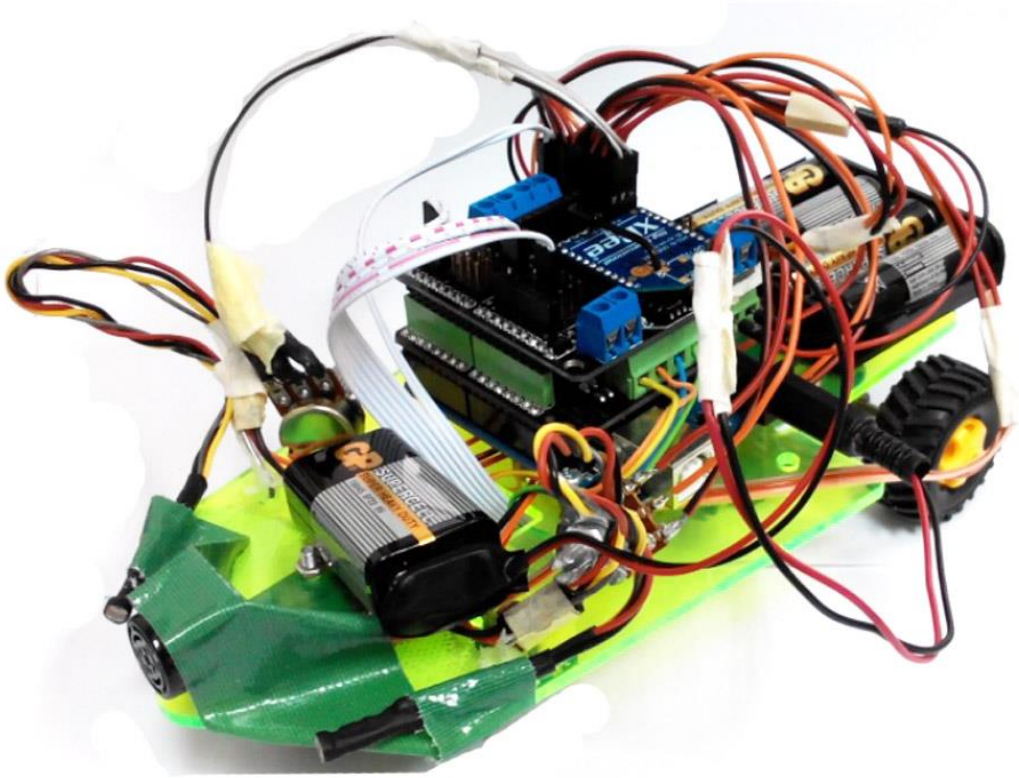
Building a Line Follower Robot
Rev2.0: Guidebook

Sem 2 22/23

Mechatronics Workshop



Wish you have learned something new.
Enjoy and give it a try!

Acknowledgement

THIS GUIDE BOOK IS INTENDED FOR INTERNAL PURPOSES OF IIUM MECHATRONICS WORKSHOP COURSE (MCTE2101/ MCTA1101).

Supervisor
Assist Prof Dr Zulkifli Zainal Abidin

Authors
Muhammad Irsyad bin Sahalan
Muhammad Hilmi bin Idris

Editors
Muhammad Amirul Bin Abdullah
Zakariya Bin Zainol

Year: 2015

Edited by
Dr Khairul Affendy Md. Nor
Dr Aimi Shazwani Ghazali
Dr Nur Liyana Azmi
Dr Mohd Asyraf Mohd Razib
Dr Hasmawati Antong
Dr Nor Hidayati Diyana Nordin

CONTENT

PART A : Introduction

- Project Outcomes
- Description of the Project
- Robot's Operation
- Assessments
- Bill of Materials
- Tools
- Programming Software

PART B: Build the Base

PART C: Power Management

PART D: Actuators

- I. Motor Driver
- II. Servo Motor

PART E: Sensors

- I. Digital Infrared Sensor
- II. Auto-calibrated Line Sensor

PART F: Communication

- Refer H05 Bluetooth Module

PART G: Integration

PART H: References

PART A: INTRODUCTION

Assalamualaikum!

Now you reach final part of your Mechatronics Workshop where you are going to spend your time building your Line Follower robot for your final project. The built robot will compete and tested on the competition day! This guide will provide you with a step by step on how to make this robot to a certain extend. Included with this guide is the instruction including some issue and troubleshooting technique.

Currently, we are developing this tutorial, any suggestion/comment that would make it easier to understand is highly appreciated! ☺

OK! Let's start!

Project Outcomes

- Introduce students to Mechatronics concepts and systems.
- Enable students to have basic skills in handling mechatronics components.
- Expose students to hardware and software integration for mechatronics systems development.
- Design and implement simple mechatronics systems in groups and demonstrate the work with the report.

Bill of Materials

No	Items	Quantity
1	Arduino Uno	1
2	Car chasis	1
3	Tamiya tire and twin gearbox	1
4	4 x AA Battery Holder	1
5	2A Motor Shield	1
6	Arduino Uno Expansion Shield	1
7	Castor wheel	1
8	Spacer set (10mm)	2
9	Spacer set (15mm)	6
10	RC Micro Servo motor	1
11	Digital Infrared sensor	1
12	Auto-calibrated line sensor	1
13	XBee/ BlueBee Bluetooth module	1
14	Bread board	1
15	USB B type cable	1
16	Jumper (male to male)	1
17	SKXBEE	1
18	Mini USB cable	1

Tools

- Pliers
- Soldering iron
- Solder
- Screwdriver
- Wire strippers
- etc.

Programming Software

- Arduino IDE programmer (www.arduino.cc)
- XCTU
- Teraterm

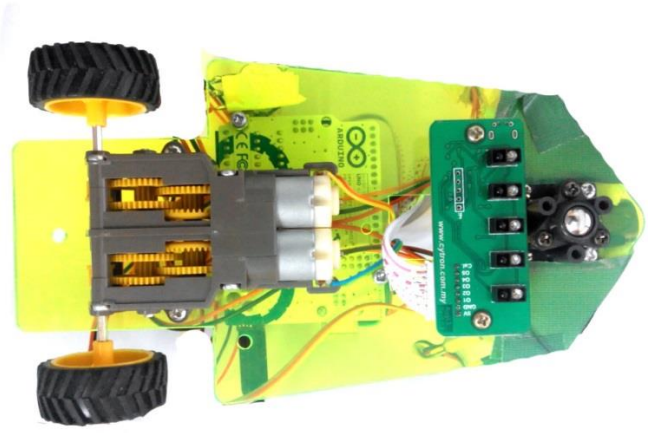
PART B: BUILD THE BASE

To ease the development process, we arrange the contents of this tutorial according to the flow of making the robot. Let start!

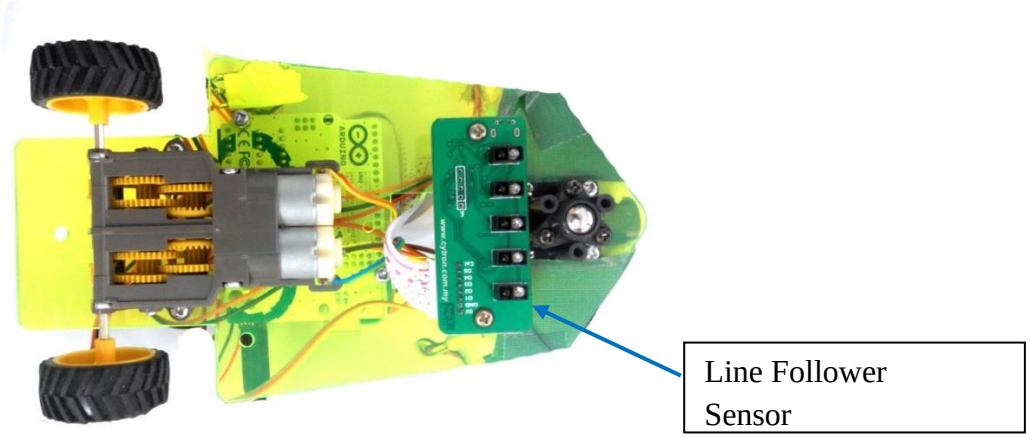
Chassis->motor gearbox tyre->caster wheel->Arduino Uno->Motor Driver->Battery holder->Expansion shield->line sensor->servo->IR sensor

Main body frame

The platform that the robot is going to be built on is a premade chassis body with a gearbox, ball caster (to allow steering), and tires.



Chassis:
We are going to be using a premade chassis with precut holes for mounting various pieces of hardware. For now, we will be using the mounting holes that will hold the gear box and the set of 4 holes for the ball caster.

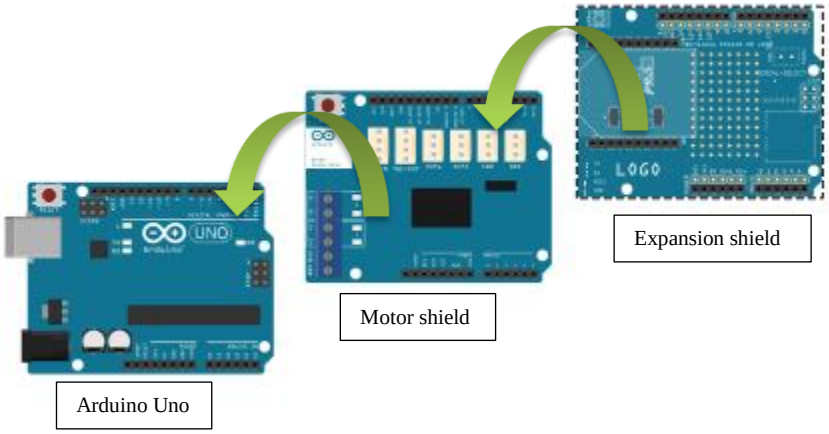


Ball Caster: This will be the front-turning wheel that will allow the robot to swivel one way or the other without scraping the chassis. The ball caster is easy to mount; add the spacer to give it enough height so that the platform stays level with the ground (or you can leave it out as long as the sensors that will be mounted are not too close to the ground). Then add the ball caster and use the bolts and nuts to mount it to the chassis.

Gearbox and wheels: This gearbox has 3 different speed settings and we will be using the middle speed (type C). There are four large gears and one very small gear that we will be using to get the slow speed. There is a diagram that came with the gearbox that shows how to mount the gears in what order. Once the gearbox is put together, mount the wheels by sliding them on the end of each axel.

Microcontroller Arduino + Shields

Stack sequence of Arduino and shields

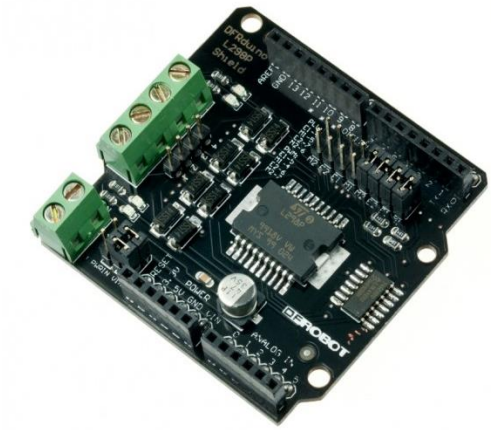


http://arduino.cc/en/Main/arduinoBoardUno

Descriptions: The Arduino is already pre-built out of the box; that will be used later when we finally build the program and test out the robot. This is the brain of the robot; this will hold the code that will allow the robot to follow the line and this will have most of the signal connections plugged in from the motor control board and sensors.

.

1. MOTOR SHIELD (Model : L298P)



Descriptions: This motor shield allows Arduino to drive two channel DC motors. It uses a L298N chip which deliveries output current up to 2A each channel. The speed control is achieved through conventional PWM which can be obtained from Arduino's PWM output Pin 5 and 6. The enable/disable function of the motor control is signalled by Arduino Digital Pin 4 and 7.

The Motor shield can be powered directly from Arduino or from external power source. It is strongly encouraged to use external power supply to power the motor shield.

Motor Terminal: Two DC motors are connected to blue motor terminals. The male header behide the terminals are the same as the
motor terminals.

Motor Terminal

PWRIN : The motors can be powered by external power supply when the motor current exceeds the limits provided from the Arduino. The switch between external and Arduino power is implemented by two jumpers. For our Mechatronics Workshop, I would recommend you to **use external power supply** from 6V battery provided by the workshop.

The motors are powered by external power supply

The motors are powered by Arduino power supply

Table: Pin Configuration for motor control

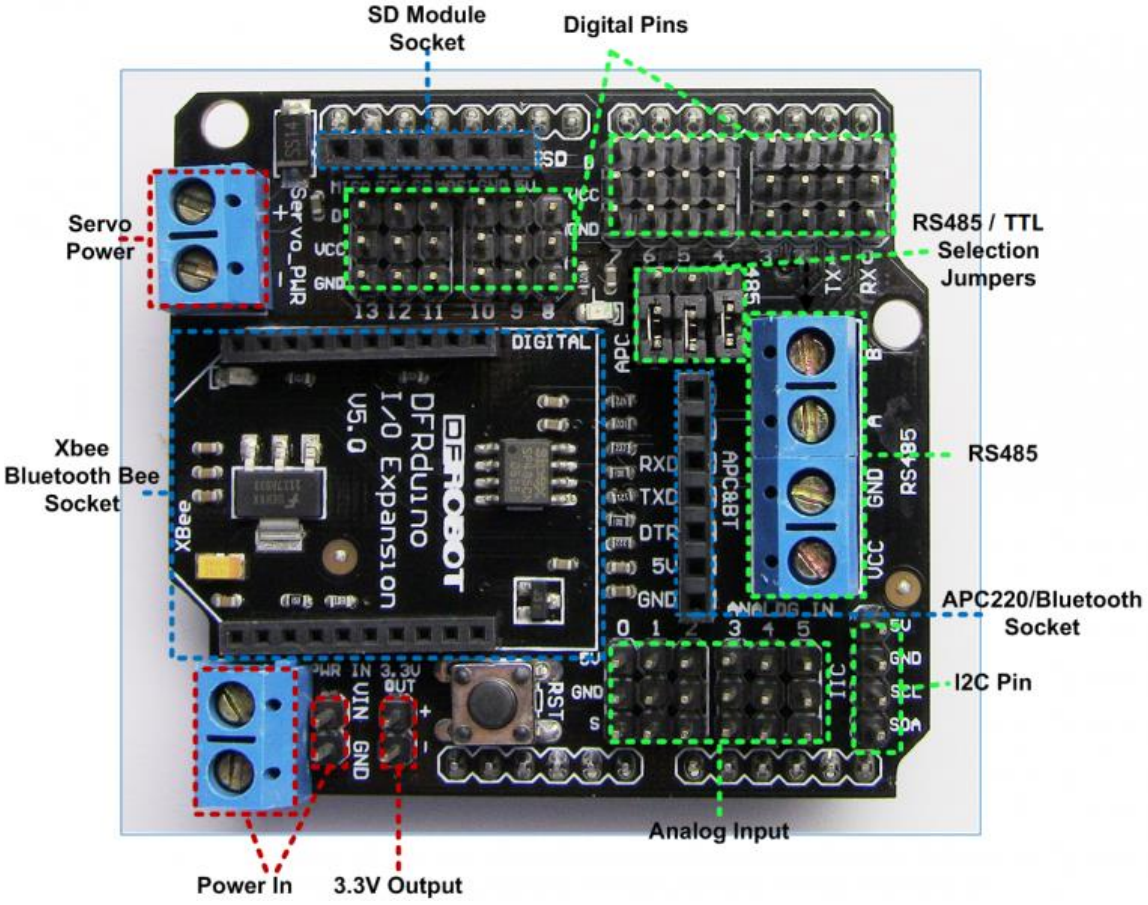
"PWM Mode"

Pin	Function
Digital 4	Motor 1 Direction control
Digital 5	Motor 1 PWM control
Digital 6	Motor 2 PWM control
Digital 7	Motor 2 Direction control

2. IO EXPANSION SHIELD FOR ARDUINO

http://www.dfrobot.com/wiki/index.php/IO_Expansion_Shield_For_Arduino(V5)_(SKU:_DFR0088)

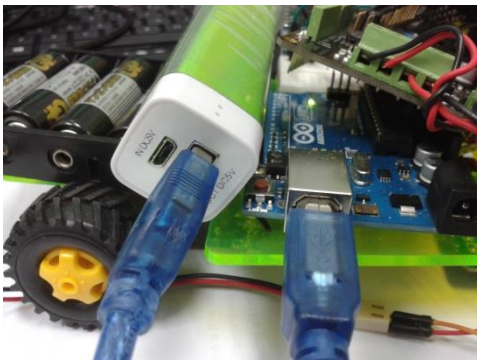
Descriptions: Use to ease connection of Xbee Module and other analog and digital sensors. Every analog or digital pins come with their own voltage and ground pin. Thus, you don't need to use a separate breadboard in order to deal with those pins anymore.



PART C: POWER MANAGEMENT

Note: Please ensure that you follow the instructions on this section as to avoid damaging the components.

For this robot, you will require to have two power supplies; power bank and AAx4 batteries. These power supplies are common so I bet everyone afford to buy them. The power bank is used to power up Arduino Uno via USB type B port. Arduino Uno requires 5V to operate so a typical DC5V power bank should work well disregard of the mAh value.

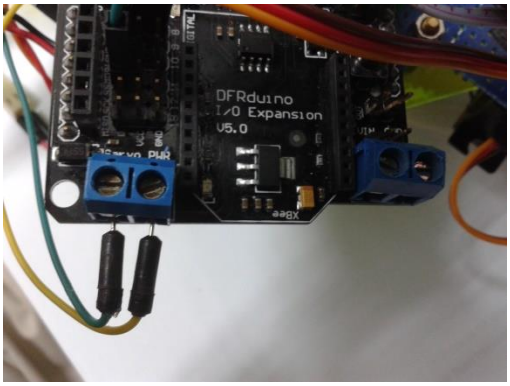


On the other hand, the AAx4 battery should be connected to the 2A motor driver shield and to the servo power on the expansion shield. The reason for having another power supply for the motors is to provide enough power to run the motors. The Tamiya dual DC motor runs on 2A while servo motor ….. Our experience tells us that whenever we tried to power motor directly from the 5V supply of Arduino Uno, it pulls away the power causing the Arduino insufficient to run and other connected components to disrupt.

Another thing that you have check is the power jumper position. There are two ways of providing power to the motors: VIN or PWRIN. VIN means to take the power from the Arduino power supply while PWRIN is to acquire from the external power supply. For this project ensure is the jumper is on the PWRIN position.

The motors are powered by external power supply

The motors are powered by Arduino power supply



PART D: ACTUATORS

2A Motor Driver

As for this part, you can just refer back to the description given in the Introduction

Connection

Apart from **mounting** and **power source**, be sure to connect the two pairs of wires from the motors to the 2A Motor Driver shield. Below is an example of DC motor to the 2A motor shield. You may have to switch among these connection to get it tally with the code.

	Wire	Terminal
Motor 1	red	M1+
	black	M1-
Motor 2	red	M2-
	black	M2+

Sample code

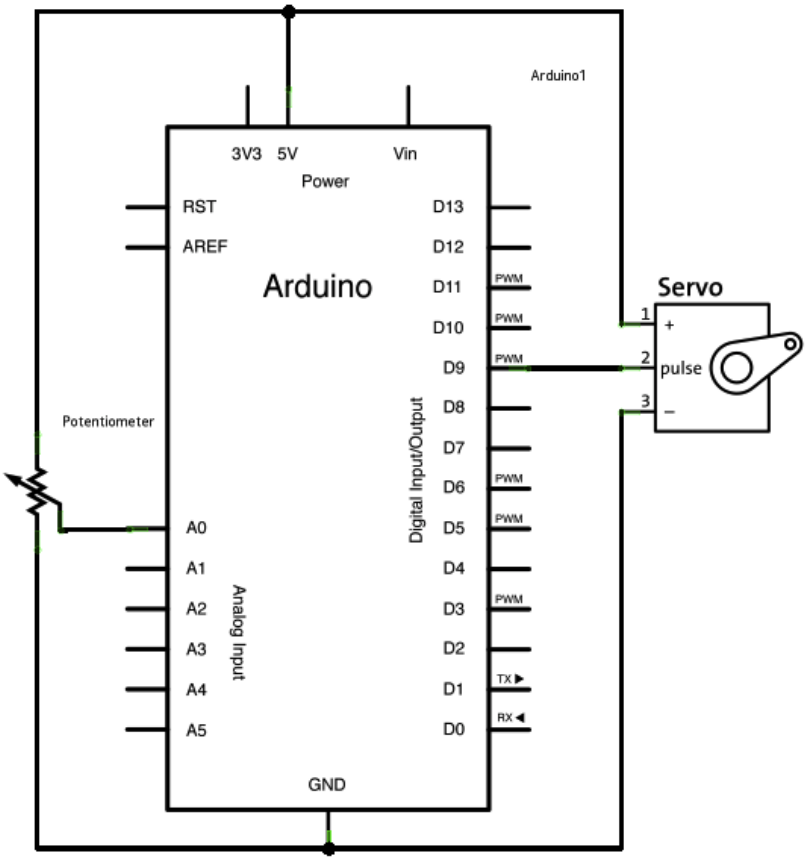
```
//pin initialization for motor driver
int A_IA = 5;
int A_IB =  4;  //motor 2
int B_IA =  6;
int B_IB =  7;  //motor 1

void setup()
{
       pinMode(A_IA,OUTPUT);
       pinMode(A_IB,OUTPUT);
       pinMode(B_IA,OUTPUT);
       pinMode(B_IB,OUTPUT);
}

void loop()
{
       forward(200, 200);
       backward(200, 200);
       right(200, 100);
       left(100, 200);
       stop();
}

//direction movement
void forward(int pwm_left,int pwm_right){
//right tyre
  analogWrite (A_IA, pwm_left );   //speed control
  digitalWrite(A_IB, HIGH      );   //direction
```

```
//left tyre
  analogWrite (B_IA, pwm_right);   //speed control
  digitalWrite(B_IB, HIGH      );   //direction
}

void backward(int pwm_left,int pwm_right){
//right tyre
  analogWrite (A_IA, pwm_right);   //speed control
  digitalWrite(A_IB, LOW       );   //direction
//left tyre
  analogWrite (B_IA, pwm_left);    //speed control
  digitalWrite(B_IB, LOW       );   //direction
}

void right(int pwm_left,int pwm_right){
//right tyre
  analogWrite (A_IA, pwm_right);   //speed control
  digitalWrite(A_IB, HIGH       );   //direction
//left tyre
  analogWrite (B_IA, pwm_left   );   //speed control
  digitalWrite(B_IB, HIGH        );   //direction
}

void left(int pwm_left,int pwm_right){
//right tyre
  analogWrite (A_IA, pwm_right  );   //speed control
  digitalWrite(A_IB, HIGH        );  //direction
//left tyre
  analogWrite (B_IA, pwm_left);  //speed control
  digitalWrite(B_IB, HIGH      );   //direction
}

void Stop(){
//right tyre
  digitalWrite(A_IA, 0);   //speed control
  digitalWrite(A_IB, HIGH);   //direction
//left tyre
  digitalWrite(B_IA, 0);   //speed control
  digitalWrite(B_IB, HIGH);   //direction
}
```

RC Micro Servo Motor

Intro

Connection



Sample codes

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott http://people.interaction-ivrea.it/m.rinott

#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
```

```
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value
between 0 and 1023)
  val = map(val, 0, 1023, 0, 179);     // scale it to use it with the servo (value
between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the
scaled value
  delay(15);                           // waits for the servo to get there
}
```

PART E: SENSORS

Digital Infrared Sensor

Intro

Connection

Infrared Sensor	Arduino
Blue	Ground
Brown	VCC (+6V to +36V)
Black	Output signal

Sample Codes

```
/*
  Digital Infrared Sensor

  Displays the digital value reading through the serial communication.
*/

// set pin number
const int sensorPin = 2;     // the number of the pushbutton pin

int input = 0;  //initialize the sensor input variable

void setup() {
  // initialize serial communication to use serial monitor
  Serial.begin(9600);
  // initialize the pushbutton pin as an input:
  pinMode(sensorPin, INPUT);
}
```

```
void loop(){

  input = digitalRead(sensorPin);  // read and store the sensor value in input
  Serial.println(input);  // print out the digital reading from the variable;
  delay(10);
}
```

Auto-Calibrated Line Sensor

Intro

3. LSS05 -DIGITAL LINE DETECTION SENSOR

Descriptions: This newly developed Auto-calibrating line sensor is super easy to use. Come with 5 pairs of IR transmitter and receiver, it can covers line detection of 1cm to 3cm wide, dark color or bright color line. With 1 press, it will start \"recognizing\" the surface under it, calibrating the threshold between dark and bright. it takes 4 to 5 seconds only. After that, it is done, being stored in internal non volatile memory, it will still \"recognize\" the line even after power off and on again. Come with connector and wires.

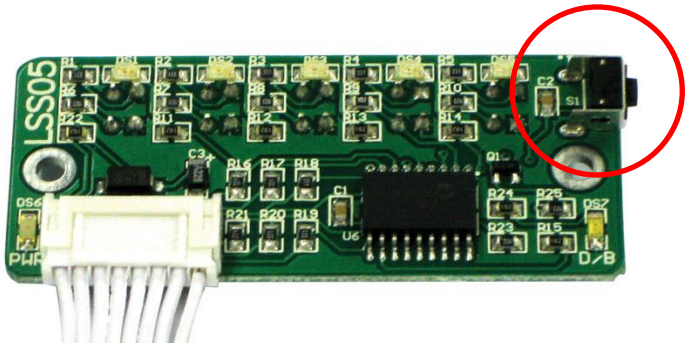
Features:

- 5V power
- 5 digital output representing logic of 5 IR sensor
- 1 press to start calibration
- 2 press to toggle logic into Dark On mode, to sense dark line
- 3 press to toggle logic into Bright On mode, to sense bright line
- Calibration button for easy of calibration

Calibrating Line Following Sensor LSS05

Why does my robot not run according to line although I have already programmed it? This is because the line sensor may not giving the correct value. So how do we calibrate it? Let’s see how to deal with it. I guarantee you are going to love it.

Check for the RED circle in the following figure. This is the push button on LSS05 that we are going to use to calibrate the LSS05 line sensor. To calibrate the line sensor, you only need to push it ONCE, then swing your LSS05 sensor left and right repeatedly with the sensor facing the line. Please keep the distance between robot and floor constant during calibration. Do not lift the base. The sensor is automatically calibrated so there is no need for us to worry about the calibration procedure. The indicator LEDs will blink while calibrating. After the indicator LEDs stop blinking, it's DONE. Now it has learned to 'see'.



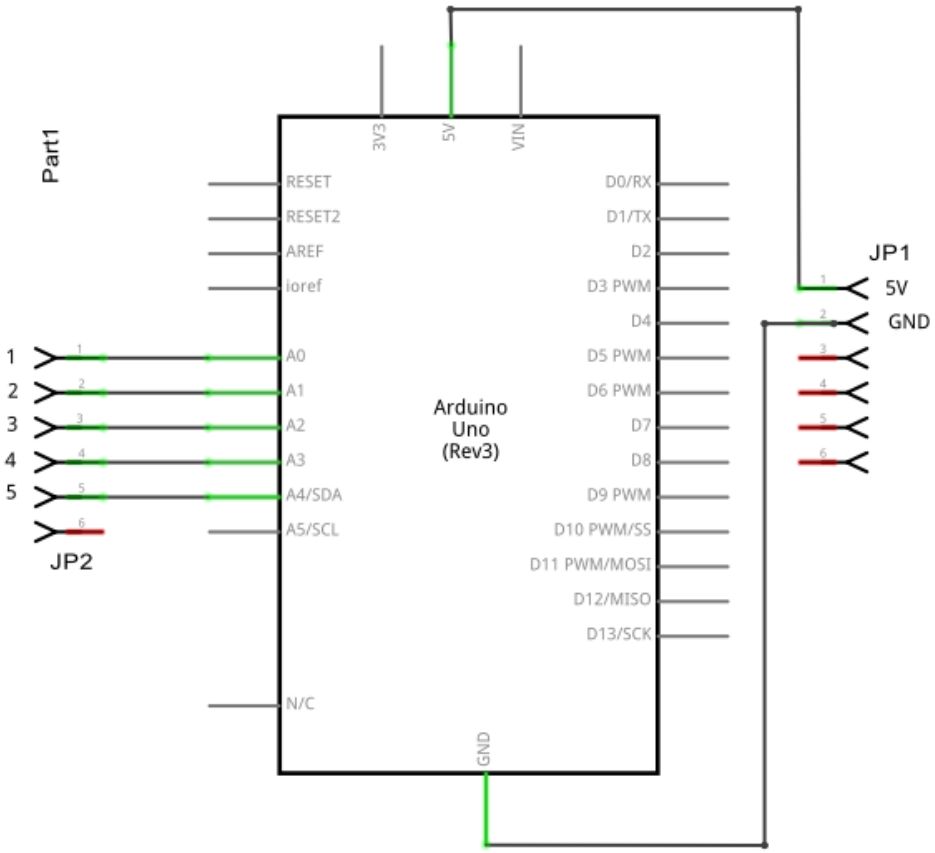
Well, it's not finished yet. The sensor also comes with extra 2 modes, namely dark line following modeand bright line following mode. It's easy to change the mode.

Dark line following mode (We will use this mode!!) – Press the button twice within 1.5 second. Then the sensor indicator LEDs will light up when it detects a dark surface.

Bright line following mode – Press the push button 3 times within 1.5 second. The sensor indicator LEDs with light up when it detects a bright surface.

Now your mobile robot is fully functional and ready to go. Put it on the line and test it. Sometimes you might need to recalibrate the sensor if the sensor senses dirt on the line surface during previous calibration.

Connection



Sample Codes

```
const int pin_1 = A0;
const int pin_2 = A1;
const int pin_3 = A2;
const int pin_4 = A3;
const int pin_5 = A4;  //pin initialization for sensors

 int DS_1 = 0;
 int DS_2 = 0;
 int DS_3 = 0;
 int DS_4 = 0;
 int DS_5 = 0;                //variables initialization

void setup() {
 Serial.begin(9600); //initialize serial communication to use serial monitor
(9600=baud rate)

 pinMode(pin_1     , INPUT);
 pinMode(pin_2     , INPUT);
 pinMode(pin_3     , INPUT);
 pinMode(pin_4     , INPUT);
 pinMode(pin_5     , INPUT);  //declare pins as INPUT for sensors
}
```

```
void loop() {
 DS_1 = digitalRead(pin_1);
 DS_2 = digitalRead(pin_2);
 DS_3 = digitalRead(pin_3);
 DS_4 = digitalRead(pin_4);
 DS_5 = digitalRead(pin_5);   //declare variables equal to the reading of the sensors

 Serial.print(DS_1);
 Serial.print("\t");
 Serial.print(DS_2);
 Serial.print("\t");
 Serial.print(DS_3);
 Serial.print("\t");
 Serial.print(DS_4);
 Serial.print("\t");
 Serial.println(DS_5);        //display the reading of the sensors at the serial
monitor
}
```

PART F: COMMUNICATION

Refer module H05 Bluetooth Module

Android Application

Intro

Yup, you bet so. How cool is that to control your robot from your phone!
This tutorial will only cover for Android. If you are an IPhone or Windows user, you can find other apps that suits.

Setup

1. Go to Google Playstore and search for ArduinoBluetooth Controller by Estacado's ltd.
2. Install the apps.
3. Open the application. While you BlueBee is turn on, pair your device to the BlueBee.
4. Enter the password.
5. You can choose any of the interface. For starter, I suggest you use the controller mode interface.
6. Tap on your phone@ tab setting button. Select Set Command option in the list.
7. Configure you button here according to your preference.
8. Once finish, tap the return button and you are ready to go.

PART G: INTEGRATION

Do you expect us to show the next steps? SORRY!

No, we not going to spoon feed you brothers and sisters. This is where your understanding and skills in programming take place. Use your creativity to assemble all the bits and pieces of the robot's program into a complete working robot. Use what you had learned in the past from your programming lesson.

Well the only clue that we can offer you:

- Use any control statement (e.g. if else, for loop, while, do while, case) to organize your program flow.
- List down all the pin connection so that you won't get confused.
- Arrange your pin connection wisely.

Take your time in making the program. Good luck in making you robot. See you in the competition day!

Best of luck!
MCT Workshop Instructors

PART H: REFERENCES

- http://www.arduino.cc/
- http://cytron.com.my/
- http://www.dfrobot.com/
- https://www.sparkfun.com
- https://www.pololu.com