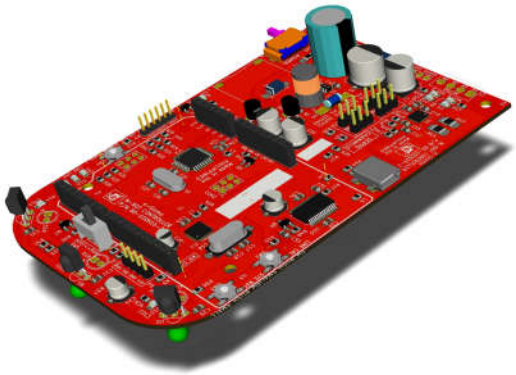
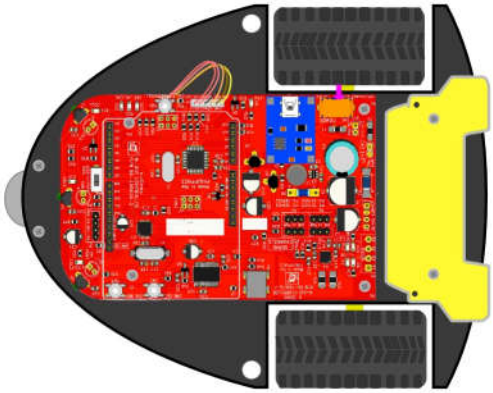
PBOT 2018 MOBILE ROBOT CONTROLLER

Technical Manual Rev 1r0





e-Gizmo PBOT 2018 Mobile Robot Controller is a mobile robot controller with on-board MCU ATmega168P, 3-Channel collision sensors, 3-Channel line sensors with line calibration, Motor driver, 4 servo connections, built in DC-DC booster from 3.7V battery Li-on to 7.2V DC output, optional for 7v2 Ni-MH battery. Shield style interface connectors allows you to use a wirelesss controller shields like Bluetooth, Wi-Fi and any other device for interface.

This robot kit is absolutely for students, hobbyists, and for researchers that would like to learn basic Arduino programming. For Line following, Collision avoidance, solving the maze track, sumo fight mobile robot and other application that is suitable for PBOT controller.

Features:

- Operates on single Lithium Ion (included) battery
- Efficient Motor Driver circuitry enables the operation of motor to its maximum power at the same time eliminating the need for an heatsink
- On board 4-channel Servo Motor driver port
- On board Li-ion battery Charger. You can charge the battery even using you old cellphone charger.
- Sensors managed by independent controller
- Collision sensor are pulsed to saved battery power.
- Line sensors are digitally calibrated, making calibration process delightfully easy
- Arduino Compatible
- Program it using Arduino IDE 100% code compatible with Arduino UNO(1)
- ATMEGA168 microcontroller sporting 16K User program memory space.
- Arduino external I/O pin layout allows you to plug in any Arduino compatible shield(2).

Notes:
1: PBOT 2018, with its ATMEGA168 microcontroller, offers memory space (FLASH/RAM/EEPROM) half of what Arduino Uno offers. But this hardly matters. 16K memory space is more than enough to fit even fairly advanced Mobile Robot application.

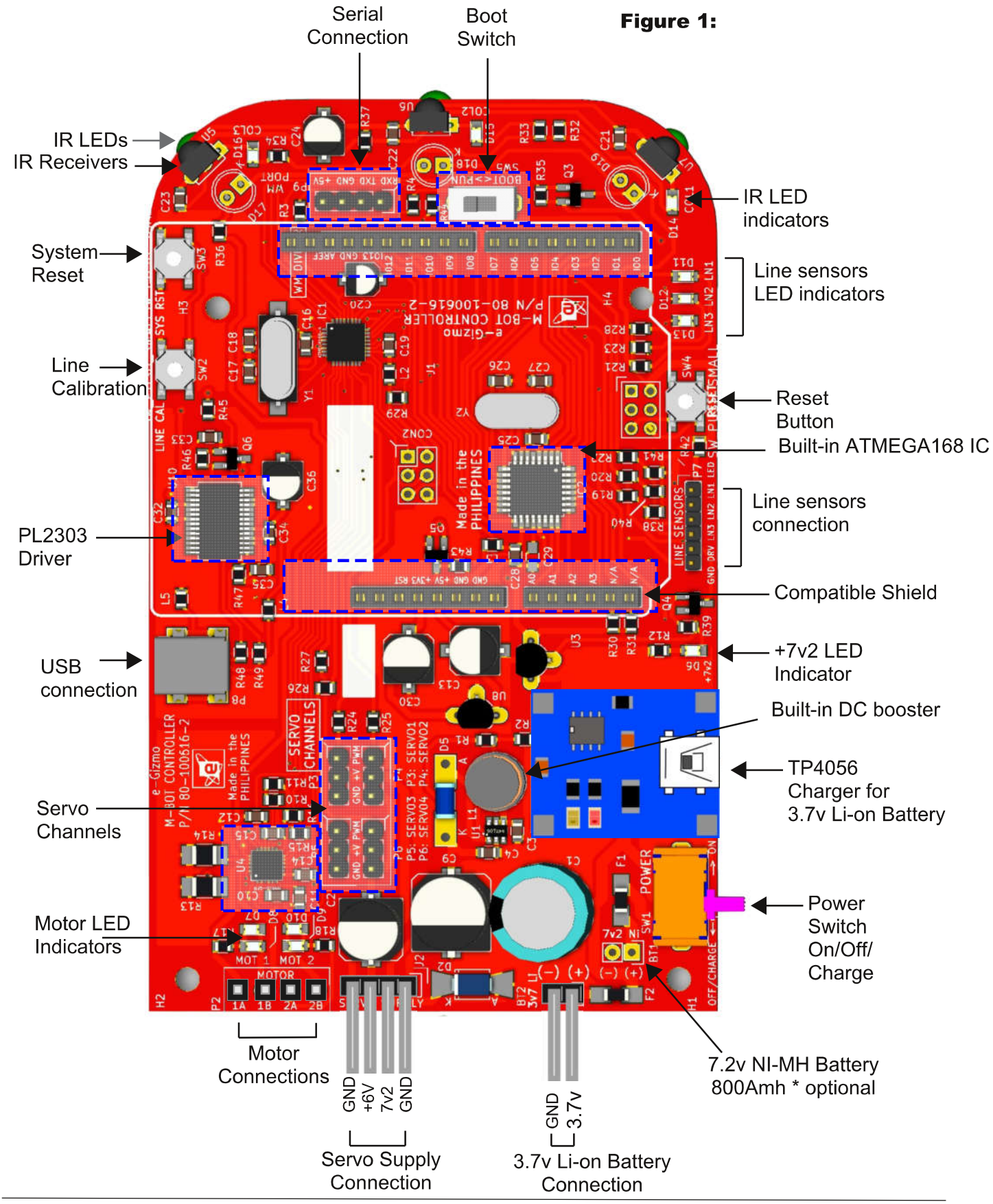
2: Analog pins A4 and A5 are used to interface with the system controller, hence are not available for user shields/application.

General Specifications:

Battery: 3.7v Li-on Rechargeable Battery
Optional: 7.2V NI-MH 800mAh rechargeable
With boost dc voltage output: 7.2vdc
On Board Peripherals:
- IC ATmega168 w/ 16KB Flash Memory
- IC A3966 Dual Full-bridge PWM Motor driver
- 2-Ch DC Motor 6V 1.5A
- 3-Ch IR Line Sensor -CNY70 sensor, 10mm range
- With Battery charger circuit for 3.7v

PCB Dimension: 62X67mm

Figure 1:



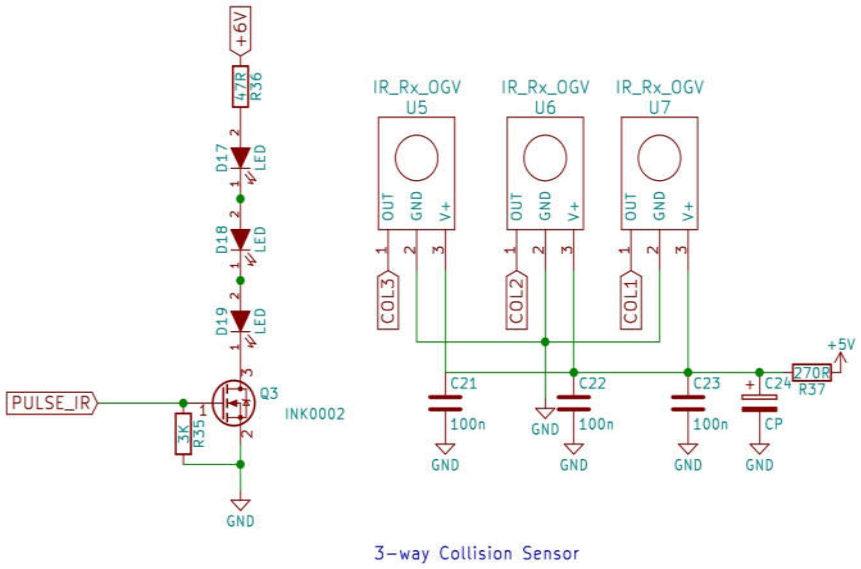
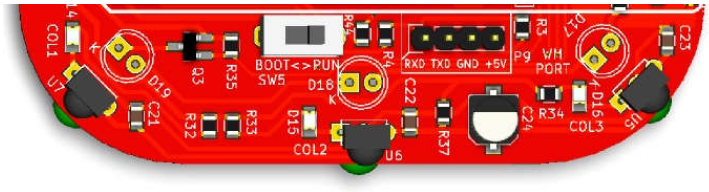
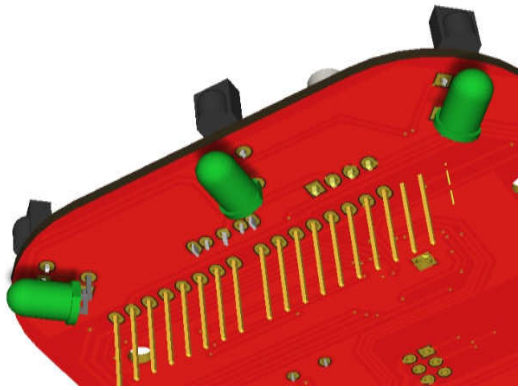


Figure 2: Schematic of three channel collision detector.

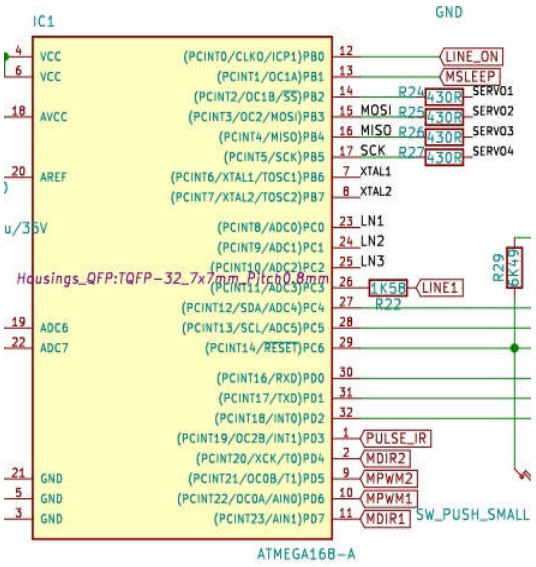




The 3 channel collision sensor detects objects reflected the beam send by an array infrared emitter. The beam will reflect back in a sec and gives the mcu an output. If the objects comes close on the sensors detection view, the beam refelects back to trigger the detector.

The collision sensor is an IR or infrared reflection sensor that is able of detecting up to 220mm (20 cm) away from the sensor tip. Detection distance is a bit dependent on the colour of the reflecting surface. Dark and dull surface generally results in shorter detection range.

U5 to U7 is the IR detector array and D17 to D19 forms the emiiter array. The IR detectors will respond only to IR beams that are chopped at a 40 kHz rate.



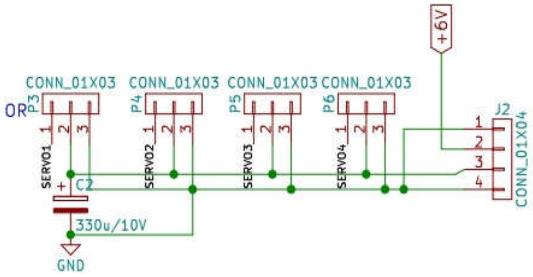


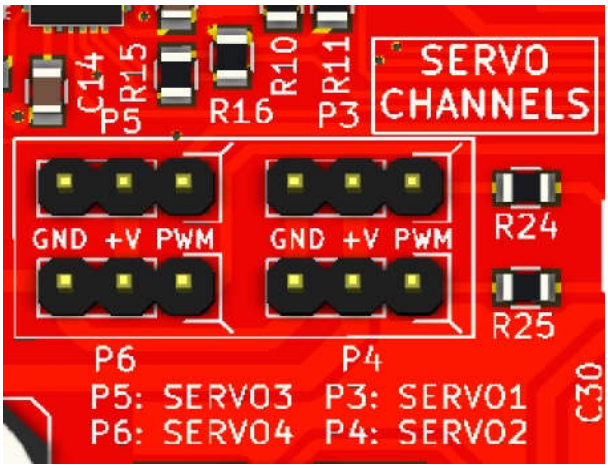
Figure 3: Servo Channel schematic diagram.

PBOT2018 controller has four servo channels on board with built-in 3 pins male connector to easy connect the servo motors. P3 to P6 is the pin name of servo.

Note if you are using SG-90 and other servo motors when you connect it directly to the PBOT2018, you should have them powered by a seperate supply from the PBOT2017 controller board rather than from the boost converted voltage from 3.7v battery, because servo motors can use a bit of current, to cause a voltage distortion or power off/ shutdown the whole system.

But using 7v2 Ni-MH battery it can supply all the PBOT2018 controller with servo motors like ZS-F135MG (e-GRA) and SG -90. We recommend to use serparate supply for the servo if you are using high torque type of motors from MG-995 and higher.

See Page 19 for proper use of servo supply source.



Servo channel illustration.

Servo Channels

P3: Servo 1
P4: Servo 2
P5: Servo 3
P6: Servo 4

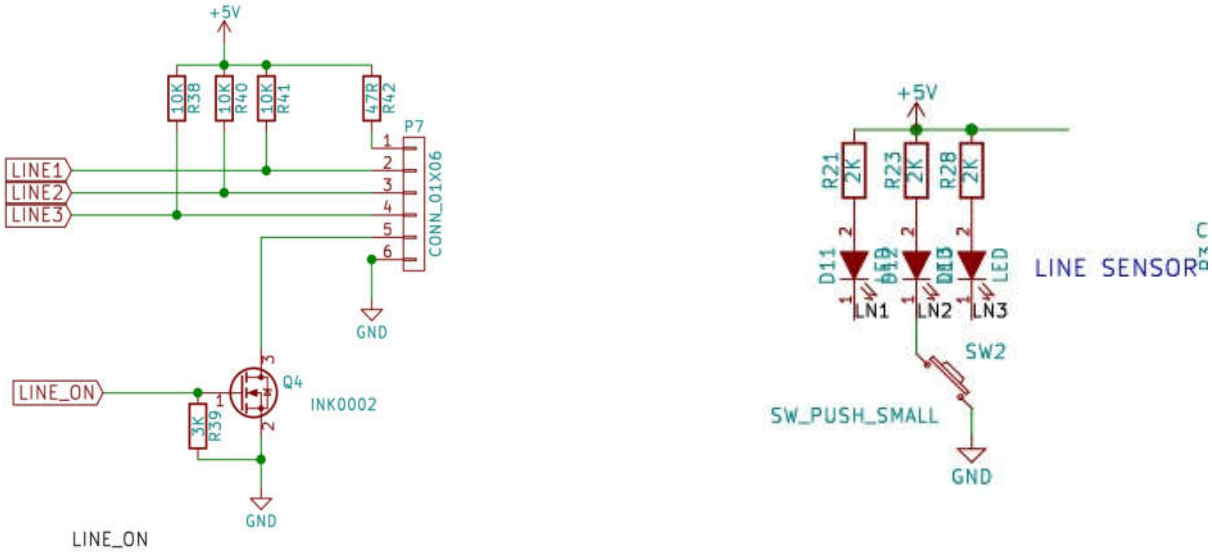
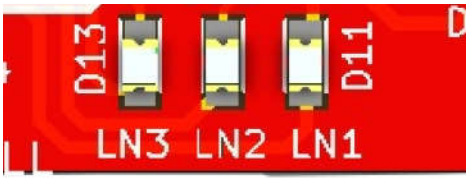
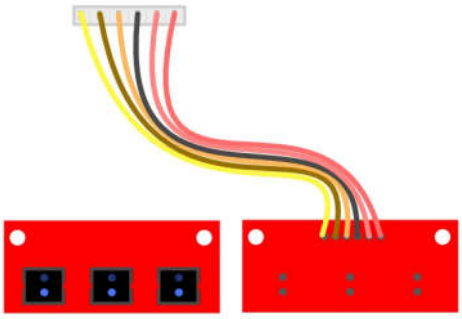


Figure 4: Circuit Diagram for 3-channel line sensor.



The sensitivity of the three comparators can be seperately set by adjusting their reference voltage through their corresponding adjust trimmers.

The process is trouble-free and straightforward. The reference voltage is fed to the positive (+) input of the comparator. Then if the analog input fed through the - input exceeds the reference voltage, the comparator output switches to logic low. Or else, it assumes a logic HIGH state.



An analog comparator mostly converts analog voltage appear at its input into a single bit digital logic signal.

3-Channel analog comparator is a common analog interface circuit. It can be used as well with other sensors with 0-5VDC output range functioning as a single bit analog to digital converter (ADC).

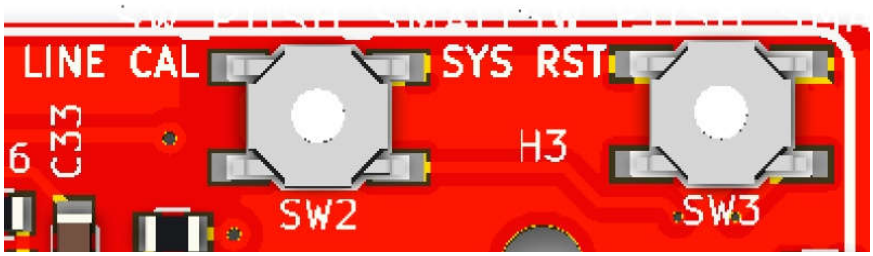


Figure 5: Line Calibration and System Reset.

Reading the 3channel Line sensors:

Calibration Instructions:

For the new eGizmo PBOT 2018 Mobile Robot Controller features first thing to do is to calibrate the line sensors. How to calibrate it? Read the ff.

1. After uploading your code for linesensors.
   Turn OFF the POWER switch.

2. Place the eGizmo PBOT 2018 controller to the Black track then Press and Hold LINE CAL and SYS RST, while pressing and holding the buttons, Turn ON the POWER Switch. LN2 (D12 LED indicator) is ON.

3. First RELEASE the SYS RST followed by LINE CAL. Make sure the 3CH Line sensors are faces on the Black track and you will see the LN1 and LN3 (D11 and D13 LED indicators) are
   Turn ON and LN1 is blinking. Now Press LINE CAL once for the Black color calibration.

4. Next, if the L3 (D13 LED indicator) is blinking. Place the 3CH Line sensors on the White line. then Press LINE CAL again once for the White color calibration. After that you will see all the LEDS for linesensors are ON.

5. Now your eGizmo PBOT 2018 Controller line sensors are calibrated. Then Press the RESET button. You can now trace the line and DONE.

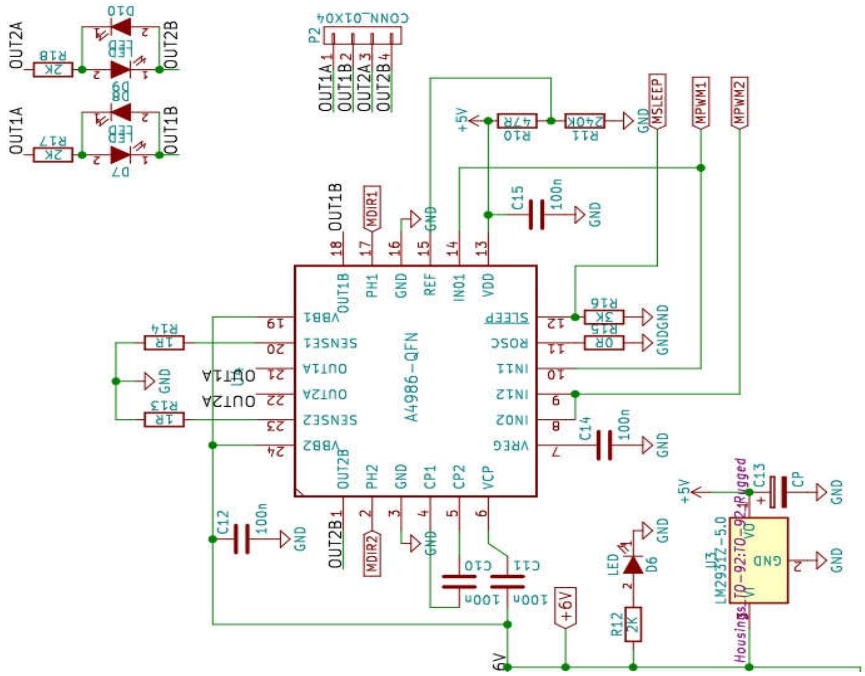
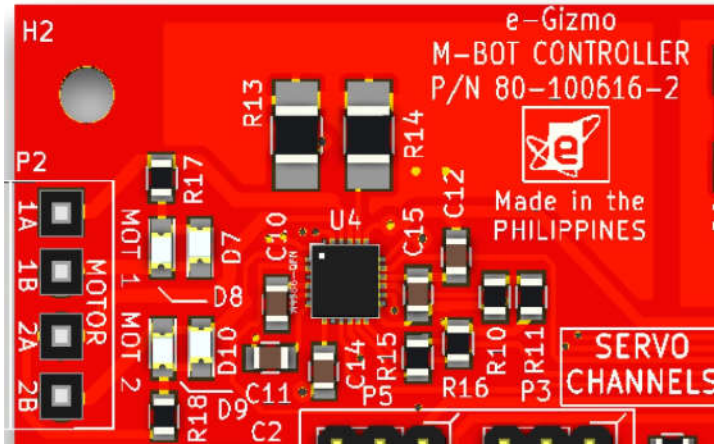


Figure 6: Motor Driver circuit and Motor 1, Motor2 output connections.

The Motor driver circuit is the connection circuit between your controller and the motor. The controller can scheme the operation of the motor through a series of simple logic output combinations through this circuit.

A4986 dual full-bridge PWM Motor driver with overcurrent protection and internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation.

The outputs are protected from shorted load and short to ground events, to protect the driver from thermal damage.



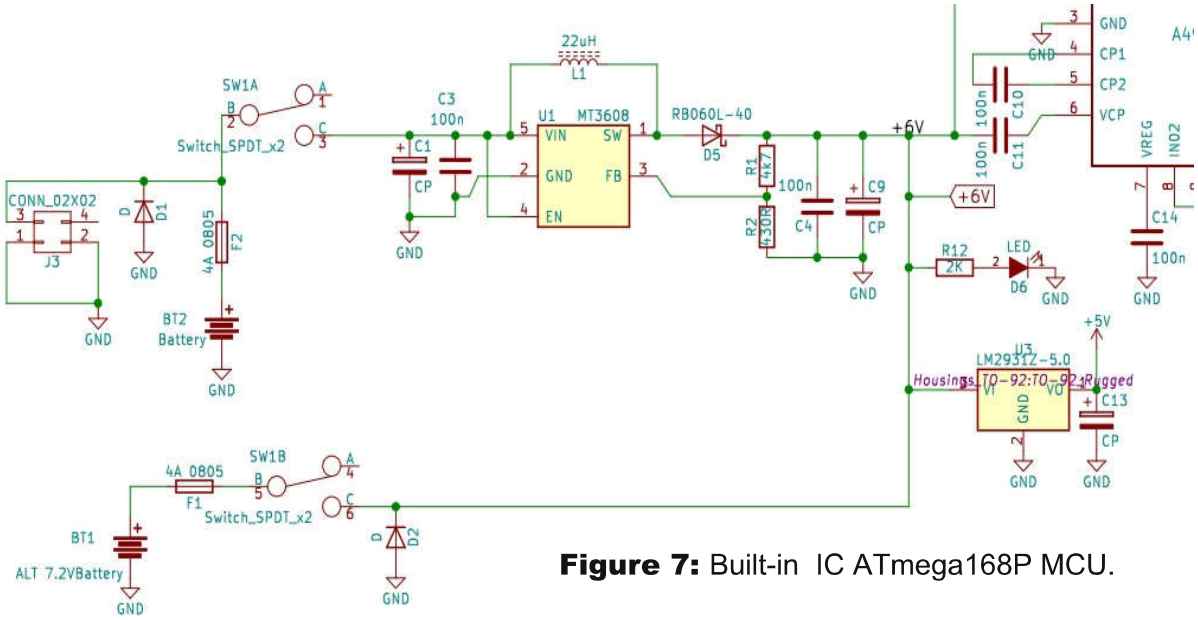
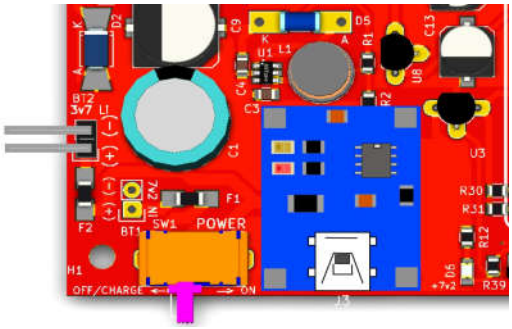


Figure 7: Built-in IC ATmega168P MCU.



This is the charging section for 3.7v Li-on Battery 1580mAh (BT2) and 7v2 Ni-MH battery (BT1) connections (Optional). SW1 Power Switch ON, OFF/Charge.

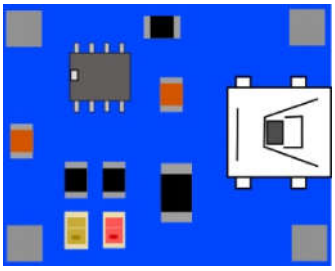
Note: If the power shutdown accoured, kindly unplug the battery BT2 then plug In again. (The battery has its own circuit protection.)
Also apply this if you trying to upload your codes or charge the battery.

Tips for uploading:
Press and Hold the RST (SW3) then Switch On the Power and Click Upload. Release RST when done.

The PBOT2018 controller board has a TP4056 module for Li-on battery charger (J3), just connect USB cable Type Mini B to Type A to the module with battery plug on the BT2 connection. This module has indicator the Red LED lights will on, if it is the battery is on charging mode. If the battery is fully-charged the Green LED indicator will lights on.

The board also has a DC-DC booster converter from 3.7VDC to 7.2V and the maximum current limit for the M-BOT is 1.5A,

Important: *Charge your battery,if not enough power to run the motors.* ***Full Charging*** *hours approximately* ***1 hour and 32 mins (2 hrs max)***



The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter.

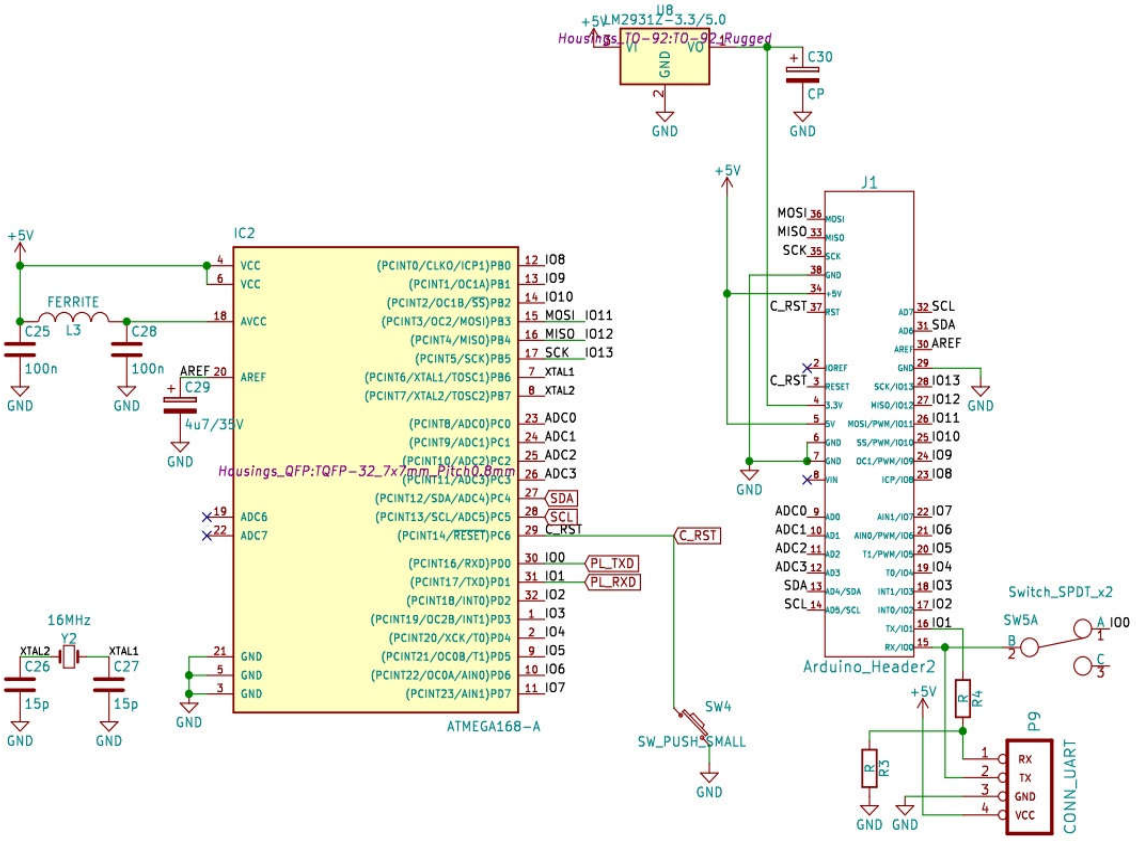
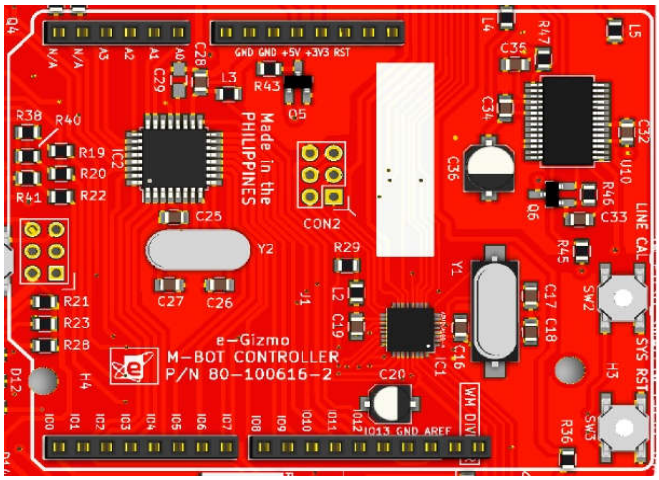
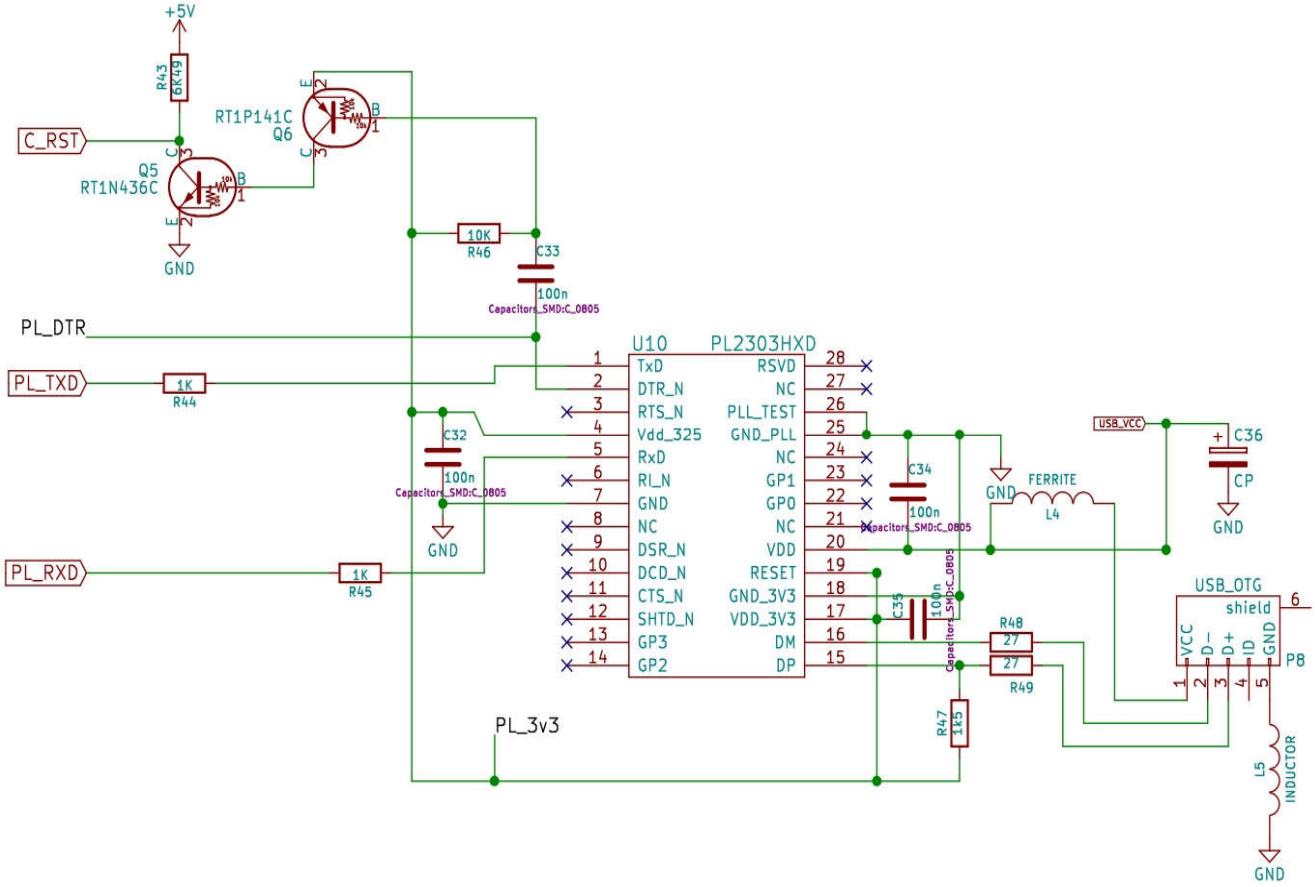


Figure 7: Built-in IC ATmega168P MCU.

Most important on board is the microcontroller, the PBOT2018 controller has a built-in ATmega168P IC with compatible shield and the software is Arduino IDE. It has its own library to use the "eGizmo_PBOT2018 library" must be added to your My documents> Arduino>libraries.

Some examples are Line follower, Collision avoidance, solving maze, Motor Test, and servo controlled.





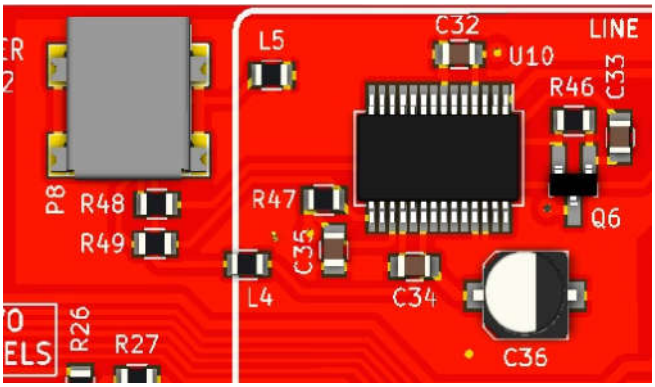


Figure 8: Prolfic Driver.

With Built-in PL2303 Driver U10 to connect easily on your computer via USB mini type B P8, just install the driver manually. You may visit the site and downloads at

http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

This device will let your microcontroller communicates with the PC through a USB port the same way you will with a PC COM port.

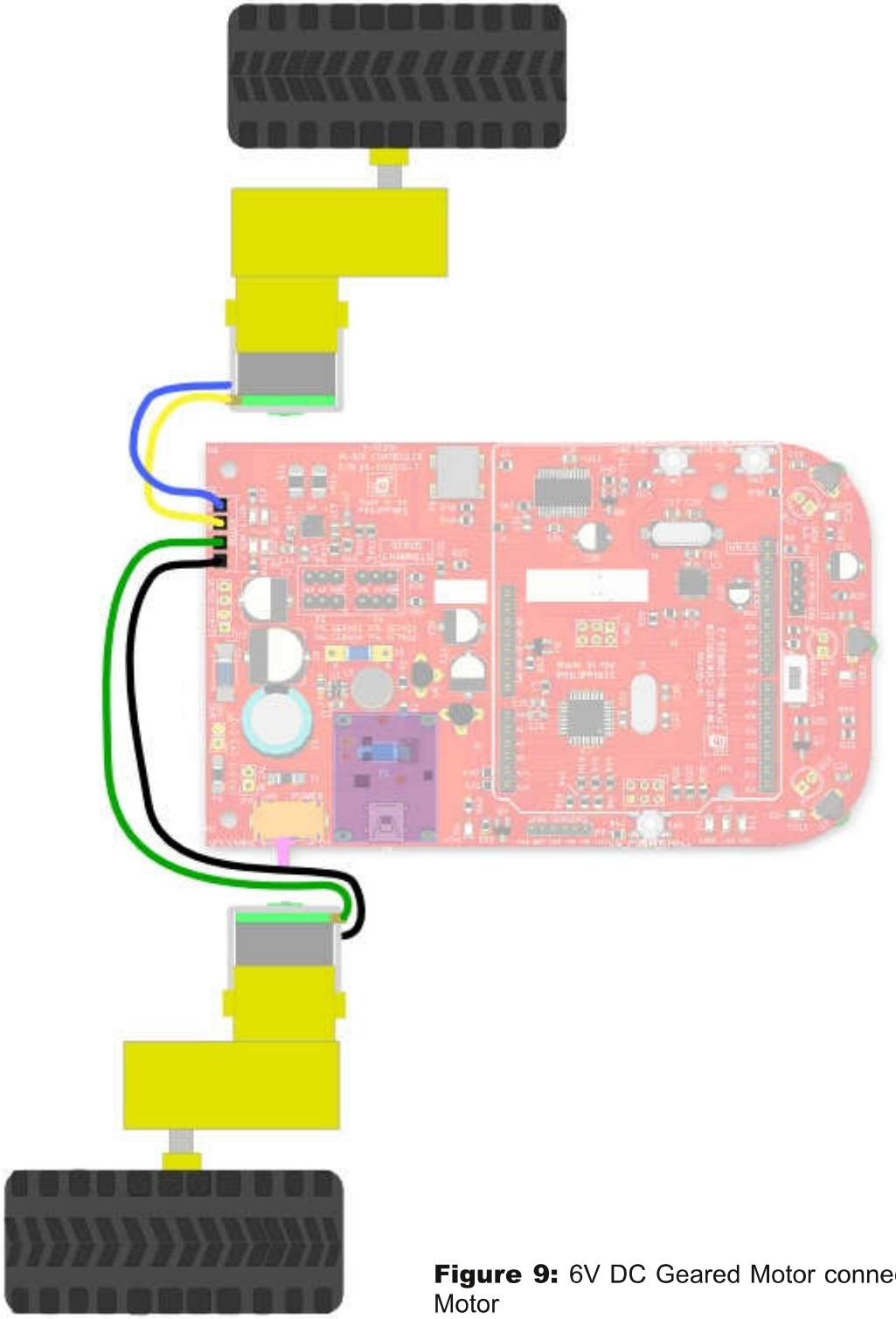


Figure 9: 6V DC Geared Motor connections to Motor
connections.

Adding eGizmo_PBOT2018 Library:
Go to> My Documents>Arduino>libraries.

Tips for uploading:
Press and Hold the RST (SW3) then Switch On the Power and Click Upload. Release RST when done.

```
/***********************************************
  e-Gizmo PBOT2018 Controller -
MOTOR_TEST.ino

  This is for motor sample sketch to controls
  and move forward, reverse, turning right,
  turning left, and stop the motors.

  Set motor speed:
  PBOT.SPEED(whichmotor, speed);
  Note: Forward Speed = 0 Full Stop;
                        255 High, limit 250
  Where:
      whichmotor = MOTOR_A or MOTOR_B,
       MOTOR_BOTH
      speed = 0 to 255

  Set motor direction:
  PBOT.DIRECTION(whichmotor, dir);
  Where:
      whichmotor = MOTOR_A or MOTOR_B,
        MOTOR_BOTH
      dir = MOTOR_FWD, MOTOR_REV


  Codes by
  e-Gizmo Mechatronix Central
  http://www.e-gizmo.net
  June 5,2018
***********************************************/

#include "eGizmo_PBOT2018.h"
#include <Wire.h>


eGizmo_PBOT2018 PBOT;

void setup() {
  Wire.begin();
  PBOT.BEGIN();
  PBOT.ALLON(); //Turn on all motors
  // Need for intial start of motors
  PBOT.SPEED(MOTOR_BOTH, 120);
}

void loop() {

  //Move Forward
  PBOT.DIRECTION(MOTOR_BOTH, MOTOR_FWD);
  PBOT.SPEED(MOTOR_BOTH, 120);
  delay(2000);

  //Reverse
  PBOT.DIRECTION(MOTOR_BOTH, MOTOR_REV);
  PBOT.SPEED(MOTOR_BOTH, 120);
  delay(2000);

  //Turn Right
  PBOT.DIRECTION(MOTOR_A, MOTOR_FWD);
  PBOT.SPEED(MOTOR_A, 120);
  PBOT.SPEED(MOTOR_B, 0);
  delay(2000);

  //Turn Left
  PBOT.DIRECTION(MOTOR_B, MOTOR_FWD);
  PBOT.SPEED(MOTOR_B, 120);
  PBOT.SPEED(MOTOR_A, 0);
  delay(2000);

  //Stop
  PBOT.SPEED(MOTOR_BOTH, 0);
  delay(2000);

  //Extreme Right
  PBOT.DIRECTION(MOTOR_A, MOTOR_FWD);
  PBOT.DIRECTION(MOTOR_B, MOTOR_REV);
  PBOT.SPEED(MOTOR_BOTH, 120);
  delay(2000);

  //Extreme Left
  PBOT.DIRECTION(MOTOR_A,MOTOR_REV);
  PBOT.DIRECTION(MOTOR_B,MOTOR_FWD);
  delay(2000);
}
```

PBOT 2018 LINE FOLLOWER

Line follower are one of the basic mobile robot designs used by hobbyists and educators. This simple application makes use of line sensors and motor logic applying conditional statement applications such as if-statements, while-statements, etc.

This application makes use of infrared sensors. From basic optics, the infrared transmitter (IR LED) outputs IR light like a typical LED and is thus received by an IR receiver (IRDA). Each time the reciever senses the bounced back IR light, a certain amount of voltage is fed as output to the comparator, thus gives a digital output.

Shown in Figure below, this is the simplest line following logic for a line follower mobile robot. The program is very simple, each time a black line is detected, the mobile robot should turn to a certain direction. For example, if the left line sensor is triggered, the mobile robot should turn right and vice versa. However, if the center line sensor is triggered, the mobile robot should not hesitate to go forward and continue line tracking.

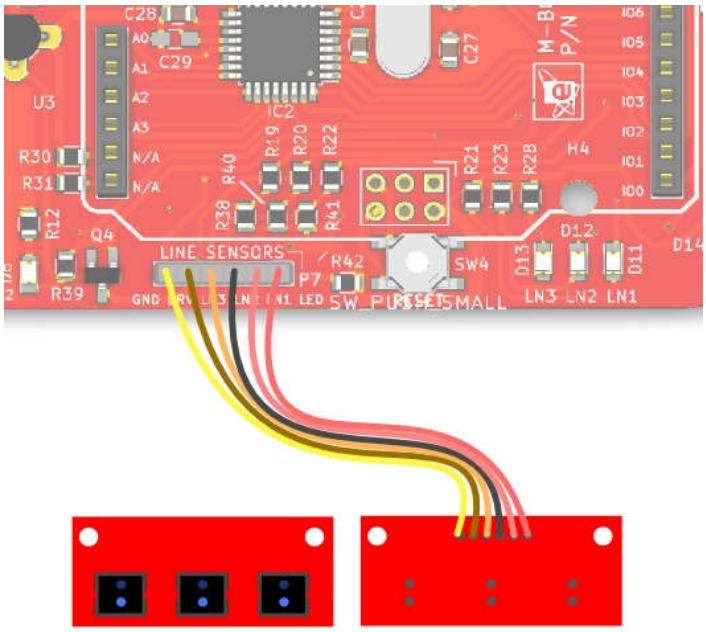
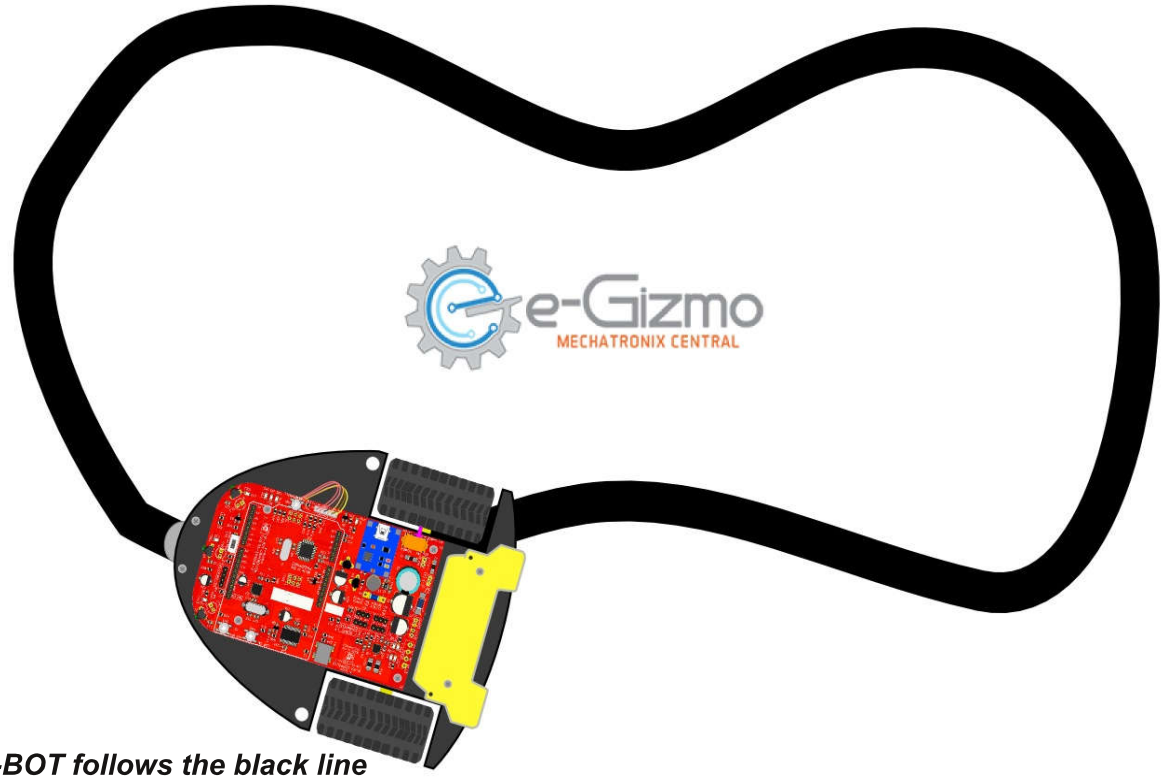


Figure 10: Line sensor connections.

Other applications:
- Doodle robots
- Cleaning robots
- Line follower racing robots



M-BOT follows the black line

/**
e-Gizmo PBOT2018 Controller
- LINEBOT.ino

This is for Linesensor sample sketch to
follows the black line or white line track.

Reading the 3channel Line sensors:
Calibration Instructions:

For the new eGizmo PBOT2018 Controller features first thing to do is to calibrate the line sensors. How to calibrate it? Read the ff.

1. After uploading your code for linesensors. Turn OFF the POWER switch.

2. Place the eGizmo PBOT controller to the **"Black line"** then Press and Hold LINE CAL and SYS RST, while pressing and holding the buttons, Turn ON the POWER Switch. LN2 (D12 LED indicator) is ON.

3. First RELEASE the SYS RST followed by LINE CAL. Make sure the 3CH Line sensors are faces on the **"Black line"** and you will see the LN1 and LN3 (D11 and D13 LED indicators) are Turn ON and LN1 is blinking. Now Press LINE CAL once for the white color calibration.

4. Next, if the L3 (D13 LED indicator) is blinking. Place the 3CH Line sensors on the **White "track".** then Press LINE CAL again once for the black color calibration. After that you will see all the LEDS for linesensors are ON.

5. Now your eGizmo PBOT Controller line sensors are calibrated. Then Press the RESET button. You can now trace the line and DONE.

PBOT.LS1_LEFT();
PBOT.LS2_CENTER();
PBOT.LS3_RIGHT();

Note:
Output reads:
***White line** detected = **0** or **LOW**;*
Black line detected = **1** or **HIGH**;

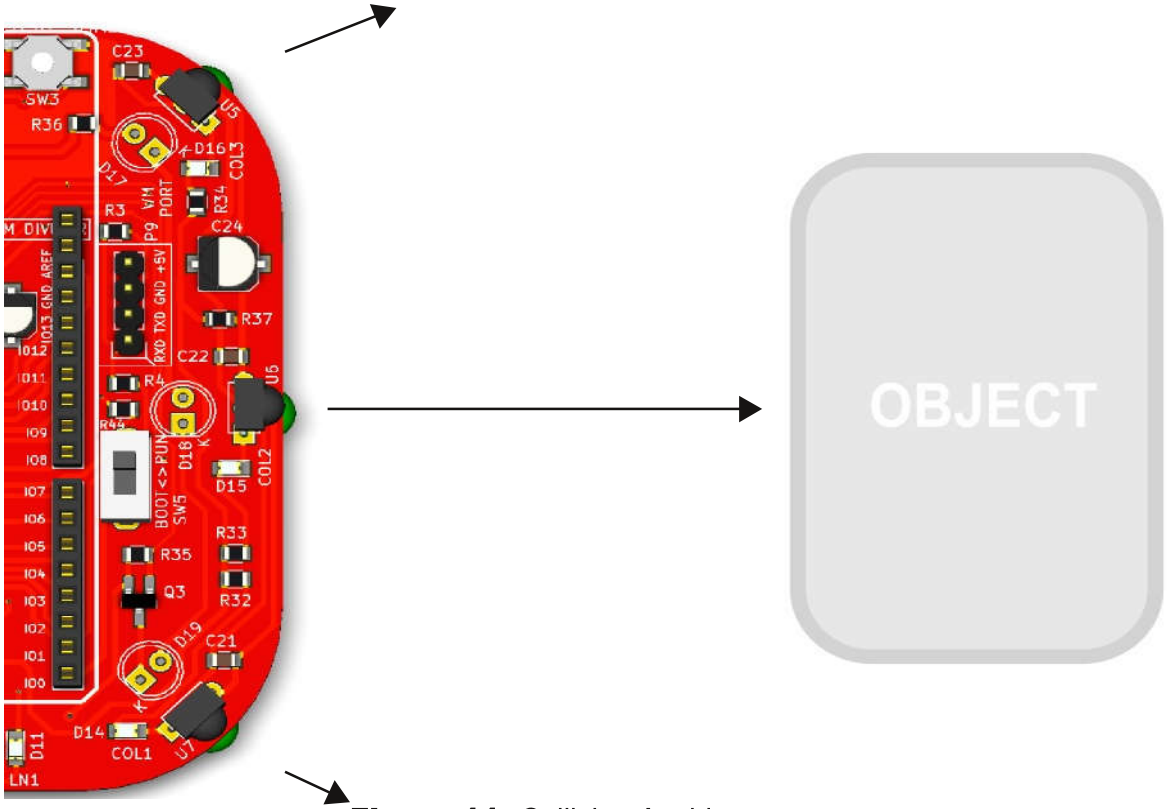
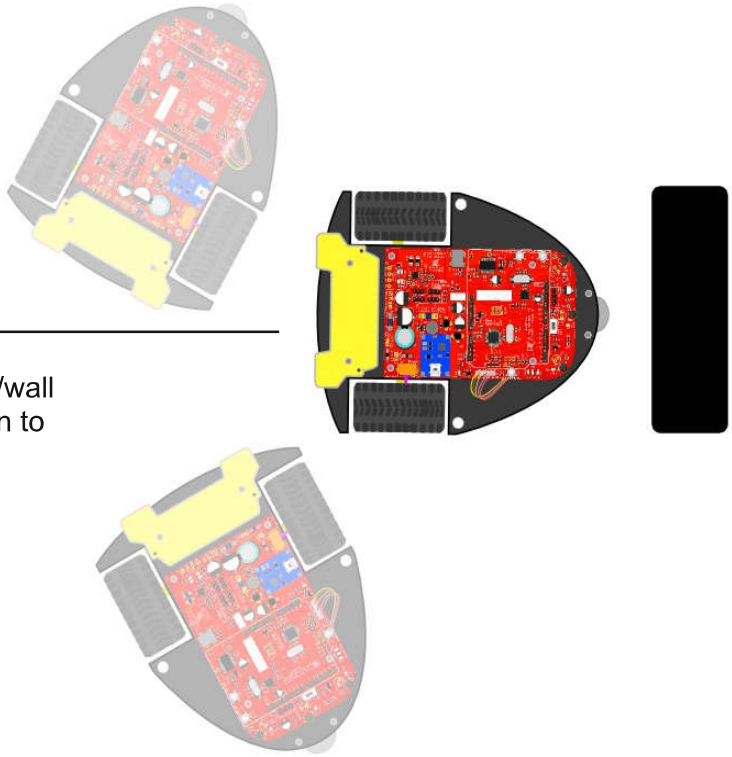


Figure 11: Collision Avoidance.



PBOT2018 will reverse if there's an obtraction object/wall and change direction to continue.

PBOT2018 SUMO

A sumobot is technically a mobile robot programmed to be like a sumo wrestler. Its basic logic is to move away the target out of the ring while avoiding getting out of the ring itself. The trend of sumobots started as collision sensors were incorporated on mobile robots as a kind of sport or game. Some would also organize sumobot competitions wherein hobbyists can show off their DIY robots and their programming skills.

The theory behind the sumobot is the use of infrared transmitters and receivers. Similiar to the line sensor, a certain voltage output is fed to logic gates to identify diigital logic. In other words, when an object is detected, a "HIGH" logic level will fed to the MCU wherein conditions may be applied.

For a sumobot, the collision sensor may trigger the mobile robots maneuvers wherein the robot may turn around and avoid an enemy or ram the enemy directly so that it enemy is thrown away from the ring.

Again, sample codes are provided on the same folder as this technical manual as a demo program for the said application.

Sensors applied:
- Line sensors
- Collision sensors

Other applications:
- Fall avoiding robot
- Robot wrestling

Adding eGizmo_PBOT2018 Library:
Go to> My Documents>Arduino>libraries.

Tips for uploading:
Press and Hold the RST (SW3) then Switch On the Power and Click Upload. Release RST when done.

```
/*********************************************
  e-Gizmo PBOT Controller
- AVOID_COLLISION.ino

  This is for collsion sample sketch to avoid
  crashing on the wall or any objects that blocking
  way of the PBOT.

  Reading the 3channel Collision sensors:

  PBOT.COL1_RIGHT();
  PBOT.COL2_CENTER();
  PBOT.COL3_LEFT();

  Output reads:
      No object detection = 0 or LOW;
      Object detected     = 1 or HIGH;

  Codes by
  e-Gizmo Mechatronix Central
  http://www.e-gizmo.net
  June 5,2018
*********************************************/

#include "eGizmo_PBOT2018.h"
#include <Wire.h>


eGizmo_PBOT2018 PBOT;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  PBOT.BEGIN();
  PBOT.ALLON();
  PBOT.SPEED(MOTOR_BOTH, 120);
}

void loop() {
  if(PBOT.COL2_CENTER()){ //center object detection
   change_direction();
  }
  delay(200);
}

//Sample Function for changing directions
void  change_direction(void) {
  boolean cw = HIGH;
  PBOT.DIRECTION(MOTOR_BOTH, MOTOR_REV);

  delay(500);
  if (cw) {
    cw = false;
    PBOT.DIRECTION(MOTOR_A, MOTOR_FWD);

  } else {
    cw = true;
    PBOT.DIRECTION(MOTOR_B, MOTOR_FWD);
  }
  delay(500);
  PBOT.DIRECTION(MOTOR_BOTH, MOTOR_FWD);
}
```

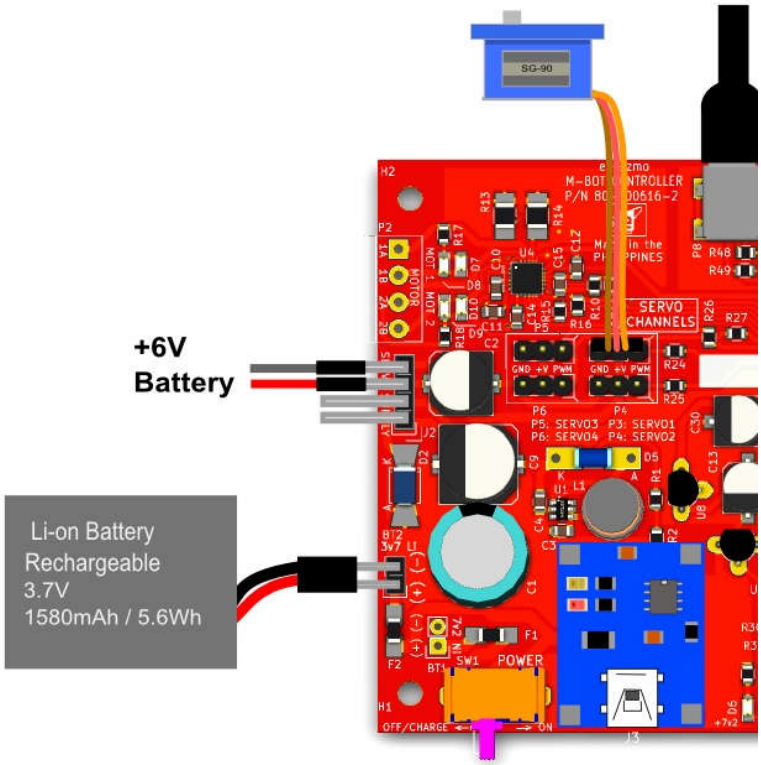


Figure 12: Sample connection for Servo Supply you need another supply +6V for +V of servo channels.

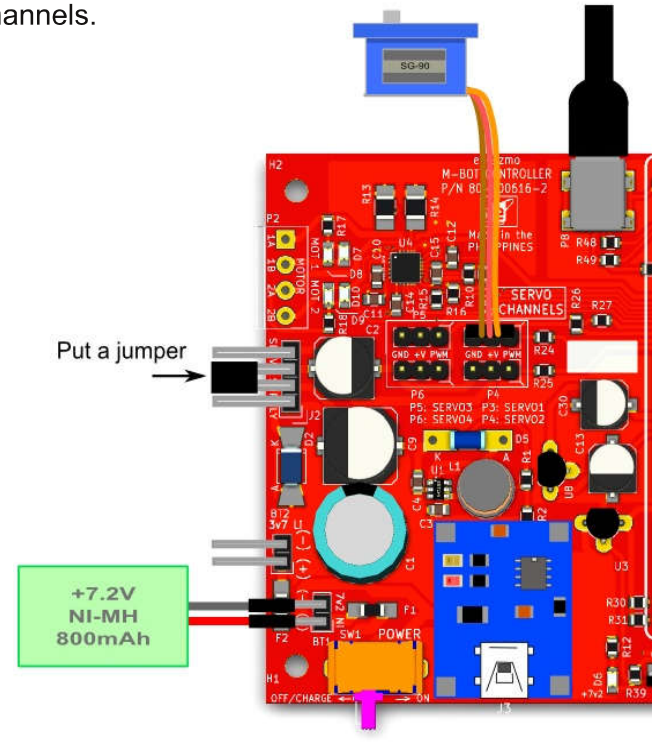


Figure 13: Sample connection with 7.2v Ni-MH battery. Put a jumper for servo supply

Adding eGizmo_PBOT2018 Library:
Go to> My Documents>Arduino>libraries.

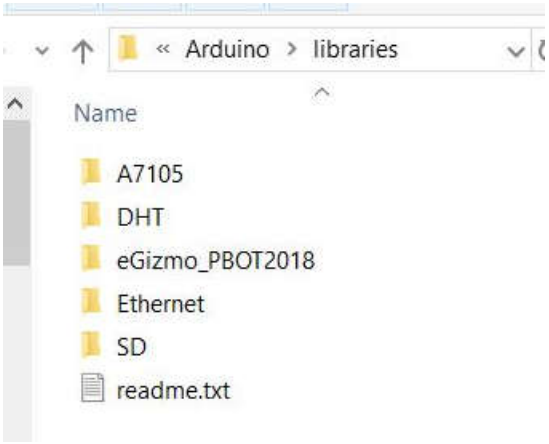
Tips for uploading:
Press and Hold the RST (SW3) then Switch On the Power and Click Upload. Release RST when done.

```
/***********************************************
        e-Gizmo PBOT2018 Controller - SERVO_SWEEP.ino

  This is for SERVO sample sketch
  to sweeps the shaft back and forth
  across 180 degrees.

   SERVO Control
   ex. PBOT.SERVO(whichSERVO,pulsewidth);
   where:
    whichservo = 1 to 4 ,ignore other values
    pulsewidth = 0 to 180 (degrees)
                 - value less than 500 stops the SERVO PWM generator
                 - the pulsewidth converted to degrees from 0 to 180 (default)

  Codes by
  e-Gizmo Mechatronix Central
  http://www.e-gizmo.net
  June 5,2018
***********************************************/

#include "eGizmo_PBOT2018.h"
#include <Wire.h>

eGizmo_PBOT2018 PBOT; // Create PBOT object to control a SERVO
int POS = 0; //servo start position

void setup() {
  Wire.begin();
  PBOT.BEGIN();
}

void loop() {

   for(POS = 10; POS < 170; POS += 1)  // goes from 10 degrees to 170 degrees
   {                                   // in steps of 1 degree
     PBOT.SERVO(1, POS);               // tell servo to go to position in variable 'pos'
     PBOT.SERVO(2, POS);
     PBOT.SERVO(3, POS);
     PBOT.SERVO(4, POS);
     delay(15);
     // waits 15ms for the servo to reach the position

   }
   for(POS = 170; POS>=10; POS-=1)     // goes from 170 degrees to 10 degrees
   {
     PBOT.SERVO(1, POS);               // tell servo to go to position in variable 'pos'
     PBOT.SERVO(2, POS);
     PBOT.SERVO(3, POS);
     PBOT.SERVO(4, POS);
      delay(15);                       // waits 15ms for the servo to reach the position
   }

}
```

First you need an Arduino IDE:
Download the latest Arduino 1.8.5,
you can use any version from 1.0.6 and up with gizDuino Patch file (boards list)
- www.e-gizmo.net
- www.github.com/e-Gizmo ; Click 'Repositories'
Search: "Arduino IDE"

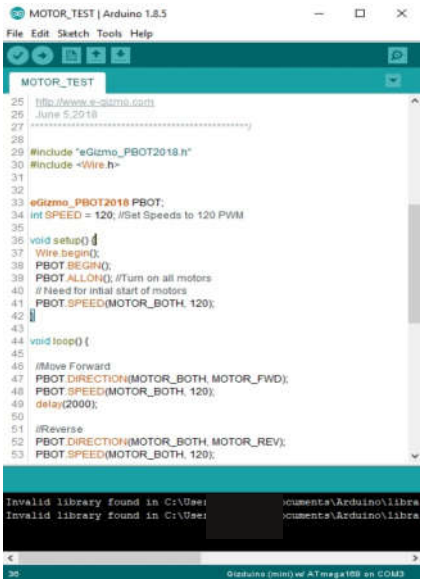
Install Prolific driver for PBOT2018:
Download the Prolific Driver **goo.gl/FqCb15**

Add the eGizmo_PBOT2018 library: Go to My Documents> Arduino> libraries.

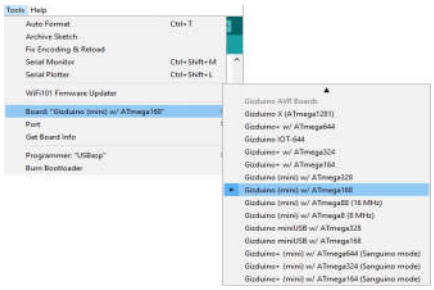


Examples:

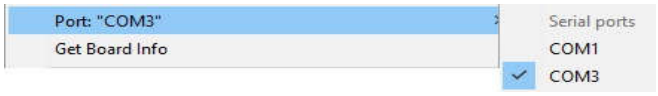
Open Arduino IDE: Then Open sample codes go to Files> Examples> eGizmo_PBOT2018> MOTOR_TEST.ino.



Select the Board: Tools> Board: "GizDuino (mini) w/ ATMEGA168"



Select the Port: Tools> Port: COM # (not COM1)



Tips for uploading:
Press and Hold the RST (SW3) then Switch On the Power and Click Upload. Release RST when done.

Click 'Upload'