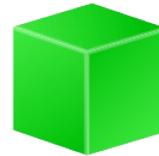


# EBOT with EGRA

## MANUAL GUIDE

### VARIANTS:

- EBOT STANDARD
  - > Line follower
  - > Maze Solver
  - > Sumo fight
- EBOT MAZE
  - > with Ultrasonic sensor
- EBOT SUMO
  - > with Ultrasonic sensor and bumper accessory
- EBOT SOCCER
  - > with Bluetooth Shield
- EBOT with GRIPPER
  - > with Gripper arm
  - > with PS2 Controller
- **EBOT with EGRA**
  - > with EGRA (Robotic Arm)
  - > with PS2 Controller



### Included:

- PS2 Controller
- USB Cable Type A to mini B
- 9V 1A adaptor





# EBOT w/ EGRA

## DESCRIPTIONS

### VARIANTS:

- **EBOT w/EGRA\*** - 4x4 entry level mobile robot fused with mini robotic arm our very own e-Gra. Roving and Pick n' Place functions made more easier. With PS2 Controller wireless device to control it.



# EBOT w/ EGRA

## SPECIFICATIONS

### General Specifications:

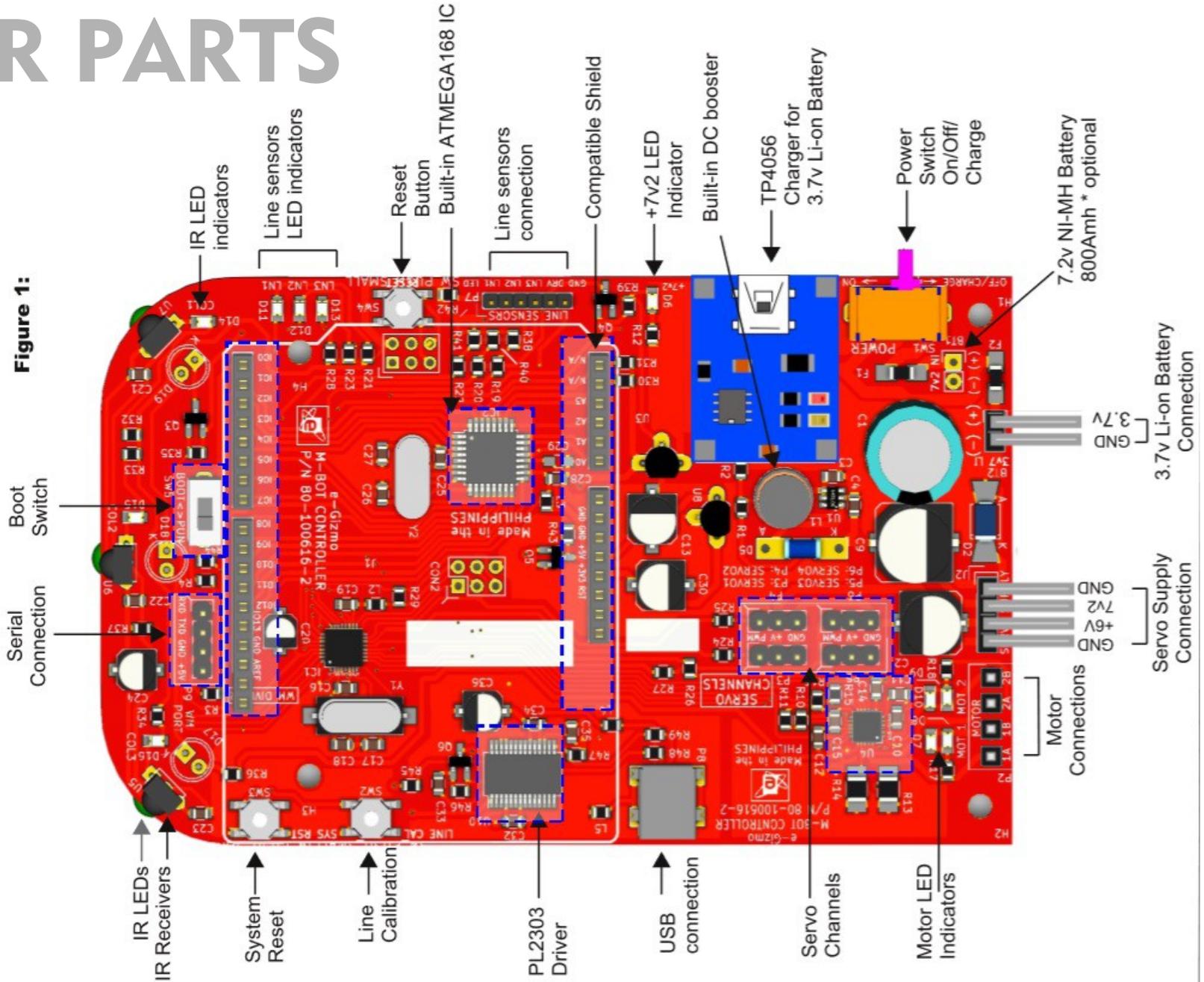
- **Battery:** 7.2V Ni-MH 800mAh rechargeable
- **On-board Peripherals:**
  - IC Atmega168 w/ 16KB Flash memory
- Dimensions: 180mm x 138mm x 120 mm with spoiler
- Weight: Approx. 0.9 KG

### Features:

- Pick N' Place of an object.
- with PS2 controller wireless device to control it,

# PBOT 2018 BOARD

## MAJOR PARTS

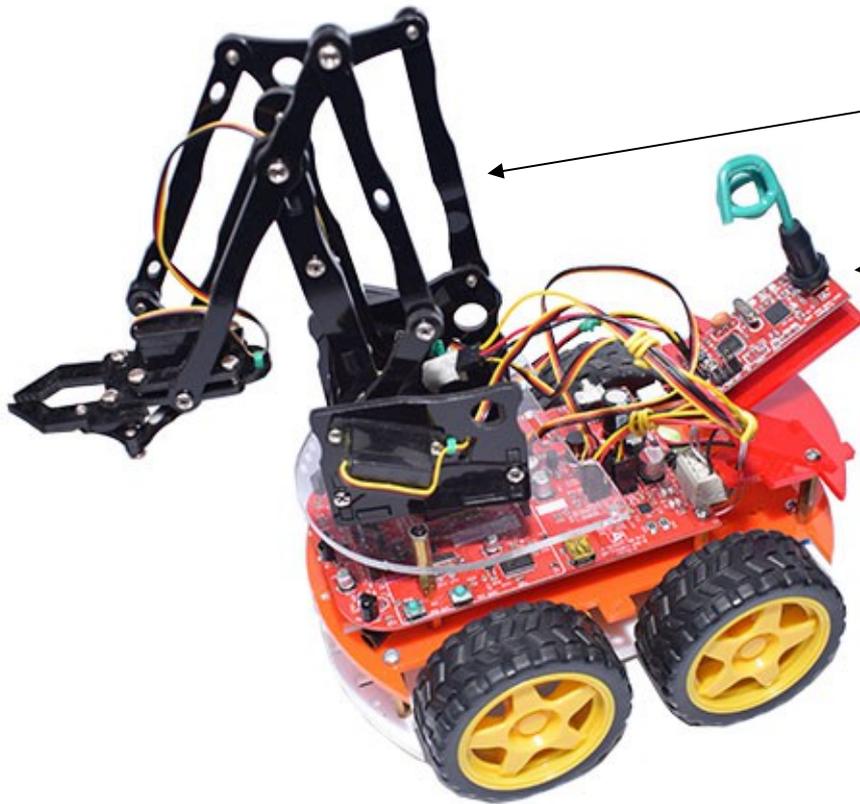




# With PS2 Controlled

# EBOT w/ EGRA

Connections



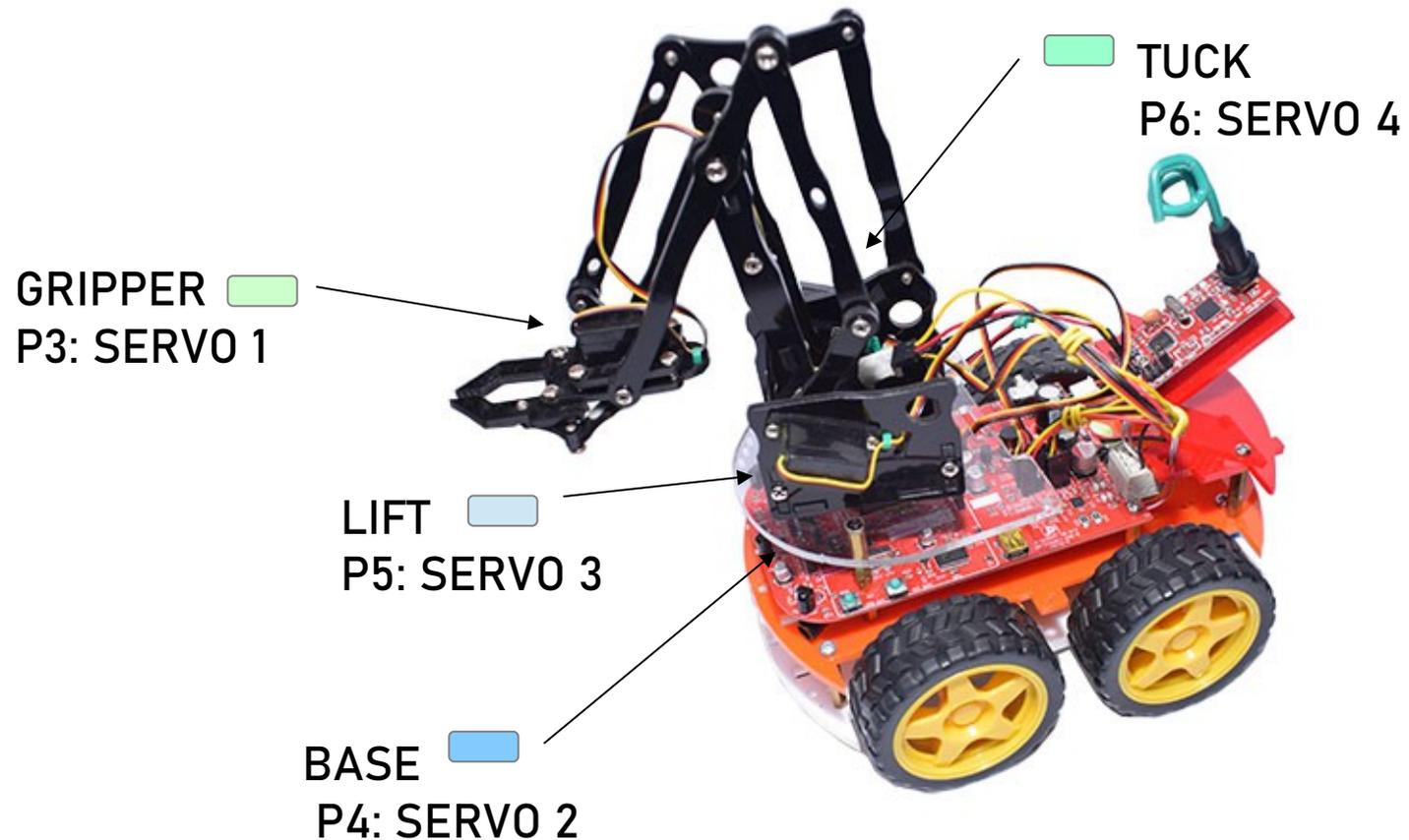
 E-GRA (Robotic Arm)  
- for Pick N' Place of an objects.

 UHF Standard Receiver Only  
- to control the arm and motors.

PS2 Controller



# Pin assignments



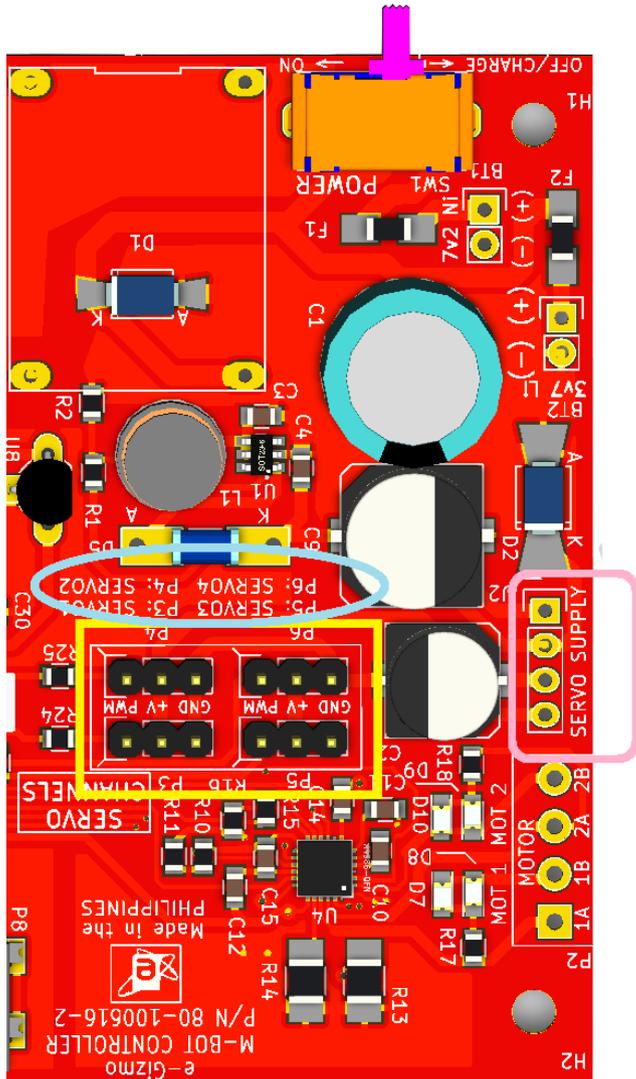


# SERVO CONNECTION

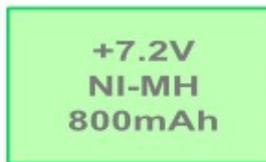
 Servo labels #1-#4

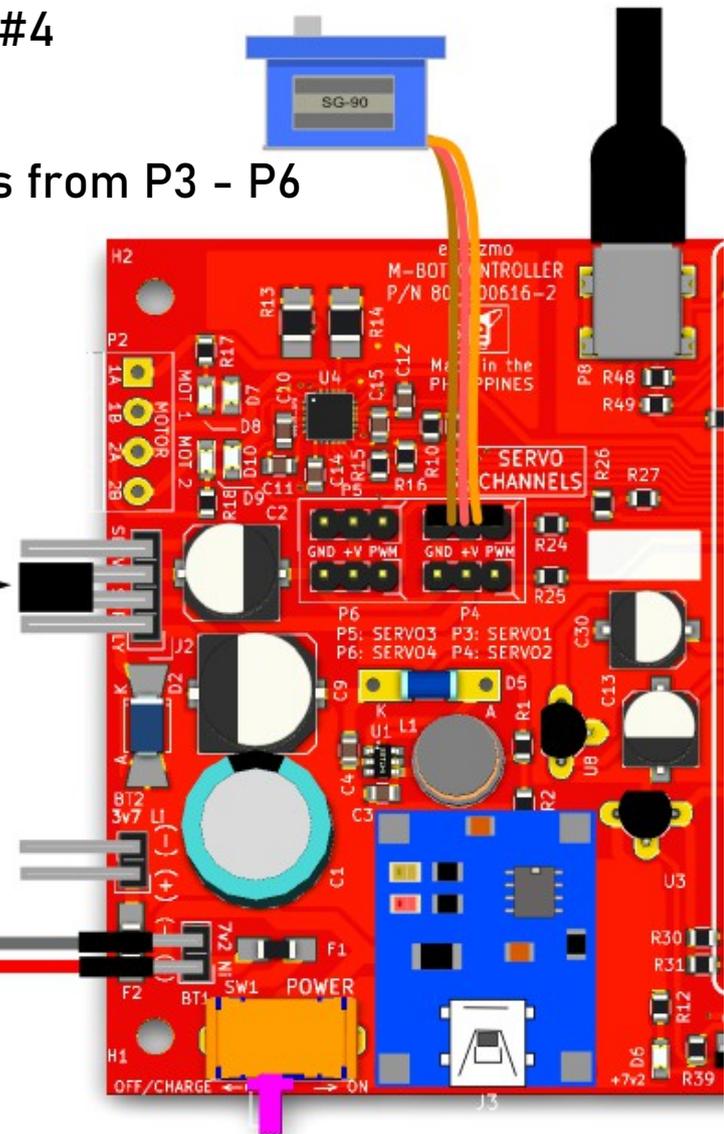
 Servo connectors from P3 - P6

 Servo Supply J2



Put a jumper

 +7.2V  
NI-MH  
800mAh



# Softwares and library

## Arduino IDE

[www.e-gizmo.net/oc/kits](http://www.e-gizmo.net/oc/kits) documents/ARDUINO IDE SOFTWARES

Download Arduino 1.8.5 egizmo themes new (Windows)

Choose your Arduino IDE for your OS.

## Drivers **Install this first!**



PL2303\_Prolific\_DriverInstaller\_v1.10.0

[www.e-gizmo.net/oc/kits](http://www.e-gizmo.net/oc/kits) documents/ARDUINO IDE SOFTWARES

Download Prolific Driver v10.0 (Windows)

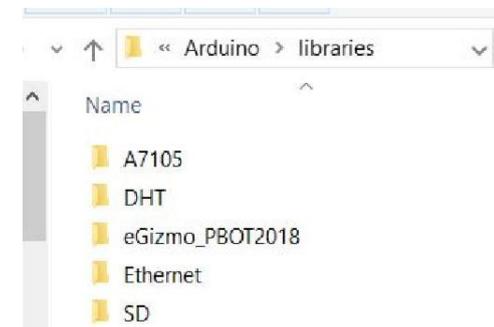
(For Mac OS users) Download md\_PL2303\_MacOSX

## Library **Add to My Documents > Arduino > libraries**

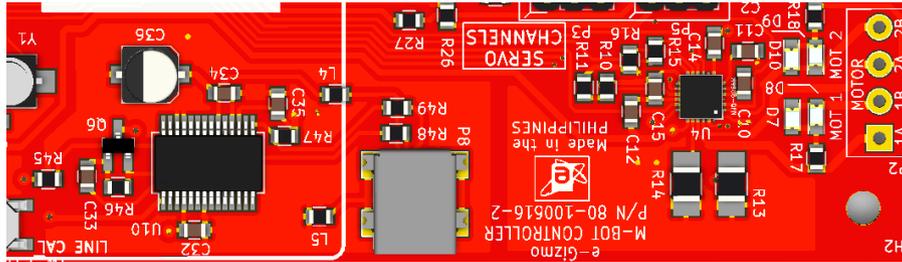
[www.e-gizmo.net/oc/kits](http://www.e-gizmo.net/oc/kits) documents/PBOT2018

Download eGizmo\_PBOT2018.zip and eGizmo\_PS2Controller

(Unzip this before you move to libraries)



# Connect the EBOT to PC



USB Connector

USB Cable



Open Arduino IDE.

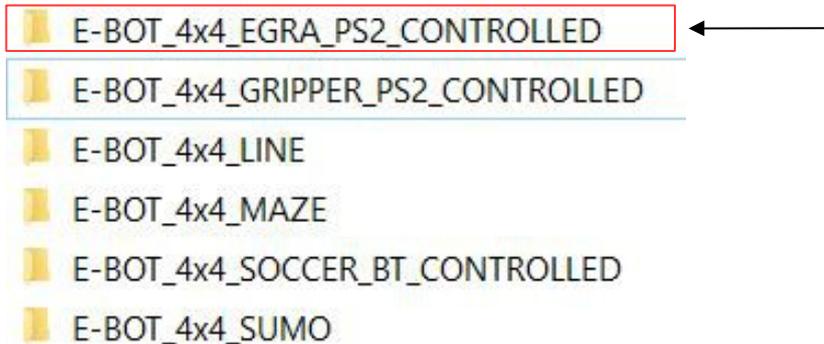
# Uploading eBot e-Gra.ino

On the Arduino IDE.

1. For EBOT w/ EGRA codes you need PS2 controller

Open the E-BOT\_4x4\_EGRA\_PS2\_CONTROLLED.ino

Go to File>Open> E-BOT\_4x4\_EGRA\_PS2\_...ino



2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168



## Sketch

3. Port select

Go to Tools>Port>COM#

Select the correct port

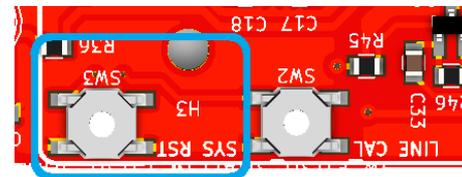
Go to Device Manager

if you're not sure.

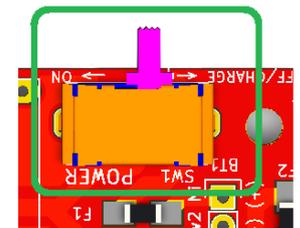


Tips for uploading:

Press and Hold the SYS RST (SW3) switch ON the power and Click Upload. Release RST when done.



SYS RST



POWER SW

# PS2 Controller button functions

Universal PS2 Controller.

L1 and L2  
= Gripper Pick or release

R1 and R2  
= Lift up and Down

Buttons  
Up = Forward  
LEFT = Turn Left  
Right = Turn Right  
Down = Reverse

Shape button  
Triangle and Cross  
= Tuck up and down  
Square and Circle  
= Base turn left  
And turn right

Left Joystick  
= Motor Controls



# MOTOR CONTROLS SYNTAX

## SYNTAX

Set motor direction:



PBOT.DIRECTION(whichmotor,dir);

where:

whichmotor = MOTOR\_A or MOTOR\_B, MOTOR\_BOTH  
dir = MOTOR\_FWD,MOTOR\_REV

Set motor speed:

PBOT.SPEED(whichmotor,speed);

where:

whichmotor = MOTOR\_A or MOTOR\_B, MOTOR\_BOTH  
speed = 0 to 255, 0 = Full Stop, 80 = Neutral,255 = High, limit 250

EXAMPLES;

```
PBOT.DIRECTION(MOTOR_BOTH, MOTOR_FWD); // Move Forward  
PBOT.SPEED(MOTOR_BOTH, 80);
```

# SERVO SYNTAX FOR ARM

## SYNTAX

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)

## EXAMPLES;

```
PBOT.SERVO(1, 15);
```

```
delay(1000);
```

```
PBOT.SERVo(1, 90);
```

```
delay(1000);
```

# Turn your Ebot Sumo for More Functions like...

## EBOT STANDARD

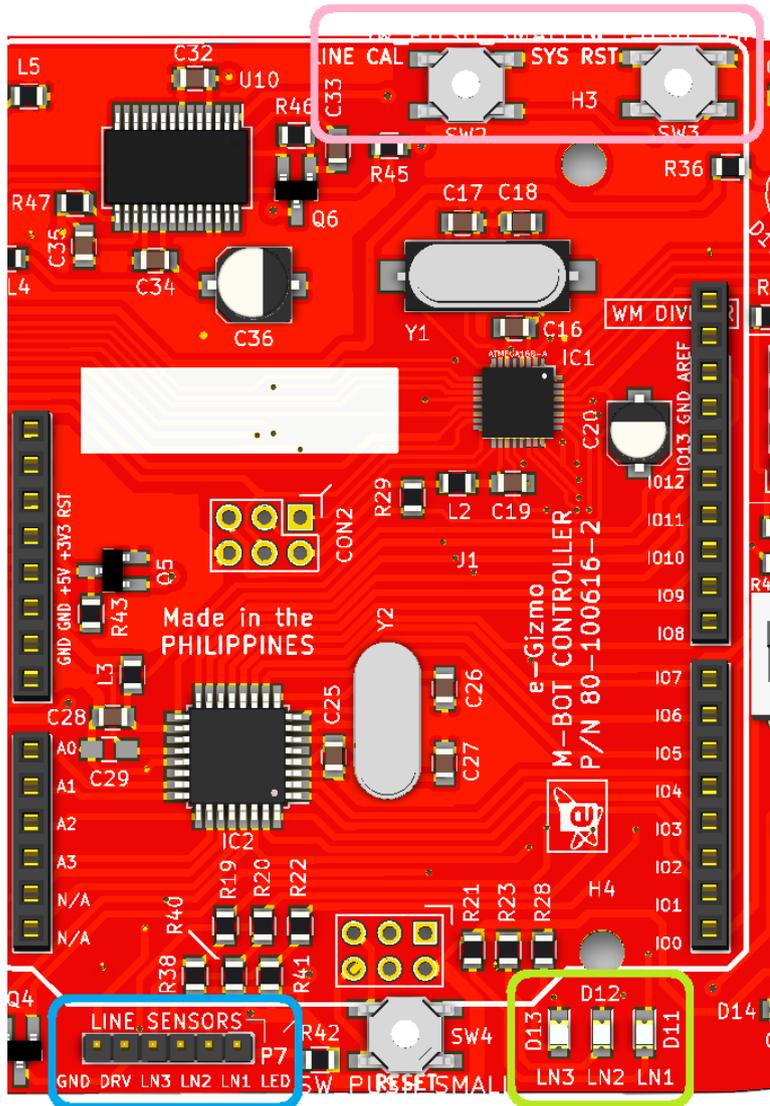


Go to Next Page...



# LINE SENSOR

## LINE FOLLOWER CONNECTIONS



**LINE CAL** – for making line sensor calibrated as easy.  
(to calibrate read the page 5)

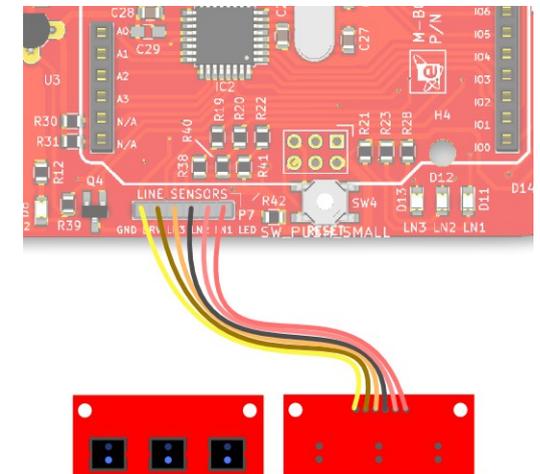
**LINE SENSORS CONNECTION (P7):**  
GND,DRV,LN3,LN2,LN1

**LINE SENSOR LED INDICATORS:**

- LN3(D13), LN2(D12),LN1(D11)
- if LED is ON, black color detected
- if LED is OFF, white color detected

**LINE SENSOR ARRAY:**

**3 channel Line sensors**



# Uploading Linebot.ino

## For line follower

On the Arduino IDE.

### 1. Line Follower codes

Go to File>Examples>eGizmo\_PBOT2018>LINEBOT



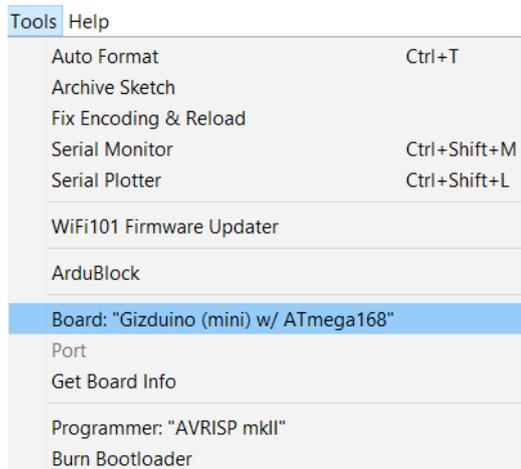
### 3. Port select

Go to Tools>Port>COM#

- Select the correct port
- Go to Device Manager if you're not sure.

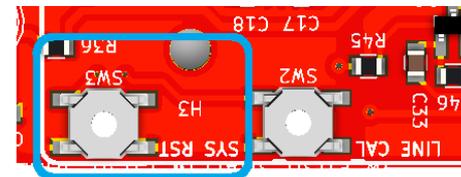
### 2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168

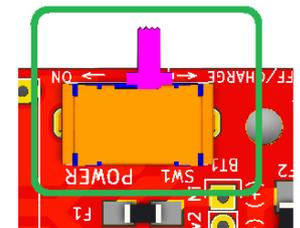


Tips for uploading:

- Press and Hold the SYS RST (SW3) switch ON the power and Click Upload. Release RST when done.



SYS RST

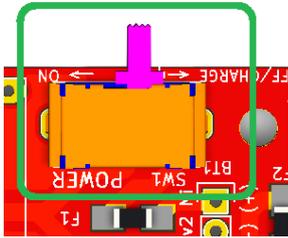


POWER SW

# LINE SENSOR

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

Turn Off



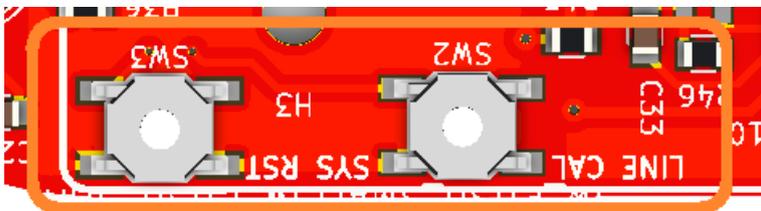
2. Place the eGizmo PBOT controller to the "black line" then

A.



Press and Hold LINE CAL and SYS RST, while pressing and holding the buttons,

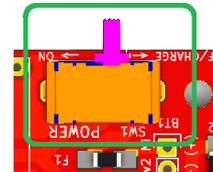
B.



## CALIBRATION 1,2

Turn ON the POWER Switch.

C.



Turn On

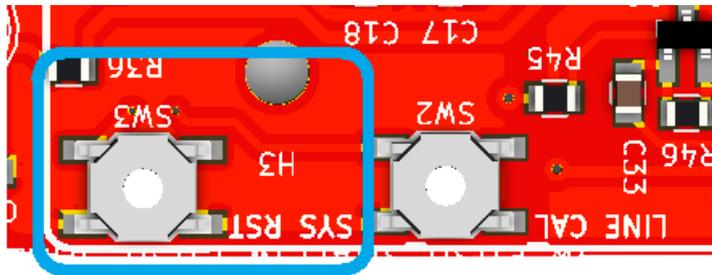
LN2 (D12 LED indicator) is ON.



# LINE SENSOR

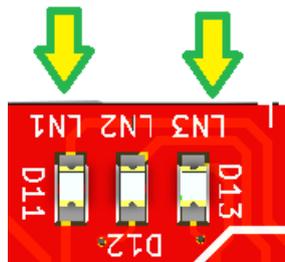
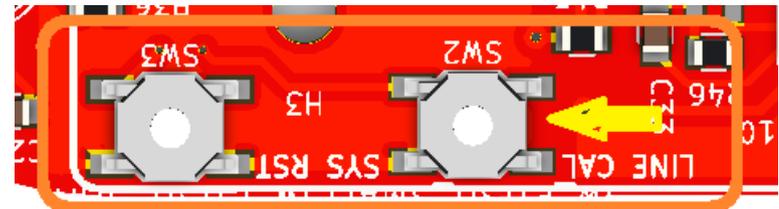
## CALIBRATION 3

3. First RELEASE the SYS RST followed by LINE CAL.  
Make sure the 3CH Line sensors are faces on the "black line"



Now Press LINE CAL once for the black color calibration.

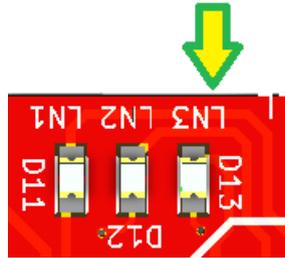
and you will see the LN1 and LN3 (D11 and D13 LED indicators) are Turn ON and LN1 is blinking.



# LINE SENSOR

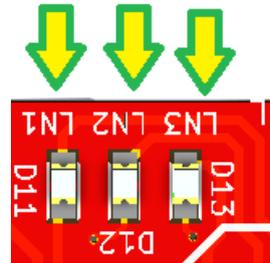
## CALIBRATION 4,5

4. Next, if the L3 (D13 LED indicator) is blinking.

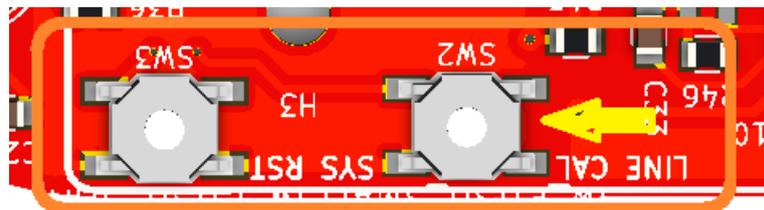


After that you will see all the LEDs for linesensors are ON.

Place the 3CH Line sensors on the "White track".



Then Press LINE CAL again once for the white color calibration.



5. Now your eGizmo PBOT Controller Linesensors are calibrated. Then Press the RESET button or Switch off and turn ON again. You can now trace the line and DONE.

# LINE FOLLOWER SYNTAX

## SYNTAX

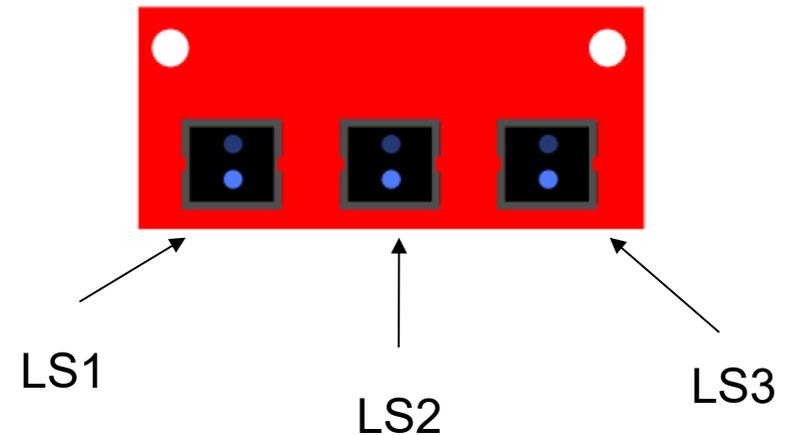
- `PBOT.LS1_LEFT();`
- `PBOT.LS2_CENTER();`
- `PBOT.LS3_RIGHT();`

## OUTPUT

- *WHITE LINE DETECTED = 0 or LOW*
- **BLACK LINE DETECTED = 1 or HIGH**

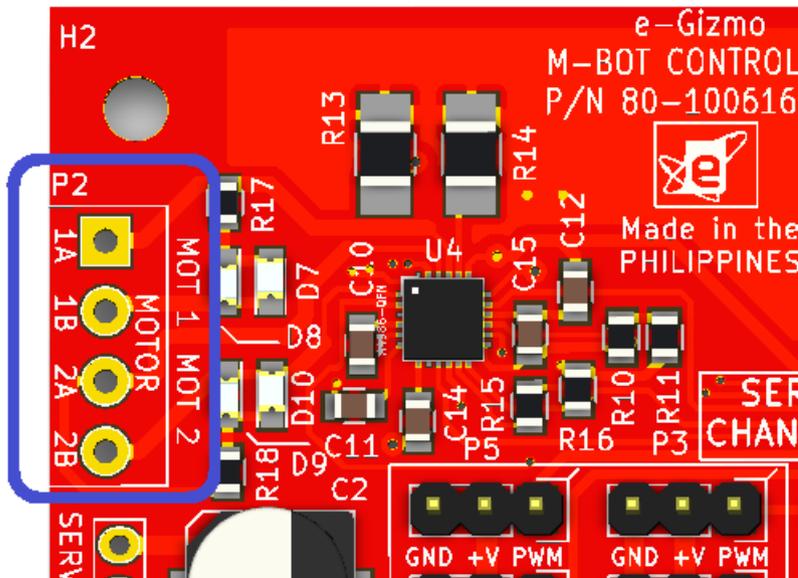
## EXAMPLES;

```
If (PBOT.LS2_CENTER() == HIGH) // if line sensor 2 center is high
{
  Serial.println("Black line detected!");
}
```



# MOTORS

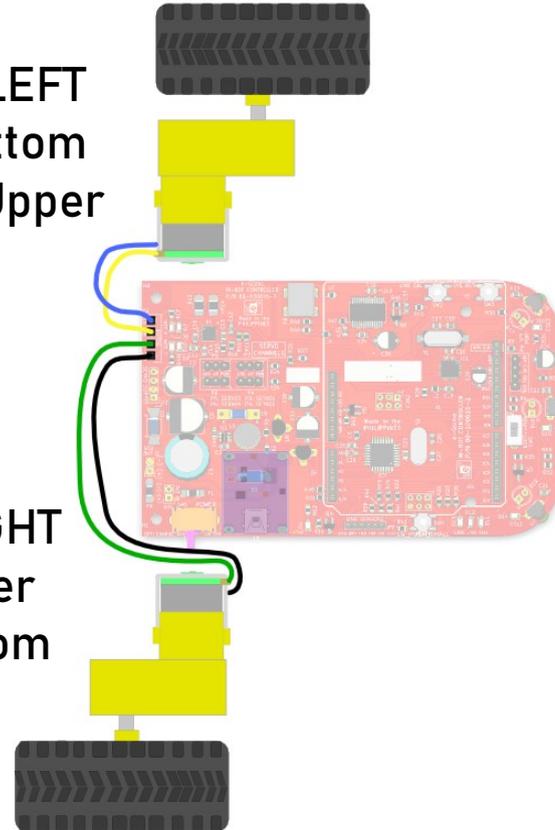
## CONNECTION SECTION



 MOTOR WIRE CONNECTIONS (P2):  
MOT 1 – 1A,1B ; MOT2 – 2A,2B

MOT 1 = LEFT  
Blue-Bottom  
Yellow-Upper

MOT 2 = RIGHT  
Green-Upper  
Black-Bottom



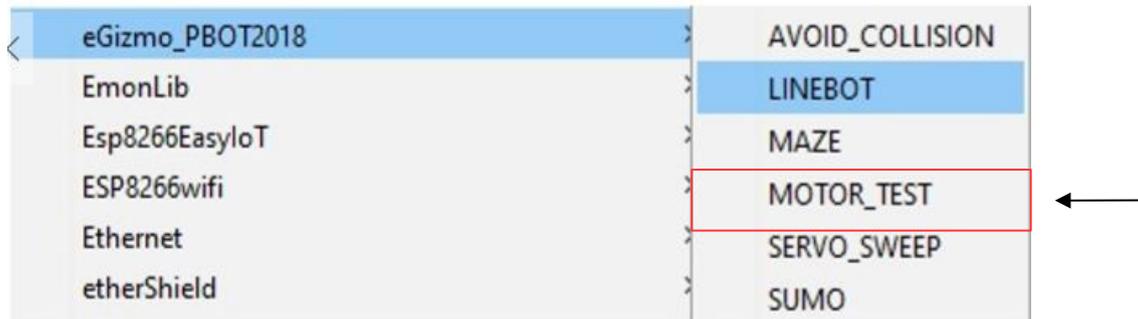
\*Please don not depends on the color of the wires.  
Follow the location of it.

# Uploading Motor Test.ino

On the Arduino IDE.

## 1. Motor controls codes

Go to File>Examples>eGizmo\_PBOT2018>MOTOR\_TEST



## Direction/Speed

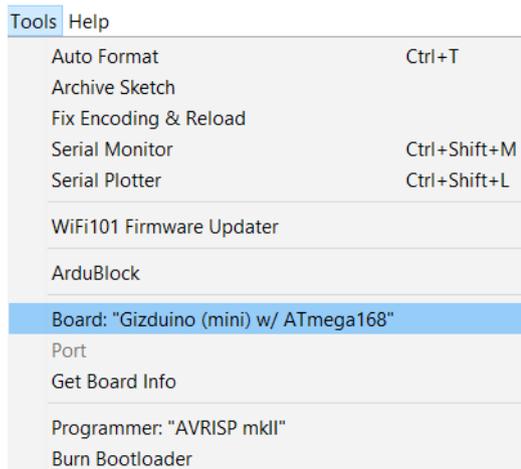
## 3. Port select

Go to Tools>Port>COM#

- Select the correct port
- Go to Device Manager if you're not sure.

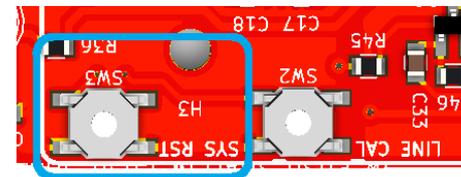
## 2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168

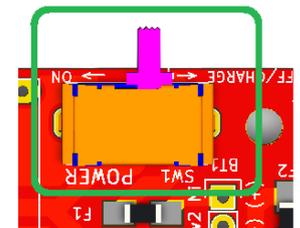


Tips for uploading:

- Press and Hold the SYS RST (SW3) switch ON the power and Click Upload. Release RST when done.



SYS RST



POWER SW

# MOTOR CONTROLS SYNTAX

## SYNTAX

Set motor direction:

- `PBOT.DIRECTION(whichmotor,dir);`

where:

whichmotor = MOTOR\_A or MOTOR\_B, MOTOR\_BOTH  
dir = MOTOR\_FWD,MOTOR\_REV

Set motor speed:

- `PBOT.SPEED(whichmotor,speed);`

where:

whichmotor = MOTOR\_A or MOTOR\_B, MOTOR\_BOTH

- speed = 0 to 255, 0 = Full Stop, 80 = Neutral,255 = High, limit 250

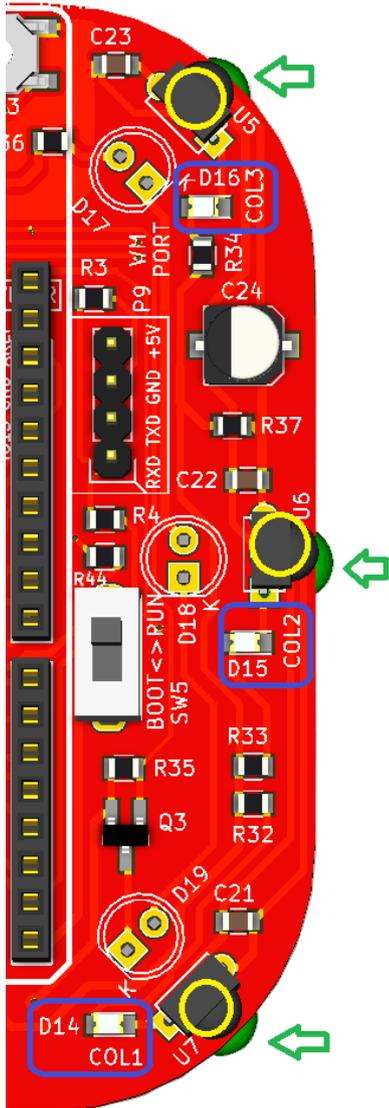
EXAMPLES;

```
PBOT.DIRECTION(MOTOR_BOTH, MOTOR_FWD); // Move Forward  
PBOT.SPEED(MOTOR_BOTH, 80);
```



# MAZE/SUMO

## IR SENSORS CONNECTIONS



IR RECEIVERS 38Khz  
- U5, U6, U7



IR LED TRASMITTER  
- D17, D18, D19  
- Distance Range: 2 - 4 inches



IR LED INDICATORS  
- COL1 (D14), COL2 (D15), COL3 (D16)  
- if LED is ON, Object detected  
- if LED is OFF, No object detected

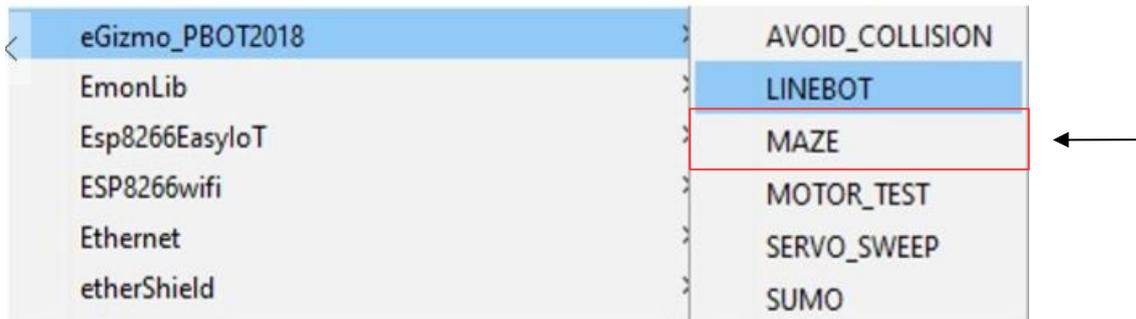
# Uploading Maze.ino

## Maze Solver

On the Arduino IDE.

### 1. Maze Solver codes

Go to File>Examples>eGizmo\_PBOT2018>MAZE



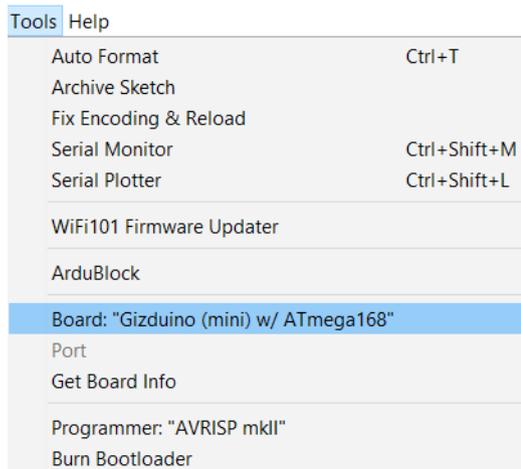
### 3. Port select

Go to Tools>Port>COM#

- Select the correct port
- Go to Device Manager if you're not sure.

### 2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168



Tips for uploading:

- Press and Hold the RST (SW3) switch ON the power and Click Upload. Release RST when done.

# Uploading Sumo.ino

## Sumo Fight

On the Arduino IDE.

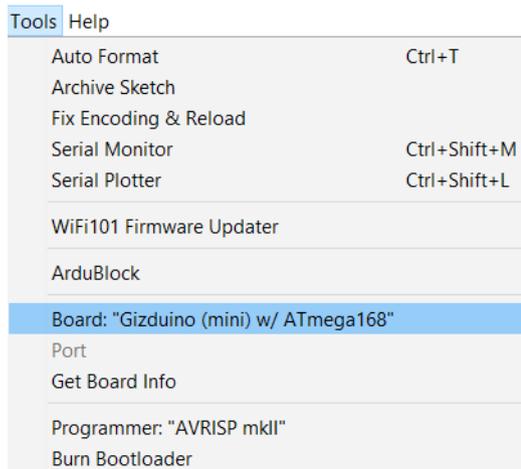
### 1. Sumo fighting codes

Go to File>Examples>eGizmo\_PBOT2018>SUMO



### 2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168



### 3. Port select

Go to Tools>Port>COM#

- Select the correct port
- Go to Device Manager if you're not sure.

Tips for uploading:

- Press and Hold the RST (SW3) switch ON the power and Click Upload. Release RST when done.

# IR SENSORS SYNTAX

## SYNTAX

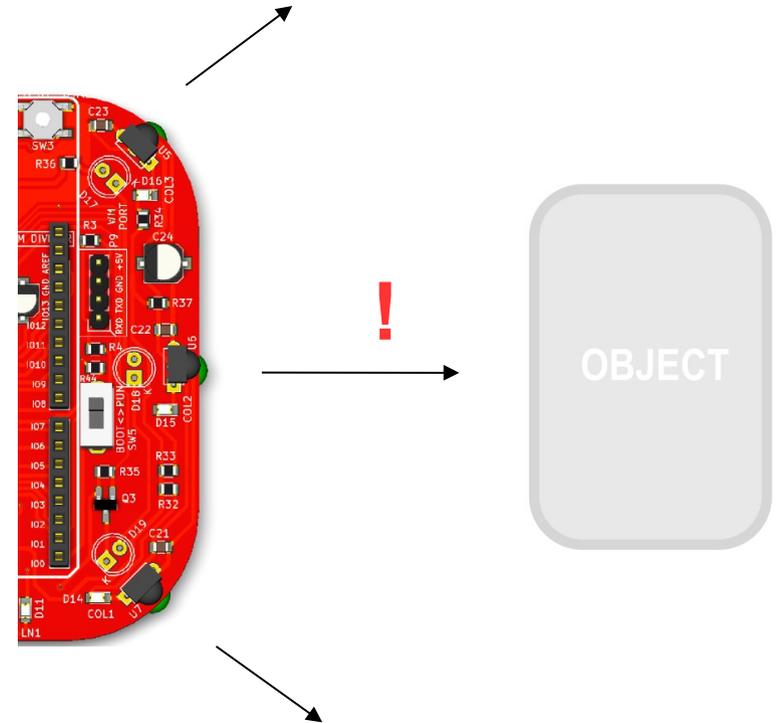
- `PBOT.COL1_RIGHT();`
- `PBOT.COL2_CENTER();`
- `PBOT.COL3_LEFT();`

## OUTPUT

- *NO OBJECT DETECTED = 0 or LOW*
- OBJECT DETECTED = 1 or HIGH

## EXAMPLES;

```
If (PBOT.COL2_CENTER() == HIGH) // if IR sensor 2 center is high
{
  Serial.println("Object detected!");
}
```

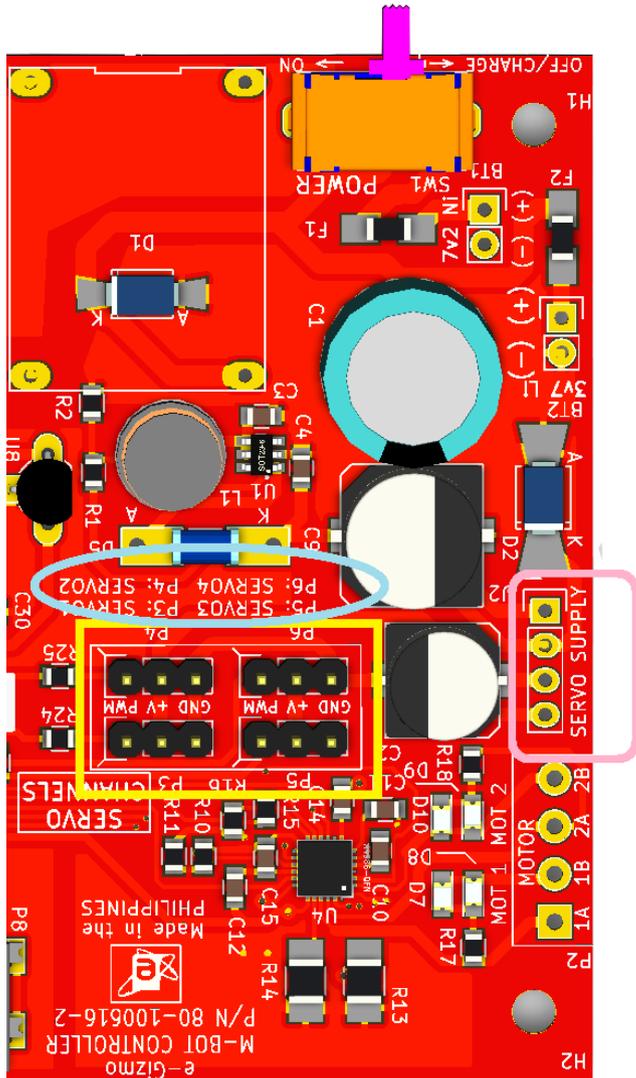


# SERVO CONNECTION

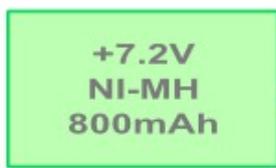
 Servo labels #1-#4

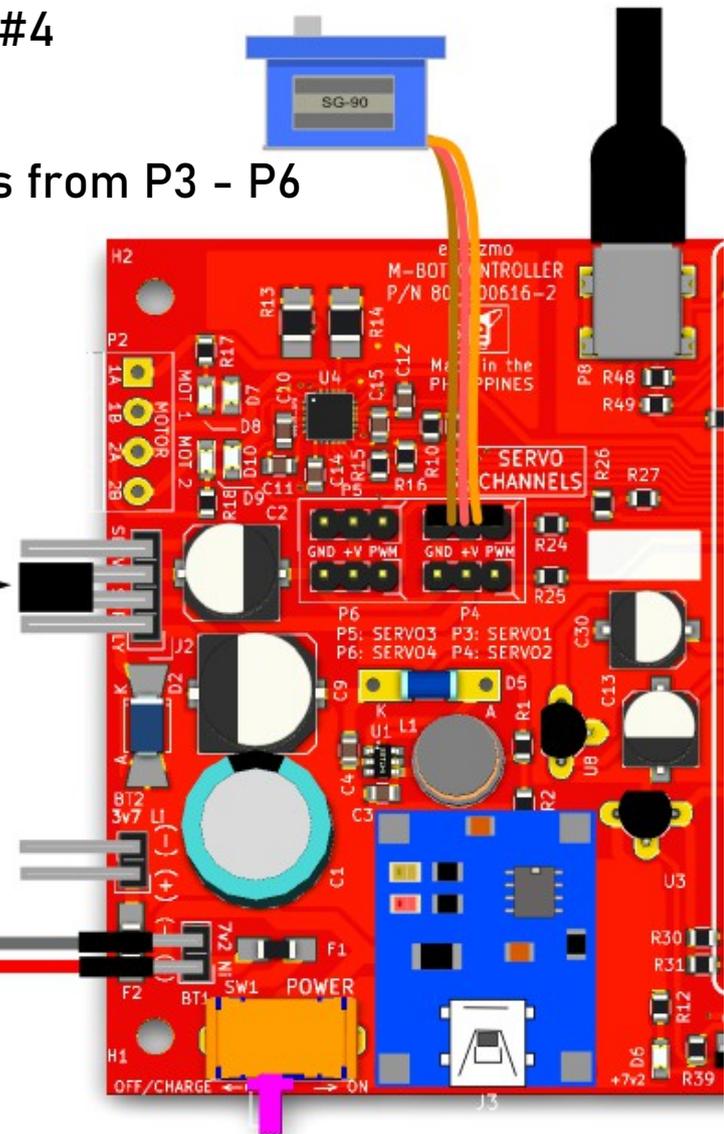
 Servo connectors from P3 - P6

 Servo Supply J2



Put a jumper

 +7.2V  
NI-MH  
800mAh

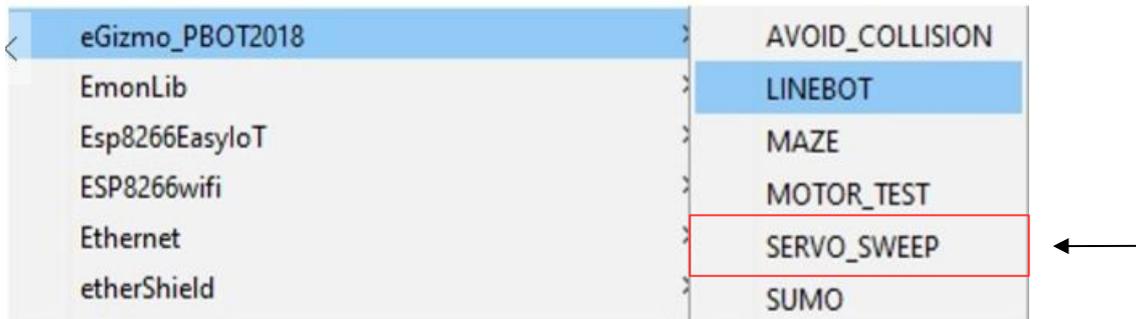


# Uploading Servo sweep.ino

On the Arduino IDE.

## 1. Maze Solver codes

Go to File>Examples>eGizmo\_PBOT2018>  
Servo\_Sweep



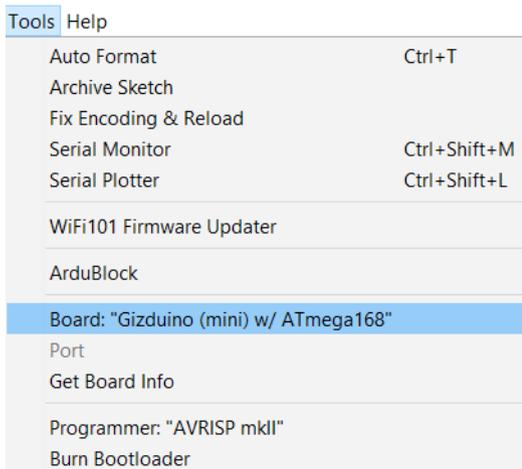
## Servo Controls

## 3. Port select

Go to Tools>Port>COM#  
Select the correct port  
Go to Device Manager  
if you're not sure.

## 2. Board select

Go to Tools>Boards>gizduino (mini) w/ Atmega168



Tips for uploading:

Press and Hold the RST (SW3)  
switch ON the power and  
Click Upload. Release RST  
when done.



# SERVO SYNTAX

## SYNTAX

PBOT.**SERVO**(which**SERVO**,**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)

## EXAMPLES;

```
PBOT.SERVO(1, 15);
```

```
delay(1000);
```

```
PBOT.SERVo(1, 90);
```

```
delay(1000);
```



# For more info:

Website: [www.e-gizmo.net](http://www.e-gizmo.net)

- Egizmo Tech blog
- Facebook: eGizmoMechatronics
- Youtube Channel: e-Gizmo Mechatronics Central