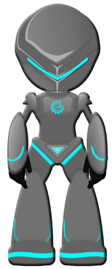
e-Gizmo
MECHATRONIX CENTRAL

# Kto12 BOT

# MANUAL GUIDE

# ROBOTICS

## A K-12 STEM EDUCATION SYSTEM

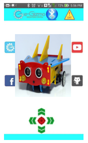


Setup
- 100% Arduino Compatible
- ATMEGA168P — w/ PL2303 Driver
- On Board Li-on Charger — 3.7V Rechargeable battery

Loop
- BEGIN — Ardublock Ready!
- Distance detection
- Maze Solver
- Collision Avoidance
- Motor Controls
- Sumo Fight
- Light & Dark Sensor
  - Automatic Headlights
  - Turn-ON in dark ambiance
- Collision/Shack detection
- Sound/Noise detection — Clap commands
- RGB LED lighting
- Bluetooth Controlled
- UHF wireless Controlled

Included:
- USB Cable Type A to mini B

*(Optional) Add HC-05 module or UHF Receiver & PS2 Controller for Wireless controlled



# 1. Table of Contents

# 2. Table of Contents

# 3. Table of Contents

# 4. Table of Contents

# I. FEATURES & SPECIFICATIONS

## On-Board Sensors

- Distance Sensor HC-SR04 (up to 5 meters range)
- LDR (Light Dependent Resistor)
- Mini Microphone (Digitial and Analog sense)
- Vibration Sensor (SW-18015P)
- Output device: Speaker 8ohms

## Li-ion Charger

- TP4056 Li-ion Battery Chargeable

## Motor Driver & connections

- 2 Channel Motor Drivers
- 2 ZebraZS-135 Motors

## LED Lights

- 2 Yellow LED Headlights
- 1 RGB LED

## On-Board IC

- Atmega168P (16KB Flash Memory)
- PL2303 Driver

## On-Board Perpherals

- On/Off Switch
- 3.7V Rechargeable Battery
- USB Mini B Connector
- Connection for I2C and Serial

- With eGizmo_KBOT Library

- PCB Dimensions: mm x mm

# II. MAJOR PARTS

# Kto12BOT BOARD

## Motor Driver

2 DC Motors

## Power Switch

On/Off Charging Switch

## Charger/Battery

Li-ion Charger TP4056

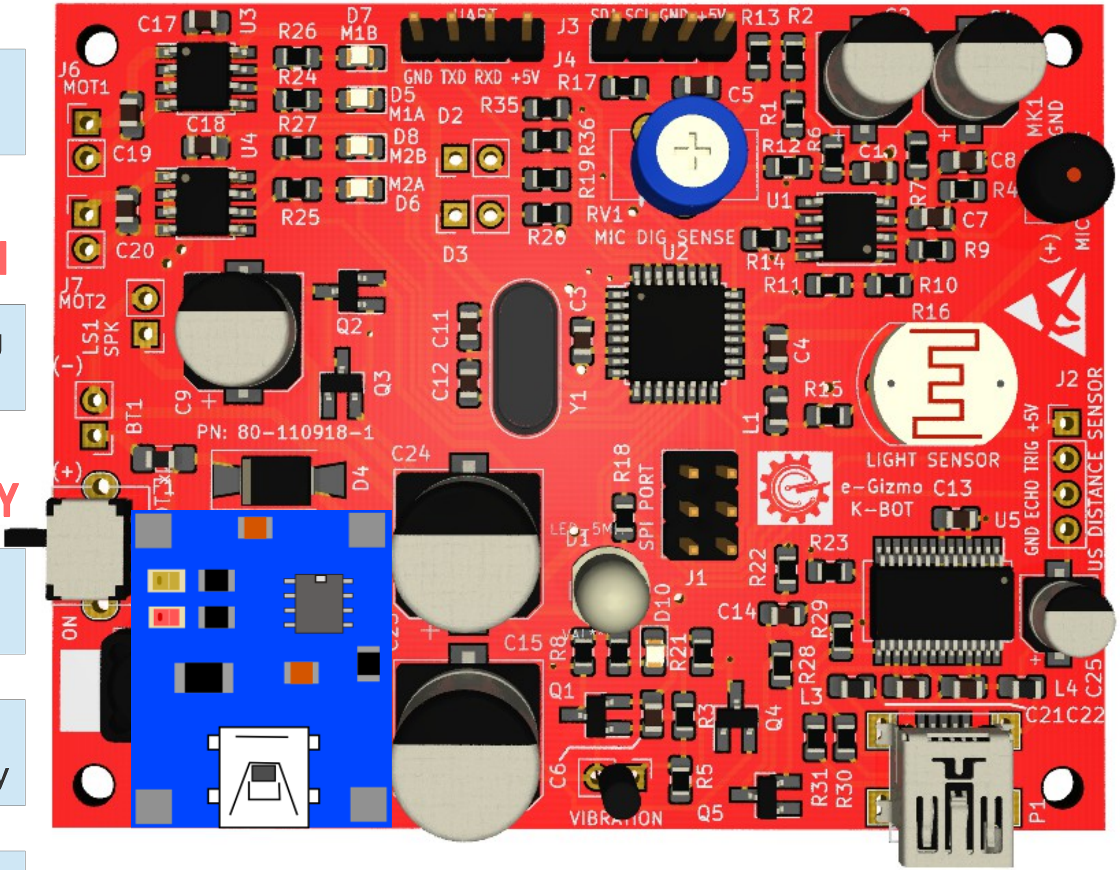
3.7V Li-ion Rechargeable Battery

USB Connector MiniTypeB

2 Channel Motor Drivers

Serial Connections

I2C connections



Prolific Driver PL2303

## Drivers

ATMega168P 16KB Flash Memory

## Microcontroller

Chassis

## Sensors

Vibration Sensor SW-18015P

Sound Sensor Analog/Digital Mini Microphone

Light/Dark Sensor ,LDR

HC-SR04 Distance Sensor Up to 4.5 Meters

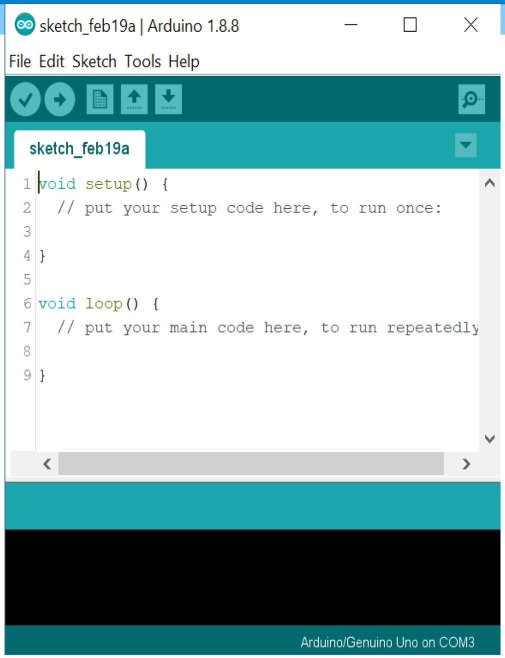
## Output Components

Speaker 8ohms Buzzer

2 Headlights Yellow LED

RGB LED lights

# III. SOFTWARES AND LIBRARY INSTALLATION


sketch_feb19a | Arduino 1.8.8
File Edit Sketch Tools Help
sketch_feb19a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly

}
```

Arduino/Genuino Uno on COM3

## 1. Arduino IDE

www.e-gizmo.net/oc/kits documents/ARDUINO IDE SOFTWARES

- Download Arduino 1.8.8 (Windows)
- Or choose your Arduino IDE for your OS.

## 2. Drivers Install this first!

Go to Arduino 1.8.8 folder>Drivers>PL2303 Driver

- Install the PL2303_Prolific_DriverInstaller_v1.10.0
- (For Mac OS users) Download md_PL2303_MacOSX

## Library Already added

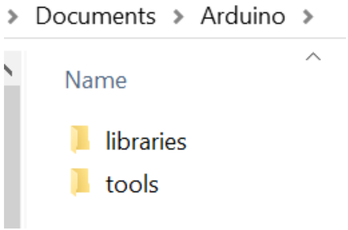
- eGizmo_Kto12BOT

Examples
>BluetoothControlled
> FireAlert
> Headlights
>LightNDarkSensor
> LightSpeedCtrl
> Maze Solver
> Motor_Test
>PS2Controlled
> Sensors (Distance,Vibration,Sound,Light&Dark)
> Siren_Sounds
>Sound_Sensor
> SumoFight

## 3. Ardublock Add this on...

- Ardublock – 031319.jar

Copy the ***tools*** folder which contains "ardublock jar" file and Place it to My Documents>Arduino folder.

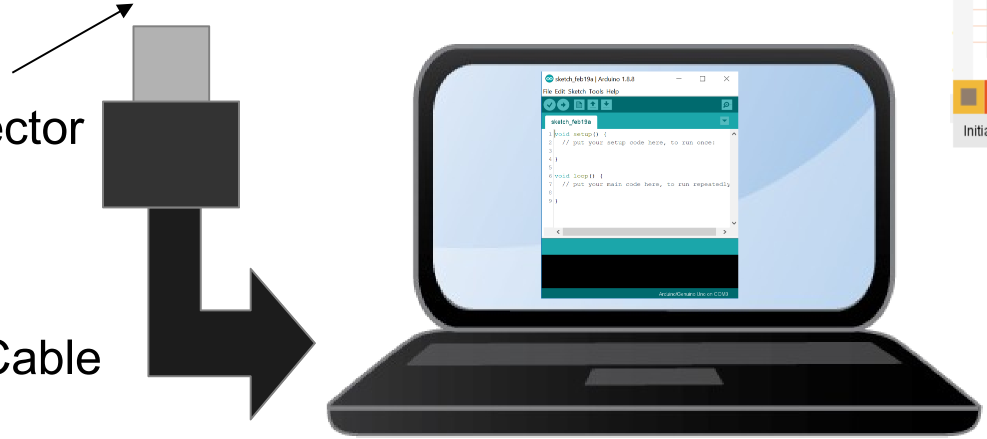


> Documents > Arduino >

Name
libraries
tools

Be sure that you did not
open the Arduino IDE yet.
ReOpen the IDE after you add this.
( Go to Tools> Ardublock must be included)

# Connect the Kto12BOT to PC



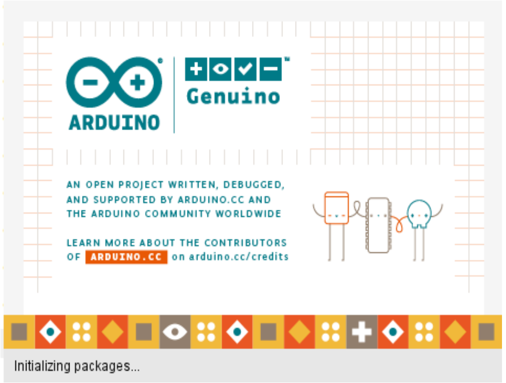
USB Connector

USB Cable

sketch_feb19a | Arduino 1.8.8
File Edit Sketch Tools Help
sketch_feb19a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly
  
}
```

Arduino/Genuino Uno on COM3



AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

Initializing packages...

Open Arduino IDE.

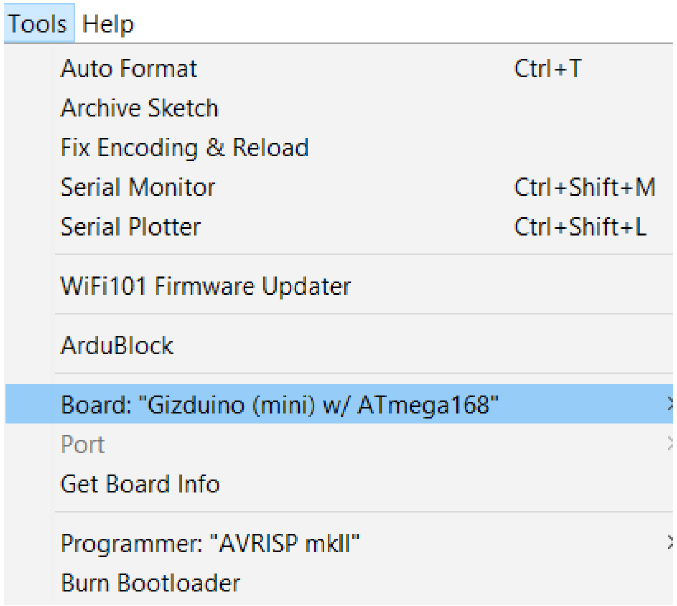
# IV. Arduino IDE 1.8.8

## The Software

On the Arduino IDE.

1. Board select

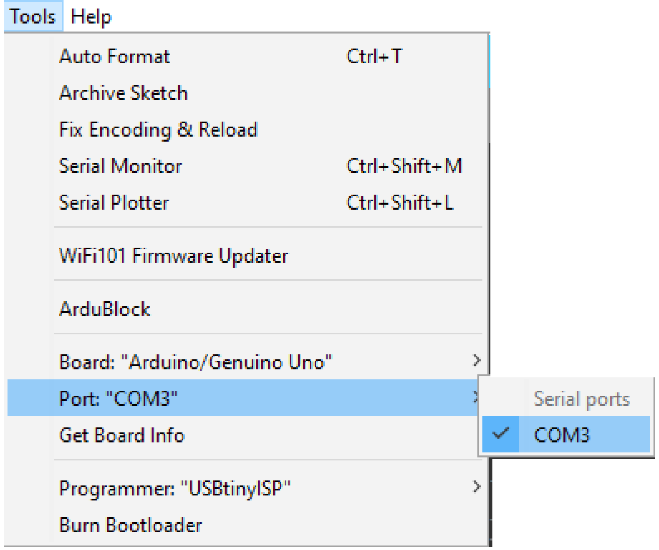
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168



3. Port select

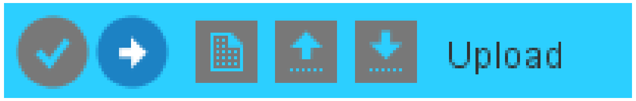
   Go to Tools>Port>COM#

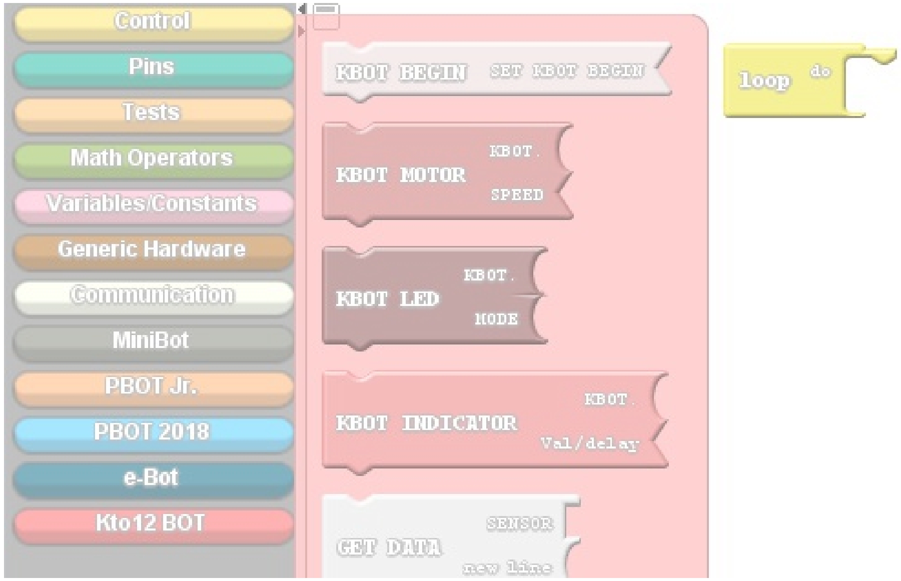
- Select the correct port
- Go to Device Manager if you're not sure.



4. Click Upload

   Wait until its Done Uploading



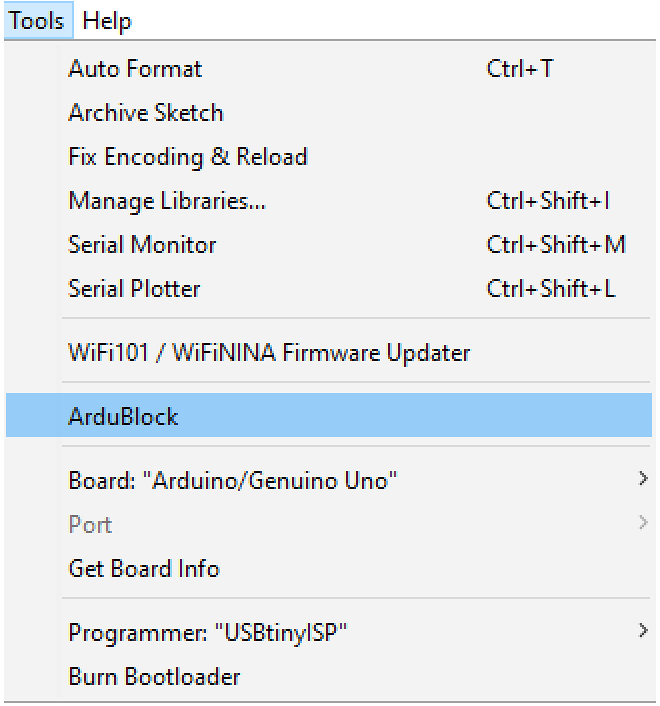


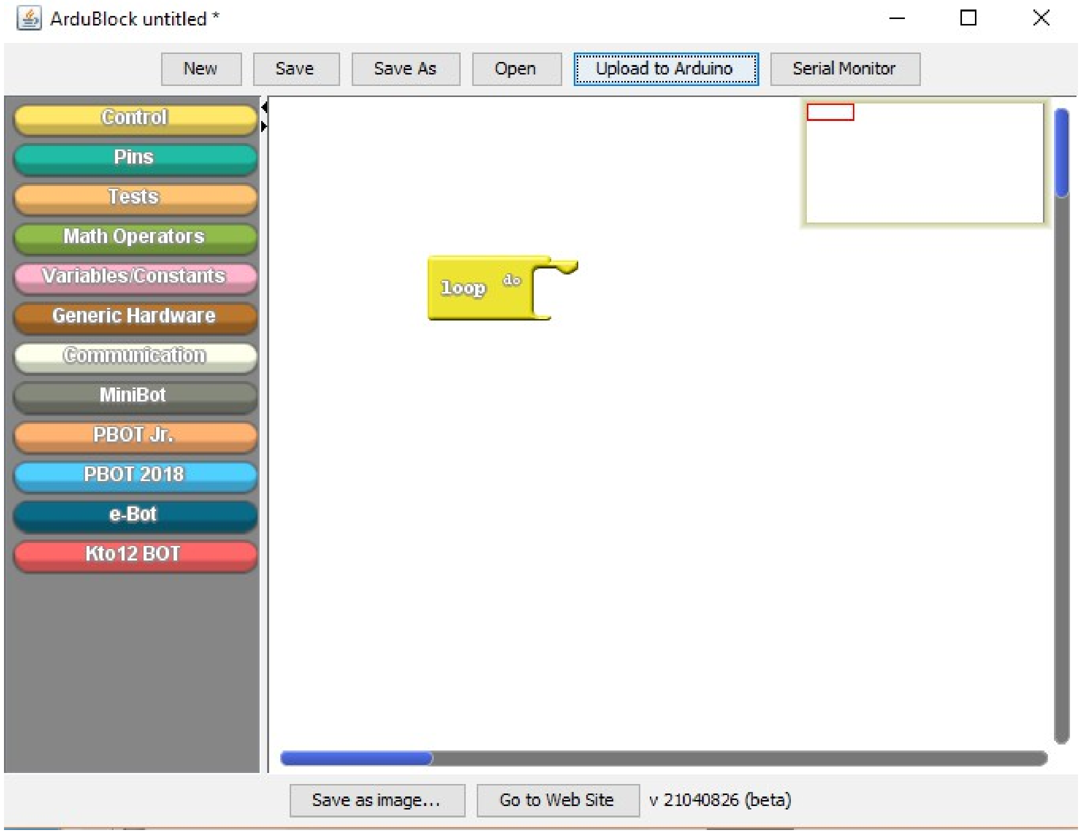
# V. Getting Started with ArduBlock

# a. Intro to Ardublock

On the Arduino IDE.

1. Open the Ardublock.
   Go to Tools>Click the ArduBlock.

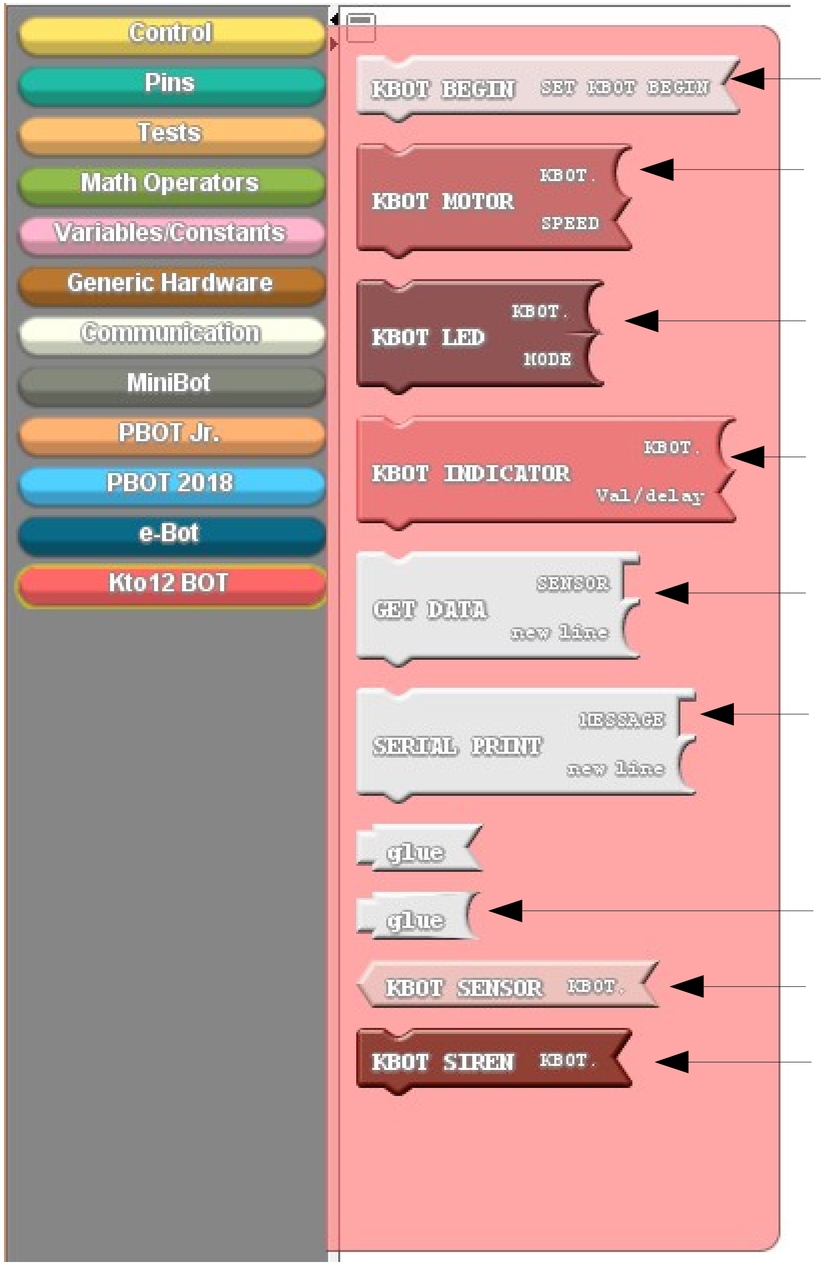

Tools
Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ArduBlock
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "USBtinyISP"
Burn Bootloader



# b. Block descriptions

On the Ardublock.

Click the Kto12 BOT block.



**KBOT BEGIN** – setting all the pins.

**KBOT MOTOR** – to controls the motors

**KBOT LED** – to use the LEDs

**KBOT INDICATOR** – to controls the RGB and Headlights delay.

**GET DATA** – read the sensors

**SERIAL PRINT** – print the data To the Serial Monitor.

Default 9600 baudrate.

**GLUE** – attached the sensors

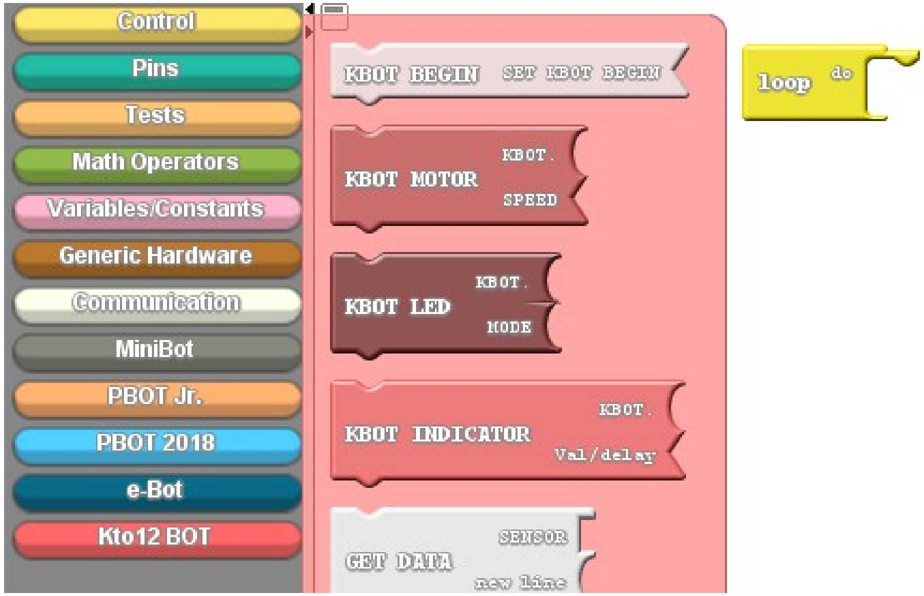
**KBOT SENSOR** – to use the sensor

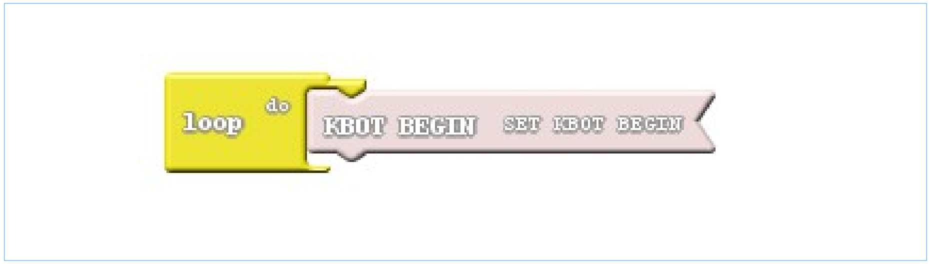
**KBOT SIREN** – for sounds

# c. ADD BEGIN BLOCK

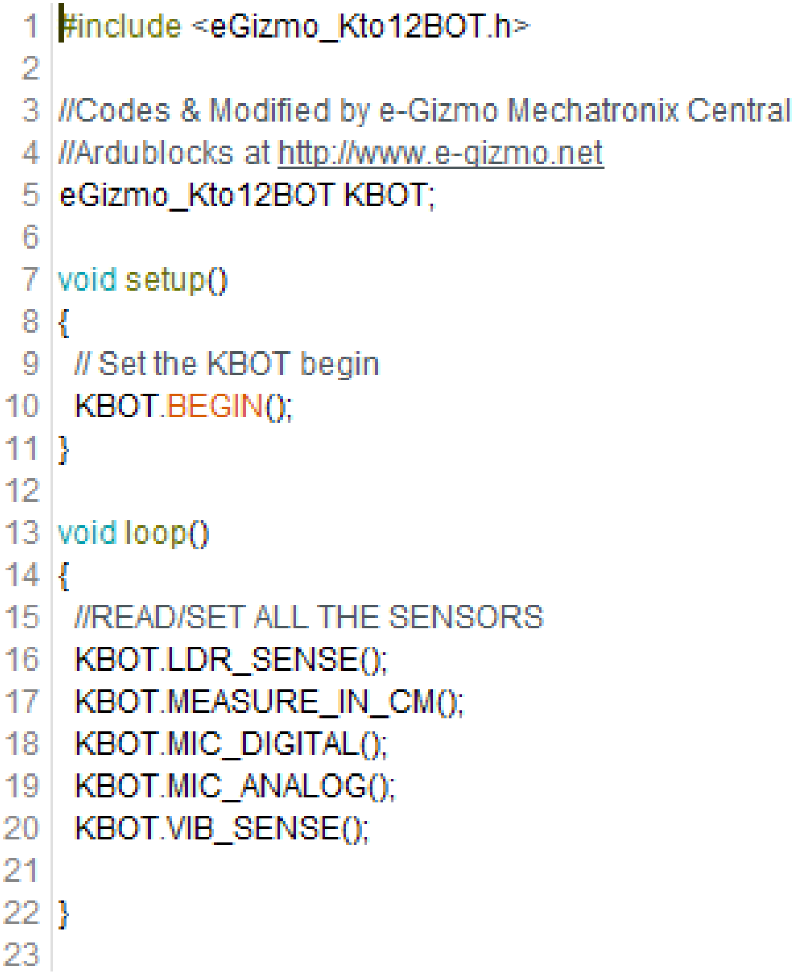
Always add the KBOT BEGIN to set it.

1. Click and Drag it and attached to Loop block.



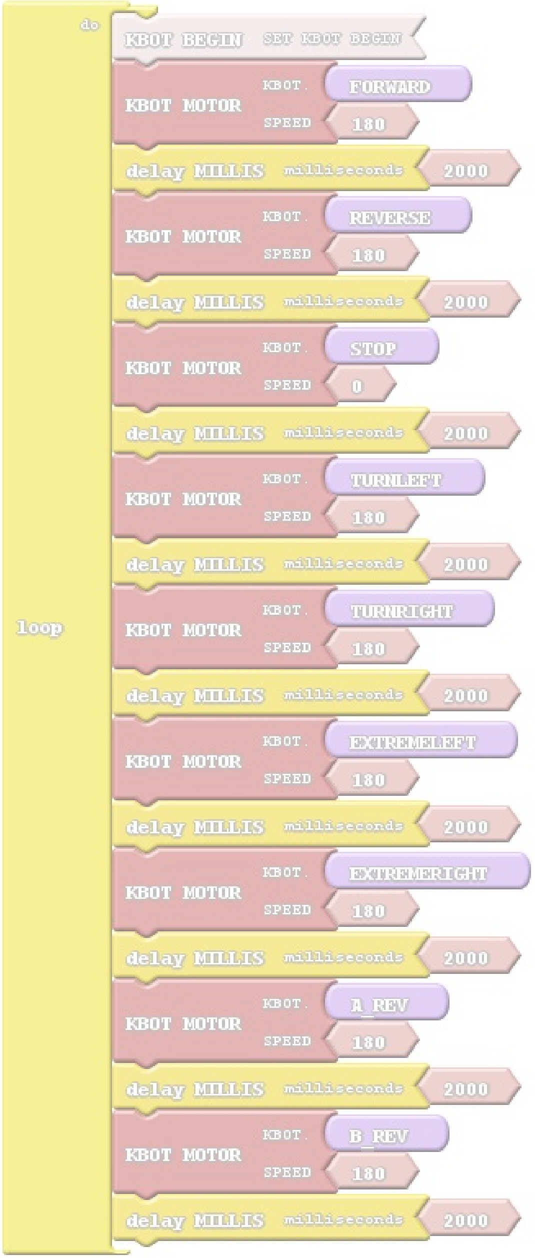


2. If you click Upload to Arduino.
(Blocks to Text Generated)



```
#include <eGizmo_Kto12BOT.h>

//Codes & Modified by e-Gizmo Mechatronix Central
//Ardublocks at http://www.e-gizmo.net
eGizmo_Kto12BOT KBOT;

void setup()
{
  // Set the KBOT begin
  KBOT.BEGIN();
}

void loop()
{
  //READ/SET ALL THE SENSORS
  KBOT.LDR_SENSE();
  KBOT.MEASURE_IN_CM();
  KBOT.MIC_DIGITAL();
  KBOT.MIC_ANALOG();
  KBOT.VIB_SENSE();

}

```

# VI. Ardublock Examples

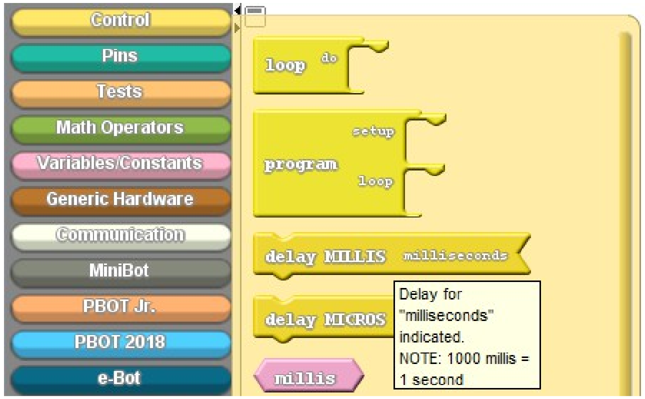


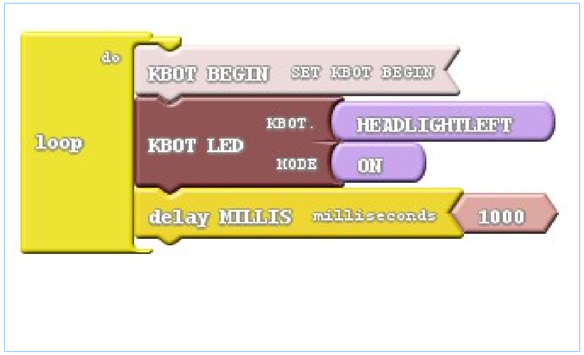
```
loop
  do
    KBOT BEGIN   SET KBOT BEGIN
    KBOT MOTOR   KBOT. FORWARD        SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. REVERSE        SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. STOP           SPEED 0
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. TURNLEFT       SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. TURNRIGHT      SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. EXTREMELEFT    SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. EXTREMERIGHT   SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. A_REV          SPEED 180
    delay MILLIS milliseconds 2000
    KBOT MOTOR   KBOT. B_REV          SPEED 180
    delay MILLIS milliseconds 2000
```

# 1. BLINKING THE HEADLIGHTS

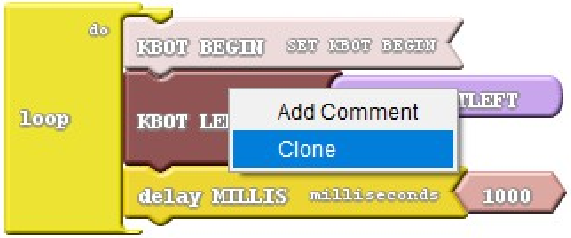
Add the KBOT LED to set it.

1.Click and Drag it and attached to Loop block.

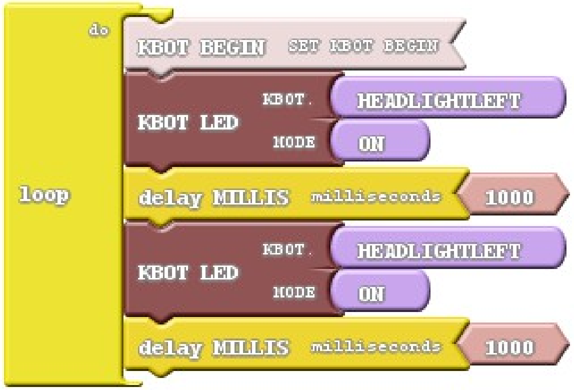




2.You can clone the block by pressing Right-Click of your mouse then select clone.



3.Add to the 3rd blocks.



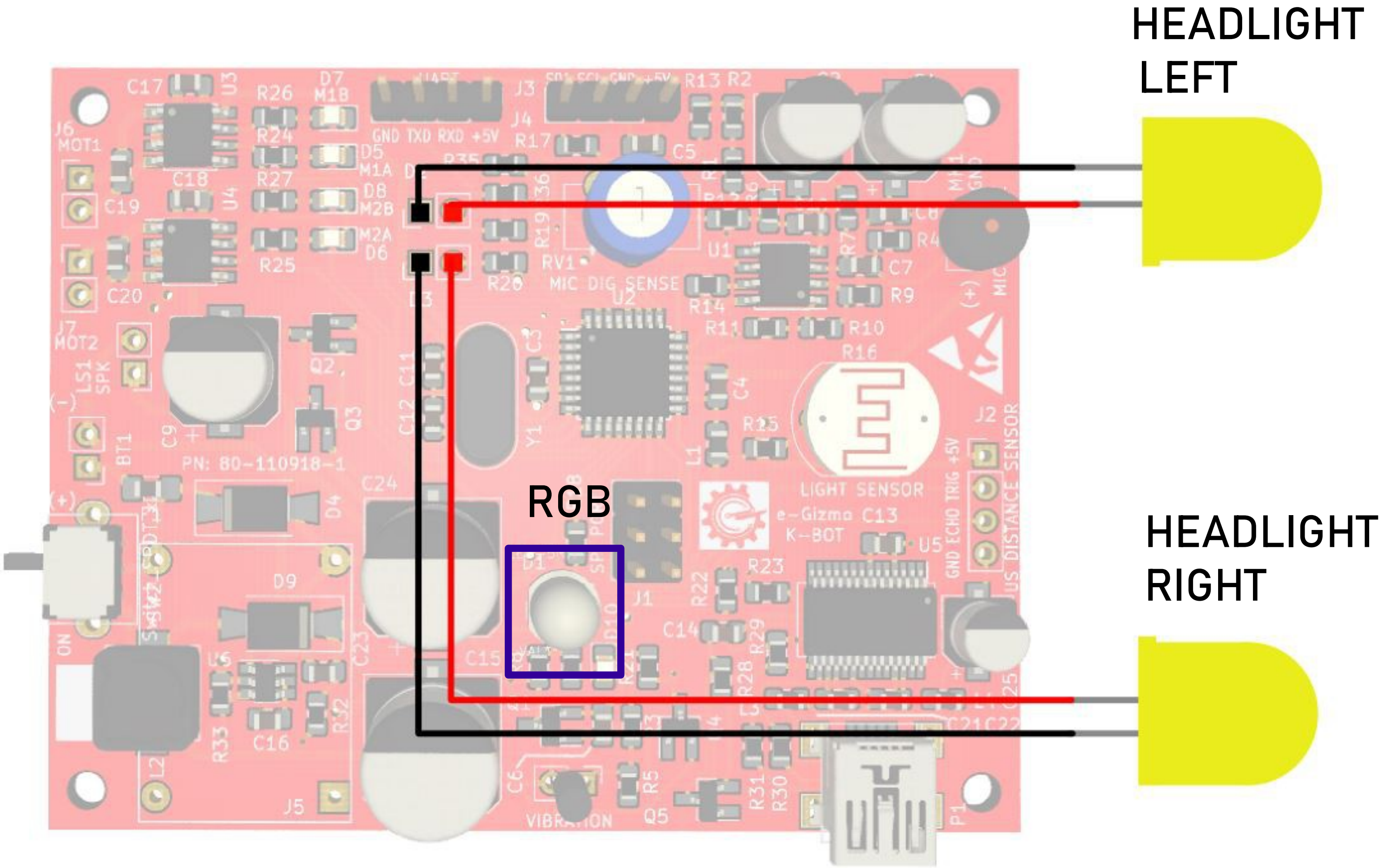
4.Then Click
Upload to Arduino.

It will add this line on the loop.

```
KBOT.HEADLIGHT_LEFT(1);
delay( 1000 );
KBOT.HEADLIGHT_LEFT(1);
delay( 1000 );
```

See what happens?

# 1.a LED Lights diagram



HEADLIGHT
LEFT

RGB

HEADLIGHT
RIGHT

# 1.b LED SYNTAX & BLOCKS

```
// SYNTAX
```

- KBOT.RGB(1);
- KBOT.HEADLIGHT_RIGHT(1);
- KBOT.HEADLIGHT_KEFT(1);

Where:

- 1 = ON-state
- 0 = OFF-state

- KBOT.REDRGB(240);
- KBOT.WHITERGB(250);
- KBOT.SIGNAL_LIGHT(1000);

Where:

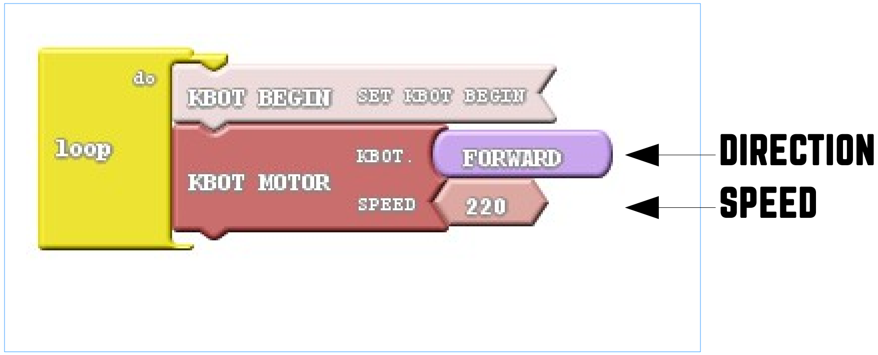
- 240 = rgb set to color red
- 250 = rgb set to color white
- 1000 = delay time for blinking two headlights

# 2. Controlling the Motors

# 2.A MOVE FORWARD

First add the “KBOT BEGIN” to set it.

1.Click KBOT MOTOR and Drag it
Then attached to Loop block.



DIRECTION

SPEED

2.Then Click Upload to Arduino.

It will add this line on the loop.

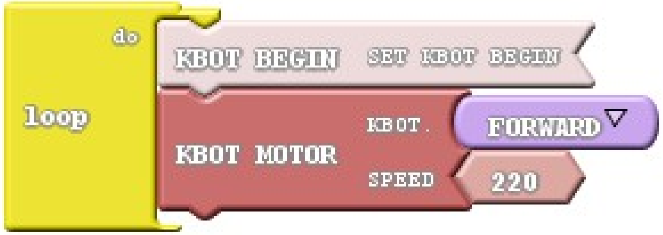
```
KBOT.FORWARD(220);
```

See what happens?

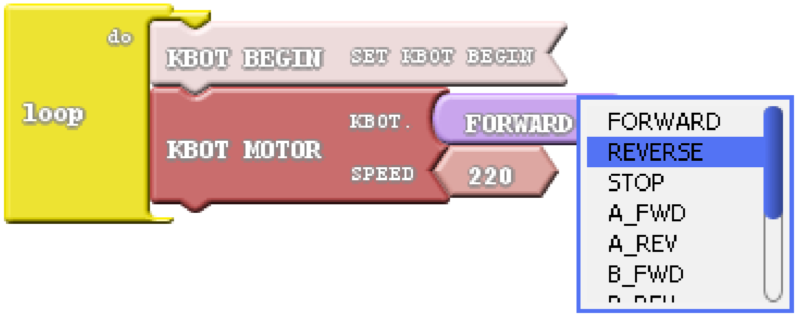
# 2.B REVERSE

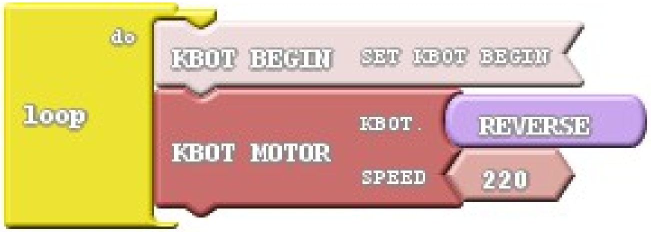
Select the direction of the motors.

1.Click the drop down arrow (shown below).



2. Select the REVERSE.





3.Upload to Arduino.

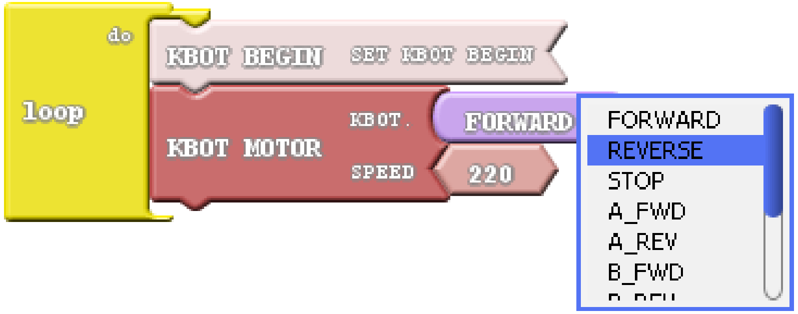
It will add this line on the loop.

```
KBOT.REVERSE(220);
```

See what happens?

# 2.c MOTOR DIRECTIONS SYNTAX & Blocks

Select the directions.
1.Click the drop down arrow (shown below).



- KBOT.FORWARD(SPEED);
- KBOT.REVERSE(SPEED);
- KBOT.STOP(0);
- KBOT.A_FWD(SPEED);
- KBOT.A_REV(SPEED);
- KBOT.B_FWD(SPEED);
- KBOT.B_REV(SPEED);
- KBOT.TURNLEFT(SPEED);
- KBOT.TURNRIGHT(SPEED);
- KBOT.EXTREMERIGHT(SPEED);
- KBOT.EXTREMELEFT(SPEED);

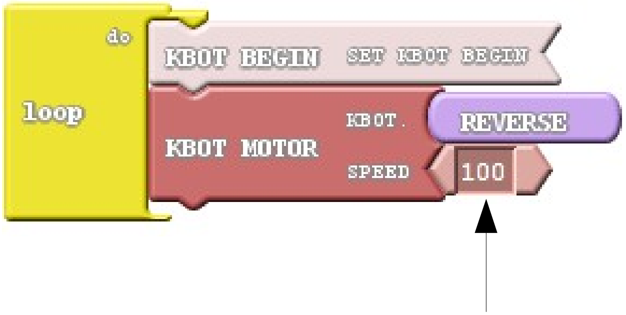
EXAMPLE: //inside the loop

KBOT.FORWARD(220); // Move Forward

# 2.D Changing Motor Speed & blocks

Set the Speed.

1.Click the number block. Change the Value from 0 to 255.



**SPEED**

- KBOT.FORWARD(SPEED);
- KBOT.REVERSE(SPEED);
- KBOT.STOP(0);
- KBOT.A_FWD(SPEED);
- KBOT.A_REV(SPEED);
- KBOT.B_FWD(SPEED);
- KBOT.B_REV(SPEED);
- KBOT.TURNLEFT(SPEED);
- KBOT.TURNRIGHT(SPEED);
- KBOT.EXTREMERIGHT(SPEED);
- KBOT.EXTREMELEFT(SPEED);

where:

SPEED = 0 , Full Stop; 255 = Maximum Speed
180 = Normal Speed

EXAMPLE: //inside the loop

KBOT.REVERSE(100); // Move Backward

# Save your work.

# 2.E UPLOADING MOTOR TEST.ABP

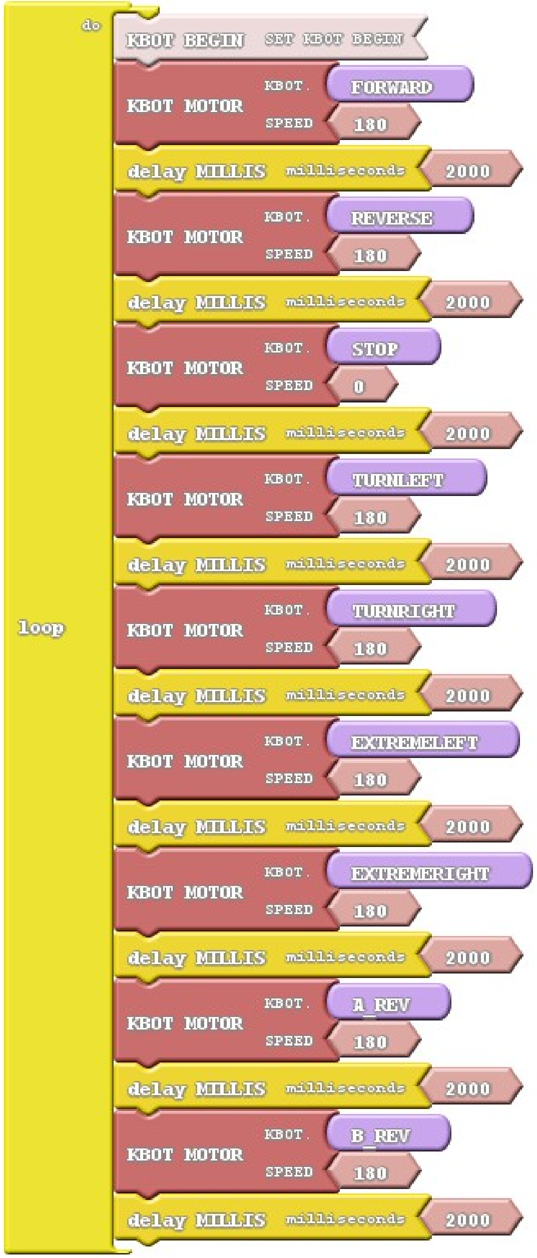
On the Ardublock.
1. Click OPEN.

Select the MOTOR_TEST.abp

With 2000 milliseconds delay time.
(2 seconds) and 180 speeds.

- Forward
- Reverse
- Stop
- Turn left
- Turn right
- Extreme left
- Extreme right
- A reverse
- B reverse

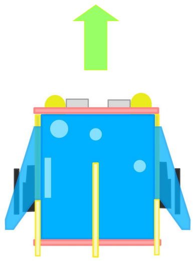
2. Upload to Arduino.



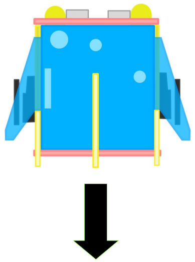
```
loop
do
  KBOT BEGIN  SET KBOT BEGIN
  KBOT MOTOR  KBOT. FORWARD
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. REVERSE
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. STOP
              SPEED 0
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. TURNLEFT
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. TURNRIGHT
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. EXTREMELEFT
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. EXTREMERIGHT
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. A_REV
              SPEED 180
  delay MILLIS  milliseconds 2000
  KBOT MOTOR  KBOT. B_REV
              SPEED 180
  delay MILLIS  milliseconds 2000
```

# 2.F MOTOR MOVEMENTS ILLUSTRATION

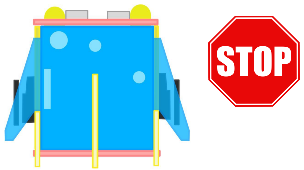
Forward



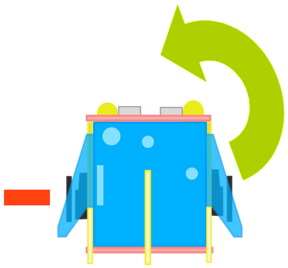
Reverse



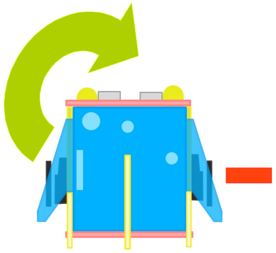
Stop



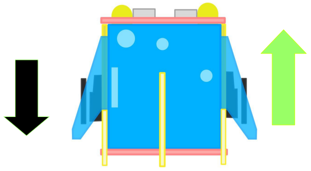
Turn Left



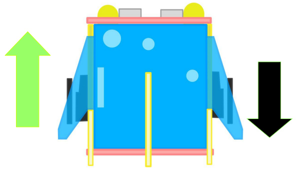
Turn Right



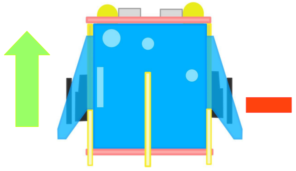
Extreme Left



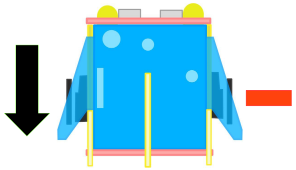
Extreme Right



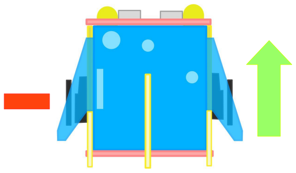
Forward A



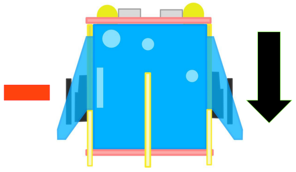
Reverse A



Forward B



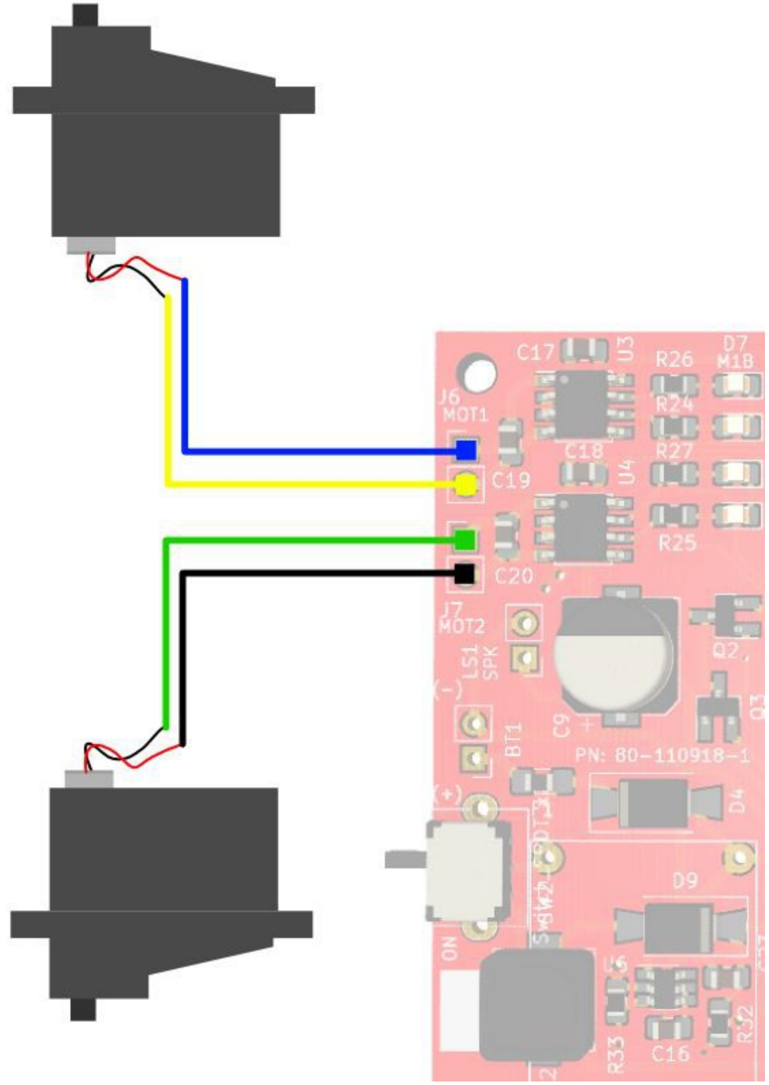
Reverse B



# MOTOR CONNECTIONS

- Left Motor (J6)
  - M1B → Black wire
  - M1A → Red wire
- Right Motor (J7)
  - M2B → Black wire
  - M2A → Red wire

LEFT
MOTOR 1 / A



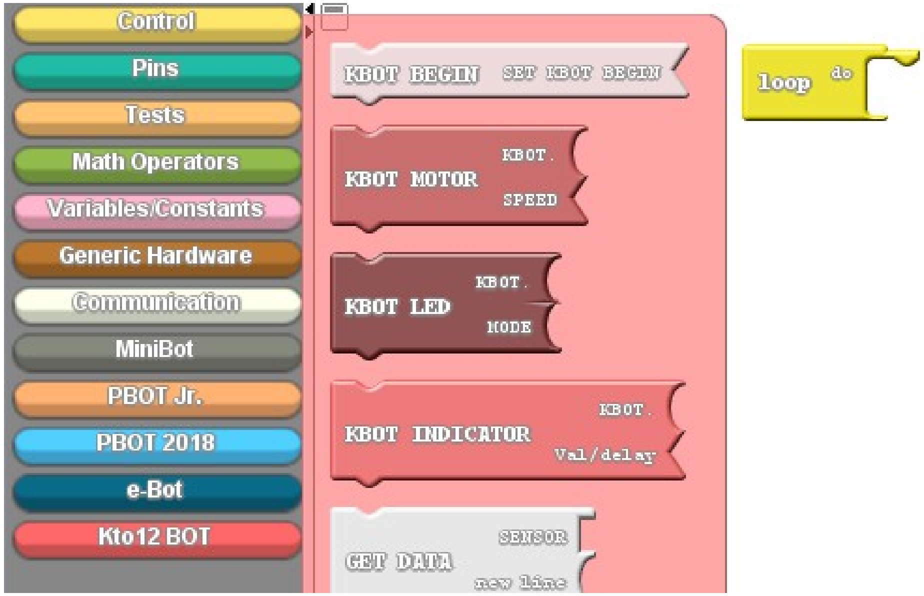
RIGHT
MOTOR 2 / B

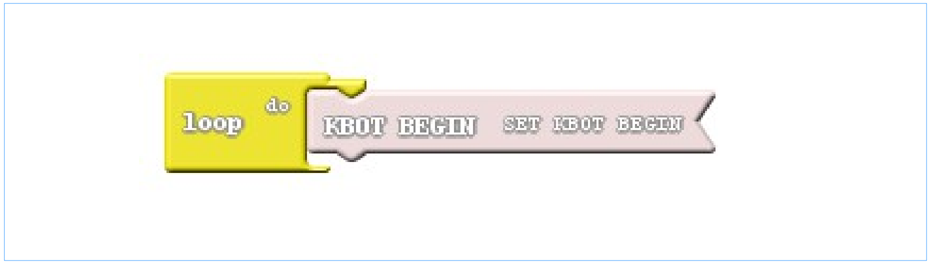
# 3. Distance Sensor
# &
# Detecting Objects

# 3.A ADD BEGIN

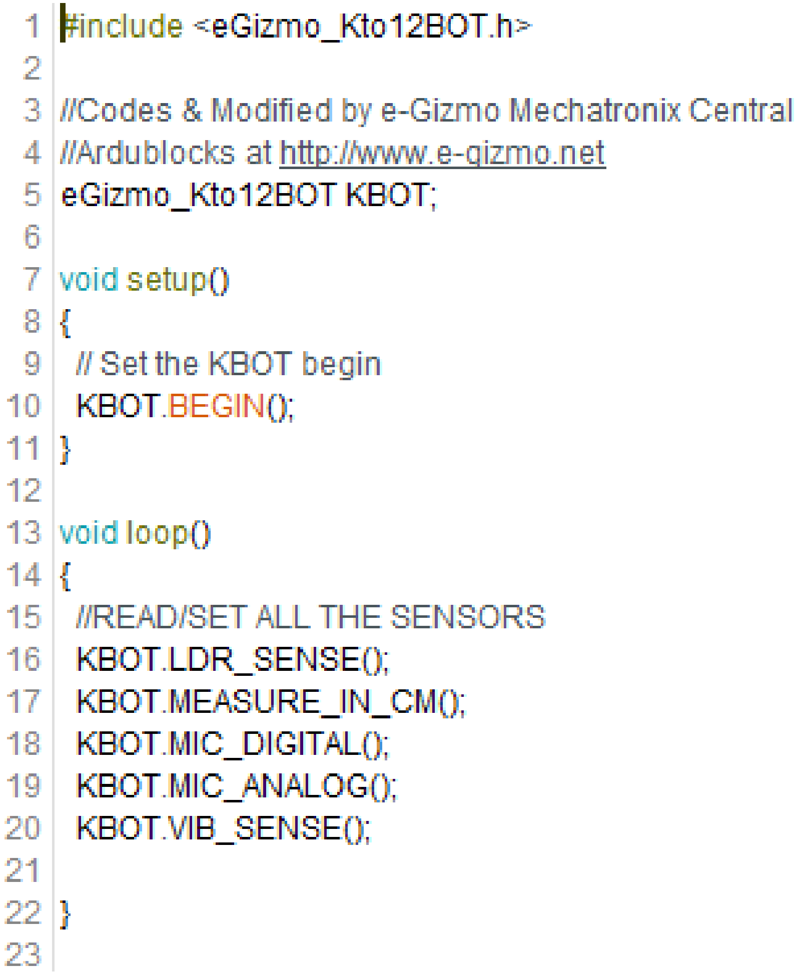
Always add the KBOT BEGIN to set it.

1. Click and Drag it and attached to Loop block.





2. If you click Upload to Arduino. (Blocks to Text Generated)

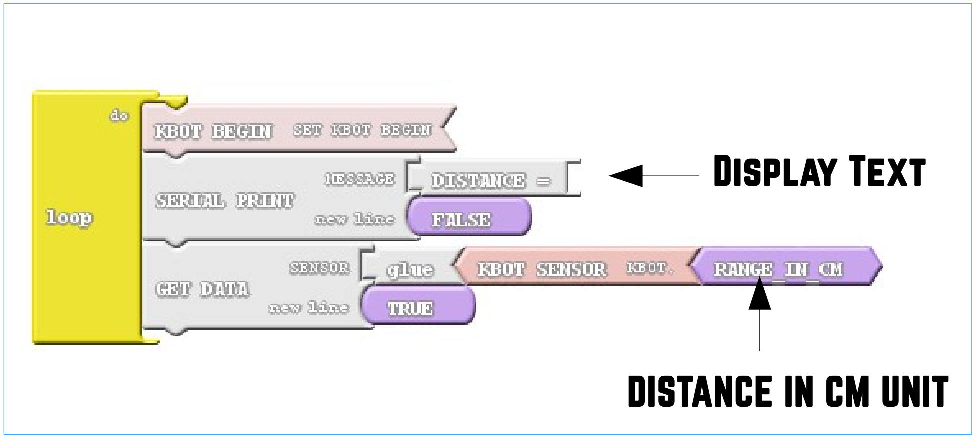


```
#include <eGizmo_Kto12BOT.h>

//Codes & Modified by e-Gizmo Mechatronix Central
//Ardublocks at http://www.e-gizmo.net
eGizmo_Kto12BOT KBOT;

void setup()
{
  // Set the KBOT begin
  KBOT.BEGIN();
}

void loop()
{
  //READ/SET ALL THE SENSORS
  KBOT.LDR_SENSE();
  KBOT.MEASURE_IN_CM();
  KBOT.MIC_DIGITAL();
  KBOT.MIC_ANALOG();
  KBOT.VIB_SENSE();

}

```

# 3.B Distance Sensor blocks

First add the “KBOT BEGIN” to set it.

1.Click KBOT SENSOR and Drag it
Then attached to Loop block.



Display Text

DISTANCE IN CM UNIT

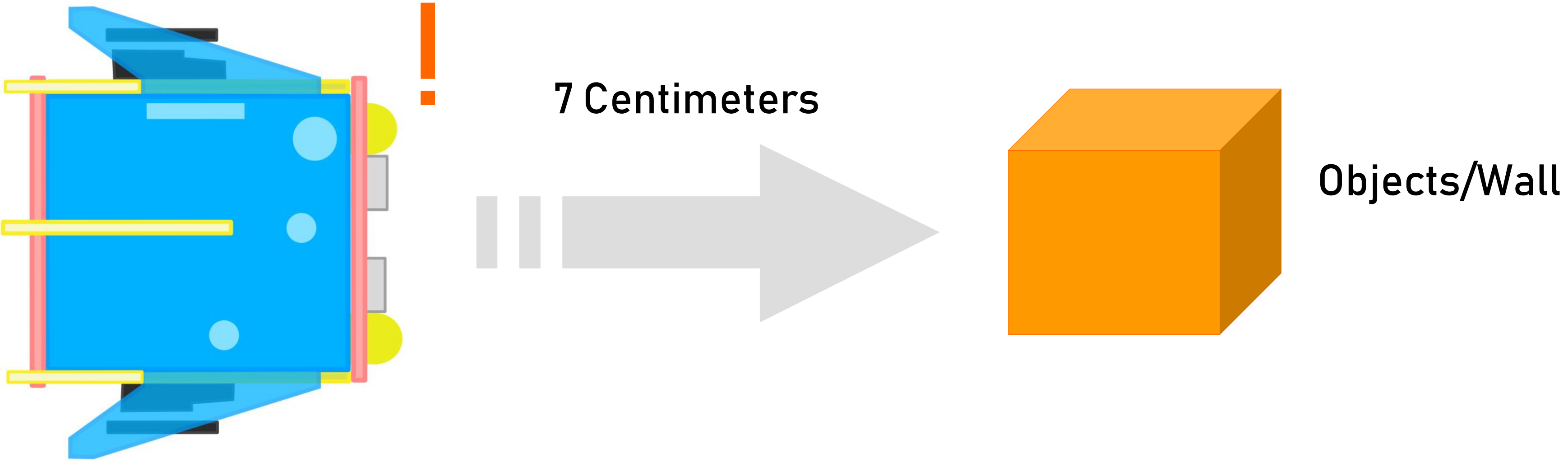
2.Then Click Upload to Arduino.

It will add this lines to the loop.

```
KBOT.PRINT(“DISTANCE=”);
KBOT.GET_DATA(KBOT.RANGE_IN_CM);
KBOT.PRINTLN(“”);
```

See what happens?

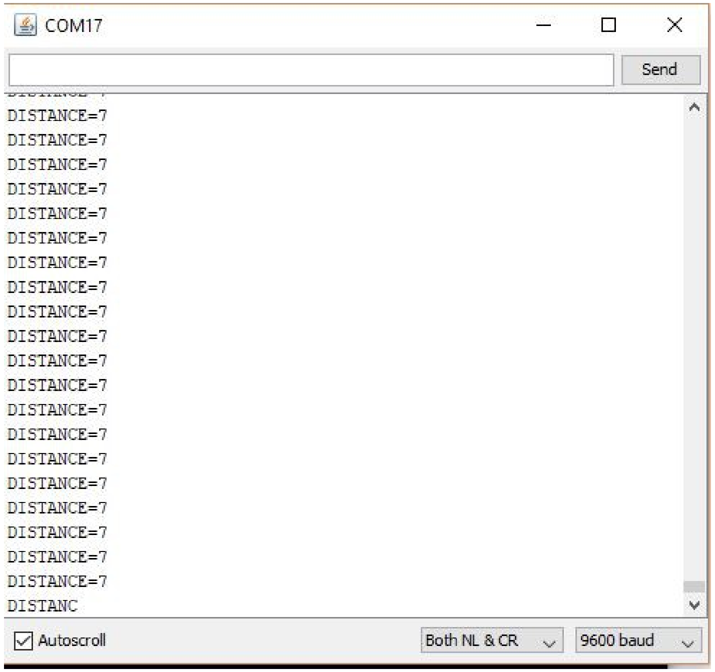
# 3.C Reading Distance output Illustration



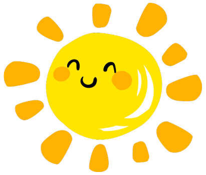
7 Centimeters

Objects/Wall

Serial Monitor



```
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANCE=7
DISTANC
```

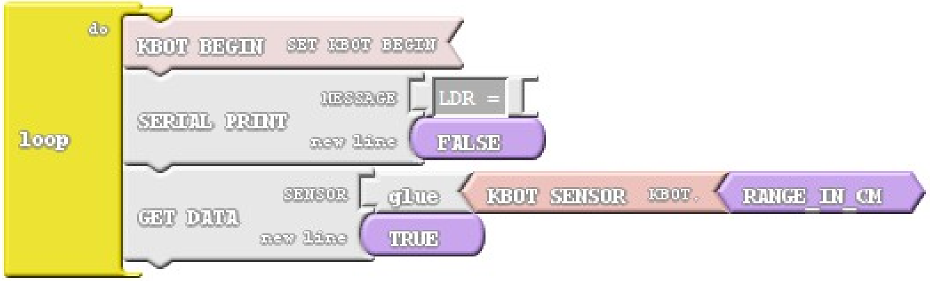


# 4. Light & Dark sensor

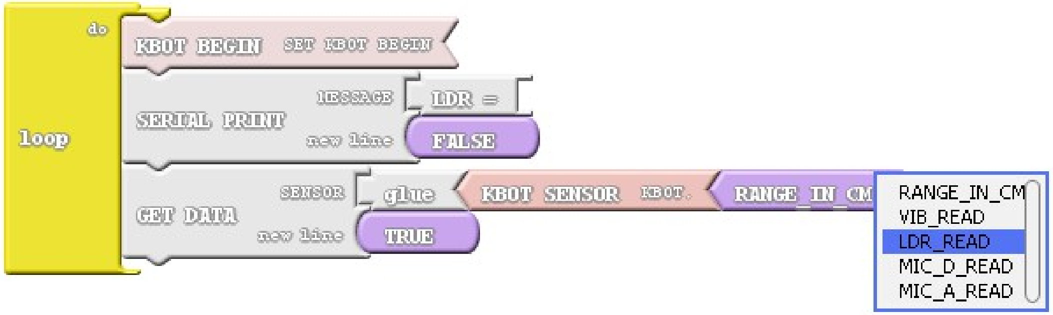
# 4.A Light & Dark Sensor Blocks

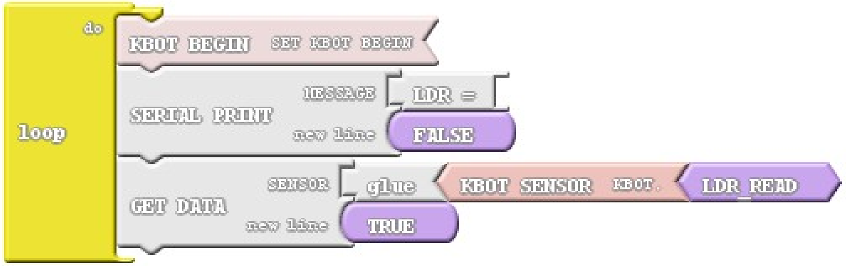
Select the LDR sensor to read.

1.Click the message (shown below) change it to LDR =.



2. Select the LDR_READ.





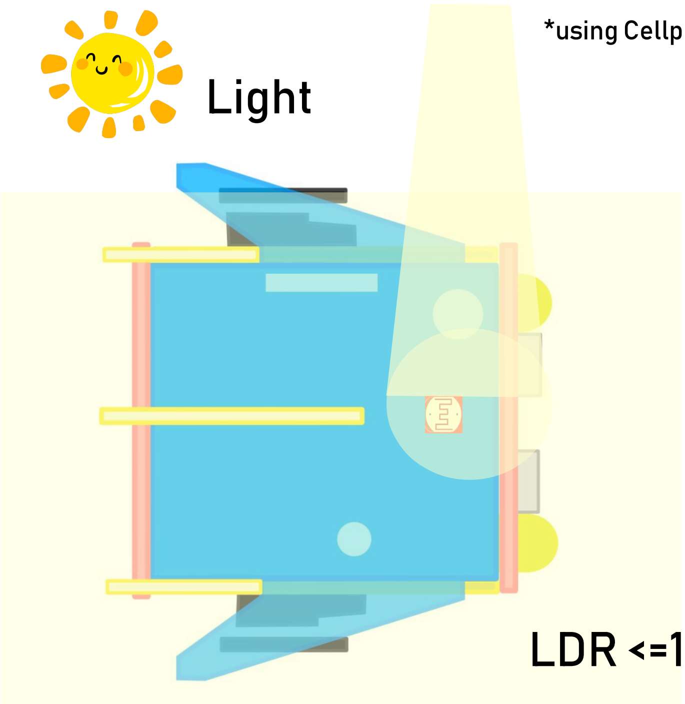
3.Upload to Arduino.

It will add this line on the loop.

```
KBOT.PRINT(“LDR=”);
KBOT.GET_DATA(KBOT.LDR_READ);
KBOT.PRINTLN(“”);
```

See what happens?

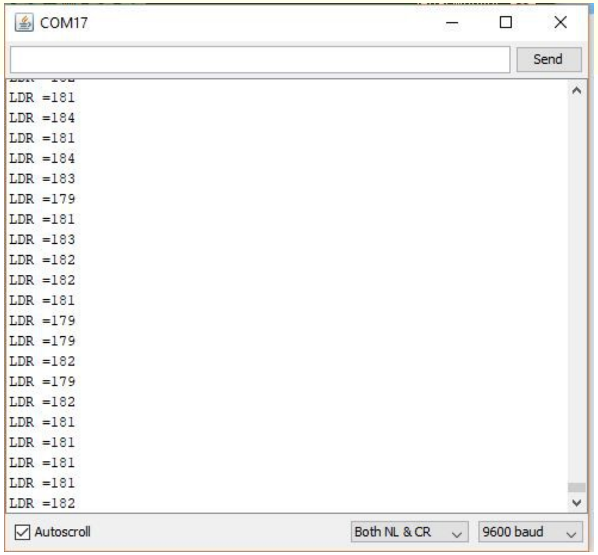
# 4.B Reading Light & Dark Output ILLUSTRATION



Light

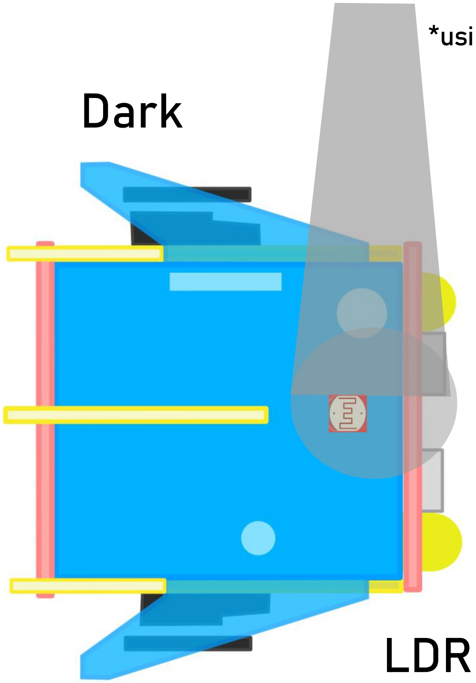
*using Cellphone Flashlight

LDR <=184



```
LDR =181
LDR =184
LDR =181
LDR =184
LDR =183
LDR =179
LDR =181
LDR =183
LDR =182
LDR =182
LDR =181
LDR =179
LDR =179
LDR =182
LDR =179
LDR =182
LDR =181
LDR =181
LDR =181
LDR =181
LDR =182
```

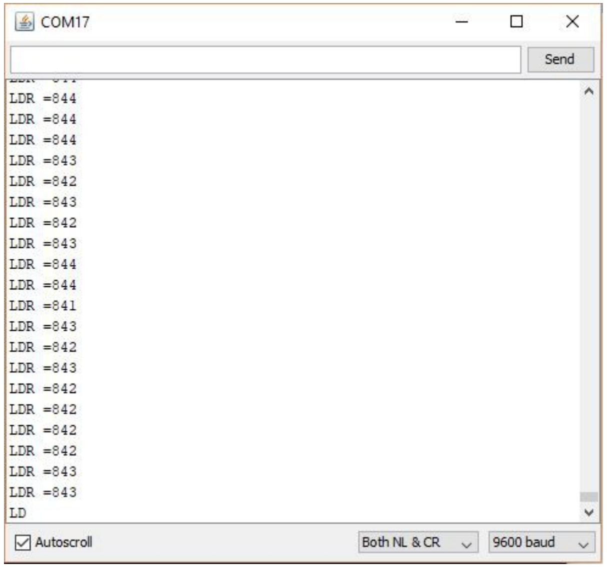
☑ Autoscroll Both NL & CR 9600 baud



Dark

*using hand for cover

LDR >=843



```
LDR =844
LDR =844
LDR =844
LDR =843
LDR =842
LDR =843
LDR =842
LDR =843
LDR =844
LDR =844
LDR =841
LDR =843
LDR =842
LDR =843
LDR =842
LDR =842
LDR =842
LDR =842
LDR =843
LDR =843
LD
```

☑ Autoscroll Both NL & CR 9600 baud

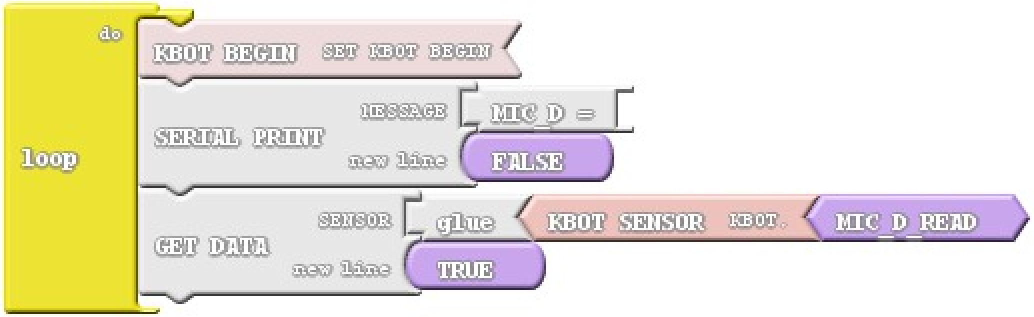
# 5. Sound Detection Sensor

# 5.A SOUND SENSOR BLOCKS

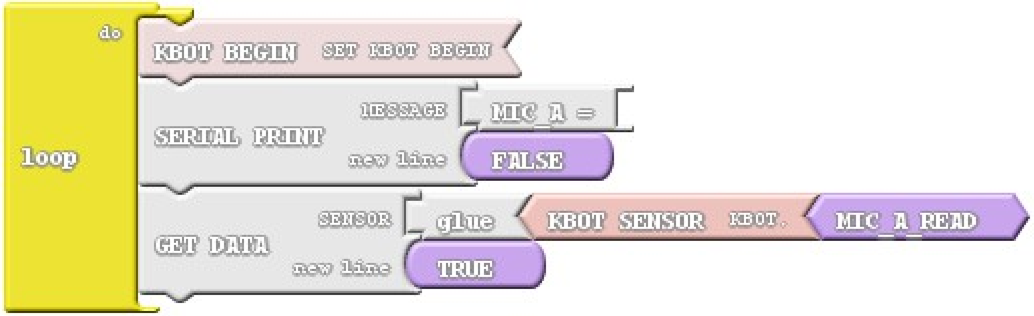
Select the Sound sensor to read.

1.Click the message (shown below) change it to MIC_D =.

2. Select the MIC_D or MIC_A.



OR



3.Upload to Arduino.

It will add this line on the loop.

```
KBOT.PRINT("MIC_D=");
KBOT.GET_DATA(KBOT.MIC_D_READ);
KBOT.PRINTLN("");
```

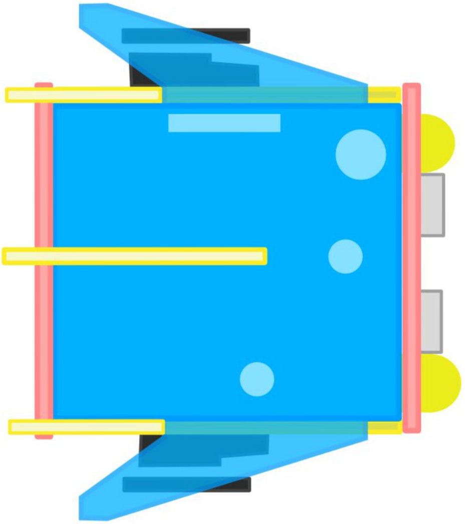
Digital Output
See what happens?

```
KBOT.PRINT("MIC_A=");
KBOT.GET_DATA(KBOT.MIC_A_READ);
KBOT.PRINTLN("");
```

Analog Read
See what happens?

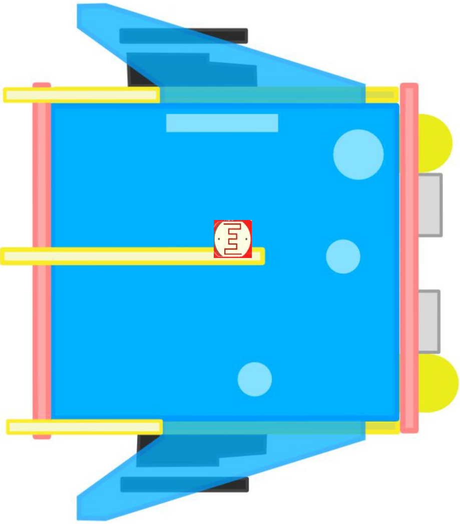
# 5.B Reading Sound Output ILLUSTRATION

Digital Output



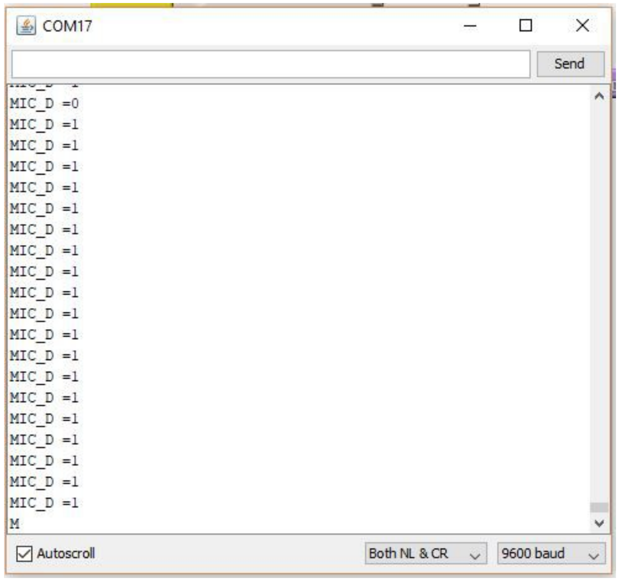
*Clap sounds

Analog Output



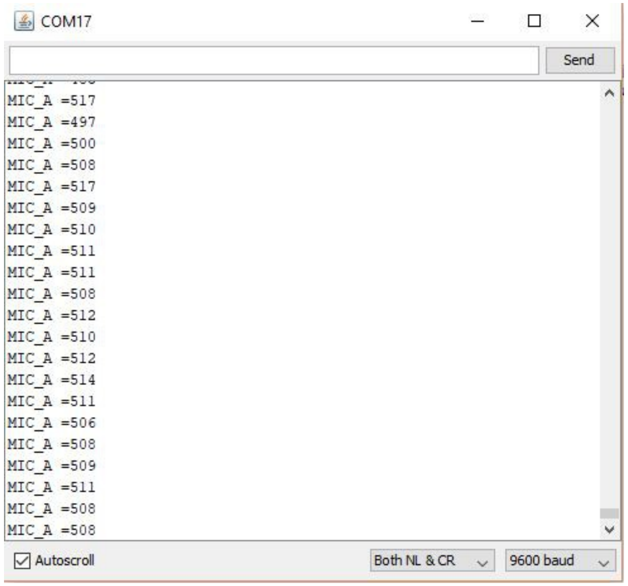
MIC = 1   No Sounds detected   MIC <=497

MIC = 0   Sounds detected   MIC >=517



```
MIC_D =0
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
MIC_D =1
M
```

☑ Autoscroll   Both NL & CR   9600 baud



```
MIC_A =517
MIC_A =497
MIC_A =500
MIC_A =508
MIC_A =517
MIC_A =509
MIC_A =510
MIC_A =511
MIC_A =511
MIC_A =508
MIC_A =512
MIC_A =510
MIC_A =512
MIC_A =514
MIC_A =511
MIC_A =506
MIC_A =508
MIC_A =509
MIC_A =511
MIC_A =508
MIC_A =508
```

☑ Autoscroll   Both NL & CR   9600 baud

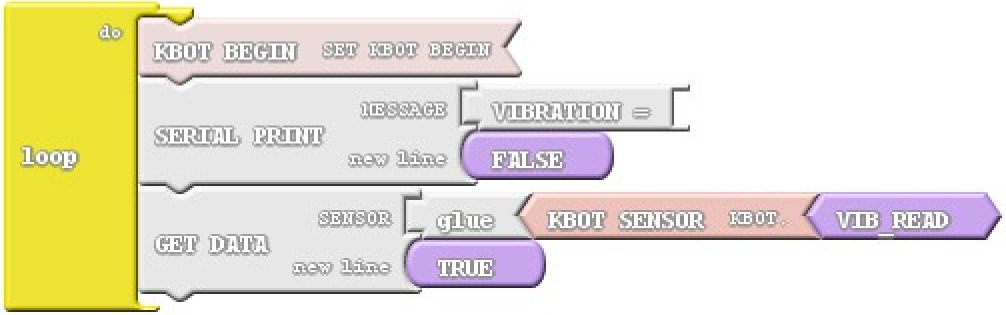
# 6. Vibration Sensor

# 6.A VIBRATION SENSOR BLOCKS

Select the Vibration sensor to read.

1.Click the message (shown below) change it to VIBRATION =.

2. Select the VIB_READ.



loop do
KBOT BEGIN SET KBOT BEGIN
SERIAL PRINT MESSAGE VIBRATION = new line FALSE
GET DATA SENSOR glue KBOT SENSOR KBOT. VIB_READ new line TRUE

3.Upload to Arduino.

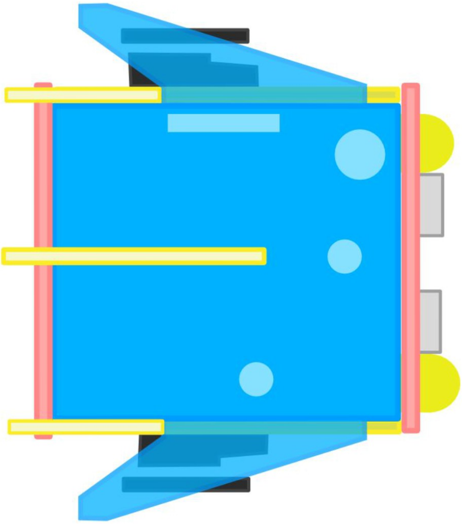
It will add this line on the loop.

```
KBOT.PRINT(“VIBRATION=”);
KBOT.GET_DATA(KBOT.VIB_READ);
KBOT.PRINTLN(“”);
```

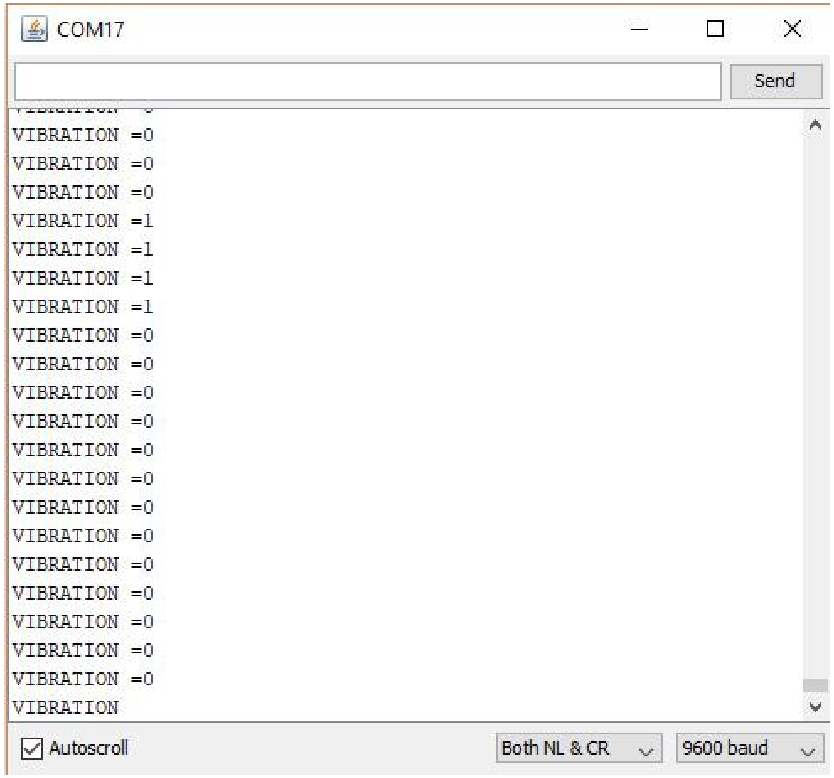
See what happens?

# 6.B Reading Vibration Output ILLUSTRATION

## Digital Output



* You need strong impact / Shake



```
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =1
VIBRATION =1
VIBRATION =1
VIBRATION =1
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION =0
VIBRATION
```

☑ Autoscroll   Both NL & CR   9600 baud

VIBRATION = 1, SHAKE DETECTED
VIBRATION = 0, NO DETECTED

# 6.C ALL SENSORS SYNTAX & BLOCKS

READING THE LDR,DISTANCE,SOUND,VIBRATION

//Setting the LDR sensor for reading

- KBOT.LDR_SENSE();
- KBOT.MEASURE_IN_CM();
- KBOT.MIC_DIGITAL();
- KBOT.MIC_ANALOG();
- KBOT.VIB_SENSE();
-

DISPLAY ALL THE SENSOR'S OUTPUT VALUE

//Getting the value and display it on the Serial Monitor

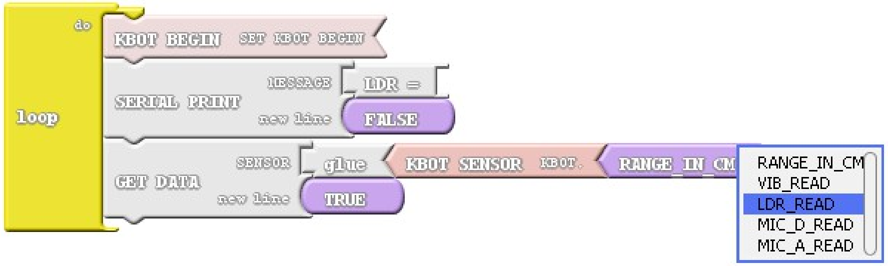
- KBOT.GET_DATA(K12BOT.LDR_READ);
- KBOT.GET_DATA(K12BOT.RANGE_IN_CM);
- KBOT.GET_DATA(K12BOT.MIC_D_READ);
- KBOT.GET_DATA(K12BOT.MIC_A_READ);
- KBOT.GET_DATA(K12BOT.VIB_READ);

EXAMPLE: //inside the loop

```
KBOT.MEASURE_IN_CM();
KBOT.GET_DATA(K12BOT.RANGE_IN_CM); //Display the value
KBOT.PRINTLN(“ ”);
```

Select the directions.

1.Click the drop down arrow (shown below).



Where:

- LDR_READ
- RANGE_IN_CM
- MIC_D_READ
- MIC_A_READ
- VIB_READ

# Save your work.

# 6.D UPLOADING SENSORS.ABP

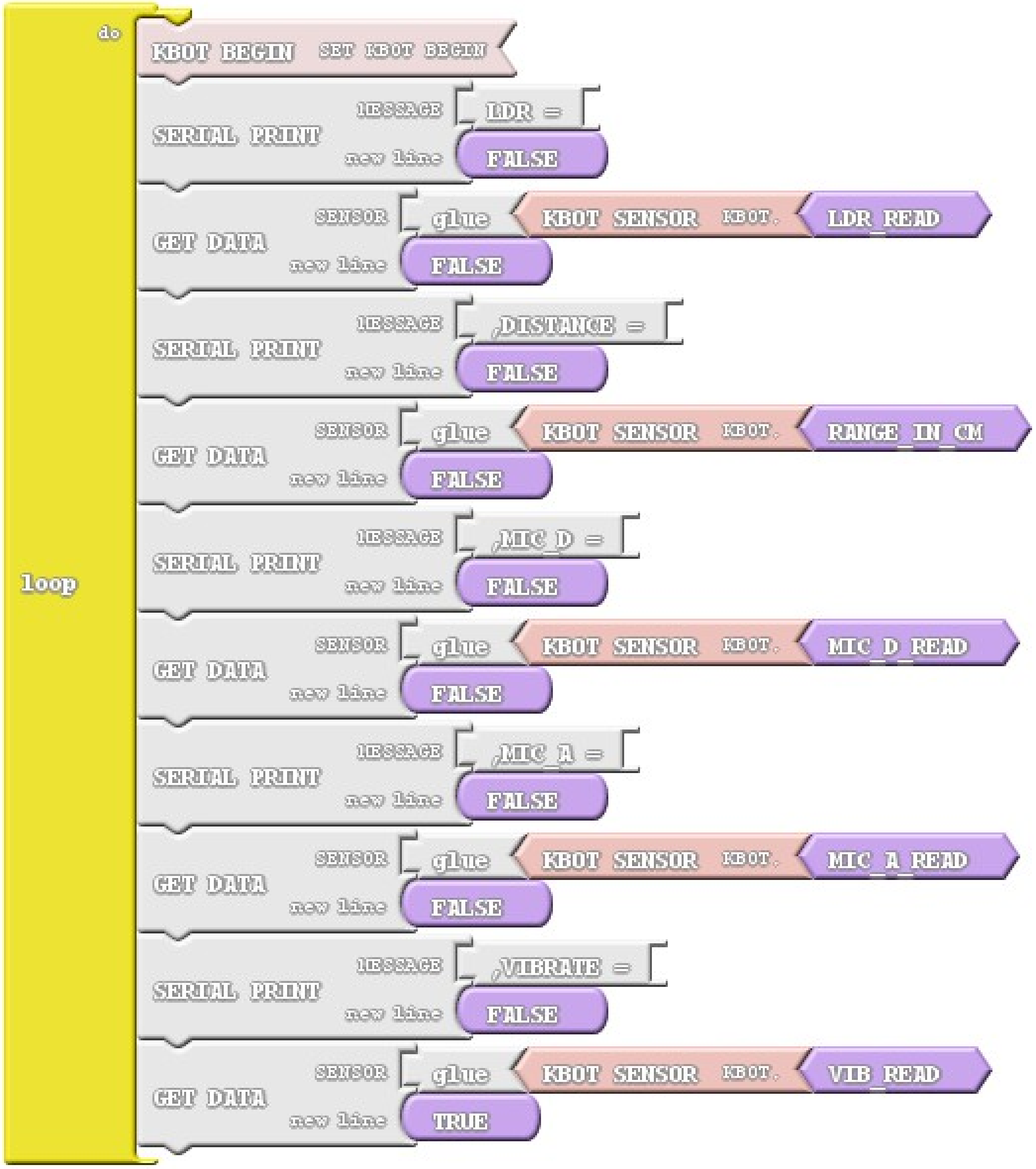
On the Ardublock.
1. Click OPEN.

Select the SENSORS.abp

Displaying data in Serial Monitor.

- LDR read
- Range in centimeter
- Mic Digital output
- Mic Analog output
- Vibration read

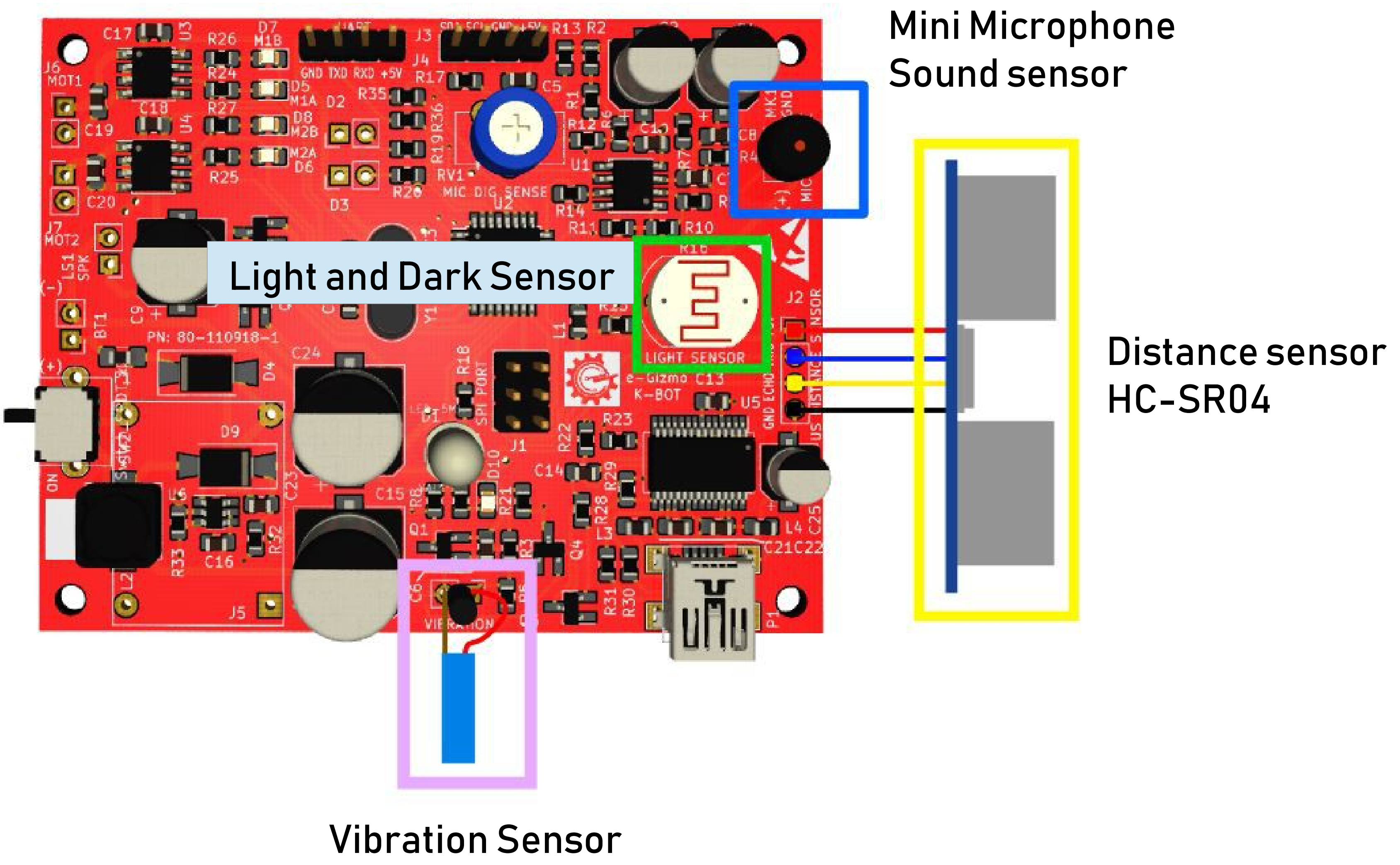
2. Upload to Arduino.



```
loop
do
  KBOT BEGIN   SET KBOT BEGIN
  SERIAL PRINT   MESSAGE  LDR =
                 new line FALSE
  GET DATA   SENSOR  glue  KBOT SENSOR  KBOT.  LDR_READ
             new line FALSE
  SERIAL PRINT   MESSAGE  ,DISTANCE =
                 new line FALSE
  GET DATA   SENSOR  glue  KBOT SENSOR  KBOT.  RANGE_IN_CM
             new line FALSE
  SERIAL PRINT   MESSAGE  ,MIC_D =
                 new line FALSE
  GET DATA   SENSOR  glue  KBOT SENSOR  KBOT.  MIC_D_READ
             new line FALSE
  SERIAL PRINT   MESSAGE  ,MIC_A =
                 new line FALSE
  GET DATA   SENSOR  glue  KBOT SENSOR  KBOT.  MIC_A_READ
             new line FALSE
  SERIAL PRINT   MESSAGE  ,VIBRATE =
                 new line FALSE
  GET DATA   SENSOR  glue  KBOT SENSOR  KBOT.  VIB_READ
             new line TRUE
```

# 6.E Sensor Locations

# SENSORS DIAGRAM



Mini Microphone
Sound sensor

Light and Dark Sensor

Distance sensor
HC-SR04

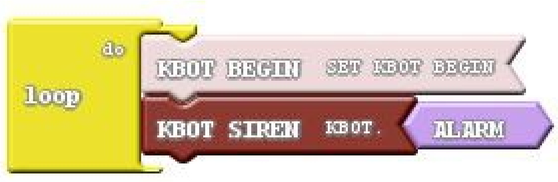
Vibration Sensor

# 7. Siren/Alarm

# 7.A SIREN BLOCKS

Select the Alarm.

1. Select the SIREN and set the alarm.



3.Upload to Arduino.

It will add this line on the loop.

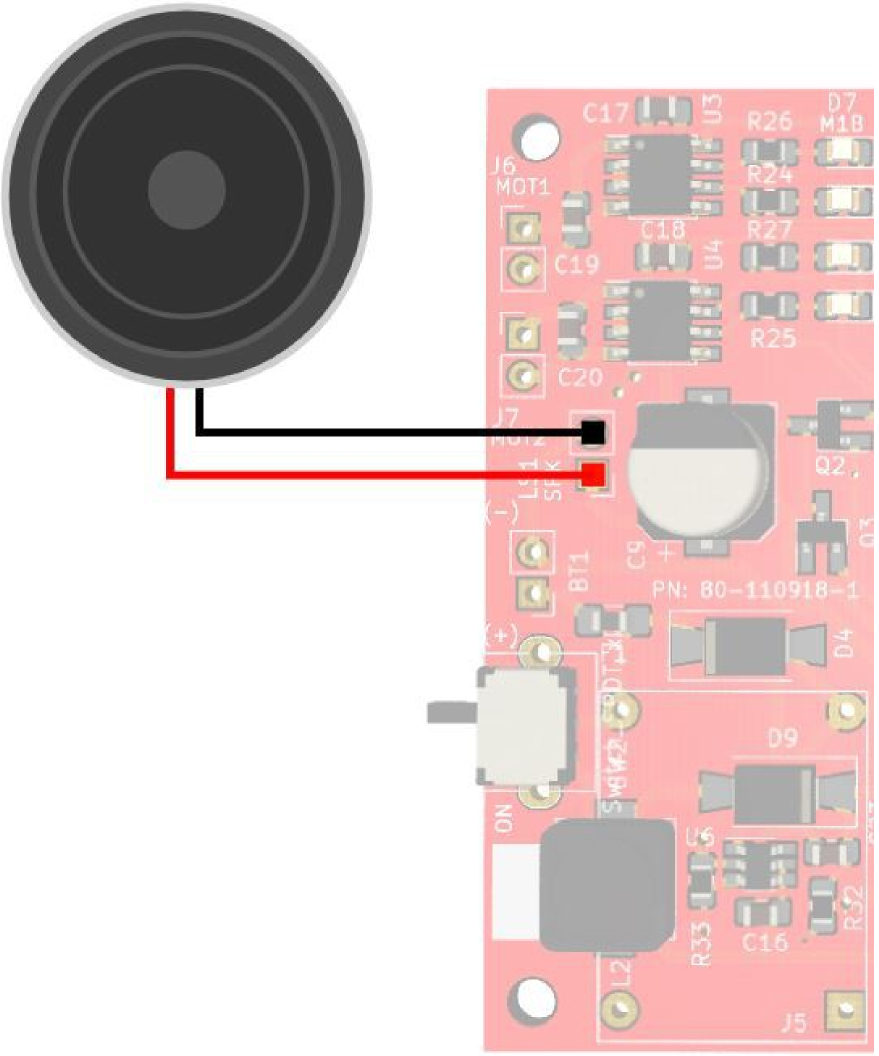
```
KBOT.ALARM():
```

See what happens?

# 7.B SPEAKER CONNECTION

## SIREN/ALARM

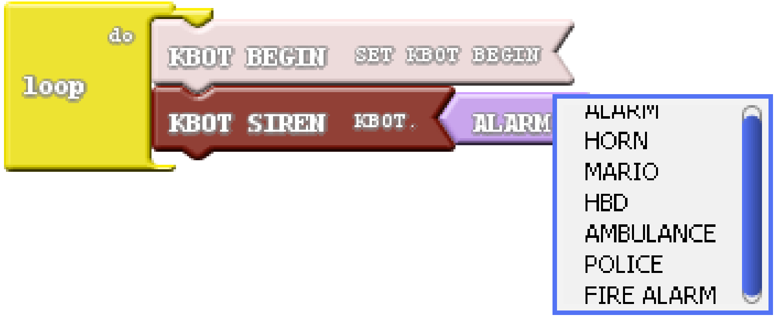
SPEAKER



# 7.c SIREN/ALARM SYNTAX & BLOCKS

// SYNTAX

- KBOT.MARIO();
- KBOT.HBD();
- KBOT.ALARM();
- KBOT.HORN();
- KBOT.AMBULANCE();
- KBOT.POLICE();
- KBOT.FIRE();

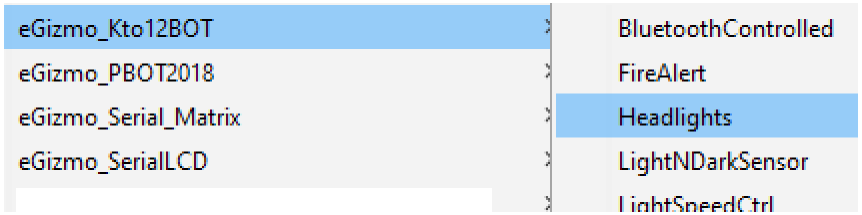


# VII. eGizmo_Kto12BOT Library “Examples“

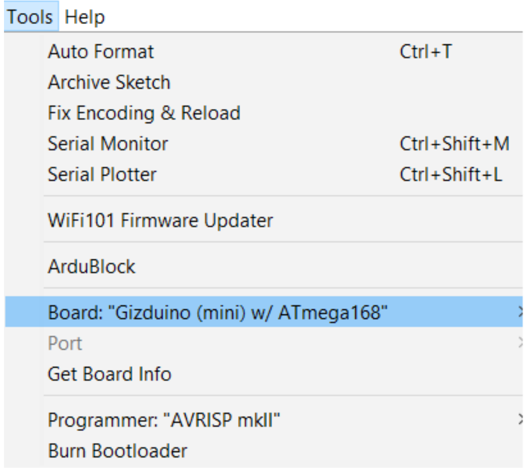
# UPLOADING HEADLIGHTS

On the Arduino IDE.

1. Headlights.ino codes
   Go to File>Examples>eGizmo_Kto12BOT>Headlights



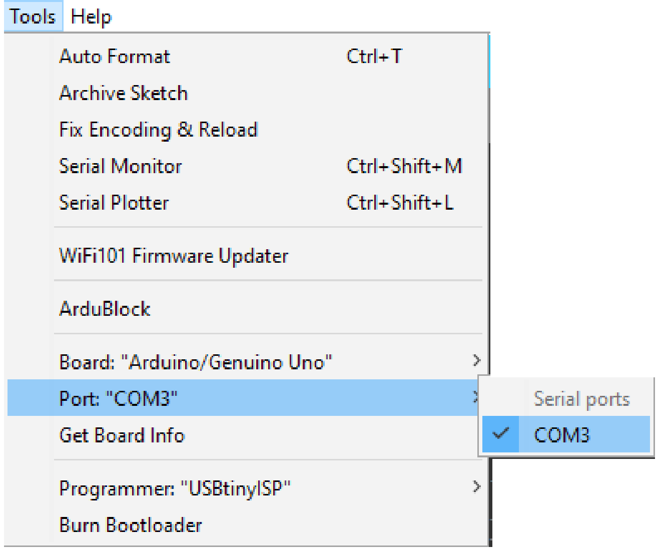
2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168



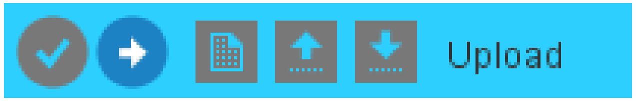
See what happens?

3. Port select
   Go to Tools>Port>COM#

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



4. Click Upload
   Wait until its Done Uploading

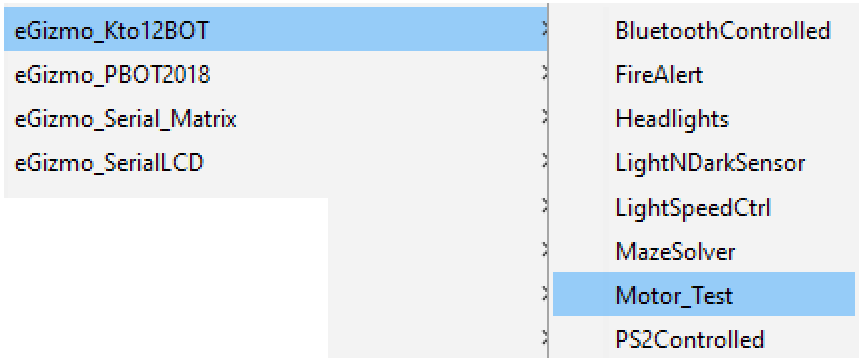


# UPLOADING MOTOR TEST

On the Arduino IDE.

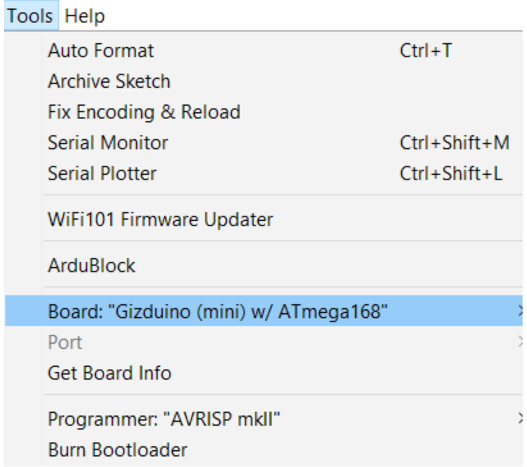
1. Motor_Test.ino codes

   Go to File>Examples>eGizmo_Kto12BOT>Motor_Test



2. Board select

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



Board: "Gizduino (mini) w/ ATmega168"

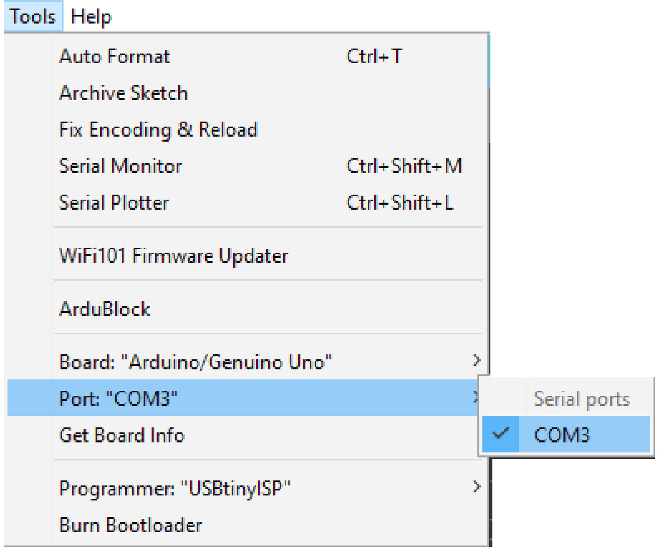
Programmer: "AVRISP mkII"

See what happens?

3. Port select

   Go to Tools>Port>COM#

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



Board: "Arduino/Genuino Uno"

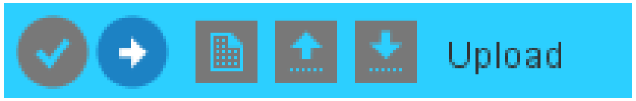
Port: "COM3"

✓ COM3

Programmer: "USBtinyISP"

4. Click Upload

   Wait until its Done Uploading

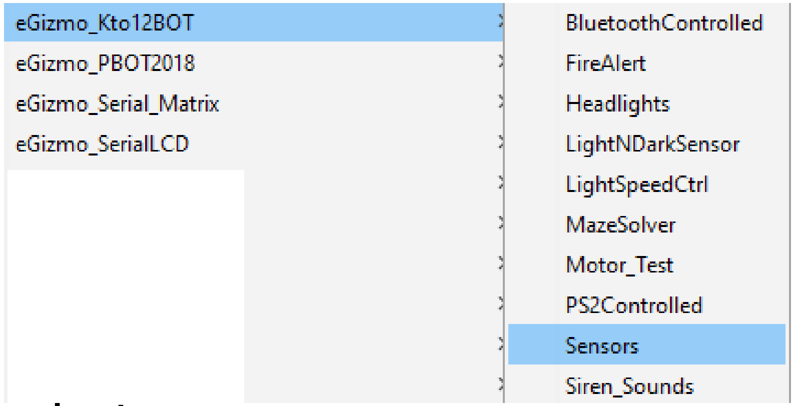


# UPLOADING SENSORS

On the Arduino IDE.

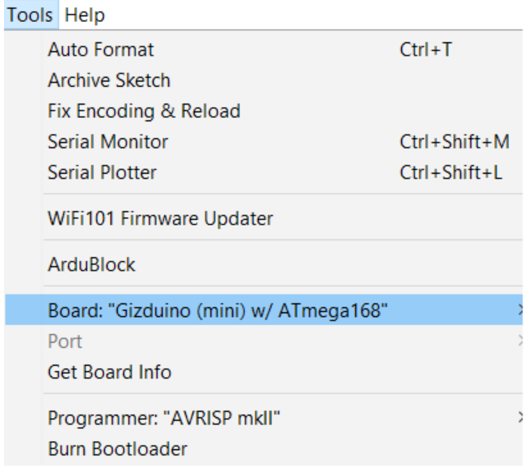
1. Sensors.ino codes

   Go to File>Examples>eGizmo_Kto12BOT>Sensors



2. Board select

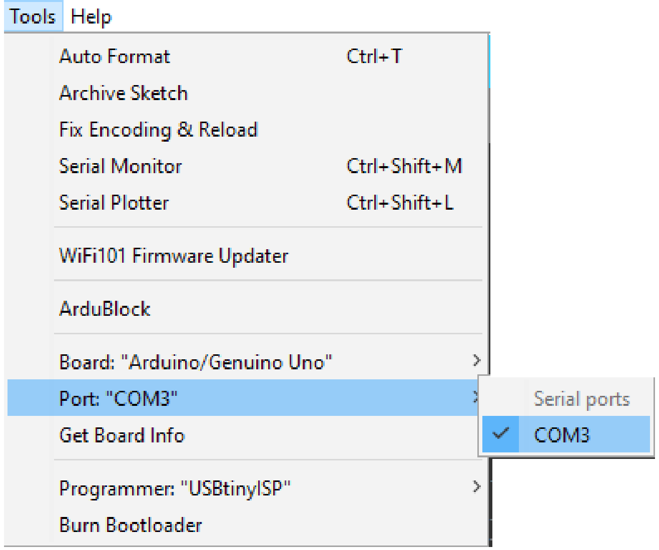
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168



3. Port select

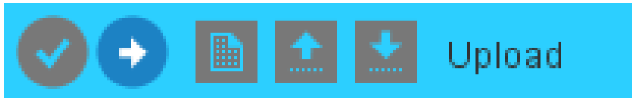
   Go to Tools>Port>COM#

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



4. Click Upload

   Wait until its Done Uploading

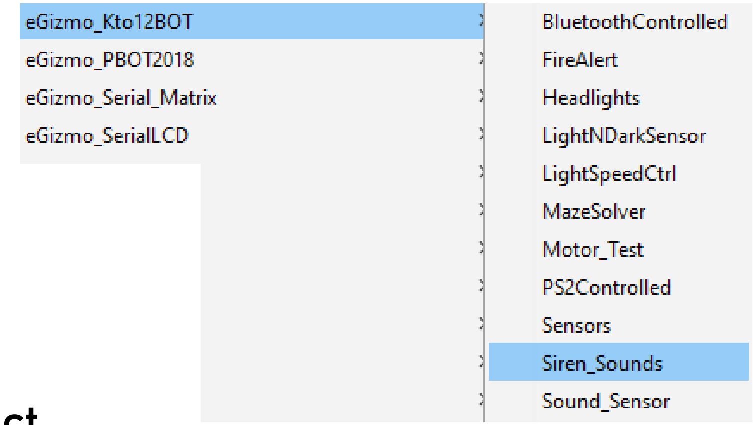


See what happens?

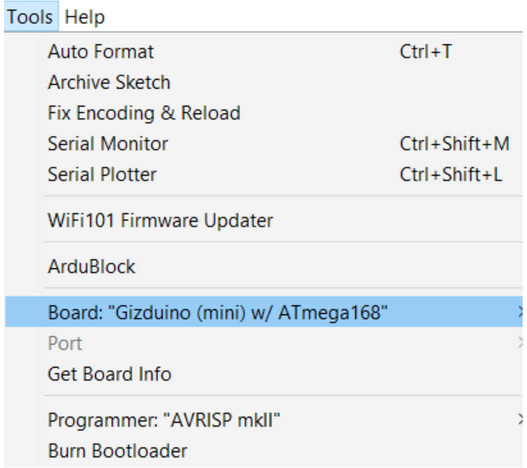
# UPLOADING SIREN_SOUNDS.INO

On the Arduino IDE.

1. Siren_Sounds.ino codes
   Go to File>Examples>eGizmo_Kto12BOT>Siren_Sounds

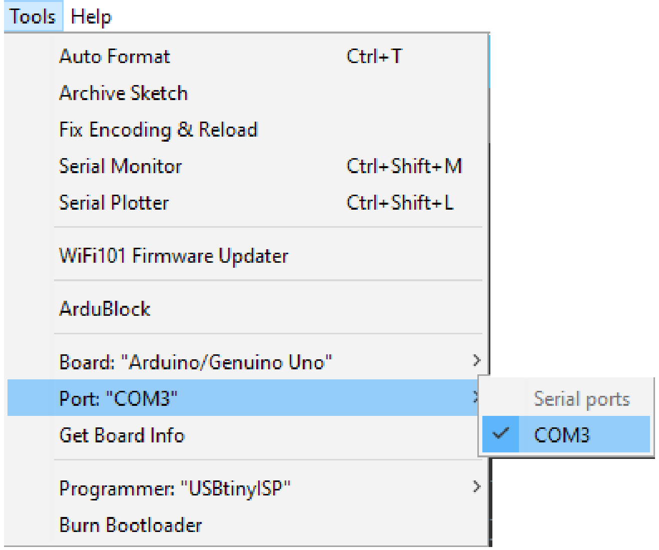


2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168

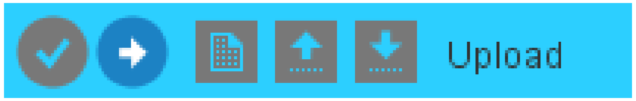


3. Port select
   Go to Tools>Port>COM#

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



4. Click Upload
   Wait until its Done Uploading



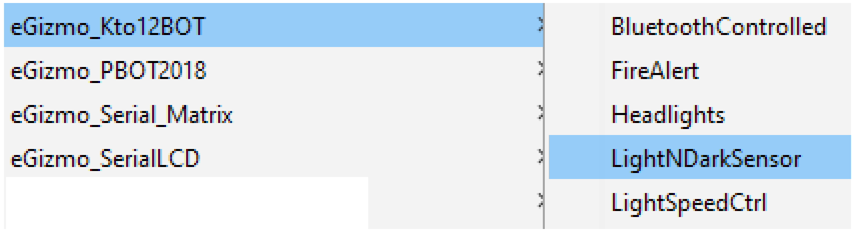
See what happens?

# Light and Dark Sensor Applications

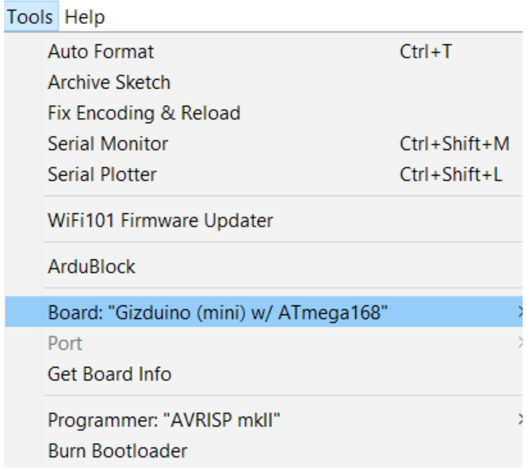
# A. UPLOADING LIGHTNDARKSENSOR WITH HEADLIGHTS FUNCTION

On the Arduino IDE.

1. LightNDarkSensor.ino codes
   Go to File>Examples>eGizmo_Kto12BOT>
   LightNDarkSensor



2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168

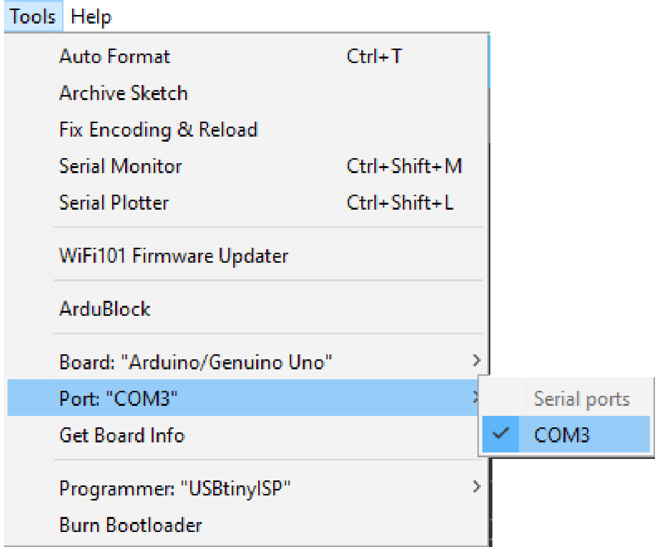


Board: "Gizduino (mini) w/ ATmega168"

Programmer: "AVRISP mkII"

See what happens?

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

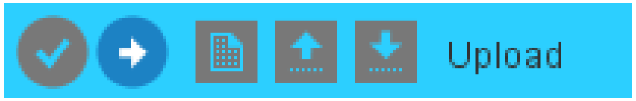


Board: "Arduino/Genuino Uno"

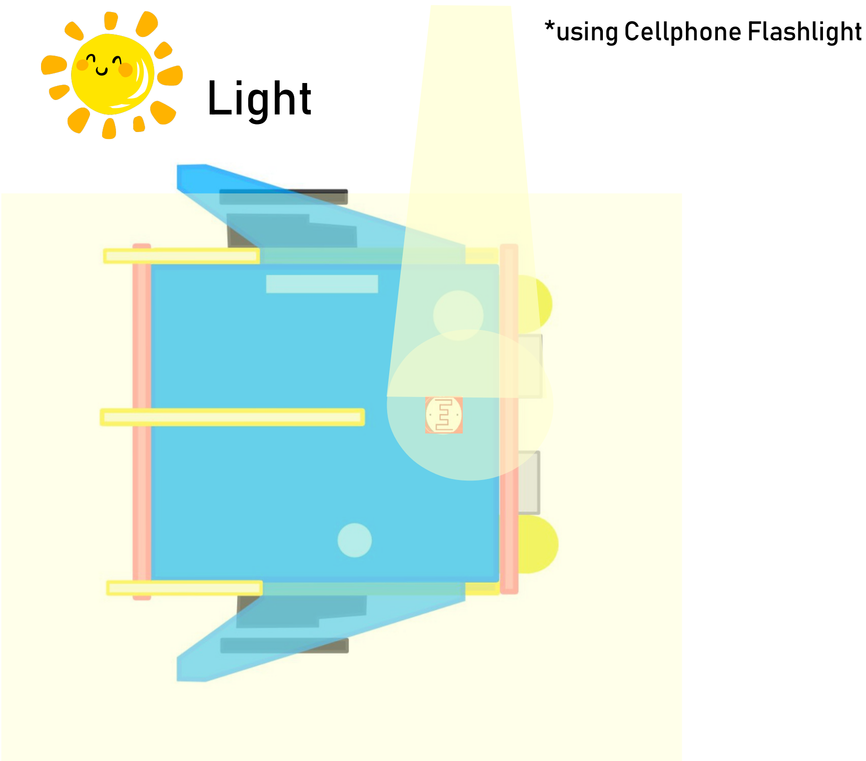
Port: "COM3"

Programmer: "USBtinyISP"

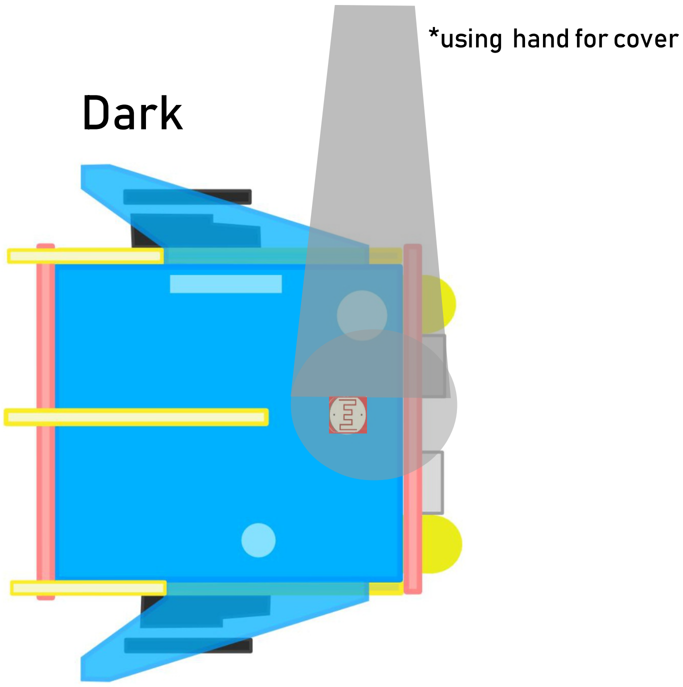
4. Click Upload
   Wait until its Done Uploading



# B. LIGHTNDARK SENSOR ILLUSTRATION



```
//IF SENSE LIGHT, THE HEADLIGHTS WILL TURNED ON
if(KBOT.LDR_READ < 980 && KBOT.LDR_READ > 600){
  KBOT.HEADLIGHT_LEFT(1);
  KBOT.HEADLIGHT_RIGHT(1);
  KBOT.RGB(1);
}
```

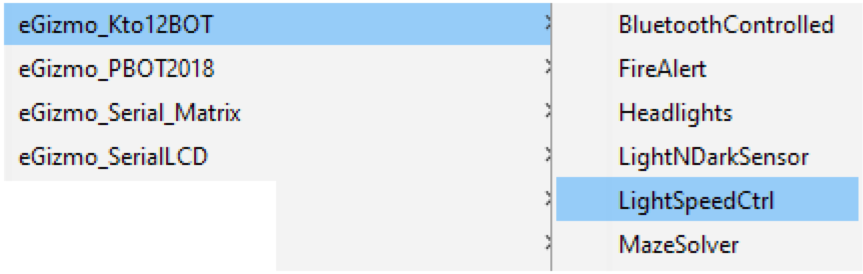


```
//IF SENSE DARK, HEADLIGHTS WILL TURNED OFF
if(KBOT.LDR_READ < 500 && KBOT.LDR_READ > 450){
  KBOT.HEADLIGHT_LEFT(0);
  KBOT.HEADLIGHT_RIGHT(0);
  KBOT.RGB(0);
}
```

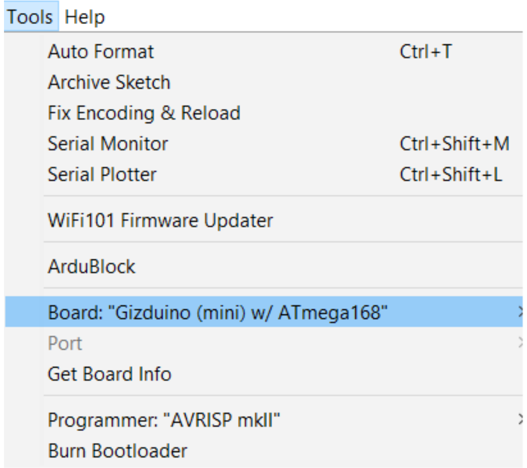
# C. UPLOADING LIGHTSPEEDCTRL WITH MOTOR FUNCTION

On the Arduino IDE.

1. LightSpeedCtrl.ino codes
   Go to File>Examples>eGizmo_Kto12BOT> LightSpeedCtrl



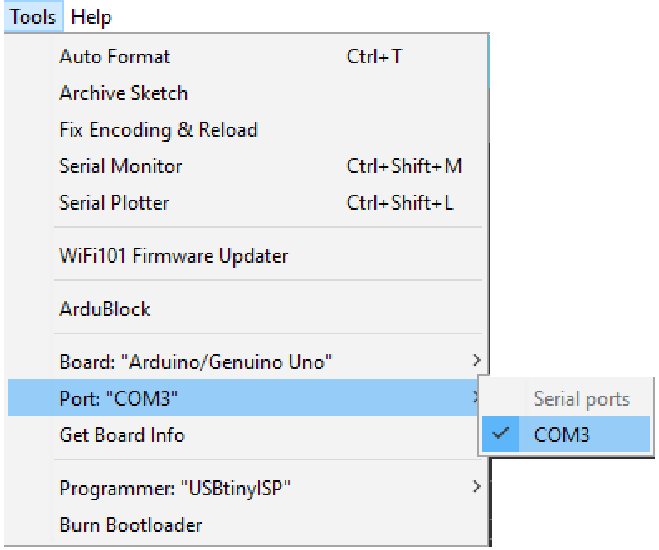
2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168



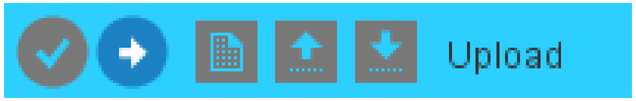
See what happens?

3. Port select
   Go to Tools>Port>COM#

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



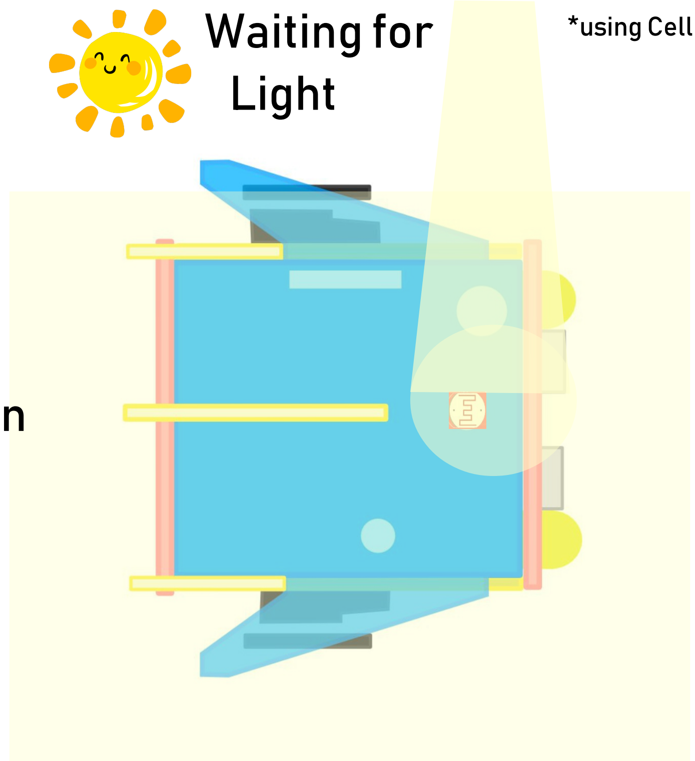
4. Click Upload
   Wait until its Done Uploading



# D. LightSpeed controls Motor illustration

The range of analog
Is from 0 to 1023.

To move it forward.
You need a range between
400 to 600



Waiting for
Light

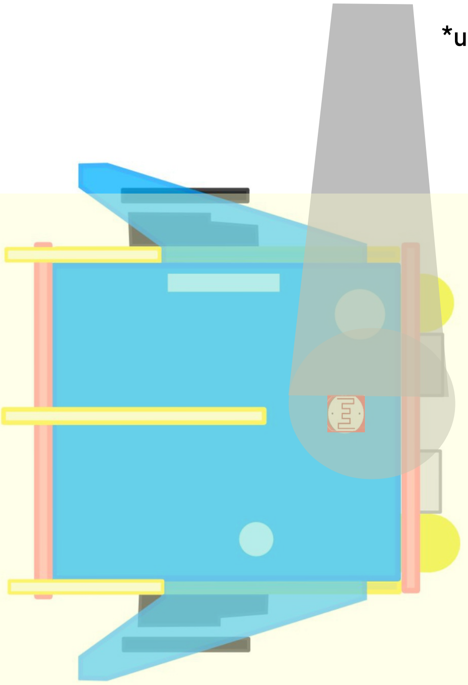
*using Cellphone Flashlight

```
 //IF SENSES THE LIGHT WITHIN THIS RANGE THE HEADLIGHTS WILL TURNED ON AND MOVE FORWARD
if(KBOT.LDR_READ < 400 && KBOT.LDR_READ > 60){
   /*MOVE FORWARD*/
   KBOT.FORWARD(255);
   KBOT.HEADLIGHT_LEFT(1);
   KBOT.HEADLIGHT_RIGHT(1);
   KBOT.RGB(1);

 }
```

# E. LightSpeed controls Motor illustration

If dark or normal lights

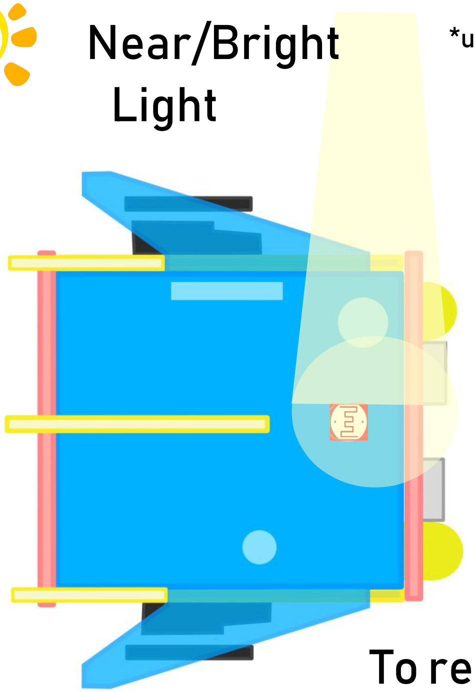
*using hand for cover



To Stop.
You need a range between
250 to 600

Near/Bright
Light

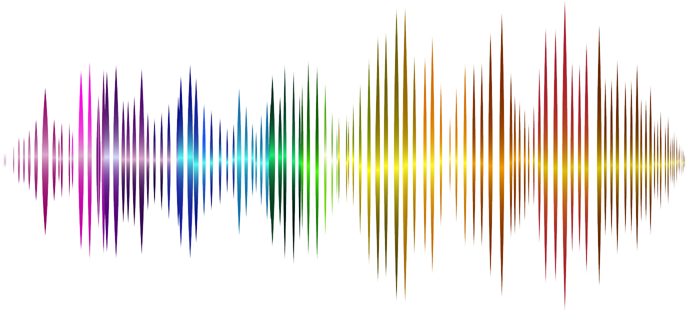
*using Cellphone Flashlight



To reverse.
You need a range
Between 20 to 100

```
//STAND BY WAITING FOR LIGHT DETECTION
if(KBOT.LDR_READ < 600 && KBOT.LDR_READ > 250){
  KBOT.STOP(0);
  KBOT.HEADLIGHT_LEFT(0);
  KBOT.HEADLIGHT_RIGHT(0);
  KBOT.RGB(1);
}
```

```
//IF LIGHT IS SO INTENSE THE MOTOR REVERSE
if(KBOT.LDR_READ < 100 && KBOT.LDR_READ > 20){
  KBOT.REVERSE(255);
  KBOT.HEADLIGHT_LEFT(0);
  KBOT.HEADLIGHT_RIGHT(0);
  KBOT.RGB(0);
}
```



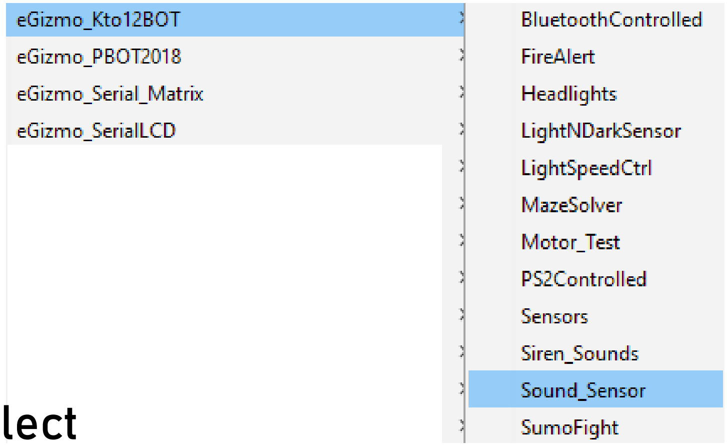
# IX.Sound Sensor Example

# A. UPLOADING SOUND SENSOR WITH MOTOR FUNCTIONS

On the Arduino IDE.

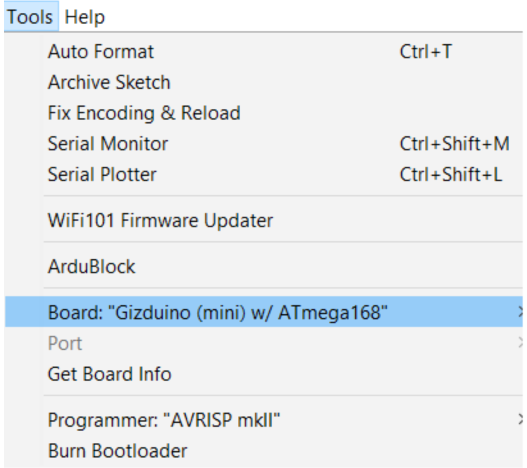
1. Sound_Sensor.ino codes

   Go to File>Examples>eGizmo_Kto12BOT>Sound_Sensor



2. Board select

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

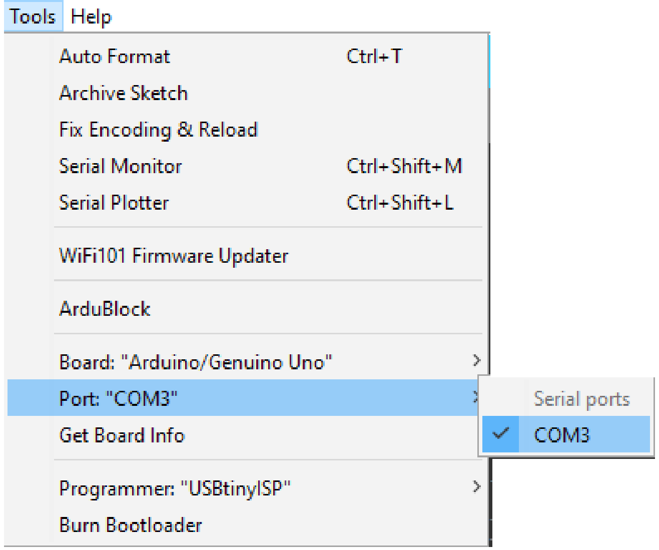


See what happens?

3. Port select

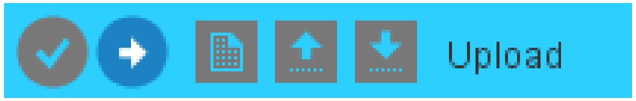
   Go to Tools>Port>COM#

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



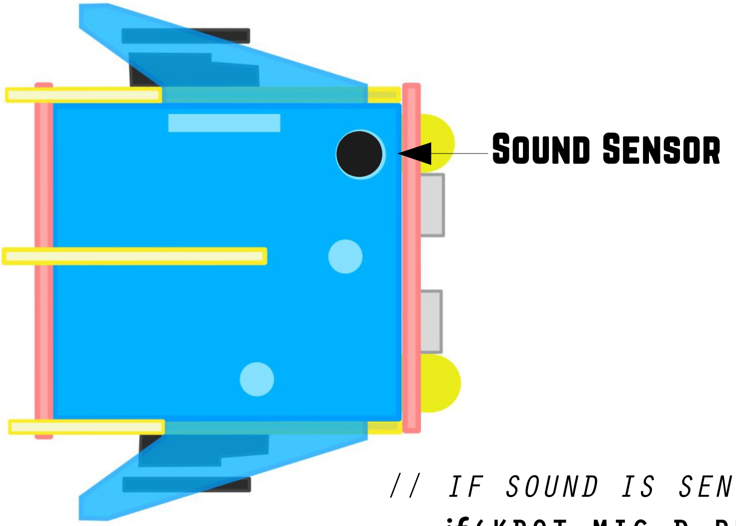
4. Click Upload

   Wait until its Done Uploading



# B. SOUND SENSOR ILLUSTRATION

Digital Output

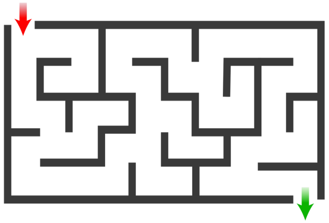


MIC = 1 Sound detected
MIC = 0 No Sounds detected

*Clap sounds

If you Clap once, the output must be Equal to 1.

```
// IF SOUND IS SENSED, THE HEADLIGHT WILL TURN ONS AND ROBOT WILL MOVE FORWARD
    if(KBOT.MIC_D_READ == 1){
       KBOT.FORWARD(255);
       KBOT.HEADLIGHT_LEFT(1);
       KBOT.HEADLIGHT_RIGHT(1);
       delay(2000);
    }
    else
    {
// HEADLIGHTS OFF AND MOTOR STOPS
       KBOT.HEADLIGHT_LEFT(0);
       KBOT.HEADLIGHT_RIGHT(0);
       KBOT.STOP(0);
    }
```

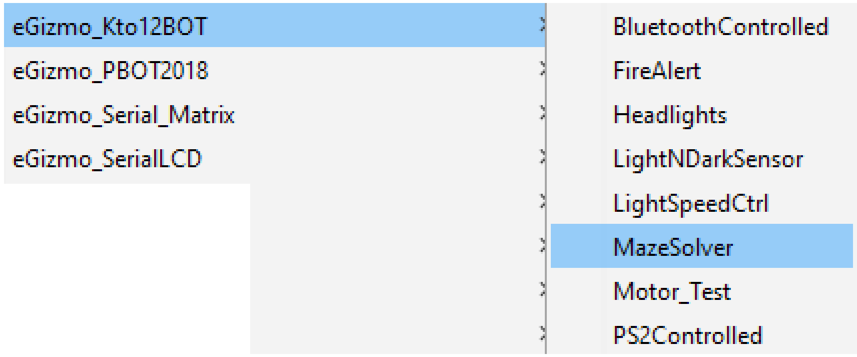


# X. Maze Solver
# or
# Collision Avoidance

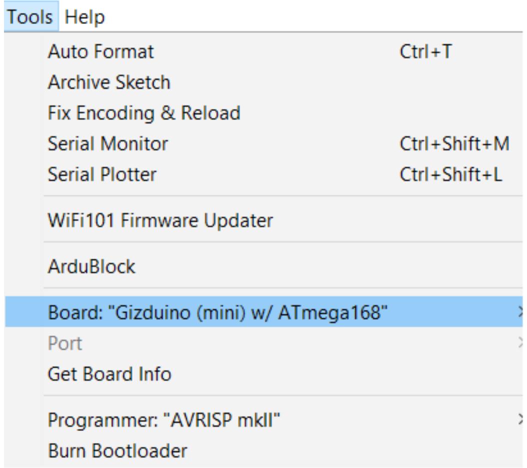
# UPLOADING MAZE SOLVER

On the Arduino IDE.

1. MazeSolver.ino codes
   Go to File>Examples>eGizmo_Kto12BOT>MazeSolver

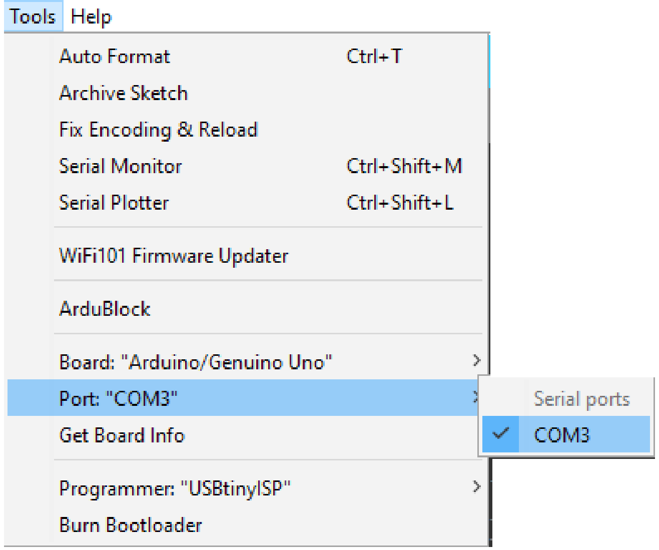


2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168

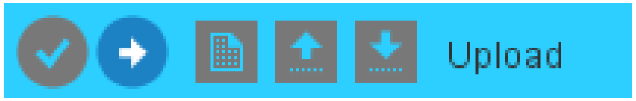


3. Port select
   Go to Tools>Port>COM#

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

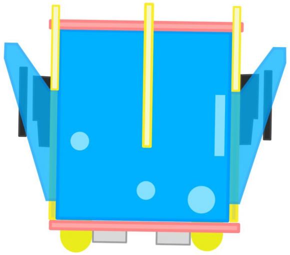


4. Click Upload
   Wait until its Done Uploading

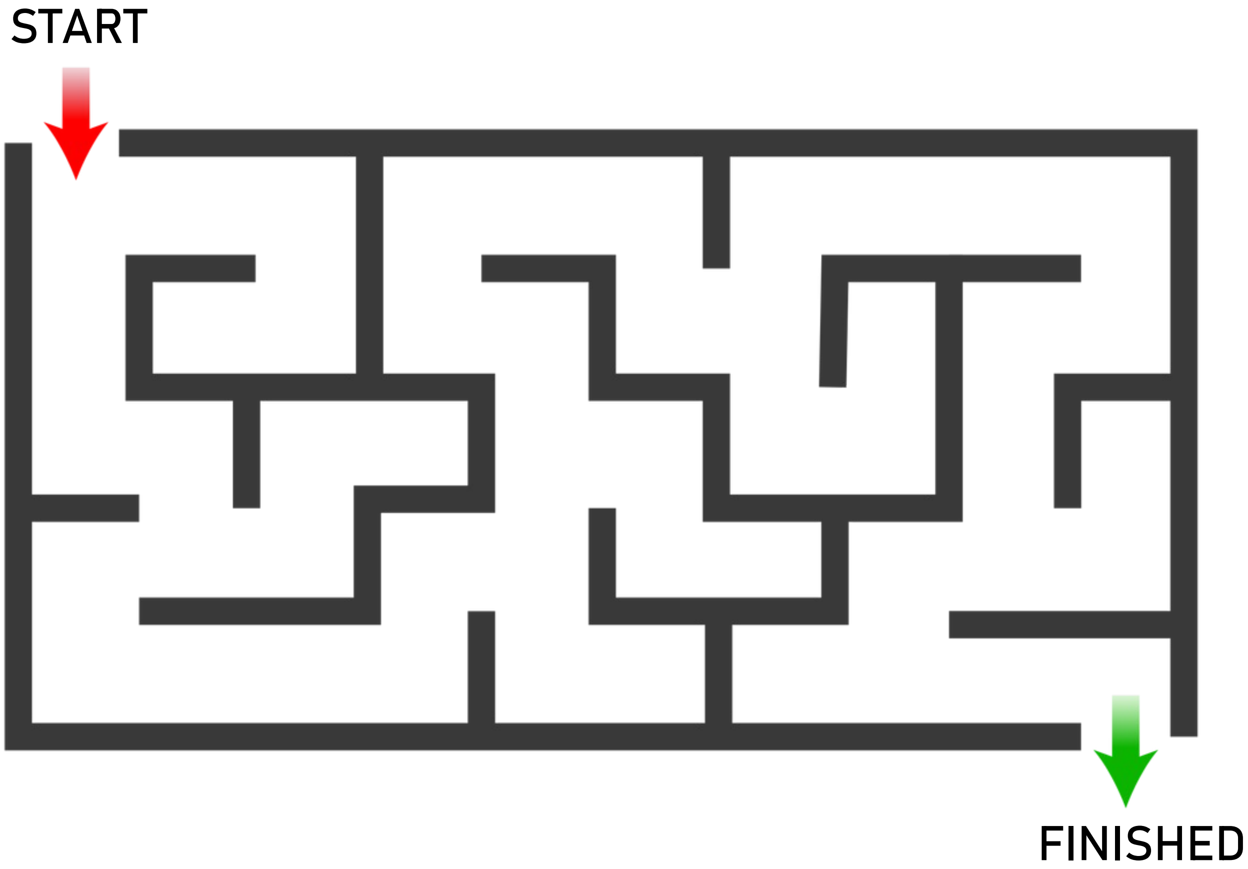


See what happens?

# B. Sample Maze Track

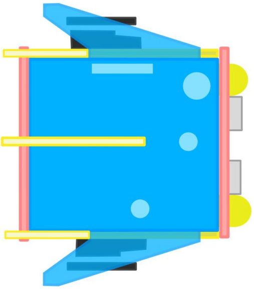


SAMPLE MAZE TRACK

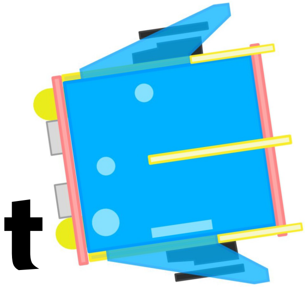


# C. MAZE Sample Codes

```
// IF NO OBSTRUCTION, MOVE FORWARD
 if(KBOT.RANGE_IN_CM > DETECT_RANGE || KBOT.RANGE_IN_CM == 0){
   KBOT.FORWARD(SPEED);
   KBOT.HEADLIGHT_LEFT(1);
   KBOT.HEADLIGHT_RIGHT(1);
   KBOT.RGB(1);

 }
 // IF THERE'S A WALL, REVERSE THEN TURN LEFT AND LOOK FOR STRAIGHT PATH
 if((KBOT.RANGE_IN_CM <= DETECT_RANGE && KBOT.RANGE_IN_CM !=0)){
   KBOT.HEADLIGHT_LEFT(1);
   KBOT.HEADLIGHT_RIGHT(0);
   KBOT.EXTREMELEFT(SPEED);
   delay(1000);

 }
```



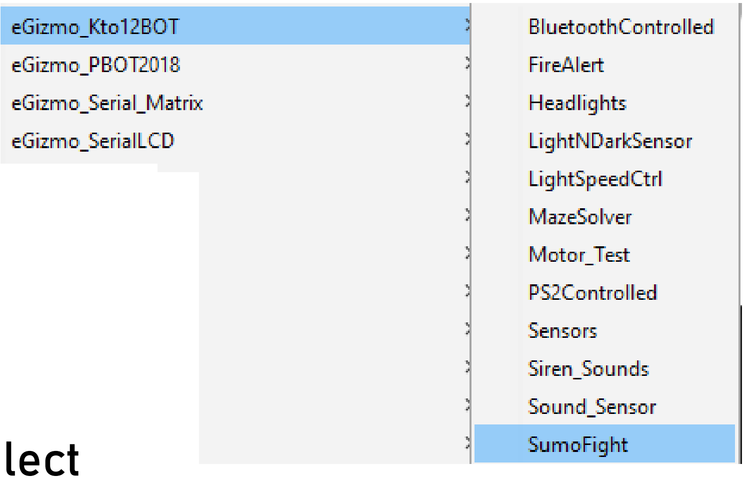
# XI.Sumo Fight



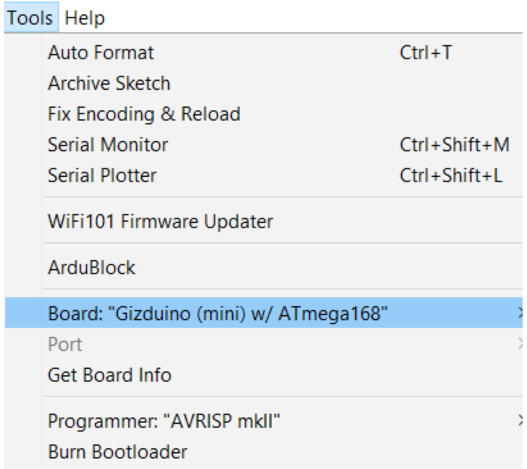
# A. UPLOADING SUMO FIGHT

On the Arduino IDE.

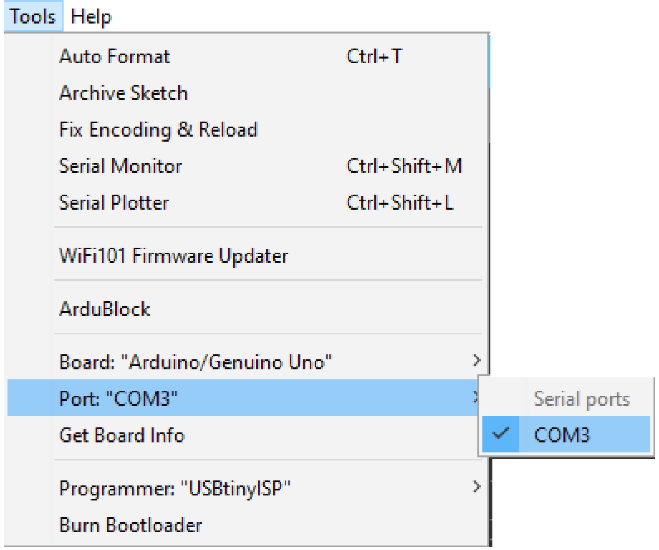
1. SumoFight.ino codes
   Go to File>Examples>eGizmo_Kto12BOT>SumoFight



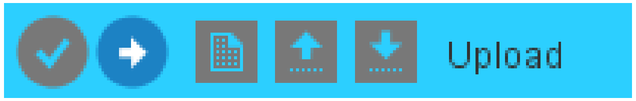
2. Board select
   Go to Tools>Boards>gizDuino (mini) w/ Atmega168



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



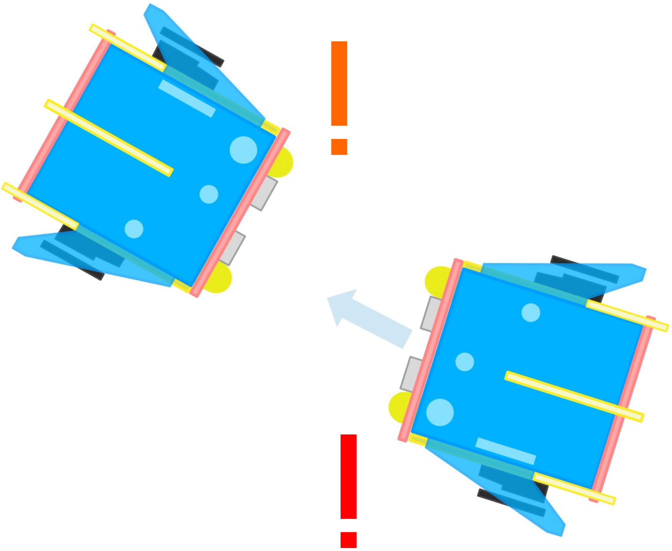
4. Click Upload
   Wait until its Done Uploading



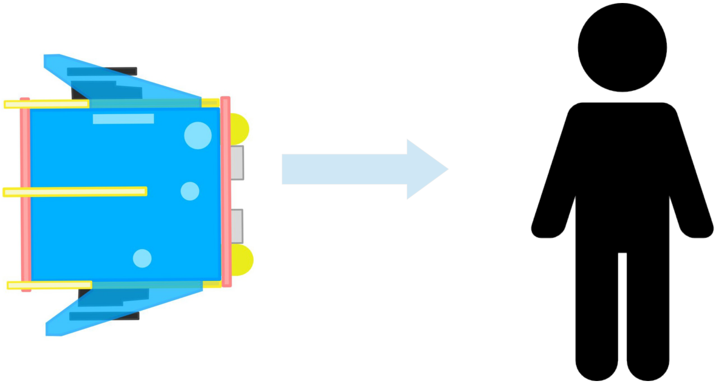
See what happens?

# B. SUMOFIGHT OR HUMAN FOLLOWER

If the Robot detects the intruder
It will move forward and bump it.



Human Follower



# C. SUMO SAMPLE CODES

```
// IF OPPENENTS DETECTED, BUMP ON FULL SPEED FORWARD
 while(KBOT.RANGE_IN_CM < 6 && KBOT.RANGE_IN_CM > 0){
   KBOT.FORWARD(FULL_SPEED);
   KBOT.HEADLIGHT_LEFT(1);
   KBOT.HEADLIGHT_RIGHT(1);
   DELAY(1000);
   break;
 }
 // IF THERES NO OPPENENTS, IN NORMAL SPEED MODE
 if(KBOT.RANGE_IN_CM > 6){
   KBOT.EXTREMERIGHT(NORMAL_SPEED);
   KBOT.HEADLIGHT_LEFT(0);
   KBOT.HEADLIGHT_RIGHT(1);
   KBOT.RGB(1);
   delay(100);
 }
```

e-Gizmo
MECHATRONIX CENTRAL

# For more info:

- Website: www.e-gizmo.net
- Egizmo Tech blog:www.e-gizmo.com/wp
- Facebook: eGizmoMechatronix
- Youtube Channel: e-Gizmo Mechatronix Central