



WHY CHOOSE US? ([HTTP://JSUMO.COM/WHY-CHOOSE-US/](http://jsumo.com/why-choose-us/)) | SHIPMENT AND INFORMATION ([HTTP://JSUMO.COM/SHIPMENT-AND-INFORMATION/](http://jsumo.com/shipment-and-information/)) | CART ([HTTP://JSUMO.COM/CART](http://jsumo.com/cart)) | CHECKOUT ([HTTP://JS](http://js))

JSUMO
ULTIMATE PARTS AND
ROBOT SUMO BLOG
(<http://jsumo.com/>)



For robot geeks! | CONTACT US (<http://jsumo.com/contact>)

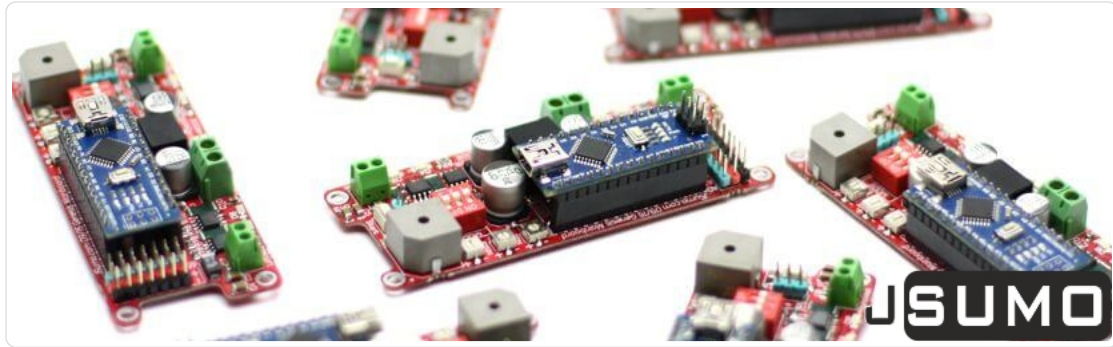


JSumo (<http://jsumo.com>) » Blog! (<http://jsumo.com/blog/>) » **Minisumo Tutorial for Genesis Arduino Mainboard**

Minisumo Tutorial for Genesis Arduino Mainboard

06

Dec



Minisumo Tutorial for Genesis Arduino Mainboard

By Firat Dede (<http://jsumo.com/author/firat-dede/>)

General Robotics (<http://jsumo.com/category/general-robotics/>), Mini Sumo Robot (<http://jsumo.com/category/mini-sumo-robot/>), Robot Parts (<http://jsumo.com/category/robot-parts/>)

0 Comments (<http://jsumo.com/minisumo-tutorial-genesis-arduino-mainboard/#respond>)

In that tutorial, we will look to Arduino sketches for Genesis based mini sumo robot.

First we should look the **Genesis's** (<http://jsumo.com/shop/genesis-arduino-robot-controller-board/>) connection table. That table determines any type of robot's program.

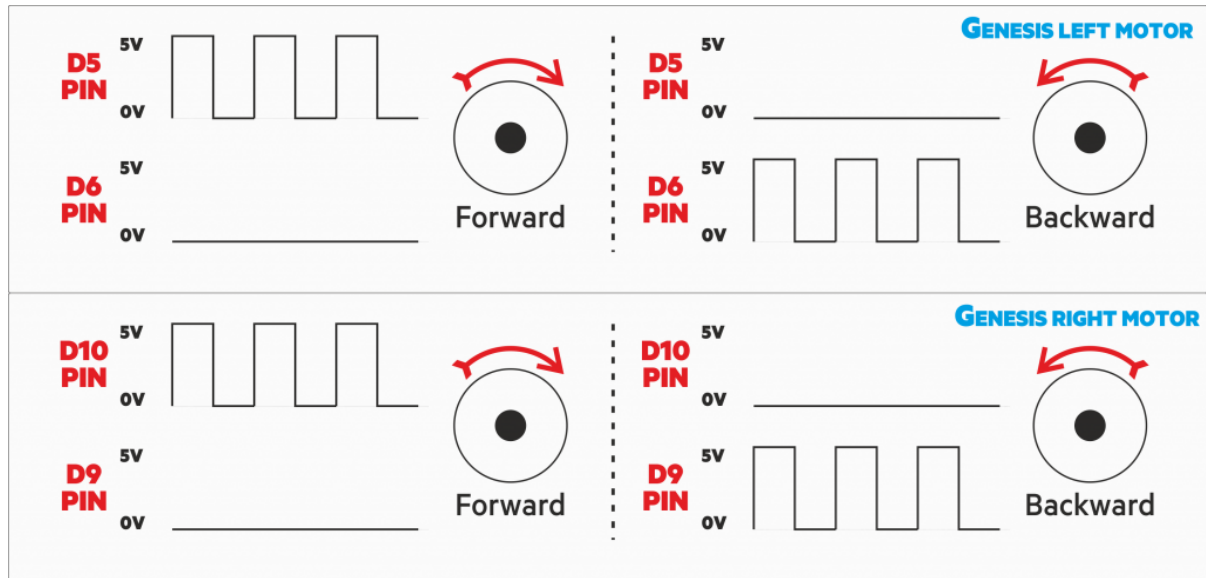
Arduino Pins	Purpose	Arduino Pins	Purpose
D0	Empty	D11	Empty
D1	Empty	D12	Mode Led
D2	Empty	D13	Arduino Nano Built-in Led, Buzzer
D3	Button Input, Start Module Input	A0	Empty
D4	Empty	A1	Empty
D5	Left PWM Channel 1	A2	Mode Switch 3 Input
D6	Left PWM Channel 2	A3	Mode Switch 2 Input
D7	Empty	A4	Mode Switch 1 Input
D8	Empty	A5	Edge Turn A5 Set Trimpot

D9	Right PWM Channel 1	A6	Turn A6 Set Trimpot
D10	Right PWM Channel 2	A7	Speed A7 Set Trimpot

From that table and that product page (Click here for Genesis Page (<http://jsumo.com/shop/genesis-arduino-robot-controller-board/>)) we are understanding motor control pins are **D5, D6 – D9, D10**. Two indicator pins for giving signals to the outside is D12 and D13 (Led & Buzzer). So I will write first motor control routine based on that pins.

Motor Control with Genesis

Here is important: For driving Dc motors each motor driver uses 2 PWM channel. That means if I want to write forward the left Motor. I will give High to D6 (Or giving PWM signal to D6) and making low signal the D5 Pin. But if I want to drive backward I will give a high signal (Or PWM signal) to D5 and I will send a low signal to D6 pin. At below you will find visual of What I am meaning.



(<http://jsumo.m5industry.netdna-cdn.com/wp-content/uploads/2015/12/signal-pwm.png>)

And What is PWM? **Please look that wiki page.** (https://en.wikipedia.org/wiki/Pulse-width_modulation) But shortly we are sending a special square signal to Motor driver chips with fast On/Off (490 times in second). And we change that signal's On & Off Times and at result it affects motors On/Off times which comes as speed control (Because we can't see 490 times open-closed situation at motor also motor works as smoothing filter that on-off signals)

If we understand that controlling principle we can start to writing programs. First I want to declare all pins to Arduino program beginning:

```
//MOTOR CONTROL
```

```
int RPwm1 = 10;
```

```
int RPwm2 = 9;
```

```
int LPwm1 = 5;
```

```
int LPwm2 = 6;
```

```
//LED & BUZZER
```

```
int Buzzer = 13;
```

```
int ArduLed = 12;
```

At upside, you can see that we declared only motor control pins and Led & Buzzer pins. And all that pins will be output (It will send data to the outside of Arduino Nano). So I declare this at setup section.

```
void setup()
```

```
{
```

```
pinMode(RPwm1, OUTPUT);
```

```
pinMode(RPwm2, OUTPUT);
```

```
pinMode(LPwm1, OUTPUT);
```

```
pinMode(LPwm2, OUTPUT);
```

```
pinMode(Buzzer, OUTPUT);
pinMode(ArduLed, OUTPUT);
digitalWrite(Buzzer, LOW);
digitalWrite(ArduLed, LOW);
}
```

After that declaration, we can pass the main loop function of Arduino. In first sketch, we will just light the led, give beep sound from buzzer and we will make some movements with motors. Here is loop function:

```
void loop() {
//Buzzer On, Led On Both Motors Forward
digitalWrite(Buzzer, HIGH);
digitalWrite(ArduLed, HIGH);
analogWrite(LPwm1, 250);
digitalWrite(LPwm2, LOW);
analogWrite(RPwm1, 250);
digitalWrite(RPwm2, LOW);
delay(500);
//Buzzer Off, Led Off, Both Motors Stopped
digitalWrite(Buzzer, LOW);
digitalWrite(ArduLed, LOW);
analogWrite(LPwm1, 0);
digitalWrite(LPwm2, LOW);
analogWrite(RPwm1, 0);
digitalWrite(RPwm2, LOW);
delay(500);
//Buzzer On, Led On Both Motors Backward
digitalWrite(Buzzer, HIGH);
digitalWrite(ArduLed, HIGH);
digitalWrite(LPwm1, LOW);
analogWrite(LPwm2, 250);
digitalWrite(RPwm1, LOW);
analogWrite(RPwm2, 250);
delay(500);
//Buzzer Off, Led Off, Both Motors Stopped
digitalWrite(Buzzer, LOW);
digitalWrite(ArduLed, LOW);
analogWrite(LPwm1, 0);
digitalWrite(LPwm2, LOW);
analogWrite(RPwm1, 0);
digitalWrite(RPwm2, LOW);
delay(500);
}
```

Please check the above code. At loop function we have 4 sections. Forward, Stop, Backward and Stop. I colored motor control functions with red and orange colors. That code will make our robot 2 second forward, 0,5 second stop, 2 second backward and 0,5 second stop.

But for a more clean code. We should write one function for controlling both motor and working time. And I want to declare motor speed value from -100 to +100. So here is our function:

```
void Set_Motor (float Lval, float Rval, int timex){ //3 variables in total, Left and Right Motor Speed Values and Time variable.
Lval = Lval*2.5; // We multiply by 2.5 (Our set motor function works between -100, +100 values for motor speed control.
Rval = Rval*2.5;
if (Lval >=0) { // If Lval (Left Value) is bigger than 0 we understand motor will go forward.
analogWrite(LPwm1, Lval); // Multiplied value comes here so 100 becomes 250 Which is nearly full (250/255)
digitalWrite(LPwm2, LOW);
} else { //If Lval < 0 software comes to here.
Lval=abs(Lval); // We take Absolute value of Lval (we wrote with minus sign)
digitalWrite(LPwm1, LOW); // Motor Backward Routine
analogWrite(LPwm2, Lval);
} // Same concept for right motor.
if (Rval >=0) {
```

```

analogWrite(RPwm1, Rval);
digitalWrite(RPwm2, LOW);
} else {
Rval=abs(Rval);
digitalWrite(RPwm1, LOW);
analogWrite(RPwm2, Rval);
}
delay(timex); // How much motor will work on that speeds here we declare with timex value.
}

```

So after that function if I want My robot to go forward at full speed for 2 seconds. Just I will write:

Set_Motor(100, 100,2000); //Set_Motor(Left Speed, Right Speed, Time);

Please control yourself, according to the function. *And one information more. When we multiply 100 with 2.5 It gives us 250/255 speed. Genesis's mosfet drivers need little off time so that 5/255 is good for giving breath to motors.*

And now our stop statement is only that line:

Set_Motor(0, 0,500); // 500 millisecond stop

If you understand motor control functionality. We can continue to other aspects of Genesis.

Sensor Inputs & Trimpots for Genesis

You can attach your sensors to Genesis board with sensor inputs. We designed the board for maximum 7 sensors (Actually there is 2 D0 and D1 pins at the middle bottom of the board but that pins main usage is communication with PC so we are not using unless needed) You can easily attach the sensors with a seperate power source and signal lines.

So let's look the sensor pins:

A0: Analog Input, Digital Input or Output
A1: Analog Input, Digital Input or Output
D8: Digital Input or Output
D7: Digital Input or Output
D4: Digital Input or Output
D11: Digital Input or Output
D2: Digital Input or Output

We gave output 2 analog pins (A0-A1) and 5 digital pins (D8-D7-D4-D11-D2). Our purpose for giving outside that analog pins is using at edge sensor at mini sumo robots. That two analog pins (A0 & A1) can work as a comparator in software with using Genesis's trimpots. (For example, if A0 pin's analog voltage input is higher than A5 trimpot's value go backward or dance!)

And trimpot inputs

[ROBOT SHOP](#)
[A5: Analog Input, Attached to Edge Trimpot](#)
[A6: Analog Input, Attached to Turn Trimpot](#)
[CONTACT \(HTTP://JSUMO.COM/CONTACT\)](#)
[A7: Analog Input, Attached to Speed Trimpot](#)

[NEW! PHOTO CONTEST FOR CONCURSO \(APRIL\) \(HTTP://JSUMO.COM/SPRING-PHOTO-COMPETITION/\)](#)
[LEARN](#)
[MY ACCOUNT](#)

[MICROSTART: CHEAP START MODULE \(HTTP://JSUMO.COM/SHOP/START-MODULE/\)](#)
[WHAT'S NEW? \(HTTP://JSUMO.COM/NEW-PRODUCTS-FROM-JSUMO/\)](#)

```

//TRIMPOTS
int SPD = A7;
int TRN = A6;
int ETRN = A5;
//INPUTS
int Redge = A0;
int Ledge = A1;

```

```

int Value = 0;
void setup()
{
  Serial.begin(9600);
}
void loop() {
  Value=analogRead(SPD);
  Serial.print(Value);
  Serial.print("\t");
  Value=analogRead(TRN);
  Serial.print(Value);
  Serial.print("\t");
  Value=analogRead(ETRN);
  Serial.print(Value);
  Serial.print("\t");
  Value=analogRead(Redge);
  Serial.print(Value);
  Serial.print("\t");
  Value=analogRead(Ledge);
  Serial.println(Value);
  delay(300);
}

```

That code will send your PC to analog values of that pins in every 300 millisecond. (Genesis must be attached to computer via USB cable.)

And digital pins can be easily controlled with the following query.

```

if (digitalRead(Button)==0) {
  Statements;
}

```

Here is one example from 3 opponent sensor minisumo code:

```

//Opponent Sensor Control Routine
if (digitalRead(MSens)==LOW) {Set_Motor(MaxSpeed, MaxSpeed,1); digitalWrite(Buzzer, HIGH); LastValue=5;} else
if (digitalRead(LSens)== LOW) {Set_Motor(-40, TurnSpeed,1); digitalWrite(Buzzer, HIGH); LastValue=7;} else
if (digitalRead(RSens)==LOW) {Set_Motor(TurnSpeed, -40,1); digitalWrite(Buzzer, HIGH); LastValue=3;}

```

Here we need to explain some declarations. We have two different variables which are MaxSpeed and TurnSpeed. we declare them at the beginning of code like that:

```

int MaxSpeed = 90;
int TurnSpeed = 45;

```

So when you substitute at Setmotor functions. That speeds will turn motor in nearly full speed and nearly half speed.

First we declare all things at the beginning of code (Pin names, values...) Later we write in setup function what that pins are making (Input or output). And we declare our motor controlling function after setup. And at last stage in loop function We control our sensors, trimpots, and dipswitches with analog input and digital input query functions. (If statement is suited nicely).

Dipswitch, Button & Start Input

These components are our robot's, the system's input elements. Dipswitch and buttons share same mechanical base. The just different thing is button is momentary switch. But dipswitch settings stay.

And they work as digital inputs (Button is pressed or not pressed, Dipswitch is changed or not changed – Clear.) So in the beginning of program first I am entitling with better names.

```

// DIPSWITCH & BUTTON
int Button = 3; // Can be used as start pin too.
int DS1 = A4;

```

```
int DS2 = A3;
int DS3 = A2;
```

The button is connected to D3, and 3×2 Dipswitch pins connected to A4, A3, A2. Secondary pins of Button and dipswitch is connected to the ground line. So we should open software pull-ups at a program in the setup section.

```
digitalWrite(DS1, HIGH); //Making high in setup section adds software pullup resistor to input.
digitalWrite(DS2, HIGH); //Making high in setup section adds software pullup resistor to input.
digitalWrite(DS3, HIGH); //Making high in setup section adds software pullup resistor to input.
digitalWrite(Button, HIGH); //Making high in setup section adds software pullup resistor to input.
```

And when we work on program now we can query the button's input like that:

```
if (digitalRead(Button)==0) { // Is button pressed? If it is not, it is giving +5V from pullup resistor.
```

And when button pressed or dipswitch position changed to upside that inputs pins start to give 0V (GND).

For instance here is our board test function program. If you press the button while giving electricity to the board it will make some test movements.

```
if (digitalRead(Button)==0) {
tone(Buzzer, 18, 300); // Pin, Frequency, Duration
while (1) {
if (digitalRead(DS1)==0 && digitalRead(DS2)==0 && digitalRead(DS3)==0) {
Serial.print("Board Test");
Set_Motor(80,80,1000);
Set_Motor(0,0,1000);
Set_Motor(-80,-80,1000);
Set_Motor(0,0,1000);
tone(Buzzer, 18, 300);
tone(ArduLed, 18, 300);
}
}}
```

I marked with red the query statements. They look for first is button pressed? and later program looks for if 3 dipswitch pin (DS1, DS2, DS3) is at up side.

What should you do when you get Genesis Board?

First be calm! Don't panic 😊 Connect your motors, sensors. Our recommendation for sensor connection is

1. 2 x Edge Sensors QTR1A or QRD1114 sensors go to A0, A1 (A0 Right Edge, A1, Left Edge)
2. 3 x Opponent Sensors MZ80, MR45 or any 5V output sensor connects to D8, D4, D2 (D8 Left, D4 Middle, D2 Right)
3. And connect motors. You can connect nearly all models of Pololu motors, our kronos model motors. If the motor draws less than 4 amperes while working no problem.
4. Connect your power supply to Genesis. We recommend 2S,3S, 4S LiPo battery.
5. If Arduino is not programmed please program by downloading mini sumo program.
For Minisumo program click here (RAR). (<http://jsumo.m5industry.netdna-cdn.com/wp-content/uploads/2015/12/Genesis-Minisumo-code2.rar>)

When you do that steps your robot will start to live. You should read the comments well.

Which things can you tune? Change?

Firstly if your robot uses more or less than 3 sensors you need to change that sensor query statements.

Secondly you can change MaxSpeed and TurnSpeed variables they change with the different type of motor.

Thirdly you can change retreat time when robot sees edge sensors. The bottom line is at our program.

```
Set_Motor(-100, -100,35);
```

For fast robots 35 milliseconds enough. But if your robot works slower please make higher value like 100-200 ms.

Share this post



Author



Firat Dede (<http://jsumo.com/author/firat-dede/>)

Hello! I am the addicted Robot Maker who lives in Istanbul, Turkey. I work at robotics business. And I am the leader of JSumo Team. At here, I share my experiences. I competed with my team at All Japan Robot Sumo Tournament at 2011 (23rd), 2012 (24th), 2013 (25th) and 2014 (26th) Contests.

Leave a Reply

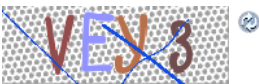
Your email address will not be published. Required fields are marked *

Comment

Name *

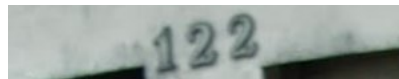
Email *

Website



CAPTCHA Code

*



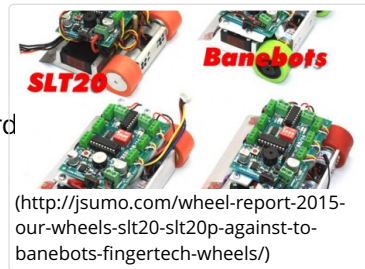
Type the text



[Privacy & Terms](http://www.google.com/intl/en/policies/)
(<http://www.google.com/intl/en/policies/>)

Post Comment

RELATED POSTS



22 Wheel Report 2015 – Our Wheels (SLT20 – Feb
SLT20P) Against to Banebots & Fingertech Wheels
(<http://jsumo.com/wheel-report-2015-our-wheels-slt20-slt20p-against-to-banebots-fingertech-wheels/>)

read more > (<http://jsumo.com/wheel-report-2015-our-wheels-slt20-slt20p-against-to-banebots-fingertech-wheels/>)

07 International Robotics Competition, Robochallenge 2015 – Oct
The countdown has begun!
(<http://jsumo.com/international-robotics-competition-robochallenge-2015-the-countdown-has-begun/>)

Between October 31 and November 1, Electronics Students League and ETTI Robochallenge, in partnership with the Faculty of Electronics,...

read more > (<http://jsumo.com/international-robotics-competition-robochallenge-2015-the-countdown-has-begun/>)

03 Picor (Mini Sumo)

Jul

(<http://jsumo.com/picor-mini-sumo/>)

Technical Drawings of minisumo.

...

read more > (<http://jsumo.com/picor-mini-sumo/>)



29 Robochallenge Romania 2015 is waiting for You!
Jul

(<http://jsumo.com/robochallenge-romania-2015-waits-you/>)

Passion, science, strategy or creativity... These are some of the words that describe the largest robotics competition in Romania! We...

read more >

(<http://jsumo.com/robochallenge-romania-2015-waits-you/>)

JSumo Loves You!



We create good parts for good projects. Founded in 2011, JSumo is a team and robotics hardware provider, supporting all robotics clubs, societies all over the world.

Our mission is adding at least to one brick to the wall when you are creating wonderful projects. We are with you with our society's projects
(<http://jsumo.com/blog>), shop page
(<http://jsumo.com/shop>) and forum
(<http://jsumo.com/forum>).



Killer Products from JSumo



JSumo Wheel 38x11mm Pair (Soft Silicone)
(<http://jsumo.com/shop/thin-silicone-wheel-38x11mm-pair/>)

silicone-wheel-38x11mm-pair/)

\$13.95



TCRT5000 Genuine Optocoupler
(<http://jsumo.com/shop/tcrt5000-genuine-optocoupler/>)

geniune-optocoupler/)

\$1.95

MP12 Micro Gear Motor 6V 1000RPM

Contact Us

📍 **Address:** JSumo, M5 Industry Mechatronics Corporation, ITU, ARI TechnoPark-3 Building- Istanbul, Turkey

☎ **Phone:** (+90) 212 221 17 19

✉ **Email:** sales[[@]]jsumo.com (mailto:sales[[@]]jsumo.com)

🕒 **Working Days/Hours:** Mon - Sun / 9:00 AM - 8:00 PM (UTC +2)

What We Have For You?

Sensors (20)
([Http://jsumo.com/Product-Category/Sensors/](http://jsumo.com/Product-Category/Sensors/))

Mechanical (39)
([Http://jsumo.com/Product-Category/Mechanical/](http://jsumo.com/Product-Category/Mechanical/))



Are you our Hero?

JSumo

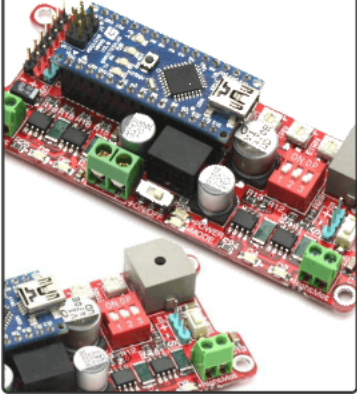
GROWING.

We are looking for worldwide Distributors.


(http://jsumo.com/contact/)

GENESIS

The Most Powerful




(http://jsumo.com/shop/genesis-arduino-robot-controller-board/)




(http://jsumo.com/shop/mp12-micro-gear-motor-6v-1000rpm/)

\$9.95



(http://jsumo.com/shop/mp12-micro-gear-motor-6v-650rpm/)

\$9.95



(http://jsumo.com/shop/mp12-micro-gear-motor-6v-500rpm/)

\$9.95

Wheels (15) (Http://jsumo.com/Product-Category/Wheels/)	
Electronics (60) (Http://jsumo.com/Product-Category/Electronics/)	+
Motors (21) (Http://jsumo.com/Product-Category/Motors/)	
Batteries & Chargers (11) (Http://jsumo.com/Product-Category/Batteries-Chargers/)	+
Kits (15) (Http://jsumo.com/Product-Category/Kits/)	
Magnets (16) (Http://jsumo.com/Product-Category/Magnets/)	

All prices are in USD currency. Thanks for your shopping.

VISA

PayPal

DISCOVER

Master

Master Card

AMERICAN EXPRESS

Powered By Coffee © Copyright 2011 - 201

http://jsumo.com/minisumo-tutorial-genesis-arduino-mainboard/

Page 9 of 9