Delta Library for Robotics Projects with Arduino
-The Unicorn

The Unicorn Library from DeltaINC:

What is Unicorn Library or Delta library?

Hello world , This is DeltaINC, giving you a new library for making your Arduino projects easy and effective. Usually you need to write more and more lines of code making Arduino projects like Obstacles avoiding robot, Bluetooth controlled robot, voice controlled robot,etc.. . But with Delta Library you only needs two statements in your Arduino program to make Obstacles avoid robot or for a Bluetooth Controlled Robot.

```
#include <Delta.h>    // including Delta Library for short our Arduino Program
void setup()
{
  UnicornBegin();     //starts the Unicorn Library
}
void loop()
{
  UnicornBTcontrol();  // Enables you to control the robot with Bluetooth
}
```

You will never feels hard to use our Library. Because made this Library is made in a way to reduce the code and also we provide you clean'n'clear documentations and tutorials.

This Article Contains the list of functions that are available in the Delta Library(Unicorn Library):

This Article can be separated in to two parts. One is the Unicorn part and the other is one for the custom circuit makers who wants to work with the Delta library

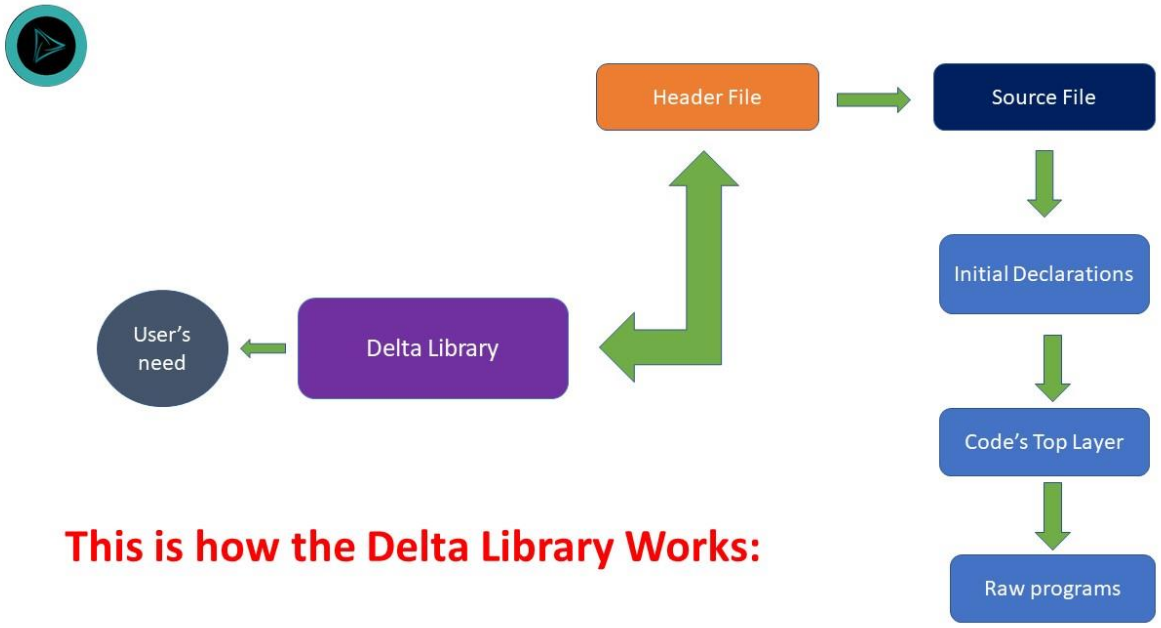
An Introduction to Unicorn Shield :

Unicorn is a external sheid for Arduino nano which allows the users to make Bluetooth controlled robot, obstacles avoiding robot, both Bluetooth and obstacles avoiding robot and many other Arduino related robots in very very less time. Unicorn allows the user to just plug-in play the components (Arduino nano , Bluetooth Module, Ultrasonic Sensor)in a board.

This Delta Library is for both the Unicorn and other Electronic circuits which plays with Arduino in the field of making robotics projects.

Delta Library for Robotics Projects with Arduino
-The Unicorn

How the Delta Library Works?:



This is how the Delta Library works . The Library consist of more and more codes which are nested with in many functions. So , the user needs only few lines of code to make a huge projects with ease.

Difference between Unicorn Lib and Normal Lib in Delta Library:

Unicorn Lib :

Unicorn lib consist of codes as same the the normal lib but the pin assignment pre configured in order to use the Unicorn Sheild. i.e: On using the Unicorn lib user can not define the pins according to their comfort they can only use the already defined pins (2,3,4,5,11,10). How ever in Normal lib the User change the pin as their wish. This pin lock is only for using the Unicorn Sheild effectively.

Now let us discuss the functions and their uses in detail:

Preface:

First we are going to discuss about the functions for the Unicorn Shield with a neat tabular column and the we are going to see about Delta Library for custom circuits. Don't get confuse ,both the Unicorn and Normal Library are nested in the Delta Library and also don't worry about memory management we implemented the source code in a memory efficient way.

Delta Library for Robotics Projects with Arduino
-The Unicorn

The below table will give you a full clarity on the functions available in this Delta Library.

The examples given below are just for giving a little clarity on the functions . You can find more clear and neat examples in the examples folder in arduino which are attached from the Delta library.

For finding examples in Arduino IDE:

File→examples→Delta→Unicorn (or) custom circuits

Function syntax	Description	Example
Unicorn functions:		

UnicornBegin();	This function setups the required pins as INPUT or OUTPUT and also contains other necessary declarations.
	Note: this must be embedded on void setup()
		Example for Note;
		void setup()
		{
		UnicornBegin();
		}

UnicornBTcontrol();	This functions allows the users to communicate the arduino via Bluetooth as a Bluetooth Controlled robot.
	i.e: after embedded this on void loop() you can control the robot for more tutorials visit out blog which is in the below of this article
		Example :
		
		void loop()
		{
		UnicornBTcontrol();
		}

UnicornGo();	This functions makes obstacles avoidance for your projects.
	For Other information about the project visit our blog
		Example:
		void loop()
		{
		UnicornGo();
		}

FindObject();	This function returns the value of detected object. Here value 30 = 23 CM	Examples:
		{
		Variable=FindObject();
		Serial.println(Variable);
		}

UnicornDuo();	This functions makes both obstacles avoidance and Bluetooth control.
	For more details visit our blog
		Example:
		Void loop()
		{
		UnicornDuo();
		}

Unicornfwd();	This functions make the robot to move forward.
	It writes HIGH on 2,4 and LOW on 3,5
		Example:
		{
		Unicornfwd();
		}

Unicornbwd();	This function make the robot to move backward.
	It writes LOW on 2,4 and HIGH on 3,5
		Example:
		{
		Unicornbwd();
		}

Unicornstop();	This functions stops the movement.	Example: {

Delta Library for Robotics Projects with Arduino -The Unicorn

	It writes LOW on2,3,4,5	Unicornstop(); }

UnicornturnL();	Turns left and stops(
	It writes HIGH on 3,4 and low on 2,5 for and stops)
		Example:
		{
		UnicornturnL();
		}

UnicornturnR();	Turns right and stops
	It writes HIGH on 2,5 and LOW on 3,4
		Example:
		{
		UnicornturnR();
		}

UnicornSpinL();	Make the robot to spin on left side	Example:
		{
		UnicornSpinL();
		}

UnicornSpinR();	Make the robot to Spin right side	Example:
		{
		UnicornSpinR();
		}

Robotic Functions:		
Basic Declarations→	(Purpose) This two statements must be declared before writing the void setup and void loop. In the parenthesis of the variable_name you have to pass the Arduino pin numbers to use it for your projects. Below rows will discuss about passing the Arduino pin details to the Delta Library. You can replace the variable_name as your desired name.	#include<Delta.h> Delta variable_name();
Declarations of Object:		
For Bluetooth controlled robot	This type declarations are need for Bluetooth controlled robot. Replace the left motor +, left motor – , right motor + , right motor – with your desired Arduino pins . "for better clarifications open the examples and see our blog"	#include<Delta.h> Delta variable_name(left motor +, left motor – , right motor + , right motor –);
For Obstacles avoiding robot	This type declarations are need for Obstacles avoiding robot. Replace the left motor +, left motor – , right motor + , right motor – , Trig pin , Echo pin with your desired Arduino pins . Trig pin and Echo pin are need to be connected with HC-05 Ultrasonic Sensor "for better clarifications open the examples and see our blog"	#include<Delta.h> Delta variable_name(left motor +, left motor – , right motor + , right motor –,Trig pin,Echo pin);

Delta Library for Robotics Projects with Arduino
-The Unicorn

For Both Obstacles avoidance and Bluetooth Control	This type declarations are need for Obstacles avoiding robot. Replace the left motor +, left motor – , right motor + , right motor – , Trig pin , Echo pin with your desired Arduino pins . Trig pin and Echo pin are need to be connected with HC-05 Ultrasonic Sensor "for better clarifications open the examples and see our blog"	#include<Delta.h> Delta variable_name(left motor +, left motor – , right motor + , right motor –,Trig pin,Echo pin);
For only using Ultasonic sensor to find distance	Replace the Trig pin , Echo pin with your desired Arduino pins . Trig pin and Echo pin are need to be connected with HC-05 Ultrasonic Sensor "for better clarifications open the examples and see our blog"	#include<Delta.h> Delta variable_name(Trig pin,Echo pin);

start();	This function setups the required pins as INPUT or OUTPUT and also contains other necessary declarations.
	Note: this must be embedded on void setup()
		#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}

BTcontrol();	This functions allows the users to communicate the arduino via Bluetooth as a Bluetooth Controlled robot.
	i.e: after embedded this on void loop() you can control the robot for more tutorials visit out blog which is in the below of this article
		#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.BTcontrol();
		}

objavoid();	This functions makes obstacles avoidance for your projects.
	For Other information about the project visit our blog
		#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.objavoid ();
		}

FindObj();	This function returns the value of detected object. Here value 30 = 23 CM	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();

		}
		void loop()
		{
		Variable_name.FindObj();
		}

Duo();	This functions makes both obstacles avoidance and Bluetooth control.
	For more details visit our blog
		#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.Duo();
		}

fwd();	Make the robot to Move forward. Refer the blog and example codes for better understanding.	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.fwd();
		}

bwd();	Makes the robot to move backward	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.bwd();
		}

stopper();	Stops the movement of the robot	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.stopper();
		}

left();	Make sthe robot to turn left and then stop	#include<Delta.h> Delta variable_name(pass the Pin no as instruct above); void setup()

		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.left();
		}

right();	Makes the robot to turn right and then stops	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.right();
		}

SpinL();	Makes the robot to spin in left side	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.SpinL();
		}

SpinR();	Makes the robot to spin in right side	#include<Delta.h>
		Delta variable_name(pass the Pin no as instruct above);
		void setup()
		{
		variable_name.start();
		}
		void loop()
		{
		variable_name.right();
		}

Hope you will find us useful . Visit our website for more information

For more details and tutorials visit our blog:

Thank you for choosing us.