Disco-Droid

Bluetooth Ready BB-8 Dome FX Module

Roadmap

V1.0.6 - Current Final Board Design for Version 1
>Hardware Features
>>32 PWM Output (Can handle Motors, LEDs, and Servos, etc..)
>>7 Analog Inputs (Can also be used for Digital Outputs)
>>3 Digital Outputs/Inputs
>>3W Speaker Output
>>Bluetooth Ready

>Software Features
>>Randomized Sound file playing
>>3 Pre-set Visualizer patterns
>>Small compact Code (using only 14% of the Arduino's memory, and 12% buffer memory)
==========

V1.1.6 - Same Board Design Utilizing Bluetooth remote connectivity
>Hardware Features
>>Same as V1.0.6

>Software Features
>>When left alone, it will follow V1.0.6 program functions
>>Bluetooth Feature will be utilized to trigger the Dome light pattern changes, and sounds.

*In order to utilize V1.1.6 features a future Bluetooth Remote will be released to compliment the Disco-Droid Module.

Introduction

The Disco Droid is an Arduino based module that will mainly provide Visual and Audio Effect for BB-8's Dome. In order to create visual effects on the dome, more pins were needed that what an ordinary Arduino can provide, as such there was a need to create a module that has integrated output expander in order to accommodate the needs of the dome, and a few more extras for future upgrades.

The Disco-Droid is capable working alone, or remotely controlled by another Bluetooth 4.0 LE device, it has a total of 32 12-bit PWM outputs, 7 spare analog pins from the Arduino, and 3 Digital I/O pins also from the Arduino, the Module has also integrated a MP3 Playback module that can drive a single 3W speaker.

Most of all the different parts that was selected for this module has taken into account that many builders wish to make their domes as light as possible, in order to do so the Disco-Droid was design with the smallest off-the-shelf parts and modules in order to conserve on weight and size. The dimensions of the circuit board is just 2 x 3.5 x .7 inches, just slightly longer than your credit card.

Common trouble-shooting and assembly instructions

If you have gotten a completely assembled kit, all the components should have been soldered in, all you have to do is to attach the wires for the LEDs. Figure 1.0 will show you how your module should look like when you receive it.

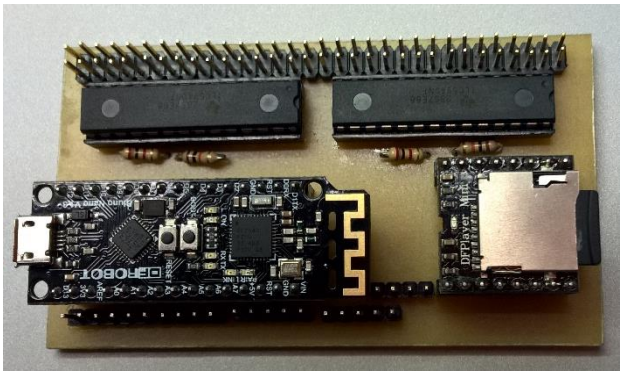

Figure 1.0 Disco-Droid Module with no attached Wires.

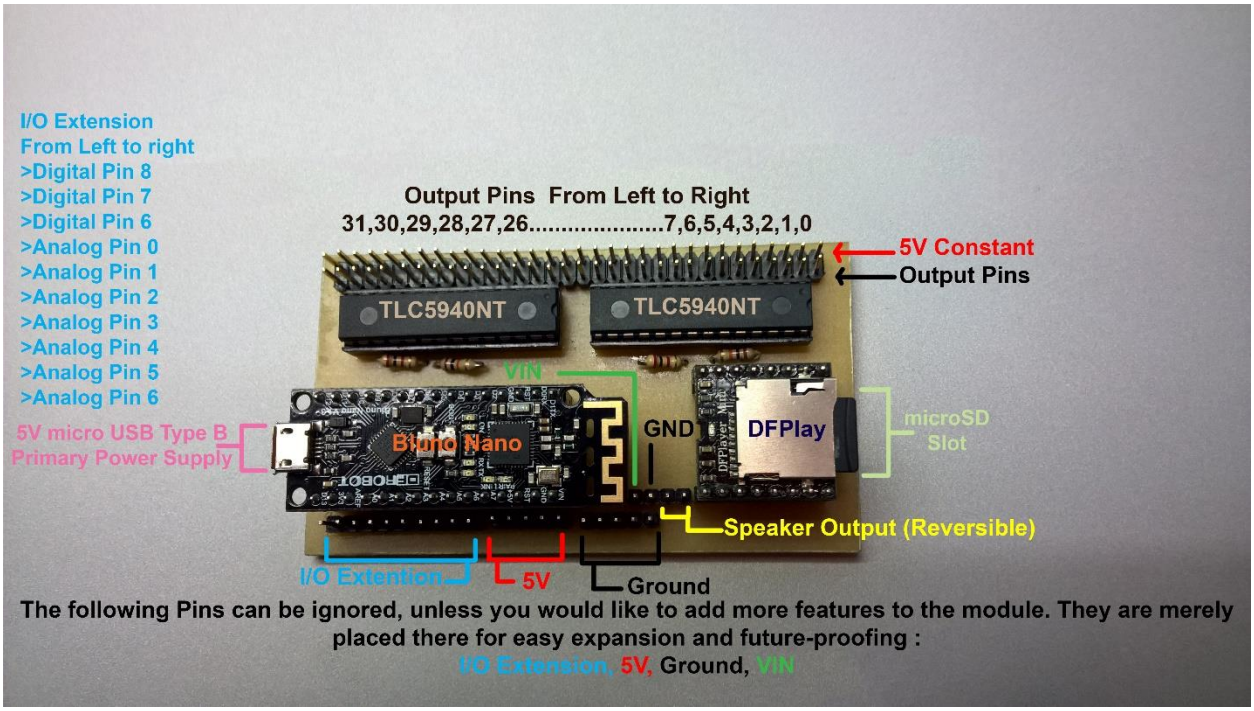

Figure 1.1 Completely labeled Disco-Droid Module.

For completely assembled kits, you will need to wire the LEDs and the speaker in. The LEDs in the package are bundled and labeled for quick and easy plugging.

When plugging the LEDs please be sure that the **red wires** are facing outside, and plugged into the 5V constant. The sequence in which you plug the LEDs are very important if you do not wish to modify the pre-loaded program. The proper LED wiring sequence is shown below.

Output Pins 0-4	→	Front		Top	Logics
Output Pins 5-9	→	Front		Bottom	Logics
Output Pins 10-13	→	Side			Logics
Output Pin 29	→	Radar		Eye	LED

Output Pin 30	→	Holoprojector	LED
Output Pin 31	→	PSI	LED
Output Pins 28-24	→	Optional 5 Radar Eye LED	

The other set of wires that will need to be connected is for the speaker. Simply follow Figure1.1 to locate where the speaker output should be connected, orientation does not matter in this case.

All modules will have been tested to work before shipping, as such attaching the LED and speaker wires, and connecting the 5V power supply will immediately power up the module, should it somehow not power up or there are lose parts upon arrival, please message me and I will assist you to get the module working.

*Should there be lose parts upon arrival please immediately send me a photo and do not power it up yet, by design having lose components will not damage the circuit, but it is always better to be safe.

Basic Program Modification

Disco-Droid is based on Arduino, an open-sourced platform used by many hackers and enthusiasts, it is quick and easy to understand and learn. In order to open the program of the Disco-Droid Module, you will need to download the Arduino IDE, the download link is below.

Arduino IDE Download Link

After installing the Arduino IDE, you will need to open the Libraries folder that I uploaded and copy the two folders inside into your Arduino Library, this adds the libraries needed by Arduino in order to properly program the Disco-Droid.

*The Arduino Library is typically found in C:\Program Files (x86)\Arduino\libraries

The program is written to be very easy to understand, I have placed comments on almost all the lines of the program to help you understand it. However, should you really just want to make modifications to the visualizer and nothing drastic it will only require you to change a few key words!

```
//////////Change this portion to change the visualizers////
Sweeper Fr_Logics_1(0, 4, 200 );
Fader Fr_Logics_2(5, 9, 1 );
Randomizer Sd_Logics(10, 13, 75 );
MP3Sequencer Sequencer(0);
////////////////////////////////////////////////////////////
```

Figure 2.0

In order to change the visualizer sequence for the Logics lights, all you have to do is replace the Keywords that are highlighted in yellow. The suggested refresh rate for each sequence is boxed in blue, you can opt to follow the values given, or you may experiment with the values yourself.

*Refresh rate value is in milliseconds. Ex. Sweeper will sweep from 1 LED to another every 200 milliseconds, Randomizer will generate a new set of brightness values every 75 milliseconds.

For example, you want to change Front Logics 2 to have the Randomizer sequence, you may simply change the code, from Fader Fr_Logics_2(5, 9, 1); to Randomizer Fr_Logics_2(5, 9, 75);

Each MP3 module may give a slightly different output value, generally it will give almost the same values, however, if you are not satisfied with the reaction of the PSI LED to the speaker output of the Droid, you can also easily tweak a few values.

```
void loop()
{
  bool Playing = digitalRead(mp3In);//check if MP3 is still playing
  Sequencer.Update(Playing);
  int PSI = analogRead(speakerIn);//Get Speaker value
  PSI = constrain(map(PSI, 400, 950, 600, 4095), 0, 4095);//Solve for PSI output brightness
  Fr_Logics_1.Update();//Update Front Logics 1 Display
  Fr_Logics_2.Update();//Update Front Logics 2 Display
  Sd_Logics.Update();//Update Side Logics
  Tlc.set(31, PSI);//Set PSI Brightness
  Tlc.set(30, 4095);//Set Holoprojector brightness
  Tlc.set(29, 4095);//Set Radar Eye Brightness
  Tlc.update();//Push new values to ICs
}
```

Figure 2.1

The Values highlighted in yellow are can be tweaked to your liking, it can make your droid more or less responsive depending you the values you enter.

The format is as follows map(PSI, fromLow, fromHigh, toLow, toHigh)

- fromLow is the anticipated lowest input value from the speaker
- fromHigh is the anticipated highest input value from the speaker
- toLow is the desired lowest LED output brightness
- toHigh is the desired highest LED output brightness

Once you have finished making your desired changes, simply plug the Bluno Nano into your computer, and upload your program.