AIGO RGB fan controller rough schematics

(cause I don’t know any schematics programs !!)

The AIGO Dr12 ICESTORM RGB fan pack comes in sets of 3, 5, 8 or more depending on the version you select on aliexpress and are quite cheap for what tey are capable off

They are dual side RGB fans and have a single controller which can take 8 fans and also 4 led strips. I haven’t seen any connector ports for daisy chaining these guys but it is quite possible once you see my discussion points on why is it so

# Schematics

## Input port

The input port is essentially just for ‘POWER’ and has a 5 pin connector which as 12v and 5v written on both sides. Starting from the 12v silk screen labels here are the pinout description

|  |  |  |
| --- | --- | --- |
| S no | Pin number | Description |
| 1 | Pin 1 | 12 v rail |
| 2 | Pin 2 | 12 v rail |
| 3 | Pin 3 | GND |
| 4 | Pin 4 | GND |
| 5 | Pin 5 | 5 v rail |

The 12v rails connect to the motor which seems to be rated at 12 v, The 5v rail is powering the RF-controlled RGB controller situated in the middle of the board (more on that later).

The design of the board is a little strange with no PWM control on the fan (either 12v is supplied or not) and the 5v rail directly powering the RGB controller. Also, there is no thermal management anywhere on the board (assuming the amount of copper trace on the 12v and 5v rail itself can also spread the heat quite effectively, but this is open to discussion as I am no product engineer)

## Fan ports

The design of the fan connector is a standard 5 pin connector having the pinout as described heare starting from the side from which the ‘FAN x’ silkscreen print arrow points in each section (the head of the arrow points to the farthest pin = PIN 6)

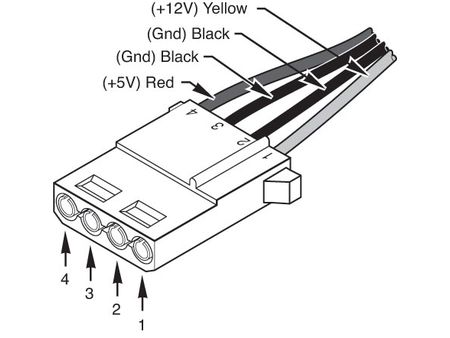
|  |  |  |
| --- | --- | --- |
| S no | Pin number | Description |
| 1 | Pin 1 | 12 v rail |
| 2 | Pin 2 | GND |
| 3 | Pin 3 | GND |
| 4 | Pin 4 | DATA IN |
| 5 | Pin 5 | 5 v rail |
| 6 | Pin 6 | DATA OUT |

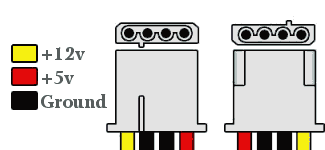
The presence of the DATA in and DATAOUT seems to suggest the Ws2812X type of ‘neopixels’, let’s see that in the next sections.

As it suggests, the 12 v rail is for the FAN, the 5v is purely for the RGB controller. Creating an open source version of the controller with esp32 and/or similar chip would be interesting.

## Molex Connector:

The molex connector uses the standard pinout as shown below





Notice the sloping notches and match them to the orientation of your molex connector to correctly figure out the 5v and 12v rails. You can damage your equipment (PS, oscilloscope, logic analyser) if they aren’t rated for these voltages.

## The RF board

Here is the meat of the DR12 controler box, the RGB board controller

On a first inspection the usual suspects are there,

1. A power section for the STC chipset (which is running the code)
2. The STC 15w104 chipset
3. A RF section of the board, probably 433 MHZ (as the crystal is a 6.7458)

We can do a little better ☺

The other side of the board reveals a RF antennae printed onto the board and a board version (CSY007-RF-5v-24v\_v21)

The a board search on google with this name came up with a whole lot of companies selling a simple RF controller, a one with description is given below (translate using google translate)

<http://www.china-nengyuan.com/product/21012.html>

so its just a RGB controller, and this closes in on our assumption on the protocol (must be the ws2812b) type (neopixel protocol or equiv)

### The STC Chipset

Here are the details on the STC chipset:

Model: STC 15W104 (STC 15w104 hsw355XA onboard printed name)

Working voltage (V)) : 2.5-5.5

Flash program memory: 4 K

Large capacity SRAM : 128

Serial port and wake up after power down: -

SPI : -

Ordinary locator / timer ( TO-T external pin can also wake up with power ): 2 ( T0/T2 )

PCA/CCP/PWM can be used as an external interrupt and can be powered down to wake up: -

Power-down wake-up timer: Yes

Standard external interrupt support point wake up: 5

A/D8 way (can be used as 1 way PWM can be used as 3 way D/A ): -

The comparator can be used as an external power failure detection when 1 channel A/D : -

EEPROM: 1K

Internal low-voltage detection interrupt and wake-up with power-down: Yes

Watchdog: Yes

Internal high reliability reset ( optional reset threshold voltage ) : 16 levels

Internal high precision clock: Yes

Output clock and reset externally: yes

Program encrypted transmission ( anti-blocking ): Yes

You can set the password for the next update program: Yes

Support RS485 download: yes

Package: SOP-8 DIP-8 DFN-8

Umm, for an 8 pin chipset, quite decent. I was somewhat doubtful that the security bit must be set in order to avoid getting the firmware read out (but I’ll let better engineers be the judge of that), but I am not interested as this entire board will be replaced with a better one

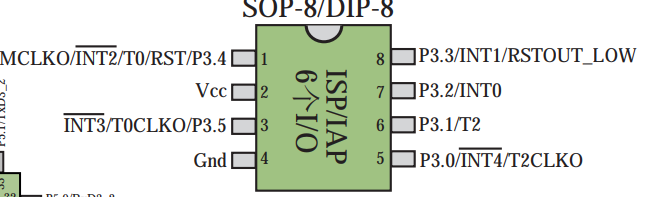
### Pinout to main board

The pinout to the main board is very simple

|  |  |  |
| --- | --- | --- |
| S no | Silk screen name | Description |
| 1 | 05 + | 5v rail |
| 2 | 05 - | GND |
| 3 | D | DATA OUT |
|  |  |  |
|  |  |  |

The entire board is physically supported by just these three SOLDER JOINTS !!! so it is possible that this board is modular and AIG may have better controller circuits (with more features) and they just slap them on based on the product line you buy. NEAT !!! but fragile !!!

The D pin is connected via resistor to PIN no 1 of the SOP-8 package chip version



SO that is controlling the RGB DATA OUT. This pin is connected to FAN 1’s in port. SO all the animation that come up will begin with FAN 1.

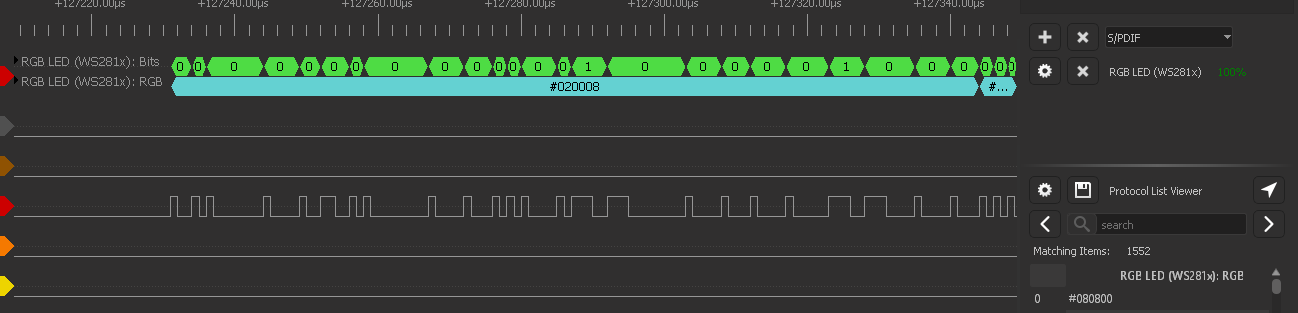
Now we know what to do

1. TO start off, HOOK ONTO these three pins (5v, GND and DATA OUT) and see the protocol being followed
2. Once we know the protocol, write a simple esp32 app
3. Test esp32 app
4. Successful, create a board with the following
   1. Better power delivery section
   2. Fuses and over-current protection
   3. Esp32 based rgb controller
   4. PWM control for fan
   5. Support for controller-to-controller daisy chaining
   6. REST API and BLE API for controller

These are my requirements for this project and I welcome additional work on this.

## Protocol analysis

I started diggininto the board with my trusty DSLOGIC pro logic analyser and sure enough the protocol is the neopixel protocol of the ws281x protocol



With this knowledge I am done, now I can start concentrating on writing my esp32 code.

## Update: 20-Jan 2019

I was wrong about the chipset of the RGB. Now I am at a stand still on development as driving the LED via FastLED lib on arduino is not giving proper results. I am definitely going to run the info from my logic analyser for figuring out the correct protocol (or worst case bitbang them once I figure out the protocol)