Edzelf Esp-radio Alternate Module Sroto&Gargees™ Version

(WeMos D1R1 + VS1053B)

Successor of NodeMCU+ VS1053A version

An Open Source Documentation

By

srotogargees.business.site

c/o surenbono76@gmail.com

Advantage:

- Wire Free (Plug’n’Play)
- Cheap and Smoother than Edzelf Esp32 radio
- Stronger Signal on vs1053b than esp32 (stabled by usetft option)
- Unlimited Data, Copper Broadband 24/7 run Tested (esp32 failed)
- Simple & Compatible with other shields
- Windows Xp capable with some effort
- Wifi with LED indication (usetft option)
- Better buffer fill capability
- Uno style board
- Modifiable web interface
- Not watchdog manual reset for initial sketch upload like esp32
- Single sketch page & capable of OTA upload modification

Disadvantage

- Not Effortless (manual esp8266 platform install)
- Bare Minimum Setup (WeMos d1r1 + vs1053b only)
- Not Local host SD/ TF web File player yet (void programming skill)
- Needs 2wire vs1053b hard solder jumper modification & sacrificing 2 vs1053b pins (X-DCS- RX & X-CS- TX)

  (Conflicting with redundant D1 excess SPI’s on pin D6 & D7)

- radio.ini needs re-upload in some case
- Exposed Wifi password on web interface setup
- Jedi mind tricking program (D1 as Nodemcu)
- OTA preset select ,OTA treble & bass control only
- Only runs smoothly when (usetft) defined

DRIVER: CH341SER

Download latest bin: http://micropython.org/

Flashed esp8266-20190125-v1.10.bin on nodemcu-flasher-master for this project.

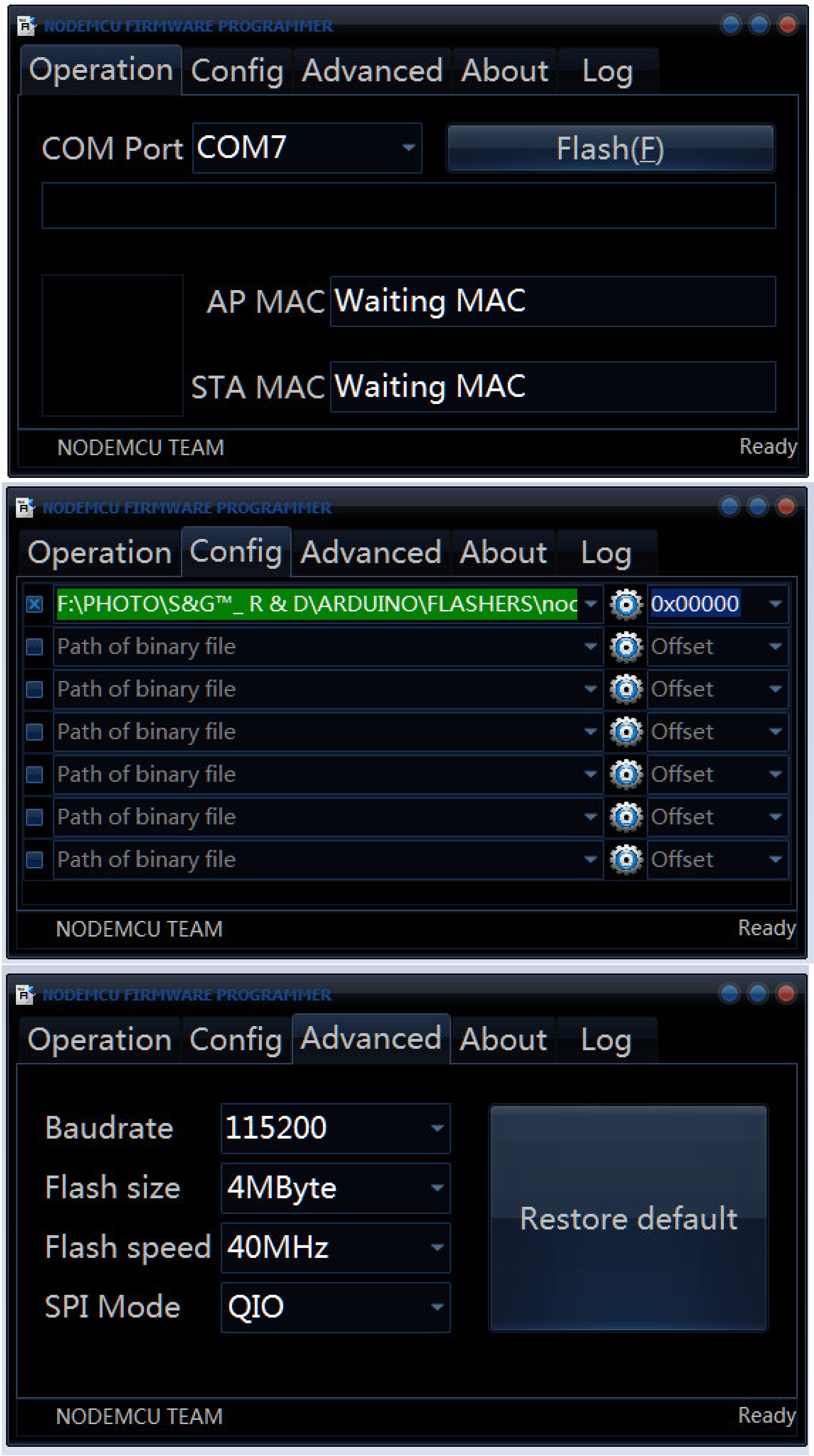
PLATFORM

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Integrated Development Environment /Toolchain (IDE)

https://www.arduino.cc/ en/Main/Software

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.



COM Port COM7

AP MAC Waiting MAC

STA MAC Waiting MAC

☒ F:\PHOTO\S&G™_ R & D\ARDUINO\FLASHERS\noc 0x00000

□ Path of binary file Offset

□ Path of binary file Offset

□ Path of binary file Offset

□ Path of binary file Offset

□ Path of binary file Offset

□ Path of binary file Offset

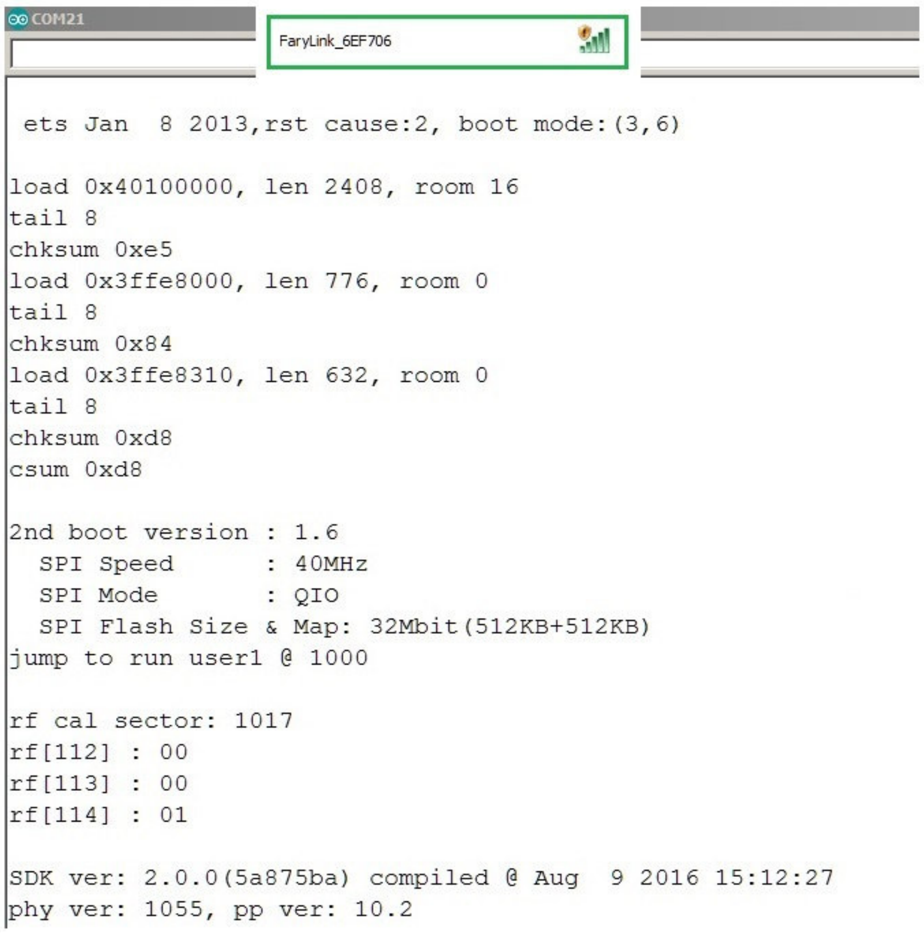
Baudrate 115200

Flash size 4MByte

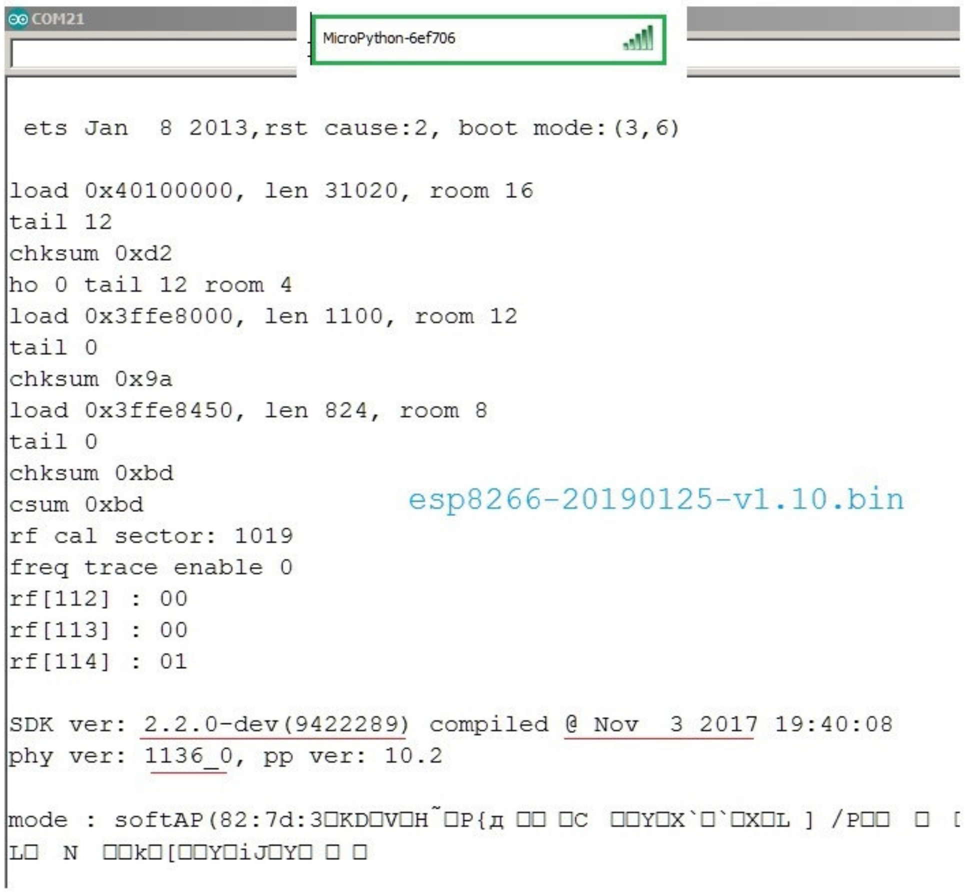
Flash speed 40MHz

SPI Mode QIO

NodeMcu Flasher

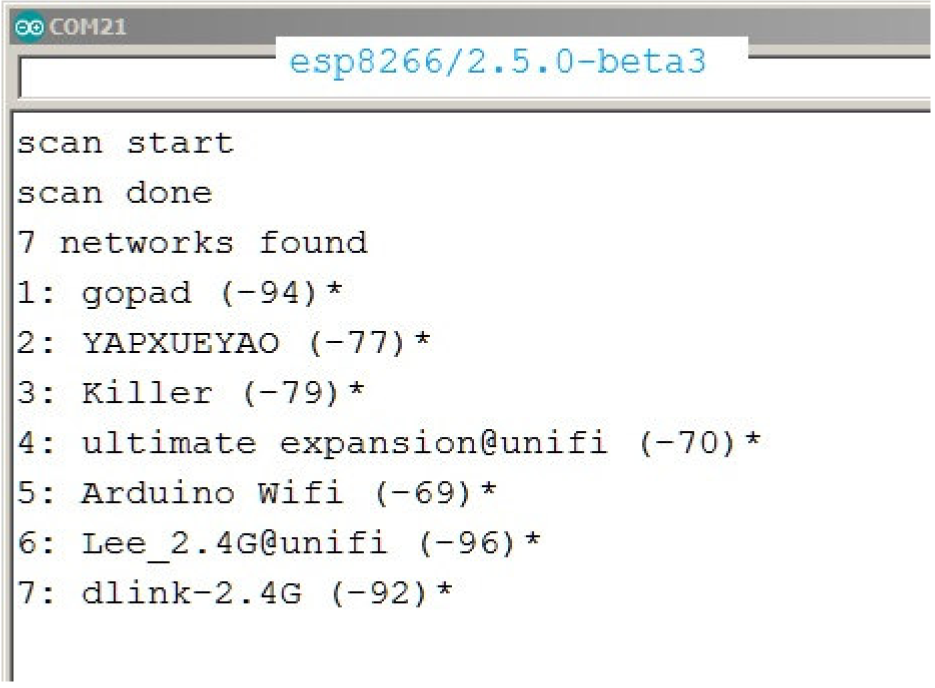


```
 ets Jan  8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 2408, room 16
tail 8
chksum 0xe5
load 0x3ffe8000, len 776, room 0
tail 8
chksum 0x84
load 0x3ffe8310, len 632, room 0
tail 8
chksum 0xd8
csum 0xd8

2nd boot version : 1.6
  SPI Speed      : 40MHz
  SPI Mode       : QIO
  SPI Flash Size & Map: 32Mbit(512KB+512KB)
jump to run user1 @ 1000

rf cal sector: 1017
rf[112] : 00
rf[113] : 00
rf[114] : 01

SDK ver: 2.0.0(5a875ba) compiled @ Aug  9 2016 15:12:27
phy ver: 1055, pp ver: 10.2
```



```
 ets Jan  8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 31020, room 16
tail 12
chksum 0xd2
ho 0 tail 12 room 4
load 0x3ffe8000, len 1100, room 12
tail 0
chksum 0x9a
load 0x3ffe8450, len 824, room 8
tail 0
chksum 0xbd
csum 0xbd
rf cal sector: 1019
freq trace enable 0
rf[112] : 00
rf[113] : 00
rf[114] : 01

SDK ver: 2.2.0-dev(9422289) compiled @ Nov  3 2017 19:40:08
phy ver: 1136_0, pp ver: 10.2

mode : softAP(82:7d:3□KD□V□H˜□P{д □□ □C  □□Y□X`□`□X□L ] /P□□  □ [
L□  N  □□k□[□□Y□iJ□Y□ □ □
```

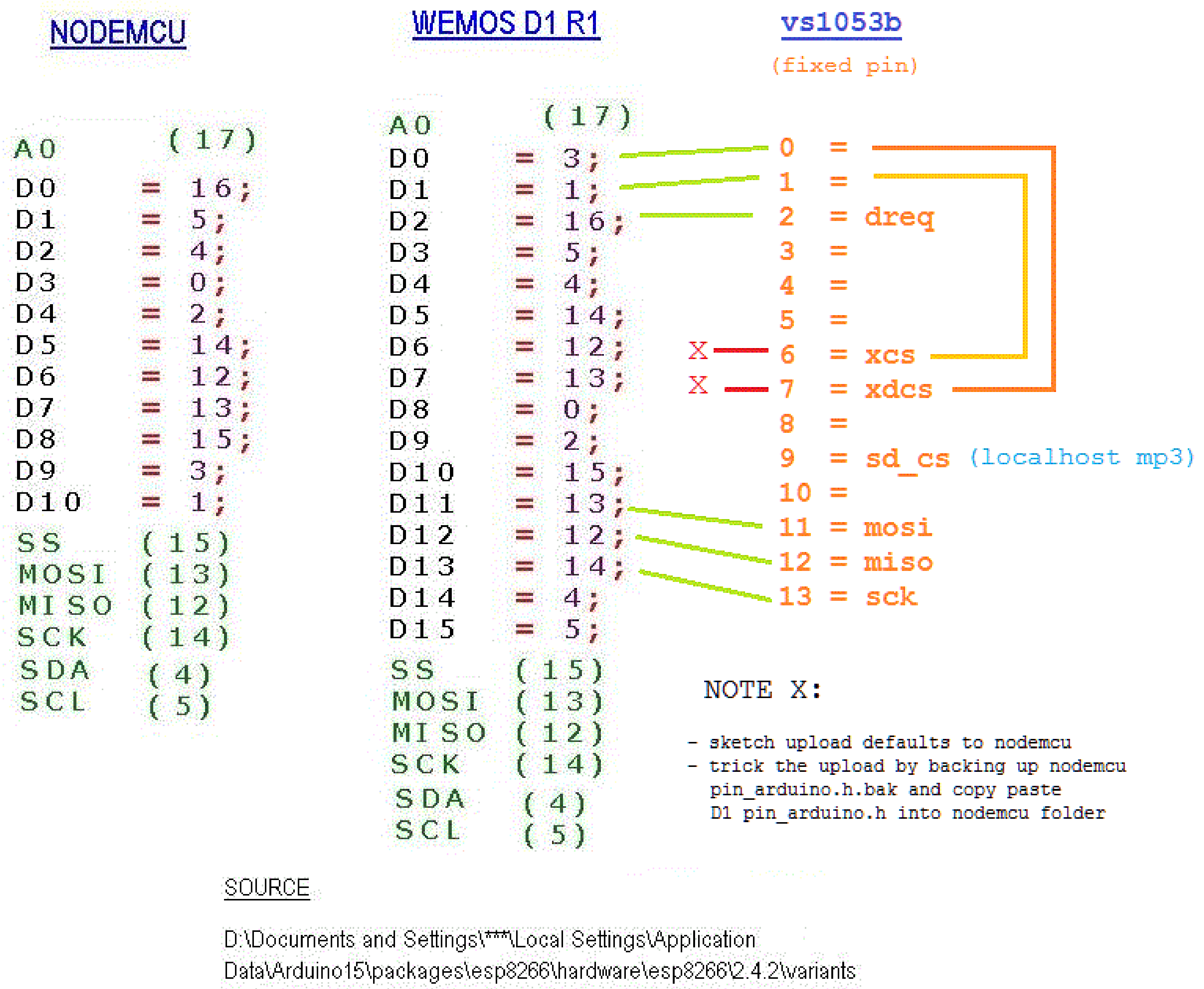
Comparison before after flashing



```
scan start
scan done
7 networks found
1: gopad (-94)*
2: YAPXUEYAO (-77)*
3: Killer (-79)*
4: ultimate expansion@unifi (-70)*
5: Arduino Wifi (-69)*
6: Lee_2.4G@unifi (-96)*
7: dlink-2.4G (-92)*
```

Wifi Scan Sketch Examples Output on Serial Monitor

Migration Brainstorming Notes



NODEMCU

A0	(17)
D0	= 16;
D1	= 5;
D2	= 4;
D3	= 0;
D4	= 2;
D5	= 14;
D6	= 12;
D7	= 13;
D8	= 15;
D9	= 3;
D10	= 1;
SS	(15)
MOSI	(13)
MISO	(12)
SCK	(14)
SDA	(4)
SCL	(5)

WEMOS D1 R1

A0	(17)
D0	= 3;
D1	= 1;
D2	= 16;
D3	= 5;
D4	= 4;
D5	= 14;
D6	= 12;
D7	= 13;
D8	= 0;
D9	= 2;
D10	= 15;
D11	= 13;
D12	= 12;
D13	= 14;
D14	= 4;
D15	= 5;
SS	(15)
MOSI	(13)
MISO	(12)
SCK	(14)
SDA	(4)
SCL	(5)

vs1053b

(fixed pin)

	0	=
	1	=
	2	= dreq
	3	=
	4	=
	5	=
X	6	= xcs
X	7	= xdcs
	8	=
	9	= sd_cs (localhost mp3)
	10	=
	11	= mosi
	12	= miso
	13	= sck

NOTE X:

- sketch upload defaults to nodemcu
- trick the upload by backing up nodemcu pin_arduino.h.bak and copy paste D1 pin_arduino.h into nodemcu folder

SOURCE

D:\Documents and Settings***\Local Settings\Application Data\Arduino15\packages\esp8266\hardware\esp8266\2.4.2\variants

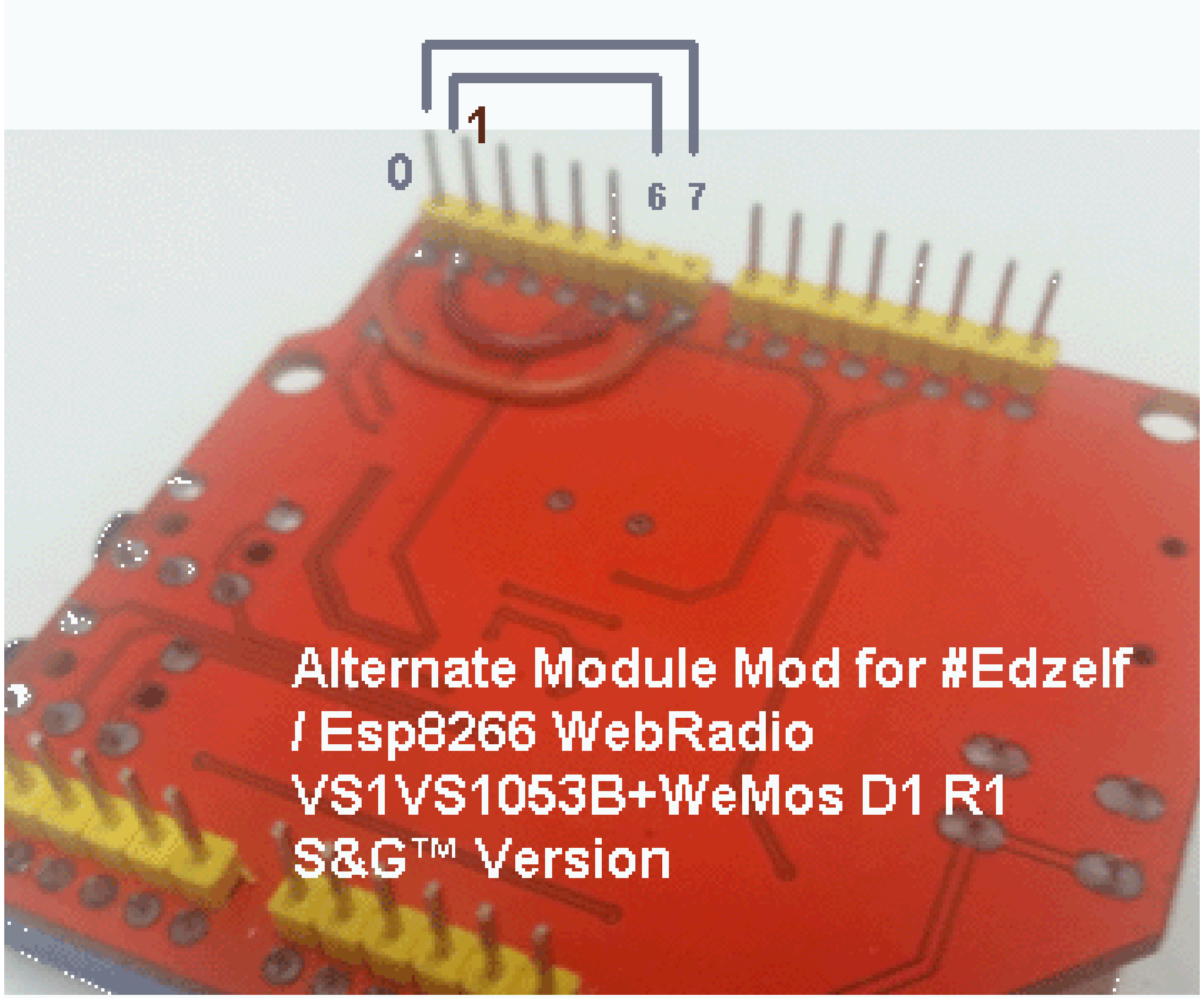
Succeeded Algorithm Examples

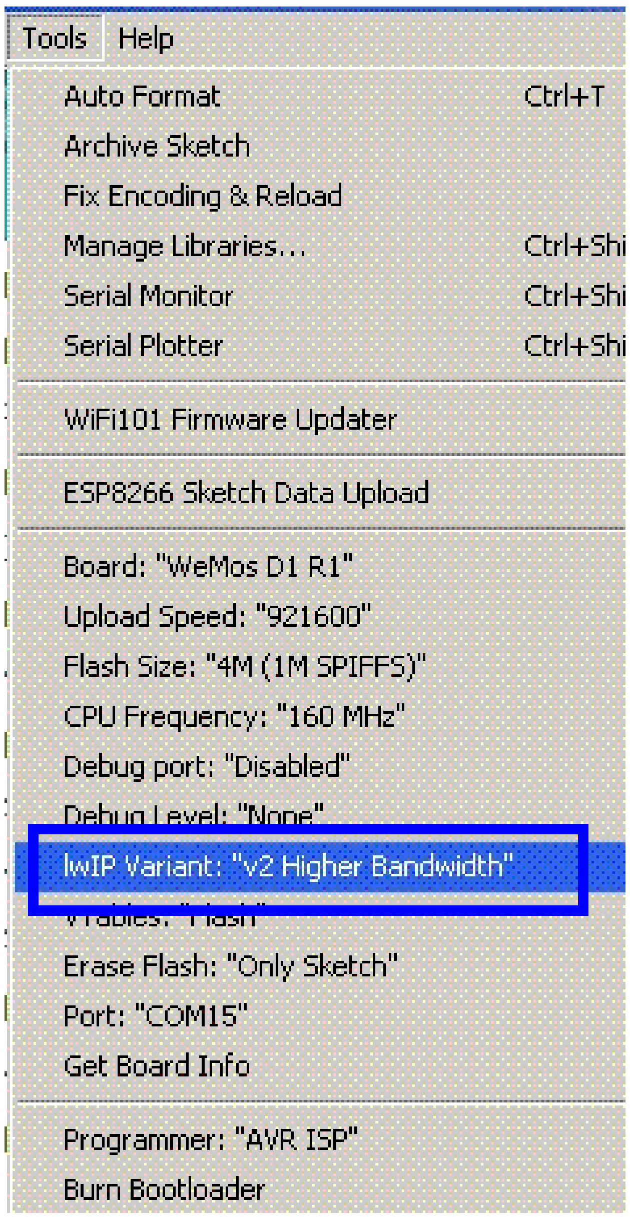
```
// Pins for VS1053 module

#define VS1053_DCS  3

#define VS1053_CS   1

#define VS1053_DREQ 16

 // Pins CS and DC for TFT module (if used, see definition of "USETFT")

#define TFT_CS 15

#define TFT_DC 2 //blinks inbuilt LED when wifi on

// Control button (GPIO) for controlling station

#define BUTTON1 4 // SDA

#define BUTTON2 17 //A0

#define BUTTON3 5 // SCL
```

Note x:

- A0 can be programmed with resistors to work more than one button.
- SDA & SCL can be reserved for i2c OLED display
- CS_SD of vs1053b can be used for LOCALHOST play
- "USETFT" should be defined for smooth play on bare minimum else returns choppy streaming.

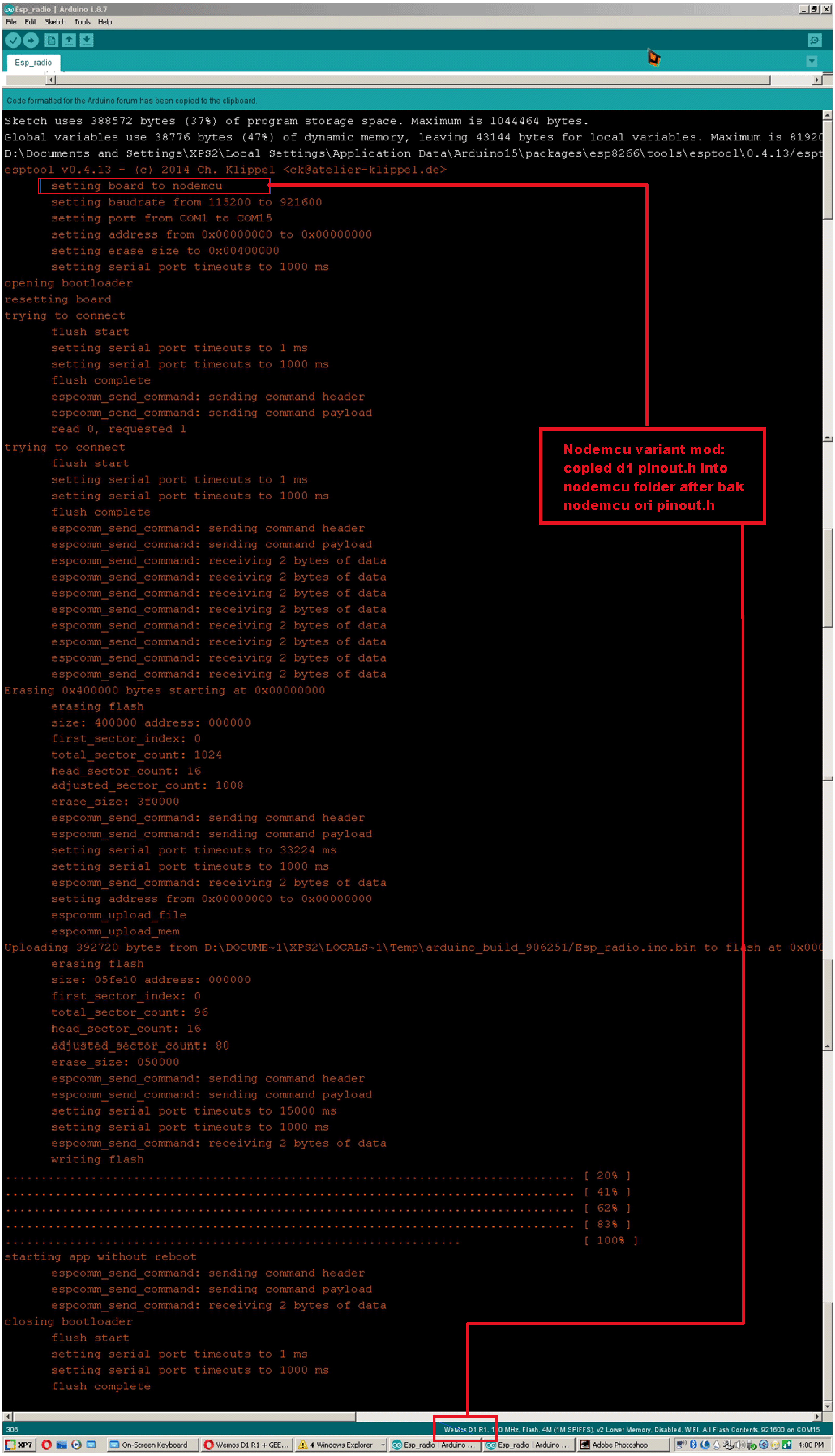
VS1053B Hard Solder Jumper Setup


0
1
6
7
Alternate Module Mod for #Edzelf / Esp8266 WebRadio VS1VS1053B+WeMos D1 R1 S&G™ Version



```
Tools   Help
Auto Format                          Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries...                  Ctrl+Shi
Serial Monitor                       Ctrl+Shi
Serial Plotter                       Ctrl+Shi
WiFi101 Firmware Updater
ESP8266 Sketch Data Upload
Board: "WeMos D1 R1"
Upload Speed: "921600"
Flash Size: "4M (1M SPIFFS)"
CPU Frequency: "160 MHz"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Higher Bandwidth"
vTables: "Flash"
Erase Flash: "Only Sketch"
Port: "COM15"
Get Board Info
Programmer: "AVR ISP"
Burn Bootloader
```

Common Tool Selections



Code formatted for the Arduino forum has been copied to the clipboard.

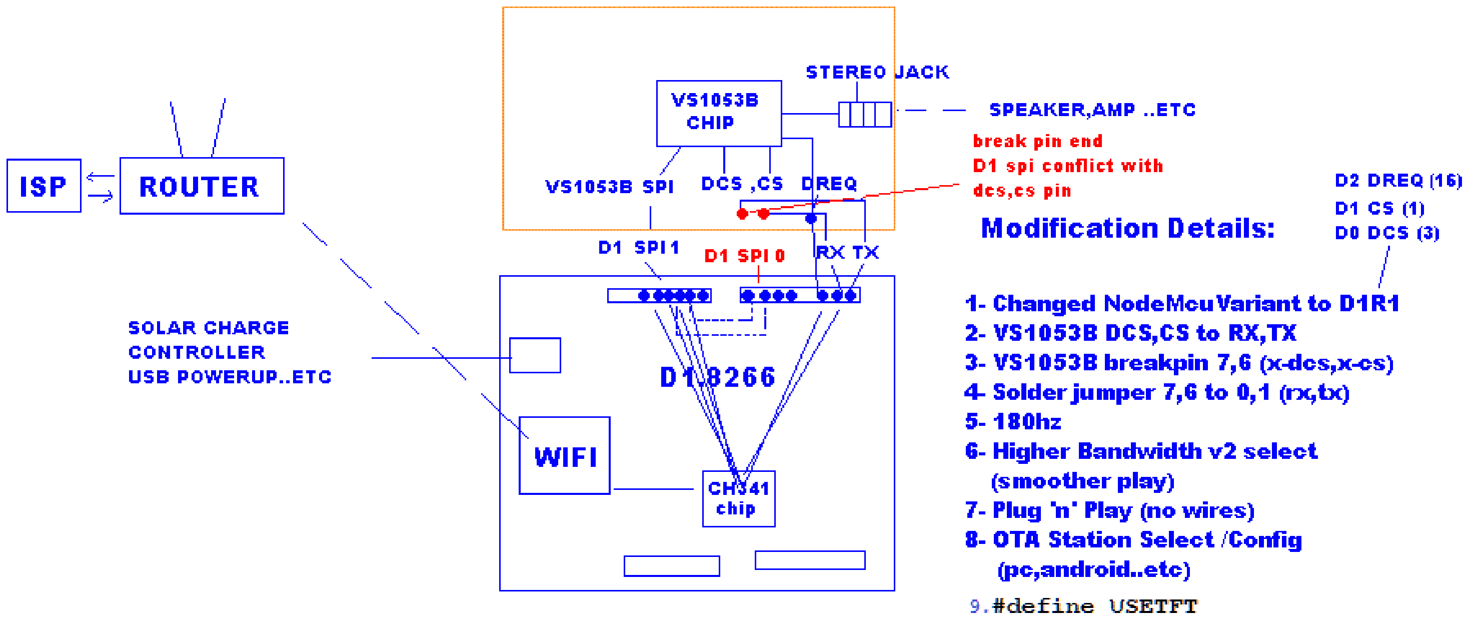
```
Sketch uses 388572 bytes (37%) of program storage space. Maximum is 1044464 bytes.
Global variables use 38776 bytes (47%) of dynamic memory, leaving 43144 bytes for local variables. Maximum is 81920
D:\Documents and Settings\XPS2\Local Settings\Application Data\Arduino15\packages\esp8266\tools\esptool\0.4.13/espt
esptool v0.4.13 - (c) 2014 Ch. Klippel <ck@atelier-klippel.de>
      setting board to nodemcu
      setting baudrate from 115200 to 921600
      setting port from COM1 to COM15
      setting address from 0x00000000 to 0x00000000
      setting erase size to 0x00400000
      setting serial port timeouts to 1000 ms
opening bootloader
resetting board
trying to connect
      flush start
      setting serial port timeouts to 1 ms
      setting serial port timeouts to 1000 ms
      flush complete
      espcomm_send_command: sending command header
      espcomm_send_command: sending command payload
      read 0, requested 1
trying to connect
      flush start
      setting serial port timeouts to 1 ms
      setting serial port timeouts to 1000 ms
      flush complete
      espcomm_send_command: sending command header
      espcomm_send_command: sending command payload
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
      espcomm_send_command: receiving 2 bytes of data
Erasing 0x400000 bytes starting at 0x00000000
      erasing flash
      size: 400000 address: 000000
      first_sector_index: 0
      total_sector_count: 1024
      head_sector_count: 16
      adjusted_sector_count: 1008
      erase_size: 3f0000
      espcomm_send_command: sending command header
      espcomm_send_command: sending command payload
      setting serial port timeouts to 33224 ms
      setting serial port timeouts to 1000 ms
      espcomm_send_command: receiving 2 bytes of data
      setting address from 0x00000000 to 0x00000000
      espcomm_upload_file
      espcomm_upload_mem
Uploading 392720 bytes from D:\DOCUME~1\XPS2\LOCALS~1\Temp\arduino_build_906251/Esp_radio.ino.bin to flash at 0x000
      erasing flash
      size: 05fe10 address: 000000
      first_sector_index: 0
      total_sector_count: 96
      head_sector_count: 16
      adjusted_sector_count: 80
      erase_size: 050000
      espcomm_send_command: sending command header
      espcomm_send_command: sending command payload
      setting serial port timeouts to 15000 ms
      setting serial port timeouts to 1000 ms
      espcomm_send_command: receiving 2 bytes of data
      writing flash
................................................................................ [ 20% ]
................................................................................ [ 41% ]
................................................................................ [ 62% ]
................................................................................ [ 83% ]
................................................................                 [ 100% ]
starting app without reboot
      espcomm_send_command: sending command header
      espcomm_send_command: sending command payload
      espcomm_send_command: receiving 2 bytes of data
closing bootloader
      flush start
      setting serial port timeouts to 1 ms
      setting serial port timeouts to 1000 ms
      flush complete
```

Nodemcu variant mod: copied d1 pinout.h into nodemcu folder after bak nodemcu ori pinout.h

306   Wemos D1 R1, 160 MHz, Flash, 4M (1M SPIFFS), v2 Lower Memory, Disabled, WIFI, All Flash Contents, 921600 on COM15

Edzelfs ESP8266 Alt Mod WebRadio Flowchart

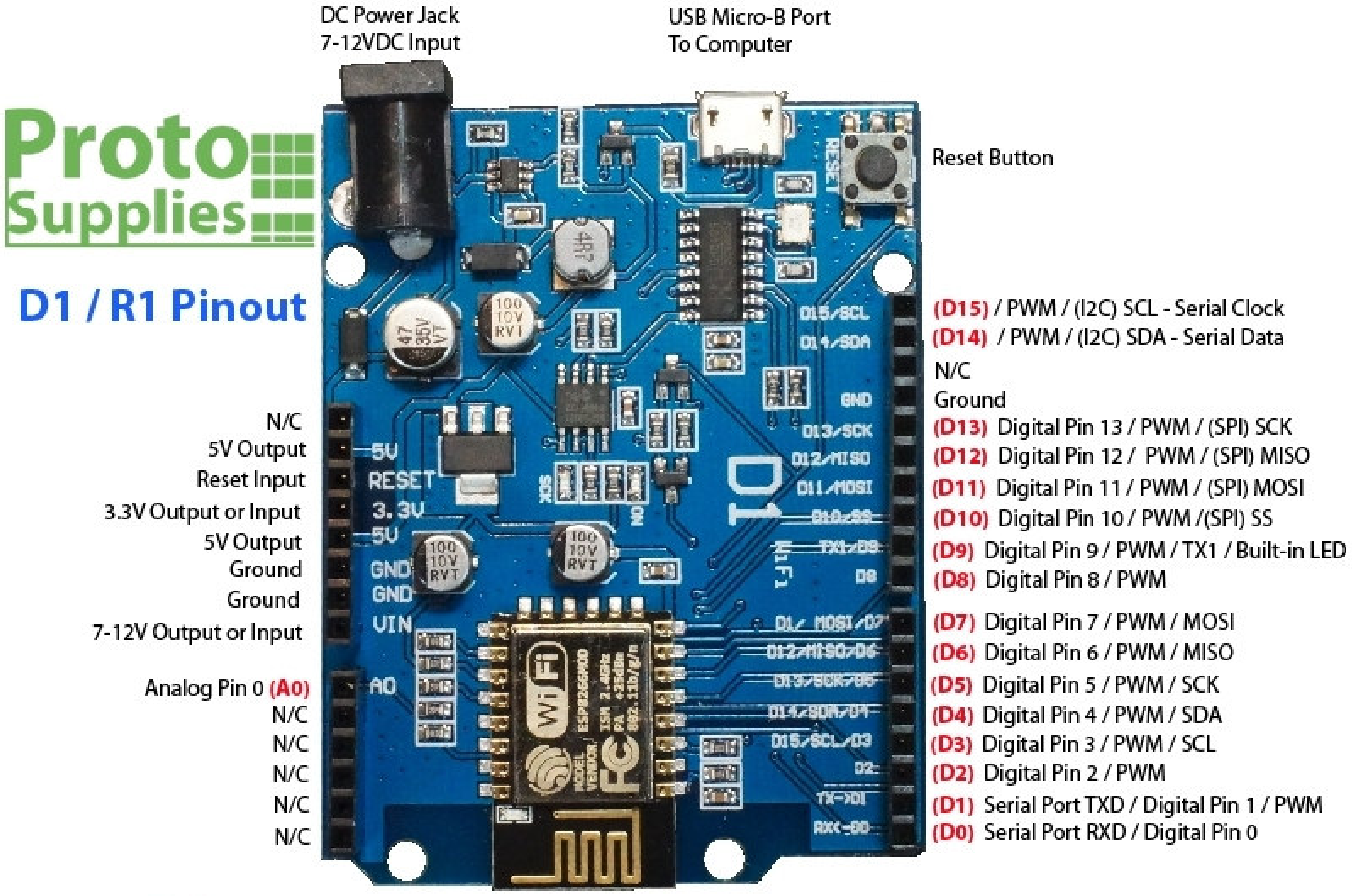
(No Wires Version Mod,Plug,Upload n Play)



Modification Details:

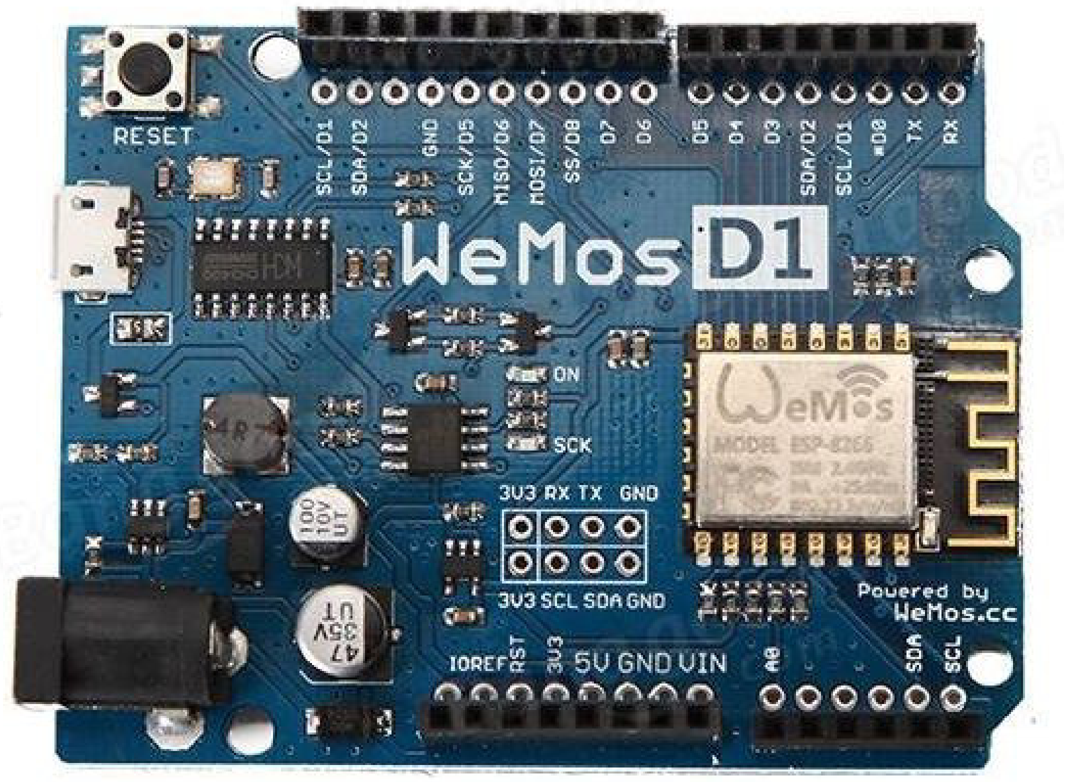
1- Changed NodeMcu Variant to D1R1
2- VS1053B DCS,CS to RX,TX
3- VS1053B breakpin 7,6 (x-dcs,x-cs)
4- Solder jumper 7,6 to 0,1 (rx,tx)
5- 180hz
6- Higher Bandwidth v2 select
(smoother play)
7- Plug 'n' Play (no wires)
8- OTA Station Select /Config
(pc,android..etc)
9.#define USETFT

Modded by https://srotogargees.business.site/


DC Power Jack
7-12VDC Input
USB Micro-B Port
To Computer
Reset Button
Proto
Supplies
D1 / R1 Pinout
N/C
5V Output
Reset Input
3.3V Output or Input
5V Output
Ground
Ground
7-12V Output or Input
Analog Pin 0 (A0)
N/C
N/C
N/C
N/C
N/C
(D15) / PWM / (I2C) SCL - Serial Clock
(D14) / PWM / (I2C) SDA - Serial Data
N/C
Ground
(D13) Digital Pin 13 / PWM / (SPI) SCK
(D12) Digital Pin 12 / PWM / (SPI) MISO
(D11) Digital Pin 11 / PWM / (SPI) MOSI
(D10) Digital Pin 10 / PWM /(SPI) SS
(D9) Digital Pin 9 / PWM / TX1 / Built-in LED
(D8) Digital Pin 8 / PWM
(D7) Digital Pin 7 / PWM / MOSI
(D6) Digital Pin 6 / PWM / MISO
(D5) Digital Pin 5 / PWM / SCK
(D4) Digital Pin 4 / PWM / SDA
(D3) Digital Pin 3 / PWM / SCL
(D2) Digital Pin 2 / PWM
(D1) Serial Port TXD / Digital Pin 1 / PWM
(D0) Serial Port RXD / Digital Pin 0

Red numbers in paranthesis are the name to use when referencing that pin.

WeMos.8266_D1 R1



WeMos.8266_D1 R2

5V	
RST	
3V3	
5V	
GND	
GND	
VIN	
A0	

WEMOS
D1 R1

D15	SCL	GPIO5			
D14	SDA	GPIO4			
GND					
D13	SCK	GPIO14			
D12	MISO	GPIO12			
D11	MOSI	GPIO13	Rx0*		
D10	SS	GPIO15	Tx0*	10K Pull down	
D9	TX1	GPIO2		10K Pull up	Built in led
D8		GPIO0		10K Pull up	
D7	MOSI	GPIO13			
D6	MISO	GPIO12			
D5	SCK	GPIO14			
D4	SDA	GPIO4			
D3	SLC	GPIO5			
D2		GPIO16			
D1	Tx	GPIO1			
D0	Rx	GPIO3			

	IOREF
	IOREF
	RST
	3V3
	5V
	GND
	GND
	VIN
	A0
GPIO4	SDA
GPIO5	SCL

WEMOS
D1 R2

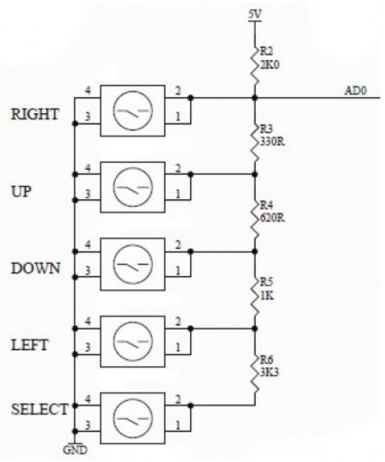
NodeMCU V0.9
Compatible

D1	SCL	GPIO5			
D2	SDA	GPIO4			
GND					
D5	SCK	GPIO14			
D6	MISO	GPIO12			
D7	MOSI	GPIO13	Rx0*		
D8	SS	GPIO15	Tx0*	10K Pull down	
D7	MOSI	GPIO13	Rx0*		
D6	MISO	GPIO12			
D5	SCK	GPIO14			
D4		GPIO2	Tx1	10K Pull up	Built in led
D3		GPIO0		10K Pull up	
D2	SDA	GPIO4			
D1	SLC	GPIO5			
D0		GPIO16			
Tx		GPIO1			
Rx		GPIO3			

COLLECTED ARDUINO PINOUT REFERENCE

UNO	~ D1R1	VS1053B	~ R32	1602 BTN LCD	2.4' TFT	LYRA
19/A5	R1 D15 5 SCL	-	22	A5 SCL		
18/A4	R2 D14 4 SDA		21	A4 SDA		
AREF	-	15 AREF	RST	AREF		
GND	GND	14 GND	GND	GND		
13	R3 D13 14 SCK >	13 SCK >	18	-	SD_SCK	
12	R4 D12 12 MISO <	12 MISO <	19		SD_D0	
~11	SPI R5 D11 13 MOSI >	11 MOSI >	SPI 23		SD_D1	
~10	D10 15 SS >	10	5	BACKLIT ! OUTPUT ASSIGNED ONLY	SD_SS	
~ 9	D9 2	9 TF/SD-CS	13	ENA	LCD_D1	
8	D8 0	8 X-RESET	12	RS	LCD_D0	
7	R5 D7 13 MOSI >	≠ 7 X-DCS	14	DB7	LCD_07	
~6	R4 D6 12 MISO <	≠ 6 X-CS	27	DB6	LCD_06	
~5	R3 D5 14 SCK >	5	16	DB5	LCD_05	
4	R2 D4 4 SDA	4	17	DB4	LCD_04	
~3	R1 D3 5 SCL	3	25		LCD_03	
2	D2 16	2 DREQ	26		LCD_02	
1 <TX	D1 1 <TX	1	TX 1			
0 >RX	D0 3 >RX	0	RX 3			
A0	17		2	←BTN→ › SELECT	CD_RD	
A1			4		CD_WR	
A2			35 /15		CD_RS	
A3			34 /33		CD_CS	
A4			36 /32		CD_RST	
A5			39			

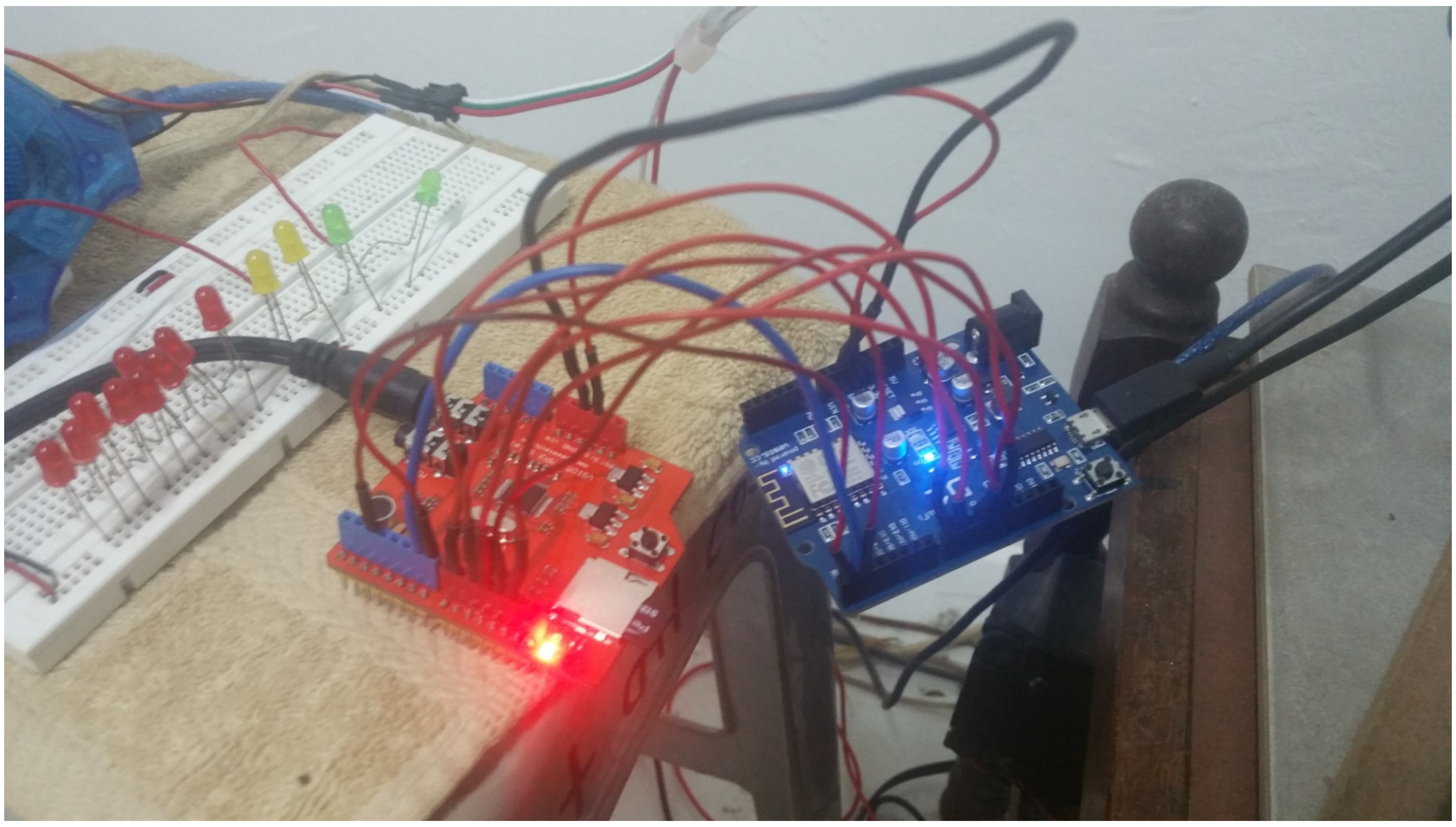
MODULE POWERUP
IOREF
RESET
3.3V
5V
GND
GND
VIN<

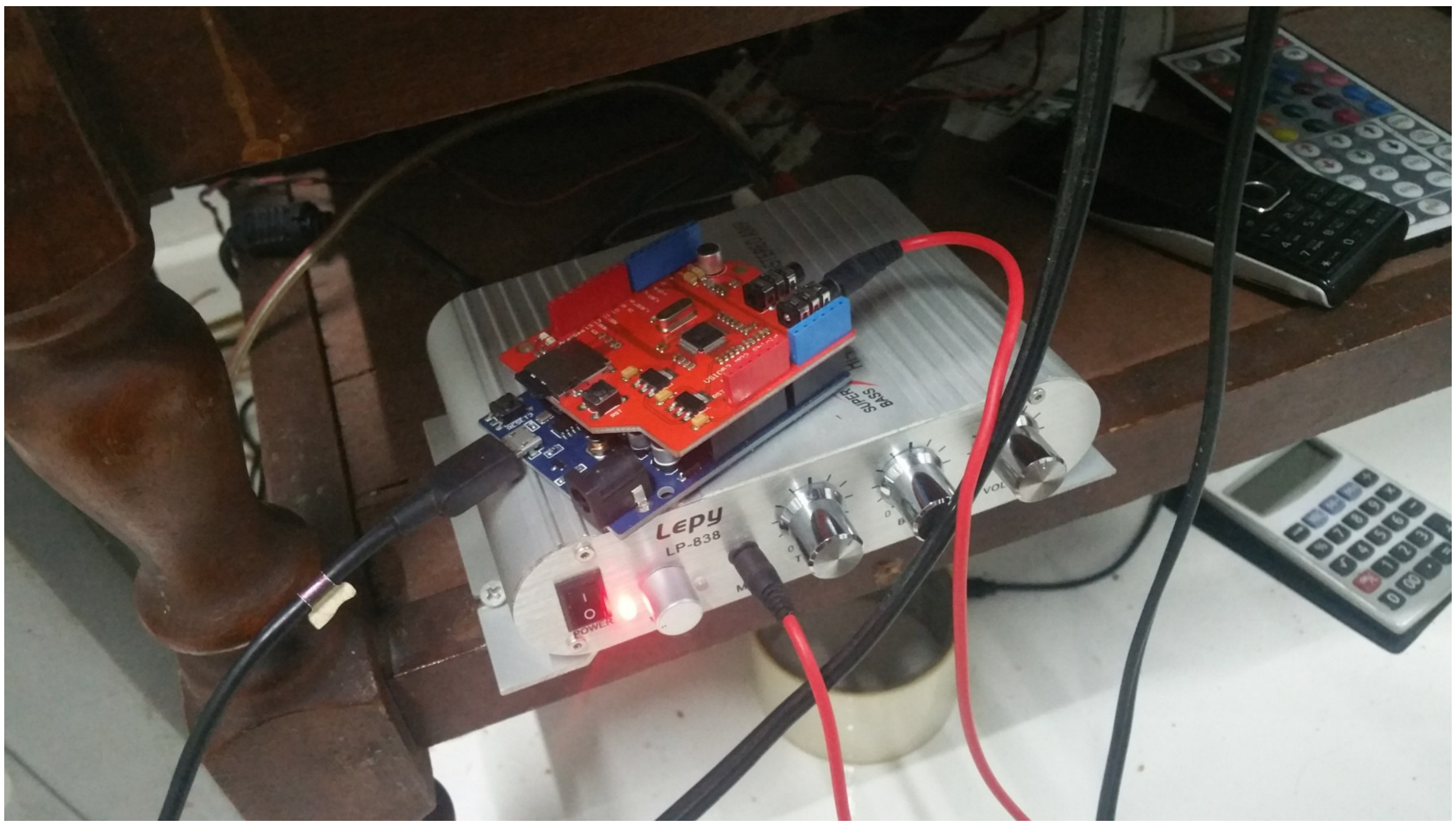


ISOTERICS:

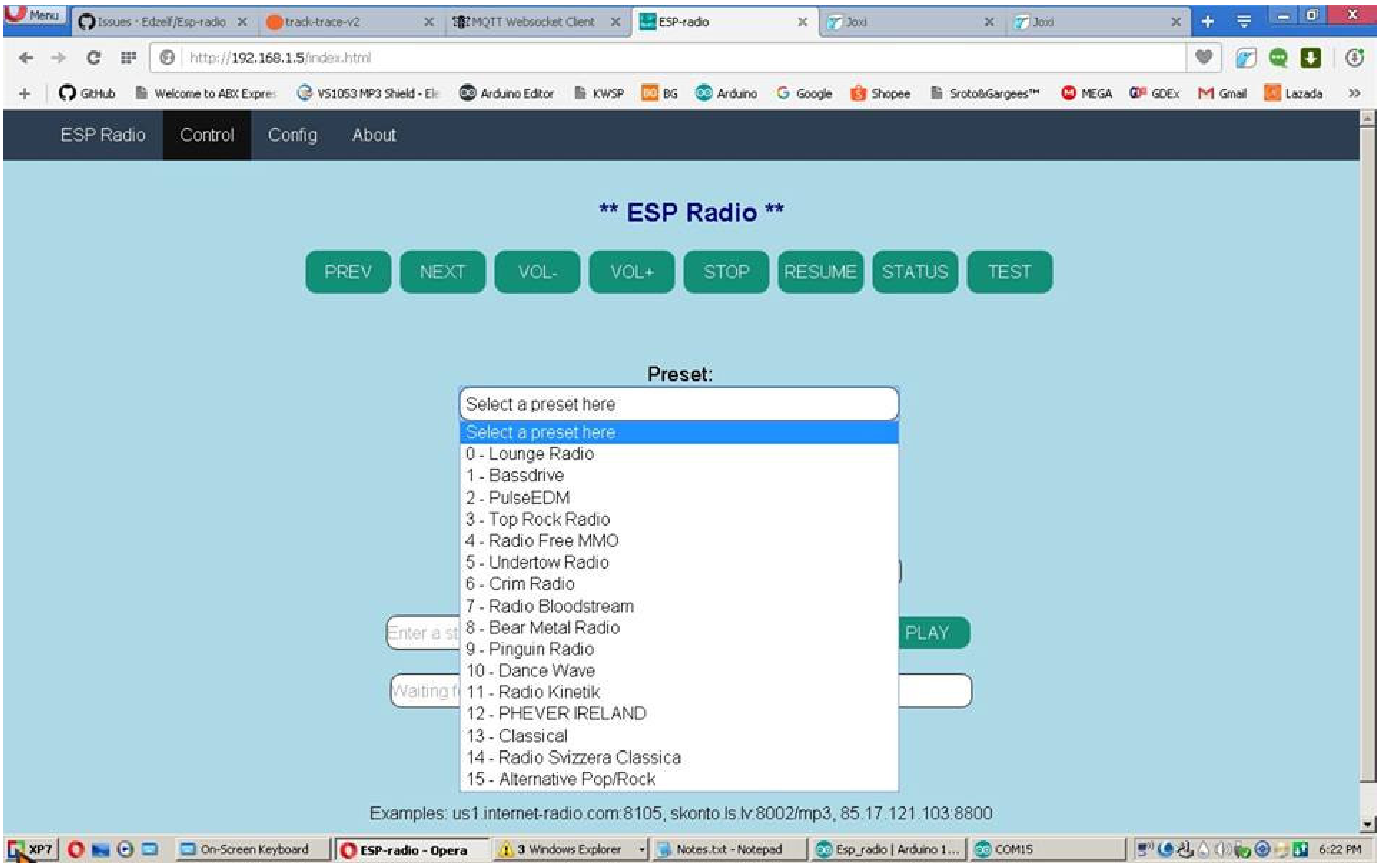
~ = Variable Output Pin (PWM) Pulse Width Modulation
(Fade fx not just On or Offs)
AREF = Analog Reference
SPI = Serial Peripheral Interface
- SCK - Clock signal from master to slave
- MISO - Master In Slave Out
- MOSI - Master Out Slave In
- SS - Slave Select signal selects slave devices

GND = DC Negative (-)
SCL = Analog Serial Clock Link
SDA = Analog Serial Data
A = Analog (Pure Sine/Half Sine/Rectified Sine...etc)
TX = Serial Transmit (byte)
RX = Serial Recieve (1byte=8bit)




Lepy
LP-838
SUPER BASS
POWER

SOLAR CHARGE CONTROLLER
RESETTABLE
COMBINATION PADLOCK
4.30



** ESP Radio **

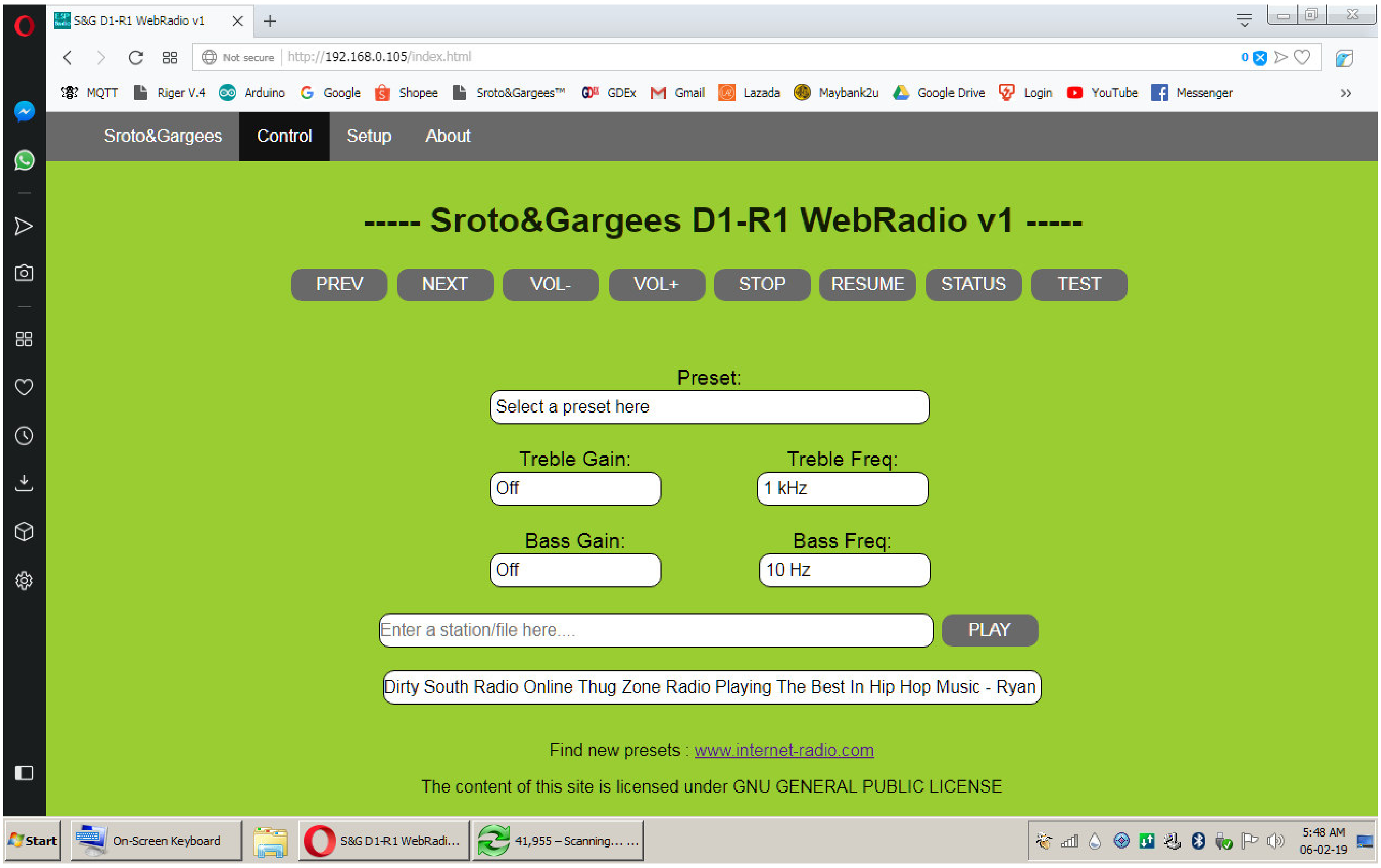
Preset:

Select a preset here

- 0 - Lounge Radio
- 1 - Bassdrive
- 2 - PulseEDM
- 3 - Top Rock Radio
- 4 - Radio Free MMO
- 5 - Undertow Radio
- 6 - Crim Radio
- 7 - Radio Bloodstream
- 8 - Bear Metal Radio
- 9 - Pinguin Radio
- 10 - Dance Wave
- 11 - Radio Kinetik
- 12 - PHEVER IRELAND
- 13 - Classical
- 14 - Radio Svizzera Classica
- 15 - Alternative Pop/Rock

Examples: us1.internet-radio.com:8105, skonto.ls.lv:8002/mp3, 85.17.121.103:8800

Edzelf Esp-radio Default Web Interface with Fav preset Addition via radio.ini



----- Sroto&Gargees D1-R1 WebRadio v1 -----

Preset: Select a preset here

Treble Gain: Off

Treble Freq: 1 kHz

Bass Gain: Off

Bass Freq: 10 Hz

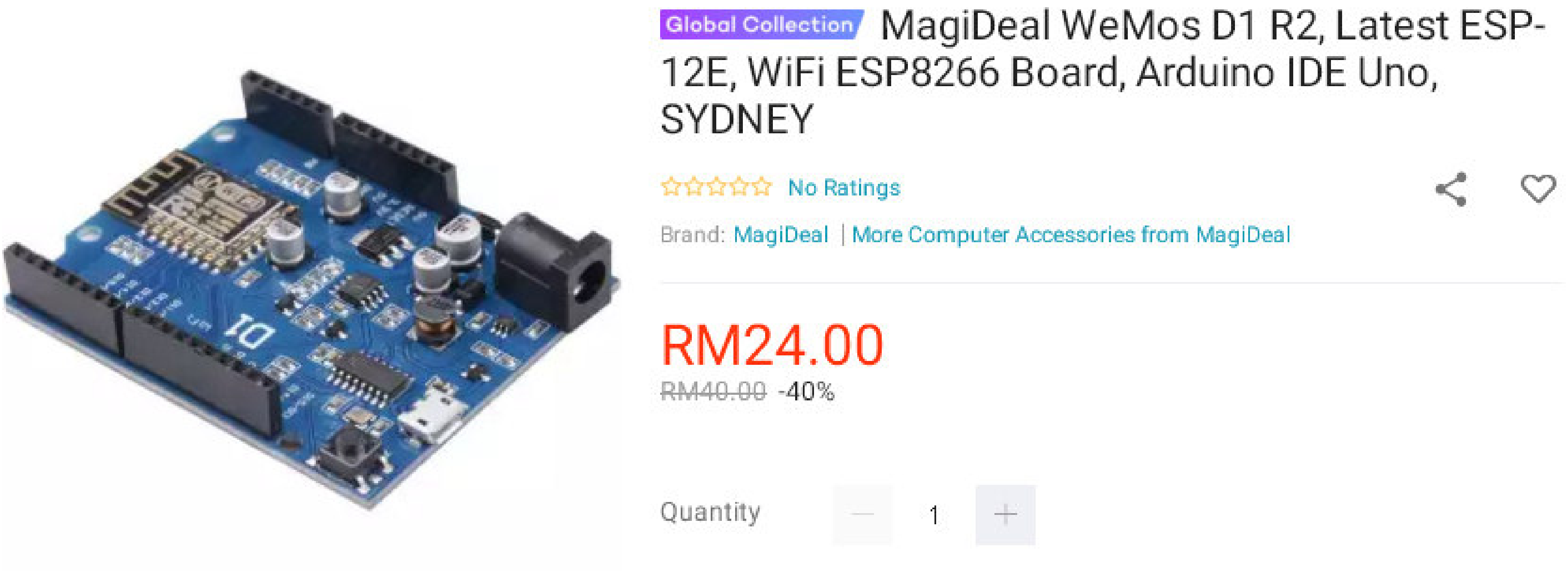
Enter a station/file here....

Dirty South Radio Online Thug Zone Radio Playing The Best In Hip Hop Music - Ryan

Find new presets : www.internet-radio.com

The content of this site is licensed under GNU GENERAL PUBLIC LICENSE

Customizations Example 1



Global Collection MagiDeal WeMos D1 R2, Latest ESP-12E, WiFi ESP8266 Board, Arduino IDE Uno, SYDNEY

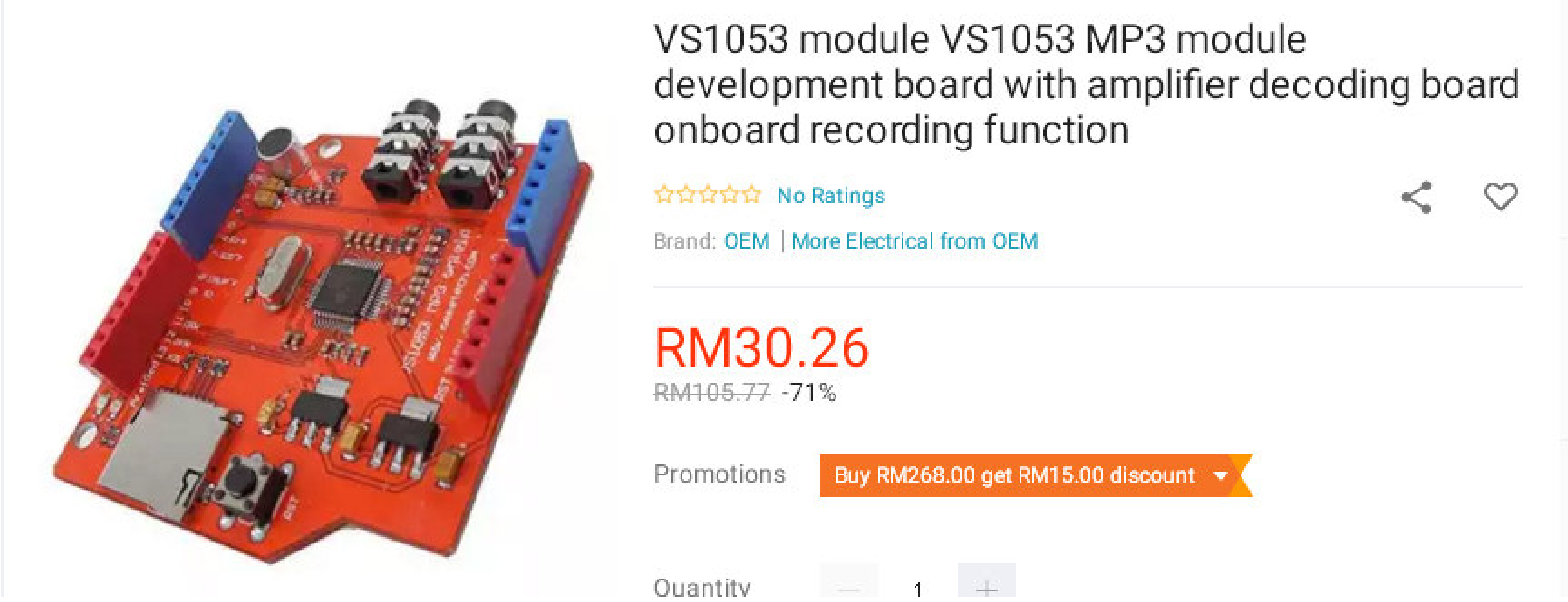
☆☆☆☆☆ No Ratings

Brand: MagiDeal | More Computer Accessories from MagiDeal

RM24.00

~~RM40.00~~ -40%

Quantity − 1 +



VS1053 module VS1053 MP3 module development board with amplifier decoding board onboard recording function

☆☆☆☆☆ No Ratings

Brand: OEM | More Electrical from OEM

RM30.26

~~RM105.77~~ -71%

Promotions Buy RM268.00 get RM15.00 discount ▾

Quantity − 1 +

Project Process 1 VS1053B Prep (HARDWARE)

1. Hard soldering jumpers vs1053b pin_vs_7 to pin_vs_0(RX=D0=3) & pin 6 to 19(TX=D1=1)… and breaking both 7 & 6 pin on vs1053b with 2 wire underneath vs1053b.Multimeter check continuity and solder errors.
2. Plug shield to D1 R1.
3. Ok next is the Jedi tricking D1R1 as nodemcu by replacing arduino-pin.h of d1 into nodemcu variant folder, find it challenge, saber up younglings..backup existing .h incase u need it later… the sketch points to nodemcu pin when its in the flash mode since its not compatible with D1R1 pin out on D1R2 maybe…that’s why I still choose D1R1 since I had a chance for the sd on pin D9 to play local host if I had a chance to learn to code and add into later…or a new D1R3 instead..(Or is it a r2d2..) instead without any redundant SPI pin that still existed on R2 on pin_D9 for SD/TF play on vs1053b to exist without any sacrificial pin breaking in the future….

Project Process 2 (SOFTWARE)

1. Install Arduino IDE.
2. Install stable ep8266 platform (Google it)
3. Install board driver (CH341SER)
4. Get proper micro usb data cable for board most hp battery charger wont work,plugin to pc.
5. Download esp8266 flasher (nodemcu-flasher-master)
6. Flash board with latest micropython.bin (Google it)
7. Run Arduino IDE ,select your board and its port listed in the device manager and open the serial monitor, Reset board and check for any response.
8. If garbled text select proper speed and re-reset until something readable.
9. Check- get device info,--normally if something is displayed than its workable typically (unknown board ,vid,pid.etc)
10. Run a simple basic built-in LED blink sketch from examples. Later modify that sketch and run for better algorithm understanding if you want.
11. Google Edzelf esp-radio download zip file from github.Unzip and load on IDE ,read documentation supplied get some basic clues. Open esp-radio.ino read it. Install all required libraries claimed by the sketch #includes, lessens upload error.
12. Find esp-radio.ino, 3 html.h & a css.h file, copy it into a new folder for it will not run with other folder around (data/radio.ini) also for a backup reason or Option 2- copy just, 3 html.h & a radio.css.h into tinyxml folder in the document library or similar. Customizations is possible on its title, headers and color other are not recommended unless posses advance xml knowledge.
13. Typically for esp-radio initial upload=Under tool Select proper listed Board (WeMos d1 r1), Upload speed 11500, flash size 4mb 1mb spiff, 160mhz cpu, iwIp v2 higher bandwidth, vtable flash, erase all flash on initial uploads, later radio.ini will be uploaded via web interface by connecting to Broadcasted AP to spiff and select sketch upload only if got modification via OTA port or serial as you wish. Try different options and try to understand what it is.
14. Personally it took me a while until I got the algorithm right it was just this 2 pin X-DCS & X-CS PIN redefinition after redefinition …slept off drowning my frustration and continued with persistence armor the next day until I finally found the answers for the source of the bug by trial fritzzing it…separated and powered by wire and found why it was not working in the 1st place…it was the redundant SPI conflict so I had a action plan to make it work plugged in by breaking 2 vs pin and applying a hard soldered jumper.

15. THE ALGORITHM THAT WORKED

16. #define USETFT // Choppier without also tiny wifi LED indication

17. #define VS1053_DCS 3 // VS_X-DCS_PIN-7 JUMPER RX=D0=3

18. #define VS1053_CS 1 // VS_X-CS_PIN-6 JUMPER TX=D1=1

19. #define VS1053_DREQ 16 // D3

20. Finally Upload sketch, try to debug yourself if upload errored, Google it, youtube it, read closed and open issues on Github 1st avoiding escalating same resolved published issues..Etc. Get use to errors …it's a free software so be tolerant. Boost your self confidence by finding its solutions, believe in yourself .Believe me I had second thoughts when it errored a few time on me by choosing an alternate module than the originally proposed Nodemcu since the vs1053b is made for Uno and mega but I insisted to give it a try not knowing anything about it when I online shopped it with my limited budget but I made it to work on d1 r1 too by luck and persistence….i read VS1053B datasheets but nothing in it was newbie friendly.

21. Ok..if you got it right…everything is READY to rock

22. Prep preset playlist in radio.ini

23. Browse for your fav presets and add into radio.ini along with your ssid/password…save it to desktop or anywhere you want.

24. Scan for AP with esp-radio, pw is the AP name.

25. Open esp-radio web interface on browser with 192.168.4.1

26. Select Config page…find & upload radio.ini from there restart, plug in headphone, amp, speakers..Etc... if ini upload success AP are no longer in wifi broadcast..if else do it again ….highly recommend use for unlimited data on broadband or optic ISP … use or not still same charges applies…its audible after a few sec or more…

27. Now find the ip of the esp-radio by looking in the IDE OTA port for web interface preset control….Open browser, key in the routed ip address and there you can OTA select preset and control vs sound equalization.

28. You also have options to upload new modified or update sketch remotely via this link from selecting the IDE OTA port to the now standalone esp-radio server within the same routed network the IDE connected to without serially connecting the microcontroller (esp-radio server) to the pc.

Ps: ..the migration was totally led by d.i.y figure dit-dout challenge , trail & error, online research, interest, basic knowledge ,time ,pain-killer ,frustration handling skills, money & luck … probably a hobby.

-End of Edzelf Esp-radio Sroto&Gargees Version Documentation -