This modification changes a LCD displays SD card slot to a wireless SD card slot. It uses webDAV protocol to manage the SD card inserted into the Displays SD port. For this modification a WeMos D1 Mini was used. https://wiki.wemos.cc/products:d1:d1_mini . This is an inexpensive small board that has the programmer built in. Both the original and the many clones of it sold on sites like eBay will work for this project. The D1 Mini used for this modification had the original circuit board antenna removed and a SMD antenna jack mounted in its place for longer range. WeMos also has a version of this device with a built in antenna jack. The retired version WeMos D1 Mini Pro https://wiki.wemos.cc/products:retired:d1_mini_pro_v1.1.0 is the better one for this project if you want to use one with a built in external antenna. It is available on many ecommerce sites like eBay. The newer version will work but the size will make it more difficult to mount on the display board.

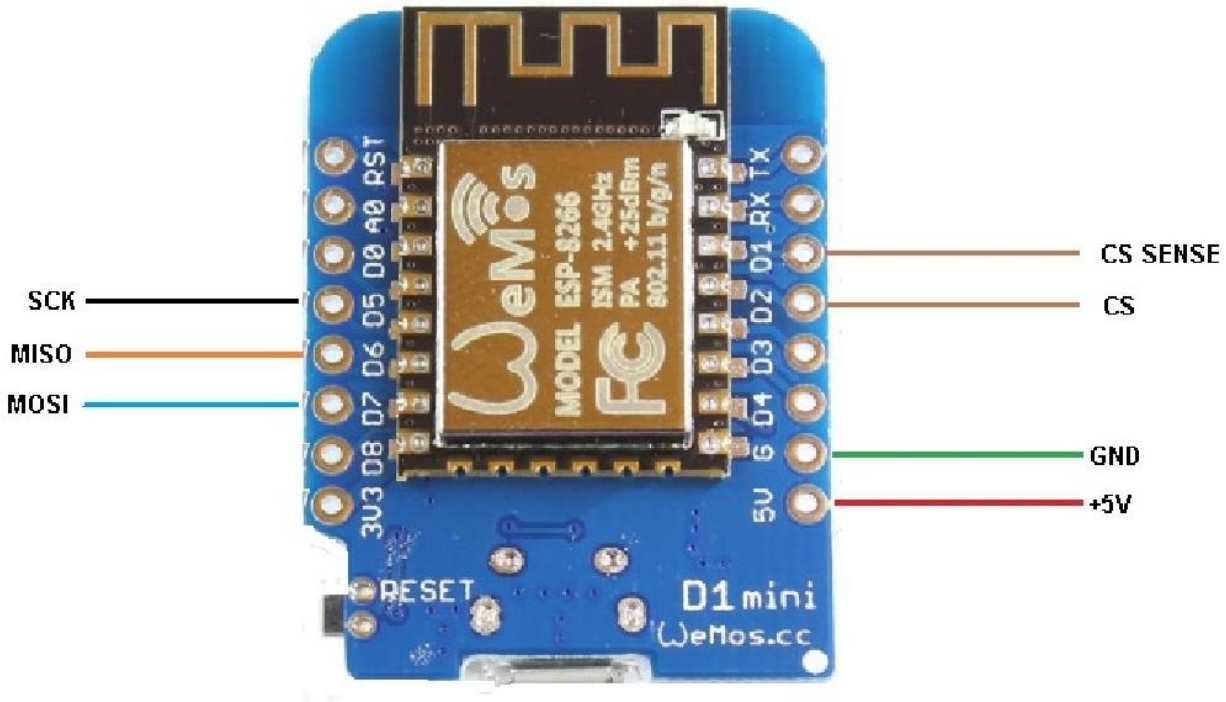
The sketches can be sent to it using the Arduino IDE. The device itself is not based on Arduino but thanks to the folks who developed the ESP8266 Arduino core https://github.com/esp8266/Arduino it is as easy to use as a true Arduino device. There is a lot of information on the internet about programming and setting up the Arduino IDE to use with an ESP8266 based device as well as sending and sending sketches to ESP8266 devices. The WeMos Mini series has the programming circuitry built in so it is easy to send sketches to.

The EspWebDav library located at https://github.com/ardyesp/ESPWebDAV has an example program as well as an example schematic of how to use it for a wifi sd card. The sketch can be compiled but you will need to use an older version of the esp8266 board " 2.4.0" or "2.4.1"and an older version of the SdFat library "1.0.5 to compile it. It will compile but will not run if you try using up to date versions of these two libraries. The two sketches included with this will compile and run using up to date software. If you decide you want to build the device shown in their schematic you will need to connect GPIO15 to GND or it will not work. There are other changes that can help but that one prevents the device from running at all.

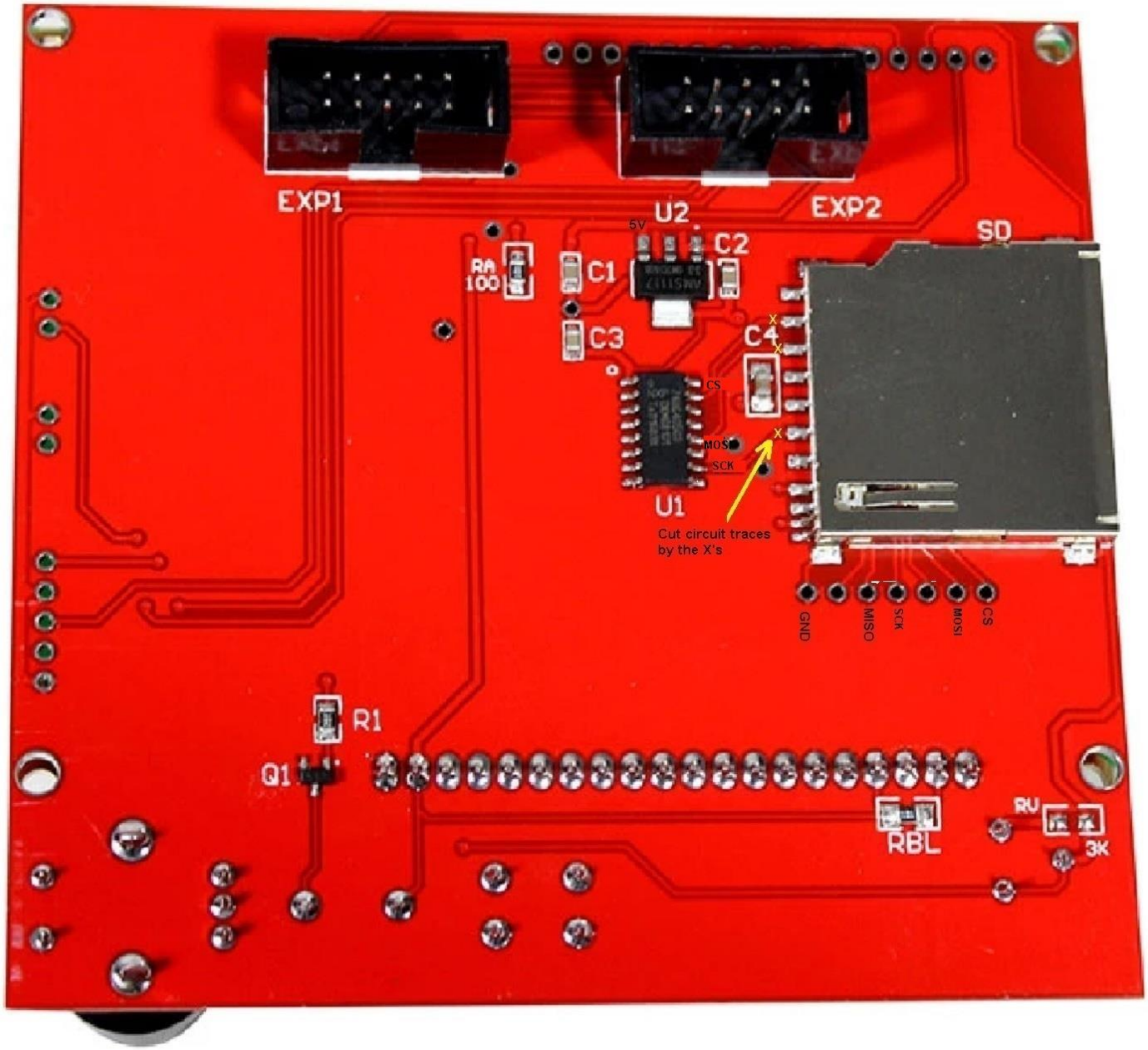
This is a general guide on how to modify a display card so the SD card can be remotely worked on. There are resources available on the internet on how to work with ESP8266 devices, the programming language, soldering skills and so on. It is up to you to make sure you have the skills needed and also to make sure this will actually work. But for many it should be enough information to assist them with a very useful project.

The display used is made by Bigtree-Tech.com and sold by them and others. It has the connections bought out underneath the Sd card slot which makes it easier to work with. The real RepRapDiscount devices as well as the one sold by Geeetech are not as easy to use for this. You will need to unsolder the display from the circuit board to access some of the traces needed if you use one of them. There are other Display cards that will work for this as well as control boards with SD card ports. As long as you can separate MOSI, CS and SCK between the SD card port and the board you should be able to use the schematic to modify your device.

Here is the WeMos D1 Mini used for the interface between the cards SD port and WiFi. If you are going to be close to the 3D printer you can leave the card as is. I found it worked well inside the house but could not be connected to in the workshop which is about 40 feet away with 2 block walls in the way. It is however easy to add an antenna if it has the type antenna shown. This has an ESP-12F device on it. You can modify the other types but they are not as easy to do. Or you can purchase a WeMos Mini D1 Pro which has an antenna jack built in. There are two version of the Mini Pro. The discontinued one that is sold by a lot of companies on eBay is the better device for this project. It looks identical to the D1 Mini with an antenna jack replacing the circuit board antenna. Try to purchase the Wemos device with nothing soldered into the holes.



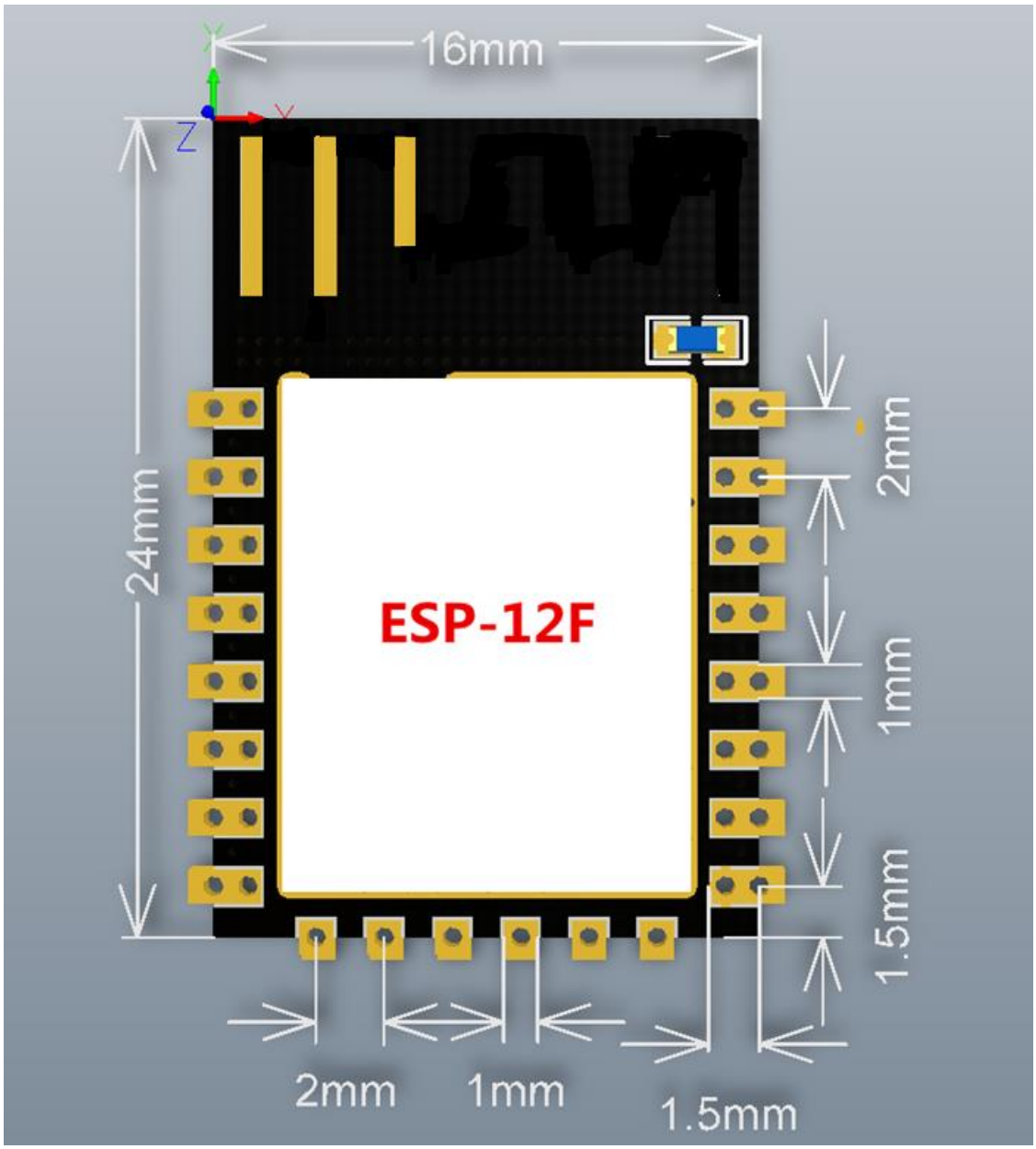
This back panel picture has markings to show where the circuit traces need to be cut as well as the places that are soldered to. An X-acto blade was used to cut the 3 traces and carefully remove a small section of the copper in between the cuts. Since mine lives in a workshop I covered all solder joints and the cut area on the board with clear nail polish. Yes it is not real Conformal coating but it works well enough for this project.



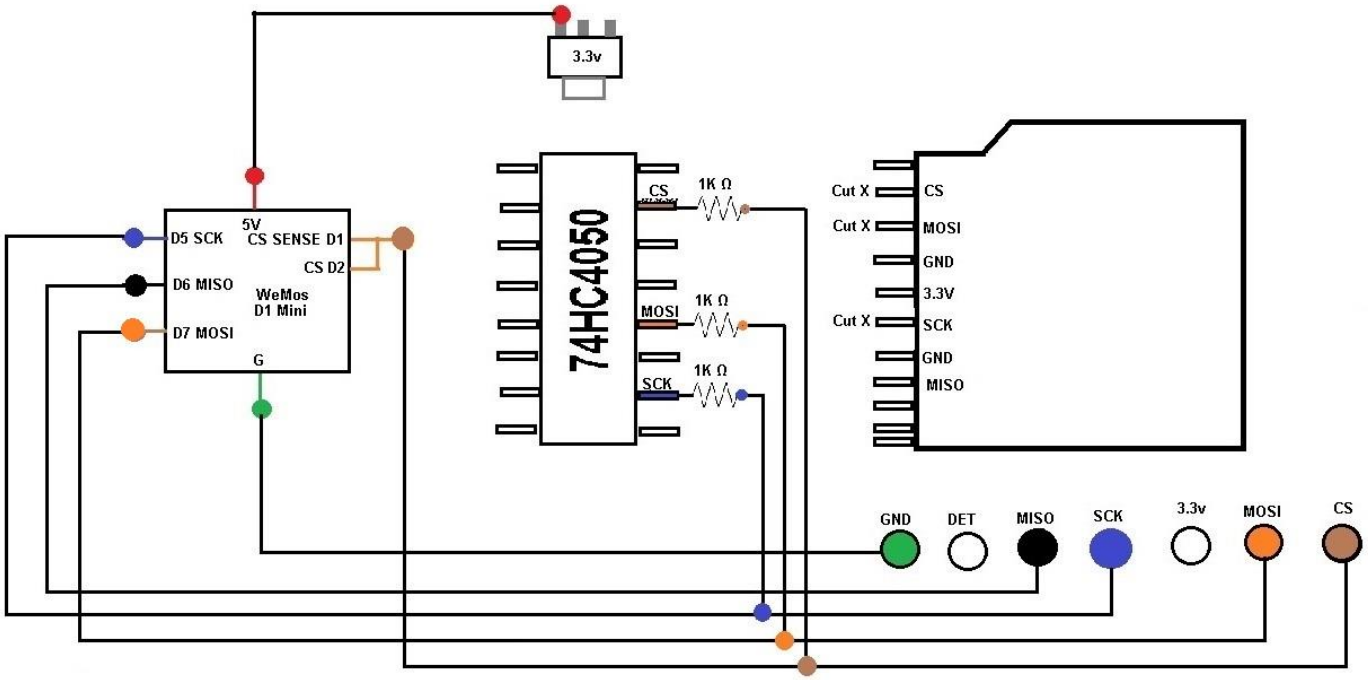
The following shows how to modify an ESP-12F for an External antenna. The Wemos D1 Mini uses one of these.

Here is the device with the pcb antenna traces cut. This leaves the ground to the left, the antenna in the center and a place to solder the jack on the right. Panel mount SMD jacks were used, they fit perfectly on the circuit traces that are left. You may need to grind a small amount of the back pcb if your jack will not fit onto the board. IPX / U.fl connectors also will fit but may not withstand any tugging to remove the antenna cable. If you want you can apply a small amount of epoxy to the rear of the jack to strengthen the mounting.

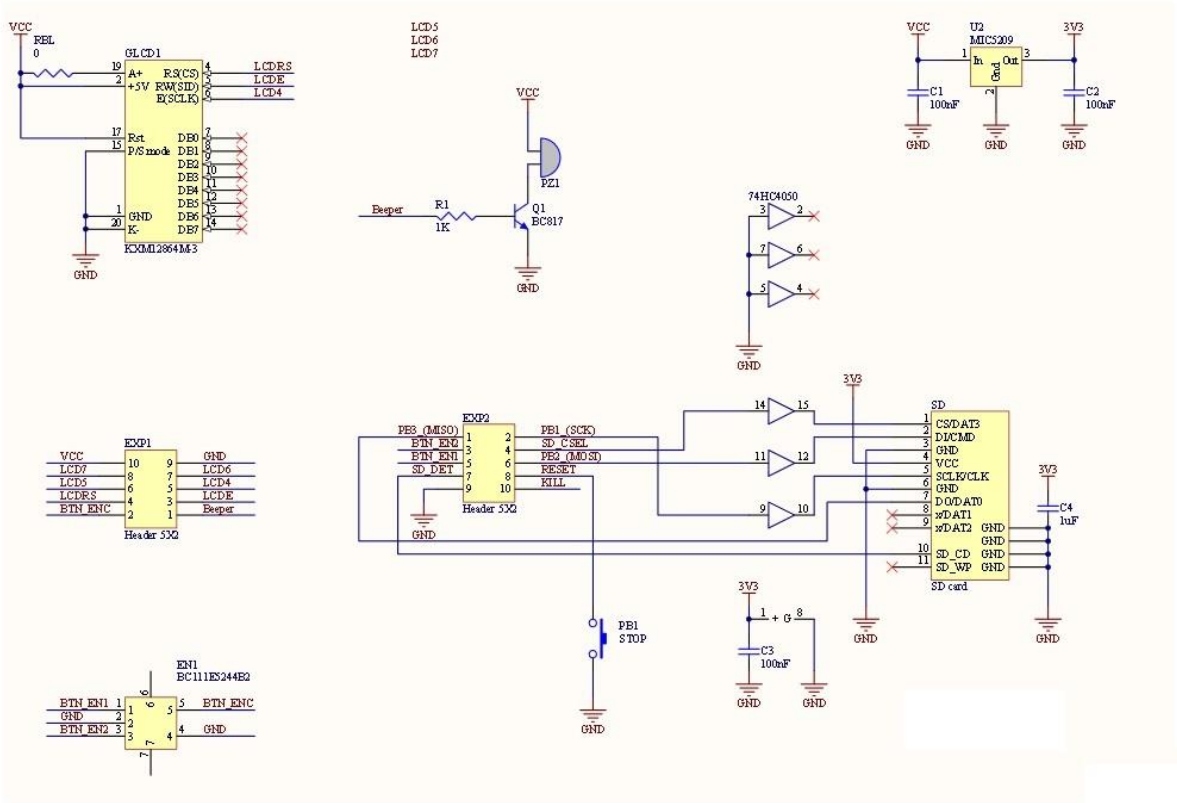
None of this is needed if you buy the WeMos D1 Mini Pro as it has a U.fl connection on the board as well as a ceramic antenna.



Here is the schematic of the modified display and the schematic for the circuit board itself.



Cut the circuit traces for MOSI, CS and SCK at the solder pads for the SD port on the display card.
Solder the G on the WeMos card to the GND and solder the 5V to the upper left lead on the voltage regulator.
Run a lead from D6 on the WeMos card to MISO. On the 74HC4050 device solder 1K olm resistors to the leads and wire the circuit to the schematic.



Included are 2 versions of the WebDAV program for ESP type devices. These are not Arduinos but they can be programmed using the same Arduino program. There is a lot of information out there on how to do this. It uses the ESP8266 board and 2 Libraries, wifiManager for ESP8266 and EspWebDav.

The important settings for the ESP8266 board for a WeMos D1 Mini and D1 Mini Pro are Lolin(WeMos) D1 Rd & mini CPS Frequency 160MHZ Flash size 4M (3M SPIFFS).

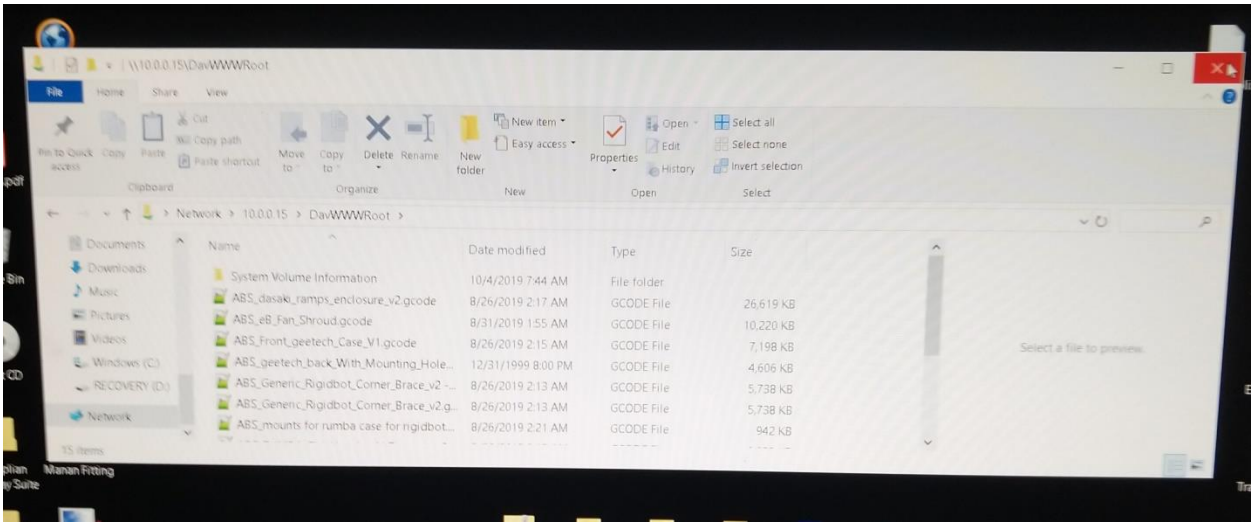
I'm not going to go into how the sketches work as they are self-explanatory,

There are two versions of the sketch, one is wificardino.ino . Another version is called switchedsdcard.ino. This version has a momentary switch between TX (GPIO3) and ground. When it is booting close the switch until the LED lights up. It will be ready to redo the setup at that time.

Make sure to use good SD cards and they have to be formatted in either FAT or FAT32 format. Since it is easy to move files on and off with the device you do not need huge cards . I have a 16GB one and am able to use 32GB SD cards. If the initialization of the SD card fails the LED light will flash rapidly 100 times and the program will not work. Do not use Windows to format the cards, use the program by the SD card association.
https://www.sdcard.org/downloads/formatter/

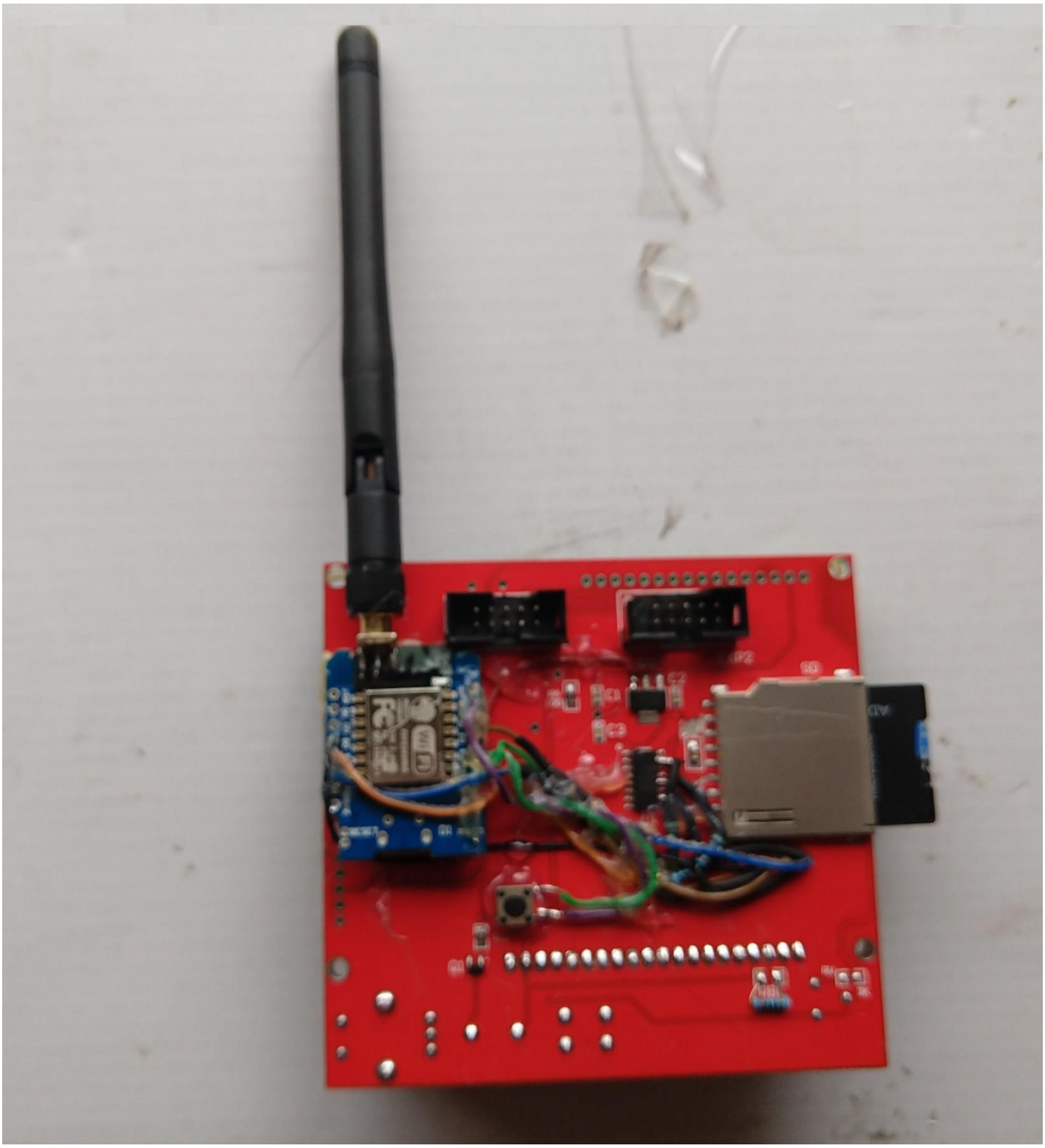
For windows to use the SD card bring up file explorer. Type \\hostname\DavWWWRoot\ or \\ip\DavWWWRoot\ for mine it is \\10.0.0.12\DavWWWRoot\. It will bring up the SD card and you can move, delete and copy files. It also can be used with Apple and Android devices.

If you are using this with ESP3D or Octoprint you just need to refresh the files list to see the changes made.



Do not remove the SD Card while this is in operation. Doing so will screw up the program and you will need to reboot to access it. If you power up without the SD card in the display you will need to power back down so the program can initialize the card. You should be able to do anything needed other than reformatting the card via the WiFi SD card.

Here is the finished board. It’s not pretty but works very well.



The switchedsdcard.ino sketch was used so a small push button was attached to the board. If it is held as the board is powered up the LED will light and the board will start up in access mode with a DNA and after you select it the webserver will start again at address 192.168.4.1. If this was a mistake you can either wait 10 minutes or reboot and it will go back to default. Hopefully you will find this useful.

Troubleshooting.

First, if you are using a SD Card or a SD card adaptor with a MicroSD card make sure the switch on the side is not in lock position.

If you are not able to initialize the SD card and your wiring is correct try using the SD associations formatter at https://www.sdcard.org/downloads/formatter/ to format it. The card needs to be formatted in either FAT or FAT32 format

I found with one card I needed to remove the partition and remake it using MiniTool Partition Wizard then format it with the S Association formatter before it would work. It turned out shortly afterwards that the SanDisk card was not a real card from Sandisk as it continued to have intermittent problems. So back to Amazon it went. Things were much easier after that with a real SD card.

One thing to note is how you wire this matters as far as being able to initialize the SD card. The default speed for the SdFat library is 50MHZ. If you have longer wiring you may need to lower the speed down. This does affect the transfer speed some but your WiFi signal and congestion and such affect it more. The place in the sketches where this can be changed is #define SPI_SPEED SD_SCK_MHZ(50) This was set to 35 while bread boarding the device to test it. It was raised back to 50 MHZ when the display was wired. If yours acts up try lowing the speed to see if it helps.

For mine the board's 5V power was enough to run both the display and SD card Wifi. If you have power problems you might need to remove the 5V lead from the 3.3v regulator and power it separately. Power problems may show up as the display scrambled looking with missing pixels, the printer controller board rebooting or corrupted files when you write to the SD card. When the SD card is being written to is when it draws the most milliamps, for a marginal system this might be enough to cause problems.

Note , per the developer of the EspWebDav library this only works for 3D printers and Arduino based SD card readers. Not all 3D printer controllers can be used for instance Creality V2.0 and V2.1 boards will not work with this. There may be others too that will not work. Rambo, Rumba, Ramps 1.4 boards are known to work and others may. Before you do this you should check your board's schematic to see how it works. Per the developer of the library. SD cards have two modes of operation - a propriety mode and SPI slave mode. The EspWebDav library and 3d printers/Arduino etc. will work in SPI mode. Computers, digital camera's etc. use the propriety mode. It won't work with those devices.

The odds are you will not experience any of these problems. The ones I experienced were a fake SD card, Creality printers not working with SPI mode SD cards, the SPI speed issue while I had this laid out on a breadboard and the SD card being locked.