Automated UPS Shutdown System Using ESP8266

This project uses an ESP8266 to automatically shut down a PC and UPS during a power outage.

How It Works

This project uses an ESP8266 NodeMCU as the brains of the operations and a SG90 Servo as the actuator for the button as I don't want to modify the UPS itself. This code for the ESP8266 is written using the platformio ecosystem.

Power Supply

The system is powered by a single 18650 LiPo battery which is being charged through a TP4056 board. This is ideal because it has a protection board builtin. The battery is charged through a standard 5v charger and the battery is directly connected to the ESP using the VCC and GND pins.

Servo Connection

The servo is connected through the same VCC and GND pins, however I have added a couple of filtering capacitor to help reduce servo jittering. The servo ideally needs 6v to function effectively however we're under powering it which can lead to jitter during load. This is one fundamental flaw of this system as of right now.

Code Structure

All the code is written under `src/main.cpp` file. We are using the D4 pin as the signal pin to move the servo. In the setup function we are running the WiFiManager which is responsible for connecting WiFi once connected we are using the `webserver` function to run the webserver. In the webserver we are calling the `toggle_switch` function which is moving the servo when the button is pressed.

WebServer Function

This function uses the LittleFS library to load the HTML,CSS and the JS files. Although this is overkill for such small functionality but it's nice to have total control over these bits and pieces however remember to go to `Platformio Extension>Platform>Build Filesystem Image` and then `Platformio Extension>Platform>Upload Filesystem Image` to get these files on to the ESP or LittleFS will complain it can't find the files. Guess how I know!

Note:
The HTML, CSS and the JS files along with any other file you may want to use need to go in the `data` folder for it to work.

Dependencies

- ESP8266WebServer

- WiFiManager (*external package*)
- ESP8266Ping (*external package*)
- LittleFS
- Servo

Why Does This Exist?

I recently bought a new computer and a 1500VA UPS, this thing has a cooling fan built into it which runs literally all the time. The fan unfortunately is louder than the new PC and I have to put it in the other room to mitigate the noise. I have to walk all the way around to turn the UPS ON/OFF each time. On top of that I tend to leave the PC on and during outages I might not be close to turn it off. This is how my old UPS batteries died. This is how I came up with this project. The main goal of the project is to quickly turn off the computer and the UPS automatically.

The Master Plan

This might seem like it's a glorified button press but it's not. I want to create a system that will let me do the following.

1. Let me turn the UPS ON/OFF anytime I want.
2. Will be able to detect if the Power is out.
3. If the power is out then it will shutdown the PC and then proceed to turn off the UPS.

So basically this project will be responsible for quickly shutting down the PC safely and then turning the UPS off to save power. This will also allow me to turn the UPS back on when the power comes back (*this can also easily be automated*)

Limitations

This new UPS has a warranty which means we're not allowed to mod it in any way. This means we can't open it up and hook things up directly, instead we must work in a non destructive way so if we ever need to claim the warranty we can without issue.

Development Log

Hi, this project isn't my first ESP project but it's the first one where I decided to use platformio and C++ instead of MicroPython. However I have been working with analogue electronics for quite some time, nothing too smart, just smart enough to get stuff done. Needless to say I had a few shortcomings.

Power Supply

9/13/2025

First thing I realized when I started designing this system was that I needed a constant power supply that didn't depend on either of the power sources ie UPS and mains power. However when I looked things up all the videos said the ESP couldn't be powered using the 18650 LiPo battery, which is true but not if you're using an NodeMCU module which in this case I am. Great!

We now have managed to create a power system that works. We managed to connect to the WiFi and are able to run the servo fine for the time being. We are using 3.8v to 4.2v max but the server requires 6v to function properly. This is problematic because our current setup doesn't provide sufficient power to the servo and can lead to brownouts (*which does occur as of now*)

Creating a Prototype Board

9/14/2025

Since I will be powering the board using a battery and the servo also needed power from the same pins I have two choices, either use a breadboard like a sane person or make a prototype board by soldering headers.

I don't know about you but I for one dislike connecting so many jumpers to get a small thing working. I would rather have a couple of headers for the the servo and the battery and have a few headers for the ESP to connect to. Within 20 minutes I had a working prototype board, this was a solid win.

Servo Trouble & Almost killing the ESP

9/14/2025

I connected the servo and the battery to the new prototype board. The servo was an old hand me down from a friend. I didn't realize anything when i powered it through the battery but the power cable was oddly warm. I was thinking the cable must be crappy, added it to the long list of things to replace. Since I was actively developing the system at this stage, I decided to connect the ESP to the PC, I removed the battery from the board and connected the micro USB.

Not more than 5 seconds later I see smoke and then a glowing red spot. I take the ESP out of the prototype board and plug it in again. It's dead no power. What a shame, I unplugged and investigated it and the component crumbled away it was ash. I looked at the board and I realized that it was a fuse, I got lucky. I decided to take a thin wire and bridge the gap for now as I can't find a replacement fuse where I live and decided to focus on the servo.

I took apart the servo and decided to see the wiring and it was a mess, the wires were crossed which caused the fuse to blow and that also explains the hot battery cable. I had to take the whole assembly out to get the wires back on correctly however that was a recipe for more problems, because now the servo didn't work correctly. Fast forward a few hours I managed to get the thing to work.

Creating The Toggle Button Code

9/15/2025

I wanted to be fancy on this one, instead of using the html css and js in the main file I wanted to be a bit fancy and use actual HTML, CSS and JS files for the frontend to work like normal projects work. This resulted in some issues. I asked ChatGPT so many times but it still couldn't help me figure out why the files wouldn't load. Turns out in the `platformio.ini` file I needed to have `board_build.filesystem =`

`littlefs` for things to work correctly. After adding that and building the filesystem and uploading it, everything worked flawlessly.