1 Overview

This project is a WiFi powered pump for a drip irrigation system for a home garden. This guide includes all the necessary steps and instructions for setting it up yourself, as well as descriptions of how I went about finding all the information I used. My background is in software development, so this project is a completely new experience for me.

1.1 Parts

It took me a while to figure out all the parts I needed for this project. Here's the list, to the best of my ability, of parts that you'll need for the arduino powered circuit.

For the Circuit

- 1 x LinkNode D1 board[1] by LinkSprite
- 2 x 3.3v Solar Panel
- 2 x 1N4007 High Voltage, High Current Rated Diode
- 2 x TP4056 Battery Charger
- 2 x 18560 Rechargable Lithium Ion Battery (and holder)
- 2 x 0.9V-5V to 5V USB DC-DC Booster
- 1 x KY-019 5V One Channel Relay
- 1 x USB-A to Micro USB cable
- 1 x USB-A to leads cable[2].
- 1 x 5V water pump with leads exposed[3]
- Airline aquarium tubing[4]
- 1 x Computer with internet connection and the ability to host a webserver[5]
- Wires

The tricky part of picking your supplies is choosing the right pump and tubing. I chose a 5V pump because the garden beds I'm creating are relatively small, so a pump of that power *should* be able to push the water all the way through the system.

Additionally, you'll find it hard to find drip irrigation tubing that fits a small pump. Drip irrigation tubing usually comes in 1/4in to 1/2in sizes, but if you've got a smally pump you'll have to "roll your own" tubing so to speak. You can do this by making small slits in the tubing at regular intervals; you don't want holes that are too large as that will prevent the water from passing all the way through the tubing. Conversely, you don't want to small of a slit because that will prevent the water from coming out at that particular spot.

If you want or need a larger pump/tubing then you can halve the number of **solar panels**, **diodes**, **batteries**, **voltage boosters**, and **battery chargers**, and you can remove the **USB cable with leads**, but you'll need to be able to supply power to the pump externally, and in a way where you can still pass the power supply through the relay.

For the Garden Beds

-

1.2 Network Overview

Overall, the connections are fairly simple. The ESP8266 boards will need to have access to a router, as well as the credientials necessary for accessing the network. Once an hour, the boards will wake up and make a request to the webserver asking for any commands that are waiting there. If the webserver has any, it will reply with the last command that was given to it. The supported commands are `on`, which tells the arduino to turn on the pump for 5 minutes, and `logs`, which tells the arduino to send back the logs stored on the board. Any other command will be ignored, and the arduino will shut down again once it reaches anything other than `on` or `logs`.

On the opposite side, a user will be able to go to the URL of the webserver, and will be able to issue commands from a simple UI. The commands are sent to the server and stored until the arduino asks for them.

[1] You can use any ESP8266 board for this project, but I found LinkNode boards to be the easier to hit the ground running with.
[2] Essentially, you just need a USB-A cable with the opposite end stripped off.
[3] These pumps are the same ones that are typically used in home fish tanks.
[4] You'll need to make sure that the tube's diameter matches the pump's diameter.
[5] I'm running requests through a Digital Ocean droplet, but something as simple as a Raspberry Pi 1 should do the trick

2 Controller

The Controller is powered by an arduino-knockoff board with an ESP8622 chip already wired in. The board I used is a LinkSprite D1, which uses the same architecture as a Wemos D1 R1 board (hence the D1 configuration in the platformio project), but you can use any ESP8622 board you want provided that it

1. has at least one output pin, and
2. you can make the necessary connection for sleeping.

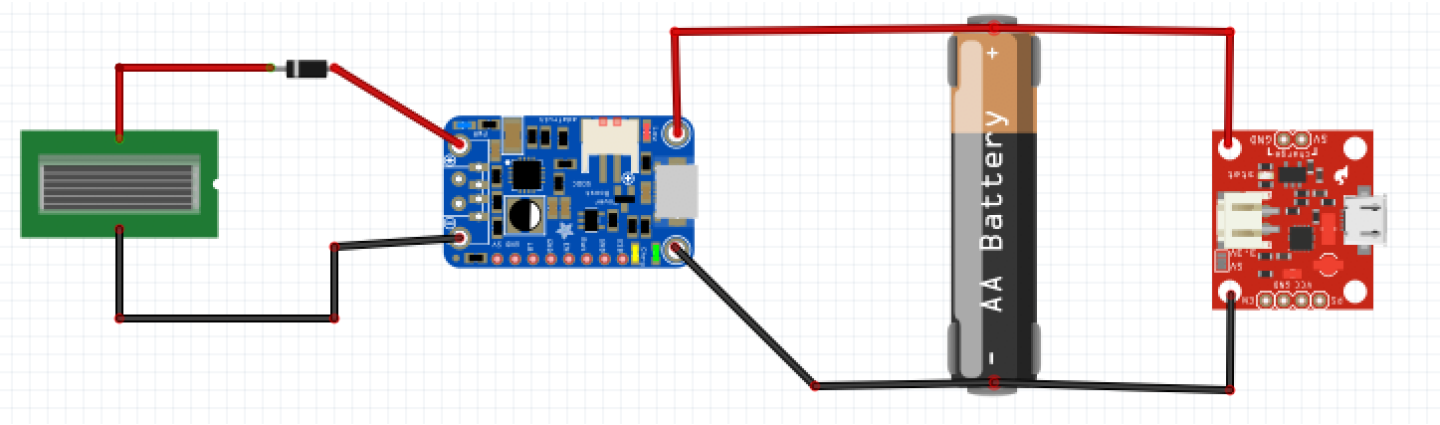
2.1 Power

NOTE: You'll need 2 of these circuits if you're using a small 5V pump.
Unfortuately, the ESP8266 is a power hungry board. Luckily, we have a way to put the board to sleep so that is stops using so much power. We only want the board to check in for a few seconds every hour, and in the worst case the board will be running for 5 minutes maybe twice a day[6]. The rechargable battery circuit is fairly simple, and uses the following items:

- 1 x 3.3V Solar Panel
- 1 x 1N4007 High Voltage, High Current Rated Diode
- 1 x TP4056 Battery Charger
- 1 x 18560 Rechargable Lithium Ion Battery (and holder)
- 1 x 0.9V-5V to 5V USB DC-DC Booster

The solar panel will essentially charge the battery while the circuit is not in use. The diode prevents electricty from flowing back through the solar panel, which is important because we only want the solar panel to provide power to the battery, not the other way around. Since the solar panel provides 3.3V, we use a battery that also provides 3.3V, which is why we need the step up booster.

Below is a diagram showing the basic setup of the power circuit:

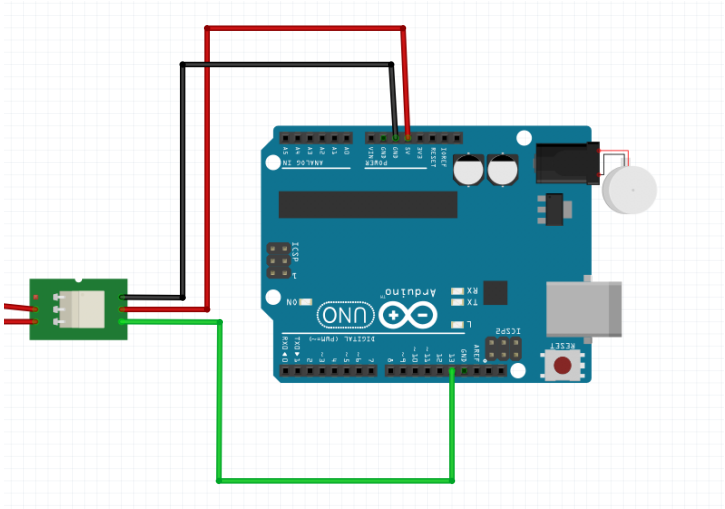


The USB cable will come out of the red component and into whatever component needs power.

2.2 Relay Circuit

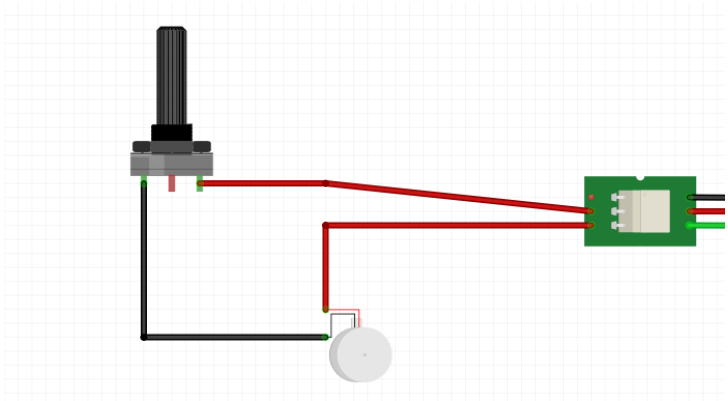
A relay is a component that allows you to stop or start the flow of electricity through the system. You control it by sending a HIGH signal to the SIG pin on the relay. This is easy enough to do on the arduino; on the software side, we will just use a `digitialWrite(outputPin, HIGH)` to switch the relay on. Below is a diagram of the board conencting into the relay[7]:

[6] depending on the plants you're growing
[7] I had to use an UNO here to represent the board, but just imagine any ESP8266 board here. Also, the white disk represents the power source.



On the other side of the relay, we'll need to connect a power source to the NO (normally open) pin, and connect the VCC for the pump to the C (common) pin, and then run the GND of the pump to the GND of the power supply. We want to use the NO pin of the relay because in that configuration, the default state is "off". The third pin on that side of the relay is NC (normally closed), and it would cause the circuit to default to "on".

Below is a diagram of only the relay setup for the pump[8]:



[8]Unfortuately, Fritzing doesn't have a pump object, so I've used a potentiometer object here to represent it. Also, the power supply is represented by the white disk.

References

[1] Igor Fonseca Albuquerque. Solar Charged Battery Powered Arduino Uno. https://www.hackster.io/igorF2/solar-charged-battery-powered-arduino-uno-645d89, 2016.