Lucas Bolanos

5/3/23

# 

# 

# Youtube Link

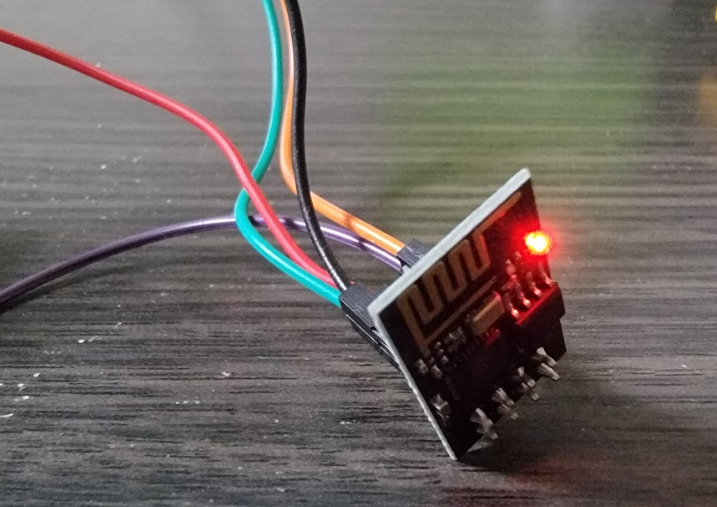
<https://www.youtube.com/watch?v=aUQXYP6OnkM>

# Purpose

Provide a mobile/portable sensor that is connected to a web/smartphone application. Purpose is to provide parents a device that can monitor the temperature and humidity levels in a child’s room in order to monitor their conditions at all times. Especially useful for sick children, say that they have the flu or a cold, and using something like a humidifier. It can be important to keep the room moist, but not too much, and at a certain temperature. The fan automatically turns on to cool the room off if it gets too hot and sends notifications to the user’s phone to let them know the current room's status.

# Functionality

ESP8622



* Self-contained SOC with integrated TCP/IP protocol stack that can give arduino wifi access.

L293D





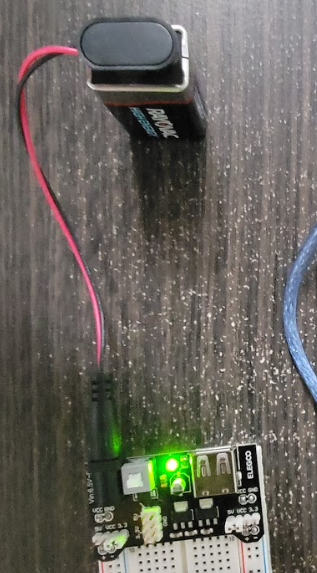
* Pins Used on H-Bridge 1
  + Enable 1,2
  + Input 1
  + Output 1
  + GND
  + Output 2
  + Input 2
  + Vcc 2
* H-Bridge integrated circuit commonly used for controlling speed and direction of DC motors.
* Provides for Arduino a way to drive the motor in either direction and control the speed of the motor.
* Contains 4 high-current half-H drivers and can control two DC motors.
* Handles up to 600mA per channel.
* Arduino sends signals to h-bridge to control direction/speed using pulse width modulation signals.

DHT11



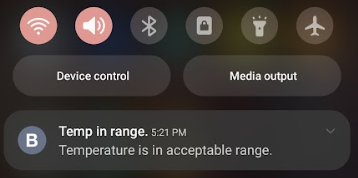
* Measures humidity and temperature.
* Uses a capacitive humidity sensor and a thermistor to measure the surrounding air and outputs a digital signal on the data pin it's connected to.
* Downside/drawback to DHT11 is it may only retrieve new data every 2 seconds.

Breadboard Power Supply Module



* Outputs 3.3v and 5v
* 3.3V for DHT11 and the esp8622
* 5V for the h-bridge and fan.

Notifications



# How tested

* Tested across multiple networks (home networks/mobile hotspots).
* Placed sensor in varying temperature environments to test reading accuracy.
* Placed next to the humidifier and ran for varying lengths to measure levels.
* Tested fan turning on and off.
* Tested packet sending/loss rate through Blynk interface.
* Tested multiple different widgets to monitor levels.
  + Pinging to different virtual pins.
* Tested different setups and wiring configurations for DC motor
  + 1: Using transistors with resistors.
  + 2: Using the H-BRIDGE bus.
* Running ESP8622 module on different current supplies (3.3v v.s. 5v)
* Running DHT11 across different currents.
* Tested power supply module with barrel jack v.s. 9v battery v.s. Usb cable.

# Challenges/Lesson Learned

Learned a ton about setting up wireless/wifi capability between the Arduino and a web/smartphone application. Learned how to generate authentication tokens to allow communication between wifi modules on Arduino and application. Learned about different baud rates and how they interfaced with different modules on the Arduino unit. This led to learning more about virtual pins and how they send/receive information to transmit information to the hardware from the software, and vice versa.

Learned about the power supply module and different configurations it can work under. Learning how to set one rail to 3.3v and the other to 5v was very helpful.

The biggest challenge on this project and also the learning lesson ended up being solving how to make the DC motor/Fan communicate consistently with the Blynk app. I ran into several issues with the DC motor not wanting to operate correctly in conjunction with all the other modules and mission statements of the overall project. For instance, on it’s own I could consistently get the motor spinning and turning off. However; when paired with other modules and even a button widget via the application it ran into many issues.

1. Power draw issue/Fan Communication with App Issue
   1. The motor would turn on, but once on it would draw so much power that the other modules would either begin failing intermittently or completely. This seemed to be the case regardless of the supplied voltage, 3.3V or 5v. The setup used here to power and control the fan was a combination of a transistor paired with a resistor connecting the fan to the digital pins on the board.
   2. The solution ended up being switching to the H-BRIDGE bus and running it completely off the power supply module versus the arduino power directly. This taught me much about the electronic noise a dc motor generates and the dangers it can pose when powered directly by the Arduino board rather than a power supply module. As for the transistor/resistor setup v.s. The H-Bridge, the h-bridge seemed to allow better communication on the virtual pin side of the Blynk application to the digital pin the fan was connected to on the board itself. Through the combination of being able to control the fan direction as well as PWM implementation, this ended up being the best solution for the situation and issue regarding fan control.