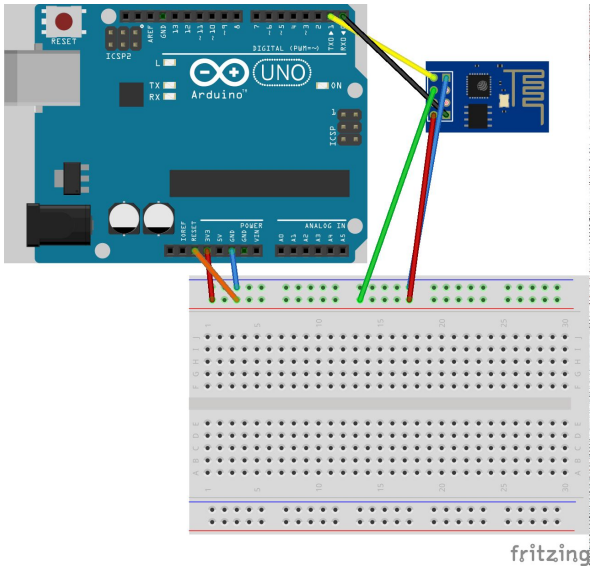
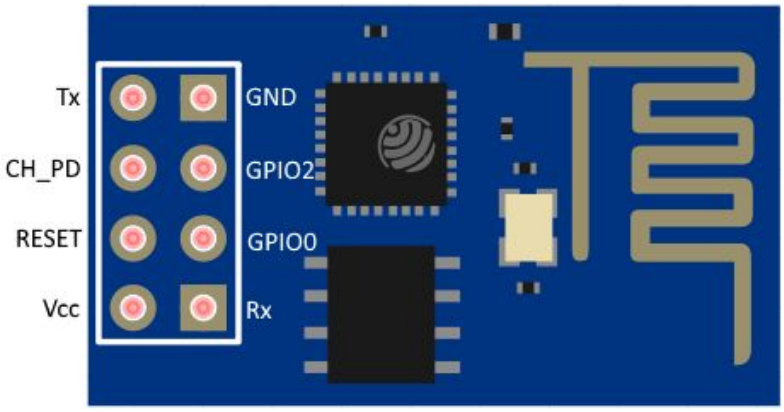
Instructions

ESP Setup

Connect the ESP to the Arduino like so (left are ESP pins, right are Arduino pins):

GND to GND
RX to 0
TX to 1
VCC and CH_PD to 3.3V





Open the Arduino IDE's Serial Monitor. Set line endings to "Both NL and CR". After typing in AT and pressing Enter, an "OK" should pop up - try setting the baud rate to 9600 or 115200 if either doesn't work. If nothing pops up on either, you may need to update the ESP firmware.

Build Circuit

Now that the ESP is ready to work with, the circuit should be completed so that we'll eventually be able to test our code.

First, the connection between the relay and the coffee maker should be figured out. Always make sure that the coffee maker is unplugged while working with its internal circuitry. Open up the coffee maker however you have to so that you can get to the wires that lead to and are from the external plug. Find the wire that supplies voltage to the coffee machine. This will most likely be red and coming directly from the external plug. You'll want to make a cut to that wire between its connection to the external plug and its connection to anything else; the 'anything else' will likely be the switch that allows the coffee-drinker to turn the machine on and off. Now there are two ends of that wire where the cut was made. Take the end that is

connected to the external plug and strip enough of the insulation to get it into the relay. Connect this end to the middle pin. You'll need a screwdriver to secure this wire in place. Assuming that you want the coffee maker to activate when a signal (high voltage) is sent to the relay, connect the other end of the just-cut wire to the NO (Normally Opened) pin. The rest of the circuit should be completed with jumper cables according to the diagram below.

Code

Make sure Blynk works by going to the Blynk website and testing the "Blynk Blink" sketch first over USB, then over Wi-Fi. Both need to work correctly to make sure communication between Arduino, ESP, and Blynk works. You can verify that Wi-Fi is working once the Serial Monitor prints out "Connected to Wi-Fi" and "Ready".

Use the Blynk Blink Wi-Fi sketch as your base. Create variables for your Arduino's latitude and longitude location, and its maximum km range. You will need to figure out a function for determining the km distance between two latitude and longitude positions, in order to calculate the distance between the Arduino and the phone.

The Blynk app on the phone should have a GPS Stream widget - this will contain the phone's location data in latitude and longitude. A BLYNK_WRITE function to the pin of the GPS Stream should calculate the distance using the phone's location data, check whether it is within range, and set the coffee machine's pin to HIGH or LOW.

Testing Tips

Once the code and circuit are ready, the project is ready to test.

If you don't have access to Wi-Fi that allows devices on the network, you could try using a phone hotspot instead. Use phones with good service. Make sure that the GPS stream widget is chosen in the blink app and that the virtual pin that it writes to matches the one that you call from the code in the BLYNK_WRITE method. Connect the Arduino to your computer and review the messages in the serial terminal to help debug. The ESP does not have a connection to the Blynk server until a ready/ping message is written to the terminal.

Introduction

This project is a proximity-based coffee brewer. Coffee is a beverage consumed daily by millions of people, so why not integrate the brewing process seamlessly into your daily routine? This is perfect for anyone that wants coffee to start brewing once they arrive at work or home - by checking the location of the connected Arduino to your phone's location data, it can automatically start brewing when you are nearby and deactivate when you leave.

Conclusion

If time allowed, we would've wanted to create an app independent from Blynk that would be able to set custom brewing times and the location of the Arduino, along with gyroscope support (detect whether phone is in pocket, on table, etc.) We might also want to use a stronger Wi-Fi chip and an external power source for the ESP for more stability.

Our greatest trouble was definitely the ESP - communicating to it and telling it to connect to the Internet gave us many struggles with the various errors it would give us. The moment it connected to Blynk through Wi-Fi was a great relief.

References

Distance in km between latitude and longitude:
https://stackoverflow.com/questions/10198985/calculating-the-distance-between-2-latitudes-and-longitudes-that-are-saved-in-a

Blynk sketches to test code:
https://examples.blynk.cc

Wiring Arduino to ESP:
https://www.esp8266basic.com/flashing-instructions.html