Rachel Tong Apr 17, 2019 @04:25 PM CDT

Arduino Communication to NodeMCU

1. Open the WifiChip_Sample.ino code in Arduino
2. Copy and paste your unique GoogleScriptID into the String GScriptID
3. Upload code to the Arduino **without any connections on RX and TX**
4. Plug the NodeMCU in using a MicroUSB to USB cable. You may use any USB outlet - it does not need to be connected to a computer.
    1. The NodeMCU will have a solid blue LED turn on once it has connected to WiFi. If this LED does not turn on, there is no internet connection available.
5. Making connections on the Arduino and NodeMCU:
    1. Connect the TX pin of the Arduino to the RX pin of the NodeMCU. Leave the TX pin of the NodeMCU unconnected.
    2. Connect the GND from the Arduino to the GND from the NodeMCU
6. The NodeMCU listens to the Serial port and waits for a message
    1. The message is a string with values separated by commas. **The first value is the GoogleScriptID** that you found when setting up Google Sheets.
    2. The Arduino code needs to run **Serial.println**(message). Serial.println is required. Serial.print will not work. **You may only have ONE serial.println statement in your code when you want your WiFi chip to run. Any of your extra debugging statements will need to be commented out.**
7. The NodeMCU does a POST request to Google which adds a new row with the column
    1. Example: the NodeMCU will post a new row of data with 2 columns: 5, 10
    2. If you don't see any outputs on your Google Sheet, try hitting the RST button on the NodeMCU and ensuring the built in LED turns on solid.
    3. The Arduino code should limit the rate at which it transfers data. We recommend sending a datapoint every 10 seconds. Transfers of once every 3 seconds was found to crash either the NodeMCU or the Google Sheet.