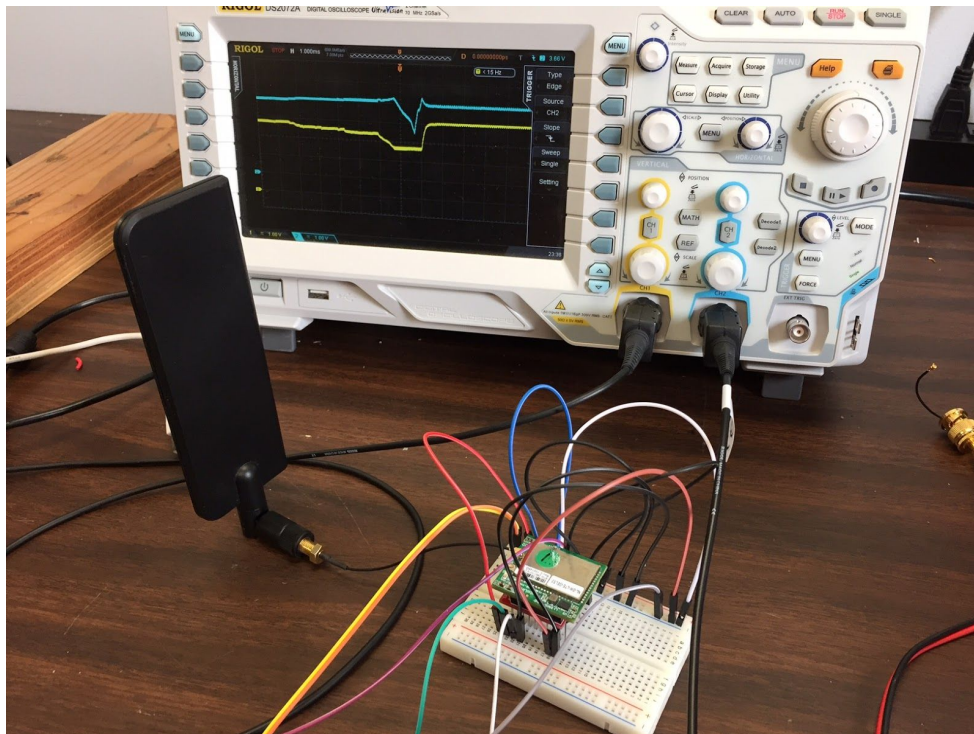


Prototyping with Cellular Modems and Modules

Why Breadboards do not work

NimbeLink Corp

Updated: January 2018



© NimbeLink Corp. 2018. All rights reserved.

NimbeLink Corp. provides this documentation in support of its products for the internal use of its current and prospective customers. The publication of this document does not create any other right or license in any party to use any content contained in or referred to in this document and any modification or redistribution of this document is not permitted.

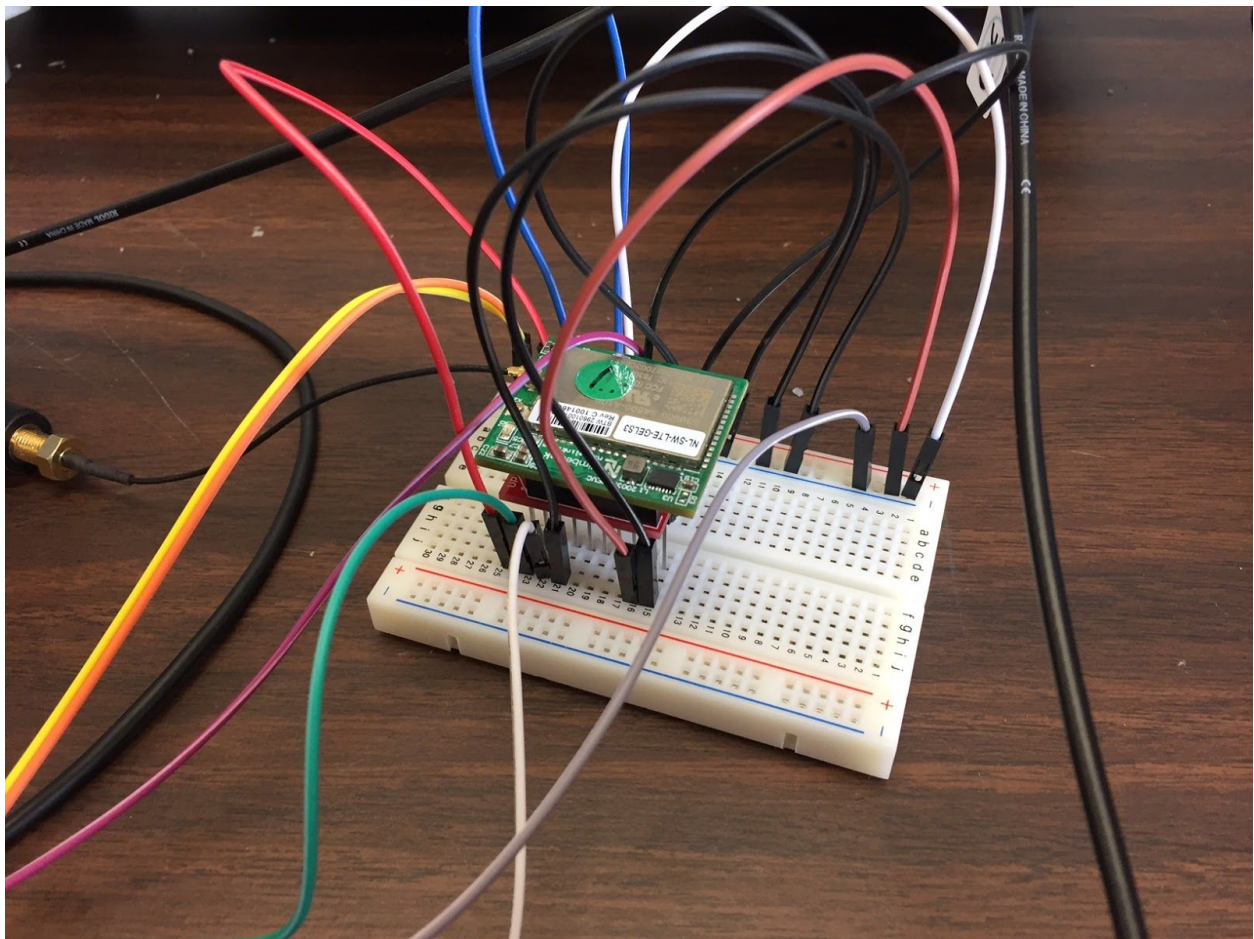
While efforts are made to ensure accuracy, typographical and other errors may exist in this document. NimbeLink reserves the right to modify or discontinue its products and to modify this and any other product documentation at any time.

All NimbeLink products are sold subject to its published Terms and Conditions, subject to any separate terms agreed with its customers. No warranty of any type is extended by publication of this documentation, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose and non-infringement.

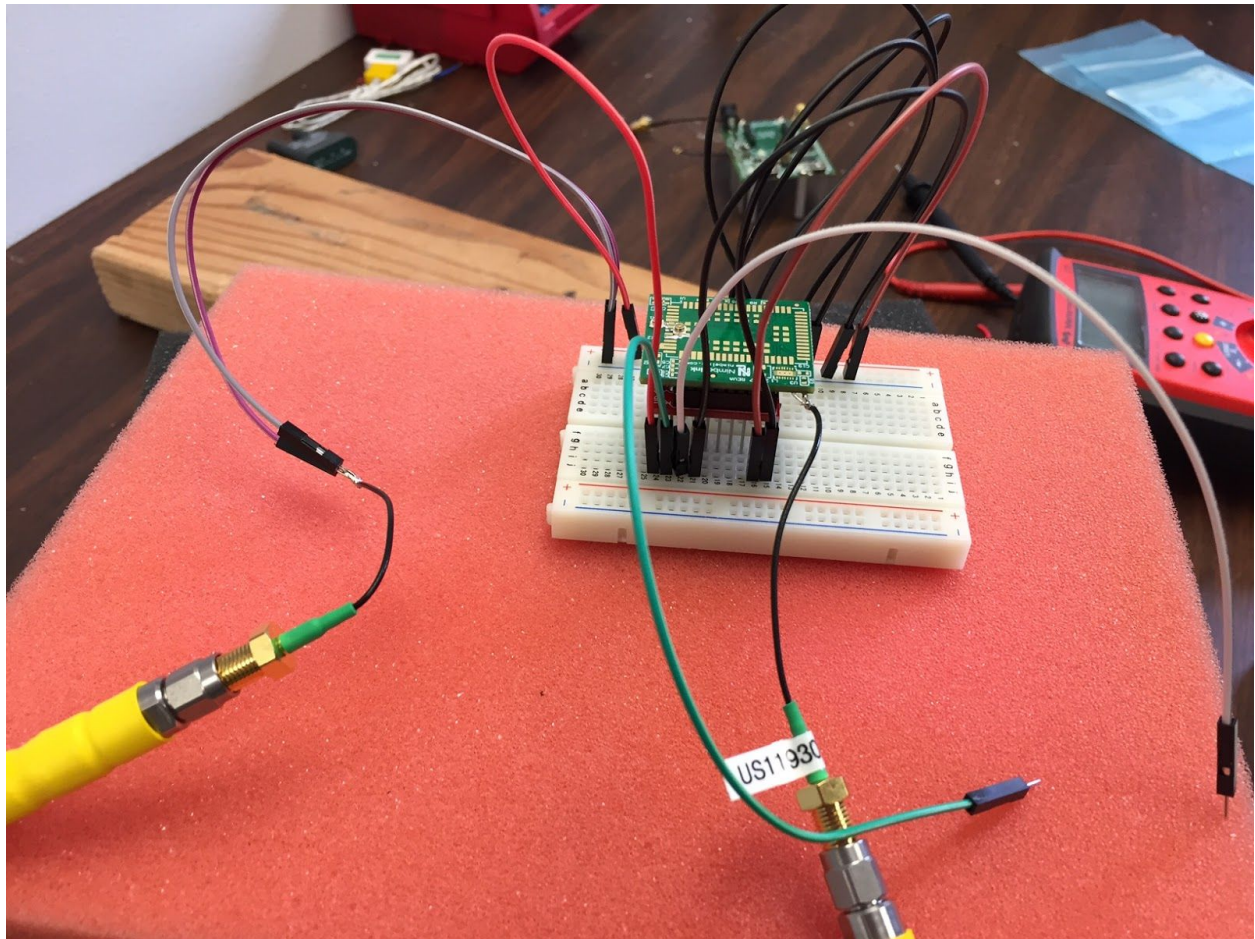
NimbeLink is a registered trademark, and Skywire is a trademark, of NimbeLink Corp. All trademarks, service marks and similar designations referenced in this document are the property of their respective owners.

Breadboards are great for quickly wiring together simple circuits like LEDs, logic gates, and even some low power microcontrollers. While simple to use, the external wires and internal electrical conductors do not work well with higher speed and higher power circuits because they are very high impedance. Instead of acting like low-resistance conductors, they behave like high-resistance conductors at high frequencies. High impedance conductors wreak havoc on high efficiency DC/DC converters and power amplifiers like the ones used in cellular designs. This document explains why this is a problem and give some solutions to those wanting to prototype fast.

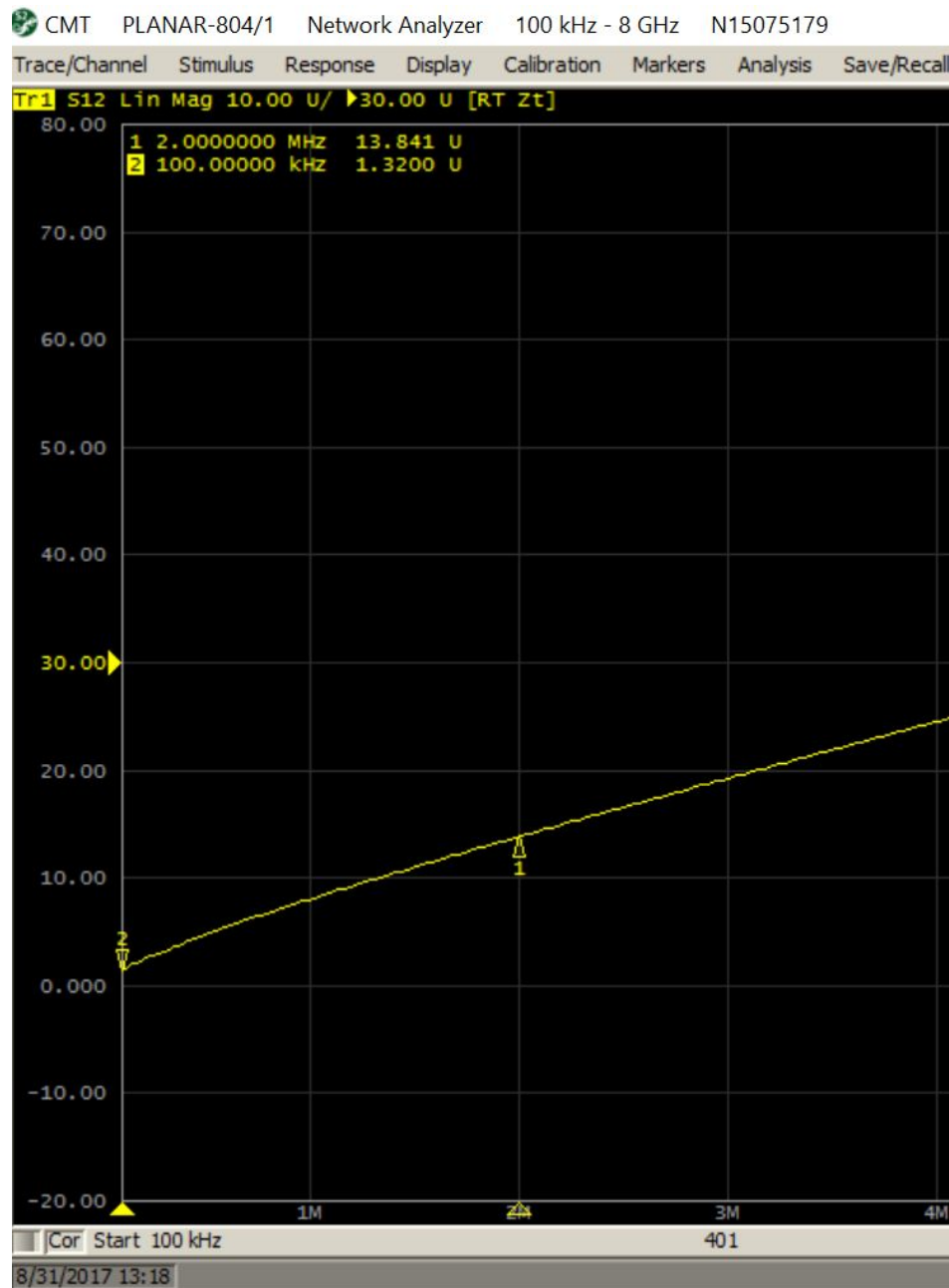
I put together an example of how many customers have tried to use the Skywire cellular modem with a breadboard. First, the Skywire pins do not fit into standard breadboard sockets, so many have purchased an expander adapter to make it fit. This is the red Circuit board underneath the green Skywire in the image below. Next, we see many jumper wires connecting the power and ground pins to the upper rail connections on the breadboard. And lastly we have jumper wires coming out that connect a USB-to-serial converter from the Skywire UART pins to a PC over USB.



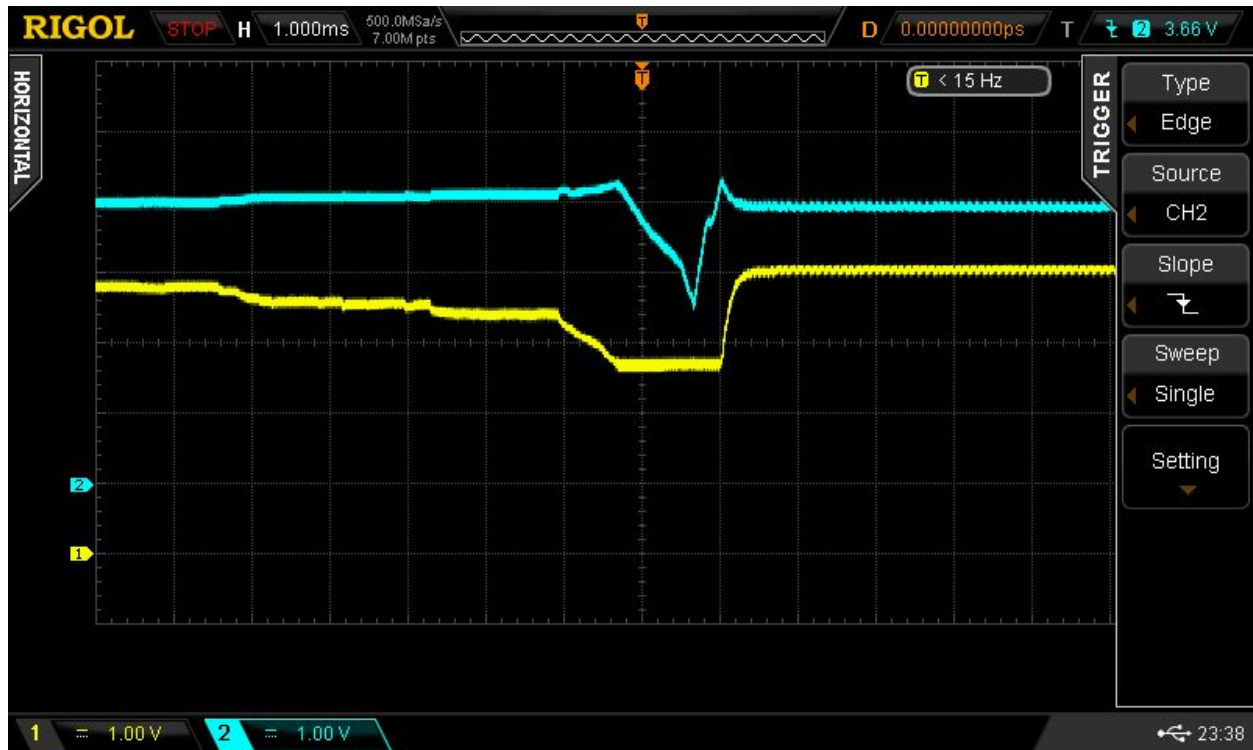
The impedance of the power delivery circuit is important. A low impedance means that the power supply can easily supply power to the circuits that need it. Using a VNA (Vector Network Analyzer) I was able to measure the power delivery impedance from the power supply jumper leads, through the breadboard, expansion board, and up to the modem's DC/DC regulator input. A blank Skywire modem PCB was used to measure just the impedance of the pins and the PCB. The coaxial cable soldered to the PCB and the jumper wires in the setup below shows a convenient way to connect to the VNA.



Below is the output of the VNA. The output has been converted to a real number for ease because the real impedance is a complex number that corresponds to capacitance and inductance. A marker(1) was placed at 2MHz, which is the switching frequency of a lot of small DC/DC converters. The impedance at 2MHz is ~14ohms. (It's important to note that the VNA is measuring impedance across a range of frequencies, a resistance measurement with a multimeter is a measurement at 0Hz or DC, and would have a value of ~0 ohms)



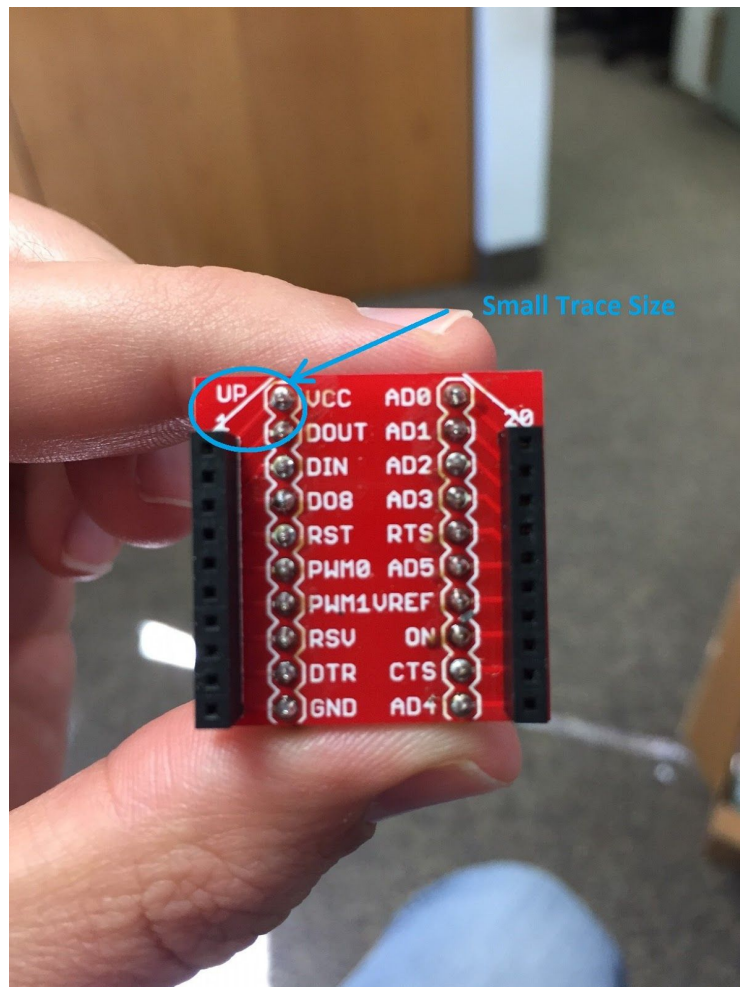
While 14 ohms doesn't seem like a lot, cellular modems are capable of several amp bursts when communicating with the cellular network. A 2Amp burst at 14 ohm impedance would result in a 28V drop across the power input, browning out the modem. And this is in fact what happens when I try to use the breadboard setup. I probed the input (yellow) and the output (blue) of the DC/DC converter on the modem. As the current demand increases the input voltage drops across the large impedance until DC/DC stops regulating. The output voltage sharply drops off causing the modem to brownout around 2.5V and lockup. The modem is now unresponsive and needs a system reset to regain communications.



So what causes the high impedance that results in a voltage drop?

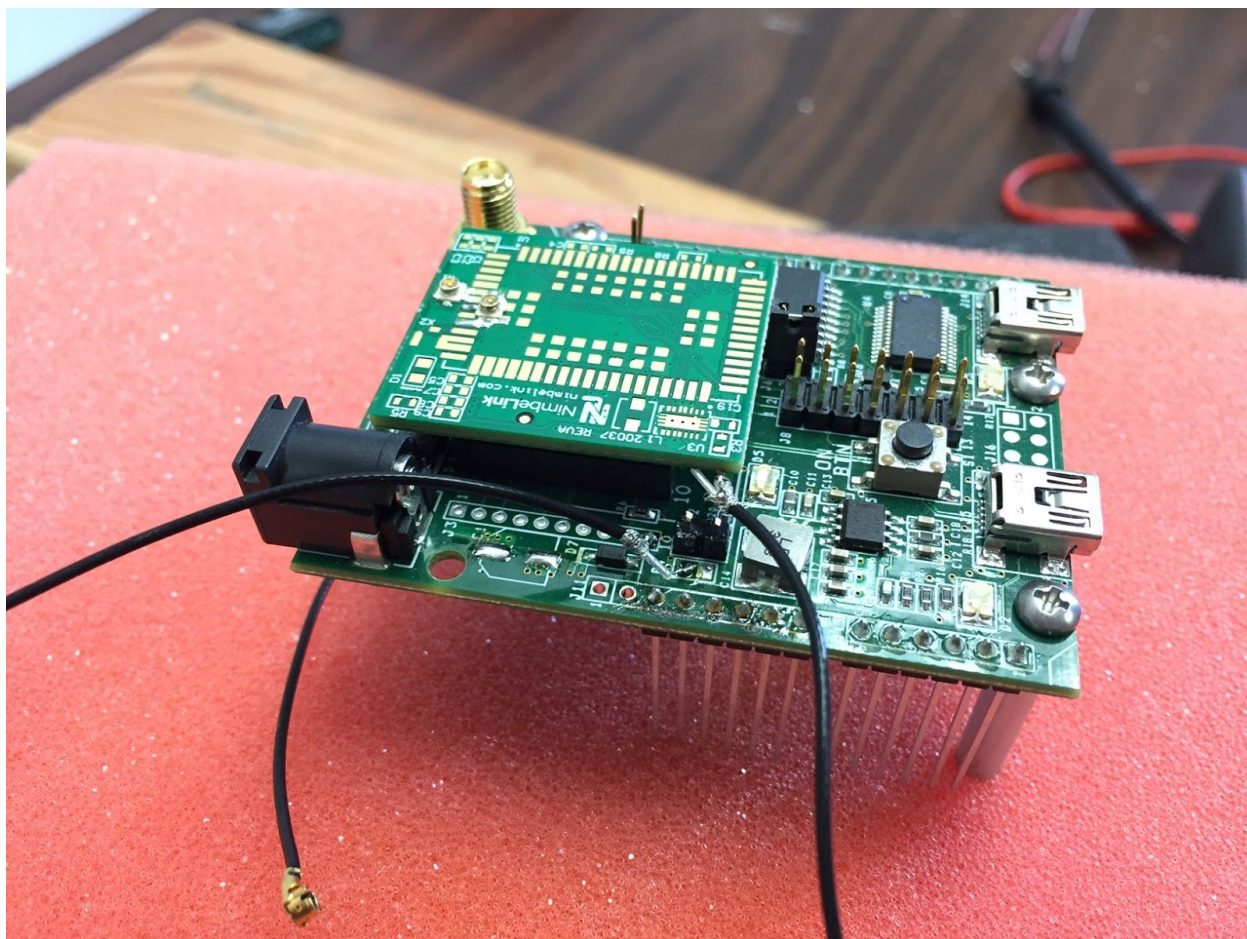
The answer is inductance caused by the long jumper wires and the contact to contact capacitance in the breadboard itself. Recalling circuit theory, voltage and current cannot change instantaneously in the presence of capacitance and inductance respectively. The higher the inductance and/or capacitance, the slower this rate of change. When a cellular modem demands a fast burst of power, the charging and discharging action of the capacitance and inductance will not allow power to be delivered to the modem before it has browned out.

I've already mentioned that the long leads and the breadboard itself contribute to the high impedance, but another contributor in this case is the red adapter PCB. Below is a picture of the adapter PCB's narrow power trace that is not wide enough to provide a low impedance power path.

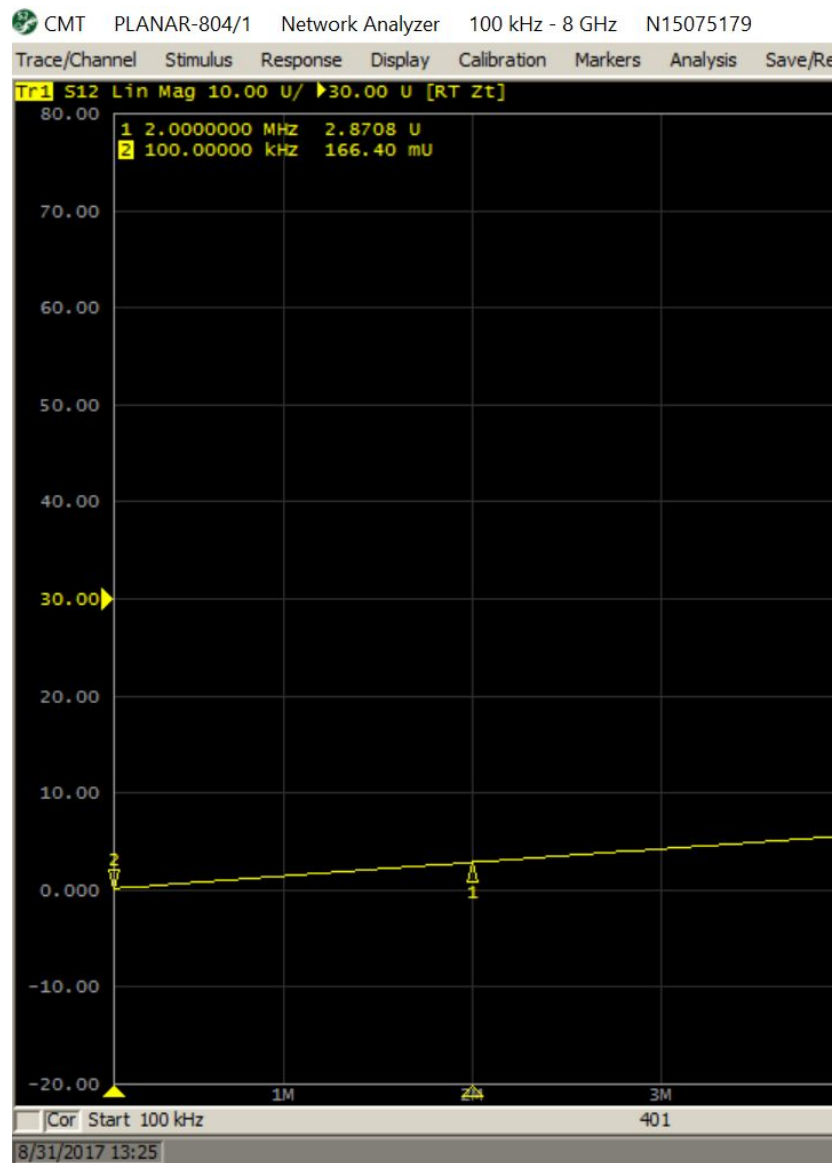


So what can we do to fix it?

Eliminate as many sources of impedance as possible. This can be achieved by creating a shorter, and wider power path between the power source and the modem. We can demonstrate this by using a devkit that has a power regulator closer to the modem compared to the bench supply through jumper wires and breadboard farther away from the modem. Below is the NimbeLink SWDK (Skywire Development Kit) modified so it can be measured with a VNA.



Two coax connections to the VNA are soldered to the output of the regulator on the development kit, and the input to the DC/DC on the Skywire modem. Once again, I am using a blank PCB for the Skywire modem to measure just the impedance of the traces and the pins.



The new impedance at 2MHz is 2.87 ohm, a dramatic improvement compared to the 14 ohms of the breadboard. An observant reader might say that at 2A with 2.87 ohms still creates a 5.74V drop and should cause the modem to brownout. The reader would be correct if not for input bulk capacitance placed close to the input of the DC/DC converter on the Skywire modem. This bulk capacitance acts as a reservoir to supply power for short periods of time. The reason bulk capacitance doesn't help out the breadboard setup is that the time between bursts of power is faster than it can recharge with the high impedance input. With the low impedance input on the devkit, the bulk capacitance can recharge fully in time for the next power burst. This is also why adding a larger bulk capacitor doesn't always work. A larger capacitor takes longer to recharge and the high impedance slows the rate of charge.

What's the best way to prototype with the Skywire modem?

It might sound like a sales pitch, but the answer is any of the NimbeLink development kits. We have many different kits that interface to major microcontroller/processor development kits like TI, ST, Freescale/NXP, Raspberry Pi, and others. These boards have integrated power supplies, and all the connections needed to interface between a microcontroller devkit and the Skywire modem. We provide design files for our development kits free of charge so customers can copy them into their custom designs. NimbeLink's goal is to accelerate customer Skywire designs to market, not profiting from devkit sales. We offer them at competitive prices at Digikey, Symmetry, Arrow Electronics and others. Here is the link to our development kits:

<http://nimbelink.com/skywire-development-tools/>

**Note the Arduino Cellular Shield is not recommended for new designs, as it cannot handle the power requirements needed for LTE. Please use the SWDK instead.*

What if I don't want to buy a NimbeLink Devkit?

If you do not want to purchase a devkit, then you must rigorously test your adapter solution to ensure that the power delivery is adequate for the modem you want to use under all conditions. All modems have different power demand characteristics, so just because a prototyping solution works for one modem type does not mean it will for all. Another point of error is the power solution might be adequate for issuing AT commands to the modem, but as soon as the modem tries to connect to the network, it will brown out.

Trying to debug prototypes with power related uncertainty is frustrating and can lead to delays. Engineers will be spending precious time validating a prototype setup instead of working on the final custom product. When customers reach out for support, one of the first questions we ask is "Can we have a picture of your setup?" and "Have you tried your tests with a NimbeLink devkit?" This helps eliminate unnecessary troubleshooting due to how the prototype was wired. Purchasing a NimbeLink devkit helps eliminate early problems so customers can focus on validating the technology, completing their design, and software verification.