

A LONG-RANGE SOLUTION FOR TRIGGERING ANALYZERS WITH ARDUINO MICROCONTROLLERS

07/14/2022

Introduction

Remote measurements using VNAs are becoming an increasingly popular method due to the increased portability of USB vector network analyzers. For applications such as far-field antenna measurement setup and using a VNA as an insertable remote sensor, the ability to remotely control the VNA becomes imperative. This remote connection is usually accomplished from PC-PC using Component Object Model/Distributed Component Object Model (COM/DCOM) interfacing or Transmission Control Protocol (TCP) interfacing. WIFI is typically used to connect these two separate PCs, but what options exist when WIFI is not available? Ethernet can certainly be used to bridge these gaps within a specific range, of course. Fiber optics can also be used for a much greater distance, but then cost becomes a limiting factor.

An affordable and effective option is a simple Arduino & transceiver setup. The cost is modest, while the range is dramatically increased.

VNA Automation Basics

Copper Mountain Technologies (CMT) metrology-grade VNAs are USB-based. USB-driven instrumentation has no built-in computer and separates the measurement module from the processing module. Remote processing allows the VNA to be very portable and lightweight since a PC handles all the processing. Additionally, the CMT VNA software programs will work with any x86 processor, meaning most desktops, laptops, and even some tablets.

When measurements are automated with CMT VNAs, the programming language communicates directly with the software, which then communicates through the USB connection to the VNA. The communication link is important to be aware of since the software must run in the background to make and maintain a connection. If the VNA software is closed during automation, the connection will be lost, and automation will fail.

There are two automation methods for CMT VNAs. One option uses SCPI commands and the other uses COM (Component Object Model) functions. These methods send commands to the VNA software, which triggers the software to perform a function. Method selection depends on user preference. However, before sending any commands, a connection between the VNA software and the automating PC must be made.

Connecting the VNA to the same PC executing the automation programming simplifies the VNA and PC connection. A TCP/IP socket or TCP/IP HiSLIP connection can bridge the two. Either of these connections can be made in one of two ways:

- 1) Standard TCP client library offered by each programming language.

2) VISA.

These connection methods will not be covered in this application note but are clearly explained in this [Automation Guide for CMT VNAs application note](#).

The key takeaway is that a connection must be established with the VNA software to automate. No matter the method, automation can begin after establishing a connection. A user can make this connection across devices utilizing some of the methods mentioned earlier in the introduction. Microcontrollers themselves cannot connect to the VNA software since they are not x86-based. However, they will be used alongside the automation and will be connected to the automation script itself. Utilizing microcontrollers will be covered later in this application note.

Establishing a Connection Between Two Arduino Microcontrollers

A microcontroller is a microcomputer designed to control embedded systems. Components of a microcontroller are typically memory, peripherals, and a processor. Arduino is a company that produces microcontrollers that pair with their own IDE (Integrated Development Environment), which runs on a computer. The IDE is used to write and upload computer code to the physical board. While Arduino is not the only company to make microcontrollers, their products will be utilized for this application note. Figure 1 is the image of the Arduino Micro utilized. Other manufacturers' products can also achieve the result of this application note.



Figure 1: Arduino Micro

Peripherals for the microcontrollers can be attached via the header pins available on the boards. The software is entirely open-source. Premade libraries are available to simplify the coding of embedded software. Altogether this makes the Arduino products very user-friendly and easy to learn.

Communication between two Arduino Micros can be done in a multitude of ways. It can be as simple as connecting a couple of jumper wires from board-to-board, or as complex as utilizing radios. Transceivers are very popular peripherals utilized for point-to-point data transmission in these systems. These small radios come in a plethora of frequencies and form factors, making them great for any project.

The 433 MHz RFM69HCW transceiver module from Adafruit was used for this test, as shown in Figure 2.

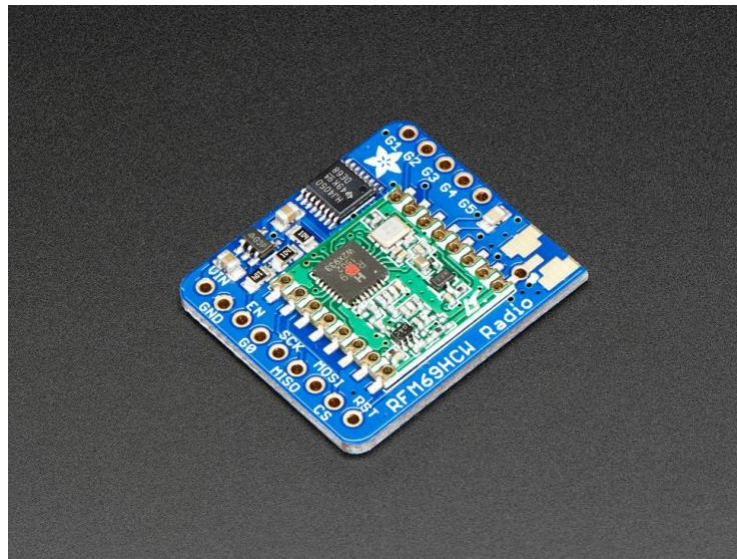


Figure 2: Adafruit RFM69HCW

The RFM69 connects to the Arduino Micro through the SPI and operates in the lower ISM band at 433 MHz. This frequency provides the signals with a solid range and good wall penetration. It operates off a max 150mA current draw for transmission and roughly 30mA while receiving. The data sheet lists a range of 500 meters line of sight with the proper antennas. Of course, these specifications are indicated with the power output set to max, which is +13 to +20 dBm or up to 200 mW.

The radios utilize AirSpayce's Radiohead library, with many examples that can be uploaded and tested immediately. These examples range from a 2-node network to a more complex multi-node network. These are a great way to test radios and wiring. [Visit this link](#) for more information on connecting to the transceivers.

Connecting the Arduino Micro with the VNA Software

As previously mentioned, the Arduino microcontrollers can't directly interact with VNA software. The workaround is using Python or any language that can establish a serial connection with Arduino Micro. The serial library with Python was used, as it is open-source and easy to understand.

Once the serial connection is made between the Arduino Micro and Python, communication can occur through the serial port. The serial connection allows Arduino to tell Python what to do, and then Python will send commands to the software. For instance, if the Python script receives a *1* from the Arduino Micro, it will collect data from the VNA software from the current trace, and then send a response to the Arduino. This *1* can be sent from a button press, a sensor reaching a specific value, or many other triggers. The possibilities become endless when considering the opportunities with an Arduino.

Not only can the Arduino talk to Python, but it can also listen to the Python script. For instance, the results are communicated to the Python script if a limit test is performed in the software. The information about whether it is a pass or fail can be sent through the serial port. When the Arduino receives the reply, it can perform a designated action. The designated action could be blinking an LED, playing a noise through a speaker, or outputting a PWM signal to something. Also, there are many possibilities to tie in other testing equipment.

Connecting the Arduino Micro with the VNA Hardware

While the Arduino can't directly interface with the software, it can interface with the VNA hardware. On the CMT VNAs, there are BNC ports for an input and output trigger. Utilizing the Arduino Micro IO pins, you can directly connect to these. The IO pins can generate 3.3V or 5V square wave pulses; the voltage depends on the model of Arduino. Using this signal, you can trigger the VNA.

For triggering, this makes the scripting connection unnecessary. This is strictly a hardware-to-hardware connection; thus, it is straightforward yet robust. An Arduino to software connection is only needed when you plan to add more functionality to the setup.

Use Cases Once Connected

When you combine the previous sections, you can create some very useful setups. To display a real-world use case, this section will describe and explain the example setup. Figure 3 is a simple block diagram of the setup. Note that the Arduino is connected to the PC at the stationary node for this example. This is not necessary if only triggering the VNA.

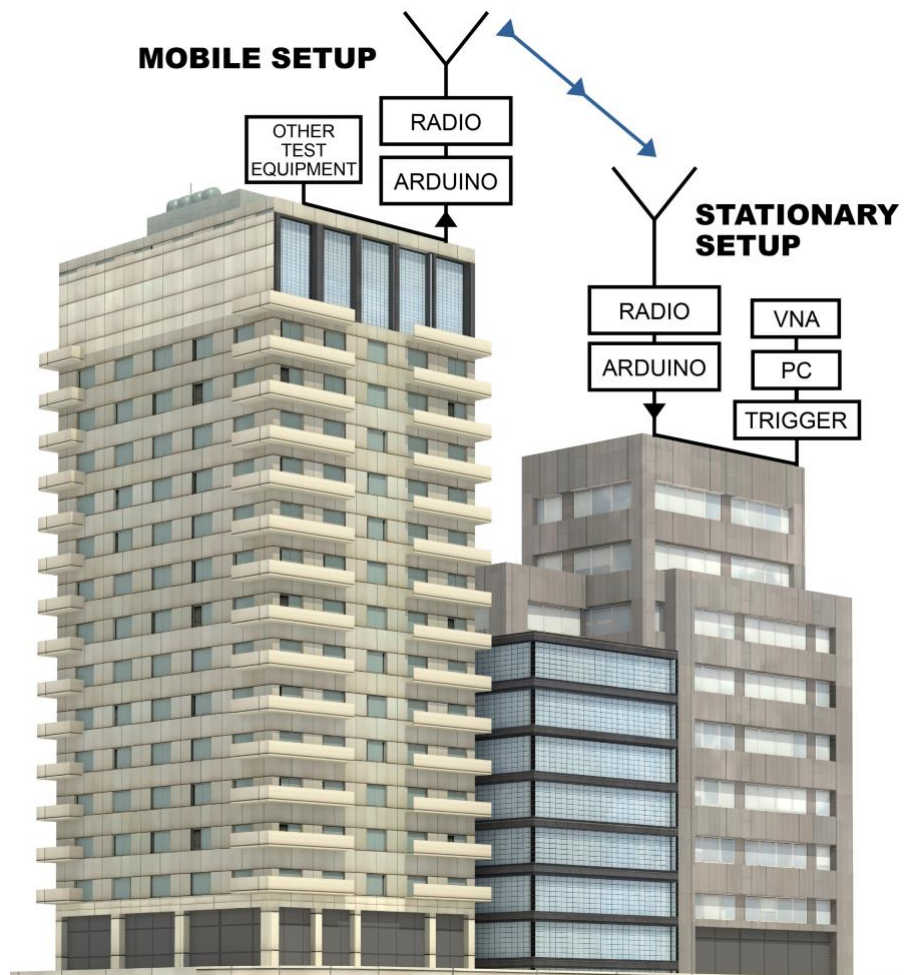


Figure 3: Dual Node Block Diagram

The scenario involves remotely triggering a VNA from some distance away. Once the sweep finishes, a limit test is run on the trace, and the results are then sent back to the initial Arduino Micro. Only one Arduino Micro is interfacing with the VNA and software, while the other is used only as a mobile transceiver.

Both the stationary and mobile sides of the setups are identical. The only difference is that the stationary node will be connected to a PC, but the mobile node will operate independently. As mentioned in the previous section, technically, you can utilize the stationary node without connecting to any form of PC. The VNA can be triggered using the external trigger port on the rear panel. With no PC, it will limit the ability to communicate with scripts to perform more complicated tasks.

Figures 4 and 5 show the breadboarded and protoboard versions of one of the complete modules. These modules are wired the same as the Adafruit link mentioned above. 3 LEDs (green, red, and blue) were added on open digital pins for signaling things, like a pass or fail of the limit test. A capacitor was added in

parallel with the transceiver's Vin and GND to make sure the Arduino could drive the transceiver at the maximum output power. On the trigger sending node, a simple push button to manually send a trigger signal was included. The node attached to the VNA utilized a BNC to alligator clip. The clip was attached to an open digital pin and ground, and then the BNC to the trigger input on the back of the VNA.

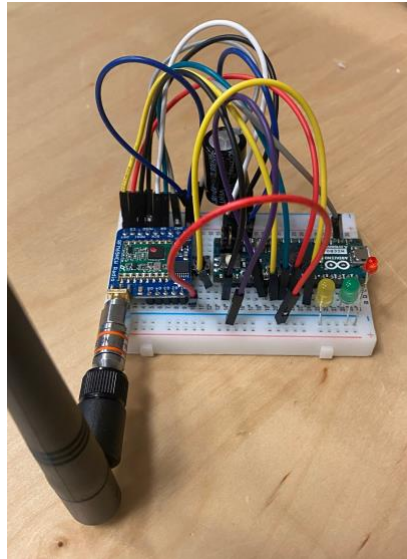


Figure 4: Breadboarded Module

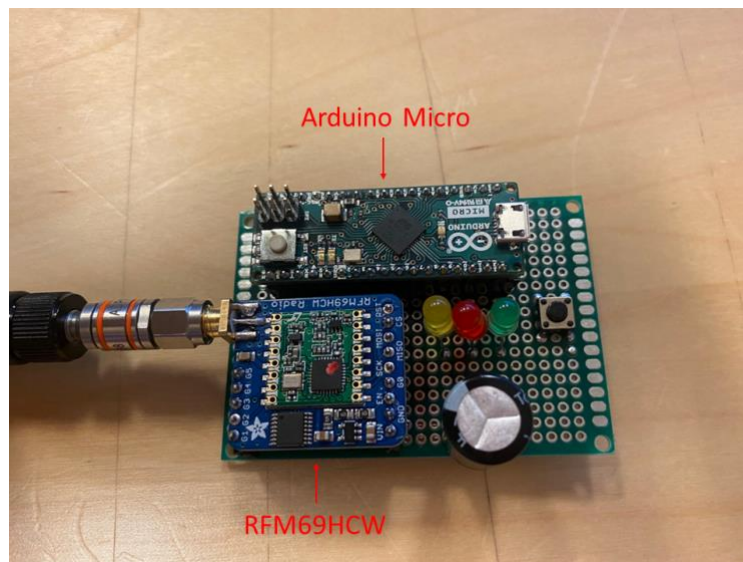


Figure 5: Protoboard Module

After both nodes had been put onto the protoboard, two SMA end-launch adapters were soldered onto the transceivers. To this, a male-to-female adapter attached two quarter wavelength 433 MHz whip-tilt antennas. These 50 Ohm antennas provided a VSWR under 2.0 while being very affordable.

One node was attached to a PC and VNA inside CMT headquarters. The mobile node was powered by a portable power bank and taken outside and down the street. When the button on the mobile node was pressed, it sent a trigger to the stationary node inside CMT headquarters. The yellow LED blinked on both nodes. The stationary node then triggered the VNA and sent the command in the Python script to wait until the VNA finished sweeping. The script then performed a limit test on the current trace. If the trace passed the test, a 1 was sent from the script to the stationary station. If the trace failed, a 0 would be sent instead. After receiving the results, the stationary node indicated the results to the mobile node. Once received, the mobile node flashed the corresponding LED: green for a 1 and red for a 0. The results of the trigger were achieved with only a button press.

This setup was tested in Indianapolis, a large city where the CMT headquarters is located, so testing was limited. However, the configuration achieves a guaranteed 75 meters of range, including the penetration of surrounding buildings' walls. This range will be far more significant in the intended use case of point-to-point with fewer obstacles in between.

This test operated with a simple Pass/Fail indication, but it would be a simple matter to send actual measurement data over the radio link with speeds commensurate with the link data rate.

For more information on scripting, [visit this link for details on Triggering VNAs with Arduino Microcontrollers](#). In addition, the exact code utilized for this project is available, including the Python file and the two Arduino sketches. For any questions or comments about this portion of the project, [please reach out to our expert engineers](#) who are glad to support you.

Conclusion

Interacting and interfacing with test equipment from remote locations is becoming an increasingly necessary topic these days. Engineers can take this concept even further with the benefits of the lightweight USB VNAs from CMT. The portable, compact VNAs allow measurements to be taken directly in the field and on-site. This information can be collected in person or relayed to the software the device runs on.

Integrating Arduinos or any microcontrollers allows for very complex situations to be turned into simple projects. The example shown in this application note is just the tip of the iceberg. The engineers at CMT hope to support engineers around the world to continue to push the boundaries of what is possible in the test and measurement world.

Copper Mountain Technologies aims to innovate and bring new solutions to VNA users every day. If having any trouble using a CMT VNA or to learn more about this project, in particular, [contact the CMT support team](#) for assistance.