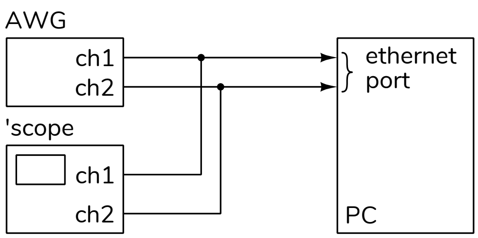
10 Mb/s Ethernet Transmitter

Introduction

In this lab you will use a hex editor to create a file containing an Ethernet frame. Then you will use a supplied script to convert this file to a 10 Mb/s Ethernet waveform in a format suitable for the AWG. You will load this file into the AWG which will generate an Ethernet waveform that will be received by the Ethernet interface on the lab PC.

You will use a 'scope to measure the interface voltages and signal timing and the Wireshark protocol analyzer to display the frame contents.



Configure the AWG and Verify Waveform

Create a folder and download the `lab9.zip` file from the course web site, and extract the files to this folder. Put `test1.raf` and `test2.raf` on your flash drive and plug the flash drive into the AWG.

The configuration of the AWG is somewhat complicated by the need to use both channels to obtain a differential output. The AWG also has be set up in burst mode so that one waveform (frame) is transmitted every 16 ms. This is the time interval between the NLP ("normal link") pulses that are included in the waveform created by the software.

To configure the AWG start by using the menu item:

- [Utility] / [Set to Default] / [OK] to reset most of the settings.

Then configure the AWG to output the generated waveforms on Channels 1 and 2 with a 16ms gap between waveforms:

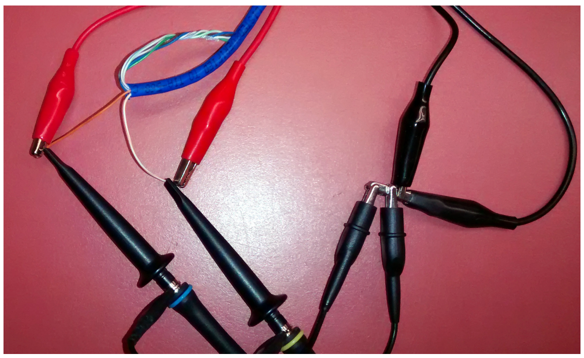
- press [Arb]
- press [Arb Mode] until Freq is selected (you may need to press [▽] to reach this menu item)
- set Freq/Period to Period (you may need to press [△] to reach this menu item)
- set the Period to the waveform period printed by the software (68.266 μs for the test waveforms)
- press [Ampl] until it is in HiLevel/LoLevel mode
- set HiLevel to 1 V
- set LoLevel to 0 V
- press [Select Wform] to enter the waveform selection menu
- select [Stored Wforms]
- select [File Type] [Arb File]
- press [Browser] until Dir is selected
- select the D: drive (USB)
- press [Browser] until File is selected
- select `test1.raf` and press [Read]
- press [Burst]
- verify that the mode is NCycle with 1 waveform cycle per burst and internal triggering
- set the burst Period to 16 ms
- Press [CH1/CH2] to switch to configuring Channel 2. Repeat the above steps for Channel 2 but use the file `test2.raf` instead.

Now align the waveform phases and enable both outputs:

- press [Arb] [Align Phase] to ensure the start of both waveforms are time-aligned
- press [Output1] and [Output2] to turn on both outputs

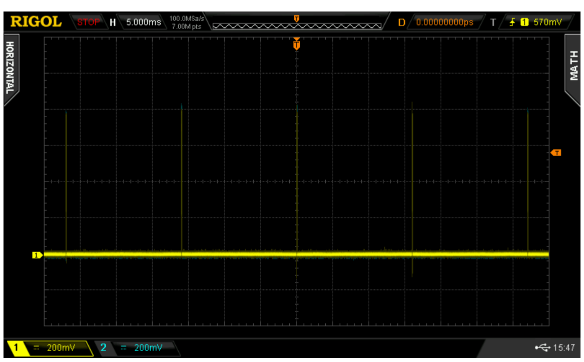
Note: the AWG may disable burst mode and alignment between channels when the waveform changes. You'll need to measure the waveforms on the two channels as described below to check the waveform periods, amplitudes, polarities and alignment between channels.

Use two BNC-to-alligator clip cables to connect the AWG channel 1 and 2 outputs to the probes for the two 'scope channels. Connect all of the ground leads together. Do not hook up the Ethernet cable yet (although it's shown in the photo below). Do not connect either waveform generator output to ground!



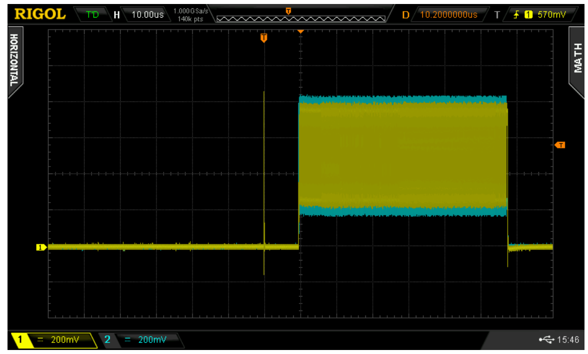
For a stable display of the waveform you will either need to capture and hold one waveform (simplest) or use the holdoff trigger feature[1] to ensure the 'scope triggers only once per frame.

If you superimpose the traces from the two channel you should see a 10BaseT Ethernet frame being transmitted every 16ms:



and each frame should be preceded by a normal link pulse:

[1]On the 'scope's Trigger menu select Channel 1 source, rising edge, select Settings / Holdoff and set the holdoff to a value longer than the waveform duration (e.g. 80 microseconds) but shorter than the time interval between frames (about 16 ms). Set the trigger level to about 500 mV.



If the two channels are not aligned you will see two sets of pulses every 16 ms.

Enable the Math channel and display the differential voltage (A-B with A as channel 1 and B as channel 2) as well as the two AWG channels superimposed. The initial portion of the frame should be something like:

Connect to Ethernet Interface

A Cat 5 UTP cable should be connected to the lower RJ-45 connector on the back of the PC and brought out to exposed wires. If necessary, ask the instructor for help in locating the right port and cable.

Strip enough insulation from the ends of the orange/white and orange wires so you can connect them to both the 'scope probes and AWG outputs (see photo above):

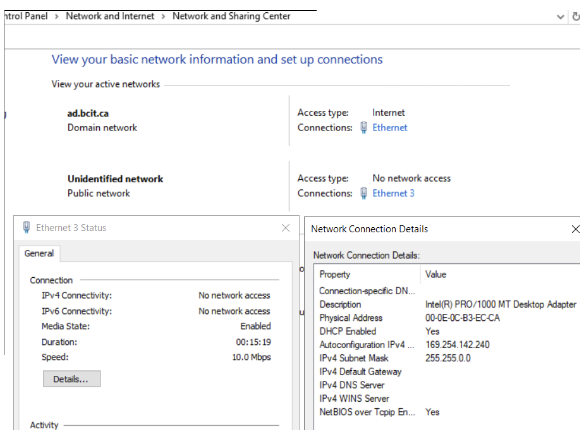
- the white/orange wire is Ethernet signal **RX+** and should be connected to AWG channel 1 and 'scope channel 1.
- the orange wire is Ethernet signal **RX-** and should be connected to AWG channel 2 and 'scope channel 2.

Each AWG channel has a fixed 50 ohm output impedance which matches the approximately 100 ohm impedance of the cable and Ethernet port. Note that the AWG output voltage drops due to the load presented by the Ethernet port.

Also verify that the signal is not distorted. If the end of the cable is not connected, the unterminated cable will reflect the signal back with a delay of $\approx$ 10 ns per meter of cable and result in ISI that will be visible on the 'scope.

View Packet Contents Using Wireshark

From the Windows Start menu search for "View Network Status". Find the network interface with the network interface card (NIC) that is not connected[2]:

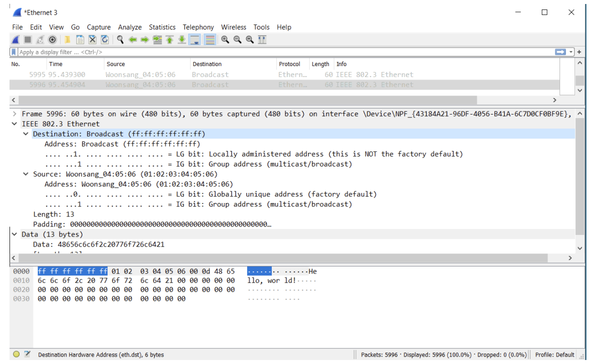


Run the Wireshark protocol analyzer, and double-click on the network interface found above to start capturing frames.

If the Type/Length field is less than 0x600, Wireshark will treat the packet as IEEE 802.3. The rarely-used 802.3 format uses the Type/Length field as a length field and includes a 4-byte LLC header following the type length field. Disable decoding of the 802.2 LLC protocol under the menu item Analyzer / Enabled Protocols... and de-selecting the LLC checkbox.

You should now see packets being displayed that say "Hello world" in the data portion as shown in the screen capture below.

[2] It will probably say "No network access" as shown above.



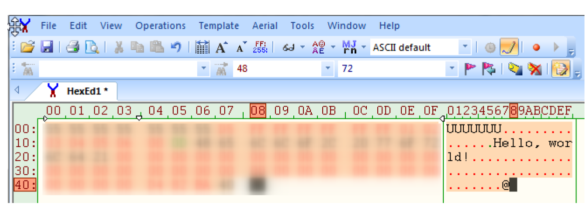
Create an Ethernet Frame

Now that you have verified that everything is configured and connected properly, you need to create and transmit your own Ethernet frame to the PC.

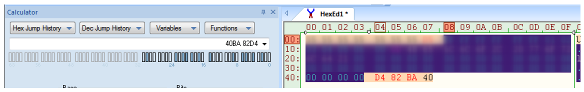
Run HexEdit (available as "HexEdit 4" on AppsAnywhere) and use the menu item File⇒New to create a new file. Select a file size of 0 bytes. Right-click in the data area and select "Allow Changes". Use the Insert key to switch between inserting and overwriting existing values. Use the Tab key to switch between entering values as hex or ASCII characters.

Review the lecture notes describing the Ethernet frame contents.

Add a preamble and SFD. The destination address should be the broadcast address (six bytes of `0xff`). The source address should be the last six digits of your student ID (e.g. if you ID was A00123456 you would use a source address of `01:02:03:04:05:06`). Set the length/type field to a two-byte big-endian number whose value is the number of bytes in the packet between the preamble and FCS. This value will prevent decoding of the payload because it will not correspond to a known Ethertype (e.g. IP). The data field should contain your name padded with zeros to reach the minimum frame length.



To compute the FCS, select the bytes *following the preamble* and compute the 32-bit CRC using the menu item Operations⇒Checksum⇒CRC 32. Add this 4-byte value to the end of the packet. Note that the CRC value computed by HexEdit is a 32-bit value that must be transmitted in order from LS to MS bit (you will have to reverse the byte order; the program that generates the bits will reverse the bit order).



Use File⇒Save to save the data to a file called **frame.bin**. Make sure the length of the file in bytes is the minimum Ethernet frame length[3] plus the preamble length (including the SFD).

Convert Frame Data to Waveform

The **etherlab.py** script reads a file named **frame.bin** and encodes the bytes as a Manchester-encoded waveform in lsb-first order. Therefore, if the bytes in your file represent a complete and correct Ethernet frame (including preamble and CRC) the output waveforms will represent a valid 10BASE-T frame.

The program reads the **frame.bin** file and generates two files, **channel1.raf** and **channel2.raf** corresponding to the files that should be output on channels 1 and 2 of the AWG respectively.

Make sure Python is installed (e.g. launch it from AppsAnywhere), open a Powershell window in the directory containing your **frame.bin** file and the **etherlab.py** file, and enter the command **py etherlab.py**[4]. The program should create the two **channel*n*.raf** files. Copy these to your flash drive so you can load them into the AWG.

The program will also print the duration of the waveform. Record this; you need the waveform duration when configuring the AWG.

Check your Frame

Load the **channel1.raf** and **channel2.raf** files into the AWG as described earlier and view the decoded frames using Wireshark.

Take a screen capture (or two) showing the contents of the 802.3 header and data sections in both hex and ASCII, similar to that shown above, for your report.

[3]Unless your name is very long, it should fit into the minimum-length Ethernet frame.

[4]Shift-right-click on the File Explorer Window and select "Open Powershell Here".

Troubleshooting

You can view statistics for adapter *name* using the command:

get-netadapterstatistics *name* **|format-list ***

where *name* is the name of the network adapter connected to the AWG as found earlier.

If the "ReceivedPacketErrors" counter is increasing, then there is likely an error in your Ethernet frame. For example, an incorrect CRC ("FCS").

Report

Submit a report including the following:

- the usual identification information
- a screen capture of the HexEdit display clearly showing your packet's contents in hex and ASCII
- a 'scope screen capture showing the single-ended and differential voltages at the start of the packet showing the start of the preamble
- screen captures from Wireshark showing the contents of the received Ethernet header and data fields, showing both the decoded and raw (hex/ASCII) sections.
- Answers to the following questions:

  (1) What is the minimum length of an 802.3 frame, not including the preamble, header, and FCS?

  (2) What is the frequency of the preamble waveform[5]? Why?

  (3) Assuming the minimum frame size, how many bytes will be transmitted per frame, including preamble and FCS?

  (4) What part(s) of the frame you created do(es) not appear in the Wireshark display?

[5]Read the square wave's period from the 'scope screen capture and compute the frequency. The answer may not be what you expect!

Appendix - RJ-45 Pinouts

The diagram below, from the IEEE 802.3 standard, shows the pin assignments for 10BASE-T interfaces using RJ-45 jacks:

Contact	MDI signal
1	TD+
2	TD–
3	RD+
4	Not used by 10BASE-T
5	Not used by 10BASE-T
6	RD–
7	Not used by 10BASE-T
8	Not used by 10BASE-T

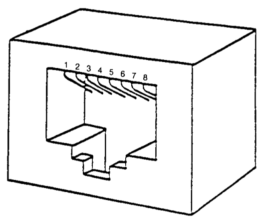


Figure 14–21—MAU MDI connect

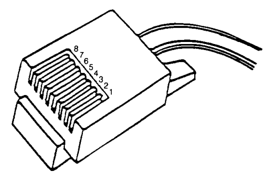


Figure 14–22—Twisted-pair link segment connector

PC network interfaces are wired as DTE's and so RX+ and RX- are (differential) inputs.

Unfortunately, there are two conventions for wiring these connectors using color-coded 4-pair cables. In one convention (EIA/TIA 568A) TD+ and TD- are connected to orange-white and orange respectively while pins RD+ and RD- are connected to green-white and green wires respectively. In the other convention (EIA/TIA 568B) the green and orange colours are swapped (just the wire colours, not the pins).

If the cable was connected according to the 568B standard you *may* need to use the green-white and green wires instead of orange-white and orange.