Experiment 3 Documentation

This is a Bus Snooping Attack. For this experiment the required instruments and software’s are:

1. Haha Board(circuit Board)
2. USB cable
3. Analog Discovery 2
4. Atmel Flip
5. Laptop
6. Waveforms

This was a simple experiment in terms of hardware connection and everything. We first connected the board to the computer. Downloaded the given files from the website. Then programmed the microcontroller with the given .hex file. Since my accelerometer was working previously from the self test program before, programming “U\_1.hex” file worked for me. After pressing the MCU RST for a couple of times after programming, the values in the LED started changing again

A green circuit board with yellow lights

Description automatically generated

This is what the Haha board looked like after programming U\_1.hex file into the microcontroller. After that I turned off the Haha board. After that I connected oscilloscope probes to the “MISO” and “SCK” pins of the SPI header on the Haha board. And those pins were connected to the Ground pins of the analog discovery 2. There were several pins in the Analog Discovery 2. After connecting the pins from Haha board to Analog discovery 2, I opened waveforms to start the experiment. I turned on the board and set the setting for the waveforms accordingly. This is what the experiment looked like:

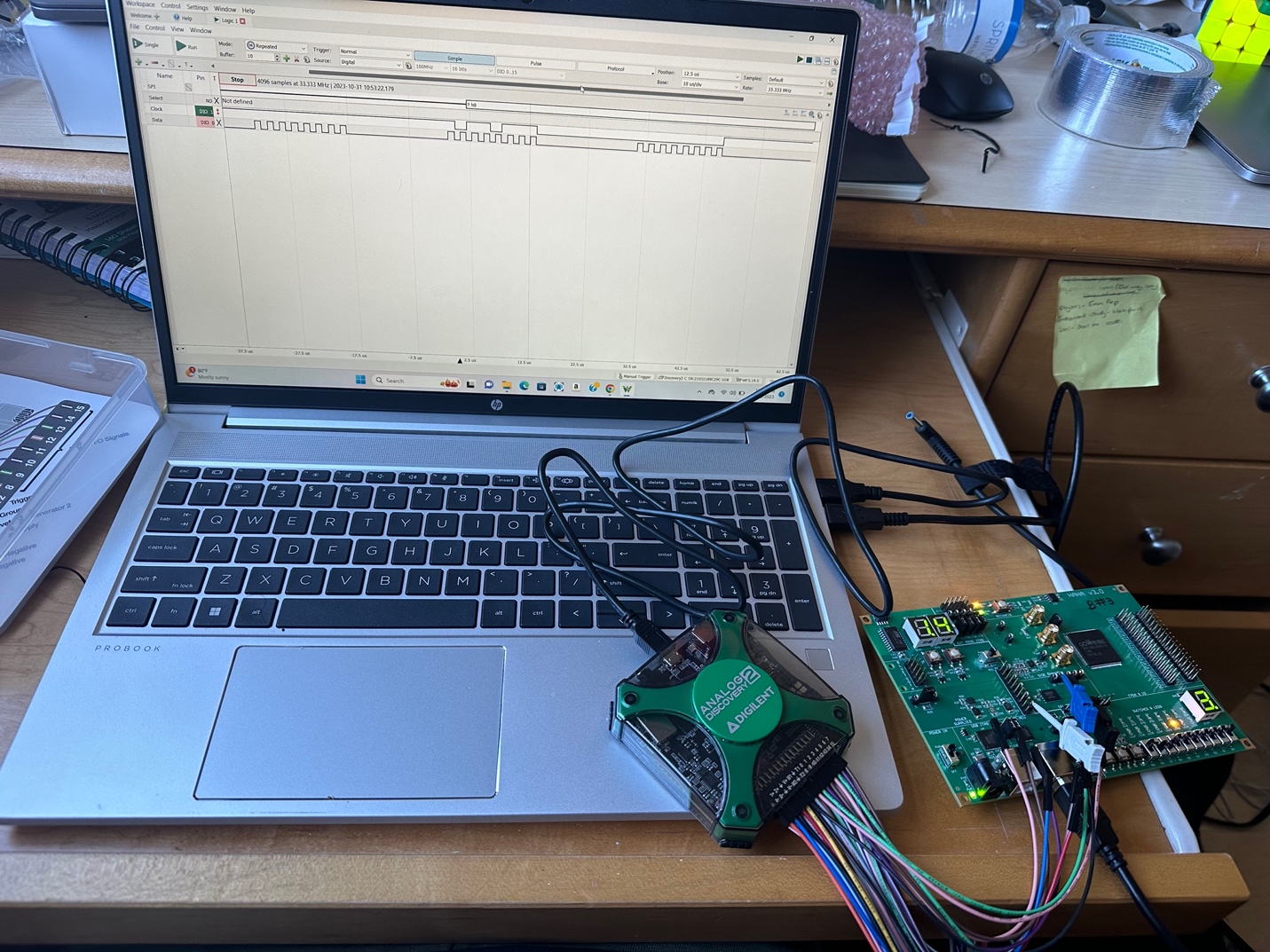


Image: Ground pins connected to MISO and CLK

1. Which hex file did you use to run this experiment?
   * I used the U\_1.hex file to run this experiment.
2. Which line would you say is the clock line? How fast is the clock running at?  
   - The first line which was set on edge is the clock line. The line which changes less frequently compared to the other line or the line which has longer bumps is the clock line . It showed that it processed 4096 samples at 33.33MHz.
3. Which line would you say is the data line?  
   - The data line is the second line which has more frequent bumps.
4. Is it possible to set up the oscilloscope to both trigger and decode this data line bus? Please submit a screenshot of the scope triggering on a start condition and successfully decoding the transmitted data?  
   - F
5. Can you describe the general data packet format? a. For U\_1.hex: i. If you tilt the PCB in one direction (try different angles) how does the data packet change? How does it stay the same? b. For U\_2.hex: i. What is the pattern in the data? Does it ever remain the same? Why would a pattern like this show up on the bus?  
   - a. For U\_1.hex:

i. The tilt of the PCB would change the accelerometer data reflected in the data packets, which would show variations in the values corresponding to the accelerometer's axes.

b. For U\_2.hex:

i. The pattern in the data represent periodic signals or a repeating sequence that is part of the device's normal operation. Patterns repeat when the data represents a regular status update..

1. If you didn’t have conveniently labeled traces to probe, how you would figure out which traces would you probe if you needed to figure out how the board worked?  
   - If a circuit board doesn't have conveniently labeled traces to probe, figuring out which traces to probe requires a methodical approach. Here’s how you might go about it:  
   1. Visual Inspection: Begin with a close examination of the PCB to identify components and their connections.   
   2. Datasheets and Pinouts: Refer to the datasheets for the major components on the board.  
   3. Circuit Functionality: Consider the functionality of the board.  
   4. Signal Tracing: Use a multimeter in continuity mode to trace the paths.  
   5. Power and Ground: Identify the power supply and ground traces, which are typically wider to accommodate higher currents.  
   6. Test Points: Look for exposed pads that may serve as test points.  
   7. Logic Analyzer: Use a logic analyzer with multiple probes to test several points.  
   8. Oscilloscope: An oscilloscope can help identify active lines by showing signal waveforms.  
   9. Specialized Equipment: In some cases, you might use X-ray imaging or a PCB scanner that can help reveal inner-layer traces in multi-layer PCBs.  
     
   By combining these strategies, you can build a map of the critical traces on the PCB and gain insights into how the board operates, even without labeled traces.
2. How would you prevent such attacks? If the data transmitted over this bus was critically important and could not be intercepted, how would you secure the communications? How would you then debug any issues that arose because of your fix?
   * To prevent bus snooping attacks, encryption and authentication mechanisms can be implemented in the communication protocol. Additionally, physical security measures such as shielding the bus lines or using tamper evident packagign can be employed. If data security is paramount, secure communication protocols can be used. Debugging encrypted communication can be challenging but logging and error handling mechanisms can help identify issues without compromising the security of the data transmission.

