EE/CprE/SE 491 WEEKLY REPORT 10

11/15/2024 – 11/21/2024
Group number: sdmay25-04
Project title: Wireless Mesh Network for Pesticide Spray Monitoring and Mapping

Client: Claussen Lab- Iowa State University

Advisor: Nathan Niehart

Team Members/Role:
Software Team
- **Ashley Falcon:** IDEs and Microcontrollers, Group Communicator
- **Drew Scheidler:** Mesh Networking; Note Taker
- **Henry Hingst:** Mesh Networking; Group Leader

Hardware Team
- **Hector Perez Prieto:** Microcontroller; Circuit Design and Testing
- **Yok Quan Ong:** Circuit Design and testing; Microcontroller
- **Wesley Smith:** Circuit Design/Simulation; Microcontrollers; Note Taker

---

Weekly Summary
- **Hardware Team**
    - Worked on finalizing the wheatstone bridge and differential amplifier portions of our circuit
        - Working on ways to make the voltage output more linear when graphed with resistance values
        - Got an equation that relates voltages to resistances
        - Found resistance values that are easily accessible to purchase
    - Looked in adding all components of our circuit into one final circuit
        - Includes adding voltage regulators and diodes that protect the ADC from receiving too much or too little voltage
    - Graphing the maximum and minimum voltages from the worst case scenarios
        - Both 1% and 5% worst case scenarios
- **Software Team**
    - Lab work with SD card component
        - Experimented with circuit setup to get MC to communicate with SD card component

        - Troubleshot example code
        - Wrote code from scratch using documentation and examples as reference
        - Successfully communicated between MC and SD card component
        - Created and wrote txt files on SD card using MC code
    - Lab work with ADC
        - Set up circuit to route 3 different power sources into 3 ADC channels/pins
        - Wrote and tested code to take in readings from 3 power sources and display mV readings to console in human-readable format
        - Wrote and tested code to take in readings from 3 power sources and write mV readings to SD card in human-readable format
    - Determined how sending and receiving wifi packets is done in the mesh network framework provided by Espressif
        - The example sets up a wifi event handler that does the physical transmitting and receiving and provides the code with a handful of different data structures for wifi packets
        - We'll use this handler by calling esp_mesh_send() & esp_mesh_recv() to send and receive wifi packets
    - Continued modifying the mesh example to use 802.15.4 instead of 802.11
        - Theoretically, this will be achieved by simply calling esp_wifi_set_protocol() after the initialization of the wifi event handler
        - It may need to be called multiple times in order to set the proper protocol for each wifi interface (Mesh nodes use both Access Point and Station interfaces)

Past week accomplishments

- **Ashley Falcon:**
    - Continued debugging SD card example card
        - Compiled simplified version that excluded "fluff" code (such as error messages for applications not being used)
        - Expanded understanding of the interwoven nature of the microcontroller's processes
    - Experimented with ACD features
        - Determined how to sample 3 different voltage sources from various channels rather than just 1
    - Intertwined ADC oneshot and SD card code
        - Took samples from various voltage sources
        - Stored measurements in a formatted file
        - Saved file to the SD card

- **Drew Scheidler:**
    - Developed code for MC communication with an SD card, leveraging documentation and examples
    - Successfully enabled MC to create and write text files on the SD card
    - Designed and implemented ADC functionality to measure and log voltage from three power sources
    - Routed power sources to ADC channels, displaying and recording human-readable voltage data to the console and SD card
- **Hector Perez Prieto:**
    - Finalizing wheatstone bridge
        - Looked into what was making our voltage vs resistance graph nonlinear and if there was any way to make it more linear
        - Helped find the best fit polynomial equation that covers all the data points that we need
            - Data points came from worst case scenarios ran in LTSpice
- **Henry Hingst:**
    - Determined how sending and receiving wifi packets is done in the mesh network framework provided by Espressif
        - The example sets up a wifi event handler that does the physical transmitting and receiving and provides the code with a handful of different data structure for wifi packets
        - We'll use this handler by calling esp_mesh_send() & esp_mesh_recv() to send and receive wifi packets
    - Continued modifying the mesh example to use 802.15.4 instead of 802.11
        - Theoretically this will be achieved by simply calling esp_wifi_set_protocol() after the initialization of the wifi event handler
        - It may need to be called multiple times in order to set the proper protocol for each wifi interface (Mesh nodes use both Access Point and Station interfaces)
- **Yok Quan Ong:**
    - Simulated the worst case based on the latest version of the Wheatstone Bridge design
        - Output the worst case and plot in measure vs step output
        - Run for 1% and 5%
        - Export the data and plot to the curve
    - Look up some voltage regulator that regulates 5v and -5v

- **Wesley Smith:**
    - Worked on changing bridge resistances to standard resistances that we can purchase while maintaining our output resistance range
    - Simulated the voltage regulator and the voltage inverter together in place of our voltage sources in the bridge
        - Noticed slight voltage drop, documented and can easily be accounted for
    - Worked in creating a polynomial function to describe how our bridge detects voltage
        - Took the data output from LTspice as a text file
        - Input it into MATLAB
        - Extracted a curve and function to describe a curved line of best fit for the curve
        - Should help when programming the ADC

NAME	**Individual Contributions**	**Hours this week**	**Cumulative Hours**
Ashley Falcon	SD/ADC coding and lab work	**10**	**63**
Drew Scheidler	SD Card Component, ADC, and MC lab work	**10**	**65**
Hector Perez Prieto	Final Circuit Design and Data Analysis	**7**	**61**
Henry Hingst	Mesh Packet Transmission & 802.15.4 Modification	**6**	**56**
Yok Quan Ong	Circuit Simulation, Worst Case, Data Analysis	**7**	**59**
Wesley Smith	Circuit Simulation, Data analysis with MATLAB	**7**	**63**

Plans for the upcoming week

- **Hardware Team**
    - Try to finalize hard resistances for the bridge
    - Simulate the voltage drops for the regulators and average those
    - Documents all the voltage drops
    - Figure out how to set up the protection diode for the ADC
    - Have a block diagram for the overall design
    - Simulate how much measurement changes with the changes in supply voltage
    - Quantify the uncertainty as much as we can

- **Software Team**
    - Sensor and System Design
        - Finalize sensor labeling system (e.g., 1-9)
        - Define system-level description of node operations, including data flow and sampling behavior (e.g., count samples vs. wait for intervals)
    - Data Storage
        - Determine the desired file structure (e.g., one file per sensor)
        - Finalize file format (e.g., .csv with headers for voltages, digital values, and metadata)
        - Include raw digital values and plan for descriptor table use to convert values to mV
    - Code Development and Testing
        - Calibrate ADC for accurate voltage readings
        - Abstract ADC and SD card functionality into reusable code modules
        - Test ADC-SD interaction to ensure reliable data logging
    - Microcontroller Identification and Communication
        - Implement a host ID system to distinguish between MCs
        - Develop a strategy to transmit minimal data and perform conversions at the base station
    - SD Card and Timing
        - Research and finalize SD card formatting to ensure files are structured and readable
        - Address timing challenges using existing examples and explore averaging techniques as needed
    - Metadata and Context
        - Define and include necessary metadata in stored data (e.g., sensor ID, timestamps)
    - Determine how to properly create a wifi packet data structure to be used by esp_mesh_send()
        - Once successful, create a test program to send and receive wifi packets
    - Test attempts to determine if esp_wifi_set_protocol() is truly all that is needed to modify the wifi mesh framework to use 802.15.4 instead of 802.11

<u>Summary of weekly advisor meeting</u>

- Professor Neihart provided feedback on both the hardware and software teams.
- Hardware:
    - The Wheatstone Bridge is good
    - Try to find a hard resistor values if possible
        - Avoid resistors in series for final design

    - Voltage regulator simulation is good
        - The simulation has voltage drop, it's noted but shouldn't effect too much
    - Put protection diode before the ADC
        - Try to figure out how to set that up
    - Create a block diagram for an overall design
    - Know the uncertainties in the component variation
    - Simulate how much measurement changes with the changes in supply voltage
    - Quantify the uncertainty as much as we can

- Software:
    - Sensor Labeling: Finalize sensor numbering system (e.g., 1-9)
    - System Description: Define node data flow and sampling behavior (e.g., count samples or wait for intervals)
    - Data Storage:
        - Use text files, likely in .csv format, with headers for voltages and digital values
        - Store one file per sensor, including raw digital values and metadata (e.g., ID, timestamp)
        - Consider descriptor tables for converting raw values to readable formats (e.g., mV)
    - Microcontroller Identification: Use host IDs to distinguish between devices
    - Transmission and Conversion: Transmit minimum data process conversions at the base station
    - SD Card Format: Ensure files are structured and readable, though not necessarily human-readable
    - Timing and Averaging: Address timing using existing examples; consider averaging where needed
    - Metadata: Include identifiers (sensor ID, time) for contextual clarity