Senior Design Meeting Agenda — Microgrid Team

10:00AM Thursday, March 11th, 2021
205 Stinson Remick
Meeting Leader: *Maia*
Secretary: *Kelsey*

Design Changes/ Concerns

- Realised we can't charge 12V battery with 12V
    - Considering rectifiers
    - Considering using 10V at battery storage
- Unsure about amperage of inverters to fit other system requirements
- Found necessary physical connections and wiring
- Finalize parts to purchase for testing subsystems

Creating testing experiments for subsystems

Generation Subsystem

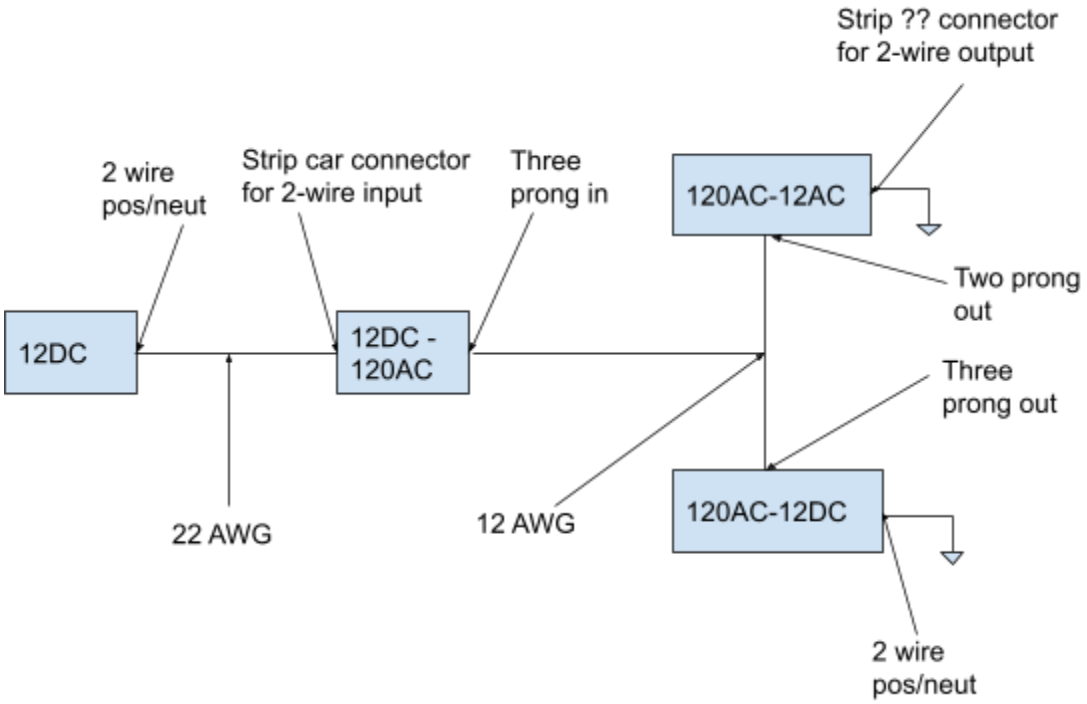
- Switch DC/DC converters
    - Connect (using a breadboard) the high-voltage side of the DC-DC converters to a 12V power supply. Connect the output of the DC-DC converter to one of the PCBs we made last semester and see if the light comes on, indicating that it is being powered correctly.
- Solar Panels
    - Using one cell, connect (using a breadboard) the positive lead to one end of a small, grounded resistor and the negative lead to ground. Use a multimeter to measure the voltage and current and compare to our expected values of 1.5V and 400mA.
    - Options for panels to buy:
        - First choice: out of stock
        - Second choice: pack of 4
        - Third choice: pack of 10
        - Fourth choice: original choice
- Bypass diodes
    - Test that the diode works properly by setting the multimeter to Ohms and checking that it conducts in the forward direction but not in the reverse direction.
- Fuses (630mA and 1A)
    - Run current through the fuses and record when they break (using an ammeter or a multimeter's continuity test setting). For a successful test, the current at which the fuses break should be above the max current expected from the subsystem the fuse is connected to but not too much higher than that value. In other words, the current should match the fuse's current rating.
- Test Solar light
    - Buy a charge controller?

- Blocking diodes
    - Test that the diode works properly by setting the multimeter to Ohms and checking that it conducts in the forward direction but not in the reverse direction.
    - To check that the diode can handle the max reverse current that it should see in this project, connect the negative lead of the diode to a 24V power supply and measure the current, which should be 0 if the diode is working correctly.

Distribution Subsystem

- Inverter
    - Use 22 AWG wire to connect 12V DC (potentially one of the lead-acid batteries) to the inverter.To do this, strip the car-charger port off of the inverter to leave the positive and negative leads. Connect 12 AWG wire to positive and negative lines the opposite side of the inverter with a three-prong head and measure output voltage & current.
- Rectifier
    - Plug the three-prong end of the rectifier into the wall for ~120V AC. Connect 12 AWG wire to the opposite side of the inverter and measure output voltage & current, expecting around 12V DC.
- Transformer
    - Plug the two-prong end of the transformer into the wall for ~120V AC. Connect 22 AWG wire to the positive and negative ends on the transformer output and measure output voltage and current.
- Switches
    - Wire up 12AWG wire on either side of a 4-terminal switch, connect one end to 120V AC. Toggle switch and measure output voltage and current with switch both open and closed
    - Four-Pin Switch
    - We could also use two-pin switches with the T-Taps
    - Schafer didn't seem concerned about internal resistance so I guess we won't be either?
- Motor
    - Connect 12V AC from transformer to motor leads, load motor (with a pencil, makeshift fan blade, or something similar) and measure current and voltage.
    - Expectation: ~4W Real Power
- Resistor + LED
    - All the 2W AC LEDs are 2-pin - use a T-Tap connector
        - Wire T-Taps
        - Looping Wiring System
    - 2W, 2-Pin LEDs Some on Amazon
        - 1.5 Watts for half price - would have to just increase resistor size
    - Connect 12VAC from transformer to wired up resistor/LED combo, measure current and voltage. Expectation: 4W Real Power
- Connections & Two-Line System
    - In wiring 12 AWG and 22 AWG wire for the other components, use heat shrink or the connectors Sylvia found

- Two-prong outlet connectors would be extremely helpful
  - 2-Prong Outlets
  - 2-Prong Plug



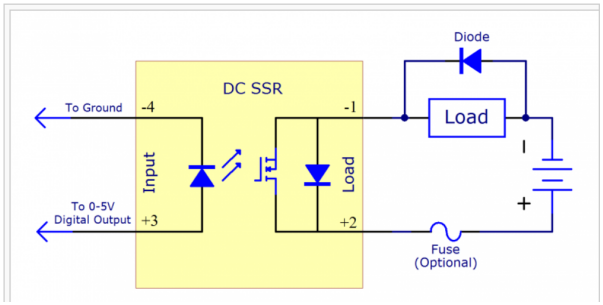
Storage Subsystem

- Battery
  - Charging
    - Charge current anywhere from 1A to 4A
    - Need voltage greater than 12 V => revisit generation, use this sheet for calculations
    - May need current limiting resistor (use above sheet to determine)
  - Discharging
  - Parallel wiring
    - Need F2 wire connections. Will use ring connectors and a screw to wire them
- DC/DC converter
  - Connect (using a breadboard) the high-voltage side of the DC-DC converters to a 12V power supply. Connect the output of the DC-DC converter to one of the PCBs we made last semester and see if the light comes on, indicating that it is being powered correctly.

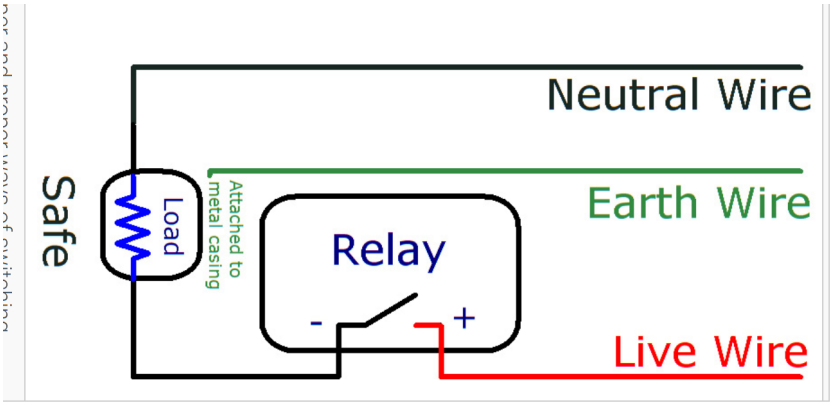
Control Board Subsystem

- SSR (w/ breadboard)
  - Connect the SSR using the technique given by Jameco HERE, where Vin1 (-) is connected to GND and Vin2 (+) is connected to a digital I/O that is either set high

or low; Vout3 (-) is connected between the - battery terminal line the 1ohm R load and Vout4 (+) is connected to the battery + terminal



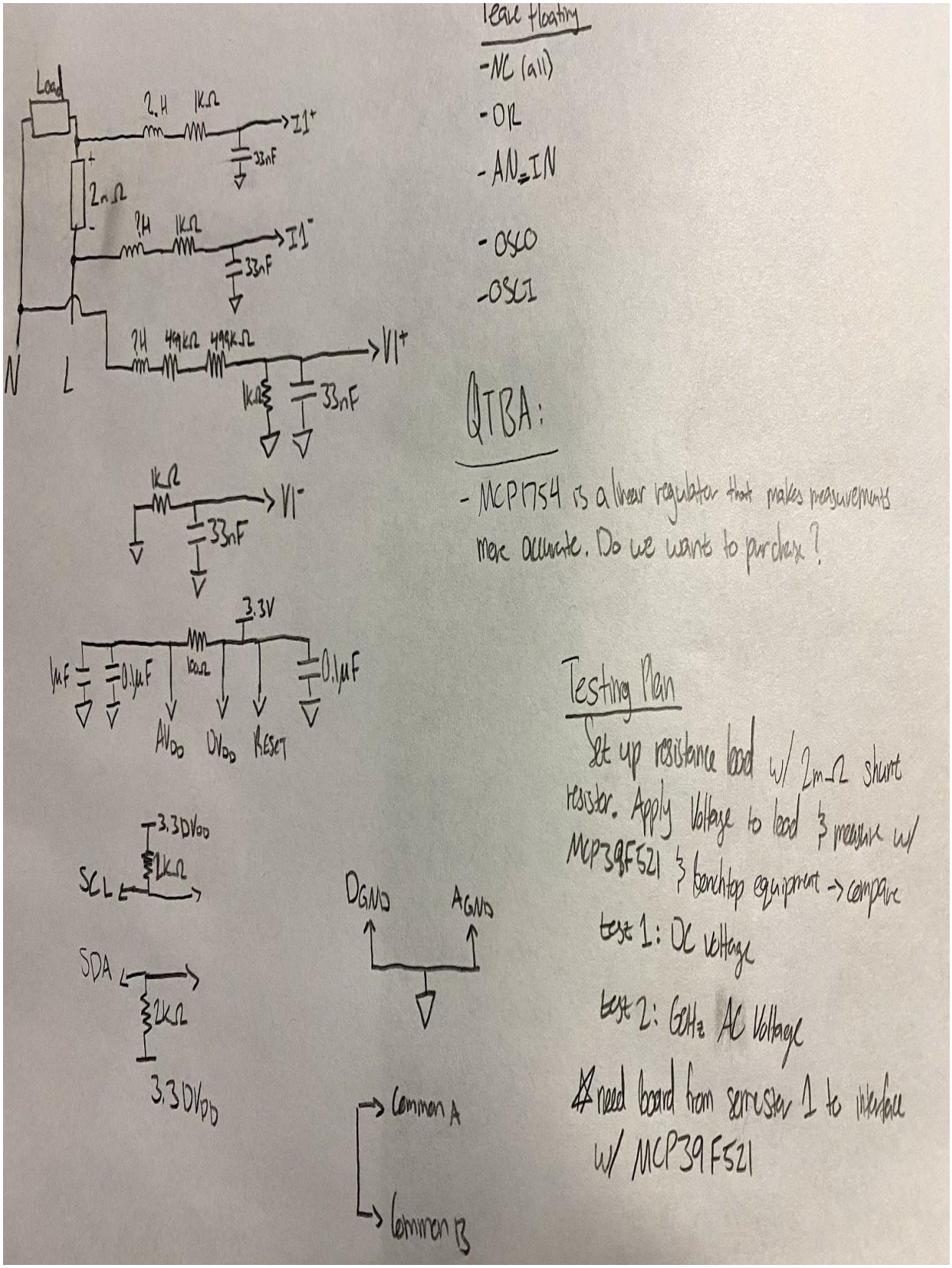
- Can use this setup on a breadboard for testing, need to clarify/talk through how to wire it in the actual system



- [Link for wiring in the final build]

- **LoRa** (w/ breadboard)
  - Use 2 LoRas, each with an adapter board, 915MHz antenna, microcontroller (use boards from last semester with pinheaders), and accompanying circuitry (capacitors and resistors to solder onto the adapter board that can be found in the lab)
  - Send an I2C message from the transmitting microcontroller to LoRa A, LoRa wirelessly transmits to LoRa B, I2C to receiving microcontroller
    - Set up virtual GUI for LoRa that Maia found on Microchip to adjust LoRa settings (set up LoRa for transceiving and adjust center frequency, etc)

Power Monitoring Subsystem



leave floating

- NC (all)
- OPL
- AN_IN
- OSCO
- OSCI

QTBA:

- MCP1754 is a linear regulator that makes measurements more accurate. Do we wants to purchase?

Testing Plan

Set up resistance load w/ 2mΩ shunt resistor. Apply voltage to load & measure w/ MCP39F521 & benchtop equipment -> compare

test 1: DC voltage

test 2: 60Hz AC voltage

*need board from semester 1 to interface w/ MCP39F521

- Power Chip (w/ breadboard)

    - Use surface mount adapter board to connect pinheaders and use with breadboard
- Power Chip communication with Microcontroller (w/ breadboard)
- Power sample wiring
- Mulitplexers (w/ breadboard)
    - Use last semester board to control multiplexer

Display Subsystem

- UART communication between Matlab and board (test with breadboard)
    - Setup includes physical setup with tx/rx from uart module adapted to usb format (can use pickit adapter and wires from sr design lab)
    - Code to configure and establish connection:
        - Create a serial port object using the built in `serialport` function

- The function has two input values, the specified communication port and the baud rate
- For instance, creating an object would look like `s = serialport('COM1', 9600);`
- Dot notation is used to configure and display the property values i.e `s.BaudRate`, `s.Port` etc.
- The serial port object and the instrument communication settings must be identical for reading data.

    - Configure other serial port communication settings.

- `Parity`, `DataBits`, `StopBits` and `Terminator` will depend on the format that data will be transmitted from the PIC microcontroller.
- `Getpinstatus` reads the serial pin status and returns s struct giving information on whether the pin is clear to send or receive data.
- Since the display system will only be reading and not writing data, the `configureTerminator` function can be configured appropriately.

    - Read data from microcontroller

- Depending on how the data is processed on the microcontroller, the functions `read` and `readline` will be used.
- `Readline(s)` reads ASCII data until the first occurrence of the terminator from the serial port connection and returns data as a string without the terminator
- `Read(s, count, datatype)` reads the number of values specified by count in the form specified by datatype from the serial port connection s. for all numerical datatype types, data is a row vector of double values. For the text type datatype uses 'char' or 'string'
- Will likely test with simple binary I/O from LED signal

- Simple GUI representation of test data in real-time