Power Consumption Test Plan

Equipment needed: Joulescope JS110, Breadboard Prototype w/battery management circuitry

Values needed:

1. Combined MSP430 and SmartMesh power consumption during sleep (no sensors active & SmartMesh inactive)
2. Combined CO2, MSP430, and SmartMesh charge consumption for a CO2 measurement
3. CO2 power consumption while sleeping between measurements
4. Combined PM2.5, MSP430, and SmartMesh charge consumption for a PM2.5 measurement
5. PM2.5 power consumption while sleeping between measurements
6. Combined Anemometer, MSP430, and SmartMesh charge consumption for an airspeed measurement for both Ultrasonic and Hot Wire anemometers
7. Anemometer power consumption while sleeping between measurements

Instructions for Combined MSP430 and SmartMesh power consumption during sleep:

1. Unplug all sensors from system, and unplug 5V, 3.3V, and RXD jumpers from EnergyTrace System on MSP430
2. Set up power supply and Joulescope to supply 3.7V to battery management circuitry, with the Joulescope set up to monitor current consumption to the battery management circuitry
3. Set up host SmartMesh node
4. Change line 6 from #define powerTest false to #define powerTest true, if necessary
5. Run “indoor\_air\_quality\_v2.ino”
6. Turn on power supply
7. Ensure node connects to host node, and transmits two packets consisting of (48879, 48879)
8. Turn off power supply
9. Start collecting data on Joulescope
10. Turn on power supply
11. Let ~5-10 packets of (48879, 48879) transmit to host node
12. Stop data collection on Joulescope
13. Analyze JouleScope data, and obtain average power consumption from the low, flat portions of the graph (when the MCU is in sleep). Store this number in the Power Consumption Test Results google sheet under “Typical Power Consumption (off)” for “Micro Com”. Make sure to include correct units with prefix (e.g. mW for milliWatts, uW for microWatts)

Instructions for Combined Sensor, MSP430, and SmartMesh charge consumption, along with Sensor sleep power consumption

1. Unplug all sensors, except sensor under test, from breadboard, and unplug 5V, 3.3V, and RXD jumpers from EnergyTrace System on MSP430
2. Set up power supply and Joulescope to supply 3.7V to battery management circuitry, with the Joulescope set up to monitor current consumption to the battery management circuitry
3. Set up host SmartMesh node
4. Turn on power supply
5. Run “indoor\_air\_quality\_v2.ino”
6. Change line 6 from #define powerTest false to #define powerTest true, if necessary
7. Ensure node connects to host node, and transmits one packet consisting of (48879, 48879), followed by sensor initialization and one packet of sensor data
8. Turn off power supply
9. Start collecting data on Joulescope
10. Turn on power supply
11. Let ~5-10 packets of sensor data transmit to host node
12. Stop data collection on Joulescope
13. Analyze JouleScope data, and obtain average charge consumption from the higher portions of the graph, which should be approximately the same amount of time as the time needed for that sensor to collect its data (~15 seconds CO2, ~30 seconds PM2.5, ~10 seconds Hot Wire Anemometer, ~3 seconds Ultrasonic Anemometer). Store this number in the Power Consumption Test Results google sheet under “Charge Consumed (On)” for that sensor. Make sure to include correct units (e.g. mC for milliCoulombs, uC for microCoulombs)
14. Using the same JouleScope data, obtain average power consumption from the lower, flat portions of the graph (when the sensor is asleep). Take this number and subtract the power consumption obtained for the MSP430 and SmartMesh sleep power consumption. Store this number in the Power Consumption Test Results google sheet under “Typical Power Consumption (off)” for that sensor. Make sure to include correct units with prefix (e.g. mW for milliWatts, uW for microWatts)

Instructions for calculating sensor measurement periods

1. Export the Power Consumption Test Results google sheet. Save it as "Power Consumption Test Results - Power Consumption.csv" in the directory /Documentation/Power Testing
2. Run the python script “powerData.py”. Follow the prompts it gives you. It will calculate the measurement periods for each sensor, and display other useful information about the components power consumption.
3. If you would like, there are a few easy to edit values in the “powerData.py” script. These are available in the first ~20 lines of the program, and are as follows:
   1. BAT\_MAH: capacity of your battery(s), in mAh (milliAmp-hours)
   2. BAT\_VOLT: nominal voltage of your battery(s), in volts
   3. BAT\_NUM: number of battery(s), if wired in parallel
   4. MAX\_T: maximum allowable sensor measurement period, in hours
   5. GOAL\_DAYS: Goal battery life, in days