

# EPS Subsystem - Next Steps

## Pivotal Tasks

- Update I2C commands to reflect updated hardware and firmware.
  - The newest version of the I2C manual we have been instructed to use is revision 2.2. The previous implementations are based on revision 2.0
  - The read commands are mostly the same, but command 23 onward have some additional functionality and command numbers.
    - Specifically, some of the defaults and inputs/outputs have been condensed into bitwise commands
  - The write commands are again mostly correct, however, with some clarifications. The previous manual had command numbers that were off by one (1 indexed versus 0 indexed), the new manual clarifies that the write commands are 0 indexed. The reserved/not implemented command is now implemented, so there is a new command, this shifts around the numbers of some of the commands.
- Test solar cell input and recharging
  - Once M2I is done with fabricating the solar cells, 4-pin molex cables can be used to connect each of the 6 faces to their corresponding inputs on the EPS module. However, one of the faces will not have a solar array, so nothing will be plugged into it.
  - Starting with just one of the solar arrays, connect it to the EPS and see if you can read substantial voltage coming in on that particular interface. Make sure to have the M2I members who are working on the solar arrays present.
  - After you are sure that voltage is being registered on the EPS, we want to make sure that the battery is actually being charged by this method. I recommend you devise some sort of set up and have the battery charge overnight. It will obviously have to be a low battery in the first place so you can detect a change.
- Make I2C command functions pass by pointer so that values can be returned rather than just printed out.
  - Currently, most of the EPS commands simply print out to the UART line for debugging. It would be useful to change these to pass-by-pointer methods so the value can be stored and retrieved. The value returned by the functions should still indicate if the method succeeded or not.

## Stretch Goals

- Battery Capacity calculations, including discharge and recharge
  - The EPS module does not inherently have a method by which we can determine the total energy capacity of the battery
  - The simplest way to determine the “health” (meaning the approximate capacity) of the battery is to look at the battery voltage, or Vbatt. The user manual has voltage values which correspond to the End of Charge (essentially the max value) and the overdischarge (essentially minimum battery voltage value) voltages. This is the current plan and requires basically no further implementation, however it isn’t as robust.
  - A more advanced way would be to take measurements of the EPS at different points in time and then integrate between the instantaneous power of these two measurements, calculating the energy discharged by the satellite and recharged by the solar arrays
    - I already have some sample calculations that show how this works, but it only measures the Vbatt bus. This may or may not be enough. I am uncertain if the Vbatt voltage and current commands take into account all current draws on the system (for example a low impedance resistor on the 5v bus would be a good test to see if that affects the Vbatt bus).
    - The main issue is in regards to the recharging. I am really confused by how we are supposed to use the given commands to determine the instantaneous power generated by the solar cells.
      - The given commands for each axis (X, Y, Z) are: X+ current, X- current, and X-axis voltage.
      - I don’t understand what the X-axis voltage value refers to. Maybe the voltage difference between the X+ and X- faces?  
Documentation I have seen doesn’t make this clear.
- More robust health checks for determining if the EPS is functioning correctly. I categorize this as a stretch goal because for the most part, if the EPS isn’t working, then the satellite is a paper weight in space.
  - Currently our metrics are: if certain battery heaters are on/off, Vbatt voltage, and if implemented, the battery capacity.