EPS Interfacing Function Library

# Description:

Build a function library using an Arduino (or another microcontroller if you like) that can be used (or easily translated) by the On-Board Computer (OBC) to interface with EPS system.



Figure – Clyde Space EPS

# Design Specifications:

## Electrical:

1. Communication with EPS is done through a I2C bus (2 wire bus, data and clock)
2. EPS is to be treated as a slave that they will be controlled by the OBC
3. Use Arduino (maybe a raspberry pi?) and its I2C bus to interface with EPS.

## Software:

1. Using the I2C commands and definitions found in the EPS datasheet, create a series of functions that the OBC can use to send commands to the EPS
2. Can be done using the Arduino IDE language (C based), however avoid using classes, instead stick with standard function definitions.
3. You may use the Arduino I2C libraries
4. The structure of the Functions is open-ended, however think of methods to reduce length of code by avoiding repeated code.
5. The OBC will unlikely be able to directly use the code, however it will provide pseudo code and an architecture for quick translation to the OBC’s respective language.

# Left to be Answered:

|  |  |
| --- | --- |
| Requirement in Question | Answer |
| Do all commands need to be made? | Ideally yes, however some functions maybe deemed to be unneeded |
| What Language is the OBC using? | At this time, it will likely be written in C, however this may change. |
| Do all commands need to be made in separate functions? | Not exclusively, if there is another way to approach each command to reduce code, then by all means try. |

# What to do to Complete Project:

1. Find the section on I2C commands in EPS datasheet
2. Research I2C Communication Protocol
3. Obtain competency in programming Arduino’s (or another microcontroller), including:
   1. Programming structures (for loops, if statements, defining functions etc.)
   2. Using Arduino I2C libraries
4. Use another I2C device to practice using I2C protocol
5. Begin writing code for EPS Interfacing Library
   1. **Note:** To avoid damage, the EPS can only be used to test code under special supervision. Initially, testing code on EPS will be restricted until a later date
   2. Using an oscilloscope or I2C converter to monitor I2C outputs is a great way to check and see if the right signal is being sent.
6. If interested, more information on the OBC architecture can be shared by Alex White, the SAT Computer Project Manager

# Important Notes:

* Use Github (Power Systems repository) to store completed or in progress work. Avoid using Slack too send key files that others can use. Likely will get lost in active chats.
  + Note: Properly organize files on Github so we don’t have a mess! Use folders with good names!
* Any new and better ideas should be brought up to the team, such as cost reductions, design changes, better parts, additions etc.
* For clarifications, just ask!!
* Everything in this document is subject to change.