Structure for the I2C Interface

This document outlines the development cycle of the I2C Interface project. It outlines the .py files included in the whole project, and what each one actually does.

The filestructure is organized as such:

- i2ctool
  - bin
    - i2ctool.py
    - csvlib.py
    - helperFunctions.py
    - SMWClient.py
    - Device List.csv
  - dev
    - lots of random things used by the original developer, not actually necessary to the function of the program

i2ctool.py

Acts as the "main" function. This orchestrates the command line interface and calling the required functions. The main function runs in a while loop until the `exit` command is given. It executes three functions: `board_level`, `bus_level`, `device_level`. The functions handle which user commands are available, based on the current board/bus/device "level". Certain commands allow you to move between the levels, with `return` always moving up a level.

SMWClient.py

Contains a single class `SMWClient`. This class handles calling commands on the BC and returning the text output.

A single instance of this class is called in `i2ctool.py`, and stored in the variable `smw`. This variable is passed around the entire program for calling whatever commands may be necessary.

csvlib.py

Contains classes for: generic `bus` object, generic `device` object, and a `voltageRegulator` object. These classes are used throughout `i2ctool.py` to keep things (relatively) simple and clean. Each supported command in the i2ctool interface should have a corresponding function in a class in `csvlib.py`.

helperFunctions.py

This file is exactly what it sounds like. It contains functions which help prevent `i2ctool.py` from becoming too code bloated.

Usage

Copy the entire `bin` folder to a directory on the SMW. `cd` to the `bin` folder and run the command `python i2ctool.py`. This will kick up the i2ctool interface. Input the cname for the [Scout] card you wish to interface with, and then use the `help` command to get usable commands at each stage. At the time of writing (10 Aug '18), I cannot promise what will and won't work.

The CSV File

The `Device List.csv` file contains an exhaustive list of buses and devices found on various boards. A single row corresponds to a single device. The contents of the columns are: bus number, bus name, hex-address, device name, part#, device type. *(For device type, unless it is specified "vr" (voltage regulator) it will treat the device as a default device class)*

At the start of the program, the csv is read into the variable `exh_bus_list` and consistently referenced.