

# 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.

## main.py

Acts as the main method. Will orchestrate the command line interface and calling the required functions. Is gonna have a lot of includes in it.

## xtlogin.py

Will contain a class for interacting with the blade controller. This class should be usable from anywhere, and will likely be passed as arguments to basically everything there is.

## csvlib.py

Stores all the information on the individual I2C devices. Handles reading from the CSV and comparing to the devices actually present on the board. Needs to have a good level of forward-compatibility or else this application will be basically useless. This file will probably have the most code in it, just because of the size of everything involved.

For reading in the I2C devices:

- Have the script run through the /sys/bus/i2c/devices/\* folder and populate everything that way.
- Now check the read-in list against the excel sheet and form your present/not present list that way.
  - Should create a framework that functions, then figure out how to make it correct.

## usage.py

This file will have all the functions called by main.py. It'll be a time. Have fun with all of this! I believe in you!

# Functions

---

- Global
  - exit
  - set\_cname
  - get\_ps
  - adv\_ps
  - cmd
  - help
- Board
  - get\_board
  - read\_excel
  - goto\_bus
  - buses/bus\_list/i2cdetect

- walk\_bus
  - pop\_by\_type
- Bus
  - walk\_bus/walk/pop/populate
  - get\_device
  - disp\_types
  - send\_cmd
- Device
  - watch/monitor
  - get\_reg
  - set\_reg
  - get\_prop
  - set\_prop