

# Project Description

By David Adams and Samuel Graves

## Overview:

For our final project we decided to recreate homework #5 but over sockets in python. This was done with the Raspberry Pi acting as a server, holding all the information and access to the parts, and the Host Computer acting as a client, requesting information and control without direct access to the data. This was done via the <socket> library and used homework #5 as a framework for the class, menu, and file structure.

## Sockets:

As explained above, the Pi acts as the server and binds with the created socket with the hard-set Host IP and Port. The Host Computer acts as a client, connecting with the server with which the server then binds to it. All information is sent using the thread's send function, by first using the utf-8 format to encode the required data. The data is then received by using the thread's receive function with a set buffer size. It is then decoded to be used as needed.

## SNX4XX:

This is the part class used in homework #5 and is integrated and iterated upon. Most of the data members and methods stay the same, but what and how methods are used are changed due to the nature of the project. Only the server has access to the SNX4XX to create a level of artificial security that would normally exist in a real scenario. This means whenever sending/receiving object information, it needs to be properly formatted, encoded, and parsed. Similar to HW#5, this project has a menu with six options that the client interacts with. When the client chooses an option, both the client and server find the chosen option within their cascading if statements. Then, for example, in the "Print the part list" option, the client gets ready to receive the information while the server loops and sends the information to the client. This menu will continue to loop on both sides until the client chooses option 6: "close the program".

## Aftermath:

We had **a lot** of unforeseen issues with this project that ended up putting us back at least a week before even being able to get started. We originally wanted to work with a Pi Car, before the car decided to not work for a week, killed one of our Pi's, and killed our momentum. After getting a new Pi we were left with only a week left and a lot of other work piling up. We decide to switch over to socket programming which took some growing pains in itself. When we were finally able to get into the flow of using sockets, it was already time to submit. We had ideas that we wanted to implement and were working on implementing that we had to remove for submission. Overall, we are happy with the project and the skills that we learned while working on it but disappointed with the scenario that we were in. We hope you enjoy our project and the work that we put into it.