For the application of voice control, it is mainly consisting of five parts: IOS/Android client, command-line client, the server and the Arduino board.

The server side can be seen as a bridge between clients and the Arduino board, and it receives the message and command from the clients and then through the serial port, the commands can be sent to the board so that household appliances can be controlled.

The server is implemented in C and the interface for clients of the server is socket. Firstly, the server starts running before any of clients for waiting for connection. Once it gets a certain connect, it will use “fork()” to have a child process to handle this communication, and the parent process will continue to listen other connections. So, this server can process multiple connection and control requirements. In the child process, there is a “split string” functionality to recognize which is the message that the client sent and which is the command that the client brings. For instance, the data structure is like this synopsis:

[message]/[command]

The message is the certain words that client says, and the command such as “0”, “1” is defined to enable and disable the Arduino board, and the character parts “A/a”, “B/b” etc. are used to control the certain devices. Simply, “open the kitchen light/A” means “turn on the kitchen light”. In the server, this string will be divided into two parts according to the slash. The first part will be written into the log file, and the second part will be written into the serial port and then the board will read the command to execute specific command.

In conclusion, the technologies used here are:

1. TCP/IP socket: for the communication between clients and server.
2. Multi-process technique: the function fork can implement handling many clients.
3. Splitting message and command: the function “strtok” can separate specific commands.
4. Serial port read and write: implementing the communication between server and board.