

General Architecture

Definitions

Base Station – A lightweight server that will be in the users house. This device controls and manages devices.

Device – A physical object that the user wishes to control with their iOS application. Examples would be garage doors, and sprinkler systems.

iOS – The operation system that is present on iPhones and iPads. In this context iOS refers to the iOS 6.0

Resolution Server – A server operated by our client. The server will have a database that will store associations between a serial numbers and IP address.

Serial Number – The MAC address of a base station.

Overview

The Remote Home system will consist of four main components an iOS application, a Resolution Server, a Base Station, and individual Devices. The iOS application will act as a remote control for the devices. This application will store the associated base station in a local database. Users will be able to register new base stations, modify base stations, and control devices from this application. This application will make use of the Resolution Server to associate a serial number from a base station to a dynamic IP address. The Base Station will be a server in the users home that will control individual devices. The Base Station will have a web interface where a user can set up new Devices, make new users, set permissions, and modify properties of existing Devices. The Base Station will periodically update the Resolution Server with the external IP address. Devices are individual components that will have hardware modification so that the Base Station can send messages to perform actions. The Resolution Server will be operated by the client and will be a known point in the cloud by both the iOS application and the Base Stations. This will require the operator of the Resolution Server to have a domain name or static IP address. The Resolution Server will have a database of IP address and serial number pairs that will be passed to iOS client on request, and updated by the Base Stations.

Base Station Architecture

iOS Architecture

Overview

The iOS application will act as the remote control for the Remote Home system. The application will consist of views. The application will also use a SQLite database that will store the base stations that have been registered. When the application is first started the “first time registration” controllers will run, this will require the user to register a valid base station. If the user has at least one base station registered the application will start the “main view” controllers. This is a UINavigationController view controller that will present a list of base stations. In addition a add button will be in the upper right hand corner of the list view. If the user presses this button they can add a new base station. The user can swipe across a cell of a base station to delete the base station from the SQLite database. If the user selects a base station a new list will populate with the individual devices the user will have an edit button in the upper right hand corner of the list view. If the user presses this button they will be presented with a form where they can modify the properties of the base station. If a user selects a device they will be presented with the correct device controller.

First time registration (iOS)

Overview

The purpose of the first time registration is to force the user to register a base station with the phone so that they can control devices. We will display this view if there are no base stations registered in the SQLite database.

Instruction View Controller

The purpose of this view controller is to display a scroll view with instruction embedded in it. These scroll view will instruct the user to set up their base station and connect devices. At the bottom of the view controller will be a button so that the user can advance to the registration view.

Registration View Controller

The purpose of this view controller is to allow the user to register a new base station. The view will consist of three text boxes and a register button. The view controller will check to see if all three fields are filled before connecting to the resolution server. If any of the fields are empty a UIAlertView will be displayed telling the user to fill out the empty field.

If all three fields are populated and the user clicks the register button the device will attempt to make a TCP connection to the resolution server on port 80. At this point the device will start a timeout timer. If the TCP connection fails to open before the timeout fires the system will close the connection and present a UIAlertView to the user. The alert will instruct the user to check their connection and/or try again later.

If the connection is successful the server will send the connection DDNSConnected (See “Bidirectional iOS to Resolution Server Communication”) signal to the phone. At this point the phone will send the HRHomeStationsRequest with the serial number provided by the serial number field. At this point the Resolution Server will look up the serial number. If the Resolution Server finds the serial number it will respond with HRHomeStationReply with the correct IP address and serial number. If the Resolution Server fails to find the serial number it will respond with HRHomeStationReply with ‘null’ for the IP address and the correct serial number.

If the phone receives a null for the IP address it will present the user with an UIAlertView. This view will inform the user to check the serial number and make sure that they set up the base station correctly. If an IP address is sent the device will register the device in the SQLite server and present a UIAlertView to the user. This view will inform the user that the device was successfully registered. At this point the device will go into the main view.

Resolution Server Architecture

Overview

The Resolution Server stores the IP address of each Base Station along with a unique identifier for that station and allows the iOS application to ask for the IP address of a Base Station. The server will run as a background daemon on a computer with a domain name or a static IP address. It will use port number 8128. The IP address and identifier of the Base Stations are stored in a SQLite database. The server itself will be written in Python 2.7. It will be written in a few classes. The Finder classes will handle the SQLite database. There will also be a class to handle connections.

SQLite Database

This database has a single table, called devices, with two columns: ID and IP. Both columns are text and the column ID is the primary key.

Finder Class

The Finder class abstracts database interaction with the rest of the program. It will have methods that roughly correspond to request made to the server. It may also have some utility methods.

Communication Protocols

Bidirectional iOS to Resolution Server Communication

Overview

This communication protocol defines the messages that will be passed between an iOS client and the Resolution Server. This will allow the iOS client to loop up IP address for a base station from a serial number.

DDNSConnected

This message is passed when the Resolution Server acknowledges a connection from a iOS client.

```
{ "DDNSConnected": [ { "Connected": true } ] }
```

HRHomeStationsRequest

This message is sent from the iOS client to the Resolution Server. This message is a request for IP addresses based on serial number. "(StationDID)" field will be replaced by a base station serial number.

```
{ "HRHomeStationsRequest" : [ { "StationDID" : "(StationDID)" }, { "StationDID" : "(StationDID)" }, ... ] }
```

HRHomeStationReply

This message is sent from the Resolution Server to the iOS client. This will tell the iOS client the association between serial numbers and IP addresses. "(StationDID)" is the station serial number. "xxx.xxx.xxx.xxx" is the IPv4 address of the base station. If the station cannot find the IPv4 address it will fill the "xxx.xxx.xxx.xxx" field with a null.

```
{ "HRHomeStationReply" : [ { "StationDID" : "(stationDID)", "StationIP" : "xxx.xxx.xxx.xxx" }, { "StationDID" : "(stationDID)", "StationIP" : null }, ... ] }
```

Unidirectional Base Station to Resolution Server Communication

Overview

This communication protocol defines the messages that will allow the Base Station to update Resolution Server with its current IP address.

HRHomeStationUpdate

This message is sent from a Base Station to the Resolution Server. It contains the station's unique identifier and its current IP address.

```
{“HRHomeStationUpdate”:  
  {“StationDID”:“(StationDID)”,“StationIP”:“(xxx.xxx.xxx.xxx)”}}
```