Remote Home

Introduction

- Team
  - James Wiegand (Team Lead)
  - Christopher Jensen
  - Joshua Kinkade
  - Brian Vogel

- Client
  - June Knight L-3 Communications
  - Dr. Jeff McGough

Problem

- Smart Phone Controlled Home
  - Smart Phone Controlled Garage Door Opener
  - Smart Phone Controlled Sprinkler System

- Create a custom smartphone app, that is able to control a household device, such as a garage door or sprinkler system.

Remote Home Framework Overall Design

- Remote Home Framework Communication Protocols
  - JSON
  - TCP/IP
- Remote Home Framework Device Management
  - Add new base station
  - Authenticate with a base station
  - Select a device
- Device controllers
  - Extendable
  - Each device has it's own controller

Remote Home Framework Communication Protocols

- iOS to DDNS
  - Query IP addresses of base stations
- DDNS to iOS
  - Return IP address requested
- iOS to Base Station (Connection)
  - Send password over TLS (Transportation Layer Security)
- Base Station to iOS (Connection)
  - On success send all connected devices (online or offline) and their status
  - On failure respond with bad password error
- iOS to Base Station (Control)
  - Once authenticated, send over data to control device
- Base Station to iOS (Control)
  - Send status of device being controlled (if applicable)

iOS to DDNS Protocol

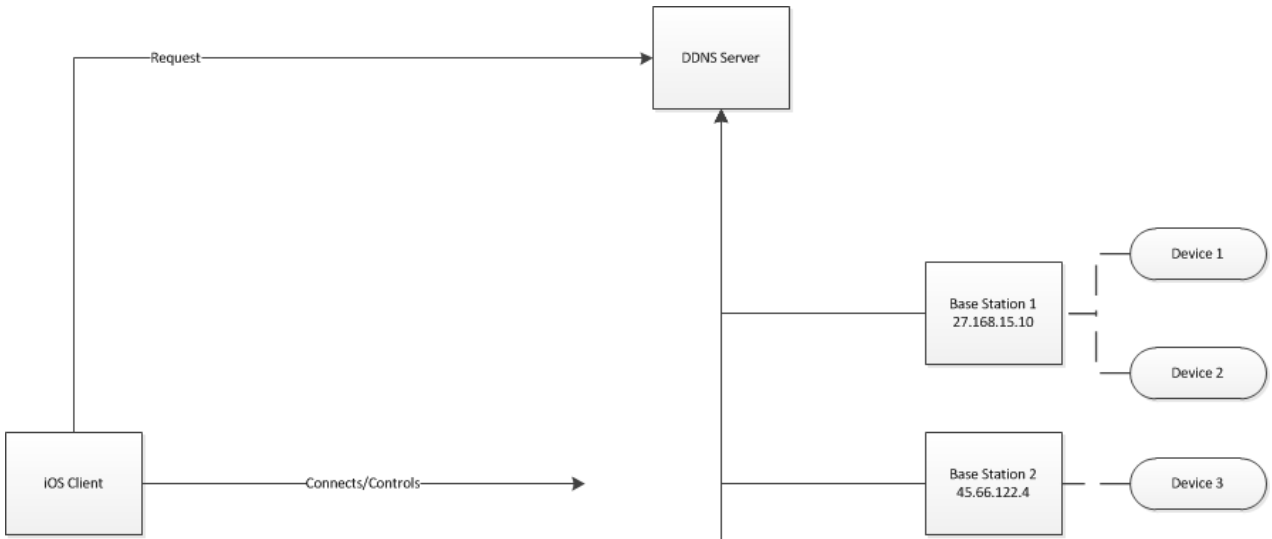
- iOS to DDNS

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

- DDNS to iOS

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

Remote Home Communication Overview



When a Base Station is turned on a user enters a password.

The base station is then registered with the DDNS.

The iOS client requests the IP address of all registered base stations in the clients SQLite database.

The iOS client negotiates the connection to a base station.

If the connection is successful the base station sends a list of devices.

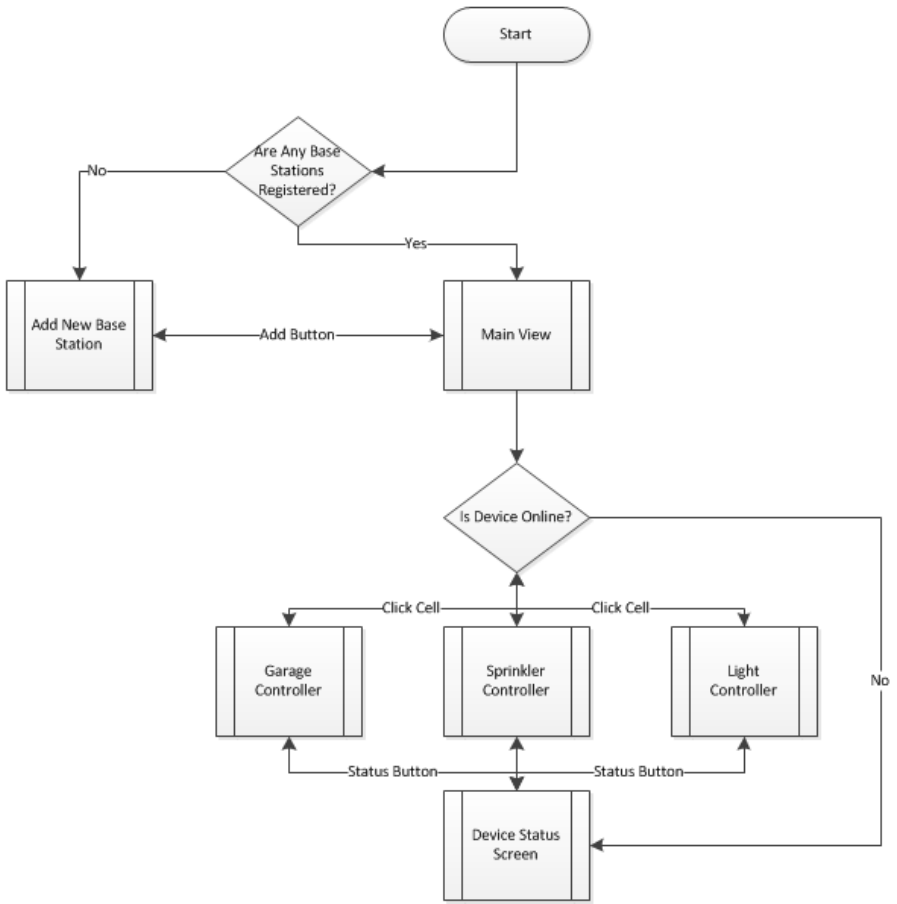
The iOS client can now
control a device connected to the base station

Remote Home Framework User Stories

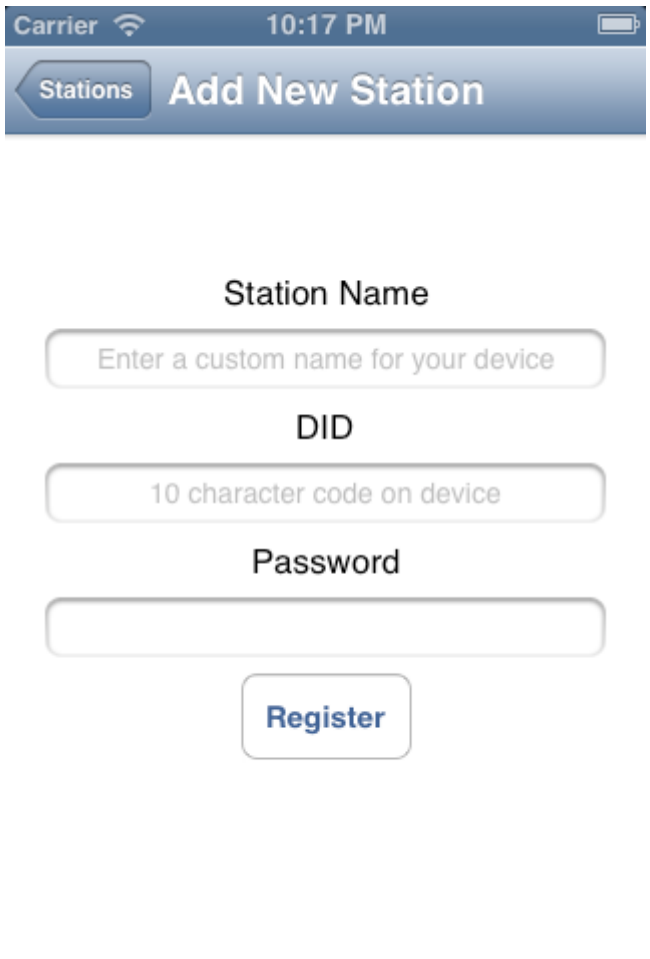
- Main View
  - List view of base stations can drill down to devices
  - Have a button to add a new base station
- Status View
  - If there is an error show the error code and description here
- Register View
  - First view when the app is opened for the first time
  - Fields to name a base station, enter a password, and
- Garage Door Opener
  - Progress bar to show the status of the door (percent open or closed)
  - Button to open/close the door

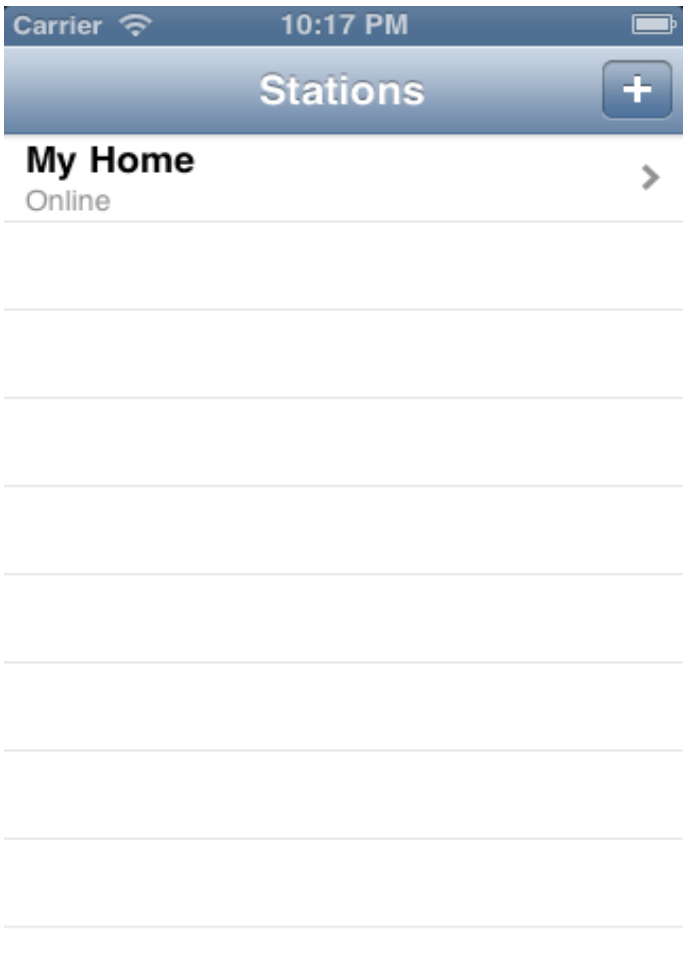
Remote Home Framework Overview

Remote Home (iOS) Program Flow

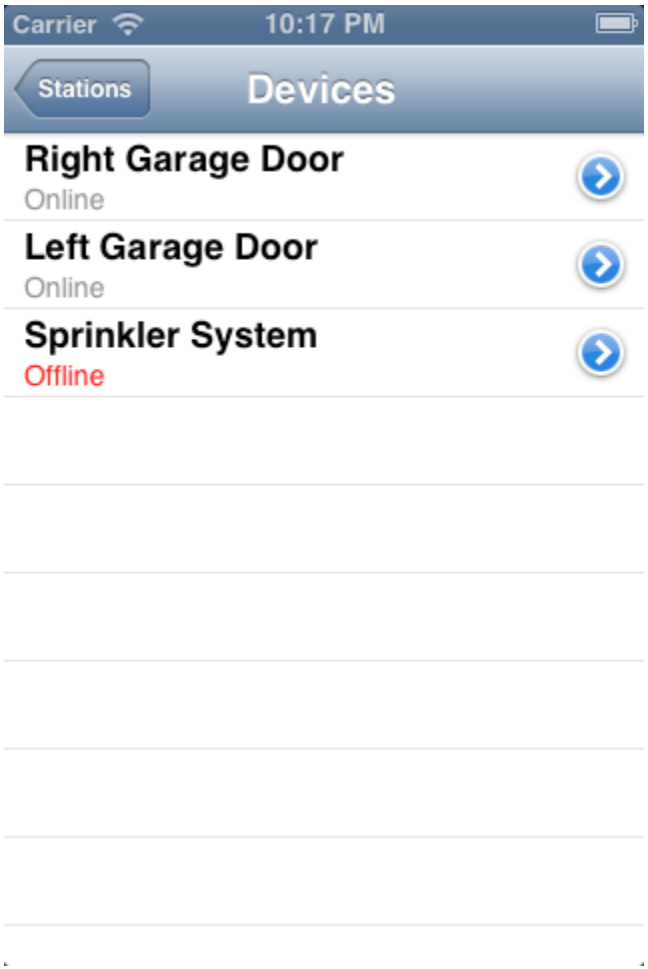


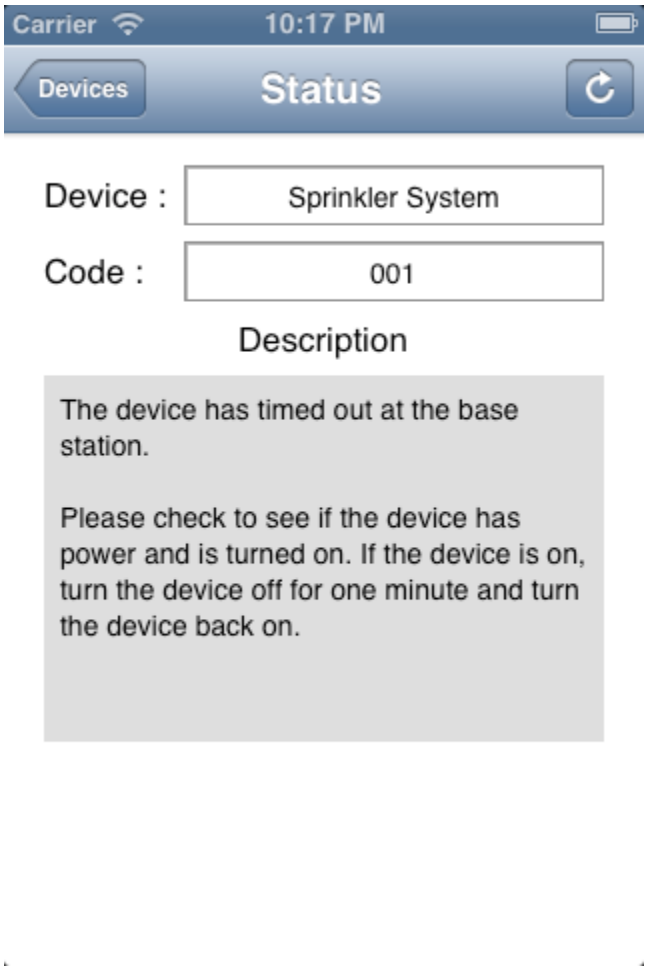
iOS App Prototype





iOS App Prototype





Remote Home Framework Backlog

- Main View
  - UI
  - Communication protocol for receiving connected devices from base station
  - SQL transaction for registering and deleting base stations
  - Navigation view control logic
  - Controller logic
- Status View
  - UI
  - Communication protocol for status
- Register View
  - UI
  - Communication protocol for DDNS
  - Communication protocol for registration of base station
- Garage Door Opener
  - UI
  - Communication protocol to send instructions to the base station to open the door
  - Communication protocol to receive state of the garage door from the base station
- Misc
  - Create plist (property list, a type of flat file) to associate error codes to an English description of the error
  - Create a communication helper to query the DDNS server, used if the app is resumed from a suspended state.

Requirements

- DDNS: Must resolve DID to IP address, maintain list of active DID/Address pairs.
- Gateway: Must report IP updates to DDNS. Must track attached devices and act as a “middle-man” for communications between phone and device.

Hardware Risks

- Bluetooth Signal Range
- Arduino compatibility

Testing

- Garage controller can be tested using standard Arduino debugging tools
- Communications protocol can be tested through traditional "Echo" application
- Gateway can be tested by monitoring on-board logs

Testing

- DDNS: make many requests with virtual phones, check for data consistency
- iOS: use unit testing framework

Budget

- 2 Arduino boards with Bluetooth shields- $200
- iPhone - $600
- 8 channel SSR - $75
- Apple Developer Account - $100
- Garage Door Opener - up to $200
- Total – up to $1175

Deliverables

- iPhone application
- Software for Home Station
- Software for DDNS server
- Prototype Arduino control board
- Documentation

Questions?