Bluetooth Low Energy Beacon Remote Sensors and Thermostat Bridge

Presented by: Mitch Grogg

December 11, 2015

CPET 491 – Senior Design II

Computer Engineering Technology

Instructor: Professor Paul Lin

Advisor: Professor Hal Broberg

Outline

- Introduction
- Problem and Solution
- Requirements
- System Design
  - Hardware
  - Software
- Integration and Testing
- Cost and Schedule
- Lessons Learned
- Demo
- Question and Answer

Summary

3

- Wireless remote temperature measuring system
  - Bluetooth Low Energy (BLE)
  - Location dependent
  - Eddystone proximity-aware BLE beacon format
- Proof of concept for employer
  - WaterFurnace International

Problem

4

- Multi-zone heating and cooling issues
  - Thermostat cost
    - $150+ / thermostat
    - Ugly
  - Labor intensive installation
- Connected devices
  - Becoming more numerous
  - Consumer “app fatigue”
  - Questionable security

Solution

5

- Remote Sensors
  - Small, wireless
  - Low power
  - Inexpensive
  - Connect without a mobile app

- Thermostat Bridge
  - Aggregates sensor data
  - Data stays local
  - View data with mobile devices
  - Hidden out of sight

Requirements

6

- Three remote sensors
  - Battery Powered
  - BLE Eddystone beacons
  - At least one transmission per 5 seconds
  - User-assignable zone ID number
- Thermostat bridge
  - BLE receiver
  - Web server
    - Dynamic web frontend
  - Parse and display remote sensor values
    - Temperature
    - Status

Hardware Design

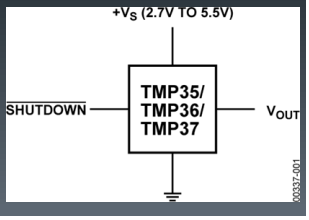
7

- Remote Sensors
  - Rigado BMD-200 BLE Module
    - Nordic Semiconductor NRF51822 MCU
      - ARM Cortex-M0
      - 10-bit ADC
      - Digital I/O peripherals
    - Ceramic Antenna
    - Passive components
    - FCC Certified

Hardware Design

8

- Remote Sensors Continued…
  - Analog Devices TMP36
    - Low voltage, precision centigrade temperature sensors
    - +2.7 VDC to +5.5 VDC operation
    - Less than 50 µA supply current
    - Less than 0.5 µA Shutdown current
    - 10 mV/°C linear voltage output
    - SOT23-5 surface mount package



Hardware Design

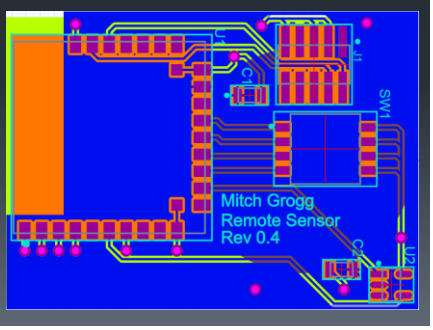
9

- DIP Switch
  - Four position
  - Used to assign a “Zone ID”
  - Connected to BMD-200 internal pull-up GPIO
- Battery
  - CR2032 3VDC Lithium coin battery

Hardware Design

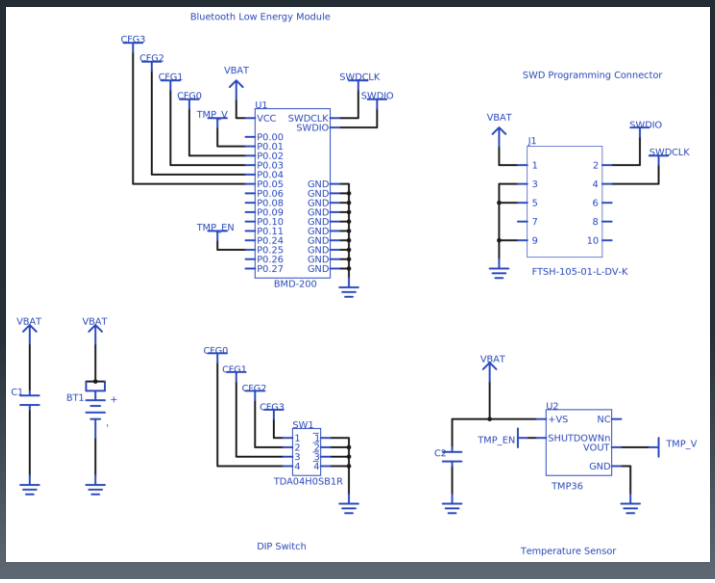
10

- Connector
  - SWD programming/debug connector
  - 10 pin
- Printed Circuit Board
  - Manufactured by OSH Park
  - 2 layer PCB
  - 1” x 1.5” x 0.063”



Hardware Design - Schematic

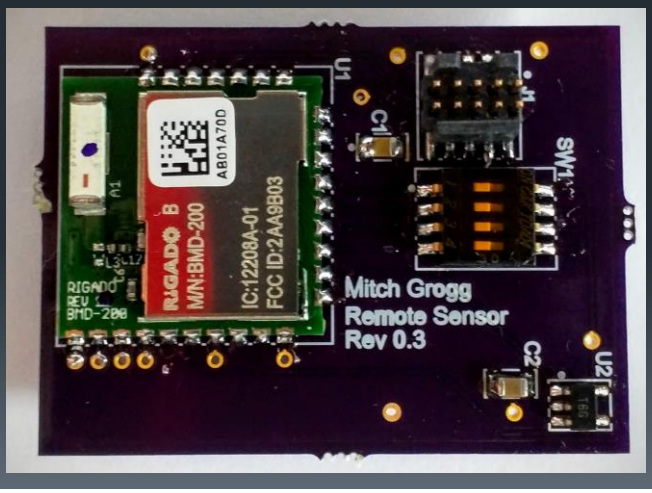
11



Hardware Design

12

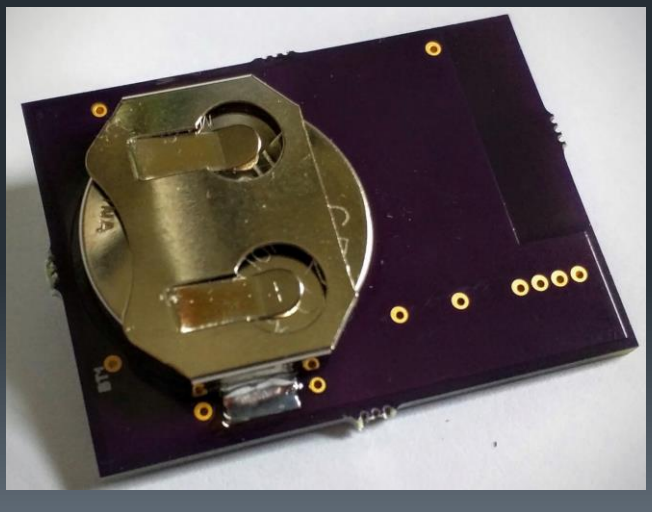
- Complete PCB - Front



Hardware Design

13

- Complete PCB - Back



Hardware Design

14

- Thermostat Bridge
  - Raspberry Pi single board computer
  - Bluetooth 4.0+ USB adapter
  - Ethernet network connection
  - 5 VDC AC power converter

Software Design
Remote Sensors

15

- Programming Tools
  - Keil µVision 5 IDE
  - Keil MDK-ARM Compiler
  - Nordic Semiconductor
    - nRF51 Software Development Kit (SDK) version 10.0.0
    - S110 SoftDevice version 8.0 BLE protocol stack and API
  - Segger J-Link Flash Programmer/Debugger

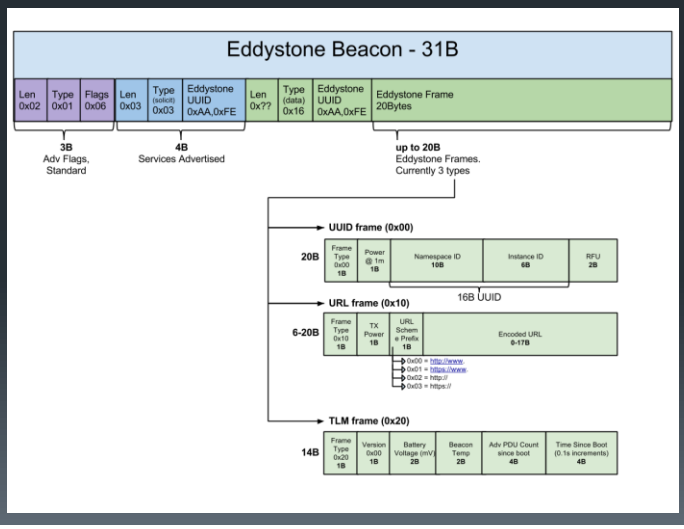
Software Design
Remote Sensors

16

- Eddystone™ Bluetooth Low Energy Format
  - An open beacon format from Google
  - Proximity beacon
  - Capable of working across Android and iOS devices
  - Eddystone-URL
    - Transmits a URL
    - Received by compatible mobile apps
    - Sensor Zone ID and ADC value encoded in URL
      - http://tstat.biz?<zone ID>&<ADC value>
- Example URL With Data:
  - `http://tstat.biz?1&202`

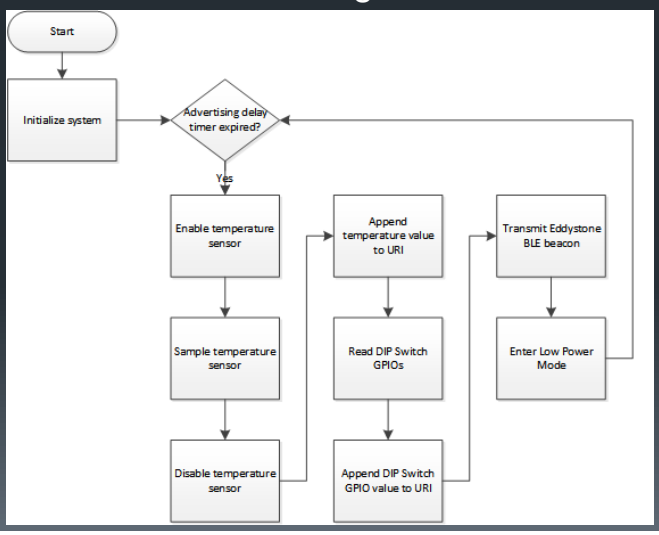
Software Design Remote Sensors

- Eddystone-URL Specification



Software Design Remote Sensors

- BLE Beacon Transmitting Software Flow Chart



Software Design Thermostat Bridge

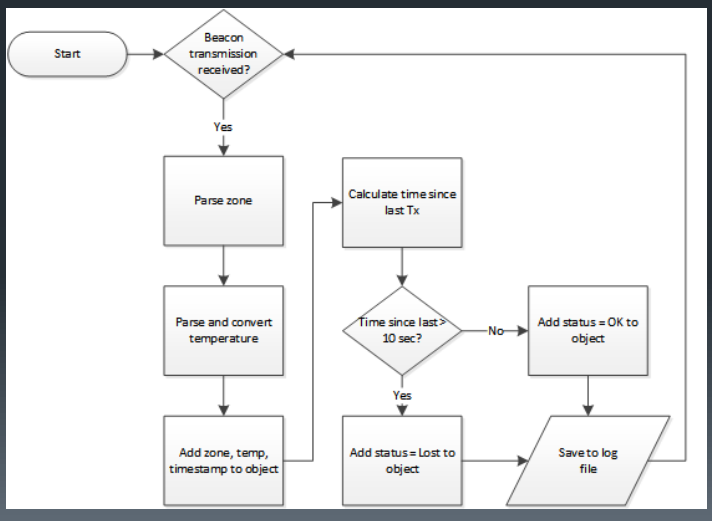
19

- Node.js
  - Open source libraries
    - eddystone-beacon-scanner package
      - Convenience methods for Eddystone format
      - Receives beacon transmissions
    - fs package
      - Allows for file system manipulation
    - http-server
      - Lightweight web server
      - Configured to serve web page over local network

Software Design Thermostat Bridge

20

- BLE Beacon Receiving Software Flow Chart



Software Design
Thermostat Bridge

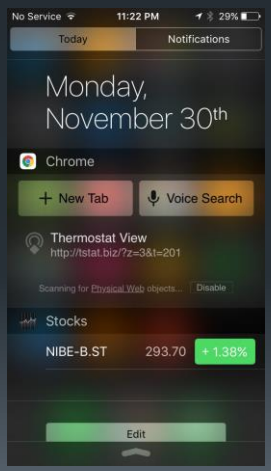
- Web Framework
  - Skeleton.css
    - Open source
    - Lightweight, responsive HTML/CSS framework
  - jQuery
    - Popular, open source
    - Client-side Javascript library

Remote Sensors

Zone 1	Zone 2	Zone 3
Temperature: 71	Temperature: 64	Temperature: 69
Status: Lost	Status: Lost	Status: OK

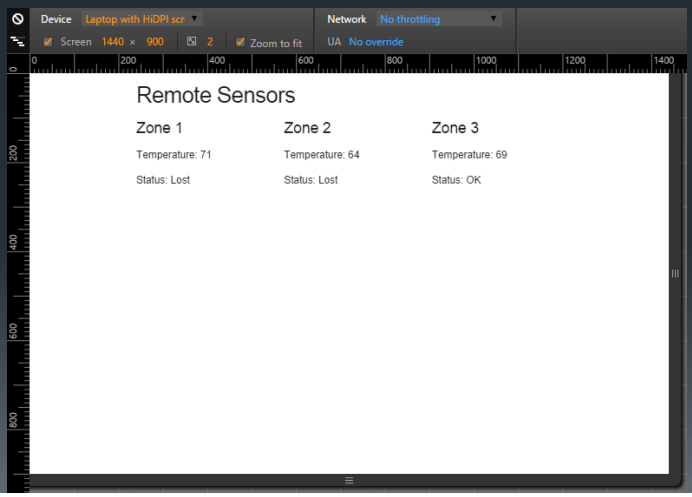
Integration and Testing

- Eddystone Transmission Test



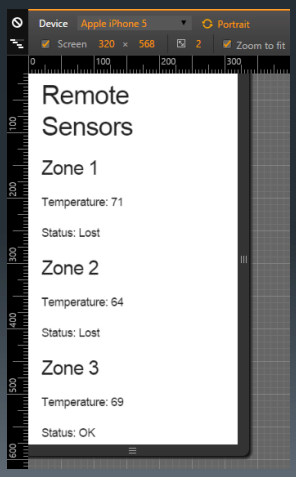
Integration and Testing

- Responsive Website Test (Temperature, Status)
- Laptop simulated



Integration and Testing

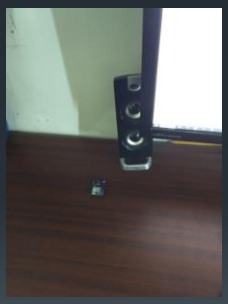
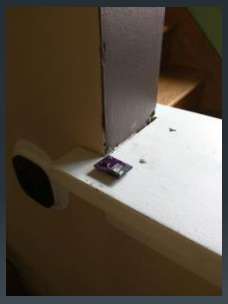
- Responsive Website Test (Temperature, Status)
- iPhone 5 simulated



Integration and Testing

25

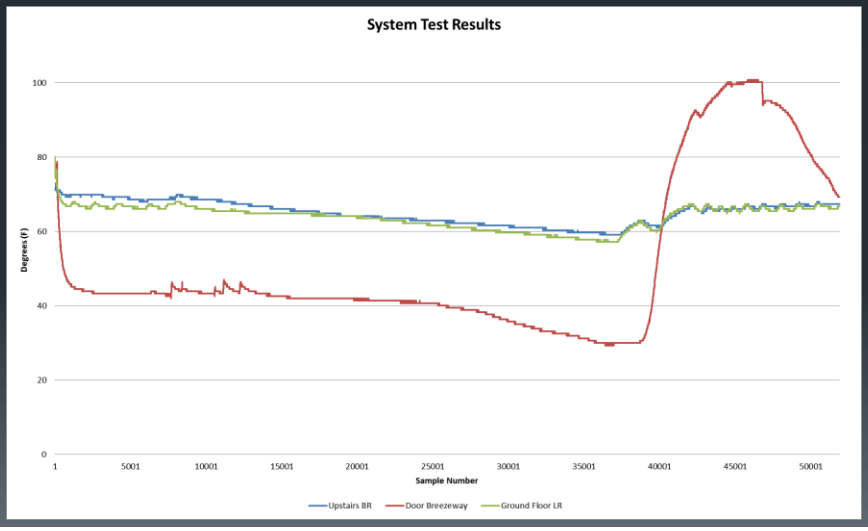
- Temperature Range Test
  - Requirement: 40°F - 90°F


Integration and Testing

26

- Temperature Range Test



Cost

- Underestimated
- Did not factor in Segger J-Link, ordering multiple extras

	Estimated			Actual		
Item	**Qty**	**Cost Each**	**Estimated Cost**	**Actual Qty**	**Cost Each**	**Actual Cost**
Raspberry Pi	1	$ 40.00	$ 40.00	1	$ 37.98	$37.98
Segger J-Link	0	$ 86.97	$ -	1	$ 86.97	$86.97
CAT5e Ethernet Cable	1	$ 10.00	$ 10.00	0		$0.00
Bluetooth Low Energy USB Adapter	1	$ 10.00	$ 10.00	1	$ 14.99	$14.99
Bluetooth Low Energy Dev Kit	1	$ 75.00	$ 75.00	1	$ 59.95	$59.95
Bluetooth Low Energy SoC Modules	5	$ 15.00	$ 75.00	7	$ 9.95	$69.65
5V Power Supply	1	$ 5.00	$ 5.00	0		$0.00
MicroUSB Cable	1	$ 2.00	$ 2.00	0		$0.00
LED	10	$ 0.50	$ 5.00	0		$0.00
Battery Connector	5	$ 0.50	$ 2.50	7	$ 0.28	$1.96
Fabricated PCBs	5	$ 10.00	$ 50.00	1	$ 45.65	$45.65
Batteries	5	$ 3.00	$ 15.00	1	$ 6.38	$6.38
Passive Components	1	$ 20.00	$ 20.00	1	$ 90.17	$88.21
		Material Total	**$ 309.50**		**Material Total**	**$ 411.74**

Schedule

- Underestimated
- Much interleaving of tasks

Labor Cost (Hrs)		
Task / Activity	**Estimate**	
Research Components and Software Libraries	5	
Create Schematic Capture	20	
Layout PCBs	20	
Develop remote sensor firmware	15	
Develop thermostat bridge software	20	
Develop web frontend	30	
Procure project materials	5	
Assemble remote sensors	5	
Test functionality of hardware	5	
Test functionality of software	5	
Test functionality of integrated system	5	
Write final report	30	
Create presentation	20	**Actual**
Labor Hr Totals	**185**	**108**

Lessons Learned

29

- Programming/debug hardware layout first.
  - Saves time breadboarding
  - Fewer failure points
- If using a new framework (Eddystone), expect changes
  - Library unknowingly changed during development
  - Unnecessary debug time

Lessons Learned

30

- New chip manufacturer?
  - Expect a learning curve
  - Nordic Semiconductor
    - Much different than Microchip and TI.
    - From SDKs to documentation style

Conclusion

- Project was successful!
- Learned a lot
  - Bluetooth
  - Node.js
  - Nordic nRF SDK
  - Time Management

Demostration

33

Questions?