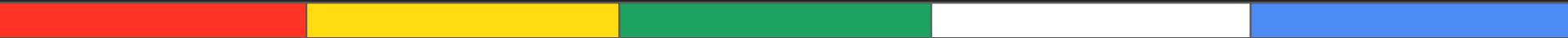


Final Presentation

Firmware

Scott Nakashima

Ryan Walser



Overview of Project

- ▶ Design and write the embedded programming used on the environmental sensor modules (weatherboxes)
- ▶ Poll and collect data from sensors
- ▶ Send data to the server
- ▶ Replace and improve current firmware for the “Apple” generation, with additional functionality

Motivation & Goals

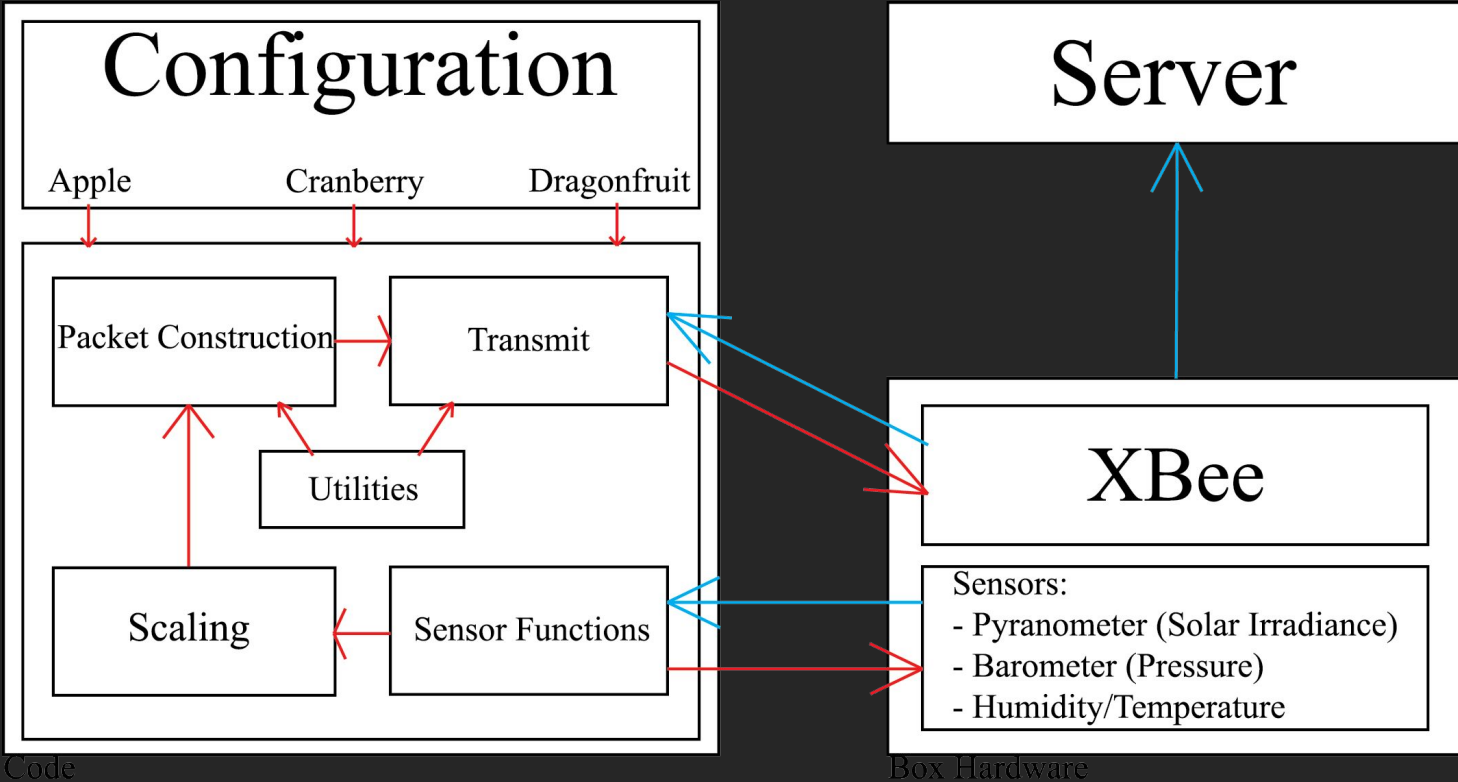
Motivation

- ▶ Gain experience in writing firmware for embedded systems
- ▶ Practice software/firmware design methods
- ▶ Use of relevant tools
- ▶ Writing platform agnostic software and unit tests

Goals

- ▶ Working code (verifiable on Apple)
- ▶ Modular Implementation
- ▶ Well-documented
- ▶ Easily configurable for any generation
- ▶ Unit tests

Overall Design (Block Diagram)



Code

Box Hardware

Algorithm

Initialization

Version Check

Initialize corresponding
functions & variables

Initial health diagnostics

Execution (Continuous Loop)

Check weatherbox battery health

if good battery voltage:

Poll sensors

Data manipulation

Construct packet

Transmit packet

Clear packet buffer

if poor battery voltage:

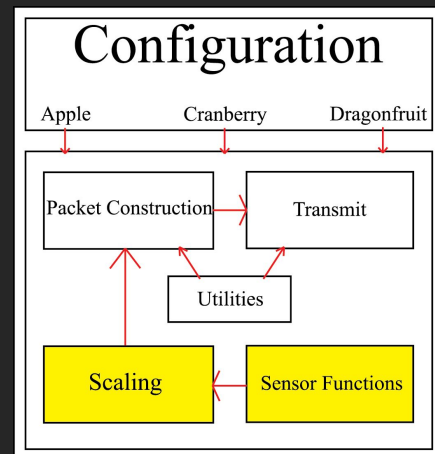
Power save routine

Modules



Sensors

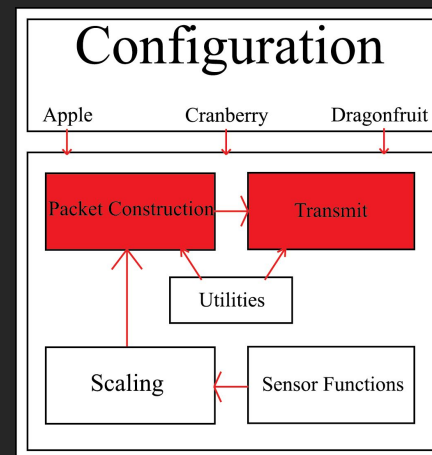
- ▶ Depending on configuration/generation:
 - ▷ Different sets of functions
 - ▷ Different pin configurations
 - ▷ Managed by Configuration Module
- ▶ Data is scaled using predetermined values from original code
- ▶ Uses functions from external libraries



Code

Transmit

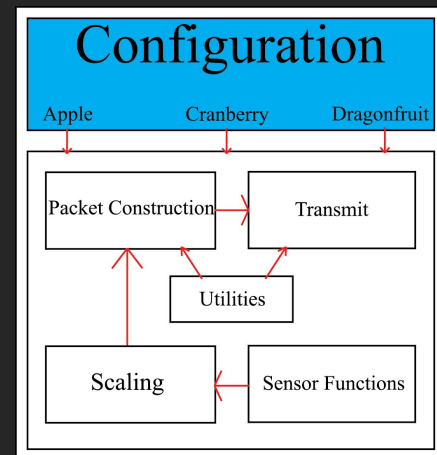
- ▶ Two methods of transmitting
 - ▷ UART
 - ▷ Binary
- ▶ Method managed by Configuration Module
- ▶ Uses external XBee library
- ▶ Components:
 - ▷ Initialization/Clear
 - ▷ Construction
 - ▷ Transmit
 - ▷ Test Packet Construction



Code

Configuration

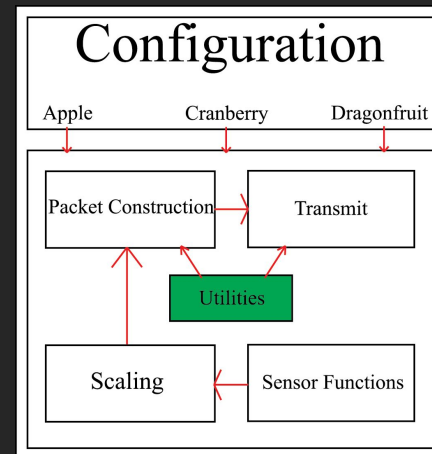
- ▶ Generation declaration
- ▶ Pin configuration
- ▶ Function pointer implementation
 - ▷ Sensors
 - ▷ Transmit
 - ▷ Routines



Code

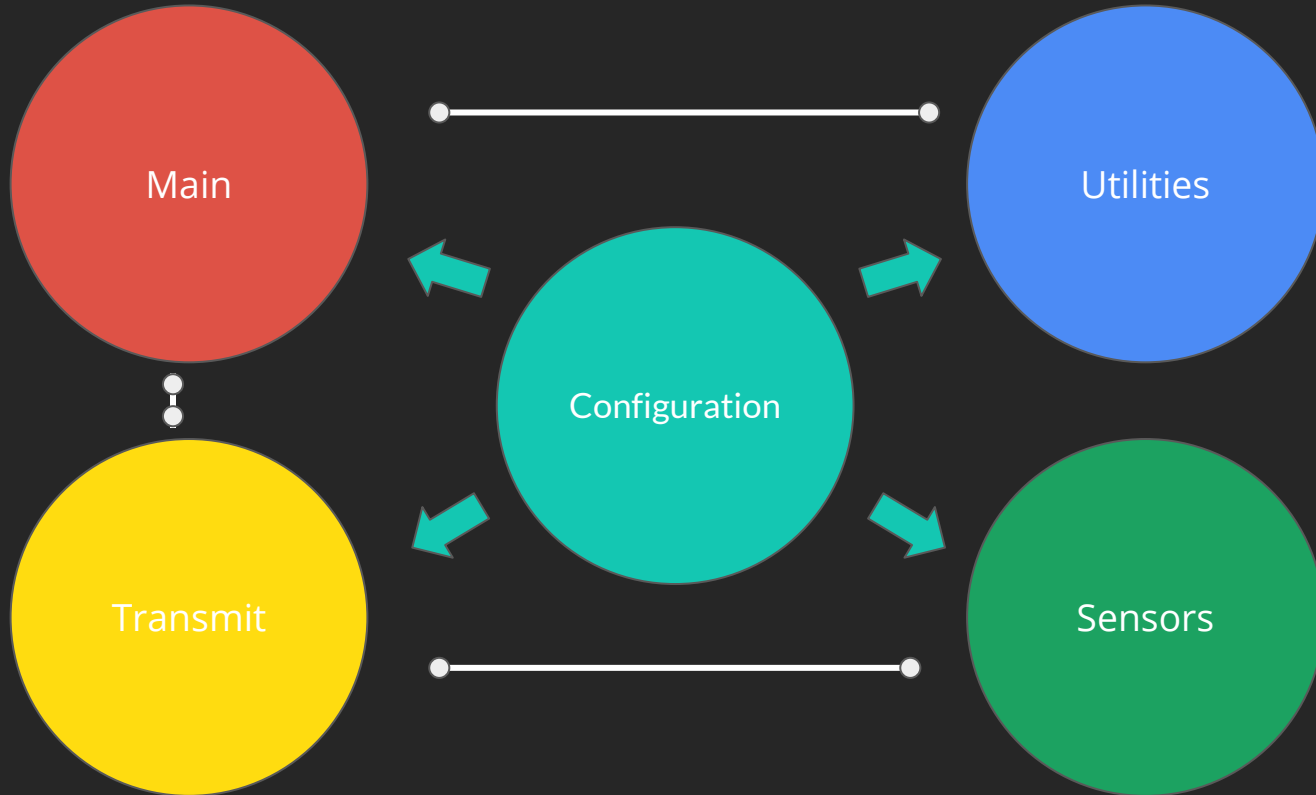
Utilities

- ▶ Macro definitions
- ▶ Health functions
 - ▷ Health Sample Initialization
 - ▷ Health Check
 - ▷ Health Data Transmit
- ▶ Power Management functions
 - ▷ Switches sleep state of XBee and sensors



Code

Module Integration



Evolution from Previous Code

Addressed Problems:

- Global variable use
- Difficult to trace
- Portions cannot be tested independent of hardware
- Only configurable for Apple

Improvements:

- Variable control
- Readability
- Unit testing
- Easy configurations

Major Problems & Solutions

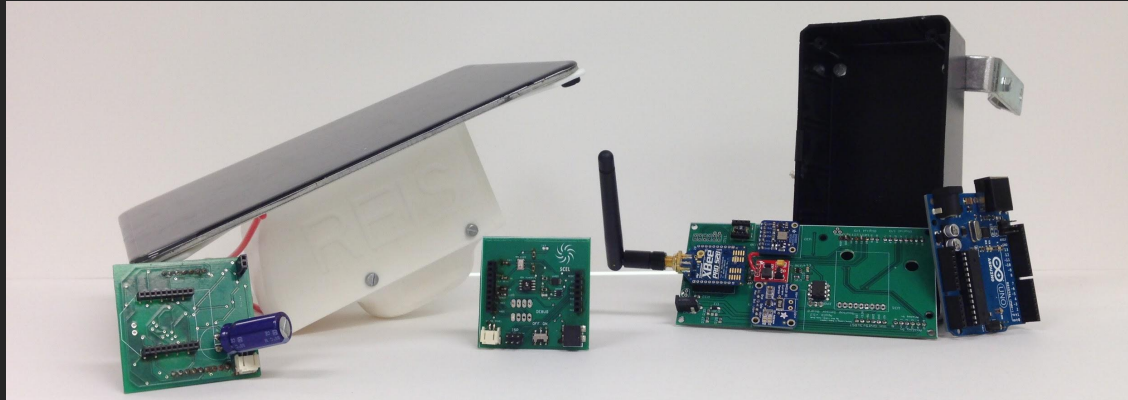
- ▶ Time management
 - ▷ During integration stage, our work overlapped
- ▶ Variable Management
 - ▷ Memory leak
 - ▷ Global decisions
 - ▷ Arduino IDE features
- ▶ XBee Transmission
 - ▷ XBee's configuration
 - ▷ Confirming packet transfer

Final Status

- ▶ Individual Modules tested and verified
 - ▷ Sensors, Transmit, Configuration, Utilities
- ▶ Two main routines implemented
 - ▷ Normal
 - ▷ PowerSave
- ▶ Easy implementation for new generations
 - ▷ Write sensor functions
 - ▷ Add pin configurations

Future Work (Next Semester)

- ▶ Add pin configurations
 - ▷ Cranberry
 - ▷ Dragonfruit
- ▶ Verify code on remaining generations
- ▶ Complete unit testing for each module
- ▶ Error code library



Remaining/Anticipated Problems

- ▶ Writing appropriate/meaningful unit tests
 - ▷ Proper *Stubs* and *Mocks*

Questions?

