Iowa State University

Client - Texas Instruments
TI Representative - Mark Easley
ISU Advisor - Dr. Nathan Neihart

Sterling Hulling - E E
Parker Larsen - Cpr E
Taylor Weil - Cpr E
Sean Griffen - Cpr E

BeagleBone Green Gateway IoT Hub

beagleboard.org

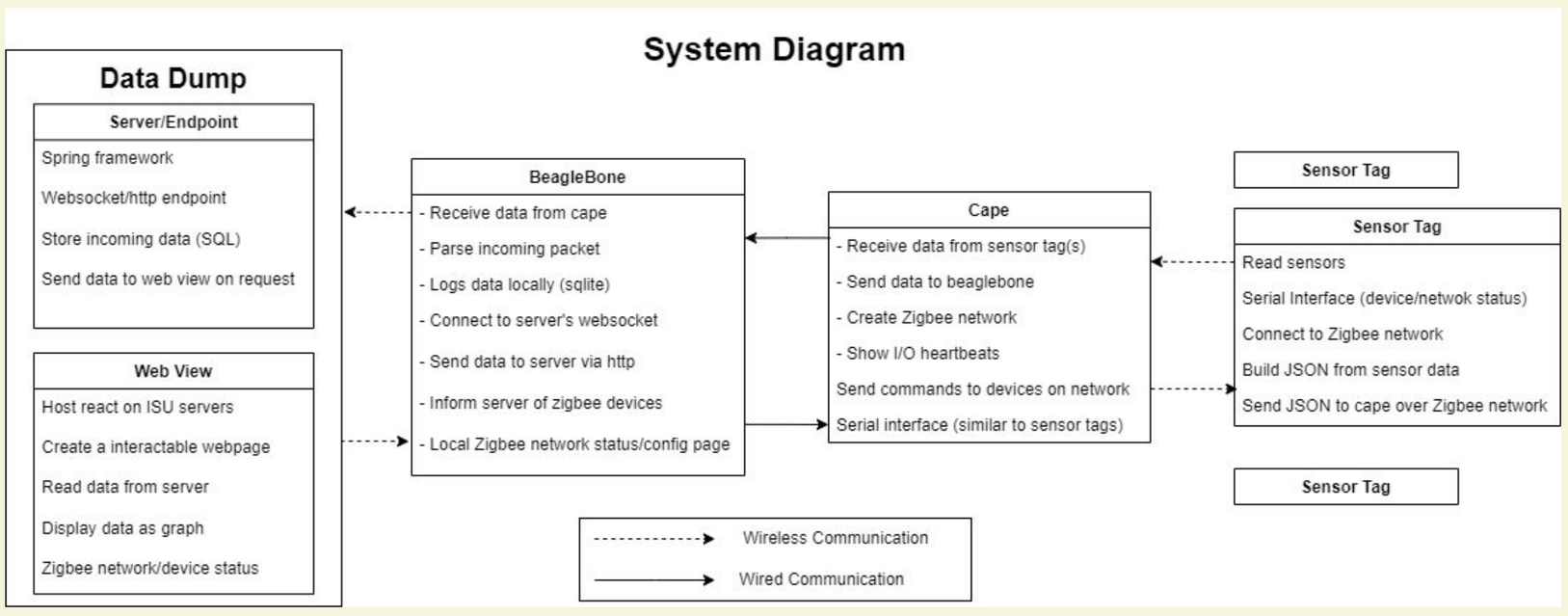
Group sddec21-07

Problem Statement

The BeagleBone Green Gateway is an IoT-centric microcontroller with built-in Wifi, bluetooth, a Linux Kernel, and GPIO headers. Our mission was to extend its capabilities further by designing and implementing a cape expansion pack that communicates with various devices over a Zigbee network. Our goal was to provide an open-source foundation that could be easily extended by future developers and enthusiasts.

Approach

To demonstrate our IoT hub, referred to as the Cape, we themed the project around aggregating data from sensors around a car. Sensor Tag boards with the same chip as the Cape are used to periodically report data across the zigbee network.



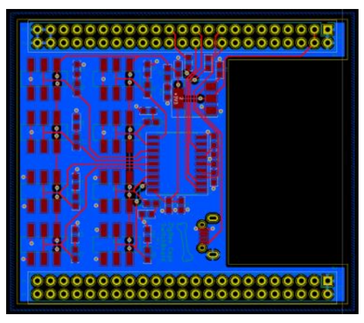
Design

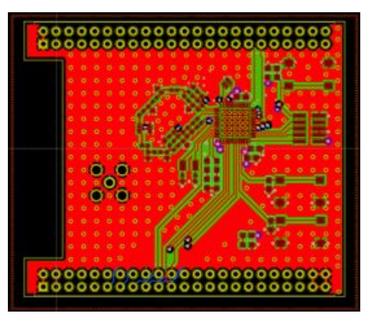
Our project was twofold:

1) Create and build a PCB Cape based on TI's CC1352P microchip that could interface with the BeagleBone.
2) Create software for the Cape to make it into a Zigbee controller, and relay information from the network end-devices to the Beaglebone.

Hardware

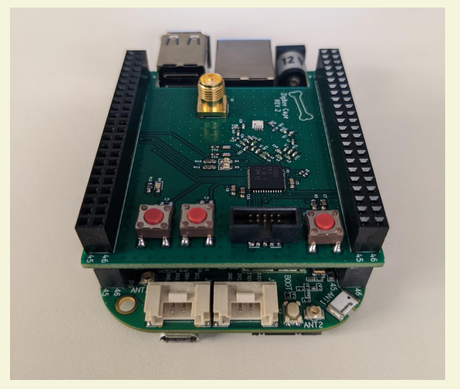
The hardware was split into 2 different boards. The first board added the ability for the BeagleBone to connected to a ZigBee network through the use of a cape. The cape communicated with the BeagleBone over UART via the pin headers. The second board talked with the BeagleBone over ZigBee through an additional cape. This board utilized SPI to talk to an IO expander to receive input from a variety of switches.

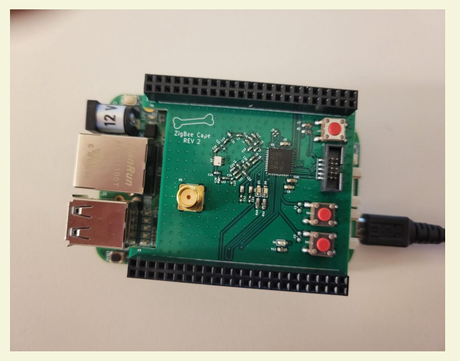


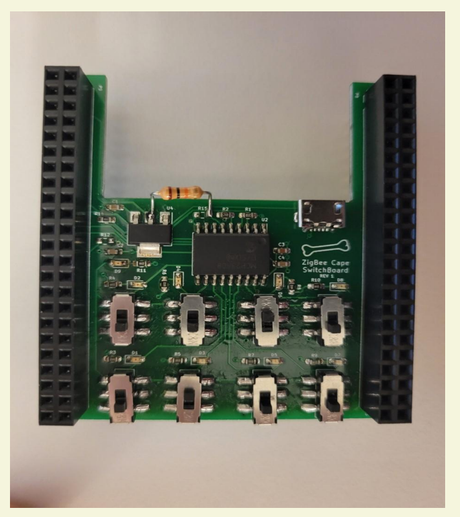


Software

The software needed to be divided into a number of projects across multiple platforms. The BeagleBone has been programmed in Cloud9 IDE using JavaScript. The Cape and Sensor Tags are programmed using TI-RTOS with Code Composer Studio, and built on top of a number of open source demo projects for the CC1352 MCU. The BeagleBone will report incoming data to a Web Server built with Java Spring Boot and will be visible on a Web App created with React. Each component is meant to operate independently and seamlessly with the only user interactions needed being initialization.







Implementation and Testing: In order to test our design, we used the demo boards provided by Texas Instruments since they closely resemble our final design. Once our first revision of the cape was complete, we replaced the demo board with it to verify that everything was correctly implemented. After we got each component working with our custom board, we setup our project for our demo. Because our project aimed at developing a modular platform and wasn't developed for a singular use case, our demo showed the collection of temperature and light data from end-devices on the Cape's Zigbee network. In reality, the Cape could receive data from any Zigbee device and relay that to any user-specified endpoint.

Challenges We Faced: Due to the number of technologies we used, there was a significant learning curve to get started. Our team had to learn the ins and out of concepts including Zigbee, UART, PCB design, and a number of programming languages. We also faced challenges when the devices did not work as anticipated. It took a lot of collaboration between Software and Hardware to accomplish our goals.