EE/CprE/SE 491 WEEKLY REPORT 6

March 6 - March 12

Group number: 9

Project title: Arinc429 Portable Receiver APP and Firmware

Client &/Advisor: Colin Cox & Daji Qiao, Mathew Wymore

Team Members: Eduardo Contreras, Riley Millam, Nicholas Morgan, Jared Staskal, Nate Trotter

- **Weekly Summary** *This week we continued to research Bluetooth Low Energy communications and got ready to begin planning our Bluetooth Low Energy api for connecting the Flutter app to the receiver. The firmware team also continued to make progress becoming comfortable and familiar with the hardware and software we will be using to run and write our firmware.*

- **Past week accomplishments**
    - Getting familiar with the Hardware from the client - All
        - Now that we've received the hardware we have started to test things with it to get more comfortable thinking about developing with it.
        - Have attempted some ESP32 IDF firmware programming starter examples to familiarize ourselves with different ESP32-S3 functionality and hardware pin layout.
    - Compiled notes on BLE technology for the team - Eduardo Contreras
        - Researched Bluetooth Low Energy technology and standards
        - Compiled the information neatly for the team to use

- **Individual contributions**

NAME	**Individual Contributions** *(Quick list of contributions. This should be short.)*	**Hours this week**	**HOURS cumulative**
Eduardo Contreras	● Experimented with ESP32 DevKit and tried programming some starter examples in ESP-IDF ● Researched and started BLE API documentation	5	36
Riley Millam	● Has been in charge of communication with the client and advisors. ● ESP32 programming	4	37
Nicholas Morgan	● Went over BLE modules and protocols ● Planning to start a datasheet for BLE	4	40
Jared Staskal	● Researched Bluetooth Low Energy technology	7	41
Nate Trotter	● Familiarized, researched, experimented, with new hardware ● Researched BLE more in depth	6	42

- **Plans for the upcoming week**

Now that we've spent time researching communicating over BLE and how to plan that out we need to start planning that by creating our BLE API documentation.
The firmware people will continue to work with the hardware to make sure they are ready to implement the firmware when the time comes.

Source: Embedded Centric

BLE (Bluetooth Low Energy)

Communication Methods

1) Point-to-point (1:1)
    - 20 connections possible
    - Max App Throughput: 1.4 Mbps (1 connection)
2) Data Broadcast (1:Many)
3) Mesh (Many:Many)

Generic Access Profile (Gap)

- Provides full standard framework for controlling a BLE device
- Defines how BLE devices can discover and connect with one another
- Responsible for setting the BLE device address (48 bits)

GATT

- Responsible for the data exchange of BLE
- Two device roles: GATT Server, GATT Client
    - GATT Server:
        - Usually Implemented on the embedded device side.
        - Implements the attributs table structured in the form of Services and Characteristics
        - Holds the useful data to be accessed by a remote client
        - Server contains one or more GATT services

Service

- Encapsulates zero or more functionality-related user data containers called characteristics
- Example)
  - Service: Heart Rate
  - Characteristics: Heart Rate Measurement, Body Sensor Location

Characteristics

- Encapsulates at least two attributes:
  - Characteristics declaration attribute
    - Contains metadata about the characteristic
  - Value attribute
    - Contains the user data in its value field

GATT Client

- Usually implement this on smartphone, tablet, or a computer
- Interested in accessing the data
- GATT Operations:
  - Service(s) discovery:
    - Obtain the structure of all available services and characteristics on the server side
  - Read Characterictic/descriptor:
    - Allows a client to read the current value of a characteristic value or a descriptor

- Writing Characteristic/descriptor:
    - Allows a client to write to a characteristic value
- Exchange ATT MTU
    - ATT MTU: attribute protocol maximum transmission unit