Arinc429 Portable Receiver APP and Firmware

Design Document

Senior Design Project Group 9
Collins Aerospace - Colin Cox
Daji Qiao and Mathew Wymore
Eduardo Contreras, Riley Milliam, Jared Staskal, Nick Morgan, Nate Trotter
sddec23-09@iastate.edu
https://sddec23-09.sd.ece.iastate.edu

Executive Summary

Development Standards & Practices Used

Hardware

- Arinc429
  - Arinc Evaluation Board
  - HI-3593
- ESP32
  - ESP32-S3 Devkit
  - ESP32-S3-WROOM-1

Software

- Visual Code Studio
  - ESP-IDF
    - Micropython
    - C
  - Flutter Framework
    - Dart

Engineering Standards

- IEEE SA - IEEE Publishes Standard Addressing Real-Time Architecture for Embedded Systems - Embedded System
- IEEE SA - IEEE 1118.1-1990 - Serial Bus
- IEEE SA - IEEE 802.15.1-2002 - Bluetooth
- ARINC 429 - Arinc 429

Summary of Requirements

Flutter Application

- Efficiently communicate with the EPS32 over BLE
  - Send and receive data to and from the ESP32
- Interpret, decode, Arinc labels
- Create Arinc labels through a GUI
- Display necessary information to users through a GUI

ESP32

- Effectively send and receive data to and from Arinc429
- Efficiently communicate with the Flutter application over BLE
    - Send and receive data to and from the Flutter application
- Validate information received from Flutter application

Applicable Courses from ISU Curriculum

Some of the courses we've taken here at Iowa State that have prepared us for this project are: CPRE/S E 185, CPRE 186, CPRE 288, CPRE 388, CPRE 488, COM S 309, COM S 319, S E 329, and S E 339. These courses have prepared us for this project by equipping us with the necessary tools to form our own plan of action and with the skills to execute it properly.

New Skills/Knowledge Acquired

This project has allowed us to explore new skills and build upon the experience we've acquired from Iowa State University with a new layer of skills. From this project we've gained knowledge in mobile app development using the Flutter framework, BLE GATT API development, and firmware development within the ESP environment.

Table of Contents

1 Team

1.1 Team Members:

Jared Staskal, Nick Morgan, Nate Trotter, Eduardo Contreras, Riley Millam

1.2 Required Skill Sets for your Project:

App Development
OOP Experience
An Understanding of the Necessary Technologies

- BLE Communication
- Arinc429 System
- ESP32S2 System

Embedded Systems Development
C Programming
Development Using ESP-IDF

1.3 Skill Sets Covered by the Team:

Our team has experience with mobile application development, embedded systems development, as well as experience with C and Java programming. More specifically: Jared Staskal, Nick Morgan, Eduardo Contreras, and Riley Millam have experience with mobile application development; Nick Morgan, Nate Trotter, Eduardo Contreras, and Riley Millam have experience with embedded systems development. Additionally, all members of the team have experience with Java and C programming.

1.4 Project Management Style Adopted by the Team:

We plan to apply an Agile Development Style. We made this decision to accommodate the two different parts of the project, possibly moving at different paces. An agile style will allow the two parts to move separately without conflict while applying a waterfall style may cause dependency issues between the two pieces.

1.5 Initial Project Management Roles:

Communications Manager: Riley Millam
App Dev Team:
    Jared Staskal
    Nick Morgan
Firmware Dev Team:
    Eduardo Contreras

Riley Millam
Nate Trotter

2 Introduction

2.1 Problem Statement

Our problem we are trying to solve is the need for a bulky and expensive data receiver and transmitter. We will be altering a size and cost-effective bus reader to meet a variety of requirements so it can be used in the avionics industry and replace the bulky and expensive one.

2.2 Requirements & Constraints

Firmware

Functional Requirements	
Number	Requirement
1	ESP32S3 to read in data from Holt 429 receiver
2	ESP32S3 to decode labels using the Flutter App
3	ESP32S3 to send data over Bluetooth Low Energy
4	ESP32S3 to connect to Smartphone
5	ESP32S3 to receive labels over Bluetooth Low Energy
6	ESP32S3 rejects incorrectly formatted words (incorrect parity, etc.)
7	Send multiple labels over BLE

Non-Functional Requirements	
Number	Requirement
1	Ability to quickly and easily update firmware
2	Easy to understand and use
3	Send labels in less than a second

Non-Functional Requirements	
4	Receive labels in less than a second
5	Decode labels in less than a second

Flutter Application

Functional Requirements	
Number	Requirement
1	Communicate with the chip over Bluetooth Low energy
2	Read Arinc labels from the BLE
3	Decode the Arinc labels
4	Send Arinc labels over BLE
5	Receive multiple incoming labels at a time
6	Handle errors with the data(incorrect parity, etc.)
7	Display rate of information received
8	Define and store new labels
9	Display SDI(Source Direction Indicator)
10	Display SSM(Sign/Status Matrix)
11	Listen for multiple labels being sent over BLE
12	Display the data bits according to their meaning(defined by the labels)

Non-Functional Requirements	
Number	Requirement
1	Reliably connect to the chip (numbers tbd)
2	Easy to understand and use
3	Send labels quickly (numbers tbd)
4	Receive labels quickly (numbers tbd)
5	Decode labels quickly (numbers tbd)
6	Needs to be available for android and iOS
7	Needs to be maintained to be compatible with future OS updates
8	Be palatable to look at

Constraints and other requirements

1	The project must be completed by the end of the Fall 2023 semester
2	The project must use the ESP23S3 Microprocessor
3	The mobile app needs to use the Flutter framework
4	The microcontroller should communicate with the mobile app over Bluetooth Low Energy

2.3 Engineering Standards

IEEE SA - IEEE Publishes Standard Addressing Real-Time Architecture for Embedded Systems - Our project focuses heavily on embedded systems and it is important for us to conform to the standards of embedded systems.

IEEE SA - IEEE 1118.1-1990 - One of our primary means of communication would be through serial buses; it is in our best interest to conform to bus standards to properly communicate this information.

IEEE SA - IEEE 802.15.1-2002 - Our other means of communication is through bluetooth; it is in our best interest to conform to bluetooth standards to properly communicate information between a server and client.

ARINC 429 - This is the word format that we will be sending and receiving.

2.4 Intended Users and Uses

The people who benefit from this project is Collins Aerospace, it allows them a way of transmitting data which is both size and cost effective. It also allows for Bluetooth low energy communication, which allows the end user to receive the data conveniently on their smartphone. Those using this will be a variety of people in the aerospace industry who want to view data which is being transmitted from labels.

3 Project Plan

3.1 Project Management/Tracking Procedures

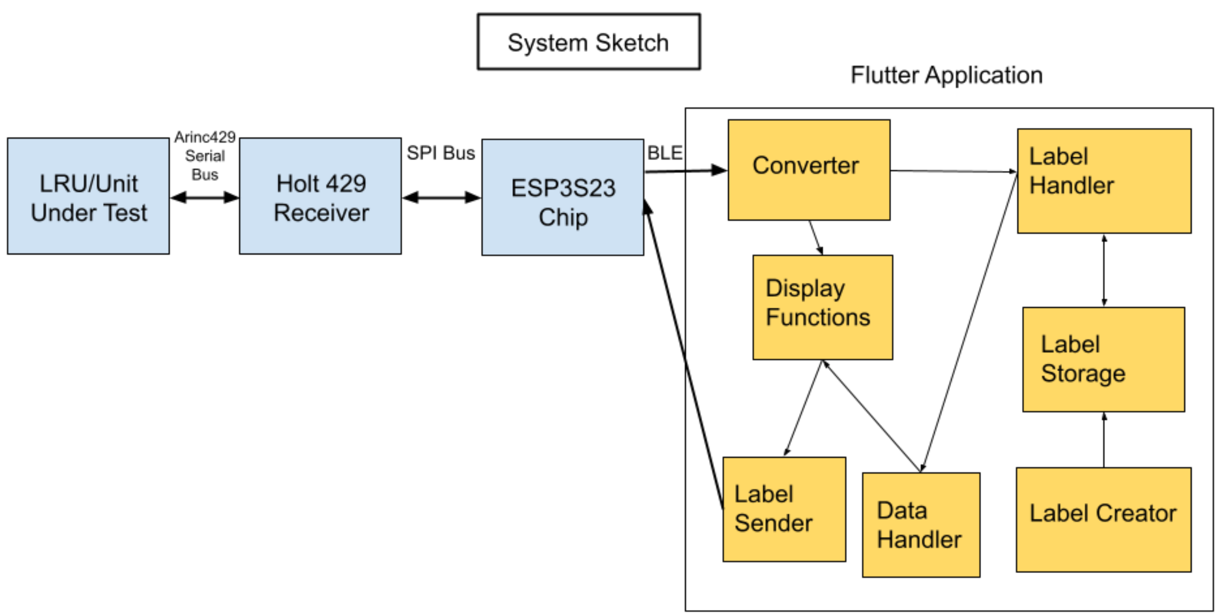
We will be using an agile approach as it allows us to better work through the different components. We've already split up our system into the various necessary components based off of our requirements and agile will work well with the development of individual components as we will be able to ensure that they all work properly as units and then move outwards to ensure that they work as a whole.

We will be using Github to coordinate our work alongside Trello and Discord to delegate and organize our work.

3.2 Task Decomposition

There are two parts to the project, the Firmware for the ESP32-S3 and the Flutter Application. For the Firmware there are two important tasks: reading in the information from the Holt receiver, and broadcasting the words over Bluetooth Low Energy.

For the Flutter Application we've gone through and decided that we will be dividing it up into the sections illustrated in the following figure



1. Where the converter accepts the 32 bit word in whatever form it is in, and packages it into a more usable object containing an array of the bits in the word, and various metadata about the transmission of the data.
    a. Read the word from BLE

    b. Extract the necessary information
    c. Convert it into the correct form
2. The Label Handler accepts the object from the Converter and reads out the label bits from the word and searches for them in the Label Storage. Once it has found the label corresponding with the data it was given it reads how the label defines the data and sends that to the data handler.
    a. Binary search for the label
    b. package the label information into the word object
3. The Data Handler accepts the data object and the label information from the Label Handler and converts the different bits in the data section into the different formats as specified by the label.
    a. Functions for converting bits into different data forms
    b. Convert the bits into the correct format according to the label
4. The Display Functions accepts the data directly from the Converter in order to display the metadata involved. It also accepts the processed data from the Data Handler in order to display that to the user.
    a. Display the metadata and other non-”data” information(SSM, parity bit, LRU)
    b. Display the processed data
5. The Label Storage component will just be storage for user created labels that will be called upon by the Label Handler as described. It will be updated by the Label Creator when a user creates a label.
    a. Define storage forma
6. The Label Creator component will allow users to input custom labels to define how the data is read and displayed as well as the meaning of some of the other fields in the word.
    a. Create functions to write to the label storage
    b. Create screens to handle the user input

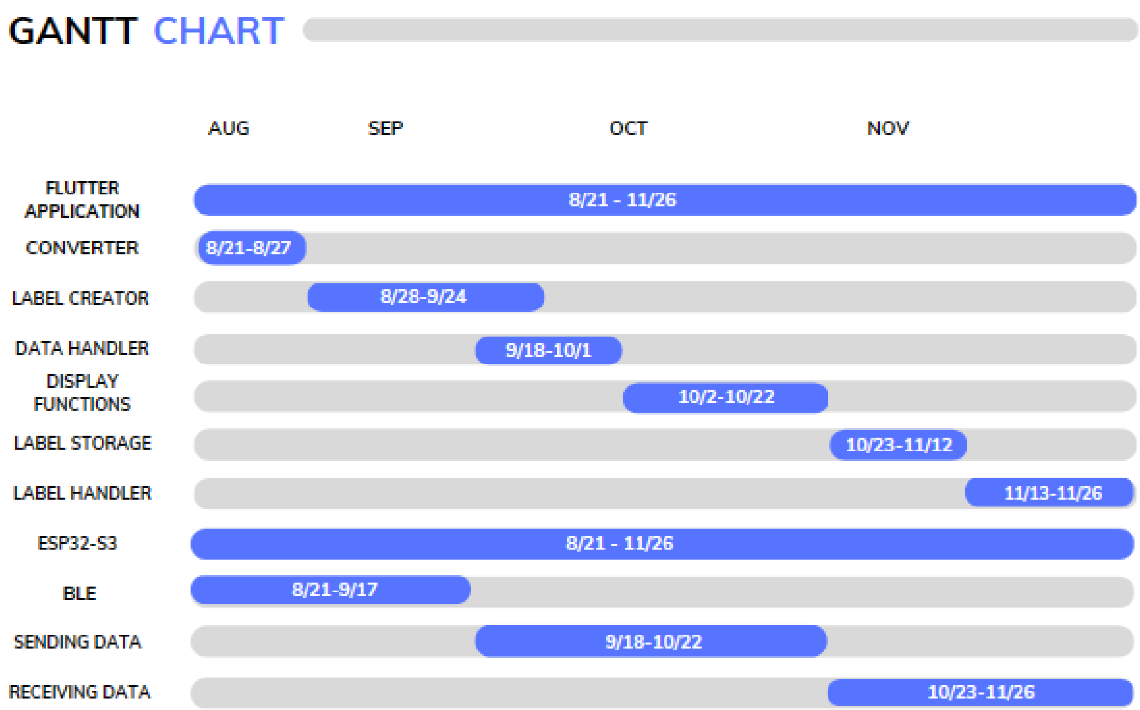
<u>Hardware</u>

7. We will need to implement Bluetooth Low Energy (BLE) on the ESP32S3 so that it can send and receive data to our flutter application. This will be worked on during the first sprint, so it should be done in 2 weeks.
8. Our Esp32s3 will have to send ARINC429 data. We will have to write the firmware for that, and this will be done over 2 sprints. (1 month)
9. Our Esp32s3 will have to receive ARINC429 data. We will have to write the firmware for that, and this will be done over 2 sprints. (1 month)

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- **Sprint 1** - BLE on Flutter App, BLE on ESP32S3, Convertor
- **Sprint 2 -** Sending Data on ESP32S3, Label Creator, Display Functions
- **Sprint 3 -** Sending Data on ESP32S3, Label Creator
- **Sprint 4 -** Receiving Data on ESP32S3, Data Handler
- **Sprint 5 -** Receiving Data on ESP32S3, Storage Handler
- **Sprint 6 -** Finish Debugging on ESP32S3, Label Handler
- **Sprint 7 -** Debugging entire project, wrap everything up

3.4 Project Timeline/Schedule



3.5 Risks And Risk Management/Mitigation

Task	Sub Task	Risk Factor	Probability(0-1)	Cost(1-5)
Firmware	Read Data	Loss of data during transmission	.01	4
	Transmit Data	Loss of data during transmission	.01	4
Flutter Application	Converter	Reading data over BLE fails	0.1	5
	Label Handler	Binary Search Takes Longer than Expected	0.2	2
	Label Storage	Creating files works differently on different operating systems	0.2	2
	Label Creator	User created custom labels are inaccurate to the Arinc429 word format	.2	3
	Data Handler	Casting errors when handling different data types	.1	1
	Display Functions	On-screen display is significantly out of place on different mobile platforms	.1	1

3.6 Personnel Effort Requirements

Flutter Application		
Task	Effort(hours)	Explanation
1.a	5	This is a generous estimate, as this task could come with a lot of issues and may take a while to troubleshoot
1.b	2	Writing a method to parse through the bits to find the LRU, SSM, SDI and others
1.c	2	Constructing the necessary object from the information shouldn't be too much hassle, and the work of deciding what the object looks like should be finished before work on this starts
2.a	4	This should be a pretty standard binary search, however there is a chance for duplicate labels in which case we will need to account for user input and interfacing with the storage system
2.b	4	A decision on how to store the data definitions needs to be reached. Once that is decided generating the necessary objects shouldn't be too difficult
3.a	2	Reading in bits and converting them to different data types is a pretty standard operation and should be easily implemented
3.b	3	Efficiently storing different data types together could cause some issues, however none of the data types should be too complex and the variety of different data types should also be relatively compact
4.a	3	Standard frontend things, just showing some numbers and text
4.b	3	Very similar to 4.a
5.a	3	Deciding on how to store the labels and how to deal with multiple labels of the same number should be done before implementation so this will just be translating those into code
6.a	3	Writing to a file could be tricky but other than that this is just inserting an item into a sorted list
6.b	4	The bulk of this will be handling user input which is always a bit time consuming
Firmware		
Task	Effort(hours)	Explanation
7	6	Implementing BLE on firmware might take a while to troubleshoot and test. Rough Estimate
8	15	We will have to develop firmware to send ARINC429 Data. This is rough estimate and will probably take longer

9	15	We will have to develop firmware to receive ARINC429 Data. This is rough estimate and will probably take longer

3.7 Other Resource Requirements

Collective list of resources required to actualize our project:

- Holt Arinc Evaluation Board
    - Arinc429
- Arinc429 list of Standard Labels
- Holt Arinc Evaluation Board Datasheet
- ESP32-S3 Development Board
    - ESP32-S3-WROOM-1
- ESP32-S3 Datasheet
- ESP-IDF Documentation
- Breadboard, Jumper Wires

4 Design

4.1 Design Context

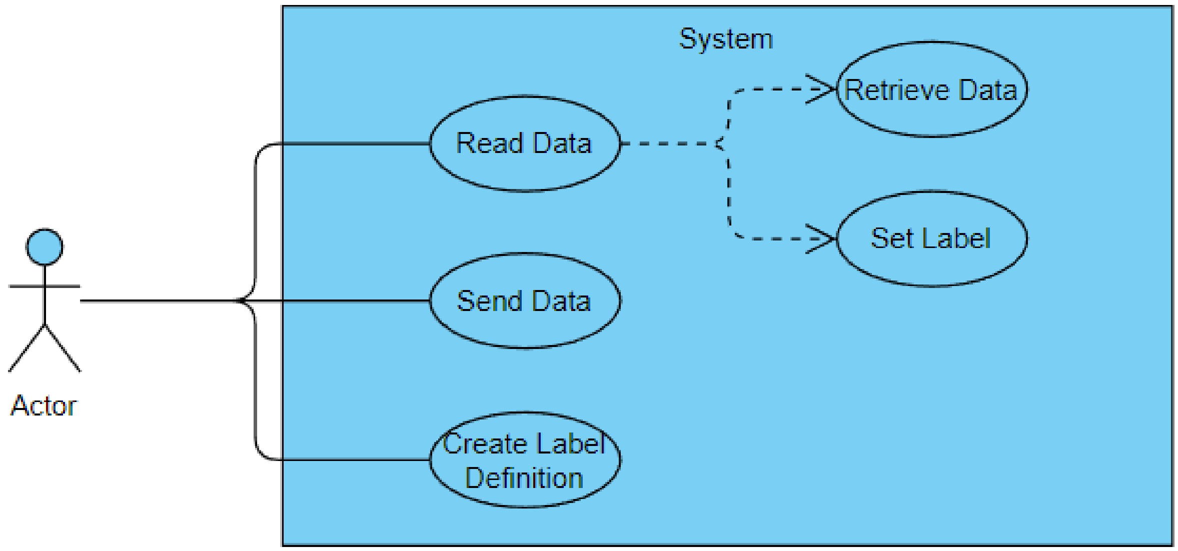
4.1.1 Broader Context

Our project exists within a very specific and proprietary space. Thus we are designing specifically with those working in the avionics field. This means that our users will most likely be technologically competent and knowledgeable about the data being conveyed. Due to the specificity it only really addresses the need for efficiency and ease of use when people are doing their jobs.

Area	**Description**	**Examples**
Public health, safety, and welfare	How does your project affect the general well-being of various stakeholder groups? These groups may be direct users or may be indirectly affected (e.g., solution is implemented in their communities)	This method of interacting with the Arinc429 data would allow for a more productive work day by eliminating the time consuming task of connecting to the Arinc429 directly.
Global, cultural, and social	How well does your project reflect the values, practices, and aims of the cultural groups it affects? Groups may include but are not limited to specific communities, nations, professions, workplaces, and ethnic cultures.	Our project directly supports those in the avionics industry and their needs, values, and practices by giving them a long-needed solution to a problem they consistently face
Environmental	What environmental impact might your project have? This can include indirect effects, such as deforestation or unsustainable practices related to materials manufacture or procurement.	Using smaller pieces of equipment lowers the environmental impact of manufacturing and waste from disposal
Economic	What economic impact might your project have? This can include the financial viability of your product within your team or company, cost to consumers, or broader economic effects on communities, markets, nations, and other groups.	Product should be low cost with the use of the ESP32-S3 chip Additionally by increasing the ease with which engineers can use Arinc 429 technology will increase production overall

4.1.2 User Needs

Aerospace and systems engineers need a way to easily read and write Arinc 429 data because it is currently very inconvenient.



4.1.3 Prior Work/Solutions

There exist other products to read and write data using Arinc 429. However, the existing products are bulky and expensive. The pros of our target solution are that it should be smaller and easier to use than existing solutions. This comes at the cost of requiring a cellphone and lower reliability as a result of communicating over bluetooth low energy.

The current method of communicating with the Arinc 429 requires directly connecting to the device itself in order to read the information being transmitted to/from the device. However, with this solution we're able to pass this information to the ESP32-S3 and transmit it through BLE to our Flutter application, effectively allowing for wireless querying of the Arinc429 data from your smartphone.

4.1.4 Technical Complexity

The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles and the problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

4.2 Design Exploration

4.2.1 Design Decisions

Key design decisions:

1. We decided to use the ESP32-S3 chip as it is a powerful and relatively inexpensive chip

2. We decided to use the Flutter framework to develop our mobile application due to its cross platform compatibility and ease of use
3. BB830 Circuit Board

4.2.2 Ideation

The client had already completed thorough research on all of our design decisions, and made recommendations based off of his knowledge in the industry. Also, because this was a niche issue, there were few options to explore anyway. The circuit board was decided to be used for testing because we already had it from class and there was no need to look for other options.

4.2.3 Decision-Making and Trade-Off

Our ideated options were chosen specifically by the client. His decision making for the ESP32 chip was because it was small and inexpensive, which was something he had not been able to use prior to this. It also had a lot of documentation regarding troubleshooting and getting it to work properly with bluetooth low energy, so it was the perfect option.

The final decision of the flutter application was arrived at because it was cross-platform and allowed for us to use an inexpensive chip. It was also a way of communicating which allowed us to transmit data to smartphones.

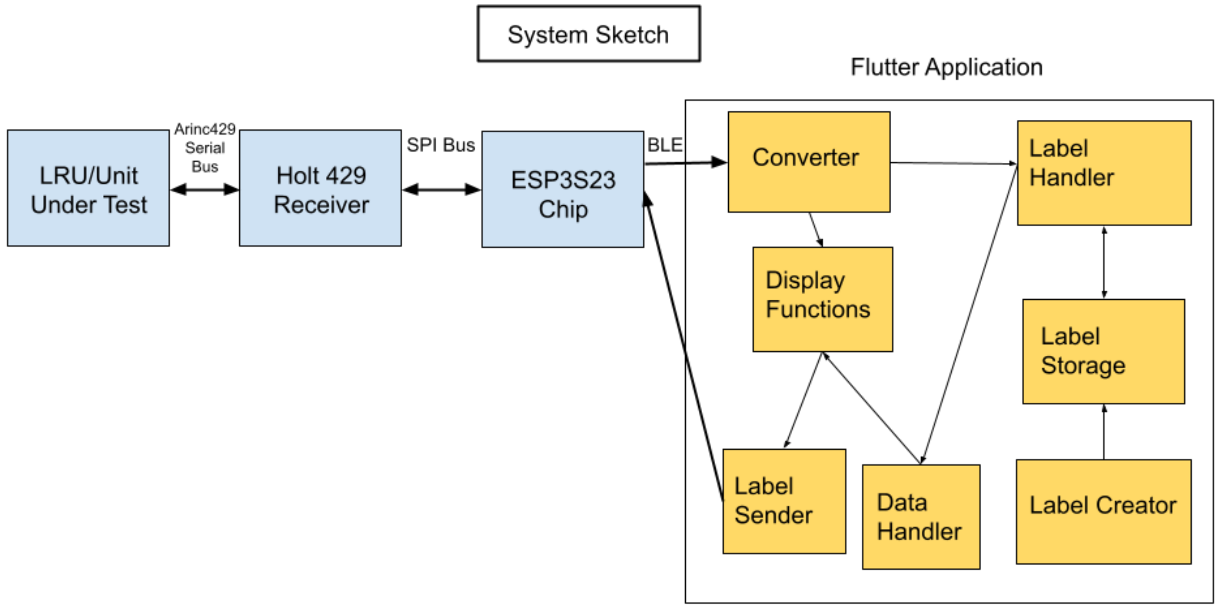
The circuit board was decided on because we all already had it, it worked effectively, and we didn't need anything that did something more for just testing.

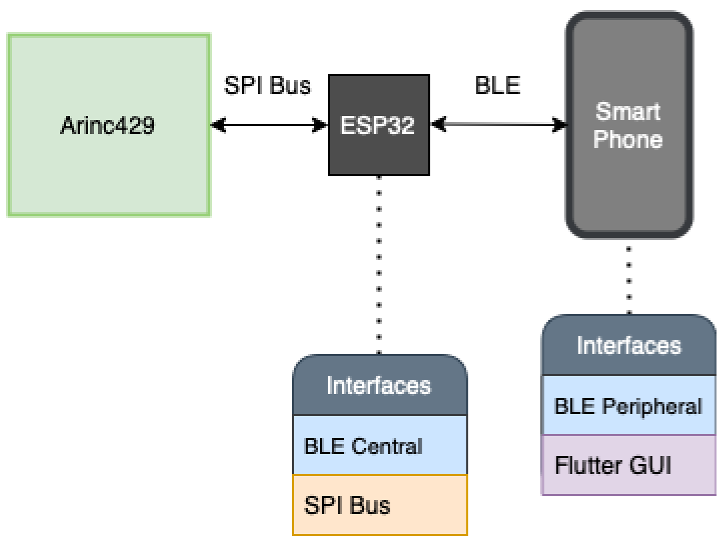
4.3 Proposed Design

Since a lot of the concepts and hardware needed to implement our project are new to our team, we have been doing a lot of research to familiarize and prepare ourselves for the software development part of our project. We have looked over datasheets for the ESP32-S3 microcontroller. We haven't done any implementation yet, however, we have tested the ESP32-S3 microcontroller in order to try to use BLE, as well as other applications. We have also been experimenting with and setting up the new environments needed to implement the firmware and application of our project. Our firmware team has been trying out some given ESP-IDF examples to work with some of the functions that will be useful to our project.

We have completed a system sketch of our overall project that also includes the main functions needed to implement the flutter application. The system sketch shows how each major component in our project will be communicating with each other, so that we can get a good understanding of what we need to implement to effectively send and receive data between devices.

4.3.1 Design Visual and Description





System Interfaces

For the flutter application the Converter takes in the data over Bluetooth Low Energy and converts it to a word object that includes the 32 bit word and various metadata.

The Label Handler takes the word object and grabs the label bits from it. It then finds the definitions corresponding to that label in the Label Storage and sends the definitions to the Data Handler.

The Data Handler takes in the data definitions from the label, and the data word object and converts the data bits from the word into the formats specified by the label definition.

Finally the Display Functions take in the data word from all the different sources and displays the data in the correct format with the name of the data fields and their values properly represented.

The other data flow is that users can use the Label Creator to define label numbers and related data fields' names and formats. These are submitted and stored in the Label Storage.

The Label Storage keeps a sorted list of the user defined labels and some previously defined default labels.

The Label Sender will send a single byte to the Gatt service via the Set Label characteristic. This will notify the microcontroller to filter out other labels and only send the specified label over the connection. It will also handle the Send characteristic.

4.3.2 Functionality

Figma Prototype Link

Mock Up of our Application

The first screen is the main screen. It displays the data coming in over BLE if a connection is established. There are two drop downs, one each for the Label Name field and the Label

Number field. The Label Name dropdown allows the user to choose between different definitions for the same label number. The Label Number field allows the user to specify which label they want to receive data for. The buttons below the data give the option to connect to a BLE server or disconnect if already connected. They also give the option to switch between sending and receiving data. Finally the buttons give the option to define a label.

If not connected and the connect/disconnect button is pressed it brings the user to the third screen to choose a connection to send/receive data with.

If the user is receiving data, then the data will be being displayed through the shown screens, if the user is planning to send data they will be able to input the data to send by tapping on the data field and inputting the desired data.

When the Define Label button is pressed it brings the user to the second screen where they can define a new label. This is done by selecting a range of bits from the top left fields. Then users input a name for the bit field in the top right corner as well as a format for the data to be read from using the dropdown in the center. The bit fields in the middle show which bits are already accounted for and color codes them by the format through which the data should be displayed. The bit range edges are highlighted in gray with the range between them crossed out. These are subject to change.

With our current design, we're able to fully satisfy our needs of communicating between the Arinc429, the ESP32-S3, and the Flutter application. With this design we're able to leverage BLE communication to transmit the Arinc429 data to the Flutter application in a form that is both efficient and easily digestible for a user. Additionally this design allows us to directly communicate with the Arinc429 in an effective manner that alleviates unnecessary bloat.

4.3.3 Areas of Concern and Development

One of main concerns for our flutter application is to create an effective app layout that will be displayed similarly across different devices, since our application will be cross platform. Our goal is to design the application output so that it is consistent across all mobile devices. If the information displayed on the app is out of place/order, it can lead to misinterpretation of the Arinc429 data.

We'll aim to leverage Flutter's multi-platform development tools to properly design an application that is effectively displayed across all devices. Part of our decision to use Flutter was due to its ability to support developers with proper tools to assist with multi-platform development.

4.3.4 Strengths and Weaknesses of Design Approach

ESP32-S3 Strengths:

- Cheap
- Portable
- Support for WiFi and BLE
- ESP-IDF has good documentation

ESP32-S3 Weaknesses:

- USB plugin weak

Flutter Strengths:

- Cross-platform able
- Hot-Reloading
- Great Documentation

Flutter Weaknesses:

- Big App Sizes
- Relatively new language

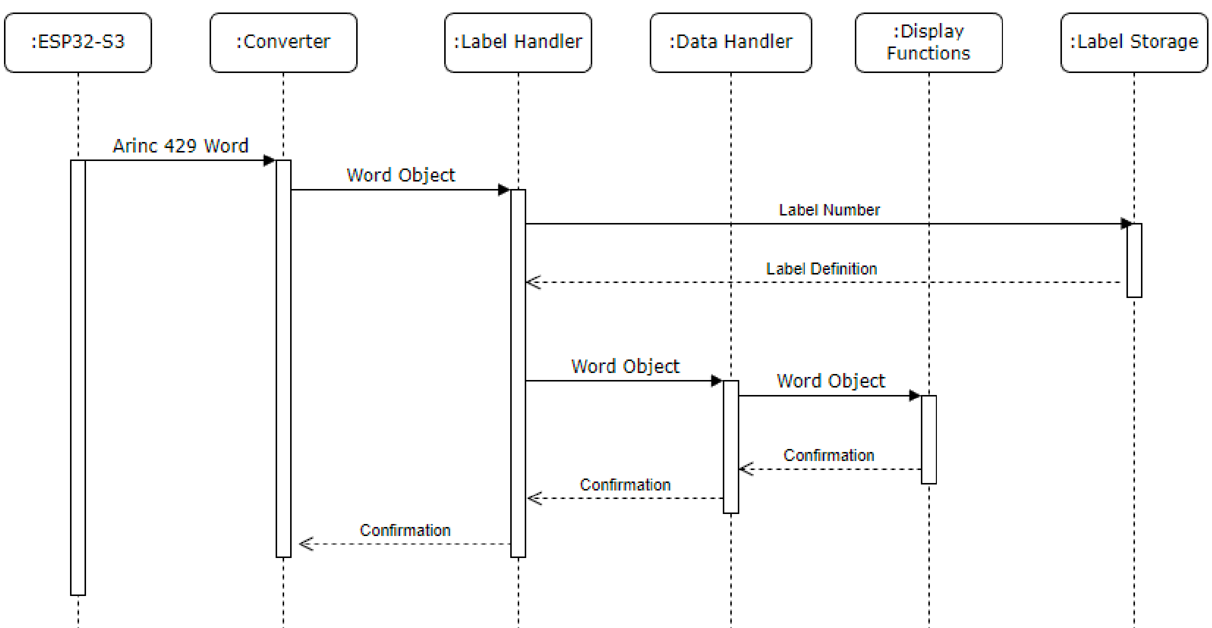
4.3.5 Design Analysis

[Will update this section with a reflection of the effectiveness of the design proposed in section 3.3 upon project completion as well as observation, thoughts, and ideas to modify the design]

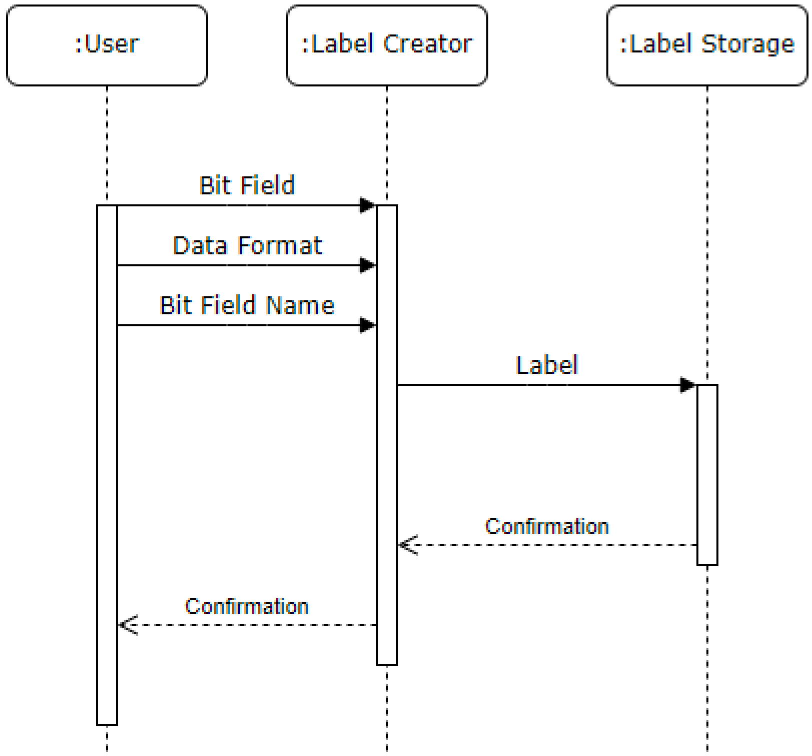
4.3.6 Design Plan

The Flutter application will comprise the 7 components described above in section 4.3.1. With the user interfaces as shown in section 4.3.2 and user flows shown in the Figma prototype.

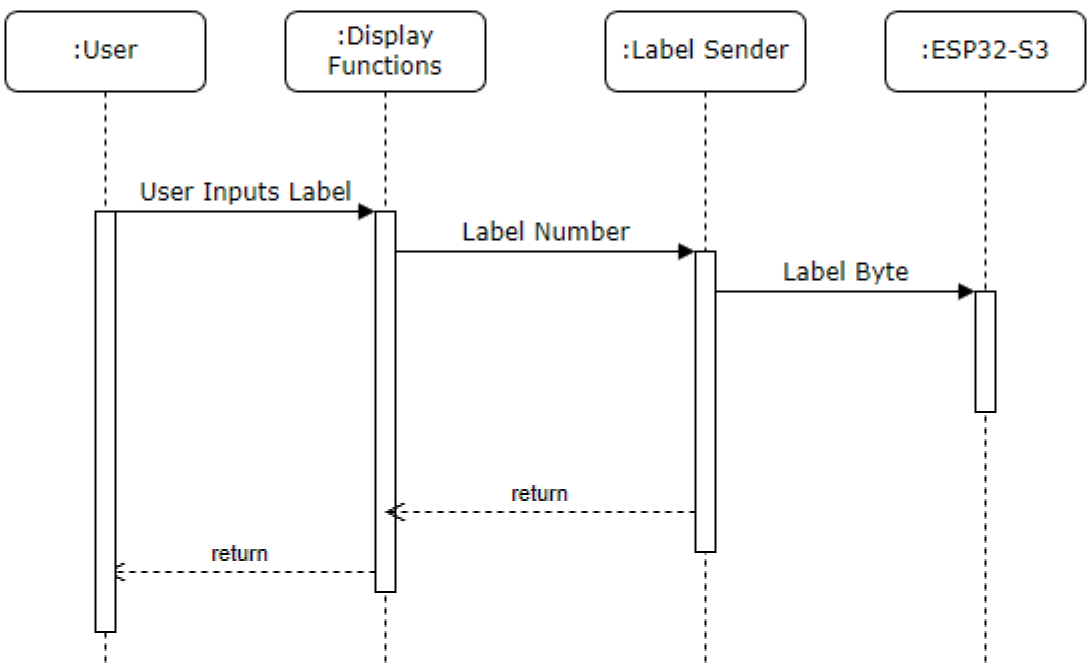
The following sequence diagrams describe the flow of information through the system for the various different types of user interactions:



An Arinc 429 word is sent over bluetooth



User Creates a Label



User Inputs a Label to Broadcast

5 Testing

Our strategy for testing will be to break down the process into several different steps. This includes testing different components individually, interfaces, integration, along with the qualities of the device from broader and more specific perspectives. Through this, we hope to cover all bases to ensure there is no way for any issues to pop up as we move through the testing phases which we are unaware of. We also hope that this will give us a better understanding of our design as a whole because of how much information it will show us about each individual part, that way if anything unexpected does happen to slip through, it will be able to be easily identified. The most difficult challenges we envision with testing the device will be incorporating BLE and learning how to check its performance, as well as the wide range of tests needed from firmware to Flutter.

5.1 Unit Testing

We will be unit testing each component of our Flutter application using Flutter's built in testing tools as well as Mockito.

Since we already have a plan for our different components we will just do unit testing over each component:

1. Converter:
    a. Use mockito to emulate receiving data over BLE and write unit tests to check that the data is correctly used
2. Label Handler:
    a. Mockito to emulate retrieving the label from the Label Storage
    b. Unit test creating the object to use later
3. Label Storage:
    a. Unit tests to ensure that everything being added is added in order and that the storage stays sorted
    b. Unit test for searching for an element
4. Label Creator:
    a. Unit tests for all of the different input fields
    b. Unit test for sending the label to storage
5. Data Handler:
    a. Unit tests for converting the bit fields into the correct formats
    b. Unit tests for converting the data object to the finished project
6. Display functions:
    a. Unit test that the input is handled correctly
7. ESP32S3
    a. We will be using the built-in ESP-IDF unit testing which uses the Unity testing framework

5.2 Interface Testing

Our project consists of one primary interface that will interact directly with users; that being our Flutter application GUI. For our GUI we will utilize Flutter's built in testing framework and tools to adequately test that communication between the application and the users is as expected.

Our project also consists of two primary units: the ESP32 and our Flutter application which will interface. These units will need to communicate the information that they receive/decipher with each other through BLE. We intend to test that proper communication is benign established by ensuring the information that we are receiving is the information that we are sending; information is being deciphered properly. By ensuring that the information generated by each unit is valid before sending and is valid after receiving, we can ensure that our communication between each unit is acting appropriately.

We intend to use built-in Flutter support for validation information before it is sent and after it is received on the Flutter side. We intend to implement a validation framework for our ESP32 firmware as well as leverage the ESP-IDF's built-in validation framework. Additionally, we will use an application, BLE Scanner for testing BLE connectivity and communication.

5.3 Integration Testing

https://docs.flutter.dev/testing/integration-tests

We will be using increment testing when checking the integration works properly. This will be done using a variety of different testing devices. Flutter has a testing method already incorporated for this using different testing packages, of which we will use the end-to-end test. The integration of other parts of the system will be tested for using documentation provided from the client. Because testing will be done incrementally, we do not believe there will be any roadblocks which are insurmountable, just several smaller ones which are more specific. These specific issues will be able to be referenced in the documentation as they pop up to allow for easier fixes.

5.4 System Testing

System level testing strategy is testing the whole project as a whole. If we put a certain input in, the whole project should work as expected and should expect a certain output. This is working as project as a whole, not the inner workings of the project.

We will most likely be using the built-in Unit testing in ESP-IDF on one end, and then the Flutter App testing tools on the other end. We will have to use both in conjunction as both will have to send and receive.

5.5 Regression Testing

First we will start by checking what dependencies or functions are needed to implement the new additions to our flutter application. Then we can see the control flow of our functions needed by the new additions and see what new changes need to be made to the app functionality and how those changes will affect the rest of the program output. We will test our new additions by also testing new code individually without any dependencies and partially test our new flutter app code often as we build the new additions.

We have to ensure on the flutter application side of our project that all bits from the Arinc429 32-bit word are still correctly converted to receive the right information on the user's side. Incorrect output information on the app could mislead field engineers.

We also have to check that BLE is still working to send data on the microcontroller and receive data on the flutter application because BLE contains each service and characteristic of the avionic device that we will be testing.

5.6 Acceptance Testing

We will set up a Webex meeting in order to demonstrate the functionality of our application in depth through the emulators included in our development environments. We will also demonstrate the functionality of the complete system on actual hardware using one of our android phones and the ESP32-S3 microcontroller.

5.7 Security Testing

We can do some penetration testing to make sure that people cannot access the BLE connection that we are using.

5.8 Results

[N/A: Will update upon project completion with testing results and their compliance with the requirements]

6 Implementation

Since we have split up our team into two separate teams, firmware team and flutter application team, we are able to implement different components of the project in parallel. As we make progress on both sides of the project we will continuously test the communication between the different hardware devices used in our project to make sure that each component can still send data to each other before we continue to the next major module in our project. In section 3.3 and 3.4 we have provided Sprints and a Gantt chart showing the incremental progress of the firmare and flutter application implementation.

7 Professionalism

This discussion is with respect to the paper titled “Contextualizing Professionalism in CapstoneProjects Using the IDEALS Professional Responsibility Assessment”, International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

From the IEEE code of ethics, it’s mentioned that as a member one must:

> *... accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or environment.*

This above requirement addresses that an engineer must be competent in their work by acting responsibly, communicating effectively, acting with integrity, and putting the welfare of the public above all else. As an entry in the table it would show as follows:

Decision Responsibility	Make decisions with the confidence that they will cause no intended harm and only act in the welfare of the public	Make decisions only with the intent to help and not harm

Work Competence can be summarized as being able to adequately perform in that field, *Financial Responsibility* can be stated as the ability to recognize restraints and act within them. *Communication Honesty* is one’s ability to act honestly when relaying information. *Health, Safety, Well-Being* is an individual’s responsibility to consider safety when working. *Property Ownership* can be stated as the ability to respect private information. *Sustainability* is the obligation one has to the environment. *Social Responsibility* can be summarized as the need to act with the best interests of society.

The IEEE code of ethics and NSPE version are very similar. Both have sections describing being honest and showing integrity. They each have sections promoting competence with engineering practices and the work which will be completed. They also have sections supporting safety.

However, IEEE has more statements regarding accepting criticism, not accepting bribery, treating all types of people fairly, and avoiding conflicts of interest. The NSPE version has sections regarding sustainability, financial responsibility, and property ownership.

7.2 Project Specific Professional Responsibility Areas

Work Competence	Yes, our client would most likely want us to work on only what we are supposed to. We must also make sure that high quality work is performed to ensure a reliable product. Deadlines must be set and met to ensure that project goals are met in a timely manner.	High, our team is not performing deceptive acts and works on our area of competence
Financial responsibility	Yes, because we have a certain budget, provided by our client and senior design class, that we are able to use towards our project (if needed).	High, our team is acting a faithful trustee towards our client
Communication honesty	Yes, we must truthfully update our client on project progress to receive accurate feedback. Communication helps the team know if we are on the right track or if changes need to be made regarding	High, We are reporting our understandable work truthfully with no deception

	project work.	
Health, safety, and well-being	No, there are no risks to health and safety possible	N/A, no way to risk the health of anyone
Property ownership	Yes, hardware given to us by the client must be used carefully and taken care of.	High, Our team is respecting the property and information we have received.
Sustainability	No, our product will not have a negative impact on the environment or natural resources on a local or global level.	N/A
Social responsibility	Yes, although it is just one product. Our product will be used to improve the avionics industry once it is completed. Our goal is to make sure we make it effectively and efficiently so that this will be the case.	High, Our team is indeed acting respectfully and ethically to enhance the honor and reputation of the engineering profession.

7.3 Most Applicable Professional Responsibility Area

One professional responsibility which is important to our project and our team has demonstrated a high level of proficiency is communicating honestly. To our project thus far, this has included communicating how much research we have done and how much we know to each other, our advisors, and the client. In the future as we complete more technical work, we will continue to communicate our progress honestly and effectively.

The impacts we observed were our advisors and our client being able to help us more efficiently, because if there is something we don't understand we communicate it effectively to them clearly and honestly. We have also found that we have been able to help each other understand things better when we communicate our knowledge honestly.

In the future, this will include being honest about the technical work which we are able to complete and capable of completing. It will impact our project by helping us to avoid spending extended periods of time on things we are stuck with as well as coming into less roadblocks in general.

8 Closing Material

8.1 Discussion

[Will update with project results upon completion]

8.2 Conclusion

At the end of this first semester we've been able to fully flesh out our design and our approach for completing this project as well as produce prototypes for us to work off of in the coming semester. For the next semester we intend to fully implement our Flutter application and ESP32 firmware in coordination with our Project Plan and Design plan, as detailed in section 3 and 4 respectively.

[Will update with reflection on goals met upon completion]

8.3 References

[1] Espressif Systems, "ESP32-S3-WROOM-2," Version1.1, July 2021 [Revised March 2023].

[2] United Electronic Industries. (2023), *ARINC-429 TUTORIAL & REFERENCE* [Online]. Available: https://www.ueidaq.com/arinc-429-tutorial-reference-guide#top

[3] Flutter. (2023), *Flutter documentation* [Online]. Available: https://docs.flutter.dev/

8.4 Appendices

Service	**Arinc 429 BLE**
UUID	48c174f7-6e8c-493d-9138-71a0602e57d8

Characteristic	Set Label
Description	Set the Lable that is being communicated
UUID	9f2ee2a8-3dfb-405d-8658-9bc14d6808a0
Properties	Write
Size	1 byte

Attributes:

Attribute Name	Type	Size	Description	Possible Values
64s digit	unsigned int	2 bits	The leftmost digit	0-3
8s digit	unsigned int	3 bits	The middle digit	0-7
1s digit	unsigned int	3 bits	The rightmost digit	0-7

Characteristic	Receive
Description	Receive the Arinc 429 word from the chip over BLE
UUID	81429dc5-2b11-420a-83ab-b61626fca73e
Properties	Read
Size	4 bytes

Attributes:

Attribute Name	Type	Size	Description	Possible Values
Parity	boolean	1 bit	The parity bit	0 - False 1 - True
SSM	unsigned int	2 bits	The Sign Status Matrix	0 - 3
Data	bit[19]	19 bits	The data from the word	19 bits
SDI	unsigned int	2 bits	The Source Destination Indicator	0 - 3
Label	3 digits Octal	8 bits	The label that determines how the data is interpreted	0-377

Characteristic	Send
Description	Send the Arinc 429 word from the Flutter app over BLE
UUID	9bf802ff-d75d-46a0-891b-c38003897fa3
Properties	Write
Size	4 bytes

Attributes:

Attribute Name	Type	Size	Description	Possible Values
Parity	boolean	1 bit	The parity bit	0 - False 1 - True
SSM	unsigned int	2 bits	The Sign Status Matrix	0 - 3
Data	bit[19]	19 bits	The data from the word	19 bits
SDI	unsigned int	2 bits	The Source Destination Indicator	0 - 3
Label	3 digits Octal	8 bits	The label that determines how the data is interpreted	0-377

BLE GATT API document

8.4.1 Team Contract

The Team: ENJNRs(Team 9)

Team Name: Team 9____________

Team Members:
1)Jared Staskal______________________ 2) Nick Morgan____________________
3) Nate Trotter_____________________ 4) Riley Millam____________________
5) Eduardo Contreras________________

Team Procedures

1. *Team Meetings:*
    a. Thursdays at 3:15, along with other times as needed.
        i. On Webex or TLA depending on the week.

2. *Methods of Communication:*
    a. Email for higher priority communication, and Discord for regular communication

3. *Decision-making Policy:*
    a. Consensus, if none reached backup to majority vote

4. *Procedures for Record Keeping:*
    a. Everybody will log their own minutes and put them onto the team report each week to archive them
    b. Meeting notes will be kept in a shared document

Participation Expectations

1. *Team Meetings:*
    a. It's expected that all team members attend team meetings.
        i. Inability to attend sound be communicated prior to meetings
    b. It is expected that all team members actively participate in meetings such as relay status updates, provide insight into issues, and pay attention in discussions.

2. *Responsibility for Team Assignments, Timelines, and Deadlines:*
    a. Teammates should communicate what they are working on
    b. Teammates should communicate when they think it will be done
    c. If they don't get it done, explain and communicate why they couldn't

3. *Communication:*

    a. Respond to others in a reasonable time frame
    b. Regularly update others on goals, progress, and completed tasks.

4. *Commitment to Team Decisions and Tasks:*
    a. Team members should communicate their opinions on decisions
    b. Teammates should communicate what tasks they will complete
    c. Team members should have commitment to complete what they say they will do and toward the team's decisions unless communicated otherwise

Leadership

1. *Roles for each Team Member:*
    a. Riley Millam
        i. Communication with the Client
        ii. Firmware Development
    b. Jared Staskal
        i. App Development
    c. Nick Morgan
        i. App Development
        ii. Firmware Development
    d. Eduardo Contreras
        i. Firmware Development
    e. Nate Trotter
        i. Firmware Development

2. *Strategies for Supporting and Guiding the Work of All Team Members:*
    a. Try to find information on provided documents or on the internet
    b. Reach out to other team members for help
    c. Reach out to client/professor

3. *Strategies for Recognizing the Contributions of All Team Members:*
    a. In the weekly report we will say what we have completed
    b. In our weekly meetings with the client we will present a powerpoint with the tasks we completed

Collaboration and Inclusion

1. *Skills, Expertise, and Unique Perspectives:*

    a. Eduardo Contreras
        i. App Development
        ii. C Programming

    iii. Embedded Systems Development
    iv. Shell Programming
    v. Computer Engineering Perspective

b. Jared Staskal
    i. App Development
    ii. C Programming
    iii. Software Engineering Perspective
    iv. Mathematics Perspective

c. Nate Trotter
    i. C Programming
    ii. Embedded Systems Development
    iii. Electrical Engineering Perspective

d. Riley Millam
    i. App Development
    ii. C Programming
    iii. Embedded Systems Development
    iv. Computer Engineering Perspective

e. Nick Morgan
    i. App Development
    ii. C Programming
    iii. Embedded Systems Development
    iv. Computer Engineering Perspective

2. *Strategies for encouraging and support contributions and ideas from all team members:*
    a. Be open-minded to new ideas, avoid tunnel vision
    b. Be kind to each other to encourage discussion
    c. Brainstorm as many ideas from everyone as possible
    d. Let people have take on some aspects of the project as an individual if they prefer

3. *Procedures for identifying and resolving collaboration or inclusion issues:*
    a. Let the other team members know they are not being inclusive
    b. Let the instructor know of the situation
    c. Let the client or advisors know

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
    a. BLE API Document
    b. ESP32S3 Firmware - Design
    c. Flutter Application - Design

2. Strategies for planning and assigning individual and team work:
    a. Discuss what each person's strengths and interests are
    b. Come to a consensus as with what each person should do based on this
3. Strategies for keeping on task:
    a. The understanding that this is a group project and lack of effort will reflect poorly, not only on yourself but also on the rest of the team

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
    a. As a group, we'll work to encourage the individual to be a more active participant

2. What will your team do if the infractions continue?
    a. Continued infractions will result in stern warnings
    b. Further infractions will result in discussion with professors

****

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*
b) *I understand that I am obligated to abide by these terms and conditions.*
c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*
1) Nate Trotter DATE 2-16-23
2) Riley Millam DATE 2-16-23
3) Jared Staskal DATE 2-16-23
4) Nick Morgan DATE 2-16-23
5) Eduardo Contreras DATE 2-16-23