

Arinc 429 Portable Receiver and Flutter Application



Eduardo Contreras, Nate Trotter, Riley Milliam, Nick Morgan, Jared Staskal
Advisors: Daji Qiao & Mathew Wymore | Client: Collins Aerospace - Colin Cox

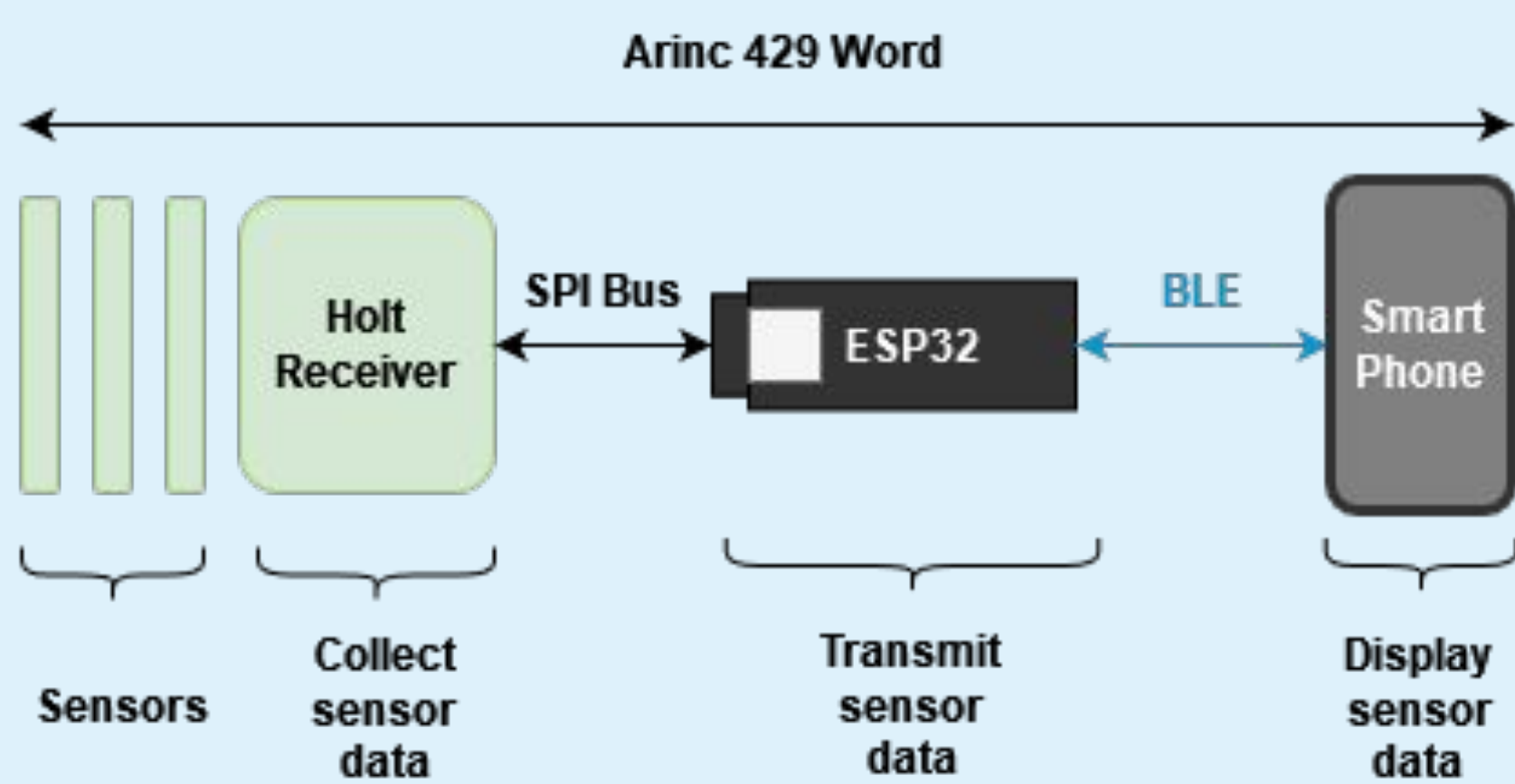


Introduction

Today, in the avionics industry, a bulky and expensive data receiver/transmitter is required to read data from sensors. To solve this problem, we've designed a mobile application paired with a microcontroller to replace this cumbersome equipment with commonly used technology.

System Functionality

In our design, the *Holt Receiver* communicates with the onboard sensors to collect data regarding the status of the avionic system. From there, this information is sent to our *ESP32* via an SPI bus which is then transmitted through BLE to our *Smart Phone* where it is parsed, formatted, and displayed for easy user inspection.



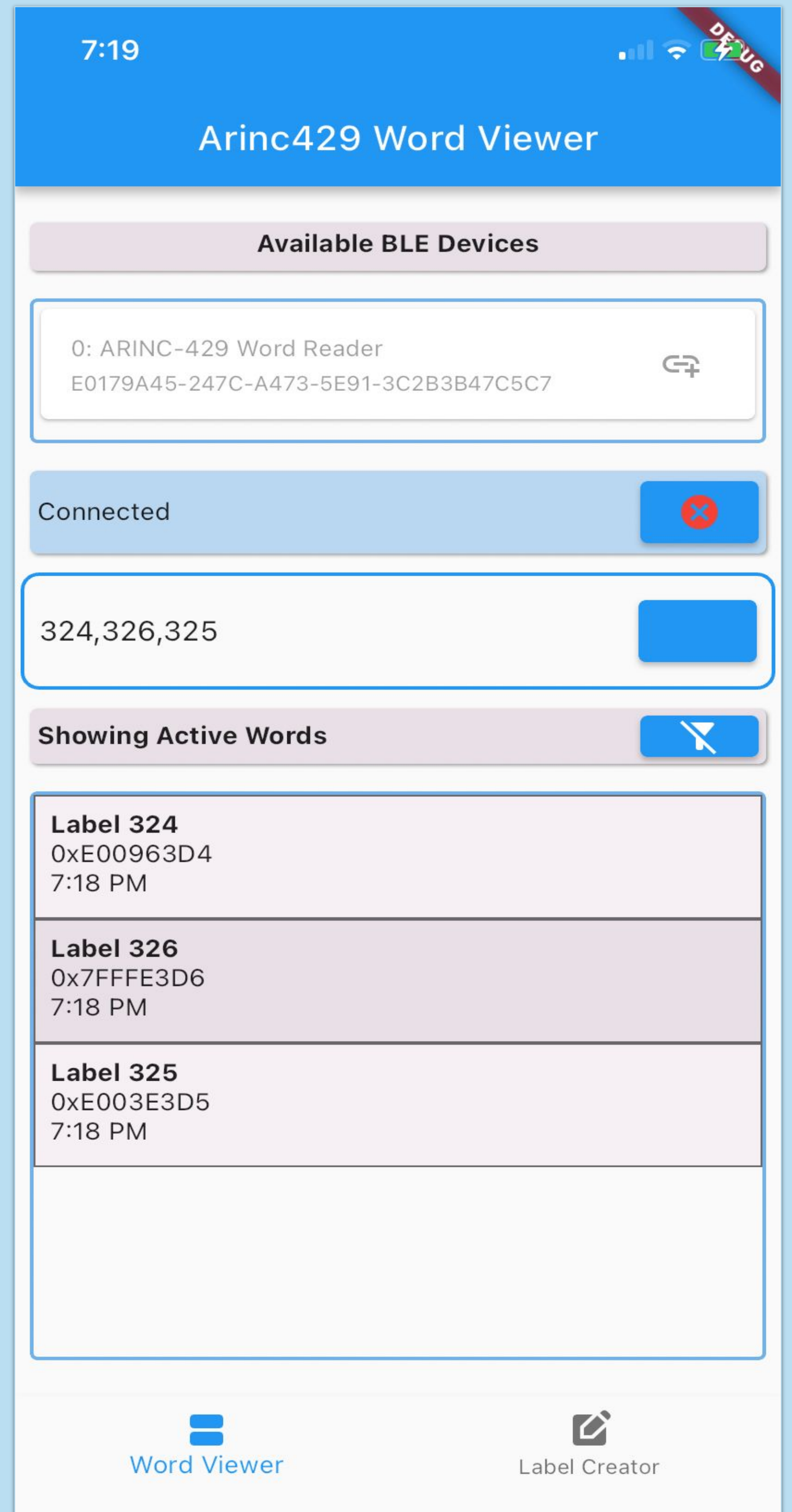
For security, our system is restricted to only communicate with our combination of *Holt Receiver* and *ESP32*.

Technical Details

- The firmware was developed in C on an ESP-32 S3 Microprocessor.
- The mobile application was developed in Dart using the Flutter framework with different developers using either Android Studio or Visual Studio Code.
- Bluetooth connectivity used the Flutter reactive BLE library.

Who will use this?

This was developed to help engineers in the avionics field inspect sensor outputs.



Requirements

Functional

1. The Flutter Application shall communicate with the ESP-32 microcontroller via Bluetooth Low Energy
2. When the Flutter application receives an Arinc 429 word the Flutter application shall display the data from the decoded word to the user
3. When the user defines a label the Flutter application shall store the label definition for later use
4. The ESP-32 should filter the labels and only forward a specific list of labels
5. When the Flutter application sends a label to the ESP-32 the ESP-32 shall add the label to the list of labels being allowed through the filter
6. The ESP-32 shall read in data from the Holt 429 Receiver

Non-Functional

1. The Flutter application must connect to the ESP-32 within 10 seconds of selecting the device
2. The Flutter application must be available for both Android and iOS
3. The Flutter application must be able to receive at least 100 words a second
4. The ESP-32 firmware must be easily updated and maintained
5. The Flutter application must be able to decode each word in less than half a second

Constraints

1. The project must use the Flutter framework
2. The project must use the ESP-32 Microprocessor
3. The mobile application and the ESP-32 must communicate via Bluetooth Low Energy

Testing

We tested the mobile app using our mobile devices: a Galaxy S22 and an iPhone 11. We also tested using a simulated Google Pixel 2 and a Pixel 3a.

We had a variety of unit tests for the Flutter application making sure that user created labels were being handled and stored properly.

For the Firmware we mostly did system testing as the major functional piece was simply forwarding the data. We checked that data was being forwarded to bluetooth correctly by intercepting the data with Android app called nRF Connect with our ESP reading data from an AHRS Demo testing device that includes multiple sensors.