CANdroid: Freeing ISOBUS Data and Enabling Machine Big Data Analytics

Yang Wang1, Andrew Balmos1, James Krogmeier1, Dennis Buckmaster2
1: Electrical and Computer Engineering, Purdue University
2: Agricultural and Biological Engineering, Purdue University

Introduction

The past decade has seen a tremendous increase in the number of agricultural machines whose internal data communications follow the ISO 11783 (ISOBUS) standard. In many cases, the ISOBUS data never leaves the vehicle because the exporting process is commonly a painstakingly manual task with little reward. However, advancements in big data analytics have increased the value of data and created a demand for an easy and efficient way of acquiring ISOBUS messages. Recent attempts to collect ISOBUS messages for observing in-field fuel usage [1,2] were made but the cloud integration was lacking. CANdroid, a recent re-development of ISOBUS [3], is a standard Android tablet enhanced by CAN peripherals that directly connect to the equipment’s ISOBUS. As a result, developers can leverage proven existing cellular and WiFi connectivity and exploit app development knowledge that software engineers already have.

This poster discusses the design of CANdroid including an example Android app that utilizes the Open Ag Data Alliance (OADA) API server. Additionally, we will present a reference design for processing the uploaded data using a modern, highly distributed stream processing scheme known as the Kappa architecture [5]. In particular, the log-centric database Apache Kafka [6] is used to coordinate a variety of single-purposed microservices that accomplish various processing tasks such as conversion of ISOBUS messages to raw data, creation of visualizations, such as yield maps, monitoring of equipment health, or pushing real-time alerts and data back to online consumers like CANdroid. This design paradigm allows for highly distributed, horizontally scalable processing chains that are simple to extend with little to no side effect to existing services.

Development

For hardware, we choose Nexus 9 as the main hardware platform for ISOBUS data collection and uploading. The tablet is built into a rigid enclosure for durability.

For software, there are three main parts:
1. We have built a custom Linux image that enables the support for J1939 kernel module and USB2CAN adapter drivers.
2. We have developed an open-source library called CANdroid-lib. It bridges the interactions between the kernel module and the Android app.
3. We have also developed CANdroid Logger app to collect data and upload them to the Cloud.

Experiment

We utilized CANdroid to collect ISOBUS data during a 2-hour long manure spreading session. CANdroid was connected to a tractor via its ISOBUS diagnostic port. There were 6 passes of manure spreading in total. In addition, during the manure spreading session, various aspects of the tractor (speed, RPM, location, etc.) were constantly changing. We bookkept some data such as vehicle speed and PTO on paper. Our intention was to compare the data recorded on paper with the parsed data from CANdroid to validate the usefulness of data collected from CANdroid.

Results

We utilized microservices to parse CANdroid’s collected data based on some known Parameter Group Numbers (PGNs) and plotted them. It is apparent that all these plots follow the same pattern that indicates there were 6 passes in total.

References