Design Document

Client
Vermeer

Project Members
John Shelley - Web Master - Software Engineer
Alec Johanson -Team Lead - Computer Engineer
Ahmad Muaz Mohd Khairi- Communication - Electrical Engineer
Koray Celik - Advisor
Arun Somani - Advisor

Android Bridge for Off-Highway Vehicles

Table of Contents:

Project statement

Modern off-highway/agricultural systems use outdoor rated LCD displays to implement user interfaces. These current solutions are either too expensive, or put users at risk being an unsafe distance from the machines. Vermeer wishes to replace these expensive display systems with a more inexpensive solution as well as provide a more flexible development environment. A J1939 CAN system is used on most Vermeer systems, but would like to be expandable to other protocols. Vermeer has expressed their wish of emphasizing research and development with solutions utilizing android systems due to their prior experience with this technology. The solution must be capable of both receiving and sending messages to the CAN bus from a safe distance.

System Level Design

System Requirement

- Must be operational outdoors
- Must be operational in a high-vibration environment
- Must be able to communicate over a 10m distance
- Must be operational for > 16hrs
- User friendly for the android user
- Extensible to other Vermeer machines

Functional Decomposition

The main communication pertaining to our project is between a VT (virtual terminal) and a controller on the unit (in our target case a 605 baler). The controller communicates to the VT through a CAN bus starting with a series of hand shakes, setting up the initial UI. Then during normal operation, the controller reads data from point to point connections with sensors. The controller analyzes the data and send UI update information through the CAN bus. The connection between the controller and the VT will now be wireless utilizing android instead of a proprietary VT system.

System Analysis

The android bridge for off-highway equipment has a three piece system. The different pieces allows us to keep the design more modular.

Controller

The controller is what obtains the information from outside diagnostic tools and then outputs that to the Can Bus which then translates that to distribute it to the android system. The controller is where most of the functional work is done.

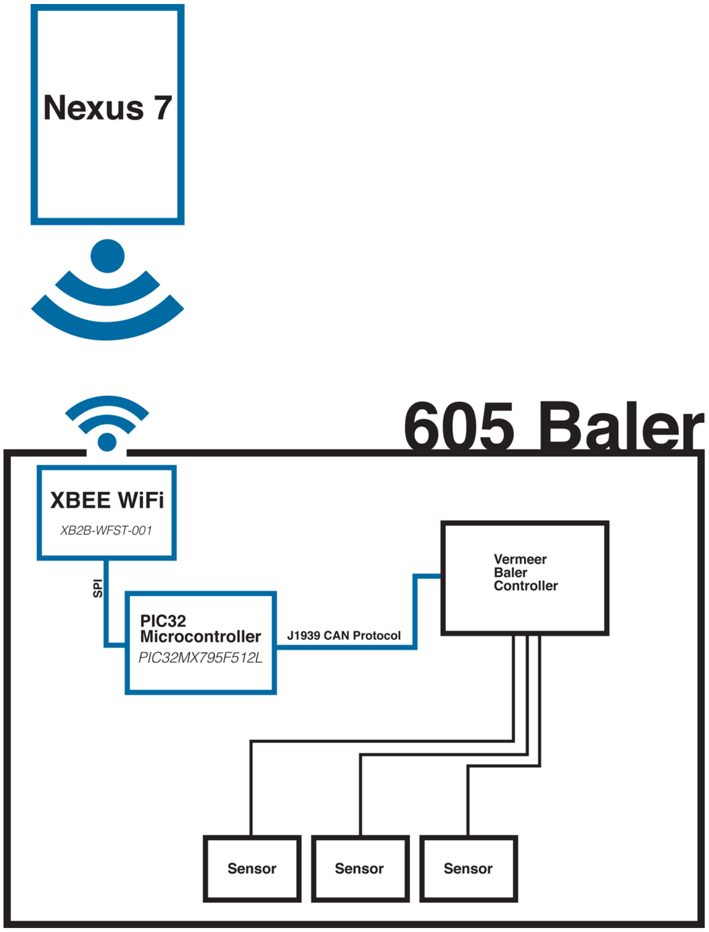
RF Bridge

The RF Bridge is what receives information from the Controller through the Can Bus. This will then distribute the information over a wifi signal. This signal must be secure and only intended for the specified device, therefor an AdHoc similar network approach will take place. We also have the decision to make with assigning IP addresses to the tablets or statically defining them ourselves.

Android Device

The Android device then takes the input from the RF Bridge, which we now know was received from the Controller through the Can Bus. The Android tablet will then output the User Interface in order to visualize what the controller is receiving from the diagnostic tools such as hay, corn, or bean harvesters and such.

Block Diagram of Key Concept



Detail Description

I/O Specification

In the scope of our project, the main I/O specification is in communication with the android device. The input to our portion consists of a CAN message sending data in a form similar to an .xml file. This consists of information for updating the user interface. Our output consists of a CAN signal to the controller directing the controller to execute tasks. The specifics of this protocol are defined in documents that we currently do not have access to.

Interface Specification

The interface specification for the Android design is fairly straight forward. The implementation of the interface will be very strongly structured. with a detailed instruction of what is to be placed and in a relative manner where. This will take place with a design documented that is already preloaded on the tractor's peripherals. The can bus will then transfer that to the android device which will translate what the user should eventually see.

This in turn makes it hard on the natural design of Android devices. Specifically Android has two natural design implementations, universal design through a multitude of device sizes, and predefined layouts for fast ui loading. The dynamic creating of designed layouts will take a significant amount of application usage on the device. However since this will be used in a long term processed activity like farming, the loading for the initial use shouldn't be a problem.

Hardware Specification

❖ Microcontroller
Model number: PIC32MX360F512L-80I/PT
This is a PIC32 microcontroller. This microcontroller was chosen because it has CAN integration built in. The microcontroller is reliable and widely used with a large community and capable of running at 80MHz.

❖ RF transmitter
Model number: XB2B-WFST-001
The RF transmitter used is the XBee Wi-Fi. This model uses a RPSMA Antenna which allows for a greater Wi-Fi range, since it allows us to mount our own antenna. This also allows the client to change the antenna in the future based on their specific needs and constraints.

❖ Tablet

The tablet is also a 2nd Generation Nexus 7 tablet. Vermeer chose this device because of two main reasons: the price, and the usability. The Nexus 7 is one of the cheapest Android tablets currently on the market. It's initial selling price was roughly at $300, and now has significantly dropped to approximately $180. The Nexus is easy to use because of the Android software that is being ran on top of it. You will notice that this also puts a limit on our software specification, but in no way hinders what is capable because of the Android SDK.

Software Specification

On the Client side of the process we have a fairly specific standard. Because of the tablet we have chosen to go with, namely the Nexus 7, we are prone to use Android and the Android SDK as our choice of software framework.

This however doesn’t put us with a roadblock or hindrance because of the nature of Android. This still gives us full control over the hardware, and allows for third-party open source libraries which are in abundance when it comes to Android.

Because we have decided to use the Nexus 7, this also eliminates any version of Android below 4.4, however it may be nice to include up to 4.0 since it is a more major version and could potentially include the first generation of Nexus.

Simulations and modelling

As for the simulation, we first try the CAN bus system interface by Victor which is called CANoe 7.6. It is kind of showing the virtual terminal from the equipment itself. It makes easier understanding on how the CAN bus system works.

We also used a prototype of the equipment from Vermeer and the VT system that has already been develop by Vermeer. We can test our android bridge directly to the system whether our system works or not.

Implementation Issues Challenges

In implementing the android bridge, we need to send the data simultaneously from the CAN to android. The problem is the system in the equipment is only connected by one wire, that means it can only give one data at a time. If we did not get the data for the equipment all at a time, the data that maybe need to pay attention did not get at the right time.Therefore our challenges is to get all the data once at a time.

Testing: Procedure and Specifications

Environment:

In order to test the system, we plan to modularly test the pieces first, and then integrate them later. For example the Android will get a set of test data to test the UI on, We will simulate the RF Bridge with a set of test data from a test controller. We will then integrate the systems in hopes of testing the process as a whole.

The environment for testing will be open fields, because this will be used on farm equipment. Most likely warmer weather, and potentially in slight rains. The basic general characteristic for testing will be the average farmer’s work day, since these will be mounted on Farm Technology.

Design:

Overall the functions of the specific sub pieces such as Android Device, and RF Bridge should both be able to communicate with each other as well. To test this we will have to open up a socket and make sure data is being communicated across as well.
We will require that the android create ui dynamically and test this individually, as well as through the communication of the RF Bridge. Likewise the RF Bridge should be able to translate the Can

Bus’ information. We will be testing this with a direct working Can Bus and sending various signals across. Those that would be verified and those that would not such as invalid signals.