OBD Bluetooth Interface and Android App

Project Concept Document

Patrick Landis (pal25)
Schuyler Thompson (sdt16) (Current Team Lead)
Advisor: Swarup Bhunia, PhD

January 21, 2013

1 Introduction

Every car in the United States since 1996 is required, by law, to contain on-board diagnostics. This standard, codified in SAE J1939 as on-board diagnostics II (OBD-II), is used for everything from reading fuel levels, to checking engine warnings for diagnostic purposes. The OBD-II specification specifically requires cars to make available a port in every car that can be used to read messages. A device is then plugged into this port to read messages over various protocols implemented by the car manufacturer. Currently, technicians must manually plug in a device into this port in order to read OBD-II messages. Our project aims to add wireless functionality for OBD-II via Bluetooth to be interfaced with Android compatible devices. Furthermore this project aims to make communication cryptographically secure.

2 Project Requirements

This project has quite a few objectives because there are quite a few parts to it. In Appendix I is a diagram to show the purposed information exchange for this project.

2.1 OBD-II Interface Component

- The biggest requirement for this project is the development of a circuit which will have the functionality to interface directly to a car's OBD-II port.
- This component will need to be able to act as a transceiver with the OBD-II port.
- The component will also be responsible for encoding and decoding messages for each of the protocols allowed in the OBD-II standard which are:
    1. SAE J/1850 (PWM/VPW)
    2. ISO 15765-2 / SAE J2284 (CAN)
    3. SAE J2411 (Single Wire CAN)

4. ISO 9141-2/14230-4 (Keyword Protocol)
5. Ford DCL (Mostly Depreciated)

- This component will be required to communicate with the central microcontroller in via some still-undecided standard protocol.
- This component will be powered from the OBD-II port.

2.2 Bluetooth Interface Component

- This component will be required to communicate with the central microcontroller in via some still-undecided standard protocol.
- This board will be responsible for communicating via Bluetooth to Android enabled devices.
- This component will be powered from the OBD-II port.

2.3 Central MCU

- This MCU must be able to communicate with the OBD-II interface component via some still-undecided standard protocol.
- This MCU must be able to communicate with the Bluetooth interface component via some still-undecided standard protocol.
- This component will be powered from the OBD-II port.
- MCU will be used to control message flow of the system.

2.4 Bluetooth Functionality Android-side

- We will be writing an Android app to receive data from the OBD reader.
- We will use the Android SDK to implement the bluetooth functionality
- The app will be able to display all of the basic messages from the diagnostic part of OBD.
- This includes the fault codes, monitor readiness, and whether the malfunction indicator light (MIL) is on or off.
- The OBD fault codes follow a pattern of N0000, a letter followed by 4 numbers. The app will contain the database of all of the standard codes and their meanings for easy lookup.

- The monitor readiness is indicating whether the all of the monitors used by the OBD system are ready to make a reading. They are reset when the OBD system is reset, and if the monitors are set to not ready, the car will fail any emissions check performed by a DMV.
- The MIL is the light commonly known as the check engine light.

2.5 Cryptographically Securing Messages

We will use Bluetooth's Secure Simple Pairing for security. This is mandatory for all BT 2.1+ devices.

- Use "Just Works" form of pairing.
- The pairing key will be stored on the non-Android side of communication.
- Both devices must acknowledge the pairing request.

3 Appendix I

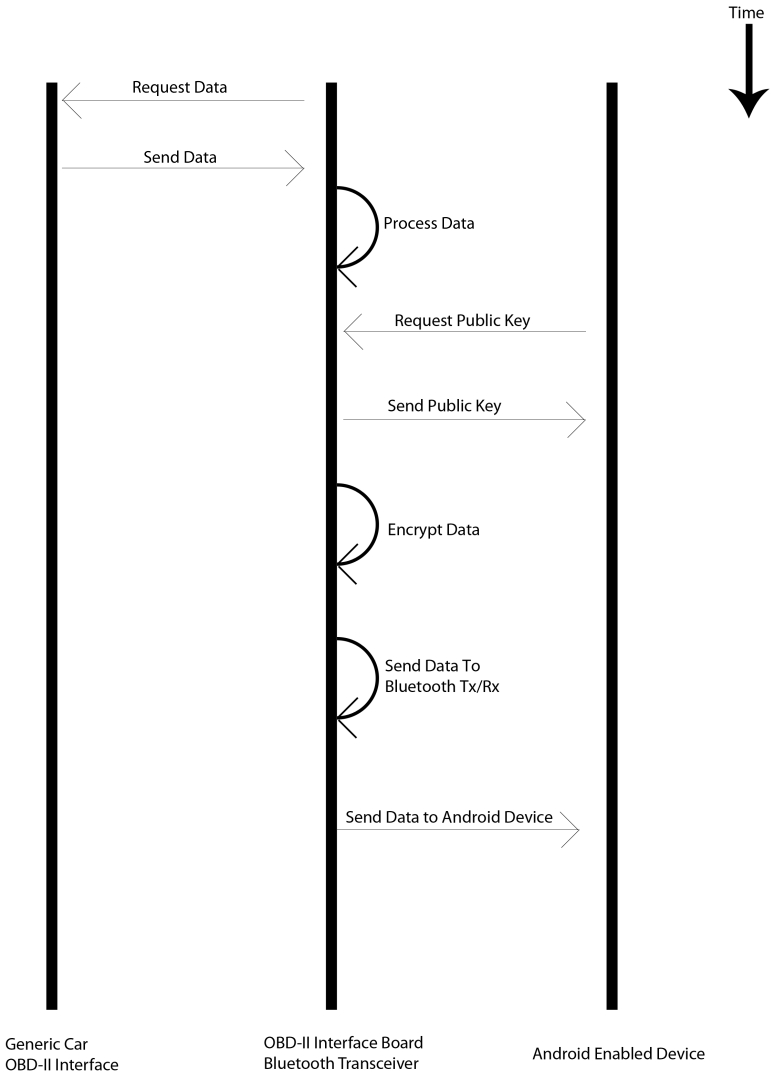

Figure 1: Information Flow Diagram