Wireless OBD II – CAN Bus
Embedded System Design

Carmen Bovalino
January 2014
Table of Contents
Objective ....................................................................................................................................................... 1
Background .................................................................................................................................................. 1
Description of Proposal ............................................................................................................................... 2
System Operation ............................................................................................................................................ 3
System Requirements ..................................................................................................................................... 3
System Architecture ....................................................................................................................................... 4
System Test Plan ............................................................................................................................................ 6
Deliverables ................................................................................................................................................... 6

Objective

The objective of this project is to develop an embedded system that will communicate with an automobile’s OBD-II (On-board Diagnostics) port via the ISO 15765 CAN signaling protocol. This project will encompass all phases of development including product concept, design, prototyping, and implementation. This project will also provide experience with the CAN bus, the OBD-II protocol, and wireless interfaces such as Bluetooth. The scope of the project will include PCB design and layout along with selecting, programming, and debugging microcontrollers as well as other peripherals.

Background

OBD-II is used in almost every car on the road today. OBD systems provide vehicle repair technicians as well as owners with the status of various vehicle sub-systems. Our project will provide a very simple, user friendly interface for vehicle owners, also saving mechanics time in the process. The final product will be a modular system providing simplified information to vehicle owners, while capturing detailed information in the background for an auto mechanic’s use.
Description of Proposal

This project will have a modular design including a main transmitter/receiver with an OBD-II connector along with a wireless sub-module including RGB indicator LEDs, a character LCD display, and pushbuttons. The main module will be attached to the OBD-II port in the vehicle and will handle the communication to and from the vehicle’s CAN bus. The microcontroller for this module will be the NXP LPC1768 Cortex-M3 with a 12MHz crystal. This module will also handle wireless communication such as Bluetooth and store CAN Bus data to a Micro SD card for later review. There will be a Mini USB port allowing connection to a computer for a control interface and possible DFU mode. The sub-module will communicate wirelessly with the main module and display important information to the driver. This module can be mounted in anywhere in the vehicle. This module will have twelve RGB LEDs which will be used to display the status of I/M (Inspection & Maintenance) Monitors as well as other diagnostic information such as RPM and speed. The twelve I/M Monitors test the operation of emission-related systems or components and detect out-of-range values. These monitors must be “Ready” in order to pass most state vehicle emissions inspections. Having a user-friendly display of the status of these monitors will help notify drivers when their vehicle is ready to be tested. This will save drivers as well as mechanics time by vastly reducing the amount of stops at an auto repair shop to check and see if the monitors are ready. The character LCD screen will display more detailed information for advanced users, but navigating through menus is not necessary for average users.
**System Operation**

The idea is to have auto mechanics provide the device to a customer when they need to wait for additional I/M Monitors to complete before returning or they are having a problem that the mechanic cannot reproduce. When the driver notices an issue while driving, they can press one of the buttons on the sub-module which will trigger a timestamp to be recorded for use with the logs. The auto mechanic can later review the logs from the device focusing around the recorded timestamp to investigate the cause of the issue based on real-time diagnostics at that moment. Advanced users could also purchase one of the devices for their own personal use.

**System Requirements**

Below is the list of tentative constraints for the system design:

- Main module shall include an OBD-II plug to connect to the vehicle
- Main module powered from OBD-II connector in vehicle (Voltage range 8V – 17V)
- Main module dimensions should be around 2.5” x 2.5” x 1.5” and shall fit under the dash of a vehicle
- When the vehicle ignition is off, the system shall draw less than 1mA
- When the vehicle ignition is on, the system shall draw less than 300mA
- The system shall wirelessly transmit real-time CAN bus messages or use a custom protocol to communicate with the sub-module via a Bluetooth SPP (Serial Port Protocol) connection
- The system shall also store CAN bus messages in flash memory (SD card)
- The sub-module shall provide a friendly user-interface with a menu system which will control the main module wirelessly through the Bluetooth connection
- The system shall have a defined and documented API for the Bluetooth interface
System Architecture

**Figure A – OBD-II Module Block Diagram**

OBD-II Module:

- NXP LPC1768
- 12MHz Crystal Oscillator
- CAN Transceiver (MCP2551)
- Bluetooth Module (Roving RN-42)
- Xbee 2mW – Series 2 (ZigBee Mesh)
- 3.3V Voltage Regulator
- 5V Voltage Regulator
- Micro SD Card Socket
- Mini USB Socket (VCOM, Debugging, DFU)
- Flash Memory (I²C or SPI)
- Beeper
Diagnostics Module:

- NXP LPC11U24
- 12Mhz Crystal Oscillator
- Bluetooth Module (Roving RN-42)
- Xbee 2mW – Series 2 (ZigBee Mesh) [Optional]
- RGB LED PWM Drivers
- 12 RGB LEDs
- Character LCD
- 8 Pushbuttons
- I²C GPIO Expanders
- 3.3V Voltage Regulator
- 5V Voltage Regulator
- Mini USB Socket (Power, Debugging, DFU)
- Beeper
System Test Plan

Below outlines the plans to verify correct functionality of the system as well as verify the system meets the specified requirements.

OBD-II Module:

- Create test mode in firmware to verify accurate data transmission between OBD-II module and sub-modules
- Create test mode in firmware to verify valid data storage to flash memory (SD card)
- Use a CAN bus emulator to spoof vehicle messages to our system
- Use an ammeter to verify correct current draw while vehicle ignition is on and off
- Verify correct operation of OBD-II module in a real vehicle

Diagnostics Module:

- Create test mode in firmware to verify accurate data transmission between OBD-II module and diagnostics module
- Use a CAN bus emulator to spoof vehicle messages to our system
- Have test users verify that the menu system is user-friendly and provide UI feedback
- Verify correct operation of diagnostics module and diagnostic module in a real vehicle
- Verify that correct I/M Monitor statuses are displayed with LEDs
- Verify that the correct timestamp is recorded to flash memory

Deliverables

- Overview of system and block diagram of system architecture
- Choose components for OBD-II module and diagnostics module
- Create schematic for OBD-II module and diagnostics module
- Create Bill of Materials for OBD-II module and diagnostics module
- Design and create PCB layout for OBD-II module and diagnostics module
- Create document outlining firmware architecture and test procedures for OBD-II module and diagnostics module
- Firmware development for OBD-II module and diagnostics module
- Surface-mount all components to OBD-II module PCB and diagnostics module PCB
- Create Bluetooth protocol specification document for OBD-II module and diagnostics module
- Debug firmware for OBD-II module and diagnostics module
- System testing and create results document for OBD-II module and diagnostics module