Final Project Proposal

Zak Rowland

Oregon Institute of Technology

Author Notes:

Submitted in partial fulfillment of the requirements of Embedded Project Proposal (CST 374)

Revision History			
Revision	**Date**	**Reason**	**Name**
1.0	June 8th, 2020	Published	Zak Rowland

Signatory Page

The signatures on this page indicate that the individuals listed have reviewed and agreed to the proposed project.

Name	Date	Relationship/Position	Signature
Zak Rowland	June 8th, 2020	Self	X Zak Rowland
Kevin Pintong		Program Director	X Kevin Pintong
		Senior Project Supervisor	X

Abstract

The proposed project is a wireless (Bluetooth) diagnostic tool that interfaces with the OBD-II port in a vehicle. This system will consist of two major modules: the Bluetooth transciever module that interfaces with the port, and the handheld module that includes a computer, touchscreen, and rechargeable battery. Features of the system will be limited to the essentials to keep costs low. The user will be able to read a list of diagnostic (trouble) codes and their descriptions as well as clear the codes. There will be another mode that will read sensor data and display it to the user including speed, oil pressure, and coolant temperature. A simple user interface will utilize touchscreen input for actions like selection, scrolling up and down, and changing modes or settings.

Table of Contents

Final Project Proposal

Project Management

Schedule

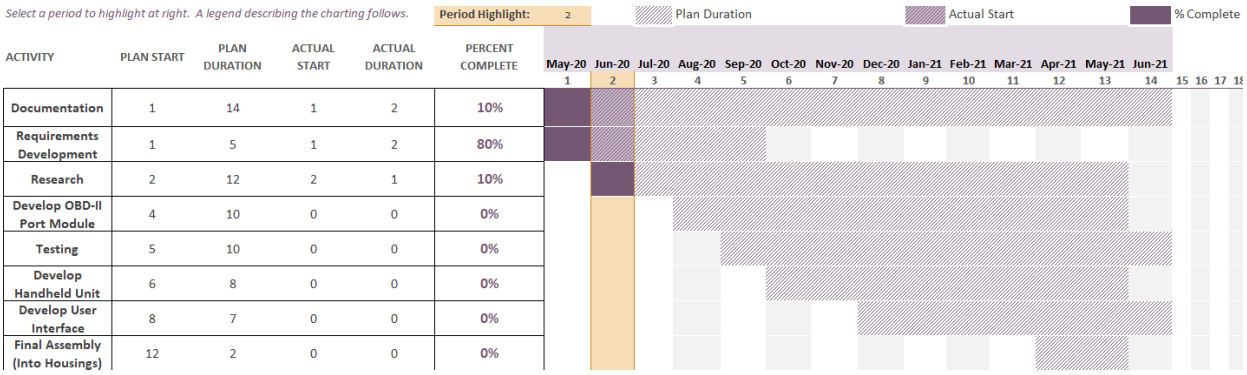

Figure 1. Proposed timeline for developing the project.

Non-Recurring Engineering Costs Estimate

Description	Cost Estimate	Total Cost
Computer or embedded systems engineer	$60,000 per year	
Product development and testing	$50	
Components	$150	
Housing manufacturing	$15	
		$60,215

Table 1. Estimated non-recurring engineering costs.

Change Management Procedures

In the event that a change must happen in the requirements or design of the system, engineering change requests will be submitted to the project supervisor as well as a memo if necessary.

Status Reports

The status of the project will be updated to management or the customer with memos, demonstrations, or meetings.

Skills and Knowledge Areas

Developing this project will require knowledge of many hardware components and protocols. First, I need to learn about the OBD-II port and how to interface with it, and learning

about the CAN Bus protocol will come along with this. Then, I will learn about Bluetooth and how to transfer and receive data to and from the OBD-II port. After that, I will learn how to drive a touchscreen and receive input from it. Finally, I have to design power circuits for the OBD-II port module and the rechargeable battery in the handheld module. I may also learn things about modeling and 3D printing to design a housing.

Conceptual Overview

Introduction

Ever since I got my first car, I have always found it fun to repair them or upgrade parts. I have had to borrow OBD-II diagnostic scan tools many times to perform my own repairs, however I have never had a tool like this of my own. Because of this, I want to create a tool that is all open source and low cost so it is accesible to as many as possible.

Problem Statement

I have always wanted a handheld unit that can read diagnostic and sensor information from an OBD-II port, but there are no products with the features I desire that are of a reasonable cost. I want to make vehicle repair and troubleshooting more accesible to anyone interested in doing so.

Intended Audience

This product is targeted mainly towards home mechanics who want to perform their own repairs. However, even paid technicians at a repair shop or electronics hobbyists/engineers could benefit from a low cost, open source option to interface with a vehicle wirelessly.

Proposed Project

I want to create a wireless handheld OBD-II diagnostic tool that uses a touchscreen. The system I propose consists of two primary modules: a Bluetooth transceiver that interfaces with the OBD-II port, and a handheld rechargeable unit that displays information on a touchscreen

and receives input from the user. The handheld units default mode will be scanning for and clearing diagnostic (trouble) codes, however there will be an option to switch modes and read sensor data like speed, coolant temperature, and oil pressure.

Relation of Proposed System to Existing Systems

While products like this do already exist, there is no reasonably priced solution (< $300) that provides the required features, such as having a dedicated handheld unit. The solutions that do exist, like the one pictured in Figure 2 below, can easily cost over $500 dollars, and are filled with advanced features.

Figure 2. An example of an existing solution that is high cost, from Amazon.com [1]

These products are aimed more towards professional mechanics and technicians who work on multiple vehicles a day or manage shops. The low cost options, like the one pictured in Figure 3 below, all connect to a proprietary phone app which I want to avoid.

Figure 3. An example of an existing solution that is low cost, from Amazon.com [2]

Many people, including myself, do not like installing apps from random unknown companies. Not only that, I find myself using my phone for other things while working on cars like a flashlight, music, or looking up information. My system is different because it will have its own dedicated handheld unit, be open source, and will be as low cost as possible while retaining the necessary functionality.

Deliverables

All source code and documentation will be delivered to program director Kevin Pintong as well as made open source. The components used to build the project will remain in my possession.

System Description

System Block Diagram

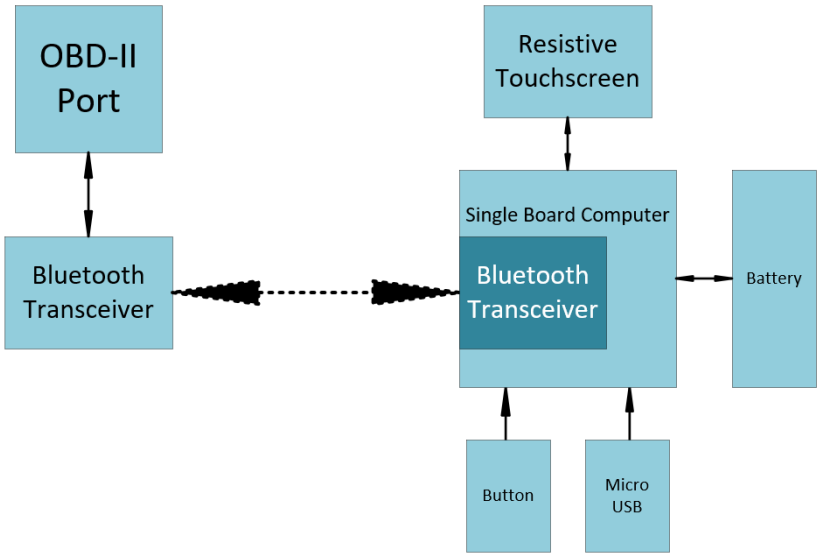

Figure 4. A block diagram of the proposed system

System General Description

Upon power up, the systems' default mode is to scan diagnostic codes. To do this, the circuit connected to the OBD-II port must read all the codes. This data will be sent from the OBD-II port module to the handheld unit using a pair a Bluetooth transceivers. The computer

inside the handheld unit must process this information, then drive a display for the user to see. Input from the user will be processed through the touchscreen as well. A list of codes (or a message indicating lack thereof) will be displayed to the user and will be able to tap on a code to view its description/possible cause. The user will have the option to clear all these codes, which involves sending a command over Bluetooth to the OBD-II module so it can act accordingly. There will also be an alternate mode for viewing sensor information live (at least 30 times per second,) and a settings menu to adjust various options like brightness or units.

Major Subsystems

The major subsystems of the project are the power system for the OBD-II port module, the rechargeable battery system for the handheld module, and the user interface.

Hardware Platform Description

I would like to use a single board computer such as a Raspberry Pi or microcontroller such as the PIC series. I think a Raspberry Pi would be a good option because they can run an operating system, have a lot of memory, and are well documented. I have never used a Pi, so more research is required to know if the computer could work for this project.

Software Platform Description

Since I am leaning towards using a Raspberry Pi, I will likely be writing programs in C or C++ and scripts in Python or Java for a Linux based operating system, such as Raspbian.

Requirements

1. The system shall work with the 2008 Nissan Altima OBD-II protocol, ISO15765-4 (CAN-BUS.)
    a. Other vehicles that use the same protocol may work with the system, however it is not required.
2. The system shall be able to read and clear diagnostic (trouble) codes.

    a. The user interface will display the diagnostic codes in list form with buttons to scroll up and down through the list.
        i. The list will display the diagnostic codes (e.g. P0011) only.
        ii. The user must touch one of the diagnostic codes to read the description or possible cause.
    b. The user interface will provide a button to clear all diagnostic codes.
3. The system shall be able to read sensor data at minimum 30 times per second including speed, coolant temperature, and oil pressure.
    a. The user interface will display the data in decimal format.
        i. The option for digital gauges may be implemented.
    b. The data will be displayed in units of miles per hour for speed, Fahrenheit for temperature, and pounds per square inch for pressure.
        i. The option for metric units may be implemented.
4. The system shall use Bluetooth 4.0 or greater for data transfer.
    a. The Bluetooth version shall be 4.0 or greater because previous versions do not support Bluetooth Low Energy (BLE.) The range of the connection is also improved with newer versions.
5. The Bluetooth transceiver circuit that interfaces with the OBD-II port shall be powered by the OBD-II port.
    a. A power circuit will be designed to ensure power from the port is reliable.
6. The handheld unit shall use a resistive touchscreen to display information and receive user input.
    a. The screen shall be resistive instead of capacitive because resistive touch is lower cost and can be used with gloves.
    b. The touchscreen of the handheld unit shall be 3.5” or greater diagonally.
    c. The touchscreen of the handheld unit shall operate at a resolution of at least 320 x 480 pixels.
7. The user interface shall not require multi-touch or swiping.
    a. Operation with single presses simplifies the user interface and is more accessible to those with disabilities.
8. The rechargeable battery shall be recharged via a Micro-USB port.

    a. The port shall be Micro-USB instead of another version, such as USB-C, because Micro-USB components are widely available at a lower cost.
9. The rechargeable battery shall power for the handheld unit at full load for at least 2 hours.
    a. The rechargeable battery must include protection and charging circuitry.
10. The charge level of the rechargeable battery shall be displayed on the user interface.
    a. The charge level will be displayed as a percentage, 100% being fully charged.
11. The brightness level of the touchscreen shall be adjustable in the user interface in 10% increments, 100% being maximum brightness.
12. The housing of the handheld unit shall include a magnet in the back.
    a. The magnet will allow users to easily mount the device on the various magnetic dash mounts available (or any other magnetic surface.)
13. The handheld unit shall include a physical ON/OFF power switch.
14. The system shall use a single board computer or microcontroller.
15. The code and documentation developed shall be open source.

Glossary

Term	Definition
OBD-II	OBD-II stands for On-Board Diagnostics (version 2) and is the interface used in all vehicles model year 1996 or newer.
CAN Bus	CAN stands for Controller Area Network and it is the bus protocol used in vehicles and other embedded systems to send and receive data to and from all the various devices (nodes) connected to it. The OBD-II port has connections to the CAN Bus.

Appendix

(empty)

References

[1] Amazon, Autel MaxiCOM MK808BT Diagnostic Scan Tool, https://www.amazon.com/dp/B07MBSZH48/

[2] Amazon, OBD2 Scanner Bluetooth 4.0 OBDII Scan Tool for Android Device & iOS, https://www.amazon.com/Scanner-Bluetooth-Android-Foseal-Diagnostic/dp/B07MB8XH6Q