## Table of Contents

8. Resources and Schedule
   8.1. Tools, environments, Hardware and Software
   8.2. Schedule

9. Technical Concept
   9.1. Preliminary Design – High level architecture
   9.2. Feasibility

# 1. Problem Statement

Our team will design the software for a device that will be used to monitor traffic in a Controller Area Network on a Chrysler automobile. This design will include the user interface to control the device, the handling and decoding of network messages after they have been collected by hardware, the display of these messages on the device's LCD screen, and the storage of these messages in a log. We will also provide an API to ease future expansion of the device's capabilities.

## 1.1. Background and History

Chrysler uses micro-controllers throughout the electronic subsystems in their automobiles to monitor and adjust automotive operation, using a variety of Controller Area Network protocols for communication. Because there are various protocols for communication, Chrysler would like a device that can monitor a network for traffic using any of these protocols, as well as providing RS232 connectivity for communication with a charging unit and PCMCIA capability to upload data to a PC for analysis. There is currently no device available that provides these capabilities for use in design and troubleshooting situations for Chrysler automobiles. A device such as this is needed to monitor Controller Area Network traffic, logging up to an hour of network messages, translating the meaning of the messages, and displaying these decoded messages, as well as upload the stored log to a PC.

## 1.2. Issues

### 1.2.1 Specific Customer

Our specific customer and sponsor is Dr. Medidi. He is working with Chrysler to develop this device, and will provide the hardware upon which the software developed for this project will eventually be implemented.

## 1.3. Value of a technology solution

This device will allow Chrysler development engineers to more easily debug and test the interaction between various components in a vehicle. It will facilitate easier collection, analysis, and understanding of data transmitted through the vehicle's network. Because the device will maintain a sequential log of network messages, it will also make timing analysis possible.

### 1.4. Competitive products

Currently there is no device that actively monitors a network using any of multiple protocols, as well as PCMCIA and RS232 connectivity.

### 1.5. Business environment

In the aggressive and competitive nature of the automotive business, time to market and performance are critical. As new technology is developed for automobiles, it must be tested not only for proper operation on a bench, but proper interaction with the other systems within a vehicle. This device our team is writing the control software for will speed and improve the process for testing such technology as it is introduced, thereby giving Chrysler a competitive advantage. Our software must be designed to maximize the functionality of this device while being as having intuitive as possible an interface to minimize the time required learning how to use the device.

## 2. Requirements

### 2.1. Goals – What problem are we solving

We intend to design the software for the device mentioned above. We must develop a user interface, the network message decoder, the message storage log, and the display software. We must also provide an API with documentation for the code we generate so future modifications and additions to the software are easily made. We will not design the hardware for this device – it will be provided by Chrysler via Dr. Medidi. However, our code must be designed with the hardware in mind so that it will fit within the memory the device will have and operate properly with the device's micro-controller architecture.

### 2.2. Functional Requirements

The software for the device must -

1) Translate the data received from the network into subsystem messages that can be understood by the engineer.
2) Display the sequence of messages as they are observed on the network.
3) Keep a log of the messages capable of storing one hour of data.
4) Permit configuration of the device from the keypad
5) Create an API that facilitates the above requirements

## 2.3. Non-Functional Requirements – architectural characteristics

The software for the device will -

1) Provide a user-friendly interface with an LCD panel and a hexadecimal keypad for a user input and output
2) Provide a PCMCIA interface for the hookup to a remote PC for performing off-line analysis and diagnosis
3) Provide a RS232 interface for interfacing with a vehicle's external charging circuitry
4) Must be able to work on top of a simple kernel with minimal system calls and not using memory management
5) Must be able to retrieve data using the J1850 protocol
6) The system needs to be modular so it is expandable to accommodate the addition of the CAN 2.0 and SCI network protocols at a future date
7) Provide an API to allow for easy future expansion and modification of the device software

## 2.4. Constraints – the boundaries

### 2.4.1 Hardware Constraints

- This team will not be responsible for the creation of the device hardware - the hardware design will be provided by Chrysler. If there is enough time at the end of the Spring 2000 semester, we will port the software developed with an emulator to the hardware platform.
- The memory available on the destination hardware is relatively limited, therefore our software must be compact and efficient.

### 2.4.2 Software Constraints

- User input is restricted to the hex keypad and the four directional keys
- Output messages must be short enough to be displayed on the 3"x5" LCD screen
- A menu system will allow the user to modify the various parameters defining system operation without requiring a complex input device
- The software we develop must be small enough to fit into the available memory in the final device
- Software must require minimal kernel functionality, and not require dynamic memory management (a smaller kernel reduces the code size)
- The software must be modular to allow use of multiple network protocols (J1850, SCI, CAN 2.0) without specialized high-level code
- Software must provide a well-documented API to easily add support for additional network messages in the future

### 2.5. Business philosophy issues to which the system must conform

Much of the information used in the development of this software and hardware system is proprietary, and therefore must not be made generally available beyond the parties involved in the design. This device is intended for Chrysler internal use, and many of the details as to why choices have been made with regard to the hardware are not made known to us.

## 3. Specifications

### 3.1. Hardware Specifications (note: these requirements constrain the software that will be run on the final device)

- Handheld device
- 3”x5” LCD screen for user output
- Hex keypad and 4-way cursor movement keys for user input
- PCMCIA connectivity to a PC for data upload
- RS232 input for interfacing with external charging circuitry
- Support for J1850, SCI and CAN protocols

### 3.2. Software Specifications (note: these are the requirements to which we will design our system software)

- Developed and run on emulator
  - Ported to device-specific hardware platform as time permits
- Fully configurable with a hex keypad and cursor movement keys
  - Provide an easy-to-understand menu system to operate the device
- Able to store one hour’s worth of network traffic in memory
- Use modular design for network protocols supported
  - Support for J1850 required
  - Support for SCI and CAN optional as time permits
- Provide an API for high-level modules
- Provide documentation for the API

# 4. Cost/benefit Analysis

## 4.1 Estimated life cycle costs (3-5 yrs)

This device will be updated to suit the needs of Chrysler design engineers in the future, and so must be designed with extensibility in mind. To this end, our software will be modular so additional self-contained elements can be easily added. Furthermore, the API we provide will allow engineers to create additional high-level capability relatively easily. The modular nature will also make upgrade to different hardware fairly easy, because there will be little hardware-specific code to change. Therefore, cost to maintain and upgrade the device software should be relatively small in the foreseeable future.

## 4.2 Tangible benefits – estimated returns

The increase in Chrysler's design engineers' ability to test, debug, and fine-tune the electronic components in their vehicles will allow them to bring these technologies to market faster and with fewer problems. This process improvement will save more money than the investment in materials.

## 4.3 Intangible benefits – to be completed

# 5. Risk Assessment

## 5.1. Potential Risks

- Unavailability of necessary tools (emulator for HC12 micro-controller, cross compiler, debugging tools, etc.)
- Lack of direct communication with eventual user, Chrysler vehicle designers
- Portability of software to the hardware platform
    - Software is too large to fit in available memory
    - Requires too many kernel services

## 5.2. Risk Mitigation

- Unavailability of necessary tools (emulator for HC12 micro-controller, cross compiler, debugging tools, etc.)

Extensive research will be done to locate an emulator for the HC12 chip. If one that supports the specialized features in certain versions of the HC12 cannot be located, a general HC12 emulator will be used and we will simulate the specialized features in software. There are emulators available for this family of chips, so we will be able to develop a proof of concept for our software system.

- Lack of direct communication with eventual user, Chrysler vehicle designers

Dr. Medidi is our primary interface with the eventual user of this device, and he is our customer, so is responsible for the definition of this project. We are developing this product for Dr. Medidi, who is operating on behalf of Chrysler. Conference calls will be made with Dr. Medidi and the Chrysler engineers to obtain direct input from the ultimate users of the product about what they need. We will also be providing a preliminary user menu system and API specifications to Chrysler for validation.

- Portability of software to the hardware platform

We will communicate with Dr. Medidi to help monitor our progress and gain input on how to prevent excessive kernel requirements, as well as the eventual capabilities and constraints on the final hardware platform. This will help us develop software that will fit the final destination hardware with minimal rewrite required.

    - Software is too large to fit in available memory

    Reuse of high-level code will minimize the compiled software's memory requirements. This also facilitates a common structure to the handling of data, with only the protocol-specific software being unique.

- Requires too many kernel services

We will provide our own rudimentary memory handling routines, thus reducing the requirements of the kernel to managing processor tasks. We will refrain from using system calls in our code, further reducing the burden on the kernel to provide services for our use.

# 6. Organization

## 6.1. The Team Roles

**Leader** – Dave Swanson
Responsible for coordinating and running meetings

**Recorder/Document Coordinator** – Ben Komada
Responsible for keeping and distributing detailed notes of all meetings and maintains the team notebook

**Communicator** – Makia Minich
Responsible for all communication between the team and outside parties

**Research Coordinator** – Rob Allred
Responsible for coordinating and leading outside research and setting priorities on information to be gathered

**Facilitator** – Derek Flint
Acts in a neutral role to help resolve disputes; responsible for ensuring sound meeting structure

## 6.2. Rules, reporting

### 6.2.1. Meetings

Meetings will be arranged and held on the weekends when a quorum of the team members can get together to work and make decisions. Impromptu meetings can be held before and after CSE 449 to coordinate efforts and make decisions as to how to proceed and future efforts. In addition to these we will be meeting with Dr. Medidi every other Thursday from 1:00 to 2:00 PM to report on our progress and on any new developments from the Chrysler Corporation. Dr. Medidi is to be our sponsor and will provide information and communication with the Chrysler Corporation.

### 6.2.2. Meeting Agenda/Structure

At every meeting, we will begin with a structured agenda followed by an open session where discussed takes place. During the structured section:

Everyone (in turn) reports the progress they've made
Everyone (in turn) reports any problems they've encountered, fixes they've done and suggestions from others.
New developments/discoveries based on research.
Update schedule and top priority items.

Any new questions relating to direction and any queries in current information.
Assignment of new action items.

At all meetings, the agenda will be followed. Upon completion of scheduled business, open session will be used to discuss topics of concern to the team. Both the Leader and Facilitator will ensure that proper process is observed during meetings.

### 6.2.3. Minutes

Minutes will be taken at each meeting by the Recorder and distributed by email to those that were not at the meeting within 24 hours. These will include a short summary of pertinent developments from that meeting and a copy will be kept in our course notebook.

### 6.2.4. Performance Evaluations

All team members will be evaluated by the other team members after every incremental prototype completion. The numbers will be made public to the team. These evaluations will affect the team members' grades much as in 286/386. There will be a total of 4 points to be awarded and divided as each team member sees fit. These points can be averaged and used as a multiple on the total grade at the end of the semester.

### 6.2.5. Design Changes

Any design changes that affect another member of the team must be made only by unanimous consent of the team. Design changes, which affect only one person will be made only majority vote. All design changes are to be made only to improve the structure of the design, not to merely make the coding of one part easier.

# 7. Design Process

## 7.1. Methodology

We will be using a waterfall approach to the design of this system, using top-down design for the software architecture. This will allow us to most effectively adapt the system to Chrysler's needs. High-level software can be designed for general lower-level protocol implementations, this maximizing the amount of code reuse during future additions. Top-down design will also allow us to compartmentalize the functions we create into a "black box", to which the API allows access.

## 7.2. Deliverables: documentation – what, how

### 7.2.1 – Format

Format will be in either Microsoft Office 97 or 2000 for all documents. Documents will have a header, which will include the following: the page number, the document name and the team or sub-team name. Standard font is Times New Roman with topic headings in size 14 and body text in size 12. Points need to be listed in outline form in numbered format such as used in this document.

### 7.2.2 – Document Coordinator

The recorder will act as document coordinator this roll will be handed around to allow all team members to partake in the engineering process.

### 7.2.3 – Version Control

The team leader will provide revision control. Archives will be made regularly so progress and functionality will not be lost in the event of catastrophic errors.

### 7.2.4 – Document Approval

All documents are to be unanimously approved by the team before they are released. Any changes that must be made must be submitted to the document coordinator.

# 8. Resources and Schedule

## 8.1. Tools, environments, Hardware and Software

### 8.1.1. Hardware

Hardware circuit design of the hand held unit has been provided to the team. The Chrysler Corporation will also be responsible for providing the automotive electrical system to test the diagnostic unit. However, the hardware must be considered while developing the system software so that software will be appropriate to load on to the final hardware device.

### 8.1.2. Software

A cross compiler to be supplied or recommended by the client will run on Windows NT or Linux-based PCs. The choice about which OS will be utilized depends on the availability of software development and testing tools for the hardware components. The software we will create will be compiled and tested on the system we choose, then uploaded to the hardware described above if time permits.

## 8.2. Schedule – to be completed

# 9. The Concept

## 9.1 Proposed Use of Technology

At this time the technology involved will include the following central areas of interest with the following division of work. The Chrysler Corporation will be providing the hardware including any input devices and LCD display, hex-pad, mouse, and any output ports. This will also include the circuit board design that connects these components. In addition they will need to provide a test circuit or wiring harness similar to that used in their automobiles to allow us to test the software design with real-world test hardware. Our team will be providing the software interface between the user and this diagnostic device and the controllers in the automotive electrical system. A lot of the engineering will include researching individual pieces of this API which will include but may not be limited to, a system Kernel, a software interface for the LCD screen, a software interface for the hex-pad, and interfaces with the different input/output protocols. Our team will be responsible for integrating these pieces and developing any that we cannot locate from the resources at our disposal.

## 9.2 Preliminary Design

The Design of this unit will be separated into five different, specific tasks that will be running on a common kernel and finally the GUI interface, seven parts in total. These parts will be borrowed as much as possible from preexisting software and integrated to conserve as much time and energy as possible. A GUI will be added on top of this operating system to allow effective user interface with the unit.

## 9.3 Feasibility – to be completed