

Rapid Prototyping Environment for Climate Control Development

Final Proposal

2/21/2014

Michigan State University

Senior Design - ECE 480 - Team 4

Project Sponsor:

General Motors

Project Facilitator:

Lixin Dong

Team Members:

Guisepppe Ferro

Omar Ali

Weiham Yan

Sitong Ge

Ricardo Johnson

Yingshuyu Li



Executive Summary

During the development of a climate control system in a vehicle, Rapid Control Prototyping (RCP) must be done in order to ensure that the optimal design is chosen. Accomplishing this is a timely task, so software is used to virtually design the system for the ease of modifying parameters in the developmental stage thus saving time and money. Due to the benefits that RCP offers, General Motors is focusing their resources into developing an RCP climate control system through the software dSPACE. This software allows for the virtual design of an Electrical Control Unit (ECU) for the handling of a real time signal communicated through a Local Interconnect Network (LIN) which then controls actuators and sensors to modulate and detect the lighting and temperature in an automobile. With the requirements for signal systems increasing, analog data transfer techniques are showing their limitations. LIN permits General Motors to transition from analog to digital communication which allows for the advanced processing of signals sent from the ECU and delivered to the actuators and sensors.

After designing, implementing and testing the RCP climate control environment, the system will provide greater accuracy and control of LIN based actuators and sensors to modulate temperature, save on wiring costs, and offer the ability for real time correction during signal acquisition.

Table of Contents

1 - Introduction.....	2
2 - Background.....	2
2.1 - dSPACE	
2.2 - Phases	
2.3 - Hardware-in-the-Loop (HIL) Simulation	
2.4 – Local Interconnect Network (LIN)	
2.5 - Controller Area Network (CAN) vs LIN	
2.6 - Companies uses of LIN in HVAC systems and other systems	
3 - Objectives and Design Specifications.....	5
3.1 - Mission Statement	
3.2 - List of Objectives	
3.3 - Design Specifications	
4 - Fast Diagram.....	7
5 - Conceptual Design Descriptions.....	8
5.1 - Open Loop Controller	
5.2 – Proportional Integral (PI) Closed Loop Controller	
5.3 – Proportional Derivative (PD) Closed Loop Controller	
5.4 – Proportional-Integral-Derivative (PID) Closed Loop Controller	
6 - Ranking of Conceptual Design Specifications.....	8
7 - Proposed Design Solution.....	9
8 - Risk Analysis.....	10
8.1 - Sensors Disabled	
8.2 - Sensor Error or Dysfunction	
8.3 - Actuators Disable and Errors	
8.4 - Controller Errors and Software Error	
9 - Project Management Plan.....	11
9.1 - Non-Technical Roles	
9.2 - Technical Roles	
9.3 – Gantt Chart	
10 - Budget.....	13
11 - References.....	14

1 - Introduction

With a constant need for new sophisticated technology to meet the consumer's needs, General Motors climate control systems must be modified in order to stay competitive. GM's climate control systems are moving from analog interface actuators and sensors to digital devices that use LIN serial communication to communicate between the Electrical Control Unit (ECU) and the actuators which are used to modulate the temperature on the interior of the vehicle. Designing and building an ECU is not a simple task. In the phase of control algorithm implementation, register level programming for the ECU is required. For these reasons, the designing and testing of control algorithm implementation became extremely difficult. For reducing the level of difficulty, Rapid Control Prototyping (RCP) systems have been developed over time which removes the need for register level programming in the prototyping phase of an ECU [10]. Alternatively, Simulink models can be created through the Real Time Workshop (RTW) in order to create a controller that will be implemented into the RCP environment.

Analog systems signals degrade over time, provide less than optimal storage of original signal, and are not easily recognized by a computer. With the use of LIN, digital communication to and from the actuators and sensors provide greater accuracy, the advanced ability to communicate with computers for signal acquisition and controller integration, and the ability to retain the original signal recorded for future playback. LIN will also save on initial wiring costs and time resulting in improved system functionality.

The design task will be to design and develop a rapid prototyping environment for climate control development. Two sensors including an Indicator Light Solar Sensor (ILSS) and a Relative Humidity and Temperature Sensor (RLS) connected to six LIN based actuators including Air Inlet, Air Delivery, Temperature Blend flaps, Blower Motor, Evaporator, and Heater Core will be analyzed using the automotive industries most sophisticated testing software dSPACE in order to complete the prototyping environment by the expected delivery date. With the help and full support from our sponsors at General Motors, the rapid prototyping climate control environment will help support General Motors HVAC controls and software verification groups.

2 - Background

2.1 - dSPACE

Technology is growing, and solutions for automotive engineering for efficiency, productivity and innovation are imperative. With the use of dSPACE this tool provides for developing, testing and calibrating ECU's in the automotive field. The application dSPACE is also used in various fields such as medical engineering industries and aerospace. As an independent system partner, dSPACE has done pioneering work on tools for ECU development, and cooperates closely with the automotive industry. The dSPACE application supports all the development phases with its products, from architecture-based system design and block-diagram-based function prototyping to electronic control unit auto-coding and Hardware-in-the-Loop (HIL) testing. The advantages for automotive customers are considerable savings in cost and time, greatly enhanced software quality, and more efficient cooperation between manufacturers and suppliers. With the works of General Motors the concepts of understanding the process of dSPACE is critical. The process of developing and testing electronic control units is based on five phases of what's known as the V-cycle. The dSPACE hardware and software cover four of these five phases, except the first phase which is control design [6].

2.2 - Phases

In rapid control prototyping, control algorithms are taken from a mathematical model and implemented as a real-time application so that the control strategies can be tested with the actual controlled system, such as a car or a robot. Simulink is used as the input and simulation tool, and Simulink Coder, also from MathWorks, is used as the code generator. The application dSPACE provides

the necessary hardware platform consisting of a processor and interfaces for sensors and actuators, plus the Simulink blocks needed to integrate the interfaces into the Simulink model Real-Time Interface (RTI).

2.3 - Hardware-in-the-Loop (HIL) Simulation

In HIL simulation, a simulator mimics the environment in which an ECU will function for examples are cars, airplanes, and robots. First the ECU's inputs and outputs are connected to the simulator's inputs and outputs. The simulator then executes a real-time model of the ECU's working environment, which can consist of Automotive Simulation Models (ASMs) from dSPACE. This method provides a way to test new functions reproducibly in a safe environment, before a prototype of the product has even been produced. The advantage of HIL simulation in comparison with ECU tests in real prototype vehicles is that the ECU tests can be performed very early during the development process. Errors are detected and eliminated very early and cost-efficiently.

Optimizing the control functions so that they fit specific applications is an integral part of ECU and controller development. To achieve this, the parameters of the ECUs are adjusted during ECU calibration. The dSPACE application modular system of software and hardware performs this final step in the development process [6].

2.4 - Local Interconnect Network (LIN)

To reduce the amount of wires that is handled by communications between systems, many automotive manufacturers have created different bus systems that are incompatible with each other. The reasoning for this effort is caused by a conjunction of wires that is needed to connect these components. The solution to this problem which is most commonly used in the automotive sector has been replacing their bus systems with intelligent systems. The new bus, which is called LIN, is a bus that was invented to be used in simple switching applications for example: car seats, door locks, sunroofs, rain sensors, mirrors and many other applications. The LIN bus is a sub-bus system based on a serial communications protocol. The bus is a single master/multiple slave bus that uses a single wire to transmit data which can be seen in figure 1 below. The bus detects defective nodes in the network, data checksum, parity check, safety and error detection. The benefit of using LIN is to reduce costs, and components can now be driven without crystal or ceramic resonators. Time synchronization permits the correct transmission and reception of data. The system is based on a UART/SCI hardware interface that is common to most microcontrollers. A LIN frame consists of a header and a response part. To initiate a communication with a slave the master sends the header part. If the master wants to send data to the slave it goes on sending the response part. If the master requests data from the slave the slave sends the response part [9].

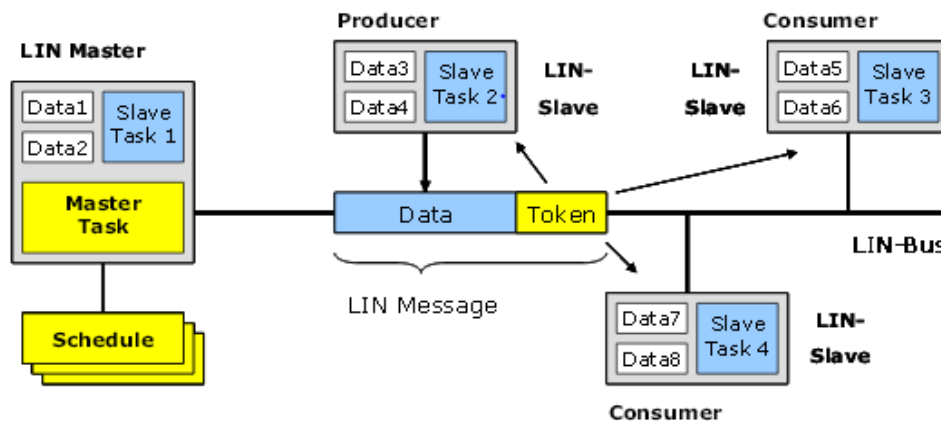


Figure 1 - Lin Network Overview

2.5 - Controller Area Network (CAN) vs. LIN

A mechanism used in the automotive industry to control the amount of wiring is called CAN. CAN is a multi-drop, multi-master serial bus that provides communications between controllers, sensors and actuators [5]. LIN, on the other hand, is a Master/Slave scheduled single wire bus [11].

Messages and data transmission for CAN and LIN differ in their characteristics. CAN's messages contain an ID that identifies the source or content and the messages are all broadcast messages. Each node is allowed to transmit messages at any time while each receiver decides to process or ignore the message. Data transmission is synchronous therefore CAN systems don't have a line for clock [5]. Conversely, LIN monitors data and checksum byte of messages received from slave and bus information. This is done to check for errors which serve as a reference with its clock base. The slave is one of the 2-16 members on the bus. It receives or transmits data when the master sends an appropriate ID. The slave is also able to detect and report ID parity error, byte field framing error (i.e. invalid stop bit), data error (i.e. data transmitted does not match data received, data transmitted is not received, fixed form data received is incorrect), and invalid checksum [11].

There are two state types on the CAN bus, recessive and dominant. With recessive CANH and CANL are not driven in this state while with dominant CANH and CANL are driven in this state. CAN is based on a seven layer OSI model. The data link and physical layer are the most important. The physical layer has no specifications, therefore the driver/receiver are open to the system. This layer uses wires and connectors as its transport medium. For the data link layer it is defined by the CAN specification in terms of filtering, error detection, signaling and coding. No noise can be induced on this layer or it will disrupt the signal [5]. Similarly, LIN also has a dominant and recessive state. The dominant signal produces logic '0' with low bus voltage (ground) while the recessive signal produces a logic '1' high bus voltage (battery) [11].

2.6 - Companies uses of LIN in HVAC systems and other systems

LIN Consortium Steering Committee includes a total of 85 companies including for example: BMW, Audi, Volkswagen, Daimler Chrysler, Volvo, etc. [11]. LIN 2.0 is used for communication between components in vehicles, and the modern automotive networks use a combination of LIN for low-cost applications primarily in body electronics which is shown in the table 1 below [12]. In BMW's HVAC system, when activated, the automatic air conditioning system channels fresh air from outside the vehicle, cools it, reduces its humidity and, depending on the temperature setting, warms it again. Air temperature, volume and quality can be precisely controlled, whether the system is operating in manual or automatic mode. In automatic mode the distribution of airflow and volume is controlled automatically, while the temperature is held steadily at your selected setting. This feature also works in a convertible. In convertible mode the air conditioning system is based on the external temperature, the solar intensity, and relative wind speed of the vehicle. The settings in the cabin are automatically adjusted to reflect these conditions [4].

Table 1 – LIN Applications

Application Segments	Specific LIN application examples
Roof	Sensor, light sensor, light control, sun roof
Steering Wheel	Cruise Control, wiper, turning light, climate control, radio
Seat	Seat position motors, occupant sensors, control panel
Engine	Sensors, small motors
Climate	Small motors, control panel
Door	Mirror, central ECU, mirror switch, window lift, seat control switch, door lock

LIN is also used in pick-up trucks. In the modern pick-up truck, there can be up to 20 ECUs controlling various functions such as ABS, gearbox, lighting etc. However, today many of the simpler

functions are connected directly to the controlling ECUs and the amount of wiring is therefore substantial. The introduction of a multiplexed network such as LIN would not only cut the wiring cost and weight considerably but also introduce new and more flexible solutions of connecting components. Investigations have been done concerning the technique behind LIN as well as the hardware and software resources needed in order to implement LIN-communication between components. A demonstrational implementation of the LIN-protocol was successfully carried out on the light control panel of a Volvo pick-up truck, which enlightens some of the benefits of using LIN [1].

Beginning in May of 2009, the SAE J2602 LIN was standardized to all GM vehicles and light duty trucks. This ruling contained GM specified vehicle subnets or LIN sub-system protocols. GM needed a low cost body data link to replace low speed dual wire CAN links [11].

3 - Objectives and Design Specifications

3.1 - Mission Statement

The MSU design team's goal is to provide functional abstract software prototyping solutions that can be used by our customer to integrate in their climate control development system.

3.2 - List of Objectives

- Investigate and implement a general LIN handler using dSPACE system.
- Design and interface LIN handler to device protocol and functional messaging.
- Wire and demonstrate LIN device and sensor functionality using dSPACE system.
- Allow the customer to modify our driver design for reusability.
- Document our design strategies and interfacing solutions.

3.3 - Design Specifications

A graphical model with the use of MATLAB/Simulink is necessary to create a model base that is a unique control design in efforts to save cost and time. A virtual model based development would enable to see which individual components meet the specifications and which do not. The controlled design will then be integrated into dSPACE for prototyping. Once prototyping is complete, our conceptual design could then be constructed and tested.

3.3.1 - Low Cost:

- A virtual design and using LIN actuators will reduce cost, and allow low-end multiplexed communication networks to interconnect. This will now allow components to be driven without crystal or ceramic resonators receiving data from the sensor connections.

3.3.2 - Specifications using the latest versions of the following software:

- LIN - replacing the analogue driving electronics and also develops new ways of controlling pressure sensors in control systems
- dSPACE – allow us to optimize the control designs for the real ECU as often as needed until our requirements are met, all without any manual programming. Our block diagram, and virtual design would be automatically implemented on the system and calculated in real time.
- MATLAB – is a high level language program that is used for numerical computation, visualization of models/applications, and programming. In our project MATLAB will be used for model visualization.
- SIMULINK - block diagram environment for model-based designs and multi-domain simulations will be needed to support simulation and analytical development.
- RTW (Real Time Workshop) – for use with SIMULINK enhances embedded codes to build programs that could stand-alone simulations on our external computer.
- CDD_Standard (Control Desk) – software used for ECU development, measurement, and diagnostics.

- RTI(MABX blockset) – set ups bypass applications in the modeling environment. It allows selecting model inputs and outputs flexibly without ECU code modification.
- RTICAN(RTI CAN blockset) – extension for Real-Time Interface for developing and testing control functions through the use of the CAN bus system.
- MCCPPC(compiler for MABX) – generates executable object code for PC processors.
- RTILINM(RTI LIN Multi-Message blockset) – extension for Real-Time interface for handling LIN setups, controlling, and configuring LIN frames from one single SIMULINK

3.3.3 - Performance:

- Capabilities must be at least 90 percent efficient
- Ability to correct a real-time issue

3.3.4 - Delivery Date:

- April 25th, 2014 is the official design day

3.3.5 - Quantity:

- One design that demonstrates a dSPACE set-up that reads relative humidity, sensor temperature and glass temperature from sensors that is being driven through multiple LIN actuators with associated required diagnostics and actuator learning ability
- Six programmed LIN actuators are expected in our design, with a detailed description of how our design operates

3.3.6 - Environmental Conditions:

- Electro Magnetic Compatibility (EMC)
- Test edge cases

3.3.7 - Safety:

- All wiring harnesses must be secured and covered properly
- LIN actuators must be enclosed
- LIN actuators must be installed properly

3.3.8 - Reliability:

- Must serve the typical lifespan of modern day vehicles

3.3.9 - Electrical Loading:

- 13 V DC power supply for dSPACE Micro Auto Box
- 10 mA maximum at 12 V DC for ILSS and RHS sensors

3.3.10 - ILSS and RHS Sensor Response:

- 20 millisecond (ms) minimum response time to LIN Master for ILSS and RHS sensor shown in figures 2 and 3.



Figure 2 - ILSS Sensor



Figure 3 - RHS Sensor

3.3.11 - Maintenance:

- Must be easily maintainable and upgradeable so that future customers will be able to upgrade as the years go by.
- Provide a detail schematic of our software designs

3.3.12 - Connectors:

- 8 connectors.
 - One for each of the two sensors
 - One for each of the six actuators

3.3.13 - Wire Thickness (Gauge):

- 6-8 foot pigtail with 22 gauge wire

3.3.14 - Packaging:

- Test bench with integrated actuators and sensors

3.3.15 - Operating Instructions:

- Operating should be straightforward with included manual and actuator diagrams

3.3.16 - Initial Prototyping Cost:

- Initial Prototyping must be under \$500 budget given by MSU ECE department

4 - Fast Diagram

The Function Analysis System Technique (FAST) Diagram below is a way to show all of the functions that the motor controller system will implement. This diagram moves from left to right dealing with the main function first and transitioning to the primary function and finally the secondary functions following the primary. The main function of this system is model Simulink controller. The next columns are the different primary functions that are needed for the primary function to work. The next subsequent columns are secondary functions that describe how the primary or secondary functions will be achieved. This FAST diagram is an easy way to visualize the basic functions of the system and how they rely on one another and can be seen in figure 4 below.

Team 4 - Fast Diagram

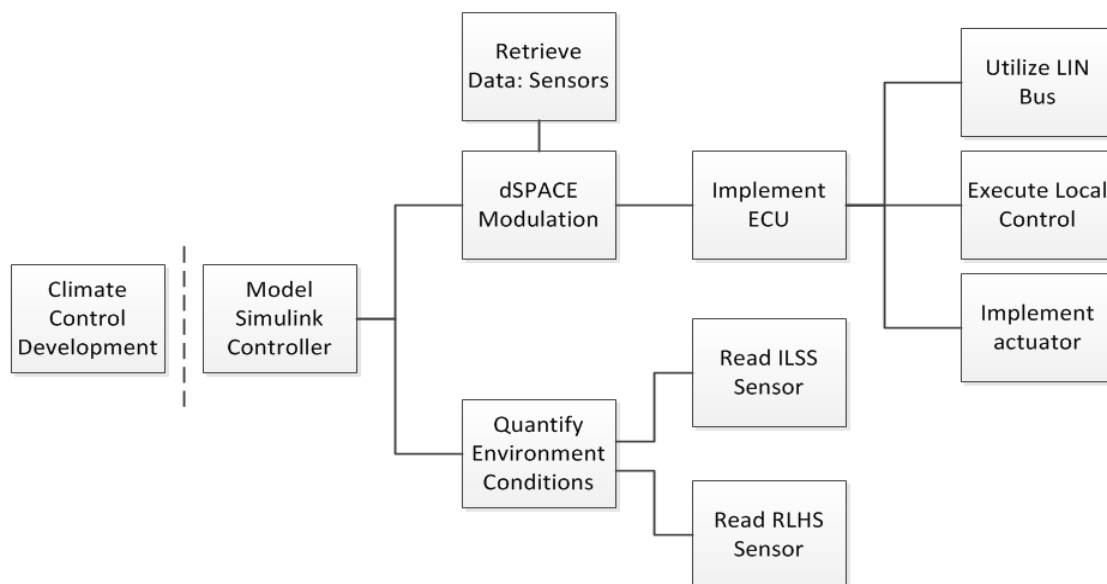


Figure 4 - FAST Diagram of Proposed Climate Control Development

5 - Conceptual Design Descriptions

In order to control the functionality of the actuators and sensors, a controller must be developed to achieve the desired outcome. The parameters of Proportional-Integral (PI), Proportional-Derivative (PD), and Proportional-Integral-Derivative (PID) controllers are compared to see which controller will give the optimal results.

5.1 - Open Loop Controller

Achieve the cheapest and quickest solution where the RCP is set to a certain value and the system strives to reach the setting. No feedback is transmitted to the master therefore sensors are not needed for this design which brings a lower cost in the overall construction of the climate control prototype.

5.2 - PI Closed Loop Controller

Achieve the most accurate temperature within one degree of user input through the precise control of the actuators. PI controllers use integral action, through constant summing of its tuning parameters, to gather feedback from the sensors based on the duration of the measured process to eliminate offsets in the controller [2].

5.3 - PD Closed Loop Control

Minimize the system response time. PD controllers, due to having a quick rise and transient response time [2], will be the best controller to match the high speed feedback from the sensors to the LIN master.

5.4 - PID Closed Loop Control

Combine the benefits of the PI and PD controllers which will maximize the speed and accuracy of the feedback signal from the actuators and sensors to the LIN master.

6 - Ranking of Conceptual Design Specifications

In order to see the best design solutions, the prior four designs were compared using a feasibility matrix which is shown in table 2 below. This matrix has three important details to each of the designs which are cost feasibility, implementation complexity, and lead time. Due to the fact that the controllers do not cost anything to program, the cost section will correspond to the possible integration of the open loop or closed loop feedback components, such as sensors and actuators, into a vehicle. The four designs are ranked on a scale from 1-5 in each category with a rating of 5 representing a great feasibility option and a rating of 1 representing an infeasible option.

Table 2 - Prototype Conceptual Design Stages

Design #	Description	Cost Feasibility (5-great 1-poor)	Implementation Complexity (5-simple 1-difficult)	Lead Time (5-best 1-worst)	Average Feasibility Rank
1	Open Loop Controller (No Feedback from Sensors)	5	5	5	5
2	Closed Loop PI Controller	3	3	3	3
3	Closed Loop PD Controller	3	3	3	3
4	Closed Loop PID Controller	3	2	2	2.33

7 - Proposed Design Solutions

To achieve all the features of a functional climate control system, two systems will be considered: open loop and closed loop feedback. The open loop system is a cheap and easy control system to implement but, because there is no feedback, the ECU will not observe the output of the actuators therefore corrections cannot be made. The next solution would be a closed loop feedback control system where the ILSS sensor and RHLS sensors would provide feedback to the ECU and apply adjustments to the controller inputs to regulate the actuators. The closed loop feedback controller is what will be evaluated as the optimal design and is shown in figure 5 below.

A Simulink modeled controller must be designed on the basis of response time and precision of climate modulation which is achieved through regulation of the actuators by the feedback from the sensors. The ILSS and RHS sensors should provide the ECU with a 20 ms minimum continuous average filtered overhead measurement [7]. In order to meet the fast response of these sensors, PI, PD, and PID controllers will be tested and compared to see which controller provides the quickest and accurate result with the least amount of steady state oscillations and other controller errors. Once further testing is conducted regarding the response time of the actuators, a unique controller will be constructed for each actuator that provides the greatest performance when integrated into the climate control prototyping environment.

The output of the ECU will be a unique signal to drive each of the six actuators. These signals will be transmitted using LIN bus protocol. The LIN protocol will act as a second feedback to the ECU to make adjustment to the controller if errors occur in actuator signal reception. This can be seen in figure 5. The speed of LIN signal transmission varies between 1kbit/s - 20kbit/s depending on temperature [7, 11]. Table 3 shows test edge cases of the ILSS sensor and records the LIN count. This number can be converted to a speed of 50ms - 1s transmission time which is the minimum and maximum speeds that the LIN protocol can transmit data to the LIN slaves. The measurements below in table 3 are recorded by the ILSS sensor [7]. The LIN based sensors are able to transmit data faster than the LIN bus is capable of delivering the message. For this reason, the controller implemented to drive actuators will require the fastest possible execution in order to refrain from compounding the delay of the transmitted signals.

Table 3 - ILSS Sensor Test Cases for LIN Variant

Actual temperature (°C)	Count (N)	Engineering (E) Temperature Value
-40°	0	-40°
0°	80	0°
25°	130	25°
85°	200	85°
87.5°	255	87.5°

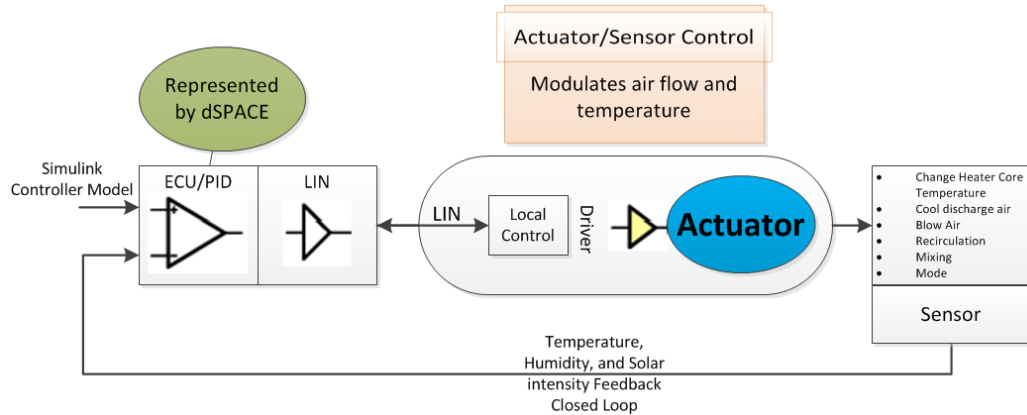


Figure 5 - Climate Control Logical Flow Diagram

8 - Risk Analysis

8.1 - Sensors Disabled

Sensors are the first process in climate transmission. Once the sensors are not functional, such as a damaged or a short in the circuit, any input that could be used from the controller would be lost. For example, if the customer adjusted the climate control settings, the direct result would be non-responsive. Another example could be, when the customer selects "AUTO" scale on their temperature control panel, the automobile's air conditioning system should automatically respond to the environment's temperature. This feature usually turns the compressor on and off depending on whether the temperature reaches the appose conditions. If the RLH sensor is defective, the controller would not be able to provide the signal to transmit the temperature to the desired value. The final result could cause undesirable temperatures in the vehicle.

8.2 - Sensor Error or Dysfunction

Accuracy from the sensors could also be effected by software crashes and faults in components. These factors can lead the sensors to send wrong information to the controller. As a result, inaccurate responses from the system can occur. One example can be from the solar intensity sensor, which measures the solar radiation/intensity inside the vehicle. If inadequate data is sent from the sensors, the controller would misestimate the amount of heat energy radiating on the outside of the vehicle, and that could cause components to burn. Finally, the sensor error effects the decision making of the controller on actuator behaviors.

8.3 - Actuators Disable and Errors

Similar as to what was mentioned before, problems with actuators could cause inaccurate results. For example, when the customer is using the air conditioning system, the air inlet and delivery actuators control the air circulation internally and externally within the vehicle. If the air inlet and delivery actuators extract air particles when the customer switches from internal to external , it would be a

problem delivering fresh air. Furthermore, problems with the temperature blend flaps, blower motor, evaporator and heat core would lead to errors in air circulation.

8.4 - Controller Errors and Software Error

In perspective, controller and software are core parts in the process of dealing with data transmission. Any faults can cause problems similar to what was stated previously. Results would include error in coding, undesired temperature settings, and defaults in components on the integrated circuit. Risk analysis allows for qualitative control and protection against faults in the system. The system should be able to respond to any unexpected situations. In accordance to risk analysis, to ensure quality, defensive mechanisms should be enforced to maintain product functionality and safety.

9 - Project Management Plan

9.1 - Non-Technical Roles

Table 4 - Team Non-technical Roles

Name	Non-Technical Role
Guiseppe Ferro	Management
Omar Ali	Web Design
Weiham Yan	Web Design
Sitong Ge	Documentation
Ricardo Johnson	Presentation
Yingshuyu Li	Lab Coordinator

9.2 - Technical Roles

Table 5 - Technical Roles

Name	Technical Role
Guiseppe Ferro	ILSS Sensor Control and Programming
Omar Ali	RLS Sensor Control
Weiham Yan	Actuator Interfacing
Sitong Ge	Controller Design and Programming
Ricardo Johnson	LIN Integration
Yingshuyu Li	dSAPCE Integration

9.3 - Gantt Chart

The Gantt chart is a way for the group to stay organized and have an overview of the different tasks needed to complete this project in the time allotted. Naming the individual tasks, giving each one a deadline, and putting it under different categories will keep us organized. The Gantt chart will allow us to be able to see what tasks need to be done and monitor the progress in a visual way. The Gantt chart will start from the research aspect of the project to the finalized product which will be displayed on design day.

Table 6 - Gantt Chart

Task Name	Duration	Start	Finish
Project Definition and Objectives	16 days	Fri 1/10/2014	Fri 1/31/2014
First Meeting with Team 4	1 days	Mon 1/14/14	Tue 1/14/14
First Contact with GM Sponsor	1 days	Wed 1/15/14	Thu 1/15/14

Research dSPACE/Lin Operating System	3 day	Fri 1/15/14	Fri 1/17/14
Meet with GM Sponsors via Teleconference	7 days	Mon 1/20/14	Tue 1/28/14
Acquire multiple licenses	7 days	Mon 1/20/14	Tue 1/28/14
Review meeting notes to discuss with facilitator	2 days	Tue 1/21/14	Wed 1/22/14
First Meeting with Facilitator	8 days	Wed 1/22/14	Fri 1/31/14
Start Pre-Proposal and Gantt Chart	7 days	Thu 1/23/14	Fri 1/31/14
Divide sections to equalize writing load	7 days	Thu 1/23/14	Fri 1/31/14
Research on past projects that are similar	5 days	Thu 1/23/14	Wed 1/29/14
Find out if any like systems are available	5 days	Thu 1/23/14	Wed 1/29/14
Preliminary research on LIN/dSPACE found	5 days	Thu 1/23/14	Wed 1/29/14
Overview of the problem	5 days	Thu 1/23/14	Wed 1/29/14
Executive Summary	5 days	Thu 1/23/14	Wed 1/29/14
Risk Analysis/Objective and Design Specs	5 days	Thu 1/23/14	Wed 1/29/14
Conceptual Design Development	5 days	Thu 1/23/14	Wed 1/29/14
Start web page design with GM logo	5 days	Thu 1/23/14	Wed 1/29/14
Find out how to display for Design Day	5 days	Thu 1/23/14	Wed 1/29/14
Second meeting with GM and deliverables	5 days	Tue 1/28/14	Sat 2/1/14
Start creating a logic flow/diagram for LIN driver	4 days	Tue 1/28/14	Fri 1/31/14
Determine systems input/output parameters	4 days	Tue 1/28/14	Fri 1/31/14
Second meeting with Facilitator	1 day	Wed 1/29/14	Wed 1/29/14
Pre-proposal Due	0 days	Mon 2/3/14	Mon 2/3/14
Acquire Material to Begin Design	5 days	Mon 2/3/14	Fri 2/7/14
Pre-trial software/understanding/learnings	5 days	Mon 2/3/14	Fri 2/7/14
Pre-trial actuators/understanding/learnings	5 days	Mon 2/3/14	Fri 2/7/14
Begin Programming/Virtual Design	9 days	Mon 2/10/14	Thu 2/20/14
Program Simulink to dSPACE	2 days	Mon 2/10/14	Wed 2/12/14

Utilize LIN to actuators and sensor	9 days	Mon 2/10/14	Thu 2/20/14
Connect entire system	9 days	Mon 2/10/14	Thu 2/20/14
Begin testing parameters of virtual design	9 days	Mon 2/10/14	Thu 2/20/14
Begin Building Prototype for Display	20 days	Mon 3/1/4	Thu 3/20/14
Brainstorm Ideas	20 days	Mon 3/1/4	Thu 3/20/14
Start prototyping	20 days	Mon 3/1/4	Thu 3/20/14
Testing prototypes	20 days	Mon 3/1/4	Thu 3/20/14
Determine test edge cases	20 days	Mon 3/1/4	Thu 3/20/14
Collaborate on errors for final design	28 days	Mon 3/1/4	Thu 3/28/14
Final display prototype design	15 days	Tue 4/1/14	Tue 4/15/14
Repair/Finalize Design Specs	15 days	Tue 4/1/14	Tue 4/15/14
Due Dates	86 days	Fri 1/10/14	Tue 4/15/14
Proposal Presentation	1 day	Fri 2/14/14	Fri 2/14/14
Final Proposal Due	1 day	Mon 2/21/14	Mon 2/21/14
Site Visit to GM for Testing	1 day	Mon 2/24/14	Mon 2/24/14
Team Technical Lectures	1 day	Mon 3/24/14	Mon 3/24/14
Design Issues Paper	1 day	Fri 4/11/14	Fri 4/11/14
Professional Self-Assessment Paper	4 days	Wed 4/16/14	Wed 4/16/14
Final Report Due	1 day	Wed 4/23/14	Wed 4/23/14
Design Day	1 day	Fri 4/25/14	Fri 4/25/14

10 - Budget

The Budget for this project is set to five hundred dollars. Because we are being supplied the licenses, actuators, sensors, and LIN bus from our sponsor, we currently do not require any of the capital allotted to us for this design project.

11 – References

- [1] Anders Rylander & Erik Wallin. "LIN - Local Interconnect Network - for use as sub-bus in Volvo trucks". 2003. Available online: <http://www.mrtc.mdh.se/rtis/contrib/Exjobb/03.PDF>
- [2] Anti-Windup Control Using a PID Controller. February 19, 2014 Available online: <http://www.ni.com/white-paper/9733/en/>
- [3] Automotive HVAC Market by Vehicle Type (Passenger Cars, LCVs & HCVs), Technology (Manual & Automatic), Components & Cabin Comfort Market by Type (Power Seats, Power Windows, Heated Seats & Sunroofs) - Global Trends & Forecast to 2018. February 01, 2014. Available online: <http://www.marketresearch.com/MarketsandMarkets-v3719/Automotive-HVAC-Vehicle-Type-Passenger-7750028/>
- [4] BMW AG: BMW Insights: Technology Guide: Automatic air conditioning. February 01, 2014. Available online: http://www.bmw.com/com/en/insights/technology/technology_guide/articles/automatic_air_conditioning.html
- [5] "Controller Area Network." Michigan State University. ECE 480 Team 5 Senior Design, 01 Nov. 2013. Web. 27 Jan. 2014. Available online: <http://www.egr.msu.edu/classes/ece480/capstone/fall13/group05/docs.html>
- [6] *dSPACE Application fields in the automotive industry available online 2014* <https://www.dspace.com/en/ltc/home/applicationfields/automotive.cfm>
- [7] Gilles Delorme. "Solar/Ambient Light/Temperature Sensor With Indicators." General Motors. 15 Jan. 2013
- [8] K.J. Astrom and B. Wittenmark, *Computer-Controlled Systems, 3rd Ed*, Prentice Hall, 199
- [9] "Local Interconnect Network." *Wikipedia*. Wikimedia Foundation, 29 Jan. 2014. Web. 01 Feb. 2014. Available online: http://en.wikipedia.org/wiki/Local_Interconnect_Network
- [10] M. Honek, J. Csamb'ál, S. Wojnar, M. Kopačka, P. Simončí, and M. Lauko. "RAPID CONTROL PROTOTYPING SYSTEM DSpace USED FOR CONTROL OF COMBUSTION ENGINE PROCESSES." *Slovak University of Technology*. N.p., n.d. PDF Available Online: http://dsp.vscht.cz/konference_matlab/MATLAB10/full_text/039_Honek.pdf
- [11] Natalie A. Wienckowski, Merv Rose Jr. "J2602/LIN 2.0 Seminar CTIS # 28302." General Motors. 19 May, 2010
- [12] National Instruments. "Introduction to the Local Interconnect Network (LIN) Bus". February 01, 2014 Available online: <http://www.ni.com/white-paper/9733/en/>