LART-CS08

Preliminary Design Report

Senior Design - Spring 2008

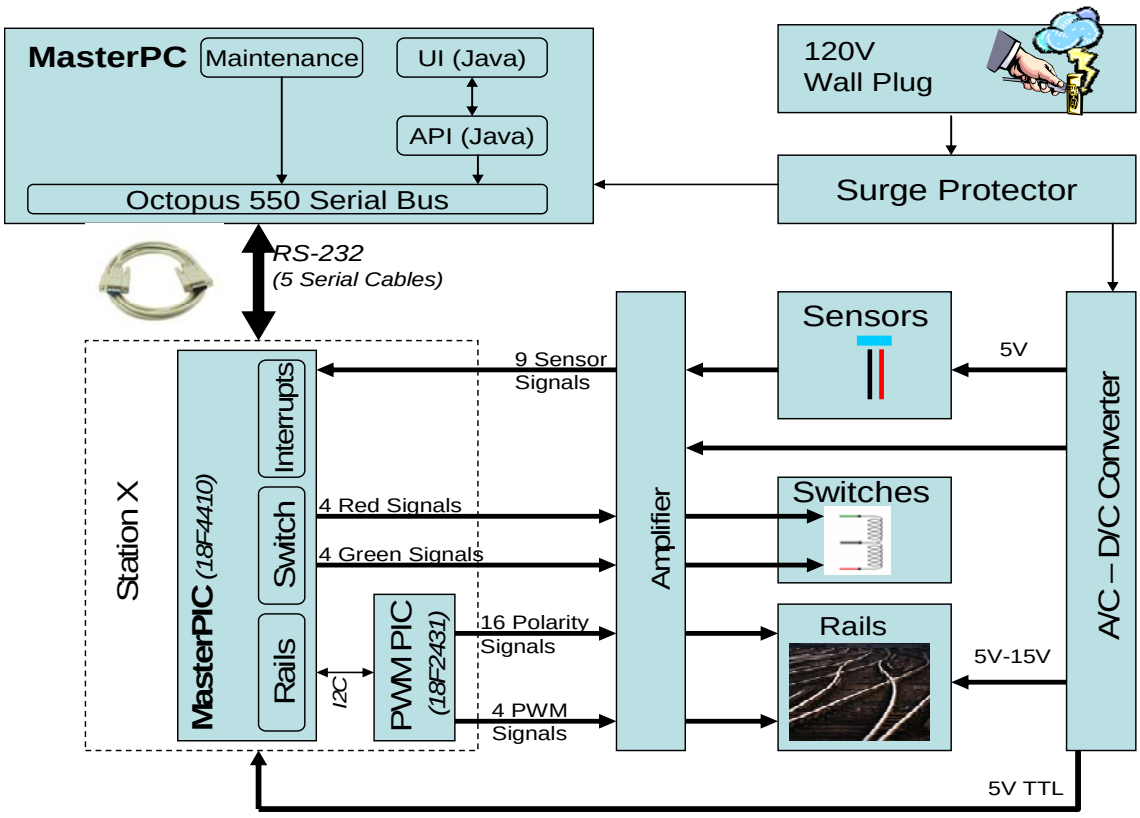
Lafayette College

2/11/2008

PRELIMINARY DESIGN REPORT

OVERALL SYSTEM DESIGN

1.1 Overall System Block Diagram



The MasterPC shall communicate to each station via RS-232 standards, using the packet structures defined in latter sections. To avoid collisions of using a single wire as the ether, the communication shall be done in parallel via the Octopus 550. This board will enable the MasterPC to have 8 serial ports, all of which can be connected to other stations.

The information coming from the MasterPC shall be analyzed via the PICs at each station. These PICs shall not only control switching, but also send the rail & speed information to a secondary PIC specialized in generating unique PWM signals. Furthermore, whenever a sensor is triggered, it shall generate an interrupt, which will send a packet to the MasterPC to notify it of the triggered sensor.

The power shall be supplied from a 120V wall plug, which feeds into a surge protector. The output of this feeds both the MasterPC and the A/C-D/C converter, which will be used to supply the rest of the system with the desired power levels.

1.2 Overall Requirements Analysis

	Requirements	UI Control	Networking	Low Level
R001	rail switch control	✓	✓	✓
R002	engine power control	✓	✓	✓
R003	train proximity monitoring	✓	✓	✓
R004	expandability and adaptability	✓	✓	✓
R005	control and monitoring speed	✓	✓	✓
R006	applications programming interface	✓		
R007	maintenance user interface	✓		
R008	demonstration application	✓		
R009	modifications of the CFE layout			✓
R010	power input	✓		✓

	Requirements	UI Control	Networking	Low Level
GPR001	documentation	✓	✓	✓
GPR002	environmental	✓	✓	✓
GPR003	EMI/EMC		✓	✓
GPR004	hazmats	✓	✓	✓
GPR005	safety and good practice	✓	✓	✓
GPR006	reliability	✓	✓	✓
GPR007	maintainability	✓	✓	✓
GPR008	sourcing sustainability	✓	✓	✓
GPR009	global sustainability	✓	✓	✓
GPR010	ethics report	✓	✓	✓
GPR011	project demonstration	✓	✓	✓
GPR012	final disposal of projects	✓	✓	✓

	Requirements	UI Control	Networking	Low Level
ER001	trains will not crash	✓	✓	✓
ER002	train acceleration	✓	✓	✓

A more detailed discussion of the requirements will follow in requirements analysis for each group.

1.3 Overall Risk Assessment

With many interconnected parts, the failure to complete a subsystem of this project could drastically alter the design and cause setbacks. The networking subsystems that will be implemented on the PIC controller are very crucial to the whole system. If the RS-232 connection fails to work between the PIC and PC we will have to redesign with new hardware that could affect many aspects of the low level hardware and API software. We are familiar with serial communications and some group members have programming and testing experience with the PIC. The simplified communication of RS-232 seemed a more appropriate goal than standard Ethernet. Another risky aspect of this networking is the conversion of digital data from the PC to analog signals for low level hardware. A failure to get useful signals out on parallel wires for hardware usage will cause a redesign with different hardware.

The low level hardware is less abstract than programming on the PIC controllers. There is more than one solution to most designs that can be implemented if the original fails. It is risky because a subsystem failure will cause major schedule problems while waiting for new hardware in the mail and could cause the system to go over budget.

Less risky is the software on the PC. There is more experience with software of this kind than there is with the hardware. The software can be rewritten or changed on the fly and recompiled very quickly. Any problems can be fixed much faster with a smaller impact on the overall system. The UI is perhaps the least risky subsystem in the software because it is on the very end of the entire system and is mostly cosmetic. The API is a risky because there is not a lot of experience with designing software in this format. It must be well documented and if it cannot communicate through RS-232 correctly we will have similar networking issues described above with the PIC.

Subsystems	Calculated Risk Factor	Risk Level
UI/API Group: User Interface	0.199	Low
UI/API Group: Auto/Manual Control	0.531	High
UI/API Group: Packet builder	0.5203	High
UI/API Group: Packet decoder	0.5736	High
Networking Group: RS-232Communication	0.53	High
Networking Group: Sensor interrupts	0.7733	High
Networking Group: Rail control	0.7834	High
Networking Group: Switch control	0.7834	High
Low-Level Group: Overall system power	0.4	Medium
Low-Level Group: Rail power	0.467	Medium
Low-Level Group: Switching	0.377	Medium
Low-Level Group: Sensors	0.490	Medium

1.4 Overall Budget Analysis

Direct Costs:

Octopus-550, PICs, PCBs, logic chips.

Total direct costs: $288.70

The ECE department has allotted $4000 for the project. Our current estimate is less than 10% of this figure, so any cost overrun we experience should be manageable.

Indirect Costs:

Labor (2250 Hours), computers (8), wires, resistors, capacitors.

Total indirect costs: $34,505.00

Management has organized the project into three different parts. We have estimated that each of these three parts will require a total of 250 hours of technical work. Additionally, we expect each person to spend a total of 100 hours during the semester on documentation and meetings, which adds another 1500 labor hours to the project. At $10.00 per hour for each person, labor alone will cost $22,500. We will also be using 8 computers for the project, at an estimated cost of $1500.00 for each computer. Finally, we have added $5.00 for components such as wire, resistors, and capacitors.

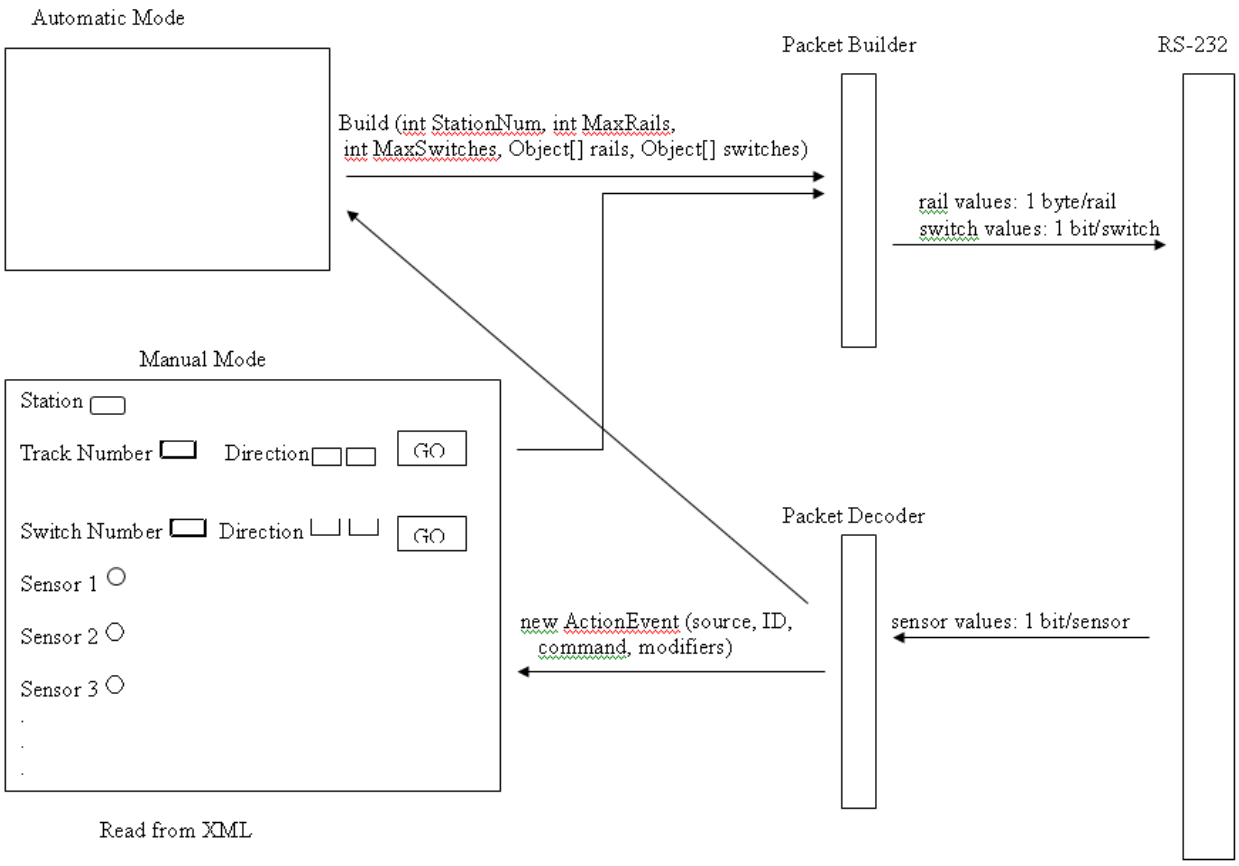
Total costs for project: $34,793.70

The total cost for the project is just under $35,000, with the vast majority of this money coming from indirect costs. The total cost to the ECE department is estimated to be less than $300.00. Even if the direct costs were to double, the project would still cost far less than the $4,000.00 that the ECE department is able to devote to the project.

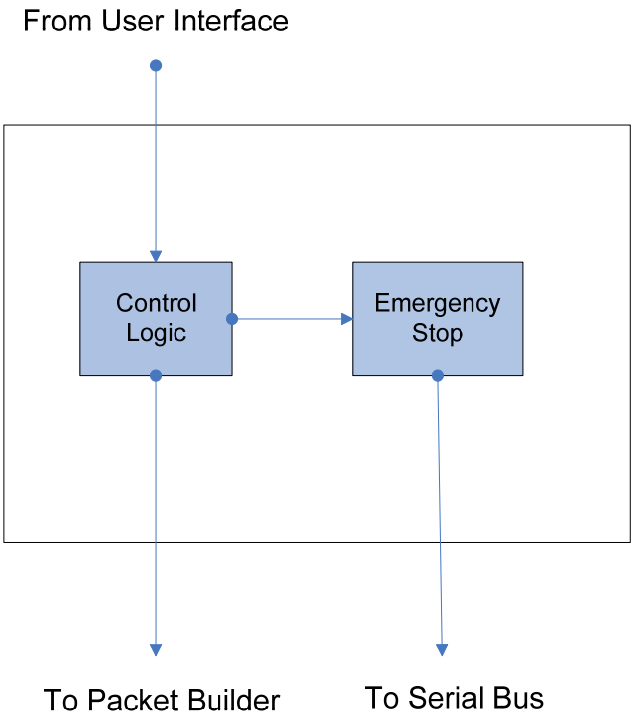
USER INTERFACE / CONTROLLER GROUP

2.1 UI/Controller Sub-System Specifications

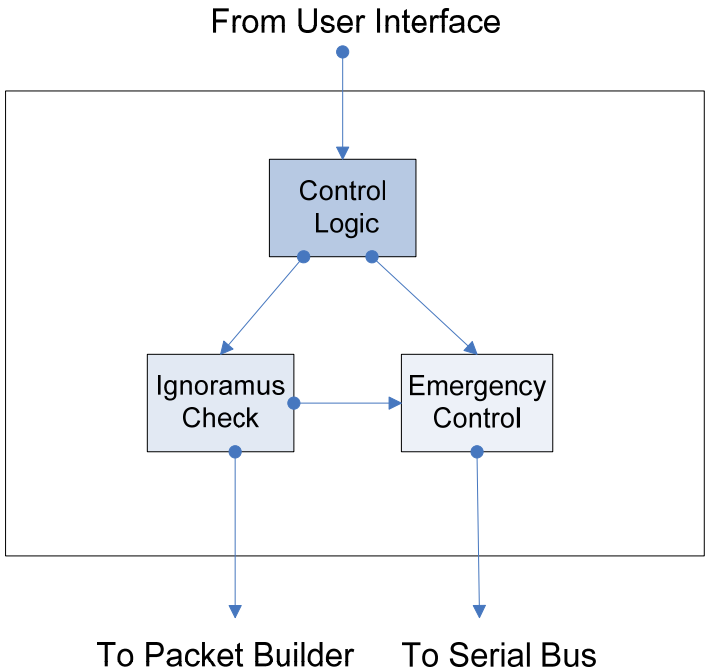
BLOCK DIAGRAMS



This is the block diagram of the software portion of the train controller.



Automatic control diagram – the control logic communicates directly with the packet builder.



Manual control diagram – the control logic comes from the user interface and goes through a check before going to the packet builder.

DESIGN DESCRIPTION

The GUI and all the coding will be programmed in JAVA and Swing.

USER INTERFACE:

This is basically the GUI that a train operator would use to monitor the trains and to control their movements and speed. It will also include an emergency stop button that the operator could press if there is potential for disaster. The emergency stop button would have a packet assigned to it so it would bypass the Packet Builder module and go straight to the power signals. The GUI will also include sliders for each station rail so that we can control the direction and the speed at the same time for each train. Also we will have options for the train to be stopped, go straight or go in reverse. For the automatic control, the coding will be done in the Automatic Control section, and all that would be needed for the GUI to do is to tell the program how many trains will be running. The GUI also has diagrams for each station so as that when we get the sensor signals we can show where the train is by using animation. See attached.

AUTOMATIC CONTROL:

In this there will be control logic which will choose which stations the trains will be running, what speed they will be running at, and the direction they will be moving. The logic will also have a failsafe, which would detect an error if for example, trains are going too fast, and then the whole system will pause. The operator can also at his discretion stop the whole system. This also sends the sensor signals to the GUI.

MANUAL CONTROL:

In this there will be control logic which will send the required signals to the serial bus. The logic will also have a failsafe, which would detect an error if for example, the operator assigns contradictory directions, and then the whole system will stop. The operator can also at his discretion stop the whole system. This also sends the sensor signals to the GUI.

PACKET BUILDER:

In this, it would take the logic from the control blocks and then convert them to serial data in the form of a destination address, source address, data needed for movement and speed of train, and error code. This will then send it to the serial bus.

PACKET DECODER:

In this, it would take the logic from the serial bus and then convert them so the logic can read it. The signals that will be taken would be from the sensor so as we can display where the train is. This will then send it to either automatic or manual control.

2.2 UI/Controller User Interface Test Plan

PACKET BUILDER

Test Requirement: Receive information from the user about train speed, direction and power, and generate a byte sequence for RS-232 transmission.

Testplan:

a. Receive information from the user about the train speed, direction and power, and compare the byte sequence with the expected byte sequence.

b. Input a specific zone power off, and check to see if it powers off all the power controls in the specified station by assigning the relevant value to the individual bits in the byte sequence.

c. Input an emergency stop from the user, and check to see if it powers off all the zones, and compare the RS-232 byte sequence to the expected value.

d. Test via hardware by checking the bits coming out of the RS-232 port to see if it matches the expected values.

PACKET DECODER

Test Requirement 1: Verify the correct decoding of the RS-232 byte sequence with the sensor information of a particular train.

Testplan:

a. Input a byte sequence to power on a specific sensor and verify that the flag to turn the specific sensor on is a logic high.

b. Test via hardware by checking the bits coming out of the RS-232 port when a train reaches a particular sensor and verify the correct assignment of bit values in the byte stream.

Test Requirement 2: For automatic control, verify that the RS-233 byte sequence assigns the proper bit values when the trains are not in their initial positions for a specific schedule.

Testplan:

a. After receiving information from the user about a specific automatic schedule, check to see the received byte sequence from the RS-232 port. If one or more trains are not in their required initial starting positions, check to see that a dialog box pops up reminding the user to place the trains at the specific initial positions for the specific schedule.

MANUAL CONTROL

Test Requirement 1: Verify that all the manual controls light left or right movement, speed, direction, zone power and emergency stops are implemented correctly.

Testplan:

a. Allow the user the enter the desired manual control settings and verify the packet output.

Test Requirement 2: Verify the fail safe unit to prevent erroneous combinations of inputs.

Testplan:

a. Allow the user to enter a combination of inputs which might cause train collisions and check to see that the byte stream assigns the required bit value assignments to perform an emergency stop.
b. Also, verify that a dialog box pops up stating that the input was an erroneous combination.

AUTOMATIC CONTROL

Test Requirement 1: Verify the initial positions of the specific trains for specific schedules.

Testplan:

a. Assign the trains in the wrong initial positions for a specific schedule, and check to see that dialog box pops up telling the user that the trains need to be in specific locations for that specific schedule.

Test Requirement 2: Verify that a train breakdown message is shown in a dialog box when a train fails to reach a specific sensor at the expected time.

Testplan:

a. Intentionally pick up a moving train and check to see that the dialog box is displayed.

2.3 UI/Controller Requirements Analysis

	Requirements	UI / Manual Control	Automatic Control / Failsafe	Packet Builder	Packet Decoder
R001	Rail Switch Control	✓	✓	✓	
R002	Engine Power Control	✓	✓	✓	
R003	Train Proximity Monitoring	✓	✓		✓
R004	Expandability and Adaptability	✓	✓	✓	✓
R005	Control and Monitoring Speed	✓	✓	✓	✓
R006	Applications Programming Interface	✓	✓	✓	✓
R007	Maintenance User Interface	✓		✓	✓
R008	Demonstration Application		✓		
R009	Modifications of the CFE Layout				
R010	Power Input	✓	✓	✓	✓

R001 Rail Switch Control:
The manual and automatic control will use the packet builder to send to the hardware to specify positions for rail switches.

R002 Engine Power Control:
The manual and automatic control will use the packet builder to send signals telling the hardware to control the engine power.

R003 Train Proximity Monitoring:
The hardware will communicate train locations with the computer through the packet decoder. The UI will display locations on the screen and if the system is in automatic mode the controller will monitor their positions.

R004 Expandability and Adaptability:
All four sections are running on a PC. If anything in the system needs to be expanded the software will be able to handle it with minimal redesign.

R005 Control and Monitoring Speed:
The PC and hardware will communicate very quickly compared to the train speeds so all control and monitoring will be accurate.

R006 Applications Programming Interface:
We will write a document allowing a programmer to interface with all parts of the system.

R007 Maintenance User Interface:
Our user interface will allow a user to control every aspect of the system.

R008 Demonstration Application:
Our automatic control will allow anybody to demonstrate the function of the system.

R009 Modifications of the CFE Layout:
All four sections are running on a PC and will not need any modification to the overall layout.

R010 Power Input:
The computer can run from a standard wall outlet.

2.4 UI/Controller Risk Assessment

FUNCTIONAL DESCRIPTION

The function of the software is to control the system. It includes a user interface, an auto/manual control and failsafe module, a packet builder module, and a packet decoder module.

User Interface

This interface will be designed mostly for the user to interact with the system. It will be able to control all signals (speed and direction) that it will send out to the stations.

Methodology Maturity	Methodology Complexity	Software Maturity	Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.1	0.1	0	0

Potential for Failure Factor = (0.1 + 0.1 + 0.1 + 0.1 + 0 + 0)/6 = 0.067

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0.3	0

Consequence for Failure Factor = (0.9 + 0.3 + 0)/3 = 0.4

Risk Factor**:** P+C – (P*C) = 0.067 + .04 – (0.067 * 0.4) = 0.199 low risk

Auto/Manual Control and Failsafe

This module monitors the automatic and manual operations to make sure the correct signals going out to the stations are consistent with the mode the system is operating in (manual or automatic).

Methodology Maturity	Methodology Complexity	Software Maturity	Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.5	0.7	0.1	0.3	0.1	0.1

Potential for Failure Factor = (0.5 + 0.7 + 0.1 + 0.3 + 0.1 + 0.1)/6 = 0.3

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.7	0.3	0

Consequence for Failure Factor = (0.7 + 0.3 + 0)/3 = 0.33

Risk Factor: P+C – (P*C) = 0.3 + 0.33 – (0.3 * 0.33) = 0.531 high risk

Packet Builder

The packet builder module assembles the bytes that will be sent out to the stations and makes sure they are correct.

Methodology Maturity	Methodology Complexity	Software Maturity	Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.1	0.3	0	0

Potential for Failure Factor = (0.1 + 0.1 + 0.1 + 0.3 + 0 + 0)/6 = 0.1

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0.5	0

Consequence for Failure Factor = (0.9 + 0.5 + 0)/3 = 0.467

Risk Factor**:** P+C – (P*C) = 0.1 + 0.467 – (0.1 * 0.467) = 0.5203 high risk

Packet Decoder

The packet decoder module interprets the signals coming back from the sensors or stations and changes the system accordingly.

Methodology Maturity	Methodology Complexity	Software Maturity	Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.1	0.3	0.3	0.3

Potential for Failure Factor = (0.1 + 0.1 + 0.1 + 0.3 + 0.3 + 0.3)/6 = 0.2

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0.5	0

Consequence for Failure Factor = (0.9 + 0.5 + 0)/3 = 0.467

Risk Factor: P+C – (P*C) = 0.2 + 0.467 – (0.2 * 0.467) = 0.5736 high risk

2.5 UI/Controller Cost Analysis

DIRECT COSTS

All our software is open source (eclipse + JAVA swing) therefore there are no direct costs associated with actually buying software programs for using JAVA, which is the language that we are going to use.

We will need to buy an Octopus-550 because we need several serial ports to attach to the different PICs from the MasterPC. The cost is $115.

- Octopus-550: $115
- No additional software costs

Total Direct Costs: $115.00

INDIRECT COSTS

- Design & Building: 200 hours
- Testing: 50 hours
- Total Labor: 250 @ $10.00 = $2,500
- Computers: 8 @ $1500.00 = $12,000.00

Documentation, Meetings, Presentations (5 people):

- Labor: 100 Hours / pp @ $10.00 = $5,000

Total Indirect Costs: $19,500.00

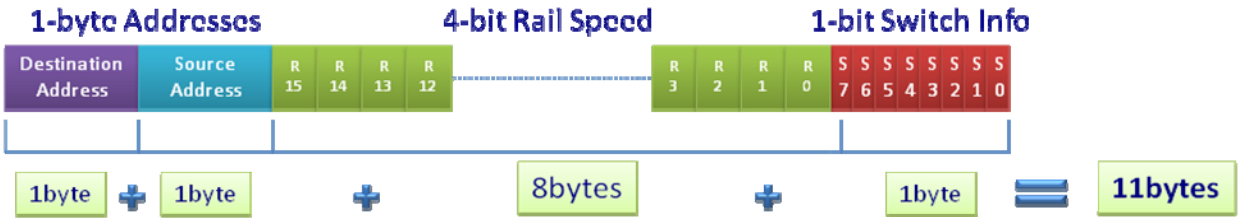
Total System Cost: $19,615.00

Networking Group

1.1 Networking Sub-System Specifications

RS-232 Communication

The MasterPC shall communicate to all the stations via the serial port using RS-232 standards. Packets that are sent to the stations shall have the following form:



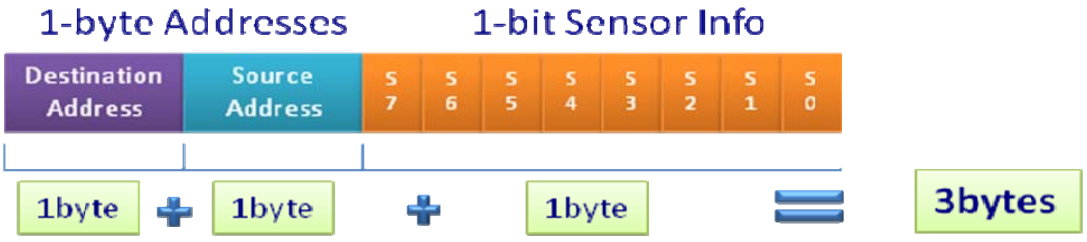
Each 4-bit rail data shall range from 0-F (16 different speed commands) whereas the 1-bit switch info shall control whether the rail should switch or not.

The speed of the RS-232 communication shall depend on the maximum speed that the PICs can handle while also maintaining a low probability of error during transmission.

PIC Microcontroller

Sensor Interrupts:

The sensors shall all be connected to the PIC's interrupt generating pins. Every time the sensor state changes, it shall trigger an interrupt which shall send a frame to the MasterPC to notify where the train is. This frame shall have the following format:



Rail Control:

The PIC shall analyze the information coming from the MasterPC and send the necessary commands to the hardware. For controlling rail power & speed, we shall use either a second PIC specialized in generating 4 different duty cycles at once. This means that we can have up to 4 different speeds at any station at any time.

Switch Control:

The PIC shall analyze the information from the PC, and based on the previous state of the coils, it shall send the necessary power to either the Red or the Green coil. An internal timer from the PIC shall be kept to count until we have supplied enough (0.3J) of energy to the coils. Furthermore, a watch-dog timer shall be implemented to guarantee that we do not power the coils for more than 50ms.

3.2 Networking Test Plan

RS-232 COMMUNICATION

Test Requirement 1: Verify that the correct packet is sent to the PC.

Testplan:

a. Write code to send out a specific packet to the hyperterminal and verify in the hyperterminal that the packet sent was actually received.
b. Assign the same value to the destination and the source address, the value assigned to the specific PIC at the specific station and verify that the sent byte is received by self.

Test Requirement 2: Verify that the correct packet is received from the PC.

Testplan:

a. Send out a packet frame via hyperterminal and write code to send back the received frame back to the hyperterminal and verify that the correct packet is received.

SENSOR INTERRUPTS

Test Requirement 1: Verify the correct operation of interrupt subroutines.

Testplan:

Use either the CFE or lab equipment to generate sensor outputs that feed into the PIC and check if the desired interrupts are generated, and that correct packets are sent to the PC.

RAIL CONTROL

Test Requirement 1: Verify that the proper signaling for the power in different sections of the rail is assigned as determined by the byte stream received from the user.

Testplan:

a)Write specific code to assign a default PWM frequency to power the rails one at a time, and check that the desired rails are powered with correct polarity.

b)Write the same code in the PC, and check that the PIC receives this information correctly, and powers the rails correctly.

Test Requirement 2: Verify that the Pulse Width Modulation (PWM) signals to control the speed of the motor is being assigned the correct frequency and duty cycle for one out of the sixteen pre-determined speeds.

Testplan:

a. Measure the PWM output pins of the PIC via an oscilloscope and verify that the frequency and duty cycle is being assigned properly.
b. Check that the PIC is able to generate 4 genuinely different duty cycles at any given time.

SWITCH CONTROL

Test Requirement 1: Verify the desired allocation of power to either the Red or Green coil depending on the information sent by the PC.

Testplan:

a. Check that if a switch is ordered to be turned by the PC,

Test Requirement 2: Verify that illegal switches are not being allowed that would allow two trains to crash.

Test Requirement 3: Verify the desired allocation of power to either the Red or Green coil depending on the previous state

3.3 Networking Requirements Analysis

	Requirements	RS-232	Sensor Interrupts	Rail Control	Switch Control
R001	rail switch control	✓		✓	✓
R002	engine power control	✓		✓	✓
R003	train proximity monitoring		✓		
R004	expandability and adaptability	✓	✓	✓	✓
R005	control and monitoring speed	✓	✓	✓	
R006	applications programming interface				
R007	maintenance user interface				
R008	demonstration application				
R009	modifications of the CFE layout				
R010	power input				

R001 – Rail Switch Control:

The tracks shall be controlled by the PC using the RS-232 Communication. The rail control and switch control will be programmed into the PIC to control the rail switches.

R002 – Engine Power Control:

The rail information shall be sent over the RS-232 in the correct packets to control the speed and direction. The packet sent from the PC to the PIC will include a 4-bit number (representing 0-F – the 16 speeds) which the PIC's logic will then interpret to power the correct rail the proper amount.

R003 – Train Proximity Monitoring:

Each sensor shall be connected to the PIC's interrupt generating pins. This shall monitor all the train proximity sensors in the track.

R004 – Expandability and Adaptability:

The system shall be expandable if the octopus is used we can add more stations. Also more sensors could be added because we can

R005 – Control and Monitoring Speed:

3.4 Networking Risk Assessment

RS-232 Communication

This is the communication from the computer to the 5 stations. We have devised a crude packet structure for the data. This subsystem is the structure of the packet and the retrieval of these packets by the PIC controller.

Methodology Maturity	Methodology Complexity	Hardware/Software Maturity	Hardware/Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.3	0.3	0.3	0.1	0.3	0

Potential for Failure Factor = (0.3 + 0.3 + 0.3 + 0.1 + 0.3 + 0)/6 = 0.2167

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0	0.5	0.7

If the RS-232 does not work we need to find a different method of data transmission. This would mean redesigning logic inside the PIC and possibly needing completely different hardware. It would delay the low level hardware progress as well as the software development.
Consequence for Failure Factor = (0 + 0.5 + 0.7)/3 = 0.4

Risk Factor:
P+C – (P*C) = 0.2167 + .4 – (0.2167 * 0.4) = **0.53 high risk**

Sensor Interrupts

This subsystem will monitor the sensors and send a packet of information back to the PC when a sensor has been tripped. We will generate an interrupt in the PIC when this event occurs to make sure the data is sent and sensor information is not missed. This subsystem will be software written in the PIC.

Methodology Maturity	Methodology Complexity	Hardware/Software Maturity	Hardware/Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.7	0.5	0.7	0.3	0.7	0.7

Potential for Failure Factor = (0.7 + 0.5 + 0.7 + 0.3 + 0.7 + 0.7)/6 = 0.6

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.7	0.3	0.3

Consequence for Failure Factor = (0.7 + 0.3 + 0.3)/3 = 0.4333

Risk Factor:
P+C – (P*C) = 0.6 + .4333 – (0.6 * 0.4333) = **0.7733 high risk**

RAIL CONTROL

The 4 bit data per rail will be converted into usable analog values for lower level hardware. The 4 bit data is a 16 value range from 0 (rail off) to F (full speed). A second PIC will convert this digital data to a PWM signal for use on the rails.

Methodology Maturity	Methodology Complexity	Hardware/Software Maturity	Hardware/Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.7	0.5	0.7	0.5	0.3	0.3

Potential for Failure Factor = (0.7 + 0.5 + 0.7 + 0.5 + 0.3 + 0.3)/6 = 0.5

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.7	0.5	0.5

Consequence for Failure Factor = (0.7 + 0.5 + 0.5)/3 = 0.5667

Risk Factor:
P+C – (P*C) = 0.5 + .5667 – (0.5 * 0.5667) = **0.7834 high risk**

SWITCH CONTROL

Signals will be sent out on parallel lines to appropriate low level hardware that will send power to the individual coils. There will be checks to make sure they coils are not over powered or oppositely powered. This subsystem will be software written for the PIC chip.

Methodology Maturity	Methodology Complexity	Hardware/Software Maturity	Hardware/Software Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.7	0.5	0.7	0.5	0.3	0.3

Potential for Failure Factor = (0.7 + 0.5 + 0.7 + 0.5 + 0.3 + 0.3)/6 = 0.5

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.7	0.5	0.5

Consequence for Failure Factor = (0.7 + 0.5 + 0.5)/3 = 0.5667

Risk Factor:
P+C – (P*C) = 0.5 + .5667 – (0.5 * 0.5667) = **0.7834 high risk**

3.5 Networking Cost Analysis

DIRECT COSTS:

Network

- PICs: $3.94 * 10 (two at each station) = $39.40
- Microchips MPLab – free software

Total Direct Costs: $39.40

INDIRECT COSTS

Network

- Design & Building: 200 hours
- Testing: 50 hours
- Total Labor: 250 hours @ $10.00 = $2500.00

Documentation, Meetings, Presentations(5 people):

- Labor: 100 Hours / pp @ $10.00 = $5,000

Total Indirect Costs: $7,500.00

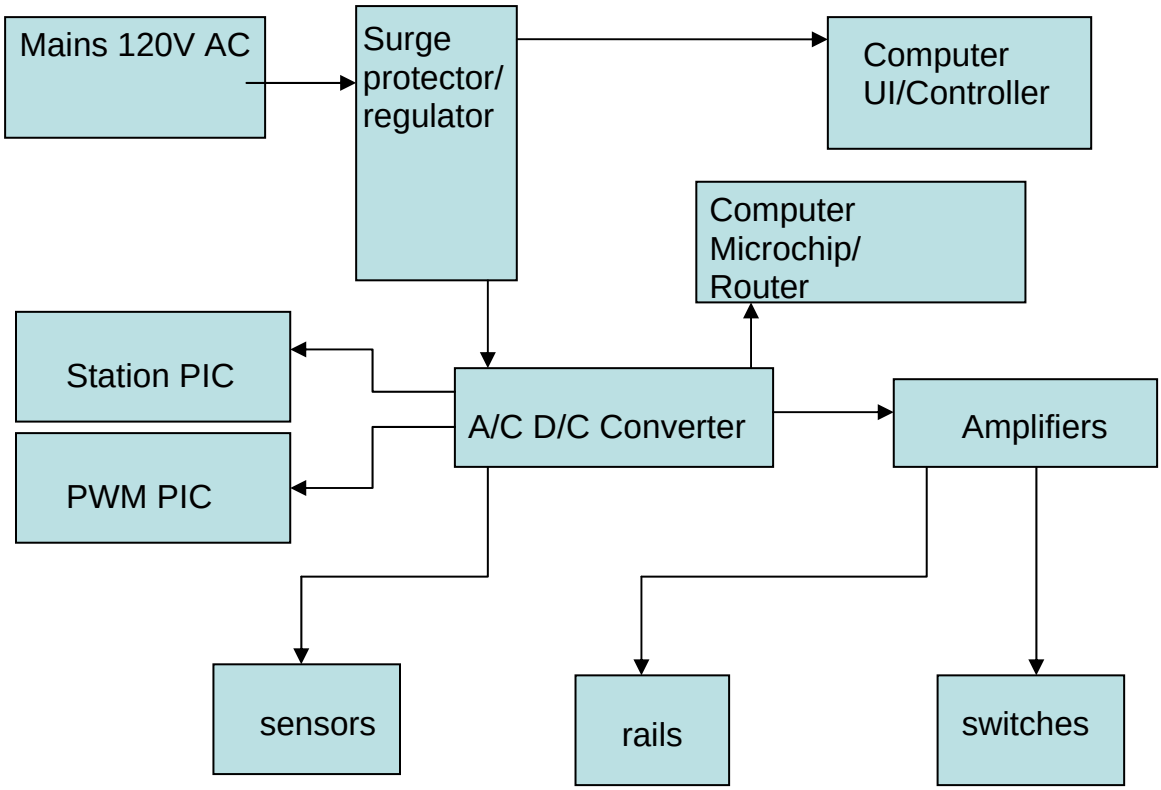
Total System Cost: $7,539.40

LOW LEVEL GROUP

4.1 Low-Level Sub-System Specifications

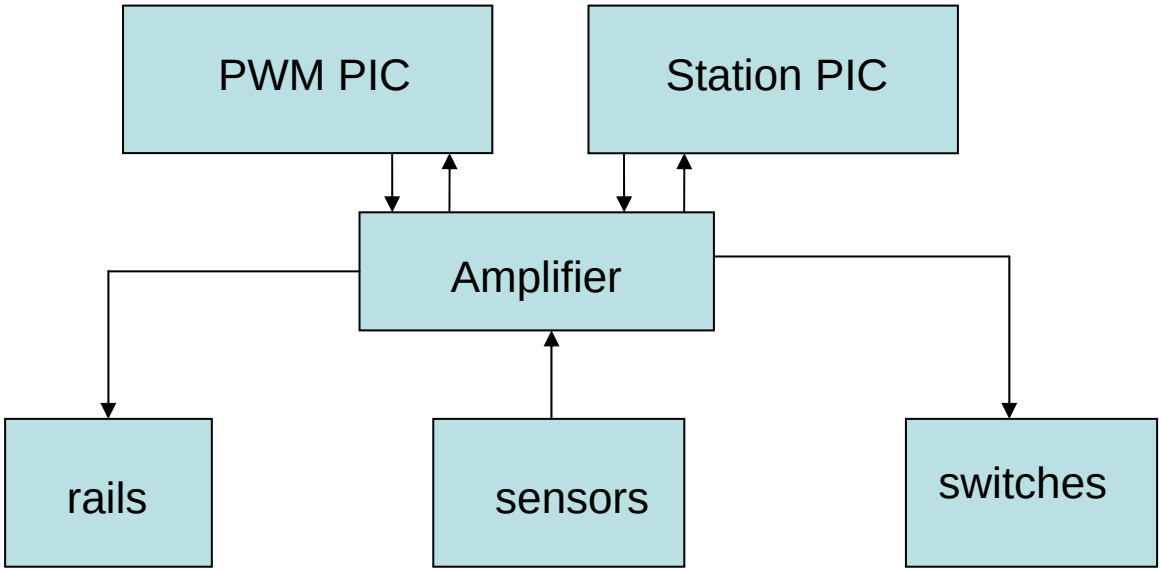
The low-level circuitry will be responsible for powering the entire system and amplifying signals as a buffer between each station microcontroller and the rail parts.

POWER BLOCK DIAGRAM



System description of how the project will be powered. The Main 120V AC block is the cord that is going into the wall which will be fed into the surge protector/regulator. The computer will be connected directly to the surge protector. The surge protector will feed directly to an A/C to D/C converter which will power the system. Each station microchip and the router microchip will get its necessary power from the A/C to D/C converter circuit. Also the amplification circuits will receive their power from this circuit as well as the sensors on the track. The rail and switch circuits will receive their power from the amplifier circuits which give the proper power from the signals received.

Signals Block Diagram



Station Microchip

This circuit will receive information off of the network and will output the necessary signals to the rails and the switches which will be received by the amplifier circuits.

PWM Microchip

This circuit will receive orders from the Station PIC to generate unique PWM signals, and send them to rails and switches which will be received by the amplifier circuits.

Amplifier

This circuit will receive rail and switch signals from the station microchip and amplify them to the necessary voltages and send them as outputs. This circuit will also take the sensor voltages as inputs and amplify them to the proper values and send the signal as an input to the PWM microchip.

Sensors

This circuit will take the sensor outputs and feed them to the amplifier circuit.

Rails

This circuit will accept the signals from the amplifier and power the rails, including the PWM signal for speed.

Switches

This circuit will accept the signals from the amplifier and power the switches.

4.2 Low-Level Test Plan

TEST FOR REED SWITCHES:

Testing of the circuits that are to include the reed switch will take place with the use of a bread board / prototyping board. The circuit will be connected to the sensors on the track, one at a time, and we will verify that the voltage output is the desired logic level that will eventually be transmitted back to the controlling computer.

Inputs: None
Outputs: Logic voltage to the PIC

TEST FOR THE SWITCHING MECHANISM:

The unique problem presented in testing this component of the track hardware is that if the solenoid for the switching is activated for more than 50ms then it will melt its plastic housing. Because of that testing for this component will consist mostly of computer simulations, or, if possible, we will use extra switches, not connected to the track with a bread board / prototyping board to insure that the circuit behaves as expected.

Inputs: Which switch and Red or Green
Outputs: >50ms of power to a switch

TEST FOR THE SPEED CONTROL:

The speed control component of the track will be tested in two ways. First it will be tested external from the track. It will be tested with the outputs that will eventually go to the rails instead connected to a volt meter. Once this test is passed we will connect the circuit to the rails and verify that the train behaves as expected when power is supplied.

Inputs: Rail segment and Speed
Outputs: Power to the rails of the track

TEST FOR POWER SUPPLIES:

The power supplies that we will create for this system will be tested by ensuring that the output, both voltage and current, meet their specifications.

Inputs: 120 V AC from the wall outlet
Outputs: Various DC voltages

4.3 Low – Level Requirements Analysis

	Requirement	Power Conversion	Direction/ Speed	Rail Switching	Sensors/ Proximity
R001	rail switch control			✓	
R002	engine power control		✓		
R003	train proximity monitoring				✓
R004	expandability and adaptability		✓	✓	✓
R005	control and monitoring speed				
R006	applications programming interface				
R007	maintenance user interface				
R008	demonstration application				
R009	modifications of the CFE layout		✓	✓	
R010	power input	✓			

R001 – Rail Switch Control

The rail switching circuit will take signals from the amplifier circuit which took signals that were taken from the network from software that allowed the independent programmed control of the rail switches.

R002 – Engine Power Control

The direction/speed or rails circuit will take signals from the amplifier which were taken from the network from software that allowed independent control of the energized track.

R003 – Train Proximity Monitoring

The proximity circuit will provide feedback of train location and allow independent programmed monitoring of all train proximity sensors.

R004 – Expandability and Adaptability

All circuits will interface directly with the train support expandability and adaptability. If the project were to expand it would only be necessary to increase the amount of low level circuits and now new designs would be needed.

R009 – Modifications of the CFE Layout

The train layout will not be modified as well as the tracks, track-switches, train sensors and trams themselves. The wiring connecting directly to the switches and sensors will also not be modified but the busses will be jumpered and augmented with additional hardware.

R010 – Power Input

The power conversion circuit will take the power directly from the wall (120V AC, 60Hz), and convert that power to any voltage needed in the system with multiple outputs, meeting this requirement.

4.4 Low-Level Risk Assessment

OVERALL SYSTEM POWER:

This module is responsible for supplying power to all components of the system, including the computers, the PICs, and the logic chips. It will be necessary to convert AC power from the wall to the appropriate DC voltage levels for each component requiring power.

Methodology Maturity	Methodology Complexity	Hardware Maturity	Hardware Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.3	0.1	0	0

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0	0.1

P(F) = (0.1 + 0.1 + 0.3 + 0.1 + 0 + 0) / 6 = 0.1
C(F) = (0.9 + 0 = 0.1) / 3 = 0.333

Risk Factor = 0.1 + 0.333 – (0.1 * 0.333) = **0.4 (Medium Risk)**

RAIL POWER:

Rail power will be controlled by varying the duty cycle of a PWM signal coming from the PIC. This module will be responsible for amplifying the PWM signal and transferring it to the rails to control the speed and direction of the train.

Methodology Maturity	Methodology Complexity	Hardware Maturity	Hardware Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.3	0.3	0.3	0.3	0	0

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0	0.1

P(F) = (0.3 + 0.3 + 0.3 + 0.3 + 0 + 0) / 6 = 0.2
C(F) = (0.9 + 0 = 0.1) / 3 = 0.333

Risk Factor = 0.2 + 0.333 – (0.2 * 0.333) = **0.467 (Medium Risk)**

RAIL SWITCHING:

This module is used to control the path that the train will take at each intersection of rails. A voltage supplied to a coil will change the position of the rails at each junction.

Methodology Maturity	Methodology Complexity	Hardware Maturity	Hardware Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.1	0.1	0	0

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0	0.1

P(F) = (0.1 + 0.1 + 0.1 + 0.1 + 0 + 0) / 6 = 0.0667
C(F) = (0.9 + 0 = 0.1) / 3 = 0.333

Risk Factor = 0.0667 + 0.333 – (0.0667 * 0.333) = **0.377 (Medium Risk)**

SENSORS:

The sensors are used to determine the position of the trains on the track. When the switch is activated, a short circuit exists across the terminals. Otherwise, an open circuit exists across the terminals. This module will be responsible for outputting a digital signal indicating whether or not a sensor is activated.

Methodology Maturity	Methodology Complexity	Hardware Maturity	Hardware Complexity	Dependency of Schedule on External Functional Groups	Dependency of Performance on External Functional Groups
0.1	0.1	0.3	0.1	0	0

Degradation of Technical/Operational Performance at System Level	Schedule Impact	Cost Impact
0.9	0.3	0.1

P(F) = (0.1 + 0.1 + 0.3 + 0.1 + 0 + 0) / 6 = 0.1
C(F) = (0.9 + 0.3 = 0.1) / 3 = 0.433

Risk Factor = 0.1 + 0.433 – (0.1 * 0.433) = **0.490 (Medium Risk)**

4.5 Low – Level Cost Analysis

DIRECT COSTS

Low Level Hardware

- Printed Circuit Boards: $20.00 * 7 = $140.00
- Logic Chips: $0.50 * 28 = $14.00

Total Direct Costs: $154.00

INDIRECT COSTS

Low Level Hardware

- Design & Building: 200 hours
- Testing: 50 hours
- Total Labor: 250 hours @ $10.00 = $2500.00
- Resistors, wires, capacitors etc = $5.00

Documentation, Meetings, Presentations(5 people):

- Labor: 100 Hours / pp @ $10.00 = $5,000

Total Indirect Costs: $7,505.00

Total System Cost: $7,659.00

6. Preliminary System Acceptance Test Strategy

The overall testing strategy for the LART-CS08 system will involve both overall tests to check a fully integrated system as well as tests to check if each of the subsystems is functioning.

The individual subsystems in the LART-CS08 system will be tested using the individual test plans included in this report. These test plans will test if subsystem specific operations are working correctly.

The overall system will be tested by running the following test procedures:

1. Power and operate a train to travel across each switching junction in the LART system and be able to direct it in all possible directions. The software interface should be able to monitor these changes in track position in real time.

2. Power and operate a train across each rail segment testing that each segment can operate at all 16 speed levels. Verify that the speed transitions from standstill to maximum speed occur relatively smoothly. The software interface must be able to monitor and display in real-time which segment(s) is currently powered and the speed of a train.

3. Power and operate a train to travel across each railway segment in the LART system and be able to activate and deactivate each proximity sensor. The software interface should be able the activation and deactivation of each sensor in real time.

4. Analyze the expandability and adaptability of the system in terms of the extent of modifications necessary and document results.

5. Test multiple trams operating on the LART with manual control by an operator using the software user interface. There should be no collisions, delays in response time or missed detection of events.

6. Test multiple trams operating on the LART with automatic control by controller software. There should be no collisions, delays in response time or missed detection of events.

7. Observe and document that no modifications to the original hardware configuration of the system have been made.

8. Observation that the system uses standard power from a 120AC 60Hz wall socket.

9. The following documents shall be delivered Users Manual, Maintenance Manuel, System Acceptance Test Plan, System Acceptance Test Report, QA Audit report, Prelimary Design Review Report & Critical Design review report. Additional documentation will include sustainability analysis report, an ethics report and an application programming interface document.

10. Test and document that all system parts and equipment meet the Environmental requirements (GPR002).

11. Test and document that all system parts and equipment meet the EMI/EMC requirements (GPR003).

12. Test and document that all system parts and equipment meet the Hazmats requirements (GPR004) as well as meet RoHS compliance.

13. Test and document that all system parts and equipment meet the Safety and Good requirements (GPR005) and a electrical safety plan is developed

14. Analyze and document that the system meets at least the requirement of 'System wide MTTR less than 1 week over the system lifetime' (GPR007).

15. Analyze and document that the system meets at least the requirement of 'System wide MTBF of greater than 100 hours over the system lifetime' (GPR006).