

FAILURE ANALYSIS OF NETWORK BASED ACCESSIBLE PEDESTRIAN SIGNALS IN CLOSED- LOOP OPERATION

Final Report

KLK719

N11-02



**National Institute for Advanced Transportation
Technology**

University of Idaho



Richard Wall, Brian Johnson, Michael Kyte

March 2011

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle FAILURE ANALYSIS OF NETWORK BASED ACCESSIBLE PEDESTRIAN SIGNALS IN CLOSED-LOOP OPERATION		5. Report Date March 2011	
		6. Performing Organization Code KLK719	
7. Author(s) Wall, Dr. Richard; Johnson, Dr. Brian; Kyte, Dr. Michael		8. Performing Organization Report No. N11-02	
9. Performing Organization Name and Address National Institute for Advanced Transportation Technology University of Idaho PO Box 440901; 115 Engineering Physics Building Moscow, ID 83844-0901		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. DTRT07-G-0056	
12. Sponsoring Agency Name and Address US Department of Transportation Research and Special Programs Administration 400 7th Street SW Washington, DC 20509-0001		13. Type of Report and Period Covered Final Report: August 2009 February 2011	
		14. Sponsoring Agency Code USDOT/RSPA/DIR-1	
15. Supplementary Notes:			
16. Abstract The potential failure modes of a network based accessible pedestrian system were analyzed to determine the limitations and benefits of closed-loop operation. The vulnerabilities of the system are accessed using the industry standard process know as Failure Modes and Effects Analysis. This analysis reveals that because the closed-looped operation increases operational observability, there is only one failure mode that is not detectible. Investigation on reliability of hardware and software indicate that the probability of this particular event is extremely low. The three primary components of a distributed real-time control system are hardware, software, and communications. Environmental conditions are the predominant factors that determine hardware reliability. The system was tested for its ability to function in the presence of electrical transients specified by the National Electrical Manufacturers Association (NEMA). All hardware components meet or exceed NEMA temperature requirements. Software errors are difficult to detect and difficult to predict. Best practices in development and testing can minimize the number of errors. Testing includes multiple years of both laboratory and field testing.			
17. Key Words Traffic control devices; traffic signal control systems; signalized intersections; warning devices for the handicapped; pedestrian safety; pedestrian areas; pedestrian vehicle interface.		18. Distribution Statement Unrestricted; Document is available to the public through the National Technical Information Service; Springfield, VT.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 24	22. Price ...

TABLE OF CONTENTS

Figures.....	i
Introduction.....	1
Background.....	1
Clarification of terminology:	3
Description of Tasks	4
Task 1: Review of Advanced Accessible Pedestrian Station Engineering.....	4
Task 2: Design System Test Program.....	8
Task 3: System Testing.....	9
Tasks 4 - 7: Design and Integration of Closed-Loop Software	10
Findings; Conclusions; Recommendations.....	10
Results of Review of Advanced Accessible Pedestrian Station Engineering.....	10
Future Work.....	15
References.....	17
Appendices:	19
Appendix I. Steps to Generate an FMEA Matrix	19
Appendix II.AAPS FEMA Analysis.....	20

FIGURES

Figure 1: AAPS system block diagram.....	2
Figure 2: Block diagram of a classical closed-loop control system.....	3
Figure 3: System reliability over the life cycle of a device or system.	5
Figure 4: Typical state diagram for APB.....	9

INTRODUCTION

The primary focus of the work provided for by this grant was to access the factors that affect the safe operation of the Advanced Accessible Pedestrian System (AAPS). This report addresses the activities associated with completing each task of the proposed work by describing the methods and materials utilized along with the results that the tests provided. This report also discusses proposed changes to the hardware design as well as to the software that defines the AAPS operating characteristics.

BACKGROUND

Researchers at the University of Idaho (UI) have developed a new accessible pedestrian system (APS) based upon an enabling technology that has been designated as Smart Signals. This enabling technology is based upon a network of microprocessor pedestrian stations that communicate to an interface controller located in the traffic controller cabinet. The architecture of the Smart Signals AAPS is described in detail in a National Institute for Advanced Transportation Technology (NIATT) final report for project KLK715 [1]. Readers of this report are encouraged to read the KLK715 report if they unfamiliar with the ongoing Smart Signals research.

In summary, as shown in the system block diagram seen in Figure 1, the AAPS is comprised of an advanced pedestrian controller (APC) and multiple advanced pedestrian buttons (APB) located next to intersection crosswalks. The APC, located in the traffic controller cabinet, interfaces at the field terminals of the load switch outputs that drive the traffic signals to sense the pedestrian signal on-off status. The APC also connects to the field terminals where the conductors from conventional buttons terminate. The APC contains a power transformer to convert the 120VAC power to 12VAC. This power is distributed to all pedestrian buttons to power the microprocessors required to play the audio messages associated with APS operations prescribed by the Manual for Traffic Controller Devices [2]. Communications between the APC and APBs is provided using 10 Mbps Ethernet over power lines (EoP).

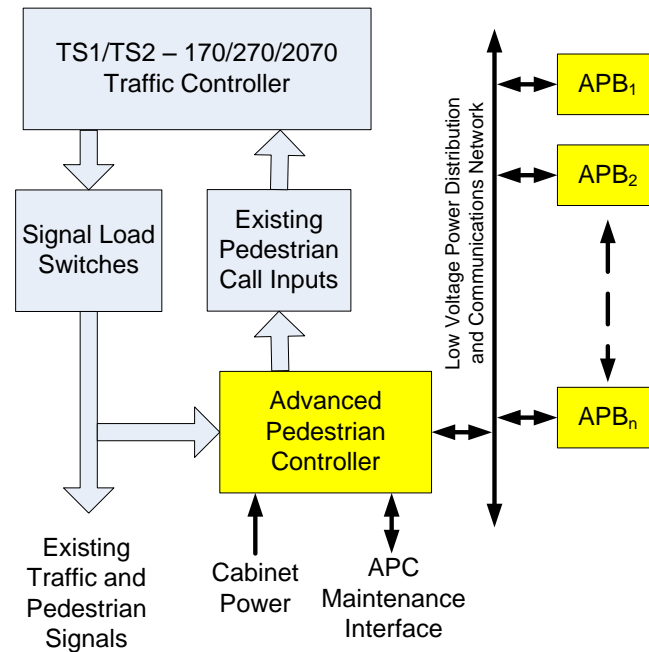


Figure 1: AAPS system block diagram.

One of the advantages of the Smart Signal technology is the ability for bidirectional communications. This is in contrast to conventional traffic signal devices (visible signals and audible messages) that operate in open-loop fashion. The primary concern addressed by this research is the verification that the audible message that instructs a pedestrian that the WALK sign is active for the designated crosswalk. For a pedestrian who is blind or has low vision acuity, the correctness of this message is critical for his or her safety. The research project set out to take advantage of the bidirectional communications to overlay the operations control algorithms with an operations validation process involving closed-loop communications.

Equally important for reliable APS operations is the disruption to traffic flow in the event of an APS failure. Current AAPS safe-fail operation generates constant pedestrian calls on all pedestrian inputs. Should an intersection operate with one approach activated by a vehicle or pedestrian call, then the constant call for that approach would cause that phase to be served for the minimum green time even though there is no pedestrian or vehicle traffic. This could cause the traffic on the preferred approaches to stop after a maximum green time. It is also possible that false pedestrian detection could disrupt coordinated traffic patterns. In either

event, reliable operation and prompt repair of an APS is highly desirable to efficiently serve pedestrians and vehicles.

Reliable operation is critical to user confidence in that the pedestrian system will provide them with the service they expect. Feedback by means of an audible, visual, and tactile feedback enhances pedestrian awareness for pedestrians who have normal vision as well as those who have low vision or who are blind.

Clarification of terminology:

Throughout this report, the term “closed-loop” will be used. This term has different meanings to different industries. A common example of closed-loop control is a cruise control employed on many automobiles. As illustrated in Figure 2, the measured error signal is generated from the computed difference between the reference input and measured output. The sensor is a device or instrument that measures the response of the system being controlled. The controller is a collection of electronics and/or electromechanical devices that perform an algorithm to apply the appropriate control signals to the system under control. In general, the feedback signal corrects the control actions sent to an external dynamic system [3]. This feedback signal is continuous in nature and the loss of the feedback usually constitutes a system failure.

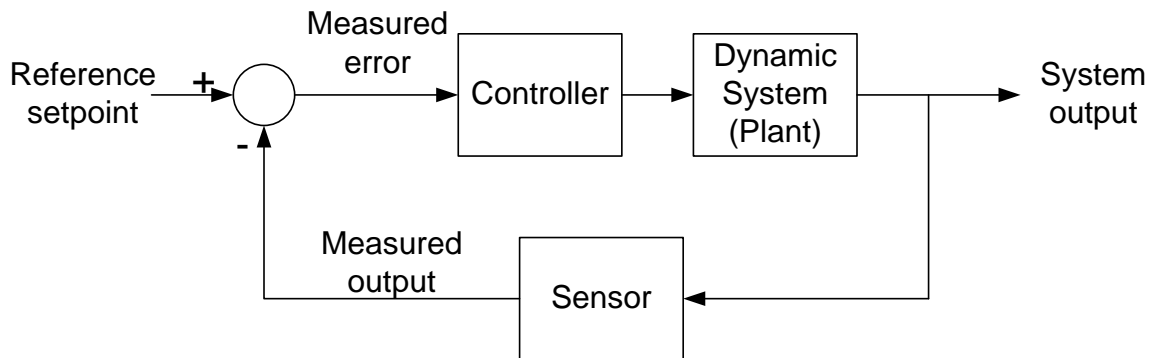


Figure 2: Block diagram of a classical closed-loop control system.

Chapter 3 of the Traffic Signal System Handbook describes a closed-loop system as distributed processor traffic control system with control logic distributed at three levels [4]. The levels of control are a local controller, an on-street master, and an office computer. This handbook also describes three control modes that the system may operate: “free” mode or time-of-day mode, manual mode, and responsive mode. [5]. The time-of-day modes and manual modes do not fit well with the classical context of closed-loop control. The responsive mode of operations as defined by the Traffic Signal Systems Handbook is closest to the classical definition of closed-loop control. However, loss of feedback for a responsive system does not necessarily constitute a system failure.

Throughout this report, the term “closed-loop” will be used in the sense of classical control theory. Error signals or conditions are determined by detecting a difference between the desired operation and the actual indicated operation. The error signal will be used to change the system that results in the safest mode of operation. More specifically, the closed-loop control paradigm will be applied to the Smart Signals distributed real-time safety critical control system.

DESCRIPTION OF TASKS

The following subsections of this report addresses the tasks that were proposed and describes the activities completed for each task.

Task 1: Review of Advanced Accessible Pedestrian Station Engineering

There are two fundamental components of all computer based systems: the hardware and the software program that the computer or computers execute. In a distributed control system such as Smart Signals, the communications link between devices has the added component of possible data corruption from electrical interference. The design of the AAPS paid particular attention to the mode of operation in the presence of failure results in defaulting to the safest operations for both pedestrians and vehicle operators. The following sections will address each of the three components of distributed real-time safety critical control systems, the failure modes, and the mitigating action.

Task 1.1: Accessing Hardware Reliability

Components and systems that are collections of components have failure rates that are stochastically determined by testing a multiplicity of units over a range of environmental operating and storage conditions. Failure rates are not uniform throughout the life cycle of a system or component as demonstrated by the graph in Figure 3. After the initial burn in time that includes component infant mortality, the reliability levels off to a constant rate over its useful life until the period of wear-out occurs. Wear-out is more of an issue with mechanical devices than for electronic components. Reliability is measured as mean time between failures (MTBF) or mean time to failure (MTTF) with units of hours. (Discussions with manufacturers of industrial electronic systems reveal the new product development for an otherwise fully functional system is more likely to occur because of the unavailability of critical components due to supplier discontinuation.)

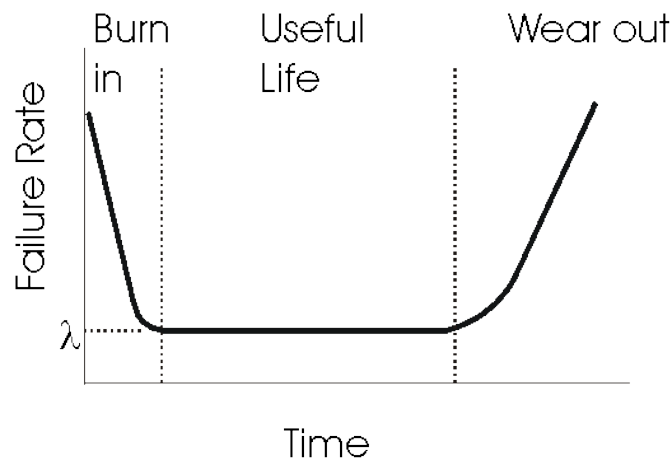


Figure 3: System reliability over the life cycle of a device or system.

The military handbook on Reliability Prediction of Electronic Equipment [6] provides a systematic process for determining the system reliability. Commercial software is also available that is based on MIL-HDBK-217F as well as other reliability standards to assist in determining system reliability. Contributing factors include normal operating temperature, time operated at extreme temperatures, and the device maturity. It is beyond the scope of this report to explain the process of determining reliability. Section 2.1.5 of the National Electrical Manufacturers Association (NEMA) TS 2 standard for Traffic Controller

Assemblies with National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) Requirements-Version 02.06 specifies that operating temperature range be from -35°C to $+74^{\circ}\text{C}$ [7].

Section 2.1.6 of the NEMA TS 2 standard describes the transient testing for inputs, outputs, and power connections. Transient testing was completed in December 2010 at Schweitzer Engineering Laboratories (SEL) of Pullman, WA. The company is internationally renowned for producing computer based power system protection equipment. Prior to testing, researchers from the UI met with engineers at SEL to review the engineering design of the APC interface and APB circuit boards. After implementing design modifications, the equipment was brought back to SEL for testing. The transient wave characteristics specified in Section 2.1.6 of the NEMA standard were generated using commercial transient generators commonly used to test power system protection equipment. The test included conditions when power was not applied to the AAPS as well as when the AAPS was operating under full power.

Task 1.2: Accessing Software Reliability

There exists a profile similar to the reliability curve shown in Figure 3 for software error detection. Frequently the error rate metrics are in units of errors detected per unit of time such as months or hours of operation. Software reliability is a function of coding practices and code complexity. Software reliability is distinctly different from hardware reliability because software does not degrade over time. The mean execution time shall be used to recalculate the MTBF values. It is assumed that if a failure occurs in a service, it will take half the execution time to reach the fault. After release of a product, software errors are encountered when untested execution paths are run revealing a performance malfunction. Software errors can remain undetected throughout the life cycle simply because the system is never expected to perform under the circumstance that would execute the code where the error exists. The longer a product is in service, the higher the probability that the software error will be revealed.

Code coverage evaluation tools available in many high end integrated development environments (IDE) for software development can help to reduce the number of branches that are untested when a product is released. The process for using this tool requires that the program be run for a period of time with as many variations of input conditions as possible. After a set period of time, the execution of the program is halted and the diagnosis of the code coverage report is analyzed. The code coverage tool reports functions (subroutines) that are executed and the amount of time required to complete the function. It also reports functions that are not executed. The analysis of the program code should reveal the next course of action. Possible conclusions are that there are no conditions that would result in calling the function not covered; in which case, the code should be removed from the program. Another conclusion is that the inputs were never placed into the condition that results in the function being called. For this case, the test suite should be modified to generate necessary conditions.

Software errors are always present starting at the time when a system is put into service. It is more appropriate to address software quality in terms of maintainability, errors per thousand lines of source code, algorithm complexity, and number of decision points or code branches. The programming language also effects software reliability. Unfortunately, the programming language “C”, the most prevalently used programming language for embedded systems, has one of the lowest ratings for being able to generate unambiguous programs.

A search of the literature reveals that following a few, but relatively simple, practices when developing computer programs will result in discovering and clearing most software errors [8,9]. The code inspection is predicted to eliminate up to 90% of software errors prior to the systems introduction to the public [10].

For our metrics of evaluation of software quality, we maintained a record of software issues and posted them on the internet using Wiki server software for members of the development team to review. The errors were discovered by observing performance of AAPS systems over 15 months of field trials encompassing four test sites in the U.S.

Task 1.3: Accessing Communications Reliability

Real-time distributed control using Ethernet network communications has existed since before 1980, and today it is considered to be a mature technology [11]. Ethernet over power line (EoP) uses the 60Hz power conductors to also carry Ethernet communications. From its inception, HomePlug was intended to be a standard that would allow products to use existing home electrical wiring to communicate with each other and connect to the Internet. The first HomePlug standard, HomePlug 1.0, was released in June 2001 [12]. To this author's best knowledge, to date, the only other traffic control device that uses EoP for real-time communications is the Autoscope vehicle detection system [13]. The issue of EoP communications reliability was the subject of a master's degree thesis in 2010 [14].

Task 2: Design System Test Program

The purpose and challenge of software testing is succinctly stated in the syllabus of a course that studied this subject. "Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we cannot completely test a program with moderate complexity. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time, and quality [15]."

It became most proficient to use the operation code for testing unless the operation of a specific hardware component came into question. The functionality of the software was modeled using state diagrams that were generated for each possible operational sequence. These state diagrams were generated prior to writing any software to describe the system functionality during different modes of operation. Subsequently, processor code was written to implement the structure described by the state diagrams. After code development, the program execution was stepped through by a test team to verify that based upon the input

conditions, the code execution transitions to the proper next state. An example of one such state diagram is shown in Figure 4 [16]. The APB program required six state diagrams to fully model all APB operations.

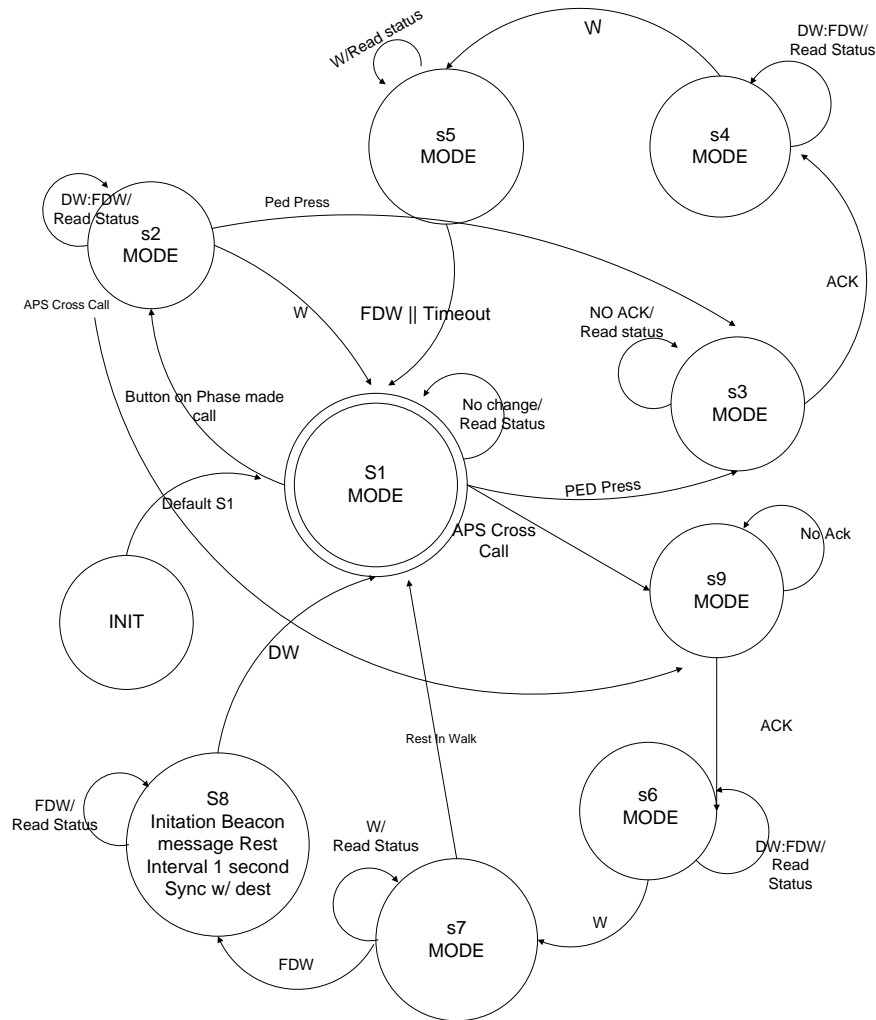


Figure 4: Typical state diagram for APB.

Task 3: System Testing

The evaluation process known as FMEA or Failure Modes and Effects Analysis was initiated during the Apollo space mission in the 1960's. The purpose of this analysis is to determine the possible weaknesses in a system design and determine the level of effort and cost that is needed to reduce risk associated with failures. The ten steps listed in Appendix I are

commonly used to generate the FMEA matrix [17]. The process of generating the FMEA matrix is not an exact science; it requires numerous subjective estimations. The quality of the matrix and the decisions derived depend on the experience and knowledge possessed by the group that generates the matrix. Data used to generate our fault matrix resulted from observations and tests of the AAPS in the laboratory and in the field at beta test sites.

Tasks 4 - 7: Design and Integration of Closed-Loop Software

These four tasks encompass the software that provides the closed-loop supervisory control algorithms to fault detection and safe-fail operations. The closed-loop software monitors the temporal validity of the data exchanged between the APC and individual APBs. Even for data that has not changed in value, the confidence level in the data's validity degrades over time in real-time controls unless reinforced in a timely manner.

FINDINGS; CONCLUSIONS; RECOMMENDATIONS

Results of Review of Advanced Accessible Pedestrian Station Engineering

Hardware:

The basic assumption that will be made concerning hardware is that the system was installed and configured correctly. This is a reasonable assumption and one that is made each time a new traffic signal system is commissioned. Personnel installing the equipment are expected to have thoroughly tested the system prior to permitting use in the public domain.

i. Beta Test Site Field Experience:

The factors that are used to determine the reliability of integrated circuits clearly reveal that using a few large scale integration components has higher reliability than systems of numerous small scale integration components. Experience with reliability studies readily demonstrate that the reliability of the system is dominated by hardware connections.

The engineering of the AAPS took these observations derived from studying the reliability prediction methods into consideration during the design process. All components were

verified to be rated for the industrial component temperature range of -40°C to $+85^{\circ}\text{C}$. The operating power and voltage ratings were also verified for all components. Beta testing in St. Paul, MN began in February 2010 and continues with no cold or hot weather related failures. There are currently three additional test installations for the AAPS: Lafayette, IN, Las Vegas, NV, and Moscow, ID.

This is not to say that there have been no equipment failures in any of the four test installations. However, as will be described later in this report, the failures do not result in an increased risk to pedestrians or the operators and passengers of motor vehicles.

To verify that the AAPS consistently reverted to the safe-fail operating conditions, all possible fault scenarios listed in Appendix II were tested in the laboratory. Additionally, APB and APCs did fail at installations in St. Paul, MN, Lafayette, IN, and Las Vegas, NV. In all instances, the AAPS correctly executed the fail-safe operating mode. The following case example revealed the system's ability to perform safely during failures.

At the St. Paul intersections, conductive foam was packed between circuit boards to protect them during shipping. One APC began intermittent operation and was replaced. In each instance, the APS went mute and persistent calls were placed on the traffic controller. Inspection of the APC revealed that the conductive foam was not removed before installation. After the conductive foam was removed, the system has now been functioning correctly for the past 13 months.

Both in Lafayette and Las Vegas, one APS failed to boot correctly. In both cases, the APC placed a persistent call exclusively for the crossing served by the failed APB. The APBs were replaced and sent back to the UI for testing. Analysis of the failed APBs from Lafayette and Las Vegas revealed a design flaw with the APS. The solution was to remove a power up reset integrated circuit for the EoP transceiver and allow the microprocessor to reset the EoP transceiver during the microprocessor's boot up process. No other circuit modifications were required.

During our preparation to have the AAPS certified for NEMA rating, it was noted that there was a requirement for systems to tolerate a 500 ms power outage without the traffic device rebooting. The requirement was met by installing a 25,000 μ F, 25 VDC capacitor across the APC's unregulated power supply. The APC is now capable of operating correctly for power outages of less than 600ms.

ii. Transient Protection

Modifications to the APC hardware following the meeting with SEL engineers were confined to increasing conductor size and reducing the length of wires to reduce the electrical impedance of the grounding circuits. These modifications were suggested by SEL engineers based upon numerous years of electrical protection equipment design and manufacturing. Due to the low cost of the modifications, there were no tests to determine if the original APC design would meet transient protection requirements. No design changes were recommended for the APB.

The AAPS equipment was tested for functionality after each transient test. In all cases, the AAPS equipment functioned normally.

Software:

The primary means of software testing has been through laboratory tests and demonstration field installations. Four test sites have been in service for up to 13 months as of this report. One intersection has recorded over 3,000 pedestrian calls without any software error reported or recorded by the APC event log.

A record of software testing and reported errors is listed on the Wiki web site <http://pedlab.ece.uidaho.edu/dokuwiki/aaps:testing>. This on-line report demonstrates the systematic testing performed for the 2010 development phase. Inspection of this on-line document reveals that this documentation continues today.

i. Communications

After significant evaluation and testing, Sapp [14] notes several environmental factors that determine the reliability of EoP. Since the communication uses frequencies up to 40 MHz, the home 120VAC wiring seems to be ill suited for Ethernet communications. 40 db of attenuation of the high frequency signals can reduce the effective communication rate from 13 Mbps to less than 1.9 Mbps. The AAPS requires a maximum effective data rate of 0.99 Mbps. Laboratory tests show that even at this small bandwidth, there were no communication failures.

Issues that affect the attenuation are wire gauge, wire capacitance, and termination impedance. Of the 4 beta test sites, only one site was unable to utilize existing pedestrian button conductors for EoP communications. It has since been determined that a transient protection component with high device capacitance was mistakenly placed across the APB power terminals thus shorting out much of the high frequency energy at this site.

ii. FMEA Matrix

The FMEA matrix is provided in Appendix II. The issues are listed in order of severity of possible consequences in the event of failure. In the final analysis, the only possible undetectable failure is if the APB hardware were to malfunction in such a way as to play a WALK message and report that it is playing some other message. This error is undetectable by the AAPS system.

To assess the risk, the probabilities of occurrence of a very particular combination of hardware and/or software errors; the probability that a pedestrian is at the crosswalk who would not be able to see the WAIT signal and the probability that the pedestrian would not be able to determine that the parallel traffic had a red signal, would have to be multiplied together. The probability that a pedestrian with the specified limitations is at the crosswalk when the processor failed in this specific way would be very difficult to determine. But that probability will only reduce the total probability given the probability for a specific processor. The failure rate or probability of failure is the inverse of the MTBF. The published MTBF for the NXP LPC2468 processor used in the APBs is 2,580,000,000 hours or

approximately 300,000 years. Subsequently, the possibility of this particular failure precipitating into this prescribed scenario is less than one chance in 2,580,000,000.

The primary source of software errors in the AAPS, as with any software based product, are a result of product feature enhancement sometimes referred to as “feature creep.” We wish that we could say we always followed our own advice. In the rush to satisfy customer demand, best practices are ignored and the time is not taken to implement peer code inspection. Software errors related to upgrades and enhancements are usually detected in production. We can only cite the following justifications:

1. Limited resources – graduate students have a full course load.
2. Good ideas are exciting and code inspections are boring.
3. Our industrial partner is anxious to accommodate paying customers.

In an effort to minimize risk due to failure, the AAPS system was designed to maximize operational observability. The primary means of providing the fault security is the expectation of bidirectional communications in a deterministic manner. The message protocol for the AAPS is described in the NIATT report for KLK715. This report shows that each time the APC sends a status update packet to each individual APB with a *SetRequest* message each APB in return generates a *GetResponse* message. If the APC fails to receive the appropriate *GetResponse* message, that APB is flagged by the APC as failed and appropriate action is taken as described below.

Based upon the FMEA analysis, a strategic fault mitigation and recovery plan was developed by the NIATT development team and our industrial commercialization partner. The following policies were established and are listed in order of priority:

1. Faults will be detected and mitigated at the location closest to where an incorrect action would have the direst consequences.
2. The APB will not play any audible message if the pedestrian signal status provided by network communication was older than 500 ms.
3. All pedestrian stations become inert in the event of a loss of network communications due to any failure. (Inert – no audible messages, LED indications, or vibrotactile actions.)

4. APB will revert to the locator tone whenever the pedestrian signal status indication from the APC is not in a WALK condition.
5. The APC will place a constant call on any APB that does not respond to a status update within 500 ms.
6. The APC will change the LED indication on the APC panel from green to red.
7. The APC will continue to send pedestrian status information to all APBs and look for a response message.

In the final analysis, we see that closed-loop operation causes all but one failure mode to be observable. The ability to detect failures also provides the ability to report. Present capabilities of the AAPS allow alert email messages as well as remote diagnostics provided there is access to network services. The operation of the AAPS is maintained to the highest degree possible in the event of a failure of any one APB.

FUTURE WORK

It is expected that, with the number of lines of source code required for the AAPS, software errors will continue to be identified. Working closely with our industrial partner, the errors will be identified and corrected.

Regardless of system reliability, the traffic industry favors an independent device to monitor traffic and now pedestrian control devices. The control of traffic signals is currently monitored by a device called a conflict monitor (CM) or malfunction management unit (MMU). These devices contain a hardware configuration circuit board on which jumpers indicate which signals are compatible by detecting the voltages on the outputs of the signal load switches. The CM or MMU is expected to detect conflicts between vehicle traffic signals as well as between pedestrian and traffic signals. Even though the AAPS is capable of detecting such conflicts, the industry practice expects an independent testable monitor that can detect incorrect operations and place the pedestrian stations into a benign state. We are initiating research into development of a pedestrian fault monitor (PFM) that, in the event of an incorrect operation, will de-energize all pedestrian station until manually reset.

Recent studies have shown that beaconing is a significant aid for helping blind pedestrians to complete a street crossing without straying outside the crosswalk. A second speaker will be added to all APBs that are directed toward the crosswalk being served to provide a beaconing signal for blind and low vision pedestrians.

We are also investigating ways to incorporate passive video detection with APB operations. Preliminary investigation has revealed a safety concern of inattentiveness and/or distraction resulting in pedestrian related crashes. It would be our intention to use the passive detection to alert pedestrians of potential dangers.

REFERENCES

- [1] Wall, R. W. “K L K 7 1 5 : Commercialization and Field Distribution of Smart Pedestrian Call Signals,”
http://www.webs1.uidaho.edu/niatt/research/Project_Descriptions/KLK715.htm.
- [2] “Manual for Traffic Controller Devices 2009,” US Department of Transportation Federal Highway Administration, available at <http://mutcd.fhwa.dot.gov/>.
- [3] Digital Control System Analysis and Design, 3rd Ed., C.L. Phillips, and H.T. Nagle, Prentice Hall Publishers, Saddle River, NJ, 07458, ISBN 0-13-309832-X, 1995, pp. 1.
- [4] “Traffic Control Systems Handbook,” Federal Highway Administration, Office of Transportation Management, October, 2005, Chapter 3, pp 3-11 – 3-12, available at: <http://ntl.bts.gov/lib/jpodocs/edldocs1/13480/ch3.pdf>.
- [5] Abbas, M., H. Charara, N. Chaudhary, and Y. Jung, “Distributed Architecture and Algorithm for Robust Real-Time Progression Evaluation and Improvement,” Report No. FHWA/TX-06/0-4729-2, Texas Transportation Institute , The Texas A&M University System , College Station, Texas 77843-3135, October 2005, pp 11-12.
- [6] “MIL-STD-217F Reliability Prediction of Electronic Equipment,” U.S. Department of Defense, December 2, 1991. <http://snebulos.mit.edu/projects/reference/MIL-STD/MIL-HDBK->
- [7] “Traffic Controller Assemblies with NTCIP Requirements—Version 02.06” NEMA TS 2-2003, The Association of Electrical and Medical Imaging Equipment Manufacturers Association, available for order at <http://www.nema.org/stds/ts2.cfm>.
- [8] Stapko, T., “10 Tips Make Embedded-System Code Easy to Maintain,” Electronic Component News, January 2, 2008, available at <http://www.ecnmag.com/Articles/2008/01/10-Tips-Make-Embedded-System-Code-Easy-to-Maintain/>.
- [9] Holzmann, G.J., “The Power of Ten -- Rules for Developing Safety Critical Code,” *IEEE Computer*, June 2006, pp. 93-95.
- [10] Almeida, J.R., J.B. Camargo, B.A. Basseto, and S.M. Pax, “Best Practices in Code Inspection for Safety-Critical Software,” *IEEE Journal on Software*, May/June 2003, pp. 56-63.
- [11] Moss, B., “Real-time Control on Ethernet,” *Dedicated Systems Magazine*, 2000-Q2, available at http://www.omimo.be/magazine/00q2/2000q2_p053.pdf.
- [12] HomePlug Powerline Alliance, <http://www.homeplug.org/home/>.

- [13] Autoscope Video Detection, Image Sensing System, Inc, <http://autoscope.com/>.
- [14] Sapp, Z., “Real Time Network Control for Advanced Accessible Pedestrian Systems Using Ethernet Over Power Line,” Master of Science Degree Thesis, University of Idaho, August 2010, pp. 21.
- [16] Pan, J., “Syllabus for Software Testing,” Carnegie Mellon University, Spring 1999, available at http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/.
- [17] Craviotto, C., “Pedestrian Station Design Using Distributed Real-Time Processing,” Master of Science Degree Thesis, University of Idaho, June 2010, pp. 73.
- [18] “10 Steps to Creating a FMEA,” Leadership, Lean, and Six Sigma Academy, available at <http://lssacademy.com/2007/06/28/10-steps-to-creating-a-fmea/>.

APPENDICIES:

Appendix I: Steps to Generate an FMEA Matrix

1. **List the key process steps in the first column.** These may come from the highest ranked items of your FMEA matrix.
2. **List the potential failure mode for each process step.** In other words, figure out how this process step or input could go wrong.
3. **List the effects of this failure mode.** If the failure mode occurs, what does this mean to us and our customer...in short what is the effect?
4. **Rate how severe this effect is** with 1 being not severe at all and 10 being extremely severe. Ensure the team understands and agrees to the scale before you start. Also, make this ranking system “your own” and don’t bother trying to copy it out of a book.
5. **Identify the causes of the failure mode/effect** and rank it as you did the effects in the occurrence column. This time, as the name implies, we are scoring how likely this cause will occur. So, 1 means it is highly unlikely to ever occur and 10 means we expect it to happen all the time.
6. **Identify the controls in place to detect the issue** and rank its effectiveness in the detection column. Here a score of 1 would mean we have excellent controls and 10 would mean we have no controls or extremely weak controls.
7. **Multiply the severity, occurrence, and detection numbers** and store this value in the RPN (risk priority number) column. This is the key number that will be used to identify where the team should focus first. If, for example, we had a severity of 10 (very severe), occurrence of 10 (happens all the time), and detection of 10 (cannot detect it) our RPN is 1000. This means all hands on deck...we have a serious issue!
8. **Sort by RPN number and identify most critical issues.** The team must decide where to focus first.
9. **Assign specific actions with responsible persons.** Also, be sure to include the date for when this action is expected to be complete.
10. **Once actions have been completed, re-score the occurrence and detection.** In most cases, we will not change the severity score unless the customer decides this is not an important issue.

Appendix II: AAPS FEMA Analysis

I. Incorrect audible message:

- a.** Description: This error is manifested by an incorrect audio message being played.
 - i. A message indicates that the WALK signal is on when the WAIT signal is flashing or in the solid on state
 - ii. The APB would give the walk message for the wrong crosswalk when the WALK signal is on
 - iii. A WAIT or LOCATOR tone is played when the WALK signal is on
- b.** Possible causes:
 - i. Software program error
 - ii. System setup error, the incorrect audio files were programmed to the APB
 - iii. Pedestrian button failed
 - iv. Pedestrian fails to press the button
- c.** Potential adverse effects:
 - i. Pedestrian enters street and collides with a vehicle
 - ii. Creation of a multi-vehicle collision while attempting to avoid the pedestrian
 - iii. Sighted pedestrian enters street because the WALK signal is on
 - iv. Low vision pedestrian is not able to cross the street
- d.** Detection method:
 - i. Incorrect *GetResponse* message to APC
 - ii. User reports malfunction to traffic agency

II. Failure to place a pedestrian call:

- a.** Description: Pedestrian presses the button but he or she never receives a WALK signal.
- b.** Possible causes:
 - i. Defective APB
 - ii. Connector from APC to cabinet ped call inputs is disconnected
 - iii. Partial APC electronics failure
 - iv. EoP communications failure
 - v. Power failure
 - vi. APC failure
 - vii. Pedestrian button failed
 - viii. Pedestrian fails to press the button
- c.** Potential adverse effects:

- i. Pedestrian crosses intersection with a red traffic light indication for parallel traffic movement. Result could be pedestrian – vehicle collision or creation of a multi-vehicle collision.
 - ii. Pedestrian enters crosswalk when parallel traffic has green indication but without WALK signal indication. Possible collision due to right turning or left turning traffic not realizing pedestrian is in the crosswalk.
 - iii. Low vision pedestrian is denied access to crosswalk and unable to cross the street.
- d. Detection method:
 - i. No *GetResponse* message from the APC
 - ii. User reports malfunction to traffic agency
 - iii. APC front panel LED red for failed APB
 - iv. Entry in APC event log

III.No audible output:

- a. Description: There is no audible locator tone and there is no audible WAIT message played when the button is pressed. However the call is placed and APB red LED turns on. There is no audible message when the WALK signal is on.
- b. Possible causes:
 - i. Partial electronics failure
 - ii. Incorrect configuration during setup
 - iii. Speaker failure
- c. Potential adverse effects:
 - i. Low vision pedestrian is denied access to crosswalk and unable to cross the street
 - ii. Loss of pedestrian confidence in pedestrian signals resulting in ignoring pedestrian signals
- d. Detection method:
 - i. User reports malfunction to traffic agency

IV.No button pressed LED on APB after press:

- a. Description: Red on APB fails to turn on after button press at local street corner or at corner at the other end of the crosswalk. Audio messages are played correctly and pedestrian calls are placed.
- b. Possible causes:
 - i. Partial electronics failure
- c. Potential adverse effects:

- i. Deaf or pedestrian with hearing loss will not know that call has been placed
- d. Detection method:
 - i. User reports malfunction to traffic agency

V. No communications between the APC and APB:

- a. Description: The APC and all APBs have power. No audio messages and APB red LED does not acknowledge call has been placed. WALK signal is served on every green phase.
- b. Possible causes:
 - i. Partial electronics failure
 - ii. RF interference or RF attenuation inhibits EoP signal
 - iii. APC fails to boot correctly
 - iv. APB fails to boot correctly
- c. Potential adverse effects:
 - i. Same as **II. Failure to place a pedestrian call**
- d. Detection method:
 - i. APC front panel LEDs all red
 - ii. User reports malfunction to traffic agency
 - iii. Entry in APC event log

VI. Failure to correctly detect a WALK or WAIT single indication:

- a. Description: WALK or WAIT signal voltages are not detected. Walk message is never played.
- b. Possible causes:
 - i. Low AC signal voltage
 - ii. Partial APC electronics failure
 - iii. Field wiring connection cable unplugged
 - iv. Broken wire or loose connection on field terminal
- c. Potential adverse effects:
 - i. If both WALK and WAIT signal voltages cannot be detected, APBs will go mute. Calls will be placed if button is pressed.
 - ii. Low vision pedestrian is denied access to crosswalk and unable to cross the street.
- d. Detection method:
 - i. User reports malfunction to traffic agency
 - ii. Entry in APC event log
 - iii. Web page status will indicate both WALK and WAIT signals are off

VII. No 12VAC power to APB

- a.** Description: Power conductors from APC to APB have no voltage.
- b.** Possible causes:
 - i. Short circuit on one or more 12VAC power conductors from APC to APB
 - ii. Short circuit protection resistors on APC termination board are open
 - iii. Broken conductor
 - iv. Broken terminal
- c.** Potential adverse effects:
 - i. Same as **II. Failure to place a pedestrian call**
- d.** Detection method:
 - i. The *GetResponse* message from the APC
 - ii. User reports malfunction to traffic agency

VIII. Power supply failure:

- a.** Description: System does not operate. No audible locator tones at APB stations. No power LED indication on APC.
- b.** Possible causes:
 - i. No cabinet power
 - ii. APC power switch in off position
 - iii. APC 120VAC circuit breaker open
- c.** Potential adverse effects:
 - i. Same as **II. Failure to place a pedestrian call**
- d.** Detection method:
 - i. User reports malfunction to traffic agency
 - ii. Unable to log in remotely

IX. APC fails to boot correctly

- a.** Description: APC fails to begin executing program. All front panel APB LEDs on APC are red. System not functional. Constant call placed on all ped call outputs. All APS are muted but have power.
- b.** Possible causes:
 - i. APC CPU failure
 - ii. Partial APC electronics failure
 - iii. EoP module fails to communicate with APB
 - iv. Software error
- c.** Potential adverse effects:
 - i. See items II through IV above
- d.** Detection method:

- i. The *GetResponse* message from the APC
- ii. User reports malfunction to traffic agency

X. APB fails to boot correctly

- a. Description: APB has power but does not play beacon tone or place calls.
- b. Possible causes:
 - i. Software error
 - ii. Partial electronics failure
- c. Potential adverse effects:
 - i. See items II through IV above
- d. Detection method:
 - i. The *GetResponse* message from the APC
 - ii. User reports malfunction to traffic agency