Maintainability Plan - TSV

ECE 492 - Spring 2015

Latest revision: 14 May 2015
Prepared by: Hansen Liang
Modified by: Katherine Nellis

Abstract

This document presents an analysis of the LFEV-TSV 2015 system to prove that the system can be repaired in the case of a failure in less than 1 week as specified by GPR007.

Table of Contents

Executive Summary

This maintainability plan details the measures and plans put in place to keep the TSV system maintained and repaired in case a common failure occurs in one of the system's parts. Since a well-designed system can be quickly repaired when needed, the TSV system will require less than one week to repair. We measure the ease of repair by using mean time to repair. In addition to the hardware maintainability, the software maintainability is also discussed in this document due to addition and improvement of software features for microcontroller and the AMS boards.

MTTR Analysis

MTTR → Mean Time to Repair
MTBF → Mean Time Before Failure

Pack Mechanical/Electrical

Lafayette College Machined Components

- MTTR (Spare Parts on hand): 1 day
- MTTR (No Spare Parts): 4-7 days manufacturing time + 1 day installation = 8 days

Note: Machine shop availability varies dramatically over the course of the year. Parts could be done in as little as one day if the shop demand is low or more than a week during high load periods. It is recommended to keep at least 1 spare component of the discharge path on hand in the event of a failure due to the unpredictability of the machine shop.

I2C/TWI LCD2004 Module (DFRobot)

- MTTR (Spare Parts on hand): 1 day
- MTTR (No Spare Parts): 7 days shipping + 1 day installation = 8 days

Power Lock Connectors (Newark)

- MTBF: 1120 Hours = 46.7 days (assuming continuous charging/discharge cycles and unplugging at each one)
- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 15 days shipping + 1 day installation = 16 days

Note: Since the MTBF is fairly low, failures may be common depending on the usage of the battery pack. Only one spare of each powerlock connector is necessary as parts can be ordered and delivered in less time than the failure rate.

24V Charging Fan (Mouser)

- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

AIRs and Charging Relays (Gigavac)

- MTBF: 2.22 Million Hours
- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 15 days shipping + 1 day installation = 16 days

Note: At least one spare relay of each type should be on hand in the event of a failure. Since the MTBF is so high, it is unlikely that multiple relays will break down before a new one can be ordered in, so one spare of each type should be sufficient.

Pack Plugs and Housings (Mouser)

- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

Anderson Charging Connectors (Mouser)

- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

Charging Fuses and Holders (Mouser)

- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

Note: GPR005 Requires at least 5 spare fuses of each type. A spare holder is not necessary.

Discharging Fuses and Holder (Zoro Tools)

- MTBF: 0.080 failures/million hours ≈ 0 failures/week
- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 15 days shipping + 1 day installation = 16 days

Note: GPR005 Requires at least 5 spare fuses of each type. A spare fuse holder is necessary to reduce the MTTR to under one week.

DIN Rail Terminal Blocks (Mouser)

- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

Battery Cells (AA Portable Power Corp)

- MTBF: 3350 hours (continuous charge/discharge) = 139.6 days
- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 22 days shipping + 1 day installation = 23 days

Note: Since the MTTR is 23 days without spare parts, it is advised to keep 7 spare cells. Since the cells in the battery pack encounter similar usage, their life cycle is likely to also end around the same time and cause all seven to fail very close to each other. Having an entire set ready to replace the dying cells will keep the MTTR under one week.

Pack Manager

Atmel - AT90CAN32 (Digi-Key)

- MTBF: Unknown
- MTTR (Spare Parts on hand): 1 day installation
- MTTR (No Spare Parts): 3-4 days shipping + 1 day installation = 5 days

PacMan Breakout Board (Advanced Circuits PCB)

- MTBF: 28.3 million hours
- MTTR (Spare Completed Board on hand): 1 day installation
- MTTR (Individual Component Failure, No Spare Part): 4 days shipping + 1 day installation = 5 days
- MTTR (Spare Parts on hand): 2 days manufacturing + 1 day installation = 3 days
- MTTR (No Spare Parts): 7 days PCB and part shipping + 2 days manufacturing + 1 day installation = 10 days

Note: In order to reduce the MTTR to under one week, it is advised to keep a completed spare PacMan breakout board on hand. PCB manufacturing is the critical link as it takes at least 7 days to produce. By contrast, if a singular component on the breakout board fails, parts can be ordered from Mouser or Digikey, shipped in about 4 days, and installed in one day. At the minimum all parts necessary to build the board (including the PCB) should be available.

AMS Boards

AMS Boards (Advanced Circuits PCB)

- MTBF: 95.8 million hours
- MTTR (Spare Completed Board on hand): 1 day installation
- MTTR (Individual Component Failure, No Spare Part): 4 days shipping + 1 day installation = 5 days
- MTTR (Spare Parts on hand): 2 days manufacturing + 1 day installation = 3 days
- MTTR (No Spare Parts): 7 days PCB and part shipping + 2 days manufacturing + 1 day installation = 10 days

Note: In order to reduce the MTTR to under one week, it is advised to keep at least one completed AMS of each cell configuration on hand. PCB manufacturing is the critical link as it takes at least 7 days to produce. By contrast, if a singular component on the breakout board fails, parts can be ordered from Mouser or Digikey, shipped in about 4 days, and installed in one day. Since the MTBF is so large, it is unlikely that two boards will fail before being able to build more space AMS boards. At the minimum all parts necessary to build the board (including the PCB) should be available.

Software Maintainability

PacMan Operating System Maintainability

It should be noted that as of May 14, 2015, the LFEV 2015 team has begun constructing the new library of C code for the BoB microcontroller; however, it is incomplete. The code programming the AT90CAN32 can be found both on the LFEV 2015 website and in the repository on https://bitbucket.org/ece492-sp15/pacman-software. The general functionality is there but extensive work needs to be done to establish CAN communication in accordance to the VSCADA protocol, as well as extensive testing of other classes. For maintainability purposes, the suggested code editor is AtmelStudio, which has, or can be easily downloaded to the AEC 400 desktop computers. In reference to a repository, members of the LFEV 2015 found best results using SourceTree, as opposed to the recommended TortoiseHg.

AMS Firmware Maintenance

The AMS firmware source code is available for download on the SP15 project website, and the instructions for programming the AMS boards can be found in the document **"PICkit3 Programming Manual"** also available on the website for download. In case any future code

modifications are to be made or any boards require programming/reprogramming, the document will guide the operation.

CAN Interface Maintainability

The LFEV-2015 TSV and VSCADA systems rely on the CAN bus interface to communicate with each other. Thus, it is of paramount importance that the compatibility between the systems be maintained throughout all changes and updates to both systems.

We ensure the compatibility of the CAN bus interface compatibility by adopting the same version of CAN library on the Linux systems on both sides. The VSCADA team uses Linux kernel 2.6.25 with CAN patches that support the same SocketCAN protocol.