

Lafayette College | Electrical and Computer Engineering

TSV Accumulator Maintenance Manual

ECE 492 Spring 2017

Assembled by Emilie Grybos, Meredith Guro, & Martin Townley

Overview	4
Principles of Operation	4
Advanced Maintenance	5
PacMAN	5
Firmware	5
Hardware	5
AMS Board	6
Firmware	6
Hardware	6
AMS I2C Communication	6
Cells	6
Dead Cells	6
OverCharged Cells	6
Replacing Cells	7
Pack	7
Safety Loop	7
Charging	7
Advanced Calibration	8
AMS Boards	8
Pack	8
Schematic links	9
Pacman	9
AMS	9
Control panel	9
BOM links	10
Mechanical Drawings	10
Wiring Diagrams	10
Block Diagrams	10
Pack	10
AMS	11
Gerber Files Links	11
QA Testing	11
Software	12
PacMAN	12

Binaries	13
Source	14
Makefile	14
AMS	14
Binaries	14
Source	14
Makefile	14
Toolchain	14
ATP -12 Completed	14

Overview

The purpose of this document is to give further insight into maintaining and debugging the “Packs” that provide Tractive System Voltage in LFEV-SP17. To see the User Manual (which includes high level information about the TSV system, basic debugging and maintenance) go to the following link:

<https://sites.lafayette.edu/ece492-sp17/files/2017/01/UserManual.pdf>

Principles of Operation

The TSV subsystem delivers the high voltage power in order to power the Formula SAE hybrid electric vehicle. Each pack or accumulator container, numbered 1-4, supplies around 24 V through seven LiFePO_4 battery cells which are all monitored through the accumulator management system (AMS). The AMS consists of a pack manager (PacMAN) computer and seven accumulator management system boards (AMS boards), one per cell. The PacMAN and AMS boards communicate via I2C. Cell voltage and Cell temperature are sent to the PacMAN to be displayed on the LCD. The four containers are wired in series to provide ~96 V in the assembled accumulator system. The subsystem interacts with other subsystems via data acquisition through CAN and via the vehicle’s safety loop.

All information regarding the TSV subsystem (schematics, wiring diagrams, memos, firmware/hardware, etc), some of which is linked in this document, can be found on the 2017 LFEV website on the TSV page. See: <http://sites.lafayette.edu/ece492-sp17/subsystems/tsv/>

Advanced Maintenance

PacMAN

Firmware

The Pacman is set up such that important parameters are configurable from the control panel and LCD screen. Most configurations (I2C addresses, calibration factors) are stored in params.h. Certain parameters (i.e. number of cells, pack ID for CAN communications) can be configured from the control panel using the pushbuttons and the LCD display. For more information on setting these configurations, see the LCD display navigation documentation

It is important to note that the current makefile preserves the eeprom when programming the board. If software revisions are made, and changes are made relevant to the EEPROM storage, then it might be beneficial to wipe the EEPROM while programming. In order to set the programmer to wipe the EEPROM clean while programming the board, the makefile must also be changed. To set the programmer to wipe the EEPROM, change the fuse section of the makefile from

```
lfuse:w:0xcf:m -U hfuse:w:0x91:m -U efuse:w:0xff:m  
to  
lfuse:w:0xcf:m -U hfuse:w:0x99:m -U efuse:w:0xff:m
```

It is important after altering the fuse setting in the makefile to use the command “sudo make fuse” to have the changes go into effect.

The software also controls the storing and displaying of the 5 most recent state transitions, and displays pertinent information to a recent fault. If a pack is faulting, or going into unexpected states, using this display may be useful for debugging.

Hardware

The crystal has presented itself as a major weak spot on the PacMAN. If the heartbeat LED is not blinking, it is most likely due to the flakiness of the crystal. We suggest creating solder bridges to its two neighboring capacitors to ensure a good connection. If need be, you may want to jumper wires directly to the microcontroller in order to create a sturdier connection. It would be ideal if a different crystal was chosen as this crystal is really not meant for hand soldering.

If the LCD in a pack displays that the PacMAN current is 0.00 A, while you know on a bench it's approximately 70 mA, you should check and or touch up the solder connection on U4. This is the Pack voltage and charge sensor (address 0x44 in I2C).

AMS Board

Firmware

Some AMS boards as of Spring 17 have an updated watchdog timer. These AMS boards will fail in the AMSVU unless they are reprogrammed with the original value as commented in the source files.

Hardware

The most likely point of failure on the AMS is U2. It is unknown why they fail, but when they do, they can pull 500 mA and overheat randomly. Replace U2 as needed on boards and attempt to find the root cause of the problem.

AMS I2C Communication

You cannot have two items on the I2C Bus with the same address. This will cause the board to act up and send into Fault 3.

Check both sides of U8. The HV and LV I2C lines should look the same, though HV is 5V peak and LV is 3.3V peak.

Use Embedded chip for Tektronix Oscilloscope to view the I2C bus on the AMS board side. View the Data it is sending is correct and not absurd. (00 indicates the PIC on the AMS board is asleep).

You can attempt decrease the on-chip watchdog timer within the “watchdog.c” file by decreasing the shift amount WDTPS. This has been used as a patch fix when AMS boards sporadically stop responding.

Follow the following guide for troubleshooting I2C communication. You may simulate a Master in order to see if the AMS boards (slaves) respond to a master that is not the PacMAN.

<http://sites.lafayette.edu/ece492-sp14/files/2014/02/AMS-I2C-RealTerm-Memo.pdf>

Cells

Dead Cells

If a Cell reads < 1V, attempt charging the individual cell to 3.4V. If it maintains this charge, you are good to go. If it can't maintain a charge, you need to replace the cell.

OverCharged Cells

If a cell has been overcharged, (4V or above), try directly discharging the cell through the big green power resistors in 400. Each green resistor is 1Ω, wire them all in parallel. Let the overcharged cell discharge to 3.4V. Check back on the cell in an hour to see if it has not risen back to 4V. If it has again risen to an overcharged voltage, you need to replace the cell.

Replacing Cells

See this document on the LFEV 2017 website for the procedure to replace cells:

<https://sites.lafayette.edu/ece492-sp17/files/2017/01/CellReplacementProcedure.pdf>

Pack

Safety Loop

If the Safety Loop is not closing as expected, check to see if the 37 pin connector has been screwed in all the way. It has been known to be an unreliable connection unless tightened as much as possible.

Also, ensure that all wires in that

Charging

If a Pack is unable to charge, check the continuity from the charging port to the connectors that insert into the pack man. The connectors are known to fail from normal use.

You may also see if the PacMAN charging circuit works by shorting the two leads of P6 together and observing the Charge LED turn on as well as the charge relay click to ensure the PacMAN is not the problem.

If for some reason SOC indicates 100% after the pack has been discharged, you can manually force the pack's SOC to 50%. To do so, disconnect the AMS boards from the PacMAN. Press reset on the control panel. Reconnect AMS boards. Press reset on the control panel. SOC should now be at 50% and the pack can be charged.

Advanced Calibration

AMS Boards

Calibration of the AMS board is done through the AMSVU. Follow this document to Calibrate the AMS boards: <https://sites.lafayette.edu/ece492-sp17/files/2017/01/AMSVU-User-Manual.pdf>

You may also adjust the WatchDog timer on the PIC to decrease the amount of time an AMS board is asleep. This decreases the likelihood that an AMS board will send 0x00 as data when it should not.

Pack

Be sure to set the Pack ID and number of cells in order to see accurate CAN data. In order to do this, the configuration parameters menu must be used. Press enter from the main screen to go to the configuration parameters menu. There are multiple parameters that are stored in the EEPROM that can be configured for this menu. To configure a parameter, first its address must be entered, followed by the desired value. Below is a table of the parameters, their addresses, and other relevant data:

Config Address	Config Parameter	EEPROM addr	Size (Bytes)	Default Value	Units
0x01	pack id	0x08	1	0x01	n/a
0x02	# cells	0x09	1	0x07	cells
0x03	lock state	0x02	1	n/a	bool
0x04	max cell temp	0x03	1	0x3C	degrees C
0x05	max cell voltage	0x04	1	0x28	tenth volts
0x06	screen reset time	0x05	1	0x0F	seconds
0x07	screen sleep time	0x06	1	0x78	seconds

It is important to note that the parameters can only be edited if the pack is “unlocked.” The lock state is stored at config address 3, and can be toggled by entering the “password” as the value to edit. THE PASSWORD TO TOGGLE THE LOCK STATE IS 0xFF.

The figure below demonstrates how to navigate the menus on the lcd screen using the control panel on the packs:

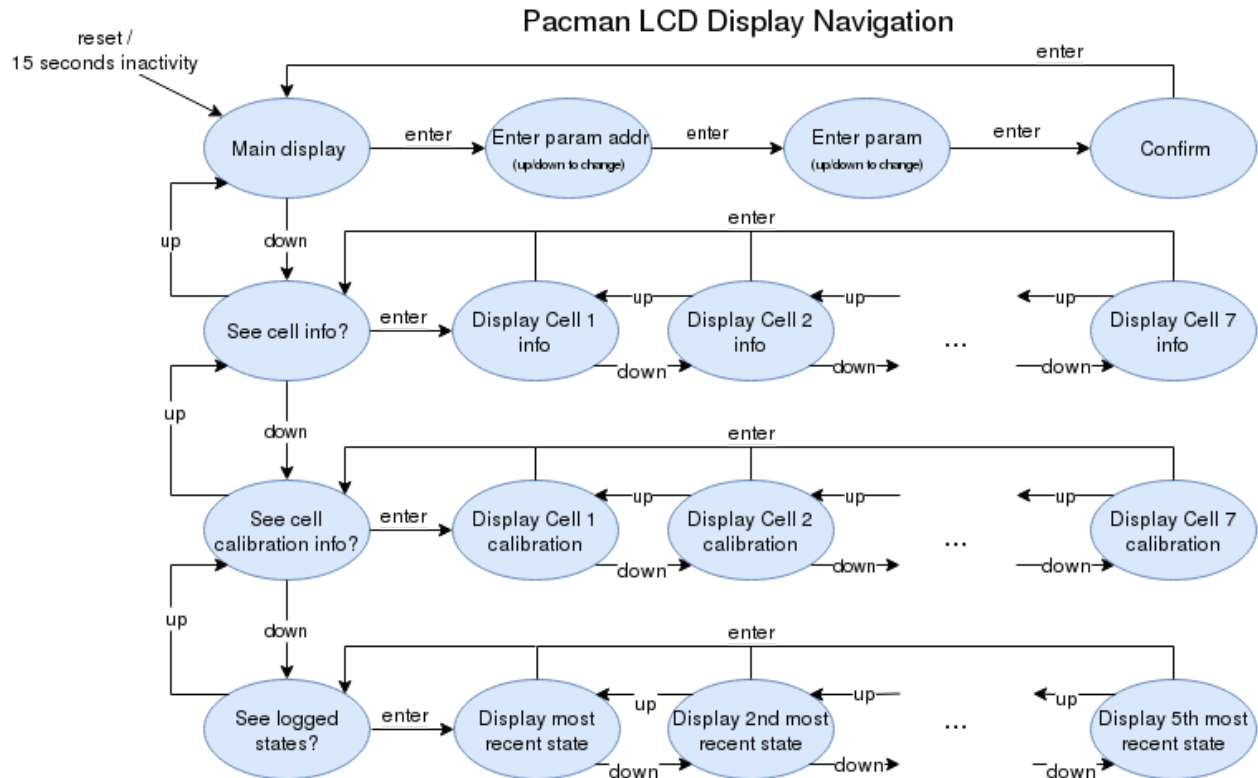


Figure 1. PacMAN LCD Display Navigation

Schematic links

Pacman

Below is the Rev 0.8 Schematic which is used in the LFEV-17 implementation of the Accumulator. This is also the Schematic for Rev 0.9, as the only difference between R0.8 and R0.9 is the PCB layout.

https://sites.lafayette.edu/ece492-sp17/files/2017/01/PacMAN_Rev_0_8_Schematic.pdf

AMS

Below is the Rev 0.4 schematic of the AMS boards:

https://sites.lafayette.edu/ece492-sp17/files/2017/01/AMS_Schematic_Rev_0_4_2016.pdf

Control panel

<https://sites.lafayette.edu/ece492-sp17/files/2017/01/panel-pcb.pdf>

BOM links

Pacman <https://sites.lafayette.edu/ece492-sp17/files/2017/01/PacMAN-BOM.pdf>

AMS https://sites.lafayette.edu/ece492-sp17/files/2017/01/AMS_BOM.pdf

Control panel <https://sites.lafayette.edu/ece492-sp17/files/2017/01/Control-Panel-PCB-BOM.pdf>

Pack <https://sites.lafayette.edu/ece492-sp17/files/2017/01/Accumulator-BOM-Pack-BOM.pdf>

Mechanical Drawings

To view the mechanical drawings of pertinent parts in a pack, download the following zip file:

<https://sites.lafayette.edu/ece492-sp17/files/2017/01/Mechanical-Drawings.zip>

Wiring Diagrams

See the Accumulator Interconnect Document for internal TSV wiring information at this link:

https://sites.lafayette.edu/ece492-sp17/files/2017/01/TSV_ICD.pdf

Block Diagrams

Pack

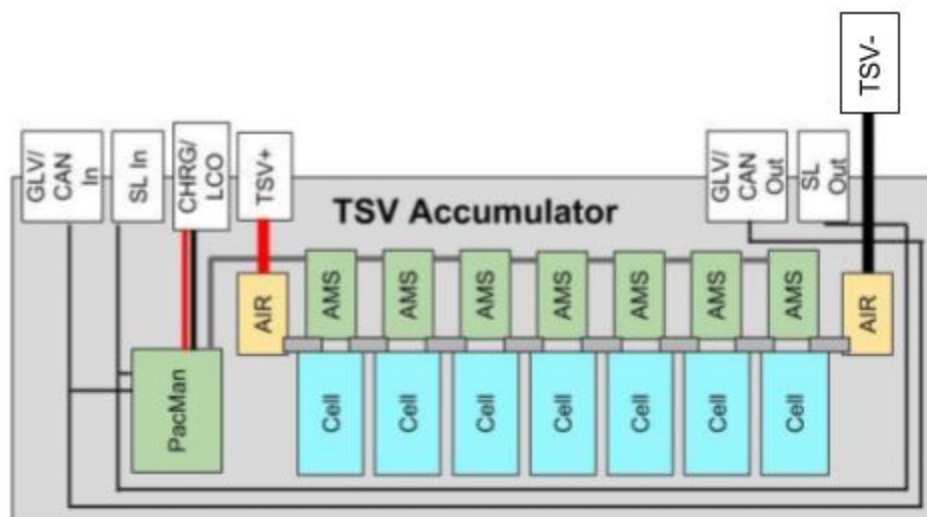


Figure 2. Block diagram of a Pack in the TSV Accumulator. Taken from LFEV-SP2017 Final Presentation.

AMS

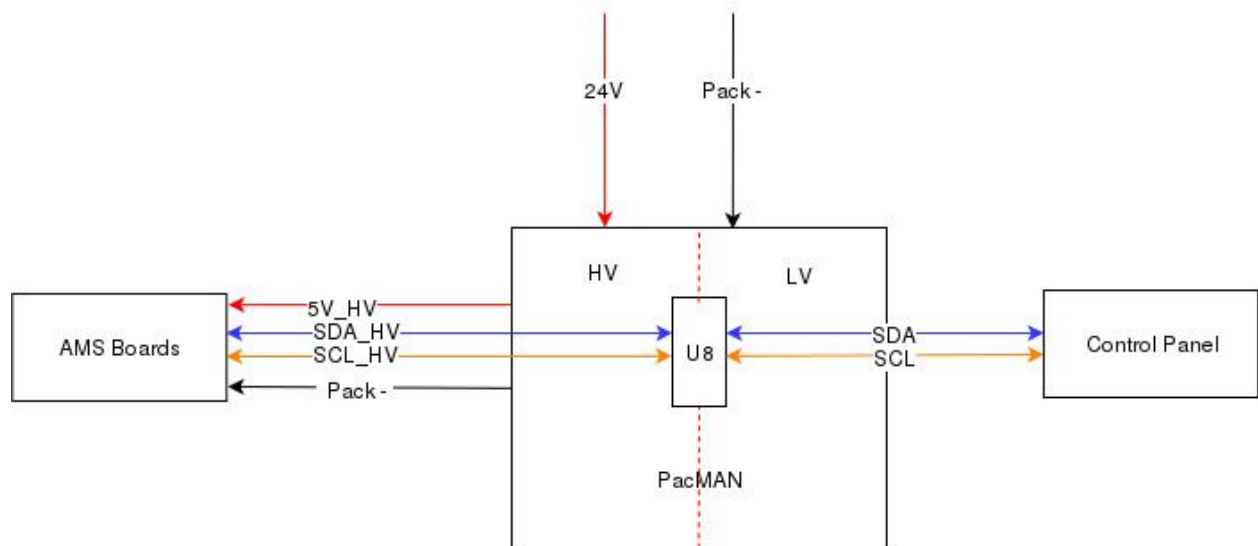


Figure 3. Block Diagram of I2C links in AMS

Gerber Files Links

PacMAN https://sites.lafayette.edu/ece492-sp17/files/2017/01/L17-TSV-1_hardware_rev_0_8.zip

AMS Board https://sites.lafayette.edu/ece492-sp17/files/2017/01/AMS_PCB_Layout.zip

Control Panel https://sites.lafayette.edu/ece492-sp17/files/2017/01/L16-TSV-3_KiCad.zip

QA Testing

To see all QA results see

https://sites.lafayette.edu/ece492-sp17/files/2017/01/Accumulator_Memo_1.zip

<http://sites.lafayette.edu/ece492-sp17/testing/atp/qa-testing/>

<http://sites.lafayette.edu/ece492-sp16/test-result-documents/>

Software

PacMAN

The current version of Pacman software is v0.16 and the source code can be found at:

https://sites.lafayette.edu/ece492-sp17/files/2017/01/PacMAN_Software_0_16.zip

It is written in C for the ATMEL AT90CAN32 microcontroller. The documentation for microcontroller can be found here: <http://www.atmel.com/Images/doc7679.pdf>

The tool chain is the same as used in previous years and more details about Pacman programming can be found in the programming manual, located here:

https://sites.lafayette.edu/ece492-sp17/files/2017/01/PacMAN-Programming-Manual_2017.pdf

The software is built on the Atom Threads RTOS. Atmel TWI and CAN libraries are utilized to achieve communication.

The code in main.c sets up tasks listed in tasklist.c and starts the RTOS. Functions that generate LCD screens are detailed in lcd.c. Functions that utilize TWI libraries to perform I2C communication are detailed in i2c.c. The remaining c files detail tasks that run continuously:

- task_button.c retrieves button presses on the control panel
- task_can.c transmits CAN messages
- task_charge.c performs integration of current and calculates state of charge
- task_config.c performs state transitions
- task_gui.c calls function in lcd.c to set the display output
- task_heartbeat.c blinks an LED on PacMan to indicate the computer is operating
- task_i2c.c calls functions to perform I2C communication tasks
- task_safety.c opens and closes the safety loop relay
- task_watchdog.c resets off chip watchdog

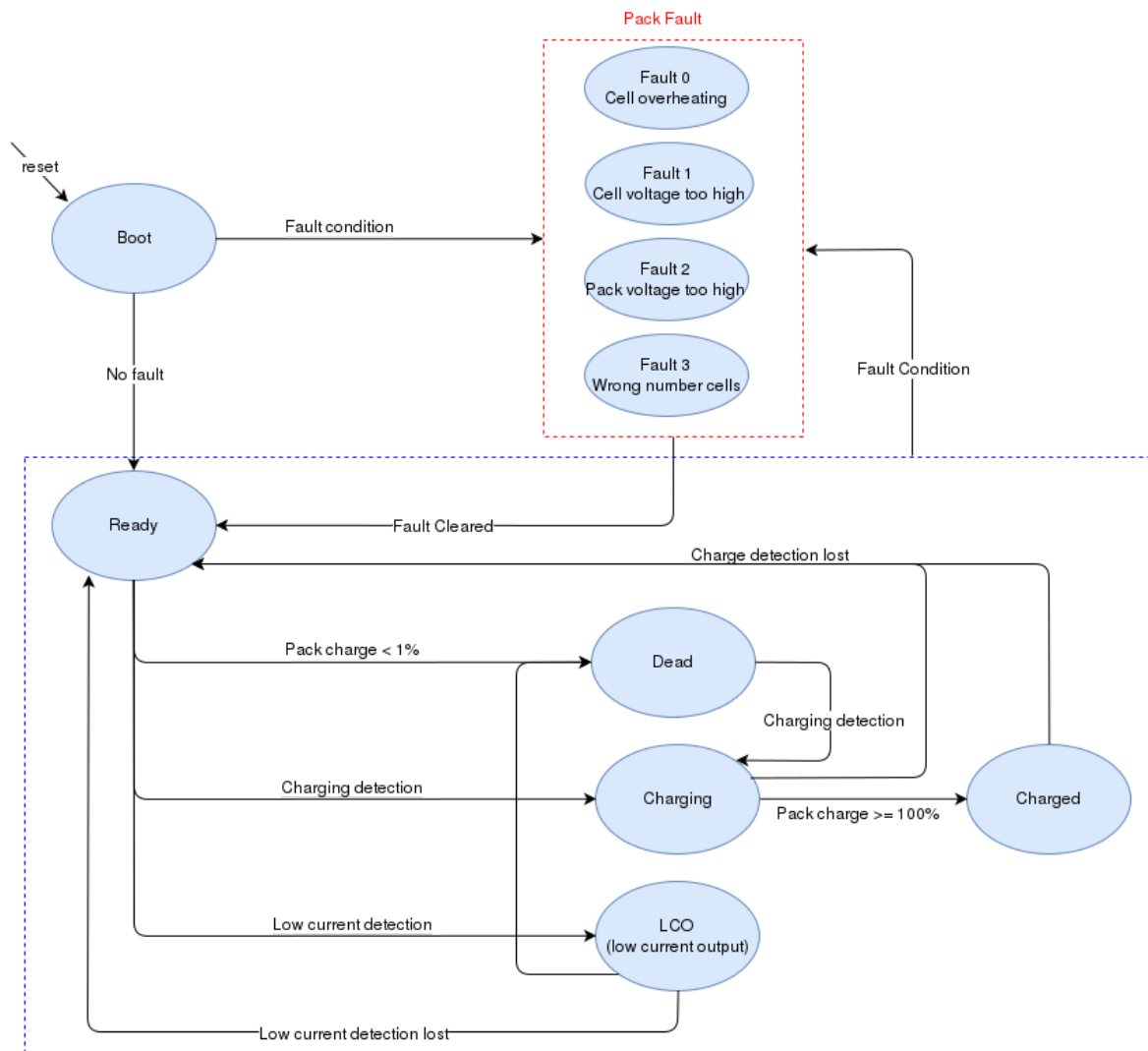


Figure 4. PacMAN State diagram.

The pacman is capable of storing and displaying the 5 most recent state transitions, and can display pertinent information to a recent fault. If a pack is faulting, or going into unexpected states, using this display may be useful for debugging. Navigating the control panel displays is further outlined in the control panel section.

Binaries

https://sites.lafayette.edu/ece492-sp17/files/2017/01/PacMAN_binaries.zip

Check the git for the most up to date software; this contains software as of 5/4/2017.

Source

<https://github.com/LafayetteFormulaElectricVehicle/TSV>

Makefile

All Make Files are in their respective folders. See the ReadMe for descriptions of the different versions of Software.

AMS

Binaries

https://sites.lafayette.edu/ece492-sp17/files/2017/01/AMS_binaries.zip

Source

https://github.com/LafayetteFormulaElectricVehicle/TSV/tree/master/AMS%20Software%203_31

Makefile

Makefile located in the “firmware” folder.

Toolchain

Requires MPLab IDE and the PIC programmer. See the AMSVU user manual for more information.

ATP -12 Completed

Look at ATP-12 on the ATR

<http://sites.lafayette.edu/ece492-sp17/testing/atr/>