TRACTIVE SYSTEM INTERFACE MAINTENANCE MANUAL

ECE492 Spring 2017

LAFAYETTE COLLEGE

TABLE OF CONTENTS

INTRODUCTION

This document is intended describe the current design of the Tractive System Interface and its intended use. Future improvements, as well as errata, will also be covered so that future students may understand the pitfalls we encountered, as well as our perceived path forward for the board. This document is based off of the LFEV 2017 team's TSI maintenance manual.

The TSI motivation was to accomplish the following tasks. The current subsystem is designed to meet each of these requirements.

- TSV Power Deliver
    - Precharge relay closes on motor controller startup
- Throttle
    - Determine throttle plausibility
    - Isolate (plausible) analog throttle signal and pass to motor controller
    - Throttle level input to microcontroller
- Brake System
    - Brake overtravel switch opening safety loop
    - Brake pressed signal read and used in system state logic
    - Brake light illuminating when brake pressed
- IMD
    - Read IMD PWM corresponding to resistance measurement
    - Open safety loop if IMD fault detected
- Voltage/Current measuring
    - Read voltage and current values from TSV
- Shutdown System
    - State diagram according to Formula Hybrid rules
    - Drive Button
    - RTDS
    - Drive light
- Tractive System Active Lights
    - TSAL and cockpit HV present light powered on when TSV is present
    - Lights powered from DC/DC isolator
- CAN Communication
    - TSI sends data back to SCADA (TSV voltage and current, throttle plausibility, throttle level, IMD status, brake pressed)
    - CAN isolator connects motor controller to can network for data collection/debugging

SCHEMATIC

This schematic itself contains many comments and annotations. Additionally, each component should contain a sizable amount of information. This includes footprint, datasheet, and DigiKey part number, among other characteristics. Here we will discuss the reason behind our component selection, as well as their intended functionality.

MAIN SHEET

AIRS MEASUREMENT

This optoisolator takes in the +24V signal from the AIRs, and outputs a Safety_Loop variable to PIN7 of the microcontroller. This may be used to determine the state of the safety loop, and cause the system to fall out of the Drive State.

IMD PWM LPF

The IMD outputs a PWM signal that corresponds to the resistance seen across the TSV. This filter was to produce an analog voltage that could then be read by the microcontroller (PIN60, ADC), and passed to VSCADA over CAN. At the conclusion of ECE492-SP18, the PWM is read by the ADC and passed to SCADA.

CURRENT MEASUREMENT

The current sensor we are using produces a differential output, that we then put through this unity-gain differential amplifier to produce a single-ended output. The signal, I_Measure, then goes into PIN58 of the microcontroller (ADC) to be passed along to VSCADA over CAN. The current measurement is also to be used to detect if the current is reaching the threshold of the fuses.

At the conclusion of ECE492-SP18, this has been implemented but several changes will need for next year. The current sensor outputs are currently wired backwards, this has been manually corrected. The current sensor is in the high voltage domain, so the connector will need to be moved to the high voltage side of the board and isolated. As currently configured the single ended output of the diffamp only outputs for positive current. This should be rebiased to measure positive and negative current.

TEMPERATURE SENSOR

The temperature sensor communicates with the microcontroller over SPI. At the conclusion of ECE492-SP18, the temperature is correctly read and passed to SCADA.

CURRENT ENABLE

There were concerns over the current draw of the Current Sensor. To fix this, we added a low-side switch so that the microcontroller (PIN44) could power on/off the sensor. The sensor is only turned on when the safety loop is closed. At the conclusion of ECE492-SP18, this has been successfully implemented.

SPARE ADC SWITCH

The intention of this switch was to give the user spare ADC's for unforeseen future use. The control signals of the microcontroller all exist on the same pins as the ADC's. Therefore, this switch may be toggled after programming in order to free up these additional pins.

ISOLATORS

THROTTLE

This isolated differential amplifier passes the low-voltage throttle signal (Throttle_LV) to the high voltage side of the board. The voltage divider on the input is intended to bring the voltage down to a 0-2V range. On the high voltage side, the op-amp has a gain such that the signal is back to the 0-5V range used by the motor controller. Should also be noted that this configuration is based upon the application described in the part datasheet (here: https://docs.broadcom.com/docs/AV02-3563EN). At the conclusion of ECE492-SP18, this has been successfully implemented.

VOLTAGE MEASUREMENT

This is the same isolator as the one found in the Throttle Isolation circuit described above. The difference here is the voltage divider on the input, intended to divide a 120V TSV to a 0-5V range. The isolated output, V_Measure, then goes to the microcontroller (PIN59) where it is read by an ADC, scaled to a 0-96V scale, and sent to VSCADA. At the conclusion of ECE492-SP18, this has been successfully implemented.

CONNECTORS

The previous ribbon cables have been replaced by Molex connectors. Each Molex connector corresponds to an external Deustche connector. The connectors were laid out in a way that each receptacle can only be connected to the corresponding board connector to avoid wiring errors. J10 should be changed to a 6 pin connector identical to J7 and J8 to provide power to the CAN isolator. The remaining connectors for the JTAG programmer, High Voltage connections, as well as spare pins functioned simply and securely.

The overall system wiring diagram should be addressed for the pinout of the external connectors.

DC/DC CONVERTER

The DC/DC Converter works as intended, and may be confirmed by viewing the PWR LED next to the microcontroller, which turns on whenever 5V is supplied. In future runs, it is recommended that this converter change to a dual output. This is due to the 24V-to-10V regulator (LM7810). This regulator needed to create 10V referenced from the CH_GND, however, since it was regulating from the 24V coming into the DC/DC, isolation had to be sacrificed. This resulted in the +24V_RTN and CH_GND pins being soldered together on the J7 connector. This may not be possible as the IMD ground connections might not be isolated.

CAN TRANSCEIVER

The CAN Transceiver is identical to the one found on the PacMan boards. At the conclusion of ECE492-SP18, this has been successfully implemented and CAN transmission and receiving was implemented.

THROTTLE PLAUSIBILITY

The majority of the circuitry contained in this sheet was created to comply with the FSAE Hybrid Rules for Throttle Plausibility checks.

10V REGULATOR

This is the regulator discussed above in *DC/DC Converter*. Due to the necessity to create the 10V from the 24V going into the DC/DC, ground isolation was sacrificed. If a dual-output DC/DC is used to replace the current one, creating 5/12V output, this regulator will have to be replaced with a low-dropout equivalent.

APPS 5V OFFSET BIAS

The purpose of these two resistor ladders is to bias the two pedal potentiometers 5V from one another. A potentiometer was added to each offset resistor ladder to accommodate for the poor throttle pot tolerance. In order to properly bias the throttles, a voltmeter should be connected across APPS1_10 and APPS1_RTN and RV2 should be adjusted so the meter reads 5V. To bias APPS2 a voltmeter should be connected between APPS2_5 and RTN_GLV and RV1 should be adjusted so the meter reads 5V. This biasing method should be improved on the next iteration of the board. A window comparator with an LED for feedback would be a good replacement for the multimeter.

The APPS1 and APPS2 signals are then buffered to keep the load from disrupting the throttle pot voltage dividers. 100k resistors were added to drain the buffer decoupling capacitors in the event a throttle pot wire is disconnected. These will need to be added to the board.

APPS1 STEP DOWN

This unity gain differential amplifier is used to step the 5-10V APPS1 signal to a 0-5V range for use in the window comparators. This was chosen as the method to step the voltage down, since the MCP6004 uses all four available op-amps on the window comparisons, while we still had one free TS912 on U18.

OPEN/SHORT WINDOW

These window comparators check for open/short circuits coming from the APPS. They currently allow for roughly 93% of the travel to pass without causing an implausibility. If APPS1_ISO or APPS2 is less than 0.25V or greater 4.75V the throttle will be implausible. This is to prevent a throttle signal wire being shorted to power or ground or a pedal mounting failure from causing the motor controller to receive an improper throttle voltage.

DIFFERENCE WINDOW (0.5V)

The differential amplifier finds the difference between the two buffered APPS inputs, then uses a window comparator to check that they are within 0.5V of each other. If the difference is outside this range an implausibility occurs.

THROTTLE PLAUSIBILITY OUTPUT

If all throttle plausibility checks are OK, U8 outputs low else it outputs high. If the throttle is implausible a power mosfet grounds the APPS2 signal. If it is plausible then the throttle signal is sent to the throttle select switch.

THROTTLE SELECT SWITCH

This analog switch allows whichever throttle signal is manually set with the jumper (JP1). The Throttle_SEL signal has a pull-down resistor to ensure that GND is selected upon startup. This switch allows the microcontroller to drive the throttle value to zero based on drive state, or other input signals.

BRAKE PRESSED

The optoisolator is used to step the 24V signals down to 5V for use with the microcontroller. The microcontroller is able to read if the brake has been pressed and is used for determining the state of the car. At the conclusion of ECE492-SP18, this has been successfully implemented.

The brake signal is powered by the AIRs voltage to comply with the safety loop rules. The airs voltage passes through a brake pressure switch before being sent to the optoisolator and brake light. Unfortunately, the brake pressure switch has an open resistance of 15K. To account for this a 500 ohm 5W pull down resistor was added across the brake light. This causes the optical isolator and brake light to be driven at too low of a voltage to turn on. The brake light is only rated for 12V, so a 12V Zener diode was added to drop the voltage.

STATUS LIGHTS

RTDS/DRIVE LED

These are the same optoisolators used throughout the rest of the board. The three control signals (PIN47, PIN48, and PIN49) from the microcontroller are used to alert the driver to the status of the system inside the cockpit. When the car enters drive mode the ready-to-drive signal is played for two seconds, in accordance to FSAE rules. In its current configuration the RTDS is current limited by the optical isolator and does not sound loud enough. This can be remedied by using a power mosfet to drive the RTDS. The drive LED turns on when in drive mode and blinks in overcurrent mode.

IMD STATUS SWITCH

This high-side switch was used due to the inverted OK output of the IMD. The IMD board drive a relay found in the safety loop

STATUS/DEBUG LEDS

Five LEDs are placed on the board for debugging/system state tracking of the board. These may be changed/reconfigured using PIN36, PIN37, or PIN38 of the microcontroller. The bottom LED turns on when there is a hardware implausibility.

TSAL

A Murata RUW15SL12C is used to step down the 96V TSV to 12V to drive the TSAL. This converter has a minimum stable output of 0.125A so a wirewound resistor was added to draw the minimum amount of current.

PCB LAYOUT

The PCB functions as currently fabricated, however, there are several changes that should be made before another board is ordered. These changes can be found in the errata.

FIRMWARE

The git repository for the TSI firmware can be found here: https://github.com/austinmam/TSI

Programming instructions may be found in our Git here:
https://github.com/austinmam/TSI/blob/master/PacMAN-Programming-Manual_2017.pdf

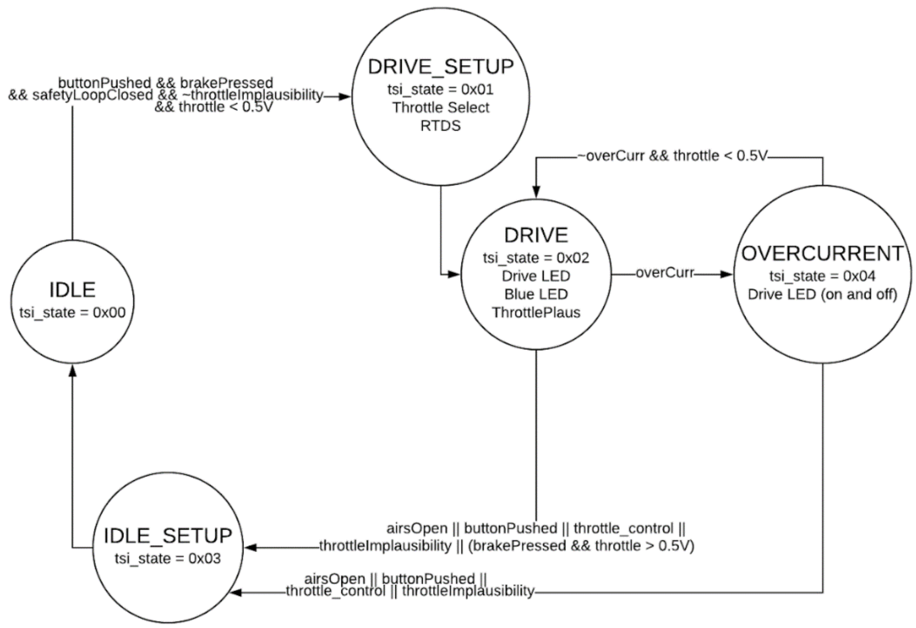
The Makefile, and tasklist.c need to be altered to add/modify tasks.

The following files are used in the TSI firmware:

- main.c
- params.h: Used to create global variables
- task_button: Reads when the drive button has been pressed
- task_can: Creates packets to be sent over CAN to SCADA
- task_can_receive: Creates packets to be received over CAN from SCADA
- task_heartbeat: Blinks heartbeat LED to indicate power to TSI board
- task_overcurrent: Determines if current is close to fuse threshold
- task_readApps: Reads the voltage from the throttle
- task_readCurrent: Reads the current from TSV
- task_readIMD: Reads IMD PWM signal
- task_readVoltage: Reads voltage from TSV
- task_safety: State machine control of the car
- task_temp: Reads temperature of board
- tasklist: Determines which tasks are to occur during runtime

At the conclusion of ECE492-SP18, most major features have been implemented, although further modifications can be made. These changes will be outlined in the errata.

A state transition diagram is shown below:



ERRATA

The purpose of the errata is to outline changes and modifications that can be made in further years.

HARDWARE

Schematic

- For isolation U13 should be a dual output 12/5V regulator and an LDO should replace U5.
  - If not CH_GND should be connected to +24V_RTN, they are currently jumpered on the board to establish this
  - The CH_GND should be disconnected from the ground plane
  - May not be possible if the grounds are not isolated on the IMD
- APPS2 microcontroller input should be taken from buffered signal
- R16, R18, and R19 should be removed
- A1_LV should go to R4 and A2_LV should go to R3
- 96R (2W or greater) should be added across pins 7 and 8 on U1
- Throttle plausibility
  - Add 100k drain resistors for APPS1 and APPS2 inputs

    - Each plausibility check should have a way to be debugged easily
- Allow for setting overcurrent threshold using spare ADC (pin 54 or 55) with current limiting resistor (100k or greater)
    - Can be accomplished a number of ways
        - One option is use resistive dividers between 5v and gnd of different values with a pin header and jumper for selection
        - Another is to use a pot with pins 1 and 3 across 5v and gnd and the wiper going to the uC
- Current sensor fixes
    - Connector should be on high voltage side
    - Diffamps should be biassed to +2.5V so positive and negative currents can be measured
    - Will need to be isolated
- Add hardware to show correctly biased throttle pots
- RTDS
    - Optical isolator doesn't provide enough current to run it loud enough. A power MOSFET should be added
- Debugging
    - All debugging LEDs should be located in one place
    - Add LEDs to show each throttle plausibility condition.
    - They should be visible through a window/clear cover so the cover doesn't need to be removed to diagnose problems

Layout

- Footprints
    - The pad for C3 should be made larger to fit the Cap
    - C3 should be moved further from the J13 connector
    - Add test points for bias voltage measurement
    - Pins 6 and 8 should be switched on U1 footprint
    - Possibly move components farther from U1 for easier repairs
- Connectors
    - Mounting holes for J9 need to be made larger
    - Move J9 to opposite side of the board
    - J13 for CAN Isolator should be a 6 pin connector for power
        - Same pinout as other 6 pin connectors
- Mask
    - Label switch and jumper positions
    - Label adjustment pots

Enclosure

- ADD EXTERNAL PROGRAMMING PORT

Components

- Try and standardize components with other subsystem boards (ie stick to 2 or 3 kinds of opamps, all 0805 smd passive components, same molex connectors)

FIRMWARE

- The biggest feature that was not properly implemented was the current sensing. The ADC is able to read the incoming signal, but the reading is incorrect. This will need more debugging in the future.
- Due to the current sensor not coming to fruition, the overcurrent state was never properly debugged. The intent was that when the current is too high and there was a chance of blowing one of the fuses, the car would enter the OVERCURRENT state where the throttle would be deactivated until the current is back below a certain level.
- Remove line 93 in main.c. This line sets all pins on port A as an input even though some of these pins are used as outputs. We did not notice this until competition as the firmware behaved correctly a majority of the time, but could be the cause of some erroneous behavior.
- Current schematic shows the Drive Button is hooked up to pin PA5. However, this had to be changed to pin PE3 during competition.