Aero-Sensors Module

For: Aero Sub-System Validation

Lead: Philippe Grimard
Designer: Joshua Lafleur

Date: July 31st, 2022

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1. DESIGN AIM

The goal of the aero-sensors module is the validation of CFD methodologies. This will confirm or deny the current modeling practices within the aero-subsystem. It will also aid in developping more accurate and precise modelling techniques for future car and team generations.

The module will need to record various environment variables and save them to medium and long term storage. The variables in the initial implementation of the software to be logged inlcude absolute pressure sensors, air speed sensor, air temperature sensor, vehicle velocity, and linear potentiometers. As the scope of the variables is dynamic, additional scalability should be considered for robust long term support of changing data logging objectives.

The absolute pressure sensors will be positioned across various airfoils as pressure taps to record various key points. The air speed sensor as well as the air temperature sensor will be positioned in free-flowing air to get the least noisy measurements. The air speed sensor will also need to be pointed in the direction of travel of the vehicle, ideally along the centerline. Linear potentiometers will be placed at all suspension springs to record the compression forces of each spring and of the total system. Vehicle velocity will be recorded from the CAN bus.

Data storage will be supported through both long term on an SD card as well as supporting interfacing and data logging to the CAN bus. The SD card will be native to the main board of the module.

User input and output will be through a button and a RGB LED. The button will allow the user to start a run and stop a run. By stopping a run, the module will prepare itself by creating a new run and waiting for the user to start it. The RGB LED will show RED if the module encountered an error, GREEN if the module is ready to start a run, and BLUE if the module is recording a run.

2. ELECTRONIC COMPONENTS

Ideal Quantity	Model Number	Purpose
16	MPRLS0025PA00001A	Absolute pressure sensor over I2C
1	NPA-700B-030D	Differential pressure sensor and temperature sensor over I2C
1	STM32F103C8T6 “Blue Pill”	MCU development board
1	SD card slot – JLCPCB SD-112	SD Card holder
2	MAX7356EUG+	1x8 I2C multiplexer
1	TJA1050	CAN transceiver
4	Linear Potentiometer	Measure spring compression
1	Push Button	User Input
1	RGB LED	User Output

Table 1: Main Electronic Components

3. TECHNICAL SPECIFICATION

The module will record the values at a frequency of 100Hz with aid from an RTOS. Each recording will be saved between measurements so that large amounts of RAM will not be needed during runtime. The recordings will be saved in chunks to a Fat filesystem with support from FatFS.

Since there will be a large number of the MPRLS0025PA00001A sensors each sharing the same I2C address, a MAX7356EUG+ I2C multiplexer will be on each of the Blue Pill I2C busses to facilitate support for upto 16 sensors. Additional connectors will be available on each of the I2C busses to support more sub-modules with unique addresses.

The MPRLS0025PA00001A does a conversion once prompted by I2C command, and takes upto 5ms for the conversion to be ready. The NPA-700B-030D continuously measures the differential pressure and temperature of its environment, and has data ready to be polled at the applications discretion. The linear potentiometers are analog devices which continuously output their measurements for polling by the Blue Pill ADC.

The daughter boards of the MPRLS0025PA00001A are 18.40mm long and 15.80mm wide. The NPA-700B-030D daughter board is 24.15mm long and 15.3mm wide. Both daughter boards do not have machine holes for mounting due to designing for small size applications, and are therefore to be mounted into rails with a retaining clip.

The connectors to be used thoroughout the module are part of the Molex KK 254 (also Molex 6410) family of connectors. Due to the large number of connectors, connector locking mechanisms were deemed less important than the cost savings of a less robust but more cost effective connection system. Long term considerations of this should be kept and if un-mating forces are too low for the environments faced by this module, the connectors can be glued together. This is a small note and deemed unimportant as this module is a testing and verification module and not a critical system.

4. SOFTWARE SPECIFICATION

The entire codebase will be hosted on the concordia-fsae/firmware github repository in the long term stable build system. The entire PCB technical package will be hosted in the concordia-fsae/circuits github repository. The current PCB technical package is built in KiCad v6.0.2 release build.

The Blue Pill is to be run towards the top end of its maximum frequency at 64 Mhz to allow for quick execution in this time sensitive application. It will run off a high speed external crystal which is internal to the Blue Pill board. The APB1 and APB2 busses will be run at 32 and 64 Mhz respectively for support of high speed periphals. The system SYSTICK will be run off of a dedicated timer internal to the STM32 MCU.

The module will poll the user input button at a frequency of 10 Hz to deduce whether or not the measurement state machine is to be changed. The description of the state machine can be seen in Appendix I.

The system will continuously convert the raw inputed analog volatages from the linear potentiometers and store them temporarily in an array of multiple samples. At the end of each sampling run, the system will filter the analog readings into a single less noisy reading for each of the potentiometers.

Every 10ms, the RTOS will call a sensor measurement function which will record the current timestamp, start the measurement of all the MPRLS0025PA00001A sensors, record the air speed and air temperature, and finally wait for the MPRLS0025PA00001A sensors to have data ready. While the system waits for the MPRLS0025PA00001A sensors to end their conversion which can take upto 5ms, the system will save the last recording to the SD card.

5. MANUFACTURING AND COMPONENT COSTS

The manufacturing will be done in the three following steps:

1. PCB manufacturing with JLCPCB
2. PCB assembly of select components with JLCPCB
3. PCB assembly of select components internally

Designator	Comment	Footprint	LCSC
C1	0.1u	402	C307331
C2, C3	1uF	402	C52923
C4, C5	10nF	402	C15195
C6, C7	2.2uF	402	C12530
H1, H2, H3, H4	MountingHole_Pad		
J1	SD-112	SD-SMD	C961673
R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R17, R18	2.2k	402	C25879
U1, U2	MAX7356EUG+	TSSOP-24	C2150941
U4, U6, U7, U8, U9, U10	MPRLS0025PA00001A	LGA-12	C3041242
U5	TJA1050	SOIC-8_3.9x4.9x1.27P	C6952
U18	LP2985-5.0	SOT-23-5	C74511
U19	LP2985-3.3	SOT-23-5	C89826

Table 2: Components to be Assembled by JLCPCB. Items in red are experiencing inventory shortages.

Designator	Comment	Footprint	LCSC
U4, U6, U7, U8, U9, U10	MPRLS0025PA00001A	LGA-12	C3041242
U1, U2	MAX7356EUG+	TSSOP-24	C2150941
U17	NPA-700B-030D	SOIC-14	
U3	BluePill Breakout Header	2.54mm Female Header 20 pin	
J2, J3, J4, J5, J6, J7, J8, J15, J16, J17, J18, J19, J20, J21, J22, J23, J24, J25, J26, J27, J28, J29, J30, J31, J32, J33, J34, J35, J36, J37, J38, J39, J40, J41, J42	DigiKey: 0022272041, 0022122044, 0050579504, 0008510108	1x4 2.54mm	

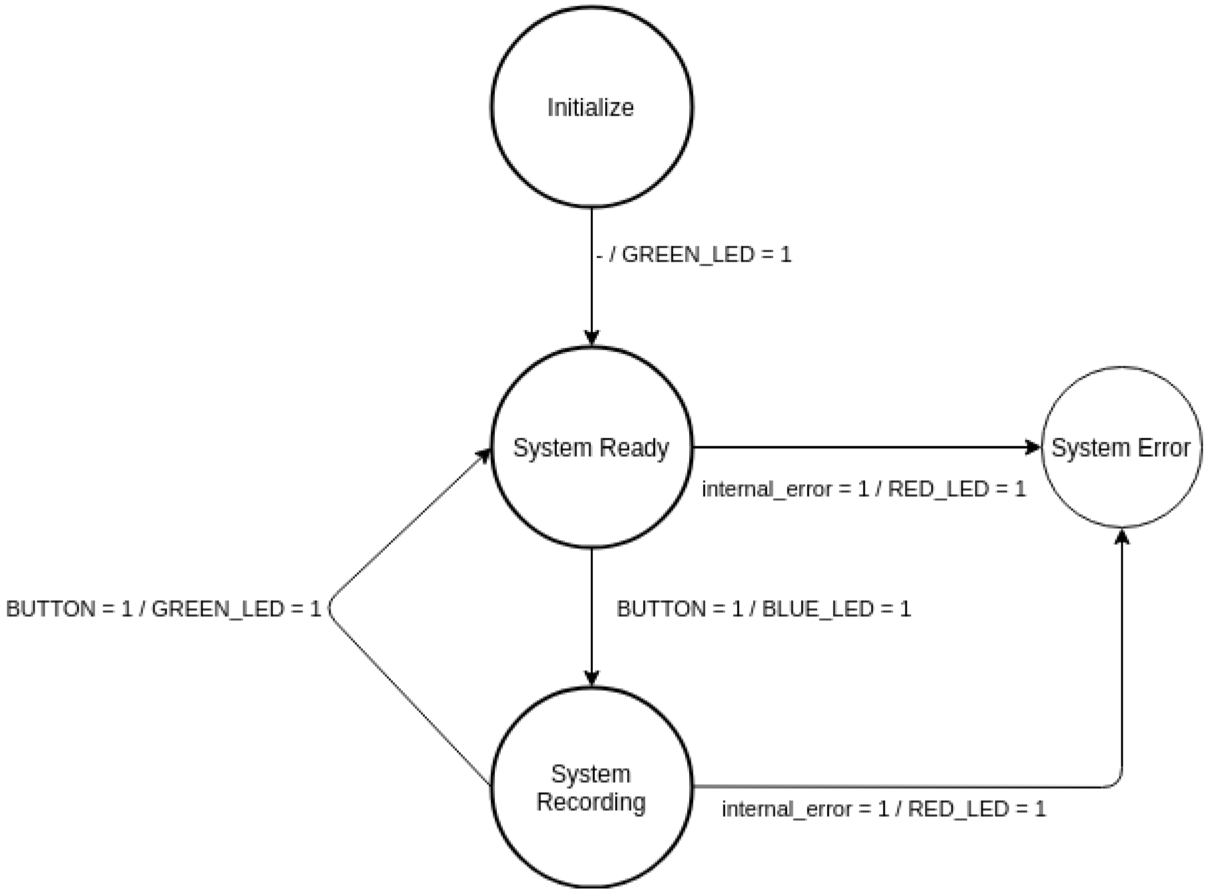
Table 3: Components to be Assembled Internally

Part Number	Quantity	Supplier	Unit Cost ($)	Total Cost ($)
PCBM (5 boards) PCBA (2 boards)	1	JLCPCB	42.91	42.91
0022272041	30	DigiKey	0.53	15.9
0022122044	18	DigiKey	1.395	25.11
0050579504	50	DigiKey	0.67	33.5
0008510108	200	DigiKey	0.0536	10.72
MPRLS0025PA00001A	16	DigiKey	10.98	175.68
MAX7356EUG+	3	DigiKey	6.55	19.65
NPA-700B-030D	1	DigiKey	57.48	57.48
Total				380.95

Table 4: Cost of each Electrical Component and Overall Module Cost

APPENDIX

I.


Initialize
- / GREEN_LED = 1
System Ready
internal_error = 1 / RED_LED = 1
System Error
BUTTON = 1 / GREEN_LED = 1
BUTTON = 1 / BLUE_LED = 1
System Recording
internal_error = 1 / RED_LED = 1