

Problem Statement and Goals

Mechatronics Engineering

Team 25, Preliminary
Ahmed Nazir, nazira1
Stephen Oh, ohs9
Muhanad Sada, sadam
Tioluwalayomi Babayeju, babajejt

Table 1: Revision History

Date	Developer(s)	Change
09/24/22	Ahmed Nazir	Goals, Stretch Goals
09/25/22	Stephen Oh	Problem Statement, Input/Output, Stakeholders

1 Problem Statement

Competitive engineering extracurricular activities at the University level revolves around the design, build, test, learn (DBTL) cycle. In teams that have not yet reached maturity in completing high quality and timely DBTL cycles, the design/building phases are over-prioritized, leaving the testing/learning phases under-resourced.

Most teams struggling to achieve timely DBTL cycles compete in deeply technical applications of engineering. Problematically, deeply technical applications demand significant amounts of time into the design/build phase of a competition year to create a working prototype.

1.1 Problem

Our capstone group recognizes the challenge teams face with the under-allocation of resources for testing, and seek to produce a solution that reduces the time to obtain and store quality testing data and extract value from the raw data they collect.

1.2 Inputs and Outputs

The solution will collect raw data as input and output Key Performance Indicators (KPI) statistics generated from test data.

Raw data input to the system will normally be aggregated upon completion of the test. The system will then communicate the contents of the aggregated data to a computing device. Users will have the opportunity to provide input on the raw data collected before saving it to a database. This input should allow the user to modify the raw data to delete inconsistent data points before sending output. The user will then submit the data to a database with additional information to specify key details of the test. An application will then allow users to view generated KPI statistics on the test data stored in the database.

Inputs:

- A. Raw data collected by testing equipment.
- B. Communication method mode select between wired versus wireless transfer of raw test data.
- C. User modifications of raw test data file to maintain consistency in data through data point modifications.
- D. User notes on modified test data file to specify details of test data collected.
- E. Data processing mode select between generating KPI metrics in real time, live data collection, versus generating KPI metrics in post with data from a complete test.

Outputs:

- A. Graphical view of test case data over time.
- B. Tabular view of Key Performance Indicators.
- C. Historical trends of related test Key Performance Indicator results.
- D. Testing trends across sub-teams.

1.3 Stakeholders

Formula Electric teams exemplify a highly technical, extra-curricular engineering team at the University level. On a yearly basis, these teams compete in competitions organized by a governing body, Formula SAE, who judge the quality and effectiveness of engineering ideas, designs, fabrications, and tests applied on a team's vehicle.

Specifically, ideal candidates who can benefit from our capstone will be Formula Electric teams with 0-4 years of competition experience. Teams without four years of experience are ideal because they typically cannot finish comprehensive tests to validate sub-team designs and overall vehicle design in a competition year.

As a result, our primary stakeholder will be McMaster's Formula Electric team because the team has 3 years of competition experience and faces difficulty validating all aspects of the vehicle through comprehensive testing.

1.4 Environment

The environment for our Arduino will include the use of the Arduino Software IDE which has a text editor for writing code, an area to receive messages, a toolbar for common functions and multiple menus for various uses. In the environment, it will allow our Arduino to be programmed and communicate with the various programs we have running.

Our main environment for software programming will be VSCode. This is where we will write our various codes to run functions that will program our Arduino and other functions that we will need. We will also use it to write our SQL code which will continually be receiving data from our hardware components.

2 Goals

Goals	Explanation
Hardware collects data points from tests	Our final hardware component should be able to record test data for the main components we want to measure. It should measure vibration, shock, temperature and humidity
A user interface which interacts and submits the data	A modern GUI that is intuitive and easy to use, which should be able to communicate with our hardware to receive the measurements from the sensor. The GUI should have two modes, the first showing the live data as it comes from the device. The second mode should allow for the user to preview the data from the test, and extract certain important parts before sending it into the database
A record of the historical data should be organized	To keep testing consistent and comparable, all previous test data should be organized into a database which cannot be edited to prevent users from tampering with test results. Historical data will allow Formula teams to be able to see how certain modifications to their design can improve or hurt motor test results.
The final product should have a website which displays the data in a useful manner	Once all the tests are sent to the database, we will have a website which shows all the major test information in a data analytics platform to quickly and easily see if the vehicle is hitting targets
Setup of our device should be quick and easy	When a user would like to conduct a test our hardware should be hassle free and not add too much time to the overall setup

3 Stretch Goals

Goals	Explanation
Custom Website	Creating a GUI from scratch would make the website more customizable to our particular application in terms of UI/UX
Critical Alert	The critical alert feature would display to the user on both the App GUI and on the screen that the current test being performed is exceeding the operating conditions of the device being tested. This will bring attention to the user and they will be able to stop the test
Predictive Data	Since all the testing data and test conditions are being stored in our database, we could create a machine learning model which will be able to tell us how the motor will perform under certain conditions
Security	When data is being transmitted from our hardware device to our computer, the data should be encrypted and secure to prevent tampering with data