# Development of Data Acquisition System using non-invasive Hardware and 3D software for Electric all terrain Vehicle

Akashaykumar H Sunagar
Department of Electronics and Communication
Hubli, India
akshayhsunagar96@gmail.com

Saroja V. Siddamal
Dept of Electronics and Communications
Hubli, India
sarojavs@kletech.ac.in

***Abstract*— As today's modern vehicles are equipped with tons of electronics, data acquisition becomes easy as the necessary data regarding the vehicle is already available in the ECUs. The objective of this paper is to build a data acquisition system for a vehicle. The system is a part of Electric All-terrain Car developed to participate in SAE India BAJA 2018. The DAQ is designed using non invasive hardware and 3D graphic user interface in Unity 3D game engine. The proposed DAQ system aims at acquiring, interpreting, visualizing and analyzing the Suspension Travel, Inclination, Current Consumption and SoC, GPS location of an electric all-terrain vehicle. A low cost Ultrasonic Reflector sensor is used to measure the suspension travel.**

***Keywords*—Data acquisition, Unity3D game Engine, non-invasive hardware, Electric All-terrain Vehicle, SoC, Suspension travel, I2C-bus.**

## I. Introduction

Data acquisition systems are widely utilized in industries today. It is applied in almost every step of manufacturing process like research, development, product control, process control, quality control, testing and validation. Data Acquisition is a process of acquiring real-world data and representing the acquired data in a meaningful format that can be analyzed and interpreted. Data Acquisition System (DAQ or DAS) consists of DAQ Hardware and DAQ Software Application. DAQ Hardware consists of sensors, actuators and signal processing unit (usually preamplifier or direct micro-controller). Sensors sense the changes in real world physical parameters and convert them into electrical signals. These electrical signals are conditioned and processed by the signal processing microcontroller unit, where the sensor data is converted into a meaningful format. Actuators convert electrical signals into corresponding changes in physical real world parameters. Actuators accomplish the necessary controlled change in the physical parameter decided by the micro-controller unit.

DAQ Hardware is embedded in the physical environment from which the data is to be acquired. DAQ hardware is built by micro-controller boards and other supporting circuits. DAQ software application interprets and analyses the acquired data. It makes sense of the data by comparing it with pre-defined tolerances, admissible variations and other affecting parameters. DAQ Software usually runs on a general purpose computer.

The concept of data acquisition is the essential part of the vehicular control systems. All the functionalities carried out by the electronic control units (ECUs) in an automobile can be boiled down to data acquisition, data processing and controlled decision making. Modern advancements in vehicles have given rise to representation of the vehicular data on the instrumentation cluster or on the smart phone of the driver. The automotive DAQ systems also have a data logging feature which acts similar to a black box in an airplane. It logs all the data regarding the inputs of the driver and the corresponding responses from the various sub-systems.

The definition of data acquisition itself implies measuring and quantifying the physical world; it is only justifiable for the software application to be a projection of the physical world, i.e. simulating the real physical world in its true three-dimensions. To accomplish this, Unity3D game engine is used. Implementing 3D graphical user interface (GUI) into the DAQ software makes it very convenient to visualize the acquired data especially for automotive systems to understand the behaviour and performance of various subsystems like suspension travel of the suspension system, the pitch, roll and yaw of the roll cage or chassis, speed (or RPM), steering angle etc. Table and graph based 2D DAQ software interface would only provide time based analytics of the data, whereas the 3D environment facilitates locations based analytics in addition to time based analytics.

## II. Literature Survey

A framework for designing and implementing vehicular data acquisition system consists of three logically separated layers: acquisition, processing and evaluation [1]. DAQ is also used in automotive industry especially in testing and validation. In an automobile, acquisition of vehicular data of various subsystems, processing the acquired data for representation and finally is for evaluating the performance based on the data [1]. As today's modern vehicles are equipped with tons of electronics, data acquisition becomes easy as the necessary data regarding the vehicle is already available in the ECUs (Electronic Control Unit). In addition to the vehicle instrumentation (speedometer, fuel indicator, odometer, trip, rotations per minute-RPM, range etc.) the ECUs evaluate critical data from all the sensors and convey it to the driver in the dashboard (ex: check engine light). This type of DAQ is usually categorized as Diagnostics DAQ.

Furthermore, OEMs provide with a tool called On-Board Diagnostics (OBD) Scanner. It runs diagnostics of all the subsystems and logs the errors to help the mechanic in easy repair. When the vehicle is under development, then external sensors have to be used to evaluate the performance of the subsystems. This type of data acquisition has become popular by acquiring data by tapping the CAN bus and OBD of the vehicle [2].

Data acquisition systems are vaguely used in exchange with data logging systems. Data Logging systems processes the sensors' data and store it locally in the DAQ hardware and can be accessed through special tools. Data loggers were used previously in automobiles before the advancement in computers [3]. The stored data was accessed through a plug-in communication interface module. With the advancements in modern micro-controllers, sensors and communication technologies, the data can be easily transmitted to the user in real-time thus giving rise to real-time data acquisition systems. The acquired data is fed to electronic unit, where it is transmitted wirelessly to the DAQ software. These have revolutionized the testing procedure and made it convenient to check the errors on spot. With added functionality today's data acquisition systems come with both real-time data transmission and data logging features. In the personal computation era, the DAQ software has also evolved to handle huge chunks of data and perform computations on them in a fraction of a second. Often times, the DAQ is implemented for vehicles by a combination of acquiring data from on board ECUs and an added customizable system for custom designs [2].

## III. Conceptual Design

The proposed system involves design and implementation of DAQ system with non-invasive hardware components which can fit easily to any automobile (especially project cars). A novel user interface for DAQ software is proposed. This is obtained by integrating 3D environment into DAQ software user interface and non-invasive methods to acquire data in the DAQ hardware. The DAQ software and DAQ hardware are designed and implemented for an electric all-terrain vehicle which is also a project buggy for SAE India BAJA 2018.

The proposed DAQ system aims at acquiring, interpreting, visualizing and analyzing the following vehicular data of an electric all-terrain vehicle.

- Suspension Travel
- Inclination
- Current Consumption and SoC
- GPS location

The DAQ hardware architecture consists of two micro-controllers namely, sensor controller and communication controller.

### *A. DAQ Hardware System*

All the sensors are interfaced with the sensor controller (ATMEGA328 Arduino Pro Mini) as shown in the fig. 1 Each Ultrasonic sensor HC-SR04 uses two digital GPIO pins, one for trigger and one for echo. The four sensors account for pins-2, 3, 4, 5, 6, 7, 8 and 9. The current consumption is measured from the HTFS-200P current sensor and the SoC of the battery is computed by executing the Coulomb Counting algorithm. The inclination sensor MPU-6050 uses the I2C bus - A4 for SDA (Serial Data) and A5 for SCL (Serial Clock). And the GPS module uses the declared serial communication pins- 10 and 11. The sensor controller first evaluates the suspension travel data from all the four ultrasonic sensors. It then reads the appropriate data from the inclination sensor module MPU6050 through the I2C bus. Finally it receives the GPS location from the GPS module SKG13BL.

All of this information is communicated to the communication controller (Arduino-LoRa Transceiver) through the I2C bus (A4 for SDA and A5 for SCL). The communication controller formats the received data and puts it in an appropriate frame format as shown in the fig. 2.

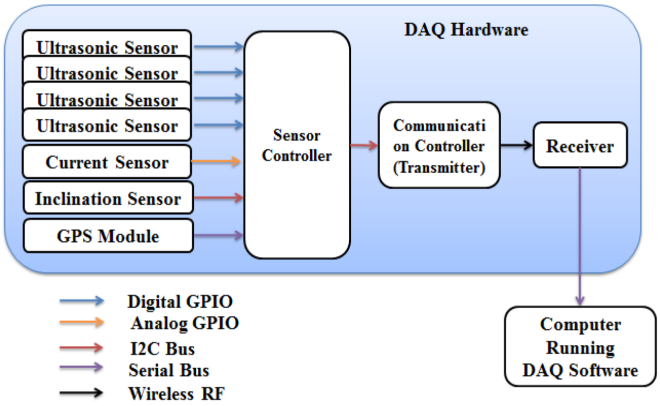


Fig. 1: Hardware Integration

The table 1 shows the frame format used in the design of DAQ.

Table 1: Frame format and Representation

| Frame Format | Representation |
|---|---|
| **15.3849** | Latitude of the Buggy's GPS location |
| **75.1041** | Longitude of the Buggy's GPS location |
| **23.56** | Effective Length of the Front Right Shock Absorber |
| **24.49** | Effective Length of the Front Left Shock Absorber |
| **23.09** | Effective Length of the Rear Right Shock Absorber |
| **23.91** | Effective Length of the Rear Left Shock Absorber |
| **-7** | Roll or Inclination of the buggy along X-Axis |
| **+5** | Pitch or Inclination of the buggy along Z-Axis |
| **48.94** | Current Consumption of the buggy at this instant |
| **73** | State of Charge (SoC) of the battery pack |

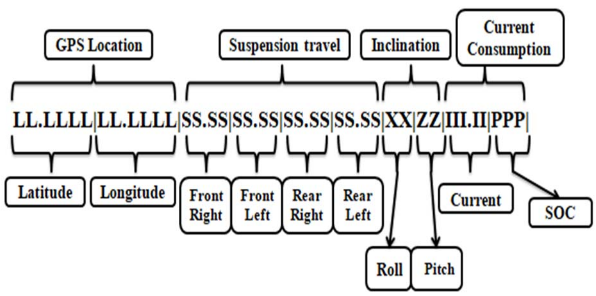


Fig. 2: Formatting of acquired data into a proper frame

The communication controller also acts as a LoRa transmitter. It transmits the data frame at a particular frequency of 956 MHz. On the receiver side, the receiver controller is also tuned at the same frequency. The receiver controller is connected to the computer and it communicates the received data frame to the computer via the serial communication port of the computer.

*B. Suspension Travel Data Acquisition and Visualization*

The all-terrain capability for any vehicle is primarily achieved by its suspension system. Hence monitoring the response of the suspension system becomes important as the buggy runs on different terrains.

To measure the suspension travel, a linear potentiometer is used in industries. It has a resistive static rod of fixed resistance upon which, a movable contact moves to output the effective resistance based on its position. The linear potentiometer is mounted in parallel to the suspension system arrangement (spring and damper assembly). When a voltage is applied to this potentiometer, the output voltage will be proportional to the effective resistance, which in turn is proportional to the suspension travel.

But the linear potentiometer is not compatible and not a good choice to use in a custom build and it also cost around a $100 making it unfeasible for the car. The ultrasonic sensor which works on the theory of reflection of high frequency sound waves is a manageable substitute.

Ultrasonic sensors when triggered emit short high-frequency sound pulses. When these pulses strike an object, they bare reflected back as an echo signal to the sensor. The sensor micro-controller records the time difference (t) between sending the signal and receiving the echo. The speed of sound (v) in air under room conditions is 343 m/sec.

$$v = 343\, m/sec = 0.0343\, cm/\, usec \qquad 1$$

$$Wkt, \quad Distance, d = v * t \qquad 2$$

In this case, the distance is 2d because the wave is reflected back,

$$2d = v.t \qquad 3$$
$$d = (v.t)\, /\, 2 \qquad 4$$

The design is to mount the ultrasonic sensor on one side of the shock absorber and a reflector plate on the other side as shown in the fig. 3.

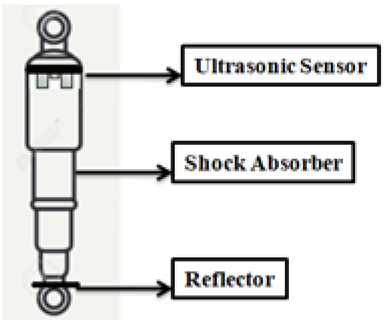


Fig 3: Ultrasonic sensor arrangement to measure effective suspension length

The effective suspension length of the shock absorbers when the buggy is standing still on a 0° inclination floor is recorded as the reference value to calculate the suspension travel. Then by comparing the acquired instantaneous effective suspension lengths with the reference value, the suspension travel is calculated. If the instantaneous effective suspension length is less than that of the reference value, then the shock absorber is said to be compressed by an amount equal to the difference in the values and vice versa.

In real world, the compression and expansion of the shock absorber is seen by the movement of the wheel assembly along with the tie rods, connecting arms and drive shafts etc. In Unity, the buggy is 3D modeled by referring the original real world vehicle, so the same technique is applied in DAQ software to visualize the suspension travel data by moving the 3D models of the wheel assembly with respect to the 3D model of the roll cage as shown in the fig 4.

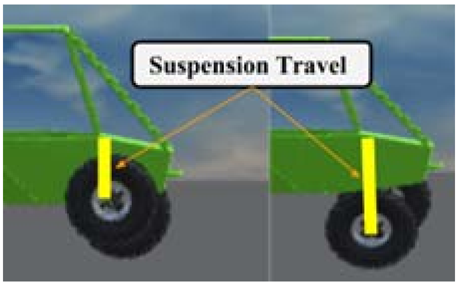


Fig 4: Suspension travel visualization by moving the wheel assembly

*C. Inclination Data Acquisition and Visualization*

To measure the roll, pitch and inclination of vehicle MPU-6050 which can be easily interfaced with generic Arduino micro-controllers is used. This sensor measures the inclination in all the three axis. MPU-6050 sensor contains a MEMS (Micro-Electro-Mechanical Systems) accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware. Therefore it captures the x, y and z channels at the same time. The sensor uses the I2C-bus to interface with Arduino. The MPU-6050 combines both an accelerometer and a gyro.

To access the inclination data of all the three axes, the sensor controller sends commands to the MPU-6050 through I2C bus. The inclination data of the vehicle along the X- and Z-axes is visualized manipulating the rotation component of the 3D model of the vehicle according to the acquired inclination value. In addition to this, as the buggy's inclination along the X-axis is also the inclination of the terrain (gradient) upon which the buggy is moving, the instantaneous altitude of the terrain is calculated by correlating the inclination data with respect to the location of the buggy. By doing this, the path traversed by the buggy is visualized by depicting the gradient and altitude.

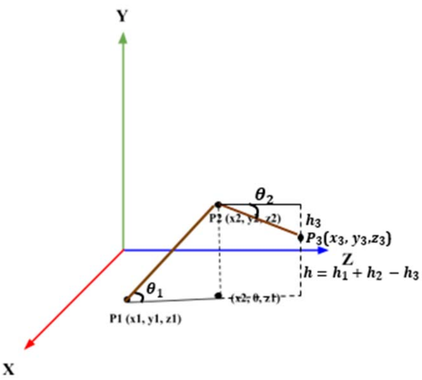


Fig 5: Path Traversal of the vehicle

Consider a scenario where the buggy moves from point $P_1(x_1,\ y_1,\ z_1)$ to point $P_2(x_2,\ y_2, z_2)$ as shown in the fig.5. Let the inclination of the buggy and (in turn) terrain be $\theta_1$. Here, the X, Y and Z coordinates represent the longitude, altitude, and latitude of the buggy respectively at that particular location. The latitude and the longitude of the buggy's location are acquired through SKG-13L GPS module and the inclination along the X-axis by the MPU6050 accelerometer sensor. Let the altitude of the buggy or the height along the Y-axis at point $P_1$ be $h_1$. Let the altitude of the buggy at any point be *h*.

The basic idea is to represent the instantaneous altitude data in the DAQ software by instantiating a cube. The planar distance between the two points is equal to the distance between the points in the XZ plane. This distance is calculated by

$$d_1 = (z_2 - z_1)^2 + (x_2 - x_1)^2 \qquad 6$$

The instantaneous altitude Y-axis at point $P_2$ is calculated by

$$h_2 = d_1 tan\theta_1 \qquad 7$$

The Altitude of the buggy at point $P_2$ is the sum of the altitudes at previous points

$$h = h_1 + h_2 \qquad 8$$

Now, consider a third point $P_3(x_3, y_3, z_3)$ and the inclination at point $P_2$ is $\theta_2$(which is negative, indicating downhill).

If the distance between $P_2$ and $P_3$ is given by $d_2$.The instantaneous altitude along the Y-axis at point $P_3$ is given by

$$h_3 = d_2 tan(-\theta_2) \qquad 9$$

Again, altitude of the buggy at point $P_3$, is the sum of the altitudes at previous points.

$$h = h_1 + h_2 + h_3 \qquad 10$$

Now, if the buggy has moved downhill from $h_2$ to $h_3$. this is quantified by the negative sign of inclination of $\theta_2$.

$$h = h_1 + d_1.tan\theta_1 - d_2.tan\theta_2 \qquad 11$$

By equation (11), it is evident that the altitude of the buggy at point $P_3$ is decreased from point $P_2$ by an amount equal to the instantaneous altitude at point $P_3$. Therefore, by adding more points like this, the instantaneous altitude calculated at individual points are added cumulatively to the overall altitude to simulate the altitude of the buggy. Generalizing the above approach, the overall altitude ($h_k$) of the buggy at point $P_k$ is equal to the sum of all individual instantaneous altitudes of the previous points.

$$h_k = \left(\sum_{k=2}^{k} h_{k-1}\right) + h_1$$

$$h_k = \left(\sum_{k=2}^{k} d_{k-1}.\tan\theta_{k-1}\right) + h_1 \qquad 12$$

After computing the altitudes of the buggy at all the points of its traversal, it is visualized by adding and instantiating cube of height equal to the overall altitude of the buggy at every point. With Google maps loaded on the terrain, the visual representation of the altitude would look as shown in the fig. 6 by blue lines.

## D. Current Consumption and SoC Computation

To measure the current consumption of the buggy, a current clamp sensor HTFS 200-P is used. It measures the current based on the Hall Effect principle. It provides electrical isolation, high accuracy and a fast response.


Fig. 6: Visual Representation of buggy's altitude along a path

The Coulomb Counting algorithm determines the state of charge (SoC) of the battery pack. The HTFS 200-P is used to measure the current incoming to the battery and outgoing of the battery. The time interval for current flow is found and it is multiplied with the current to give the charge coming into the battery while charging or going out of the battery while discharging. This calculated charge is added to or subtracted from the SoC remaining respectively. The data of SoC remaining is acquired. Fig.7 shows the flowchart for calculating SoC.

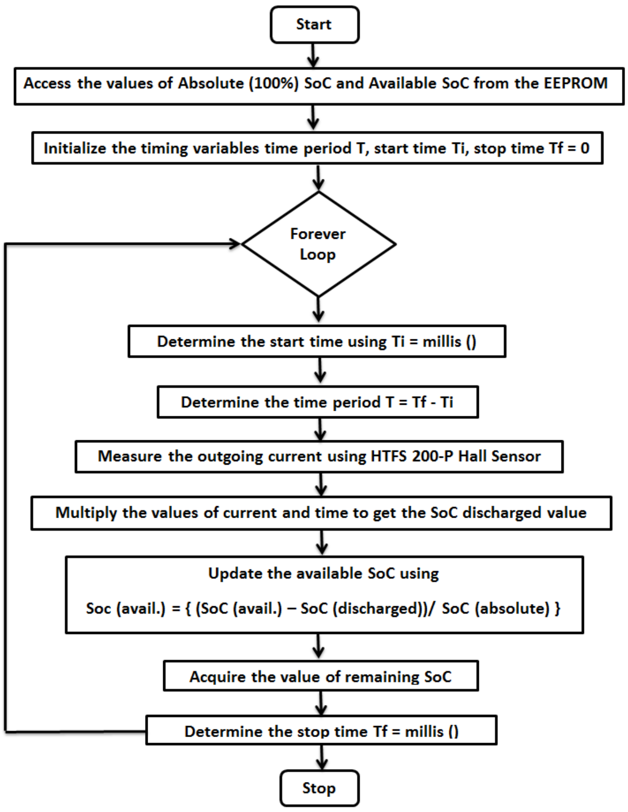


Fig. 7: Flowchart for calculating SoC using Coulomb Counting algorithm

## E. GPS Location and Tracking

To track the location of the vehicle, the GPS module SKG13BL is used. It is a complete GPS engine module with features like super sensitivity, ultra-low power and small form factor. The GPS signal is applied to the antenna input of module, and a complete serial data message with position, velocity and time information is presented at the serial interface.

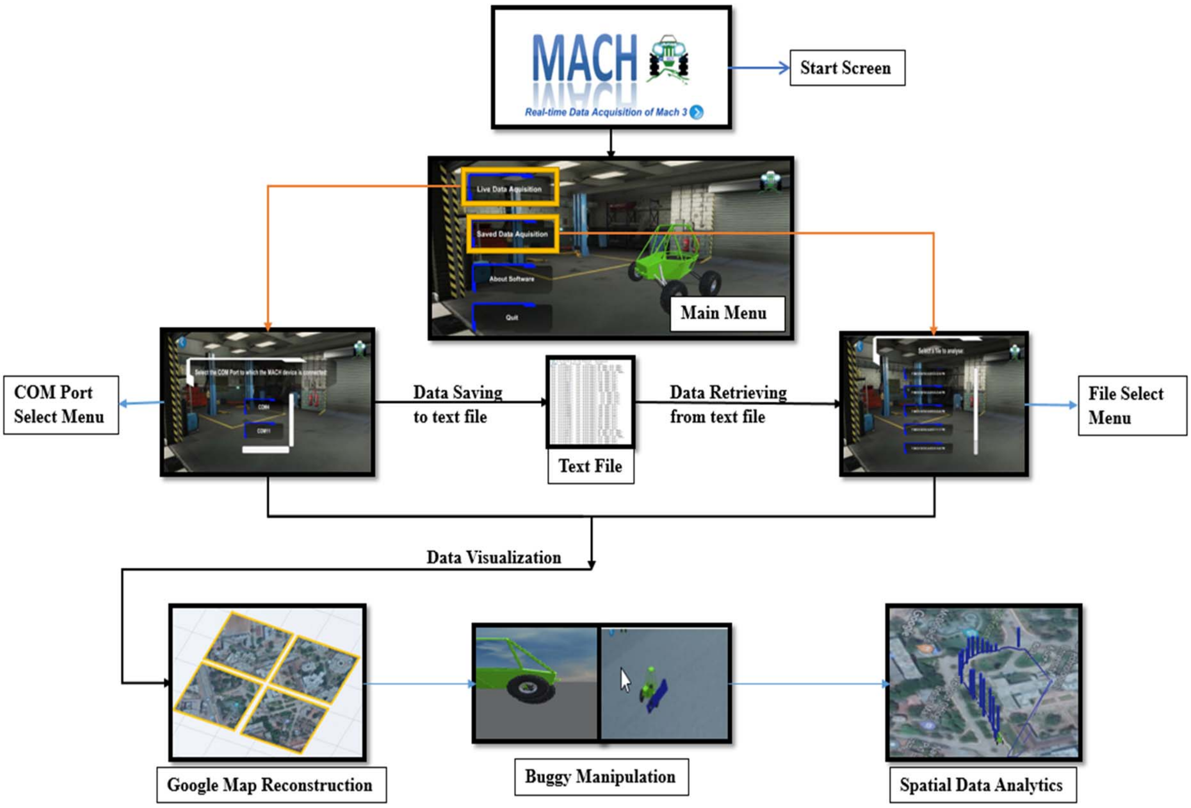


Fig. 8: DAQ Software Application Integration

The Software Serial library is included to allow serial communication on other digital pins of the Arduino compatible micro-controllers, using software to replicate the functionality. It is possible to have multiple software serial ports with speeds up to 115200 bps.

*a) Construction of terrain from Google Maps and real-time buggy tracking*

As with any 3D platforms, Unity has an option of applying images as textures to the 3D models. But in addition to that, Unity has a WWW class in its scripting API (Application Programming Interface) which provides simple access to web pages.

Placement of the 3D model of the buggy on the real-time GPS location marker (latitude and longitude) is required. To facilitate this, Google maps static API is used. Static Maps API is used to create a map based on URL parameters sent through a standard HTTPS request. The HTTPS request displays the map as an image. Static Maps API is intended for website and mobile developers who want to include Google maps images within a webpage or mobile application. Fig.9 shows the static map image for the GPS coordinates (15.369023, 75.121614).

Multiple map fragments with static maps images as textures are used in construction of big map is reconstructed like solving a jigsaw puzzle.

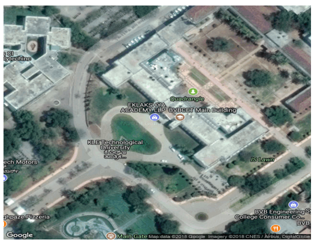

Fig. 9: Static map image for the GPS coordinates (15.369023, 75.121614)

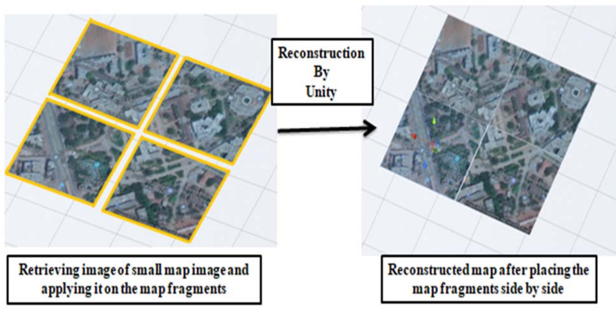


Fig. 10: Reconstruction of large map from smaller individual map fragments

## IV. RESULTS

### A. DAQ Software Application

The DAQ software application includes an interactive main menu to handle all the different functionalities and sub-menus as shown in fig. 8.

The DAQ software data visualization is tested to visualize data of each sub-module implemented.

1) The GPS tracking data of the DAQ hardware circuit is visualized as shown in the fig. 11 and 12. The DAQ hardware circuit was moved around in the college ground and the DAQ software is run to get the following results as shown in the fig 11.

Figure 11: GPS tracking (Top camera view)

The inclination data acquisition module is tested by carrying the DAQ hardware circuit as shown in fig12.

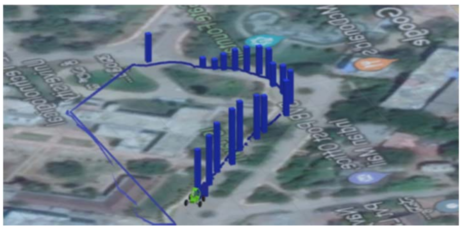


Figure 12: Inclination data mapping

## V. CONCLUSIONS

The modern day vehicles are equipped with ECUs. Acquiring various data and analyzing these data becomes easy. This work - a non invasive DAQ is designed and developed to meet custom requirements of a project car. This is integrated into an electric all-terrain vehicle developed to participate in SAE India BAJA 2018. To give the 3D view of the vehicle, the software application is built using 3D graphic user interface in Unity 3D game engine. The proposed DAQ system acquires suspension travel, inclination of the vehicle along with main battery of 48V, 110Ah. The vehicle also has a 12V auxiliary battery. This is used for to run all other auxiliary systems in the vehicle and to run the fans to cool the main battery and the tractive system (BLDC motor and controller). The DAQ measures the current consumption and computes the state of charge of the battery. GPS location of an electric all-terrain vehicle is also traced.

## REFERENCES


[1] M. Wittmann *et al.*, "A holistic framework for acquisition, processing and evaluation of vehicle fleet test data," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, 2017.

[2] L. Baghli, K. Benmansour and M. Djemai, "Development of a data acquisition and tracking system for vehicles," *3rd International Symposium on Environmental Friendly Energies and Applications (EFEA)*, St. Ouen, 201.

[3] F. Salewski, S. Kowalewski, "Hardware/software design considerations for automotive embedded systems", *IEEE Transactions on Industrial Informatics*, vol. 4, no. 3, pp. 156-163, Aug. 2008.

[4] E. Althammer, T. Gruber, S. Fuchs, "Intelligent Road Safety Using Infrastructure to Vehicle Communication-A Holistic Systems Issue", *17th Interdisciplinary Information Management Talks*, pp. 309-320, 2009.

[5] Perez, A., M. I. Garcia, M. Nieto, J. L. Pedraza, S. Rodriguez, J. Zamorano, "Argos: An Advanced In-Vehicle Data Recorder on a Massively Sensorized Vehicle for Car Driver Behavior Experimentation", *IEEE Transactions on Intelligent Transportation Systems*, pp. 463-473, 2010

[6] K. Arshak, A. Arshak, E. Jafer, D. Waldern, J. Harris, "Low-power wireless smart data acquisition system for monitoring pressure in medical application, " Microelectronics International, vol. 25, no. 1, 2008.

[7] Liberios VOKOROKOS, Branislav MADOŠ, Norbert ÁDÁM, Anton BALÁŽ, “Data Acquisition in non-invasive Brain Computer Interface using Emotiv Epoc Neuroheadset”, Acta Electrotechnica et Informatica, Vol. 12, No. 1, 2012, Pages 5–8, DOI: 10.2478/v10198-012- 0001-y.