REPORT

Problem Statement: To upload the data from sensors to a cloud

WEEK 1

The problem statement or the objective of what had to be done was comprehended and the best ways to achieve the goal was decided. One needed a cloud platform and a tool to retrieve the data. Firebase by google was chosen for the project for a cloud platform where one could have real-time database. A real-time database was created by us and we incorporated a HTML page to retrieve the data. This HTML page would update each time the database was update. In order to enable data protection and provide security, the HTML was kept authenticated, that is, only those logged onto their accounts could view the data.

WEEK 2

A Node-MCU was used and how it is used with firebase was understood. To familiarise ourselves with the working of Node-MCU a number of basic programs were conducted on the same including controlling an LED using the database. Up till now a real-time database was created where the data that had to be changed manually. The next stage would be to ensure the same happens using sensors. A temperature and ultrasonic sensors were connected to the Node-MCU with the working and connections analysed.

WEEK 3

The temperature and ultrasonic sensors which were connected to the Node MCU successfully uploaded the real-time temperature and distance values as and when updated to the firebase real-time database once the Node-MCU was connected to the database. The data that was retrieved by the authenticated HTML had to now be changed to public.

In order to retrieve the data, a different method had to be opted and incorporated. This method was the Datasnapshot which would send a snapshot of the data from a JSON file to the HTML. After this issue being resolved, another one was observed, the HTML page would not update values on its own and that required manual refreshing.

WEEK 4

The readings of the temperature, Hall effect, IR and ultrasonic sensors were sent to the database with a few changes made in the code to incorporate the same. In order to solve the previous issue, another HTML was created to retrieve data.

Instead of just keeping it as a HTML page though, it was made into an android application which would update the values on its own without needing manual refreshing. The sensor values could be sent to and retrieved from the database any time which would complete the entire project

WEEK 5

The IR sensor required a delay of less than 100 microseconds which could not be achieved due to the uploading of data through Wifi .

The maximum Speed we could get was 300 microseconds , which would interrupt the calculations , and give wrong values for the RPM .

The idea of using an IR sensor now , could not be interpreted as we could not see any way to improve the speed .

Another issue now , is that there is only one Analog pin on the Node-MCU , but we require to connect 4 temperature sensors .

The temperature sensor being used is LM35 , an analog sensor and not a digital sensor , so in order to connect many of them , the possible ways are to use either a MUX or an analog pin extender .

A 74HC153 MUX can give upto 8 outputs , which now resolves the problem .We used two selection lines , and can upto 4 sensors .

Another sensor was to be incuded : Accelerometer MPU6050 . With this sensor we could get the Acceleration of the vehicle.

REPORT BY:

Nikitha Sharma - PES1201800618

Sakshi Vattikuti - PES2201800157

Triambaka Naresh - PES1201800225