BikeBeat: An Implementation of an Android Application using Real Time Data from a Motorcycle using Arduino Microcontroller and Bluetooth

Mansi Kamblí1, Bhavisha Mehta2, Shreyash Paratkar3, Parth Kamdar4
Assistant Professor1, BE Student2,3,4
Department of Computer Engineering
K. J. Somaiya College of Engineering, Mumbai, India

Abstract:
This paper presents a project aimed at real time collection of basic motorcycle parameters such as distance travelled and fuel level, using an Arduino microcontroller to tap into the sensors that provide these values to the dashboard in the motorcycle. The distance travelled by the motorcycle is calculated in the microcontroller after reading input from the Hall-Effect sensor pre-installed on the motorcycle. The fuel level inside the fuel tank is estimated by the microcontroller by measuring the voltage drop inflicted due to the fuel in the tank. This data is transferred wirelessly to an application running on a smartphone with Android OS. The data traversal is carried out using Bluetooth wireless data exchange technology. The values obtained are further manipulated by the Android application to create summarized reports which are customized to each user, along with timely generated notifications. The Android application will also provide real time distance and fuel estimation to a given destination based on the user’s driving style, time estimation to pre-plan a trip and reminders in case of low fuel.

Keywords: Arduino, Android, Bluetooth, Fuel Level, SQLite Database, Hall-Effect Sensor, Pulse Width Modulation, Voltage Divider,

I. INTRODUCTION

The current technology used in expensive, premium motorcycles with Electronic Control Units (ECU’s) helps in modifying the transmission characteristics of the vehicle hence increasing the performance of the vehicle [1]. None of the manufacturers have invested their resources in the development of any interface between a mobile phone and a motorcycle. Though many projects using Arduino microcontroller aimed at calculating the motorcycle speed, distance travelled and fuel levels constantly increasing and decreasing in the motorcycle have been undertaken and implemented successfully by people [2]-[4],[8], there hasn’t been any interfacing of theirs with communication devices. The application being implemented in this project aims at integrating all the three factors namely distance, time and fuel in order to generate suitable reports. The application makes use of the basic functionality that is part of all the classes of motorcycles, ranging from economical to sumptuous. The goal of the project is to deliver a system that consists of a microcontroller collecting data from the already installed Hall-Effect sensor and fuel tank circuit of the motorcycle, modifying the data into readable values and transferring this data to the Bluetooth module. The microcontroller being used for this implementation is the Arduino Uno microcontroller board based on ATmega328P. The Bluetooth module being used is the HC-05 Bluetooth module which can be set to be either Master or Slave configuration [5]. The readable data collected by the Bluetooth module is subsequently sent to an Android application running on a smartphone with Android - KitKat 4.4 OS and above. The Android application breaks down the data received from the Bluetooth module and stores it in a relational SQLite database in real time. After the collection is finished, the tuples from the tables generated are used to process results such as total distance travelled, total time taken, average speed, highest instantaneous speed i.e. the top speed, amount of fuel consumed in the trip and the fuel average per liter given by the motorcycle. These results are shown to the user in graphical ways to enhance his/her user interface experience. The implemented system depends on the following for its smooth working:

A. Arduino Uno microcontroller and HC-05 Bluetooth module
The Arduino Uno microcontroller has to be given a constant DC supply of +9V for the entirety of the journey to facilitate accurate data processing and power supply to the HC-05 Bluetooth Module. The Bluetooth Module is expected to work without disruptions due to physical or logical forces and is expected to send each and every packet of data to the intended user’s mobile phone without loss of packets thus giving a more accurate result.

B. Android phone
It is expected that the user’s phone has an Android - Kitkat 4.4 OS and above on his/her phone. It is also expected that the user keeps the phone and the android application switched on for the entirety of his/her journey.

C. Fuel tank circuit and Hall-Effect Sensor
The fuel tank circuit and the Hall-Effect sensor of the motorcycle are expected to under a constant supply of the motorcycle’s battery (~12V DC). The fuel tank is assumed to be in a stationary state most of the time such that its inclination with respect to the ground doesn’t disrupt the ball mechanism of the fuel tank which in turn shift the level of the rheostat to give out an absurd change in the voltage level.
II. WORKING

A. Distance Calculation using Hall-Effect sensor

A Hall-Effect sensor is a transducer, a device that converts signal in one form of energy to a signal in another form of energy. The Hall-Effect sensor of the motorcycle gets a 12V supply from the motorcycle’s DC battery. This voltage helps the sensor to create a magnetic field around it. The magnetic flux lines of this magnetic field are “cut” by the magnets present in the front wheel of the motorcycle. The Vcc of the sensor is connected to the motorcycle’s 12V power supply. The GND of the sensor is provided by the motorcycle and the Vout or Signal pin of the Hall-Effect sensor is connected to Arduino’s PWM pin. (Figure 1.1) According to the principle of electromagnetic induction, there is a production of electromagnetic force (emf) as a result of the changes occurring in the magnetic field due to the cutting of the magnetic field by the magnets. This generates an instantaneous voltage in the output of the Hall-Effect sensor every time a magnet cuts the magnetic field [6]. The motorcycle’s front wheel has 8 magnets infused in it, thus for every 8 times a voltage ~5V is observed in the output, it can be inferred that the wheel of the motorcycle has completed one rotation. The motorcycle’s front wheel’s diameter was measured to be ~55cm. Hence the distance travelled by the motorcycle in 8 HIGH’s on the output of the sensor is equal to ~1.728m.

\[
\text{Circumference of the wheel} = \pi \times \text{Diameter of the wheel} \\
= \pi \times 0.55m \\
=1.728m
\]

The output of the Hall-Effect sensor is given as input to one of the PWM input pins of the Arduino board. Pulse Width Modulation is used to create a square wave which is a signal switched between ON and OFF. Reading the input of this pin in Arduino using pulseIn() function, gives an accurate response whenever the magnetic field is cut, and hence, the distance can be calculated for every 8 HIGH’s observed through the function. The diagram in figure 1.2 shows the different PWM square waves observed for different speeds. The distance is updated by adding 1.728 m to a distance variable every time a set of 8 successive highs are observed.

The pseudo-code for the same is as follows.

```
if (PWM_output==HIGH)
    number_of_high_signals=number_of_high_signals+1
if (number_of_high_signals==8)
    number_of_high_signals=0
    Distance=distance+1.728
```

B. Fuel Tank Level Calculation

The fuel tank of a motorcycle comprises of a floating ball, which rests on the surface of the fuel. This floating ball is connected to a rheostat that is, a variable resistance. The variable resistance is the main responsible value in calculating the fuel level inside the motorcycle’s fuel tank [8]. The fuel present in the tank brings about a change in the variable

![Figure 1. Hall-Effect Sensor Connection Diagram](http://ijesc.org/1.png)

![Figure 2. Distance – PWM Wave Mapping Diagram](http://ijesc.org/2.png)
Resistance as its level increases or decreases, which when connected to the 12V DC battery of the bike, causes a voltage drop in the total circuit [9]. Our system exploits this attribute in order to calculate the fuel consumed by the user. Thus, the total effective voltage in the circuit is measured by the Arduino by tapping into the fuel tank’s output. Readings of voltage for different amounts of fuel in the tank were observed as shown in the table 1.

<table>
<thead>
<tr>
<th>Fuel (In Litres)</th>
<th>Measured Voltage (in Volts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11.97</td>
</tr>
<tr>
<td>1</td>
<td>11.01</td>
</tr>
<tr>
<td>2</td>
<td>9.98</td>
</tr>
<tr>
<td>3</td>
<td>8.99</td>
</tr>
<tr>
<td>4</td>
<td>8.03</td>
</tr>
<tr>
<td>5</td>
<td>7.01</td>
</tr>
<tr>
<td>6</td>
<td>5.98</td>
</tr>
<tr>
<td>7</td>
<td>5.04</td>
</tr>
<tr>
<td>8</td>
<td>3.98</td>
</tr>
<tr>
<td>9</td>
<td>2.96</td>
</tr>
<tr>
<td>10</td>
<td>2.01</td>
</tr>
<tr>
<td>11</td>
<td>0.99</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 1. Fuel Voltage Readings

It was hence concluded from this data, that
\[ V_{\text{out}} = V_{\text{cc}} + V_{\text{drop}} \]
Where,

\[ V_{\text{out}} \] is the Output Voltage
\[ V_{\text{cc}} \] is the Voltage provided by the 12V Battery
\[ V_{\text{drop}} \] is the Voltage Drop caused due to the variable resistance.

It can be observed from the figure 2.1 (bottom of the page) that the output voltage in the case when there is minimum fuel in the tank comes out to be 12V. Since the operating voltage of the Arduino Uno microcontroller is around 5V, prolonged exposure to a incoming voltage greater than 5V can be harmful for the analog pin, accepting the input. Hence it calls for the installation of a voltage divider circuit. Using two resistors R1 (1 MΩ) and R2 (100 kΩ), the input voltage is divided, before passing it on to the analog pin. The input given to the Arduino, is multiplied by the division factor based on the sum of the resistors attached in series, to get the actual output voltage. This also makes sure that the Arduino doesn’t draw any significant current from the motorcycle’s in-place circuit, thus ensuring flawless functioning of the motorcycle’s in-place circuit.

C. Data packet
The data obtained from the above two functions represents two fields of the data packet which has to be sent to the Android application. These two fields are tagged with a timestamp of the time when the data was recorded in the Arduino. This field is given by the Arduino function millis() which gives the time elapsed in milliseconds since the last time the program was loaded in the Arduino’s memory. After addition of this field the data packet of the Arduino looks like the table 2.

<table>
<thead>
<tr>
<th>Time in milliseconds</th>
<th>Distance covered in meters</th>
<th>Fuel voltage output</th>
</tr>
</thead>
</table>

Table 2. Data packet

D. Bluetooth Connectivity with Android Application
HC-05 is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. A customized service has been created which is used to communicate with the Arduino microcontroller. The reason to run Bluetooth connectivity as a service is that it can be run in the background without disturbing the main User Interface thread’s functioning [10]. The main UI thread gives two calls, one is “start” which starts the Bluetooth service and “stops” which destroys the service. Once the service is started the first thing it does is to check if Bluetooth is supported on the device or not. If the device doesn’t support Bluetooth it will show an error and close the application else if it does support Bluetooth, it goes on to check if Bluetooth is enabled or not. If Bluetooth is not enabled user is prompted to switch it on. Then the application connects to the HC-05, using its
MAC address. Once the connection is successful the application opens up an input-stream to read the data coming from the HC-05. The data received is then converted to a string and is parsed. (Figure 3.1)

E. Android Application Database

The SQLite database is implemented using a helper class in the Android application [11]. The helper class is responsible for the creation of the database and implementation of its schema. The database for the implemented system requires creation of two tables, one which stores all the records of the latest trip, and another one which stores the results of all the trips. An object of this class is called in the main activity’s initialization function to create the database when the application is installed. Another object of the helper class interacts with the Bluetooth service being employed by the main activity.

Every time a data packet is obtained from the input stream, it is converted into a string, and the three fields in the string are separated. These three fields are parsed in order to convert the timestamp, distance and fuel voltage to their respective data types. These three values are then immediately stored in the SQLite database table for last trip’s records. The data being received is continuously operated upon to calculate the total distance travelled, total time taken, average speed, top speed achieved, fuel consumed and fuel average per liter given by the motorcycle respectively. After the end of the trip, these variables along with the date and time of the particular trip are stored in the other table that stores the statistics of all the trips. This data is then sent as notifications to the user using the NotificationCompat class in Android and also to the main activity that is responsible for displaying the generated reports to the user and using it for the other functionalities of the application. (Figure 4.1)
III. CONCLUSION

In this paper, we have presented a way to retrieve data from installed sensors on a motorcycle and transmit them using Bluetooth as a communication medium to an Android application. The data obtained has a wide range of applications, from implementing security measures to a personalized economical fuel saver. Android being a versatile tool for application implementation opens up a horizon of opportunities to generate applications employing this data. However, in this humble attempt of ours we have limited the application’s services to the user. These services are aimed at helping the user take useful decisions based on the results obtained from the application.

IV. REFERENCES


V. ACKNOWLEDGEMENT

We would like to sincerely thank our project mentor Prof. Mansi Kambli of K.J.S.C.E for her constant guidance and input to our project, Prof. Poonam Bhogle and Prof. Prasanna Shete also from K.J.S.C.E for their essential insight. We would also like to extend our heartfelt gratitude towards all the executives at Vihang Honda Service Centre, Thane, India, for their suggestions and assistance in gaining product specific knowledge of the Honda Unicorn CB Dazzler motorcycle. Lastly we would like to thank our parents and friends for their continuous support and motivation in making this project possible.