HIGH SPEED REAL-TIME EMBEDDED SYSTEM FOR PAVEMENT TEXTURE MEASUREMENTS

by

SACHIN JAYARAM

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2014

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr. Roger Walker for guiding me throughout my master career with his invaluable advice in my research work. His guidance has helped me to complete my research and writing of my thesis.

I would like to thank Dr. Jonathan Bredow and Dr. Michael T Manry for being a part of my thesis committee and taking interest in my research.

I genuinely appreciate the guidance and help received from my peer Vikram Simha and the help of all my friends for their moral support.

Finally I would like to thank my parents and my family for their constant encouragement throughout my student career which has helped me to reach where I'm today.

November 12, 2014

Abstract

HIGH SPEED REAL-TIME EMBEDDED SYSTEM

FOR PAVEMENT TEXTURE

MEASUREMENTS

Sachin Jayaram, Master

The University of Texas at Arlington, 2014

Supervising Professor: Roger Walker

The Texas Department of Transportation (TxDOT) collects road profile data from the state highway network and processes the same in order to determine the various anomalies on the road surface. These factors can adversely affect the characteristics and performance of the vehicle such as skid, fuel consumption, ride comfort, safety and tire wear. The International Roughness Index (IRI) and Mean Profile Depth (MPD) are the statistics used for quantifying ride and texture.

Pavement texture is of particular importance as it can affect skidding in wet weather conditions. This thesis mainly focuses on the development and implementation of a real-time embedded system which measures macro and mega texture and computes MPD at highway speeds. The system is based around Intel's multicore NUC, permitting the use of parallel processing for the simultaneous measurements and calculation of the MPD. The texture module includes a 78 KHz LMI laser and a distance encoder. MPD section locations are maintained by GPS position data. The research is funded in part by Intel Corporation.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

INTRODUCTION

The Texas Department of Transportation (TxDOT) collects road profile data from the state highway network[12] and processes the same in order to determine the various anomalies on the road surface. These factors can adversely affect the characteristics and performance of the vehicle such as fuel consumption, ride comfort, safety and tire wear. Road profile information is of great significance to manage roads and to follow up on road conditions.

1.1 Objective

The main objective of the current Research is to develop and implement an embedded system which computes the texture profile and calculates the mean profile depth in real time. The system uses a combination of GPS and distance measurements to identify the exact location of each texture 100mm section. The use of parallel processing by multi-threading allows the multi core processor, The Next Unit of Computing (NUC), to perform simultaneous texture measurements and gather GPS data using the Global Sat GPS module.

Pavement texture is defined by the irregularities on a pavement surface that deviate from an ideal, perfectly flat surface. ISO 13473-1 [3] differentiates the road texture into 4 different categories based on the texture wavelength.

- Micro texture - wavelengths smaller than 0.5 mm.
- Macro texture - wavelengths between 0.5 mm and 50 mm.
- Mega texture - Wavelengths between 50 mm and 0.5 m.
- Unevenness - Wavelengths between 0.5 m and 50 m.

A real-time system is one which “controls an environment by obtaining data, processing them, and returning the results as quickly as possible to affect the environment at that instant of time.”[4]

Global positioning system (GPS) currently is available in more than 3 billion devices throughout the world. MPD section locations are maintained by GPS position data.

1.2 Organization of Thesis

The next chapter focuses on the background and basic definitions related to texture. Chapter 3 focusses on the system overview containing the main hardware components such as the texture module containing DT9816, signal interface board and power board, GPS module BU-353S4 and the Next unit of computing (NUC) multi-core processor which are all required for the real time texture measurement. Chapter 4 mainly discusses the steps involved in the calculation of MPD and GPS coordinates and the algorithms. A simulation was carried out to verify real time calculation of MPD and to simultaneously receive GPS coordinates. The process has been explained in chapter 5. Chapter 6 provides the results and plotting of the real time run and the conclusion.

Chapter 2

Thesis Background

2.1 Motivation for Texture Project

Over the years the transportation department has been interested in the skid characteristics of pavements as skid is a major factor which determines the safety of highways. The texture of the surface is mainly responsible for determining skid resistance and frictional properties of a pavement in contact with a given tire. Two main components of texture are: macrotexture and microtexture [2].

Pavement macrotexture is defined as the deviation of a pavement surface from a true planar surface with a dimension along the surface of 0.5mm to 50mm as discussed in chapter 1. Macrotexture determines the rate at which skid-resistance decreases with increasing speed. Results have shown that pavements with more macrotexture indicated a smaller rate of skid resistance with increasing speed.

Pavement microtexture is defined as the deviation of a pavement surface from a true planar surface with a dimensions along the surface of less than 0.5 mm. It is believed that at low vehicle speed, the microtexture is mainly responsible factor in determining the skid resistance of a pavement. The contact of a tire and the pavement surface is due to the sharp fine particles in a surface, which make up the microtexture that penetrate the thin water film.

It is apparent from above knowledge that both macrotexture and microtexture are of at most importance when determining the skid resistance of the surface. In order to calculate these components, lasers are used as they provide most accurate measurements and high resolution lasers are required for microtexture measurements because of the smaller size. In this project we are using the high resolution Selcom laser for the calculation of both macrotexture and microtexture.

2.2 Texture related terms [2]

a) *Pavement Texture*: The deviation of a pavement surface from a true planar surface.

b) *Profile*: The surface profile is generated if a sensor, like a laser spot or a tip of a needle continuously touches the pavement surface while moving along the surface. Two coordinates describe the surface of the profile, one along the surface plane called distance and the other in a direction perpendicular to the surface plane called amplitude.

c) *Distance*: it is one of the coordinates of the profile of a surface along the surface plane. The distance can be measured in a traverse or longitudinal direction in relation to the travel direction on a pavement, or in any direction between them.

d) *Amplitude*: it is the other coordinate of profile of the surface perpendicular to the surface plane.

e) *Texture Wavelength*: The minimum distance between periodically repeated parts of the curve is defined as the Texture Wavelength and is measured in meters (m) or millimeters (mm).

f) *Texture Depth*: In three dimensional case, it is defined as the distance between the surface and the plane through the top of the three highest particles which are well spaced within a surface area in the same order of a size as that of a tire/pavement surface.

g) *Profile Depth*: it is the difference between the amplitude measurements of pavement macrotexture and a horizontal line through the top of the highest peak within a given baselength.

h) *Mean Profile Depth*: The average value of the profile depth over a certain distance (base length).

i) *Estimated Texture Depth (ETD)*: is obtained from the calculation of mean profile depth using the below mentioned formula.

$$ETD = 0.2 + (0.8 * MPD), \text{ all the dimensions are in mm.}$$

2.3 Advantages of Texture measurements

- Monitor road network conditions for pavement management systems and skid,
- Grade the quality of repaired or newly constructed sections and
- Research the condition of specific sites.

In chapter 3 we will be discussing the various hardware tools being used and their characteristics.

Chapter 3

System Overview

In Chapter 2, we mainly discussed the topic regarding the motivation behind texture project, basic definitions and the advantages. During the process of measuring Mean Profile depth (MPD) using the Texture Module, various sensors are used to calculate data which are converted to digital values and passed on to the NUC processor present inside the module. The module is connected to the accelerometer, power, distance encoder, and infra-red start signal using four different connecters. The texture program downloaded onto the NUC processor is interfaced with the data acquisition board DT9816 and stores the sensor data on to a text file which is processed in both real time and non-real time. The BU-353S4 module is used to obtain the coordinates of the location where the profile is being calculated.

The processor has been coded to work in real time where in the data collected from the laser would be processed producing the results instantaneously. The texture module is mounted in the front end of the van and the measurement of road surface is done with the help of single point laser. The setup would look like the one in fig 3.1 [2].

Figure 3.1 Texture Laser System

3.1 Signal Interface Board

The signal interface board is interfaced with the texture laser system. The board uses a +12V to +/-15V DC-DC converter to power the accelerometer. Two 100 Hz analog filters from Frequency Devices are used to filter the accelerometer and laser analog outputs when the texture laser system is used to compute profile. No external filter is used for texture measurements. The circuit diagram for the signal interface board is shown in Figure 3.2 [2].

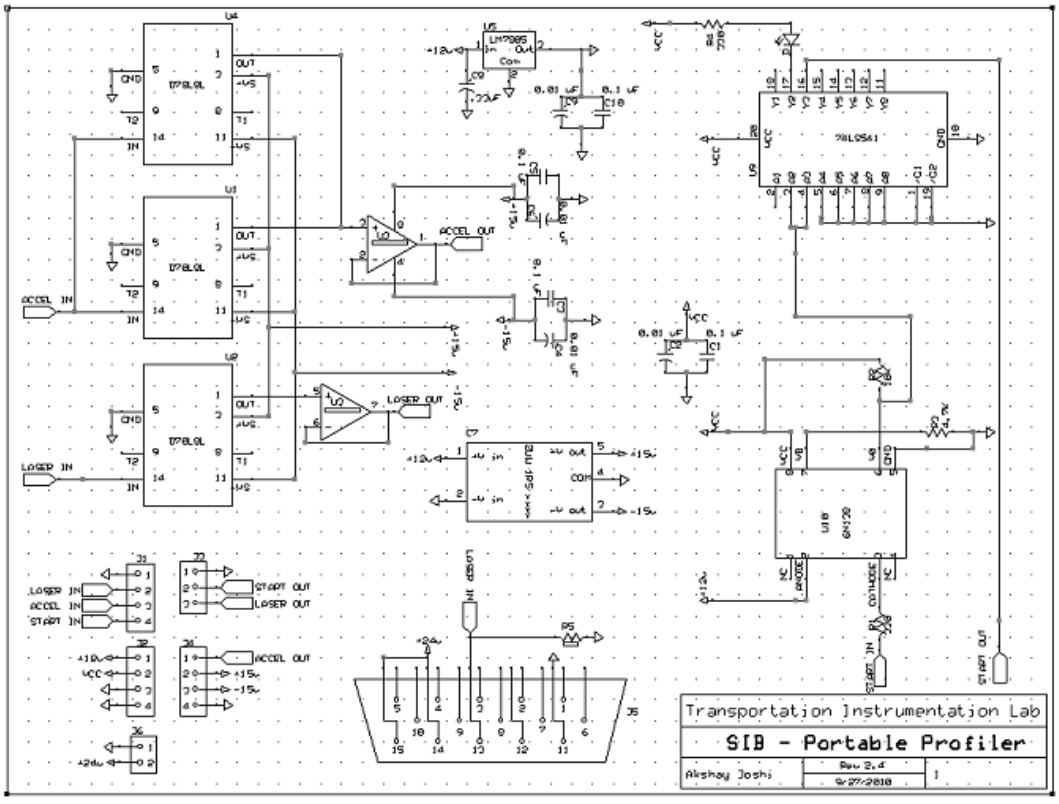

Figure 3.2 Signal Interface Module

Texture Laser Power: Figure 3.3 [2] provides connections between the two boards and the DT9816 module.

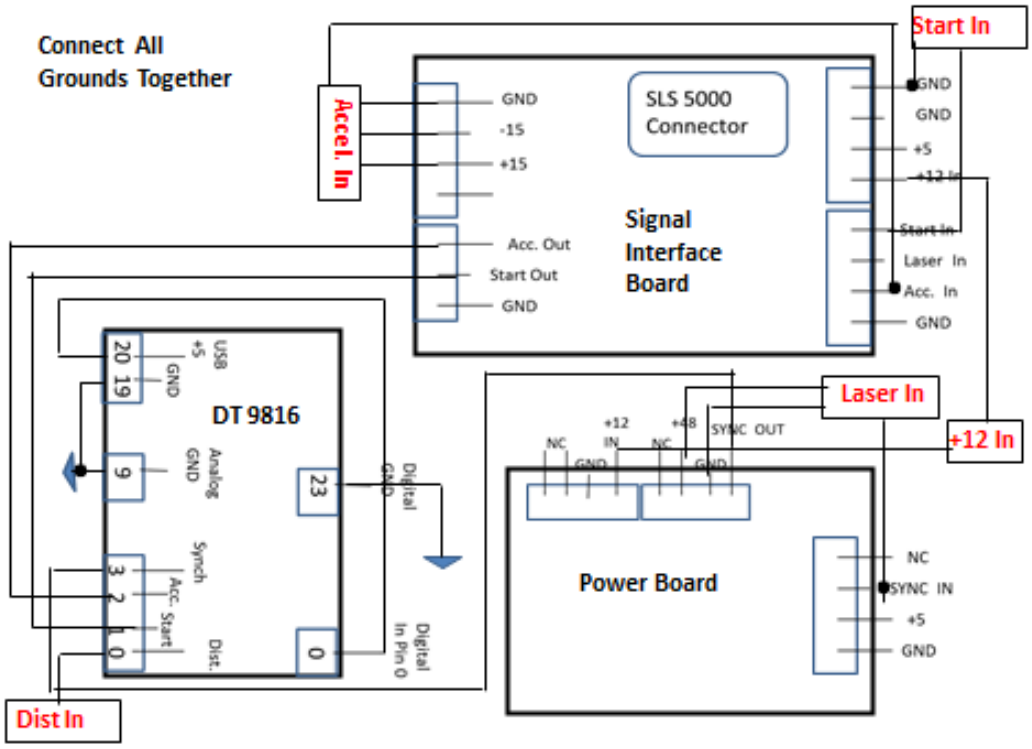

Figure 3.3 Signal Interface Board, Power Board and DT9816 Connections

Texture Laser Profiler: 78KHz texture laser has been used to construct a profile/texture module. The texture module when used in the profile mode provides certifiable profile data. Figure 3.4 [2] provides a figure of the texture module attached to the front end of the van.

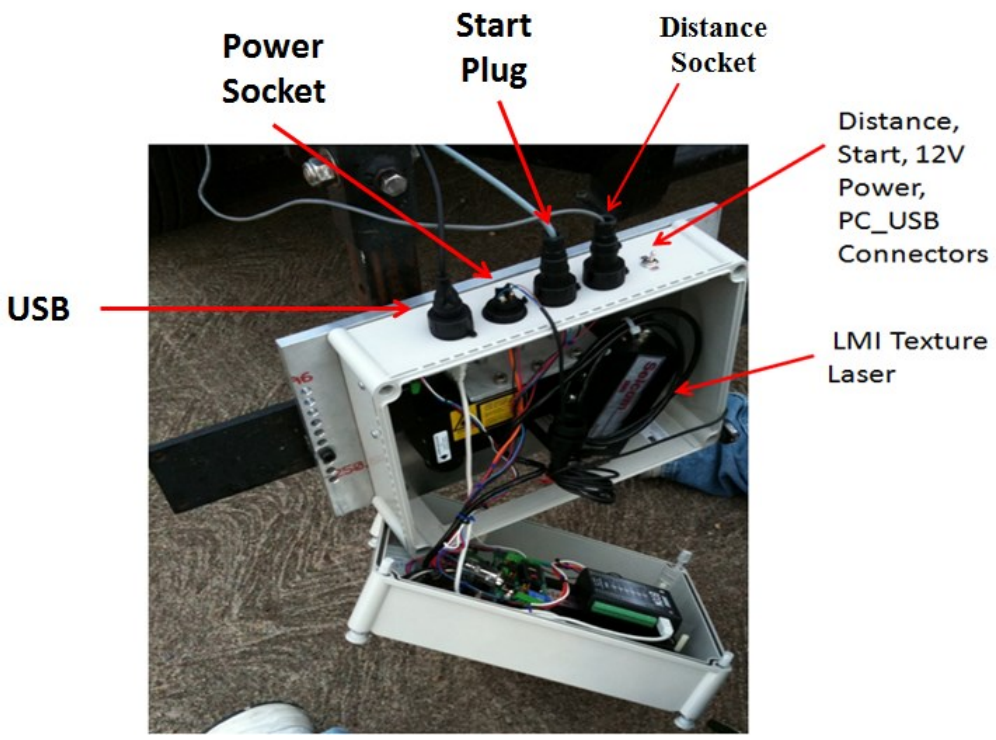

Figure 3.4. Texture Module Attached to a Van

Mean profile depth is the method used to measure texture [1]. As we noted earlier MPD is computed by determining the average of the height of the highest peaks in two equal segments of a 100 mm section of texture profile. The method followed in order to remove the outliers was to look for values greater than certain number of standard deviations, σ, from the mean, μ, of each 100 mm segment. The outliers are always removed before computing the MPD using the following equation:

90% outliers = |Reading ± mean| > 1.65*standard deviation.

3.2 Introduction to GPS Module

The BU-353S4 is a USB magnet mount GPS receiver that features a highly sensitive, low power consumption chipset in an ultra-compact form factor compatible with Microsoft Windows operating system. SiRF star IV chipset powers the BU-353S4 which provides a superior performance in urban canyons and in dense foliage. Micro power mode available on the BU-353S4 allows the receiver to work in a hot start-like condition almost continuously while consuming very little power. The BU-353S4 GPS module was chosen as it is suitable for automotive navigation which would assist in providing the location update where the road profiling takes place. Fig 3.5[6] provides the GPS module used to obtain the current location.

Figure 3.5. BU-353S4 GPS Module

Most computer programs that provide real time position information expect data to be in NMEA format. This data includes the complete PVT (Position, Velocity and Time) solution computed by the GPS receiver. The GPS receiver has a two letter prefix 'GP' that defines the device which uses that sentence type which is followed by a three letter sequence that defines the sentence contents. The main idea of NMEA is to transmit a line of data called a sentence that is totally self-contained and independent with respect to other sentences. There are standard sentences for each device category and there is also the ability to define proprietary sentences for use by the individual company [7].

The beginning of each sentence starts off with a '$' and ends with a carriage return/line feed sequence which cannot be more than 80 characters. The entire data is present in a single line with different entities separated by a comma, hence comma is used as a delimiter to extract individual information. The last field of each sentence is the checksum which begins with a '*' and two hex digits representing an 8bit exclusive OR of all the characters except the '$' and '*'.

3.2.1 BU-353S4 Hardware Connection

The hardware interface for BU-353S4 GPS module is designed to meet the NMEA requirements which follows its standard of 4800 b/s (bit per second rate) with 8 bits of data, no parity, and one stop bit. A total of 480 bits can be transmitted in one second at a baud rate of 4800 b/s and a total of 6 sentences are transmitted as the BU-353S4 output with each line having 82 characters. The hardware is connected to the processor using the USB cable [7].

3.22 NMEA Sentences and Decoding

First word of every sentence in the NMEA format is called a data type which basically defines the interpretation of the remaining part. Each data type has its own interpretation and is defined in the NMEA standard. The different sentences provided by BU-353S4 are $GPGGA, $GPGSA, $GPGSV, $GPRMC respectively which are the most important NMEA sentences, wherein the $GPGGA gives the fix information, $GPGSA provides overall satellite data, $GPGSV provides the detailed satellite data and $GPRMC gives the recommended minimum data for GPS [7].

GGA – essential fix data which provides 3D location and accuracy data.

$GPGGA,114552.000,3243.9722,N,09706.8054,W,1,07,1.4,190.9,M,-23.7,M,0000*68

Where:

GGA	Global Positioning System Fix Data
123519	Fix taken at 11:45:52 UTC
4807.038,N	Latitude 32 degree 043.9722' N
01131.000,E	Longitude 97 degree 06.8054' W
1	Fix quality: 0 = invalid
	1 = GPS fix (SPS)
	2 = DGPS fix
	3 = PPS fix
	4 = Real Time Kinematic
	5 = Float RTK
	6 = estimated (dead reckoning) (2.3 feature)
	7 = Manual input mode
	8 = Simulation mode
07	Number of satellites being tracked

1.4	Horizontal dilution of position
190.9, M	Altitude, Meters, above mean sea level
-23.7, M	Height of geoid (mean sea level) above WGS84 ellipsoid

(empty field) time in seconds since last DGPS update

(empty field) DGPS station ID number

*68	the checksum data, always begins with *

GSA – mainly provides the information regarding the nature of the fix. Also includes the number of satellites being used in the current solution and the DOP (dilute of precision), which is an indication of the effect of satellite geometry on the accuracy of the fix.

$GPGSA,A,3,17,08,04,26,07,15,24,,,,,,2.0,1.4,1.5*3A

Where:

GSA	Satellite status
A	Auto selection of 2D or 3D fix (M = manual)
3	3D fix - values include: 1 = no fix 2 = 2D fix 3 = 3D fix
17, 08...	RNs of satellites used for fix (space for 12)
2.0	PDOP (dilution of precision)
1.4	Horizontal dilution of precision (HDOP)
1.5	Vertical dilution of precision (VDOP)
*3A	the checksum data, always begins with *

GSV – Satellites in view shows the data about the satellites that the unit might be able to find based on it viewing mask and almanac data. One GSV sentence can provide data up to 4 satellites and hence requires three sentences to get the complete information.

$GPGSV,3,1,12,17,68,307,12,08,37,159,44,04,29,192,34,26,18,247,26*7C

Where:

GSV	Satellites in view
3	Number of sentences for full data
1	sentence 1 of 2
12	Number of satellites in view
17	Satellite PRN number
68	Elevation, degrees
307	Azimuth, degrees
12	SNR - higher is better for up to 4 satellites per sentence
*26	the checksum data, always begins with *

RMC – NMEA has its own version of essential GPS PVT (Position, Velocity and Time) data which is represented in the RMC sentence.

$GPRMC,000050.000,A,3243.9718,N,09706.8053,W,0.29,164.24,220514,,,A*70

Where:

RMC	Recommended Minimum sentence C
123519	Fix taken at 12:35:19 UTC
A	Status A=active or V=Void.
3243.9718,N	Latitude 32 deg 43.9718' N
09706.8053,E	Longitude 97 deg 06.8053' W

0.29	Speed over the ground in knots
164.24	Track angle in degrees True
220514	Date – 22nd of May 2014

3.3 Next Unit of Computing

The Intel Next Unit of Computing (Figure 3.6 [8]) DC3217BY is the processor which is performing the all the calculations. The NUC provides multi-processing capability and occupies very little space and can be fit into the texture module. It has a visibly smart Intel® Core™ i3-3217U (3M Cache, 1.80 GHz) processor, the next unit of computing is a full scalable and complete unit of computing using the smallest possible form factor. One of the main reasons for choosing the NUC processor is because of its multi-processing capability, size and the cost.

Figure 3.6 NUC Processor

The next chapter entails a brief introduction on the Texture algorithm for calculating the mean profile depth and GPS data retrieval through User datagram protocol(UDP).

Chapter 4

Mean Profile Depth and GPS Algorithm

In chapter 3, the different hardware components used for the project were discussed. In this chapter, we mainly focus on how mean profile depth is calculated both in real time and post processing based on whether the data is being calculated in real time or already available. The various steps involved in calculation of MPD and GPS data are discussed in the following section.

The user is asked to choose between real time and post processing and either for continuous or unique values of GPS locations.

4.1 Real Time Processing

In real time processing the analog output from the single point laser is connected to the DT9816 data translation board which provides 16-bit resolution and gains of 1 and 2 for effective full-scale input signal ranges of ±10 V and ±5 V. The digital values obtained from the output of the data translation board is used to calculate the mean profile depth. A server program is written to obtain the GPS data through the Com port using the BU-353S4 GPS module. The GPS coordinates is saved in a queue buffer which is obtained and concatenated with the MPD values whenever there is a change in coordinates or at every instant of MPD calculation based on the user selection. Mean profile depth and the location of the coordinates are noted down on a text file as and when the data gets collected in real time.

4.11 Analog to Digital Conversion

DT9816 - System Operations

The DataAcq SDK [5] provides functions for the following general system operations:

1) Initializing a device

2) Specifying a subsystem

3) Configuring a subsystem

4) Calibrating a subsystem

5) Handling errors

6) Handling messages

7) Releasing a subsystem and driver

1) Initializing a Device

To perform a data acquisition operation, your application program must initialize the device driver or the device you are using with the olDaInitialize function. This function returns a device handle, called HDEV. You need one device handle for each device. Device handles allow you to access more than one device in your system.

If you are unsure of the DT-Open Layers devices in your system, use the olDaEnumBoards function, which lists the device name, device driver name, and system resources used by each DT-Open Layers device in your system, or the olDaGetBoardInfo function, which returns the driver name, model name, and instance number of the specified board, based on its board name.

2) Specifying a Subsystem

The DataAcq SDK allows you to define the following subsystems:

1) Analog input (A/D subsystem)

2) Analog output (D/A subsystem)

3) Digital input (DIN subsystem)

4) Digital output (DOUT subsystem)

5) Counter/timer (C/T subsystem)

6) Tachometer (TACH subsystem)

A device can have multiple elements of the same subsystem type. Each of these elements is a subsystem of its own and is identified by a subsystem type and element number. Element numbering is zero-based; that is, the first instance of the subsystem is called element 0, the second instance of the subsystem is called element 1, and so on. For example if two digital I/O ports are on your device, two DIN or DOUT subsystems are available, differentiated as element 0 and element 1.

Once you have initialized the device driver for the specified device, you must specify the subsystem/element on the specified device using the olDaGetDASS function. This function returns a subsystem handle, called HDASS. To access a subsystem, you need one subsystem handle for each subsystem. Subsystem handles allow you to access more than one subsystem on a device.

If you are unsure of the subsystems on a device, use the olDaEnumSubSystems or olDaGetDevCaps function. olDaEnumSubSystems lists the names, types, and number of elements for all subsystems supported by the specified device. olDaGetDevCaps returns the number of elements for a specified subsystem type on a specified device.

3) Configuring a Subsystem

You configure a subsystem by setting its parameters or capabilities. Call the olDaConfig function to configure the parameters before performing the operation.

4) Calibrating a Subsystem

Some devices provide a self-calibrating feature, where a specified subsystem performs an auto-zero function. To determine if the specified subsystem supports this capability, call the olDaGetSSCaps function, specifying the capability OLSSC_SUP_AUTOCAL. If this function returns a nonzero value, the capability is supported. To calibrate the subsystem, call the olDaAutoCalibrate function. Note that the subsystem must be stopped before calling this function, or an error is returned.

5) Handling Errors

An error code is returned by each function in the DataAcq SDK. An error code of 0 indicates that the function executed successfully (no error). Any other error code indicates that an error occurred. Your application program should check the value returned by each function and perform the appropriate action if an error occurs.

6) Handling Messages

The data acquisition device notifies your application of buffer movement and other events by generating messages. To determine if the subsystem can post messages, use the olDaGetSSCaps function, specifying the capability OLSSC_SUP_POSTMESSAGE. If this function returns a nonzero value, the capability is supported. Specify the window to receive messages using the olDaSetWndHandle function or the procedure to handle these messages using the olDaSetNotification Procedure function.

7) Releasing the Subsystem and the Driver

When you are finished performing data acquisition operations, release the simultaneous start list, if used, using the olDaReleaseSSList function. Then, release each

subsystem using the olDaReleaseDASS function. Release the driver and terminate the session using the olDaTerminate function.

Before Configuring the Subsystem, Data Flow Modes are selected. The DataAcq SDK defines the following data flow modes for A/D, D/A, DIN, and DOUT subsystems.

1) Single value

2) Continuous

In this project continuous A/D subsystem is used in order to calculate the digital values from the analog input available from the laser.

Continuous Operations

For a continuous operation, you can specify any supported subsystem capability, such as a channel-gain list, clock source, trigger source, buffer, and so on, and then configure the subsystem using the olDaConfig function. The olDaStart function is used to start a continuous operation.

In order to stop a continuous operation, perform either an orderly stop using the olDaStop function or an abrupt stop using the olDaAbort or olDaReset function. In an orderly stop (olDaStop), the device finishes acquiring the specified number of samples, stops all subsequent acquisition, and transfers the acquired data to a buffer on the done queue, whereas In an abrupt stop (olDaAbort), the device stops acquiring samples immediately, the acquired data is transferred to a buffer and put on the done queue. Figure 4.1 [5] explains the function of continuous analog input operation in DT9816.

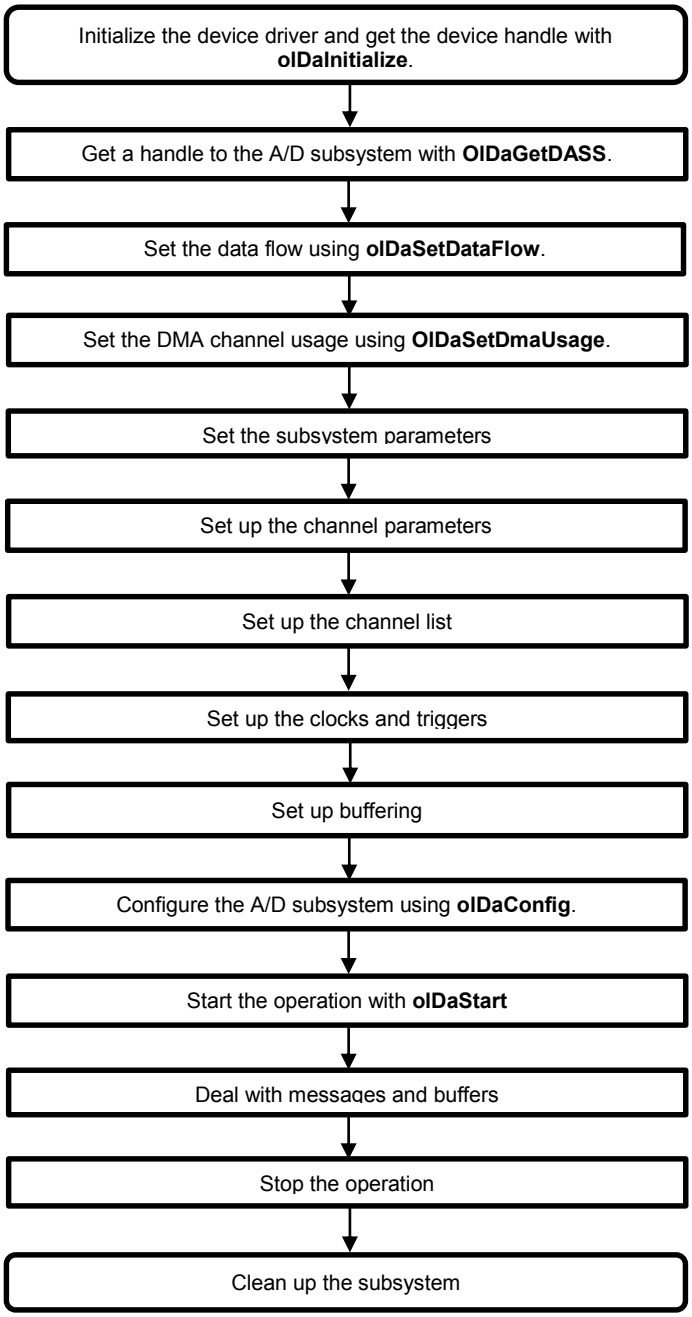

Figure 4.1 Continuous Analog Input Operation

Based on the sampling frequency the data is obtained continuously in digital form which is processed to calculate the mean profile depth. Since DT9816 is a 6 different successive approximation A/D converter, six different values are obtained which represent the start signal, distance signal, 2 laser (one filtered and the other one without filtering) and accelerometer readings.

4.12 Calculation of MPD

The channel 1 is continuously monitored to check for the start signal and as soon as the start signal is received the process is initiated. The channel 0 represents the distance data which helps to calculate an interval of 100mm +/- 2mm. The channel 4 is continuously read in order to collect the texture laser data in digital form from the DT9816 and is stored for every 100mm. The data gets divided into segments each having a base length of 100 mm (3.9 in.). The slope of all the segments are suppressed by subtracting the linear regression of the segment. Further the segment is divided into half and the height of highest peak in each half of the segment is calculated. The difference between the average height value and the height of the measured average of each segment is reported as the Mean Profile Depth (MPD). The results of this calculation has helped to predict the speed dependence of wet pavement friction [1].

There is a possibility of obtaining an invalid reading as a result of deep surface troughs or local photometric properties of the surface. In order to eliminate it, these readings must be replaced when the value is higher or lower than the range of profile surrounding their location. The invalid value is replaced with a value interpolated between the previous and following location [1]. The maximum proportion of outliers is limited to 20% and when it exceeds 10%, caution should be used in interpreting the data and

proportion of invalid reading is reported. The procedure for calculating the mean segment depth is shown in Fig 4.2 [1].

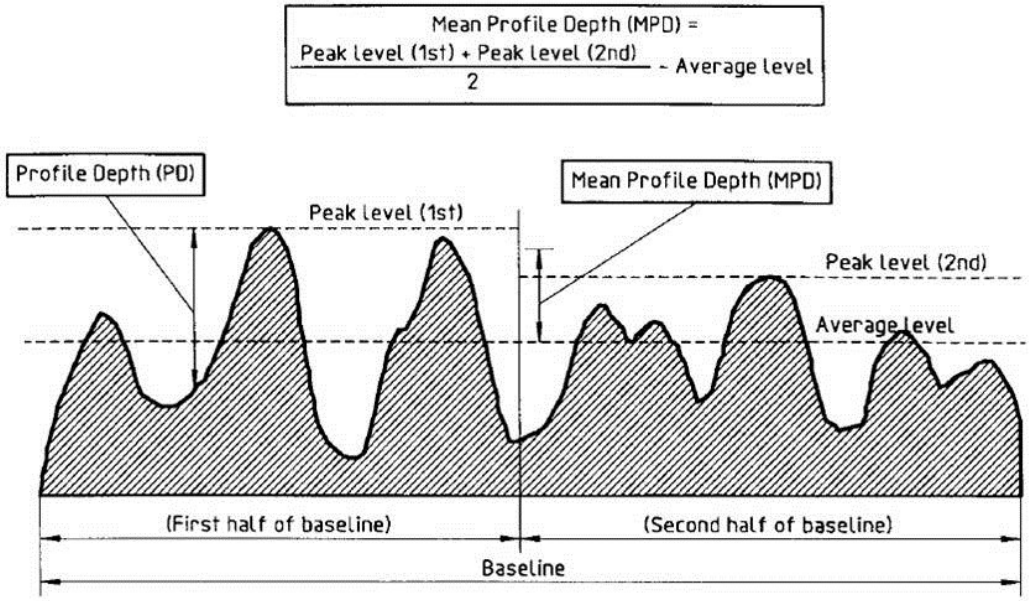

Figure 4.2 Procedure for computation of Mean Segment Depth

4.13 GPS data retrieval

In real time operation the BU-353S4 module is connected to the main texture module in order to obtain the coordinates of the location where the data is being collected. The GPS module is connected to one of the COM port of the NUC processor. The server program is written in order to obtain the GPS coordinates transmits the same to the main program using UDP sockets where in the complete data is parsed to get the individual entities like latitude, longitude, altitude, heading, speed and timestamp which is stored in a queue. This data is concatenated along with the MPD values and written on to a text file as the final output. The following section explains the method used to receive GPS values from a COM port [9].

1) Creating a port handle

The serial port's handle is the handle that can be used to access the object of serial port. CreateFile function is used to create the serial port handle. The following code explains the process of creating a handle.

```
HandlePort = CreateFile (portName,                    // Specify port device: "COM3"
            GENERIC_READ | GENERIC_WRITE, // Specify mode that open device.
             0,                                       // the device isn't shared.
            NULL,                                    // the object gets a default security.
            OPEN_EXISTING,                    // Specify what action to take on file.
            0,                                       // default.
            NULL);                                  // default.
```

2) Restoring a configuration

The GetCommState function is used to get the current device-control and then fills to a device-control block (a DBC structure) with the current control settings for a specified communications device.

3) Modifying a configuration

The configuration has to be changed according to the hardware requirements, for the GPS module we use 4800 baud rate, 8 bit data size, no parity and one stop bit is being used.

4) Storing a configuration

The next step is the storage of new configuration that is modified already into device control. The SetCommState API function is called to store the configuration. The SetCommState function configures a communications device according to the specifications in a device-control block (a DCB structure).

5) Receiving Data

The ReadFile function is used to read the data and store it in a buffer which is then transmitted to the main program using UDP sockets. Most of the data reception is done as reading a file.

4.14 Multi-threading

Once the GPS data is received from the COM port, it is being sent to the client program which is performing the MPD calculation. The server uses UDP sockets to transmit the data to the client using Multi-threading. Multi-threading helps in the simultaneous execution of the data retrieval and transmission. In the client program the reception is done using Multi-threading as well and hence the MPD calculation and the GPS data are acquired simultaneously since we are using the next unit of computing (NUC) multiprocessor.

4.2 Post Processing

At the beginning of the program the user is asked to choose either between Real time or Post processing operation. If post processing is chosen then the road surface data is already available, stored as a .dat file. The .dat file has the six channel data collected, which are parsed and read by the main program. Once the data are tokenized, the six channel data are obtained separately which are the Start, Distance, Laser left, Accelerometer left, Laser right and Accelerometer right. These individual data are then used to calculate the mean profile depth in the same fashion as how it is explained earlier in this chapter. After the calculation of Mean profile depth, the values are written on to a text file which is used as a reference to understand the texture of the road surface.

Real time operation is a big advantage as the MPD of the surface is calculated on the go and the nature of the surface can be studied instantaneously. Chapter 5 we will be discussing the implementation of the system using the concept mentioned in here.

Chapter 5

System Implementation by Simulation

This chapter focuses mainly on the initial implementation of the system by simulation on the multi core processor with some of the different hardware units listed in chapter 3 and the methods that are used in the chapter 4. A brief introduction to UDP sockets is also discussed in the latter part of the chapter. Throughput and resource usage are two of the most important issues to consider in the design of the system. The system was setup to work in the laboratory for simulation with the help of IDL board and was then tested later with the actual texture module.

5.1 Simulation using IDL board

The IDL-800 digital Lab offers a unique platform on which digital electronics circuits may be constructed with speed and ease. The system combines simple, easy to use, features with a versatile solder less breadboard area [11]. Figure 5.1[13] represents the basic IDL 800 digital board. The unit has most of the necessary equipment built in which avoids the need for an external equipment. The availability of switches which indicated the digital voltages from 0V to 5V was one of the main reasons to choose the respective board.

Figure 5.1 IDL 800

The start signal, distance signal and the laser data was simulated in the laboratory using the IDL board. The .dat file obtained from the run was used as the reference for simulation. One of the 5 inputs from the IDL board was the one which produced a digital value of -5V, 0V or 5V to obtain the start signal where the value being output is reduced to less than 2.5V to indicate the starting point, a square wave was used to represent the distance signal to produce pulses at known intervals which helped to calculate the distance traversed and the third input was the variable voltage connection which could be varied to indicate the presence of texture manually during the run.

The three IDL board inputs were fed as input to the DT9816 analog input channels and the output was connected to a multi core processor which converted the

continuous incoming analog input values to digital in real time. These digital values were then processed again in real time to obtain the MPD values along with the GPS coordinates. These values were documented onto a text file and stored. Once the simulation was successful, the same procedure was carried on a texture module. Figure 5.2 represents the setup used to simulate the calculation of Mean Profile depth and obtain the GPS locations.

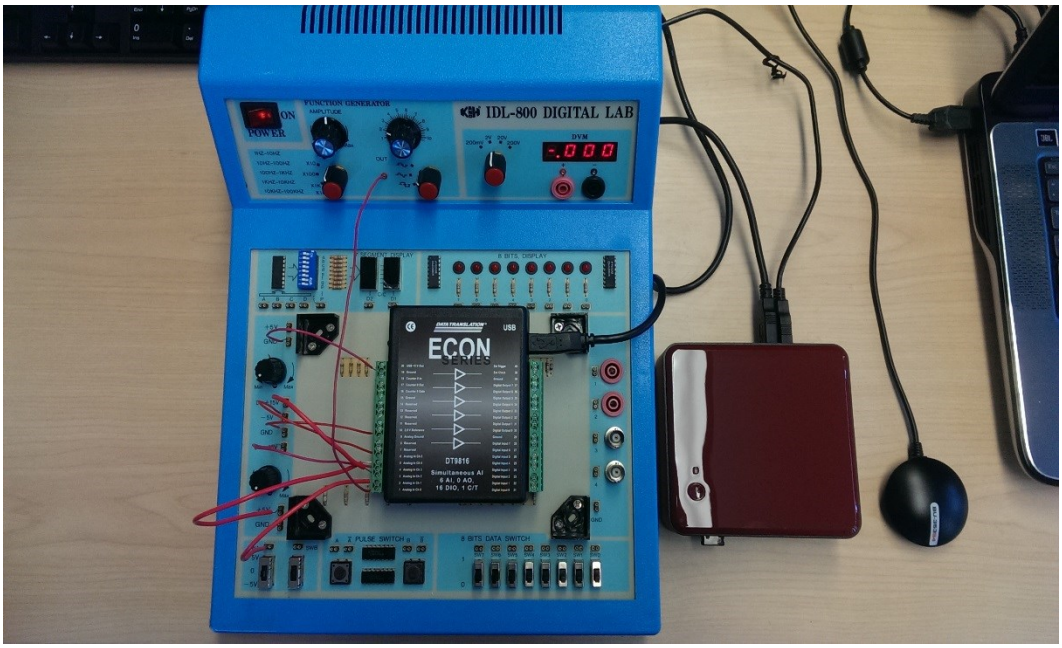

Figure 5.2. Calculation of MPD Simulation in Real Time

The above figure shows the connections of IDL and the data translation DT9816 board. The GPS module and the DT9816 outputs are connected to the Next unit of computing (NUC) multi core processor.

The start signal and the distance signal were simulated similar to the way as obtained from one of the real time runs with the texture being manually produced by the user by varying the voltages and the MPD for those values were calculated in real time.

5.2 Simulation Results

The results were tabulated and plotted once the simulation in real time was completed. Figure 5.3 represents the MPD values obtained for the texture data produced by the user using the simulation board.

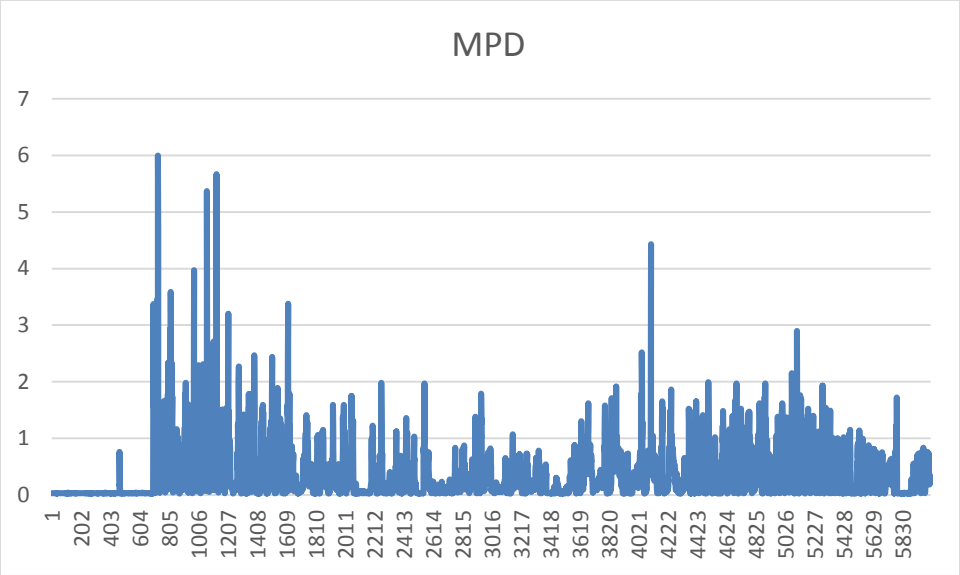

Figure 5.3 MPD for manually generated texture in real time by simulation

As and when the MPD value is collected the GPS values currently on the top of the queue is obtained and concatenated along with the MPD value and written on to a text file, which would be analyzed.

Table 5.1 represents the output format containing both the MPD values and GPS coordinates. As the simulation run took place in the lab, GPS coordinates remained the same throughout.

Table 5.1 Output Text file containing MPD and GPS coordinates

MPD	LATITUDE	LONGITUDE	ALTITUDE	HEADING	SPEED	TIME
0.41	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.31	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.33	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.39	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.33	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.41	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.69	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
1.31	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.38	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.38	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.42	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.40	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.39	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.35	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.34	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.34	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.86	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.91	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
1.42	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
1.31	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.45	3243.9929	9706.8144	166.3999	329.5899	0.89	175320
0.64	3243.9929	9706.8144	166.3999	329.5899	0.89	175320

5.3 UDP Sockets

A socket is an interface between an application and a network. The application is responsible for creating a socket and the socket type indicates the style of communication i.e. either reliable or best effort and connection or connectionless oriented [14]. Once the setup is configured the application is able to transfer data to the socket using network transmission and receive as well. There are two types of sockets, TCP/IP or SOCK_STREAM and UDP or SOCK_DGRAM. In this project the communication is done through UDP (User Datagram Protocol). The server socket is collecting the GPS data from a com port and transmitting it to the client using UDP. The different steps involved in the creation of server sockets are socket initialization, receiver address structure setup, sending data to the client by defining a send socket and using a sendto API. The client socket requires socket initialization followed by setting up of a receiver socket, binding it and then receiving the data using the recvfrom API. The system was initially built to support TCP/IP using the help of MSDN (The Microsoft Developer Network) [10] and then was converted to UDP due to TXDOT specifications.

The next chapter discusses the results of the Real time run along with the future work that can be done to maximize the throughput.

Chapter 6

Real Time Implementation and Conclusion

In the previous chapter we discussed the system implementation by simulation, real time implementation was carried out once the results of simulation were analyzed and obtained as per the requirement which were plotted in table 5.2.

Figure 6.1 represents the block diagram of the Real time setup of road profile texture measurements.

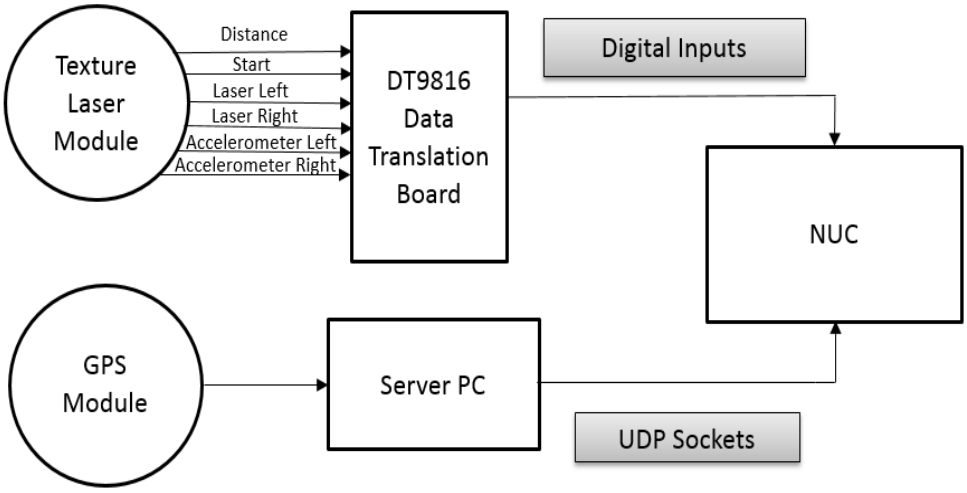

Figure 6.1 Real Time Implementation Block diagram

Over the years the portable profiler/texture module has been used to provide data to compute both profile and macrotexture (using post processing) of the wheel path tested. This led to the investigation to find out any possible effect of texture on the International Roughness Index (IRI) computed on the surface profile. IRI is defined as the roughness measurement statistic used to report pavement roughness. The MPD was calculated on 11 different surface types and then compared amongst themselves and with the IRI to find out if any correlation existed between them.

It was found that there were no major correlation between the IRI and MPD on the sections evaluated. The Correlation coefficient was found to be 0.2 which indicates very less correlation, as a correlation coefficient of 1 indicates perfect correlation and 0 indicates no correlation.

The graph for IRI vs. MPD were plotted to check the correlation and the same has been displayed on Figure 6.2 [2].

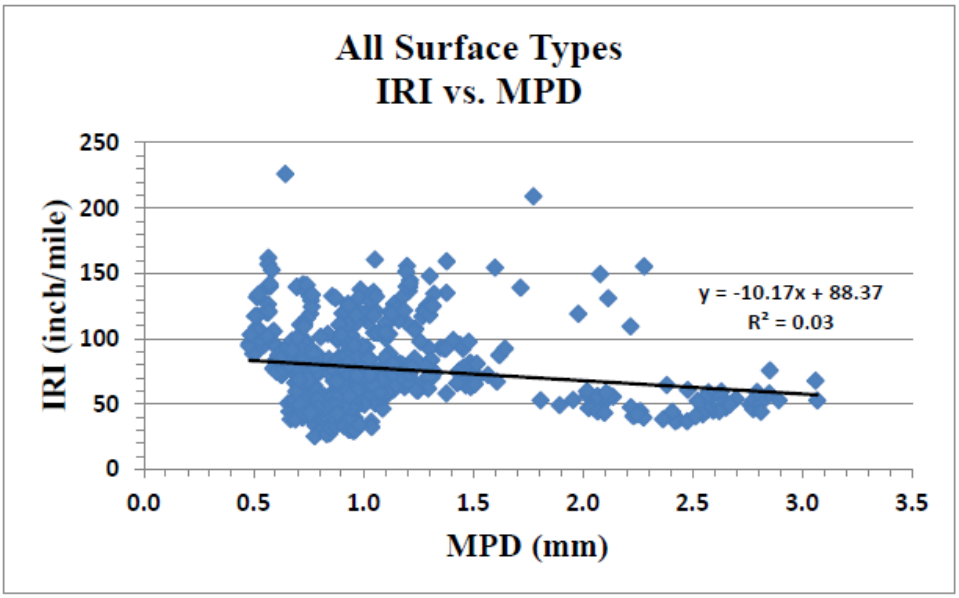

Figure 6.2 IRI vs. MPD for all Surface types

Further the MPD's for different surface types were compared to test the hypothesis that the MPD's across different surface types were the same and was found out that the MPD's differ significantly across various surface types and the hypothesis was disregarded.

The box plots in Figure 6.3 [2] supports this findings. It is seen that the PFC's and the flexible base box plots have considerably higher MPD's compared to others.

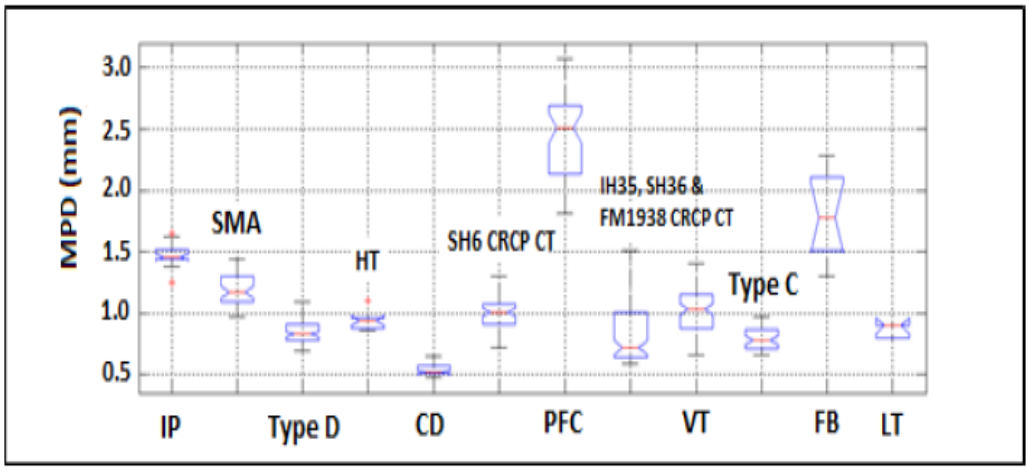

Figure 6.3 Box plots of MPD's per individual sections

6.1 Real Time Run

The Texture module was setup in the lab for a real time run and the surface used to calculate the MPD is shown in Figure 6.4. The setup is shown in Figure 6.5, Infra-red sensor was used as a start signal, a TTL square wave from signal generator was used as a distance signal and the laser was used to get the texture surface data.

Figure 6.4 Texture Surface used for Real time run

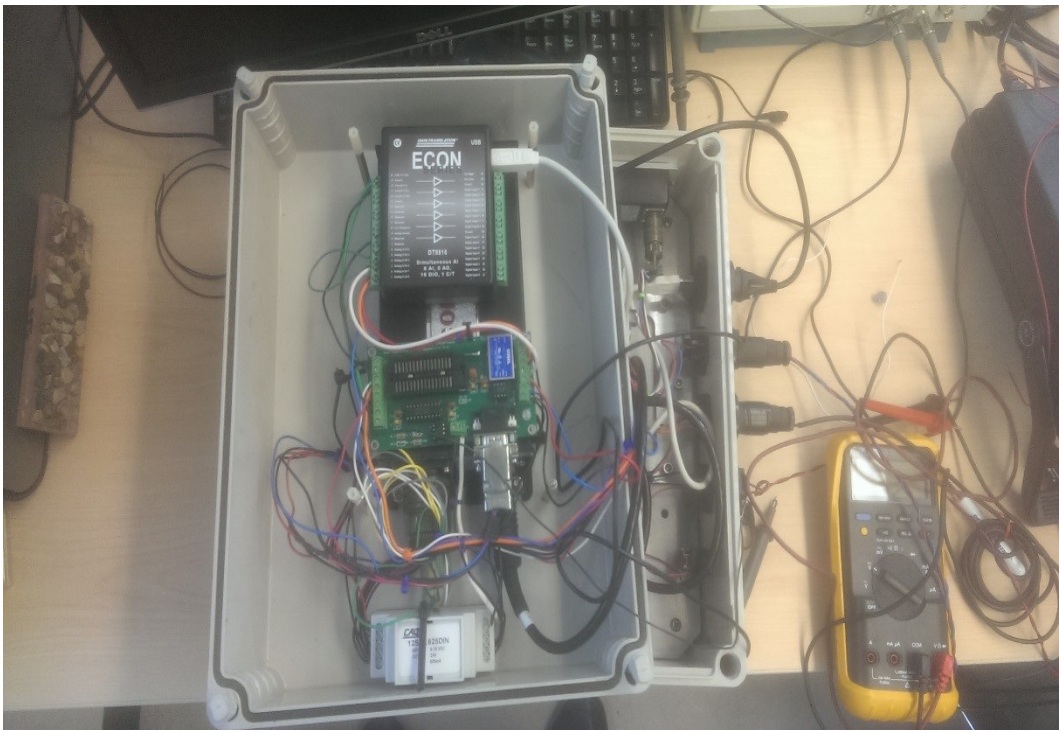

Figure 6.5 Real time run setup

6.2 Data Processing and Results

The real time run was carried out in the laboratory environment on the surface mentioned above in Figure 6.4 and once the data collection was stopped, the program was terminated and the data collected was processed in real time. The MPD calculation and the GPS data retrieval from the GPS server program was done concurrently due to the usage of multi-threading in a multi-core processor.

Once the MPD of each 100m section was calculated in real time, the GPS data was concatenated along with the MPD values with the help of a queue buffer. The GPS program has been written to receive the data every second and is stored in a queue. At the time of writing the MPD value on to a file, the respective GPS value stored in the queue buffer is concatenated. Initially before the run, the user is asked to choose for repeated GPS or unique values and based on the decision, the GPS coordinates are concatenated all the time or only when there is a change in coordinates respectively. Figure 6.6 represents the Real time run GPS data obtained in NMEA standard format which will be transmitted to the main program using UDP sockets.

```
$GPGGA,175311.026,3243.9924,N,09706.8129,W,1,03,5.4,166.5,M,-23.7,M,,0000*6D
$GPGSA,A,2,17,26,15,,,,,,,,,,6.3,5.4,3.2*30
$GPRMC,175311.026,A,3243.9924,N,09706.8129,W,1.00,331.00,070814,,,A*78
$GPGGA,175312.026,3243.9925,N,09706.8135,W,1,03,5.4,166.5,M,-23.7,M,,0000*62
$GPGSA,A,2,17,26,15,,,,,,,,,,6.3,5.4,3.2*30
$GPRMC,175312.026,A,3243.9925,N,09706.8135,W,0.84,331.00,070814,,,A*7A
$GPGGA,175313.026,3243.9921,N,09706.8138,W,1,03,5.4,166.5,M,-23.7,M,,0000*6A
$GPGSA,A,2,17,26,15,,,,,,,,,,6.3,5.4,3.2*30
$GPRMC,175313.026,A,3243.9921,N,09706.8138,W,0.49,331.00,070814,,,A*73
$GPGGA,175314.026,3243.9925,N,09706.8129,W,1,03,5.4,166.5,M,-23.7,M,,0000*69
$GPGSA,A,2,17,26,15,,,,,,,,,,6.3,5.4,3.2*30
$GPGSV,3,1,12,17,61,256,19,26,27,271,21,15,08,305,17,11,29,051,21*76
$GPGSV,3,2,12,28,60,003,07,30,65,154,,08,49,143,,01,42,072,*72
$GPGSV,3,3,12,04,42,159,,07,32,149,,09,20,224,,06,07,192,*7B
$GPRMC,175314.026,A,3243.9925,N,09706.8129,W,1.38,331.74,070814,,,A*74
$GPGGA,175315.026,3243.9925,N,09706.8133,W,1,03,5.4,166.5,M,-23.7,M,,0000*63
$GPGSA,A,2,17,26,15,,,,,,,,,,6.3,5.4,3.2*30
```

Figure 6.6 Real time run GPS data in NMEA format

Table 6.1 Real Time run, Parsed GPS values

Latitude	Longitude	Altitude	Heading	Speed	Time
3243.992432	9706.812532	166.399994	331.624536	1	175311.0313
3243.992432	9706.813477	166.399994	331.223465	0.84	175312.0313
3243.992188	9706.813477	166.399994	331.985421	0.49	175313.0313
3243.992432	9706.8125	166.399994	331.739994	1.38	175314.0313
3243.992188	9706.814453	166.399994	332.758456	1.33	175316.0313
3243.992188	9706.814453	166.399994	331.940002	0.48	175317.0313
3243.992676	9706.813477	166.399994	331.940002	0.54	175318.0313
3243.992432	9706.813477	166.399994	329.589996	1.18	175319.0313
3243.99292	9706.814453	166.399994	329.589996	0.89	175320.0313
3243.99292	9706.814453	166.399994	329.589996	0.75	175321.0313
3243.993164	9706.814453	166.399994	330.26001	1.83	175322.0313
3243.993408	9706.81543	166.399994	330.980011	1.49	175323.0313
3243.993408	9706.81543	166.399994	330.980011	1.11	175324.0313
3243.993652	9706.816406	166.399994	331.190002	2.03	175325.0313
3243.993652	9706.818359	166.399994	330.790009	1.39	175326.0313
3243.993896	9706.818359	166.300003	331.100006	1.19	175327.0313
3243.994385	9706.819336	166.300003	331.089996	2.12	175328.0313
3243.994629	9706.820313	166.300003	331.089996	0.66	175329.0313
3243.99585	9706.821289	166.300003	331.220001	2	175330.0313
3243.99707	9706.822266	166.300003	331.579987	2.62	175331.0313
3243.997803	9706.822266	166.300003	331.619995	3.37	175332.0313
3243.998779	9706.823242	166.300003	330.489993	3.87	175333.0313
3243.998779	9706.824219	166.300003	329.679993	2.73	175334.0313
3243.998779	9706.824219	166.300003	329.700012	1.99	175335.0313
3243.998779	9706.825195	166.300003	329.600006	1.36	175336.0313
3243.998291	9706.825195	166.300003	329.600006	1.12	175337.0313
3243.998779	9706.826172	166.300003	329.690002	1.6	175338.0313
3243.999512	9706.826172	166.300003	329.690002	0.75	175340.0313
3243.999756	9706.825195	166.300003	330.179993	1.55	175341.0313
3244.000488	9706.825195	166.300003	330.940002	3.16	175342.0313
3244.001221	9706.826172	166.300003	331.929993	3.4	175343.0313
3244.001465	9706.827148	166.300003	331.880005	2.36	175344.0313
3244.000977	9706.828125	166.300003	331.170013	2.08	175345.0156
3244.001221	9706.828125	166.300003	331.070007	2.22	175346.0313
3244.001465	9706.828125	166.300003	331.070007	2.22	175347.0313
3244.001709	9706.828125	166.300003	331.049988	2.05	175348.0313

Once the GPS data was obtained in NMEA format on the main client program, the individual required data was parsed and stored in a text file and a queue buffer. Table 6.1 represents the part of the parsed values of Latitude, Longitude, Altitude, Heading and speed which was obtained during the real time run.

The parser was written in such a way that any GPS data received would be first converted into NMEA format if it already isn't in the standard format and then would parse the entire data. Since the run took place in the laboratory the coordinates remain the same for almost the entire duration of the run. Table 6.2 represents the final data obtained from the output text file of the real time run.

These data were then concatenated along with the MPD values and Figure 6.7 displays the data containing MPD along with the GPS coordinates.

The output text file containing the MPD values and GPS coordinates were received from the real time run, a graph was plotted for the MPD values in Excel spreadsheet.

Table 6.2 Real Time Run Final Output

MPD	Latitude	Longitude	Altitude	Heading	Speed	Time
0.41	3243.99243	9706.812532	166.399994	331.624536	1	175311.0313
0.31	3243.99243	9706.813477	166.399994	331.223465	0.84	175312.0313
0.33	3243.99218	9706.813477	166.399994	331.985421	0.49	175313.0313
0.39	3243.99243	9706.8125	166.399994	331.739994	1.38	175314.0313
0.86	3243.99218	9706.814453	166.399994	332.758456	1.33	175316.0313
2.03	3243.99218	9706.814453	166.399994	331.940002	0.48	175317.0313
0.89	3243.99267	9706.813477	166.399994	331.940002	0.54	175318.0313
1.11	3243.99218	9706.814453	166.399994	332.758456	1.22	175316.0313
1.43	3243.99218	9706.814453	166.399994	331.940672	0.38	175317.0313
0.56	3243.99267	9706.813477	166.399994	331.940562	0.74	175318.0313

	A	B	C	D	E	F	G	H
1	MPDL	MPDR	Latitude	Longitude	Altitude	Heading	Speed	Time
2	0.41	0.41	3243.99292	9706.814453	166.399994	329.589996	0.89	175320
3	0.31	0.31	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
4	0.33	0.33	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
5	0.39	0.39	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
6	0.33	0.33	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
7	0.36	0.36	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
8	0.35	0.35	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
9	0.38	0.38	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
10	0.37	0.37	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
11	0.41	0.41	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
12	0.37	0.37	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
13	0.44	0.44	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
14	0.57	0.57	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
15	1.34	1.34	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
16	1.3	1.3	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
17	0.69	0.69	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
18	0.53	0.53	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
19	0.38	0.38	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
20	0.38	0.38	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
21	0.42	0.42	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
22	0.4	0.4	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
23	0.39	0.39	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
24	0.35	0.35	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
25	0.34	0.34	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
26	0.34	0.34	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
27	0.38	0.38	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
28	0.36	0.36	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
29	0.37	0.37	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
30	0.34	0.34	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
31	0.37	0.37	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
32	0.35	0.35	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
33	0.41	0.41	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
34	0.39	0.39	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
35	0.53	0.53	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
36	0.86	0.86	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
37	0.83	0.83	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
38	0.91	0.91	3243.99292	9706.814453	166.399994	329.589996	0.75	175321
39	0.38	0.38	3243.99292	9706.814453	166.399994	329.589996	0.75	175321

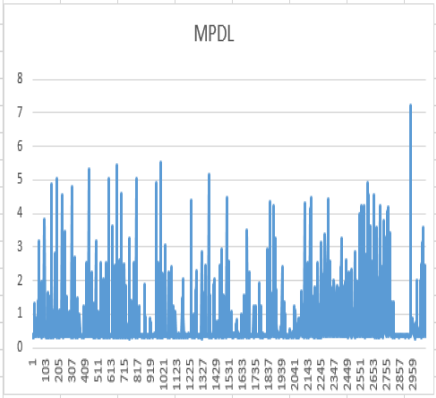

Figure 6.7 Real Time Run GPS data and MPD values

The MPD of the texture surface that was used for the real time run had already been calculated and the resulting figure indicated similar MPD values and hence was verified to be correct. Figure 6.8 represents the graph with MPD values along Y-axis and base length of 100mm along the X-axis.

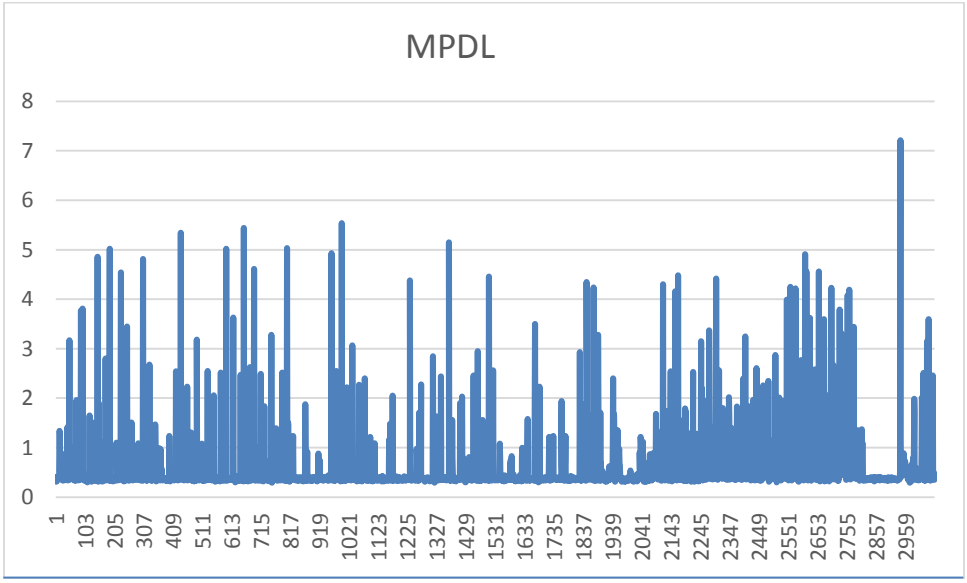

Figure 6.8 Graph of MPD

6.3 Conclusion

Because of this research we are able to obtain the MPD values from the texture surface data received along with the GPS coordinates of the location where the run is taking place concurrently due to multi-threading and in real time. This system is currently built to obtain MPD values from one laser channel and advancements can be made to calculate the Mean Profile Depth for two laser channels i.e. for both left and right wheel path with the use of two NUC processors. The NUC board (s) can be placed in the texture module.

Appendix A

Terminology

a) *Amplitude*: it is the other coordinate of profile of the surface perpendicular to the surface plane.

b) *Distance*: it is one of the coordinates of the profile of a surface along the surface plane. The distance can be measured in a traverse or longitudinal direction in relation to the travel direction on a pavement, or in any direction between them.

c) *Estimated Texture Depth (ETD)*: is obtained from the calculation of mean profile depth using the below mentioned formula.

ETD = 0.2 + (0.8 * MPD), all the dimensions are in mm.

d) *Mean Profile Depth*: The average value of the profile depth over a certain distance (base length).

e) GGA – essential fix data which provides 3D location and accuracy data. (NMEA format)

f) GSA – mainly provides the information regarding the nature of the fix. Also includes the number of satellites being used in the current solution and the DOP (dilute of precision), which is an indication of the effect of satellite geometry on the accuracy of the fix. (NMEA format)

g) GSV – Satellites in view shows the data about the satellites that the unit might be able to find based on it viewing mask and almanac data. One GSV sentence can provide data up to 4 satellites and hence requires three sentences to get the complete information. (NMEA format)

h) *Multithreading*: Multithreading is a widespread programming and execution model that allows multiple threads to exist within the context of a single process.

i) *Outliers:* Invalid readings caused due to dropouts as a result of deep surface troughs or local photometric properties of the surface [1]. The outliers are always removed before computing the MPD using the following equation:

90% outliers = |Reading ± mean| > 1.65*standard deviation.

j) *Profile*: The surface profile is generated if a sensor, like a laser spot or a tip of a needle continuously touches the pavement surface while moving along the surface. Two coordinates describe the surface of the profile, one along the surface plane called distance and the other in a direction perpendicular to the surface plane called amplitude.

k) *Pavement Texture*:. Pavement texture is defined by the irregularities on a pavement surface that deviate from an ideal, perfectly flat surface. ISO 13473-1 [3] differentiates the road texture into 4 different categories based on the texture wavelength.

Micro texture - wavelengths smaller than 0.5 mm.

Macro texture - wavelengths between 0.5 mm and 50 mm.

Mega texture - Wavelengths between 50 mm and 0.5 m.

Unevenness - Wavelengths between 0.5 m and 50 m.

l) *Pavement macrotexture*: is defined as the deviation of a pavement surface from a true planar surface with a dimension along the surface of 0.5mm to 50mm as discussed in chapter 1. Macrotexture determines the rate at which skid-resistance decreases with increasing speed. Results have shown that pavements with more macrotexture indicated a smaller rate of skid resistance with increasing speed.

m) *Pavement microtexture*: is defined as the deviation of a pavement surface from a true planar surface with a dimensions along the surface of less than 0.5 mm. It is believed that at low vehicle speed, the microtexture is mainly responsible factor in determining the skid resistance of a pavement. The contact of a tire and the pavement surface is due to the sharp fine particles in a surface, which make up the microtexture that penetrate the thin water film.

n) *Profile Depth*: it is the difference, within a certain longitudinal/lateral distance in the same order of length as that of tire/pavement interface, between the profile and a horizontal line through the top of the highest particle within this profile.

o) RMC – NMEA has its own version of essential GPS PVT (Position, Velocity and Time) data which is represented in the RMC sentence.

$GPRMC,000050.000,A,3243.9718,N,09706.8053,W,0.29,164.24,220514,,,A*70

p) *Texture Wavelength*: The minimum distance between periodically repeated parts of the curve is defined as the Texture Wavelength and is measured in meters (m) or millimeters (mm).

q) *Texture Depth*: In three dimensional case, it is defined as the distance between the surface and the plane through the top of the three highest particles which are well spaced within a surface area in the same order of a size as that of a tire/pavement surface.

References

[1] ASTM E 1845, 'Standard Practice for Calculating Pavement Macro texture Mean Profile Depth'

[2] Fernando, Emmanuel, Walker, Roger S., FHWA/TX-13/0-6610-1, Impact of Changes in Profile Measurements Technology on QA Testing of Pavement Smoothness, January 2013, Report 6610-1, Texas Department of Transportation

[3] ISO 13473-1, *Characterization of Pavement Texture by Use of Surface Profiles*– Part1 *Determination of Mean Profile Depth*, International Standards Organization (1998).

[4] Martin, James (1965). *Programming Real-time Computer Systems*. Englewood Cliffs, NJ: Prentice-Hall Inc. p. 4. ISBN 013-730507-9.

[5] DataAcq SDK User's Manual - Data Translation

[6] http://www.usglobalsat.com/p-688-bu-353-s4.aspx#images/product/large/688.jpg

[7] http://www.gpsinformation.org/dale/nmea.htm#hardware

[8] http://content.hwigroup.net/images/products/xl/166525/2/intel_next_unit_of_computing _ dc3217by.jpg

[9] http://www.codeproject.com/Articles/3061/Creating-a-Serial-communication-on-Win

[10] http://msdn.microsoft.com/en-us/library/windows/desktop/bb530742 (v=vs.85).aspx

[11] http://www.flite.co.uk/flite-idl-800-digital-lab.htm

[12] https://ftp.dot.state.tx.us/pub/txdot-info/rti/psr/4480.pdf

[13] http://www2.uacj.mx/IIT/lab/Digitales/Equipo/Probador.jpg

[14] https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=sockets%20ppt

BIOGRAPHICAL INFORMATION

Sachin Jayaram was born In Karnataka, India in 1990. He successfully completed his Bachelor of Engineering in Electronics and Communication from Visvesvaraya Technological University, India in 2012. He enrolled for master's program in Electrical Engineering at the University of Texas at Arlington after the completion of his bachelor's. His area of interest are mainly focused on real time embedded systems and wireless communications and is interested in working on automation and real time projects.