

**THE USE OF GIS TECHNOLOGY IN THE DEVELOPMENT OF AN  
INTERACTIVE GEODETIC NETWORK INFORMATION SYSTEM WITH  
APPLICATION IN KENYA"**

**BY**

**WASHINGTON ABUTO  
F56/P/9061/2005~**

**A PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE  
OF MASTER OF SCIENCE IN GEOGRAPHIC INFORMATION SYSTEM AT  
THE UNIVERSITY OF NAIROBI**

University of NAIROBI Library

**0404671 0**

**OCTOBER 2007**

### DECLARATION

This project is my original work and has not been submitted for a degree in any other university.



Signature

Date

Washington Abuto

This project has been submitted for examination with our approval as University Supervisors.



Signature

Q 2 / o y / f l ^

Date

Dr. Ing. Musyoka



Signature

Date

Mr. D. K. Macoco

## ABSTRACT

Interactive Geodetic Information System with application in Kenya is a project that was conducted within Nairobi between May and September 2007 to demonstrate how Geographical Information System (GIS) and other support software can be used to solve problems at Survey of Kenya related to data storage, retrieval, display and updating of the geodetic control network information. It examines data in the card system and introduces a database approach which is linked to a GIS. This system is developed with ArcGIS 9.2 and Access Database. Macromedia Dreamweaver 8 and Fireworks 8 were also used for internet applications and viewing of geodetic network control monuments. Visual Basic2005 was used for developing transformation programmes.

The database query function provides searches by station name, station identifier or by map sheet number. Searches by map sheet number retrieve control stations within a map sheet. It introduces a means of computing transformation of coordinates using double-precision format. A computer programme to interactively transform from geocentric Cartesian coordinates to geodetic coordinates and vice versa was developed. This programme carries a subroutine that can interactively project these coordinates according to UTM.

The study on the database focuses on horizontal and vertical control data. The information used is based on data available from the recent (2003) digital mapping of Nairobi and the existing records within the Geodetic Division of Survey of Kenya. Monuments were photographed with a digital camera at close range from different positions and their images included in the database. Benchmarks were coordinated by a hand-held GPS so as to enable display in a GIS. The results generated by this new system show an improvement over the old system in the areas of data storage, retrieval speed, coordinate transformation and presentation of details. Interactive querying of the geodetic control network stations on the map is now possible.

## TABLE OF CONTENTS

Declaration .....	ii
Abstract .....	iii
Table of Contents .....	iv
List of Figures .....	vii
List of Tables .....	ix
List of Appendices .....	x
List of Acronyms .....	xi
List of Notations .....	xii
Acknowledgements .....	xiv
1.0 Introduction .....	1
1.1 Motivation for the Study .....	6
1.2 Organization Setting .....	7
1.2.1 Background .....	7
1.3 Justification .....	9
1.4 Objectives .....	10
1.4.1 Specific Objectives .....	10
1.5 Problem Statement .....	11
1.6 Scope .....	12
1.7 Organization of the report .....	12
2.0 Literature review .....	13
3.0 Coordinate Transformation and Geodetic Databases .....	15
3.1.1 The Inverse Transformation .....	16
3.1.2 The relation between ellipsoidal height and orthometric height .....	17
3.2.1 The Transverse Mercator projection .....	17
3.2.2 The Inverse Gauss-Kruger projection .....	19
3.3.1 Datum Transformation .....	20
3.3.2 Universal Transverse Mercator (U T M ) system .....	21
3.4 Definitions of some database terms used in this document .....	22

4.0 Methodology.....	23
4.1.1 Identification of user requirements.....	23
4.1.2 Hardware and Software used.....	24
4.1.3 Software was used.....	25
4.1.4 Internet based software application development tools.....	25
4.1.5 Field Work.....	25
4.2.1 Development of the Database.....	26
4.2.2 Method Used in Linking Access Database Records With ArcGIS (9.2)..	28
4.2.3 Internet application development methodology.....	29
5.0 Transformation of coordinates and other support applications.....	31
5.1 Database applications.....	31
5.2 Internet Applications.....	33
5.2 Internet based application for displaying images of benchmark Monuments.....	33
5.3 Internet based application for displaying images of geodetic GPS network control points.....	38
6.0 Tests and Results.....	41
6.1 Benefits.....	41
6.1.1 SQL Statement for station search using station name as the search criteria.....	41
6.1.2 SQL Statement for station search using station identifier number as the search criteria.....	44
6.1.3 SQL Statement for station search using map sheet number as the search criteria.....	46
6.2.1 Graphic Query...,*.....	48
6.3.1 Transformation computation and database queries combined with calculations.....	49

6.3.2 SQL Statement for generating results shown in Table 6.3.3A.....	50
6.3.3 SQL statement for generating results shown in Table 6.3.3B.....	52
7.0 Conclusion and Recommendation.....	54
7.1 Conclusions.....	54
7.2 Recommendations.....	55
References.....	57
Appendices.....	59

**List of Figures**

Figure 1.0A	Sample status information.....	2
Figure 1.0B	A survey pillar.....	3
Figure 1.0C	Image of a steel rod with brass cap attached.....	4
Figure 1.1	GPS control network for Nairobi.....	7
Figure 3.1	A section of ellipsoid.....	15
Figure 3.1.2	Relation between ellipsoidal and orthometric height.....	17
Figure 3.3.1	Datum transformation process.....	20
Figure 4.1.5	Image of station KJ11 a GPS station.....	26
Figure 4.2.1	Entity Relationship Diagram.....	27
Figure 5.1	Sample data entry form.....	32
Figure 5.2A	Internet based display of Benchmarks.....	33
Figure 5.2B	Internet based display of Benchmark LXI/7.....	34
Figure 5.2C	Internet based display of Benchmark LXI/7.....View 2.....	35
Figure 5.2D	Internet based display of Map sheet for Bench Marks.....	36
Figure 5.3A	Internet based display of GPS Control Points.....	38
Figure 5.3B	Internet based display of GPS monument.....	39
Figure 5.3C	Internet based display of GPS monument KJ05.....	39
Figure 6.1.1A	A search query by station name.....	42
Figure 6.1.2A	Search by station Identifier number.....	44
Figure 6.1.3A	Search by Map Sheet number.....	46

Figure 6.2 A	Result of graphic query (for station KJ11).....	48
Figure 6.3A	Sample output of converted UTM coordinates.....	49



**List of Tables**

Table 6.1.1 A Query result of GPS station searched by station name.....	43
Table 6.1.1 B Query result of GPS station searched by station identifier....	45
Table 6.1.3 B Search by Map Sheet number.....	47
Table 6.3.3A Computed geodetic to UTM coordinates compared with existing .....	51
Table 6.3.3 B Computed UTM to geodetic coordinates compared with existing .....	53

**List of Appendices**

Appendix 1	Coordinate transformation from Geodetic to UTM program listing
Appendix 2	Coordinate transformation from UTM to Geodetic Program listing
Appendix 3	Coordinate transformation from Geocentric (Cartesian) to Geodetic program listing
Appendix 4	Coordinate transformation from Geodetic to Geocentric (Cartesian) program listing
Appendix 5	Coverts Degrees Minutes and Seconds to Decimal Degrees program listing
Appendix 6	Converts Decimal Degrees to Degrees Minutes and Seconds program listing
Appendix 7	Database Table Structure
Appendix 8	Sample Data Entry Forms for database
Appendix 9	Sample Computations
Appendix 10	Sample Database Records
Appendix 11	The hard copy digital Map of Nairobi

**List of Acronyms**

DBMS	Database Management System
DEM	Digital Elevation Model
E-R	Entity Relationship Diagram
GIS	Geographic Information Systems
GPS	Global Positioning System
HTML	Hypertext Markup Language
ITRS	International Terrestrial Reference System
JICA	Japan International Cooperation Agency
NCC	Nairobi City Council
NGDI	National Geospatial Data Infrastructure
SOK	Survey of Kenya
WGS 84	World Geodetic System 1984
SQL	Structured Query Language

**List of Notations**

a	Semi-major axis
b	Semi-Minor axis
h	Ellipsoidal height
f	Flattening
phi	Latitude in decimal degrees
phif	Foot point latitude angle in decimal degrees
rphi	Latitude in radians
N	Radius of curvature in the prime vertical eta (as defined in 3.2.1C)
a	alpha (as defined in 3.2.1C)
$\varphi$	phi (Latitude)
$\theta$	theta (as defined in 3.1.2 b)
$\gamma$	gamma (as defined in 3.2.1C)
$\delta$	delta (as defined in 3.2.1C)
$\epsilon$	epsilon (as defined in 3.2.1C)
X	Longitude
i	(X-Xo) Longitude difference
e	eccentricity
ko	Scale factor at the central meridian (0.9996)
ee	eccentricity squared (e )
eep	second eccentricity squared (e' <sup>2</sup> )
aPI	PI
arad	converted value in radians
adeg	converted value in degrees
adegree	individual^value in degrees
aminute	individual value in minutes
asecond	individual value in seconds

Latphi	avariable to hold value for latitude
longlamda	avariable to hold value for longitude
yy	avariable to hold value for computed value by a function computing extra terms to be added to the foot point latitude value
xx	avariable to hold value for computed value by a function computing extra terms to be added to the longitude value,
sinthetataqb	$(\text{Sin}^3 0) \text{sintheta}^3$
costhetataqb	$(\text{Cos } 0) \text{costheta}$
FN	False Northing
FE	False easting
Lamda	Longitude
Rphi	Latitude in Radians
Northings	Grid northing
Eastings	Grid Easting

Additional explanation is included within the programmes enclosed

### Acknowledgments

The result of this work has been achieved due to the co-operation with all the Lecturers at the Department of Geospatial and Space Technology who supported me either directly or indirectly.

I am sincerely grateful to my supervisors Dr. Ing. Musyoka, and Mr.D.K. Macoco for their continued assistance, and advice during the preparation of this project report.

## 1.0 Introduction

A geodetic network can be described as a geometric object in which the various network points are uniquely defined by their coordinates. The coordinates of network points can in selected cases be used to recover the datum position and orientation parameters. A geodetic network is important because it provides a means of controlling infrastructure development of any country as well as a basic framework for the development of any geospatial data infrastructure. A National Geospatial Data Infrastructure (NGDI) supports sharing of data at national level by means of a set of standards such as national reference systems, a national topographic template, a national elevation model, geographical names, administrative boundaries and metadata standards that describe in a consistent manner each of the national infrastructure holdings (Groot et al., 2003).

The broad objective of this project is to create an Interactive Geodetic Control Network Information System applicable to Kenya using Geographic Information System (GIS) Technology. "A geographic information system is a system for managing spatial data and associated attributes. In the strictest sense, it is a computer system capable of integrating, storing, editing, analyzing, and displaying geographically referenced information. In a more generic sense, GIS is a tool that allows users to create interactive queries, analyze the spatial information, and edit data, maps and present the results of all these operations" <sup>^</sup> <http://en.wikipedia.org/wiki/GIS>. In this application, a GIS provides the function of data input, processing, analysis and output of geodetic control data stored in a database. A GIS has two key components; the graphic part and the database part. The two components require linkage. The linkage allows data in the database to be viewed using a GIS software. A database is a collection of non-redundant data sharable between different application systems (Howe, 1982).

This project considers a database composed of horizontal control data and vertical control data that serve as the basis of all GIS placement for both the horizontal and vertical layers. The monuments are physical objects erected on the ground to define the precise location of a control station on the ground. Monuments are briefly discussed in this report for completeness by showing some of their images. From the physical point of view, a control station (or point) is composed of a stable monument typically set on concrete pillar. An example of a pillar is shown by figure 1.0B. Alternatively, a steel rod set on

concrete and driven in the ground with or without brass plate attached can serve the same purpose. As an example, a steel rod with brass cap attached is shown in Figure 1.0C. Both of these monuments define a survey mark. The difference is that with the second case it is not easy to locate the point for reuse when compared with the pillar which is easier to locate. In the same way, a point buried underground may require other survey methods to assist in its location. Locating monuments in this project made use of hand held GPS particularly where the monument was damaged. It is important to know the condition of these monuments to be sure that they are suitable for surveying purposes. One of the objectives of this study is to assess the control point status so as to incorporate this information in the database as "status". Along with the status, the date information was included. The date of photography provides evidence of the actual visit date which also serve as the status date. It is recorded by a digital camera before being transferred to the database. When a monument is photographed, the assessment date should appear in the background to serve as photographic evidence. Figure 1.0A clarify how status information was collected during the fieldwork.

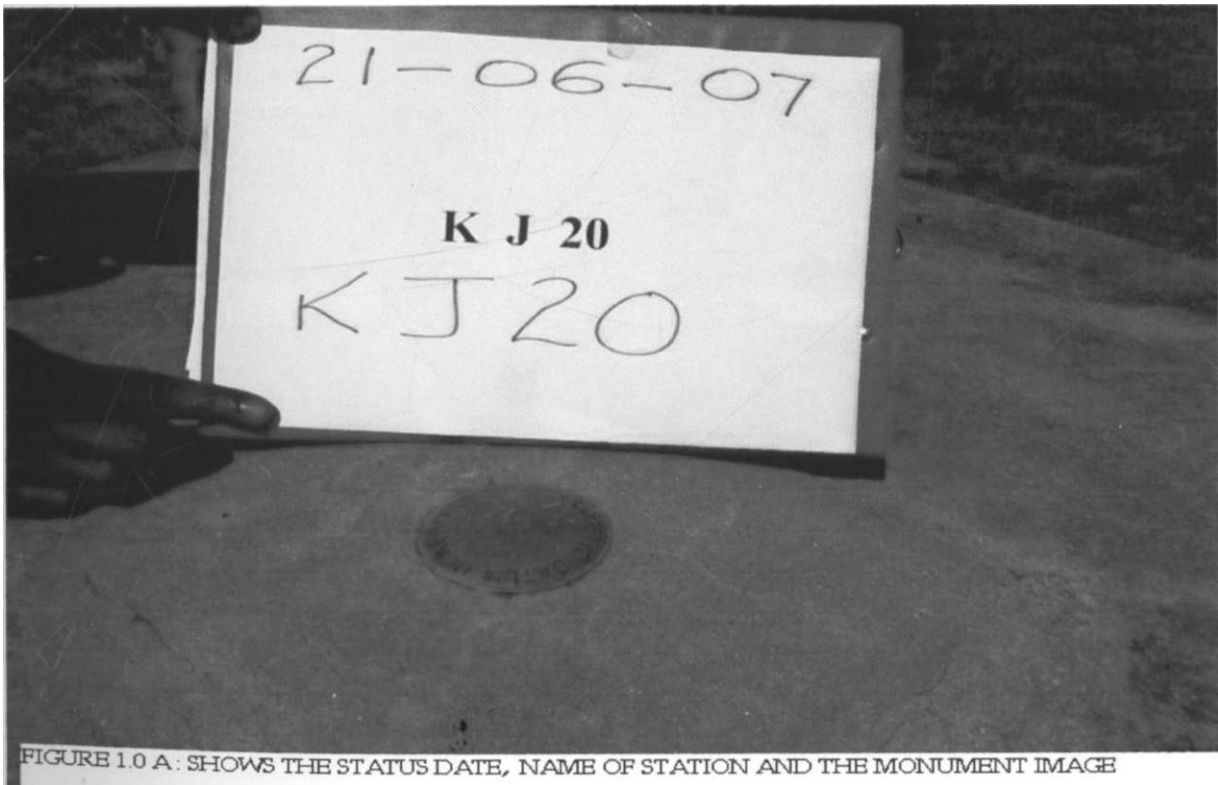


FIGURE 1.0 A: SHOWS THE STATUS DATE, NAME OF STATION AND THE MONUMENT IMAGE



FIGURE 1.0 B Shows a photograph of a Survey Pillar with a Leica GPS system 500 aerial mounted



FIGURE 1.0 B: A Survey Pillar \*

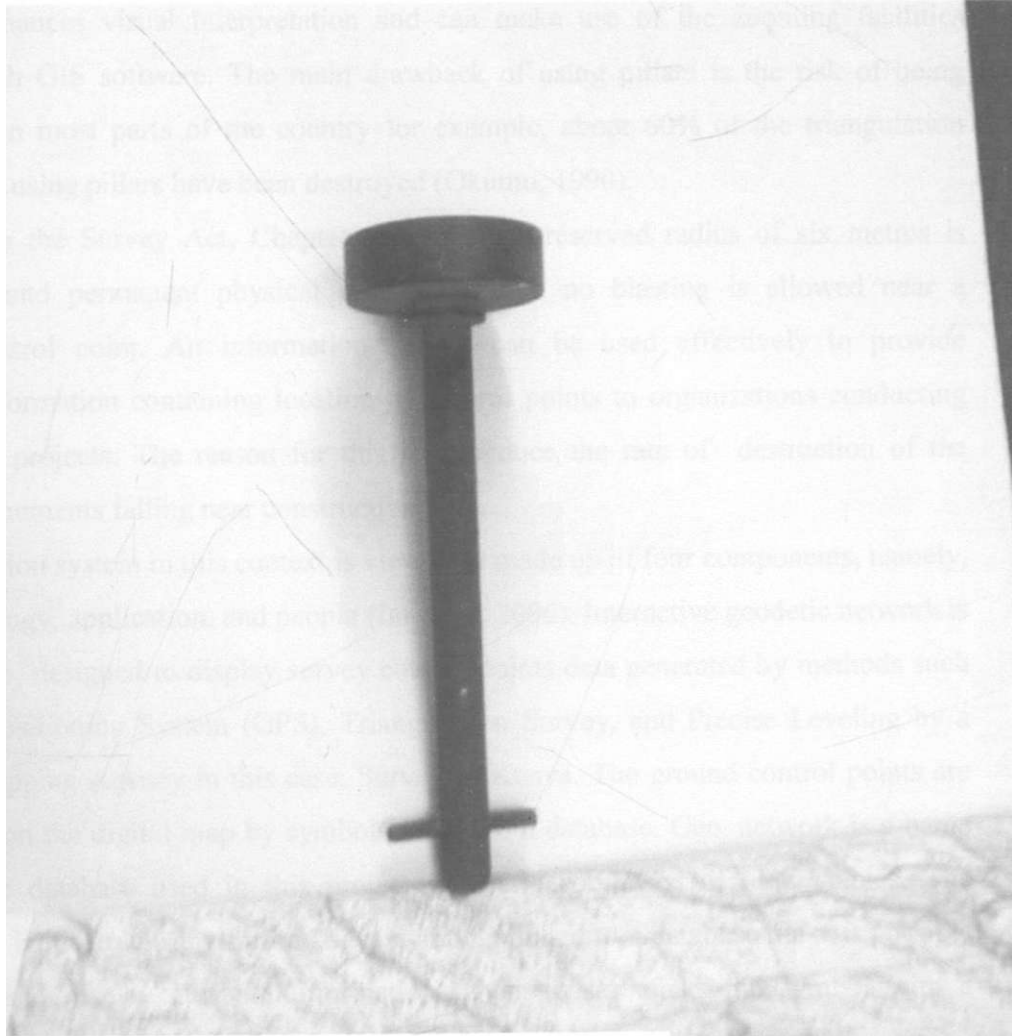


FIGURE 1.0 C: Steel Rod with brass-cap attached

Figure 1.0 C above shows a steel rod with brass-cap attached as used in a Nairobi mapping project described in section 1.1. The brass cap is marked by a cross at the top to define its center. A station name is also included on the same top surface. The cross defines its center precisely. The intention of an information system described in this project is to improve the process of locating control points during reconnaissance or status assessment process. It is therefore designed to include additional data content such as digital photographs of control points and their immediate surroundings. This was achieved by incorporating digital photographic images of the individual points as well as including raster background. The raster background displays as a layer on the computer screen and assists in identifying the location of the station with its surroundings. This

approach enhances visual interpretation and can make use of the zooming facilities available with GIS software. The main drawback of using pillars is the risk of being vandalized. In most parts of the country for example, about 60% of the triangulation stations built using pillars have been destroyed (Okumu, 1990).

According to the Survey Act, Chapter 299(1989), a reserved radius of six metres is required around permanent physical monuments and no blasting is allowed near a geodetic control point. An information system can be used effectively to provide graphical information containing location of control points to organizations conducting construction projects. The reason for this is to reduce the rate of destruction of the geodetic monuments falling near construction sites.

An information system in this context is viewed as made up of four components, namely, data, technology, application, and people (Lo et al., 2006). Interactive geodetic network is a digital map designed to display survey control points data generated by methods such as Global Positioning System (GPS), Triangulation Survey, and Precise Leveling by a National Mapping Agency in this case, Survey of Kenya. The ground control points are represented on the digital map by symbols linked to a database. Geo network is a name given to the database used in this project and it is composed of nine tables. The composition is illustrated by figure 4.2.1. Symbols linked to a database make it possible to retrieve and display attribute information when a user clicks the symbol with a computer mouse for purposes of getting information regarding that station.

Geodetic datum is a means of representing the figure of the Earth, usually as an oblate ellipsoid of revolution, that approximates the surface of the Earth locally or globally, and is the reference for the system of geodetic coordinates. The definition of datum includes the size and shape of the ellipsoid, its location and orientation, and its relation to the geoid by means of geoid undulations and deflection of the vertical (Leick, 2004).

Geodetic coordinates are angular coordinates (longitude and latitude) closely related to spherical polar coordinates and defined relative to a particular earth geodetic datum. Also referred to as geographic coordinates.

### **1.1 Motivation for the Study**

The Strategic Plan, a policy document, for the Ministry of Lands and Settlement (now Ministry of Lands) of 2003/4-2005/6 of Kenya identified some of the weaknesses that impede efficiency and effective delivery of services as being the use of manual records management, analogue equipment, manual handling of survey records and poor storage of data and information.

This project provides a solution to one of the above Ministerial problem by addressing the management of survey records. This process is concerned with the conversion of manual geodetic card system to automated system to improve service delivery of geodetic data.

This project uses Nairobi as a case study area. Nairobi was selected because of convenience and because it has coordinate data obtained recently by use of aerial photography in the year 2003. The study for the spatial data framework for Nairobi City of 2003 is a project that was conducted by the following organizations: Japan International Cooperation Agency (JICA), Survey of Kenya (SOK) and Nairobi City Council (NCC) with a common purpose of improving urban problems related to traffic management, cadastral management and water and sewage management. A project of this nature needed aerial photography. The aerial photography covered an area of 702 square kilometers. The purpose of this study was to transfer technology, construct a GIS model system, and establish spatial data infrastructure at a scale of 1:2500 and 1:5000 using techniques of GPS surveying, ordinary leveling, and aerial triangulation (JICA study team, 2003).

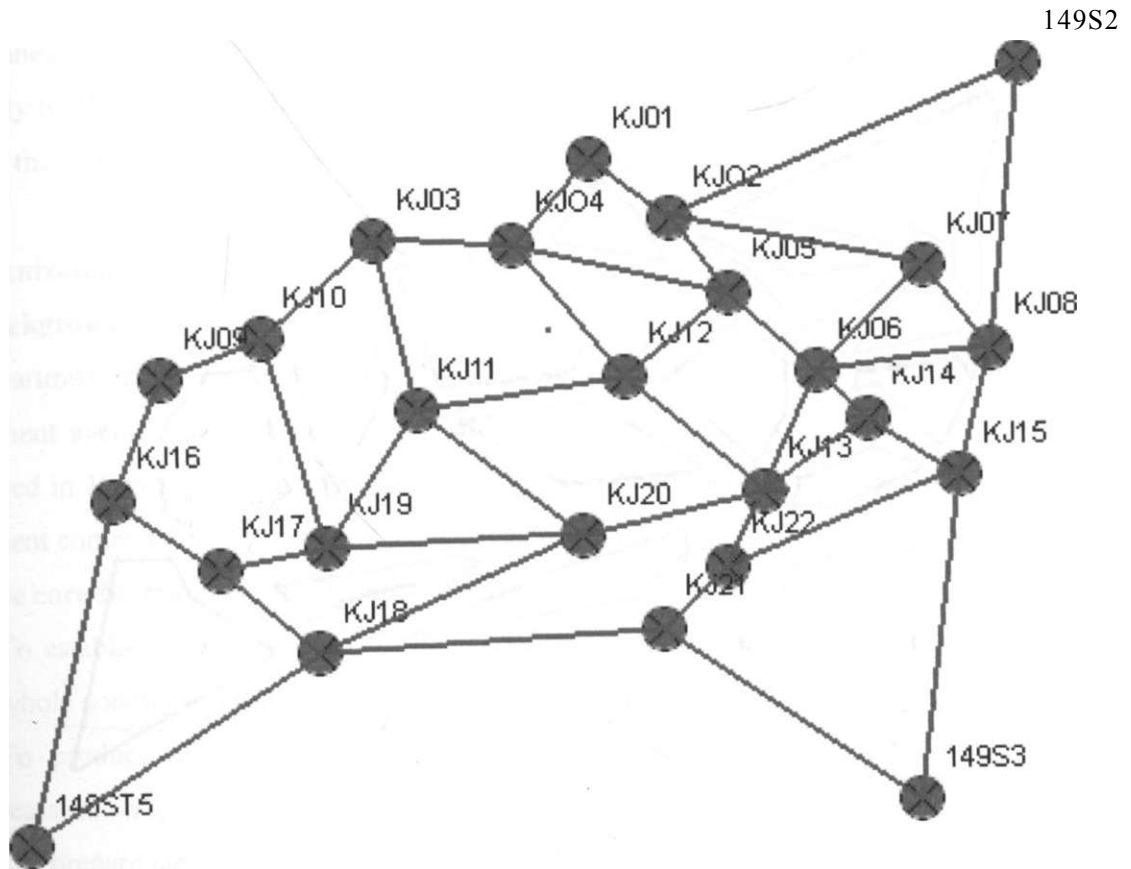


Figure 1.1: Geodetic GPS network for Nairobi

Figure 1.1 above shows the horizontal geodetic GPS control network fixed in 2003 by JICA and Survey of Kenya using Static GPS method (JICA Study Team, 2003).

All the stations in figure 1.1 were visited to assess their status. The points were also photographed and their records included in the database for this project. Three of the above points in the network are part of the existing second order triangulation stations, these are: 149S3, 149S2, and 148ST5. Apart from the GPS control points, a mixture of second and 4th order control points were also included totalling to 375 in number. The 4<sup>th</sup> order control points are used for minor control survey work such as cadastral surveys. All the fourth order control points falling within Nairobi were included in the database but located in a separate table (as shown in figure 4.2.1) and can be displayed as a layer. Levelling height control that were used during the Nairobi mapping were also included. All the three layer data: the GPS stations, trigonometric stations and benchmark stations

have been arranged to display against a raster background. The raster background used are scanned and geo-referenced topographic maps covering the study area. The possibility to display or not to display the raster background is included for flexibility in viewing the control points. This can be achieved when the user selects the layer of interest.

## **1.2 Organization Setting**

### **1.2.1 Background**

The Department of Surveys commonly known as Survey of Kenya (SOK) is the official Government agency for land surveying, hydrographic surveying and mapping. It was established in 1903 as a section of then Public Works Department. In the year 2003, the Department commemorated 100 years of its existence.

### **1.2.2 The core functions of the Department are:-**

- o To establish and maintain a National Geodetic Control Network that covers the whole country to facilitate other surveys and research,
- o To produce and maintain plans of property boundaries in support of land registration to guarantee security of land tenure,
- o To prepare and maintain Registry Index Maps (RIM) and Preliminary Index Diagrams (PID) to support land registration and to ensure security of land tenure,
- o To produce and continuously update national topographic base maps for the whole country at various scales for development planning and for production of other maps.
- o To survey and inspect common boundaries with other countries,
- o To prepare and publish the National Atlas of Kenya.
- o To carry hydrographic surveys for safe navigation, exploration and exploitation of natural resources of rivers, lakes, seas and oceans,
- o To calibrate and maintain survey equipment in order to ensure correct measurements
- o To provide quality control and assurance of geographic data produced by other organizations.
- o To establish and maintain National Spatial Data Infrastructure (NSDI) (Ministry of Lands and Settlement, 2004).

### 1.3 Justification

The present technological advancement is making it more and more possible to adapt to new techniques and procedures. There is increased demand for services that require quick retrieval and display of geodetic information. Prices of software and computers have been declining. From the local newspapers today (2007), one can find a branded computer with hard disk capacity of 100GigaBytes (GB) at approximately US\$ 900 which in the year 2000 could cost over US\$1500. The price differences are due to the abolition of taxation on computers and their accessories. Improved efficiency due to automated searches will enhance geodetic network extension and reduce cost to many users. Data generated by other users can be included in the national network database if data standards are developed and agreed upon among the user community. This action will improve consistency and future planning for more network points. Improved quality assurance of survey data will be possible. More skilled and experienced human resource is now available in the country and their ability to use information system will not pose any problem. Stored digital data is easy to update and can be accessed by many users. Monuments are easy to search in the field by using query results from the system. Leveling or vertical control network can be easily added when the database created is operational. Data visualization can be improved by data processing capabilities of GIS analytic functions and by customization. Reports can be generated more easily. Interactive maps give better flexibility to users. Linking of interdependent data layers using GIS functions can provide more insight to the user. This can be done for example, by including a layer having Ortho-photo maps where available to enhance visual location of points to a user. The system is being mainly developed to capture GPS data which were not thought of at the time the manual system of cards were developed.

Remote sensing analysts stand to benefit from the use of the data generated by an interactive network because they must be sure that their image analysis is accurate with reference to coordinate data. This is achieved by field work process where they can go out to the actual places shown in the images and confirm that what they think they see on the image is actually true. A coordinate transformation software is currently not available in the Department of Surveys, Ministry of Lands so a system for conducting this service will be very valuable to the technicians who require to transform coordinates from

geographicals ( or geodetic coordinates) to UTM map projection system. Sections within the department that would need this application includes: Cadastral surveying, Geodetic section, Cartographic drawing offices and Photogrammetric unit. In digital cartography it will support the process of computing sheet corner coordinates and for use in geo-referencing. Transformation of coordinates software also serve as a system for computing or checking coordinate data that goes to the database developed in this project.

#### **1.4.0 Objectives**

The broad objective is to create an interactive geodetic network information system using GIS technology that can replace the present manual card system.

##### **1.4.1 Specific Objectives:**

The following are the specific objectives designed for solving the problems:

1. To conduct a study on current data handling procedures at Survey of Kenya Geodetic section. This will involve identifying current methods of storage of geodetic data such as cards, registers, and computation files, maps and plans and their retrieval in the manual system.
2. To assess the present status of survey monuments used in the Nairobi Mapping Project conducted in year 2003. The assessment is made by making field visit to each station recording its status and taking photographic images of each control point for inclusion in the database as part of the station information.
3. To design a database for both horizontal and vertical controls and to key in data related to GPS, Trigonometric stations, and Benchmarks including support documents such as plans and computation files within the project area.
4. To scan, rectify and geo-reference the topographic maps covering the study area and to link the imagery with the database records.
5. To demonstrate internet based application for viewing GPS stations, and Benchmarks used in the Nairobi mapping.
6. To develop application program for transforming coordinates in Cartesian to Geodetic and from Geodetic to UTM coordinates system and reverse computation of the same.



7. To write some Structured Query Language (SQL) statements for querying the database created.

### **1.5 Problem Statement**

The problem is that the National Geodetic Network data kept at Survey of Kenya (SOK) is still in manual card system and therefore impedes efficiency in service delivery to the public. In addition, the following problems have been identified:-

- There is no proper means of recording Global Positioning System (GPS) generated coordinates and associated data.
- Existing maps depicting geodetic control points cannot be queried interactively.
- Information supply with respect to completeness, detail, timeliness and accessibility is limited as only one user at a time can access the data. The space in the card can only contain limited amount of information.
- There is no means of querying the status of survey control points (monuments for vertical and horizontal control) as to their suitability for occupation and use.
- Retrieval of coordinate data that fall within a map sheet of interest is slow.
- Database searches by station name or by station identifier has not been developed at Survey of Kenya making retrieval of this information slow.
- A GIS based system that can record and display control points have not been developed at (SOK).
- Paper material requires large storage space and has limited lifespan when handled regularly. Sharing of map data in paper form poses deterioration problems.
- Transformation and conversion software for coordinates such as the conversion of UTM to geographical and vice versa are currently not available at SOK. Earlier attempts were made at SOK in the year 1983 where UTM coordinate transformation software was recorded and run on an 8 inch diskette. This system was based on Disk Operating Systems (DOS) which cannot be used with current windows operating system unless updated due to technological development in recent times. New software\* is therefore required to be developed for handling transformation based on current operating systems.
- Internet based display of geodetic network monuments have not been developed.

## **1.6 Scope**

The scope of the data content keyed into the database will be limited to Nairobi being the study area. The status information has been field collected for the GPS stations and the benchmarks. The status information for classical trigonometric stations was not intended to be collected due to the time limit specified for this project. The structure for its database has however been designed together with associated data entry form. The status information therefore can be included in future by updating existing records to add photo images of the trigonometric control points when they become available.

Some aspects of coordinate transformation such as from Geodetic to Cartesian, Geodetic to Universal Transverse Mercator and their inverse computation is part of this project.

Internet based display of geodetic monument status targeting new control points have been included. The Helmert transformation is briefly discussed in 3.3.1 but is not included in the program development. This is because transformation parameters have not been agreed upon nationally.

## **1.7 Organization of the report**

The report is organized into chapters; Chapter 1 introduces the terms used and provides a summary of topics covered. Chapter 2 reviews the concept of interactive maps with particular reference to geodetic data handling and uses.

Chapter 3 reviews the concept of coordinate transformation and the geodetic database. Chapter 4 is concerned with the methodology. Chapter 5 discusses the study area and the three components of the study and how GIS together with the Access database were created and linked with the graphic software (ArcGIS 9.2). The second part is concerned with the internet application for displaying image data of the GPS stations and the Benchmarks. The third part discusses the application programs developed for the transformation. Chapter 6 discusses tests and results and presents some of the system output results. Chapter 7 presents the conclusion, discussions and recommendations.

## 2.0 Literature review

Various countries are in the process of improving their geodetic network information systems, for example Puerto Rico's new geodetic network is a high accuracy geodetic reference network developed to support GIS-based cadastral system (<http://www.govtech.com/gt/95306>).

In the United States, the National Geodetic Survey (NGS) department of commerce is responsible for the national geodetic network known as the National Spatial Reference System (NSRS). NSRS database is reported to contain more than 850,000 geodetic monuments that make up the NSRS network (<http://events2.esri.com/uc/2006/sag/list/index.cfm?fa=detail&SID=427>).

The information about each monument is organized in a "data sheet" and is available to users by the web from the NGS database. This approach differs from the principle of an interactive map but shares having data in the database. This project will also explore the web functionality.

At Survey of Kenya, the geodetic data consisting of horizontal coordinates is recorded on cards while Benchmark data is recorded on tracing material not drawn to scale. Retrieval is done manually and geodetic control networks are plotted on topographic map sheet of 1:50,000 Scale while for the whole country is plotted on a map sheet of 1:1,000,000 scale. Very close points are not included at this scale but are recorded on the card. The updating of topographic maps is done only when new editions are printed due to high printing costs. The last such update was done in 1985. Several new points have been observed in recent years but have not been reflected in the maps. This category of map is also static that is a map that cannot be queried directly. The data shown on the map has limited details which are not linked to the cards. The cards are sometimes misplaced causing delays in retrieving information. Users need to know the status of control points in advance to facilitate proper planning of any new survey project. Users also need a system that is able to search the records quickly using the station name or finding the control points in a particular map sheet. The number of control network continues to reduce mainly because of lack of information by the members of the public and due to construction and pure vandalism making survey work difficult, slow and expensive. This

situation can however be improved by including information on status of the pillar in the database and collecting information on control points to enable effective statistical analysis , decision making and re-establishment of destroyed monuments.

### 3.0 Coordinate Transformation and Geodetic Databases

3.1 A brief review of coordinate transformation is included to explain the basis for the coordinate transformation programs used later in this project.

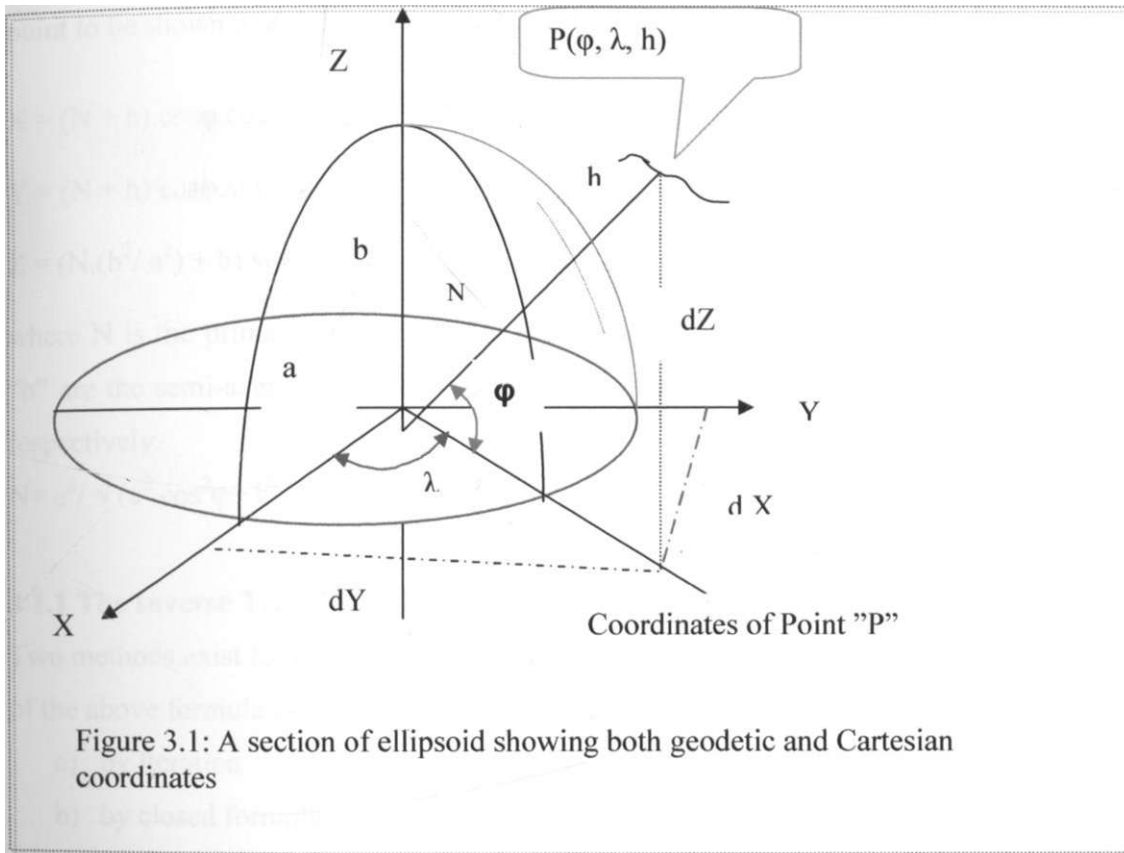


Figure 3.1 above shows a section of an ellipsoid which is a mathematical figure formed by revolving an ellipse about its minor axis. The ellipsoid also known as spheroid, is defined by two quantities, the length of the semi-major axis "a" and the flattening "f". It serves as a geometric structure for mathematical formulations and computations.

Global Positioning System (GPS) coordinate system is based on WGS84 as the geocentric reference frame. The World Geodetic System defines a reference frame for the earth, for use in geodesy and navigation. This reference frame is globally consistent within  $\pm 1$  metre. Current geodetic realizations of the geocentric reference system family, <sup>ITRS</sup> (International Terrestrial Reference System) maintained by the IERS are geocentric, and internally consistent, at the few-cm level, while still being metre-level

consistent with WGS84 (<http://en.wikipedia.org/wiki/WGS84>). In many cases. Surveyors would need to convert coordinates generated by GPS to a local coordinate system. It is therefore necessary to have a means of converting to the local system. The relation between Cartesian and Geodetic assumes that the Cartesian coordinates of a point in space X, Y, Z has the same origin with an ellipsoid of revolution. This enables the same point to be shown in ellipsoidal coordinates e.g. (Hofmann-Wellenhof et al., 1992).

$$X = (N + h) \cdot \cos(\varphi) \cdot \cos(\lambda)$$

$$Y = (N + h) \cdot \cos(\varphi) \cdot \sin(\lambda) \tag{3.1a}$$

$$Z = (N \cdot (b^2/a^2) + h) \cdot \sin(\varphi)$$

where N is the prime vertical radius of curvature at latitude  $\varphi$ . The parameters "a" and "b" are the semi-axes of the ellipsoid and  $\varphi$ ,  $\lambda$ , h are the latitude, longitude, and height respectively.

$$N = a^2 / \sqrt{a^2 \cdot \cos^2(\varphi) + b^2 \cdot \sin^2(\varphi)} \tag{3.1b}$$

### 3.1.1 The Inverse Transformation

Two methods exist for computing the ellipsoidal coordinates ( $\varphi$ ,  $\lambda$ , h which is the inverse of the above formula as follows:

- a) by Iteration
- b) by closed formula

It has been demonstrated that there is no significant difference between the two methods this has been stated for example by Hofmann-Wellenhof et al, (1992). The closed formula for transforming X, Y, and Z into  $\varphi$ ,  $\lambda$ , and h is given below:

$$\varphi = \arctan \left( \frac{Z + b \cdot e^2 \cdot \sin^3 \varphi}{(N + h) - a \cdot e^2 \cdot \cos^3 \varphi} \right) \tag{3.1.1a}$$

$$\lambda = \arctan (Y/X) \text{ where } X \geq 0$$

$$= (180 + \arctan (Y/X)) \text{ where } X < 0 \text{ and } y \geq 0$$

$$\lambda = (-180 + \arctan (Y/X)) \text{ where } X < 0 \text{ and } y < 0$$

$$h = \sqrt{X^2 + Y^2} - N$$

$$e^2 = (a^2 - b^2) / a^2 = 2 \cdot f - f^2$$

where  $\theta = \arctan (Z \cdot a / p \cdot b)$

$$= (a^2 - b^2)/b^2, \text{ the second numerical eccentricity} \quad (3.1,2b)$$

WGS84 parameters are as follows (Kaplan, 1996).

The equatorial cross-section of the earth has a radius of 6,378.137 km, the semi minor axis,  $b$ , in WGS84 is taken to be 6,356.7523142 km and the flattening can be derived from  $f = 1 - b/a$ ,  $e = a \cdot e / b$  also  $e = \sqrt{1 - (b^2/a^2)}$

### 3.1.2 The relation between ellipsoidal height and orthometric height.

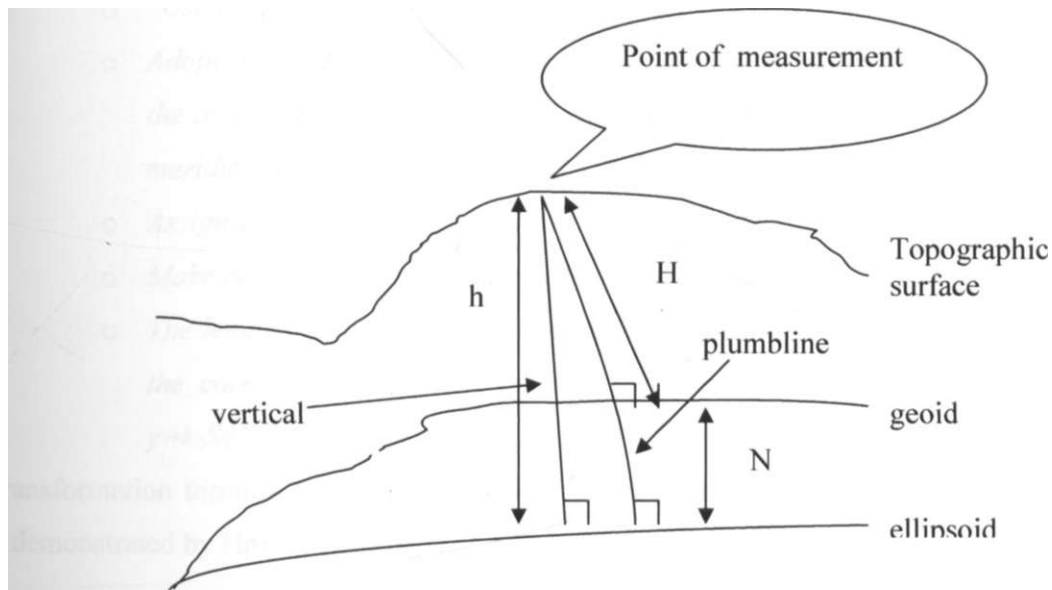


Figure 3.1.2: Relation between ellipsoidal height and orthometric height.

As illustrated in figure 3.1.2 above,  $h = H + N$  where  $h$  is the ellipsoidal height,  $H$  is the elevation also called orthometric height,  $N$  is the geoidal height (also called undulation).

### 3.2.1 The Transverse Mercator projection

Map projection use a mathematical model to convert three dimensional reality such as positions on the earth's curved surface into a two dimensional surface represented by a map. Map Projection is achieved by different methods. One such method which is also suitable for Kenya (due to its limited longitudinal extent) is the Transverse Mercator

Projection. Transverse Mercator is a cylindrical and conformal projection. Transverse implies that the cylinder is rotated 90 degrees relative to the equator ([http://en.wikipedia.org/wiki/Transverse\\_Mercator\\_projection](http://en.wikipedia.org/wiki/Transverse_Mercator_projection)). This makes the projected surface aligned with the central meridian rather than the equator as in the case with Mercator projection. It is an ideal projection for mapping within the equatorial region. Conformal projection preserves angles after it has been mapped on a plane. This means that angular measurements made on the projection surface are true. The projection is based on a cylinder designed to be smaller than the spheroid. The flattened cylinder has key projection demands as follows:

- o *"Use conformal mapping conditions.*
- o *Adopt a central meridian  $X_0$  that passes more or less through the middle of the area to be mapped. For convenience, the longitude defining the central meridian is re-labeled to start at  $l = 0$ .*
- o *Assign  $x = 0$  to the image of the central meridian.*
- o *Make center of the mapped region to coincide with the  $y$  axis of the map.*
- o *The length of the mapped central meridian should be  $k_0$  times the length of the corresponding elliptic arc  $S(p)$  for example, at the central meridian  $y = k_0 S(p)$ " (Leick, 2004).*

Transformation formula for converting ellipsoidal coordinates to local grid for example is demonstrated by Hofmann-Wellenhof et al.(1992) by the following formula:

$$y = B(\phi) + (t/2) \cdot N \cdot X^2 \cdot \cos^2 \phi + (t/24) \cdot N \cdot \cos^4 \phi \cdot (5 - t^2 + 9\phi^2 + 4\pi^4) \cdot I^4 + (t/720) \cdot N \cdot (\cos^6 \phi) \cdot (61 - 58 \cdot t^2 + t^4 + 270t^2 - 330 \cdot t^2 \cdot r^2) \cdot \epsilon^6 + (t/40320) \cdot N \cdot (\cos^8 \phi) \cdot (1385 - 3111 \cdot t^2 + 543 \cdot t^4 - t^6) \cdot C^8 + \dots \quad (3.2.1a)$$

$$x = e \cdot N \cdot \cos \phi + (1/6) \cdot N \cdot \cos^3 \phi \cdot (1 - t^2 + r^2) \cdot I^3 + (1/120) \cdot N \cdot \cos^5 \phi \cdot (5 - 18 \cdot t^2 + t^4 + 14 \cdot r^2 - 58 \cdot t^2 \cdot r^2) \cdot \epsilon^5 + (1/5040) \cdot N \cdot \cos^7 \phi \cdot (61 - 479 \cdot t^2 + 179 \cdot t^4 - t^6) \cdot \epsilon^7 + \dots \quad (3.2.1b)$$

where,

$t = \lambda - \lambda_0$  longitude difference

$N = a / (b \cdot \sqrt{1 + r^2})$  Radius of curvature in prime vertical

$H = t \cdot (\cos \phi)$  is auxiliary quantity



$e = (a - b)/b$  is the second numeric eccentricity

$t = \tan(p)$  is an auxiliary quantity

$X_0$  is longitude of the central meridian

the pair  $(y, x)$  corresponds to  $(\phi, X)$

$B(\phi)$  is the arc length of the meridian and is also the ellipsoidal distance from the equator to the mapping point.

$$B(\phi) = a \cdot [\phi + B \cdot \sin 2\phi + y \cdot \sin 4\phi + 5 \cdot \sin 6\phi + e \cdot \sin 8\phi + \dots] \quad (3.2.1 c)$$

$$a = ((a+b)/2) \cdot (1 + (1/4) \cdot n^2) + (1/64) \cdot n^4 + \dots$$

$$B = - (3/2) \cdot n + (9/16) \cdot n^3 - (3/32) \cdot n^5 + \dots$$

$$y = (15/16) \cdot n^2 - (15/32) \cdot n^4 + \dots$$

$$5 = (35/48) \cdot n^3 + (105/256) \cdot n^5 + \dots$$

$$e = (315/512) \cdot n^4 + \dots$$

$$n = (a-b) / (a+b)$$

**3.2.2 The Inverse Gauss-Kruger projection** e.g. Hofmann-Wellenhof et al. (1992) is given below.

The formula for mapping ellipsoidal point  $Y, X$  in the plane into a point  $(\phi, \lambda)$  on the ellipsoid is shown below by a series expansions formula.

$$\begin{aligned} \phi = & \phi_f + (t_f / (2 \cdot N_f^2)) \cdot (-1 - r_f^2) \cdot x^2 + (t_f / (24 \cdot N_f^4)) \cdot (5 + 3 \cdot t_f^2 + 6 \cdot x^2) \cdot r_f^2 - 6 \cdot t_f^2 \cdot r_f^2 - 3V - \\ & 9 \cdot t_f^2 \cdot r_f^4) \cdot x^4 + (t_f / 720 \cdot N_f^6) \cdot (-61 - 90 \cdot t_f^2 - 45 \cdot t_f^4 - 107 \cdot r_f^2 + 162 \cdot t_f^2 \cdot r_f^2 + 45 \cdot t_f^4 \cdot r_f^2) \cdot x^6 \\ & + (t_f / (40320 \cdot N_f^8)) \cdot (1385 + 3633 \cdot t_f^2 + 4095 \cdot t_f^4 + 1575 \cdot t_f^6) \cdot x^8 + \dots \end{aligned} \quad (3.2.2a)$$

$$\begin{aligned} \lambda = & X_0 + (x / (N_f \cdot \cos(\phi_f))) + (1 / (6 \cdot N_f^3 \cdot \cos(\phi_f))) \cdot (-1 - 2 \cdot t_f^2 - r_f^2) \cdot x^3 + (1 / (120 \cdot N_f^5 \cdot \cos(\phi_f))) \cdot (5 + 28 \cdot t_f^2 \\ & + 24 \cdot t_f^4 + 6 \cdot t_f^2 \cdot r_f^2 + 8 \cdot t_f^2 \cdot r_f^4) \cdot x^5 + (1 / (5040 \cdot N_f^7 \cdot \cos(\phi_f))) \cdot (-61 - 662 \cdot t_f^2 - 1320 \cdot t_f^4 - 720 \cdot t_f^6) \cdot x^7 + \dots \end{aligned} \quad (3.2.2b)$$

where the terms with the subscript  $f$  must be calculated based on the foot point latitude  $\phi_f$

The series expansion for the foot point latitude is given by

$$\phi_f = y + B' \cdot \sin 2y + y' \cdot \sin 4y + 5' \cdot \sin 6y + c' \cdot \sin 8y + \dots \quad (3.2.2c)$$

where

$$a' = (1/2) \cdot (a + b) \cdot (1 + n^2)/4 + n^4/64$$

$$B' = (3/2) \cdot n - (27/32) \cdot n^3 + (269/512) \cdot n^5 + \dots$$

$$y' = (21/16) \cdot n^2 - (55/32) \cdot n^4 + \dots$$

$$5' = (151/96).n^3 - (417/128).n^5 + ..$$

$$e' = (1\ 097/512).n^4 +$$

and the relationship,

$$y = y/a'$$

### 3.3.1 Datum Transformation

The datum transformation involves transformation of coordinates from one system into another system. The process may be carried out as separate transformation of three transformations:

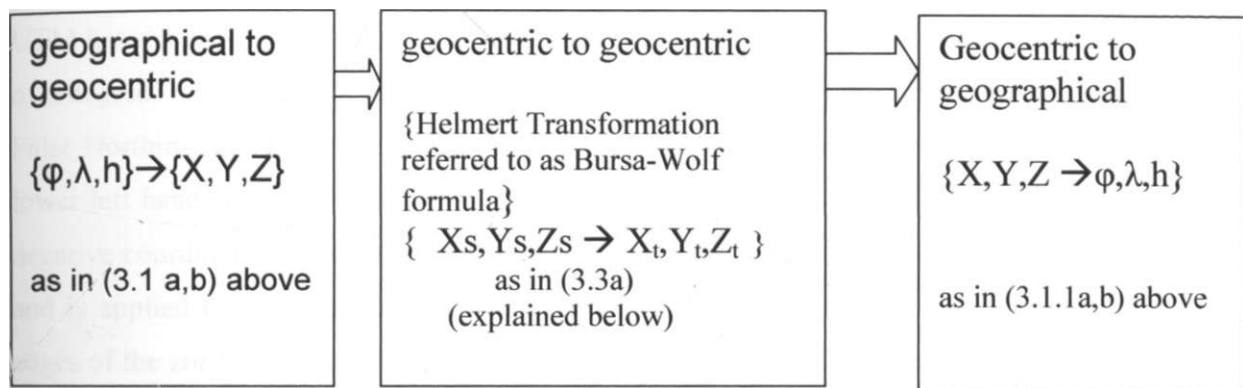


FIG.3.3.1: Datum transformation process

The geocentric to geocentric transformation is referred to as Helmert transformation expressed in matrix form with seven parameters where  $\{X_s, Y_s, Z_s\}$  describe the source geocentric coordinate system and  $\{X_t, Y_t, Z_t\}$  describe the target geocentric coordinates system

$$\begin{matrix} |X_t| \\ |Y_t| \\ |Z_t| \end{matrix} = \begin{matrix} |1 & -R_z & +R_y| \\ | +R_z & 1 & -R_x| \\ | -R_y & +R_x & 1| \end{matrix} * \begin{matrix} |X_s| \\ |Y_s| \\ |Z_s| \end{matrix} + \begin{matrix} |dX| \\ |dY| \\ |dZ| \end{matrix} \quad (3.3 a)$$

$dX, dY, dZ$  is the translation vector to be added to the point position vector in the source coordinate system in order to transform from source to target system

$R_x, R_Y, R_Z$  are small rotation angles expressed in radians to be applied to the position vector  $M$  is a scale correction to be applied to the position vector of the source coordinates  $S_n$  as to obtain the correct scale in the target coordinates system.

$$M = (1 + dS \cdot 10^{-6}) \tag{3.3b}$$

where  $dS$  is the scale correction in parts per million

### 3.3.2 Universal Transverse Mercator (UTM) system

The Universal Transverse Mercator projection is a form of the Transverse Mercator projection. It covers the world between latitude  $80^\circ$  S and  $84^\circ$  N. The main difference between Transverse Mercator and Universal Transverse Mercator discussed above, is that the UTM has many of the defining parameters held fixed. The UTM is split into sixty zones of 6 degrees longitude with adjacent zones overlapping by 30 minutes. False Easting and False Northing are applied to enable the origin of the grid projection to be viewed in the lower left hand corner by convention. The result is that it does away with the need for negative coordinates. In addition to the above, UTM has a central scale factor of 0.9996 and is applied to the conformal coordinates to avoid large distortions within the outer edges of the zone. The scale varies in an east-west direction. The Scale in the north-south direction does not vary. The ellipsoid used in Kenya for conformal mapping is defined in the Survey Act Cap 299 and is the (modified) Clarke 1880 having **elements; semi-major** axis  $a=6,378,249$  meters, and ellipticity  $f = 1/293.465$ . The applicable Zones for use within Kenya for UTM are 36 and 37. Nairobi falls within zone 37. **Zone** number can be calculated by the formula (Hofmann-Wellenhof et al., 1992).

$$Z_n = \text{INT}((180 \pm X)/6) + 1 \tag{3.3.2a}$$

In the above formula the + sign is applied to eastern longitudes and - sign to the western longitudes.

The formula below can be used, to compute the central meridian

$$C = 6 \cdot Z_n - 183 \text{ where } \tag{3.3.2b}$$

$C$  = Central meridian and  $Z_n$  = Zone number and the function  $\text{INT}()$  is used to convert values inside the bracket to integer. »

### 3.4 Definitions of some database terms used in this document

The following terms are used in this document and require clarification :

- o An Entity is a thing such as an object which the enterprise recognizes as capable of an independent existence, and which can be uniquely identified. For example a map, computation files, Field notes, Cards, Machines etc.
- o An Attribute is a property of an entity, for example an attribute of a computation file can be computation identifier, computation type, date completed, agency involved and the name of the Surveyor responsible,
- o A Relationship is an association between two or more entities. For example, an entity Department may be associated with an entity employee via a relation employs.
- o By definition, an entity occurrence must be uniquely identifiable. An attribute whose value identifies an entity occurrence is called an entity identifier,
- o Entity relationship diagrams show individual entity occurrences and their relationships.

A database management system based on relational model is a relational database management system. A database management system provide controlled access to the data. The term database is sometimes used to mean data itself (Ramon, et al., 2005).

## **4.0 Methodology**

### **4.1.1 Identification of user requirements**

#### **Background**

The system for storing and accessing geodetic information at Survey of Kenya, Ministry of Lands is based on the manual card system. For a user to obtain the coordinate of any survey station, a station is first identified from the topographic map or the trigonometric chart of the area. From here the name of the station is obtained and used to locate the card. A user who knows the station name can skip this step, and use the station name directly for the search. The card system is indexed and cards are stored in a removable drawer within the Geodetic (Technical) section. An external user who requires coordinate data is given a photocopied version of the contents of the card. The card retrieval is managed by an officer assigned the work of retrieving map and coordinate data to the members of the public. This information is not free and is currently charged per station at a fee of (US\$ 3.5). The use of card system for recording geodetic data is believed to have started by early 1950s. Most of survey activities that are well documented were carried out around this period. The information in the card include horizontal coordinates with additional information about topographic sheet, the order and sometimes geodetic coordinates ( latitude and longitude) as well as height data in feet. If the station has height in metres it is indicated. Benchmark heights are provided separately in the form of linen map sheets. Benchmarks have no horizontal coordinate information but they have height data in feet units. A sheet contains a number of benchmark stations. Each station has its own individual compartment where a location sketch is drawn and height information shown. This is comparable to the card system used in the horizontal control system but with different information. Turning back to the horizontal control, the status of damaged trigonometric stations is indicated on the card as "pillar destroyed" on the limited space available. This activity is not usually up-to-date and in most cases not done. Both the horizontal and vertical information in either the card or the sheet have reference to the computation file where they originated.

The challenge now is to bring all these pieces of information together in the development of the database to satisfy Survey of Kenya's need.

The procedure adopted during the development of the database followed the following steps:

- To study data content requirement and use this information for further work like the creation of tables.
- To conduct database design for horizontal and vertical data including support documents such as computation files, maps and registers.
- To photocopy all the cards relevant to the study area and use them for entering data in the database.
- To convert height information given in feet to metres before keying them in the database.
- To scan, rectify and geo-reference existing topographic maps covering the study area.
- To experiment with orthophoto imagery of Nairobi mapping project.
- To create application software for data entry forms and use the forms to enter data to each table created. This was iterative and adjustment was done where necessary. Screen print of these forms are shown in Appendix 8
- To develop Internet based application as a means of viewing data from remote station.
- To develop transformation software for transforming geodetic coordinates to UTM coordinate system and the inverse computation ( UTM to Geodetic).
- To develop transformation from Cartesian (geocentric) system (such as WGS84 ) to geodetic system along with their inverse computation. The printout of the six source programs are included in the Appendices 1 to 6
- To test new programs for correct response. Testing was done by data for Nairobi Digital Mapping project and other sources.

#### **4-1.2 Hardware and Software used :**

The following hardware were used for the project:

- PC Pentium IV Toshiba Centrino Duo with 80GB hard disk of 1 GB RAM,
- SGH-X820 Samsung Mobile'Phone with built in camera of 2mega pixels,
- Hand-held Garmin GPS 12XL 12 channel.
- Plotter HP Designjet 1055cm model No. C6075B

**4.1.3 The following Software Packages were used :**

- ARC-GIS 9.2 an ESRI product,
- Access 2002 - Database Management,
- Visual Basic 2005 Express Edition,
- Samsung studio software as supplied with Mobile phone above for downloading image photographs to the computer.
- Windows XP with service pack 2 operating system

**4.1.4 Internet based software application development tools**

- Macromedia Dreamweaver 8,
- and Macromedia FireWorks 8

**4.1.5 Field Work**

Fieldwork was conducted by visiting each of the GPS control point and getting its status. In a similar way, the same was done for the Benchmark stations.

The monuments were photographed with a digital camera from different locations. The digital images were later processed and included in the database so that they can be viewed when queried.



Figure 4.1.5 : GPS point (KJ11)

#### 4.2.1 Development of the Database

A total of nine tables were created and named as follows: GPSStations, Geoid ht, Trig Station, H Mapsheet, Hcomputation, Surveyor, V Computation, V Mapsheet and Benchmark. The content of their data is shown in the next diagram. A more detailed indication of table structure is given in Appendix 7 titled sample database structure. There were difficulties in identifying the records such as computation files and the process took longer than planned. After defining the structure, data entry forms were created and used. Sample entry forms are shown in Appendix 8. Data was inspected and corrected for any entry errors. Figure 4.2.1 shows how the tables were related.



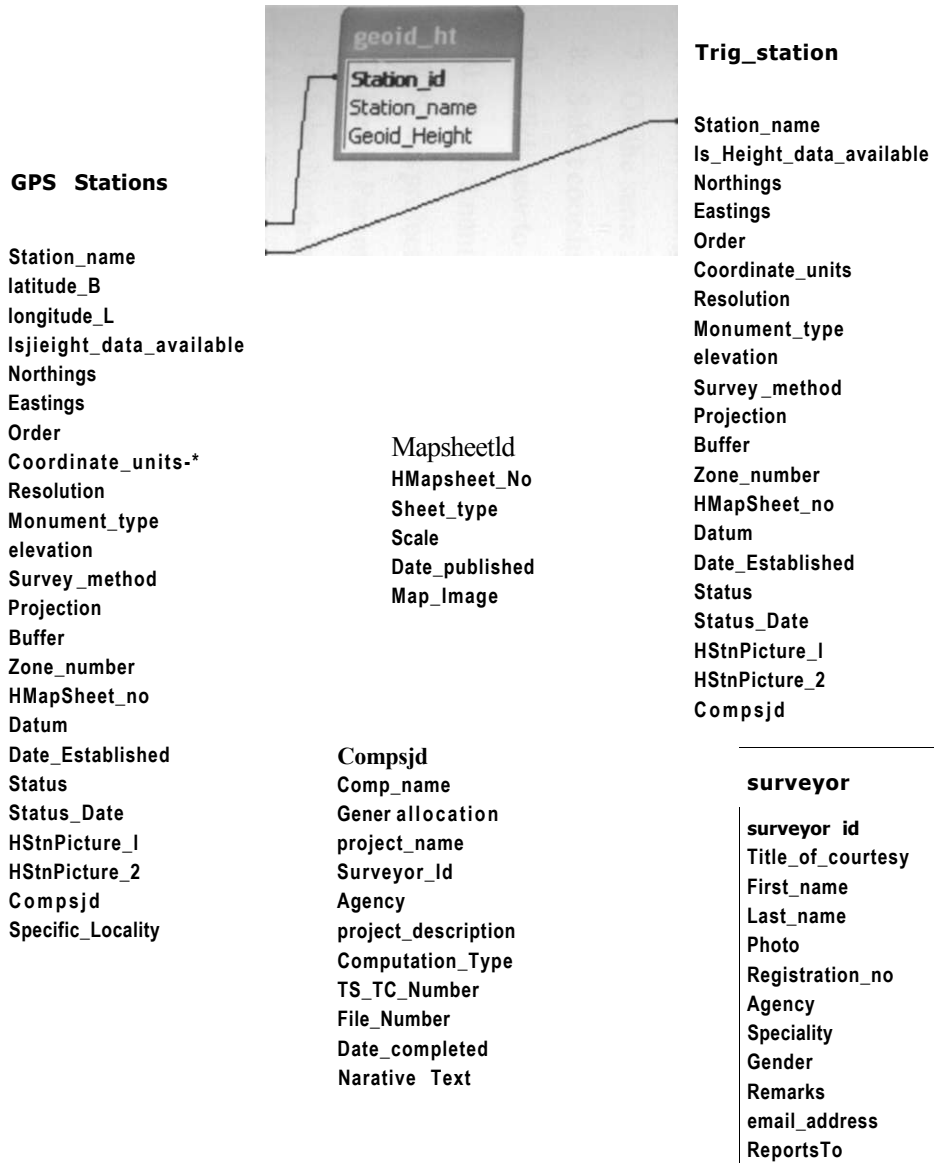
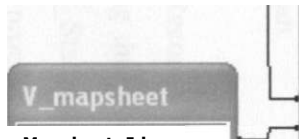


FIGURE 4.2.1: ENTITY RELATIONSHIP DIAGRAM

**V\_Computation**

Gompsjd  
Comps\_name  
General\_Location  
project\_name  
Surveyor\_Id  
Agency  
project\_description  
Computation\_Type  
GL\_TS\_Number  
File\_Number  
Date\_completed  
Narrative Text

n



Mapsheets\_Id  
Mapsheets\_Name  
Sheet\_type  
Scale  
Date\_published  
Mapjimage

**Benchmark**

Benchmarkjd  
Benchmark\_name  
Geodetic\_coord\_available  
Northings  
Eastings  
Order  
Coordinate\_units  
Resolution  
Monument\_type  
elevation  
Leveling\_method  
Projection  
Buffer  
Zone\_number  
VMapSheet\_no  
Datum  
Date\_Established  
Status  
Status\_Date  
VStnPicture\_1  
VStnPicture\_2  
Mapimage  
Comps\_Id  
Mapsheetsjd  
SHAPE

**Tables were** separated to avoid having repeating groups but were related to one another through the links shown by figure 4.2.1. Relating tables allow data to be queried from different tables. The tables were related by defining common fields with the same data type. The data type determine what type of data a field can hold. Examples of data types are: Date, String, Double, Single, Integer and Boolean.

#### **4.2.2 Method Used In Linking Access Database Records With ArcGIS(9.2)**

The following numbered 25 steps were used during the process of linking the database with the graphic software. This example use GPS Stations table which has coordinate data recorded as Northings and Eastings. A similar process was done for the Benchmarks and for Trigonometric stations table.

The list of steps for linking GPS station table to the GIS is given below:

1. Open Arc Map by clicking the icon
2. Click on tools
3. Select XY data
4. Search for data source {choose a table from the map}
5. Click on the data {e.g GPS Stations}
6. Select Eastings for X and Northings for Y Fields
7. On the same Add XY data window: Click on Edit
8. Select coordinate system
9. Click new to display New projected coordinate system window.
10. Key in a name such as GPS UTM
11. Select projection name for example Transverse Mercator
12. Key in Parameters for example False Eastings 500,000
13. False Northings 10,000,000
14. Central meridian 39 ( for the project area since Nairobi is in Zone 37)
15. Scale factor 0.9996 \*
16. Latitude of origin 0.000
17. Geographical coordinate system

18. Click on select
- 19.** Click on Africa
20. Click on arc 1960 projection
21. Click Finish
22. Double Click on GPS Stations Events
23. On Layer Properties window click Labels
24. Method -Leave it as default
25. Select Apply and OK.

#### **4 2.3 Internet application development methodology**

The second experiment involved exploring the possibility of using internet based software for displaying geodetic network to a remote user to find out the status of monuments by interactively passing a computer mouse over the control point on a screen or by clicking the symbol for the monument on the screen. This is made possible by the use of what is referred to in Fireworks as Hotspots. The use of hotspots is to incorporate interactivity into the web pages. Hotspots create an image map, which is a Hypertext Markup Language (HTML) code that defines a hot region in an HTML document. The regions as described in fireworks documentation, do not necessarily link anywhere but they are used to trigger behavior such as when a mouse is passed over them. In this application, Macromedia Dreamweaver 8, and Macromedia FireWorks 8 software were used. Macromedia Dreamweaver 8 is a HTML editor for designing, and developing websites, web pages and web applications. Fireworks was used in this application to display and animate web graphics created. The method adopted involved first creating a site using a wizard in Dreamweaver provided for this purpose and following direction given by the application. This was followed by setting the directory where all the data to be accessed are stored and designing the website. The images collected in the field were then linked to the graphic image in the hotspot regions which are areas around the control point symbols. Linking process was done for one control point at a time and checked for correct operation. The following steps describe the detailed process of linking:

- 1 Start Arc-Map select a file in this case GPS Stations and save it as a GIF
2. Open the map in Fireworks, and insert hotspots and reduce the size to cover one control point at a time with Edit Insert.
- 3. Open images and open a new canvas**
- 4. Specify the required size**
5. In the new canvas create Frames using Frames History available on the right side of the screen.
6. Select Image (the photograph of the particular monument) and copy
- 7. Paste in the frame and fit canvas.**
8. Copy the next image and paste it on the next frame. More frames can be added if available.
9. To change the delay for animation select GIF Animation looping and select forever as a choice. Speed of animation can also be changed by altering the default values.
10. Export the frame set through file
11. Select Export Wizard and select continue.
12. Select export after specifying the file name
13. Select HTML and Images from export drop down menu.
14. Select put the images in the sub-folder, then select Export
15. Click the hotspots to see the link menu.
16. On the link menu, specify the html file for example KJ1 1.htm that has just been exported. ( KJ 11 is a name of one of the stations).
17. Specify the target (Parent) then select <file> export the entire map in Fireworks.
18. Specify the file name of the final map to file and return to Dreamweaver.
19. In Dreamweaver, link GPS Stations with the final export maps from Fireworks such as GPS\_Networkfinal.htm.

The process of linking as indicated above was repeated for each station.

Similar steps were done for the Benchmark control points.

## 5.0 Transformation of coordinates and other support applications

The solution to transformation of coordinate problem was achieved by developing the following programs based on the formula shown in section 3.

1. Geodetic to UTM program transforms geodetic coordinates given in Latitude and Longitude to UTM coordinate system in Northings and Eastings. A sample listing is given in Appendix 1A for points located south of equator and Appendix 1B for points in the north.
2. UTM to Geodetic program transforms coordinates given in UTM coordinates system of Northings and Eastings to geodetic (Latitude, Longitude) in 2D. A sample listing is given in Appendix 2A for points in the southern latitude, Appendix 2B for points in the north of equator.
3. Geocentric to Geodetic program computes the geodetic values given the Cartesian X,Y,Z. A sample listing is given Appendix 3. Computational examples in the form of screen print-out is given in Appendix 9
4. Geodetic to geocentric program computes the geocentric coordinates X,Y,Z given the geodetic coordinates. A listing of this program is provided in Appendix 4. Sample screen print-out is given in Appendix 9.
5. Conversion of degrees, minutes and seconds to decimal degrees. Sample listing is given in Appendix 5 and computation using the program is given in Appendix 9.
6. Conversion of decimal degrees to degrees, minutes and seconds a sample listing is given in Appendix 6.

All the programs were written in Visual Basic 2005 using double precision data type as windows applications and have been tested for correct function.

### 5.1 Database applications

Database tables required data entry forms to be developed for accepting the records to be entered by the user\* Data entry forms were therefore developed to perform this function. An example of such a form is shown by figure 5.1. Additional database entry forms are included in Appendix 8. Sample database records are given in Appendix 10.

GPS\_Stations

Stationjd	
Station_name	148ST5
LatitudeJJ	1 29' 17.70455" S
Longitude_L	36 37' 01 .93966"E
Is_height_data_Availabl^	
Northings	9835375.07
Eastings	234861.94
order	r
CoordinateJJnits	Meters
Resolution	1 0.01 -
Monument_type	IRON_PIN
elevation (metres)	1751.642
Survey_method	Static
Projection	UTM
Buffer	6.096
Zone_number	1 37^J
HMapSheet_no	I48/3
Datum	Arc60
Date_Established	21/02/2003
Status	INTACT_USABLE
Status_Date	04/07/2007
HStnPicture_1	C:\Documents and Settings\W. AE
HStnPicture_2	C:\Documents and Settings\W. AE
comps_Id	1 n j J

stationjd	
Station_name	I48UT105
Is_height_data_Availal	
Northings	9881623.57
Eastings	241255.04
order	T3
CoordinateJJnits	(Meters ~3
Resolution	0.01
Monument_type	I.P.C.U 3
elevation	r 2313.28
Survey_method	Triangulation "3
Projection	UTM
Buffer	6.096
Zone_number	3 7 3
MapSheet_no	H8/1
Datum	Arc60 3
Date_Established	02/04/1970
Status	jPillar destroyed "3
Status_Date	03/07/2007
HStnPicture_1	JC:\Documents and Settings\W. AE
HStnPicture_2	C:\Documents and Settings\W. AE
comps_Id	I HI
FID	I 3
SHAPE	I 3

Record: H \ j | f 1 M M •\* of 25

Record: |< 1 1 • | M !•\*! of 375

'Sure 5.1: Sample data entry forms for GPS and triangulation stations displayed together.



FIGURE 5.2 B: INTERNET VIEW OF LXI/7 BENCHMARK STATION

FIGURE 5.2B: Close range view of the benchmark station LXI/7.





FIGURE 5.2C : A DIFFERENT VIEW OF LXI/7

Figure 5.2C: A different view of the benchmark LXI/7 using the internet application.

»

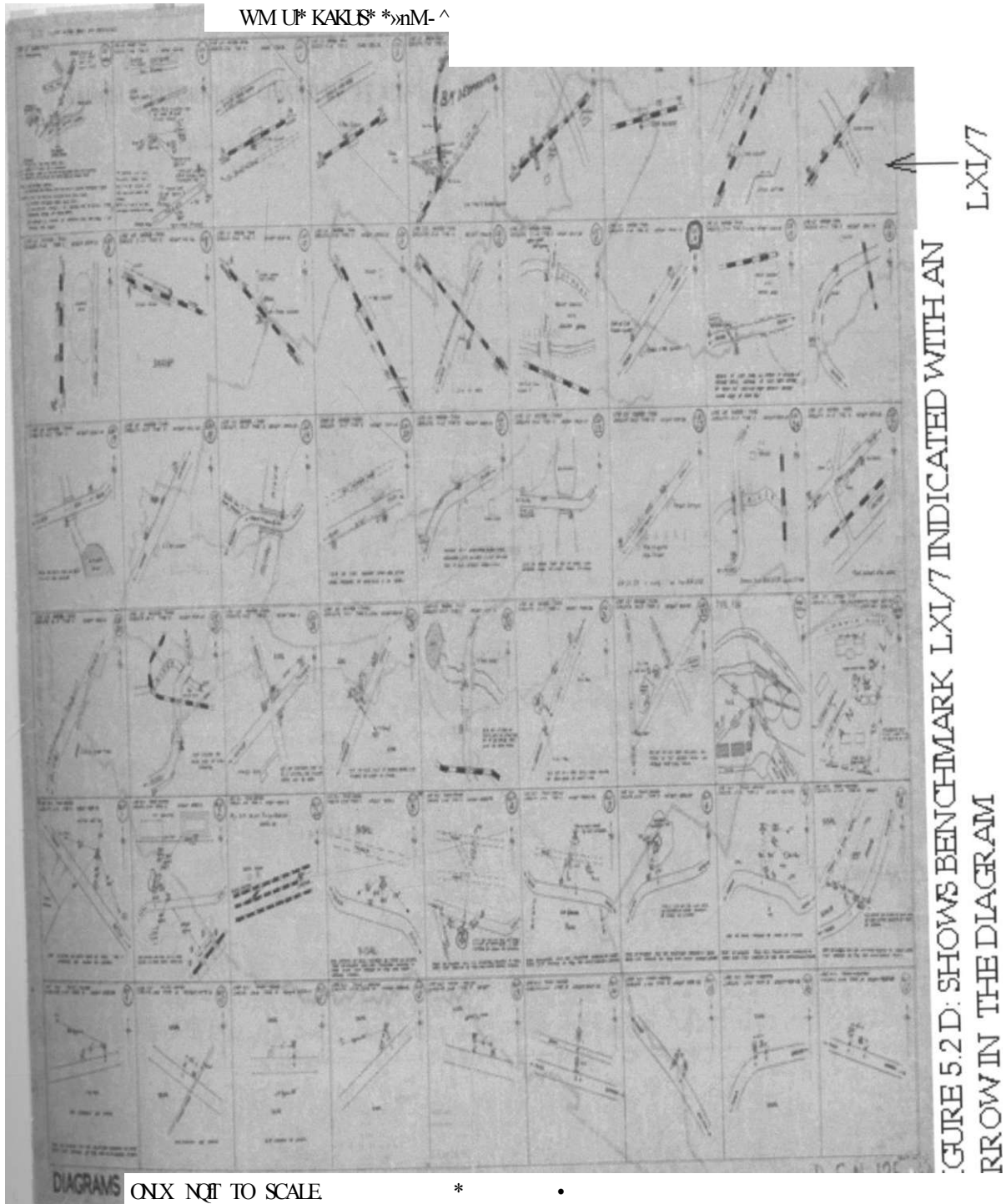


Figure 5.2D: Shows an Internet application to relate the benchmark station LXI/7 with **diagram** where it is drawn in the manual system. This diagram is a scanned image of **benchmark** diagram which is not drawn to scale. It has similar functions as the card

system used for storing horizontal control data. A sketch of how to locate each monument is drawn in its own compartment. The above diagram has a capacity to store (6X9) or 54 individual benchmark data. The height information shown on the diagram is given in feet.

**5.3 Internet based application for displaying images of geodetic GPS network control points.**

Internet display of geodetic GPS control points is demonstrated by figure 5.3A. photographic images of GPS monuments are displayed when station symbol is clicked with a mouse.

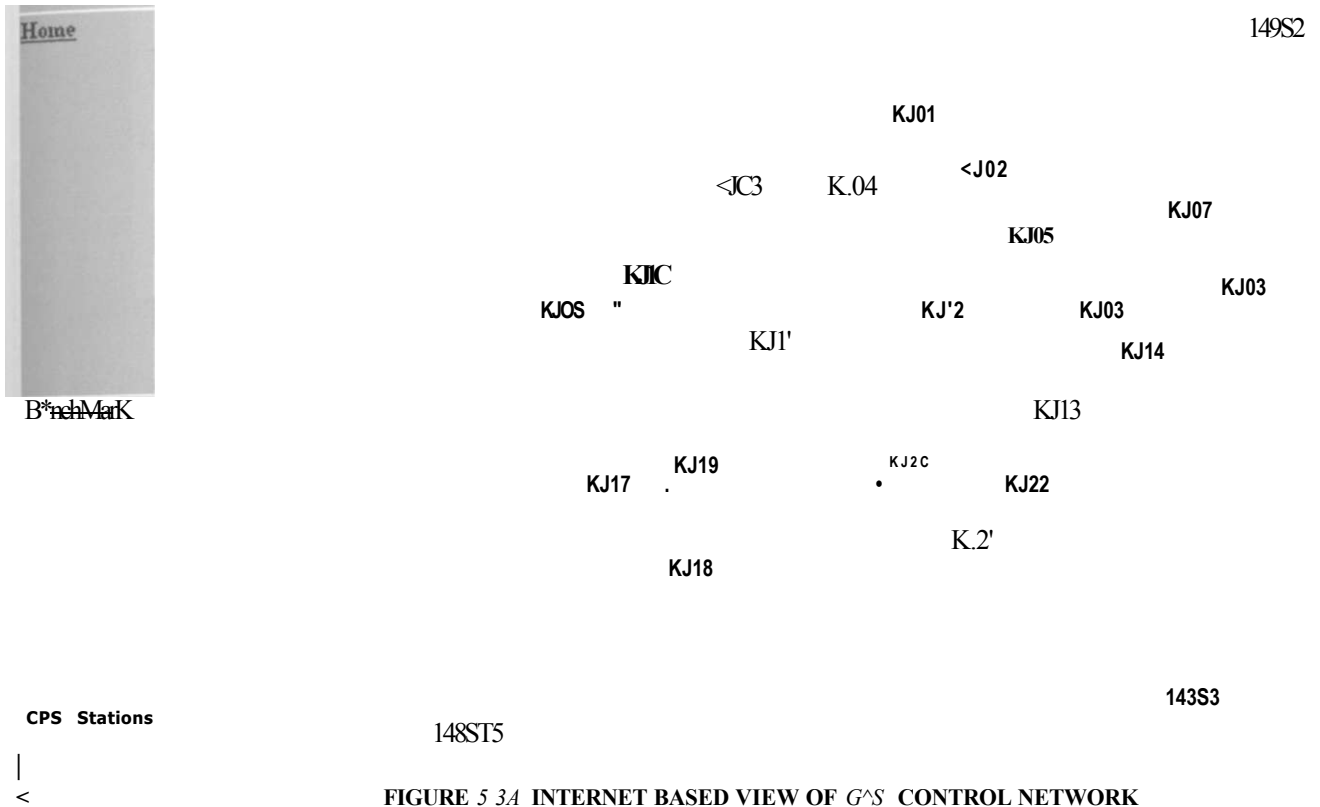


Figure 5.3A above shows the image of GPS geodetic network for Nairobi. Monuments associated with the control points are displayed alternately when the monument symbol is clicked by a computer mouse. Figure 5.3B below gives an example of a close view image of the control point. Figure 5.3C gives a view of the same station but from a different position.



FIGURE 5.3B: IMAGE OF A GPS MONUMENT FOR STATION KJ5

The internet ap-  
propriate data  
software available  
made to procure  
demonstrated by u



FIGURE 5.3 C: GPS CONTROL POINT KJ05 VIEWED FROM A DIFFERENT LOCATION

The internet application requires additional software to make it possible to display coordinate data from a database. The version of internet application development software available at Survey of Kenya does not have this capability and effort is being made to procure a more complete version. It is however possible to display images as demonstrated by the above images.

## 6.0 Tests and Results

All the three components of this system; the database, the Internet application and the transformation programs worked as planned.

### 6.1 Benefits

This system has direct benefits to departmental officers engaged in managing geodetic records and to those who require transformation of coordinates. It will save considerable amount of time and increase productivity of staff. The following section presents examples of database query statements and the resulting screen display of the searches.

#### 6.1.1 SQL Statement for station search using station name as the search criteria

The following is an SQL statement for performing a station search by station name and displaying the result of the search for one GPS station.

```
SELECT GPSSStations.Stationid, GPS Stations.Station name,
GPSSStations.Northings, GPSSStations.Eastings, GPSSStations.elevation,
GPS Stations.Status, GPS Stations.Status Date, H computation.Comp name,
GPS Stations.Zone number
FROM H computation INNER JOIN GPS Stations ON H computation.Comps id =
GPS Stations.Comps Id
GROUP BY GPS Stations.Station id, GPS Stations.Station name,
GPS Stations.Northings, GPS Stations.Eastings, GPS Stations.elevation,
GPS Stations.Status, GPS Stations.Status Date, H computation.Comp name,
GPS Stations.Zone number
HAVING (((GPS Stations.Station name)=[PLEASE KEY IN STATION NAME TO
SEARCH:]));
```

## Enter Parameter Value

A screenshot of a graphical user interface dialog box. At the top, the text "PLEASE KEY IN STATION NAME TO SEARCH:" is displayed in a monospaced font. Below this text is a single-line text input field containing the characters "KJ11". At the bottom of the dialog box, there are two rectangular buttons: "OK" on the left and "Cancel" on the right. The entire dialog box is enclosed in a dark border.

FIGURE 6.1.1A:STATION" NAME TO BE SEARCHED

Figure 6.1.1 A is displayed when the select query shown above is performed. The user can then key in the station name to be searched. If the station is found, the Table 6.1.1 shown below will list the result of the search. The example given is for a GPS station named KJ11.



e<sup>1</sup> Query\_SFARCH\_BY\_STATION\_NAME: Select Query

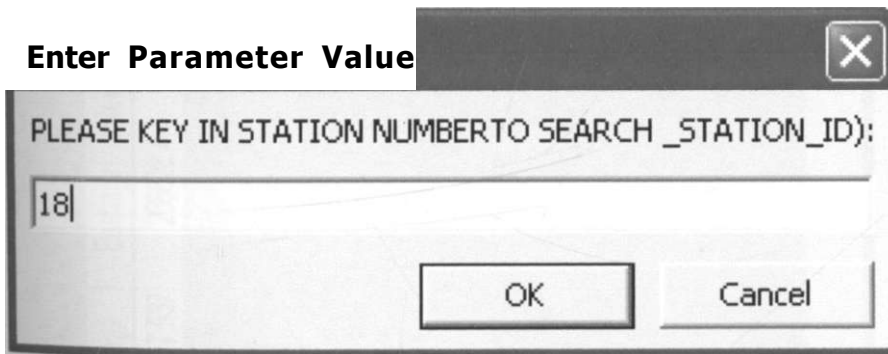


Station_id	Station_name	Northings	Eastings	elevation (metres)	Status	Status Date	Comp Name	Zonenumber
14	KJ11	9859246.533	256421.085	1561.377	INTACT_USABLE	21/06/2007	TS366	37

Record: 1 of 1

TABLE 6.1.1 A :SHOWS QUERYRESULT AFTER DATABASE SEARCH BY STATIONNAME "KJ11

```
FROM Hcomputation INNER JOIN GPS_Stations ON H_computation.Comps_id =
GPSStations.CompsJd
GROUP BY GPS_Stations.Station_id, GPSStations.Stationname,
GPS_Stations.Northings, GPSStations.Eastings, GPSStations.elevation,
GPSStations.Status, GPSStations.StatusDate, Hcomputation.Compname
HAVING (((GPS_Stations.Station_id)=[PLEASE KEY IN STATION NUMBERTO
SEARCH (STATION ID:)]));
```



**FIGURE 6.1.1B : STATION IDENTIFIER MJMBER TO BE SEARCHED**

is a search by station identifier,,jt gets data from two tables in the database and displays it as shown in table 6.1.1 B below when the user enters a station identifier number and clicks OK.

r  
i s

QueryJMRCHJYTATIONJD: Select Que\*\_y



Stationjd	Stationjame	Northings	Eastings	elevation (metres)	Status	Status_Date	Comp_Name
f KJ15		9855517.537	286628.4	1365.482	INTACT_USABLE	03/07/2007	TS366

Record M|j| \ |•|\_| of 1

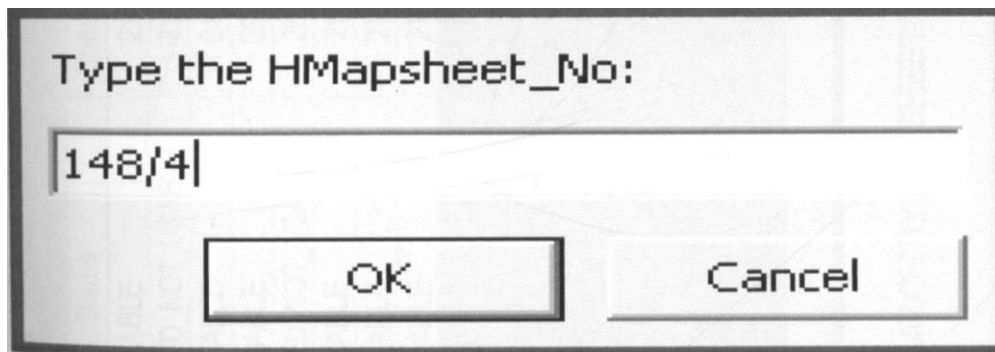
TABLE 6.1.1B: SHOWS QUERYRESULT OF DATABASE SEMCH FOR GPS STATION WITH STATION IDENTIFIER NUMBER" 18"

### 6.1.3 SQL Statement for station search using map sheet number as the search criteria.

GPS stations contained within this map are retrieved and displayed.

```
SELECT GPSStations.Stationid, GPSStations.Stationname, GPSStations.Status,
GPS Stations.Status Date, H Mapsheet.Mapsheet Id, GPS Stations.HMapSheet no,
GPSStations.Northings, GPSStations.Eastings
FROM HMapsheet INNER JOIN GPS Stations ON H Mapsheet.HMapsheet No =
GPS_Stations.HMapSheet_no
GROUP BY GPS Stations.Station id, GPS Stations.Station name,
GPS Stations.Status, GPS Stations.Status Date, H Mapsheet.Mapsheet Id,
GPS Stations.HMapSheet no, GPS Stations.Northings, GPS Stations.Eastings
HAVING (((GPS Stations.HMapSheet no)=[Type the HMapsheet No:]));
```

### Enter Parameter Value




The image shows a standard Windows-style dialog box. At the top, the title bar reads "Enter Parameter Value". Below the title bar, there is a label "Type the HMapsheet\_No:" followed by a text input field. The input field contains the text "148/4". At the bottom of the dialog box, there are two buttons: "OK" on the left and "Cancel" on the right.

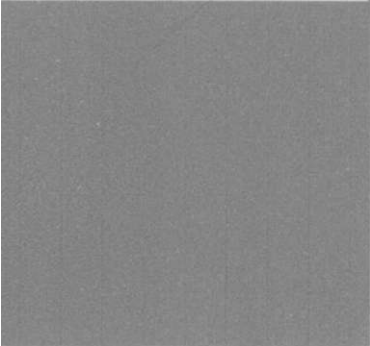
**FIGURE 6.1.3: SHOWS ENTRY OF MAPSHEET INTUIVIBER TO SEARCH:**

Figure 6.1.3 Shows entry of map sheet number 148/4 used here as the search criteria. The results of this query are given in Table 6.1.3 B on the next page.

sF Query\_by\_HMapsheet: Select Query



Stationjd	Station_name	Status	Status Date	MapSheetJd	HMapSheet_no	Northings	Eastings
\$KJ 11		INTACT_USABLE	21/06/2007	4	148/4	9859246.533	256421.085
15KJ12		DESTROYED_NOT_USABLE	21/07/2007	4	148/4	9861107.686	268083.412
16 KJ13		INTACT_USABLE	04/07/2007	4	148/4	9854722.673	275771.774
21 KJ18		INTACT_USABLE	29/07/2007	4	148/4	9845915.976	250962.559
22 KJ19		DESTROYED_NOT_USABLE	29/07/2007	4	148/4	9851765.427	251375.325
23 KJ20		INTACT_USABLE	21/06/2007	4	148/4	9852293.211	265707.212
24 KJ21		INTACT_USABLE	21/06/2007	4	148/4	9847119.741	270063.268
25 KJ22		INTACT_USABLE	21/06/2007	4	148/4	9850526.016	273677.291




Record: \_M\_ J | f • | • | J of 8

FIGURE 6.1.3B: SHOWS RESULT OF SEARCH BY MAPSHEET NUMBER 148/4 TO GET THE GPS STATIONS ON THIS SHEET

## 6.2 Graphic Queries

This is a Sample results from the system in response to a graphic query of Station KJ1




---

<Top-most layer >

---

rts Events      Location:    1256,418.722    9,859,247.678 Meters

Field	Value
Station Jd	14
Station_name	KJ11
latitude_B	1 16'21. 43488"S
longitude_L	36 48' 39.80047"E
Is_Height_data_available	1
Northings	9859246.533
Eastings	256421.085
Order	2
Coordinate_units	Meters
Resolution	0.01
Monumentjype	BRASS_CAP_WITH_IRON_PIN
elevation	1561.377
Survey_method	Static
Projection	UTM
Buffer	6.096
Zone_number	37
HMapSheet_no	148/4
Datum	Arc60
Date_Established	25/02/2003
Status	INTACT_USABLE
Status_Date	21/06/2007
HStnPicture_1	SP_A0138.jpg
HStnPicture_2	SP_A0018.jpg
Comps_Id	11
Specific_Locality	<null>
Shape	Point

Figure 6.2A : Shows result after graphic query performed on KJ 11. It displays the coordinates and other details.

The interactivity is achieved by clicking the control point shown above in blue colour, to this case a layer of Orthophoto image was used as a background.

### 6.3.1 Transformation computation and database queries combined with calculations

0 ( d ®

**Form1**

22 October 2007 v

**TRANSFORMATION OF GEODETIC COORDINATES TO UTM COORDINATES**

Input Station Name 149S3

Output: Northings^astings

Ellipsoid Used Clarke 1880(M odified) Unit: Metres

Northings(N)

Latitude-(Phi)	-1.46863576111111	S v	With FN	9837592.78712578
Longitude(Lamda)	37.0624021166667	E v	Without FN	-162407.212874219

Unit: Decimal Degrees

Semi-major Axis(a) 6378249.145 Eastings(E)

Flattening (f) 0.003407561379 With FE 284419.099898831

Without FE -215580.900101169

Zone(Zn] 37

For southern Latitudes enter minus sign first then the value

For Clarkel 880 a=6378249m>0.003407561379

For WGS84 a=6378137.0m, f=0.00335281066474

Transform to  
UTM-Coordinates

FN and FE are 10000000 and 500000 respectively

October2007

Figure 6.3 A : Sample output of converted UTM coordinates after entering geodetic coordinates.

Figure 6.3 A Shows a sample of one of the coordinate transformation program result. In this transformation, coordinate value in geodetic system of Latitude and Longitude are entered on the input side of the form and result is obtained on the right hand side as output. Reverse computation is carried out by using UTM to Geodetic program as shown in Figure 6.3 B below. Conversion programs to change decimal degrees to degrees, minutes and seconds can then be used to view the results from this computation in

degrees minutes and seconds format. More samples of conversion screen print-outs including Cartesian to geodetic transformation are given in Appendix 9.

### Form1

26 October 2007 v

#### TRANSFORMATION OF UTM COORDINATES TO GEODETIC COORDINATE SYSTEM

Station Name	149S3			
	Input Northings and eastings		Output: Latitude and Longitude	
			Units in decimal degrees	
Ellipsoid Used	Clarke 1880(Modified) v			
Units in Metres				
Northings(N)	9837592.787	Latitude	1.46864721402153	4
Eastings(E)	284419.100	Phif	1 4694813276331	
Zone(Zn)	37			
Semi-Major Axis	6378249.145	Longitude	37.062402117587	
Flattening(f)	0 003407561379			
For Darke 1880 a= 6378249.145 m, f=0.003407561379			Convert	
For WGS84 a=6378137.0,f=0.00335281066474			Coordinates To	
			Geodetic	
Bessel 1841: a=6377397.155, f= 0.003342773182				

October 2007

FN and FE are 1000000 and 500000 m respectively

Figure 6.3 B : Sample output of converted geodetic coordinates after entering UTM coordinates.

### 6.3.2 SQL Statement for generating results shown in Table 6.3.3A

This database query gives results as in Table 6.3.3 A

```
SELECT utmcomputation.Stationid, utmcomputation.Stationname,
utm computation.Northings, utm computation.computed Northing, [northings]-
|computed_northing] AS delta N, utmcomputation.Eastings,
|eastings]-[computed eastings] AS deltaE,
|delta N]* 1000 AS delta_N_mm, [delta E]* 1000 AS delta_E_mm
FROM utm computation;
```



Stationjd / Station_name	Northings	computed_Northing	delta N	Eastings	computed_Eastings	delta_E	delta_N_mm	delta E mm
J148ST5	9835375.07	9835375.0701364	-0.0001364	234861.94	234861.939938346	6.165399E-05	-0.136399641633034	0.0616539909970015
2149S2	9878315.023	9878315.0229305	6.950088E-05	289949.592	289949.591983393	1.660699E-05	0.0695008784532547	0.0166069949045777
3149S3	9837592.787	9837592.78691732	8.268096E-05	284419.1	284419.099898831	0.000101169	0.082680955529213	0.10116898920387
4KJ01	9873011.157	9873011.15738194	-0.000381941	266117.332	266117.332201338	-0.000201338	-0.381940975785255	-0.201338029000908
5KJ02	9869785.482	9869785.48216355	-0.00016355	270618.912	270618.911928019	7.198099E-05	-0.163549557328224	0.0719809904694557
6KJ03	9868559.858	9868559.85803088	-3.08808E-05	254126.032	254126.032048162	-4.8162E-05	-0.0308807939291	-0.048161979066208
7KJ04	9868370.837	9868370.83726515	-0.000265151	261753.13	261753.12938759	0.0006124100	-0.265151262283325	0.612410018220544
8KJ05	9865665.45	9865665.45049073	-0.000490731	273833.428	273833.428309532	-0.000309532	-0.490730628371239	-0.309531984385103
9KJ06	9861333.871	9861333.87085937	0.0001406297	278642.179	278642.178666734	0.000333266	0.140629708766937	0.333265983499587
10KJ07-*	9867125.458	9867125.45738351	0.0006164908	284706.218	284706.218200042	-0.000200042	0.616490840911865	-0.20004203543067
11KJ08	9862608.535	9862608.53497992	2.007931E-05	288340.853	288340.853158417	-0.000158417	0.0200793147087097	-0.158416980411857
12KJ09	9861021.492	9861021.49205506	-5.50598E-05	242256.828	242256.82754862	0.0004513800	-0.0550597906112671	0.45138000859879
13KJ10	9863207.837	9863207.83716523	-0.000165232	247825.54	247825.540348379	-0.000348379	-0.16523152589798	-0.348378991475329
14Kill	9859246.533	9859246.53259065	0.0004093498	256421.085	256421.084698574	0.000301426	0.409349799156189	0.301425985526294
15KJ12	9861107.686	9861107.68601059	-1.05891E-05	268083.412	268083.412290768	-0.000290768	-0.010589137673378	-0.290767988190055
16KJ13	9854722.673	9854722.67268016	0.0003198404	275771.774	275771.774347906	-0.000347906	0.319840386509895	-0.347906025126576
17KJ14	9858741.974	9858741.97357034	0.0004296601	281528.489	281528.489234534	-0.000234534	0.429660081863403	-0.23453397443518
18KJ15	9855517.537	9855517.53653497	0.0004650299	286628.4	286628.400254349	-0.000254349	0.46502985060215	-0.254348968155682
19KJ16	9854377.754	9854377.7545729	-0.000572899	239534.745	239534.744710855	0.000289145	-0.572899356484413	0.289144983980805
20KJ17	9850571.212	9850571.21181503	0.00018497	245434.915	245434.914329473	0.0006705270	0.18496997654438	0.670527020702139
21KJ18	9845915.976	9845915.9761744	-0.000174399	250962.559	250962.558612101	0.000387899	-0.174399465322495	0.387898995541036
22KJ19	9851765.427	9851765.42664268	0.0003573187	251375.325	251375.324405096	0.0005949040	0.357318669557571	0.594904005993158
23KJ20	9852293.211	9852293.21146696	-0.000466961	265707.212	265707.211392134	0.000607866	-0.466961413621902	0.607865978963673
24KJ21	9847119.741	9847119.74073084	0.0002691597	270063.268	270063.268125251	-0.000125251	0.26915967464447	-0.125251011922956
25KJ22	9850526.016	9850526.01625966	-0.000259658	273677.291	273677.290866101	0.0001338990	-0.259658321738243	0.133899040520191

TABLE 6.3.3 A :SHOWS THE DIFFERENCES OF COMPUTED UTM VALUES WITH THE PROGRAM GIVEN IN APPENDIX 1A AND THE COORDINATES FOR NAIROBI MAPPING PROJECT. THE LAST TWO COLUMNS INDICATE THE DIFFERENCES IN MILLIMETRES

The interpretation of these results is that the programs developed computes the transformation accurately and the small differences in the order of 0.02mm to 0.7 mm appearing in the last two columns above could be due to rounding off errors that occur when the reverse computation use rounded values of a limited number of decimal places.

### 6.3.3 SQL statement for generating results shown in Table 6.3.3B

This database query gives results as in table 6.3.3B. It calculates the differences obtained from subtracting the original values from the reverse transformation computation. The last two columns show the values in seconds after multiplying the difference in decimal degrees by 3600 to get values in seconds. The SQL query used here also demonstrates the capability of retrieving information of data stored in a database by performing a query. This saves considerable time and avoids retyping data which can introduce errors.

```

SELECT lat_long_table.Station_id, lat_long_table.Station_name,
lat_long_table.Latitude_B_decimal_deg, latlongtable.LatitudebyUTMconvert,
[Latitude B decimal deg]-[Latitude by UTMconvert] AS delta_lat,
lat_long_table.Longitude_L_decimal_deg, lat_long_table.Logitude_by_UTMconvert,
[Longitude L decimal deg]-[Logitude by UTMconvert] AS deltaLong, [delta_lat]*3600
AS LatInSec, [deltaLong]*3600 AS Long_in_Sec
FROM lat_long_table
WHERE (((lat_long_table.Station_id)<26));

```

<i>jaationj</i>	<i>LatitudTBJeSleg^^</i>								
I148ST5	1.4982512638888!	1.48826381772897	-1.75538500799544E-05	35.5172054311111	36.3172064616951	-5 84002E-10	•0.0631338602878357	-2.10241 E-06	
JU9S2	1 10042096566667	1.1004231151C061	-8.14843394003617E-06	37.1123510277776	37.1123610279368	-1 58998E-10	•0.0293343621841302	-5.72393E-07	
149S3	1.46863576111111	1 46864721402153	-1.145291042004E-05	37.0624021166667	37.0624C2117587	-9 20295E-10	-0.041230477512144'	-3.31306E-06	
KJ01	1.14823430277776	1.1482448472339	-1.05444561203319E-05	35.8982475366667	36 39824756^8721	1.794596 E-09	•0.037930042C321147	6.460547 E-06	
KJ02	1.1774312	1.17744159846291	-1.03984629103662E-05	33.9386614388889	36.3386614395493	-6.604E-10	•0.0374344664762383	-2.37744E-06	
KJ03	1.188398225	1.18841328156109	•1 20565810893365E-05	33.7905105388889	36.79051C5384774	4.115037E-10	-0.0434336915237712	1.4814' 3E-06	
KJCu	1.19016163355556	1.19017296999871	-1.13394431503158E-05	35.8590157222222	36.3590157236867	-1.4645 E-09	•0.040821995:400568	-5 2722 E-06	
KJ05	1.21470742777776	1.21471785976332	-1.04319855393494E-05	35.967509058333:	36.3675090555633	2.770001 E-09	•0.037555147943^577	9.972004E-06	
KJ06	1.25390860277776	1.25391391266648	-1.03099087001013E-05	37.0106770555556	37.3106770585627	-3 00710E-09	•0.037115671:203646	-1.08256 E-05	
KJ07	1.20157690277776	1.2015832445:709	-9.34175930983879E-06	37.0651905361111	37.36519C53/3287	1.782404 E-09	•0.0336303335155996	6.416653 E-06	
KJ08	1.24244691111111	1.2424532525:859	-9.3414274^993189E-06	37.0978134338889	37.39781:4624756	U13298E-09	•0.0336291389279708	5.087873E-06	
KJ09	1.25646212222222	1.25647312875816	-1 40065359393699E-05	35.6838494361111	36.5838494402347	-4.1236E-09	•0.050423529:838917	-1 4845 E-05	
KJ10	1.236/3&8388886	1.23675278771265	-1.3^98823^701552E-05	35.7338817416667	36.7338817385608	3.105903E-09	•0.0475157655725589	1.118125E-05	
KJ11	1.2726206	1.27263346671705	-1.266671^7043914E-05	35.8110556361111	36.3110556820454	4.065697 E-09	•0.0456301813796902	1.463651 E-05	
KJ12	1.2558794833333:	1.25589381909623	-1.13357628993111 E-05	35.9158259472222	36.315B2594/6259	2.596302 E-09	-C. 04380874643968	9.3^6687 E-06	
KJ13	1.31367029722222	1.31368137846535	-1.10812331293837E-05	35.9848448166667	36.3848448135569	3.109804E-09	•0.0398324392679412	1.119529E-05	
KJ14	1.27736620555556	1.2773754330:936	-1 02274838003849E-05	37.0365881444444	37.3365861423524	2.092001 E-09	•0.0368189416803055	7.531204E-06	
KJ15	1.30655962222222	1.30656360086089	-9.97865867002901 E-06	37.0823819277776	37.3823819255062	2.271605E-09	•0.0359231712121044	8.177778 E-06	
KJ16	1.3165029	1.31651789147532	-1.49914753203253E-05	33.6593469333333	36.3593469359542	-2 62089 E-09	•0.053939311152091'	-9.i3522E-06	
KJ17	1.35096676388889	1.350931452C789	-1.4688190010D482E-05	33.7123078372222	36.71230790327^9	-6.0497E-09	-0.052877484C361734	-2.17789E-05	
KJ18	1.39310199166667	1.3931 649C537	-I 44988703293932E-05	35.7619181416667	36.7619161451722	-3 50551 E-09	•0.0521359331876154	-1.26198E-05	
KJ19	1.3402196533333:	1.34023355631841	-1.38979850801135E-05	33.7656746361111	36.7656746414801	-5 36900 E-09	-0.0500327462884387	-1.93284E-05	
KJ20	1.33556230277776	1.33557460956132	-1.23067835393061 E-05	33.8944154555556	36.894415461029	-5.4734E-09	-0.0443344207433019	-1.97042E-05	
KJ21	1.38237515555556	1.38238741767417	-1.22621186103513E-05	33.9335062166667	36.33350E2155587	1.107999E-09	•0.0441436269961848	3.988796 E-06	
KJ22	1.35160262222222	1.35161424247059	-1.16202483693812 E-06	35.9659983330556	36.36999633^2703	-1 21470E-09	-0.0418328941319324	-4.37292E-06	

TABLE 6.3.3 B: SHOWS THE DIFFERENCES BETWEEN FORWARD AND REVERSE UTM COMPUTATION USING UTM VALUES AS INPUT. THE LAST TWO COLUMNS GIVE RESULT IN SECONDS.

## **7.0 Conclusions and Recommendations**

### **7.1 Conclusions**

The system developed in this project addresses some of the problems in Survey of Kenya. These problems are summarized as lack of a database for storing geodetic records, no interactive capability of querying a geodetic map, retrieval of co-ordinates information is slow, inefficient updating of geodetic map and a means of transforming coordinates is not available. The following statements present the conclusions of this study:

A means of storing coordinate records is now possible. This has been provided by creating the geodetic network database and linking the database records with GIS software. A sample database record is shown in Appendix 10. A means of querying the database is provided by clicking the monument symbol in the digital map with a computer mouse. The possibility to display photographic images of the control stations interactively is demonstrated by incorporating digital photographic images as well as the actual information in the GPS Stations table and the Benchmark table. These functionality have been demonstrated in chapter 5 and 6 of this document.

The possibility to compute transformation of coordinates is provided and sample computations is included in Appendix 9. The comparative differences of the UTM computation shown in Table 6.3.3A and Table 6.3.3B was found to be very small when the differences between reverse and forward computation values were compared.

Cartesian to geodetic and its inverse computation also gave good results when tested using sample computation. The result of this test is given in Appendix 9.

The results in these tables confirm that conversion programs achieved their design objective. The possibility to update geodetic control maps quickly is now possible. This is provided by simply adding new control points in the database. Additional layer of graphic information is easy to add in a GIS environment.

The storage problem is addressed by having map data in digital form and storing them in compact disc (CD) which requires small space for storage. Possibility to view the same data among different users can be provided by networking the computers.

Improved Visual interpretation is provided by having a topographic background as well as digital photographic images that can be interactively viewed alternately as demonstrated both in the GIS software and by the internet model.

The inclusion of orthophoto layer in the GIS improves visual interpretation further.

Service to the members of the public is improved because the access speed to information is much faster for retrieving data with computers when compared to the manual method of accessing the cards.

GIS approach to display of geodetic network information offered more flexibility and superior output of data.

The design objectives were met as the solution to the problems above have been demonstrated and confirmed by testing

## **7.2 Recommendations**

The system as designed is now capable of recording geodetic data and retrieving textual information stored in the database. The following recommendations will further improve the operation of the database.

The status information for classical trigonometric stations should be collected to confirm their suitability for occupation. The collected information can be included in future by updating existing records to add photo images of the trigonometric control points.

Construction companies should consult with Survey of Kenya for information regarding the position of control points. This will help reduce the rate of destruction of survey Monuments.

-

The Survey of Kenya (SOK) should invest in software dedicated to geodetic computerization program similar to those used in this project. The database management software however should have a higher capacity since Access has a memory limit of 2 Giga Bytes (2GB). The use of Internet approach to display database records requires additional software module to be procured. SOK need to add these modules for improved functionality. In addition to the software, SOK need to invest on a dedicated server for effective internet application.

Transformation programmes need to be customized to a particular project area to reduce data entry errors associated with keying in of the parameters such as the flattening, zone and semi-major axis. If these parameters are incorporated in the program, the user will be expected to be more productive. Incidences of incorrect input of values will also be limited to the coordinate data being transformed only.

More precise transformation parameters between WGS84 and the local datum (Arc 1960) are needed. This will enable the development of accurate datum transformation in future.

## REFERENCES

1. Groot R, McLaughlin J, 2003: Geospatial Data-Infrastructure-Concepts, Cases, and Good Practice, Oxford University Press
2. Howe D. R, 1982: Data Analysis for Database Design, ITC LIS 16.
3. Hofmann-Wellenhof B, Lichtenegger. H and, Collins. J, 1992: GPS Theory and Practice, (fourth -revised edition), Springer Wien NewYork
4. JICA Study Team, 2003: The study for the establishment of the spatial data framework for the city of Nairobi in the Republic of Kenya, JICA
5. Kaplan E, 1996: Understanding GPS principles and Applications, Artech House Publishers.
6. Leick A, 2004: GPS Satellite Surveying, (Third edition), John Wiley and Sons
7. Lo C.P, Yeung A.K.W, 2006: Concepts and Techniques of Geographic Information Systems, Prentice-Hall of India
8. Ministry of Lands and Settlement, 2004: Strategic Plan 2003/4-2005/2006, Ministry of Lands and Settlement, Kenya
9. Okumu B.M, 1990: Report on Destruction of Survey Monuments Submitted to the Land Surveyors Chapter of the Institution of Surveyors of Kenya, ISK
10. Ramon A. Mata-Toledo, Cushman. P, 2005: Fundamentals of SQL Programming, Tata McGraw-Hill.
11. Schneider D. I, 2005: An Introduction to Programming Using Visual Basic.NET, Prentice-Hall of India.
12. Survey Act Chapter 299, 1989, The Government Printer, Kenya
- <sup>13</sup>- Vine M, 2005: Microsoft Access VBA Programming 2<sup>nd</sup> Edition, Thomson
- <sup>14</sup>- <http://www.govtech.com/gt/95306> ( Accessed on 22/10/07)
- <sup>15</sup>- <http://vents2.esri.com/uc/2006/sag/list/index.cfm?fa=detail&SID=427>  
(Accessed on 22/10/07)
- <sup>16</sup>- [http://en.wikipedia.org/wiki/Transverse\\_Mercator\\_projection](http://en.wikipedia.org/wiki/Transverse_Mercator_projection) (Accessed on 26/10/07)
- <sup>17</sup>- <sup>13</sup><http://en.wikipedia.org/wiki/WGS84> (Accessed on 20/10/07)

18. <http://en.wikipedia.org/wiki/Gis> ( Accessed on 26/10/07)



## APPENDIX 1A

### COORDINATE TRANSFORMATION FROM GEODETIC TO UTM SOUTHERN LATITUDE

Public Class Form 1

'Coordinate Transformation Geodetic To UTM Southern Latitude

'For Part of Msc GIS Project By W.Abuto

•F56/P/9061/2005

'October 2007

•Programme Computes the Northings and the Eastings given the geographical coordinates and display the resulting coordinates in UTM

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'Dim LatPhi As Double ' = CDb1(TxtLatPhiL.Text)' Latitude Value

'Dim LongLamda As Double - CDb1(txtLongLamdaL.Text)' Latitude Value

'Dim a As Double - CDb1(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

'Dim f As Double ' = CDb1(TxtfL.Text) ' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

'Dim ZoneL As Double - CDb1(txtZoneL.Text)' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

'for clarke 1880 a=6,378,249 international metres, ellipticity fis 1/293.465

'as in the formular

Dim alpha As Double

Dim beta As Double

Dim phi As Double

Dim gamma As Double

Dim delta As Double

Dim epsilon As Double

Dim eta As Double

Dim L As Double

Dim N As Double

Dim t As Double

Dim FN As String = "1000000" 'Term added to get False Northings

Dim FE As Double = 500000 'Term added to get False Eastings

MsgBox(ComputeBPhi(alpha, beta, phi, gamma, delta, epsilon)) 'display of  
computed foot point Latitude value(arc Length of Meridian)

'arcLength = Val(TxtarcLengthL.Text) The BPhi value

'YY = arcLength ' the other terms of y

MsgBox(ComputeYY(t, N, phi, L, eta)) 'the result of yy computation

'display as message of xx value

MsgBox(ComputeXX(t, N, phi, L, eta)) 'the result of yy computation

'function call to display the eastings value already computed and converts it to string  
for display in the TextBox

'FE is added and all converted to string

TxtEasting.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta)) 'to display result  
of True Easting to the user in a textBox

TxtEastings.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta) + FE) 'to  
display result of False Easting computation to the user in a textBox

TxtarcLengthL.Text = Convert.ToString(ComputeBPhi(alpha, beta, phi, gamma,  
delta, epsilon) + ComputeYY(t, N, phi, L, eta)) 'to display result of arc Length  
computation to the user in a textBox

'the ellipsoidal distance from the equator

'Displays the resulting total of both the Bphi value and the extra terms computed for  
the Northings

txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text) + Val(FN)) The  
total added values of Bphi plus YY value

'txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text)) 'The total added  
values of Bphi plus YY value

End Sub

•

Dim t As Double

Dim FN As String = "10000000" Term added to get False Northings

Dim FE As Double = 500000 'Term added to get False Eastings

MsgBox(ComputeBPhi(alpha, beta, phi, gamma, delta, epsilon)) 'display of  
computed foot point Latitude value(arc Length of Meridian)

'arcLength = Val(TxtarcLengthL.Text) The BPhi value

'YY = arcLength ' the other terms of y

MsgBox(ComputeYY(t, N, phi, L, eta)) 'the result of yy computation

'display as message of xx value

MsgBox(ComputeXX(t, N, phi, L, eta)) 'the result of yy computation

'function call to display the eastings value already computed and converts it to string  
for display in the TextBox

'FE is added and all converted to string

TxtEasting.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta)) 'to display result  
of True Easting to the user in a textBox

TxtEastings.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta) + FE) 'to  
display result of False Easting computation to the user in a textBox

TxtarcLengthL.Text = Convert.ToString(ComputeBPhi(alpha, beta, phi, gamma,  
delta, epsilon) + ComputeYY(t, N, phi, L, eta)) 'to display result of arc Length  
computation to the user in a textBox

'the ellipsoidal distance from the equator

'Displays the resulting total of both the Bphi value and the extra terms computed for  
the Northings

txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text) + Val(FN)) 'The  
total added values of Bphi plus YY value

'txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text)) "The total added  
values of Bphi plus YY value

End Sub

Function ComputeBPhi(ByVal alpha As Double, ByVal beta As Double, ByVal phi As Double, ByVal gamma As Double, ByVal delta As Double, ByVal epsilon As Double) As Double

Dim LatPhi As Double = CDbL(TxtLatPhiL.Text)' Latitude Value

Dim LongLamda As Double = CDbL(txtLongLamdaL.Text)' Latitude Value

Dim ZoneL As Double = CDbL(txtZoneL.Text) ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim a As Double - CDbL(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

Dim f As Double - CDbL(TxtfL.Text) ' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

•Dim ZoneL As Double '= 37 '= CDbL(txtZoneL.Text) ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

TxtaL.Text = Convert.ToString(6378249.145) 'default setting for a value

TxtfL.Text = Convert.ToString(0.003407561379) 'default setting for f value

'txtZoneL.Text = Convert.ToString(37) 'default setting forZone value...CHANGED

Tor clarke 1880 a=6,378,249 international metres, ellipticity f is 1/293.465

a = 6378249.145 'clarke 1880 values for a as default

f= 0.003407561379 'clarke 1880 values for f

Dim b As Double

Dim K As Double

Dim BPhi As Double

Dim ns As Double

$b = a * (1 - f)$  'the semi-minor axis

$\text{phi} = \text{LatPhi} * (\text{Math.PI} / 180)$  'Converting Decimal degrees to Radians the ->phi values

$\text{ns} = (a - b) / (a + b)$  ' same as small (n) in the formula

$\alpha = (1 / 2) * (a + b) * (1 + (1 - M) * (\text{ns}^2) + (1 / 64) * (\text{ns}^4))$

$\beta = -(3 / 2) * \text{ns} + (9 / 16) * \text{ns}^3 - (3 / 32) * \text{ns}^5$

```

gamma = (15/ 16) * ns ^ 2 - (15 / 32) * ns ^ 4
delta = -(35 / 48) * ns ^ 3 + (105 / 256) * ns ^ 5
epsilon = (315/512) * ns ^ 4

```

•LatPhi is what the user entered in decimal Degrees but the computer use only values in Radians

```
K = 0.9996
```

```
BPhi = alpha * (phi + beta * Math.Sin(2 * phi) + gamma * Math.Sin(4 * phi) + delta
* Math.Sin(6 * phi) + epsilon * Math.Sin(8 * phi))
```

Return K \* BPhi ' to return the result of this computation(scale applied to UTM **0.9996** multiply with result of Bphi )

```
End Function
```

```
Function ComputeYY(ByVal t As Double, ByVal N As Double, ByVal Phi As
Double, ByVal L As Double, ByVal eta As Double) As Double
```

'Y = BPhi+YY so this part computes YY part, already BPhi has been computed above

```
Dim LatPhi As Double = CDbI(TxtLatPhiL.Text) ' Extracting Latitude Value from
textBox
```

```
Dim LongLamda As Double = CDbI(txtLongLamdaL.Text)' Latitude Value
```

```
Dim a As Double = CDbI(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid
e.g clarke 1880 (modified)
```

```
Dim f As Double = CDbI(TxtfL.Text) * The flattening Value of the Local ellipsoid
e.g clarke 1880 (modified)
```

```
i
```

```
Dim ZoneL As Double = CDbI(txtZoneL.Text) ' The Zone Value of the Local
ellipsoid e.g clarke 1880 (modified)
```

'for clarke 1880 a=6,378,249 international metres, ellipticity f is 1/293.465

a = 6378249.145 'clarke 1880 values for a as default

0.003407561379 'clarke 1880 values for f

```
Dim b As Double 'semi-minor axi*
```

```
Dim YY As Double ' the remaining terms other than BPhi
```

```

Dim eep As Double 'second eccentricity
Dim Lamda As Double 'the longitude value
Dim LamdaZero As Double 'a variable for zero longitude value
Dim K As Double
'ZoneL = 37
Phi = LatPhi * (Math.PI / 180)
Lamda = LongLamda * (Math.PI / 180) 'convert to radians
LamdaZero = (6 * ZoneL - 183) * (Math.PI / 180) 'Value converted in Radians
b = a * (1 - f) 'the semi-minor axis f is given
eep = (a ^ 2 - b ^ 2) / (b ^ 2) 'the second numeric eccentricity
eta = Math.Sqrt(eep * (Math.Cos(Phi)) ^ 2) a term used in the computation
N = (a ^ 2) / (b * Math.Sqrt(1 + eta ^ 2)) 'computes the radius of curvaturte in prime
vertical
t = Math.Tan(Phi) 'Tan(Phi)
'LamdaZero is the Longitude of the central meridian computed above
L = (Lamda - LamdaZero) 'Longitude defference already converted to radians
This section computes the remaining terms other than those containing BPhi(Arc
Length -or the footpoint Latitude)in double precicion mode
YY = (t / 2) * N * ((Math.Cos(Phi)) ^ 2) * (L ^ 2) + ((t / 24) * N * ((Math.Cos(Phi)) ^ 4))
* (5 - (t ^ 2) + 9 * (eta ^ 2) + 4 * (eta ^ 4)) * (L ^ 4) + (t / 720) * N * ((Math.Cos(Phi)) ^
6) * (61 - 58 * (t ^ 2) + (t ^ 4) + 270 * (eta ^ 2) - 330 * (t ^ 2) * (eta ^ 2)) * (L ^ 6) + (t /
40320) * N * ((Math.Cos(Phi)) ^ 8) * (1385 - 3111 * (t ^ 2) + 543 * (t ^ 4) - (t ^ 6)) * (L
^ 8)

K = 0.9996

Return K * YY ' to return the result of YY computation(constant applied to UTM
0.9996)

End Function

```

Function ComputeXX(ByVal t As Double, ByVal N As Double, ByVal Phi As Double, ByVal L As Double, ByVal eta As Double) As Double

'XX = computes X value The eastings

Dim LatPhi As Double = CDbI(TxtLatPhiL.Text) ' Extracting Latitude Value from textBox

Dim LongLamda As Double = CDbI(txtLongLamdaL.Text)' Latitude Value

Dim a As Double = CDbI(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

Dim f As Double = CDbI(TxtfL.Text)' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim ZoneL As Double = CDbI(txtZoneL.Text) ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

a = 6378249.145 "clarkel880 values for a as default

f= 0.003407561379 'clarkel880 values for f

Dim b As Double

Dim K As Double

Dim XX As Double

'Dim Zn As Double'not used substituted with zoneL

Dim eep As Double 'as a check = 0.006850116152

Dim Lamda As Double

Dim LamdaZero As Double

Lamda = LongLamda \* (Math.PI / 180)

LamdaZero = (6 \* ZoneL - 183) \* (Math.PI / 180) 'Value converted in Radians

b = a \* (1 - f) 'the semi-minor axis f is given

eep = (a<sup>2</sup> - b<sup>2</sup>) / (b<sup>2</sup>) 'computes the e<sup>2</sup>

Phi = LatPhi \* (Math.PI / 180)

t= Math.Tan(Phi) 'Tan(Phi)

eta = Math.Sqrt(eep \* (Math.Cos(Rhi))<sup>2</sup>)

LamdaZero is the Longitude of the central meridian computed above

L = (Lamda - LamdaZero) 'Longitude defference In Radians

N = (a ^ 2) / (b \* Math.Sqrt(1 + eta ^ 2)) 'computes the radius of curvatute in prime vertical

K = 0.9996

This section computes the X term and returns a value XX which is in UTM format

The computation is done on double precision mode

XX = N \* L \* Math.Cos(Pi) + (1 / 6) \* N \* ((Math.Cos(Pi)) ^ 3) \* (1 - 1 ^ 2 + eta ^ 2) \* L ^ 3 + (1 / 120) \* N \* ((Math.Cos(Pi)) ^ 5) \* (5 - 18 \* t ^ 2 + t ^ 4 + 14 \* eta ^ 2 - 58 \* t a 2 \* eta ^ 2) \* L ^ 5 + (1 / 5040) \* N \* ((Math.Cos(Pi)) ^ 7) \* (61 - 479 \* t ^ 2 + 179 \* t ^ 4 - t ^ 6) \* L ^ 7

Return K \* XX ' to return the result of XX computation after converting to UTM by multiplying by 0.9996

End Function

End Class



## APPENDIX IB

### COORDINATE TRANSFORMATION FROM GEODETIC TO UTM NORTHERN

#### LATITUDE

Public Class Form1

'For Part of Msc GIS Project By W.Abuto

•F56/P/9061/2005

•October 2007

'Programme Computes the Northings and the Eastings given the geographical coordinates and display the resulting coordinates in UTM

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'Dim LatPhi As Double ^1 = CDb1(TxtLatPhiL.Text)' Latitude Value

'Dim LongLamda As Double - CDb1(txtLongLamdaL.Text)' Latitude Value

'Dim a As Double - CDb1(TxtaL.Text)' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

'Dim f As Double ' = CDb1(TxtfL.Text)' The flattening Value of the Local ellipsoid  
e.g clarke 1880 (modified)

'Dim ZoneL As Double '= CDb1(txtZoneL.Text)" The Zone Value of the Local  
ellipsoid e.g clarke 1880 (modified)

'for clarke 1880 a=6,378,249 international metres, ellipticity f is 1/293.465

'as in the formular

Dim alpha As Double

Dim beta As Double

Dim phi As Double

Dim gamma As Double

Dim delta As Double

Dim epsilon As Double

Dim eta As Double

Dim arcLength As Double \*

Dim L As Double

```

Dim N As Double
Dim t As Double
Dim FN As String = "0" 'Term added to get False Northings
Dim FE As Double = 500000 'Term added to get False Eastings
Dim YY As Double
MsgBox(ComputeBPhi(alpha, beta, phi, gamma, delta, epsilon)) 'display of
computed foot point Latitude value(arc Length of Meridian)
arcLength = Val(TxtarcLengthL.Text) 'The BPhi value
YY = arcLength ' the other terms of y
MsgBox(ComputeYY(t, N, phi, L, eta)) 'the result of yy computation
'display as message of xx value
MsgBox(ComputeXX(t, N, phi, L, eta)) 'the result of yy computation
'function call to display the eastings value already computed and converts it to string
for diplay in the TextBox
'FE is added and all converted to string
TxtEasting.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta)) 'to display result
of True Easting to the user in a textBox
TxtEastings.Text = Convert.ToString(ComputeXX(t, N, phi, L, eta) + FE) 'to
display result of False Easting computation to the user in a textBox
TxtarcLengthL.Text = Convert.ToString(ComputeBPhi(alpha, beta, phi, gamma,
delta, epsilon) + ComputeYY(t, N, phi, L, eta)) 'to display result of arc Length
computation to the user in a textBox
'the ellipsoidal distance from the equator
'Displays the resulting total of both the Bphi value and the extra terms computed for
the Northings
txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text) + Val(FN)) The
total added values of Bphi plus YY value
'txtNorthings.Text = Convert.ToString(Val(TxtarcLengthL.Text)) 'The total added
values of Bphi plus YY value

End Sub

```

Function ComputeBPhi(ByVal alpha As Double, ByVal beta As Double, ByVal phi As Double, ByVal gamma As Double, ByVal delta As Double, ByVal epsilon As Double) As Double

Dim LatPhi As Double = Cdbl(TxtLatPhiL.Text) \* Latitude Value

Dim LongLamda As Double = Cdbl(txtLongLamdaL.Text)' Latitude Value

Dim ZoneL As Double = Cdbl(txtZoneL.Text)' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim a As Double - Cdbl(TxtaL.Text)' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

Dim f As Double - Cdbl(TxtfL.Text)' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

•Dim ZoneL As Double '= 37 '= Cdbl(txtZoneL.Text)' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

TxtaL.Text = Convert.ToString(6378249.145) 'default setting for a value

TxtfL.Text = Convert.ToString(0.003407561379) 'default setting for f value

'txtZoneL.Text = Convert.ToString(37) 'default setting for Zone value...CHANGED

Tor clarke 1880 a=6,378,249 international metres, ellipticity f is 1/293.465

a = 6378249.145 'clarke 1880 values for a as default

f= 0.003407561379 'clarke 1880 values for f

Dim b As Double

Dim K As Double

Dim BPhi As Double

Dim ns As Double

$b = a * (1 - f)$  'the semi-minor axis

$\Phi = \text{LatPhi} * (\text{Math.PI} / 180)$  'Converting Decimal degrees to Radians the  $\sim$ >phi values

$ns = (a - b) / (a + b)$  ' same as sirfcill (n) in the formula

$\alpha = (1 / 2) * (a + b) * (1 + (1 / 4) * (ns^2) + (1 / 64) * (ns^4))$

$$\text{beta} = -(3 / 2) * \text{ns} + (9 / 16) * (\text{ns}^{\wedge} 3) - (3 / 32) * (\text{ns}^{\wedge} 5)$$

$$\text{gamma} = (15 / 16) * (\text{ns}^{\wedge} 2) - (15 / 32) * (\text{ns}^{\wedge} 4)$$

$$\text{delta} = -(35 / 48) * (\text{ns}^{\wedge} 3) + (105 / 256) * (\text{ns}^{\wedge} 5)$$

$$\text{epsilon} = (315 / 512) * (\text{ns}^{\wedge} 4)$$

'LatPhi is what the user entered in decimal Degrees but the computer use only values in Radians

$$K = 0.9996$$

$$\text{BPhi} = \alpha * (\text{phi} + \text{beta} * \text{Math.Sin}(2 * \text{phi}) + \text{gamma} * \text{Math.Sin}(4 * \text{phi}) + \text{delta} * \text{Math.Sin}(6 * \text{phi}) + \text{epsilon} * \text{Math.Sin}(8 * \text{phi}))$$

Return K \* BPhi ' to return the result of this computation(scale applied to UTM 0.9996 multiply with result of Bphi )

End Function 'ComputeBPhi

Function ComputeYY(ByVal t As Double, ByVal N As Double, ByVal Phi As Double, ByVal L As Double, ByVal eta As Double) As Double

'Y = BPhi+YY so this part computes YY part, already BPhi has been computed above

Dim LatPhi As Double = CDbI(txtLatPhiL.Text)' Extracting Latitude Value from textBox

Dim LongLamda As Double = CDbI(txtLongLamdaL.Text)' Latitude Value

Dim a As Double = CDbI(txtaL.Text)<sup>1</sup> Semi-major Axis Value of local ellipsoid eg clarke 1880 (modified)

Dim f As Double = CDbI(txtfL.Text)' The flattening Value of the Local ellipsoid eg clarke 1880 (modified)

Dim ZoneL As Double = CDbI(txtZoneL.Text)" The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

'for clarke 1880 a=6,378,249 international metres, ellipticity f is 1/293.465

a = 6378249.145 'clarkel 880 values for a as default

0.003407561379 clarkel880 values for f

Dim b As Double 'semi-minor a\*is

Dim YY As Double ' the remaining terms other than BPhi

```

Dim eep As Double 'second eccentricity
Dim Lamda As Double 'the longitude value
Dim LamdaZero As Double 'a variable for zero longitude value
Dim K As Double
'ZoneL = 37
Phi = LatPhi * (Math.PI / 180)
Lamda = LongLamda * (Math.PI / 180) convert to radians
LamdaZero = (6 * ZoneL - 183) * (Math.PI / 180) 'Value converted in Radians
b = a * (1 - 0 the semi-minor axis f is given
eep = (a ^ 2 - b ^ 2) / (b ^ 2) 'the second numeric eccentricity
eta = Math.Sqrt(eep * (Math.Cos(Phi)) ^ 2) 'a term used in the computation
N = (a ^ 2) / (b * Math.Sqrt(1 + eta ^ 2)) 'computes the radius of curvatute in prime
vertical
t = Math.Tan(Phi) Tan(Phi)
'LamdaZero is the Longitude of the central meridian computed above
L = (Lamda - LamdaZero) 'Longitude defference already converted to radians
'This section computes the remaining terms other than those containing BPhi(Arc
Length -or the footpoint Latitude)in double precicion mode
YY = (t / 2) * N * ((Math.Cos(Phi)) ^ 2) * (L ^ 2) + ((t / 24) * N * ((Math.Cos(Phi))
^ 4) * (5 - (t ^ 2) + 9 * (eta ^ 2) + 4 * (eta ^ 4)) * (L ^ 4) + (t / 720) * N *
((Math.Cos(Phi)) ^ 6) * (61 - 58 * (t ^ 2) + (t ^ 4) + 270 * (eta ^ 2) - 330 * (t ^ 2) * (eta ^
2)) * (L ^ 6) + (t / 40320) * N * ((Math.Cos(Phi)) ^ 8) * (1385 - 3111 * (t ^ 2) + 543 * (t
^ 4) - (t ^ 6)) * (L ^ 8)
K = 0.9996
Return K * YY ' to return the result of YY computation(constant applied to UTM
0.9996)
End Function 'ComputeYY

Function ComputeXX(ByVal t As Double, ByVal N As Double, ByVal Phi As
Double, ByVal L As Double, ByVal eta As Double) As Double

```

'XX = computes X value The eastings

Dim LatPhi As Double = CDBl(TxtLatPhiL.Text)' Extracting Latitude Value from  
textBox

Dim LongLamda As Double = CDBl(txtLongLamdaL.Text)' Latitude Value

Dim a As Double = CDBl(TxtaL.Text) \* Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

Dim f As Double = CDBl(TxtfL.Text)' The flattening Value of the Local ellipsoid  
e.g clarke 1880 (modified)

Dim ZoneL As Double = CDBl(txtZoneL.Text)' The Zone Value of the Local  
ellipsoid e.g clarke 1880 (modified)

a = 6378249.145 'clarkel880 values for a as default

f = 0.003407561379 'clarkel880 values for f

Dim b As Double

Dim K As Double

Dim XX As Double

'Dim Zn As Double'not used substituted with zoneL

Dim eep As Double 'as a check = 0.006850116152

Dim Lamda As Double

Dim LamdaZero As Double

Lamda = LongLamda \* (Math.PI / 180)

LamdaZero = (6 \* ZoneL - 183) \* (Math.PI / 180) 'Value converted in Radians

b = a \* (1 - f) 'the semi-minor axis f is given

eep = (a ^ 2 - b ^ 2) / (b ^ 2) "computes the e<sup>2</sup>

Phi = LatPhi \* (Math.PI/180)

t = Math.Tan(Phi) 'Tan(Phi)

eta = Math.Sqrt(eep \* (Math.Cos(Phi)) ^ 2)

'LamdaZero is the Longitude of the central meridian computed above

L = (Lamda - LamdaZero) 'Longitude defference In Radians

N = (a ^ 2) / (b \* Math.Sqrt(1 + eta ^ 2)) 'computes the radius of curvatute in prime

vertical

K = 0.9996

This section computes the X term and returns a value XX which is in UTM format

The computation is done on double precision mode

$$\begin{aligned} XX = & N * L * \text{Math.Cos}(\text{Phi}) + (1 / 6) * N * ((\text{Math.Cos}(\text{Phi}))^3) * (1 - t^2 + \eta \\ & a^2) * L^3 + (1 / 120) * N * ((\text{Math.Cos}(\text{Phi}))^5) * (5 - 18 * t^2 + t^4 + 14 * \eta^2 \\ & . 58 * t^2 * \eta^2) * L^5 + (1 / 5040) * N * ((\text{Math.Cos}(\text{Phi}))^7) * (61 - 479 * t^2 \\ & + 179 * t^4 - t^6) * L^7 \end{aligned}$$

Return K \* XX to return the result of XX computation after converting to UTM by multiplying by 0.9996

End Function 'ComputeXX

End Class

## APPENDIX 2A

### COORDINATE TRANSFORMATION FROM UTM TO GEODETIC SOUTHERN LATITUDE

**Public** Class Fornil

' Coordinate Transformation from UTM To Geodetic FOR POINTS SOUTH OF  
EQUATOR

'For Part of Msc GIS Project By. W Abuto

'F56/P/9061/2005

•October 2007

'Programme Computes the geographical coordinates

'Using UTM Values and displays the results in Latitude and Longitude

'This is the reverse computation and can serve

'as a check for Geodetic to UTM computation.

' Include higher number of decimal places if available

' for this check on reversability

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click

i

TxtZoneL.Text = "37"

Dim Northings As Double = Cdbl(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = Cdbl(TxtEastings.Text)' Easting Value

Dim ZoneL As Double = Cdbl(TxtZoneL.Text)' zone

' Dim a As Double '= Cdbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

Dim aphif As Double

Dim yb As Double

Dim betab As Double \*

Dim gammab As Double



## APPENDIX 2A

### COORDINATE TRANSFORMATION FROM UTM TO GEODETIC SOUTHERN LATITUDE

public Class Form1

' Coordinate Transformation from UTM To Geodetic FOR POINTS SOUTH OF  
EQUATOR

'For Part of Msc GIS Project By. W Abuto

'F56/P/9061/2005

•October 2007

'Programme Computes the geographical coordinates

'Using UTM Values and displays the results in Latitude and Longitude

'This is the reverse computation and can serve

'as a check for Geodetic to UTM computation.

' Include higher number of decimal places if available

' for this check on reversability

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click

i

TxtZoneL.Text = "37"

Dim Northings As Double = Cdbl(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = Cdbl(TxtEastings.Text)' Easting Value

Dim ZoneL As Double = Cdbl(TxtZoneL.Text)' zone

' Dim a As Double '= Cdbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

Dim aphif As Double

Dim yb As Double

Dim betab As Double \*

Dim gammab As Double

Dim deltab As Double

Dim epsilonb As Double

Dim etaf As Double

Dim X As Double

Dim Nf As Double

Dim tf As Double

Dim lamda As Double

Dim LamdaZero As Double

Dim PhifAs Double

MsgBox(ComputeBPhif(yb, betab, gammab, deltab, epsilonb)) ' In value in degrees

aphif = (ComputeBPhif(yb, betab, gammab, deltab, epsilonb)) 'use output of the  
function computeBphif

'and stores it in the variable aphif in degrees

Txtphif.Text = Convert.ToString(ComputeBPhif(yb, betab, gammab, deltab, epsilonb))

This is a sting equivalent value of phif in degrees

'which has to be converted to display in the TextBox "txtPhif"

'display of computed foot point Latitude value(arc Length of Meridian)

MsgBox(ComputeYYf(tf, Nf, X, etaf)) 'From yyf computation in degrees

TxtLatPhiL.Text = Convert.ToString(ComputeYYf(tf, Nf, X, etaf) + aphif) 'adding  
result of footpoint Latitude to the extra term aphif

alamda = MsgBox(ComputeXXf(LamdaZero, Nf, Phif, X, tf, etaf))

MsgBox(ComputeXXf(LamdaZero, Nf, Phif, X, tf, etaf))'

TxtLongLamdaL.Text = Convert.ToString(ComputeXXf(LamdaZero, Nf, Phif, X,  
tf, etaf)) 'to display the resulting value

'in decimal degrees from the function "computeXXf()"

End Sub

Function ComputeBPhif(ByVal yb As Double, ByVal betab As Double, ByVal  
gammab As Double, ByVal deltab As Double, ByVal epsilonb As Double) As Double

Dim Eastings As Double = CDbI(TxtEastings.Text)' Easting Value

TxtfL.Text = "0.003407561379 " 'default value of f , user to change value if f is  
different

Dim f As Double = CDbI(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g  
clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is  
different

Dim a As Double = CDbI(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

TxtZoneL.Text = "37" ' The Zone Value of the Local ellipsoid e.g clarke 1880  
(modified) set as default value

Dim ZoneL As Double = CDbI(TxtZoneL.Text) 'The code above sets the default  
value of 37 user should edit it if zone is 36

Dim Northings As Double = CDbI(TxtNorthings.Text) The Northings in the  
TextBox

'Dim Eastings As Double = CDbI(TxtEastings.Text) 'The Eastings in the TextBox  
'Adopted parameters —> clarke 1880 a=6,378,249 international metres, ellipticity f is  
1/293.465

Dim b As Double

Dim y As Double

'Dim E As Double

Dim BPhif As Double

Dim alphab As Double

Dim ns As Double

y «(1 / 0.9996) \* (Northings - 10000000)' Scaling up

= Eastings - 500000)' Points to the west do not require subtraction

o = a \* (1 - f) 'the semi-minor a\*is

$ns = (a - b) / (a + b)$  ' same as small (n) in the formula

$$\text{alphab} = (1 / 2) * (a + b) * (1 + (1 / 4) * (\text{ns}^{\wedge} 2) + (1 / 64) * (\text{ns}^{\wedge} 4))$$

yb = (y / alphab) Y is a variable representing the Northings value keyed in by user

$$\text{betab} = (3 / 2) * \text{ns} - (27 / 32) * (\text{ns}^{\wedge} 3) + (269 / 512) * \text{ns}^{\wedge} 5$$

$$\text{gammab} = (21 / 16) * (\text{ns}^{\wedge} 2) - (55 / 32) * \text{ns}^{\wedge} 4$$

$$\text{deltab} = (151 / 96) * (\text{ns}^{\wedge} 3) - (417 / 128) * \text{ns}^{\wedge} 5$$

$$\text{epsilonb} = (1097 / 512) * \text{ns}^{\wedge} 4$$

<sup>1</sup> phi = LatPhi \* (Math.PI / 180) 'Converting Decimal degrees to Radians the ->phi values

' LatPhi is what the user entered in decimal Degrees but the computer use only values in Radians

$$\text{BPhif} = \text{yb} + \text{betab} * (\text{Math.Sin}(2 * \text{yb})) + \text{gammab} * (\text{Math.Sin}(4 * \text{yb})) + \text{deltab} * (\text{Math.Sin}(6 * \text{yb})) + \text{epsilonb} * (\text{Math.Sin}(8 * \text{yb}))$$

Return BPhif\* (180 / Math.PI) to return the result in degrees

End Function ' ComputeBPhif()



```
alphanb = (1 / 2) * (a + b) * (1 + (1 / 4) * (ns ^ 2) + (1 / 64) * (ns ^ 4))
```

```
yb = (y / alphanb) 'Y is a variable representing the Northings value keyed in by user
```

```
betanb = (3 / 2) * ns - (27 / 32) * (ns ^ 3) + (269 / 512) * ns ^ 5
```

```
gammab = (21 / 16) * (ns ^ 2) - (55 / 32) * ns ^ 4
```

```
deltanb = (151 / 96) * (ns ^ 3) - (417 / 128) * ns ^ 5
```

```
epsilonb = (1097/512) * ns ^ 4
```

```
' phi = LatPhi * (Math.PI / 180) 'Converting Decimal degrees to Radians the ~>phi  
values
```

```
' LatPhi is what the user entered in decimal Degrees but the computer use only  
values in Radians
```

```
BPhif= yb + betanb * (Math.Sin(2 * yb)) + gammab * (Math.Sin(4 * yb)) + deltanb *  
(Math.Sin(6 * yb)) + epsilonb * (Math.Sin(8 * yb))
```

```
Return BPhif* (180 / Math.PI) 'to return the result in degrees
```

```
End Function ' ComputeBPhif()
```

```

alphab = (1 / 2) * (a + b) * (1 + (1 / 4) * (ns ^ 2) + (1 / 64) * (ns ^ 4))
yb = (y / alphab)' V is a variable representing the Northings value keyed in by user
betab = (3 / 2) * ns - (27 / 32) * (ns ^ 3) + (269 / 512) * ns ^ 5
gammab = (21 / 16) * (ns ^ 2) - (55 / 32) * ns ^ 4
deltab = (151 / 96) * (ns ^ 3) - (417 / 128) * ns ^ 5
epsilonb = (1097/512) * ns ^4
' phi = LatPhi * (Math.PI / 180) 'Converting Decimal degrees to Radians the ->phi
values
' LatPhi is what the user entered in decimal Degrees but the computer use only
values in Radians
BPhif= yb + betab * (Math.Sin(2 * yb)) + gammab * (Math.Sin(4 * yb)) + deltab *
(Math.Sin(6 * yb)) + epsilonb * (Math.Sin(8 * yb))
Return BPhif* (180 / Math.PI) 'to return the result in degrees
End Function' ComputeBPhif()

```

Function ComputeYYf(ByVal tf As Double, ByVal Nf As Double, ByVal X As Double, ByVal etaf As Double) As Double

\*Y = BPhi+YY so this part computes YY part, already BPhi has been computed above

'Dim LatPhi As Double = Cdbl(TxtLatPhiL.Text) ' Extracting Latitude Value from textBox

' Dim LongLamda As Double = Cdbl(txtLongLamdaL.Text)' Latitude Value

Dim Northings As Double = Cdbl(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = Cdbl(TxtEastings.Text)' Easting Value

Dim YYf As Double ' the remaining terms other than BPhi

TxtfL.Text = "0.003407561379 " 'default value of f, user to change value if f is different

Dim f As Double = Cdbl(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is different

Dim a As Double = Cdbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

'TxtZoneL.Text = "37" ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim ZoneL As Double = Cdbl(TxtZoneL.Text) The code above sets the default value of 37 user should edit it if zone is 36

'TxtaL.Text = Convert.ToString(6378249.145) 'default setting for a value

'TxtfL.Text = Convert.ToString(0.003407561379) 'default setting for f value

TxtZoneL.Text = Convert.ToString(37) 'default setting for Zone value

' Dim a As Double = Cdbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

'Dim f As Double = Cdbl(TxtfL.Text)' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

'a = 6378249.145 'clarkel880 values for a as default and to prevent data entry ^stakes

' f= 0.003407561379 'clarke 1880 values deliberately included within the program to prevent entry errors

' for use in other areas , specific local ellipsoid parameters will have to be substituted  
 $X = (1 / 0.9996) * (\text{Eastings} - 500000)$  'for southern latitude and west of central meridian

Dim b As Double 'semi-minor axis

Dim Phif As Double

Dim eep As Double 'second eccentricity

' The value of Txtphif.Text is in decimal degrees cPhif is a variable to store this value

Dim CPhif As Double = CDbl(Txtphif.Text) 'retrieval from the TextBox after being computed by the function ComputeBphif() above

' set latitude to negative value because point is south of equator before being used in the computation of the extra term

Phif = CPhif \* (Math.PI) / 180 ' To convert Phif to Radians since Computation is done in Radians and point is south of equator

b = a \* (1 - f) 'the semi-minor axis f is given

$eep = (a^2 - b^2) / (b^2)$

etaf = Math.Sqrt(eep \* (Math.Cos(Phif))^2) 'a term used in the computation

tf = Math.Tan(Phif) 'Tan(Phi)

This section computes the remaining terms other than those containing BPhi(Arc Length -or the footpoint Latitude)in double precision mode

$Nf = (a^2) / b * (\text{Math.Sqrt}(1 + \text{etaf}^2))$  'Also this can be used- $\rightarrow (a^2) / (\text{Math.Sqrt}(a^2 * (\text{Math.Cos}(\text{Phif}))^2 + (b^2) * (\text{Math.Sin}(\text{Phif}))^2))$  'computes the radius of curvatures in prime vertical '

$YYf = (tf / (2 * Nf^2)) * (-1 - (\text{etaf}^2)) * (X^2) + (tf / (24 * (Nf^4))) * (5 + 3 * (tf^2) + 6 * (\text{etaf}^2) - 6 * (tf^2) * (\text{etaf}^2) - 3 * (\text{etaf}^4) - 9 * (tf^2) * (\text{etaf}^4)) * (X^4) + (tf / (720 * (Nf^6))) * (-61 - 90 * (tf^2) - 45 * (tf^4) - 107 * (\text{etaf}^2) + 162 * (tf^2) * (\text{etaf}^2) + 45 * (tf^4) * (\text{etaf}^2)) * (X^6) + (tf / (40320 * Nf^8)) * (1385 + 3633 * (tf^2) + 4095 * (tf^4) + 1575 * (tf^6)) * (X^8)$

Return YYf \* (180 / (Math.PI))' The Value of the Extra term YYf converted to Decimal  
I Agrees.

I<sup>end</sup> Function \* end of function ->ComputeYYf()



Function ComputeXXf(ByVal LamdaZero As Double, ByVal Nf As Double, ByVal Phif As Double, ByVal X As Double, ByVal tf As Double, ByVal etaf As Double) As Double

Dim Northings As Double = CDb1(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = CDb1(TxtEastings.Text)' Easting Value

'the suscript f is added to mean angle phif has been used—the footpoint Latitude value as computed by function "ComputeBPhif()<sup>M</sup>

'TxtZoneL.Text = Convert.ToString(37) 'default setting forZone value

'Dim a As Double = CDb1(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

'Dim f As Double = CDb1(TxtfL.Text)' The flattening Value of the Local ellipsoid e.g clarke 1880 (modified)

i

TxtfL.Text = "0.003407561379 " 'default value of f, user to change value if f is different

Dim f As Double = CDb1(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is different

Dim a As Double = CDb1(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

TxtZoneL.Text = "37" ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim ZoneL As Double = CDb1(TxtZoneL.Text) 'The code above sets the default value of 37 user should edit it if zone is 36

Dim CPhif As Double

CPhif = CDb1(Txtphif.Text) ' To store the contents of the TextBox information in the variabl

\* = (1 / 0.9996) \* (Eastings - 500000) 'for southern latitude and East of central Pridian .\*

Dim b As Double 'the semi-minor axis of the ellipsoid used

Dim XXf As Double 'A variable to hold computed Longitude value to be computed below

Dim eep As Double 'the second numeric eccentricity

' Dim Lamda As Double ' a variable for Longitude

'Lamda = LongLamda \* (Math.PI / 180) 'to convert to Radians since value keyed in by the user is in degrees

LamdaZero = (6 \* ZoneL - 183) \* (Math.PI / 180) 'Value converted to Radians

b = a \* (1 - f) 'the semi-minor axis f is given

eep = (a<sup>2</sup> - b<sup>2</sup>) / (b<sup>2</sup>)

Phif = CPhif \* (Math.PI / 180) ' to convert to Radians the cphif value computed by function "ComputeBPhif()<sup>M</sup> since trigonometric computations that follow are required in Radians

etaf = Math.Sqrt(eep \* (Math.Cos(Phif))<sup>2</sup>) 'auxiliary quantity

tf = Math.Tan(Phif) 'Tan(Phi)

Nf = (a<sup>2</sup>) / (Math.Sqrt(a<sup>2</sup> \* (Math.Cos(Phif))<sup>2</sup> + (b<sup>2</sup>) \* (Math.Sin(Phif))<sup>2</sup>))

'computes the radius of curvatute in prime vertical

'This section computes the Lamda Value and returns a value xxf in decimal degrees

XXf = LamdaZero + (X / (Nf \* (Math.Cos(Phif)))) + (1 / ((6 \* Nf<sup>3</sup>) \* (Math.Cos(Phif)))) \* (-1 - 2 \* tf<sup>2</sup> - etaf<sup>2</sup>) \* X<sup>3</sup> + (1 / (120 \* Nf<sup>5</sup> \* Math.Cos(Phif))) \* (5 + 28 \* tf<sup>2</sup> + 24 \* tf<sup>4</sup> + 6 \* etaf<sup>2</sup> + 8 \* tf<sup>2</sup> \* etaf<sup>2</sup>) \* (X<sup>5</sup>) + (1 / (5040 \* Nf<sup>7</sup> \* (Math.Cos(Phif)))) \* (-61 - 662 \* tf<sup>2</sup> - (1320 \* (tf<sup>4</sup>)) - 720 \* tf<sup>6</sup>) \* X<sup>7</sup> 'Computes the Longitude (Lamda)

Return XXf \* (180 / (Math.PI)) 'to return the result of XXf computation after converting to degrees by multiplying by 180/PI

End Function 'end of function ComputeXXf()

End Class

**APPENDIX 2B**  
**COORDINATE TRANSFORMATION FROM UTM TO GEODETIC**  
**(FOR POINTS NORTH OF EQUATOR)**

Public Class Form1

' Coordinate Transformation from UTM To Geodetic FOR POINTS NORTH OF EQUATOR

'For Part of Msc GIS Project By. W Abuto

•F56/P/9061/2005

•October 2007

'Programme Computes the geographical coordinates

'Using UTM Values and displays the results in Latitude and Longitude

'This is the reverse computation and can serve

'as a check for Geodetic to UTM computation.

' Include higher number of decimal places if available

' for this check on reversability

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

**i**

'TxtZoneL.Text = "37"

Dim Northings As Double = CDb1(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = CDb1(TxtEastings.Text)' Easting Value

Dim ZoneL As Double = CDb1(TxtZoneL.Text)' zone

' Dim a As Double = CDb1(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e-g clarke 1880 (modified)

Dim aphif As Double

Dim yb As Double

Dim betab As Double \*

Dim gammab As Double

```

Dim deltab As Double
Dim epsilonb As Double
Dim etaf As Double
Dim X As Double
Dim Nf As Double
Dim tf As Double
Dim lamda As Double
Dim LamdaZero As Double
Dim Phif As Double
MsgBox(ComputeBPhif(yb, betab, gammab, deltab, epsilonb)) ' In value in degrees
aphif = (ComputeBPhif(yb, betab, gammab, deltab, epsilonb)) 'variable aphif retrieves
'and stores the content of this function which is in degrees
Txtphif.Text = Convert.ToString(ComputeBPhif(yb, betab, gammab, deltab,
epsilonb)) 'This value of phif in degrees
'is displayed in the TextBox "txtPhif"
'display of computed foot point Latitude value(arc Length of Meridian)
MsgBox(ComputeYYf(tf, Nf, X, etaf)) 'From yyf computation in degrees
TxtLatPhiL.Text = Convert.ToString(ComputeYYf(tf, Nf, X, etaf) + aphif) 'adding
result of footpoint Latitude to the extra term aphif
alamda = MsgBox(ComputeXXf(LamdaZero, Nf, Phif, X, tf, etaf))
MsgBox(ComputeXXf(LamdaZero, Nf, Phif, X, tf, etaf))'
TxtLongLamdaL.Text = Convert.ToString(ComputeXXf(LamdaZero, Nf, Phif, X,
tf, etaf)) 'to display the resulting value
'in decimal degrees from the function "computeXXf()"
End Sub

```

Function ComputeBPhif(ByVal yb As Double, ByVal betab As Double, ByVal  
gammab As Double, ByVal deltab As Double, ByVal epsilonb As Double) As Double

Dim Eastings As Double = CDbl(TxtEastings.Text)' Easting Value

TxtfL.Text = "0.003407561379 " 'default value of f, user to change value if f is  
different

Dim f As Double = CDbl(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g  
clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is  
different

Dim a As Double = CDbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

Dim ZoneL As Double = CDbl(TxtZoneL.Text) The code above sets the default  
value of 37 user should edit it if zone is 36

i

Dim Northings As Double = CDbl(TxtNorthings.Text)' The Northings in the TextBox

Dim b As Double

Dim y As Double

Dim ns As Double

Dim K As Double

Dim BPhif As Double

K = 0.9996

y = (1 / K) \* (Northings)' REVERSE COMPUTATION TO HANDLE SCALE

i

b = a \* (1 - f) 'the semi-minor axis

ns = (a - b) / (a + b)' same as small (n) in the formula

Dim alphab As Double = (1 / 2) \* (a + b) \* (1 + (1 / 4) \* (ns ^ 2) + (1 / 64) \* (ns ^ 4))

yb = (y / alphab) 'Y is a variable representing the Northings value keyed in by user  
"y" divided by alphab

$$\text{betab} = (3 / 2) * \text{ns} - (27 / 32) * (\text{ns}^{\wedge} 3) + (269 / 512) * (\text{ns}^{\wedge} 5)$$

$$\text{gammab} = (21 / 16) * (\text{ns}^{\wedge} 2) - (55 / 32) * (\text{ns}^{\wedge} 4)$$

$$\text{deltab} = (151 / 96) * (\text{ns}^{\wedge} 3) - (417 / 128) * (\text{ns}^{\wedge} 5)$$

$$\text{epsilonb} = (1097 / 512) * (\text{ns}^{\wedge} 4)$$

' phi = LatPhi \* (Math.PI / 180) 'Converting Decimal degrees to Radians the —>phi  
values

' LatPhi is what the user entered in decimal Degrees but the computer use only  
values in Radians

$$\text{BPhif} = \text{yb} + \text{betab} * (\text{Math.Sin}(2 * \text{yb})) + \text{gammab} * \text{Math.Sin}(4 * \text{yb}) + \text{deltab} * \text{Math.Sin}(6 * \text{yb}) + \text{epsilonb} * (\text{Math.Sin}(8 * \text{yb}))$$

Return BPhif\* (180 / Math.PI) 'to return the result in degrees

End Function <sup>1</sup> ComputeBPhifO

Function ComputeYYf(ByVal tf As Double, ByVal Nf As Double, ByVal X As  
Double, ByVal etaf As Double) As Double

'Y = BPhi+YY so this part computes YY part, already BPhi has been computed  
above

Dim Northings As Double = CDbl(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = CDbl(TxtEastings.Text)' Easting Value

Dim YYf As Double ' the remaining terms other than BPhi

Dim K As Double

K = 0.9996

TxtfL.Text = "0.003407561379 " 'default value of f , user to change value if f is  
different

Dim f As Double = CDbl(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g  
clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is different

Dim a As Double = Cdbl(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid e.g clarke 1880 (modified)

'TxtZoneL.Text = "37" ' The Zone Value of the Local ellipsoid e.g clarke 1880 (modified)

Dim ZoneL As Double = Cdbl(TxtZoneL.Text) 'The code above sets the default value of 37 user should edit it if zone is 36

X = (1 / K) \* (Easting - 500000) 'for southern latitude and west of central meridian

Dim b As Double 'semi-minor axis

Dim PhifAs Double

' The value of Txtphif.Text is in decimal degrees cPhif is a variable to store this value

Dim CPhif As Double = Cdbl(Txtphif.Text) 'retrieval from the TextBox after being computed by the function ComputeBphifQ above

Phif = CPhif \* (Math.PI) / 180 \* To convert Phif to Radians since Computation is done in Radians and point is south

Dim eep As Double 'second eccentricity

b = a \* (1 - f) 'the semi-minor axis f is given

eep = (a ^ 2 - b ^ 2) / (b ^ 2)

etaf = Math.Sqrt(eep \* (Math.Cos(Phif)) ^ 2) 'a term used in the computation

tf = Math.Tan(Phif) 'Tan(Phi)

'This section computes the remaining terms other than those containing BPhi(Arc Length -or the footpoint Latitude)in double precision mode

Nf = (a ^ 2) / b \* (Math.Sqrt(1 + etaf ^ 2)) 'Also this can be used->(a ^ 2) / (Math.Sqrt(a ^ 2 \* (Math.Cos(Phif)) ^ 2 + (b ^ 2) \* (Math.Sin(Phif)) ^ 2)) 'computes the radius of curvature in prime vertical '

YYf = (tf / (2 \* Nf ^ 2)) \* (-1 - (etaf ^ 2)) \* (X ^ 2) + (tf / (24 \* (Nf ^ 4))) \* (5 + 3 \* (tf ^ 2) + 6 \* (etaf ^ 2) - 6 \* (tf ^ 2) \* (etaf ^ 2) - 3 \* (etaf ^ 4) - 9 \* (tf ^ 2) \* (etaf ^ 4)) \* (X ^ 4) + (tf / (720 \* (Nf ^ 6))) \* (-61 - 90 \* (tf ^ 2) - 45 \* (tf ^ 4) - 107 \* (etaf ^ 2) + 162 \* (tf ^ 2) \* (etaf ^ 2) + 45 \* (tf ^ 4) \* (etaf ^ 2)) \* (X ^ 6) + (tf / (40320 \* Nf ^ 8)) \* (1385 + 3633 \* (tf ^ 2) + 4095 \* (tf ^ 4) + 1575 \* (tf ^ 6)) \* (X ^ 8)

Return  $YYf * (180 / (\text{Math.PI}))$  The Value of the Extra term  $YYf$  converted to  
Decimal degrees minus

End Function ' end of function —>ComputeYYf()

Function ComputeXXf(ByVal LamdaZero As Double, ByVal Nf As Double, ByVal  
Phif As Double, ByVal X As Double, ByVal tf As Double, ByVal etaf As Double) As  
Double

Dim Northings As Double = CDb1(TxtNorthings.Text)' Northing Value

Dim Eastings As Double = CDb1(TxtEastings.Text)' Easting Value

'the suscript f is added to mean angle phif has been used—the footpoint Latitude value as  
computed by function "ComputeBPhif()"

'Dim f As Double '= CDb1(TxtfL.Text)' The flattening Value of the Local ellipsoid  
e.g clarke 1880 (modified)

TxtfL.Text = "0.003407561379 " 'default value of f, user to change value if f is  
different

Dim f As Double = CDb1(TxtfL.Text)' Semi-major Axis Value of local ellipsoid e.g  
clarke 1880 (modified)

TxtaL.Text = "6378249.145" 'default value of a , user to change value if a is  
different

Dim a As Double = CDb1(TxtaL.Text) ' Semi-major Axis Value of local ellipsoid  
e.g clarke 1880 (modified)

'TxtZoneL.Text = "37" \* The Zone Value of the Local ellipsoid e.g clarke 1880  
(modified)

Dim ZoneL As Double = CDb1(TxtZoneL.Text) ' I he code above sets the default  
value of 37 user should edit it if zone is 36

Dim CPhif As Double

CPhif = CDb1(Txtphif.Text) ' To store the contents of the TextBox information in  
the variable



X = (1 / 0.9996) \* (Eastings - 500000) 'for southern latitude and west of c e n ^ ^ l  
meridian

Dim b As Double 'the semi-minor axis of the ellipsoid used

Dim XXf As Double 'A variable to hold computed Longitude value to be c o m p u ^ ^ d  
below

Dim eep As Double 'the second numeric eccentricity

' Dim Lamda As Double ' a variable for Longitude

'Lamda = LongLamda \* (Math.PI / 180) 'to convert to Radians since value keyecl i xi  
by the user is in degrees

LamdaZero = (6 \* ZoneL - 183) \* (Math.PI / 180) 'Value converted to Radians

b = a \* (1 - 0 the semi-minor axis f is given

eep = (a ^ 2 - b ^ 2) / (b ^ 2) 'second numerical eccentricity value

Phif = CPhif \* (Math.PI / 180) ' to convert to Radians the cphif value computed ^ >  
function "ComputeBPhifQ" since trigonometric computations that follow are require ^ - n  
Radians

etaf = Math.Sqrt(eep \* (Math.Cos(Phif)) ^ 2) 'auxiliary quantity

' Dim Nfh As Double

tf = Math.Tan(Phif) Tan(Phi)

Nf = (a ^ 2) / (Math.Sqrt(a ^ 2 \* (Math.Cos(Phif)) ^ 2 + (b ^ 2) \* (Math.Sin(Phif^  
2))) 'computes the radius of curvatute in prime vertical

'This section computes the Lamda Value and returns a value xxf in decimal deg r c ^ ^ ^

XXf = LamdaZero + (X / (Nf \* Math.Cos(Phif))) + (1 / ((6 \* Nf ^ 3}  
Math.Cos(Phif))) \* (-1 - 2 \* tf ^ 2 - etaf ^ 2) \* X ^ 3 + (1 / (120 \* Nf ^ \$  
Math.Cos(Phif))) \* (5 + 28 \* tf ^ 2 + 24 \* tf ^ 4 + 6 \* etaf ^ 2 + 8 \* tf ^ 2 \* etaf ^ 2) \*  
^5) + (1 / (5040 \* Nf ^ 7 \* (Math.Cos(Phif)))) \* (-61 - 662 \* tf ^ 2 - (1320 \* (tf ^ ^ -  
720 \* tf ^ 6) \* X ^ 7 'Computes the Longitude (Lamda)

Return XXf \* (180 / (Math.PI)) 'to return the result of XXf computation a j ^ e r  
converting to degrees by multiplying by 180/PI

End Function 'end of function ComputeXXf()

i

End Class

Dim XXf As Double 'A variable to hold computed Longitude value to be computed below

Dim eep As Double 'the second numeric eccentricity

' Dim Lamda As Double ' a variable for Longitude

'Lamda = LongLamda \* (Math.PI / 180) 'to convert to Radians since value keyed in by the user is in degrees

LamdaZero = (6 \* ZoneL - 183) \* (Math.PI / 180) 'Value converted to Radians

b = a \* (1 - 0 the semi-minor axis f is given

eep = (a ^ 2 - b ^ 2) / (b ^ 2)

Phif = CPhif \* (Math.PI / 180) ' to convert to Radians the cphif value computed by function "ComputeBPhifO" since trigonometric computations that follow are required in Radians

etaf = Math.Sqrt(eep \* (Math.Cos(Phif)) ^ 2) 'auxiliary quantity

tf = Math.Tan(Phif) 'Tan(Phi)

Nf = (a ^ 2) / (Math.Sqrt(a ^ 2 \* (Math.Cos(Phif)) ^ 2 + (b ^ 2) \* (Math.Sin(Phif)) ^ 2))

'computes the radius of curvatute in prime vertical

This section computes the Lamda Value and returns a value xxf in decimal degrees

XXf = LamdaZero + (X / (Nf \* (Math.Cos(Phif)))) + (1 / ((6 \* Nf ^ 3) \* (Math.Cos(Phif)))) \* (-1 - 2 \* t f ^ 2 - etaf ^ 2) \* X ^ 3 + (1 / (120 \* Nf ^ 5 \* Math.Cos(Phif))) \* (5 + 28 \* tf ^ 2 + 24 \* tf ^ 4 + 6 \* etaf ^ 2 + 8 \* tf ^ 2 \* etaf ^ 2) \* (X ^ 5) + (1 / (5040 \* Nf ^ 7 \* (Math.Cos(Phif)))) \* (-61 - 662 \* tf ^ 2 - (1320 \* (tf ^ 4)) - 720 \* tf ^ 6) \* X ^ 7 'Computes the Longitude (Lamda)

Return XXf \* (180 / (Math.PI)) 'to return the result of XXf computation after converting to degrees by multiplying by 180/PI

i

End Function 'end of function ComputeXXfQ

End Class

### APPENDIX 3

#### COORDINATE TRANSFORMATION FROM GEOCENTRIC TO GEODETIC

Public Class Form1

'Programme designed to convert coordinates from Geocentric to Geodetic

'targetting GPS data available in Cartesian coordinate system and required in Geodetic

Phi.Labda.h

'DSN: by Washington Abuto

•F19/P/9061/2005

'For MSC GIS project

'14th October 2007

'In Visual Basic 2005

'It is necessary to include the parameters of the Ellipsoid the semi-major axis and the flattening

' from there other parameters are computed including b value and eccentricity

'Cdbl() transforms argument to double precision

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim X As Double = Cdbl(txtXcoord.Text) 'txtXcoord is the X Coordinates keyed by the user as input

Dim Y As Double = Cdbl(txtYcoord.Text) 'txtYcoord is is the Y coordinates keyed by the user as input

Dim Z As Double = Cdbl(txtZcoord.Text) 'txtZcoord is is the Z coordinates keyed by the user as input

Dim a As Double = Cdbl(txtA.Text) 'txtA is is the a Value ->semimajor axis keyed in by the user as input

'Dim h As Double 'Note: x,y,z,a,f are given values

Dim f As Double = Cdbl(txtF.Text) 'txtf is is the f Value ->flattening keyed in by the user as input

Dim N As Double

Dim phi As Double

Dim \* eep As Double

Dim b As Double

Dim p As Double

'Dim theta As Double

Dim ee As Double

Dim sinthetaqb As Double

Dim costhetaqb As Double

MsgBox(CalculateNeuw(a, phi, f)) ' for the display of results of calculated radius of prime vertical

MsgBox(Calculatephi(Z, b, eep, sinthetaqb, p, a, ee, costhetaqb)) ' for display of results of calculated Latitude

MsgBox(CalculateLabda(X, Y))' for display of results of calculated Longitude

MsgBox(CalculateHt(p, phi, N))' for display of results of calculated height

' This section is concerned with display of results to the textboxes

txtPhi.Text = Convert.ToString(Calculatephi(Z, b, eep, sinthetaqb, p, a, ee, costhetaqb))

eep means  $e^2$ , eep and ee means  $e^2$ ,

'theta is angle in radians, p is computed from  $p = \sqrt{x^2 + y^2}$

txtLabda.Text = Convert.ToString(CalculateLabda(X, Y)) 'Labda is the required Longitude , eep means  $e^2$ , eep and ee means  $e^2$ ,

'theta is angle in radians, p is computed from  $p = \sqrt{x^2 + y^2}$

txtHt.Text = Convert.ToString(CalculateHt(p, phi, N)) ' Ht is the height required  $h = (p/\cos(\phi) - N)$

'eep means  $e^2$ , eep and ee means  $e^2$ , theta , and phi are angles in radians, p is computed from  $p = \sqrt{x^2 + y^2}$

End Sub

Function calculateNeuw(ByVal a As Double, ByVal phi As Double, ByVal b As Double) As Double

'Dim txtNeuw As Double = CDbl(txtNeuw.Text)' radius of curvature

^ote: x,y,z,a,f are given values

Dim aa As Double = CDbl(txtA.Text) 'declaring a again cannot work because

. is declared in the function above

Z As Double = CDbl(txtZcoord.Text)' Are values keyed in -the geocentric

'coordinates in cartesian system

Dim X As Double = CDBl(txtXcoord.Text)

Dim Y As Double = CDBl(txtYcoord.Text)

Dim ff As Double = CDBl(txtF.Text)' assigning variable ff to hold value in TextBox

Dim f As Double

• for WGS84 a=6378137 metres,b= 6356752.3142metres,f=1-b/a = 0.003352810672

a = aa

f= ff

b = a \* (1 - 0 'and f= 1 - b / a , e = sqrt( 1-(b<sup>2</sup>/a<sup>2</sup>))

Dim p As Double = Math.Sqrt(X<sup>2</sup> + Y<sup>2</sup>)

'Dim f As Double

Dim theta As Double = Math.Atan((Z \* a) / (p \* b))' is this theta in radians?

•Return theta ~>for check computation

Dim eep As Double = ((a<sup>2</sup>) - (b<sup>2</sup>)) / (b<sup>2</sup>)

Dim ee As Double = ((a<sup>2</sup>) - (b<sup>2</sup>)) / (a<sup>2</sup>)

Dim sinthetaqb As Double 'means --> sin<sup>3</sup>(theta)

Dim Costhetaqb As Double 'means —> cos<sup>3</sup>(theta)

sinthetaqb = (Math.Sin(theta))<sup>3</sup>

'check Return sinthetaqb

Costhetaqb = (Math.Cos(theta))<sup>3</sup>

'check by Return value of Costhetaqb

phi = Math.Atan((Z + b \* eep \* sinthetaqb) / (p - a \* ee \* Costhetaqb))

'Check by Return value of Costhetaqb

Dim N As Double 'N is the Radius of Curvature in the Prime Vertical

N = (a<sup>2</sup>) / (Math.Sqrt((a<sup>2</sup>) \* (Math.Cos(phi))<sup>2</sup> + (b<sup>2</sup>) \* (Math.Sin(phi))<sup>2</sup>))

Return N

End Function

Function Calculatephi(ByVal z As Double, ByVal b As Double, ByVal eep As Double,  
ByVal sinthetaqb As Double, ByVal p As Double, ByVal a As Double, ByVal ee As  
double, ByVal costhetaqb As Double) As Double

Dim aa As Double = CDBl(txtA.Text) 'declaring a again cannot work, because it is declared in the function above

Dim ZZ As Double = CDBl(txtZcoord.Text)

Dim X As Double = CDBl(txtXcoord.Text)

Dim Y As Double = CDBl(txtYcoord.Text)

Dim ff As Double = CDBl(txtF.Text)

Dim f As Double

• for WGS84 a=6378137 metres,b= 6356752.3142metres,f=1-b/a = 0.003352810672

z = ZZ

a = aa  
f = ff

b = a \* (1 - f) 'and f = 1 - b / a , e = sqrt(1-(b<sup>2</sup>/a<sup>2</sup>))

p = Math.Sqrt(X<sup>2</sup> + Y<sup>2</sup>)

• for WGS84 a=6378137 metres,b= 6356752.3142metres,f=1-b/a = 0.003352810672

b = a \* (1 - f) 'and f = 1 - b / a , e = sqrt(1-(b<sup>2</sup>/a<sup>2</sup>))

p = Math.Sqrt(X<sup>2</sup> + Y<sup>2</sup>)

'Dim h As Double

Dim theta As Double = Math.Atan((z \* a) / (p \* b))' is this theta in radians?

'Return theta —>for check computation

eep = ((a<sup>2</sup>) - (b<sup>2</sup>)) / (b<sup>2</sup>)

ee = ((a<sup>2</sup>) - (b<sup>2</sup>)) / (a<sup>2</sup>)

sinthetaqb = Math.Sin(theta) ^ 3 'means --> sin<sup>3</sup>(theta)

costhetaqb = Math.Cos(theta) ^ 3 'means --> cos<sup>3</sup>(theta)

Dim phi As Double

'Check by Return value of Costhetaqb

Dim N As Double 'N is the Radius of Curvature in the Prime Vertical

Dim rphi As Double = (phi / 180) \* Math.PI

'Dim degphi As Double

•Phi = (phi / 180) \* Math.PI

•phi convert phi value in radians •\*

N = (a<sup>2</sup>) / (Math.Sqrt(a<sup>2</sup> \* (Math.Cos(phi))<sup>2</sup> + (b<sup>2</sup>) \* Math.Sin(phi)<sup>2</sup>))

' degphi is phi in degrees, rphi is phi in Radians

phi = (Math.Atan((z + b \* eep \* sinthetaqb) / (p - a \* ee \* costhetaqb)))

**Return** phi \* 180/Math.PI

End Function

**Function** CalculateLabda(ByVal X As Double, ByVal Y As Double) As Double 'now ok

Dim Labda As Double

**Labda** = Math.Atan(Y / X)

**Return** (Labda \* 180) / (Math.PI)

End Function

Function CalculateHt(ByVal p As Double, ByVal phi As Double, ByVal N As Double)

As Double

Dim aa As Double = CDBl(txtA.Text) 'declaring a again cannot work because it is declared in the function

'above so an alternative approach is used here and aa is introduced instead of a

Dim Z As Double = CDBl(txtZcoord.Text)

Dim X As Double = CDBl(txtXcoord.Text)

Dim Y As Double = CDBl(txtYcoord.Text)

Dim ff As Double = CDBl(txtF.Text)

Dim a As Double The semimajor axis

Dim b As Double The seminor axis

Dim f As Double 'The flattenig axis

<sup>1</sup> for WGS84 constants for example as in Hofmann-Wellenhof et.al,GPS Theory and Practice

I | f6378137.00000 metres,b= 6356752.31425 metres,f=1-b/a = 0.00335281066474

**a** = **aa** <sup>1</sup> see notes above

\*= ff' same case as that of aa

= **a** \* (**1 - f**) 'and f= **1 - b / a** , e = sqrt(**1 -(b<sup>2</sup>/a<sup>2</sup>)**), calculation of b value

P"Math.Sqrt(X <sup>2</sup> + Y <sup>2</sup>)

h As Double

\* **a** \* (**1 - f**) 'and f= **1 - b / a** , e = sqrt(**1 -(b<sup>2</sup>/a<sup>2</sup>)**)

\*Math.Sqrt(X <sup>2</sup> + Y <sup>2</sup>)

Dim theta As Double = Math.Atan((Z \* a) / (p \* b))' is this theta in radians?

•Return theta ~>for check computation

Dim eep As Double = ((a ^ 2) - (b ^ 2)) / (b ^ 2)

Dim ee As Double = ((a ^ 2) - (b ^ 2)) / (a ^ 2)

Dim sinthetaqb As Double = Math.Sin(theta) ^ 3 'means --> sin^3(theta)

Dim Costhetaqb As Double = Math.Cos(theta) ^ 3 'means ~> cos^3(theta)

phi = Math.Atan((Z + b \* eep \* sinthetaqb) / (p - a \* ee \* Costhetaqb))

N = (a ^ 2) / (Math.Sqrt(a ^ 2 \* (Math.Cos(phi)) ^ 2 + (b ^ 2) \* Math.Sin(phi))

'to compute ellipsoidal height

h = (p / Math.Cos(phi)) - N

Return h

End Function

End Class



## APPENDIX 4

### COORDINATE TRANSFORMATION FROM GEODETIC TO GEOCENTRIC

Public Class Form1

Phi, Labda, h ---> X,Y,Z

' DSN: by Washington Abuto

' F56/P/9061/2005

' for MSC GIS Project October2007

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim a As Double = Cdbl(txta.Text) 'a if the semimajor axis

Dim phi As Double = Cdbl(txtPhi.Text) ' phi is the Latitude

Dim labda As Double = Cdbl(txtLabda.Text) ' Labda is the Longitude

Dim f As Double = Cdbl(txtf.Text) ' f is the flattening

Dim h As Double = Cdbl(txtht.Text) The Ellipsoidal Height entered by user

'Dim aTxtNeuw As Double = Cdbl(txtNeuw)

Dim N As Double

MsgBox(GeocentricX(a, phi, f, h, N)) 'Messagebox for Xcoordinate value

MsgBox(GeocentricY(a, phi, f, h, N)) 'Messagebox for Ycoordinate value

MsgBox(GeocentricZ(a, phi, f, h, N)) 'Messagebox for Zcoordinate value

'TxtNeuw.Text = Convert.ToString(GeocentricZ(a, phi, 0))

'Function GeocentricX(ByVal a As Double, ByVal phi As Double, ByVal f As Double, ByVal h As Double, ByVal N As Double) As Double

txtX.Text = Convert.ToString(GeocentricX(a, phi, f, h, N))

'Function GeocentricY(ByVal a As Double, ByVal phi As Double, ByVal f As Double, ByVal h As Double, ByVal N As Double) As Double

txtY.Text = Convert.ToString(GeocentricY(a, phi, f, h, N))

'Function GeocentricZ(ByVal a As Double, ByVal phi As Double, ByVal f As Double, ByVal h As Double, ByVal N As Double) As Double

totZ.Text = Convert.ToString(GeocentricZ(a, phi, f, h, N))

'Function PrimeveticalNeuw(ByVal a As Double, ByVal phi As Double, ByVal f As Double) As Double

```

MsgBox(PrimeveticalNeuw(a, phi, f))' Radius of curvature in the prime vertical
'aTxtNeuw.Text = Convert.ToString(PrimeveticalNeuw(a, phi, f))
End Sub
Function PrimeveticalNeuw(ByVal a As Double, ByVal phi As Double, ByVal f As
Double) As Double
'Dim txtNeuw As Double = CDb1(txtNeuw.Text)' radius of curvature
Dim N As Double 'N is the Radius of Curvature in the Prime Vertical
'Dim phi As Double
Dim b As Double
•f= 1 - b / a
b = a * (1 - f)
Dim rphi As Double
rphi = (phi / 180) * Math.PI
'rphi convert phi value in radians
N = (a ^ 2) / (Math.Sqrt(a ^ 2 * (Math.Cos(rphi)) ^ 2 + (b ^ 2) * Math.Sin(rphi) ^ 2))
Return N
End Function
Function GeocentricX(ByVal a As Double, ByVal phi As Double, ByVal f As Double,
ByVal h As Double, ByVal N As Double) As Double
Dim labda As Double = CDb1(txtLabda.Text)' Labda is the Longitude
Dim rLabda As Double = (labda / 180) * Math.PI
Dim b As Double
b = a * (1 - f)
Dim X As Double
Dim rphi As Double
rphi = (phi / 180) * Math.PI
N = (a ^ 2) / (Math.Sqrt(a ^ 2 * (Math.Cos(rphi)) ^ 2 + (b ^ 2) * Math.Sin(rphi) ^ 2))
X = (N + h) * Math.Cos(rphi) * Math.Cos(rLabda)
Return X
End Function

```

```

Function GeocentricY(ByVal a As Double, ByVal phi As Double, ByVal f As Double,
ByVal h As Double, ByVal N As Double) As Double
Dim labda As Double = CDbl(txtLabda.Text)' Labda is the Longitude
Dim rLabda As Double = (labda / 180) * Math.PI
Dim b As Double
b = a * (1 - f)
Dim Y As Double
Dim rphi As Double
rphi = (phi / 180) * Math.PI
N = (a ^ 2) / (Math.Sqrt(a ^ 2 * (Math.Cos(rphi)) ^ 2 + (b ^ 2) * Math.Sin(rphi) ^ 2))
Y = (N + h) * Math.Cos(rphi) * Math.Sin(rLabda)
Return Y
End Function

```

```

Function GeocentricZ(ByVal a As Double, ByVal phi As Double, ByVal f As Double,
ByVal h As Double, ByVal N As Double) As Double
Dim b As Double
b = a * (1 - 0)
Dim Z As Double
Dim rphi As Double
rphi = (phi / 180)* Math.PI
N = (a ^ 2) / (Math.Sqrt(a ^ 2 * (Math.Cos(rphi)) ^ 2 + (b ^ 2) * Math.Sin(rphi) ^ 2))
Z = ((b ^ 2 / a ^ 2) * N + h) * Math.Sin(rphi)
Return Z
End Function
End Class

```

## APPENDIX 5

### CONVERTS DEGREES MINUTES AND SECONDS TO DECIMAL DEGREES

Public Class Form1

Purpose: to aid users in converting Data for input to geo database

'This programme use entered values in degrees minutes and seconds to to work out the values in decimal degrees and also compute the radians

'i.e computes the decimal degrees and radians given values in degrees minutes and seconds

Program Coded by W. Abuto

'Program developed in October 2007

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
    Dim aDegree As Double = Cdbl(TextBox1.Text)
```

```
    Dim aMinute As Double = Cdbl(TextBox2.Text)
```

```
    Dim asecond As Double = Cdbl(TextBox3.Text)
```

```
    This message box displays the results of computed values of decimal degrees
```

```
    MsgBox(DecimalDegrees(aDegree, aMinute, asecond))
```

```
    This message box displays the results of converted values of decimal degrees in radians
```

```
    MsgBox(ToRadians(aDegree, aMinute, asecond))
```

```
'Textbox4 is used for displaying the converted values in decimal degrees
```

```
    TextBox4.Text = Convert.ToString(DecimalDegrees(aDegree, aMinute, asecond))
```

```
'Textbox5 is used for displaying the converted values in Radians
```

```
    TextBox5.Text = Convert.ToString(ToRadians(aDegree, aMinute, asecond))
```

```
End Sub
```

This function DecimalDegreesQ computes the decimal degrees using the values entered by the user in degrees, minutes and seconds

```
Function DecimalDegrees(ByVal deg As Double, ByVal min As Double, ByVal sec As Double) As Double
```

```
    Dim adeg As Double *
```

```
    adeg = deg + min / 60 + sec / 3600
```

```
Return adeg
```

```
End Function
```

This function ToRadians() computes the radians using the values entered by the user in degrees, minutes and seconds

```
Function ToRadians(ByVal deg As Double, ByVal min As Double, ByVal sec As  
Double) As Double
```

```
Dim aPI As Double = Math.PI
```

```
Dim arad As Double
```

```
arad = (deg + min / 60 + sec / 3600) * aPI / 180
```

```
Return arad
```

```
End Function
```

```
End Class
```

## APPENDIX 6

### CONVERTS DECIMAL DEGREES TO DEGREES MINUTES AND SECONDS

Public Class Form1

'Decimal Degrees To Degrees Conversion Program

'Program developed by W. Abuto October 2007 to aid users in converting Data for input to geo database

This programme use the decimal degrees to work out the values in degrees minutes and seconds

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim adeg As Double = CDb1(TextBox1.Text)

'Dim aRadian As Double = CDb1(TextBox6.Text)

MsgBox(Discretedeg(adeg))

MsgBox(DecdegToRadians(adeg))

TextBox2.Text = Convert.ToString(Discretedeg(adeg))

TextBox3.Text = Convert.ToString(DiscreteMin(adeg))

TextBox4.Text = Convert.ToString(DiscreteSec(adeg))

TextBox6.Text = Convert.ToString(DecdegToRadians(adeg))

'code is for combining separate degrees ,minutes and seconds to a string viewed in user interface

Dim str1, str2, str3 As String

str1 = TextBox2.Text

str2 = TextBox3.Text

str3 = TextBox4.Text

TextBox5.Text = str1 & " " & str2 & " " & str3

End Sub

This function computes the degrees component from the decimal degrees

Function Discretedeg(ByVal adeg As Double) As Double

Dim Deg As Double

Dim Min As Double \*

Dim Sec As Double

```

Deg = Int(adeg)
Min = Int((adeg - Deg) * 60)
Sec = (((adeg - Deg) * 60) - Min) * 60
Return Deg
Return Min
Return Sec
End Function

```

This function DiscreteMin() —>computes the minutes component from the decimal degrees

```

Function DiscreteMin(ByVal adeg As Double) As Double
    Dim Deg As Double
    Dim Min As Double
    ' Dim Sec As Double
    Deg = Int(adeg)
    Min = Int((adeg - Deg) * 60)
    'Sec = (((adeg - Deg) * 60) - Min) * 60
    Return Min
End Function

```

This function computes the seconds component from the decimal degrees

```

Function DiscreteSec(ByVal adeg As Double) As Double
    Dim Deg As Double
    Dim Min As Double
    Dim Sec As Double
    Deg = Int(adeg)
    Min = Int((adeg - Deg) * 60)
    Sec = (((adeg - Deg) * 60) - Min) * 60
    Return Sec
End Function

```

```

Function DecdegToRadians(ByVal adeg As Double) As Double
    Dim aPI As Double = Math.PI *

```

'it is very important to include math.pi,without it the computation will give zero value

```
Dim aRadian As Double
```

```
aRadian = adeg * aPI / 180
```

```
Return aRadian
```

```
End Function
```

```
End Class
```



'it is very important to include math.pi,without it the computation will give zero value

```
Dim aRadian As Double
```

```
aRadian = adeg * aPI / 180
```

```
Return aRadian
```

```
End Function
```

```
End Class
```

# APPENDIX 7

## SAMPLE DATABASE TABLE STRUCTURE

## I Benchmark : Table

Field Name	Data Type	Description
TSH^SSE	AutoNumber	Auto number Given By Computer Automatically
Benchmark_name	Text	The SOK Fundamental Bench Mark Number
Geodetic_coord_available	Yes/No	Please select Yes OR No
Northings	Number	The UTM Northing coordinates positive North in metres-(X)
Eastings	Number	The UTM Easting coordinates in metres-(Y) positive East
Order	Number	The accuracy classification
Coordinatejnits	Text	The unit of measurements used for coordinates- select-the default is meters
Resolution	Number	The standard error of the point in meters
Monument_type	Text	The nature of physical point
elevation	Number	Height above the sea level (orthometric height) Jn meters
Leveling_method	Text	Please Select from the list
Projection	Text	Projection Used
Buffer	Number	Buffer radius is 6.096 meters (20 feet radius survey Act- all monuments Kenya)
Zone_number	Number	Universal Transverse marcator zone number- select
VMapsheet_no	Text	Map sheet number
Datum	Text	Select the datum
Date_Established	Date/Time	The Date when the station was established
Status	Text	Select the most current status
Status_Date	Date/Time	Date when the station was last visited to confirm the status
VStnPicture_1	Text	photo of station view1
VStnPicture_2	Text	photo of station view2
Comps Id	Number	comps

Field Properties

General Lookup j

Field Size	Long Integer
New Values	Increment
Format	
Caption	station Identifier
Indexed	Yes (No Duplicates)

A field name can be up to 64 characti

## 1 Trig^station: Table

Field Name	Data Type	
Station_j d H H H B H	AutoNumber	Auto number Given By Computer Automatically
Station_name	Text	The SOK station Number
Is_Height_data_available	Yes/No	Please respond with Yes OR No (trigonometric Heighting)
Northings	Number	The UTM Northing coordinates positive North in metres-(X)
Eastings	Number	The UTM Easting coordinates in metres-(Y) positive East
Order	Number	The accuracy classification default order is 4
Coordinate_units	Text	The unit of measurements used for coordinates- select-the default is metres
Resolution	Number	
Monumenttype	Text	The nature of physical point
elevation	Number	Height above the sea level (orthometric height) Jn metres
Survey_method	Text	Please Select from the list
Projection	Text	Projection Used
Buffer	Number	Buffer radius is 6.096 meters (20 feet radius survey Act- all monuments Kenya)
Zone_number	Number	Universal Transverse marcator zone number- select
HMapsheet_no	Text	Map sheet reference number
Datum	Text	Select the datum
Date_Established	Date/Time	The Date when the station was established
Status	Text	Select the most current status
Status_Date	Date/Time	Date when the station was last visited to confirm the status
H5tnPicture_1	Text	photo of station view1
H5tnPicture_2	Text	photo of station view2
Compsjd	Number	comps identifier from computation table
Soecific Locality	Text	Name of soecific Location of the station

Field Properties

General Lookup i

Field S...p

1 mm Trippr

1

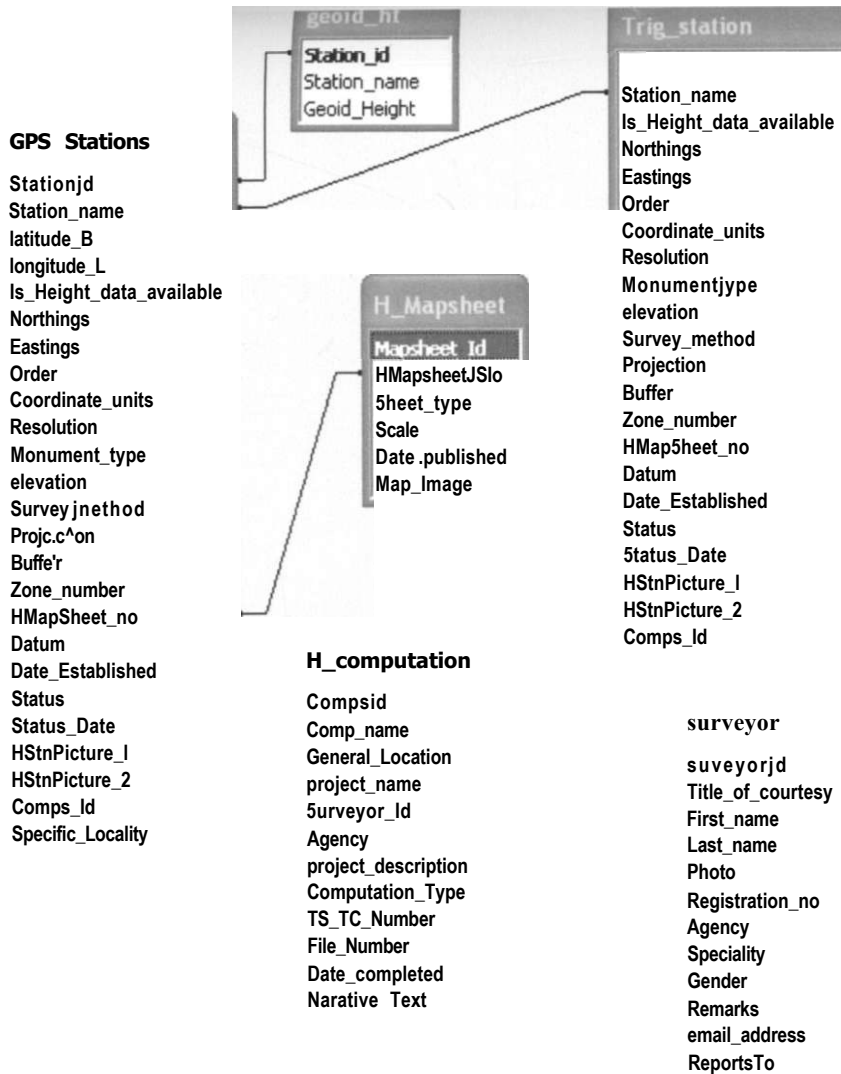


FIGURE 4.2.1: ENTITY RELATIONSHIP DIAGRAM

		<b>Benchmark</b>
		Benchmark_id
		Benchmark_name
		Geodetic_coord_available
		Northings
		Eastings
		Order
		Coordinate_units
Camps_Jd	n	Resolution
Comps_name		Monumenttype
General_Location		elevation
project_name		Leveling_method
Surveyor_Id		Projection
Agency		Buffer
project_description		Zone_number
Computation_Type		VMapsheet_no
GL_TS_Number		Datum
File_Number		Date_Established
Date_completed		Status
Narative_Text		5status_Date
		VStnPicture_1
		VStnPicture_2
		Mapimage
		Comps_Id
MapsheetId		Mapsheetjd
Mapsheet_Name		SHAPE
!sheet_type		
!Scale		
iDate ..published		
•Map_Image		

## APPENDIX 8

SAMPLE DATA ENTRY FORMS CREATED FOR DATABASE



m GPS\_Stdtions

n X : S3 Trig\_station

Stationjd	
Station_name	148ST5
Latitude_B	1 29' 17.70455" S
Longitude_L	36 37' 01 .93966"E
Is_height_data_Availabl^	
Northings	9835375.07
Eastings	234861.94
order	1
CoordinateJJnits	Meters i J
Resolution	1 0.01 -1
Monument_type	IRON_PIN 3
elevation (metres)	1751.642
Survey_method	Static
Projection	UTM
Buffer	6.096
Zone_number	1 37 j J
HMapSheet_no	1148/3
Datum	Arc60
Date_Established	21/02/2003
Status	INTACT_USABLE
Status_Date	04/07/2007
HStnPicture_1	C:\Documents and Settings\W. AE
HStnPicture_2	C:\Documents and Settings\W. AE
comps_Id	1 11 j j
	1

stationjd	I 1
Station_name	I48UT105
s_height_data_Availal	
Northings	9881623.57
Eastings	241255.04
order	
CoordinateJJnits	Meters
Resolution	0.01_i J
Monument_type	I.P.C.U
elevation	2313.28
Survey_method	Triangulation
Projection	UTM
Buffer	6.096
Zone_number	37^J
Map5heet_no	I48/I
Datum	Arc60 J d
Date_Established	02/04/1970
Status	Pillar destroyed U
5tatus_Date	03/07/2007
HStnPicture_1	C:\Documents and Settings\W. AE
HStnPicture_2	C:\Documents and 5ettings\W. AE
comps_Id	1
FID	1
SHAPE	"ZJ

Record: H | | [ 1 • | H !•\*! of 25

Record: &lt; | | f 1 • | H !•\*! of 375





0 1 ®

compjd	1
Comp_Name	GL/4A/6
Locality	NAIROBI
Project Name	NAIROBI CON
surveyorjd	20
Agency	Directorate Of Overseas Survt *
project description	NAIROBI GEODETIC BENCHM^J
computation_type	Non Regorous
GL_TS_Number	GL/4A/6
File_number	GL/4A/6
Date_Cfimpleted	12/02/1959
Narative_Text	Heights of Benchmarks for Nairobi Control

**H\_computation**

compjd	1
Comp_Name	T5102/IV
locality	CENTRAL PROVINCE
Project Name	CENTRAL PROVINCE
surveyorjd	29
Agency	Survey of Kenya J
project description	TRIANGULATION
Computation Jype	(Least Squares Adjustment J
TS_TC_Number	AL44/XI
File_number	AL44/XI
Date_Completed	14/06/1963
Narative Text	Computed by I.F Cuthil. Nairobi and sorroi

# APPENDIX 9

## SAMPLE COMPUTATIONS

TRANSFORMATION OF GEODETIC COORDINATES TO UTM COORDINATES					
26 October 2007 v	COMPUTATION FORM FOR NORTHERN LATITUDE				
Input:	Station Name	SKP3		Output: Northings,Eastings	
Ellipsoid Used	Clarke 1880(Modified)	●		Unit: Metres	
Latitude-(Phi)	1.27079833333333	N	v	With FN	140572.764789637
Unit: Decimal Degrees					
Longitude(Lamda)	35.3998405555556	E	v	Without FN	140572.764789637
ZonefZn)	36				Eastings(E)
Flattening (f)	0.003407561379			With FE	767060.184886496
Semi-major Axis(a]	6378249.145			Without FE	267060.184886496
For Clarkel 880 a=6378249m,f=0.003407561379				Transform to	
For WGS84 a=6378137.0m. f=0.00335281066474				UTM-Coordinates	

FN and FE are 10000000 and 500000 respectively

FIGURE 6.3 B: SHOWS THE RESULT OF TRANSFORMING GEODETIC COORDINATES TO UTM COORD. SYSTEM (NORTH OF EQUATOR)

26 October 2007 v

## TRANSFORMATION OF UTM COORDINATES TO GEODETIC COORDINATE SYSTEM

Station Name	SKP3	FOR POINTS NORTH OF EQUATOR	
	Input Northings and eastings	Output: Latitude and Longitude	
		Units in decimal degrees	
Ellipsoid Used	Clarke 1880(Modified) v		
Units in Metres			
Northings(N)	140572.77	Latitude	1.27081358810247
Eastings(E)	767060.20	Phif	1.27192119596507
Zone(Zn)	36		
Semi-Major Axis	6378249.145	Longitude	35.3998406912983
Flattening(f)	0.003407561379		
For Clarke 1880	a= 6378249.145 m, f=0.003407561379	Convert Coordinates To Geodetic	
For WGS84	a=6378137.0, f=0.00335281066474		
Bessel 1841:	a=6377397.155, f= 0.003342773182		

FIGURE 6.3C: SHOWS THE TRANSFORMATION OF UTM COORDINATES TO GEODETIC COORDINATES SYSTEM (THE REVERSE COMPUTATION)

ffl Forml

•H(E

## Conversion of Coordinates from Geocentric to Geodetic

Station Name REFERENCE4

18 October 2007 •

Input: X,YZ-and-Semi Major Axis..a, and Flattening.f Output: Latitude,Longitude and Height h

X Coord 4210520.621

Phi Latitude 46.9999999923913

Y\_Coord 1128205.600

Z\_Coord 4643227.495

Semi Major Axis(a) 6378137.0

Longitude\_Labda 14.9999999985839

Flattening(f) 0.00335281066474

Ellipsoid Used WGS84-Spheroid V

For Clarke1880-a=6378249m,f=0.003407561379

Ellipsoidal\_Ht 1999.99911595136

For WGS84 a=6378137.0m&gt;0.00335281066474 (Metres)

All Units to be in Metres. Angles in Decimal Degrees

Convert  
Geocentric to  
Geodetic

October 2007

FIGURE 6.3 D : SHOWS RESULT OF COORDINATE TRANSFORMATION  
FROM CARTESIAN TO GEODETIC COORDINATE SYSTEM

## Conversion of Coordinates from Geodetic to Geocentric

Station Name	REFERENCE4	19 October 2007	v
Input: a, f ,phi Xabda, ellipsoidalheight(h)		Output: xyz	
Ellipsoid_used	WGS84-Spheroid	J	
Latitude (Phi)	47	X_Coord	4210520.62095676
Longitude (Lamda)	15	Y_Coord	1128205.60009995
EllipsoidalHt (h)	2000		
Semi-Major Axis	6378137.0	Z_Coord	4643227.49622366
Flattening	0.00335281066474		

All units of Measurements in metres. Angles are in Decimal degrees

For Clarke1 880-a=6378249m,f=0.003407561379

ConvertGeodetic to  
Geocentric

For WGS84 -a=6378137.0m,f=0.00335281066474

If Latitude is South of Equator ,Key in Minus first then the Latitude

October 2007

FIGURE 6.3 E: SHOWS RESULT OF TRANSFORMATION OF GEODETIC COORDINATES TO CARTESIAN COORDINATE SYSTEM

Form1

22 October 2007 v

TRANSFORMATION OF GEODETIC COORDINATES TO UTM COORDINATES

Input: Station Name 149S3

Output: Northings, Eastings

Ellipsoid Used	Clarke 1880(Modified)	v	Unit: Metres
Latitude-(Phi)	-1.46863576111111	S v	Northings(N)
Unit: Decimal Degrees			With FN 9837592.78712578
Longitude(Lamda)	37.0624021166667	E   v	Without FN -162407.212874219
Semi-major Axis(a)	6378249.145		Eastings(E)
Flattening (f)	q.003407561379		With FE 284419.099898831
ZonefZn)	37		Without FE -215580.900101169

For southern Latitudes enter minus sign first then the value

For Clarkel 880 a=6378249m>0.003407561379

For WGS84 a=6378137.0m, NO.00335281066474

Transform to UTM-Coordinates

FN and FE are 1000000 and 500000 respectively

**FIGURE 6.3F : SHOWS TRANSFORMATION OF GEODETIC COORDINATES TO UTM COORDINATE SYSTEM (SOUTH LATITUDE)**

**Formi**

C o n v e r s i o n   P r o g r a m

26 October 2007

STATION   SKP3

Purpose

LATITUDE NORTH

v

**Angle In dd mm ss**

Degrees

Minutes

seconds

16

14.874

Convert To Degrees and  
to Radians

Answer In Decimal Degrees

Answer in Radians

1.27079833333333

0.0221796150455231





C o n v e r s i o n P r o g r a m

26 October 2007 v

STATION SKP3 Purpose LONGITUDE EAST

### Angle In dd mm ss

Degrees	Minutes	seconds
35	23	59.426

Convert To Degrees and  
to Radians

Answer In Decimal Degrees

Answer in Radians

35.3998405555556

0.617843772375463



**EB Forml****Conversion Program**

25 November 2007

STATION 149S3 Purpose LONGITUDE EAST v

**Angle In dd mm ss**

Degrees	Minutes	seconds
37	03	44.64762

Convert To Degrees and  
to Radians

Answer In Decimal Degrees

37.0624021166667

Answer in Radians

0.646860945633949

**FIGURE 6.3M: SHOWS CONVERSION OF LONGITUDE IN DEGREES  
MINUTES AND SECONDS TO DECIMAL DEGREES**

**Form1**

26 October 2007 v

## Conversion Decimal Degrees To Degrees Minutes And Seconds

Station Name	HYPOTHETICAL	Purpose	LONGITUDE EAST	t i
Decimal Degrees		Degrees	Min	Sec
53.9546782955193	53	57	16.84186386S	
Value In Radians	Convert to dd mm ss format			
0.941686783111134				
	Degrees Minutes Seconds			
	53 57 16.8418638694834			

FIGURE 6.3G SHOWS THE RESULT OF CONVERSION OF AN ANGLE IN DECIMAL DEGREES TO DEGREES MINUTES AND SECONDS

## APPENDIX 10

### SAMPLE DATABASE RECORDS

<i>Stationjd</i>	Station name	Latitude B	Longitude L
1	148ST5	1 29' 17.70455" S	36 37' 01 .93966"E
	2149S2	1 06' 01.51548"S	37 06' 44 49970"E
	3149S3	1 28' 07. 08874"S	37 03' 44. 64762"E
	4KJ01	1 08' 53. 64349"S	36 53' 53.69124"E
	5KJ02	1 10' 38. 75232"S	36 56' 19 ,18118"E
	6KJ03	1 11' 18. 23361"S	36 47' 25.83794"E
	7KJ04	1 11' 24. 58187"S	36 51' 32. 45660"E
	8KJ05	1 12' 52. 94674"S	36 58' 03. 03261"E
	9KJ06	1 15' 14. 07097"S	37 00' 38.43740"E
	10KJ07	1 12' 05 .67685"S	37 03' 54. 68593"E
	11 KJ08	1 14' 32. 80888"S	37 05' 52.12847"E
	12KJ09	1 15' 23. 26364"S	36 41'01.85797"E
	13 KJ10	1 14' 12. 26252"S	36 44'01. 97427"E
	14KJ14	1 16' 21. 43488"S	36 48' 39.80047"E
	15 KJ12	1 15' 21.16614"S	36 54' 56.97341"E
	16 KJ13	1 18'49. 21307"S	36 59' 05 44134"E
	17 KJ14	1 16' 38 . 51834"S	37 02' 11. 71732"E
	18 KJ15	1 18* 23. 61464"S	37 04' 56. 57494"E
	19KJ16	1 18' 59 . 41044"S	36 39' 33. 64896"E
	20 KJ17	1 21'03. 48035"S	36 42' 44. 30843"E
	21 KJ18	1 23' 35. 16717"S	36 45' 42.90531"E
	22 KJ19	1 20' 24. 79077"S	36 45' 56.42869"E
	23KJ20	1 20' 08. 02429"S	36 53' 39. 89564"E
	24 KJ21	1 22' 56. 55056"S	36 56' 00. 62238"E
	25 KJ22	1 21'05. 76944"S	36 57' 57. 59399"E

> height_dataj	Northings 1	Eastings 1	order	Coordinate Uni
2	9835375.07	234861.94		2 Meters
0	9878315.023	289949.592		2 Meters
0	9837592.787	284419.1		2 Meters
0	9873011.157	266117.332		2 Meters
0	9869785.482	270618.912		2 Meters
0	9868559.858	254126.032		2 Meters
0	9868370.837	261753.13		2 Meters
0	9865665.45	273833.428		2 Meters
0	9861333.871	278642.179		2 Meters
0	9867125.458	284706.218		2 Meters
0	9862608.535	288340.853		2 Meters
0	9861021.492	242256.828		2 Meters
0	9863207.837	247825.54		2 Meters
0	9859246.533	256421.085		2 Meters
0	9861107.686	268083.412		2 Meters
0	9854722.673	275771.774		2 Meters
0	9858741.974	281528.489		2 Meters
0	9855517.537	286628.4		2 Meters
0	9854377.754	239534.745		2 Meters
0	9850571.212	245434.915		2 Meters
0	9845915.976	250962.559		2 Meters
0	9851765.427	251375.325		2 Meters
0	9852293.211	265707.212		2 Meters
0	9847119.741	270063.268		2 Meters
0	9850526.016	273677.291		2 Meters

Resolution	Monument_type	elevation (metres)	Survey_method	Projection	Buffer	Zone_number	HMapSheet_no
0.01	IRON_PIN	1751.642	Static	UTM	6.096	37	148/3
0.01	IRON_PIN	1389.794	Static	UTM	6.096	37	149/1
0.01	IRON_PIN	1727.096	Static	UTM	6.096	37	149/3
0.01	BRASS_CAP_WITH_IRON_PIN	1454.634	Static	UTM	6.096	37	148/2
0.01	BRASS_CAP_WITH_IRON_PIN	1429.742	Static	UTM	6.096	37	148/2
0.01	BRASS_CAP_WITH_IRON_PIN	1646.087	Static	UTM	6.096	37	148/2
0.01	BRASS_CAP_WITH_IRON_PIN	1518.012	Static	UTM	6.096	37	148/2
0.01	BRASS_CAP_WITH_IRON_PIN	1407.386	Static	UTM	6.096	37	148/2
0.01	BRASS_CAP_WITH_IRON_PIN	1387.763	Static	UTM	6.096	37	149/3
0.01	BRASS_CAP_WITH_IRON_PIN	1363.585	Static	UTM	6.096	37	149/1
0.01	BRASS_CAP_WITH_IRON_PIN	1377.832	Static	UTM	6.096	37	149/1
0.01	BRASS_CAP_WITH_IRON_PIN	1834.37	Static	UTM	6.096	37	148/3
0.01	BRASS_CAP_WITH_IRON_PIN	1703.072	Static	UTM	6.096	37	148/1
0.01	BRASS_CAP_WITH_IRON_PIN	1561.377	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1473.976	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1471.391	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1386.312	Static	UTM	6.096	37	149/3
0.01	BRASS_CAP_WITH_IRON_PIN	1365.482	Static	UTM	6.096	37	149/3
0.01	BRASS_CAP_WITH_IRON_PIN	1782.569	Static	UTM	6.096	37	148/3
0.01	BRASS_CAP_WITH_IRON_PIN	1752.437	Static	UTM	6.096	37	148/3
0.01	BRASS_CAP_WITH_IRON_PIN	1624.835	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1684.073	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1525.04	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1507.525	Static	UTM	6.096	37	148/4
0.01	BRASS_CAP_WITH_IRON_PIN	1495.617	Static	UTM	6.096	37	148/4



Datum	Date_Establish^	Status
Arc60	21/02/2003	INTACT_USABLE
Arc60	21/02/2003	INTACT_USABLE
Arc60	21/02/2003	INTACTJJSABLE
Arc60	25/02/2003	DESTROYED_NOT_USABLE
Arc60	25/02/2003	INTACTJJSABLE
Arc60	25/02/2003	INTACT_USABLE
Arc60	25/02/2003	INTACT_USABLE
Arc60	25/02/2003	INTACT_USABLE
Arc60	26/02/2003	INTACT_USABLE
Arc60	26/02/2003	USABLE_FAIR
Arc60	26/02/2003	USABLE_FAIR
Arc60	25/02/2003	INTACT_USABLE
Arc60	24/02/2003	DESTROYED_NOT_USABLE
Arc60	25/02/2003	INTACT_USABLE
Arc60	25/02/2003	DESTROYED_NOT_USABLE
Arc60	26/02/2003	INTACTJJSABLE
Arc60	25/02/2003	INTACTJJSABLE
Arc60	25/02/2003	INTACTJJSABLE
Arc60	25/02/2003	DESTROYED_NOT_USABLE
Arc60	25/02/2003	INTACT_USABLE
Arc60	26/02/2003	INTACTJJSABLE
Arc60	26/02/2003	DESTROYED_NOT_USABLE
Arc60	24/02/2003	INTACT_USABLE
Arc60	24/02/2003	INTACT_USABLE
Arc60	26/02/2003	INTACT_USABLE

<u>Status</u>	<u>Date</u>	<u>HStnPicture_1</u>
	04/07/2007	C:\Documents and Settings\W. ABUTO\My Documents\SP_A0207.jpg
	03/07/2007	SP_A0190.jpg
	04/07/2007	SP_A0205.jpg
	02/07/2007	SP_A0181.jpg
	02/07/2007	SP_A0179.jpg
	02/07/2007	SP_A0187.jpg
	02/07/2007	SP_A0185.jpg
	02/07/2007	SP_A0177.jpg
	02/07/2007	SP_A0151.jpg
	03/07/2007	SP_A0193.jpg
	03/07/2007	SP_A0198.jpg
	29/06/2007	SP_A0163.jpg
	04/07/2007	SP_A0209.jpg
	21/06/2007	SP_A0138.jpg
	21/07/2007	SP_A0147.jpg
	04/07/2007	SP_A0203.jpg
	21/06/2007	SP_A0149.jpg
	03/07/2007	SP_A0200.jpg
	29/06/2007	SP_A0160.jpg
	29/07/2007	SP_A0158.jpg
	29/07/2007	SP_A0043.jpg
	29/07/2007	SP_A0156.jpg
	21/06/2007	SP_A0140.jpg
	21/06/2007	SP_A0026.jpg
	21/06/2007	SP_A0142.jpg

1	Datum	/Date_Establish^	Status
<b>Arc60</b>	21/02/2003	1	INTACT_USABLE
Arc60	21/02/2003		INTACT_USABLE
Arc60	21/02/2003	1	INTACTJJSABLE
Arc60	25/02/2003		DESTROYED_NOT_USABLE
Arc60	25/02/2003	1	INTACTJJSABLE
Arc60	25/02/2003		INTACT_USABLE
Arc60	25/02/2003		INTACTJJSABLE
Arc60	25/02/2003	1	INTACT_USABLE
Arc60	26/02/2003	1	INTACTJJSABLE
Arc60	26/02/2003		USABLE_FAIR
Arc60	26/02/2003		USABLE_FAIR
Arc60	25/02/2003		INTACT_USABLE
Arc60	24/02/2003		DESTROYEDJsiOT_USABLE
Arc60	25/02/2003		INTACT_USABLE
Arc60	<25/02/2003		DESTROYED_NOT_USABLE
Arc60	26/02/2003		INTACT_USABLE
Arc60	25/02/2003		INTACTJJSABLE
Arc60	25/02/2003		INTACTJJSABLE
Arc60	25/02/2003		DESTROYED_NOT_USABLE
Arc60	25/02/2003		INTACT_USABLE
Arc60	26/02/2003	1	INTACT_USABLE
Arc60	26/02/2003		DESTROYED_NOT_USABLE
Arc60	24/02/2003	1	INTACTJJSABLE
Arc60	24/02/2003		INTACTJJSABLE
Arc60	26/02/2003		INTACTJJSABLE

<u>Status</u>	<u>Date</u>	<u>HStnPicture</u>	<u>1</u>
	04/07/2007	C:\Documents and Settings\W. ABUTO\My Documents\SP_A0207.jpg	
	03/07/2007	SP_A0190.jpg	
	04/07/2007	SP_A0205.jpg	
	02/07/2007	SP_A0181.jpg	
	02/07/2007	SP_A0179.jpg	
	02/07/2007	SP_A0187.jpg	
	02/07/2007	SP_A0185.jpg	
	02/07/2007	SP_A0177.jpg	
	02/07/2007	SP_A0151.jpg	
	03/07/2007	SP_A0193.jpg	
	03/07/2007	SP_A0198.jpg	
	29/06/2007	SP_A0163.jpg	
	04/07/2007	SP_A0209.jpg	
	21/06/2007	SP_A0138.jpg	
	21/07/2007	SP_A0147.jpg	
	04/07/2007	SP_A0203.jpg	
	21/06/2007	SP_A0149.jpg	
	03/07/2007	SP_A0200.jpg	
	29/06/2007	SP_A0160.jpg	
	29/07/2007	SP_A0158.jpg	
	29/07/2007	SP_A0043.jpg	
	29/07/2007	SP_A0156.jpg	
	21/06/2007	SP_A0140.jpg	
	21/06/2007	SP_A0026.jpg	
	21/06/2007	SP_A0142.jpg	

C:\Documents and Settings\W. ABUTO\My Documents\SP\_A0208.jpg

SP\_A0191.jpg

SP\_A0206.jpg

SP\_A0180.jpg

SP\_A0179.jpg

SP\_A0188.jpg

SP\_A0186.jpg

SP\_A0178.jpg

SP\_A0152.jpg

SP\_A0194.jpg

SP\_A0197.jpg

SP\_A0164.jpg

SP\_A0210.jpg

SP\_A0018.jpg

SP\_A0148.jpg

SP\_A0204.jpg

SP\_A0150.jpg

SP\_A0201.jpg

SP\_A0161.jpg

SP\_A0158.jpg

SP\_A0040.jpg

SP\_A0157.jpg

SP\_A0022.jpg

SP\_A0027.jpg

SP\_A0143.jpg

|Specific Localit

<u>Benchmark Id</u>	<u>Benchmark Name</u>	<u>lis</u>	<u>Geodetic</u>	<u>cool</u>	<u>Northings</u>
					9857699
					9856636
3V/6			•		9856151
4IV/10					9850940
5 IN/11			•		9860934
6MT3			•		9864552
7LX1/7			~	r r	9865101
8 LXI/4			T	J	9861151
9KISM7			•		9872070
10148ST5			0		9835375.07

<u>Eastings</u>	<u>order</u>	<u>Coordinate Units</u>
242367		1 Meters
250711		1 Meters
257729		1 Meters
265543		1 Meters
259842		1 Meters
264234		1 Meters
269945		1 Meters
264177		1 Meters
264177		1 Meters
234861.94		2 Meters



Resolution_For	Height_Data	Monument_type	elevation (metre)	Leveling_method	Projection	Buffer	Zone_number	VMapSheet_no
0.001	Iron_pin_	n_conc	1894.686	precise levelling	UTM	6.096	37	GL/5A(921)
0.001	Iron_pin_	n_conc	1794.618	precise levelling	UTM	6.096	37	GL/4A/6
0.001	Iron_pin_	n_conc	1661.835	precise levelling	UTM	6.096	37	GLA/5A(952)
0.001	Iron_pin_	n_conc	1636.665	precise levelling	UTM	6.096	37	GL/18A/6
0.001	Iron_pin_	n_conc	1649.843	precise levelling	UTM	6.096	37	GL/4A/6
0.001	Iron_pin_	n_conc	1611.336	precise levelling	UTM	6.096	37-	
0.001	Iron_pin_	n_conc	1533.415	precise levelling	UTM	6.096	37	GL/18A/6
0.001	Iron_pin_	n_conc	1596.98	precise levelling	UTM	6.096	37	GL/18A/6
0.001	Iron_pin_	n_conc	1590.489	precise levelling	UTM	6.096	37	
0.001	Iron_pin_	n_conc	1751.642	precise levelling	UTM	6.096	37	

Datum	Date_Establish^	Status	Status Date	VstnPicture 1
Arc60		Intact	29/06/2007	C:\Documents and Settings\W. ABUTO\My Documents\SP_A0162.
Arc60		Intact	29/06/2007	SP_A0167.jpg
Arc60		Intact	29/06/2007	SP_A0154.jpg
Arc60		Intact	07/07/2007	SP_A0171.jpg
Arc60		Intact	05/07/2007	SP_A0221.jpg
Arc60		Intact	05/07/2007	SP_A0212.jpg
Arc60		Intact	02/07/2007	SP_A0175.jpg
Arc60		Intact	02/07/2007	SP_A0169.jpg
Arc60		Intact	02/07/2007	SP_A0183.jpg
Arc60		Intact	02/07/2007	SP_207.jpg

VstnPicture_2	Mapimage	comps_Id
C:\Documents and Settings\W. ABUTO\My Documents\SP_A0219	C:\Documents and Settings\W. ABUTO\My Documents\SP_A0219	
SP_A0168.jpg	2.jpg	4
SP_A0155.jpg	1.jpg	1
SP_A0172.jpg	10.jpg	4
SP_A0220.jpg	2.jpg	3
SP_A0213.jpg	NOT FOUND	1
SP_A0176.jpg	21.jpg	3
SP_A0170.jpg	^i-.jpg	3
SP_A0184.jpg	NOT FOUND	2
SP_A208.jpg	SP_A208.jpq	4
		4

Mapsheetid | Shape

7 Point

1 Point

4 Point

5 Point

1 Point

99 Point

6 Point

6 Point

99 Point

99 Point

Benchmark

OA IQ2.I-2.00a

<u>Mapsheetsid</u>	<u>Mapsheets Name</u>	<u>Sheet Type</u>	<u>Scale</u>
1	D_SN125_2	Leveling_sheet	Not to scale
2	D_SN125_66	Leveling_sheet	Not to scale
3	D_SN125_3	Leveling_sheet	Not to scale
4	D_SN125_1	Leveling_sheet	Not to scale
5	D_SN125_10	Leveling_sheet	Not to scale
6	DN_SN125_21	Leveling_sheet	Not to scale
7	DN_SN125_43	Leveling_sheet	Not to scale
8	D_SN125_65	Leveling_sheet	Not to scale

V\_mapsheet

01/02/2008

I Date Published [ Mapimage

01/01/19502.jpg

01/01/1950 66.jpg

01/01/1950 3.jpg

01/01/19501.jpg

01/01/195010.jpg

01/01/1950 21 .jpg

01/01/1950 43.jpg

01/01/195065.jpg

## APPENDIX 11

HARD COPY VERSION OF THE DIGITAL MAP OF NAIROBI