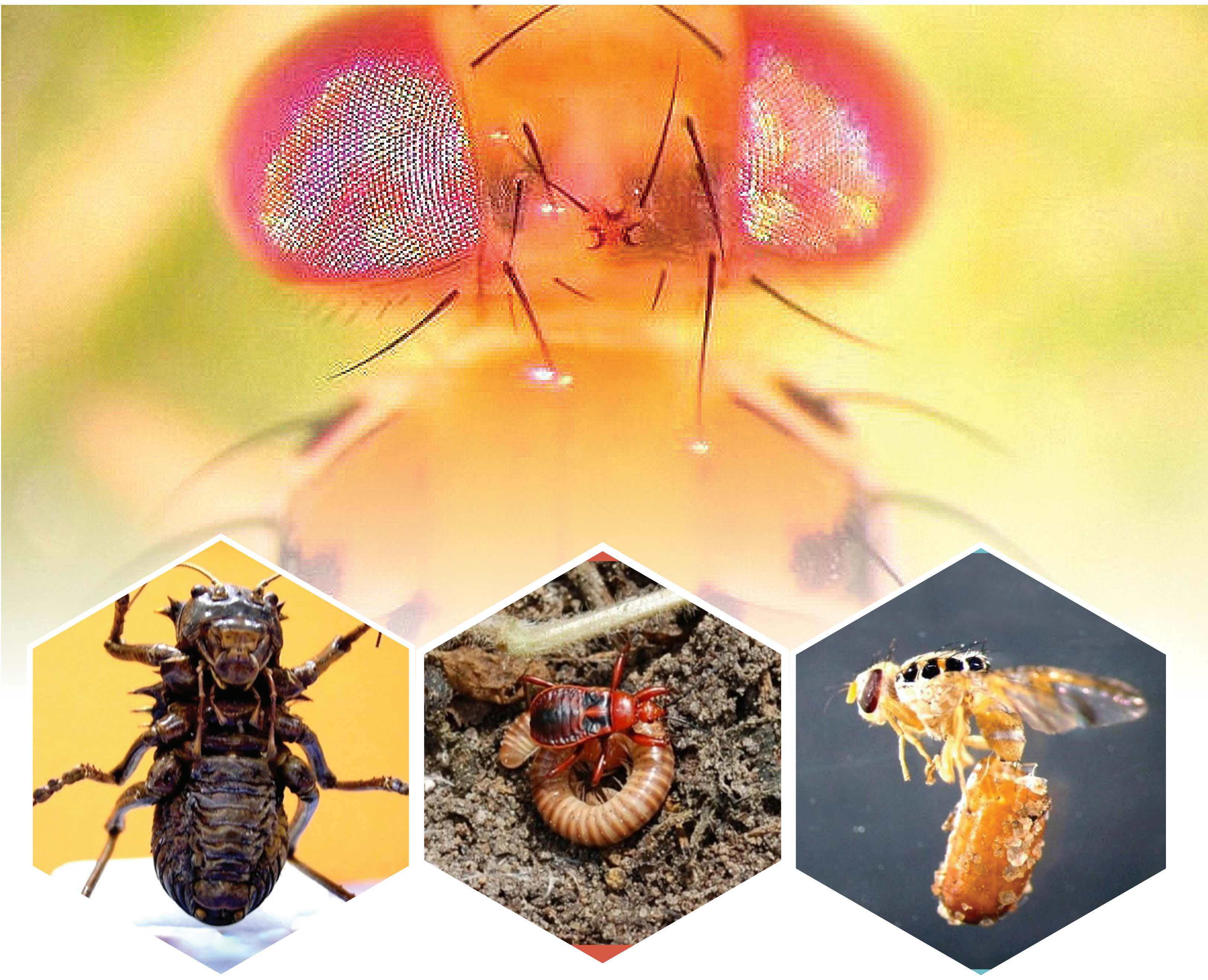
Volume 1, Number 2, December 2017

SCIENCE & DEVELOPMENT



CBAS
College of Basic and Applied Sciences
A Journal of the College of Basic and Applied Sciences (CBAS), University of Ghana

Automatic Satellite Dish Positioning for Line of Sight Communication using Bluetooth Technology

Robert A. Sowah[1*], Godfrey A. Mills[1], Joseph Y. Nortey[1], Stephen K. Armoo[1] and Seth Y. Fiawoo[1]

[1] Computer Engineering Department, Legon, University of Ghana

Corresponding author: rasowah@ug.edu.gh

ABSTRACT

Satellite dishes are used to receive beams of signals from satellites and other broadcasting sources which are then focused onto an antenna. The dish needs to be adjusted to get the desired azimuth and elevation for maximum signal reception. To overcome the difficulty of adjusting it manually, it would be beneficial to have a system that aligns the satellite receiver by mechanical means while allowing the user to interact with the system remotely to achieve a line of sight communication with the satellite source of interest. This paper proposes the design and development of a system which receives user specifications from an Android application via Bluetooth by either specifying the direction of orientation of the dish or selecting a satellite of interest. A control system interacting with the developed user interface achieves this. It employs a microcontroller, a GPS device, a compass and two servo motors to manage the orientation of the dish on its horizontal and vertical axes. The Smartphone utilizes its Bluetooth socket to communicate with the Bluetooth module interfaced to the microcontroller. A database containing information on available satellites is included in the Android application which is transferred to the microcontroller for computation of azimuth and elevation angles when the GPS coordinates and compass headings are obtained from their respective devices. Tests carried out showed positive results for control of the orientation of the satellite dish in various directions over a 50m radius. The automatic adjustment functionality provided precise direction for line of sight communication when users chose their satellite of focus.

Keywords: Line of Sight, Satellite Dish Positioning, Bluetooth technology, remote control and communication

Introduction

Satellites form an essential part of communication systems worldwide because they carry large amounts of data, telephone traffic and television signals. The use of satellites in everyday life is evident in the many homes and offices that are equipped with various forms of antennas which are used to receive signals from satellites located farther away from the earth. Satellites are positioned farther away from the earth because the gravitational pull of the earth is weaker at relatively higher levels and stronger at lower levels, so that communication satellites are usually mounted about 36,000 km away from the earth (Cheruku, 2010). Since about 42% of the earth's area is visible from a satellite, satellite communication has an edge over other means of communication (Vaneli-Corali, A. *et al.,* 2007). To prevent satellites fr--om swaying from their positioned orbits, some form of stabilization is required. Attitude control provides such stabilization through the orientation of the satellite with a system that keeps it moving evenly through its orbits (Galactics, 2016). With geostationary satellites, the satellite can receive or transmit messages to any transmitter or transceiver that is within a fixed geographical area visible to the satellite at all times.

Communication between observers on the earth and satellites in space is made possible by means of antennas. A satellite dish is a type of antenna designed to focus on a specific broadcast source which receives information by reflecting signal beams and focusing them into a relatively narrow beam that hits its parabolic surface and then passes the signal on to the feed horn, from where it is further transmitted to the receiving equipment (Nice & Harris, 2000).

The art of remote communication between users of devices and their corresponding systems is becoming more and more complex daily, with diverse ways such as electrical wires, component cables, storage media, computer buses, radio signals and infrared beams, and an even greater variety of connectors, plugs and protocols (Franklin & Layton, 2000).

Wired communication is used to describe any type of communication process that relies on the direct use of cables and wiring to transmit data. Examples of wire-based communication technologies include telephone networks and fiber-optic communication (Tatum, 2016).

This notwithstanding, wireless technologies have changed how people and devices everywhere in the world can correspond with each other, where user or device location is not expected to be fixed for communication, with a great migration from wired to wireless technologies in many instances of life (Dutta, 2015, Adewusi, 2000). .

Wireless communication technologies are of varying types, ranging from Infrared (IR) to Bluetooth. The distance between communicating devices is widely dependent on the type of wave involved in communication, which allows the transmission of data from within a few meters to thousands of kilometers. Bluetooth was developed in 1994 at Ericsson and operates in the unlicensed industrial, scientific and medical (ISM) band from 2.4 to 2.485 GHz, using a spread spectrum, frequency hopping, full-duplex signal at a rate of 1600 hops/sec. It is one popular method of wireless communication between devices (Rappaport, 2002). This method of data communication uses short-range radio links to replace cables between computers and their connected units (Sairam *et al.,* 2016). The power consumption associated with Bluetooth communication is minimal and its range of communication between devices ranges from 1 to 100 meters, depending on the class of the Bluetooth device (Rappaport, 2002; Franklin & Layton, 2000).

Many smart phones today can communicate using Bluetooth. This makes useful systems that need some form of wireless communication protocol with users, ranging from home automation to industries, offering substantial benefits to wireless network operators. Android, which is open-sourced and backed by Google, has statistics in its favour, indicating the increasing popularity of their devices on the market, providing more incentives to develop applications for Android over closed and PC operating systems (Jackson, 2011).

The system developed in this paper takes advantage of the fast-growing popularity of Android and Bluetooth technologies to develop an Android application that would use permission from users' Bluetooth.

With very little error in misalignment that could lead to a complete loss of a signal from a satellite source, the coordination of the dish is very instrumental to the strength of signal received from the transmitting satellite (Giangrandi, 2016). A dish must therefore be positioned at a particular elevation and azimuth to achieve the strongest possible signal. Users will be allowed to select a satellite dish from a predefined database with essential parameters for a two-dimensional automatic orientation of their satellite dishes, allowing the achievement of a line of sight communication with the broadcasting source.

Satellite Dish Positioning Systems

Various studies and researches have been done over the years on the design and development of satellite dish positioning systems through remote means. *Kyaw Oo et al.* developed Satellite Dish Positioning Control by DC Motor Using Infra-red (IR) Remote Control which sought to allow a user to position a satellite dish remotely through Infra- red (IR) communication (Me *et al.*, 2006). In their system, rotation of the dish was allowed on its horizontal axis. This consisted of a PIC 16F877A microcontroller to provide intelligence, and relay drivers and a DC motor to effect motion. The use of IR technology allowed a marginal distance between the user and the dish, promoting flexibility.

Commercial dish positioning systems have been developed to aid in positioning and general installation of satellite dishes. Infra-red (IR) communication has been the preferred choice of communication between the user and the satellite.

Edgefx Technologies designed and developed the Remote Alignment of 3D Dish Positioning by Android Application. Remote operation of the system was achieved by any device running the Android operating system with a Graphical User Interface based touch screen operation. Components employed included an 8051-series microcontroller, a Bluetooth device, motor drivers and DC motors.

Wilkinson and Swenson (1999) published a paper on the design of a satellite tracking station for remote operation and multi-user observation. The paper discusses the design of a satellite ground station at Utah State University to maintain multiple satellites by remote operation over the internet (Wilkinson & Swenson, 1999). This system allowed various users to remotely position the dish of the ground station via a Java interface to monitor low-orbiting satellites based on their locations in reference to the ground station. A server daemon design running on Sun Solaris 2 provided connections to the users via an internet connection to the hardware drivers. This hardware part of the system was made up of a Tattletale 8 microcontroller for receiving and computing position information of the satellite dish, a UNIX workstation for tracking station users over an internet connection, optical encoders for the relative positions of azimuth and elevation, and two DC motors for providing motion to the dish.

Voormansik (2009) carried out satellite signal strength measurements using the International Space University Ground Station and the University of Tartu Ground Station for satellites in Low Earth Orbits (LEO) (Voormansik, 2009). This required an antenna that points to the orbits in focus and an operator that has access to the ground station to communicate with its satellite to measure satellite signal strengths.

Rafael *et al.* published a development of an automated system for maneuvering parabolic dish antennas used in satellite communication (Rafael, Gonçalves & Leite do Prado, 2012). In this work, the key steps required to manually maneuver an antenna dish were automated. A 3.2 meter diameter antenna dish, a digital satellite receiver and a GPS were employed to capture signals from the satellite source and focus them onto the antenna, decode the channels of the satellite and determine the position of the antenna respectively. User-guided-interfaces developed with Java programming language for use on a computer were also provided that offered information about the movement of the antenna toward the reference position, the spatial position of the antenna and the carrier-to-noise ratio. After the antenna reached the reference position, the system monitored the quality of the reception through the carrier-to-noise ratio obtained, with a minimum value of 8 dB guaranteeing good reception (Ha, 1986).

Based on the reviewed literature in consonance with the need in developing countries, we proposed a cost-effective Satellite Positioning System with added functionalities. The system adapts the portable nature of Android devices and the growing market for their Bluetooth communication capability to provide an interface for users' interaction with their systems for remote orientation of the satellite dish for line of sight communication. An on-board unit made up of a microcontroller, a GPS module, a Bluetooth module, a compass module and servo motors is assembled and attached to the support of the satellite dish receiver. The microcontroller receives parameters based on the user's choice of satellite from the Android device via Bluetooth communication. Computation for the desired azimuth and elevation angles is made when the compass module provides the horizontal angle of direction of the dish and the GPS module provides position and altitude information from the receiver to the microcontroller. The microcontroller then sends the processed values of azimuth and elevation to the motors to effect movement and orientation for maximum signal reception.

System Design and Development

System Architecture

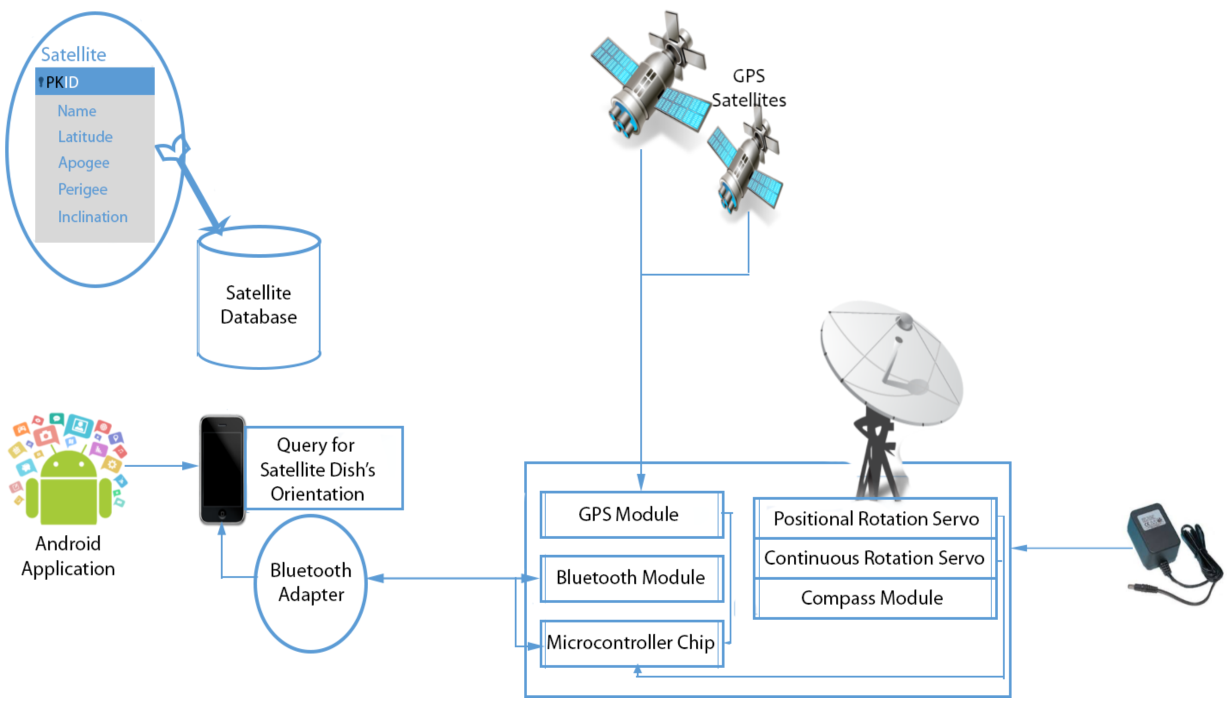


Fig. 1: System Design and Architecture

The overall system design and architecture is depicted in Figure 1 and shows the various developed hardware and software modules and their corresponding integration. It consists of an Android application running on a smartphone equipped with the Bluetooth adapter which interfaces with the hardware modules (Positional Rotation Servo, Continuous Rotation Servo and Compass)

System Modules

Android User Interface

The Android Application provides the interface by which users of the satellite dish interact with the dish. With Bluetooth technology, the application needs to access the Bluetooth adapter of the Android device, scan for available devices and allow pairing and a connection to occur on request. If the user is authorized to access the Bluetooth module of the dish, he/she will be allowed to activate an 'auto-orient' feature, whereby a satellite source will be chosen from a database for the system to automatically position itself. The user will also be allowed to change the orientation of the dish along its vertical and horizontal axes by pressing directional buttons corresponding to their respective commands.

The Android application has the following requirements:

1. Application should have a perceptive user interface.
2. Application should allow users to search for available Bluetooth devices, select their preferred device's corresponding module identity (satellite dish's Bluetooth module) and connect to it.
3. Users should go through an authorization stage where a password is verified for security reasons

before access to the control features of the application is granted.

4. Users should be able to align the dish to left, right, upward or downward positions using directional buttons.
5. Users should be able to select a preferred satellite from a list of satellites available in the application.
6. Users should be able to command the dish to position itself automatically through the application.
7. The flowchart for the operation of the Android application is depicted in Figure 2 (A).

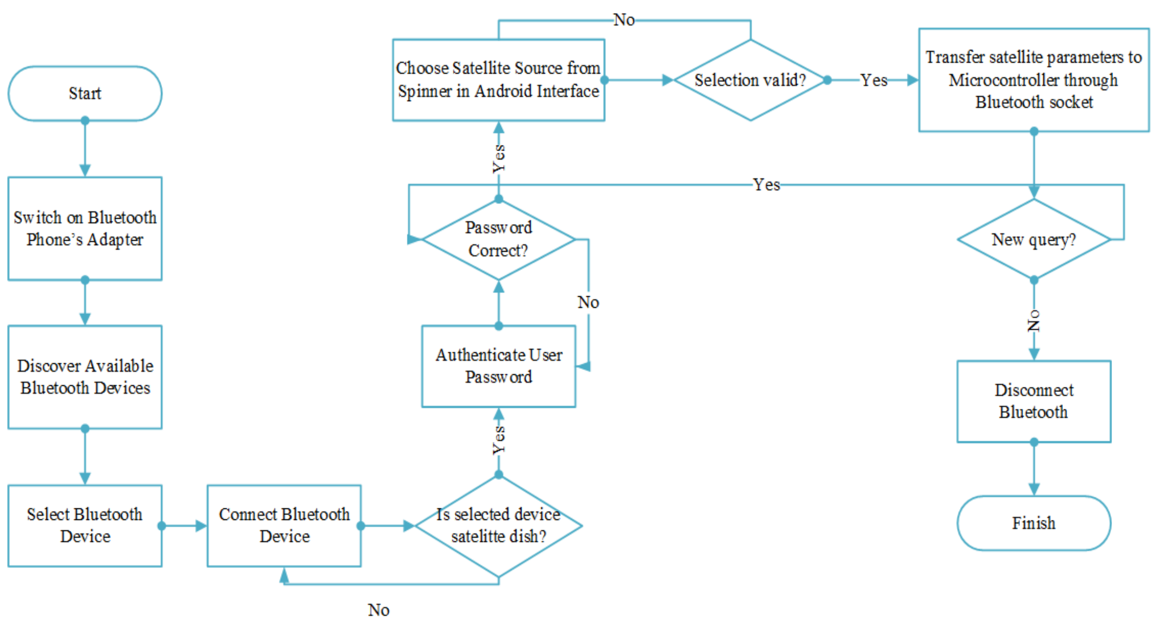


Fig. 2: A) Operation of Android Application

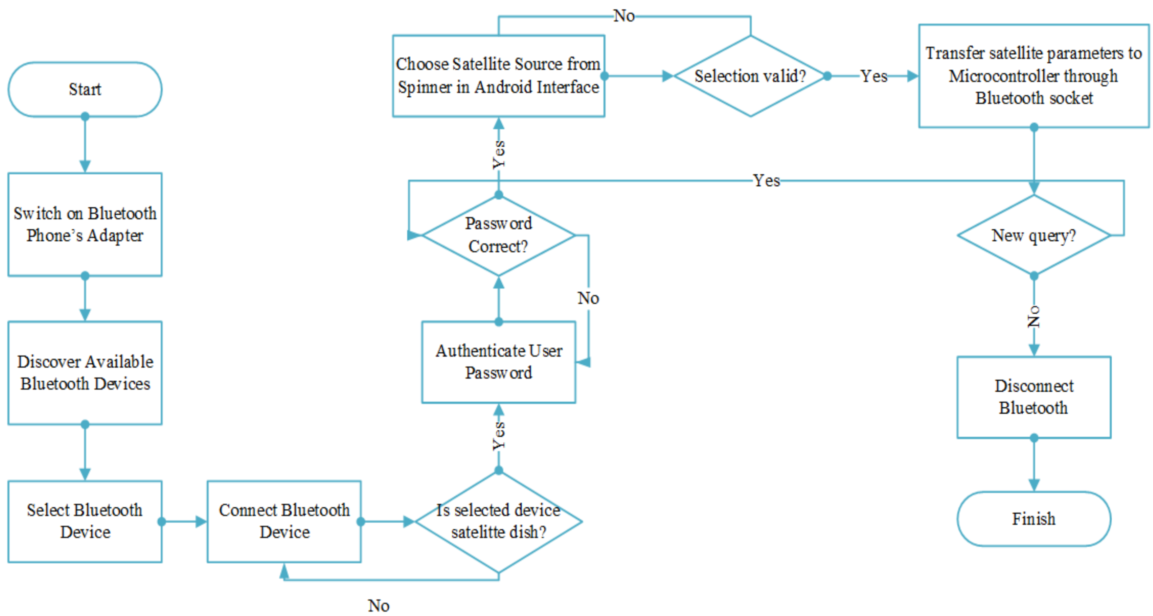


Fig. 2: B) Operation of Satellite Mount System

Satellite Database

The database is a predefined one and contains two tables: *sqlite_sequence* and *satellite Info*. The *sqlite_sequence* is an automatically generated table within the database which is used to implement the automatic increment functionality of the *satelliteInfo* ID attribute. The *satelliteInfo* table contains information such as the satellite's name, perigee, apogee, longitude and inclination. Upon selecting a desired satellite from a spinner in the developed Android interface, an activity is launched to ensure that the satellite information selected is retrieved from the database and then used to compute the corresponding elevation and azimuth of the satellite receiver dish for auto alignment.

Satellite Dish Mount System

The power supply unit is the essential subsystem which provides power to the various components of the entire system: the microcontroller, servos, GPS device, Bluetooth and compass modules. With the exception of the Bluetooth and compass modules which needed 3.3V, all other components and the microcontroller needed an average of 5V direct current supply.

The requirements of the subsystem comprising the mount are:

1. System should allow communication with the user by allowing an open and dedicated connection.
2. System should receive positional commands from the user and effect them with the help of the corresponding actuators.
3. System should compute and readily present GPS coordinates for computation at any given time.
4. System should provide the compass heading with respect to the earth's magnetic north to enable tracking of the azimuth of the satellite dish receiver.
5. The flowchart for the operation of the Satellite Mount module is depicted in Figure 2(B).

The microcontroller is the unit containing all the information that controls all the hardware components which are interfaced with it. The positional rotation servo is used to alter the elevation of the dish whilst the continuous rotation servo is used to allow change in azimuth of the dish. The Bluetooth module is responsible for the communication between the Android application and the dish where the commands are sent from the device and received by the module, then transferred serially to the microcontroller to effect corresponding commands.

The compass and GPS module are especially responsible for the auto-orientation functionality of the system. The compass is responsible for providing the heading information for the dish, allowing it to keep track of its azimuth in relation to the earth's magnetic north pole. The GPS module provides the position coordinates of the dish's location which are used to compute the elevation and azimuth of the dish with respect to that position. The SkyNav SKM53 Series was the preferred choice because of its capabilities in harsh GPS visibility environments and a tracking sensitivity of –165 dBm, which give it a wide extension of position coverage. It is also proficient in environments of temperature ranging from 40 to 85°C. Bluetooth modules come in a variety of classes - 1, 2 and 3 - which determine their range of operation. The class 1 Bluetooth module allows communication with other Bluetooth devices within a radius of 100m, which is the maximum area for Bluetooth devices. The HC-05 Bluetooth module was employed because of the wider range it provided at a cheaper cost. The HMC6352 compass module has multiple operating modes that balance the use of power with query acquisitions and rapid heading updates.

The scope of the system required a 2-dimensional rotation in the vertical and horizontal axes. This demanded that two different motors be utilized. The elevation angle was required to be controllable from 0° to 90° and the azimuth controllable from 0° to 360°. Continuous rotation servos do not provide feedback to the system; nonetheless, they allow a maximum degree of rotation about the pivot. Positional rotation servos allow rotation from 0° to 180° with feedback to the system, which is a much preferable option for control systems. Since elevation positioning did not require a full range of operation, a positional rotation servo was required for that functionality, with limited motion within a 90° range and a continuous rotation servo for the azimuth positioning functionality. The HS-133 and the HSR1425-CR which are positional and continuous rotation servos respectively were selected.

System Design

A flow diagram showing the operation of the system is shown below:

A flow diagram showing the overall operation of the system is shown below in Figure 3. When the system starts, it selects the satellite Bluetooth device and connects it. Querying the satellite dish for new satellite position and aligning them to user's specified choices and customizations. After that operation, it finishes the setup and waits for new user preferences by going through the same cyle.

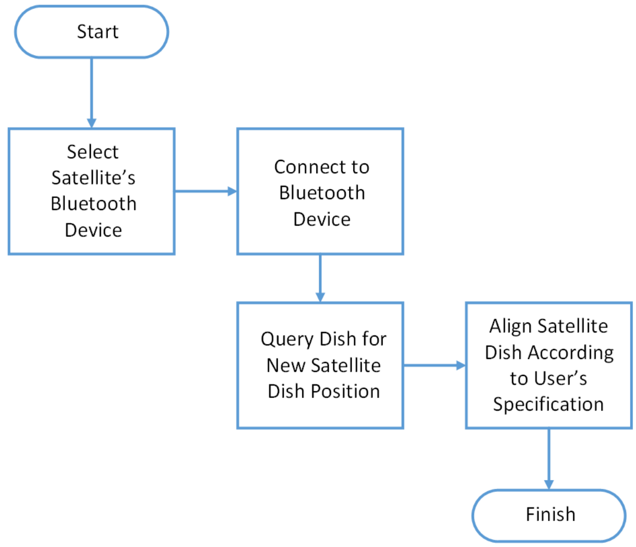


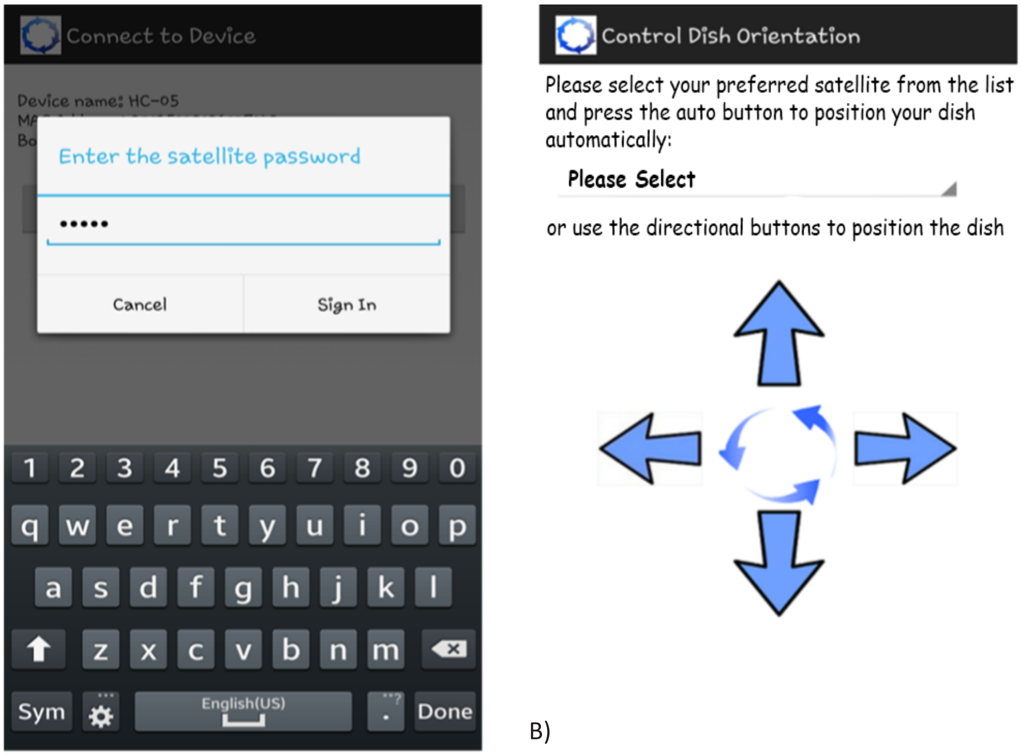
Fig. 3: Flow Chart for System Operation

System Implementation and Testing

The implementation of the system involved the development of the Android application, creating and populating the database containing satellite information, building the power supply, programming of individual components and the integration and testing of the entire system, the components of which are taken singly as incremental modules. The various implementation processes of the subsystems comprising the overall system are discussed in the following subsections.

Android Application Development

The layout interface of an Android application was defined in an XML file instead of the particular *.java* file implementing that activity. This was done to make it more manageable. The function of the main activity of the application is to allow a connection to other classes within the Android project. To enable discovery of available Bluetooth devices, a class within the Android project is first called with the option given for a user to choose one of the devices to connect to. This activity automatically checks the state of the Bluetooth adapter on the device. If the device does not have a Bluetooth adapter, it means that the application cannot locate and control other Bluetooth devices and the application closes; otherwise, the Bluetooth adapter is automatically switched on and allowed to discover other Bluetooth devices in its radius of view. On selection of an available Bluetooth device, its name, its MAC address and bond status are saved. With the MAC address' correspondence with the satellite dish's Bluetooth module, authentication of user's access is made by means of a password before adjustment functionality of the dish is allowed. A *.java* class is responsible for the mutator (setter) and accessor (getter) methods of the satellite parameters from the database. The mutator method is used to set the values of the records in the database during insertion at the initialization of the application and the accessor method is used to retrieve queried records. The developed interface for system user authentication and the Android's application activity control is shown in Fig. 4A and 4B below.



A) B)

Fig. 4: A) User Authentication Dialog Interface and B) Android's Application Control Activity

Development of Satellite Database in SQLite

Creating and using databases in Android involves one of two approaches: dynamically creating the database at runtime or having a predefined database with data already stored in it. The first approach is of great use in applications that involve storing data for applications which could be constantly updated by a user. The second approach is essential when the data to be used in the application is known and of a large size, which might otherwise slow down the application in case of access of data items from the database, were the earlier approach used. Since the database of satellite information required numerous records which had predefined parameters, the second option was brilliant. The *SQLite*Database created was moved to the assets folder of the Android project which copies it into the system database path of the application. The *SQLiteDatabase* API then opens and accesses it normally.

Calculation of Elevation and Azimuth Angles

The angles for the elevation and azimuth of any satellite dish receiver are dependent on the geographical coordinates and altitude of the dish relative to that of the satellite in terms of its sub-satellite point and height. In the computation of the angle of elevation of the satellite dish in relation to the satellite in focus, triangular trigonometry was used. The assumption made was that the earth is a perfect sphere with an origin (0,0,0) and an even radius, R=6371000 m.

For the computation of distance *d*, the user coordinates and sub-satellite points are converted to Cartesian coordinates *(x,y,z)* and the straight line distance (SLD) computed between them for the calculation of angle *e* from the triangle obtained.

Assuming the sub-satellite point position, $pos(\theta_1, \varnothing_1)$ and the coordinates of the user position, $pos(\theta_2, \varnothing_2)$,

$$xpos_1 = R\cos(\theta_1)\cos(\varnothing_1) \quad (1)$$

$$ypos_1 = R\cos(\theta_1)\sin(\varnothing_1) \quad (2)$$

$$zpos_1 = R\sin(\theta_1) \quad (3)$$

Similarly,

$$xpos_2 = R\cos(\theta_2)\cos(\varnothing_2) \quad (4)$$

$$ypos_2 = R\cos(\theta_2)\sin(\varnothing_2) \quad (5)$$

$$zpos_2 = R\sin(\theta_2) \quad (6)$$

The SLD between them is computed as

$$SLD = \sqrt{(x_{pos_1} - x_{pos_2})^2 + (y_{pos_1} - y_{pos_2})^2 + (z_{pos_1} - z_{pos_2})^2} \quad (7)$$

$$where\ -90° \le \theta \le 90° (latitude)$$
$$and\quad 0 \le \emptyset \le 360\ (longitude)$$

With the SLD computed, the cosine rule is again used to find the angle *e*, which is employed to find the angle *r*.

From the cosine triangle,

$$(R + h_s)^2 = (R + h_a)^2 + d^2 - 2(R + h_a)(d)\cos(r) \quad (10)$$

$$r = \cos^{-1}\left(\frac{(R + h_a)^2 + d^2 - (R + h_a)^2}{2(R + h_a)(d)}\right) \quad (11)$$

$$ElevationAngle = r - 90°$$

$$whereR = Radius\ of\ the\ Earth$$

$$h_a = Altitude\ of\ satellite\ dish$$

$$h_s = Altitude\ of\ satellite$$

From spherical trigonometry, with a spherical triangle drawn on the surface of the earth;

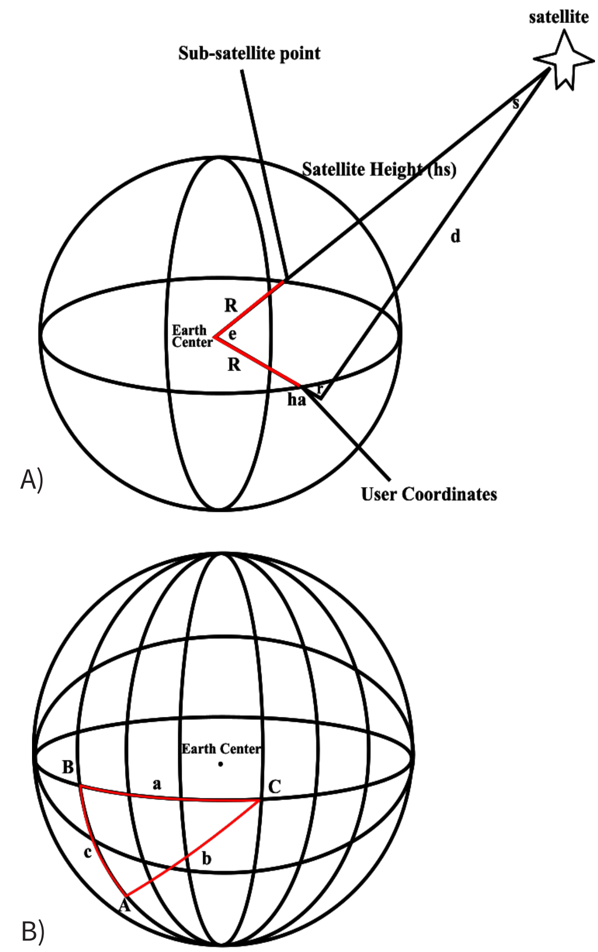


Fig. 5: Calculation of A) Azimuth and B) Elevation Angle

Azimuth refers to the rotation of the whole antenna around a vertical axis.

From the laws of cosines,

$$\cos(b) = \cos(a)\cos(c) + \sin(a)\sin(c)\cos(B) \quad (12)$$

$$where\ B = lon_2 - lon_1$$

$$and\ c = 90° - lat_1$$

$$and\ a = 90° - lat_2$$

$$b = \cos^{-1}(\cos(a)\cos(c) + \sin(a)\sin(c)\cos(B))$$

From the sine rule,

$$\frac{\sin(A)}{a} = \frac{\sin(B)}{b} = \frac{\sin(C)}{c} \tag{13}$$

$$\Rightarrow A = \sin^{-1}(a\frac{\sin(B)}{b})$$

Programming of Individual Hardware Components

The Bluetooth module was first set in AT mode to configure some modifications to selected default properties of the module, such as its name, for easy identification by the satellite dish receiver and a baud rate of 9600 for communication with the microcontroller. It was then programmed to receive ASCII characters from the Bluetooth device of the Android mobile. Available ASCII were now converted to characters in the microcontroller with *switch()* statements implementing their respective conditions where 'a' meant an automatic positioning, '*u*' referred to an upward signal, '*d*' to a downward signal, '*l*' to movement in the left direction and '*r*' to movement towards the right. Since the microcontroller received characters from the Bluetooth socket of the mobile device serially, there was the need for concatenation and then conversion into floating point variables corresponding to the height and longitudes of selected satellites.

With the positional servos, the shaft producing the output rotates 180º both clockwise and anti-clockwise, with stops placed in the gear mechanism to prevent turning beyond these limits. Given their limited range of rotation, it was preferred that they be used in the control of the elevation of the satellite dish. The servo was programmed to respond to the commands corresponding to upwards and downwards movements. In cases where the maximum elevation is reached, the servo motor does not respond to upward commands; the same applies to downward commands for minimum elevation.

Since continuous rotation servos allow rotation in either direction indefinitely, they were the preferred choice for rotation along the horizontal axis. With continuous rotation servos, the control signal sets the speed and direction of the servo— either clockwise or anticlockwise— rather than its position. When positional rotation servos are instructed to move, they will move to the position and hold it. The *servo.writeMicroseconds()* function was used to control the servo by pulse width modulation (PWM). A value greater than 1500 allowed the servo to move in the clockwise direction whilst a value less than that allowed the servo to be moved in the opposite direction. With 1500 in the function, the servo's neutral position is achieved. A delay was introduced after each movement to determine the duration of the pulse, determining the angle of rotation. Since continuous rotation servos do not have feedback control, a count variable was used to keep track of the azimuth of the dish, which had a maximum value of 359 and a minimum of 0 with a modulus operand.

The GPS module was required to provide the microcontroller with three parameters for the computation of the dish's azimuth and elevation - the latitude, longitude and height of the dish. Due to the limited capability of the ports of Arduino Mega to support interrupts, only specific ports had the capacity to perform serial reception of data. The TX and RX pins of the GPS module were therefore assigned to ports 51 and 53 of the Arduino board respectively. The GPS module was assigned a baud rate of 9600. Functions defined in the *TinyGPS* header were used in the code to obtain the current latitude, longitude and altitude of the dish.

The compass module had the basic task of tracking the heading of the satellite dish in relation to the Earth's magnetic north. It was required that the RX and TX pins of the compass module be connected to the data (SDA) and clock (SCL) lines (20 and 21 respectively) of the Arduino board.

Hardware and System Integration

With the testing of the functionalities of the individual components completed, it was useful to integrate the individual components to form the hardware subsystem.

Embedded Unit Circuitry Test

The testing of the unit circuitry was done to ensure that it met the voltage and current requirements of each component and the entire unit. Supplying a lesser voltage will cause certain components of the system not to respond to certain query signals. With the 5V 65mA power requirement not met, GPS transmission was not possible. The compass demanded voltage levels between 2.7 and 5.2 V. The Bluetooth module ideally demanded voltage of 3.3V but had allowance and worked normally up to 6V. The microcontroller board requires a voltage of 5V, while there was the possibility of frying the chip with voltages in excess of that. The unit employed a power circuit which rectified the 220V AC supply to ~11.7V DC. Due to power fluctuations during the testing phase, a 5V DC regulator was employed to create the expectation that the output of the DC regulator will always be in the region of 5V. Tests conducted produced the same voltage at the output of the DC regulator.

System Testing and Results

Once the hardware integration and power requirements tests were complete, it was necessary to test the entire hardware system with the Android application. The Android application was installed and started on a Samsung Galaxy Note 3 SM-N9005 which required user authorization for the application to access and turn on the system's Bluetooth adapter (if previously off) and begin discovery of available Bluetooth devices. On selection of the satellite dish's Bluetooth device from the list, the option was given for pairing and connecting to it with a successful authentication. A blank field or wrong password entry presented an alert and prevented further use of the application by the user till a correct entry was made. With approved credentials, access was gained to the Android activity that allowed the control of the satellite dish receiver.

The directional buttons were first tested to ensure their correspondence to the various directions they represented. In the initialization state, the elevation of the dish is set to 0° - in parallel with the horizontal, with no response to input indicating further movement downwards. The button with an arrow moving upwards produced a corresponding shift in position, increasing the elevation from its original position to a maximum elevation angle of 90°. The buttons which issued commands to allow for clockwise and anticlockwise rotation allowed unlimited change in azimuth angles in both directions.

When the auto-rotation functionality of the application was triggered without the selection of any satellite source, a prompt issued an alert that a broadcasting source be selected for that feature to execute. Various satellite sources were selected for the auto-rotation functionality of the application to transmit satellite information to the microcontroller for computation and allow orientation of the dish in computed values of azimuth and elevation. The main aim of this test is to measure the difference in the values obtained from the system azimuth and elevation calculations and alignment in contrast to the expected values. The automatic alignment process involves the following stages: parameters of satellite selected by user are received by Bluetooth module and sent to microcontroller; microcontroller performs computation for azimuth and elevation and rounds it off to the nearest integer; system positions itself in accordance to resulting computation as shown in Figure 6. Since the servo motors only allowed rotational changes with no decimal places, there was a little allowance for error.

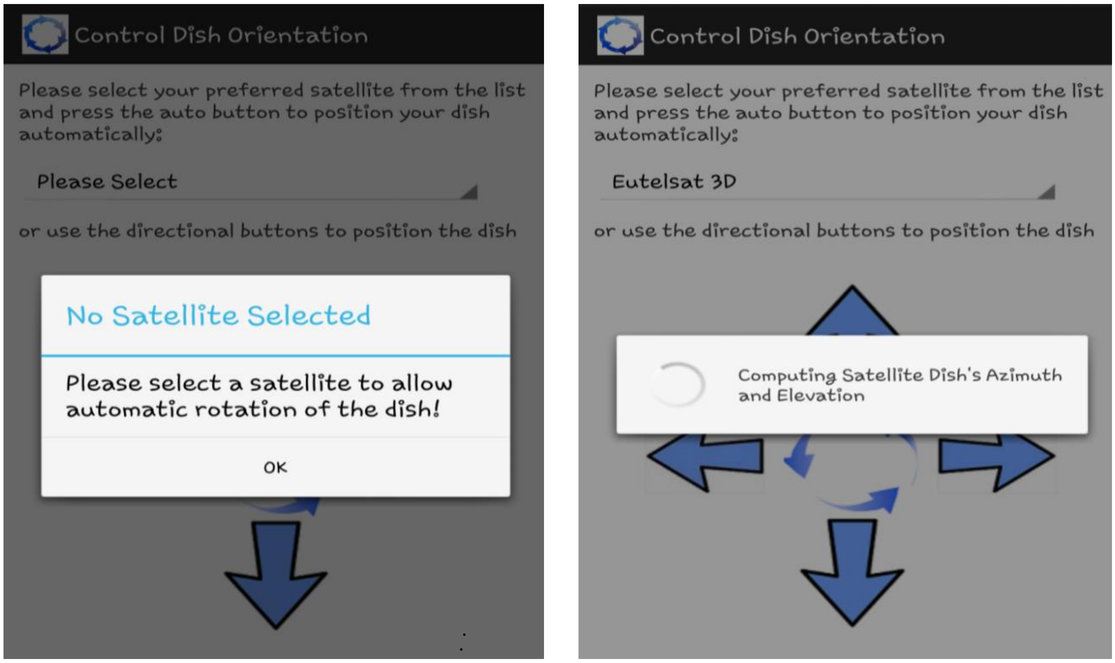


Fig. 6: Results of Android interface testing for satellite dish positioning system

Table 1: Comparison between Expected and System's Azimuth and Elevation for 7 different satellites

Expected Azimuth	265.41	102.64	138.43	101.31	108.72	100.19	112.78
System's Azimuth	264	102	138	102	108	100	112
Expected Elevation	31.58	61.57	81.14	58.76	69.91	55.96	73.17
System's Elevation	32	62	81	59	70	56	73

Table 1 above shows the results of tests with 7 satellites selected showed an error margin ranging from -0.0068% to 0.00692%, which is satisfactory as satellite dish receivers will maintain a line of sight communication with sources from the computation. Satellites sources not in the range of view of the satellite dish were also selected for testing with a negative response from the system.

Conclusion

The design procedures for both the hardware and software components were realized. The developed system was successfully tested and deployed. After successful completion of this research and subsequent deployment, the following benefits are realized by users of the system: (1) ease of control of system (2) remote control of system (3) automatic alignment for line of sight communication with satellite source.

Acknowledgements

The authors would like to acknowledge the technical and nurturing environment provided by the staff and faculty of the Department of Computer Engineering, University of Ghana.

References

A Look at the Basics of Bluetooth Technology. (2015, April 12). BluetoothFacts. Retrieved from https://www.bluetooth.com/Pages/Basics.aspx

Adewusi A. A., (2000, May). *Wireless communication: Today's Need Tomorrow's Promise*. Paper presented at the meeting of Electrotechnical Conference, Lemesos

Cheruku, D. (2010). *Satellite communication*. New Delhi: I.K. International Publishing House.

Dutta, P. (2015). *What Are the Advantages of Wireless Communication? | eHow*. *eHow*. Retrieved 11 April 2015, from http://www.ehow.com/facts_4885908_what-advantages-wireless-communication.html

Fast Facts. *FastFacts*. (2015). Retrieved 12 April 2015, from https://www.bluetooth.com/Pages/Fast-Facts.aspx

Franklin, C. & Layton, J. (2000). *How Bluetooth Works*. *HowStuffWorks*. Retrieved 26 March 2015, from http://electronics.howstuffworks.com/bluetooth.htm

Galactics. (2016). *Satellite antenna bearing calculator*. *Giangrandi.ch*. Retrieved 22 April 2015, from http://www.giangrandi.ch/electronics/satcalc/satcalc.shtml

Ha, T. (1986). *Digital satellite communications*. New York: Macmillan Publishers.

Jackson, W. (2011). *Android apps for absolute beginners*. [Berkeley, California.]: Apress Publishers.

Me, O., Chaw, N., & Hla, T. (2006). Satellite Dish Positioning Control by DC Motor Using IR Remote Control. *International Journal of Electronics and Computer Science Engineering*, 3, 119-207.

Nice, K. & Harris, T. (2002). *How Satellite TV Works*. *HowStuffWorks*. Retrieved 21 October 2014, from http://electronics.howstuffworks.com/satellite-tv.htm

Rafael, C., Gonçalves, B., & Leite do Prado, P. (2012). *Development of an Automated System for Maneuvering Parabolic Dish Antennas Used in Satellite Communication*. Paper presented at the meeting of ABCM Symposium Series in Mechatronics, 5, 69-78.

Rappaport, T.S. (2002). *Wireless Communications: Principles and Practice, Edition 2. Upper Saddle River, New Jersey. Prentice Hall Publisher.*

Satellites - Attitude Control: Overview. (2016). *Satellites. spacesim.org*. Retrieved 26 March 2015, from http://satellites.spacesim.org/english/anatomy/attitude/index.html

Satellite Communication. (2007). Retrieved 17 December 2014, from http://www.scatmag.com/technical/Satellite%20Communication_aug07.pdf

Sairam, K., Gunasekaran, N., & Redd, S. (2002). Bluetooth in Wireless Communication. *IEEE Communications Magazine*, (6), 90-96.

Tatum, M. (2016). *What are Wired Communications? (with pictures)*. *wiseGEEK*. Retrieved 11 April 2015, from http://www.wisegeek.com/what-are-wired-communications.htm

Voormansik, K. (2009). *Satellite Signal Strength Measurements with the International Space University Ground Station and the University of Tartu Ground Station* (Master's Thesis). University of Tartu, Faculty of Science and Technology, Institute of Physics.

Wilkinson, M., & Swenson, C. (1999). *Design of a Satellite Tracking Station for Remote Operation and Multi-User Observation*. Paper presented at the meeting of 13th Annual AIAA/USU Conference on Small Satellites, IIb, 1-16.-

BIOLOGICAL SCIENCES

2 **Morphometric Studies of the Sweet Potato Weevil, *Cylas* Species-Complex In Southern Ghana**

Maxwell K. Billah, Ayaovi Agbessenou, David D. Wilson, Wouter Dekoninck and Carl Vangestel

19 **Cocoa Pod Husk Plus Enzymes is a Potential Feed Ingredient for Hy-Line Silver Brown Laying Hens**

Thomas N. Nortey, Dorinda V. Kpogo, Augustine Naazie and Emmanuel O. K. Oddoye

PHYSICAL SCIENCES

31 **Petrogenetic Evolution of the Eastern Buem Volcanic Rocks, South-Eastern Ghana**

Naa A. Agra , Daniel Kwayisi, Prince O. Amponsah, Samuel B. Dampare, Daniel Asiedu and Prosper M. Nude

50 **Characterization of Rock Samples from Yale Area of the Upper East Region of Ghana**

Samuel A. Atarah and Gabriel K. Atule

60 **Numerical Analysis of Graphene Cladded Optical Fibre**

Ferdinand A. Katsriku, Grace G. Yamoah and J-D Abdulai

72 **Performance Evaluation of Chromatic Dispersion Compensation Techniques in Single Mode Fibre for Radio over Fibre Applications**

Isaac Dankwa, Ferdinand A. Katsriku, Grace G. Yamoah and J-D Abdulai

85 **Automatic Satellite Dish Positioning for Line of Sight Communication Using Bluetooth Technology**

Robert A. Sowah, Godfrey A. Mills, Joseph Y. Nortey, Stephen K. Armoo and Seth Y. Fiawoo

INTEGRI PROCEDAMUS

CBAS
College of Basic and Applied Sciences
University of Ghana

DigiBooks

ISSN 2550-3421
9 772550 342008