Measuring human body orientation for analyzing posture in yoga

Submitted By –

Arihant Jain

Ashish Kumar Singh

Samanth Hosadurga Chinivar

Contents:

Abstract: The Orientation of Human Body plays a very important role in fields such as sports, yoga, medical etc. Proper orientation of the body is required to do specific tasks. For Example: A budding tennis player should have the correct knowledge of how is hand position should be while playing a forehand or a backhand shot. Similarly, when a person is doing Yoga, he/she needs to do it correctly, knowing the exact body orientation.

In this project, we have decided to build a system or a network which can help to detect the orientation of the body for various applications. We have chosen a posture of yoga and implemented our design to perform the task. We have built this using Arduino Microcontroller , accelerometers, gyrometers(for sensing the movement of the body) and a Zig bee network. The purpose of Zig bee network would be to establish a communication network between the nodes where the sensors are connected and the main node which is connected to the computer. The microcontroller would process the information and display in the computer. So through this we would get the coordinate axis representation of various nodes and hence we could get the body orientation information.

Introduction: The developed countries use the technology of measuring correct body orientation in various fields such as sports, medicine , physical therapy etc, to maximize the output of a person in this field. Determining the proper flow of energy for the corresponding orientation of body and optimizing it, technical analysis can be done for various body postures, actions and practicing these orientations can maximize the output.

We have proposed a low cost, robust system that can be used in yoga to detect various yoga positions, in sports for stroke and technique analysis like in cricket, tennis (serve analysis), Badminton, Javelin Throw etc., for military purposes like proper positioning of hand and body while shooting, for determination of . In all these system we first determine the orientation of object which may be body or any other device.

In our project we are designing and fabricating very small device which can be used for such varied applications. We will be showing the analysis of a particular Yoga position. The device uses an Inertial Measurement Unit(9 DOF) to measure the relative orientation of the various body parts.The Graphical User Interface(GUI) that is designed in Matlab will be open source and user friendly so that any person can understand the amount of energy flow (determined by amount of impact) with respect to corresponding orientation of its Body with respect to standard coordinate system.

1.1 Objective :

To develop and fabricate a Data Acquisition Unit which will wirelessly transmit real time sensors data using Zigbee to PC for analysis.

To analyze the tilt angle of various body parts such as trunk, shoulder, elbow, wrist in the particular yoga position using MATLAB GUI.

To determine the kinematic flow of energy through the body and also determine the acceleration, angular velocity and other parameters required to study the proper orientation.

1.2 Existing Technology :

The existing technology uses high definition cameras and slow motion video analysis. The thermal sensor technology and Marker technology are very efficient but it is expensive and is not durable.

Kinect Technology of Microsoft also uses high definition sensors but the processing is slow and only applicable for 2 dimensional.

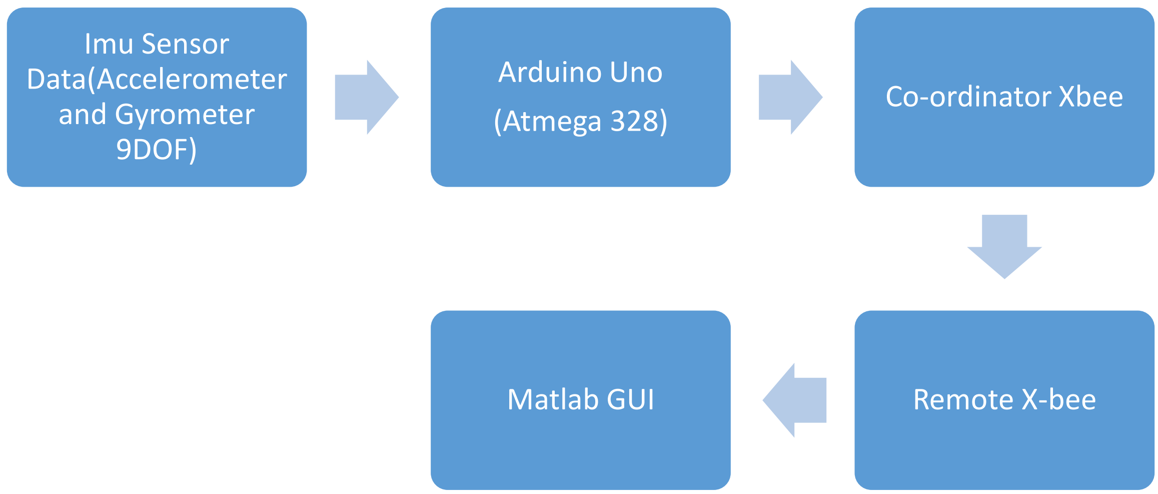
1.3 Proposed System :

We are using Inertial Measurement Units (IMU) as sensors and placing them at different parts of the body of a person performing Yoga for determining the correct orientation. The experiment would be performed on different persons and the corresponding graphs would be plotted. From the analysis of the data the position orientation of different people can be compared and the best possible orientation for this application can be suggested. It is envisaged that this application can provide feedback to the people to get the perfect orientation.

The analysis will be real time. For real time analysis different sensor nodes will be communicating using Zigbee attached with it to the central computer.

2.0 Hardware System:

2.1 Hardware Block Diagram:

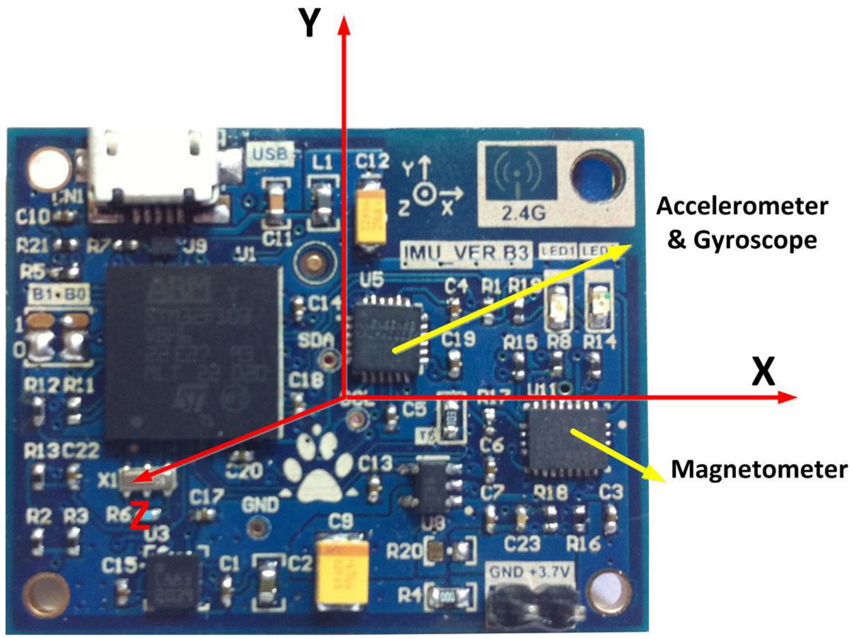


2.2 Overview of the components:

1. Inertial Measurement Unit: It comprises of Accelerometer, Gyroscope and Magnetometer. We are using BNO 055 which is a 9 degree of freedom IMU.

Accelerometer: Measures proper acceleration, also called the four- acceleration. This proper acceleration is associated with the weight of a test mass. An accelerometer is a device that measures proper acceleration. The proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration (rate of change of velocity). Instead, the accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device.

Gyroscope: Measures or maintains orientation based on the principles of angular momentum. Gyroscope measure angular velocity, how fast something is spinning about an axis. Three selectable measurement scales, with rates up to 2000° per second. When trying to monitor the orientation of an object in motion, an accelerometer may not give you enough information to know exactly how it‘s oriented.



2. <u>XBEE Modules (XBEE S1):</u> ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on an IEEE 802.15 standard for personal area networks. ZigBee devices are often used in mesh network form to transmit data over longer distances, passing data through intermediate devices to reach more distant ones. This allows ZigBee networks to be formed ad-hoc, with no centralized control or high-power transmitter/receiver able to reach all of the devices. Any ZigBee device can be tasked with running the network.

ZigBee is targeted at applications that require a low data rate, long battery life, and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for periodic or intermittent data or a single signal transmission from a sensor or input device. Applications include wireless light switches, electrical meters with in-home-displays, traffic management systems, and other consumer and industrial equipment that requires short-range wireless transfer of data at relatively low rates. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs, such as Bluetooth or Wi-Fi.

3. Arduino Uno (ATMEGA 328): They are Atmel‘s micro controller having various features like USART, I2C, SPI, Analog to Digital converter, PWM etc. In this system, micro controller is used for interfacing the IMU and using I2C protocol for getting raw data from IMU and SPI protocol for SD card interface. It also uses USART for data transfer through ZigBee.

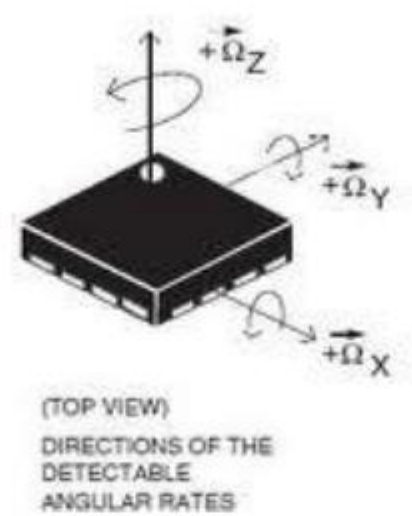
2.3 Sensor Employed:

2.3.1 BNO 055 IMU: The BNO 055 inertial Measurement Unit is a 9 degree of freedom sensor that measures acceleration forces and angular velocity of the object. It has three sensors in it as described above, the accelerometer, the gyroscope and the magnetometer. So it is a 9 degree of freedom sensor. The magnetometer gives the heading direction. Bosch is the first company to get this right by taking a MEMS accelerometer, magnetometer and gyroscope and putting them on a single die with a high speed ARM Cortex-M0 based processor to digest all the sensor data, abstract the sensor fusion and real time requirements away, and give out data in quaternions, Euler angles or Roll, pitch yaw.

The BNO 055 can output the following sensor data:

- Absolute Orientation (Euler Vector, 100 Hz): Three axis rotation data based on a 360 degree sphere.
- Absolute Quaternion (Quaternion, 100 Hz): Four point quaternion output for more accurate data manipulation.
- Angular Velocity Vector (100 Hz): Three axis of rotation speed in rad/s.
- Acceleration Vector(100 Hz) : Three axis of acceleration (gravity + linear motion) in m/s^2
- Magnetic Field Strength Vector (20 Hz): Three axis of magnetic field sensing in micro tesla (μT).
- Gravity Vector (100 Hz): Three axis of gravitational acceleration in m/s^2.
- Temperature (1 Hz) : Ambient temperature in degree Celsius.

Direction:



Electrical Datasheet Specification:

OPERATING CONDITIONS BNO055						
Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage (only Sensors)	V_{DD}	--	2.4	--	3.6	V
Supply Voltage (µC and I/O Domain)	V_{DDIO}	--	1.7	--	3.6	V
Voltage Input Low Level (UART, I2C)	V_{DDIO_VIL}	V_{DDIO} = 1.7-2.7V	--	--	0.25	V_{DDIO}
		V_{DDIO} = 2.7-3.6V	--	--	0.3	V_{DDIO}
Voltage Input High Level (UART, I2C)	V_{DDIO_VIH}	V_{DDIO} = 1.7-2.7V	0.7	--	--	V_{DDIO}
		V_{DDIO} = 2.7-3.6V	0.55	--	--	V_{DDIO}
Voltage Output Low Level (UART, I2C)	V_{DDIO_VOL}	V_{DDIO} > 3V , I_{OL} =20mA	--	0.1	0.2	V_{DDIO}
Voltage Output High Level (UART, I2C)	V_{DDIO_VOH}	V_{DDIO} > 3V , I_{OH} =10mA	0.9	0.8	--	V_{DDIO}
POR Voltage threshold on VDDIO-IN rising	V_{DDIO_POT+}	V_{DDIO} falls at 1V/ms or slower	--	1.45	--	V
POR Voltage threshold on VDDIO-IN falling	V_{DDIO_POT-}		--	0.99	--	V
Operating Temperature	T_A	--	-40	--	+85	°C
Total supply current normal mode at T_A (9DOF @100Hz output data rate)	I_{DD} + I_{DDIO}	V_{DD} = 3V, V_{DDIO} = 2.5V	--	--	12.3	mA
Total supply current Low power mode at T_A	I_{DD_LPM}	V_{DD} = 3V, V_{DDIO} = 2.5V	--	--	0.4	mA
Total supply current suspend mode at T_A	I_{DD_SuM}	V_{DD} = 3V, V_{DDIO} = 2.5V	--	--	0.04	mA

We have used 4 IMU sensors placed at different body locations. We get the fused value of the angles or orientation values from the sensor. We are using the Roll Pitch Yaw sense of rotation.

2.3.1 Definitions: Roll axis is defined as the Y-axis. Angular rotation along the Y-axis Ωy is called pitch mode which will cause the pitch angle to change.

Pitch axis is defined as the X-axis. Angular rotation along the X-axis Ωx is called roll mode which will cause the roll angle to change.

Yaw axis is defined as the Z-axis. Angular rotation along the Z-axis Ωz is called yaw mode which will cause the yaw angle to change.

Roll Pitch Yaw (RPY) Convention

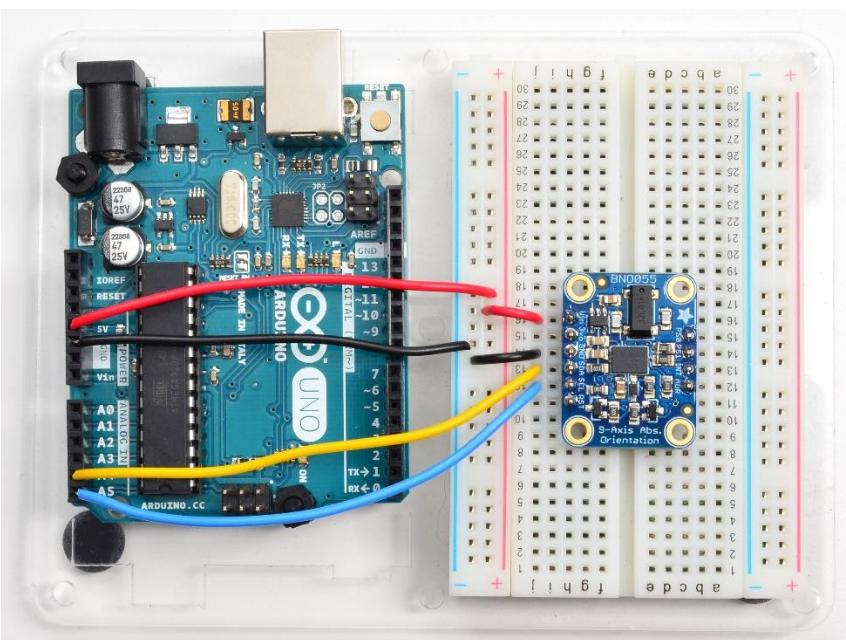
Rotation about x0 of angle γ + Rotation about y0 of angle β + Rotation about z0 of angle α

All rotations are about fixed frame (x0, y0, z0) base vectors

Homogeneous Matrix and Angles are identical between these two conventions:

$$\text{Roll Pitch Yaw XYZ } (\gamma,\beta,\alpha) \Leftrightarrow \text{Euler ZYX } (\alpha,\beta,\gamma)$$

2.3.2 Hardware Design of IMU with Arduino: The hardware connection of one IMU sensor with Arduino Uno is shown. The Arduino Uno has two Pins named SDA (Serial Data Analog) and SCL(Serial Clock) to let communication happen between the sensor and the Arduino. The Arduino board sends a serial clock pulse to the IMU sensor and in return the IMU sensor sends the analog data back to arduino. This is explained in the Digital Interface part in the coming section.



2.3.3 Digital Interface: The registers embedded in the BNO055 may be accessed through the I2C serial interfaces. The serial interfaces are mapped onto the same pins. To select/exploit the I2C interface, the SCL and SDA pin of BNO055 must be connected to the SCL and SDA pins of Arduino.

Pin Description of BNO 055.

Pin Name	Pin Description
Vin	3.3V to 5V power supply input.
GND	The common/GND pin for power and logic.
SCL	I2C clock pin, connect to your microcontrollers I2C clock line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.
SDA	I2C data pin, connect to your microcontrollers I2C data line. This pin can be used with 3V or 5V logic, and there's a 10K pullup on this pin.
ADR	Set this pin low to change the default I2C address for the BNO055 if you need to connect two ICs on the same I2C bus. The default address is 0x28. If this pin is connected to 3V, the address will be 0x29
PS0 and PS1	These pins can be used to change the mode of the device (it can also do HID-I2C and UART).They should normally be left unconnected.

2.3.4 I2C Serial Interface: The BNO055 I2C is a bus slave. The I2C is employed to write data to registers whose content can also be read back. There are two signals associated with the I2C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. When the bus is free both the lines are high.

I^2C terminology

Term	Description
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by the master

2.3.4.2 I2C Communication:

One of the most common communication buses in the world is the I2C bus. It may not be as well known as USB or Ethernet, but much of our world of electronic devices is completely dependent on it. An I2C bus can potentially have multiple masters and many 'slave' devices sharing the same bus, although you rarely see multiple masters in the real world. Signal contention is avoided by means of an open-collector drive scheme, where no device will ever drive a signal high, it only drives low; for a high, the bus is pulled up by a resistor. An addressing scheme allows a device to determine if it is being queried by the master. A master sees that a slave device is present when the slave responds to its address by sending an Acknowledgement bit (driving SDA low); if no such addressed device exists, the SDA line is pulled high, and this is interpreted by the Master as a Not-Acknowledgement.I^2C is appropriate for peripherals where simplicity and low manufacturing cost are more important than speed. Common applications of the I^2C bus are: Reading configuration data from SPD EEPROMs on SDRAM, DDR SDRAM, DDR2 SDRAM memory sticks (DIMM) and other stacked PC boards, Reading real-time clocks. Turning on and turning off the power supply of system components.

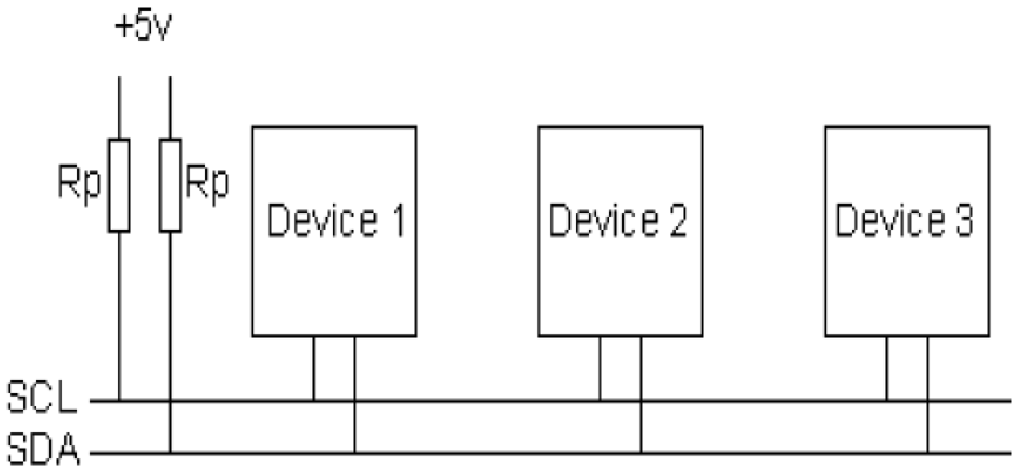


Fig 2.6 – I^{2}C PHYSICAL BUS

2.3.4.3 Integrating four BNO 055 sensors together: We have used a Multiplexer to integrate our four BNO 055 sensors that would communicate with the Arduino. The physical address of the IMU is 0x28

2.4 Microcontroller Arduino Uno:

2.4.1 ATMEGA 328:

Features:

- Operating Voltage 1.8-5.5 V
- 16 MHz Crystal Oscillator
- 28 Input Output pins
- 1 SPI Serial Interface

- 1 Programmable Serial UART
- Two Wire Interface
- 8 channel 10 bit ADC
- TQFP package

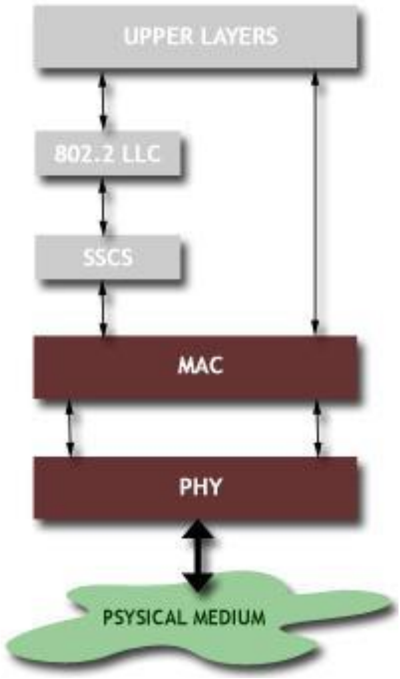
Usage: Interfacing IMU s and other sensors with it. It can easily be boot loaded with Arduino software.

2.5 Introduction to Zigbee Network:

ZigBee is a standard defined by the IEEE (Institute of Electrical and Electronics Engineers) back in 1998. When Wi Fi and Bluetooth could not be suitable enough for some applications, they started to define new protocol, new standard. ZigBee aim to have lower data rate compared to Bluetooth, at 250Kbps (Bluetooth is 1Mbps back then, now is higher with EDR-Enhanced Data Rate).More importantly, ZigBee aims for lower power applications which should offer years (2 to 3 years) of operation on non-rechargeable battery and it can connect to more nodes (2^ 16 = 16-bit address > 65,000); while Bluetooth only offers 8 devices connection in piconet. ZigBee designed for control and monitor while Bluetooth for cable replacement. ZigBee uses RF (Radio Frequency) to communicate. Many people think that ZigBee and XBee is the same thing. That‘s not true. ZigBee is a standard communications protocol for low-power, wireless mesh networking. Xbee is a brand of radio that supports a variety of communication protocols, including Zig-Bee, 802.15.4, and WiFi, among others. ZigBee is the name of a specification for a suite of high level communication protocols for WPANs (ZigBee, 2004). ZigBee is based on the IEEE 802.15.4 standard (IEEE, 2003) and the ZigBee alliance is cooperating with the IEEE 802.15.4 working group. The IEEE 802.15.4 defines the physical (PHY) layer and the Medium Access (MAC) layer. The specification for the PHY defines a low-power spread spectrum radio operating at frequency bands such as 2.4 GHz, 915 MHz and 868MHz. The specification for the MAC layer defines how multiple 802.15.4 radios operating in the same area can share the airwaves. The MAC layer specification also defines different network topologies. The ZigBee specification defines application profiles that allow devices from different manufacturers to communicate with each other.

2.5.1 Zigbee Device Architecture:

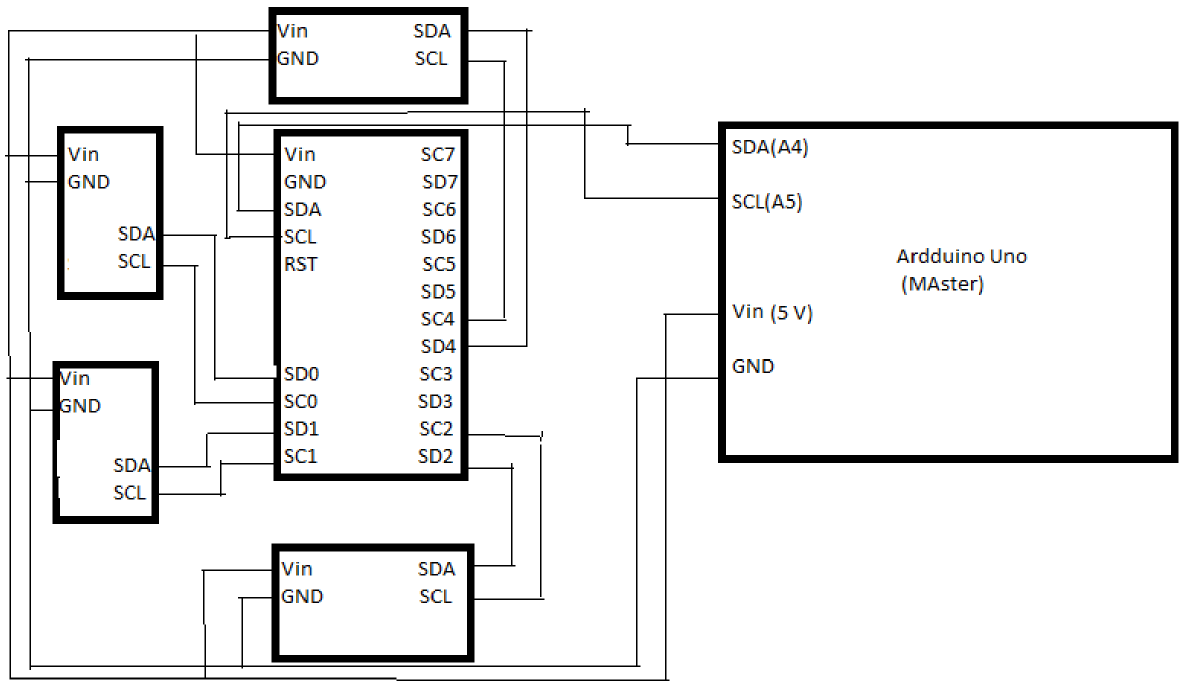
The architecture of a Zigbee network is illustrated in the following figure:



The PHY contains the RF (Radio Frequency) transceiver and its low-level control mechanism. The PHY also contains a MAC sub layer that provides access to the physical channel for all transfer types. The upper layers consist of a network layer and an application layer. The network layer provides network configuration, manipulation, and message routing and the application layer provides the intended function of a ZigBee device. The LLC (Logical Link Control) is responsible for the logical link functions of one or more logical links. Command packets generated by the LLC are called PDUs (Protocol Data Units). The LLC accesses the MAC sub layer through the SSCS (Service Specific Convergence Sub layer).

2.6 HARDWARE CIRCUIT DIAGRAM:

The overall hardware circuit consists of a multiplexer, four BNO 055 sensor board, one main (master) Arduino uno board. The Arduino board is connected to the Laptop where it interacts with the Software. The Hardware Implementation Circuit is shown below:



3.0 CODE

The Arduino code for interfacing the IMU sensors and getting the fused valued of Roll Pitch Yaw is given below:

new_imu

```
//#include <I2C.h>

#include <Adafruit_BNO055.h>


#include <Wire.h>
#include <Adafruit_Sensor.h>

#include <utility/imumaths.h>

#define TCAADDR 0x70

Adafruit_BNO055 bno1 = Adafruit_BNO055(55);
Adafruit_BNO055 bno2 = Adafruit_BNO055(56);
//Adafruit_BNO055 bno3 = Adafruit_BNO055(57);
//Adafruit_BNO055 bno4 = Adafruit_BNO055(58);

float s1[3], s2[3];
void tcaselect(uint8_t i) {
  if (i > 7) return;

  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
}

void setup()
{
```

```
{
  sensors_event_t event1;
  sensors_event_t event2;
  int i;
  Serial.begin(115200);
  //Serial.println("Orientation Sensor Test");

  tcaselect(0);
  bno1.begin();

  tcaselect(1);
  bno2.begin();

  tcaselect(0);
  delay(10);
  bno1.begin();

  tcaselect(1);
  delay(10);
  bno2.begin();

  //Serial.print("Setting the sensonrs to 0.....");
  for (i=0; i<1000; i++) {
    tcaselect(0);
    bno1.getEvent(&event1);
    s1[0]=event1.orientation.x;
    s1[1]=event1.orientation.y;
    s1[2]=event1.orientation.z;
```

```
}
void loop()
{

  sensors_event_t event;

  tcaselect(0);

  bno1.getEvent(&event);
  Serial.print(event.orientation.x-s1[0], 3);
  Serial.print("a");
  Serial.print(event.orientation.y-s1[1], 3);
  Serial.print("a");
  Serial.print(event.orientation.z-s1[2], 3);
  Serial.print("a");


  tcaselect(1);
  bno2.getEvent(&event);
  Serial.print(event.orientation.x-s2[0], 3);
  Serial.print("a");
  Serial.print(event.orientation.y-s2[1], 3);
  Serial.print("a");
  Serial.print(event.orientation.z-s2[2], 3);
  Serial.println("");
  delay()
```

The IMU sensor gives the fused value of the Roll Pitch and Yaw data. This fused value is filtered and the noise in measuring the angles is cancelled. What we get out of the sensor is the relative orientation value of angles in the X, Y, Z direction. Now, to get the value of position we have to double integrate the data we get from the IMU sensor. This is the toughest thing to do as double integration of the acceleration data brings a lot of drift in the position value which is too difficult to handle. One way of correcting this issue is using a Kalman Filter. The Kalman Filter is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies and produces estimates of unknown variables that tend to be more precise than those based on single measurement alone. The implementation of Kalman Filter to get position from the IMU sensor data by double integration is one of the most hot topic of research currently.

3.1 How to get meaningful position data from the IMU sensor?

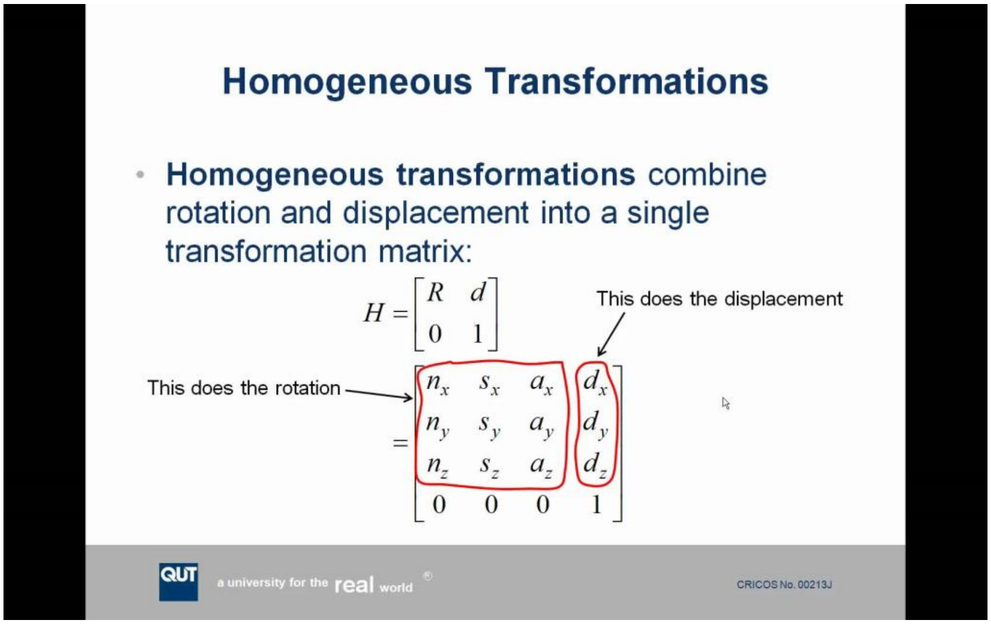
To solve the above problem we came up with the idea of finding the position of a particular sensor with respect to a fixed point in the body. The whole idea is to determine how one sensor is moving with respect to the fixed sensor. We placed one sensor in the base of neck and made it our origin as this point is fixed and does not move. Now, we placed our other sensors at a fixed distance from this sensor. Having the roll pitch yaw value and knowing the fixed distance of each link we can build what is known as a Homogeneous Transformation Matrix.

A homogeneous transformation matrix gives a relative orientation between two points. It consists two parts, Rotation and translation. The Rotation part of the matric comes of the Roll Pitch yaw values. The translation sets the fixed distance where the sensors are placed. Suppose the first sensor is placed at a distance of d towards the x axis of the fixed sensor the coordinate of this sensor will be (d,0,0) with respect to the fixed sensor. The Rotation Matrix based on Roll Pitch Yaw is given by:

$$
\begin{aligned}
&R_Y R_Z R_X \\
&= \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \\
&= \begin{pmatrix} \cos B \cos C & -\cos B \sin C & \sin B \\ \sin C & \cos C & 0 \\ -\sin B \cos C & \sin B \sin C & \cos B \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \\
&= \begin{pmatrix} \cos B \cos C & -\cos B \sin C \sin A + \sin B \sin A & \cos B \sin C \sin A + \sin B \cos A \\ \sin C & \cos C \cos A & -\cos C \sin A \\ -\sin B \cos C & \sin B \sin C \cos A + \cos B \sin A & -\sin B \sin C \sin A + \cos B \cos A \end{pmatrix}
\end{aligned}
$$

Where A= angle around X(pitch), B= angle around Y(Roll) and C= angle around Z(yaw).

The Homogeneous matrix is 4*4 matrix given by :



So based on this to get the position of the sensor we use the following equation:

$$H_2^{0} = H_1^{0} H_2^{1}$$

Where H_1^0 is the Homogeneous transformation matrix between sensor 1 and fixed sensor and similarly for H_2^{1}.

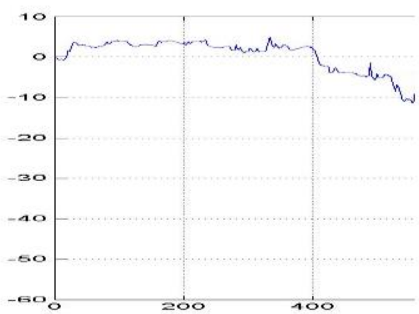
To get relative position from this we use the following equation:

$$\begin{matrix} X \\ Y \\ Z \end{matrix} = H_2^{0} \begin{matrix} X' \\ Y' \\ Z' \end{matrix}$$

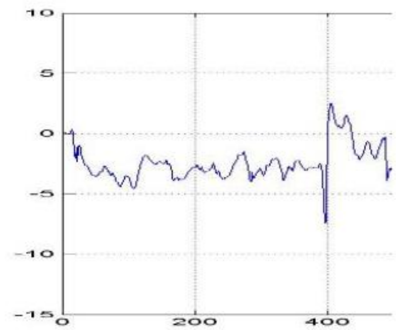
Where [x,y,z,1] is a 4x1 vector for the current position of sensor 2 with respect to fixed sensor and [x’,y’,z’,1] is the starting position of sensor 2 with respect to fixed sensor.

4.0. Output graph:

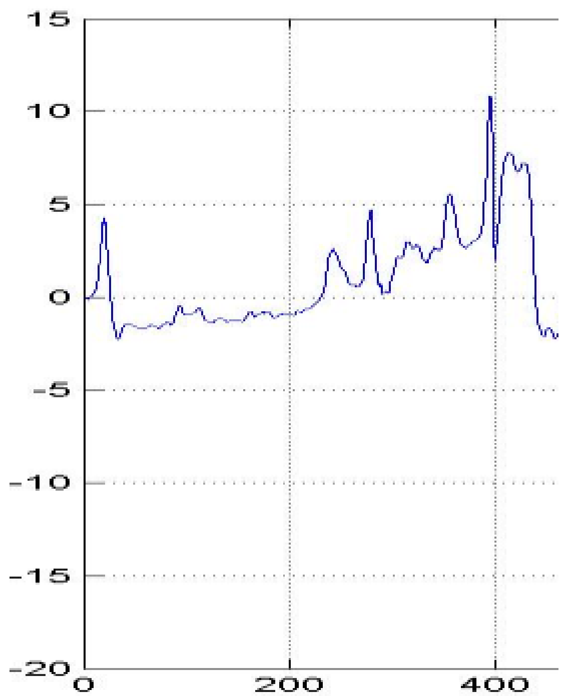
Here is the plot of the Roll, Pitch, Yaw data from the imu sensors:



Pitch



Roll



The plot of the angles from the sensors in MATLAB

5.0 Problems faced: We faced a number of problems. At first we used MPU 6050 which is 6 DOF sensor. This gave a lot of errors in the angle measurement due to a large noise which produced a lot of drift. So we decided to use 9 DOF BNO 055. Next, we tried to double integrate the acceleration to get the position data. This is a very hectic task. It causes drift of large scale which is very difficult to get remove. Due to time constraints we could not implement the Kalman Filter to get position as it is very difficult to implement. So we decided for an alternate approach. We faced a lot of problems while configuring the multiplexer. The wiring of Multiplexer did not let us collect data and it took a long time for us to configure the problem of loose connections.

6.0 Future Work : We have decided to implement the Kalman filter algorithm to bring the actual position data out. Also, we have decided to learn Processing software to implement a Virtual implementation of this.

7.0 Cost Analysis:

Materials	**Quantity**	**Unit Cost(each)**	**Total Cost**
9 DOF Inertial Measurement Unit	4	34$	136$
Arduino	1	32$	32$
Zig BEES	2	15 $	30$
Xbee adapter	2	14$	28$
Ftdi cable	1	5$	5$
Jumper wires	40 pieces	6$	6$
Long Connectors	70 pieces	12 $	12$
Velcro Bands	4	3$	12$
Multiplexer	1	25$	25$
Total			286$

7.0 Conclusion: The implementation of the this prototype is very useful for various fields like sports, yoga, dance etc. It has great utility in the field of Medicine like it can be used to treat Parkinson disease. The proto type proposed does not really guarantee successful detection of position and orientation in the various fields.

8.0 References

- Joint Angle Tracking with Inertial Sensors- Mahmood Ahmed Algohari- Phd Thesis
- Xsens MVN : Full 6 DOF human Motion Tracking using Miniature Inertial Sensors-By Daniel Roetenberg, Henk Luinge