IP JOYSTICK

Project objective-:

The project aims at developing a handheld device which can be used to detect the motion of the hand and simulate the coordinates and orientation of the device on a computer screen.

Hardware configuration-:

1. Accelerometer and gyroscope-Accelerometer is a device which gives acceleration with respect to gravity as a function of voltage and gyroscope gives angular velocity around its own axis as a function of voltage.
2. Atmega –Atmega is used for calculating tilt from the data of accelerometer and gyroscope and sending these datas to computer.
3. Computer –For image processing and graphics.

Software configuration-:

The graphics part was made using opengl and a 3-d modelling software 'blender'. After the object was made in blender, it was exported as an object file to the folder where the code was being written. Then using a sample code from the internet for a simpler object, the code for our object was developed.

Then in the code written for graphics, a function 'ip' was created which did the image processing part. This function was called after a fixed interval of time. Simultaneously, the data from the tiltometer was also taken from UART, and both combined to show the respective changes in the graphics.

Plan of action-:

The device will use one 3-axis accelerometer and one 2 axis gyroscopes to find the orientation of the device. The analog values given by the two devices will be read by a microcontroller and processed to give information about angle rotated. This will be sent to a computer for processing via serial communication. The computer will do image processing to find the coordinates of the device by detecting the spherical lighted ball on top of the device and display the motion of the device on the screen.

For this, the project is divided in two parts. First part aims at building a tilt meter with an accelerometer and gyroscope. The second is associated with the image processing and the graphics design.

1.The tilt meter

For building the tilt meter, we had to deal with the changing reference axes in the frame of our device. Every orientation of a given triangular axis in space can be described as consecutive rotations about initial axes. A 3x3 orthogonal rotation matrix is used to relate values in the frame of rotation and the initial frame. Basically multiplying a 3x3 rotation matrix with the "rotation frame coordinates" gives the coordinates in "initial frame". So three consecutive rotations result in multiplying three 3x3 rotation matrices. *The Euler angles of roll-pitch-yaw are commonly used to describe rotation between earthfixed frames and body frames. The rotation is based on the simple rotations and is given by*
R= Rz(_)Ry(_)Rx(_);

This matrix was updated at every stage of data acquisition. This involved multiplying it with another matrix for small angle rotations. In updation matrix these small angles will be "angular velocity*dt".
After multiplying updation matrix from matrix A we get the the final angles after time 'dt'. In this way we find the tilt from the gyroscope. Tilt from the accelerometer is also measured with the help of this rotation matrix.

The most important task was to filter the data that comes from accelerometer and gyroscope. Ideally, we could have used any one of the 3 axis accelerometer or the 2 axis gyroscope but we could not use just one device to calculate the tilt as both the sensors have some practical limitations. The accelerometer does not respond to immediate changes in acceleration (although over a large time, it give a good reading of the acceleration) and also gives short term fluctuations, whereas a gyroscope gives accurate readings for instantaneous changes but drifts from the initial value as the time goes on. Thus filtering was required to get good readings.

We first understand the linear complementary filter implemented for finding one dimensional tilt.It can be understand as a low pass filter on accelerometer value and high pass filter on gyroscope value i.e :

angle = (a)*(angle + gyro*dt) + (1-a)*(x_acc);

Basically, if you know the desired time constants and the sample rate, you can pick the filter

coefficient a .Because

t=(a*dt)/(1-a) ;

Time constant defines where the boundary between trusting the gyroscope and trusting the accelerometer is. For time periods shorter than half a second, the gyroscope integration takes precedence and the noisy horizontal accelerations are filtered out. For time periods longer than half a second, the accelerometer average is given more weighting than the gyroscope, which may have drifted by this point.

We apply anology based on 1 dimensional filter and implement it in three dimension.Angle find by multiplying updation matrix to matrix A is multiplied by const ‘a’ and added to angle find by accelerometer multiplied by ‘1-a’.We got the desired constant ‘a’ according to our sampling rate and desired time constant.

2.Image processing:

For the image processing, opencv was used as this is faster than matlab which is normally used in other projects. Blobs library was used for this purpose.

1. The ball on the top of the device was detected using blobs functions.

2. Then the largest blob was selected amongst all the blobs.

3. The area of the blobs used to calculate the radius of the blob. Then, using the radius of the original, the expected distance from the camera was calculated.

4. The centroid of this largest blob was used as the center of the ball.

5. Using the radius of ball in pixels and the actual radius of the ball, the change in postion from the center of the screen was calculated.

Thus all the three coordinates of the ball were found in space. To remove noise from lighting conditions around the ball, a bulb was fitted inside the bulb.

What can be done??

By increasing processing speed and adding a magnetometer to it we can make the device to response faster which can be used as a gaming perspective.Also the object used can be replaced by any other device(a part of fun).The same concept can be used in designing a 3d wireless mouse.