1 Image display

The intention was to rotate the motor at a given speed, such that POV would occurs. POV also called persistence of vision, is the phenomenon that states that the human eye always retains images for a fraction of a second (around 0.04 second).

> ref til http://www.mediacollege.com/glossary/p/persistence-of-vision.html

.

> image... Image in a raster grid

The image that had to be displayed, was drawn in a raster grid. Using the encoders , would it be possible to provide some form of information of the angular position of the LED strips, and thereby which pixels had to be displayed.

> image in raster grid , and the propellor placed in angle..

A problem faced here was on how to determine which pixel had to be displayed, in a discrete image description. The problem here is given an angle and a endpoint on how to draw a straight line within these two points, and how to determine the in between pixel positions.
Finding the endpoint and angle was pretty easy, the angle was given by the encoders, and the endpoint could easily be calculated trig identities.

To draw the line between two points (being the startpoint and the endpoint) were an modified version of Bresenham line algorithm used, modified as the original algorithm is only capable of handling cases from where the slope between the two points is between 0 and 1.

The Bresenham line algorithm is an algorithm used for determining the in between points in raster grid. It start by computing the change in the x-axis and y-axis between the start and endpoint, and thereby define a major axis as the one which change with the highest change.
The major axis is used define the direction which the algorithm should move along.
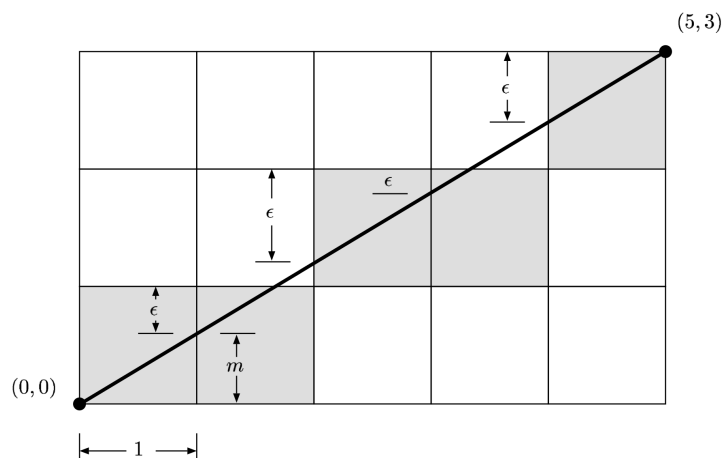

Figure 1: Bresenham line algorithm applied on a simple case

> http://www.idav.ucdavis.edu/education/GraphicsNotes/Bresenhams-Algorithm.pdf

in the example given in figure 1 the major axis is X. In a main loop will the variable x be incremented by one, and for each incrementation will the y coordinate be incremented with an value m which is defined as

$$m = \frac{dy}{dx}$$

while incrementing the y value is an the error bound ϵ kept which represent the the negative distance between the real line and top edge pixel. The value is incremented with m for each time the major axis coordinate is incremented by one.

ϵ becomes greater than zero if the real line has moved upwards according to the previous x value, when it occurs will the y value be incremented with one, and *epsilon* will be recomputed to its new value as 1 - *epsilon*.

As long the epsilon value is below 0, will the y coordinate only increment in steps of m.