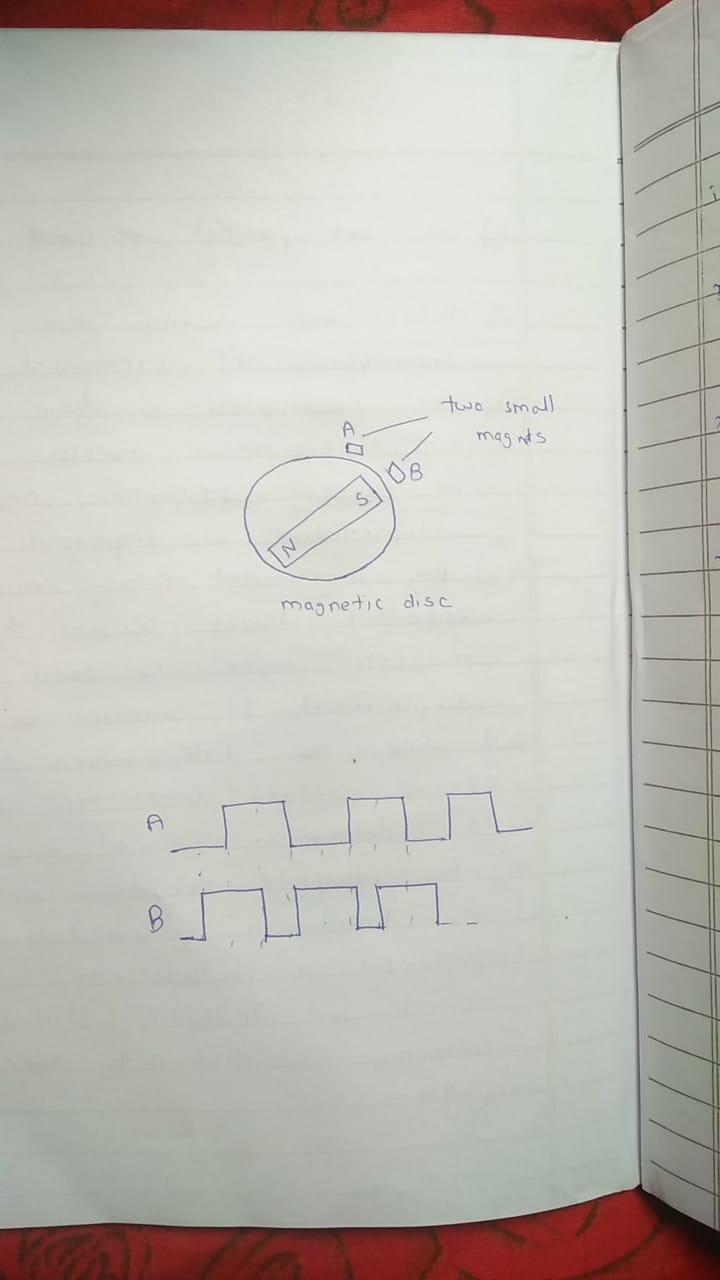
*LOCALISATION USING ENCODER*

Aim: To get the bot to the coordinates accurately.

Equipments used: Encoder motor, SRA Board, Bluetooth Module HC05, Battery, Motor Driver L298 and a plywood plank, Castor wheel and wheels.

Procedure:

1. First we measure the ticks of an encoder motor.
2. An encoder is a device that determine how far a rotational mechanism has turned.
3. So if we join the encoder to the shaft of the motor we can determine the distance the distance travelled by the wheel.
4. An encoder motor is a combined device which contain an encoder with a motor.
5. There are mainly two types of encoders
6. Incremental Encoder.
7. Absolute Encoder.
8. The construction of this two types is quite similar however they differ in physical properties and the interpretation of movement.
9. An incremental encoder (quadratory encoder) utilizes sensors that use optical, magnetic index to count angular displacement. It contains a transparent disk which contains opaque sections that are equally spaced to determine movement.
10. An absolute encoder contains component also found in incremental encoders. They implement a photodetector and LED light source but instead of a disc they contain a disc with concentric circle patterns.
11. In our project we use a magnetic incremental encoder.
12. In this type of encoder there is a magnetic disk which rotates with the shaft of the motor.
13. The disc is surrounded by two small magnetic as shown in the fig.



1. As the disc rotates the north and south poles of the disc passes close to the small magnets.
2. When there is attraction between small magnet and the disc it gives high value

(+5V) and when there is repulsion it gives low value (0V).

1. Thus it produces two square waves that are 90 out of phase as the disc rotates.
2. Depending upon which wave is leading we can determine the direction of rotation of the wheel.
3. Ticks is the measure of how many times a pulse is generating.
4. If we know the ticks per revolution of the encoder then we can calculate the angle turned by the wheel and it will give us the distance travelled.
5. So first we calculated the ticks of encoder motor for which we use the interrupt pins of ATMEGA 16.
6. Interrupt is used when we have to carry out a function in background without interrupting the ongoing function.
7. D2 and D3 are the interrupt pins in ATMEGA 16.
8. One square wave (A) is send in to D2 and another square wave (B) is send to send to any pin i.e. we used A5.
9. Whenever there is falling edge in A the interrupt function will be called and the status of B will be checked.
10. If B is high (1) means wheel is rotating clockwise so the ticks will be increasing. If B is low (0) means wheel rotates in counter clockwise direction and the ticks will be decreasing.
11. Now we have the ticks of the encoder so we can calculate the distance travelled by the wheel, Now we will come to giving the coordinates by HC05 Bluetooth Module.
12. The HC05 Bluetooth Module have 5 pins status Rx, Tx, Vcc, GND, EN. We can only use 4 pins i.e. Rx,Tx, Vcc and GND where Rx pins is connected to Tx of SRA Board. Tx is connected to the Rx of the SRA Board. Vcc is connected to 5V. GND is connected to GND.
13. The Bluetooth Module will receive the coordinates that will be send by the mobile Bluetooth app.
14. We use the app “HC05 Bluetooth Module” for this purpose. In this app you can send only ASCII and HEX values. We choose HEX because in HEX “1” means “1” in decimal and it’s very easy to interpret the values.
15. We used the foot conversion.1 unit = 1 foot = 30.48 cm. That means if we send the coordinates (3, 4) that will be 3 foot in x and 4 foot in y.
16. This coordinates are converted in “r” and “theta” in MCU.
17. We placed the bot as if it heads towards +ve x-axis at the origin.
18. The bot will take a spot turn according to the “theta” value and reach the given coordinate by the value of “r” which it manipulated.
19. The bot takes the spot turn means the distance from the centre of the distance between the two end wheels to the wheels end is taken as the radius of the rotation. Let take it as (r).
20. Then we can easily convert the coordinates into distance travelled by the bot by using:

**Distance travelled by bot = (theta/360)\*2\*Pi\*r.**

1. Then we can convert the distance into ticks as we know the ticks per revolution and ticks per circumference.
2. The bot will spot turn until this ticks are achieved.
3. Now the bot has turn “theta” degrees it has to go “r” distance.
4. First we will convert “r” into ticks as we know the ticks per circumference of the wheel.
5. Thus the bot will go straight until this ticks are achieved.
6. That what we written in the code.

Problems we faced.

1. **Measuring Ticks.**

First we used the arduino code to call the interrupt pins but that code was not working so we used the resistive code to do the same.

**2. Bluetooth Module.**

There was a problem related to the app that we used to send the coordinates. The app can only send the values upto two digit that means max value is FF (255).If we have to send negative HEX values the it should be of 8 digit starting with no. greater than 9 until F. Thus we were unable to send the negative coordinates. So to solve this problem we made our main convention that we were sending four values instead of two. Two for signs and two were exact value of x and y.

We made a convention like FF (255) for negative sign and F (25) for positive sign.

So if we want to send (3,-4) we would rather send it by sending four inputs like (F, 3, FF, 4 ).

**3. Motor speeds.**

The motor speeds were different on the same pwm and therefore the bot was not going straight. To overcome this we used PID. PID stands for Proportionately Integration Differentiation. In PID we calculated the difference between the ticks of the right and left wheel.

**Difference = Left – Right**

If the difference is positive that means the left motor is moving faster so MCU will slow down the left motor by decreasing its pwm value. We used only “P” from PID that means the value of the pwm will decrease by (Kp \* Difference) where Kp is the proportionality constant. The value of Kp is determined by trial and error method.