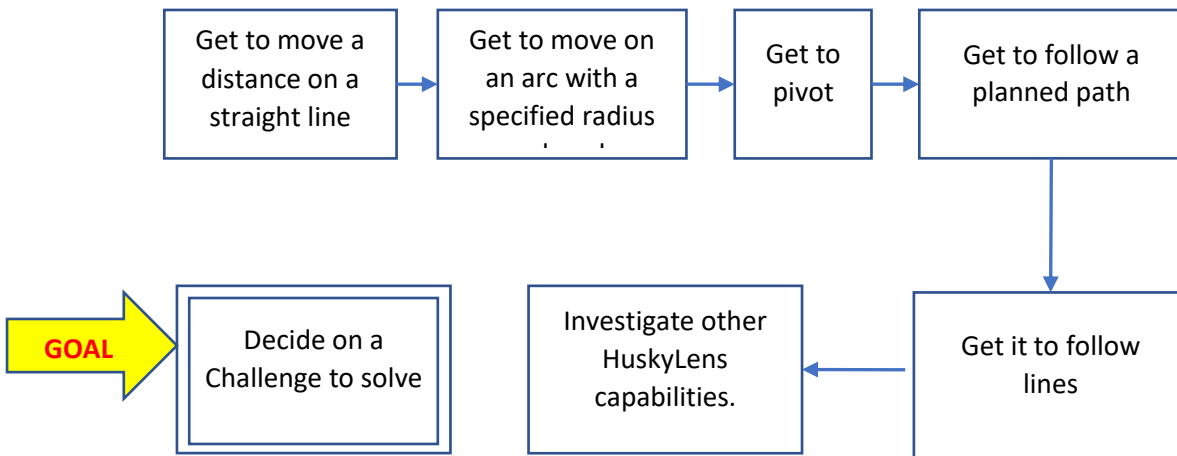


# Harry Mini-Project Brief

CBP 31-01-23. Here's a suggested workflow



## Moving along a straight line

- Sketch 3402\_D\_Straight\_Line.ino
  - variable **dx** (library) is the distance a wheel moves in mm for one step
  - variable **desiredDist** how far you want the robot to move in mm
  - function **setSpeeds(...)** is found in **CBPFBO\_StepperAX.h**
  - function **stepMotors(...)** is also found there
  - The while loop to take steps, each iteration take 1 step for left & right
- Try it out for a couple of desired distances

## Moving along an arc

- Sketch 3402\_D\_Arc
  - The 'algorithm' is discussed at this link [https://colin-price.wbs.uni.worc.ac.uk/Courses\\_2021\\_22/Comp2403/CBP\\_Notes\\_Book/Ch1\\_Kinematics\\_StepperMotors.pdf](https://colin-price.wbs.uni.worc.ac.uk/Courses_2021_22/Comp2403/CBP_Notes_Book/Ch1_Kinematics_StepperMotors.pdf)
  - You specify the **radius** of your arc and its angle in **degrees**
- Complete and test the sketch. Try various arcs

## Pivoting

- Sketch 3402\_D\_Pivot
  - You specify the angle in **degrees** you wish Harry to pivot
- Try various angles

## Get to follow a planned path

- Sketch 3402\_D\_Paths
  - Shows how to connect a line and an arc
- Now create an interesting path.

## Line Following

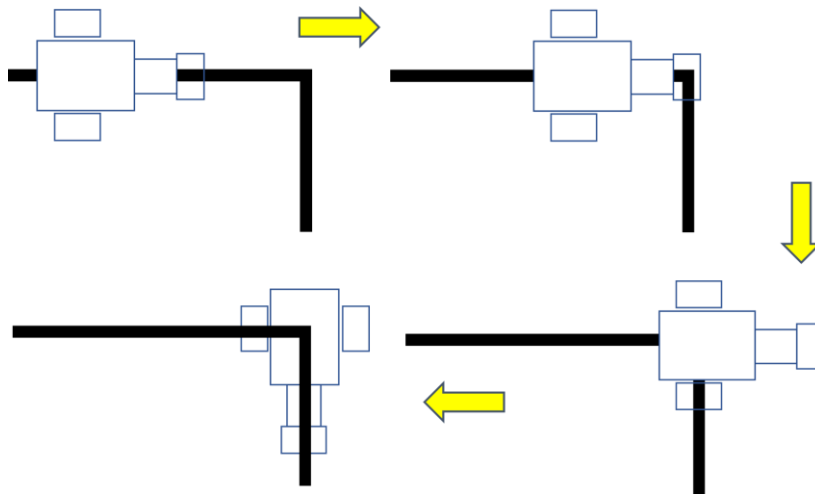
- Sketch 3402\_Husky\_LineFollower
  - Looks at the origin of the vector Huskylens finds on the line
  - Error calculated using half the sensor/screen width
  - Error normalised to 1
  - Error passed to library function **moveABit(...)** in the library **CBPFBP\_StepperA**

## Junction Detection

- Sketch 3402\_D\_Husky\_JuncDetect
  - See what this does and see what you can make of it.

## How to Navigate a Path

Here's one way – following lines and pivoting.



Another way would be to get Huskylens to read bar-codes placed where you want it to move. The red squares in the fulfilment centre shown below are bar-codes

