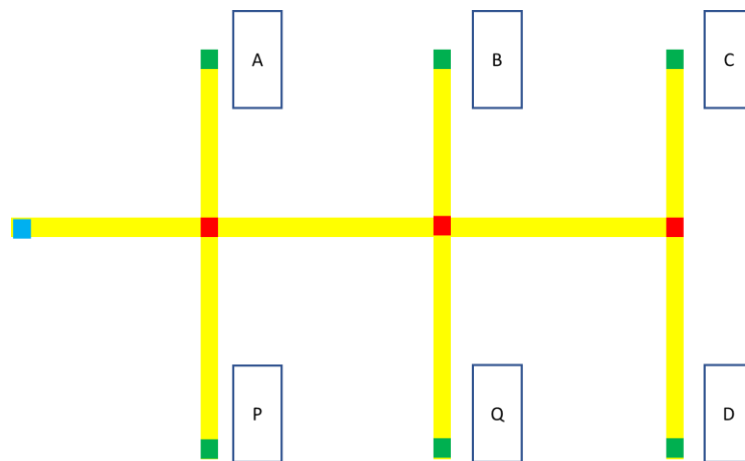


Factory Floor / Warehouse Robots

Basic Idea

Here you will start with an existing line-follower robot and extend this so it can recognize junctions between lines and choose which path to take at each junction. This could model a factory floor where components are moved around between assembly stages, or items are picked up from a shopping list. Line-following is just one way of obtaining navigation for warehouses, laser range-finding or vision systems can be used when the robot is given a map of the environment.



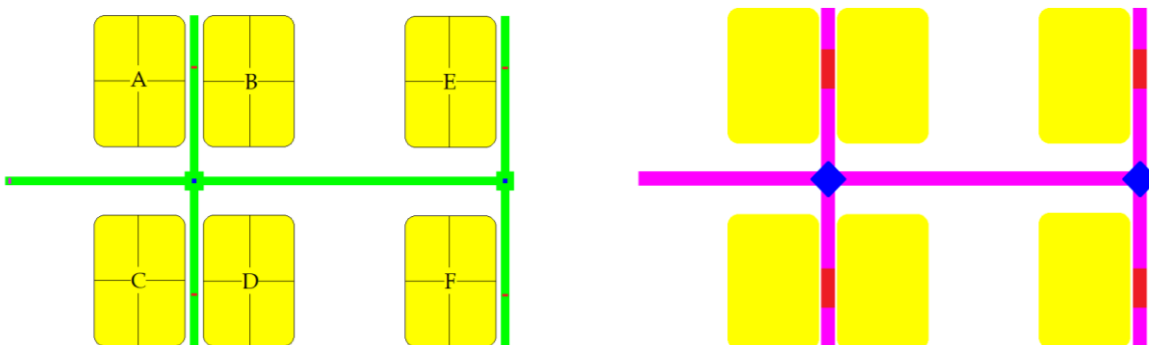
The above graphic shows the idea. The robot start-end is shown in blue, locations of the racks in green and junctions in red. Let's say components from some of the racks need to be retrieved to the starting point. There are various approaches to doing this, where there is a given shopping list.

- 1) A single robot could traverse the entire grid and collect all items
- 2) A single robot could retrieve one item at a time
- 3) A robot would be assigned to each rack, so several robots would collect the items, each would collect one.

I'm sure you could imagine other ways of doing this

Designing the Floor

Your image should be a .png, square with lengths a power of 2. The following examples are both 1024 x 1024 (though they are shown cropped in the diagram).



A good way of getting the track widths correct is to start with the example provided in the assets. Look at the world **CBP_3352_LinearCamera_1.wbt**. The floor image is **worlds / textures / tri_color.png**. You could load this into your photoshop software and use as guidance.

Coding the Controller

Start with the existing controller **CBP_3352_LinearCamera_2**. The file **CBP_3352_Helper2.c** contains the function **imageSegment(... double* colourFound)**; which returns the hue of the line detected as **colourFound** as well as the number of pixels of this color.

You should extend this code; a Finite State Machine of some sort is probably the best architecture to use.