REQUIREMENT SPECIFICATION 1.0

USER REQUIREMENTS

1. The user should be able to select a manual or automatic mode.
    a. In manual mode the next destination of the train can be chosen.
    b. In automatic mode the train goes to either random or sequential stations until stopped.
2. The user should be presented with a simple display that can be controlled using directional arrows and a confirm button.

HARDWARE REQUIREMENTS

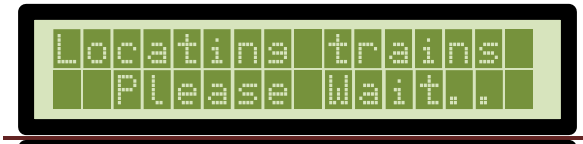
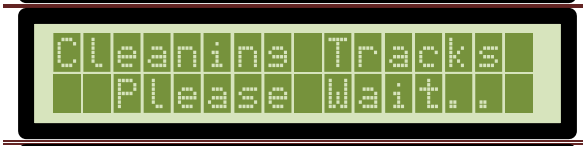
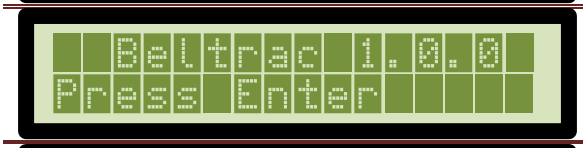
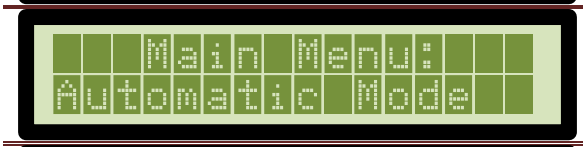
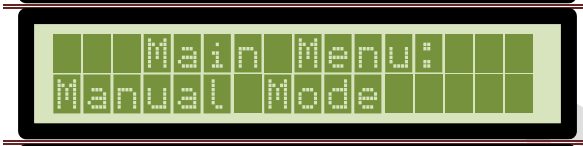
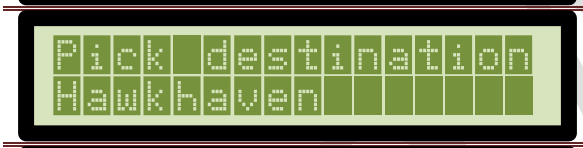
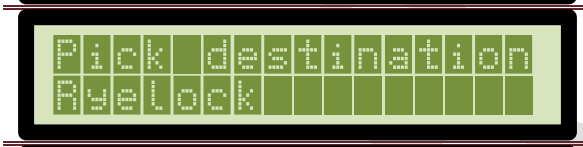
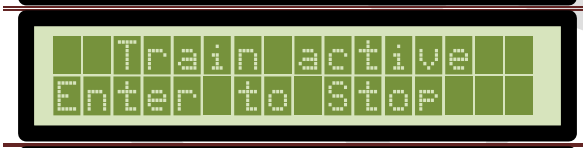
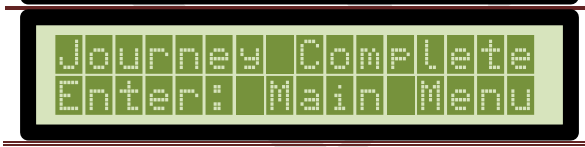
1. There should be a small train (00 gauge) which is controlled by a microcontroller with the user interface attached.
2. The train should travel at multiple speeds and stop at multiple stations with points used to provide multiple possible routes.
3. Sensors on the track or the train should detect its location and this information should be used to manoeuvre the train.
4. The maximum voltage of the train should never be exceeded.
5. The train should appear to gain or lose speed smoothly.
6. The tracks should be able to isolate a siding so that the train on it can be held, allowing a different train a turn on the tracks.
7. The train should be able to travel forwards and backwards at equal maximum speeds.
8. The layout should be designed that if the train is moved forward that no matter where it is it will always trigger a sensor on its journey.
9. Fail safes such as circuit breakers should be in place to protect the train and the equipment.

SOFTWARE REQUIREMENTS

1. The microcontroller should be able to recognise the location of the train and act accordingly.
2. It should anticipate the possibility that a sensor has failed and should be able to act if the sensors are not called in sequence.
3. The software should be able to plot a route from any station to any station.
4. The software should be able to activate a cleaning train at any time to clean all the tracks and it should be able to manage the activation of both this and the main train by isolating sidings.
5. The software should be able to reascertain the location of the train in case of a power failure by moving it forward until it triggers a sensor.
6. The software must not perform any actions that could damage the train or any other equipment.

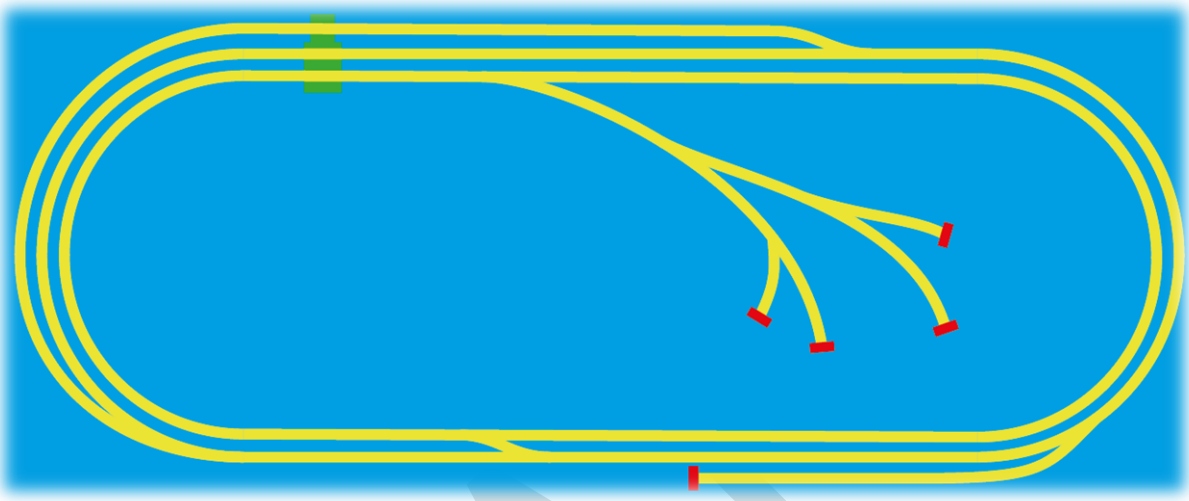
To help in understanding these requirements a few examples are provided below,

EXAMPLES OF OUTPUTS

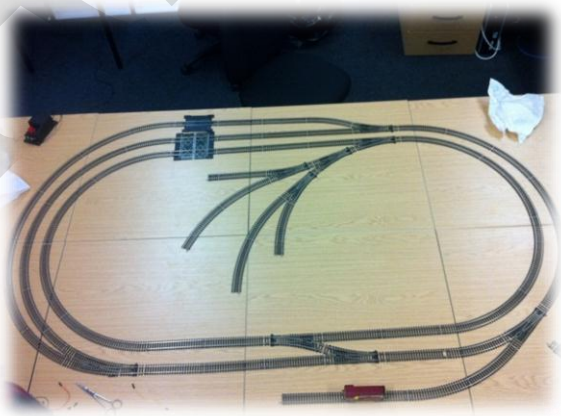
	This would be the starting screen, we need to wait while the system moves the active train onto a sensor
	This would then show while the cleaning train travels the whole track, dusting the tracks, we need to wait a little longer…
	This would be the welcome screen, lets press enter
	This is the first page of the main menu, lets choose manual mode by pressing down
	And press enter when we have it
	Ok so we don't like the idea of going to Hawkhaven so let's press down and choose a different destination
	Ok so Ryelock seems nice, lets press enter and send the train there
	(the words 'train active' will flash) let's let the train arrive
	We did it! The train is safely at Ryelock, if we press enter we can start again and send the train somewhere else

EXAMPLES OF DESIGN

TRACK LAYOUT



This is the preliminary layout for the track, it looks a lot like this picture although when the picture was taken the central hub was facing the other way, this had to be changed so that the train would always trip a sensor by moving forwards.



The red bars are buffers and the green boxes are decorative level crossings.

All the points are automated, there will be sensors at strategic locations although they arent in place yet.