GCE Computing

Doverbroeck's Colage
62315

Michael R. Bell
7894

22/01/2013 build

[BELTRAC]

Model railway automation.

CONTENTS

DEFINITION, INVESTIGATION AND ANALASIS

DEFINITION – NATURE OF THE PROBLEM TO BE INVESTIGATED

THE END USER

The project is being produced for John Peter Thomas, the owner of a small computer company. In his own words, this is a brief description of his company:

> The company is a lifestyle company, i.e. we keep it relatively small, we enjoy our work and we're driven more by the quality of the work we're offered rather than the money. That said, we've recently grown a little to 10 developers who work on a range of bespoke projects for a wide variety of clients. We started out writing code for embedded systems, but have more recently specialised in code for web applications and mobile platforms. We've written primarily for the iPhone, but more recently we've been looking at Android apps. On occasion we are asked to develop full blown web sites, though we have tried to stay away from these as much as possible. We've recently employed a graphics designer for the first time and will most likely take on more web based projects as time moves on.
>
> We've taken on projects for students from d'Overbroecks in the past and without exception have been really pleased with the results. I'm looking forward to seeing a preliminary requirements specification soon.

THE PROBLEM

Mr Thomas has a lobby where people wait before meeting him, as this would be where potential clients would find themselves waiting for meetings with him and his staff it would be good to have something to entertain potential clients while they wait, which at the moment they don't have.

Providing something interesting to entertain has several advantages:

- It prevents potential clients from getting bored and so keeps their mood light for the meeting ahead which can increase the chances of success
- It can display what the company is capable of or, if not created by the company, can imply an ability to produce similar things or an interest in the particular subject matter. Again, this can increase the chances of a deal being struck as well as providing ideas for alternative products

RESOURCES PROVIDED

In terms of what Mr. Thomas is willing to provide me it would really just be some floor space and a power supply to use, being England this would be 230~240V(rms) at 50Hz

FURTHER STEPS

Having gathered a bit of information I arranged a meeting with Mr Thomas, in conversation I asked him some questions about the project, I noted down some basic points based on the responses to the questions:

- We decided to call the project Beltrac
- Beltrak will be on a static platform (no wheels)
- Beltrak will be a loop
- Beltrak will have a cleaning train which dusts the track when switched on
- Beltrak will have a standard train which can be controlled
- Beltrak will have a small display with a simple menu system
- Beltrak will have automated points that can change to navigate the trainto its destination
- Beltrak will have sensors on the track to detect the location of the train
- Beltrak will have an integrated cuircuit breaker to protect the train from surges and from the control board supplying to much current
- Beltrak will have an easy to use interface that requires no pre instruction (though a manual will be provided

Of course, this alone is not enough to create a requirement specification but the information discussed is a good start as well as an opertunity to get to know Mr Thomas a bit better.

INVESTIGATION AND ANALYSIS

ASERTAINING THE USER REQUIREMENTS

TALKING TO THE USER

To begin the whole project process I began by emailing mr. Thomas with a brief explanation of what I am looking to do:

> Dear MR Thomas,
>
> I am an A-level student looking to produce a computing project, I am aware that this may not be the first time you have received a request like this, as my teacher Alan informs me that you often act as a client for projects, despite this I hope you will be willing to review my proposal as I am confident in its merit, particularly to someone with such a vast appreciation of complicated electronic nick-nacks.
>
> Did you ever play with Hornby or some other brand of model railway as a kid? Because I sure did! It was a good, carefree time in my life. my proposal is there for an attempt to revive the former, stress relieving, glory by attempting to attach the subtle science of railway signalling to something on a smaller scale by creating a railway that would be safe enough to automatically carry passengers from A to B with minimal loss of limbs.
>
> In principal it would be a similar model to that of the docklands light railway, that is, a train controlled by a computer off the train, using data gathered from the tracks about the location of the train and other obstacles, with the advantage of not actually carrying passengers which eliminates the necessity for a member of railway staff on hand to unjam passengers from the doors (of course the dispatch panel on board the train could be simulated externally for the full experience).
>
> you are probably wandering what this project could possibly do for you but picture this, the world industry is changing, for all you know your company may go into "railway related work" someday and it would certainly be swell to have a fully automated model railway to entertain waiting clients
>
> Should you have any questions I would be happy to indulge them,
>
> Michael Bell

His 1st reply was:

Hello Michael

Alan did tell me that you had a project that you'd like to do. As Alan probably told you we do like to display interesting projects in our entrance area and I've been very happy to showcase student projects before so your project definitely does interest me. A working train set in the office would definitely draw some interest especially if it has a significant control component.

What sort of control hardware do you have in mind? There's a lot of buzz these days about the Raspberry Pi which is an excellent piece of kit but I suspect that you should consider something more dedicated to control such as the Arduino range of boards. They're programmable in C and have a wide range of analog and digital inputs and outputs.

Alan has asked me to look at one of his other students' projects this year too - perhaps the two of you could get together and come down to meet us some time in the next week or so.

Cheers

-John

After this we began an exchange in which I sent mr. Thomas a series of questions and he replied with answers to each question, though not the original format, for readability I have separated out each question and the responses given to it. My questions / responses are written in Lilac and mr thomas's are written in brown,

- Would you like an interface to control the train and if so what would be your preference on what it is? LCD, computer program etc.
    - An LCD interface is fine, how many characters? how many lines? colour or b&w?
        - It would have 2 lines probably about 20 chars each
            - That's fine, but you'll need to specify exactly what's going to appear on each of the lines. I suggest for simplicity that you try to adopt a menu structure, but maybe you have other ideas.
                - No, a menu structure sounds good to me
    - A standard PC computer interface would also be great - you might find it sensible in your development to build a PC interface simply to provide yourself with a simple easy to change interface. Programming LCD displays is often trickier. But, clearly we'll need some sort of user interface in the final version and an LCD interface would be fine, assuming of course that its comprehensive and ideally intuitive. We'd like our visitors to be able to play with it when they're waiting for us and it needs to be self explanatory if that's going to work.
        - The lcd i had in mind is quite easy to program, it is designed specificaly for the arduino board and operating it is as easy as sending it a string of text
            - Sounds good - is it colour?
                - No its just 2 lines of black and white characters

    - Clearly we need to agree on what the user interface is going to allow us to do - that will require quite a bit of discussion.
        - I know you want it to allow selection of a destination, and to initiate the cleaning train, what else would you like it to do?
            -
- The track needs to be cleaned regularly as dust can damage the trains, would you like to clean it yourself or have a second train that cleans the track at regular intervals?
    - A secondary train is an excellent idea. Perhaps it could appear at various intervals ? It would be nice to control this from the user interface, i.e. set intervals, initiate cleaning and so on.
        - Thats fine though i should say that the abrasive pads on the bottom of the train must be changed by hand when they get clogged up with dust
            - OK - that's something that we can organise on a regular basis.
- Would you like the train system to turn itself on and off at set times? Eg. Running only during office hours
    - An automatic mode is an excellent idea - again a variety of scenarios chosen from the user interface would be good.
- Would you like the primary train to (in automatic mode) run itself to a scedual or pick stations at random
    - Both, set by the user interface.
- How many stops would you like?
    - How many can we have ?
        - I would say about 5 would be a good idea
            - OK.
- Would you like the trains to move relitivly quickly or slowly?
    - Various speeds - sometimes fast, sometimes slow, again set by the user interface - scripted would be good.
        - Ok i will
- Would you like electronic signals on the track? Eg. Lights, semaphores etc.
    - Definitely.
    - Can I please see some diagrams of how you're planning on laying out the track? Do you have any photographs? It would be good to get a really good idea of how big and complicated the system is.
        - I will have lots of time to do that this weekend
            - Could I see some please.
                - I will send you some pictures ASAP
    - What are your thoughts on user interface functionality?
        - I think that there should be a manual and automatic mode, the modes could be switched by a switch or a key, in automatic mode the screen displays the name of the prgram on the top line and what the train is currently doing eg. "Awaiting Right Ahead" or "slowing for stop in block 1"
        - In manual mode the program would have a simple menu struture allowing options like, setting what the train does in automatic mode, picking the next station for the train to reach, initiating cleaning, setting on and off times and checking that the track is clear.
        - When in automatic mode, customers could interface with the train using large colourful iluminated buttons rather than a booreing looking lcd display with options like "go" "stop now" "stop at next station" etc.
            - That sounds fine.

- What are you planning to use as your control hardware? I know that Leo is thinking seriously about using an arduino board - are you going down the same route or do you have some alternatives in mind ?
  - Same route

During this conversation mr Thomas requested some pictures of a basic track layout and the train so I captured these images and sent them to him:

This was his response:

> Thanks for these - I assume that the image of the track is the track you'd like to use. If so, it looks good but I'd like to understand what functionality you are thinking of including. By which I mean things like,

what is the function of the `orphan' pieces in the centre?

how many sensors are you planning on using?

how many points are there in total ? presumably you're intending to control the movement of the train by modifying the position of the points and controlling the speed and the direction of the train - that should do everything you want.

You mentioned a `cleaning' train. Presumably that will sit somewhere off the main track and move out into the central section when required, at which point presumably the train itself will have to move elsewhere - all of which will again presumably be under microprocessor control?

In order to understand exactly what you're proposing, I really would like to see a reasonably complete description of the functionality of the entire operation of the train line. Can you get this to me fairly soon so that we can internally make a final decision on whether the project is interesting enough to take up some of our foyer space.

Thanks

-John

REQUIREMENTS 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 stoped
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 gague) which is controlled by a microcontroller with the user interfase 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 manuver the train
4. the maximum voltage of the train should never be exceded
5. the train should appear to gain or loose speed smoothly

6. the tracks should be able to isolate a sideing 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 mater where it is it will always trigger a sensor on its journey
9. failsafes 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 reacertain the location of the train in case of a power failure by moveing it forward untill it trigures a sensor
6. The software must not perform any actions that could damage the train or any other equipment

DESIGN

NATURE OF THE SOLUTION

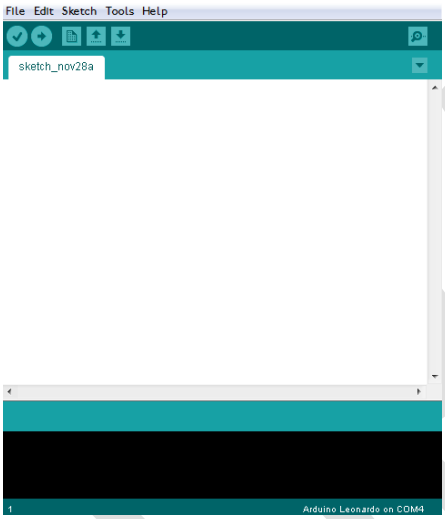
HARDWARE

PROTOTYPEING BOARD

The design process began with choosing a prototypeing board to use, I chose the Arduino Leonardo, depicted here.

This particular modle (the Leonardo) is the first Arduino board to integrate the main processing and the usb processing onto one chip, it is just as effective as the other boards but a bit cheeper

The idea behind getting a prototypeing board is that it can easily be programmed to control the train (with the adition of the motor sheild) which I will cover later



It also has digital imputs which can be used to receive information about the location of the train and send information to relays controling the points

Arduino uses an easy to grasp language and writing environment (depicted here) based on processing which was a beginners language created at MIT

MOTOR CONTROL BOARD

As well as the arduino board an additional piece of hardware is required to output to the motor as the arduino board has no analogue outputs and can only go up to a potential difference of 9V which is not enough to power the train at its maximum speed (12V).

There was only one clear choice for this, the Arduino motor shield (depicted here)

What it concists of is an imput voltage terminal and two analogue motor outputs that can control the output voltage, direction and apply eddie current breakeing by opening a switch between the terminals.

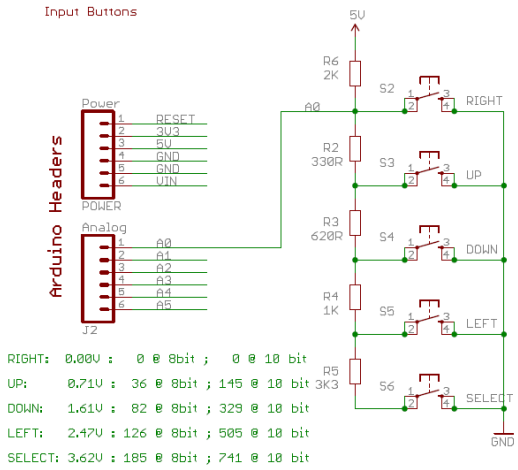
It is also very easy to use as it simply slots into the top of the arduino board and can be controlled simply by writing to certain pins on the arduino board

OUTPUT DISPLAY

The final piece of hardware I needed to decide on was some sort of output, in the end I settled on the Fretronics LCD Sheild, there where many lcd systems out there but this one stood out for several reasons.

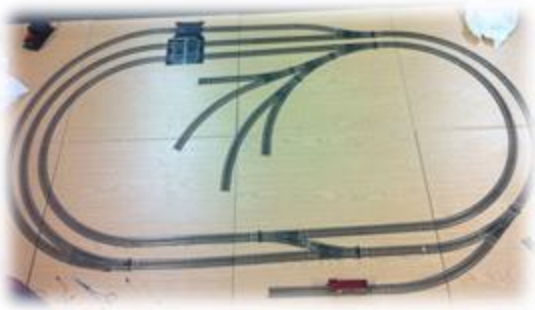
First off, it easily slots into the top of the motor shield, when the motor shield is slotted into the arduino board and interfaceing it is very easy as it uses a library built into arduino called “Liquid Crystal” and writing to it is as simple as setting the cursor in a defined location and then writing text to it in a line from that location.

Secondly, it has a backlight in a nice shade of blue which makes the display easy to read (the brightness of the backlight can even be controlled by the arduino board).



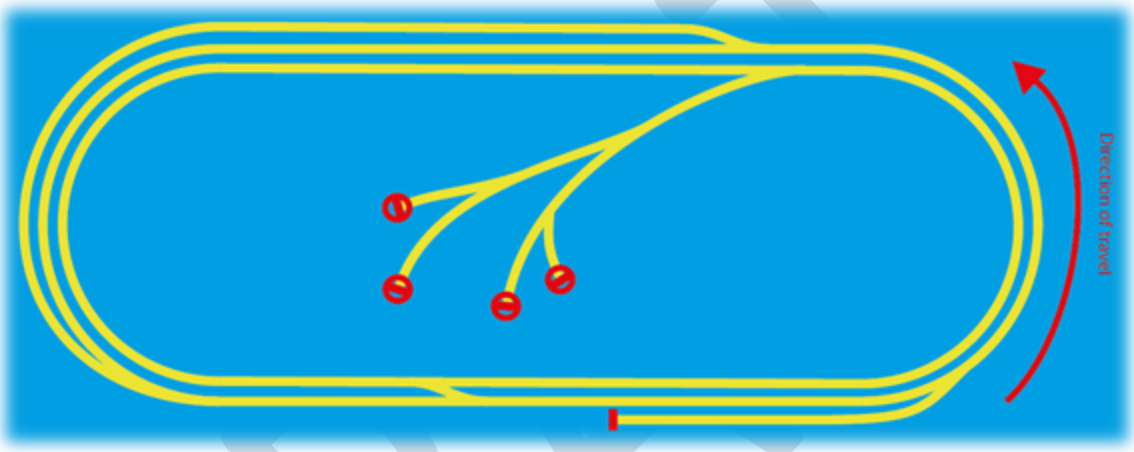
Thirdly the lcd shield has 5 buttons on it (up, down, left, right and select) which can be used to tab through a menu system to highlight and select items. In this case it also uses a clever resistor network (shown here) which means that rather than takeing up 5 digital pins the buttons only take up one analogue pin by outputting a different voltage depending on which button is pressed.

TRACK

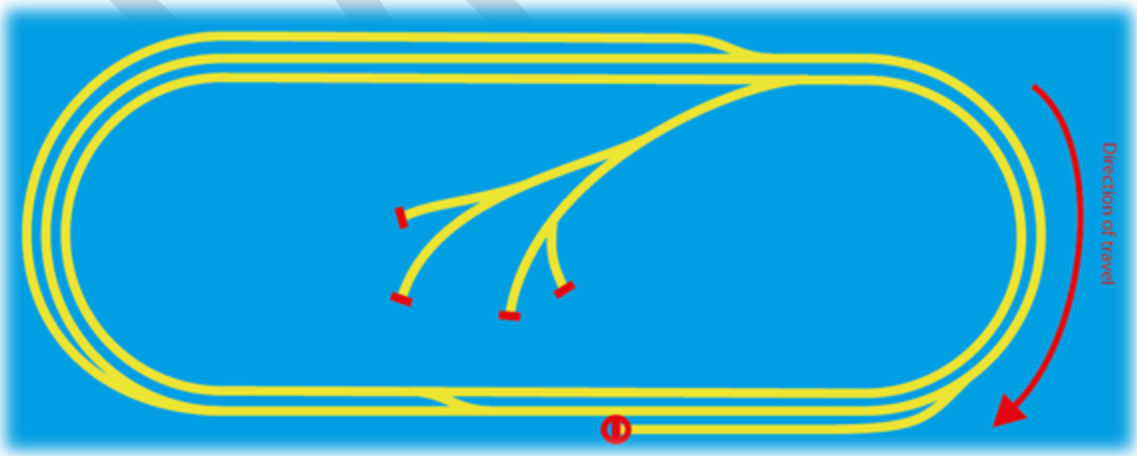


As I already own the track to be used in this project there was no decision making involved in choosing what type of track to use which is not a great loss as there is very little choice available out there and with the options that are available there is little difference between them.

Shown here is the basic layout that Hornby suggests using when constructing the track, however, the requirement specification dictates that *"the layout should be designed that if the train is moved forward that no mater where it is it will always trigger a sensor on its journey"* this is there for a good reason as I have illustrated below:



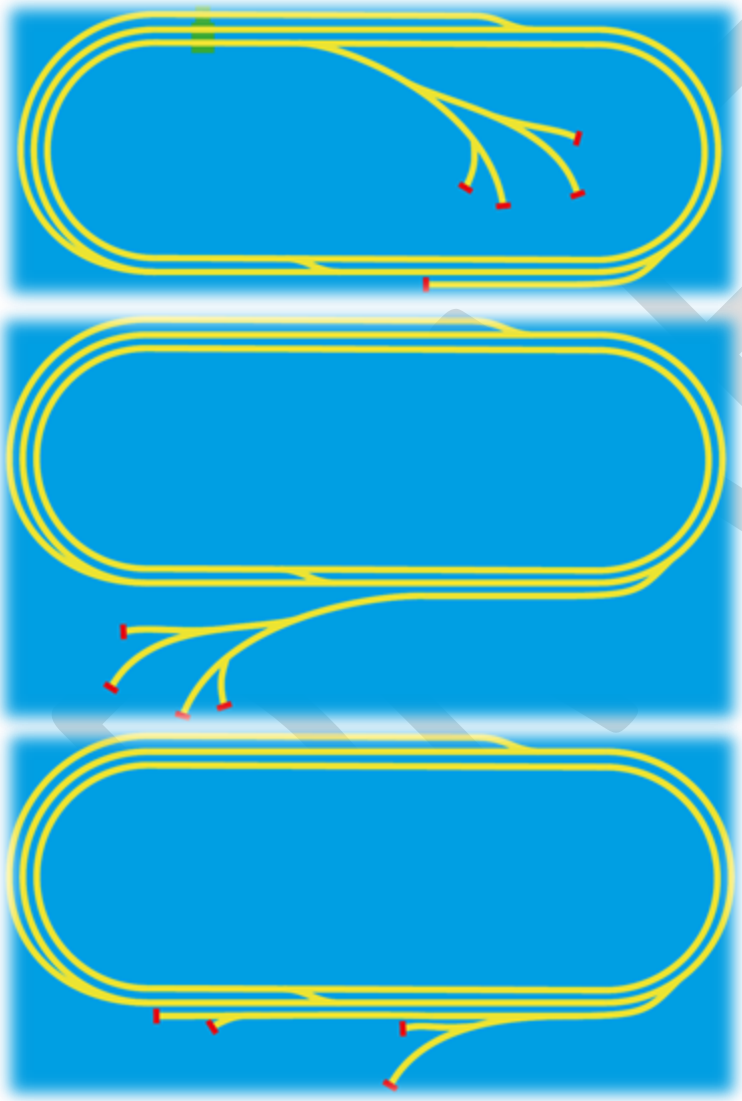
The Train is traveling clockwise around the track and, if the points are set right, it could arrive at one of the circled areas, if it does then the train will run up against a buffer and stop but the board could find itself unaware of the trains location. This is a problem if one of the sensors fails because the control board would loose the position of the train and be unaware that the train has hit a buffer, this would mean the train will keep running and grind its wheels into oblivion. If the train is traveling clockwise instead then this would happen:



The train can still end up in a red circle and therefore find itself in a similar situation of the board fails to realise where the train is so we need to do something to prevent this from happening. We have several options:

1. Design a track layout where the train cannot run up against a buffer in a given direction and only oppose the direction when apsolutly necessary, eg. Backing into a siding.
2. Program the board to recognise when a train has skiped a sensor by noticing that the sensor after the failed sensor has been tripped and then
    a. Compensate by skiping to the instruction set on the next sensor
    b. Stop the train if it should have stoped at the previous sensor
    c. Display an error message telling the user to replace the sensor
3. Program a failsafe into the board in which if the train has been traveling for a given time without tripping a sensor it stops and displayes an error message.

These are all good options so I will try to implement all 3, the combination should prevent this problem from ever happening, here are a few designs I have chosen that conform to point 1



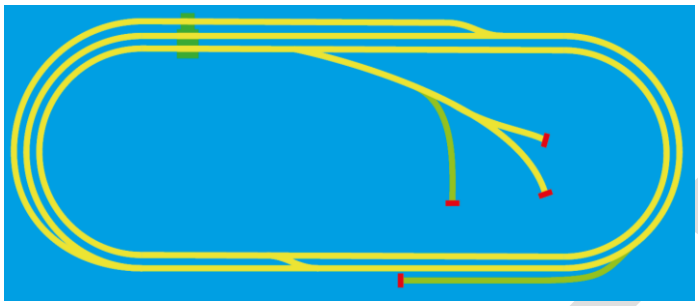
In all 3 the track is designed for the train to travel anticlockwise normaly but all 3 could be reflected for the train to travel the other way

The issue with the lower two is that they both take up quite a bit of space so I chose to go with the top design.

ISOLATABLE SIDINGS

Another important element of the design is controlling which of the two trains (the main train and the cleaning train) are active and which is waiting in a siding.

My solution to this problem is to use relays, a relay is a device which can be switched on or off by sending a current to it. The idea behind using these is to move the train onto a separate piece of track that can then be used, if the design incorporates two of these then the control board can isolate one of the two trains on one of these pieces of track while the other runs.



What could happen is that the two sidings highlighted in green are separated from the other track by a tiny gap and so do not share current with them, the control board supplies them with power separately which can be switched on and off at will. When the control board wants to switch trains it backs the train to go offline onto the siding on which it belongs, then isolates that piece of track, it then sets the points for the next train and then de-isolates the piece of track that that bit is on and drives the train off it.

LOCATION DETECTION

An important part of the design is deciding how to detect the location of the train on the tracks without interfering with, or risk derailing the train, I had many different options on how to do this which I will go through.

The best place, in my opinion, to draw inspiration from is full scale railways themselves. The simplist and oldest viable method is using something called a treadle (shown here).

What happens is, the train passes over the treadle and the rim of the wheel presses the metal bar (highlighted). The bar is basically an electrical switch and when pressed down it sends a signal to a signal box. In the days before computers these where widly used in combination with relays to control the signals and tell the signalman the position of the train.

The issue with using this on a much smaller scale is how fiddley and easily bent the metal bar that the rim touches would be if it was scailed down to 00 gague which is the track size so this option isn't really feasible.

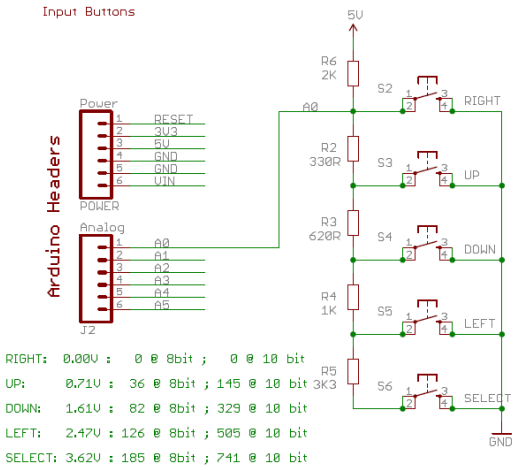
Another option is to use something called track circuits, these are created by running electric current through tracks which is spilled into other tracks by the train wheels, while this is a cheep and effective solution on real railways, they would be very difficult to use on a Hornby set because the track must be a complete circuit for the train to move.

A third option is to use some kind of magnetic detection, these too are used on a grand scale. This box shown here is imaginitivly called a magnet, it can detect when a train passes over it using magnetisim as well as measure the speed of the train and tell the cab computer what the signal ahead is showing.

Realisticly of course, we don't need it to do anything more than detect the location of the train but the idea of using magnetism appealed to me a lot so I did some research and discovered an easy way of using it to detect the location of the train.

This is a hall effect switch. When the south pole of a magnet passes over the upturned face it allows current to flow between the middle and right hand terminals (the left is used to power it). One of these isn't much use but if many of them where combined in a network throughout the track they could send an electrical pulse to the prototypeing board every time a train passed over them, allowing the board to keep track of which block (a given section of track) the train is in without ever touching the train or interfering with the current through the train. Not only that but they are tiny and so easily fit in the small gap between the track and the train. For these reasons I chose to use hall effect switches to locate the train.

The next decision to make then was how to provide the magnetic field needed to activate the switch, it needed to be strong enough to breach the gap between the train and the tracks but the magnet needed to be small enough to not touch the switch or anything else. In physics when one thinks "powerful perminant magnets" the next thaught is always "Neodynium" a rare earth metal which when used in an alloy with iron and boron produces extremely powerful magnets, these magnets are usualy quite expensive but as I needed them to be small they wouldn't cost that much, show in an image of the magnets I have chosen, they are 3mm thick, perfect for attaching to the train (there is a gap of about 7mm between it and the track)

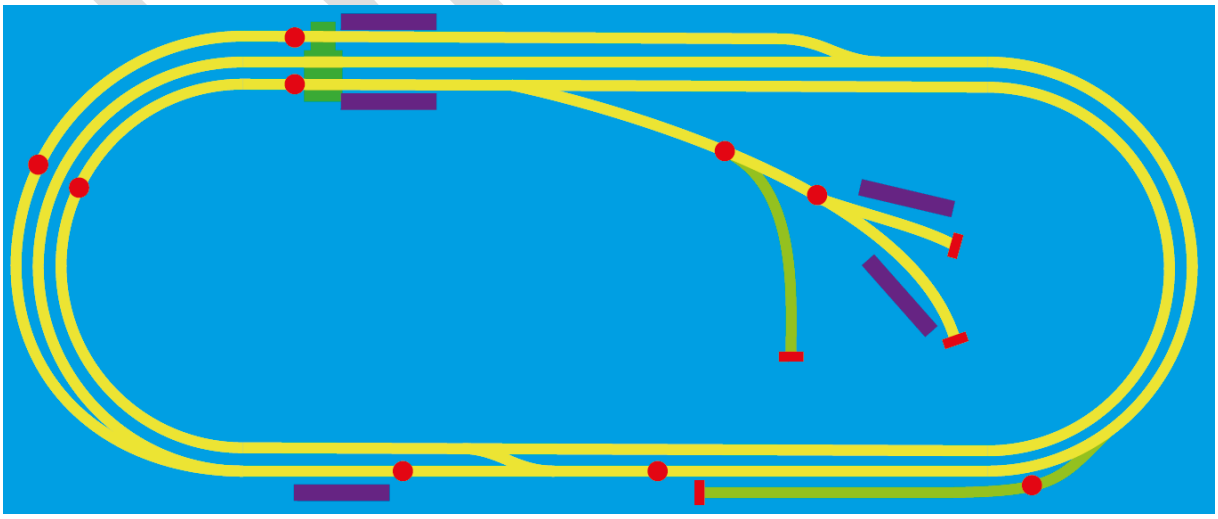


After deciding to use the Hall Effect switches I then had to decide how to receive information from them with the prototypeing board, this was important because there wouldn't be enough inputs on the board to receive data from each sensor indevidualy, luckily though, I could draw inspiration from the opensource design of the lcd display, which took the inputs from 5 different buttons but only used one analogue input plug to do it. Here is the section of the schematic showing how it worked.

The idea is that it outputs a different voltage depending on which button is pressed, all I had to do was replace the buttons with the hall effect switches and read the analogue pin to find out which switch is being pressed, the code to do this is shown below:

This can easily be adapted to be used in Beltrak.

The last phase in the design was to decide where to place the sensors on the track



In this image the purple bars represent stations, the green track is an isolatable siding and the red dots are sensors.

SOFTWARE

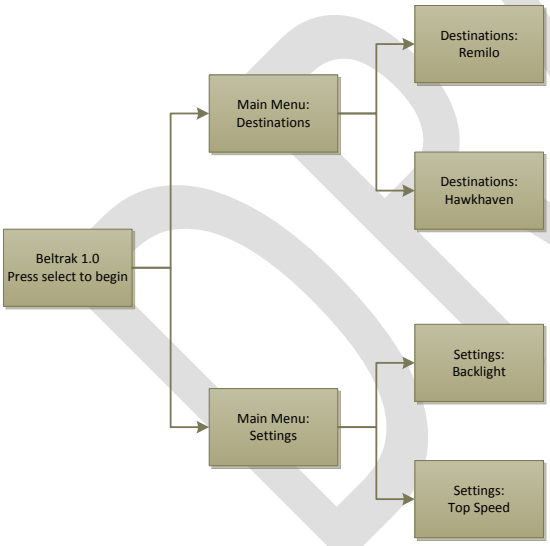
DESIGN OBJECTIVES

After much deliberation I decided to program the arduino board to run as a finite state machine.

This means that the machine will be in a fixed state, for example, speeding up, and is programed to move into another state when a certain triggering condition is met for example "reached maximum speed" at which point it would switch into a different state eg. "traveling at constant speed"

This has several important advantages, most of all the fact that the requirements for changing state can be programmed into an array, the array can be edited to allow for changes in sensor positions or track arangements without having to change the base code. This means that the software could potentially work for any track arrangement simply by reprograming the array. It also saves writing out the code repeatedly for diferent conditions as the array makes the code adaptable

I have also decided on using a simple menu structure for the LCD display that resembles a tree with branches:



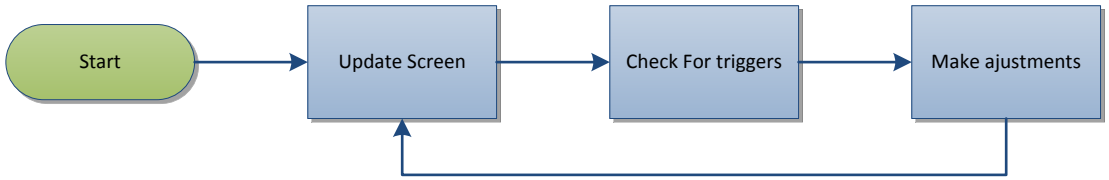
Navigation through this system should be easy enough, the arrow keys navigate the paths and the select key chooses the final option.

This method was outlined in the requirement specification and I still believe it to be the best way of providing a menu within the constraints provided, that is to say, a 16 by 2 lcd display with 4 arrow keys and a select button.

There will also be ceartain other screens displayed, for example when its starting up a loading screen might be displayed and when the train is running a progress screen might be displayed.

PROCESSES AND MODULES

OVERALL PROCESS



This is the full process for the Beltrac software, the process is looped over and over again, endlessly from power on to power off. The code (and therefore the design) can be devided into these 3 sections.

UPDATE SCREEN

DATA STRUCTURES

MENU STRUCTURE ARRAY

The menu structure array is a 3 dimensional array in which the text for the display and the sectional instructions are stored based on their position in the structure, as shown here:

Position	[0][]	[1][]	[2][]
[][0]	Welcome to beltrak	Destinations	Hawkhaven
[][1]	~	#	Remilo
[][2]	~	#	~
[][3]	~	Settings	Backlight
[][4]	~	~	Top Speed

This table shows the contence of [][][0], [][][1] and [][][2] the first of which contains a 1 if the table displays a tilde, a 2 if the table contains a hash and a zero if the cell contains text. The second and third contain the first and second lines of text to be shown on he screen.

To understand this system imagine starting on the welcome to Beltrac cell, if you press an arrow key and you don't go over the edge of the table you move in that direction. If you reach a tilde you move back one in the direction you came. If you reach a hash you move forward one in the direction you entered the cell. When you reach a piece of text that is displayed on the screen.

This is a rough outline of the array as the full version would be much more complicated.

SWITCHING CONDITIONS

The conditions for switching between different states will be defined by a three dimentional array containing encoded conditions.

The idea behind this is that when the board initiates a journey it looks up the conditions in the array where they are stored for a particular destination or action.

In the array the 1st dimension is the number for a destination or instruction for example [1][x][x] might contain directions to get to Hawkhaven, [30][x][x] might send out the cleaning train.

The second dimension contains the position in the instruction set for example [x][0][x] is the first instruction, [x][4][x] is the fifth instruction.

In the third demotion, position [0] is the condition, [1] is the value of that condition, [2] is the state to change to when hitting that condition, [3] is the value of that state, for example:

[0]	[1]	[2]	[3]
B	4	S	1
In block	Four	Set speed to	Setting one

Conditions:

Code	Condition
B	In block x trigger this condition, where a block is when the train is over a sensor and x

	is the number of that sensor
W	Start a timer for x miliseconds, trigger when the timer ends

States:

Code	Action
S	Set the Proportional Potential Difference and therefore the speed to x where x means: [X] → PPD 0 → 0% 1 → 50% 2 → 100% -1 → -50% -2 → -100%
C	Set point x to converge
D	Set point X to diverge

IMPLIMENTATION