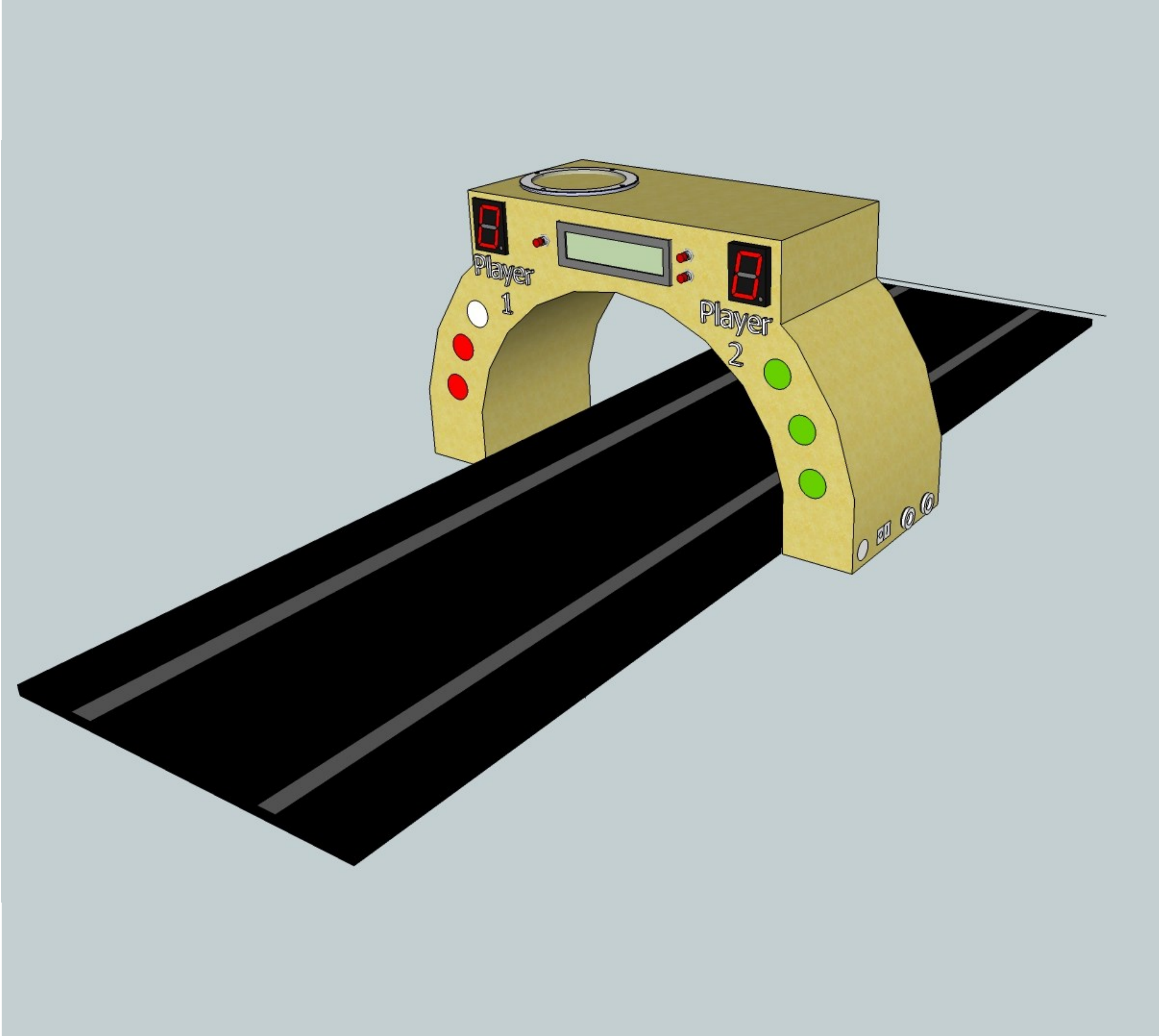
Jacob Burge - 9030

Electronics Coursework

Scalextrics Start Gate

Contents

Contents



Product Brief and Analysis

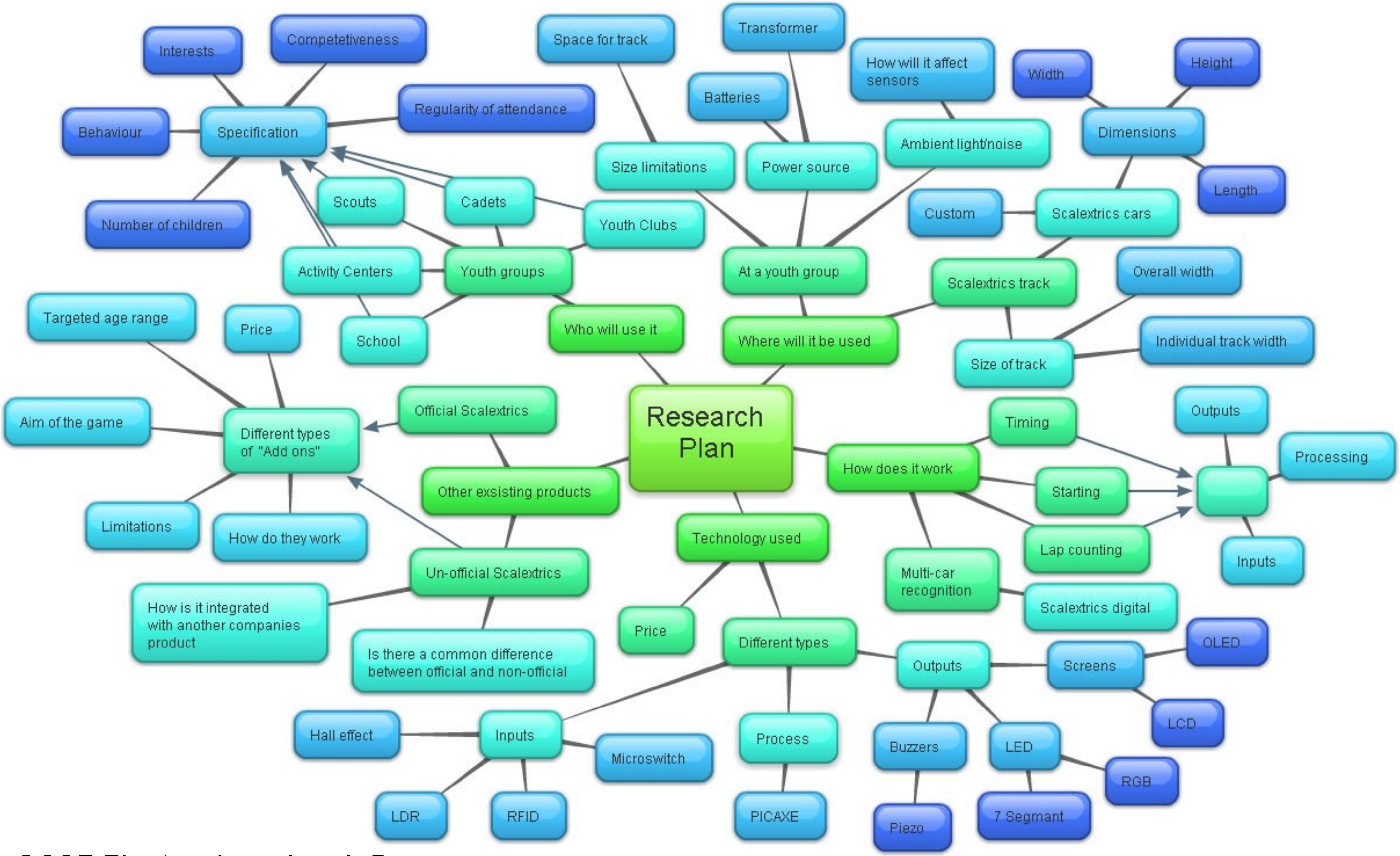
Context: Racing of ‘Scalextric’ style cars or remote control cars is a popular event in after school clubs, and a reliable method of starting and measuring performance would enhance the activity.

Task: Design and make a starting gate which counts down for the ‘Scalextric’ version of a Formula 1 grand prix. It can have both visual and audible outputs, and the timing sequence can be physically or remotely activated.

Brief: To design and build an electronic start gate, for use with any ‘Scalextric’ set. It will be simple to use and child friendly, but fun for all.

http://www.clownshobbies.co.za/

Research Plan



Existing Products

http://www.scalextric.com/shop/track/accessories/c8215-lap-counter-and-timer/

http://www.scalextric.com/shop/track/digital-track/c7042-advanced-6-car-digital-powerbase/

http://www.scalextric.com/shop/track/digital-track/c7041-digital-pit-lane-game/

Lap Counter and Timer — £26.49

For only £26.49 this is a good product, it provides you with lap counting up to 99 laps and will tell you your fastest lap. It is made out of good quality plastic and the buttons are very sturdy. It is aimed at any existing scalextrics owners as it is an add-on product. It also looks very good, featuring a carbon fibre effect on the outside shell; this can be very appealing to younger audiences. It works by featuring micro switches built into the track, these are pushed in and triggered by the cars guide. Overall this is great value for money.

Scalextric Digital Advanced 6 Car Powerbase — £149.99

This is quite a step up from the other Lap counter although it comes with many more features. It allows up to 6 cars to race at one time and power levels can be defined for each player to allow handicap racing. Another impressive feature this has to offer is ghost cars, allowing you to race the computer in real! This product is aimed at the more serious Scalextrics racers but definitely delivers. I think it works using RFID as it is part of the digital range which allows more than one car on each lane. If this is not the case then it must be a different form of unique identification.

Digital Pit Lane Game — £44.99

I think this product could be aimed at younger children, trying to make the scalextrics experience more fun. As a result it has to be safe; from the picture we can see there are no sharp edges or small parts young children could try to eat. It is a relatively small add on product, but adds a whole new dimension to the existing product. It means the race becomes more tactical and more skill is required. It is also part of the digital range, so I assume it works using RFID. I think it has been styled very effectively and looks as realistic as it could get, especially the red and green hanging

Associated Products

http://www.coolest-gadgets.com/20071123/something-a-swimmer-can-count-on-for-a-good-workout-finis-lap-track/

Lap Counter for Swimming— £51.40

This device is fully waterproof and sticks to the side of a swimming pool. It is designed to help swimmers with there train by providing length counting features and even timing. It has a plastic design, allowing it to be very durable. It features big buttons as the user is doing sport and may struggle to press a fiddly little button. It has multiple mounting methods which makes it a very versatile product, making it excellent value for money. In terms of safety the product has no sharp corners which the swimmer could injure themselves on, so it is very safe.

http://www.get-a-gatecustombmxgates.com/index.php?id=94

BMX Start Gate – From £1700

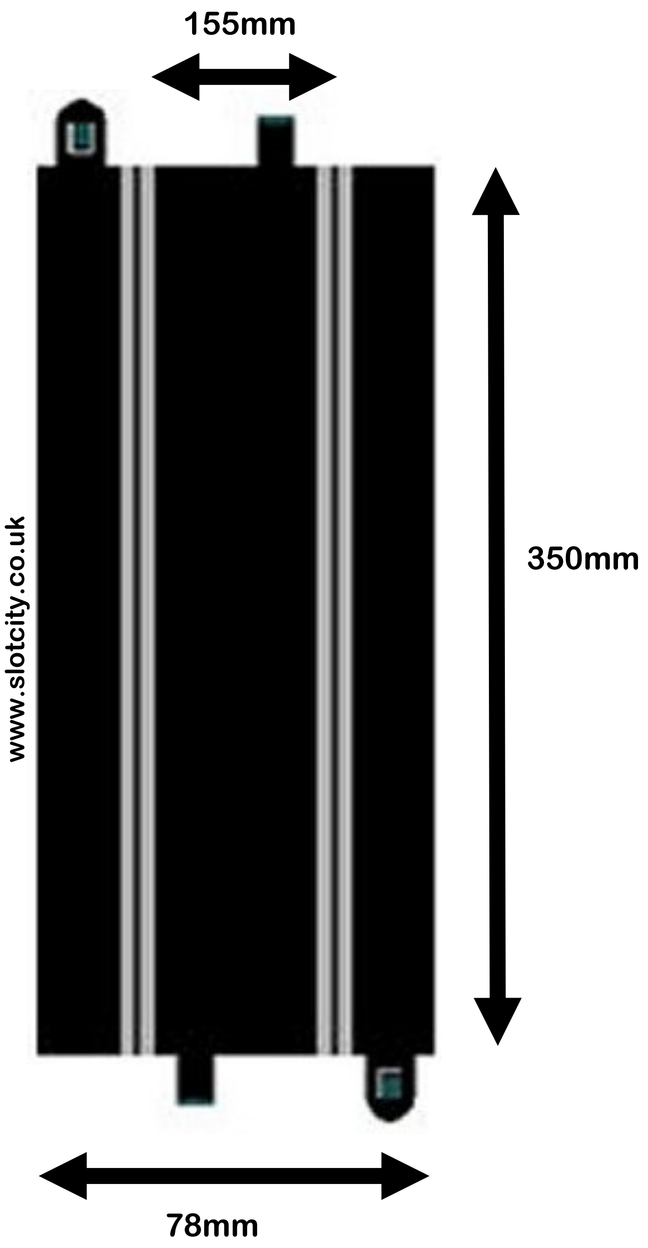
This is part of a kit to build a BMX start gate. This features a mechanical barrier which prevents riders from starting until the countdown sequence has finished. It provides both visual and audible alarms relating to the start time. All the parts are either made of durable plastic or metal and is designed to give enough strength to support the riders. I personally wouldn't choose this product on its looks, however function should come over looks. I think to improve the product it could be more in integrated so that there is less to install on race day.

http://www.mindsetsonline.co.uk/product_info.php?products_id=1010055

Rail Race Timer — £71.94

This timer records times of a model drag race. The timer is triggered the second the start light goes green and then stops the timer when each car crosses the line. It is accurate to 1/100th of a second which I feel is a suitable degree of accuracy for its purpose. The product comes with a neat case which can be constructed around the circuitry or a case can be designed for it. This is a very good feature for the target audience, schools. I think it is also great value for money as it is able to record two times simultaneously to such a degree of accuracy.

Environmental Limitations



McLaren MP4-12C GT3 — £39.99

Overall length - 140 mm
Wheelbase - 83 mm
Weights - Car 86 grams; Case 156 grams
Scale - 1/32

http://www.scalextric.com/shop/cars/high-detail/c3389-mclaren-mp4-12c-gt3/

Ford Fiesta RS WRC — £37.49

Overall length - 123 mm
Wheelbase - 77 mm
Weights - Car 80 gm
Scale - 1/32

http://www.scalextric.com/shop/cars/rally/c3284-ford-fiesta-rs-wrc/

Porsche RS Spyder, Team Essex — £39.99

Overall length - 147 mm
Wheelbase - 91 mm
Weights - Car 82 gm; Case 156gm
Scale - 1/32

http://www.scalextric.com/shop/cars/endurance/c3197-porsche-rs-spyder/

Input Components

http://www.rapidonline.com/Electronic-Components/Digital-Position-Sensor-Hall-Effect-Sensor-Honeywell-519707

Digital Position Sensor/Hall Effect Sensor — £1.94

Scalextrics cars have magnets to hold them to the track, therefor this could be used to determine when a car crosses the finish line. However it would have to be calibrated making programming slightly more challenging. Also, interference could be an issue, if both cars finish at the same time the magnetic fields may effect each other. This could be a cheap solution though.

http://www.rapidonline.com/Electronic-Components/Distance-measuring-sensor-digital-output-81725

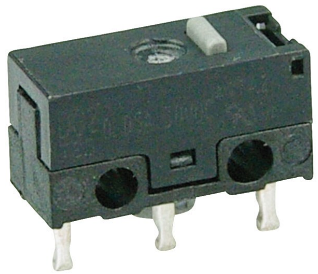
Distance measuring sensor - digital output — £10.54

Like the ultrasonic sensor this gives a digital output of the distance to an object. It works differently though and therefor this module is slightly cheaper. Instead of ultrasonic it uses infra red. There could be a much cheaper way to do this however, using just am infra red led in the track and a sensor overhead the same result could be achieved.

http://www.rapidonline.com/Electronic-Components/Ultrasonic-Range-Finder-SRF05-82276

Ultrasonic Range Finder SRF05 — £17

This sensor is ideal as one could be placed above each lane in the start gate. However these modules are not cheap at £17 each it would make for an expensive solution. It maybe possible to use only one however if two cars were to finish at the same time there would be no way of telling them apart. They are also quite big and require complex programming, both disadvantages.



http://www.rapidonline.com/Electronic-Components/Cherry-Sub-Subminiature-Microswitches-DG23-Series-520207

Cherry Sub Subminiature Microswitches — £1.14

A mechanical switch is the last option. This micro switch will trigger with little force and there are many attachments compatible with this switch. The device would have to sit next to the track and physically knock the car. This could cause the car to run off the track and also may damage the car over time. This would be something the customer not be happy with.

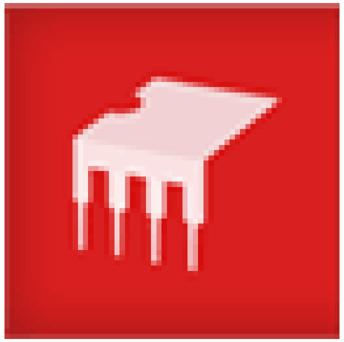
Process Components

When deciding upon processing components there are two main choices — Picaxe and GENIE. Both have there own advantages and disadvantages:

www.gatewaymedia.net

GENIE Microcontrollers

- You can use basic or flowchart
- You can simulate in circuit wizard
- Some commands are simplified in circuit wizard
- You can still use Picaxe hardware with it



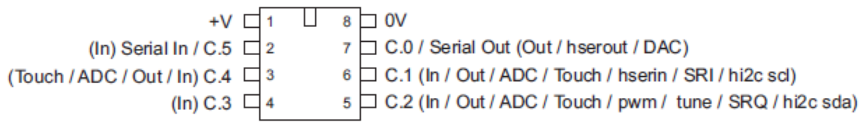
http://www.picaxe.com/

Picaxe Microcontrollers

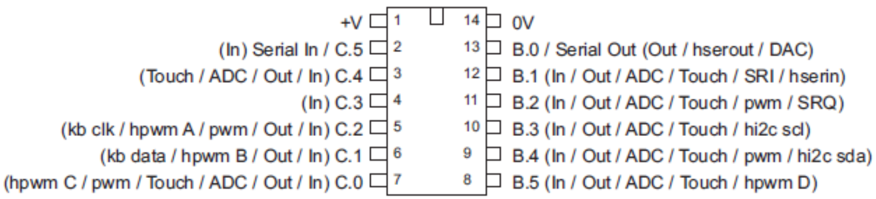
- Greater range of options
- You can use basic or flowchart
- Widely used meaning lots of support online
- Lots of useful resources

After choosing one of these there are then more choices to make. Depending upon your pin requirements you must choose a suitable unit. Here are just a few of the various chips:

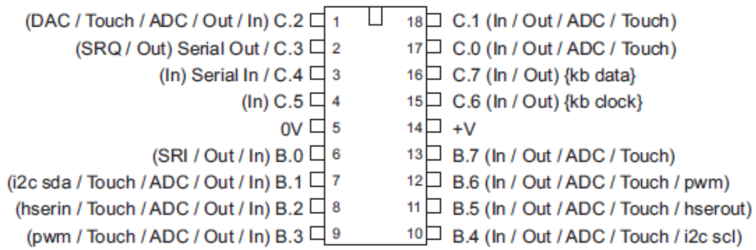
PICAXE-08M2



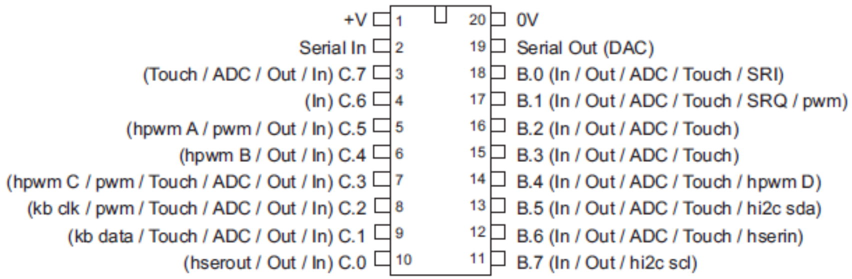
PICAXE-14M2



PICAXE-18M2



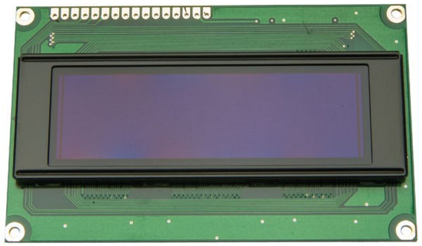
PICAXE-20M2



http://www.picaxe.com/docs/picaxe_manual1.pdf

As a result of the wide range of choice, the final specification need to be decided upon before the chip can be selected. However simply because of the high level of support material, I am fairly sure I will choose a PICAXE chip.

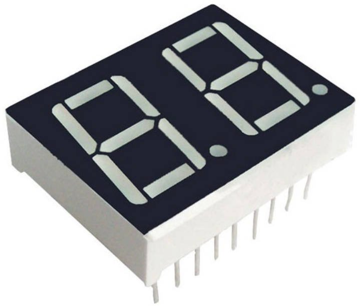
Output Components



http://www.rapidonline.com/Electronic-Components/20-x-4-OLED-Alphanumeric-Display-500186

20 x 4 OLED Alphanumeric Display — £26.60

Although expensive this display offers easy control over the Serial interface. It can display 80 characters in total which allows a lot of information to be displayed at any time. As this display is OLED it also means no backlight is required and it can still be read easily in darker places. However this display is expensive — possibly to costly for the target market.



http://www.rapidonline.com/Electronic-Components/14-2mm-Dual-digit-displays-200090

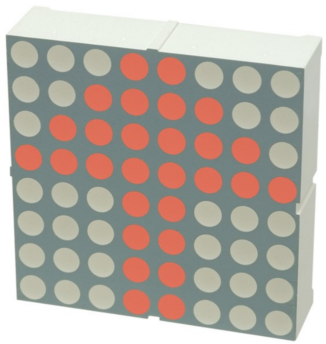
14.2mm Dual digit displays — £0.74

A lot cheaper compared to both the OLED and LCD display, however this display requires either a lot of digital I/O pins or a dedicated driver. This leads to more complex programming as well as circuit design. Another disadvantage to using this display is the number of characters that can be displayed—only 2. Also, this is limited to numbers so letters and symbols cannot be displayed.

http://www.techsupplies.co.uk/epages/Store.sf/en_GB/?ObjectPath=/Shops/Store.TechSupplies/Products/AXE133

Serial LCD Display — £6.99

Unlike the OLED display, this display must have an extra backlight to allow it to be read in the dark. Another disadvantage is the viewing angle is significantly less. However it is a cheap solution and for the price of an OLED you could use 3 of these, thus allowing more information to be displayed. Like the OLED version this display also operates using the Serial interface making it easy to control.



http://www.rapidonline.com/Electronic-Components/38mm-Dot-matrix-display-72954

38mm Dot matrix display — £4.60

Like the seven segment display this is a cheap solution, however relative to the Serial LCD display it is not worth the saving really. Although numbers, symbols and letters can be displayed you are limited to around 2 per display. Also this type of display would require complex programming along with circuit board design. One advantage however would be the size, making it a candidate for displaying the countdown to the start.

Power Options

There are three options for powering my start gate: Batteries, from the track or via a transformer from the mains. Current scalextrics product use all three in there multiple products.

http://www.scalextric.com/shop/track/accessories/c8215-lap-counter-and-timer/

http://www.scalextric.com/shop/track/digital-track/c7042-advanced-6-car-digital-powerbase/

http://www.scalextric.com/shop/track/digital-track/c7039-digital-lap-counter/

www.drinkstuff.com

www.slotcity.co.uk

http://www.scalextric.com/shop/track/accessories/p9200-uk-wall-plug-transformer/

Powered by batteries

This would be an effective method however with lots of sound and displays they would have to be replaced quite often and they are relatively expensive. Also room for a battery box has to be left inside the casing.

Powered from the track

Powering the start gate from the track may reduce the power provided to the cars making them move slower, something the target market would not be impressed by. It would also require a piece of track to be integrated with the start gate.

Powered from the mains

This would eliminate any problems in which the scalextrics cars were slowed down, however it would mean two mains supplies would be required and therefor there would be lots of cables everywhere. This could even be a hazard with people jumping around to put cars back on the track.

Target Market — Customer Profile

Category	Profile	Affect on Product
Age	11 — 16	This age range is going to mean the aesthetics do not have to be smart, more casual. Also it can feature more complex features as teens are one the most technologically able generations.
Environment	Secondary School	Although aesthetics do not need to be perfect it should be fairly durable and tough, but also as compact as possible so it can simply be shoved in a cupboard at them end of lunch time.
Buyer	School Teacher	Because it is designed for an in school purpose, the product must appear to be educational to allow the teacher to justify the purchase. Due to recent budget cuts it must also be fairly cheap.
User	Teacher / Student	The main user will be pupils from the school who attend the club, therefor the product should include a feature to prevent cheating through false starts.
Track Type	Scalextrics	Due to the design brief this product is going to be specifically designed for scalextrics track. This means it must fit the size limitations of the track and cars.
Driver Ability	All	Because the club must be open to all, the start gat must be able to cope with massive variations in ability — so could possible include a handicapping system.
Time to Play	Lunchtimes	As the product will only be in operation for an hour at lunch time it must be quick and easy to setup. This will also dictate the required size for lap counters etc.

Initial Specification

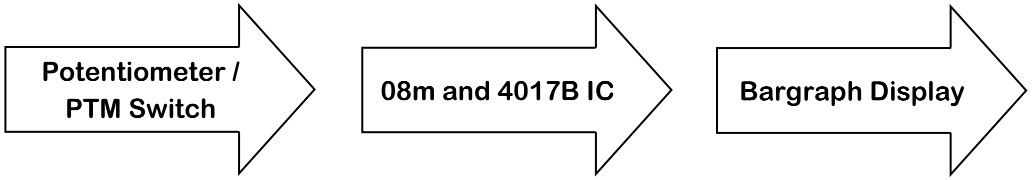
My product will...

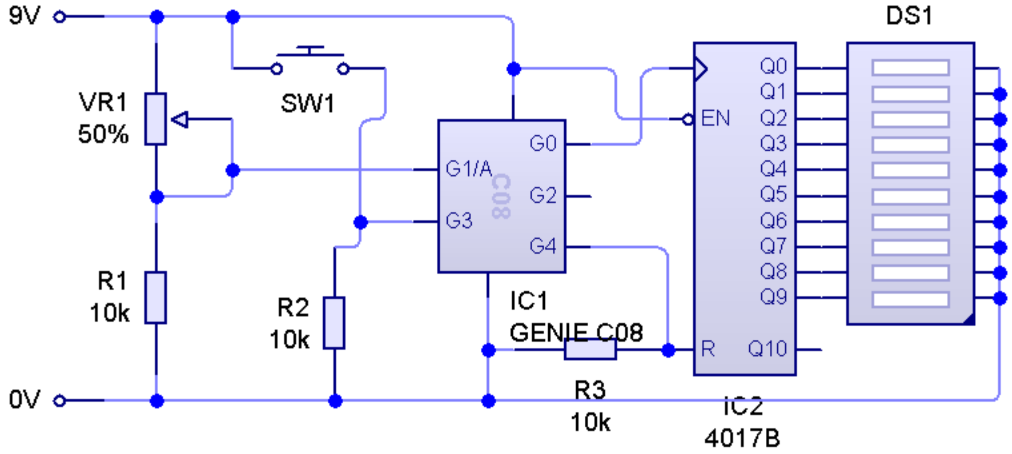
- Be designed for use by 11 — 16 year olds
- Be compatible with standard Scalextrics track and cars
- Cost between £40 — £50 for the consumer to buy
- Be aesthetically pleasing for the target market
- Be safe for use in school
- Be able to store user profiles and appropriate handicaps as well as user information such as fastest lap time
- Use light and sound to signal the countdown sequence to both the cars and players to simulate a real racing experience.
- Use a form of display to show relevant information
- Be powered from the mains
- Record laps up to and including 99
- Record lap times with a maximum of 999 seconds
- Have a timer with accuracy of 0.01 seconds
- Be able to record times and laps for two lanes at once.
- Use a durable material and be a compact size
- Include a form of false start detection

Circuit Ideas

Lap Counting

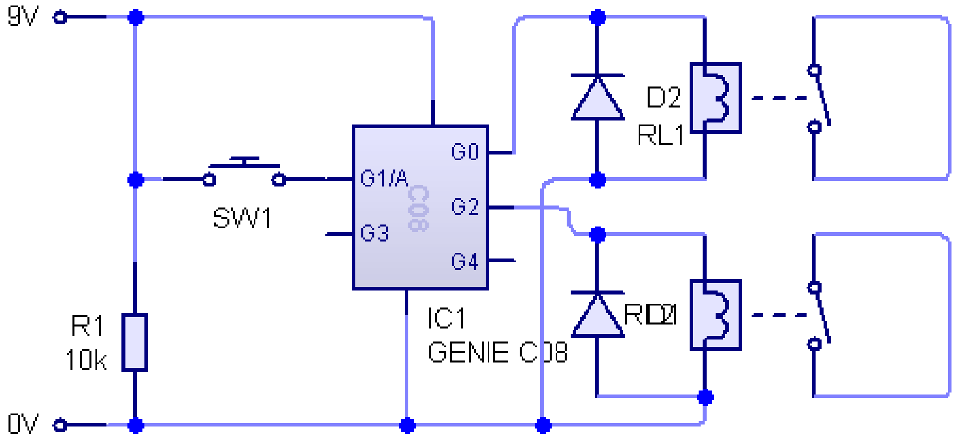
To allow the user to select there profile, this circuit uses an 08m to read the value of a potentiometer. The result from the potentiometer is then displayed on a bargraph display using a 4017b IC to save pins on the PIC chip. Once the user has selected there profile, they press the enter button and the lights on the bargraph display will perform a pattern to confirm the selection.

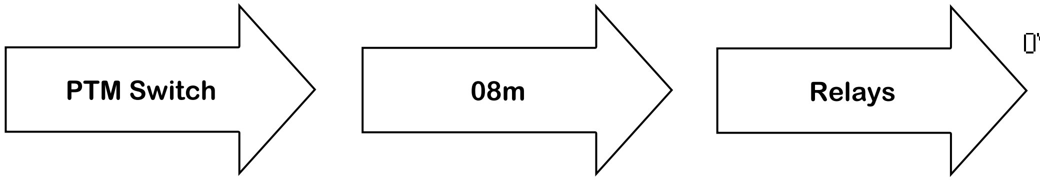




False Start Prevention.

This circuit will cut of the power supply to the track even when the trigger on the controller is pressed by breaking the circuit to the controller. This circuit could be done analoguely or digitaly, making it digital makes it easier to communicate with the rest of the circuit. Hardware wise it only requires a 3.5mm audio jack and socket to hack the scalextrics setup.

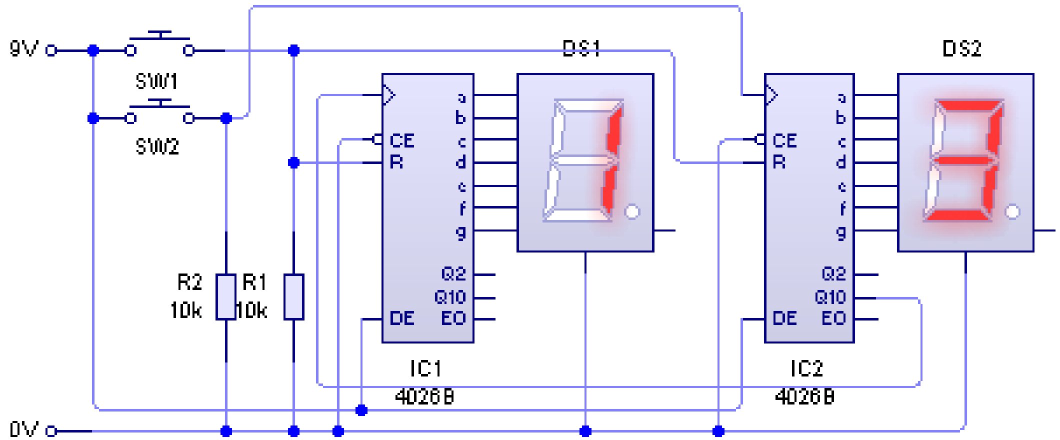




Circuit Ideas

Lap Counting

Lap counting is one of the key parts of the scalextrics start gate, one way this could be done is by using a couple of 4026 seven segment drivers. A micro switch would be used next to or under the track and this would clock the driver IC. The current lap is then displayed on the seven segment displays as shown in the circuit diagram. This could also be used with an analogue sensor in line with a microcontroller or op-amp to give the correct output to the seven segment driver.



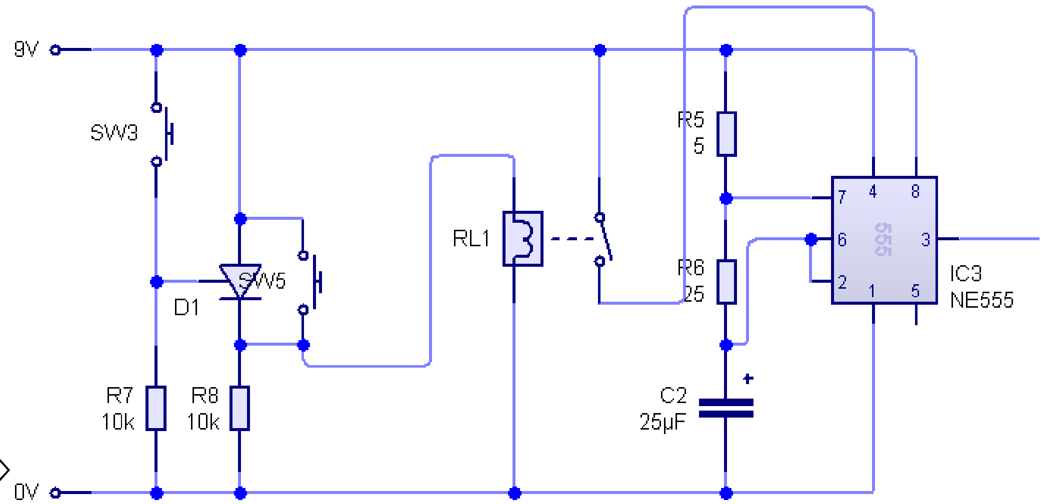
Push To Make Switch

555 IC and 4017B IC

Red and Green LED

Timing

Similarly to the lap counting, timing could also be achieved by using a micro switch to trigger it; however for this circuit design two micro switches are required, one to start and one to reset the thyristor which starts and stops the timer. The timing device would be a 555 timer IC with 2 resistors and a capacitor to create an astable circuit. This circuit would then be attached to two or three 4026B ICs to drive seven segment displays in order to display each lap time.



Push To Make Switch

555 IC and 4026B IC

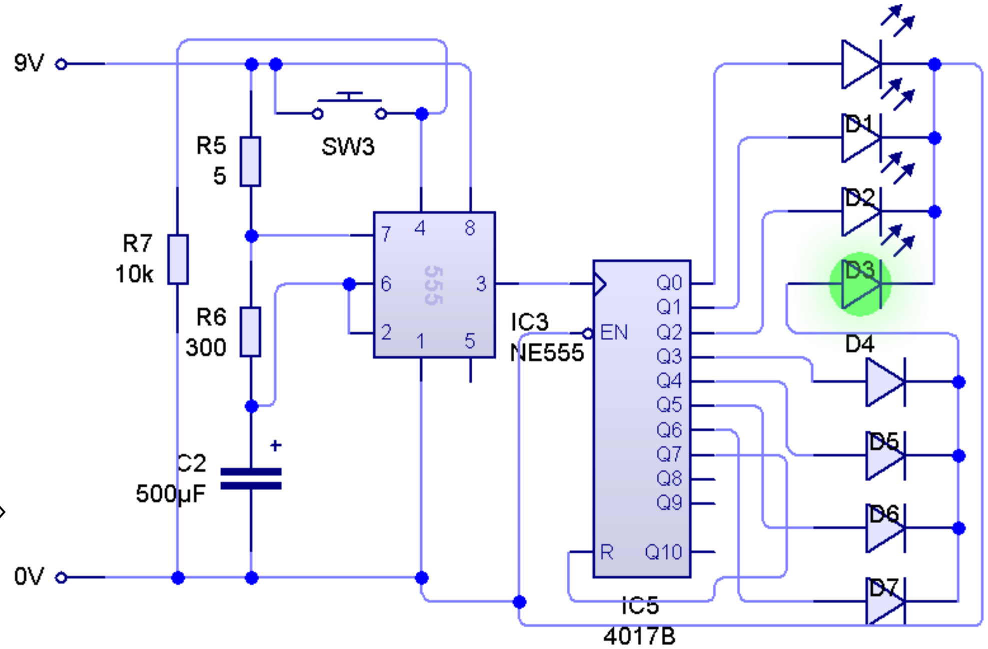
Seven Segment Displays

Circuit Ideas

Starting

The final element to a basic start gate is the start. This circuit requires the user to hold down a PTM switch for the duration of the starting sequence and the led's display a 3 second count-down. This is done by using a 555 timer IC to provide a clock for a 4017 IC Decade Counter which lights 3 red led's and then a single green led for 4 times the length as the reds ones are on for. It then resets itself and returns to the first red led output. It could also be easily adapted to start the timer.

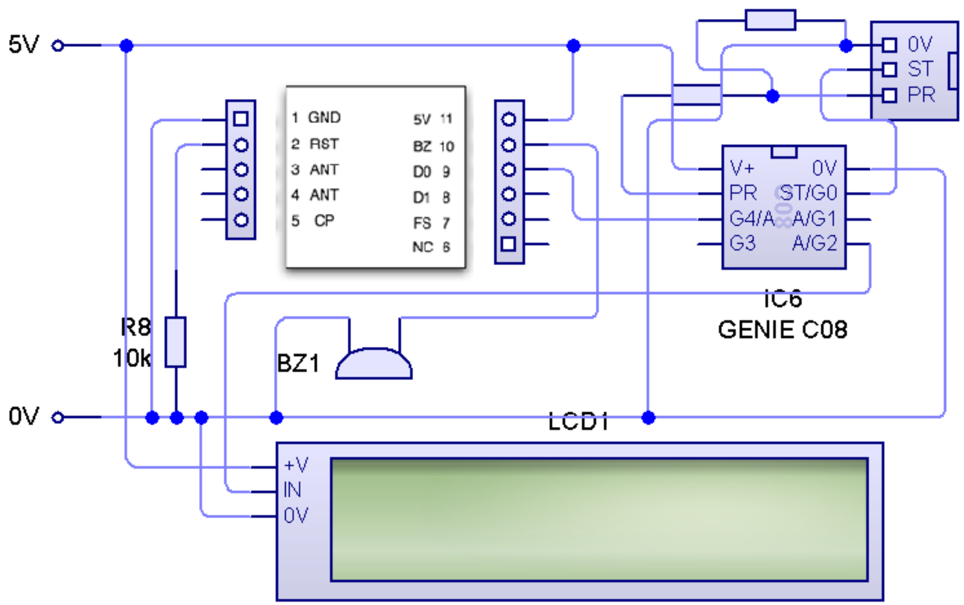
Push To Make Switch → 555 IC and 4017B IC → Red and Green LED



Starting

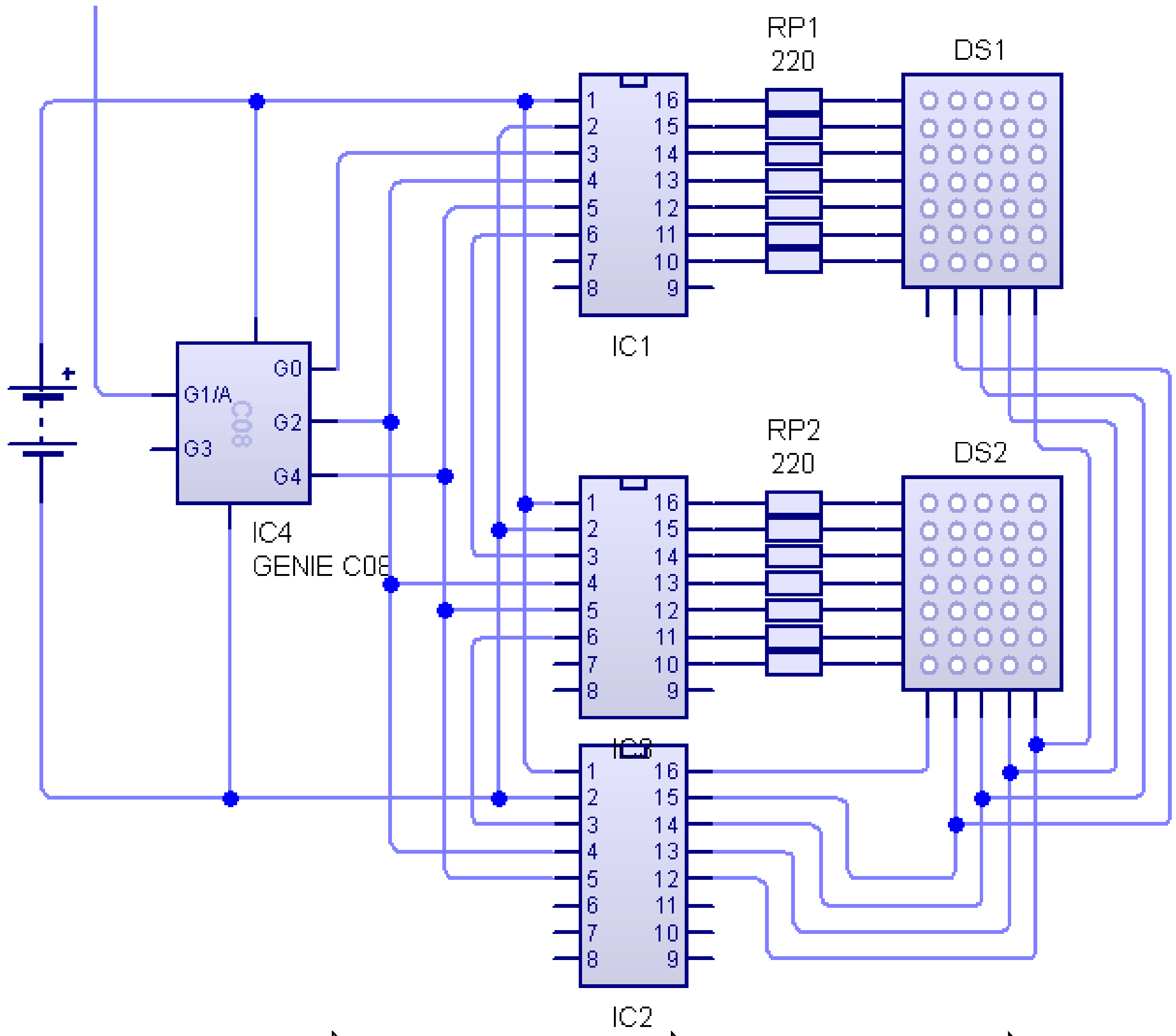
This circuit uses an RFID reader as the input, this allows each racer to have a unique ID and card; this means handicaps can be stored. The RFID reader outputs serial data into an 08m2 Picaxe chip for processing the data. I plan to have user ID and data stored in the built in EEPROM and then it can be looked up and displayed on the LCD display. The buzzer automatically beeps when a card is scanned. This circuit could then output data to another chip which would be responsible for starting and timing the races.

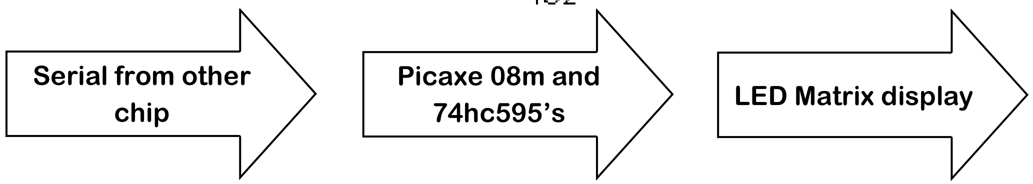
ID-20 RFID Reader → PIC 08m2 → Buzzer and LCD



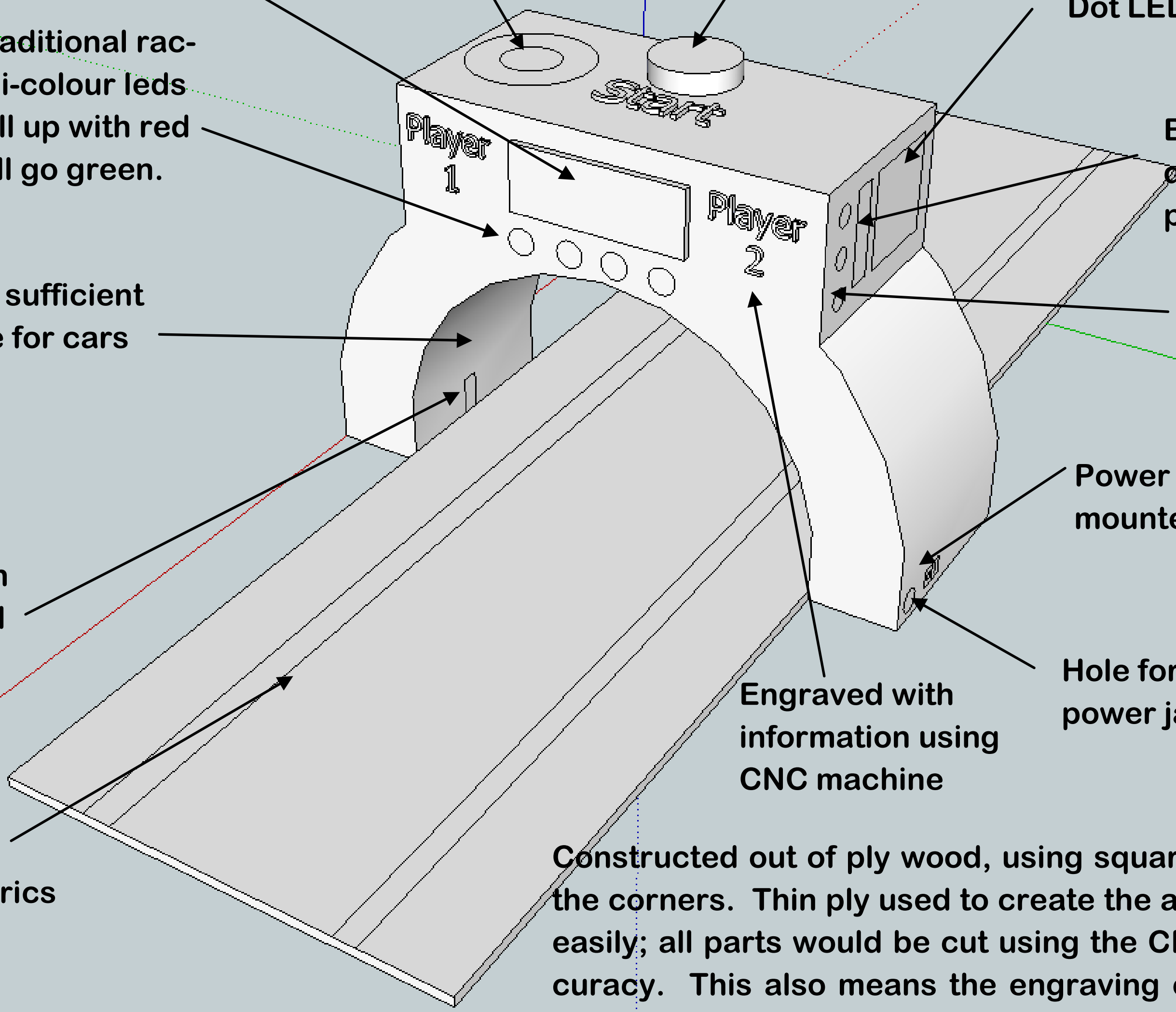
Circuit Ideas

This circuit could be implemented into the scalextrics start gate to allow the countdown to be displayed very clearly for the user. It would work from a serial input coming from another chip, this would be necessary as it would not be able to maintain the required clock speed if it was required to complete other tasks. This serial data would then be processed by the 08m Picaxe chip. Serial data would then be outputted to 3, 74hc595 shift registers. These shift registers give an 8 bit output which would be used to drive the LED matrix's. The 74hc595 shift registers require a clock input as well as serial input and latch. The shift register saves the serial input status on the rising edge of the clock. All 24 different outputs must be inputted and then the latch pin can be pulled high which outputs the values of each pin. The LED matrix's require current limiting resistors to protect the LED's, as there are so many required I will most likely use a resistor array. Also the shift registers are not capable of sinking enough current for 8 LED's to be on full brightness; therefor transistors would be required to sink the current. Like the resistors there will be many transistors so I will probably use a transistor array, for example the ULN2803.



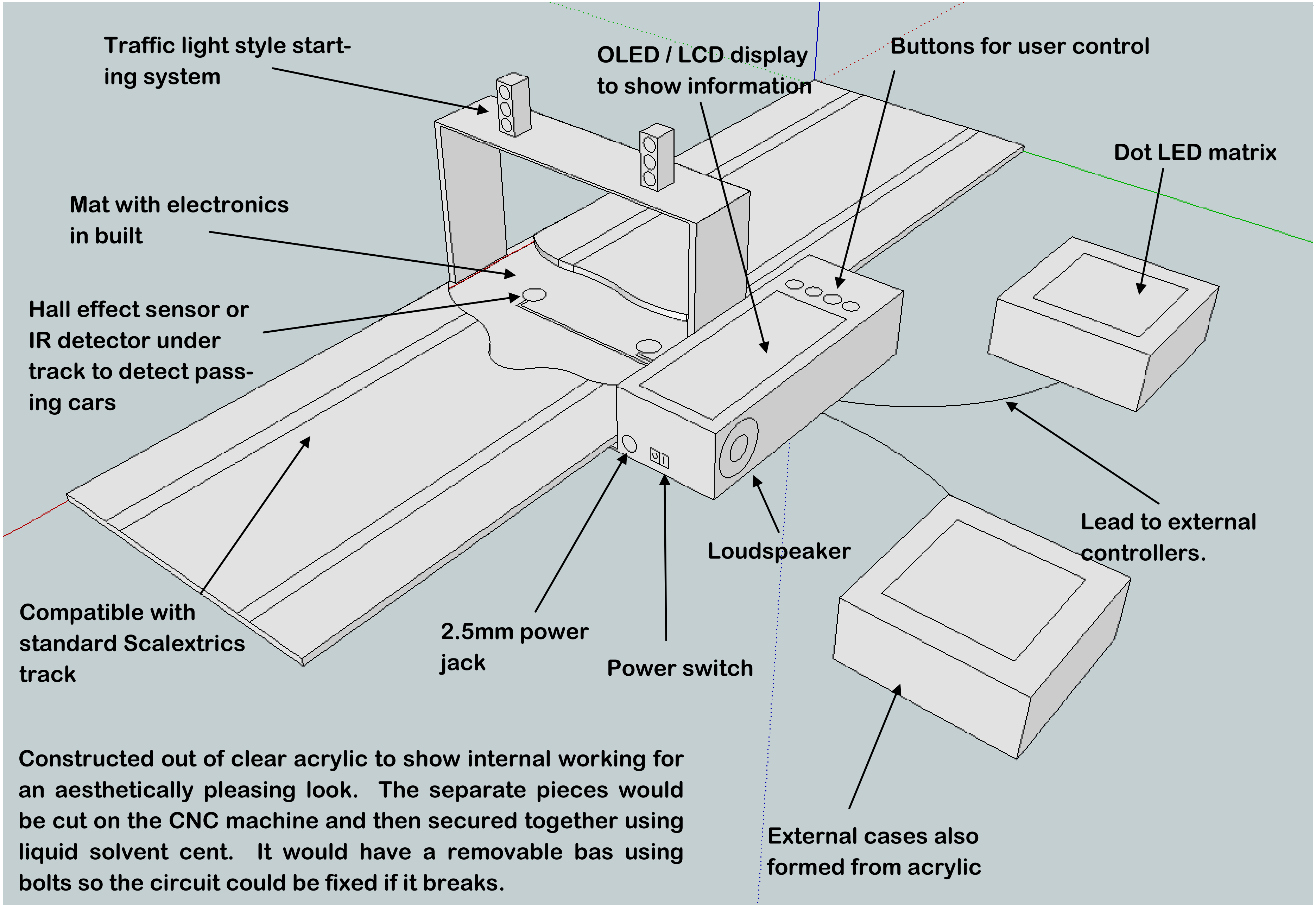


Casing Ideas — Idea 1



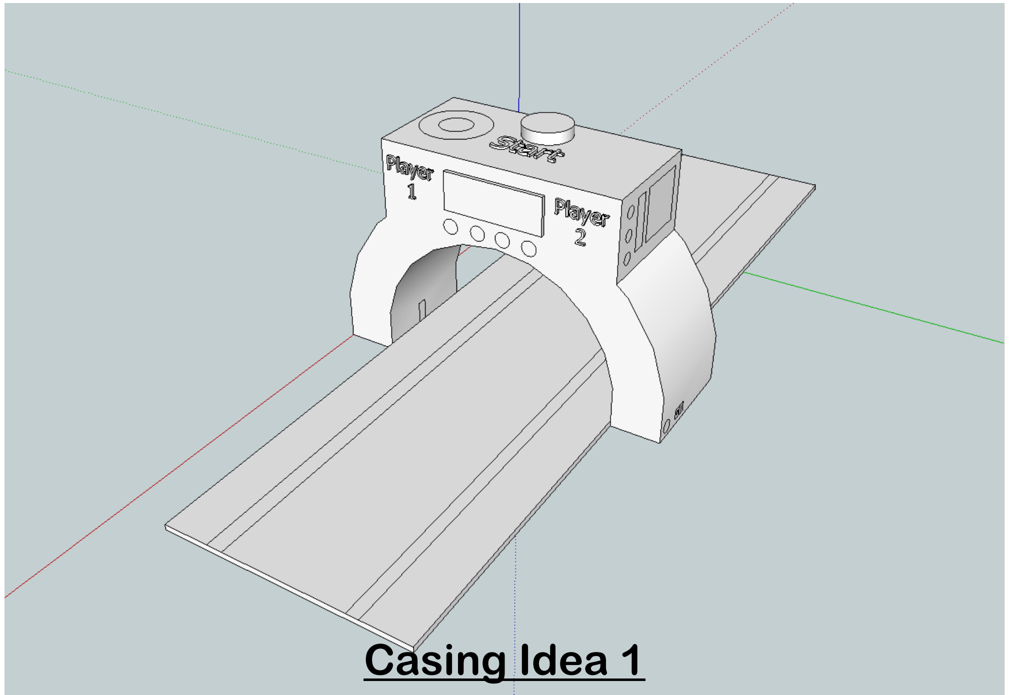
Constructed out of ply wood, using square dowel to support the corners. Thin ply used to create the arches as they bend easily; all parts would be cut using the CNC machine for accuracy. This also means the engraving can be done at the same time, and be placed exactly.

Casing Ideas — Idea 2



Constructed out of clear acrylic to show internal working for an aesthetically pleasing look. The separate pieces would be cut on the CNC machine and then secured together using liquid solvent cent. It would have a removable bas using bolts so the circuit could be fixed if it breaks.

Casing Idea Issues



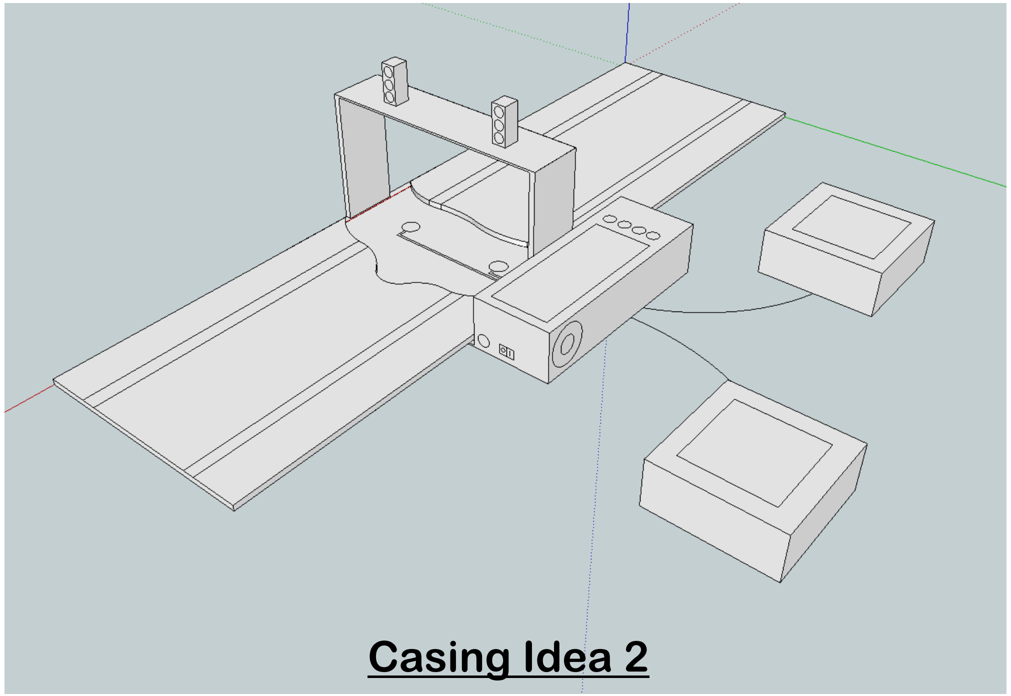
Casing Idea 1

Advantages

- Has an aesthetically pleasing look with the round arch
- Does not require any adaptation to the track
- Features a large start button on top for easy operation
- Bulky so can fit a big circuit board in

Disadvantages

- Difficult shape to manufacture
- Players have to be on opposite sides of the track meaning cables must criss cross over
- Bulky design could be a problem when storing the product



Casing Idea 2

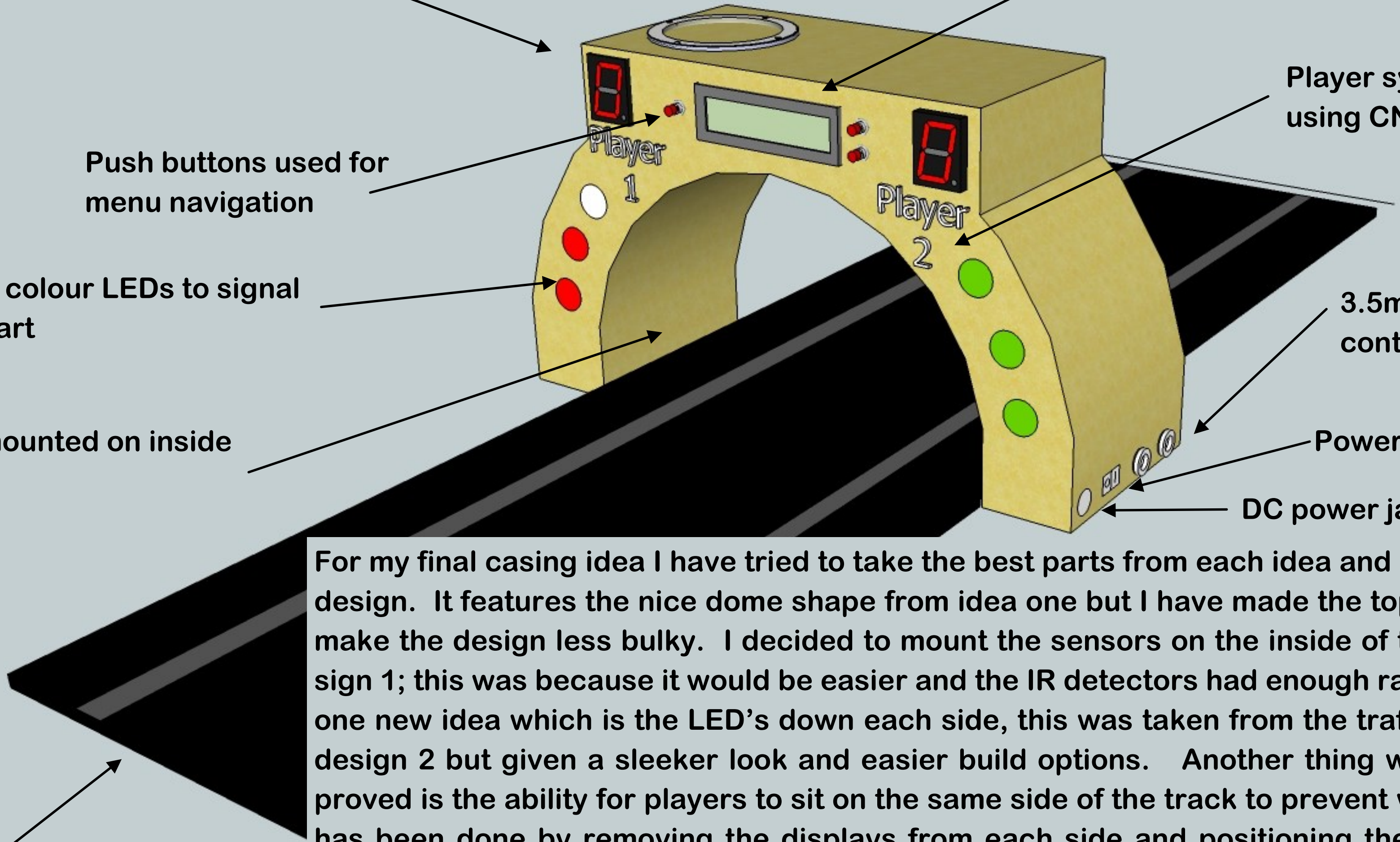
Advantages

- Compact design so easy to store
- Players can be seated anywhere in front of the product
- Individual start lights for each track and therefor works for handicap races

Disadvantages

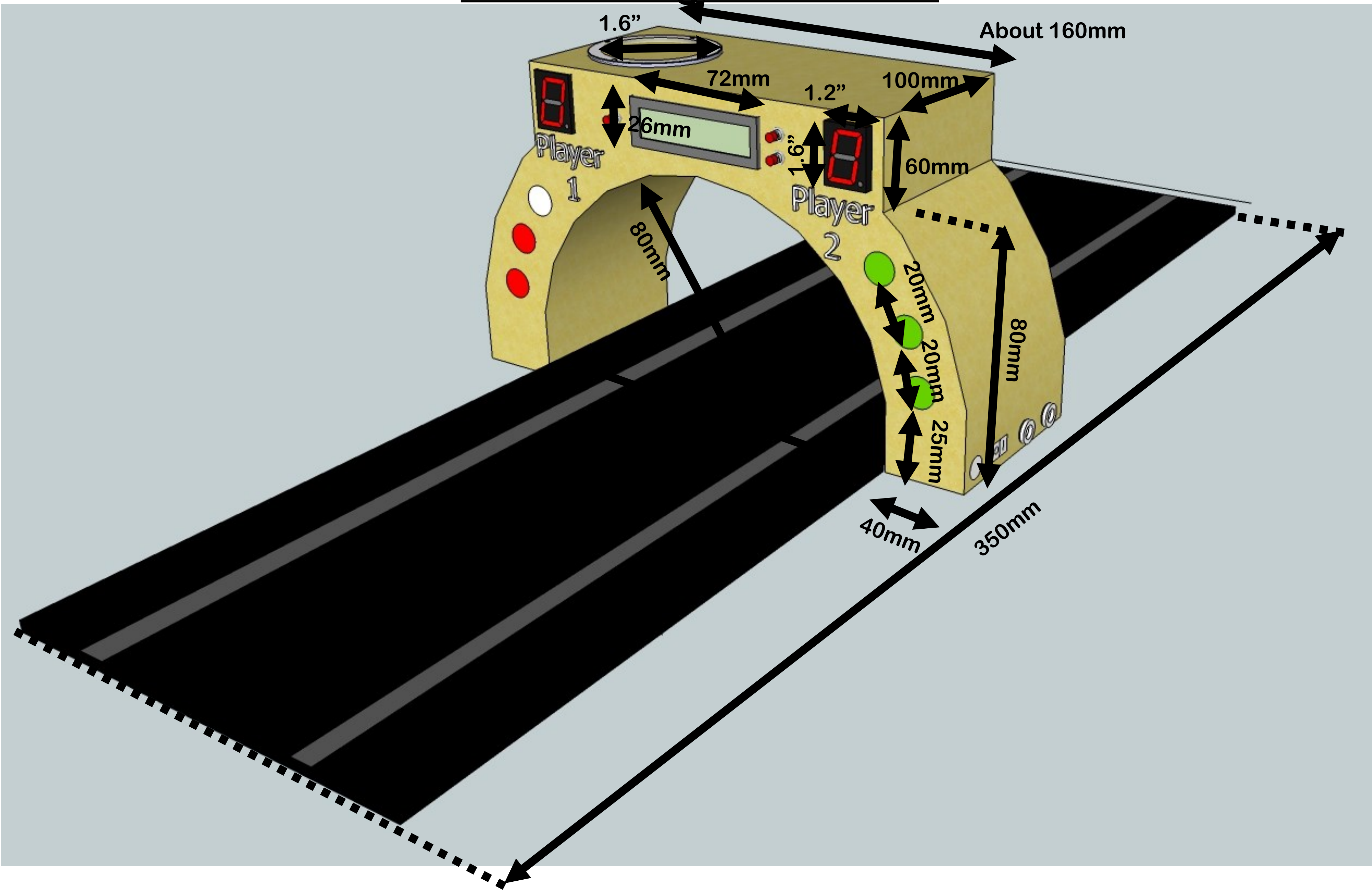
- Sensor must go under the track and therefor could require modification
- Players individual panels mean more wires to get tangled
- Very square design, not so nice to look at

Final Casing Idea

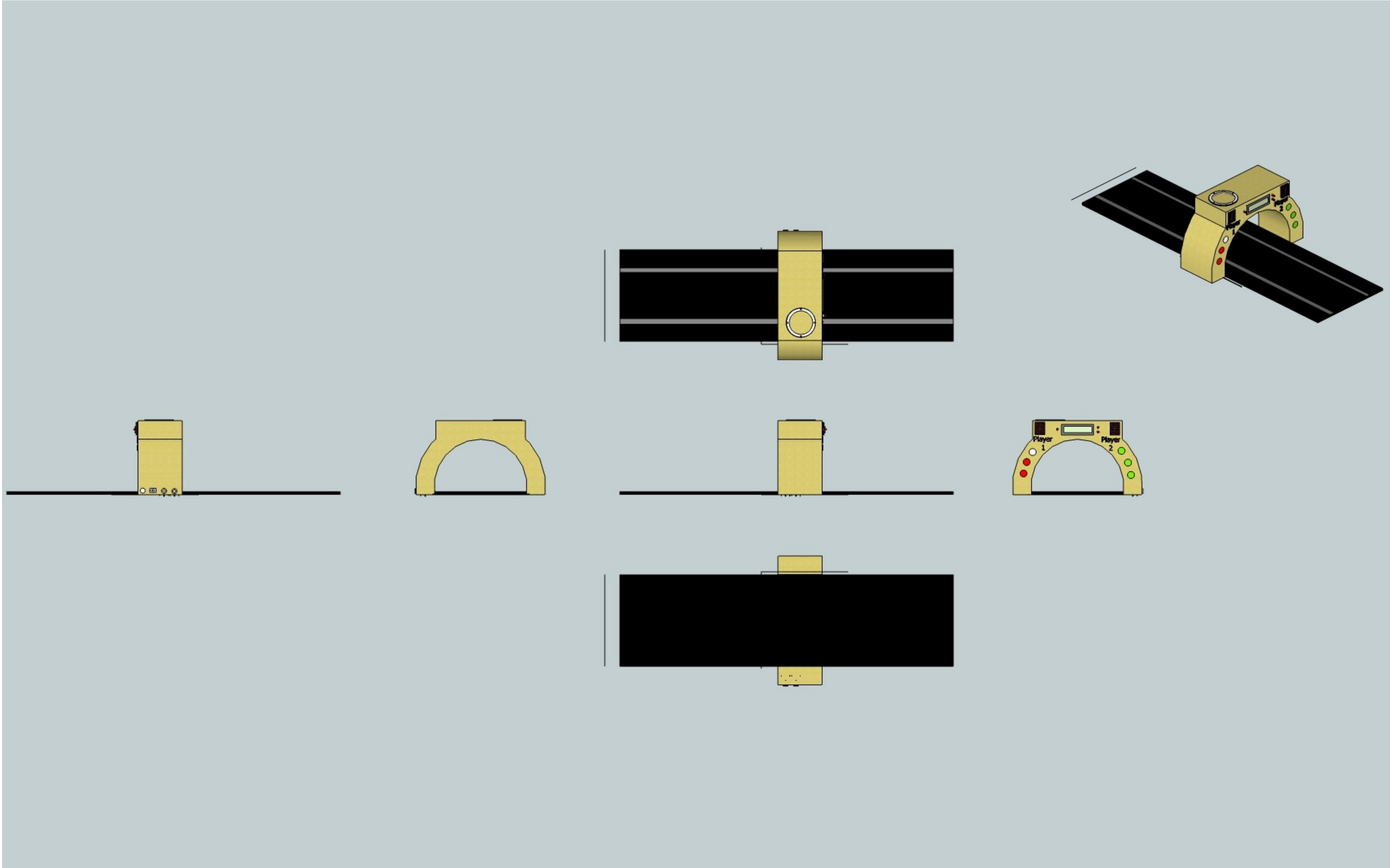


For my final casing idea I have tried to take the best parts from each idea and create an optimum design. It features the nice dome shape from idea one but I have made the top part shallower to make the design less bulky. I decided to mount the sensors on the inside of the arch like in design 1; this was because it would be easier and the IR detectors had enough range. I have added one new idea which is the LED's down each side, this was taken from the traffic light system on design 2 but given a sleeker look and easier build options. Another thing which has been improved is the ability for players to sit on the same side of the track to prevent wires tangling, this has been done by removing the displays from each side and positioning them on the front instead. The back farceur will be removable for maintenance as well as the top piece ; the back will be secured by bolts and the top panel will slot into groves, held in by the back.

Final Casing Dimensions

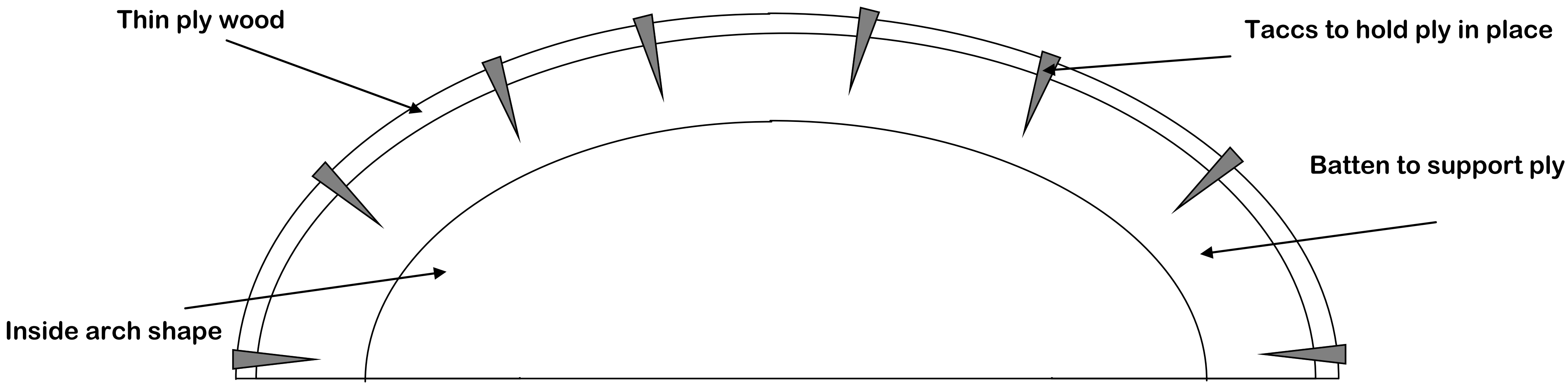


Orthographic View



Creating the Arch Shape

To create the arch shape out of wood I will use thin ply wood which is relatively flexible. As demonstrated in the picture on the right ply wood wilol bend if it is thin enough and I plan to support it in such a position in order to achieve the archway in my design. I will make a batten slightly smaller than the size of the arch, and this will have the flexible ply glued and tacked onto it like demonstrated in the diagram below. The inside of the arch, unlike the outside will probably not require the additional strength from the taccs as the natural spring of the ply will push out and hold itself in position — so for this I will just use glue to support it. I will still use the CNC machine to cut these panels and I will calculate there size using the radius of the arch and the formula for the circumference of a circle — Circumference = $2\pi r$.



Final Manufacture

Production of parts

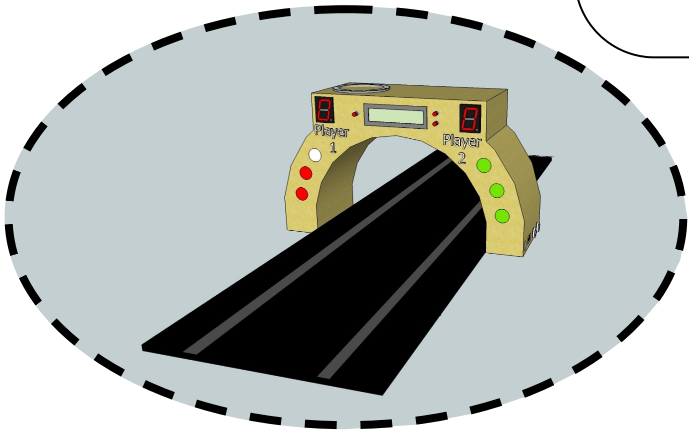
I plan to design the whole product on 2D design and then use the CNC machine to cut out the pieces, I have chosen this method as it will allow me to create precise pieces which will fit together well in order to provide an aesthetically pleasing look. Also I can mill out holes for the LCD, loudspeaker and other components which must be fitted to the front face, very easily. This however would be a difficult process for manufacture on a large scale, so it would probably be made by blow moulding the shell if it was produced on mass.

Installing Circuitry

The case is designed so that the majority of the circuit board will fit in the top section of the casing and only wires will go down the sides to meet the connections. All the external components will be hot glued into place were as the circuit board will be fixed down with sticky circuit board supports.

Materials

The product will be made mainly out of ply wood of varying thicknesses. I have chosen this material as it is easy to work with and will provide a robust design. However the roof will be made from a piece of clear acrylic as well as windows for the starting LEDs. This is because I want the user to be able to see in but to prevent damage it needs a material closing it off; obviously wood cannot be used because it opaque. I also plan to use Vacuum Forming plastic for the mat which will go under the track as it is a very thin material but still has some strength.



Assembly

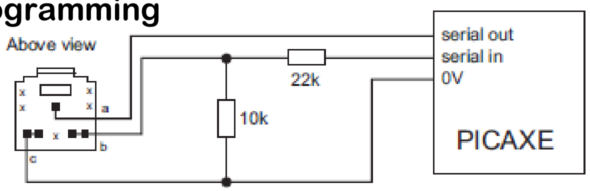
After the parts have been cut and finished up I will assemble them together to check everything is correct, then once sure of this I will use wood glue to secure all the separate pieces together. The arc will be formed by thin bendable ply and may have to be tacked in place onto battens on the inside of the shell, if it is too springy for the glue to support.

Finishing of parts

Due to the inability of the CNC machine to work on 3 dimensions I will have to finish some pieces of by hand, for examples joins of which I require an angle. Also any corners will have to filed out with a mini file as the CNC machine is unable to form a square. Edges must also be sanded down to ensure a high quality of manufacture.

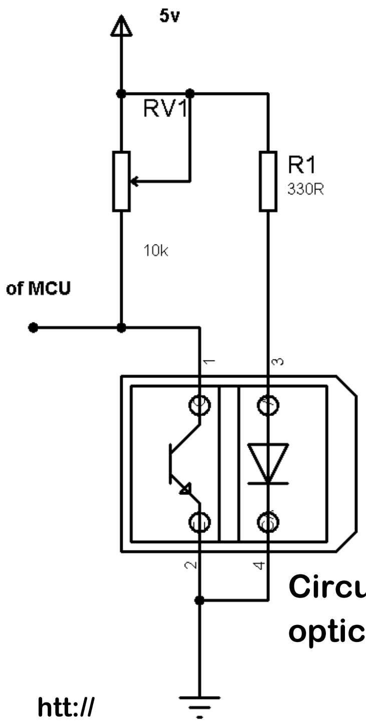
Bread boarding sensors

Circuit diagram for programming



www.picaxe.com

Bread boarding pro-grammer break out board



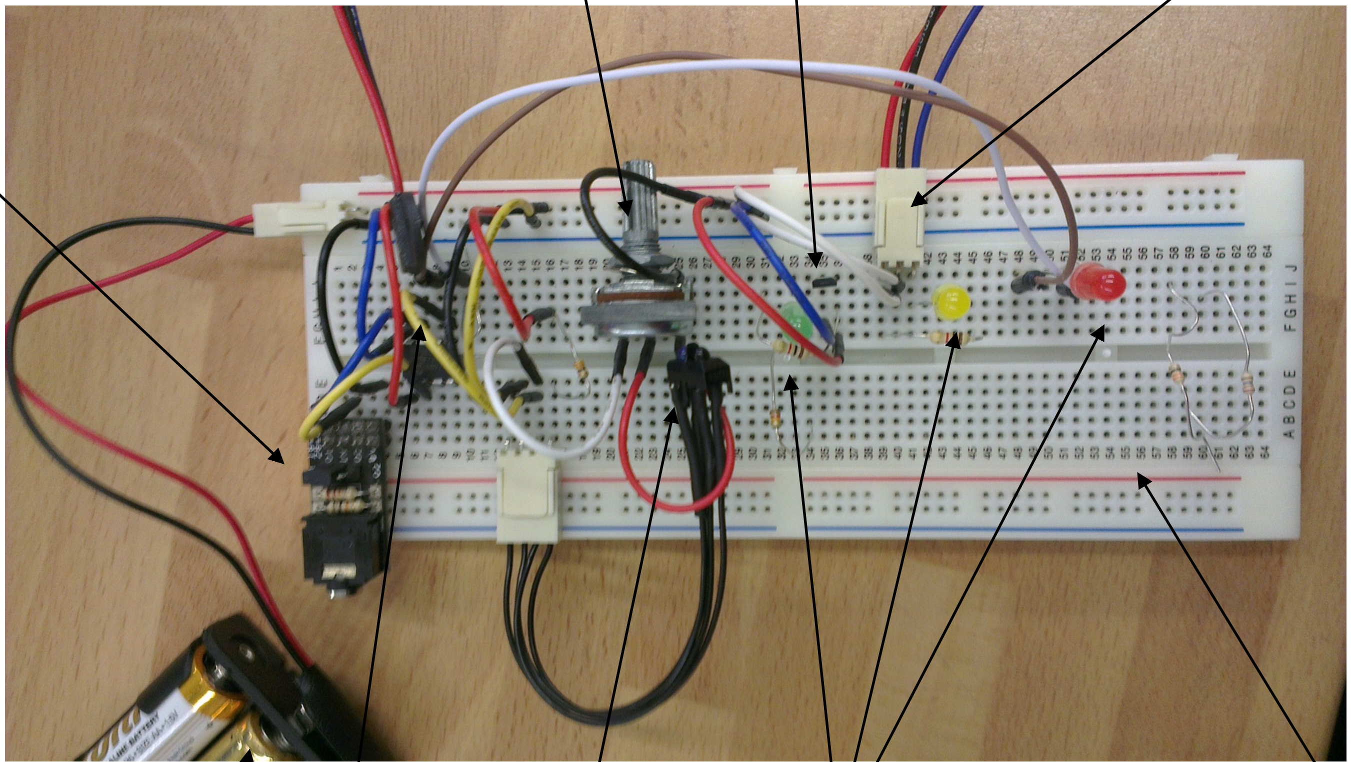
Circuit diagram for optical detector

http://
www.extremeelectronics
.co.in/avrprojects/

10K potentiometer for optical detector calibration

Hall effect sensor H501

Hamilton Hall effect sensor



Pic 08M

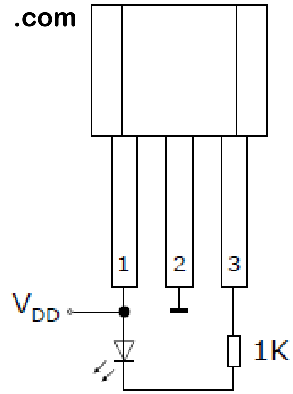
4.5V Power supply

Optical detec-

LEDs to display digital signal

Breadboard

www.produktinfo.conrad
.com



1— Vcc

2— Emitter

3— Collector

Circuit diagram for hall effect sensor

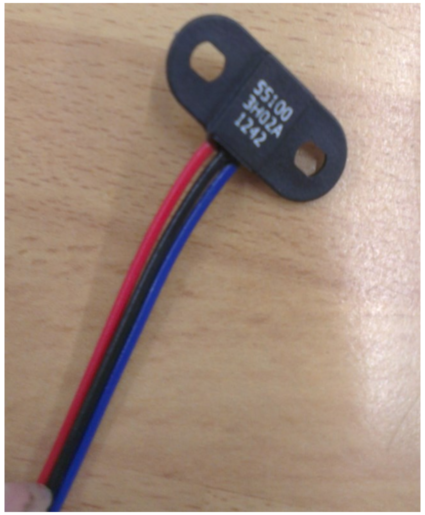
Sensor research

Sensor options:

Hall Effect Sensor Range		
Object	Max range — 55100 Hamilton hall effect (mm)	Max range—Unipolar hall switch H501 (mm)
Neodynium Slice	9	7
HDD Magnet — Large	24	22
Standard Magnet	2	1
Optical Proximity Sensor Range		
Object	Optical proximity sensor TCRT5000 (max / min in mm)	
Scalextric Car	6 — 36	

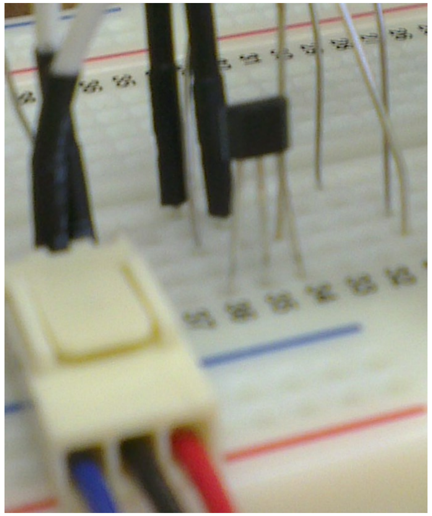
Optical proximity sensor TCRT5000

This optical proximity sensor uses an infrared LED to transmit and detects the level of light reflected through a phototransistor. This is a good solution as it has a further range than the hall effect sensor and does not require modification to any cars. It would however need a more complex casing design to be mounted under the track.



55100 Hamilton hall effect

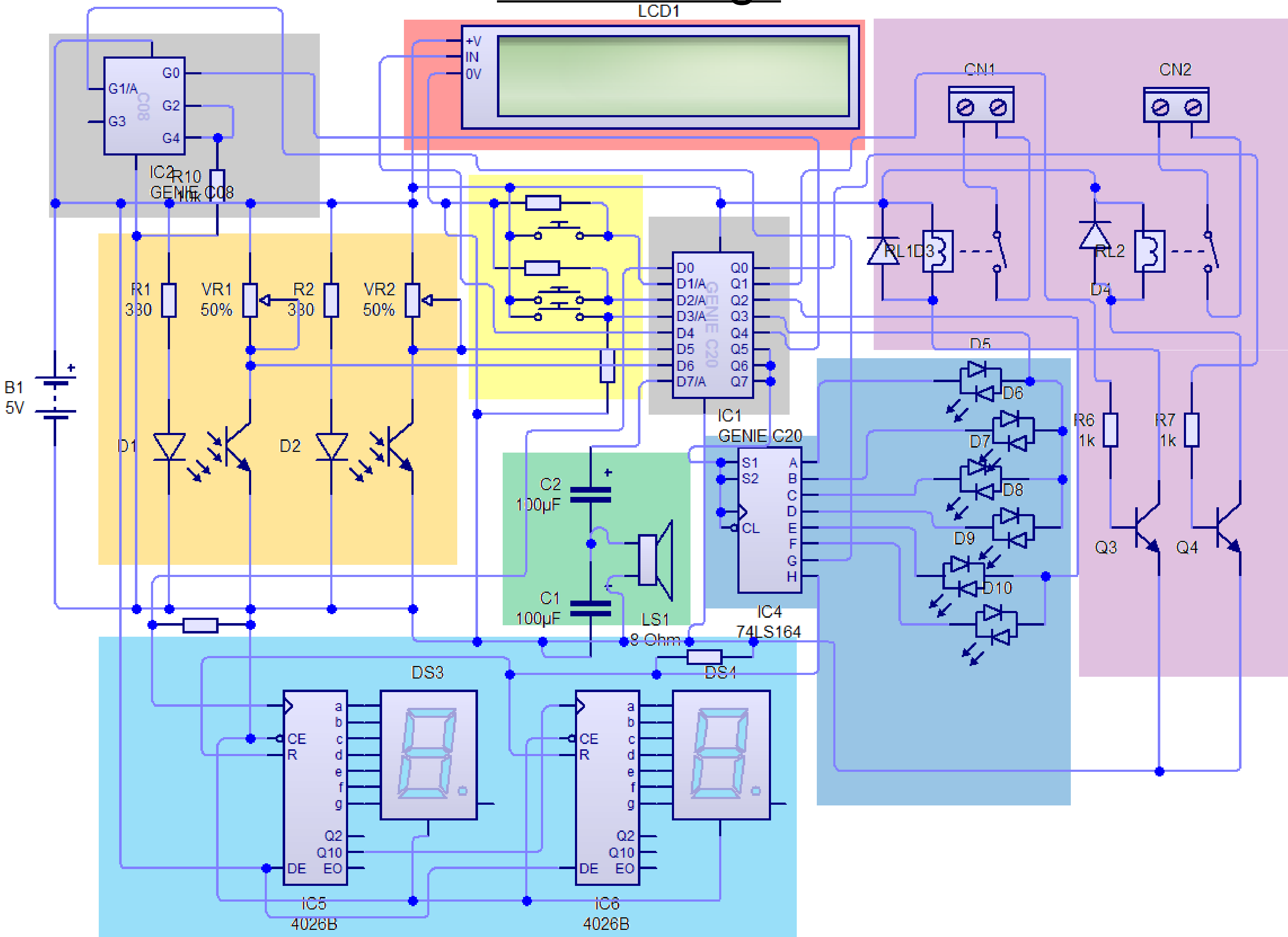
This Hall effect sensor would be a good option for detecting a Scalextrics car passing; it could be mounted under the track and would require no modification to the track. However they are very expensive costing £4 each and would instead require modification to the car in the form of a neodymium rare earth magnet as shown by the results of my investigation.



Unipolar hall switch H501

This is a much cheaper version of the Hamilton hall effect sensor however to my surprise has a very similar performance—only a couple of mm less range in each case. Therefor if I decide to use a hall effect sensor I will use this one over the Hamilton version. It requires very little circuitry to make it work so would be a good option in my project.

Circuit design



Circuit design

LCD Display

This display will be used to communicate options to the user and provide menus which make navigation of the programme easier. It will also be used to display the results of races.

Photo Transistors

These are used to detect when a car passes over them, they will be situated under the track in the gap between the two rails. They each have their own 10K potentiometer for calibration of the sensors and the LEDs have a 330R protection resistor.

Seven Segment Displays

These displays are daisy chained of one clock and they share a reset pin. They will be used to allow the user to select there profile before the race, and then used individually in order to display the countdown sequence for the race; the two allow racers to start at different times.

Start Lights

The start light give the product an aesthetically pleasing feel, making it appear more realistic, they use a 75hc595 shift register to control 8 bi colour LEDs. The start sequence will feature 1 red,2 red, 3 red, 4 red and then all lights will turn green. In total this setup uses 5 pins to control 16 different LEDs.

PTM Switches

These will enable the user to navigate start menus and control the product. They use a 10K pull down resistor to create a potential divider over the PIC chip.

Loudspeaker

The loudspeaker will play sounds when buttons are pressed and also be used to make noise in the start sequence. They are wired in with capacitors to smooth the power supply.

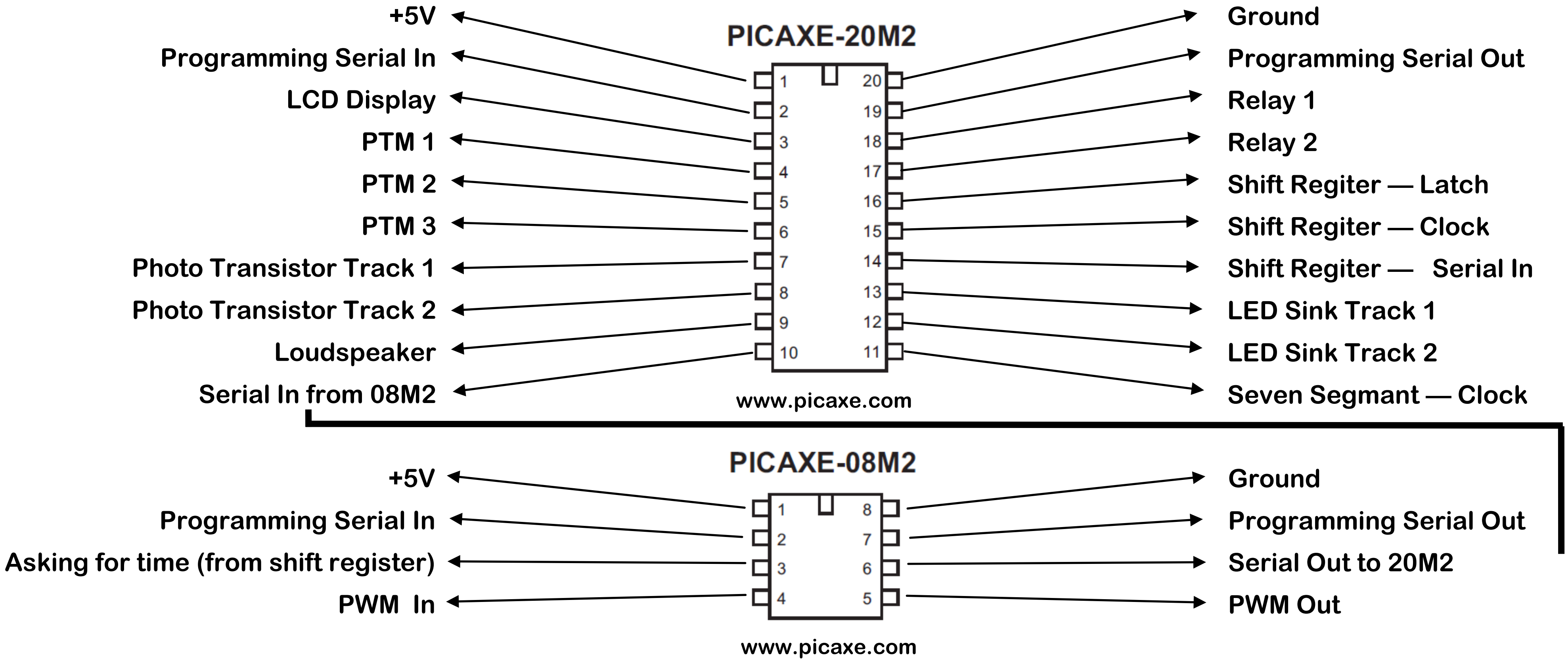
PICAXE Chips

There are two chips in the circuit, a 20m2 for the control of all the inputs and outputs and processing plus there will be an 08m2 for timing purposes; these will communicate through serial. This has to be done because the PICAXE system does not have an effective timing command for milliseconds.

Relay

The relays will be used to cut off the controller from the power base, this will prevent racers starting before the countdown is up. The controller consists of a linear potentiometer and therefor the relay will break the positive power line to the controller. The controllers use 2.35mm jacks so are fairly standard.

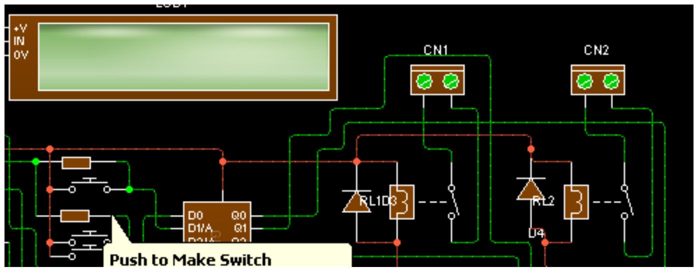
PIC Pinouts

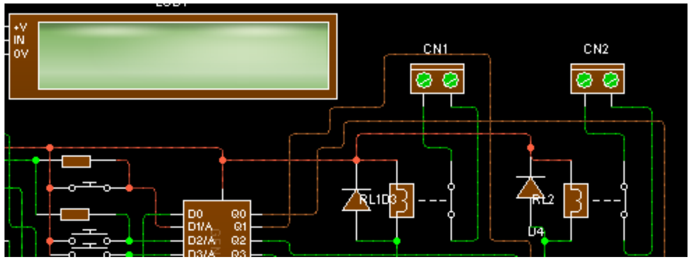


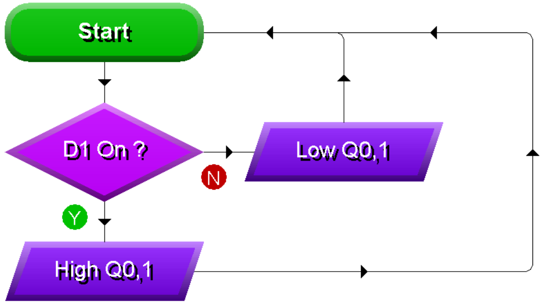
Because Circuit Wizard does not have PICAXE chips the circuit diagram is not 100% correct. Therefor I have had to work out the pin outs for the chip externally, making sure that the function required by each pin is available on the chip. Two chips are required because of the lack of a millisecond count command on the 20M2. The 08M2 acts as a real time clock but more precise than a standard one as it will work in milliseconds. It will do this by having an interrupt set to a digital input feeding from a PWM pin. The PWM pin will be set to 1Khz, so the pin will be pulsed 1000 times a second. On every pulse the 08M2 will increase a word variable and 10ms will write the value to the 20M2 over a serial connection.

Circuit Testing

Relays

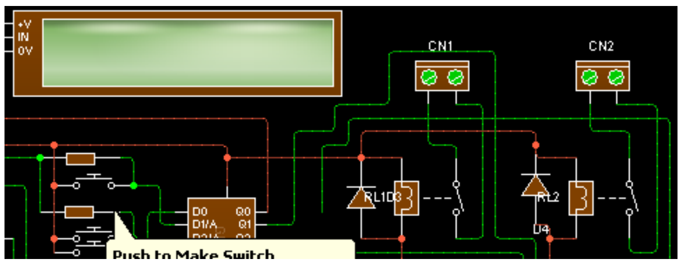


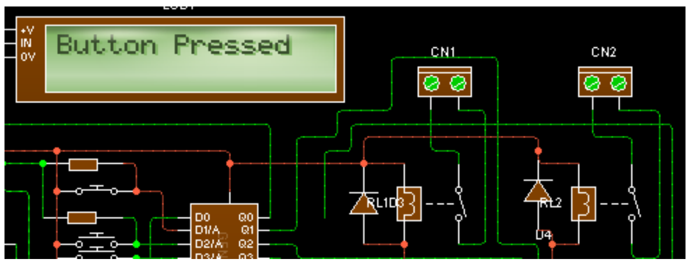


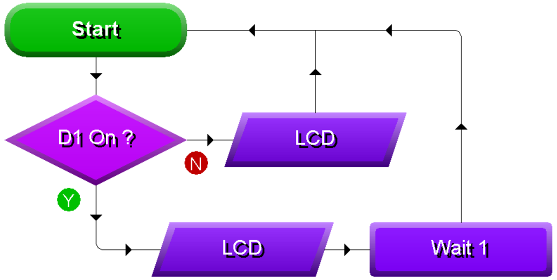


This sample program displays that when the first push to make button is pressed the relays are enabled thus allowing the controllers to be used. Then when the button is released the controllers are no longer active.

LCD Display

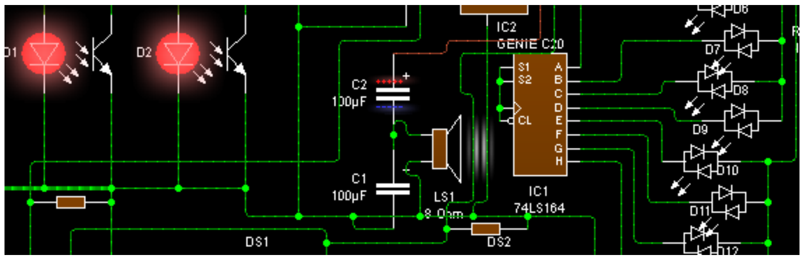


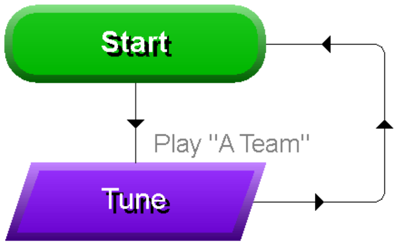




This sample program simply demonstrates the ability to show messages on the LCD display.

Loudspeaker

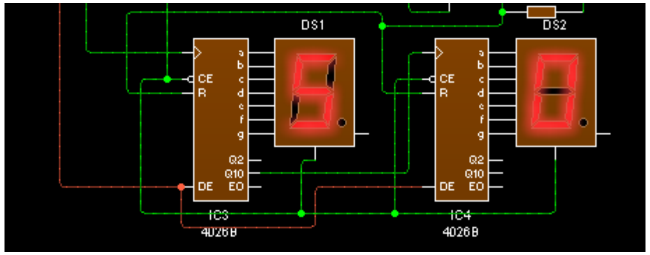


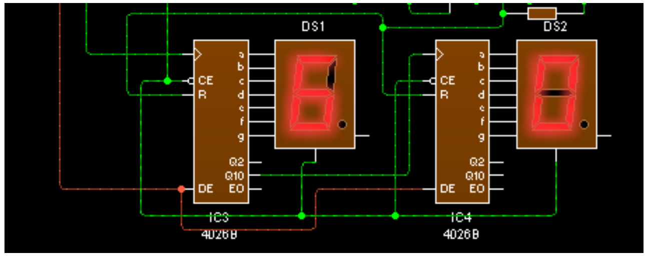


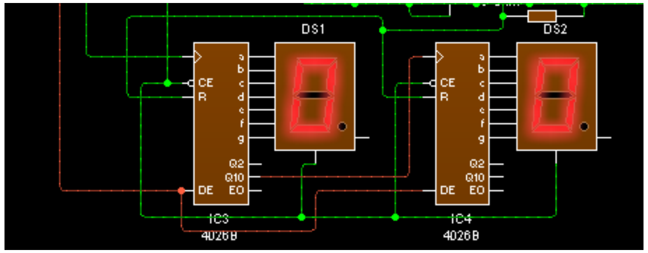
This sample program simply demonstrates the ability to show messages on the LCD display.

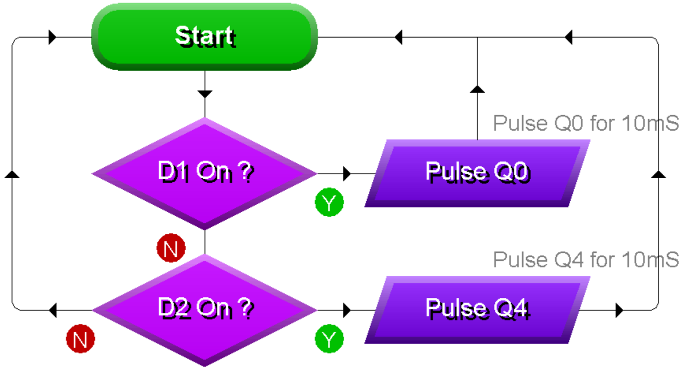
Circuit Testing

Seven Segment Displays



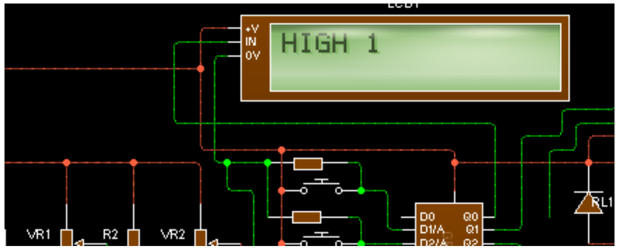


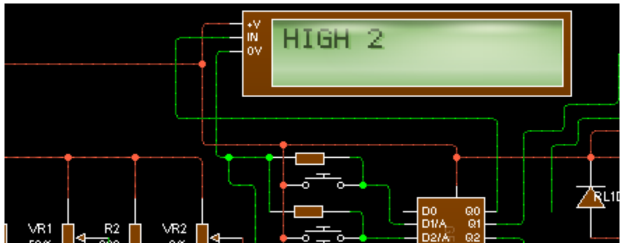


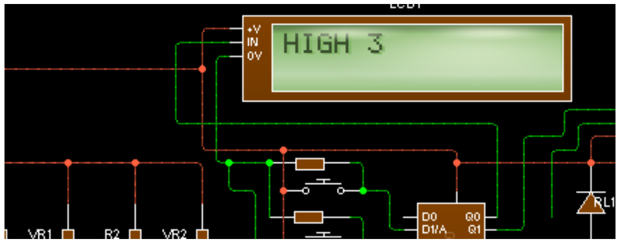


The seven segment displays are joined together however will be used separately, so for example if one must display a 6 and the other a 8; the reset pin will be pulsed to reset both. Then 68 will be clocked in so one display has a 6 and the other has an 8. This basic program demonstrates the reset function and clock function but using PTM switches instead.

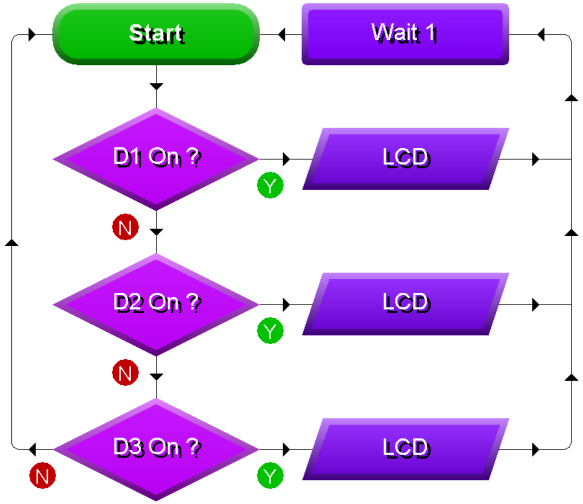
Push Buttons



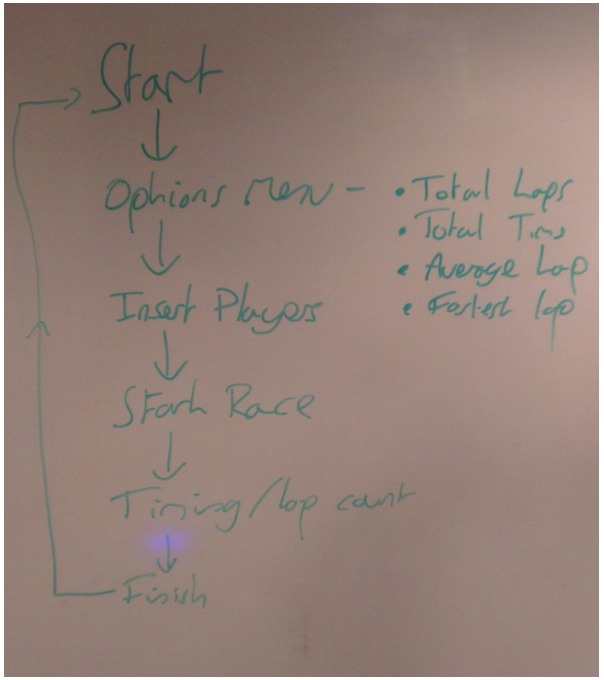




This basic program demonstrates that each push to make switch functions correctly and can be used easily in the main program.

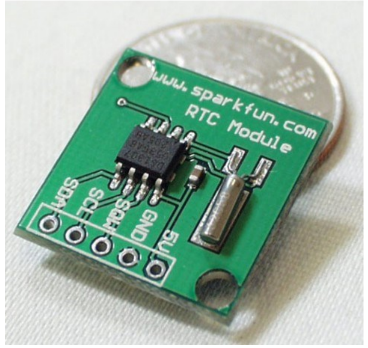


Programming Development



I started of my programming by drawing a flowchart of the functions I needed the device to be able to do. This is in essence the pattern my program follow however it has sub routines added in in order to tidy the place up and save memory. An addition to this was the program for the 08M2 which I had not realised would be necessary by this point until I started looking for an appropriate command and could not find one.

After realising my need for a external clock chip I was frustrated to find that others on the market did not record milliseconds. Consequently I decided to design my own real time clock which is were I had the idea of using PWM to create a steady pulse. Although it would not be as accurate as a real time clock it does boast an advantage in the price and also degree of accuracy in the short term; although over time a real time clock would be necessary. Another advantage of using the 08m is that an external crystal is not required.

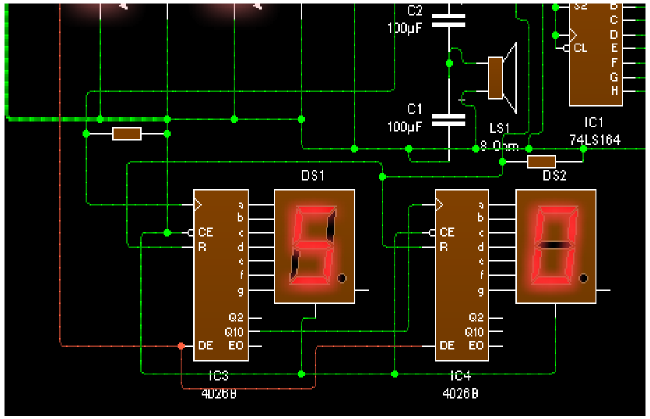


www.electronics-lab.com

```
'=====================================================================
' ***** Shiftout LSB first *****
shiftout_LSBFirst:
            for counter = 1 to bits         ' number of bits
            mask = var_out & 1              ' mask LSB
            low sdata                       ' data low
            if mask = 0 then skipLSB
            high sdata                      ' data high
skipLSB:    pulsout sclk,1                  ' pulse clock for 10us
            var_out = var_out / 2           ' shift variable right for LSB
            next counter
            return
'=====================================================================
```

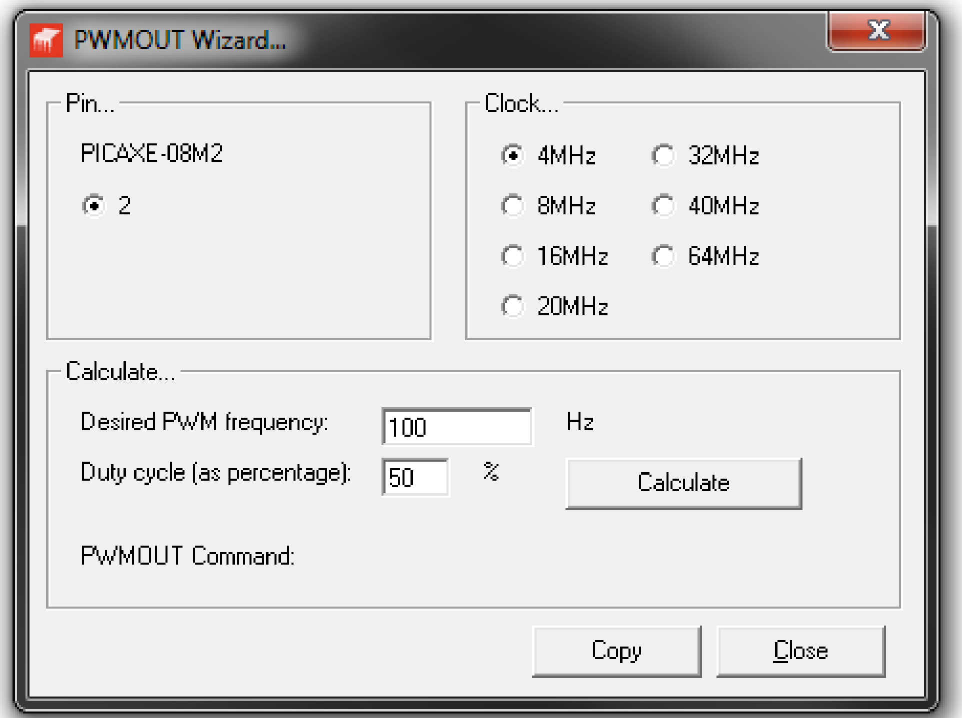
Another issue I encountered was that the command I planned to use for controlling the shift register is only available on X2 microcontrollers. However luckily picaxe provide a sub routine which does the same job; this greatly simplified my programming for the shift register.

The final problem I encountered was lack of pins on the 20M2, however I was hoping I would not have to move up to a 28X2 as I would have way to many pins and the price would be getting to high. Luckily I managed to combine the seven segment displays so they only use one clock, and the reset for both is connected to a spare output on the shift register, allowing for precise control of the displays but without the issue of pin shortage.



08M2 Programming

```
pwmout pwmdiv64, 2, 155, 312'sets pwn for 100hz

main:
do'waits for timer pin to go high
b5=0
loop while pin4=0
b0 = 0'resets timer variables
b1 = 0
b2 = 0
b5=1'tells interrupt to restart itself
setint %00001000,%00001000'setup interrupt
do
pause 1
serout c.1,N4800_4,(165,b0,b1,b2)'transmit serial data
loop while pin4=1'waits for timer pin to go low
goto main

interrupt:
if b0<255 then'adds to timer
inc b0
else if b1<255 then
inc b1
b0=0
else if b2<255 then
inc b2
b0=0
b1=0
endif

if b5=1 then'resets interrupt
setint %00001000,%00001000
endif
return
```



The first line sets up the PWM which will be used as a clock pin; I created this command using the PICAXE PWM wizard in order to obtain a frequency of 100Hz—or 100 times a second. This will provide the 1/100ths of a second accuracy I require. Next the program waits for the 20M2 to start the timer, once started the interrupt is enabled on the pin into which the PWM is pulsing. Every time the interrupt pin goes high, which will be every 1/100th of a seconds the timing variable is added to. This had to be done through 3 bytes as the serial command does not accept word variables and also a word variable would be unable to count for long enough. Using a word variable the timer would reset at (255*255)/100/60 which equals just over ten minutes, instead the counter is able to go for 46 hours before resetting — plenty long enough I feel! While the timer enable pin from the 20M2 is high, the 08M constantly sends out the data from the timer using a serout command. The command first sends a qualifying byte of 165 to tell the 20M2 to start recording, then it follows with the three timer variables in sequence. On the other end the 20M2 pieces the time back together and calculates the seconds and milliseconds from it.

20M2 Programming

```
gametype = 4
players = 1
totaltime=30'setup variables
user2 = 1
user1 = 1
laps = 1

Main:
gosub cursorline1:
serout C.7,N2400,(253,1) 'Ready to Race?
pause 750
gosub cursorline2:
serout C.7,N2400,(253,2) 'Press / to start
if pinc.6=0 then goto main  'Check Button
do loop while pinc.6=1 'Waits for button to go low
'=====================================================================
'=====================================================================
'Setup the Race through series of menus
'=====================================================================
'=====================================================================
gosub cleardisplay


gosub cursorline1:
serout C.7,N2400,(253,3) 'Select Race type
do 'waits for item to be selected on menu
if pinc.5 = 1 then
inc gametype
if gametype <= 3 then 'creates the loop
gametype =8 endif
if gametype >= 9 then
gametype =4 endif
gosub cursorline2:
serout C.7,N2400,(253,gametype) 'changes LCD
endif 'increase game mode

if pinc.4 = 1 then
dec gametype
if gametype <= 3 then 'creates the loop
gametype =8 endif
if gametype >= 9 then
gametype =4 endif
gosub cursorline2:
serout C.7,N2400,(253,gametype) 'changes LCD
endif 'decrease game mode

loop while pinc.6=0
gametype = gametype -3 'makes variable easier to understand
'=====================================================================
```

The initial setup asks the user if they are ready to race and if so to press the select button. Then the user is taken through a series of menus in order to determine the preferences for the race, these include number of players, total laps or total time, race type and allow the users to load there profile to enable the handicapping function. All these menus are displayed on the LCD display making it a user friendly experience.

```
return
'=================================================================
'=================================================================
'Displays 7 seg
'=================================================================
'=================================================================
Display7Segmant:
resetting = 1'sets variable to reset display
gosub Lights'resets display through shift register
resetting =0
gosub Lights
displayValue=displayValue1*10'makes diplayvalue1 the tens on the display by multiplying
displayValue=displayValue+displayValue2'adds on diplayvalue1
generaluse=0'sets up loop variable
do
pulsout b.7,1'pulses the clock
inc generaluse
loop while generaluse<displayValue'loops until all pulses are out

return
```

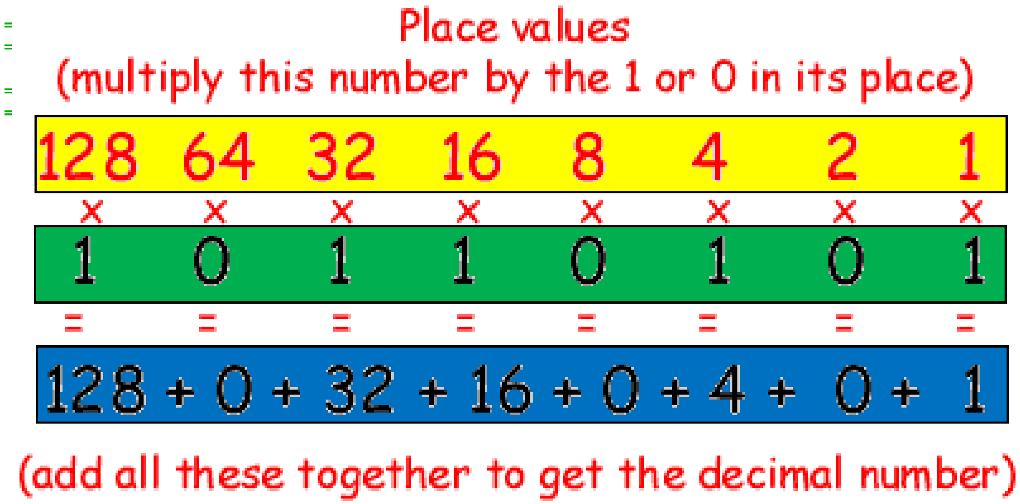
This sub routine is responsible for setting the seven segment displays, it does this by first multiplying the first number by 10 and the adding it to the second number; this is so only one pin has to be used on the chip for a clock instead of having two different clocks. By going to the subroutine lights, the display is reset through the shift register and the calculated value is then clocked into the displays. It counts the pulses in a do loop to ensure it has pulsed enough times.

```
'=================================================================
'=================================================================
'Recovers time
'=================================================================
'=================================================================
RecoverTime:

SERIN [200], c.0,N4800_4,(165),timerr1,timerr2,timerr3'recieve serial in from 08m2
seconds=timerr1*timerr2'calculates seconds and milliseconds from data in
milliseconds =seconds
seconds=seconds/100
generaluse=seconds*100
milliseconds=milliseconds-generaluse
generaluse=timerr3/100
generaluse=timerr3*100
generaluse=timerr3-generaluse
milliseconds=milliseconds+generaluse
generaluse=timerr3/100
seconds=seconds+generaluse

return
```

The Recover time subroutine waits for serial data to be received from the 08M2, the command waits for a qualifying byte and if this is correct reads in the time variables to three more. If no data is received the command times out after 200ms. Once the data has been received a mass of calculations is done in order to produce to overall variables, one seconds and one milliseconds.

20M2 Programming

```
return
'====================================================================
'====================================================================
'Works the shift Register
'====================================================================
'====================================================================
Lights:
'Shift Regiter Latch          B.2
'Shift Regiter Clock          B.3
'Shift Regiter Serial In      B.4
'LED Sink Track 1             B.5
'LED Sink Track 2             B.6   sink for green, source for red
var_out = 0
gosub shiftout_LSBFirst'shifts out 0s to clear shift register
pulsout b.2,10
low b.5
low b.6
'calculate the shift register data
generaluse = resetting'reset pin?

if timerr=1 then'timer active?
generaluse =generaluse+128
endif

if countdown1=1 then '1 seconds     track1
generaluse=generaluse+14
else if countdown1=2 then '2 seconds
generaluse=generaluse+12
else if countdown1=3 then '3 seconds
generaluse=generaluse+8
else if countdown1=0 then'0 seconds
high b.5
endif

if countdown2=1 then '1 seconds     track 2
generaluse=generaluse+112
else if countdown2=2 then '2 seconds
generaluse=generaluse+96
else if countdown2=3 then '3 seconds
generaluse=generaluse+64
else if countdown2=0 then'0 seconds
high b.6
endif

var_out = generaluse
gosub shiftout_LSBFirst'shifts out data to shift register
pulsout b.2,10
return
```



spaceplace.nasa.gov

I have used binary in order to calculate a value which refers to each output from the shift register. This saves a lot of hassle in the programming and also means that the shiftout command can be used, which makes the program tidier and more efficient. The binary is calculated using the same method as above. The values in the green box are dependant upon the outputs required status. Then the value in the green box is multiplied by that in the yellow box and finally all are added together in the blue box. This gives a unique value which, in this case gives the binary value of %10110101 — this is the pattern of the shift register.

First of all the subroutine clears all the data from the shift register and pulses pin b.2 to latch it in this position. The following lines are all responsible for calculations the overall binary value talked above. It takes into account the seven segment reset, countdown of the race for each player and also the pin which controls the 08M2. Once this has been done it goes to the shiftout sub routine to send the data to the shift register. Finnaly b.2 is pulsed again to latch the output statuses.

20M2 Programming

```
do
gosub RecoverTime'get time
if startTime1 <= seconds then
countdown1=startTime1+3-seconds'check time against players start time
endif
if startTime2 <= seconds then
countdown2=startTime2+3-seconds
endif
if countdown1 =0 and countdown2=0 then'manages countdown process
high b.0 ' enable track 1
high b.1 ' enable track 2
gosub lights
tune C.1, 8, %00000000,($C3)
countdown1 = 10
countdown2 = 10
else
if countdown1 =0 then
high b.0 ' enable track 1
gosub lights
tune C.1, 8, %00000000,($C3)
countdown1 = 10
endif
 if countdown2 =0 then
high b.1 ' enable track 2
gosub lights
tune C.1, 8, %00000000,($C3)
countdown2 = 10
endif
endif
if countdown1 =3 or countdown2 =3 or countdown1 =2 or countdown2 =2 or countdown1 =1 or countdown2 =1 then'beeps countdown
tune C.1, 8, %00000000,($00)
gosub lights
endif
readadc C.3,  generaluse'checks for car passing on track 1
if generaluse>245 and countdown1=10then
inc startTime1
countdown1=11
else
countdown1=10
endif

readadc C.2,  generaluse'checks for car passing on track 2
if generaluse>245 and countdown2=10then
inc startTime2
countdown2=11
else
countdown2=10
endif

gosub RecoverTime 'gets time

loop while seconds<totaltime'waits for total time to be reached
timerr=0'stops and resets timer
gosub lights
'Chariots of Fire 18 seconds
tune C.1, 15, %00000000,($71,$76,$78,$7A,$38,$35,$7C,$71,$76,$78,$7A,$F8,$7C,$71,$76,$78,$7A,$38,$35,$7C,$75,$76,$75,$71,$F1)
low b.0'disable controllers
low b.1
startTime1=-10'remove 10 from laps
startTime2=-10

if starttime1>starttime2 then'set variable to winner
generaluse=1
else
generaluse=2
endif
```

This part of the programme is the part which controls the main part of the race, the starting, finishing and lap counting. It starts by constantly checking the time until the racers start time is reached; it the initialises the start sequence involving all the lights and sounds. Once the countdown reaches zero, then the controllers are enabled and the race begins.

Now the program starts recording the laps the user has completed; for this particular game mode the race is based on most laps in a certain time period. Every time the cars pass the gate the lap count is increased and time checked to ensure the finish time has not passed.

Once the time is up the loudspeaker is used to play chariots of fire while the racers come to a stop; after the 18 seconds theme tune is finished the tracks are disabled and the result is displayed on the LCD display. Finally the user profiles handicap number is adjusted and then the system returns to the home screen. The handicap works by incrementing the handicap number for a win and decreasing it for a loss. The start time for each player equals the handicap number squared; this provides a parabolic start time making the time difference increase more the better the player.

Design Issues

My design should not have any social or ethical issues in it however there are a few issues relating to environmental impact and sustainability of my product. The wood should be sourced sustainably so the environment is not adversely effected by it. I could check this by referencing the FSC's supplier database, there is a link too it on this website http://www.forestry.gov.uk/website/forestry.nsf/byunique/infd-7m8fz7. Materials such as acrylic are all bad for the environment as they are produced from crude oil, however only a small amount of this will be used so it should be fine. The acrylic is better as there is less of a sustainability issue, although the problem is still there as oil is a finite resource. On the whole I believe the product has very few issues of this nature.

Final Specification

***Green text has been either modified or added** and black text is the same.*

My product will...

- Be designed for use by 11 — 16 year olds
- Be compatible with standard Scalextrics track and cars
- Cost between £30 — £40 for the consumer to buy
- Be aesthetically pleasing for the target market
- Be safe for use in school
- Be able to store 50 user profiles and appropriate handicaps
- Use light and sound to signal the countdown sequence to both the cars and players to simulate a real racing experience
- Feature an easily navigable menu to change options for the race
- Use a form of display to show relevant information
- Be powered from the mains
- Record laps up to and including 99
- Record lap times with a maximum of 6500 seconds
- Have a timer with accuracy of 0.01 seconds
- Be able to record times and laps for two lanes at once
- Use a durable material and be a compact size
- Prevent false starts
- Different race types such as fastest laps, most laps in a certain time, fastest lap and average lap

Plan of manufacture for Casing

Time Period	To Do	Tools Required	Quality Control	Health and Safety
2 hours	Design all the panels which make up the final case.	2D design	Ensure all sizes are correct	None
1 hour	Cut the panels in the correct widths of wood on the CNC machine.	CNC Machine	Sand down all the edges, use file too	Monitor carefully whilst machine is in operation
30 mins	Ensure parts fir together and use a file to adjust any internal angles so it does.	Mini Files	Take care so no large gaps arise	Take care when using file
20 mins	Fit in components which will be not easily accessible when assembled and decide on cable routing.	Hot glue gun	Ensure a flush finish on all components	Be careful of heat from glue gun
1 hour	Fit all parts together using glue and taccs to secure it.	Hammer/ Glue	Make sure everything is in the correct position before leaving	Use the hammer with caution!
30 mins	Fit all the final components in place and wire everything up to the circuit board.	Hot Glue gun	Fit all components flush with surface	Be careful of heat from glue gun
30 mins	Use sand paper to finish up the wood and remove protective coating from acrylic.	Sand Paper	Ensure the product is aesthetically appealing	None
Remaining Time	Test out the casing and modify if possible			

Plan of manufacture for Circuit

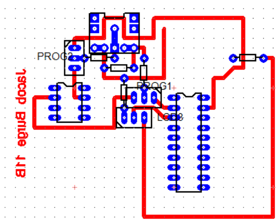
Time Period	To Do	Tools Required	Quality Control	Health and Safety
5 hours	Design circuit board and compile parts list of required components. Print out onto acetate.	Circuit Wizard	Run simulations to check circuit functions correctly	None
	Cut photo resist board down to size, place acetate on photosensitive side of board.	Scissors	Ensure photo sensitive board is the correct way up	Careful when using guillotine to cut PCB
	Place photo resist board in UV light box for 2.5 minutes.	UV Light Box	None	Only switch on light box with lid shut
1 hour	Put the photo resist board in developer solution only briefly.	Developer tank	Only dip for 10 seconds max	Careful of acid in developer tank
	Put the photo resist board in the etch tank for 15—45 minutes depending on the acids strength.	Etchant tank	Check all the copper is eroded before removing	Careful of acid in etchant tank
	Using the PCB rubber clean up the board ready for soldering.	PCB Rubber	Check all the tracks have been cleaned	None
15 mins	Drill holes in PCB.	Mini Pillar Drill	Check all holes are drilled	Do not put finger under drill
2 hours	Populate PCB and crimp wires from external components as well as heat shrinking any connections.	Soldering Iron, Crimp kit	Ensure good quality solder joints throughout and no short circuits	Soldering iron is hot, do not touch
Remaining Time	Test out all aspects of the circuit and modify if possible			

Overall Lesson Plan

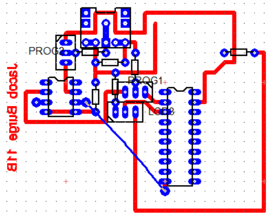
Project Plan	Oct-13					Nov-13										Dec-13							Jan-14										Feb-14			
	15	16	18	21	25	5	6	8	11	15	19	20	22	25	29	3	4	6	9	13	17	18	6	10	14	15	17	20	24	28	29	31	3	7	11	12
PCB design	■	■	■	■																																
Parts list					■																															
PCB manufacture						■	■																													
Drill holes in PCB							■																													
Populate PCB							■	■	■																											
Wire external components									■	■																										
Program and test circuit										■	■	■	■	■																						
Design caseing on 2D design														■	■	■	■																			
Cut panels on CNC machine																	■	■	■																	
Ensure parts fit together																			■	■																
Finish up the edges with sand paper																				■	■	■														
Secure components into panels																						■	■													
Glue together casing																							■	■												
Fit all components in casing																									■	■										
Tidy up the casing																										■	■									
Testing and modifications																											■	■	■	■	■	■	■	■	■	■

Project Work	Oct-13					Nov-13										Dec-13							Jan-14										Feb-14			
	15	16	18	21	25	5	6	8	11	15	19	20	22	25	29	3	4	6	9	13	17	18	6	10	14	15	17	20	24	28	29	31	3	7	11	12
PCB design	■	■	■	■	■	■	■																													
Parts list					■	■	■																													
PCB manufacture											■	■																								
Drill holes in PCB												■																								
Populate PCB												■	■	■	■																					
Wire external components																■		■	■	■																
Program and test circuit																					■	■	■								■	■	■	■	■	■
Design caseing on 2D design								■	■	■	■													■	■											
Cut panels on CNC machine																										■										
Ensure parts fit together																										■	■									
Finish up the edges with sand paper																											■									
Secure components into panels																												■	■	■	■					
Glue together casing																												■	■	■	■					
Fit all components in casing																												■	■	■	■					
Tidy up the casing																															■					
Testing and modifications																																	■	■	■	■

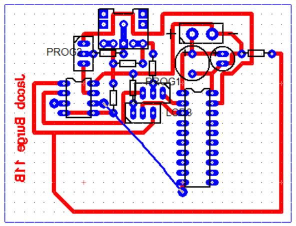
Circuit Board Design Diary



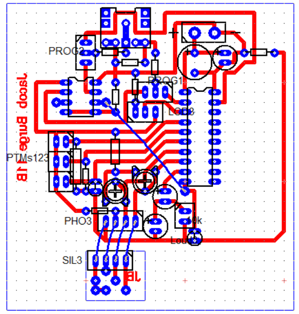
Put the 20M2 and 08M2 chips in place and the shared programming port.



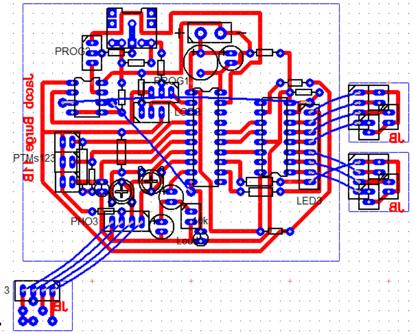
Wired in the PWM link for the 08M2 and the communication pins for the 20M2.



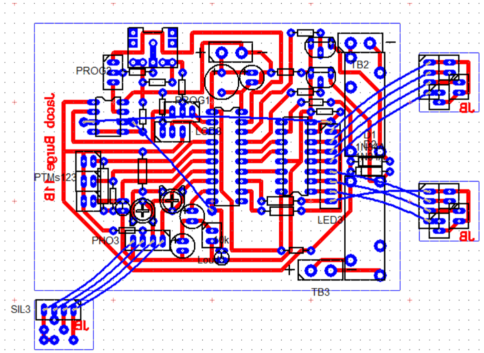
Added in the power terminal blocks, smoothing capacitors and connected to chips.



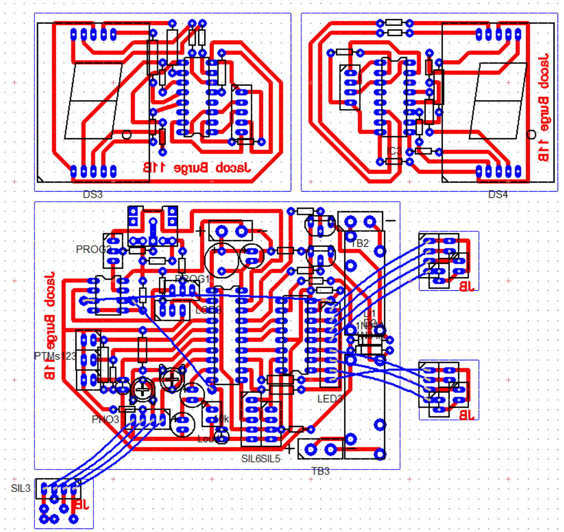
Put in PTM switches, loudspeaker and SIL for off board photo transistors and LCD display.



Added shift register and appropriate SIL pins as well as LED hook up board.

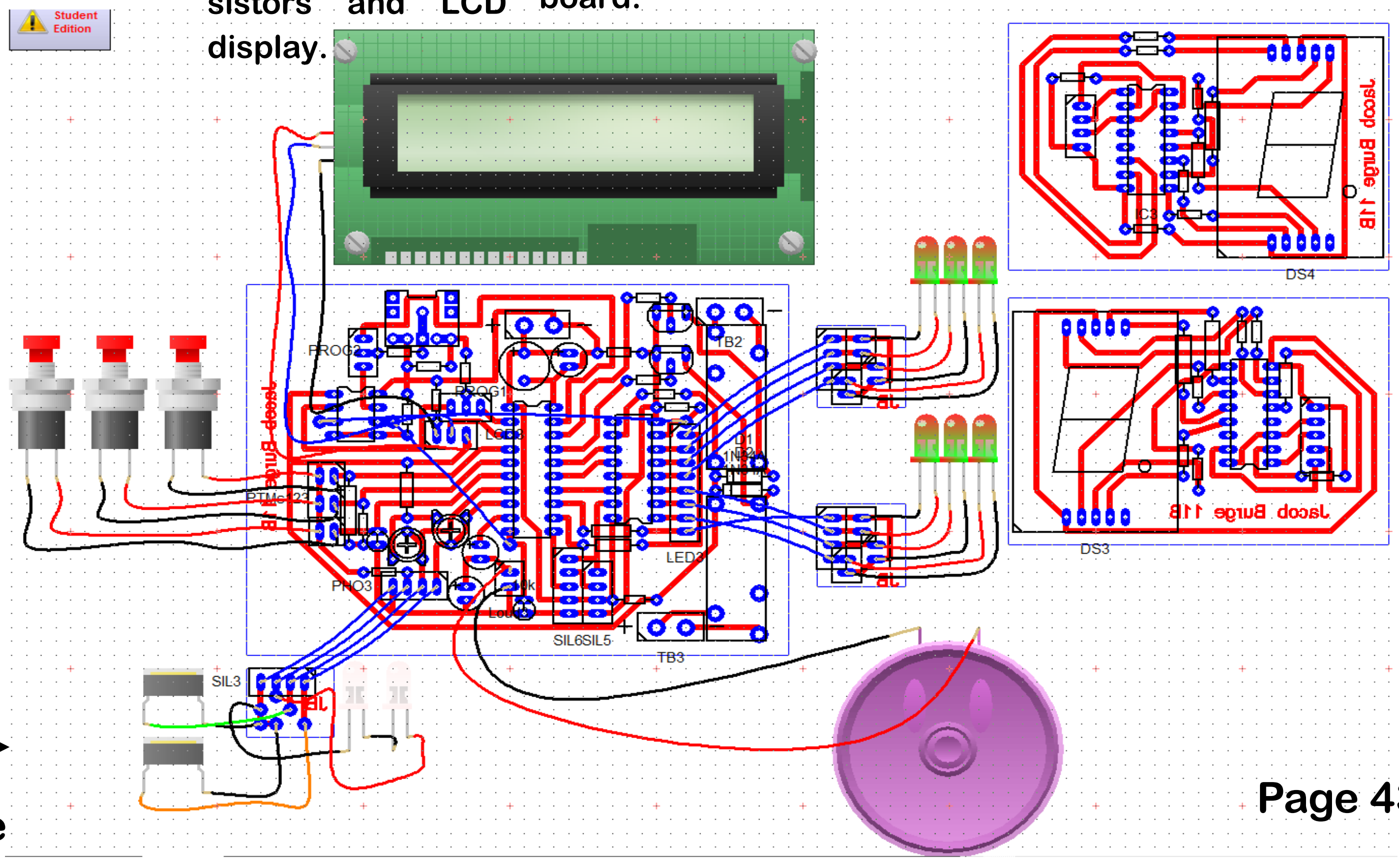


Put relays in place with terminal blocks and appropriate transistors.

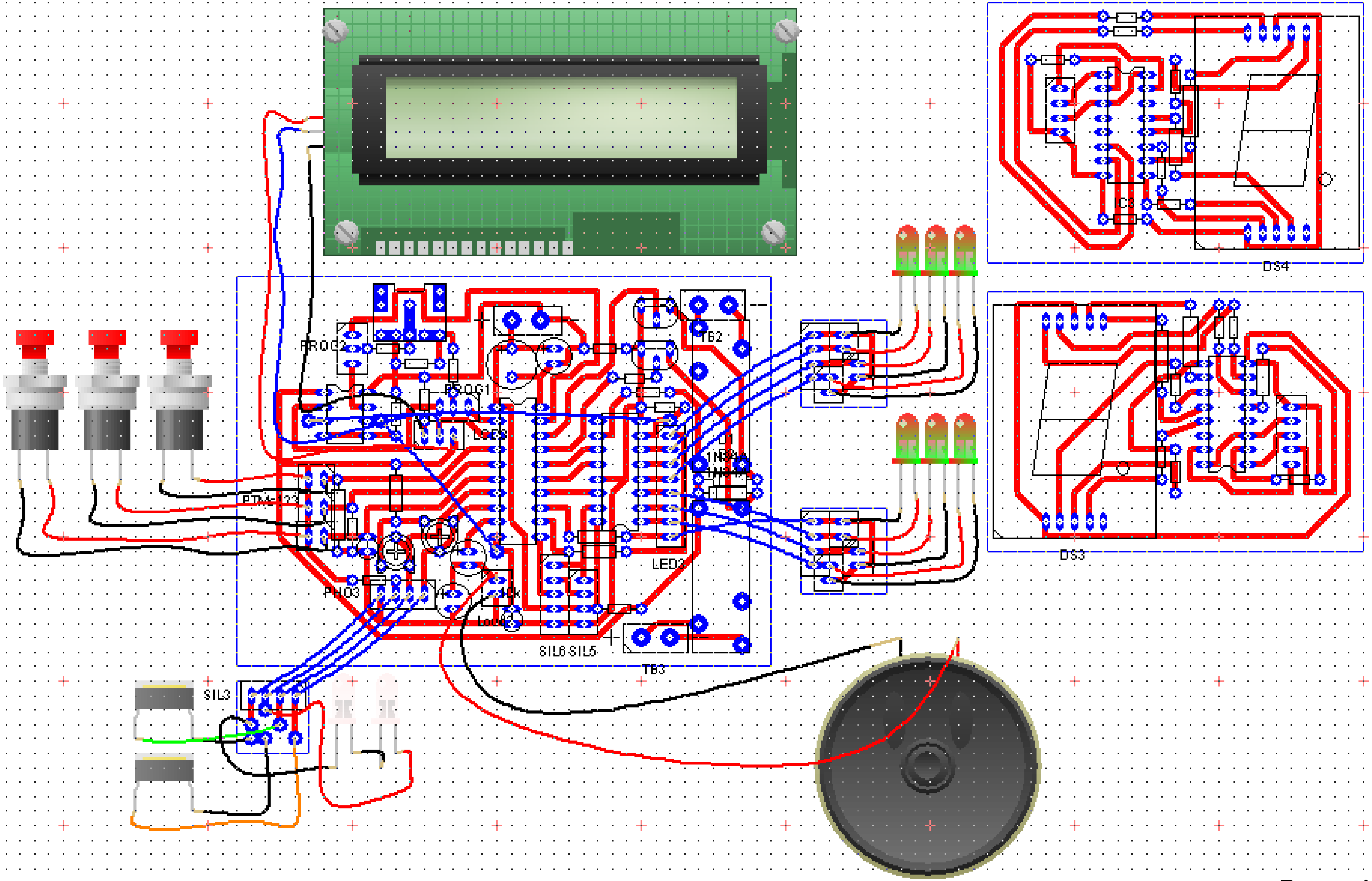


Finally added off board components to design for simulation purposes.

Final Circuit Design →

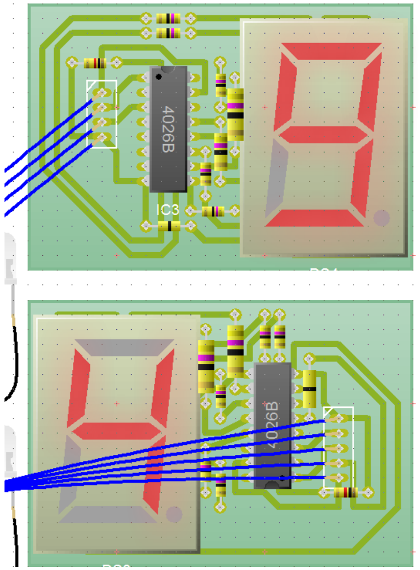


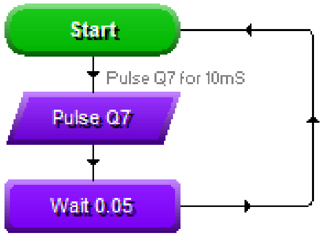
Circuit Board Design



Simulations

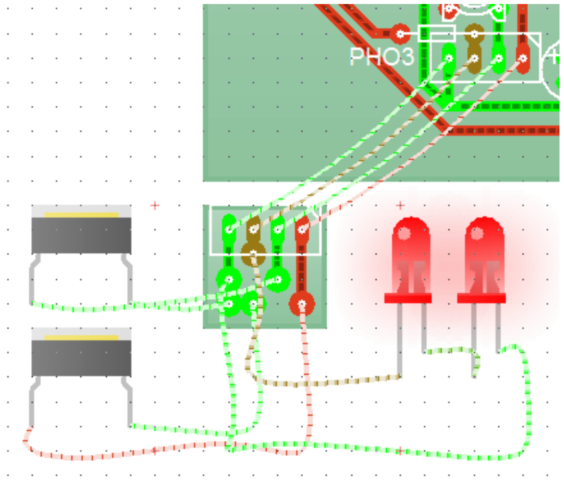
Seven Segment Displays





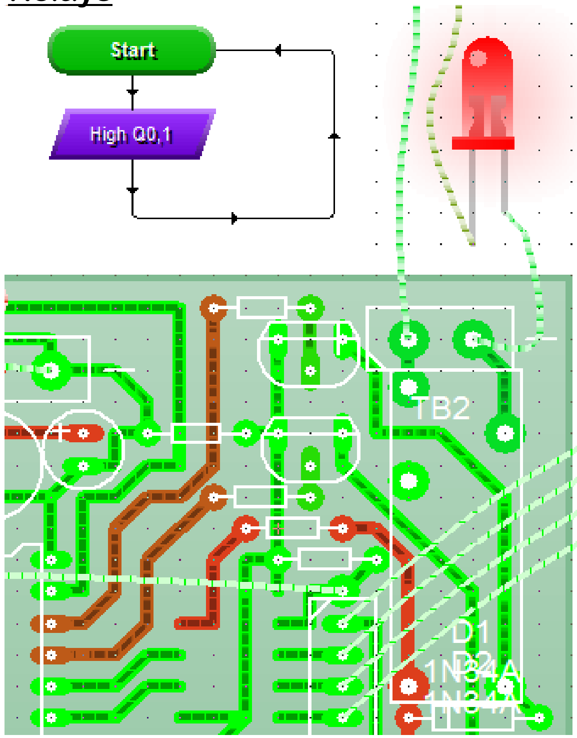
This program simply clocks the seven segment displays to check they are all connected correctly, and communicating for the 10's output.

Photo Transistors



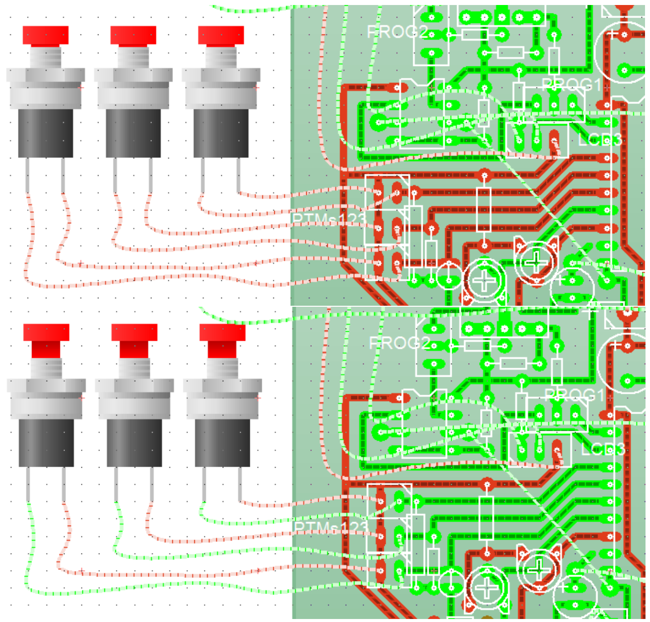
Both LED's are illuminating correctly and also as you can see from the current flow, the photo transistors (which are set to opposite ends of the scale) are also working.

Relays



Both relays work but this just demonstrates the top one making the circuit and illuminating the LED.

Push Buttons

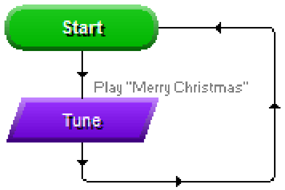


All the PTM switches are working as demonstrated here with a current flow diagram, all on and then all of.

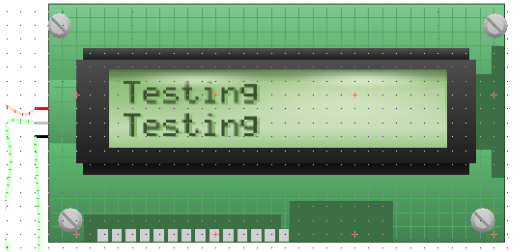
Simulations

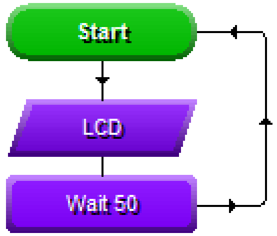
Loudspeaker

This program demonstrates the loud speaker is connected correctly by the sound waves being given out.



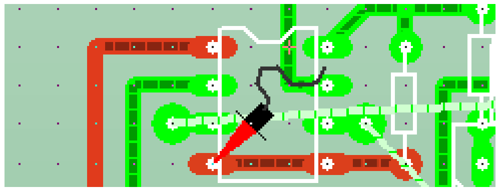
LCD Display

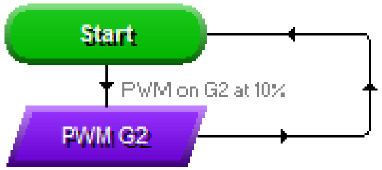


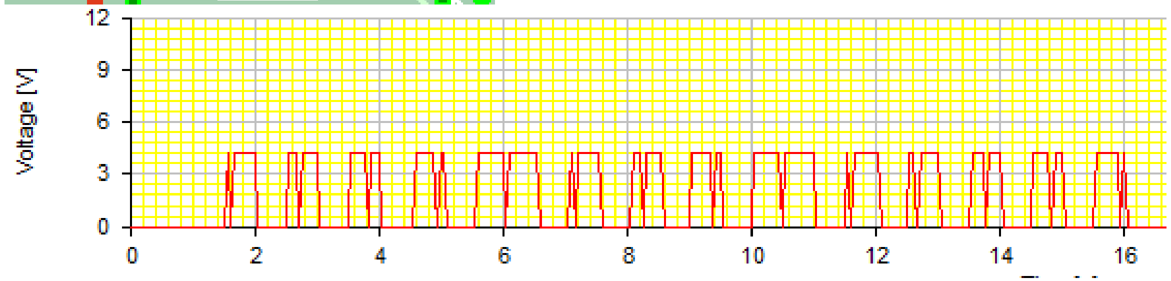


This simple program demonstrates the LCD display works correctly in the current configuration.

PWM Real time Clock

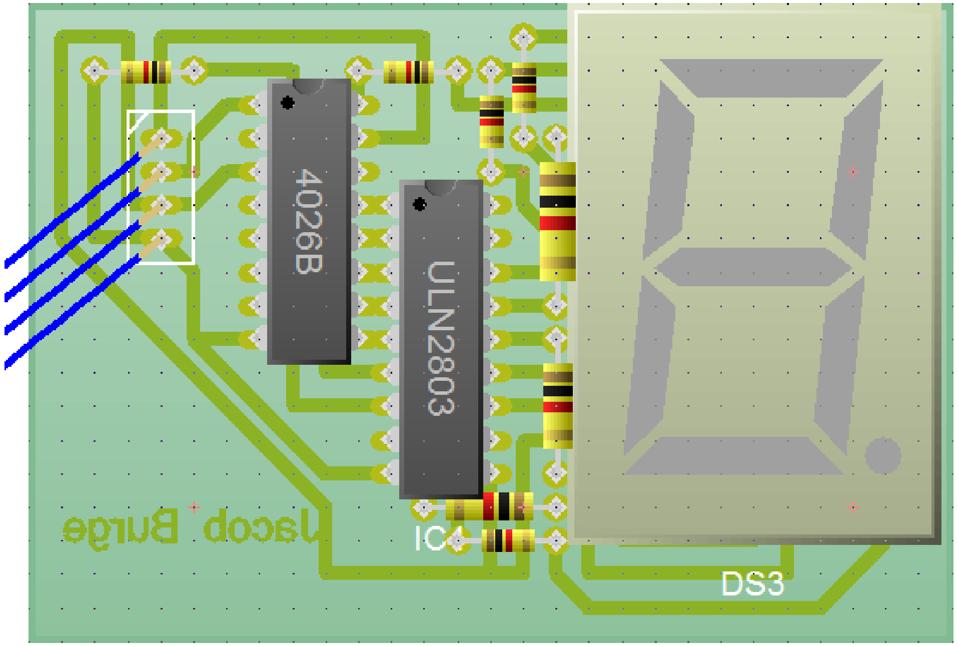




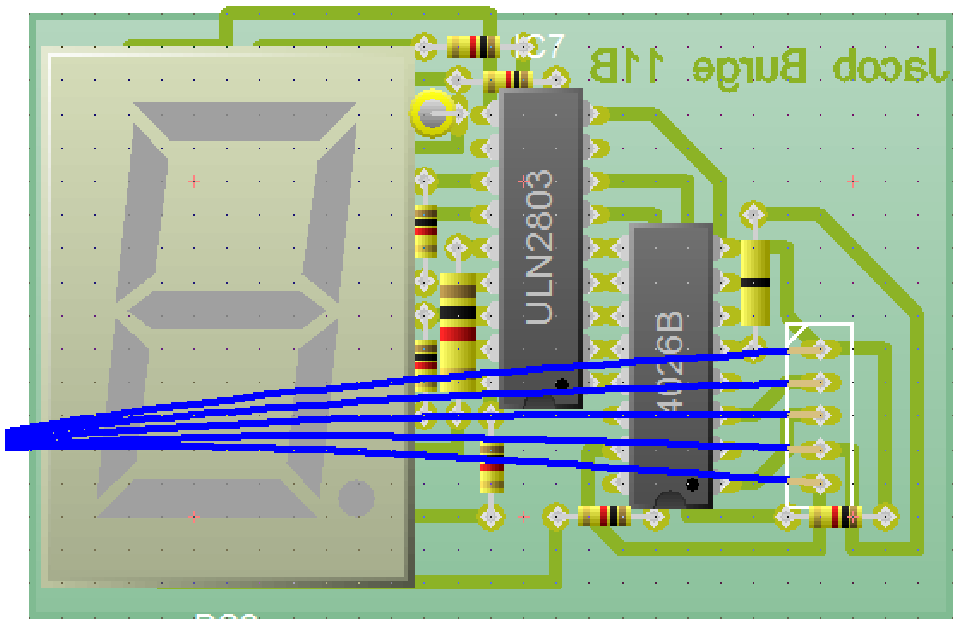


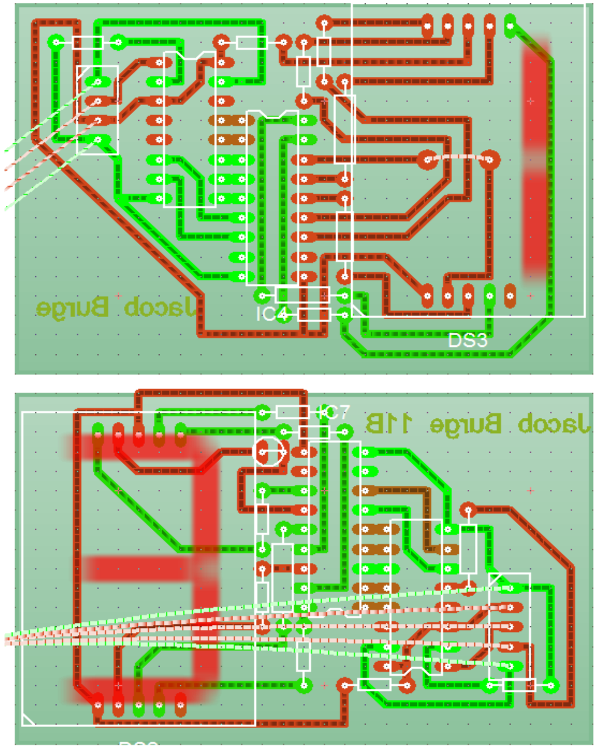
This circuit demonstrates the PWM real time clock on the 08M2. From the diagram you can see the mark part of the mark space ratio given out by the PWM from the 08M2. Bellow is a graph of the PWM working, although it is not very easy due to the limits of simulation it proves that it will work in the real world. The rest of the functions for this chip is all software based.

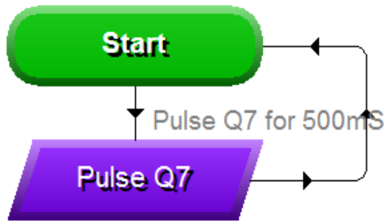
Seven Segment Problem



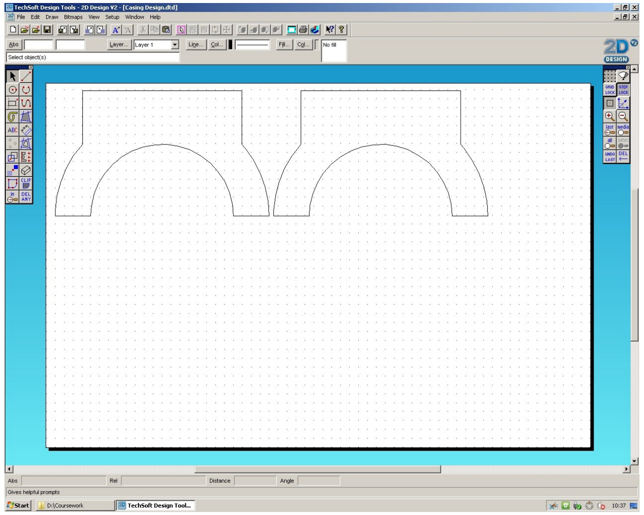
Unfortunately, it appears to be hard to come by a 1.2” seven segment display with common cathode. Rapid only sell common anode ones and no were else on the web appears too. As a result I have changed my design to use common anode displays; therefor a Darlington array (uln2803) is required in order to convert the high signal from the 4026b driver and decoder chip to one which sinks current from the displays. Luckily it has worked quite well with the board not turning out too much bigger; also the displays will be brighter as I will have more control over the brightness through adjusting the resistors to limit the current. Below is a screenshot of the new setup working along with the basic piece of code which increments the count by 2 every second:



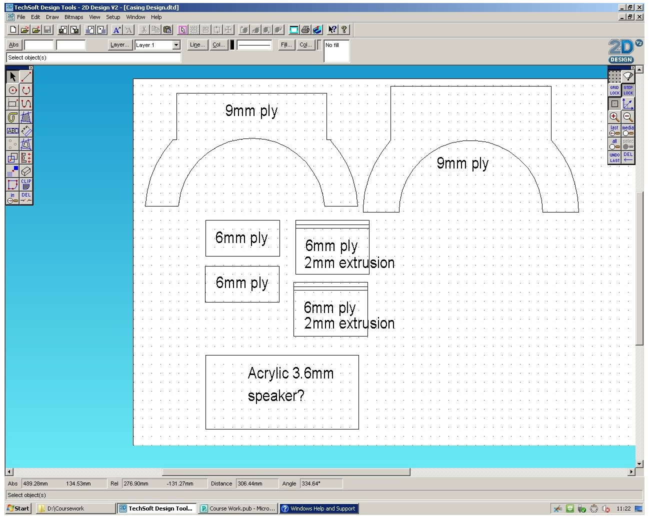




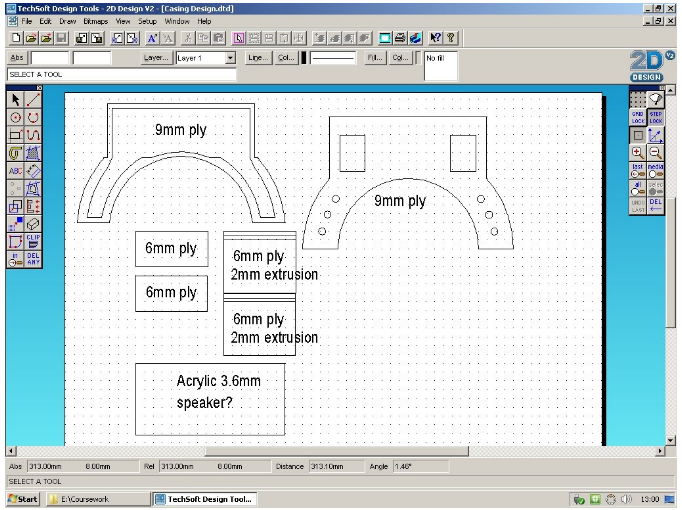
Casing Design



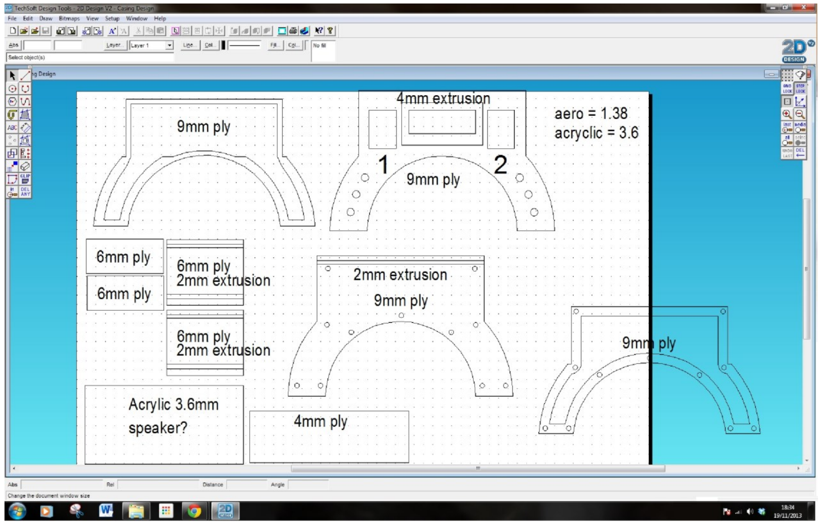
1. Initial outline



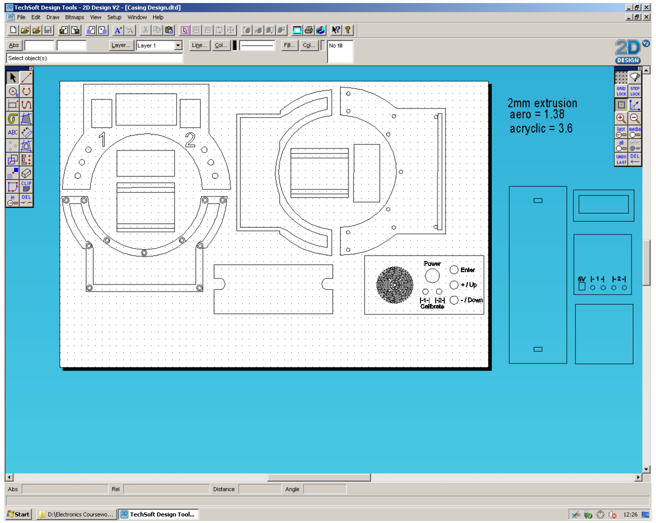
2. Side walls and top with extrusions for top to slide into



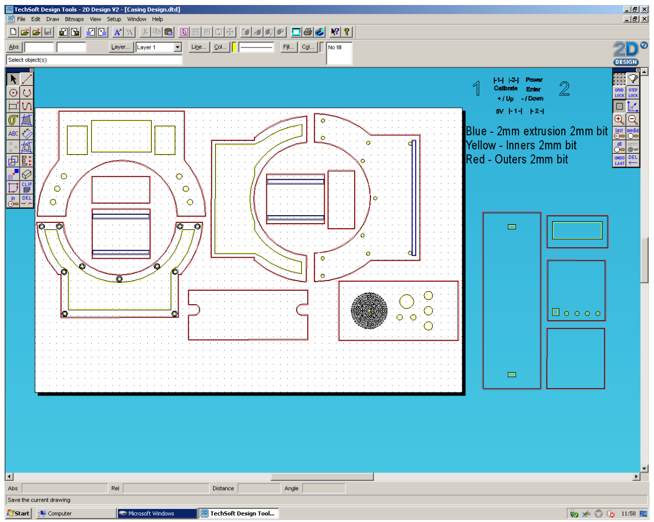
3. Component holes added



4. Internal supportive frames added

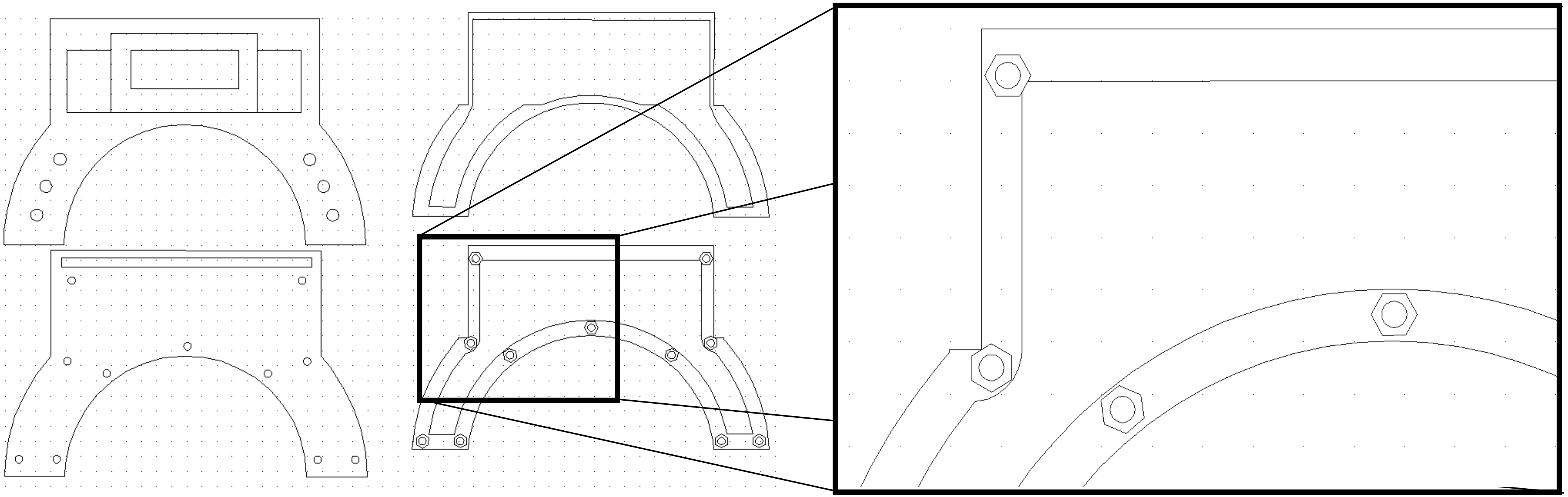


5. Final details and layout on CNC bed



6. Tool path contouring

Removable Back Design



To allow access to the electronics inside the casing I needed to make one of the panels removable. I decided to use the back as it would provide access to all the regions of the casing and it was the easiest face to make removable. I decided that I would use bolts to fix the back on securely but still allow the face to be removed. However this required the use of a nut on the other side to ensure the bolt was secure. I decided to recess the bolt shape within the internal frame allowing me to push the nut in and it stay there. From the picture you can see the nut shape which is recessed and then a central hole which is were the bolt comes through. However I have had several problems with this system; firstly because the frame had to be quite small to allow all the electronics to fit within the case, the nut recesses went over the edge in some places. Although this came out okay, it lead to the structure being significantly weakened leading to one of the corners of the frame snapping. Another problem I faced was the nut not being correctly lined up with the bolt which made its own thread in the wood, this lead to the nut being pushed out of the frame and all the bolts having to be undone to re insert this nut. I tried hot gluing them in place, however this was not strong enough – next time I would make the bolt holes bigger so that this did not happen. Finally some of the nut locations interfered with other electronic components meaning I had to cut the bolt off and glue in place for a decorative feature; offering no extra strength. Although overall the sytem is now effective and with a bit of refining definitely the best option.

Parts List

PCB Parts List

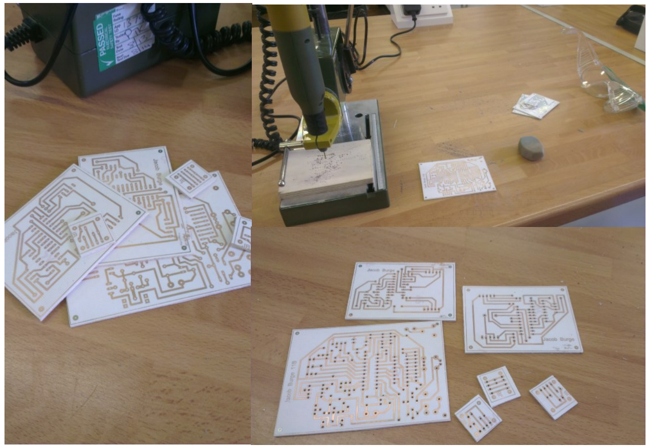
Name	Cost	Quantity	Total
0 Resistor (1/4W)	£0.00	2	£0.01
0.5 (W) x 0.5 (H) in Printed Circuit Bo	£0.06	1	£0.06
0.6 (W) x 0.6 (H) in Printed Circuit Bo	£0.24	2	£0.48
0k Resistor (1/4W)	£0.00	13	£0.04
0k Resistor (1W)	£0.00	2	£0.01
100µF Electrolytic Capacitor	£0.04	4	£0.16
100k Potentiometer	£0.05	2	£0.10
10k Resistor (1/4W)	£0.00	5	£0.02
10k Resistor (1W)	£0.00	1	£0.00
1k Resistor (1/4W)	£0.00	11	£0.03
1k Resistor (1W)	£0.00	1	£0.00
1N34A Diode	£0.01	2	£0.02
2.6 (W) x 1.8 (H) in Printed Circuit Bo	£2.10	1	£2.10
22k Resistor (1/4W)	£0.00	1	£0.00
2-pin Terminal Block	£0.95	3	£2.85
3.7 (W) x 2.7 (H) in Printed Circuit Bo	£3.78	1	£3.78
330 Resistor (1/4W)	£0.00	1	£0.00
4026B Decade Counter	£0.46	2	£0.92
7-Segment Display (Common Cathod	£1.62	2	£3.24
BC548B NPN Transistor	£0.08	2	£0.16
Download Socket	£0.39	1	£0.39
PICAXE 08M2	£1.82	1	£1.82
PICAXE 20M2	£2.42	1	£2.42
Red/Green Bi-Colour LED	£0.20	6	£1.20
SPST Switch	£0.57	3	£1.71
IR Detector	£1.45	2	£2.90
LCD Display	£7.00	1	£7.00
Relay	£0.70	2	£1.40
SIL Pins	£0.89	1	£0.89
3.5mm Jack	£0.27	2	£0.54
			£34.25

Casing Parts

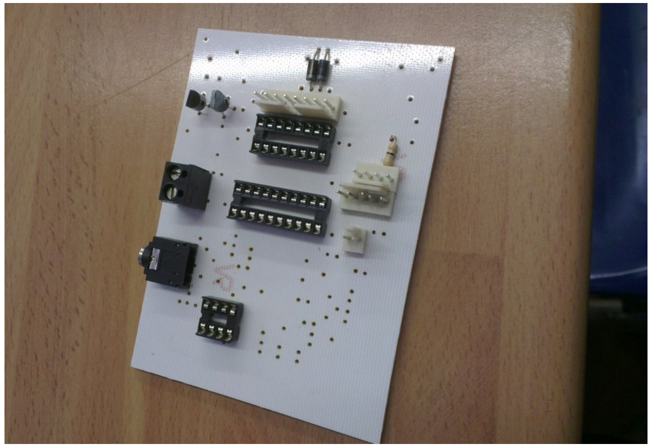
Name	Cost	Quantity	Total
Clear Acrylic	£26.88 per m2	100x80mm	£0.25
Aero Ply	£1.75 per sq ft	205x100mm	£0.39
9mm Ply	£1.40 per sq ft	320x200mm	£0.96
4mm ply	£1.20 per sq ft	280x100mm	£0.36
M5 nut and bolt	99p pack of 10	1	£0.84
			£2.80

Total Cost
£37.05

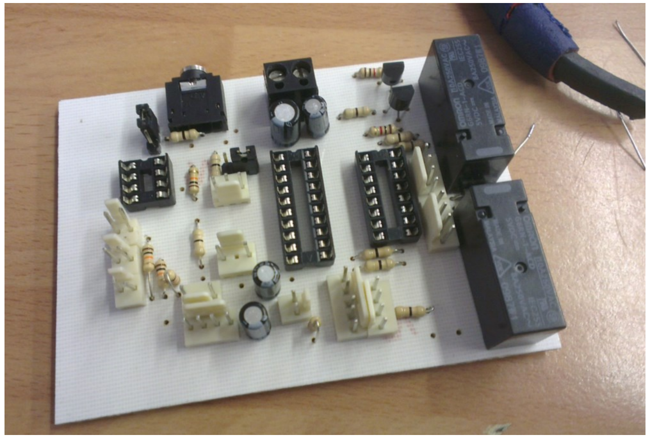
PCB Manufacture — 1



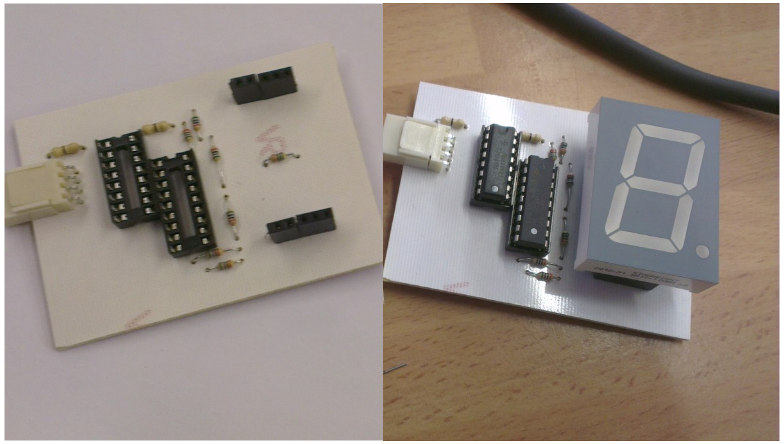
Drilling the component holes



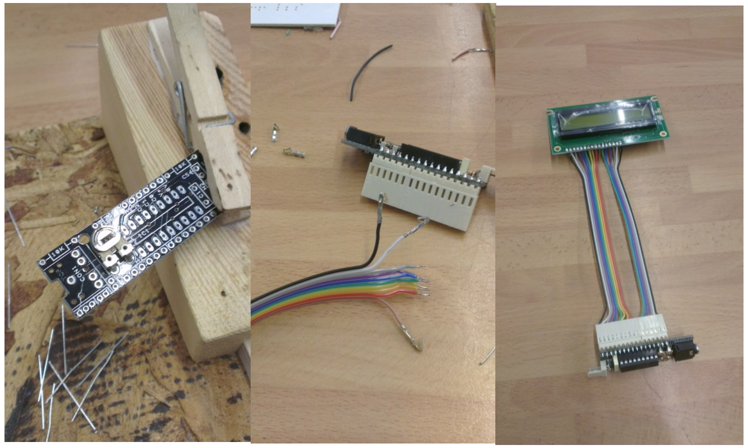
Beginning to populate the PCB



Almost fully assembled main board



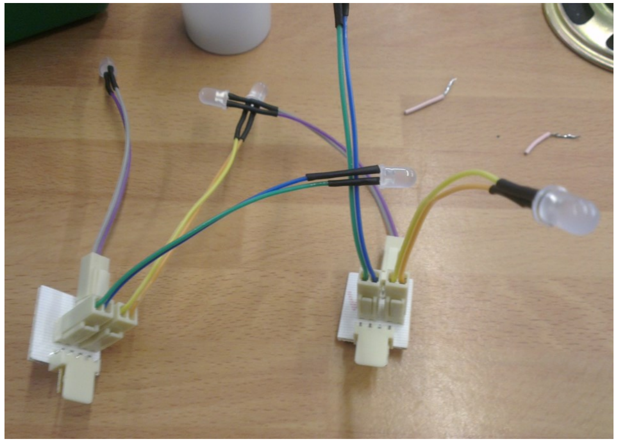
Seven segment display boards



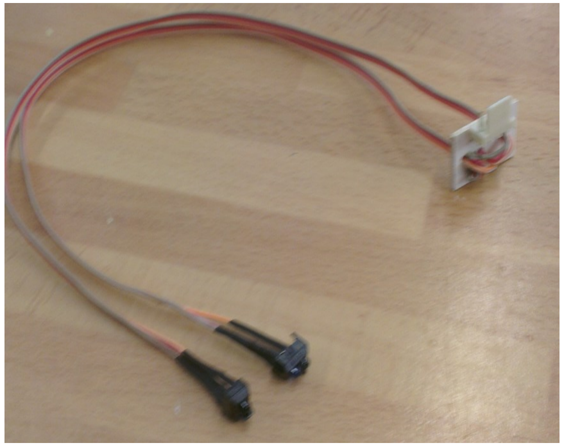
LCD control board assembly

Loudspeaker ready to be wired up

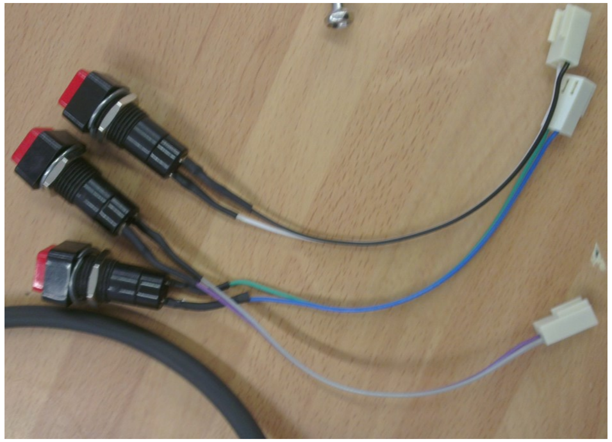
PCB Manufacture — 2



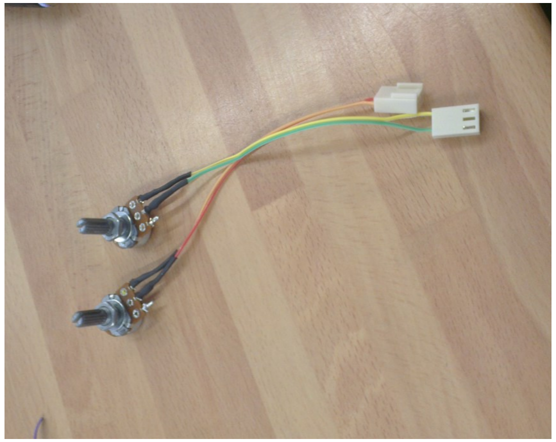
LEDs crimped and boards assembled



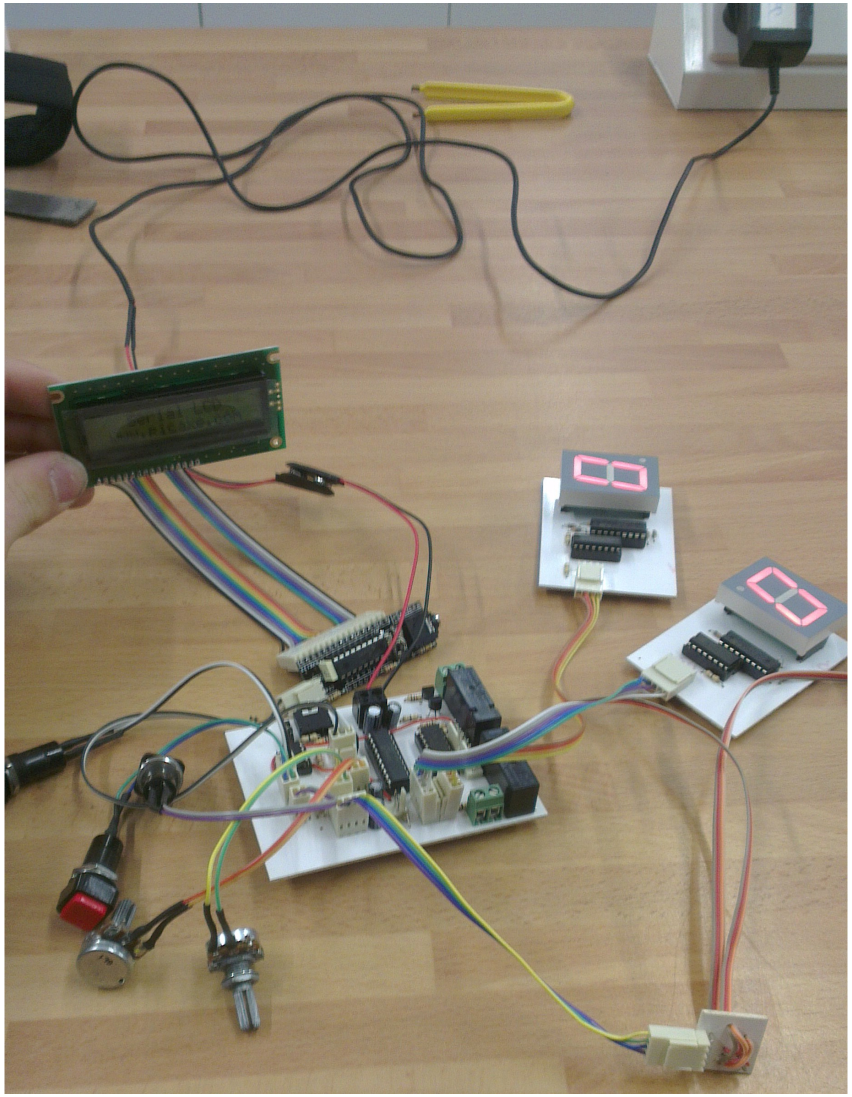
IR detectors wired up



PTM switches with crimps

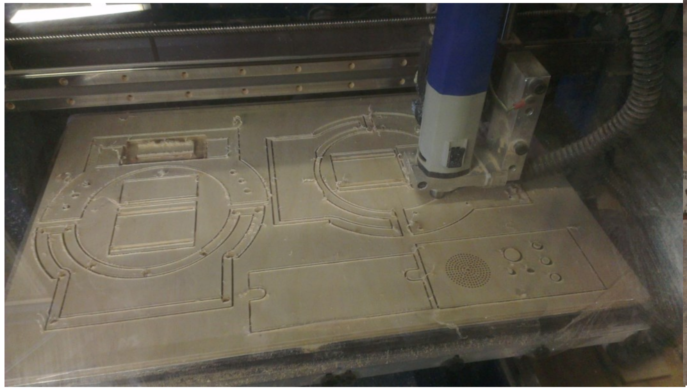
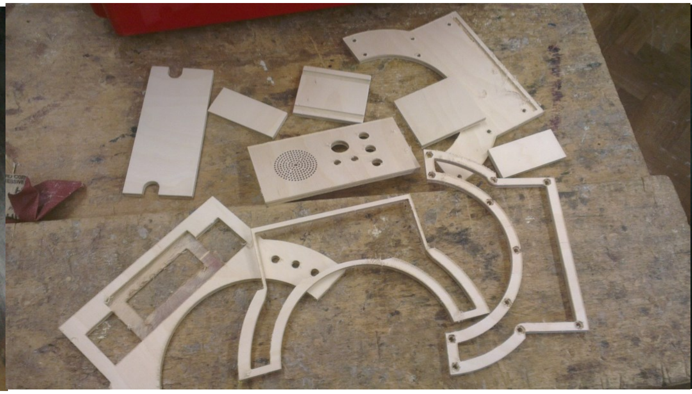


Potentiometers with crimps attached

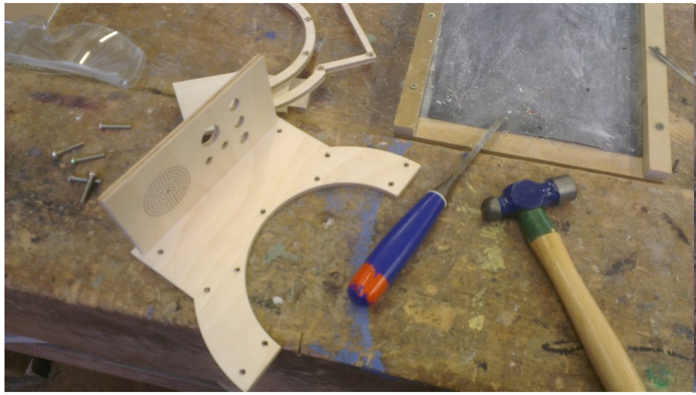


Completed circuit board

Casing Manufacture — 1

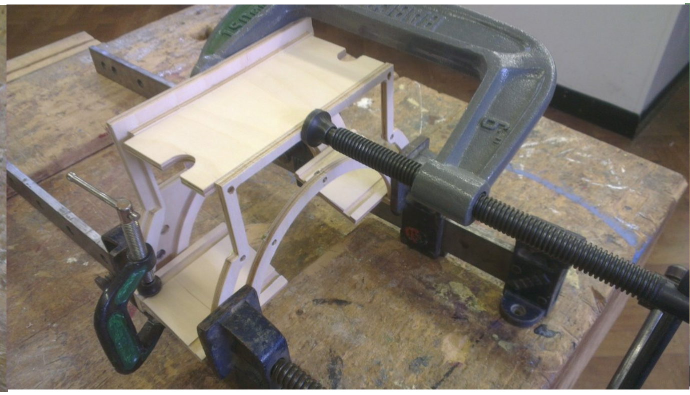


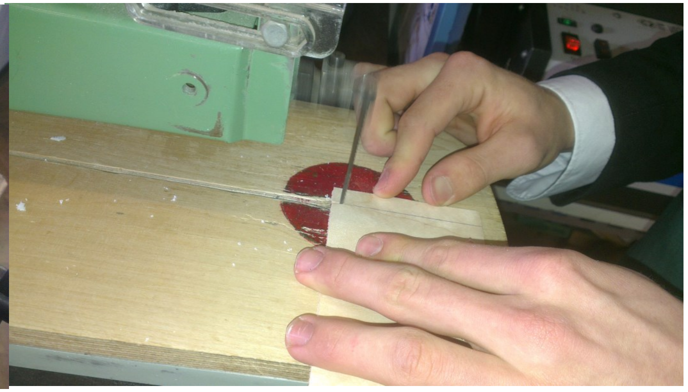
15.1.14 — After designing the casing on 2D design it was cut on the CNC Milling machine in 6mm ply. Initially all the parts came out with rough edges so required some clean up work before assembly could begin. I used mini files and glass paper to tidy up each of the pieces.



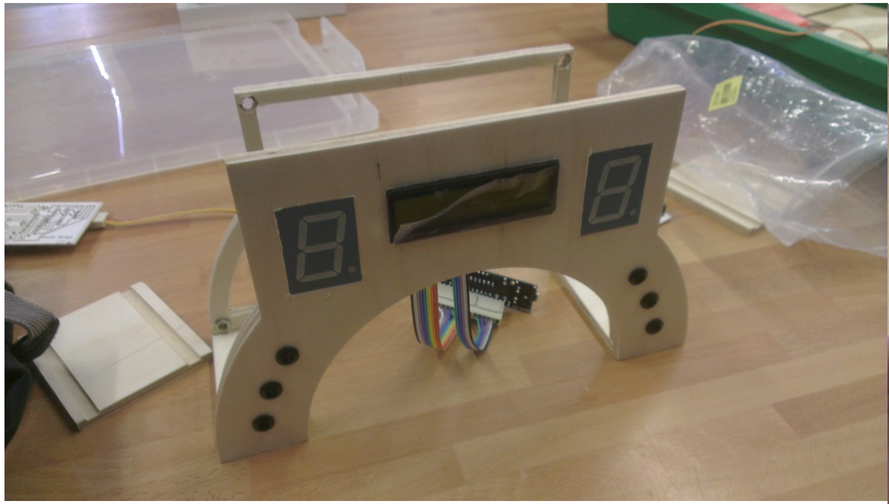
16.1.14 — I realised that my last minute change to the whole thing being cut from 6mm ply had lead to the slot from the back panel being the incorrect size; I used a chisel to fix the issue. Upon roughly fitting the parts together I realised that the change in material thickness had led to quite a few problems, however it did fit relatively well. Finally I glues the structural frame to the back of the front panel and left under some sanding discs to set

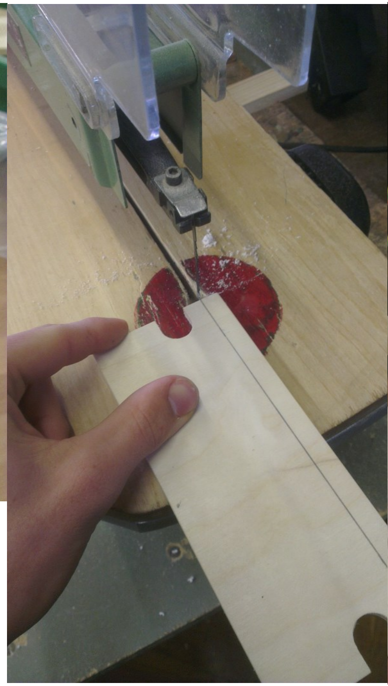
Casing Manufacture — 2

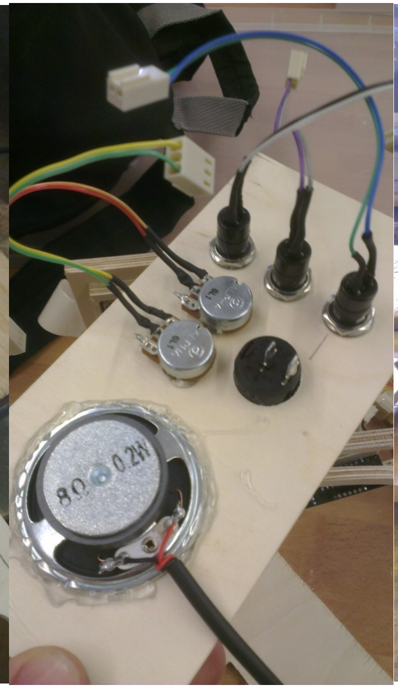




17.1.14 — Next I pushed each of the nuts into the front of the back structural frame ready for it to be glued it place. Now I glued and clamped the base plates and back structural frame to the front panel and front structural frame. I used the side and top panel to ensure the others were square. I also cut the aero for the curved sides down to size.

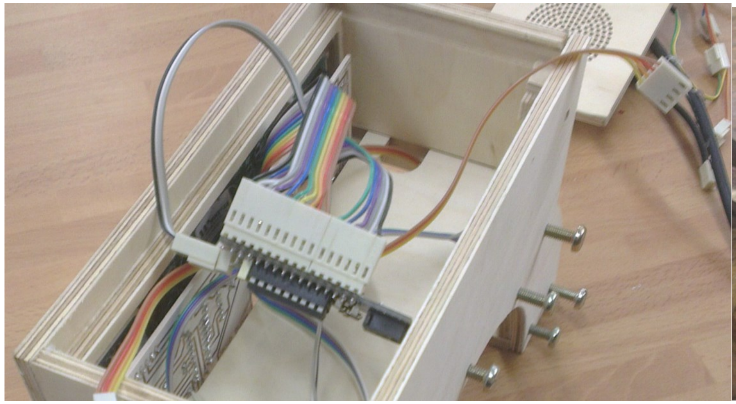


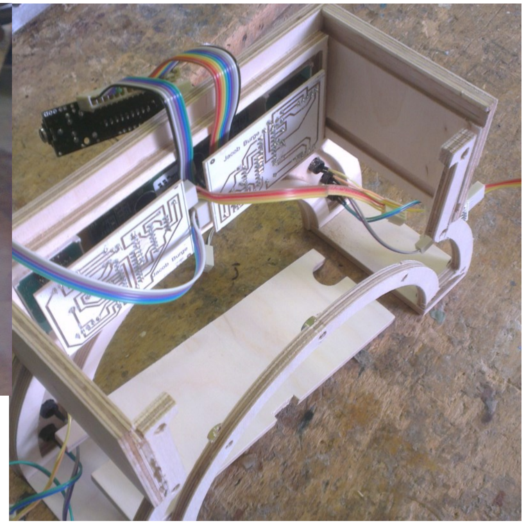




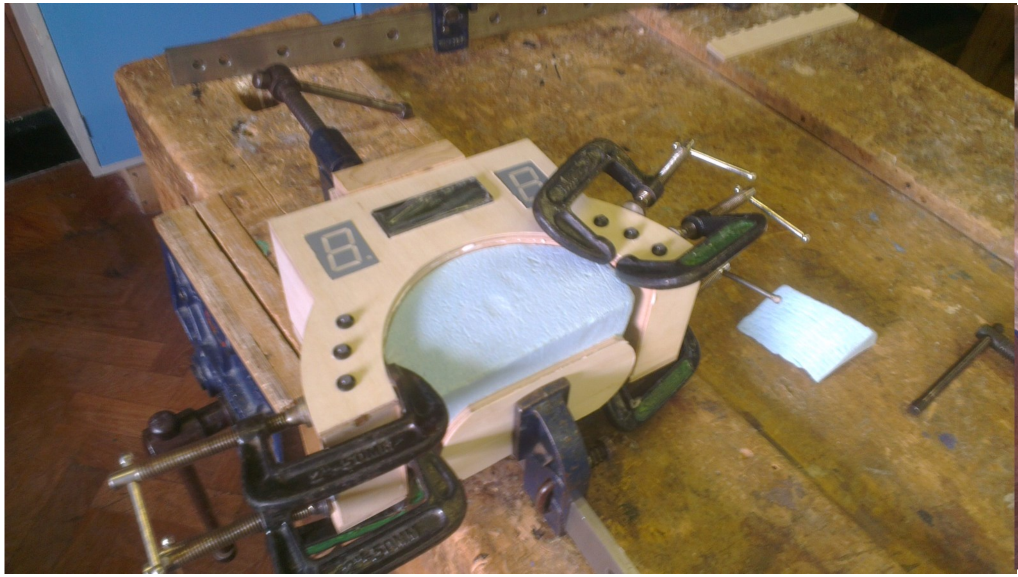
20.1.14 — Firstly I populated the front panel and top panel with the panel mount components. Then trimmed the circuit board support to enable me to slide it in from the side rather than the back, because I had forgotten about this step before gluing the back on. Then I finally set up the side walls to glue and clamped it.

Casing Manufacture — 3



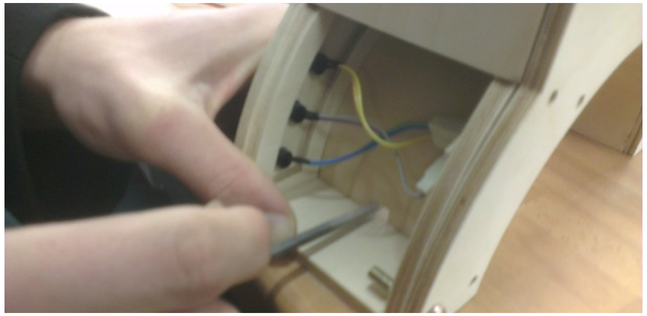


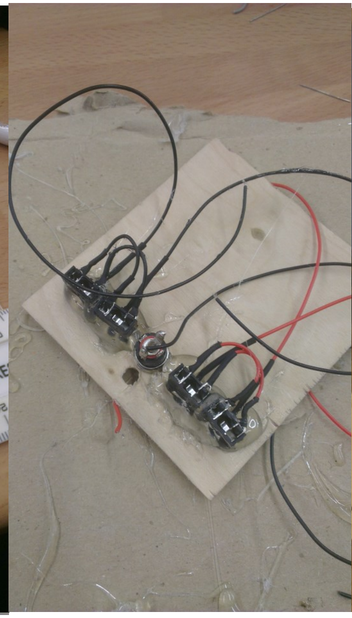
24.1.14 — I had to remove part of the back support panel as it prevented the top panel sliding in populated. Once this was done I installed the PCB support panel and hot glued it in place, then finally slid in the top panel and attached the rear panel; it fitted well.



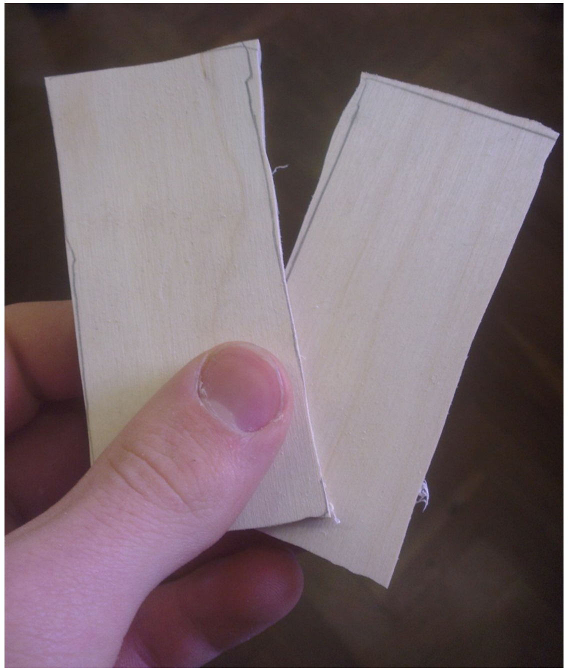
28.1.14 — Next I worked out how to glue the middle arch into place and then glued it. In the end I used a piece of blue foam cut to the radius of the arch to push it against the support panel for gluing. I used a sash clamp with a scrap piece of ply to evenly distribute the pressure across the foam holding the aero ply in position

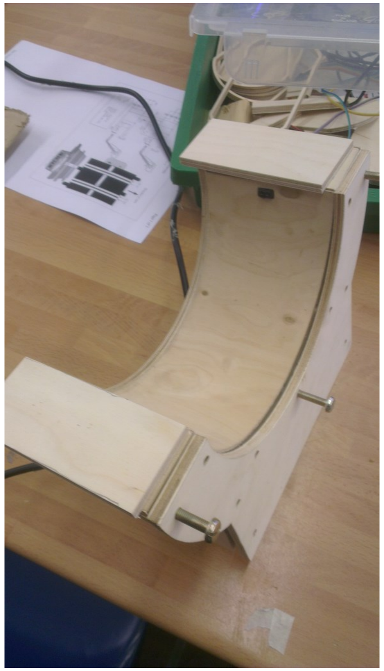
Casing Manufacture — 4





<u>29.1.14</u> — Once the central arch was glued in place I used the mini drill to cut out a rectangle and then I filled it down so the IR proximity sensor could fit in. I also measured out, drilled the holes and mounted the 3.5mm jacks, 2.1mm DC jack and power LED in the side panels and finally glued them both in place.



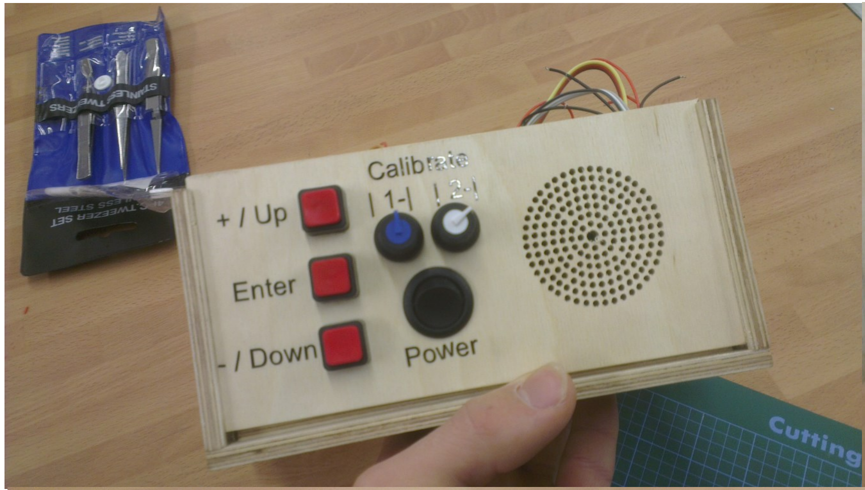


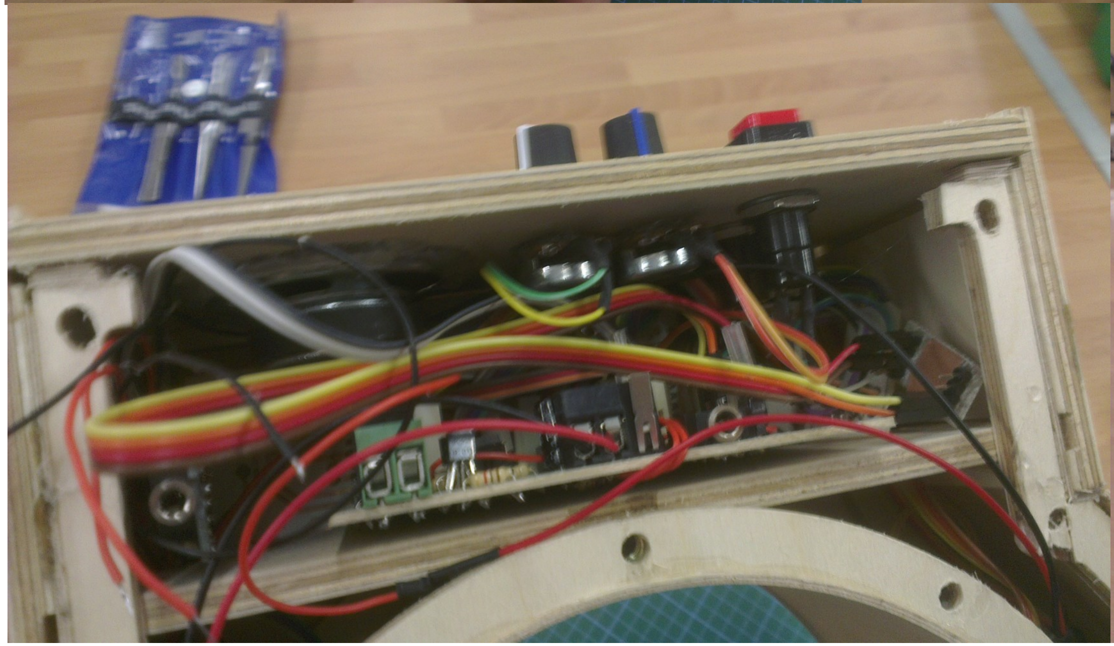
<u>31.1.14</u> — Finally I had to add some extra ply to the base of the start gate because I had underestimated the curve of the arch and consequently the clearance between car and roof was very tight. I cut out these two ply rectangles and glued underneath to add 6mm of extra height. I also added some rubber feet which prevented the start gate slipping, and added another 3mm of height.

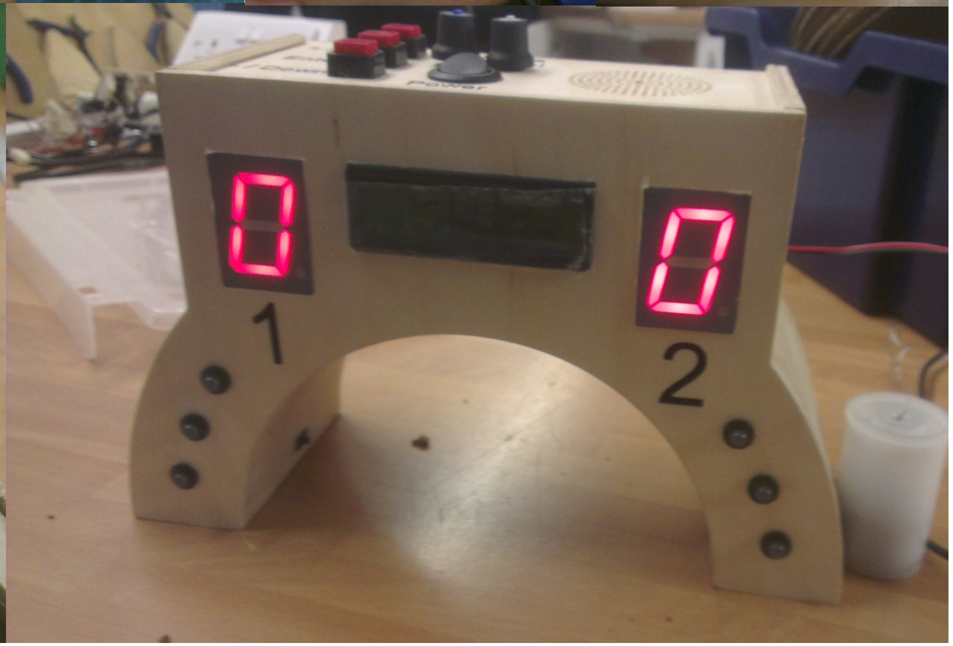
With the casing finished I had a quick little tidy up, removing any pencil marks and sanding down some of the edges further to reduce the overhangs of parts which did not fit exactly. I also attempted to glue in some of the nuts which secured the back panel, however this was very fiddly and the glue gun was to bulky. Consequently this had little effect. Finally I put the PCB inside and connected all of the off board components up.

Casing Manufacture — 5

<u>3.2.14</u> — The very last thing to do was add the vinyl's which label each of the buttons and make the casing more aesthetically pleasing as well as easier to use.





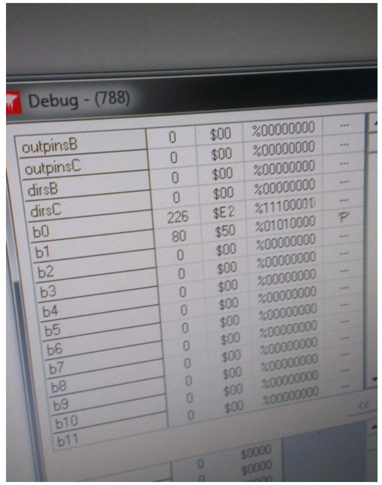


Circuit Testing

IR Detectors

The IR detectors are on pins c.2 and c.3 of the 20m2; after writing this quick piece of sample code which does a "readadc" on both pins and then sends the values to the computer it is apparent that they both work and appear to be sensitive enough.

```
main:
'check track 1
readadc c.2,b0
'check track 2
readadc c.3,b1
'send data to computer
debug
goto main
```

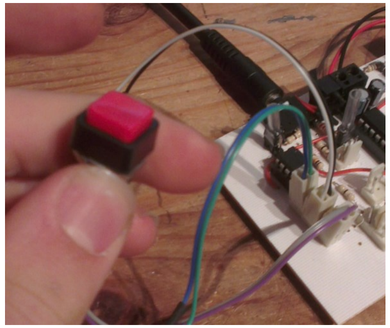


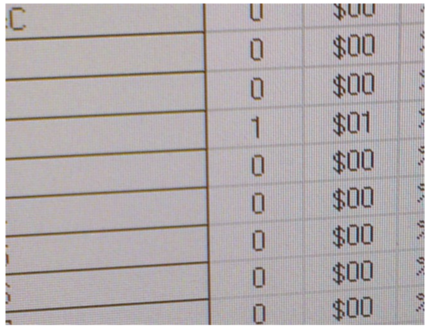
LCD Display

This also worked first time, although initially I had some problems with ascii characters displaying instead of the number I had sent to the display however this was easily solved by looking back at some old code and discovering a "#" in front sorts that out.

Loudspeaker

The loudspeaker caused some problems, at first no noise was produced when I loaded up this sample code; after further investigation it turned out one of my filtering capacitors was of the wrong value. After this the sound could be heard but it was very quiet. It turns out that the loudspeaker was to big to be driven by the PIC; consequently I had to downsize and afterwards it worked perfectly.



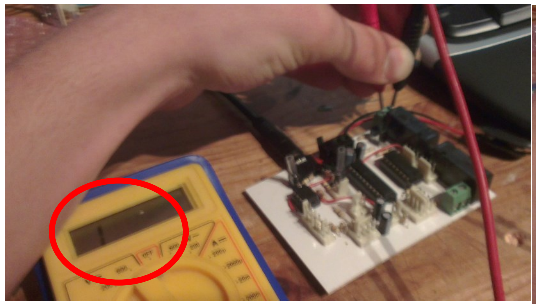


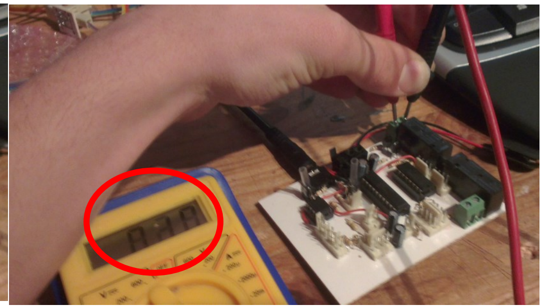
PTM switches

These worked fine and my sample code made it easy to check.

Relays

The relays proved difficult to install as they were different to the template on circuit wizard. This meant I had to drill a few extra holes. I also had a problem with the back EMF diodes being the wrong way round meaning they shorted the relay coil stopping it from working.

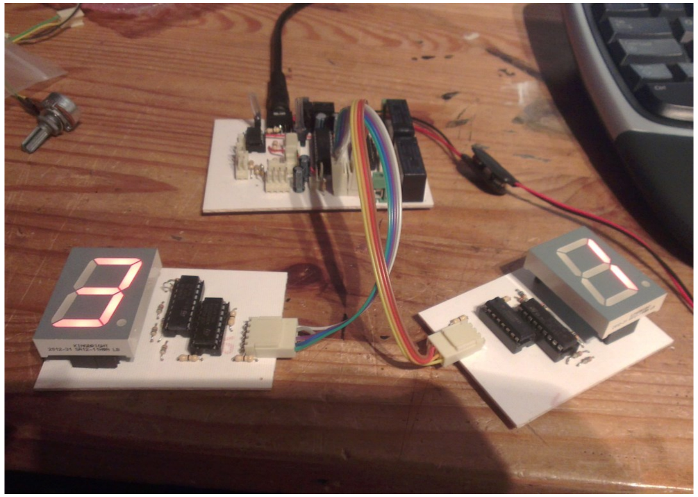




Circuit Testing

Seven Segment Displays

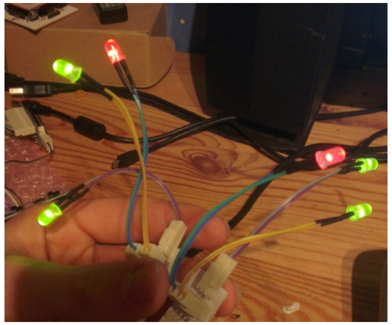
```
;set valuse on each display
b0=3
b1=1
;combine individual valuse
b2=b1*10
b2=b2+b0
;clock in data do display
do
inc b3
pulsout b.7,1
loop while b3<b2
;wait
pause 20000
```



The seven segment displays have proven very problematic; firstly I had some distortion in the numbers which was evidently a short between two segments. I found it and it turns out it was in the original design; after solving this however the display would only count with the multi meter between reset and ground (I discovered this whilst poking around with the voltmeter). So I decided to put a 10K pull down resistor on the reset line; I think that this problem has something to do with the reset being connected to the shift register. After getting the basics working I wrote a sample piece of code which allows you to input two numbers and it displays one on each display — this is what it will be used for so it was good proof of concept. It does this by combining the two numbers and then pulsing the clock line really quickly this number of times.

Shift Registers

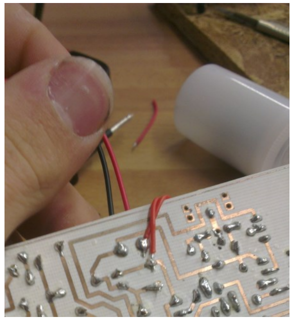
```
;subroutine to update shift register
update:
low b.3
gosub shiftout_MSBFirst
pause 100
high b.3
pause 100
low b.3
return
```

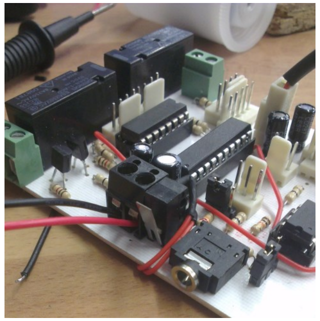


Surprisingly the shift register was very easy to get running after checking my pin allocations numerous times I was able to gain control of the LEDs. As I am not really sure how the start sequence will work yet I am still playing around with combinations but the setup to give a lot of flexibility which is good.

Reset Button

Whilst testing all of the individual components of the circuit I realised that it would be a lot easier if I were able to reset the chip without having to unclip the power connector. So for the purposes of testing I added a PTB microswitch in line with the power line. This cuts of the power to the circuit without me having to disconnect it manually.



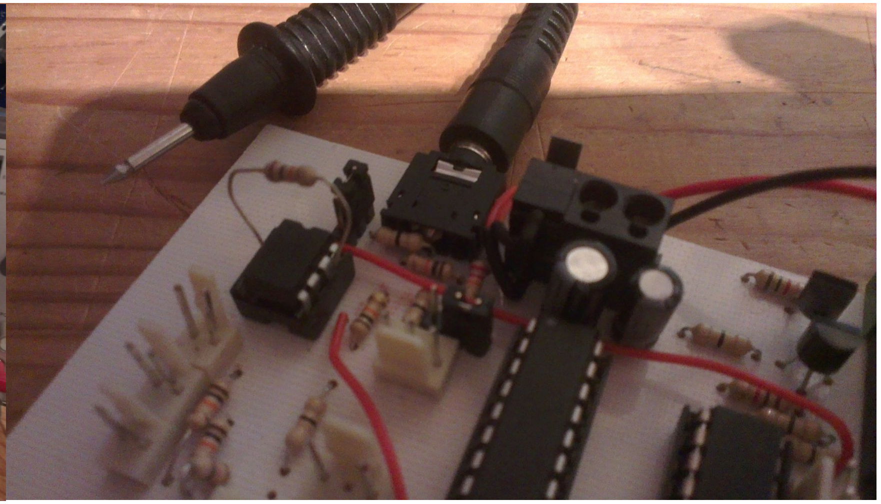


Real Time Clock

08M2 Serial Communication

This like I predicted has taken up the most time to work out why it was not working initially. After checking over my code I was sure everything was correct and there were no continuity errors with the connection between chips. Because I have a shared programming socket I was only able to monitor one chip at a time using the debug function; consequently I was unaware of the fact that the 08M2 was in fact not working when I was watching the 20M2 on the serial monitor. I used an oscilloscope to monitor the serial data line and discovered in fact the 08M2 only worked when connected to the programming socket. From here it was obvious that the 10K pull down on the serial in line was in the wrong place. After changing it so each chip had its own pull down resistor everything worked fine. I also took the opportunity to check the accuracy of the PWM on the 08M2. It looks good.

Serial line goes dead when programming socket is connected to 20M2.



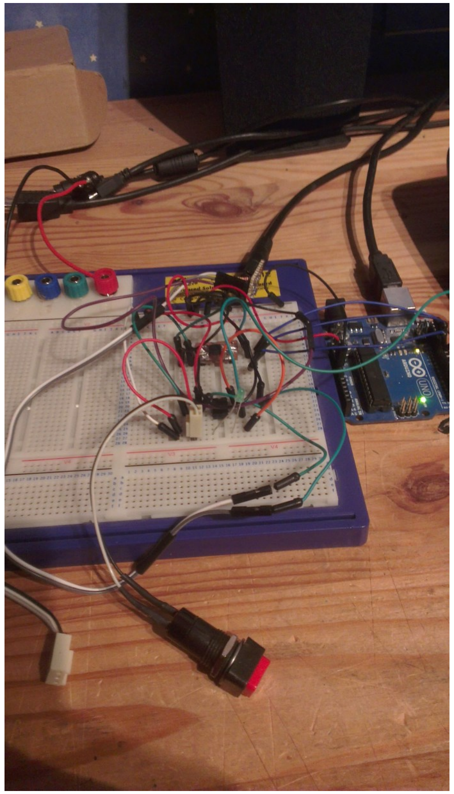
10K resistor solves the problem.

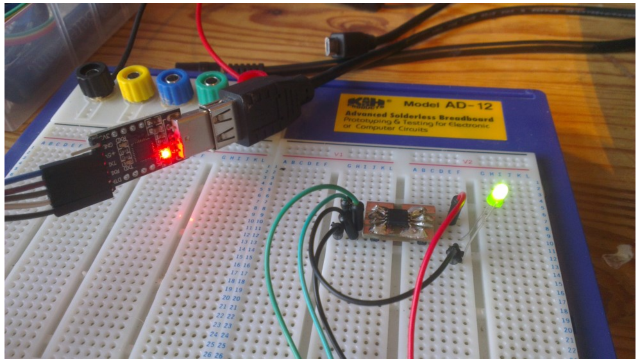
PWM works well (20% duty).

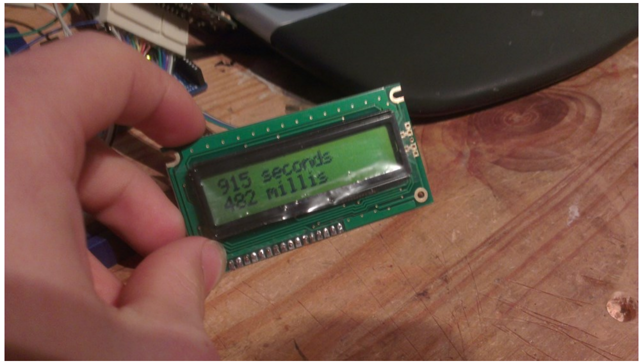
Although the PWM looks fine from the oscilloscope trace the timing is not accurate at all. I believe that the interrupt is not quick enough and therefor misses out pulses from the PWM line. Unfortunately it is not possible to change the clock frequency as too achieve a PWM frequency of 100Hz, the clock can only be at 4Mhz. This means that I think I will have to replace the 08M2 with a different chip capable of internal timer interrupts.

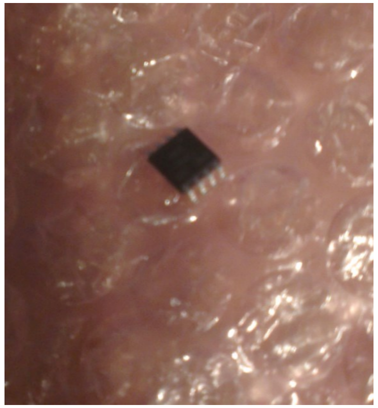
Solution to 08m2 problem

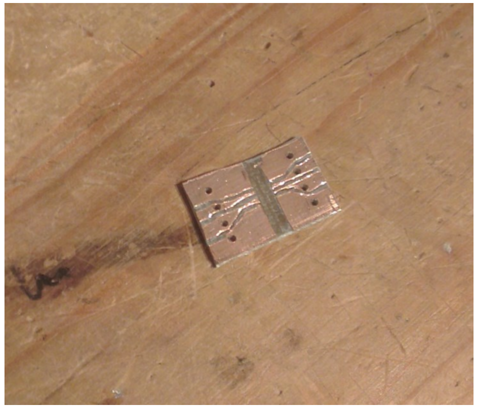
My solution to the problem uses an ATMEL chip, the ATTINY85 using the Arduino boot loader. I decided upon this chip as it was small and is able to carry out 3 interrupts with good precision. The Arduino boot loader utilizes timer 0 for a function called millis(), which returns the number of milliseconds which the program has been running. I designed and made a breakout board for the chip as it was surface mount. After this I hooked it up to a breadboard and used the Arduino uno as an isp to program the chip. I wrote a simple piece of code, which when a pin is pulled high by the picaxe 20m2, the millis() is read and then sent to the 20m2 via serial in 3 word variables. This took some experimentation to workout the serial which is sent by the Arduino and which command was required to successfully read it on the picaxe. I experimented with using a qualifying byte in front of the 3 word variables to try and improve the reliability of the communication; however it just made it worse. After playing around with baud rate I finally settled upon N4800 as it lead to reliable, fast communication. Although occasionally it does still provide the wrong number, this is very infrequent and overall works very well providing timing to 1/1000th of a seconds and capable of recording races up to 70 minutes long.

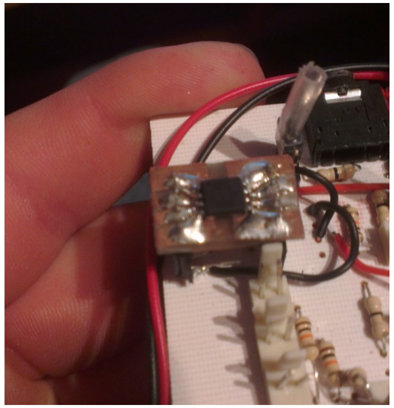












Serial Communication

PICAXE Programming Editor - [D:\Electronics Coursework\Programs\Attiny comunication.bas]

```
main:

if pinc.6=1 then
var_out=%10000000
gosub update
goto recieve
var_out=%00000000
gosub update
pause 500
endif
goto main

recieve:
serin c.0,T4800_4,b0
serin c.0,T4800_4,b1
serin c.0,T4800_4,b2
serin c.0,T4800_4,b3
debug

gosub cleardisplay
gosub cursorline1
serout C.7,N2400,(#w0," seconds")
gosub cursorline2
serout C.7,N2400,(#w1," millis")

pause 250
goto main

cursorline1:
 serout C.7,N2400,(254,128)
 return

cursorline2:
 serout C.7,N2400,(254,192)
 return

cleardisplay:
serout C.7,N2400,(254,1)
pause 30
return

update:
low b.3
gosub shiftout_MSBFirst
pause 100
high b.3
pause 100
low b.3
return
```

If button is pressed

Update shift register to pull pin 2 on ATTINY HIGH

Receive time

Update shift register to pull pin 2 on ATTINY low

Receive 4 bytes from ATTINY, stored into Word variables

Update display with elapsed time

Sub routines for LCD Display

Sub routines for shift register

SoftwareSerialBlinkattiny85 | Arduino 1.0.5

```
#include <SoftwareSerial.h>

SoftwareSerial TinySerial(3, 4);

void setup()
{
  // Open serial communications a
  TinySerial.begin(4800);
pinMode(PB1,INPUT);
}

void loop()
{
if(PB1==1){
unsigned long int milliseconds =millis();

word seconds = milliseconds/1000;

word millisecs = milliseconds - (seconds*1000);

byte b = highByte(seconds);

byte a = lowByte(seconds);

byte d = highByte(millisecs);

byte c = lowByte(millisecs);

TinySerial.write(a);
delay(10);
TinySerial.write(b);
delay(10);
TinySerial.write(c);
delay(10);
TinySerial.write(d);
delay(500);
}
}
```

Include software serial for communication

Start Serial at 4800—same on 20M2

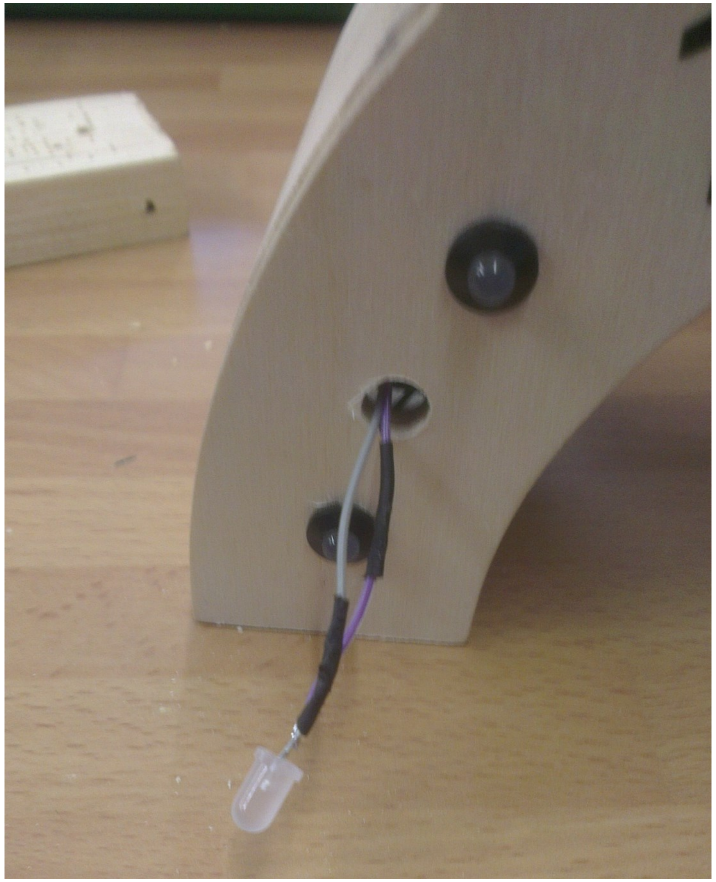
Pin 2 goes HIGH

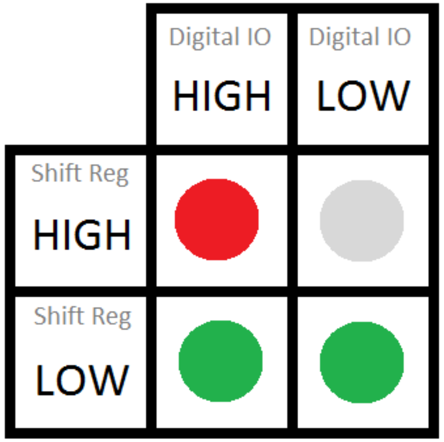
Change millis() into 2 word variables

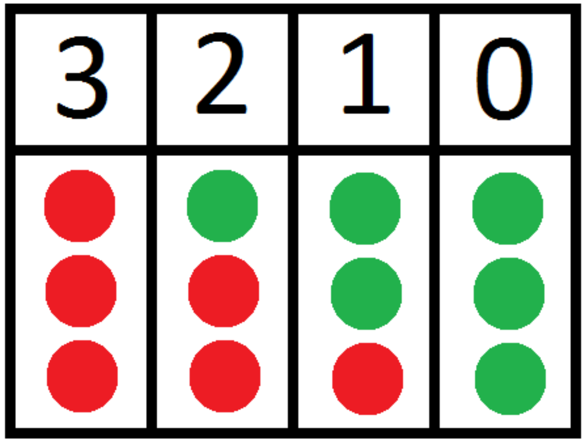
Split the 2 word variables into 4 bytes

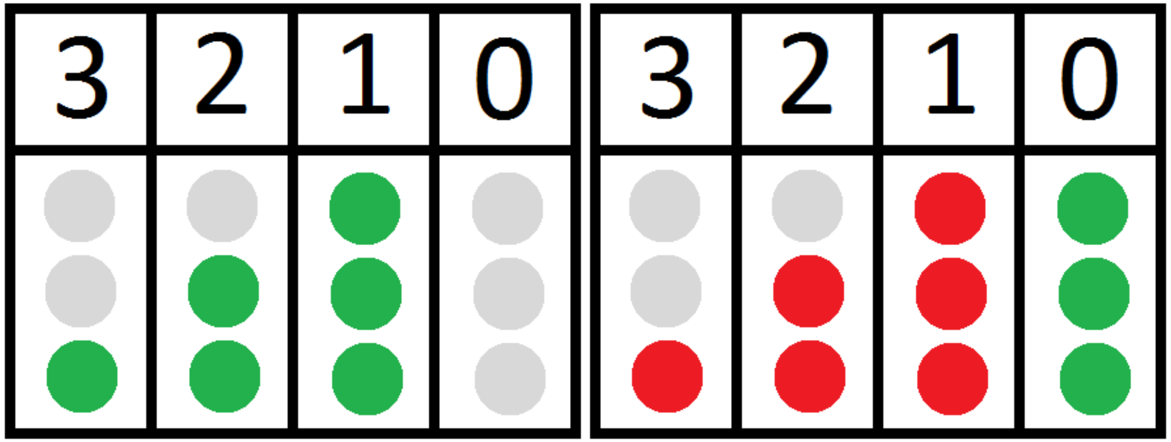
Send the 4 bytes of data out on the serial port

Bi-Colour LEDs



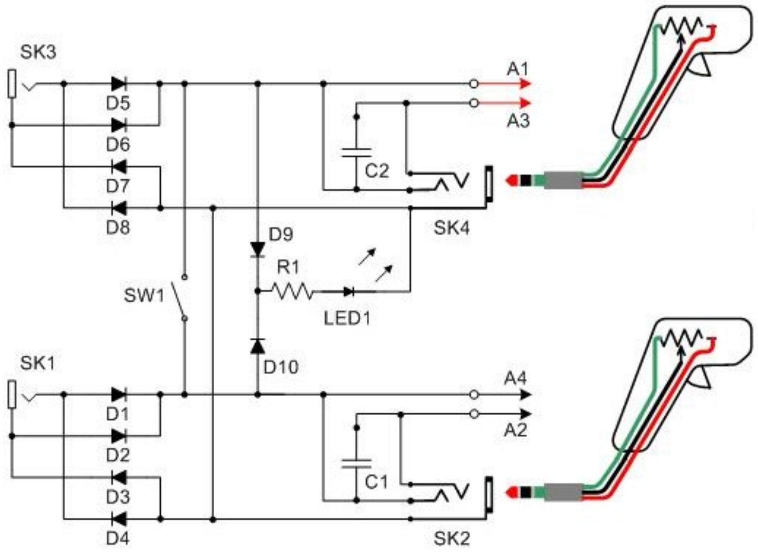






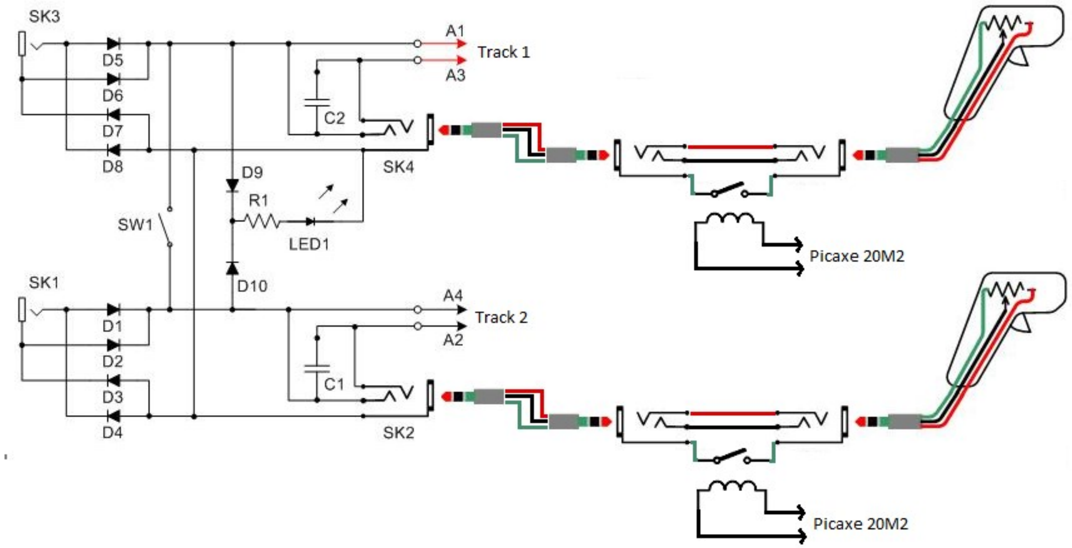
One problem that I faced whilst programming the start gate was the bi-colour LEDs not behaving as I had expected. Top left of the four tables is the 'truth table' for the LEDs using the digital IOs and shift register outputs. For my start sequence I had planned to do something along the lines of that in the bottom right table. However, from the truth table you can see that this is not possible with the current circuit design. Initially I tried switching the LEDs around, however this left me with exactly the same truth table, not solving my problem. I am unsure as to why this happens, although I think it may be a problem with the shift register not sinking current, only sourcing. Then I decided upon the start sequence demonstrated top right, however I have now decided that this is too confusing. It also leads to the LEDs changing brightness which does not look great. Consequently I have settled with the start sequence pictured bottom left. This involves the LEDs sequentially turning green; go is then signalled by all the LEDs turning off. To fully utilise the bi-colour functionality of the LEDs I plan to have a similar affair at the end of the race, however this time in red.

Scalextric Controllers

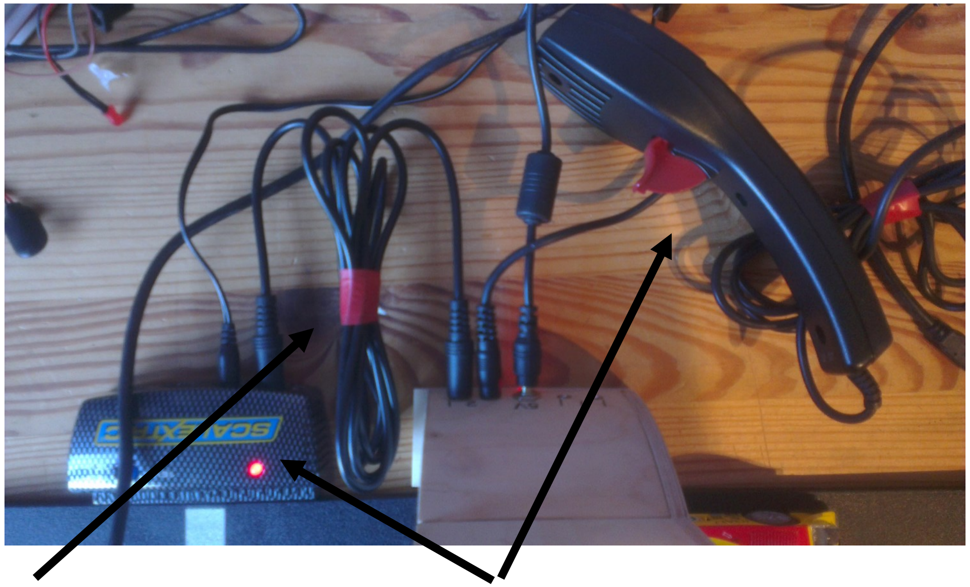


picprojects.org.uk - Original schematic

To enable the 20M2 to cut off the controllers preventing false starts, I used a relay for each controller. To interface this with the controllers I decided to use to 3.5mm headphones jacks; one has the controller plugged into it and the other has an aux cable plugged in to it and the scalextric power base. After studying the circuit diagram I realised I could break any of the three connections going to the controller (a linear potentiometer) to prevent the car from moving. I decided upon the wire that is grounded because if anything were to go wrong with my circuit, like a short circuit, then there would not be 16V on the loose. So using 4 3.5mm jacks I connected recreated the circuit diagram by adding jumper wires and long wires which would connect to the relay. I then tested both by plugging the controller into one and aux cable into the other, going to the scalextrix powerbase. Then I touched the tips of the two long wires together and the car moved. Disconnect the wires and the car didn't move.



Modified with relay between controller and powerbase



Aux cable connecting start gate to power base

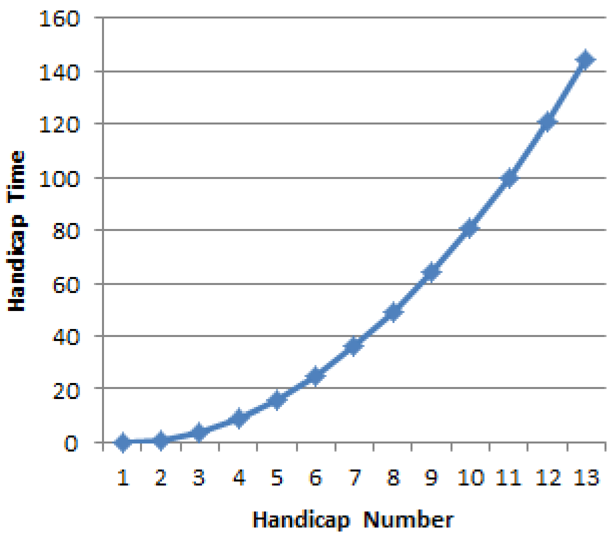
Original Scalextric power base and controller

Handicap System Coding

```
read user1,generaluse
displayvalue1=user1
user1=generaluse
read user2,generaluse
displayvalue2=user2
user2=generaluse

if user2>user1 then
user2 =user2-user1
user1 = 0
else if user2<user1 then
user1 = user1-user2
user2 = 0
else if user2=user1 then
user1=0
user2=0
endif

user1 = user1*user1
user2 = user2*user2
```

```
if laps1>laps2 then
read displayvalue1,generaluse
if generaluse<12 then
generaluse=generaluse+1
write displayvalue1,generaluse
endif
read displayvalue2,generaluse
if generaluse > 0 then
generaluse=generaluse-1
write displayvalue2,generaluse
endif

generaluse=1

else if laps1<laps2 then
read displayvalue2,generaluse
if generaluse<12 then
generaluse=generaluse+1
write displayvalue2,generaluse
endif
read displayvalue1,generaluse
if generaluse > 0 then
generaluse=generaluse-1
write displayvalue1,generaluse
endif
```

Calculating the handicap



The system is able to save 50 player profiles to the EEPROM, meaning the data is saved even when the chip is disconnected from the power. The number stored must be smaller than 255 and is increased after a win and reduced after a loss. In the piece of code to the left, the two numbers for each player are recovered from the EEPROM. The largest number, then has the smaller number subtracted from it, and then the smaller numbers is changed to zero. This means if the total time is 1 minute, the player with the worst handicap does 1 minute and then the better player does 1 minute minus there handicap time. This means the slowest player gets the whole race time, even if there handicap is higher than 0. After this function is applied to the handicap number, the new handicap number is squared for each player to calculate the amount of time that will be subtracted from the total time, for there overall time. This makes the handicap exponential which prevents players getting ridiculously high handicap numbers.

Changing the Handicap

Due to the limitations of the 20M2, I had run out of variables by this point; therefor I had to use the variable “displayvalue”, used for the seven segments in order to keep track of the user profiles. The code is very simple, it determines which user won, adds a handicap point to their profile and then removes a point from the losers profile. If a draw occurs, then neither of the profiles are changed.

Bug Fix

```
else if generaluse=1 then
var_out=%01111111
gosub update

high b.0'enable controllers
high b.1
tune C.1, 8, %11111111,($07)
gosub RecoverTime
Timer=W9

else if generaluse=10 then
high b.5
high b.6
```

Unfortunately, after the race has run and the program has gone back to the start it gets stuck at a digital read command. Initially, I thought it was a problem with the EEPROM write function, removing part of the code. However after commenting out sections of the code, I was able to track down the faulty piece of code by a process of elimination. It turned out to be a tune command which was setting all the digital pins high. After checking this in the “getting started” guide I was able to rectify the problem by setting all the bits to 0. It turns out the bits which I had changed were designed to flash LEDs in time with the tune; not change the pitch like I had thought.

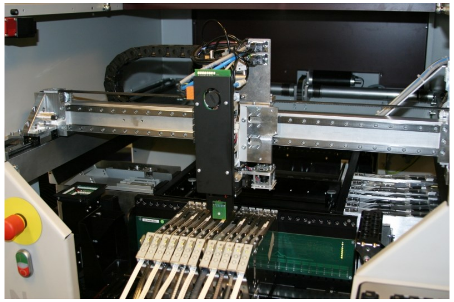
Syntax:

TUNE pin, speed, (note, note, note...)

TUNE pin, speed, LED_mask, (note, note, note...)

Industrial Processes

http://www.faulhaber.com/



http://www.dimagrp.com/

Circuit Board

The use of pick and place machinery could be used to mass produce the PCB for a much lower price than I could for a one of production. The use of SMD technology would also make the PCB smaller and therefor the casing could be more compact. However this would require a total redesign of the PCB, although the schematic would stay very much the same.

http://www.kavia.info/wp-content/uploads/arburg.jpg

Casing

In terms of mass production of the casing, injection moulding could be used to mass produce it. Although the initial setup cost is quite large, in the end it would work out cheaper. The initial set up cost is high because a mould has to be designed and machined out of metal. The case would have to be produced in 3 parts; firstly the main body of the case, also the top panel with cut-outs for the components and finally the back section.

Testing

Specification	Test	Evaluation
Be designed for use by 11 — 16 year olds	Ask 11—16 year olds to use the start gate	PASS - The majority of 11-16 year olds found my product easy to use
Be compatible with standard Scalextrics track and cars	Test it on a Scalextric track	PASS - The start gate works on a standard Scalextric track
Cost between £30 — £40 for the consumer to buy	Refer to the parts list	PASS - My product cost £37.05 to produce
Be aesthetically pleasing for the target market	Ask 11—16 year olds if they like the look of the start gate	PASS - On the whole, the target market liked the looks of the product
Be safe for use in school	Evaluate possible dangers in a risk assessment	PASS - Only a few possible risks which are very unlikely to occur
Be able to store 50 user profiles and appropriate handicaps	Test memory slots randomly between 1 and 50	PASS-All the EEPROM slots tested were able to store a handicap value
Use light and sound to signal the countdown sequence to both the cars and players to simulate a real racing experience	Does the start gate use light and sound to signal the start	PASS - The start gate does use light and sound
Feature an easily navigable menu to change options for the race	Ask people to use the start menu and ask for feedback	PASS - The overall opinion was that navigating the menu was easy
Use a form of display to show relevant information	Check the LCD whilst the start gate is in operation	PASS - Yes, the LCD displays relevant information
Be powered from the mains	Plug it into the mains, test it works	PASS - Works from the mains
Record laps up to and including 99	Set a long race time and try to reach over 99	PASS - Records up to 99 laps

Testing

Specification	Test	Evaluation
Record lap times with a maximum of 6500 seconds	Does variable responsible go that high in the code	PASS - Variable is capable of reaching up to 6025
Have a timer with accuracy of 0.01 seconds	What are the specifications of the real time clock	PASS/FAIL - Although timer has the accuracy it is not utilized
Be able to record times and laps for two lanes at once	Are both lanes of the Scalextrics track recognised	PASS - Both lanes are recognised by the start gate
Use a durable material and be a compact size	Check for any weak spots in the casing design and is it relatively compact compared to the Scalextrics track	PASS-No weak spots found and the start gate does not look out of proportion to the track
Prevent false starts	Try and start before go	PASS – Unable to start before the countdown had reached 0
Different race types such as fastest laps, most laps in a certain time, fastest lap and average lap	Is there an option for different races in the start menu	FAIL - There are no options regarding the type of race

Evaluation

Improvements which could be made

If I were to make a second version of this product, I would alter several things. Firstly, I would use a chip such as the 20X2 which is more powerful than the 20M2 and can store more variables and code. This would allow me to fix several problems which have occurred as a result of being limited by the electronic hardware. Firstly, I was unable to satisfy my final specification because I ran out of memory on the chip, and therefor could not program multiple game modes. If I had used a 20X2 then I would have twice the memory for storing the program and therefor would have been able to make the product better. Secondly, the clock speed of the chip let me down; it limited what could be done during the race in order to balance accuracy with functionality. For example I did not have time to update the LCD during the race because this would have lead to laps not being counted. If the clock had been running faster then this would have been less of a problem. Another annoying problem because of this is that the device only checks the time when a car passes the line. This is because if the two chips were in constant communication, virtually none of the laps would be counted. Consequently, the race is not the exact time specified. And finally, although my real time clock solution provided 1/1000th of a second accuracy, I was unable to utilize this because of the limited variable I could store on the 20M2. One problem that I would aim to try and solve in the future is the un-reliable nature of the communication between the 20M2 and the RTC chip. This causes races to be cut short some times or even go on forever! Currently I am unsure as to why the time is not always correctly transferred, but it is definitely something that could be improved. Finally, regarding the casing; next time I would spend far longer checking the 2D design of the panels to ensure they fit together correctly. I could have printed them on paper and cut them out to check the dimensions and then rectify any misalignments. A lot of my time was taken up by modifying the casing that was cut using the CNC machine and it meant that the quality was not the standard I was hoping for. The main problem was my last minute changing of materials leading to some thickness issues.

Solution to initial problem and Satisfaction of the brief

Task: Design and make a starting gate which counts down for the 'Scalextric' version of a Formula 1 grand prix. It can have both visual and audible outputs, and the timing sequence can be physically or remotely activated.

Brief: To design and build an electronic start gate, for use with any 'Scalextric' set. It will be simple to use and child friendly, but fun for all.

In conclusion; I have created a Scalextric start gate which counts down and starts a race like the task was. I have also satisfied my brief as the design is compatible with any Scalextric set, but it is also fairly simple to use and safe for use in school by children. Although there are still improvements which could be made, the product full fills the brief and task set.

Final Product