NEA Coursework Part 1

Design Problem
Too many accidents occur due to drivers accelerating too far over the speed limit on roads. My aim is to remove this fear for bystanders by implementing a wider range of more efficient speed measuring devices. This should act as a deterrent for people who are usually driving over the speed limit, and will mitigate the amount of speed tickets which are being given out to drivers.

Design Brief:
My system will utilise a beam breaking system, which will measure the time in which a vehicle travels from one point to another. This will then be turned into a speed which can be read by the PIC, and will trigger a police siren if the speed is too high.

Research:

https://www.rac.co.uk/drive/advice/cameras/speed-cameras/#:~:text=Speed%20cameras%20work%20by%20recording,the%20colour%20and%20vehicle%20type.

Specification:

> Use a 5V Power Supply

> When the first light beam is broken, latch the timer to an ON state within 1ms (5% Tolerance)

> When the beam is broken, latch the timer to an OFF state within 1ms (5% Tolerance)

> The time from the timer is stored to use for finding the speed

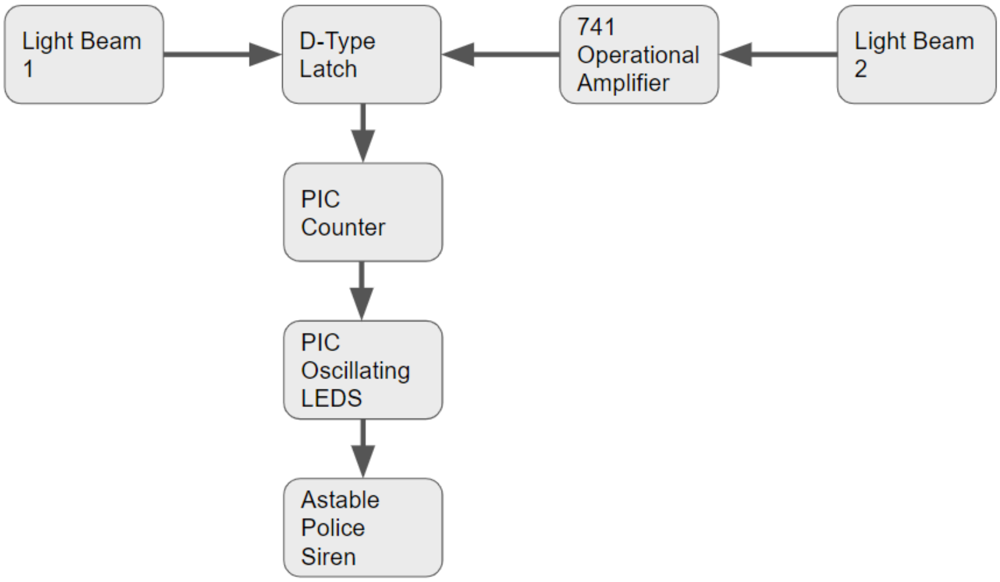
> The speed of the object is measured using the time (5% Tolerance in measurements)

> If speed is over 0.1 m/s then trigger an alarm

> Alarm sounds for 5s, followed by a 1s break, and repeats until reset button is pressed

Block Diagram:

Block Diagram



This block diagram shows how every circuit will interface and interact with one another, this is very helpful and important to visualise how my circuit as a whole will function.

NEA Coursework Part 2

Subsystem 1:

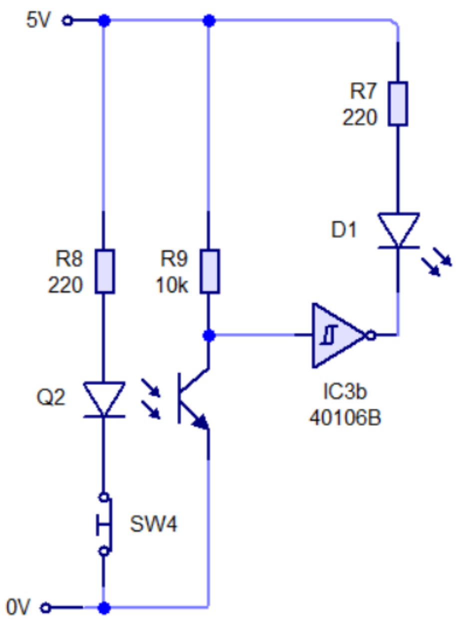
Infrared beam circuit

Specification:

> When the beam is broken, start the count
> The speed camera beam should trigger an output within 0.5s (5% tolerance)
> The speed camera beam must be wide enough to allow an object to pass
> Voltage must drop from 5V to <1V to Trigger

Circuit Design:

The circuit will consist of an Infrared LED and a phototransistor, when an object breaks the contact between these two components, the resistance of the phototransistor will drop and a signal will travel across a schmitt inverter which outputs a logic 1



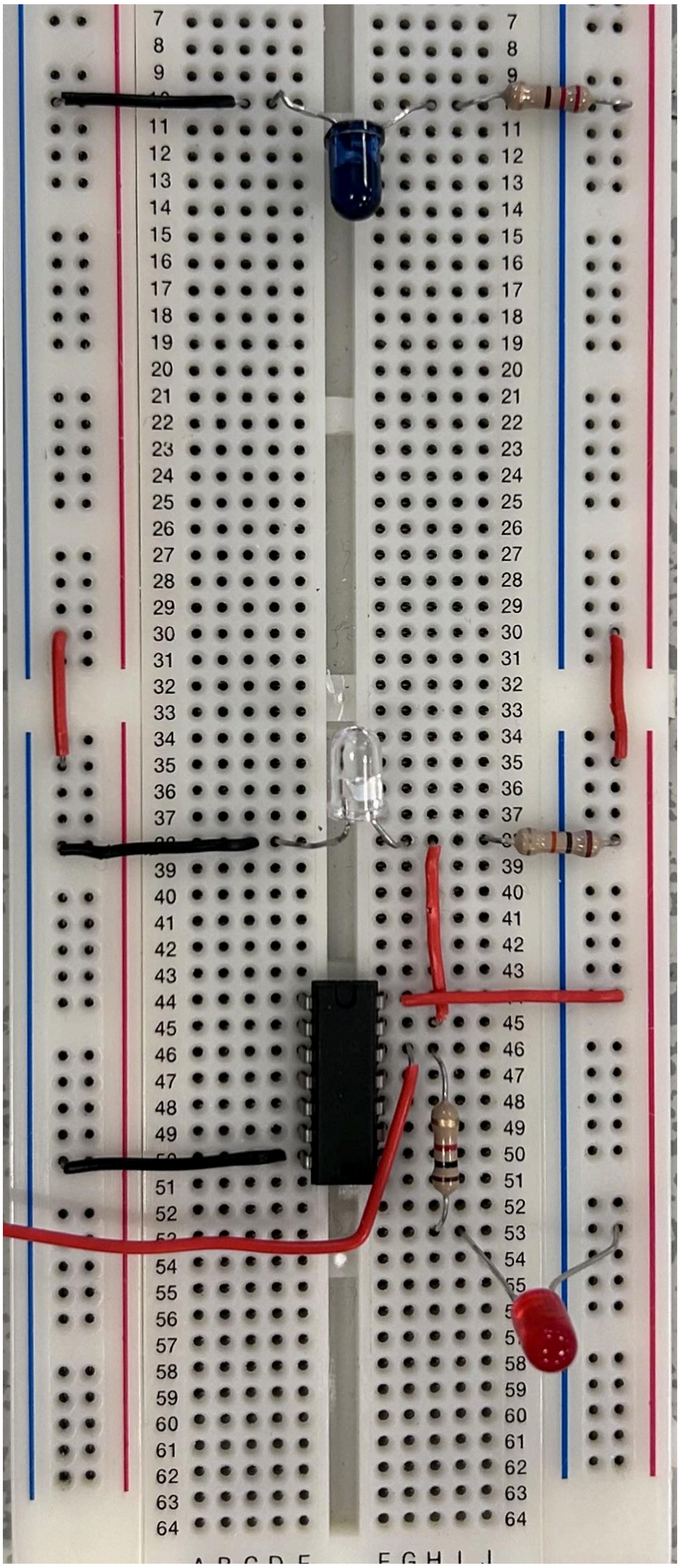
Alternative Circuit Design:

An alternative circuit design for this subsystem would be to use a MOSFET instead of a schmitt inverter. This will have the same output as the first circuit design but will be less efficient in building the circuit with less components.

Reasons for choice of subsystem Design

Using a schmitt inverter will give a wider range for leniency and tolerances due to the voltage in which it switches.
As I will be using schmitt not gates elsewhere in my system, it is more efficient to use this as it will reduce the amount of chips needed overall.

Photo of Subsystem



There is an infrared LED and a phototransistor facing each other which acts as the beam in this subsystem. When the beam is covered, the voltage drops, if this reaches the schmitt threshold, the red LED will turn on and provide an output.
In the photos above, I have shown that the test LED is turned off when not covered, and the LED turns on when the beam between the components is broken.

Test Procedures and Equipment Used

To test this infrared beam circuit, I had implemented a test LED to confirm that the beam itself was triggering the correct output when broken, this could also be timed using a stopwatch to confirm that the output should be triggered within the 0.5 seconds specified.

I have also used a multimeter to measure the voltage in which the schmitt inverter triggers, this should show the voltage that then will be used for the inputs of the next subsystem.

Test Results

When the LED is covered, the voltage drops from 5V to 0.5V

When the LED is covered, the output is triggered in the time shown below. This shows that the beam is triggered within the 0.5s threshold.

When the beam is covered the voltage drops, which produces a logic 1 through the schmitt inverter and triggers the LED which I have placed into the circuit.

Comparison of results against specification
Compared to the specification, I have found that the voltage drop is approximately what I was looking for and this will help when leading into the next subsystems in the overall project.

The system also activated in under 0.5 seconds, which also met the specification as the cars will be moving fast when being detected and the beam must be triggered fast with no errors.

Evaluation of Subsystem.

Overall, the subsystem works as expected, when the beam is broken, the output, which is originally low, turns high due to the schmitt trigger before the output. This provides for a clean switch between the high and low states. After using an LED to test the functionality of the circuit, I have found that it works quickly and efficiently.

The gap in between the infrared emitter and the photo-transistor is wide enough for an object to pass, which is needed for my speed camera functionality.

When tested, I found that the circuit worked as stated in the specification. The beam is working as intended and triggers an output signal in a short amount of time which is especially needed for my subsystem.

No changes were needed to be made to the system, and the overall quality of the system is high. All wires were clearly built to show the voltages of all inputs.

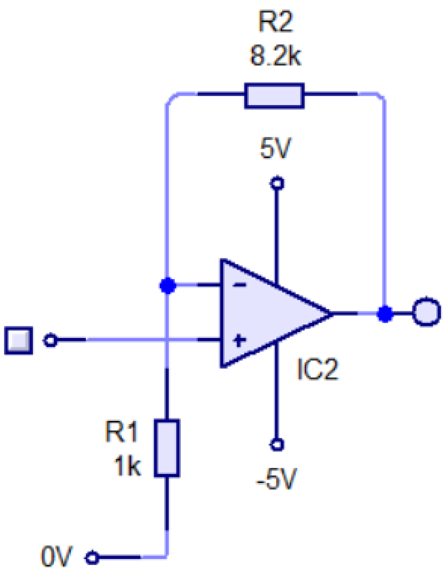
Subsystem 2: Operational Amplifier

Specification:

> When receiving an input from the speed camera beam, amplify the voltage to 5V

> Ensure that later subsystems receive a sufficient voltage

Circuit Design:



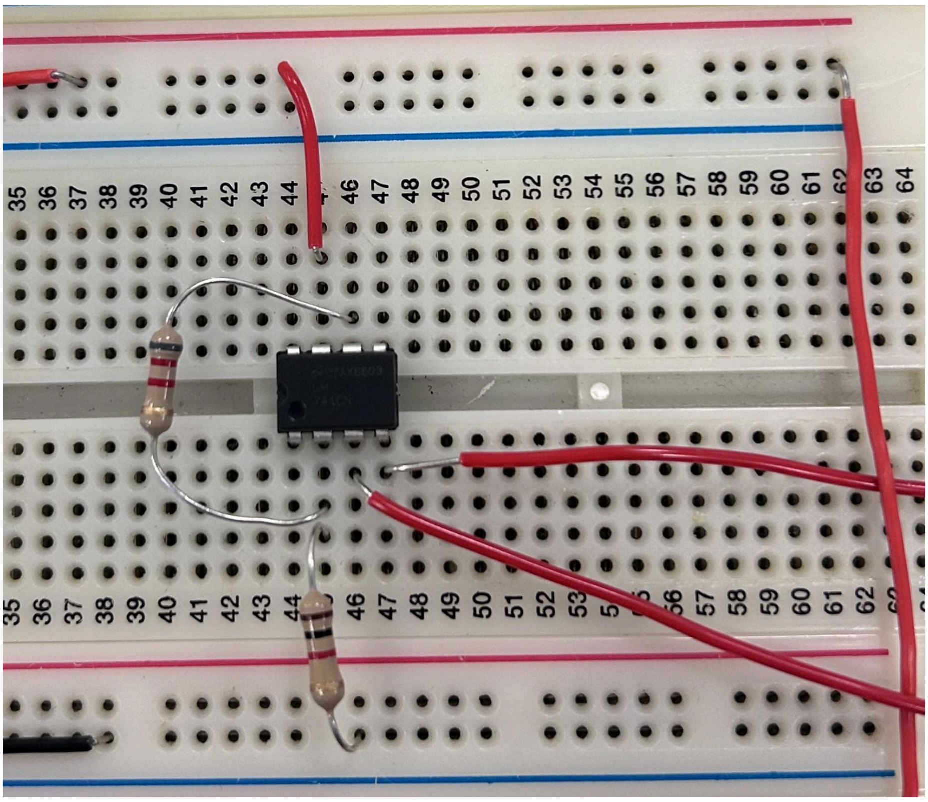
This circuit utilises a Non-Inverting 741 Operational Amplifier along with a 9k ohm and a 1k ohm resistor in place to give a gain of 10.
This should raise the voltage from the output of the break the beam circuit from 0.5V to 5V which is the normal voltage for my system.

Reasons for choice of subsystem Design

Due to this subsystem utilising two phototransistor beam circuits, the beam for ending the speed camera timer will be operated from a second beam which is located much further away from the main circuit.

This subsystem needs to take the 0.5V from the output of the beam system and amplify this in order to act as a sufficient input to trigger my other subsystems such as the PIC Counter. Due to the input voltage being originally very low, a higher voltage amount is needed for my PIC system to accept an input voltage.

Photo of Subsystem



Test Procedures and Equipment Used

Using a multimeter, I was able to test the before and after voltages of the system, and have used the gain calculation to figure out the correct resistor amounts which are needed. These tests have shown that the output of the second break the beam circuit, has been amplified to a voltage close to 5V, this means that the rest of the circuit can run from 5V.

Test Results


3.06
AUTO POWER OFF
ON/OFF
PNP
NPN
hFE
TEMP
Cx
10A
mA
COM
VΩHz
20A 15SEC MAX
UNFUSED
200mAMAX
FUSED
Rapid


0.00
AUTO POWER OFF
ON/OFF
PNP
NPN
hFE
TEMP
Cx
10A
mA
COM
VΩHz
20A 15SEC MAX
UNFUSED
200mAMAX
FUSED
Rapid

Comparison of results against specification

The subsystem does not directly amplify the voltage to 5, but does amplify it to a sufficient amount.
The voltage amplified is enough to power the system as a whole, which was needed in order to keep the speed camera circuit as a whole functional.
The break the beam system dropped to a voltage of 3.06V which may not be enough to power the rest of the system, this is why we amplified this to 5V to ensure that the system is functioning as needed.

Evaluation of Subsystem

Overall, the subsystem works as expected and provides the outputs needed given the initial inputs from the other subsystems.
Using 5V into the positive and negative Op-Amp voltage inputs, this means that the voltage will not exceed 5V and be too much for the circuit to function.

When tested, I found that the circuit does not reach 5V exactly, but saturates before that, this could be to do with the resistor values, yet it does not need to be edited due to the voltage being sufficient enough to function fully.

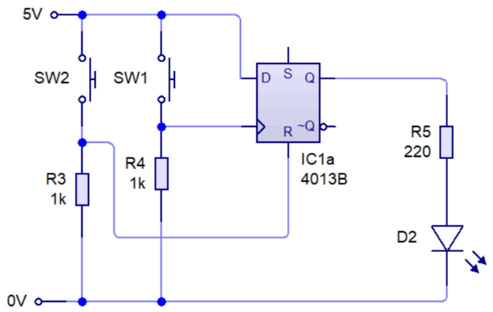
The overall quality of this system is perfect and the only improvement which could be made is changing the resistor values to reach 5V.

Subsystem 3: D-Type Latch

Specification:

> When a car passes through the first beam, latch onto being always on
> When a car passes through the second beam, latch onto being always off

Circuit Design:

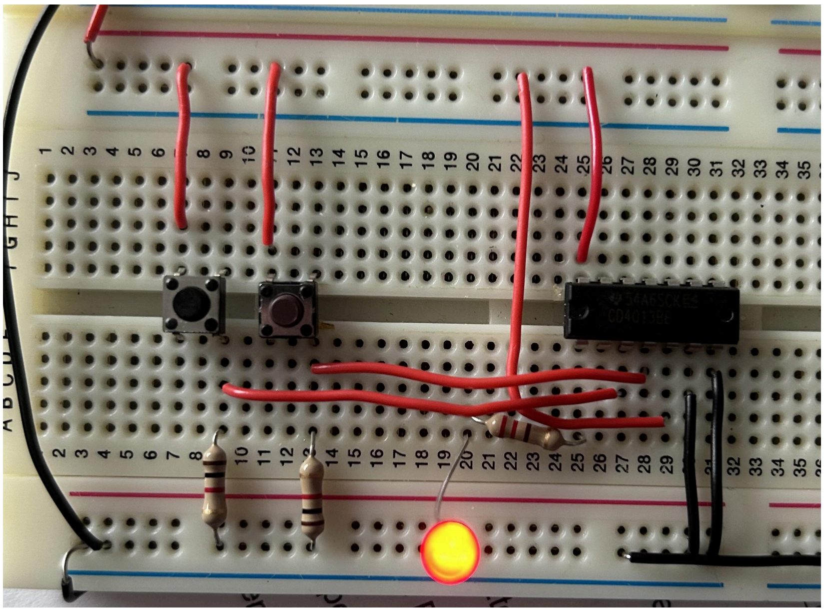


This circuit utilises a 4013 Chip which is a D-Type flip flop chip.
When receiving an input from the first break the beam system, it will toggle the latch onto an always on state.
When receiving an input from the second break the beam system, it will toggle the latch onto an always off state.
This will keep the counter for the speed camera system on using the other subsystems involved with the system as a whole.

Reasons for choice of subsystem Design

The break the beam system only provides a small trigger of an input to any other subsystems, which means that a subsystem was needed to be able to carry the input and keep it at a logic 1 in the system. Allowing the beam to trigger this latch means that the short input can be made into a long pulse for other subsystems.

Photo of Subsystem



Test Procedures and Equipment Used

To test the voltage changing and latching, I used a multimeter to record the voltages of the latching system and the de-latching system.
These provided results of the voltage spiking from 0V to 5V after blocking the infrared beam, which then activated the latch.
When blocking the second beam however, the voltage then dropped from 5V to 0V, to show that the latch has been turned off.

Test Results


0.00
20
AUTO POWER OFF
26
ON/OFF
PNP
NPN
hFE
TEMP
10A
mA
COM
VΩHz
20A 15SEC MAX
UNFUSED
200mA MAX
FUSED
Plug P
Rapid

Comparison of results against specification

In the specification, the two break the beam systems linked into the triggering system of this latch.
These worked very well as the voltages did exactly as they needed to, to provide a stable input into the next subsystem.
The voltage also was quick to drop from 5V to 0V, which means that the system will not be latched on, for any more time than what is needed to measure the speed of the item going past the beams.

This should mean that no false records are measured by my system.

Evaluation of Subsystem

This system as a whole works as needed and intended. The system provides sharp changes between the voltages 0V and 4.2V which is the saturation voltage of the Op-Amp before the latching system.

This system takes in the correct inputs of the break the beam subsystems, which only provide a small pulse of current into the latch, yet the latch takes this and sets the logic and output to 1 or 5V.

This system also only utilises one single chip, which saves on the overall product list, meaning this subsystem is simple to create and has a huge role in the speed camera system overall.

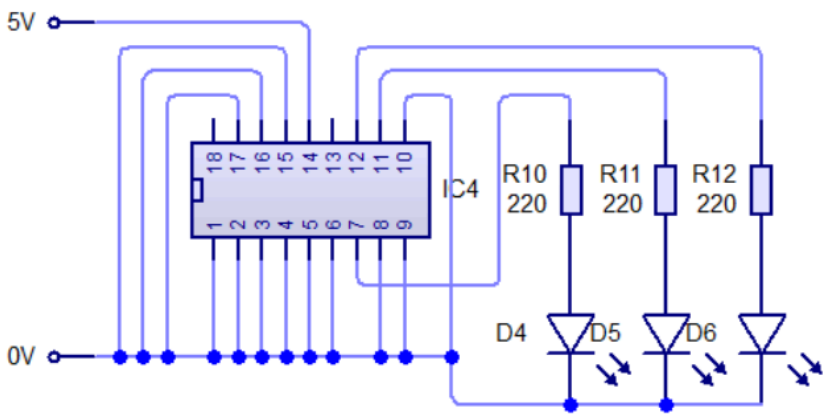
The build quality of the system is high and utilises red and black wires to determine where the positive voltage and no voltage is connecting which also plays a role in the testing of my subsystem for if any flaws become prevalent, which is not likely due to the build quality.

Subsystem 4: PIC Counter

Specification:

> When car passes the first beam, start the count
> When a car passes the second beam, stop the count
> Stop the count quickly to minimise errors

Circuit Design:



The circuit itself houses two subsystems as it has been made using a PIC microcontroller.
The microcontroller chip specifically is a PIC16F88 which comes with a compiler translating software.
The circuit design has been made this way in order to avoid any wastage of materials and to keep the overall cost down yet keeping the build quality high.
The circuit was originally designed using assembly code, then was converted into a HEX file which was read as machine code onto the 16 bit microcontroller.

Alternative Circuit Design:

A D-Type counter could be used instead of my PIC as these are also a viable option when it comes to counting up or down.
The counter in my PIC utilises 8 bits to count down from 255, which means that 8 D-Type counters will be needed in order to keep my circuit running efficiently.

Reasons for choice of subsystem Design

The reason I chose to use a PIC microcontroller for my subsystem design is because it was the best option to limit waste of materials and money.
A PIC only uses one chip and can count up to 8 bits, which is very necessary for my whole system.
Using the alternative circuit would waste too many supplies.

Also, the count on my sub system must work very specific counting down 1 increment every 40ms, which 255 x 40ms = 10.2s
This is easier to code on a PIC then it is to use components to adjust the timing on the counter.

Photo of Subsystem

Photos of code:

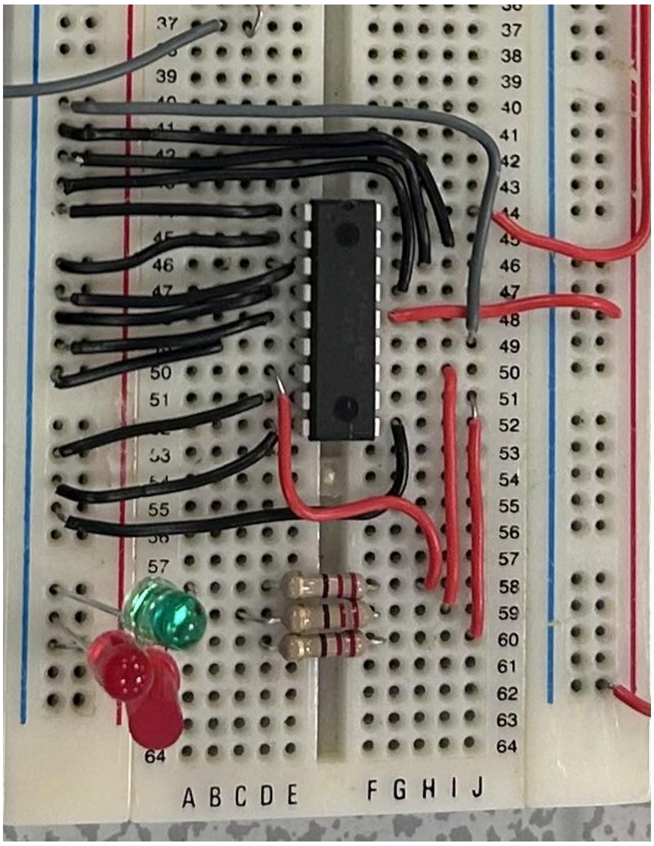
```
;************************************************************************
; DEFINITIONS
;************************************************************************
    list    p=16F88             ; tells the assembler which PIC chip to program for
    radix   dec                 ; set default number radix to decimal
    ;radix  hex                 ; uncomment this to set radix to hex
    __config h'2007', 0x3F50    ; internal oscillator, RA5 as i/o, wdt off
    __config h'2008', 0x3FFF
    errorlevel -302             ; hide page warnings
W EQU h'00'                     ; pointer to Working register
F EQU h'01'                     ; pointer to file
;****** REGISTER USAGE ******

; Register page 1
TRISA       EQU h'85'           ; data direction registers
TRISB       EQU h'86'
OSCCON      EQU h'8F'           ; internal oscillator speed
ANSEL       EQU h'9B'           ; ADC port enable bits

; Register page 0
STATUS      EQU h'03'           ; status
PORTA       EQU h'05'           ; input / output ports
PORTB       EQU h'06'
INTCON      EQU h'0B'           ; interrupt control
ADRESH      EQU h'1E'           ; ADC result
ADCON0      EQU h'1F'           ; ADC control
COUNTDOWN   EQU h'20'           ; general use byte registers B0 to B27
LIGHTS      EQU h'21'
B2          EQU h'22'
B3          EQU h'23'
B4          EQU h'24'
B5          EQU h'25'
B6          EQU h'26'
B7          EQU h'27'
B8          EQU h'28'
B9          EQU h'29'
B10         EQU h'2A'
B11         EQU h'2B'
B12         EQU h'2C'
B13         EQU h'2D'
B14         EQU h'2E'
B15         EQU h'2F'
B16         EQU h'30'
B17         EQU h'31'
B18         EQU h'32'
B19         EQU h'33'
B20         EQU h'34'           ; used in interrupt routine
B21         EQU h'35'           ; used in interrupt routine
B22         EQU h'36'
B23         EQU h'37'
B24         EQU h'38'
B25         EQU h'39'
B26         EQU h'3A'
WAIT40      EQU h'3B'
WAIT1       EQU h'3C'           ; counters used in wait delays
WAIT10      EQU h'3D'
WAIT100     EQU h'3E'
WAIT1000    EQU h'3F'
ADCTEMP     EQU h'40'           ; adc loop counter

;****** REGISTER BITS ******
```

```
;****** REGISTER BITS ******

C           EQU h'00'           ; carry flag
Z           EQU h'02'           ; zero flag
RP0         EQU h'05'           ; register page bit
INT0IF      EQU h'01'           ; interrupt 0 flag
INT0IE      EQU h'04'           ; interrupt 0 enable
GIE         EQU h'07'           ; global interrupt enable

;**********************************************************************
; VECTORS
;**********************************************************************
;The PIC16F88 reset vectors
ORG         h'00'               ; reset vector address
            goto start          ; goes to first instruction on reset/power-up
ORG         h'04'               ; interrupt vector address
            goto interrupt
;

;**********************************************************************
; SUBROUTINES
;**********************************************************************
; Predefined wait subroutines - wait1ms, wait10ms, wait100ms, wait1000ms
wait1ms                         ; (199 x 5) + 5 instructions = 1000us = 1ms @ 4MHz resonator
            movlw d'199'        ; 1
            movwf WAIT1         ; 1
loop5ns
            clrwdt              ; 1 this loop 1+1+1+2 = 5 instructions
            nop                 ; 1
            decfsz WAIT1,F      ; 1
            goto loop5ns        ; 2
            nop                 ; 1
            return              ; 2
wait10ms
            movlw d'10'         ; 10 x 1ms = 10ms
            movwf WAIT10
loop10ms
            call wait1ms
            decfsz WAIT10,F
            goto loop10ms
            return
wait100ms
            movlw d'100'        ; 100 x 1ms = 100ms
            movwf WAIT100
loop100ms
            call wait1ms
            decfsz WAIT100,F
            goto loop100ms
            return
wait1000ms
            movlw d'10'         ; 10 x 100ms = 1000ms
            movwf WAIT1000
loop1000ms
            call wait100ms
            decfsz WAIT1000,F
            goto loop1000ms
wait40ms
            movlw d'40'         ; 40 x 1ms = 40ms
            movwf WAIT40
loop40ms
            call wait1ms
            decfsz WAIT40,F
            goto loop40ms
return
```

```
; MAIN PROGRAM
;**********************************************************************
;****** INITIALISATION ******
start
            bsf STATUS,RP0      ; select register page 1
            movlw b'01100000'   ; set to 4MHz internal operation
            movwf OSCCON
            clrf ANSEL          ; disable ADC (enabled at power-up)
            bcf STATUS,RP0      ; select register page 0
;the data direction registers TRISA and TRISB live in the special register set. A '1' in
;these registers sets the corresponding port line to an Input, and a
;'0' makes the corresponding line an output.
Init
            clrf PORTA          ; make sure PORTA output latches are low
            clrf PORTB          ; make sure PORTB output latches are low
            bsf STATUS,RP0      ; select register page 1
            movlw b'11111111'   ; set port A data direction (0 = output bit, 1 =
                                ; input bit)
            movwf TRISA ;
            movlw b'00000000'   ; set port B data direction (0 = output bit, 1 =
                                ; input bit)
            movwf TRISB         ;
            bcf STATUS,RP0      ; select register page 0

;****** PROGRAM ******
;************* remove semicolons from next two lines to enable interrupt routine************
; When PORTA, 1 is held, start a decrementing counter of 255
; if PORTA, 1 is let go, the vehicle moving is too fast as it has not let the timer decrease to 0
; flicker lights PORTB, 3 & 4 to show the count is decrementing
; if too fast, oscillate a red and blue LED, PORTB, 5 & 6 fifteen times
; if count decrements to 0, goto normal and flash PORTB, 1 three times


main
        bsf INTCON,INT0IE   ; set external interrupt enable
        bsf INTCON,GIE      ; enable all interrupts

                            ; Wait for the input at PORTA, 1
        btfss PORTA, 1      ; Check if PORTA, 1 is set
        goto main           ; Keep waiting if PORTA, 1 is not set
        goto countstart

                            ; Input detected, start the counter
countstart
        bsf PORTB, 0        ; Test LED to show that input porta, 1 was accepted
        movlw d'255'        ; Add 255 loops
        movwf COUNTDOWN     ; Named COUNTDOWN
count
        bcf PORTB, 0        ; Clear the test LED
        bsf PORTB, 3        ; Blink portb, 3 every time the count decrements
        call wait40ms       ; 1m distance, 255 loops at 40ms = approx 10 seconds
        btfss PORTA, 1      ; If port a, 1 is set then go to toofast
        goto toofast
        decfsz COUNTDOWN,F  ; If the counter reaches 0, then goto normal
        goto count          ; If counter is not 0, then repeat the count
        goto normal
```

```
normal
        bcf PORTB, 4
        bcf PORTB, 3
        bsf PORTB, 1        ; flash PORTB, 1 three times,, will be connected to a green led
        call wait1000ms
        bcf PORTB, 1
        call wait1000ms
        bsf PORTB, 1
        call wait1000ms
        bcf PORTB, 1
        call wait1000ms
        bsf PORTB, 1
        call wait1000ms
        bcf PORTB, 1
        goto endlights

endlights
        clrf PORTB          ; Clear Port B
        clrf PORTA          ; Clear Port A
        goto main
;**********************************************************************
; INTERRUPT SERVICE ROUTINE
;**********************************************************************

W_SAVE      EQU B2              ; backup registers used in interrupts
interrupt
            movwf W_SAVE        ; Copy W to save register
            btfss INTCON,INT0IF ; check correct interrupt has occurred
            retfie              ; no, so return and re-enable GIE

;*********The interrupt service routine (if required) goes here*********

            bcf INTCON,INT0IF   ; clear interrupt flag
            movf W_SAVE,W       ; restore W
            retfie              ; return and re-set GIE bit
            END                 ; all programs must end with this
```

Photo of Physical Circuit.

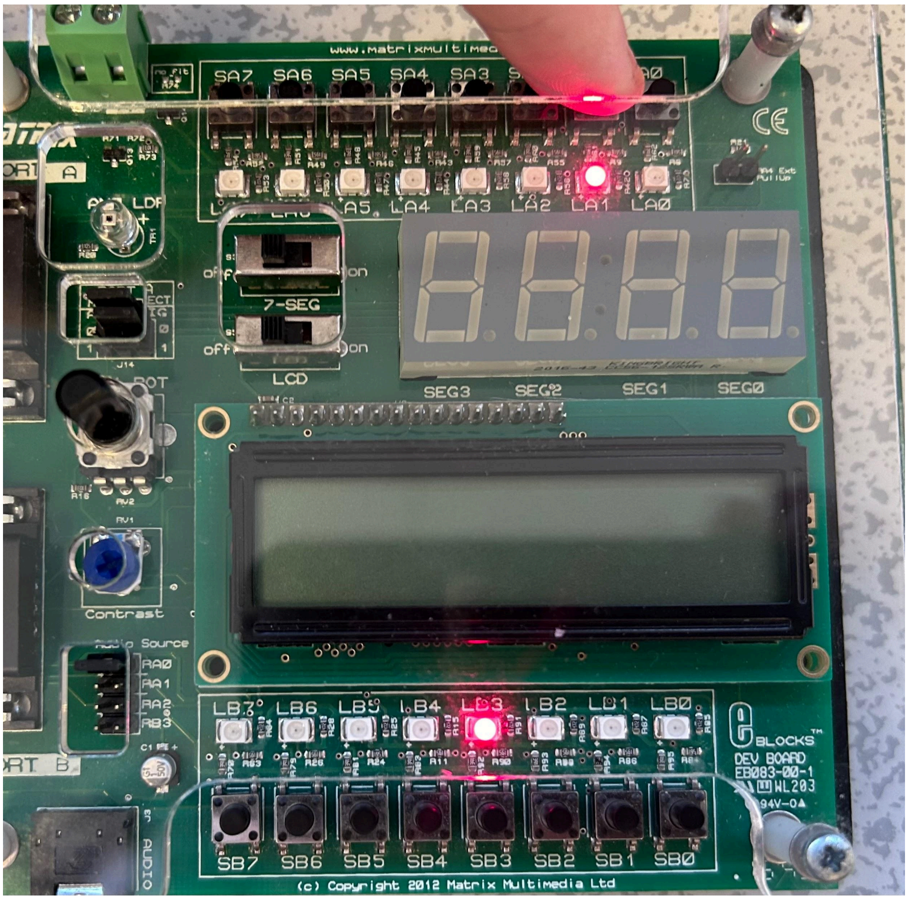


Test Procedures and Equipment Used

To test if the counter was working as intended, I altered my code to include a flashing light on PORTB, 3 every time the counter incremented. I then added the PIC microcontroller into the coding board where I can simulate the inputs and outputs. When the code was executed, I was able to see the flashing lights on the PIC which showed me that the count was working as intended.

Test Results

The test results show that the counter has been incrementing every 40ms as the counter stops at roughly 10.2 seconds.
The tests also show that the outputs for the count were working as the LEDs were lit up for every count, and these two outputs were in the same block of code / loop.
These test results show that the system is working as intended



Comparison of results against specification

The specification states that the count is to start counting down when the first beam is passed. This function both works with the break the beam system as well as the test buttons which were implemented in the coding board when developing the system.

Evaluation of Subsystem

Overall, the system works well and the build quality is perfect for the subsystem.
The wires are clearly divided through red and black wiring, and no wires are crossing components.

This counter works as needed and counts to 10.2 seconds in 40ms increments and this has been tested, giving results which I am very happy with.

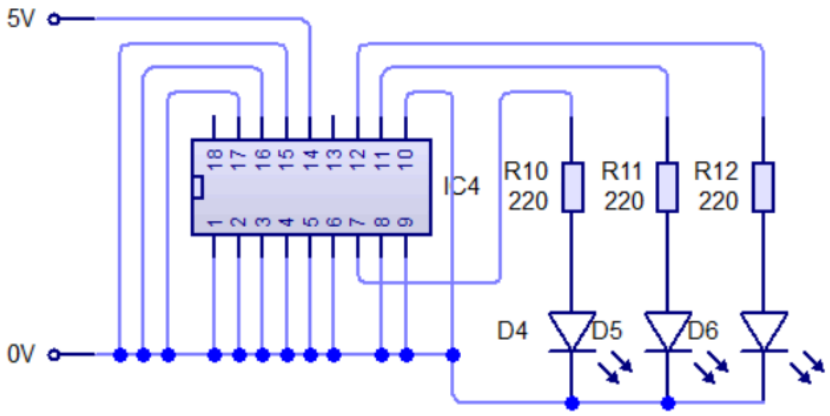
One way which I could improve this is maybe by adding different speeds which they increment at, meaning I can change the speed required to set off the police siren and lights.

Subsystem 5: PIC Oscillating LEDs

Specification:

> If the car is moving too fast, turn on the red LEDs one at a time to mimic police lights
> If the car isn't moving too fast, then turn on the green LED
> Green LED should turn on after 10.2s (2% tolerance)

Circuit Design:



This subsystem links in with the PIC counter as they work together in order to display the LEDs

If the second beam is passed after the counter reaches 0, the green LED lights up and flashes 3 times to indicate that the car is not moving faster than the designated speed. However, if the second beam system is passed before the counter reaches 0, then the red oscillating LEDs turn on and indicate that the car is moving too fast.

These LEDs are bright and display vivid colours to warn the police if the person driving is going too fast.

Alternative Circuit Design:

Oscillating LEDs using an astable may have been another good choice for making my subsystem. This would utilise a 555 timer to create an adjustable astable to create a siren-like LED display.

Reasons for choice of subsystem Design

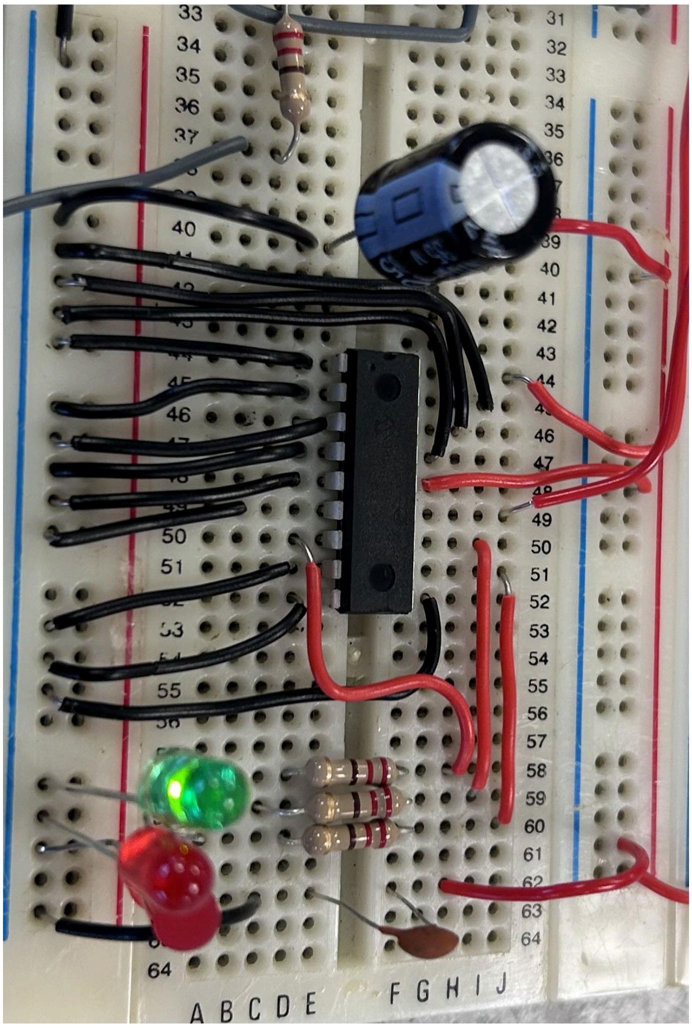
Using a PIC however reduces the overall product cost and keeps the amount of chips being used down, as the PIC creates the oscillation through loops created in the assembly code.

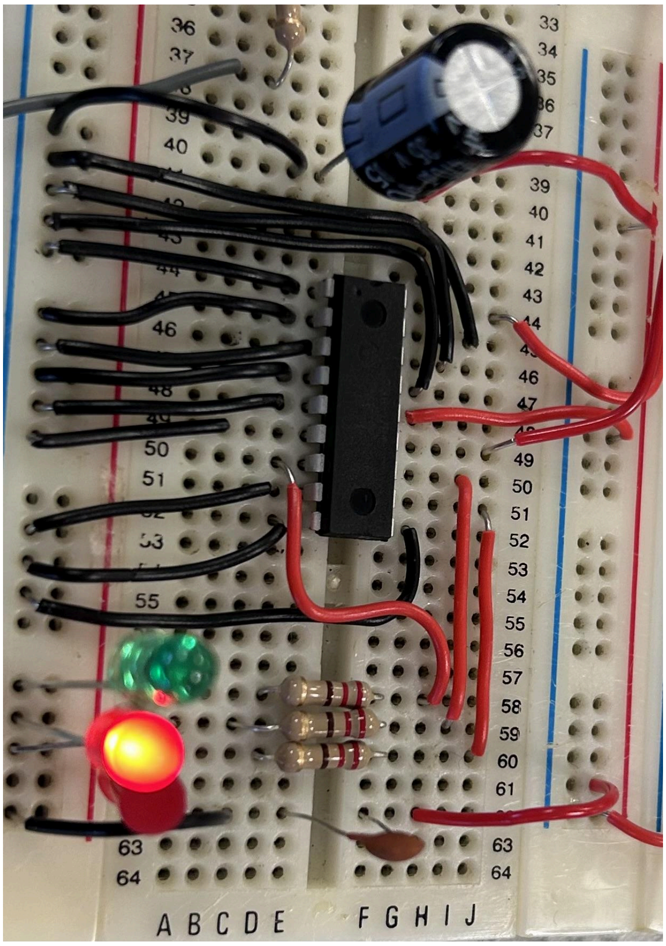
Photo of Subsystem

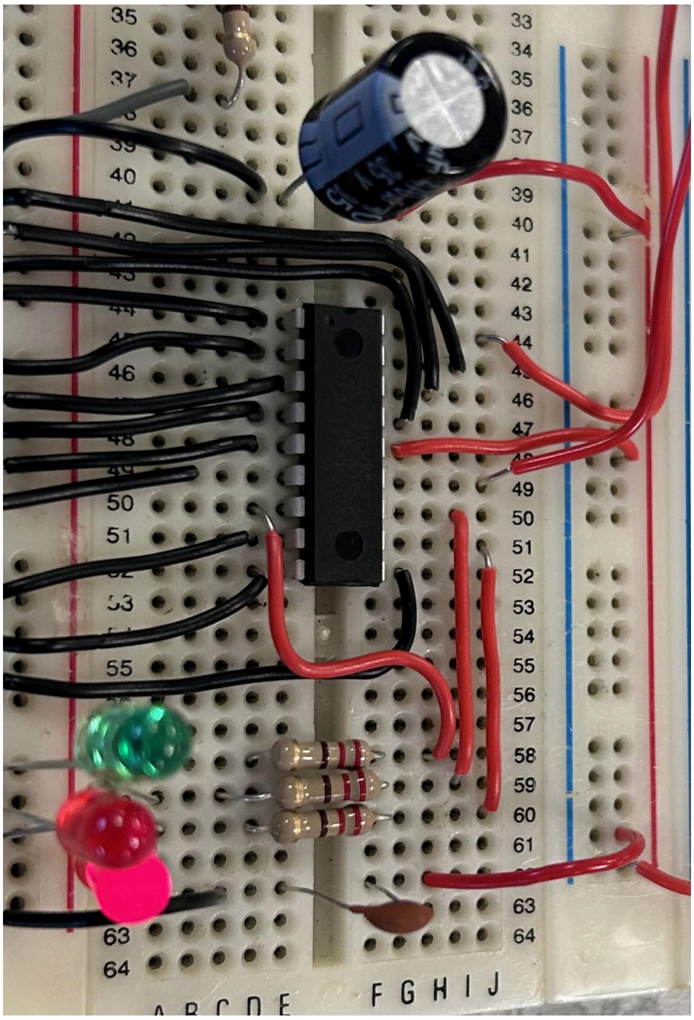
Photo of code:

```
toofast
        movlw d'15'         ; Load 15 loops into working regsiter
        movwf LIGHTS        ; Named LIGHTS
lightstart
        bcf PORTB, 3
        bsf PORTB, 7        ; This output will trigger the oscillating police siren
        bcf PORTB, 6        ; Create an oscillating light, to go along with the hardware siren
        bsf PORTB, 5
        call wait100ms      ; Wait 1s
        call wait100ms
        call wait100ms
        call wait100ms
        call wait100ms
        bcf PORTB, 5
        bsf PORTB, 6
        call wait100ms
        call wait100ms
        call wait100ms
        call wait100ms
        call wait100ms
        decfsz LIGHTS,F     ; If loop countdown reaches 0 then goto endlights
        goto lightstart     ; If the counter is not 0, then loop again
        goto endlights
```

Photo of physical circuit:


A B C D E
F G H I J


A B C D E
F G H I J



Test Procedures and Equipment Used

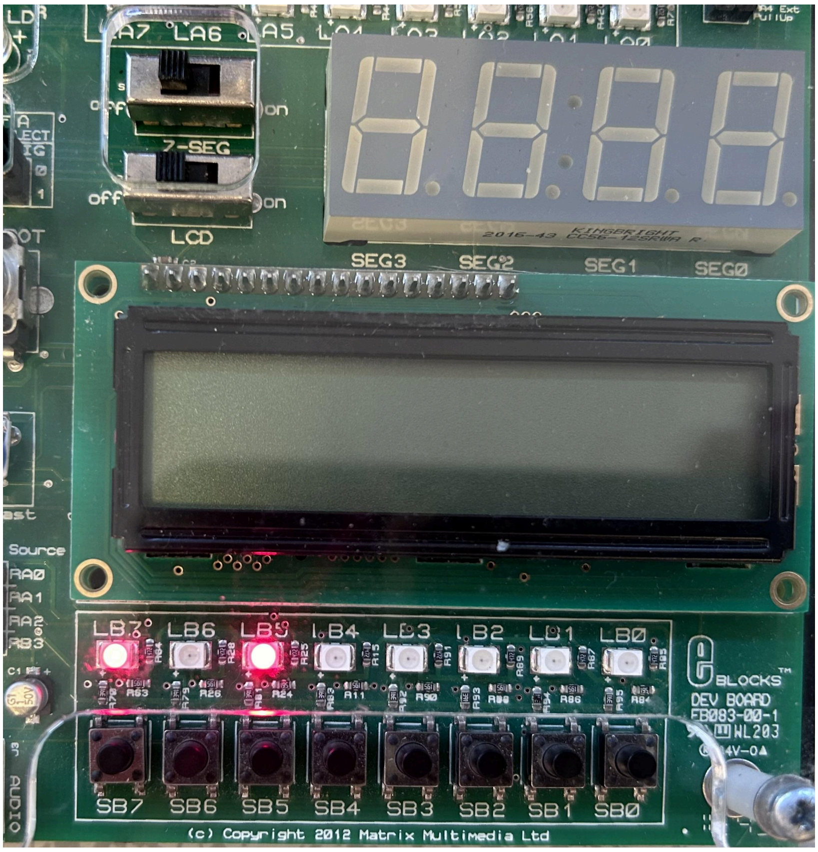
Using the PIC microcontroller board I was able to test all of the inputs and outputs, as well as to see if any errors made themselves apparent.

Using this, I was able to see that the outputs of portb,5 and 6 were both functioning properly as the siren outputs, as well as the output for the green LED.

We can also see here that the Green LED turns on after 10.4s yet this is through human testing which also accounts for reaction times. (Human error)

Test Results


RESET
START/STOP
START/STOP
S·A·T·Z
H M S 1/100
000 10 42
SCHOOL
EVENT TIMER



Comparison of results against specification

The specification has been met for both of the points, this is because when the gates are passed too quickly, the red LED sirens start to display, this is exactly as what was needed in my specification.
However if the second beam is passed after the previous subsystem counter ends, then the green LED flashes which is also a necessary part of my specification which has been met.

Evaluation of Subsystem

Overall, my subsystem functions very well and the wires are clearly labelled red or black to denote positive volts or 0 volts, this helped with testing to see where the inputs were working or not.

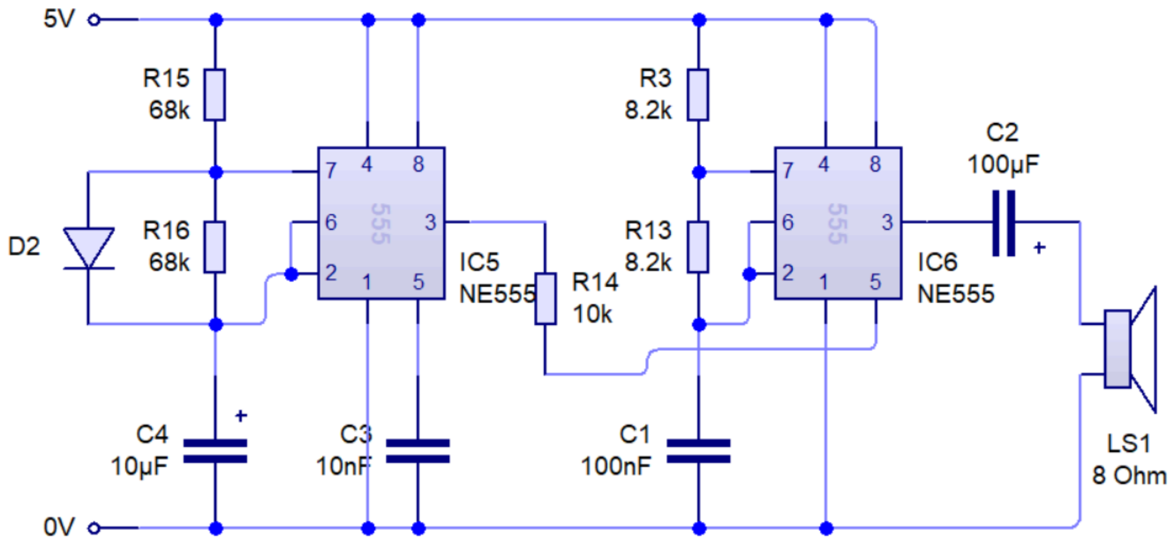
The system also matches the specification well which was much needed as this means the subsystem works as intended and there are no large issues which may hinder the progress of the circuit building.

Subsystem 6: Oscillating Police Siren

Specification:

> Sound an alarm if the car is moving too fast.

Circuit Design:

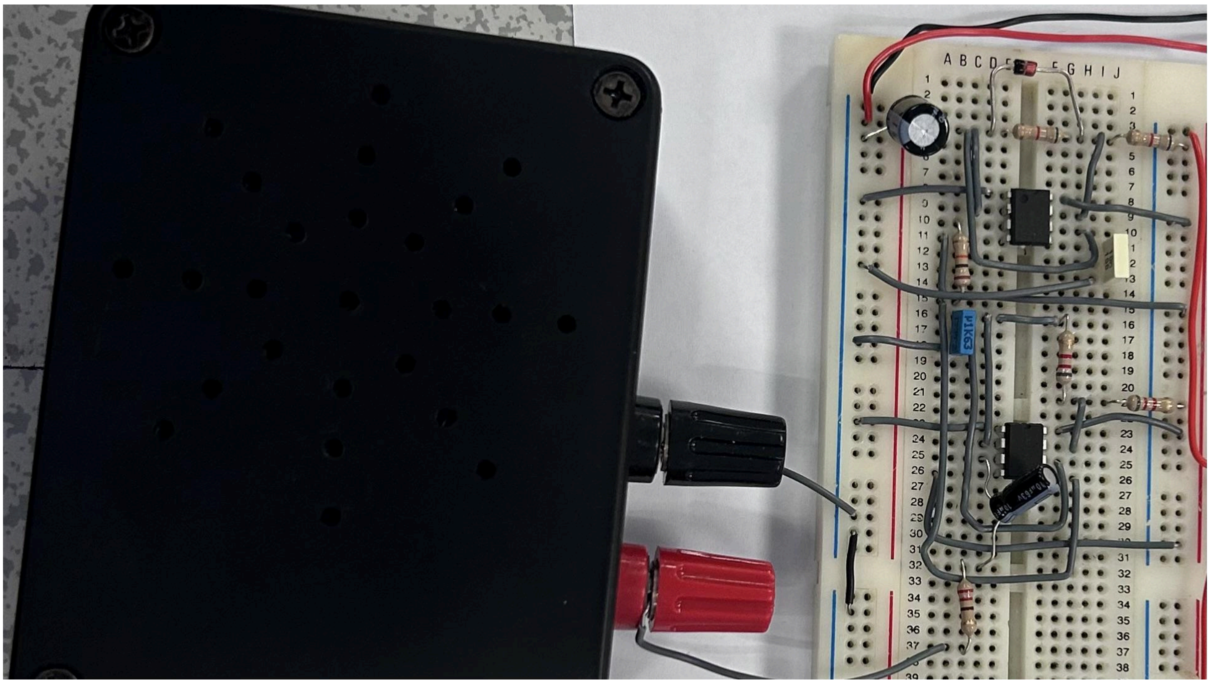


I used two 555 timers to create an astable siren which was similar in sound to the real police sirens which are used in the UK.
This utilised some capacitors to reduce the DC in the system to make the AC sound more distinct when interacting with the speaker

Reasons for choice of subsystem Design

I chose this subsystem design because it was the easiest to test with good results which were needed. It also utilised DC to provide an AC signal using the astables which was needed with the speaker as this speaker utilises AC only.

Photo of Subsystem



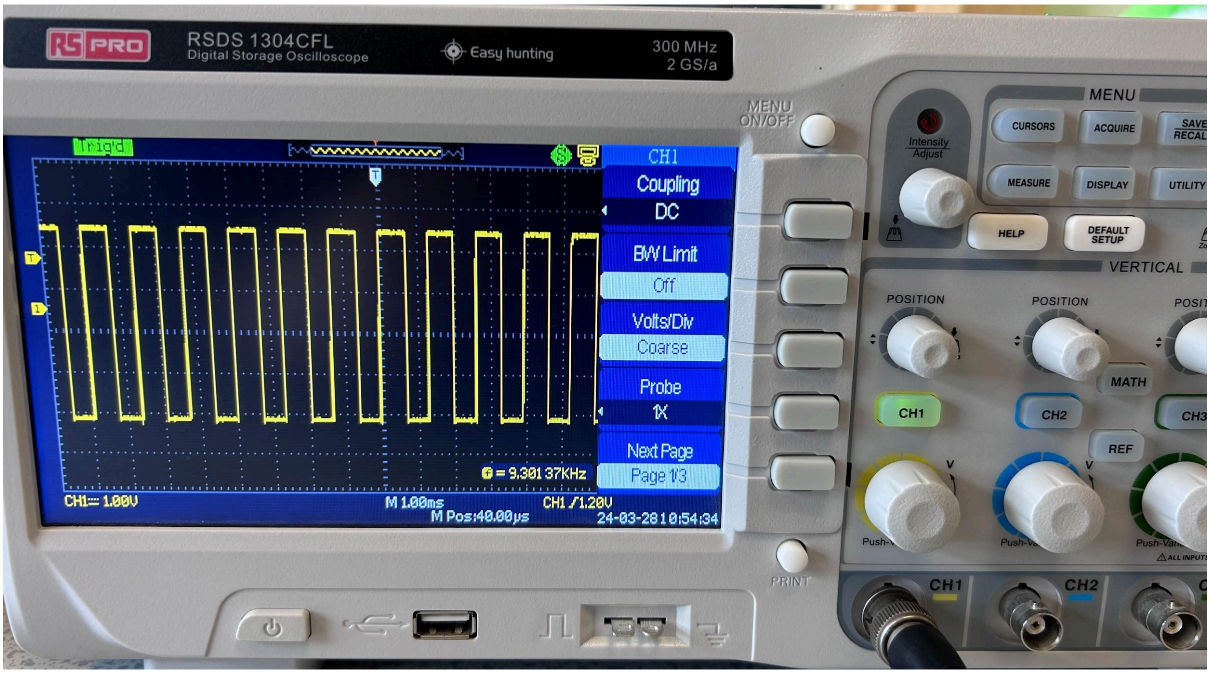
Test Procedures and Equipment Used

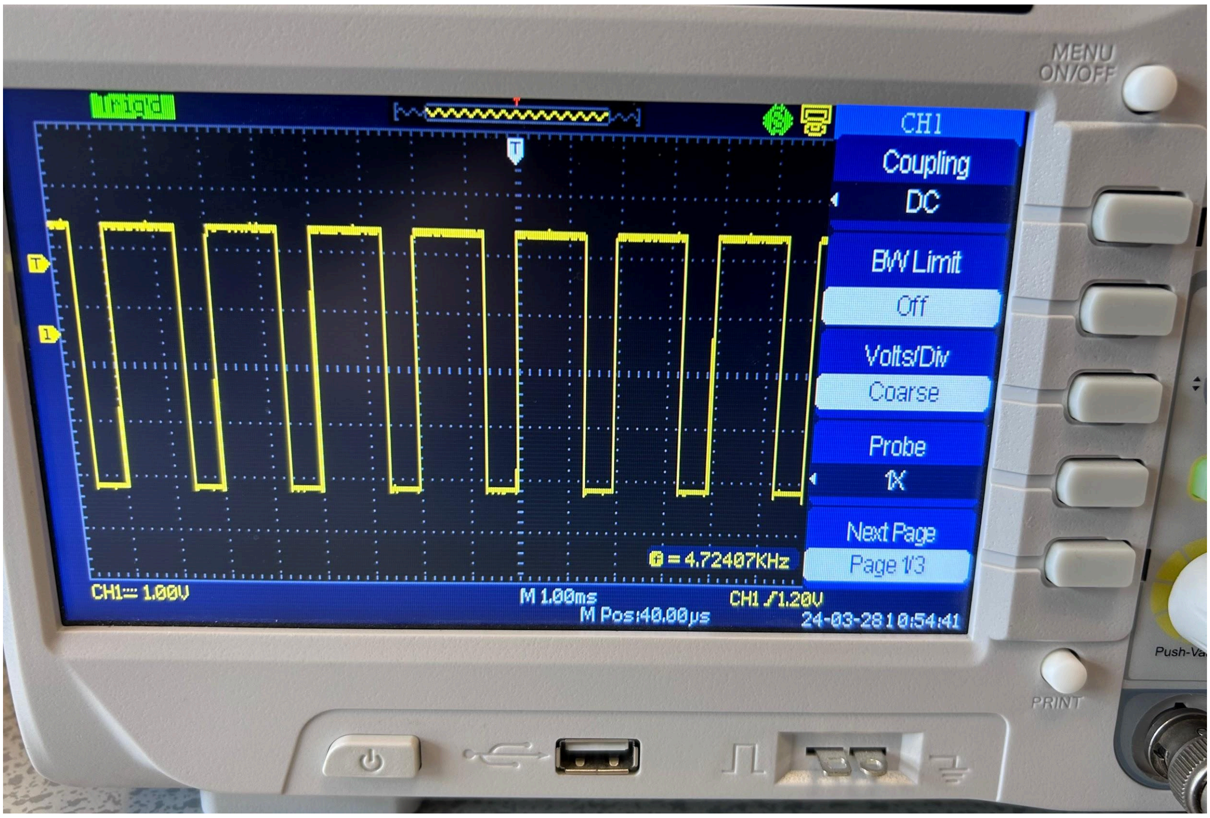
I am going to use an Oscilloscope to get a square wave visualisation of my Police siren functioning.

I am using 5V power in order to get these results as this is what the whole circuit is using in the specification.

I will use the oscilloscope to measure the output into the speaker which has its output produced by the two astable circuits.

Test Results





Comparison of results against specification

These results were what was needed in my specification, the oscilloscope gave great results showing me that my expected results from the specification were very close to my actual results.
They provided a quick change in voltage to produce a square wave which is exactly what was needed to create such a siren.

Evaluation of Subsystem

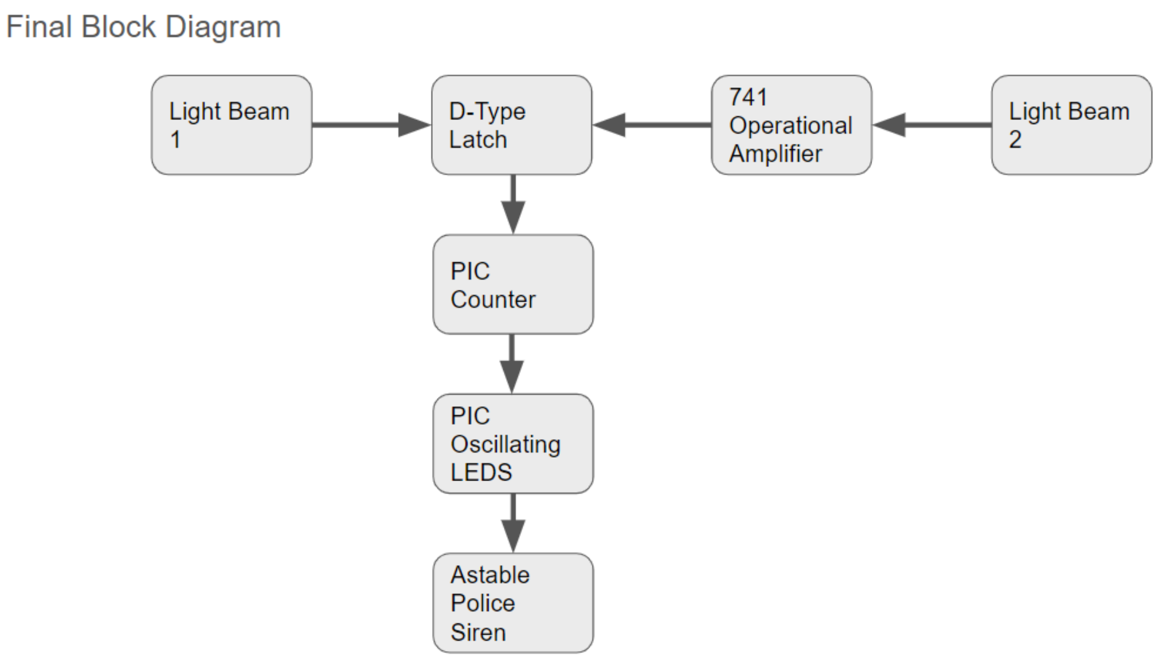
Overall, I believe that this subsystem is working amazingly well as it can take an incoming DC signal from the PIC and turn this into a working AC signal to produce a siren noise.

One issue which could have been prevalent is the PIC not producing enough voltage to power the siren, yet this was no issue.

One improvement I could make is to add a lower resistor value to make the speaker sound louder.

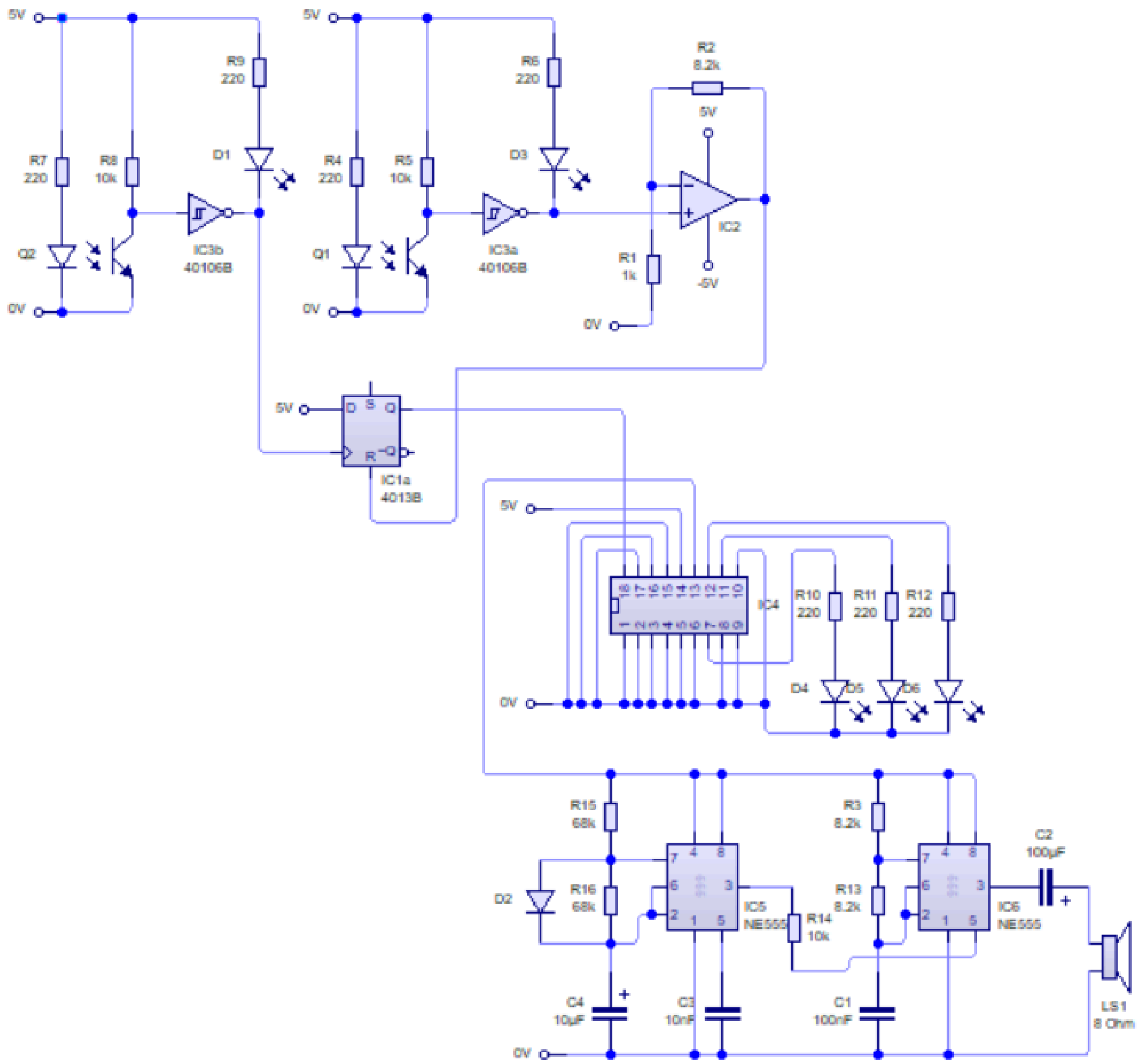
NEA Coursework Part 3

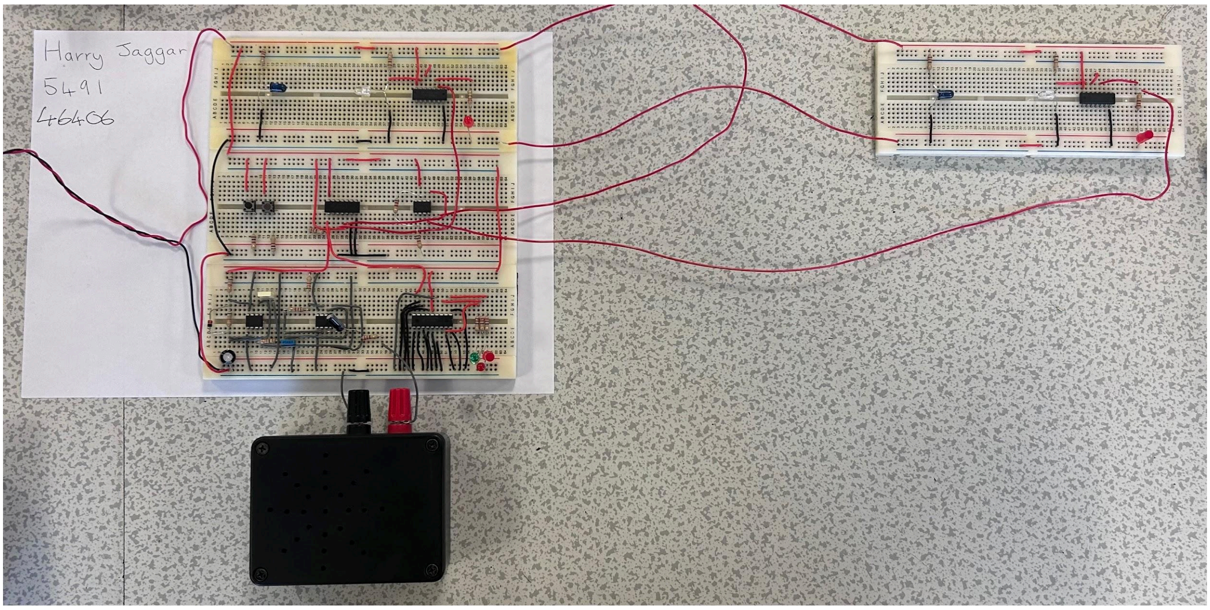
Block Diagram of Final Circuit:



The block diagram above shows how each subsystem leads into one another, this is helpful for any user as they can see how they all interact with each other.

Circuit Diagram of Final Circuit:





I have made a circuit diagram through the software application circuit wizard, and then used this to be able to build a correct representation of this diagram using real components. This helped me a lot as I was able to demonstrate how the circuit would look and work without having to fully build the functioning circuit at once.

Component List

Name	Cost	Quantity	Total
0V Voltage Rail (Ideal)	0.00	3	0.00
10µF Capacitor (100V)	0.00	1	0.00
100µF Capacitor (100V)	0.00	1	0.00
100nF Capacitor (100V)	0.00	1	0.00
10k Resistor (1/4W)	0.00	3	0.00
10nF Capacitor (100V)	0.00	1	0.00
1k Resistor (1/4W)	0.00	1	0.00
220 Resistor (1/4W)	0.00	7	0.00
40106B Hex Schmitt Inverter	0.00	1	0.00
4013B Dual D-Type Flip-Flop	0.00	1	0.00
5V Voltage Rail (Ideal)	0.00	5	0.00
-5V Voltage Rail (Ideal)	0.00	1	0.00
68k Resistor (1/4W)	0.00	2	0.00
8.2k Resistor (1/4W)	0.00	3	0.00
Diode (Ideal)	0.00	1	0.00
LM741 Operational Amplifier	0.00	1	0.00
Loudspeaker (8 Ohm)	0.00	1	0.00
NE555 Bipolar Timer	0.00	2	0.00
Opto-Isolator (Ideal)	0.00	2	0.00
Red LED (0.2 in, 5 mm)	0.00	5	0.00
		Total	0.00

The circuit component list above shows every single component which I had used in the making of my 6 subsystems. This helps anyone who is aiming to recreate the system, and also shows me how much the circuit would cost if I were to add the costs to the components themselves.

Test Plan for the Final Circuit

Subsystem 1: Break the beam

I will use a multimeter to gain the voltage readings from the inputs and outputs of the circuit. This should give me readings which will drop from 5V to 2.5V when the beam is broken, and invert this to give me an output when the voltage drops.

Subsystem 2: Operational Amplifier

This subsystem will be tested using a multimeter where I can find out the input voltage value from the second break the beam system, and then test the output voltage to see if it matches the value which is labelled in my specification.

Subsystem 3: D-Type Latch

I shall use a multimeter to record the voltages when the switches to trigger the latch are off, this should be 0V.
When the switch is pressed to trigger the latch, I will measure this voltage which should be approximately 5V on the input.
The latch should output 5V when active and tested.

Subsystem 4: PIC Counter

Using the PIC microcontroller board I will be able to test whether the counter is working by adding in a test LED which will flash when the count increments correctly as stated in the assembly code.
Using a multimeter I can also test the voltages of the inputs to the PIC to see if they are functioning correctly.

Subsystem 5: PIC Oscillating LEDs

Using the PIC microcontroller board again I will be able to test the outputs on the board for the different LEDs, this is helpful as I could test the PIC on the board before I got the whole built circuit working itself.
I can also use a stopwatch to time the count for the Green LED, as this should flash after 10.2 seconds as the assembly code states.

Subsystem 6: Police Siren

Using an oscilloscope I will be able to map out the AC wave which is created by the two astables connected into each other.
As the speaker utilises an AC wave, this needs to be measured by the oscilloscope where I can see the amplitude of the wave.

Test Results

Subsystem 1: Break the beam

Subsystem 2: Operational Amplifier

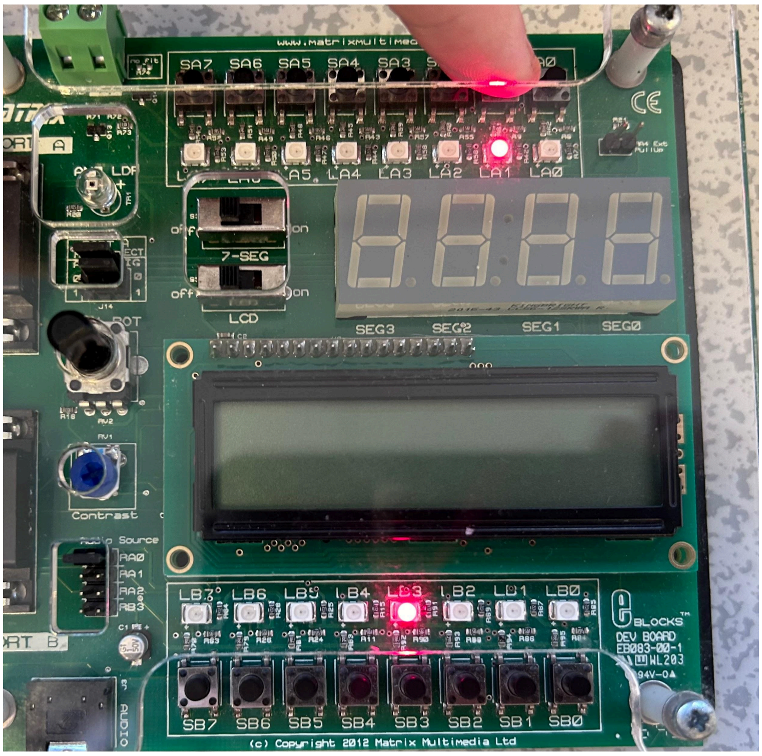
Input Voltage:

Output Voltage:

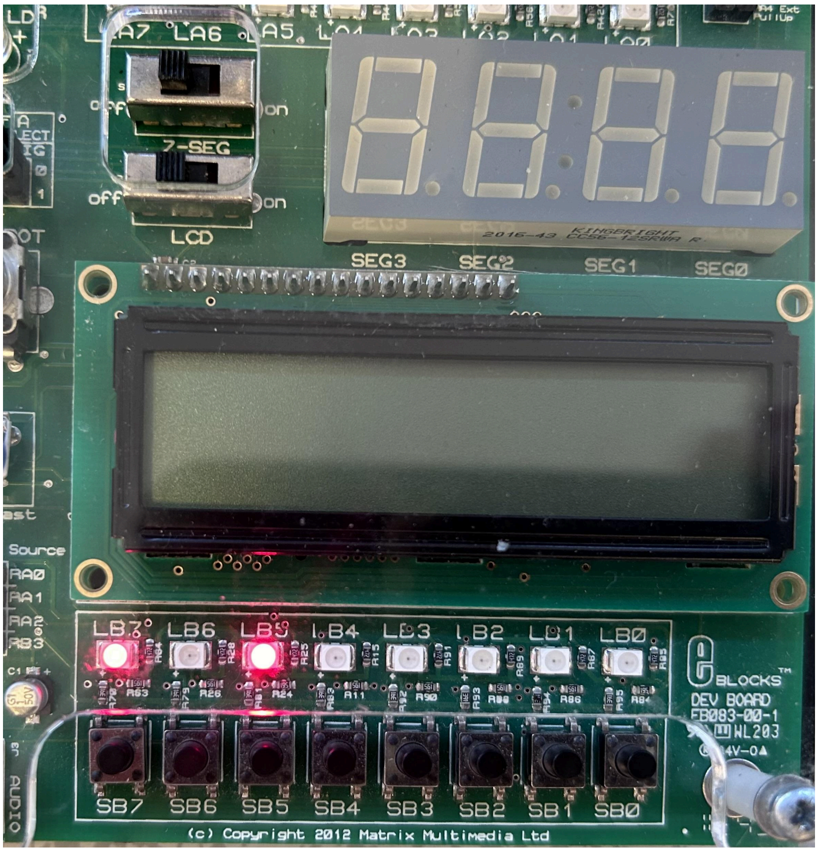
Subsystem 3: D-Type Latch

Output Voltage:

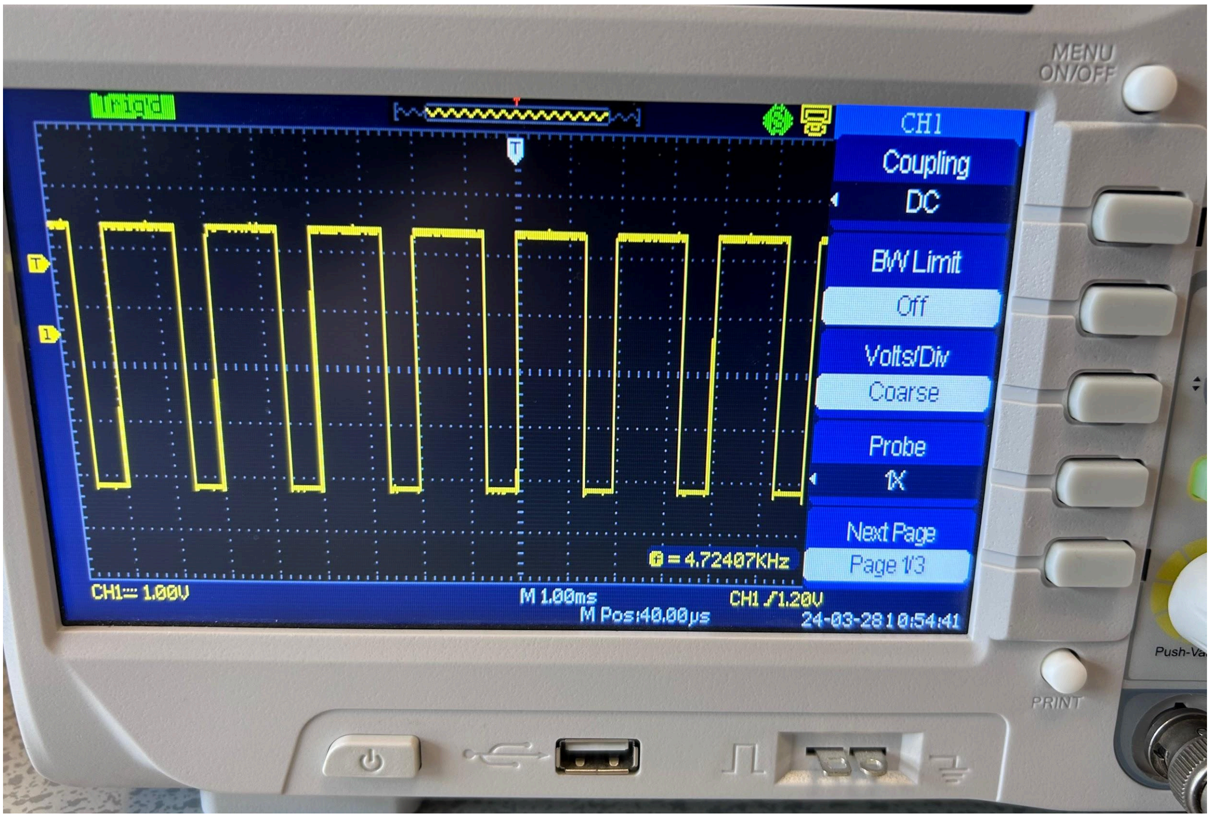
Subsystem 4: PIC Counter



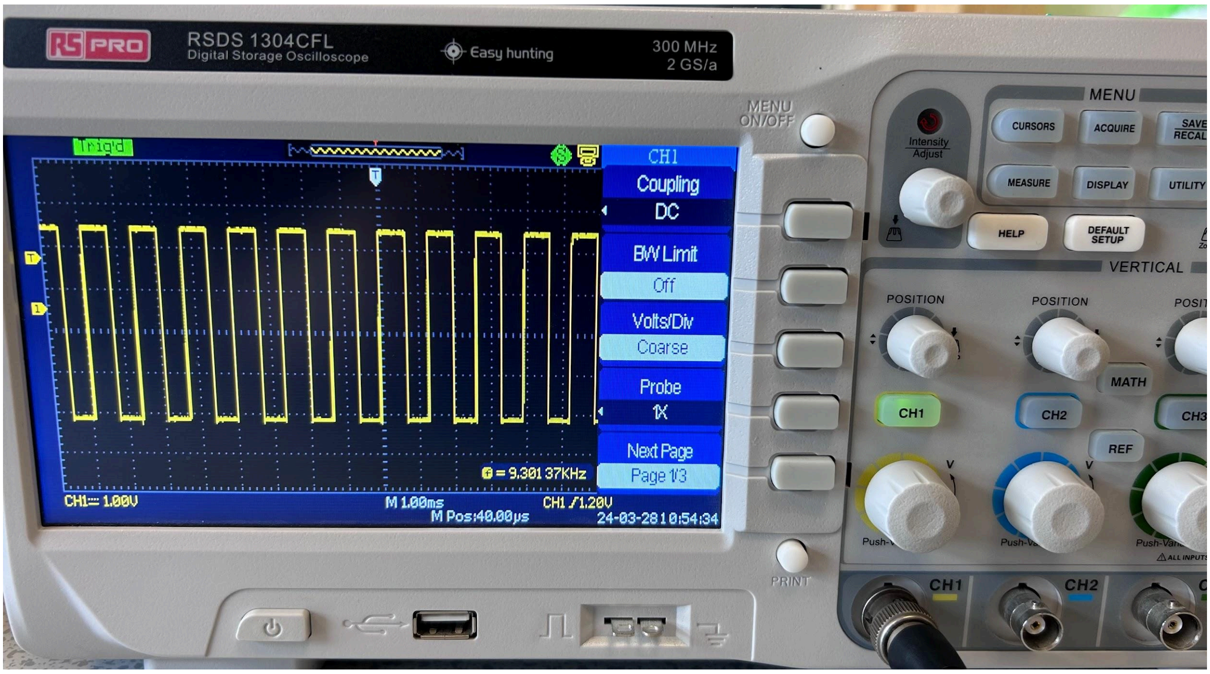
Subsystem 5: PIC Oscillating LEDs



Subsystem 6: Police Siren



MENU ON/OFF
Trig'd
CH1
Coupling
DC
BW Limit
Off
Volts/Div
Coarse
Probe
1X
Next Page
Page 1/3
= 4.72407KHz
CH1 1.00V
M 1.00ms
M Pos:40.00μs
CH1 1.20V
24-03-28 10:54:41
PRINT



RS PRO
RSDS 1304CFL
Digital Storage Oscilloscope
Easy hunting
300 MHz
2 GS/a
MENU ON/OFF
Trig'd
CH1
Coupling
DC
BW Limit
Off
Volts/Div
Coarse
Probe
1X
Next Page
Page 1/3
= 9.30137KHz
CH1 1.00V
M 1.00ms
M Pos:40.00μs
CH1 1.20V
24-03-28 10:54:34
PRINT
MENU
CURSORS
ACQUIRE
MEASURE
DISPLAY
UTILITY
Intensity Adjust
HELP
DEFAULT SETUP
VERTICAL
POSITION
POSITION
MATH
CH1
CH2
REF
CH1
CH2

Analysis of results Compared to Original Specification

Original Specification:

> When the first light beam is broken, start the timer (2% Tolerance)

> When the beam is broken, end the timer (2% Tolerance)

> The time from the timer is stored to use for finding the speed

> The speed of the object is measured using the time (5% Tolerance in measurements)

> If speed is over 0.1 m/s then trigger an alarm

> Alarm sounds for 5s, followed by a 1s break, and repeats until reset button is pressed

Compared to the original specification, there have been a few changes in the circuit design which will add some new items to the specification.

Using the speed equation:

Speed = Distance / Time

I can use the values for the distance and time which I have set, these being:

Speed = 1m / 10s

Speed = 0.1m/s

I have also used a different method for measuring the speed, which now doesn't store the speed into the PIC registers.

Instead, I have used a decrementing counter in the PIC using 8 bits.
This can also use a changing distance if I use different wait values.
This is a better way to manage the speeding system and works more efficiently.

Interfacing issues.

One interfacing issue which I had come across was the noise and interference which was surrounding my PIC.
Due to there being 5 other working subsystems surrounding my PIC, there was a lot of noise surrounding the microcontroller. This was mitigated using capacitors, which eliminated the DC noise which was interrupting my PIC. This fixed the PIC from malfunctioning and allowed it to work perfectly.

Another interfacing issue which I had managed to deal with was the lack of voltage which was powering my latching system.
Due to the second beam being 1 metre away from the latch, there was a lot of internal resistance which had affected the overall performance of the beam system.
By adding in a non inverting operational amplifier system, I was able to boost the signal from the second beam, which ensured that it was at a level which was working with the rest of the system. (Working from a 5V power supply)

User Guide:

To use the speed camera device, follow these simple instructions.

> Place both beam circuits a metre apart from each other to work at the appropriate distance. (This distance can be changed if the PIC counter is also changed.)

> Allow a car to travel through the first beam system, triggering the latch and starting the counter.

> If a car passes through the second gate in under 10 seconds or is travelling over 0.1 m/s then the police siren should trigger and the red LEDs should oscillate also.

> If the car passes the second break the beam system in under 10 seconds or is travelling under 0.1 m/s then the green LED will flash 3 times to indicate that the car is not moving faster than the speed limit.

Evaluation:

Overall, the quality of my circuit is very high, and all functionality is prevalent.

All the subsystems work as intended, and the testing has shown that all of the various inputs and outputs are functioning perfectly.

There is not much deviation from the original specification, whereas any specification prompts which were changed were to produce a more efficiently working circuit which did not have as much of a confusing and complex set of instructions.

There have been many different alternative circuits which could have been used yet I chose the circuits and subsystems which utilised the least amount of chips and components as this reduced the overall price of the circuit and avoided any wastage of materials.

When the 5V travels through the break the beam circuit, it is reduced to 0.5V following the Schmitt inverter, this is why the signal is then passed through a non inverting amplifier which raises this voltage to approximately 5V which is sufficient for my circuit as this is what the whole circuit runs off of.

The 5V is then passed through to the set or reset for the D-Type latch which then provides a latched state of off or on, depending on which beam was broken.

If the beam is broken too fast, then the PIC recognises that the latch has turned off due to the lack of an input voltage to the PIC and turns the police siren on as well as the LEDs

This is what is needed for my whole function to work correctly, and after testing I am confident in stating that my circuit functions as stated.

Possible Improvements:

I have gathered a list of possible improvements to be made, which may increase the build quality of my circuit.

> Changing resistor values on the beam to allow the LEDs to light up brighter when the beam is broken. These will ensure that there is clear indication of the outputs being recognised by the interlinked subsystems.

> Ensuring that the operational amplifier does not saturate by increasing the overall voltage which the circuit runs off of, e.g +/-9V.

> Tidying some wires to ensure that no confusion is presented during the use of the circuit.

> Changing the capacitor values and resistor values on the Police Siren to ensure that the siren is loud enough for speeding cars to hear. If the siren is too quiet, then they might have issues hearing or reacting to the speed camera and this will not make an improvement on the roads where these are being implemented.

> The PIC itself is currently being used to produce an output voltage which would be fed into the Police Siren, this is too much stress for the PIC as it only produces a small amount of current, to change this I would link the output of the PIC into the 555 Timers with an AND GATE to produce the same output as normal.

All of these improvements are not crucial or necessary but would provide better results when testing.