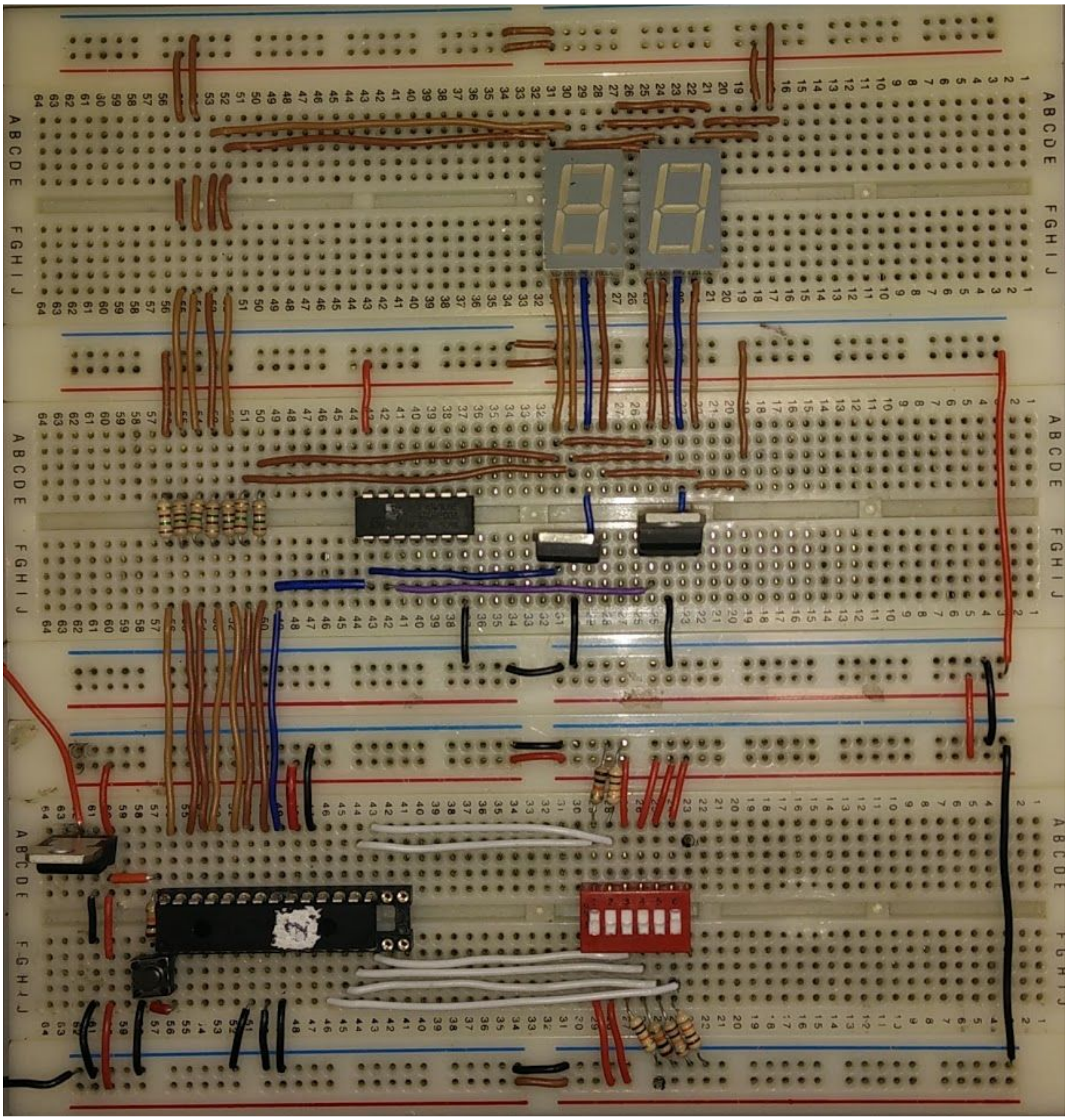
ELECTRONICS A-LEVEL COURSEWORK
Binary to Decimal Converter

Aa)

When learning electronics/computer science people often find it challenging to successfully convert binary to decimal. I will build a system that converts binary to decimal so that they can practise.



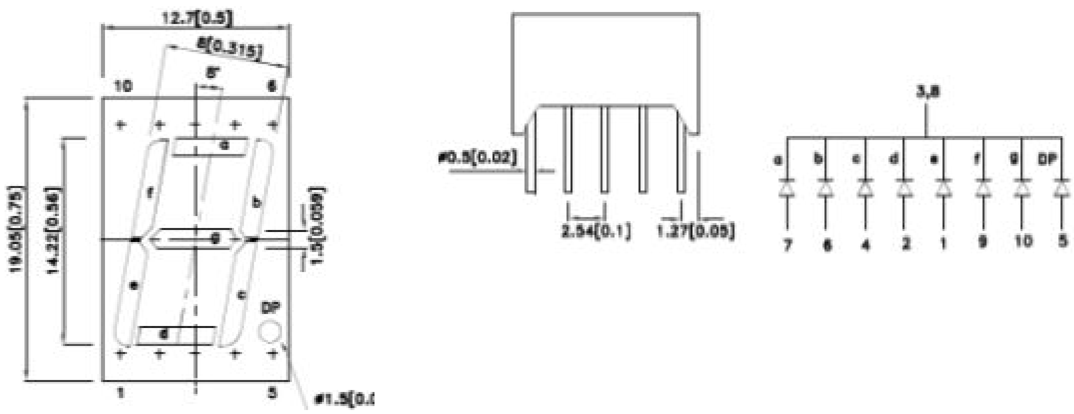
<u>Ab)</u>

Research 1: Displays

Firstly I decided to research the type of displays available to use in my coursework and whether they would be suitable in a solution to my problem.

1. LED displays.

These displays are made of segments of light emitting diodes which can be switched on or off to give a desired symbol. These require a lot of power to display.
To further research the LED displays I searched for the standard values of a 7-Segment display. In our electronics classroom I found the 'KINGBRIGHT SC56-11SRWA' display, from which I used the datasheet to find the following information.



Absolute Maximum Ratings at TA=25°C

Parameter	Super Bright Red	Units
Power dissipation	75	mW
DC Forward Current	30	mA
Peak Forward Current [1]	155	mA
Reverse Voltage	5	V
Operating / Storage Temperature	-40°C To +85°C	
Lead Solder Temperature[2]	260°C For 3-5 Seconds	

Notes:
1. 1/10 Duty Cycle, 0.1ms Pulse Width.
2. 2mm below package base.

Forward voltage: 1.85V, Max voltage: 2.5V[1]

2. LCD displays.

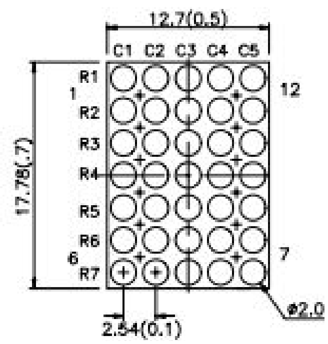
Liquid crystal displays use crystals that do not emit light directly, but use a backlight to produce images. They can be used as a 7-segment display. They are more efficient and have low power consumption as they only require power to change the display; this would

[1] http://www.kingbrightusa.com/images/catalog/SPEC/SC56-11SRWA.pdf (Pages 1 and 2)

give my circuit a long battery life which is useful to the consumer. Unfortunately, these are only visible in bright light, and so are not ideal over a range of light conditions. Many LCDs contain backlights, which are commonly made from LEDs, so require similar power as simply using an LED 7-segment display. [2]
We did not have any LCD displays in the electronics department.

3. Dot matrix displays.

A dot matrix display is a rectangular array of 35 LEDs arranged in a 5x7 grid where all the anodes are connected in the rows and the cathodes are connected in the columns (or vice versa depending on whether it is common cathode or common anode). Dot matrix displays allow for a wider range of characters, but are more complex to code and would require larger frequencies of multiplexing as there are 5 different columns. In the electronics department I found we use TC0711EWA dot matrices, so I decided to research these to assess their suitability.



Absolute Maximum Ratings at T_A=25°C

Parameter	Bright Red	Units
Power dissipation	120	mW
DC Forward Current	25	mA
Peak Forward Current [1]	150	mA
Reverse Voltage	5	V

V_F	Forward Voltage	Bright Red High Efficiency Red Green Yellow Super Bright Red	2.0 2.0 2.2 2.1 1.85	2.5 2.5 2.5 2.5 2.5	V	IF=20mA
I_R	Reverse Current	All		10	uA	VR = 5V

The dot matrix display has a higher power rating (120mW) than the 7-Segment display (75mW); however it has similar current specifications (max ~30mA). [3]

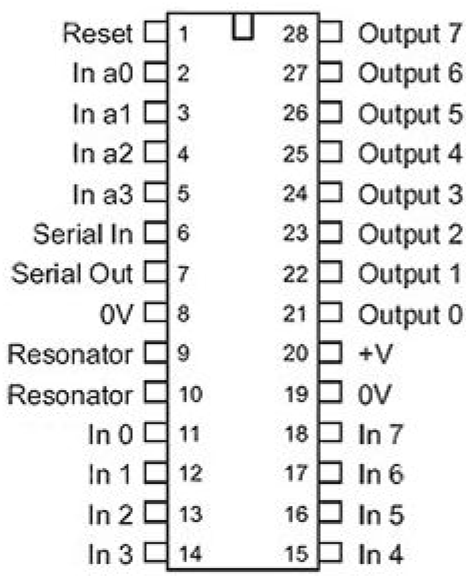
From this information I decided a LED 7-Segment display would be more suitable for this application, as it is more readily available for me than an LCD display, and as it works well in both dim and lit conditions when the display is bright. The 7-Segment display also allows for all the characters necessary (0-9) and so, although the dot matrix allows for more (ie other letters or symbols) and is more aesthetic, it is not necessary in this solution. Also the dot matrix display would be more difficult to program and therefore I have chosen to use the 7-Segment display.

[2] https://en.wikipedia.org/wiki/Liquid-crystal_display
[3] https://www.usbid.com/assets/datasheets/62/tc07.pdf

Research 2 - Programmable chip

In my coursework I plan to use a programmable chip. The ones available to me are the 28x1 chips used in the AQA syllabus. I searched for the datasheet for this and found the following data.

A microcontroller is a 'low-cost integrated circuit that contains memory, processing units, and input/output circuitry in a single unit'[4] and is more economical as it uses less components to achieve the same result by using programming.[5]



In0-7 (often called PORTC) can be configured as inputs or outputs.
Output 0-7 (often called PORTB) can only be used as outputs.
Ina0-7 (often called PORTA) can only be used as analogue inputs.
The serial in and serial out pins must be tied low for proper operation as if left floating the program will not run[6].
+V pin must be connected to between 3-5.5V DC [7] so that the chip will operate. It is recommended to use AA cells, or a 5V voltage regulator from a 9V DC supply; therefore I will need a voltage regulator for this to ensure my voltage does not exceed 5.5V, so I must use a 5V regulator if I am to include a 28x1 chip in my circuit.

A resistor of at least 4.7kΩ is required on the reset pin tied to high to ensure proper operation.

Current ratings:

Sourced or sunk through chip : 90mA
Sourced or sunk per pin : 20mA[8]

Threshold voltage ratings:
In my circuit I will be using 5V supply so:
High: 0.8 x Power supply = 4V
Low: 0.2 x Power supply[9] = 1V

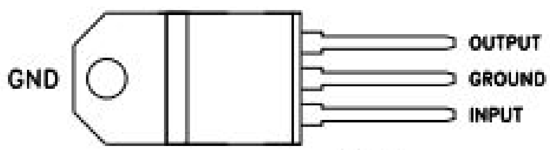
---

[4] http://www.picaxe.com/docs/picaxe_manual1.pdf (Page 12)
[5] http://www.picaxe.com/docs/picaxe_ocr.pdf (Page 3)
[6] http://www.picaxe.com/docs/picaxe_manual1.pdf (Page 35)
[7] http://www.picaxe.com/docs/picaxe_manual1.pdf (Page 25)
[8] http://www.picaxe.com/docs/picaxe_manual1.pdf (Page 8)
[9] http://www.picaxe.com/docs/picaxe_manual1.pdf (Page 8)

Research 3 - Voltage Regulator

As I have found I will need a voltage regulator for my picaxe chip, I decided to research voltage regulators. I found the following information:

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	T_J = 25 °C	4.8	5	5.2	V
V_O	Output voltage	I_O = 5 mA to 1 A, V_I = 7 to 18 V	4.75	5	5.25	V
V_O	Output voltage	I_O = 1 A, V_I = 18 to 20V, T_J = 25 °C	4.75	5	5.25	V



Maximum ratings

Table 1: Absolute maximum ratings

Symbol	Parameter		Value	Unit
V_I	DC input voltage	for V_O= 5 to 18 V	35	V
		for V_O= 20, 24 V	40	
I_O	Output current		Internally limited	
P_D	Power dissipation		Internally limited	
T_{STG}	Storage temperature range		-65 to 150	°C
T_{OP}	Operating junction temperature range	for L78xxC, L78xxAC	0 to 125	°C
		for L78xxAB	-40 to 125	

[10]

From this I found that the regulator has a stable voltage of 4.8-5.2 volts (depending on the regulator) which is within range of my application (needing to be no greater than 5.5V). I also found that I need a minimum supply of 7V and maximum 35V to ensure a stable 5V output, and that the regulator has a current regulation of 100mA.

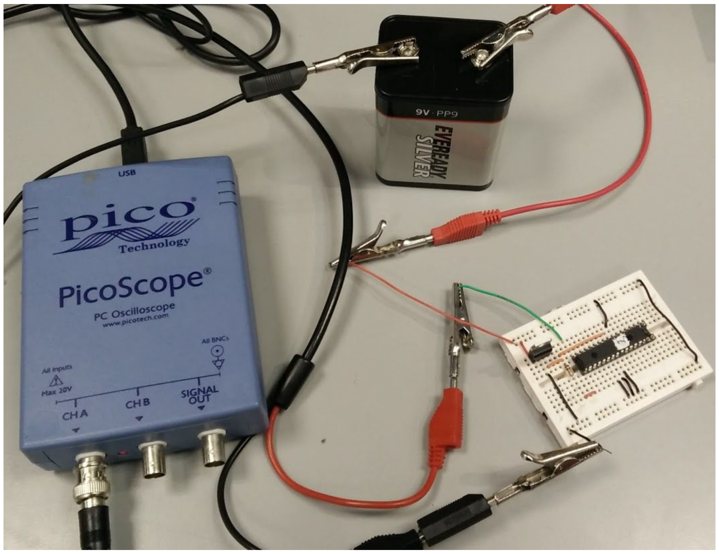
[10] https://www.mouser.com/ds/2/389/l78-974043.pdf (Page 4, Page 14)

Ac)

Practical investigations

1. Frequency of a 28X1 chip

Diagram:



Method:

A simple program was written and programmed onto the 28x1 chip. This was then placed in a circuit and the frequency measured using a picoscope and the program 'PicoScope 6' by attaching the picoscope between the 0v and B0 pin. Because I conducted a voltage regulator test previously, which is used in this circuit, I deduced it unnecessary to complete this test over a range of supply voltages.

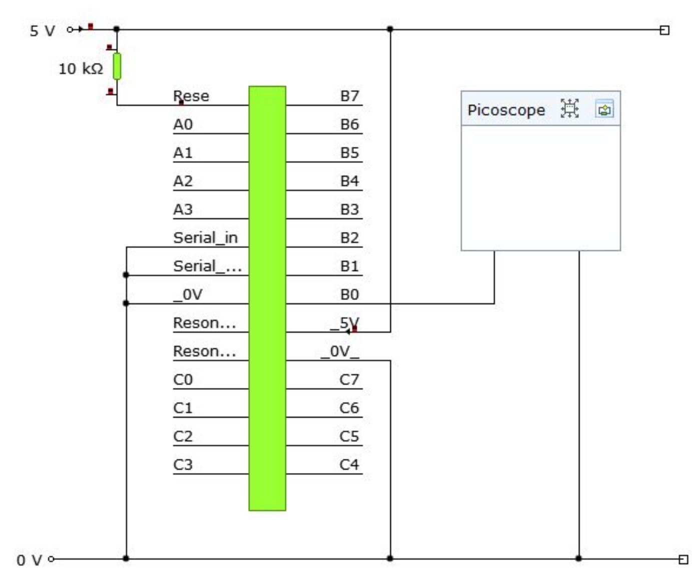
Equipment list: Multimeter (frequency), Power supply, 28x1 chip, Voltage regulator Measurement precision & accuracy.

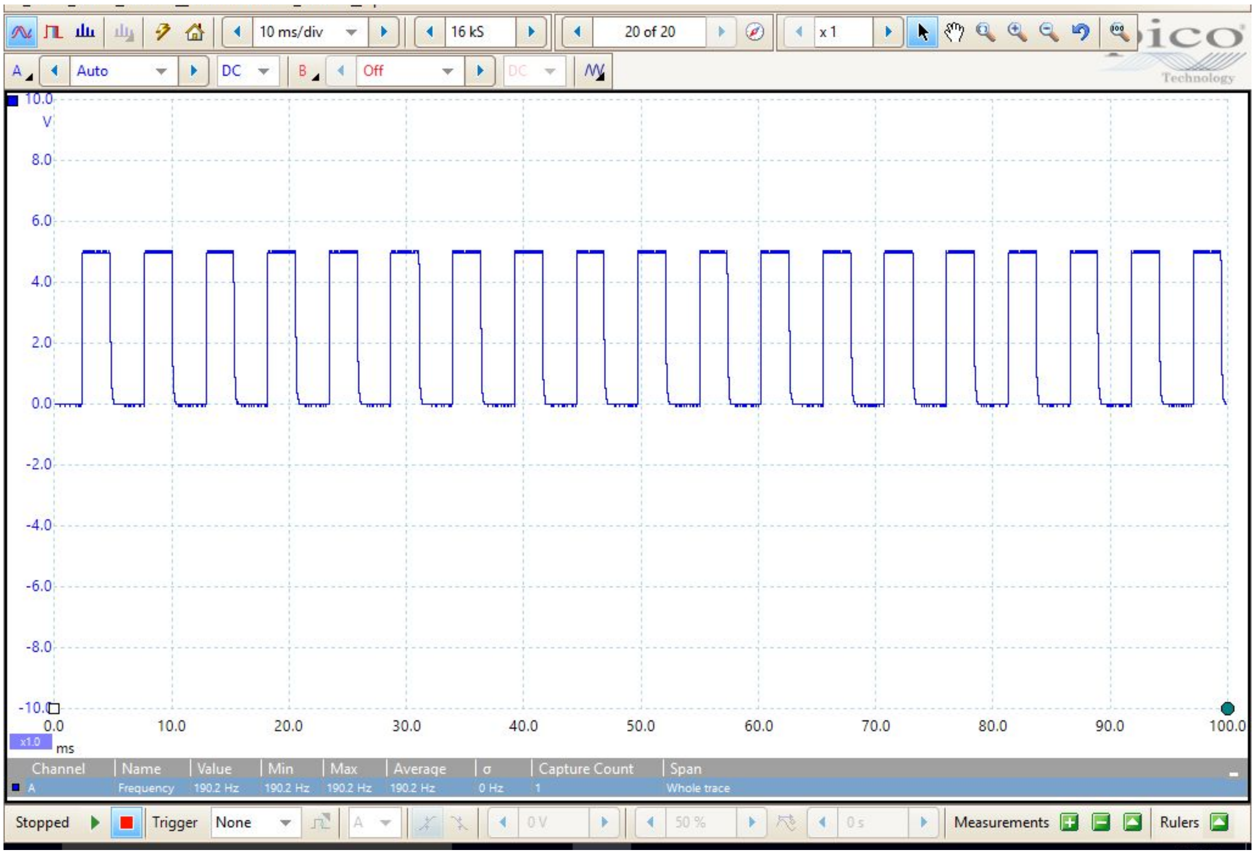
Program:

```
MOVW   0x00
MOVWR TRISB

Flash:
MOVW   0xFF
MOVWR PORTB
MOVW   0x00
MOVWR PORTB
JMP    Flash
```

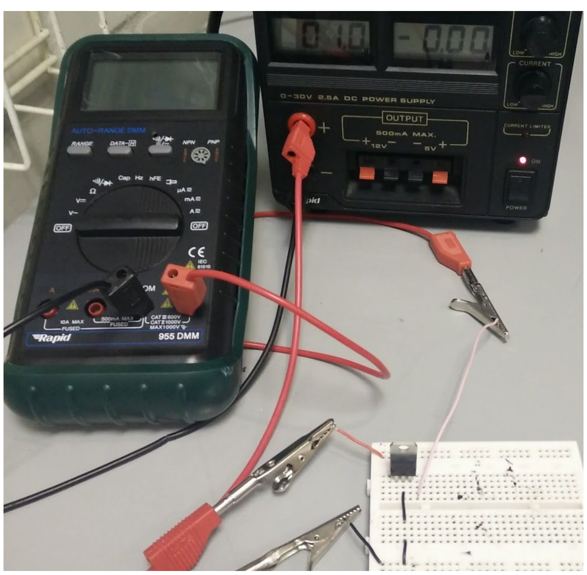
Circuit diagram:

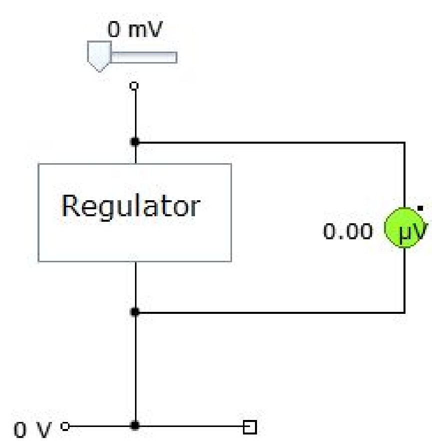




I found the frequency to be 190.2 Hz; however, this value must be multiplied by 6, as my program takes 6 clocks, giving 1141.2Hz.

2. Voltage regulator output over varying supply voltages to measure consistency and value.





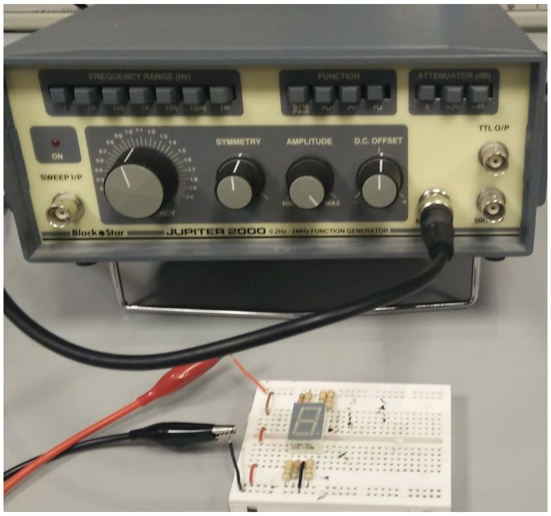
Method:

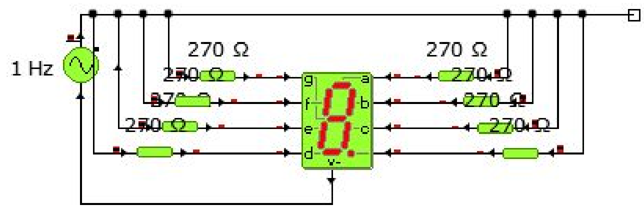
A voltage regulator was placed in a circuit and connected to a variable power supply - as seen to the left. The voltage out of the regulator and the supply voltage was recorded. Equipment list: Multimeter, Power supply, Voltage regulator.

Supply voltage (V)	Regulator Voltage out (V)	Deviation from written value
1	0	-5
2	1.328	-3.672
3	2.344	-2.656
4	3.221	-1.779
5	4.23	-0.77
6	5.09	+0.09
7	5.1	+0.1
8	5.1	+0.1
9	5.1	+0.1
10	5.1	+0.1
11	5.1	+0.1
12	5.1	+0.1
13	5.1	+0.1
14	5.1	+0.1
15	5.1	+0.1

From this we can see that the regulator operates at a steady 5.1V (written value 5V) from a supply voltage of 7V onwards. If a 6V battery were to be used, it could fall below the 5V required output if the battery did not supply quite 6V, so for my operations I must ensure that the supply voltage must be 7V or greater.

3. Minimum frequency for a multiplexed display to appear permanently





Method: A signal generator was set to 'square wave' and connected to a 7-Segment display. The visibility of the display was recorded across different frequencies until it appeared fully solid, to determine the minimum frequency required.

Equipment list: Signal generator, 7-Segment display, 28x1 chip, Voltage regulator

Frequency	Visibility
1	Slow flashing
10	Fast flashing
100	Solid display

At 100 Hz the display appears to be solidly on, so the minimum frequency for the display to be solid must be between 10 and 100 Hz.

Frequency	Visibility
20	Very quick flashing
30	Noticeable flickering
40	Small flickering
50	Miniscule flickering
60	Solid display
70	Solid display
80	Solid display
90	Solid display

The minimum frequency for my display must therefore be 60Hz

Minimum brightness test:

Method: I measured the light output from my phone screen directly above at varying thresholds; I then showed 10 people my phone screen on these same thresholds and asked them to rate the visibility of the display.

Equipment: Phone, Light meter

Display threshold	Luminosity (lux)	Too dim	Okay	Too bright
0%	1	10	0	0
25%	17	1	9	0
50%	43	0	8	2
75%	68	0	3	7
100%	90	0	1	9

I then did some further testing to find what specific brightness was the minimum required to be viewed as visible.

Person number	Minimum brightness (lux)
1	9
2	7
3	5
4	7
5	10
6	8
7	5
8	6
9	7
10	7
Average	7.1

Ad)

Behaviour specification:

1. My circuit will read a 6 bit binary input and output the corresponding two digit decimal output on a display.
2. My display will be visible, solid and easily read.

Numerical specification:

1. My display will have a light output of at least 7 lux.
2. My circuit will respond to a change in input in less than 1 second.
3. My circuit must multiplex at a rate of at least 60Hz.
4. My circuit will operate fully to the behaviour specification and achieve my other parameters over a supply voltage of 7-12V.

Ae)

My display will have a light output of at least 7 lux measured immediately above the display so that it is bright enough to see in darkness, but not so bright that it is hard to see in lit settings (see section Ac, test for minimum brightness).

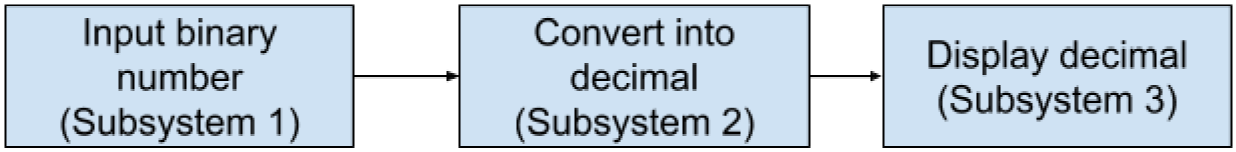
My circuit will respond in at least 1 second so that users do not have wait too long for the output to be shown, and so the operation appears smooth; according to studies, a response time 100ms is perceived as instantaneous and a response of 1 second or less is enough for users to feel that they are freely interacting with the information[11].

My circuit must have a multiplex frequency of at least 60Hz, as my initial research tests indicate this is the minimum frequency to be solidly visible.

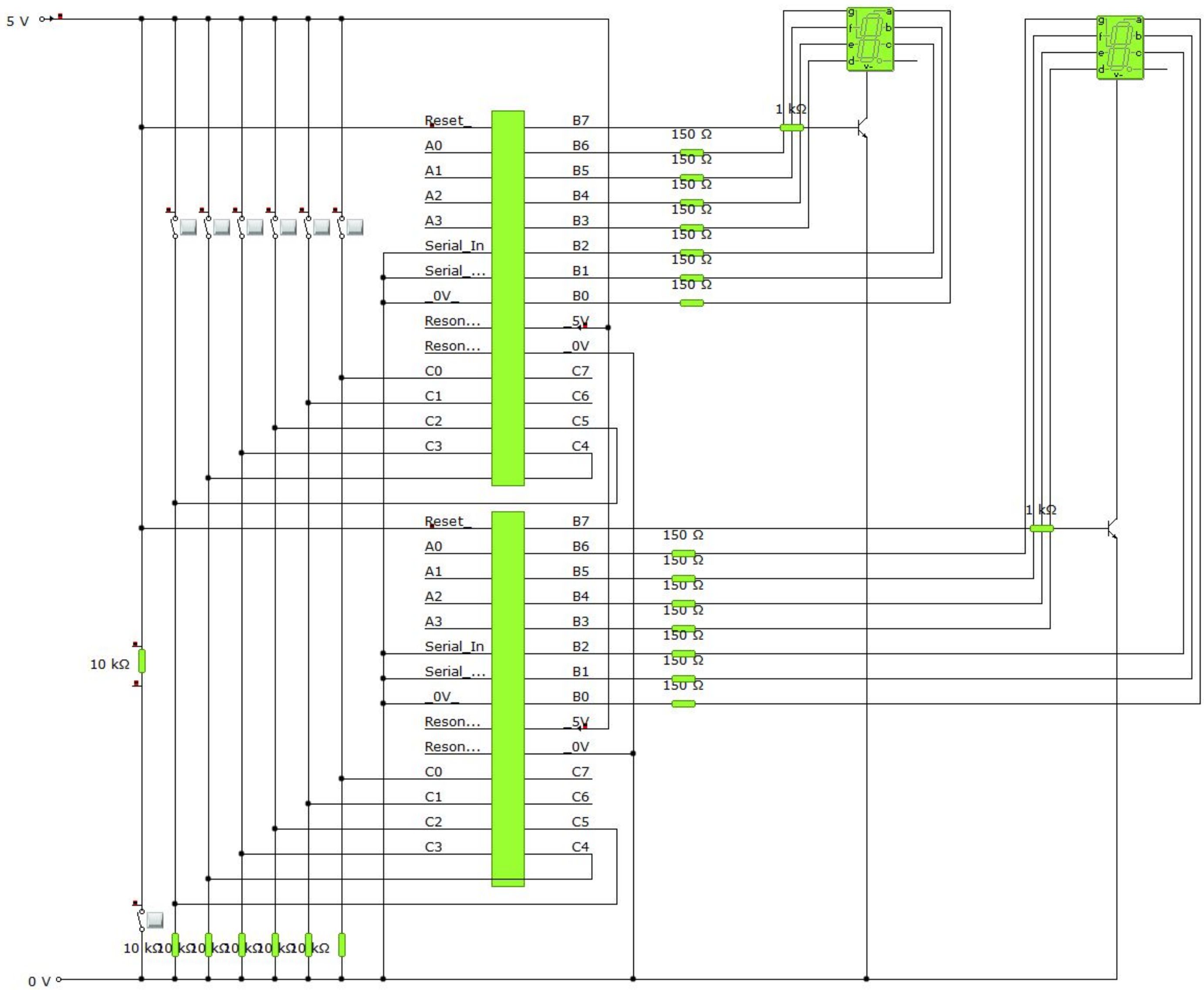
My circuit will operate from a supply voltage range of 7-12V because the regulator needs a minimum of 7V to output a stable 5V supply to the circuit, and up to 12V because the circuit is intended to be of use to teachers/students who will only have access to domestic power supplies which do not often exceed 12V (for example 9V batteries are readily available).

[11] https://www.pubnub.com/blog/2015-02-09-how-fast-is-realtime-human-perception-and-technology/

Af)

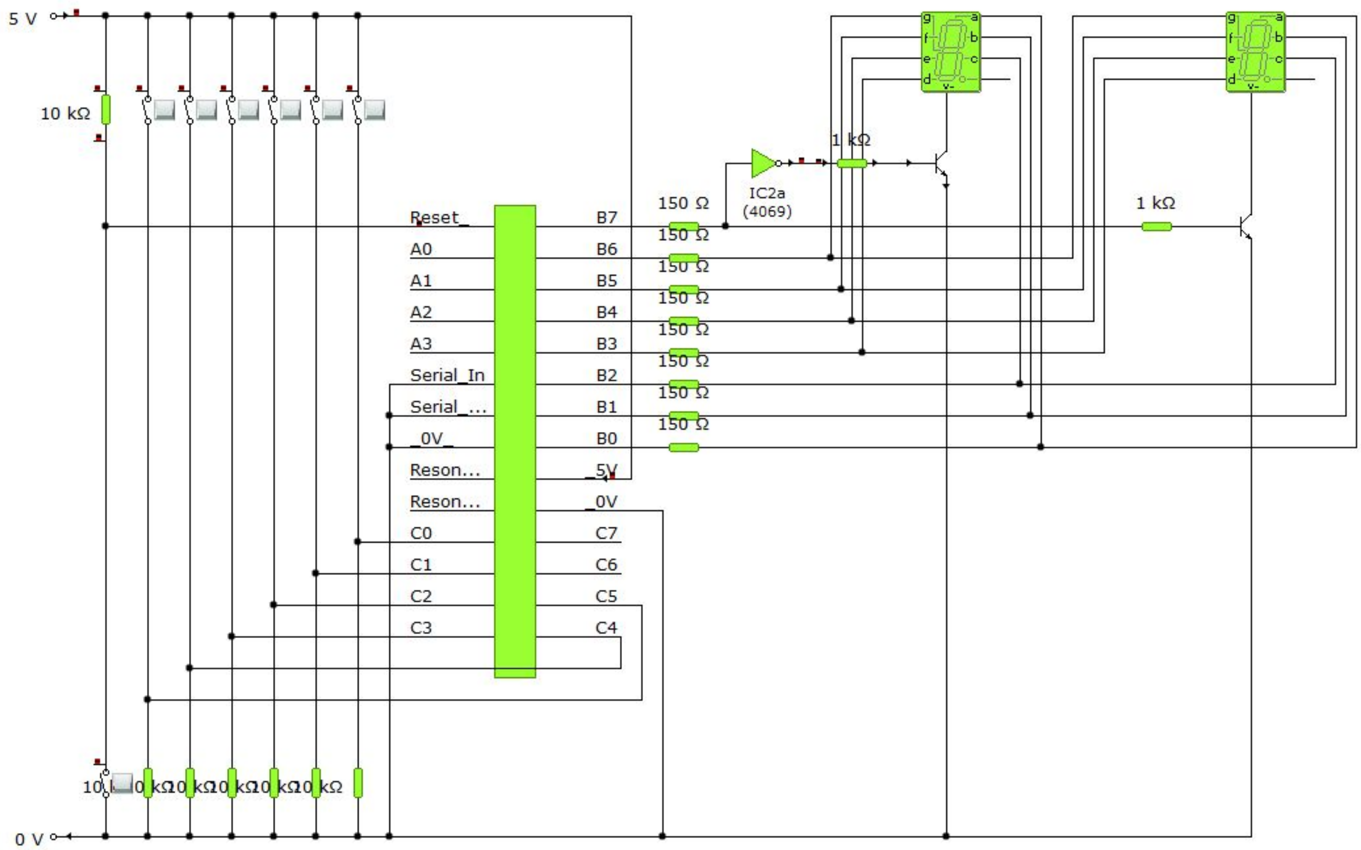


Solution 1:



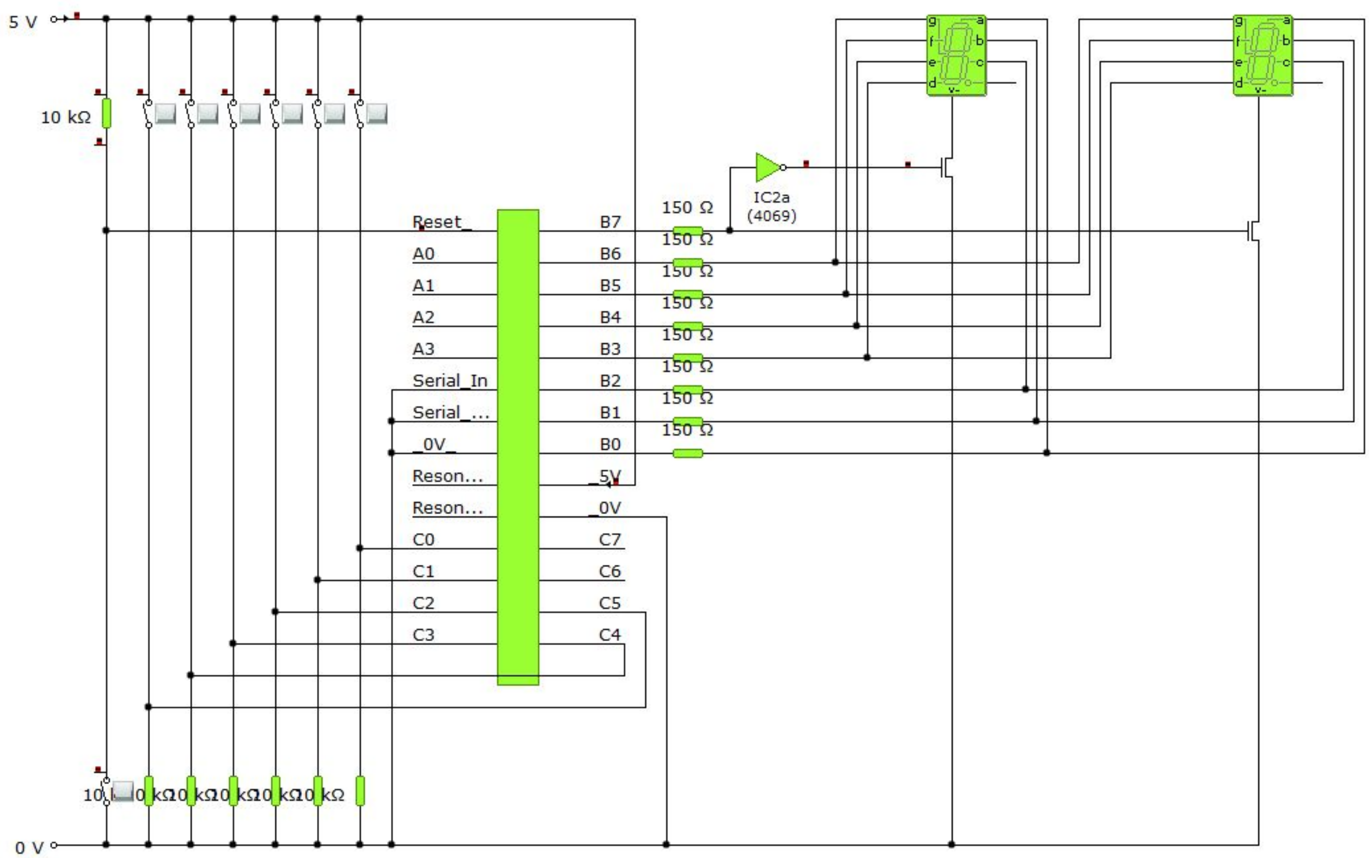
This is my initial design for my circuit. It uses user operated switches to input the binary number, and two 28x1 chip to process this into decimal of each display. It then uses this to turn the display on, and displays the number input in binary on the switches, in decimal on the 7 Segment displays.

Solution 2:



In this solution I have altered the circuit so it will only need one chip. As my circuit uses two displays, I am also using the chip and transistors to multiplex; instead of using two chips to do this, I have added a not gate onto one of the displays so that it reduces the component need considerably.

Solution 3:



In this solution I have changed the transistors to MOSFETs. This reduces the number of resistors needed and therefore is simpler to build.

Solution 1		Solution 2		Solution 3	
Advantages	Disadvantages	Advantages	Disadvantages	Advantages	Disadvantages
Easy to program, only need to program for one display per chip.	More components needed than solution 3. This means it will be more complex to build, and less economical.	Less components needed than solution 1, so more economical.	Transistors - Current driven, NOT gates do not supply much current. Also they require a protective resistor, so the build will be more complex and have extra costs.	MOSFET - voltage switching is ideal as NOT gates do not supply much current. Also no protective resistor is required for a MOSFET.	Harder to program - need to operate both displays on one program.
			Harder to program - need to operate both displays on one program.	Less components needed than solution 1 and 2 so most economical.	

Cost:

The switches, corresponding pull down resistors, and 28x1 chip are all constant over my three solutions; therefore I will only consider the components which vary. All prices are found from the same source - Rapid Electronics. All components are sold in packs of 5 minimum, so the price for 1 component is the same as 2, 3, 4, and 5 components.

Solution 1:

Component	Amount	Cost
28x1 Chip	1	£8.31[12]

Total: £8.31

Solution 2:

Component	Amount	Cost
4069 NOT gate	1	£0.438[13]
NPN transistor	2	£0.945[14]
1kΩ Protective Resistor	2	£0.288[15]

Total: £1.671

Solution 3:

Component	Amount	Cost
4069 NOT gate	1	£0.438[16]
N-channel MOSFET (IRF630)	2	£0.550[17]

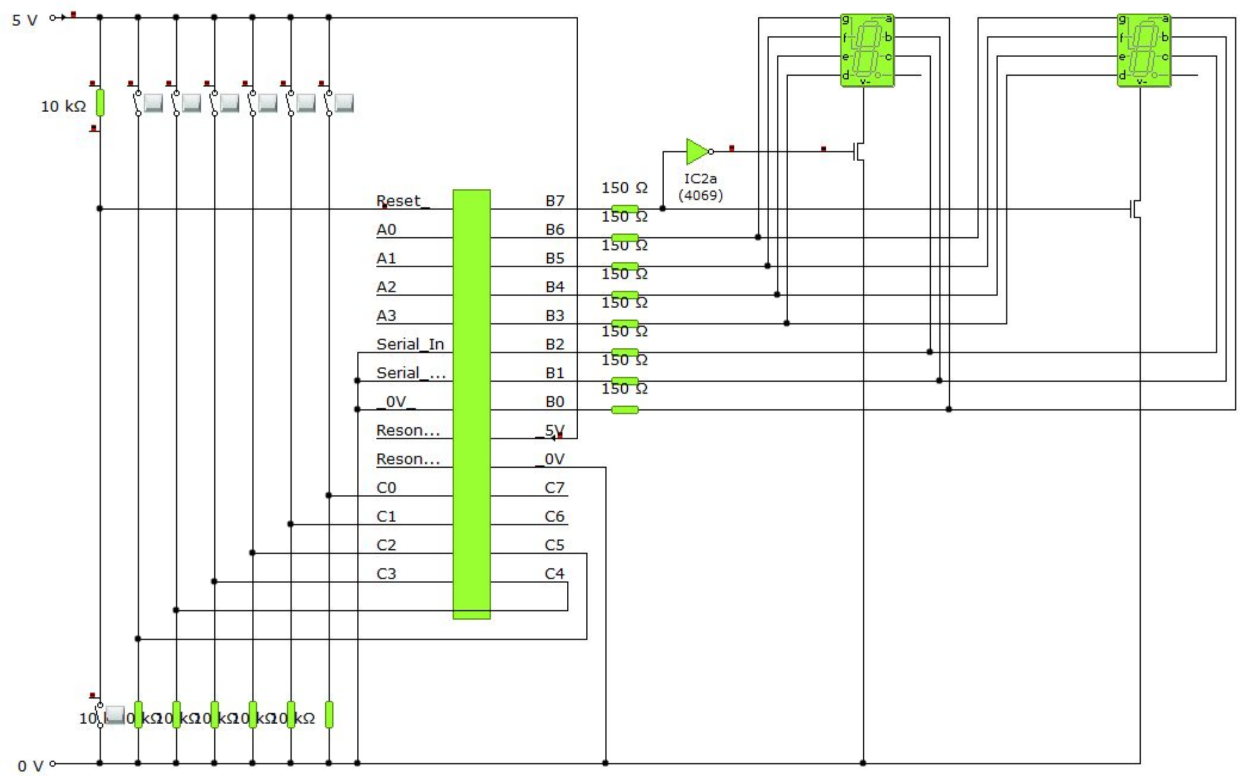
Total: £0.988

From this we can see that the most economical solution is solution 3, then solution 2, and finally solution 1 (from cheapest to most expensive).

[12] https://www.rapidonline.com/picaxe-28x1-chip-13-0862
[13] https://www.rapidonline.com/ti-cd4069ube-hex-inverter-83-0398
[14] https://www.rapidonline.com/catalogue/search?Query=BC109%20npn%20transistor
[15] https://www.rapidonline.com/truohm-knp05sj-1k-5-5w-wire-wound-resistor-51-8995
[16] https://www.rapidonline.com/ti-cd4069ube-hex-inverter-83-0398
[17] https://www.rapidonline.com/st-irf630-mosfet-n-200v-9a-47-0290

Ag)

Final solution



Upon consolidating my alternative solutions, I have chosen solution 3 as my final choice. I chose this because it has the least amount of electronic components needed compared to solution 1 and solution 2. This means it is simpler to build, and therefore less susceptible to errors and faults; additionally, this means it is more economical, as it uses less components to achieve the same result. I have chosen solution 3 over solution 2 because MOSFETs have more capability switching compared to transistors[18]; my circuit will be switching at high frequencies, so it is more suitable to chose MOSFETs. Additionally, one of the transistors/MOSFETs will be run from the output of the NOT gate. NOT gates do not supply much current but have no issues supplying voltage, and therefore it is wiser to use MOSFETs in my solution compared to transistors. The NOT gate provides 5V max output, and the MOSFETS (IRF630) have a minimum 2V threshold voltage[19], so there will always be enough voltage to allow the MOSFET to conduct. Also using MOSFETs means there are no current complications that a transistor would bring, for example having to calculate protective resistor values. Additionally, using MOSFETs instead of Transistors reduces the amount of components necessary as Transistors would require protective resistors, so is

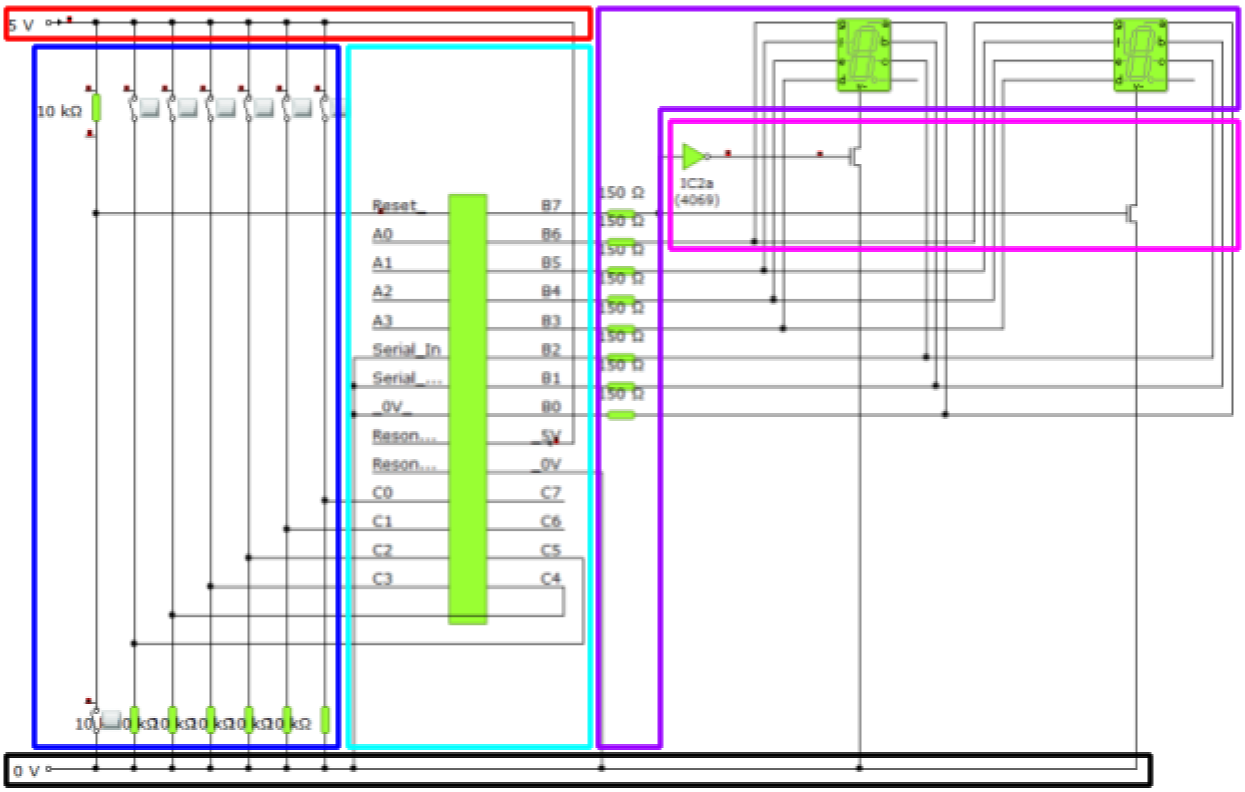
[18] https://www.electronicproducts.com/Analog_Mixed_Signal_ICs/Discrete_Power_Transistors/MOSFET_vs_IGBT.aspx

[19] https://www.usbid.com/assets/datasheets/62/tc07.pdf

therefore more simple. All the components in my solutions are readily available in the electronics department, and so I must also consider the cost of each solution to ensure my final choice is suitable. Solution 3 is much less expensive than solution 1 and 2; therefore it is logical to choose the more economical and efficient solution, 3.

Ba & Bd)

Explanation of the subsystems. My Bb subsystem is the processing subsystem (28x1 chip).



Sequence of steps

Step 1: Input 6 bit binary number on the switches, with the least significant bit on the right.
Step 2: Press the enter button to load the data onto the chip.
Step 3: The 28x1 chip follows the program written to convert this number into decimal and sets the correct output pins high to show the LHS display figure.
Step 4: The not gate inverts this signal, turning on the RHS display.
Step 5: The 28x1 chip follows the program written to convert this number into decimal and sets the correct output pins high to show the RHS display figure.
Step 6: The not gate inverts this signal, turning on the LHS display.
Step 7: Repeat through steps 3-6 until a new number is desired to be converted.

Description of overall circuit operation

The whole circuit is run from a stable 5V DC supply, which interfaces the circuit with the power supply. This is to prevent any damage to the circuit by having too much voltage input, and also to ensure a stable, steady signal and current output, which a regulator supplies. Each switch is connected to a pin of PORTC on the chip. When the circuit is powered up, the initial binary values of the switches will be read by PORTC of the 28x1 chip. The 28x1 chip then follows a programmed code to convert this number into decimal, and will set specific pins high on PORTB. B7 on the 28x1 chip is input to the multiplexing circuit. When B7 is high, the input to the NOT gate is 1, so the output of the NOT gate is 0, so the

input of the left MOSFET will be 0. This is lower than the required input voltage to saturate and so the left MOSFET will not conduct. As there is no current flow, the LHS display will not turn on. When B7 is high, the input to the right MOSFET is high (5V) and so the right MOSFET will conduct. As there is a closed circuit, current flows, and so the RHS display will turn on.
When B7 is low, the input to the NOT gate is 0, so the output of the NOT gate is 1, so the input of the left MOSFET will be 1. This is higher than the required input voltage to saturate and so the left MOSFET will conduct. As there is current flow, the LHS display will turn on. When B7 is low, the input to the right MOSFET is low and so the right MOSFET will not conduct. As there is an open circuit, no current flows, and so the RHS display will not turn on. The other outputs (B0-6) are connected through a resistor to the segments on both displays, and as B7 alternates between 0 & 1, B0-6 alternates between the data to turn the specific segments on the LHS, and the data to turn on the specific segments on the RHS.

Power rail

A 5V regulator is connected to the positive DC supply. On voltages < 7 V the output will not regulate on a steady voltage; on voltages ≥ 7 V the output will saturate at +5V. This is used to interface the power supply to the circuit, to prevent damage due to excess voltage, and to provide a steady signal for all components.

0V Rail

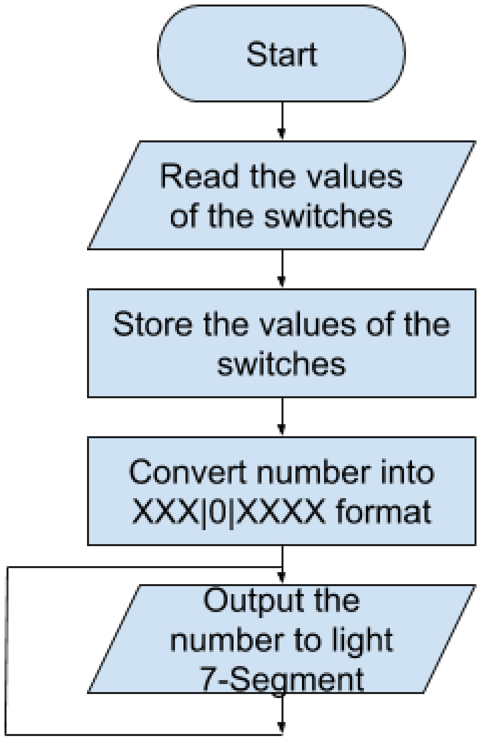
Connected to 0V DC supply

Input subsystem

This system consists of 8 switches in potential divider circuits which represent each bit of a 6 bit binary number, with D0 being the least significant bit on the right. When the switch is open, the loop is not closed circuit, and so no current flows; therefore, V_{out} of the potential divider is 0V, so represents logic 0. When the switch is closed, the loop is closed, and so current flows; therefore the V_{out} of the potential divider is 5V, (as the switch has negligible resistance) so represents logic 1.

Process subsystem

This subsystem takes a binary input from the switches, and outputs the specific pins to be high/low to light up the corresponding number in decimal on the display. For example the number 15 would be inputted on the switches by setting 001111 on the switches. The code (explained below) would be run. After decoding this number, the chip would store two numbers in the memory locations B2 and B3, B2 for number one to light the left hand side seven segment display, and B3 for number 5 to light the right hand side seven segment display. These would be repeatedly loaded onto the outputs one after another to achieve multiplexed displays until the reset/enter switch is pressed.




```
Program
MOVW        0XFF        ;Make port C inputs
MOVWR       TRISC

Conversion:
MOVW        0x00
MOVWR       PORTB
MOVWR       B0          ;Clears values in PORTB & B0 to 0
MOVRW       PORTC       ;Read the values of the switches
MOVWR       B1          ;Stores the values of the switches
ANDW        0x20        ;Tests for a high on B5 (highest input)
JPZ         CHECK4      ;If 0 jumps to test next highest input
MOVW        0x62        ;If 1 stores 32 in a special format:
MOVWR       B0          ;XXX|0|XXXX    Tens|Carry|Units

CHECK4:
MOVRW       B1          ;Reads the value of the switches
ANDW        0x10        ;Tests for high on B4 (2nd highest input)
JPZ         CHECK3      ;If 0 jumps to test next highest input
MOVRW       B0          ;If 1 it adds 16 in special format to the stored value
ADDW        0x26
MOVWR       B0

CHECK3:
MOVRW       B1
ANDW        0x08        ;Test for a high on B3 (3rd highest input)
```

```
JPZ       CHECK2
MOVRW     B0
ADDW      0x08     ;If 1 adds 8 in special format to the stored value
MOVWR     B0       ;Testing the last 3 bits for highs
ANDW      0x1F     ;And with 0001 1111 to isolate 2nd section
JPZ       CHECK2
SUBW      0x02
JPZ       CHECK2
SUBW      0x02
JPZ       CHECK2
SUBW      0x02
JPZ       CHECK2
SUBW      0x02
JPZ       CHECK2   ;Finding the number in the units by subtracting away 2
                   ;at a time
MOVRW     B0
ADDW      0x20
SUBW      0x0A
MOVWR     B0       ;If it doesn't jump adding 22 to the number

CHECK2:
MOVRW     B1
ANDW      0x04     ;Testing bit B2 for a high
JPZ       CHECK1
MOVRW     B0
ADDW      0x04
MOVWR     B0
ANDW      0x0F     ;Testing the values of the other units
JPZ       CHECK1
SUBW      0x02
JPZ       CHECK1
SUBW      0x02
JPZ       CHECK1
SUBW      0x02
JPZ       CHECK1
SUBW      0x02
JPZ       CHECK1
MOVRW     B0
ADDW      0x20
SUBW      0x0A
MOVWR     B0

CHECK1:
MOVRW     B1
ANDW      0x02     ;Testing bit B1 for a high
JPZ       CHECK0
MOVRW     B0
ADDW      0x02
```

```
MOVWR    B0
ANDW     0x0F       ;Testing the other units for high
JPZ      CHECK0
SUBW     0x02
JPZ      CHECK0
SUBW     0x02
JPZ      CHECK0
SUBW     0x02
JPZ      CHECK0
SUBW     0x02
JPZ      CHECK0
MOVRW    B0
ADDW     0x20
SUBW     0x0A
MOVWR    B0

CHECK0:
MOVRW    B1          ;Testing bit B0 for a high
ANDW     0x01
JPZ      LHS         ;Jumping to the LHS if 0
MOVRW    B0
ADDW     0x01        ;Adding 1 to the value in the stored value
MOVWR    B0

LHS:
MOVRW    B0          ;Reads the stored value
ANDW     0xE0       ;Bitmask to only read the values XXX0 0000
JPZ      LHS0
SUBW     0x20
JPZ      LHS1
SUBW     0x20
JPZ      LHS2
SUBW     0x20
JPZ      LHS3
SUBW     0x20
JPZ      LHS4
SUBW     0x20
JPZ      LHS5
SUBW     0x20
JPZ      LHS6       ;Subtracts from the highest possible value until it is
                    ;0, where it jumps to the value

RHS:
MOVRW    B0         ;Reads the stored value
ANDW     0x0F       ;Bitmask to only read 0000 XXXX
JPZ      RHS0
SUBW     0x01
JPZ      RHS1
```

```
SUBW        0x01
JPZ         RHS2
SUBW        0x01
JPZ         RHS3
SUBW        0x01
JPZ         RHS4
SUBW        0x01
JPZ         RHS5
SUBW        0x01
JPZ         RHS6
SUBW        0x01
JPZ         RHS7
SUBW        0x01
JPZ         RHS8
SUBW        0x01
JPZ         RHS9            ;Subtracts from the value until 0 where it jumps to
                            ;the output for that value

LHS0:
MOVW        0xF7            ;Sets the specific pins high to show 0
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS

LHS1:
MOVW        0xA1            ;Sets the specific pins high to show 1
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS

LHS2:
MOVW        0x6F            ;Sets the specific pins high to show 2
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS

LHS3:
MOVW        0xED            ;Sets the specific pins high to show 3
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS

LHS4:
MOVW        0xB9            ;Sets the specific pins high to show 4
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS

LHS5:
MOVW        0xDD            ;Sets the specific pins high to show 5
MOVWR       B2              ;Stores this in memory location B2
JMP         RHS
```

```
LHS6:
MOVW        0xDF        ;Sets the specific pins high to show 6
MOVWR       B2          ;Stores this in memory location B2
JMP         RHS

RHS0:
MOVW        0xF6        ;Sets the specific pins high to show 0
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS1:
MOVW        0xA0        ;Sets the specific pins high to show 1
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS2:
MOVW        0x6E        ;Sets the specific pins high to show 2
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS3:
MOVW        0xEC        ;Sets the specific pins high to show 3
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS4:
MOVW        0xB8        ;Sets the specific pins high to show 4
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS5:
MOVW        0xDC        ;Sets the specific pins high to show 5
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS6:
MOVW        0xDE        ;Sets the specific pins high to show 6
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS7:
MOVW        0xE0        ;Sets the specific pins high to show 7
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS8:
MOVW        0xFE        ;Sets the specific pins high to show 8
MOVWR       B3          ;Stores this in memory location B3
```

```
JMP       DISR

RHS9:
MOVW      0xFC       ;Sets the specific pins high to show 9
MOVWR     B3         ;Stores this in memory location B3
JMP       DISR

DISL:
MOVRW     B2         ;Reads the value in memory location B2
MOVWR     PORTB      ;Moves this to PORTB to light the LHS LEDs
JMP       DISR       ;Jumps to display RHS

DISR:
MOVRW     B3         ;Reads the value in memory location B3
MOVWR     PORTB      ;Moves this to PORTB to light the RHS LEDs
JMP       DISL
```

This code polls each bit from highest to lowest for a high, and then stores it in a unique devised format XXX|0|XXXX representing Tens|Carry bit | Units. This is because the input switches are being operated in binary, but the chip is reading the switch outputs in hexadecimal, and so cannot directly read the switch values. The special format allows you to do polling much more easily. You read the first 3 bits as a hexadecimal number, which represents the 10's column in decimal. The 4th bit acts as a carry. When adding the units, it is possible for it to =16 which would move it into D4 (ie 16 = XXX1 0000 but this needs to be separated over the two displays (1, 6) so needs to be stored this way as 10 and 6) so a carry bit is needed to prevent that from affecting the left hand number, allowing smoother operation. The last 4 bits are units and are read and formatted in hexadecimal. For example if the number 34 was inputted on the switches (XX10 0010) then the system would store this as 0110 0100, which is 0x64 (6 representing 30 on the LHS as the left hand side will always be double its 'true decimal value' as all the numbers are moved one place to the left to give space for the carry bit, and 4 representing 4 on the RHS, with each half being counted as a hexadecimal number). If the number 29 was inputted on the switches (XX01 1101) then the system would store this as 0100 1001, which is 0x49. If the number inputted on the switches was 15 (XX00 1111), then the system would store this as 0010 0101, which is 0x25.

Buffer subsystem

When the output of the 28x1 chip B7 is low, the input to the NOT gate is low, so the output of the NOT gate is high. Therefore the common of the left hand side display (LHS) is 0, and the common of the right hand display (RHS) is 1. Since these displays are common cathode, the LHS display will be on, and the RHS display will be off. When the output of the 28x1 chip B7 is high, the input to the NOT gate is high, so the output of the NOT gate is low. Therefore the common of the LHS display is 1, and the common of the RHS display is 0. Since these displays are common cathode, the LHS display will be off, and the RHS display will be on.

Output subsystem

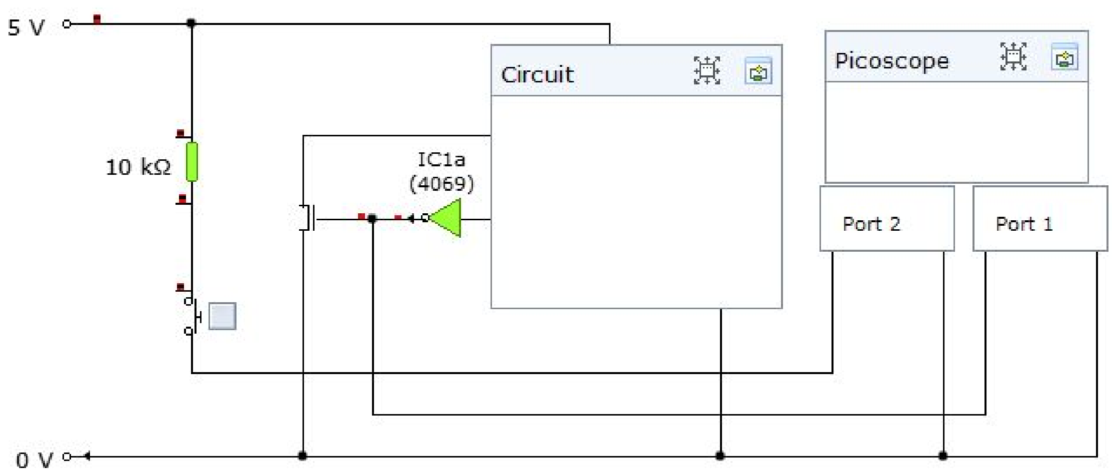
The sectors of the 7-Segment displays are connected to B0-B6 of the 28x1 chip. The decimal point is not connected to anything as it is not going to be used in this system. When the outputs of the 28x1 chip go high, the corresponding segments of the display (which is on according to Buffer subsystem explanation) will light up, outputting a decimal number equivalent of the binary input on the switches.

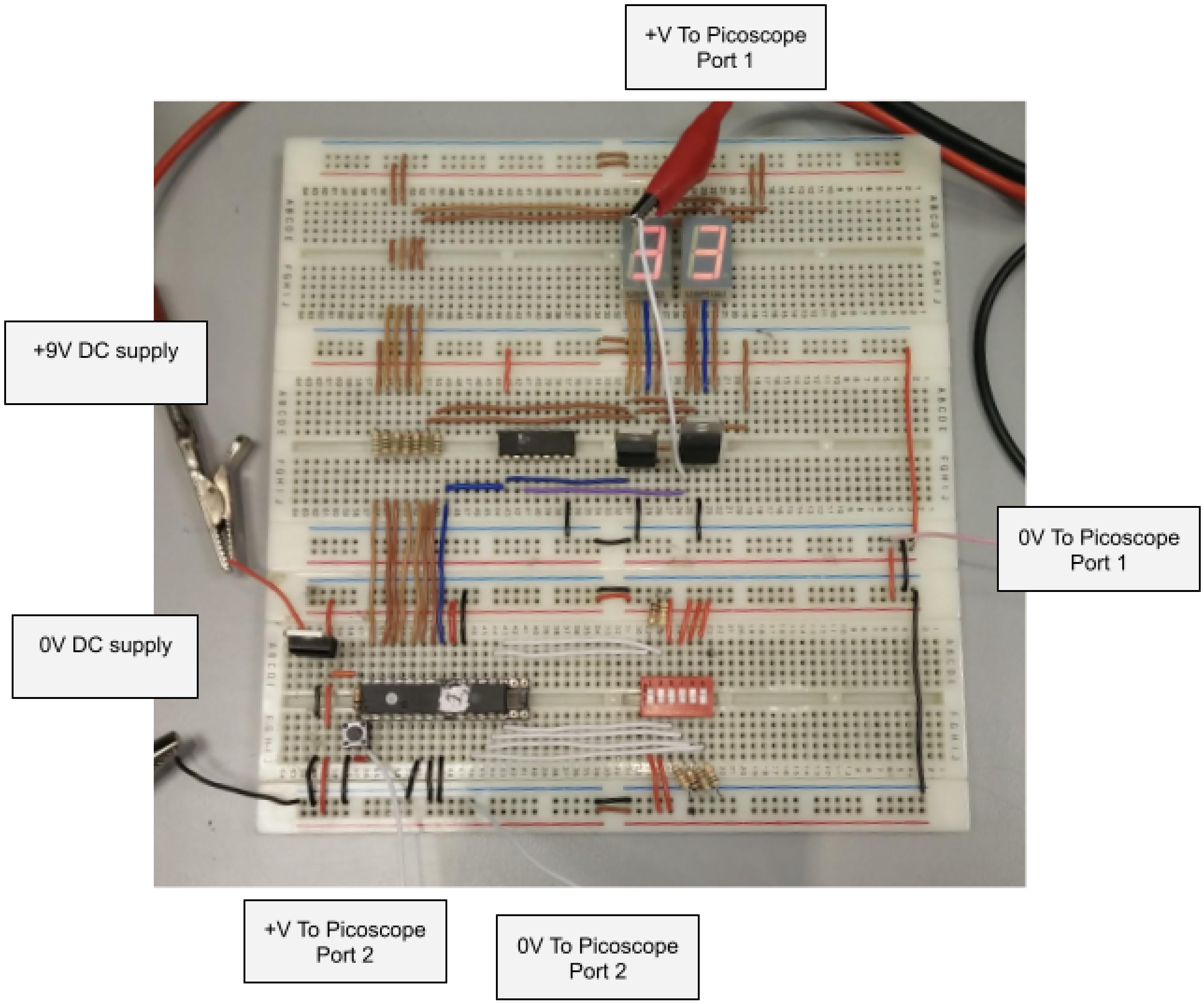
Bb)

Test 1: Response time of the 28x1 chip

Method: The circuit will be switched on with the number 0 initially on the binary switches. The numbers 0, 33 and 63 (so as to see the range of operation time) will be loaded onto the switches in turn and the time taken between pressing the enter button and the output changing is recorded. In picoscope this is found by clicking on the line where the enter switch goes high, and the line where multiplexing begins (the output goes low initially). Port 1 is connected between the enter switch and resistor, and 0V. Port 2 is connected on gate of the MOSFET, and 0V.

Equipment: Circuit, Picoscope, Wires, Power supply.




+V To Picoscope Port 1
+9V DC supply
0V To Picoscope Port 1
0V DC supply
+V To Picoscope Port 2
0V To Picoscope Port 2

Joanna Merrick

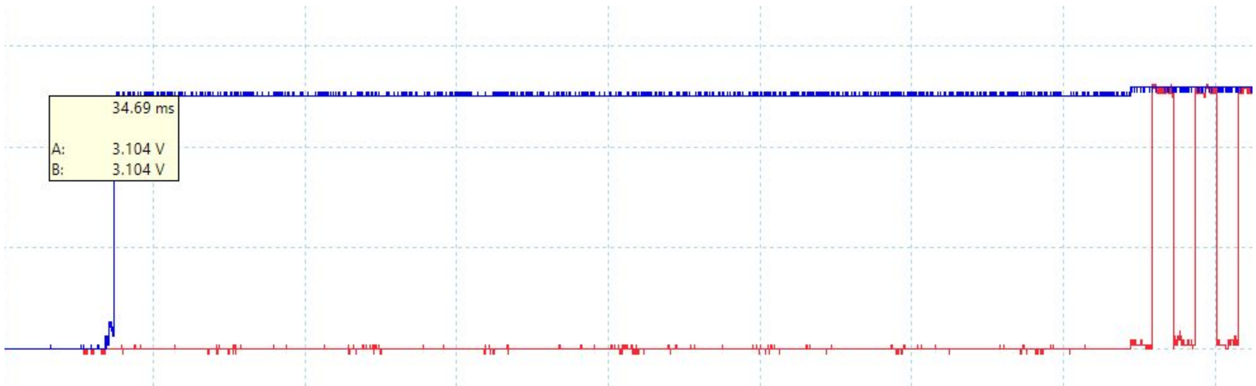
Graph from picoscope showing response time of the 28x1 chip

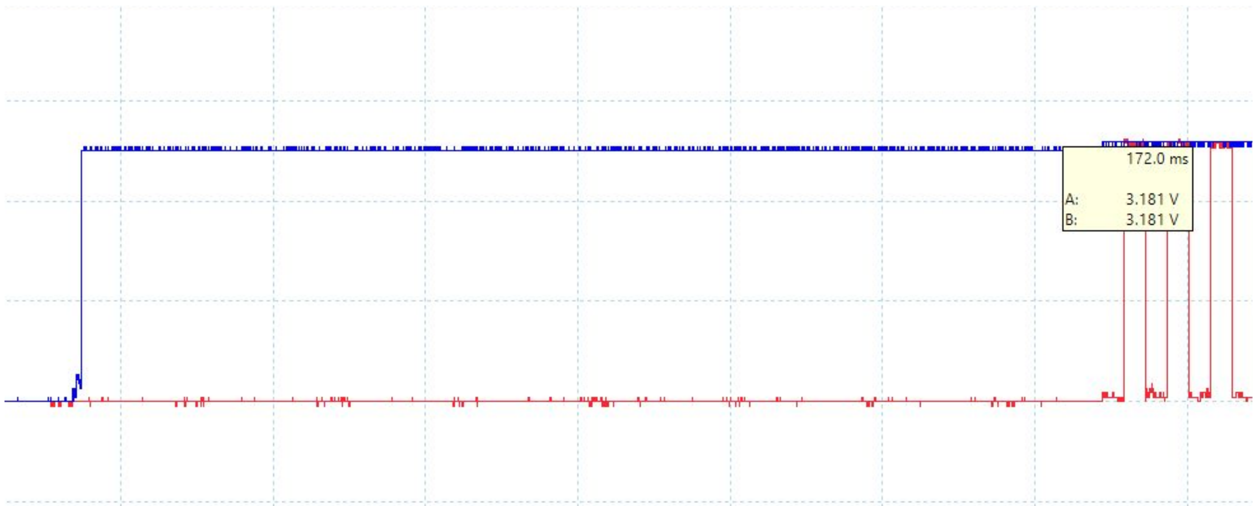
Key: **Red line** = Port 1 of picoscope (Vout on the gate of the MOSFET), **Blue line** = Port 2 of picoscope (Vout between the enter switch and resistor).

Number: 0

Number: 33

Number: 63





Number	Time 1 (ms)	Time 2 (ms)	Response Time (ms)
0	34.97	76.47	41.5
33	37.85	103.5	65.65
63	34.69	172.0	137.31

From this we can see that my response time range is 41.5 - 137.31 ms. According to my specification my circuit must respond in at least one second (1000 ms) as at this time the user feels they are interacting with the system. At 100ms the user perceives the response as instantaneous. The upper limit of my response time is less than the maximum value of 1 second, therefore I can say my circuit meets the requirements of the specification.

Test 2: Valid output from the 28x1 chip

Method:
Each possible number (0-63) in binary is inputted on the switches. Instead of testing the output of all the PORTB pins for each number to confirm that the number is correct, I will simply indirectly test the chip is outputting the right value by comparing the input pins to the visual output on the 7 segment display. Otherwise I would have to make about 450 measurements to test each input, if I were to test each output pin. For photographic evidence of the visual output for each corresponding input see section Cb.
Equipment: Circuit, Power supply, wires.

Binary input	Visual output
0000 0000	00
0000 0001	01
0000 0010	02
0000 0011	03
0000 0100	04
0000 0101	05
0000 0110	06
0000 0111	07
0000 1000	08
0000 1001	09
0000 1010	10
0000 1011	11
0000 1100	12
0000 1101	13
0000 1110	14
0000 1111	15
0001 0000	16
0001 0001	17

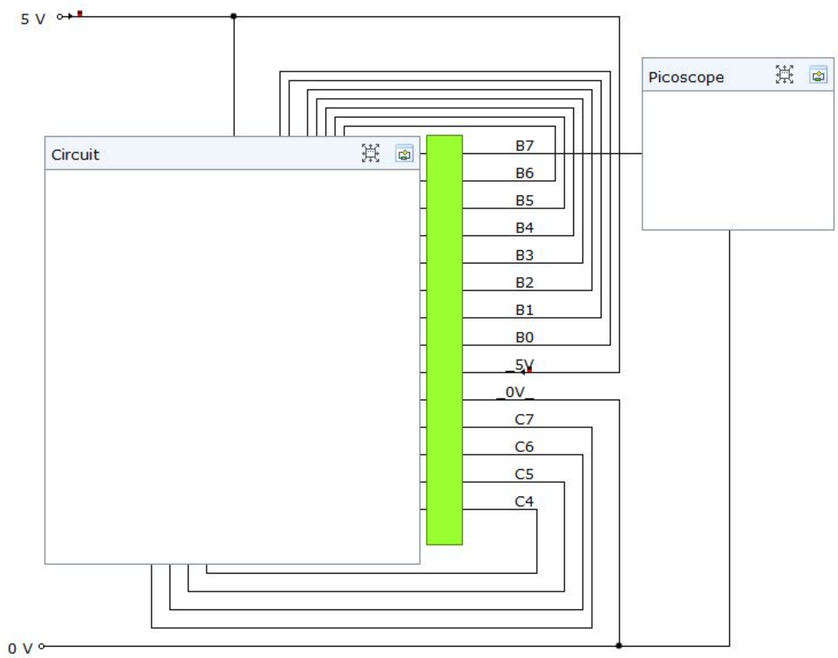
0001 0010	18
0001 0011	19
0001 0100	20
0001 0101	21
0001 0110	22
0001 0111	23
0001 1000	24
0001 1001	25
0001 1010	26
0001 1011	27
0001 1100	28
0001 1101	29
0001 1110	30
0001 1111	31
0010 0000	32
0010 0001	33
0010 0010	34
0010 0011	35
0010 0100	36
0010 0101	37
0010 0110	38
0010 0111	39
0010 1000	40
0010 1001	41
0010 1010	42
0010 1011	43

0010 1100	44
0010 1101	45
0010 1110	46
0010 1111	47
0011 0000	48
0011 0001	49
0011 0010	50
0011 0011	51
0011 0100	52
0011 0101	53
0011 0110	54
0011 0111	55
0011 1000	56
0011 1001	57
0011 1010	58
0011 1011	59
0011 1100	60
0011 1101	61
0011 1110	62
0011 1111	63

Test 3: Frequency of multiplexing

Method: A random, non specific number was loaded onto the binary switches and the enter button pressed. The picoscope was connected across B7 (as this output is connected to the NOT gate and the MOSFETs, so will be equal to the multiplexing frequency) and 0V.

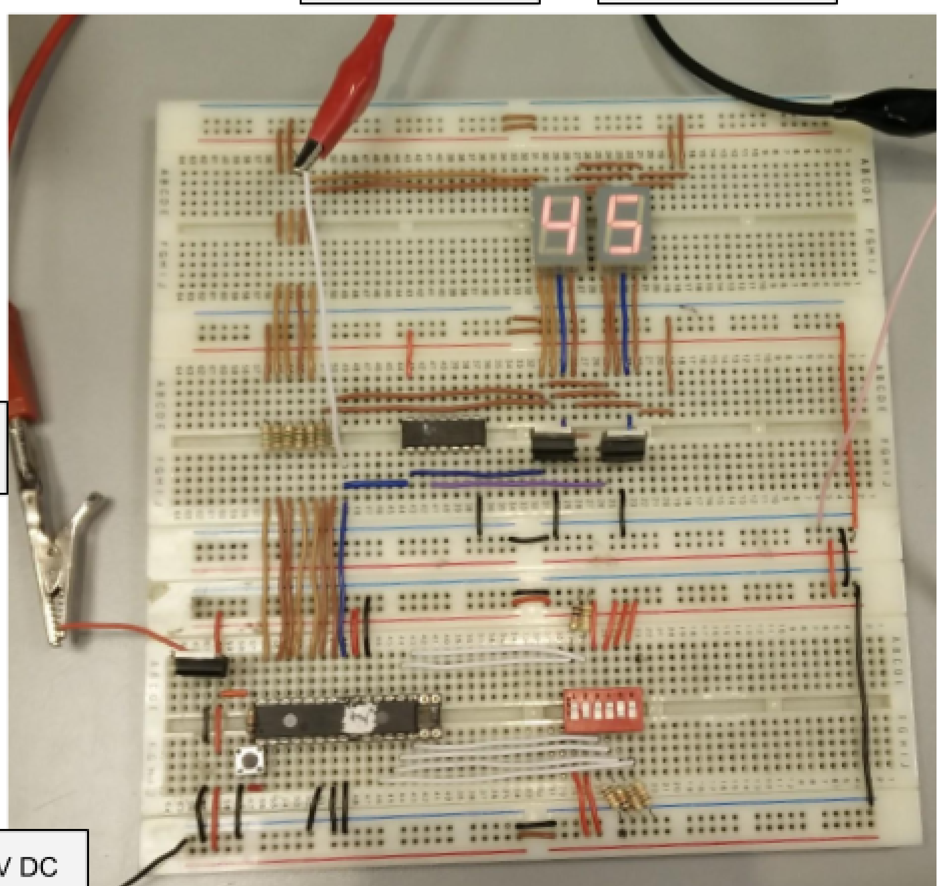
Equipment list: Circuit, Power supply, picoscope



To picoscope from B7

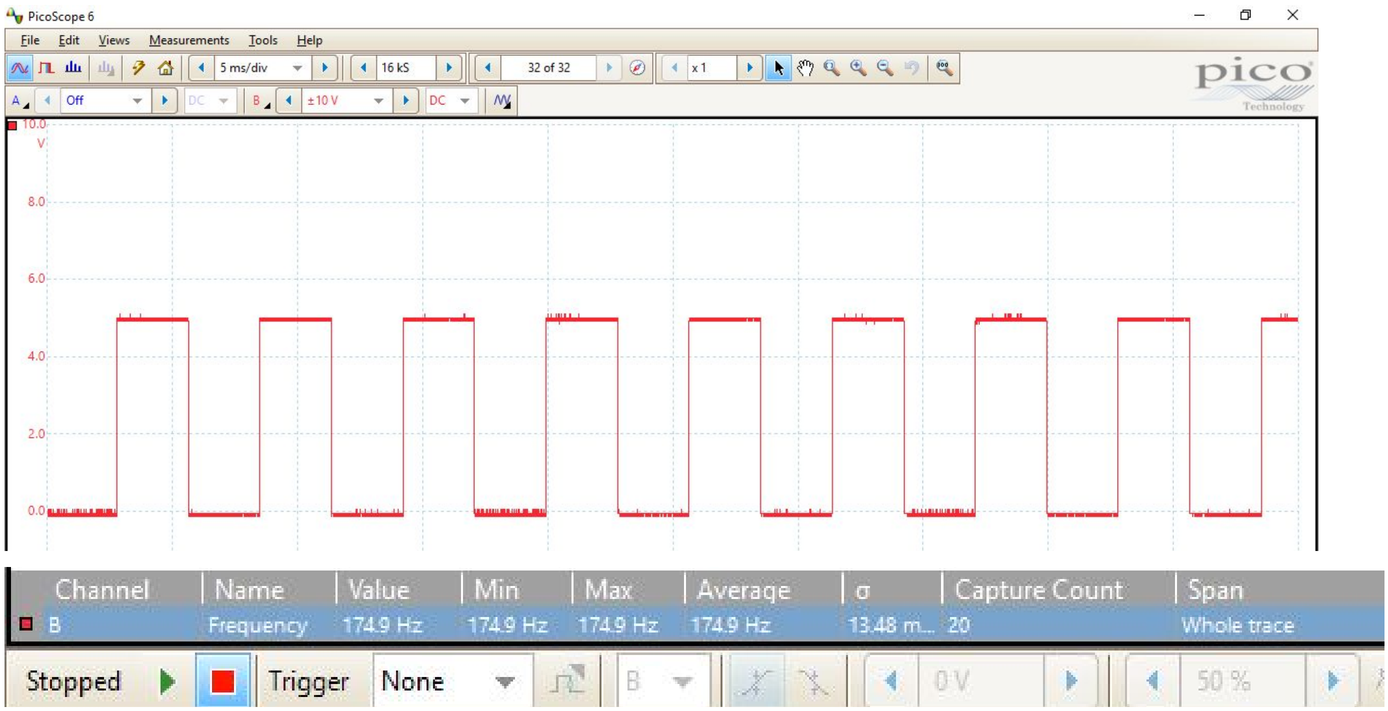
To picoscope, from 0V

To +9V DC supply



To +0V DC supply

<u>Graph to show frequency of multiplexing for the 28x1 chip in the circuit</u>



From this we can see the frequency is 174.9 Hz, a lot larger than the required frequency set by my specification of 60Hz, so I can be confident that this is operating to the required level.

<u>Bc)</u>

1. Multiplexing the 7-Segment displays

My project uses two seven segment displays, so in order to interface my 28x1 chip to my displays I would need to multiplex these displays so they appear to be on permanently. Multiplexing is where you turn a display on and off fast enough for it to appear permanently on to the human eye. A MOSFET can be used for each display to act as a switch to allow both displays to appear on permanently. In order to use only 1 chip, a NOT gate can be connected to the input of one MOSFET; this will invert the signal from the chip, so that both will appear on permanently whilst only using 1 chip.

One issue I discovered when trying to create the code for this problem, was that the display was not solid and noticeably flickered due to the program being quite long, causing issues multiplexing. To combat this issue I made use of the memory locations in the 28x1 chip and once the number had been ascertained by the chip, I stored the output value to light the specific number required for each side in a specific memory location and simply coded it to this:

```
LHS0:
MOVW        0xF7        ;Sets the specific pins high to show 0
MOVWR       B2          ;Stores this in memory location B2
JMP         RHS

LHS1:
MOVW        0xA1        ;Sets the specific pins high to show 1
MOVWR       B2          ;Stores this in memory location B2
JMP         RHS

(etc 0-9)....

RHS0:
MOVW        0xF6        ;Sets the specific pins high to show 1
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

RHS1:
MOVW        0xA0        ;Sets the specific pins high to show 2
MOVWR       B3          ;Stores this in memory location B3
JMP         DISR

(etc 0-9)....

DISL:
MOVRW       B2          ;Reads the value in memory location B2
MOVWR       PORTB       ;Moves this to PORTB to light the LHS LEDs
JMP         DISR        ;Jumps to display RHS

DISR:
MOVRW       B3          ;Reads the value in memory location B3
MOVWR       PORTB       ;Moves this to PORTB to light the RHS LEDs
JMP         DISL
```

Due to this there are now only 6 lines of code in my multiplex code, which means the frequency for multiplexing is much higher, achieving a solid display.

2. Polling for an input on the PIC chip

Code:

```
MOVW      0XFF        ;Make port C inputs
MOVWR     TRISC

Conversion:
MOVW      0x00
MOVWR     PORTB
MOVWR     B0          ;Clears values in PORTB & B0 to 0
MOVRW     PORTC       ;Read the values of the switches
MOVWR     B1          ;Stores the values of the switches
ANDW      0x20        ;Tests for a high on 00X0 0000 (highest input)
JPZ       CHECK4      ;If 0 jumps to test next highest input
MOVW      0x62        ;If 1 stores 32 in a special format:
MOVWR     B0          ;XXX|0|XXXX    Tens|Carry|Units
```

This is the only code I use to poll on my PIC chip; the rest of the code is simply repetition of this code to ascertain the number and formatting it with logic.

Initially, my circuit had a long response time, as it had to repeatedly poll for the input and go through the full code. To combat this I included an enter switch and so there is no need to repeatedly poll for an input, which therefore increases the reaction speed of the code as it is shorter and does not have to loop around infinitesimally. Also this means that the code only has to poll for an input once because it utilises the memory locations by storing the values in them, and then retrieving and editing these values rather than constantly polling the inputs.

To poll for this input I clear the values in PORTB and the memory location B0 to prevent any unwanted data tampering the system. The chip then reads the value of PORTC and stores it in memory location B0.

It then ANDs this value with 0x20 to test the highest possible bit (B0) 00X0 0000 for a high, and if it is 0, the number must be lower than 32 and so moves along the code to test the bit for 16 (000X 0000), then 8, then 4 etc. If 32 (or any of the tested numbers) is present it stores this in a special format where XXX|0|XXXX represents 10s, a carry bit, and units, so 32 would be stored as 0110 0010.

3. Voltage regulator

In order to interface my power supply to my 28x1 chip, I must use a regulated supply as they require a supply of between 3-5.5V to function properly and prevent damage, so I will use a voltage regulator. This is necessary as otherwise the chip will not function.

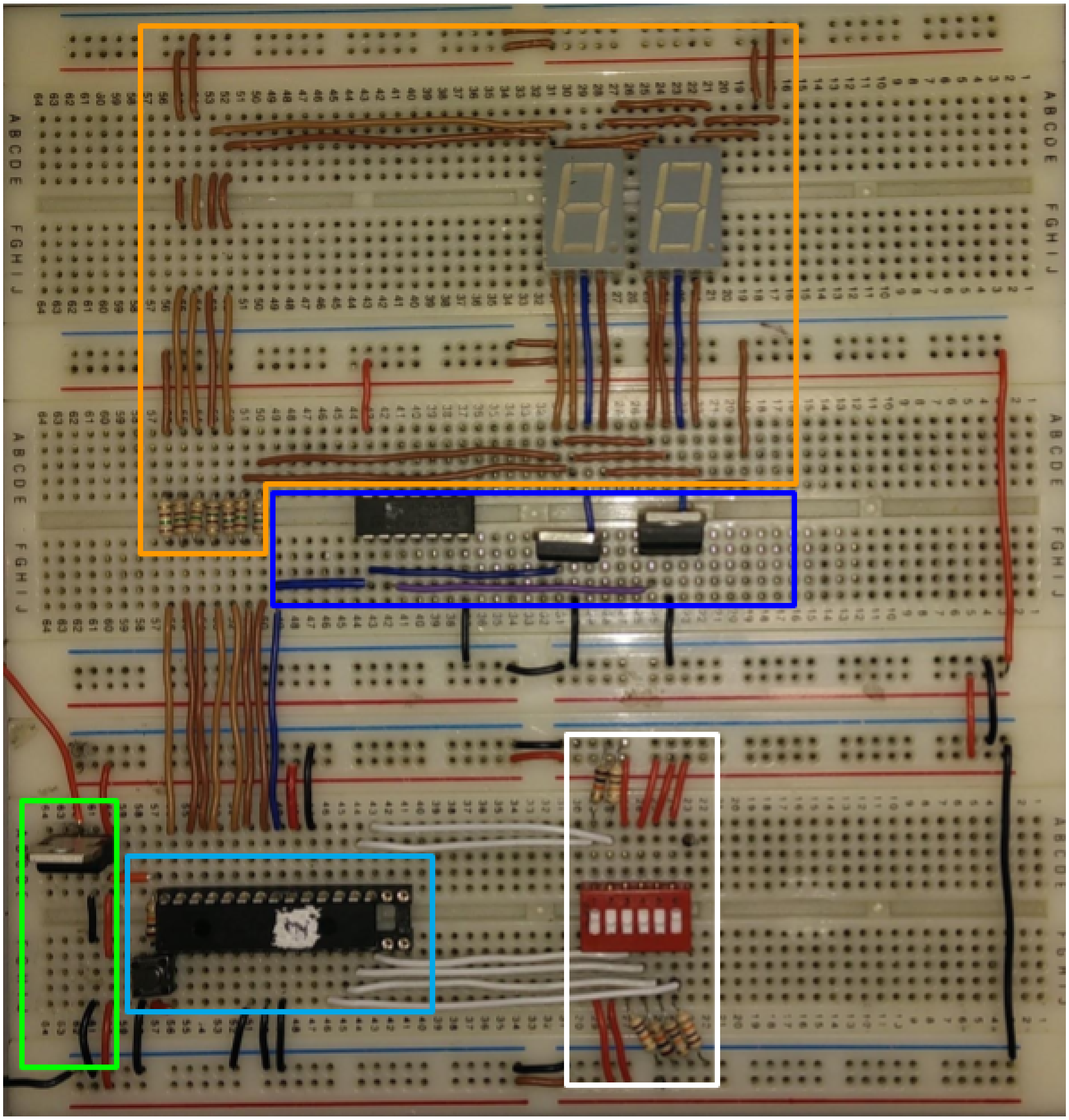
Bf) **Risk assessment - Building Prototype Circuit**

Key:

Potential Outcome	Value		Likelihood/Probability	Value		Value	Rating
Minor Injury	1	x	Unlikely	1	=	1-5	Low
Injury needing medical attention	2		Low possibility	2			
Injury – off work/College	3		Possible	3		6-11	Medium
Serious Injury/long term Sickness	4		Probable	4			
Fatality	5		Near Certainty	5		12+	High

Hazards identified	Persons concerned	Value	Risk	Control measures and significant findings to reduce the risk	Value	Risk
Overheating components	Builder Equipment	3x4= 12	High	Ensure components have protective resistors, or resistors to discharge static, should they be necessary.	1x1= 2	Low
Polarised components	Builder Equipment	3x5= 15	High	When wiring up polarised components, take care to do so correctly, as the electrolytic capacitors explode when wired incorrectly.	1x2= 2	Low
Contamination, Component damage	Builder Equipment	4x3= 12	High	Eating and drinking must be prohibited when doing lab and build work, as food may become contaminated and the drink may cause damage to sensitive electrical components.	3x1= 3	Low
Slips, trips and falls	Builder Others in lab	4x4= 16	High	All long wires, bags, etc must be moved into safety and out of the way, before starting any practical work.	4x1= 4	Low
Electrocution due to running project off mains	Builder Others in lab	5x5= 25	High	Ensure all coursework is run off a stable battery supply (eg 9V).	1x5= 5	Low
PIC chip	Equipment	1x5=5	Low	Ensure the PIC chip is run off a stabilised 5V, and ensure all other inputs/outputs are correctly connected before powering the circuit.	1x1=1	Low
Cuts	Builder Others in lab	3x4= 12	High	Use cutters/stippers appropriately.	2x2= 4	Low

Be/Bg)



Output - Light output system	**Working as intended**
Process - Buffer/logic system	**Working as intended**
Process - 28x1 Chip system	**Working as intended**
Regulator subsystem	**Working as intended**
Input - Switches subsystem	**Working as intended**

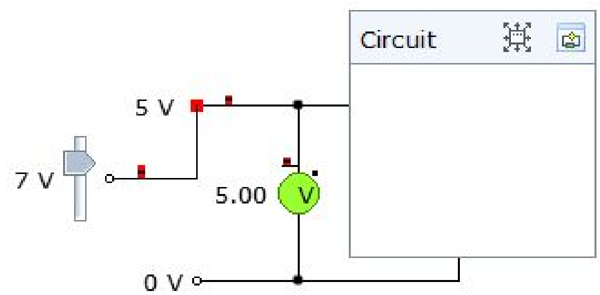
Bh)

I completed the construction of my circuit with no/minimal guidance from my electronics teacher (Richard Turkington). It all works as intended.

Ca)

**Note for ALL specification points & specifically numerical specification 3:
There is no need to test these parameters over the full supply range as I have tested that my regulators ability to supply a stable voltage over the supply range in the initial testing. I am using the same regulator, and everything in my circuit is supplied from the output of this. To further verify this assumption I will test the voltage regulator in the final circuit as well to ensure that this is supplying a stable voltage out when in operation.**

Method:
The voltage across the voltage regulator is measured over the supply range of 7-12V whilst loaded with the circuit active.



Equipment:
Circuit, Multimeter, Power supply, Wires.

Voltage in	Voltage out	Deviation
7		
8		
9		
10		
11		
12		

As all my tests require me to test over a full range of binary input values, I have the test results as a single table after each method explanation.

Testing for behaviour specification 1

Method: Each binary value is inputted on the switches and the number on the 7-Segment display recorded with a photograph to ensure the correct number is displayed.
Equipment: Circuit, Power supply, wires.

Test for behaviour specification 2: Visibility

Method: Each binary value is inputted on the switches and whether it is Visible, Solid, and Readable is recorded.

Equipment: Circuit, power supply, wires.

Test for numerical specification 1: Light output of the display

Method: In a dim room, all possible switch combinations are inputted, and the light intensity of the display output is recorded by placing a light meter above the 7-Segment display.

Equipment: Circuit, light meter, Power supply, Wires.

Table of results - Behaviour specification 1,2, Numerical specification 1.

Binary input	Expected output	Visual output	Visible/ Solid/ Readable (✔/✘)	Luminosity (lux)
0000 0000	00			
0000 0001	01			
0000 0010	02			
0000 0011	03			
0000 0100	04			
0000 0101	05			
0000 0110	06			
0000 0111	07			
0000 1000	08			
0000 1001	09			
0000 1010	10			
0000 1011	11			
0000 1100	12			
0000 1101	13			

0000 1110	14			
0000 1111	15			
0001 0000	16			
0001 0001	17			
0001 0010	18			
0001 0011	19			
0001 0100	20			
0001 0101	21			
0001 0110	22			
0001 0111	23			
0001 1000	24			
0001 1001	25			
0001 1010	26			
0001 1011	27			
0001 1100	28			
0001 1101	29			
0001 1110	30			
0001 1111	31			
0010 0000	32			
0010 0001	33			
0010 0010	34			
0010 0011	35			
0010 0100	36			
0010 0101	37			
0010 0110	38			
0010 0111	39			

0010 1000	40			
0010 1001	41			
0010 1010	42			
0010 1011	43			
0010 1100	44			
0010 1101	45			
0010 1110	46			
0010 1111	47			
0011 0000	48			
0011 0001	49			
0011 0010	50			
0011 0011	51			
0011 0100	52			
0011 0101	53			
0011 0110	54			
0011 0111	55			
0011 1000	56			
0011 1001	57			
0011 1010	58			
0011 1011	59			
0011 1100	60			
0011 1101	61			
0011 1110	62			
0011 1111	63			

Test for numerical specification 2: Response time

Method: The circuit will be switched on with the number 0 initially on the binary switches. The numbers 0, 33 and 63 (so as to see the range of operation time) will be loaded onto the switches in turn and the time taken between pressing the enter button and the output changing is recorded. In picoscope this is found by clicking on the line where the enter switch goes high, and the line where multiplexing begins (the output goes low initially). Port 1 is connected between the enter switch and resistor, and 0V. Port 2 is connected on gate of the MOSFET, and 0V.

Equipment: Circuit, Picoscope, Wires, Power supply.

Number	Time 1 (ms)	Time 2 (ms)	Response Time (ms)
0			
33			
63			

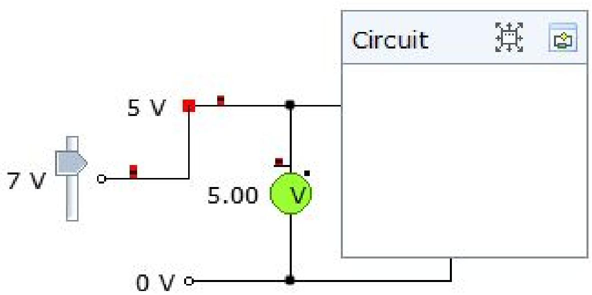
Cb)

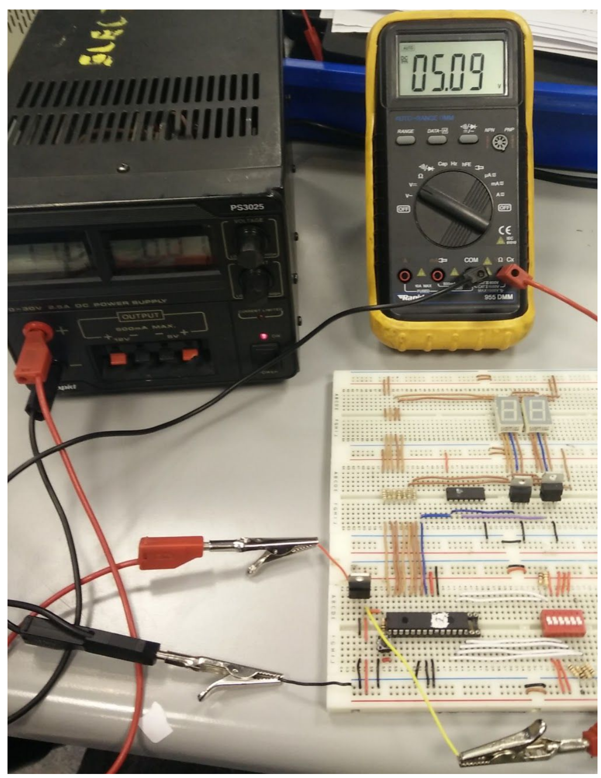
Note for ALL specification points & specifically numerical specification 3:
As I have tested my regulator to always supply a stable voltage over the supply range in the initial testing and everything in my circuit is supplied from the output of this, there is no need to test these parameters over the full supply range.
To further verify this assumption I will test the voltage regulator in the final circuit prior to other test procedures to ensure that this is supplying a stable voltage out when loaded.

Pre test: Voltage across a loaded voltage regulator to ensure reliable and stable operation.

Method: The voltage across the voltage regulator is measured over the supply range of 7-12V (as to stay in line with my specification) whilst loaded with the circuit active.

Equipment: Circuit, Multimeter, Power supply, Wires.



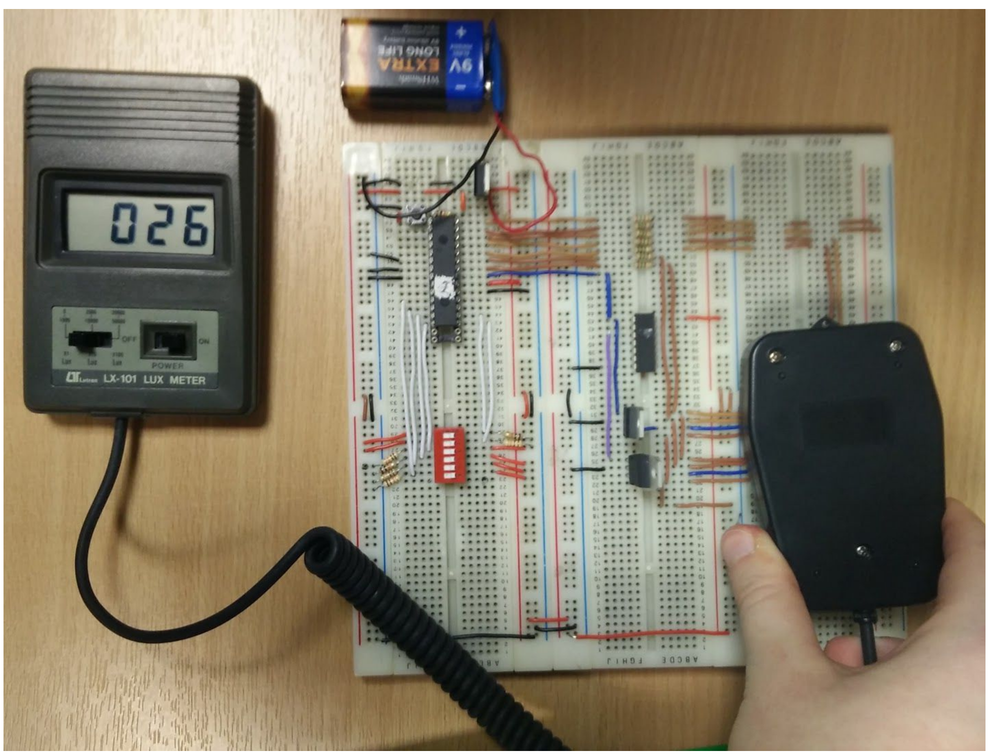


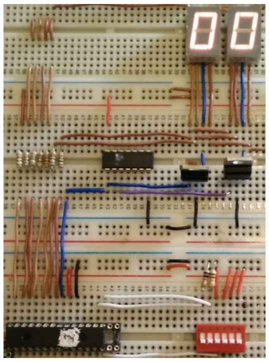
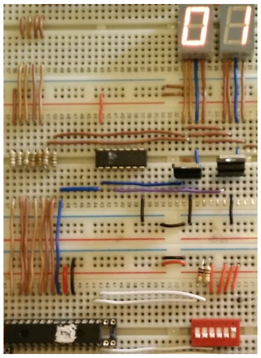
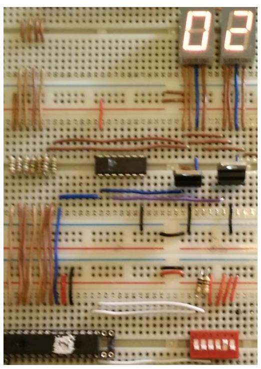
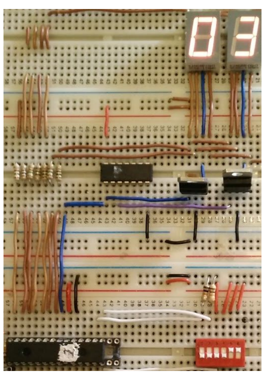
Voltage in	Voltage out	Deviation
7	5.09	+0.09
8	5.09	+0.09
9	5.09	+0.09
10	5.09	+0.09
11	5.09	+0.09
12	5.09	+0.09

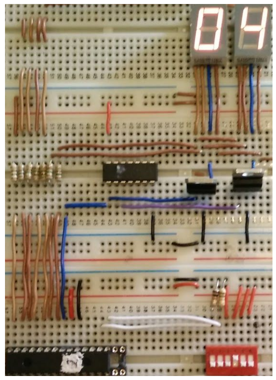
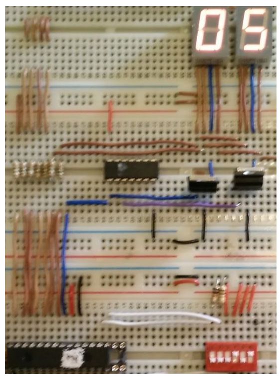
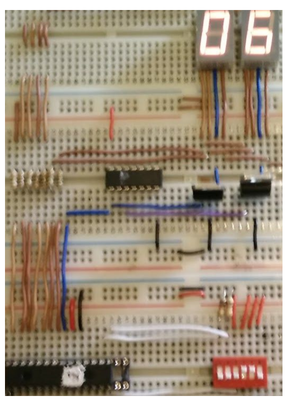
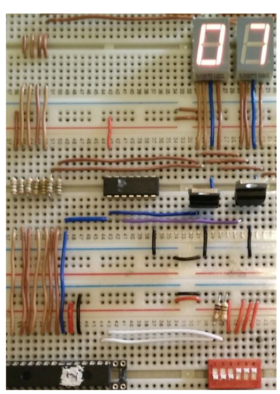
Testing for behaviour specification 1 & 2

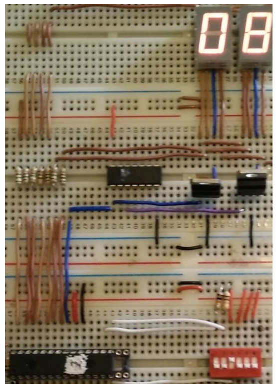
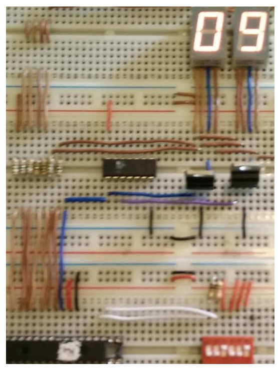
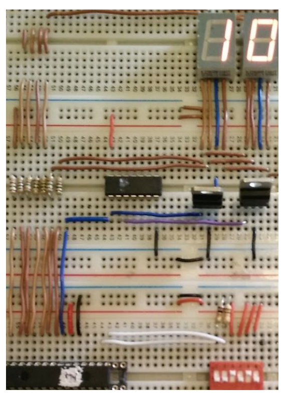
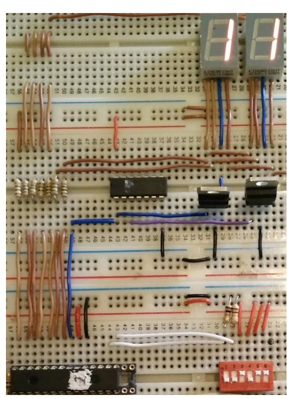
Method: Each binary value is inputted on the switches and the number on the 7-Segment display recorded with a photograph to ensure the correct number is displayed. For each number I write down if the display is solid, and measure the luminosity. For luminosity, as the room was dark to remove any background intensity, and the light meter did not contain a backlight, providing photo evidence of the readings is fairly problematic; however, I do include a picture of how the results were taken for evidence.

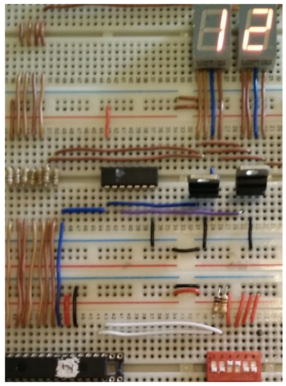
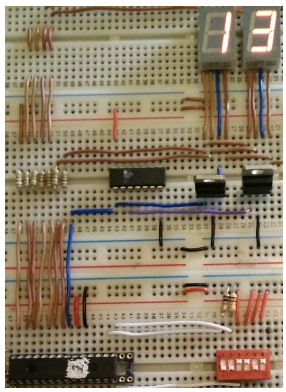
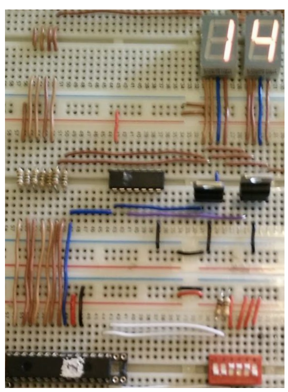
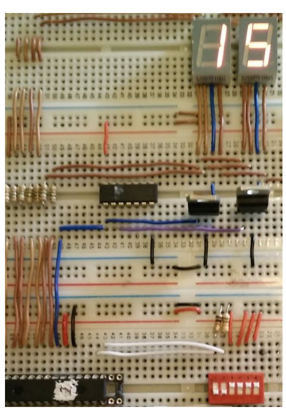
Equipment: Circuit, Power supply, Light meter.

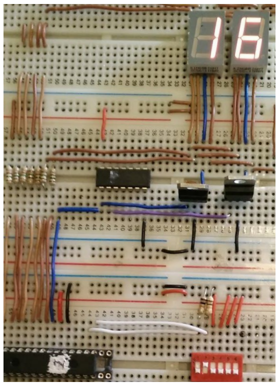
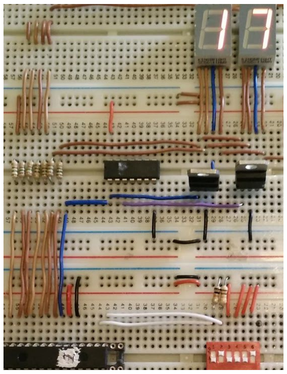
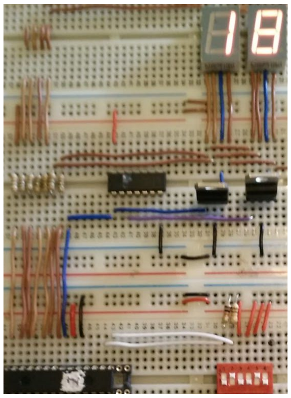
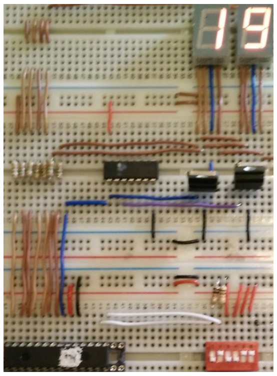


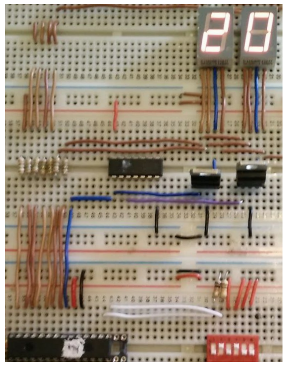
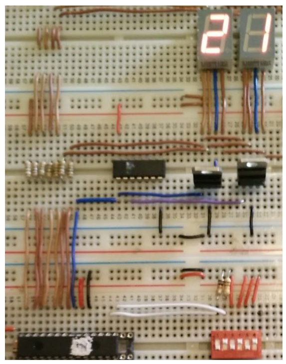
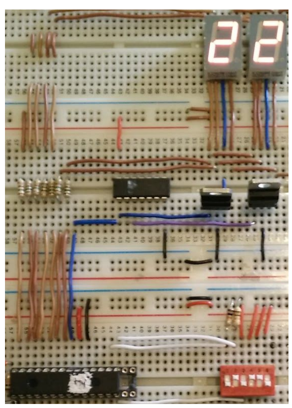
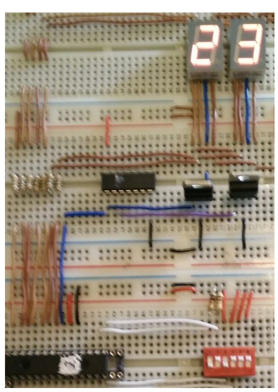
Binary input	Expected output	Actual output	Visible/Solid/ Readable	Luminosity (lux)
0000 0000	00		✔✔✔	37
0000 0001	01		✔✔✔	23
0000 0010	02		✔✔✔	33
0000 0011	03		✔✔✔	37

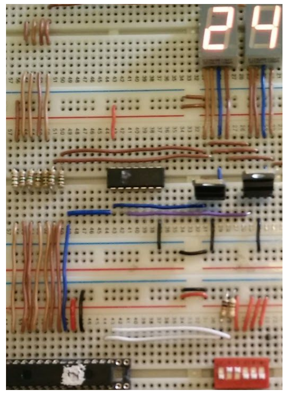
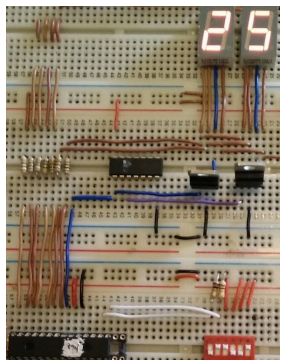
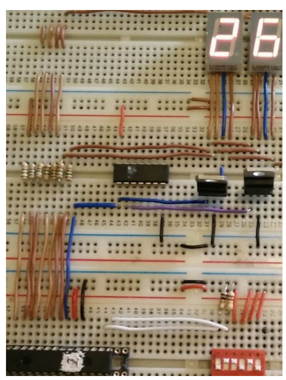
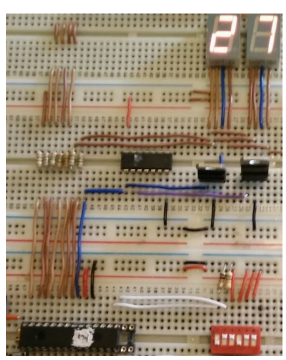
0000 0100	04		✔✔✔	35
0000 0101	05		✔✔✔	32
0000 0110	06		✔✔✔	29
0000 0111	07		✔✔✔	27

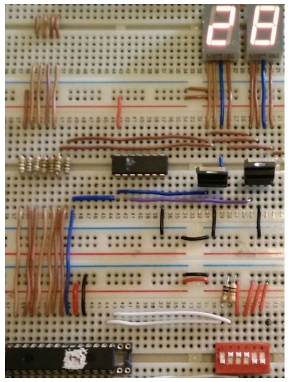
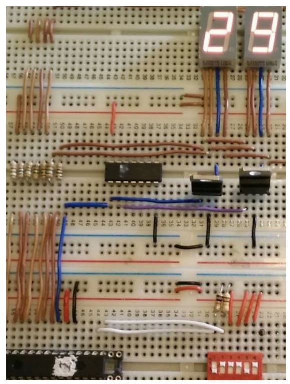
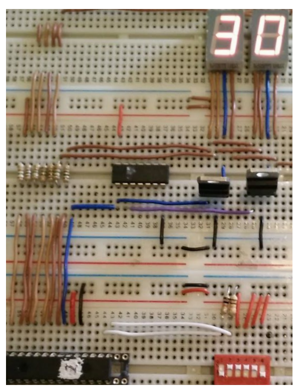
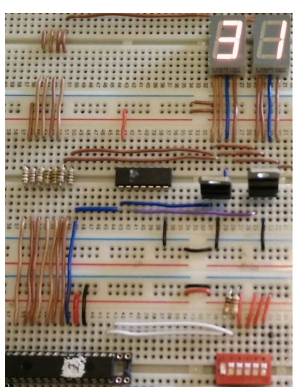
0000 1000	08		✔✔✔	44
0000 1001	09		✔✔✔	40
0000 1010	10		✔✔✔	27
0000 1011	11		✔✔✔	17

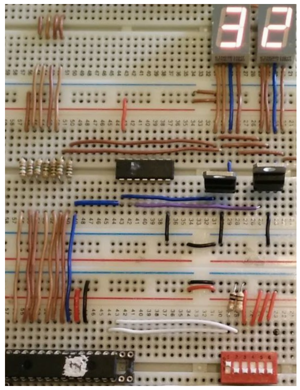
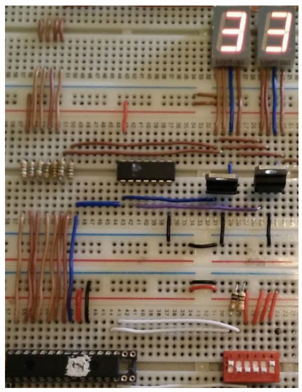
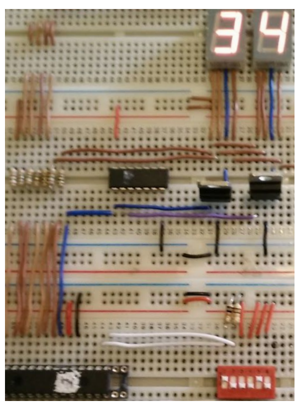
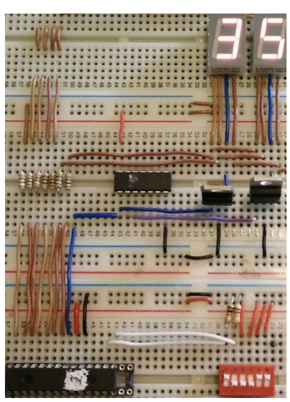
0000 1100	12		✔✔✔	27
0000 1101	13		✔✔✔	28
0000 1110	14		✔✔✔	23
0000 1111	15		✔✔✔	28

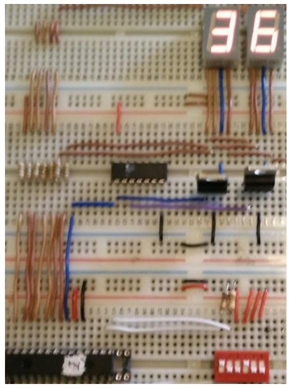
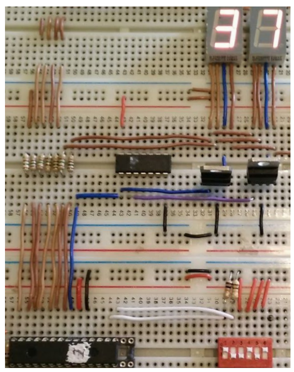
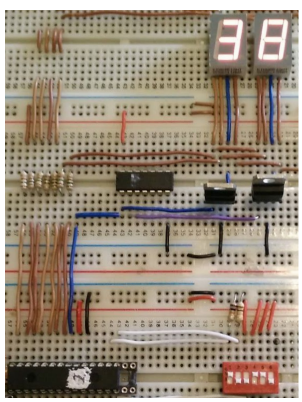
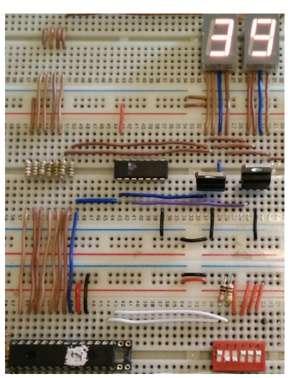
0001 0000	16		✔✔✔	30
0001 0001	17		✔✔✔	18
0001 0010	18		✔✔✔	35
0001 0011	19		✔✔✔	33

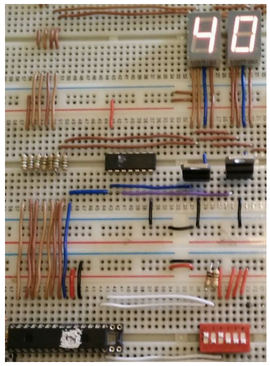
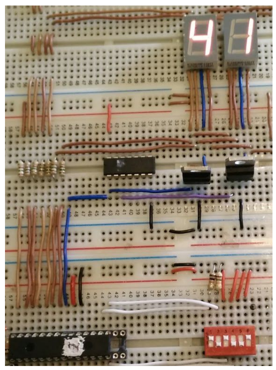
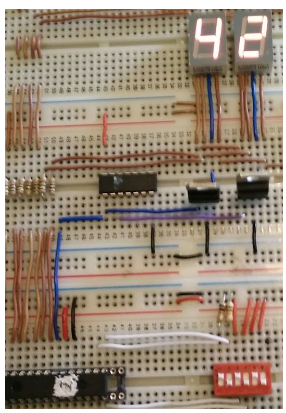
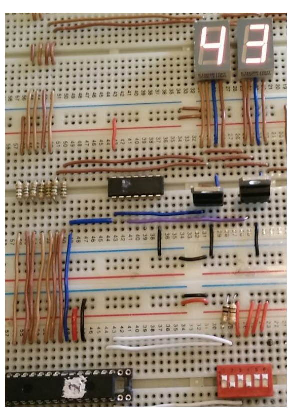
0001 0100	20		✔✔✔	33
0001 0101	21		✔✔✔	18
0001 0110	22		✔✔✔	32
0001 0111	23		✔✔✔	32

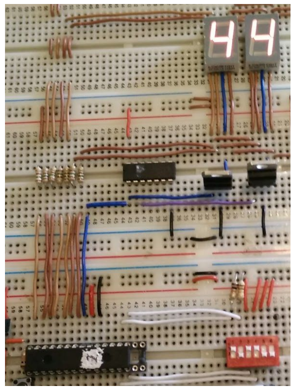
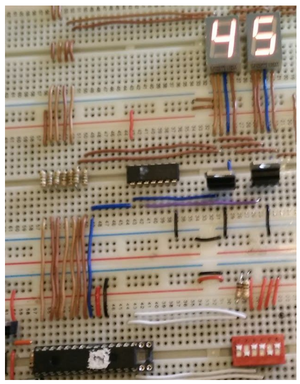
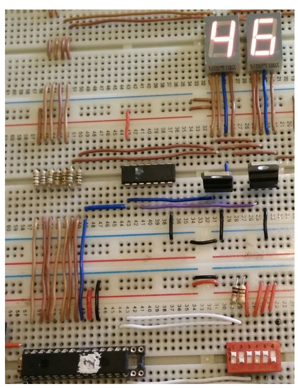
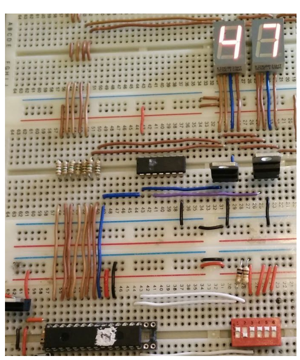
0001 1000	24		✔✔✔	34
0001 1001	25		✔✔✔	34
0001 1010	26		✔✔✔	39
0001 1011	27		✔✔✔	19

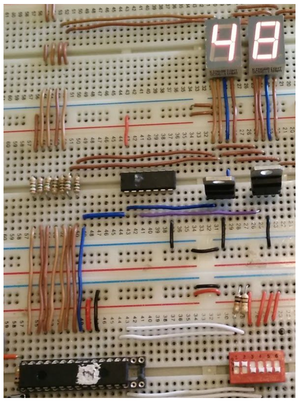
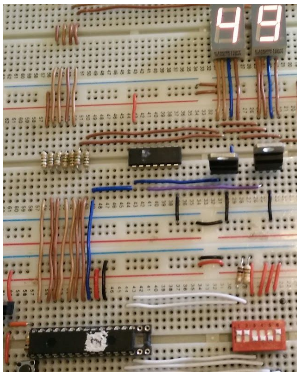
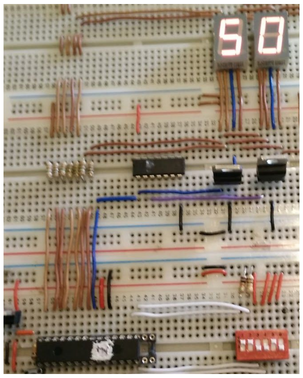
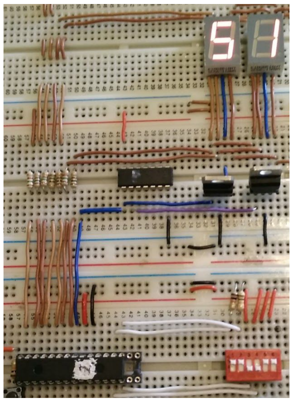
0001 1100	28		✔✔✔	35
0001 1101	29		✔✔✔	37
0001 1110	30		✔✔✔	26
0001 1111	31		✔✔✔	20

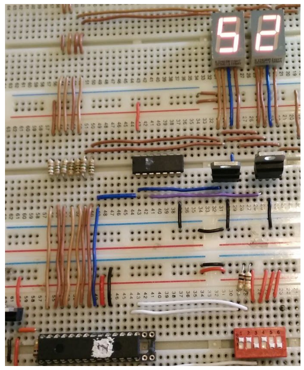
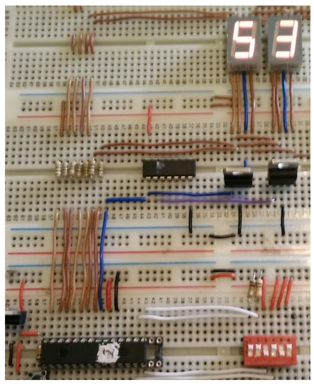
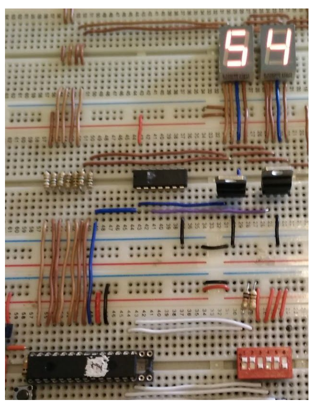
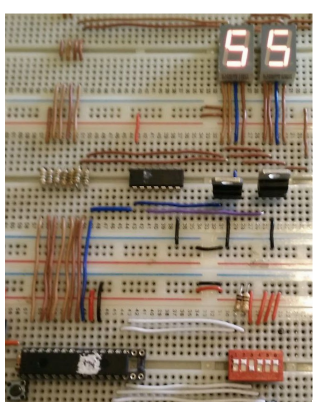
0010 0000	32		✔✔✔	33
0010 0001	33		✔✔✔	34
0010 0010	34		✔✔✔	35
0010 0011	35		✔✔✔	36

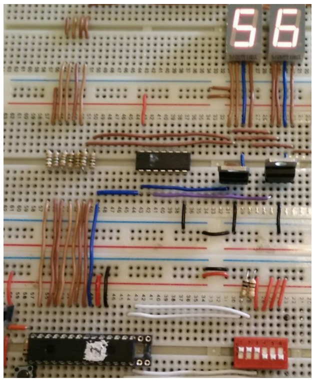
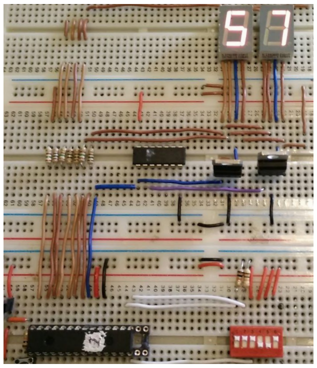
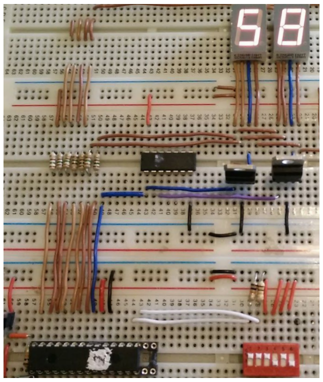
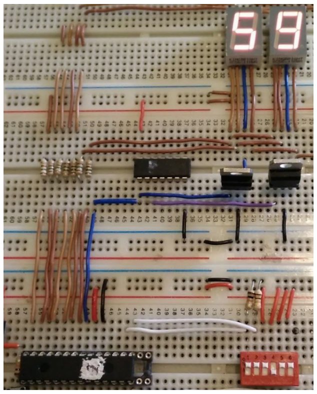
0010 0100	36		✔✔✔	34
0010 0101	37		✔✔✔	20
0010 0110	38		✔✔✔	41
0010 0111	39		✔✔✔	36

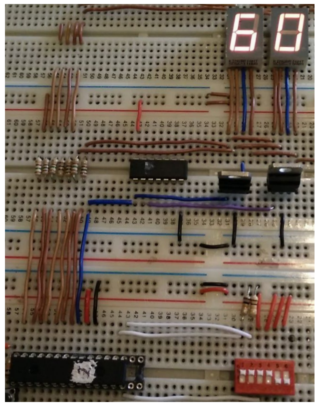
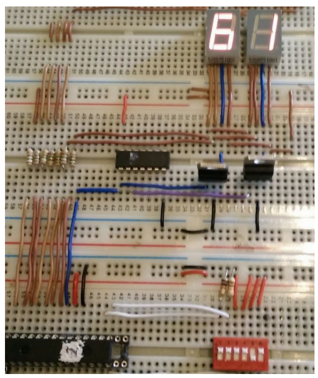
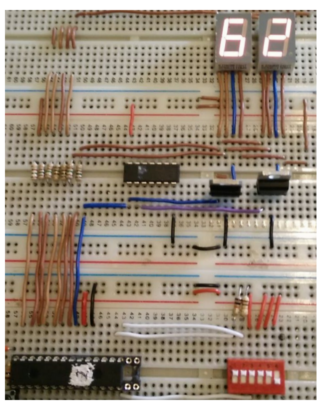
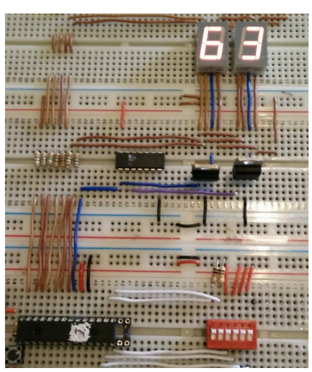
0010 1000	40		✔✔✔	32
0010 1001	41		✔✔✔	14
0010 1010	42		✔✔✔	29
0010 1011	43		✔✔✔	33

0010 1100	44		✔✔✔	31
0010 1101	45		✔✔✔	33
0010 1110	46		✔✔✔	33
0010 1111	47		✔✔✔	23

0011 0000	48		✔✔✔	38
0011 0001	49		✔✔✔	36
0011 0010	50		✔✔✔	29
0011 0011	51		✔✔✔	20

0011 0100	52		✔✔✔	32
0011 0101	53		✔✔✔	30
0011 0110	54		✔✔✔	28
0011 0111	55		✔✔✔	23

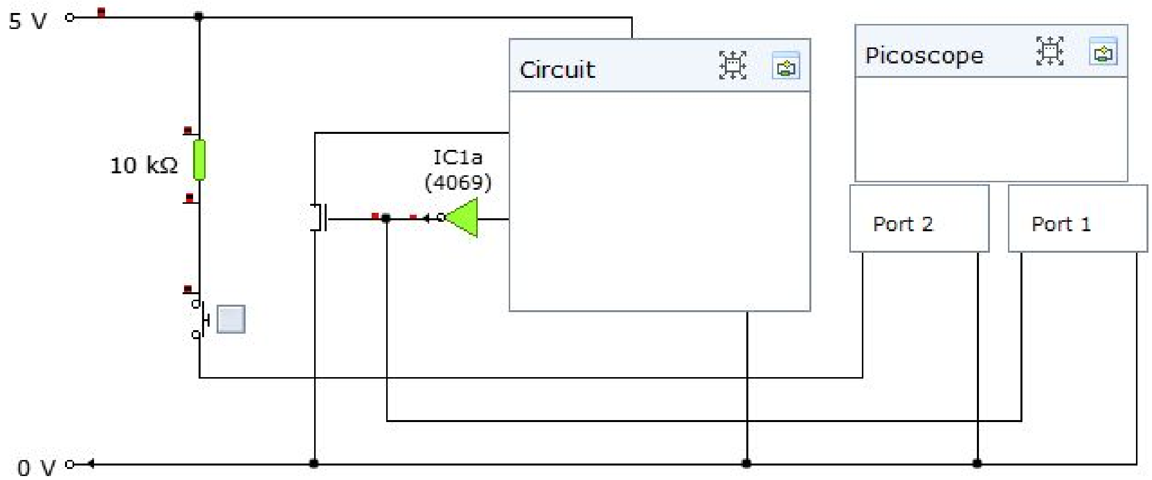
0011 1000	56		✔✔✔	37
0011 1001	57		✔✔✔	20
0011 1010	58		✔✔✔	39
0011 1011	59		✔✔✔	37

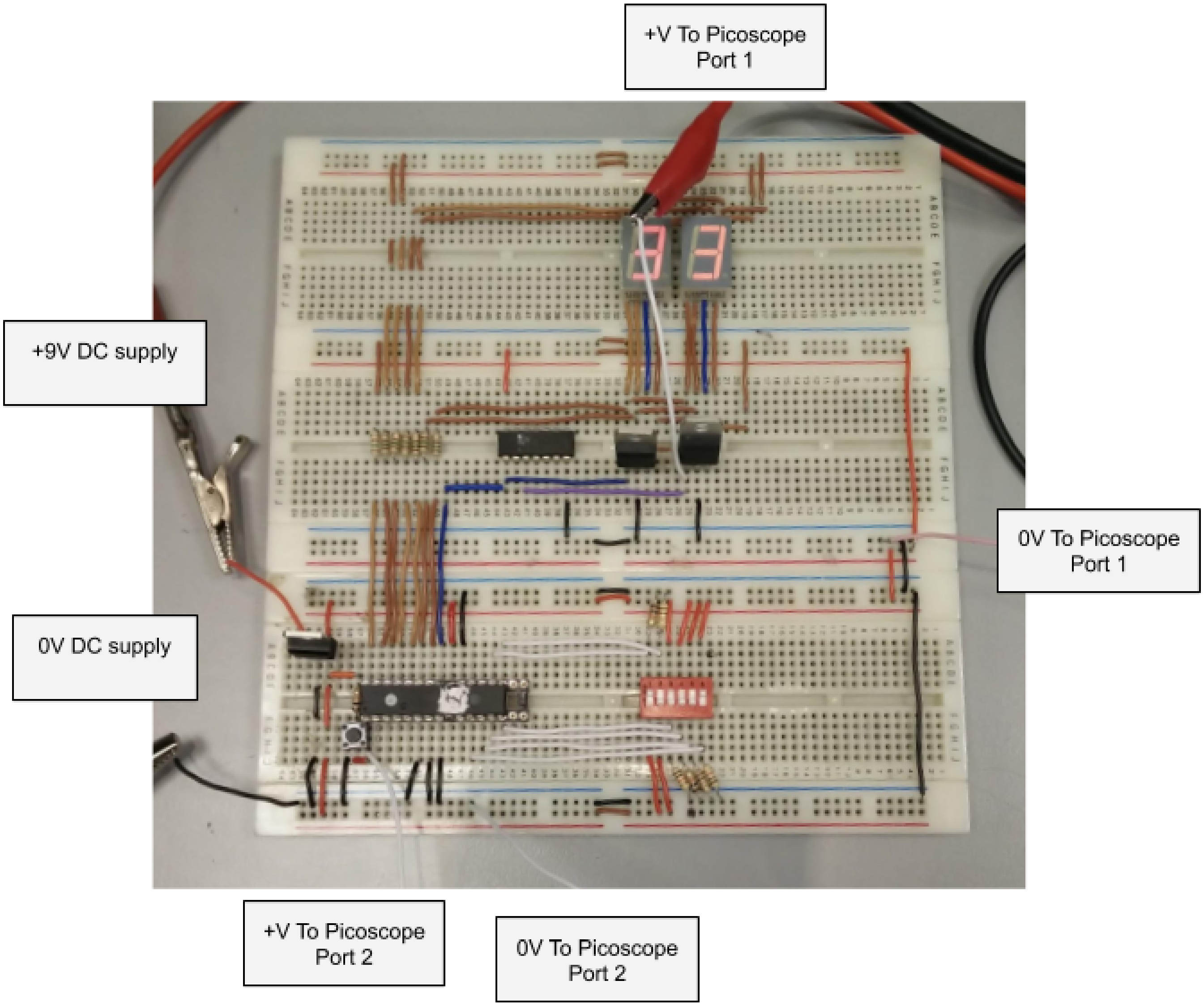
0011 1100	60		✔✔✔	34
0011 1101	61		✔✔✔	22
0011 1110	62		✔✔✔	35
0011 1111	63		✔✔✔	35

Test for numerical specification 2: Response time

Method: The circuit will be switched on with the number 0 initially on the binary switches. The numbers 0, 33 and 63 (so as to see the range of operation time) will be loaded onto the switches in turn and the time taken between pressing the enter button and the output changing is recorded. In picoscope this is found by clicking on the line where the enter switch goes high, and the line where multiplexing begins (the output goes low initially). Port 1 is connected between the enter switch and resistor, and 0V. Port 2 is connected on gate of the MOSFET, and 0V.

Equipment: Circuit, Picoscope, Wires, Power supply.

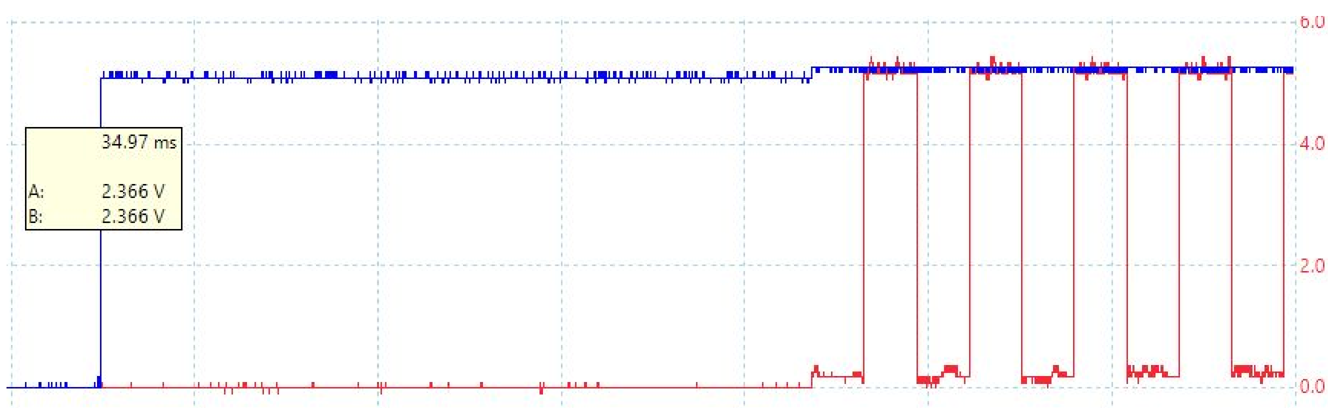



+V To Picoscope Port 1
+9V DC supply
0V To Picoscope Port 1
0V DC supply
+V To Picoscope Port 2
0V To Picoscope Port 2

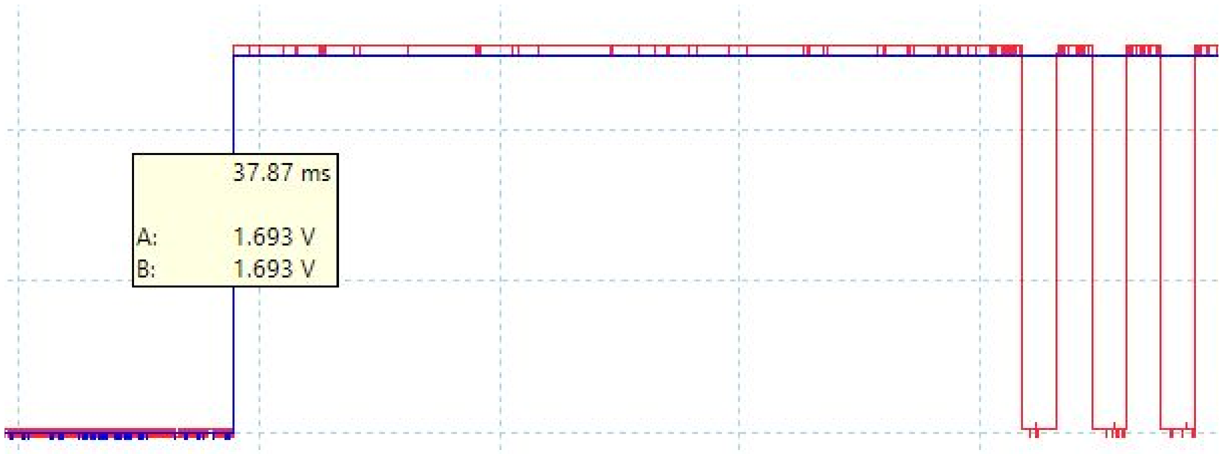
<u>Graph from picoscope showing response time of the 28x1 chip</u>

Key: **Red line** = Port 1 of picoscope (Vout on the gate of the MOSFET), **Blue line** = Port 2 of picoscope (Vout between the enter switch and resistor).

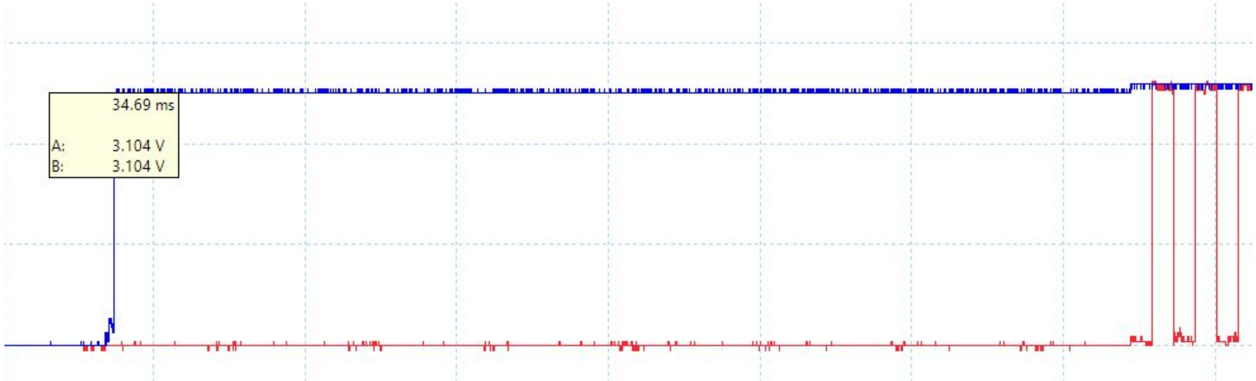
Number: 0

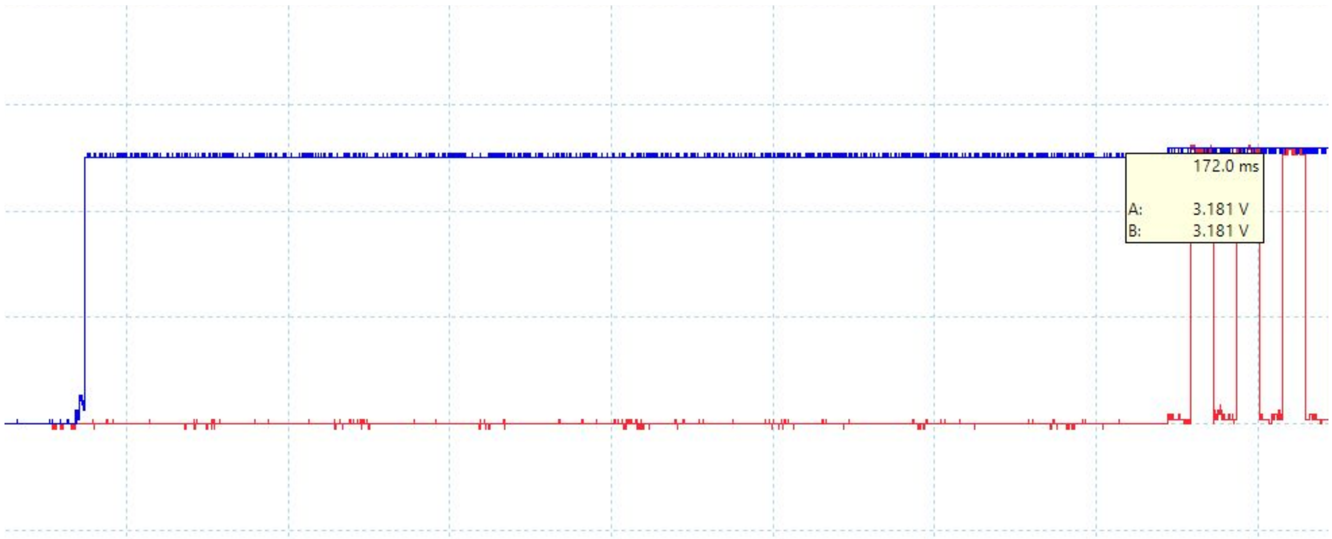


Number: 33



Number: 63





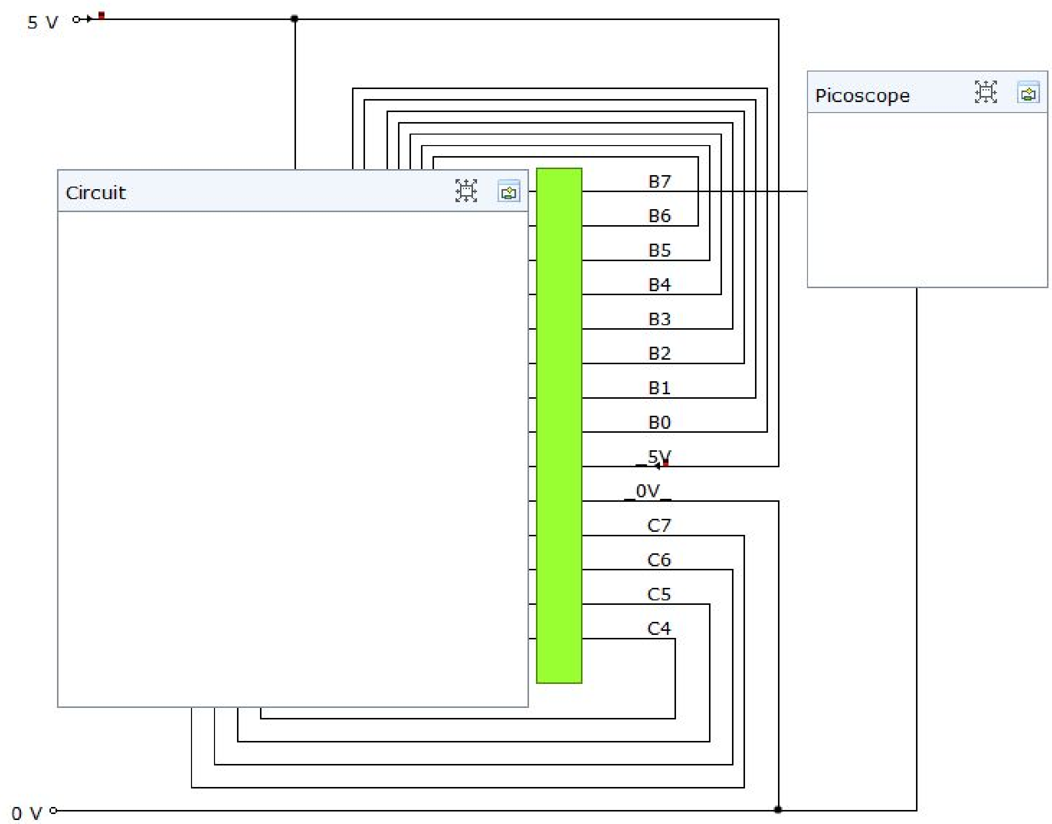
Number	Time 1 (ms)	Time 2 (ms)	Response Time (ms)
0	34.97	76.47	41.5
33	37.85	103.5	65.7
63	34.69	172.0	137.3

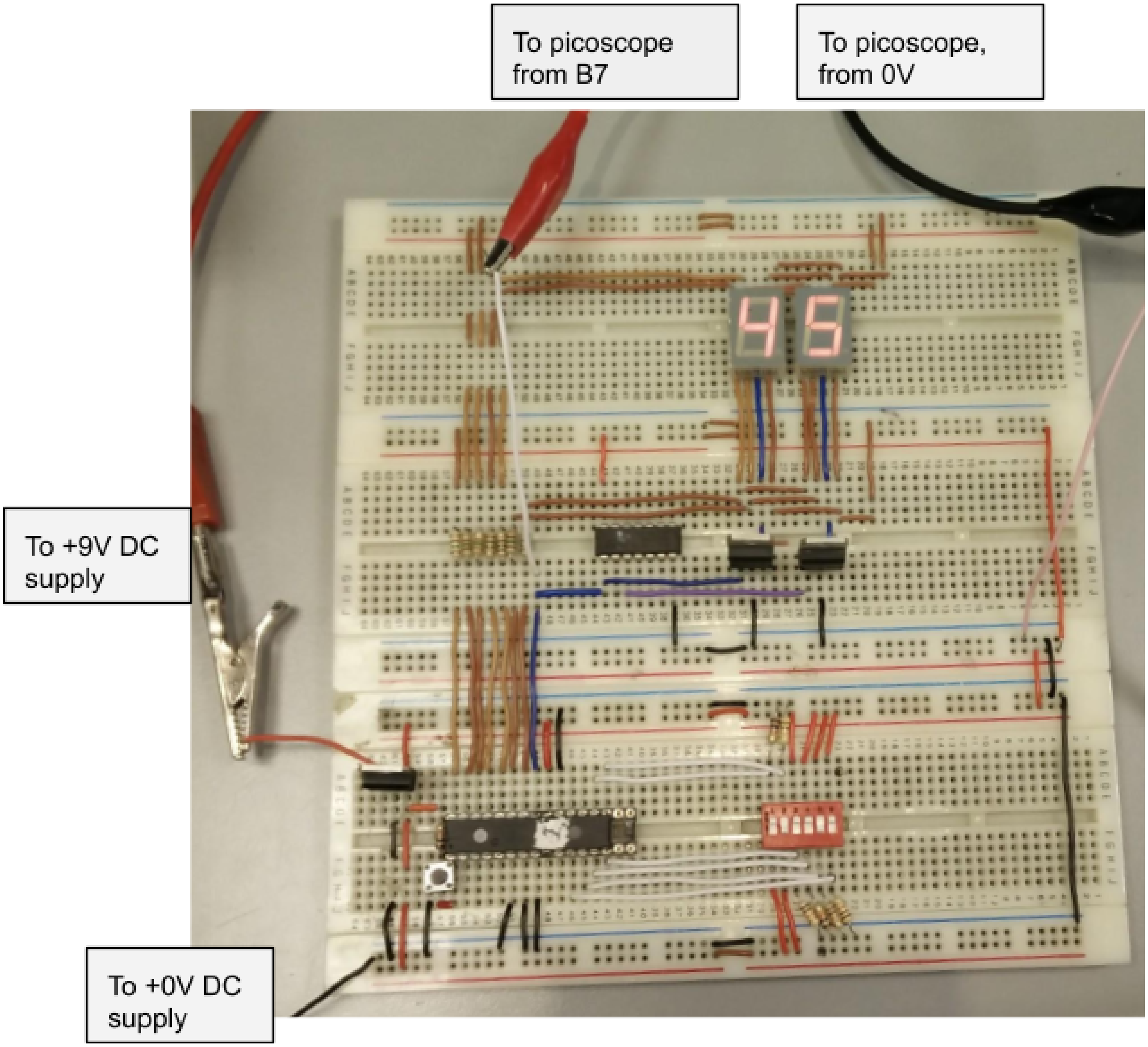
From this we can see that my response time range is 41.5 - 137.3 ms. According to my specification my circuit must respond in at least one second (1000 ms) as at this time the user feels they are interacting with the system. At 100ms the user perceives the response as instantaneous. The upper limit of my response time is less than the maximum value of 1 second, therefore I can say my circuit meets the requirements of the specification.

Test 3: Frequency of multiplexing

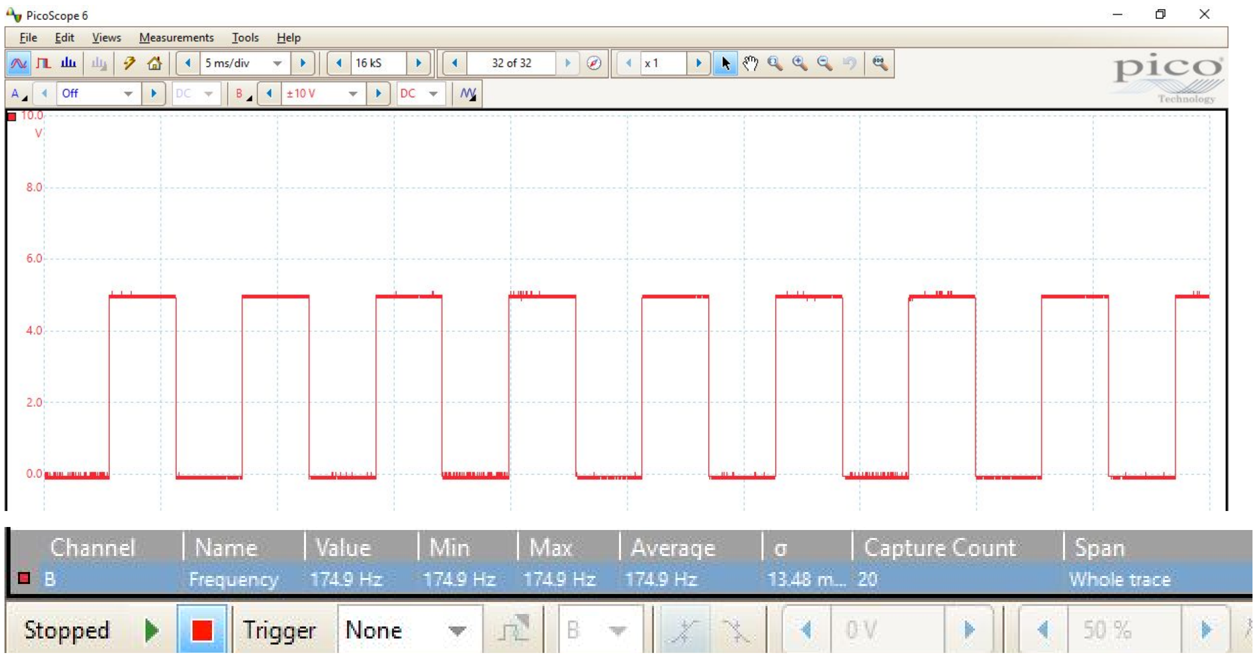
Method: A random, non specific number was loaded onto the binary switches and the enter button pressed. The picoscope was connected across B7 (as this output is connected to the NOT gate and the MOSFETs, so will be equal to the multiplexing frequency) and 0V.

Equipment list: Circuit, Power supply, Picoscope.




To picoscope from B7
To picoscope, from 0V
To +9V DC supply
To +0V DC supply

Graph to show frequency of multiplexing



From this we can see the frequency is 174.9 Hz, a lot larger than the required frequency set by my specification of 60Hz, so I can be confident that this is operating to the required level.

Cc)

1. Picoscope

The picoscope has in input impedance of 1MΩ (±1%), and an error of ±3% of the full scale (±200 μV). However I have used the picoscope time scale, which has no quoted error; therefore since there must be some error, I assumed an error of ±1ms on the time scale. From this I can find a range of reaction times for my circuit by taking the upper value + upper error, and the lower value − lower error. This gave me the final range of: 40.5 - 138.3 ms.

The frequency was internally computed by the picoscope and so the error on the frequency is negligible.

2. Multimeters[20]

Voltmeter has a stated error of 0.5% (±3 digits). The input impedance of the multimeter is 10MΩ. It is important that the multimeters have a high input impedance because they can only give accurate readings if their impedance is high.

In section Ac and Cb, it was found that my regulator stabilised at a measured 5.1V both isolated and in the circuit. The error in this value is:

$$Error = \frac{0.5}{100} \times 5.1V = 0.0255V$$

$$Resolution\ error = 1 \times 0.001V = 0.001V$$

$$Total\ error = 0.0255 + 0.001 = 0.0265V$$

So the final value of the voltage out of the regulator is 5.1 ± 0.0265 V (5.0735 to 5.1265 V) The picaxe runs on a supply of 3 to 5.5V DC (as mentioned in section Ab above), so my regulator is within the required range for my circuit to operate safely and properly.

3. Light meters[21]

I found the following accuracy of the light meters I used on a datasheet online. Using this I calculated the error for the minimum and maximum recorded values (14 and 44 lux).

ELECTRICAL SPECIFICATIONS (23± 5 ℃)		
Range	Resolution	Accuracy
2,000 Lux	1 Lux	± (5 % + 2 d)
20,000 Lux	10 Lux	± (5 % + 2 d)
50,000 Lux	100 Lux	± (5 % + 2 d)
Note : Accuracy tested by a standard parallel light tungsten lamp of 2856K temperature.		

[20] https://www.rapidonline.com/pdf/90-6561.pdf

[21] http://www.instrumentsgroup.co.za/index_files/Lutron/database/pdf/LX-101.pdf

Lower error: $\pm(14 \times 0.05) + 2 = \pm 2.7$

Upper error: $\pm(44 \times 0.05) + 2 = \pm 4.2$

From this I can find a range of luminosities for my display by taking the upper value + upper error, and the lower value − lower error. This gave me the final range of: 11.3 - 48.2 lux.

Additionally, it was fairly difficult to put the light meter in exactly the same place every time, and the reader's area was slightly larger than my display, so moving it even slightly would alter the reading, so there is some methodical error in place here too. However, I made sure that I always put the light meter in the same place by putting a whiteboard mark on the circuit that I used to align the light meter to, reducing this error.

4. Frequency generator

The Jupiter 2000 function generator has a minimum frequency increment of 0.05×10^n. The error will therefore be ½ an increment: 0.025×10^n. In my initial tests and for my specification I found that the minimum frequency for the multiplexed displays to appear solidly was 60Hz. Accounting for the error, this value becomes: 60 ± 0.25 Hz.

Cd/Da/Db)

Original problem:
When learning electronics/computer science people often find it challenging to practise and convert binary to decimal.

Solution:
I will build a system that converts binary to decimal so that they can practise.

Specification

Behaviour specification:

1. My circuit will read a 6 bit binary input and output the corresponding two digit decimal output on a display.
2. My display will be visible, solid and easily read.

Numerical specification
1. My display will have a light output of at least 7 lux.
2. My circuit will respond to a change in input in less than 1 second.
3. Must multiplex at a rate of at least 60Hz.
4. My circuit will operate fully to the behaviour specification and achieve my other parameters over a supply voltage of 7-12V.

Justification
My display will have a light output of at least 7 lux measured immediately above the display so that it is bright enough to see in darkness, but not so bright that it is hard to see in lit settings (see section Ac, test for minimum brightness).

My circuit will respond in at least 1 second so that users do not have wait too long for the output to be shown, and so the operation appears smooth; according to studies, a response time 100ms is perceived as instantaneous and a response of 1 second or less is enough for users to feel that they are freely interacting with the information[22].

My circuit must have a multiplex frequency of at least 60Hz, as my initial research tests indicate this is the minimum frequency to be solidly visible.

My circuit will operate from a supply voltage range of 7-12V because the regulator needs a minimum of 7V to output a stable 5V supply to the circuit, and up to 12V because the circuit is intended to be of use to teachers/students who will only have access to domestic power supplies which do not often exceed 12V (for example 9V batteries are readily available).

[22] https://www.pubnub.com/blog/2015-02-09-how-fast-is-realtime-human-perception-and-technology/

Parameter 1:
My circuit will read a 6 bit binary input and output the corresponding two digit decimal output on a display.

Measurement of parameter 1
The visual outputs of each binary input were photographed, and can be viewed in section Cb.

Assessment of measurement of parameter 1
All photographs indicate that the correct decimal output was given for the corresponding input for each binary value, which therefore meets my specification.

Discussion of fitness of purpose for parameter 1
The problem was that when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. Since all binary inputs have been shown to correctly map to the corresponding decimal value output, this specification point has been met; therefore, I now have a system that converts binary to decimal, which will aid those in education and learning.

Parameter 2:
My display will be visible, solid and easily read.

Measurement of parameter 2:
For each decimal output, I recorded whether the display was visible, solid and readable.

Assessment of measurement of parameter 2
In section Cb all boxes have been ticked, and the pictures of the display outputs provide evidence for this, indicating that the system was visible solid and readable for all input and output values. Therefore my solution meets my specification.

Discussion of fitness of purpose for parameter 2
The problem was that when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. In order for this to be useful to the user, the output must be easily discernible; therefore this specification must be met for my solution to be complete. My tests indicate that my system is visible, solid and easily read across all inputs and outputs; therefore, the user of the system can now use this without straining their eyes, or being put under unnecessary pressure in other visible ways.

Numerical parameter 1:
My display will have a light output of at least 7 lux.

Measurement of numerical parameter 1:
I measured the visual output of my circuit to be a range of 14 - 44 lux.

Assessment of measurement of numerical parameter 1
Using the datasheet for my lightmeter I found an error of (5% ± 2d), which gave me a range of 11.3 - 48.2 lux. I have met numerical parameter 1, as a measurement of 11.3 - 48.2 lux lies within my specified range of at least 7 lux.

Discussion of fitness for purpose for parameter 1
The problem was that, when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. If my circuit was not visible then it would be useless to the user, and therefore not fit for purpose. Assuming that this could be used under any light level, for convenience of the user, it must have a certain light value output in order to be seen in dark and light. My display will have a light output of at least 7 lux measured immediately above the display so that it is bright enough to see in darkness, but not so bright that it is hard to see in lit settings.

Numerical parameter 2:
My circuit must respond to a change in input in less than 1s.

Measurement of numerical parameter 1:
I measured my circuit to respond to a change in input between 41.5 - 137.3 ms.

Assessment of measurement of numerical parameter 1
I have met numerical parameter 2, as my measurement had a range of 40.5 - 138.3 ms. The upper limit of this is less than the 1s required, and so can safely be said to lie within my specification of less than 1s.

Discussion of fitness for purpose for parameter 1
The problem was that when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. If my circuit did not respond in a quick time period, people would get frustrated and my solution would not be optimal in aiding learning. Therefore, my circuit will respond in at least one second so that users do not have wait too long for the output to be shown, and so the operation appears smooth; according to studies, a response time of 1 second or less is enough for users to feel that they are freely interacting with the information, and a response time 100ms is perceived as instantaneous.

Numerical parameter 3:
My circuit must must multiplex at a rate of at least 60Hz

Measurement of numerical parameter 1:
I measured the frequency of multiplexing on my circuit to be 174.9 Hz, with negligible error.

Assessment of measurement of numerical parameter 1
I have met numerical parameter 3, as my measurement was 174.9 Hz, which is greater than the required 60Hz, so meets my specification well.

Discussion of fitness for purpose for parameter 1
The problem was that when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. If my circuit were to flicker or not appear solid, it my cause irritation to user's eyes and focus, as well as patience. Therefore my circuit must have a multiplex frequency of at least 60Hz, as my initial research tests indicate this is the minimum frequency to be perceived as solidly visible.

Numerical parameter 4:
My circuit will operate fully to the behaviour specification and achieve my other parameters over a supply voltage of 7-12V

Measurement of numerical parameter 1:
As seen in my initial tests and in section Ca, my voltage regulator outputs a stable voltage of 5V over the supply voltage range of 7-12V. Everything in my circuit has been shown to work from the output of the regulator, giving a 5V DC supply.

Assessment of measurement of numerical parameter 1
I used multimeters to measure the voltage across the regulator; in section Cc I found the error on this to be: 5.1 ± 0.0265 V. The picaxe runs on a supply of 3 to 5.5V DC (as mentioned in section Ab), so my regulator is within the required range for my circuit to operate safely and properly. Since my circuit has been shown to work from the regulator over a supply range of 7-12V, this meets my specification.

Discussion of fitness for purpose for parameter 1
The problem was that when learning binary, people become confused about converting between the number systems binary and decimal. I intended to create a system to convert binary into decimal so that people could practise. My circuit must be usable by the average person, as well as consistent with my voltage regulator needs. Therefore, my circuit will operate from a supply voltage range of 7-12V because the regulator needs a minimum of 7V to output a stable 5V supply to the circuit, and up to 12V because the circuit is intended to be of use to teachers/students who will only have access to domestic power supplies which do not often exceed 12V.

Dc)

Bibliography

http://www.kingbrightusa.com/images/catalog/SPEC/SC56-11SRWA.pdf (Last accessed 26/04/2018)
https://en.wikipedia.org/wiki/Liquid-crystal_display (Last accessed 26/04/2018)
https://www.usbid.com/assets/datasheets/62/tc07.pdf (Last accessed 26/04/2018)
http://www.picaxe.com/docs/picaxe_manual1.pdf (Last accessed 26/04/2018)
http://www.picaxe.com/docs/picaxe_ocr.pdf (Last accessed 26/04/2018)
https://www.mouser.com/ds/2/389/l78-974043.pdf (Last accessed 26/04/2018)
https://www.pubnub.com/blog/2015-02-09-how-fast-is-realtime-human-perception-and-technology/ (Last accessed 26/04/2018)
https://www.rapidonline.com/picaxe-28x1-chip-13-0862 (Last accessed 26/04/2018)
https://www.rapidonline.com/ti-cd4069ube-hex-inverter-83-0398 (Last accessed 26/04/2018)
https://www.rapidonline.com/catalogue/search?Query=BC109%20npn%20transistor (Last accessed 26/04/2018)
https://www.rapidonline.com/truohm-knp05sj-1k-5-5w-wire-wound-resistor-51-8995 (Last accessed 26/04/2018)
https://www.rapidonline.com/ti-cd4069ube-hex-inverter-83-0398 (Last accessed 26/04/2018)
https://www.rapidonline.com/st-irf630-mosfet-n-200v-9a-47-0290 (Last accessed 26/04/2018)
https://www.electronicproducts.com/Analog_Mixed_Signal_ICs/Discrete_Power_Transistors/MOSFET_vs_IGBT.aspx (Last accessed 26/04/2018)
https://www.usbid.com/assets/datasheets/62/tc07.pdf (Last accessed 26/04/2018)
https://www.rapidonline.com/pdf/90-6561.pdf (Last accessed 26/04/2018)
http://www.instrumentsgroup.co.za/index_files/Lutron/database/pdf/LX-101.pdf (Last accessed 26/04/2018)

Acknowledgements:

During this project I received guidance from Richard Turkington (Electronics teacher) and Stan Lucas (Department technician).

I also used the IKES support booklets, by Ian Kemp, for guidance during this project.