

optical human interfacing using visible light (the light frame)

by Elis Rhys Hussey

FILE	7705_ELIS_RHYS_HUSSEY_OPTICAL_HUMAN_INTERFACING_USING_VISIBLE_LIGHT__THE_LIGHT_FRAME__78227_447133640.DOCX (4.46M)		
TIME SUBMITTED	21-APR-2016 03:51PM	WORD COUNT	7865
SUBMISSION ID	56214208	CHARACTER COUNT	48929



Optical Human Interfacing
Using Visible Light
(The Light Frame)

By

Elis Hussey

Supervisor: Dr Gary Jones

Second Marker: Mr. Gareth Jones

Project submitted as part of the requirements for the award of
B.Eng. (Hons) Computer Systems & Electronics

April 2016

Abstract

This report documents the design and creation of the Lightframe: an optical means of human interfacing. The report will explore and discuss research on the matter relevant to the topic and explain the methods utilized alongside noting achievements of the project.

Special thanks to Gary Jones for his continued support and expertise in supervising.

Table of Contents

Abstract	2
Chapter 1: Introduction	4
1.1: Rationale	4
1.2: Project Aims	4
Chapter 2: Research Methodology	6
2.1: Methods of Research	6
2.2: Required resources	7
2.2.a: Hardware	7
2.2.b: Software	8
Chapter 3: Literature Review and Relevant Research	8
3.1: Function relevant research	8
3.1.a: The interface	8
3.1.b: Microcontrollers.....	10
3.1.c: Communications.....	13
3.1d: Collimation	14
3.2: Market research.....	15
3.2.a: Bernard Szajner	15
3.2.b: Jean Michel Jarre/ Yan Terrien.....	15

3.2.c: Serafim ODIN	16
3.2.d: Celluon Epic Laser Keyboard	16
Chapter 4: Design of the system.....	17
4.1: The Interface	17
4.1.a: The microcontroller.....	20
4.1.b: The Frame	20
4.2: The code (Flowchart)	25
4.2.a: The code expanded further goal : MIDI	27
Chapter 5: Implementation and Analysis	29
5.1: Simulation	29
5.2: Comparator Testing and Calibration.....	30
5.3 The Code	33
5.3.a: Simulation code.....	33
5.3.b: Practical code and PIC program	36
5.3.c: The strobing effect	36
Chapter 6: Conclusion.....	37
6.1: Bill of Materials	37
6.2: The Conclusion.....	38
References.....	40

Chapter 1: Introduction

In this chapter, the concept of the project will be discussed alongside the projected progress and ambitions going forward with the project.

1.1: Rationale

The intent of the major project is to create a device for human interfacing using optical methods; in this case, visible light was the means of interfacing chosen. A great deal of applications require a direct physical connection in order to function as an interface, where some of which may be more suited to ideally implementing a contactless method of human interfacing. The implementation of such methods may open an avenue of not only convenience but also accessibility, whether it be in the form of musical instrument or as a computer input device, replacing a keyboard and or a mouse.

As the demand for alternative means of interfacing grows (I.E, touchscreens, voice control, motion control amongst others), the appeal for the use of such methods is vast, while offering quick and simple means of navigating and interacting with devices.

Eliminating the need for direct contact, it allows for greater and easier access for those who may have disabilities or physical constraints, be it arthritis or missing fingers amongst anything else that may impede someone's ability to use computers with conventional means of inputs. The aforementioned is easily overcome as all that is required is a direct line of sight interrupting response interface with optic devices.

Not only is there the scope for aiding accessibility for those with difficulties, there is also scope for use in a laboratory environment, where the maintenance of sterility is of paramount importance, as it aids the maintenance of cleanliness by offering fewer opportunities for substances to become lodged and stored underneath any surfaces. Keyboards are more than suited to harbouring stated contaminants, with the gaps in the keys, which would also be unwanted in a workshop type environment, as shavings and small parts may become lodged under the keys, obstructing them from functioning as designed.

1.2: Project Aims

The aim of the project is to create a system that allows for human interfacing without requiring physical human contact, it was originally proposed the use of visible or invisible light were to be used, but it has since been deemed more appropriate to utilise only visible light for the purposes of the initial design. More ambitious branches of the application would implement invisible light, such as the use of a "pseudo touchscreen", where view behind the interface would be undesirable to be obstructed by the use of a visible light matrix in front of the screen.

The ultimate goal in this project at this time as an initial aim for a final product, would be to create a "light board" in which would create outputs according to interferences between the point of emitting the light and the receiver. Two axes will be used, as one axis would result in a product overly large in length, thus

rendering itself highly impractical, two axes will remedy this by significantly reducing the length in exchange for a lesser amount of height.

Having two axes also boost the number of available inputs, assuming a single axis board of 16 inputs is used, consolidating 8 from that axis and placing them on another axis to form a matrix will allow for a total of 64 different inputs, not only consolidating size for practicality, but also providing a far greater scope of functionality.

In Figure 1, as seen to the right it is demonstrated how different points within the board will operate fundamentally and on a minor scale (showing a 2x2 matrix)

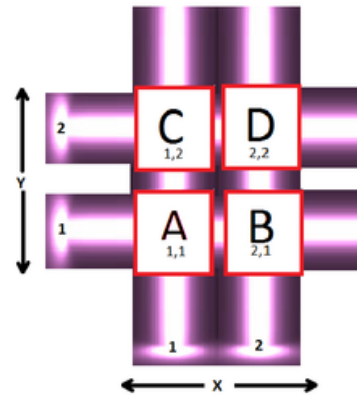


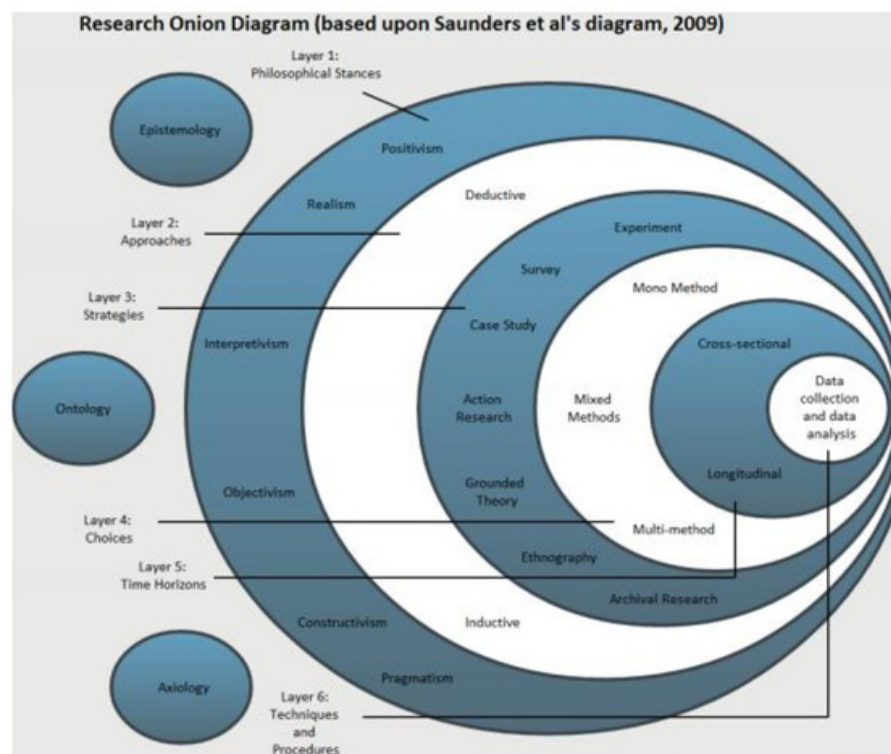
FIGURE 1

Chapter 2: Research Methodology

In this chapter the employed means of research and used techniques will be discussed and how they will be implemented in the aid of going forward with this project. The methods will also be analysed in their effectiveness and relevance, alongside a brief discussion on other available methods.

2.1: Methods of Research

In order to discuss the methods the “Research Onion” model will be employed in order to assist in this task. The onion is a concept of dividing the research into various topics and stages, while stemming from a core branching into relevant familial relations until the implemented method is reached and defined. Below in – the research onion is illustrated, showing how from the core of the research is data collection and analysis, then that it eventually branches into three clear separate primary means of conducting research.



(1)

The three primary approaches are Ontology, Axiology and Epistemology. Ontology and Axiology are subjective approaches, with ontology concerning more with the nature of reality and construct

opinions based on how the world operates, construction of society and how it influences everything. Axiology revolves around the role of the researcher predominantly, which produces a more biased, opinion based and narrow result. The dissection of these methods reveals epistemology to be the more effective and applicable means for researching in this project.

As Epistemology is the primary mode of research utilised in this report, due to it being dependant on data and that the data is being dealt with is based on fact, while defining acceptable knowledge within the field of research and what's known to be true through testing.

Although Epistemology is the primary method of choice, many aspects from all of the methods will be utilised to ensure maximum accuracy and relevance. It will include an extent of experimental / test based research conducted in the actual process of the creation of the project. For the purposes of further reference in the report the method will be referred to as 'Experimental Epistemology'.

Experimental Epistemology was deemed a more appropriate approach to the problem due to the nature of the task at hand having somewhat unpredictable results in certain areas due to unforeseeable complications, for example; a component or tool being non-functional or damaged prior to application. There is also the case that some chosen components may be reliant on outside variable factors, for example photodiodes, light strength and their resultant outputs.

The advantages of following this method is that it's a far more practical approach to the matter, providing flexibility in solving issues that arise and ability to adjust for errors and outside variable parameters as and when they arise. The disadvantages are that there is not a huge amount of data in this particular subject that is available, as there aren't many similar products widely marketed and there is not an extensive amount of widely known projects relating to the matter, meaning there will be a greater reliance on the experimental side.

2.2: Required resources

In order to complete the project to an appropriate standard many resources will be required. The following resources may be used, or will be proposed for future use in the process of completing the project, as listed under hardware and software.

2.2.a: Hardware

- PC (With Windows operating system and internet access)
- Pickit3 Programmer
- DC Power Supply
- Signal Generator
- Multimeter
- Oscilloscope/ picoscope
- PIC Microcontroller
- Frame
- LEDs

- Optical detection devices
- MidiController

2.2.b: Software

- Proteus
- MPLAB
- Microsoft Word
- CCS
- Picoscope 6
- MikroC
- VirtualMIDIsynth

Chapter 3: Literature Review and Relevant Research

3.1: Function relevant research

3.1.a: The interface

The project will require the use of a frame for socketing and aligning the matrix of lights, a frame was chosen with the desire of limiting surface area of operation so that the user would be less likely to “miss” the inputs, while remaining obvious and easy to use.

The frame’s details are as follows:

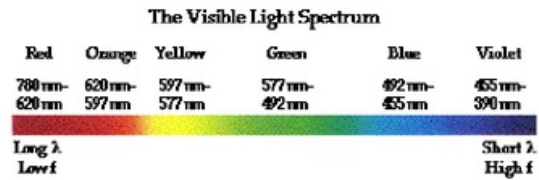
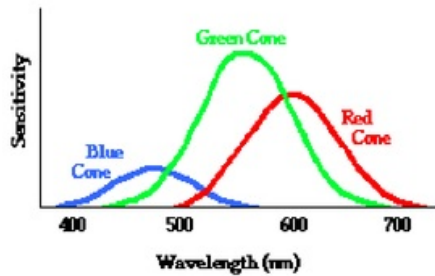
- Outside Frame 194mm x 145mm x 40mm (x,y,z)
- Inside frame 170mm x 120mm (x,y)

The project will require the use of optics, for sending and receiving signal.

To send the signal, Green LEDs are to be used.

As disclosed in the project proposal, green is the optimal choice for use in the project, due to the peak sensitivity of the human eye being approximately 550 nanometres, which is near to the peak wavelength of green’s spectrum. (2)

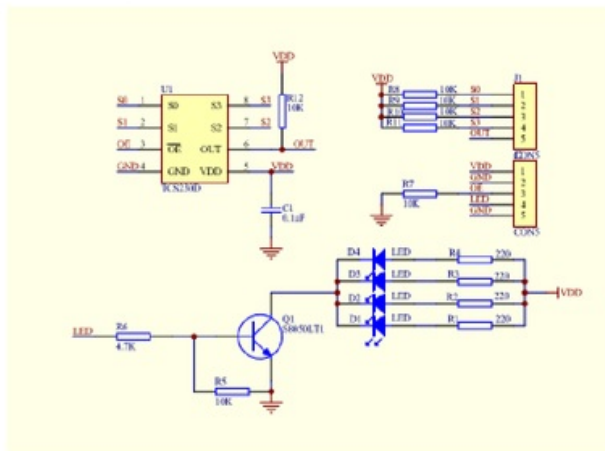
(2)



While lasers would provide a highly visible, intense column of light, following inquiries into the use of them, they are deemed unsafe for eyes, but the used LEDs must have sufficient intensity to impact opposing photodetectors across the distance that separates them, while not impacting their neighbouring sockets.

Research into the receiver side yielded promising results, providing two options for use, being the inexpensive photodiode and the effective colour sensor.

Below, the colour sensor TCS3200 can be seen along with the illustration of its circuit design.



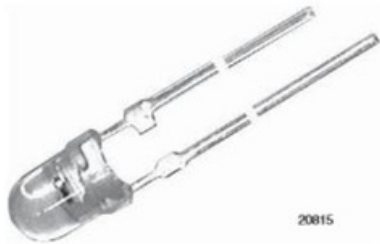
(3)

The colour sensor is a good option to pursue for the project as it is fully programmable, can connect directly to the microcontroller, and due to its means of colour sensing it is able to reduce and eliminate interference from other sources as you can program it to detect exactly as is desired.

The cost is not great, being priced at \$7.90 (3)(roughly £5), however to be used on a scale of 16 inputs, the cost would quickly mount to £80, rendering this means relatively impractical.

The second means of detecting the photons emitted from the green LEDs, is a photodiode, one specifically doped so that its peak sensitivity is nested in the heart of the green spectral wavelength.

The photodiode found that can effectively do this, is the TEPT4400, shown below.



The TEPT 4400 is 3 millimetres in diameter and comes at a relatively negligible cost in contrast to the colour sensor, even when used in the numbers as desired.

It has a peak sensitivity of 570 nanometres, which is ideal, as it is advertised as sensitive as human eye. When the green photons reach the semiconductor in the heart of the component, electrons are “excited” so that they contain enough energy to allow current to flow through the doped semiconductor.

(4)

Despite the fact that this component can be easily interfered as it still can be allowed to detect a greater range of wavelengths than the colour sensor, its inexpensive cost and simple implementation determines itself to be the most suitable means of detection. [3] The means of interference can also be circumvented with simple hardware modifications of the frame itself.

3.1.b: Microcontrollers

For the intended goals of the project, it will require the use of a microcontroller. There are a number of available options, but to minimum of 25 input/output pins is required, meaning a 40 pin device was needed, with a USART compatible pin for serial communications.

PIC16F18877

Initially the PIC16F18877 was found, it appeared to be an ideal answer to the base requirement of the circuit as a microcontroller; it had 40 pins, USART communications and many other features. It featured low power consumption and many core and communications peripherals, while boasting many other desirable features. The issue which

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	28
CPU Speed (MIPS)	8
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 2-SPI, 2-I2C
Capture/Compare/PWM Peripherals	5 CCP
Timers	1 x 8-bit, 4 x 16-bit
ADC	24 ch, 10-bit
Comparators	2
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	40
XLP	Yes
Cap Touch Channels	24

was later discovered that this is a future product and does not yet exist, the specifications are displayed on the right.

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	14
CPU Speed (MIPS)	5
RAM Bytes	368
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	2 x 8-bit, 1 x 16-bit
ADC	8 ch, 10-bit
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Pin Count	40

PIC18F(2/4)5K50

For a project with a greater aspirational end goal, the 18F series is ultimately more desirable, being more advanced than the 16F, as a standard they typically are compliant and ready to support USB communications. These particular models are highly compatible with the requirements of the project, while providing scope to progress further with the idea. The difference

PIC16F877

In lieu of the availability of the original designated microcontroller a simple alternative is feasible, it's an inexpensive choice, while still providing the basic key features that were required- the 17 Input/output pins and USART communications capabilities. The PIC is also readily available for use, which presents itself as a good alternative

Parameter Name	Value
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	12
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	1 ECCP /
Timers	2 x 8-bit, 2 x 16-bit
ADC	25 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, FS Device, USB 2.0
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	40
XLP	Yes
Cap Touch Channels	25

between the two is primarily the amount of pins presented, the 25k50 being 28 pin while the 45k50 being 40 pin.

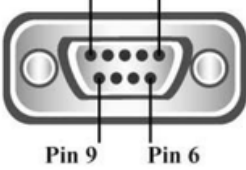
3.1.c: Communications

The RS232 standard is a relatively outdated means of serially communicating data between devices, it was the primary means of doing so prior to USB. It is easier to implement however was far slower and potentially larger than USB. (5)

BL-830 RS232 Pinout

Pin 1	RI
Pin 2	TXD
Pin 3	RXD
Pin 4	DSR
Pin 5	GND
Pin 6	DTR
Pin 7	CTS
Pin 8	RTS
Pin 9	Power Input /DCD

**RS232 Pinout
(9 Pin Female)**



Pin 5 Pin 1
Pin 9 Pin 6

It is possible to achieve communication between a PC and a microcontroller with the use of an intermediary device or relevant connections. In order to communicate directly into the RS232 ports of a PC an intermediary device is required – such as the MAX232 driver. The driver must be connected to the USART pins of the microcontroller. The Tx pin must be wired to the MAX232 as transmit pin, and the RX pin must be connected to the corresponding max 232 port. For the goals of the project it is not necessary to implement the receive pins for RA232 as the project is merely an interface to be sending commands to the PC or other attached device.

A far more efficient, easy to implement and even cheaper alternative to the MAX 232 is available – which is the use of a USB TTL cable – all that's required is to connect the cable to the USB port of a computer, attach the red wire on the other end to 5v, black to ground, white to Tx of the chip and green to Rx. The image of the cable can be seen below, so that the wiring references can



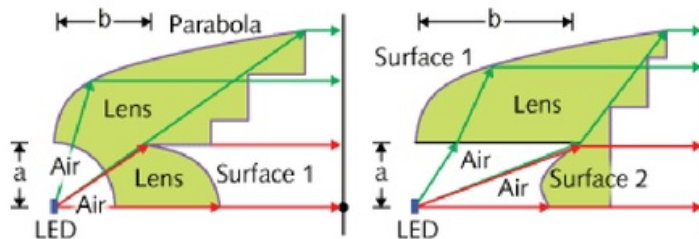
be clearly seen and understood.

(6)

The final requirement of this means of communications is simply a compatible program such as PuTTY, Hyperterminal or Teraterminal. These function simply as tools to display information being relayed on the RS232 connections. (7)

3.1d: Collimation

Collimation effectively works through reflecting light signals and taking advantage of the effects curved lenses can have on light paths, with reflective surfaces it 'straightens' the path by focusing it on a more minute area. The issue with collimation and this project is that numerous collimators will be required and they are not conventionally low costing, as can be seen to the right, a single lens costs £15.60, across 16 LEDs that would mount to a cost in excess of £250 in lenses alone. (8)



3.2: Market research

3.2.a: Bernard Szajner



Bernard szajner built the first framed laser harp which was little more than “a simple triangle with the cells, interfaced to the expanders via a CV circuit made by Didier Badez.” The harp was estimated to have cost approximately £10,000 to build at the time. It employed the use of a Spectra Physics Argon (oem type : not with solid external casing), 5W, a diffraction grating, working with C/V and gate. (9)

3.2.b: Jean Michel Jarre/ Yan Terrien



Jean michel Jarre popularised the lazer harp with the frameless incarnation but it was in fact Yan Terrien who had redesigned and rebuilt Bernard szajner’s concept. To build this he had used 20 Watts, Argon, (Coherent or Spectra Physics), there was no blanking, each beam was a position of the scanner and used a scanner as the positioning device. Also a PCL-812 board (A/D converter was used)

(10)

3.2.c: Serafim ODiN



The Serafim virtual mouse pad displays on an 8x8cm area and functions like most touchpads on laptops, has multiple functions according to positioning of fingers and movements. There is not much information readily available on this product but it is likely that it functions very similarly to the “celluon epic” laser keyboard.

(11)

3.2.d: Celluon Epic Laser Keyboard

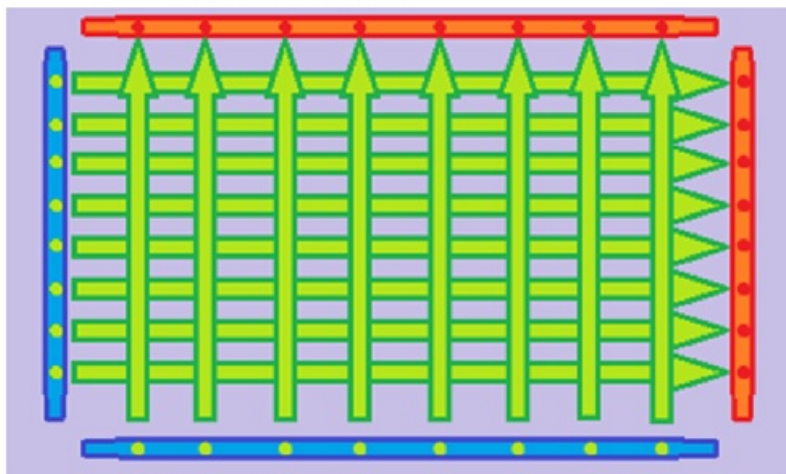


The Epic virtual projected keyboard functions by sending a laser out and an IR detector detects the reflections from the fingers and is capable of loading up to six characters per second and transmit over bluetooth. A drawback of this product is that it requires a flat and opaque surface to function optimally.

(12)

Chapter 4: Design of the system

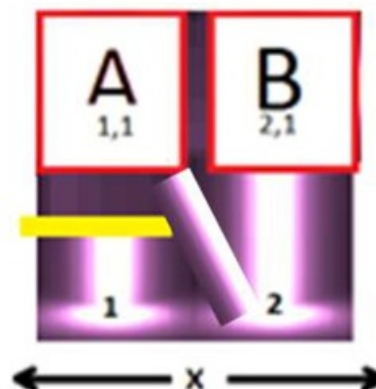
4.1: The Interface



As shown below, the general structure of the layout of the LEDs and the path of travel for the emitted light, onto the receivers (photodiodes). This outlines how the project will be constructed. However there are concerns that arise from this structure as explained below with the aid of the image to the right. The image illustrates the beam from light 1 being interfered, however receiver A is still having a small amount of light reach it, due to refraction that may occur. This is an issue that needs to be addressed as it can compromise the entire functionality of the system.

During early testing of the interface it was noted that interference from neighbouring LEDs being constantly on and the opposing neighbouring photodiodes were being “excited” despite the desired beam path being blocked. There was also another issue which was relatively unforeseen where reflection from the high intensity light impacted undesired photodiodes when using the interface upon insertion of a digit. The early concern was also confirmed when the Y axis inputs were installed, due to their distance from the opposing components, a lot of diffraction was observed and hence allowed interference on the opposing neighbouring photodiodes. This could be circumvented in four ways, the first by the addition of tube-like sockets for the photodiodes to sit in, the additional distance provided by the extension of the tube, will reduce the interfering light reaching the photodiode.

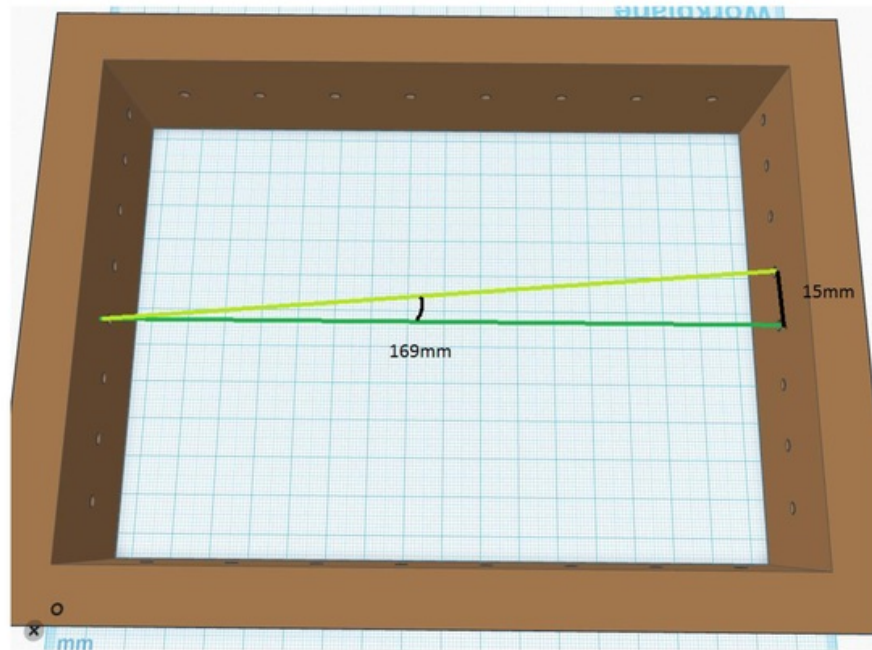
Another means of doing so is by adding collimating lenses, but the issue that arises with lensing is the size availability of lenses and the costing, for function in this project, custom made lenses would have to be created, by contracting an external specialist company. A more unorthodox method in which the issue could have been prevented would be using a more uniformly shaped



square, the Y axis being greater so that the spacing between each LED and between each photodiode could be increased, reducing the amount of diffracted light that would be reaching the photodiodes.

While collimating lenses are extremely effective due to the increased intensity and coherence they offer as well as the collimation, should this have been the method of choice, there would have needed to be a maximum of 5 degrees from normal of collimation, at the sizes specified for the components, this would be highly expensive to implement. The result of five degrees stems from a worst case calculation given the measurements shown below from the longest distance and smallest spacing between the components.

$$\begin{aligned} &\tan^{-1} \\ &(O/A)=(15/169) \\ &=5.0721^\circ \end{aligned}$$



The final and desired method of circumventing the issue is by having only one of the LEDs active on each axis at any given time, which can be achieved with a strobing effect which will allow for this to occur without the user having to notice the effect. While the designated LED is active, only the input from the opposing photodiode is being registered by the pic. This method is the most effective as all it calls for is a small addition to the code and consolidation of some of the layout of the hardware.

Having opted for the strobing method, it becomes imperative to select a chip with 40 pins as a minimum considering the microcontroller will need to drive the intervals for the LEDs as they switch. As the LEDs are being split into two separate arrays, this can be taken advantage of to consolidate pins on the PIC – one on each axis can be assigned to the same pin on the pic, reducing the addition required pins to eight rather than sixteen, reducing the minimum required input / output pins to 26, leaving potential for future development.

4.1.a: The microcontroller

The designated microcontroller for this project will need to have a minimum of 16 input pins, alongside USART compatible pins, totalling 18 pins for minimum basic Input/output function excluding any power pins. If any enhancements or changes to the designed standard function are required to be made going forwards, a 28 pin device would not accommodate for much in the way of additional ports and functions, meaning a 40 pin device will be needed to allow for project development.

Microcontroller selection

In order to choose the microcontroller, it was necessary to identify what functions are required of the microcontroller to perform. Initially, it was determined that a minimum of 16 input pins were required to be input to the PIC, eight for the X axis input values and eight for the Y axis input values. As well as the 16 inputs, it will need to have eight output pins for the LED strobing effect and USART ports, so another two pins are required although one of the USART (Rx) pins won't be in use and will function as a "dummy pin". The controller also needs to be able to receive and send data reasonably quickly so a decent clock speed and a good CPU speed is necessary. Ideally, the PIC will have USB compliance as a standard so that the project can potentially be adapted as progress is made, which will also result in a more marketable product.

A minimum program size of eight kilobytes will be needed to run the system as designed (to also allow for continued future development of the project) with a reasonable amount of RAM.

Using the PIC selector tool on the microchip website, the PIC18F45K50 was chosen, below, is displayed, showing the PIC being the only result that followed from inputting these minimum requirements into the application, without having input all the desired criterion.

All 8-bit Microcontrollers Products

Reset All Filters Show/Hide Columns Download Switch Views: Summary Show All Specs

To SORT a column click the column header. Use CTRL key to select multiple values.

Product	Program Memory Size (Kbytes)	RAM (bytes)	Pin count	Max. CPU Speed MHz	Internal Oscillator	UART	Number of USB Modules
Reset PIC18F45K50	Reset 32	Reset 2048	Reset 40	Reset 48	Reset 16 MHz	Reset 1	Reset 1

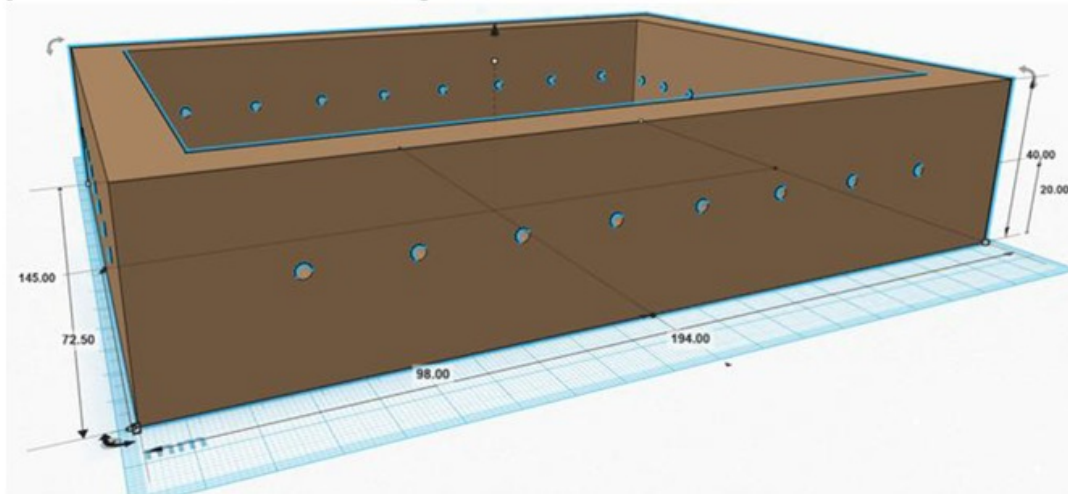
(13)

4.1.b: The Frame

A frame is required as an outline for the interface, to provide rigidity and a structure for the project, alongside providing an effective means of securing the LEDs and receivers in place and aligned with each other. The frame will also serve as a recognisable visual reference point for human interfacing, as well as ease of understanding, while providing a physical boundary for

case of use.

The frame was acquired and measured so it could be customised for housing the LEDs and photodiodes, evenly spaced and in line, with each LED directly opposing a corresponding photodiode. Below, is a 3D rendering with measurements of the dimensions in millimetres.



First the outside axes of the frame were measured and the X axis of the frame was measured at 194 millimetres in length while the Y axis of the frame was measured at 145 millimetres in length, with the depth being 40 millimetres. Following the outside measurements, the internal axes of the frame were measured, to determine the spacing required between each bored hole.

The inside axes of the frame were then measure

Due to the nature of light, the bores were required to be as far apart as possible, as light spreads out from the centre point, as a result of diffraction. This is a problem for the entire system, as each photodiode is one input, the value of which is decided according to whether its line of sight with its corresponding LED is broken or not. If other light signals are having a significant enough of an impact on the output signal of the photodiode, it would render the desired functionality and design ineffective.

Eight evenly spaced holes were bored into both sides of each axis of the frame for the board, at a diameter of 3 millimetres, while a counter bore on one side of each axis.

The counter bore was used in order to socket the LED, while maintaining the same collimation that the 3 millimetre hole provides after the LED.

The holes having been bored at 3mm in diameter will help cancel out cross interference from

other light sources when active; only allowing the light source directly opposite to correspond with the photodiode. The holes on the light source sides, having been counter bored at 5 millimetres in diameter allows for socketing of the LEDs, while the through bore of 3 millimetres in diameter allows for natural collimation of the emitted light.

There is a distance of approximately 115mm between the X axis LEDs and photodiodes and a distance of approximately 190mm between the Y axis LEDs and photodiodes. The primary concern is that the photodiodes may not react enough to the light from the LEDs over the given distance, while another is the rate at which the light of the diodes may spread.

In lieu of available practical collimators for use on the board in accordance with physical limitations where the board is inapplicable for compromising, use of tubular implements may be applied to the board in order to provide further natural collimation, by minimising spread of the LED light, resultantly minimising the entry of the light from peripheral LEDs.

USB TTL 232

In order to create a serial output of data to a computer, a USB TTL 232 cable will be implemented and connected to the USART pins of the microcontroller. Pin C6 should be wired to the receive wire as transmit pin, while the receive pin will not need to be implemented as the project is merely an interface to be sending commands to the PC or other attached device.

The data output is to be producing signals for an RS232 port, to be displayed on a PC screen through the use of software (PuTTY/ Hyperterminal/TERA)

5V voltage regulator

To ensure that excessive voltages do not get applied to components in the circuits such as the microcontroller, voltage regulators will need to be implemented, as a standard precaution for protecting components in the circuits and maintaining a steady voltage on the output of the regulator.

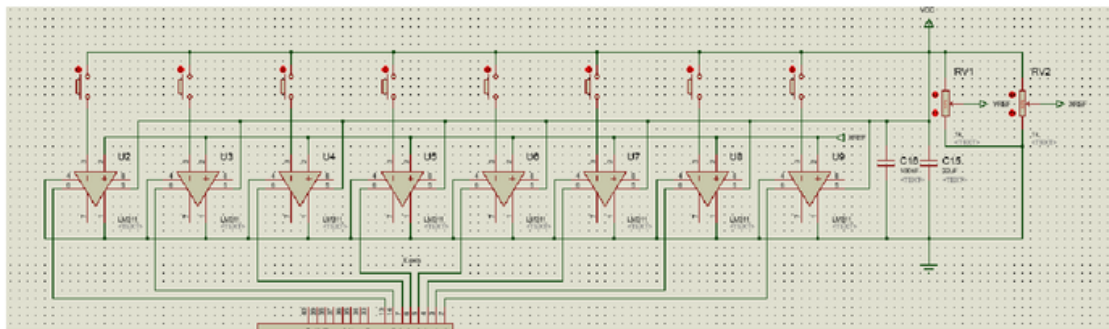
As well as for functional reasons it was also implemented for use supplying comparators, despite their ability to function with far greater voltages, their outputs would mimic that of the greater voltage input and therefore the output would exceed the limit of which the microcontroller could function.

PIC18F845K50

The PIC18F45K50 was set up to use its internal oscillator at 16MHz, it is important to have a clock at a high enough frequency as an effective interface requires having a minimal lagging factor.

Below, the proteus schematic of the PIC18F45K50 can be seen

Lm 339 – Quad Comparators



Comparators were implemented to allow the light matrix to function as switches, when the photodiodes' corresponding light sources are disrupted, the output to the comparator increases, when this output becomes larger than the set reference voltage, the comparator “switches on” the corresponding pin, creating an output of “on” for the PIC to receive.

The reference voltage is to be set by potentiometer, one for each axis, due to the distance on each axis differing from one another, assuring the output would vary on each axis.

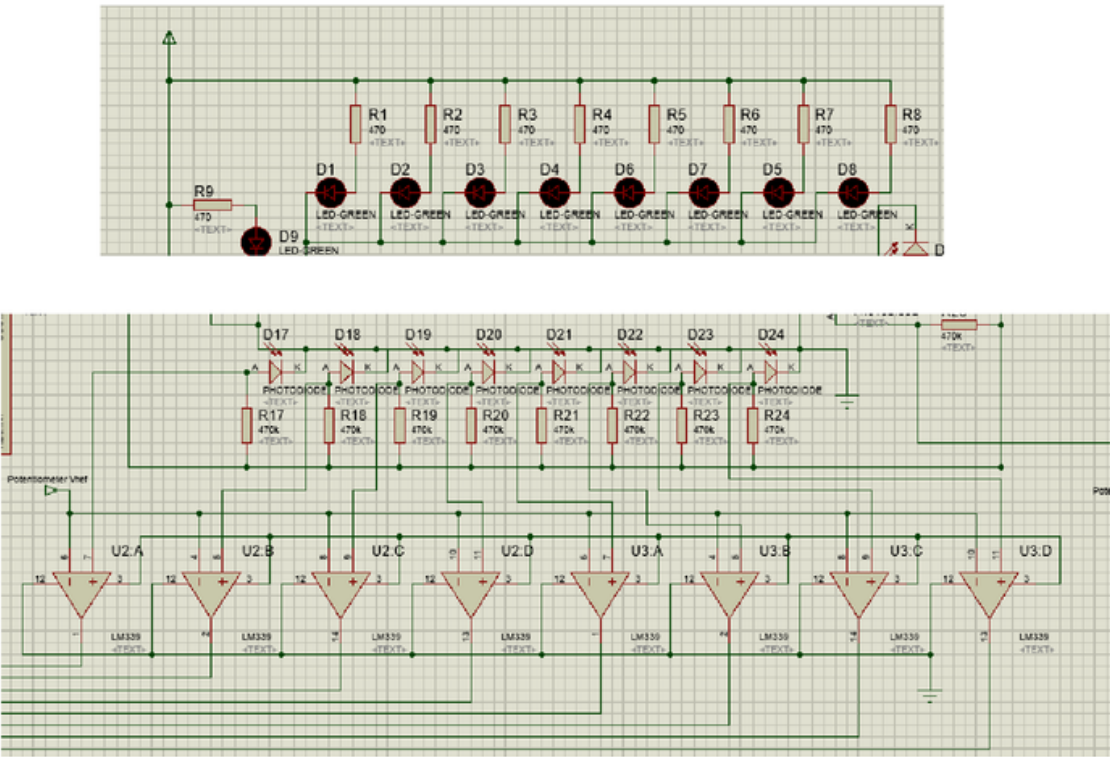
The reference voltage was set by considering the minimal change when the line of sight was interfered, alongside considering changes when line of sight from neighbouring light sources were interfered, to eliminate the difficulty created from the diffraction of light interfering on non-corresponding light/receiver collaboration.

The Light matrix

Upon initial testing of the basic functionality of the arrays of LEDs it was found that The light matrix on the X axis functioned as expected, but when the Y axis was attempted to be set, it was discovered the used green diodes were not strong enough to provide a large enough impact on the photodiodes across the distance, indicating a more powerful, coherent light source was required. It was found that the L-7083ZGD Kingbright high intensity LED was available, which functioned at 520-530 nanometres, proving perfect for the use in the project with minimal

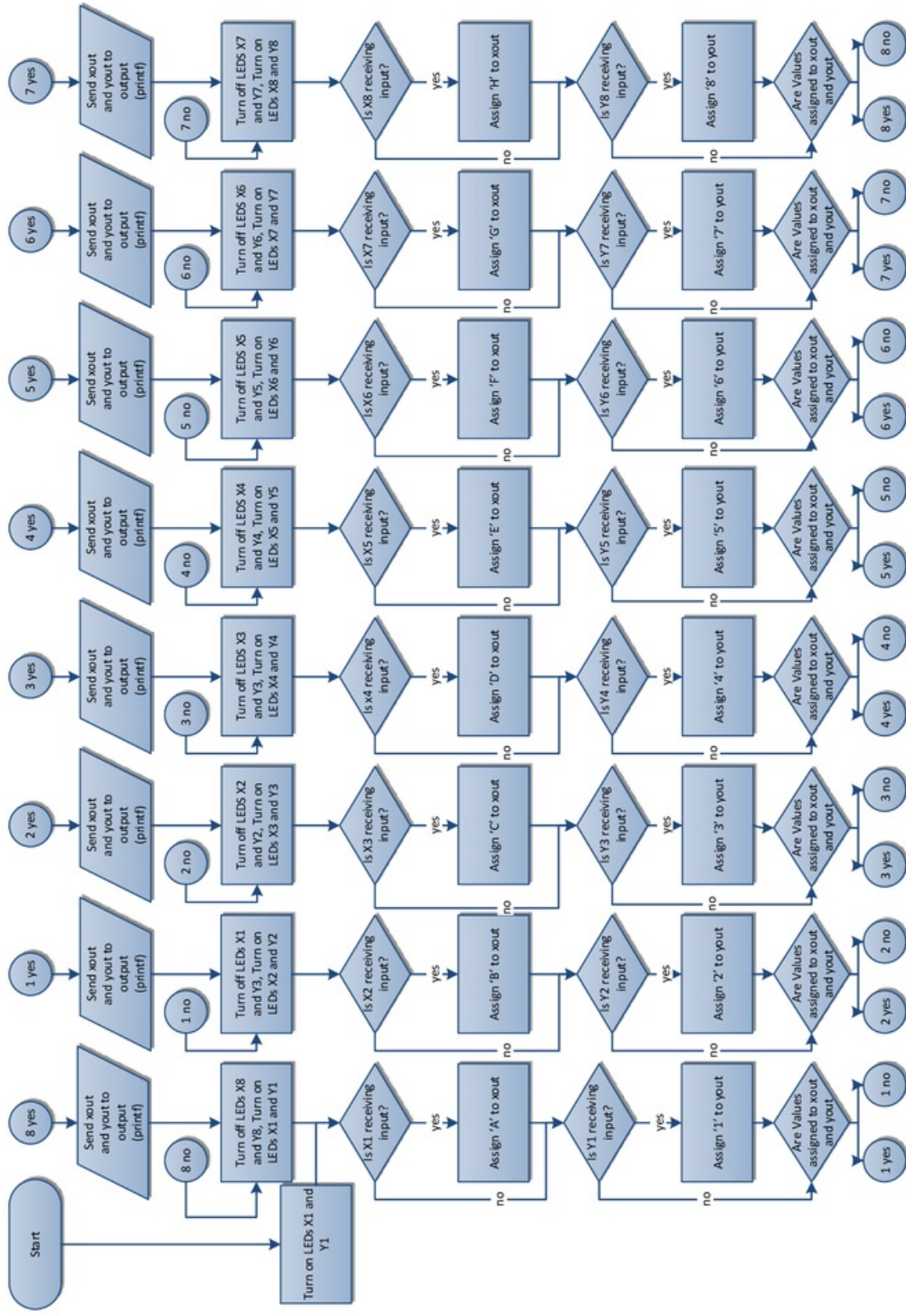
modification required. The New LEDS were found to be more than adequate for the task, so were kept and used. The difference in the impact of the lower thresholds were now sufficient enough to provide a reference voltage for the comparator to work effectively with.

Below, the layout of the interface is shown for the X axis, the Y axis being identical, merely transposed. It shows how the LEDs are to be placed opposing the photodiodes, which in turn are connected into comparators.



4.2: The code (Flowchart)

A flowchart was created to illustrate the design of the functioning code in a clear manner to be observed in aid of producing the code and also explaining the code. It can be seen that a series of decisions are made in order to determine what the output of the code is to be printed. The representation of the function and progression of the code, shows how the output for xout and yout is determined based on the inputs from X1 to X8 and Y1 to Y8, alongside the delays of the LED light output. The flowchart is represented on the following page.



4.2.a: The code expanded further goal : MIDI

For the purposes of the ideal demonstration of the functionality of the interface, it was decided that an audio output would be implemented. In order to do this, the code needs to be optimised for MIDI output.

MIDI is controlled by 3 messages, the status message, the note message and the message for the volume. Each message is one byte long and the output is determined based on the value of these bytes.

The NOTE ON message is structured as follows:

- *Status byte : 1001 CCCC*
- *Data byte 1 : 0PPP PPPP*
- *Data byte 2 : 0VVV VVVV*

where:

"CCCC" is the MIDI channel (from 0 to 15)

"PPP PPPP" is the pitch value (from 0 to 127)

"VVV VVVV" is the velocity value (from 0 to 127) [4]

The programming language used, C, does not have a means to enter binary in the code as a standard usable value, as C is already using binary to drive the programming, so in order to bypass this, that a value may be sent as output, through reviewing the CCS compiler manual, the int8 declaration would be implemented. Int8 ensures the value uses 8 bits to identify the value, limiting the value between 0 and 255 [5]

Using the above identifier ensures that the used data is in the correct format for use in a midi controller. If there were to be greater or fewer bits used in the instructions, the MIDI controller would not interpret the information it receives as intended.

The MIDI standard has an already determined set of instruction values; each data byte has a range of 0 to 127 to be used, where the value in the first data byte determines the used note, while the value in the second determines the volume.

Below, the MIDI value chart for the notes to be used in the first data byte can be seen, which was used to choose the 8 values for the intended notes: 60-C, 62-D, 64-E, 65-F, 67-G, 69-A, 71-B and 72-C.

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

[6]

For the volume the value 120 was taken as a maximum, as a sensible number, being easily divisible by eight, allowing a uniform increase in volume steps along the axis.

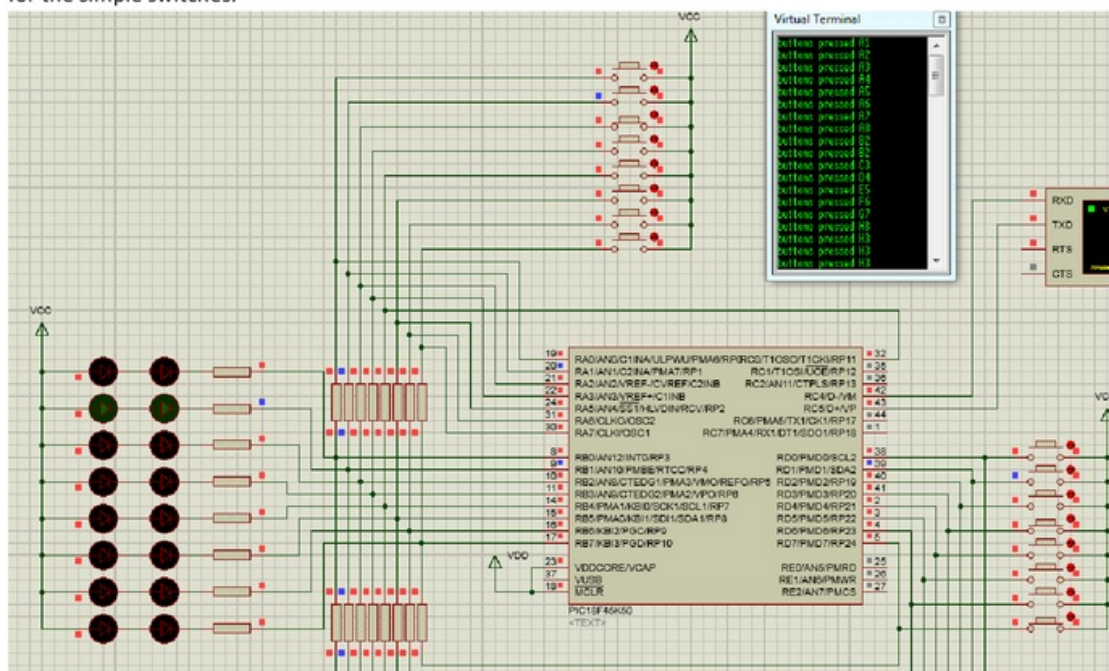
Below is an image of The Lightframe, with labelled values on each receiving end to illustrate the corresponding values of the volume as well as the note values.

Chapter 5: Implementation and Analysis

In this chapter, the results of implementation will be discussed and analysed. This includes both the software simulation and the physically conceived entity of the project. There are a few differences between the physical incarnation and the simulation, which will be discussed appropriately as they arise within this chapter.

5.1: Simulation

As can be seen below, the simulation is working perfectly, outputs are being generated to the virtual terminal as is programmed from the PIC. The photodiodes have been substituted for standard buttons as an acceptable alternative as it is not possible to simulate the signals of light and its intensity and how it diminishes over distance and as the comparators are effectively operating as DC/digital switches, it was deemed acceptable to switch these for the simple switches.



Another difference to standard is that there is a different test code running the simulation opposed to the hardware, as proteus did not have the files for the previously selected Microprocessor, many alternatives were reviewed as an acceptable temporary substitute. Ultimately the PIC18F45J50 was selected for this purpose, as it was relatively close in spec to the 45K50, however the Pin layout was slightly different, so the pin layouts differed somewhat, with a temporary set up port on pin C0 to accommodate for pin A2 and also an error within proteus' configurations had arisen, in the simulation pin C6 was refusing to transmit data, so a temporary reassign was made to pin C4 instead.

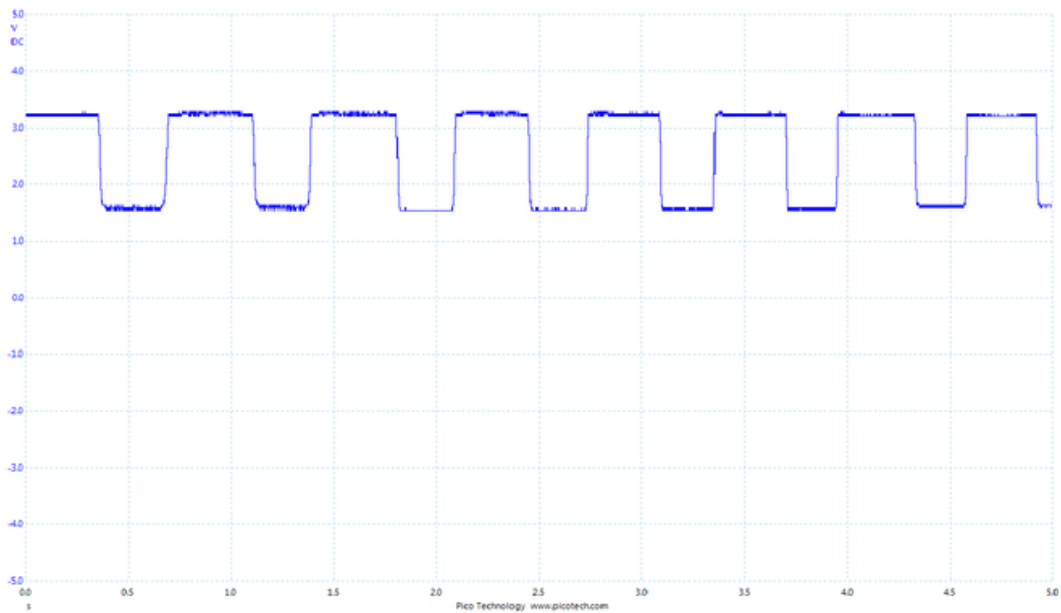
5.2: Comparator Testing and Calibration

The comparators were tested in conjunction with the functionality of the photodiodes, to demonstrate its usefulness in the application of the project. The comparators' ability to create rail to rail output essentially allows the board to function as an interface of switches, but is required to be calibrated with the use of a reference voltage.

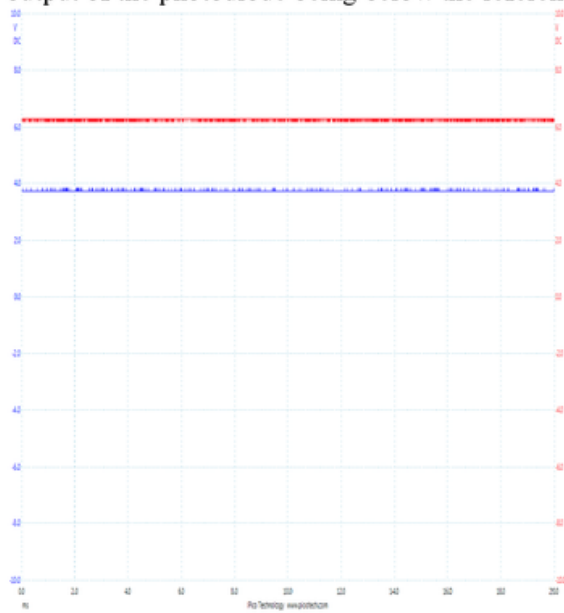
The comparator was set up with an LED facing directly into a counterpart photodiode, with the input to the comparator coming from the anode of the photodiode. The reference voltage was provided through implementing a potentiometer, which was necessary due to the nature of results from light based communications being easily influenced.

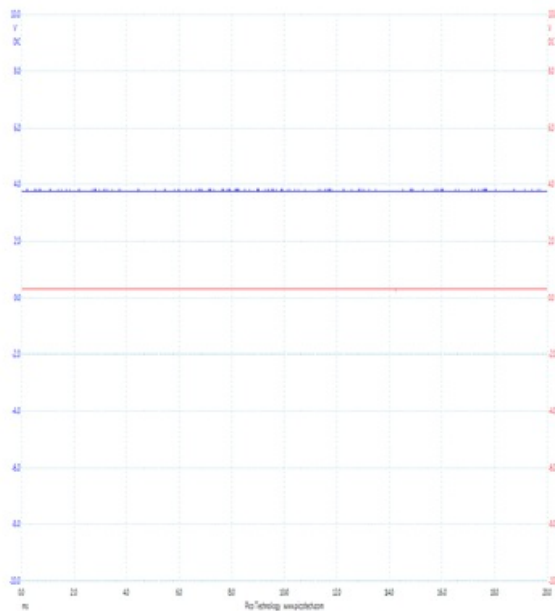
In order to calibrate the comparator with the potentiometer, the difference in the outputs of the photodiodes when the light from the LED is blocked and when there is a direct line of sight was observed, by noting the voltages at both points. The potentiometer was then set to output a voltage in between the two given points, creating a suitable reference voltage for the comparator to use.

Below, the voltage output from the photodiode can be seen, oscillating due to the light signal being blocked from the LED; with a voltage difference of nearly two volts. This is ample change in voltage and will prove easy to determine a value for the reference voltage to set on the output from the potentiometer; in this case a value of 2.5 volts, a secure value between the peak and trough of the output.



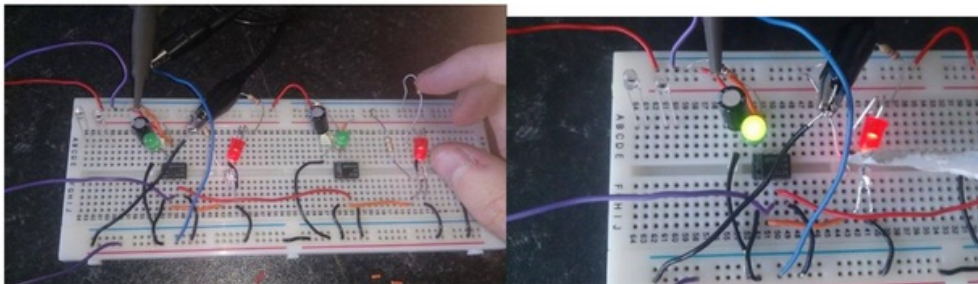
Below it can be seen the comparator generating its output, where the photodiode's output exceeded the assigned reference voltage from the potentiometer, alongside the result from the output of the photodiode being below the reference voltage.





To illustrate the functionality of the circuit using the comparator, an LED was placed on the output of the comparator, so that when there is an output being generated from the comparator, the LED would begin to emit light as an indicator of current being generated.

Upon denying the photodiode's line of sight with its fuelling light source, the current being permitted to travel through the diode is restricted, causing an increase in current to be diverted to the comparator's input. The increased current is greater than that of the set reference voltage on the potentiometer, hence causing the comparator to "turn on" its output pin and as such, the output is that of the high end of the rails.



5.3 The Code

5.3.a: Simulation code

In order for the microcontroller to drive the circuit, it requires a code so that the pins can detect any inputs, and create outputs. The microcontroller determines which signals to output through the USART Tx port by detecting the inputs from the photodiodes.

So that the microcontroller can transmit through RS232 using its USART ports, the settings for RS232 standard must be initialised, as can be seen in the image below, following the setting of the clock speed. As can be seen below in the screen capture of the code, these settings were placed into the header (.h) file for the PIC which contains the declarations and definitions to be called.

```
#include <18F45K50.h> // includes the pic setup file
#define ADC=10
#define delay(internal=16MHz) // sets internal clock to 16MHz and initialises delay
#define rs232(baud=9600,parity=N,xmit=PIN_C4,rcv=PIN_C5,bits=8,stream=PORT1)
// sets up the RS232 standard output/inputs

#define FUSES FCMEN //Fail-safe clock monitor enabled
#define FUSES IESO //Internal External Switch Over mode enabled

#define DELAY 1 //defines delay 1

#define ZERO_RAM //initialises zero ram
```

The set baud rate is 9.6k and the pins C4 and C5 are set to transmit and receive (respectively), as is required by the standard, despite the fact no signals are to be received and no connections will be made on the receiving pin. The stream was set as stdout so the default transmission for any commands are sent through the RS232 standard, to ensure a serial output is created based on the input from the LED / photodiode arrays. The internal clock is set to 16MHz. PortC4, was artificially set to be the transmit pin of the microcontroller, due to the aforementioned issue with proteus coverage. pin being set during the initialising of the RS232 command.

```
5 int a=1,b,d,e,f,z;
6 char x='.',y='.'; // set char values to .
7 while(1)
8 {
9     delay_ms(100);
10
11     b=255-a; // set b to value of 255 - a
12     output_B(b); // according to b, output to Port B
```

Above, it can be seen a series of integers being set up, their purpose to be discussed as and when relevant. Two char variables were also initialised, x and y, and assigned the value '.', so that when their values differ from this, they send an output intended for display, set to correspond with specific results from the interface. Following the declarations, the delay is set (as 100ms in this example), so that each time the while loop starts there is sufficient time for the lights to strobe to avoid the interference from neighbouring sources.

As a is given the value one initially, while b is given 255, a is then subtracted from 255, which begins the process of switching the LEDs by running through the port values. According to the value b is given, port B outputs on one of the pins and turns on the corresponding LEDs.

```

14 d=input_d(); // get input from port d
15 z=d & a; // clear all but the interested bit
16 if(z) // Assign x according to the value of z
17 {
18     if(z==1) {x='A';} // Assign x the value A
19     if(z==2) {x='B';} // Assign x the value B
20     if(z==4) {x='C';} // Assign x the value C
21     if(z==8) {x='D';} // Assign x the value D
22     if(z==16) {x='E';} // Assign x the value E
23     if(z==32) {x='F';} // Assign x the value F
24     if(z==64) {x='G';} // Assign x the value G
25     if(z==128) {x='H';} // Assign x the value H
26 }

```

Above, any inputs on port d are assigned to 'd' then anded bitwise with the binary integer value of a – this causes a clear value to emerge on z, so a series of if statements were made so that x could be assigned values A through H, based on the value of z and according to the given input from the interface; the code causes the microcontroller to only correspond to the first part of the output and the Y axis inputs (D ports) correspond to the second part of the output.

```

28 e=input_a(); // get input
29 if(input(PIN_C0)) bit_set(e,4); /** simulation code only - pins mismatch - require additional port
30 else bit_clear(e,4); // temporarily uses the pin C0 as though it were on the port a input set
31 z=e & a; // clear all but the interested bit
32 if(z) // Assign y according to the value of z
33 {
34     if(z==1) {y='1';} // Assign y the value 1
35     if(z==2) {y='2';} // Assign y the value 2
36     if(z==4) {y='3';} // Assign y the value 3
37     if(z==8) {y='4';} // Assign y the value 4
38     if(z==16) {y='5';} // Assign y the value 5
39     if(z==32) {y='6';} // Assign y the value 6
40     if(z==64) {y='7';} // Assign y the value 7
41     if(z==128) {y='8';} // Assign y the value 8
42 }

```

Similarly to the code corresponding to the x axis, the port A inputs are read and y is assigned a value (however in this case it is between one and eight), but due to the constraints the substitute microcontroller harbour, pin C0 had to substitute for pin A3, so an if statement was implemented to set e as 4, if an input was on C0 and if not, to clear e.

```

43     if((x!='.') && (y!='.')) // if x and y are both not . as assigned earlier
44     {
45         printf("buttons pressed %c%c\n\r",x,y); //output x and y
46         x='.';y='.';
47     }
48     a*=2; //double the value of a (move it to the next bit of the register)
49     if(a==256)a=1; // if a is 256 - reset back to 1, sends bit back to original register
50
51 }
52
53

```

Finally, an if statement was implemented to create the output on the USART, if both x and y are no longer '.', then a printf command is sent through the USART, then x and y are returned to their '.' State. With the RS232 set as stdout, the print command is to be sent directly on the pin automatically, to be displayed on a pc screen with the aid of software (hyperterminal, PuTTY, etcetera).

Following the if statement a is doubled so that the next pin will create an output for the next LEDs, this continues for each loop until reaching 256, at which point a is returned to the value of one, beginning the LED switching cycle over again.

5.3.b: Practical code and PIC program

There were a number of minor changes to the code due to the issue previously discussed with Proteus' lack of coverage for the designated microcontroller. The code was actually simplified, the RS232 setup commands in the header file were reverted to the standard Tx and Rx pins of C6 and C7 (respectively). The main code merely had to have the instructions relating to the C0 pin removed, as the designated microcontroller.

Finally the ports were changed, the output to LEDs was reassigned to PORTA, the X axis inputs were reassigned to PORT D and the Y axis inputs were reassigned to PORT B.

In order to program the PIC MPLAB IPE in conjunction with pickit3 was used, in which a hex file released from the build in proteus was released and then programmed onto the pic.

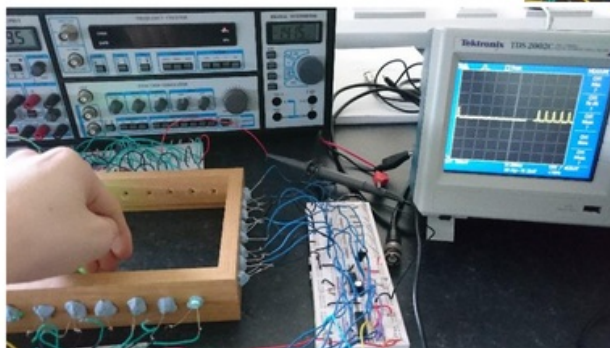
5.3.c: The strobing effect

The strobing effect was functioning as desired, at first programmed to strobe more slowly at 100ms, then changed to 10ms, followed by a change to 1ms. The three changes were made due to 100ms being an initial value for the delay was hugely noticeable to the eye and also would mean almost a second to create an output ($8 \times 100\text{ms} = 0.8\text{s}$). The value was changed to 10ms, which was still noticeably slow. While the issue was no longer heavily impacted by response time, there was too much visible flickering for a user to comfortably see. The delay was then changed to 1ms, which showed no visible flickering, proving to be satisfactory to the aims of the project.

To the right it can be seen an oscilloscope probe was set to measure on an output of a photodiode. The oscilloscope is displaying a uniform pulse, which is in accordance to the programming of the strobe.



The image displayed below demonstrates the photodiode responding to its light source being disrupted, proving that a desirable output is being



achieved.

Chapter 6: Conclusion

6.1: Bill of Materials

A table displaying the costs of all components was created; all values were rounded up to the nearest penny and most of the components' costs were obtained from the website for RS components, due to the majority of the establishment's components being obtained from RS, however, the price for L-7083ZGD could not be found, but was found on Farnell's website and thus was obtained from there.

38 Capacitors

Quantity	References	Value	Stock Code	Unit Cost
4	C25,C27,C29,C31	22uF		£0.20
12	C26,C28,C30,C32,C48-C62	100nF		£0.20
2	C33-C34	22pF		£0.22
5	C35-C39	1uF		£0.40
8	C40-C47	10uF		£0.23
Sub-totals:				£7.48

38 Resistors

Quantity	References	Value	Stock Code	Unit Cost
1	R1	270		£0.12
15	R2-R16	470		£0.12
16	R17-R32	470k		£0.15
2	R33-R34	220		£0.12
4	R35-R38	1k		£0.02
Sub-totals:				£4.64

6 Integrated Circuits

Quantity	References	Value	Stock Code	Unit Cost
1	U1	PICF45K50		£2.13
4	U20,U25-U27	LM339		£0.27
1	U24	LM2940et		£1.10
Sub-totals:				£4.31

32 Diodes

Quantity	References	Value	Stock Code	Unit Cost
16	D1-D16	L-7083ZGD		£0.27
16	D17-D32	PHOTODIODE		£0.35

Sub-totals: £9.92

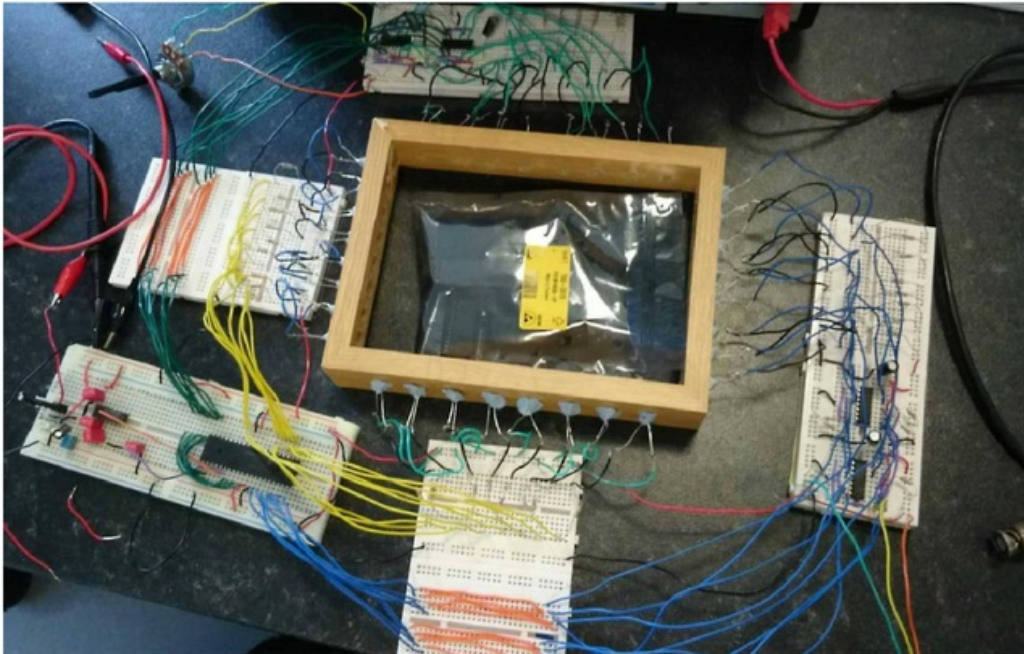
2 Miscellaneous

Quantity	References	Value	Stock Code	Unit Cost
1	---	TTL-RS232-USB		£2.13
1	---	Frame		£1.00
Sub-totals:				£3.13
Totals:				£29.48

6.2: The Conclusion

The project was ultimately incomplete, with issues regarding the photodiodes being previously damaged and not enough time to replace, re-solder and begin testing again. It is believed that the circuit in theory should be functioning as desired due to the simulation working entirely as intended and following the programming of the PIC, the LEDs were strobing successfully. Despite the incomplete project, it has been successful in numerous ways, as at least the microcontroller is behaving as desired and the simulation is functioning wholly as designed; the problem resides in what could be resolved by minor changes, however sadly, there was not enough time remaining at this point to correct the issue.

Below, the Lightframe and its circuitry can be observed.



The costing also appeared to be of a reasonable result and the components came to a total

of £29.48.

To improve the project, it could be arranged to look more attractive, this is possible by transferring the circuits to a printed circuit board, but again, sadly time did not allow for this development.

The project has plenty of scope for improvement and can also offer versatility in its development. In order for human interfacing to be easier, collimated light alongside some form of smoke machine so that the lights are visible to the human eye, allowing for easy interfacing could have been implemented.

The development potential of the project is great, the project as is could be used to substitute a keyboard for a pc, or using invisible light a mouse pad, it could also substitute with significant expansion - a touchscreen on a non-touchscreen pc. This could be achieved by creating attachments to clip on the axes of a pc screen, utilise invisible light and the same method of using X, Y co-ordinating, with the correct configuration with the PC and assign to pixels, transforming any household computer into a modern, efficient touchscreen device.

On an industrial level this project could be implemented to measure objects, or even taken a step further and use as a means of 3D rendering an object, by using as a form of a 3D scanner, an object is passed through the centre and according to which beams are broken, it renders onscreen. This device would be limited as it could not replicate complex shapes without serious modification, including adding LEDs and photodiodes on each axis as well as needing to include more sensors with smaller gaps between. The shape of the frame would ideally have to be changed to a circular shape in order to attempt to obtain maximum detail, but should this be able to be implemented, it would be able to work in conjunction with 3D printing with extraordinary effect.

References

1. **University, Derby.** <https://onion.derby.ac.uk/onion.pdf>. <https://onion.derby.ac.uk>. [Online]
2. **classroom, physics.** Visible-Light-and-the-Eye-s-Response. <http://www.physicsclassroom.com/class/light/Lesson-2/Visible-Light-and-the-Eye-s-Response>. [Online]
3. **dfrobot.** colour sensor . http://www.dfrobot.com/index.php?route=product/product&product_id=540. [Online]
4. **Vishay.** tept4400 Pdf. <http://www.vishay.com/docs/81341/tept4400.pdf>. [Online]
5. **taltech.** serial_intro. http://www.taltech.com/datacollection/articles/serial_intro.
6. **FDTI.** <http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm>.
<http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm>. [Online] FDTICHIP.
7. **camiresearch.** http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html.
http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html. [Online]
8. **laserfocusworld.** <http://www.laserfocusworld.com/articles/print/volume-48/issue-06/world-news/efficient-led-collimators-have-simple-design.html>. <http://www.laserfocusworld.com/articles/print/volume-48/issue-06/world-news/efficient-led-collimators-have-simple-design.html>. [Online]
9. **Szajner, Bernard.** Laser Harp. <http://www.stephenhobley.com/blog/laser-harp-2009/the-laser-harp-pages/bernard-szajner/>.
10. **Jarre, YanTerrien/Jean Michel.** Frameless Laser harp. <http://www.stephenhobley.com/blog/laser-harp-2009/the-laser-harp-pages/yan-terrien/>.
11. **serafim.** 339-odin-product.html. <http://www.serafim-tech.com/products/339-odin-product.html>. [Online]
12. **Celluon.** /products_epic_overview.php. <http://www.celluon.com/>. [Online]
13. **microchip.** PIC selector. <http://www.microchip.com/>. [Online]
14. **development, music software.** Midi Tutorial. <http://www.music-software-development.com/midi-tutorial.html>. [Online]
15. **CCS.** ccs_c_manual.pdf. https://www.ccsinfo.com/downloads/ccs_c_manual.pdf [. [Online]
16. **dit, electronics.** midi note numbers.
http://www.electronics.dit.ie/staff/tscarff/Music_technology/midi/midi_note_numbers_for_octaves.htm. [Online]

Bibliography

Practical PIC microcontroller projects - *Penfold, R. A*

An introduction to PIC microcontrollers - *Penfold, R. A*

Schmitt trigger as an alternative to buffer insertion for delay and power reduction in VLSI interconnects - [Saini, S.](#), [Veeramachaneni, S.](#), [Kumar, A.M.](#), [Srinivas, M.B.](#)

An infinite beam laser harp with external MIDI I/O functionality - [Ferenc, G.](#), [Popovic-Bozovic, J.](#)

Cooperative modulation classification by using multiple sensors in dispersive fading channels - [Markovic, G.B.](#)

https://www.ccsinfo.com/downloads/ccs_c_manual.pdf

<http://www.physicsclassroom.com/class/light/Lesson-2/Visible-Light-and-the-Eye-s-Response>

<http://www.adafruit.com/product/1334>

<http://www.dfrobot.com/image/data/SEN0101/TCS3200%20TCS3210.pdf>

http://www.dfrobot.com/index.php?route=product/product&product_id=540

<http://www.tpub.com/neets/book7/26g.htm>

<http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/diffrac.html#c1>

<http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/schmitt.html>

<http://www.ti.com/lit/an/scea046/scea046.pdf>

<http://www.theledlight.com/technical1.html>

<http://home.epix.net/~otisrci/pic.html>

http://www.eeweb.com/blog/circuit_projects/midi-controller-built-around-a-pic

<http://www.music-software-development.com/midi-tutorial.html>

<https://ccrma.stanford.edu/~craig/articles/linuxmidi/misc/essenmidi.html>

<http://www.midi.org/techspecs/midimessages.php>

<http://www.music-software-development.com/midi-tutorial.html>

<http://www.ccarh.org/courses/253/lab/midiprog/>

<http://www.maxmidi.com/diy/serial/8051/>

<http://www.rossbencina.com/code/midipic>

<http://www.petesqbsite.com/sections/express/issue18/midifilespart1.html>

<http://www.adafruit.com/product/1334>

<http://www.tpub.com/neets/book7/26g.htm>

<http://hyperphysics.phy-astr.gsu.edu/hbase/phyopt/diffrac.html#c1>

<http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/schmitt.html>

<http://www.ti.com/lit/an/scea046/scea046.pdf>

<http://www.theledlight.com/technical1.html>

<http://home.epix.net/~otisrci/pic.html>

<http://www.stephenhobley.com/blog/laser-harp-2009/the-laser-harp-pages/bernard-szajner/>

<http://www.laserfocusworld.com/articles/print/volume-48/issue-06/world-news/efficient-led-collimators-have-simple-design.html>

DATASHEETS

<http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC18F45K50#documentation>
(PIC)

<http://ww1.microchip.com/downloads/en/DeviceDoc/30000684B.pdf> (PIC)

<http://www.farnell.com/datasheets/1853479.pdf> (LED)

<http://www.vishay.com/docs/81341/tept4400.pdf> (photodiode)