LED dot matrix driving topologies and panelling

By: Willem Johannes de Villiers
206010591

Supervisor: Dr R. Wilkinson

Co-supervisors: Mr T. du Bruyn & Mr A. Bredekamp

Submitted to the Department of Electrical Engineering in partial fulfilment of the requirements for the Masters Technology in Electrical Engineering at the:

CAPE PENINSULA UNIVERSITY OF TECHNOLOGY

October 2014

Cape Town

Declaration

I, Willem Johannes de Villiers, submit this thesis in partial fulfilment of the requirements of the degree of Master of Technology in Electrical Engineering.

I hereby declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of Cape Peninsula University of Technology.

Signature of author ___________________

Date ___________________

Abstract

Abstract – Different LED dot matrix driving topologies and concurrent data transfer methods were investigated in this dissertation. The LED dot matrix driving topologies was first implemented and then tested. The designs tested were: the constant voltage, constant current, bi-polar and use of an existing LED dot matrix display driving integrated circuit (IC). All four driving topologies were evaluated to determine the brightest design. It was found that the constant current design performed the best, while the existing driving IC did not function at all. The bi-polar was a very close second while the constant voltage design performed the worst. The improvements in the bi-polar design with respect to the refresh rate were not enough to warrant the use of the bi-polar design in future permutations of the hardware. It was decided to improve the constant current driving topology and use this as the backbone for the LED dot matrix displays. Since the size of LED dot matrix displays are increasing, a smarter driving algorithm in conjunction with hardware needed to be developed. It was therefore decided to subdivide a large LED dot matrix display into smaller sections called panels. These panels can be connected in a horizontal, vertical, or a combination of the two to suit the user requirements. The panels were connected to adjacent panels to aid in determining the display size and configuration. Display graphics data was then sent to all the panels concurrently even though each panel still updated serially. This improved the brightness and refresh rate of the entire LED dot matrix display. Controlling this display required the use of a powerful processor as the algorithm had to subdivide graphics and distribute these to the panels to be displayed. Making use of pseudo-addressing, each panel connected to the system was assigned a temporary address. A Raspberry Pi computer was used to implement and execute the display algorithm. For connecting the Raspberry Pi computer to the LED dot matrix display panels, an interface card was developed. This interface card was only activated once the Raspberry Pi was connected. The on-board microcontroller on the interface card controlled the LED dot matrix display brightness. This was done by measuring the ambient light and adjusting the digital resistors placed on the LED dot matrix panels. The value of these resistors in conjunction with the current sources used in the hardware design determined the brightness of the display.

Keywords: panelling, LED dot matrix driving topologies, Raspberry Pi, interface card

Acknowledgements

I would like to acknowledge and extend my gratitude to the following persons who have made the completion of this thesis possible:

Dr R. Wilkinson for his guidance as my supervisor.
Mr T. du Bruyn for his guidance as my supervisor.
Mr A. Bredekamp for his guidance as my supervisor.
Mr J. Wheeler for the use of the facilities at the Centre for Instrumentation Research (CIR) at the Cape Peninsula university of Technology.
Mr G. Mutch for his discussion on ARM processors.
Mr S. Winberg for his discussion on FPGA's.
My friends and family for their support.

Contents

List of Figures

List of Tables

Glossary

ADC	-	Analogue to digital converter
ARM processor	-	Advanced RISC Machines processor
BGA package	-	Ball Grid Array package
Bi-polar	-	Two polarities
dsPIC	-	Digital Signal Peripheral Interface Controller
Electroluminescence	-	Luminescence produced electrically
FIFO	-	First In First Out
Firmware	-	Application specific software written to complete a specific purpose
FPGA	-	Field Programmable Gate Arrays
GaAs	-	Gallium arsenide
GaAsP	-	Gallium arsenide phosphide
IC	-	Integrated Circuit
I2C	-	Multi-master communication bus
LAN	-	Local Area Network
LCD	-	Liquid Crystal Display
LED	-	Light Emitting Diode
LUX	-	SI unit of illuminance, equal to one lumen per square meter
mcd	-	millicandela
OLED	-	Organic Light Emitting Diode
PCB	-	Printed Circuit Board
Pixel binning	-	Process of combining adjacent pixels together
PWM	-	Pulse Width Modulation
RAM	-	Random Access Memory
RGB	-	Red, Green, Blue
ROM	-	Read Only Memory
SD	-	Secure Digital
SIL	-	Single In Line
SPI	-	Serial Peripheral Interface
UML	-	Unified Modelling Language

1. Introduction

1.1 Subject

With LED displays being used more and more in industry due to their power requirements being lower than that of an LCD display, the need to design smarter display drivers arose. These displays were required to be brighter and larger for maximum viewer enjoyment.

This, however, had a limitation as the display brightness diminished with the size of the display. A new driving algorithm therefore needed to be found to increase the brightness without compromising the size or re-configurability.

In this thesis an improved driving algorithm will be explored and tested. This display algorithm will divide a larger LED dot matrix display into smaller displays or panels. These panels will be interconnected to determine the shape and size of the display. Also, the algorithm will be implemented in such a manner as to update each of these individual panels at the same time, thus increasing the brightness and the refresh rate.

1.2 Background

Due to the dearth of academic work on this specific topic, certain deductions about LED dot matrix display driving topologies and algorithms had to be made. Reviewing the practical work from industry it was found that the LED dot matrix displays are interconnected in a serial daisy chain manner. This works well, but these LED dot matrix displays all suffer from two major limitations, namely the display brightness and the refresh rate.

Due to this, the current sizes of LED dot matrix displays are limited. The larger the LED dot matrix display, the more the display is prone to the aforementioned limitations. These can be seen clearly at the airport or along our freeways in Cape Town. When the message changes on the display, the display updates from the top left corner to the bottom right corner.

Since these displays only show text messages, the refresh sequence is not really a problem, but graphics on these displays will not work as it will seem as if the graphics move out of synchronisation.

1.3 The need for research

The need to upgrade or improve the current LED dot matrix display driving algorithm has become necessary. This is due to display sizes increasing and the need for high speed graphics. The requirement from industry is to derive a way in which graphics can also be displayed without it being out of synchronisation.

1.4 Research Statement

In this dissertation four LED dot matrix driving topologies will be evaluated and the brightest of them chosen to further the design of a reconfigurable display. Furthermore a display algorithm will be designed and practically implemented were a LED dot matrix display will be divided into smaller displays or panels and all graphics updated in parallel. This display will be reconfigurable in size and layout and the display brightness will be adjusted automatically.

1.5 Main Contribution of the thesis

The main contribution in this thesis is to evaluate different LED dot matrix driving topologies and the development of a display software algorithm and hardware architecture in which the display will be divided into smaller panels. These panels will then be updated concurrently.

1.6 Outline of thesis

This thesis will be divided into six chapters and Appendices:

Chapter 2 is the literature review in which the background and the reason for this research will be explained.

In chapter 3 the hardware and software methodology will be discussed. The development of the hardware and software will be elaborated on.

Chapter 4 is devoted to the results from the evaluated designs.

In Chapter 5 the results obtained from the evaluated designs will be discussed, the process of adapting the hardware design explained.

Chapter 6 The conclusions and recommendations of the thesis. Future recommendations are included.

Appendix A	-	AS1116 schematic
Appendix B	-	A6276 schematic
Appendix C	-	Bi-polar schematic
Appendix D	-	Constant Voltage design schematic
Appendix E	-	Final display design schematic
Appendix F	-	Display interface card schematic
Appendix G	-	AS1116 PCB
Appendix H	-	A6276 PCB
Appendix I	-	Bi-polar PCB
Appendix J	-	Constant Voltage PCB
Appendix K	-	Final display PCB
Appendix L	-	Display interface card PCB
Appendix M	-	AS1116 bill of materials
Appendix N	-	A6276 bill of materials
Appendix O	-	Bi-polar bill of materials
Appendix P	-	Constant Voltage bill of materials
Appendix Q	-	Final display design bill of materials
Appendix R	-	Display interface card bill of materials

1.7 Summary

This chapter starts by giving the subject, background and the need for the research to be conducted on this subject. It also briefly explains what the aim of the research is and what will be done. Further the shortcomings in academia are noted as well as in the current circuit designs and possible solutions could be found. The thesis layout is also included.

2. Literature Review

2.1 History of LEDs

H.J. Round first observed electroluminescence from a diode in 1907 while experimenting with diodes for radios using silicon carbide (History of LED Lights, 2011). In the following years no advancement was made in the field of electroluminescence. However, in 1927 when Oleg Vladimirovich Losev made the first LED, there was still no practical application for the device (Timeline on the development of the Light Emitting Diode – Seer LED – Sustainable Lighting Solutions, 2011). This changed in 1961 when American scientists Robert Biard and Gary Pittman discovered infrared radiation from GaAs when current was passed through it (History of LED Lights, 2011). This gave the first practical use of LEDs and a patent was issued.

The first physical practical LED was invented by physicist Nick Holonyak in 1962 (Nick Holonyak, 2011). This development came as a result of studying wide band tunnelling in diodes made of GaAsP material (50 year history of the LED – 9/22/2010 – Electronics Weekly, 2011). Due to his work in this area, visible LEDs could be manufactured.

In 1968 gallium arsenide phosphide was used for the first time by the Monsanto Company for the mass production of LEDs (History of LED Lights, 2011). Due to this development, companies started using LEDs in handheld devices such as calculators, for example the HP9100A/B (HP 9100A/B, 2011).

Gallium aluminium arsenide was used in the 1980s for the production of LEDs. This made the LEDs ten times brighter and allowed for colour manipulation. The lower voltage requirements of these LEDs made them a viable replacement for everyday lighting (History of LED, 2011).

The first high brightness LED was produced in 1993 and paved the way for the production of white LEDs. However, the production of these white LEDs only started in 1995 by mixing blue and yellow light (Timeline on the development of the Light Emitting Diode – Seer LED – Sustainable Lighting Solutions, 2011).

With the recent development of organic LEDs made of carbon material graphene, a new generation of LEDs has been developed (Super material will make lighting cheaper and fully recyclable, 2011). Since these Organic Light Emitting Diodes or OLEDs are cheaper to manufacture, companies are using them in their products.

2.2 Development of LED displays

With the development of LEDs so did the application for which they are used. Initially LED displays were used in small handheld devices (HP 9100A/B, 2011). Other applications of LED displays being used today include alarm clocks, microwave ovens, home theatre systems and many more.

Due to the production cost of LEDs becoming cheaper and the brightness increasing LEDs are used more often in the large signage industry. An example of this can be seen at the Fenway Park in America where the largest LED dot matrix display is currently being used (“Green Monster” Has a New Sidekick: Mitsubishi Electtic Diamond Vision | Business Wire, 2011). This display shows game scores and graphics in real time.

2.3 LED display types

There are many different types of LED displays on the market. Some of these displays can be used as building blocks for bigger displays. Since the aim of the display is to be used outdoors and at sporting events, the display has to be of a considerable size and brightness.

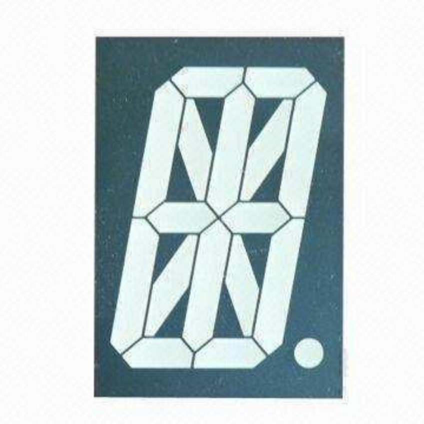


Figure 2.1: 16 Segment LED display (Global Sources, 2014.)

The display has to show not only numbers but alpha characters as well. This can only be done on either a fourteen or sixteen segment display, or a dot matrix display. Fourteen and sixteen segment displays do not have the ability to display graphics. In Figure 2.1 an example of a sixteen segment display can be seen (LED Alphanumeric Display, 2011). As the multimedia requirements of industry increase, the need for more than just numbers and alpha characters becomes more apparent. Since these displays are not available in very large sizes and cannot be stacked to reach the optimum size, a handmade design has to be created making use of individual LEDs. The construction cost in terms of labour and the limitations on the type of information that can be displayed is of such a nature that an alternative was investigated.

Dot matrix LED displays can be made with individual LEDs, or a premade pluggable unit can be bought. By making use of the premade pluggable unit, production costs can be lowered. Further, this type of display can show graphics and normal text. This enables the display to be used for more than just sporting events. It can be used as a billboard and information board in shopping malls. Multiple examples of premade pluggable dot matrix LED display units can be seen in Figure 2.2 on page 8.

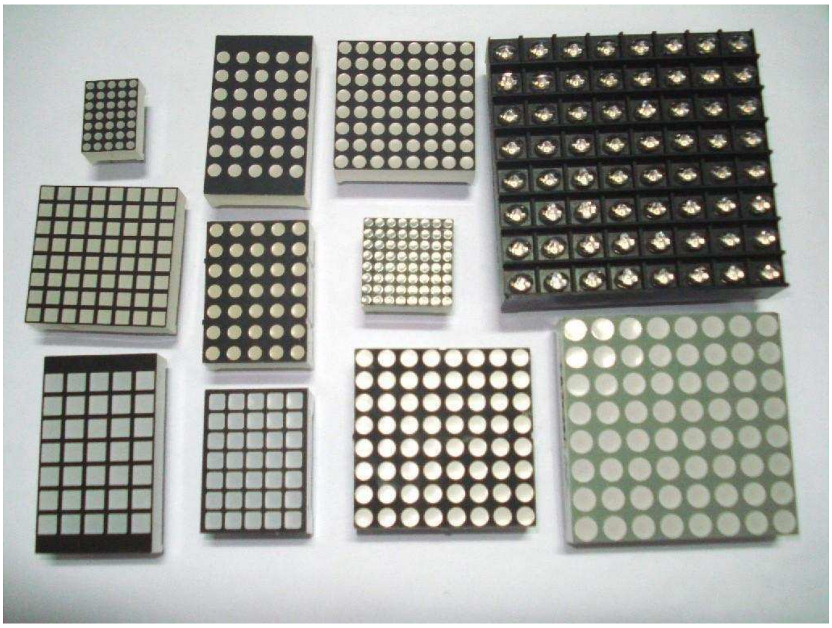


Figure 2.2: LED dot matrix display units (DIYTrade.com, 2014)

These units can be stacked or cascaded in such a manner that a larger display can be constructed. This is usually done in multiples of eight, making use of an eight bit microcontroller, as this enables easier driving. A 40-pixel by 56-pixel size is thus the smallest size of a LED panel that can be constructed when making use of a 5-pixel by 7-pixel LED dot matrix unit. Each of these LED dot matrix display units can display a character or symbol, hence a total of 40 characters could be displayed at any given time. These characters will, however, be too small to be seen at long distances away from a LED dot matrix billboard. Hence pixel binning will have to be used. This is the process in which adjacent LEDs are grouped together to make larger pixels. By doing this the resolution of the LED dot matrix billboard will be lowered, but the size of each character will be larger and appear brighter. This enables the billboard to be legible from long distances.

2.4 Display colour

These premade pluggable LED dot matrix display units are manufactured in standard colours and some in bi-colour as well as RGB. The general wavelength of LEDs is between 430nm to 800nm (LED Colors, 2011). The wavelength in basic form, determine the colour that the LED will emit. Of these, the human eye can only perceive wavelengths from 390nm to 780nm (Visible Light and the Eye's Response, 2011), which represent colours ranging from violet to dark red. Since this display is intended for outdoor use, other factors like weather conditions and the sensitivity of the eye to different colours need to be taken into account.

Weather conditions play a large role in the visibility of the display. In good weather conditions all colours can be perceived by the viewer within the visible spectrum. Hence any human visible spectrum LED can be used. In good weather the human eye is most sensitive to wavelengths of 0.5um which is the colour yellow (Light: Optipedia, Free optics information from SPIE, 2011). This can be seen in Figure 3.3 on page 19. However, good weather at outdoor events cannot be guaranteed and compensation for bad weather has to be made. Investigating how light scatters and why certain colours can be seen better than others in unfavourable weather was discovered by Lord Rayleigh (Preetham, A.J. 2002). He found that the percentage of light scattering is inversely proportional to the fourth power of the wavelength. This can be seen in Figure 3.2 on page 18 (Rayleigh Scattering - OPT Telescopes, 2011). It is noted that the percentage scattering for the shorter wavelengths is considerably more than that of the longer wavelengths. This means that the longer the wavelength, the lower the percentage of scattering that occurs and the further the display will be visible in bad weather conditions. Taking into account the visible spectrum of the human eye and Rayleigh scattering, colours with wavelengths from 570nm to 700nm should be used for the LED dot matrix billboard.

2.5 Driving topologies for LED display panels

A de-multiplexing system is used to display data on the LED dot matrix displays. This system allows for fewer wires to be connected to the display and less current is required to drive it. This is the case with all LED dot matrix display units that can be purchased commercially. Cascading these LED dot matrix display units into a large panel has to be controlled in the same manner. A basic block diagram of this method of displaying data on the LED dot matrix display can be seen in Figure 2.3 on page 10. This is the core design of all the driving methods for the LED dot matrix display that will be implemented (Floyd, T.L. 2006. *Digital Fundamentals Ninth Edition*).

For the display to operate, one line of data have to be displayed and then the next consecutive line have to be displayed until all the lines have been latched out (M. Stabile. *Controlling a Dot Matrix LED Display with a Microcontroller*). This enables an x,y co-ordinate controlling system (Abd El-Shafy Nafeh, A. (2010) Microcontroller-Based Moving Display Powered by Photovoltaic Energy). Each LED in this LED dot matrix display panel thus has an address. Making use of this structure limits the number of IO pins required from the microcontroller to control the dot matrix LED display as the data is serially sent from the microcontroller to the shift registers. These shift registers then latch the data out in a parallel manner to display the data on the LED dot matrix display.

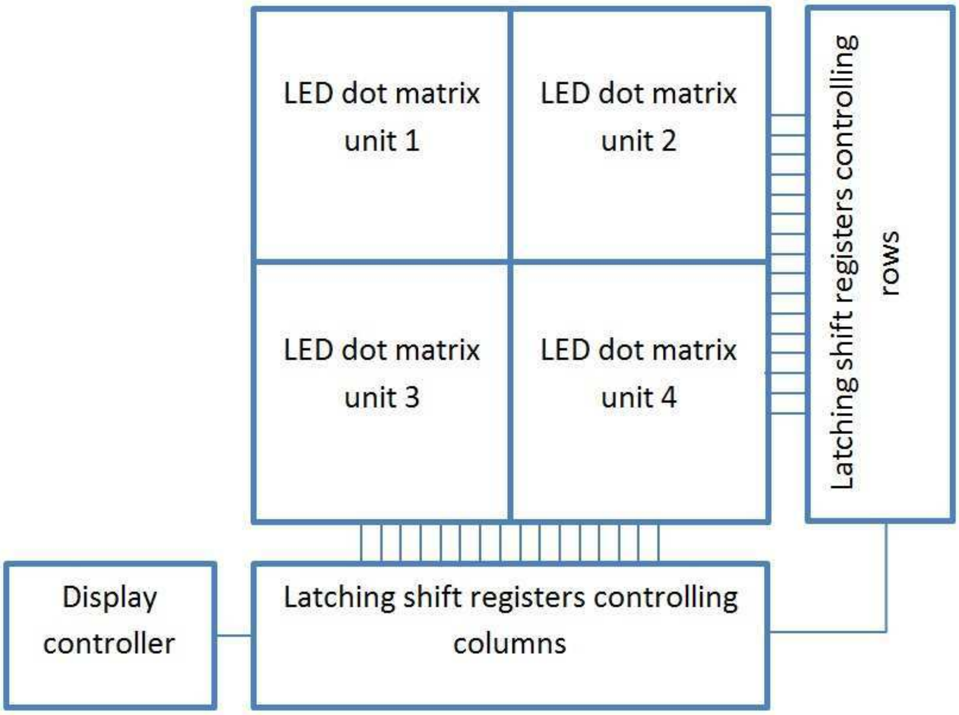


Figure 2.3 Cascading of LED dot matrix display units

Brightness is important in the design since the display will be used outdoors. Driving the LED dot matrix panels can either be done by making use of constant voltage or current systems. This does not yet change the brightness of the LED dot matrix panel. Since LEDs are current sensitive, it was decided to make use of constant current sources. This enables the easy change of the current to vary the brightness of the LED dot matrix panel itself. An alternative to this is to increase or decrease the on and off time of the individual LEDs by making use of a PWM system.

Heat is reduced by making use of the constant current driving topology. This makes the entire system more efficient. This is due to the use of current mirrors instead of resistors to control the flow of current. Since transistors have an on state voltage as well as the LEDs, the only voltage requirement is that of the components itself. There are resistors to control the current flow as will be the case of the constant voltage mode driving. Also by making the system a constant current system the supply voltage can vary considerably without having an adverse affect on the actual brightness of the individual LEDs themselves.

Another driving topology that increases the bandwidth, at which the LED can be pulsed, is that of bi-polar pulsing (DAWSON R.D. (1980)). This technique involves forward pulsing the LEDs as to illuminate them and then in the off state pulse them in reverse bias mode. By doing this, the overall refresh rate of the display can be increased. This means that the display can also be pulsed faster with a larger duty cycle as this will remove ghost images from the LED dot matrix display panel.

Constant current source drivers are commercially available and can also be purchased in packaging suitable for an assembly line. This enables the use of a pick and place machine and will lower production time and hence cost. These constant current shift registers can be purchased in either 8-bit or 16-bit packages. One such IC is the A6276 (A6276 16-Bit Serial Input, Constant-Current Latched LED driver. 2011). It has the ability to deliver current from 1mA to 90mA with the change of a single external resistor. This resistor will be replaced with a digital potentiometer connected to a microcontroller. By changing the resistance of this digital potentiometer the brightness of the entire LED dot matrix panel can be adjusted. A light sensitive sensor will be placed in such a manner so as to measure the ambient light and automatically adjust the display brightness. This will ensure optimum brightness in all light and weather conditions.

Another method of driving the LED dot matrix panels is by making use of the AS1116 IC (AS1116, 64 LED Driver with Detailed Error Detection. 2010). These devices can be cascaded to build larger LED dot matrix displays and panels. It has the ability to store the required display graphics information temporarily and keep the display active without the need to refresh the displayed graphics data constantly. Also the display brightness can be controlled making use of a single external resistor or in software. The device has an internal register that controls the brightness, making use of a PWM system. This allows for the brightness of each LED dot matrix module to be individually adjusted. An added feature of this IC is that it includes error detection circuitry. This can be used to relay maintenance information to the control circuit and inform the user of particular faults.

2.6 Flicker fusion threshold

Flicker fusion threshold refers to the highest frequency at which the human eye can detect flicker (Flicker fusion threshold Summary | BookRags.com, 2011). This is an important part of the refresh rate of the LED dot matrix display. Each line has to be updated at such a rate that the entire display seems continuous. The higher the refresh or update rate of the LED dot matrix display, the less likely it is that flicker will be detected. The lowest frequency at which each row of the LED dot matrix display can be refreshed, is 16Hz (Flicker fusion threshold Summary | BookRags.com, 2011). In the design it was decided to use a minimum of 50Hz for each individual row of the LED dot matrix display. A disadvantage of increasing the refresh rate much more than this is that the on time of each row of the display is so short that the brightness could be compromised.

2.7 Display communication protocol

The display can be controlled by either making use of the standard parallel daisy chain structure or by making use of a serial interface. Both of these have limitations as the data stream on the communication buses needs to be updated continuously. If the LED dot matrix display is not broken into units, the entire display needs refreshing at the same time. Hence the total brightness of the display will be lower.

By changing the structure in such a way, the LED dot matrix display operates in panels and the brightness of the entire display can be increased. This is done by making the off time of each LED less since each panel operates separately. This makes the design modular. Repair costs will be lower since a broken panel can be replaced with a working one on site and repaired off site.

Each of these LED dot matrix panels requires its own control logic to keep the data received on the communication bus in RAM and active on the display. These panels can either be connected in a serial manner or in parallel. The disadvantage of the serial manner is that the entire LED dot matrix display will update from one side to the other and this can lead to a lower overall data refresh rate of the display and compromise the brightness of the display.

Since overall refresh rate is part of the design criteria, a concurrent design structure will be implemented. This will enable each LED dot matrix display panel to be updated at exactly the same time. Hence the observer will not notice the update roll of data. This ensures that only the panels that require a data update are updated. The panels that required no change in state will remain active and will not use more processing power form the display card driver.

2.8 Display Card

This is the heart of the control logic as without this the display cannot function in concurrent mode.

Investigations in the use of microcomputers, FPGAs and a Raspberry Pi computer revealed that each of these could be used. It was, however, found with further investigation that the normal 8-bit microcontroller can only display static graphics. Making use of a 32-bit dsPIC showed a dramatic improvement in the performance of the 8-bit microcontroller. However, the design would not be expandable past 4 LED dot matrix display panels. This is due to the RAM requirements of the display driver algorithm.

A more powerful option was evaluated and that is making use of an FPGA with a built in microcontroller. This allowed for the graphics to be scrolled and the display to be expanded. However, the cost involved was of such a nature that the design was not used in this dissertation.

A cheaper alternative had to be found, thus a Raspberry Pi computer was investigated (Raspberry Pi. 2013.). It was found that this device consists of an ARM processor with a Linux platform. The RAM and ROM requirements were sufficient to run the planned display algorithm. Making use of the Raspberry Pi computer had another design effect. It divided the display card into two parts: the Raspberry Pi computer and the display interface card.

2.9 Problem statement

The goal of this research is to design an LED dot matrix display that will update graphics in parallel and automatically adjust the brightness.

Due to the demand in industry to design and build larger displays such as those used at major sporting events, there has been increased research into the effectiveness and power consumption of LEDs. These displays have to be visible in all weather conditions and still function correctly. Further the need for LED displays to show more than just numbers, has also arisen. Advertisers use these mediums as a way of generating revenue and hence some form of graphics is required to get their message across. These displays have to have the capability to vary brightness, as this will ensure lower power consumption at night. There is also a possibility of creating a portable system. One in which batteries, fuelled by solar or wind power can be used to power the display.

Increasing the brightness will be made possible by dividing a large LED dot matrix display into panels. These panels will have their own driving topologies and the entire system will be controlled making use of a Raspberry Pi computer and a display interface card.

2.10 Research objective

The literature review has shown the need by industry for larger and brighter LED dot matrix displays. These displays have to be of such a nature that they can be seen in all weather conditions and are easily adaptable for any application. The LED dot matrix display should:

- be reconfigurable for indoor and outdoor applications
- be expandable by making use of extra LED dot matrix display panels
- be able to alter brightness automatically
- be visible in bad weather conditions
- have the highest possible refresh rate without compromising brightness
- use modular design and
- be able to do pixel binning for lower resolution applications

The objective of this research is to design and develop the brightest possible LED dot matrix display from known configurations that can be expanded in an octal manner and controlled with a reconfigurable computer with a display interface card.

2.11 Research questions

1. Is it possible to design a LED dot matrix display in a panel structure?
2. Is it possible to design these panels with cost effective components?
3. Is it possible to design the display in such a way that it auto-corrects for ambient light so as to always be at the required brightness?
4. Is it possible to design the system to be completely modular?
5. Is it possible to have a concurrent reconfigurable computing display card and interface card controlling the entire display?

3. Methodology

This methodology will be separated into two major sections: hardware and software. This is done because the project involves both hardware and software design.

In the hardware methodology, four different LED dot matrix driving topologies will be evaluated. These consist of constant current, constant voltage, bi-polar and making use of an existing integrated circuit (IC).

Then the display card, Raspberry Pi and on-board panel detection circuitry will be discussed (Upton & Halfacree, 2012:11-26). The automatic brightness control and how this function works in conjunction with the display driver will also be considered.

The software design drawings are based on UML2.0 software modelling language (Pilone & Pitman, 2005:115-178). This models the software portion of this project.

A system overview is given by making use of Use Case Diagrams. This is followed by activity diagrams to show how each of the individual functions will be implemented.

A complete system software overview is then shown as an activity diagram to give the reader an understanding of how the system will function.

3.1 Hardware methodology

3.1.1 Initial system hardware overview

The initial hardware design was thought to consist of only a display card with a microcontroller and LED dot matrix display panels connected. This can be seen in Figure 3.1 below. It was, however, found that this implementation would not be sufficient for the system to operate, and new hardware was then evaluated.

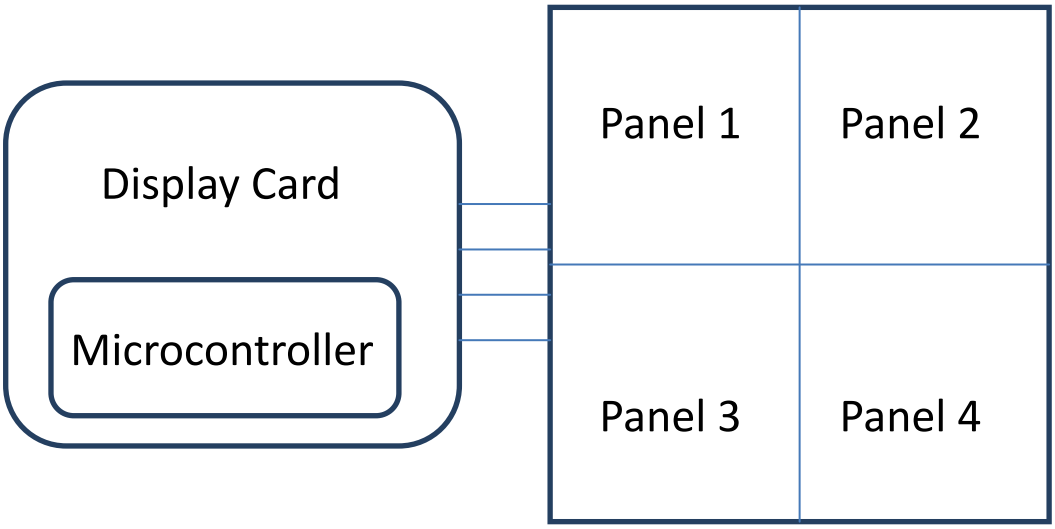


Figure 3.1: Initial hardware design (De Villiers, 2014)

Each of these LED dot matrix panels required driving circuits. It was decided as part of the scope to only design and test four different LED driving topologies. These were chosen to illustrate practically how LEDs are current devices and how the switch on and switch off times affect the brightness. These designs will be evaluated later in this chapter.

3.1.2 LED dot matrix modules

Before any hardware could be designed for the LED dot matrix display, LED dot matrix modules had to be chosen. The criteria for the modules were that they must be 8 by 8 and a minimum of 50mcd (The LED Light, Inc. 2014). The colour also needed to be chosen as the human eye reacts differently to different colour intensities (Light: Optipedia, 2011). The display needed to be visible in all weather conditions. Due to Raleigh Scattering it was found that the colour red would be the best choice (Rayleigh Scattering - OPT Telescopes, 2011).

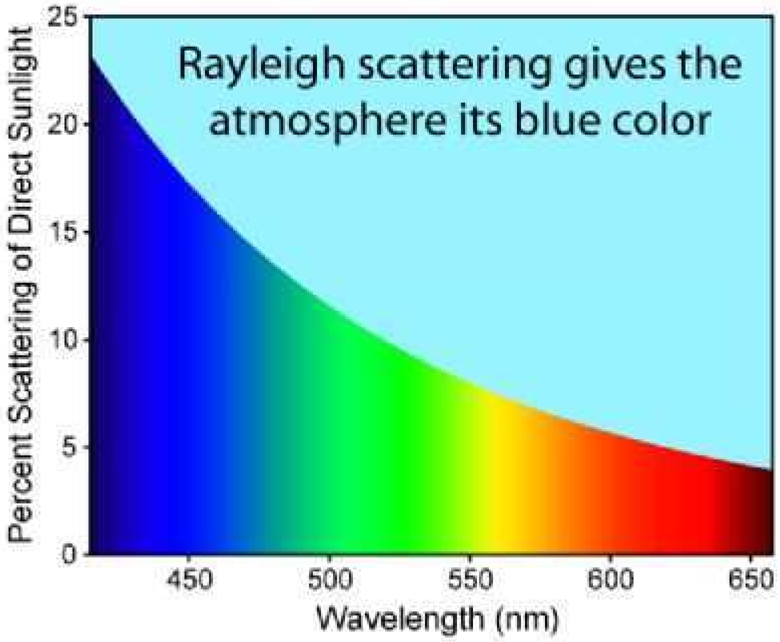


Figure 3.2: Rayleigh Scattering (Rayleigh Scattering - OPT Telescopes, 2011)

As red, or longer wavelengths is the preferred colour for all weather conditions, it was favoured. However the human eye is most sensitive in the green colour spectrum as seen in Figure 3.3 on page 19 (Light: Optipedia, 2011). Thus it was decided to compromise and select yellow LEDs for the modules as the human eye perceives this colour as fairly bright. Modules that have a peak output wavelength of roughly 580nm were investigated. It was found that the TC23-11SYKWA had a peak output waveform of 590nm which is ideal and hence this was chosen (TC23-11SYKWA, 2011). Design and construction of the LED dot matrix driving topologies could thus follow.

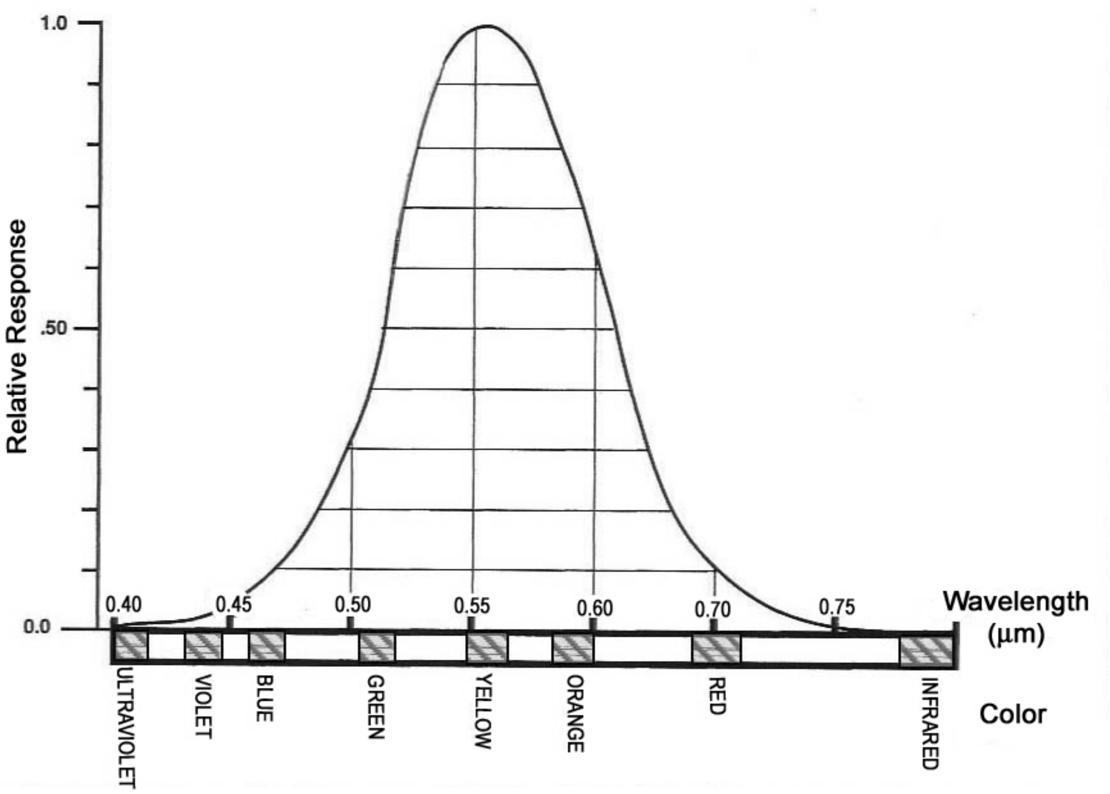


Figure 3.3: Human eye sensitivity to light (Light: Optipedia, Free optics information from SPIE. 2011)

3.1.3 LED dot matrix driving topologies

Driving by making use of constant current, constant voltage and bi-polar LED dot matrix driving topologies was done using shift registers. As this is a grid system it was done by making use of XY positioning software. An example of this XY grid matrix system can be seen in Figure 3.4 on the next page. Before displaying any information on the LED dot matrix panels one should note that any LED in the 8 by 8 LED dot matrix premade pluggable units can be addressed and switched on. This is done by enabling either an entire row or entire column. Then the corresponding column or row in which the LED resides, is addressed. Depending on the design, either the columns or the rows could be used. This will change the behaviour and the algorithm of the code in its totality. It was decided to design the code to address all the desired columns and then address all the desired rows. This made the display refresh from the top left corner to the bottom right corner in the current design. Since each of the driving topologies was placed in a 16 by 16 matrix and then connected in parallel with each other, the refresh rate increase as each panel of 16 by 16 updated in parallel to one another.

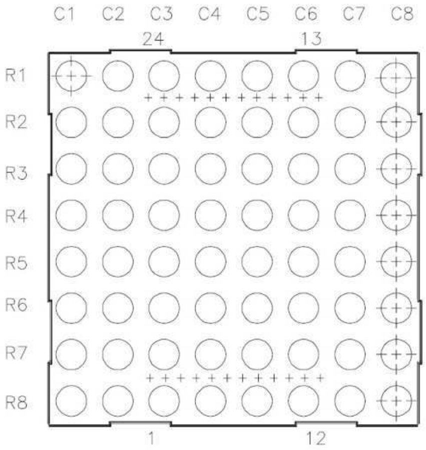


Figure 3.4: LED dot matrix XY co-ordinate (TC23-11SRWA, 2001)

Thus making use of four of the 16 by 16 LED dot matrix panels in the current design, the refresh rate will be the same as if only one of the 16 by 16 panels were used. If however, the driving of these panels was done by connecting them all in series with each other, than the refresh rate of the four panels and thus the entire display would halve for every 200% growth in size of the display.

This has two negative effects. The speed at which new information can be displayed is now four times slower by placing four panels in series. If this were scaled up to a hundred displays, this will result in a refresh rate of seven times slower than just one panel. The larger the display becomes the greater the risk of flicker, since the relevant LEDs in the dot matrix configuration can only be enabled for a brief period of time, after which power is removed from the particular LED and the next LED enabled. By connecting the 16 by 16 display panels in parallel, both the flicker and the refresh rate will be unaffected.

One major drawback of connecting the displays in parallel is that a more powerful processor will be required to run all the displays. Hence a Raspberry Pi was chosen because it has an 800MHz ARM processor (Raspberry Pi. 2013).

3.1.4 LED dot matrix driving using a bi-polar topology

Bi-polar driving of LEDs makes the switch on and switch off time of the individual LEDs quicker. This will aid in increasing the refresh rate of the entire display. It works on the principle of first forward biasing of an LED and then instead of just switching off the forward bias current it is instead reversed. Hence the LED is now reverse biased. This makes the switch off time of the individual LED quicker (DAWSON R.D. (1980)). In Figure 3.5 a basic schematic can be seen of Bi-polar pulsing of a LED.

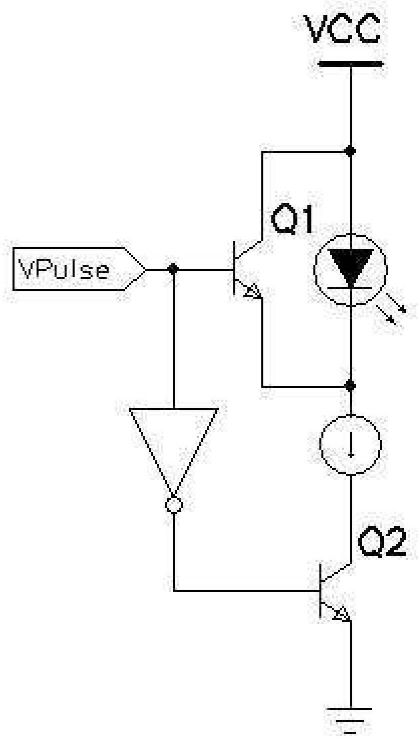


Figure 3.5: Bi-polar basic schematic (De Villiers, 2012)

Here the LED is on when transistor Q2 is on and transistor Q1 is off. To ensure this, an inverter is placed on the base of Q2 which will invert the signal received from V_{Pulse}. Once a logic high is placed on the base of the transistor Q1, it then places V_{CC} on the cathode of the LED and breaks the constant current source current flow to ground. This in turn switches off the LED and reverse biases the LED. It was designed in this manner so as not to place a high reverse voltage across the LED itself. Prolonged exposure to high reverse voltages and currents can damage the LED (Baldengineer, 2014). When transistor Q2 is switched on and transistor Q1 is off, this enables the current to pass from VCC to the constant current source and then to ground. Hence no resistor is required to limit the current as the current is predetermined by the constant current source. Should it happen that both transistor Q1 and transistor Q2 are switched on at the same time, transistor Q1 will enable the current to bypass the LED. This will ensure the LED does not come on in the event of a false signal. The chosen LED has an on state or forward voltage of 2V at a minimum (TC23-11SYKWA, 2011). This is more than the chosen transistor V_{CE} of 0.7V. Current will flow from VCC though transistor Q1 to ground via transistor Q2 (PBSS5320T, PNP Low V_{CEsat} (BISS) transistor. 2004).

The constant current source chosen was the A6276 latching constant current shift register (A6276 16-Bit Serial Input, Constant-Current Latched LED Driver. 2011). This shift register will replace the constant current source and transistor Q2 in Figure 3.5 on the previous page. Since the LED panel consists of four LED dot matrix units each, and each of these units have 64 LEDs mounted on them, it was decided to limit the component count by making use of a standard 16 by 16 matrix. This enables the use of the A6276 16-bit, latching constant current source driver. The IC does, however, not supply the current but rather sinks the current to ground. The schematic representation can also be seen in Figure 3.5 on the previous page. Current can also be adjusted by making use of a single resistor or in the case of the schematics designed, a digital potentiometer which can be seen in Appendix C on page 95. This makes the change of brightness of the display quite simple from a microcontroller side. The microcontroller can either use a light sensor configured for use with the analogue to digital converter, or the signal can be digitised and transmitted via the I2C or SPI bus. The micro controller or processor will then interpret the received signal and either brighten or dim the display accordingly.

The transistors used to drive the circuit were chosen to be the 2N2222 and the PBSS5320T (PN2222A/MMBT2222A/PZT2222A, NPN General Purpose Amplifier. 2009. PBSS5320T, PNP Low V_{CEsat} (BISS) transistor. 2004). As this can become a viable product to sell in the future, it was chosen for its ease of availability. It was also chosen for the current that it can supply as well as sink, depending on the configuration in which it is placed. Base resistors were chosen only to limit the current supplied by the logic to drive the LED dot matrix panel. It was decided to make use of 100kΩ as this enables the use of Field Programmable Gate Arrays (FPGA's) or microcontrollers as the driving logic.

3.1.5 LED dot matrix driving topology using existing driving IC

The next driving topology was done by making use of a 64 LED dot matrix driver. The AS1116 was used (AS1116, 64 LED Driver with Detailed Error Detection. 2010). This IC has an impressive number of features that also includes flicker control, display brightness and error detection. Once placed in the error detection mode, it can report back to the microcontroller if one or more of the LEDs on the LED dot matrix panel is faulty. This can be a handy feature as maintenance on these displays can become an expensive proposition when the display has reached a significant size. The new LED dot matrix displays placed along all the major routes in Cape Town also feature this mode and the display enters into test mode every so often. It then transmits via a wireless link to the control room if the display requires maintenance. As for display brightness, making use of this IC, it is adjustable via a single resistor or digital potentiometer controlled by the microcontroller. A single AS1116 and a 64 LED dot matrix configuration can be seen in Figure 3.6.

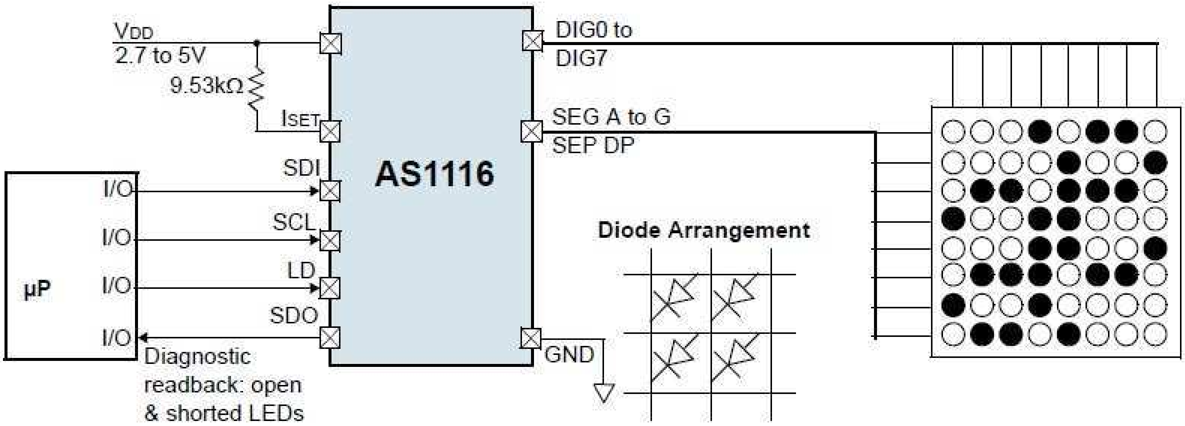


Figure 3.6: Single configuration of AS1116 (AS1116, 2010)

The microcontroller receives brightness information from a LUX sensor and then translates it back to display brightness. During the day or in more brightly lit conditions the display will be close to or at its brightest setting, while at night or indoors, the display will be switched on with less current. This limits viewer discomfort and optimises the display for use in any configuration and placement.

Display flicker control limits the necessity to programme data from the microcontroller or control logic of the display driver. This is tied directly to the brightness. The brighter the display, the less likely it is that the viewer will detect flicker on the display as it updates. In low light conditions though, when the display is set to be less bright, flicker might be detected if this is done by making use of programming only.

These devices can also be cascaded to form a larger more versatile display. This can be seen in Figure 3.7 (AS1116, 2010). Also a completed schematic can be seen in Appendix A on page 86.

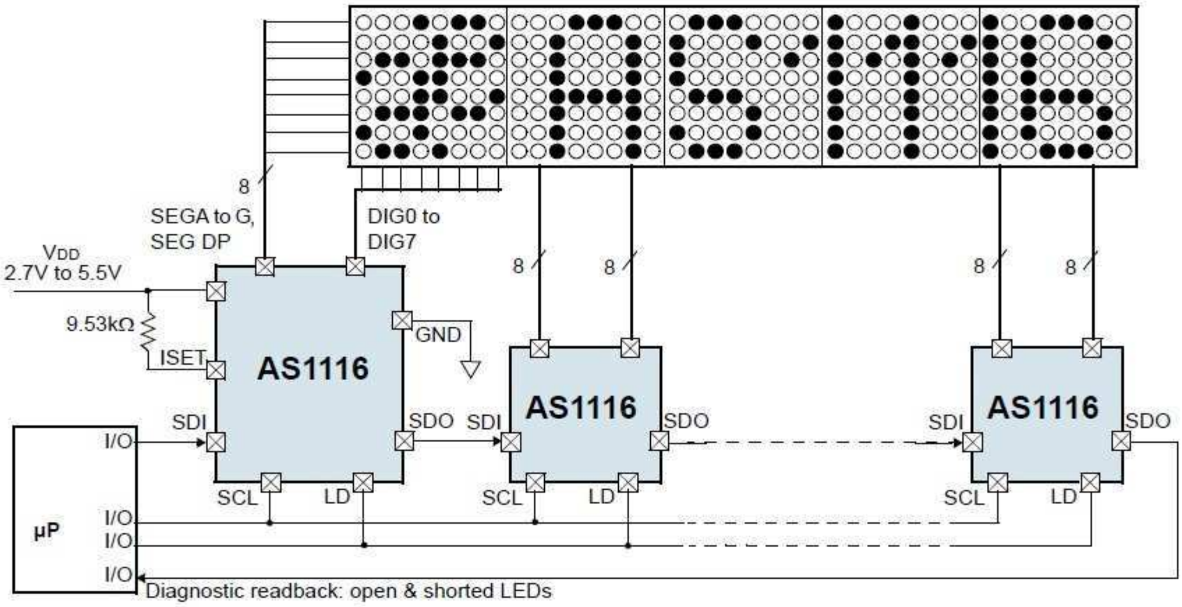


Figure 3.7: Cascaded configuration of the AS1116 (AS1116, 2010)

It can be seen that this IC uses the SPI protocol for data transmission and communication (Microcontroller Interfacing Techniques, 2005). The SPI interface is capable of updating at frequencies of up to 10MHz. This means that a display with this as its backbone design can be significantly large before multiple panels are required. This is due to the flicker fusion threshold (Flicker fusion threshold Summary | BookRags.com. 2011).

3.1.6 LED dot matrix driving topology constant current

A constant current driving topology was the next to be investigated. In basic terms a constant current will be passed though the LEDs as can be seen in Figure 3.8. For a complete schematic refer to Appendix B on page 90.

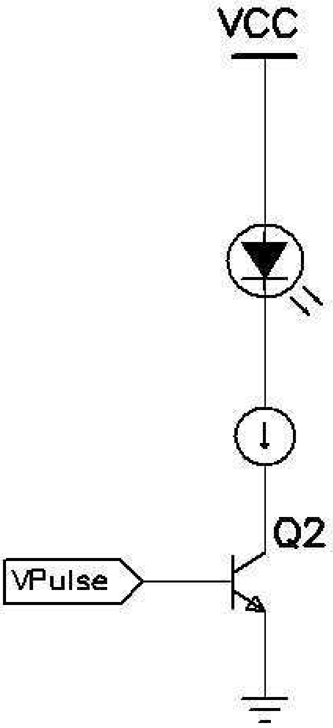


Figure 3.8: Constant current (De Villiers, 2012)

This current can be changed to vary the brightness of the entire display. Pulse width modulation can also be used to adjust the display brightness (Curran, 2010). The A6276 IC was chosen as the backbone for the design as it is a latching constant current shift register (A6276, 2011). Data is then serially clocked into the device using the SPI communication protocol. The selected latching shift registers will then sink the required current as set by the external resistor, and drive the LEDs. This external resistor can then be used to vary the brightness of the display. A functional block diagram of the A6276 IC can be seen in Figure 3.9 on the next page.

Making use of the latch enable pin enables all the required data to be clocked into the IC before the actual constant current sources are activated. This latch enable pin aids in the removal of ghost images on the LED dot matrix display. Ghost images form when there is a change in the displayed data and the display is not cleared while the new data is being inserted into the shift registers.

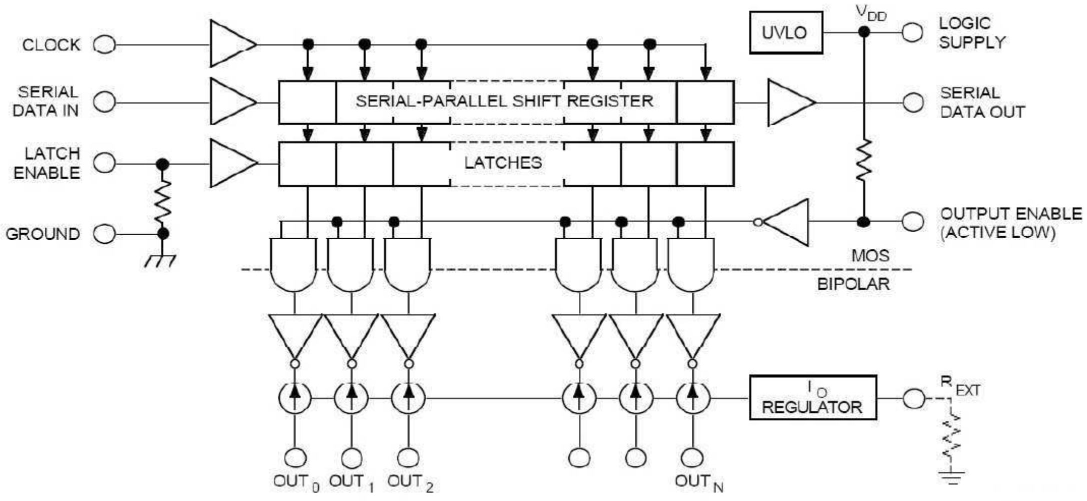


Figure 3.9: Functional block diagram of the A6276 (A6276, 2011)

3.1.7 LED dot matrix driving topology constant voltage

Finally a constant voltage design was evaluated. The circuit was designed taking into account the forward potential required for a single LED to switch on. The resistor was chosen to ensure that the maximum current rating of the LEDs would not be exceeded. In this design the forward voltage of the LED is 2V with a forward current of 20mA (TC23-11SYKWA, 2011). The maximum peak current is rated at 175mA. This can only be done if the pulse width is 1/10 and the frequency of the pulse is 10kHz (TC23-11SYKWA, 2011). During the operating phase of the display, the 1/10 duty cycle at a frequency of 10kHz cannot be guaranteed. Due to firmware malfunctions, it was decided to design conservatively and keep to the maximum continuous current of 20mA.

Shift registers were used to aid in the placement of the different powered and unpowered LEDs in the display matrix. Making use of shift registers the display will update line by line in the configuration. Each line has a limiting current resistor of 150Ω as seen in Figure 3.10 on page 27. These limiting resistors, limit the current in each column and are switched to active making use of the HEF4094BT shift registers (HEF4094B, 8-stage shift-and-store bus register. 30 October 2008).

In order to simplify the programming of the display, the circuit was designed in such a manner that a logic high on the row and column driving transistors would enable them. Transistor Q3 sinks the current while transistor Q2 sources the current after being enabled using transistor Q1. 5V was used for the display supply as the shift registers also operate at this potential. The complete schematic can be seen in Appendix D on page 101

Calculations:

$$V_{Total} = Vcc - V_{F\ LED} \qquad \text{Equation 3.1}$$

$$= 5V - 2V$$

$$= 3V$$

$$R_{LED} = V_{Total} / I_{LED} \qquad \text{Equation 3.2}$$

$$= 3V/20mA$$

$$= 150\ \Omega$$

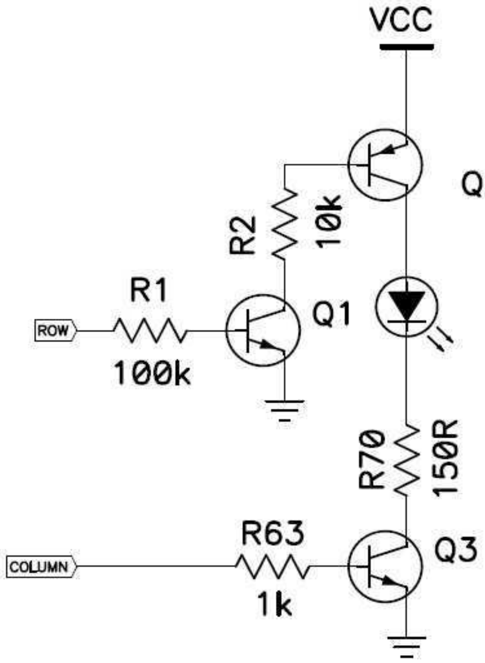


Figure 3.10 Constant Voltage (De Villiers, 2012)

A limitation of this design is the brightness control. It can only be set by making use of pulse width modulation (PWM) techniques (Curran, 2010). Thus flickering of the displayed information might be visible to the viewer.

3.1.8 Initial Display card design

Initial designs of the display card used a microprocessor or microcomputer. This was done by making use of the PIC16F690 (PIC16F685/687/689/690, 20-Pin Flash_Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology. 2005). Even though this microcontroller is what is known as a single core or chronological processor, it can, with the correct algorithm, run the ports in concurrent mode. It was soon discovered that an upgrade to the system needed to be designed as the microprocessor ran out of RAM. The calculations needed for the display algorithm required double floating point numbers for character placement in a large display and hence a new or more powerful microcomputer needed to be chosen for the display card.

A more powerful microcontroller was chosen for the design of the new display card. The dsPIC33FJ128MC802 (dsPIC33FJ32MC302/304,dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04, 2011). This digital signal processor has the ability to manage the use of double floating point numbers as it has an onboard mathematical facility. Making use of the onboard phase lock loop the digital signal processor can run up to 40MHz. This will work perfectly if the display itself only consists of a maximum of 4 panels to control in concurrent mode. However, since the display could consist of multiple panels and not just four, it was decided to find an even more powerful processor.

FPGA’s were considered but cost became the next factor in the design. A suitable FPGA with internal microprocessor for optimal speed was too costly for the design. Hence an ARM processor was chosen. Most powerful ARM processors are in a BGA form factor and hence not easy to work with in a prototype phase (Toshiba, 2000). So a Raspberry Pi was chosen (Raspberry Pi, 2013). At the heart of the Raspberry Pi computer is an ARM processor and a software development platform based on Linux. The display driving algorithm was implemented in embedded C.

3.1.9 Raspberry Pi computer

The Raspberry Pi computer was chosen to drive the display panels as it has an on-board 800MHz ARM processor (Raspberry Pi, 2013). The LED dot matrix display panels in parallel requires more processing power than that which can be provided by the average microcontroller. The algorithm used to drive any dot matrix display requires the use of double floating point numbers.

This computer had external IO ports which were unique to the specific type of computer. These IO port pins could be used to either drive external electronics or to receive information from external electronics. The Raspberry Pi computer model used for this project was equipped with 17 IO port pins. These IO port pins can only work on a 3V logic level. Also with this computer came 250Mb random access memory (RAM) and an Secure digital (SD) Card slot. The SD card was used as the hard disk drive of the computer. Here all the software required to drive the LED dot matrix panels was programmed.

3.1.10 Display card chosen

The display card consisted of a 26-pin boxed header connector, two 8-pin boxed headers, a PIC16F819 microcontroller, four RJ45 connectors and a TEMT1000 opto-transistor (2532 Series, 2012; PIC16F818/819, 2004; TEMT1000, 2008; RJ-45, 2003). The 26-pin boxed header connector connects to the Raspberry Pi computer via a ribbon cable, while the two 8-pin boxed header connectors connected to the display panels using SIL connectors for layout detection and pseudo-addressing. The PIC16F819 microcontroller in conjunction with the TEMT1000 opto-transistor was used to measure ambient light and adjust the LED dot matrix display brightness. The RJ45 connectors connect the four LED dot matrix panels data input to the display card via straight LAN cables. The display interface card schematic can be seen in Appendix F on page 113.

The display card did not have the on-board intelligence to show a message on the display, but was primarily there to interface the Raspberry Pi to the external hardware. The only onboard intelligence was the display brightness control. The display brightness could also be done with the Raspberry Pi. It was decided that to lighten the load and make it seem more streamlined to the viewer that it would be implemented separately. This immediately allowed the system to operate concurrently.

Brightness control or the logic was only enabled once the Raspberry Pi was connected. The 5V logic from this device was connected to the external interrupt pin of the microcontroller. The main reason for using the 5V logic from the Raspberry Pi instead of the 3V logic was the microcontroller the PIC16F819 is a 5V device (PIC16F818/819, 2004). Once the microcontroller was active, it would start reading the ambient light intensity. This was done making use of the opto-transistor the TEMT1000 (TEMT1000, 2008).

Calibration of the opto-transistor was required. Calibration was done by adding a series potentiometer as seen in Figure 3.11. Stability of the light sensor can be done in two basic methods: one in software, and the other with a capacitor. In software the code can be written to slow down the sampling from the microcontroller.

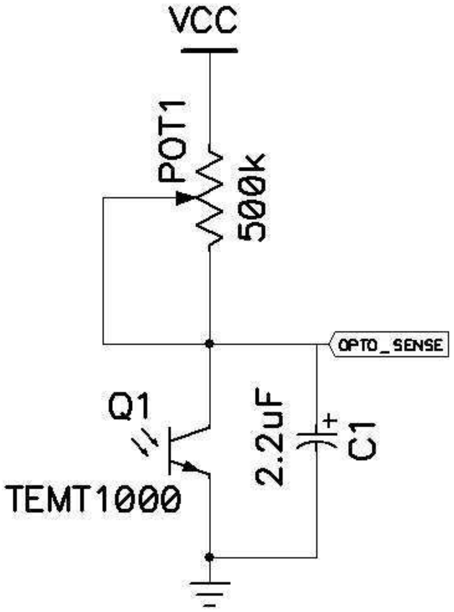


Figure 3.11: Ambient light sensing (De Villiers, 2012)

Making use of the software method and slower sampling reduced the number of quick changes in display brightness with quick fluctuations in ambient light. This stopped nuisance display brightness adjustments as perceived by the viewer. Also by slowing the sampling rate, unwanted harmonics could be removed from the circuit.

The second method was to add a small capacitor in parallel with the opto-transistor. This capacitor would act as a form of temporary memory or hysteresis. By doing this quick changes in ambient light would not affect display brightness and display brightness flickering could be eliminated.

By using the capacitor and slowing the sampling rate in software, an even more stable system can be achieved. Reducing the unwanted harmonics and building in a memory buffer allowed for a delay in changing display brightness. This would make the display more visually appealing to the viewer.

To calibrate the display brightness sensing on the microcontroller side, two potentiometers were connected. The configuration can be seen in Figure 3.12.

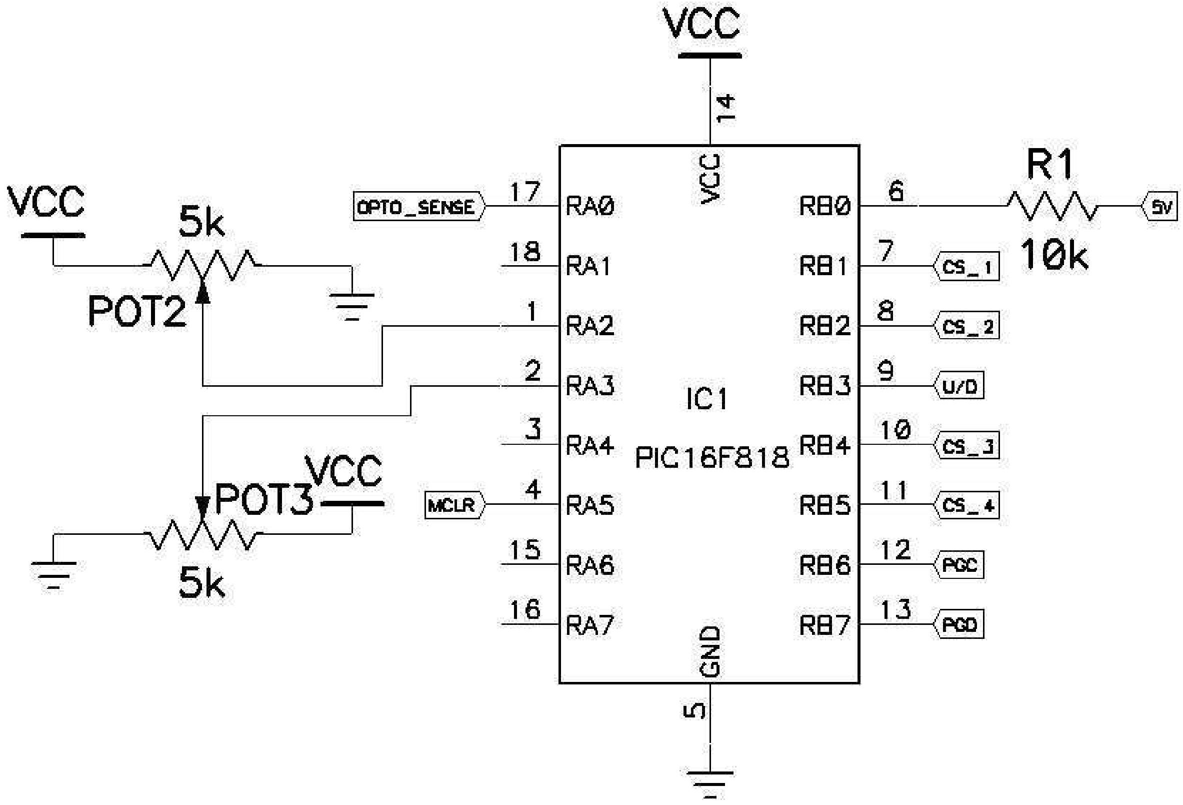


Figure 3.12 Microcontroller ADC calibration circuit (De Villiers, 2012)

The first potentiometer POT2 was connected to the V_{REF-} and the second POT3 was connected to V_{REF+} (PIC16F818/819, 2004). Both of these potentiometers are connected in such a way as to adjust the voltage reference on these pins. The relative ground and full scale deflection could then be adjusted individually. Doing this the resolution of the ADC could be adjusted.

The 10-bit ADC of the PIC16F819 was used (PIC16F818/819, 2004). If the two V_{REF-} and V_{REF+} had not been included, then the internal default was GND and VCC respectively. Then the minimum step size would be 4.88mV.

Step Size_1 = Full scale/Number of steps Equation 3.3

= 5V/1024 (10-Bit ADC)

= 4.88mV

Now by changing only the V_{REF-} and V_{REF+} potentials to have a difference of 2V for full scale, the step size decreased to 1.95mV.

Step Size_2 = Full scale/Number of steps

= 2V/1024 (10-Bit ADC)

= 1.95mV

This meant that smaller changes in the ambient light could be detected by the ADC to which the opto-transistor was connected. The sensitivity of detecting changes in ambient light had thus increased by a theoretical factor of 2.33.

Change in sensitivity = Step Size_1 / Step Size_2

= 4.55 / 1.95

= 2.33

3.1.11 Display card to Raspberry Pi interface

Interfacing the Raspberry Pi to the LED dot matrix display panels could not be done directly as the Raspberry Pi is a 3.3V device and the logic from the display panels return 5V (Upton & Halfacree, 2012:180-207). The MC14013B IC could accept a logic high from 2.75V in the current configuration (MC14013B, 2005). Thus the 3.3V from the Raspberry Pi output logic would be sufficient and no converter from the Raspberry Pi to the LED dot matrix display panels was required. From the LED dot matrix display panels, logic needed to be implemented to prevent an accidental over-voltage on the Raspberry Pi.

Instead of designing this on each of the panels it was implemented on the display card as these electronics would be used repeatedly but never at the same time. It was decided to make use of a voltage divider circuit as this was the simplest implementation. Making use of Kirchhoff's voltage divider law the resistor values was calculated (Glasspoole, 2008: 20-22). E12 resistors were chosen for cost and availability (Glasspoole, 2008: 12). The schematic can be seen in Figure 3.13.

$$\begin{aligned} V_{\text{Raspberry Pi}} &= (R_2 / (R_1 + R_2)) * V_{\text{panel}} \qquad\qquad \text{Equation 3.4} \\ &= ((1k / (1k + 560R)) * 5V \\ &= 3.2V \end{aligned}$$

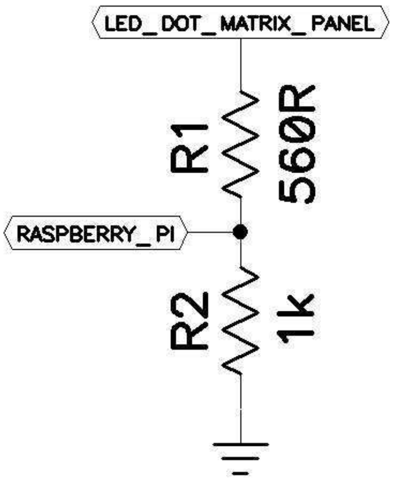


Figure 3.13: Raspberry Pi to LED dot matrix display panel interface (De Villiers, 2012)

The perceived input voltage to the LED dot matrix display panel was still 3.3V as the voltage divider circuit had almost no effect. The reason for this is the input current to the MC14013B IC was a maximum of 0.1uA (MC14013B, 2005). Calculating the potential difference across R1, the 560R in Figure 3.13 yielded 56uV. Thus the input voltage as perceived by the LED dot matrix display panel, was 3.299V. This is still within the accepted tolerance for the electronics to register a logic high on the input.

$$\begin{aligned} V_{Drop} &= I * R && \text{Equation 3.5} \\ &= 0.1uA * 560R \\ &= 56uV \end{aligned}$$

$$\begin{aligned} V_{LED\_dot\_matrix\_display\_panel} &= V_{Raspberry\ Pi} - V_{Drop} && \text{Equation 3.6} \\ &= 3.3V - 56uV \\ &= 3.299V \\ &\approx 3.3V \end{aligned}$$

This voltage divider circuit was used only twice as the rest of the logic for the LED dot matrix display panels only received logic levels form the Raspberry Pi and never returned any logic information to it. These two voltage divider configuration interfaces are linked to the horizontal and vertical display layout detection circuitry.

To detect the vertical and horizontal display layout a D-latch was used. The interface between the display card and the LED dot matrix display panels was done making use of SIL connectors on the panel side. On the display card side small 8-pin boxed headers were used: one set for vertical and the other for horizontal display layout detection. This will be described in more detail when the LED dot matrix display panel is discussed.

3.1.12 Display card to panel interface

To interface the data lines from the display card to the LED dot matrix display panels an RJ45 connection with a straight network work cable configuration was chosen (RJ-45, 2003; Broadcast Software International). Only four of these were implemented on the display card as it was decided that interfacing four displays would be the scope of the study.

The connections in these RJ-45 connectors were as follows: Vdc, GND, Output enable, Strobe, Clock, SDI, Up and Down and chip select. These interface directly with the A6276 from the Raspberry Pi connection via the display card (A6276, 2011). This will be discussed in more detail later in the chapter.

3.1.13 Display panel detection

From the previous LED dot matrix display panel logic designs, tests were done and it was decided to use the constant current design as the base design was done and already discussed earlier in this chapter, only further developments will be discussed here. The completed schematic can be seen in Appendix E on page 107.

To determine the LED dot matrix display layout a D-latch was used. Since each panel may not have any intelligence, it was decided that the MC14013B would be used (MC14013B, 2005). This IC had two D-latches and thus one of the D-latches was used for the horizontal interface and the other for the vertical interface. This was a slow method of determining the layout of the LED dot matrix display, but it required the lowest count of wires and allowed for pseudo addressing. Thus the LED dot matrix display panels could be arranged in any rectangular shape or size. If a hardcoded address had been used, then it would require a form of memory to be implemented on each of the display panels and the display would not be versatile.

Making use of the D-latch a system of daisy-chaining the LED dot matrix display panels vertically and horizontally was designed. Making use of SIL connectors and placing them off centre allowed for the panels to only be connected in a specific way. Refer to Figure 3.14 on the next page.

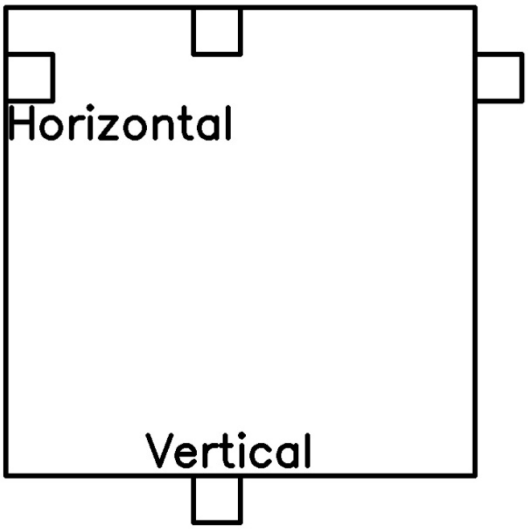


Figure 3.14: Panel detection connectors placement (De Villiers, 2012)

Also by placing the connectors off centre the panels could be daisy chained to build any rectangular configuration. An example of a daisy chain connection configuration is shown in Figure 3.15. Here four panels have been connected in a horizontal daisy chain connection.

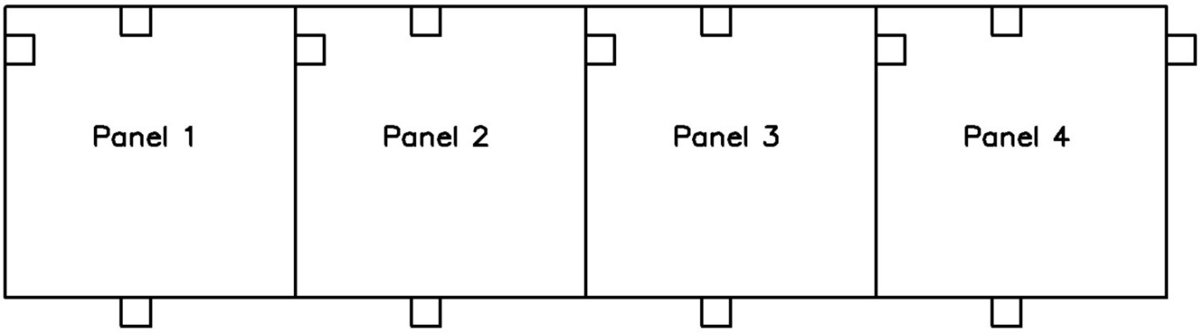


Figure 3.15: Horizontally daisy chained connection (De Villiers, 2012)

To determine the configuration of the LED dot matrix display all the D-latches were cleared, making use of software. Then a single high bit was pulsed into the first D-latch in the configuration on panel 1. This bit was then pulsed to the next D-latch on panel 2 until the software could detect it on the output of panel 4. Once the single high bit had been recognised, the number of panels in the system was returned. This number would then aid in the auto configuration of the LED dot matrix display. This entire design is based on the first in first out or FIFO system configuration of connecting shift registers (Floyd, 2006: 492 - 527).

3.1.14 Display panel brightness adjustment mechanism

Since the constant current design for the LED dot matrix display panel was chosen and the design was based on the A6276 IC the adjustment of brightness could be done with a single resistor (A6276, 2011). This resistor changed the output current of each of the 16 NPN constant current drivers.

Two design possibilities were implemented, one were the user was required to manually adjust the brightness using a potentiometer, and automatic adjustment using a microcomputer. Since manual adjustment was not practical and the design specification stated that the brightness had to be automatically controlled, it was decided to use a digital potentiometer.

Digital potentiometers are available in SPI, I2C or in a simple up/down logic configuration with a chip select pin (Microcontroller Interfacing Techniques, 2005). They are also available in volatile and non-volatile configurations (DS1666, 1995; MCP4021/2/3/4, 2006). It was decided to make use of a non-volatile and simple up/down logic digital potentiometer. The MCP4021 was chosen (MCP4021/2/3/4, 2006).

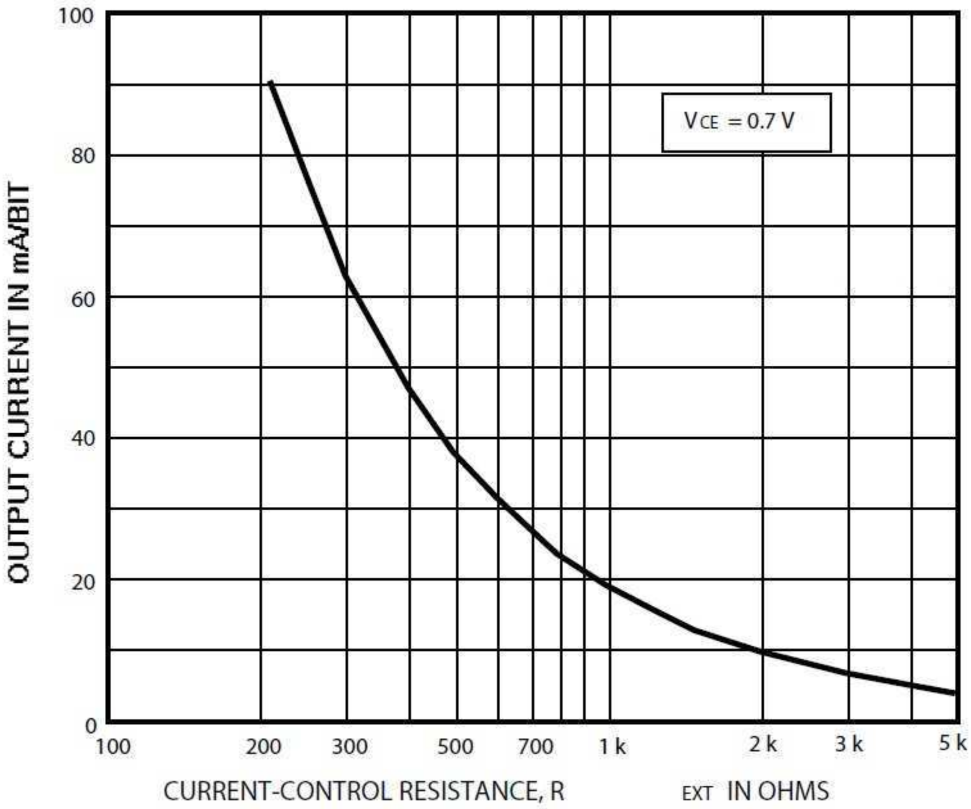


Figure 3.16: External resistance and output current characteristic curve (A6276, 2011)

Each LED dot matrix display panel made use of these. There was only one A6276 IC per panel (A6276, 2011). A resistor was placed in series with the digital potentiometer to aid in limiting the current to the 16 NPN constant current drivers. The total minimum resistance measured across the digital potentiometer and the series resistor was 220Ω. According to the A6276 datasheet the maximum output current was then 85mA per NPN constant current driver (A6276, 2011). This was well within the safe operating range of the LED dot matrix modules used as the peak forward current was rated at 175mA (TC23-11SYKWA, 2011). It was also noted that the higher this total resistance became the lower the output current per NPN constant current driver would be. This can be seen in Figure 3.16.

3.1.15 Final system hardware overview

This is the overview of the final hardware implementation that was done. Here it is seen that the display card was divided into two parts and can be seen in Figure 3.17

on the next page. This being the Raspberry Pi computer and the display interface card.

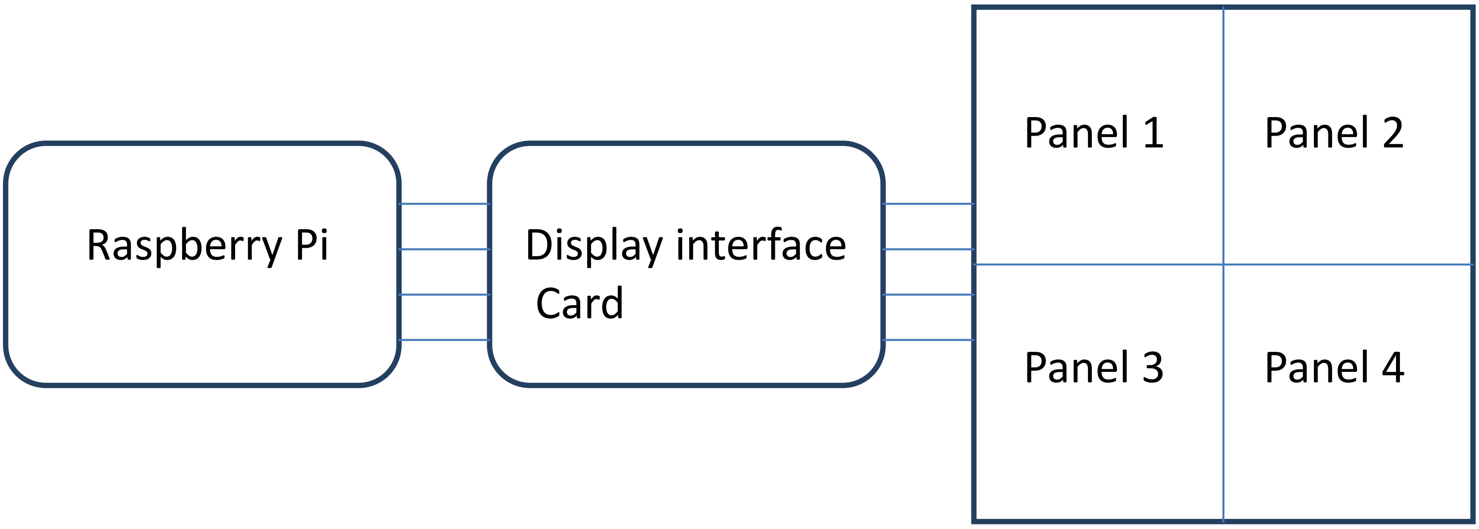


Figure 3.17: Final system hardware overview (De Villiers, 2014)

Each of the panels uses the same constant current driving topology. This was evaluated earlier in the chapter.

3.2 Software Methodology

The software was designed making use of incremental and iterative development techniques (Dr. Cockburn, May 2008).

Starting with an incremental development platform the project was divided into smaller parts or sub-systems. These were implemented and tested as each individual element of the program was completed.

Then making use of the iterative development techniques the existing working code was redesigned to ensure better efficiency and use of the embedded resources available for the project.

To start the development, requirements were gathered and refined. These were then taken and a conceptual design was done on paper. This design was then implemented and tested. Coding and testing was done on a modular basis, followed by integration testing.

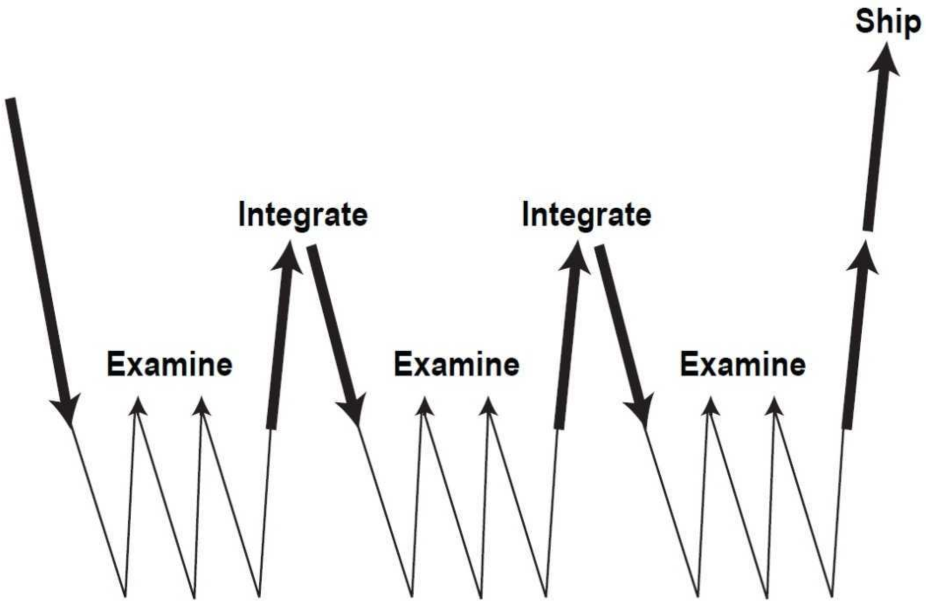


Figure 3.18 Incremental and Iterative development (Dr. Cockburn, May 2008)

Examination of the test results revealed that the code required had to be refined. After refining the code, it was integrated into the project as a whole as a working module. This development technique was used for all the code generated during this course of this project. A graphical view of this development technique can be seen in Figure 3.18 on the previous page.

Making use of this development technique did slow the production of the project. Each module was integrated into the project after testing to reduce the need for system debugging.

If only iterative development had been used, the project code could have been done faster, but might have required large amounts of debugging in the end to discover where the faults were. Therefore development speed was balanced with the requirement for extra debugging.

3.2.1 Functional requirements

Table 3.1: List of functional requirements

3.2.2	System Use Case
3.2.3	Use Case Extended
3.2.4	Display Message
3.2.5	Auto Configuration
3.2.6	Panel Layout Detection
3.2.7	Horizontal Panel Detect
3.2.8	Vertical Panel Detect
3.2.9	Text Size
3.2.10	Scroll Text Configuration
3.2.11	Message Activity
3.2.12	Display Configuration
3.2.13	Normal Operation
3.2.14	Display Brightness Control

In Table 3.1 the list of functional requirements are stated. Each of these will be discussed later in this chapter.

3.2.2 System Use Case

Depicted in Figure 3.19 is the system use case diagram. This shows the highest view of the system: a level indicating the key functionality and role players.

This diagram did not show how the system would operate. It did however show the key functionality of the system. It also indicated the high level or direct relationship between modules within the system.

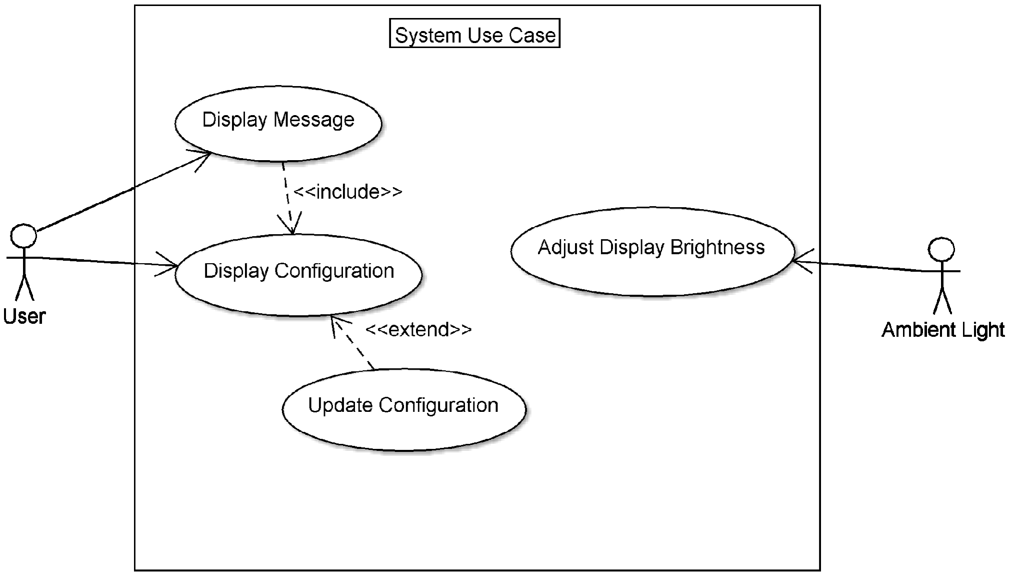


Figure 3.19: System Use Case Diagram (De Villiers, 2014)

Features that the user can change or have control over, were to display a customised message and how it would display this message. It, however, did not indicate how or on which medium it would be displayed.

Another feature to which the user had no control but was affected by external input was the adjustment of display brightness. Without this, the user would have to manually adjust the brightness of the display.

As seen in Figure 3.19 the display message functionality uses the display configuration. Display configuration in turn is an extension of update configuration. Making use of this means the system can remember the current configuration and will not require the user to input this every time a new message is entered.

By display configuration controlling update configuration the user interaction is limited while full functionality is retained.

Information not depicted in Figure 3.19 on the previous page is that the user-controllable functionality will be implemented on an ARM processor, while the brightness control will be implemented on a microcontroller. Making use of this implementation enables concurrent operation of the system.

3.2.3 Use Case Extended

The numbers in brackets in the following section refer to the elements of the figures. Figure 3.20 shows how the display message extends, which is an extended version of the system functionalities. This means all the functionality the system will have and how they are interlinked with each other. Since this is still a Use Case it will not show how each function works. It will rather show graphically how key components or functions will interact with each other.

Taking a look at (1) display message, it uses message (2) and display configuration (5). This conforms to Figure 3.19 on the previous page, but in more detail.

Message (2) in turn makes use of update message (3) and the last message used (4). The reason for this is for the system to remember what the user has entered and not prompt the user to enter a non-changing or constant message repeatedly. However, this still gives the user the option to change the message at any given stage.

Looking at the display configuration (5) branch of this Use Case diagram. This can be broken into two major functionalities of the system, update configuration (6) and last used configuration (7).

This allows the system to remember the user-confirmed configuration, but with the flexibility to change it later, should the need arise.

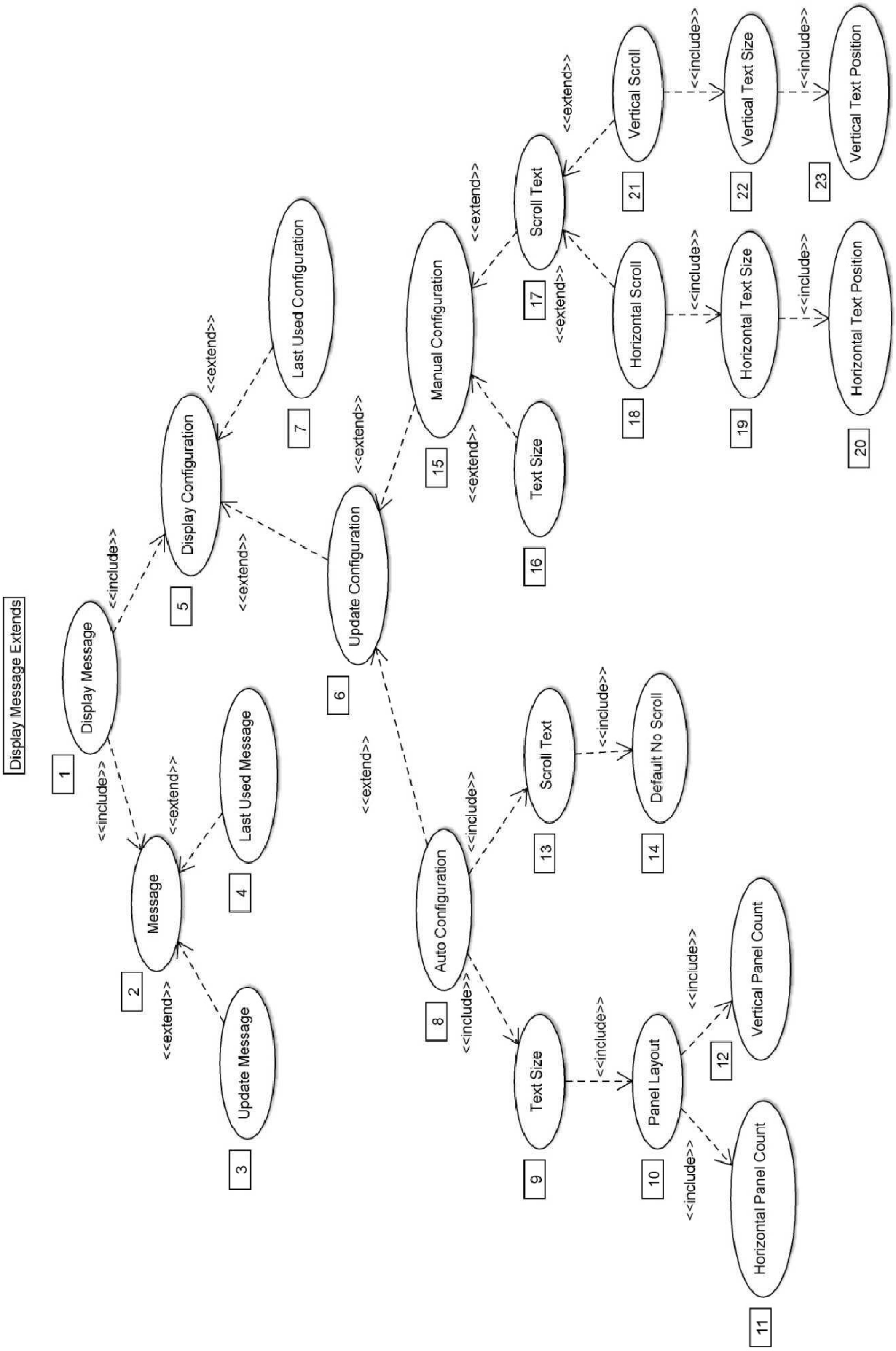


Figure 3.20: Use Case Extended (De Villiers, 2014)

Using the update configuration (6) path, this can be broken up into an auto configuration (8) and a manual configuration (15).

Auto configuration (8) will only be used on start-up. This is not indicated here as this is a Use Case diagram. However, auto configuration (8) is the father of text size (9) and scroll text (13).

Text size (9) is dependent on the panel layout (10). The bigger the display the larger the text size can be. The panel size is determined by detecting the number of horizontal panels (11) and vertical panels (12). This ratio of horizontal and vertical size then gives the basic area of the display as a whole. This is then used to determine the text size (9).

The second part of the auto configuration (8) branch sets the scrolling of text (13). This is set to a default value of no scroll (14). This sets the display to a type of static mode.

After the entire auto configuration (8) has been done, the user can manually re-configure the system (15). The user can change the text size (16) from small to large or from large to small. This text size will be used to determine the character sets to be used.

The user can also change the scroll text (17) parameters, making use of functionality for horizontal scroll (18) and vertical scroll (21).

Horizontal scroll (18) uses horizontal text size (19) and horizontal text position (20) to set whether or not the text should scroll and from which point in the display it will start. The exact same process is used for the vertical scroll (21), using the vertical text size (22) and vertical text position (23).

3.2.4 Display Message

The activity diagram of Figure 3.21 on page 47 is a broad overview of all the steps required to display a message on the LED dot matrix display.

On initial boot (1) or start-up of the system, it will immediately do a display auto configuration (2). On completion of the auto configuration (2) the system will prompt the user for a message (3) to display. The display configuration (4) will be displayed next. If the user does not wish to change the configuration (5) then the message will be displayed (12).

If, however, the user would like to change the message configuration (5), the user will be required to confirm the text size (6). If this is not the required text size (6), the user can change it by entering the desired text size (7).

Next it will prompt the user to verify the scroll details (8). If the scroll details need to be changed, the user can enter the desired scroll details (9). If the scroll details are correct, the system will update the display configuration (10).

Once again the user will have the opportunity to verify the message configuration (11). If the configuration data or requirements need to change, then the program will return and repeat from text size (6).

If the user confirms that the current configuration is as desired, then the message will be displayed (12).

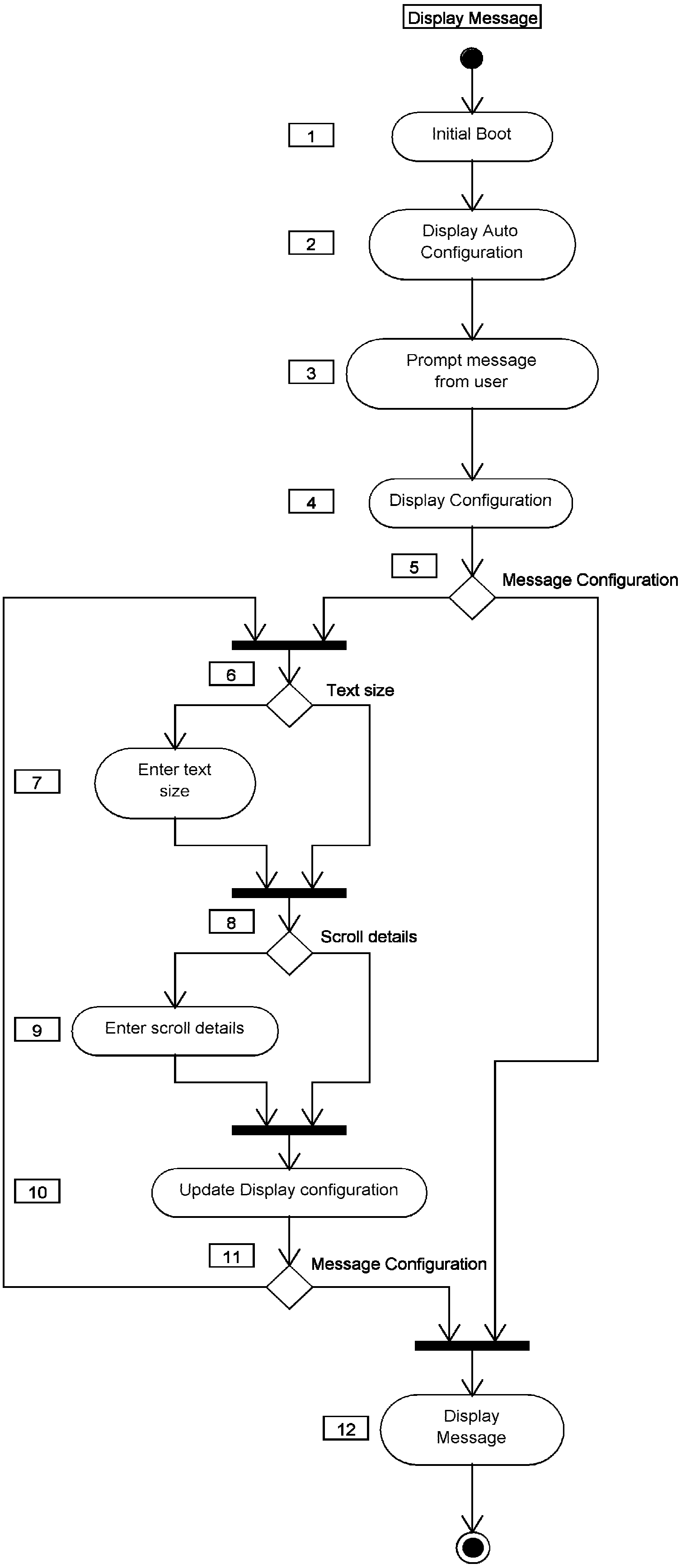


Figure 3.21: Display Message Activity Diagram (De Villiers, 2014)

3.2.5 Auto Configuration

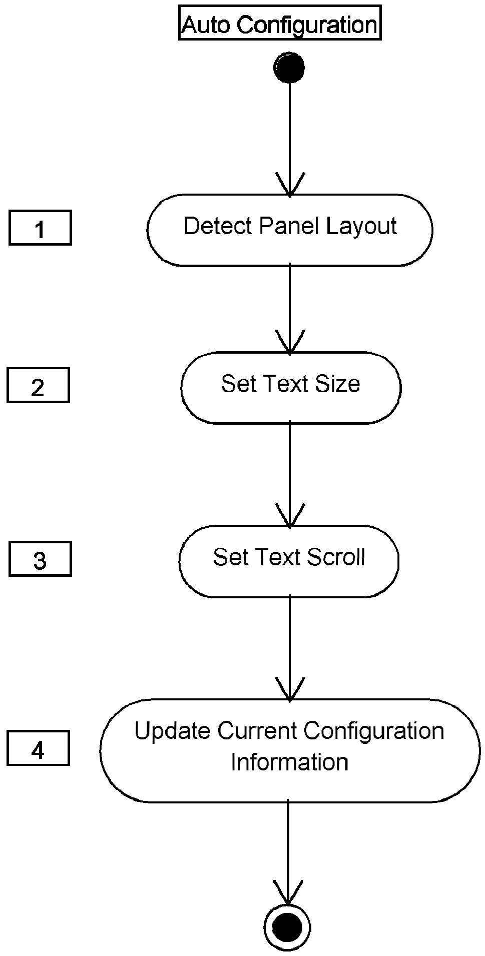


Figure 3.22: Auto Configuration Activity Diagram (De Villiers, 2014)

In the Figure 3.22 on page 48 auto configuration activity diagram, the core procedure for the auto configuration is depicted. This is only a high level view of how the auto configuration will function, indicating only key functions but not how they perform the actual tasks.

Initially the auto configuration will detect the panel layout (1). After the panel layout detection has been completed, it will then set the text size (2), set the text scroll (3) and up-date the current configuration information (4).

Each of these processes will be discussed in more detail in a future diagram. This is just to give the user an idea of how the auto configuration functionality is structured.

3.2.6 Panel Layout Detection

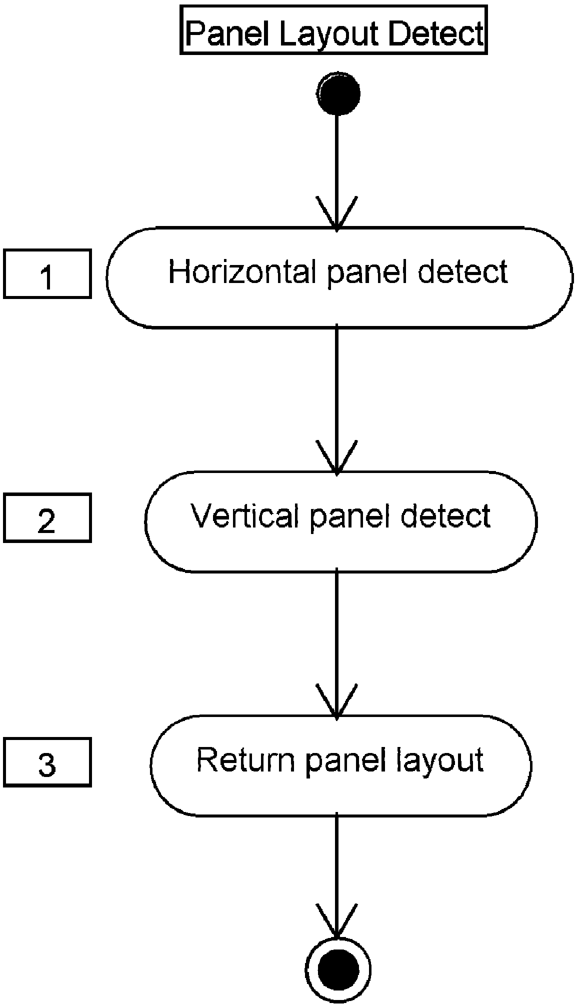


Figure 3.23: Panel Layout Detection Activity Diagram (De Villiers, 2014)

As seen in Figure 3.22 the auto configuration activity diagram has a detect panel layout (1) function. Here in Figure 3.23 panel layout detection activity diagram the functions for that high level view will be shown.

During the panel layout detection function, the system will do a horizontal panel detect (1) first and then a vertical panel detect (2). These can be swopped around, but for simplicity the design will be implemented exactly as it is here. After the panel detection has been done, the system will return to the panel layout (3). This value is then used by other functions which will be discussed in detail in coming sections.

3.2.7 Horizontal Panel Detect

The procedure used is depicted in Figure 3.24 on the next page the horizontal panel detect activity diagram. Shown here is the detailed version of the horizontal panel detect (1) as seen in Figure 3.23 on the previous page.

To start the horizontal panel detects a large number “n” of 0s (1) need to be pulsed out of the system. This is done to clear all the hardware registers which was discussed earlier in the hardware section of the design methodology. The number of 0s is currently restricted to a 255 in software for simplicity. This can be made available to the user to change at a later stage should the display be larger than a 255 panels horizontally.

After all the hardware registers have been cleared to 0’s a single pulse with a value of 1 is transmitted (2) after which it is followed by 0’s (3) again. Here a loop type of structure is followed. The loop starts after the single high pulse of 1 has occurred (2).

A horizontal counter (4) will be used to keep track of the number of repetitions of the loop as this will indicate the number of horizontal panels. After every pulse of 0 (3), the system will increment the horizontal panel count (4) by 1 and read the input pin value (5). The value of the pin will be evaluated (6) and if it is not equal to 1, the loop will repeat.

If it equals 1, the loop will stop and the number of horizontal panels will be returned (7) to the system. Making use of this method of panel detection is slow, but requires only 3 wires and no direct addressing, which enables the circuit construction to be done in any configuration the user requires.

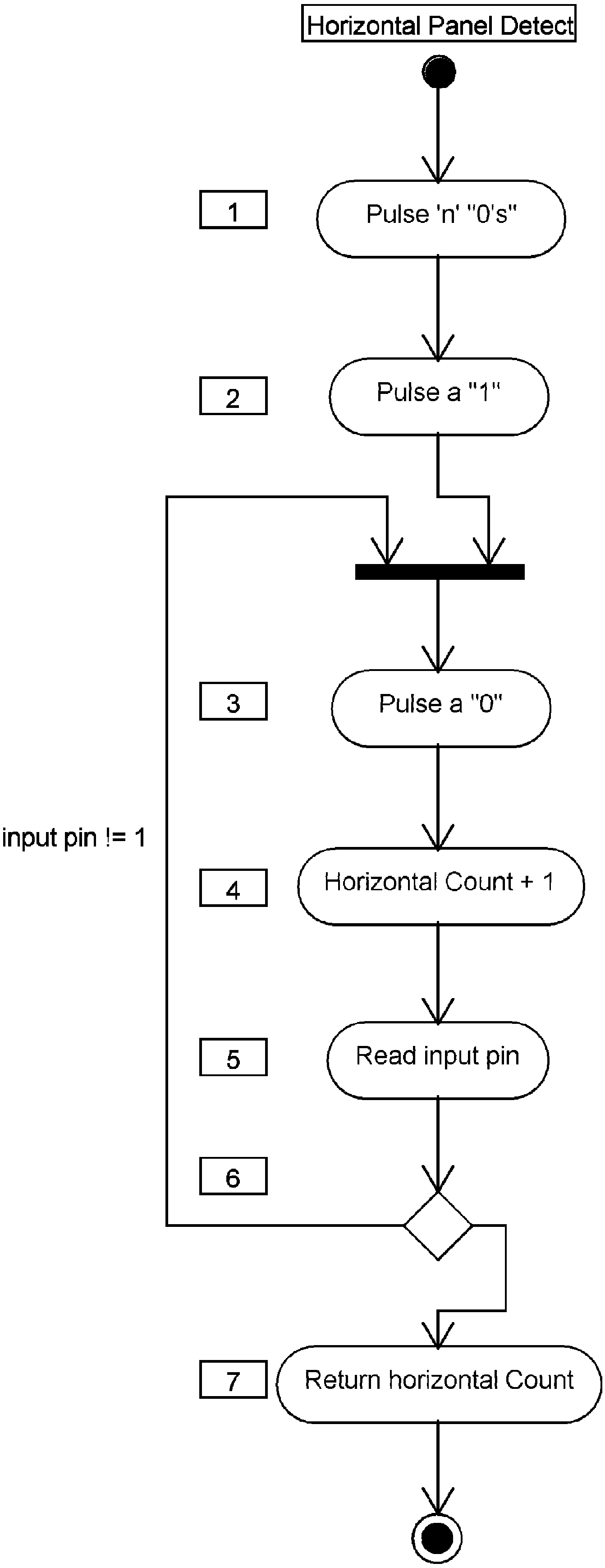


Figure 3.24: Horizontal Panel Detect Activity Diagram (De Villiers, 2014)

3.2.8 Vertical Panel Detect

The vertical panel detect activity diagram as seen in Figure 3.25 on page 53 shows how the system will complete this procedure. It is done in much the same way as the horizontal panel detection as seen in Figure 3.24 on page 51. However, the main difference is that it will do so for the vertical instead of the horizontal layout.

For simplicity the system will complete the horizontal panel layout first as depicted in Figure 3.23 on page 49. The system could detect the vertical panel layout first, but it was chosen on a design level to rather do the horizontal panel layout first. This does not mean that the system could not be reconfigured later to rather detect the vertical panel layout first instead of the horizontal panel layout.

As depicted this procedure will also clear the shift registers on the panels and a set number of 255 pulses of 0s (1) will be pulsed to do so. This fixed number of pulses can later be made available to the user to change as this can be included as an extra feature in future developments. It was however decided from a design and implementation point of view to simplify the implementation and hence only 255 pulses was chosen and is currently hard coded into the system.

After the 255 pulses of 0s a single pulse of 1 will be placed into the shift registers. This will be followed by a 0 pulse (3) and a vertical panel counter will be increased (4). Now the system will read the output of the last register of the panels connected to the system (5). If this input was read as a low or 0 (6), the system will pulse another 0 and increment the counter once again. This will be done until the system detects a high or a 1, after which it will return the vertical panel count value (7) to the system and proceed to the next stage of returning the entire panel layout (3) on a system wide scale as seen in Figure 3.23 on page 49.

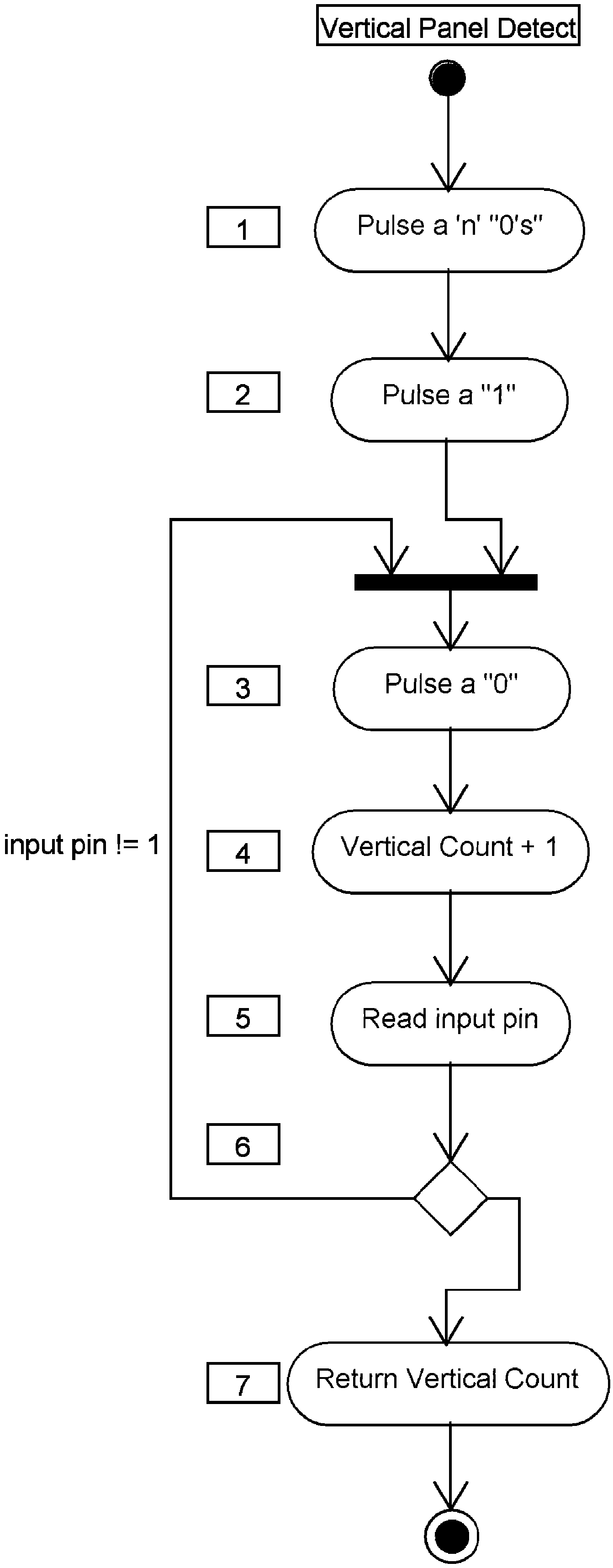


Figure 3.25: Vertical Panel Detect Activity Diagram (De Villiers, 2014)

3.2.9 Text Size

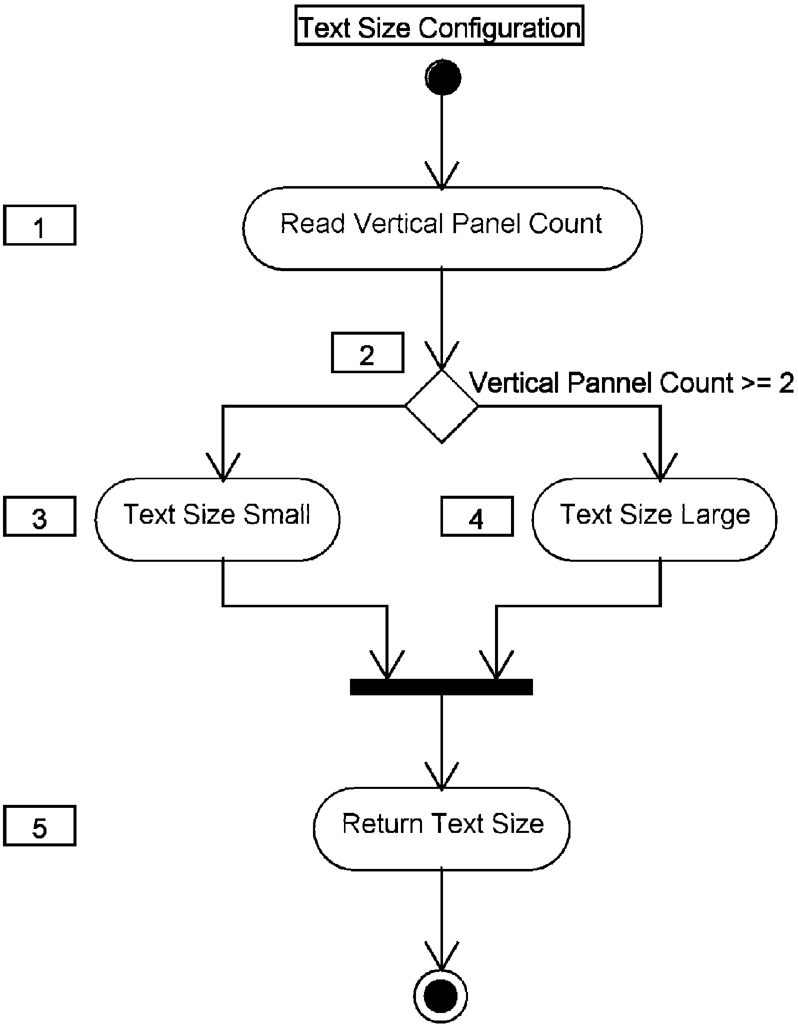


Figure 3.26: Text Size Configuration Activity Diagram (De Villiers, 2014)

In Figure 3.26 the text size configuration activity diagram is seen. This activity or procedure takes place after the panel layout detection activity or section of the program in the system has been completed. This is shown in Figure 3.22 on page 48.

For the text size to be automatically set by the system it only requires the vertical panel count (1). The reason for this is to determine what text size, and in particular, the text height that would fit into the allotted display area. Hence the check whether the vertical panel count is equal to or greater than 2 panels (2). If it is greater the automatic text size will be set to large (4), but if the vertical panel count is lower than 2, the small text size will be set (3).

It was done in this manner as to accommodate a large display layout which will require larger text to be displayed. Hence the bigger the display, the larger the text should be. As this is done only to demonstrate the functionality it was chosen to only have two text sizes. This will be increased in future distributions of this project.

After the text size has been chosen by the system this text size will be returned as a system wide value (5). This value was utilised later when text physically needed to be displayed and when the user wished to change the text size when updating the manual configuration as seen in Figure 3.21 on page 47.

3.2.10 Scroll Text Configuration

After the panel layout (1) and text size (2) has been set, the system will set the scroll properties of the text (3) as seen in Figure 3.22 on page 48. Here in Figure 3.27 on the next page is the scroll text activity diagram to depict how the user can manually change the configuration after the auto configuration was completed.

The system would firstly detect whether or not the auto configuration had been done (1). If this had not been done the system will set the horizontal scroll to false (10) as well as the vertical scroll to false (11). This is done by default and then returned system wide (12) and can later be changed by updating the system configuration as seen in Figure 3.19 on page 42.

However if the auto configuration was completed the system would rather display the current configuration data (2) even if the current configuration was still the same as the auto configuration data.

The user would then be requested to verify the current scroll configuration settings (3). If these settings were correct, then the scroll settings would be returned (12).

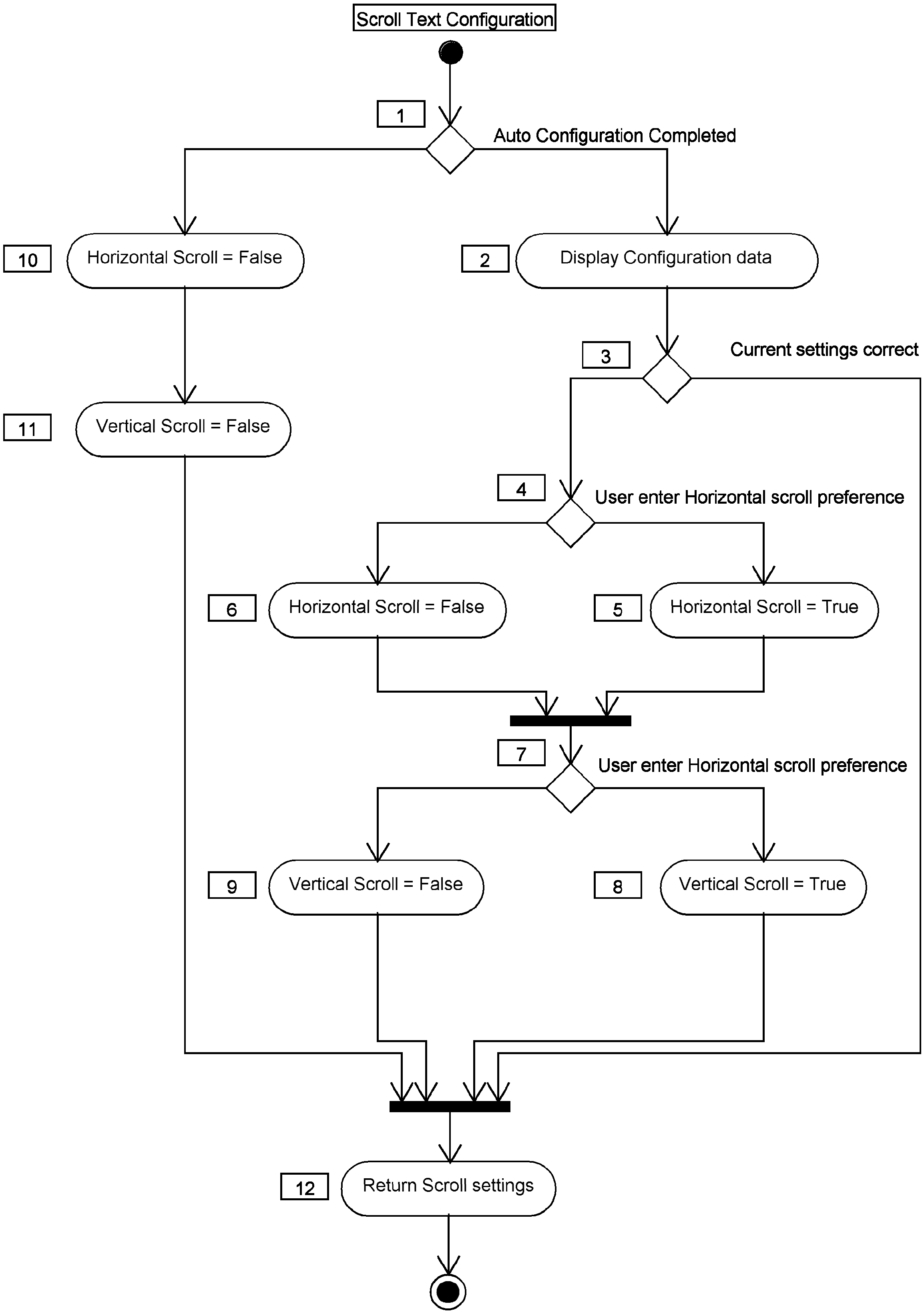

Scroll Text Configuration
1
Auto Configuration Completed
10
Horizontal Scroll = False
2
Display Configuration data
3
Current settings correct
11
Vertical Scroll = False
4
User enter Horizontal scroll preference
6
Horizontal Scroll = False
5
Horizontal Scroll = True
7
User enter Horizontal scroll preference
9
Vertical Scroll = False
8
Vertical Scroll = True
12
Return Scroll settings

Figure 3.27: Scroll Text Configuration Activity Diagram (De Villiers, 2014)

If they were not as per the user requirement, the user would be asked what the horizontal scroll preference is (4).

Here the user could decide between making horizontal scroll true (5), or false (6). It does not matter which as long as the user selects one of the two. From here the user would be requested to select a vertical scroll (7) preference.

Here the user could select between making the vertical scroll true (8) or false (9). These scroll configurations could be both set to true or false, or any other combination as per the user preference. After all these scroll settings have been set according to the user preference the settings will be returned (12).

3.2.11 Message Activity

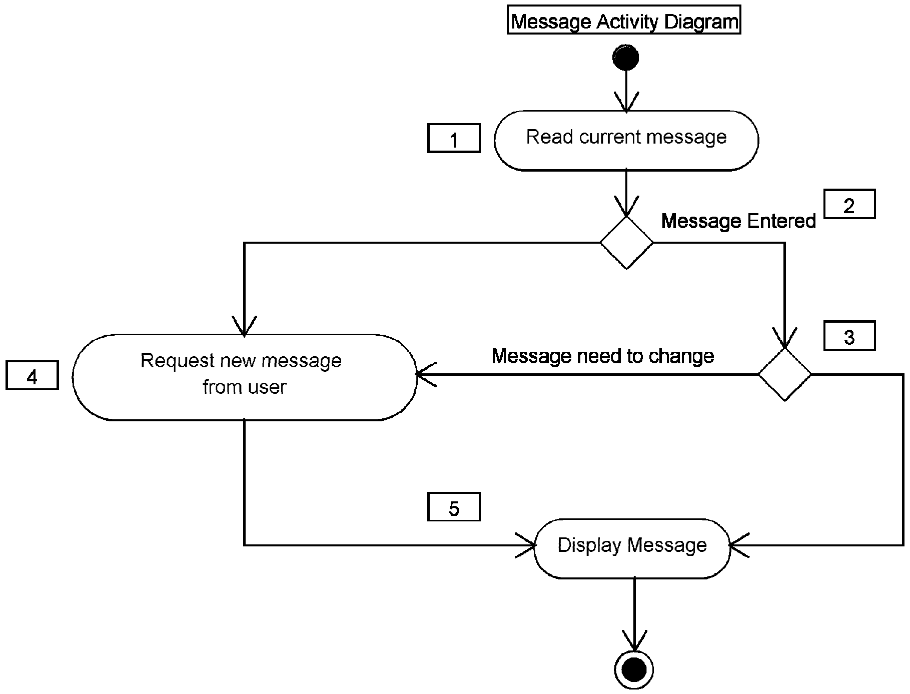


Figure 3.28: Message Activity Diagram (De Villiers, 2014)

During the initial boot or start of the system and after the auto configuration had been done the user would be prompted to enter a message (3) as seen in Figure 3.21 on page 47. In this activity diagram Figure 3.28 the process will be elaborated on.

When this process is first started it will attempt to read the current message that is in the system (1). However if there is no message entered (2), it will request a new message from the user (4). Once the message has been entered it will proceed to display message (5).

If there is a message in the system (2), it will ask the user if they wish to change the message (3). If so then once again the user will be requested to enter a new message (4) after which it will be displayed (5).

If the user would not like to change the message that is currently in the system (3), the system will then proceed to displaying the current user message (5).

3.2.12 Display Configuration

Figure 3.29 on the next page gives a broad overview of the display configuration process as it will be implemented in the system. At the initial start of this procedure the system will read the current configuration (1). If this configuration is not valid (2), the system will automatically do the automatic configuration (3) process.

If the configuration is valid (2) or the automatic configuration (3) has been completed successfully, the new or current configuration will be displayed (4) to the user. The user will then have the option of changing the configuration (5) or accepting it as is.

If the user would like to change the configuration information (5), then the system will request the user to enter new configuration information (6). This will then be displayed back to the user (4) and once again the user has to verify whether the configuration information is to his/her specification (5).

Once verified the system will update the current configuration data (7) and return to normal operation as will be seen in Figure 3.30 on page 61.

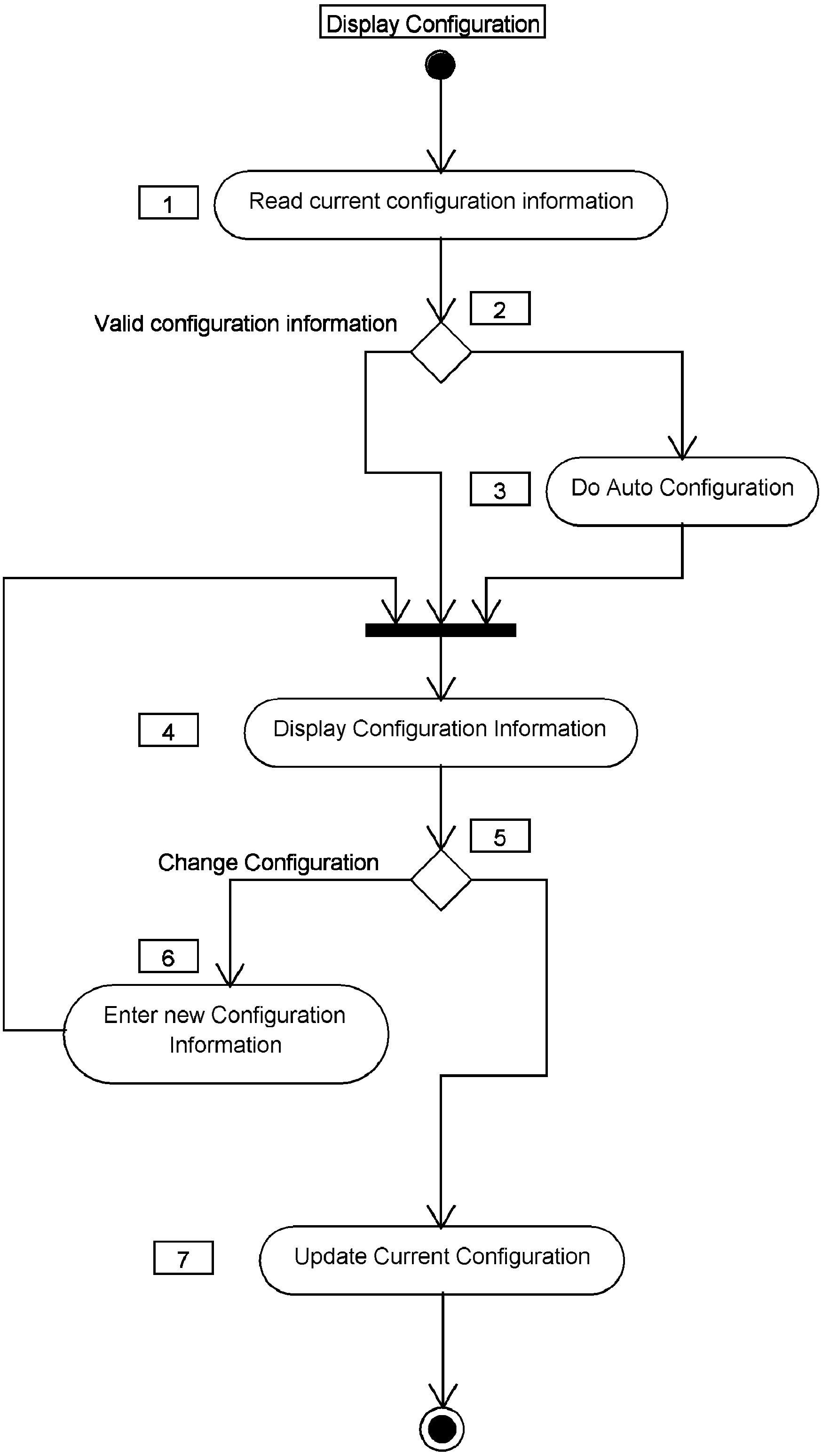

Display Configuration
1
Read current configuration information
Valid configuration information
2
3
Do Auto Configuration
4
Display Configuration Information
Change Configuration
5
6
Enter new Configuration Information
7
Update Current Configuration

Figure 3.29: Display Configuration Activity Diagram (De Villiers, 2014)

3.2.13 Normal Operation

Depicted in Figure 3.30 on the next page is the basic activity diagram of how the system will function during normal operation. After the initial boot (1) of the system, it will complete the auto configuration of the display (2) connected to the driving circuit. It will then prompt the user for a message (3).

This can be a custom message of any length containing numbers and letters. The display configuration will then need to be confirmed or verified (4), before the message can be displayed (5).

The system will continually check if the user has pressed a button on the keyboard (6). If not, the system will continue to display the current message with the current configuration.

If a button was pressed on the keyboard, the system will then prompt the user for a new message to display (3), after which it will once again ask the user to verify the configuration (4).

This new message (3) and configuration (4) will then be used to display the message (5) once again.

This process will continue until the power has been switched off.

A command line interface to control the display was implemented as this required the least amount of resources.

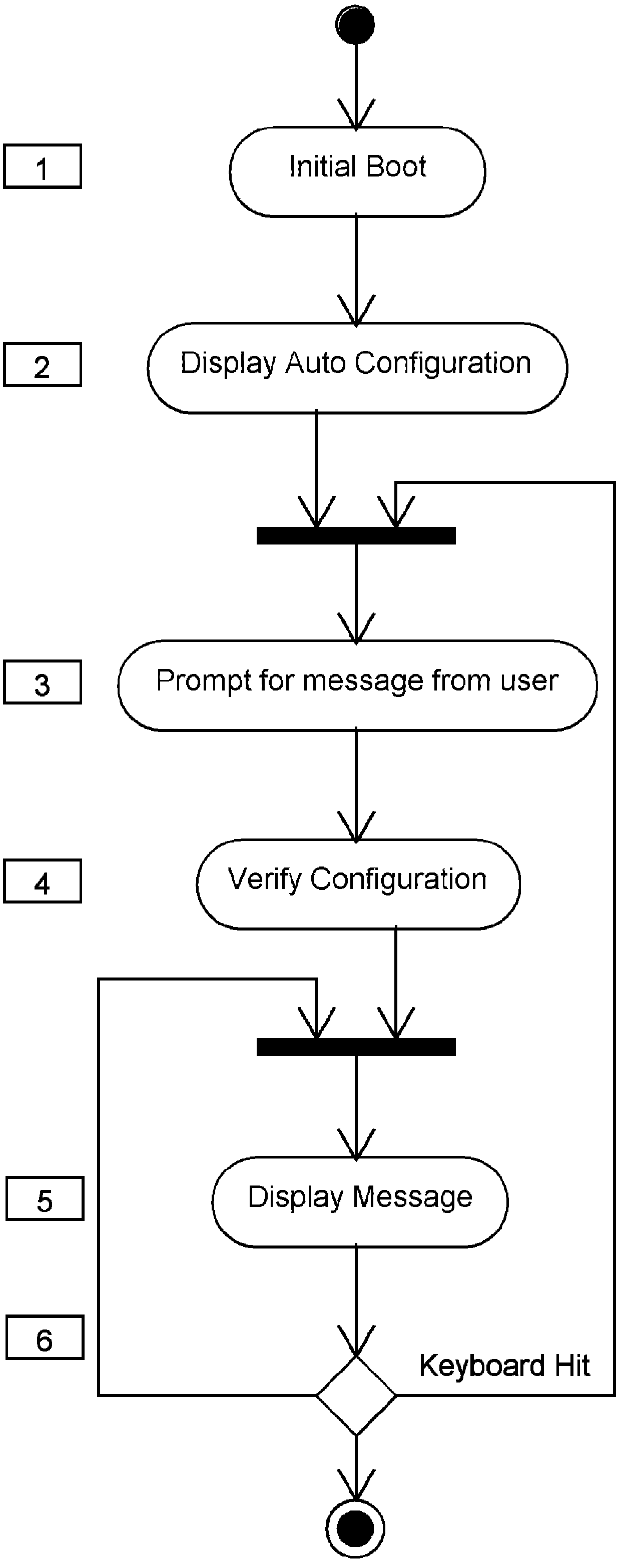


Figure 3.30: Normal Operation of the System (De Villiers, 2014)

3.2.14 Display Brightness Control

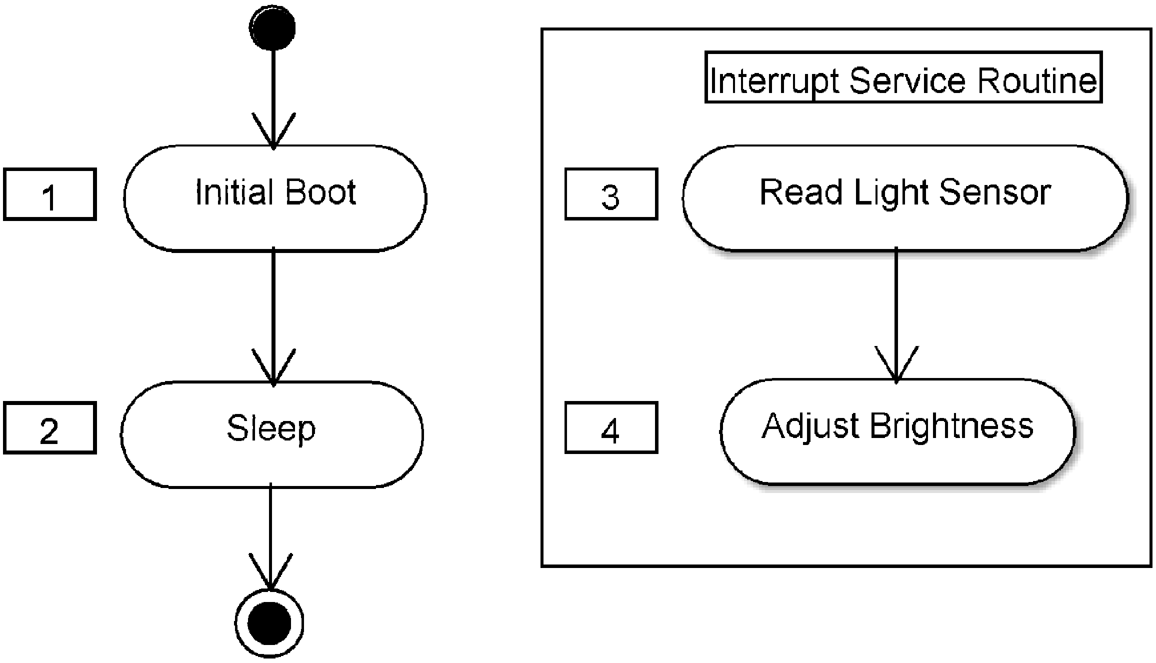


Figure 3.31: Display Brightness control (De Villiers, 2014)

Display brightness control is done with the onboard microcontroller, the PIC 16F818 (PIC16F818/819, 18/20-Pin Enhanced Flash Microcontrollers with nanoWatt Technology. 2004). This microcontroller will be placed in sleep mode (2) after initial boot (1) at power up. Once the Raspberry Pi has been connected and powered, a logic high will be placed on the interrupt pin of the microcontroller.

This will start the interrupt service routine as seen in Figure 3.31. Here the microcontroller will read the light sensor (3) value and adjust the display brightness (4) for all the connected LED dot matrix display panels. This will continue until either the Raspberry Pi is powered down or power removed from the display interface card.

4. Results

4.1 LED display panel

During the initial design of the system it was decided to make use of 8 by 8 LED dot matrix display units. These units would then be configured in a square to form 16 by 16 LED dot matrix panels. By doing this the display can be configured in any shape or size making use of square LED dot matrix panels.

If the LED dot matrix panels had been made to be rectangular, the system would still work. It would just not be as easy to reconfigure the display to be the desired size. Also by implementing it in this way there was no waste in electrical components that were underutilised in the implementation of the design. This is because most common shift registers are either 8-bit or 16-bit in architecture.

For the re-configurability and testing of the system it was decided to build four LED dot matrix display panels. These panels were then interconnected in the different layout configurations and performed as expected.

4.2 Initial LED display panel tests

After the initial designs of the LED dot matrix display panels, tests were done mainly for brightness as the brightest design configuration needed to be found. The four initial designs were the constant voltage, constant current, bi-polar and making use of an existing LED dot matrix driving IC.

This was done in a controlled environment with a Major Tech LUX meter, the MT943 (MT943. Lux Meter). This LUX meter was placed in an enclosure exactly a meter away from the LED dot matrix panels. The temperature of the room was also measured at 25℃. These are the set standards at wh ich LED’s luminescence are measured according to the datasheet of the TC23-11SYKWA LED dot matrix display modules used (TC23-11SYKWA, 2011).

Each of the panels was then flooded with a 50% duty cycle at a refresh frequency of 800Hz. Each of the LED dot matrix display panel topologies received the same input data. By doing this the output of each LED dot matrix display panel could be evaluated objectively.

Of the four initial designs the only one that did not perform at all was the existing driving IC. The AS1116 (AS1116, 2010). Correspondence with the technical staff of the manufacturer led to the conclusion that the way the IC was implemented was correct, but that their IC just did not perform according to specification (Koschier, M. 2012).

The LED dot matrix display that was the least bright of the designs was that of the constant voltage display. Even with the display at maximum usable duty cycle it did not perform nearly as well as the remaining two designs with only 8.3 LUX measured.

Measuring the bi-polar design configuration showed a dramatic improvement over the constant voltage design. Varying the current being supplied to the LED dot matrix panel enabled this design to be the second brightest, measuring a maximum of 43 LUX.

Placing the constant current design in the test rig indicated that this was the brightest design with 45 LUX measured. Since this design was the brightest, it was chosen to be the backbone of future designs in this project.

4.3 Display card

The initial display card contained all the processing and interfacing with the LED dot matrix display panels. Making use of the PIC16F690, an 8-bit microcontroller, the LED dot matrix panels could show static messages but these were very limited (PIC16F818/819, 18/20-Pin Enhanced Flash Microcontrollers with nanoWatt Technology. 2004). This was due to a lack in on-board RAM and ROM.

The next display card design incorporated a more powerful microcontroller, the dsPIC33FJ128MC802 (dsPIC33FJ32MC302/304,dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04, High-Performance, 16-bit Digital Signal Controllers. 2011). Making use of this microprocessor allowed for the display to be more versatile in what it could display and then how it would be displayed. However, since this project might be used on a larger scale, it was decided to investigate more powerful processors.

This led to the use of FPGA's as these devices in conjunction with a microprocessor could run larger LED dot matrix displays with more complex graphics. It soon became apparent that the cost of this solution would be a deciding factor. Thus finding a cheaper yet equivalent solution was required.

Hence the Raspberry Pi computer was chosen (Raspberry Pi. 2013). This computer is based on an ARM processor architecture with a Linux platform. Making use of this platform reduced the development time and doubled as a user interface structure. Since the Raspberry Pi computer is a pre-built computer, it was decided to make use of the on-board IO pins to interface with the LED dot matrix display panels.

The display card was thus broken into two parts. The Raspberry Pi computer and an interface board. The display interface board has no intelligence with regards to determining what should be displayed on the LED dot matrix display panels. It does have intelligence for controlling the display brightness. This was done making use of an opto-transistor the TEMT1000 (TEMT1000, 2008). The display brightness microcontroller remained in sleep and only awoke once the presence of the Raspberry Pi computer was detected (Raspberry Pi. 2013).

The opto-transistor, connected in series with a pre-set potentiometer and in parallel with a capacitor, worked very well. Refer to Figure 3.11 on page 30. This allowed for easy calibration of the sensitivity of the opto-transistor. Making use of the capacitor allowed for the LED dot matrix display to reject quick or sudden changes in ambient light. This made the display more appealing to the viewer as the brightness would slowly adjust to suit the environment it was placed in.

Making use of this concurrent computing structure allowed for smooth changes in the display brightness as well as the smooth display of the required graphics.

4.4 LED dot matrix display panel detection

Since one of the requirements was that the LED dot matrix display panels should not have any intelligence, the challenge was to find a way in which the system could detect the physical display configuration.

Initial design thoughts were to include some form of static memory on each panel. This could work but would force the installer to build the display in specific configurations, which would be in direct violation of the display configuration requirement.

The next design that was tested was making use of PCB mount slide switches. This seemed to be more versatile but would only work on a small scale. Up to 16 panels if 4 way slide switches were used. Hence a more static type of solution had to be found. One that does not require hard coding or static memory.

Thus making use of D-latches the display configuration could be determined. This was a slow method of determining the display configuration, but seemed to work very well. As soon as the LED dot matrix display panels were connected in a different configuration, the software on initialisation to drive the display detected the change.

4.5 LED dot matrix display panel

The LED dot matrix display panel design took shape and as the design had to operate from 5V, all electronic components associated was chosen to meet this requirement. Since these LED dot matrix display panels may not contain any on-board intelligence, they were designed making use of constant current shift registers to aid in the displaying of the graphics. Digital pots were used to adjust the current that the constant current shift registers would sink. This in turn adjusted the display brightness. Also to assist in the controlling of the display graphics normal shift registers were used.

Tests were done and the minimum data feed frequency of 12.8kHz was used to display the graphics. This allowed for optimum brightness in the current configuration while not exceeding the maximum peak forward current of 2.72A.

4.6 Software development

Software development was done on all the tested hardware, but only the final version was completely tested. As discussed here earlier, while deploying the software on the selected hardware, it was found that the hardware could not adhere to all the requirements of this project.

All software development was done making use of incremental and iterative development techniques (Dr. Alistair Cockburn. (May 2008) Using Both Incremental and Iterative Development). Making use of this technique decreased the development time in each hardware environment. This also helped in deciding which hardware to implement the system on.

The final version of the software performed as expected without over taxing the ARM processor. This means that the hardware will not require any artificial cooling to run the software outdoors for prolonged periods of time.

Chapter 5 - Discussion of results

5.1 Discussion of initial results

5.1.1 Existing driving IC

During the testing of the LED dot matrix display panel configurations, it was found that one of the configurations did not work at all. This was the existing IC configuration making use of the AS1116 (AS1116, 64 LED Driver with Detailed Error Detection. 2010). Correspondence was done with the technical staff at Austria Microsystems.

The test code for making use of this AS1116 IC was sent to Austria Microsystems. They recommended that the scan limit be verified. It seemed that the code was written in such a manner that the data was being clocked into the device in the wrong register. After the code had been updated, the system was tested again. Yet, no data was shown on the LED dot matrix display.

Tests were then done in the SDI and SDO pins of the device and found that the data being inserted into the device was not the same as the data exiting the device 16 clock cycles later. This can be seen in Figure 5.1.

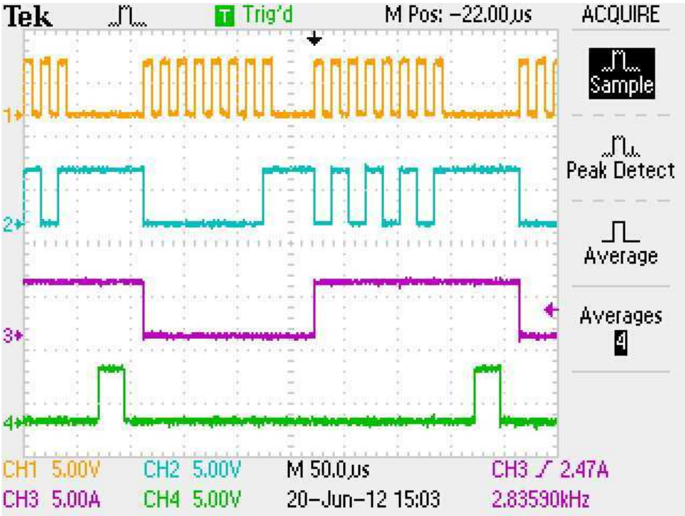


Figure 5.1: Initial waveforms during testing

In Figure 5.1 on the previous page, channel 1 is the clock pulse with channel 2 being the input data to the SDI pin. Channel 3 is the output of the SDO pin and channel 4 is the enable data load pin. Data was inserted into the registers when the enable data load pin transitioned from low to high (AS1116, 64 LED Driver with Detailed Error Detection. 2010).

Channel 3 should be an exact replica of channel 2, just 16 clock cycles later. It was however, not the case. This led to Austria Microsystems stating that the start-up procedure needed to be revised. Updating this allowed for the SDI data to be replicated on the SDO pin 16 clock cycles later.

Since still no LED's on the display were switching on, the technician at Austria Microsystems suggested that the device be tested making use of the exit from shut down command. The resulting waveforms can be seen in Figure 5.2.

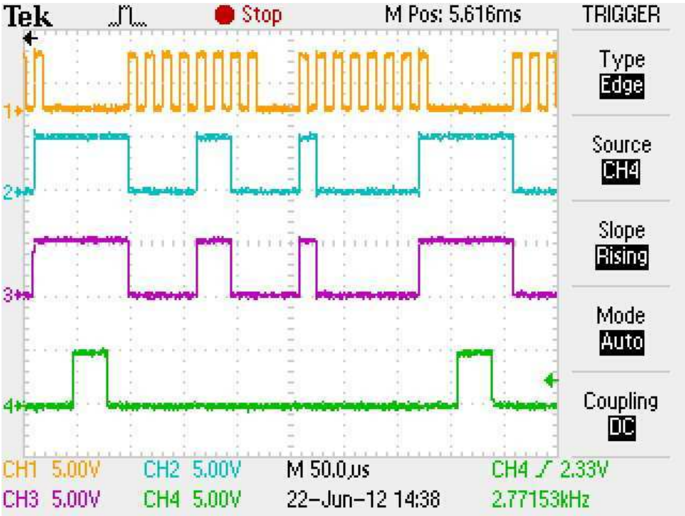


Figure 5.2: Exit from shut down command

It was seen that the SDO and SDI waveforms were the same when referring to channel 3 and 2 respectively. It was clear that the device did wake from shut down mode, yet still none of the connected LEDs switched on when the display code for flooding the display was applied.

After careful bit bashing, it was noted that the SPI data stream from the microcontroller was in fact mirrored. Thus the SPI data stream driver was updated and the resulting waveforms can be seen in Figure 5.3.

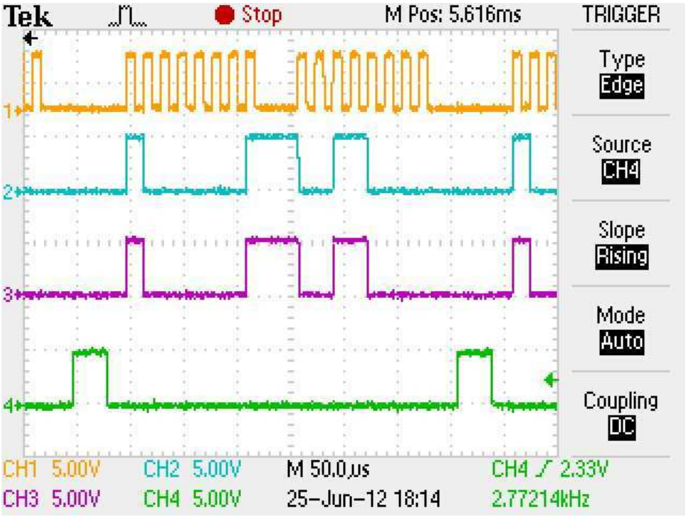


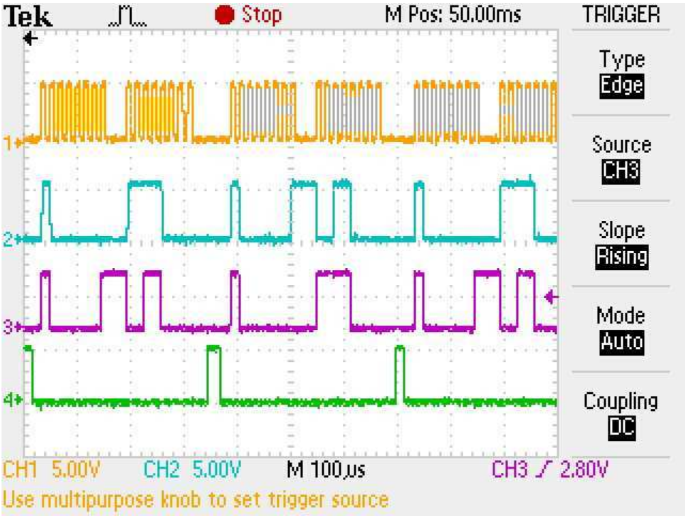
Figure 5.3: Reversing the SPI data stream

Observing the waveforms in channel 2 and 3 in Figure 5.3 above and Figure 5.2 on page 69 shows that the SPI data stream were successfully reversed. This reversal in the SPI data stream was continued for displaying data on the LED dot matrix display driver. This did not enable the display at all and still no data was present.

Tests were done to ensure that the schematic design was followed correctly by removing the LED dot matrix units from the display panels. The schematic can be seen in Appendix A on page 86. Measuring the potential from the LED enable pins of the AS1116 to ground was 346mV. This is not nearly enough to enable most LEDs and it was thus deduced that the AS1116 did not switch on the output pins. A single LED was also placed in the circuit after removing the LED dot matrix units. These did not switch on. As a last resort these LEDs were deliberately placed in reverse biased mode to determine if the schematic to PCB transfer did not happen correctly. The LEDs did not light up at all.

The technician suggested that the wires carrying the data from the microcontroller might have some noise on them and that separating the wires or shielding them from one another, might be a consideration. This was done as well as high frequency capacitors placed on the SPI data lines as close as possible to the AS1116 IC.

As a final test request from Austria Microsystems the technician requested the device be woken from shut down and the display test be initialised. This was done and the resulting waveform is seen in Figure 5.4.



M Pos: 50.00ms

TRIGGER

Type Edge

Source CH3

Slope Rising

Mode Auto

Coupling DC

CH1 5.00V CH2 5.00V M 100µs CH3 ∕ 2.80V

Use multipurpose knob to set trigger source

Figure 5.4: Final test request from Austria Microsystems

Looking at channel 2 and 3 it is evident that the input signal to the AS1116 IC is the same as the output signal, just 16 clock cycles later. The enable load data pin also rose after each data entry had been done as can be seen on channel 4.

After all these tests had been done and results send to Austria Microsystems, the LED dot matrix display panel still did not work. Hence it was considered that the IC was either of a defective production run or that the IC did in fact not work at all (Koschier, 2012).

5.1.2 Constant Voltage design

This design was done to illustrate that LEDs are current devices and not voltage devices. Also the design was done in a very conservative manner. The reason for this was that once the LED dot matrix display panel was active and any of the software written malfunctioned, then the LED dot matrix units would not be destroyed. Hence the continuous current that was allowed to pass through the LEDs were 30 mA. This value was chosen according to the manufacturer's specifications stipulated in the datasheet (AS1116, 64 LED Driver with Detailed Error Detection. 2010).

To adjust the brightness in this specific LED display configuration, either the potential or duty cycle need to be adjusted. Frequency can also be used, but to allow the display to be brighter, the frequency needs to be lowered. This in turn would make the display flicker to the viewer due to the flicker threshold (Flicker fusion threshold Summary | BookRags.com. 2011). Hence this method was not used and a minimum of 12.8 kHz was used. Making use of this frequency each LED would switch on and off at 50Hz which the human eye cannot detect.

The design performed well with 480mA as its running current. However, it lacked the brightness required to drive the LED dot matrix display for outdoor use. It will work well in a workplace or office environment where the ambient light is not as bright.

This design was hence not chosen for future development.

5.1.3 Bi-polar design configuration

The bi-polar design showed a dramatic improvement over the constant voltage design with a brightness of 43 LUX. The backbone for this design is the next design that will be discussed, the constant current design. The primary reason for the implementation of this design is to improve the refresh rate.

This is done by first forward biasing the required LEDs, and then reverse biasing them to switch them off. When the power is removed from LEDs they switch off at a fast rate, but by reverse biasing them, they can switch off much faster. This technique is generally used in fibre optic systems to increase the data transfer rate (DAWSON R.D. (1980)). This did increase the refresh rate and allowed the system to be used at a lower frequency of 7.68kHz.

Even though the frequency at which this configuration is lower and the refresh rate better, it was still not the brightest of the designs tested. Also for an increase in refresh rate the cost in components did not warrant this design to be used in future versions of this system.

5.1.4 Constant current design

This design performed the best of all the designs tested with brightness at 45 LUX. Also since this design is based on constant current and LEDs are current devices, this result was expected. Also, since the LEDs are allowed to switch off at their own pace after power has been removed, there is a short lingering period in which residual light is still being emitted.

This light is not present in the bi-polar design and most probably why the constant current design performed better in terms of brightness. This is also the reason why the refresh rate could not be lowered as this would allow for ghost images to form on the LED dot matrix display. By lowering the data feed frequency there will be flickering on the display. This would make for a poor viewing experience.

Brightness control is done by varying a single resistor on the A6276 R_{EXT} pin (A6276 16-Bit Serial Input, Constant-Current Latched LED Driver. 2011). A digital potentiometer was chosen for the final design. Initially potentiometers that required user adjustment was used in the design for simplicity.

The constant current design was thus chosen for its simplicity and for being the best performer in terms of brightness.

5.2 Discussion of results

5.2.1 Display card development

During the initial stages of the project the LED dot matrix driving topologies were tested by making use of 8-bit microcontrollers. This worked well and the displays did show graphics. Unfortunately the graphics could not be animated and more complex graphics could not be attained. The limitation lay not in the software, but rather in the RAM and ROM of the microcontroller of 4kb and 256bytes respectively. Externally connected memory could aid in solving the ROM problem, but the RAM requirements could not be met as easily.

Thus a more powerful microcontroller with a 16-bit architecture was chosen. The dsPIC33FJ128MC802 (dsPIC33FJ32MC302/304,dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04, High-Performance, 16-bit Digital Signal Controllers. 2011). This microcontroller had 128kb ROM and 16384kb RAM. Since this device had more programmable memory, the code for all the graphics could be placed within the microcontroller and did not require external memory to be added. With the increased RAM the microcontroller could allow for the graphics to be more complex and animated. Also with this microcontroller more than just one LED dot matrix display panel could be operated at the same time. However, since this project was scheduled to be made larger in future developments, it was decided that this was not the proper route to follow.

FPGAs with internal microcontrollers were investigated. The reason for using an internal microcontroller is for efficiency. Low end FPGAs normally do not have a very strong mathematical function and cannot store large numbers without sacrificing some of the IOs macro cells. Hence an internal microcontroller with a mathematical function is needed. Microcontrollers are primarily used in these FPGA microcontroller configurations to compute and store data. The unfortunate disadvantage of this was the cost involved. This was therefore not chosen but rather kept as a possible future development should the project become commercially viable.

The Raspberry Pi was chosen as this device is built on an ARM processor architecture. The ARM processor on the Raspberry Pi has a clock speed of 800MHz. This is more than fast enough to ensure smooth graphics on the LED dot matrix display. Further the Raspberry Pi computer model B has 256Mb RAM. These are major improvements of the initial microcontrollers chosen. Also the ROM on this computer is limited to the size of the SD card inserted into the device (SanDisk, SanDisk SD Card. 2004). Since all the code and the operating system are stored on the SD card, the entire project will not be susceptible to vibration faults as all components are static. The Linux operating system on which the algorithm was implemented also doubled as a user interface. This reduced the development time substantially.

Due to making use of the Raspberry Pi computer an external display interface card had to be designed. This would have been incorporated into the display card if a microcontroller or FPGA was used.

This display interface card had to be designed in such a way as to connect the 3.3V Raspberry Pi to the 5V led dot matrix display. This interface also had to connect the SPI port pins to the respective displays via RJ-45 connectors (RJ-45. EIGHT CONTACT, TEN POSISTION, VERTICAL MOUNT, UNSHIELDED JACK, 2003). Initially interface level shifting ICs were investigated, but since only two of the pins were required to input data into the Raspberry Pi from the LED dot matrix display and not the remaining interfacing pins, a simpler solution was sought. Also these input pins from the LED dot matrix display electronics can be operated at a low frequency and therefore a voltage divider network was decided on.

Voltage divider networks work well in low frequency applications, but at high frequencies unwanted harmonics can lead to false data being generated. Referring to Figure 3.13 on page 33 it can be seen how this interface was implemented. The other pins from the Raspberry Pi are all outputs and did not require any level shifting. The LED dot matrix driving electronics can all work from 2.75V to register a high and the Raspberry PI delivering 3.3V as a high. Therefore no interfacing electronics had to be developed.

5.2.2 Display brightness control

Display brightness was implemented as a secondary part of the display card or interface card. This was done with an 8-bit microcontroller the PIC16F819, instead of using the Raspberry Pi (Raspberry Pi. 2013; PIC16F818/819, 18/20-Pin Enhanced Flash Microcontrollers with nanoWatt Technology). Making use of this implementation reduces the need for the Raspberry Pi to control the brightness as well. Thus smooth brightness control can be attained without compromising on display graphics being displayed.

Testing the display brightness in this configuration allowed for smooth transitions from dim to bright and vice versa. This was done in a visual manner and it was evident that the display brightness did not jump or seem jagged while transitioning.

5.2.3 LED dot matrix panel detection

An initial design consideration was to place an addressing IC or hardware addressing on to the LED dot matrix display panels. This was, however, not implemented as these either required some form of intelligence on the LED dot matrix display panels or required large numbers of data lines.

Thus a simpler hardware solution was sought. Implementing D-latches enables the system to scan the connected LED dot matrix display to detect the number of panels connected. This was done in a horizontal and a vertical manner.

Even though this is a slow method of detecting the number of LED dot matrix display panels, it performed reliably. Calculating the time it would take to determine the number of panels in a display with 255 horizontal and 255 vertical panels equated to 1.275ms at 400 kHz.

This detection is only done once at power up of the system and hence considered to be sufficient and will be used in future versions of the LED dot matrix display.

5.2.4 LED dot matrix display panel

The design of the LED dot matrix panels was based on the constant current design as indicated earlier. This design performed exactly the same in terms of brightness as in the earlier tests. The only change to the LED dot matrix display electronics was the addition of the D-latches.

Adding the D-latches did require the panels to be interconnected, not with wires but more directly. Hence SIL connectors where placed off centre in the PCB design. This can be seen in Figure 3.14 on page 36. Making use of this off centre configuration ensured that when the LED dot matrix panels are connected together that the orientations of the panels remain the same. Thus no need for accelerometers on the LED display panels as the display can only be configured in a specific manner. This seemed to work well and reduced the assembly time of interconnecting the individual LED dot matrix display panels.

5.2.5 Software development

During the testing of the software required to operate the LED dot matrix display, it was found that the initial hardware did not have the required resources. Different hardware was therefore investigated. The algorithm was also revised to be less resource intensive.

This allowed the final hardware, the Raspberry Pi, to operate in an idle state. With the original algorithm the Raspberry Pi ARM processor operated at 100% capacity. Running the Raspberry Pi processor at 100% capacity made the LED dot matrix display flicker as the user entered a new message and configuration for the message to be displayed next. This was deemed not acceptable and hence reworked.

With the Raspberry Pi ARM processor in an idle state the display did not flicker as the user entered a new message with the configuration for the message to be displayed. As for the core SPI algorithm to change normal IO ports into SPI drivers these operated with success. All the LED dot matrix display panels updated at the same time as the IO pins for which the SPI driver was compiled received the data streams at the same time.

Thus the revised algorithm was a major improvement on the original algorithm and hence implemented. Future upgrades for the software were also considered and the code was structured in such a manner that the upgrades could be implemented, should the need arise.

6. Conclusions and Recommendations

6.1 System hardware overview

The design evolved from the initial system hardware design of only using a display card and the LED dot matrix display panel as seen in Figure 3.1 on page 17. Implementing an iterative and incremental design, it was found that the initial hardware model would not be sufficient (Dr. Alistair Cockburn. (May 2008)). Upgrades in the hardware were done using the same architecture, and investigating more powerful microcontrollers and FPGAs. It was clear that the hardware architecture had to be changed.

This hardware architecture changed to one where the Raspberry Pi is used (Raspberry Pi. 2013). Also included in the new hardware design was the need for a display interface card as seen in Figure 3.17 on page 39. This allowed the Raspberry Pi computer to interact with the LED dot matrix display panels. This interaction was then used to build a modular LED dot matrix display. One in which every module could be individually upgraded or replaced if required.

6.2 LED dot matrix display panel development

It was decided that four LED dot matrix driving topologies would be investigated and the design that performed the best with reference to brightness would be used as a platform for further development. The four driving topologies evaluated were the constant voltage, constant current, bi-polar and a method where by an existing driving IC was used.

Each of the designs was built and placed in a test rig with a LUX meter placed exactly a metre away. The temperature of the room was also controlled and constant at 25℃. The data update frequency for all the desi gns was set at 800Hz. The LED dot matrix display panels were then flooded to evaluate the brightness of each design.

Making use of the exiting driving IC, the AS1116 delivered an 0LUX reading as it did not switch on at all (AS1116, 64 LED Driver with Detailed Error Detection. 2010). After lengthy correspondence with the manufacturer, the AS1116 was deemed to be faulty (Koschier, M. 2012).

From the driving topology of the designs that functioned, the constant voltage was the design with the lowest LUX reading of 8.3LUX. The bi-polar design showed a major improvement over the constant voltage design in brightness. It measured 43LUX. The design configuration that performed the best with regards to brightness measured 45 LUX. This design was the constant current design. It was thus decided to use this design for further development.

6.3 Display panel detection

The detection of the LED dot matrix display layout could have been done in a number of ways. The initial design thought was to include a form of static memory onto the display panels. One of the requirements of the system was to have no self-driving intelligence on the panels, thus this was not implemented.

The second design that was tested made use of slide switches to give physical addresses to each panel. On a small scale this would work well, but should the display be used commercially and the size exceed the limitation of the slide switches, it would not work.

A solution that did not have any internal intelligence or static memory had to be found. D-latches were implemented. Connecting the D-latches in a daisy chain configuration between the LED dot matrix display panels ensured that the display size and layout could be found. This was done at the start-up of the system and determined the behaviour of the display.

6.4 Display brightness control

Brightness control was implemented on the display interface card. A secondary microcontroller here detected the presence of the Raspberry Pi computer. Once the

Raspberry Pi computer was connected to the PIC16F819, it measured the ambient light. This measurement was then used to adjust the LED dot matrix display panel brightness.

The TEMT1000 opto-transistor in conjunction with the calibration potentiometer and capacitor ensured for smooth transitions in display brightness. It also ensured no sudden change in the display brightness, due to the introduction of hysteresis.

6.5 Display graphics algorithm

The traditional method for updating LED dot matrix display graphics makes use of a serial data feed method. Making use of this limits the display size, due to the fact that the viewer can detect the updating of the display. LED dot matrix display brightness is also compromised by the increase in display size.

Thus a new algorithm was designed to allow for concurrent updating of the LED dot matrix display. This was done by breaking the entire display into panels. These panels were still serially updated but each of the panels receives the required display of graphics at the exact same time.

The limitation of the display size was not due to the brightness or the refresh rate, but rather the size of the controller and power supply. The user inputted the required graphics to be displayed. The algorithm then calculated and distributed the graphics to each individual panel. This was done in parallel.

6.6 Future developments

Communication between the display interface card and the LED dot matrix panels could be improved by making use of fibre optics. This will ensure the display can be made larger than a conventional display making use of twisted pair cabling.

Sensing or diagnostic electronics could be added to the LED dot matrix panels. This can then return panel information and report electrical faults. The diagnostic information would be very useful should the LED dot matrix display become large.

Interfaces with a real time clock and external timing triggers could also be added to the display interface card. This will allow the display to be used at sporting events where accurate time information is required.

References

2532 Series 0.100” Centre Shrouded Header. 2012, *HO CHIEN ELECTRONIC GROUP, INC*

50 year history of the LED – 9/22/2010 – Electronics Weekly. 2011. *50 year history of the LED – 9/22/2010 – Electronics Weekly.* [ONLINE] Available at: http://www.electronicsweekly.com/Articles/2011/02/18/49501/50-year-history-of-the-LED.htm. [Accessed 12 May 2011].

A6276 16-Bit Serial Input, Constant-Current Latched LED Driver. 2011. *Allegro MicroSystems, Inc.*

Abd El-Shafy Nafeh, A. (2010) Microcontroller-Based Moving Display Powered by Photovoltaic Energy. In: Fahmy, F.H. Sadek, S.M. Ahamed, N.M. Zahran, M.B. ,ed. *International Conference on Renewable Energies and Power Quality***.** Granada (Spain), 23rd to 25th March 2010

AS1116, 64 LED Driver with Detailed Error Detection. 2010. *Austria Microsystems*

Baldengineer. *LED Basics.* http://www.baldengineer.com/tutorials/led-basics/ [12 June 2014]

Broadcast Software International. *Network Cabling Guide.* Eugene

Curran, M. 2010. *Controlling High Brightness LEDs Using the dsPIC® “GS” Series of Switch Mode Power Supply Digital Signal Controllers.* 22 August 2010

DAWSON R.D. (1980) LED Bandwidth Improvement by Bipolar Pulsing. IEEE Journal of Quantum Electronics. Volume 7 (QE - 16), pg. 697 to 699.

DIYTrade.com. 2014**.** *Led Dot Matrix Display.* http://www.diytrade.com/china/pd/6240345/Led_Dot_Matrix_Display.html#normal_img [September 2014]

Dr. Alistair Cockburn. (May 2008) Using Both Incremental and Iterative Development. Software Engineering Technology

DS1666, DS1666S. Audio Digital Resistor. 1995, *Dallas Semiconductor*

dsPIC33FJ32MC302/304,dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04, High-Performance, 16-bit Digital Signal Controllers. 2011, *Microchip Technology Inc.*

Flicker fusion threshold Summary | BookRags.com. 2011. Flicker fusion threshold Summary | BookRags.com. [ONLINE] Available at: http://www.bookrags.com/wiki/Flicker_fusion_threshold. [Accessed 15 July 2011].

Floyd T.L. 2006. *Digital Fundamentals*. 9th ed. United States of America

Glasspoole, E.E. 2008. *Fundamentals of Electronics*. Durban: Shades Technical Publications CC.

Global Sources. 2014. *16-segment 2.3-inch Single-digit LED Alphanumeric Display, Used for Instrument Panel.* http://www.globalsources.com/gsol/I/7-segment/p/sm/1053845541.htm#1053845541
[September 2014]

“Green Monster” Has a New Sidekick: Mitsubishi Electric Diamond Vision | Business Wire. 2011. *“Green Monster” Has a New Sidekick: Mitsubishi Electric Diamond Vision | Business Wire.* [ONLINE] Available at: http://www.businesswire.com/news/home/20110405005676/en/%E2%80%9CGreen-Monster%E2%80%9D-Sidekick-Mitsubishi-Electric-Diamond-Vision. [Accessed 12 May 2011].

HEF4094B, 8-stage shift-and-store bus register. 30 October 2008. *NXP Semiconductors*

History of LED. 2011. *History of LED.* [ONLINE] Available at: http://www.solidstatelighting.org/LED___A_History.html. [Accessed 12 May 2011].

History of LED Lights. 2011. *History of LED Lights.* [ONLINE] Available at: http://www.ledlights.org/history-of-led-lights.php. [Accessed 12 May 2011].

HP 9100A/B. 2011. *HP 9100A/B.* [ONLINE] Available at: http://www.hpmuseum.org/hp9100.htm. [Accessed 12 May 2011].

Koschier, M. 2012. Email to Austria Microsystems, 10 July

Light: Optipedia, Free optics information from SPIE. 2011. *Light: Optipedia, Free optics information from SPIE*. [ONLINE] Available at: http://spie.org/x32836.xml. [Accessed 15 July 2011].

LED Alphanumeric Display. 2011. *Ningbo Flying Electronics Co. ,Ltd.* [ONLINE] Available at: http://www.tradevv.com/chinasuppliers/nbflying/product-list/LED-Alphanumeric-Display-catalog-ae4.html. [Accessed 13 May 2011].

LED Colors. 2011. *LED Colors*. [ONLINE] Available at: http://www.hudnut.com/ledcolors.htm. [Accessed 31 May 2011].

MC14013B. Dual Type D Flip-Flop. February 2005, *On Semiconductor*

MCP4021/2/3/4. Low-Cost NV Digital POT with WiperLock^{TM} Technology. 2006 *Microchip*

Microcontroller Interfacing Techniques, 2005. *BiPOM Electronics, Inc.*

M. Stabile. *Controlling a Dot Matrix LED Display with a Microcontroller*. University of California, Santa Barbara: Media Arts and Technology, 2008.

MT943. Lux Meter. *Major Tech*

Nick Holonyak. 2011. *Nick Holonyak*. [ONLINE] Available at: http://www.nndb.com/people/404/000169894/. [Accessed 12 May 2011]

PBSS5320T, PNP Low V_{CEsat} (BISS) transistor. 2004. *Phillips Semiconductor*
PIC16F685/687/689/690, 20-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology. 2005, *Microchip Technology Inc.*

PIC16F818/819, 18/20-Pin Enhanced Flash Microcontrollers with nanoWatt Technology. 2004, *Microchip Technology Inc.*

Pilone, D. & Pitman, N. June 2005. *UML 2.0 in a Nutshell.* United States of America: O'Reilly Media.

PN2222A/MMBT2222A/PZT2222A, NPN General Purpose Amplifier. 2009. *Fairchild Semiconductor*

Preetham, A.J.(2002) Modeling Skylight and Aerial Perspective.

Raspberry Pi. 2013. [ONLINE] Available at: http://www.raspberrypi.org/ [Accessed 12 February 2013].

Rayleigh Scattering - OPT Telescopes. 2011. *Rayleigh Scattering - OPT Telescopes*. [ONLINE] Available at: http://www.optcorp.com/edu/articleDetailEDU.aspx?aid=2265. [Accessed 31 May 2011].

RJ-45. EIGHT CONTACT, TEN POSISTION, VERTICAL MOUNT, UNSHIELDED JACK, "-NF" NON-FLANGE. 28 January 2003, *Bel Stewart Connector*

SanDisk, SanDisk SD Card. 2004. *SanDisk Corporation*

Super material will make lighting cheaper and fully recyclable. 2011. *Super material will make lighting cheaper and fully recyclable.* [ONLINE] Available at: http://www.sciencedaily.com/releases/2010/02/100205115810.htm. [Accessed 12 May 2011].

TC23-11SRWA, 58mm (2.3 INCH) 8x8 DOT MATRIX DISPLAYS. 2001, *Kingbright*

TC23-11SYKWA, 58.34mm (2.3 INCH) 8x8 DOT MATRIX DISPLAY. 2011, *Kingbright*

TEMT1000, TEMT1020, TEMT1030,TEMT1040. Silicon NPN Phototransistor, RoHS Compliant. 9 September 2008, *Vishay Semiconductors*

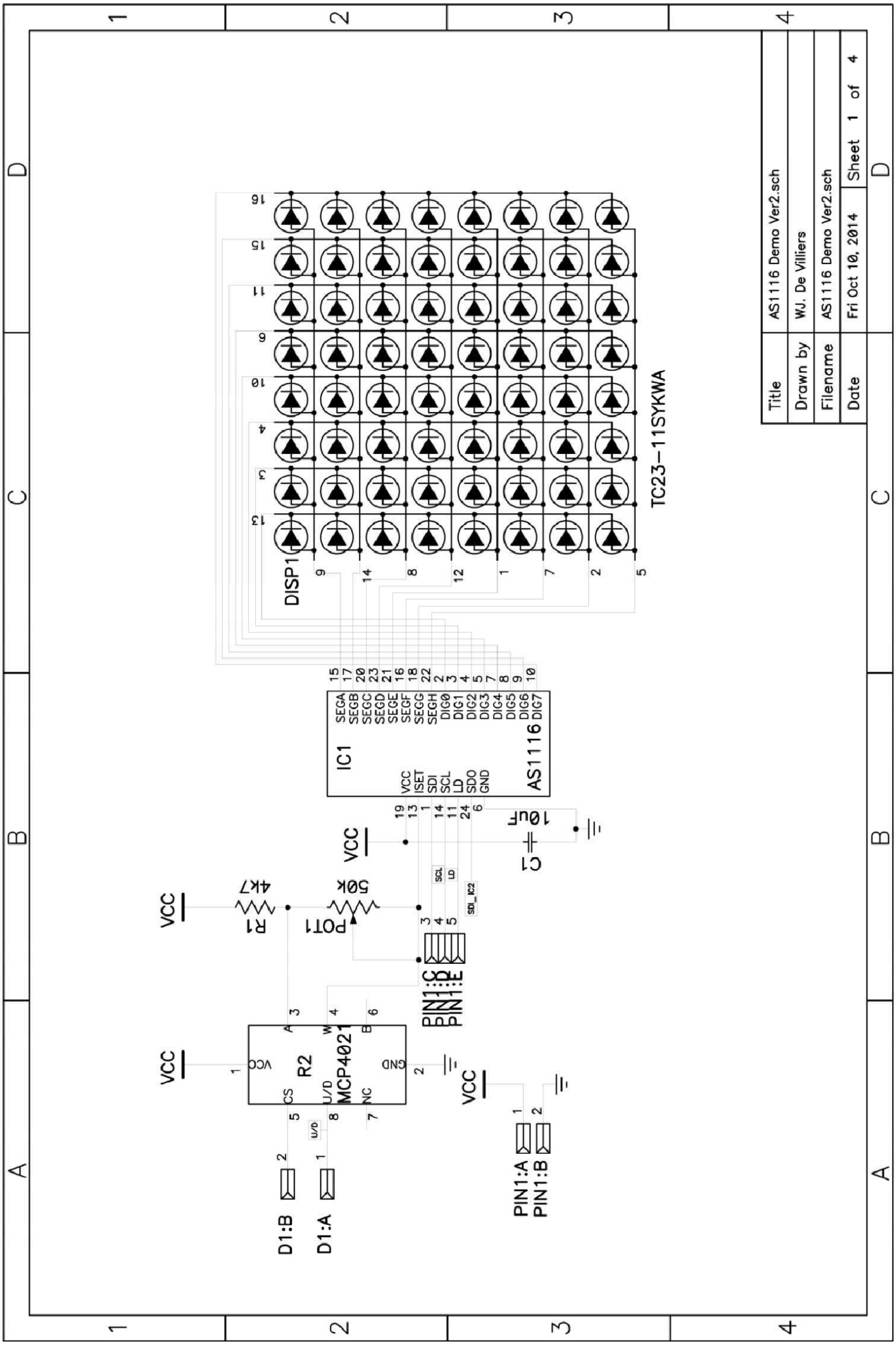
The LED Light, Inc. 2014. *LED Application Notes - LED Basics*. http://www.theledlight.com/technical1.html. [12 June 2014]

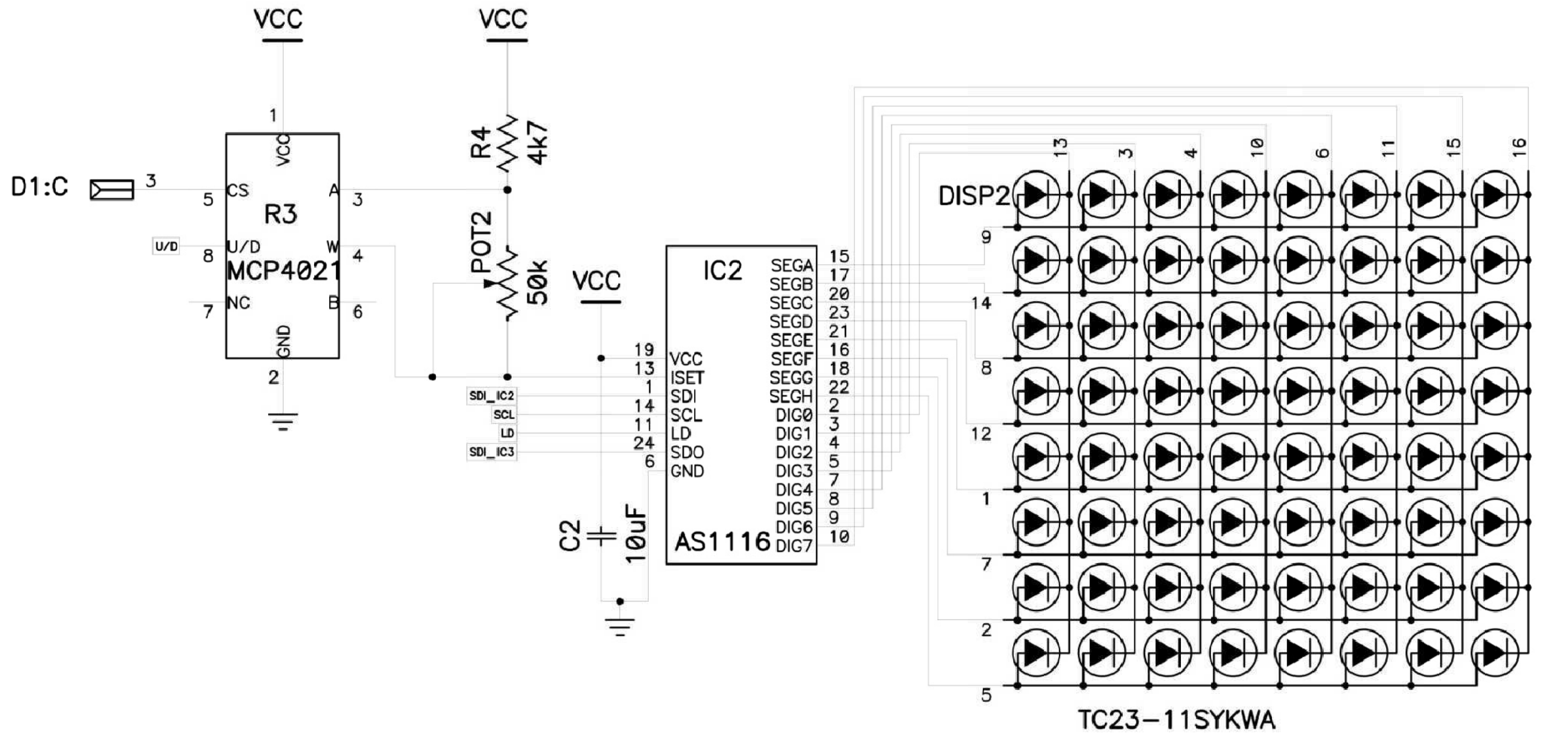
Timeline on the development of the Light Emitting Diode – Seer LED – Sustainable Lighting Solutions. 2011. *Timeline on the development of the Light Emitting Diode – Seer LED – Sustainable Lighting Solutions.* [ONLINE] Available at: http://www.seerled.com.au/leducation.html. [Accessed 12 May 2011].

Toshiba. 2000. *ASIC Packaging Guidebook.* June 2000
Upton, E. & Halfacree, G. 2012. *Raspberry Pi User Guide.* Great Brittan & United States: A John Wiley and Sons.

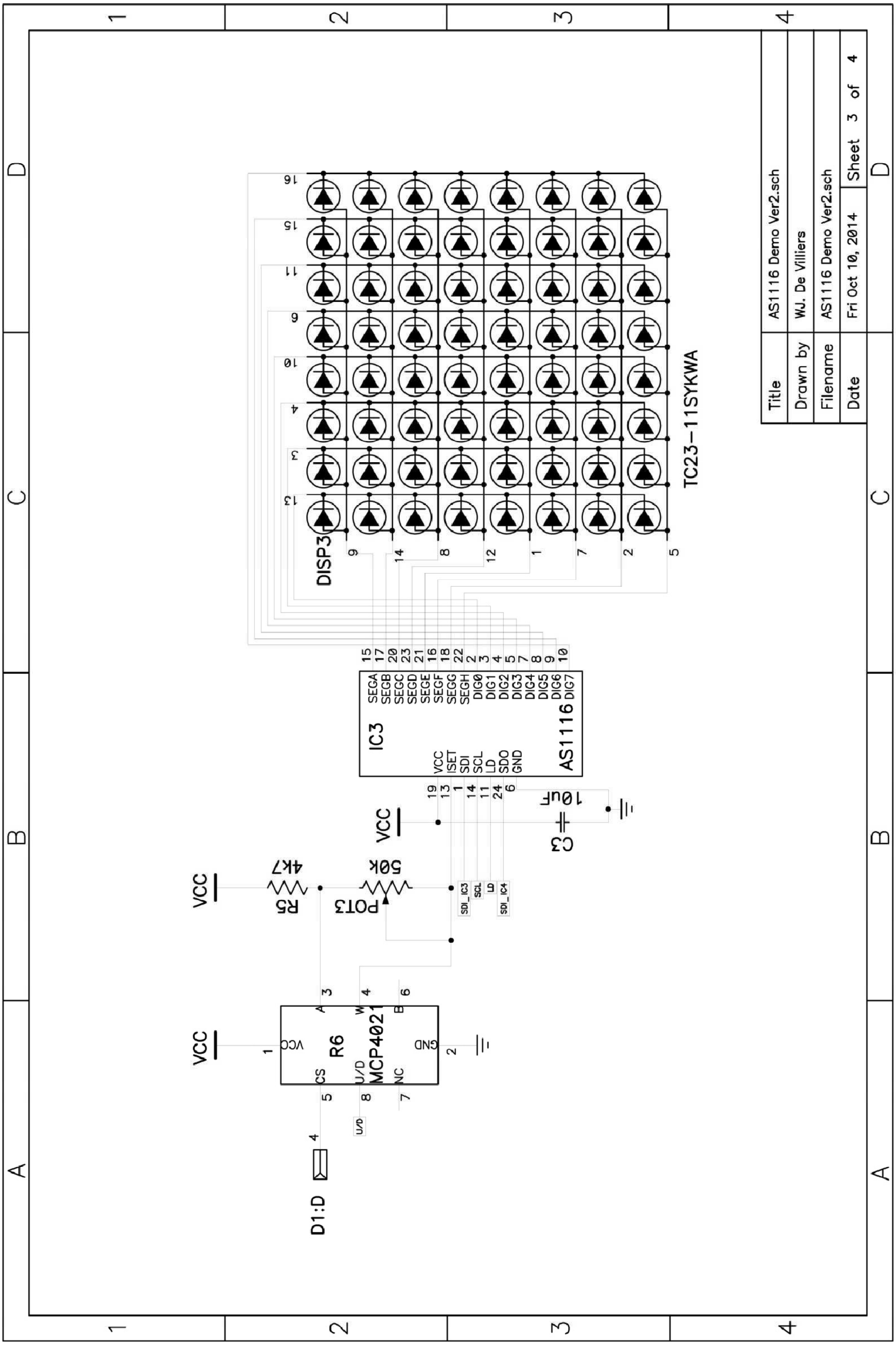
Visible Light and the Eye's Response. 2011. *Visible Light and the Eye's Response*. [ONLINE] Available at: http://www.physicsclassroom.com/class/light/u12l2b.cfm. [Accessed 31 May 2011].

Appendix A - AS1116 Schematic

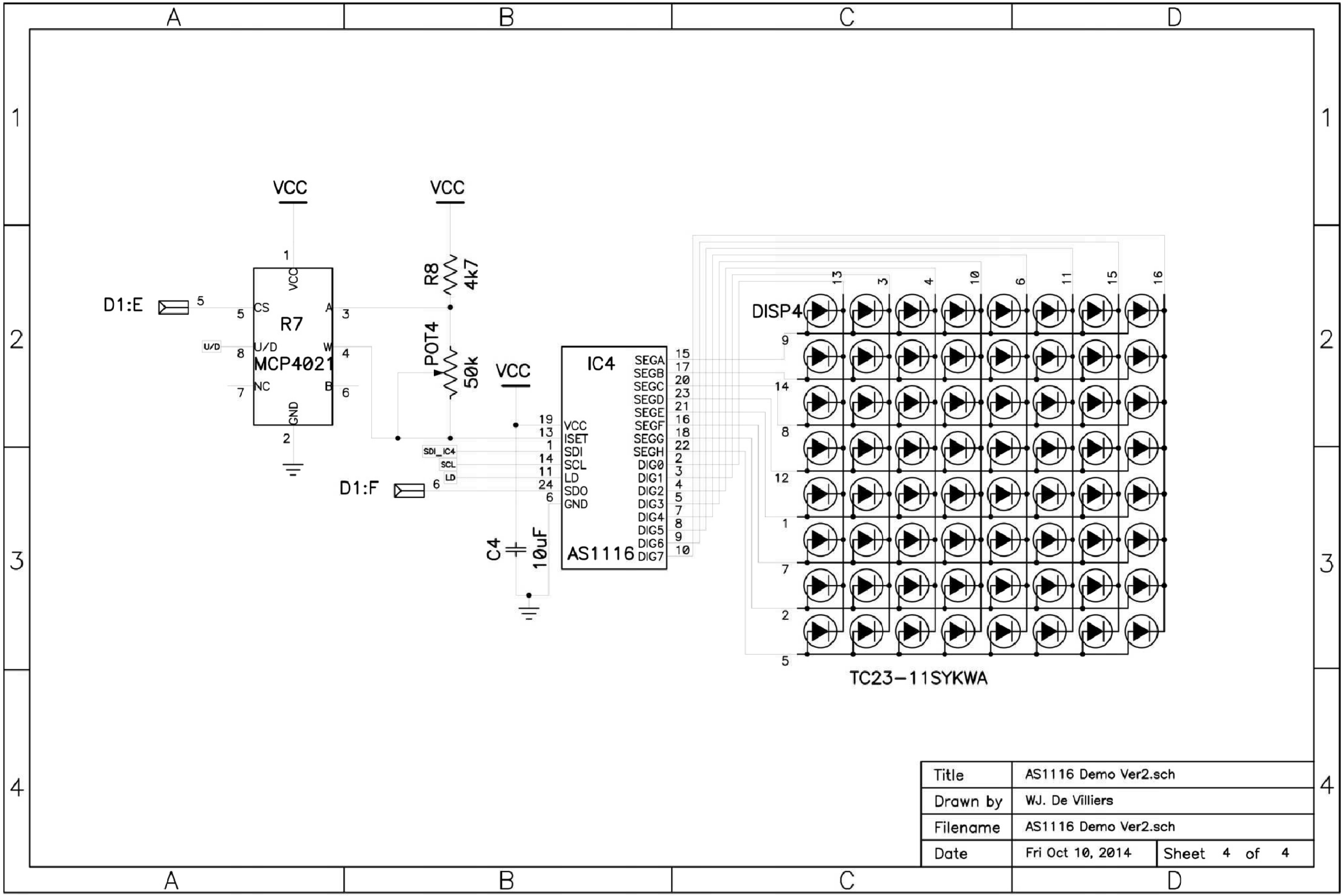




Title	AS1116 Demo Ver2.sch		
Drawn by	WJ. De Villiers		
Filename	AS1116 Demo Ver2.sch		
Date	Fri Oct 10, 2014	Sheet 2 of 4	

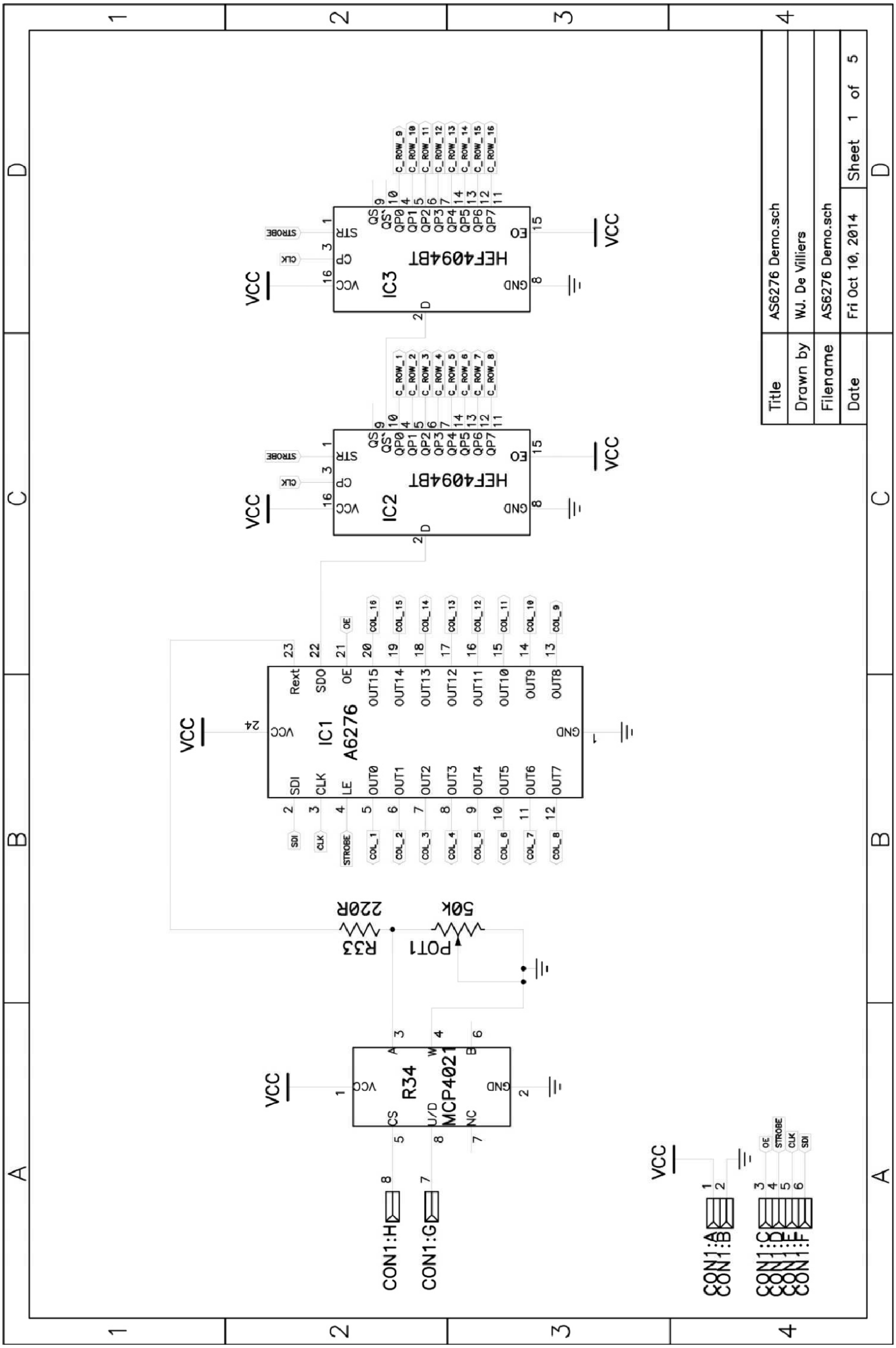


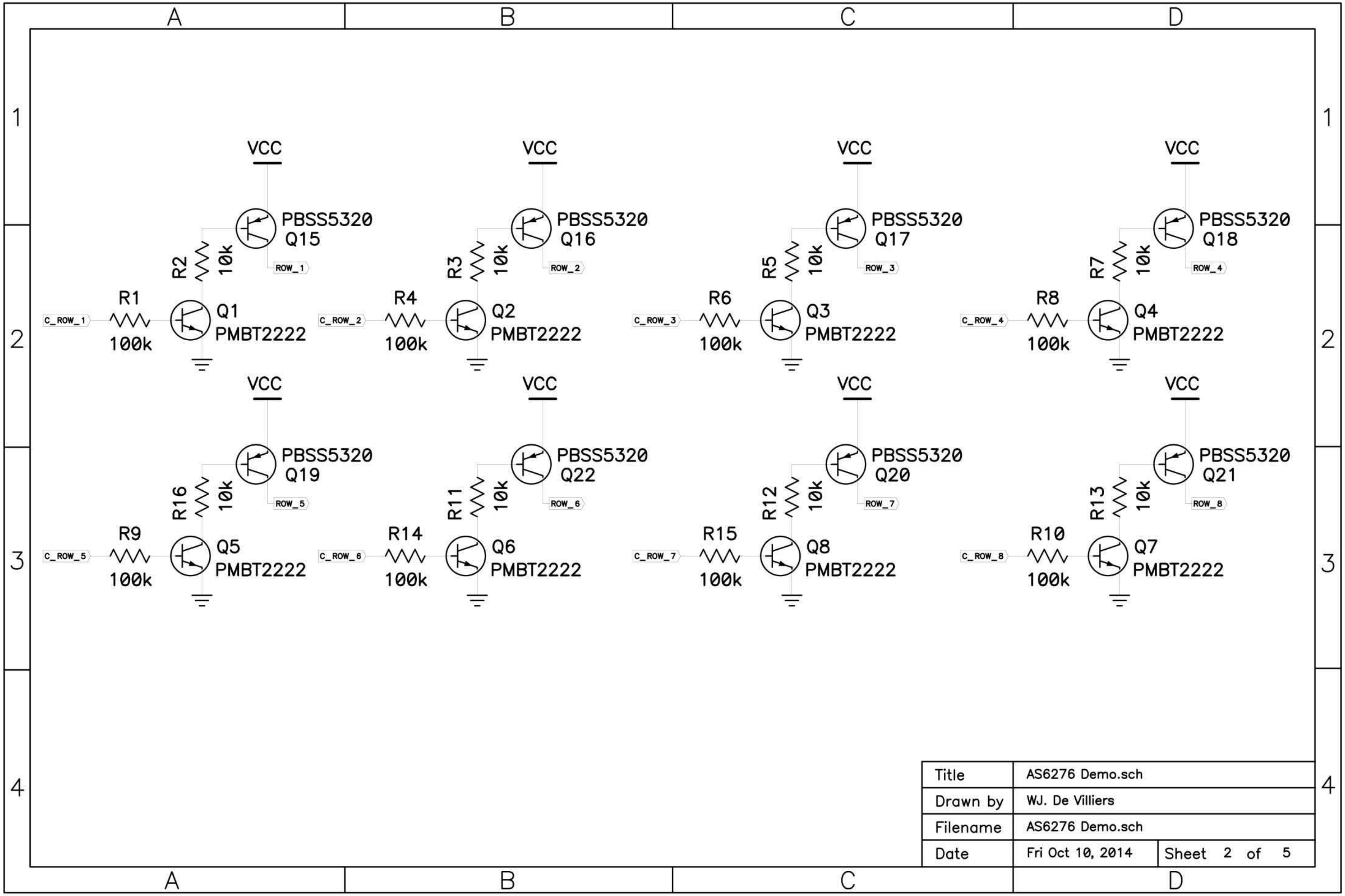
AS1116 Demo Ver2.sch
WJ. De Villiers
AS1116 Demo Ver2.sch
Fri Oct 10, 2014
Sheet 3 of 4
Title
Drawn by
Filename
Date
DISP3
TC23-11SYKWA
IC3
AS1116
SEGA
SEGB
SEGC
SEGD
SEGE
SEGF
SEGG
SEGH
DIG0
DIG1
DIG2
DIG3
DIG4
DIG5
DIG6
DIG7
VCC
ISET
SDI
SCL
LD
SDO
GND
VCC
C3
10uF
R5
4K7
POT3
50K
SDI_IC3
SCL
LD
SDI_IC4
R6
MCP4021
VCC
CS
U/D
NC
A
W
B
GND
U/D
D1:D



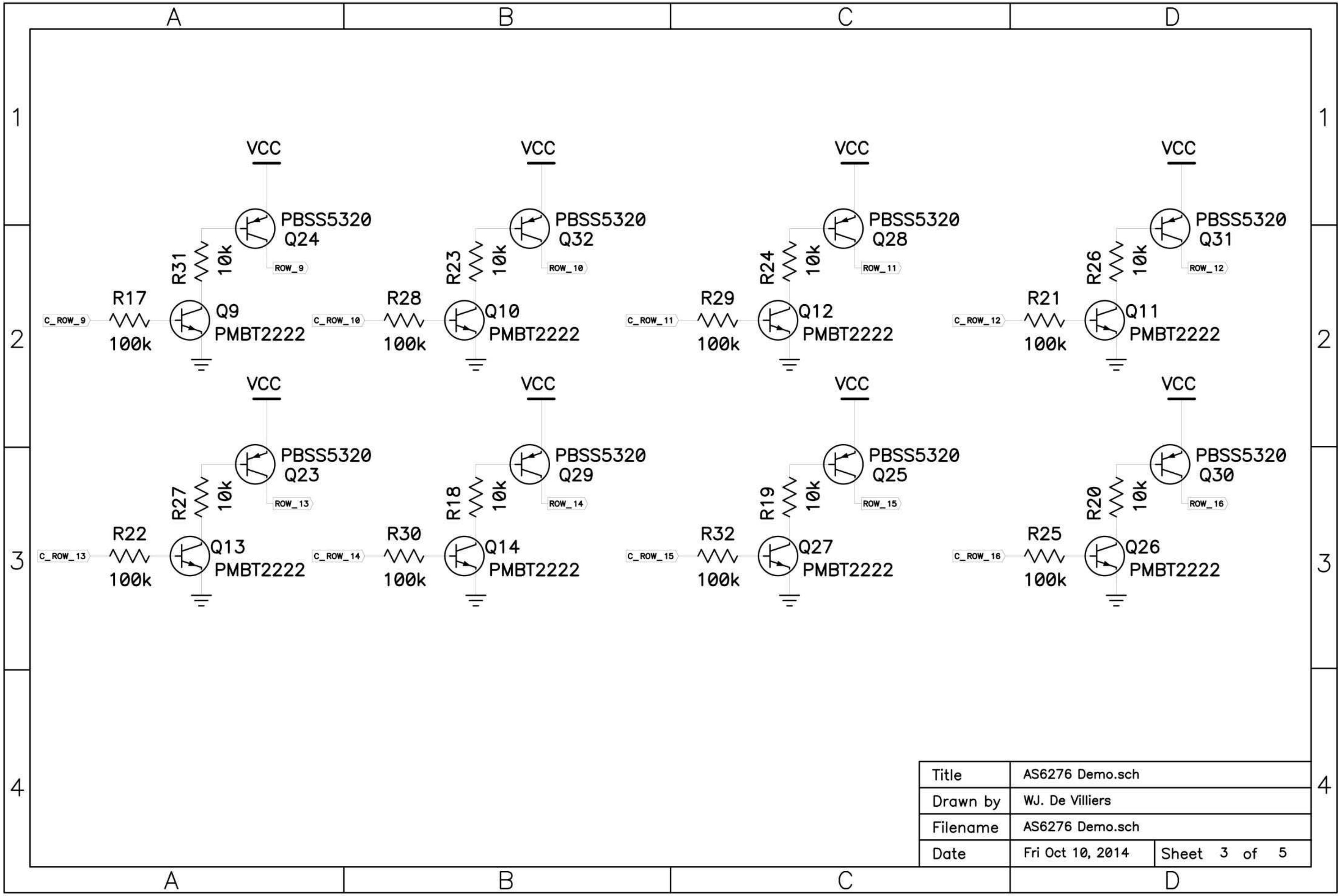
Title	AS1116 Demo Ver2.sch
Drawn by	WJ. De Villiers
Filename	AS1116 Demo Ver2.sch
Date	Fri Oct 10, 2014 — Sheet 4 of 4

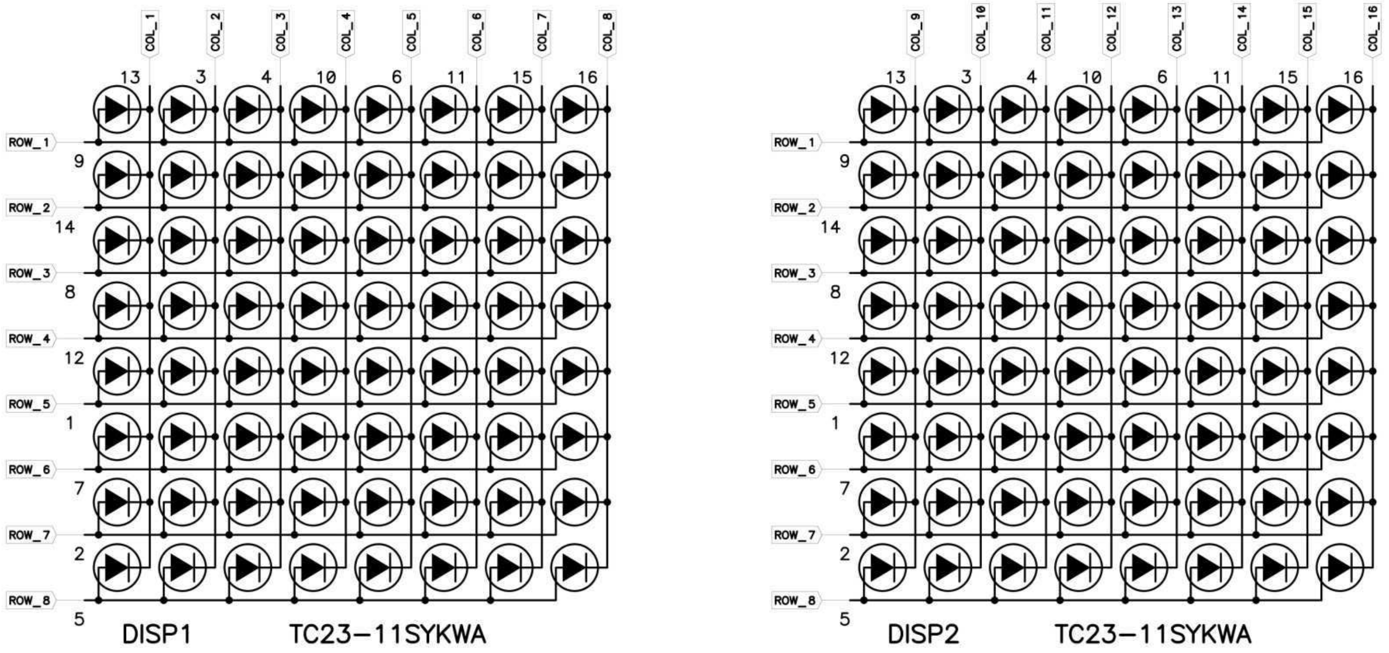
Appendix B - A6276 Schematic



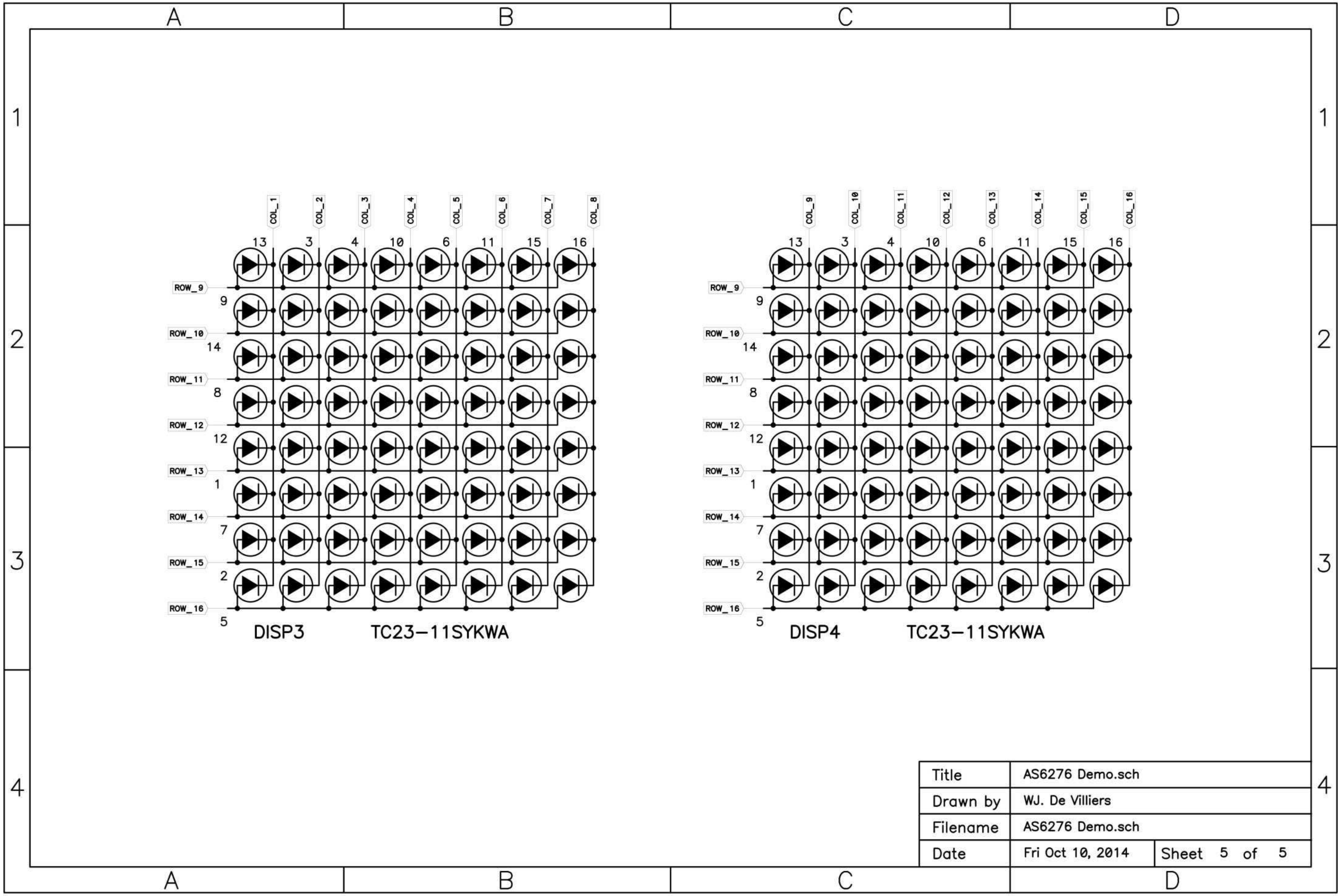


Title	AS6276 Demo.sch	
Drawn by	WJ. De Villiers	
Filename	AS6276 Demo.sch	
Date	Fri Oct 10, 2014	Sheet 2 of 5


A
B
C
D
1
2
3
4
VCC
PBSS5320
Q24
Q32
Q28
Q31
Q23
Q29
Q25
Q30
R31
R23
R24
R26
R27
R18
R19
R20
10k
ROW_9
ROW_10
ROW_11
ROW_12
ROW_13
ROW_14
ROW_15
ROW_16
C_ROW_9
C_ROW_10
C_ROW_11
C_ROW_12
C_ROW_13
C_ROW_14
C_ROW_15
C_ROW_16
R17
R28
R29
R21
R22
R30
R32
R25
100k
Q9
Q10
Q12
Q11
Q13
Q14
Q27
Q26
PMBT2222
Title
AS6276 Demo.sch
Drawn by
WJ. De Villiers
Filename
AS6276 Demo.sch
Date
Fri Oct 10, 2014
Sheet 3 of 5

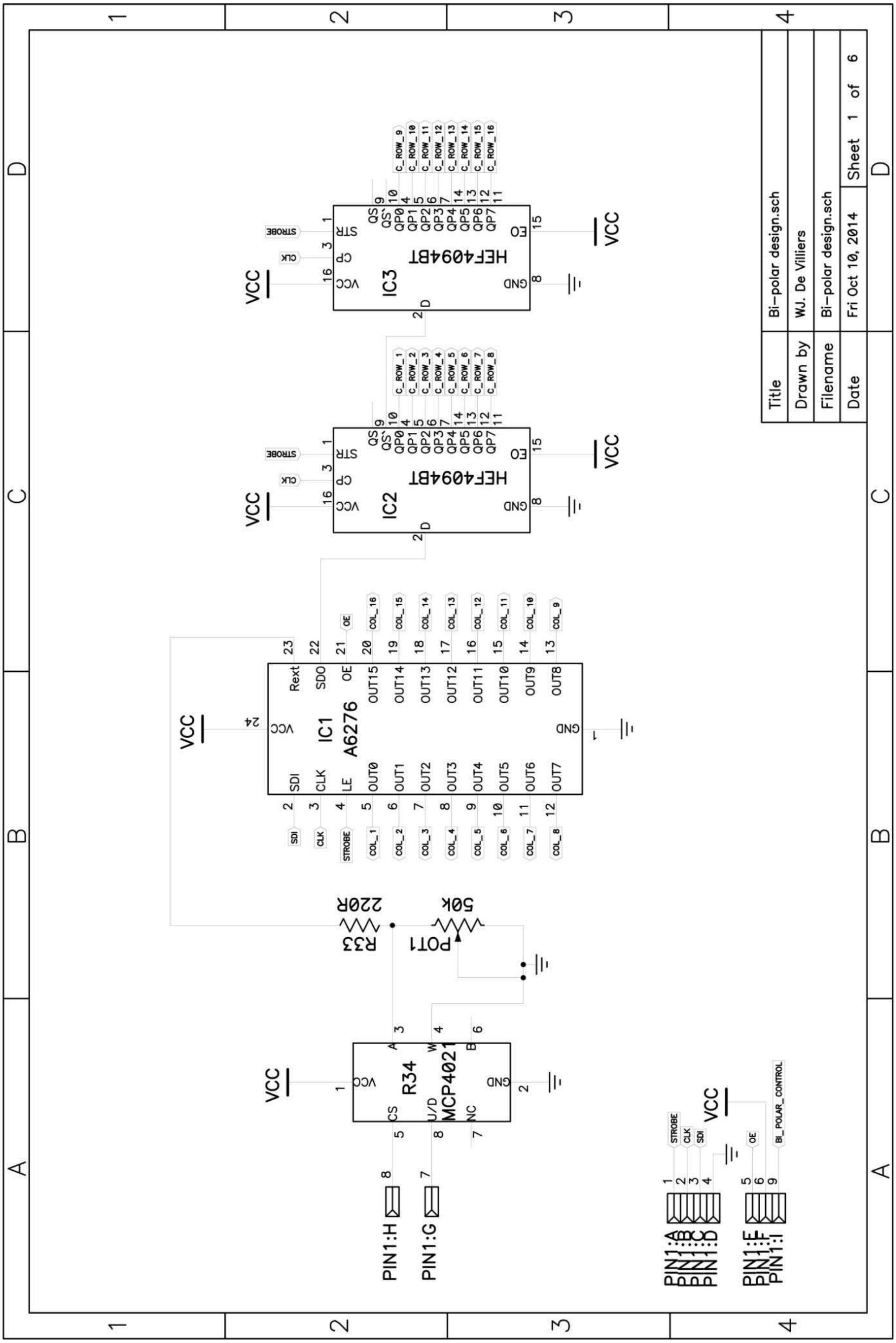


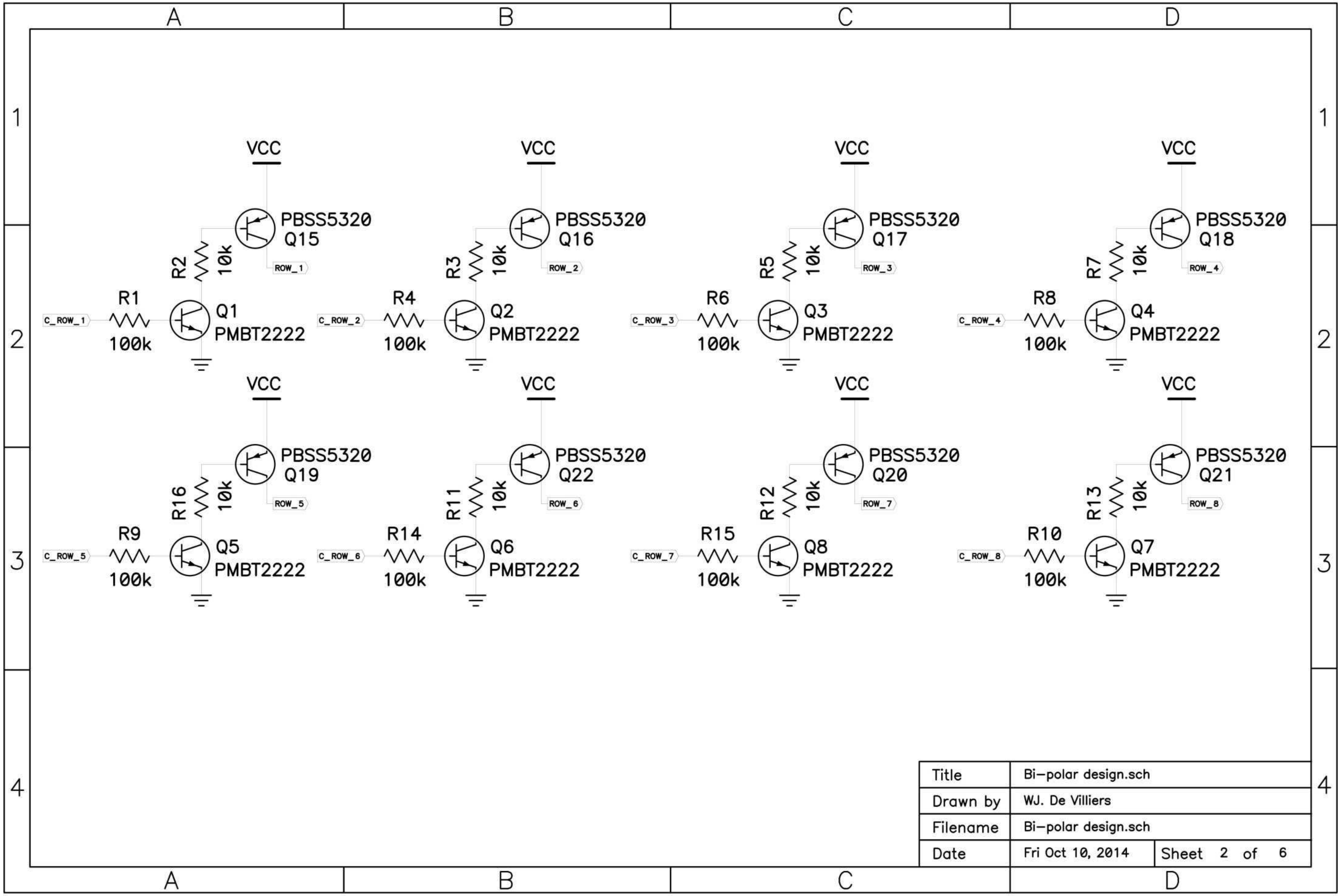
Title	AS6276 Demo.sch	
Drawn by	WJ. De Villiers	
Filename	AS6276 Demo.sch	
Date	Fri Oct 10, 2014	Sheet 4 of 5



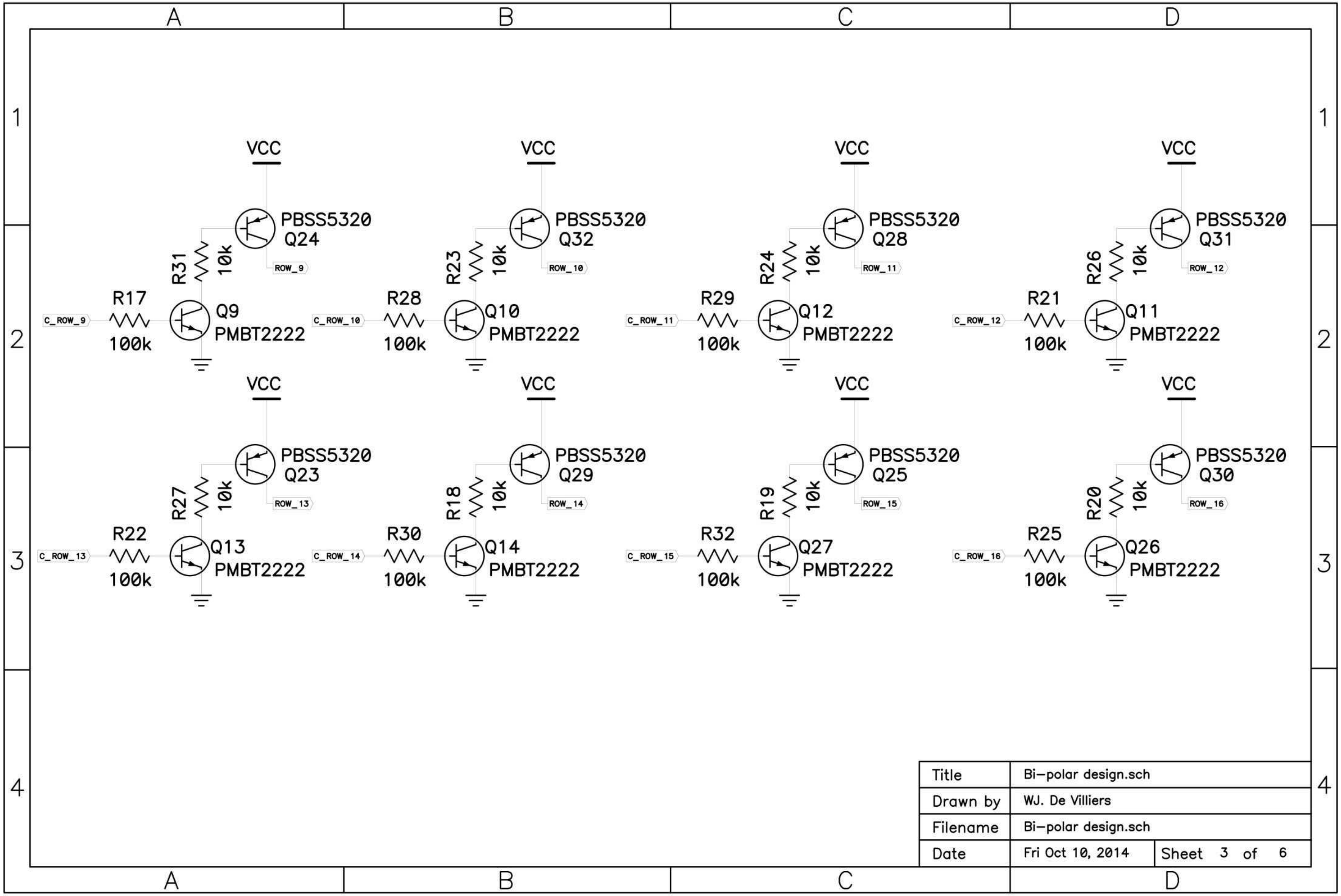
Title	AS6276 Demo.sch		
Drawn by	WJ. De Villiers		
Filename	AS6276 Demo.sch		
Date	Fri Oct 10, 2014	Sheet 5 of 5	

Appendix C - Bi-polar Schematic

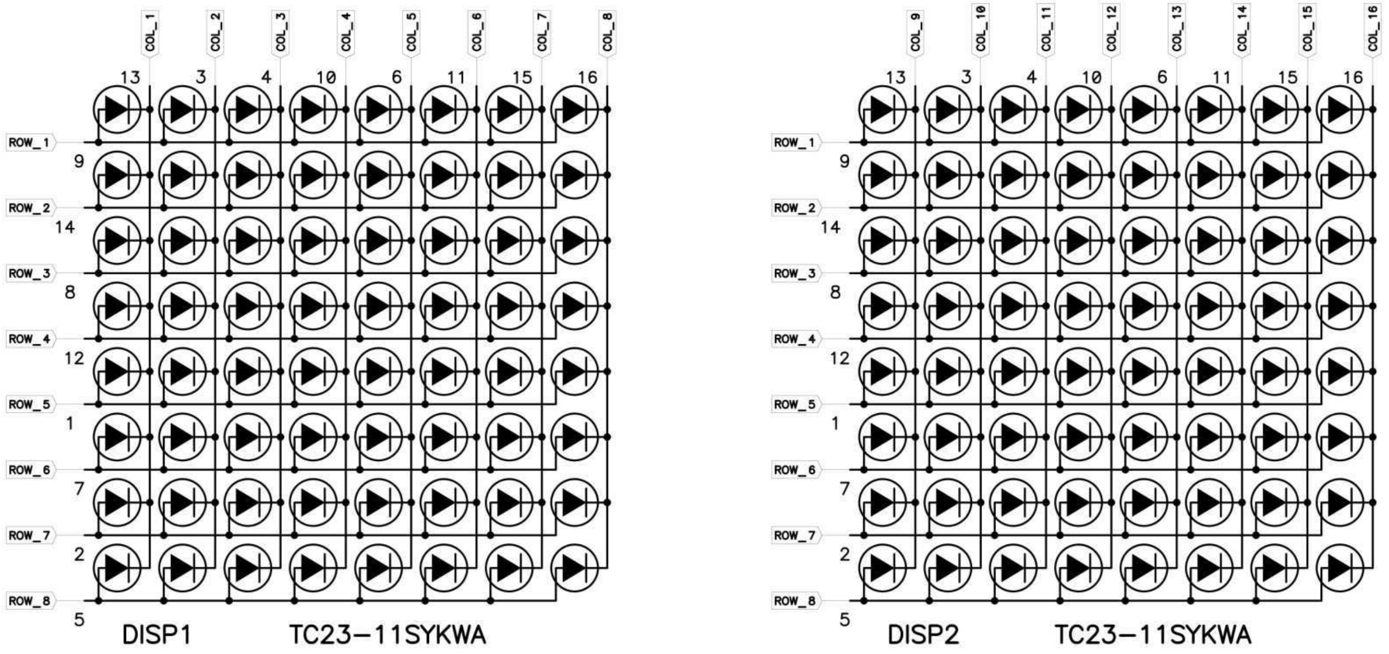




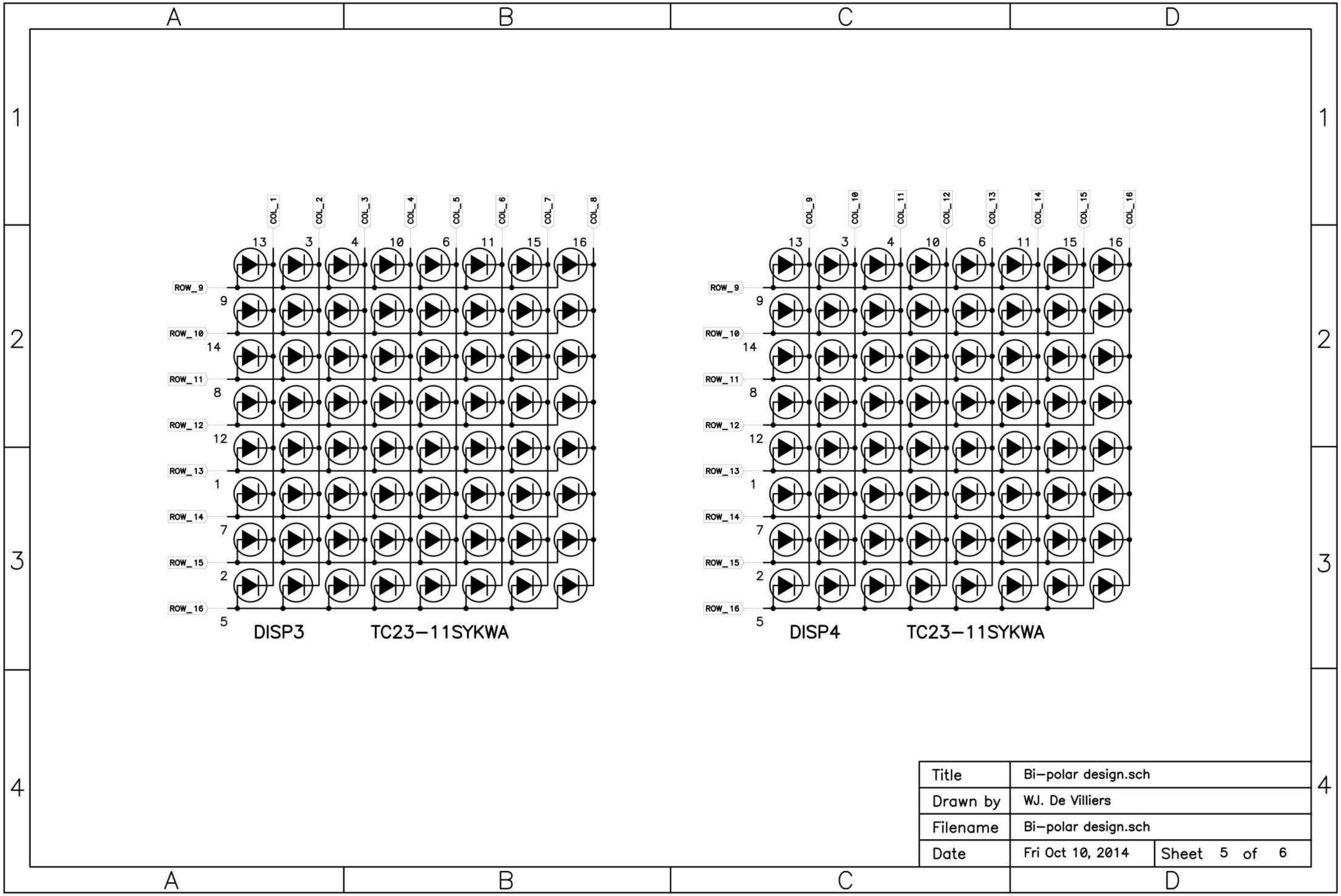
Title	Bi–polar design.sch	
Drawn by	WJ. De Villiers	
Filename	Bi–polar design.sch	
Date	Fri Oct 10, 2014	Sheet 2 of 6



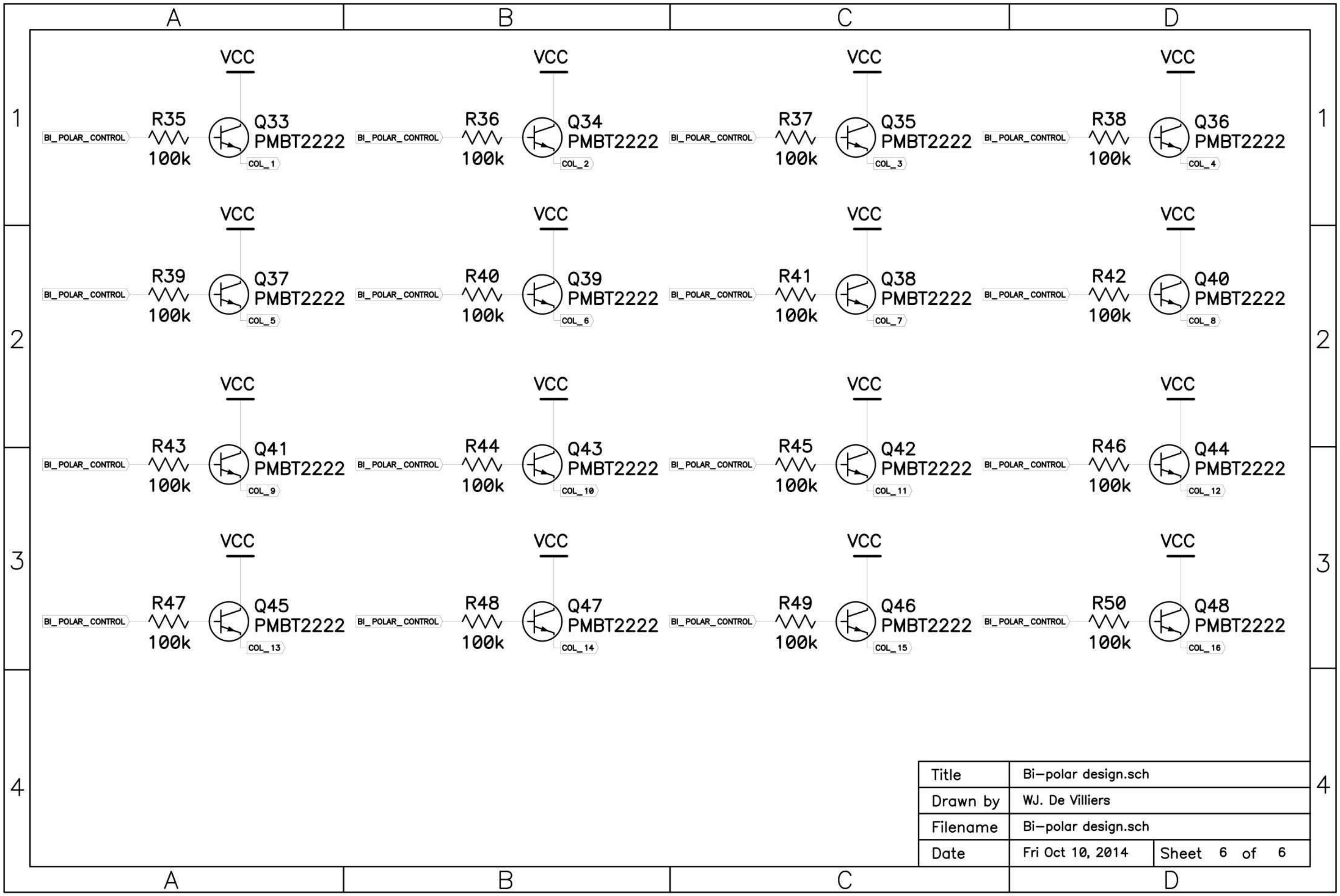
Title	Bi–polar design.sch	
Drawn by	WJ. De Villiers	
Filename	Bi–polar design.sch	
Date	Fri Oct 10, 2014	Sheet 3 of 6



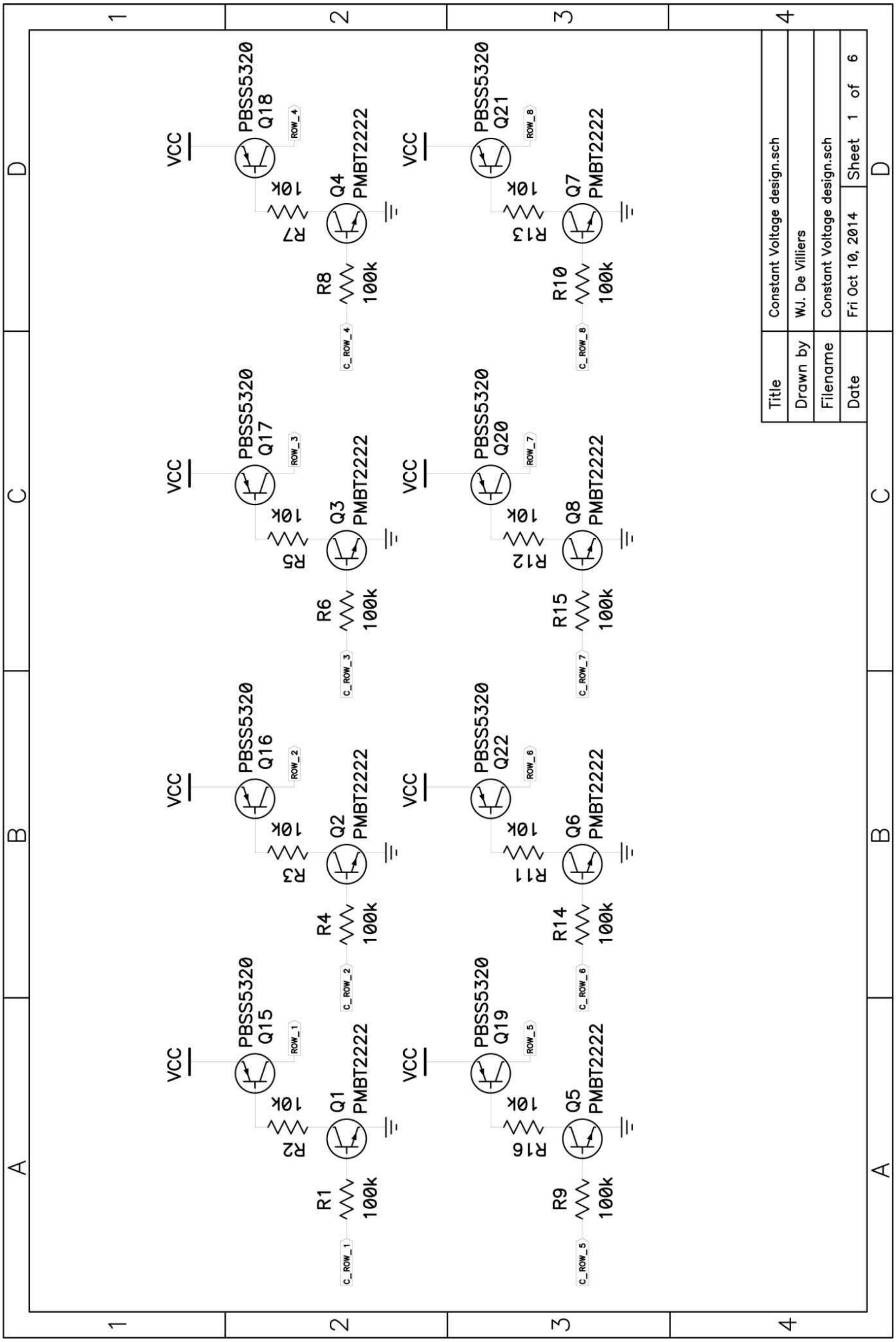
Title	Bi−polar design.sch	
Drawn by	WJ. De Villiers	
Filename	Bi−polar design.sch	
Date	Fri Oct 10, 2014	Sheet 4 of 6

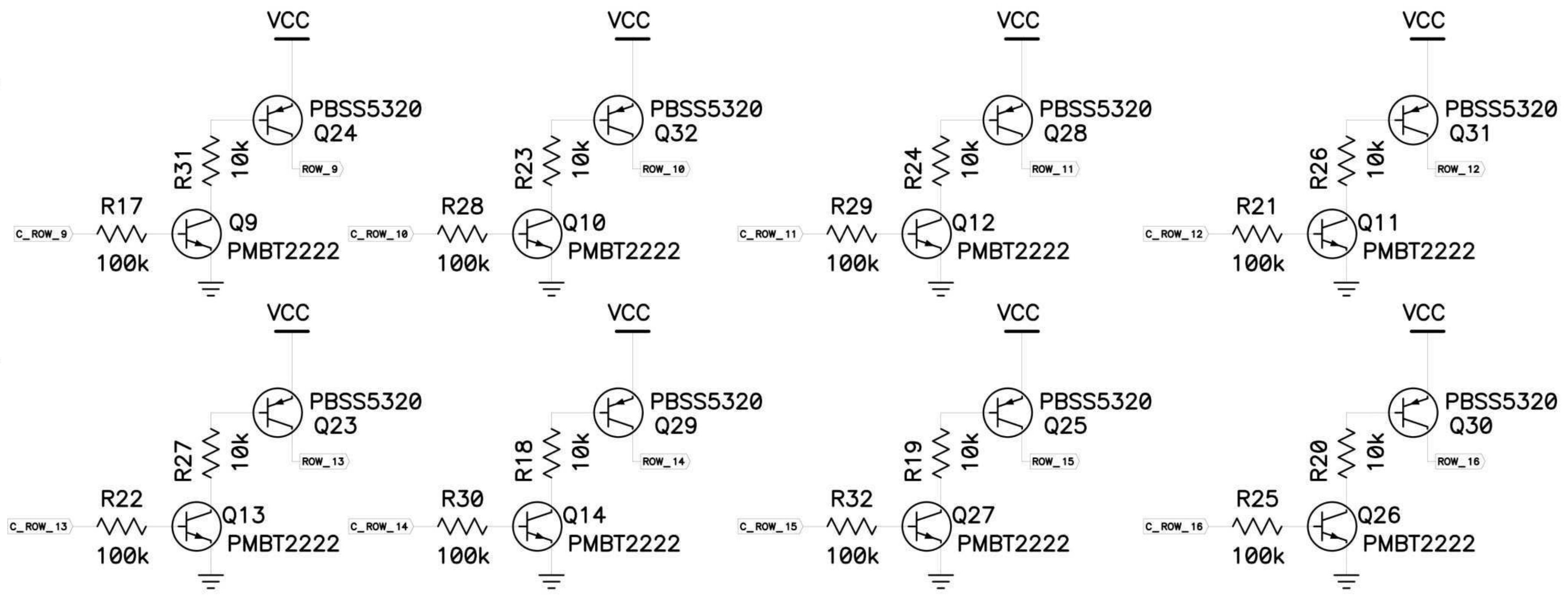


Title	Bi−polar design.sch	
Drawn by	WJ. De Villiers	
Filename	Bi−polar design.sch	
Date	Fri Oct 10, 2014	Sheet 5 of 6

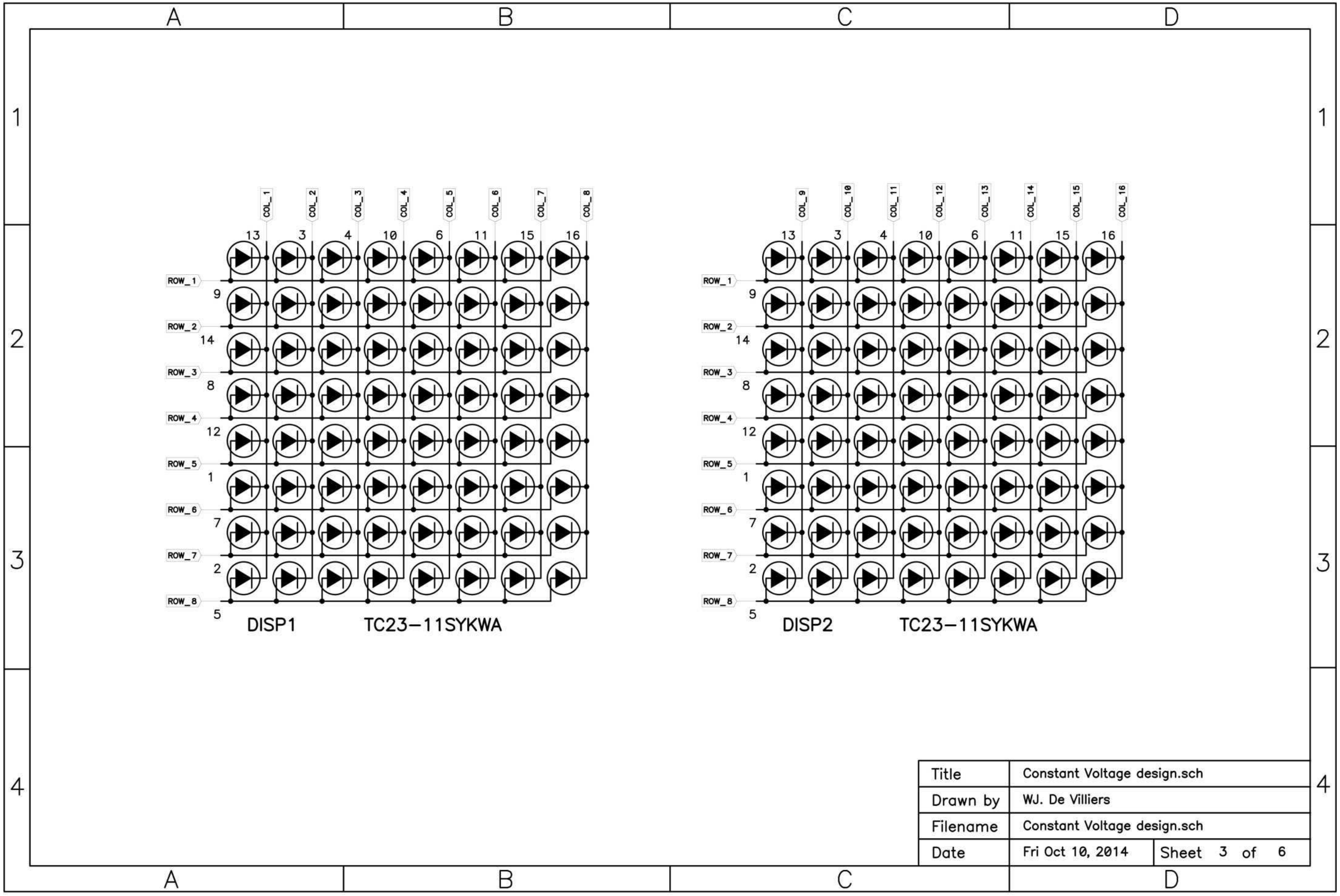

A
B
C
D
1
2
3
4
VCC
BI_POLAR_CONTROL
R35
100k
Q33
PMBT2222
COL_1
R36
100k
Q34
PMBT2222
COL_2
R37
100k
Q35
PMBT2222
COL_3
R38
100k
Q36
PMBT2222
COL_4
R39
100k
Q37
PMBT2222
COL_5
R40
100k
Q39
PMBT2222
COL_6
R41
100k
Q38
PMBT2222
COL_7
R42
100k
Q40
PMBT2222
COL_8
R43
100k
Q41
PMBT2222
COL_9
R44
100k
Q43
PMBT2222
COL_10
R45
100k
Q42
PMBT2222
COL_11
R46
100k
Q44
PMBT2222
COL_12
R47
100k
Q45
PMBT2222
COL_13
R48
100k
Q47
PMBT2222
COL_14
R49
100k
Q46
PMBT2222
COL_15
R50
100k
Q48
PMBT2222
COL_16
Title Bi–polar design.sch
Drawn by WJ. De Villiers
Filename Bi–polar design.sch
Date Fri Oct 10, 2014
Sheet 6 of 6

Appendix D - Constant Voltage design Schematic

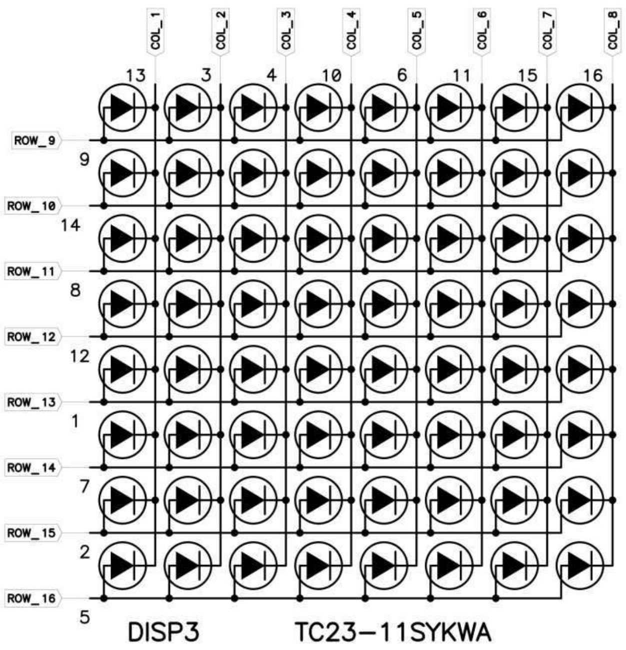


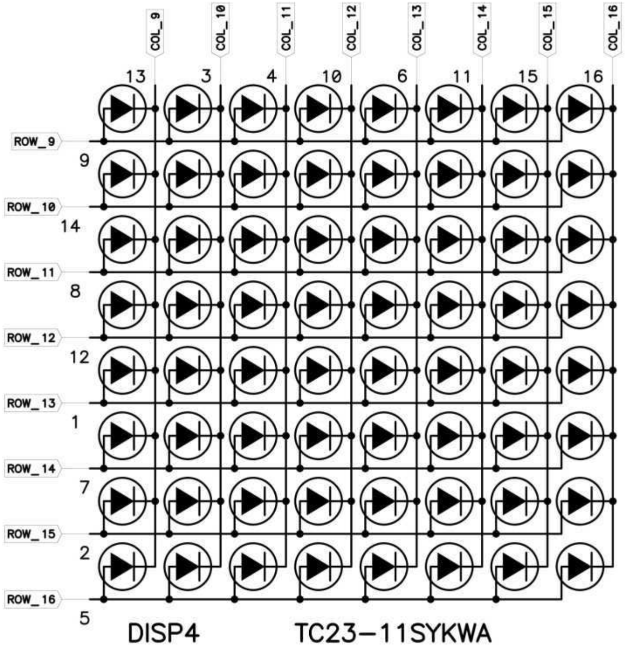


Title	Constant Voltage design.sch	
Drawn by	WJ. De Villiers	
Filename	Constant Voltage design.sch	
Date	Fri Oct 10, 2014	Sheet 2 of 6

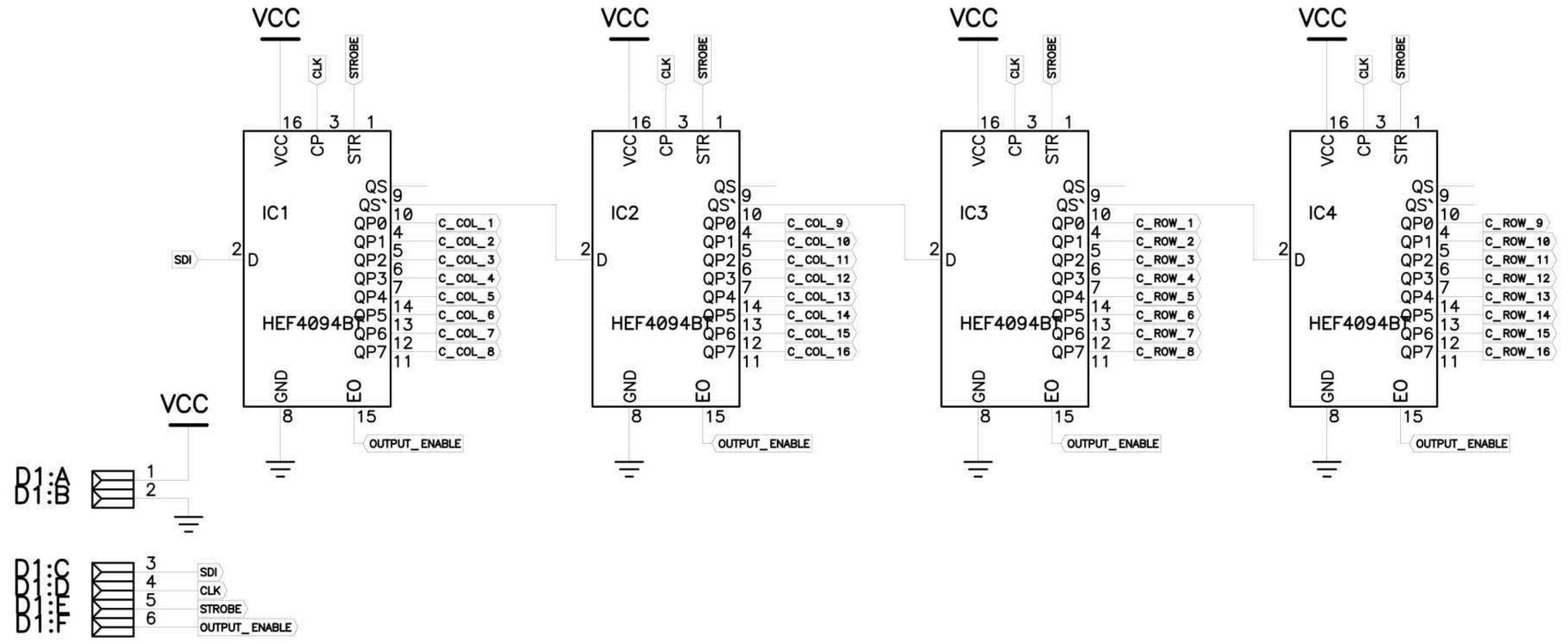


Title	Constant Voltage design.sch	
Drawn by	WJ. De Villiers	
Filename	Constant Voltage design.sch	
Date	Fri Oct 10, 2014	Sheet 3 of 6

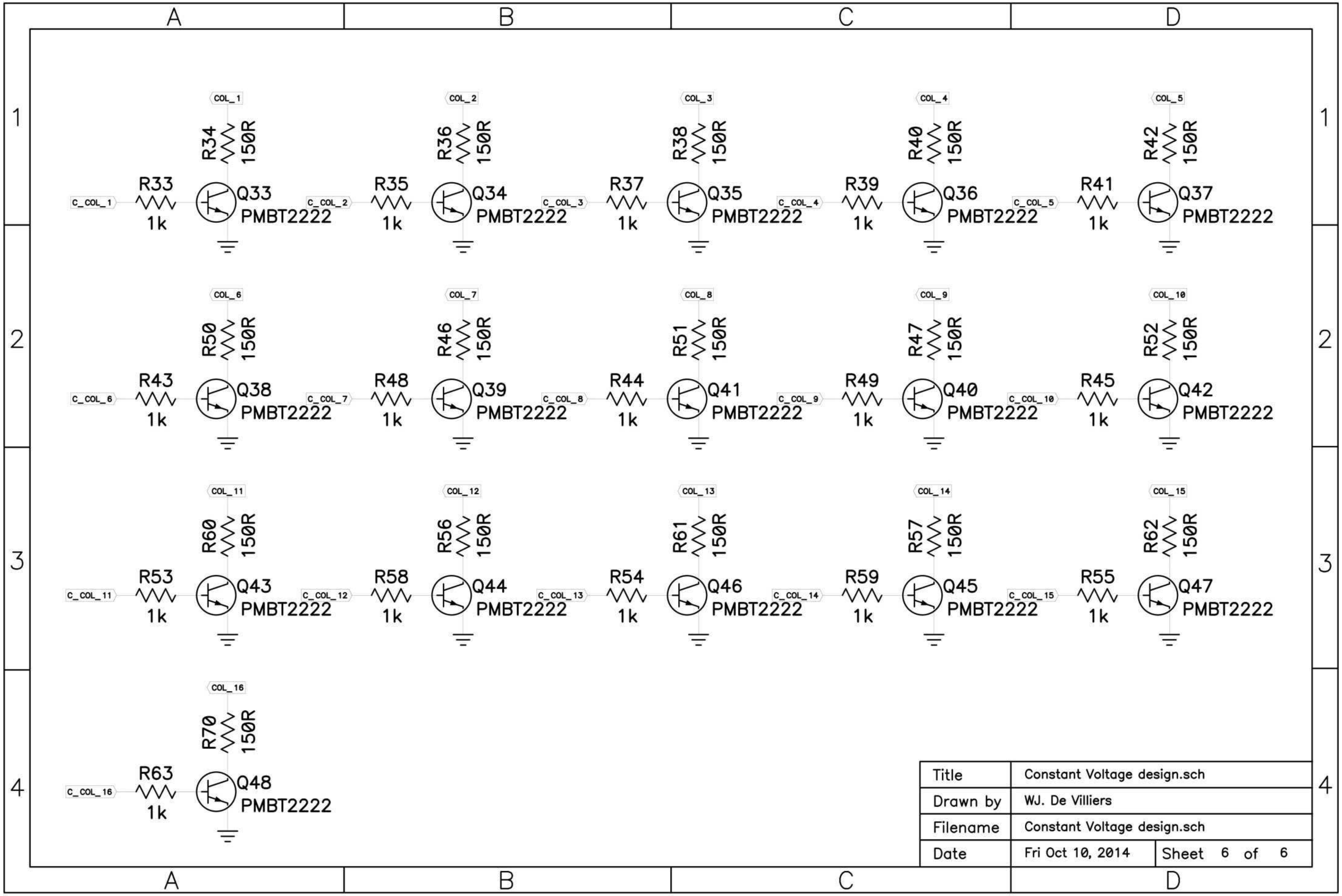




Title	Constant Voltage design.sch	
Drawn by	WJ. De Villiers	
Filename	Constant Voltage design.sch	
Date	Fri Oct 10, 2014	Sheet 4 of 6

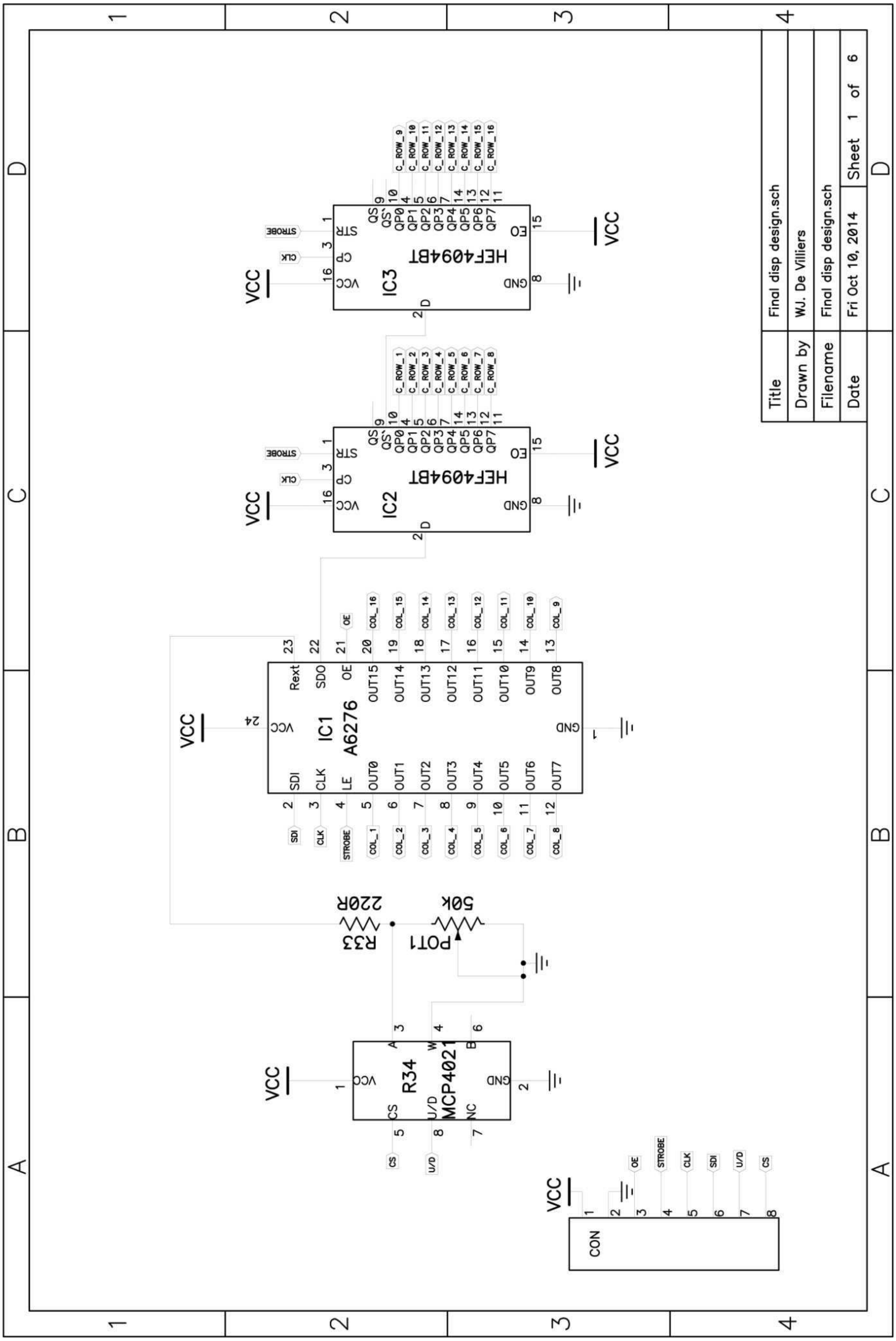


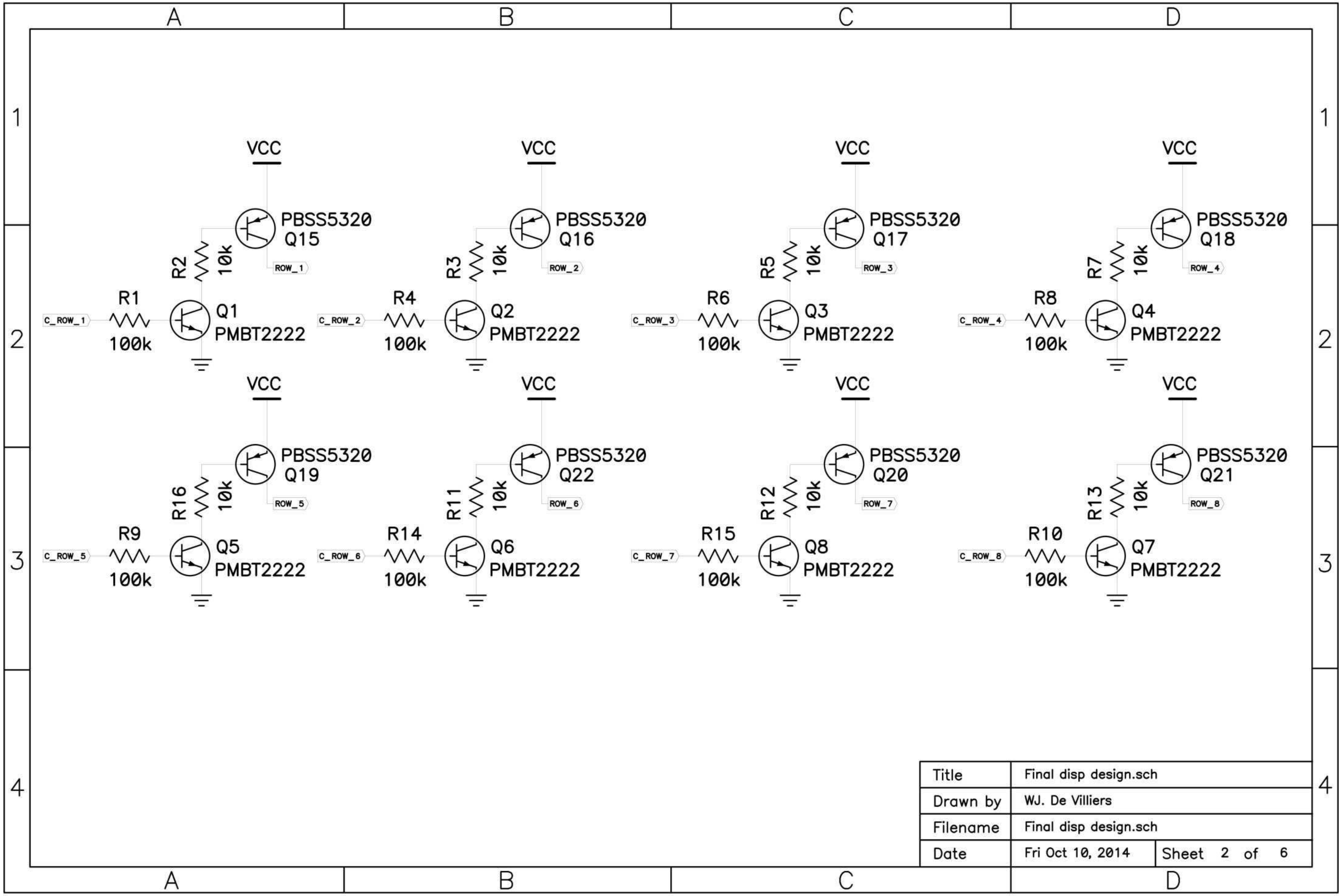
Title	Constant Voltage design.sch		
Drawn by	WJ. De Villiers		
Filename	Constant Voltage design.sch		
Date	Fri Oct 10, 2014	Sheet 5	of 6



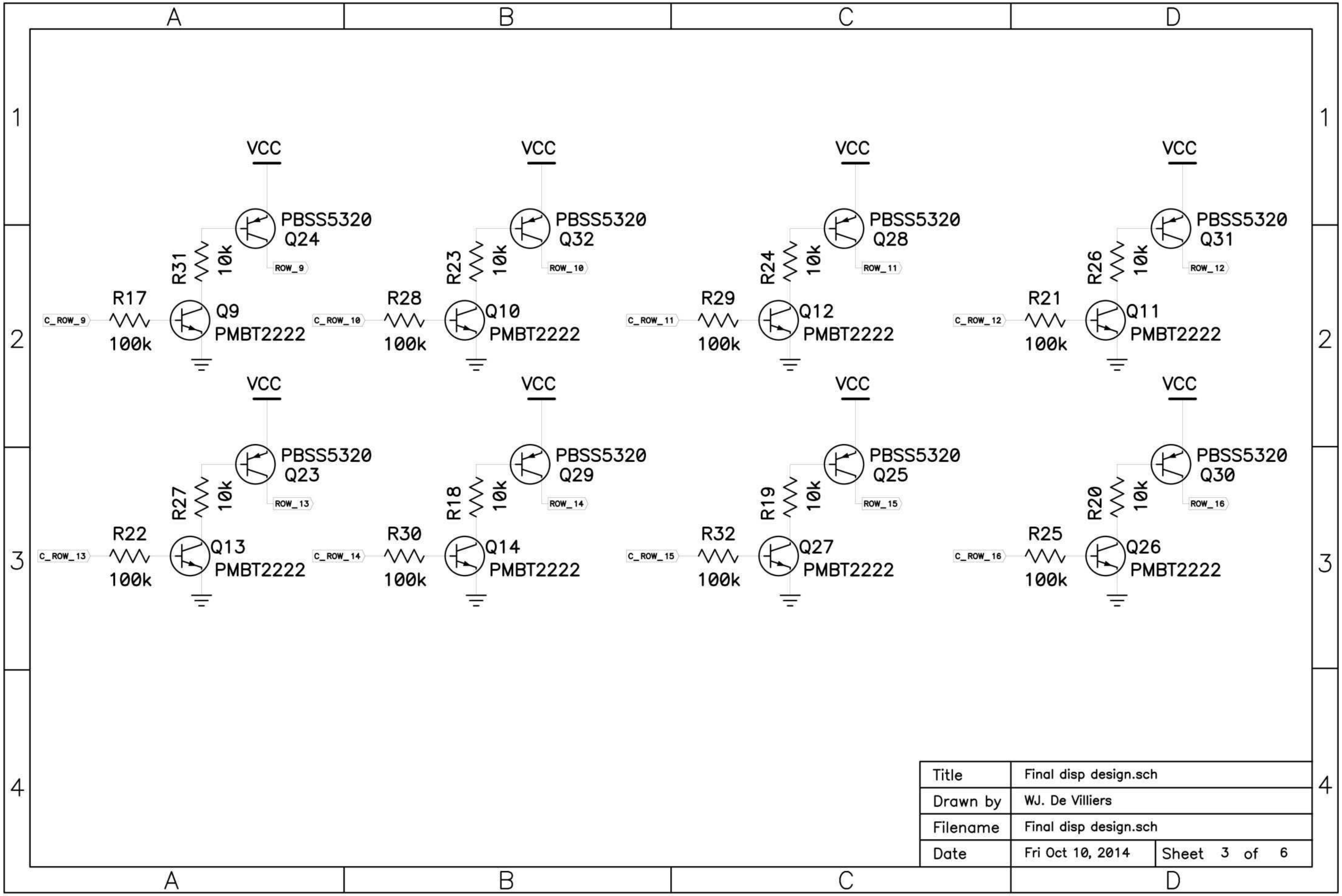
Title	Constant Voltage design.sch		
Drawn by	WJ. De Villiers		
Filename	Constant Voltage design.sch		
Date	Fri Oct 10, 2014	Sheet 6 of 6	

Appendix E - Final display design schematic

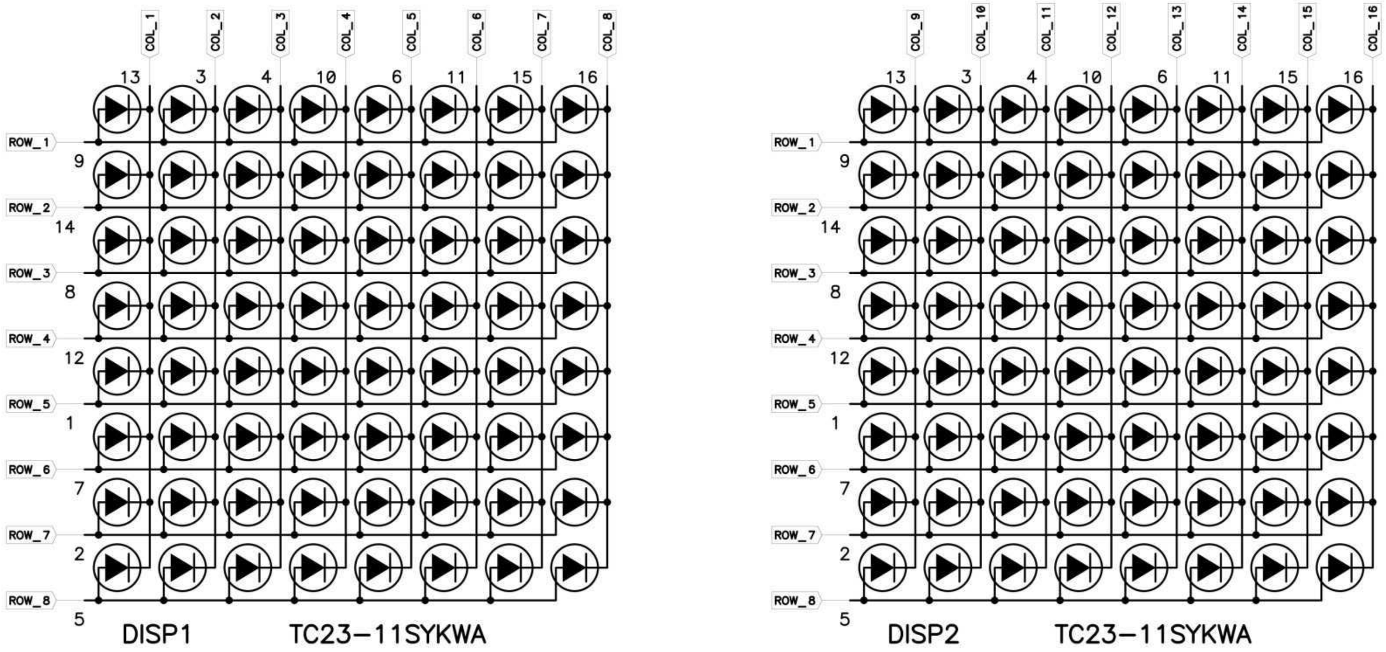




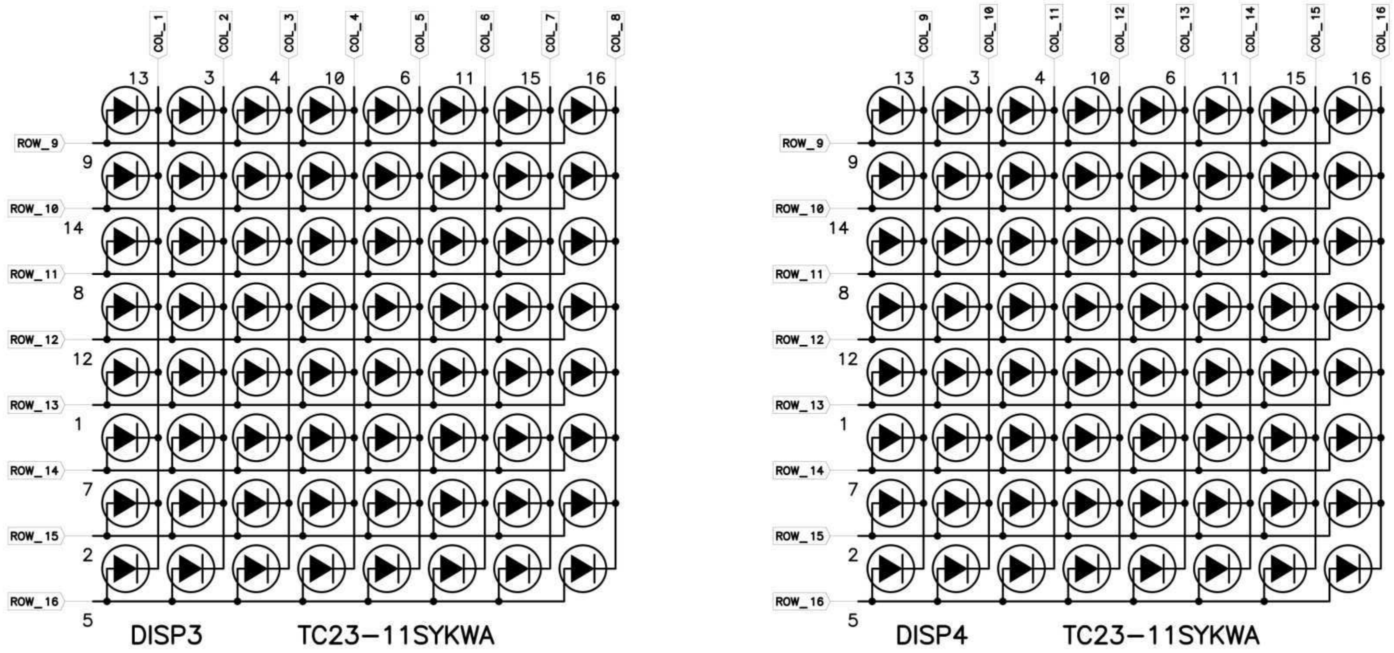
Title	Final disp design.sch		
Drawn by	WJ. De Villiers		
Filename	Final disp design.sch		
Date	Fri Oct 10, 2014	Sheet 2 of 6	



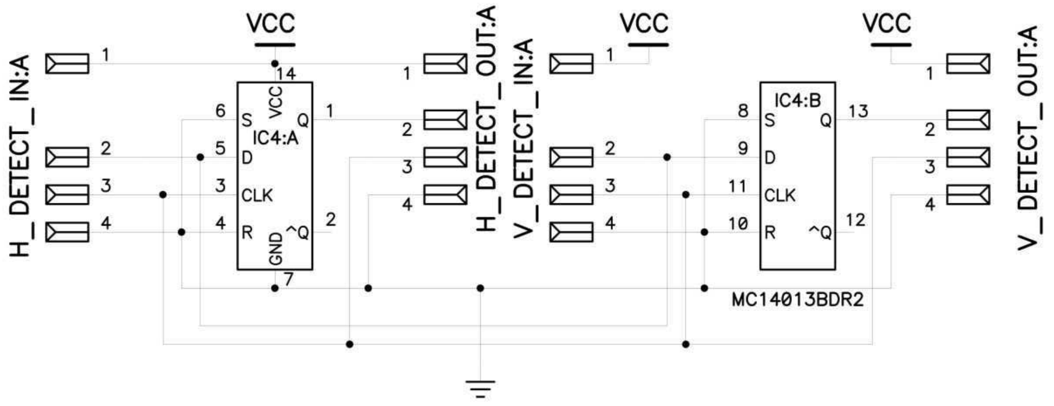
Title	Final disp design.sch	
Drawn by	WJ. De Villiers	
Filename	Final disp design.sch	
Date	Fri Oct 10, 2014	Sheet 3 of 6



Title	Final disp design.sch	
Drawn by	WJ. De Villiers	
Filename	Final disp design.sch	
Date	Fri Oct 10, 2014	Sheet 4 of 6

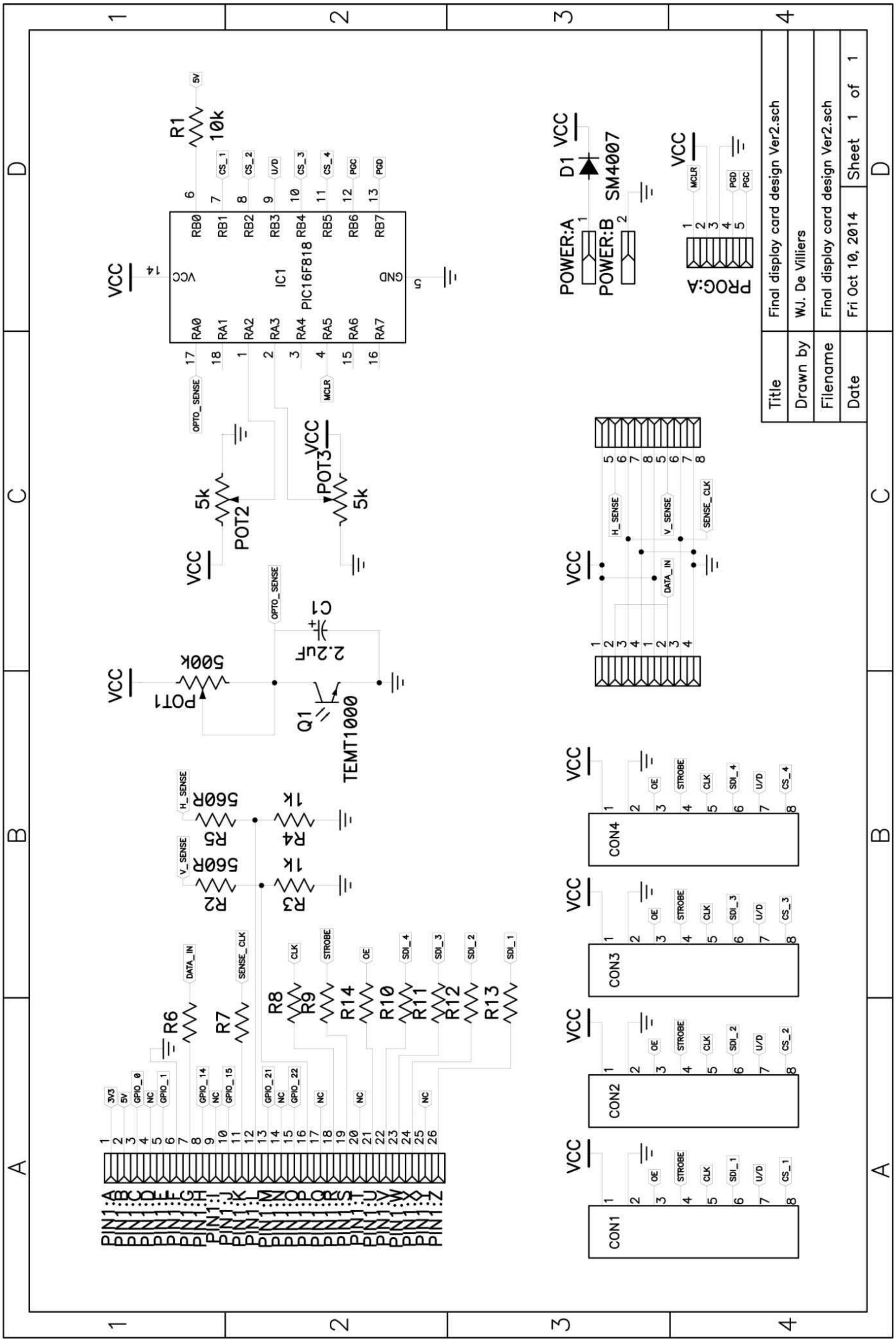


Title	Final disp design.sch		
Drawn by	WJ. De Villiers		
Filename	Final disp design.sch		
Date	Fri Oct 10, 2014	Sheet 5 of 6	



Title	Final disp design.sch
Drawn by	WJ. De Villiers
Filename	Final disp design.sch
Date	Fri Oct 10, 2014 Sheet 6 of 6

Appendix F - Display interface card schematic



Title Final display card design Ver2.sch
Drawn by WJ. De Villiers
Filename Final display card design Ver2.sch
Date Fri Oct 10, 2014
Sheet 1 of 1
IC1 PIC16F818
R1 10k
5V
VCC
GND
RB0 RB1 RB2 RB3 RB4 RB5 RB6 RB7
RA0 RA1 RA2 RA3 RA4 RA5 RA6 RA7
CS_1 CS_2 U/D CS_3 CS_4 PGC PGD
OPTO_SENSE
MCLR
D1 SM4007
POWER:A
POWER:B
PROG:A
POT1 500k
POT2 5k
POT3 5k
C1 2.2uF
Q1 TEMT1000
R2 560R
R3 1k
R4 1k
R5 560R
V_SENSE
H_SENSE
SENSE_CLK
DATA_IN
R6 R7 R8 R9 R10 R11 R12 R13 R14
CLK STROBE OE SDI_1 SDI_2 SDI_3 SDI_4
3V3 5V GPIO_0 NC GPIO_1 GPIO_14 NC GPIO_15 GPIO_21 NC GPIO_22 NC NC NC
CON1 CON2 CON3 CON4
1 2 3 4 A B C D

Appendix G - AS1116 PCB

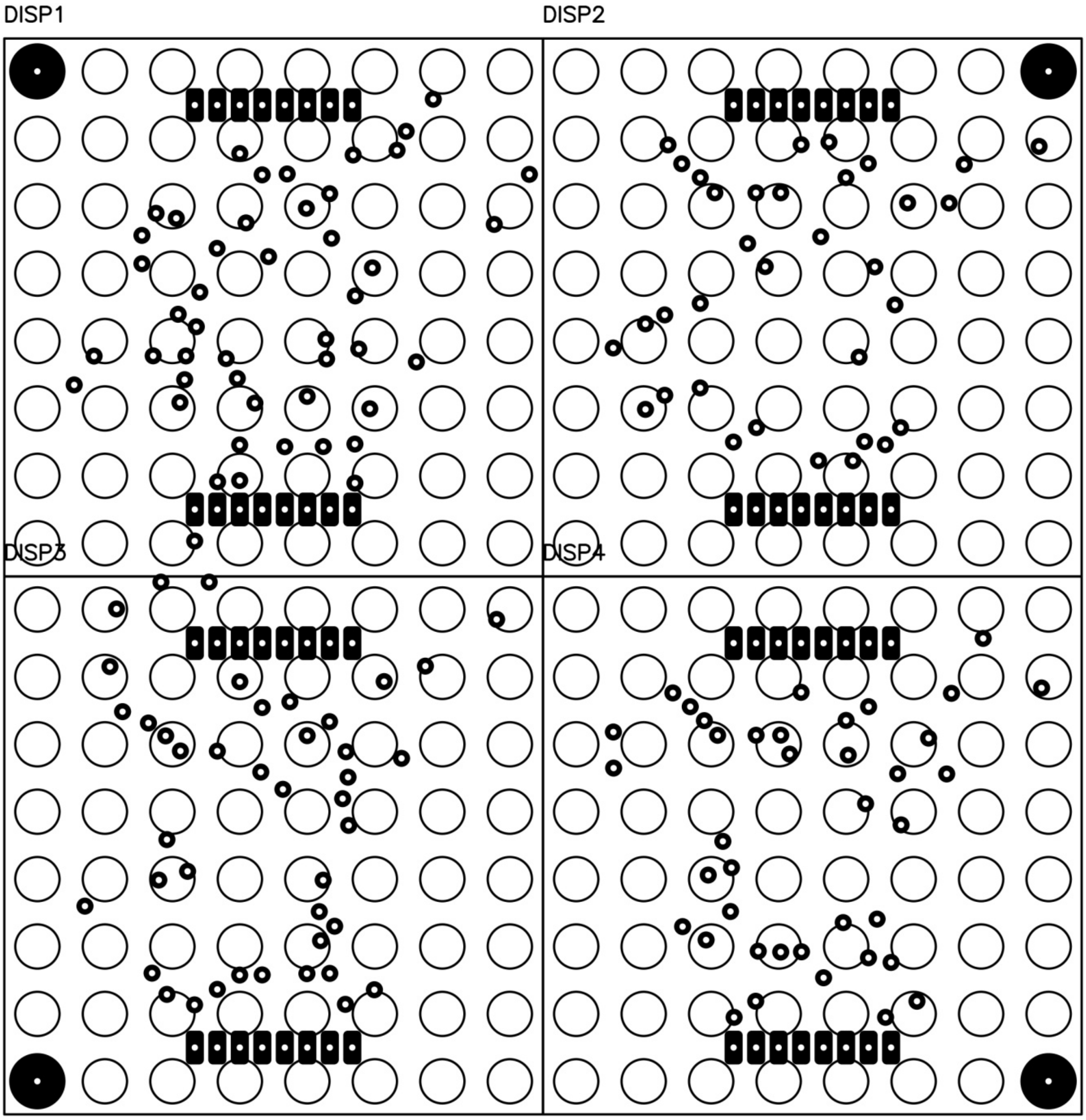


Figure G1: AS1116 Top silk

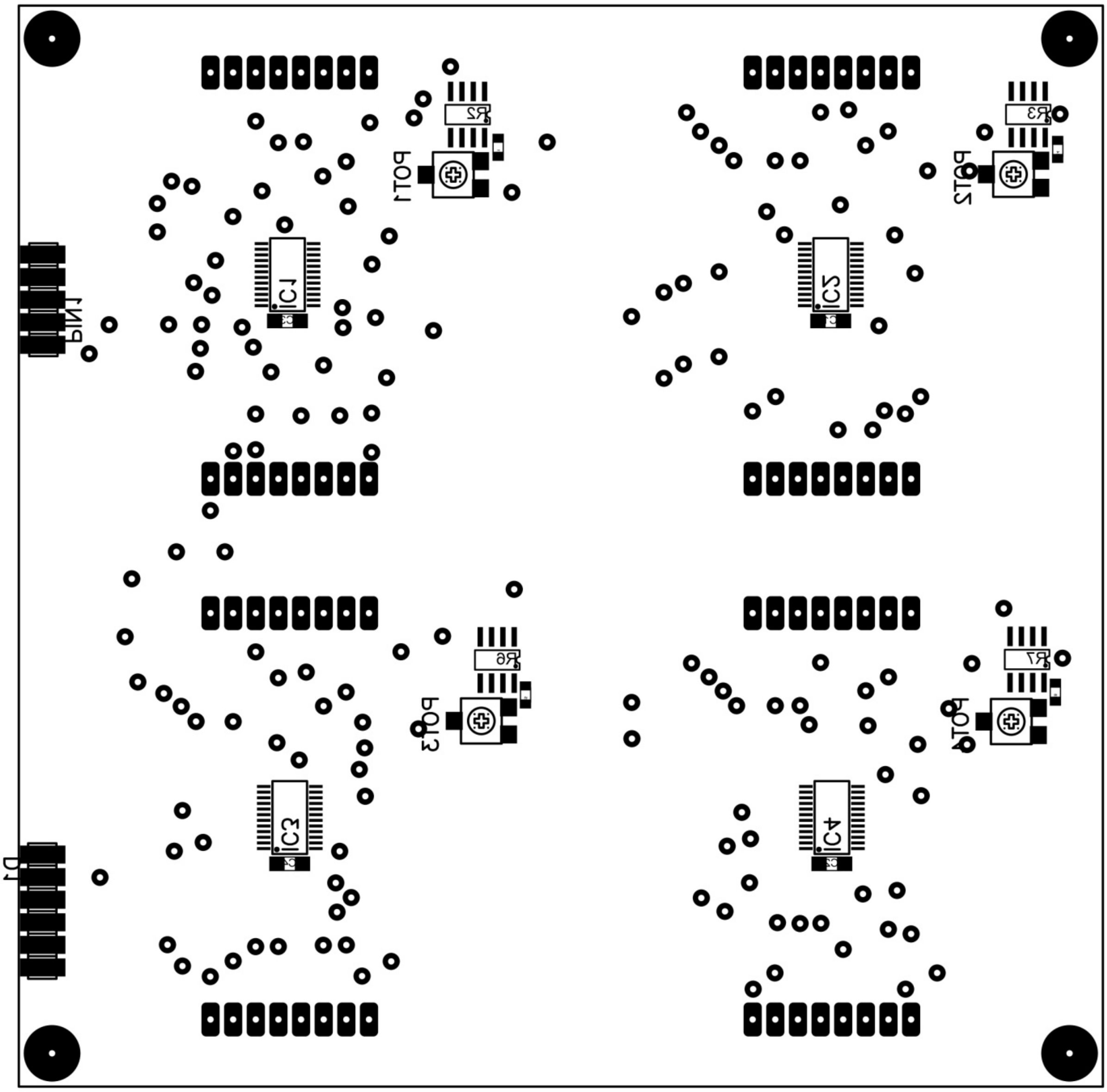


Figure G2: AS1116 Bottom silk

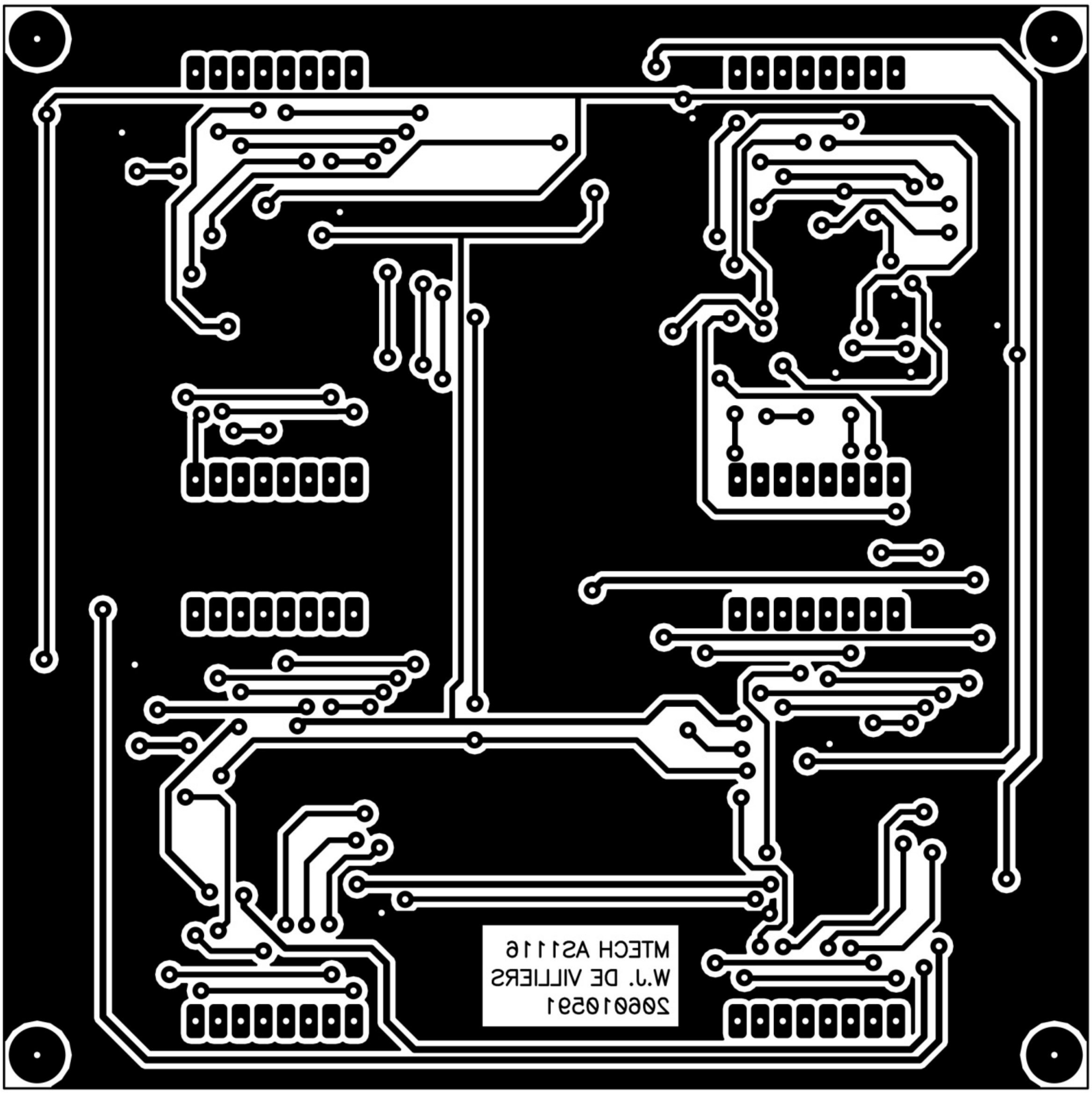


Figure G3: AS1116 Top layer

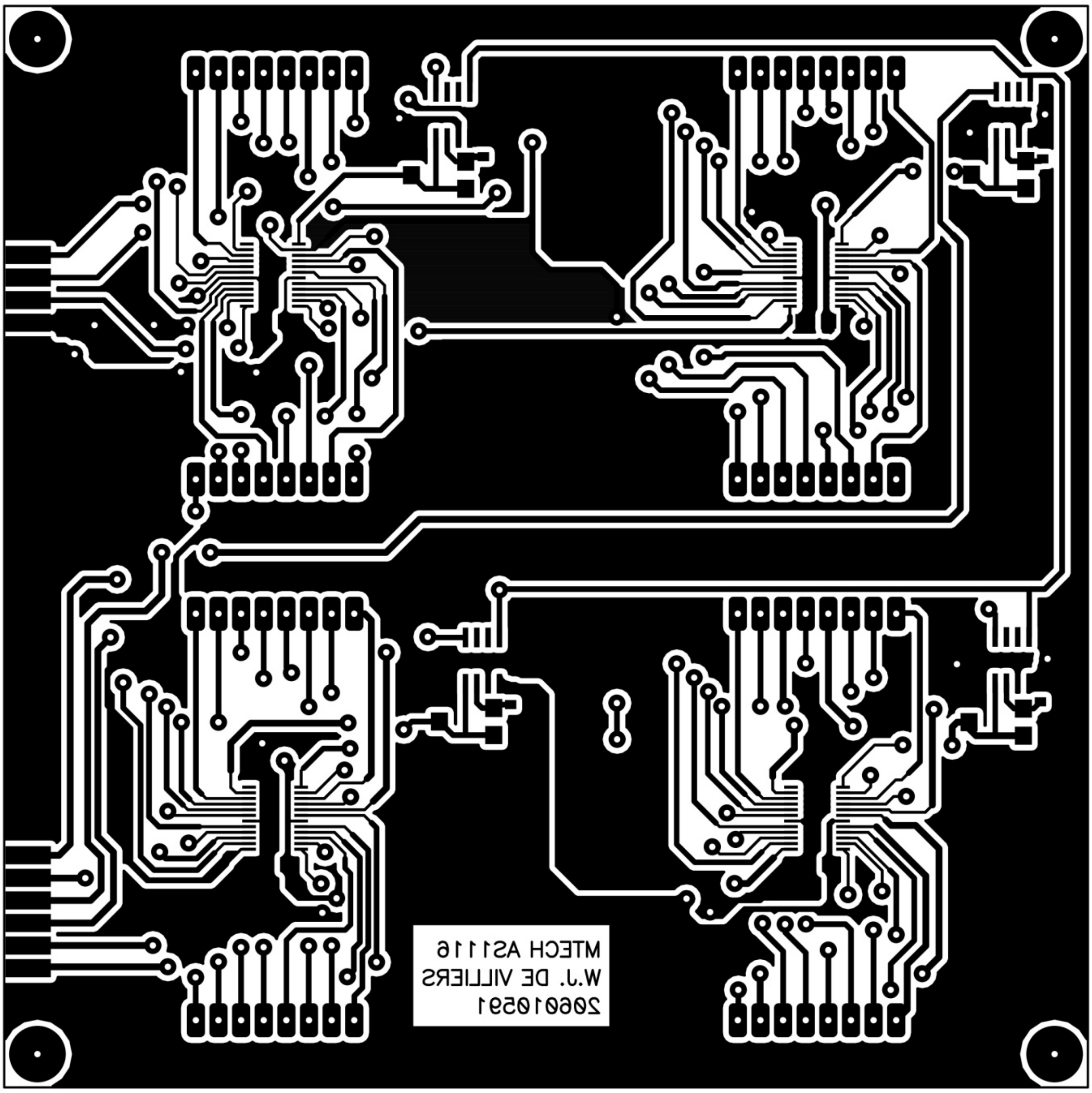


Figure G4: AS1116 Bottom layer

Appendix H - A6276 PCB

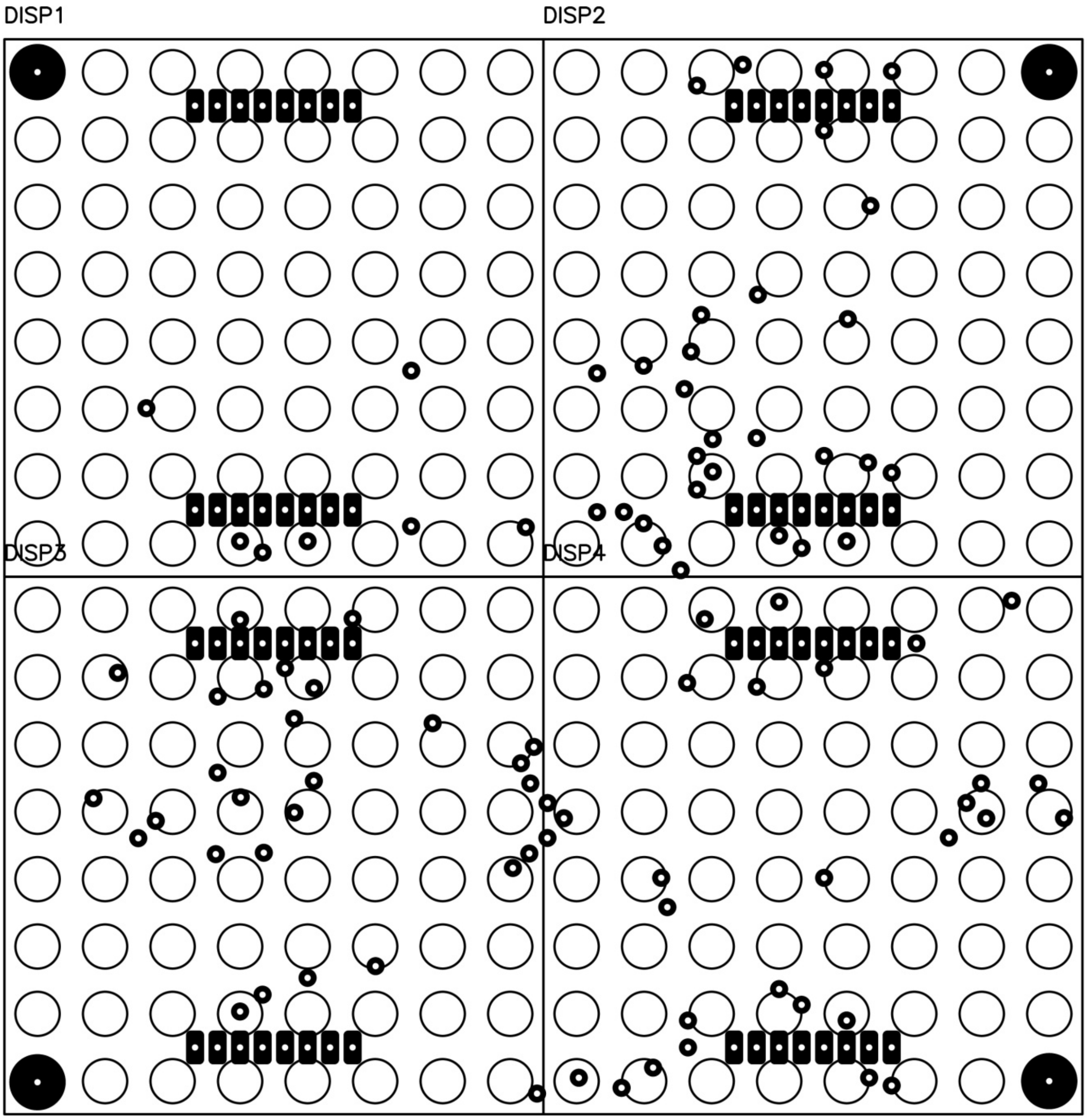


Figure H1: A6276 Top silk

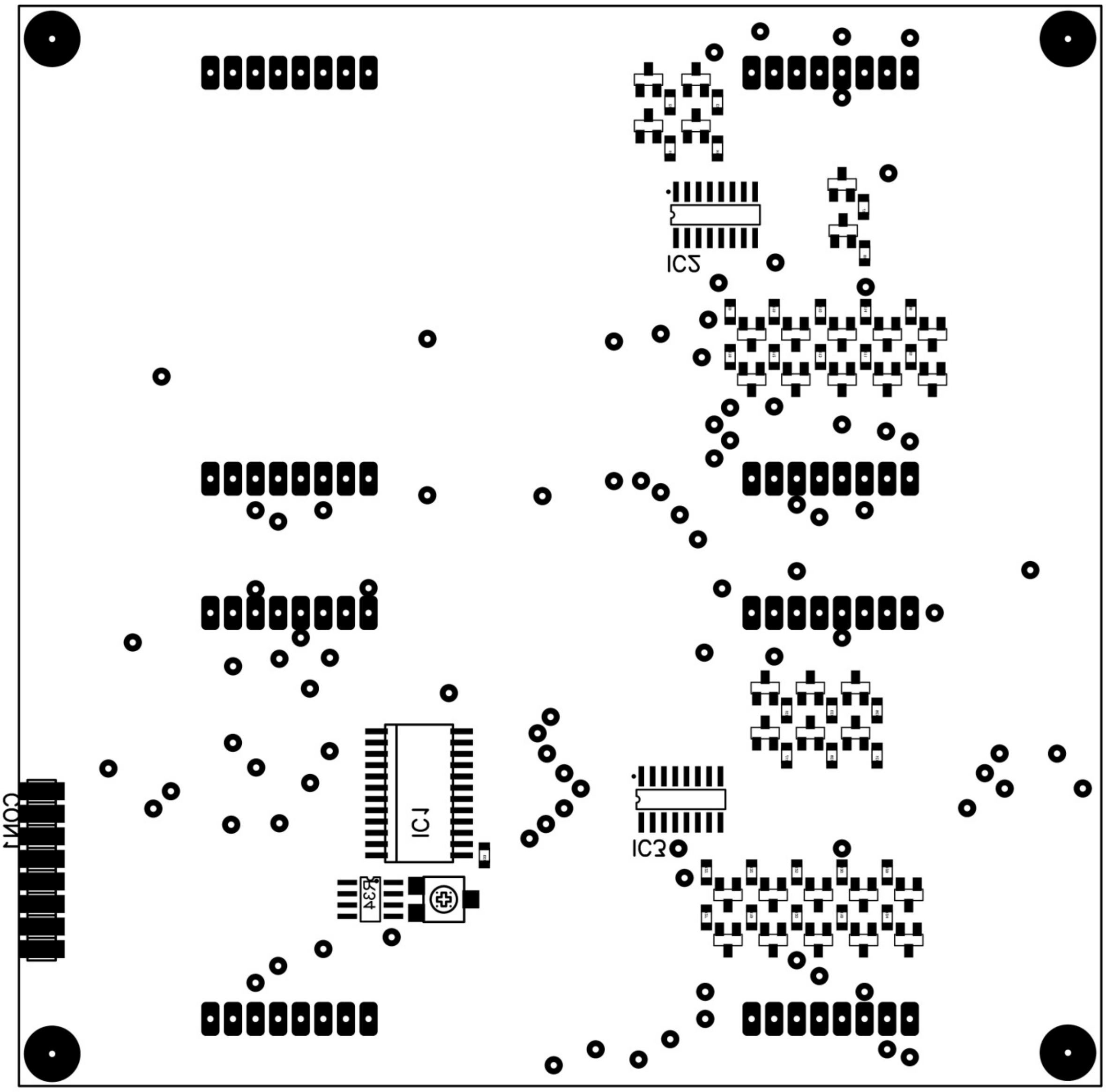


Figure H2: A6276 Bottom silk

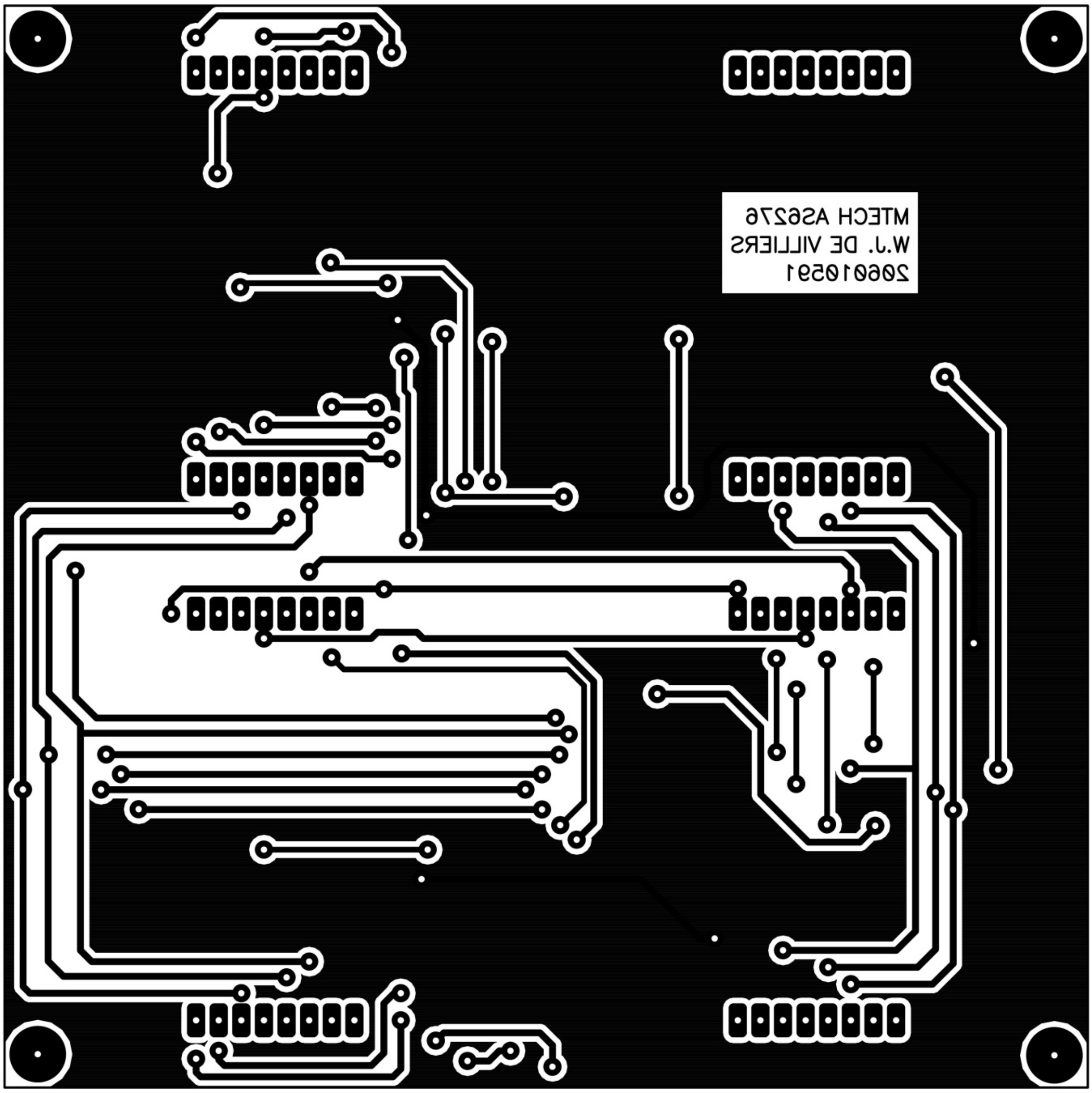


Figure H3: A6276 Top layer

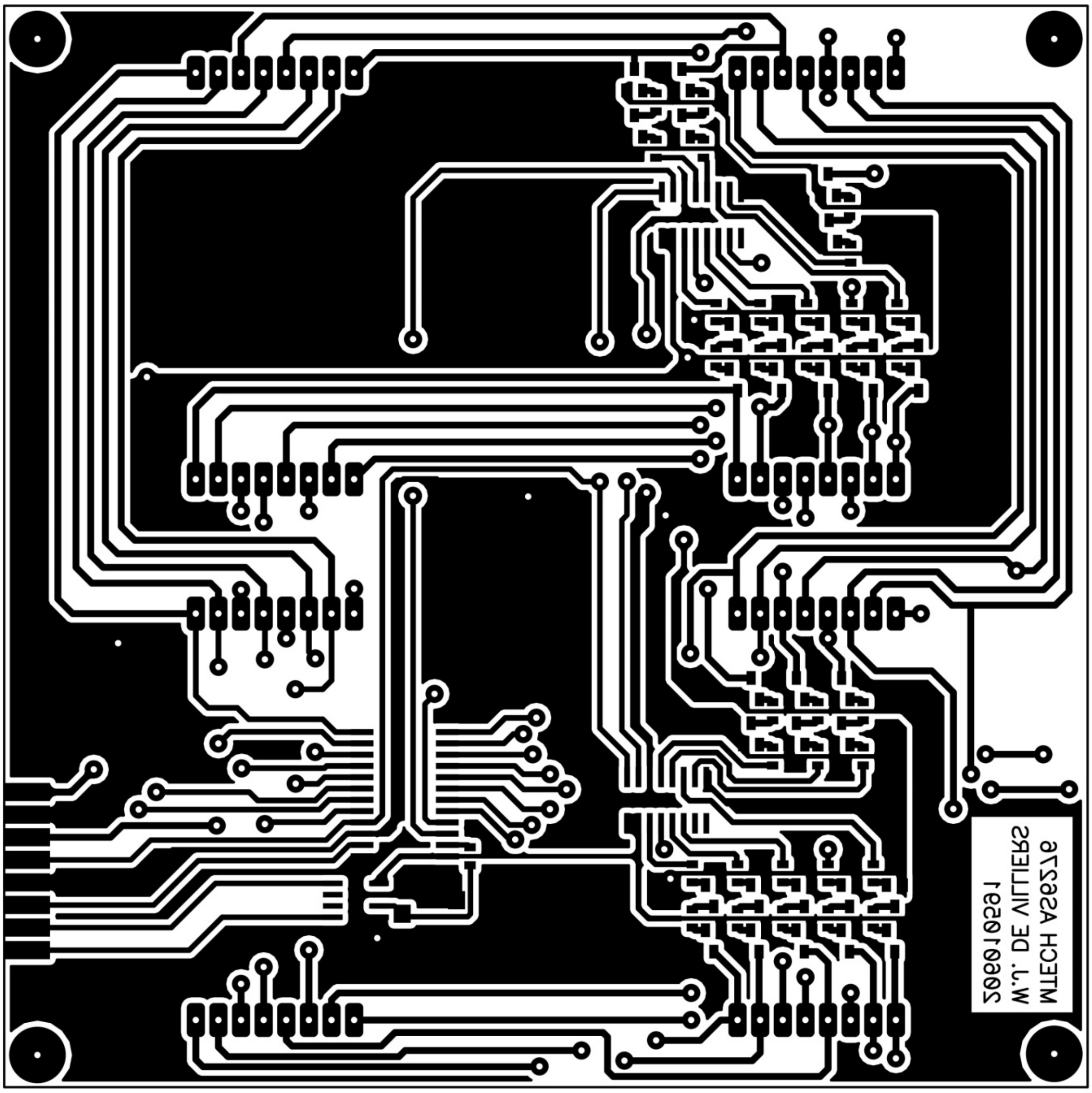


Figure H4: A6276 Bottom layer

Appendix I - Bi-polar PCB

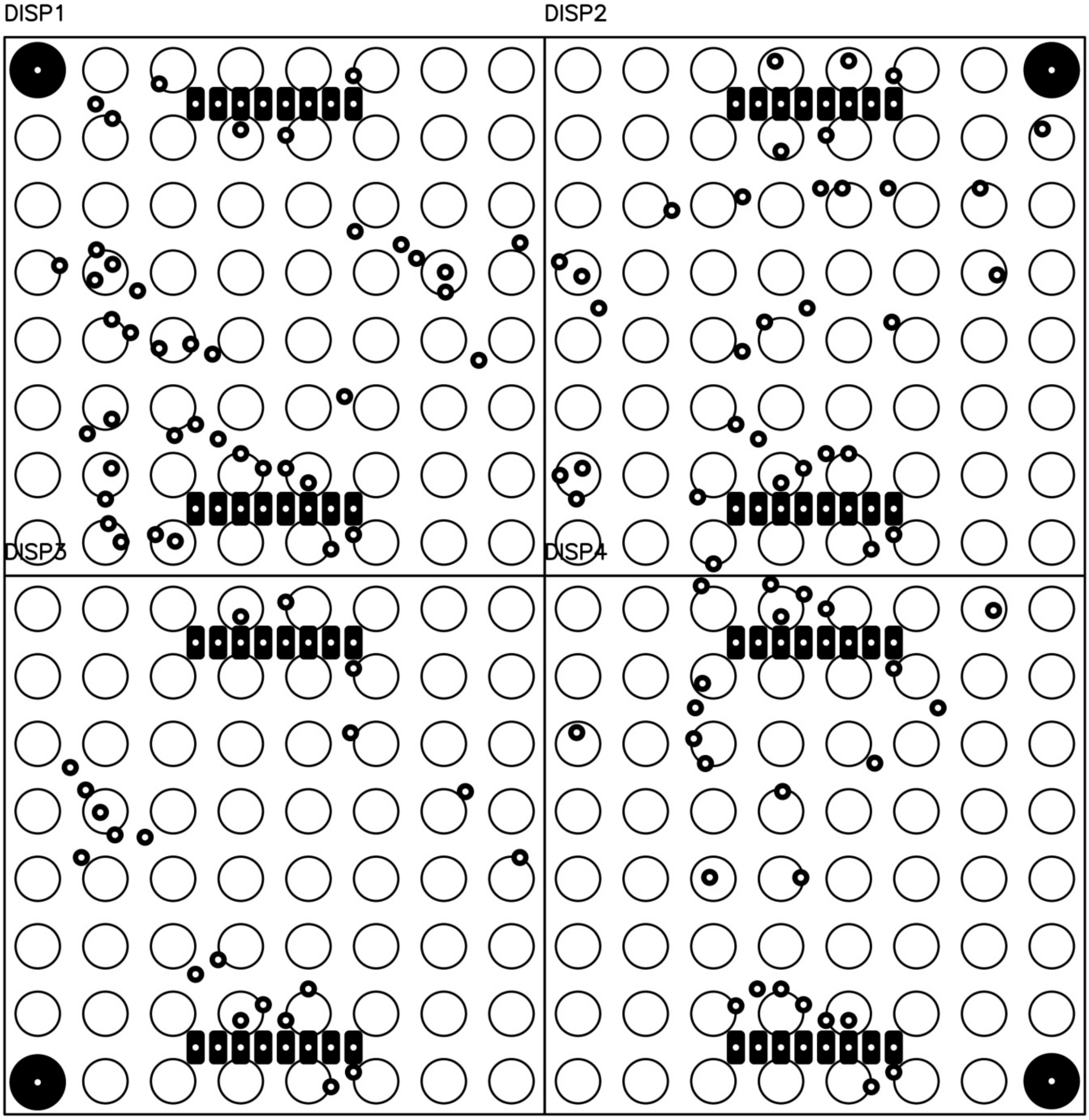


Figure I1: Bi-polar Top silk

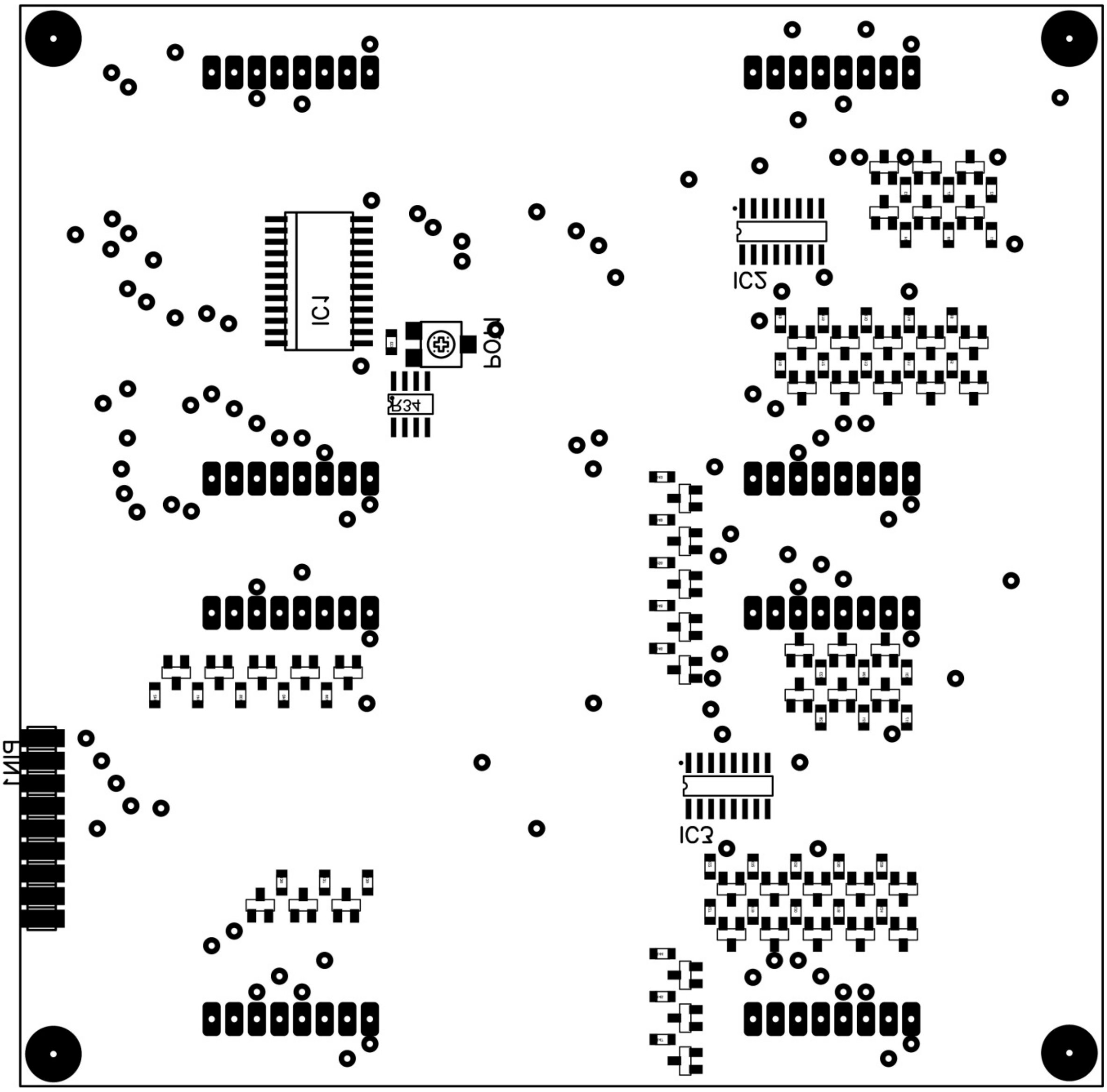


Figure 18: Bi-polar Bottom silk

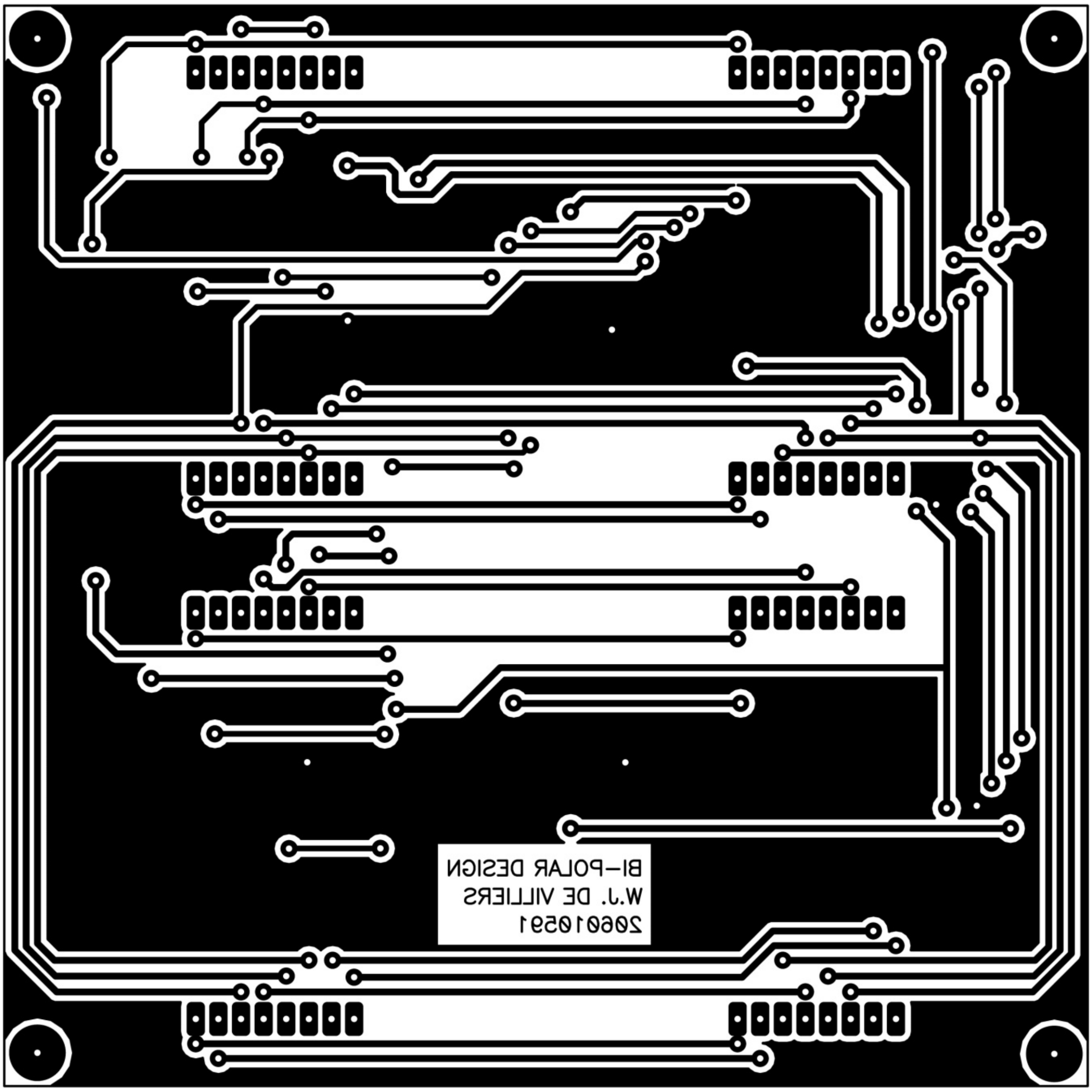


Figure I3: Bi-polar Top layer

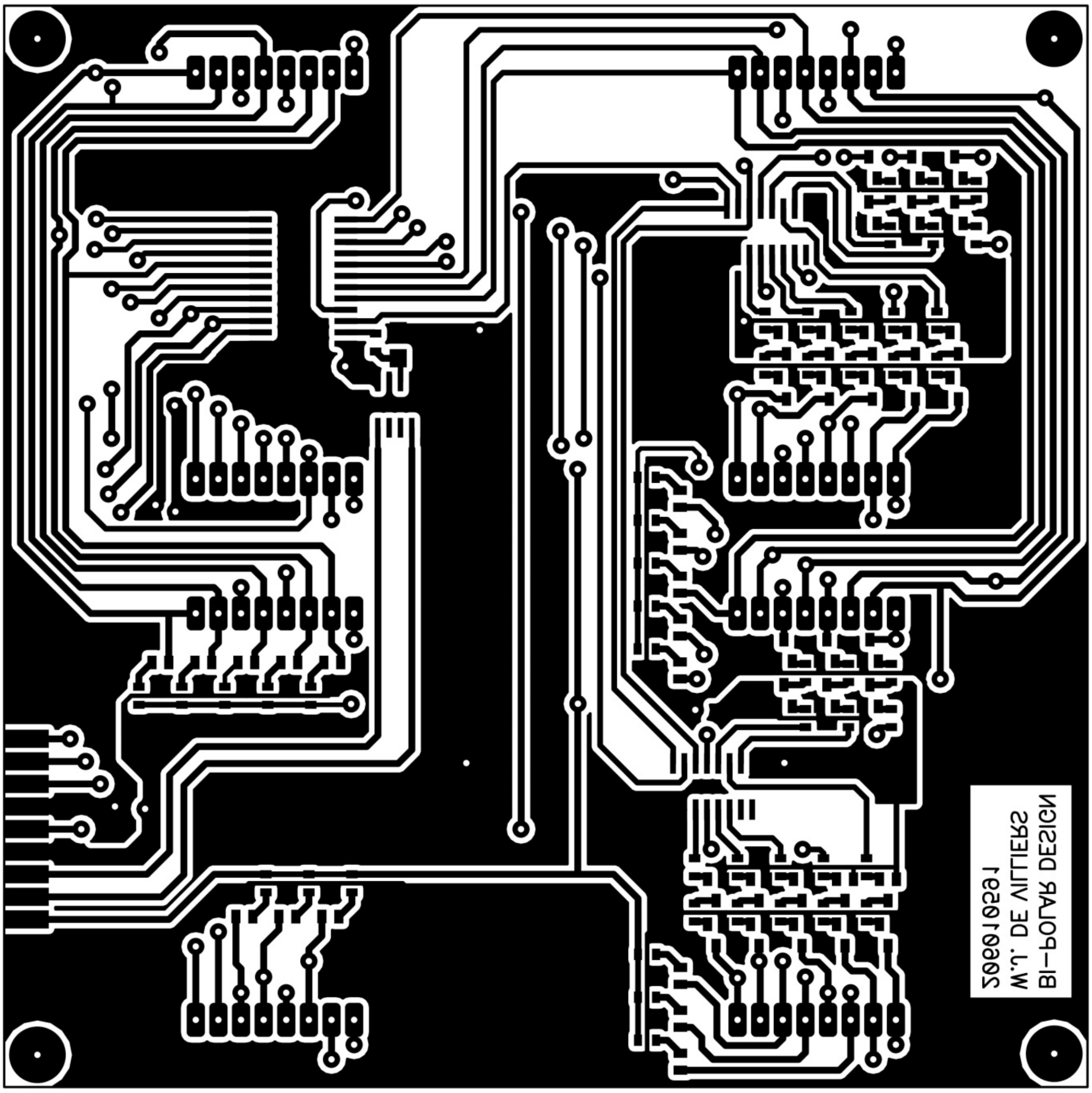


Figure I4: Bi-polar Bottom layer

Appendix J - Constant voltage PCB

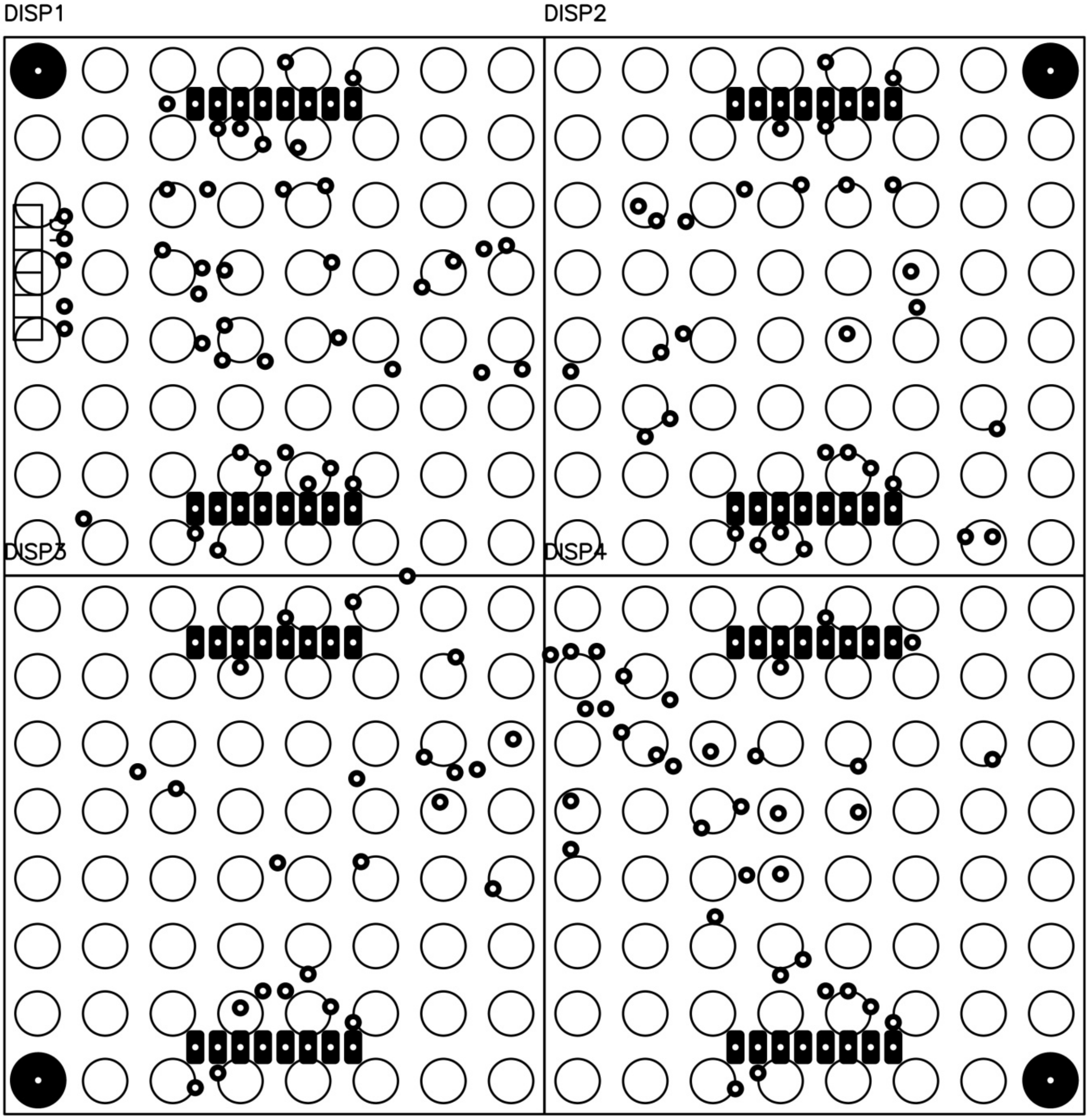


Figure J1: Constant voltage Top silk

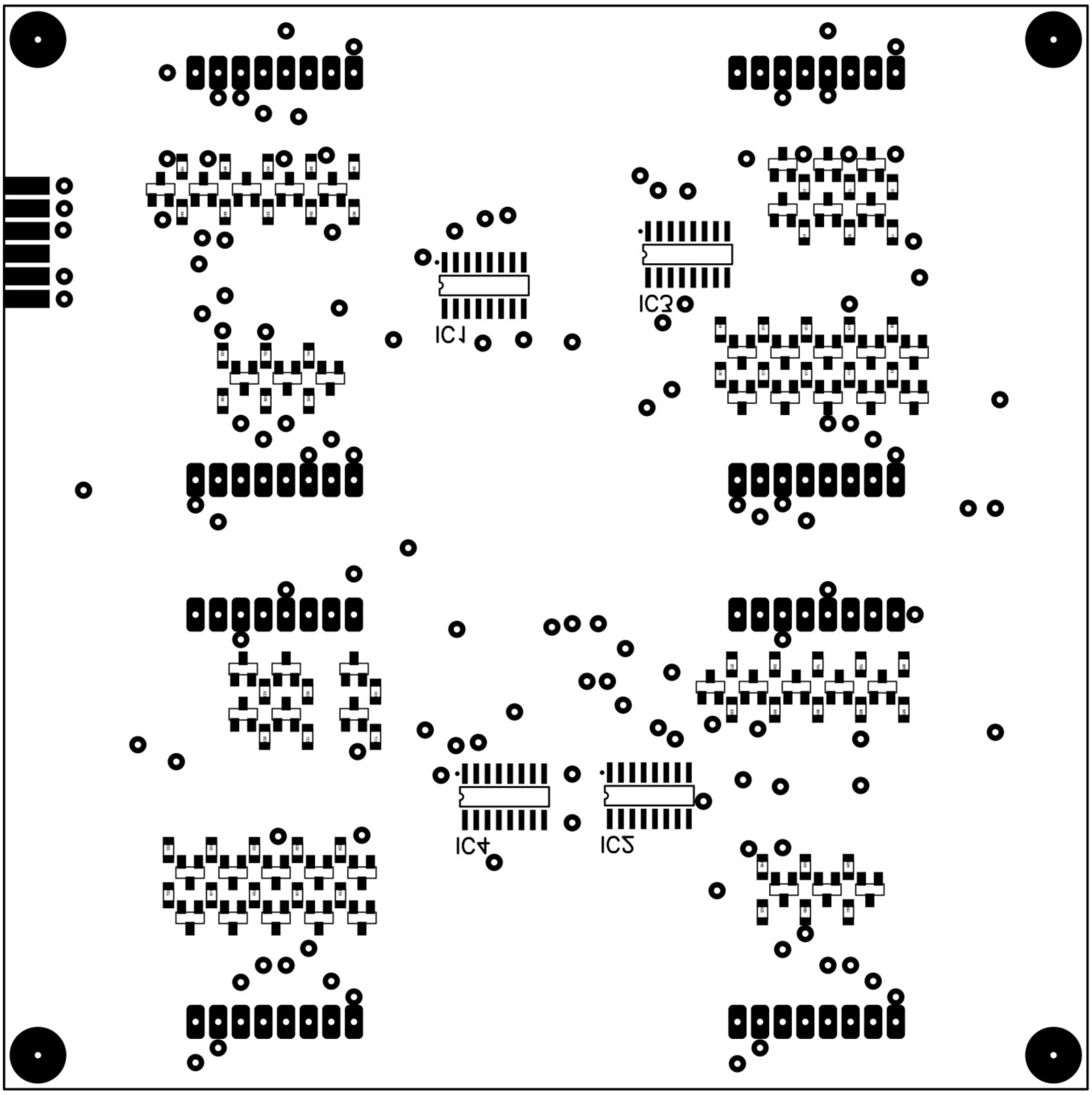


Figure J2: Constant voltage Bottom silk

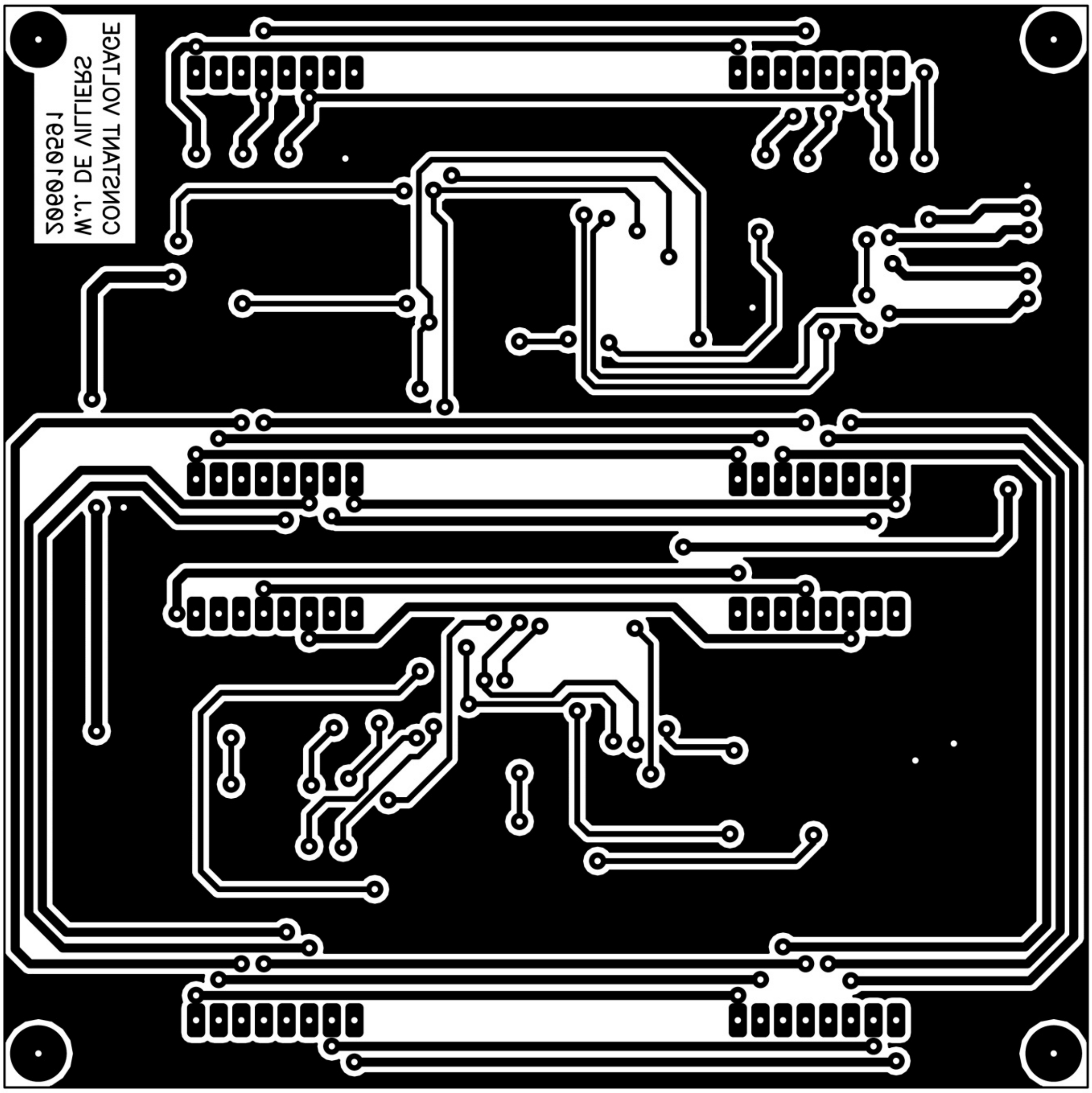


Figure J3: Constant voltage Top layer

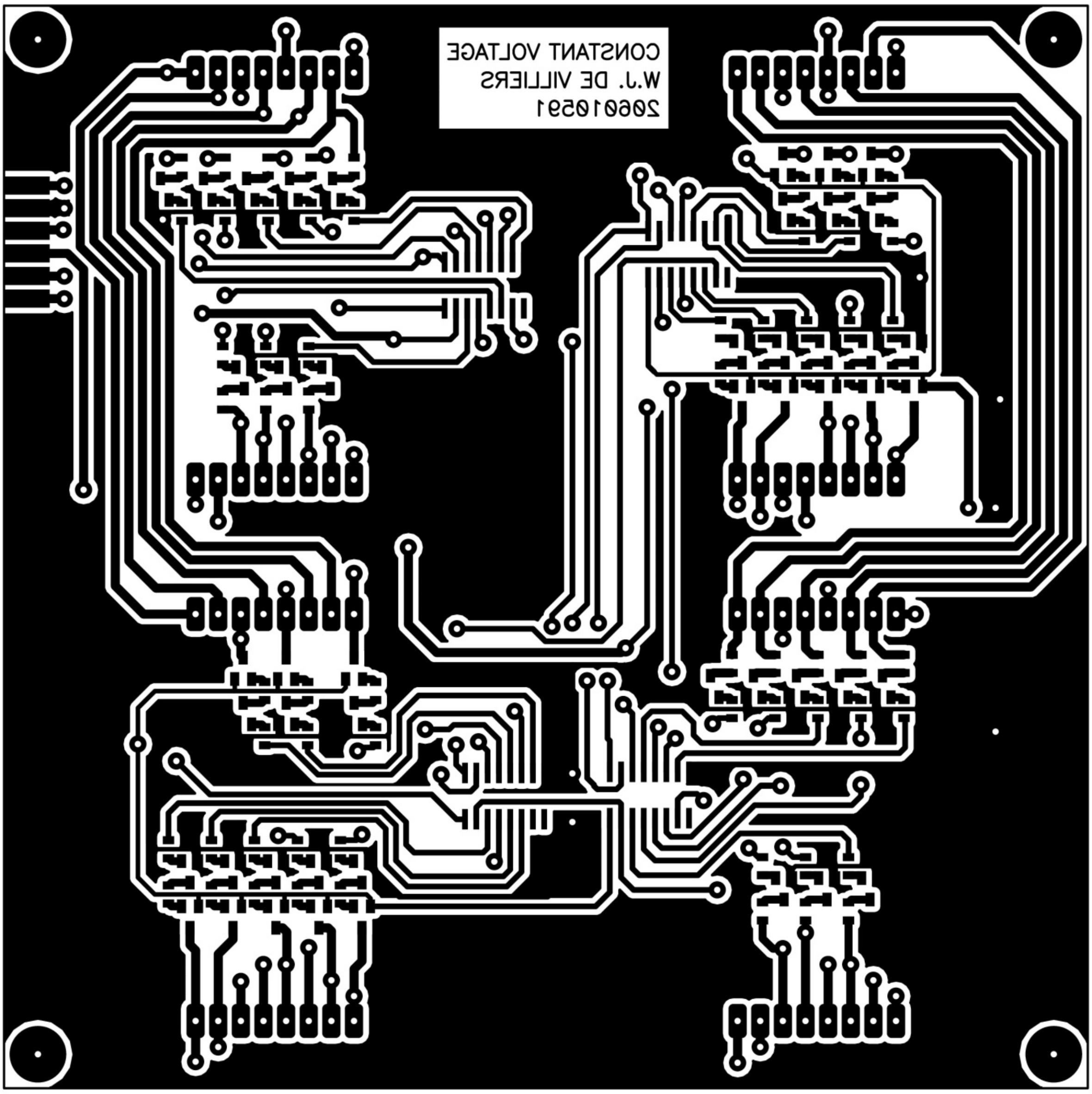


Figure J4: Constant voltage Bottom layer

Appendix K - Final display PCB

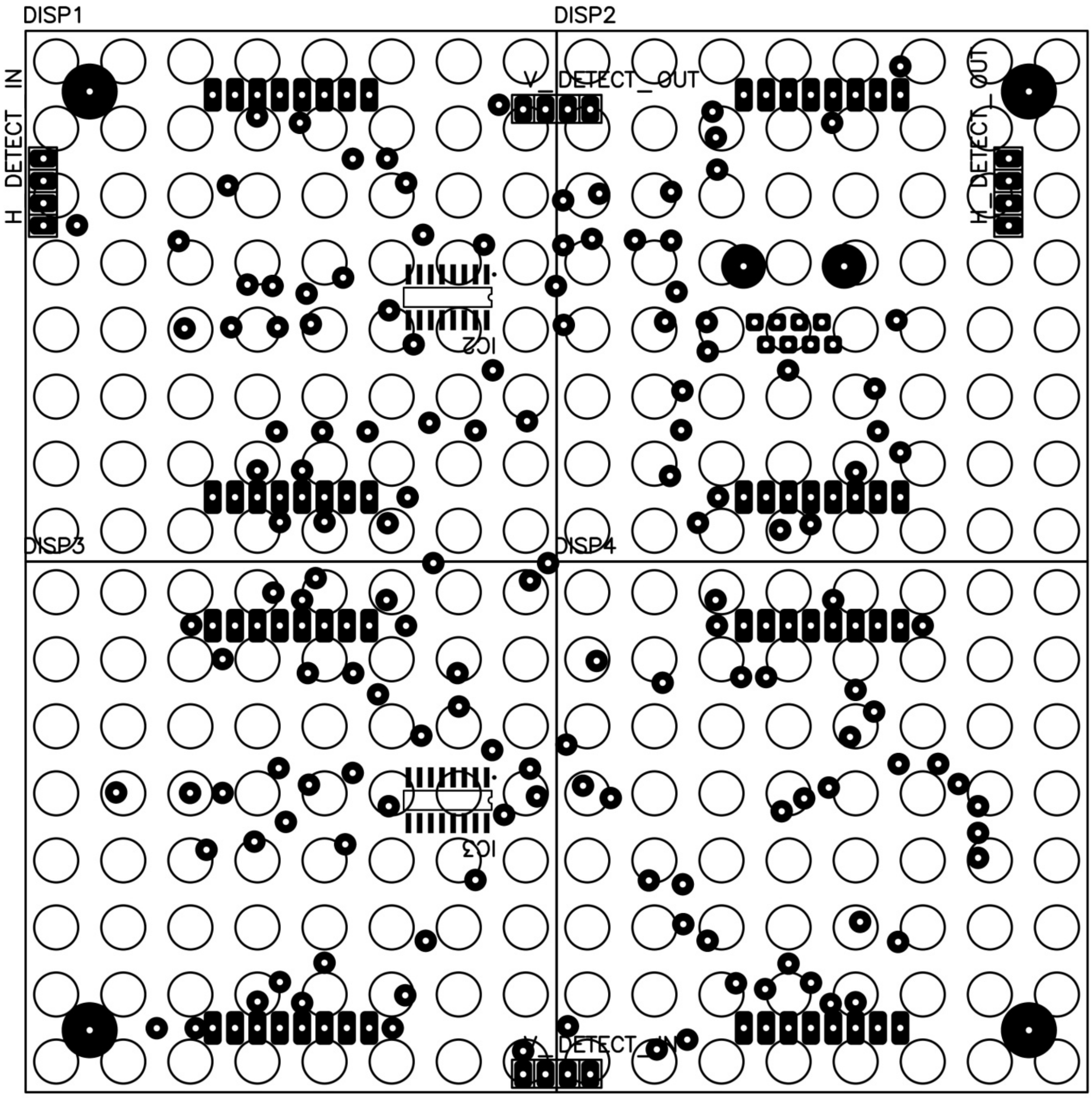


Figure K1: Final display Top silk

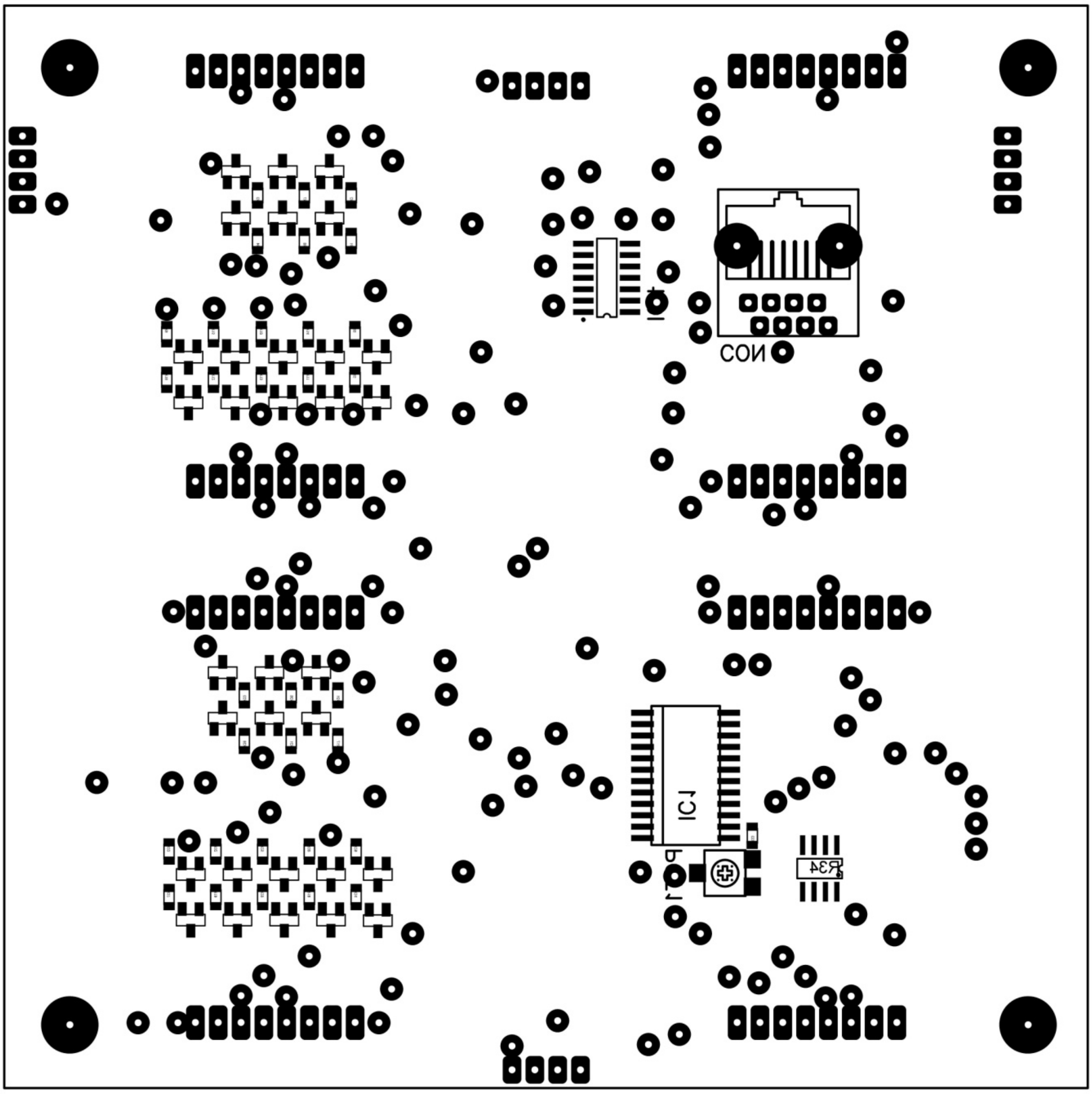


Figure K2: Final display Bottom silk

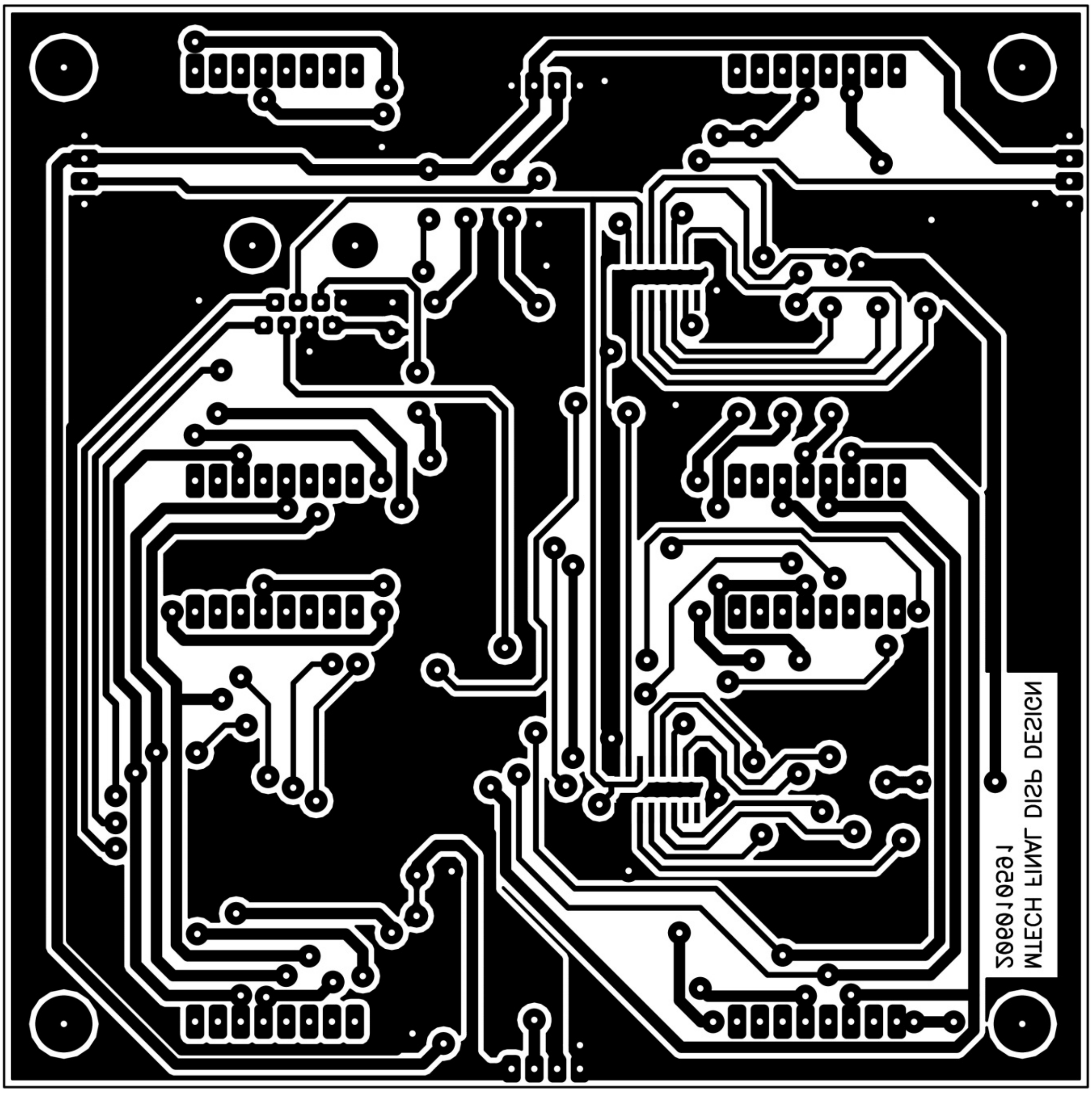


Figure K3: Final display Top layer

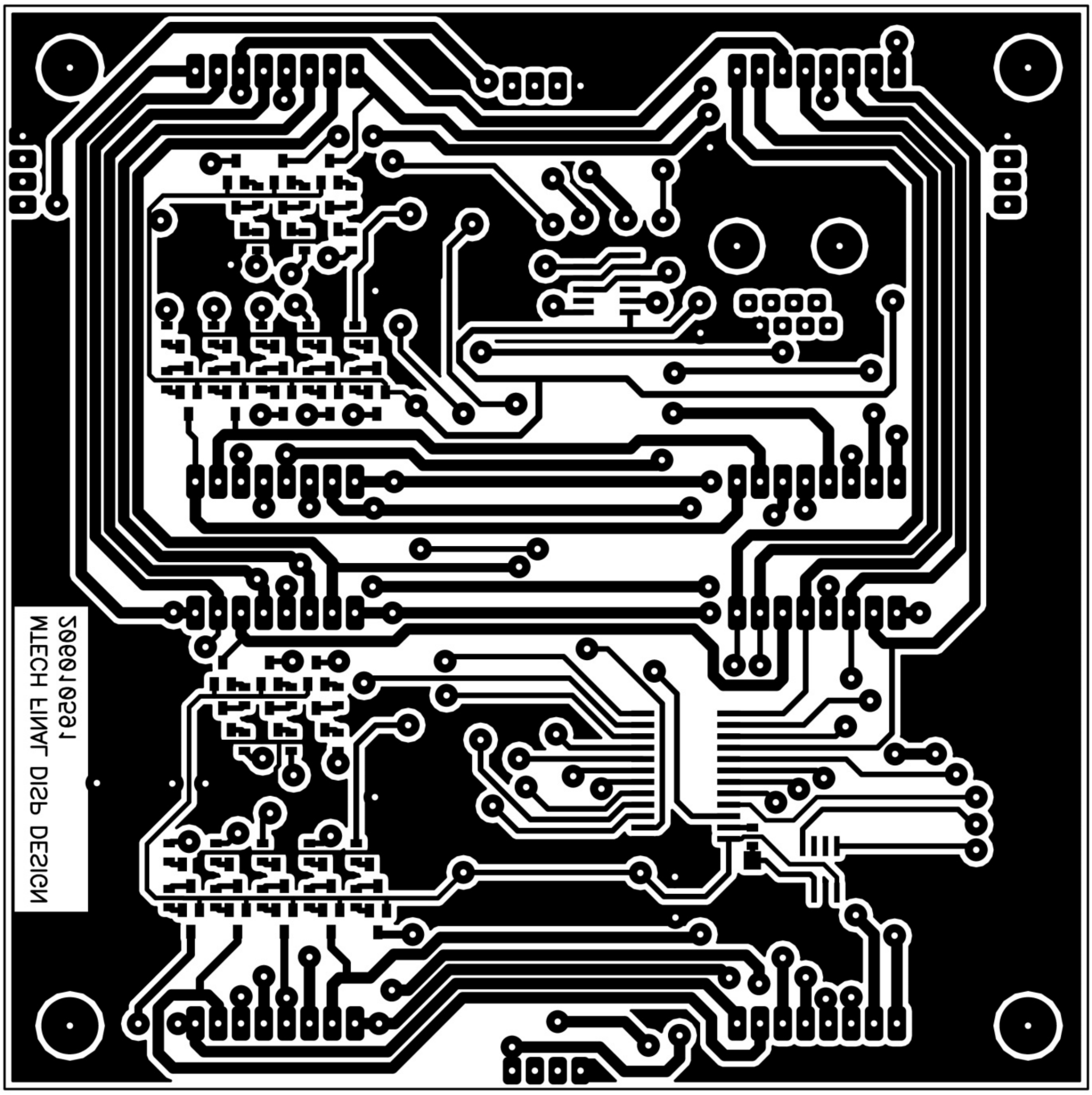


Figure K4: Final display Bottom layer

Appendix L - Display interface card PCB

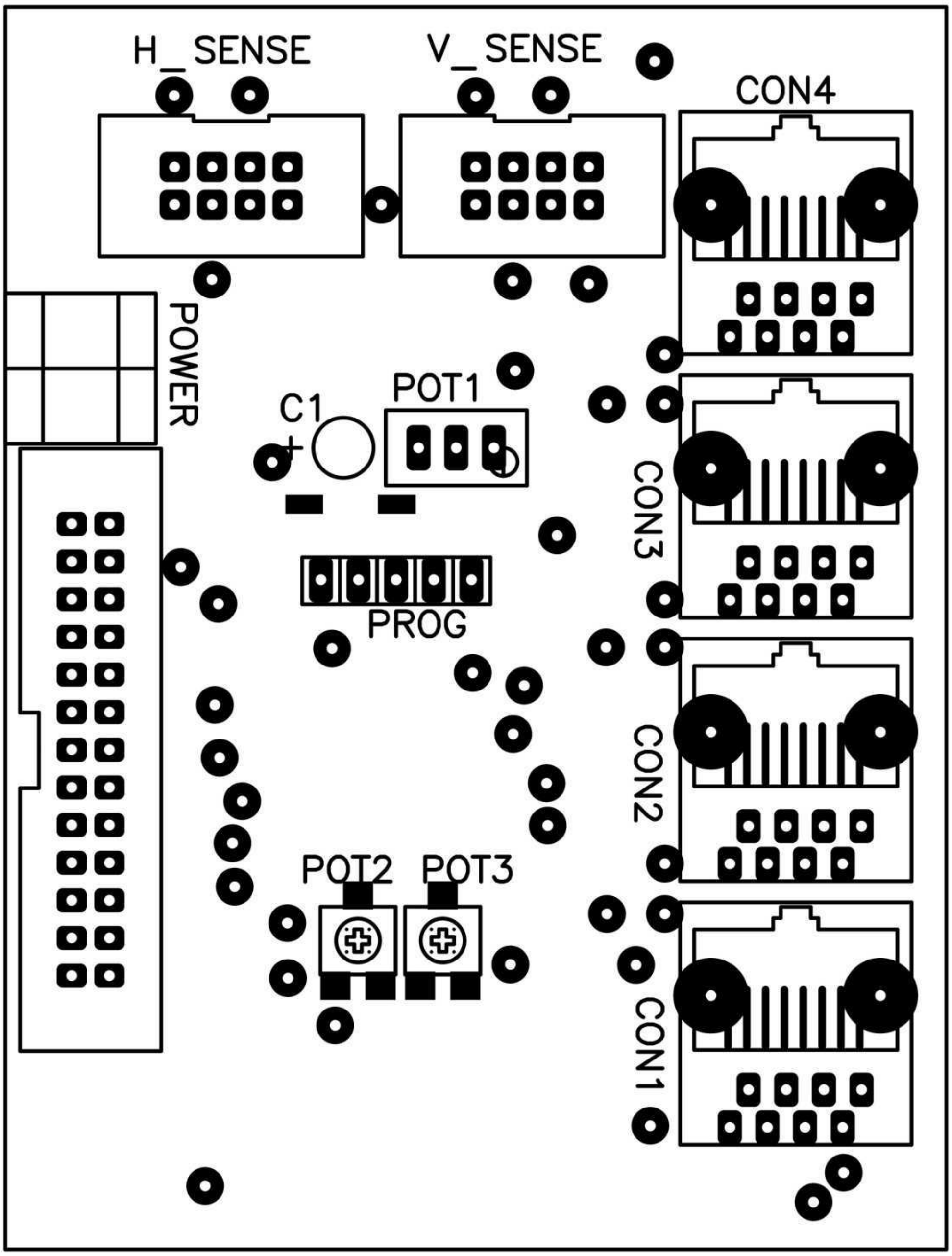


Figure L1: Display interface card Top silk

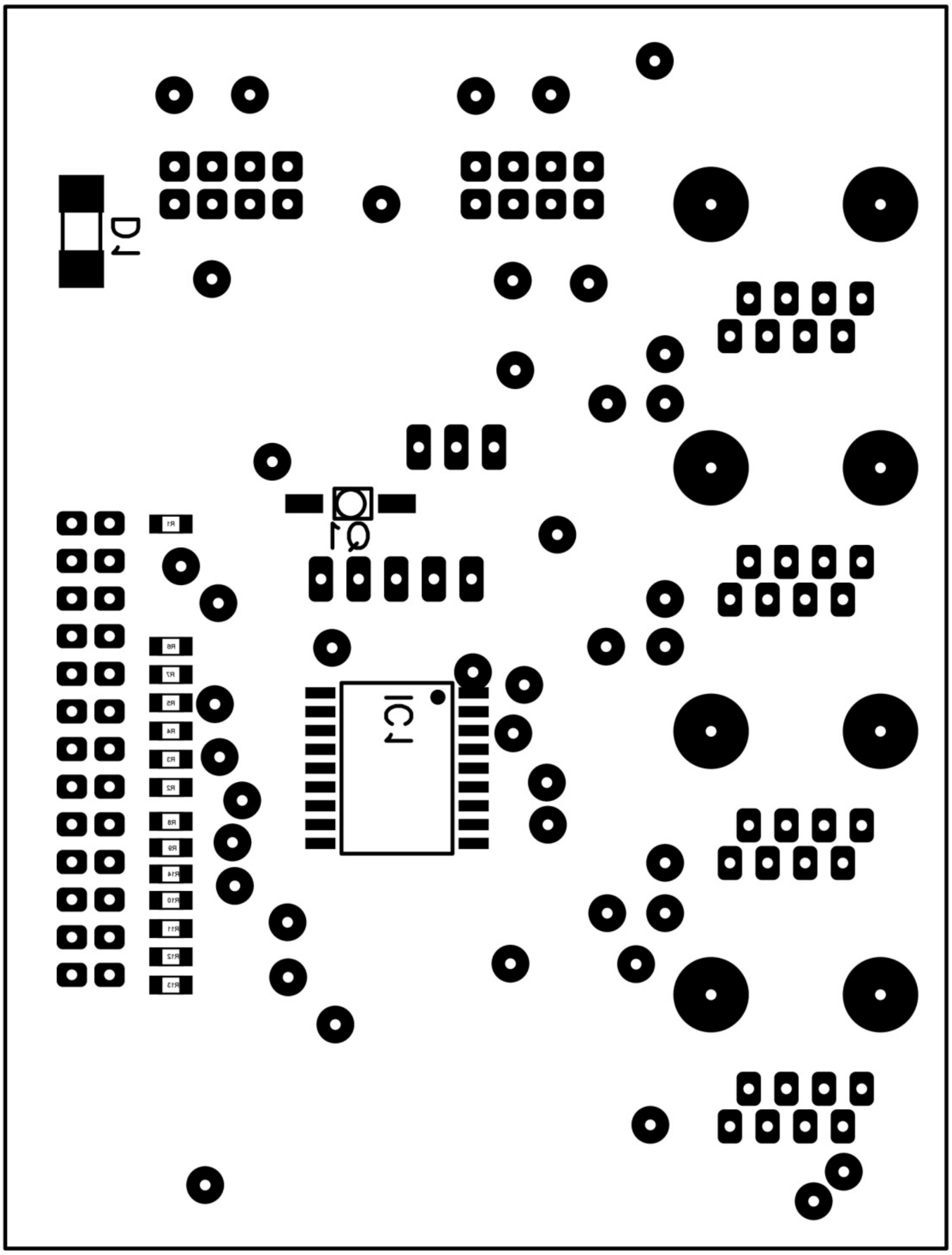


Figure L2: Display interface card Bottom silk

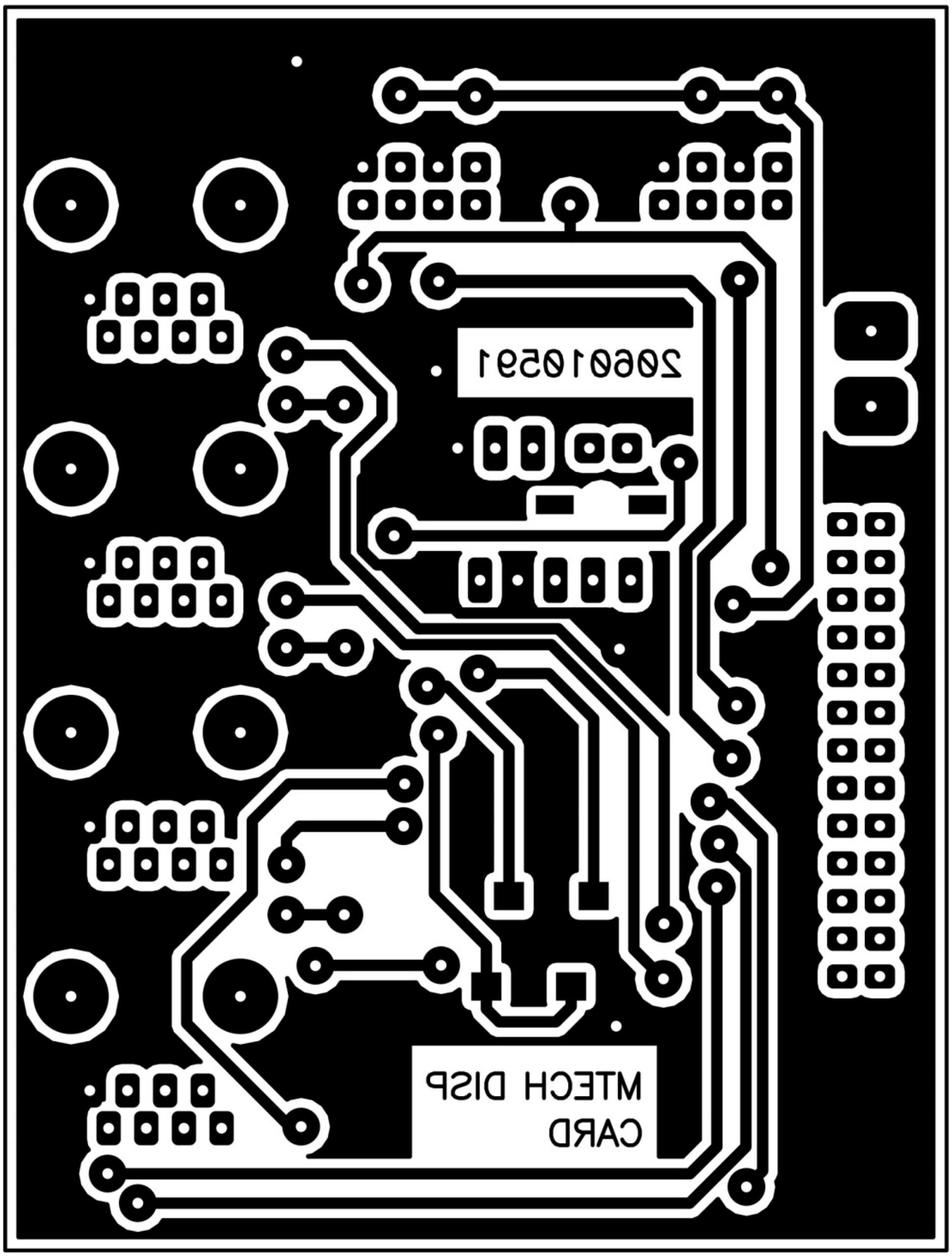


Figure L3: Display interface card Top layer

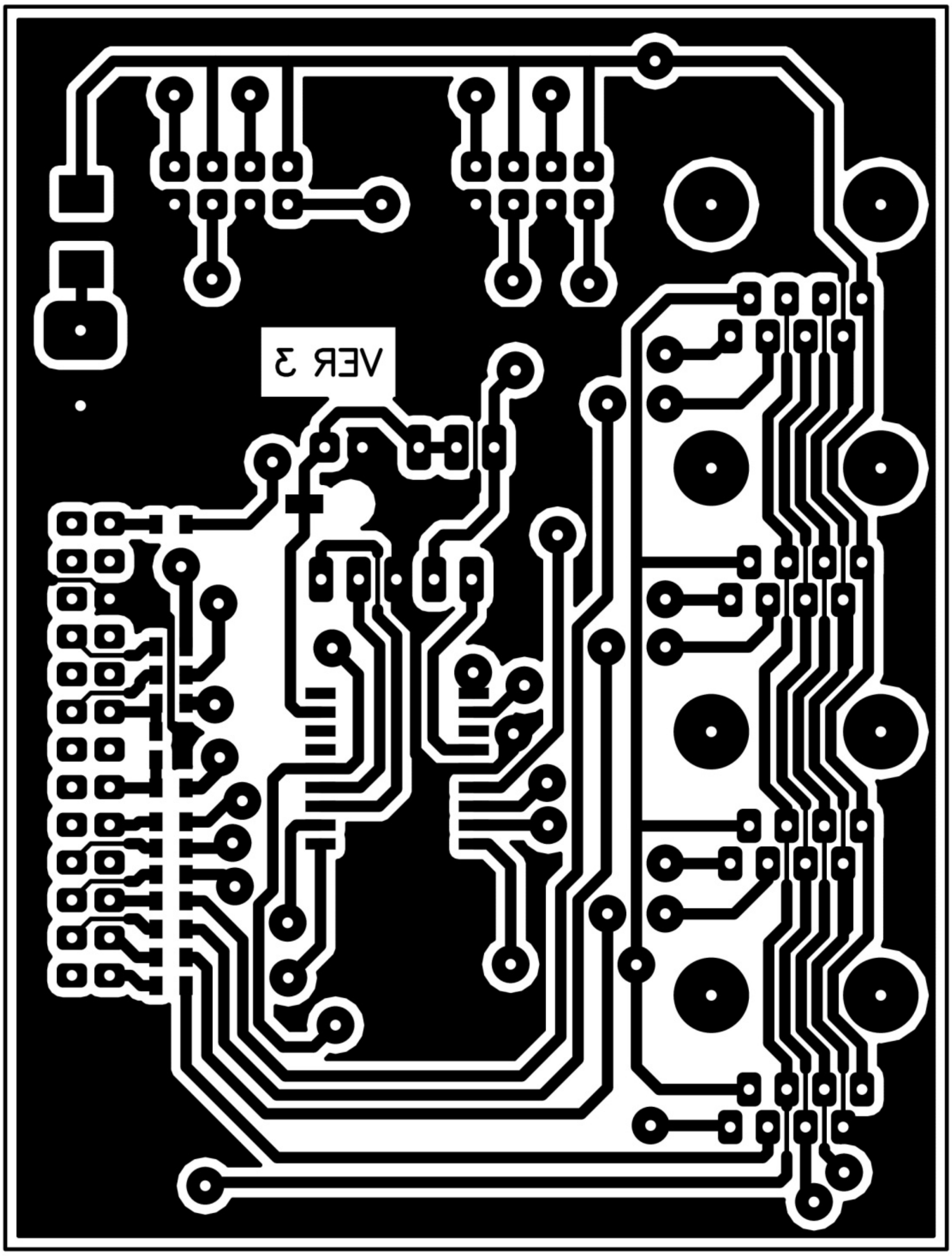


Figure L4: Display interface card Bottom layer

Appendix M - AS1116 bill of materials

Table M1: AS1116 bill of materials

<table>
<tr><th colspan="4">AS1116</th></tr>
<tr><th colspan="4">Bill of Materials</th></tr>
<tr><th>Count</th><th>Component Name</th><th>RefDes</th><th>Value</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>1</td><td>5 WAY SOCKET</td><td>PIN1</td><td></td></tr>
<tr><td>1</td><td>6 WAY SOCKET</td><td>D1</td><td></td></tr>
<tr><td>4</td><td>AS1116</td><td>IC1</td><td></td></tr>
<tr><td></td><td>AS1116</td><td>IC2</td><td></td></tr>
<tr><td></td><td>AS1116</td><td>IC3</td><td></td></tr>
<tr><td></td><td>AS1116</td><td>IC4</td><td></td></tr>
<tr><td>4</td><td>CAP 1206</td><td>C1</td><td>10uF</td></tr>
<tr><td></td><td>CAP 1206</td><td>C2</td><td>10uF</td></tr>
<tr><td></td><td>CAP 1206</td><td>C3</td><td>10uF</td></tr>
<tr><td></td><td>CAP 1206</td><td>C4</td><td>10uF</td></tr>
<tr><td>4</td><td>LED MATRIX DISPLAY 8 BY 8 CC</td><td>DISP1</td><td>TC23-11SYKWA</td></tr>
<tr><td></td><td>LED MATRIX DISPLAY 8 BY 8 CC</td><td>DISP2</td><td>TC23-11SYKWA</td></tr>
<tr><td></td><td>LED MATRIX DISPLAY 8 BY 8 CC</td><td>DISP3</td><td>TC23-11SYKWA</td></tr>
<tr><td></td><td>LED MATRIX DISPLAY 8 BY 8 CC</td><td>DISP4</td><td>TC23-11SYKWA</td></tr>
<tr><td>4</td><td>MCP4021</td><td>R 2</td><td></td></tr>
<tr><td></td><td>MCP4021</td><td>R 3</td><td></td></tr>
<tr><td></td><td>MCP4021</td><td>R 6</td><td></td></tr>
<tr><td></td><td>MCP4021</td><td>R 7</td><td></td></tr>
<tr><td>4</td><td>POT G43BT</td><td>POT1</td><td>50k</td></tr>
<tr><td></td><td>POT G43BT</td><td>POT2</td><td>50k</td></tr>
<tr><td></td><td>POT G43BT</td><td>POT3</td><td>50k</td></tr>
<tr><td></td><td>POT G43BT</td><td>POT4</td><td>50k</td></tr>
<tr><td>4</td><td>RES 0805</td><td>R 1</td><td>4k7</td></tr>
<tr><td></td><td>RES 0805</td><td>R 4</td><td>4k7</td></tr>
<tr><td></td><td>RES 0805</td><td>R 5</td><td>4k7</td></tr>
<tr><td></td><td>RES 0805</td><td>R 8</td><td>4k7</td></tr>
</table>

Appendix N - A6276 bill of materials

Table N1: A6276 bill of materials

A6276			
Bill of Materials			
Count	Component Name	RefDes	Value
1	8 WAY SOCKET	CON1	
1	A6276	IC1	
2	HEF4094BT	IC2	
	HEF4094BT	IC3	
4	LED MATRIX DISPLAY 8 BY 8 CC	DISP1	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP2	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP3	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP4	TC23-11SYKWA
1	MCP4021	R 34	
16	PBSS5320	Q15	
	PBSS5320	Q16	
	PBSS5320	Q17	
	PBSS5320	Q18	
	PBSS5320	Q19	
	PBSS5320	Q20	
	PBSS5320	Q21	
	PBSS5320	Q22	
	PBSS5320	Q23	
	PBSS5320	Q24	
	PBSS5320	Q25	
	PBSS5320	Q28	
	PBSS5320	Q29	
	PBSS5320	Q30	
	PBSS5320	Q31	
	PBSS5320	Q32	
16	PMBT2222	Q1	
	PMBT2222	Q2	
	PMBT2222	Q3	
	PMBT2222	Q4	
	PMBT2222	Q5	
	PMBT2222	Q6	
	PMBT2222	Q7	
	PMBT2222	Q8	
	PMBT2222	Q9	
	PMBT2222	Q10	
	PMBT2222	Q11	

	PMBT2222	Q12	
	PMBT2222	Q13	
	PMBT2222	Q14	
	PMBT2222	Q26	
	PMBT2222	Q27	
1	POT G43BT	POT1	50k
16	RES 0805	R 2	10k
	RES 0805	R 3	10k
	RES 0805	R 5	10k
	RES 0805	R 7	10k
	RES 0805	R 11	10k
	RES 0805	R 12	10k
	RES 0805	R 13	10k
	RES 0805	R 16	10k
	RES 0805	R 18	10k
	RES 0805	R 19	10k
	RES 0805	R 20	10k
	RES 0805	R 23	10k
	RES 0805	R 24	10k
	RES 0805	R 26	10k
	RES 0805	R 27	10k
	RES 0805	R 31	10k
16	RES 0805	R 1	100k
	RES 0805	R 4	100k
	RES 0805	R 6	100k
	RES 0805	R 8	100k
	RES 0805	R 9	100k
	RES 0805	R 10	100k
	RES 0805	R 14	100k
	RES 0805	R 15	100k
	RES 0805	R 17	100k
	RES 0805	R 21	100k
	RES 0805	R 22	100k
	RES 0805	R 25	100k
	RES 0805	R 28	100k
	RES 0805	R 29	100k
	RES 0805	R 30	100k
	RES 0805	R 32	100k
1	RES 0805	R 33	220R

Appendix O - Bi-polar bill of materials

Table O1: Bi-polar bill of materials

Bi-polar			
Bill of Materials			
Count	Component Name	RefDes	Value
1	9 WAY SOCKET	PIN1	
1	A6276	IC1	
2	HEF4094BT	IC2	
	HEF4094BT	IC3	
4	LED MATRIX DISPLAY 8 BY 8 CC	DISP1	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP2	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP3	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP4	TC23-11SYKWA
1	MCP4021	R 34	
16	PBSS5320	Q15	
	PBSS5320	Q16	
	PBSS5320	Q17	
	PBSS5320	Q18	
	PBSS5320	Q19	
	PBSS5320	Q20	
	PBSS5320	Q21	
	PBSS5320	Q22	
	PBSS5320	Q23	
	PBSS5320	Q24	
	PBSS5320	Q25	
	PBSS5320	Q28	
	PBSS5320	Q29	
	PBSS5320	Q30	
	PBSS5320	Q31	
	PBSS5320	Q32	
32	PMBT2222	Q1	
	PMBT2222	Q2	
	PMBT2222	Q3	
	PMBT2222	Q4	
	PMBT2222	Q5	
	PMBT2222	Q6	
	PMBT2222	Q7	
	PMBT2222	Q8	
	PMBT2222	Q9	
	PMBT2222	Q10	
	PMBT2222	Q11	

	PMBT2222	Q12	
	PMBT2222	Q13	
	PMBT2222	Q14	
	PMBT2222	Q26	
	PMBT2222	Q27	
	PMBT2222	Q33	
	PMBT2222	Q34	
	PMBT2222	Q35	
	PMBT2222	Q36	
	PMBT2222	Q37	
	PMBT2222	Q38	
	PMBT2222	Q39	
	PMBT2222	Q40	
	PMBT2222	Q41	
	PMBT2222	Q42	
	PMBT2222	Q43	
	PMBT2222	Q44	
	PMBT2222	Q45	
	PMBT2222	Q46	
	PMBT2222	Q47	
	PMBT2222	Q48	
1	POT G43BT	POT1	50k
16	RES 0805	R 2	10k
	RES 0805	R 3	10k
	RES 0805	R 5	10k
	RES 0805	R 7	10k
	RES 0805	R 11	10k
	RES 0805	R 12	10k
	RES 0805	R 13	10k
	RES 0805	R 16	10k
	RES 0805	R 18	10k
	RES 0805	R 19	10k
	RES 0805	R 20	10k
	RES 0805	R 23	10k
	RES 0805	R 24	10k
	RES 0805	R 26	10k
	RES 0805	R 27	10k
	RES 0805	R 31	10k
32	RES 0805	R 1	100k
	RES 0805	R 4	100k
	RES 0805	R 6	100k
	RES 0805	R 8	100k
	RES 0805	R 9	100k
	RES 0805	R 10	100k

	RES 0805	R 14	100k
	RES 0805	R 15	100k
	RES 0805	R 17	100k
	RES 0805	R 21	100k
	RES 0805	R 22	100k
	RES 0805	R 25	100k
	RES 0805	R 28	100k
	RES 0805	R 29	100k
	RES 0805	R 30	100k
	RES 0805	R 32	100k
	RES 0805	R 35	100k
	RES 0805	R 36	100k
	RES 0805	R 37	100k
	RES 0805	R 38	100k
	RES 0805	R 39	100k
	RES 0805	R 40	100k
	RES 0805	R 41	100k
	RES 0805	R 42	100k
	RES 0805	R 43	100k
	RES 0805	R 44	100k
	RES 0805	R 45	100k
	RES 0805	R 46	100k
	RES 0805	R 47	100k
	RES 0805	R 48	100k
	RES 0805	R 49	100k
	RES 0805	R 50	100k
1	RES 0805	R 33	220R

Appendix P - Constant voltage bill of materials

Table P1: Constant voltage bill of materials

Constant voltage			
Bill of Materials			
Count	Component Name	RefDes	Value
1	6 WAY SOCKET	D1	
4	HEF4094BT	IC1	
	HEF4094BT	IC2	
	HEF4094BT	IC3	
	HEF4094BT	IC4	
4	LED MATRIX DISPLAY 8 BY 8 CC	DISP1	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP2	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP3	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP4	TC23-11SYKWA
16	PBSS5320	Q15	
	PBSS5320	Q16	
	PBSS5320	Q17	
	PBSS5320	Q18	
	PBSS5320	Q19	
	PBSS5320	Q20	
	PBSS5320	Q21	
	PBSS5320	Q22	
	PBSS5320	Q23	
	PBSS5320	Q24	
	PBSS5320	Q25	
	PBSS5320	Q28	
	PBSS5320	Q29	
	PBSS5320	Q30	
	PBSS5320	Q31	
	PBSS5320	Q32	
32	PMBT2222	Q1	
	PMBT2222	Q2	
	PMBT2222	Q3	
	PMBT2222	Q4	
	PMBT2222	Q5	
	PMBT2222	Q6	
	PMBT2222	Q7	
	PMBT2222	Q8	
	PMBT2222	Q9	
	PMBT2222	Q10	
	PMBT2222	Q11	

	PMBT2222	Q12	
	PMBT2222	Q13	
	PMBT2222	Q14	
	PMBT2222	Q26	
	PMBT2222	Q27	
	PMBT2222	Q33	
	PMBT2222	Q34	
	PMBT2222	Q35	
	PMBT2222	Q36	
	PMBT2222	Q37	
	PMBT2222	Q38	
	PMBT2222	Q39	
	PMBT2222	Q40	
	PMBT2222	Q41	
	PMBT2222	Q42	
	PMBT2222	Q43	
	PMBT2222	Q44	
	PMBT2222	Q45	
	PMBT2222	Q46	
	PMBT2222	Q47	
	PMBT2222	Q48	
16	RES 0805	R 33	1k
	RES 0805	R 35	1k
	RES 0805	R 37	1k
	RES 0805	R 39	1k
	RES 0805	R 41	1k
	RES 0805	R 43	1k
	RES 0805	R 44	1k
	RES 0805	R 45	1k
	RES 0805	R 48	1k
	RES 0805	R 49	1k
	RES 0805	R 53	1k
	RES 0805	R 54	1k
	RES 0805	R 55	1k
	RES 0805	R 58	1k
	RES 0805	R 59	1k
	RES 0805	R 63	1k
16	RES 0805	R 2	10k
	RES 0805	R 3	10k
	RES 0805	R 5	10k
	RES 0805	R 7	10k
	RES 0805	R 11	10k
	RES 0805	R 12	10k
	RES 0805	R 13	10k

	RES 0805	R 16	10k
	RES 0805	R 18	10k
	RES 0805	R 19	10k
	RES 0805	R 20	10k
	RES 0805	R 23	10k
	RES 0805	R 24	10k
	RES 0805	R 26	10k
	RES 0805	R 27	10k
	RES 0805	R 31	10k
16	RES 0805	R 1	100k
	RES 0805	R 4	100k
	RES 0805	R 6	100k
	RES 0805	R 8	100k
	RES 0805	R 9	100k
	RES 0805	R 10	100k
	RES 0805	R 14	100k
	RES 0805	R 15	100k
	RES 0805	R 17	100k
	RES 0805	R 21	100k
	RES 0805	R 22	100k
	RES 0805	R 25	100k
	RES 0805	R 28	100k
	RES 0805	R 29	100k
	RES 0805	R 30	100k
	RES 0805	R 32	100k
16	RES 0805	R 34	150R
	RES 0805	R 36	150R
	RES 0805	R 38	150R
	RES 0805	R 40	150R
	RES 0805	R 42	150R
	RES 0805	R 46	150R
	RES 0805	R 47	150R
	RES 0805	R 50	150R
	RES 0805	R 51	150R
	RES 0805	R 52	150R
	RES 0805	R 56	150R
	RES 0805	R 57	150R
	RES 0805	R 60	150R
	RES 0805	R 61	150R
	RES 0805	R 62	150R
	RES 0805	R 70	150R

Appendix Q - Final display design bill of materials

Table Q1: Final display design bill of materials

Final display design			
Bill of Materials			
Count	Component Name	RefDes	Value
4	4 WAY SOCKET	H_DETECT_IN	
	4 WAY SOCKET	H_DETECT_OUT	
	4 WAY SOCKET	V_DETECT_IN	
	4 WAY SOCKET	V_DETECT_OUT	
1	A6276	IC1	
2	HEF4094BT	IC2	
	HEF4094BT	IC3	
4	LED MATRIX DISPLAY 8 BY 8 CC	DISP1	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP2	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP3	TC23-11SYKWA
	LED MATRIX DISPLAY 8 BY 8 CC	DISP4	TC23-11SYKWA
1	MC14013BDR2	IC4	
1	MCP4021	R 34	
16	PBSS5320	Q15	
	PBSS5320	Q16	
	PBSS5320	Q17	
	PBSS5320	Q18	
	PBSS5320	Q19	
	PBSS5320	Q20	
	PBSS5320	Q21	
	PBSS5320	Q22	
	PBSS5320	Q23	
	PBSS5320	Q24	
	PBSS5320	Q25	
	PBSS5320	Q28	
	PBSS5320	Q29	
	PBSS5320	Q30	
	PBSS5320	Q31	
	PBSS5320	Q32	
16	PMBT2222	Q1	
	PMBT2222	Q2	
	PMBT2222	Q3	
	PMBT2222	Q4	
	PMBT2222	Q5	
	PMBT2222	Q6	
	PMBT2222	Q7	

	PMBT2222	Q8	
	PMBT2222	Q9	
	PMBT2222	Q10	
	PMBT2222	Q11	
	PMBT2222	Q12	
	PMBT2222	Q13	
	PMBT2222	Q14	
	PMBT2222	Q26	
	PMBT2222	Q27	
1	POT G43BT	POT1	50k
16	RES 0805	R 2	10k
	RES 0805	R 3	10k
	RES 0805	R 5	10k
	RES 0805	R 7	10k
	RES 0805	R 11	10k
	RES 0805	R 12	10k
	RES 0805	R 13	10k
	RES 0805	R 16	10k
	RES 0805	R 18	10k
	RES 0805	R 19	10k
	RES 0805	R 20	10k
	RES 0805	R 23	10k
	RES 0805	R 24	10k
	RES 0805	R 26	10k
	RES 0805	R 27	10k
	RES 0805	R 31	10k
16	RES 0805	R 1	100k
	RES 0805	R 4	100k
	RES 0805	R 6	100k
	RES 0805	R 8	100k
	RES 0805	R 9	100k
	RES 0805	R 10	100k
	RES 0805	R 14	100k
	RES 0805	R 15	100k
	RES 0805	R 17	100k
	RES 0805	R 21	100k
	RES 0805	R 22	100k
	RES 0805	R 25	100k
	RES 0805	R 28	100k
	RES 0805	R 29	100k
	RES 0805	R 30	100k
	RES 0805	R 32	100k
1	RES 0805	R 33	220R
1	RJ-45 CONNECTOR VERTICAL	CON	

Appendix R - Display interface card bill of materials

Table R1: Display interface card bill of materials

Display interface card			
Bill of Materials			
Count	Component Name	RefDes	Value
1	5 WAY SOCKET	PROG	
1	CAP4MM	C1	2.2uF
1	CONNF2	POWER	
1	MULTITURN T93	POT1	500k
1	PIC16F818	IC1	
2	POT G43BT	POT2	5k
	POT G43BT	POT3	5k
11	RES 0805	R 3	1k
	RES 0805	R 4	1k
	RES 0805	R 6	1k
	RES 0805	R 7	1k
	RES 0805	R 8	1k
	RES 0805	R 9	1k
	RES 0805	R 10	1k
	RES 0805	R 11	1k
	RES 0805	R 12	1k
	RES 0805	R 13	1k
	RES 0805	R 14	1k
1	RES 0805	R 1	10k
2	RES 0805	R 2	560R
	RES 0805	R 5	560R
2	RIBBON 8P	H_SENSE	
	RIBBON 8P	V_SENSE	
1	RIBBON 26P	PIN1	
4	RJ-45 CONNECTOR VERTICAL	CON1	
	RJ-45 CONNECTOR VERTICAL	CON2	
	RJ-45 CONNECTOR VERTICAL	CON3	
	RJ-45 CONNECTOR VERTICAL	CON4	
1	SM4007	D1	
1	TEMT1000	Q1	