

# USB Powered Nixie Tube Clock Display

DD08

160020073 - Yojit Srivastava

160110033 - Anantha Hegde

16D070012 - Ishank Juneja

Faculty Advisor

Prof. Mukul Chandorkar

April, 2019



Electronic Design Lab - Spring 2019

Department of Electrical Engineering  
Indian Institute of Technology, Bombay

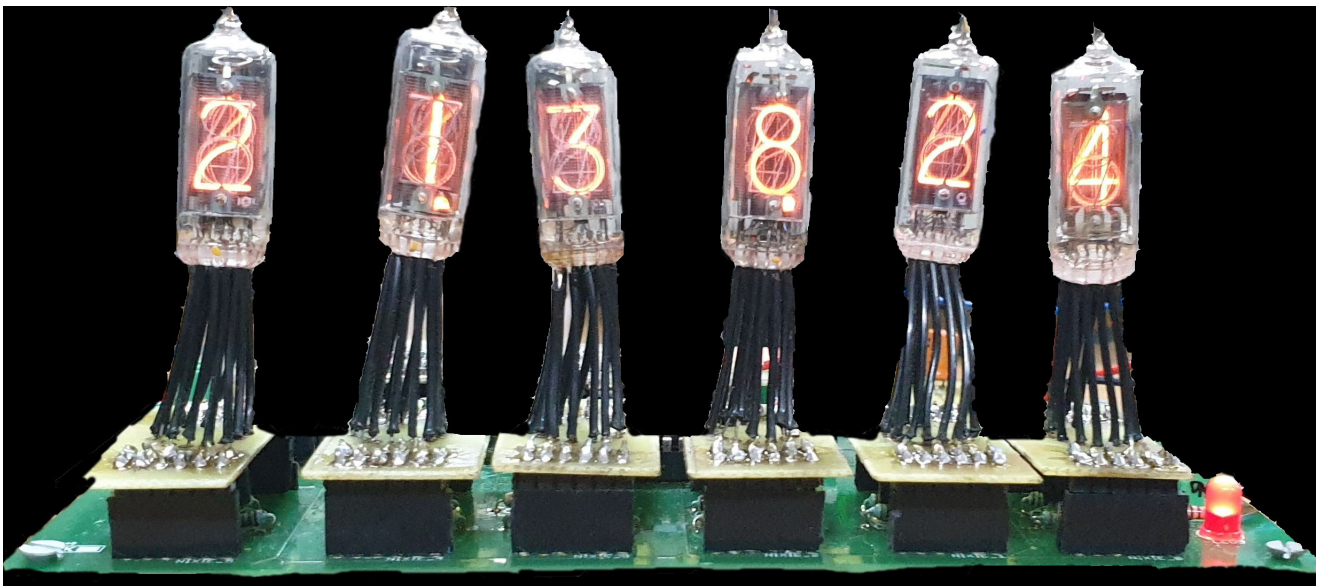
# Abstract

Nixie Tubes were a leading digit display technology during the 1970s - 80s. Nowadays, Nixie displays are admired and valued for their Retro (Classic) look.

Nixie Tube clocks have been a popular hobby project for many years now. Most of these projects use rectified AC mains as their power source since the Nixie tubes have a somewhat high operating voltage of about 170V DC.

This project is about the design of a standalone USB controlled and powered Nixie Display Clock. The time information is streamed to the clock units embedded processor via USB. The processor is then responsible for controlling the inputs to the Display front-end through control signals isolated from the High Voltage Side.

To our knowledge this project is the first attempt at designing such a Nixie display. The voltage step up from 5V to 170V in a safe and effective manner is the novelty of this project.



Nixie Tubes placed on Controller Board

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>1</b>  |
| <b>2</b> | <b>I - V Characteristics of Nixie Tube</b>                                    | <b>2</b>  |
| 2.1      | Discussion - What are Nixies ? . . . . .                                      | 2         |
| 2.2      | I - V Characteristics results . . . . .                                       | 2         |
| 2.2.1    | Forward I - V characteristics . . . . .                                       | 2         |
| 2.2.2    | Hysteresis I-V Characteristics . . . . .                                      | 5         |
| <b>3</b> | <b>Power Supply Design</b>  | <b>6</b>  |
| 3.1      | Initial Approaches . . . . .  | 6         |
| 3.2      | Existing Solutions . . . . .  | 6         |
| 3.3      | IC based Solution . . . . .   | 7         |
| 3.4      | Discrete Components Solution . . . . .  | 9         |
| 3.4.1    | Boost Converter . . . . .   | 9         |
| 3.4.2    | Flyback Converter and Transformer Design . . . . .                            | 12        |
| 3.4.3    | Some Problems . . . . .   | 13        |
| 3.4.4    | Some Solutions . . . . .  | 14        |
| 3.4.5    | Feedback . . . . .  | 14        |
| 3.4.6    | Adding a Third Winding . . . . .  | 14        |
| 3.4.7    | PCB Design . . . . .  | 15        |
| <b>4</b> | <b>Controller circuit Design</b>  | <b>17</b> |
| 4.1      | Role of the Bridge circuit . . . . .  | 17        |
| 4.2      | Design Choices for USB bridge circuit . . . . .                               | 17        |
| 4.3      | Using the FT232R to achieve Parallel out GPIO pins . . . . .                  | 17        |
| 4.4      | Bridge Circuit using the TUSB-3210 Microcontroller . . . . .                  | 18        |
| 4.4.1    | TUSB Power Supply . . . . .   | 18        |
| 4.4.2    | Micro-controller Schematic . . . . .  | 19        |
| 4.4.3    | GPIO Interfacing . . . . .  | 20        |
| <b>5</b> | <b>Front-End Driver Circuit</b>   | <b>22</b> |
| 5.1      | Description . . . . .   | 22        |
| 5.2      | Tube Control . . . . .  | 22        |
| 5.3      | Digit Control . . . . .   | 22        |
| 5.4      | Complete Schematic and Board View of the Controller plus Driver Board . . . . | 24        |

|          |   |           |
|----------|---|-----------|
| <b>6</b> | <b>Acknowledgements and References</b>            | <b>26</b> |
| 6.1      | General References . . . . .                      | 26        |
| 6.2      | References for Bridge circuit Design . . . . .    | 26        |
| 6.3      | References for Driver Circuit . . . . .           | 26        |
| 6.4      | References for Flyback Converter Design . . . . . | 27        |

# 1. Introduction

What makes our project unique is that it succeeds in powering up the Nixie Clock Display using only a **single 5V USB power supply**.

The limitation of using USB as our power source lead to another challenge; the power available through the host Single Board Computers USB port is not enough to light up numerous Nixie Tube units simultaneously. Our solution takes advantage of the persistence of human vision by rapidly cycling through the tubes, thus implementing a low powered Multiplexed Display. To implement all of these tasks, there are primarily two Hardware modules in our project The **Isolated Flyback converter**, an **SMPS topology** and a combined **Display and Control Unit**. Our project implements both of these modules as separate Printed Circuit Boards.

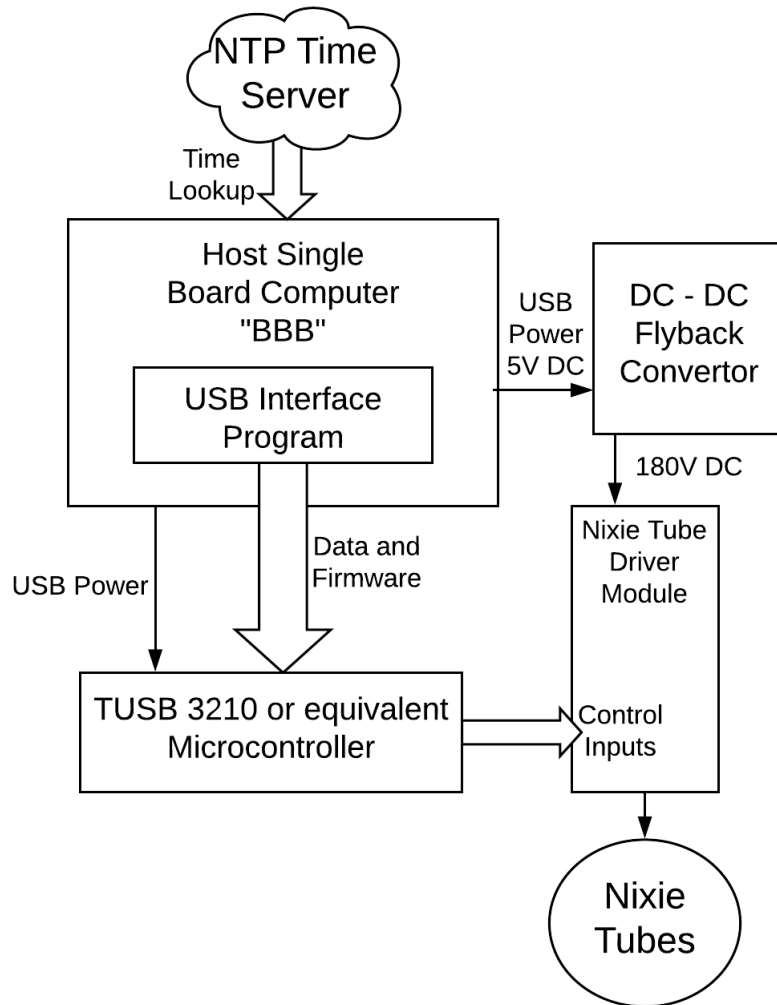


Figure 1.1: System Level Block Diagram

## 2. I - V Characteristics of Nixie Tube

### 2.1 Discussion - What are Nixies ?

Nixie tubes are Cold Cathode display tubes that illuminate when a high voltage is applied across them. The source of the light is electronic excitation and de-excitation, similar to the more modern, Neon Display tubes. For more information on their working, the reader may refer to [1].

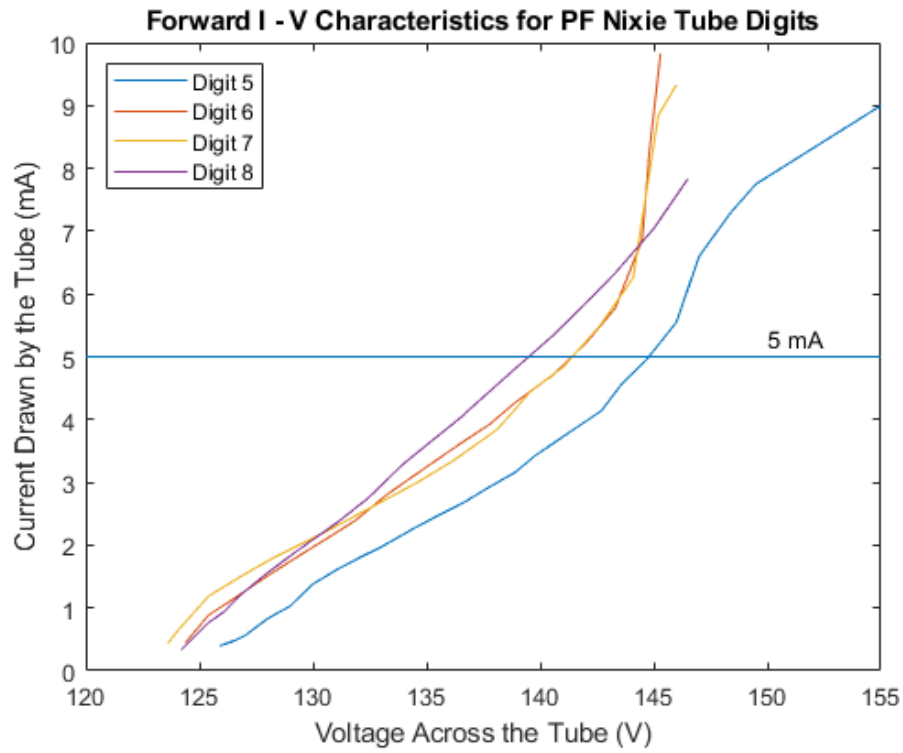
At the very start of the project we went ahead and performed I-V characteristics on the available Nixie tubes. This was an important step towards realizing the specifications of our DC - DC Flyback convertor. During the initial tube testing phase, we were in possession of two tubes of the same make. One with the single digit 0 broken (Hence forth referred to as Partially Faulty Tube (**PF Tube**) and another with all digits working . In addition to the forward I-V characteristics (Increasing voltage) we also performed the retreating I-V characteristics. The retreating characteristics involved decreasing the voltage down from the highest V reading to the turn off voltage. We were inclined to document this behaviour since the tubes were observed to have Hysteresis. For the digits for which the retreating characteristics were performed, the two plots have been overlaid.

### 2.2 I - V Characteristics results

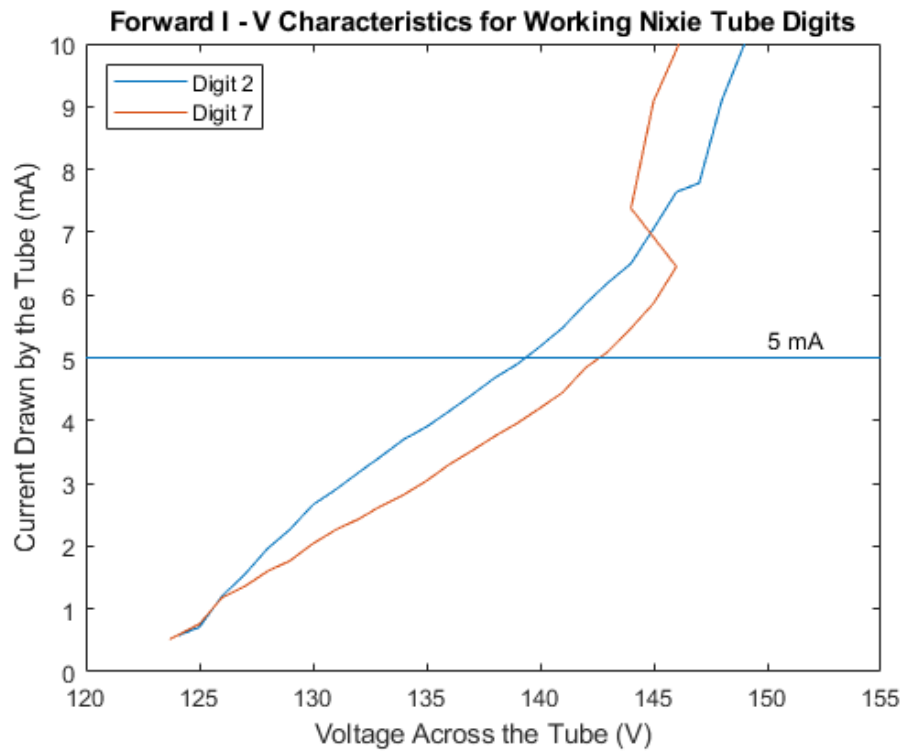
#### 2.2.1 Forward I - V characteristics

I-V characteristics for the tube with the 0 digit blown were plotted for a few digits and are shown in fig. 2.1 (a)

A similar plot has also been generated for the correct tube in fig. 2.1 (b).



(a) PF Tube Forward I-V



(b) Correct Tube Forward I-V

Figure 2.1: Forward I-V Characteristics for the two tubes

From the above plots, we can make two inferences. The first being that the current requirement at a particular operating voltage point varies slightly from digit to digit.

The second being that the effect of one digit blowing out due to over-voltage is not transferred in any way to the other digits which continue to function as desired. This is further substantiated by fig 2.2 which compares the I-V of the two tubes for the same digit.

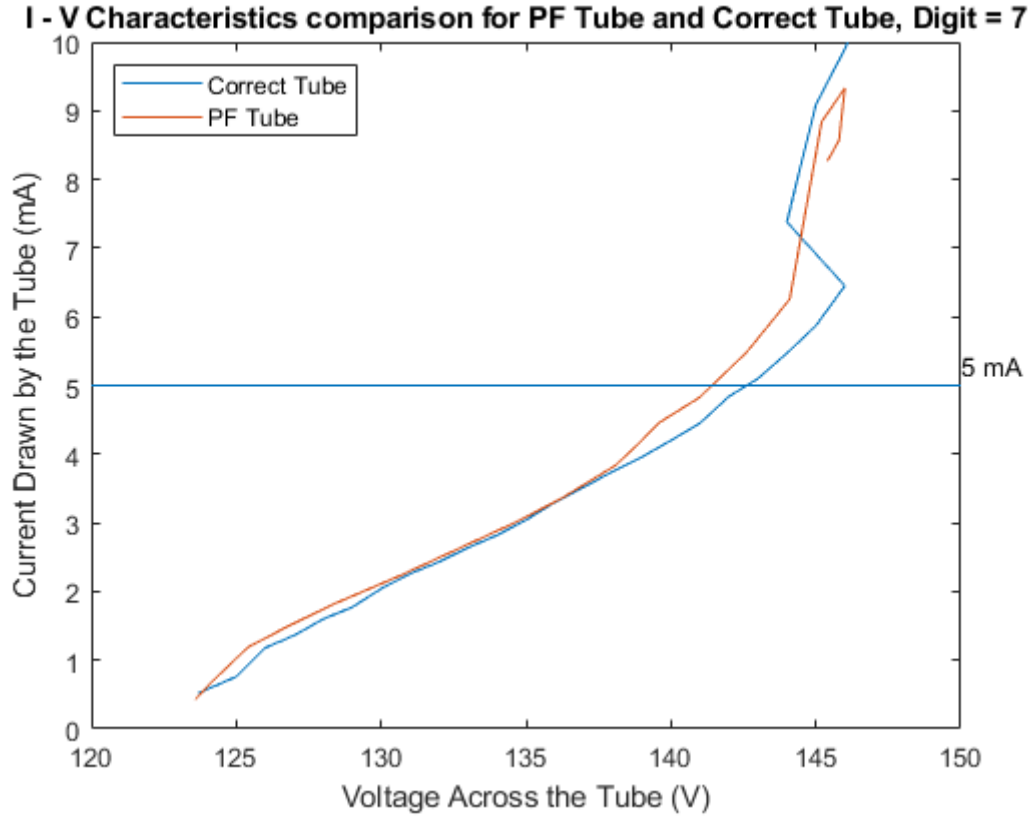


Figure 2.2: Comparison plot for Correct Tube and PF Tube



## 2.2.2 Hysteresis I-V Characteristics

Figure 2.3: As mentioned earlier, there is a Hysteresis Behaviour associated with the tubes. What this means is that the current requirement for a particular digit can be different depending on whether we are moving forward from the turn on voltage or decreasing the voltage applied. Though the current difference is low ( $\mu A$ ) range

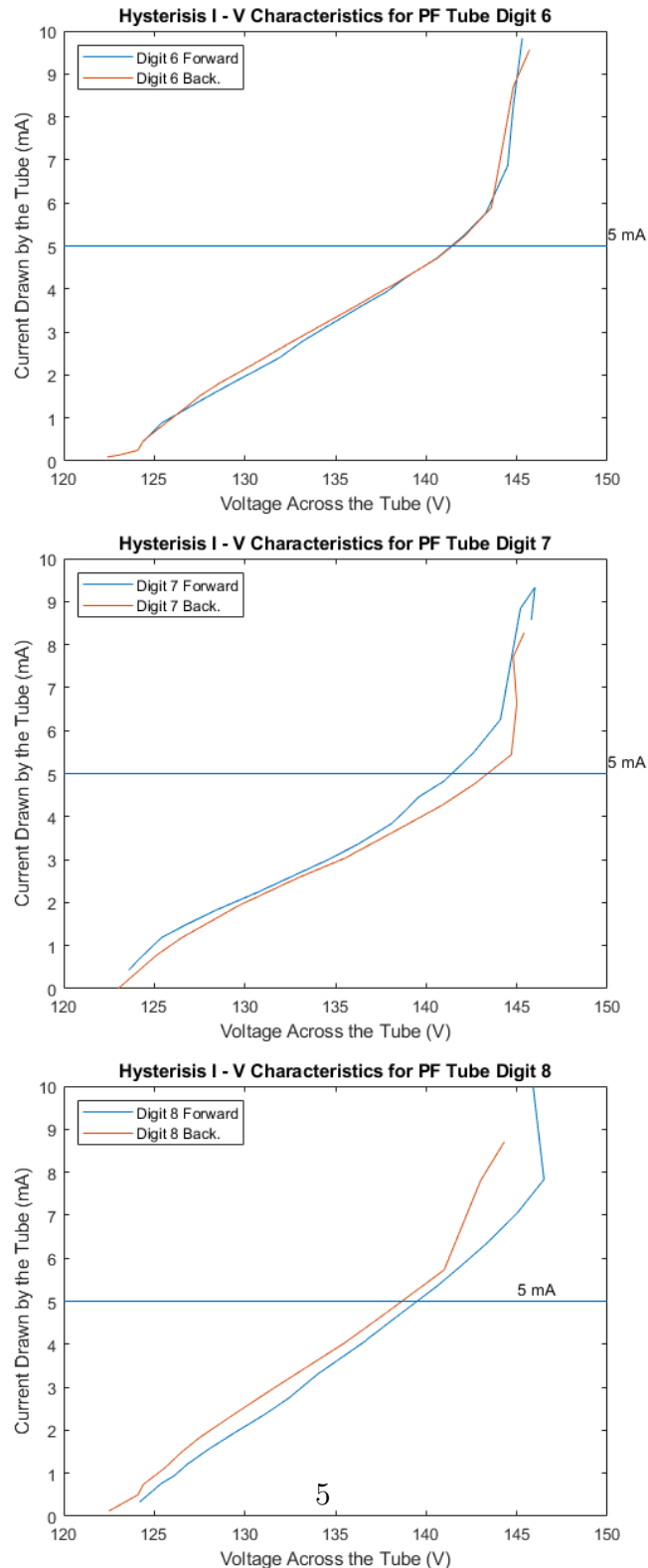


Figure 2.4: Hysteresis in I-V examples

## 3. Power Supply Design

### 3.1 Initial Approaches

In order to step up 5V DC to 180V DC, the best option was a flyback converter. A simple boost converter might have worked but the duty ratio would have been very high and more importantly no isolation would be provided which was a key point in our project. Isolation is the reason why a power supply using a rectifier and 220V mains is not preferred.

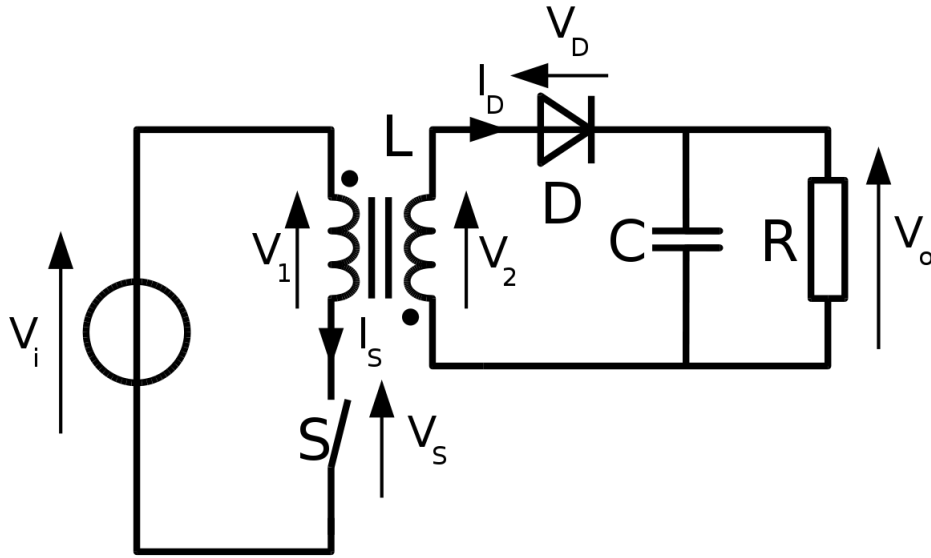


Figure 3.1: Flyback Converter

$$V_O = V_I \frac{N_2}{N_1} \frac{D}{1-D}$$

### 3.2 Existing Solutions

There are two constraints while using a USB, the voltage input from the USB is capped at 5V and the current drawn from a single USB cannot exceed 500mA .

On scouring the web we found many solutions, but most of them would be stepping up 9-12V or even the few that did step up 5V would require a current greater than half an ampere. The solution that was the closest to our requirements was found on a github page which can be found in the references.

This solution as well, was drawing more power than available, but only by a slight margin and it was decided to go forward with making the PCB for this design with the belief that tweaking a few parameters would make it a viable solution.

At the core of all such solutions was a controller IC, ( like LM3488MM) that would regulate the output voltage. Another series of ICs was the TNY series, but they are built specifically for step down converters and cannot be mirrored.

### 3.3 IC based Solution

The first approach was to use a TNY278 IC based converter circuit using a transformer salvaged from a mobile charger, since the switching IC was available to us. But, we found that it required a minimum input voltage of 5.85 volts for proper switching action. Since our power source was a USB2.0 cable, we were constrained to a maximum input of 5V. The second approach we took was to use an LM 3488 based flyback converter. This IC had an inbuilt feedback mechanism to control the switching, wherein a voltage of 1.26 volts at the feedback input of the IC ensures that the correct output has been reached by the converter. Since, the IC was capable of an input ranging from 2.97V to 40V, we found it ideal for our requirement. The feedback was provided through a simple resistor potential-divider network from the high voltage side. The output voltage is governed by the following equation:

$$V_{OUT} = V_{FB}(1 + \frac{R_{FB2}}{R_{FB1}})$$

However, this arrangement did not have the requisite isolation between the low voltage and high voltage sides. To provide the necessary isolation, it was decided to provide feedback using an optocoupler based circuit. But, the current in the high voltage side was not enough to meet the current demands of the available optocoupler, so the IC based solution was abandoned in favour of the discrete components solution mentioned in the next section.

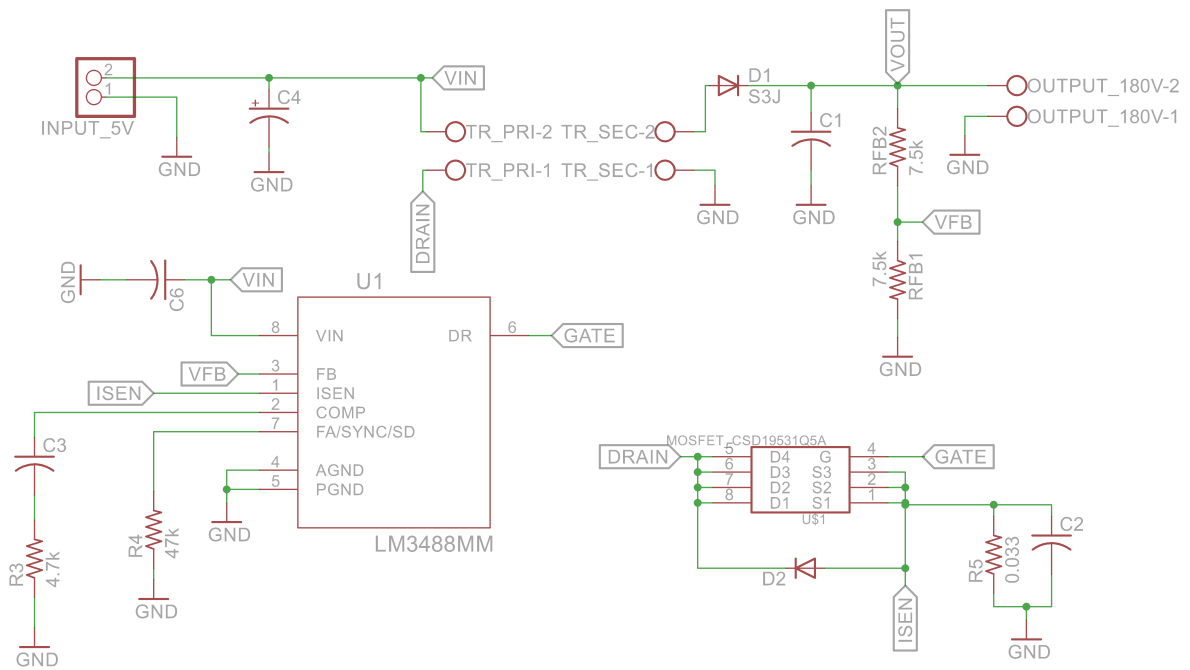


Figure 3.2: Schematic for Non-isolated LM3488 based flyback converter

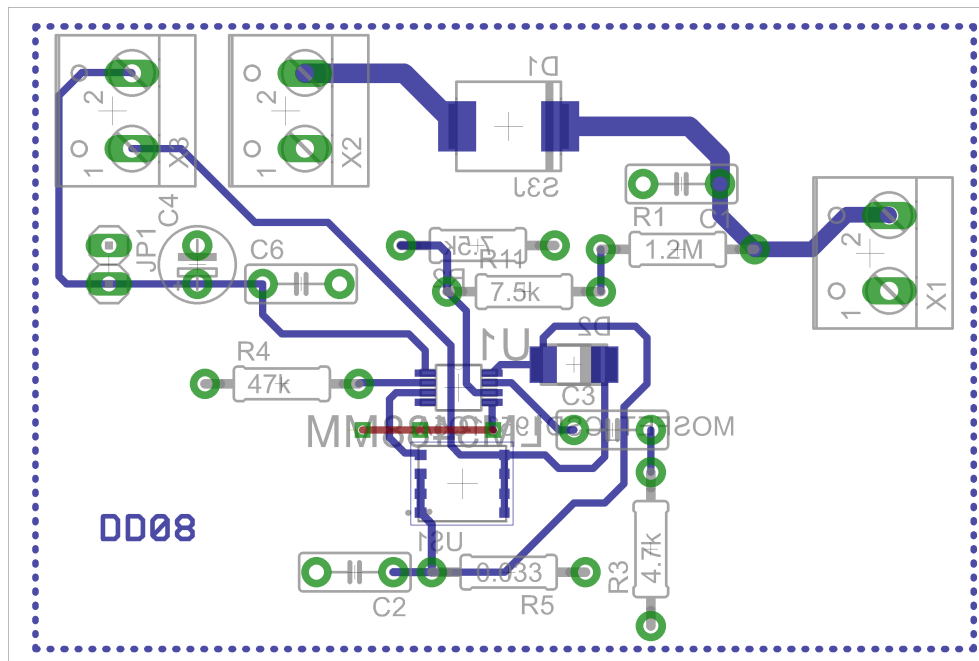


Figure 3.3: PCB for the Non-isolated LM3488 based flyback converter

## 3.4 Discrete Components Solution

### 3.4.1 Boost Converter

Having never worked with power circuits before, we first decided to make the boost converter.

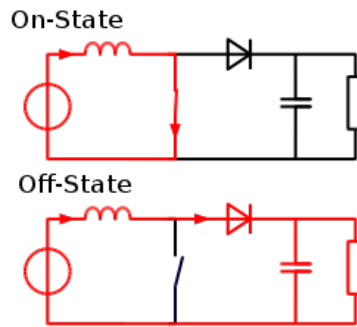


Figure 3.4: Boost Converter

$$V_O = \frac{V_I}{1 - D}$$

After choosing a switching frequency of 100kHz, we got an inductor of  $100\mu H$  a capacitor of  $100\mu F$  and in order to keep the current continuous a resistor value of  $50\Omega$  was chosen. For switching, we had two options, use either a MOSFET or a BJT. A BJT has higher losses and a higher voltage drop across it and hence we went for a MOSFET. The switch used was IRFZ44N.

We assembled the circuit, but alas the circuit would not boost.

On checking the circuit we realized that the MOSFET was not switching. But how could this be?! The datasheet clearly stated that it had a rise time and fall time in the order of nano seconds. Clearly it should switch at 100kHz. Well turns out in order to get it to switch faster, it requires a higher current to charge its gate capacitance faster and that is something the function generator could not do. So next we needed a gate driver that would be able to drive the MOSFET.

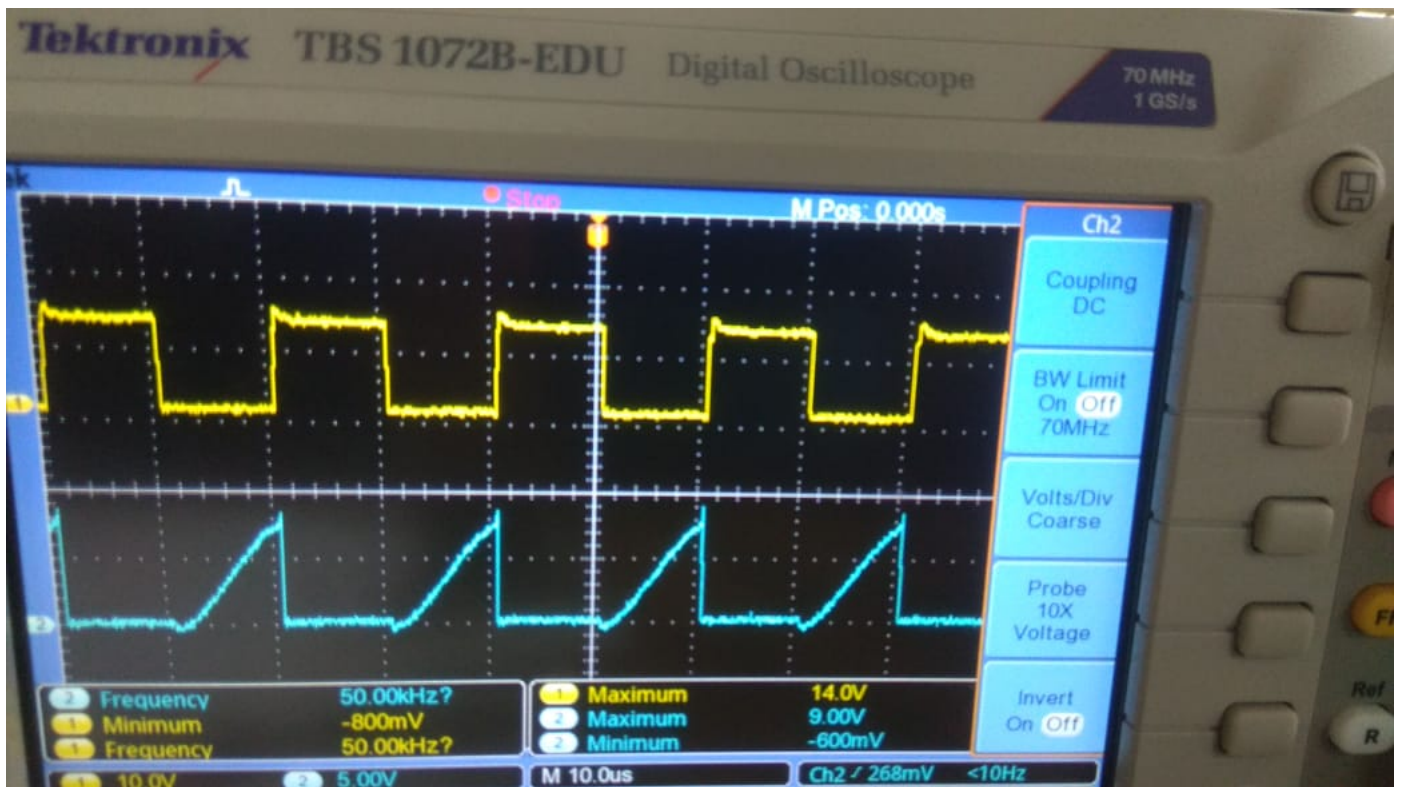


Figure 3.5: Input Signal and Mosfet Drain Signal not switching properly

We decided to use the HCPL 3120 as a gate driver and SG3542 as the controller IC. The job of the SG3542 was to generate the square pulses given a dc voltage supply. It had two resistance pots that could be used to control the frequency and the duty ratio independently. It also had other features to implement feedback but those pins were unused by us at this point. Using these we were finally able to get our mosfet to switch. But as you can observe there is a very high transient which is due to stray inductances and needs to be done away or it could harm other components in the circuit.

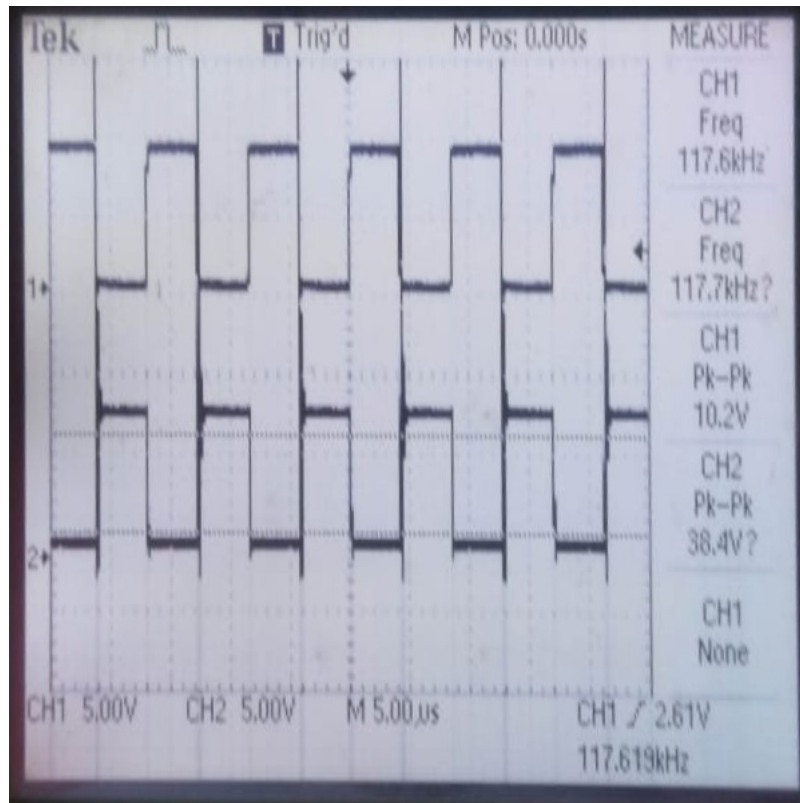


Figure 3.6: High transients

We tried to solve this by putting a free wheeling diode between the drain and the source. A better solution we found was to put in RC in series between the drain and the source that would act as a "snubber" circuit.

Another problem, with our boost circuit was that the Inductor we used had a parasitic resistance of  $170\Omega$  that was too high. So we replaced that with a better inductor which had almost negligible parasitic resistance. Finally we assembled the boost circuit again and we were able to observe the boost actions and its relation with the duty ratio.

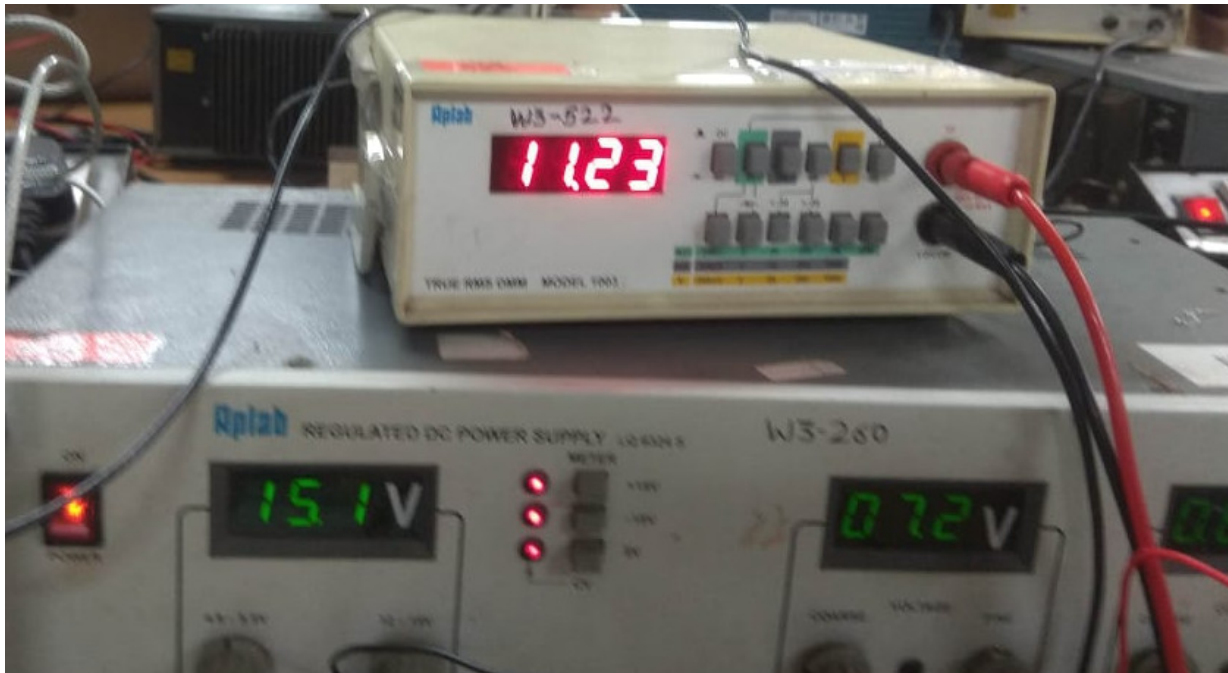


Figure 3.7: Input 7.2V , output 11.2V, duty ratio 0.4

After this we also tried a few different power converter topologies like the Buck and Buck Boost converter.

### 3.4.2 Flyback Converter and Transformer Design

Now that we were done testing the boost, we just had to replace the inductor with a transformer. In a flyback converter the entire current flows through the primary and hence the core could saturate which is why the the switching frequency needs to be high enough such that the peak current is not very high. But too high a switching frequency could lead to factors like skin effect creeping in. Fortunately at 100kHz , the core did not saturate and we did not have to worry too much about skin effect.

We designed the transformer after referring to the book 'Design of Magnetic Components for Switched Mode Power Converters' by L. Umanand and S.R. Bhat.

We wound the transformer using the winding machine in APEL Lab, on an E shaped core and the final paramaters were as follows. The input side was over designed to sustain a current of up to 1.5A. The primary side had 4 turns and an inductance of  $7.5\mu H$ . The secondary side had 200 turns and an inductance of 16mH .



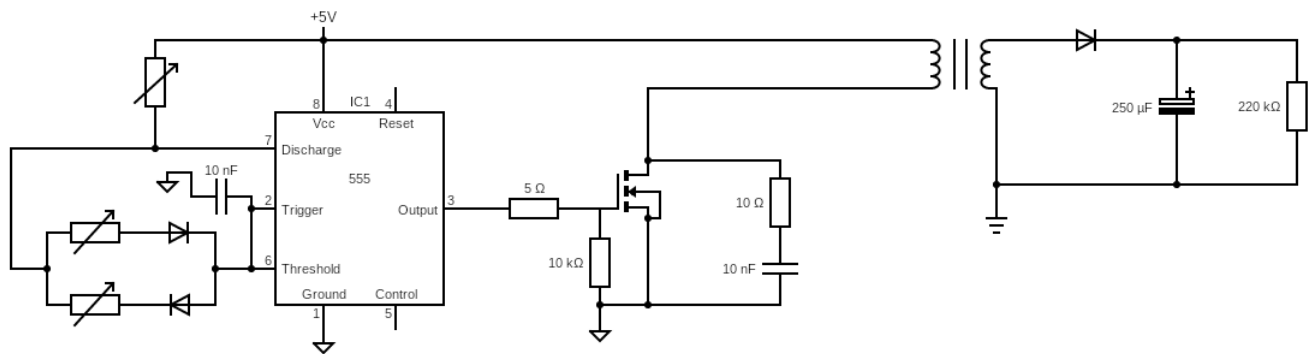


Figure 3.8: Circuit Diagram

We finally tested this on a breadboard and we were able to get an output voltage of 170V ! Care had to be taken that the 1N914 diode was replaced with a diode that could sustain reverse voltages of up to 400V and the output capacitance should also be rated for a higher voltage. While testing we made sure that the capacitor was not open and a resistor was connected to draw a current of 10mA.



Figure 3.9: Initial Flyback Converter with Nixie Glowing

### 3.4.3 Some Problems

While switching through the nixies there could be an instant of time where the capacitor would be open loaded leading the voltage to rise over that time.

SG3542 and HCPL 3120, are ics that have a VCC of 10-15V, something the USB cannot provide and hence we had to look for alternatives.

For isolation, the control side required another 5V for optocouplers that would have the ground same as the high voltage side.

### **3.4.4 Some Solutions**

We decided to use the 555 timer IC instead of the SG3542 as it could work with 5V Vcc. It wasn't such an elegant solution as in one could not change the duty ratio and frequency independently but it worked. We also realized that it could source up to 200mA and hence we decided to do away with the gate driver but this to lead to more losses due to poor switching and so we also found a 5V gate driver, the UC37322 IC.

In order to handle the issue of the capacitor being open, we decided to implement some form of a feedback.

For another 5V source, we decided to add a third winding to the transformer whose ground would be the same as the high voltage side.

### **3.4.5 Feedback**

In order to implement feedback, we decided to use the reset pin of the 555 timer. There also had to be some form of isolation between the measured signal at the high voltage side and the input signal to the reset pin. An optocoupler MCT2E was used for that purpose.

The high voltage was fed to the input of the optocoupler through a pot and the output signal of the optocoupler was fed to the negative terminal of an opamp comparator. The opamp we used was the LM324. The positive terminal of the opamp was a reference voltage derived from the usb using an LDO to keep it constant. Finally the output terminal of this opamp went to the reset pin of the 555.

We checked this implementation, by open circuiting the capacitor and the voltage did not rise above 180V. We were able to implement a simple form of on-off control!

### **3.4.6 Adding a Third Winding**

In order to add the third winding, we had to unwind the transformer and add another winding. Since the original transformer was not designed to have three windings it was difficult to predict the turns ratio that would give us 5V at the output. After some tinkering and adjusting the air gap in the transformer with different materials, 9 turns was giving us an output voltage of 165V on the secondary winding and 6.2V on the tertiary winding.

Care had to be taken to implement a similar flyback at the tertiary end as well. We used an LDO after this to get 5V from the slightly varying 6.2V.

### 3.4.7 PCB Design

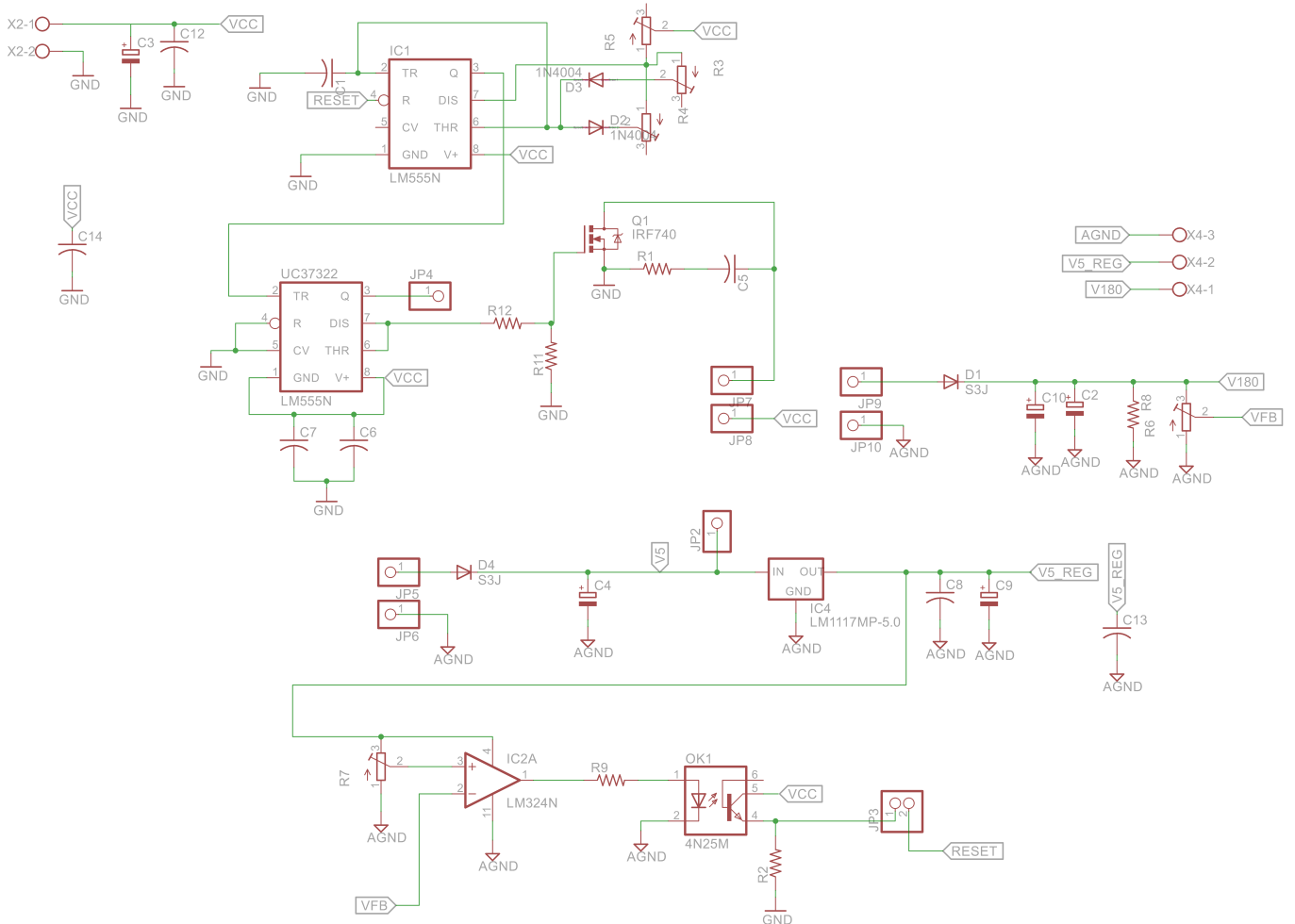


Figure 3.10: Schematic

Because of stray inductances, care was taken to add a decoupling capacitor for each IC. Also the gate driver and the MOSFET were kept as close as possible. In order to reduce the length of the traces we went for a two layer PCB, although it turned out to be much harder to solder.

We also kept no ground plane on the high voltage side to minimize ringing effects.

Here's the final picture of the PCB of our converter.

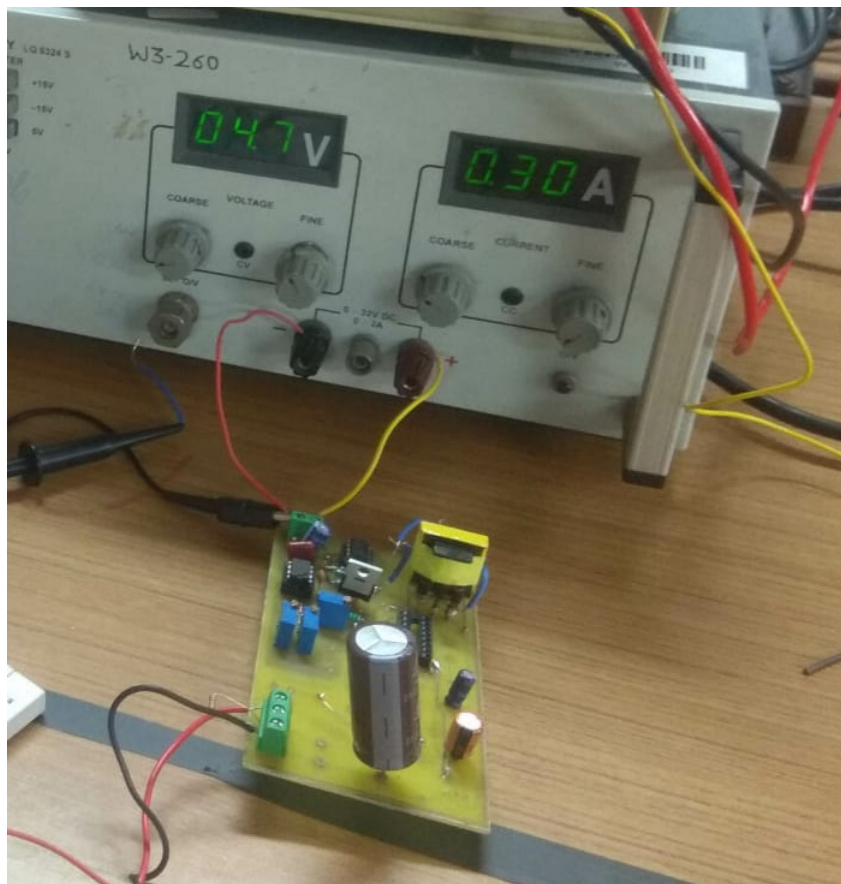


Figure 3.11: Flyback Converter on a PCB

## 4. Controller circuit Design

### 4.1 Role of the Bridge circuit

The subsequent sections document the design decisions performed in creating the USB Bridge circuit. The USB Bridge is responsible for displaying the timing information from the Host computer and then changing the control signals to reflect the same. Ideally the frequency of the communication between the host and the controller should be kept to a minimum. And the multiplexing should be implemented by the controller itself with only the required port configurations for the current timing cycle (Lasting one second due to HH::MM::SS format) coming in from the Host.

Keeping this in mind, the firmware for the controller was written and is available at the reference [2].

### 4.2 Design Choices for USB bridge circuit

The objective of the Bridge circuit can essentially be boiled down to toggling control signals directly or indirectly from a program running on a Host Computer.

Two options were considered to implement,

The classic USB - UART interface, the **FT232** re-purposed as some GPIO pins using some of its additional functionality

An 8052 based micro-controller with USB enumeration built into it - The **TUSB3210**.

### 4.3 Using the FT232R to achieve Parallel out GPIO pins

The FT232R is a very popular USB-UART interface and is used invariably in EEPROM programmers and many USB peripherals. The FT232R also has a lot of functionality beyond USB-UART interfacing. Two such modes available are the Asynchronous and Synchronous Bit-Bang Modes. These modes allow us to freely toggle about the output pins on the FT232R, thereby providing the Bridge interface.

This approach was explored, but later discarded due to some driver limitations. There are two kinds of drivers that FTDI (The company that manufactures FT232) has made available - The VCP Drivers and the D2XX drivers. Every linux distribution comes loaded with the VCP drivers by default, so initially it was used. However the only Host side USB interface compatible with the FT232 under the VCP drivers is the **IOCTL()** system call in linux. The issues involved with using IOCTL for Bit-Banging are listed below.

- The IOCTL system call allows only to flip a single bit every system call.
- This could lead to unpredictable timing behaviour due to the latency involved between the OS communicating with the FT232 device.
- The delay here refers to the time it takes to communicate between the Host PC and the board. This delay is non - deterministic and depends on when the OS - schedules the FT232 controlling process.
- Taking into account timing constraints, this unreliable timing could lead to synchronization problems

A more convenient programming interface that allows for Bulk USB transfer is provided by the extensive functionality of the D2XX driver API. However, these drivers do not have complete documentation that would allow for easy installation on an Ubuntu 16.04 system. Keeping all these factors in mind the TUSB3210 micro controller was chosen instead. Another interesting thing to note is that the cost of the FT232R IC is comparable to the TUSB3210 controller with just a 20% premium to be paid for the much more general TUSB Controller based solution [3].

## 4.4 Bridge Circuit using the TUSB-3210 Microcontroller

The TUSB3210 is a RAM only Micro-controller which used to be a very popular Texas Instruments controller for USB peripherals such as Keyboards and Mice until a few years ago. The firmware for the TUSB3210 controller is stored in the RAM only. This is not very common for Micro-controllers and a discussion as to the pros and cons of such an arrangement can be found at [4].

To build a schematic around the TUSB controller, the schematics referred to were [5] The vc33 DSP Board and [6] The TI 3210 evaluation board.

### 4.4.1 TUSB Power Supply

The TUSB controller requires 2 power Supply levels, 3.3V and 1.8V, these were created in accordance with the application notes provided in the TUSB3210 Data sheet, the schematic is provided here for reference.

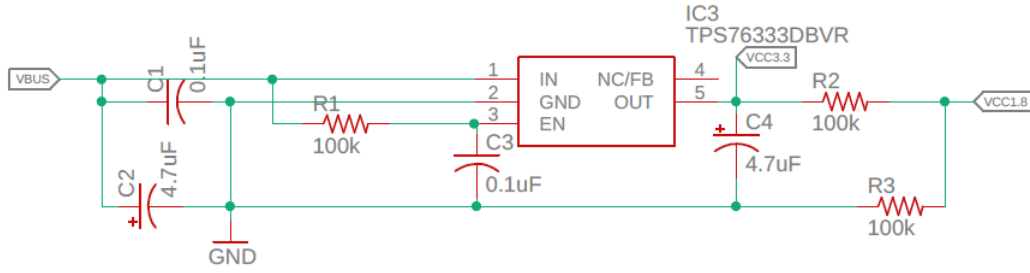


Figure 4.1: Dual Power supply for the TUSB 3210 Controller

Like many micro-controllers, the TUSB3210 requires an enable based regulator. When enable is low, the regulator cannot output current, but after the R-C delay of R1 - C3 is complete, the regulator is activated and the controller is powered up. This is done so that controller peripherals have time to power up before any program starts running on the controller.

#### 4.4.2 Micro-controller Schematic

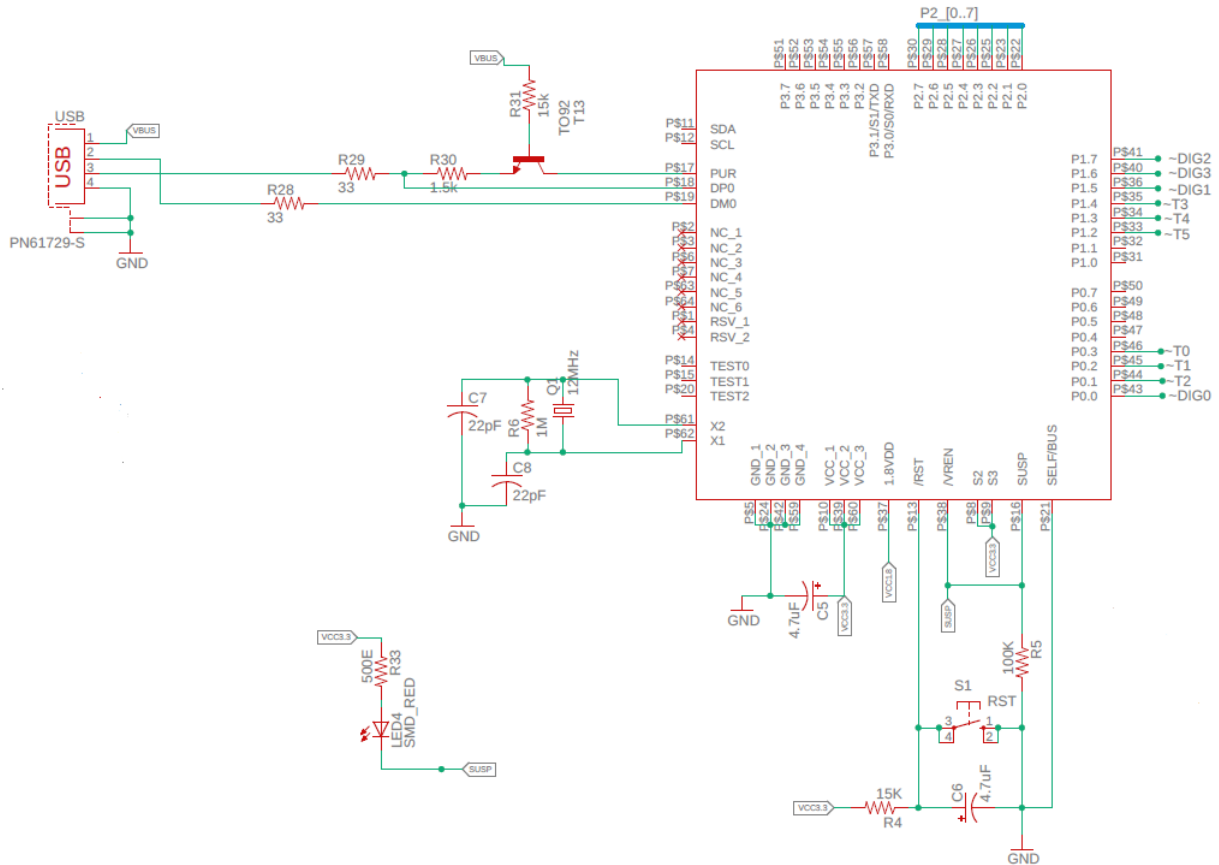


Figure 4.2: The arrangements required for the TUSB3210 Controller

The arrangements for the Crystal oscillator, Reset button and USB connections have been done in accordance with standard USB practices and the reference schematics. The result is shown in figure 4.2.

At this stage the reader is warned about a certain issue with using Schematic symbols and Footprints for the TUSB3210 controller. Not all footprints (pin-outs on the layout) available online for the TUSB3210 are correct. In our project, we got our circuit board printed with a footprint in which two pins were interchanged, this had to be painstakingly corrected with the help of WEL staff.

### 4.4.3 GPIO Interfacing

As explained earlier, the GPIO pins of the micro-controller are being used as the control signals for the High voltage (HV) side.

For the control signals to be isolated from the HV side, opto-coupler isolation was used for each of the control signals involved. Control signal isolation between the control side and the High voltage side is standard practice in electronics and is required for the safety and reliability of electronic equipment.

The current sourcing capability of the GPIO pins is limited by internal hardware of the Micro-controller to about  $100\mu A$ , which is on the lower side as far as micro-controllers go [7]. Because of this, a BJT switch based current amplification stage was added between the Port pins of the controller and the input LEDs of the MCT2E opto-couplers used. The arrangement is depicted in figure 4.3.

It is important to note here that for the output Photo-transistor of the MCT2E to serve as an effective switch, sufficient current must be pushed through the input LED. In an earlier configuration we had taken the series resistance of the input LED to be  $220\Omega$  instead of  $30\Omega$  this proved to be an issue since the output transistors never switched completely off in this case.



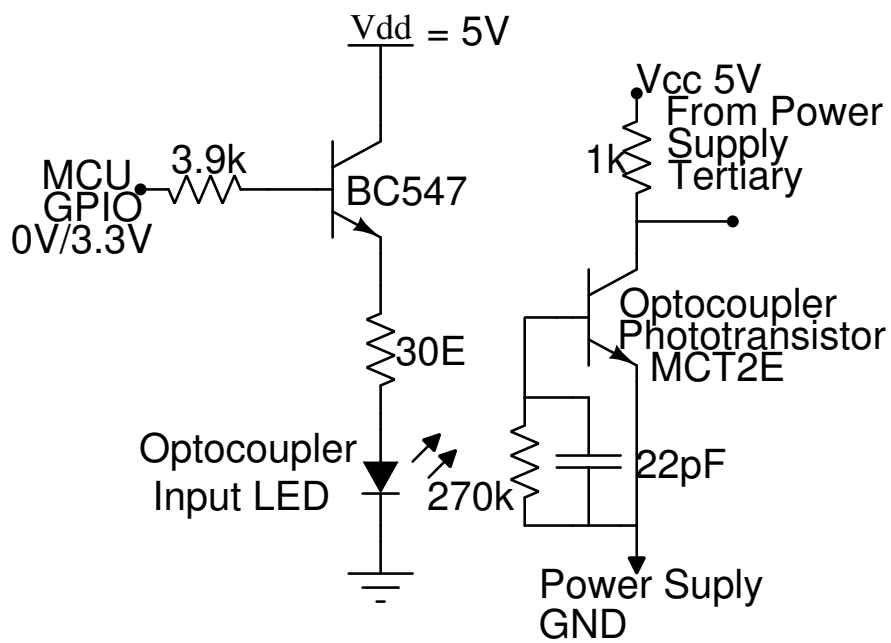


Figure 4.3: Opto-coupler Interfacing with Controller GPIO pins

## 5. Front-End Driver Circuit

### 5.1 Description

The driver circuit is the front end of the display that uses the control signals generated by the controller stage to implement the Multiplexed display. It requires a total of 10 control signals, 6 control signals for selecting which tube lights up at any instant (Tube Control) and 4 control signals which are first decoded and then decide which of the 10 digits is selected at any instant. Both of these controls are described in this Section of the report.

### 5.2 Tube Control

As described in the previous section, six control signals are dedicated to selecting which tube is on at any given time. The tubes are turned on and off using a Darlington pair [8] and [9] like arrangement of High Voltage Rating Transistors (MPS A42 and A92). The configuration is shown in figure 5.1

In this configuration when the NPN transistor is on, the Tube is on and the tube is off otherwise.

### 5.3 Digit Control

Each Nixie Tube has a total of twelve pins, one Anode and eleven cathodes corresponding to the digits 0-9 and the Decimal point. Barring the decimal point, there needs to be a mechanism to choose between the 10 digits at any given instant.

An off the shelf solution is available for the same in the form of the 74141 TTL IC (No longer manufactured, but a Russian equivalent is available on Ebay). We were fortunate to find some of these ICs in WEL lab.

The Difference between the 74141 and a regular BCD decoder such as the 7442 is the presence of open collector outputs with Zener Diode Clamping and use of high voltage rating transistors at the outputs. This is crucial for the Nixie Tube application since a large voltage could appear at the Cathodes of the Tube which are not turned on.

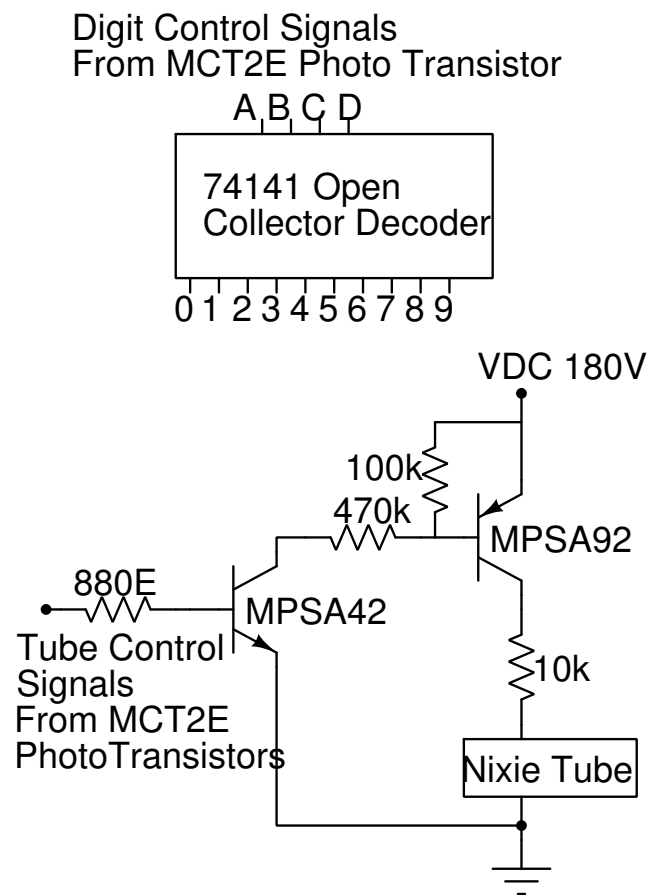


Figure 5.1: Tube Control Switching using the Sziklai Darlington Transistor Configuration and Digit Control using the 74141

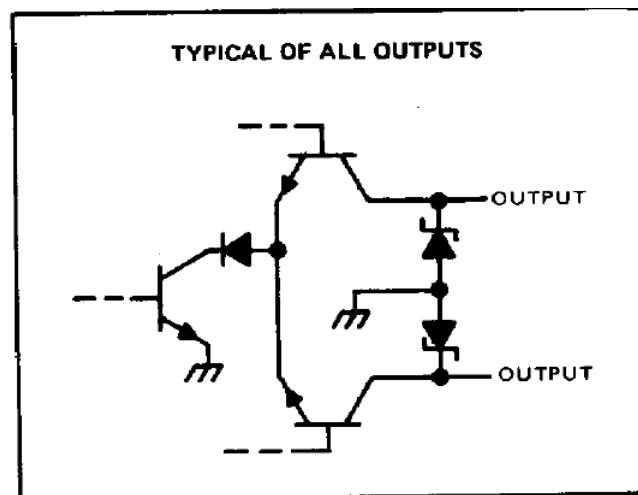


Figure 5.2: Output pins on the 74141 have the required Switching action and voltage protection for Nixie Digit Control

## 5.4 Complete Schematic and Board View of the Controller plus Driver Board

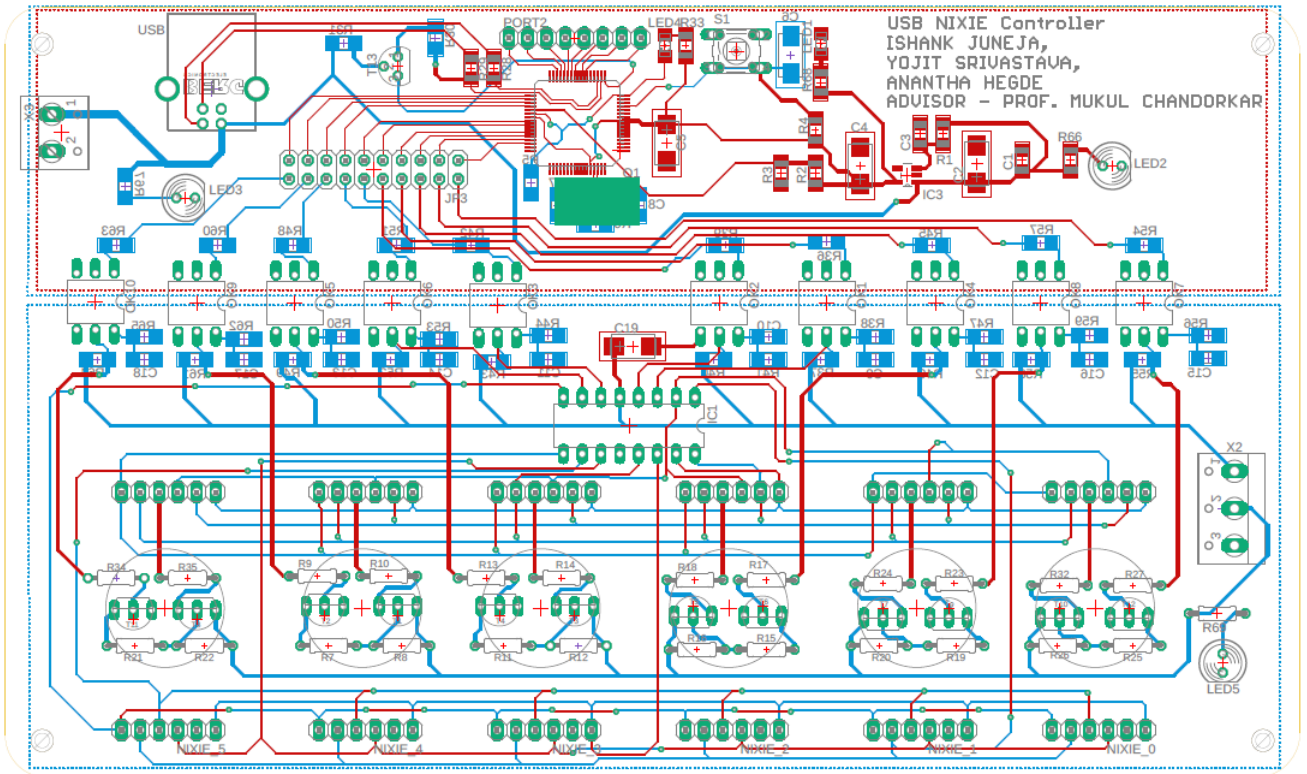


Figure 5.3: Controller Board Layout for the Nixie Tubes

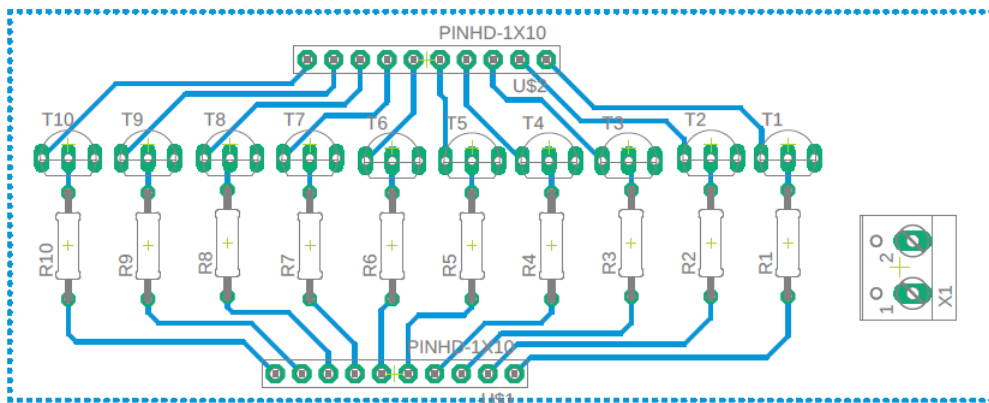
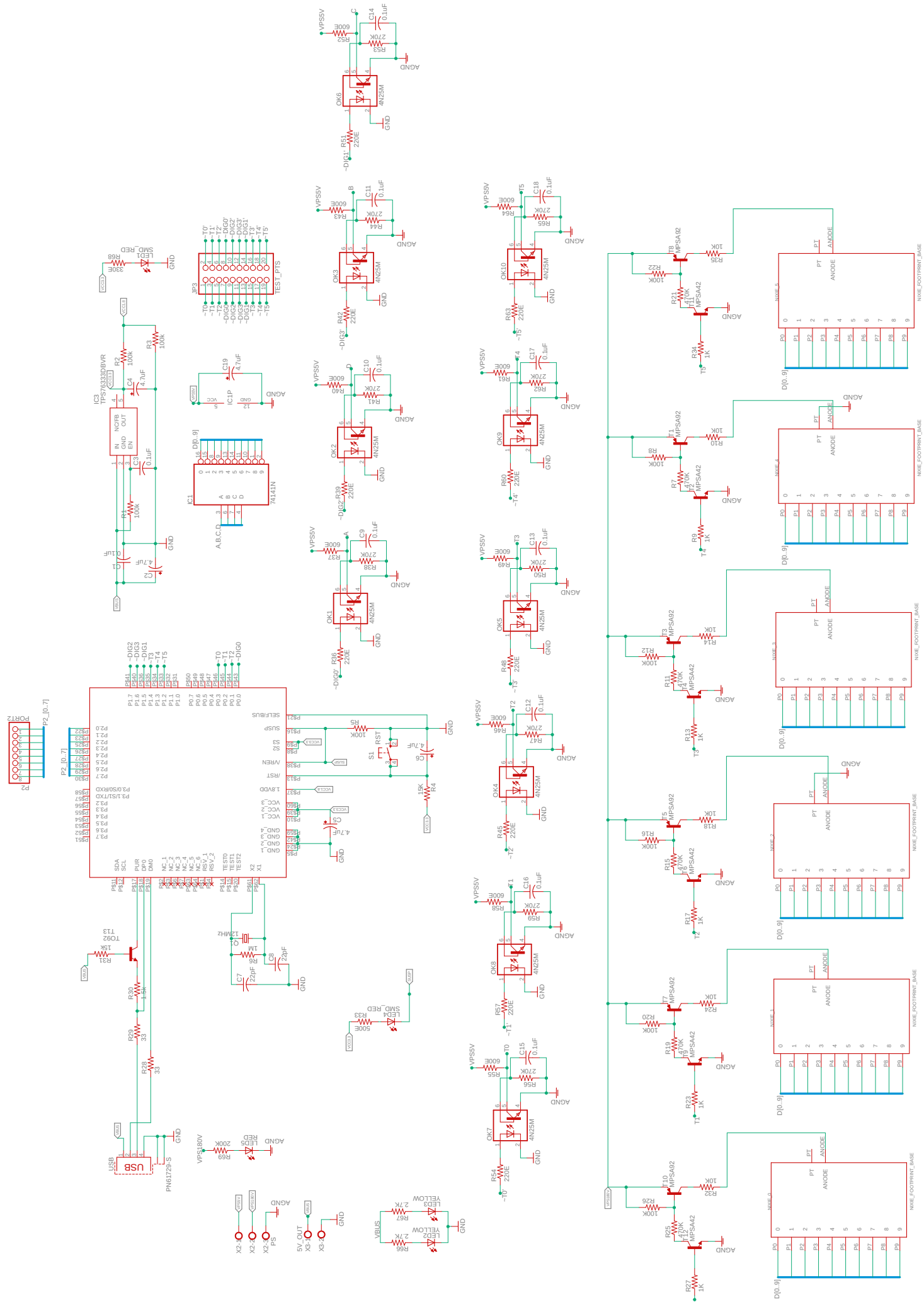


Figure 5.4: GPIO Current Amplification stage



## 6. Acknowledgements and References

In the process of implementing our project, we received tremendous help from different bodies of the department. In addition to our faculty advisor Prof. Mukul, we would like to thank, in no particular order

**Mr. Joginder Yadav**, Research assistant, Machines Lab, who was of immense help in making suggestions and corrections to our design

**Mr. Maheshwar Mangat**, Wadhwani Electronics Lab, who put in a lot of time into helping us understand the intricacies of building a circuit board around a microcontroller and helping with the layout thereof

**Mr. Chandrashekar Shele**, Wadhwani Electronics Lab, who used his skill to make intricate corrections to our circuit board

**Mr. Girish O**, Research assistant, WEL, who was our RA and made sure we made regular progress in our project work by helping us with his vast electronics expertise

**Ms. Sneha Mistry**, who had earlier designed a Nixie Display front-end which served as a starting point for our own design.

### 6.1 General References

[1] Physics of Nixie Tubes

### 6.2 References for Bridge circuit Design

[2] Github repo link for Project Firmware and Host programs, also hosts the Updated Web version of this project report

[3] Cost comparison of TUSB3410 and FT232 on Element 14

[4] On ROM vs RAM based microcontrollers, discussion on Electronics Stack Exchange

[5] vc33 Schematic, PEPS IITB

[6] TI Evaluation Board around the TUSB3210

[7] ATmega Controllers support upto  $40mA$  per pin

### 6.3 References for Driver Circuit

[8] A previous Nixie Clock project by Sneha Mistry

[9] Reference for Darlington pairs 74141 IC Datasheet

## 6.4 References for Flyback Converter Design

[10] Reference for Github IC based solution

[11] Design of Magnetic Components for Switched Mode Power Converters by L.Umanand and S.R. Bhat

[12] A helpful aide to designing SMPS circuits