ULTRA FAST I-V MEASUREMENT

INTRODUCTION

- Higher current would increase the internal temperature of sample, even though allowed for small interval of time
- The increased temperature would deviate the I-V measurement
- Temperature controlled methods are complex and costly
- Need to pass current or apply voltage for very small duration and make the measurement

PULSED I-V MEASUREMENT

The effective solution includes using pulses of increasing current/voltage, with pulse width less than 60us. The two successive pulses would have significant delay between them, so as to allow the sample to cool down. This delay could be controlled by us.

The pulses could be produced with the help of a microcontroller, which could simultaneously perform the task of measuring current/voltage and the corresponding voltage/current, in between the pulse.

CHALLENGES

- **<u>Microcontroller limitations</u>** – For analog reading of voltage and current value, we normally use analogRead() of arduino. This simple function takes around 100us of time to execute. However, we aim to achieve a pulse width of less than 60us, which is not possible with current implementation.
- **<u>Settling time and noises</u>** – The pulses produced would have significant settling time (as explained in future slides) as well as they are prone to noises. This might lead to error prone I-V measurements.
- **<u>Generating higher currents</u>** – Extracting current of the order>10A requires suitable components and large number of batteries, as ordinary power supplies can’t provide higher currents.

PROBLEM WITH ANALOGREAD()

Arduino provides a very simple command for reading analog values, analogRead(). However, this takes around 100us to execute because it is further decoded into simpler commands, not visible to user and it waits until conversion is done.

Instead, if we directly use ADC commands using an interrupt, we could handle this issue. Arduino has 5 ADC pins but a single ADC channel, hence we require to multiplex between 2 pins during measurement. One pin would measure voltage and other would measure current.

USING ADC COMMANDS

analogReD

```
ADCSRA = 0;        // clear ADCSRA register
ADCSRB = 0;         // clear ADCSRB register
ADMUX =0x60;     // left align ADC value to 8 bits from ADCH register
ADCSRA |= (1 << ADPS2);   // 16 prescaler for 76.9 KHz
ADCSRA |= (1 << ADEN);   // Enable ADC channel
ADCSRA |= (1 << ADSC);   // Start  ADC conversion
while(ADCSRA & (1<<ADSC));   // Wait till conversion is completed
a = ADCL ;  //Store the 8 bits ADC value
```

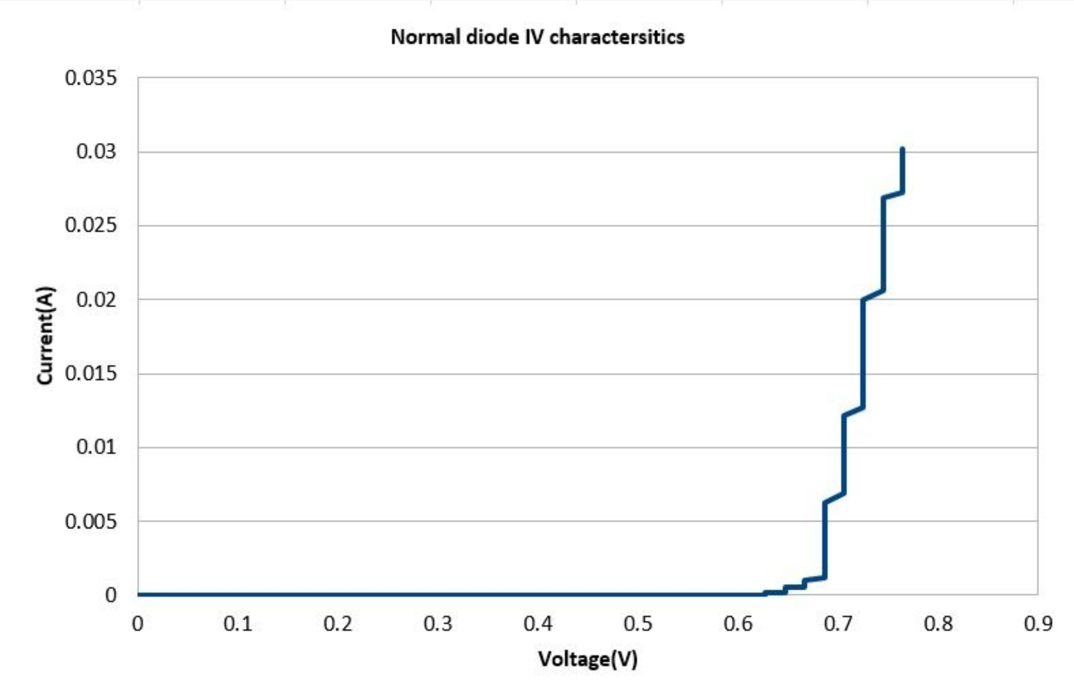
MEASUREMENT USING VOLTAGE RAMP

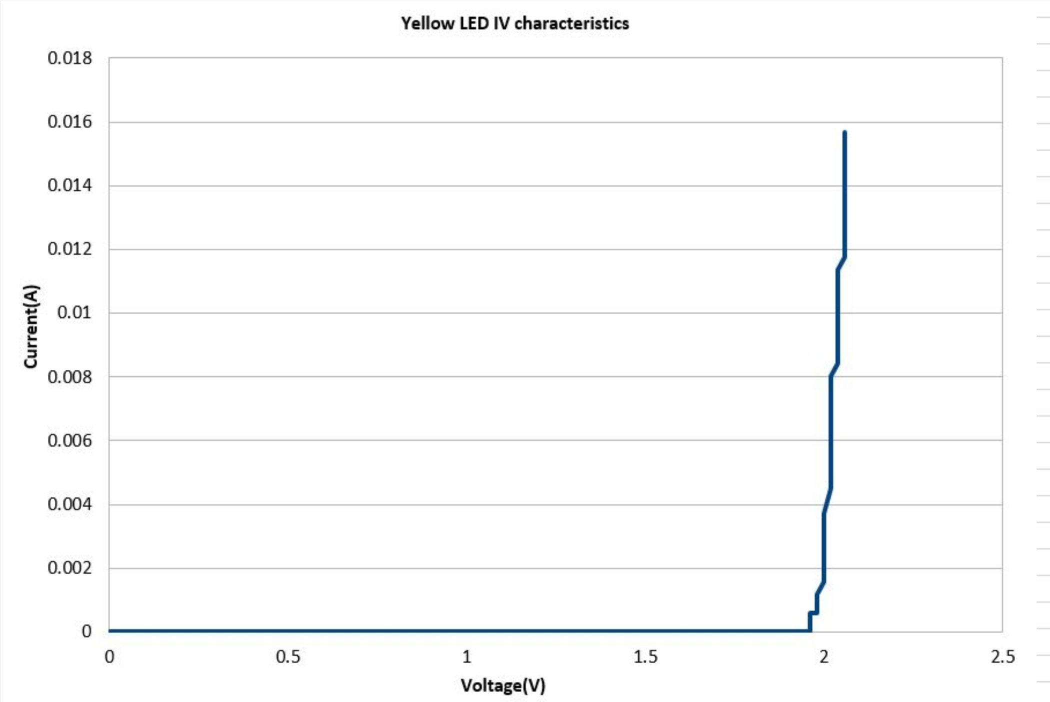
-RAMP GENERATED WITH INTEGRATOR CIRCUIT-

This was among out preliminary circuits, where I made an integrator circuit using opamp(UA741 IC)and used it for producing a very sharp ramp (0V-14V in ~1ms).

The user has to toggle a button which gives a high pulse (for very short period) as input to the circuit. As a result, a very sharp voltage ramp is produced, which is then applied to a voltage to current converter circuit as shown in the figure. The corresponding voltage and current values across sample are measured by a dedicated microcontroller(Arduino).

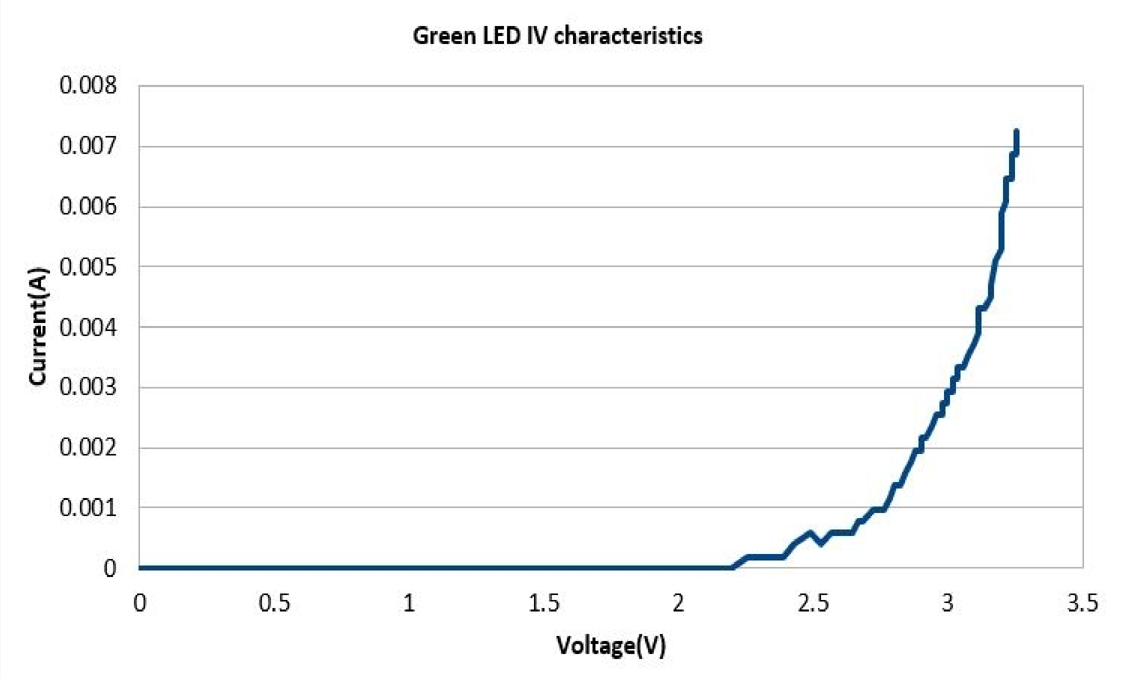
RESULTS (8-BITS ADC)

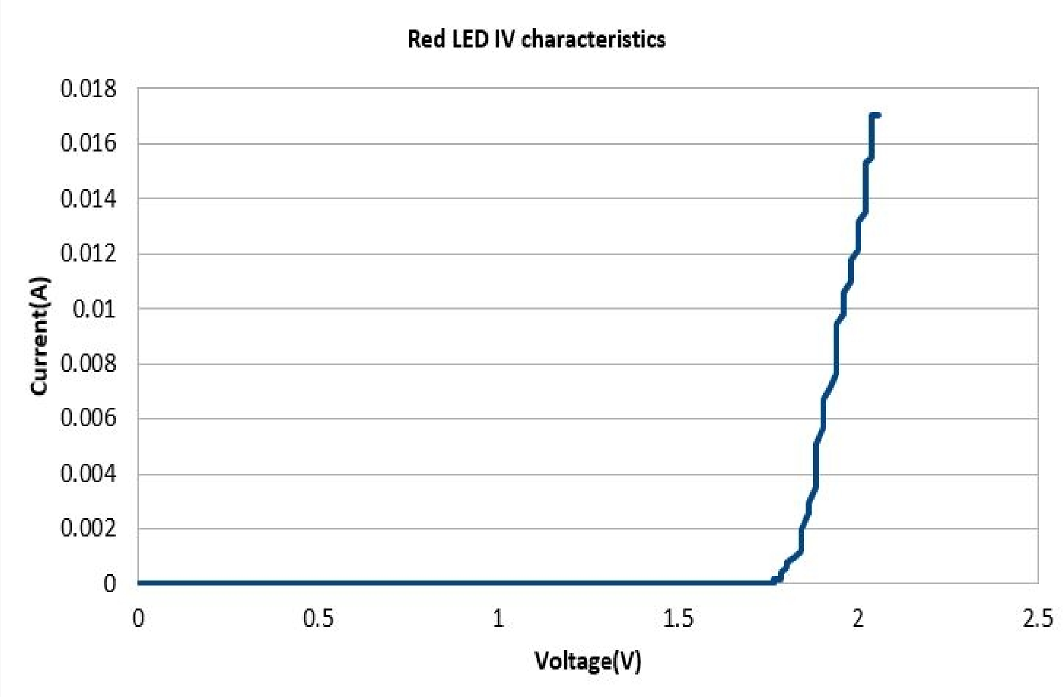




With 8 bits ADC, minimum measurable voltage by arduino=19.53mV

RESULTS (8-BITS ADC)



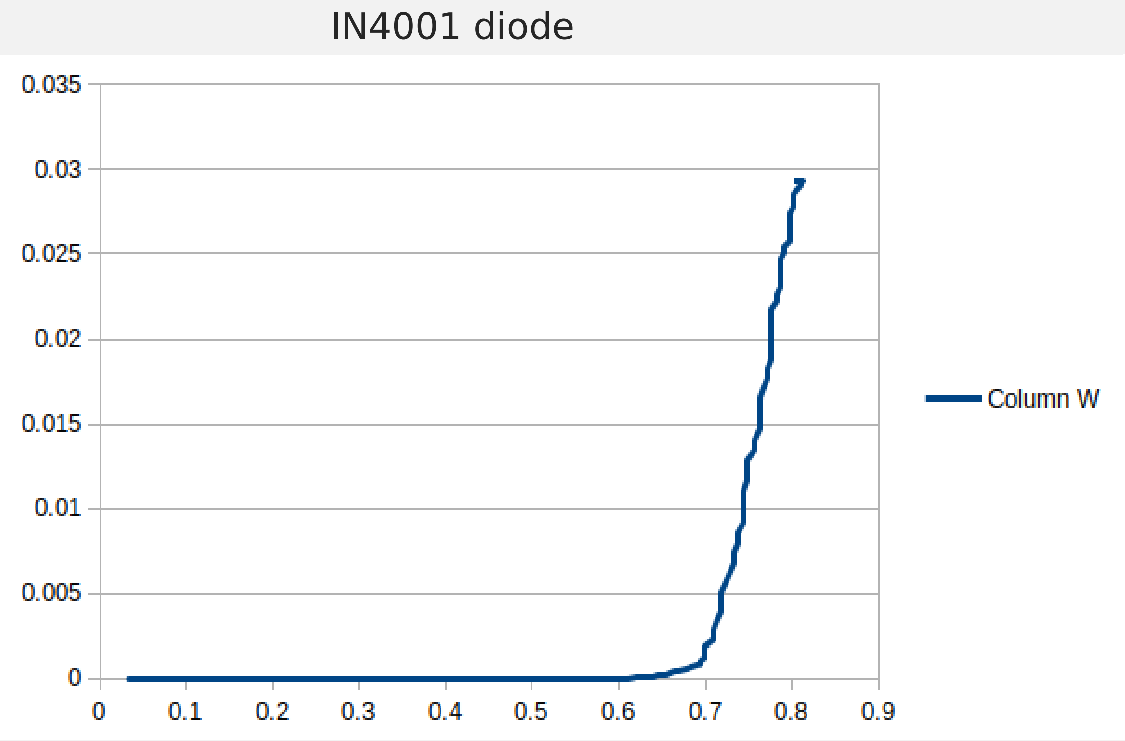


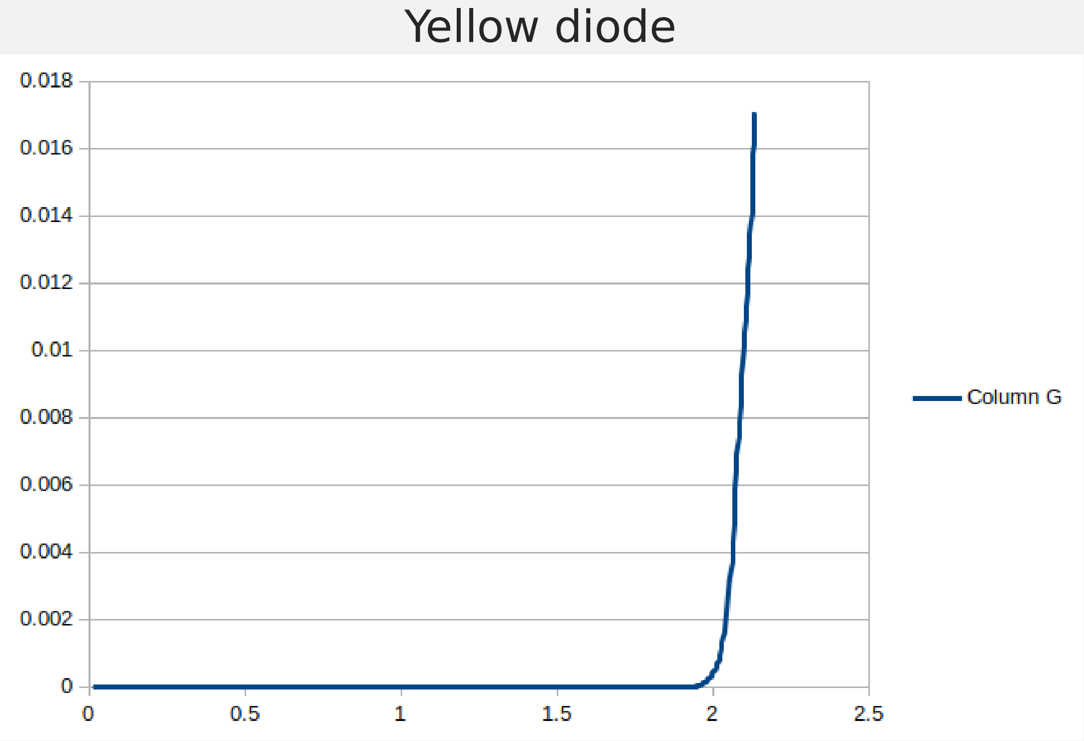
SWITCHING TO 10 BITS ADC

```
.
.
.
while(ADCSRA & (1<<ADSC));  //wait till ADC conversion is converted
a = ADCL>>6 ;   // Store the lower two bits by right shifting
b = ADCH <<2;  // Store the higher 8 bits by left shifting
y = b|a;  // Store the 10 bits ADC value
.
.
.
```

With 10 bits ADC, minimum measurable voltage by arduino=4.88mV

RESULTS (10-BITS ADC)

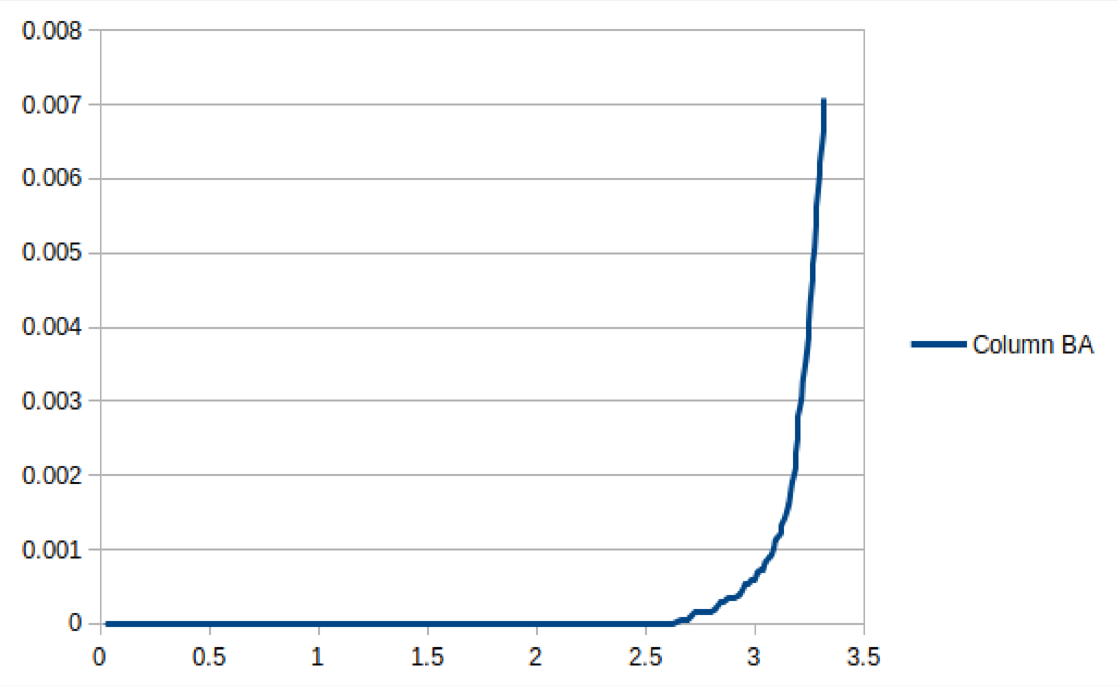




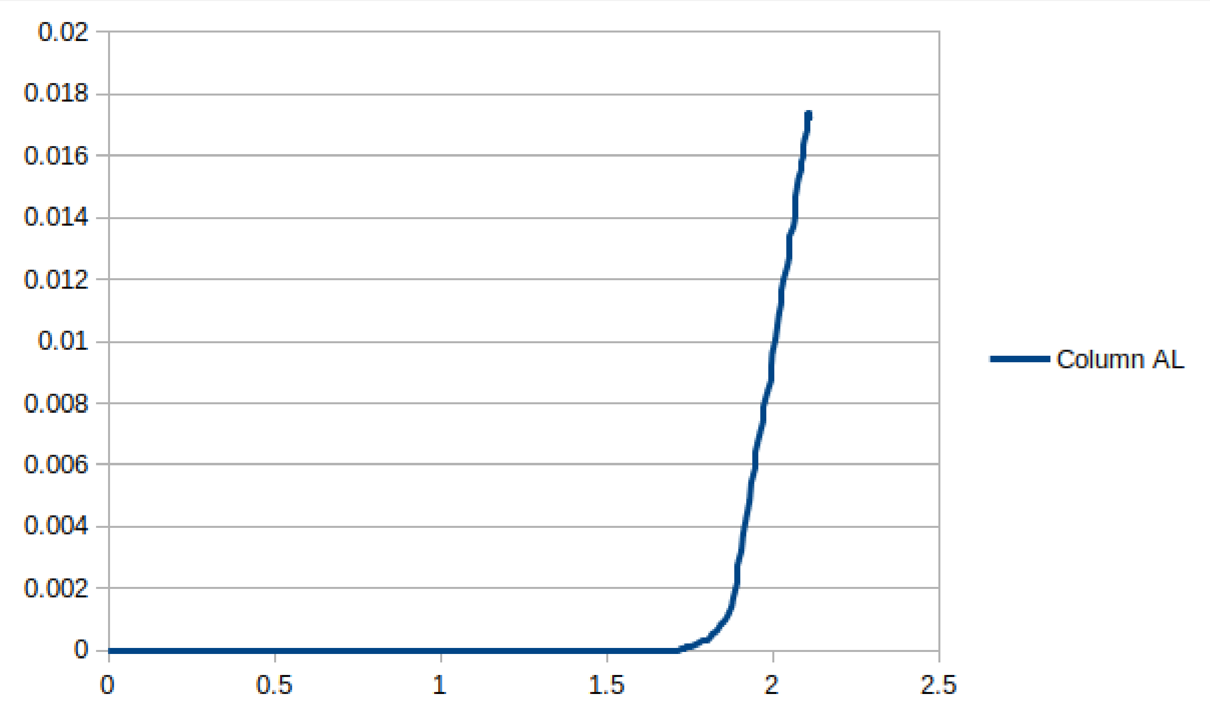
X-axis: Voltage(V), Y-axis:Current(A)

RESULTS (10-BITS ADC)

Green diode



Red diode



X-axis: Voltage(V), Y-axis:Current(A)

ARE THE PLOTS CORRECT???

No, the plots aren’t correct.

The current values rises very sharply with voltage. This is because we are using ramp of voltage. The voltage is continuously increasing with time and hence current, even while we are in measuring phase. We measure the voltage correctly, but by the time we start to measure current (roughly after 30us when voltage is measured), we arrive at different current.

As a result, current at every point is higher than expected.

SOLUTION : Use pulses instead of ramp !!!

RESULT ANALYSIS

- The past results show that we will require a ADC of 10 bit resolution.
- The curves obtained with 10 bit ADC are much smoother and have more number of I-V points.
- Development boards like Arduino could handle 10 bit ADC without any external support, hence we could easily use them.
- Currently, we were using a low current MOSFET. For obtaining higher current we need to switch to power MOSFET.
- Pulses should be used instead of ramps for proper I-V measurement and very minute temperature change.

GENERATING UNIFORM VOLTAGE PULSES

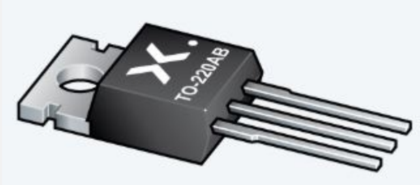
For generating pulses, I used a 8-bit digital to analog converter (DAC0808).

Voltages from 0V to 5V are mapped from 0 to 255, in arduino and are converted into binary equivalent. These signals are applied in form of pulses of very short duration to DAC and the corresponding analog pulses are generated. The duty cycle of each pulse could be controlled by arduino.

These voltage pulses are then used to generate corresponding current pulses using appropriate circuit.

SELECTING PROPER MOSFET

We require MOSFETs that could supply large drain currents. One such MOSFET is **PSMN7R0-100PS PMOS**. Another such MOSFET was BUK9Y12-100E PMOS.



Reasons for selecting PSM over BUK:

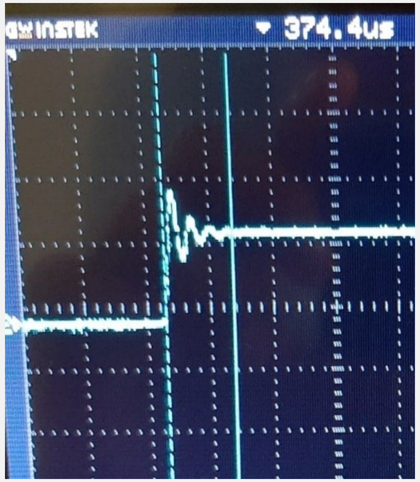
- BUK is surface mounted, while PSM is through-hole component and hence could be soldered easily.
- Heat sinks could be easily installed in PSM MOSFET.
- For Vds=3V and Vgs=4V, PSM has drain current of around 15A, while BUK has drain current >180A, which is too large, i.e PSM is less sensitive than BUK.

ADDRESSING RINGING EFFECT

Power MOSFETs have a large trans conductance and parasitic capacitances. When the MOSFET turns on, the di/dt of the drain current and the stray inductances of the drain lead and wire, cause a voltage surge across the drain and the source. The surge voltage is expressed as:

$V_{Surge}=L_D\times di/dt$, L_D=Drain stray inductance

The circuit causes ringing since the surge voltage resonates with the C_{ds} of the MOSFET and the stray inductance L_D. The surge voltage is superimposed on the V_{gs} voltage via the gate-drain capacitance C_{gd} of the MOSFET. As a result, it affects the gate inductance, causing ringing of the gate voltage



FASTER SPEED OR LESS RINGING PEAK ?

The ringing effect could be reduced by increasing the input gate resistance, which effectively increases the charging time and decreases di/dt. As a result, surge voltage decreases and hence the ringing peak.

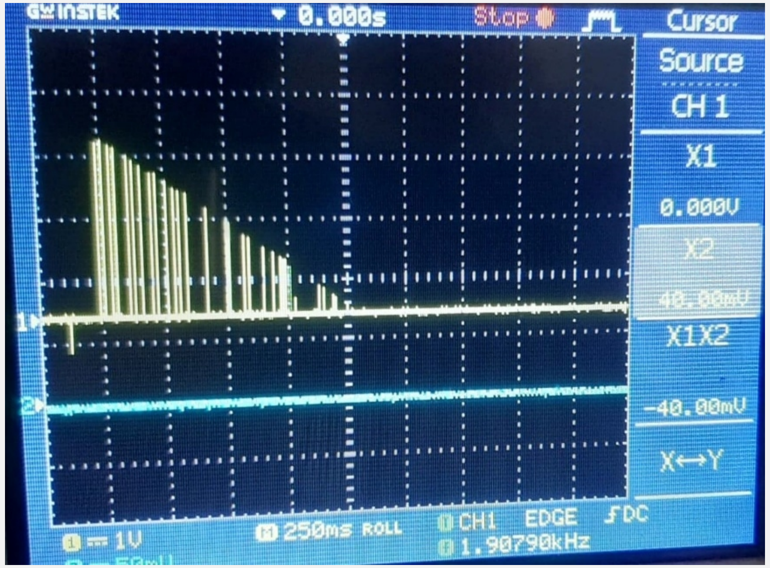
However, since the charging time increases it results in slower switching on of MOSFET. This increases the effective settling time.

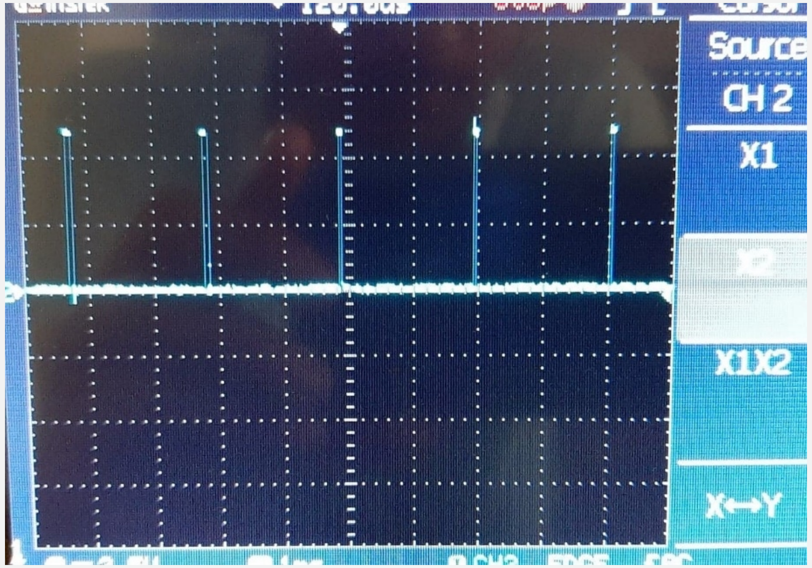
Thus, we must choose an **optimum** gate resistance and verify it through our results.

Other possible solution could be to use a RC snubber circuit to minimize the ringing peak. Careful calculations are done to calculate snubber resistance and capacitance.

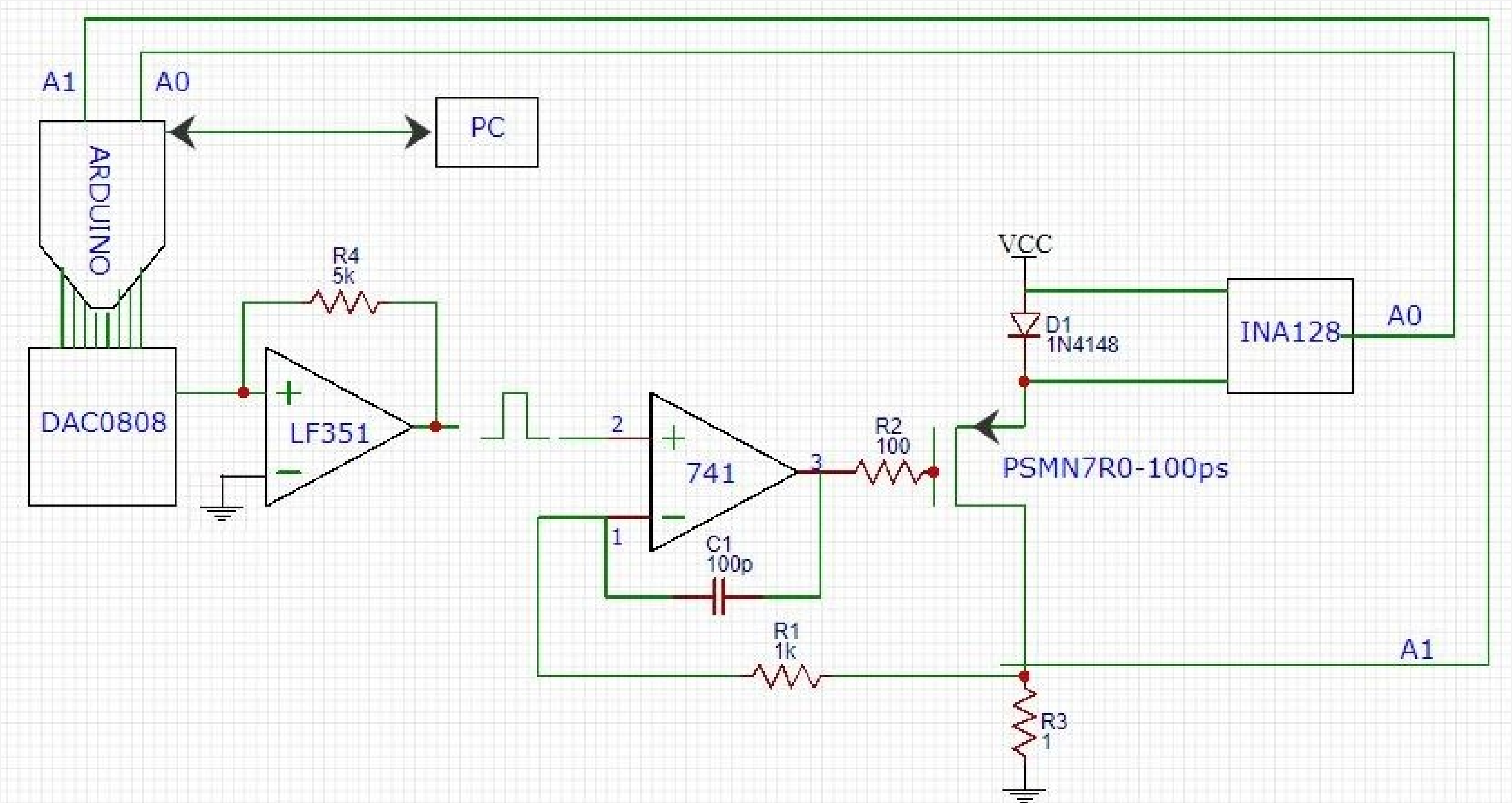
GENERATING CURRENT PULSES

Here we are forcing our circuit(feedback) to pass specific current using a very low resistor(R_c) and a power mosfet. The max. amplitude of current pulse is controlled by R_c. The opamp also acts as a buffer to avoid any loading at the input side.





CIRCUIT



A1
A0
PC
ARDUINO
R4
5k
DAC0808
LF351
2
741
3
1
C1
100p
R2
100
R1
1k
VCC
D1
1N4148
INA128
A0
PSMN7R0-100ps
A1
R3
1

SOME CHANGES TO CODE

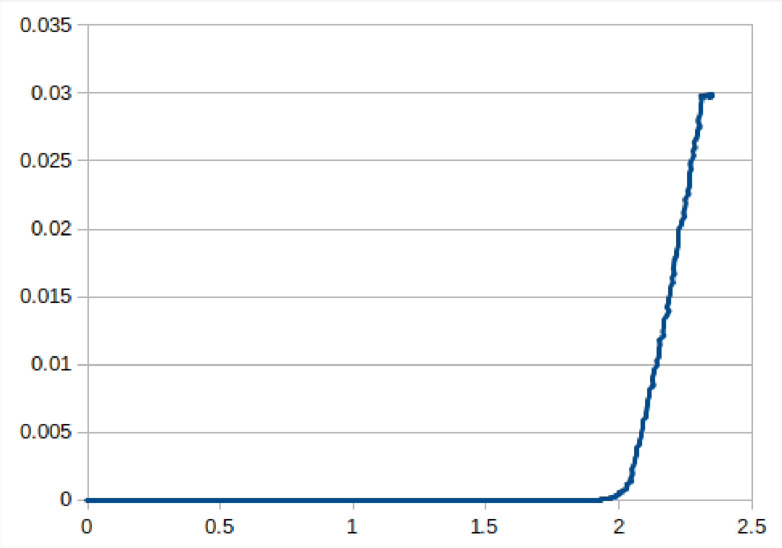
Previously, I was using interrupts to read ADC values. These interrupts would be active throughout the measurement, but we want this to be once in every pulse.

So I included some flag variables that allowed measurement once in every pulse. This leads to further delays, hence the premilenry pulses required ~400us width for better results.

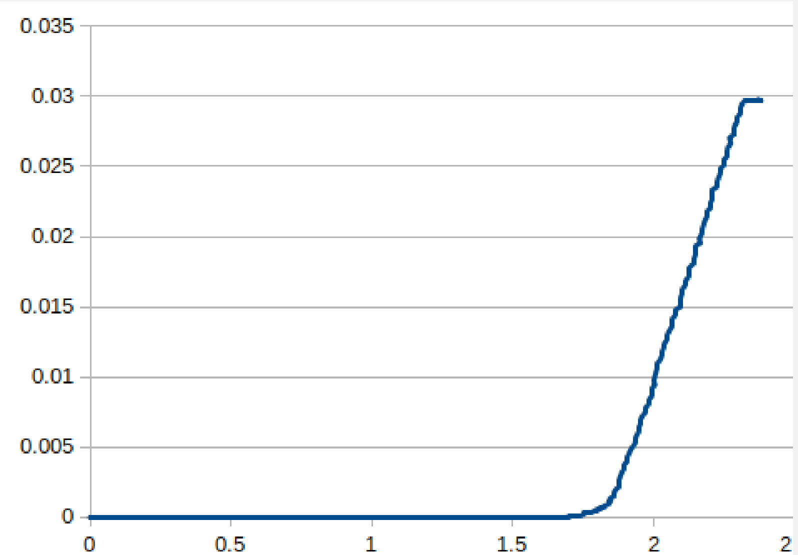
For reading voltage, I was using A0 pin of arduino. After reading the value, I switch to another pin(A1) within the code using the multiplexing feature of analog pins.

PLOTS OBTAINED

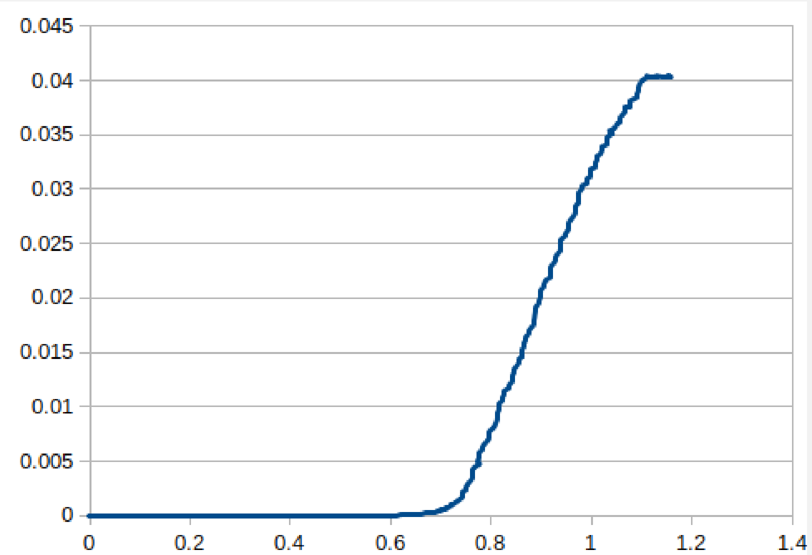
Yellow LED



Red LED



IN4001 diode



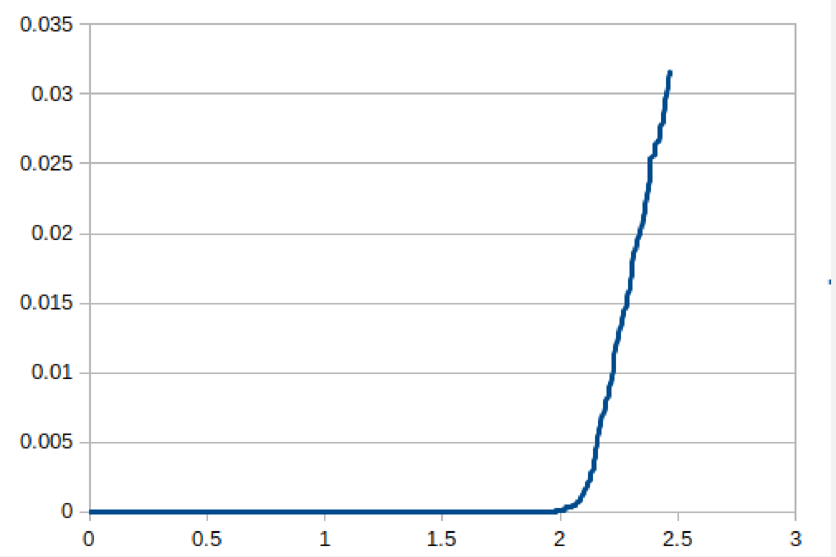
X-axis: Voltage(V), Y-axis:Current(A)

OPTIMIZING THE CODE

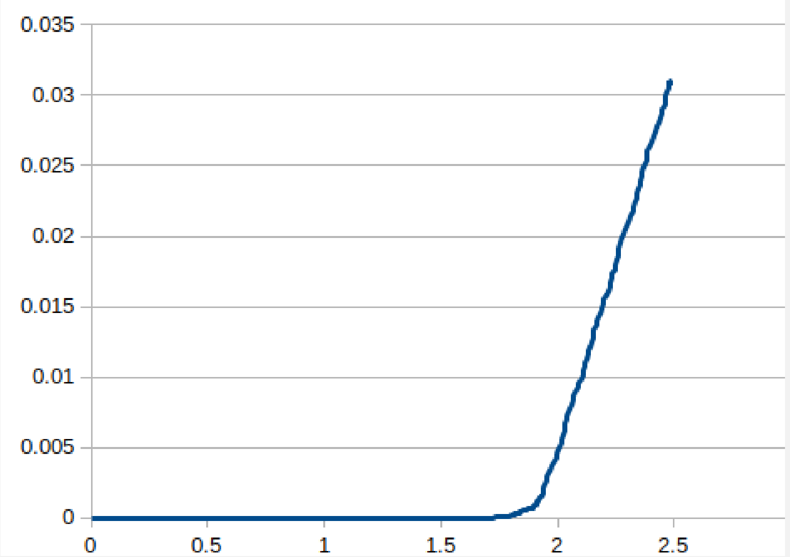
- Interrupt was replaced by function to reduce the complexity and use of variables in the program.
- Functions were explicitly called to configure a pin and read values from it, leading to a pulse width of mere ~**140us**.
- Some distortions were observed which could be removed by software manipulation.

PLOTS OBTAINED

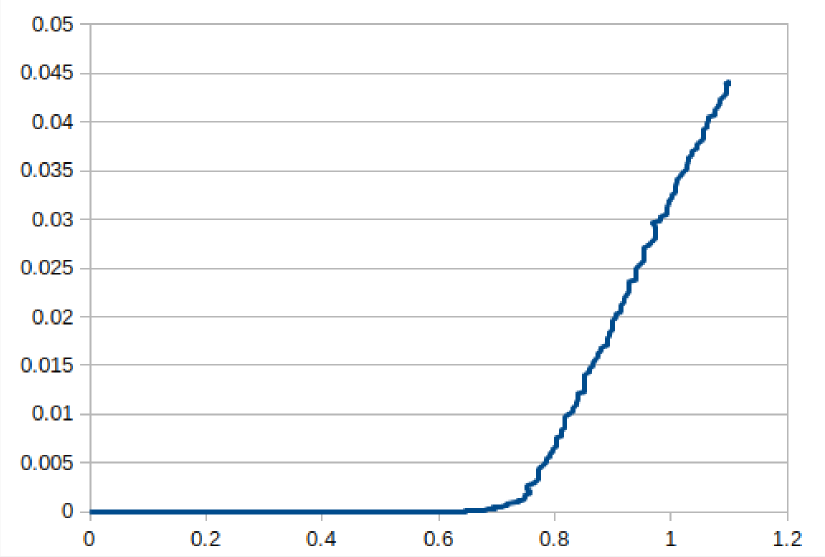
Yellow LED



Red LED



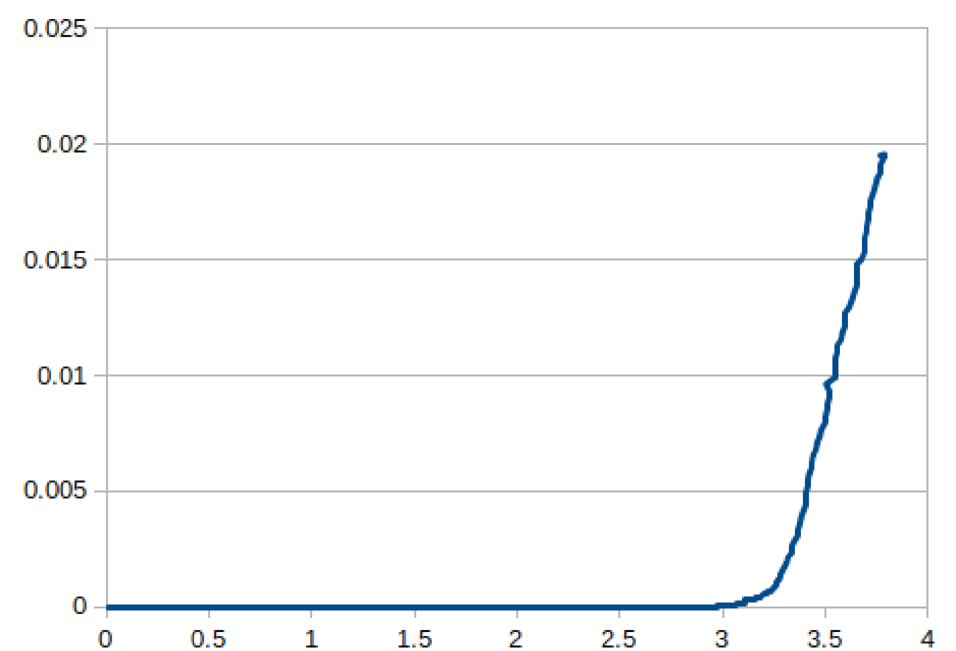
IN4001 diode



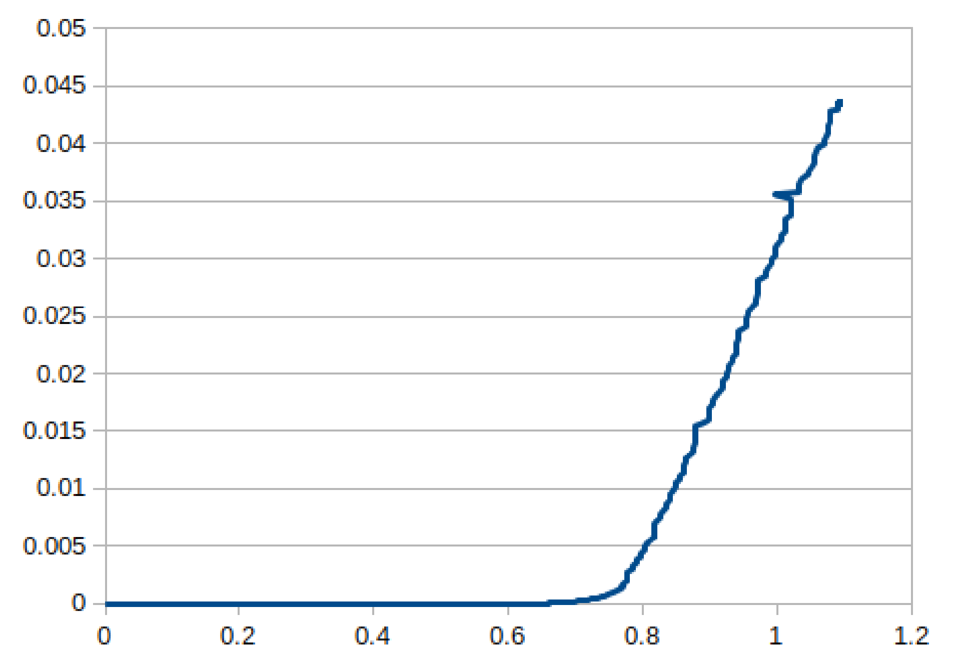
X-axis: Voltage(V), Y-axis:Current(A)

PLOTS OBTAINED

Green LED



Zener(5.6V) diode

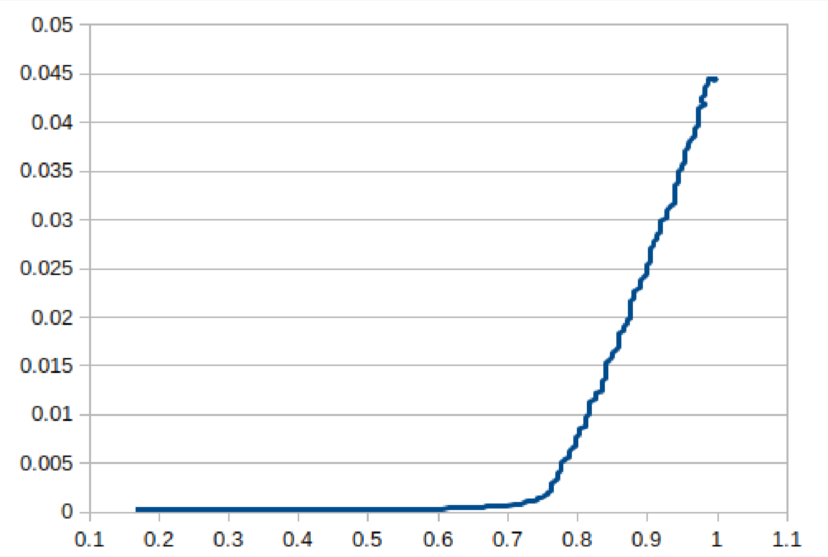


X-axis: Voltage(V), Y-axis:Current(A)

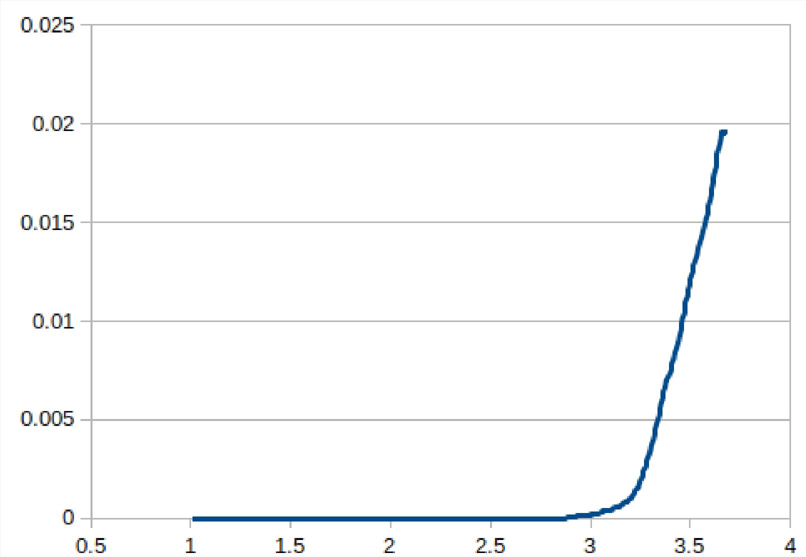
OPTIMIZATION CONTINUED…

I replaced even the functions and directly used the reading command through multiplexing. Some other optimizations lead to pulse width of ~**64us**

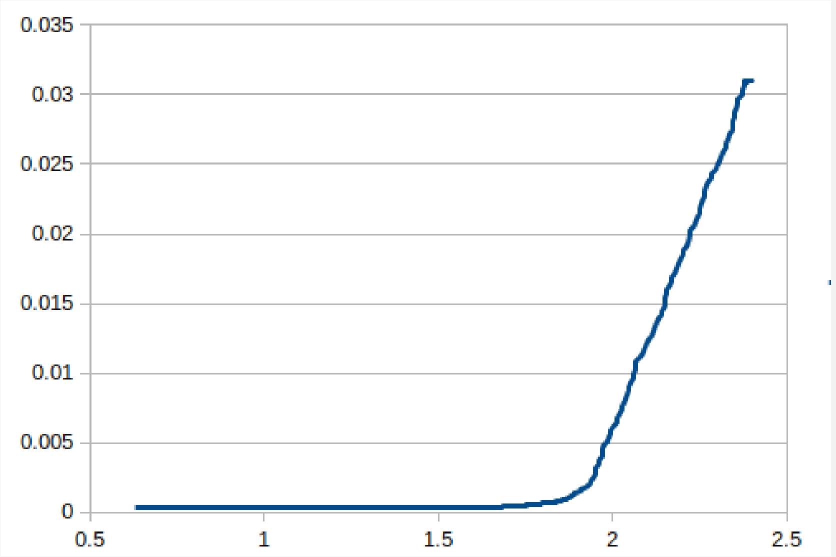
IN4001 diode



Green LED



Red LED



X-axis: Voltage(V), Y-axis:Current(A)

LET’S USE TWO PULSES !

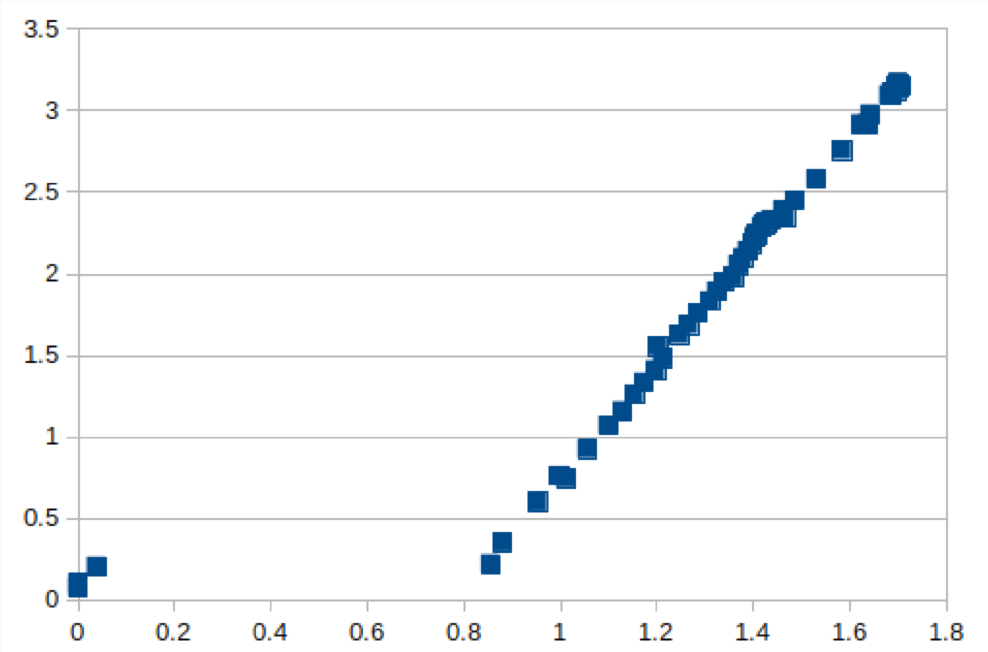
- New approach is to use 2 pulses instead of a single pulse for measurement
- Both the pulses would have similar amplitude, one would be used to measure voltage and other for the current
- As long as power supply of DAC module remains constant, we would get very similar pulses

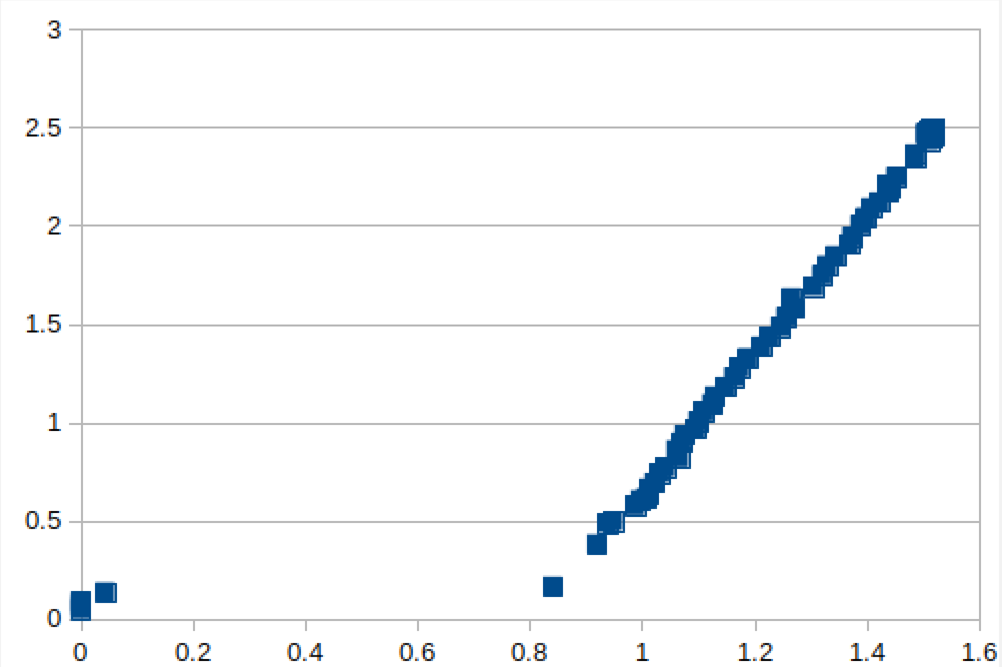
By doing this, we would reduce measurement time of one sample

For **further** experiment, we would be using power MOSFET to obtain higher currents and battery as the power source to reduce the dip in voltage.

PLOTS OBTAINED WITH IN4001 DIODE

(PULSE WIDTH : **56** MICRO SEC)





X-axis: Voltage(V), Y-axis:Current(A)

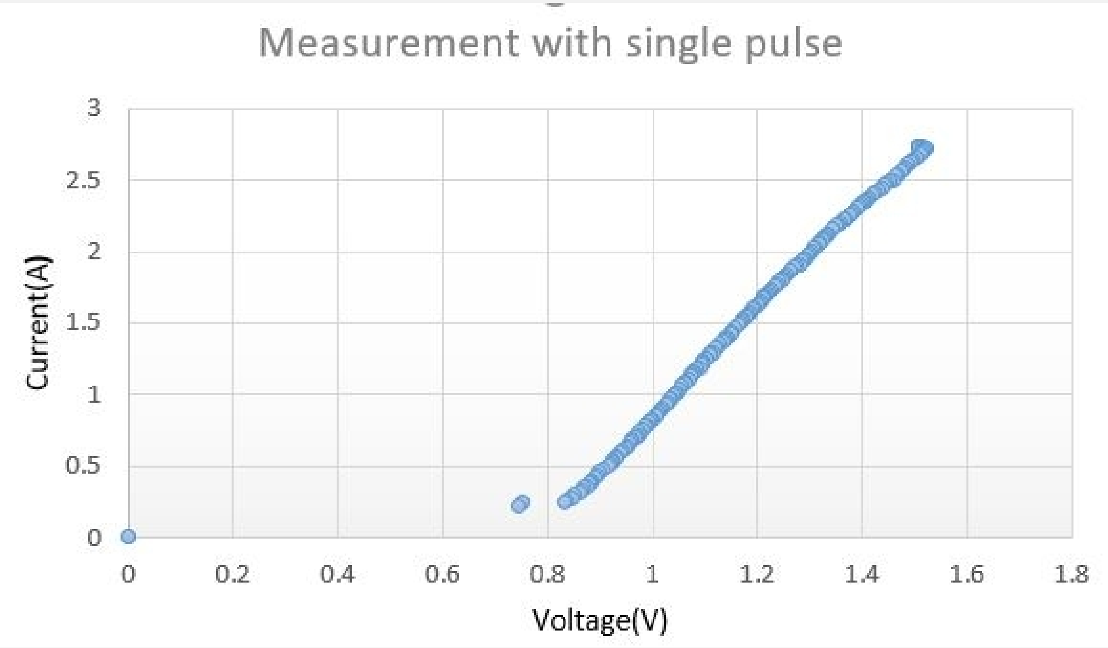
POINTS BEFORE KNEE POINT

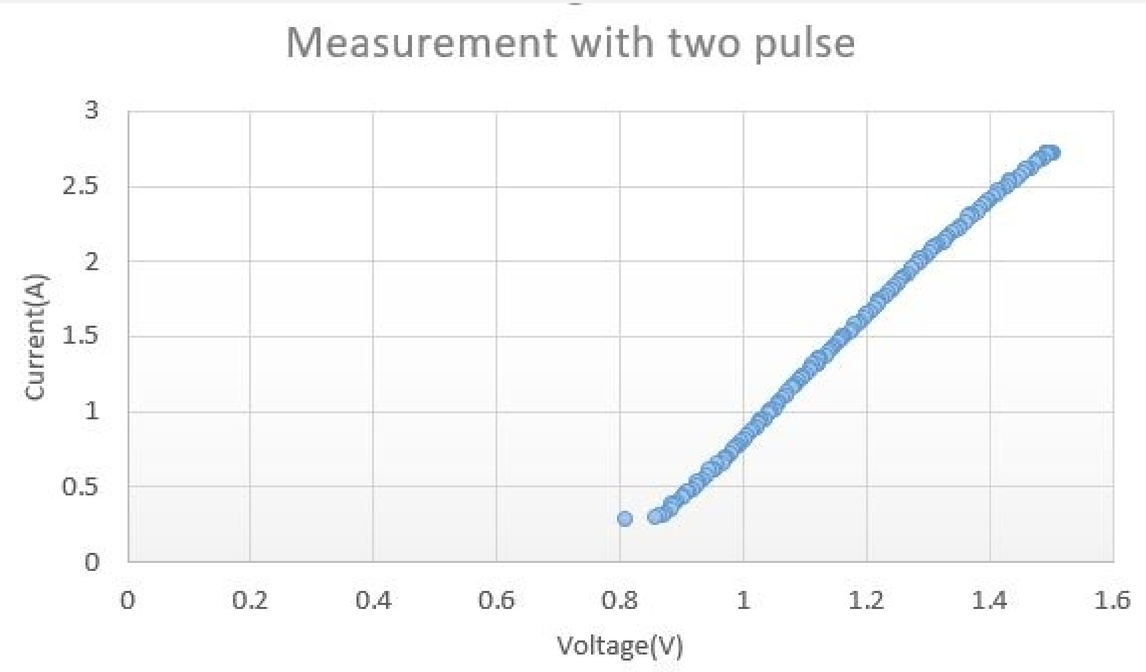
- The points before the knee point is not traced by the device.
- This happens because in our measurement, we supply a current and then measure voltage. (X-axis is current, Y-axis is voltage)
- The current applied is discrete, even for a small increase voltage rises very rapidly

For getting points before knee point, we could increase the precision bits of DAC or could use voltage pulses for measurement.

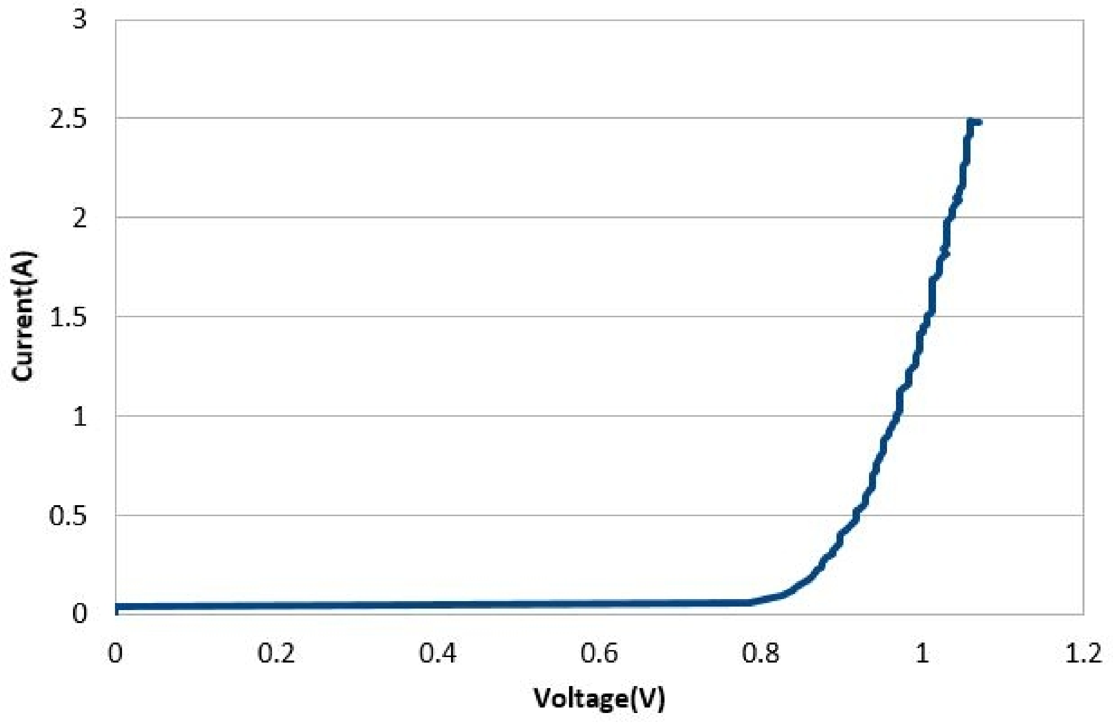
ARE TWO-PULSES PLOTS RELIABLE??

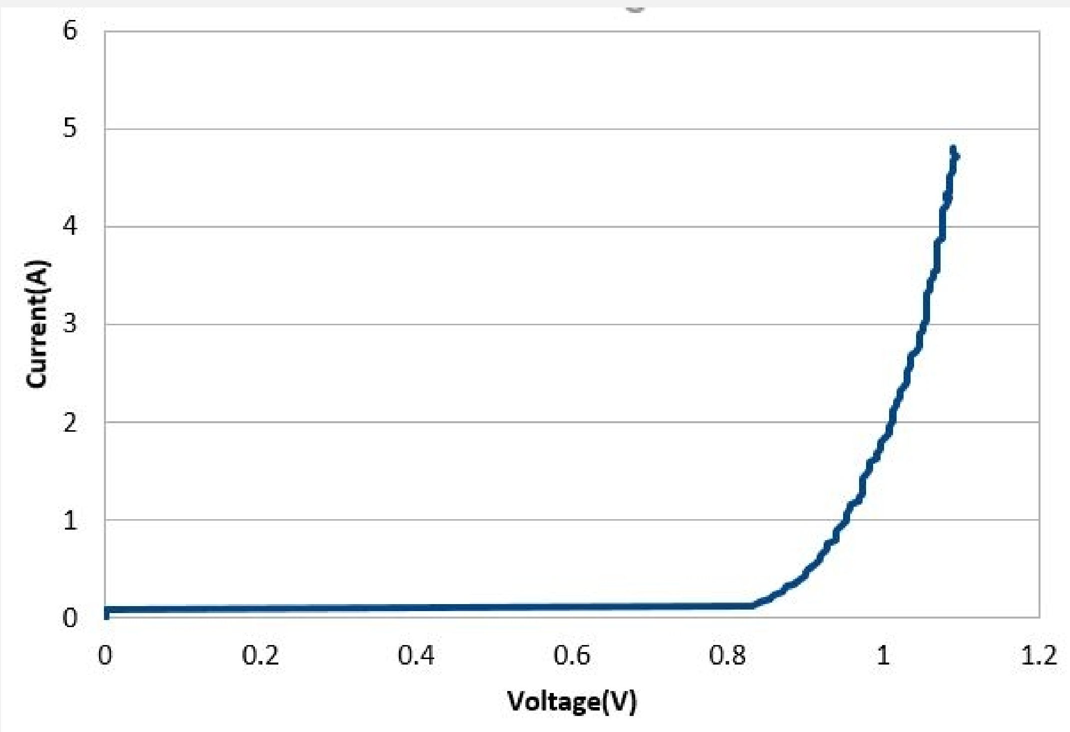
For ensuring whether this new method is doing well, I compared the results with single pulse plots.





SOME OTHER PLOTS WITH IN4001
(PULSE WIDTH : 120 MICRO SEC)

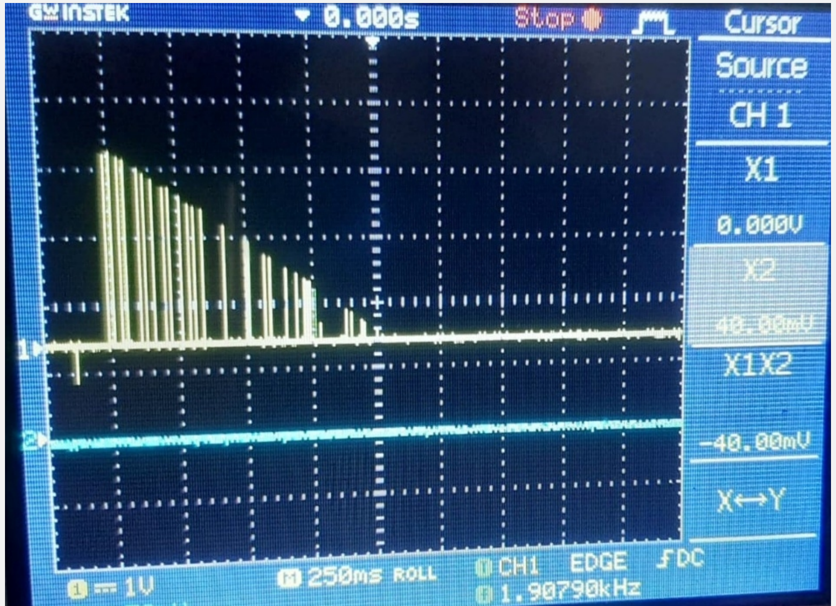




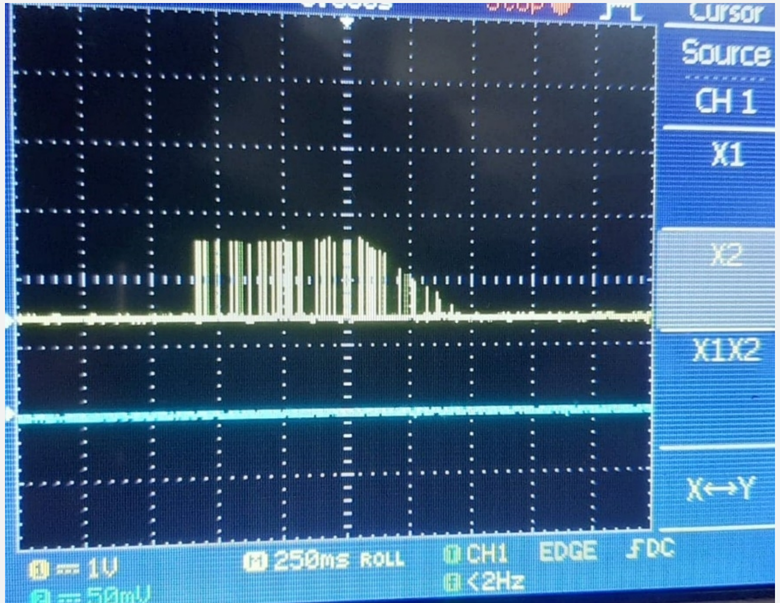
HAS THE BATTERY DRAINED??

- After few days, the pulses appearing at the shunt resistor were saturating
- Current higher than 3A couldn’t be drawn and it requires **wider** pulses
- Battery voltage went down to 3.56V -> discharged

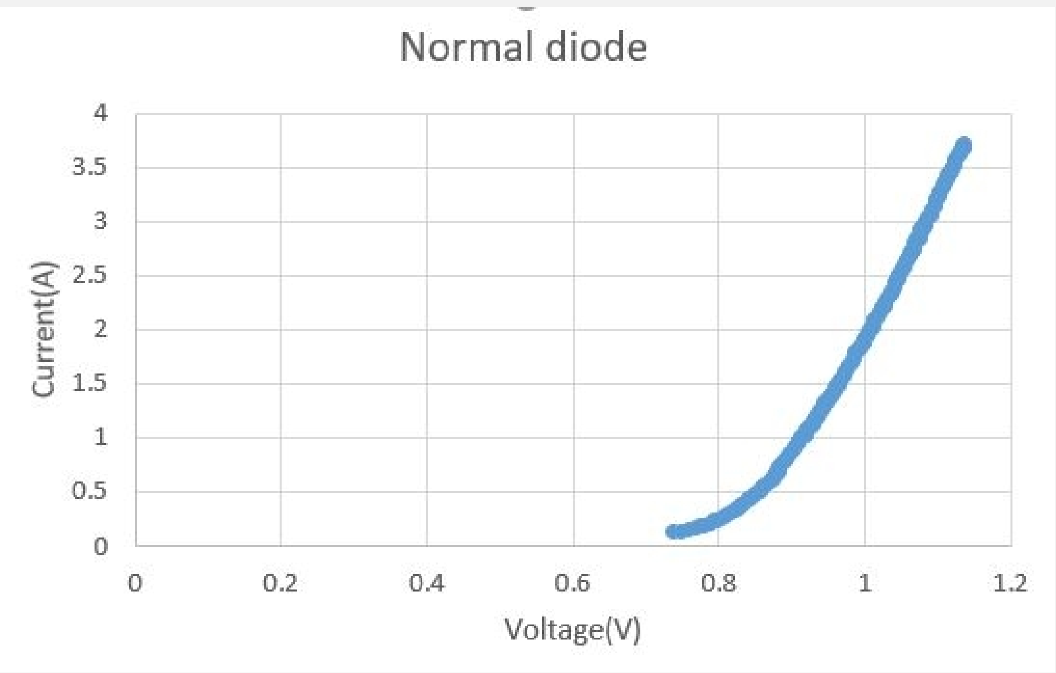
DAC output

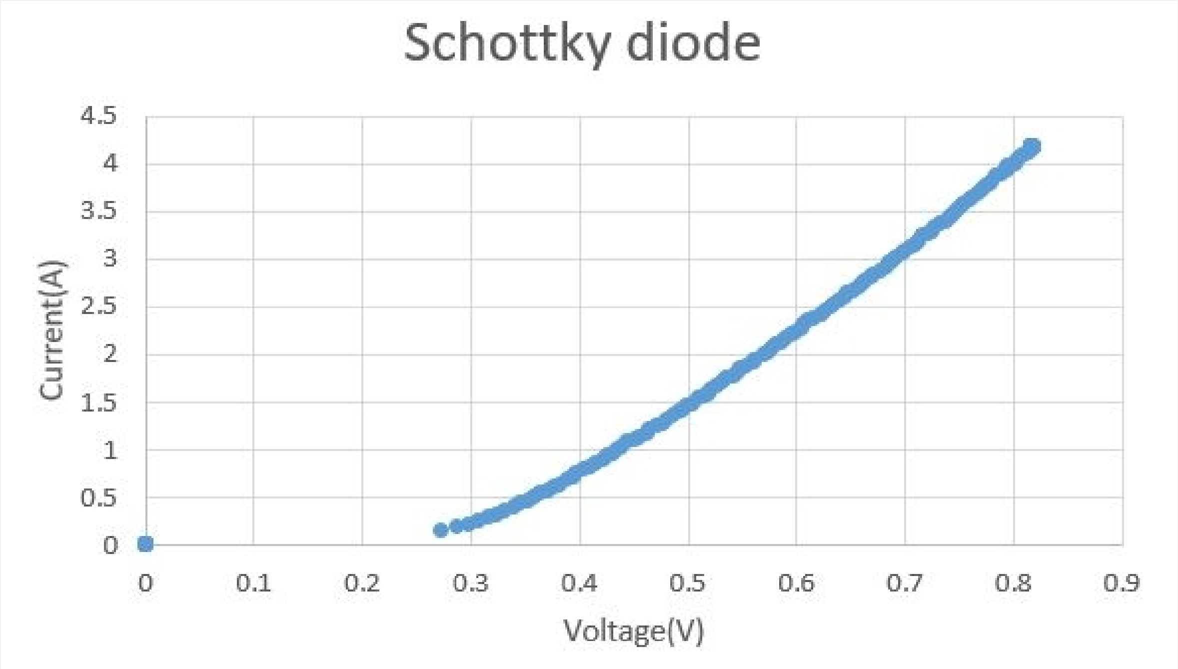


Pulse observed at shunt resistor



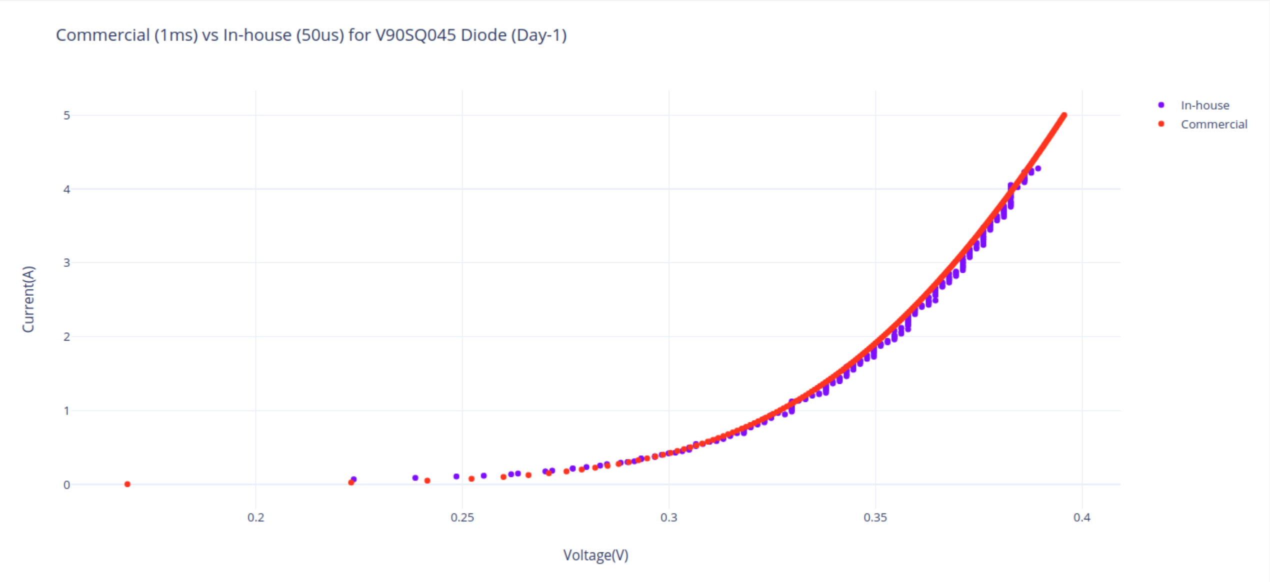
PLOTS WITH 65 MICRO SEC PULSE

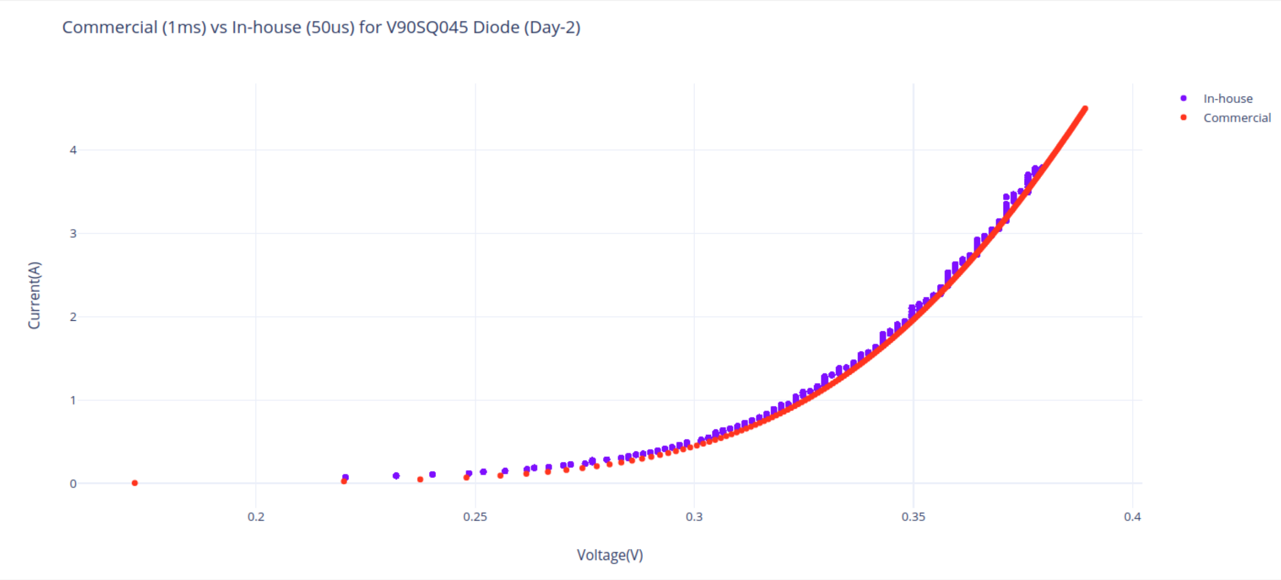




I-V curve for IN4001 exactly matched what is given in the datasheet

TESTING WITH PCB BOARD





The plots above shows the comparison between plots obtained through commercial IV tracer and In-house tracer (50us pulses) for two different days

FUTURE TASKS AND POSSIBLE IMPROVEMENTS

- Generating current pulses of shorter width and amplitude till 20A
- Improved battery performance could help us in achieving higher current
- Estimating the temperature profile of sample during measurement

THANK YOU !!!