EDL 2016-17 Project

Temperature Controller and IV Characterizer

Group Number: DD 16
Project Guide: Prof. Joseph John
TAs Associated: Ashish

Abstract

The aim of the project was to build a wide range temperature controller using Peltier Coolers to plot the IV Characteristics of two terminal devices at various temperatures. Along with the temperature controller, the IV Characteriser was also designed as part of the project. The aim was to learn about the working of the Peltier cooler and build a precise controller for it. The prototype design can achieve stability at temperatures ranging from -5^{0}C to 80^{0}C and the IV Characteristics can be plotted from a voltage range of -10 to +10 V and a current range of 20mA.

In this project, we have finally used a Bang-Bang Controller to control the temperature and get the temperature as mentioned above and the control occurs with the use of MOSFETs for ON-OFF control and Relays for direction control. We could achieve the desired temperature range and a full swing time of around 100 seconds. And the IV Characteristics were plotted for various devices with good precision.

OV Shashank 14D070021
Ayush Agrawal 14D070042
Yogesh Mahajan 14D070022
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

TABLE OF CONTENTS

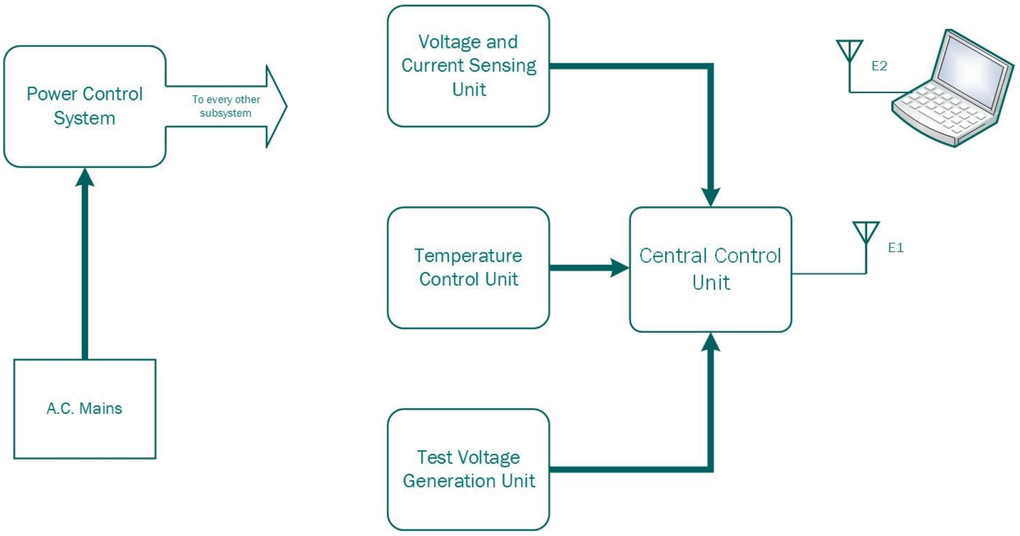
Chapter 1: Introduction

The initial aim of the project was to be able to control the Peltier Cooler to obtain a very wide temperature range of -10 to 80^0C in order to obtain the temperature variable characteristics of a diode.

Though in the research field, systems exist that provide temperature settable platforms, there is not an easy way to quickly characterize a devices' characteristic at various temperatures using the temperature settable platform systems. Usually characterization requires a user to snap the device to the platform and manually apply all the voltages and currents and at various frequencies which wastes a lot of the time which can be utilized in other productive tasks. Hence the main motivation for this project is to reduce the manual effort by designing an automated system with which, users could simply place their device in the product and quickly obtain the IV Characteristics of their device at various temperatures.

The project gives insights into design of systems to tolerate high currents of up to 4A as required by the Peltier and accurate and precise low power system design for the IV Characteriser. It also involves good amounts of requires system integration and coding for safety, speed and user friendly interface.

An overview block diagram of the project can be found below.



Specifications of the System

- Voltage Range: $\pm 10V$ (with variable precision)
- Current Range: $\pm 20mA$ (with variable precision)
- Temperature Range:$5^0C\ to\ 80^0C$ with precision of 1^0

In principle, any two-terminal device can be characterized in DC, by the system. But here are what the product is to guarantee for:

Supported 2-Terminal Devices are

- Diodes
- LEDs
- Zener Diodes
- Resistors

Chapter 2: Project Design

The four major sub-systems in the project are

- Peltier Temperature Controller Circuit
- IV Characteriser Circuit
- Power Supply System
- Microcontroller Programming and User Interface

Going into each sub-system we have

PELTIER TEMPERATURE CONTROLLER CIRCUIT

Precaution: The Peltier is a very delicate device and hence before handling a Peltier Cooler proper reading of the datasheet of the Peltier at hand must be performed so as to not burn one out.

It is important to understand that the Peltier Cooler is mainly a current controlled device and basically behaves as a variable resistance whose resistance depends on the temperature difference between its surfaces. It works by basically transferring heat from one of its sides to the other side and can work in a bi-directional fashion depending on the direction of current applied to it.

Hence it is necessary to maintain the side which is not being used, at a constant temperature to allow for the working of the Peltier Cooler. This calls for a good heat sink to be able to reach temperatures well below the room temperature.

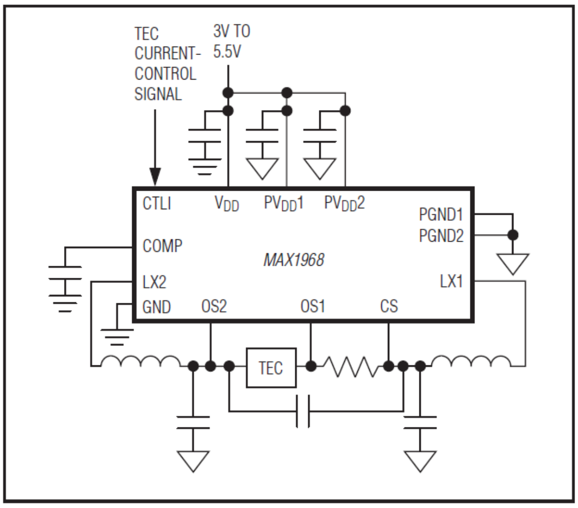
Based on our experimentation with the Peltier Cooler we could obtain the following results

- Without a heat sink, maintaining a temperature is very difficult as very quickly the heat begins to flow from the other side into the controlled side and hence an equilibrium is almost impossible to achieve.
- One Peltier Cooler rated at 6A, is unable to reach below 10 degrees at an approximate room temperature of 30 degrees Celsius if a current below 3A is provided to it. As also because of the low resistance (3 to 6 Ohms) of the Peltier device only a small voltage drop of about 6V is able to cause a current of 2A. This means that a power source of such ratings would be required to run a single Peltier cooler.
- Using 2 or 3 Peltiers in a series combination is best option among all with the Peltiers lying physically in stacked fashion so that the bottom ones are able to support the topmost Peltier in its temperature control by sucking heat from its other surfaces.

Using a stack of Peltiers in series also allows for utilisation of battery kind of voltage sources which usually provide high currents at voltage ranges of 12 to 24V and hence the full voltage range of the battery can be utilised.

For the circuit design

- It is very difficult to implement an analog kind of control system for the Peltier at such high currents because of the requirements of resistors in the current path which waste large amount of power.
- Instead the best control mechanism is to use a PWM control of the voltage across the Peltier



Another issue is that it is not advisable to provide a PWM kind of current to the Peltier and hence it requires a certain amount of smoothing using LC Filters (LC specifically to avoid the use of Resistors in the high current path) and a current sense mechanism if a precise current control is required. This leads to a requirement of two control loops, one for the current (using voltage) and the other for the temperature (using current).

Such implementations can be validated by looking at industry standard chips by manufacturers and one great example for the same is MAX1968-9 as can be seen in the adjacent figure.

Hence, we have the specifications of the system as

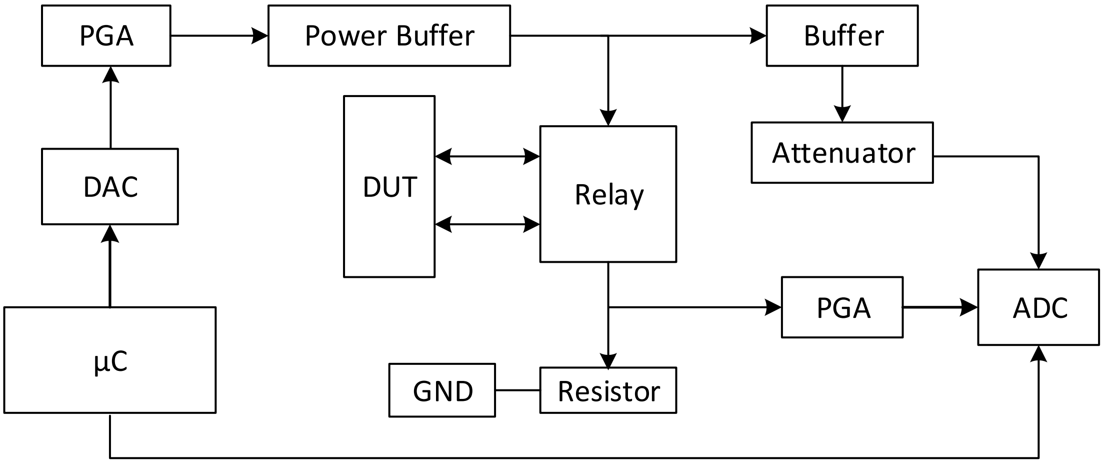
- Output Voltage Range of $\pm 12V$
- Output Current Range of $\pm 4A$
- Minimal Output Voltage and Current Ripple
- Controllable output current with an approximate precision of 100mA
- Overcurrent Signal and Protection

We went on to implement an H-Bridge Circuit of our own, and more about the specifics of the implementation will be discussed in the Project Implementation Section.

IV CHARACTERISER CIRCUIT

The circuits required for this purpose are straight-forward. It requires a programmable voltage source at the input to provide a wide range of voltage to the device. This is followed by a control sensing mechanism which most often performed using a small resistor, which does much reduce the range of the voltage drop across the device and can limit the current flow through the device in the case of a malfunction.

Appropriate arrangements must also be made while taking into consideration the precision required and the larger than DAC supply-able voltage range. A block diagram can be viewed as follows



A relay may be used for providing the switching of voltages from positive to the negative direction without compromising on the range and precision of the voltage as supplied by the DAC and worrying about having to read negative voltages from the circuit using ADCs.

Note: PGA – Programmable Gain Amplifiers; DUT – Device Under Test; Power Buffer – Required to provide large currents, basically a power Op-Amp in voltage follower configuration.

The specifications (absolute maximum) chosen for this system was

- Device power supply range of $\pm 10V$
- Device current range of $\pm 200mA$ [Aim]; $\pm 20mA$[Achieved]
- Voltage precision of 5V per 12 bits
- Variable Current Precision

POWER SUPPLY SYSTEM

The basic required specifications for this system are fixed by the other systems but because of our implementation decisions the following are the necessary requirements

- Dual supply of $\sim \pm 12V$ for the Operational Amplifiers (mostly required for any implementation)
- $+5V$ supply for DAC and ADC
- $\sim + 12V$ power voltage supply for the Peltier Cooler
- The exact values are not necessary because appropriate voltage references have been provided when accuracy is necessary. Exact precision is unnecessary even for the Peltier because the PI algorithm can tolerate certain amount of system parameter errors.

The design involves the choice of regulators or DC-DC converters based on efficiency and power requirements and necessarily a buck-boost converter if a negative voltage is required to be produced from a single positive power supply unless a centre-tap is possible when using a transformer.

The DC power supply could be either batteries to allow for a portable device or could be an SMPS. Batteries have issues with loosing charge quickly when having to supply power hungry devices like the Peltier. Hence if a wired product is feasible one can forgo the trouble of having to charge the batteries multiple times by replacing it with SMPS which provides a constant voltage over a large range of current and time.

MICROCONTROLLER PROGRAMMING AND USER INTERFACE

The choice of the microcontroller is mainly based on the implementation requirements but the speed requirements are set by the kind of computation to be done in the microcontroller.

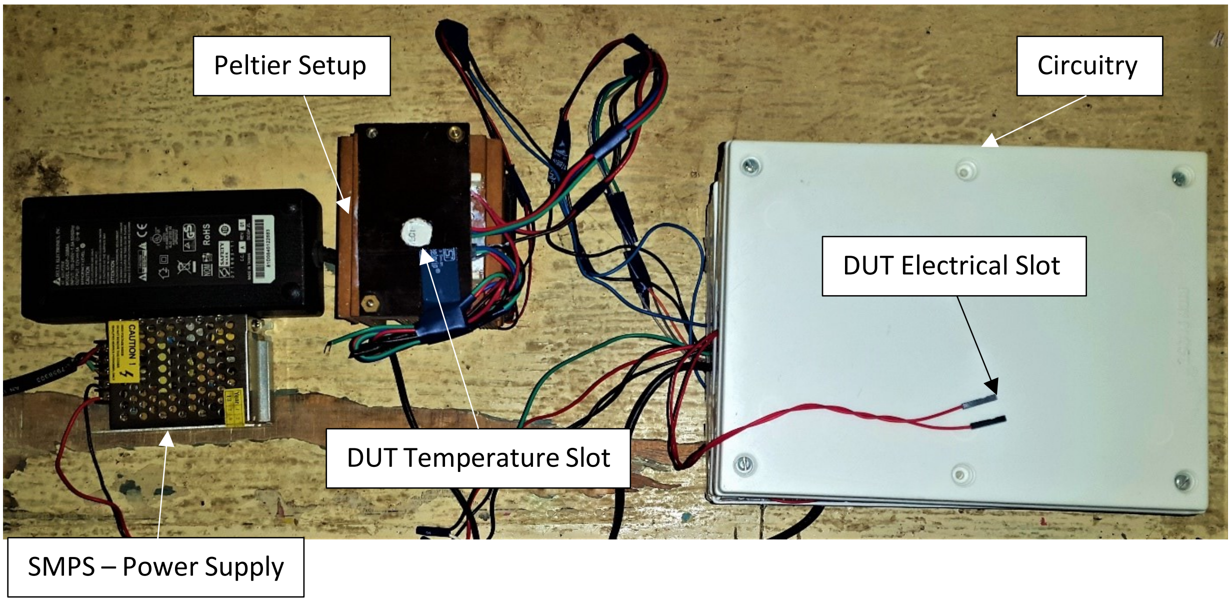
In this project, the microcontroller performs the following tasks

- Contains the PID controller for the Peltier Control
- Automated IV Characteriser
- Communication set up to communicate with the ICs and the Computer

The main purpose of the GUI is to provide a user-friendly interface for easy no-brain control of the product, with which the user can forget about the technical details of the product and simply provide the required parameter values that matter and the GUI then handles the sending of the necessary commands and displays the information obtained from the product in an easily understandable fashion.

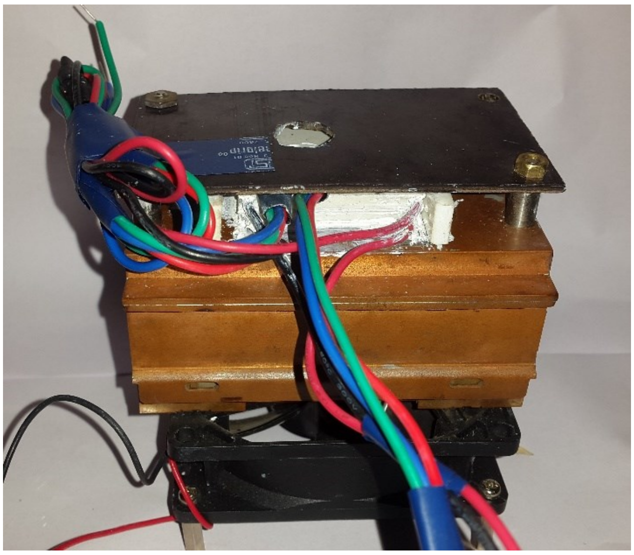
Chapter 3: Project Implementation

This section provides the specific implementation choices and design based on the overall Project Design as explained in the previous Chapter. The below is the final demonstration setup of our project with all the PCBs and devices apart from the Peltier Cooler setup encased safely in boxes.



As aforementioned we know that four subsystems in the project and we now one by one look at the prototype implementation of these systems

PELTIER COOLER SETUP



As can be seen in the final implementation image, the setup used included a stack of two Peltier Coolers as was chosen to be optimum to work using a 12V Supply and provide great cooling.

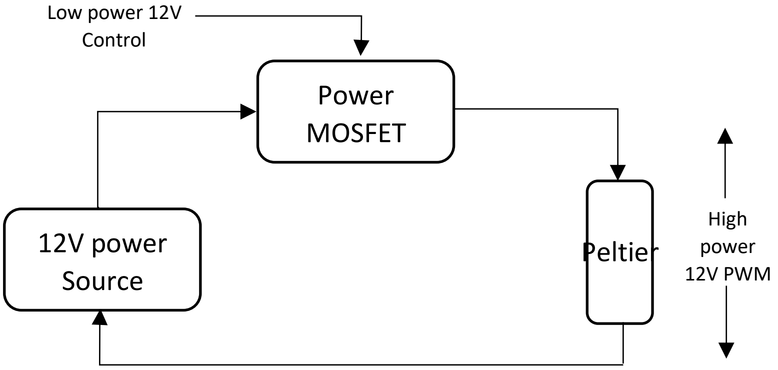
Proper isolation has been provided from all sides of the Peltier apart from a small opening to insert the device under test. This is necessary to be able to provide cool temperatures by blocking the loading which occurs due to the environment.

Also, the large heat sink and the fan underneath are also necessary to maintain the temperature of the bottom side of the Peltier cooler so that the top side can be cooled to very low temperatures.

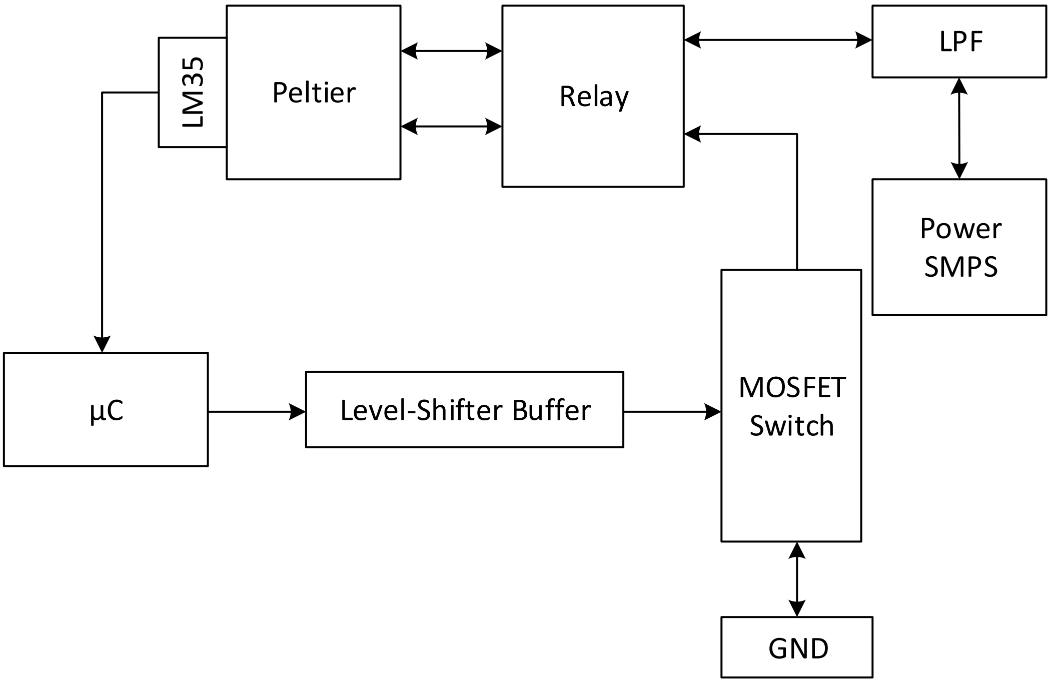
The extra wires are for an extra backup LM35 which was present to cover for a failure of the other LM35 temperature sensor being used. The other wires also include the wires from the Peltier Cooler which are shorted together to provide a series connection among the two Peltier Coolers.

PELTIER TEMPERATURE CONTROLLER

The MOSFET is used as a switching device for translating a low power PWM signal to an equivalent high power signal. The bi-directional requirement of the circuit is fulfilled with the help of two Relays connected in parallel. This circuit can be understood in three steps:



1. Understanding the MOSFET operation: We have used IRFZ44N as the power MOSFET. It switches on at gate voltages greater than 10V.
2. The micro-controller that we use provides 5V PWM. Therefore, we shifted the voltage level to 12V using a level shifter buffer.
3. This MOSFET is now connected in series to the TEC and the power supply. As the drain-source resistance of the MOSFET is very small, the power loss due to the same is also very small for a current flowing through the circuit. Also, we can assume that it would not disturb the set point for the TEC. The final flow chart for the same is as:



Note: As is mentioned in the next chapter, because of issues with the PWM control and the relay creating noise in the 5V system, we were unable to implement the PWM controller, instead using the same above circuitry we implemented a Bang-Bang Controller which turned out to be sufficient for our implementation (for more details refer Chapter 4)
Also, we tested the PWM circuitry on a breadboard using isolated power supplies and we good obtain great results and the choice of the PI parameter to control the PWM duty cycle was as follows:

- The Proportional component was chosen such that the Peltier Cooler would work in full throttle whenever it is 5^0C away from its set-point. This ensures that the Peltier Cooler would reach its set point fast and not unnecessarily get slowed down.
- This basically was implemented such that $Max\ Duty\ Cycle\ =\ Error(@5^0) * K_P$

- The value of the I parameter was tweaked by trial and error to get a low overshoot on the temperature graph.

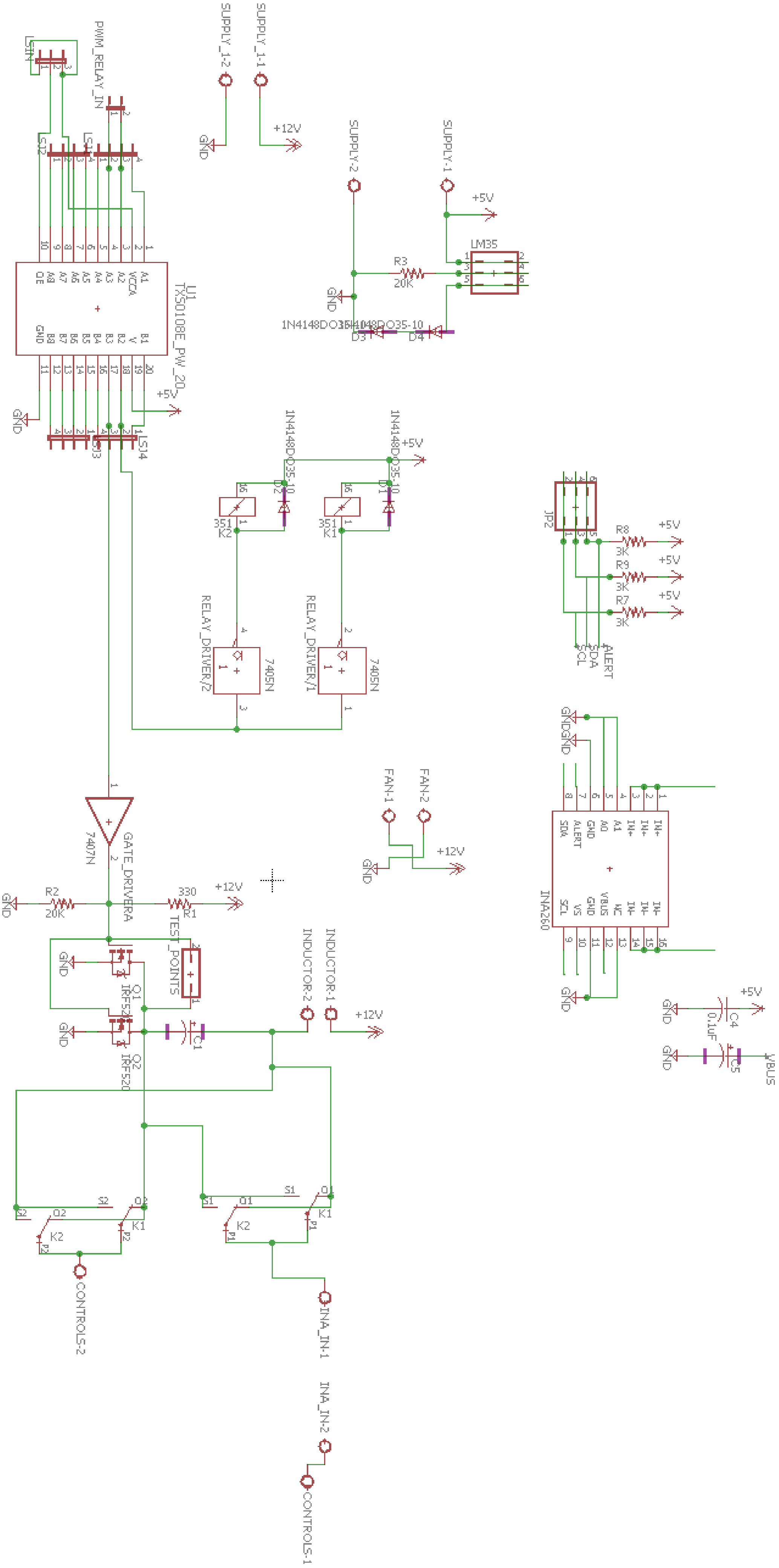
The values chosen for the Arduino Mega based system are as follows

$$K_P = 40; K_I = 0.0001$$

Note: The value of the integral was updated approximately every 10ms.

The following are the devices and components used in its implementation. These are pertaining to the schematic of the implementation circuit as shown on the schematic figure.

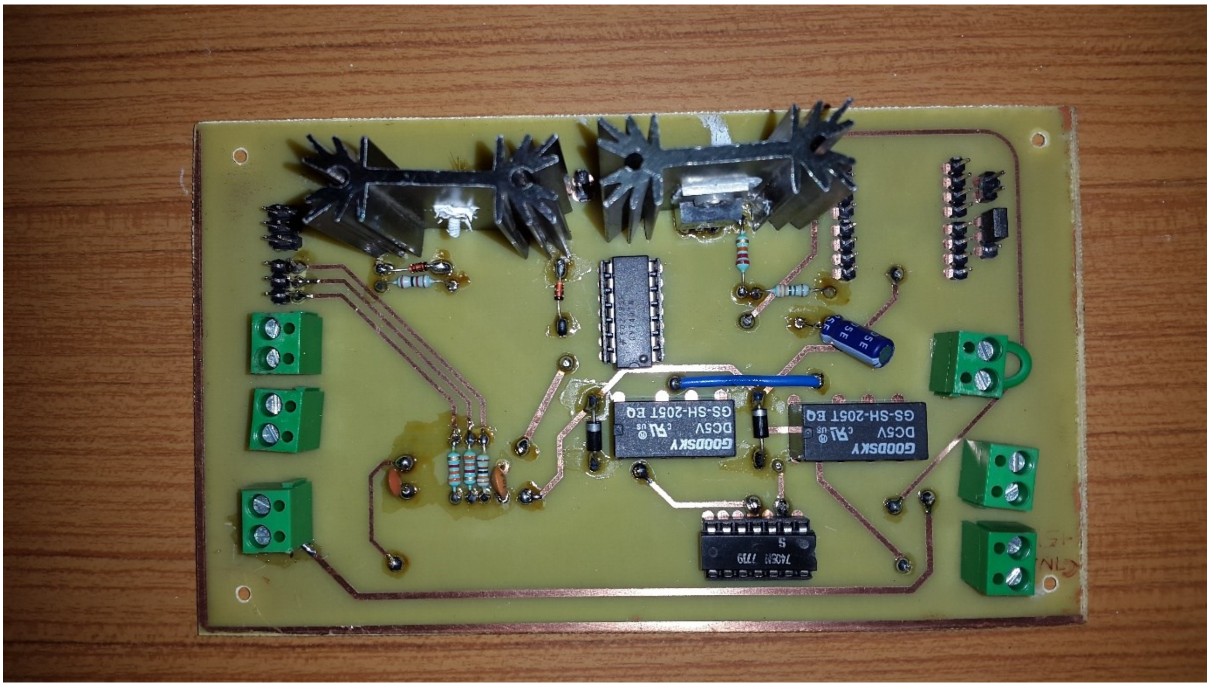
Bill of Materials		
Component	Value (as Applicable)	Utility
C1	100uF	Low Pass filter capacitor
C2/C3	0.1uF	Decoupling capacitors
CONTROLS	Screw Connector	Control pins for switching relays
D1/D2	1N4004	Relay Diode
D3/D4	1N914	LM35 diode for biasing
FAN	Screw Connector	Power pins for the Fan
GATE_DRIVER	7407N	
INA_IN	Screw Connector	Input for INA for current sensing
INDUCTOR		
JP1	Jumper Wire	
K1	351	
K2	351	
LM35	Jumper Wire	Input from LM35 to the board
LSIN	Jumper Wire	
LSJ1/ LSJ2/ LSJ3/ LSJ4	Jumper Wire	
PWM_RELAY_IN	Jumper Wire	
Q1	IRF520	Power n-MOSFET
Q2	IRF520	Power n-MOSFET
R1	330	Pull-up resistor
R2/R3	20K	Pull-down resistor
R4/R5/R6	3K	Pull-up resistors
RELAY_DRIVER	7405N	Open collector buffer for driving the relays
SUPPLY	Screw Connector	5V dc power supply
SUPPLY_1	Screw Connector	+12V dc power supply
TEST_POINTS	Jumper Wire	Testing point
U$1	INA260	INA current sensing IC
U1	TXS0108E_PW_20	



LSIN
PWM_RELAY_IN
SUPPLY_1-2
SUPPLY_1-1
GND
+12V
SUPPLY-2
SUPPLY-1
+5V
LM35
R3
20K
GND
1N4148DO35-10
1N4148DO35-10
D3
D4
LS2
LS1
U1
TXS0108E_PW_20
A1
VCCA
A2
A3
A4
A5
A6
A7
A8
OE
B1
V
B2
B3
B4
B5
B6
B7
B8
GND
+5V
GND
LS3
LS4
1N4148DO35-10
1N4148DO35-10
+5V
D2
D1
351
K2
351
K1
RELAY_DRIVER/2
RELAY_DRIVER/1
7405N
7405N
JP2
R8
3K
R9
3K
R7
3K
+5V
+5V
+5V
SCL
SDA
ALERT
GND
GND
IN+
IN+
IN+
A1
A0
GND
ALERT
SDA
IN-
IN-
IN-
NC
VBUS
GND
VS
SCL
INA260
GND
GATE_DRIVERA
7407N
R2
20K
GND
330
R1
+12V
TEST_POINTS
FAN-1
FAN-2
GND
+12V
GND
Q1
IRF520
GND
Q2
IRF520
C1
INDUCTOR-2
INDUCTOR-1
+12V
+5V
C4
0.1uF
GND
GND
C5
VBUS
S2
Q2
G2
K2
P2
K1
P2
O2
S1
O1
G1
K2
P1
K1
P1
O1
CONTROLS-2
INA_IN-1
INA_IN-2
CONTROLS-1

- There are multiple level shifter buffers on the board. The conversion required is from 3.3V level (Tiva C) to 5V level (Relay Driver) and then from 5V to 12V level.
- We forgot to add power source on the PCB design for the INA, so we decided to use the external INA test board that was developed during the testing phase.
- The particular choice of the power MOSFET was due to the fact of easy availability in the WEL lab. Also, n-MOS was used instead of p-MOS due to low drain-source resistance of the n-MOS as compared to the p-MOS. We also did not make our own customized H-bridge using power p-MOS and n-MOS for the same reasons.
- As very high current is flowing through the MOSFETs, we used big heat sinks to account for the heat losses generated during the operation for further improving the performance.

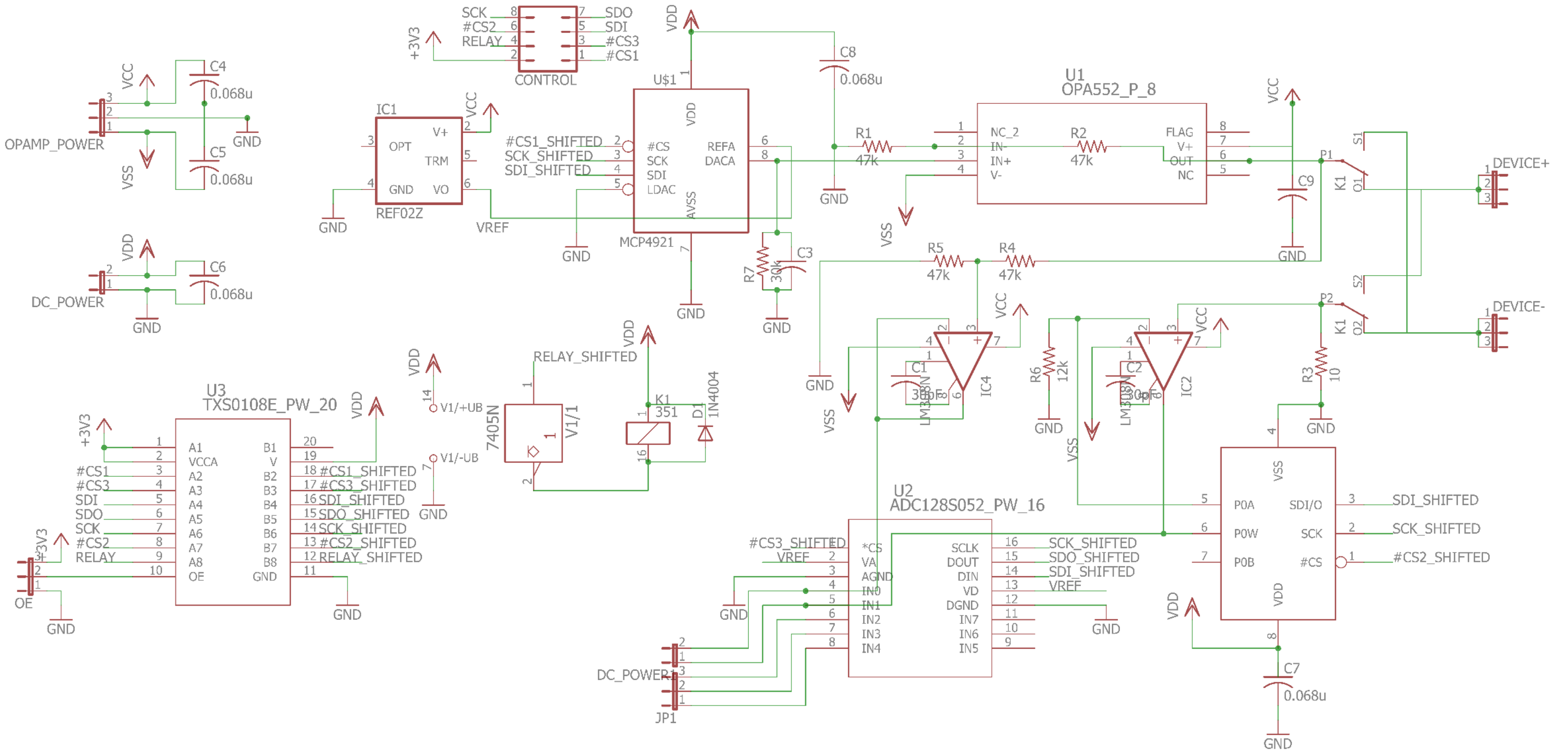
Following is the image of the printed board



IV CHARACTERISER

The following are the devices and components used in its implementation. These are pertaining to the schematic of the implementation circuit as shown on the schematic figure.

Bill of Materials		
Component	Value (as Applicable)	Utility
C1	30pF	Compensation Capacitor for Precision Operational Amplifier
C2	30pF	Same as C1
C3	20pF	Loading Capacitor for DAC
C4/5/6/7/8/9	0.068u	Decoupling Capacitor
CONTROL	8-Pin Jumper	SPI and Relay Control Pins and Level-Shifter 3.3V Supply
D1 (Diode – 1N4004)		Relay Inductor reverse current flow diode
DC_POWER	2-Pin Jumper	5V Power Supply
DEVICE+		DUT Terminal
DEVICE-		DUT Terminal
IC1 (REF02Z)		Voltage Reference for ADC and DAC
IC2 (LM308N)		Precision Operational Amplifier for Current Sense Amplification
IC4 (LM308N)		Precision Operational Amplifier for Voltage Reading buffer
DC_POWER1	5-Pin Jumper	ADC Channel Out
K1 (Relay)		Relay for device direction switching
OE	3-Pin Jumper	Output Enable select for Level Shifter
OPAMP_POWER	3-Pin Jumper	Dual Power supply for the Operational Amplifiers
R1/2	47k	x2 Amplification Feedback Resistors
R3	10	Current Sense Resistor
R4/5	47k	2x Attenuation Resistor Divider Resistors
R6	12k	Variable Gain stage feedback Resistor
R7	30k	DAC Loading resistor
U$1 (MCP4921)		Digital to Analog Converter
U$2 (MCP4161)		Digital Potentiometer for current sense variable gain
U1 (OPA552)		High Power Operational Amplifier to provide currents up to 200mA
U2 (ADC128S052)		Analog to Digital Converter
U3 (TXS0108E)		Level Shifter (3.3V to 5V)
V1 (7405N)		Open Collector Inverting Buffer as a Relay Driver


DEVICE+
DEVICE-
K1
S1
S2
O1
O2
P1
P2
U1
OPA552_P_8
FLAG
V+
OUT
NC
NC_2
IN-
IN+
V-
R1
R2
R3
R4
R5
R6
R7
47k
10
12k
30k
C1
C2
C3
C4
C5
C6
C7
C8
C9
0.068u
30pF
IC2
IC4
LM308N
U$1
MCP4921
VDD
REFA
DACA
#CS
SCK
SDI
LDAC
AVSS
IC1
REF02Z
OPT
TRM
GND
VO
VREF
CONTROL
SCK
#CS2
RELAY
SDO
SDI
#CS3
#CS1
+3V3
VCC
VSS
OPAMP_POWER
DC_POWER
U2
ADC128S052_PW_16
*CS
VA
AGND
IN0
IN1
IN2
IN3
IN4
IN5
IN6
IN7
SCLK
DOUT
DIN
VD
DGND
#CS3_SHIFTED
SCK_SHIFTED
SDO_SHIFTED
SDI_SHIFTED
JP1
POA
POW
POB
SDI/O
#CS2_SHIFTED
#CS1_SHIFTED
RELAY_SHIFTED
K1
351
D1
1N4004
V1/1
7405N
V1/+UB
V1/-UB
U3
TXS0108E_PW_20
A1
VCCA
A2
A3
A4
A5
A6
A7
A8
OE
B1
V
B2
B3
B4
B5
B6
B7
B8

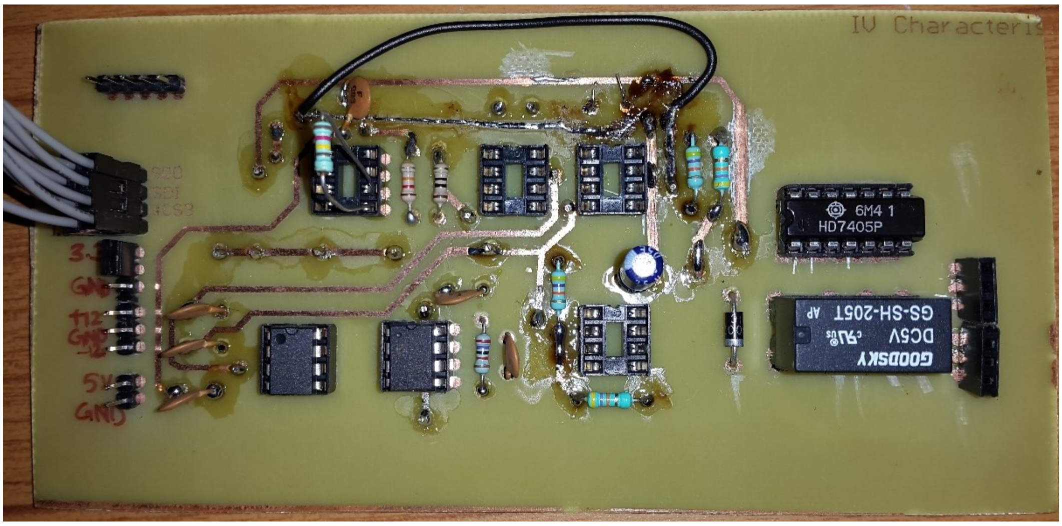
Details of the Implementation:

- REF02 is used at the reference voltage supply to ensure accuracy of operation of the ADCs and DACs
- The circuit is designed to be able to supply 10V power to the DUT in both directions with the use of the relay and a fixed x2 amplification stage combined with the power buffer in a non-inverting amplifier configuration which can supply up to 200mA in normal operation. *Note: In the final implementation, this IC was replaced by a normal Operational Amplifier because the chosen power Op-Amp did not work well under a gain of 2 and according to the data sheet, additional handling circuitry must be added which was missed out in the design. Hence our final design only supports a current of 20mA.*
- The appropriate Relay protection circuitry is to be used which involves an open collector inverting buffer 7405 whose pull up at the output is ensured by the inductor which is connected to 5V. The diode is added to carry the current flowing through the inductor when the Relay is suddenly turned off and the connection is cut-off.
- A fixed attenuation of 2x was applied at the device voltage supply. The attenuation is fixed because both the DAC and ADC have a precision of 12 bits and hence higher precision would not mean anything as the step size is as small as it can be measured and supplied.
- A variable of gain using a Digital Potentiometer is applied to the current sense resistor of 10 Ohms so that higher precision can be achieved in the measurement of the current.
- A resistance as small as 10 Ohms is chosen so as to not cause a very large voltage drop which might reduce the range of the voltage providable to the DUT
- A level shifter is necessary to make the devices compatible to be controlled by Tiva C which works at 3.3V

Choice of ICs

- The ADC was chosen from the ADCs TI had to offer based on the number of bits and the error rate which should at least a precise 10-bit data for good accuracy.
- The choice of DACs and Digital Potentiometer was made based on their easy availability in WEL and their good enough step size and accuracy. A 12-Bit SPI controllable DAC was the ideal required component.

Following is the image of the printed board (some of the ICs have been removed at the time of clicking the image)



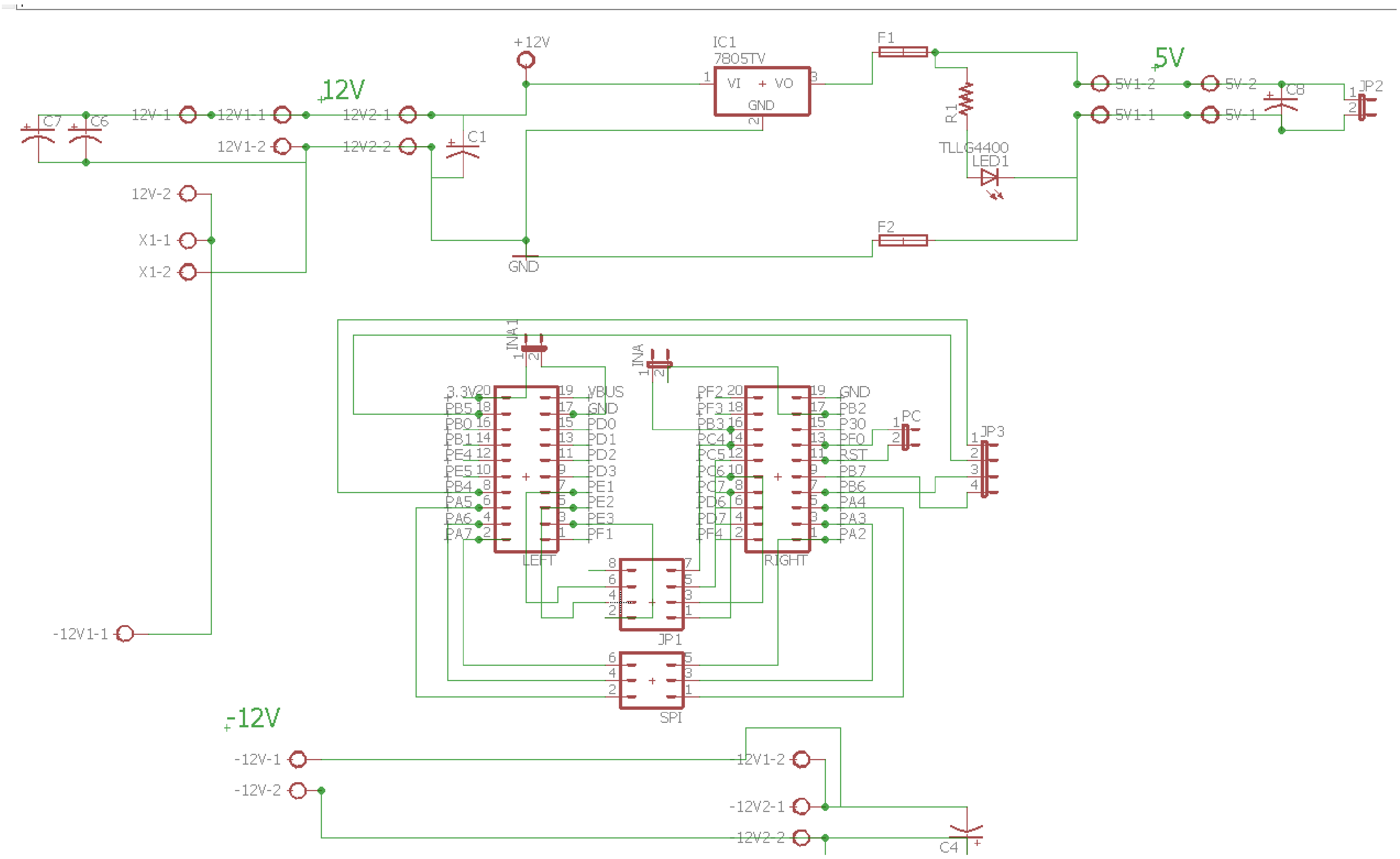
OVERALL POWER SUPPLY

The following are the devices and components used in its implementation. These are pertaining to the schematic of the implementation circuit as shown on the schematic figure.

Bill of Materials		
Component	Value (as Applicable)	Utility
-12V/-12V2	Screw Connector	-12V output
-12V1	Screw Connector	-12V Input
5V/5V1	Screw Connector	5V Output
12V	Screw Connector	+12V input
12V1/12V2	Screw Connector	+12V output
C1/C4/C6/C7/C8	100uF	Decoupling Capacitors
F1/F2	Fuse rated 5A	Fuse holder
IC1	7805	+5V voltage regulator
INA	Jumper Connector	SDA/SCA for Tiva to INA
INA1	Jumper Connector	+3.3 and GND output
JP1/JP3	Jumper Connector	Tiva C GPIO pin map
JP2	Jumper Connector	5V low power output
LED1	LED	Signal the fuse is still intact
LEFT/ RIGHT	Jumper Connector	Header for connecting Tiva C to
PC	Jumper Connector	+12V dc power supply
R1	1K	Biasing resistor for LED
SPI	Jumper Connector	Connector for SPI connections to & from Tiva
X1	Screw Connector	For main switch

Details of the implementation are as:

- The track size on the PCB is taken to be a minimum of 80mils. This is done in order to ensure that the amount of current that flows through the tracks is well supported.
- Decoupling capacitors are used for stabilizing the output voltages.
- SMPS are used for supplying +12V and -12V from two different SMPS. As both required +12V and -12V have different specification requirement, so does the rating of the SMPS
- This board also acts as a shield for the Tiva C.
- The main switch is connected to the ground. This means that if the switch is disconnected, there is no connection between the circuit and the ground.
- LED is used to indicate whether the fuse is intact or not. Glowing LED implies that the fuse is intact and vice versa.
- Ground PAD is kept large on the board in order to manage the high current generated heat efficiently.


12V
+12V
IC1
7805TV
VI
VO
GND
F1
F2
5V
C7
C6
12V-1
12V1-1
12V2-1
12V1-2
12V2-2
C1
12V-2
X1-1
X1-2
GND
R1
TLLG4400
LED1
5V1-2
5V-2
5V1-1
5V-1
C8
JP2
INA1
INA
3.3V
VBUS
PB5
PB0
PB1
PE4
PE5
PB4
PA5
PA6
PA7
GND
PD0
PD1
PD2
PD3
PE1
PE2
PE3
PF1
LEFT
PF2
PF3
PB3
PC4
PC5
PC6
PC7
PD6
PD7
PF4
GND
PB2
P30
PF0
RST
PB7
PB6
PA4
PA3
PA2
RIGHT
PC
JP3
JP1
SPI
-12V1-1
-12V
-12V-1
-12V-2
-12V1-2
-12V2-1
-12V2-2
C4

MICRONTROLLER AND USER INTERFACE

Choosing Microcontroller

Requirements

- Rich set of peripherals such as I2C, SPI, UART, USB, etc. to be able to use various components without compromise.
- Sufficient number of GPIO pins
- Easy to extend functionality of project
- Easy debugging
- Enough clock speed to support small signal analysis (Future extension of the project)
- Easily available (in WEL off course)

TM4C123GH6PM is a high performance 32-bit ARM Cortex-M microcontroller. It provides support for rich set of peripherals along with higher clock rate. That's why it is suitable for the easy customizable and extendable project like this. Its 32-Bit architecture provides extensive computation power which may be useful in small signal analysis.

Peripherals

Sr. No	Peripheral	Use
1	USB	To achieve high speed data and instruction transfer between microcontroller and application running on computer.
2	SPI	Current version of project uses SPI to control ADC, DAC and Digital Potentiometer
3	I2C	Current and Voltage across Peltier device are received through I2C interface
4	PWM Block	Signalling LED and voltage across Peltier are controlled using Pulse Width Modulation
5	GPIO Pins	Number of GPIO pins are used for controlling on board devices such as relays and ADC, ADC, etc.

Serial Peripheral Interface

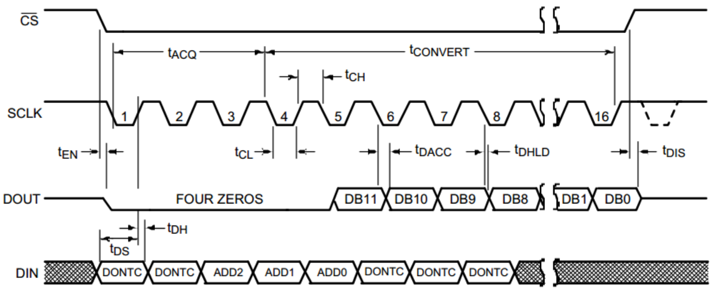
The SSI0 module on microcontroller is used as SPI to control ADC, DAC and Digital Potentiometer. These devices use 16-Bit packet format and support SPI clock speed above 1 Meg/Sec so, the configuration of SSI0 module is as follows:

- Protocol: SPI Mode 0 with MSB first
- SPI Clock Rate: 1 Meg/Sec
- Mode: Master
- Packer Width: 16-Bits

SSI0 module controls multiple devices connected in parallel pattern. So, chip selection is supported through multiple GPIO pins.

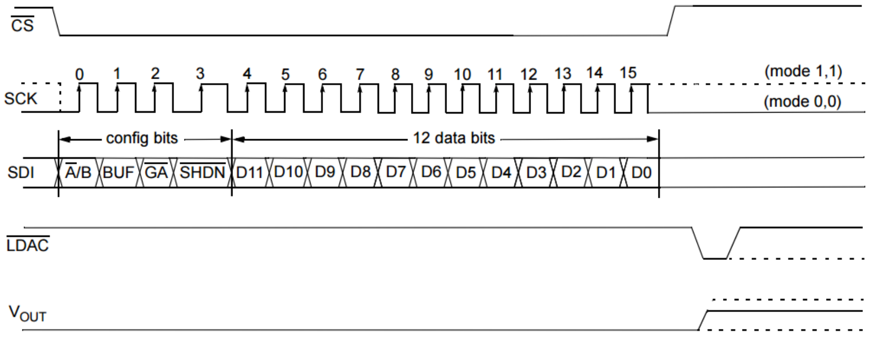
Devices:

1. ADC: ADC sends quantize value of selected signal in 16 bits out of which first 4 bits are zero. Microcontroller must send channel to be selected for next conversion while receiving current value. For detailed information please refer data sheet of the ADC. GPIO pin PA3 is used as chip select pin for the ADC.



ADC128S052 Serial Timing Diagram

2. DAC: DAC (MCP 4921) is controlled using 16-Bit message out of which first four bits are control bits which are followed by 12-Bit wide voltage value. GPIO pin PA6 is used as chip select pin for the DAC



I2C Interface

Current and Voltage across Peltier device are measured using INA 260 which implements I2C interface in I2C standard speed mode (100 Kb/s). It sends measured values in the form of 16-bit signed integer. Microcontroller is configured to use I2C0 module in standard mode to implement I2C connection with INA 260 which has the bus address of 0x44 and sets it in averaging mode to get more reliable measurements. For more information please refer the data sheet of the same.

USB Interface

Microcontroller uses USB0 module to exchange data and information with the controlling computer. It uses USB 2.0 interface in bulk transfer mode to exchange packages.

Configuration of the USB device:

Sr.No	Parameter	Value
1	Input FIFO depth	256 Bytes (in RAM)
2	Output FIFO depth	256 bytes (in RAM)
3	Device Vender ID	0x1CBE
4	Product String	Peltier and CHIL
5	Serial Number String	12345678
6	Data Interface String	Bulk Data Interface

PWM Block

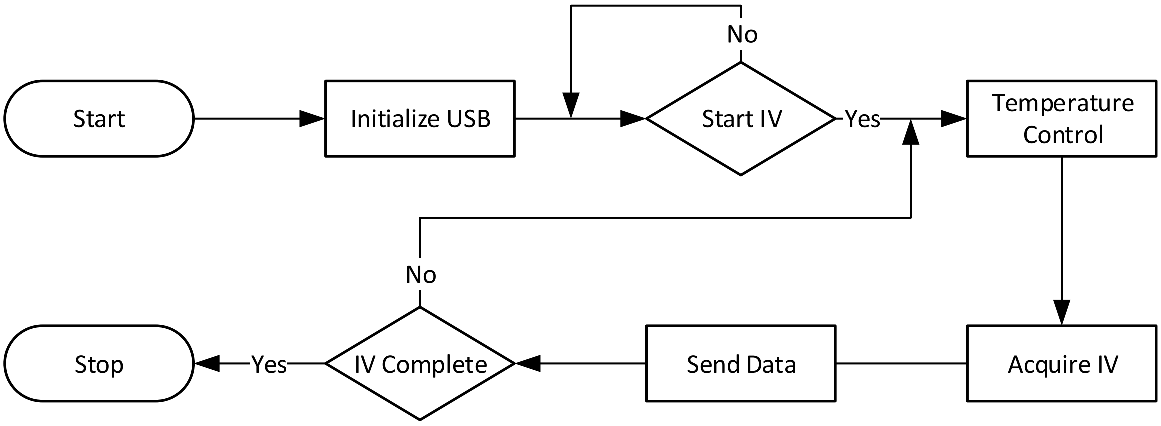
PWM block on the controller controls the onboard RGB LEDs and voltage across peltier device by varying duty cycle of PWM signal. PWM1 module is configured with clock rate of 50 MHz and pre-set value of 4000 which results into PWM frequency of 12.5 KHz.

Pin Mapping

Some of the pins are kept reserved for future extension of the project

Sr. No.	Device	Peripheral	Pins		Configurations
			Function	Name	
1	SPI	SSI0	CLK	PA2	SSI0 as SPI Master in mode0. Bit rate = 8MHz, Data width = 16bits *(read two channels one by one To get differential voltage)* All CS pins are initially pulled up
			MISO	PA4	
			MOSI	PA5	
			CS(ADC)	PA3	
			CS(DAC)	PA6	
			CS(DigiPot)	PA7	
			CS(Reserved)	PB[4-7]	
2	INA	I2C0	SCL	PB2	I2C0 as I2C Master in fast mode (1 MHz) Data width = 16 bits
			SDA	PB3	
3	PWM	M1_PWM5	Red LED	PF1	PWM frequency 50 MHz Max duty width 1000
		M1_PWM7	Green LED	PF3	
		M1_PWM6	Blue LED	PF2	
		M1_PWM4	Peltier Cooler	PF0	
4	USB	USB0	VBUS	PB1	USB 2.0
			ID	PB0	
			D+	PD5	
			D-	PD4	
5	GPIO		IV Relay	PC4	Default Low
			Peltier Relay	PC5	Default Low
			Reserve	PC[6,7]	
			Reserve	PE[1-5]	

Control Flow



Overall Picture of the Controller

The microcontroller is guided by the application running on the PC using various instructions. Application sends messages corresponding to each operation and waits until controller finishes the given task. At boot microcontroller starts USB peripheral only waits for application to send future command.

Some important aspects of the controller are as follows:

- Microcontroller initializes only USB peripheral after booting. Application later sends it command to initialize other peripherals. This helps to avoid unnecessary operations until user has not commanded to start.
- Temperature controlling function and IV acquiring function are called frequently after specific time using a timer interrupt. Current application sets timer value to get an interrupt after every 10ms. Interrupt serves as temperature controlling function for once and IV acquiring function for next three times.
- IV acquisition function acquires data only when temperature is stable which is to be specified by the user.
- Microcontroller acquires IV data for specified voltage and current interval for given temperature and sends completion signal to application and waits until called to do so again.
- Microcontroller maintains the temperature even after completion of the IV Characteristics

Code Organization

Sr.No	File Name	Description
1	Peltier_and_CHIL.c	Main function and instruction decoding functions
2	startup_ccs.c	Map of all the interrupts to corresponding handlers
3	temp_loop.h	Temperature Controlling Functions
4	temp_loop.c	
5	IC.h	IV Acquisition function
6	IV.c	
7	TivaC_USB.h	USB interrupt handler and data transfer functions
8	TivaC_USB.c	
9	usb_bulk_struct.h	Defines USB structures such as FIFOs, Device String, etc.
10	usb_bulk_struct.c	
11	basic_includes.h	All libraries and global variables

Message Packets [Future Work]

Message packets are divided into two groups, command and data. All the data packets start with byte `0x22` while command packets start with `0x21`. The second last byte of all the packets is `0x00` and last byte indicates the size of the packet. This two bytes can help to ensure coherency of the packet.

Data Packets

IV Data									
Index	0	1	2	3	4	5	6	7	8
Value	0x22	0x53	Sign	Device Voltage		Device Current		0x00	0x09

Temp Loop Data										
Index	0	1	2	3	4	5	6	7	8	9
Value	0x22	0x59	Temperature		Peltier Current		PWM Duty		0x00	0x0A

Command Packets

Start IV						
Index	0	1	2	3	4	5
Value	0x21	0x81	Temperature		0x00	0x06

Start Devices				
Index	0	1	2	3
Value	0x21	0x99	0x00	0x04

Set Device Parameters													
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Value	0x21	0x66	Sign	Voltage Max		Voltage Min		Current Max		Current Min		0x00	0x0D

Stop Devices				
Index	0	1	2	3
Value	0x21	0x55	0x00	0x04

Command Acknowledgment				
Index	0	1	2	3
Value	0x21	0xFF	0x00	0x04

Installing Softwares for TM4C Launchpad Board

Code Composer Studio 7

Code Composer Studio (CCStudio or CCS) is an integrated development environment (IDE) to develop applications for Texas Instruments (TI) embedded processors.

To install CCS run the downloaded executable and follow the instructions.

Download Link: http://processors.wiki.ti.com/index.php/Download_CCS

TivaWare for C Series

Includes royalty-free libraries (Peripheral, USB, Graphics, Sensor) and kit-/peripheral-specific code examples for all TM4C devices.

Download Link: http://www.ti.com/tool/sw-tm4c

Note: Download complete package

StellarisWare embedded USB drivers

The Stellaris USB library provides a Windows™-based INF for supported USB classes in a precompiled DLL that saves development time. This library is helpful for debugging USB peripheral with Windows PC. However, this library is not necessary for this project.

Download Link: http://www.ti.com/tool/SW-USB-WINDRIVERS

Note: This is not in circuit debugger driver which helps to program and debug device from debug unit

Libusb-win32

libusb is a C library that gives applications easy access to USB devices on many different operating systems. This library is useful for interfacing Tiva-C Launchpad with GUI using USB 2 interface.

Installing Lib-USB for specific device on Windows PC

1. Connect the USB device (Device USB not Programmer) and open Zadig.

2. Go to options and select device vendor ID (0x1CBE in this case) and install libUSB-win32 latest version
3. Verify installation using test program from Libusb filter package.

Download Links:

Zadig: http://zadig.akeo.ie/downloads/zadig_2.2.exe

Libusb filter package: https://excellmedia.dl.sourceforge.net/project/libusb-win32/libusb-win32-releases/1.2.6.0/libusb-win32-devel-filter-1.2.6.0.exe

USER INTERFACE APPLICATION

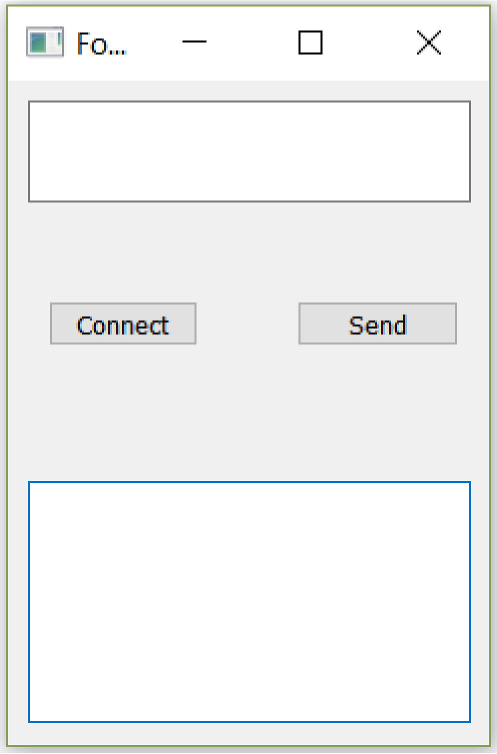
The GUI application is developed using Python 3.5 and PyQt5 with other supporting packages such as Matplotlib. Important operations of the application are:

- Interact with the user to get device parameters
- Control tasks performed by microcontroller
- Display output to the user

Some important aspects of the application

- Separate thread for receiving data from USB. This allows unblocked interaction with the user and USB device simultaneously
- Live display of device parameters
- Platform independent application (Compatible with Linux, Windows, Mac)

Note: The GUI wasn't completed as it was planned. Instead a testing version of the GUI was used and integrated with graph making ability to plot the temperature and IV Characteristics Plotter. The following image does not show the graph window. It can be seen in the next Chapter.



The instructions to use the application is as follows:

- Start the TivaC Application in the Debug Mode (the code has been completely tested only in the debug mode) and be quick to start the application.
- As soon as the application begins, click the connect button to connect it to Tiva C before its USB buffer gets filled (if it happens then Tiva enters a stalled state and must be restarted).
- The application then beings to receive and plot the temperature data from Tiva. The default set point is set to be 40 degrees
- The required set point is to be entered with a positive offset of 10 degrees and the terminal cannot accept negative values and hence the minimum possible set point is -10^{0}C. The maximum possible set point is 100^{0}C. Eg. To get the set point to 60^{0}C, you must enter a value of 70 into the UI and click the Send button.
- Once the desired steady state is reached based on the graph visible to the user, the user must then input a number greater than 110 to start the IV Plotter. Then TivaC begins to get the IV Data and the IV graph is plotted as soon as all the data is received from the device.
- Once this is done the UI resumed into the temperature mode and the temperature plot continues.

The GUI and the code running on the TivaC board had to be toned down significantly to avoid last minute debugging challenges for the code apart from the hardware. A minimal GUI instead as mentioned above and a corresponding simple code was used for the TivaC to make the project functional as a total system.

Installing Required Python-3 Packages for GUI

We have used Python 3.5 for this project so, all the libraries are for Python 3.

PyQt5

PyQt brings together the Qt C++ cross-platform application framework and the cross-platform interpreted language Python. Qt is more than a GUI toolkit. It includes abstractions of network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widgets. GUI of this project uses PyQt extensively for threading and user interface. This project is developed using PyQt5.

Installing PyQt5 using PIP3: `pip3 install pyqt5`

Installing PyQt5 designer using PIP3: `pip3 install pyqt5-dev-tools`

Note: You need root access for installing these packages. On Windows PC, you can use `Command Prompt(Admin)`

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. This project uses Matplotlib for plotting graphs.

Installing Matplotlib: `pip install matplotlib`

PyUSB

PyUSB aims to be an easy to use Python module to access USB devices. PyUSB relies on a native system library for USB access.

Installing PyUSB: `pip install pyusb`

Note for Windows user: You must have libusb-win32 installed for using PyUSB

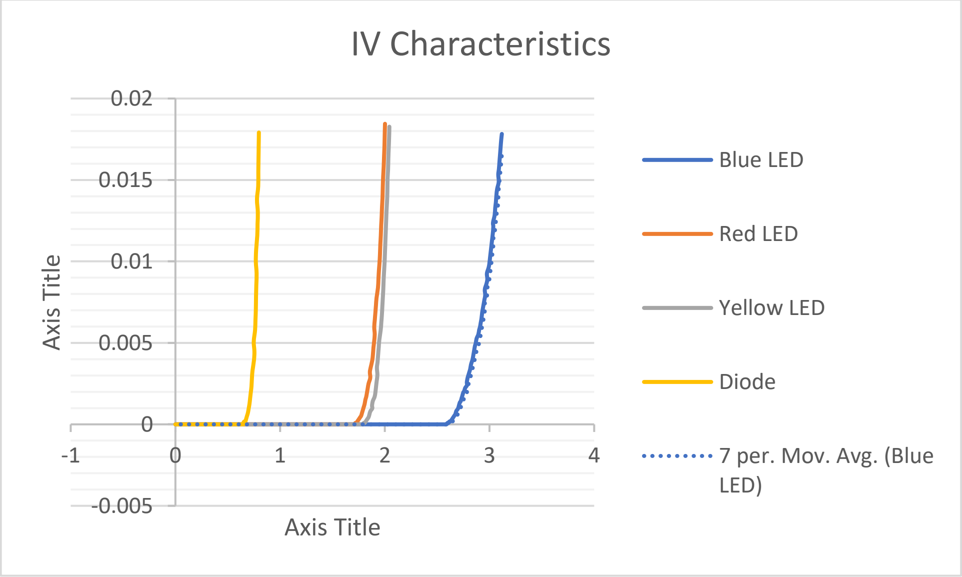
Chapter 4: Performance Evaluation

IV CHARACTERISER

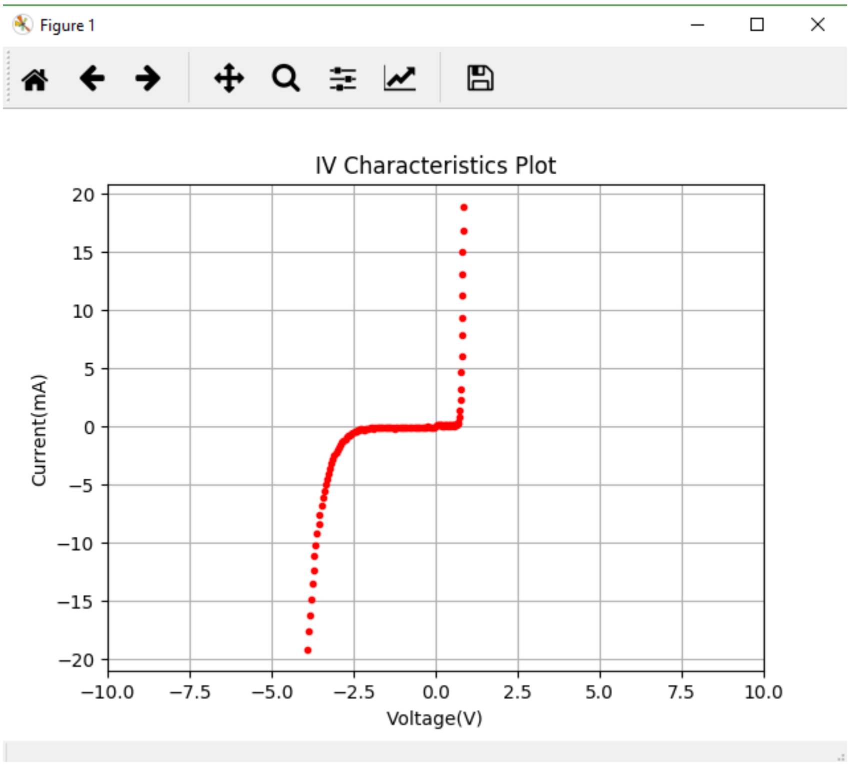
The design was first implemented on a bread board to test the design and then moved to the bread board. Some of the issues faced with its implementation on the PCB were

- Printing inaccuracies led to improper connection of certain tracks which had to manually connected using an air wire
- Noise in the power supply caused the ADC to produce erroneous values when tested with the entire system though it worked well in a standalone fashion
- Using the SPI from TivaC was also a tricky affair. It required a good understanding of the process Tiva goes through in order to get the
- Improper handling of the power supply to the ADC led to inaccurate and fluctuating readings. Patch up was made because of absence of the digital potentiometer to provide a reference supply to the ADC as it required
- The ADC used was not so stable and we had to replace 2 of them because they were burnt out. Hence it is not recommended to use the ADC that has been used in this project
- The level shifter used in this board and the peltier control board are very low power shifters and hence care must be taken while designing the PCB to ensure that the tracks are made as short as possible and the devices it controls do not require must sink current. Also because of this low power operation, it is not possible to measure the signals of the level shifter using a DSO and measurement can mess with the rest of the circuit and give erroneous results. The only way this device can be tested is by checking whether the connect component works when connected using it.

Following are the results of the standalone tests performed on the IV characteristics board



The following is the plot, from the GUI Application, of a Zener diode



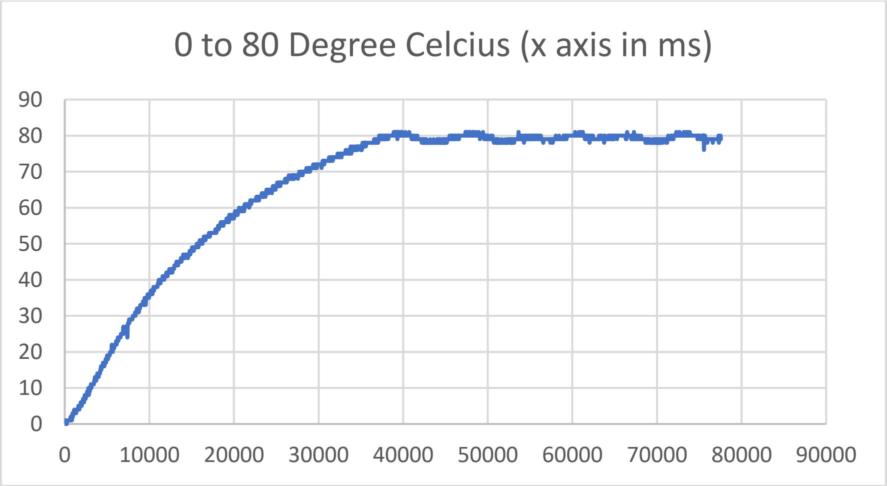
PELTIER CONTROL BOARD

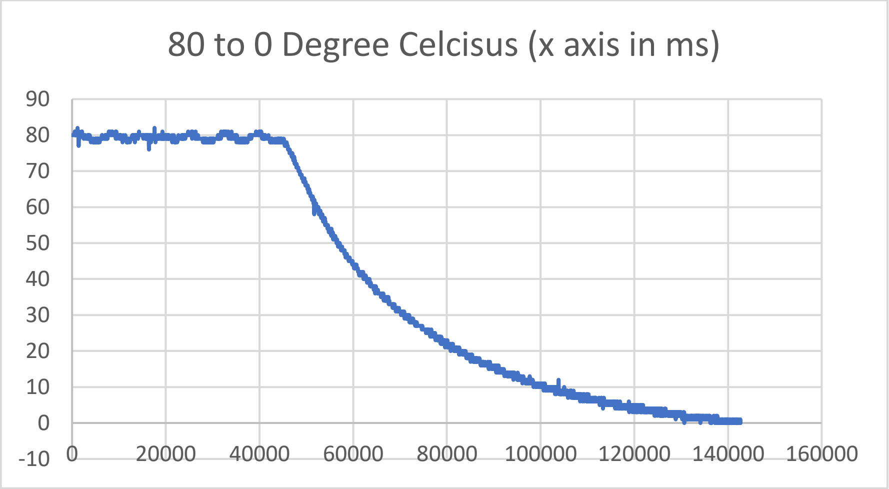
The design was tested first on a breadboard without the presence of a relay and the PCB was directly made with the Relay and the INA current shunt monitor.

Some errors were made in the making of the PCB and in the design as follows

- Relays used in the PCB, worked on a 5V control signal and had to convey a PWM current with a 12V supply, and this caused immense disturbance in the 5V supply which created a mess in the measurement of the temperature in the steady state because the LM35 supply was coupled with the Relay Supply. Possible solutions are using isolation between the 5V used for the relay and the other low power circuits. But the better option would be to implement a complete H Bridge which would not require the involvement of the 5V supply.
- Secondly the power supply to the current shunt monitor was not connected in the PCB that was printed and hence it was removed from the PCB. But a test board can instead be connected and used to measure the current flow and provision has been made in the board for that purpose.

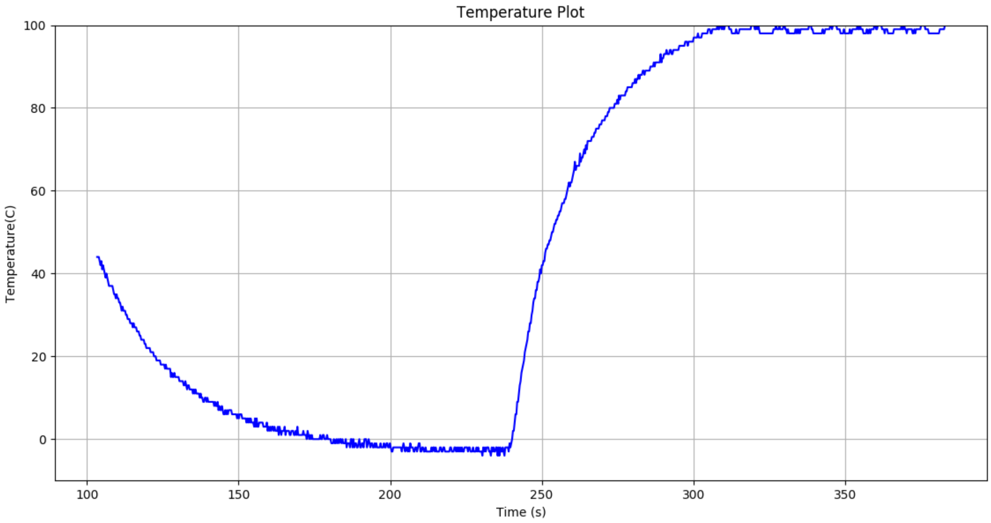
Here are some of the temperature plots based on several provided set-points. The settling time from 0^0 to 80^0 was found to be approximately 40 seconds and around 100 seconds for the reverse direction as can be seen in the graphs. The exact times vary with the room temperature, isolation provided to the Coolers and also the direction of temperature change as is intuitive.



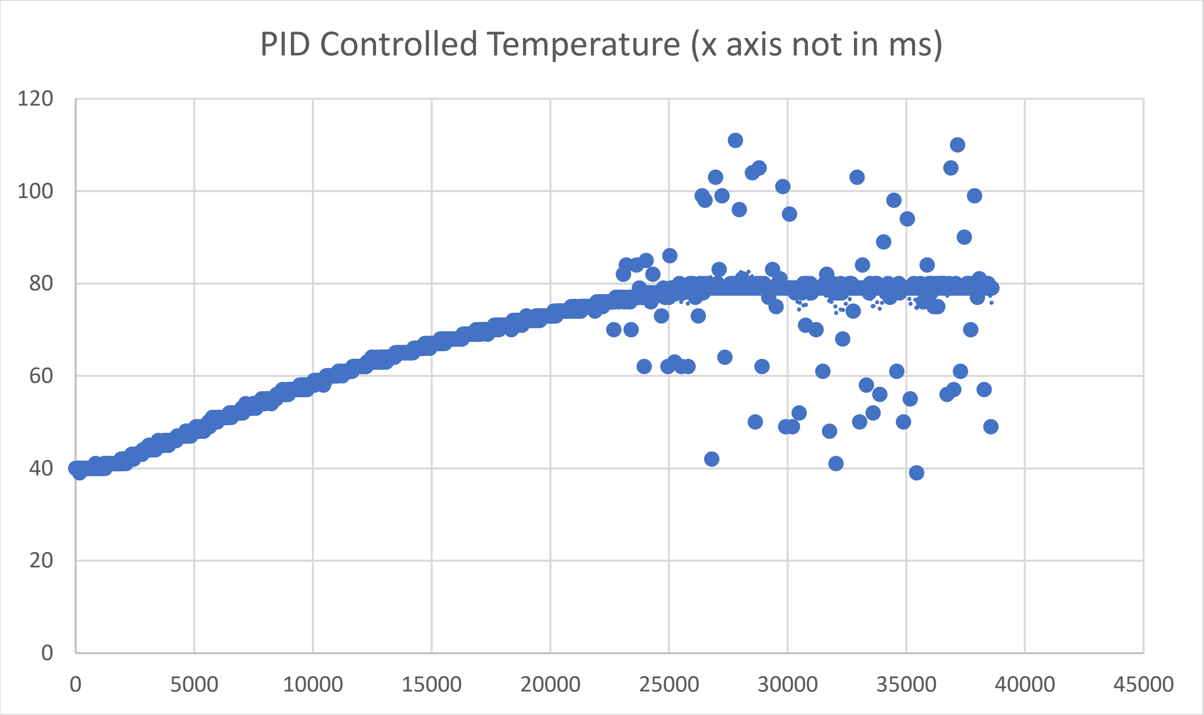


To correct for the issue with the Relay and the PWM noise so that we do not compromise on the temperature measurement we shifted to a Bang-Bang Controller which basically performs an On-Off control around the set point. We have used a 1^0C error band to turn the Peltier Cooler on and off. The results are satisfactory as very accurate control is unnecessary for our application.

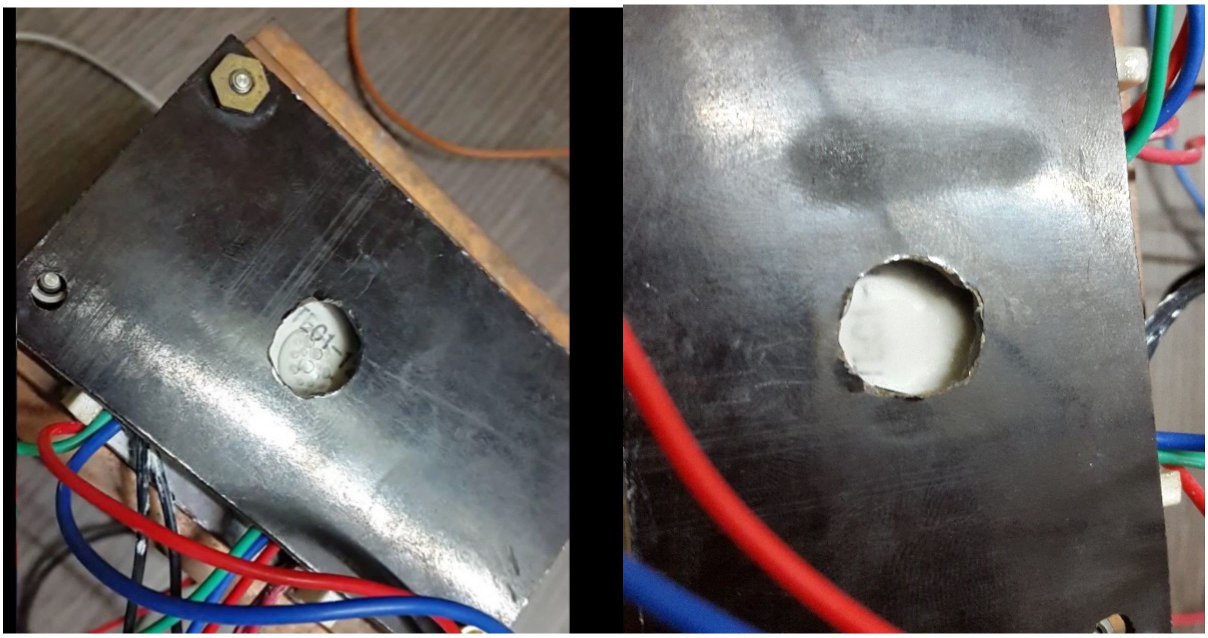
The following is an image of the GUI plotting the temperature.



The following image clearly illustrates the issue we faced with using a PWM based controller in our final circuit. As we can see though the stability achieved is indeed way better than in the case of a Bang-Bang controller, the measurement noise as soon as the PWM duty cycle beings to vary around the set-point begins to bury the actual temperature of the Peltier Surface.



We could obtain sub-zero temperatures and boiling point temperatures using our final setup, and this can be validated by looking at the following picture which show formation of ice and boiling of water on the Peltier Cooler surface. As the Peltier Cooler surface is insulated from the rest of the circuitry we were able to test the temperatures using water, and looking at the graph during that period we could observe that the additional loading applied did not affect the steady state and hence achieving good control also supported by the thermal inertia.



POWER SUPPLY

- The major issue with this board was that the pins for the TivaC were reversed and hence it was not used as a breakout board and all the tracks had to be cut out to get the TivaC board placed
- Rest of the voltages were effectively produced by the board

USING DRV595 – A MOTOR DRIVER IC (H BRIDGE)

DRV595 is an H-bridge IC. The speciality of this IC is that it can handle high power and provides internal switching for changing the directionality of the current. In principle the IC takes in analog differential input biased at 3.3V and amplifies it into corresponding PWM voltage output. This PWM output is then passed through a low pass filter to generate a final no ripple output.

The test board for testing this IC was designed. We faced many problems during its designing. This includes scaling down of the test board due to software differences, unavailability of proper inductors and inversion of the printed board. This caused many delays in conducting the testing. During the testing, we were not able to drive the IC out of the saturation region. Therefore, either we were getting zero output (at corresponding zero differential input) or we were getting the maximum output.

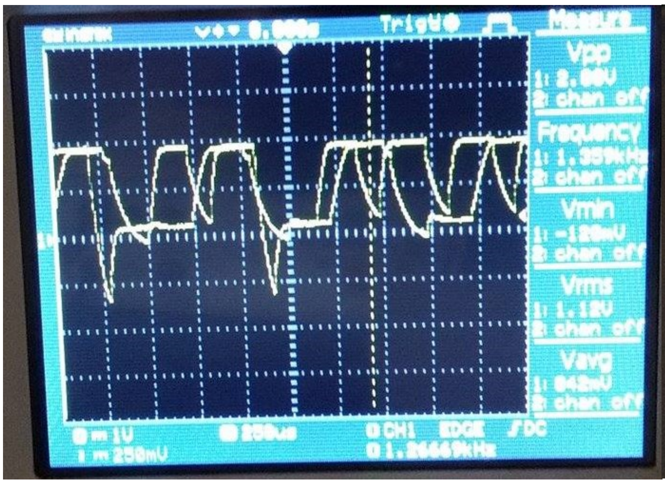
The testing conducted on DRV595 is further elaborated as follows:

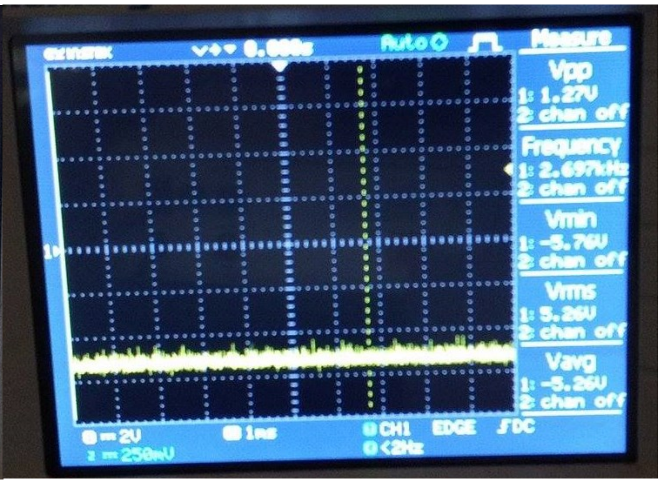
1. The test board for DRV595 was made in reference to the Evaluation board developed by TI. The corner point of the design was the availability of power inductors for smoothening the output current.
2. Due to unavailability of inductors we connected 10 smaller value inductors in parallel to each other to meet the required specifications. This worked out to be just fine.
3. The exact procedure followed and the results obtained are as follows:
    a. Apply the following voltages:

$$IN+ = IN- = \ 5V$$
$$AVcc = 5V$$
$$PVcc = 0V$$

    The output obtained was around $300\ mV$.
    This is due to the fact that the IC was not turned on as PVcc < 5.6V (turn on voltage)

    b. Now the circuit was powered up by applying $PVcc = 8V$.
    The input applied remained the same, i.e., $IN+ = IN- = 5V$
    The average output was still around 0V.
    This is due to the input differential voltage was still zero.





c. Note: The IC is biased to provide a gain of 1dB/V.
Now the differential voltage was given to be 0.1V.
Output in this case was saturated to 5.2V in spite of the fact that we connected a load at the output.

d. We tested multiple times and tried to drive the IC out of saturation, but were unsuccessful. Therefore, after spending considerable amount of time on the single IC we decided to shift onto the other best methods that could be used to achieve the goal.

Testing Notes:

(i) At the start-up of the IC, IN+ and IN- must be held at equal voltages and above 3.3V. This must be ensured for at least a second for ensuring the correct working of the IC.

(ii) Also keep an eye at the fault pin and the shutdown pin for checking the current mode of operation of the IC.

After the unsuccessful attempt at testing DRV595, we shifted our focus on using power MOSFETs for making an (indirect) current source. The idea of the MOSFET came from our mentor and other teams who have successfully implemented the same. There was a little shortcoming of getting ripple input to the Peltier, but as the device worked, we continued with the implementation.

Chapter 5: Conclusion and Future Work

The project details, goals, implementation considerations, results and problems faced during the prototyping procedure have been documented in this report. The product aims at automating the IV characteristics of a two-terminal device at various temperatures. We able to reach a point where we could implement only the DC Characteristics of the DUT and the temperature control was only based on voltage.

Future work and improvements to the project could include

- Performing the small signal analysis of the two-terminal device at various biasing conditions
- Extend support to multi-terminal devices by applying bias to the other terminals and performing the IV Characteristics across the remaining terminals. This can be used to characterise devices such as MOSFETs or BJTs whose temperature characteristics are also important.
- Temperature Control of the Peltier Cooler can be made more precise by using the double control loop as suggested in the document. This can allow the temperature settable platform to be used for various other applications which may be very sensitive to maintenance of precise temperatures.
- More dynamic changes can be introduced in the IV Characteristics to make the control more robust, immune to errors and fine grained to get high standard results.