Milestone-3 report on

Non-contact Current Meter

MON – 14

Supervisors

Prof. Anil Kottantharayil

Prof. Prem C. Pandey

RA/TA

Raghuram

Sai Praveen

Submitted By

Avinash Chowdary Narne – 200070047

Gnanendar Reddy Palagiri – 200070053

Bhuvana Chandra Pullabhotla – 200070063

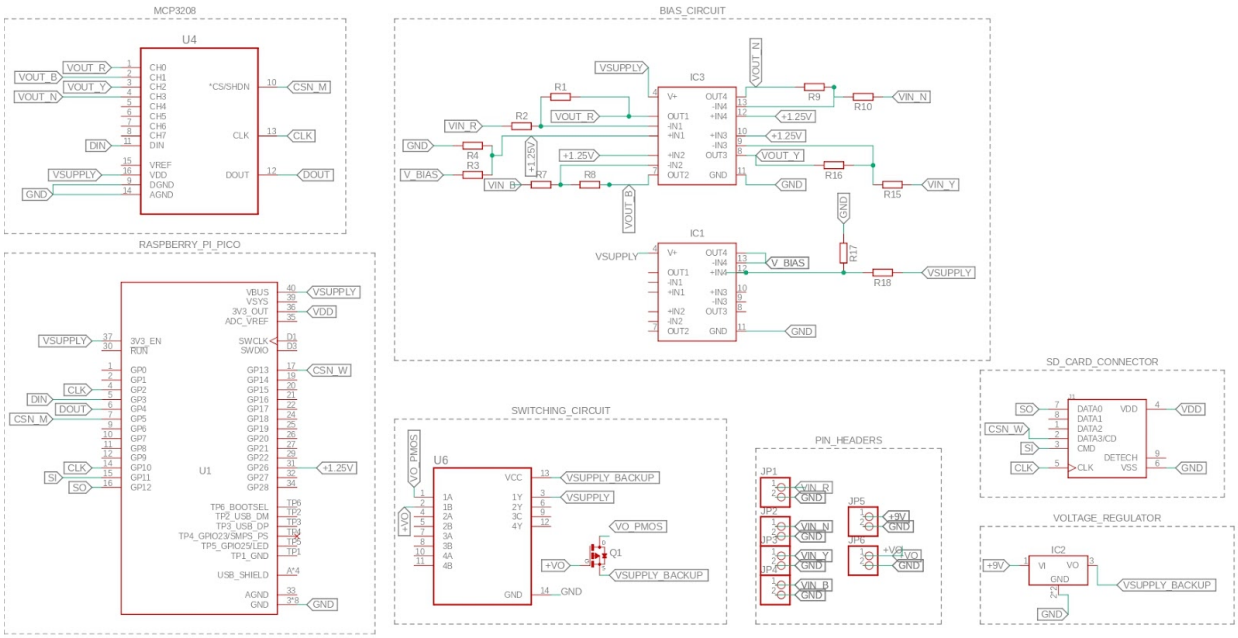
Electrical Engineering

Indian Institute of Technology Bombay

March 2023

Circuit schematic and layout

Schematic:

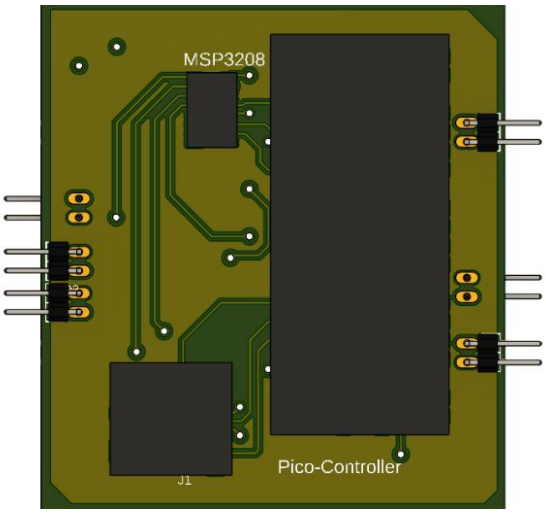


26-03-2023 22:21 f=0.92 C:\Users\bhuva\AppData\Local\Temp\Neutron\ElectronFileOutput\11568\sch-c8eaf751-0545-4a7d-86dc-edd16b8e0dc1\Schematic v39.sch (Sheet: 1/1)

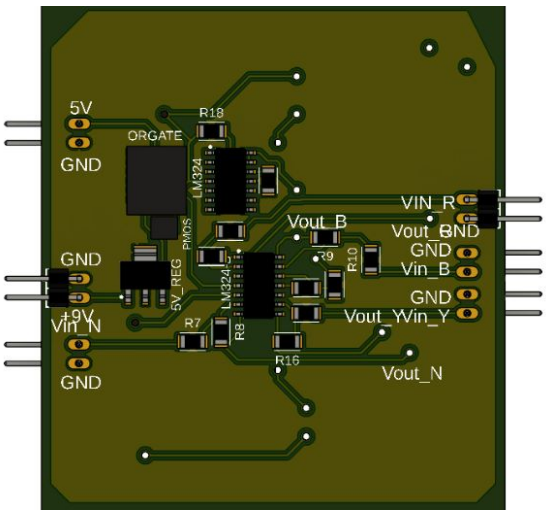
Layout:

3D:

Top layer

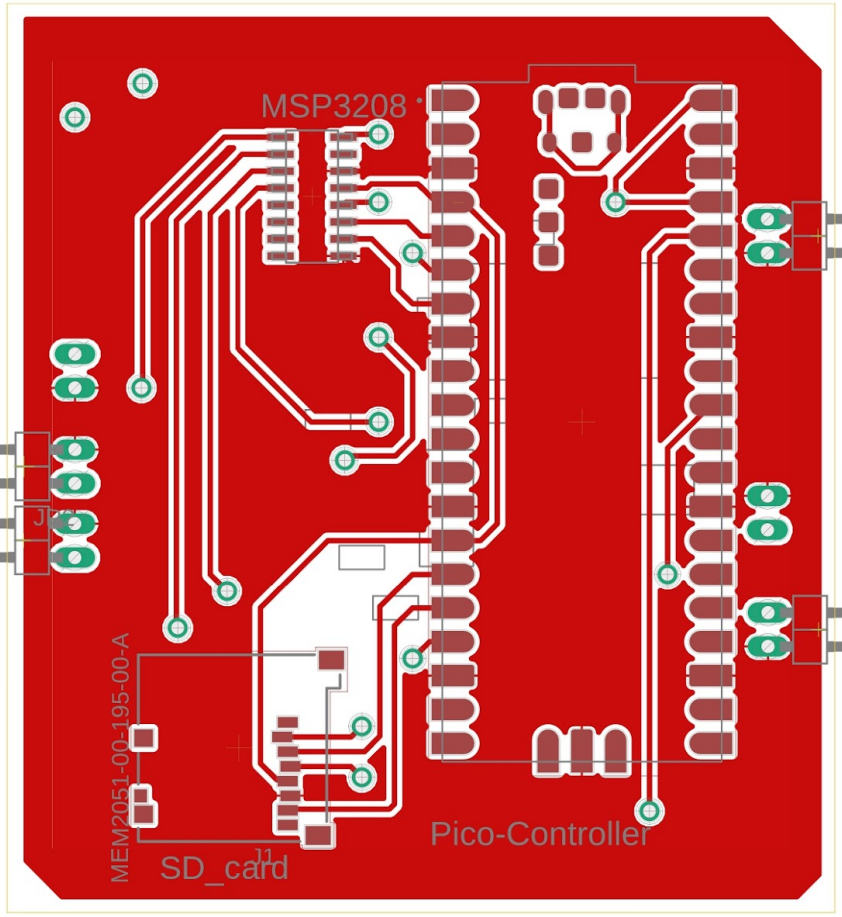


Bottom layer



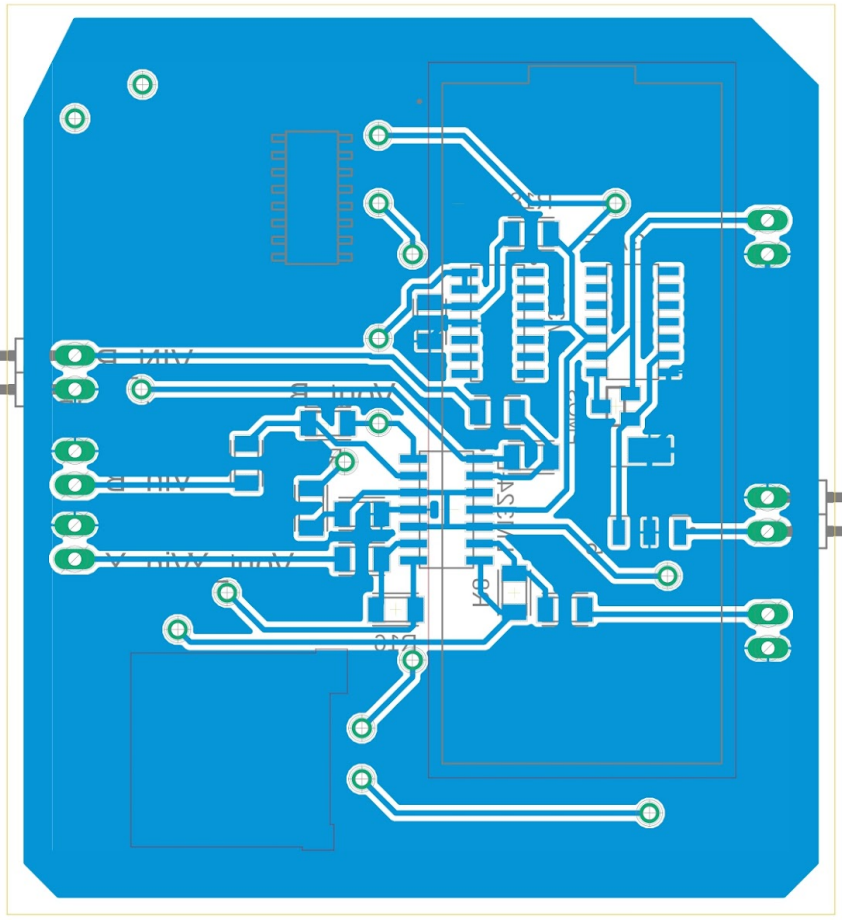
2D:

Top layer



26-03-2023 23:26 f=3.00 C:\Users\avi03\AppData\Local\Temp\Neutron\ElectronFileOutput\8288\brd-75be2e4c-8014-46da-b792-95d65e2f6eea\PCB v40.brd

Bottom layer

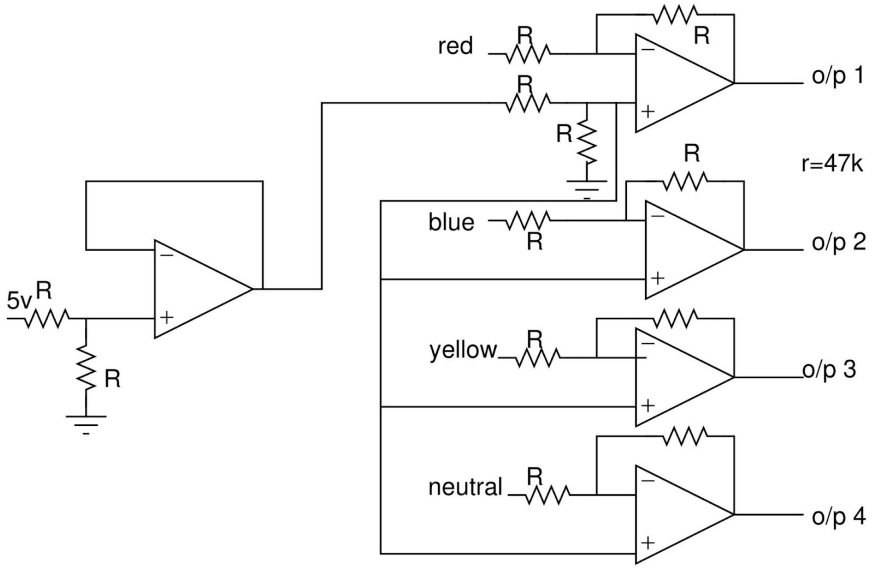


26-03-2023 23:28 f=3.00 C:\Users\avi03\AppData\Local\Temp\Neutron\ElectronFileOutput\8288\brd-75be2e4c-8014-46da-b792-95d65e2f6eea\PCB v40.brd

Description of test setup and test method

Test Setup:

Test-1:
As we are using biasing circuitry to to make the input incoming AC voltage stay above 0V,a we are checking the circuitry to see if we are getting the required voltage after biasing circuit as output
Biasing Circuit: here all resistor values are 47k and the inverting terminal input of each opamp is the clamp o/p from each wire(red,blue,yellow and neutral).Here all opamps are lm324 we chose them because they can operate with 5V supply which is convenient for us



Test-2:

We gave these four outputs we got above which are in the range 1-4V after biasing, to MCP3208 ADC which has 8 channel analog input.This MCP3208 can send data to the rpi pico w using spi interface.In this test we are sending one of the biased analog o/p's to to the ADC then to rpi pico and calculating its rms value(which is

the final output we need).

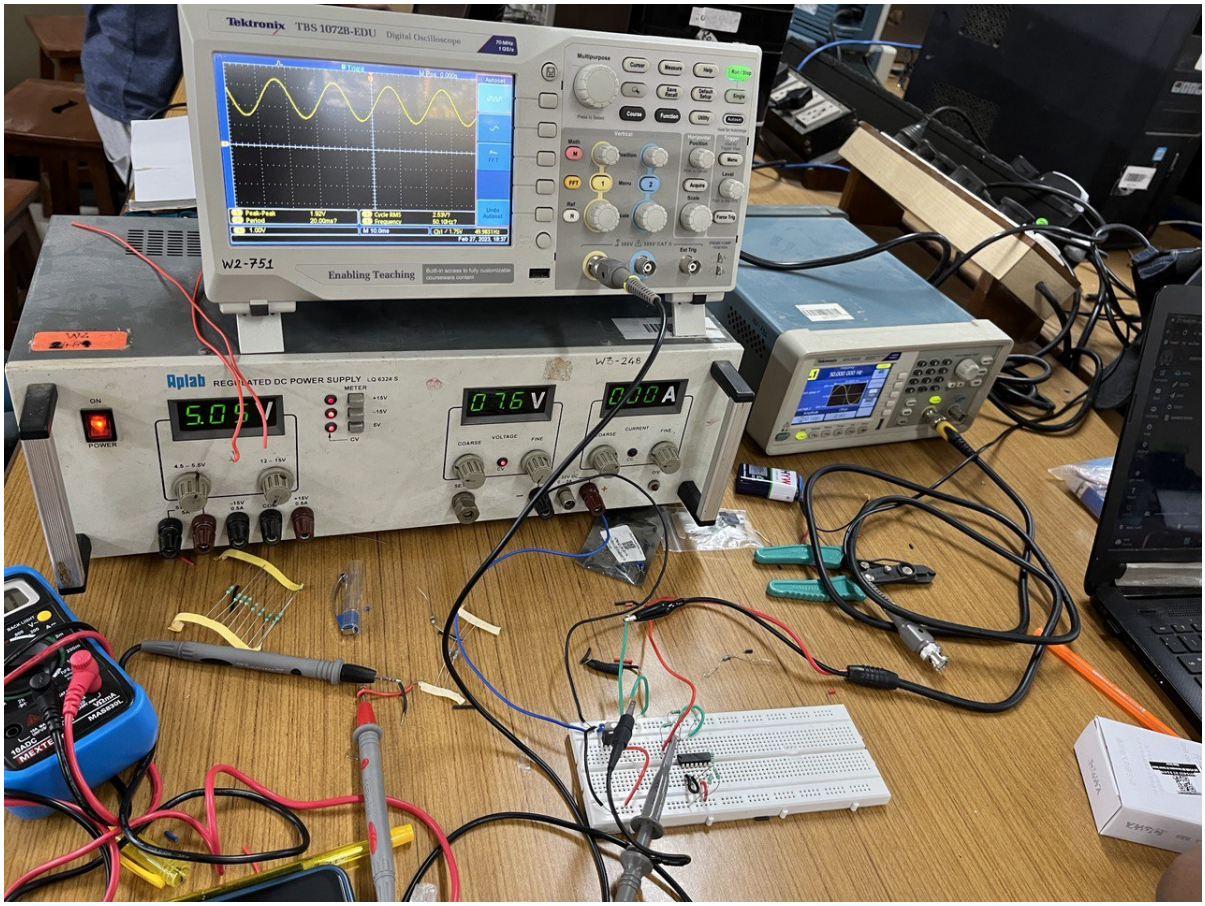
For this test the circuit is as given in schematic

Test Method:

Test-1:

We connected the biasing circuit as shown in the above figure and supplied the 5V

from the power source in the lab and then we supplied one of the inputs , say input 1 (red) from the function generator (a sine wave between 0-1.41V since the clam o/p for current in the range 0 to 30A is 1V rms.) and measured the o/p using oscilloscope.



Test-2:

Here in this test we used the similar biasing circuit as in test 1 but we now connected the circuit to the MCP3208 and raspberry pi pico w as given in schematic and tested whether it can store the correct values in the registers. The code used for testing this is given below:

```
/**
* Copyright (c) 2020 Raspberry Pi (Trading) Ltd.
*
* SPDX-License-Identifier: BSD-3-Clause
*/


#include <stdio.h>
#include <string.h>
#include "pico/stdlib.h"
#include "pico/binary_info.h"
#include "hardware/spi.h"
#include <math.h>
```

```
// Define SPI pins
#define SPI_PORT spi0    #selecting the 1st spi
#define SPI_SPEED 1000000  #configuring spi rate to sample rate of MCP3208
#define PIN_CS   15
#define PIN_voltage   26  #pin to take input  of lm324 + terminal to get
back current valuee



// Function to read analog value from MCP3208
void mcp3208_read(uint8_t channel, uint16_t *data) {
   uint8_t tx_buf[3], rx_buf[3];
   tx_buf[0] = 0b00000110 | ((channel & 0x07) >> 2);
   tx_buf[1] = ((channel & 0x03) << 6);
   tx_buf[2] = 0;
   spi_write_read_blocking(SPI_PORT, tx_buf, rx_buf, 3);
   *data = ((rx_buf[1] & 0x0F) << 8) | rx_buf[2];
}

int main()
{
  // Initialize stdio
  stdio_init_all();


  // Initialize SPI
  spi_init(SPI_PORT, 1000000);
  gpio_set_function(PIN_CS, GPIO_FUNC_SPI);
  gpio_set_dir(PIN_CS, GPIO_OUT);
  gpio_put(PIN_CS, 1);

  // Initialize variables for RMS calculation
```

```
int values[4] = {0};
float rms[4] = {0.0};
float sum[4] = {0.0};
int count = 0;
const int SAMPLE_RATE = 1000000; // Hz
uint16_t data[4];

while (true)
{
     gpio_put(PIN_CS, 0);
     mcp3208_read(0, &data[0]);
     mcp3208_read(1, &data[1]);
     mcp3208_read(2, &data[2]);
     mcp3208_read(3, &data[3]);
     gpio_put(PIN_CS, 1);


         sum[0] += pow(((float)data[0])-(PIN_voltage*2) / 4095.0, 2);
         sum[1] += pow(((float)data[1])-(PIN_voltage*2) / 4095.0, 2);
         sum[2] += pow(((float)data[2])-(PIN_voltage*2) / 4095.0, 2);
         sum[3] += pow(((float)data[3])-(PIN_voltage*2) / 4095.0, 2);

    count++;


    // Update RMS value for each channel at a fixed frequency
    if (count >= SAMPLE_RATE)
    {
        for (int i = 0; i < 4; i++)
        {
            rms[i] = sqrt(sum[i] / SAMPLE_RATE);
            sum[i] = 0.0;
            printf("Channel %d: RMS = %f\n", i, rms[i]);
        }
        count = 0;
    }


    // Delay for a short period of time to control sample rate
    sleep_ms(1000 / SAMPLE_RATE);
```

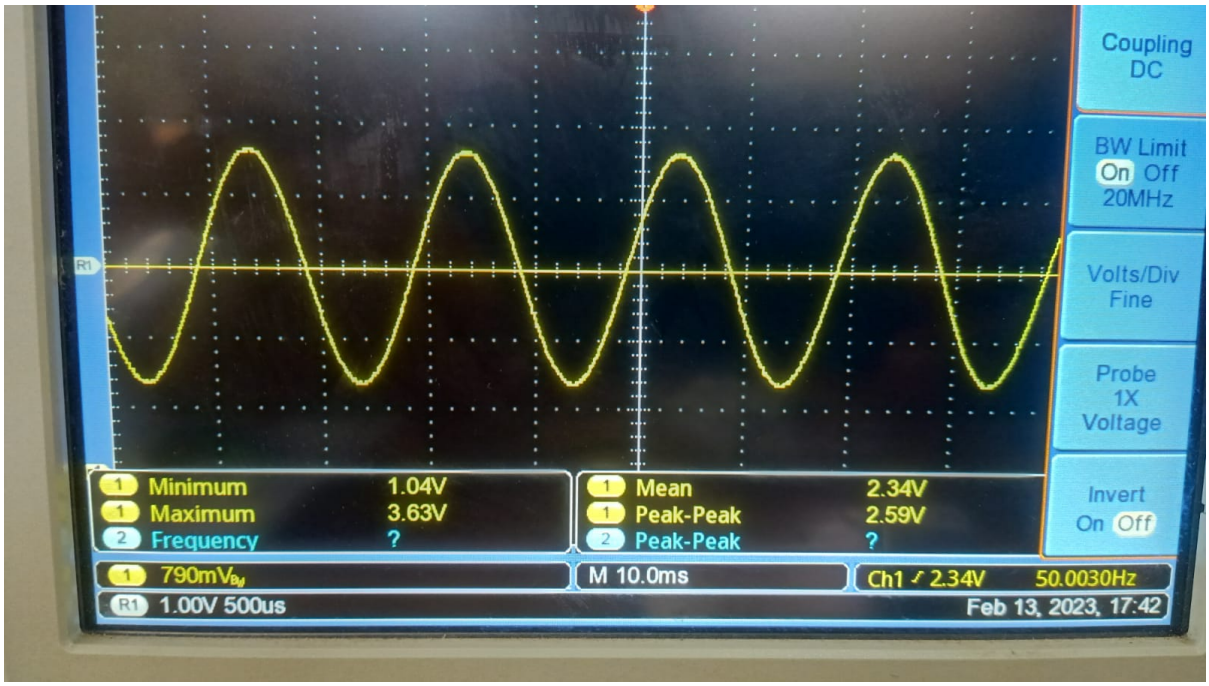
```
  }


  return 0;
}
```

We were successfully able to generate the dumpable file (u2f for r pi pic).

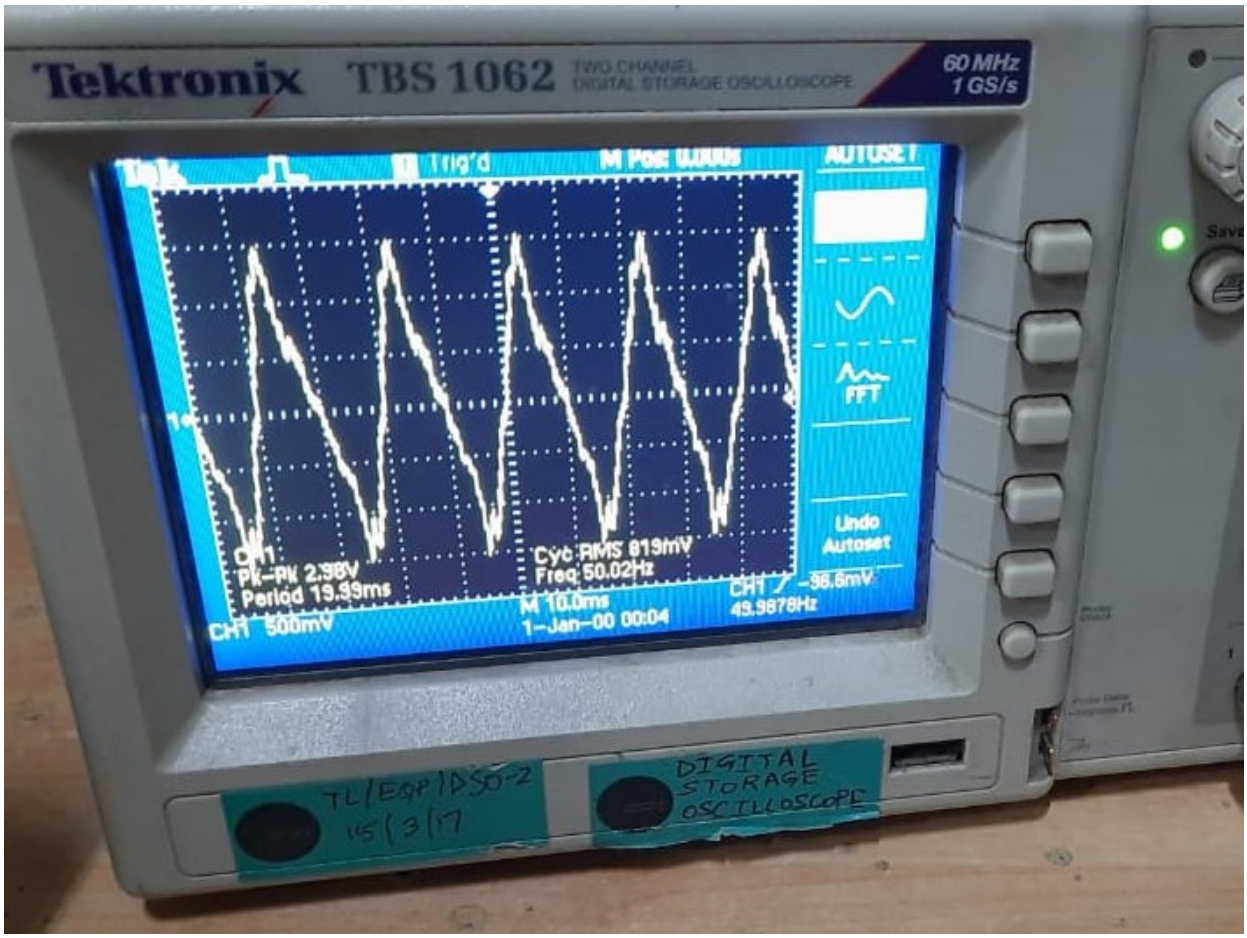
Test results

Test-1:

Here we have given input of voltage between 0-1.141V to the biasing circuit and the o/p is as expected is in the range of 1-4V

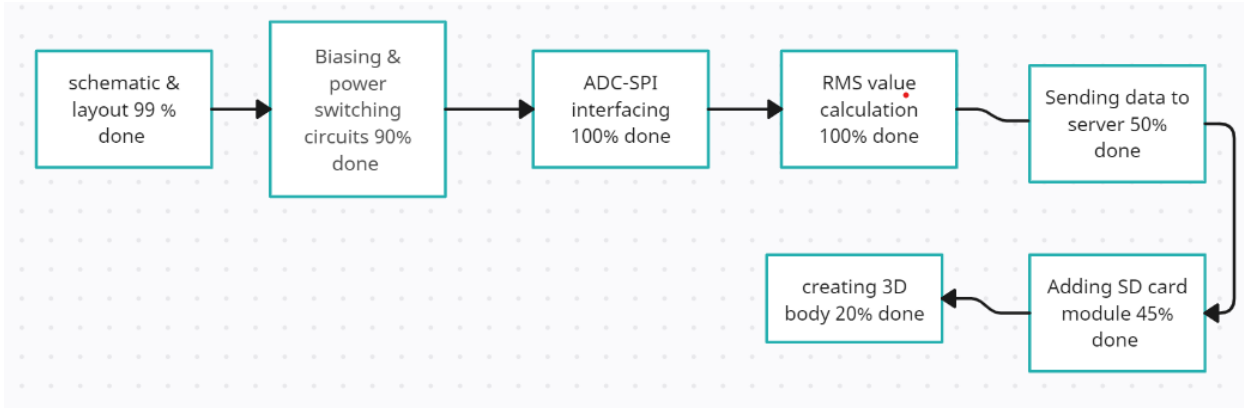


Test-2:
The entire working circuit till r pi pico gathering the data from MCP3208 to r pi pico



We were able to successfully sense the current value through clamps and bias it using the biasing circuit and transfer the analog data to MCP3208 and sample it digitally in raspberry pi pico and store those values in local variables (or registers) to transfer it to the server.

Block Diagram of testing percentages:

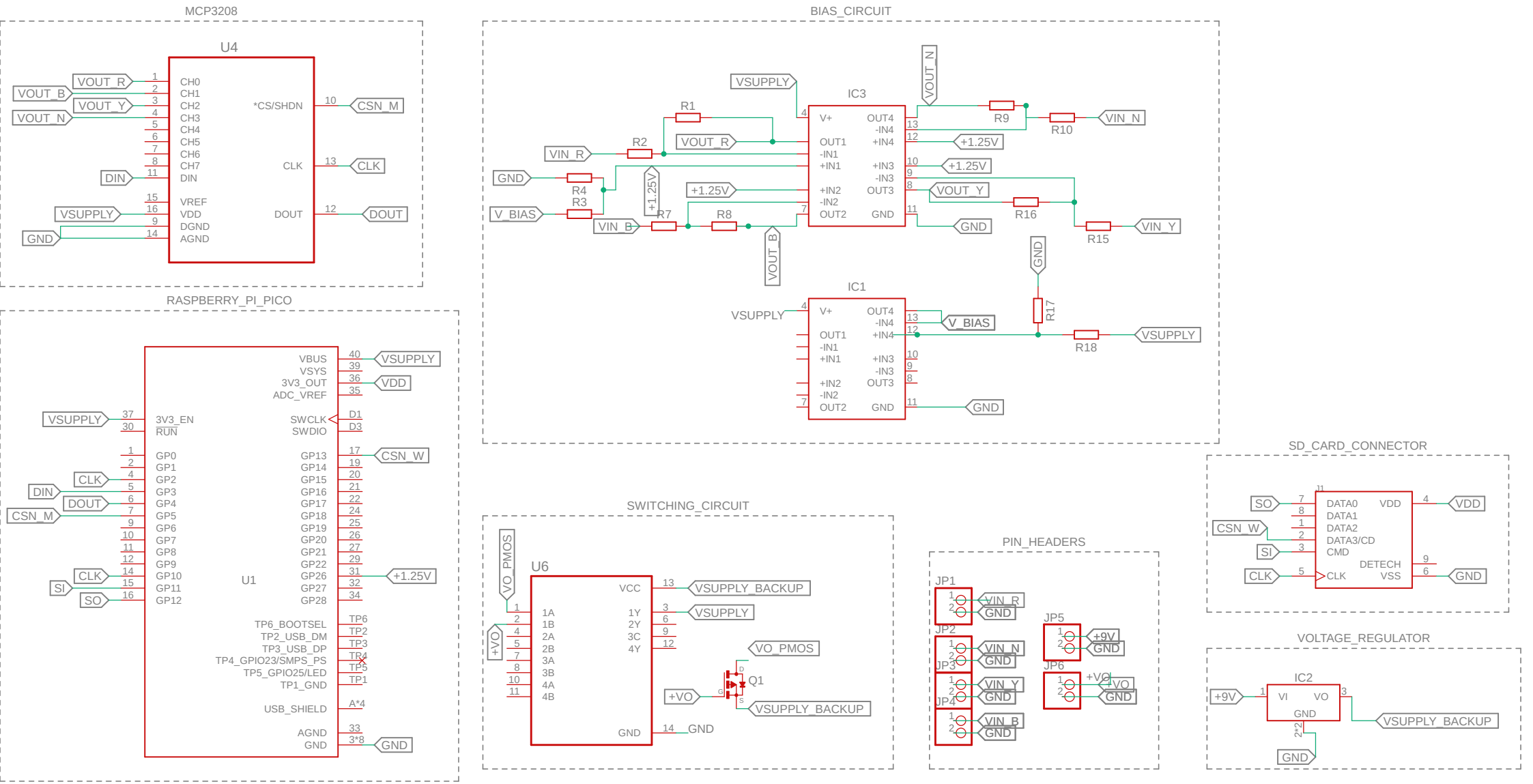


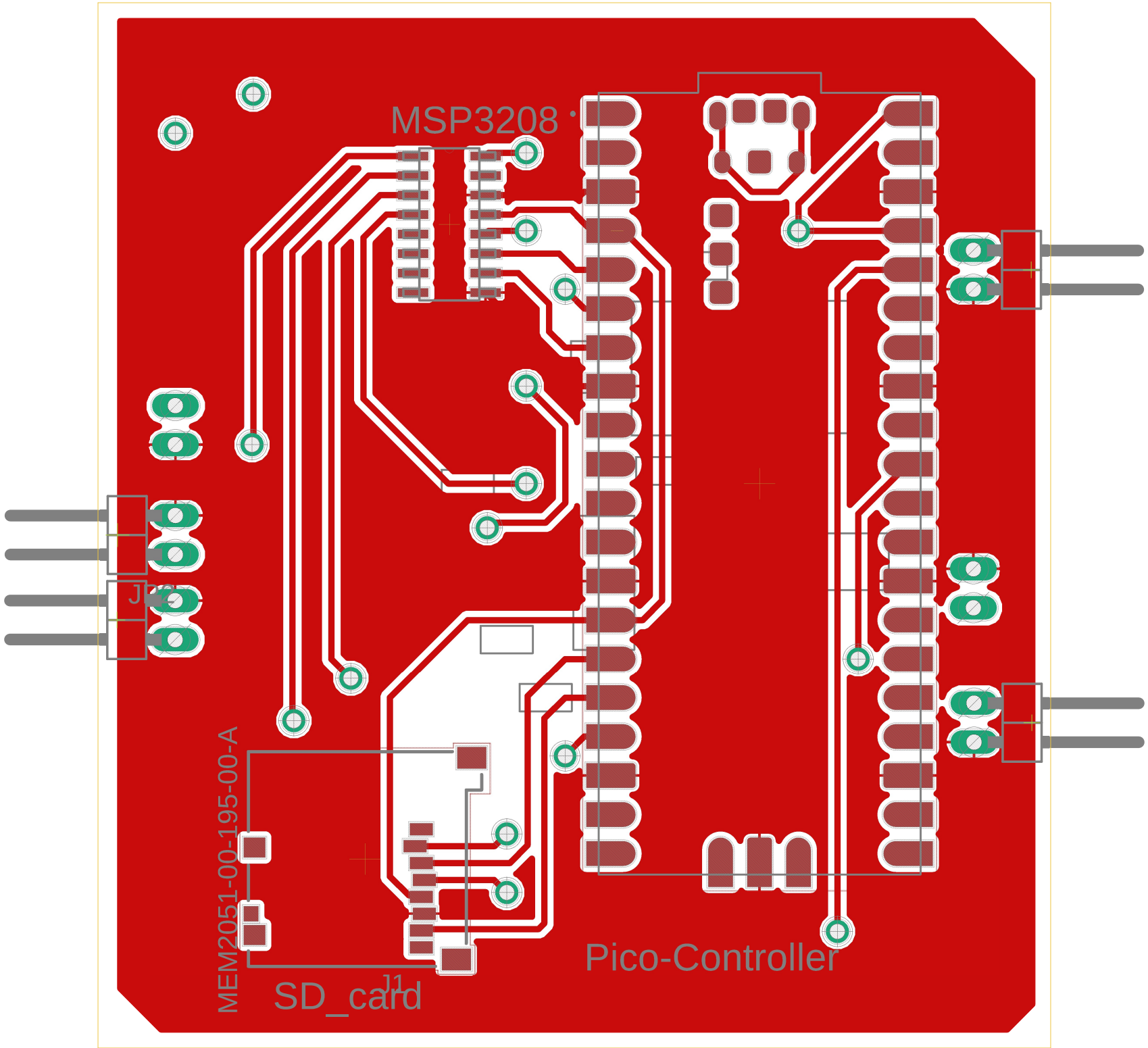
Next steps

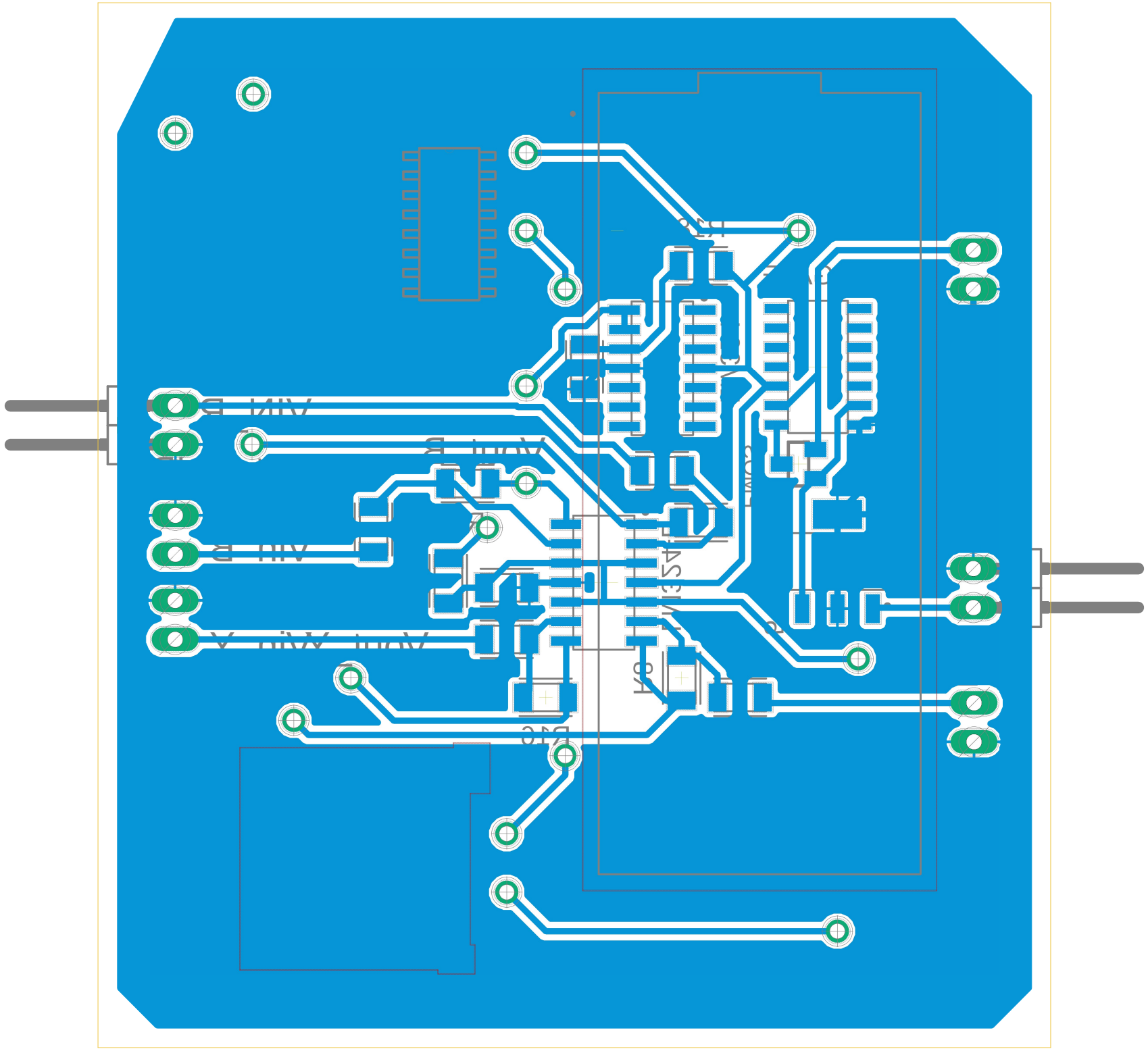
- We finished the testing by obtaining the current value from the four current wires through biasing circuitry and finding their rms values in the raspberry pi pico board.
- Remaining tests are sending the data(rms values) to a tcp server and also storing the data locally in the sd card.
- The code for sending data to the server and storing data locally in the sd card will be done by Gnanendar and Bhuvana Chandra.(approx timeline from mar 27–apr 3)
- 3D modelling ie., the outer body, holding the entire circuitry inside will be done by both Bhuvana Chandra and Avinash.(approx timeline from mar 27–apr 3 while Bhuvana and Gnanendar working on other stuff as mentioned above).
- After apr 3 we will be combining all the tested components and finally deploying the device into working state and keeping it in the body designed(this will be done between apr 4-apr 9) All three members work simultaneously.
- Between apr 10-11 we'll prepare for the presentation and report.

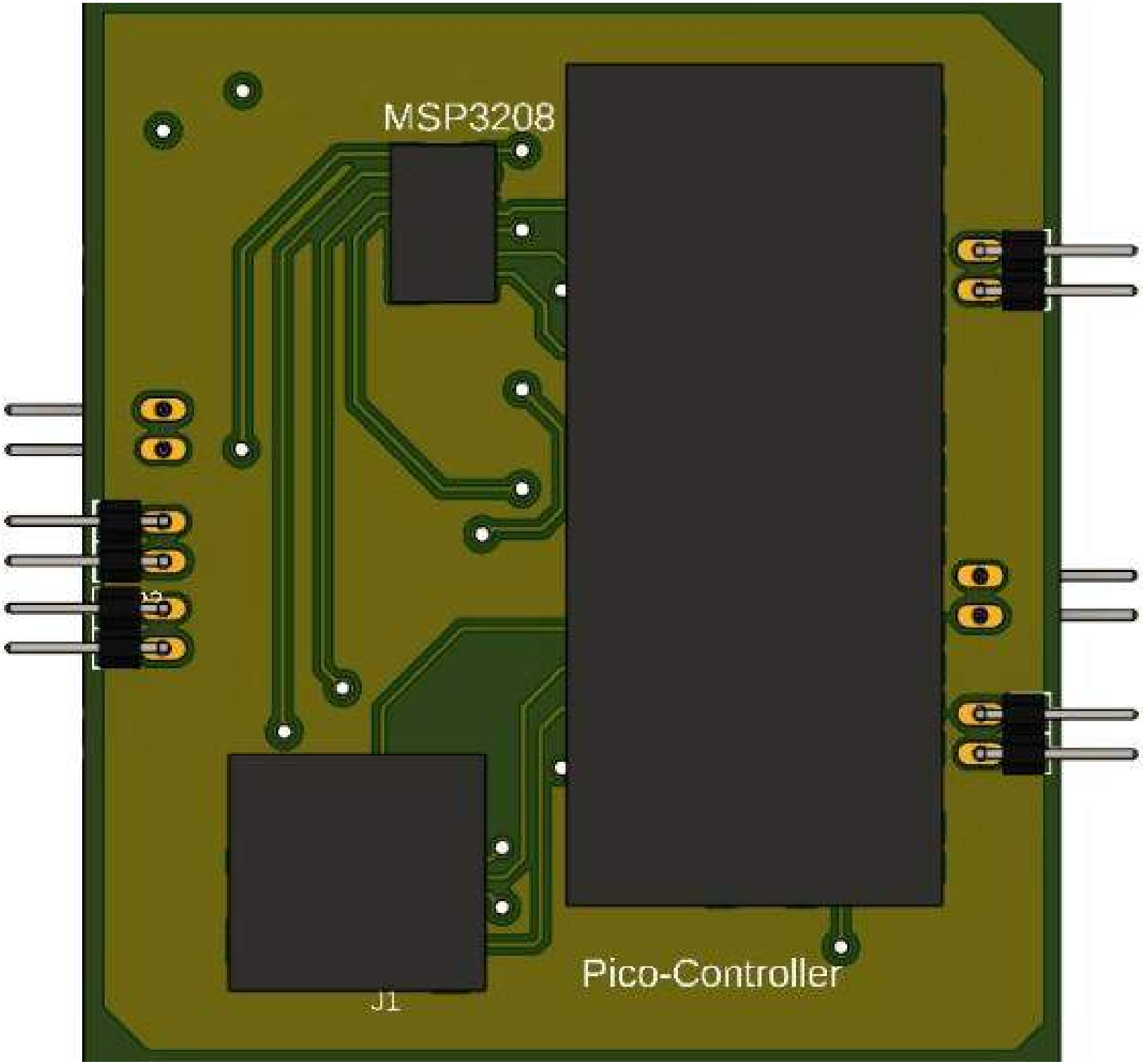
Links to demo videos

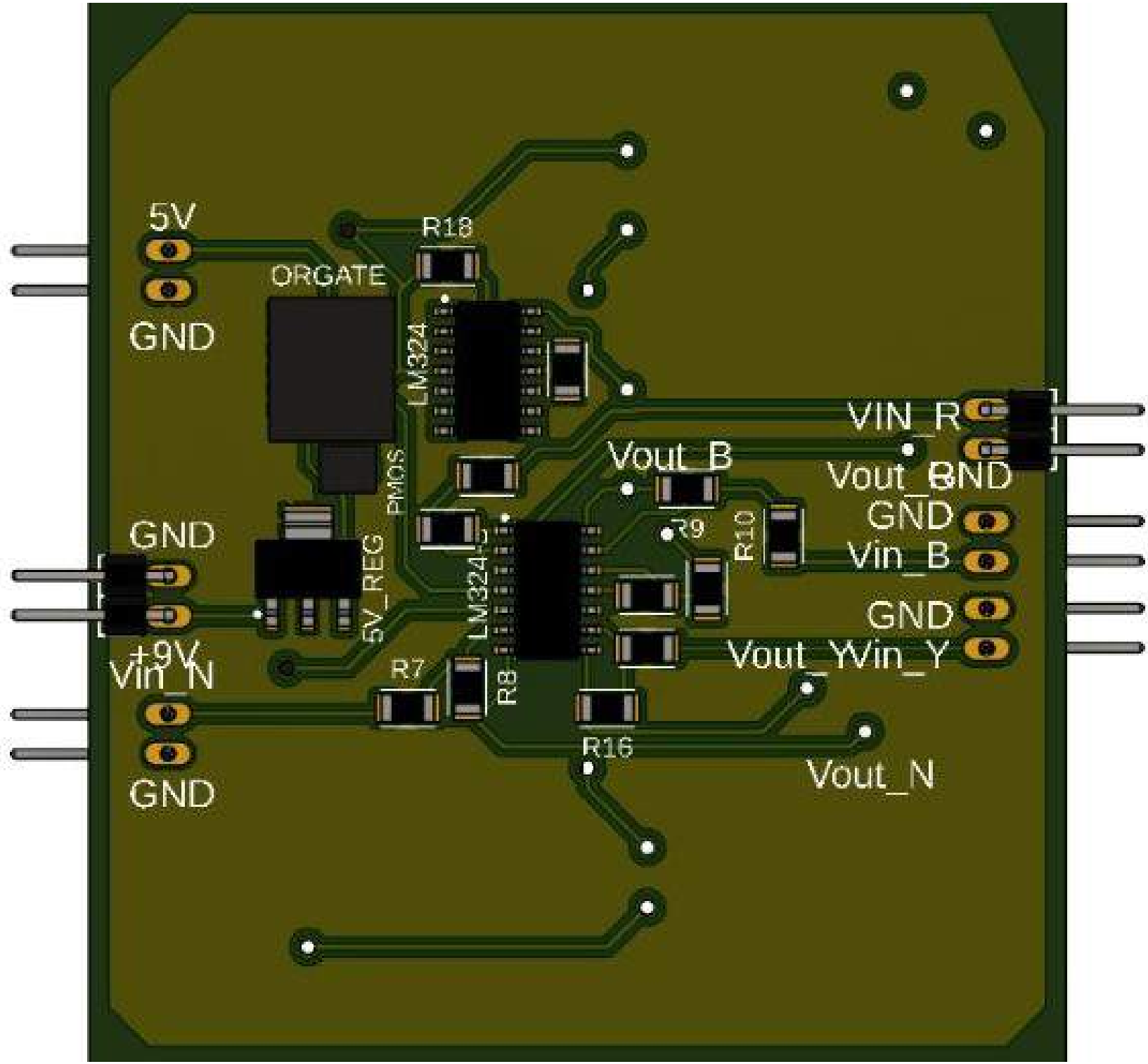
Here is the video link for testing the bias circuit and clamp - https://drive.google.com/file/d/1HCyp1g0kZuDCJ6-PMIx4grrj25Cr7hmB/view?usp=share_link


MCP3208
U4
VOUT_R
VOUT_B
VOUT_Y
VOUT_N
DIN
VSUPPLY
GND
CH0
CH1
CH2
CH3
CH4
CH5
CH6
CH7
DIN
VREF
VDD
DGND
AGND
*CS/SHDN
CLK
DOUT
CSN_M
CLK
DOUT
BIAS_CIRCUIT
VSUPPLY
IC3
VOUT_N
R1
R2
VOUT_R
VIN_R
GND
R4
R3
V_BIAS
VIN_B
+1.25V
R7
R8
VOUT_B
+1.25V
+1.25V
VOUT_Y
GND
R9
R10
VIN_N
+1.25V
R16
R15
VIN_Y
IC1
GND
R17
R18
V_BIAS
VSUPPLY
GND
RASPBERRY_PI_PICO
U1
VSUPPLY
3V3_EN
RUN
VBUS
VSYS
3V3_OUT
ADC_VREF
SWCLK
SWDIO
VDD
CSN_W
CLK
DIN
DOUT
CSN_M
CLK
SI
SO
+1.25V
TP6_BOOTSEL
TP2_USB_DM
TP3_USB_DP
TP4_GPIO23/SMPS_PS
TP5_GPIO25/LED
TP1_GND
USB_SHIELD
AGND
GND
SWITCHING_CIRCUIT
VO_PMOS
U6
+VO
VCC
VSUPPLY_BACKUP
VSUPPLY
VO_PMOS
Q1
+VO
VSUPPLY_BACKUP
GND
GND
PIN_HEADERS
JP1
VIN_R
GND
JP2
VIN_N
GND
JP3
VIN_Y
GND
JP4
VIN_B
GND
JP5
+9V
GND
JP6
+VO
+VO
GND
SD_CARD_CONNECTOR
J1
SO
CSN_W
SI
CLK
DATA0
DATA1
DATA2
DATA3/CD
CMD
CLK
VDD
DETECH
VSS
VDD
GND
VOLTAGE_REGULATOR
IC2
+9V
VI
VO
GND
VSUPPLY_BACKUP
GND


MSP3208
J1
MEM2051-00-195-00-A
SD_card
J1
Pico-Controller




MSP3208
Pico-Controller
J1


5V
GND
R18
ORGATE
LM324
VIN_R
Vout_B
Vout_B
GND
Vin_B
GND
Vout_Y
Vin_Y
R9
R10
PMOS
5V_REG
GND
+9V
Vin_N
GND
R7
R8
LM324_2
R16
Vout_N