NC State University
Department of Electrical and Computer Engineering
ECE 461/561: Spring 2019 (Dr. Dean)

Project #2: ENERGY ANALYSIS & OPTIMIZATION

by

<< SALONI SHAMBHUWANI >>

NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this test or assignment."
Student's electronic signature: __Saloni Shambhuwani____
(sign by typing your name)
Course number: _______ECE-561__________

Introduction

In this project I tried to optimize the power utilization of FRDM-KL25Z to calculate and reduce the average power used by system. The base code had an average power of about 4.7 to 4.8mW which got reduced to about 325uW. That is the power consumption on an average reduced by about 93% overall. More optimizations can be achieved by disabling accelerometer which we are not using in this experiment (saving power there), eliminating floating point math, etc. To reduce power consumption I made changes in LPTMR fields, LED period and duration, power mode and clock setup.

Power Optimization (in order performed, including dead ends (ineffective attempts))

Below is the table of all the optimizations performed to get as close to the expected average power:

Step	Otimization description	Average Power after optimization	Average Power Reduction % After every optimization
	Initial code(Lab 4 code)	4.76-4.8mW	
Opt1	Changing the LPTMR frequency	2.305-2.8mW	~42%
Opt2	Put the processor into a lower power stop mode.	1.4-1.7mW	~39%
Opt3	Changing clock setup which lets the processor wake up quickly	299-324uW	~81%

Starter code:

Figure 1 is the power schematic of KL25Z microcontroller board:

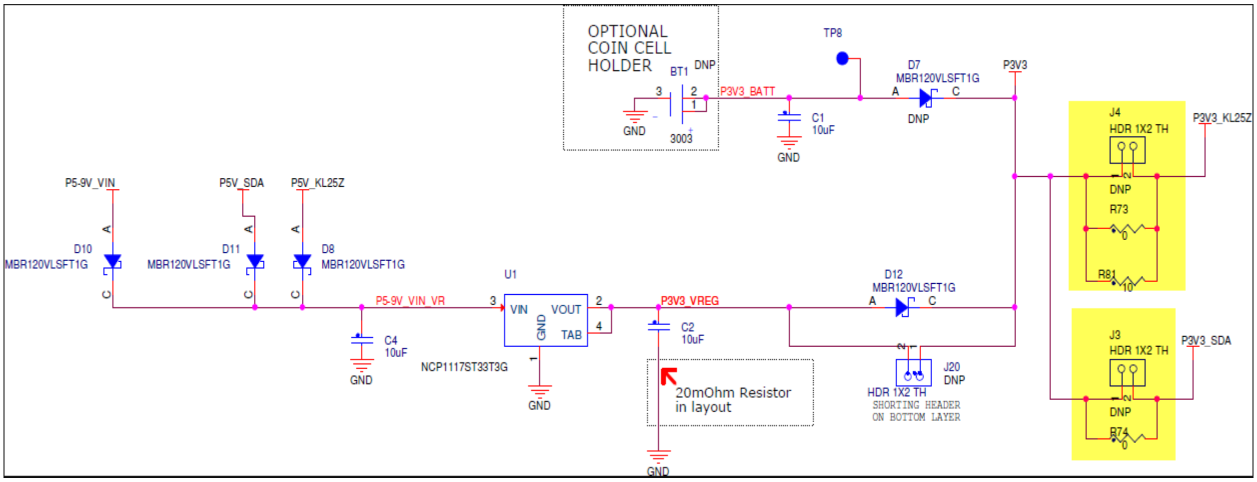
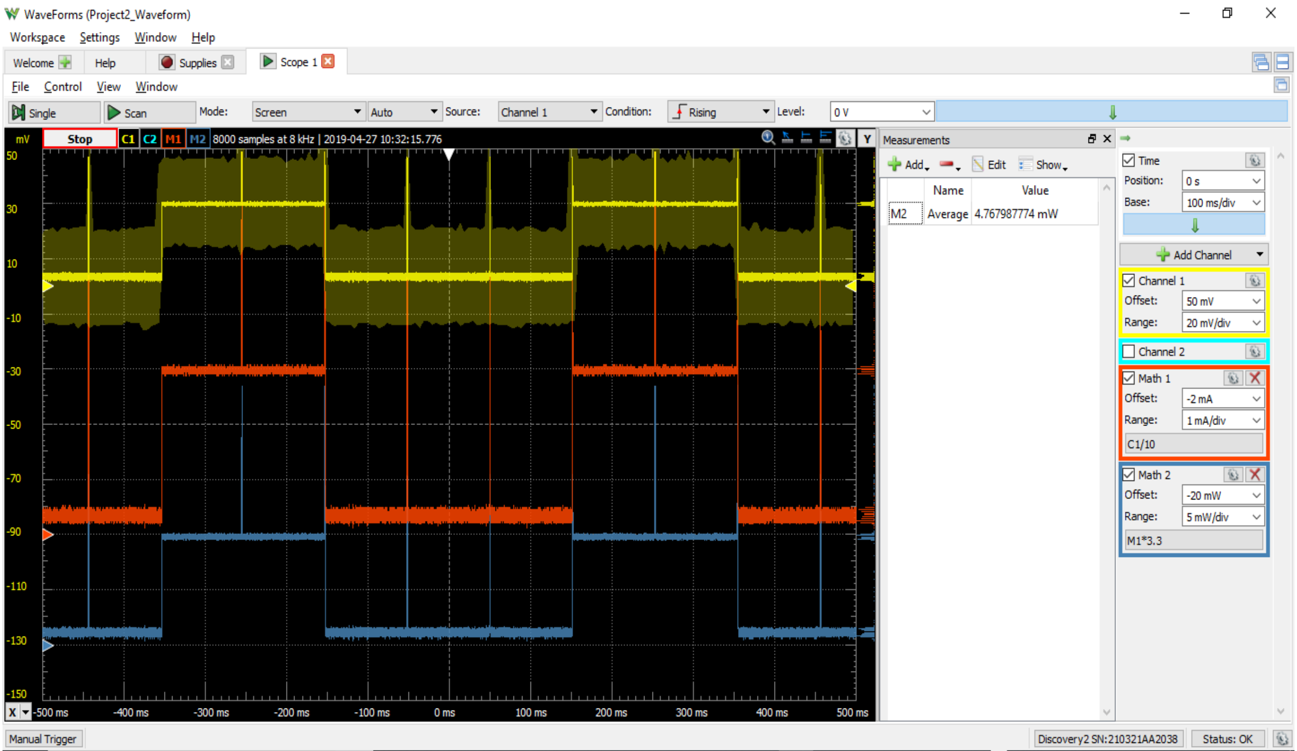


Figure 1: Freedom KL25Z power supply schematic

There is a lot which can be done to cut the power consumption, without impacting the application functionality or usability. There were already made optimizations for example removing the current flow for openSDA port by erasing the traces from jumpers highlighted above. A lot of power was saved due to this. Thus the base code gave the average power of about 4.8mW. For reading the current we connected the resistor of very small resistance (10ohms)in series with P3V3 pin. This current helped me analyze the prime areas of power consumption. The power consumption can be due to default power mode which allows many components to consume power that is not needed in the application. Another being the frequency of wakeup unit and the clock setup which allows the system to work at 48Mz (we don't need this high frequency for our application.). Power consumption can also be due to heavy load on computation (mostly floating point). I tried to touch all the parts and achieved the average power reduction more than expected. The power model of the base code is shown below:

	A	B	C	D	E	F	G	H	I	J
1	Power Model:	Initial State - w/o optimizations								
2				Measured				Calculated		
3	State	Current(mA	Voltage(mV	Voltage(V)	Time(duration	Frequency(events	Power(mW)	Duty Cycle	Avg.Power(mW)	Fraction of Total Avg.Power
4	LED On,MCU Running	5.994	59.94	3.3	1.61	6	19.7802	0.00966	0.191076732	0.039583858
5	LED Off,MCU Running	3.244	32.44	3.3	1.669	4	10.7052	0.006676	0.071467915	0.014805444
6	LED On,MCU Sleeping	3.134	31.35	3.3	99.72	4	10.3422	0.39888	4.125296736	0.854605159
7	LED Off,MCU Sleeping	0.222	2.221	3.3	99.94	6	0.7326	0.59964	0.439296264	0.091005539
8									**4.827137647**	



The power model shows that most of the average power is consumed in LED ON and MCU sleep state. This led me to the optimization 1. By reducing the time LED is on there is a chance of lower power consumption. It happen the instinct to work well! The average power for the same state reduced after 1st optimization. This can be seen in power model after optimization 1 described below.

Optimization #1: Changing the LPTMR frequency and reducing the time LED is ON

By changing the frequency I increased the number of times the wakeup event happens. The change was in CMR register field "LPTMR0->CMR = 4; " So now the interrupt is generated every 5 clock ticks. While to still keep the LED sampling every 500 ms and consume less power in sleep I increased the period of LED to 50 and reduced the duration the LED is on by 1.

So now the LED is ON for less amount of time and it is sampling every 500ms (LED blinking still visible). Below is the power model obtained after performing the 1st optimization

Power Model:	Optimization_1								
	Measured					Calculated			
State	Current(mA)	Voltage(mV)-F	Voltage(V	Time(duration pe	Frequenc	Power(mW)	Duty Cycle	Avg.Power(mW)	Fraction of Total Avg.Power
LED On,MCU Running	5.308	53.08	3.3	0.891	4	17.5164	0.003564	0.06242845	0.021781682
LED Off,MCU Running	2.764	27.65	3.3	0.891	96	9.1212	0.085536	0.780190963	0.272213578
LED On,MCU Sleeping	3.173	31.73	3.3	9.028	2	10.4709	0.018056	0.18906257	0.065965131
LED Off,MCU Sleeping	0.6206	6.21	3.3	9.14	98	2.04798	0.89572	1.834416646	0.640039609
								2.866098629	

After analyzing the average power in each state it seemed tricky how LED OFF , MCU sleep mode can take highest average power. After reading and understanding the power modes I realized the average power in this mode can be high due to default power mode causing system to consume power unnecessarily. So I decided to go for optimization two i.e. entering KL25z into lower power modes. Some functionality in the system is affected in lower power mode but we don't need them either in our application so we can go ahead with lower power stop mode.

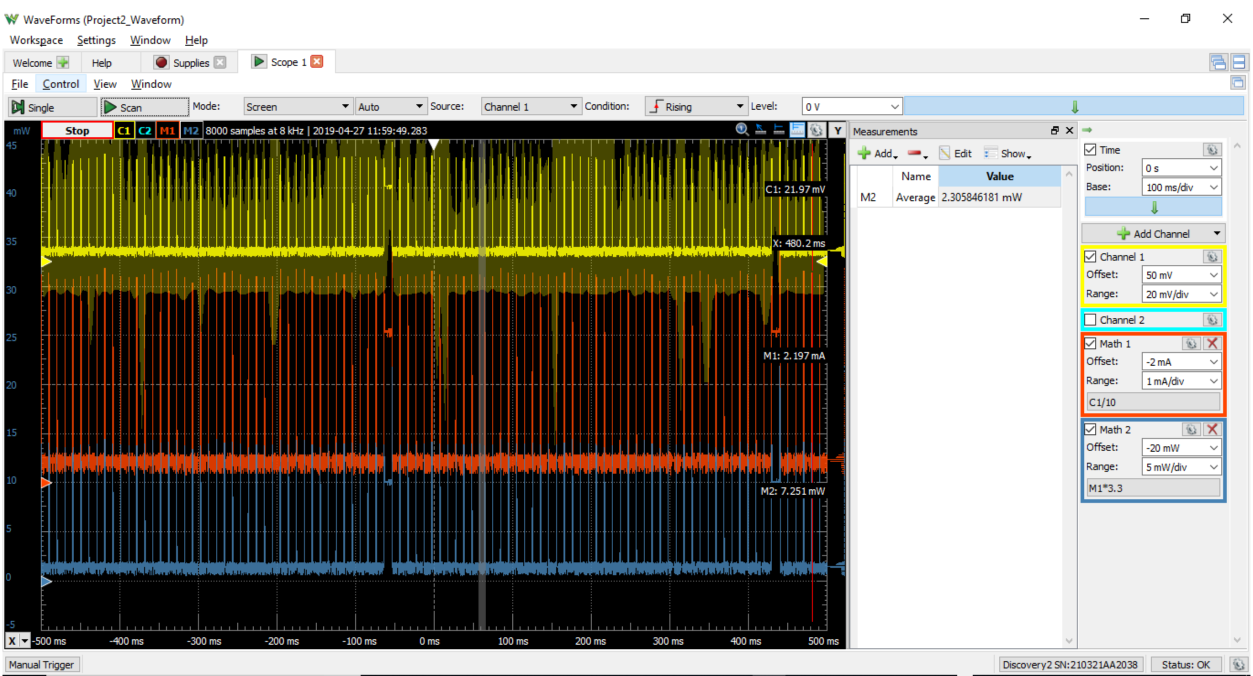


Table below shows the reduction in average power consumption after reducing the time LED is ON.

Step #Optimization	Average Power Before Optimization	Average Power after Optimization	%improvement
Changing the LPTMR frequency	4.76-4.8mW	2.305-2.8mW	~42%

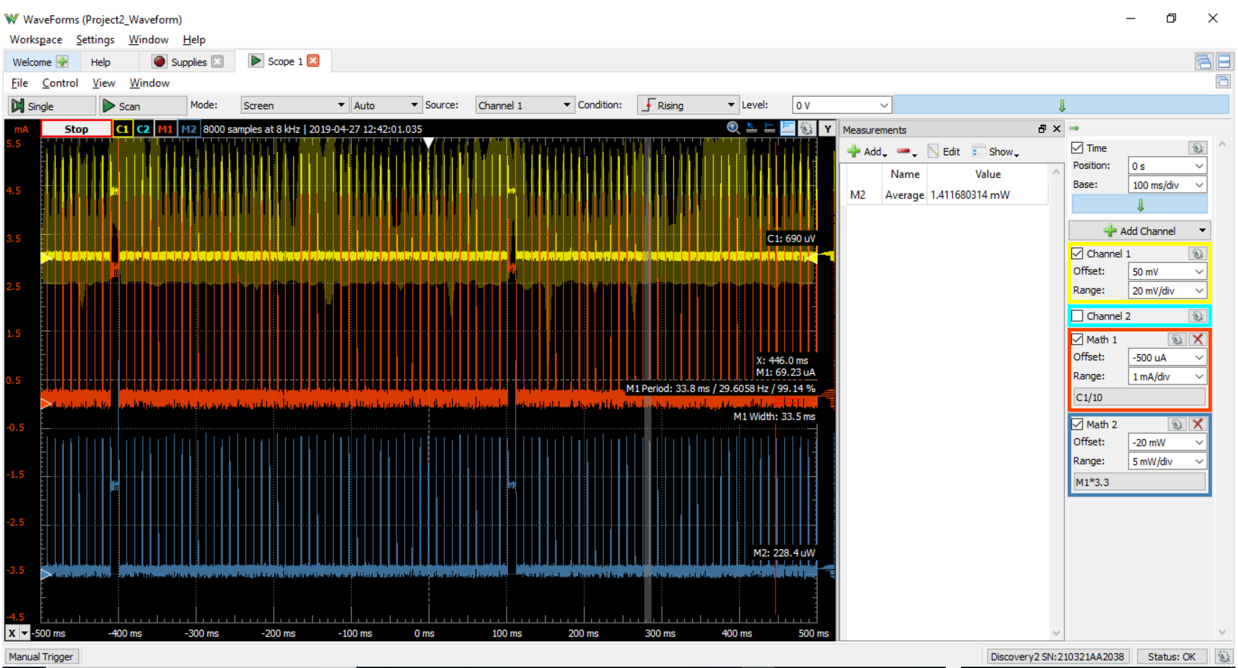
Optimization #2: Setting KL25z to work on Low Power stop Mode

FRDM-KL25z has a low power microcontroller and comes with embedded debugging interface. So 1 of the most advantageous feature of using low power mode stop mode allowed the reduction in average power consumption by approximately 40%. Below is the power model after this optimization. Average power reduced to 1.4-1.7mW which is a considerable reduction. Following setting in code was required to run the processor in lower mode:

SMC->PMCTRL &= ~0x07; SMC->PMCTRL |= 0x03; // resetting the old power mode bits and setting the bits for low power mode.

Power Model:	Optimization_2								
			Measured					Calculated	
State	Current(mA)	Voltage(mV)	Voltage(V	Time(duration	Frequency(e	Power(m	Duty Cycle	Avg.Power(mW)	Fraction of Total Avg.Power
LED On,MCU Running	5.392	53.93	3.3	0.756	4	17.7936	0.003024	0.053807846	0.030768593
LED Off,MCU Running	2.357	23.57	3.3	0.954	96	7.7781	0.091584	0.71234951	0.407338213
LED On,MCU Sleeping	2.866	28.66	3.3	9.061	2	9.4578	0.018122	0.171394252	0.098007266
LED Off,MCU Sleeping	0.2618	2.612	3.3	9.39	100	0.86394	0.939	0.81123966	0.463885928
								1.748791268	

From the above power model we can see there is still most average power in LED OFF MCU SLEEPING state though some reduction has been achieved. Reducing the MCU core frequency will help in reducing power consumption more, thus the optimization 3. The improvement (average power reduction) is shown below in the table.



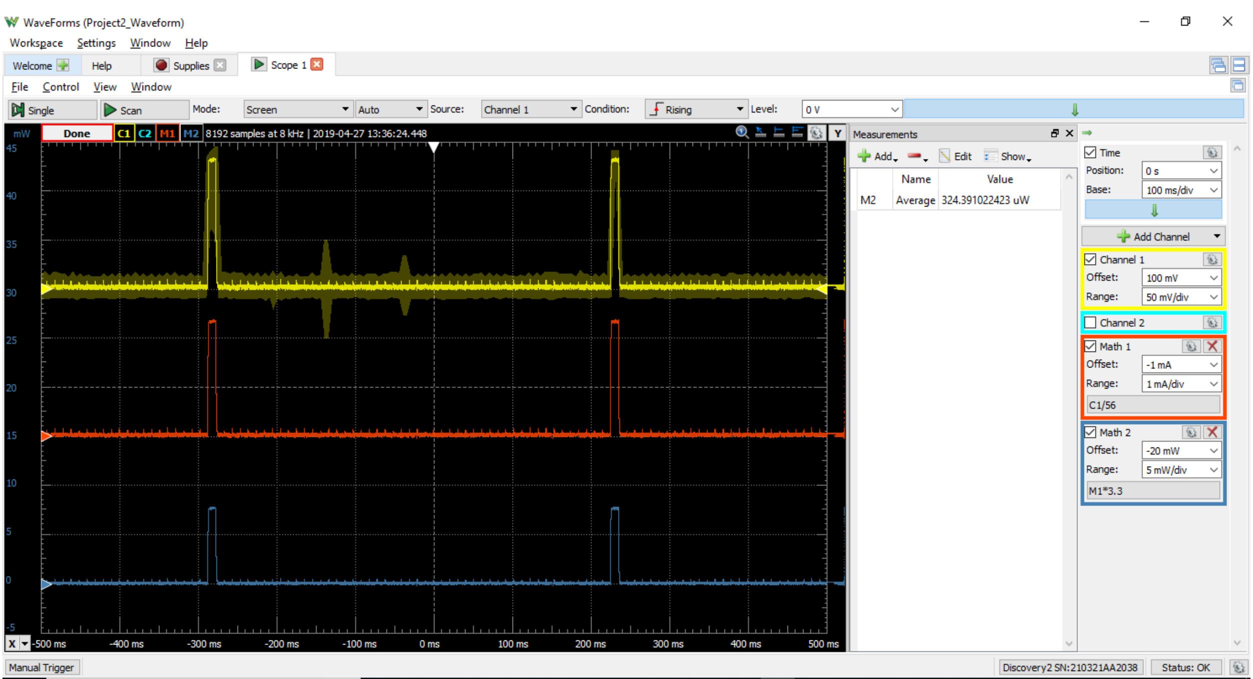
Step #Optimization	Average Power Before Optimization	Average Power after Optimization	%improvement
Setting KL25z to work on Low Power stop Mode	2.305-2.8mW	1.4-1.7mW	~39%

Optimization #3: Changing clock setup

Current consumption (therefore power) usually is linear to the used clock speed. So reducing the clock speed should greatly reduce the current. I do not need that 48 MHz clock speed i.e. clock setup 1 therefore I reduced it to 20Mz i.e. clock setup 0. The power model after this optimization is shown below:

Power Model:	Optimization_3								
	Measured					Calculated			
State	Current(mA)	Voltage(m	Voltage(V)	Time(duratic	Frequency(	Power(mW)	Duty Cycle	Avg.Power(n	Fraction of Total Avg.Power
LED On,MCU Running	2.367	129.5	3.3	0.202	4	7.8111	0.000808	0.00631137	0.021042257
LED Off,MCU Running	0.122	6.85	3.3	0.261	96	0.4026	0.025056	0.01008755	0.033632123
LED On,MCU Sleeping	2.37	128.5	3.3	9.712	2	7.821	0.019424	0.1519151	0.506488661
LED Off,MCU Sleeping	0.0407	5.99	3.3	9.8	100	0.13431	0.98	0.1316238	0.438836959
								0.29993782	

While doing this optimization the current measurement came out to be negative so I eplaced the 10 ohm resistor with higher value resistor (56 ohms). This can be seen in the current calculation in the waveform below.

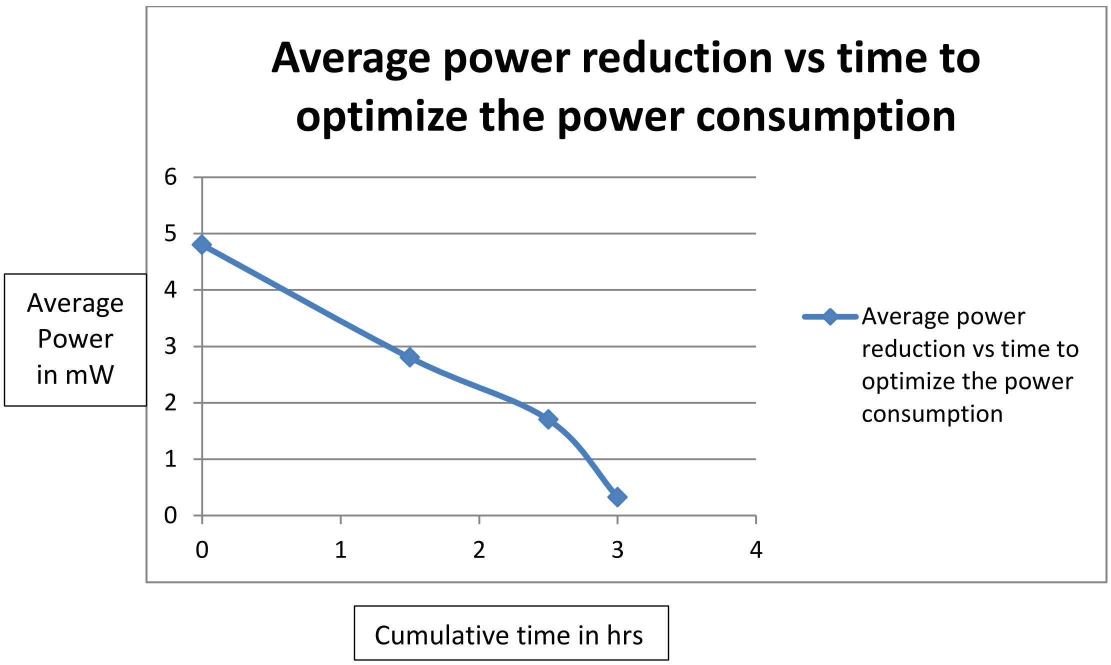


The overall power is reduced by highest amount by doing this optimization. This is because the frequency is directly proportional to current consumed. The average power reduced by about 80% than last optimized value.

Step #Optimization	Average Power Before Optimization	Average Power after Optimization	%improvement
Changing clock setup	1.4-1.7mW	299-324uW	~81%

Time for development vs Average Power

Cumulative Development Time	Average power
0	4.8mW
1.5hrs	2.8mW
2.5hrs	1.7mW
3 hrs	324uW



Lessons learned in this project, and how you might do things differently next time

To understand the power utilization in various parts/components of KL25z I did the trials for first optimization. The best way I could understand the proper flow was after measuring the current flow across all the components I realized when the board is consuming highest power. To perfectly do the optimization 1 this approach of measuring current across all prime components helped a lot. So while doing power optimization the first thing, I learned, was to realize where the current is being consumed (current directly proportional to power and average power is proportional to frequency)and if that part of system is required to be running for the particular application. The brought me down to look on basics always.

Another thing I learned was there was excess leakage while the hardware wire connections were not properly done. Even though the power consumption seemed less but the calculation showed higher power consumption on average. So I realized by the end that wire and resistor connection should be done in a way that current doesn’t flow unnecessarily causing higher power consumption. Keeping a good isolation is helpful.