Project Report

Embedded Systems and Designs

Barnabh Chandra Goswami - 19UEC161

Ayush Maherchandani - 19UEC160

Instructor: Dr. Deepak Nair

Department of Electronics and Communication Engineering
The LNMIIT Jaipur, Rajasthan

April 30, 2022

Declaration

This report has been prepared based on my work. Where other published and unpublished source materials have been used, these have been acknowledged.

Student Name	Roll Number
Barnabh Chandra Goswami	**19UEC161**
Ayush Maherchandani	**19UEC160**

Date of Submission: 28 / 04 / 2022

Table of Contents

List of Figures

List of Tables

Abstract

In this report we will talk about measuring the light intensity using Light Dependent Resistor (LDR) or Photoresistor and TM4C123GH6PM which is a 32-bit Arm Cortex-M4F based microcontroller. We will first start with the observations by defining the relation between light intensity and resistance so that we can study the graph by manually plotting it, after noting the observations we will try to calculate the constants which we will obtain from our graph and also check its accuracy with the help of datasheet of LDR then we will make a potential divider circuit and take the analog reading of the voltage from it using TM4C123GH6PM and then we will manipulate that voltage to find the resistance of the LDR and ultimately doing calculations we will find the experimental value of Light Intensity for a given LDR, which is the aim of our project.

Component Name

The components which we have used in this project are as follows:

1. Light Dependent Resistor (LDR) or Photoresistor
2. Tiva Microcontroller Board (TM4C123GH6PM)
3. X Ohm Resistance
4. Breadboard
5. Jumper Wires
6. 3.3 Volt Power Supply

Chapter 1: Introduction

We are living in the world where everything goes to be automatic from your washing machine to your AC. This present world revolves around the word automation and the devices that are automated are said to be of next generation because they restrict the involvement of humans. They are saving time and cost by being more efficient than the manual ones. But lighting systems have yet to make its move. The main objective of this project is to implement a Light Intensity Measuring Meter which is interfaced to a Tiva board. As the surrounding light changes, the light intensity gradually changes accordingly by which the resistance of LDR also changes.

We will try to find out the intensity of light using a given resistance for which we needed Light Dependent resistor also known as the Photoresistor. It has many uses like burglar alarm circuits, security purposes etc, it is a two terminal passive electronic component that is made up of a semiconductor element or compound. As the name implies the resistance of the Photoresistor depends upon the intensity of light projected onto it. With the increase in the intensity of light the resistance of the light dependent resistor decreases. Different actions can be performed through this for example by feeding the voltage signal into the microcontroller the circuit can be used to determine the darkness in the room and turn on or off the light, a similar kind of idea which we will use in this project and the microcontroller which we use here is TM4C123GH6PM.

TM4C123GH6PM is a 32-bit Arm Cortex-M4F based MCU with 80-MHz, 256-KB Flash, 32-KB RAM, 2 CAN, RTC, USB, 64-Pin. This microcontroller is targeted for industrial applications, including remote monitoring. Using this we will make a circuit of potential divider which is a passive linear circuit that produces an output voltage (V_{out}) that is a fraction of its input voltage (V_{in}). Voltage division is the result of distributing the input voltage among the components of the divider, this will help us in calculations of Light Intensity.

We will combine all of this, that is LDR, TIva Board and using potential divider circuit on breadboard. The goal is to make a Light Intensity Meter which can give its reading through the resistance of LDR and by calculations we can get the value of Light Intensity.

Chapter 2: Theory

We will first discuss some concepts which we needed to understand before heading to our aim of the project. Like some basic principles working behind Light Dependent Resistor, Tiva Microcontroller, Potential Divider etc.

2.1 Light Dependent Resistor (LDR) or Photoresistor

2.1.1 What is Light Dependent Resistor (LDR) or Photoresistor

A Photoresistor or Light Dependent Resistor is an electronic component that is sensitive to light. When light falls upon it, then the resistance changes. Values of the resistance of the LDR may change over many orders of magnitude the value of the resistance falling as the level of light increases.

Figure 2-1. Light Dependent Resistor (LDR) or Photoresistor

It is not uncommon for the values of resistance of an LDR or photoresistor to be several Megaohms (MΩ) in darkness and then to fall to a few hundred Ohms (Ω) in bright light. With such a wide variation in resistance, LDRs are easy to use and there are many LDR circuits available. LDRs are made from semiconductor materials to enable them to have their light sensitive properties. Many materials can be used, but one popular material for these Photoresistors is cadmium sulphide (CdS), although the use of these cells is now restricted in Europe because of environmental issues with the use of cadmium.

2.1.2 How a LDR works

It is relatively easy to understand the basics of how an LDR works. It is first necessary to understand that an electrical current consists of the movement of electrons within a material. Good conductors have a large number of free electrons that can drift in a given direction under the action

of a potential difference. Insulators with a high resistance have very few free electrons, and therefore it is hard to make the them move and hence a current to flow.

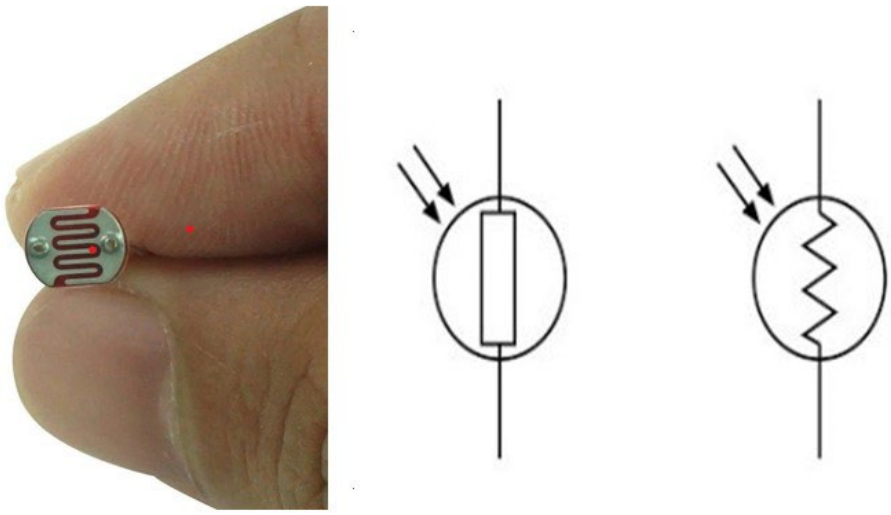

Figure 2-2. LDR circuit symbols

An LDR or photoresistor is made from semiconductor material with a high resistance. It has a high resistance because there are very few electrons that are free and able to move, the vast majority of the electrons are locked into the crystal lattice and unable to move.

Therefore, in this state there is a high LDR resistance. As light falls on the semiconductor, the light photons are absorbed by the semiconductor lattice and some of their energy is transferred to the electrons. The amount of energy transferred to the electrons gives some of them sufficient energy to break free from the crystal lattice so that they can then conduct electricity. This results in a lowering of the resistance of the semiconductor and hence the overall LDR resistance, and as more light shines on the LDR semiconductor, so more electrons are released to conduct electricity and the resistance falls further.

2.1.3 Relationship between LDR and Resistance

$$L = B\,(R_{ldr})^{m} \quad , \quad \text{where } m < 0$$

Where **L** is the Intensity of light in **Lux (lx)**,

B and **m** are the constant obtained from graph,

$\mathbf{R_{ldr}}$ is the Resistance of LDR.

2.1.4 Basic idea of values for different types of LDR

NTE Type	Diameter	Max. DC Voltage	Power Dissipation (mW)	Light Resistance (10Lux)(KΩ)	Dark Resistance (MΩ)	$\gamma \frac{100}{10}$	Response Times (ms)	
							Increase	Decrease
02-LDR1	.201 (5.0)	150	100	50 – 100	5.0	0.8	20	30
02-LDR2	.201 (5.0)	150	90	5 – 10	0.2	0.5	30	30
02-LDR3	.201 (5.0)	150	100	100 – 200	10.0	0.9	20	30
02-LDR4	.201 (5.0)	150	100	30 – 50	3.0	0.7	20	30
02-LDR12	.472 (12.0)	250	200	5 – 10	1.0	0.6	30	30
02-LDR13	.472 (12.0)	250	200	10 – 20	2.0	0.6	30	30
02-LDR14	.472 (12.0)	250	200	30 – 50	5.0	0.7	30	30
02-LDR15	.472 (12.0)	250	200	50 – 100	8.0	0.8	30	30
02-LDR20	.787 (20.0)	500	500	5 – 10	1.0	0.6	30	30
02-LDR21	.787 (20.0)	500	500	10 – 20	2.0	0.6	30	30
02-LDR22	.787 (20.0)	500	500	30 – 50	5.0	0.7	30	30
02-LDR23	.787 (20.0)	500	500	50 – 100	8.0	0.8	30	30

Figure 2-3. Information on different types of LDR with some important values

By this figure we can get the idea about what will be the Light Resistance and Dark Resistance of some certain LDR, for example in our case the Dark Resistance of our LDR is 2.5 MΩ.

2.1.5 Applications of LDR

LDR are found in many different applications and can be seen in many different electronic circuit designs. They are widely used in many different items of electronic equipment and circuit designs including photographic light meters, fire or smoke alarms as well as burglar alarms, and they also find uses as lighting controls for street lamps, they can also be used to detect nuclear radiation.

2.2 Tiva Board – TM4C123GH6PM Microcontroller

2.2.1 What is Tiva C Series TMC4123GH6PM

TM4C123GH6PM is a 32-bit Arm Cortex-M4F based MCU with 80-MHz, 256-KB Flash, 32-KB RAM, 2 CAN, RTC, USB, 64-Pin. The clocked source of the I/O port circuitry of the TIVA board can be enabled using the RCGCGPIO register. The clock source of a pin should be disabled if an I/O port is not used to save power.

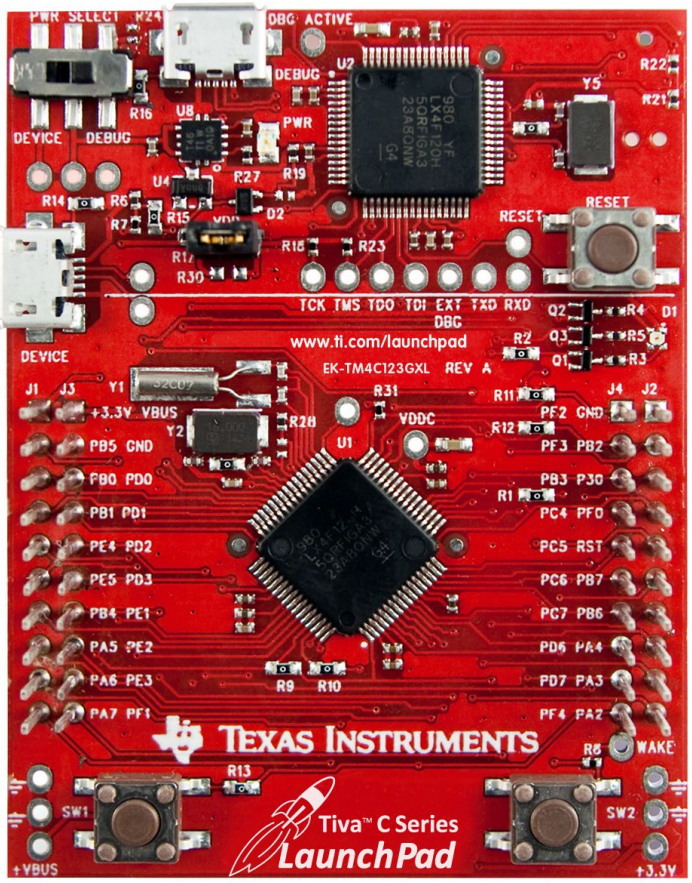

Figure 2-4. TM4C123GH6PM Microcontroller

The TM4C123GH6PM microcontroller combines complex integration and high performance with the features shown below.

Features	**Description**
Performance	
Core	ARM Cortex-M4F processor core
Performance	80-MHz operation; 100 DMIPS performance
Flash	256 KB single-cycle Flash memory
System SRAM	32 KB single-cycle SRAM
EEPROM	2KB of EEPROM
Internal ROM	Internal ROM loaded with TivaWare™ for C Series software
Communication Interfaces	
Universal Asynchronous Receivers/Transmitter (UART)	Eight UARTs
Synchronous Serial Interface (SSI)	Four SSI modules

Controller Area Network (CAN)	Two CAN 2.0 A/B controllers
Universal Serial Bus (USB)	USB 2.0 OTG/Host/Device
System Integration	
Micro Direct Memory Access (µDMA)	ARM® PrimeCell® 32-channel configurable µDMA controller
General-Purpose Timer (GPTM)	Six 16/32-bit GPTM blocks and six 32/64-bit Wide GPTM blocks
Watchdog Timer (WDT)	Two watchdog timers
Hibernation Module (HIB)	Low-power battery-backed Hibernation module
General-Purpose Input/Output (GPIO)	Six physical GPIO blocks
Advanced Motion Control	
Pulse Width Modulator (PWM)	Two PWM modules, each with four PWM generator blocks and a control block, for a total of 16 PWM outputs.
Quadrature Encoder Interface (QEI)	Two QEI modules
Analog Support	
Analog-to-Digital Converter (ADC)	Two 12-bit ADC modules, each with a maximum sample rate of one million samples/second
Analog Comparator Controller	Two independent integrated analog comparators
Digital Comparator	16 digital comparators
JTAG and Serial Wire Debug (SWD)	One JTAG module with integrated ARM SWD
Package Information	
Package	64-pin LQFP
Operating Range (Ambient)	Industrial (-40°C to 85°C) temperature range Extended (-40°C to 105°C) temperature range

Table 2-1. TM4C123GH6PM Microcontroller Features

2.2.2 Pin Diagram for TMC4123GH6PM

The figure below shows pin diagram for TMC4123GH6PM.

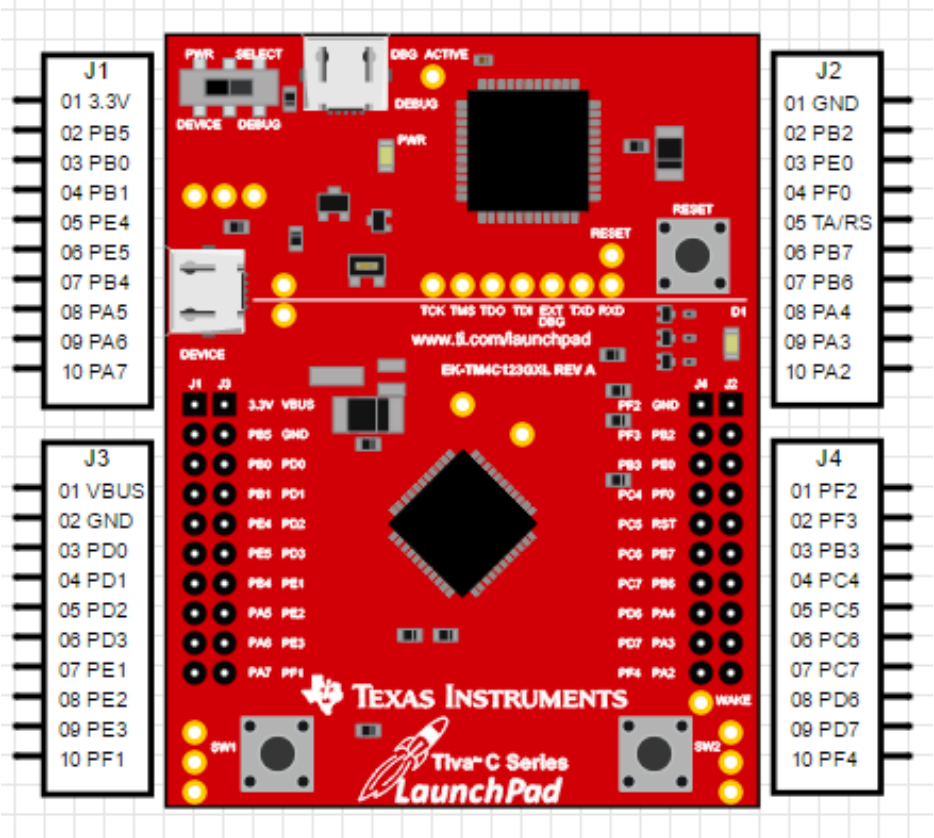

Figure 2-5. Pin Diagram of TMC4123GH6PM

2.2.3 Applications of TMC4123GH6PM

TM4C123GH6PM microcontroller is targeted for industrial applications. These include remote monitoring, electronic point-of-sale machines, test and measurement equipment, network appliances and switches, factory automation, HVAC and building control, gaming equipment, motion control, transportation, and fire and security.

2.3 Potential Divider

2.3.1 What is Potential Divider

A Voltage Divider (also known as a Potential Divider) is a passive linear circuit that produces an output voltage (V_{OUT}) that is a fraction of its input voltage (V_{IN}) or we can say it is a simple circuit that uses resistors (or thermistors / LDR's) to supply a variable potential difference. A simple example of a voltage divider is two resistors connected in series, with the input voltage applied across the resistor pair and the output voltage emerging from the connection between them.

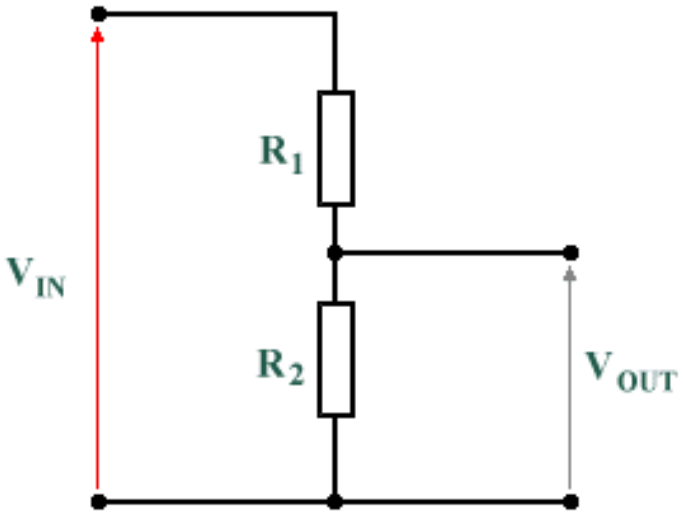

Figure 2-6. A Potential Divider Circuit

Since we know from our basic knowledge that $V_{IN} = I (R_1 + R_2)$, therefore putting the value of I from this (V_{IN} equation) to $V_{OUT} = I R_2$, we get

$$V_{OUT} = V_{IN} \left(\frac{R_2}{R_1 + R_2} \right)$$

Where V_{OUT} and V_{IN} are output and input voltages respectively,

R_1 and R_2 are resistance in series according to V_{IN}.

2.3.2 Applications of Potential Divider

A potential divider can be used for many applications, including control of temperature in a fridge or as audio-volume controls, a Multimeter and Wheatstone bridge, for measuring the resistance of the sensor, active devices in amplifiers, for measurement of voltages, a potentiometer is used as a variable voltage divider in the volume control of many radios etc.

Chapter 3: Light Intensity vs Resistance

We know that there is a relation between Light Intensity and Resistance from the formula discussed in **2.1.3.**

$$L = B\,(R_{ldr})^{m} \quad , \quad \text{where } m < 0$$

In general, we know that when Light Intensity increases the resistance of LDR decrease, reason we have already discussed in **2.1.2.**

The general graph between Light Intensity and Resistance of LDR looks like

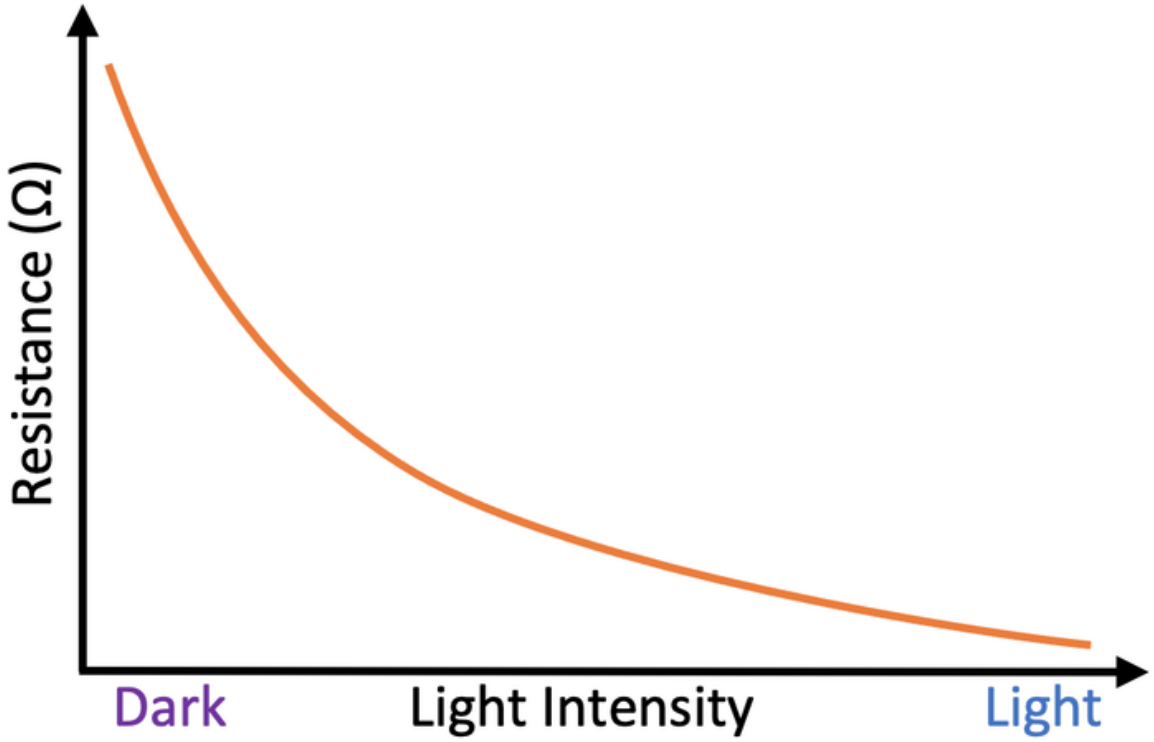

Figure 3-1. Graph between Light Intensity and Resistance of LDR

Since the value of m is always less than zero it means that there is always a inverse relationship between Light Intensity and Resistance of LDR. Now we try to make a graph by ourselves using some equipment.

3.1 Experimental Readings and Observations

To find the relation between Light Intensity and Resistance of LDR on graph by manually plotting it we will use a Lux Meter (Lux is a measuring unit of Light Intensity), in this scenario our smart phone act as Lux Meter, what we have to do is to simply download the app from the Google Play Store, here in this case we used the app named "**Lux**" (Link: Lux Light Meter Pro - Apps on Google Play) after that we will take Multimeter which will take readings of resistance between the

legs of LDR and we will use the breadboard so that the LDR can be easily placed and we have no difficulties in taking readings through Multimeter.

Now we will take the experimental readings of the Resistance of LDR.

Light Intensity (in Lux)	**Resistance of LDR (in MΩ or kΩ)**
0	2.5 MΩ
10	16.7 kΩ
20	14.0 kΩ
30	12.83 kΩ
40	11.12 kΩ
50	10 kΩ
60	9.3 kΩ
70	8.5 kΩ
80	8 kΩ
90	7.35 kΩ
100	6.9 kΩ
150	5.5 kΩ
200	5 kΩ
250	4.45 kΩ
300	4 kΩ
400	3.35 kΩ
500	2.93 kΩ
600	2.65 kΩ
700	2.375 kΩ
800	2.25 kΩ
900	2.04 kΩ
1000	1.95 kΩ
2000	1.33 kΩ
3000	0.96 kΩ

Table 3-1. Experimental Readings between Light Intensity (in Lux) and Resistance of LDR (in MΩ or kΩ)

Now plotting these values in graph one by one we get inverse curve which looks like this.

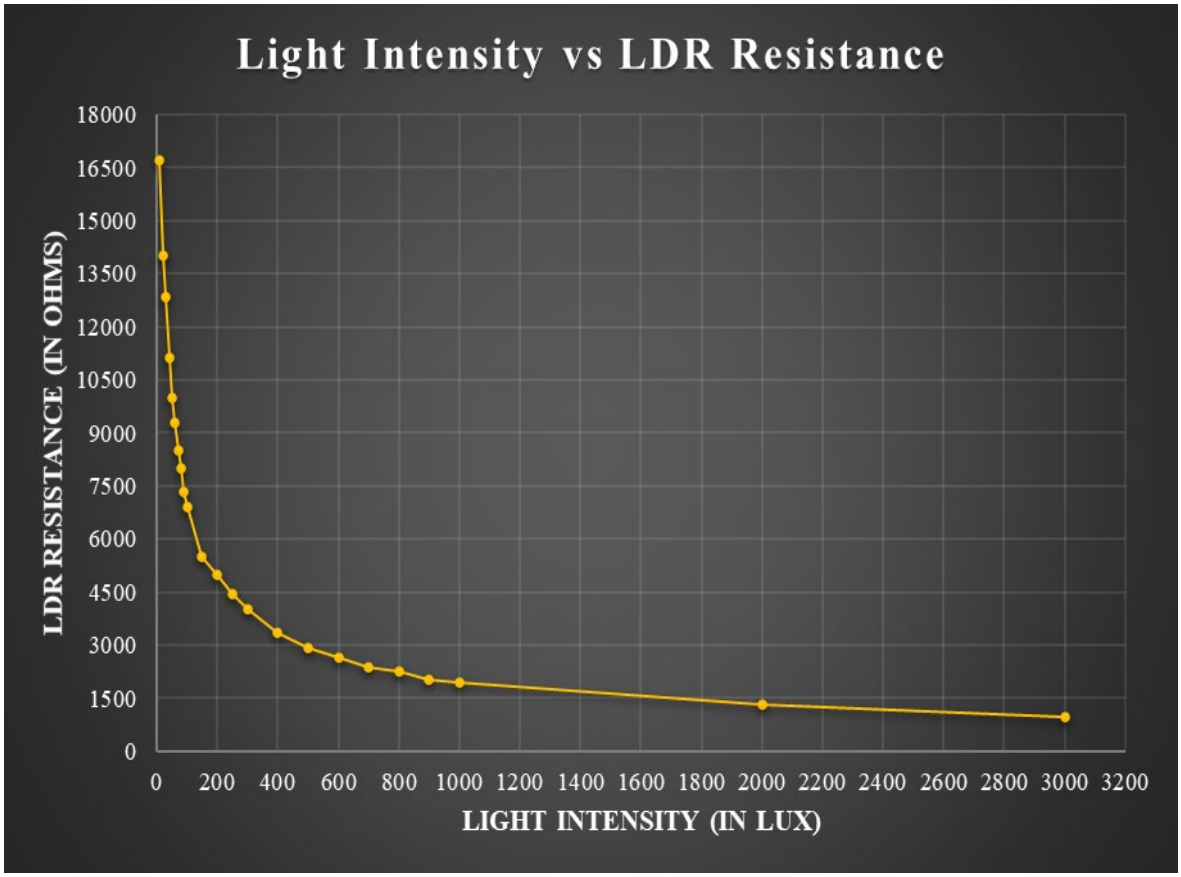

Figure 3-2. Experimental Graph between Light Intensity (in Lux) and LDR Resistance (in MΩ or kΩ)

The graph we obtained above pretty much follows the LDR inverse relationship behaviour, we are getting inverse exponential graph which is as expected since Light Intensity depends inverse exponentially on LDR Resistance by the relation 2.1.3, now after obtaining this type of graph it means that we have taken our readings precisely.

3.2 Graphical study and calculation of Constants

After obtaining experimental graph we will first look into the formula once again,

$$\mathbf{L = B\,(R_{ldr})^{m},}$$

taking both sides natural logarithm, we get,

$$\mathbf{\log_e L = m \log_e(R_{ldr}) + \log_e B,}$$

the equation we get resembles the general equation of a straight line which is **y = mx + c**, therefore on comparing these equations we can say that **y** is $\mathbf{\log_e L}$, constant **m** remains **m**, **x** is $\mathbf{\log_e(R_{ldr})}$ and constant **c** is $\mathbf{\log_e B}$.

Hence it means that if we plot a graph between x and y here in this case $\log_e(R_{ldr})$ and $\log_e L$ respectively, then we may also get a straight line graph. In ideal scenario it will be a perfect straight line. So now we will first make a table where we can write the $\log_e(R_{ldr})$ and $\log_e L$ values.

$\log_e L$ (Natural Logarithm of L, Y-Axis)	**$\log_e(R_{ldr})$ (Natural Logarithm of R_{ldr}, X-Axis)**
0	14.7318
2.30259	9.72316
2.99573	9.54681
3.4012	9.45954
3.68888	9.3165
3.91202	9.21034
4.09434	9.13777
4.2485	9.04782
4.38203	8.9872
4.49981	8.90246
4.60517	8.83928
5.01064	8.6125
5.29832	8.51719
5.52146	8.40066
5.70378	8.29405
5.99146	8.11672
6.21461	7.98276
6.39693	7.88231
6.55108	7.77275
6.68461	7.71869
6.80239	7.62071
6.90776	7.57558
7.6009	7.19293
8.00637	6.86693

Table 3-2. Data obtained of Light Intensity (in Lux) and Resistance (in MΩ or kΩ) by taking their natural logarithms

Now plotting these values in graph one by one we get a straight-line which looks like this.

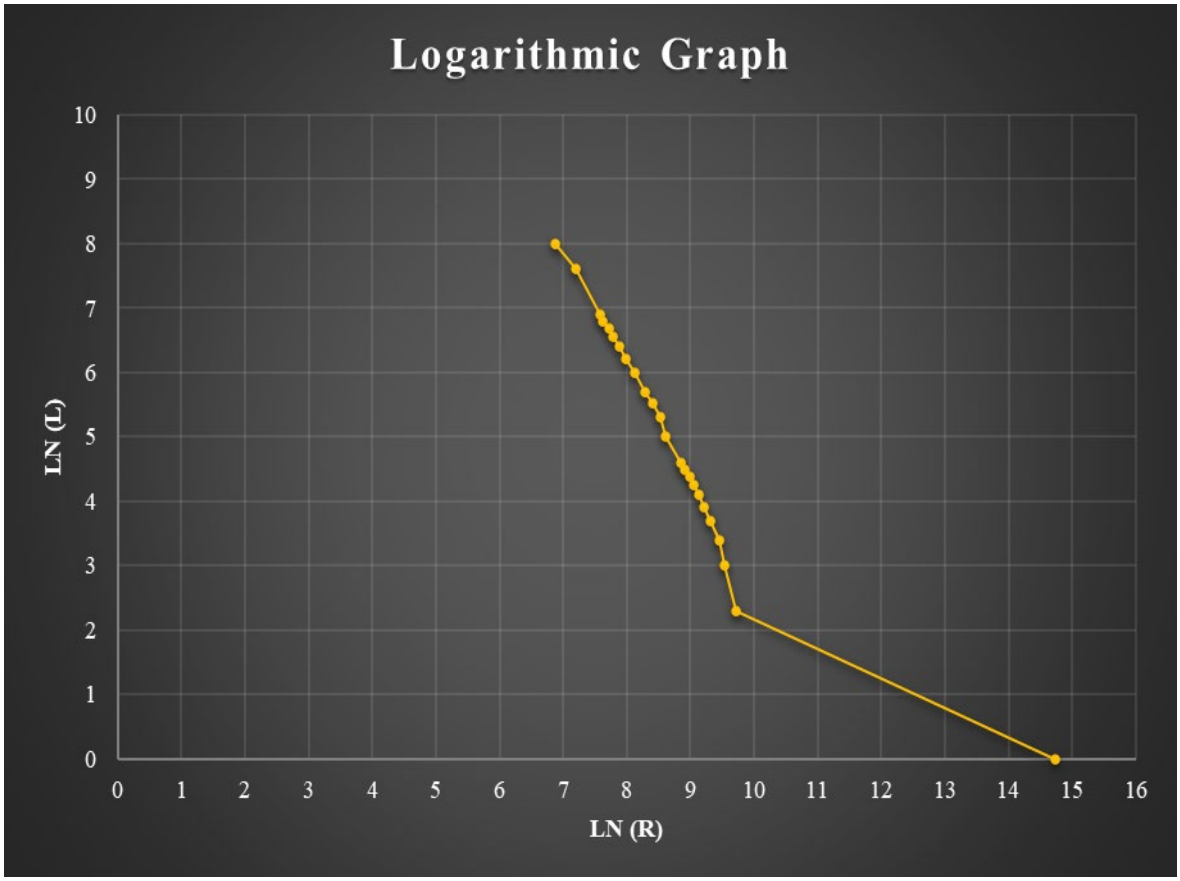

Figure 3-3. Natural Logarithmic Graph between $\log_e L$ and $\log_e(R_{ldr})$

As we can see in the graph that we are getting almost a straight line, this shows the accuracy of our readings, more the graph curve is straight more the readings are precise.

Now heading to the calculations part, we will now try to calculate the value of **m** and **c**, **m** can be calculated using slope formula.

$$m = \frac{y2 - y1}{x2 - x1}$$

And **c** can be calculated using **point slope-intercept form** which is $\mathbf{y - y_1 = m\,(x - x_1)}$, since we have to find intercept at **y – axis** therefore value of x must be 0. And at x = 0, y is **c.**

So, by manipulating the **point slope-intercept equation** we get,

$$c = -mx_1 + y_1$$

Now noting the values of **m** and **c** with every observation we have taken; we get a table below.

$\log_e L$ (y_1)	$\log_e(R_{ldr})$ (x_1)	$y_2 - y_1$	$x_2 - x_1$	Slope (m)	Constant (c)
0	14.7318	-2.3026	5.00864	-0.4597	29.46360258
2.30259	9.72316	-0.6931	0.17635	-3.9305	21.74891309
2.99573	9.54681	-0.4055	0.08727	-4.646	22.08935749
3.4012	9.45954	-0.2877	0.14304	-2.0112	22.3202803
3.68888	9.3165	-0.2231	0.10616	-2.102	22.32188059
3.91202	9.21034	-0.1823	0.07257	-2.5123	22.33270375
4.09434	9.13777	-0.1542	0.08995	-1.7138	22.36988392
4.2485	9.04782	-0.1335	0.06062	-2.2026	22.34413813
4.38203	8.9872	-0.1178	0.08474	-1.3899	22.35642028
4.49981	8.90246	-0.1054	0.06318	-1.6677	22.30472085
4.60517	8.83928	-0.4055	0.22677	-1.788	22.28372357
5.01064	8.6125	-0.2877	0.09531	-3.0184	22.23564204
5.29832	8.51719	-0.2231	0.11653	-1.9148	22.33270375
5.52146	8.40066	-0.1823	0.10661	-1.7102	22.32277967
5.70378	8.29405	-0.2877	0.17733	-1.6223	22.29188175
5.99146	8.11672	-0.2231	0.13396	-1.6658	22.2248958
6.21461	7.98276	-0.1823	0.10044	-1.8152	22.1801235
6.39693	7.88231	-0.1542	0.10956	-1.407	22.16155949
6.55108	7.77275	-0.1335	0.05407	-2.4697	22.09658577
6.68461	7.71869	-0.1178	0.09798	-1.2021	22.12198272
6.80239	7.62071	-0.1054	0.04512	-2.3351	22.04380494
6.90776	7.57558	-0.6931	0.38265	-1.8114	22.05892458
7.6009	7.19293	-0.4055	0.326	-1.2438	21.9867709
8.00637	6.86693	-	-	-	21.74023414
-	**Average value of m and c**			**-2.0119**	**22.4888964**

Table 3-3. Calculations done in excel of slope and intercept then finding their average values

After this, we finally get the values of **m** and **c** as **-2.0119** and **22.4888964** respectively.

Since **c** is $\log_e B$ therefore value of **B** we get is **5845256999.367113** or approximately **5.84 x 10^9**, now our final equation which we get from **2.1.3** becomes

$$L = (5845256999.367113)\,(R_{ldr})^{(-2.0119)},$$

We have taken such precise values because they are obtained from experimental readings and based on calculations therefore can produced errors if not taken precisely.

Chapter 4: Making of the overall circuit

We have obtained the values of **m** and **c** now we will make the overall circuit which consists of Tiva Board, Breadboard, Jumper Wires, 10 kΩ resistance, and an LDR.

4.1 Interfacing Tiva Board, LDR and Potential Divider

First of all, we will take input from PE3 pin in Tiva Board using Breadboard on which we have made our Potential Divider circuit. Then we take GND and V_{CC} connection from GND and V_{CC} pin in Tiva Board where V_{CC} is of 3.3 V.

4.2 What we will do

We will first make a Potential Divider circuit on Breadboard with Resistance of 10 kΩ and with LDR and V_{CC} = 3.3 Volt (because highest voltage measured by ADC of the Tiva board is 3.3 V).

We will take the input of the ADC as shown in the figure below and measure the voltage of the node by multiplying the value of ADC by **0.0008** (because the minimum value that can be measured by the ADC of Tiva Board is 0.0008 Volt). Refer to Figure 4-2. **Practical implementation of our project using Tiva and Breadboard**

Then we will measure the resistance of the LDR using the formula

$$\left(\frac{3.3\,V}{V_{out}} - 1\right) 10000\ Ohms$$

which is written in code 5.1 and then we will finally get the value of light intensity (in Lux) by using the relationship 2.1.3.

To give you an idea of an overall circuit, a circuit diagram in Proteus simulation is given below.

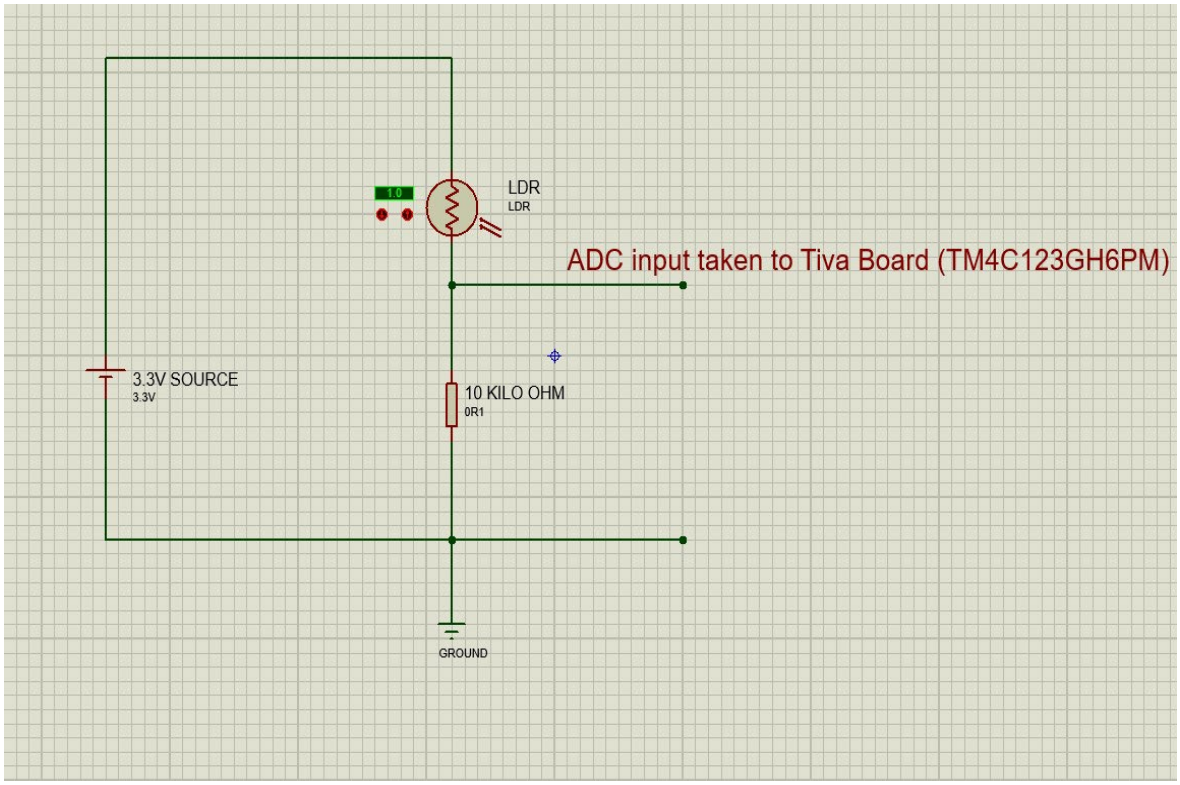

Figure 4-1. Overall diagram of our circuit which is to be made practically

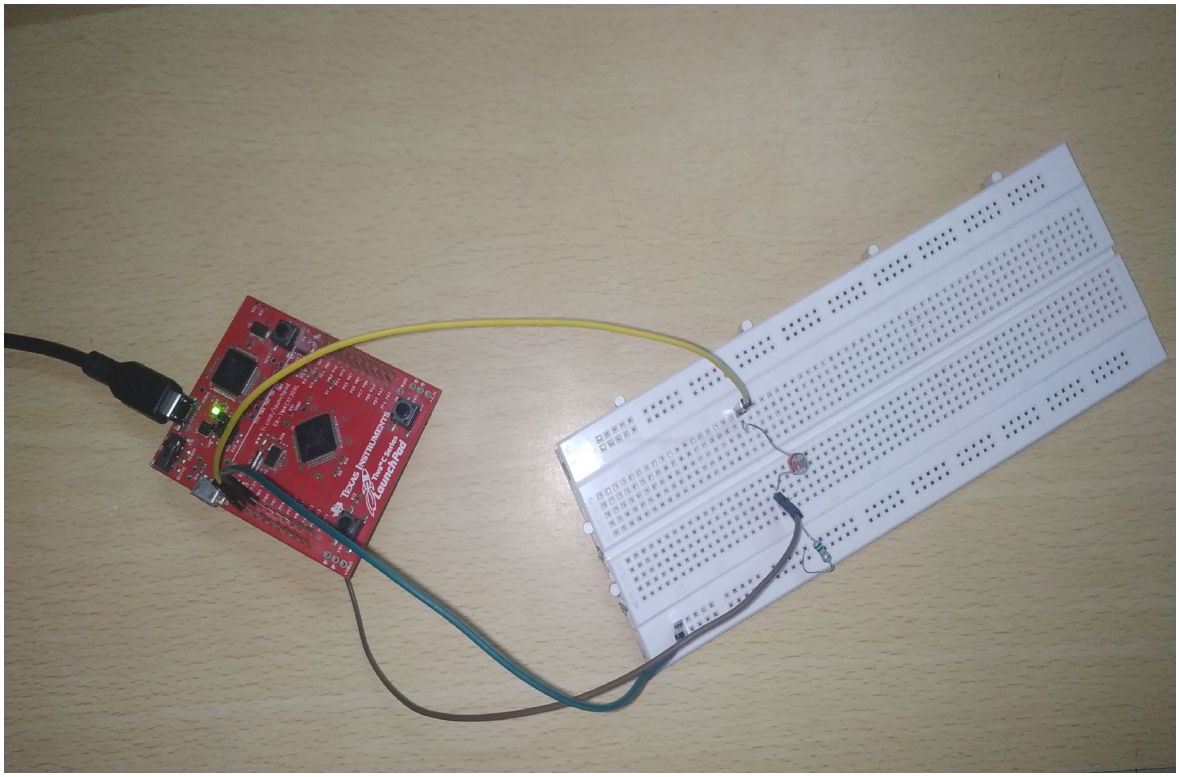

Figure 4-2. Practical implementation of our project using Tiva and Breadboard

Chapter 5: Coding on TM4C123GH6PM

We have written the code in Embedded C language on the IAR Embedded workbench IDE software platform (IAR Systems—Register for Evaluation)

5.1 Source Code

```
#include "TM4C123GH6PM.h"
#include <stdio.h>
#include <math.h>

float R_LDR;
char mesg[12];
float voltage;
float LUX;
float B = 5845256999.367113;
float m = -2.011862635;

int main(void)
{
     unsigned int adc_value;

    // The initialization sequence for the ADC is as follows:

    // 1. Enable the ADC clock using the RCGCADC register
    SYSCTL->RCGCADC |= (1<<0);     // Enabling the clock for AD0 //

    // 2. Enable the clock to the appropriate GPIO modules via the
RCGCGPIO register
    SYSCTL->RCGCGPIO |= (1<<4);    // Enable Clock to GPIOE //
    GPIOE->DIR &= ~(1<<1) ;        // Setting the Direction for the GPIO E
as Input //

    // 3. Set the GPIO AFSEL bits for the ADC input pins
    GPIOE->AFSEL |= (1<<3);          // enable alternate function for the
PE3 Pin //

    // 4. Configure the AINx signals to be analog inputs by clearing the
corresponding DEN bit in the GPIO Digital Enable (GPIODEN) register
    GPIOE->DEN &= ~(1<<3);         // disable digital function for the PE3
Pin //

    // 5. Disable the analog isolation circuit for all ADC input pins
that are to be used by writing step
    // "1" to the appropriate bits of the GPIOAMSEL register in the
associated GPIO block.

    GPIOE->AMSEL |= (1<<3);         // enable analog function for the PE3
Pin  //

    // The configuration for each sample sequencer should be as follows:

    // 1. Ensure that the sample sequencer is disabled by clearing the
corresponding ASENn bit in the ADCACTSS register.
    // Programming of the sample sequencers is allowed without having
them enabled.
```

```
    // Disabling the sequencer during programming prevents erroneous execution if a trigger event were to occur during the configuration process.
    ADC0->ACTSS &= ~(1<<3);         // disable SS3 during configuration //

    // 2. Configure the trigger event for the sample sequencer in the ADCEMUX register.
    ADC0->EMUX &= ~0xF000;     // Continuous Sampling on the SS3 //

    // 3. When using a PWM generator as the trigger source, use the ADC Trigger Source Select (ADCTSSEL) register to specify in which PWM module the generator is located.
    // The default register reset selects PWM module 0 for all generators.
    // Not needed

    // 4. For each sample in the sample sequence, configure the corresponding input source in the ADCSSMUXn register.
    ADC0->SSMUX3 = 0;          // get input from AIN0 //

    // 5. For each sample in the sample sequence, configure the sample control bits in the corresponding nibble in the ADCSSCTLn register.
    // When programming the last nibble, ensure that the END bit is set. Failure to set the END bit causes unpredictable behaviour.
    ADC0->SSCTL3 |= (1<<1)|(1<<2);          // take one sample at a time, set flag at 1st sample //

    // 6. If interrupts are to be used, set the corresponding MASK bit in the ADCIM register.
    // Not needed

    // 7. Enable the sample sequencer logic by setting the corresponding ASENn bit in the ADCACTSS register.
    ADC0->ACTSS |= (1<<3);          // enable ADC0 sequencer 3 //


    while(1)
    {
        ADC0->PSSI |= (1<<3);           // Enable SS3 conversion or start sampling data from AN0 //

        while((ADC0->RIS & 8) == 0) ;    // checking the 3rd bit of the RIS to see if the sampling is completed or not//

        adc_value = ADC0->SSFIFO3; // read ADC conversion result from SS3 FIFO //

        ADC0->ISC = 8;           // clear conversion clear flag bit //

        voltage = (adc_value * 0.0008); // convert digital value back into voltage //

        R_LDR = ((3.3/voltage) - 1)*10000; // calculating the value of resistance of LDR //

        LUX = B*pow(R_LDR,m) ; // Calculating the value of LUX //

    }

}
```

```c
#include "TM4C123GH6PM.h"
#include <stdio.h>
#include <math.h>

float R_LDR;
char mesg[12];
float voltage;
float LUX;
float B = 5845256999.367113;
float m = -2.011862635;



int main(void)
{
     unsigned int adc_value;

    // The initialization sequence for the ADC is as follows:

    // 1. Enable the ADC clock using the RCGCADC register
    SYSCTL->RCGCADC |= (1<<0);    // Enabling the clock for AD0 //

    // 2. Enable the clock to the appropriate GPIO modules via the RCGCGPIO register
    SYSCTL->RCGCGPIO |= (1<<4);  // Enable Clock to GPIOE //
    GPIOE->DIR &= ~(1<<1) ;      // Setting the Direction for the GPIO E as Input //

    // 3. Set the GPIO AFSEL bits for the ADC input pins
    GPIOE->AFSEL |= (1<<3);       // enable alternate function for the PE3 Pin //

    // 4. Configure the AINx signals to be analog inputs by clearing the corresponding DEN bit in the GPIO Digital Enable (GPIODEN) register
    GPIOE->DEN &= ~(1<<3);        // disable digital function for the PE3 Pin //

    // 5. Disable the analog isolation circuit for all ADC input pins that are to be used by writing step
    // "1" to the appropriate bits of the GPIOAMSEL register in the associated GPIO block.

    GPIOE->AMSEL |= (1<<3);       // enable analog function for the PE3 Pin  //
    // The configuration for each sample sequencer should be as follows:

    // 1. Ensure that the sample sequencer is disabled by clearing the corresponding ASENn bit in the ADCACTSS register.
    // Programming of the sample sequencers is allowed without having them enabled.
    // Disabling the sequencer during programming prevents erroneous execution if a trigger event were to occur during the configuration process.
    ADC0->ACTSS &= ~(1<<3);        // disable SS3 during configuration //

    // 2. Configure the trigger event for the sample sequencer in the ADCEMUX register.
    ADC0->EMUX &= ~0xF000;    // Continous Sampling on the SS3 //

    // 3. When using a PWM generator as the trigger source, use the ADC Trigger Source Select (ADCTSSEL) register to specify in which PWM module the generator
    // is located.
    // The default register reset selects PWM module 0 for all generators.
    // Not needed

    // 4. For each sample in the sample sequence, configure the corresponding input source in the ADCSSMUXn register.
    ADC0->SSMUX3 = 0;         // get input from AIN0 //

    // 5. For each sample in the sample sequence, configure the sample control bits in the corresponding nibble in the ADCSSCTLn register.
    // When programming the last nibble,ensure that the END bit is set. Failure to set the END bit causes unpredictable behavior.
    ADC0->SSCTL3 |= (1<<1)|(1<<2);        // take one sample at a time, set flag at 1st sample //

    // 6. If interrupts are to be used, set the corresponding MASK bit in the ADCIM register.
    // Not needed

    // 7. Enable the sample sequencer logic by setting the corresponding ASENn bit in the ADCACTSS register.
    ADC0->ACTSS |= (1<<3);         // enable ADC0 sequencer 3 //


    while(1)
    {
        ADC0->PSSI |= (1<<3);        // Enable SS3 conversion or start sampling data from AN0 //

        while((ADC0->RIS & 8) == 0) ;   // checking the 3rd bit of the RIS to see if the sampling is completed or not//

        adc_value = ADC0->SSFIFO3; // read adc coversion result from SS3 FIFO //

        ADC0->ISC = 8;          // clear coversion clear flag bit //

        voltage = (adc_value * 0.0008); // convert digital value back into voltage //

        R_LDR = ((3.3/voltage) - 1)*10000; // calculating the calue of resistence of LDR //

        LUX = B*pow(R_LDR,m) ; // Calculating the value of LUX //

    }
}
```

Figure 5-1. Code Screenshot taken from the IAR Embedded workbench IDE software

5.2 Source Code Explanation

Here we will explain our Embedded C code done for TM4C123GH6PM in some steps:

- **In this code first we are going to initialize the ADC by following the given steps:**

  1. We enable the clock for the AD0.
  2. We enable the clock for Port E and set the direction for Port E as Input.
  3. We enable the alternate function for the PE3 pin using the AFSEL.
  4. We disable the digital function for the PE3 pin.
  5. We enable the analogue function for the PE3 pin.

- **Then we are going to configure the sample sequencer by following the given steps:**

  1. We disable the Sample Sequencer 3 (SS3).
  2. We select the continuous sampling mode for the SS3.
  3. We select the AIN0 channel to receive the input.
  4. We select the mode to take 1 sample at a time and set flag at the first sample.
  5. We reenable the Sample Sequencer 3 (SS3).

- **At last we are going to read the input value and perform the arithmetic functions to get the LUX value:**

  1. We start sampling the data.
  2. We initiate condition to check if the sampling for previous sample is completed or not.
  3. We are going to store the value of ADC in the adc_value variable from the FIFO register.
  4. We are going to calculate the voltage by multiplying the adc_value with 0.0008 which is the minimum voltage this ADC can measure.
  5. We are going to calculate the resistance of the LDR in the potentiometer circuit.
  6. We are going to calculate the LUX value by using the R_LDR value.

Chapter 6: Results

From the output of this project, we can finally say that our circuit is working fine for the range between 200 Lux to 2000 Lux as you can verify this from table below that error percentage between these two (Intensity from Lux Meter and IAR Workbench Code for TM4C123GH6PM) is minimum in 200 – 2000 Lux.

Light Intensity from Lux Meter	Light Intensity from IAR Workbench IDE	Error % obtained with respect to Light Intensity from Lux Meter
0	0	-
10	1.68	83.2
20	4.14	79.3
30	7.2	76
40	10.15	74.625
50	15.8	68.4
60	20.88	65.2
70	25.38	63.74286
80	31.06	61.175
90	37.17	58.7
100	43.48	56.52
150	89.09	40.60667
200	121.1	39.45
250	159.8	36.08
300	193.6	35.46667
400	328	18
500	447.09	10.582
600	580.8	3.2
700	701.01	0.14429
800	816.9	2.1125
900	970	7.77778
1000	1258.37	25.837
2000	2650	32.5
3000	6040	101.333

Table 6-1. Experimental Values obtained on LUX Meter and IAR Workbench IDE.

The error percentage we obtained initially is high and it gradually decreases for the values between 200 Lux to 2000 Lux and then it starts to increase again. This is so because when we are at left side of the graph it doesn't show ideal straight line in its natural logarithmic range, same case is with right side of the natural logarithmic graph. The values outside the range show less and less of the ideal straight-line behaviour so the error percentage increases. So, that's why the best range to use our device is between 200 to 2000 Lux.

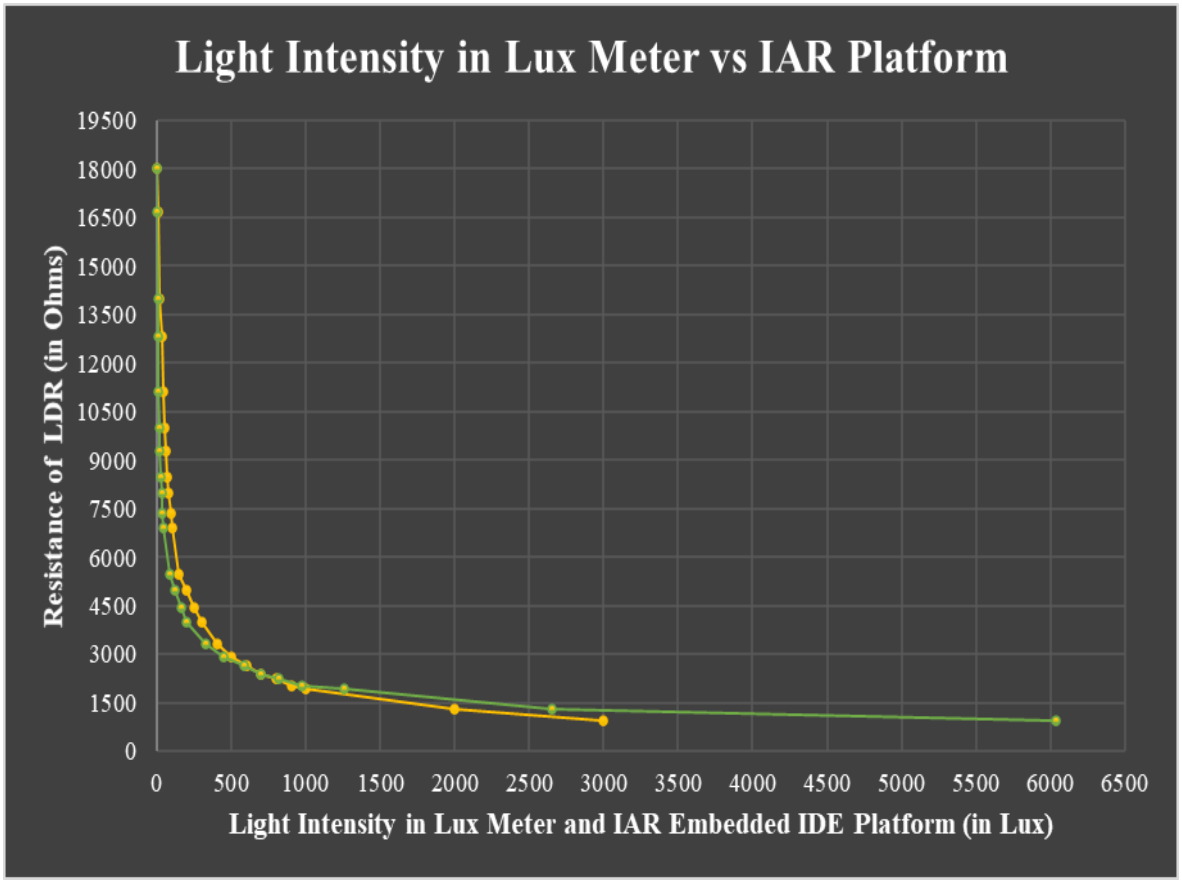

Figure 6-1. This is the graph comparing the theoretical and experimental values of the Resistance of LDR vs Lux graph

Green Line: Light Intensity readings taken from code done for TM4C123GH6PM in IAR Workbench IDE.

Orange Line: Light Intensity readings taken from Lux Meter.

Chapter 7: Conclusion

In this experiment we measure the value of Lux (Lumens per square meter) or the intensity of light, in other words we have made our own Lux Meter, which we have done by using a LDR, and the ADC functionality of the Tiva Board TM4C123GH6PM microcontroller.

By the results of the experiment, we have learnt that our project can be used to measure the range of 200 to 1000 Lux quite accurately but for the values aside from that range the error becomes very high. Therefore, our amateur Lux Meter works fine enough in range 200 to 2000 Lux.

We notice that at very high and low intensities of light our calculations are not precise enough it shows deviation from actual value when done in extreme conditions (High and Low Light), only the middle portion of graph (200 to 2000 Lux) we can say that it has good enough straight-line behaviour, we also notice that instead of using external resistance of 10 kΩ if we use resistance more than this, for a given Resistance of LDR the current in the will decrease due to which V_{ADC} will try to attain value close V_{CC} which is 3.3 Volt because in TM4C123GH6PM maximum ADC value it can attain is 3.3 Volt.

Proof: Since V_{CC} = is 3.3V, therefore $V_{CC} = I (R_{ldr} + R_{ext})$, substituting value of I in V_{ADC}, then we get an equation which looks like this:

$$\frac{V_{CC}}{V_{ADC}} = 1 + \frac{R_{ldr}}{R_{ext}}$$

Since V_{CC} is always a constant = 3.3 Volt and for a given time R_{ldr} is also a constant, therefore by putting appropriate values we will see that on increasing the external resistance (R_{ext}), V_{ADC} tries to attain the values closer to the V_{CC} which is 3.3 Volt.

Some examples where our project can be used are:

- It can be used in smart homes where the light will automatically turn on at sunset.
- It can be used in a locker for security purposes where it will send a signal whenever someone opens the locker or safe.

Bibliography

[1] Programming Tiva C Board using Embedded C - Lecture 11 - YouTube

[2] Programming Tiva C Board using Embedded C - Lecture - 12 - YouTube

[3] Programming Tiva C Board using Embedded C Lecture - 13 - YouTube

[4] Working with ADC in TIVA C - YouTube

[5] Measuring Light Intensity with Arduino | by James Carlson | Medium

[6] LDR 5mm, 12mm, & 20mm Radial Lead Type Photoresistors | NTE Electronics, Inc. (nteinc.com)

[7] Tiva™ C Series TM4C123GH6PM Microcontroller Data Sheet datasheet (Rev. E)

[8] Introduction to LDR - projectiot123 Technology Information Website worldwide

[9] Tiva TM4C123G LaunchPad Pinout, Introduction, Features and datasheet (microcontrollerslab.com)

[10] Tiva C GPIOs | Embedded Lab (embedded-lab.com)

[11] LDR (Light Dependent Resistor) Buy Online India & Hyderabad (potentiallabs.com)

[12] Light Dependent Resistor : Circuit Diagram, Types, Working & Applications (elprocus.com)

[13] LDR(Small) (cytron.io)

[14] Voltage divider - Wikipedia

[15] P7 F) Thermistors & LDRs – Edexcel Physics - Elevise