University of Applied Sciences Bremerhaven

Master of Science
in
Embedded Systems Design

Embedded Systems Project Report

On

Thermoelectric Energy Harvesting

(TEG powered wireless data acquisition for sounding rocket).

Submitted By
Syed Golam Kibria [Matr. No.: 32449]

Supervised By
Prof. Dr.-Ing. Karsten Peter

March 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
API	Application Programming Interface
ATG	Automotive Thermoelectric Generator
CCS	Code Composer Studio
CPU	Central Processing Unit
DTC	Data transfer Controller
DCO	Digitally Controlled Oscillator
DPF	Device Protection Fuse
ED	End Device
EMF	Electromotive Force
GUI	Graphical User Interface
IC	Integrated Circuit
I/O	Input/output
KWh	Kilowatt-hour
LED	Light Emitting Diode
MPPC	Maximum Power Point Control
MWh	Megawatt-hour
OBC	On Board Computer
Op-amp	Operational Amplifier
OSI	Open Systems Interconnection
PJ	Petajoule
PV	Photovoltaic
RAM	Random Access Memory
RFID	Radio Frequency Identification
SMD	Surface-Mount Device
SPI	Serial Peripheral Interface
TEG	Thermoelectric Generator
TEC	Thermoelectric Couple
TE	Thermoelectric Effect
TI	Texas Instruments
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WSN	Wireless Sensor Network

Chapter 1
INTRODUCTION

This chapter describes the following topics respectively:

1.1 Overview on Energy Harvesting

Nowadays technologies have been developing day by day to use the existing free energy sources in environment. That is because of decreasing fossil energy sources and concerning their impacts on environment. There is a huge amount of ambient energy available widely in our surroundings. The ambient energy exists in the form of wind energy, solar energy, thermal energy and mechanical energy. Unfortunately, these ambient energy sources are rarely found in a sufficient amount that it can supply adequate power for any durable purposes. This small amount of 'wasted' energy could be usable as well as useful if it is possible to capture in a proper way. It has a huge environmental impact which also plays as a significant role to solve energy crisis.

Energy harvesting is the process of extracting small amount of energy from naturally occurring energy sources. Also it is the way to improve the efficiency by capturing energy from the energy which is going to be lost as heat, sound, light, movement or vibration etc. Technologies are being developed day by day to make efficient devices for capturing ambient energy and transforming them into electrical energy. The process of energy harvesting is applicable in remote application where it can be more reliable than wall plugs or batteries. On the other hand it can be used as an alternative power source on parallel of a primary power source to enhance the reliability of the overall system.

1.2 Advantages of Energy Harvesting Devices

There is a possibility of replacing batteries for low power, small electronic devices by using the process of energy harvesting. The goal of our project is the same and we will discuss it on the next section. The advantages of using energy harvesting devices rather than traditional battery power are,

- There is no need of replacement of batteries. So it is almost maintenance free.
- The energy harvesting devices are more environmentally friendly compared to the traditional battery power which contains health hazardous chemicals and metals for human body as well as for environment.
- Also it starts a new branch of power application using ambient energy sources.

1.3 Energy Harvesting Project Idea

The idea of the project is to extract the thermal energy from the exhausted gas of the rocket engine with the help of thermoelectric generator (TEG) and with the extracting power we will try to power up the end device of ‘‘EZ430-RF2500 Development Tool’’ which measures the output voltage of TEG. The device also sends the measured data wirelessly to on board computer (OBC). In our case the OBC is a personal computer (PC).

The project idea is a proper application of thermal energy harvesting because there exists a lot of waste heat surrounding the rocket nozzle. This waste heat is tried to be used by the help of TEG. The task of the TEG is to convert the waste heat to electrical energy which means we get a certain amount of voltage across the TEG terminal when there is a temperature difference exists on the two sides of TEG. The placement of TEG in rocket exhaust nozzle is shown in **Figure 1-1** below.

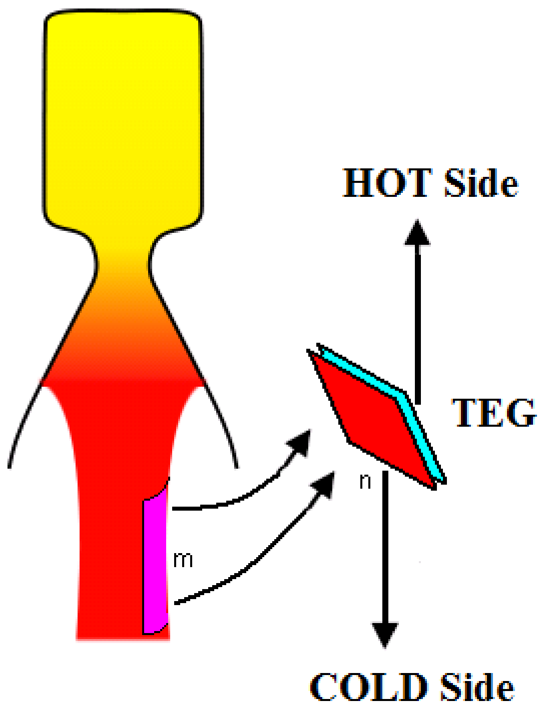


Figure 1-1: Placement of TEG in Rocket Exhaust Nozzle

We can divide the total project idea into several important steps. They are,

- Extracting enough thermal energy from TEG (Although the mounting of TEG in proper position is not our task but it is concerned to get enough thermal energy from TEG)
- Measuring the output voltage of TEG with the help of voltage adaptation circuitry.
- Programming the ‘‘EZ430-RF2500 Development Tool’’ to accept and measure the analog voltage coming from TEG to its analog-to-digital converter (ADC) channel which also sends the measured data wirelessly to OBC.

1.4 Block Diagram of the Energy Harvesting Project

The block diagram is the easiest way to represent the whole project simply. The block diagram of the project is shown in **Figure 1-2** below.

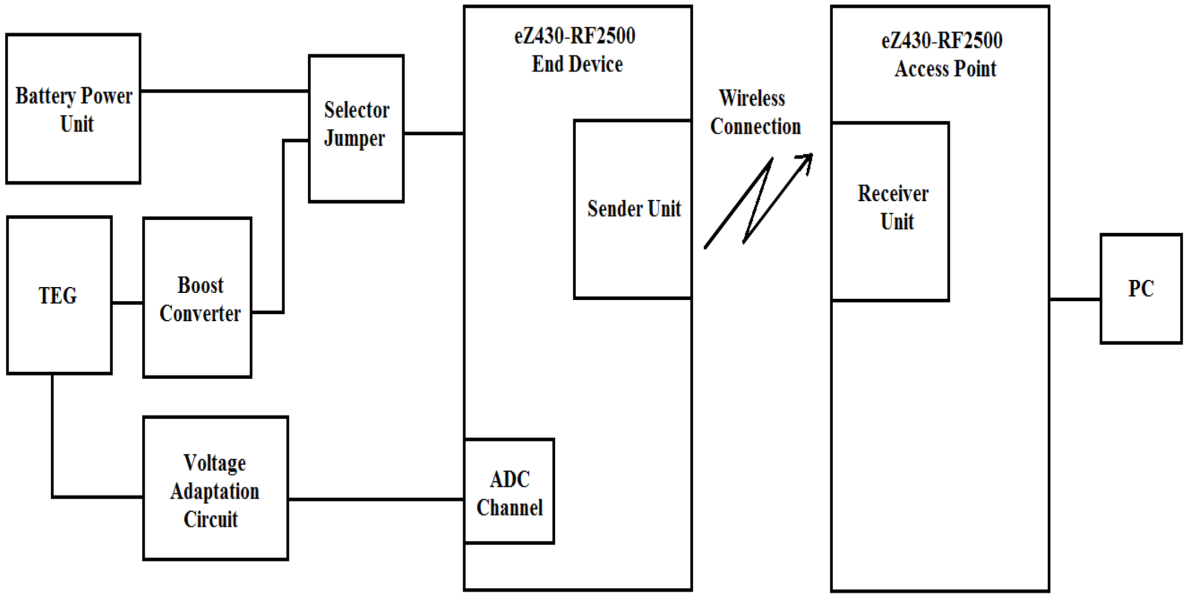


Figure 1-2: Portrayal of Energy Harvesting Project with Block Diagram

From the block diagram we can see that the TEG, end device (ED) and other circuitry are located remotely from the access point. The communication between end device and access point is a wireless communication. The measuring voltage data from TEG is sent to the access point wirelessly. The voltage comes from TEG is boosted by a boost converter, so that there is a second option to power up the end device. We can power the end device with the stand alone battery power or we can use the TEG power with boost converter. We have the opportunity to select any one of the options from the two by changing the jumper position on the selector jumper in our circuit which is described in chapter 3.

The output voltage of the TEG is measured by the end device with the help of ADC channel on its own. As we expected the output voltage range of TEG is not in the range that can be handled by the ADC channel of end device, so we need extra voltage adaptation circuit to fix the proper range for ADC channel.

On the other side the access point is connected via universal serial bus (USB) port to the PC. It receives the voltage measured data from end device and transfers the data to the PC. In PC there is a graphical user interface (GUI) designed in MATLAB. The GUI shows the measured voltage data in real time manner. Also it provides the other information with its other windows like, signal strength, power and TEG sensor temperature.

Chapter 2
LITERATURE REVIEW

This chapter describes following topics respectively:

2.1 Prologue

In this project we were provided with the wireless module named EZ430-RF2500 to transmit and receive the data that we get via TEG device. Then to plot the output behavior in MATLAB by developing a GUI and also design a boost converter and for that we took multiple steps along with the tests at every stage. As a first step we started up learning about the wireless module and the TEG. We worked as a team and completed each step with the cooperation of our team members.

2.2 Thermoelectric Power Generation

Thermoelectric power generation offers a promising technology in the direct conversion of waste heat energy into electrical power. Currently, waste heat powered thermoelectric generators are utilized in a number of useful applications due to their distinct advantages. Automotive engines reject a considerable amount of energy to the nature through the exhaust gas. Significant reduction of engine fuel consumption could be attained by recovering of exhausted heat by using TEGs. Thermoelectric power generators have emerged as an attractive alternative green technology due to their distinct advantages. Thermoelectric power generation offers a potential application in the conversion of waste heat energy into electrical energy where it is unnecessary to consider the cost of the thermal energy input. The application of this alternative green technology in converting waste heat energy directly into electrical power can also improve the overall efficiencies of energy conversion systems. Enormous quantities of waste heat generated from various sources are continuously discharged into the earth’s environment much of it at temperatures which are too low to recover using conventional electrical power generators. Thermoelectric power generation, which presents itself as an enormous alternative green technology, has been successfully used to produce electrical power in a range of scales directly from various sources of waste heat energy.

Natural or elaborately derived from fossil sources the thermal energy escapes into the environment in many places. In the view of the need to reduce the pollutant emissions and to conserve energy resources and the environment, waste heat recycling with TEGs is nowadays the focus of research and technology.

The quantity of actively dissipated, which is, cooled down thermal energy of the chemical industry and oil refineries, is more than 500 PJ only in Germany.

The amount of total heat released by industry to the environment in Germany is estimated at 2600 PJ. (1 petajoule = 10^{15} joules = 277,778 million kWh).

Based on an efficiency of only 0.1%, this corresponds to a volume of re-generatable electrical energy of: 722,222 MWh. By way of comparison, the performance of a modern nuclear power plant is: 1400 MWh

We conducted several experiments on the conceptual model of TEG mounted on the nozzle of a rocket engine, in which waste heat ejected from rocket engine was used as low grade energy and the measured data of the converted electrical performance of TEG was then wirelessly transmitted to a receiver unit connected to the host computer. A wireless sensor on the TEG is supplied with batteries, not by the TEG. The sensor measures the voltage and current, and thus determines the electrical performance of the TEG with the known resistance value(s). Here we used 10 Ω and 100 kΩ resistive loads respectively. The use of a wireless sensor allows the flexible integration of TEG/sensor system without any wiring on the host computer.

2.3 Thermoelectric Devices

Our addiction to electricity has generated a concurrent addiction to fossil fuels. However, the reserves of fossil fuels will soon be exhausted, since oil as well as gas is the limited resources. Over the years, the cost of electricity has risen to unprecedented levels due to the limited supply of oil and economic factors. Thus, renewable energy is a more attractive alternative to electricity generation, as it will also provide a cleaner environment for future generations.

A thermoelectric device converts thermal energy to electrical energy by using an array of thermocouples. This device is a reliable source of power for satellites, space probes, and even unmanned facilities. Since a thermoelectric device has no moving parts, it is reliable and can generate electricity for many years. By achieving a better efficiency, thermoelectric devices would need less radioactive material to produce the same amount of power, making the power generation system lighter. Less radioactive material will also decrease the cost of spaceflight launches. Although these devices are used mostly in spacecraft technologies, they can be also applied to technologies on earth, which might further contribute to the advancement of technology. Some applications of this technology include automobiles, computers, household appliances, etc. For example, thermoelectric devices can enhance the energy production of hybrid automobiles by producing electricity using the waste heat of the engine. If an environment has a thermal gradient, thermoelectric devices can be applied, since they require little maintenance, and provide electricity for many years.

Thermoelectric generators (also called Seebeck generators) are solid state devices that convert heat or temperature differences directly into electrical energy, using the voltage generated at the junction of two different metals. This conversion uses a phenomenon called the Seebeck effect (a form of thermoelectric effect).

Before we discuss the details about TEG used in this project, first we illustrate on thermoelectric effect.

2.4 Thermoelectric Effect

The thermoelectric effect (TE) has been known for nearly two centuries. Thomas Johann Seebeck is credited with discovering that dissimilar metals placed across a temperature gradient could deflect a compass needle. The coefficient of a material or device to generate a voltage per unit of temperature is known as its Seebeck coefficient (stated in ***V/°C***).

The thermoelectric effect is the direct conversion of temperature differences to electric voltage and vice versa. A thermoelectric device creates voltage when there is a different temperature on each side (Seebeck effect). Conversely, when a voltage is applied to it, it creates a temperature difference (Peltier effect).

Thermoelectric effect involves a fundamental interplay between the electronic and thermal properties of a system. These effects are most often observed by measuring electrical quantities (voltage and current) induced by thermal gradients. This effect can be used to generate electricity, measure temperature or change the temperature of objects. Because the direction of the heating and cooling is determined by the polarity of the applied voltage, thermoelectric devices can be used as temperature controllers.

The term thermoelectric effect encompasses three separately identified effects: Seebeck effect, Peltier effect and Thomson effect. The Seebeck effect describes how a temperature difference creates charge flow, while the Peltier effect describes how an electrical current can create a heat flow. The Thomson effect relates the reversible thermal gradient and electromotive force (EMF) in a homogeneous conductor. Electrons transfer heat in two ways, by diffusing heat through collisions with other electrons, or by carrying internal kinetic energy during transport. The former case is standard heat diffusion, while the latter is the Peltier effect. Therefore, the Seebeck effect and the Peltier effect are the opposite of one another.

We will discuss only on Seebeck effect as our TEG was used to convert electrical energy from the waste heat used by the rocket engine.

2.5 Seebeck Effect

The Seebeck effect is a phenomenon in which a temperature difference (hot and cold) between two dissimilar electrical conductors or semiconductors (thermocouple) produces a voltage difference between the two substances or metals. When a conductive material is subjected to a thermal gradient, charge carriers migrate along the gradient from hot to cold and this is the Seebeck effect. In the open-circuit condition, charge carriers will accumulate in the cold region, resulting in the formation of an electric potential difference.

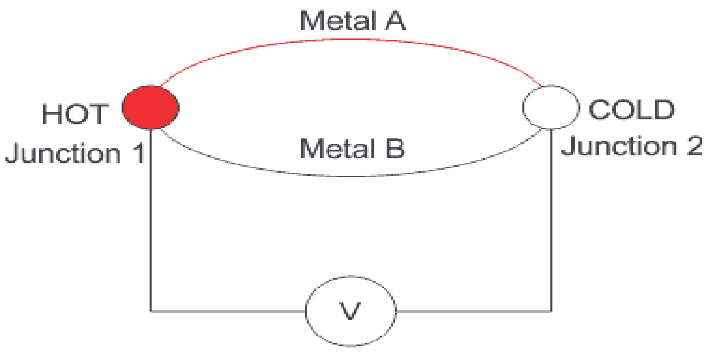


Figure 2-1: Schematic of a Thermocouple [11]

When the two different electrical conductors or semiconductors are kept at different temperatures, the system results in the creation of electrical potential. This was discovered by German physicist Thomas Seebeck.

When heat is applied to one of the two conductors or semiconductors, heated electrons flow toward the cooler one. If the pair is connected through an electrical circuit, direct current (DC) flows through that circuit.

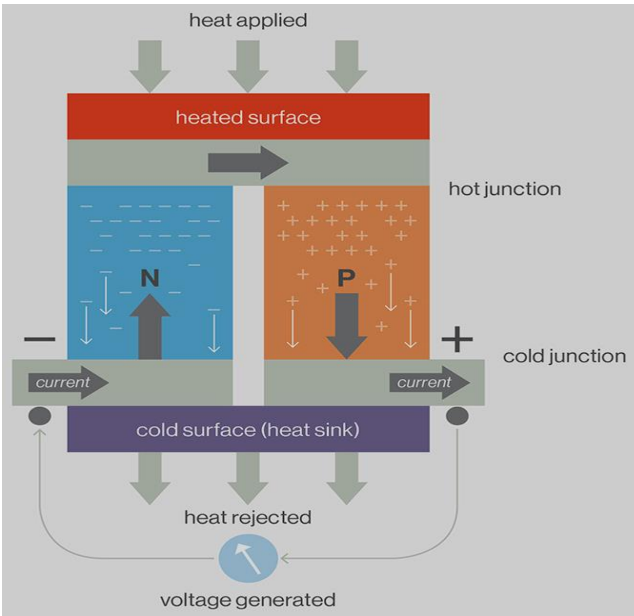


Figure 2-2: Semiconductor Thermocouple Seebeck Effect [10]

The voltages produced by the Seebeck effect are small, usually only a few microvolts per Kelvin of temperature difference at the junction. If the temperature difference is large enough, some Seebeck effect devices can produce a few millivolts. Numerous such devices can be connected in series to increase the output voltage or in parallel to increase the maximum deliverable current. Large arrays of Seebeck effect devices can provide useful, small-scale electrical power if a large temperature difference is maintained across the junctions.

The Seebeck effect is responsible for the behavior of thermocouples, which are used to approximately measure temperature differences or to actuate electronic switches that can turn large systems on and off. This capability is employed in thermoelectric cooling technology. Commonly used thermocouple metal combinations include Copper-Constantan, Iron-Constantan, Chromel-Constantan, Chromel-Alumel etc.

2.5.1 Explanation of Seebeck Effect

The valence electrons in the warmer part of metal are solely responsible for the effect and the reason behind this is thermal energy. Also because of the kinetic energy of these electrons, these valence electrons migrate more rapidly towards the other (colder) end as compared to the colder part electrons migrate towards warmer part. The concept behind their movement is,

- At hot side Fermi distribution is soft i.e. the higher concentration of electrons above the Fermi energy but on cold side the Fermi distribution is sharp i.e. we have fewer electrons above Fermi energy. The Fermi energy is the energy if the highest level of quantum state which is occupied by fermions (like electrons, protons or neutrons) at the absolute zero temperature. It is used to metal, insulators and semi-conductors.
- Electrons go where the energy is lower so therefore it will move from warmer end to the colder end which leads to the transporting energy.

Or in simple words we can come to conclusion that the electrons on a warmer end have a high average momentum as compared to the colder one. Therefore they will take energy with them as compared to the other one.

This movement results in the more negative charge at colder part than warmer part, which leads to the generation of electrical potential. If this pair is connected through an electrical circuit, it results in the generation of a direct current. However the voltage produced is few microvolts (10^{-6}) per Kelvin temperature difference. Now we all are aware of the fact that the voltage increases in series and current increases in parallel. So keeping this fact in mind we can connect many such devices to increase the voltage (in case of series connection) or to increase the maximum current (in parallel). Keeping care of only one thing is that a large temperature difference is required for this purpose. However one important thing is that we have to maintain constant, but different temperature and therefore the energy distribution at the both end will be different and hence it leads to the successful mentioned process.

2.5.2 Mathematical Expression of Seebeck Effect

The Seebeck effect is a classic example of an emf and leads to measurable currents or voltages in the same way as any other emf. Electromotive forces modify Ohm's law by generating currents even in the absence of voltage differences (or vice versa). The local current density is given by,

$$\boldsymbol{J = \sigma(-\nabla V + E_{emf})}$$

where $\boldsymbol{V}$ is the local voltage (the voltage in this case does not refer to electric potential but rather the 'voltmeter' voltage $\boldsymbol{V=\mu/e}$, where $\boldsymbol{\mu}$ is the Fermi level.) and $\boldsymbol{\sigma}$ is the electrical conductivity. In general, the Seebeck effect is described locally by the creation of an electromotive field

$$\boldsymbol{E_{emf} = -S\nabla T}$$

where $\boldsymbol{S}$ is the Seebeck coefficient (also known as thermo power), a property of the local material, and $\boldsymbol{\nabla T}$ is the gradient in temperature $\boldsymbol{T}$.

If the system reaches a steady state where $\boldsymbol{J=0}$, then the voltage gradient is given simply by the emf: $-\nabla V = \nabla T$. This simple relationship, which does not depend on conductivity, is used in the thermocouple to measure a temperature difference. An absolute temperature may be found by performing the voltage measurement at a known reference temperature.

2.5.3 Applications of Seebeck Effect

- This Seebeck effect is commonly used in thermocouples to measure the temperature differences or to actuate the electronic switches that can turn the system on or off.
- The Seebeck effect is used in thermoelectric generators, which function like heat engines, but are less bulky and having no moving parts.
- They are also used in some power plants in order to convert waste heat into additional power.
- In automobiles as automotive thermoelectric generators (ATGs) for increasing the fuel efficiency.

2.6 Thermoelectric Generator

A thermoelectric generator is a device utilizing one or more thermoelectric modules as the primary component(s), followed by a cooling system that can be either passive or active. Such as an open air heat sink, fan cooled heat sink etc. These components are then fabricated into an assembly to function as one unit called a TEG.

When heat is applied to the hot side of a TEG, electricity is produced. Almost any heat source can be used to generate electricity such as waste heat of an engine, solar heat, geothermal heat etc. In addition the efficiency of any device or machine that generates heat as a by-product can be drastically improved by recovering the energy lost as heat.

Thermoelectric generators could be used in power plants or spacecraft in order to convert waste heat into additional electrical power and in automobiles as automotive thermoelectric generators to increase fuel efficiency. A usual TEG is shown in **Figure 2-3** below.

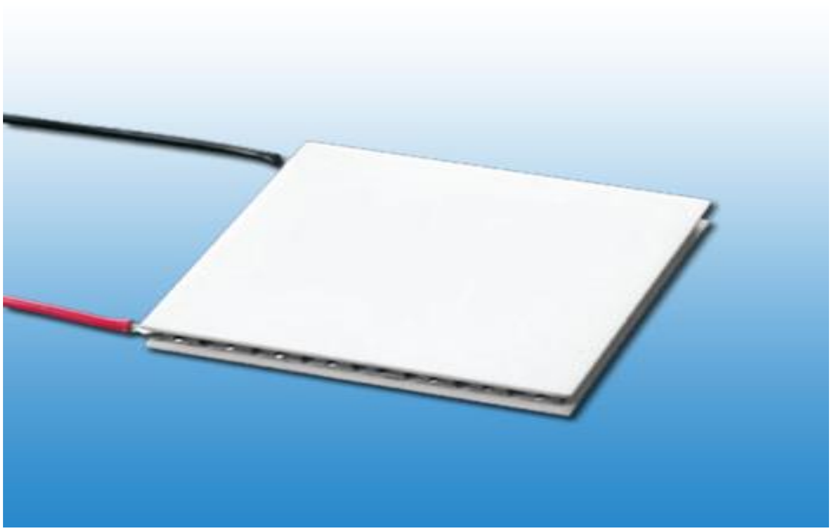


Figure 2-3: Image of a Thermoelectric Generator [14]

2.6.1 Construction of a TEG

Thermoelectric Materials:

Thermoelectric materials generate power directly from heat by converting temperature differences into electric voltage. These materials must have both high electrical conductivity and low thermal conductivity to be good thermoelectric materials. Having low thermal conductivity ensures that when one side is made hot, the other side stays cold, which helps to generate a large voltage while in a temperature gradient.

The main three semiconductors are known to have low thermal conductivity as well as high power factor are bismuth telluride (**Bi_2Te_3**), lead telluride (**PbTe**) and silicon germanium (**SiGe**).

Thermoelectric Module:

Construction of a TEG power module consists of pairs of p-type and n-type semiconductor materials with a high thermoelectric coefficient. Although many alloys can be used, bismuth telluride is the most commonly used material today. This material is sliced into small blocks, one forms the p-type conductor and the other one the n-type conductor. Each pair forms a thermoelectric couple (TEC). These thermocouples are most often connected electrically forming an array of multiple thermocouples (thermopile).

Most thermoelectric module manufacturing companies use many thermoelectric couples that are sandwiched between two pieces of non-electrically conductive materials. It is also necessary for this material to be thermally conductive to ensure a good heat transfer; usually two thin ceramic wafers are used, to form what is called a thermoelectric module.

Each module can contain dozens of pairs of thermoelectric couples called thermoelectric modules, TEC modules and sometimes Peltier or, Seebeck modules, which simply denotes whether they are being used to generate electricity (Seebeck) or produce heat or cold (Peltier).

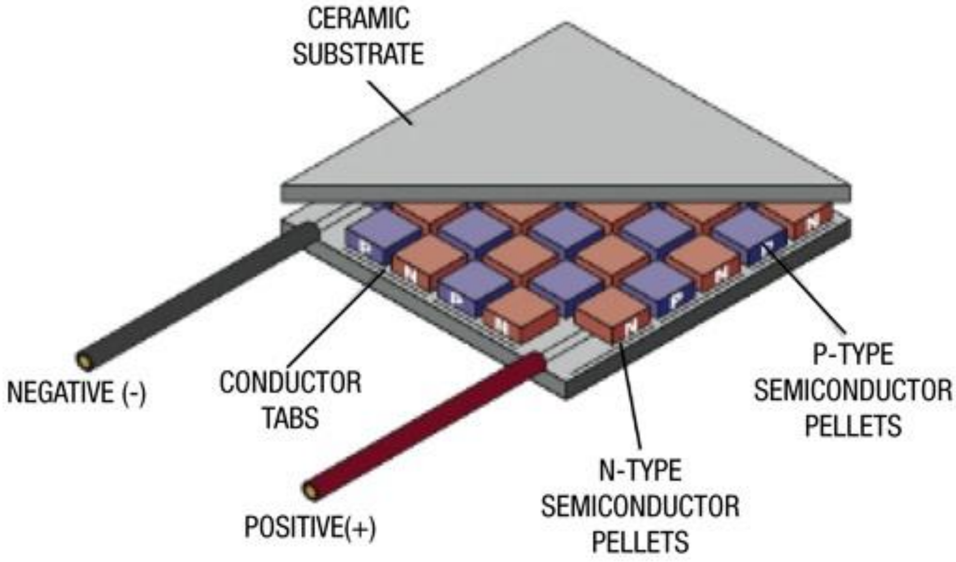


Figure 2-4: TEG Module Construction [13]

A sandwich implementing an array of n-type and p-type semiconductor legs (see **Figure 2-4**) creates the typical thermoelectric module. The legs are pellets of either bismuth telluride or antimony telluride. The legs are strapped together to create a series electrical connection and a parallel thermal connection. The top and bottom of the module is most often aluminum oxide ceramic to provide electrical insulation and good thermal conductivity.

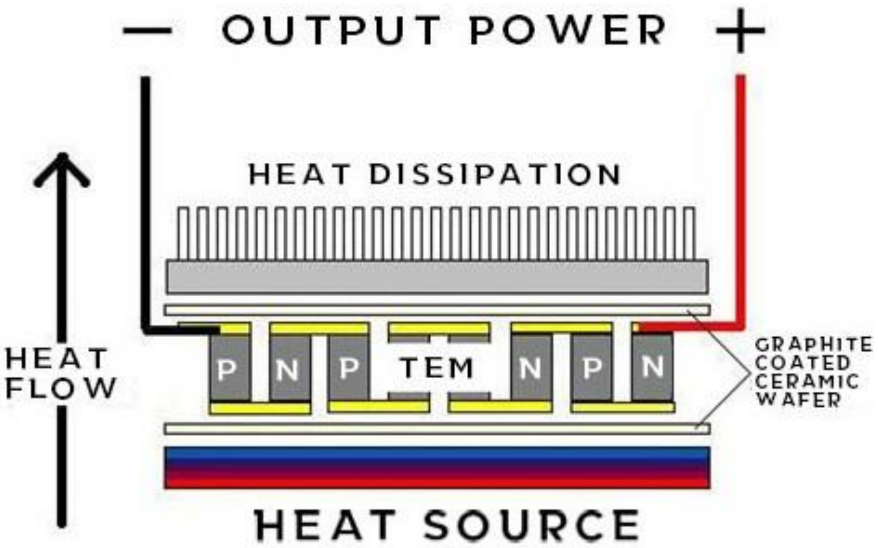


Figure 2-5a

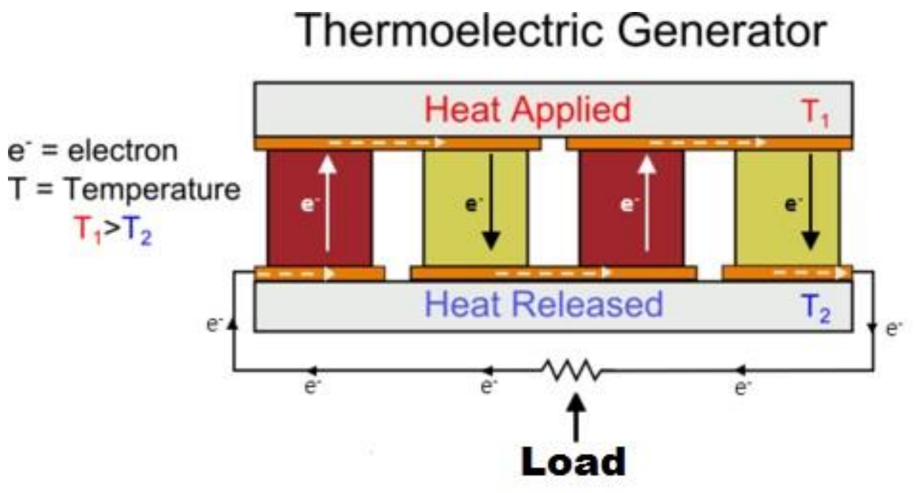


Figure 2-5b

Figure 2-5 (2-5a & 2-5b): Power Generation Configuration of a TEG [8] [9]

In **Figure 2-5b**, the current flow has been labeled in the direction of the electrons flow.

When the heat energy (T_1) is applied on the hot side of the TEG, then the cold side gets the heat released (T_2). If there is a temperature difference between the two conductive materials (T_1>T_2), the electrons start flowing and the electrical energy is obtained across the load as shown in **Figure 2-5b** above.

2.6.2 Efficiency of TEG

Thermoelectric generators have attractive features but low efficiency. The typical efficiency of a TEG is around 5–8%. Using pulsed power rather than continuous power can improve the efficiency.

2.6.3 Benefits and Practical Limitations of TEG

Benefits:

- TEG converts the energy without any moving parts, which makes it so reliable during the operation.
- It is stable and requires no maintenance due to the absence of movable parts.
- No noise during the operation and free of vibration.
- Reduced size and light in weight.
- Environmental friendly and recycles wasted heat energy.
- Lower production cost.
- Scalability, meaning that the device can be applied to any size heat source from a water heater to a manufacturer's equipment.

Practical Limitations:

- Low energy conversion efficiency.
- Slow technology progression.
- Requires relatively constant heat source.
- Relatively high electrical output resistance, which increases the self-heating.
- Relatively low thermal conductivity, which makes the TEG unsuitable for applications where heat removal is critical.

2.6.4 Practical Applications of TEG

- Thermoelectric generators are primarily used as remote and off-grid power generators for unmanned sites. They are the most reliable power generator in such situations as they do not have moving parts (thus virtually maintenance free), work day and night, perform under all weather conditions and can work without battery backup. Although solar photovoltaic (PV) systems are also implemented in remote sites, but PV systems may not be a suitable solution where solar radiation is low, i.e. areas at higher latitudes with snow or no sunshine, areas with lots of cloud, dusty deserts, forests etc.
- Cars and other automobiles produce waste heat (in the exhaust and in the cooling agents). Harvesting that heat energy, using a thermoelectric generator, can increase the fuel efficiency of the car.
- Spacecraft such as rockets need continuous power supply when they are settled in the space. The waste heat formed by the engine can be converted into electrical power using TEGs.

2.6.5 Details of TEG Used During the Experiment

The TEG that we used during the project set up in Trauen as well as in laboratory was **TEG 127-200-27** from Thermalforce, Berlin, Germany. The details are discussed below.

The TEG nomenclature is given below:
127: Number of Thermocouples
200: Maximum Temperature
27: Item Number

According to the data sheet provided by Thermalforce, the ‘‘TEG 127-200-27’’ is of 30x30 millimeter (mm) in length and width respectively and height or thickness is 4.2 mm. The image of the TEG is shown in **Figure 2-6** with the dimensions.

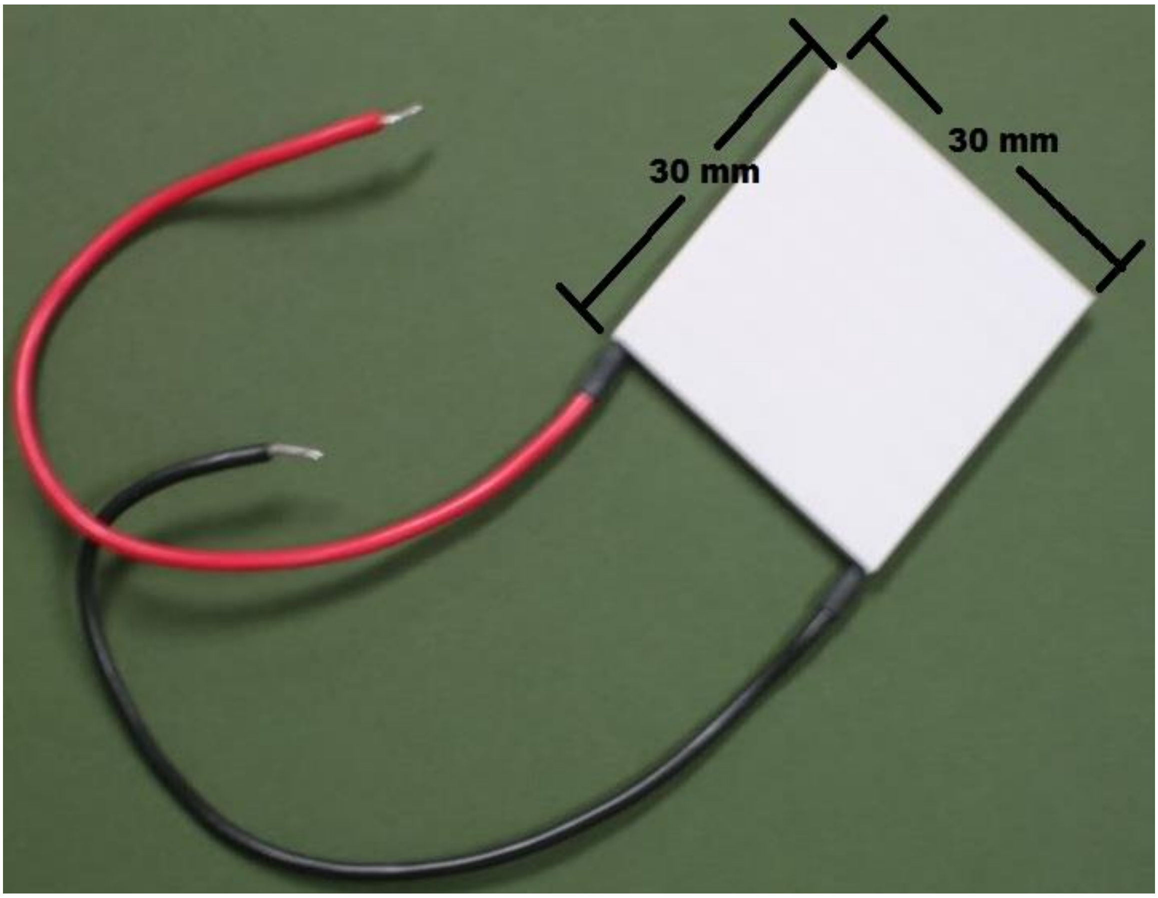


Figure 2-6: ''TEG 127-200-27'' with the Dimensions Mentioned

Mechanical Data of TEG 127-200-27:
Number of thermocouples: 127
Material: Bi_2Te_3
Weight: 15 g
Maximum temperatures: 200 °C

After investigation, taken into consideration are the following pertinent TEG parameters. These are listed in **Table 2-1** below.

Table 2-1: The ''TEG 127-200-27'' Electrical Parameters

Dimensions L×B×H	Specifications of Temp. at Different Temp. Difference (ΔT)	Open Circuit Voltage (U)	Short Circuit Current (I)	Electric Resistance (R)	Thermal-Force (α)	Thermal-Conductivity (κ)	Electric Power (P)	Efficiency (η)	Max. Temp. (T)
mm		**V**	**A**	**Ω**	**(V/K)**	**(W/K)**	**W**	**%**	**°C**
30x30x4.2	At ΔT=100K, T_{HOT}=150°C and T_{COLD}=50°C	6.05	0.75	5.5	0.056	0.280	1.13	3.113	200
	At ΔT=120K, T_{HOT}=150°C and T_{COLD}=30°C	7.19	0.93	5.5	0.056	0.280	1.66	3.803	

The following parameters for the sender unit were taken into account for safety purposes.

2.6.6 Power Module Installation of TEG

The power module has large thermal expansion. When the module is running and cycling over large temperature range, the proper mounting to ensure even pressure applied on module is very important.

Hot Side and Cold Side Identification:

The modules will generate electricity only if there is a temperature difference across the modules. The hot side must be in excellent contact with a heat source and cold side must be thermally touching and compressed against a heat exchanger that can remove the heat effectively away from the cold side face of the module in order to create a large temperature difference. We should attach the cold side to heat sink or heat exchanger and hot side to the heat source. Then the positive output is in red wire, if reversed, then in black colored wire.

The temperature on the hot side of the module can work at as high as 200 °C (TEG 127-200-27) or 260 °C (TEG 241-260-35) and intermittently up to 380 °C. But the temperature at cold side of the module cannot work properly above 100 °C. So, if the mounting is reversed, and the cold side of module is attached to heat source above 100 °C, the module will degrade quickly or fail immediately. So, to ensure the hot side attached to heat source is very important.

The **Figure 2-7** is the diagram to show which side of module is hot or cold one, where the wire direction faces towards us. (Usually, we also can see a marking ‘‘Hot Side” on hot side ceramic plate) The positive output will be on red colored wire if the installation is correct. The reversed installation will lead to negative output from red colored wire.

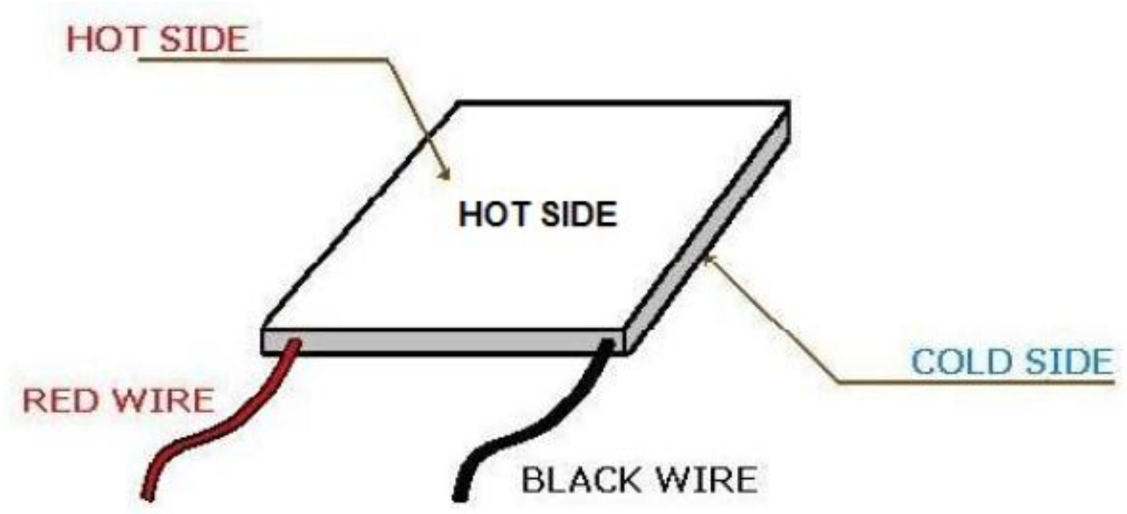


Figure 2-7: Hot and Cold Sides Identification of a TEG [12]

TEG should be mounted using the compression method. That is, the TEG is compressed between a hot plate and a heat sink that will be cooler. The compression or clamping should be created with stainless steel machine screws on either side of the TEG.

2.7 EZ430-RF2500 Module

The EZ430-RF2500 is a wireless module by texas instruments (TI) and used to transmit and receive data that we obtain through our TEG. The device is shown in **Figure 2-8** below. Here we discuss the description of the hardware part of the module and the testing of the device with sensor monitor which tells us the temperature of the surroundings by using SimpliciTI protocol which is designed for low power small networks.

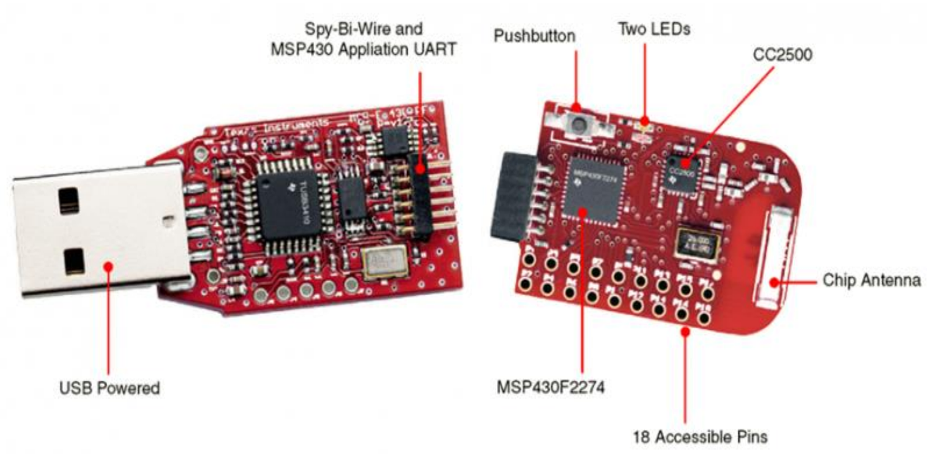


Figure 2-8: EZ430-RF2500 Development Kit [27]

The task is divided into two parts:

- **Components Embedded in the Device**
- **Testing the Device**

2.7.1 Components Embedded in the Device

MSP430F2274 16-bit Ultra-Low-Power Microcontroller:

This microcontroller is part of MSP430 family made by TI as shown in **Figure 2-9**. Its vital features are ultra-low power consumption, five low power modes, optimized to get more battery life while being able to be portable, powerful central processing unit (CPU) made of 16 bit risk architecture, has 16 bit registers, constant generators to get maximum code efficiency, very quick as the wake up time from stand by to active mode is almost 1 µs because of the digitally controlled oscillator (DCO). The F2274M series of this microcontroller has 2x built in 16-bit timers, 10 bit ADC, watchdog timer, data transfer controller (DTC) and universal serial communication interface, 32 I/O Pins and 2x Op-amps in the F2274M series devices as shown in the **Figure 2-9** below. [29]

A common application of this microcontroller is as a sensor which gets an analog input, converts it to a digital form and then processes it to display on the screen or transmits it wirelessly to receiver placed at host position. [29]

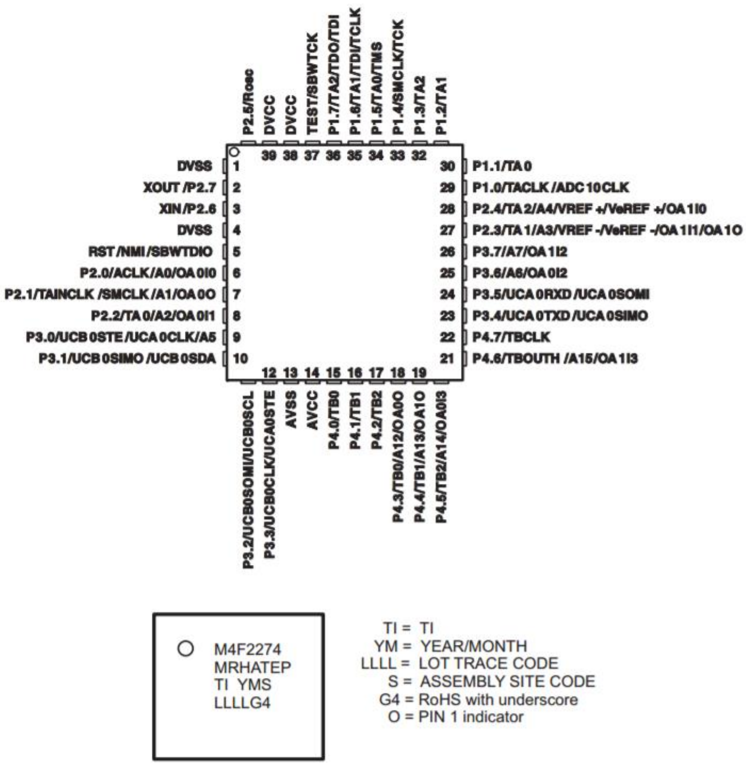


Figure 2-9: MSP430F2274 Device Pinouts (RHA Package) [28] [29]

The operating temperature range of this microcontroller is -40°C to +85°C and the operating supply voltage is 1.8V - 3.6V.The operating frequency is maximum 16 MHz at VCC >= 3.3V.

LEDs & Push Button:

There are two LEDs colored green and red connected with the general purpose I/O pins in the device for the purpose of visual feedback and there is also a push button used as an interrupt for the sake of user feedback.

Chip Antenna:

This is an surface-mount device (SMD) multilayer chip antenna and other prominent features are that it is a light weight and extremely low profile component having a capacity of 5W max, high gain, omni-directional, operating temperature of -40°C to +85°C, works with a frequency range of 2400-2500 MHz and an impedance of 50 Ohms. [31] The programmable data rate of this antenna is 1.2 to 500 Kbps and the current consumption is also very low. The antenna placed on the device can be seen in **Figure 2-8** above.

Development Pins:

There are 18 development pins available on the device and there are also 3 additional pins available when we attach the battery board. There are also 3 more pins but two of them are used as general purpose and the third one is common ground and all of these 18 pins from the target board and 6 pins from the battery board are placed at 0.1 inches to each other as shown in **Figure 2-10**.

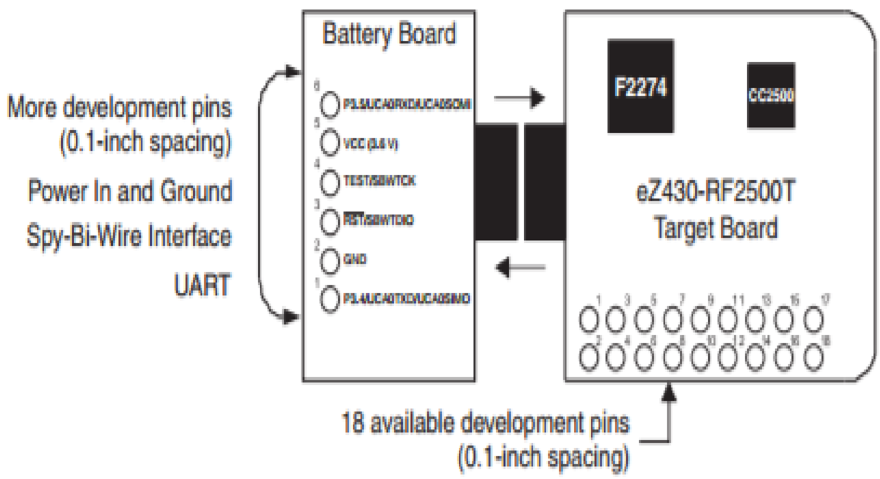


Figure 2-10: EZ430-RF2500 Development Tool [30]

2.7.2 Spy-Bi-Wire & MSP430 Application UART

RST/ SBWTDIO & TEST/ SBWTCK:

The RST pin works as a reset or a non-maskable interrupt input and also as a Spy-Bi-Wire test data input/output while programming and testing. The TEST pin selects the test mode for

JTAG pins on port1 and the device protection fuse (DPF) is connected to TEST and works as Spy-Bi-Wire test clock input during programming and test. [30]

P3.4 & P3.5 Pins:

The P3.4 serves as a general purpose digital I/O pin, transmits data output in universal asynchronous receiver transmitter (UART) mode (UART communication 2274 to PC) and slave in/master out when working in serial peripheral interface (SPI) mode. P3.5 also serves as general purpose digital I/O pin but receives data input in UART mode (UART communication from 2274 to PC) and works as slave out/master in when used in SPI mode. [30]

2.7.3 Testing the Device

In this section we will discuss about how we started after getting the device delivered by Airbus Defense and Space, Bremen and which software we used for this purpose.

In the start we looked for the required tools and the basic codes to understand the working of the device. At first we got a few options like IAR Embedded Workbench developed by third party but very powerful tool, code composer studio (CCS) developed by TI, sensor monitor and a couple of few basic codes like blinking LED and the built in default code that measures the temperature of the environment at default.

2.7.4 Software

Sensor Monitor:

In this part we are going to discuss some of the working capabilities of this tool that we used in our project. The sensor monitor is a tool in which we can see the temperature of the surroundings of the device when we have plugged it into the computer. We tried it both with one device and two devices plugged at the same time; and it works perfectly in both these cases as we increase the number of nodes. We can see an additional node in our window, as it develops a wireless sensor network (WSN) in the form of star topology. The following **Figure 2-11** below shows us easily that how we can see the nodes in the form of star topology attached and their surrounding temperatures when plugged in.

Figure 2-11: Sensor Monitor with Single Node [32]

The above **Figure 2-11** shows the output of a single node i.e. 85.4°F and the power provided to the device i.e. 3.1V. In the **Figure 2-12** below we can see the output of the sensor monitor when multiple devices (in this case 5) have been attached and the formation is the same as explained earlier i.e. star topology.

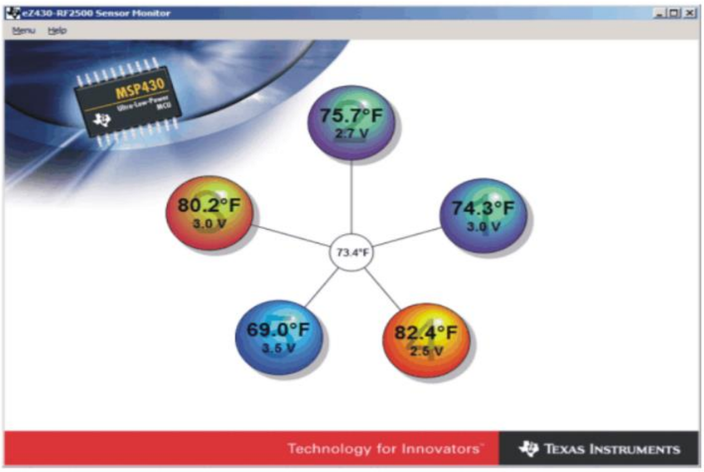


Figure 2-12: Sensor Monitor with Multiple Nodes [33]

Network Protocol:

The network protocol used by the device is named as SimpliciTI. The reason behind using this is that it consumes low power, used for simple small networks (<100 nodes), can be implemented with minor microcontroller resource requirements. As this is a small network and requires low power hence we can provide the required power using a battery with a long life. These networks typically have low data rate and low duty cycles. The number of nodes talking to each other is also limited in number and hence is a low cost package with almost ten times lesser random access memory (RAM) and program memory. The topology used by SimpliciTI is peer-to-peer, an option to use an access point for storing and forwarding messages, it requires modest resources and range extenders for the purpose of extending the range to 4 hops.

SimpliciTI also supports an extended range of applications from smoke detector and home automation to active radio frequency identification (RFID). This protocol is open to all and has no royalties attached to it.

Overview of Architecture:

SimpliciTI supports 3 layers named:

- **Application Layer**
- **Network Layer**
- **Data Link Layer**

The first layer is developed by the customer. Simple set of application programming interface (API) symbols are used for the initialization and configuration of the network along with the reading and writing messages. This architecture is actually not following the open systems interconnection (OSI) reference model strictly. The network layer in this architecture manages the sending and receiving queues i.e. Tx and Rx respectively.

SimpliciTI supports two topologies basically:

- **Peer-to-Peer connection.**
- **Star Topology where the main hub i.e. the star hub acts as a peer to all other devices.**

The APIs offered by SimpliciTI:

They enable the user in implementing a reliable network with small effort. Although we should remember that the flexibility and simplicity of the network gets sacrificed.

1. **Initialization**

   I. **SMPL Init (uint8(*pCB)(linkID))**

   It offers the initialization of the network to happen. A call of this event is that it generates an attempt for joining a network silently, and also sets up the radio as a result. If it is an API then it might also cause a generation of an encryption key or a link token.

2. **Linking**

   I. **SMPL Link (linkID t *linkID)**

   There is a listening device which is a server and will be linked to another device which will act as a client. The acknowledgment is done by the server. After receiving the acknowledgement this call will be retuned with an ID which will be used for further communication.

   II. **SMPL LinkListen(linkID t *linkID)**

   This is a partner to the Link and waits until the link message is sent from a client. It works on the first come first serve basis i.e. it will accept the first valid link message and only one call can be enforced at a moment.

3. Peer to Peer Messaging

I. SMPL Send (linkID t lid, uint8 *msg, uint8 len)

The Send function sends messages in the form of a buffer attached with a link ID to the access point, including the information about the message and its length.

II. SMPL Receive(linkID t lid, uint8 *msg, uint8 *len)

The message sent from the Send function is received by this function and returns the oldest message's length and the message itself. This function only deals with checking either the required data has reached or not and does not deal with the switching on and off of the radio.

4. Configuration

SMPL Ioctl (ioctlObject t object, ioctlAction t action, void *val)

This is done in runtime by the application for configuring the network and can be used for changing the states of the radio to wake up or sleep and for changing the channel and etc.

Chapter 3
SIMULATION and HARDWARE DESIGN

This chapter describes following topics respectively:

3.1 Overview
3.2 Elaborated Diagram of the Project with Simulated Circuit
3.3 Overview of LTSPICE
3.4 Simulation in LTSPICE
3.5 DC/DC Boost Converter Design
3.6 MATLAB GUI Design
3.7 Epilogue

3.1 Overview

The expected voltage range for different operating modes is shown in **Table 3-1** below.

Table 3-1: Expected Voltage Range for the Used Loads in Different Mode of Operation

Mode of Operation	Load Resistance	Expected Voltage Range
Power Adjustment	10 Ω	500 mV – 3 V
Idle Condition	100 kΩ	1 V – 6 V
No Load	-----	300 mV- 1 V

As we can see from the above table in idle condition, the voltage coming from the TEG is up to 6V which exceeds the maximum voltage that the sender unit (ED) can handle. For that reason we need a voltage divider that can divide the incoming voltage in equal two parts. The designed voltage divider is simulated in LTSPICE IV and it has been discussed elaborately in another section.

3.2 Elaborated Diagram of the Project with Simulated Circuit

The functional block diagram of the whole project setup with simulation is shown in **Figure 3-1**. As we can see from the figure, we used a boost converter to boost up the voltage that comes from TEG to make our end device wake up. Also we have stand-alone battery system which contains a voltage regulator to down the regulation of battery voltage from 6V to 3.3V. On the other hand we have the opportunity to use a big battery which has more voltage and power rating than previous small 6V batteries and also it does not has the thermal rundown effect. By the help of connector C2 as shown in **Figure 3-1**, we can use that big battery but we have to keep in mind that when we are using another battery through connector C2, there should not be any batteries of 6V in battery case.

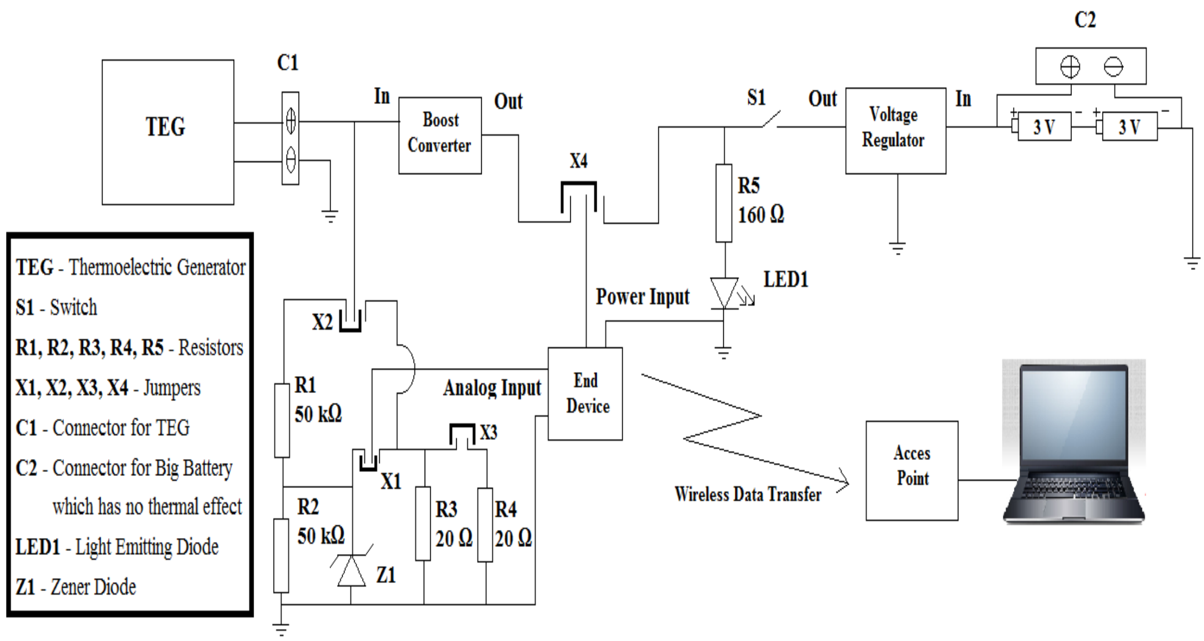


Figure 3-1: Functional Block Diagram of the Project with Simulated Circuit

We can choose either battery power or TEG with boost converter for powering the end device by changing the jumper X4. We need to keep 'on' the switch S1 while using stand-alone battery system. By glowing LED1 acknowledge that the battery system is ready for service. It is needed to mention that our boost converter has not only ability to boost up the voltage but also has the feature of down regulation. That means when voltage comes from TEG is increased more than 3.3V; the boost convert keeps its output stable to 3.3V. The boost converter has both boosted and down regulation.

Other important part of the block diagram is voltage divider. The output taken from the voltage divider is feed into the analog input of the end device. That measured voltage data is sent wirelessly to access point which is connected to a laptop and is located remotely from the end device. Then using a real time GUI in MATLAB that measures voltage data is shown graphically and is saved in an Excel file.

3.3 Overview of LTSPICE

Before going to use and simulate our circuit in LTSPICE, we would like to give a brief description of it. We used LTSPICE version IV during our project task. We chose this simulator because it is simple and easy to understand. We will not explain here each and everything but some basic things that we need to get started with LTSPICE IV. Also the terms and operations that we used in our simulation will be discussed.

Key Features of LTSPICE IV:

- It can be downloaded free in full version.
- The schematic size is unlimited.
- Provides symbol editor.
- Good combination of accuracy, robustness, speed and compatibility.

- Provide advance analysis/simulation options, FFTs, parameter sweeps, etc.
- Possible to run third party models.

3.3.1 LTSPICE IV User Interface

Although it is possible to simulate a circuit using netlist file in LTSPICE IV but we think it is easy to understand by using schematic editor for simulation. Whenever we start LTSPICE IV software we first see the opening screen which is shown in **Figure 3-2** below.

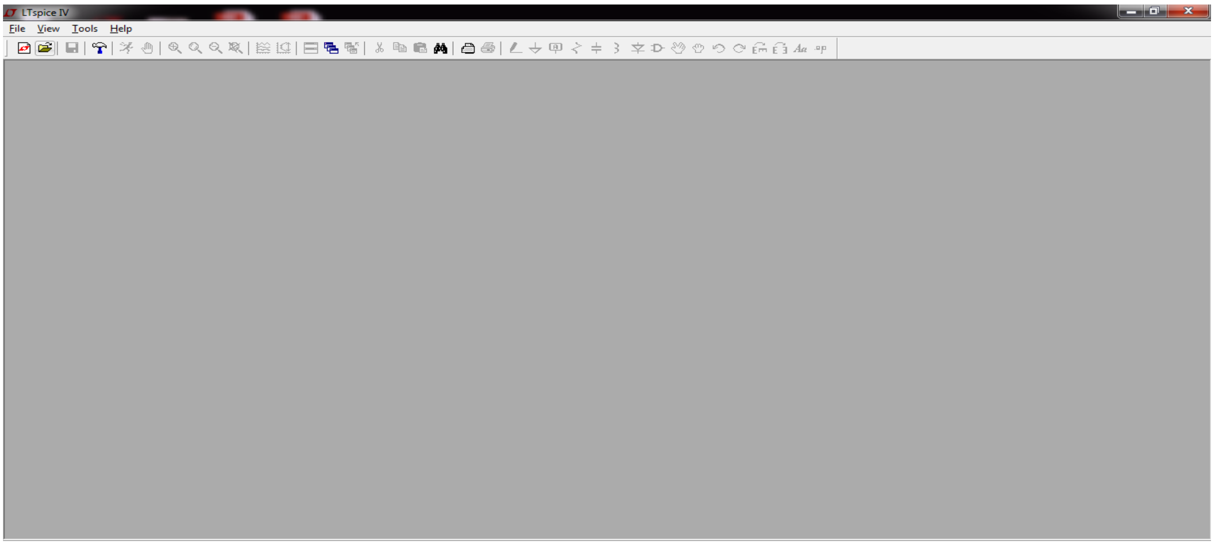


Figure 3-2: Opening Screen of LTSPICE IV

There are lots of icons we can find in the toolbar. Those are marked by their names and shown in **Figure 3-3** below.

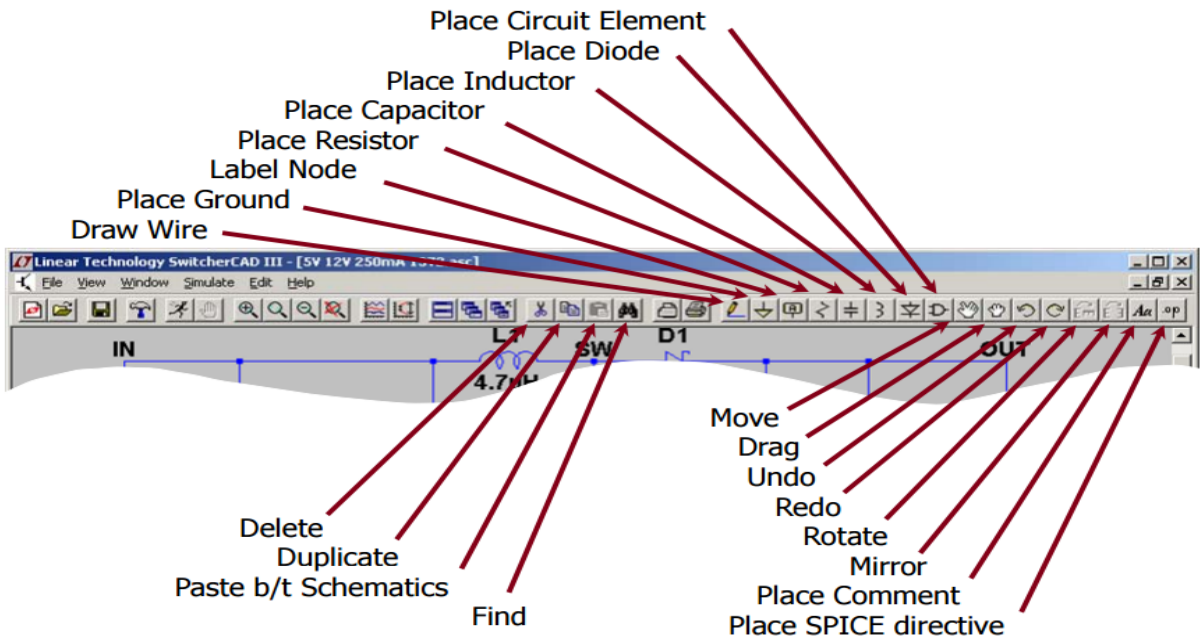


Figure 3-3: Summary of Schematic Editor Toolbar in LTSPICE IV [18]

3.3.2 Getting Started Using LTSPICE IV

To simulate a circuit in LTSPICE IV, we have to follow few steps which are shown in articles below with relevant figures.

Opening LTSPICE IV:

First we have to start LTSPICE IV so that we can start a new schematic or open an existing one. For our case we do not have any existing circuit before, we are going to draw a new one. To start a new schematic we should press the new schematic button in the upper left of the window or we can go to **File** > **New Schematic**. The process is shown in **Figure 3-4**.

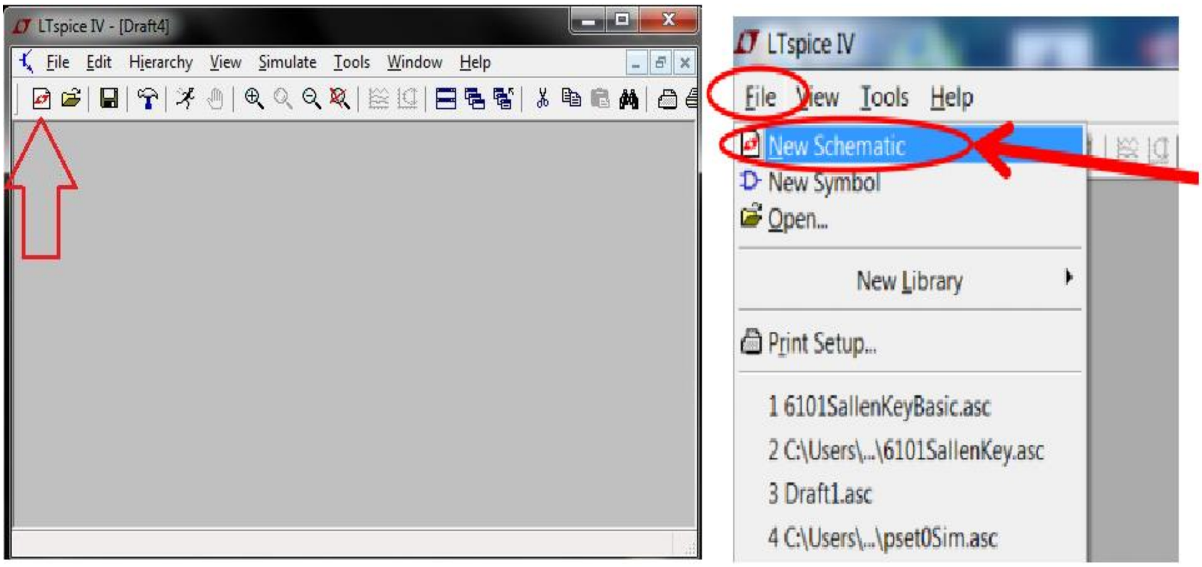


Figure 3-4: Opening a New Schematic in LTSPICE IV

By starting a new schematic we create a blank area where we can start building a schematic for design and analysis.

Building a Circuit:

We will find circuit building toolbar along the top of the schematic editor as shown in **Figure 3-5** below. From there we can take some basic circuit elements like resistor (R), inductor (L), capacitor (C), diode (D) and ground (G) directly from the toolbar. Also we can add more circuit elements by clicking the component button  or going to **Edit** > **Component**, where we can find vast catalog of available parts and systems which are included by default with LTSPICE IV.

Figure 3-5: Circuit Building Toolbar in LTSPICE IV

We can search any parts by writing their names on search bar or by clicking the [**categories**] to the left. We can see more details from **Figure 3-6**.

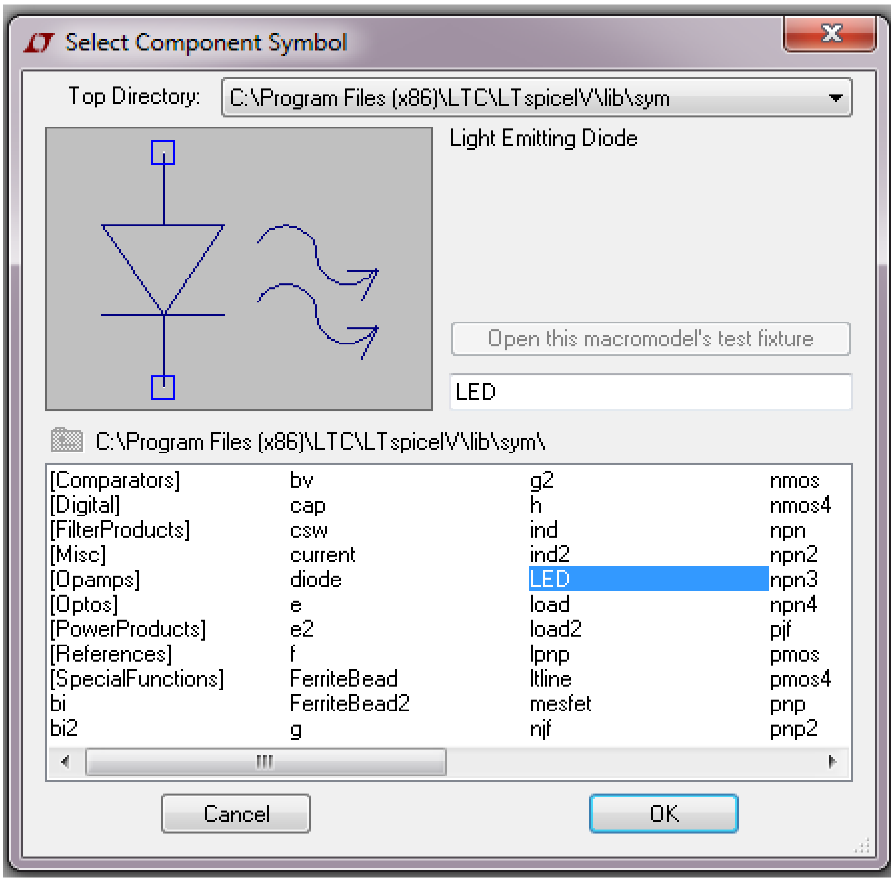


Figure 3-6: Components Selection from Component Catalog in LTSPICE IV

Whenever our components selection process is finished, we can connect each component as we like by clicking on wire  button or going to **Edit** > **Draw Wire**. We can also enter the desire value of components. For that we have to right click on the components and enter the value. For example in **Figure 3-7**, it is shown that we can edit the default value of voltage source by right click on it.

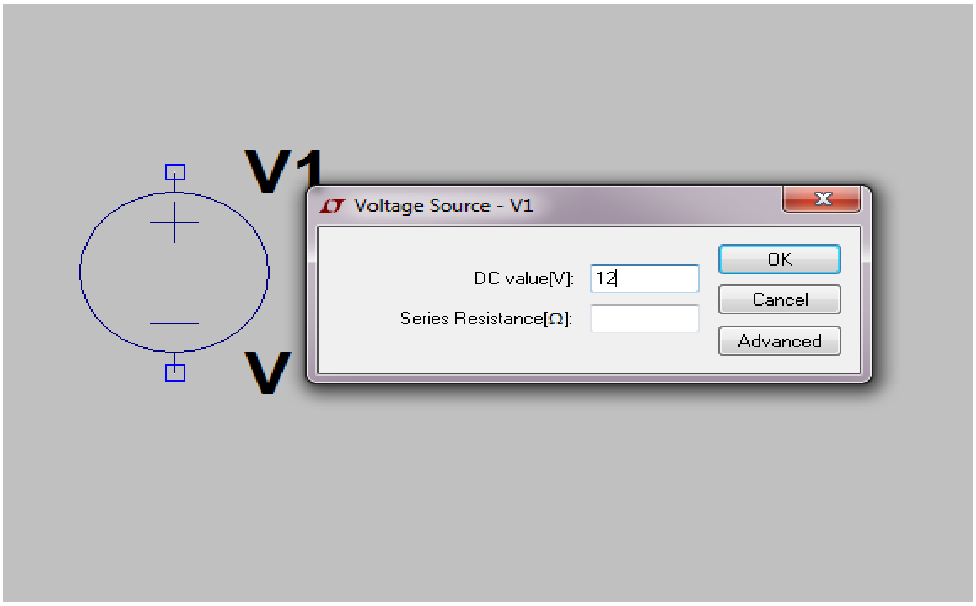


Figure 3-7: Editing Voltage Source to Enter the Desired Value in LTSPICE IV

Simulating the Circuit:

After finishing drawing a circuit and modifying the element with proper desire values we are now ready to simulate that circuit. There are some important things that we have to consider before starting the simulation. Those are,

- The circuit should be properly drawn and saved.
- The ground connection of the circuit must be connected.
- There must not be any unconnected elements on the design that are going to be simulated.
- There should not be any extra wire(s).

If all criteria provided above are fulfilled, then we can proceed for the simulation. Clicking on the  button or going to **Simulate** > **Run**, we can start simulation. As we did not select any specfic type of simulation, so after clicking on "Run" button where a simulation comand window will popup and we have to select type of simulation with appropriat values. The simulation comand window is shown in **Figure 3-8**.

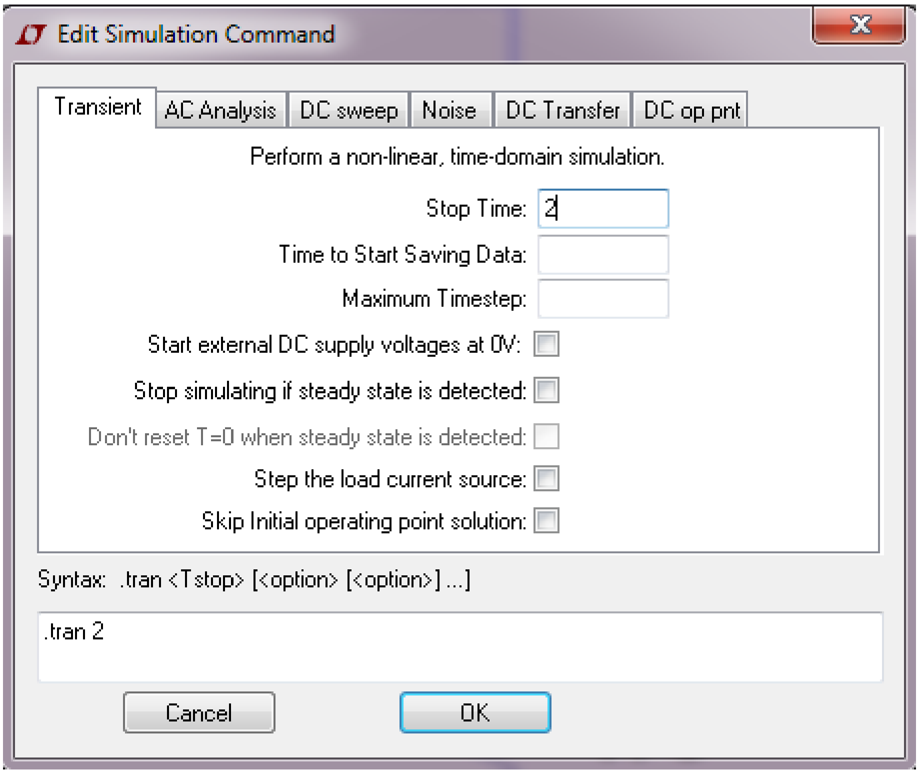


Figure 3-8: Selection of Specific Type of Simulation in LTSPICE IV

As we can see from **Figure 3-8**, we can do several types of analyses, for example AC analysis, DC operating points, transient, DC sweep and so on. For our case we like to do transient analysis of our circuit and for that we select ‘‘Transient’’ from the type of simulation tab and set the "Stop Time" value to 2. That is because we want to see the transient response of our circuit for 2 seconds. It will show with a text ".tran 2" in our schematic. We can notice that from **Figure 3-9** and **Figure 3-14** respectively.

With this short introduction to LTSPICE IV we are ready to simulate our own circuit.

3.4 Simulation in LTSPICE

To test the proper operability and safety margins, we have to simulate our circuit before building it. For that we simulate our voltage divider circuit with the help of LTSPICE IV simulator.

In simulation we used two 50 kΩ resistors connected in series to make a perfect voltage divider instead of one 100 kΩ. We also connect a zener diode which has a breakdown voltage of 3.3 V parallel to one of 50 kΩ (R2) resistors. The resistor R1 (50 kΩ) with the help of zener diode (D1) makes a voltage regulator circuit. This regulator circuit helps to get a stable output voltage without concerning of input voltage. We have to also keep concentration on the jumper position of X1 and X2 respectively. That is because by changing their positions we can change our load resistance from 10 Ω to 100 kΩ.

3.4.1 Power Adjustment Mode Simulation

It is the simplest one because on this mode the expected voltage coming from TEG is the range of 500 mV - 3V so that we do not need any voltage divider. The load resistance is 10Ω.

We can feed the output voltage taken across resistors R3 and R4 respectively connected in parallel direct to the analog input of the end device. As the maximum expected voltage is 3 V, in simulation we simulate with input voltage 3V.

From the **Figure 3-9** we can see that we made the 10Ω load resistor with combination of two 20Ω resistors. It causes to limiting the current flowing through each resistor which is better than to use a single 10Ω resistor. At the end of this topic we can find more details on that. Also keeping concentration on jumper X1 and X2. By changing the jumper position X2 we can select which load resistor we want to connect and by changing the jumper position X1 we can select from where we want to take out output.

We can see he total arrangement of power adjustment mode simulation in **Figure 3-9**.

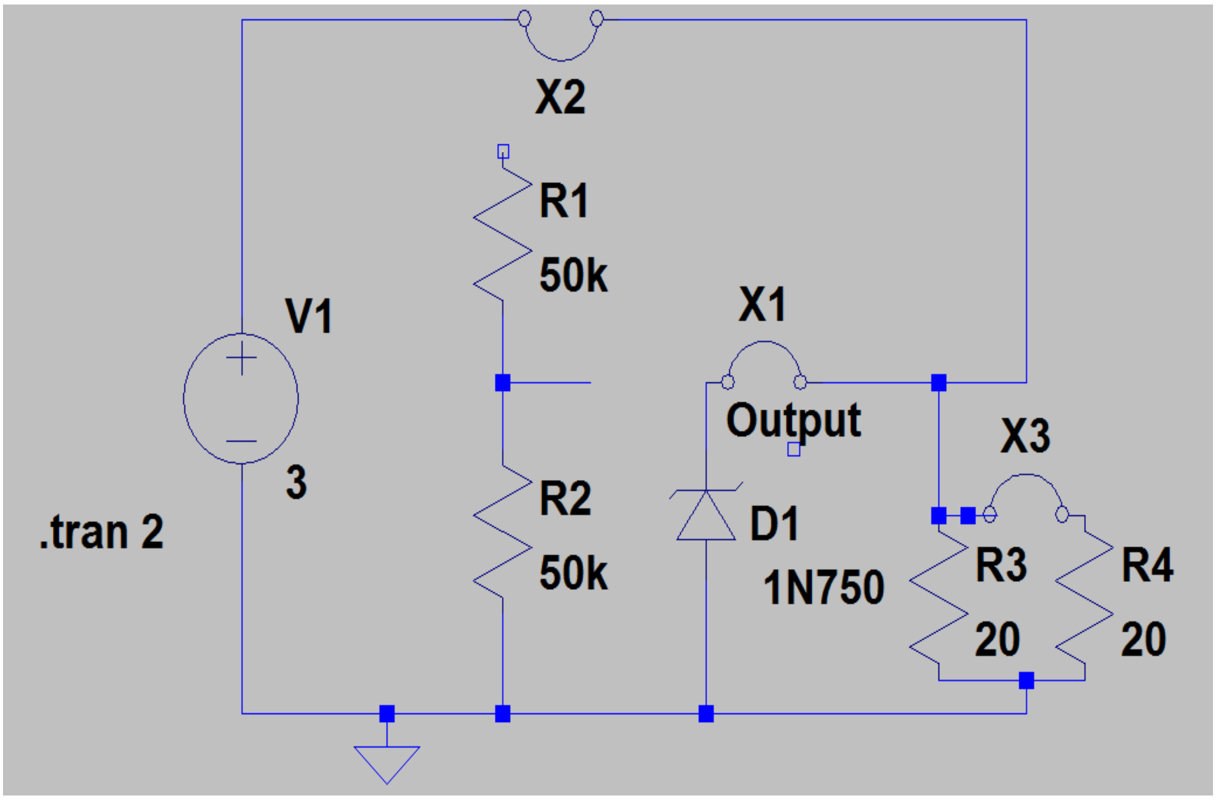


Figure 3-9: Power Adjustment Mode Simulation in LTSPICE IV

The output is taken across resistors R3 and R4 connected in parallel. It is shown in **Figure 3-10**.

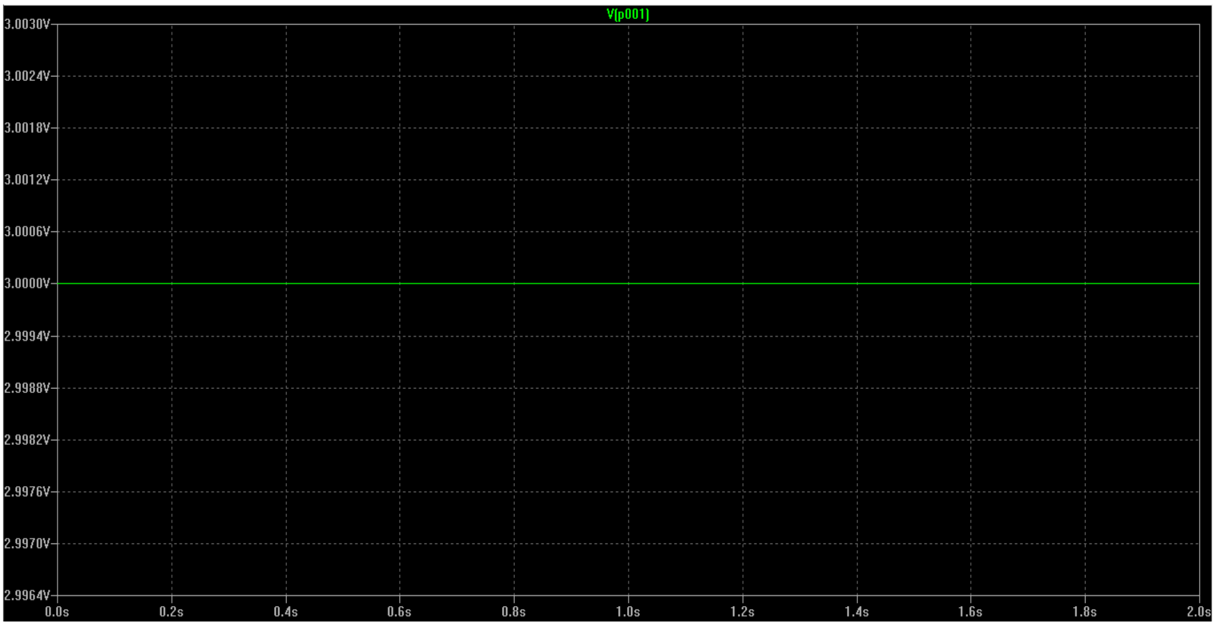


Figure 3-10: Voltage Output Waveform in Power Adjustment Mode in LTSPICE IV

From **Figure 3-10**, it does not look very fancy because the total incoming voltage is dropped in resistors R3 and R4. But if we look on the current waveform which flows through resistors R3 and R4 in **Figure 3-11**, then we find some interesting things.

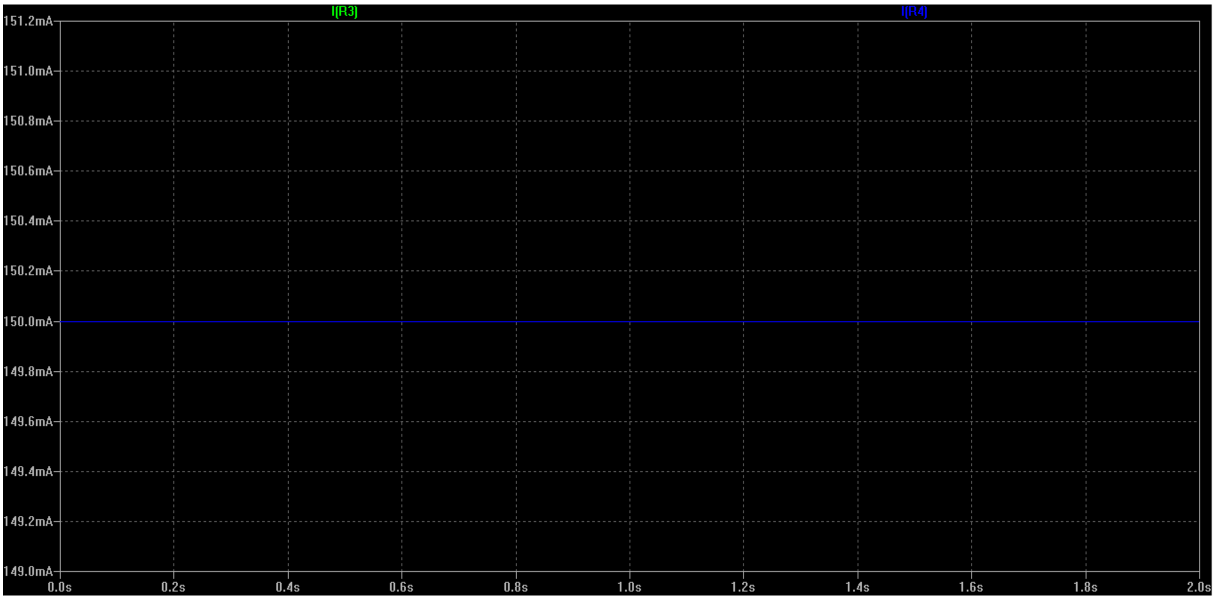


Figure 3-11: Waveform of Current Flowing Through Resistors R3 and R4 in LTSPICE IV

From **Figure 3-11** we noticed that current flowing through resistors R3 and R4 is 150 mA. It is pretty safe because we used two 20Ω resistors, ½ W resistor connected in parallel to make equivalent 10Ω resistor. But instead of those if we connected one single 10Ω resistor then current flow will be doubled which means 300 mA and also power rating of that resistor could be more than 1W. Due to the unavailability of one single 10Ω resistor and more than 1W resistor, we used two 20Ω resistors, ½ W resistor connected in parallel. The circuit with one single 10Ω resistor was simulated and it is shown in **Figure 3-12**.

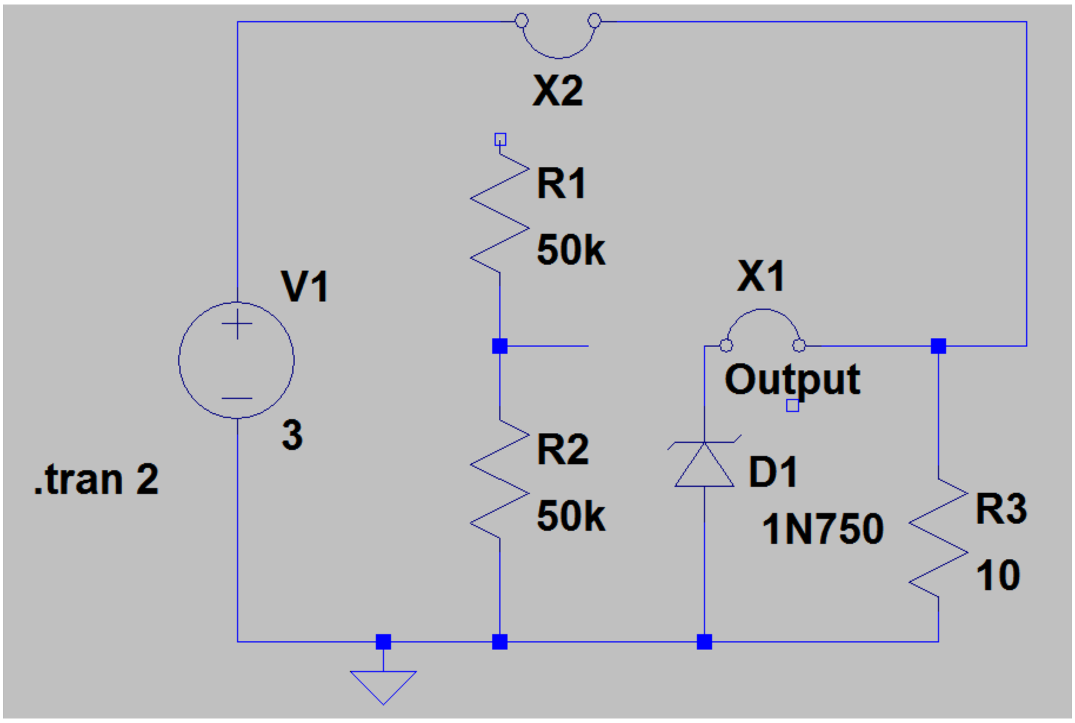


Figure 3-12: Circuit Simulated for One Single 10Ω Resistor Instead of Two 20Ω Resistors Connected in Parallel in LTSPICE IV

As we expected, current flow becomes double means 300mA which we can see from **Figure 3-13**.

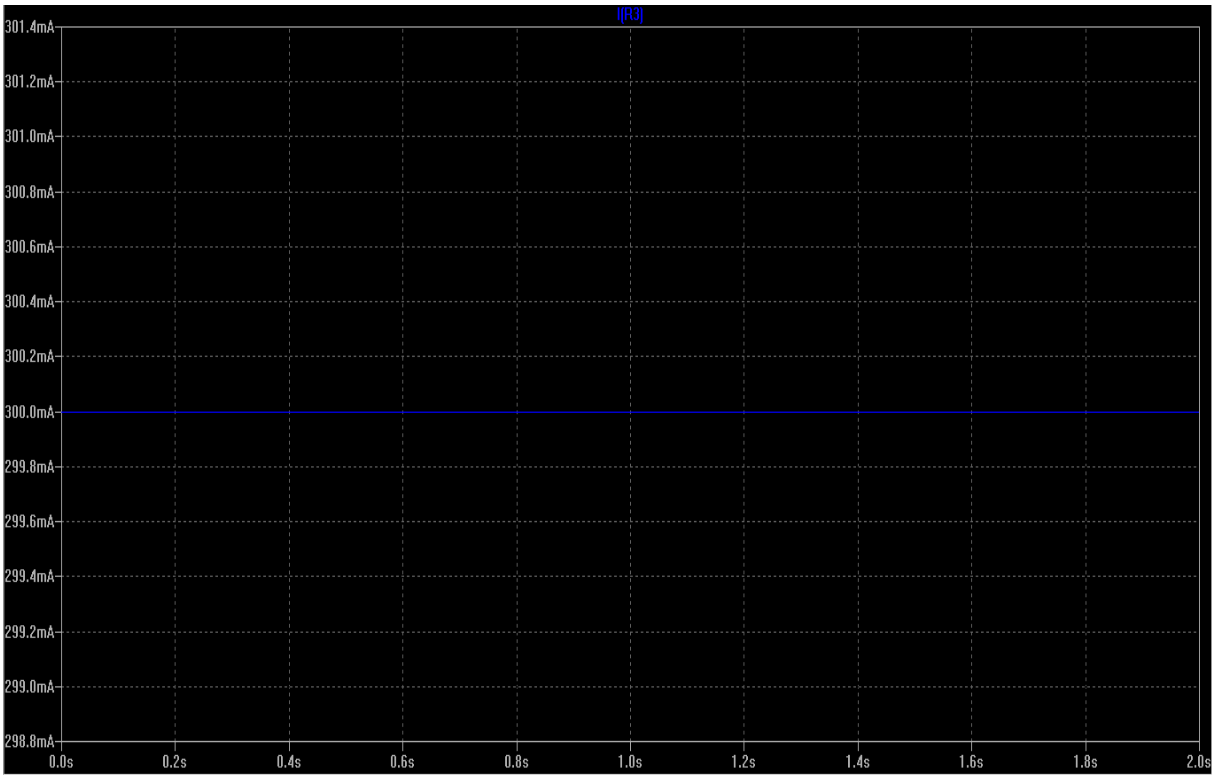


Figure 3-13: Waveform of Current Flowing Through 10Ω Resistor in LTSPICE IV

3.4.2 Idle Condition Mode Simulation

Now we simulate for 100kΩ load resistance by changing the jumper positions X1 and X2 as shown in **Figure 3-14**. We keep input voltage 6V as it is used in our practical experiment. The total arrangement we can see in **Figure 3-14**.

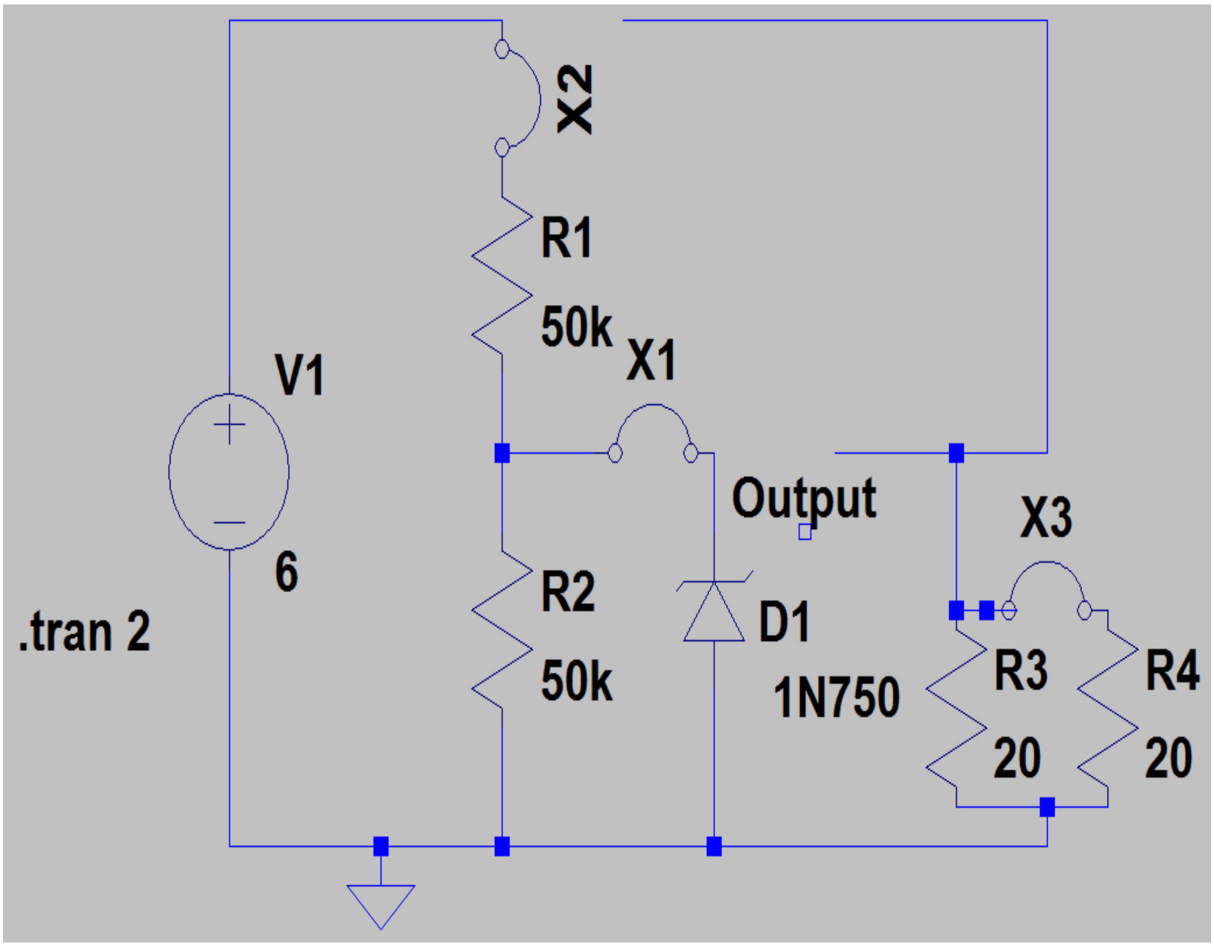


Figure 3-14: Voltage Divider Simulation in Idle Condition Mode in LTSPICE IV

The output is taking across one of the 50kΩ resistors. It is shown in **Figure 3-15** below.

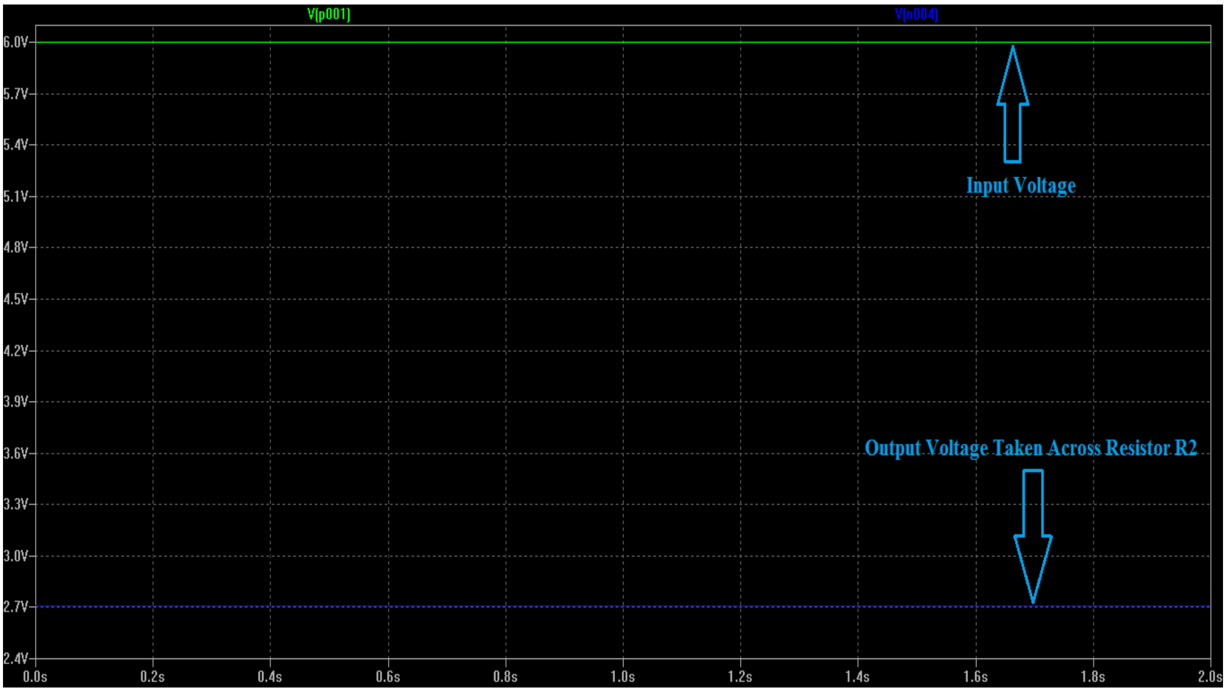


Figure 3-15: Output Voltage Waveform When Input Voltage is 6V in LTSPICE IV

We observe that there is a small mismatch in the output voltage. When the input voltage is 6V then we can expect the output voltage should be 3V in voltage divider. But actually we got 2.72V and this is because of using zener diode across the parallel of the output. It is shown in **Figure 3-16**.

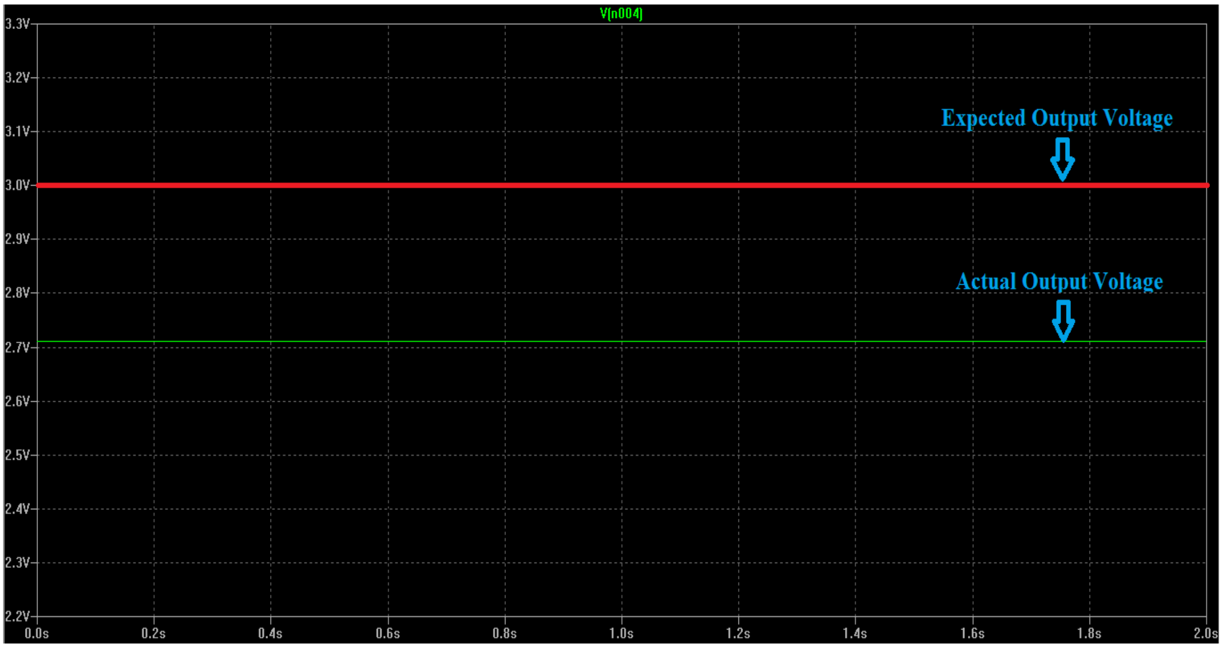


Figure 3-16: Mismatch of Output Voltage Due to Zener Diode in LTSPICE IV

For example, if the input voltage of **Figure 3-14** increases to double (12V) then the output voltage should be near about 3.3V. That is because we used the zener diode which has zener breakdown voltage of 3.3V. For that reason we can not see any change of output voltage with the change of input voltage. It is shown in **Figure 3-17**.

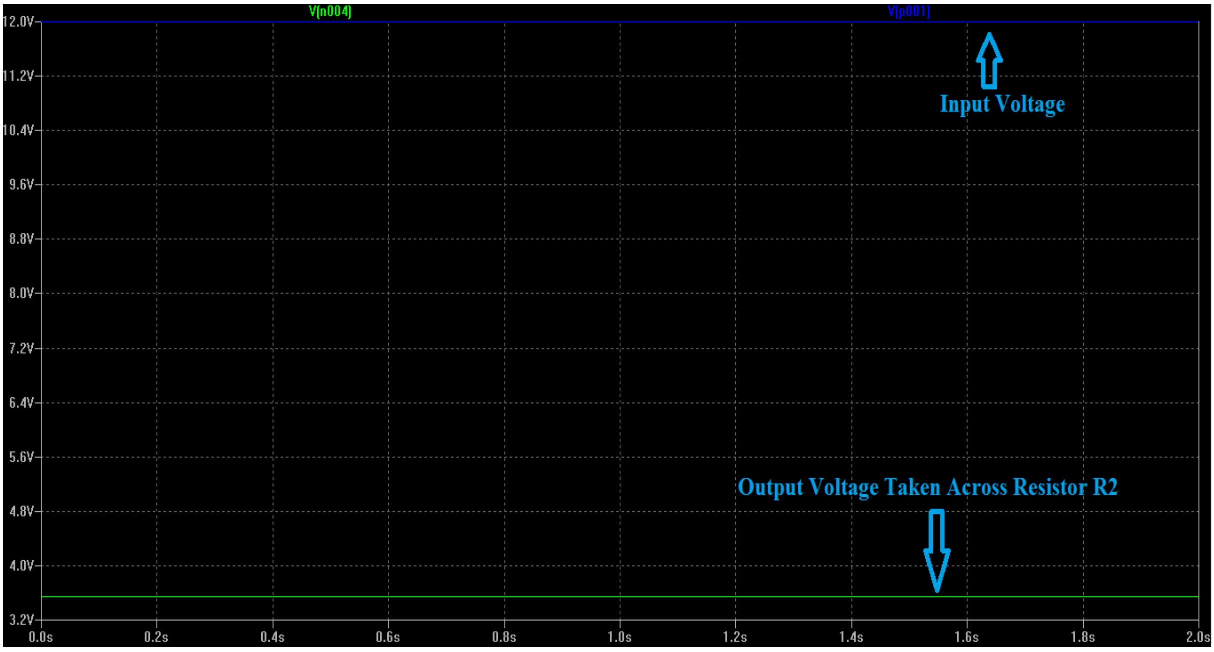


Figure 3-17: Stable Output Voltage Due to Zener Diode in LTSPICE IV

3.5 DC/DC Boost Converter Design

In this section, we looked at the possibility of powering the sensor board (sender unit) of the development kit EZ430-RF2500 from TI using the low voltage obtained from a TEG. For that a DC/DC boost converter is first simulated in LTSPICE based on the expected output voltage from the TEG during operation. Afterwards an integrated circuit was found and soldered into the circuit board.

The very first test in the field (at Trauen) was conducted with the wireless development kit EZ430-RF2500 from TI powered by using a lithium battery (see **Figure 3-18**).

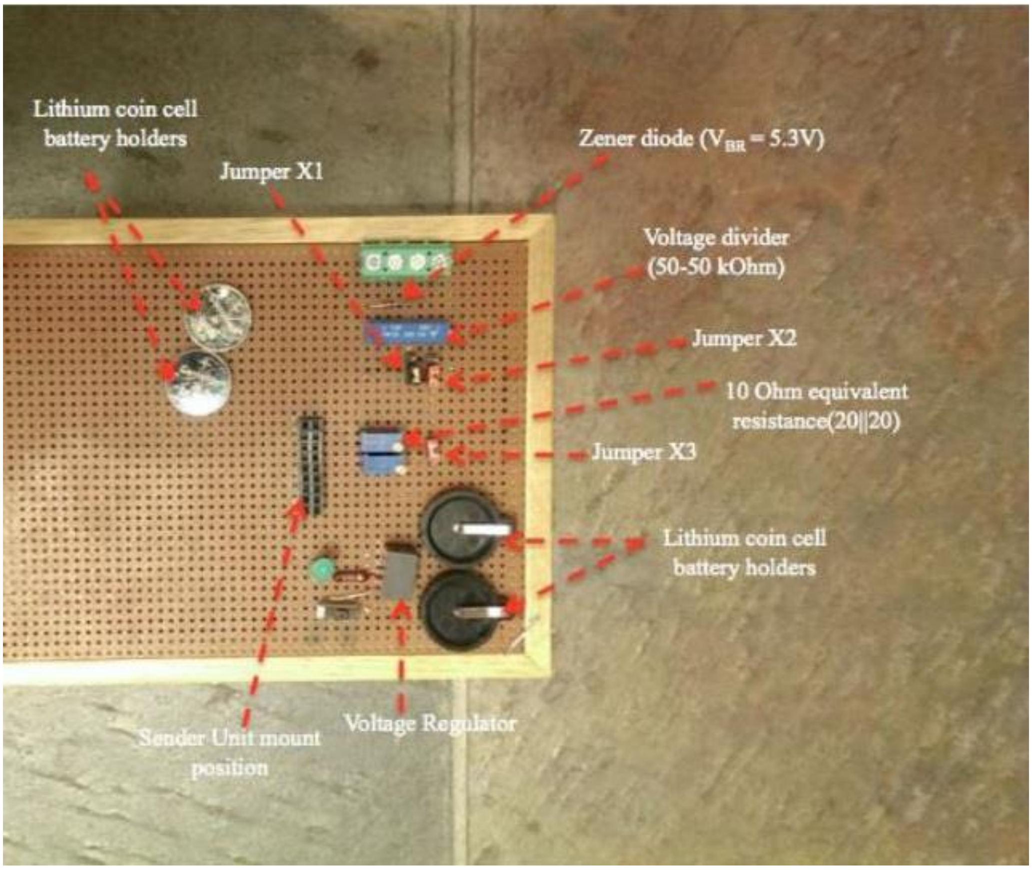


Figure 3-18: Battery Powered Circuit Showing Soldered Components

Afterwards, Compatriots at the Center of Applied Space Technology and Microgravity (ZARM) could not permit the use of batteries on their rocket for safety reasons. Therefore, there arose the need to power the sender unit (ED) via the TEG heat to voltage conversion. However, a boost converter needed to be designed and built in order to amplify the generated voltage to the requirement of the sender device.

In order to interface the TEG voltage with the sender unit and the boost converter it is necessary to investigate the following issue,

- The minimum and maximum voltage to power the sender unit.

Table 3-2: Electrical Parameters for the Sender Unit

Operating Conditions	Minimum	Maximum
Operating supply voltage	1.8 V	3.3 V
Operating power	$4.86*10^{4}$ W	

Based on the above information, basic boost converter parameters were estimated and modeled in LTSPICE software. The LTSPICE software is a Spice simulator of electronic circuits produced by linear technology (LTC). The software library includes prebuilt in integrated circuits models based on linear technology devices in order to speed simulation of switching regulators. After thorough investigation about converters, the integrated circuit

LTC3105 was found to meet the requirement needed to power up the sender unit from the low voltage obtained from the TEG.

3.5.1 LTC3105 Integrated Circuit (IC)

The LTC3105 is a 400mA step up DC/DC converter that can operate directly from low voltage such as those from photovoltaic cells and TEGs [21]. The circuit can start from voltage as low as 225mV. The device incorporates a maximum power point control (MPPC) that a user can program to set points that maximizes the energy that can be extracted from a power source. Some of its features are as follows [21]:

- Low start up voltage: 250mV
- Maximum power point control
- Output disconnect and inrush current limiting
- Automatic power adjust
- Soft start
- Output voltage range 1.6 V to 5.25V

Schematic description of the pins is shown in **Figure 3-19** below.

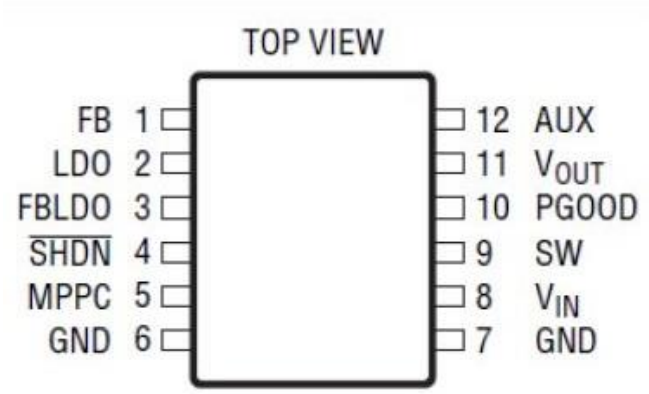


Figure 3-19: Top View Pin Configuration of LTC3105 IC

The IC chip requires very few components to be attached and soldered to the pin configuration to work very efficiently and meet the requirement of use. Based on the recommendation of the IC manufacturer, components were selected, modeled and tested in the LTSPICE environment. The **Figure 3-20** below shows the IC chip model and the selected components.

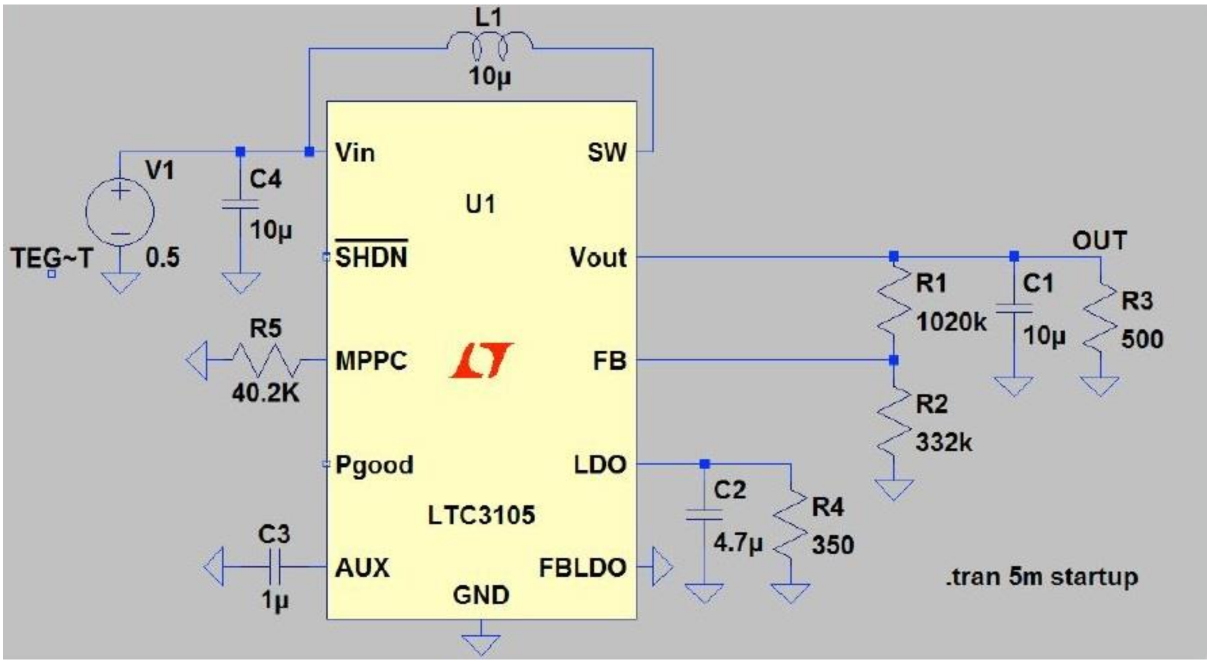


Figure 3-20: LTC3105 and Components Model in LTSPICE IV

The objective here is to check the output voltage of the step up converter to meet the required amount to power the sender unit. The LTC3105 output voltage can be selected between the range of 1.5V to 5.25V depending on the ratio of the resistance R1 and R2 at pin V_{out}. This divider is then fed-back into the circuit in order to have a loop and adjust the output voltage. The value R1 and R2 is adjusted and re-modeled with the chip several times until an output voltage sufficient enough that meets the requirement is produced. A 10µF capacitor is connected between output and ground to act as a filter and reduce the output voltage ripple. Simultaneously, the IC includes a built in linear regulator (LDO) which uses the output voltage to produce an output and this can also be used to power some external circuitry. However, in this configuration the LDO has not been used. The chip also has a shutdown pin (SHDN) which when enable, it goes into quite mode and does not consume power. This pin could also be used to shut down everything.

Since this is a boost converter, it definitely requires an external inductor between switch (SW) and V_{in}. The inductor value was selected based on the manufactures recommendation when the chip is intended for high impendence energy sources like TEG. The switch that is used for the boost conversion process is of course built in internally into the IC itself. The IC has a low voltage start-up which is propriety of the manufacturer. Internally the MOSFET devices are operated in a sub threshold region so that they are very sensitive to small voltages. And once a small voltage is given the switch kicks in and takes current from the inductor and brings the output voltage high enough and once the voltage reaches a certain value the rest of the circuit takes forward and start working.

During start up, the auxiliary voltage pin (AUX) will reach 1.4V and once it reaches that the circuit enters its normal operation. This pin is used by the start-up circuitry to generate a voltage to power the internal circuit until the output voltage reaches regulation. The AUX and V_{out} are internally connected together.

The **Figure 3-21** below shows the simulation result of the circuit. The simulation was conducted for a transient time of 5 milliseconds (ms). The important result from the simulation was the output voltage produced and the time it takes to produce the output voltage. This is needed in order to estimate the time it will take the converter to boost the sender unit. From the wave form plot it can be seen that at 2.2 ms the voltage is 1.8V which is sufficient to power the sender unit. The converter reaches steady state at 3.0 ms.

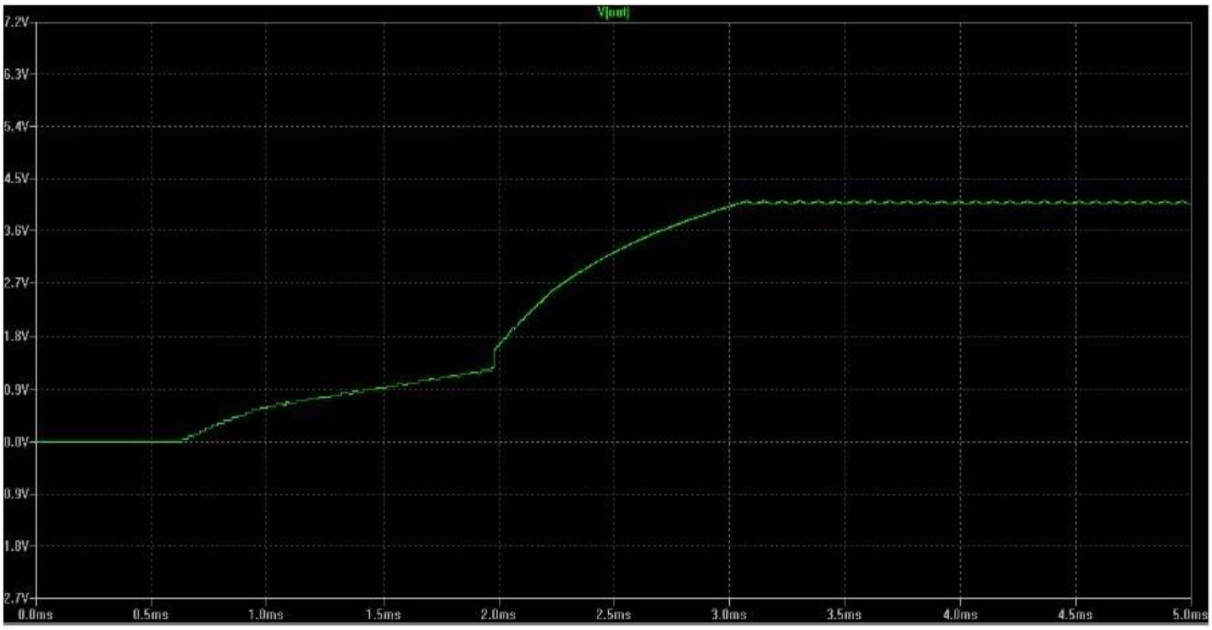


Figure 3-21: Step-Up Converter Output Waveform in LTSPICE IV

However, such a circuit has not been used in the project. Later it was found that a similar IC "U1V11F3" which can deliver the same functionality exists. It is cheaper and ready available in the market. We will further discuss about the device in next session.

3.5.2 Components Selection

The load resistance where the voltage from the TEG dropped is selected before. Resistor 10Ω in power adjustment mode and resistor 100 kΩ in idle condition mode are chosen. Also thermoelectric generator ''TEG 127-200-27'' is chosen before. Now our task is to select the rest of the components. Based on the functional block diagram as shown in **Figure 3-1**, it is shown that 'which components are selected and why' in **Table 3-3** below.

Table 3-3: Selected Components List and the Reasons for Specific Selections

Component Name	Features	Why it should be selected?
Boost Converter "Pololu U1V11F3[15]"	• Input voltage range: 0.5V to 5.5V • Output voltage: 3.3V fixed • It has ability to both boost and down regulation of input voltage. • It is integrated over temperature shutdown. • Very small and compact design.	As it has the ability to both boost and down regulation, that features made this boost converter unique. The voltage comes from TEG is not a fixed level voltage (can be less or more then 3.3V), so we need this type of boost converter.
Voltage Regulator "Traco Power TSR 1-2433[16]"	• Input voltage range: 4.75V to 36VDC • Output voltage: 3.3VDC fixed • Output Current Max. : 1.0 A • Operation temperature range: -40°C to +85°C. • Short circuit protection.	This voltage regulator has a wide input voltage range, sufficient output current and voltage. It also has a fare operating temperature range which is very important for our project.
Zener Diode "1N746A[17]"	• Zener Voltage (V_Z) : 3.3 V • Maximum zener current rating (I_{ZM}) : 110 mA • Zener Impedance (Z_Z) : 28 Ω at I_Z = 20 mA • Operating temperature range : -65°C to +200°C	This zener diode has a zener voltage of 3.3 V which helps to protect the analog input of the end device from over voltage.

3.5.3 Pololu ‘‘U1V11F31’’ Step-Up Voltage Regulator

The step up voltage converter that we selected is from a company named Pololu and the model number of that device is ‘‘U1V11F31’’. The reason of choosing this device is explained in ‘‘Component Selection’’ section above. In this section we will discuss about the device and its feature in detail manner.

The Pololu ‘‘U1V11F31’’ (0.45″×0.6″) comes with a very small package, so that it can be compatible in any tiny space. Also it has the ability to boost up the output voltage 3.3 V when input voltage is as low as 0.5 V. Besides it has the linear down-regulation mode when the input voltage exceeds the output and a true shutdown option that turns off power to the load.

Another advantage of this device is that it does not need any external component to make it function. The device is shown in **Figure 3-22** below.

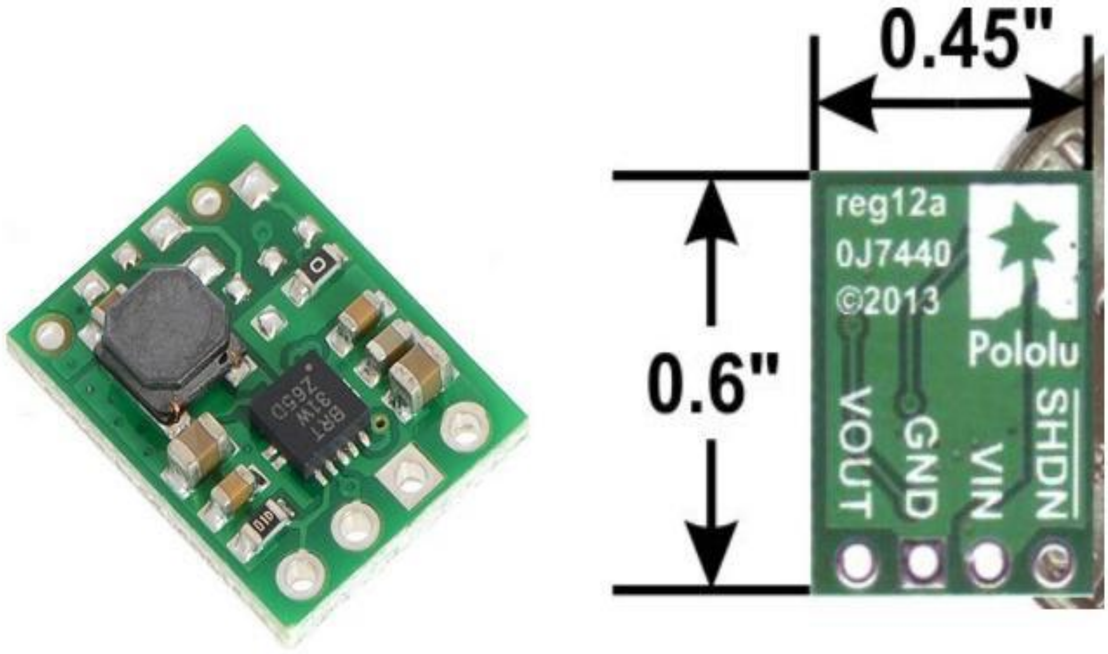


Figure 3-22: Pololu ''U1V11F31'' Step-Up Voltage Regulator

The device has four connection pins. Their functions are listed in **Table 3-4** below.

Table 3-4: Functions of Pololu ''U1V11F31'' Pins

Pin Name	Function
$\overline{\text{SHDN}}$	Shutdown
VIN	Input Voltage
GND	Ground
VOUT	Output Voltage

The input voltage gets its supply directly from the TEG and the output pin is fed to the power pin of the sender unit board.

The pin $\overline{\text{SHDN}}$ provides the function to turn off power to the load. Initially this pin is pulled up by a 100 kΩ resistor, so that it can be left disconnected. Another way is that it can be connected directly to VIN if we do not want to use the disable feature.

The disable threshold which is a function of the input voltage is discussed in **Table 3-5** below.

Table 3-5: The Conditions for Input Voltage Pin VIN of Pololu ‘‘U1V11F31’’

Condition	Things to do
VIN < 0.8 V	$\overline{\text{SHDN}}$ voltage must be above 0.9×VIN to enable the regulator and below 0.1×VIN to disable the regulator.
0.8 V ≤ VIN ≤ 1.5 V	$\overline{\text{SHDN}}$ voltage must be above 0.8×VIN to enable the regulator and below 0.2×VIN to disable the regulator.
VIN > 1.5 V	$\overline{\text{SHDN}}$ voltage must be above 1.2 V to enable the regulator and below 0.4 V to disable the regulator.

There are some few conditions that we have to keep in mind when we use this device. They are,

- The input voltage VIN must be at least 0.5 V to turn on the regulator. Once the regulator is turn on, the input voltage can go down to 0.3 V and it will maintain 3.3 V as an output voltage.
- The input voltage should not exceed 5.5 V.
- We have to think about LC spikes that might cause the input voltage to surpass 5.5 V.

3.5.4 Efficiency vs Output Current of Pololu ‘‘U1V11F31’’

The efficiency of Pololu ‘‘U1V11F31’’ Step-Up Voltage Regulator is an important issue. This is shown in **Figure 3-23** below.

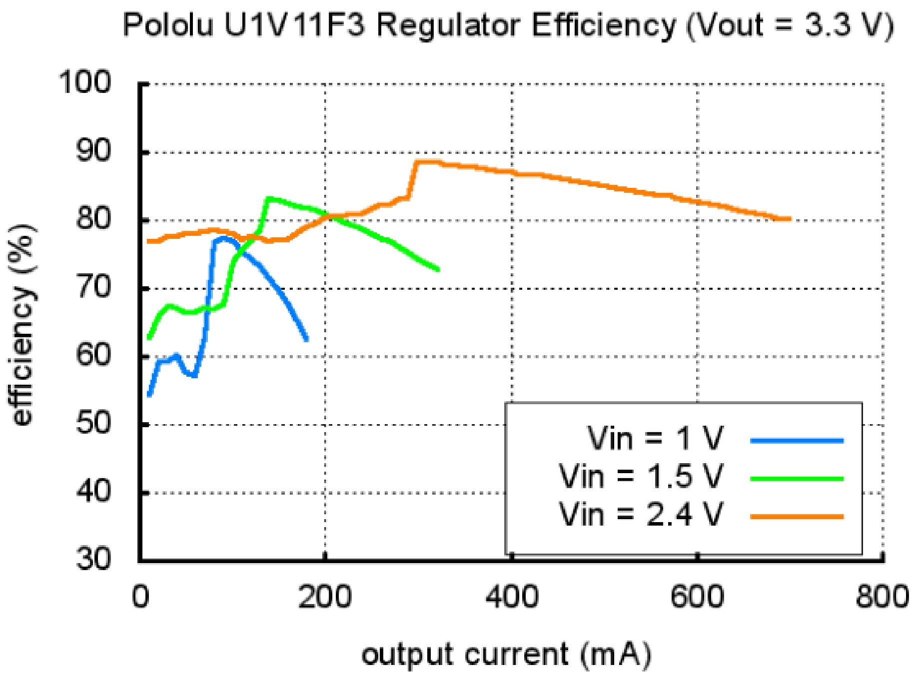


Figure 3-23: Efficiency Curve of Pololu ‘‘U1V11F31’’ [15]

It is seen from the efficiency curve that the Pololu ‘‘U1V11F31’’ Step-Up Voltage Regulator typically has an efficiency of 70 to 90%. Also if the input current crosses the switching current limit (typically somewhere between 1.2 and 1.5 A), the output voltage will begin to drop. The maximum output current is also affected by some environmental factor for example ambient temperature, air flow, heat sinking etc.

3.6 MATLAB GUI Design

In this section, we look at a detailed design (programming) and data collection of real-time data from the TI MSP430 wireless development toolkit (EZ430-RF2500). The programming language of choice used was MATLAB, a high-level fourth-generation programming language and multi paradigm numerical computing environment. The next subsection will elucidate the written program and relevant figures will be presented for visualization purpose.

3.6.1 GUI Programming

A user interface (UI) is a graphical display in one or more windows containing controls, called components, which enable a user to perform interactive tasks. The user does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user does not need to understand the details of how the tasks are performed. [20]

UI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. UIs created using MATLAB® tools can also perform any type of computation, read and write data files, communicate with other UIs, and display data as tables or as plots. [20]

In this subchapter, the programming will be explained according to the subsections as listed below,

- Create GUI Icons
- Check and open serial port
- Declare parameter variables and make figure visible
- Function definitions

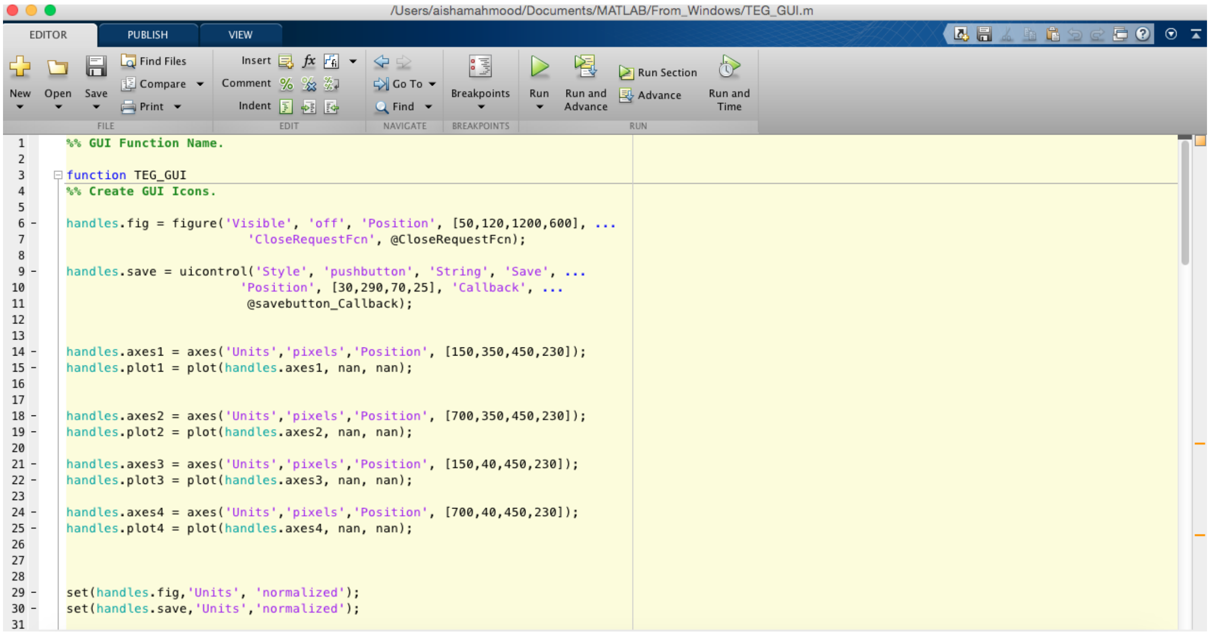
Before we go into the details of the GUI design, it is worth explaining the need for it in this project. The TEG parameters (Voltage, Power, Signal Strength, etc.) are required to be visualized in real-time and saved in order to recreate plots conveniently at a later time for analysis and revisualization purposes. As such, MATLAB was chosen as the program of choice because of its ease, growing popularity as a programming choice, and availability (already installed).

3.6.2 Creating GUI Icons

In this subsection of the program, **Figure 3-24** shows the code snippet for the **"Create GUI Icons"** section. In MATLAB, a GUI can be created in one of two ways:

- Using GUIDE – This stands for Graphical User Interface Development Environment, which itself is an interactive development environment created by MATLAB to facilitate ease of GUI design for users. This is explained briefly in Appendix A, as it is not the method exploited in the project.
- Programmatically – writing a function program for the desired GUI. In this project, the GUI is designed using this method.

The GUI is created as a function as seen in **Figure 3-24** (function TEG_GUI). Under this section, every desired icon to be created is prefaced by a “handles.” definition. A handle is a structure that contains all the properties of each created icon, and can be used conveniently to change any property at any time, as shown in lines 29 and 30 in **Figure 3-24**. The GUI figure is created first because all other GUI components are situated on its area. The figure has various properties (Name: represents the title displayed on the GUI, Visible: makes the GUI visible depending on the property value (on/off), Position: this is a four-element vector that specifies position and size of the UI on the screen [distance from left and distance from bottom, width and height of the GUI, etc.]) and also a callback function (@CloseRequestFcn: line 104, **Figure 3-27**) that closes/exits the GUI when the close button is clicked. A callback function is a function that is executed every time the icon is clicked, as is the case with the close button.



```
1   %% GUI Function Name.
2
3   function TEG_GUI
4   %% Create GUI Icons.
5
6   handles.fig = figure('Visible', 'off', 'Position', [50,120,1200,600], ...
7                        'CloseRequestFcn', @CloseRequestFcn);
8
9   handles.save = uicontrol('Style', 'pushbutton', 'String', 'Save', ...
10                          'Position', [30,290,70,25], 'Callback', ...
11                           @savebutton_Callback);
12
13
14  handles.axes1 = axes('Units','pixels','Position', [150,350,450,230]);
15  handles.plot1 = plot(handles.axes1, nan, nan);
16
17
18  handles.axes2 = axes('Units','pixels','Position', [700,350,450,230]);
19  handles.plot2 = plot(handles.axes2, nan, nan);
20
21  handles.axes3 = axes('Units','pixels','Position', [150,40,450,230]);
22  handles.plot3 = plot(handles.axes3, nan, nan);
23
24  handles.axes4 = axes('Units','pixels','Position', [700,40,450,230]);
25  handles.plot4 = plot(handles.axes4, nan, nan);
26
27
28
29  set(handles.fig,'Units', 'normalized');
30  set(handles.save,'Units','normalized');
31
```

Figure 3-24: Code Snippet for “Creating GUI Icons” Section

The ui control (user interface control) contains components such as push buttons, radio buttons, sliders, static text, etc. The first icon created is the save button with the name “handles.save”. The properties dictate the type of user interface icon (Style: “pushbutton”), the pushbutton label (String: “save”), its position, with the placement format similar to that of the aforementioned and explained created figure, and its callback function (@savebutton_Callback: line 64, **Figure 3-27**).

Next, the first plot axes (handles.axes1) are created. This can be seen in **Figure 3-27** below. The only properties defined are the units (pixels) and the position. Pixels are the default unit of the UI. Therefore, it is necessary to specify it as such so that all the icons have the same unit. This will later be normalized (in order that all the other components resize with the figure) after all the icons have been placed and the GUI is to be made visible. If another unit is chosen for an icon before normalizing the figure, that icon will not resize simultaneously with the figure. Therefore, homogeneity of units should be adhered to.

Next, the handles.plot1 handle is created. Its properties contain the axes1 handle "handle.axes1". This is essential in order to plot the chosen parameter in that axes. The other two parameters are the "x" and "y" axes, which are set to ''nan'' (not a number) to create allowance for the real-time data to be plotted later on. The other three axes and plots are created and work in a similar way.

Next, the figure unit property is set to normalize. The "set" function is used to change the property value of a structure. There is also a "get" which acquires the property value of a structure and displays it. This is very useful when a user wants to know what value is set for which property without having to go through the numerous lines of code. As we progress further, relevant points concerning this section will be further be explained.

Check and Open Serial Port:

Figure 3-25 shows the code snippet of this section. Here, line 34 reads: "newobj = instrfind;". The function ''instrfind'' is used in order to check and return all communication interface objects that exist in memory. The communication interface objects are returned as an array to ''newobj''. The if expression checks to see if the array ''newobj'' is not empty, closes the files associated with the array, and deletes it from the memory (workspace). The global that precedes the object s definition is required as it is used throughout the program and not just locally. The "disp" displays the enclosed string and the s object is created using the serial function (constructs a serial port object associated with the port, COM3 in our case) and has the following defined properties:

- Port – COM3. This is the port number that is associated with the created object. The COM port number should be checked.
- BaudRate – This is the data transfer rate in a communication channel. The specific value 9600 means that the serial port is capable of transferring 9600 bits per second (bits/s).
- Terminator – This is used to reset the position of the cursor for displaying the received data. The value CR/LF means carriage return/line feed. In this case, the cursor position is reset to the beginning of the line and then moved one line forward.
- InputBufferSize - This specifies the size of the buffer for storing data. It is 1024 in our case.

Line 43 shows the s.BytesAvailableFcnMode and its value is set to terminator. This means that a bytes-available event occurs when the terminator specified by the terminator (CR/LF) is reached or when the number of bytes specified by the BytesAvailableFcnCount (a property of the s port structure) property is available in the input buffer.

The s.BytesAvailableFcn executes the callback function @SerialDataReceived (line 74, **Figure 3-27**) when the bytes-available event s.BytesAvailableFcnMode occurs. This will be explained in a later section.

The fopen function opens the serial port s for read access. Then the COM3 open string is displayed.

```
32  %% Check and open serial port.
33
34 - newobj = instrfind;
35 - if ~isempty(newobj)
36 -     fclose(newobj);
37 -     delete(newobj);
38 - end
39
40 - global s;
41 - disp('GUI started...')
42 - s = serial('COM3','BaudRate',9600, 'Terminator', 'CR/LF','InputBufferSize', 1024);
43 - s.BytesAvailableFcnMode = 'terminator';
44 - s.BytesAvailableFcn = @SerialDataReceived;
45 - fopen(s);
46 - disp('COM3 open...')
```

Figure 3-25: Code Snippet for "Check and Open Serial Port" Section

Declare Parameter Variables and Make Figure Visible:

The lines 49 to 52 in **Figure 3-26** are required variable declarations. The default load resistance value is 100 ohms. The variables ''oldvoltage'' and ''oldstrength'' are required in order to concatenate later values of these variables that will be received in real-time. The next lines set the properties of the created figure.

- Name – displays the specified string on top of the GUI.
- NumberTitle – specifies the Figure window title number, such as Figure 1, 2 etc., where 1, 2 are the title numbers.
- Resize – this enables resizing of the figure window.
- Visible - makes the GUI visible.
- MenuBar – this either shows or hides the menu bar.

```
47  %% Declare parameter variables and make figure visible.
48
49 - oldVoltage  = 0;
50 - oldStrength = 0;
51 - Resistance  = 100;
52 - power       = 0;
53
54 - global timeElapsed;
55 - timeElapsed = 0;
56 - set(handles.fig,'Name','TEG DATA ACQUISITION');
57 - set(handles.fig,'NumberTitle','off');
58 - set(handles.fig,'Resize','on');
59 - set(handles.fig,'Visible', 'on');
60 - set(handles.fig,'MenuBar', 'none');
61
```

Figure 3-26: Code Snippet for "Declare Parameter Variables and Make GUI Visible" Section

Function Definitions:

Here the used callback functions are explained. These are:

- savebutton_Callback
- SerialDataReceived
- CloseRequestFcn

savebutton_Callback

This is shown on line 64 in **Figure 3-27**. Here, the function ''assignin'' function is used. It has arguments (WS, variable, value). The WS stands for workspace and can either be base or caller. The base refers to the MATLAB base workspace while the caller refers to the workspace of the caller function. The base is chosen in our case. Variable is the variable name in which the data will be saved, and the value stands for the data values to be saved in the variable.

This function is executed every time the save button is pressed. It is a simple and effective way of saving collected real-time data to the MATLAB workspace. Another method is the xlswrite function, which writes to excel files directly. This was not used because some latency was observed on saving data at the expense of collecting data. Moreover, additional code is required in order to ensure that all cells (for different parameters) of a particular save-session are filled. However, a convenient alternative was used. An Excel add-on was added to MATLAB and enabled users to export/import arrays directly to/from Excel. This method is only possible for Excel versions 2007 and later. This is shown in **Figure 3-28**. On pressing the save button as stated earlier, the ‘‘savebutton_Callback’’ function is executed. This saves the variables "vol, str, p" and "t" to the MATLAB workspace. It is worth mentioning that an instance of Excel should be started first. This will launch an instance of MATLAB so as to enable import/export functionality. If an instance of MATLAB is started first (and we can launch multiple instances), and then an instance of Excel, the Excel instance will launch its own instance of MATLAB, causing multiple instances running at once, and confusion. Therefore, this should be noted and adhered to.

When the MATLAB icon in the top right hand corner of the Excel window is clicked, it shows a drop down (numbered 1 in **Figure 3-28**) with the options:

- Start MATLAB
- Send Data to MATLAB
- Get Data from MATLAB
- Run MATLAB command
- Get MATLAB Figure
- MATLAB function wizard
- Preferences

Once the "Get Data from MATLAB" option is chosen, a window appears (numbered 2 in **Figure 3-28**) asking for "matrix to get from MATLAB." Once the variable name is entered and the ok button clicked, the cells are populated. The variable from MATLAB must be a column vector in order to populate the cells vertically. This can be done by adding an apostrophe **‘** to the end of the variable name. Else they would be populated horizontally.

```matlab
%% Function definitions.

    function savebutton_Callback(~,~)

        assignin('base','vol',oldVoltage');
        assignin('base','str',oldStrength');
        assignin('base','p',power');
        assignin('base','t',timeElapsed');

    end

tic;
    function SerialDataReceived(Sender, EventArgs)

        time = toc;
        if strcmp(EventArgs.Type, 'BytesAvailable')

            data = char(fscanf(Sender));
            [device,type,voltage,strength,other] = strread(data,'%s%s%3.2f%d%s','delimiter',',');
            fprintf('%s\n',data);
            oldVoltage  = [oldVoltage,voltage];
            oldStrength = [oldStrength,strength];
            power = (oldVoltage.*oldVoltage)/Resistance;
            set(handles.plot1,'XData',timeElapsed,'YData',oldVoltage);
            xlabel(handles.axes1, 'Time,[s]');
            ylabel(handles.axes1, 'TEG voltage,[V]');
            title(handles.axes1,'TEG voltage');
            set(handles.plot2,'XData',timeElapsed,'YData',oldStrength);
            xlabel(handles.axes2, 'Time,[s]');
            ylabel(handles.axes2, 'Signal strength,[%]');
            title(handles.axes2,'Tx signal strength');
            set(handles.plot3,'XData',timeElapsed,'YData',power);
            xlabel(handles.axes3, 'Time,[s]');
            ylabel(handles.axes3, 'TEG power,[W]');
            title(handles.axes3,'TEG power');
            timeElapsed = [timeElapsed time];
        end

    end

    function CloseRequestFcn(~, ~, ~)

        if exist('s', 'var')
            fclose(s)
            delete(s)
            clear s
        end
        usedtimer = timerfindall;
        if ~isempty(usedtimer)
            stop(usedtimer);
            delete(usedtimer);
            clear usedtimer;
        end
        delete(handles.fig)

    end

end
```

Figure 3-27: Code Snippet for "Function Definitions" Section

SerialDataReceived

This has the arguments Sender and EventArgs. The Sender is the handle to the function ‘‘SerialDataReceived’’ and ‘‘EventArgs’’ is a structure with various properties. The "if" conditional statement checks whether both strings in the parentheses are equal (using the function string compare). If true, then the Sender is scanned for available data and converted to a character array. The ‘‘strread’’ function reads the converted array and assigns each variable on the right of the equality sign the appropriate data type and width. Then the received data are printed (displayed) to the MATLAB command window as shown in **Figure 3-29**. The oldVoltage, oldStrength, and power have all been initialized earlier. This is the method used to concatenate arrays in MATLAB. The set function sets the "x" and "y" properties of the handle.plot1 to the specified values (timeElapsed and old Voltage). The xlabel, ylabel, and title of the handles.axes1 are set accordingly. This is done for the remaining two axes (handles.axes2 and handles.axes3).

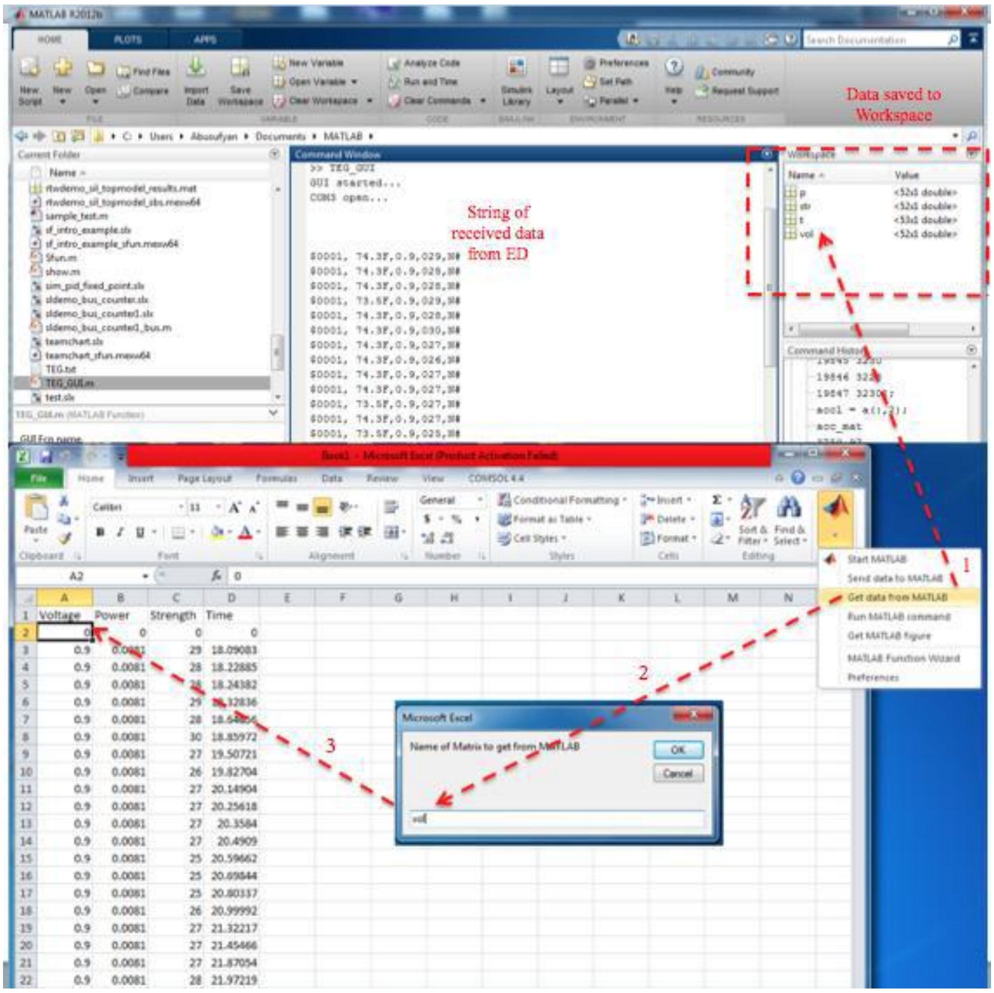


Figure 3-28: MATLAB Interface Showing Saved Data (top) to Workspace and Excel Interface Showing Imported Data (bottom)

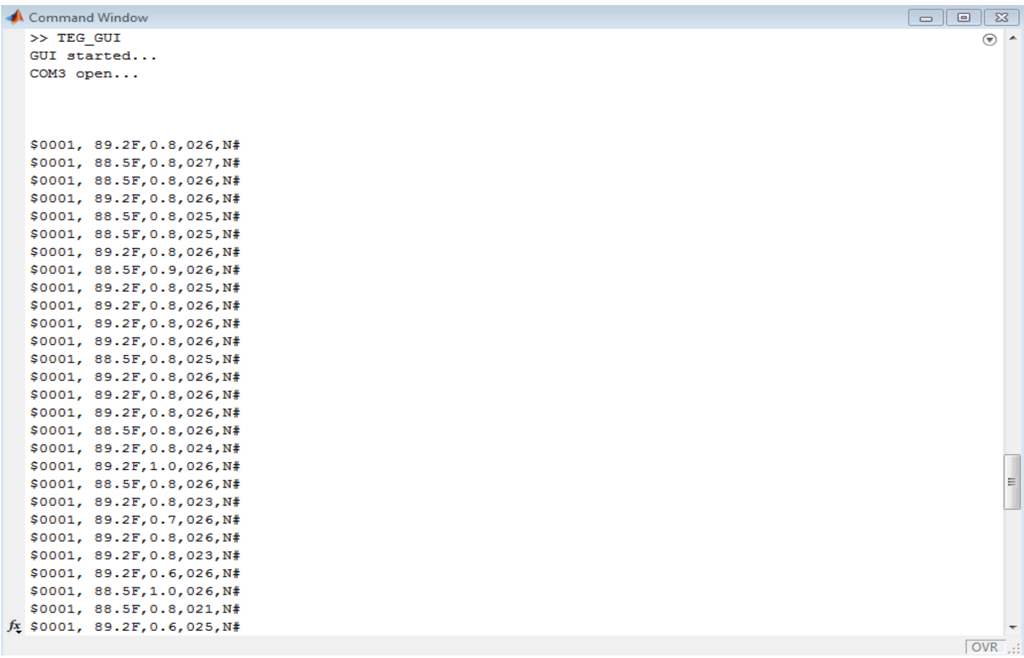


Figure 3-29: MATLAB Command Window Displaying Received Data (Device ID, Temperature (F), Voltage and Signal strength)

CloseRequestFcn

This function contains two if conditional statements. The first (line 104, **Figure 3-27**) checks whether the created serial port exists, and then closes it, deletes it, and clears it from the workspace in order that next time the GUI is launched, there would not be any existing serial port objects, avoiding errors. The other if statement is included because a timer object was to be included, to refresh the plots at intervals.

3.7 Epilogue

It was required to investigate and design a DC/DC boost converter that will amplify the voltage received from a TEG to power a sensor board. An integrated circuit was chosen and built in the Spice environment. Based on the simulation result it was shown that the IC and its selected components can fully provide the required functionality. However, the simulated IC could not be used. Instead a similar device was chosen as it does not require any external components to be attached to it in order to function. After conducting several experiments in the laboratory with the TEG, it was found that the slowly rising voltage of the thermoelectric generator cannot power the sender unit; the device requires some sort of rising edge to start. However, a rising edge can be created manually and one general conclusion was that the low voltage generated from the TEG can indeed be used to power the sender unit. But for that, a circuit that can provide rising edge is needed. Also, a GUI that could display the desired TEG parameters (Voltage, Signal Strength, Power and temperature) was designed. The code was written in MATLAB as a function file. The complete code is shown in Appendix B.

Chapter 4
EXPERIMENT and EVALUATION

This chapter describes the following topics respectively:

4.1 Overview on Experiment
4.2 Laboratory Test
4.3 Field Test

4.1 Overview on Experiment

After building the complete hardware, several tests was conducted in the laboratory to check if the chosen converter is able to power the sender unit from the low voltage received from the TEG. We also conducted several field tests in Trauen. As a first test, a DC power supply was used in the lab instead of the TEG. The DC power supply is acting as a voltage source from the TEG. The voltage supply is increased slowly from 0 to around 0.5V. At 0.5V, the voltage across the step up converter Vout pin is 3.3V. However the sender unit is unable to power up. The same experiment was conducted with different TEGs.

4.2 Laboratory Test

The laboratory test set up of the project is built up inside the laboratory with the TEG clipped with a steel mounted on the wood stand, the hot air supplied by a hot gun (not more than 200 °C), the ice inside a jug, the thermocouples and measuring units as shown in **Figure 4-1** below. We used the same instruments as the former group used as we the both groups carried out the same project.

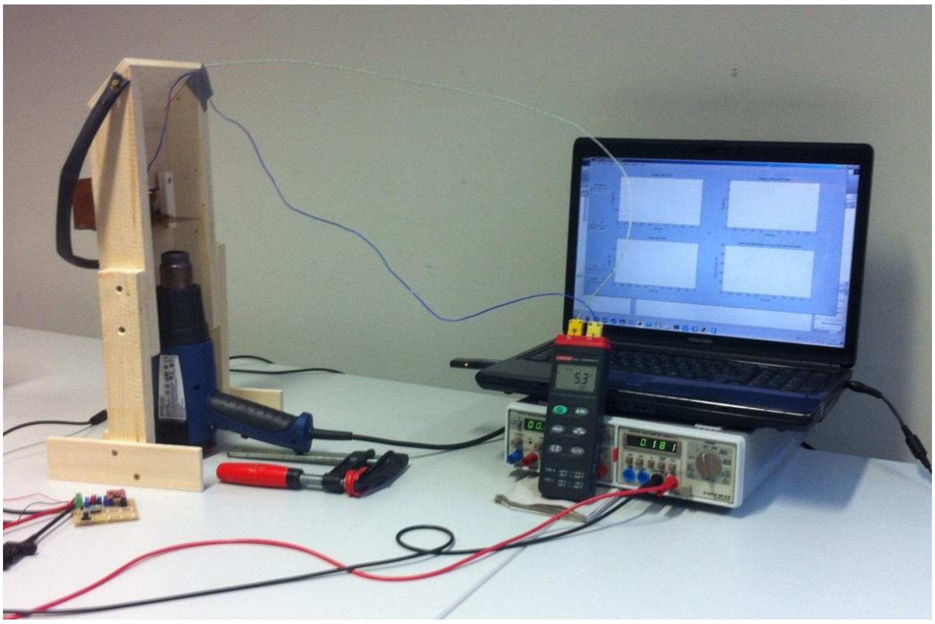


Figure 4-1: The Experimental Set Up in Laboratory

In the indoor experiment set up, the artificial hot air is supplied to the hot side and the cold temperature created artificially by the ice in the jug is applied to the cold side of the TEG. The cold side is attached to the heat sink. The artificial environment is created to produce electrical power before we conducted the final outdoor experiment in Trauen. The temperature difference across the module generates a small electrical voltage measured by the Multimeter.

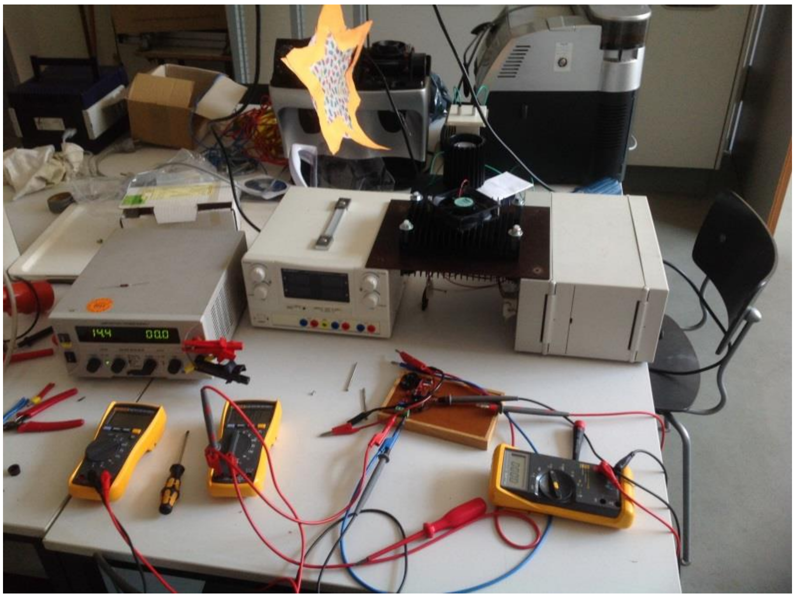


Figure 4-2a

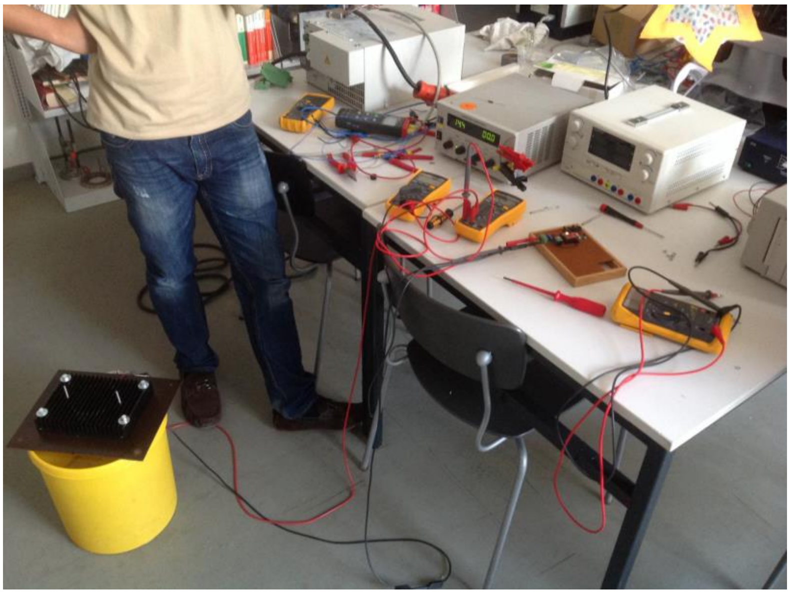


Figure 4-2b

Figure 4-2 (4-2a & 4-2b): Laboratory Test Set Up Showing Circuit Board, TEG and DC Power Supply

4.2.1 Laboratory Test Result

The plots of the laboratory test results (Voltage, Signal Strength, and Power) displayed in the MATLAB command window are shown in **Figure 4-3** and **Figure 4-4** respectively. **Figure 4-3** is when the resistance R is 10 ohms (indicated by the selected radio button). This gives a higher power value compared to when the resistance R is 100 ohms as shown in **Figure 4-4**.

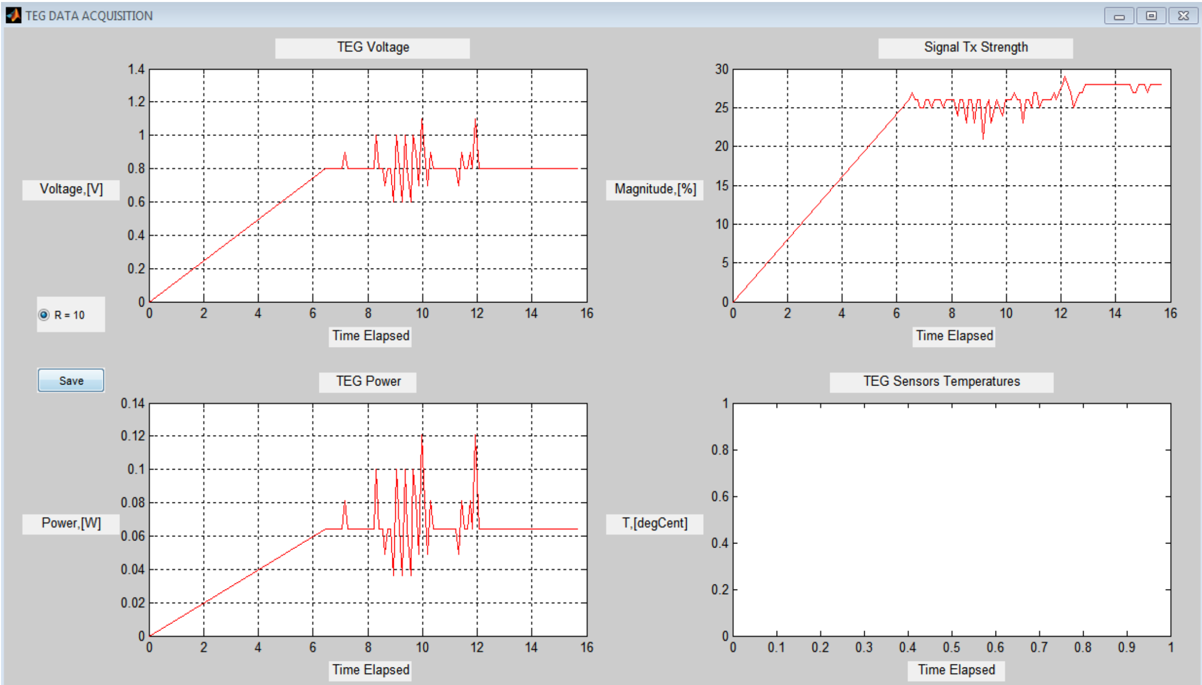


Figure 4-3: GUI Displaying Voltage, Power and Signal Strength Traces for 10 ohms Resistance

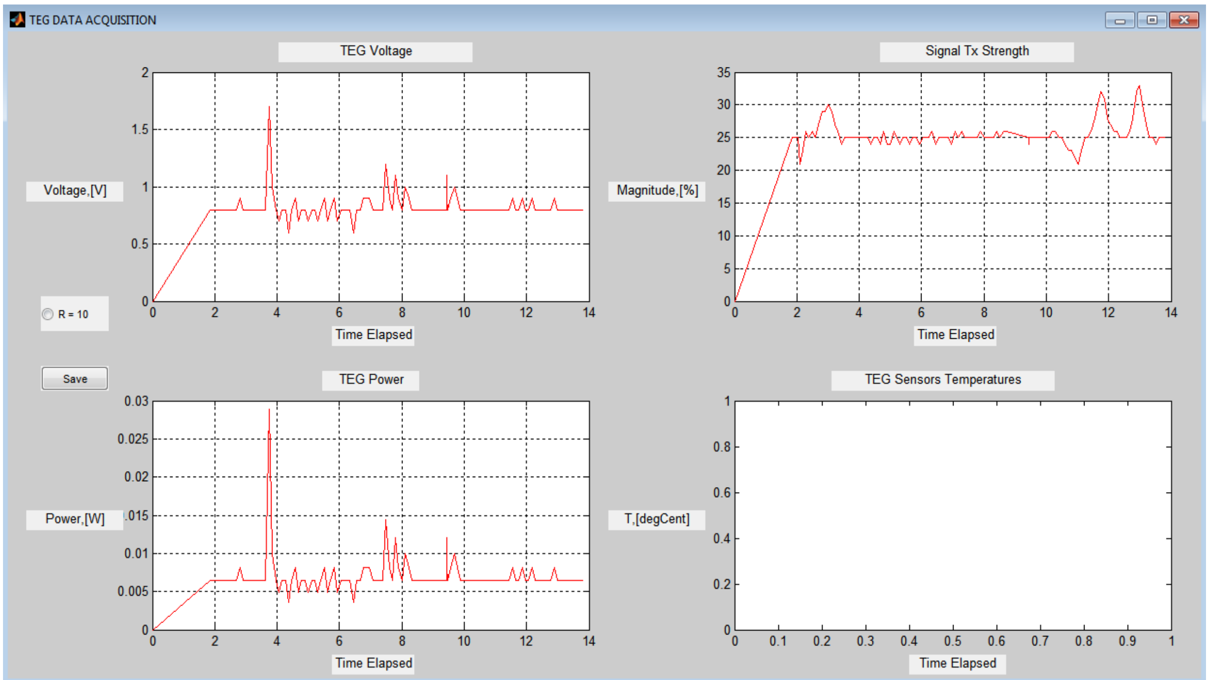


Figure 4-4: GUI Displaying Voltage, Power and Signal Strength Traces for 100 ohms Resistance

4.3 Field Test

The outdoor test set up of the project was carried out by Airbus Defense and Space, Bremen in a remote area called Trauen. The outdoor set up was conducted after building circuit and laboratory test.

This set up contains the rocket engine which emits waste heat and two TEGs (as this project was carried out by two groups) were attached inside the heat sinks separately to the two metal stands as shown in **Figure 4-5** below.

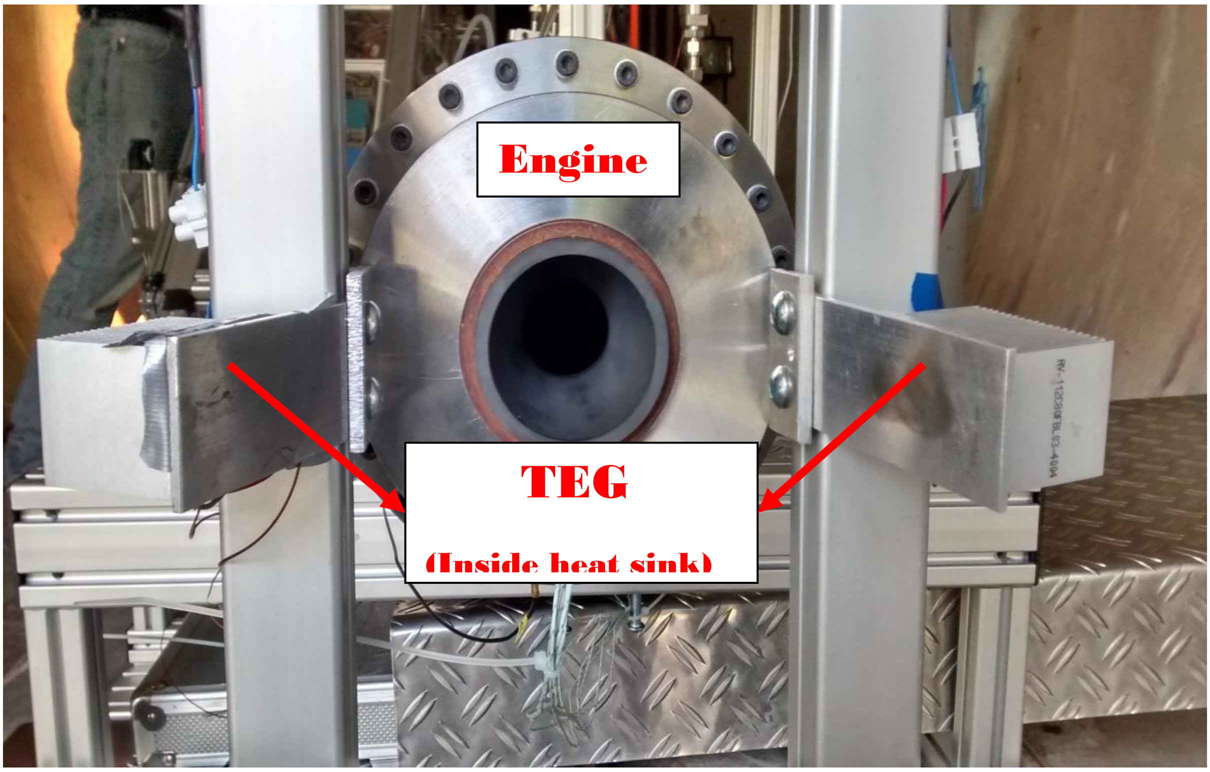


Figure 4-5: The Field Test Set Up in Trauen with Rocket Engine

The rocket engine was controlled by the controller in a control room that is situated little bit far from the set up for safety purpose. The waste heat (hot) emitted from the engine and the surrounding very low temperature (cold) of the environment creates a temperature difference across the TEG which works as Seebeck effect. This process leads to the production of a low electrical power across the resistive loads as mentioned in **Table 3-1**.

The further arrangements for the field test set up are shown in **Figure 4-6** and **Figure 4-7** below respectively.

Figure 4-6: The PC to Receive Data from Circuit Attached to TEG in Field Test

Figure 4-7: The Circuit Attached to TEG in Field Test

4.3.1 Field Test Result

Unfortunately, due to the very cold weather in Trauen during the experiment running, the battery of the designed circuit connected to the output power terminal of TEG got down and it stopped functioning properly. As a result the sender unit could not deliver the measured data to the receiver unit connected to the host computer. So we did not get the desired result of our project.

Chapter 5
CONCLUSION

The goal of the project is partially achieved. The voltage from TEG is perfectly measured and sending the measured data wirelessly to the access point is done successfully. The measured data was shown in real time manner in MATLAB GUI. Also in laboratory experiment the voltage measured data for different load in different mode of operation was investigated carefully. However, in field test we did not get any measured voltage data from the TEG. That problem occurred because of sudden shut down of our end device after engine test was started. After finishing the engine test, we went for investigation of the problem and we found that it was the problem of battery. We used lithium coin cell battery which has a thermal rundown problem. The outside temperature of Trauen was -5°C at that time. So our lithium coin cell battery was affected by that low temperature although we kept the circuit near a portable heater.

On the other hand, the DC/DC boost converter that was chosen and built in the LTSPICE environment was not used. Instead of that DC/DC boost converter another similar device was chosen which does not require any external components to make it function. But the device Pololu ‘‘U1V11F31’’ step-up voltage regulator has the limitation on its own. It does not make response with slowly rising voltage from TEG that can power up the end device. The device requires a rising edge to start. In the laboratory experiment the rising edge was created by manually switching the TEG voltage in the input of Pololu ‘‘U1V11F31’’.

Future Task:

After conducting several experiments it was found that the sender unit does not operate with a slowly rising voltage, so it requires a rising edge for it to start. This is because of its intended operation while using a battery. A Schmidt trigger based circuit could be used in the future to provide a sharp edge needed for the sender unit to start.

APPENDIXES

Appendix A

Figure A-1 shows the GUIDE startup window. This can be launched by simply typing GUIDE at the MATLAB command window. The two tabs show “Create New GUI” and “Open Existing GUI”, depending on the desired choice. If the former option is chosen, it opens the interactive development environment as shown in **Figure A-2**.

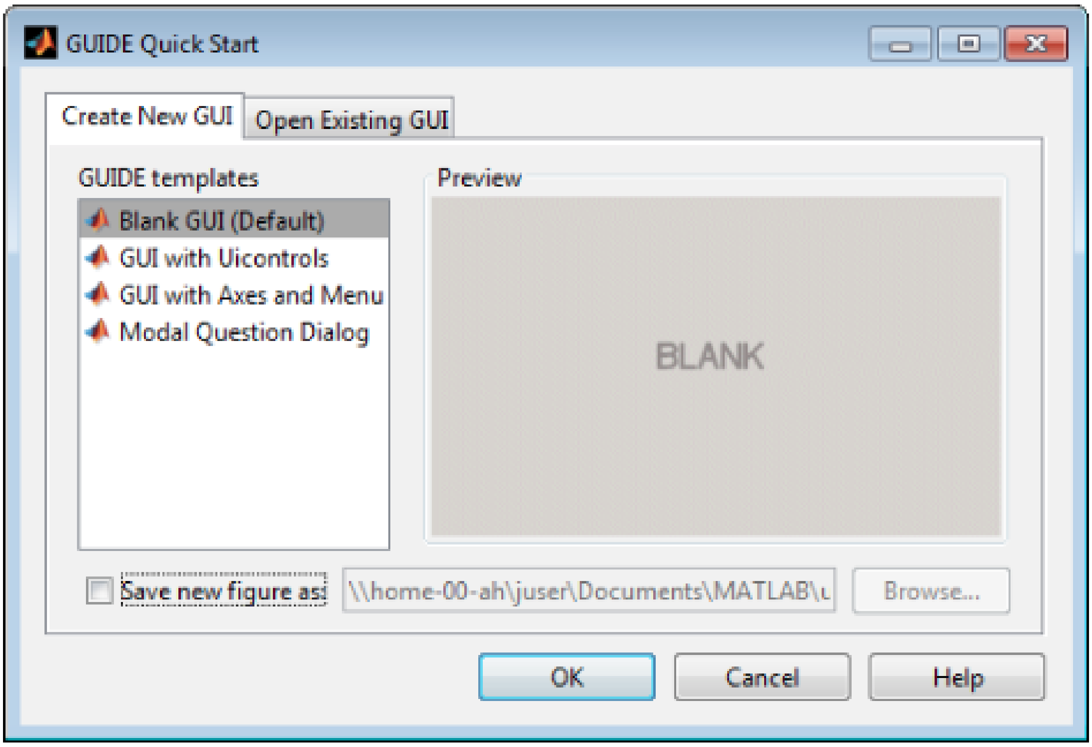


Figure A-1: GUIDE Startup Window

In **Figure A-2**, the left section contains the user interface components (pushbutton, slider, radiobutton, etc.). The user can then drag whichever icon is required and drop it into the environment to create an instance of it. Once done, it is required to save the created GUI, which creates two files (a figure file and a function file). The figure file is required in order to edit the GUI at any time, while the function file is a programmatic equivalent of the figure file, which is automatically created once the GUI is saved.

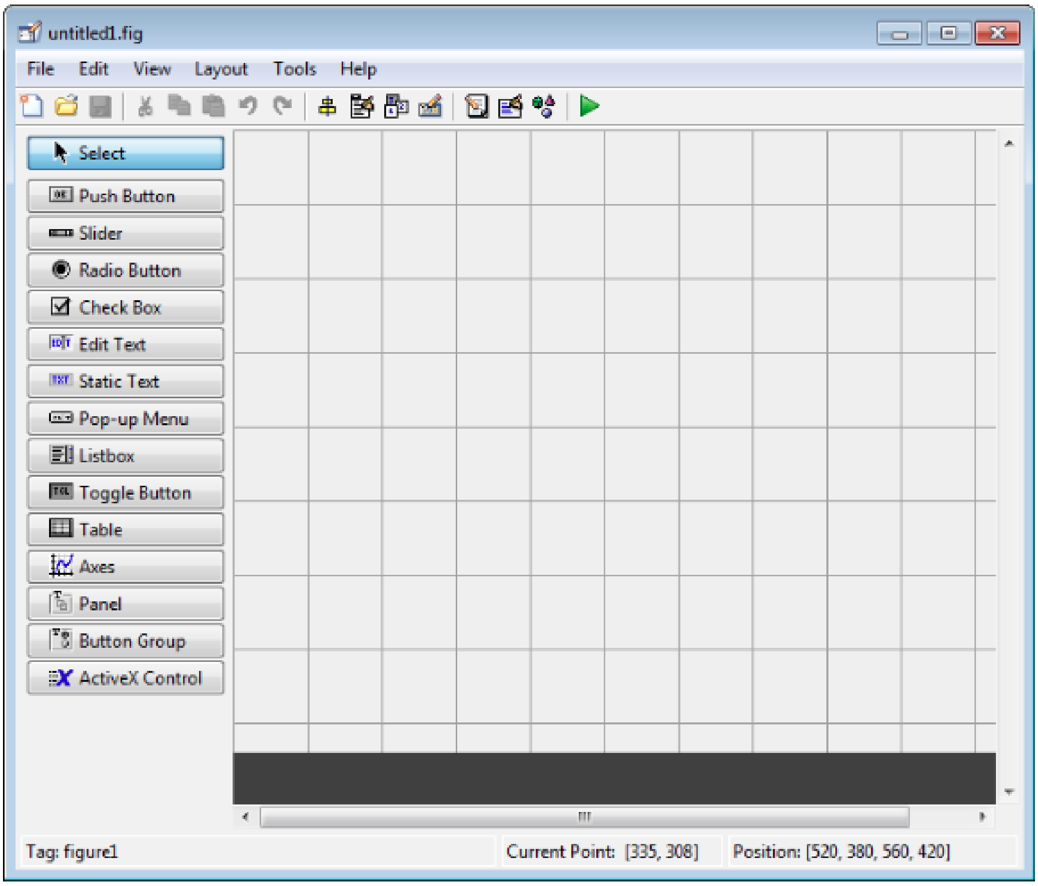


Figure A-2: GUIDE Development Environment

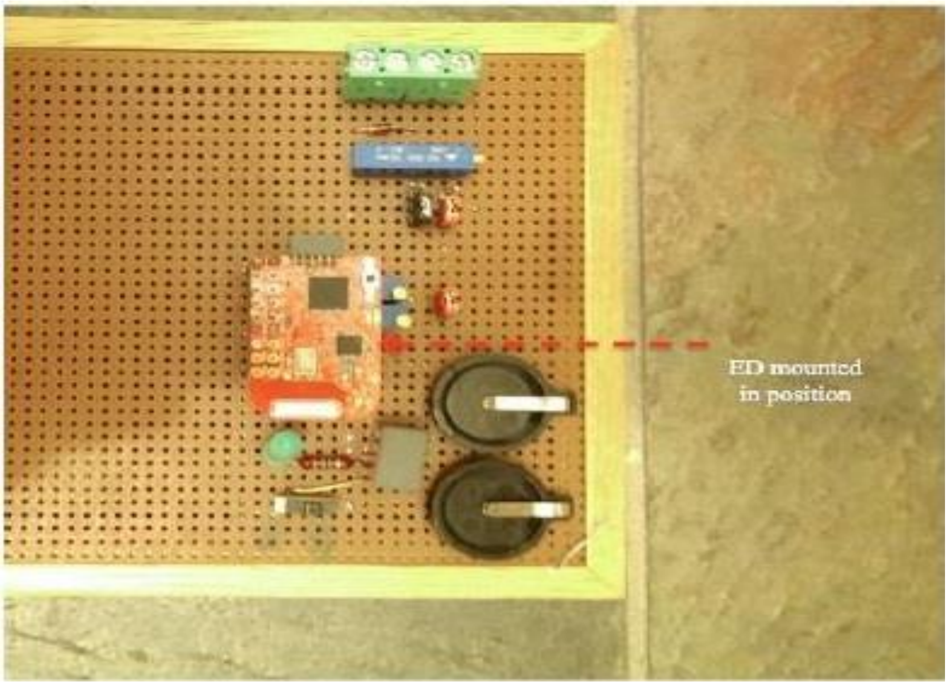


Figure A-3: Hardware Design Board Showing Sender Unit with Other Components

Appendix B

Created GUI MATLAB code.

```
function TEG_GUI
%% Create GUI Icons.

handles.fig = Figure('Visible', 'off', 'Position', [50,120,1200,600], ...
                    'CloseRequestFcn', @CloseRequestFcn);

handles.save = uicontrol('Style', 'pushbutton', 'String', 'Save', ...
                    'Position', [30,290,70,25], 'Callback', ...
                    @savebutton_Callback);

handles.axes1 = axes('Units','pixels','Position', [150,350,450,230]);
handles.plot1 = plot(handles.axes1, nan, nan);

handles.axes2 = axes('Units','pixels','Position', [700,350,450,230]);
handles.plot2 = plot(handles.axes2, nan, nan);

handles.axes3 = axes('Units','pixels','Position', [150,40,450,230]);
handles.plot3 = plot(handles.axes3, nan, nan);

handles.axes4 = axes('Units','pixels','Position', [700,40,450,230]);
handles.plot4 = plot(handles.axes4, nan, nan);

set(handles.fig,'Units', 'normalized');
set(handles.save,'Units','normalized');

%% Check and open serial port.

newobj = instrfind;
if ~isempty(newobj)
        fclose(newobj);
        delete(newobj);
end

global s;
disp('GUI started...')


s = serial('COM3','BaudRate',9600, 'Terminator', 'CR/LF','InputBufferSize', 1024);
s.BytesAvailableFcnMode = 'terminator';
s.BytesAvailableFcn = @SerialDataReceived;
```

```
fopen(s);
disp('COM3 open...')
%% Declare parameter variables and make Figure visible.

oldVoltage = 0;
oldStrength = 0;
Resistance = 100;
power = 0;

global timeElapsed;
timeElapsed = 0;
set(handles.fig,'Name','TEG DATA ACQUISITION');
set(handles.fig,'NumberTitle','off');
set(handles.fig,'Resize','on');
set(handles.fig,'Visible', 'on');
set(handles.fig,'MenuBar', 'none');

%% Function definitions.

    function savebutton_Callback(~,~)
        assignin('base','vol',oldVoltage');
        assignin('base','str',oldStrength');
        assignin('base','p',power');
        assignin('base','t',timeElapsed');
    end

tic;
    function SerialDataReceived(Sender, EventArgs)

    time = toc;
    if strcmp(EventArgs.Type, 'BytesAvailable')

    data = char(fscanf(Sender));
    [device,type,voltage,strength,other] = strread(data,'%s%s%3.2f%d%s','delimiter',',');
    fprintf('%s\n',data);
    oldVoltage = [oldVoltage,voltage];
    oldStrength = [oldStrength,strength];
    power = (oldVoltage.*oldVoltage)/Resistance;
    set(handles.plot1,'XData',timeElapsed,'YData',oldVoltage);
    xlabel(handles.axes1, 'Time,[s]');
    ylabel(handles.axes1, 'TEG voltage,[V]');
    title(handles.axes1,'TEG voltage');
    set(handles.plot2,'XData',timeElapsed,'YData',oldStrength);
```

```
    xlabel(handles.axes2, 'Time,[s]');
    ylabel(handles.axes2, 'Signal strength,[%]');
    title(handles.axes2,'Tx signal strength');
    set(handles.plot3,'XData',timeElapsed,'YData',power);
    xlabel(handles.axes3, 'Time,[s]');
    ylabel(handles.axes3, 'TEG power,[W]');
    title(handles.axes3,'TEG power');
    timeElapsed = [timeElapsed time];

    end

end

function CloseRequestFcn(~, ~, ~)

    if exist('s', 'var')
        fclose(s)
        delete(s)
        clear s
    end
    usedtimer = timerfindall;
    if ~isempty(usedtimer)
        stop(usedtimer);
        delete(usedtimer);
        clear usedtimer;
    end
    delete(handles.fig)

    end

end
```

REFERENCES

[1] Patel, Divyesh; B.Mehta, Shruti and Shah, Pratik. (2015, March). Review of thermoelectricity to improve energy quality. *International Journal of Emerging Technologies and Innovative Research, 2*(3), 847-850.

[2] Molki, Arman. (2010). Simple demonstration of the Seebeck effect. *Science Education Review, 9*(3). 103-107.

[3] Dilhac, Jean-Marie and Bafleur, Marise. (2014, September 16). Energy harvesting in aeronautics for battery-free wireless sensor networks. *Aerospace and Electronic Systems Magazine, IEEE, 29*(8), pp. 18-22.

[4] Tan, Yen Kheng (ed). (2011, December). *Sustainable Energy Harvesting Technologies – Past, Present and Future*. Crotia: InTech.

[5] David Yan. ‘‘Modeling and Application of a Thermoelectric Generator’’. In: (2011).

[6] Scott Lee. ‘‘Thermoelectric Devices’’. College of Engineering, University of Hawai`i at Mānoa Honolulu, HI 96833. [Online].Available:http://www.spacegrant.hawaii.edu/reports/22_FA09-SP10/SLee_FA09.pdf

[7] The Fundamentals of Thermoelectrics. [Online].Available: http://www.nano.physik.uni-muenchen.de/education/praktika/f1_thermoelectrics.pdf

[8] Converting heat to electricity worldwide with TEG power. [Online].Available: http://www.tegpower.com/

[9] How do thermoelectrics work? [Online].Available: https://powerpractical.com/pages/how-do-thermoelectrics-work

[10] Seebeck effect. [Online].Available: http://searchnetworking.techtarget.com/definition/Seebeck-effect

[11] Seebeck Effect. [Online].Available:http://www.electrical4u.com/seebeck-effect-and-seebeck-coefficient/

[12] TEG Power Module Installation Notes. [Online].Available: http://tecteg.com/wp-content/uploads/2016/07/POWER-MODULE-INSTALLATION-NOTES-extended2.pdf

[13] Don Scansen. ‘‘Thermoelectric Energy Harvesting’’. In: (26-10-2011). [Online]. Available: http://www.digikey.com/en/articles/techzone/2011/oct/thermoelectric-energy-harvesting

[14] [Online].Available: http://www.bombayharbor.com/productImage/0795274001263995154/Power_Generation_Modules_Teg_12630_3_4.jpg

[15] Pololu 3.3V Step-Up Voltage Regulator U1V11F3 [online]. Available: https://www.pololu.com/product/2561 [Accessed July, 2015]

[16] Data sheet of Traco Power DC/DC converters [pdf]. Available: http://www.tracopower.com/products/tsr1.pdf [Accessed July, 2015]

[17] Data sheet of Zener diode "1N746A" [pdf]. Available: https://www.fairchildsemi.com/datasheets/1N/1N746A.pdf [Accessed December,2015]

[18] LTSPICE IV Getting Started Guide [pdf]. Available: http://cds.linear.com/docs/en/software-and-simulation/LTSPICEGettingStartedGuide.pdf [Accessed December, 2015]

[19] The Zener Diode Regulator [online]. Available: http://www.electronics-tutorials.ws/diode/diode_7.html [Accessed December, 2015]

[20] MATLAB: BUILDING A GUI MANUAL.

[21] Linear Technology LTC3105 Data sheet.

[22] Texas Instruments Basic calculation of a boost converter Power Stage.

[23] Muhammad H. Rashid Power Electronics: Circuits, Devices & Applications (4th Edition).

[24] Jerry Luecke, "Analog and Digital Circuits for Electronic Control System Applications: Using the TI MSP430 Microcontroller", Elsevier, Embedded Technology Series, 2005.

[25] Mike Mitchell, "Using Integrated Analog with the MSP430", ATC 2008 – MSP430 Advanced Technical Conference, Texas Instruments, Sonthofen, Germany, 2008.

[26] EZ430-RF2500 Development Tool User's Guide, SLAU227C, Texas Instruments, June 2008.