DESIGN, CONSTRUCTION AND PERFORMANCE EVALUATION OF A STIRLING CYCLE REFRIGERATION SYSTEM

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENT FOR THE DEGREE OF
B.Sc IN MECHANICAL ENGINEERING

SUBMITTED BY

Md. Atif Yasir	**201418047**
Fatema-Tuz-Jahara	**201418058**

SUPERVISED BY

Dr. Mohammad Mamun

Professor
Department of Mechanical Engineering
Bangladesh University of Engineering and Technology (BUET)

December 2017

STUDENT DECLARATION

We hereby declare that this report is based on results obtained from our own work. All the materials that were used for the purpose of completing this work are duly acknowledged and mentioned in reference. This report, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma.

SIGNATURE OF AUTHORS

Md. Atif Yasir
Student ID: 201418047

Fatema-Tuz-Jahara
Student ID: 201418058

SUPERVISOR CERTIFICATION

This is to certify that **Md. Atif Yasir**, Student no: 201418047, **Fatema-Tuz-Jahara**, Student no: 201418058 have completed their undergraduate thesis on "**DESIGN, CONSTRUCTION AND PERFORMANCE EVALUATION OF A STIRLING CYCLE REFRIGERATION SYSTEM**" under my supervision. To the best of my knowledge, the report is their original work and was not submitted elsewhere for any purpose.

APPROVED BY

Dr. Mohammad Mamun

Professor
Department of Mechanical Engineering
Bangladesh University of Engineering and Technology (BUET)

ABSTRACT

Concerns about the environmental impact of refrigerants used in vapour-compression heat pumps and refrigerators, have prompted the Stirling-Cycle Research Group at the Military Institute of Science and Technology to investigate the feasibility of low-cost Stirling-cycle machines that use air as the refrigerant. Such machines theoretically have the highest efficiency possible for any practical thermodynamic system, and thus provide a tempting alternative to traditional vapour compression technology. This paper outlines the working principles of Stirling-cycle heat-pumps and refrigerators, and describes some of the work performed at Military Institute of Science and Technology. Some of the heat-pump development program results are also presented, and briefly discussed.

ACKNOWLEDGEMENT

We would like to sincerely thank our supervisor Dr. Mohammad Mamun who guided us to find a subject for the thesis and gave us a perfect description of the subject, which was a great help later finding our way to do our thesis. Again thanks to Dr. Mohammad Mamun for giving his valuable time and observing the process of the study and giving us directions and recommendations. We are also grateful to Brig Gen Md. Habibur Rahman, the Head of the department of Mechanical Engineering who gave us opportunity to do our thesis at Military Institute of Science and Technology Heat Engine Laboratory. Thanks to the staff, teachers and assistants who were at the university and they helped us a lot. All our acknowledgments and thanks are expressed to our teachers, classmates and our families who were supportive and friendly and encouraged us in our studies.

NOMENCLATURE

Symbol	**Definition**
Ta	Ambient temperature
T_L	Lower temperature
$\dot{Q}a$	Heat given off to the surroundings at ambient temperature
$\dot{Q}_L$	Heat taken from the surroundings at lower temperature
COP	Coefficient of performance
Q	Useful heat supplied or removed by the considered system
W	Work required by the considered system
Q_C	Heat removed from the cold reservoir
Q_H	Heat supplied to the hot reservoir
Q_{cold}	Heat collected from the cold reservoir
Q_{hot}	Heat transferred to the hot reservoir
T_{cold}	Temperature of the hot reservoir
T_{hot}	Temperature of the cold reservoir
$COP_{cooling}$	Coefficient of performance of cooling
$COP_{heating}$	Coefficient of performance of heating

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Background of the study

The Stirling Cycle Research Group at the Military Institute of Science and Technology has been conducting research on stirling-cycle machinery for almost a year. The aim of the group is to do fundamental research into stirling-cycle thermodynamics, but also to use the results of this research to develop practical machinery at a prototype level, with a view to later commercial product development.

Initial work has largely focused on stirling-cycle refrigerators, and has resulted in the production this prototype system.

1.2 Motivation

The prime motivation behind this research is the simple to understand yet highly efficient stirling cycle itself. The way it closely resembles the basic Carnot cycle and how easily and effectively one can gather all the ideas behind a basic thermodynamic cycle are some of the reasons why this group is largely motivated to spend some time behind this study.

There is also the fact that a Stirling engine will work as a refrigeration unit or a heat pump if the engine is run using an external power source and a well-built stirling cooler can even compete with the existing vapor compression refrigeration system. This also motivated the group to build a prototype system to understand the basic principles clearly.

1.3 Objectives

The objectives behind this study are to:

1) Design a stirling cycle refrigeration system
2) Build a stirling cycle refrigeration system
3) Understand the stirling cycle refrigeration system
4) Gather experimental data
5) Comprehend the applications of this system

Chapter 2: Literature Review

2.1 History

To understand the Stirling Refrigeration, the history of Stirling refrigeration cycle as well as Stirling engine are essential.

“All my improvements for diminishing the consumption of fuel, consist of the differing forms or modification of a new method, contrivance, or mechanical arrangements for heating and cooling liquids, airs or gases, and other bodies, by the use of which contrivance heat is abstracted from one portion of such liquids, airs or gases, and other bodies, and communicated with another portion with very little loss, so that in all cases where a constant succession of heated liquids or other bodies is required, the quantity of fuel necessary to maintain or supply it is by this contrivance greatly diminished” - **Robert Stirling**

Stirling Engine History

Rev'd Robert Stirling applied for the first of his patent for an engine later known as Stirling Engine and the 'Economiser' in 1816, a few months after being appointed as a minister in the Church of Scotland at age 25. Others such as Sir. George Caley had devised air engines previous to this time (c. 1807) and other devices called air engines were known as early as 1699. The 'Economiser', or regenerator, has come to be recognized as a most important portion of the patent of 1816.

Some historians have indicated that the reason for Rev'd Stirling's efforts at such a device were driven by his concern for the working people of his parishes as steam engines were being used extensively in that area and time period. Due to the lack of strength in the materials available to construct boilers ('Bessemer Iron', or Steel, was not yet available), they would frequently explode with devastating results on the people working nearby. The effects of high pressure steam on the human body are quite awful as anyone who has experienced a steam burn in the kitchen can attest.

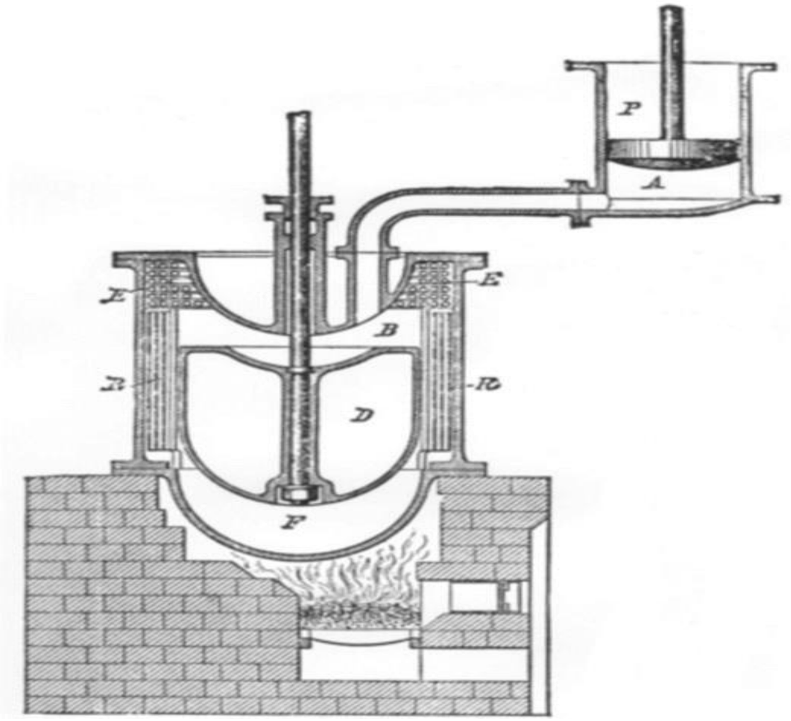


Fig 1.1: Robert Stirling’s engine along with economiser

So Rev'd Stirling invented a safer (and more efficient) replacement for the steam engine, in order to save lives and improve the conditions of his parish life. Stirling's engine would not explode because the pressures were not elevated to that level. The machine simply stopped if the heater section failed.

The best recorded implementation of these efforts was at the Dundee Foundry Company where Robert's brother, James Stirling, was employed. James was a very good engineer and a driving force in the implementation and perfection of the Stirling invention. A very large double-acting-piston machine with not one but two heater/displacer sections was built at the foundry under his direction (and we presume design). This engine powered the foundry for some years before material failures at inopportune times caused it to be replaced by a steam engine.

With Bessemer's discovery of a process to mass produce quality steel, steam engines became more powerful and much safer to operate, and so the Stirling engine nearly faded into obscurity.[1]

Stirling Refrigeration begins

In 1834, a noted British astronomer John Herschel applied the Stirling cycle for cooling. This was the first known case of using the Stirling machine for refrigeration purposes.

Scottish born John Gorrie immigrated to US in 1849; he lived in South Carolina and Florida. Gorrie may have been the first to apply the Stirling machine for making ice. From descriptions published in 1876 by Alexander Carnegie Kirk, it seems that by then Stirling cycle cooling was well known in technical circles.

Advancing Stirling Technology

Over time, advancements in Stirling cycle machines become less frequent and theses machines almost disappeared by 1900 until they were rediscovered in the 1940s by researchers at Philips in The Netherlands. The rebirth of the Stirling engines was due almost entirely to workers at Philips Research Laboratory in Eindhoven. Work on small Stirling engines started there in the mid-1930s. The objective was to provide small, quiet, thermally activated, electric-power generator for radios in areas of the world without regular power supplies. It is said the choice between steam and hot-air engines was made at the visit to the Museum of Technology in Paris by one of the technical directors of the laboratory where he saw some of the old hot-air engines displayed. He believed, rightly, that modern materials and technology could elevate the hot-air engine to a performance undreamt of in earlier engines. Philips used pressurization to significantly improve power density.

In 1946, under the direction of J. W. L. Köhler Philips applied the Stirling cycle for deep temperature use in the generation of liquefied gases (US Patent 2,907,175, March 14, 1955).

Invention of the Free-Piston Stirling Engine

Various free-piston Stirling engine configurations were invented and developed in the 1960s. These machines required no mechanical linkages since the moving parts are either driven by internal gas pressure or a linear motor/alternator. Later developments included gas bearings where both the piston and displacer are both supported on helium bearings, thus eliminating the need for any lubrication whatsoever.

Around that time, Philips was seeking to expand sales of its radios into parts of the world where grid electricity and batteries were not consistently available. Philips' management decided that offering a low-power portable generator would facilitate such sales and asked a group of engineers at the company's research lab in Eindhoven to evaluate alternative ways of achieving this aim. After a systematic comparison of various prime movers, the team decided to go forward with the Stirling engine, citing its quiet operation (both audibly and in terms of radio interference) and ability to run on a variety of heat sources (common lamp oil – "cheap and available everywhere" – was favored). They were also aware that, unlike steam and internal combustion engines, virtually no serious development work had been carried out on the Stirling engine for many years and asserted that modern materials and know-how should enable great improvements.[2]

Philips MP1002CA Stirling generator of 1951

By 1951, the 180/200 W generator set designated MP1002CA (known as the "Bungalow set") was ready for production and an initial batch of 250 was planned, but soon it became clear that they could not be made at a competitive price. Additionally, the advent of transistor radios and their much lower power requirements meant that the original rationale for the set was disappearing. Approximately 150 of these sets were eventually produced. Some found their way into university and college engineering departments around the world giving generations of students a valuable introduction to the Stirling engine.

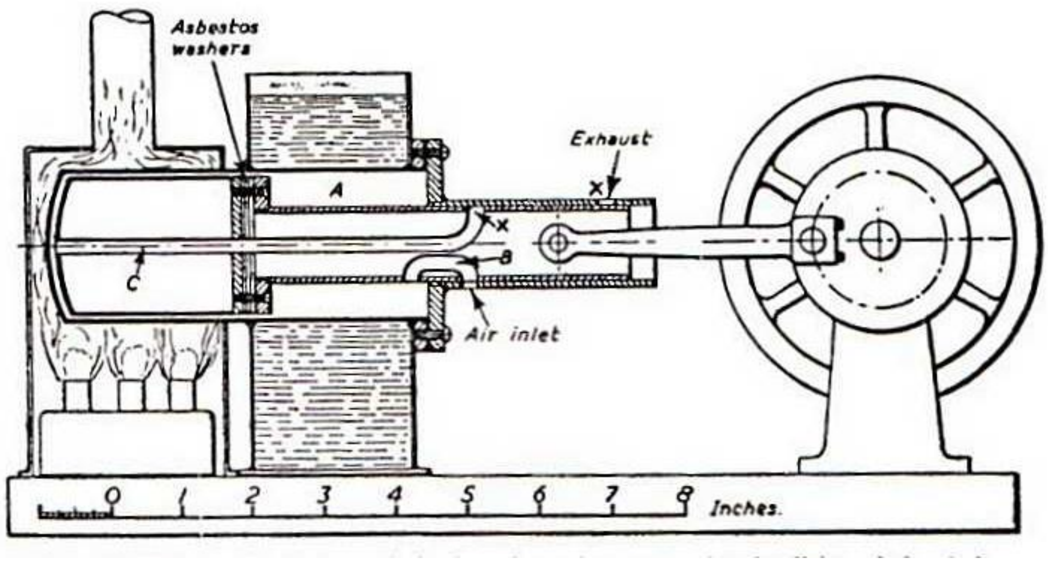


Fig 1.2: Sectional View of Simple Hot Air Engine

In parallel with the Bungalow set, Philips developed experimental Stirling engines for a wide variety of applications and continued to work in the field until the late 1970s, but only achieved commercial success with the "reversed Stirling engine" cryocooler. However, they filed a large number of patents and amassed a wealth of information, which they licensed to other companies and which formed the basis of much of the development work in the modern era.

In 1996, the Swedish navy commissioned three Gotland-class submarines. On the surface, these boats are propelled by marine diesel engines. However, when submerged, they use a Stirling-driven generator developed by Swedish shipbuilder Kockums to recharge batteries and provide electrical power for propulsion. A supply of liquid oxygen is carried to support burning of diesel fuel to power the engine. Stirling engines are also fitted to the Swedish Södermanland-class submarines, the Archer-class submarines in service in Singapore and, license-built by Kawasaki Heavy Industries for the Japanese Sōryū-class submarines. In a submarine application, the Stirling engine offers the advantage of being exceptionally quiet when running. Stirling engines are frequently used in the dish version of Concentrated Solar Power systems. A mirrored dish similar to a very large satellite dish directs and concentrates sunlight onto a thermal receiver, which absorbs and collects the heat and using a fluid transfers it into the Stirling engine. The resulting mechanical power is then used to run a generator or alternator to produce electricity.

Stirling engines are forming the core component of micro combined heat and power (CHP) units, as they are more efficient and safer than a comparable steam engine. CHP units are being installed in people's homes.

For over 50 years, DH Industries and its predecessors (a.o. Philips Cryogenics) have been designing and supplying these high efficient, extreme low temperature coolers. Over these years more than 5,000 units have been supplied worldwide, in a wide range of demanding circumstances and different applications.

Stirling Cryogenerators

These Stirling Cryogenerators provide cooling power in the range of 50 - 6,000 Watt in a temperature range of 15 - 150 Kelvin (- 258 to -123°C or -433 to - 190°F) and come in either a 1 or 4-cylinder unit and in a one-stage (down to 40K) or two-stage configuration (down to 15K). Stirling Cryogenics Cryocoolers can be integrated into any kind of (cryogenic) system or used as stand-alone units and can be used as liquefiers (Nitrogen, Oxygen, Methane, Argon, Neon, biogas etc.) or as (Helium) gas cooler.

The product range is set up in such a way that any capacity (by using multiple units) and application (by custom specific add-ons) can be met. This way the most efficient, reliable and user friendly cryogenic cooling is achieved.

It was used in the so-called hot air engine, which was considered at the time to be capable of replacing the steam engine. This was partly because the boilers used in early steam engines were prone to explosion. The counterpart of the hot air motor, the refrigerator, was first recognized in 1832. Both machines experienced high and low points during the nineteenth century.

The principle behind the machines was almost condemned to obscurity after the invention of the internal combustion engine (gas-, petrol-, and diesel motors) and compressor refrigerators with external evaporation.

In 1938 Philips Research Laboratories was looking for a means of generating electricity to power radios in remote areas where there was no electricity supply. The practically-forgotten hot air motor attracted attention. In 1946 Philips started studying the cooling techniques used in the Stirling cycle. The result was the development of the cold gas-refrigerator.

This machine, the cryogenerator, marked the start of significant cryogenic activities at Philips. So even though the Stirling hot air motor never became a commercial success, the Stirling cryogenerator is incorporated in equipment used from Antarctica to the North Pole.

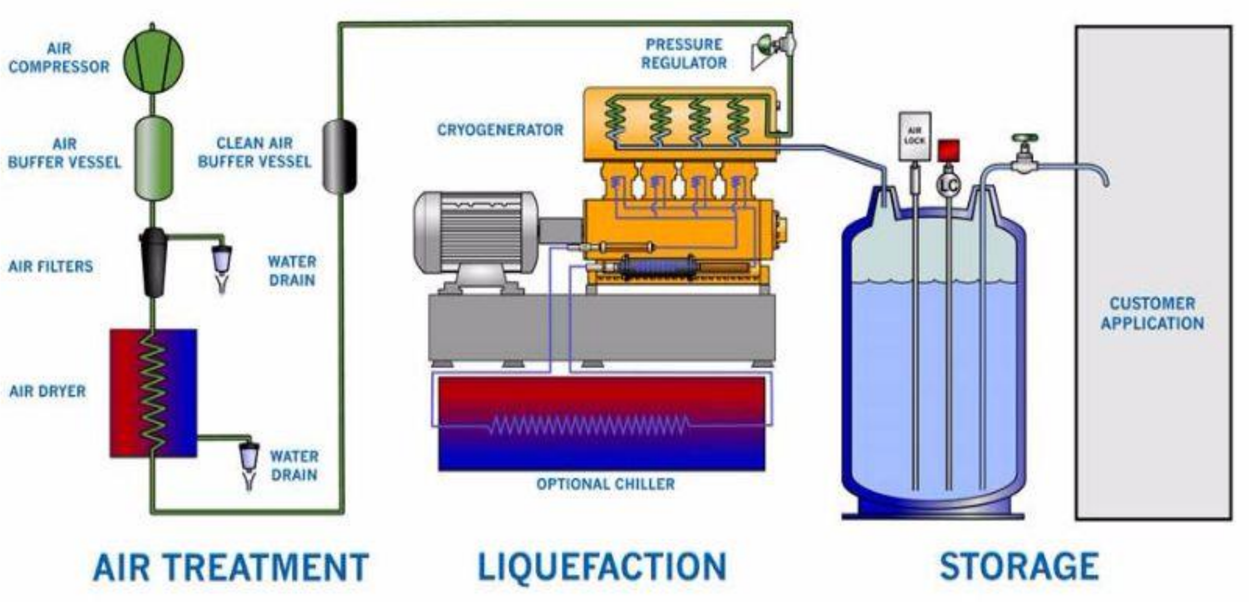


Fig 1.3: Stirling Cryogenerator

In 1990, Philips’ cooling-related activities became independent and eventually continued under the name of Stirling Cryogenics BV. Thanks to continual innovation and considerable investment in R&D, the Stirling cryogenerator is now used in advanced technological machinery for cooling gases and liquids to extremely low temperatures (200 K to 20 K).

Lastly nowadays applications with Stirling cryogenerators are used in a wide range of applications, including the production of liquid gases, cooling gases and liquids, and cooling during (industrial) processes.[3]

2.2 Types of Stirling Cycle Refrigeration System

Generally, Stirling Cycle Refrigeration Systems can have 4 configurations.

1) Alpha configuration
2) Beta configuration
3) Split-pair configuration
4) Free piston configuration

1) Alpha configuration

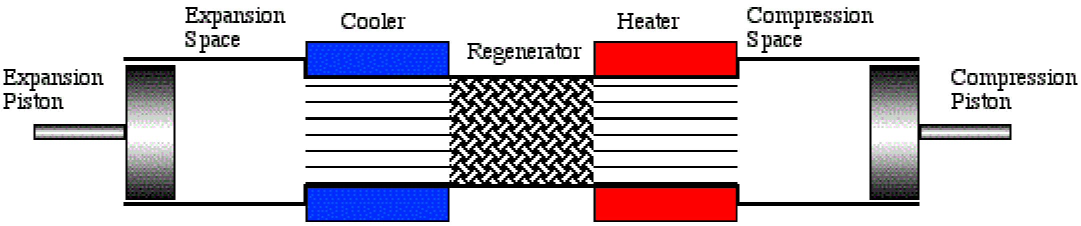


Fig 2.1: Alpha stirling refrigeration system

In this configuration, two separate cylinders are used. One for cooling side and one for the heating side. Both cylinders are equipped with pistons. The crank angle is 90° between the compression piston and the expansion piston.

2) Beta configuration

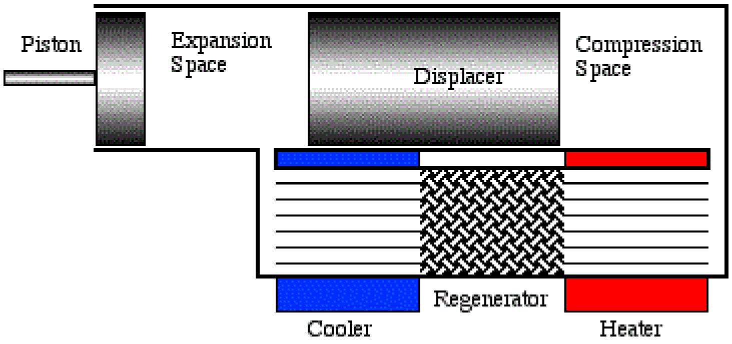


Fig 2.2: Beta stirling refrigeration system

In this configuration only one piston is used. The displacer moves forward and backward displacing the working fluid from one side to another. The crank angle is 90° between the piston and the displacer.

3) Split-pair configuration

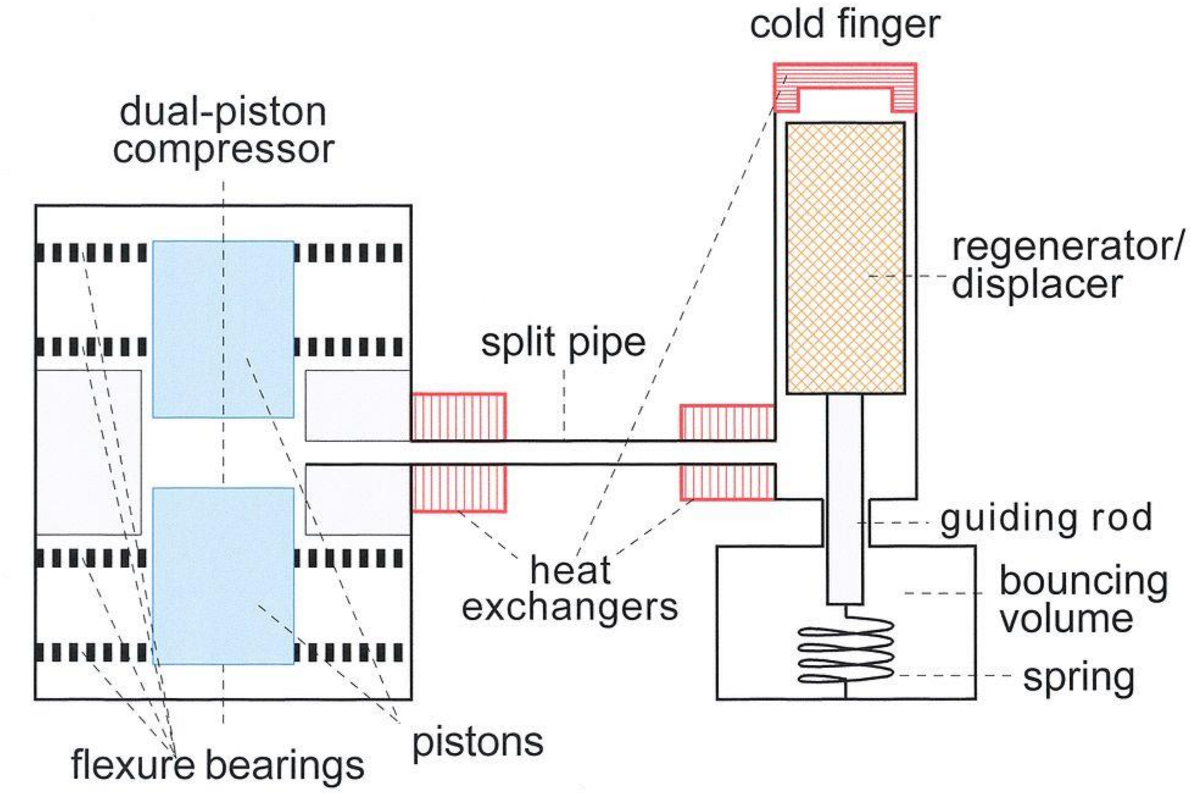


Fig 2.3: Split-pair stirling refrigeration system

Another type of Stirling cooler is the split-pair type, consisting of a compressor, a split pipe, and a cold finger. Usually there are two pistons moving in opposite directions driven by AC magnetic fields (as in loudspeakers). The pistons can be suspended by so-called flexure bearings. They provide stiffness in the radial direction and flexibility in the axial direction. The pistons and the compressor casing don't touch so no lubricants are needed and there is no wear. The regenerator in the cold finger is suspended by a spring. The cooler operates at a frequency near the resonance frequency of the mass-spring system of the cold finger.

4) Free piston configuration

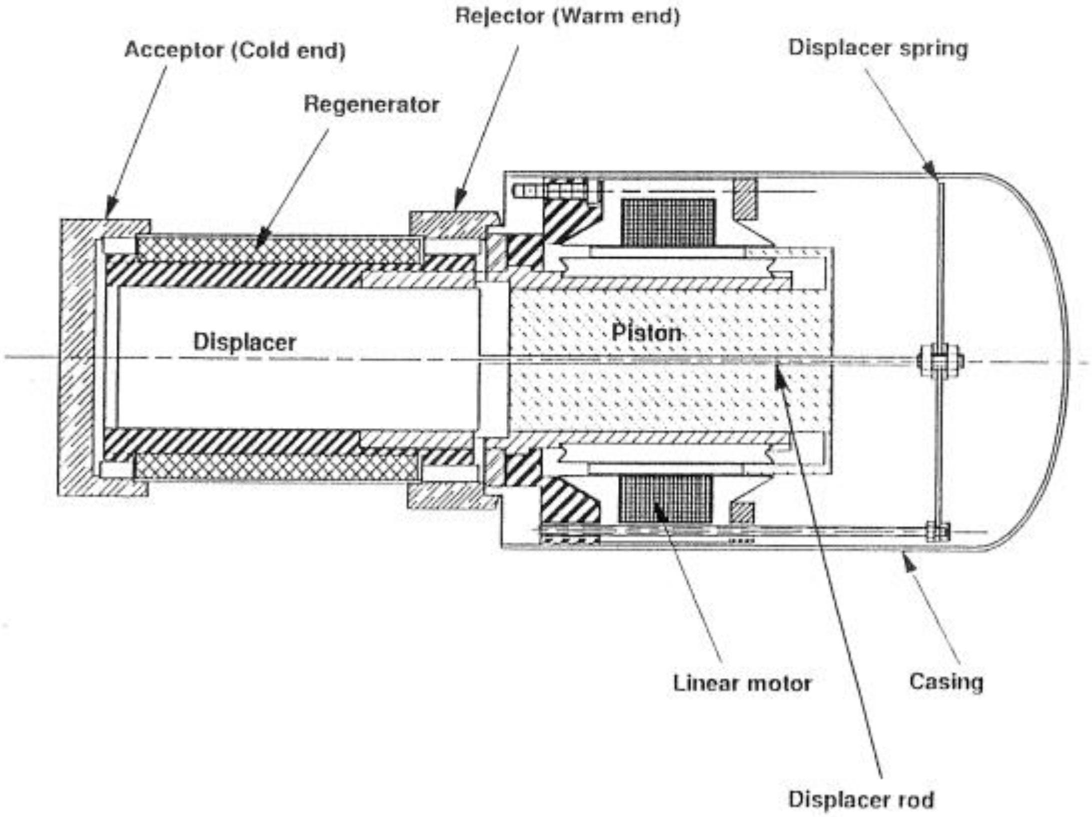


Fig 2.4: Free piston stirling refrigeration system

This stirling cooler is driven by a linear motor which is sealed within the pressure vessel of the cooler. The cooling system contains an acceptor and rejector (the internal heat exchangers of the stirling cooler), refrigerant (helium), a piston and displacer, and a regenerator. When the free-piston stirling cooler is operational, the acceptor will take up thermal energy from the refrigerator and the rejector will emit this energy to the environment. To transport thermal energy from the acceptor to the rejector about one gram of helium is shuttled between both ends inside the cooler. At the cold end the gas is expanded taking up thermal energy from the acceptor (which drops in temperature as a result) and at the warm end the cooling gas is compressed and releases thermal energy to the rejector.[4]

2.3 Basic Principles of Stirling Cycle Refrigeration System

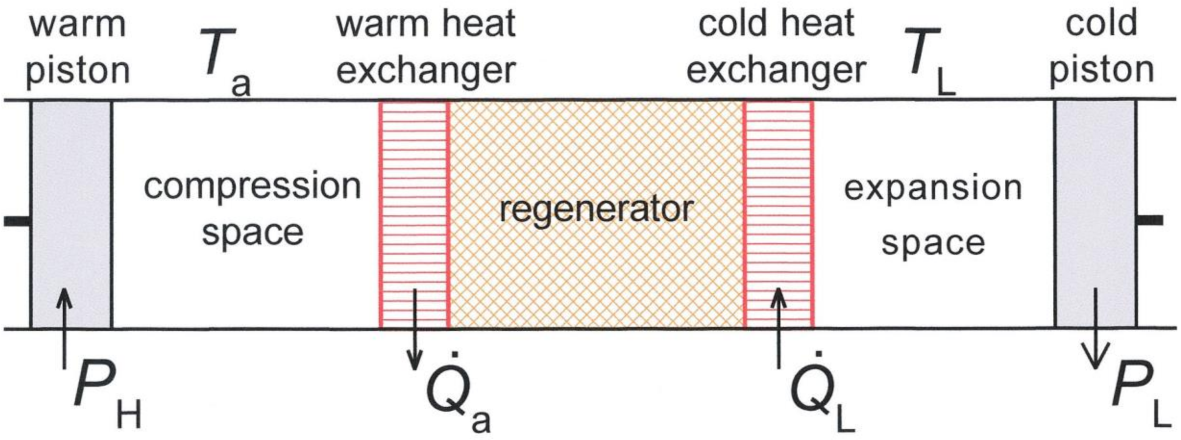


Fig 2.5: Schematic diagram of a Stirling cooler

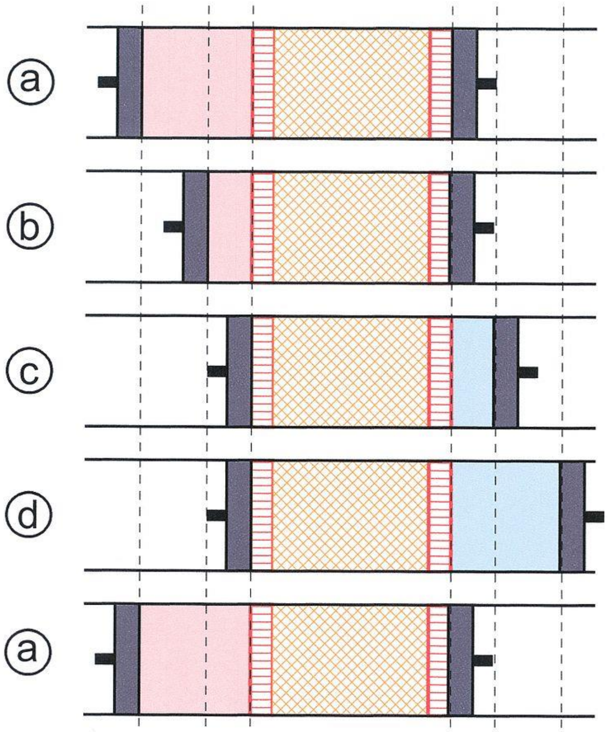


Fig 2.6: Four states in the Stirling cycle.

The basic type of Stirling-type cooler is depicted in fig 2.5. From left to right it consists of a piston, a compression space, and heat exchanger (all at ambient temperature Ta), a regenerator, and a heat exchanger, expansion space, and a piston (all at the low temperature T_L). Left and right the thermal contact with the surroundings at the temperatures Ta and T_L is supposed to be perfect so that the compression and expansion are isothermal. The work, performed during the expansion, is used to reduce the total input power. Usually helium is the working fluid.

The cooling cycle is split in 4 steps as depicted in fig 2.6. The cycle starts when the two pistons are in their most left positions:

From a to b: The warm piston moves to the right while the cold piston is fixed. The compression at the hot end is isothermal (by definition), so heat Qa is given off to the surroundings at ambient temperature Ta.

From b to c: The two pistons move to the right. The volume between the two pistons is kept constant. The hot gas enters the regenerator with temperature Ta and leaves it with temperature T_L. The gas gives off heat to the regenerator material.

From c to d: The cold piston moves to the right while the warm piston is fixed. The expansion is isothermal and heat Q_L is taken up. This is the useful cooling power.

From d to a: The two pistons move to the left while the total volume remains constant. The gas enters the regenerator with low temperature T_L and leaves it with high temperature Ta so heat is taken up from the regenerator material. At the end of this step the state of the cooler is the same as in the beginning.

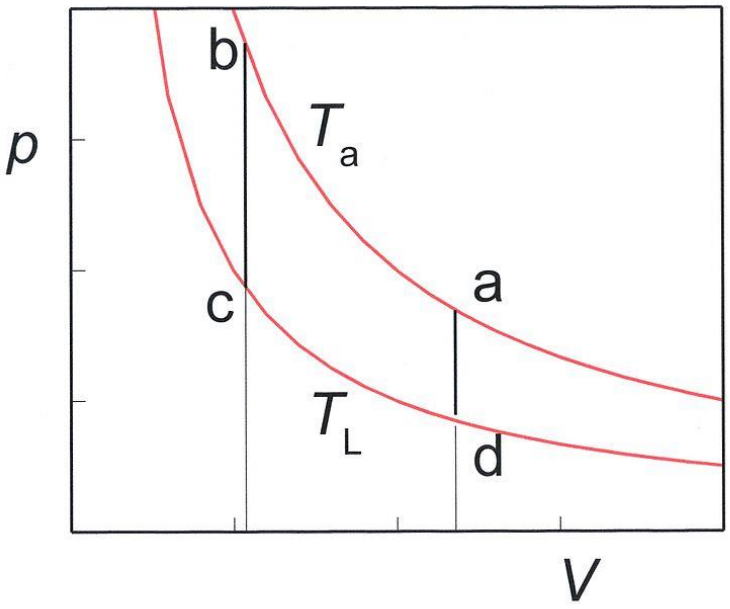


Fig 2.7: pV diagram

In the pV diagram (fig. 2.7) the corresponding cycle consists of two isotherms and two isochores. The volume V is the volume between the two pistons. In practice the cycle is not divided in discrete steps as described above. Usually the motions of both pistons are driven by a common rotary axes which makes the motions harmonic. The phase difference between the motions of the two pistons is about 90°.[2] [9]

2.4 Coefficient of Performance

The coefficient of performance or COP (sometimes CP or CoP) of a heat pump, refrigerator or air conditioning system is a ratio of useful heating or cooling provided to work required. Higher COPs equate to lower operating costs. The COP usually exceeds 1, especially in heat pumps, because, instead of just converting work to heat (which, if 100% efficient, would be a COP of 1), it pumps additional heat from a heat source to where the heat is required. For complete systems, COP calculations should include energy consumption of all power consuming auxiliaries. COP is highly dependent on operating conditions, especially absolute temperature and relative temperature between sink and system, and is often graphed or averaged against expected conditions.

The equation is:

$$COP = \frac{Q}{W}$$

Where
Q is the useful heat supplied or removed by the considered system.
W is the work required by the considered system.

The COP for heating and cooling are thus different, because the heat reservoir of interest is different. When one is interested in how well a machine cools, the COP is the ratio of the heat removed from the cold reservoir to input work. However, for heating, the COP is the ratio of the heat removed from the cold reservoir plus the input work to the input work:

$$COP_{heating} = \frac{|Q_H|}{W} = \frac{|Q_C| + W}{W}$$
$$COP_{cooling} = \frac{|Q_C|}{W}$$

Where
Qc is the heat removed from the cold reservoir.
Q_H is the heat supplied to the hot reservoir.

According to the first law of thermodynamics, in a reversible system we can show that:

$$Q_{hot} = Q_{cold} + W$$

$$W = Q_{hot} - Q_{cold}$$

Where

Q$_{\mathbf{hot}}$ is the heat transferred to the hot reservoir

Q$_{\mathbf{cold}}$ is the heat collected from the cold reservoir

Therefore, by substituting for W,

$$COP_{heating} = \frac{Q_{hot}}{Q_{hot} - Q_{cold}}$$

For a heat pump operating at maximum theoretical efficiency (i.e. Carnot efficiency), it can be shown that

$$\frac{Q_{hot}}{T_{hot}} = \frac{Q_{cold}}{T_{cold}}$$

$$Q_{cold} = \frac{Q_{hot}T_{cold}}{T_{hot}}$$

Where

T$_{\mathbf{hot}}$ is the temperature of the hot reservoir

T$_{\mathbf{cold}}$ is the temperature of the cold reservoir

Note: these equations must use an absolute temperature scale, for example, **Kelvin**

At maximum theoretical efficiency,

$$COP_{heating} = \frac{T_{hot}}{T_{hot} - T_{cold}}$$

Which is equal to the reciprocal of the ideal efficiency for a heat engine, because a heat pump is a heat engine operating in reverse. Similarly,

$$COP_{cooling} = \frac{Q_{cold}}{Q_{hot} - Q_{cold}} = \frac{T_{cold}}{T_{hot} - T_{cold}}$$

Note: the COP of a heat pump depends on its duty. The heat rejected to the hot sink is greater than the heat absorbed from the cold source, so the heating COP is 1 greater than the cooling COP.

COP(heating) applies to heat pumps and **COP(cooling)** applies to air conditioners of refrigerators.[2] [8]

2.5 Deviation from the ideal cycle

Unfortunately, practical Stirling-cycle heat pumps and refrigerators differ from the ideal cycle in several important aspects:

(i) The regenerator and heat-exchangers in practical machines have non-zero volume. This means that the working gas is never completely in either the hot or cold end of the machine, and therefore never at a uniform temperature.

(ii) The piston motion is usually semi-sinusoidal rather than discontinuous, leading to nonoptimal manipulation of the working gas.

(iii) The expansion and compression processes in practical Stirling-cycle machines are polytropic rather than isothermal. This causes temperature fluctuations in the working gas and leads to adiabatic and transient heat transfer losses.

(iv) Fluid friction losses occur during gas displacement, particularly due to flow through the regenerator.

(v) Other factors such as heat conduction between the hot and cold ends of the machine, seal leakage and friction, appendix gap effects, and friction in kinematic mechanisms all cause real Stirling-cycle machines to differ from ideal behaviour. The combination of these factors means that the Stirling Cycle differs from its ideal embodiment more than any other thermodynamic cycle; making it extremely difficult to develop an accurate mathematical model to predict system performance. On the other hand, the vast number of critical parameters that can be varied on a practical machine mean that experimental optimization is enormously time-consuming (and expensive).

Chapter 3: Theoretical Background

3.1 Working fluid for Stirling Cycle Refrigeration System

The gas used should have a low heat capacity, so that a given amount of increase in pressure leads to a large amount of heat transferred. Considering this issue, helium would be the best gas because of its very low heat capacity. Air is a viable working fluid, but the oxygen in a highly pressurized air engine can cause fatal accidents caused by lubricating oil explosions. Following one such accident Philips pioneered the use of other gases to avoid such risk of explosions.

3.2 Hydrogen as working fluid

Hydrogen's low viscosity and high thermal conductivity make it the most powerful working gas, primarily because the cooler can run faster than with other gases. However, because of hydrogen absorption, and given the high diffusion rate associated with this low molecular weight gas, particularly at high temperatures, H_2 leaks through the solid metal of the heater. Diffusion through carbon steel is too high to be practical, but may be acceptably low for metals such as aluminum, or even stainless steel. Certain ceramics also greatly reduce diffusion. Hermetic pressure vessel seals are necessary to maintain pressure inside the engine without replacement of lost gas. For high temperature differential (HTD) coolers, auxiliary systems may need to be added to maintain high pressure working fluid. These systems can be a gas storage bottle or a gas generator. Hydrogen can be generated by electrolysis of water, the action of steam on red hot carbon-based fuel, by gasification of hydrocarbon fuel, or by the reaction of acid on metal. Hydrogen can also cause the embrittlement of metals. Hydrogen is a flammable gas, which is a safety concern if released from the engine.[10]

3.3 Helium as working fluid

Most technically advanced Stirling coolers, like those developed for United States government labs, use helium as the working gas, because it functions close to the efficiency and power density of hydrogen with fewer of the material containment issues. Helium is inert, and hence not flammable. Helium is relatively expensive, and must be supplied as bottled gas. Researcher Alan Organ demonstrated that a well-designed air cooler is theoretically just as efficient as a helium or hydrogen cooler, but helium and hydrogen coolers are several times more powerful per unit volume.[10]

3.4 Air or Nitrogen as working fluid

Some coolers use air or nitrogen as the working fluid. These gases have much lower power density, but they are more convenient to use and they minimize the problems of gas containment and supply (which decreases costs). The use of compressed air in contact with flammable materials or substances such as lubricating oil introduces an explosion hazard, because compressed air contains a high partial pressure of oxygen. However, oxygen can be removed from air through an oxidation reaction or bottled nitrogen can be used, which is nearly inert and very safe.[10]

3.5 Other working fluids

Other possible lighter-than-air gases include: methane, and ammonia.[10]

3.6 Advantages of Stirling Cycle Refrigeration System

1) The ecological aptitude to respond to the environmental requirements on air pollution.
2) Reliability and easy maintenance:
   The technological simplicity makes it possible to have engines with a very great reliability and requiring little maintenance.
3) The very diverse uses because of its autonomy and adaptability to the needs. Stirling cooling systems operate with very low noise emission.
4) Absorption type refrigerators use lot of energy while compared to stirling refrigeration.
5) Hydrogen/Helium is the working fluid which is environmentally friendly compared to CFCs, it can replace as an alternative.
6) With regenerator, COP approaches carnot COP.
7) Components used in stirling refrigeration can be easily reusable and recyclable.[5] [6]

3.7 Advantages of Stirling Cycle Refrigeration System over Vapor Compression refrigeration cycle

As in the case of comparing between Stirling Cycle Refrigeration and Vapour Compression Refrigeration System certain important criteria have been entertained, such as, Coefficient of Performance, Costing, Refrigerant availability and many more.

In case of Stirling cycle refrigeration, no phase change happens in the working fluid to transfer heat or to refrigerate. The working fluid is contained within stirling cooler (closed).
But for Vapor Compression Cycle, the refrigerant is required to change phase from gas to liquid and the refrigerant circulates throughout the system which is open.
As for large capacity range, Stirling Refrigeration cycle tends to have constant high efficiency, unlike Vapor Compression Cycle. In Vapor Compression Cycle, efficiency decreases for small capacities.

The noise created by Stirling cycle refrigeration is very low whereas the noise produced by Vapor Compression Cycle refrigeration is quite high. The Refrigerant used in Vapor Compression Cycle refrigeration are mostly moderate HFCs which is not environment friendly. But the entire Stirling cycle refrigeration is green, safe and environment friendly.[5] [6]

3.8 Disadvantages of Stirling Cycle Refrigeration System

1) The price: Its cost is probably the most important problem, it is not yet competitive with other means well established. A generalization of its employment should solve this problem inherent in any novelty.
2) The ignorance of this type of cooler by the general public. Only a few people know it exists. It is therefore necessary to promote it.
3) The variety of models prevents standardization and consequently, lower prices.
4) The problems of sealing are difficult to solve as soon as one wishes to have high pressures of operation. The choice of “ideal” gas would be hydrogen for its lightness and its capacity to absorb the calories, but its ability to diffuse through materials is a great disadvantage.
5) Heat transfers with a gas are delicate and often require bulky apparatuses.
6) The lack of flexibility: The fast and effective variations of cooling are difficult to obtain with a stirling cycle refrigeration.[5] [6]

Chapter 4: Design Consideration

4.1 Introduction

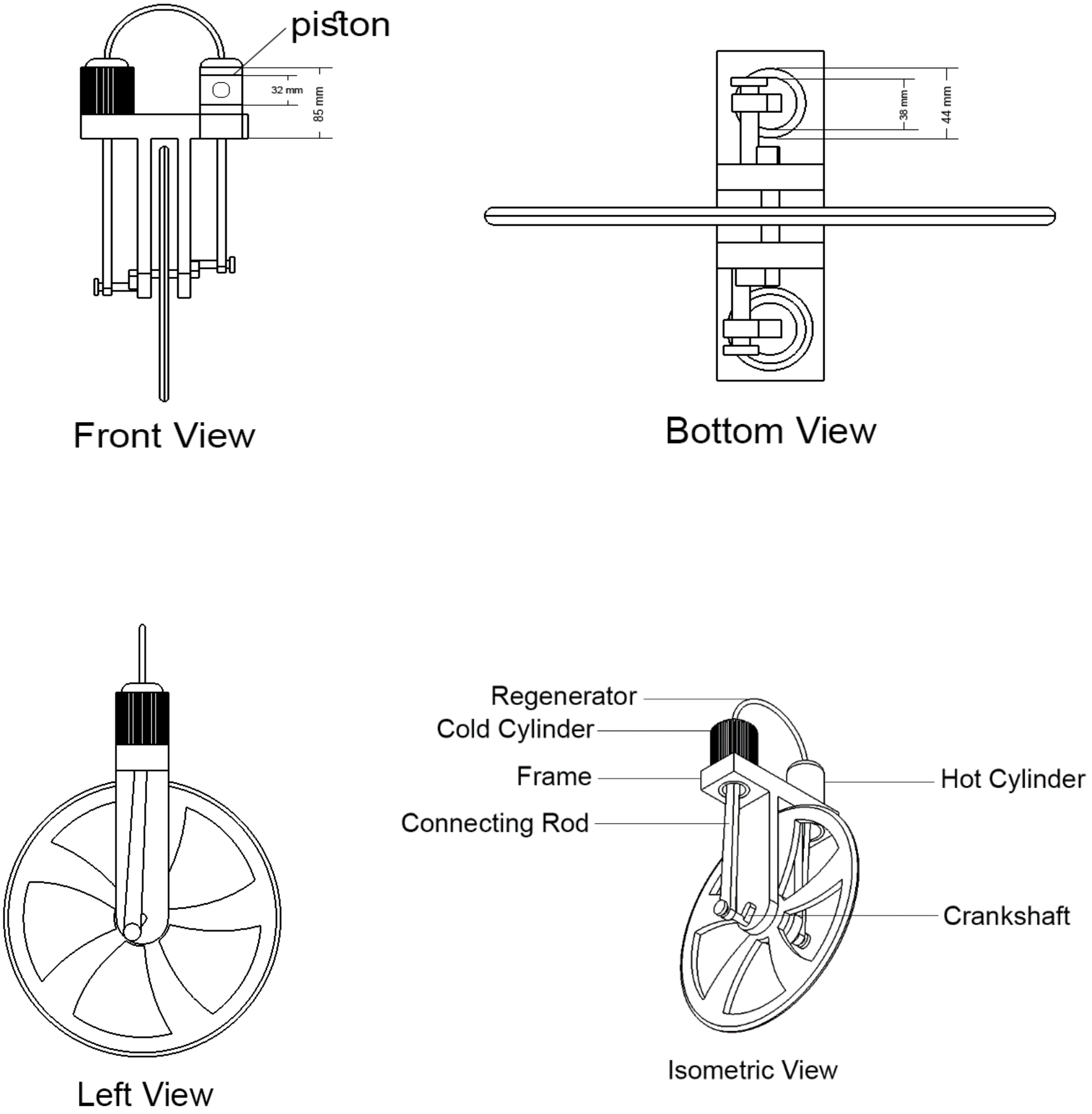


Fig 4.1: Solidworks render of the system

The basic structure was initially designed in Solidworks, a well-known CAD (Computer Aided Drawing) Software.

4.2 Factors affecting the system

The Factors influence the efficiency of the entire system are the design consideration of Cylinder, Piston, Regenerator and the Power supply.

4.2.1 Design of the Cylinders

Two cylinders are required for the system, one is hot side cylinder and the other one is cold side cylinder. Cylinders must allow good heat transfer through their surfaces. Cylinders are generally equipped with fins or heat exchangers to increase the surface area and provide more transfer of heat. The cylinders must be strong enough to withstand the continuous reciprocating motions of the pistons. It must be corrosive resistant. Typical materials for making the cylinders include cast iron.

4.2.2 Cylinder size

To construct the experimental setup, a particular dimension of cylinder has been determined.
The height of cylinder is considered to be in between 80mm-100mm. The outer diameter of the cylinder would be within the range of 40mm- 50mm.
Thickness of the cylinder is 3mm-4mm.

4.2.3 Design of the Pistons and crankshafts

A piston is a component of reciprocating engines, reciprocating pumps, gas compressors and pneumatic cylinders, among other similar mechanisms. It is the moving component that is contained by a cylinder and is made gas-tight by piston rings. In an engine, its purpose is to transfer force from expanding gas in the cylinder to the crankshaft via a piston rod and/or connecting rod. In a pump, the function is reversed and force is transferred from the crankshaft to the piston for the purpose of compressing or ejecting the fluid in the cylinder.

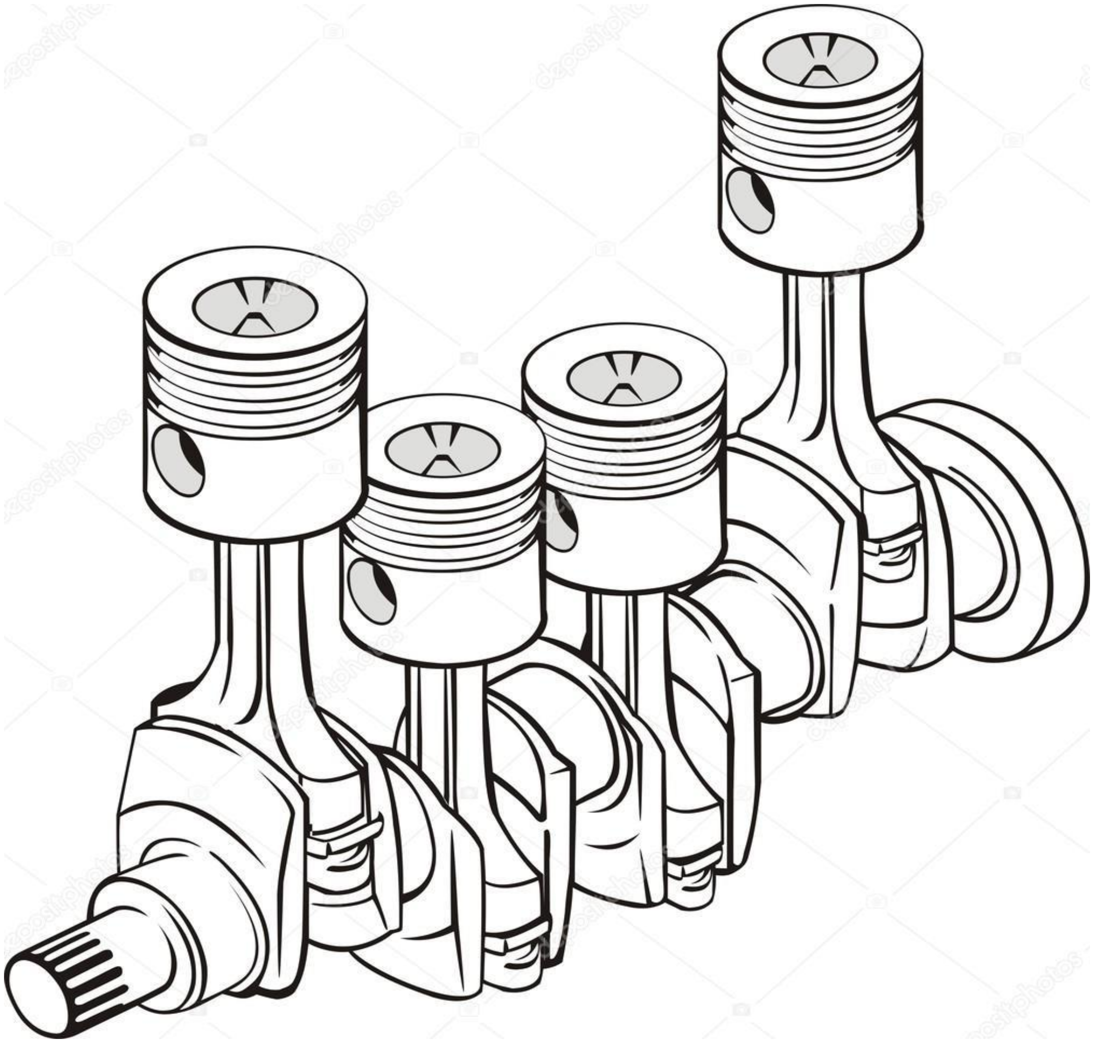


Fig 4.2: Piston and Crankshaft

A crankshaft—related to crank—is a mechanical part able to perform a conversion between reciprocating motion and rotational motion. In a reciprocating engine, it translates reciprocating motion of the piston into rotational motion; whereas in a reciprocating compressor, it converts the rotational motion into reciprocating motion. In order to do the conversion between two motions, the crankshaft has "crank throws" or "crankpins", additional bearing surfaces whose axis is offset from that of the crank, to which the "big ends" of the connecting rods from each cylinder attach.

4.2.4 Design of Energy supplying system to the Pistons

The energy supplied to the system is done by electric motor using direct shaft coupling or shaft coupling via belt drive. These systems vary from design to design and product to product. The energy is mainly supplied with AC current through electric motor.

An electric motor is an electrical machine that converts electrical energy into mechanical energy.

The reverse of this is the conversion of mechanical energy into electrical energy and is done by an electric generator, which has much in common with a motor.

Most electric motors operate through the interaction between an electric motor's magnetic field and winding currents to generate force. In certain applications, such as in regenerative braking with traction motors in the transportation industry, electric motors can also be used in reverse as generators to convert mechanical energy into electric power.

4.2.5 Design of Regenerator

In a Stirling cycle refrigeration, the regenerator is an internal heat exchanger and temporary heat store placed between the hot and cold spaces such that the working fluid passes through it first in one direction then the other, taking heat from the fluid in one direction, and returning it in the other. It can be as simple as metal mesh or foam, and benefits from high surface area, high heat capacity, low conductivity and low flow friction. Its function is to retain within the system that heat that would otherwise be exchanged with the environment at temperatures intermediate to the maximum and minimum cycle temperatures, thus enabling the thermal efficiency of the cycle (though not of any practical engine) to approach the limiting Carnot efficiency.

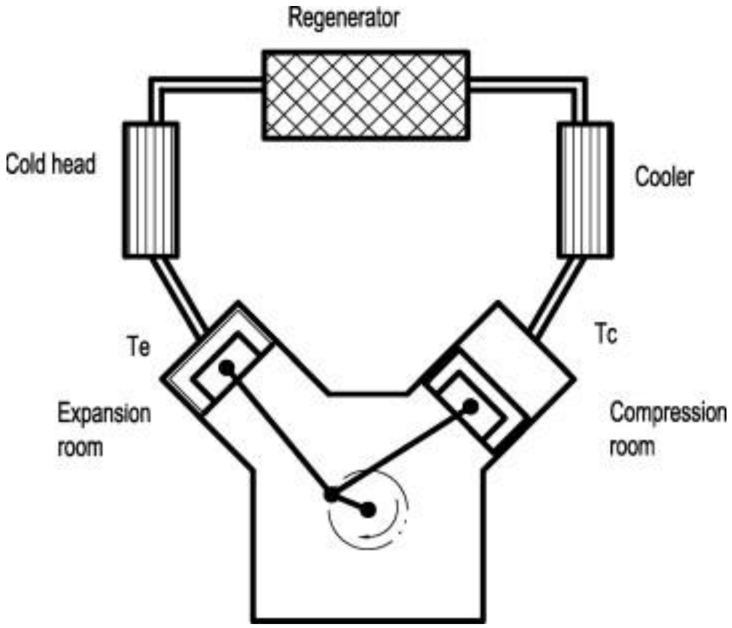


Fig 4.3: Regenerator

The primary effect of regeneration in a stirling refrigeration is to increase the thermal efficiency by 'recycling' internal heat that would otherwise pass through the engine irreversibly. As a secondary effect, increased thermal efficiency yields a higher power output from a given set of hot and cold end heat exchangers. These usually limit the engine's heat throughput. In practice this additional power may not be fully realized as the additional "dead space" (unswept volume) and pumping loss inherent in practical regenerators reduces the potential efficiency gains from regeneration.[7]

4.2.6 Regenerator Design Factors

While planning the design for the regenerator in Stirling Cycle Refrigeration System, the factors that affect the efficiency of the system is considered. The factors are:
1) The material of the regenerator
2) The composition of the regenerator

Chapter 5: Design and Implementation

5.1 Major components required for implementing the design

The Cylinders

The cylinders used for the experimental setup is made of cast iron and the dimensions are meticulously designed for the best outcome of the experiment. These cylinders are also covered on the top by two air tight lids. The lids also have a very small hole to connect the regenerator. The lids are also made of cast iron.

Fig 5.1: The cylinder

The dimension of the cylinder in the experimental setup:
1) Cylinder inner diameter 38 mm
2) Cylinder outer dia 44 mm
3) Thickness 3 mm
4) Cylinder height 85 mm

The Pistons

The piston in each cylinder has been manufactured with Aluminum. There are three piston rings are also attached to prevent the piston from knocking on the cylinder wall. The connecting rod that connects the piston with the crankshaft is made of cast iron.

Fig 5.2: The piston

The dimension of the piston in the experimental Setup
1) Piston diameter 37.6 mm
2) Piston travel 40 mm

The piston is the moving component contained in a cylinder and is made gas-tight by piston rings. Here, the pistons in reciprocating motion compresses the working fluid and rises the temperature in the hot cylinder and the fluid expands at regenerator as well as in the other cylinder. Thus it loses its heat in the process.

The Crankshaft

The reciprocating motion of the piston and connecting rod is converted to rotating motion via crankshaft which is made from cast iron. It is manufactured with convenient fit along with the piston head and connecting rod as well as the cylinder.

Fig 5.3: The crankshaft

The Regenerator

The regenerator in this case has been made by inserting stainless steel wire mesh structures inside a plastic tube. This is essentially the most basic way a regenerator can be manufactured. The stainless steel wire mesh structures act as a place where heat can be stored temporarily. Benefits from using stainless steel wire mesh structures include high surface area, high heat capacity, low conductivity and low flow friction.

Fig 5.4: Regenerator

In the experimental setup, stainless steel wire mesh is stuffed all along the plastic tube regenerator which gives the extended area for expansion of the working fluid.

The Electric Motor

The electric AC motor used in the experiment is power provider of the system. It provides rotational motion to the shaft connected with the crankshaft. The crankshaft then transmits the power to the piston converting the energy into reciprocating motion. This motor has the capacity of 0.5 HP.

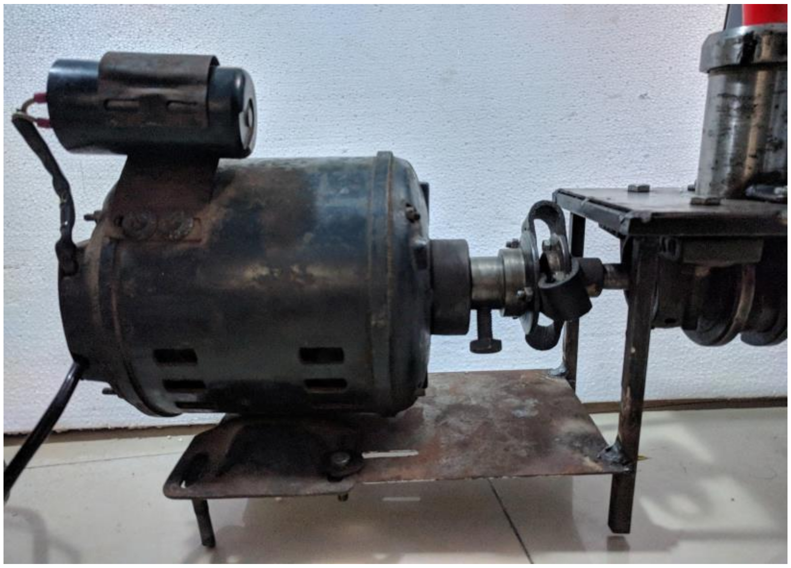


Fig 5.5: Electric motor

The electric motor used in this case is a 0.5 hp motor which has a load free rotational speed of 3000 rpm.

The Cooling Chamber

A cooling chamber has been constructed to cool down whatever is kept inside the chamber. The chamber is essentially a box built around the cylinder which isolates it from the environment and gets cold when the system is running or functioning. It is made with polystyrene.

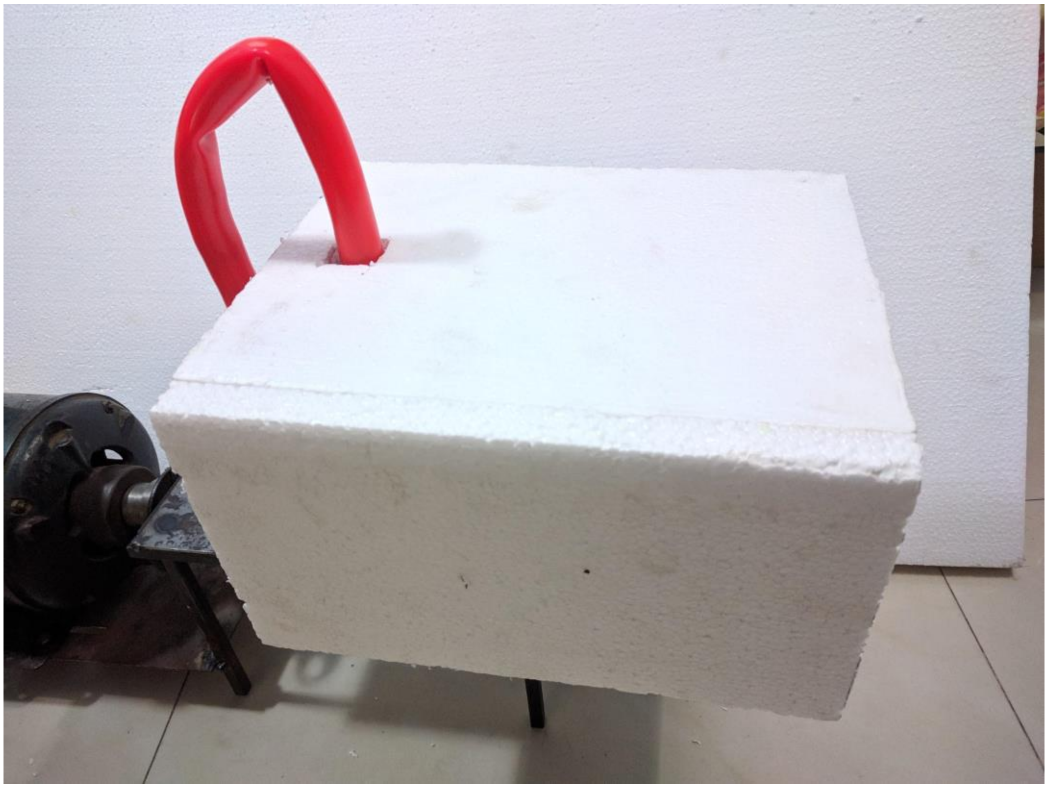


Fig 5.6: The Cooling Chamber

Fig 5.7: Inside the cooling chamber

The full setup

The final outlook of the experimental setup is illustrated below:

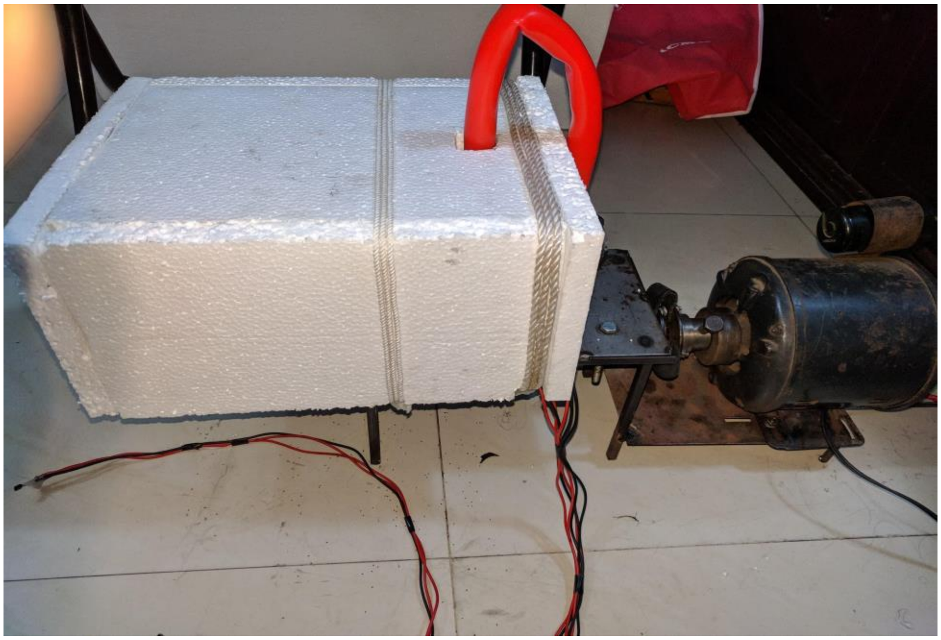


Fig 5.8: The entire experimental setup

5.2 Auxiliary components required for taking measurements

Arduino

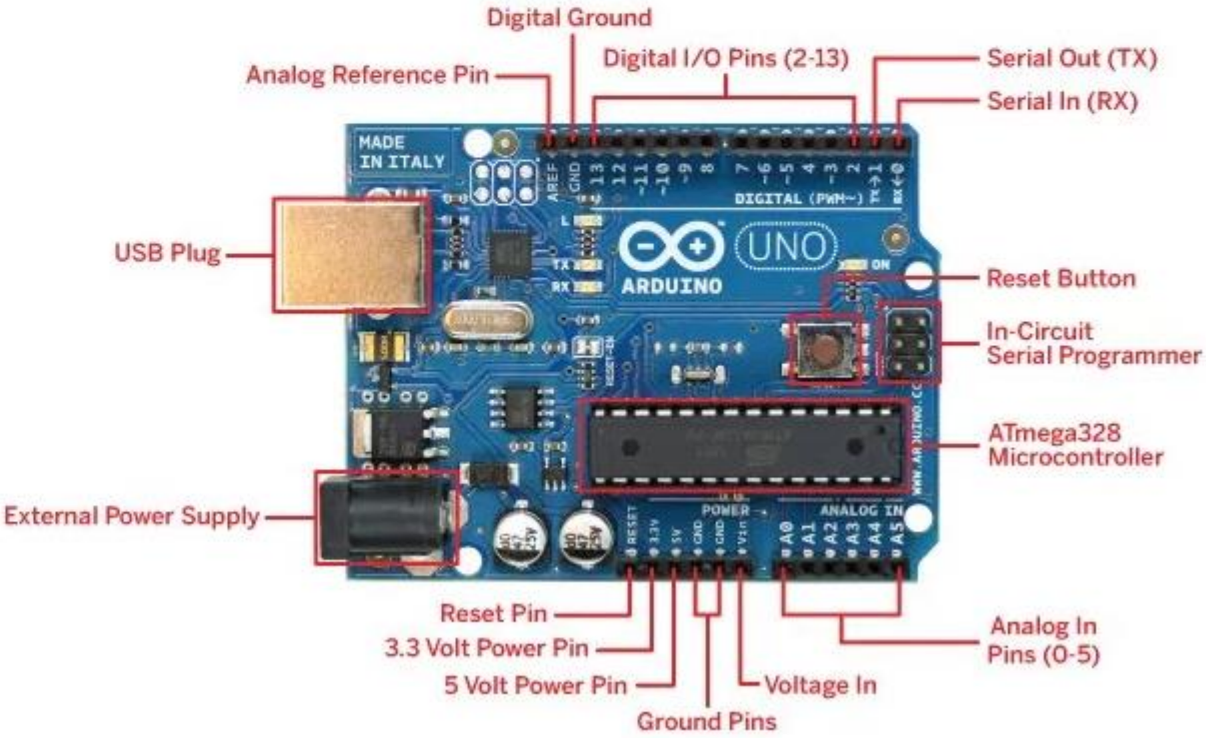


Fig 5.9: Arduino

Arduino Specifications [11]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

LM 35 Temperature Sensor

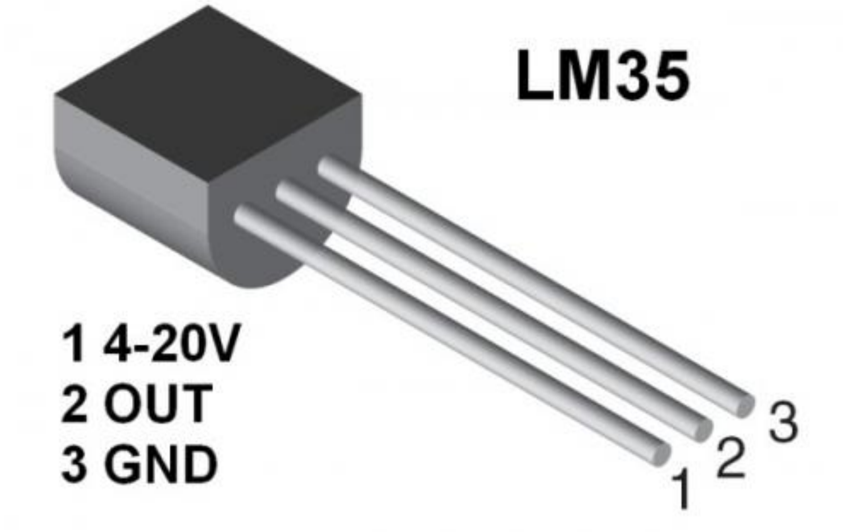


Fig 5.10: LM 35 Temperature Sensor

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

LM 35 Temperature Sensor Specifications

Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load[12]

5.3 Setup for recording and calculating experimental data

The whole system can be divided into two parts.

1) The first part uses a microprocessor (in this case an Arduino uno) and three temperature sensors (in this case LM35 temperature sensors) to read hot side temperature, cold side temperature and ambient temperature. These data are sent over to a computer through USB.
2) In the second part, a dot net application written in C# is used in the computer to read the data sent by the microprocessor and calculate coefficient of performance (COP) of the system and record them along with the temperatures in an html file.

First Part - Microprocessor Side

Sample Connection

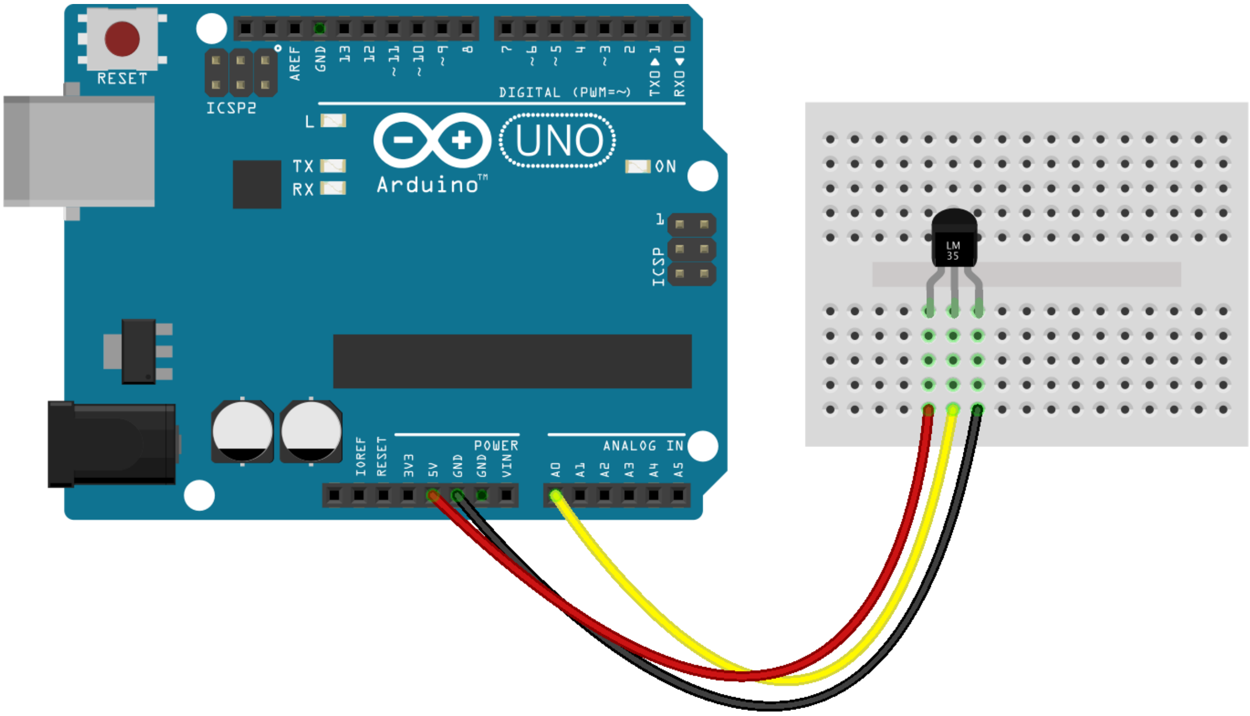


Fig 5.11: Sample Connection

The LM 35 has 3 pins.

1) The leftmost pin in this configuration is for the 5V supply
2) The rightmost pin is for the ground
3) The center pin goes to the analog input

The sensor can directly output temperature as (℃) when value from the analog input is multiplied by (500/1024) or (0.48828125).

Arduino Logic

Float c represents correction factor for the LM35 temperature sensors used. The sensors used in this experiment were fairly accurate. So, the correction factor in this case was set to zero.

Integer t represents the number of times the ***main loop*** has to go around in order for the elapsed time in the microprocessor to be equal to 1 second. So the logic is to get **2450** different temperature readings in 1 second for the hot side, cold side and the ambient side and calculate their averages and print them in a line by using “**^**” as their separator.

String data = (String)t1 + "^" + (String)t2 + "^" + (String)t3;
This line of code gets the average temperatures and implodes them in a single string by separating them using “**^**”.

Serial.println(data);
This line of code prints the data in a line. However, in this case, the line is sent via USB to the computer where a C# dot net application is setup to listen to this line and make necessary calculations afterwards.

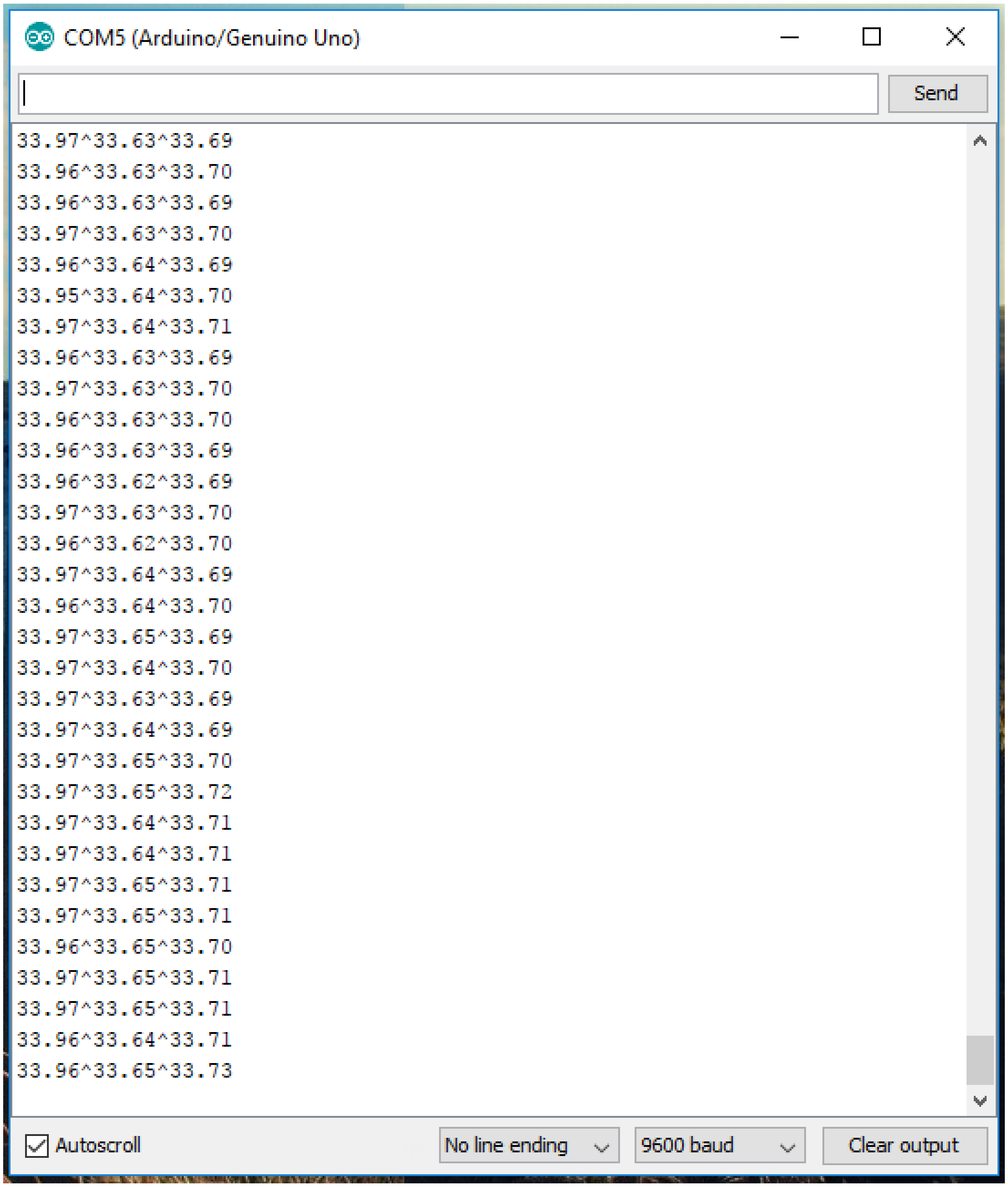
analogReference(5);
This line of code sets the reference voltage of the arduino to be equal to 5V. As the power to the board comes through an USB connection, 5V was set as the reference voltage.

Serial.begin(9600) doesn't actually print anything. Rather it initializes the serial connection at 9600 bits per second.
Both sides of the serial connection (i.e. the Arduino and the computer) need to be set to use the same speed serial connection in order to get any sort of intelligible data. If there's a mismatch between what the two systems think the speed is then the data will be garbled.
9600 bits per second is the default for the Arduino, and is perfectly adequate for the majority of users.

Serial.flush() pauses the program while the transmit buffer is flushed.

Sample Arduino Output

Using the serial monitor in arduino IDE, sample output of the arduino using the above mentioned code can be observed.

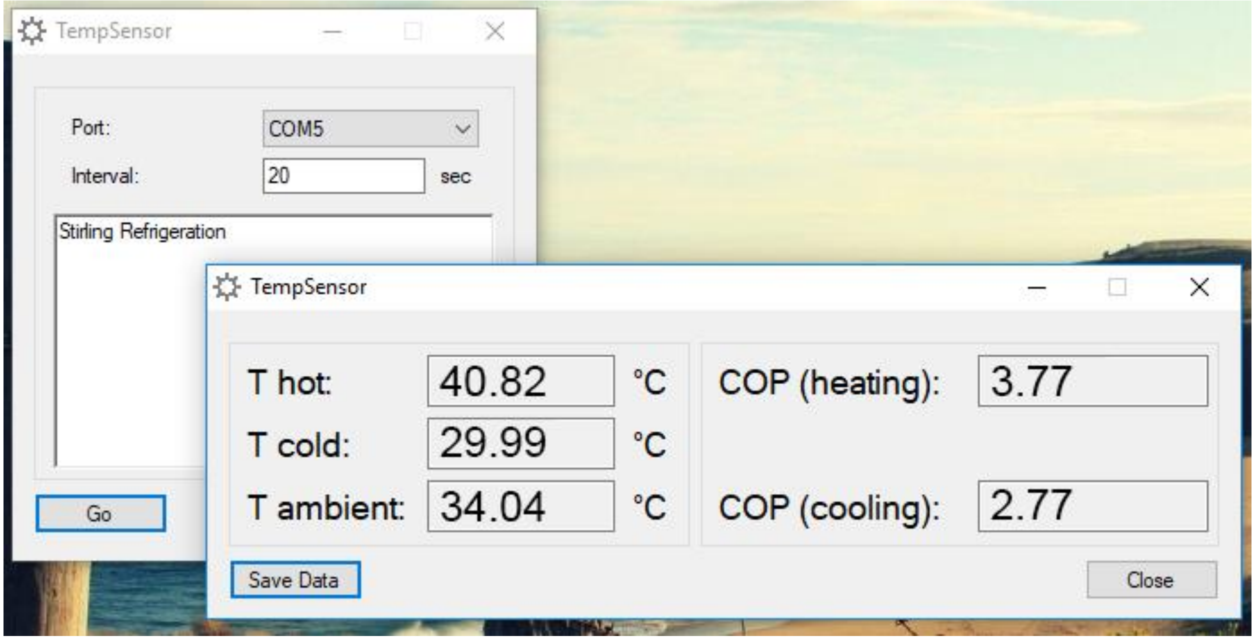


```
33.97^33.63^33.69
33.96^33.63^33.70
33.96^33.63^33.69
33.97^33.63^33.70
33.96^33.64^33.69
33.95^33.64^33.70
33.97^33.64^33.71
33.96^33.63^33.69
33.97^33.63^33.70
33.96^33.63^33.70
33.96^33.63^33.69
33.96^33.62^33.69
33.97^33.63^33.70
33.96^33.62^33.70
33.97^33.64^33.69
33.96^33.64^33.70
33.97^33.65^33.69
33.97^33.64^33.70
33.97^33.63^33.69
33.97^33.64^33.69
33.97^33.65^33.70
33.97^33.65^33.72
33.97^33.64^33.71
33.97^33.64^33.71
33.97^33.65^33.71
33.97^33.65^33.71
33.96^33.65^33.70
33.97^33.65^33.71
33.97^33.65^33.71
33.96^33.64^33.71
33.96^33.65^33.73
```

Fig 5.12: Sample Arduino Output

Second Part - Computer Side

Dot Net Application



Port: COM5

Interval: 20 sec

Stirling Refrigeration

T hot: 40.82 °C

T cold: 29.99 °C

T ambient: 34.04 °C

COP (heating): 3.77

COP (cooling): 2.77

Fig 5.13: Dot Net Application

Application Logic

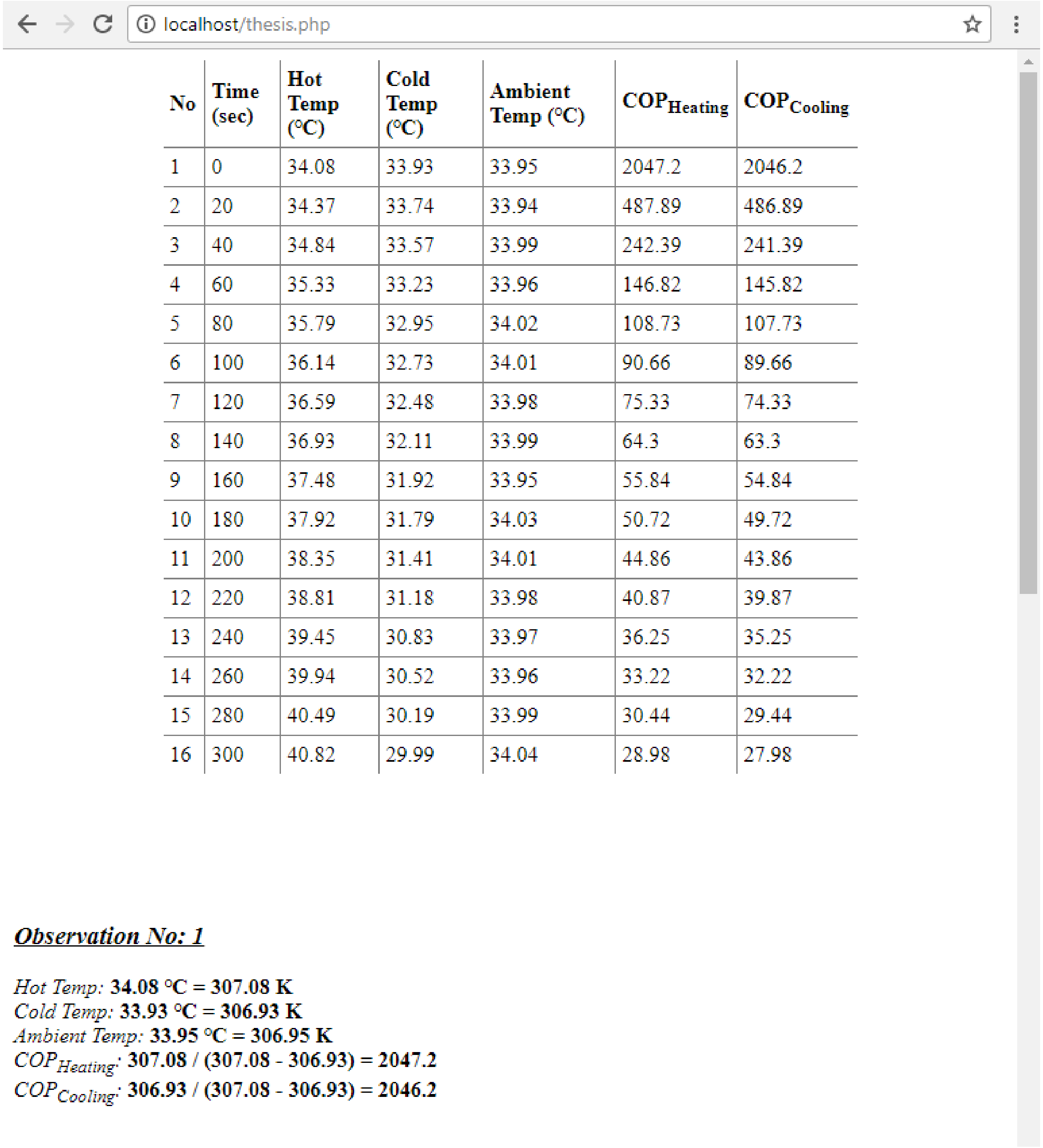
This application is written in C#.

LineReceived(string line) receives a new line from the microprocessor through USB each second in **[** T(hot) ^ T(cold) ^ T(ambient) **]** this format which is essentially the temperatures imploded together using the character “**^**”.

line.Split('^') is used to split the line into temperatures and assign them into three variables.

These variables are used to calculate the COPs each second and store them in an html file.

Application Sample Output (HTML File)



No	Time (sec)	Hot Temp (°C)	Cold Temp (°C)	Ambient Temp (°C)	$COP_{Heating}$	$COP_{Cooling}$
1	0	34.08	33.93	33.95	2047.2	2046.2
2	20	34.37	33.74	33.94	487.89	486.89
3	40	34.84	33.57	33.99	242.39	241.39
4	60	35.33	33.23	33.96	146.82	145.82
5	80	35.79	32.95	34.02	108.73	107.73
6	100	36.14	32.73	34.01	90.66	89.66
7	120	36.59	32.48	33.98	75.33	74.33
8	140	36.93	32.11	33.99	64.3	63.3
9	160	37.48	31.92	33.95	55.84	54.84
10	180	37.92	31.79	34.03	50.72	49.72
11	200	38.35	31.41	34.01	44.86	43.86
12	220	38.81	31.18	33.98	40.87	39.87
13	240	39.45	30.83	33.97	36.25	35.25
14	260	39.94	30.52	33.96	33.22	32.22
15	280	40.49	30.19	33.99	30.44	29.44
16	300	40.82	29.99	34.04	28.98	27.98

Observation No: 1

Hot Temp: **34.08 °C = 307.08 K**
Cold Temp: **33.93 °C = 306.93 K**
Ambient Temp: **33.95 °C = 306.95 K**
$COP_{Heating}$: **307.08 / (307.08 - 306.93) = 2047.2**
$COP_{Cooling}$: **306.93 / (307.08 - 306.93) = 2046.2**

Fig 5.14: Application Sample Output

Chapter 6: Analysis of the System

6.1 Experimental data, calculation and result

Experimental Data #1

No	**Time (sec)**	**Hot Temp (℃)**	**Cold Temp (℃)**	**Ambient Temp (℃)**	**$COP_{Heating}$**	**$COP_{Cooling}$**
1	0	34.08	33.93	33.95	2047.2	2046.2
2	60	34.37	33.74	33.94	487.89	486.89
3	120	34.84	33.57	33.99	242.39	241.39
4	180	35.33	33.23	33.96	146.82	145.82
5	240	35.79	32.95	34.02	108.73	107.73
6	300	36.14	32.73	34.01	90.66	89.66
7	360	36.59	32.48	33.98	75.33	74.33
8	420	36.93	32.11	33.99	64.3	63.3
9	480	37.48	31.92	33.95	55.84	54.84

10	540	37.92	31.79	34.03	50.72	49.72
11	600	38.35	31.41	34.01	44.86	43.86
12	660	38.81	31.18	33.98	40.87	39.87
13	720	39.14	30.83	33.97	37.56	36.56
14	780	39.75	30.52	33.96	33.88	32.88
15	840	39.91	30.19	33.99	32.19	31.19
16	900	40.36	29.99	34.04	30.22	29.22
17	960	40.58	29.85	34.02	29.22	28.22
18	1020	40.83	29.79	33.97	28.43	27.43
19	1080	40.96	29.74	33.96	27.98	26.98
20	1140	40.95	29.76	33.94	28.06	27.06

Observation No: 1

Hot Temp: **34.08 ℃ = 307.08 K**
Cold Temp: **33.93 ℃ = 306.93 K**
Ambient Temp: **33.95 ℃ = 306.95 K**
$COP_{Heating}$: **307.08 / (307.08 - 306.93) = 2047.2**
$COP_{Cooling}$: **306.93 / (307.08 - 306.93) = 2046.2**

Observation No: 2

Hot Temp: **34.37 ℃ = 307.37 K**
Cold Temp: **33.74 ℃ = 306.74 K**
Ambient Temp: **33.94 ℃ = 306.94 K**
$COP_{Heating}$: **307.37 / (307.37 - 306.74) = 487.89**
$COP_{Cooling}$: **306.74 / (307.37 - 306.74) = 486.89**

Observation No: 3

Hot Temp: **34.84 ℃ = 307.84 K**
Cold Temp: **33.57 ℃ = 306.57 K**
Ambient Temp: **33.99 ℃ = 306.99 K**
$COP_{Heating}$: **307.84 / (307.84 - 306.57) = 242.39**
$COP_{Cooling}$: **306.57 / (307.84 - 306.57) = 241.39**

Observation No: 4

Hot Temp: **35.33 ℃ = 308.33 K**
Cold Temp: **33.23 ℃ = 306.23 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **308.33 / (308.33 - 306.23) = 146.82**
$COP_{Cooling}$: **306.23 / (308.33 - 306.23) = 145.82**

Observation No: 5

Hot Temp: **35.79 ℃ = 308.79 K**
Cold Temp: **32.95 ℃ = 305.95 K**
Ambient Temp: **34.02 ℃ = 307.02 K**
$COP_{Heating}$: **308.79 / (308.79 - 305.95) = 108.73**
$COP_{Cooling}$: **305.95 / (308.79 - 305.95) = 107.73**

Observation No: 6

Hot Temp: **36.14 ℃ = 309.14 K**
Cold Temp: **32.73 ℃ = 305.73 K**
Ambient Temp: **34.01 ℃ = 307.01 K**
$COP_{Heating}$: **309.14 / (309.14 - 305.73) = 90.66**
$COP_{Cooling}$: **305.73 / (309.14 - 305.73) = 89.66**

Observation No: 7

Hot Temp: **36.59 ℃ = 309.59 K**
Cold Temp: **32.48 ℃ = 305.48 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **309.59 / (309.59 - 305.48) = 75.33**
$COP_{Cooling}$: **305.48 / (309.59 - 305.48) = 74.33**

Observation No: 8

Hot Temp: **36.93 ℃ = 309.93 K**
Cold Temp: **32.11 ℃ = 305.11 K**
Ambient Temp: **33.99 ℃ = 306.99 K**
$COP_{Heating}$: **309.93 / (309.93 - 305.11) = 64.3**
$COP_{Cooling}$: **305.11 / (309.93 - 305.11) = 63.3**

Observation No: 9

Hot Temp: **37.48 ℃ = 310.48 K**
Cold Temp: **31.92 ℃ = 304.92 K**
Ambient Temp: **33.95 ℃ = 306.95 K**
$COP_{Heating}$: **310.48 / (310.48 - 304.92) = 55.84**
$COP_{Cooling}$: **304.92 / (310.48 - 304.92) = 54.84**

Observation No: 10

Hot Temp: **37.92 ℃ = 310.92 K**
Cold Temp: **31.79 ℃ = 304.79 K**
Ambient Temp: **34.03 ℃ = 307.03 K**
$COP_{Heating}$: **310.92 / (310.92 - 304.79) = 50.72**
$COP_{Cooling}$: **304.79 / (310.92 - 304.79) = 49.72**

Observation No: 11

Hot Temp: **38.35 ℃ = 311.35 K**
Cold Temp: **31.41 ℃ = 304.41 K**
Ambient Temp: **34.01 ℃ = 307.01 K**
$COP_{Heating}$: **311.35 / (311.35 - 304.41) = 44.86**
$COP_{Cooling}$: **304.41 / (311.35 - 304.41) = 43.86**

Observation No: 12

Hot Temp: **38.81 ℃ = 311.81 K**
Cold Temp: **31.18 ℃ = 304.18 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **311.81 / (311.81 - 304.18) = 40.87**
$COP_{Cooling}$: **304.18 / (311.81 - 304.18) = 39.87**

Observation No: 13

Hot Temp: **39.14 ℃ = 312.14 K**
Cold Temp: **30.83 ℃ = 303.83 K**
Ambient Temp: **33.97 ℃ = 306.97 K**
$COP_{Heating}$: **312.14 / (312.14 - 303.83) = 37.56**
$COP_{Cooling}$: **303.83 / (312.14 - 303.83) = 36.56**

Observation No: 14

Hot Temp: **39.75 ℃ = 312.75 K**
Cold Temp: **30.52 ℃ = 303.52 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **312.75 / (312.75 - 303.52) = 33.88**
$COP_{Cooling}$: **303.52 / (312.75 - 303.52) = 32.88**

Observation No: 15

Hot Temp: **39.91 ℃ = 312.91 K**
Cold Temp: **30.19 ℃ = 303.19 K**
Ambient Temp: **33.99 ℃ = 306.99 K**
$COP_{Heating}$: **312.91 / (312.91 - 303.19) = 32.19**
$COP_{Cooling}$: **303.19 / (312.91 - 303.19) = 31.19**

Observation No: 16

Hot Temp: **40.36 ℃ = 313.36 K**
Cold Temp: **29.99 ℃ = 302.99 K**
Ambient Temp: **34.04 ℃ = 307.04 K**
$COP_{Heating}$: **313.36 / (313.36 - 302.99) = 30.22**
$COP_{Cooling}$: **302.99 / (313.36 - 302.99) = 29.22**

Observation No: 17

Hot Temp: **40.58 ℃ = 313.58 K**
Cold Temp: **29.85 ℃ = 302.85 K**
Ambient Temp: **34.02 ℃ = 307.02 K**
$COP_{Heating}$: **313.58 / (313.58 - 302.85) = 29.22**
$COP_{Cooling}$: **302.85 / (313.58 - 302.85) = 28.22**

Observation No: 18

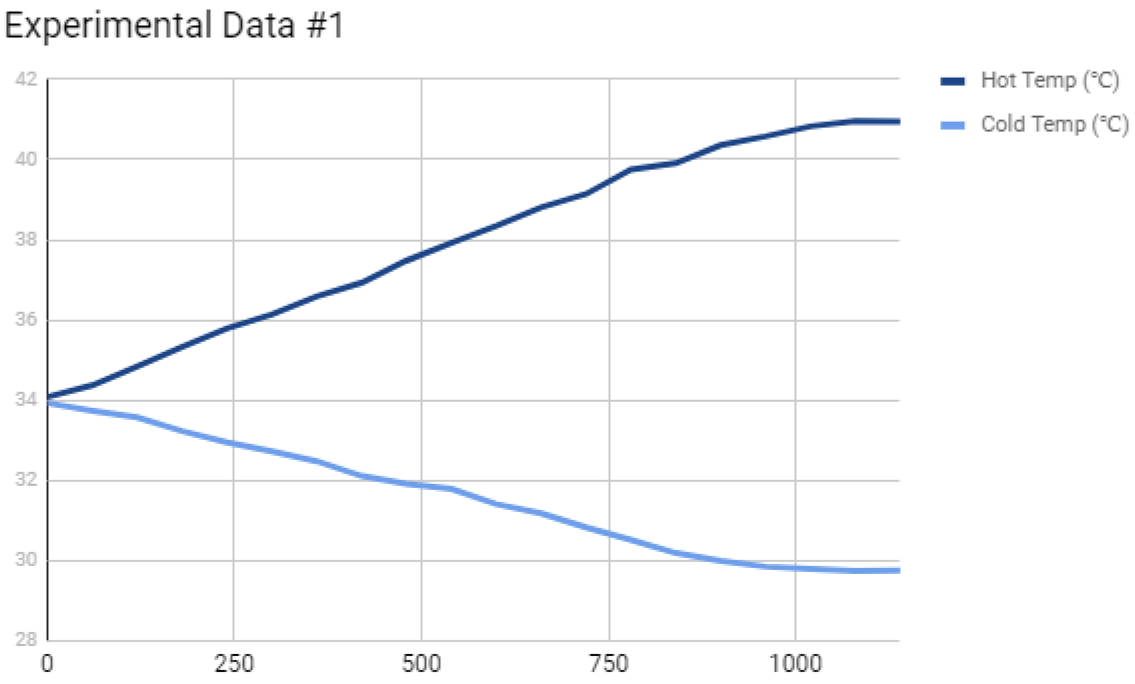
Hot Temp: **40.83 ℃ = 313.83 K**
Cold Temp: **29.79 ℃ = 302.79 K**
Ambient Temp: **33.97 ℃ = 306.97 K**
$COP_{Heating}$: **313.83 / (313.83 - 302.79) = 28.43**
$COP_{Cooling}$: **302.79 / (313.83 - 302.79) = 27.43**

Observation No: 19

Hot Temp: **40.96 ℃ = 313.96 K**
Cold Temp: **29.74 ℃ = 302.74 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **313.96 / (313.96 - 302.74) = 27.98**
$COP_{Cooling}$: **302.74 / (313.96 - 302.74) = 26.98**

Observation No: 20

Hot Temp: **40.95 ℃ = 313.95 K**
Cold Temp: **29.76 ℃ = 302.76 K**
Ambient Temp: **33.94 ℃ = 306.94 K**
$COP_{Heating}$: **313.95 / (313.95 - 302.76) = 28.06**
$COP_{Cooling}$: **302.76 / (313.95 - 302.76) = 27.06**



→ Time (sec)

Fig 6.1: Temperature Vs Time graph for Experiment #1

Experimental Data #2

No	**Time (sec)**	**Hot Temp (℃)**	**Cold Temp (℃)**	**Ambient Temp (℃)**	**COP$_{Heating}$**	**COP$_{Cooling}$**
1	0	34.21	34.14	34.09	4388.71	4387.71
2	60	34.48	33.95	34.05	580.15	579.15
3	120	34.94	33.62	33.96	233.29	232.29
4	180	35.23	33.29	33.99	158.88	157.88
5	240	35.72	32.89	34.06	109.09	108.09
6	300	36.08	32.64	33.94	89.85	88.85
7	360	36.69	32.55	33.98	74.8	73.8
8	420	36.99	32.26	34.03	65.54	64.54
9	480	37.36	31.94	33.96	57.26	56.26
10	540	37.58	31.62	34.02	52.11	51.11
11	600	37.94	31.48	33.91	48.13	47.13
12	660	38.57	31.29	33.94	42.8	41.8

13	720	38.98	30.93	33.98	38.76	37.76
14	780	39.73	30.74	34.04	34.79	33.79
15	840	40.65	30.39	34.08	30.57	29.57
16	900	40.89	30.02	34.04	28.88	27.88
17	960	41.11	29.93	33.98	28.1	27.1
18	1020	41.24	29.87	33.96	27.64	26.64
19	1080	41.58	29.84	34.02	26.8	25.8
20	1140	41.69	29.85	33.97	26.58	25.58

Observation No: 1

Hot Temp: **34.21 ℃ = 307.21 K**
Cold Temp: **34.14 ℃ = 307.14 K**
Ambient Temp: **34.09 ℃ = 307.09 K**
$COP_{Heating}$: **307.21 / (307.21 - 307.14) = 4388.71**
$COP_{Cooling}$: **307.14 / (307.21 - 307.14) = 4387.71**

Observation No: 2

Hot Temp: **34.48 ℃ = 307.48 K**
Cold Temp: **33.95 ℃ = 306.95 K**
Ambient Temp: **34.05 ℃ = 307.05 K**
$COP_{Heating}$: **307.48 / (307.48 - 306.95) = 580.15**
$COP_{Cooling}$: **306.95 / (307.48 - 306.95) = 579.15**

Observation No: 3

Hot Temp: **34.94 ℃ = 307.94 K**
Cold Temp: **33.62 ℃ = 306.62 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **307.94 / (307.94 - 306.62) = 233.29**
$COP_{Cooling}$: **306.62 / (307.94 - 306.62) = 232.29**

Observation No: 4

Hot Temp: **35.23 ℃ = 308.23 K**
Cold Temp: **33.29 ℃ = 306.29 K**
Ambient Temp: **33.99 ℃ = 306.99 K**
$COP_{Heating}$: **308.23 / (308.23 - 306.29) = 158.88**
$COP_{Cooling}$: **306.29 / (308.23 - 306.29) = 157.88**

Observation No: 5

Hot Temp: **35.72 ℃ = 308.72 K**
Cold Temp: **32.89 ℃ = 305.89 K**
Ambient Temp: **34.06 ℃ = 307.06 K**
$COP_{Heating}$: **308.72 / (308.72 - 305.89) = 109.09**
$COP_{Cooling}$: **305.89 / (308.72 - 305.89) = 108.09**

Observation No: 6

Hot Temp: **36.08 ℃ = 309.08 K**
Cold Temp: **32.64 ℃ = 305.64 K**
Ambient Temp: **33.94 ℃ = 306.94 K**
$COP_{Heating}$: **309.08 / (309.08 - 305.64) = 89.85**
$COP_{Cooling}$: **305.64 / (309.08 - 305.64) = 88.85**

Observation No: 7

Hot Temp: **36.69 ℃ = 309.69 K**
Cold Temp: **32.55 ℃ = 305.55 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **309.69 / (309.69 - 305.55) = 74.8**
$COP_{Cooling}$: **305.55 / (309.69 - 305.55) = 73.8**

Observation No: 8

Hot Temp: **36.99 ℃ = 309.99 K**
Cold Temp: **32.26 ℃ = 305.26 K**
Ambient Temp: **34.03 ℃ = 307.03 K**
$COP_{Heating}$: **309.99 / (309.99 - 305.26) = 65.54**
$COP_{Cooling}$: **305.26 / (309.99 - 305.26) = 64.54**

Observation No: 9

Hot Temp: **37.36 ℃ = 310.36 K**
Cold Temp: **31.94 ℃ = 304.94 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **310.36 / (310.36 - 304.94) = 57.26**
$COP_{Cooling}$: **304.94 / (310.36 - 304.94) = 56.26**

Observation No: 10

Hot Temp: **37.58 ℃ = 310.58 K**
Cold Temp: **31.62 ℃ = 304.62 K**
Ambient Temp: **34.02 ℃ = 307.02 K**
$COP_{Heating}$: **310.58 / (310.58 - 304.62) = 52.11**
$COP_{Cooling}$: **304.62 / (310.58 - 304.62) = 51.11**

Observation No: 11

Hot Temp: **37.94 ℃ = 310.94 K**
Cold Temp: **31.48 ℃ = 304.48 K**
Ambient Temp: **33.91 ℃ = 306.91 K**
$COP_{Heating}$: **310.94 / (310.94 - 304.48) = 48.13**
$COP_{Cooling}$: **304.48 / (310.94 - 304.48) = 47.13**

Observation No: 12

Hot Temp: **38.57 ℃ = 311.57 K**
Cold Temp: **31.29 ℃ = 304.29 K**
Ambient Temp: **33.94 ℃ = 306.94 K**
$COP_{Heating}$: **311.57 / (311.57 - 304.29) = 42.8**
$COP_{Cooling}$: **304.29 / (311.57 - 304.29) = 41.8**

Observation No: 13

Hot Temp: **38.98 ℃ = 311.98 K**
Cold Temp: **30.93 ℃ = 303.93 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **311.98 / (311.98 - 303.93) = 38.76**
$COP_{Cooling}$: **303.93 / (311.98 - 303.93) = 37.76**

Observation No: 14

Hot Temp: **39.73 ℃ = 312.73 K**
Cold Temp: **30.74 ℃ = 303.74 K**
Ambient Temp: **34.04 ℃ = 307.04 K**
$COP_{Heating}$: **312.73 / (312.73 - 303.74) = 34.79**
$COP_{Cooling}$: **303.74 / (312.73 - 303.74) = 33.79**

Observation No: 15

Hot Temp: **40.65 ℃ = 313.65 K**
Cold Temp: **30.39 ℃ = 303.39 K**
Ambient Temp: **34.08 ℃ = 307.08 K**
$COP_{Heating}$: **313.65 / (313.65 - 303.39) = 30.57**
$COP_{Cooling}$: **303.39 / (313.65 - 303.39) = 29.57**

Observation No: 16

Hot Temp: **40.89 ℃ = 313.89 K**
Cold Temp: **30.02 ℃ = 303.02 K**
Ambient Temp: **34.04 ℃ = 307.04 K**
$COP_{Heating}$: **313.89 / (313.89 - 303.02) = 28.88**
$COP_{Cooling}$: **303.02 / (313.89 - 303.02) = 27.88**

Observation No: 17

Hot Temp: **41.11 ℃ = 314.11 K**
Cold Temp: **29.93 ℃ = 302.93 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **314.11 / (314.11 - 302.93) = 28.1**
$COP_{Cooling}$: **302.93 / (314.11 - 302.93) = 27.1**

Observation No: 18

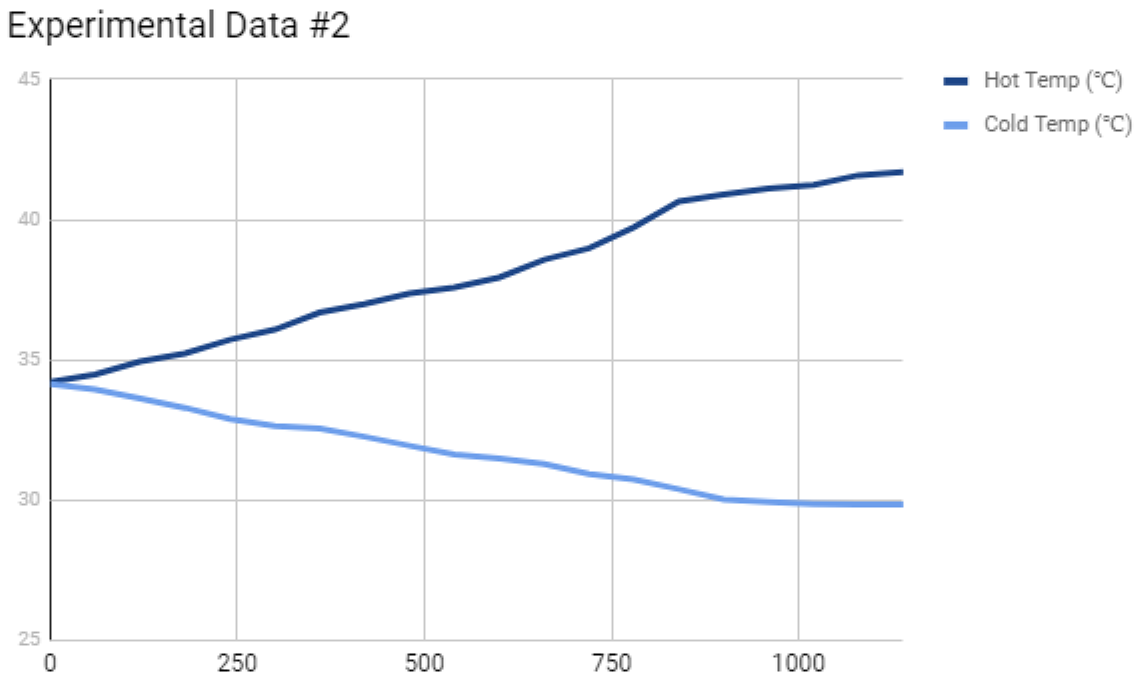
Hot Temp: **41.24 ℃ = 314.24 K**
Cold Temp: **29.87 ℃ = 302.87 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **314.24 / (314.24 - 302.87) = 27.64**
$COP_{Cooling}$: **302.87 / (314.24 - 302.87) = 26.64**

Observation No: 19

Hot Temp: **41.58 ℃ = 314.58 K**
Cold Temp: **29.84 ℃ = 302.84 K**
Ambient Temp: **34.02 ℃ = 307.02 K**
$COP_{Heating}$: **314.58 / (314.58 - 302.84) = 26.8**
$COP_{Cooling}$: **302.84 / (314.58 - 302.84) = 25.8**

Observation No: 20

Hot Temp: **41.69 ℃ = 314.69 K**
Cold Temp: **29.85 ℃ = 302.85 K**
Ambient Temp: **33.97 ℃ = 306.97 K**
$COP_{Heating}$: **314.69 / (314.69 - 302.85) = 26.58**
$COP_{Cooling}$: **302.85 / (314.69 - 302.85) = 25.58**



→ Time (sec)

Fig 6.2: Temperature Vs Time graph for Experiment #2

Experimental Data #3

No	**Time (sec)**	**Hot Temp (℃)**	**Cold Temp (℃)**	**Ambient Temp (℃)**	**$\text{COP}_{\text{Heating}}$**	**$\text{COP}_{\text{Cooling}}$**
1	0	34.02	33.98	34.08	7675.5	7674.5
2	60	34.15	33.83	34.04	959.84	958.84
3	120	34.29	33.57	33.97	426.79	425.79
4	180	34.63	33.16	33.98	209.27	208.27
5	240	34.72	32.83	34.05	162.81	161.81
6	300	35.04	32.78	33.96	136.3	135.3
7	360	35.59	32.43	33.95	97.66	96.66
8	420	36.15	32.21	34.02	78.46	77.46
9	480	36.74	31.96	33.97	64.8	63.8
10	540	36.92	31.49	34.03	57.08	56.08
11	600	37.17	31.33	33.92	53.11	52.11
12	660	37.67	31.18	33.95	47.87	46.87

13	720	37.95	30.89	33.96	44.04	43.04
14	780	38.26	30.73	34.03	41.34	40.34
15	840	38.57	30.29	34.07	37.63	36.63
16	900	38.93	30.03	34.06	35.05	34.05
17	960	39.38	29.94	33.94	33.09	32.09
18	1020	39.63	29.73	33.93	31.58	30.58
19	1080	39.91	29.68	34.04	30.59	29.59
20	1140	40.01	29.66	33.99	30.24	29.24

Observation No: 1

Hot Temp: **34.02 ℃ = 307.02 K**
Cold Temp: **33.98 ℃ = 306.98 K**
Ambient Temp: **34.08 ℃ = 307.08 K**
$COP_{Heating}$: **307.02 / (307.02 - 306.98) = 7675.5**
$COP_{Cooling}$: **306.98 / (307.02 - 306.98) = 7674.5**

Observation No: 2

Hot Temp: **34.15 ℃ = 307.15 K**
Cold Temp: **33.83 ℃ = 306.83 K**
Ambient Temp: **34.04 ℃ = 307.04 K**
$COP_{Heating}$: **307.15 / (307.15 - 306.83) = 959.84**
$COP_{Cooling}$: **306.83 / (307.15 - 306.83) = 958.84**

Observation No: 3

Hot Temp: **34.29 ℃ = 307.29 K**
Cold Temp: **33.57 ℃ = 306.57 K**
Ambient Temp: **33.97 ℃ = 306.97 K**
$COP_{Heating}$: **307.29 / (307.29 - 306.57) = 426.79**
$COP_{Cooling}$: **306.57 / (307.29 - 306.57) = 425.79**

Observation No: 4

Hot Temp: **34.63 ℃ = 307.63 K**
Cold Temp: **33.16 ℃ = 306.16 K**
Ambient Temp: **33.98 ℃ = 306.98 K**
$COP_{Heating}$: **307.63 / (307.63 - 306.16) = 209.27**
$COP_{Cooling}$: **306.16 / (307.63 - 306.16) = 208.27**

Observation No: 5

Hot Temp: **34.72 ℃ = 307.72 K**
Cold Temp: **32.83 ℃ = 305.83 K**
Ambient Temp: **34.05 ℃ = 307.05 K**
$COP_{Heating}$: **307.72 / (307.72 - 305.83) = 162.81**
$COP_{Cooling}$: **305.83 / (307.72 - 305.83) = 161.81**

Observation No: 6

Hot Temp: **35.04 ℃ = 308.04 K**
Cold Temp: **32.78 ℃ = 305.78 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **308.04 / (308.04 - 305.78) = 136.3**
$COP_{Cooling}$: **305.78 / (308.04 - 305.78) = 135.3**

Observation No: 7

Hot Temp: **35.59 ℃ = 308.59 K**
Cold Temp: **32.43 ℃ = 305.43 K**
Ambient Temp: **33.95 ℃ = 306.95 K**
$COP_{Heating}$: **308.59 / (308.59 - 305.43) = 97.66**
$COP_{Cooling}$: **305.43 / (308.59 - 305.43) = 96.66**

Observation No: 8

Hot Temp: **36.15 ℃ = 309.15 K**
Cold Temp: **32.21 ℃ = 305.21 K**
Ambient Temp: **34.02 ℃ = 307.02 K**
$COP_{Heating}$: **309.15 / (309.15 - 305.21) = 78.46**
$COP_{Cooling}$: **305.21 / (309.15 - 305.21) = 77.46**

Observation No: 9

Hot Temp: **36.74 ℃ = 309.74 K**
Cold Temp: **31.96 ℃ = 304.96 K**
Ambient Temp: **33.97 ℃ = 306.97 K**
$COP_{Heating}$: **309.74 / (309.74 - 304.96) = 64.8**
$COP_{Cooling}$: **304.96 / (309.74 - 304.96) = 63.8**

Observation No: 10

Hot Temp: **36.92 ℃ = 309.92 K**
Cold Temp: **31.49 ℃ = 304.49 K**
Ambient Temp: **34.03 ℃ = 307.03 K**
$COP_{Heating}$: **309.92 / (309.92 - 304.49) = 57.08**
$COP_{Cooling}$: **304.49 / (309.92 - 304.49) = 56.08**

Observation No: 11

Hot Temp: **37.17 ℃ = 310.17 K**
Cold Temp: **31.33 ℃ = 304.33 K**
Ambient Temp: **33.92 ℃ = 306.92 K**
$COP_{Heating}$: **310.17 / (310.17 - 304.33) = 53.11**
$COP_{Cooling}$: **304.33 / (310.17 - 304.33) = 52.11**

Observation No: 12

Hot Temp: **37.67 ℃ = 310.67 K**
Cold Temp: **31.18 ℃ = 304.18 K**
Ambient Temp: **33.95 ℃ = 306.95 K**
$COP_{Heating}$: **310.67 / (310.67 - 304.18) = 47.87**
$COP_{Cooling}$: **304.18 / (310.67 - 304.18) = 46.87**

Observation No: 13

Hot Temp: **37.95 ℃ = 310.95 K**
Cold Temp: **30.89 ℃ = 303.89 K**
Ambient Temp: **33.96 ℃ = 306.96 K**
$COP_{Heating}$: **310.95 / (310.95 - 303.89) = 44.04**
$COP_{Cooling}$: **303.89 / (310.95 - 303.89) = 43.04**

Observation No: 14

Hot Temp: **38.26 ℃ = 311.26 K**
Cold Temp: **30.73 ℃ = 303.73 K**
Ambient Temp: **34.03 ℃ = 307.03 K**
$COP_{Heating}$: **311.26 / (311.26 - 303.73) = 41.34**
$COP_{Cooling}$: **303.73 / (311.26 - 303.73) = 40.34**

Observation No: 15

Hot Temp: **38.57 ℃ = 311.57 K**
Cold Temp: **30.29 ℃ = 303.29 K**
Ambient Temp: **34.07 ℃ = 307.07 K**
$COP_{Heating}$: **311.57 / (311.57 - 303.29) = 37.63**
$COP_{Cooling}$: **303.29 / (311.57 - 303.29) = 36.63**

Observation No: 16

Hot Temp: **38.93 ℃ = 311.93 K**
Cold Temp: **30.03 ℃ = 303.03 K**
Ambient Temp: **34.06 ℃ = 307.06 K**
$COP_{Heating}$: **311.93 / (311.93 - 303.03) = 35.05**
$COP_{Cooling}$: **303.03 / (311.93 - 303.03) = 34.05**

Observation No: 17

Hot Temp: **39.38 ℃ = 312.38 K**
Cold Temp: **29.94 ℃ = 302.94 K**
Ambient Temp: **33.94 ℃ = 306.94 K**
$COP_{Heating}$: **312.38 / (312.38 - 302.94) = 33.09**
$COP_{Cooling}$: **302.94 / (312.38 - 302.94) = 32.09**

Observation No: 18

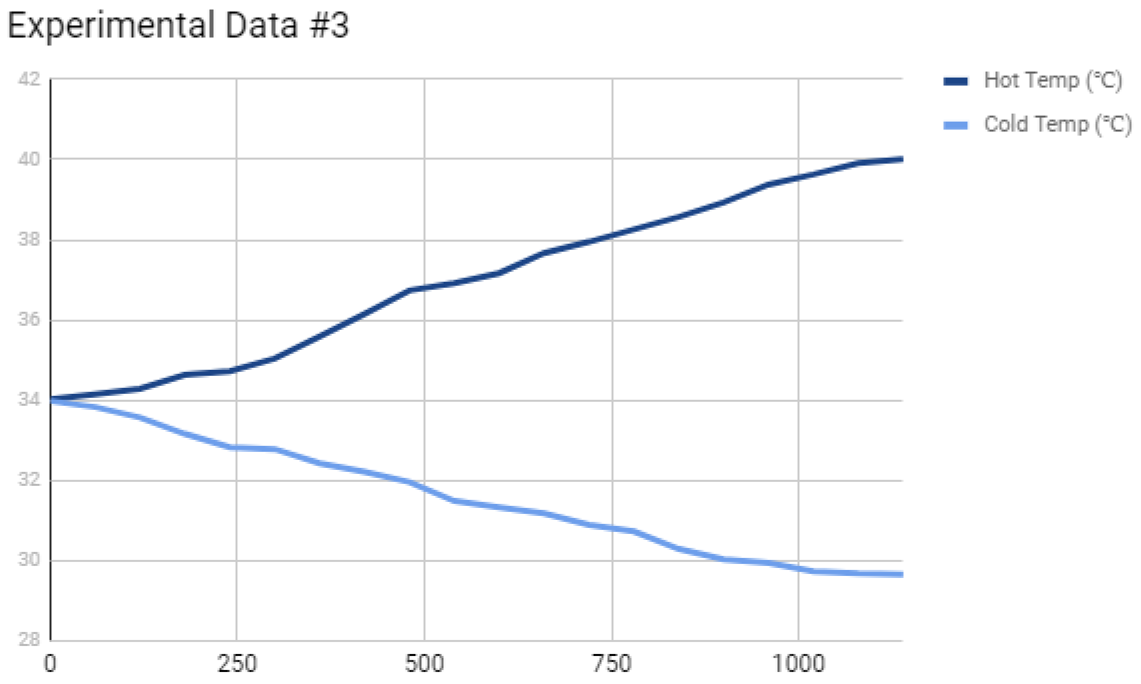
Hot Temp: **39.63 ℃ = 312.63 K**
Cold Temp: **29.73 ℃ = 302.73 K**
Ambient Temp: **33.93 ℃ = 306.93 K**
$COP_{Heating}$: **312.63 / (312.63 - 302.73) = 31.58**
$COP_{Cooling}$: **302.73 / (312.63 - 302.73) = 30.58**

Observation No: 19

Hot Temp: **39.91 ℃ = 312.91 K**
Cold Temp: **29.68 ℃ = 302.68 K**
Ambient Temp: **34.04 ℃ = 307.04 K**
$COP_{Heating}$: **312.91 / (312.91 - 302.68) = 30.59**
$COP_{Cooling}$: **302.68 / (312.91 - 302.68) = 29.59**

Observation No: 20

Hot Temp: **40.01 ℃ = 313.01 K**
Cold Temp: **29.66 ℃ = 302.66 K**
Ambient Temp: **33.99 ℃ = 306.99 K**
$COP_{Heating}$: **313.01 / (313.01 - 302.66) = 30.24**
$COP_{Cooling}$: **302.66 / (313.01 - 302.66) = 29.24**



→ Time (sec)

Fig 6.3: Temperature Vs Time graph for Experiment #3

6.2 Discussion

1) Three sets of experiments were carried out
2) All of them had a time duration of 1140 seconds
3) Ambient temperature was about 34°C in all of the cases
4) The first experiment yielded a maximum hot side temperature of 40.96°C and a minimum cold side temperature of 29.74°C
5) The $COP_{Heating}$ at these temperatures was 27.98 and $COP_{Cooling}$ was 26.98
6) The second experiment yielded a maximum hot side temperature of 41.69°C and a minimum cold side temperature of 29.85°C
7) The $COP_{Heating}$ at these temperatures was 26.58 and $COP_{Cooling}$ was 25.58
8) The third experiment yielded a maximum hot side temperature of 40.01°C and a minimum cold side temperature of 29.66°C
9) The $COP_{Heating}$ at these temperatures was 30.24 and $COP_{Cooling}$ was 29.24
10) Conventional stirling refrigerators are capable of achieving cryogenic temperatures
11) However, this experimental setup was able to lower temperature by only 4-5°C
12) Such large deviations from conventional setups can be attributed to the fact that this is only an experimental setup and it has not been made with the finest of machines
13) Although high quality lubrication oil was used, friction played an important role in hampering the effectiveness of the setup
14) Conventional systems use compressed Helium or compressed Hydrogen. In this setup, air at atmospheric pressure was used
15) Conventional systems reach rotational speeds of up to 8000 rpm. This system used 400 rpm at most which is only a fraction of the conventional rotational speed
16) Conventional systems use fins to enlarge the heat exchanging areas. Fins were not used in this experimental setup
17) The effectiveness of the regenerator was also compromised due to the fact that the stainless steel mesh used in this setup was not dense enough
18) But the experimental setup clearly demonstrated the fundamentals of the stirling cycle refrigeration system

Chapter 7: Future Development and Scope

1) **Introduction:** Due to the fact that the development of this prototype was in small scale and with a limited budget, it has certain scopes to develop further. The future scopes will increase the efficiency of the entire system. The possible future developments are discussed in this chapter.

2) **Manufacture of Cylinder:** The manufacturing of cylinder is usually done by casting. But in this case, the cylinder has been made using lathe machine. That is why the design of the cylinder was not precise. As a result, the pistons create more friction while reciprocating. So if the cylinder is manufactured via casting with higher precision, there will be less friction and as a result, more efficiency.

3) **Good Sealing:** As the working fluid is needed to be inside a closed cycle, the sealing of the entire system should be good enough to make it air-tight. In this case, the system was not fully air-tight.

4) **Fin around the Cylinder:** As the scale of the prototype was small, it was not possible to put aluminum fins around the cylinder. Fins increase heat transfer rate from and to the cylinder and increases efficiency.

5) **Material and structure of the Regenerator:** Instead of using stainless steel coil mesh, copper coils can be used to improve the regenerating effect of the system. More materials and structures can be put through testing to find the best combination for the regenerator.

6) **Working Fluid:** In this setup, air at atmospheric pressure has been used as the working fluid. But this system can work more efficiently if the working fluid is helium or hydrogen at higher pressure instead of air at atmospheric pressure. Even nitrogen can be used as working fluid and it will be more efficient than the current prototype.

Chapter 8: Conclusion

8.1 Application of the system

Refrigeration and cooling:

One of their modern uses is in cryogenics and, to a lesser extent, refrigeration. At typical refrigeration temperatures, Stirling coolers are generally not economically competitive with the less expensive mainstream Rankine cooling systems, because they are less energy-efficient. However, below about −40...−30 °C, Rankine cooling is not effective because there are no suitable refrigerants with boiling points this low. Stirling cryocoolers are able to "lift" heat down to −200 °C (73 K), which is sufficient to liquefy air (specifically the primary constituent gases oxygen, nitrogen and argon). They can go as low as 40–60 K for single-stage machines, depending on the particular design. Two-stage Stirling cryocoolers can reach temperatures of 20 K, sufficient to liquefy hydrogen and neon. Cryocoolers for this purpose are more or less competitive with other cryocooler technologies. The coefficient of performance at cryogenic temperatures is typically 0.04–0.05 (corresponding to a 4–5% efficiency).[2]

Heat pump:

A Stirling heat pump is very similar to a Stirling cryocooler, the main difference being that it usually operates at room temperature. At present, its principal application is to pump heat from the outside of a building to the inside, thus heating it at lowered energy costs.

As with any other Stirling device, heat flow is from the expansion space to the compression space. However, in contrast to the Stirling engine, the expansion space is at a lower temperature than the compression space, so instead of producing work, an input of mechanical work is required by the system (in order to satisfy the Second Law of Thermodynamics). The mechanical energy input can be supplied by an electrical motor, or an internal combustion engine, for example. When the mechanical work for the heat pump is provided by a second Stirling engine, then the overall system is called a "heat-driven heat pump".

The expansion side of the heat pump is thermally coupled to the heat source, which is often the external environment. The compression side of the Stirling device is placed in the environment to be heated, for example a building, and heat is "pumped" into it. Typically there will be thermal insulation between the two sides so there will be a temperature rise inside the insulated space.

Heat pumps are by far the most energy-efficient types of heating systems, since they "harvest" additional heat from the environment, rather than turning all their input energy directly into heat. In accordance with the Second Law of Thermodynamics, heat pumps always require the additional input of some external energy to "pump" the collected heat "uphill" against a temperature differential.

Compared to conventional heat pumps, Stirling heat pumps often have a higher coefficient of performance. To date, Stirling systems have seen limited commercial use; however, use is expected to increase along with market demand for energy conservation, and adoption will likely be accelerated by technological refinements.[2]

Portable refrigeration:

The Free Piston Stirling Cooler (FPSC) is a completely sealed heat transfer system that has only two moving parts (a piston and a displacer), and which can use helium as the working fluid. The piston is typically driven by an oscillating magnetic field that is the source of the power needed to drive the refrigeration cycle. The magnetic drive allows the piston to be driven without requiring any seals, gaskets, O-rings, or other compromises to the hermetically sealed system. Claimed advantages for the system include improved efficiency and cooling capacity, lighter weight, smaller size and better controllability.

For several years starting around 2004, the Coleman Company sold a version of the Twinbird "SC-C925 Portable Freezer Cooler 25L" under its own brand name, but it has since discontinued offering the product. The portable cooler can be operated more than a day, maintaining sub-freezing temperatures while powered by an automotive battery. This cooler is still being manufactured, with Global Cooling now coordinating distribution to North America and Europe. Other variants offered by Twinbird include a portable deep freezer (to −80 °C), collapsible coolers, and a model for transporting blood and vaccine.[2]

8.2 Conclusions

As well as their role as high-efficiency low-noise engines, Stirling-cycle machines offer an environmentally-friendly alternative to vapour-compression systems, by virtue of their ability to use air as a refrigerant (the ultimate environmentally-safe chemical).

In comparison to other thermodynamic cycles, the Stirling Cycle is extremely difficult to optimize for a practical machine. When optimized, however, Stirling-cycle machines have similar performance to traditional vapour-compression systems; although there are certain contexts where they can, in fact, provide significantly higher performance. It can therefore be said that Stirling-cycle machines provide a technically realistic alternative to current vapour compression technology. The main challenge will lie in achieving a competitive price through a low-cost mechanical design configuration, and manufacturing in sufficiently large volumes.

Chapter 9: References

1) https://en.wikipedia.org/wiki/Stirling_engine
2) https://en.wikipedia.org/wiki/Cryocooler
3) https://www.stirlingcryogenics.com/en/the-stirling-cycle
4) https://www.sciencedirect.com/science/article/pii/S1359431108000926
5) “Stirling-Cycle Heat-Pumps and Refrigerators – a Realistic Alternative?” - Stirling Cycle Research Group, Department of Mechanical Engineering, University of Canterbury, New Zealand.
6) West, C.D., Principles and applications of Stirling Engines, Van Nostrand Reinhold Co., New York, 1986.
7) https://link.springer.com/chapter/10.1007/978-1-4615-2522-6_166
8) http://iopscience.iop.org/article/10.1088/1361-6404/aa776d/meta
9) http://www.robertstirlingengine.com/cold.php
10) https://en.wikipedia.org/wiki/Applications_of_the_Stirling_engine
11) https://store.arduino.cc/usa/arduino-uno-rev3
12) http://www.ece.usu.edu/ece_store/spec/lm35dt-3p.pdf

Appendix A

Arduino Code

```
float c = 0.0;
int t = 2450;

void setup()
{
Serial.begin(9600);
analogReference(5);
}
void loop()
{

float t1=0, t2=0, t3=0;


for(int n = 0; n <=t; //main loop
{
t1 += ( analogRead(0)/1024.0)*500;
t2 += ( analogRead(2)/1024.0)*500;
t3 += ( analogRead(4)/1024.0)*500;


}

t1 = c + (t1/t);
t2 = c + (t2/t);
t3 = c + (t3/t);



String data = (String)t1 + "^" + (String)t2 + "^" + (String)t3;

Serial.println(data);

Serial.flush();

}
```

Appendix B

Dot Net Application Code

Main Entry Point:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace TempSensor
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Form 1 Design Code:

```
namespace TempSensor
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
```

```
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.button1 = new System.Windows.Forms.Button();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.richTextBox1 = new System.Windows.Forms.RichTextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.comboBox1 = new System.Windows.Forms.ComboBox();
            this.label1 = new System.Windows.Forms.Label();
            this.button2 = new System.Windows.Forms.Button();
            this.groupBox1.SuspendLayout();
            this.SuspendLayout();
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(12, 245);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75, 23);
            this.button1.TabIndex = 0;
            this.button1.Text = "Go";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // groupBox1
```

```
//
this.groupBox1.Controls.Add(this.richTextBox1);
this.groupBox1.Controls.Add(this.label4);
this.groupBox1.Controls.Add(this.textBox1);
this.groupBox1.Controls.Add(this.label3);
this.groupBox1.Controls.Add(this.comboBox1);
this.groupBox1.Controls.Add(this.label1);
this.groupBox1.Location = new System.Drawing.Point(12, 12);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(269, 227);
this.groupBox1.TabIndex = 1;
this.groupBox1.TabStop = false;
//
// richTextBox1
//
this.richTextBox1.Location = new System.Drawing.Point(11, 77);
this.richTextBox1.Margin = new System.Windows.Forms.Padding(8);
this.richTextBox1.Name = "richTextBox1";
this.richTextBox1.Size = new System.Drawing.Size(247, 143);
this.richTextBox1.TabIndex = 7;
this.richTextBox1.Text = "Experiment Details";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(225, 49);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(24, 13);
this.label4.TabIndex = 6;
this.label4.Text = "sec";
//
// textBox1
//
this.textBox1.Location = new System.Drawing.Point(128, 46);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(91, 20);
this.textBox1.TabIndex = 5;
this.textBox1.Text = "10";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(18, 49);
this.label3.Name = "label3";
```

```
          this.label3.Size = new System.Drawing.Size(45, 13);
          this.label3.TabIndex = 4;
          this.label3.Text = "Interval:";
          //
          // comboBox1
          //
          this.comboBox1.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
          this.comboBox1.FormattingEnabled = true;
          this.comboBox1.Location = new System.Drawing.Point(128, 19);
          this.comboBox1.Name = "comboBox1";
          this.comboBox1.Size = new System.Drawing.Size(121, 21);
          this.comboBox1.TabIndex = 1;
          //
          // label1
          //
          this.label1.AutoSize = true;
          this.label1.Location = new System.Drawing.Point(18, 22);
          this.label1.Name = "label1";
          this.label1.Size = new System.Drawing.Size(29, 13);
          this.label1.TabIndex = 0;
          this.label1.Text = "Port:";
          //
          // button2
          //
          this.button2.Location = new System.Drawing.Point(206, 245);
          this.button2.Name = "button2";
          this.button2.Size = new System.Drawing.Size(75, 23);
          this.button2.TabIndex = 2;
          this.button2.Text = "Exit";
          this.button2.UseVisualStyleBackColor = true;
          this.button2.Click += new System.EventHandler(this.button2_Click);
          //
          // Form1
          //
          this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
          this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
          this.ClientSize = new System.Drawing.Size(293, 283);
          this.Controls.Add(this.button2);
          this.Controls.Add(this.groupBox1);
          this.Controls.Add(this.button1);
          this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
          this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
          this.MaximizeBox = false;
```

```
            this.Name = "Form1";
            this.Text = "TempSensor";
            this.Load += new System.EventHandler(this.Form1_Load);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.ComboBox comboBox1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.RichTextBox richTextBox1;
    }
}
```

Form 1 Controller:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;

namespace TempSensor
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            if (String.IsNullOrEmpty(comboBox1.Text) || String.IsNullOrEmpty(textBox1.Text))
MessageBox.Show("Please select all the parameters");

            else
            {
                Form Form2 = new Form2(comboBox1.Text, textBox1.Text, richTextBox1.Text);
                Form2.ShowDialog();
                Form2.Dispose();
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

Directory.CreateDirectory(Environment.GetFolderPath(Environment.SpecialFolder.MyDocument
s) + @"\TempSensor\");
            string[] ports = SerialPort.GetPortNames();
            foreach (string port in ports)
            {
                comboBox1.Items.Add(port);
            }
        }


    }
}
```

Form 2 Design Code:

```
namespace TempSensor
{
    partial class Form2
    {
        /// <summary>
        /// Required designer variable.
```

```
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                try { components.Dispose(); }
                catch { }
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.serialPort1 = new System.IO.Ports.SerialPort(this.components);
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.label16 = new System.Windows.Forms.Label();
            this.label14 = new System.Windows.Forms.Label();
            this.label13 = new System.Windows.Forms.Label();
            this.textBox3 = new System.Windows.Forms.TextBox();
            this.textBox2 = new System.Windows.Forms.TextBox();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.button1 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.textBox5 = new System.Windows.Forms.TextBox();
            this.textBox6 = new System.Windows.Forms.TextBox();
            this.label8 = new System.Windows.Forms.Label();
```

```
        this.label9 = new System.Windows.Forms.Label();
        this.groupBox1.SuspendLayout();
        this.groupBox2.SuspendLayout();
        this.SuspendLayout();
        //
        // groupBox1
        //
        this.groupBox1.Controls.Add(this.label16);
        this.groupBox1.Controls.Add(this.label14);
        this.groupBox1.Controls.Add(this.label13);
        this.groupBox1.Controls.Add(this.textBox3);
        this.groupBox1.Controls.Add(this.textBox2);
        this.groupBox1.Controls.Add(this.textBox1);
        this.groupBox1.Controls.Add(this.label3);
        this.groupBox1.Controls.Add(this.label2);
        this.groupBox1.Controls.Add(this.label1);
        this.groupBox1.Location = new System.Drawing.Point(12, 12);
        this.groupBox1.Name = "groupBox1";
        this.groupBox1.Size = new System.Drawing.Size(258, 121);
        this.groupBox1.TabIndex = 0;
        this.groupBox1.TabStop = false;
        //
        // label16
        //
        this.label16.AutoSize = true;
        this.label16.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.label16.Location = new System.Drawing.Point(222, 86);
        this.label16.Name = "label16";
        this.label16.Size = new System.Drawing.Size(29, 24);
        this.label16.TabIndex = 14;
        this.label16.Text = "°C";
        //
        // label14
        //
        this.label14.AutoSize = true;
        this.label14.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.label14.Location = new System.Drawing.Point(222, 51);
        this.label14.Name = "label14";
        this.label14.Size = new System.Drawing.Size(29, 24);
        this.label14.TabIndex = 13;
        this.label14.Text = "°C";
        //
```

```
            // label13
            //
            this.label13.AutoSize = true;
            this.label13.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label13.Location = new System.Drawing.Point(222, 16);
            this.label13.Name = "label13";
            this.label13.Size = new System.Drawing.Size(29, 24);
            this.label13.TabIndex = 12;
            this.label13.Text = "°C";
            //
            // textBox3
            //
            this.textBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox3.Location = new System.Drawing.Point(111, 83);
            this.textBox3.Name = "textBox3";
            this.textBox3.ReadOnly = true;
            this.textBox3.Size = new System.Drawing.Size(105, 29);
            this.textBox3.TabIndex = 8;
            this.textBox3.TabStop = false;
            //
            // textBox2
            //
            this.textBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox2.Location = new System.Drawing.Point(111, 48);
            this.textBox2.Name = "textBox2";
            this.textBox2.ReadOnly = true;
            this.textBox2.Size = new System.Drawing.Size(105, 29);
            this.textBox2.TabIndex = 7;
            this.textBox2.TabStop = false;
            //
            // textBox1
            //
            this.textBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.textBox1.Location = new System.Drawing.Point(111, 13);
            this.textBox1.Name = "textBox1";
            this.textBox1.ReadOnly = true;
            this.textBox1.Size = new System.Drawing.Size(105, 29);
            this.textBox1.TabIndex = 6;
            this.textBox1.TabStop = false;
            //
```

```
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label3.Location = new System.Drawing.Point(6, 86);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(99, 24);
            this.label3.TabIndex = 2;
            this.label3.Text = "T ambient:";
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label2.Location = new System.Drawing.Point(6, 51);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(68, 24);
            this.label2.TabIndex = 1;
            this.label2.Text = "T cold:";
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.label1.Location = new System.Drawing.Point(6, 16);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(58, 24);
            this.label1.TabIndex = 0;
            this.label1.Text = "T hot:";
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(12, 139);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75, 23);
            this.button1.TabIndex = 5;
            this.button1.Text = "Save Data";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // button2
```

```
        //
        this.button2.Location = new System.Drawing.Point(491, 139);
        this.button2.Name = "button2";
        this.button2.Size = new System.Drawing.Size(75, 23);
        this.button2.TabIndex = 6;
        this.button2.Text = "Close";
        this.button2.UseVisualStyleBackColor = true;
        this.button2.Click += new System.EventHandler(this.button2_Click);
        //
        // groupBox2
        //
        this.groupBox2.Controls.Add(this.textBox5);
        this.groupBox2.Controls.Add(this.textBox6);
        this.groupBox2.Controls.Add(this.label8);
        this.groupBox2.Controls.Add(this.label9);
        this.groupBox2.Location = new System.Drawing.Point(276, 12);
        this.groupBox2.Name = "groupBox2";
        this.groupBox2.Size = new System.Drawing.Size(290, 121);
        this.groupBox2.TabIndex = 15;
        this.groupBox2.TabStop = false;
        //
        // textBox5
        //
        this.textBox5.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.textBox5.Location = new System.Drawing.Point(155, 13);
        this.textBox5.Name = "textBox5";
        this.textBox5.ReadOnly = true;
        this.textBox5.Size = new System.Drawing.Size(129, 29);
        this.textBox5.TabIndex = 7;
        this.textBox5.TabStop = false;
        //
        // textBox6
        //
        this.textBox6.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.textBox6.Location = new System.Drawing.Point(155, 83);
        this.textBox6.Name = "textBox6";
        this.textBox6.ReadOnly = true;
        this.textBox6.Size = new System.Drawing.Size(129, 29);
        this.textBox6.TabIndex = 6;
        this.textBox6.TabStop = false;
        //
        // label8
```

```
      //
      this.label8.AutoSize = true;
      this.label8.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
      this.label8.Location = new System.Drawing.Point(6, 86);
      this.label8.Name = "label8";
      this.label8.Size = new System.Drawing.Size(134, 24);
      this.label8.TabIndex = 1;
      this.label8.Text = "COP (cooling):";
      //
      // label9
      //
      this.label9.AutoSize = true;
      this.label9.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
      this.label9.Location = new System.Drawing.Point(6, 16);
      this.label9.Name = "label9";
      this.label9.Size = new System.Drawing.Size(134, 24);
      this.label9.TabIndex = 0;
      this.label9.Text = "COP (heating):";
      //
      // Form2
      //
      this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
      this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
      this.ClientSize = new System.Drawing.Size(578, 172);
      this.Controls.Add(this.groupBox2);
      this.Controls.Add(this.button2);
      this.Controls.Add(this.button1);
      this.Controls.Add(this.groupBox1);
      this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
      this.MaximizeBox = false;
      this.Name = "Form2";
      this.Text = "TempSensor";
      this.Load += new System.EventHandler(this.Form2_Load);
      this.groupBox1.ResumeLayout(false);
      this.groupBox1.PerformLayout();
      this.groupBox2.ResumeLayout(false);
      this.groupBox2.PerformLayout();
      this.ResumeLayout(false);

    }

    #endregion
```

```
        private System.IO.Ports.SerialPort serialPort1;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.TextBox textBox3;
        private System.Windows.Forms.TextBox textBox2;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label16;
        private System.Windows.Forms.Label label14;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.TextBox textBox5;
        private System.Windows.Forms.TextBox textBox6;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label9;
    }
}
```

Form 2 Controller:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.IO.Ports;
using System.Threading;
using System.Windows.Forms.DataVisualization.Charting;

namespace TempSensor
{

    public partial class Form2 : Form
    {
```

```
        public string folder_name =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + @"\TempSensor\";
        public string port;

        public string interval;

        public string info;
        public string report;
        public int el_time = 0;
        public int no = 1;
        public int cr = 0;



        public Form2(string p1, string p3, string p5)
        {
            InitializeComponent();
            port = p1;

            interval = p3;

            info = p5;
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            serialPort1.PortName = port;
            try
            {
                report = "<!DOCTYPE html><meta charset=\"utf-8\"><html><div
align=center><h3>Data Sheet</h2>" + info + "<br><br><table cellspacing=3 cellpadding=3
rules=all width=70%><tr><td><b>No</b></td><td><b>Time (sec)</b></td><td><b>Hot Temp
(&#8451;)</b></td><td><b>Cold Temp (&#8451;)</b></td><td><b>Ambient Temp
(&#8451;)</b></td><td><b>COP<sub>Heating</b></td><td><b>COP<sub>Cooling</b></td></
tr>";
                serialPort1.Open();
                serialPort1.DataReceived += serialPort1_DataReceived;

            }
            catch
            {
                MessageBox.Show("There was an error. Please make sure that the correct port was
selected, and the device, plugged in.");
                this.Close();
```

```
            }
        }







        private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            try
            {
                string line = serialPort1.ReadLine();
                this.BeginInvoke(new LineReceivedEvent(LineReceived), line);
            }
            catch { }
        }

        private delegate void LineReceivedEvent(string line);
        private void LineReceived(string line)
        {


            try
            {
                string[] tempinfo = line.Split('^');
                string hot = tempinfo[0];
                string cold = tempinfo[1];
                string ambient = tempinfo[2];

                textBox1.Text = hot;
                textBox2.Text = cold;
                textBox3.Text = (Convert.ToDouble(ambient) + cr).ToString();

                double hot_temp = Convert.ToDouble(hot);
                double cold_temp = Convert.ToDouble(cold);
                double ambient_temp = Convert.ToDouble(ambient) + cr;

                double cop_h = (hot_temp + 273) / ((hot_temp + 273) - (cold_temp + 273));
                double cop_c = (cold_temp + 273) / ((hot_temp + 273) - (cold_temp + 273));

                textBox5.Text = Math.Round(cop_h, 2).ToString();
```

```
                textBox6.Text = Math.Round(cop_c, 2).ToString();

                el_time++;
                if (el_time % Convert.ToInt32(interval) == 0)
                {
                    report += "<tr><td>" + no + "</td><td>" + el_time + "</td><td>" +
Math.Round(hot_temp, 2) + "</td><td>" + Math.Round(cold_temp, 2) + "</td><td>" +
Math.Round(ambient_temp, 2) + "</td><td>" + Math.Round(cop_h, 2) + "</td><td>" +
Math.Round(cop_c, 2) + "</td></tr>";
                    no++;
                }


            }
            catch
            {

            }
        }

        private void button1_Click(object sender, EventArgs e)
        {

            string name = DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");

            string final_report = report + "</table></div></html>";
            System.IO.File.WriteAllText(folder_name + name + ".html", final_report);
            MessageBox.Show("Data Saved");
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```