**ABSTRACT**

Temperature related inconveniences such as heatstroke, heat rash, frostbite, hypothermia and others have been a persistent problem for people throughout history. Some of these conditions, when left unchecked, have led to unfortunate deaths. What is even more common is the un-satisfaction that people have with the weather at various points of the year. People often complain that it is either too hot or too cold. The current technological solutions made to keep people thermally comfortable such as AC and heating units have come a long way and have been successful in helping people often comfort in their dwelling (e.g. home or car), but are not personal mobility Solutions. If one has to be out in the weather, the addition or subtraction of layers with coats and jackets or beach wear, are popular solutions, to that problem, but do not always yield upmost satisfaction, for layers become cold over time and sunburn is a serious problem. Such a suit allows the user to control and monitor the internal temperature of the suit for high temperatures to low temperatures, depending on the seasons. So we propose a Heating and Cooling Suit that can maintain the body temperature automatically, manually or by a mobile app. The mobile app displays the current temperature, humidity level within the suit.

**INDEX**

|  |  |  |
| --- | --- | --- |
| 01 | INTRODUCTION | 03 |
| 02 | EXISTING SYSTEM | 04 |
| 03 | LITERATURE SURVEY | 05 |
| 04 | PROPOSED SYSTEM | 13 |
| 05 | REQUIREMENT ANALYSIS | 14 |
| 06 | PROPOSED ARCHITECTURE / DESIGN | 23 |
| 07 | TEST IMPLEMENTATION INTERPRETATION | 26 |
| 08 | RESULT AND DISCUSSION | 48 |
| 09 | ADVANTAGES AND DISADVANTAGES | 51 |
| 10 | CONCLUSION AND FUTURE SCOPE | 52 |
|  | REFERENCES | 53 |

**INTRODUCTION**

Both very cold and very hot temperatures could be dangerous to health. Excessive exposure to heat is referred to as heat stress and excessive exposure to cold is referred to as cold stress. In a very hot environment, the most serious concern is heat stroke. At very cold temperatures, the most serious concern is the risk of hypothermia or dangerous overcooling of the body. The proposed system is a battery power heating and cooling jacket, in which the user can regulate the temperature of the jacket through his mobile. After initialization temperature sensor measures the wearable coat temperature according to person’s surroundings. The jacket to a mobile app via Wi-Fi. The mobile app will display the current body temperature and has tools to vary the temperature of the jacket according to user’s need. We can perform both heating and cooling operations as per the requirement. The jacket can work both in manual or automated mode. In automatic mode the jacket reads the temperature via temperature sensor, sends the data to the raspberry pi, the logical operation is performed by the raspberry, while signals out the peltier to either heat or cool.

**EXISTING SYSTEM**

Existing system is a heating/cooling jacket, in which the user can control the temperature through controls and thermo-electric devices that are embedded in the suit. The functionality of the suit is, once turned on, the device displays the temperature of the inside of the suit in an LCD displays. Initiating the hot or cold functions are as simple as pushing a button, and a rotary knob allows the user to control the internal temperature of the suit. The thermoelectric cooler is a solid state heat pump made of thermocouples of high-efficiency semiconductor material that creates a difference in temperature of its two sides when a voltage is applied and current runs through it. None of the current system contains IoT connectivity, Transfer of heat or cold is either through cooper tubes, through thermo conductive fluid. Existing system

are all manual, either the jacket is controlled through knobs or on and off switch.

**LITERATURE SURVEY**

**Battery Powered Heating and Cooling Jacket**

Gregory Paul and Edward Gim, David Westerfeld at IEEE Long Island Systems, Application and Technology Conference , May 2014 Proposed

A battery powered heating/cooling jacket, in which the user can control the temperature through controls and thermo-electric devices that are embedded in the jacket. The functionality of the suit is as follows: but has the ability to turn on the suit like other garment, but has the ability to turn on the temperature control device within the suit. Once turned on, the device displays the temperature of the inside of the suit in an LCD displays. Initiating the hot or cold functions are as simple as pushing a button, and a rotary knob allows the user to control the internal temperature of the suit. This application requires the development of a system that ensures both warming and cooling. There is a very interesting device that does both heating and cooling simultaneously on a small scale. The thermoelectric cooler is a solid state heat pump made of thermocouples of high-efficiency semiconductor material that creates a difference in temperature of its two sides when a voltage is applied and current runs through it. This phenomenon is called Peltier effect. The final design is a microprocessor based system that heats and cools one side of several TECs by using an H bridge circuit that is enabled and controlled by “hot and cold” pushbuttons and a rotary variable resistor.

**Solar Battery Powered Heating and Cooling Suit**

T Ajay Kumar and S Vani at Open Access International Journal , 2015 Proposed

An E-Uniform which gives better protection to the soldiers who are working in extreme weather conditions. This Uniform will make the soldier to work in any kind of environment. Here we are using Solar Panels to power up the internal circuitry of the E-uniform. A 12 V DC lead acid rechargeable battery is used for storing the energy. We are using conventional battery charging unit also for giving supply to the circuitry. Microcontroller is the heart of the circuit as it controls all the functions. The project is operated in two modes summer mode and winter mode. By selecting the mode of operation such that it can drive body heater/cooler. The heater/cooler in turn will help us to provide chilling or warming effect inside the uniform which helps the soldier to bear to any kind of external environment and he can work efficiently without heat stress or cold stress. A 12 V DC lead acid rechargeable battery is used for storing the energy. We are using conventional battery charging unit also for giving supply to the circuitry. Here we are also using a metal detector and RF communication to know about the presence of any explosive material at the other place.

**Battery Powered Heating and Cooling Suit With location spotter**

|  |
| --- |
| C. Vinod Kumar at Open Access International Journal, Feb 2015 Proposed |

Soldiers are the Army’s most important resource. Sol­diers play a vital role to protect one’s country. The term soldiers include service men and women from the Army, Air Force, Navy and Marines. They will always be the one responsible for taking and holding the duty in extreme weather conditions throughout the year. While providing security for the nation, they may face troubles in extreme hot/cold weather conditions. The proposed system is battery powered heating and cooling suit with location spotter, by which the people/user can easily control the temperature of the suit and peltier plate. The user controls the peltier plate tem­perature by varying the variable resister. The suit is very flexible to wear, convenient, cozy, healthy and less in weight. The user wears an E-Uniform as a dress, and also there is a facility to switch on TEC in the E-Uniform. The final design is a microprocessor based system that the systems makes heat and cool both sides of TECs by using battery power supply. That the heat and cool functionality selected by us­ing push button switch/mode and we can adjust heat and cool by variable resister. Generally The TECs are out­side of the circuitry which are connected through wires and posed with in the suit so user can easily adjust posi­tion of TECs it is not harmful to user body. The system also using the GPS, GSM modules, LM-35 temperature sensor and 16X2 LCD screen. The temperature sensor is used to sense the climatic temperature continu­ously and display it on LCD screen.

**A Review of Technology of Personal Heating Garments**

Faming Wang ChuansiGao KalevKuklane Ingvar Holmér at International Journal of Occupational Safety and Ergonomics(JOSE) ,2010 Proposed

Normal thick-layer protective clothing can reduce workers’ risks of getting cold injury when exposed to cold environments. Traditional protective clothing is often bulky and heavy, and can severely limit human movements, dexterity and performance. As a result, traditional protective clothing may be not suitable for those workers who are doing fine work in cold environments. Currently PHGs have some visible drawbacks, e.g., battery performance cannot meet the requirements of long exposure in cold conditions in EHGs; temperature cannot be controlled in chemical heating pads; released latent heat has little effect on the human body in both microcapsulated and packaged PCM heating garments; and liquid/air flow heating systems limit human activities. Compared with the four types of PHGs described in this overview, EHGs are expected to have a promising future. One of the main challenges is to seek new regenerated energy sources such as solar energy, wind energy, sound wave power [78], human motion and garment friction energy, and/or using a temperature gradient (Nansulate® Paint [79]) to generate long-lasting supply of electricity for EHGs. Additionally, each set of conditions must be individually evaluated for possible use of PHGs. The heating system analyses should also include such interacting items as thermal stress, cold stress, task duration, external work intensity, heating system reliability, safety, unit portability and also economic considerations.

**The Things in the Internet of Things**

Stephan Haller presented at Internet of Things Conference in Japan, 2010 Proposed

All the different definitions of the term “Internet of Things“ have in common that it is related to the integration of the physical world with the virtual world of the Internet. There are physical objects one wants to be able to track, to monitor and to interact with. Examples include inanimate objects like pallets, boxes containing consumer goods, cars, machines, fridges and maybe even the infamous carton of milk or cup of yoghurt as well as animate objects like animals and humans. These are the things of the Internet of Things – or to use a clearer term, the entities of interest . Buildings, rooms and things in the environment like rivers and glaciers can also be entities of interest. Basically any object including the attributes that describe it and its state that is relevant from a user or application perspective can be regarded as an entity of interest. In order to monitor and interact with one or more entities and make the connection to the Internet, technical communication devices are required. The devices can be attached to or embedded in the entities themselves – thus creating smart things, or they can be installed in the environment of the things to be monitored. Typical examples of devices include RFID readers, sensors and actuators, embedded computers as well as mobile phones. While there is a school of thought that regards the devices as the things in the Internet of Things, such an approach seems too limited, as businesses and consumers are more interested the physical objects rather than any technical devices needed for monitoring and communication. Having said that, it needs to be noted that devices constitute entities of interest in its own right when looking at them from a technical or management perspective. Thus, devices are a subset of all the things in the Internet of Things. However, for reasons of clarity this case where the thing, the device and the entity of interest are the same should be treated as a special case.

**From the Internet of Computers to the Internet of Things**

Friedman Mattern and Christian Floerkemeier at German computer science pioneer, 1996 Proposed

A vision in which the Internet extends into the real world embracing everyday objects. Physical items are no longer disconnected from the virtual world, but can be controlled remotely and can act as physical access points to Internet services. An Internet of Things makes computing truly ubiquitous a concept initially put forward by Mark Weiser in the early 1990s. This development is opening up huge opportunities for both the economy and individuals. However, it also involves risks and undoubtedly represents an immense technical and social challenge. The Internet of Things vision is grounded in the belief that the steady advances in microelectronics, communications and information technology we have witnessed in recent years will continue into the foreseeable future. In fact due to their diminishing size, constantly falling price and declining energy consumption processors, communications modules and other electronic components are being increasingly integrated into everyday objects today.

**Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being**

ZHIBO PANG at Royal Institute of Technology Stockholm, Sweden, January 2013 Proposed

The emerging technology breakthrough of the Internet-of-Things (IoT) is expected to offer promising solutions for food supply chain (FSC) and in-home healthcare (IHH), which may significantly contribute to human health and well-being. In this thesis, we have investigated the technologies and architectures of the IoT for these two applications as so-called Food-IoT and Health-IoT respectively. We intend to resolve a series of research problems about the WSN architectures, device architectures and system integration architectures. To reduce the time-to-market and risk of failure, business aspects are taken into account more than before in the early stage of technology development because the technologies and applications of IoT are both immature today. The challenges about enabling devices that we have addressed include: the WSN mobility and wide area deployment, efficient data compression in resource-limited wireless sensor devices, reliable communication protocol stack architecture, and integration of acting capacity to the low cost intelligent and interactive packaging (I2Pack). Correspondingly, the WAN-SAN coherent architecture of WSN, the RTOS based and multiprocessor friendly stack architecture, the content-extraction based data compression algorithm, and the CDM-based I2Pack solution are proposed and demonstrated.

**Principle Elements and Framework of Internet of Things**

Bhagyashri Katole at International Journal of Engineering and Science, July 2013 Proposed

The Internet of Things (IoT) represents the future of computing and communications. It is world of information and communication technologies (ICTs) from anytime, anyplace connectivity for anyone; we will now have connectivity for anything. It semantically means a network of interconnected objects that are uniquely addressable and connected using standard communication protocols. It will consist of connections that will multiply and create entirely new dynamic network of networks. In this, objects or things are made as smart so that they will become knowledgeable and their properties such as transformation, interactions will allow them to actively interact in environment. For example, RFID tags, sensors and actuators, NFC devices can be made seamlessly communicated and characterized by properties like modularity, reliability, scalability and robustness.[1]In Internet of Things, objects or things are made uniquely addressable by using unique way of identification. These things are heterogeneous in some capabilities. The IoT will provide them a common environment where these heterogeneous things will be able to communicate with each other using standardized communication platform. Obviously, these things will need to consume their own energy very carefully so that they not only to able to communicate for indefinitely long but also form extensive network even when infrastructure is weak or not available.

**PROPOSED SYSTEM**

The proposed system is a battery power heating and cooling jacket, in which the user can regulate the temperature of the jacket through his mobile. After initialization temperature sensor measures the wearable coat temperature according to person’s surroundings. The jacket to a mobile app via Wi-Fi. The mobile app will display the current body temperature and humidity, we can vary the temperature of the jacket according to user’s need.

**Objective:**

* The usefulness and practicality of such a suit is the motivating factor of embarking on this project.
* Ultimately, we set out to achieve a body suit that is easy to wear, comfortable, and provides simple and adequate controls that allow for any user to utilize it to their needs.
* Its functionalities can be used to prevent the unfortunate condition that are caused by heat stroke, hypothermia, and other thermally induced maladies.

**Scope:**

**Health Care Device**

Suit can be used to monitor the temperature, humidity of the patients in hospitals. The patients can are kept in optimal temperature for optimal recovery. Suit can also be used to for the old people who are susceptible to temperature change.

**Soldiers**

Soldiers generally face extreme Cold and Hot conditions, like the heat of Rajasthan to cold of Siyachin. Our suit can be used to comfort the soldiers in these regions

**REQUIREMENTS ANALYSIS**

Hardware Requirements:

* Raspberry pi
* Peltier Plate
* DHT11 Temperature sensor
* Lithium-ion Polymer Battery
* Relay
* Jacket
* Android Phone

Software Requirements:

* Apache server
* Python
* C
* PHP

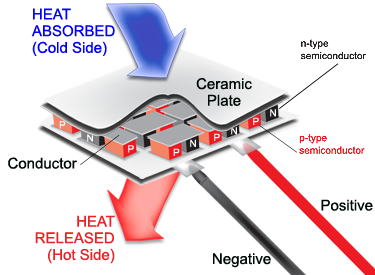
**Raspberry pi**

****Raspberry Pi is a series of credit card–sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation with the intent to promote the teaching of basic computer science in schools and developing countries. The original Raspberry Pi and Raspberry Pi 2 are manufactured in several board configurations through licensed manufacturing agreements with Newark element14 (Premier Farnell), RS Components and Egoman.The hardware is the same across all manufacturers.

Several generations of Raspberry Pi's have been released. The first generation (Pi 1) was released in February 2012 in basic model A and a higher specification model B. A+ and B+ models were released a year later. Raspberry Pi 2 model B was released in February 2015 and Raspberry Pi 3 model B in February 2016. These boards are priced between US$20 and US$35. A cut down compute model was released in April 2014 and a Pi Zero with smaller footprint and limited IO (GPIO) capabilities released in November 2015 for US$5.

All models feature a Broadcom system on a chip (SOC) which includes an ARM compatible CPU and an on chip graphics processing unit GPU (a[VideoCore](https://en.wikipedia.org/wiki/VideoCore) IV). CPU speed range from 700 MHz to 1.2 GHz for the Pi 3 and on board memory range from 256 MB to 1 GB RAMS. Secure Digital SD cards are used to store the operating system and program memory in either the SDHC or MicroSDHC sizes. Most boards have between one and four USB slots, [HDMI](https://en.wikipedia.org/wiki/HDMI) and [composite video](https://en.wikipedia.org/wiki/Composite_video) output, and a 3.5 mm phono jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like [I2C](https://en.wikipedia.org/wiki/I%C2%B2C). Some models have an RJ45 Ethernet port and the Pi 3 has on board WiFi 802.11n and Bluetooth.

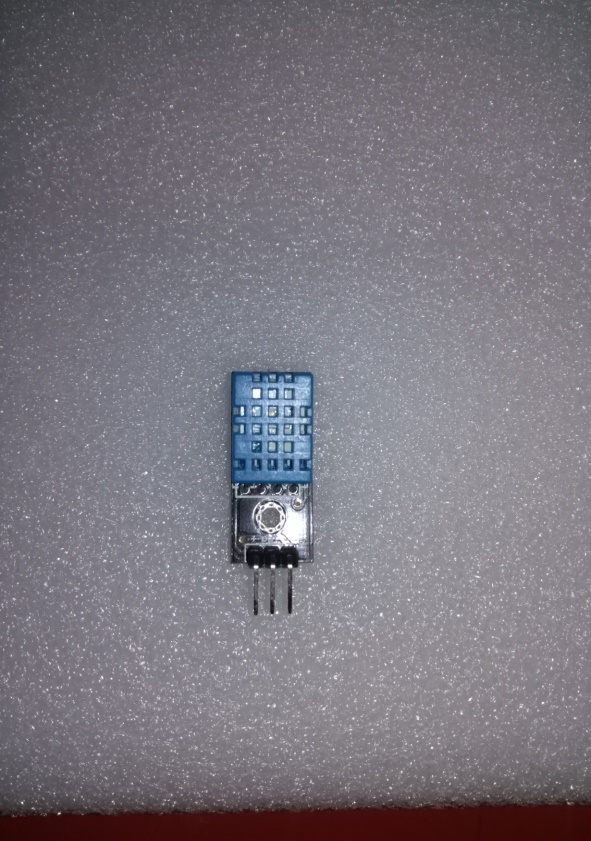
**Peltier Plate**

****

Thermoelectric cooling uses the [Peltier effect](https://en.wikipedia.org/wiki/Peltier_effect) to create a [heat](https://en.wikipedia.org/wiki/Heat) flux between the junctions of two different types of materials. A Peltier cooler, heater, or [thermoelectric](https://en.wikipedia.org/wiki/Thermoelectric) heat pump is a solid-state active [heat pump](https://en.wikipedia.org/wiki/Heat_pump) which transfers heat from one side of the device to the other, with consumption of [electrical energy](https://en.wikipedia.org/wiki/Electrical_energy), depending on the direction of the current. Such an instrument is also called a Peltier device, Peltier heat pump, solid state refrigerator, or thermoelectric cooler (TEC). It can be used either for heating or for cooling, although in practice the main application is cooling. It can also be used as a temperature controller that either heats or cools.

A Peltier cooler can also be used as a [thermoelectric generator](https://en.wikipedia.org/wiki/Thermoelectric_generator). When operated as a cooler, a voltage is applied across the device, and as a result, a difference in temperature will build up between the two sides. When operated as a generator, one side of the device is heated to a temperature greater than the other side, and as a result, a difference in voltage will build up between the two sides (the [Seebeck effect](https://en.wikipedia.org/wiki/Seebeck_effect" \o "Seebeck effect)). However, a well-designed Peltier cooler will be a mediocre thermoelectric generator and vice versa, due to different design and packaging requirements.

**DHT 11 Temperature Sensor**

****This is a multifunctional sensor that gives you temperature and relative humidity information at the same time. It utilizes a DHT11 sensor that can meet measurement needs of general purposes. It provides reliable readings when environment humidity condition in between 20% RH and 90% RH, and temperature condition in between 0°C and 50°C, covering needs in most home and daily applications that don't contain extreme conditions.

**Lithium-ion Polymer Battery**



A lithium polymer battery, or more correctly lithium-ion polymer battery (abbreviated variously as LiPo, LIP, Li-poly and others), is a [rechargeable battery](https://en.wikipedia.org/wiki/Rechargeable_battery) of [lithium-ion](https://en.wikipedia.org/wiki/Lithium-ion) technology in a [pouch format](https://en.wikipedia.org/wiki/Lithium_ion#Shapes). Unlike cylindrical and prismatic cells, LiPos come in a soft package or pouch, which makes them lighter but also less rigid.

The designation "lithium polymer" has caused confusion among battery users because it can be interpreted in two ways. Originally, "lithium polymer" represented a developing technology using a [polymer](https://en.wikipedia.org/wiki/Polymer) [electrolyte](https://en.wikipedia.org/wiki/Electrolyte) instead of the more common liquid electrolyte. The result is a "plastic" cell, which theoretically could be thin, flexible, and manufactured in different shapes, without risk of electrolyte leakage. The technology has not been fully developed and commercialized and research is ongoing.

The second meaning appeared after some manufacturers applied the "polymer" designation to [lithium-ion](https://en.wikipedia.org/wiki/Lithium-ion) cells contained in a non-rigid pouch format. This is currently the most popular use, in which "polymer" refers more to a "polymer casing" (that is, the soft, external container) rather than a "polymer electrolyte". While the design is usually flat, and lightweight, it is not truly a polymer cell, since the electrolyte is still in liquid form, although it may be "plasticized" or "gelled" through a polymer additive. These cells are sometimes designated as "LiPo"; however, from a technological point of view, they are the same as the ones marketed simply as "[Li-ion](https://en.wikipedia.org/wiki/Lithium-ion)", since the underlying [electrochemistry](https://en.wikipedia.org/wiki/Electrochemistry) is the same.

**Apache Server**

Tomcat is a Java servlet container and web server from the Jakarta project of the Apache Software Foundation (http://jakarta.apache.org). A web server is, of course, the program that dishes out web pages in response to requests from a user sitting at a web browser. But web servers aren’t limited to serving up static HTML pages; they can also run programs in response to user requests and return the dynamic results to the user’s browser. This is an aspect of the web that Apache’s Tomcat is very good at because Tomcat provides both Java servlet and Java Server Pages (JSP) technologies (in addition to traditional static pages and external CGI programming). The result is that Tomcat is a good choice for use as a web server for many applications. And it’s a very good choice if you want a free, open source (http://opensource.org/) servlet and JSP engine.

Tomcat can be used stand-alone, but it is often used “behind” traditional web servers such as Apache http, with the traditional server serving static pages and Tomcat serving dynamic servlet and JSP requests.

No matter what we call Tomcat, a Java servlet container or servlet and JSP engine, we mean Tomcat provides an environment in which servlets can run and JSP can be processed. Similarly, we can absolutely say a CGI-enabled Web server is a CGI program container or engine since the server can accommodate CGI programs and communicate with them according to CGI speciﬁcation. Between Tomcat and the servlets and JSP code residing on it, there is also a standard regulating their interaction, servlet and JSP speciﬁcation, which is in turn a part of Sun’s J2EE (Java 2 Enterprise Edition).

**Python**

Python is a widely used [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)), [dynamic programming language](https://en.wikipedia.org/wiki/Dynamic_programming_language). Its design philosophy emphasizes code [readability](https://en.wikipedia.org/wiki/Readability), and its syntax allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than would be possible in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B) or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)).The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming) or [procedural](https://en.wikipedia.org/wiki/Procedural_programming" \o "Procedural programming)styles. It features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management) and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system), allowing Python code to run on a wide variety of systems. Using [third-party](https://en.wikipedia.org/wiki/Third-party_software_component) tools, such as [Py2exe](https://en.wikipedia.org/wiki/Py2exe) or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, so Python-based software can be distributed to, and used on, those environments with no need to install a Python interpreter.

[CPython](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation).

Most Python implementations (including CPython) can function as a [command line interpreter](https://en.wikipedia.org/wiki/Command_line_interpreter), for which the user enters statements sequentially and receives the results immediately ([read–eval–print loop](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) (REPL)). In short, Python acts as a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) or [shell](https://en.wikipedia.org/wiki/Shell_(computing)).

Other shells add abilities beyond those in the basic interpreter, including [IDLE](https://en.wikipedia.org/wiki/IDLE_(Python)) and [IPython](https://en.wikipedia.org/wiki/IPython). While generally following the visual style of the Python shell, they implement features like auto-completion, session state retention, and syntax highlighting.

The main Python implementation, named [CPython](https://en.wikipedia.org/wiki/CPython), is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) meeting the [C89](https://en.wikipedia.org/wiki/C89_(C_version)) standard. It compiles Python programs into intermediate [bytecode](https://en.wikipedia.org/wiki/Bytecode" \o "Bytecode), which is executed by the [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine). CPython is distributed with a large standard library written in a mixture of C and Python. It is available in versions for many platforms, including [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows)and most modern [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems. CPython was intended from almost its very conception to be cross-platform.

**C Programming Language**

**C** ([/ˈsiː/](https://en.wikipedia.org/wiki/Help:IPA_for_English), as in [the letter *c*](https://en.wikipedia.org/wiki/C)) is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language), [imperative](https://en.wikipedia.org/wiki/Imperative_programming) computer [programming language](https://en.wikipedia.org/wiki/Programming_language), supporting [structured programming](https://en.wikipedia.org/wiki/Structured_programming), [lexical variable scope](https://en.wikipedia.org/wiki/Lexical_variable_scope) and [recursion](https://en.wikipedia.org/wiki/Recursion_(computer_science)), while a [static type system](https://en.wikipedia.org/wiki/Static_type_system) prevents many unintended operations. By design, C provides constructs that map efficiently to typical [machine instructions](https://en.wikipedia.org/wiki/Machine_instruction), and therefore it has found lasting use in applications that had formerly been coded in [assembly language](https://en.wikipedia.org/wiki/Assembly_language), including [operating systems](https://en.wikipedia.org/wiki/Operating_system), as well as various [application software](https://en.wikipedia.org/wiki/Application_software) for computers ranging from [supercomputers](https://en.wikipedia.org/wiki/Supercomputer) to [embedded systems](https://en.wikipedia.org/wiki/Embedded_system).

C was originally developed by [Dennis Ritchie](https://en.wikipedia.org/wiki/Dennis_Ritchie) between 1969 and 1973 at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs), and used to re-implement the [Unix](https://en.wikipedia.org/wiki/Unix) operating system. It has since become one of the [most widely used](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) programming languages of all time, with C [compilers](https://en.wikipedia.org/wiki/Compiler) from various vendors available for the majority of existing [computer architectures](https://en.wikipedia.org/wiki/Computer_architecture) and operating systems. C has been standardized by the [American National Standards Institute](https://en.wikipedia.org/wiki/American_National_Standards_Institute) (ANSI) since 1989 (see [ANSI C](https://en.wikipedia.org/wiki/ANSI_C)) and subsequently by the [International Organization for Standardization](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO).

C is an [imperative](https://en.wikipedia.org/wiki/Imperative_programming) ([procedural](https://en.wikipedia.org/wiki/Procedural_programming)) language. It was designed to be compiled using a relatively straightforward [compiler](https://en.wikipedia.org/wiki/Compiler), to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal [run-time support](https://en.wikipedia.org/wiki/Run-time_system). C was therefore useful for many applications that had formerly been coded in assembly language, such as in [system programming](https://en.wikipedia.org/wiki/System_programming).

Despite its low-level capabilities, the language was designed to encourage [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) programming. A standards-compliant and [portably](https://en.wikipedia.org/wiki/Porting) written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller) to [supercomputers](https://en.wikipedia.org/wiki/Supercomputer).

**PHP**

**PHP** is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf" \o "Rasmus Lerdorf) in 1994, the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group. PHP originally stood for *Personal Home Page*, but it now stands for the [recursive](https://en.wikipedia.org/wiki/Recursive_acronym) [backronym](https://en.wikipedia.org/wiki/Backronym" \o "Backronym) *PHP: Hypertext Preprocessor*.

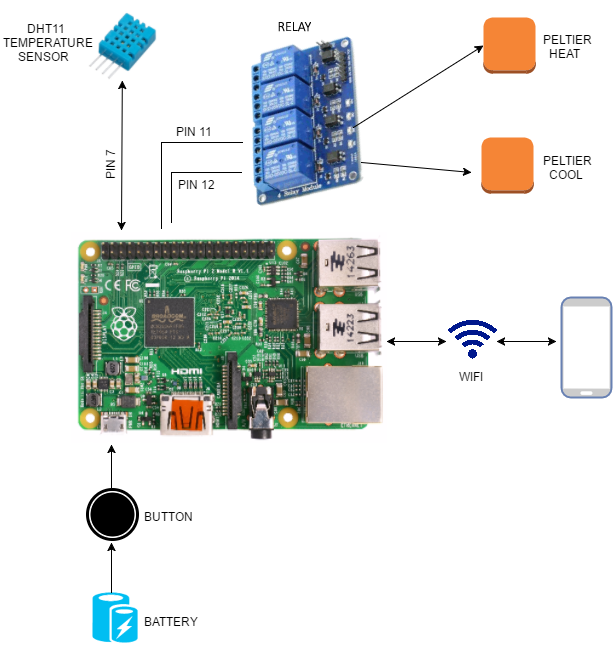
PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web content management system and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a[module](https://en.wikipedia.org/wiki/Plugin_(computing)) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).

The standard PHP interpreter, powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine" \o "Zend Engine), is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and [platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge.

The PHP language evolved without a written [formal specification](https://en.wikipedia.org/wiki/Formal_specification) or standard until 2014, leaving the canonical PHP interpreter as a [*de facto*](https://en.wikipedia.org/wiki/De_facto) standard. Since 2014 work has gone on to create a formal PHP specification.

During the 2010s there have been increased efforts towards standardisation and code sharing in PHP applications by projects such as [PHP-FIG](http://www.php-fig.org/) in the form of [PSR-initiatives](http://www.php-fig.org/psr/) as well as [Composer dependency manager](https://en.wikipedia.org/wiki/Composer_(software)) and the [Package.](https://packagist.org/)

**PROPOSED ARCHITECTURE**

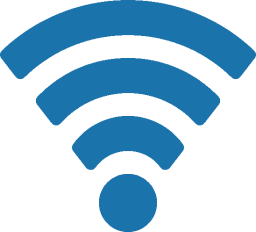
****

****

As shown in above figure, powers the raspberry and the suit which can be controlled by button.Mobile is connected to Raspberry via WiFi module. Temperature sensor DHT11 is used to assess the body temperature.The heating and cooling functionality of the peltier plate can be controlled through replays.

**BLOCK DIAGRAM FOR IoT CONNECTIVITY**

****

**** 

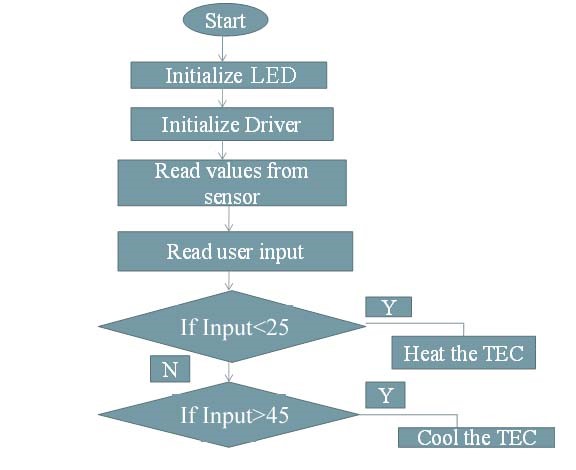
Raspberry

Pi

The above figure shows IoT Connectivity for the suit. The mobile is connected to the raspberry pi of the jacket via the WiFi module. The commands given by the user is transferred via WiFi to the raspberry pi, which processes the request of the users and executes it.



**FLOW CHART FOR HEATING AND COOLING MECAHNISM**



The above figure represents the flow diagram of the heating and cooling mechanism, once the suit is powered on the drivers get initialized, values of the temperature and humidity sensor are read, if the temperature of suit is less then user input, then heating process starts, else if the temperature of suit is greater then user input, then cooling process starts.

**TEST IMPLEMENTATION INTERPRETATION**

**SOURCECODE**

**IoT Code:**

Gpio.php

<?php

//This page is requested by the JavaScript, it updates the pin's status and then print it

//Getting and using values

if (isset ( $\_GET["pic"] )) {

$pic = strip\_tags ($\_GET["pic"]); //strip the strings from php tags

//test if value is a number

if ( (is\_numeric($pic)) && ($pic <= 7) && ($pic >= 0) ) {

//set the gpio's mode to output

system("gpio mode ".$pic." out");

//reading pin's status

exec ("gpio read ".$pic, $status, $return );

//set the gpio to high/low

if ($status[0] == "0" ) { $status[0] = "1"; }

else if ($status[0] == "1" ) { $status[0] = "0"; }

system("gpio write ".$pic." ".$status[0] );

//reading pin's status

exec ("gpio read ".$pic, $status, $return );

//print it to the client on the response

echo($status[0]);

}

else { echo ("fail"); }

} //print fail if cannot use values

else { echo ("fail"); }

?>

**Index.php**

<!DOCTYPE html>

// <html>

// <head>

// <meta charset="utf-8" />

// <title>Raspberry Pi Gpio</title>

// </head>

// <body style="background-color: black;">

// <!-- On/Off button's picture -->

// <?php

// $val\_array = array(0,0,0,0,0,0,0,0);

this php script generate the first page in function of the file

// for ( $i= 0; $i<8; $i++) {

set the pin's mode to output and read them

// system("gpio mode ".$i." out");

// exec ("gpio read ".$i, $val\_array[$i], $return );

// }

for loop to read the value

// $i =0;

// for ($i = 0; $i < 8; $i++) {

if off

// if ($val\_array[$i][0] == 0 ) {

// echo ("<img id='button\_".$i."' src='data/img/red/red\_".$i.".jpg' onclick='change\_pin (".$i.");'/>");

// }

if on

// if ($val\_array[$i][0] == 1 ) {

// echo ("<img id='button\_".$i."' src='data/img/green/green\_".$i.".jpg' onclick='change\_pin (".$i.");'/>");

// }

// }

// ?>

// <!-- javascript -->

// <script src="script.js"></script>

// </body>

// </html>

**Temperature sensor code:**

**C code:**

#include <wiringPi.h>

#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#define MAXTIMINGS 85

#define DHTPIN 7

int dht11\_dat[5] = { 0, 0, 0, 0, 0 };

void read\_dht11\_dat()

{

uint8\_t laststate = HIGH;

uint8\_t counter = 0;

uint8\_t j = 0, i;

float f; /\* fahrenheit \*/

dht11\_dat[0] = dht11\_dat[1] = dht11\_dat[2] = dht11\_dat[3] = dht11\_dat[4] = 0;

/\* pull pin down for 18 milliseconds \*/

pinMode( DHTPIN, OUTPUT );

digitalWrite( DHTPIN, LOW );

delay( 18 );

/\* then pull it up for 40 microseconds \*/

digitalWrite( DHTPIN, HIGH );

delayMicroseconds( 40 );

/\* prepare to read the pin \*/

pinMode( DHTPIN, INPUT );

/\* detect change and read data \*/

for ( i = 0; i < MAXTIMINGS; i++ )

{

counter = 0;

while ( digitalRead( DHTPIN ) == laststate )

{

counter++;

delayMicroseconds( 1 );

if ( counter == 255 )

{

break;

}

}

laststate = digitalRead( DHTPIN );

if ( counter == 255 )

break;

/\* ignore first 3 transitions \*/

if ( (i >= 4) && (i % 2 == 0) )

{

/\* shove each bit into the storage bytes \*/

dht11\_dat[j / 8] <<= 1;

if ( counter > 16 )

dht11\_dat[j / 8] |= 1;

j++;

}

}

/\*

\*check we read 40 bits (8bit x 5 ) + verify checksum in the last byte

\* print it out if data is good

\*/

if ( (j >= 40) &&

(dht11\_dat[4] == ( (dht11\_dat[0] + dht11\_dat[1] + dht11\_dat[2] + dht11\_dat[3]) & 0xFF) ) )

{

f = dht11\_dat[2] \* 9. / 5. + 32;

printf( "Humidity = %d.%d %% Temperature = %d.%d \*C (%.1f \*F)\n",

dht11\_dat[0], dht11\_dat[1], dht11\_dat[2], dht11\_dat[3], f );

}

else {

printf( "Data not good, skip\n" );

}

}

int main( void )

{

printf( "Raspberry Pi wiringPi DHT11 Temperature test program\n" );

if ( wiringPiSetup() == -1 )

exit( 1 );

while ( 1 )

{

read\_dht11\_dat();

delay( 1000 ); /\* wait 1sec to refresh \*/

}

return(0);

}

**PYTHON Code**

**AdafruitDHT.py**

#!/usr/bin/python

# Copyright (c) 2014 Adafruit Industries

# Author: Tony DiCola

# Permission is hereby granted, free of charge, to any person obtaining a copy

# of this software and associated documentation files (the "Software"), to deal

# in the Software without restriction, including without limitation the rights

# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

# copies of the Software, and to permit persons to whom the Software is

# furnished to do so, subject to the following conditions:

# The above copyright notice and this permission notice shall be included in all

# copies or substantial portions of the Software.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

# SOFTWARE.

import sys

import MySQLdb

import Adafruit\_DHT

# Open database connection

db = MySQLdb.connect("103.247.99.31","edutech","EduTech@2016","temperature",3309 )

# prepare a cursor object using cursor() method

cursor = db.cursor()

# Parse command line parameters.

sensor\_args = { '11': Adafruit\_DHT.DHT11,

'22': Adafruit\_DHT.DHT22,

'2302': Adafruit\_DHT.AM2302 }

if len(sys.argv) == 3 and sys.argv[1] in sensor\_args:

sensor = sensor\_args[sys.argv[1]]

pin = sys.argv[2]

else:

print('usage: sudo ./Adafruit\_DHT.py [11|22|2302] GPIOpin#')

print('example: sudo ./Adafruit\_DHT.py 2302 4 - Read from an AM2302 connected to GPIO #4')

sys.exit(1)

# Try to grab a sensor reading. Use the read\_retry method which will retry up

# to 15 times to get a sensor reading (waiting 2 seconds between each retry).

humidity, temperature = Adafruit\_DHT.read\_retry(sensor, pin)

# Un-comment the line below to convert the temperature to Fahrenheit.

# temperature = temperature \* 9/5.0 + 32

# Note that sometimes you won't get a reading and

# the results will be null (because Linux can't

# guarantee the timing of calls to read the sensor).

# If this happens try again!

if humidity is not None and temperature is not None:

sql = "UPDATE temperature SET temperature = '%f' WHERE id=1" % (temperature)

try:

# Execute the SQL command

cursor.execute(sql)

# Commit your changes in the database

db.commit()

print 11

except:

print "error"

# Rollback in case there is any error

db.rollback()

print('Temp = {0:0.1f} \* Humidity = {1:0.1f}%'.format(temperature, humidity))

else:

print('Failed to get reading. Try again!')

sys.exit(1)

db.close()

**App Source code**

**AndroidManifest.Xml**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="jacket " >

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity android:name=".Splashscreen"

android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name=".MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</activity>

</application>

<uses-permission android:name="android.permission.INTERNET" />

</manifest>

**Main Activity.java**

Package jacket;

import android.app.Activity;

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.webkit.WebSettings;

import android.webkit.WebView;

import android.webkit.WebViewClient;

public class MainActivity extends Activity {

private WebView mWebView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mWebView = (WebView) findViewById(R.id.activity\_main\_webview);

WebSettings webSettings = mWebView.getSettings();

webSettings.setJavaScriptEnabled(true);

mWebView.loadUrl("http://www.instazop.com");

mWebView.setWebViewClient(new MyAppWebViewClient());

}

@Override

public void onBackPressed(){

if (mWebView.canGoBack()){

mWebView.goBack();

}

else {

super.onBackPressed();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

private class MyAppWebViewClient extends WebViewClient {

}

}

**MyAppWebViewClient.java**

Package jacket;

import android.content.Intent;

import android.net.Uri;

import android.webkit.WebView;

import android.webkit.WebViewClient;

public class MyAppWebViewClient extends WebViewClient {

@Override

public boolean shouldOverrideUrlLoading(WebView view, String url){

if (Uri.parse(url).getHost().endsWith("instazop.com")){

return false;

}

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse(url));

view.getContext().startActivity(intent);

return true;

}

}

**SplashScreen.java**

Package jacket;

import android.app.Activity;

import android.content.Intent;

import android.graphics.PixelFormat;

import android.os.Bundle;

import android.view.Window;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.widget.LinearLayout;

public class Splashscreen extends Activity {

public void onAttachedToWindow() {

super.onAttachedToWindow();

Window window = getWindow();

window.setFormat(PixelFormat.RGBA\_8888);

}

/\*\* Called when the activity is first created. \*/

Thread splashTread;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_splashscreen);

StartAnimations();

}

private void StartAnimations() {

Animation anim = AnimationUtils.loadAnimation(this, R.anim.alpha);

anim.reset();

LinearLayout l=(LinearLayout) findViewById(R.id.lin\_lay);

l.clearAnimation();

l.startAnimation(anim);

anim = AnimationUtils.loadAnimation(this, R.anim.translate);

anim.reset();

ImageView iv = (ImageView) findViewById(R.id.splash);

iv.clearAnimation();

iv.startAnimation(anim);

splashTread = new Thread() {

@Override

public void run() {

try {

int waited = 0;

// Splash screen pause time

while (waited < 3500) {

sleep(100);

waited += 100;

}

Intent intent = new Intent(Splashscreen.this,MainActivity.class);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NO\_ANIMATION);

startActivity(intent);

Splashscreen.this.finish();

} catch (InterruptedException e) {

// do nothing

} finally {

Splashscreen.this.finish();

}

}

}

splashTread.start();

}

}

**ApplicationTest.java**

package jacket;

import android.app.Application;

import android.test.ApplicationTestCase;

/\*\*

\*<a href="http://d.android.com/tools/testing/testing\_android.html">Testing Fundamentals</a>

\*/

public class ApplicationTest extends ApplicationTestCase<Application> {

public ApplicationTest() {

super(Application.class);

}

}

**Alpha.xml**

<?xml version="1.0" encoding="utf-8"?>

<alpha

xmlns:android="http://schemas.android.com/apk/res/android"

android:fromAlpha="0.0"

android:toAlpha="1.0"

android:duration="3000" />

**Translate.xml**

<?xml version="1.0" encoding="utf-8"?>

<set

xmlns:android="http://schemas.android.com/apk/res/android">

<translate

xmlns:android="http://schemas.android.com/apk/res/android"

android:fromXDelta="0%"

android:toXDelta="0%"

android:fromYDelta="200%"

android:toYDelta="0%"

android:duration="2000"

android:zdjustment="top" />

</set>

**Activity\_main.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

**Activity\_splashscreen.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#FFFFFF"

android:layout\_gravity="center"

android:id="@+id/lin\_lay"

android:gravity="center"

android:orientation="vertical" >

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/splash"

android:background="@drawable/splash\_img" />

</LinearLayout>

**Manu\_main.xml**

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools" tools:context=".MainActivity">

<item android:id="@+id/action\_settings" android:title="@string/action\_settings"

android:orderInCategory="100" app:showAsAction="never" />

</menu>

**Dimens.xml**

<resources>

<!-- Default screen margins, per the Android Design guidelines. -->

<dimen name="activity\_horizontal\_margin">16dp</dimen>

<dimen name="activity\_vertical\_margin">16dp</dimen>

</resources>

**Strings.xml**

<resources>

<string name="app\_name">Homeautomation</string>

<string name="hello\_world">Hello world!</string>

<string name="action\_settings">Settings</string>

</resources>

**Styles.xml**

\<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

<!-- Customize your theme here. -->

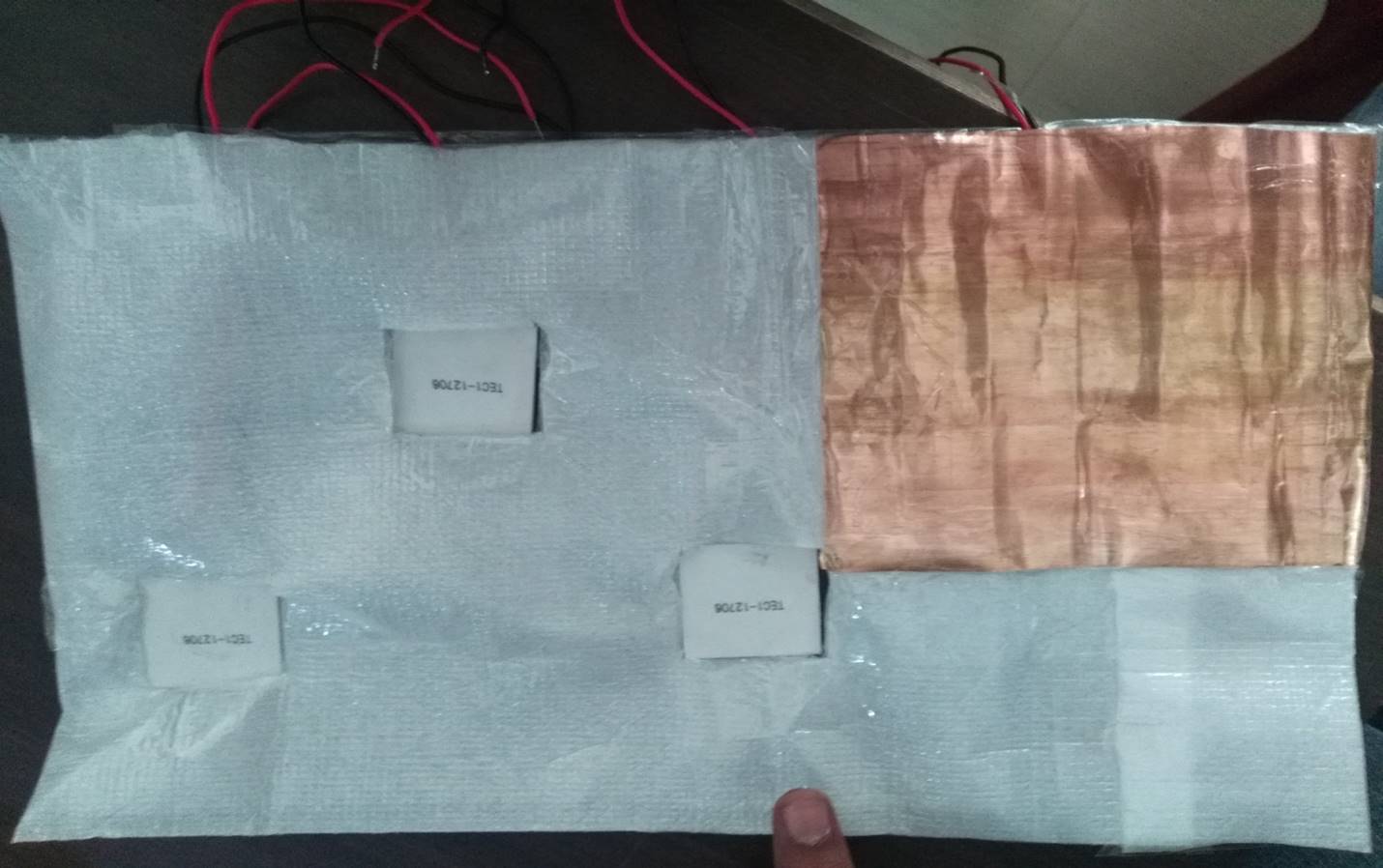
</style>

</resources>

**RESULT AND DISCUSSION**



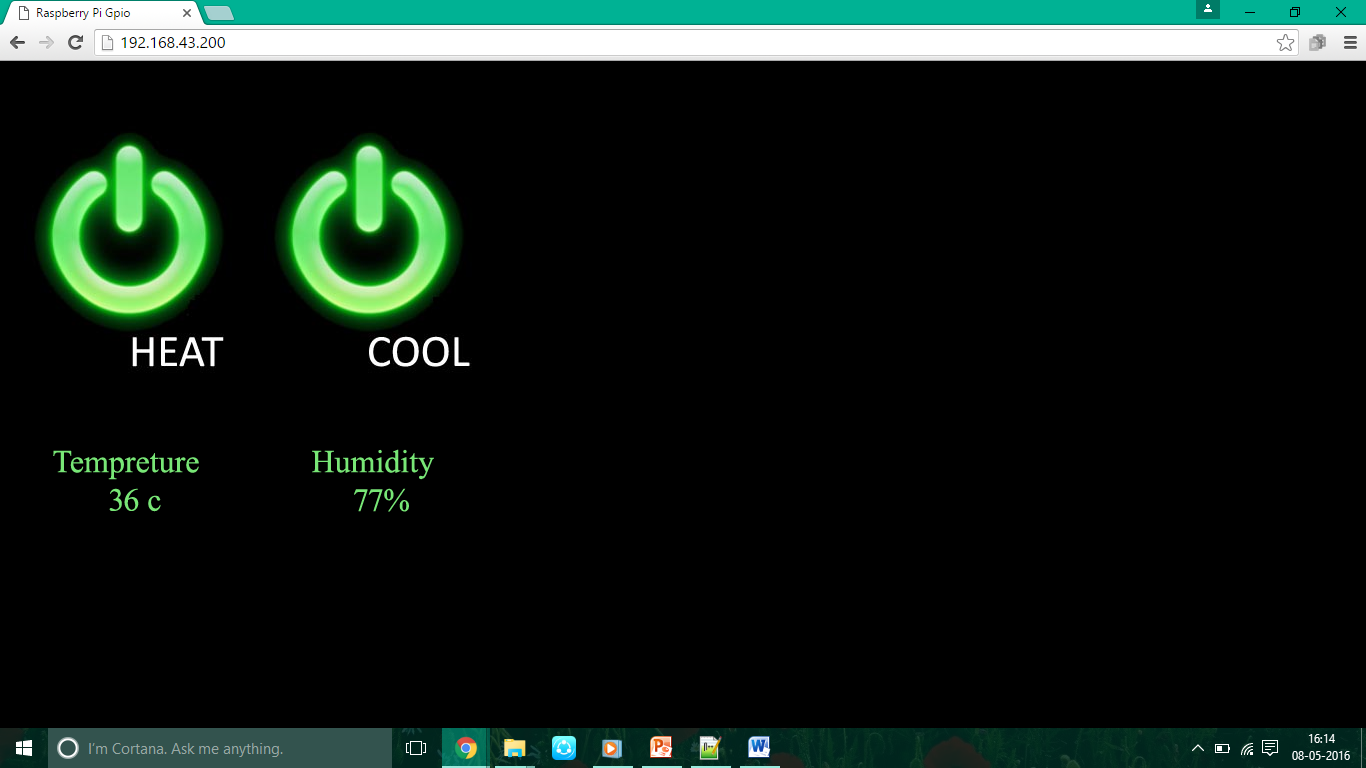
The above figure represents the electronic part of the suit,were the mobile app operates the heating and cooling operation.

****

The above figure represents the modeling of the peltier plate inside the suit, LDPE is used as an thermal insulator and copper sheet is used for transfer of heat across the body.



**The above figure represents cooling unit of the suit, the peltier plate is cooled using cpu cooler.**

****

The above figure represents user interface, were the user can control the heating and cooling functionality of the suit. The user can see the temperature and humidity of the suit.

**ADVANTAGES**

* The jacket can be easily controlled by a mobile app.
* As our electronic gadget runs on battery, it is a portable and easy to use.
* The jacket is easy to wash, as the electrical parts detachable.
* It can be used to people who are exposed to the scorching summer sun like the policeman or the industrial worker whose work environment is often a high-temperature one.
* Portable jacket will help us to have an optimal Temperature all the time.
* Suit can be used to monitor the temperature, humidity of the patients in hospitals.
* Suit can also be used to for the old people who are susceptible to temperature change.
* Soldiers generally face extreme Cold and Hot conditions, the suit can be used to comfort the soldiers in these regions.

**DISADVANTAGES**

* Limited battery resource.
* Slow cooling action
* Jacket in operation for two hours

**CONCLUSION AND FUTURE SCOPE**

In this paper gadget with multi features for different applications describes the integration of new technologies, offering ease of maintenance. By implementing this project the wearable, washable and portable jacket is used to monitor and maintain the body temperature conditions of user according to surrounding environments of that user. And also we can enhance this project in future for medical applications and to track the person with the help GPS.

The future scope of our project is to make it washable, light weight. Decrease the weight of the equipment’s and jackets much further adding pulse monitoring system and GPS location system to the implement. Decreasing the cost of equipment used and make it affordable to the common users.

**REFERENCES**

1.American Society for Testing and Materials (ASTM). Standard test method for measuring the performance of personal cooling systems using physiological testing (Standard No. ASTM F 2300-05). West Conshohocken, PA, USA: ASTM International; 2005.

2. Beecham Research, “Business Opportunities from Remote Device Management: M2M and the Internet of Things”, 2008.

3. Battery powered heating and cooling suit “paper from <http://ieeexplore.ieee.org/xpl/articleDetails.jsp>

4. Cool Vest with 3 portable reservoir options for hot and humid days-stay dry & keep cool!." Veskimo Personal Cooling Systems.N.p., n.d. Web. 13 Dec. 2013.

5. Goldsmid, H. J. ."Timeliness in the development of thermoelectric cooling." IEEE Xplore. N.p., 18 Aug. 1998. Web. 13 Dec. 2013.

6. McManis, Chuck. "H-Bridge Theory & Practice - Chuck's Robotics Notebook."H-Bridge Theory & Practice - Chuck's Robotics Notebook.N.p., n.d. Web. 13 Dec. 2013.

7. "Milwaukee Heated Jacket." Review.N.p., n.d. Web. 13 Dec. 2013.

8. Peltier effect (physics)."Encyclopedia Britannica Online. Encyclopedia Britannica, n.d. Web. 13 Dec. 2013.

9. Rowe , D. M. Thermoelectrics Handbook: Macro to Nano. . New York: Taylor & Francis., 2006. Print.

10. Sieben, Vincent ."A High Power H bridge."Engineering Society.Version 1.N.p., 8 Sept. 2003. Web. 13 Dec. 2013.

11. "SN754410 Datasheet." Texas Instruments.N.p., n.d. Web. 12 Dec.2013.

12. "THERMOVELOCITY PROTECTION." GerbingThermovelocity Protection.N.p., n.d. Web. 11 Dec. 2013.

13. "TEC1-12706 Datasheet." HEB. N.p., n.d. Web. 13 Dec. 2013