Manjil Basnet

Intelligent Nest box using Particle IoT Device

Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Thesis

November 2019

Author Title Number of Pages Date	Manjil Basnet Intelligent Nest box using Particle IoT Device 54 pages + 3 appendices 4th December 2019
Degree	Bachelor of Engineering
Degree Programme	Degree Programme in Electronics
Professional Major	Electronics
Instructors	Matti Fischer, Principal Lecturer

The purpose of this project was to design and implement a module capable of recording as well as displaying the conditions happening inside the nest box by using the particle IoT device in coordination with some sensors. The main objective behind this project was to analyze the day to day activities and obstacles that will be beneficial to ornithologists and birdwatchers.

This nest box module was constructed using the various kind of sensors which will provide analog or digital information to our IoT device i.e. Electron about the physical phenomenon happening inside or near the nest box. Sensors such as DHT22, TMP36, MQ135, etc. read the data around the nest box and these data will be transmitted to our main board (electron) which works as CPU. This electron was linked with the provided sensors with C/C++ code in the Particle IDE and later connected with some phones using the GSM services that existed in the electron. Finally, the output data provided by electron through the GSM network will be displayed on the cell phone with the help of the mobile application called Blynk.

The nest box was tested to analyze how it performs under the new particle device called Electron. The result showed that it is applicable to acknowledge the conditions around the nest box using electron and the desired sensors in the availability of the GSM network.

Keywords	IoT, Electron, Blynk, GSM

Contents

Appendices

List of Figures

List of Tables

List of Listings

List of Abbreviations

IoT - Internet of Things

GPS - Global Positioning System

RTD - Resistance Temperature Detectors

NTC - Negative-Temperature Coefficient

PTC - Positive-Temperature Coefficient

RH - Relative Humidity

NEP - Noise Equivalent Parameter

PWM - Pulse Width Modulation

IDE - Integrated Development Environment

VI - Virtual Instrument

RF - Radio Frequency

RTOS - Real Time Operating System

USB - Universal Serial Bus

GSM – Global System for Mobile communication

1 Introduction

Undoubtedly, this is the world of science and technology. Science and technology are playing a vital role in our everyday life by aiding in every field of our needs. They had helped us in gaining knowledge of our mother earth to the multiverses which are far beyond our ordinary imagination. Apart from many branches from this vast science and technology, Electronics has been assisting this human world to make this place better and scientific.

Technically, the study of the flow and control of electrons(electricity) including the study of their behavior and effects in all medium from solid to gases as well as semiconductors can be termed as Electronics. In modern world, the achievements and the progress obtained through electronics had played a vital role to make our life systematic and scientific. Life without electronics cannot be imagined in this technology dominated world. In the department of research and development, electronics plays a significant role. With the advancement in electronics and its various fields like IoT, along with human, various living creatures day to day activities can be studied for their betterment.

Among various creatures from this world, electronics has played a fundamental role in the study of birds too. Unlike human, birds are also a part of our ecosystem so their livelihood must be studied and observed for their fruitful existence. Ornithologist are the researcher of birds who study the behavior of the birds in order to know their adaptive features, the problems they are facing and the solutions for their problems. Except ornithologist, there are lots of people heavily interested on birds. This thesis will be beneficial for those ornithologists and those birdwatchers in a sense that we are constructing a bird nest using some IoT devices in order to know their livelihood.

The main objective behind this thesis project is to design a box or a device namely Nest box which will be equipped with some sensors and some IoT device in order to study and investigate the physical environments around the nest like temperature, humidity, carbon dioxide level etc. This project will be beneficial for the Ornithologists and birdwatchers in a way that they can observe the activities of their desired bird with the help of some smartphones or laptops without leaving their comfort zone.

2 Nest Box

Simply, nest is the place where birds live like houses are for humans. The nest box is the man-made cell to assist some desired birds to nest in. The box comprising the essential factors for the birds to help them dwell inside can be termed as nest box and even known as birdbox or birdhouse too. The nest box was introduced by the British conservationist Charles Waterton in the early 19th century to encourage more birdlife and wildfowl. [1]

There are lots of bird’s species which are endangered by now and some of them are Giant Ibis, Bengal Florican, California Condor, Kagu, Philippine Eagle etc. Thousands of birds around the world are currently experiencing rapid declines each day. Due to the deforestation, climate change and other factors birds are declining in numbers and many of them are in serious case of extinction too. In order to conserve such birds, we need to observe them every day without harming them so there arises a need of nest box. The main reason behind the nest box is to protect and provide the good shelter for the birds. Nest box can be of various design from wooden to metallic depending upon the birds we are observing.

A lot of researches have been carried around the world for the conservation of birds. Many of those researches are being conducted with the help of IoT devices and sensors like in Belgium many mini projects were executed by using Raspberry pi and some sensors to monitor and study some of the bird’s species. A research institute in Spain had initiated a project similar like ours in order to develop a system for real time monitoring of falcons and was named as “smart nest-boxes”. Czech University of life science have been collaborating with Czech University of technical study in order to carry out many projects on bird’s life using microcontrollers followed by sensors. Lots of researches on birds are being carried in the world in coordination with IoT devices and sensors.

2.1 Working Principle of Nest Box

This thesis purely depends upon the reliability and accuracy of the sensors and the output data provided by the microcontroller, but consideration should be taken on its cost as well as the installation procedure which must be simple and easy to follow.

First, we must choose the sensors depending upon their accuracy, operating conditions, size, input-output type, cost and other properties which are vital for our research. After selection of the sensors we need to select the microcontroller which in our case is Photon by particle devices followed by Electron. Particle devices have their own Ideal Development Environment for programming where our code will be executed in order to receive some data from the sensors and as well to provide output towards some application like Blynk or Tinker. A flow chart is designed as figure 1. to illustrate the basic plan of the project where the interconnection between the sensors, IoT devices and its programming is clearly displayed.

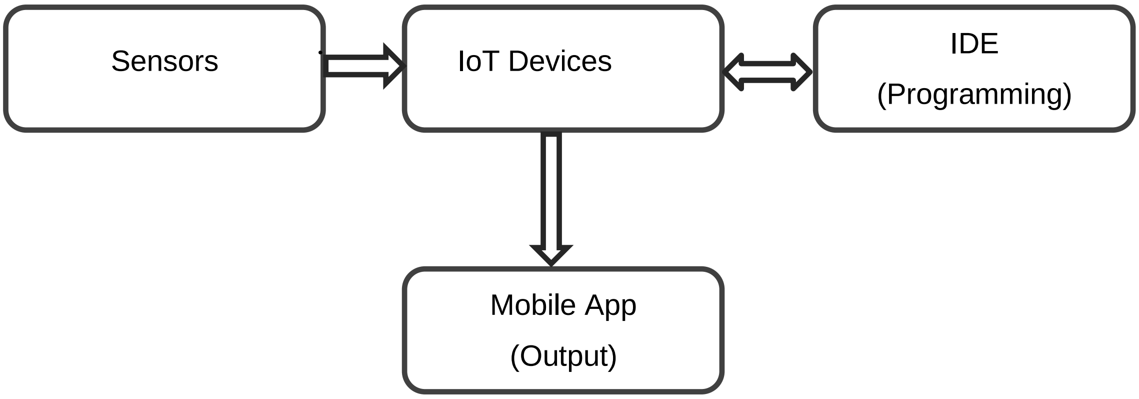


Figure 1. Working Principle of the project

2.2 Design of Nest Box

The nest box should be designed in a way where it must contain all the essential sensors embedded inside the nest box with microcontroller in it. The sensors should be chosen which will be effective for the given project regarding performance, cost, environment friendly etc. As well as it should be constructed in a way that general people can conveniently install and even, they should be able to read and study the results. Considerations must be taken while choosing the microcontroller as we have Photon as well as Electron. Nest box should be designed as necessity of the birdwatchers or ornithologists depending upon the cellular data or the Wi-Fi services. It is concluded that for shorter distances Photon will be effective but for longer ranges Electron will be preferable. One of the important factors to be considered must be power supply depending upon the devices like sensors or microcontroller. Power supply must be arranged continuously or can be arranged partly like at the time of study. As well as sensors and microcontrollers must be chosen according to the altitudes or the temperature of the environment where our project is being carried out. A simple design of the project is pictured below as figure 2. where a clear view of the data transmission and design is displayed.

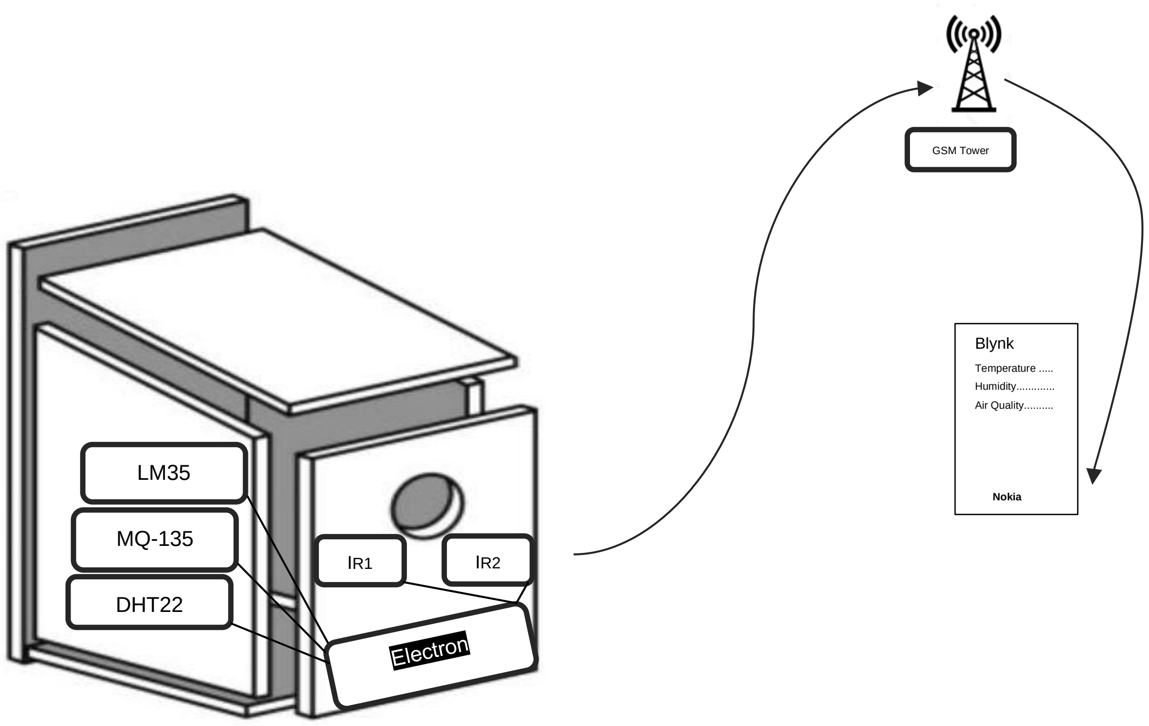


Figure 2. Simple design of Nest Box

3 Sensor

A sensor can be defined as a gadget or some modules which are capable to acknowledge the changes happening in its surrounding, and simultaneously providing a corresponding output. [2]

A physical parameter (temperature, humidity, speed etc.) will be converted into a signal which can be calculated electrically, and the conversion of this physical parameters into signal are possible through the help of sensors. Lots of sensor had been introduced till date. A device without sensors is out of imagination nowadays. Sensors are being used in every field of this modern world. An overview of its application is shown in figure 3.

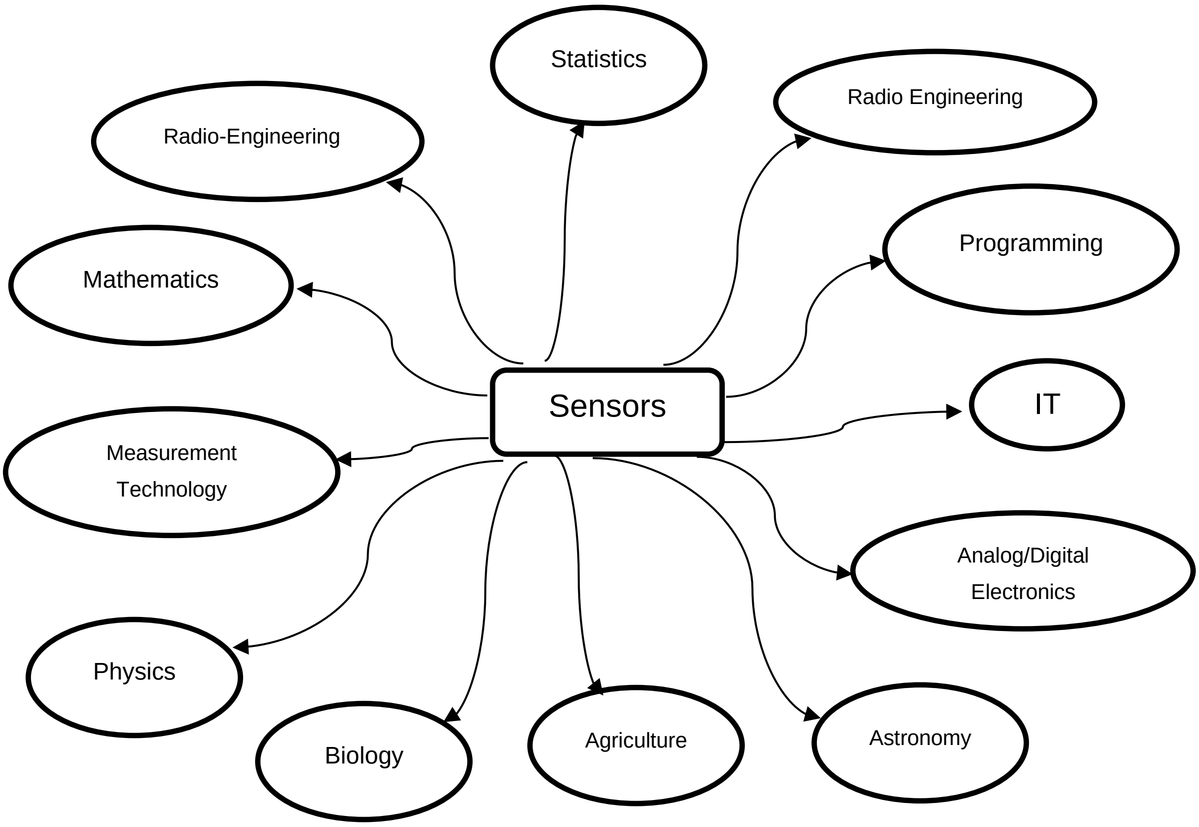


Figure 3. Application of sensors

3.1 Classification of Sensors

Sensors can be classified from different aspects. Sensors can be classified according to their primary input quantity, material, and technology, property, application, etc. As well as the transduction principle had played a fundamental role during the classification of the sensors.

Classification or differentiation of sensors in terms of property is demonstrated below with some examples. In terms of property for classification of sensors, the primary and one of the useful sensors is a temperature sensor which consists of thermistors, thermocouples, RTDs, IC, etc. Then there are pressure sensors including vacuum, elastic liquid-based manometers, etc., proximity sensors, biosensors, humidity sensors, speed sensors, tilt sensors, moisture sensors, infra-red sensors, accelerometers, magnetometers and many more.

Based on the energy or power supply requirement of the sensors, they are classified into two types as demonstrated below.
a) Active Sensors, which requires the power supply to operate. Example; LiDAR (Light detection and ranging), photoconductive cell.
b) Passive Sensor, that does not require a power supply. Example; Radiometers, film photography.

There is another term to classify sensors and that is classification based on measuring mechanism which can be classified as resistance sensing, capacitance sensing, inductance sensing, etc.

Nevertheless, there is another vital factor to help in classification of sensors which is called stimulus. According to stimulus, sensors can be catogorised as electric, acoustic, magnetic and many more. Based on stimulus, sensors are classified as shown below in simple diagram in figure 4.

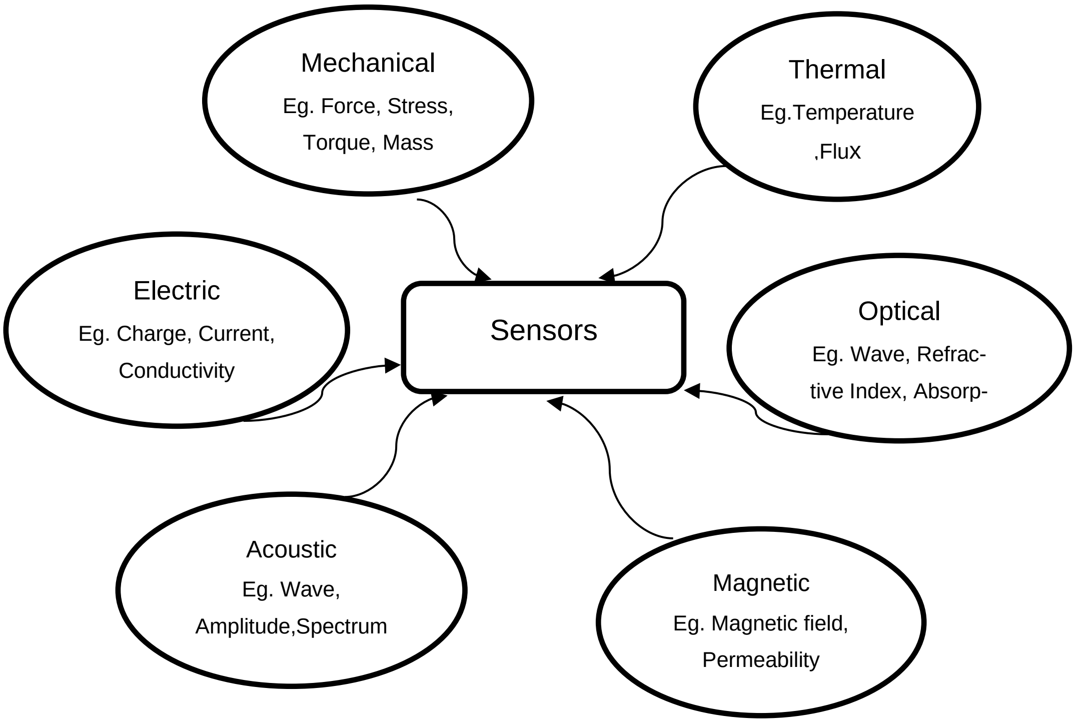

Mechanical
Eg. Force, Stress, Torque, Mass
Thermal
Eg.Temperature ,Flux
Electric
Eg. Charge, Current, Conductivity
Sensors
Optical
Eg. Wave, Refrac-tive Index, Absorp-
Acoustic
Eg. Wave, Amplitude,Spectrum
Magnetic
Eg. Magnetic field, Permeability

Figure 4. Sensors based upon stimulus

3.2 Temperature Sensors

The measurement of heat or cold of some object with reference to some standard value can be termed as temperature. The devices which can sense the temperature is known as temperature sensor. In experimental circumstances, temperature is the consequence of the heat transfer by conduction, convection or radiation and measured in Degree Celsius (°C), Degree Kelvin (°K) or Degree Fahrenheit (°F) by a device named as Thermometer. [3]

Varieties of temperature sensors are launched in the market nowadays. While choosing the temperature sensors, consideration must be taken on many aspects like temperature range, accuracy, linearity, sensitivity, hysteresis, time constant, mechanical strength, repeatability, resolution, readability (near / far), output type, price, size etc.

Temperature sensors can be differentiated based on physical principles like expansion and contraction, electrical resistance, electromagnetic waves, thermoelectric voltage etc. Some of the example sensors on the classification based on physical principal are listed in table 1.

Table 1. Classification of Temperature sensors

Physical Principle	Examples
Expansion and contraction	Bimetallic and Liquid-filled thermometers
Electrical resistance	Thermistors
Electromagnetic waves	Infrared Photography, Pyrometer
Thermoelectric Voltage	Thermocouples

The most common type of temperature sensors being used in nowadays world are discussed below.

3.2.1 Liquid or Gas Filled Thermometer

These thermometers are designed based on their expansion and contraction temperature concerning some standard values. Most of these thermometers are used in laboratories and even they are too fragile for other industrial use. They were the first invented thermometers in our history and are reliable ones too. No external energy is needed to operate these thermometers which are the advantage of using these. A bulb is connected with a capillary and a bellow element, later change in temperature will change the pressure in the supplied liquid. This change in pressure can be noticed through the bulb which will be the temperature of that surrounding. Many liquid or gases had been used for the measurement of temperature since then, but the acknowledgment of their boiling and freezing point must be taken into consideration to use them concisely as shown in table 2. To measure the temperature above 100 °C, a mercury thermometer must be used whereas to measure below -110 °C, a pentane thermometer is useful as illustrated in table 2.

‘

Table 2. Freezing and boiling point of fluid

Liquid/Gas	Freezing and boiling point (in centigrade degree)
Mercury	-39 to +357
Alcohol	-110 to +50
Toluene	-70 to +100
Pentane	-200 to +20

3.2.2 Thermocouples

Thermocouples were invented many years back and are considered as one of the earliest and most commonly used devices for measuring temperature. Thermocouples are mostly used in such applications or prototypes which are related to temperature measurements such as boilers, ovens, etc. Thermocouples are broadly deployed due to its huge range of temperature measurement which ranges from -200 °C to +2500 °C and they are not affected through shock and vibration. Another vital factor for its use is that they respond instantly to the change in temperature than other sensors.

Thermocouple consists of two dissimilar metals that are merged to form two junctions and are maintained at the two distinct temperatures respectively. Thermocouple cannot be constructed without at least two junctions with dissimilar temperatures at both junctions. Maintaining different temperatures at both junctions generate an emf within the circuit, which can be measured through some devices like a potentiometer.

The temperature of the reference junctions must be acknowledged beforehand whereas the temperature of the measuring junction will be unknown. The output emf will be obtained from the thermocouple circuit which will provide the value of unknown temperature. This generation of the output emf, when placed through two different metals with different temperatures, is simply known as the Seebek effect. Thermocouples types like J, K, and T whose voltage differs at 52 µV/°C, 41 µV/°C, and 41 µV/°C respectively at room temperature are most used thermocouples in our market. The circuit diagram behind the working principle of the thermocouple is shown in figure 5.[4]

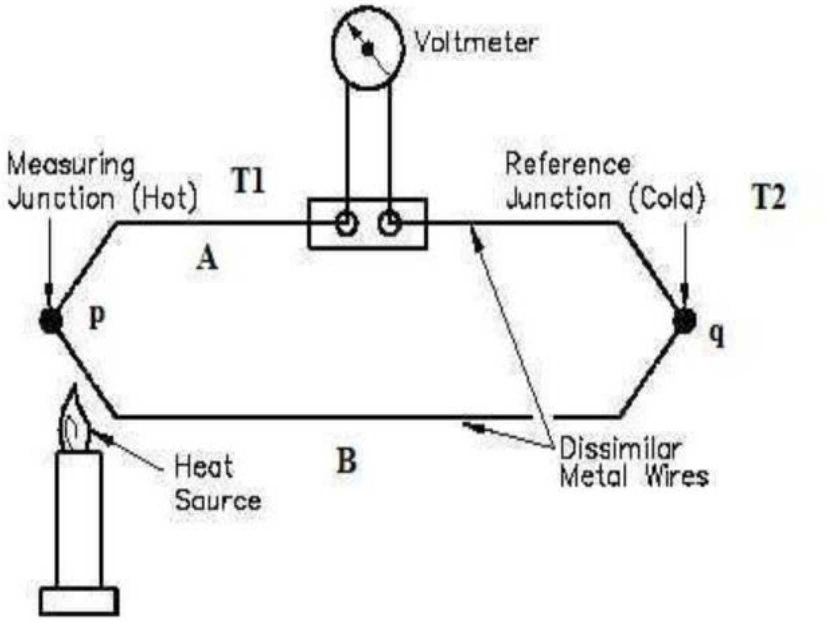


Figure 5. working principle of thermocouple [4]

3.2.3 Resistance Temperature Detector (RTDs)

RTDs or simply, Resistance Thermometer is a temperature measuring device which functions by measuring the resistance of pure electrical wire, which is referred to as a temperature sensor. Due to its high accuracy in the measurement of temperature, it's still in use. Copper, Nickel, and Platinum are the most used metals in the construction of RTDs which has excellent linear characteristics over a wide range of temperatures.

The resistance of metals depends on temperature and vice versa which is illustrated below.

Equation 1. Resistance of the metal (far range)

$$R = R_0(1 + \alpha_1 T + \alpha_2 T^2 + \cdots)$$

Equation 2. Resistance of the metal (small range)

$$R = R_0(1 + \alpha_1(T - T_0))$$

Where, R and R_0 are the resistance values at T and T_0 temperatures and α is the constant which depend on the metal [6]. This equation 1 is for huge range of temperature and the equation 2 for small range of temperature.

The resistance and temperature variations of the metals used plays a significant role in the measurement of temperature. Different metals comprised different resistance variations with respective to the temperature variations. The resistance-temperature characteristics curve of the three commonly used metals is demonstrated in figure 6.

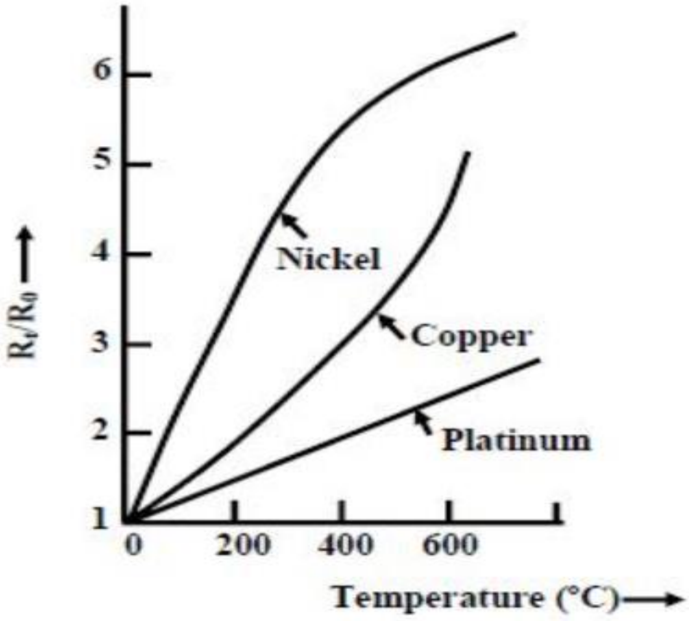


Figure 6. Resistance-temperature characteristics curve [5]

3.2.4 Thermistors

Thermistors are also temperature sensing equipment whose resistance and temperature values get altered respectively. Thermistors are very sensitive, nonlinear devices. They are comprised of semiconductor materials that display the positive (PTC) or negative temperature coefficient (NTC). The temperature is directly proportional to the resistance in PTC thermistors and vice-versa in the case of NTC thermistors. Due to its large resistance at room temperature, it cannot be affected through wires. They are frequently used in coolants, batteries, engines, and freezers for automatic shutdown processes in hot temperature especially the NTC thermistors. The resistance value is decreased when an increase in temperature, in NTC thermistors and the resistance value is increased when an increase in temperature in PTC thermistors as shown in figure 7.

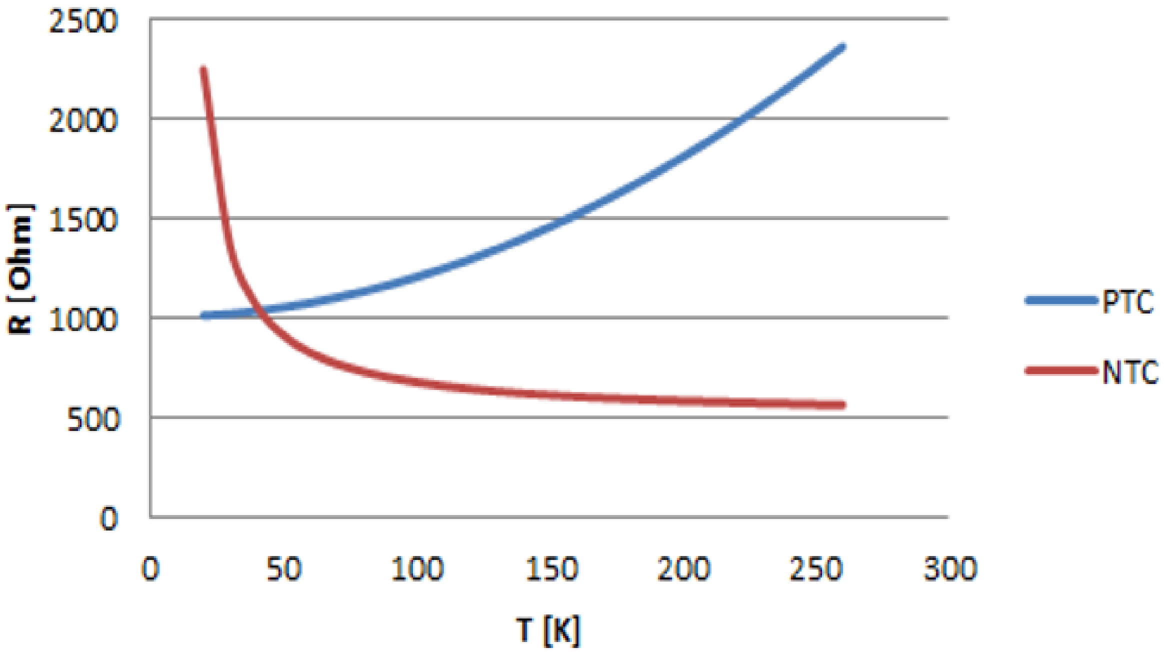


Figure 7. NTC and PTC characteristics [5]

3.2.5 Semiconductor Temperature Sensors

Integrated circuits (ICs) are known as semiconductor temperature sensors. Integrated circuits which can measure the temperature through the change of voltage across the p-n junction diode are called semiconductor temperature sensors. They are built in a way to form p-n-p and n-p-n transistor chips with the base and collector inbuild inside them. The main advantage behind the use of ICs temperature sensors is that they are very small in size so that they can be installed even on smaller devices such as MP3 players, cell phones, PCs and many more electronic devices. [6]

Due to its admirable linearity, high sensitivity and its remarkably small size, they are the best usable temperature sensors especially in electronics devices rather than others even though the temperature range is quite limited. The output will be amplified inside the integrated circuit, so measurement is pretty easier and precise than the other thermometers. Even output can be varied from analog, digital or logic depending upon our measurement circuit. Among all the sensors, they have the slowest response time (5 s to 60 s) with the narrow temperature range (-70 °C to 150 °C).

There are varieties of semiconductor temperature sensors nowadays invented and produced through many companies. Among them, LM35 and TMP36 are the ones to be discussed as these two are commonly used in most of the temperature related projects.

3.2.6 Temperature Sensor (TMP36)

TMP36 are low voltage centigrade temperature sensors and TMP36, TMP35 and TMP37 are mostly alike and all of them are used to provide the output voltage linearly proportional to the centigrade temperature. Any external calibration is not needed in the case of TMP36 and same goes for TMP3X sensors. The accuracy for these TMP36 sensors is in between ±2 degree centigrade as well as typical linearity is in between ± 0.5 degree centigrade. Most importantly the scale factor for this sensor is 10 mV/°C. In the presence of larger capacitance loads, it tends to be more and more stable. The specified temperature range for this sensor is -40°C to +125°C, which is quite low when compared to those previous temperature sensors. Low self-heating is the one of the remarkable features of this sensor. For the functional operation, less than 50 µA Quiescent current is required and the shutdown current for this sensor is 0.5 µA(maximum). The voltage range as an output is intended from 2.7 V to 5.5 V maximum. If the supply voltage is 5 V, the operation range of this device extends to 150 °C with reduced accuracy. These sensors are available in market in three packages which are TO-92, SOIC-8, and 5-lead SOT-23 and all of them are surface-mount packages. Most of them are used in industrial process control, environmental control systems, fire alarms, CPU thermal management etc. A diagram representing the pins configuration of the TMP36 temperature sensor is illustrated in figure 8. [7]

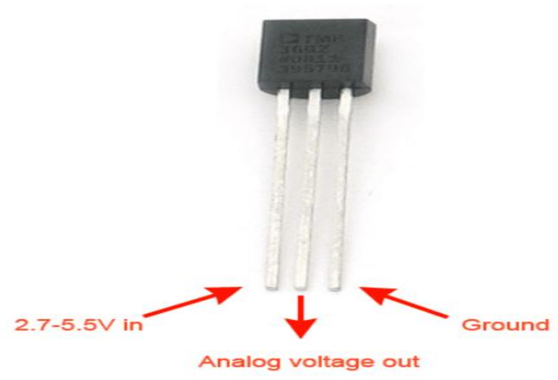


Figure 8. TMP36 [7]

3.2.7 Temperature Sensor (LM35)

LM35 is a temperature measuring integrated circuit whose output voltage is linearly proportional to the Centigrade temperature. Due to its low cost, reliability and its high accuracy, it is used in various devices for the measurement of temperature. It has very high accuracy of ± 3/4 °C. The operational temperature range of this high accurate sensor ranges from -55 to +150 degree Celsius. As well as, operational voltage of this sensor ranges from 4 V to 30 V. Due to its wafer level trimming and calibration, the cost of the sensor is quite low compared to other integrated circuit temperature sensors. The output is based on the simple basic principle which is, this sensor gives out 10 mV/°C rise in temperature. This sensor is power efficient device as it doesn't drain much power from the battery or any other power source as it only consumes just 60 µA current. Due to this low consumption of the current, self-heating of this device is low too. self-heating of the device is found to be as low as 0.1 °C. [8]
Commonly there is two packaging system for this device which are TO-46 metal package and TO-92 package. As TO-46 metal package sensor measure temperature directly when kept in contact with the surface whose temperature is to be measured, while TO-92 package is more precise as it is based on the measurement of the temperature of the terminals rather than the plastic casing as metal terminals are more conductive than the plastic casing. Usually, TO-92 is used to measure temperature rather than TO-46 package. Mostly, they are used in HVACs, battery management systems, power supplies and more suitable for remote applications. This device is more alike TMP36 as both are used for measuring temperature with almost uniform pins configuration as shown in figure 9

where the 1st pin is for the power i.e. Vcc which is 5 V for this sensor, 2nd pin is for the analog out and the 3rd one is for the ground.

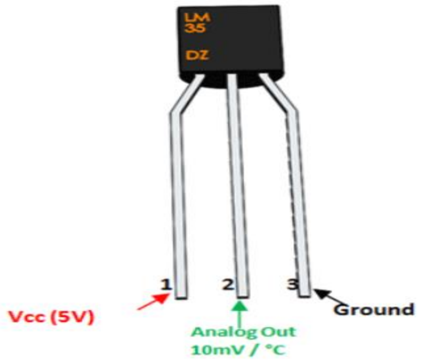
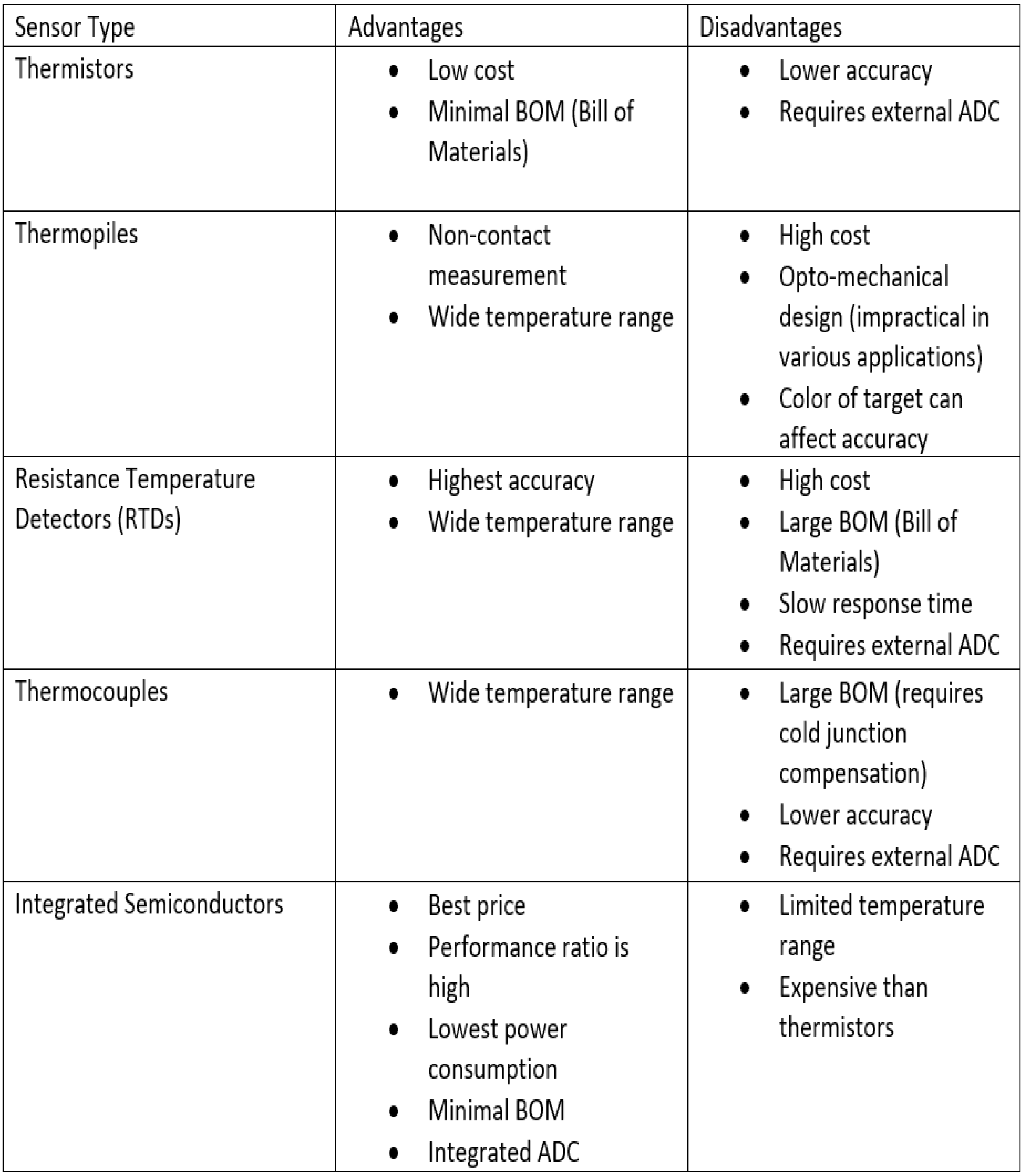


Figure 9. LM35 [8]

3.3 Pros and cons of different temperature sensors

Every temperature sensor has their own advantage and some disadvantages too depending upon their linearity, operational voltage, temperature ranges etc. A table of advantages and disadvantages of various temperature sensor is shown in table 3 depending upon various aspects like accuracy, cost, range etc.

Table 3. Pros and cons of different temperature sensors. modified [9]



Sensor Type	Advantages	Disadvantages
Thermistors	• Low cost • Minimal BOM (Bill of Materials)	• Lower accuracy • Requires external ADC
Thermopiles	• Non-contact measurement • Wide temperature range	• High cost • Opto-mechanical design (impractical in various applications) • Color of target can affect accuracy
Resistance Temperature Detectors (RTDs)	• Highest accuracy • Wide temperature range	• High cost • Large BOM (Bill of Materials) • Slow response time • Requires external ADC
Thermocouples	• Wide temperature range	• Large BOM (requires cold junction compensation) • Lower accuracy • Requires external ADC
Integrated Semiconductors	• Best price • Performance ratio is high • Lowest power consumption • Minimal BOM • Integrated ADC	• Limited temperature range • Expensive than thermistors

3.3.1 Humidity sensor (Hygrometer)

The water content in any solid, liquid and gases had been playing a vital role by affecting their physical as well as chemical behavior of these matters. Moisture is the term related to the water content in most of the solids and liquids. Humidity is simply the water content in gases (air). Humidity had been playing as a major factor in the wellness of the human or simply it played a vital role to create or eradicate our comfort zone. Like, in -30 °C in case of the dry air, the more comfortability can be felt compared to the temperature at 30 °C with humid or moist air. Apart from these, the humidity had been playing an essential role in many fields from agriculture to industries like medical, electronics, and many others.

To stabilize the comfort zone, the presence of humidity around the surrounding must be known which can only be determined by measuring it. The device used to measure the humidity is simply known as Hygrometer or Humidity Sensor. Leonardo da Vinci had invented a simple hygrometer using animal or human hair to measure humidity. It worked in a way like a length of hair changes with humidity under some tension as hair show hygroscopic characteristics under some tension. It was called a hair-tension hygrometer and was the first hygrometer in history.

There are various vital terms in the field of humidity to know it better. Even humidity is divided into two, absolute and relative humidity. The ratio of mass of the water vapor to the volume of the air is the absolute humidity. If the mass of the water vapor is considered as “m” and the total volume of air and water vapor mixture by “V”, then Absolute Humidity (AH) is given by AH = m / V. Absolute humidity is changed with the change in temperature and pressure, but the temperature is not considered while measuring absolute humidity. [10]

Relative humidity is the term often considered, whenever humidity measurement been taken place (unless specified). Relative Humidity or RH is the ratio of the actual water vapor pressure present in the air at a temperature to the maximum water vapor pressure present in the air at the same temperature [11]. Temperature and pressure both terms are taken into consideration while measuring the relative humidity. The moisture content as well as the air temperature are measured while measuring the relative humidity.

The relative humidity is measured as the ratio of the actual water vapor pressure present in the air at a temperature (actual vapor density) to the maximum water vapor pressure present in the air at the same temperature (saturation vapor density) and the formula for this relative humidity is shown in equation 3.

Equation 3. Relative Humidity [11]

$$\text{Relative Humidty (\%)} = \frac{\text{actual vapor density}}{\text{saturation vapor density}} \times 100$$

3.3.2 Types of humidity sensor

Based on the specification used for measuring humidity, there are three most common humidity sensors. They are capacitive, resistive and thermal conductive. Apart from these three major humidity sensors, there are optical hygrometer, oscillating hygrometer, gravimetric hygrometer etc. Hygrometers must be chosen on the basis of our needs which contain the perimeters like accuracy, reliability, output repeatability, cost, size and many more factors. Among various hygrometers, the three most common hygrometers are being described below.

3.3.3 Capacitive humidity sensor

Capacitance Hygrometer is based on the principle of change in capacitance with change in relative humidity. It measured the humidity which is relative humidity in a way that increase in capacitance is directly proportional to the increase in relative humidity. The change in the dielectric constant of the material used in our capacitive hygrometer is the reason behind the change in the capacitance and finally, leads towards the change in the relative humidity.

There are many advantages of this capacitive hygrometer to be used and some of them are the output voltage of this sensor is almost linear, the results provided by this sensor is more stable and they are useful in detecting wide range of relative humidity. Some of the disadvantages of this sensor is that the distance from the signaling circuit and the sensor is very limited. And as well, the circuit design will be complicated as the range of capacitance between 0 and 100% RH will be so high.

These sensors are used in a vast range of devices and applications like HVAC Systems, weather Stations, food processing, automobiles, refrigerators and many more. The manufacture of both capacitive and resistive humidity sensors is shown in figure 10 where the layers inside the capacitive and resistive humidity sensors are displayed magnificently.

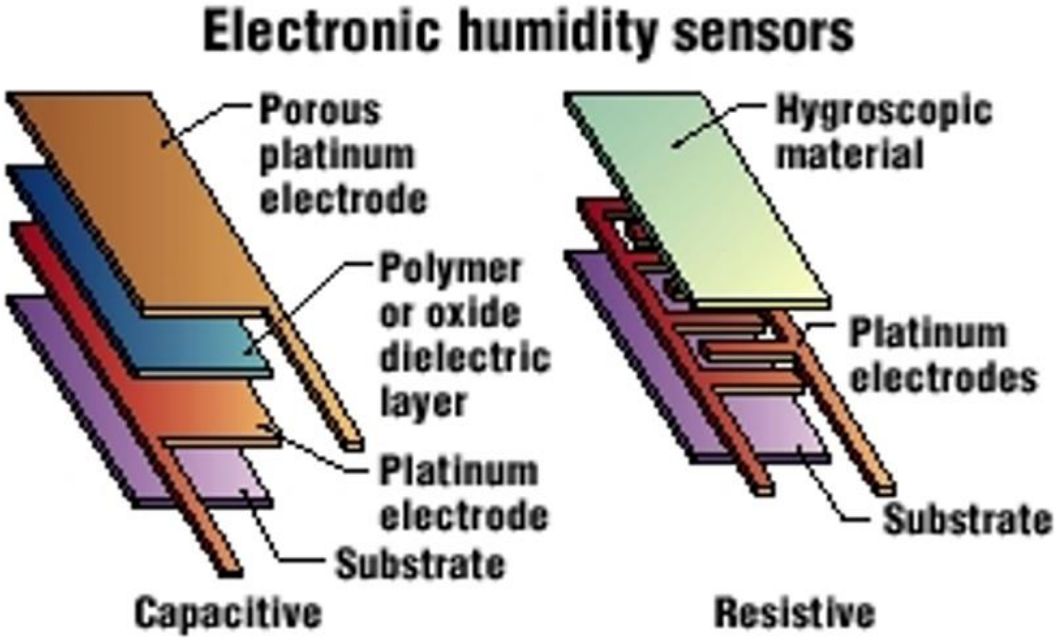


Figure 10. Construction of capacitive and resistive hygrometer [11]

3.3.4 Resistive humidity sensor

Resistive humidity sensor is based upon the impedance (resistance) or simply electrical conductivity, therefore it is even termed as Electrical Conductivity Sensors. The main principle behind this sensor is based on the principle of change in resistance (impedance) with change in relative humidity. The relationship between relative humidity and the impedance is inversely proportional which means impedance decreases as RH increases and vice versa.

The conductivity in non – metallic conductors is dependent on their humidity (moisture), this is the basic idea behind the resistive humidity sensor. The commonly used electrolyte inside the terminal or in between the electrodes is salt and but some solid polyelectrolytes and some conductive polymers are also used often. As resistive humidity sensors are very sensitive to the chemical vapors and some chemical contaminants too, they are being coated with ceramic substance to provide extra protection nowadays. The common electrodes for the construction of these sensors are made of noble metals like gold, silver or platinum. The ions in salt are utilized to measure the impedance of atoms and with the change in humidity, the resistance of the electrode's changes on either side of the salt medium, which helps to figure out the relative humidity of the desired surrounding or the applications.

The benefits of these sensors are low cost, perfect size, highly interchangeable with no calibration standards, as well as the distance between the sensor and the signaling circuit, is larger than the previous capacitive sensors. The drawbacks of these sensors are they are delicate in terms of chemical vapors and other chemical contaminants.

Apart from its own advantages and disadvantages, these resistive humidity sensors are being used in many fields and applications including from several industries, domestic and commercial applications.

3.3.5 Thermal conductivity humidity sensor

These sensors are commonly known as absolute humidity sensors as they measure the absolute humidity while the previous two sensors were measuring the relative humidity. The thermal conductivity of both dry air as well as air with water vapor are measured with these sensors and the difference between these two individual thermal conductivities provide us the exact absolute humidity as a result, this is the simple principle behind these sensors.

Two small thermistors with negative temperature coefficients are used for the construction of this device. One of the thermistors is enclosed inside the chamber filled with dry nitrogen while another one is exposed to the open environment through some venting holes. When the circuit is powered on, the resistance provided by the two thermistors is calculated and the difference between the output of these two thermistors provides us the absolute humidity. These sensors are very useful in high temperature and high corrosive environments. They are more durable, and their resolution is higher as well. Measurement reading might get affected if this sensor is exposed to any other gases than the nitrogen.

The most common applications of these sensors are they are used in drying kilns, food dehydration, pharmaceutical plants, drying machines, and clothes dryers and in many more applications. the working principle of a thermal conductive humidity sensor is displayed below as figure 11.

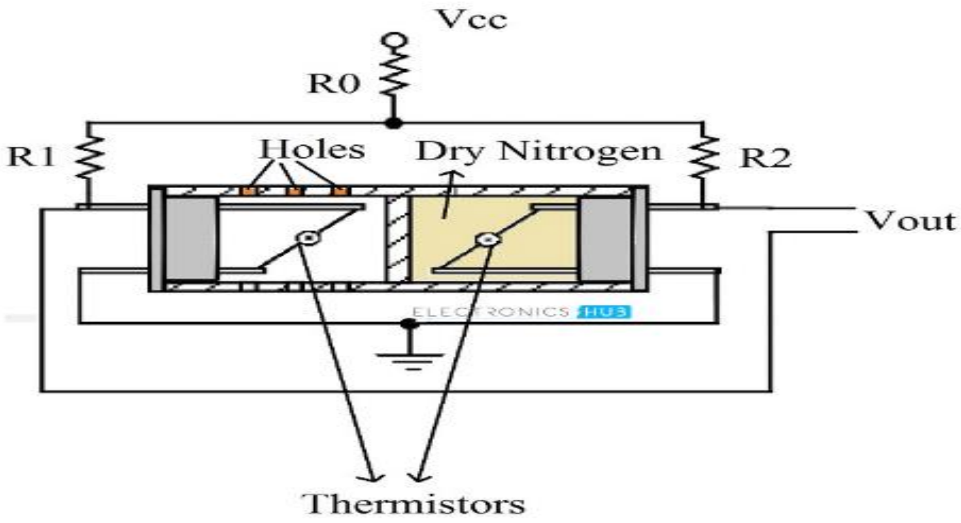


Figure 11. Construction of thermal conductive humidity sensor [12]

3.3.6 Humidity Sensor (DHT22)

DHT22 is a capacitive humidity sensor that measures both temperature and humidity. A humidity sensing component, an NTC thermistor and an IC on the back side of the sensor are the main components inside this DHT22 sensor.

Humidity is measured as explained above in capacitive humidity sensor in a way that it contains two electrodes with moisture holding substrate between them. After the change in humidity, the resistance between these electrodes changes and this change in resistance is measured which is the relative humidity which is meant to be measured. On the other hand, the temperature is measured with the NTC thermistor which is enclosed inside the DHT22. Thermistors function like a variable resistor whose resistance changes with the change in temperature and in the case of NTC thermistor, the resistance decreases with an increase of the temperature.

The characteristic features of this DHT22 sensor is displayed below [13].

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1” spacing
- Serial transmission data protocol
- Long term stability of ±0.5% per year

This DHT22 has admirable quality, superb response, strong anti-jamming capability, low power consumption, and small size too. But the price is quite expensive than expected. Due to its various advantages, it is being used in many electronics projects and in many fields of industries, medicines, and agriculture. They are commonly used in an automatic climate control sensor, weather stations, and other environmental monitoring devices.

The inner composition of the DHT22 sensor and its pins configuration is shown as figure 12 where the first pin is for the supply as Vcc, the second pin is used for the output, the third pin is not connected and the fourth one is used for ground. It is displayed through the figure that the thermistor is used inside the DHT22 for temperature sensing purposes.

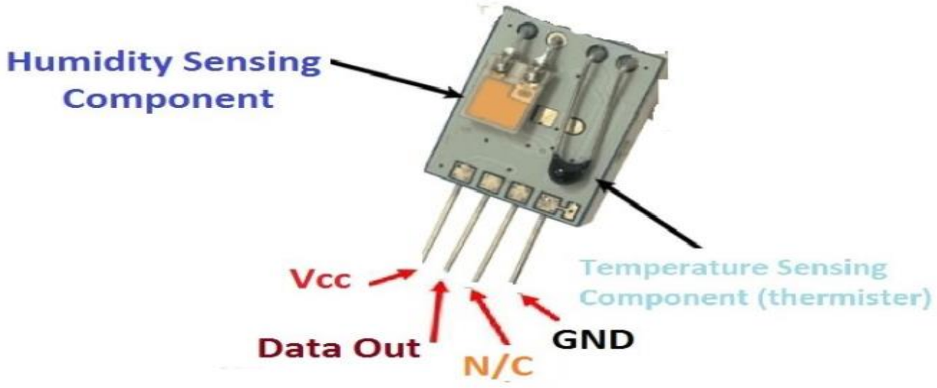

Humidity Sensing Component
Temperature Sensing Component (thermister)
Vcc
Data Out
N/C
GND

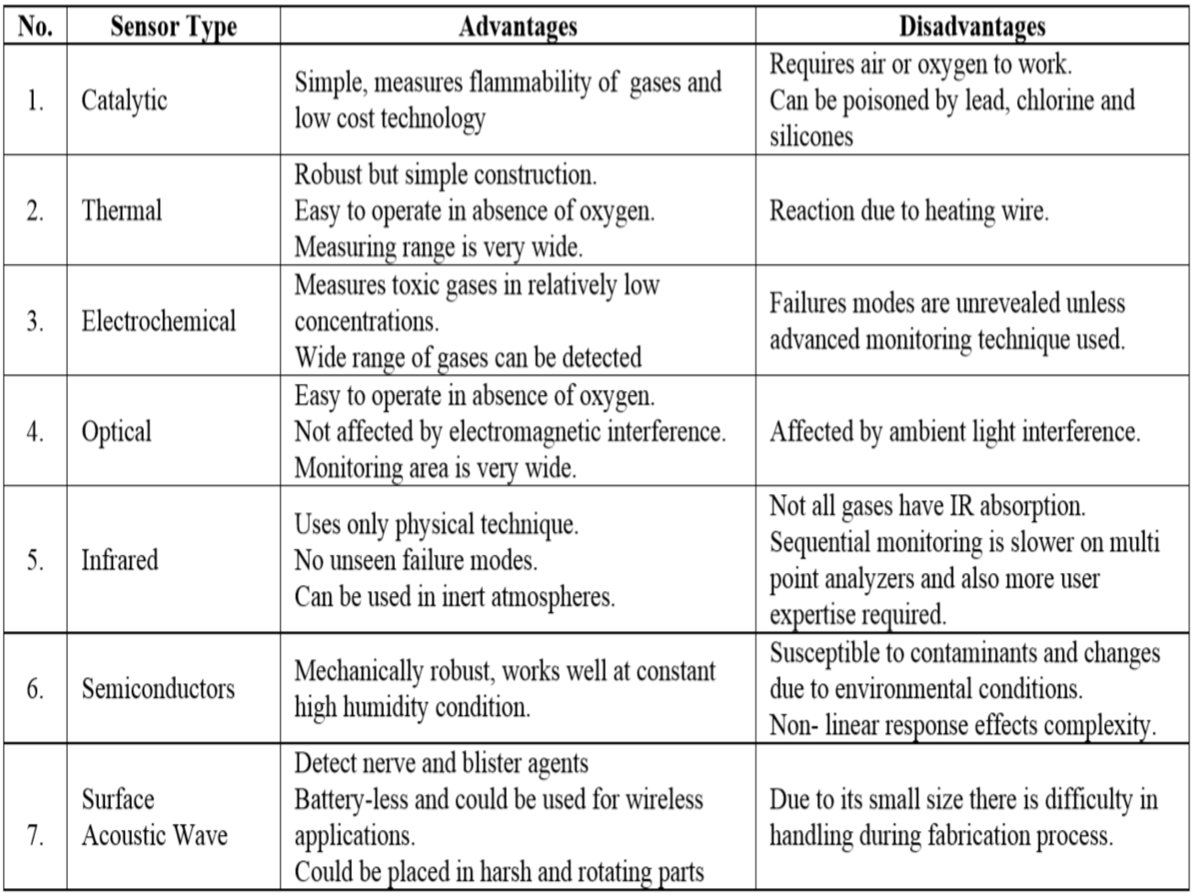
Figure 12. DHT22 [13]

3.4 Air Quality Sensor

Those sensors which can detect the poisonous gases present in the air are referred to as air quality sensor. Air quality sensors can also be named as air pollution sensors as their main objective is to detect poisonous gases like carbon dioxide, propane, benzene, smoke, alcohol and many more depending upon the types and models available in the market. These sensors worked by detecting various chemical contents which contribute to the appropriate voltage variation at the output depending upon the surrounding where these sensors are being tested.

There are various types of air pollution sensors depending upon the pollution or hazard and its chemical composition. Depending upon the various hazards present in the surrounding, there are numerous types of air pollution sensors and they are listed below as table 4 with their advantages as well as disadvantages.

Table 4. Comparison of various air pollution sensors [14]



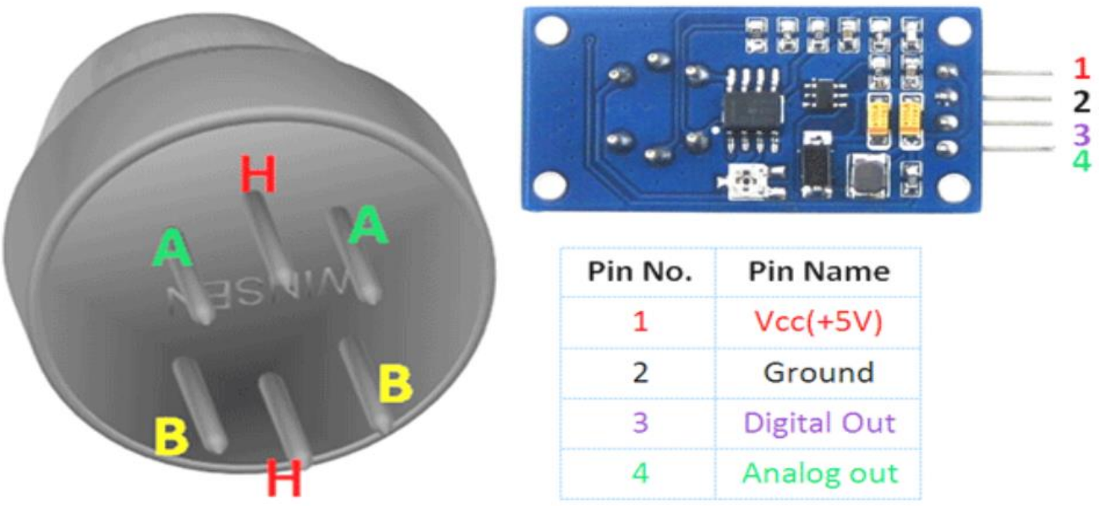
No.	Sensor Type	Advantages	Disadvantages
1.	Catalytic	Simple, measures flammability of gases and low cost technology	Requires air or oxygen to work. Can be poisoned by lead, chlorine and silicones
2.	Thermal	Robust but simple construction. Easy to operate in absence of oxygen. Measuring range is very wide.	Reaction due to heating wire.
3.	Electrochemical	Measures toxic gases in relatively low concentrations. Wide range of gases can be detected	Failures modes are unrevealed unless advanced monitoring technique used.
4.	Optical	Easy to operate in absence of oxygen. Not affected by electromagnetic interference. Monitoring area is very wide.	Affected by ambient light interference.
5.	Infrared	Uses only physical technique. No unseen failure modes. Can be used in inert atmospheres.	Not all gases have IR absorption. Sequential monitoring is slower on multi point analyzers and also more user expertise required.
6.	Semiconductors	Mechanically robust, works well at constant high humidity condition.	Susceptible to contaminants and changes due to environmental conditions. Non- linear response effects complexity.
7.	Surface Acoustic Wave	Detect nerve and blister agents Battery-less and could be used for wireless applications. Could be placed in harsh and rotating parts	Due to its small size there is difficulty in handling during fabrication process.

3.4.1 Air Quality Sensor (MQ-135)

MQ-135 is the most commonly used air quality sensor with wide range of detecting pollution or hazards like ammonia (NH3), carbon dioxide (CO2), nitrogen oxides (NOx), alcohol, benzene, smoke, etc... It can be used as either analog or as a digital sensor and as well the sensitivity of this sensor can be varied using the potentiometer. One of the main features of this sensor is that it has high sensitivity to a long life. The operating voltage for this sensor is 5V.

The pin configuration of the MQ-135 sensor including the module pins as well as sensor pins are displayed in figure 13. For the sensor, among two H pins, one is connected to the supply and another to the ground whereas A and B pins are interchangeable. If A pins are connected to the supply voltage, one of the B pins will act as output pin leaving the other pin to be the ground and vice-versa.

For the module, the sensor is powered through Vcc which is 5V whereas the ground pin is connected to the system ground. Digital out pin is used when digital data is required as output which can be obtained by setting the threshold value through the potentiometer. Analog out pin is used to display the analog voltage from 0 to 5 volt depending upon the intensity of the gases.



Pin No.	Pin Name
1	Vcc(+5V)
2	Ground
3	Digital Out
4	Analog out

Figure 13. Pin configuration of MQ-135 [15]

4 Internet of Things (IoT)

It is believed that "Kevin Ashton" co-founder of the Auto-ID Center at MIT, first mentioned the term "the internet of things” while presenting in 1999 for Procter and Gamble. He had named his presentation "Internet of Things" to draw the attention of Procter and Gamble senior management team by using the cool new trend of 1999 – the internet. But the presentation was mainly focused to bring the attention on RFIDs. In the history of internet, the first internet appliance, was a Coke machine at Carnegie Mellon University in the early 1980s. [16]

The Internet of Things (IoT) can be termed as the Internet of everything as almost everything is being internet-connected in this technological world. IoT, in recent years, has become more and more popular due to its IoT – related software, hardware, sensors, accessories, and internet connections becoming more economical and convenient.

The simplest definition of IoT - "Internet of things is way to connect computers sensors actuators through internet protocols" [16]. Simply, IoT wraps up everything that is connected to the internet.

IoT has touched every industry, including defense, ecosystem, healthcare, finance, retail, and manufacturing. Smart cities monitored by IoT had been helping in the reduction of the waste as well as conservation of the energy too. IoT has been assisting in agriculture by monitoring crops and cattle yields and predicting growth patterns too. IoT has become an essential need for everyone from business to medicines as well as farming and automation and many other sectors. It is helping people to live and work intelligently as well as to gain total control over their desires. IoT has been comprising devices from a simple gadget to the smartphones and wearables which are linked to the internet.

Defining IoT is not easy as IoT has covered all the areas and scopes which is connected through internet. In a figure 14, several approaches and views are illustrated to define IoT from connecting devices to human values. Figure 14 is illustrating how an IoT works in a clear way starting from device connection ending at smart applications for human benefits.

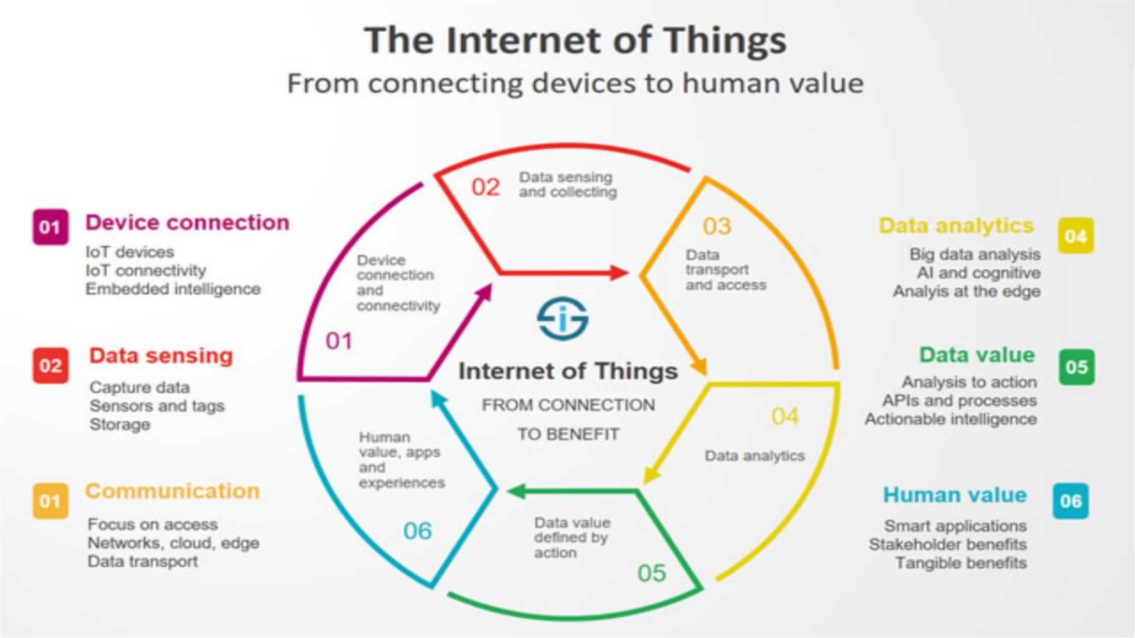


Figure 14. Several definitions of IoT [17]

4.1 Essential Terminology of IoT

It is crucial to acquire the knowledge of the IoT terminology, hardware and software in order to confirm which IoT package will be best for our desired application. The most terms being used in IoT field are being discussed below.

a) IoT Development Board

It is simply a programmable board which contains an IoT device within itself. The IoT device's processor, memory, communication ports, Wi-Fi module, sensors, input output pins etc. are all exposed by the IoT development board, in a simple way, to the user.

b) Microcontrollers

Simply, microcontrollers are small microprocessors which has a CPU, RAM, ROM and input and output ports all embedded on a single microchip. Microcontroller once programmed can function on its own as it has a saved set of instructions that it runs when needed.

c) Microprocessors

Normally, it is known as CPU (Central Processing Unit) which means it is the brain of any computing machine. It is used for general purpose programming like logical operations and contains its own RAM/ROM modules as well as digital inputs/outputs.

d) Flash Memory

These are the electronic non-volatile storage gadget like pen drives, memory cards etc. Data can be easily erased and rewritten. These are used as reusable extra storage.

e) RTOS (Real Time Operating System)

RTOS reacts to those events happening in real time, which means as soon as an event happens, a response is assured within a calculable amount of time. It plays a vital role in an embedded system, where real time response is necessary.

f) Cloud Technology

It denotes to the computing resources accessible for use over the internet. data storage, data analytics, data streaming etc. are the essential relevant resources to IoT.

g) M2M Communication (Machine to Machine)

It bounds the communication between two or more machines like sensors, actuators, computers etc. over both wired and wireless network.

h) MBaaS (Mobile Backend as a Service)

It is mainly focused for mobile application developers where it provides an infrastructure for cloud storage, data streaming, push notifications and other services. These services are exposed through the web-based application programming interface (API).

i) GPIO (General Purpose Input Output)

GPIOs are the electrical terminals simply known as "pins" which are exposed from the IC's and IoT devices to either send or receive a signal to or from the device.

j) Module

It can be considered as a unit of electronics. Usually, a single integrated circuit and at other times a group of components even including IC's, providing a logical function to the device can be termed as a "module".

k) Port

Port is an interface either electrical or radio frequency based available on a device through which external devices can communicate with or from the device. Example, Ethernet, HDMI, USB, UART, 3.5mm jack etc.

4.2 IoT Development Boards

There are lots of IoT development boards invented till yet. Among various IoT development boards, the most popular boards are being illustrated below in table 5, with their specifications in terms of processor, speed, memory, available communication modules and ports and IO pins.

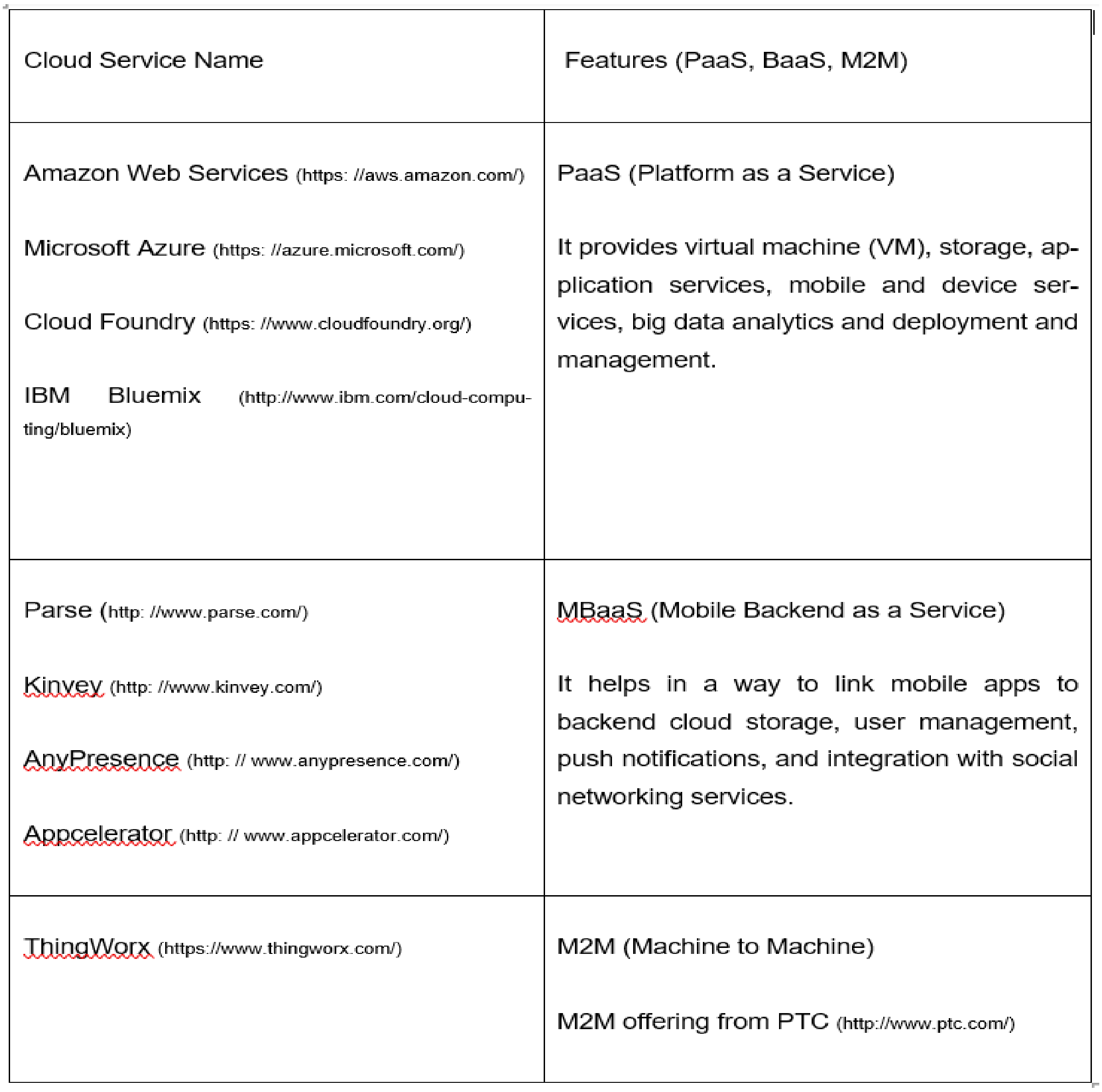
Table 5. IoT Development Boards [17]

Board Name	Microcontroller	Microprocessor	Memory	Modules	Ports	IO Pins
Raspberry Pi 1/2/3	Broadcom SoC BCM2835/6/7	Single/Quad-core ARM11/ Cortex-A7/ A53 CPU, VideoCore IV GPU	256 MB/ 512MB/ 1GB RAM	Ethernet, Wi-Fi, Serial UART, I2C	HDMI, USB, Ethernet (RJ45), GPIO	26/ 40/ 40
Arduino Mini	ATmega 328	NA	32KB Flash 2KB SRAM	NA	NA	14
Arduino Uno	ATmega32u4	Atheros AR9331	32KB Flash 2.5KB SRAM, 16MB Flash,64MB RAM	Wi- Fi, Ethernet	USB, Ethernet (RJ45)	20
Intel Edison	MCU at 100 MHz (Intel Atom Soc)	Dual-core CPU at 500 MHz (Intel Atom Soc)	4GB Flash, 1GB RAM	Wi-Fi, Bluetooth 4.0	USB, UART, SPI, GPIO	28
Libelium Waspmote	ATmega 1281	NA	128KB Flash, 8KB SRAM	Temp, Humidity, Light sensors, (optional) GPS	UART, I2C, SPI, USB	19
NodeMCU ESP8266	ESP 8266 SoC	ESP-12 Module	4 MB Flash	Wi-Fi, Serial, UART, ADC	UART, GPIO, SPI	14
BeagleBone Black	Sitara SoC AM3358/9	AM335x1 GHz ARM Cortex-A8	512MB RAM, 2/4 GB Flash storage	Ethernet, Serial, UART, ADC, I2C	Ethernet (RJ45), HDMI, USB, GPIO	24
CubieBoard	ARM Cortex -A8 CPU	A11Winner A10 SoC	512 MB/1GB RAM, 4 GB Flash memory	Ethernet, Serial, UART, ADC, I2C	Ethernet (RJ45), USB, SATA	96

4.3 Cloud Services

It is essential to know the information about the cloud services we are dealing with, when initiating the task related to IoT. Information like whether the board has open standards or not or whether the board requires some manipulation to use in the ongoing project, should be well known before hand, initiating the project. A table 6 is constructed to demonstrate the salient features of the cloud service provider like whether the cloud service is PaaS, BaaS or M2M.

Table 6. Cloud services with their features [17]



Cloud Service Name	Features (PaaS, BaaS, M2M)
Amazon Web Services (https: //aws.amazon.com/) Microsoft Azure (https: //azure.microsoft.com/) Cloud Foundry (https: //www.cloudfoundry.org/) IBM Bluemix (http://www.ibm.com/cloud-computing/bluemix)	PaaS (Platform as a Service) It provides virtual machine (VM), storage, application services, mobile and device services, big data analytics and deployment and management.
Parse (http: //www.parse.com/) Kinvey (http: //www.kinvey.com/) AnyPresence (http: // www.anypresence.com/) Appcelerator (http: // www.appcelerator.com/)	MBaaS (Mobile Backend as a Service) It helps in a way to link mobile apps to backend cloud storage, user management, push notifications, and integration with social networking services.
ThingWorx (https://www.thingworx.com/)	M2M (Machine to Machine) M2M offering from PTC (http://www.ptc.com/)

5 Particle IoT Devices

Particle was started as Hex Goods by Zach Supalla in the year 2011 mainly focused on selling the designer products online. Later Hex Goods was shut down and Zach Supalla initiated another company named Switch Devices, which dealt with lightening devices. Later, they renamed the name Switch Devices to Spark Devices. In 2013, spark transformed to an IoT platform for developers and engineers. After entering IoT field, Spark created lots of confusion in their customers regarding the name so "Particle" was the name chosen to replace the name Spark. In this way, Particle named company was founded which had been providing development kits and devices for the upliftment of IoT sector.

After entering the field of IoT, Particle had been providing some Wi-Fi Modules or GSM modules IoT devices which been helping the developers and engineers in easy connection to the internet, with one exciting feature called Over-The-Air (OTA) code deployment. The Particle module capability of deploying code over the air had removed the inconvenience of adding extra components on the prototyping boards for connectivity and had helped with a rapid code release cycle and testing / debugging.

Particle had been offering some hardware and software tools to help manage, create and prototype the IoT products. Particle had invented some IoT development boards, among them the popular hardware particle devices are Particle Core, Photon and Electron. Since these three are hardware IoT boards, the software for these IoT boards can be written in a web-based IDE contributed by Particle and the deployment of the code is done over the air. To extent the IoT experience, particle had been offering SDK's (Software Development Kit) for mobile and the web too.

5.1 Spark Core

Spark Core was the first IoT prototyping board developed by Particle which is an Arduino compatible, Wi-Fi enabled, cloud powered development platform. These Cores are accumulated with Wi-Fi modules within themselves which assist them connect to the internet or some network without the addition of some extra modules. The technical specifications of the Spark Core are as follows:

- ARM Cortex M3 CPU
- 72 MHz operating frequency
- 128 kB flash memory
- 20 kB RAM
- 12-bit ADC
- Wireless programming
- Analogue and Digital I/O pins
- TI CC3000 Wi-Fi module
- 802.11 b/g Wi-Fi support

5.2 Particle Photon

Particle developed the prototyping board called Photon by considering the feedback received from Spark Core users. It is more advanced version of Core in the terms of CPU and memory too. Like Spark Core, Photon are packaged with Wi-Fi modules to assist them connect to the internet without any extra gadget. Photon are also Arduino compatible and the software or code can be written in the web-based IDE provided by particle. Extra feature like a wake-up pin for waking up from low power modes has been added in this device. They are optimized to use less power than the Spark Core which is made possible with the deployment of new Wi-Fi module named Broadcom's BCM43362. The technical features of the Particle Photon are listed below:

- 120 MHz ARM Cortex M3 processor
- Wireless programming
- Broadcom BCM43362 Wi-Fi chip
- Supports 802.11 b/g/n Wi-Fi
- 1 MB flash memory
- 128 kB RAM
- 18 GPIO and peripheral pins

- On-board RGB LED
- Real Time Operating System (Free RTOS)
- Support for Access Point (AP) mode
- Open source design
- FCC, CE, IC certified

5.3 Particle Electron

Electron has 2G/3G module rather than Wi-Fi module, which can be used to transfer messages or connect to the specified internet. Electron is created to help in innovation and development of the IoT projects for the cellular electronics projects and products rather than Wi-Fi. Electron is optimized for low bandwidth messages and it is shipped with SIM card which offers affordable data plans for over 100 countries worldwide. Price is based upon the electron variant depending upon the generation of the cellular service provided by electron which is either its 2G or 3G. The technical specifications of the Particle Electron are listed below:

- 1 MB flash
- 128 kB RAM
- Wireless programming
- U-blox SARA-U260/U270(3G) and G350 (2G) cellular module
- STM32F205 120MHz ARM Cortex M3 microcontroller
- RGB status LED
- 30 mixed-signal GPIO and advanced peripherals
- RTOS
- FCC, CE, and IC verified

Electron board can be powered through the Lithium-Ion polymer battery rated at 3.7 VDC 1800 mAh, through the Vin (3.9 V – 12 VDC) or the USB micro B connector. The mentioned LiPo battery can be substituted with another LiPo battery rated 3.7 VDC but with a higher current rating. With little as 550 mA at 5 V, this module can be powered. At 5 VDC, the typical current consumption can range from 180 mA to 1.8 A. In the case of deep sleep if powered from the battery alone the quiescent current is 130 µA.

There are two types of FCC-approved antennas manufactured by Taoglas where 2G/3G PCB antenna (PC104.07.0165C) has the gain of 1 dBi to 2.39 dBi and the other one called LTE flex antenna (FXUB63.07.0150C) has gain of 5.00 dBi at peak. [19]

About the peripherals and GPIO, there are 30 digital pins which can perform as both input and output, analog (ADC) 12 pins as input only, analog (DAC) 2 pins as output only, PWM 13 pins as output, UART 3 pins, SPI 2 pins, I2S 1 pin, I2C 1 pin, USB 1 pin and CAN 2 pins where all of them can be operated like both input and output as well. There is a receiver pin as UART RX and transmitter pin as UART TX where both can be used as a digital GPIO and PWM too and the configuration of the pins can be observed in figure 15. As illustrated pins and many more peripherals and GPIO pins can be seen in figure 15.

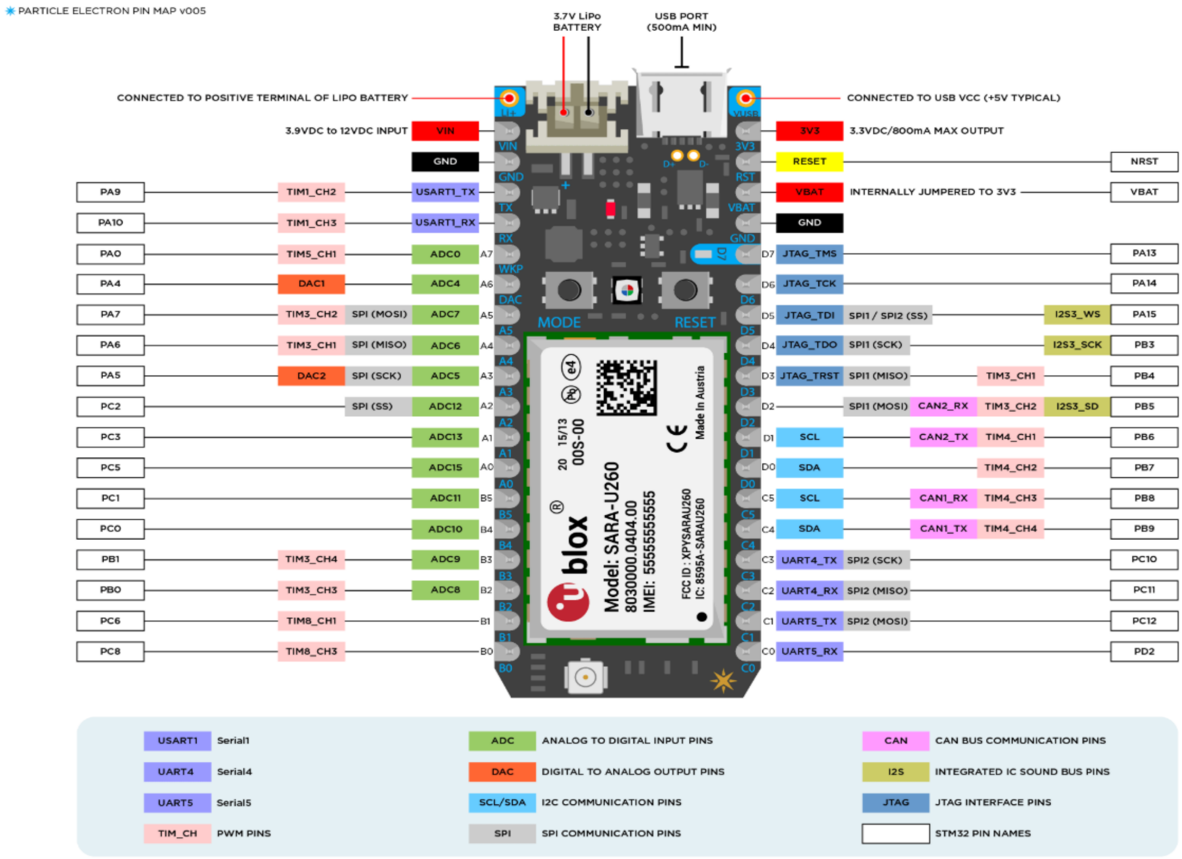
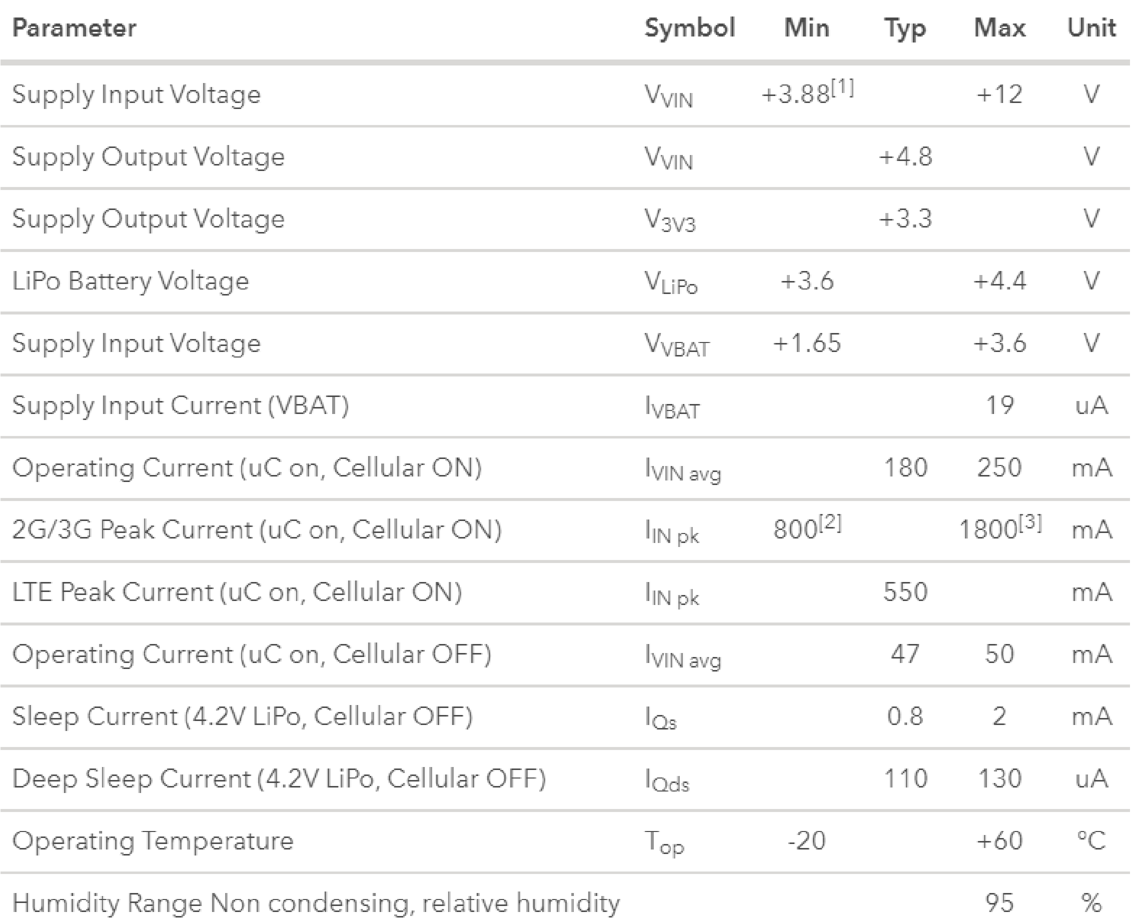

PARTICLE ELECTRON PIN MAP v005
3.7V LiPo BATTERY
USB PORT (500mA MIN)
CONNECTED TO POSITIVE TERMINAL OF LIPO BATTERY
CONNECTED TO USB VCC (+5V TYPICAL)
3.9VDC to 12VDC INPUT VIN
GND
3V3 3.3VDC/800mA MAX OUTPUT
RESET NRST
VBAT INTERNALLY JUMPERED TO 3V3 VBAT
GND
PA9 TIM1_CH2 USART1_TX
PA10 TIM1_CH3 USART1_RX
PA0 TIM5_CH1 ADC0 A7
PA4 DAC1 ADC4 A6
PA7 TIM3_CH2 SPI (MOSI) ADC7 A5
PA6 TIM3_CH1 SPI (MISO) ADC6 A4
PA5 DAC2 SPI (SCK) ADC5 A3
PC2 SPI (SS) ADC12 A2
PC3 ADC13 A1
PC5 ADC15 A0
PC1 ADC11 B5
PC0 ADC10 B4
PB1 TIM3_CH4 ADC9 B3
PB0 TIM3_CH3 ADC8 B2
PC6 TIM8_CH1 B1
PC8 TIM8_CH3 B0
D7 JTAG_TMS PA13
D6 JTAG_TCK PA14
D5 JTAG_TDI SPI1 / SPI2 (SS) I2S3_WS PA15
D4 JTAG_TDO SPI1 (SCK) I2S3_SCK PB3
D3 JTAG_TRST SPI1 (MISO) TIM3_CH1 PB4
D2 SPI1 (MOSI) CAN2_RX TIM3_CH2 I2S3_SD PB5
D1 SCL CAN2_TX TIM4_CH1 PB6
D0 SDA TIM4_CH2 PB7
C5 SCL CAN1_RX TIM4_CH3 PB8
C4 SDA CAN1_TX TIM4_CH4 PB9
C3 UART4_TX SPI2 (SCK) PC10
C2 UART4_RX SPI2 (MISO) PC11
C1 UART5_TX SPI2 (MOSI) PC12
C0 UART5_RX PD2
Li+
VUSB
MODE
RESET
blox
Model: SARA-U260
USART1 Serial1
UART4 Serial4
UART5 Serial5
TIM_CH PWM PINS
ADC ANALOG TO DIGITAL INPUT PINS
DAC DIGITAL TO ANALOG OUTPUT PINS
SCL/SDA I2C COMMUNICATION PINS
SPI SPI COMMUNICATION PINS
CAN CAN BUS COMMUNICATION PINS
I2S INTEGRATED IC SOUND BUS PINS
JTAG JTAG INTERFACE PINS
STM32 PIN NAMES

Figure 15. Pin configuration of Electron [19]

For the smooth performance of the IoT device, operating conditions of that device should be taken into consideration in order to avoid electrical and operational hazards. Different microcontrollers have different operating conditions in the context of voltage supply input as well as output, operating temperature, peak current, operating current and many more. Table 7 is generated below in order to illustrate the operating conditions as recommended by the particle.

Table 7. Particle recommended operating conditions [19]

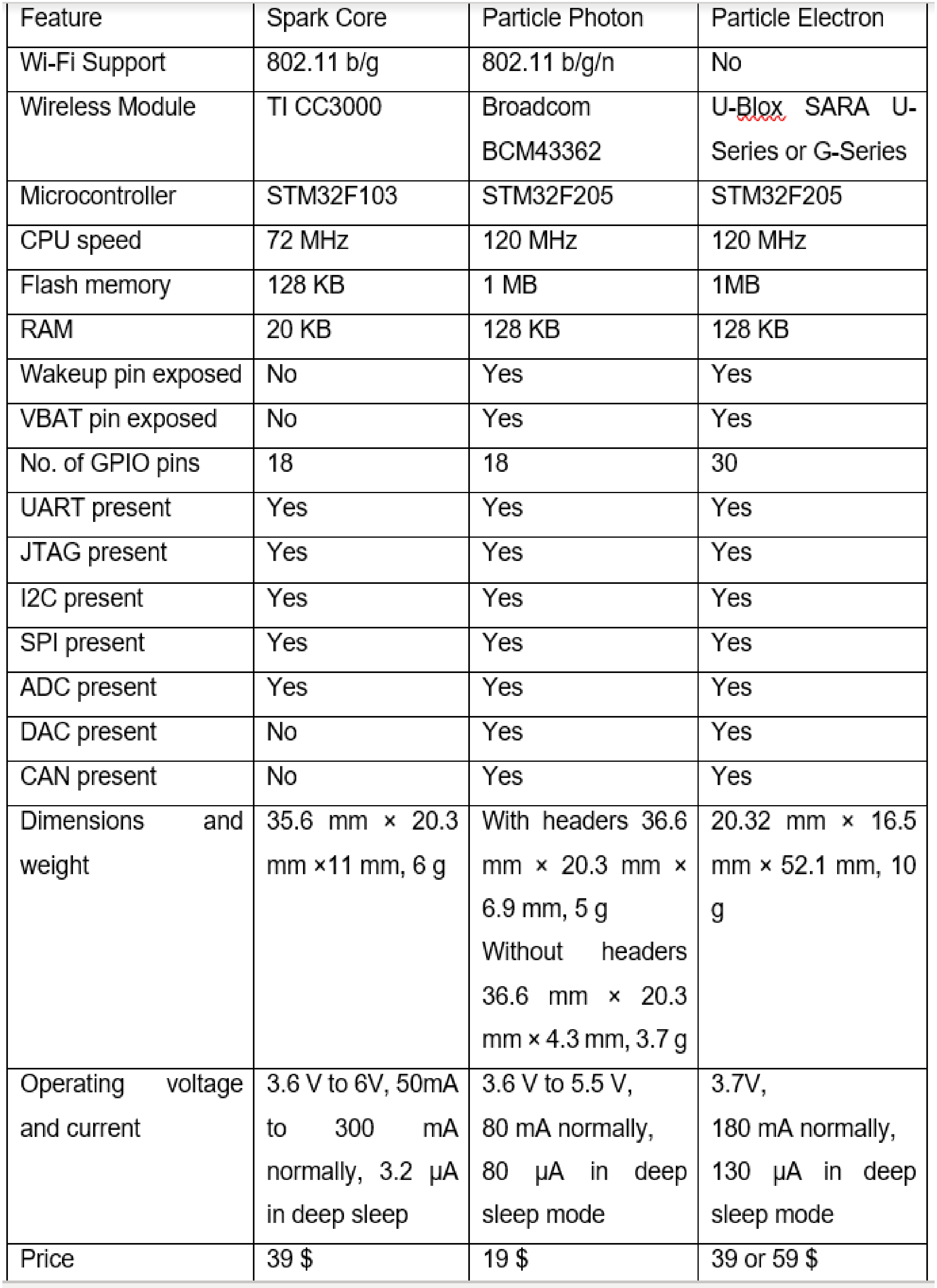


Parameter	Symbol	Min	Typ	Max	Unit
Supply Input Voltage	V_{VIN}	+3.88[1]		+12	V
Supply Output Voltage	V_{VIN}		+4.8		V
Supply Output Voltage	V_{3V3}		+3.3		V
LiPo Battery Voltage	V_{LiPo}	+3.6		+4.4	V
Supply Input Voltage	V_{VBAT}	+1.65		+3.6	V
Supply Input Current (VBAT)	I_{VBAT}			19	uA
Operating Current (uC on, Cellular ON)	$I_{VIN\ avg}$		180	250	mA
2G/3G Peak Current (uC on, Cellular ON)	$I_{IN\ pk}$	800[2]		1800[3]	mA
LTE Peak Current (uC on, Cellular ON)	$I_{IN\ pk}$		550		mA
Operating Current (uC on, Cellular OFF)	$I_{VIN\ avg}$		47	50	mA
Sleep Current (4.2V LiPo, Cellular OFF)	I_{Qs}		0.8	2	mA
Deep Sleep Current (4.2V LiPo, Cellular OFF)	I_{Qds}		110	130	uA
Operating Temperature	T_{op}	-20		+60	°C
Humidity Range Non condensing, relative humidity				95	%

5.4 Comparison between Spark Core, Particle Photon and Particle Electron

Each particle products are developed with some different specifications. Some of the technical specifications are illustrated below in table 8 with tabular comparison between Spark Core, Particle Photon and Particle Electron.

Table 8. Comparison of the particle products [17]



Feature	Spark Core	Particle Photon	Particle Electron
Wi-Fi Support	802.11 b/g	802.11 b/g/n	No
Wireless Module	TI CC3000	Broadcom BCM43362	U-Blox SARA U-Series or G-Series
Microcontroller	STM32F103	STM32F205	STM32F205
CPU speed	72 MHz	120 MHz	120 MHz
Flash memory	128 KB	1 MB	1MB
RAM	20 KB	128 KB	128 KB
Wakeup pin exposed	No	Yes	Yes
VBAT pin exposed	No	Yes	Yes
No. of GPIO pins	18	18	30
UART present	Yes	Yes	Yes
JTAG present	Yes	Yes	Yes
I2C present	Yes	Yes	Yes
SPI present	Yes	Yes	Yes
ADC present	Yes	Yes	Yes
DAC present	No	Yes	Yes
CAN present	No	Yes	Yes
Dimensions and weight	35.6 mm × 20.3 mm ×11 mm, 6 g	With headers 36.6 mm × 20.3 mm × 6.9 mm, 5 g Without headers 36.6 mm × 20.3 mm × 4.3 mm, 3.7 g	20.32 mm × 16.5 mm × 52.1 mm, 10 g
Operating voltage and current	3.6 V to 6V, 50mA to 300 mA normally, 3.2 µA in deep sleep	3.6 V to 5.5 V, 80 mA normally, 80 µA in deep sleep mode	3.7V, 180 mA normally, 130 µA in deep sleep mode
Price	39 $	19 $	39 or 59 $

5.5 Standard modes and the light patterns

5.5.1 Standard Modes

The modes of Photon are illustrated through the light patterns emitted through it. Various colors including magenta, yellow, green, etc. are emitted through photon displaying its various modes.

- When Photon is powered, it is illustrated by the white color with blinking green followed by blinking cyan and cyan at a steady state

- The breathing cyan is the mode or color which illustrates that it is connected to the internet and ready to flash codes and call functions.

- Blinking magenta is the color or mode which illustrate that over the air firmware is being updated. This is the same color to represent the photon being associated to the cloud for the first time.

- Blinking green indicates that is trying to connect to the Wi-Fi.

- Blinking blue is called listening mode which means it is waiting for some input to get connected to the Wi-Fi. To get connected to the mobile app, a photon must be in listening mode and listening mode can be obtained by holding the setup button for three seconds until the led starts blinking blue.

- Blinking white is the mode displaying that Wi-Fi is off.

- There is a mode named safe mode in photon which is displayed by breathing magenta. It is the mode when the photon is connected to the cloud, but no application firmware is running. The device operating system will be running but no programs or codes are being executed. Both the setup and reset buttons should be pressed at the same time and the reset button be released first while holding the setup button until it turns to be blinking magenta, to enter the safe mode.

- Device Firmware Upgrade (DFU) mode is the mode when custom firmware via USB should get programmed. This mode gets connected to the DFU- UTIL by accepting the firmware binary files followed by the triggering of a bootloader. To enter DFU mode, first both pins reset, and setup must be held. Then release the reset pin while holding the setup pin for a while will lead to DFU mode emitting the color flashing yellow. (first, it will flash magenta).

- Red color is the representation of the error and there are more than ten or even more faults related to orange or red. Generally, the firmware crashing is indicated by a pattern of ten red blinks. Three short blinks are called SOS pattern which will be followed by several blinks depending upon the error.

5.5.2 Fault or error modes

Blinking red indicates lots of errors and it depends upon the number of blinks so some of the errors are listed below;

- Decryption error is denoted by one orange blink.

- Two orange blinks indicate that the device is unable to connect the internet.

- Three orange blinks express that the device is connected to the internet but unable to reach the particle cloud.

- Authentication error is symbolized by the one magenta blink.

- Generic handshake error is followed by the one red blink.

- Hard fault. (One blink between two SOS pattern)

- Out of heap memory. (Eight blinks between two SOS pattern)

- Stack overflow. (Thirteen blinks between two SOS pattern)

- Heap error. (Fourteen blinks between two SOS pattern)
- Bus fault.
- Invalid length.
- SPI over-run.
- Assertion failure.
- Invalid case.
- Pure Virtual call.

6 Software

6.1 Particle Web IDE (Integrated Development Environment)

One of the important feature particle had brought in the field of IoT is that it provides a web based Integrated Development Environment (IDE) which is simply known as web IDE. For the code development on our personal computer, extra software installation is not needed anymore. The code which we will be writing will be deployed over the internet using this web particle IDE onto the photon or electron or core.

The web IDE for each and every particle device from spark core to particle photon to particle electron is at https://build.particle.io.

This above-mentioned web IDE contained a text area where our code can be written and deployed on the photon/electron to the internet. Web IDE had provided an opportunity to explore the libraries submitted by other users as well as own library can be developed as per the requirement of the project or code used. Apart from providing a web page for code execution, several buttons are situated at the edge of the web IDE to help in good execution of the code.

Buttons like flash, code, save, libraries, devices, docs, console, help, and settings are placed inside the web IDE for the better execution of code. Above all buttons, the flash button which is situated at the top left corner of the web IDE when pressed after finishing the code, will perform the following functions all in one go from verifying and compiling the code, generating the corresponding binary format to sending the binary to the claimed device over the air. Button like device button is also situated at the edge of web IDE to help manage the devices connected to the web IDE. Libraries button are located there to create the own library or explore the others library too. Docs button is there providing the total documental information about the particle devices and console button is available there in the web IDE in order to illustrate the events or simply the results created through the code executed.

The place or the page created for the execution of the desired code as well as improvised with some buttons for the modification of the code is simply the particle web IDE.

6.2 Node.js

Node.js is a runtime environment for the execution of the JavaScript programs on the command line. Node package manager (npm) is a dependency manager of library containing bundle of packages and is used to install publicly available Node.js libraries too. Node.js is available for all three major operating system from Mac OS, windows to Linux operating system.

The correct binary installable file must be downloaded from the website https://www.nodejs.org, depending upon the operating system of the PC being used. A library called particle-cli which is an open-source Node.js library is being provided by the particle which helps in the formation of command-line tool.

Depending upon the operating system, the installation of the particle CLI (Command Line Interface) is being varied. After the access of the internet, a *npm install* command must be run in the command prompt like displayed below.

For Linux: # npm install -g particle-cli [18]

For windows: D:\> npm install -g particle/cli [18]

The Node.js package named particle-cli will be installed through the command applied above as per the operating system. The package is installed globally through the command above (-g in the command), so that anybody can access the package from any directory.

Particle driver is needed for the windows user which is available at *https://s3.amazonaws.com/spark-website/Spark.zip.* [18]

The driver called spark_core.cat. must be installed in the windows operating system.

The Node.js is the required runtime environment for the management of libraries which is essential for the execution of the code for the project.

7 IoT Application

7.1 Blynk

Blynk was launched on the year 2015 and funded by Kickstarter. Specially, Blynk is the tech company which had been developing different infrastructures relating to internet of things.

Blynk is an Application Programming Interface (API) or User Interface (UI) which is said to be supporting almost all the hardware and devices. Connection to the cloud services can be obtained by using Wi-Fi, Bluetooth, GSM, Ethernet, USB (serial) etc. Device to device connection can be maintained easily through the help of bridge widget. The result or the final data of the project can be observed through emails, tweets, push notifications etc. It’s easier to use due to the direct pin manipulation and as well as history data monitoring is possible in Blynk IoT app. Almost all the IoT hardware which is connected to internet is supported by Blynk. Blynk app is marvelously designed interface builder which works smoothly on both iOS and android operating system. Lots of amazing interfaces can be developed using various widgets provides by Blynk. In figure 16, the working principle of Blynk is illustrated where it is shown that with the use of Blynk app and its server many IoT devices over the internet can be viewed or controlled through some smart phones.

Blynk can easily control both analog and digital input as well as output pins of the destined hardware i.e. microcontrollers. Virtual pins are created to send and receive data from the microcontroller to and from the blynk. Anything connected to the hardware can be controlled through the blynk. Triggering functions, converting values, reading I2C devices, controlling servo and DC motors and many more functions can be performed with this blynk application.

A blynk app is downloaded and an account must be created to get an authentication code that is needed for programming and acts as a bridge to send and receive data with the help of virtual pins. After everything is set up, with the help of the blynk app in the phone, a message or data can be transmitted and received through and from the hardware without any laptop or computer through the help of blynk server and its library which is clearly illustrated in figure 16.

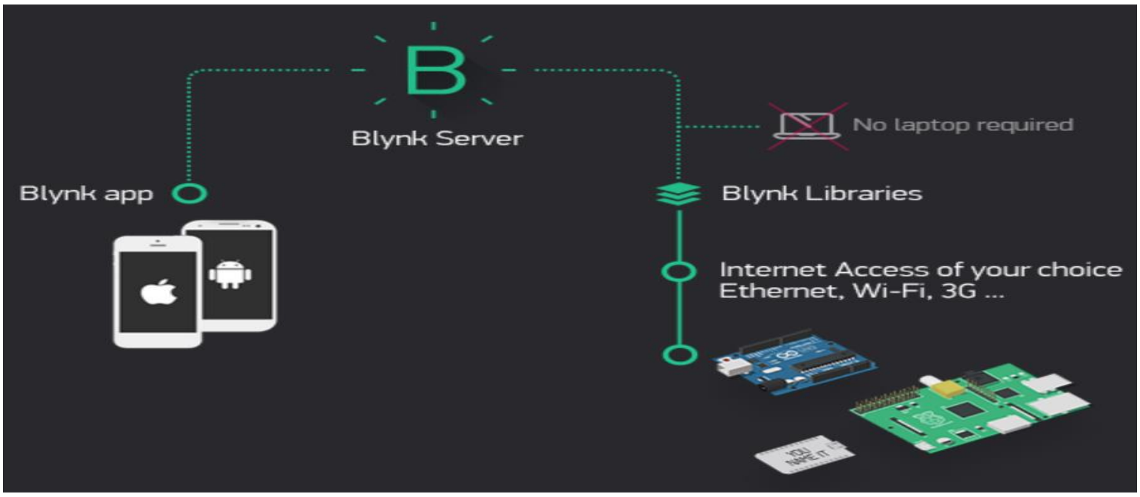


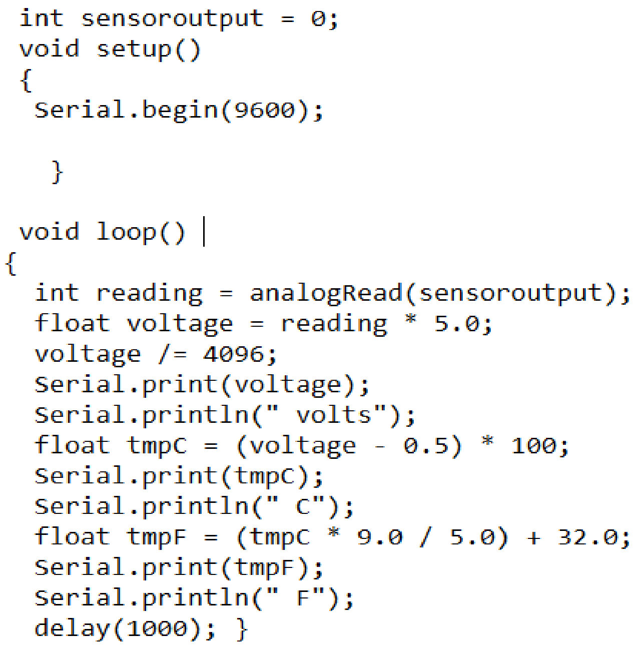
Figure 16. Working principle of Blynk [20]

8 Installations and Programming

Installations of the module were pictured in Appendix 2 showing the minimal area of the module as well as the connection between the sensors and the electron. It is displayed that the module can be packaged inside a small box letting the antenna outside for the better transmission of the data.

8.1 TMP36

TMP36 consists of three pins configuration as shown in figure 8 so the Vcc pin is connected to the 5V where the ground pin of the sensor is connected to the ground pin of the electron device. The output pin of the sensor is connected to the analog pin named A0 of the electron. Being a simple analog device, there is no requirement of the library in code. The programming of the given analog sensor is performed using the code as listed in listing 1. The listing 1 is the simple code that will display the temperature through the serial port or the command line interface.



```
 int sensoroutput = 0;
 void setup()
 {
  Serial.begin(9600);

   }

 void loop()
{
  int reading = analogRead(sensoroutput);
  float voltage = reading * 5.0;
  voltage /= 4096;
  Serial.print(voltage);
  Serial.println(" volts");
  float tmpC = (voltage - 0.5) * 100;
  Serial.print(tmpC);
  Serial.println(" C");
  float tmpF = (tmpC * 9.0 / 5.0) + 32.0;
  Serial.print(tmpF);
  Serial.println(" F");
  delay(1000); }
```

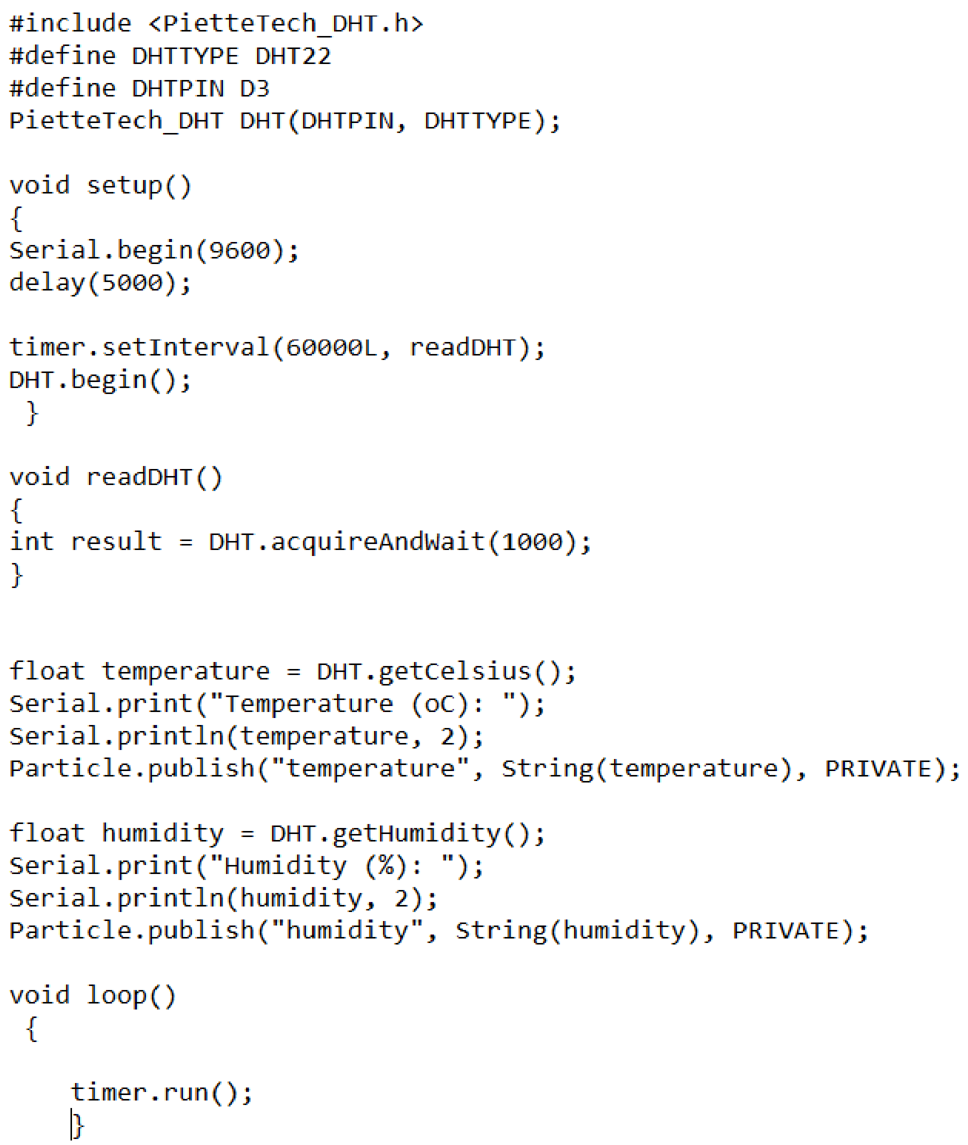
Listing 1. Programming code for TMP36

In listing 1, function setup () is used to turn on the electron device and this function also sets up the serial connection between the sensor and the electron. To run the task continuously, the function loop () is used. Inside the loop, int reading function is used to read the voltage sensor value which is later get printed by serial print () function. Functions float temC and float temF are used in code to convert the values into degree centigrade and fahrenheit respectively and again serial print() to print the converted value. Delay () function is used to call the interval for the operation happening inside the loop.

8.2 DHT22

DHT22 comprises of four pins where the first Vcc pin is connected to the 3.3 V of the electron device, the second pin is the output pin which is connected to one of the digital pins of the electron, third pin doesn’t have considerable role so it is excluded from the connection whereas the last ground pin is connected to the ground of the electron as shown in figure 12. A pullup resistor of 10 kΩ is required for the proper functioning of the module which is connected between the Vcc and output pin. The input of the microcontroller unit will be floating which must be logical high for the proper functioning of the module so pull up resistor is used to maintain the logical high state.

A library called “PietteTech_DHT.h” is created by the particle to assist the innovators and programmers for the better programming of the DHT22 / DHT11/ AM2302. With some modifications in the example code, DHT22 works great. The required main code for the operation of the DHT22 sensor is indexed as listing 2.



```
#include <PietteTech_DHT.h>
#define DHTTYPE DHT22
#define DHTPIN D3
PietteTech_DHT DHT(DHTPIN, DHTTYPE);

void setup()
{
Serial.begin(9600);
delay(5000);

timer.setInterval(60000L, readDHT);
DHT.begin();
 }

void readDHT()
{
int result = DHT.acquireAndWait(1000);
}


float temperature = DHT.getCelsius();
Serial.print("Temperature (oC): ");
Serial.println(temperature, 2);
Particle.publish("temperature", String(temperature), PRIVATE);

float humidity = DHT.getHumidity();
Serial.print("Humidity (%): ");
Serial.println(humidity, 2);
Particle.publish("humidity", String(humidity), PRIVATE);

void loop()
 {

    timer.run();
    }
```

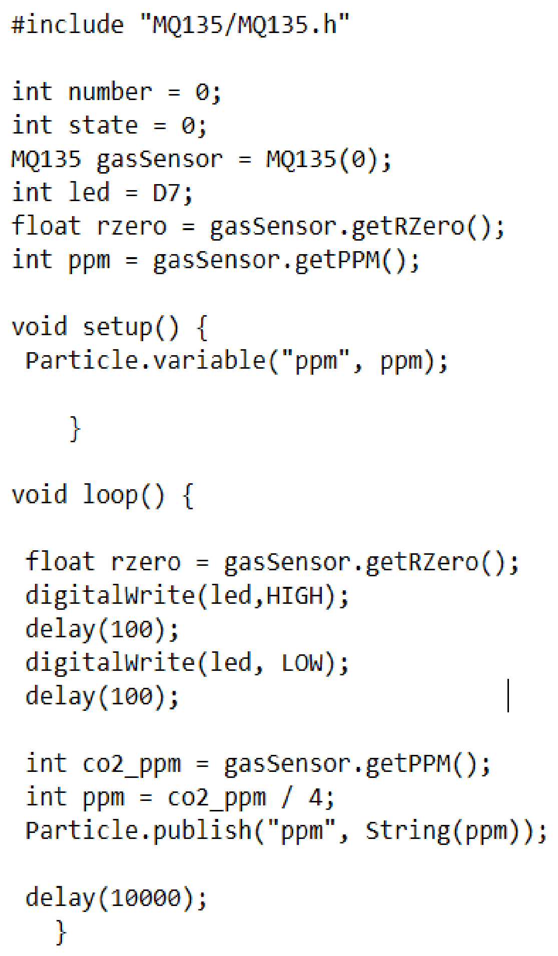
Listing 2. Programming code for DHT22

In the listing 2, #include <PietteTech_DHT.h> is the library written by the particle itself. The function #define DHTTYPE DHT22 is used to define the sensor module that is being used whereas #define DHTPIN D3 is used to define the digital pin which is connected to the output pin of the sensor. The setup () and loop () functions perform alike in listing 1. The function timer.setInterval(60000L, readDHT) is the interval to read the DHT. DHT.begin () will start the sensor whereas void readDHT () start reading the sensor and int result = DHT.acquireAndWait(1000) will check the procedure and wait for 1000 ms. The functions float temperature = DHT.getCelsius() and float humidity = DHT.getHumidity() gets the value of the tempearture and humidity whereas Serial.println(temperature,

2) and Serial.println(humidity, 2) will print the output in the serial monitor. To display the value in particle console, functions Particle.publish("temperature", String(temperature), PRIVATE) and Particle.publish("humidity", String(humidity), PRIVATE) are used. At last loop function is created where timer.run() helps the whole program to run simultaneously with the interval of the timer provided in the code.

8.3 MQ-135

As displayed in figure 13, the first pin (Vcc) is connected to the 5V of the electron, second pin (ground) is connected to the ground of the electron whereas the third pin for the digital out and fourth pin for the analog out. MQ135 was used to get data of carbon dioxide level in ppm (parts per million) using the analog pin and code used for the measurement is indexed as listing 3. In listing 3, a particle developed library is used as #include <MQ135.h>. MQ135 gas sensor is declared as MQ135 at analog pin A0 using function MQ135 gasSensor = MQ135(0). Variables are declared using int function and MQ-135 is connected to A0 of the electron. Integer ppm is declared as gas sensor ppm using the function int ppm = gasSensor.getPPM() and Particle.variable("ppm", ppm) is used to set the particle variable. Inside the loop, a float is declared in the variable is used to get the gas sensor value using function float rzero = gasSensor.getRZero() then led pin is turned on with a delay after that led pin is turned off with some delay with the functions; digital-Write(led,HIGH), delay(100), digitalWrite(led,LOW), delay(100) simultaneously. After that, the total carbon dioxide level in the surrounding will be displayed in ppm with this function int co2_ppm = gasSensor.getPPM() and it should be divided by four to get the real ppm of carbon dioxide as float voltage of electron is 4096 as compared to Arduino which is 1023 and is achieved by this function int ppm = co2_ppm / 4. Then the output is published in the particle console with this function Particle.publish("ppm", String(ppm)). At last some interval is determined with delay(10000) function.



```
#include "MQ135/MQ135.h"

int number = 0;
int state = 0;
MQ135 gasSensor = MQ135(0);
int led = D7;
float rzero = gasSensor.getRZero();
int ppm = gasSensor.getPPM();

void setup() {
 Particle.variable("ppm", ppm);

    }

void loop() {

 float rzero = gasSensor.getRZero();
 digitalWrite(led,HIGH);
 delay(100);
 digitalWrite(led, LOW);
 delay(100);

 int co2_ppm = gasSensor.getPPM();
 int ppm = co2_ppm / 4;
 Particle.publish("ppm", String(ppm));

 delay(10000);
   }
```

Listing 3. Programming code for MQ-135

9 Testing and Observations

Accuracy testing of TMP36 was performed using the environmental chamber where the temperature and humidity can be fixed. The performance or the accuracy ratio was determined using the chamber against the device to be tested with the assistance of Blynk for the output. Most of the time the temperature displayed by TMP36 is around -2 degree low than the chamber temperature. TMP36 is working perfectly well according to its datasheet where its accuracy is indexed as ±2 °C. In figure 17, the blue line is the indication of the chamber temperature whereas the orange line indicates the temperature of the TMP36.

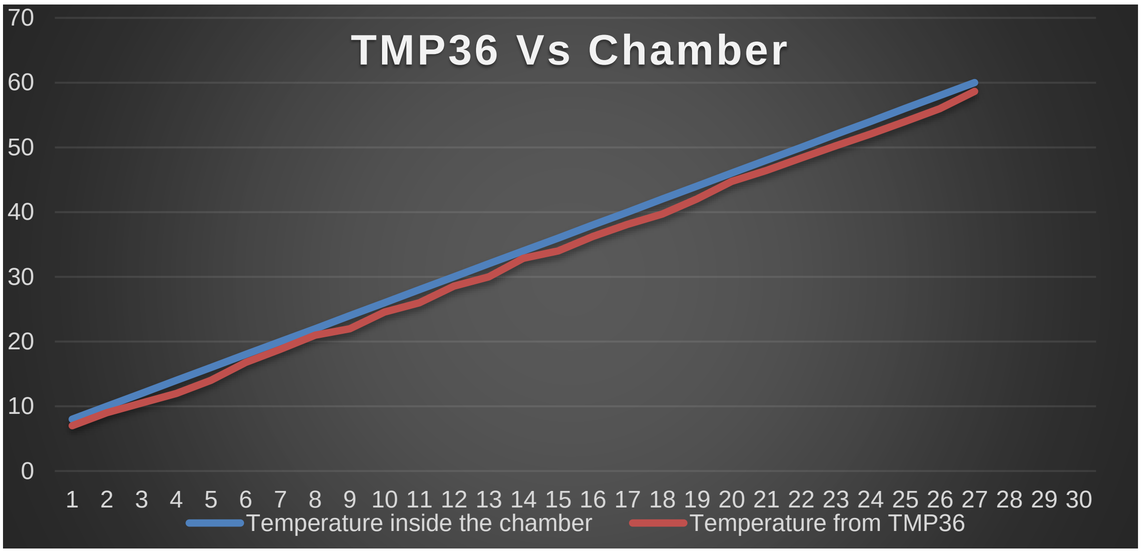


Figure 17. Accuracy testing of TMP36

According to the datasheet, the temperature accuracy of the DHT22 is in the range of ±2 °C and relative humidity is in the range of ±2 %. Alike TMP36, DHT22 has been inserted inside the chamber to know its performance level. The accuracy of the DHT22 sensor in terms of temperature and humidity is shown in figure18 and figure 19 respectively. In both figures i.e. graph, the blue line indicates the assigned value from the chamber whereas the orange lines the performance graph in terms of temperature and humidity according to the figure. It is concluded through the figure that both identities temperature and humidity are around ±2 in terms of °C and % respectively in comparison to the chamber temperature and pressure.

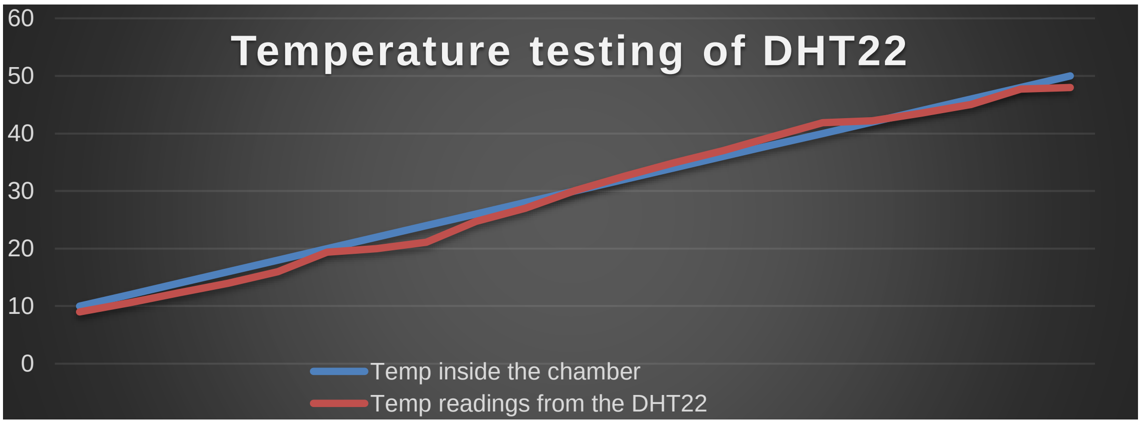


Figure 18. Temperature testing of DHT22

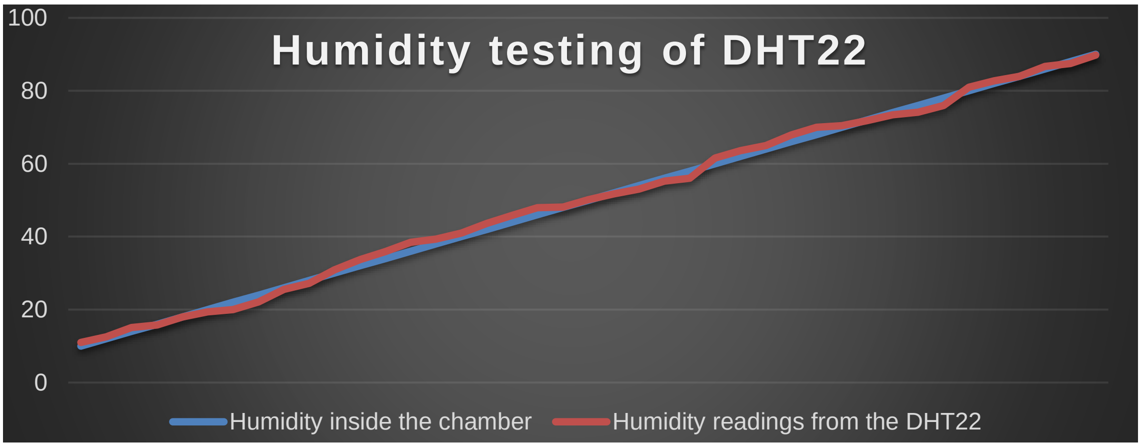


Figure 19. Humidity testing of DHT22

For the testing of MQ-135 sensor, the sensor module should be preheated before 24 hours (as stated in the datasheet) and even the output of the sensor is sometimes 0 ppm to sometimes 420 - 440 ppm. Most of the times it was challenging to get the result. Rarely, the result was around 400 to 500 ppm of carbon dioxide which means clean air, but this result was not stable so later the exclusion of this sensor was determined as it was even creating some disturbances while coding DHT22.

10 Results and Discussions

Finally, a module was created which can occupy a limited space so that it can be fixed to the bird nest without letting them notice. The main code (Appendix 1) was loaded through the particle Ideal Development Environment and the results were delightful. The module was charged through the LiPo battery, so it was not connected through the serial monitor and results were being displayed in the cell phone through the help of the Blynk application.

Again to test the functionality of the initiated module , warm air was blown through the help of a hairdryer which leads towards the increase in temperature and humidity whereas when some ice packs were introduced near the module, the temperature and humidity start decreasing rapidly and all these results were being displayed in the cell phone without the need of the laptop or desktop anymore. Even the real-time observation was performed using the Blynk application and the results were almost alike which are illustrated in Appendix 3 with the help of a graph and some data of some specified interval.

11 Comparison with previous experiment

Birdbox experiment was performed by some students of Helsinki Metropolia University of Applied Sciences in the past. To compare the results, there was no need to leave the school perimeter as some reports of their achievement was been introduced.

The performed experiment of the fellow student “Dibas Bhandari” named “Intelligent Nestbox” seems magnificent with the combination of IR sensor and Webcam too. The results obtained were impressive with the help of Lab-View and Arduino Uno as a microcontroller. The only unacceptable perimeter of that experiment was that the whole experiment was connected through a serial port of the laptop or desktop which means the far way bird nest environment cannot get monitored due to the lack of even Wi-Fi controlled module.

This experiment with the particle device i.e. photon and electron, the first one with the Wi-Fi module and the second one with the GSM module, with the cooperation of Blynk application, the far-away bird nest environment can be monitored even without the use of the laptop or desktop through the help of GSM module. For the short distance, the WI-Fi module i.e. photon is applicable whereas for the long-distance where Wi-Fi is not available, the GSM module i.e. electron is suitable to receive the information of the desired environment.

12 Future possibilities

This project can be an initiative tool for the upcoming projects to monitor not only birds but also animals and plants too. Many birds, animals, and plants are being extinct, and some are listed as endangered species. The further modification is needed for rewarding results.

Improvements like adding fast responsive components i.e. sensors, using a webcam, online audio-video transmission devices, etc. can be included in the further project. Setting up the threshold temperature, humidity, carbon dioxide level and initiating an alarm or buzzer in case of exceeding threshold value or case of intruders will be a remarkable step for the future. Adjusting the temperature, humidity as well as light intensity through the help of cell phones will be a considerable move in the future.

A similar type of innovation with some advancement will benefit the people to monitor the different states of birds and animals like breeding, hibernation, feeding, and countless information about them. With the advancement in technology and IoT devices, hope no more birds, animals, and plants will again be in the endangered list.

13 Conclusion

The intended goal of the entire project was fulfilled by creating a Nest Box with the coordination between the particle IoT devices, sensors, and the Blynk application. Initially, the code created for the sensors was tested individually with the help of a particle Wi-Fi module named photon. The experiment for the photon was implemented using the serial monitor as well as a command-line interface with the use of Wi-Fi. After obtaining the elegant result on a photon, the sensors with their respective code were tested on the particle GSM module called electron and was successful to get the splendid results that are displayed on the cell phone with the help of Blynk application.

Initially, the project seems exasperating to use particle devices due to very less resources as compared to Arduino and the particle community seems restrained in most of the cases. Multiple other difficulties were encountered and most of them were solved which leads to the accomplishment of this project. The testing of the Nest Box under the real environment was not attainable, as it was time-consuming and somewhat risky too, but the prime goal was achieved in the end.

With some further improvements in this project and with the coordination of ornithologists, this project will be working better in real environment i.e. Nest Box.

References

[1] Wikipedia, ”Nest Box,” [Online].
URL: https://en.wikipedia.org/wiki/Nest_box.
[Accessed: 2nd September 2019].

[2] Wikipedia, ”Sensor,” [Online].
URL: https://en.wikipedia.org/wiki/Sensor.
[Accessed: 2nd September 2019].

[3] Wikipedia, ”Temperature,” [Online].
URL: https://en.wikipedia.org/wiki/Temperature.
[Accessed: 4rth September 2019].

[4] T. G. B. a. N. L. Buck, Manufacturing Technology.

[5] M. Fischer, ”Sensor”.

[6] Sensortipsadmin, ”Designing with Semiconductor Temperature Sensors,” 2009. [Online].
URL:https://www.sensortips.com/temperature/designing-with-semiconductor-temperature-sensors/.
[Accessed: 9th September 2019].

[7] www.analog.com, ”Analog Devices,” [Online].
URL: https://www.analog.com/en/products/tmp36.html#product-overview.
[Accessed: 10th September 2019].

[8] Swagatam(homemade-circuits), ”LM35 Pinout, Datasheet, Application Circuit,” [Online].
URL: https://www.homemade-circuits.com/lm35-circuit/.
[Accessed: 10th September 2019].

[9] Designspark, ”Measuring Temperature Accurately with Semiconductor Sensors,” [Online].
URL: https://www.rs-online.com/designspark/measuring-temperature-accurately-with-semiconductor-sensors.
[Accessed:10th September 2019].

[10] Electronicsforu.com, ”Humidity Sensor: Basics, Usage, Parameters and Applications,” [Online].

URL:https://electronicsforu.com/resources/electronics-components/humidity-sensor-basic-usage-parameter.
[Accessed: 12th September 2019].

[11] Anusha, ”Humidity Sensor – Types and Working Principle,” June 2007. [Online].
URL:https://www.electronicshub.org/humidity-sensor-types-working-principle/#Advantages_of_Capacitive_Humidity_Sensors.
[Accessed: 13th September 2019].

[12] Electronicshub.org, ”Working of Thermal Conductivity Humidity Sensors,” [Online].
URL:https://www.electronicshub.org/humidity-sensor-types-working-principle/#Advantages_of_Capacitive_Humidity_Sensors.
[Accessed: 15th September 2019].

[13] ”Introduction to DHT22,” [Online].
URL: https://www.theengineeringprojects.com/2019/02/introduction-to-dht22.html.
[Accessed: 17th September 2019].

[14] M. N. H. A. K. Z. A. Zainab Yunusa, ”Gas Sensors: A Review,” *Sensors and transducers,* p. 75, 2014 .

[15] Components101, ”MQ-135 - Gas Sensor for Air Quality,” [Online].
URL:https://components101.com/sensors/mq135-gas-sensor-for-air-quality.
[Accessed: 20th September 2019].

[16] TechTarget, ”internet of things(IoT),” [Online].
URL:https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.
[Accessed: 25th September 2019].

[17] i-SCOOP, ”What is the Internet of Things? Internet of Things definitions,” [Online].
URL: https://www.i-scoop.eu/internet-of-things/.
[Accessed: 26th September 2019].

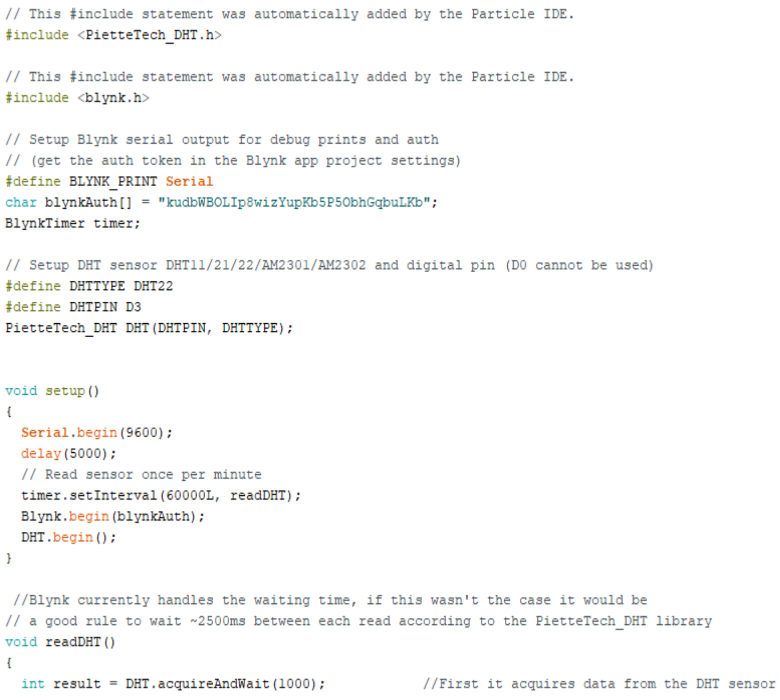
[18] K. G. A. V. Rashhid Khan, Learning IoT with Particle Photon and Electron, Birmingham: Packt Publishing Ltd., 2016.

[19] Particle, ”Particle docs,” Particle, 2019. [Online].
URL:https://docs.particle.io/datasheets/cellular/electron-datasheet/#functional-description.
[Accessed: 27th September 2019].

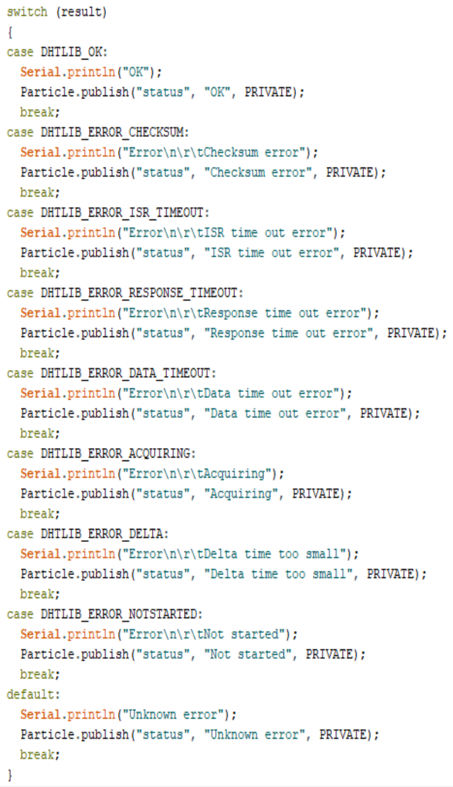
[20] B. Inc., ”Blynk,” 2019. [Online].
Available: https://blynk.io/.
[Accessed: 29th September 2019].

Appendix 1.

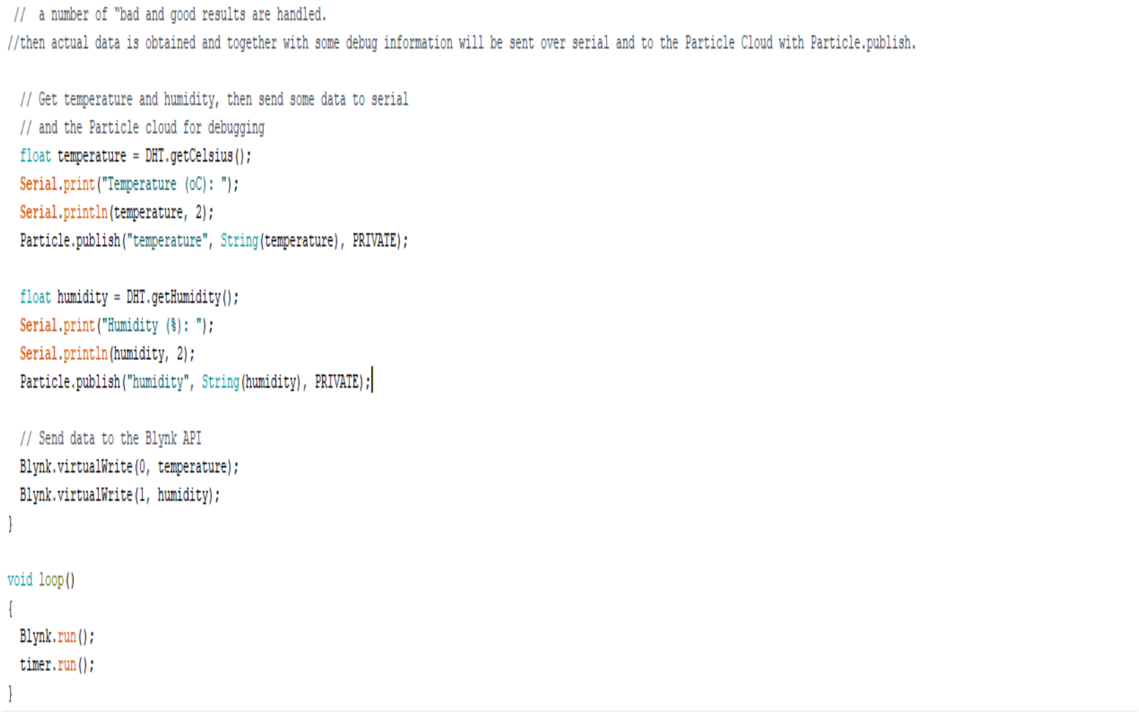
Main Code



```
// This #include statement was automatically added by the Particle IDE.
#include <PietteTech_DHT.h>

// This #include statement was automatically added by the Particle IDE.
#include <blynk.h>

// Setup Blynk serial output for debug prints and auth
// (get the auth token in the Blynk app project settings)
#define BLYNK_PRINT Serial
char blynkAuth[] = "kudbWBOLIp8wizYupKb5P5ObhGqbuLKb";
BlynkTimer timer;

// Setup DHT sensor DHT11/21/22/AM2301/AM2302 and digital pin (D0 cannot be used)
#define DHTTYPE DHT22
#define DHTPIN D3
PietteTech_DHT DHT(DHTPIN, DHTTYPE);


void setup()
{
  Serial.begin(9600);
  delay(5000);
  // Read sensor once per minute
  timer.setInterval(60000L, readDHT);
  Blynk.begin(blynkAuth);
  DHT.begin();
}

 //Blynk currently handles the waiting time, if this wasn't the case it would be
// a good rule to wait ~2500ms between each read according to the PietteTech_DHT library
void readDHT()
{
  int result = DHT.acquireAndWait(1000);            //First it acquires data from the DHT sensor
```



```
 switch (result)
 {
 case DHTLIB_OK:
   Serial.println("OK");
   Particle.publish("status", "OK", PRIVATE);
   break;
 case DHTLIB_ERROR_CHECKSUM:
   Serial.println("Error\n\r\tChecksum error");
   Particle.publish("status", "Checksum error", PRIVATE);
   break;
 case DHTLIB_ERROR_ISR_TIMEOUT:
   Serial.println("Error\n\r\tISR time out error");
   Particle.publish("status", "ISR time out error", PRIVATE);
   break;
 case DHTLIB_ERROR_RESPONSE_TIMEOUT:
   Serial.println("Error\n\r\tResponse time out error");
   Particle.publish("status", "Response time out error", PRIVATE);
   break;
 case DHTLIB_ERROR_DATA_TIMEOUT:
   Serial.println("Error\n\r\tData time out error");
   Particle.publish("status", "Data time out error", PRIVATE);
   break;
 case DHTLIB_ERROR_ACQUIRING:
   Serial.println("Error\n\r\tAcquiring");
   Particle.publish("status", "Acquiring", PRIVATE);
   break;
 case DHTLIB_ERROR_DELTA:
   Serial.println("Error\n\r\tDelta time too small");
   Particle.publish("status", "Delta time too small", PRIVATE);
   break;
 case DHTLIB_ERROR_NOTSTARTED:
   Serial.println("Error\n\r\tNot started");
   Particle.publish("status", "Not started", PRIVATE);
   break;
 default:
   Serial.println("Unknown error");
   Particle.publish("status", "Unknown error", PRIVATE);
   break;
 }
```



```
 //  a number of "bad and good results are handled.
//then actual data is obtained and together with some debug information will be sent over serial and to the Particle Cloud with Particle.publish.

  // Get temperature and humidity, then send some data to serial
  // and the Particle cloud for debugging
  float temperature = DHT.getCelsius();
  Serial.print("Temperature (oC): ");
  Serial.println(temperature, 2);
  Particle.publish("temperature", String(temperature), PRIVATE);

  float humidity = DHT.getHumidity();
  Serial.print("Humidity (%): ");
  Serial.println(humidity, 2);
  Particle.publish("humidity", String(humidity), PRIVATE);

  // Send data to the Blynk API
  Blynk.virtualWrite(0, temperature);
  Blynk.virtualWrite(1, humidity);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

Appendix 2.

Installations of the sensors with the electron

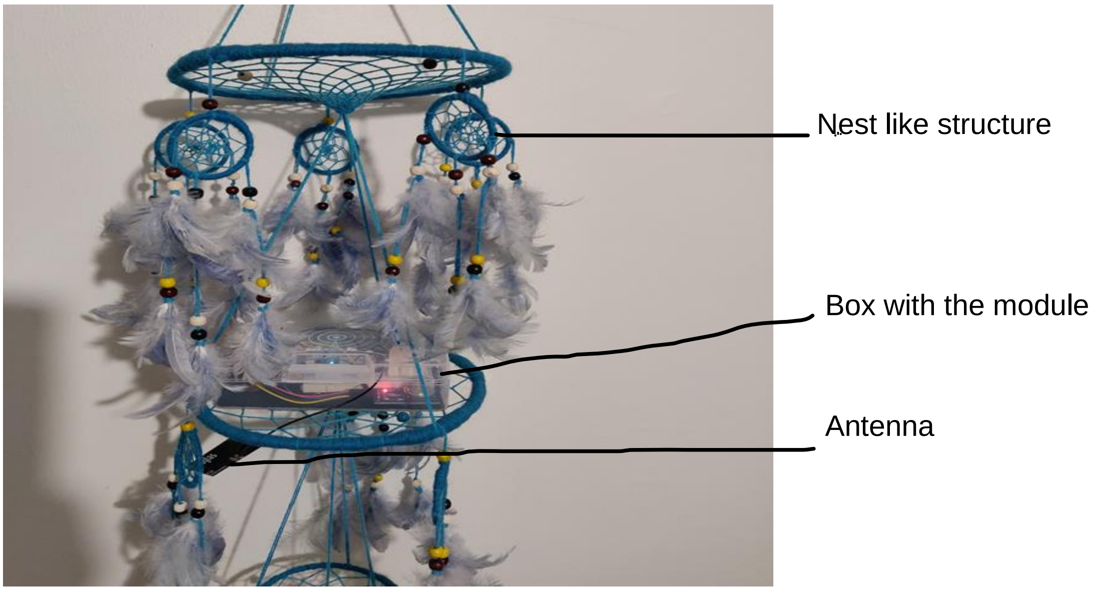


Figure 20. Picture displaying the minimal area required by the whole setup inside a nest like object

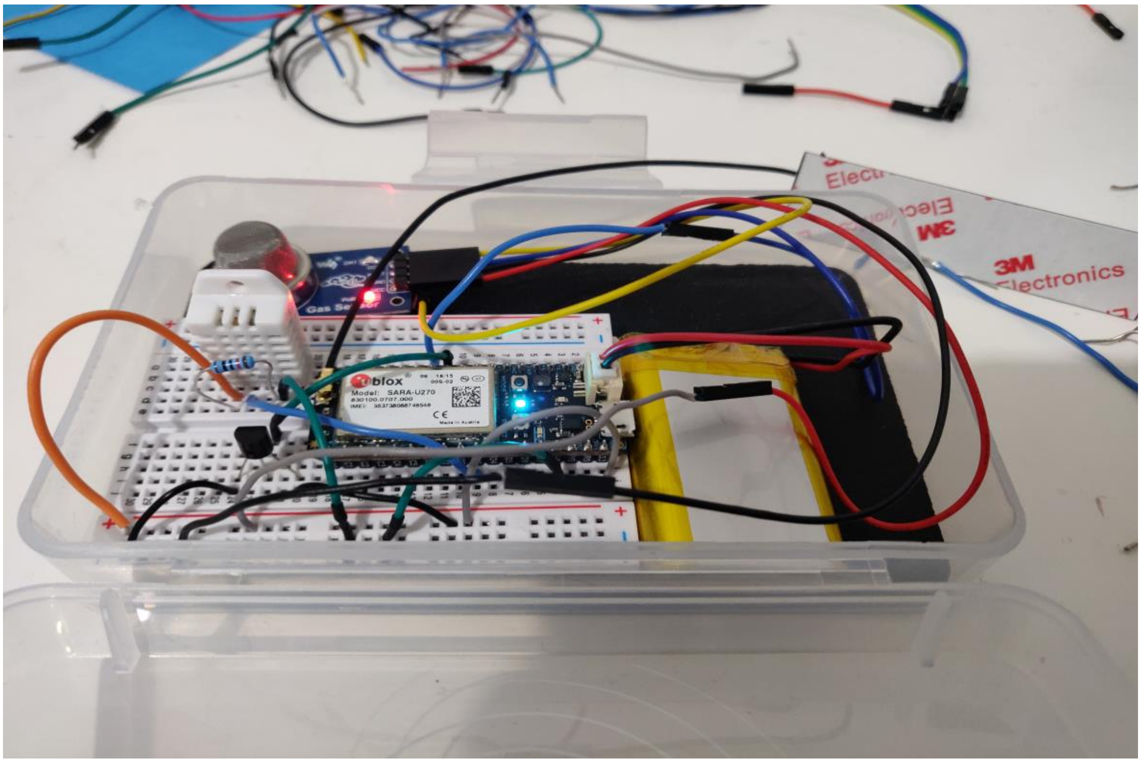


Figure 21. Installation of the sensors with the Electron

Appendix 3.

Results on Blynk

Output observed through Blynk when the real temperature outside was -4°C and humidity 95 % on December second at 20:47 as displayed in the figure below.

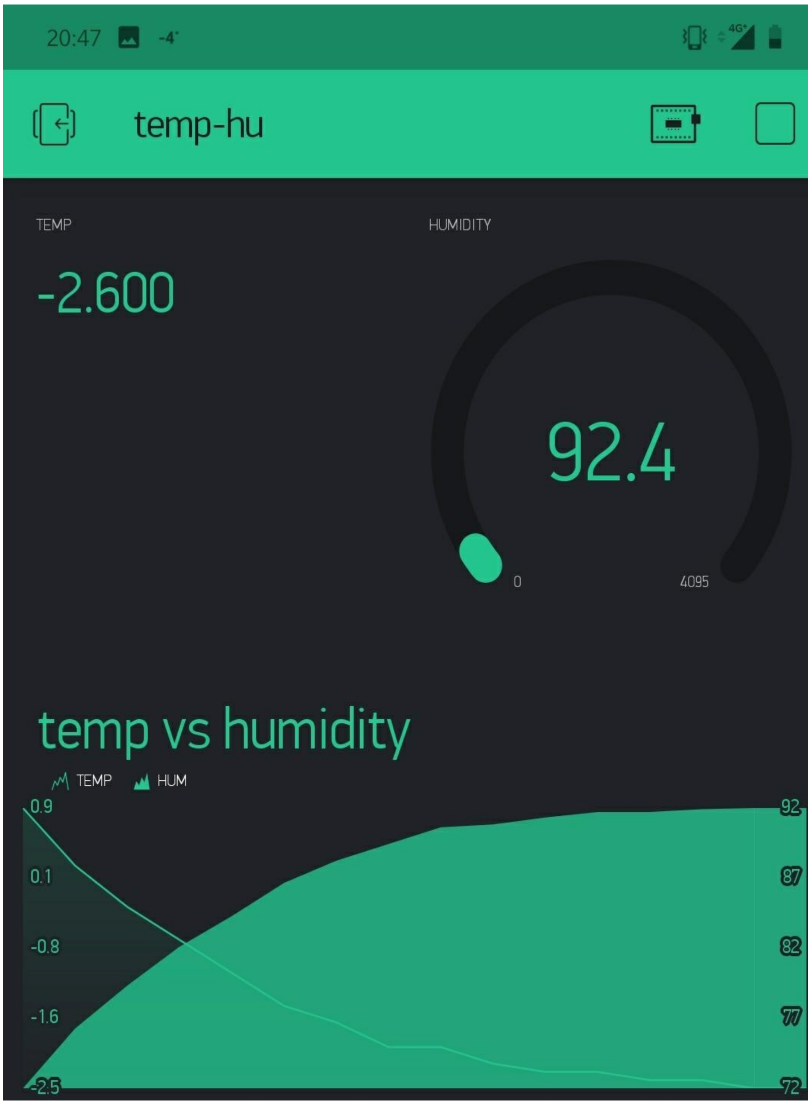


Figure 22. The instant temperature and humidity at 8:47 pm of December 2nd

The result displayed through Blynk in the one-hour interval when it was cooled and then heated again with great fluctuation in temperature and humidity can be observed below. The graph screenshot through the phone clearly shows at 8:49 pm, the temperature was lowest nearly equaling to -3 °C which was nearly equal to the temperature shown through the weather application in the mobile notification and humidity around 95%. And there's the maximum temperature gained when the module was heated nearby an oven with a temperature of around 49 °C and humidity 99.98% as clearly pictured in figure 23.

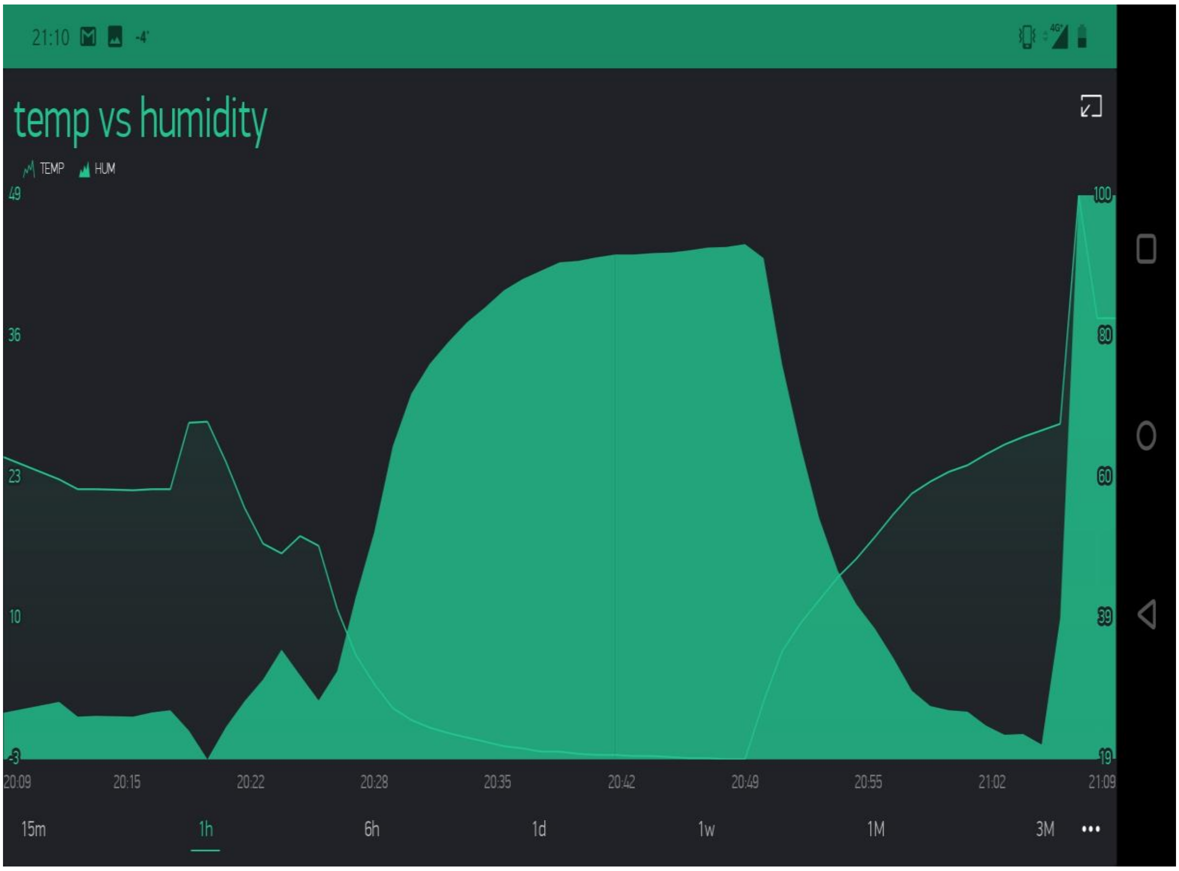


Figure 23. The results of one-hour of December 2nd evening from 8:09 pm to 9:09 pm inside and outside of the apartment simultaneously