CHAPTER 1
INTRODUCTION

Internet of Things (IoT) is an emerging technology that provides smarter services to users by interconnecting various devices via Internet and allowing these devices to exchange information with each other for some common goal [1,2].'Internet-of-Things' is a term that covers many aspects related to the extensibility of the Internet into the physical realm, by deploying specially distributed devices with embedded identification, actuation or sensing capabilities. IoT anticipates a future in which physical and digital entities can be linked, by means of appropriate information and communication technologies, to enable a whole new class of applications and services [3].

As the world is trending into new technologies and implementing it is a necessary goal to trend up in agriculture also. Many researches are done in the field of agriculture and many projects signify the use of wireless sensor network collect data from different sensors deployed at various nodes and send it through the wireless protocol [4,5]. The IoT can take advantage of cloud computing devices to generate a computing system from sensors to tools that monitor data from agricultural fields and from human on ground actors and accordingly transmit the data into the repositories[7]. IoT provides smart agricultural solutions to farmers for better yield by collecting the data from different sensors, thereby informing about the various environmental factors [8].

Climate changes with its adverse effects have been erratic over the past few decades. Due to this in recent era, climate-smart methods like smart agriculture using IoT needs to be adopted by many farmers. Monitoring the environmental factors is not the complete solution to increase the yield of crops. There are number of other factors that decrease the productivity largely. Hence, automation must be implemented in agriculture to overcome these problems. IOT is developing rapidly and widely applied in all wireless environments.

1.1 INTERNET OF THINGS

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. An abundance of conferences, reports, and news articles discuss and debate the prospective impact of the “IoT revolution”—from new market opportunities and business models to concerns about security, privacy, and technical interoperability**.**

The large-scale implementation of IoT devices promises to transform many aspects of the way we live. For consumers, new IoT products like Internet-enabled appliances, home automation components, and energy management devices are moving us toward a vision of the "smart home'', offering more security and energy efficiency. Other personal IoT devices like wearable fitness and health monitoring devices and network enabled medical devices are transforming the way healthcare services are delivered. This technology promises to be beneficial for people with disabilities and the elderly, enabling improved levels of independence and quality of life at a reasonable cost. IoT systems like networked vehicles, intelligent traffic systems, and sensors embedded in roads and bridges move us closer to the idea of "smart cities'', which help minimize congestion and energy consumption. IoT technology offers the possibility to transform agriculture, industry, and energy production and distribution by increasing the availability of information along the value chain of production using networked sensors. However, IoT raises many issues and challenges that need to be considered and addressed in order for potential benefits to be realized.

1.1.1 HISTORY

The term "Internet of Things" (IoT) was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors. Ashton coined the term to illustrate the power of connecting Radio-Frequency Identification (RFID) tags used in corporate supply chains to the Internet in order to count and track goods without the need for human intervention. Today, the Internet of Things has become a popular term for describing scenarios in which Internet connectivity and computing capability extend to a variety of objects, devices, sensors, and everyday items.

While the term "Internet of Things" is relatively new, the concept of combining computers and networks to monitor and control devices has been around for decades. By the late 1970s, for example, systems for remotely monitoring meters on the electrical grid via telephone lines were already in commercial use.

1.1.2 DIFFERENT DEFINITION, SIMILAR CONCEPT

Despite the global buzz around the Internet of Things, there is no single, universally accepted definition for the term. Different definitions are used by various groups to describe or promote a particular view of what IoT means and its most important attributes. Some definitions specify the concept of the Internet or the Internet Protocol (IP), while others, perhaps surprisingly, do not. For example, consider the following definitions.

The Internet Architecture Board (IAB) begins RFC 7452, “Architectural Considerations in Smart Object Networking’’, with this description:

The term "Internet of Things" (IoT) denotes a trend where a large number of embedded devices employ communication services offered by the Internet protocols. Many of these devices, often called "smart objects,’’ are not directly operated by humans, but exist as components in buildings or vehicles, or are spread out in the environment.

Within the Internet Engineering Task Force (IETF), the term “smart object networking” is commonly used in reference to the Internet of Things. In this context, “smart objects” are devices that typically have significant constraints, such as limited power, memory, and processing resources, or bandwidth. Work in the IETF is organized around specific requirements to achieve network interoperability between several types of smart objects.

Internet of things (IoT): A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

Note1-Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

Note 2- From a broader perspective, the IoT can be perceived as a vision with technological and societal implication.

The Internet of Things (IoT) is a framework in which all things have a representation and a presence in the Internet. More specifically, the Internet of Things aims at offering new applications and services bridging the physical and virtual worlds, in which Machine-to-Machine (M2M) communications represents the baseline communication that enables the interactions between Things and applications in the cloud.

All of the definitions describe scenarios in which network connectivity and computing capability extends to a constellation of objects, devices, sensors, and everyday items that are not ordinarily considered to be “computers’’; this allows the devices to generate, exchange, and consume data, often with minimal human intervention. The various definitions of IoT do not necessarily disagree – rather they emphasize different aspects of the IoT phenomenon from different focal points and use cases.

However, the disparate definitions could be a source of confusion in dialogue on IoT issues, particularly in discussions between stakeholder groups or industry segments. Similar confusion was experienced in recent years about net neutrality and cloud computing, where different interpretations of the terms sometimes presented obstacles to dialogue. While it is probably unnecessary to develop

a single definition of IoT, it should be recognized that there are different perspectives to be factored into discussions.

1.1.3 INTERNET OF THINGS COMMUNICATION MODEL

From an operational perspective, it is useful to think about how IoT devices connect and communicate in terms of their technical communication models. In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects (RFC 7452), which outlines a framework of four common communication models used by IoT devices. The discussion below presents this framework and explains key characteristics of each model in the framework.

1.1.3.1 Device-to-Device Communications

The device-to-device communication model represents two or more devices that directly connect and communicate between one another, rather than through an intermediary application server. These devices communicate over many types of networks, including IP networks or the Internet. Often, however these devices use protocols like Bluetooth, Z-Wave, or ZigBee to establish direct device-to-device communications, as shown in Figure 1.1.

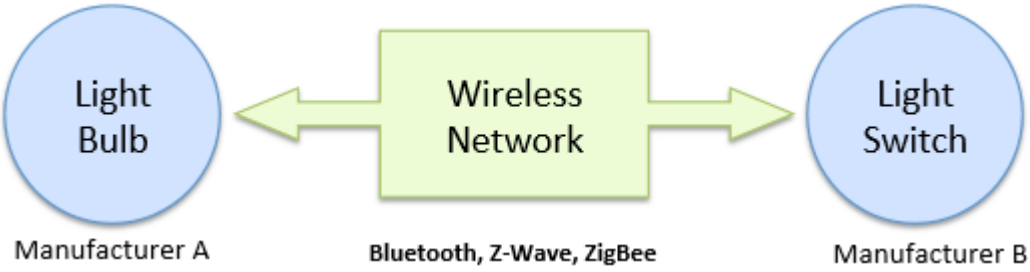


Fig. 1.1 Example of device-to-device communication model.

These device-to-device networks allow devices that adhere to a particular communication protocol to communicate and exchange messages to achieve their function. This communication model is commonly used in applications like home automation systems, which typically use small data packets of information to communicate between devices with relatively low data rate requirements. Residential IoT devices like light bulbs, light switches, thermostats, and door locks normally send small amounts of information to each other (e.g. a door lock status message or turn on light command) in a home automation scenario.

From the user's point of view, this often means that underlying device-to-device communication protocols are not compatible, forcing the user to select a family of devices that employ a common protocol. For example, the family of devices using the Z-Wave protocol is not natively compatible

with the ZigBee family of devices. While these incompatibilities limit user choice to devices within a particular protocol family, the user benefits from knowing that products within a particular family tend to communicate well.

1.1.3.2 Device-To-Cloud Communications

In a device-to-cloud communication model, the IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic. This approach frequently takes advantage of existing communications mechanisms like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service. This is shown in Figure 2.2.

Fig 1.2 Device-to-cloud communication model diagram.

Some popular consumer IoT devices like the Nest Labs Learning Thermostat and the Samsung Smart TV employ this communication model. In the case of the Nest Learning Thermostat, the device transmits data to a cloud database where the data can be used to analyse home energy consumption. Further, this cloud connection enables the user to obtain remote access to their thermostat via a smartphone or Web interface, and it also supports software updates to the thermostat. Similarly with the Samsung Smart TV technology, the television uses an Internet connection to transmit user viewing information to Samsung for analysis and to enable the interactive voice recognition features of the TV.

However, interoperability challenges can arise when attempting to integrate devices made by different manufacturers. Frequently, the device and cloud service are from the same vendor. If proprietary data protocols are used between the device and the cloud service, the device owner or user may be tied to a specific cloud service, limiting or preventing the use of alternative service providers. This is commonly referred to as “vendor lock-in’’, a term that encompasses other facets of the relationship with the provider such as ownership of and access to the data. At the same time, users can generally have confidence that devices designed for the specific platform can be integrated.

1.1.3.3 Device-To-Gateway Model

In the device-to-gateway model, or more typically, the device-to-application-layer gateway (ALG) model, the IoT device connects through an ALG service as a conduit to reach a cloud service. In simpler terms, this means that there is application software operating on a local gateway device, which acts as an intermediary between the device and the cloud service and provides security and other functionality such as data or protocol translation. The model is shown in Figure 2.3.

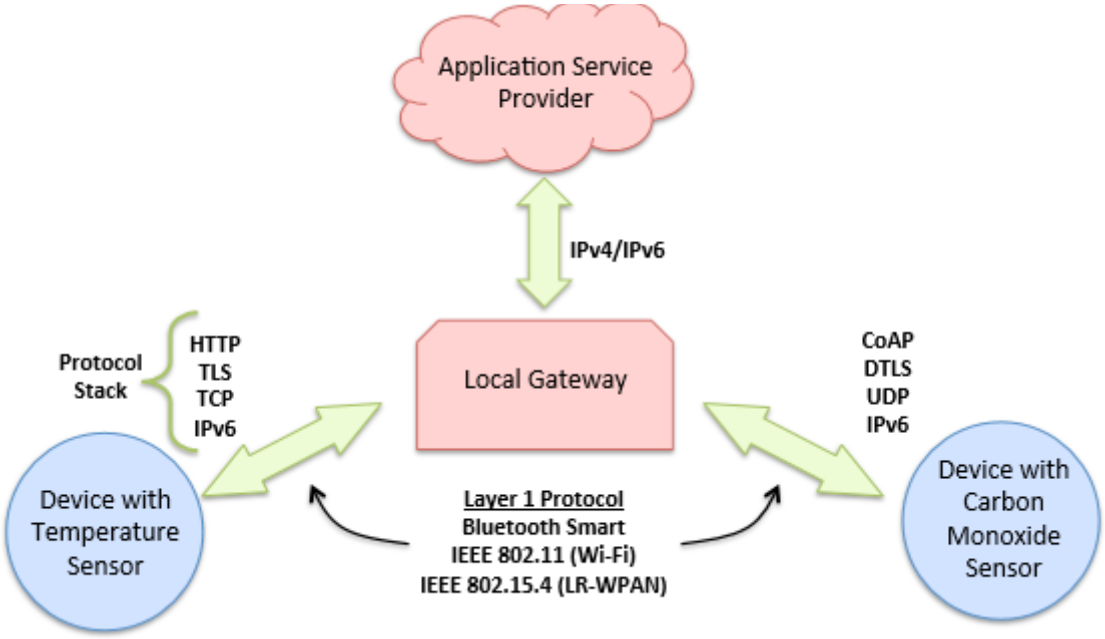


Fig. 1.3 Device-to-gateway communication model diagram.

several forms of this model are found in consumer devices. In many cases, the local gateway device is a Smartphone running an app to communicate with a device and relay data to a cloud service.

The other form of this device-to-gateway model is the emergence of “hub” devices in home automation applications. These are devices that serve as a local gateway between individual IoT devices and a cloud service, but they can also bridge the interoperability gap between devices themselves. For example, the Smart Things hub is a stand-alone gateway device that has Z-Wave and ZigBee transceivers installed to communicate with both families of devices. It then connects to the Smart Things cloud service, allowing the user to gain access to the devices using a Smartphone app and an Internet connection.

This [communication model] is used in situations where the smart objects require interoperability with non-IP [Internet protocol] devices. Sometimes this approach is taken for integrating IPv6-only devices, which means a gateway is necessary for legacy IPv4-only devices and services.48.

In other words, this communications model is frequently used to integrate new smart devices into a legacy system with devices that are not natively interoperable with them. A downside of this

approach is that the necessary development of the application-layer gateway software and system adds complexity and cost to the overall system.

The IAB's RFC7452 document suggests the outlook for this model:

It is expected that in the future, more generic gateways will be deployed to lower cost and infrastructure complexity for end consumers, enterprises, and industrial environments. Such generic gateways are more likely to exist if IoT device designs make use of generic Internet protocols and not require application-layer gateways that translate one application-layer protocol to another one. The use of application-layer gateways will, in general, lead to a more fragile deployment, as has been observed in the past.

1.1.3.4 Back-End-Data-Sharing Model

The back-end data-sharing model refers to a communication architecture that enables users to export and analyse smart object data from a cloud service in combination with data from other sources. This architecture supports "the [user's] desire for granting access to the uploaded sensor data to third parties". This approach is an extension of the single device-to-cloud communication model, which can lead to data silos where "IoT devices upload data only to a single application service provider''. A back-end sharing architecture allows the data collected from single IoT device data streams to be aggregated and analysed.

For example, a corporate user in charge of an office complex would be interested in consolidating and analysing the energy consumption and utilities data produced by all the IoT sensors and Internet-enabled utility systems on the premises. Often in the single device-to-cloud model, the data each IoT sensor or system produces sits in a stand-alone data silo. An effective back-end data sharing architecture would allow the company to easily access and analyse the data in the cloud produced by the whole spectrum of devices in the building. Also, this kind of architecture facilitates data portability needs. Effective back-end data sharing architectures allow users to move their data when they switch between IoT services, breaking down traditional data silo barriers.

The back-end data-sharing model suggests a federated cloud services approach52 or cloud applications programmer interfaces (APIs) are needed to achieve interoperability of smart device data hosted in the cloud.53 A graphical representation of this design is shown in Figure 2.4.

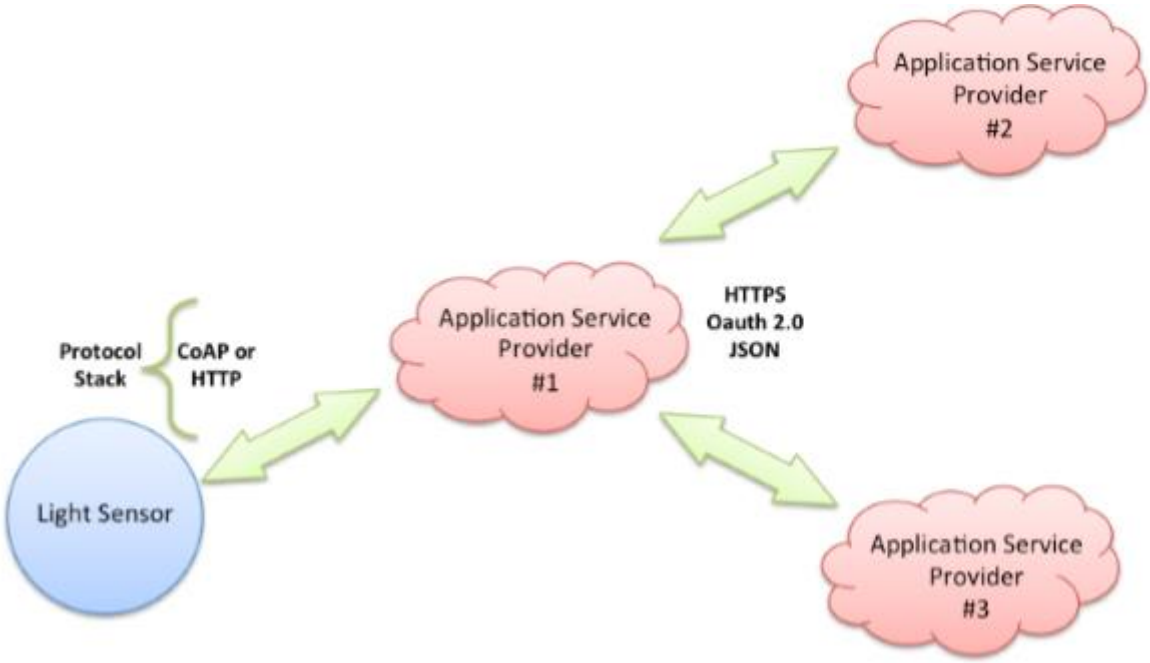


Fig 1.4 Back-end data sharing model diagram

1.1.4 SECURITY ISSUES

1.1.4.1 The IoT Security Challenges

As we note in the principles that guide our work, ensuring the security, reliability, resilience, and stability of Internet applications and services is critical to promoting trust and use of the Internet. As users of the Internet, we need to have a high degree of trust that the Internet, its applications, and the devices linked to it are secure enough to do the kinds of activities we want to do online in relation to the risk tolerance associated with those activities. The Internet of Things is no different in this respect, and security in IoT is fundamentally linked to the ability of users to trust their environment. If people don't believe their connected devices and their information are reasonably secure from misuse or harm, the resulting erosion of trust causes a reluctance to use the Internet. This has global consequences to electronic commerce, technical innovation, free speech, and practically every other aspect of online activities. Indeed, ensuring security in IoT products and services should be considered a top priority for the sector.

As we increasingly connect devices to the Internet, new opportunities to exploit potential security vulnerabilities grow. Poorly secured IoT devices could serve as entry points for cyberattack by allowing malicious individuals to re-program a device or cause it to malfunction. Poorly designed devices can expose user data to theft by leaving data streams inadequately protected. Failing or malfunctioning devices also can create security vulnerabilities

Along with potential security design deficiencies, the sheer increase in the number and nature of IoT devices could increase the opportunities of attack. When coupled with the highly interconnected

nature of IoT devices, every poorly secured device that is connected online potentially affects the security and resilience of the Internet globally, not just locally. For example, an unprotected refrigerator or television in the US that is infected with malware might send thousands of harmful spam emails to recipients worldwide using the owner's home Wi-Fi Internet connection.

When thinking about Internet of Things devices, it is important to understand that security of these devices is not absolute. IoT device security is not a binary proposition of secure or insecure. Instead, it is useful to conceptualize IoT security as a spectrum of device vulnerability. The spectrum ranges from totally unprotected devices with no security features to highly secure systems with multiple layers of security features. In an endless cat-and-mouse game, new security threats evolve, and device manufacturers and network operators continuously respond to address the new threats.

The overall security and resilience of the Internet of Things is a function of how security risks are assessed and managed. Security of a device is a function of the risk that a device will be compromised, the damage such compromise will cause, and the time and resources required to achieve a certain level of protection. If a user cannot tolerate a high degree of security risk as in the case of the operator of a traffic control system or person with an implanted, Internet-enabled medical device, then she may feel justified in spending a considerable amount of resources to protect the system or device from attack.

Several factors influence this risk assessment and mitigation calculation. Factors include having a clear understanding of the present security risks and the potential future risks; the estimated economic and other costs of harm if the risks are realized; and the estimated cost to mitigate the risks. While these kinds of security trade-offs are often made from an individual user or organizational perspective, it is also important to consider the interrelatedness of IoT devices as part of a larger IoT ecosystem. The networked connectivity of IoT devices means that security decisions made locally about an IoT device can have global impacts on other devices.

As a matter of principle, developers of smart objects for the Internet of Things have an obligation in ensuring that those devices do not expose either their own users or others to potential harm. As a matter of business and economics, vendors have an interest in reducing their cost, complexity, and time to market. For example, IoT devices that are high–volume, low–margin components that already represent a cost added to that of the product in which they are embedded are becoming quite common; adding more memory and a faster processor to implement security measures could easily make that product commercially uncompetitive.

In economic terms, lack of security for IoT devices results in a negative externality, where a cost is imposed by one party (or parties) on other parties. A classic example is pollution of the environment, where the environmental damage and clean-up costs (negative externalities) of a polluter's actions

are borne by other parties. The issue is that the cost of the externality imposed on others is not normally factored into the decision-making process, unless, as is the case with pollution, a tax is imposed on the polluter to convince him to lower the amount of pollution. In the case of information security, as discussed by Bruce Schneier, an externality arises when the vendor creating the product does not bear the costs caused by any insecurity; in this case, liability law can influence vendors to account for the externality and develop more security products.

1.1.4.2 UNIQUE SECURITY CHALLENGES OF IOT DEVICES:

- Many Internet of Things devices, such as sensors and consumer items, are designed to be deployed at a massive scale that is orders of magnitude beyond that of traditional Internet-connected devices. As a result, the potential quantity of interconnected links between these devices is unprecedented. Further, many of these devices will be able to establish links and communicate with other devices on their own in an unpredictable and dynamic fashion. Therefore, existing tools, methods, and strategies associated with IoT security may need new consideration.
- Many IoT deployments will consist of collections of identical or near identical devices. This homogeneity magnifies the potential impact of any single security vulnerability by the sheer number of devices that all have the same characteristics. For example, a communication protocol vulnerability of one company's brand of Internet-enabled light bulbs might extend to every make and model of device that uses that same protocol or which shares key design or manufacturing characteristics.
- Many Internet of Things devices will be deployed with an anticipated service life many years longer than is typically associated with high-tech equipment. Further, these devices might be deployed in circumstances that make it difficult or impossible to reconfigure or upgrade them; or these devices might outlive the company that created them, leaving orphaned devices with no means of long-term support. These scenarios illustrate that security mechanisms that are adequate at deployment might not be adequate for the full lifespan of the device as security threats evolve. As such, this may create vulnerabilities that could persist for a long time. This is in contrast to the paradigm of traditional computer systems that are normally upgraded with operating system software updates throughout the life of the computer to address security threats. The long-term support and management of IoT devices is a significant security challenge.
- Many IoT devices operate in a manner where the user has little or no real visibility into the internal workings of the device or the precise data streams they produce. This creates security vulnerability when a user believes an IoT device is performing certain functions,

when in reality it might be performing unwanted functions or collecting more data than the user intends. The device's functions also could change without notice when the manufacturer provides an update, leaving the user vulnerable to whatever changes the manufacturer makes.

1.1.4.3 IOT SECURITY QUESTIONS

a) Good design practice

What are the sets of best practices for engineers and developers to use to design IoT devices to make them more secure? How do lessons learned from Internet of Things security problems get captured and conveyed to development communities to improve future generations of devices? What training and educational resources are available to teach engineers and developers more secure IoT design?

b) Cost vs. security trade- offs.

How do stakeholders make informed cost-benefit analysis decisions with respect to Internet of Things devices? How do we accurately quantify and assess the security risks? What will motivate device designers and manufacturers to accept additional product design cost to make devices more secure, and, in particular, to take responsibility for the impact of any negative externalities resulting from their security decisions? How will incompatibilities between functionality and usability be reconciled with security? How do we ensure IoT security solutions support opportunities for IoT innovation, social and economic growth?

c) Standards and Metrics.

How do stakeholders make informed cost-benefit analysis decisions with respect to Internet of Things devices? How do we accurately quantify and assess the security risks? What will motivate device designers and manufacturers to accept additional product design cost to make devices more secure, and, in particular, to take responsibility for the impact of any negative externalities resulting from their security decisions? How will incompatibilities between functionality and usability be reconciled with security? How do we ensure IoT security solutions support opportunities for IoT innovation, social and economic growth?

d) Data Confidentiality, Authentication and Access Control.

What is the optimal role of data encryption with respect to IoT devices? Is the use of strong encryption, authentication and access control technologies in IoT devices an adequate solution to prevent eavesdropping and hijacking attacks of the data streams these devices produce? Which encryption and authentication technologies could be adapted for the Internet of Things, and how could they be implemented within an IoT device's constraints on cost, size, and processing speed? What are the foreseeable management issues that must be

addressed as a result of IoT-scale cryptography? Are concerns about managing the crypto-key lifecycle and the expected period during which any given algorithm is expected to remain secure being addressed? Are the end-to-end processes adequately secure and simple enough for typical consumers to use?

e) Field-Upgradeability.

With an extended service life expected for many IoT devices, should devices be designed for maintainability and upgradeability in the field to adapt to evolving security threats? New software and parameter settings could be installed in a fielded IoT device by a centralized security management system if each device had an integrated device management agent. But management systems add cost and complexity; could other approaches to upgrading device software be more compatible with widespread use of IoT devices? Are there any classes of IoT devices that are low-risk and therefore don't warrant these kinds of features? In general, are the user interfaces IoT devices expose (usually intentionally minimal) being properly scrutinized with consideration for device management (by anyone, including the user)?

1.1.5 PRIVACY CONSIDERATIONS

1.1.5.1 Internet of Things Privacy Background

Respect for privacy rights and expectations is integral to ensuring trust in the Internet, and it also impacts the ability of individuals to speak, connect, and choose in meaningful ways. These rights and expectations are sometimes framed in terms of ethical data handling, which emphasizes the importance of respecting an individual's expectations of privacy and the fair use of their data. The Internet of Things can challenge these traditional expectations of privacy.

IoT often refers to a large network of sensor-enabled devices designed to collect data about their environment, which frequently includes data related to people. This data presumably provides a benefit to the device's owner, but frequently to the device's manufacturer or supplier as well. IoT data collection and use becomes a privacy consideration when the individuals who are observed by IoT devices have different privacy expectations regarding the scope and use of that data than those of the data collector.

Seemingly, benign combinations of IoT data streams also can jeopardize privacy. When individual data streams are combined or correlated, often a more invasive digital portrait is painted of the individual than can be realized from an individual IoT data stream. For example, a user's Internet-enabled toothbrush might capture and transmit innocuous data about a person's tooth-brushing habits. But if the user's refrigerator reports the inventory of the foods he eats and his fitness-tracking device reports his activity data, the combination of these data streams paint a much more detailed

and private description of the person's overall health. This data-aggregation effect can be particularly potent with respect to IoT devices because many produce additional metadata like time stamps and geolocation information, which adds even more specificity about the user.

In other situations, the user might not be aware that an IoT device is collecting data about the individual and potentially sharing it with third parties. This type of data collection is becoming more prevalent in consumer devices like smart televisions and video game devices. These kinds of products have voice recognition or vision features that continuously listen to conversations or watch for activity in a room and selectively transmit that data to a cloud service for processing, which sometimes includes a third party. A person might be in the presence of these kinds of devices without knowing their conversation or activities are being monitored and their data captured. These kinds of features may provide a benefit to an informed user, but can pose a privacy problem for those who are unaware of the presence of the devices and have no meaningful influence over how that collected information is used.

These kinds of privacy problems are critical to address because they have implications on our basic rights and our collective ability to trust the Internet. From a broad perspective, people recognize their privacy is intrinsically valuable, and they have expectations of what data can be collected about them and how other parties can use that data. This general notion about privacy holds true for data collected by Internet of Things devices, but those devices can undermine the user's ability to express and enforce privacy preferences. If users lose confidence in the Internet because their privacy preferences aren't being respected in the Internet of Things, then the greater value of the Internet may be diminished.

1.1.6 UNIQUE PRIVACY ASPECTS OF INTERNET OF THINGS

Generally, privacy concerns are amplified by the way in which the Internet of Things expands the feasibility and reach of surveillance and tracking. Characteristics of IoT devices and the ways they are used redefine the debate about privacy issues, because they dramatically change how personal data is collected, analysed, used, and protected. For example:

The traditional "notice and consent" online privacy model, in which users assert their privacy preferences by interacting directly with information presented on a computer or mobile screen (e.g. by clicking "I agree"), breaks down when systems provide no mechanism for user interaction. IoT devices frequently have no user interface to configure privacy preferences, and in many IoT configurations users have no knowledge or control over the way in which their personal data is being collected and used. This causes a gulf between the user's privacy preferences and the data-collecting behaviour of the IoT device. There might be less incentive for IoT vendors to offer a mechanism for users to express their privacy preferences if they regard the data collected as being non-personal data.

However, experience shows that data not traditionally considered personal data might actually be personal data or become personal data when combined with other data.

Assuming an effective mechanism can be developed to enable a user to express informed consent of their privacy preferences to IoT devices that mechanism needs to handle the large number of IoT devices a user must control. It is not realistic to think that a user will directly interact with each and every IoT device they encounter throughout the day to express their privacy preferences. Instead, privacy interface mechanisms need to be scalable to the size of the IoT problem, while still being comprehensive and practical from a user perspective.

- IoT devices often operate in contexts in which proximity exposes multiple people to the same data collection activity. For example, a geolocation tracking sensor in an automobile would record location data about all occupants of the vehicle, whether or not all the occupants want their location tracked. It may even track individuals in nearby vehicles. In these kinds of situations, it might be difficult or impossible to distinguish, much less honor, individual privacy preferences
- IoT devices often operate in contexts in which proximity exposes multiple people to the same data collection activity. For example, a geolocation tracking sensor in an automobile would record location data about a occupants of the vehicle, whether or not all the occupants want their location tracked. It may even track individuals in nearby vehicles. In these kinds of situations, it might be difficult or impossible to distinguish, much less honor, individual privacy preferences.

1.1.7 IOT PRIVACY QUESTIONS

These privacy issues would be challenging even if the interests and motivations of all of the participants in the IoT ecosystem were well aligned. However, we know that there can be unbalanced or unfair relationships and interests between those who are exposed to personal data collection and those who aggregate, analyse, and use the data. The data source might see an unwelcome intrusion into private space, often without consent, control, choice, or even awareness. The data collector, however, might consider this a beneficial resource that can add value to products and services as well as provide new revenue streams.

Because IoT challenges our notions of privacy in new ways, key questions need to be asked when re-evaluating online privacy models in the context of IoT. Some questions that have been raised include:

a) Fairness in Data Collection and Use.

How do we resolve the marketplace relationship between data sources and data collectors in the context of IoT? Personal data has personal and commercial value that sources and collectors value

differently, both individually and in aggregate; both parties have legitimate interests that may conflict. How those distinct interests might be expressed in a way that leads to fair and consistent rules for both sources and collectors concerning access, control, transparency, and protection

b) Wide-Ranging Privacy Expectations.

Privacy norms and expectations are closely related to the social and cultural context of the user, which will vary from one group or nation to another. Many IoT scenarios involve device deployments and data collection activities with multinational or global scope that cross social and cultural boundaries. What will that mean for the development of a broadly applicable privacy protection model for the Internet of Things? How can IoT devices and systems be adapted to recognize and honor the range of privacy expectations of the users and different laws?

c) Privacy by Design.

How can we encourage IoT device manufacturers to integrate privacy-by design principles into their core values? How do we foster the inclusion of consumer privacy considerations in every phase of product development and operation? How do we reconcile functionality and privacy requirements? In principle, manufacturers should expect that privacy respecting products and practices build long-term customer trust, satisfaction, and brand loyalty. Is that a sufficiently compelling motivation, when matched against the competing desires for design simplicity and speed to market? Should devices be designed with default settings configured for the most conservative data collection mode (i.e. opt out of data collection by default)?

d) Identification.

How should we protect data collected by IoT that appears not to be personal at the point of collection or has been "de-identified", but may at some point in the future become personal data (e.g. because data can be re-identified or combined with other data)?

The Internet of Things creates unique challenges to privacy that go beyond the data privacy issues that currently exist. Strategies need to be developed to respect individual privacy choices across a broad spectrum of expectations, while still fostering innovation in new IoT technology.

1.1.8 INTEROPERABILITY / STANDARDS ISSUES:

In the traditional Internet, interoperability is the most basic core value; the first requirement of Internet connectivity is that "connected" systems be able to "talk the same language" of protocols and encodings. Interoperability is so fundamental that the early workshops for Internet equipment vendors were called "Interops"; and it is the explicit goal of the entire Internet Standards apparatus centred on the Internet Engineering Task Force (IETF).65.

In a fully interoperable environment, any IoT device would be able to connect to any other device or system and exchange information as desired. In practicality, interoperability is more complex.

Interoperability among IoT devices and systems happens in varying degrees at different layers within the communications protocol stack between the devices. Furthermore, full interoperability across every aspect of a technical product is not always feasible, necessary, or desirable and, if artificially imposed (such as through government mandates), could provide disincentives for investment and innovation. The standardization and adoption of protocols that specify these communication details, including where it is optimal to have standards, are at the heart of the interoperability discussion for IoT.

CHAPTER 2

LITERATURE REVIEW

IoT is a revolutionary technology that represents the future of computing and communications. Most of the people over all worlds depend on agriculture for food consumption and due to this reason smart IT technologies are needed to mitigate with traditional agriculture methods. Using modern technologies can control the cost, maintenance and monitoring performance. Precision agriculture sensor monitoring network is used greatly to measure agriculture related information like temperature, humidity, soil PH, water level etc. So, with IoT technology, farmers can remotely monitor their crop and equipment by phones and computers. In this chapter, we surveyed some typical applications of Agriculture IoT Sensor Monitoring Network technologies using Cloud computing as the backbone. This survey is used to understand the different technologies and to build sustainable smart agriculture.

Simple IoT agriculture model is addressed with a wireless network.

2.1 LITERATURE REVIEW

Dwarkani et. al. [9] introduced an approach which puts forward a novel methodology for smart farming by connecting a smart sensing system and smart irrigator system through wireless communication technology. In [10], Nandurkar et. al identifies a low cost and efficient wireless sensor network (WSN) technology to observe the soil moisture and temperature from various location of farm and as per the need of crop controller to take the decision whether the irrigation is enabled or not. Gutiérrez et. al. [11] have proposed a method which describes how an automated irrigation system can be developed to optimize water use for agricultural crops along with a gateway unit that handles sensor information. Lee et. al [12] proposed an IoT based monitoring system for analysing crop environment and suggests a way to analyse harvest statistics and improve the efficiency of decision-making.

Although a lot of research has been done in agricultural automation but not all the state-of-art techniques are using a fully integrated system and doesn't provide any data for analysis of better crop production in future. Furthermore, a complete automation in agriculture is not achieved due to various environmental factors [7]. In order to provide a solution to all such problems, it is necessary to develop an integrated system which will take care of almost all the factors affecting the productivity at every stage. Hence the proposed model deals about developing a smart agriculture system using IoT that is given to the farmers as a product to be benefitted from the resources.

Smart agriculture is an automated and directed information technology implemented with the IOT (Internet of Things) [13].

Zhao Liqiang et. al [17] proposed an agricultural application of wireless sensor network for crop field monitoring. These systems fully equipped with two type sensor nodes to measure humidity, temperature, and an image sensing node to compare information by taking images of crops. Parameters play an important role for taking a good decision making for healthy crop within a time. The parameters are temperature, humidity, and images. Following these methods can achieve high stability of sensors with low consumption of power. Keerthi V et. al [18] proposed a greenhouse Monitoring System based on agriculture IoT with a cloud. In a greenhouse, the system can monitor different environmental parameters effectively using sensor devices such as light sensor, temperature sensor, relative humidity sensor and soil moisture sensor. Periodically (30 seconds) the sensors are collecting information of agriculture field area and are being logged and stored online using cloud computing and Internet of Things. In [19], Rajalakshmi P. et. al explain an IOT Based Crop-Field Monitoring and Irrigation Automation system to monitor crop-field. A system is developed by using sensors and then the decision is taken by the server based on the sensed data to automate irrigation system. By using wireless transmission, the sensed data is forwarded to the web server database. The user can monitor and control the system remotely with the help of application which provides a web interface to the user. In [20], Baltej Kaur et. al proposed a smart drip irrigation system. In this, an Android mobile application is used to reduce the human involvement and it used to control and monitor the crop area remotely. Water wastage can reduce with Drip Irrigation system and it works based on information gathered from water level sensors. Some more different sensors are used to monitor the environmental conditions. In [21] G. Parmeshwaran et. al Proposed smart irrigation systems using Internet of Things. To calculate humidity and water levels of soil some wireless sensors are required. These sensed data are sent to a smart gateway through a network, using a gateway called Generic IoT Border Router Wireless Br 1000. From the gateway, the data is then sending to a web service through a network.

This chapter envisioned us about the research work done in the agriculture to automate activities using IoT technology. The next chapter will discuss the existing systems and the proposed system to give us the idea about the working of the proposed system.

CHAPTER 3

EXISTING AND PROPOSED SYSTEM

In order to provide a solution to all such problems, it is necessary to develop an integrated system which will take care of almost all the factors affecting the productivity at every stage. Hence the proposed model deals about developing a smart agriculture system using IoT that is given to the farmers as a product to be benefitted from the resources. IOT is developing rapidly and widely applied in all wireless environments. In the proposed model, sensor technology and wireless networks integration of IOT technology has been implemented based on the actual situation of agricultural system. A combined approach with internet and wireless communications, Remote Monitoring System (RMS) is proposed. In this model, we have taken two types of agricultural fields viz. paddy field which requires water lodging and a general field where soil moisture is to be maintained. Various other factors such as humidity, rain and air quality levels are considered. Major objective is to collect real time data of agriculture production environment that provides easy access for agricultural facilities such as automatically check the moisture content of soil, temperature, humidity, rain, air quality or water level in the field. The decisions can be taken accordingly to perform a specific action such as turning on/off the water pump, draining out excess water, turning on sprinklers and exhaust fans. It also includes detecting intrusion in the field and provide live feeds on the mobile phone of the farmer using an IoT platform. The camera integrated with the system keeps a watch on the field in the real time using Raspberry Pi. The model also provides alerts through push notification and email to the farmer about all the activities going on in the farm such as intrusion alerts, turning on/off of pumps and exhausts. The system also provides real time sensor value which are displayed in the form of line charts and all the sensor data gets stored which can be provided to the data scientists and statisticians for predicting better crop production in future by artificializing the best crop growing condition in a poly-house.

3.1 EXISTING SYSTEM

The existing method and one of the oldest ways in agriculture is the manual method of checking the parameters. In this method the farmers themselves verify all the parameters and calculate the readings. Traditional farming tolerated the unpredictable environment better than modern farming that relies heavily on modern procedures and equipment. The traditional farming faces many constrains when they are not attended on time. In several cases, the farmer may not be present on the farm or the cultivating land due to some reasons and thus he may face losses if he is not reported about the current status of his land periodically. He would require more manpower and labour to keep a track of his farm every moment which includes if the water level or moisture level of the soil

is less or more than normal, and if any animal has barged into the farm which may lead a considerable damage to the crop. Moreover, he cannot stay on the farm every moment to monitor his crop thus facing many shortcomings by employing more labour and cost of production.

3.2 PROPOSED SYSTEM

The system is aimed to work on two different fields automated with the help of a central mobile application to control and monitor the entire system. The whole system comprises of different sensors and devices and they are connected to open source development board and raspberry Pi 3 which are further connected to one central server via Raspberry Pi WiFi and ESP8266 module. The central device sends and receives information from users end via an internet connection. The internet connectivity to the system can be provided by various means such as 3G, LTE, Wifi, WiMax etc. the system operates mainly using two modes, namely: manual mode and automatic mode. In the manual mode, the user can himself operate the system with the help of a mobile application, while in case of automatic mode the system takes its own decisions while controlling the various devices with the help of data collected from various sensors. The general architecture of the system is given in Fig.1.1 below:

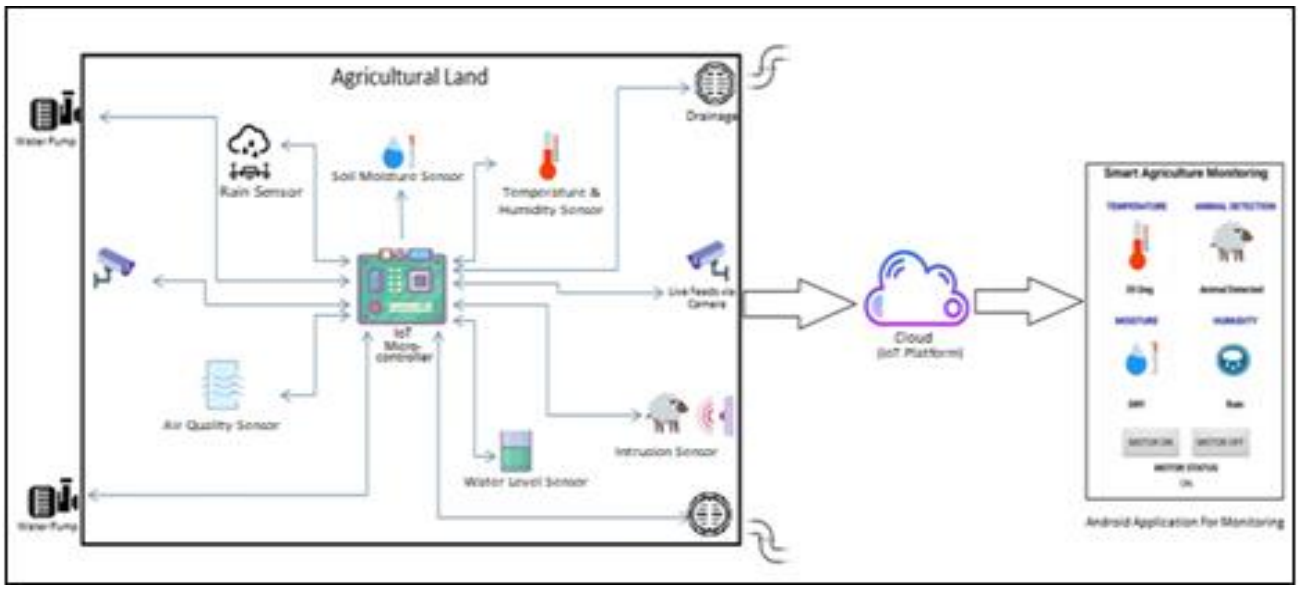


Fig. 3.1 Smart agricultural monitoring and controlling using IoT

This project presents model for smart agriculture to develop real time monitoring system for various operations of the field remotely from anywhere, anytime by mobile. In the field section, various sensors are deployed in the field like temperature sensor, moisture sensor, water level sensor, air quality sensor, camera for live feeds and PIR sensor for motion detection. The data collected from these sensors are connected to the microcontroller. In control section, the received data is verified with the threshold values. If the data exceeds the threshold value, appropriate action is taken to

resolve that issue. The farmer gets a message or push notification and automatically the desired task is performed after sensing. The values received by sensors is synced with an android application and the farmer gets the detailed description of the values in the form of charts on the android application. In manual mode, the user has to switch ON and OFF the devices using microcontroller by pressing the button in the Android Application. In automatic mode, the microcontroller gets switched ON and OFF automatically if the value exceeds the threshold point. Soon after the system boots, an alert is sent to the user. Other parameters like the temperature, humidity, moisture and the PIR sensors shows the threshold value and the water level sensor is used just to indicate the level of water inside the field for maintaining the level of water in fields like paddy. The IoT platform used for the functioning of this model is 'Blynk'. The functionality of the system is briefly described below:

3.2.1 AUTOMATIC WATER LEVEL CONTROLLING

In this module, we will maintain the level of the water in the paddy fields using an ultrasonic sensor (SR04). The sensor will check the level of water in the paddy filed by detecting the distance from the water using Doppler Effect. This method will primarily be used in paddy fields which require water to be lodged in the field. If the water level is below the mentioned threshold, the water pump will turn on until the required level has met. However, if the level is above the prescribed level, another pump will start draining out the water from the paddy field and send a notification to the user. The sensor is connected to an Arduino Uno board, which serves as a microcontroller to measure the distance. Based on the level of water the microcontroller turns on/off the relay module, which ultimately turns on/off the pump to supply water into the field. Further, in case the excess water gets drained out that will be conserved by sending the same water to the tanks.

3.2.2 MOISTURE LEVEL CONTROLLING

In this module we are trying to maintain the required moisture in the soil. If the moisture level of the soil is less than the desired level, the sprinkler attached to the water pump will automatically start for a specified time. A silver plated soil moisture sensor will undertake this process. We can program the sensitivity of this sensor as required. The sensor gives analogue input to the microcontroller based on the level of moisture present in the soil and the microcontroller is programmed to convert this analogue signal into digital signal to turn on/off the pump according to the level desired. In case we want to use fertilizers and pesticides in the field, the same can be sprinkled along with the water in the tank using manual mode.

3.2.3 TEMPERATURE LEVEL CONTROLLING

In this module, we are trying to maintain the water level of the field according to the temperature and moisture. We need sprinklers for water distribution. This process requires a digital temperature sensor (DHT11) which will monitor the temperature regularly. If the temperature of the area rises above a set value and if at the same time the soil moisture level is below a specified level, the sprinkler will again start for some specified time and will send an email and a notification on the user's phone.

3.2.4 RAIN AND AIR QUALITY SENSING

In this module, if the user is away from the farm, he will be able to see if it is raining on his farm and monitor the quality of the air in that area so that he can take according measures. If it starts raining or if the quality of air gets bad, he will be notified on his phone. This module requires a silver plated rain sensor and an air quality sensor (MQ135) to monitor the parameters and in case the air quality gets degraded, the air purifier will turn on for filtering the bad quality air from the poly-house. The data collected will be helpful in predicting the crop plantation strategies in future.

3.2.5 LIVE FEEDS AND MOTION SENSING

This module deals with sending live feeds on the mobile device of the user. In addition, if the motion sensor (HC-SR501) (out a suspected motion on the farm). The camera will be attached on top of the servo motor (MG995) for rotation of camera which will provide continuous live feeds.

CHAPTER 4
TECHNOLOGIES USED

4.1 INTERNET OF THINGS

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. An abundance of conferences, reports, and news articles discuss and debate the prospective impact of the "IoT revolution"—from new market opportunities and business models to concerns about security, privacy, and technical interoperability**.**

The large-scale implementation of IoT devices promises to transform many aspects of the way we live. For consumers, new IoT products like Internet-enabled appliances, home automation components, and energy management devices are moving us toward a vision of the "smart home'', offering more security and energy efficiency. Other personal IoT devices like wearable fitness and health monitoring devices and network enabled medical devices are transforming the way healthcare services are delivered. This technology promises to be beneficial for people with disabilities and the elderly, enabling improved levels of independence and quality of life at a reasonable cost. IoT systems like networked vehicles, intelligent traffic systems, and sensors embedded in roads and bridges move us closer to the idea of "smart cities'', which help minimize congestion and energy consumption. IoT technology offers the possibility to transform agriculture, industry, and energy production and distribution by increasing the availability of information along the value chain of production using networked sensors. However, IoT raises many issues and challenges that need to be considered and addressed in order for potential benefits to be realized.

4.2 RASPBERRY PI:

Pi is a single-board computer". Pi is a small scale computer in the size little bigger than a credit card, it packs enough power to run games, word processor like open office, image editor like Gimp and any program of similar magnitude. Pi was introduced as an educational gadget to be used for prototyping by hobbyists and for those who want to learn more about programming. It certainly cannot be a substitute for our day to day Linux, Mac or Windows PC. Pi is based on a Broadcom SoC (System of Chip) with an ARM processor [~700 MHz], a GPU and 256 to 512 MB RAM. The boot media is an SD card [which is not included], and the SD card can also be used for persist data. Now that you know that the RAM and processing power are not nearly close to the power house

machines you might have at home, these Pi's can be used as a Cheap computer for some basic functions, especially for experiments and education.

The basic things to get started with a pi:

Computer	A Raspberry Pi
Storage	SD Card and a SD card reader to image the OS [These days laptops have inbuilt card readers]
Power supply	5 volt micro USB adapter, mostly your android phone charger would work
Display	An TV/Monitor with DVI or HDMI port
Display connector	HDMI cable or HDMI to DVI converter cable
Input	USB Mouse
Input	USB Keyboard
Network	Ethernet cable

Table 4.1 Basic Things in Raspberry pi.

DESCRIPTION	MODEL A	MODEL B	MODEL B+
CHIP	Broadcom BCM2835 (CPU, GPU, DSP, SDRAM, and single USB port)		
PROCESSOR	700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set)		
RAM	256 MB	512 MB	512 MB
USB	1 (direct from BCM2835 chip)	2 on board	4 on board
STORAGE	SD Card	SD Card	MicroSD card
VOLTAGE	600mA up to 1.2A @ 5V	750mA up to 1.2A @ 5V	600mA up to 1.8A @ 5V
GPO	26	26	40

Table 4.2 Comparison between Models of Raspberry Pi

USB	Mainly used for peripherals like Keyboard, mouse and a Wi-Fi Adapter. A powered USB hub can be connected and be expanded
HDMI	This is the High Definition Multimedia Interface [HDMI] and is use to connect to a Display unit like TV or Monitor or sometimes a projector
Stereo Audio	Audio connections using a 3.5 mm jack
SD card	SD card is used as a boot device and also persistent storage. More storage can be attached to the USB
Micro USB	The micro USB port is used for supplying power to the unit
CSI connector	CSI [Camera serial Interface] is used for connecting a camera to the unit
Ethernet	Used for connecting to a network using a network cable
DSI connector	DSI [Digital serial Interface] is used for connecting a LCD to the unit

Table 4.3 Ports, Pins and their uses

4.2.1 AN INTRODUCTION TO GPIO AND PHYSICAL COMPUTING ON THE RASPBERRY PI:

One powerful feature of raspberry is the row of GPIO (General purpose input/output) pins along the edge of the board, next to the yellow video out socket

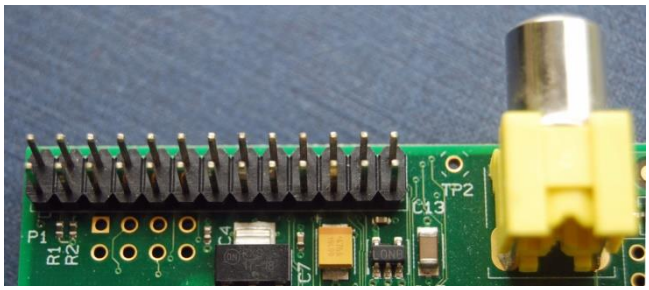


Fig. 4.1 General Purpose Input Output

These pins are a physical interface between the pi and the outside world. At the simplest level, you can think of them as a switches that you can turn on or off (input) or that pi can turn on or off(output). Seventeen of 26 pins are GPIO pins; the others are power or ground pins.

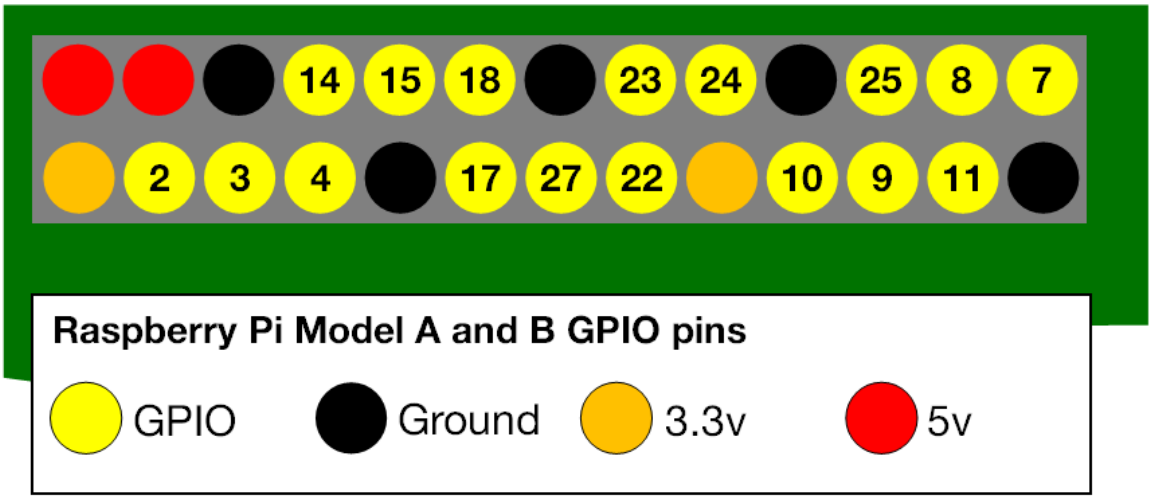


Fig. 4.2 Pin Diagram

You can program the pins to interact in an amazing ways with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example, the output can also do anything, from turning on a LED to sending a signal or data to another device. If the Raspberry pi is on a network, you can control devices that are attached to it from anywhere and those devices can send data back. Connectivity and control of physical devices over the internet is a powerful and exciting thing, and the Raspberry pi is deal for this.

4.2.2 WORKING OF GPIO PINS:

I. Input

GPIO outputs are easy; they are on or off, HIGH or LOW, 3v3 or 0v. Inputs are a bit trickier because of the way that digital devices work. Although it might seem reasonable just to connect a button across an input pin and a ground pin, the Pi can get as to whether the button is on or off. It might work properly, it might not. It's a bit like floating about in deep space; without a reference so it knows for certain when an input is received.

II. GPIO

General purpose input or output; in this specific case the pins on the Raspberry Pi and what you can do with them. So called because you can use them for all sorts of purposes; most can be used either inputs or outputs, depending on your program.

III. LED

Light-emitting-diode- a small, low power light source used widely in electronics ideal as introduction to physical computing on the Pi.

4.2.3 PHYSICAL COMPUTING:

Computing that involves tangible things connected to a computer, beyond standard input and output devices like keyboards and monitors. Think buttons, lights, robots, alarms, sensors, home automation, teddy bears called Babbage in near space and so on.

4.2.4 MODEL B IO PINS:

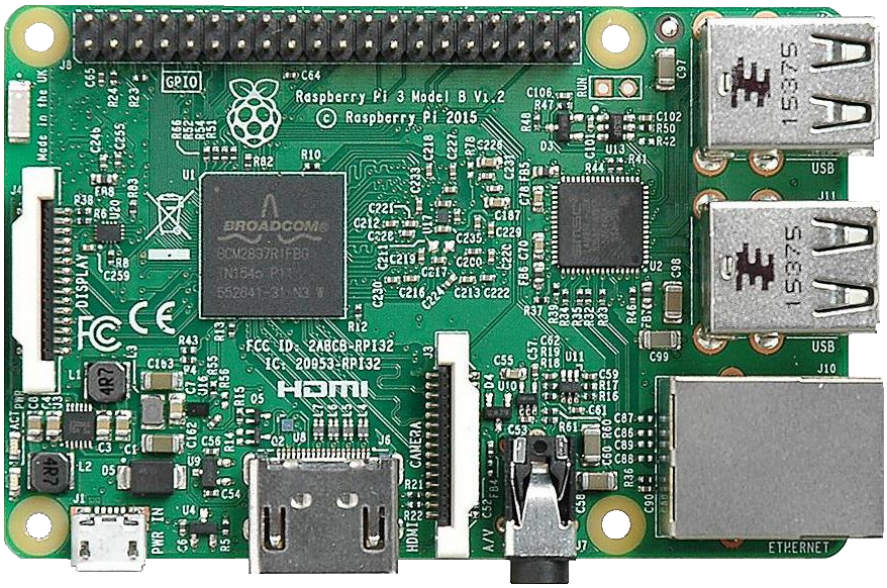


Fig. 4.3 Model B IO Pins

3.3 V Output

External circuitry may draw up to a total of 50mA max current from the 3.3V Out pins.

A. IO Pins

All IO pins are 3.3V, not 1.8V. Pins are not 5V tolerant.

I. P1 Header

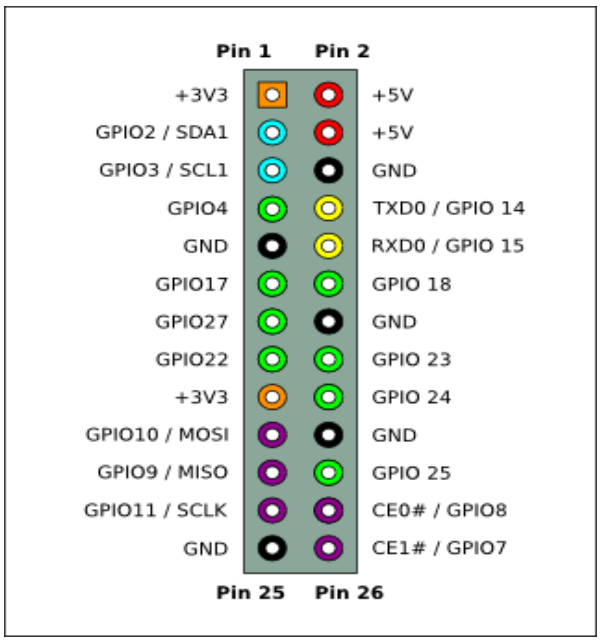


Fig. 4.4 P1 Header

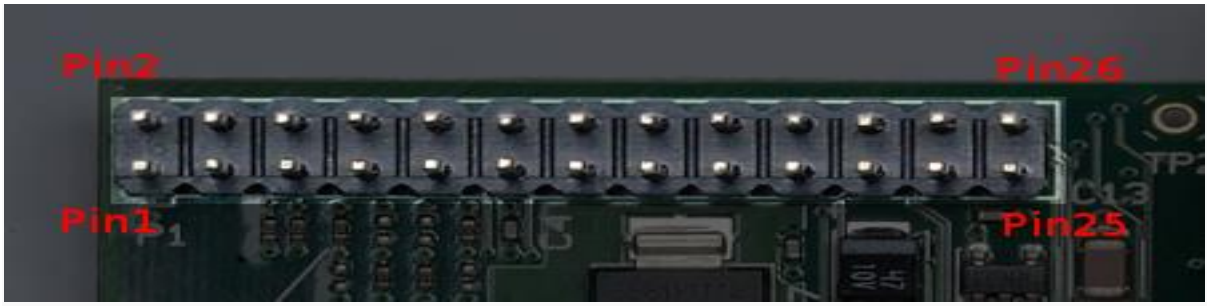


Fig 4.5 P2 Header

The Raspberry Pi Model A and B boards have a 26-pin 2.54 mm (100 mil)[1] expansion header, marked as P1, arranged in a 2x13 strip. They provide 8 GPIO pins plus access to I²C, SPI, UART), as well as +3.3 V, +5 V and GND supply lines. Pin one is the pin in the first column and on the bottom row. [2]

Revision 1 PCBs: Raspberry Pins with a revision 1 PCB (September 2012 or earlier) have a different pin assignment on the P1 connector:

- P1 pin 3 is GPIO 0 / SDA0 (not GPIO 2)
- P1 pin 5 is GPIO 1 / SCL0 (not GPIO 3)
- P1 pin 13 is GPIO 21 (not GPIO 27)

Revision 1 PCBs also do not have the P5 header (see below). See this discussion for more details of the changes between Rev 1 and Rev 2 PCBs.

II. P2 header

The P2 header is the Video Core JTAG and used only during the production of the board. It cannot be used as the ARM JTAG [3]. This connector is unpopulated in Rev 2.0 boards.

Fig. 4.6 Video Core Jtag

- Pin 1 - 3.3V (same as P1-01, 50 mA max current draw across both of them)
- Pin 7 - GND
- Pin 8 - GND

III. P3 header

The P3 header, unpopulated, is the LAN9512 JTAG.

Fig. 4.7 P3 HEADER

Useful P3 pins:

- Pin 7 - GND

IV. P5 header

The P5 header was added with the release of the Revision 2.0 PCB design.

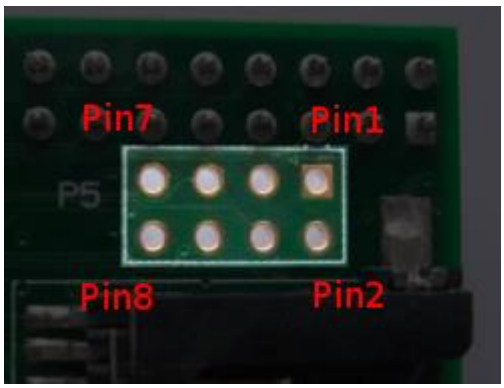


Fig. 4.8 P5 Header

As seen from the back of the board:

Pin Number	Pin Name Rev2
P5-01	5V0
P5-03	GPIO28
P5-05	GPIO30
P5-07	GND

Fig.4.9 P5 Header Pinout

As seen from the back of the board:

Note that the connector is intended to be mounted on the bottom of the PCB, so that for those who put the connector on the top side, the pin numbers are mirrored. Pin 1 and pin 2 are swapped, pin 3 and 4, etc.

An alternative way to attach this header is on top, at a slant away from the P1 header.

The new header can provide a second I²C channel (SDA + SCL) and handshake lines for the existing UART (TxD and RxD), or it can be used for an I2S (audio codec chip) interface using the PCM signals CLK, FS (Frame Sync), Din and D out.

Note that the connector is placed just off-grid with respect to the P1 connector.

V. P6 header

The P6 header was added with the release of the Revision 2.0 PCB design.

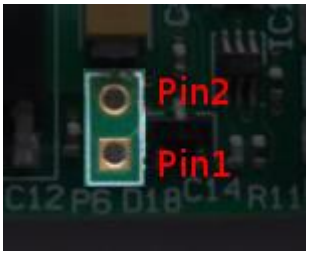


Fig. 4.10 P6 Header

i. Power-up State

It's likely all pins are set as inputs on power up (TBC).

I2C pins (e.g. P1-3 and P1-5) are therefore high due to the pull up resistors on these pins.

a) 12 C

1K8 pull up resistors are included on the RPi board so are not needed externally. When enabling the I2C port it seems both ports are enabled – is it possible to only enable 1 port and use the other I2C port as IO pins? Our assumption in Raspbian is no but we've not had reason to try and achieve this. Interfacing the RPi 3.3V I2C pins to a 5V device like an Arduino – see here.

b) SPI

The Chip Select signals are for up to two independent slave devices. It seems that with the SPI port enabled in Raspbian both the CS0 and CS1 pins are assigned to it and therefore can't be used as IO.

c) PWM Pin

The PWM pin available on the GPIO header is shared with the Audio system. This means that you can't use the PWM output and play audio through the 3.5mm jack at the same time.

4.3 ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

There are several advantages of Arduino, some of them are mentioned below:

- **Inexpensive -** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- **Cross-platform -** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment -** The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software -** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

4.3.1 ARDUINO NANO

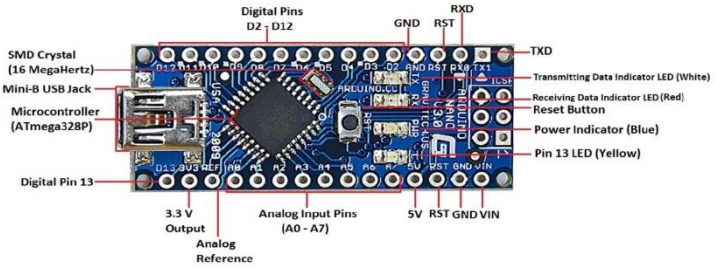


Fig 4.11: Arduino Nano Board

1. Specifications

- Microcontroller: Atmel ATmega168 or ATmega328
- Operating Voltage (logic level) 5 V
- Input Voltage (recommended) 7-12 V
- Input Voltage (limits) 6-20 V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 8
- DC Current per I/O Pin 40 mA
- Flash Memory 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by
- bootloader
- SRAM 1 KB (ATmega168) or 2 KB (ATmega328)
- EEPROM 512 bytes (ATmega168) or 1 KB (ATmega328)
- Clock Speed 16 MHz
- Dimensions 0.73" x 1.70"

2. Power:

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

3. Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

4. Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial**: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

- **I2C:** 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

4.3.2 Arduino Uno Board

Various components of Arduino are given below:

1. **Power (USB / Barrel Jack)**

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labelled **(1)** and the barrel jack is labelled **(2)**.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

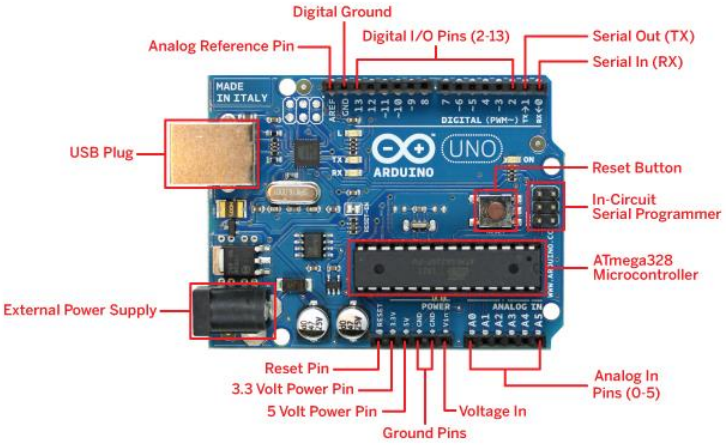


Fig 4.12: Arduino Uno Board

2. **Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)**

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labelled on the board and used for different functions.

- **GND (3)** Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.

- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

2. Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

3. Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit

4. TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program on the board).

5. Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company.

6. Voltage Regulator

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

4.4 SENSORS:

4.4.1 SOIL MOISTURE SENSOR

This sensor can be used to test the moisture of soil, when the soil is having water shortage, the module output is at high level, else the output is at low level. By using this sensor one can automatically water the flower plant, or any other plants requiring automatic watering technique. Module triple output mode, digital output is simple, analog output more accurate, serial output with exact readings.

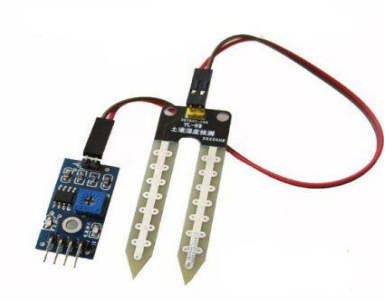


Fig 4.13 Soil and Moisture Sensor

1. Features:

- Sensitivity adjustable.
- Has fixed bolt hole, convenient installation.
- Threshold level can be configured.
- Module triple output mode, digital output is simple, analog output more accurate, serial output with exact readings.

2. Applications:

- Agriculture
- Landscape irrigation

3. Pin Details:

1: out (Active high output)

2: +5v (Power supply)

3: gnd (Power supply gnd)

4: rx (receiver)

5: tx (transmitter)

6: gnd (Power supply gnd)

4. Using The Sensor:

- Connect +5v to pin 2 and ground to pin 5 and 6.
- Pin 4 and 5 should be connected to particular transmitter and receiver pin of controller.
- Output pin may be connected to any port pins and can be used to any application.

5. Working:

Soil moisture sensors measure the water content in soil. A soil moisture probe is made up of multiple soil moisture sensors. One common type of soil moisture sensors in commercial use is a Frequency domain sensor such as a capacitance sensor. Another sensor, the neutron moisture gauge, utilize the moderator properties of water for neutrons. Soil moisture content may be determined via its effect on dielectric constant by measuring the capacitance between two electrodes implanted in the soil. Where soil moisture is predominantly in the form of free water (e.g., in sandy soils), the dielectric constant is directly proportional to the moisture content. The probe is normally given a frequency excitation to permit measurement of the dielectric constant. The readout from the probe is not linear with water content and is influenced by soil type and soil temperature. Therefore, careful calibration is required and long-term stability of the calibration is questionable.

- In This sensor We are using 2 Probes to be dipped into the Soil
- As per Moisture We will get Analoug Output variations from 0.60volts - 5volts
- Input Voltage 5V DC

4.4.2 TEMPERATURE AND HUMIDITY SENSOR

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller.

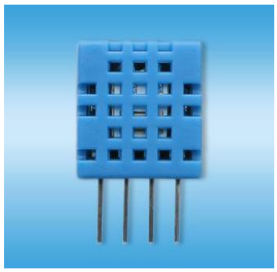


Fig. 4.14: DHT11 Digital Temperature and Humidity Sensor

1. Applications:

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

2. Features:

Low cost, long-term stability, relative humidity and temperature measurement, excellent quality, fast response, strong anti-interference ability, long distance signal transmission, digital signal output, and precise calibration.

3. Relative humidity:

- **Resolution:** 16Bit
- **Repeatability:** ±1% RH
- **Accuracy:** At 25℃ ±5% RH
- **Interchangeability:** fully interchangeable
- **Response time:** 1 / e (63%) of 25℃ 6s 1m / s air 6s
- **Hysteresis:** <± 0.3% RH
- **Long-term stability:** <± 0.5% RH / yr in

4. Temperature:

- **Resolution:** 16Bit
- **Repeatability:** ±0.2℃
- **Range:** At 25℃ ±2℃
- **Response time:** 1 / e (63%) 10S

5. Electrical Characteristics:

- **Power supply:** DC 3.5～5.5V
- **Supply Current:** measurement 0.3mA standby 60μA
- **Sampling period:** more than 2 seconds

6. Pin Description:

- the VDD power supply 3.5～5.5V DC
- DATA serial data, a single bus
- NC, empty pin
- GND ground, the negative power

7. Working:

They consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity they use the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate changes or the resistance between these electrodes changes. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller.On the other hand, for measuring temperature these sensors use a NTC temperature sensor or a thermistor. A thermistor is actually a variable resistor that changes its resistance with change of the temperature. These sensors are made by sintering of semiconductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The term "NTC" means "Negative Temperature Coefficient", which means that the resistance decreases with increase of the temperature.

4.4.3 RAIN SENSOR MODULE

The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The module features, a rain board and the control board that is separate for more convenience, power indicator LED and an adjustable sensitivity though a potentiometer.The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and when restored to the initial state, outputs high level.

Fig. 4.15 The Rain Sensor Module

1. Specifications:

- Adopts high quality of RF-04 double sided material.
- Area: 5cm x 4cm nickel plate on side,
- Anti-oxidation, anti-conductivity, with long use time;
- Comparator output signal clean waveform is good, driving ability, over 15mA;
- Potentiometer adjust the sensitivity;
- Working voltage 5V;
- Output format: Digital switching output (0 and 1) and analog voltage output AO;
- With bolt holes for easy installation;
- Small board PCB size: 3.2cm x 1.4cm;
- Uses a wide voltage LM393 comparator

2. Pin Configuration:

1. VCC: 5V DC

2. GND: ground

3. DO: high/low output

4. AO: analog output

3. Working:

Rain sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. When there is no rain drop on board. Resistance is high so we gets high voltage according to V=IR. When rain drop present it reduces the resistance because water is conductor of electricity and presence of water connects nickel lines in parallel so reduced resistance and reduced voltage drop across it.

4.4.4 PIR MOTION DETECTOR MODULE:

HC-SR501 is a pyroelectric sensor module which developed for human body detection. An integrated PIR sensor combined with a Fresnel lens which is mounted on a compact PCB, and limited components to form the module. Delay time, lux is adjustable. Customization is accepted.

Fig. 4.16: PIR Motion Detector Module.

1. Specification:

- **Compact size:** 24*32 mm
- **Supply voltage:** DC3.3-12V
- **Current drain :**≤30uA
- **Delay time:** 2s-80mins, adjustable
- **Blockade time:**2.3S
- **Trigger mode:** Repeatable triggered
- **Lux:** adjustable
- **Detecting distance:** ≤8m
- **Detecting angle:** ≤120•
- **Voltage Output:** 3.3V High/Low level signal or Open-Collector Output
- **Operation Temperature:** -20℃-+55℃
- **Infrared sensor:** dual element, low noise, high sensitivity

2. Functions

- **DC-INPUT：** supply voltage (DC3.3V-12V)
- **TEST:** test pin for output. With output, high level signal (3.3V); no output, low level signal (0V)
- **LOAD+:** anode of the load. LOAD-: cathode of the load. Voltage of the load and . DC-INPUT are the same. Max current with load is 100mA.
- **DARK_ADJ：** Lux adjustment.
- **DELAY_TIME:** delay time adjustment

3. Working:

The PIR sensors are more complicated thanthe other sensors as they consists of two slots. These slots are made of a special material which is sensitive to IR. The Fresnel lens is used to see that the two

slots of the PIR can see out past some distance. When the sensor is inactive, then the two slots sense the same amount of IR.The ambient amount radiates from the outdoors, walls or room,etc. When a human body or any animal passes by, then it intercepts the first slot of the PIR sensor. This causes a positive differential change between the two bisects.When a human body leaves the sensing area,the sensor generates a negative differential change between the two bisects. The infrared sensor itself is housed in a hermetically sealed metal to improve humidity/temperature/noise/immunity. There is a window which is made of typically coated silicon material to protect the sensing element.

4.4.5 MG995 HIGH SPEED METAL GEAR DUAL BALL BEARING SERVO:

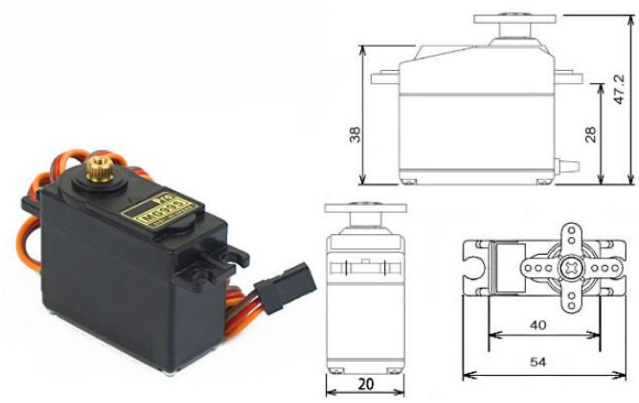


Fig.4.17 Servo Motor

The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec. This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up fast.

1. Specifications

- **Weight:** 55 g
- **Dimension:** 40.7 x 19.7 x 42.9 mm approx.
- **Stall torque:** 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- **Operating speed:** 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)

• **Operating voltage:** 4.8 V a 7.2 V

• **Dead band width:** 5 μs

• Stable and shock proof double ball bearing design

• **Temperature range:** 0 ºC – 55 ºC

4.4.6 HCSR04 ULTRASONIC SENSOR:

An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object. It is important to understand that some objects might not be detected by ultrasonic sensors. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object, but are deflected away from the Ultrasonic sensor. It is also possible for the object to be too small to reflect enough of the sound wave back to the sensor to be detected. Other objects can absorb the sound wave all together (cloth, carpeting, etc), which means that there is no way for the sensor to detect them accurately. These are important factors to consider when designing and programming a robot using an ultrasonic sensor.

Fig. 4.18 Ultrasonic Sensor

1. Working:

Ultrasonic sensors use sound to determine the distance between the sensor and the closest object in its path. How do ultrasonic sensors do this? Ultrasonic sensors are essentially sound sensors, but they operate at a frequency above human hearing.

The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off of an object and come back (Figure 1). The sensor keeps track of the time between sending the sound wave and the sound wave returning. If you know how fast something is going and how long it is traveling you can find the distance traveled with equation 1.

Equation 1. $d = v \times t$

The speed of sound can be calculated based on the a variety of atmospheric conditions, including temperature, humidity and pressure. Actually calculating the distance will be shown later on in this document. It should be noted that ultrasonic sensors have a cone of detection, the angle of this cone varies with distance, Figure 2 show this relation. The ability of a sensor to detect an object also depends on the objects orientation to the sensor. If an object doesn't present a flat surface to the sensor then it is possible the sound wave will bounce off the object in a way that it does not return to the sensor.

2. Specifications:

This section contains the specifications and why they are important to the sensor module. The Sensor modules requirements are as follows.

- Cost
- Weight
- Community of hobbyists and support
- Accuracy of object detection
- Probability of working in a smoky environment
- Ease of use
- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15º
- Ranging Distance: 2400cm
- Resolution: 0.3 cm
- Measuring Angle: 30º
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm
- Weight: approx. 10 g

3. Timing Chart and Pin Explanations:

The HCSR04 has four pins, VCC, GND, TRIG and ECHO; these pins all have different functions. The VCC and GND pins are the simplest they power the HCSR04. These pins need to be attached to a +5 volt source and ground respectively. There is a single control pin: the TRIG pin. The TRIG pin is responsible for sending the ultrasonic burst. This pin should be set to HIGH for 10 μs, at which point the HCSR04 will send out an eight cycle sonic burst at 40 kHZ. After a sonic burst has been sent the ECHO pin will go HIGH. The ECHO pin is the data pin it is used in taking distance

measurements. After an ultrasonic burst is sent the pin will go HIGH, it will stay high until an ultrasonic burst is detected back, at which point it will go LOW.

4. Wiring the HCSR04 to a Microcontroller:

This section only covers the hardware side. For information on how to integrate the software side, look at one of the links below or look into the specific microcontroller you are using.

The HCSR04 has 4 pins: VCC, GND, TRIG and ECHO.

- VCC is a 5v power supply. This should come from the microcontroller
- GND is a ground pin. Attach to ground on the microcontroller.
- TRIG should be attached to a GPIO pin that can be set to HIGH
- ECHO is a little more difficult. The HCSR04 outputs 5v, which could destroy many microcontroller GPIO pins (the maximum allowed voltage varies). In order to step down the voltage use a single resistor or a voltage divider circuit. Once again this depends on the specific microcontroller you are using, you will need to find out its GPIO maximum voltage and make sure you are below that.

4.4.7 MQ-135 AIR QUALITY SENSOR MODULE

The MQ series of gas sensors utilizes a small heater inside with an electro chemical sensor these sensors are sensitive to a range of gasses are used at room temperature. MQ135 sensor is a Sno2 with a lower conductivity of clean air. When the target explosive gas exists, then the sensor's conductivity increases more increasing more along with the gas concentration rising levels. By using simple electronic circuits, it convert the charge of conductivity to correspond output signal of gas concentration. The MQ135 gas sensor has high sensitivity in ammonia, sulfide, benze steam, smoke and in other harm full gas. It is low cost and suitable for different applications.The MQ135 gas sensor has high sensitivity in ammonia, sulfide, benze steam, smoke and in other harm full gas. It is low cost and suitable for different applications.

Fig. 4.19: Air Quality Sensor

1. Features:

- **Sensitive gas:** Ammonia, nitrogen oxide, alcohols, aromatic compounds, sulfide and smoke
- **Boost converter chip:** PT1301
- **Operating voltage:** 2.5V-5.0V
- **Dimensions:** 40.0mm*21.0mm
- **Fixing hole size:** 2.0mm

2. Operating principle:

MQ-135 gas sensor applies SnO_2 which has a lower conductivity in the clear air as a gas-sensing material. In an atmosphere where there may be polluting gas, the conductivity of the gas sensor raises along with the concentration of the polluting gas increases. MQ-135 performs a good detection to smoke and other harmful gas, especially sensitive to ammonia, sulfide and benzene steam. Its ability to detect various harmful gas and lower cost make MQ-135 an ideal choice of different applications of gas detection.

3. Interfaces:

Pin No.	Symbol	Descriptions
1	DOUT	Digital output
2	AOUT	Analog output
3	GND	Power ground
4	VCC	Positive power supply (2.5V-5.0V)

4. How to use

We will illustrate the usage of the module with an example of sensitive gas detection by connecting a development board.

- Download the relative codes to the development board.
- Connect the development board to a PC via a serial wire and the module to the development board. Then, power up the development board and start the serial debugging software. Here is the configuration of the connection between the module and the development board.
- Warm-up the sensor for a minute.
- The detected result can be checked by the LED indicator on the module. Put the sensor into a container filled with sensitive gas, you will find the indicator turns on. While take the sensor out of the container, you can see the indicator turns off.

4.4.8 CHANNEL 5V OPTICAL ISOLATED RELAY MODULE

This is a LOW Level 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.

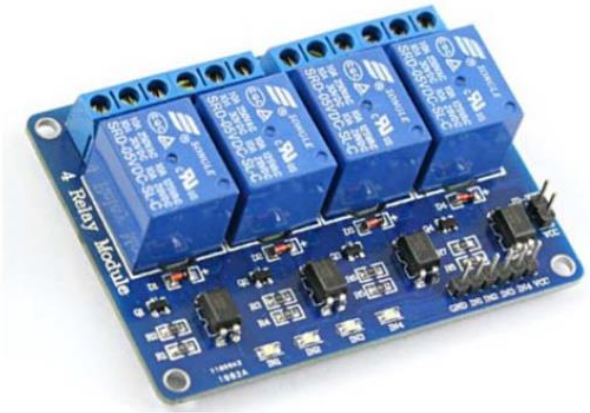


Fig. 4.20 4-Channel relay

Brief Data:

• Relay Maximum output: DC 30V/10A, AC 250V/10A.

• 4 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.

• Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).

• Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.

• Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

Operating Principle:

A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on. Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing

the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

Input:

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay module

IN2: Signal triggering terminal 2 of relay module

IN3: Signal triggering terminal 3 of relay module

IN4: Signal triggering terminal 4 of relay module

Output:

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 4 NC, 4 NO and 4 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

The chapter illustrated everything from the technologies that were used in this project, several sensors that are helping in collecting various data, their working and principle. The next chapter will deal with the code that is required to take use of these sensors and technologies for effective smart farming.

CHAPTER 5

CODE OF THE PROJECT

5.1 CODE FOR ARDUINO

```
#define BLYNK_PRINT Serial

#include <ESP8266_Lib.h>

#include <BlynkSimpleShieldEsp8266.h>

#include <DHT.h>


// You should get Auth Token in the Blynk App.

char auth[] = "32bda6a0ced449be9fc6eb26a8a5ff03";


// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Abc";

char pass[] = "arshid11";


// Hardware Serial on Mega, Leonardo, Micro...

//#define EspSerial Serial1


// or Software Serial on Uno, Nano...

#include <SoftwareSerial.h>

SoftwareSerial EspSerial(10, 11); // RX, TX


// Your ESP8266 baud rate:

#define ESP8266_BAUD 9600
```

```
ESP8266 wifi(&EspSerial);


#define DHTPIN 7          // What digital pin we're connected to

//int moisture =0;

#define echoPin 5

#define trigPin 6

int moisture =0;

float t = 0;

int distance =0;

 int gas = 0;

// Uncomment whatever type you're using!

#define DHTTYPE DHT11     // DHT 11

//#define DHTTYPE DHT22   // DHT 22, AM2302, AM2321

//#define DHTTYPE DHT21   // DHT 21, AM2301


DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;


// This function sends Arduino's up time every second to Virtual Pin (5).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void sendSensor()

{

  int duration;
```

```
  digitalWrite(trigPin, HIGH);
  delay(100);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = 15 - ((duration / 2) / 29.1);//This is used to calculate the water level by dividing the distance by 2 because it calculates double distance.
  //We are using 29.1 as the
  Serial.print(distance);
  Serial.print("CM");
  Serial.println("");
  //delay(100);
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
  int moisture = analogRead(A0);
  int rain = analogRead(A1);
  int gas = analogRead(A2);
  rain = map(rain, 1023, 0, 0, 100);
  gas = map(gas, 1023, 0, 100, 0);
  moisture = map(moisture, 1023, 0, 0, 100);
if (distance>8)
{
  digitalWrite(2,HIGH);
  digitalWrite(4,LOW);
  Blynk.virtualWrite(V7, HIGH);
  Blynk.virtualWrite(V8, LOW);
```

```
  Blynk.notify("Water Pump Turned On and Drain Off");
  Blynk.email("smartagriculture2018@gmail.com","Pump Alert!","Water Pump Turned On and Drain Off");
  }
else if (distance < 7)
{

  digitalWrite(2,LOW);
  digitalWrite(4,HIGH);
  Blynk.virtualWrite(V7, LOW);
  Blynk.virtualWrite(V8, HIGH);
  Blynk.notify("Drain Pump Turned On and Water Pump Off");
  Blynk.email("smartagriculture2018@gmail.com","Drain Alert!","Drain Pump Turned On and Water Pump Off");
  }

else{


  digitalWrite(2,LOW);
  digitalWrite(4, LOW);
  Blynk.virtualWrite(V7, LOW);
  Blynk.virtualWrite(V8, LOW);
  Blynk.notify("Drain Pump and Water Pump Off");
    }
if (moisture<50 || (moisture<50&&t>22) )
```

```
{
  digitalWrite(12,HIGH);
  Blynk.notify("Sprinklers Turned On Due to Low Soil Moisture.");
  Blynk.email("smartagriculture2018@gmail.com","Sprinkler On Alert!","Sprinklers Turned On Due to Low Soil Moisture.");
  }

else{

  digitalWrite(12,LOW);
  Blynk.notify("Sprinklers Turned Off Due to Balanced Soil Moisture.");
  Blynk.email("smartagriculture2018@gmail.com","Sprinkler On Alert!","Sprinklers Turned Off Due to Balanced Soil Moisture.");
}

if (gas>50)
{
  digitalWrite(8,HIGH);
  Blynk.virtualWrite(V9, HIGH);
  Blynk.notify("Air Purifier Turned On Due to Bad Air Quality.");
  Blynk.email("smartagriculture2018@gmail.com","Sprinkler On Alert!","Air Purifier Turned On Due to Bad Air Quality.");
  }
else {
  digitalWrite(8,LOW);
  Blynk.virtualWrite(V9, LOW);
```

```
  Blynk.notify("Air Purifier Turned Off Due to Balanced Air Quality.");
  Blynk.email("smartagriculture2018@gmail.com","Sprinkler On Alert!","Air Purifier Turned Off Due to Balanced Air Quality.");
  }
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // We shouldn't send more that 10 values per second.
  Blynk.virtualWrite(V5, h);
  Blynk.virtualWrite(V6, t);
  Blynk.virtualWrite(V1, rain);
  Blynk.virtualWrite(V2, moisture);
  Blynk.virtualWrite(V3, gas);
  Blynk.virtualWrite(V4, distance*100);
}



void setup()
{
  // Debug console
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
```

```
  // Set ESP8266 baud rate
  EspSerial.begin(ESP8266_BAUD);
  delay(10);

  Blynk.begin(auth, wifi, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, wifi, ssid, pass, "blynk-cloud.com", 8442);
  //Blynk.begin(auth, wifi, ssid, pass, IPAddress(192,168,1,100), 8442);
  dht.begin();

  // Setup a function to be called every second

  timer.setInterval(1000L, sendSensor);
  pinMode(2,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(12,OUTPUT);
  digitalWrite(2,LOW);
  digitalWrite(4,LOW);
  digitalWrite(8,LOW);
  digitalWrite(12,LOW);
}

void loop()
{
```

```
  Blynk.run();
  timer.run();
}
```

5.2 CODE FOR RASPBERRY

5.2.1 CODE FOR INITIALIZING GPIO PINS

```
// The code is used to initialize the GPIO pins and providing output to the relays
from time import time          ## import the time library
import RPi.GPIO as GPIO        ## Import GPIO library
GPIO.setmode(GPIO.BCM)         ## Use board physixal pin numbering
GPIO.setwarnings(False)
button1 = 6
button2 = 7
button3 = 8
button4 = 9
led1 = 2
led2 = 3
led3 = 4
led4 = 5
GPIO.setup(led1, GPIO.OUT)     ## Setup GPIO Pin 2 to OUT
GPIO.setup(led2, GPIO.OUT)     ## Setup GPIO Pin 3 to OUT
GPIO.setup(led3, GPIO.OUT)     ## Setup GPIO Pin 4 to OUT
GPIO.setup(led4, GPIO.OUT)     ## Setup GPIO Pin 5 to OUT

GPIO.setup(button1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
```

```
GPIO.setup(button2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(button3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(button4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.output(led1,False)
GPIO.output(led2,False)
GPIO.output(led3,False)
GPIO.output(led4,False)

while True:
    if GPIO.input(button1)== True:
        GPIO.output(led1,True)  ## Turn on Led

    else:
        GPIO.output(led1,False) ## Turn off Led

    if GPIO.input(button2)== True:
        GPIO.output(led2,True)  ## Turn on Led

    else:
        GPIO.output(led2,False) ## Turn off Led

    if GPIO.input(button3)== True:
        GPIO.output(led3,True)  ## Turn on Led

    else:
```

```
        GPIO.output(led3,False) ## Turn off Led


    if GPIO.input(button4)== True:
        GPIO.output(led4,True)  ## Turn on Led


    else:
        GPIO.output(led4,False) ## Turn off Led



GPIO.Cleanup()
```

5.2.2 Code for email notification

```
//The code is used to send an email notification to the farmer in case intrusion is detected in the farm.
from time import time
import RPi.GPIO as GPIO         ## Import GPIO library
import smtplib
GPIO.setmode(GPIO.BCM)          ## Use board physixal pin numbering
GPIO.setwarnings(False)
intrusion = 25
GPIO.setup(intrusion, GPIO.IN, pull_up_down=GPIO.PUD_UP)
smtpUser = 'smartagriculture2018a@gmail.com'
smtpPass = 'qwert12345$#'
toAdd = 'smartagriculture2018@gmail.com'
fromAdd = smtpUser
subject = 'WELCOME TO THE SMART AND ADVANCED AGRICULTURE MONITORING USING IOT AND RASPBERRY PI'
```

```
header = 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'subject: ' + subject

body =  'THIS SMART AGRICULTURE SYSTEM HAS BEEN DEVELOPED BY THE STUDENTS OF BGSBU RAJOURI, DEPTT. OF COMPUTER SCIENCE AND ENGINEERING"\n" GROUP MEMBERS"\n" 1. MOAZAM FARHAN BANDAY(03-CSE-14) "\n" 2. AQLEEM MAKHDOOMI(31-CSE-14)"\n" 3. ZAHID ASLAM SHORA(34-CSE-14)"\n" 4. UMAR FAROOQ(45-CSE-14)'

print header + '\n' + body

s = smtplib.SMTP('smtp.gmail.com',587)

s.ehlo()

s.starttls()

s.ehlo()

s.login(smtpUser,smtpPass)

s.sendmail(fromAdd, toAdd, header + '\n\n' + body)

s.quit()

count1 = 1

count2 = 1

while True:

    if GPIO.input(intrusion) == True:

        count2 =1

        count1+=1

        if count1 == 2:

            #print('hello how r u')

            smtpUser = 'smartagriculture2018a@gmail.com'

            smtpPass = 'qwert12345$#'

            toAdd = 'smartagriculture2018@gmail.com'

            fromAdd = smtpUser
```

```
                subject = 'INTRUSION DETECTED IN THE FIELD'
                header = 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'subject: ' + subject
                body =  'INTRUSION DETECTED, CHECK OUT THE LATEST FEEDS AT http://192.168.43.107:8081'
                print header + '\n' + body
                s = smtplib.SMTP('smtp.gmail.com',587)
                s.ehlo()
                s.starttls()
                s.ehlo()
                s.login(smtpUser,smtpPass)
                s.sendmail(fromAdd, toAdd, header + '\n\n' + body)
                s.quit()

        else:
                count1 =1
                count2+=1
                if count2 == 2:

                        smtpUser = 'smartagriculture2018a@gmail.com'
                        smtpPass = 'qwert12345$#'
                        toAdd = 'smartagriculture2018@gmail.com'
                        fromAdd = smtpUser
                        subject = 'NO INTRUSION DETECTED IN THE FIELD'
                        header = 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'subject: ' + subject
                        body =  'NO INTRUSION DETECTED, CHECK OUT THE LATEST FEEDS AT http://192.168.43.107:8081'
```

```
            print header + '\n' + body
            s = smtplib.SMTP('smtp.gmail.com',587)
            s.ehlo()
            s.starttls()
            s.ehlo()
            s.login(smtpUser,smtpPass)
            s.sendmail(fromAdd, toAdd, header + '\n\n' + body)
            s.quit()
    GPIO.Cleanup()
```

The chapter gave us the code for the development boards, which have been used in this project to take values from the sensors, displaying them on the Blynk Platform, initiating respective functions on some certain thresholds like turning the water pumps, drainage, sprinklers or air purifiers ON and OFF, and sending the notifications and emails to the user etc.

The next chapter will give us an overview of working of Blynk Platform for this project in the form of screenshots.

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 SCREENSHOTS OF THE BLYNK PLATFORM WORKING

The screenshots of the 'Blynk' Platform working on this project are given below:

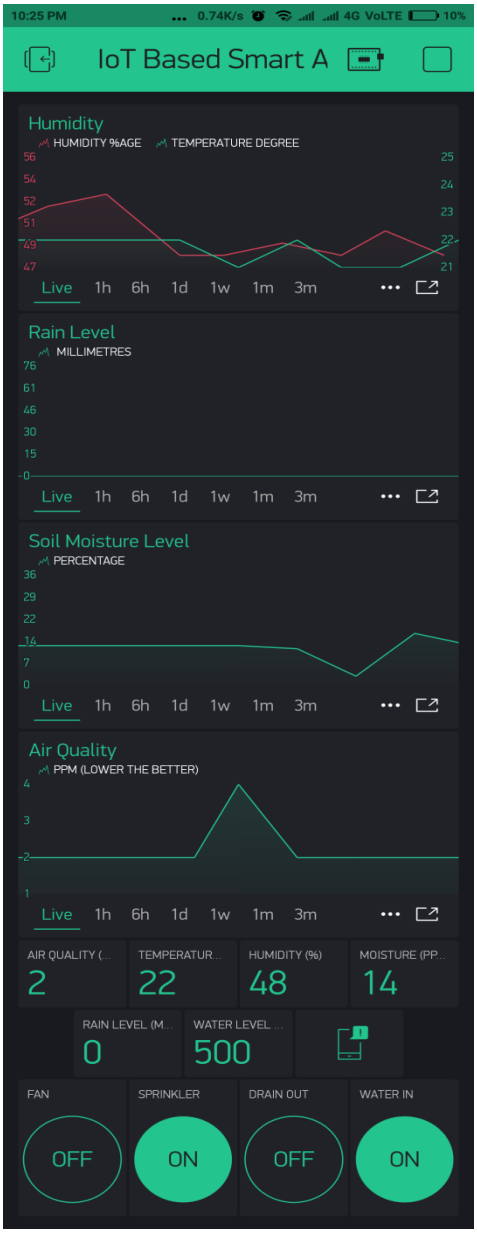


Fig 6.1

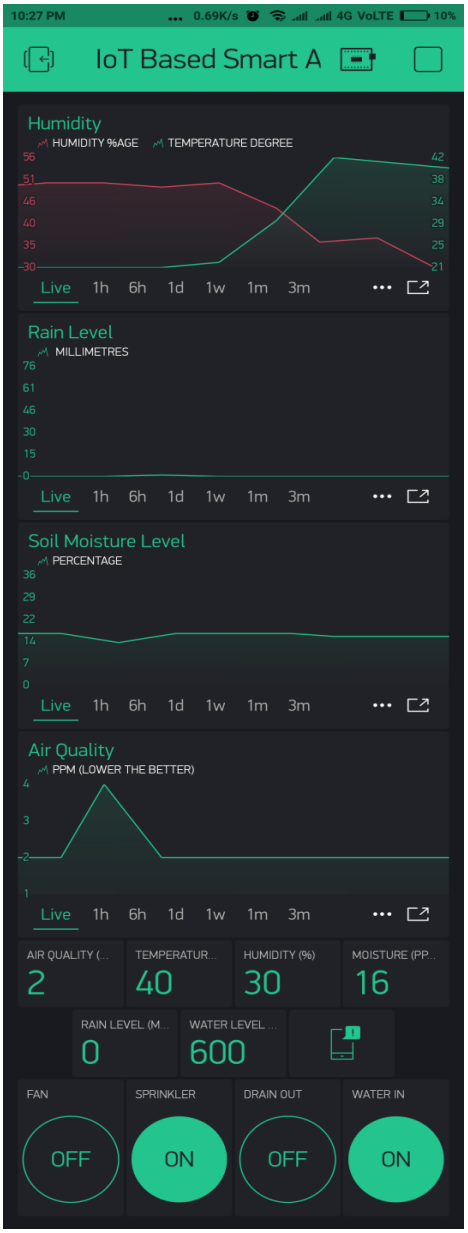


Fig 6.2

Dashboard of Blynk android app showing sprinkler, water in/out on at different levels of moisture and water level

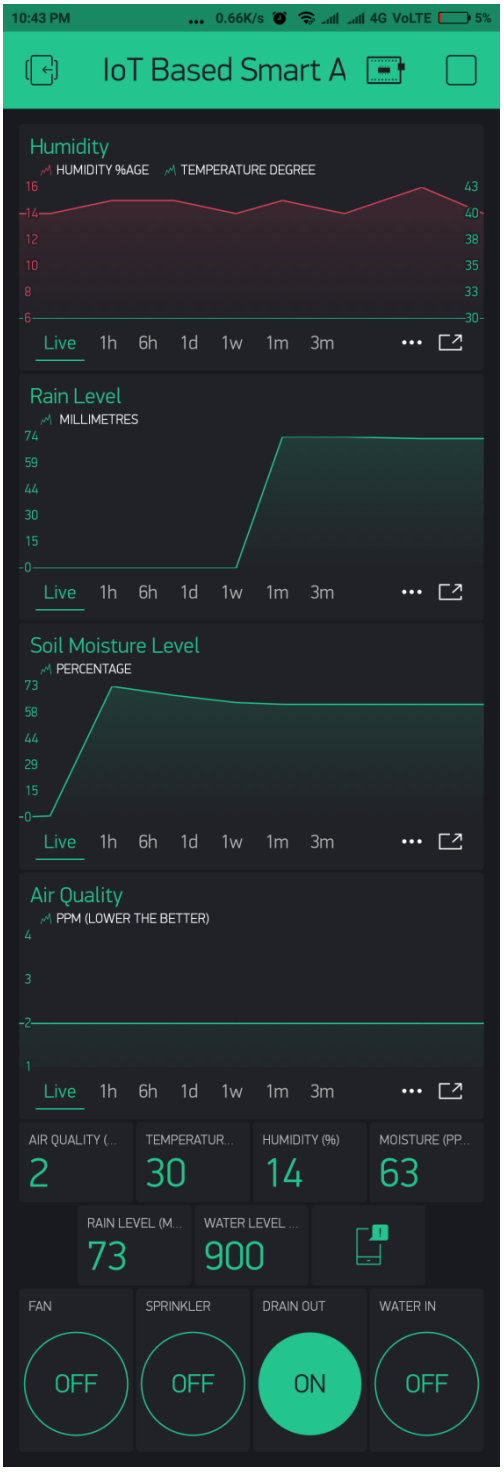


Fig 6.3

Fig 6.4

Dashboard of Blynk android app showing sprinkler off and water in/out on at different levels of moisture and water level

Fig 6.5

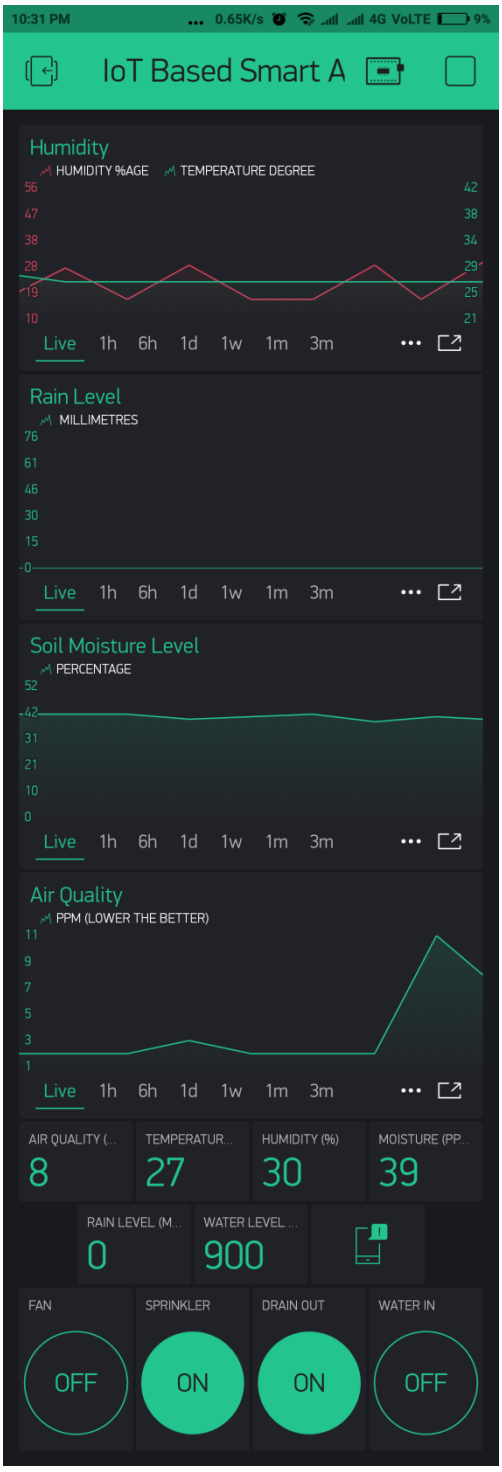


Fig 6.6

Dashboard of Blynk android app showing sprinkler on/off and water out on at different levels of moisture,humidity and water level

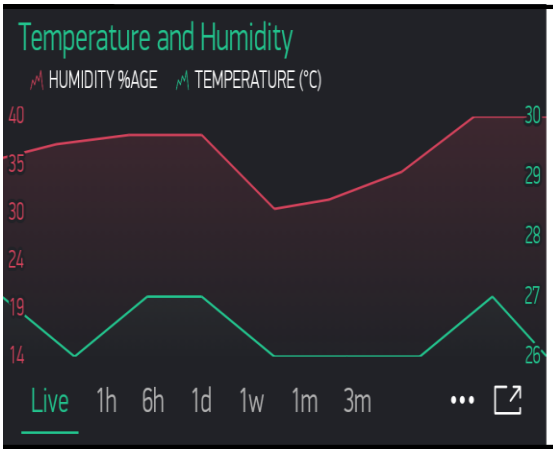
Fig 6.7 **Fig 6.8**

Dashboard of Blynk android app showing sprinkler off water in off and drain out on at different levels of moisture,humidity and water level

Fig 6.9 **Fig 6.10**

Dashboard of Blynk android app showing sprinkler on water in off and drain out on at different levels of moisture,humidity and water level

6.2 SENSORS VALUE CHARTS AND THEIR TABULAR VALUES



(a)

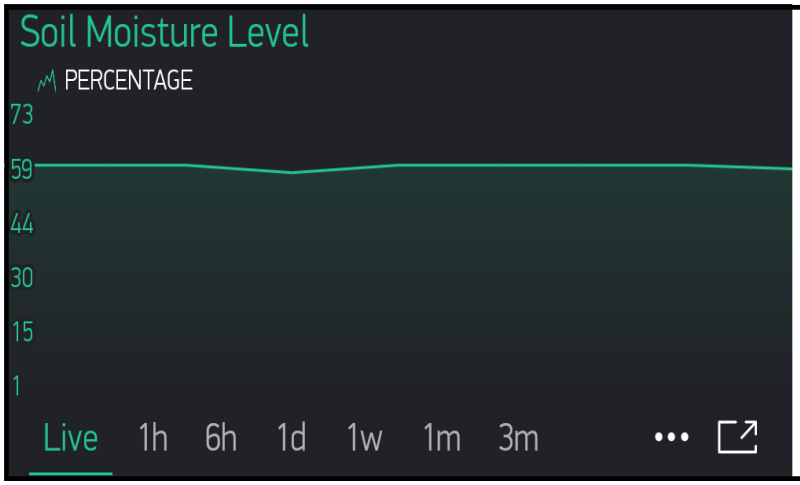
Time (X-axis)	Temperature (Y-axis)
4/14/2018 18:05:51	26.26666667
4/14/2018 18:05:52	25.78333333
4/14/2018 18:05:52	26.01666667
4/14/2018 18:05:53	26.47222222
4/14/2018 18:05:54	26.22857143
4/14/2018 18:05:54	27.18181818
4/14/2018 18:05:54	27.04545455
4/14/2018 18:05:55	27.07142857
4/14/2018 18:05:55	26.92857143
4/14/2018 18:05:56	27.21428571

(b)

Time (X-axis)	Humidity(Y-axis)
4/14/2018 18:05:51	35.13333333
4/14/2018 18:05:51	36.0625
4/14/2018 18:05:52	33.05555556
4/14/2018 18:05:53	30.21052632
4/14/2018 18:05:53	33.3125
4/14/2018 18:05:54	34.16666667
4/14/2018 18:05:55	37.31578947
4/14/2018 18:05:55	38.33333333
4/14/2018 18:05:56	38.05882353
4/14/2018 18:05:57	39.16666667

(c)

Fig. 6.11. Temperature & Humidity Sensor (a) DHT sensor line chart (b) Temperature Data (c) Humidity Data



(a)

Time (X-axis)	Soil Moisture(Y-axis)
4/14/2018 18:05:51	59.26666667
4/14/2018 18:05:51	60.00066667
4/14/2018 18:05:52	59.08125
4/14/2018 18:05:53	58.73333333
4/14/2018 18:05:53	58.7125
4/14/2018 18:05:54	57.5625
4/14/2018 18:05:55	57.66666667
4/14/2018 18:05:55	58.6375
4/14/2018 18:05:56	58.84615385
4/14/2018 18:05:57	58.84615385

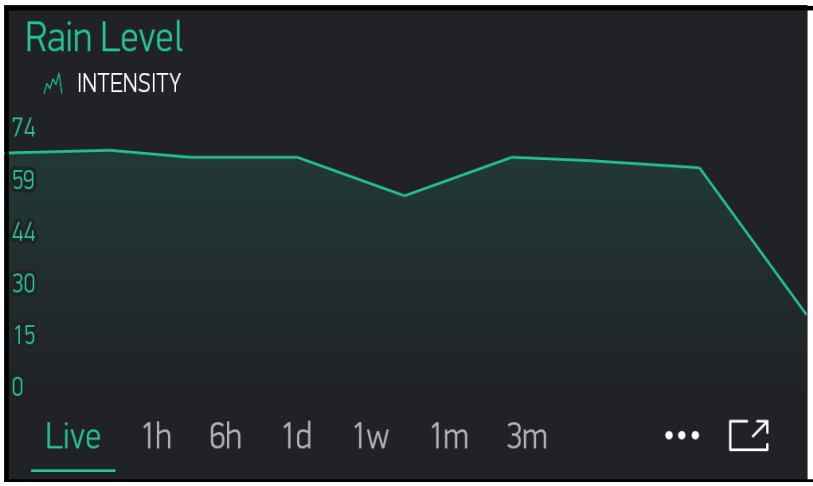
(b)

Fig. 6.12 Soil Moisture Sensor (a) Soil Moisture Sensor line chart (b) Soil Moisture Sensor Data

Time (X-axis)	Air Quality(Y-axis)
4/14/2018 18:05:51	1.957142857
4/14/2018 18:05:52	1.928571429
4/14/2018 18:05:53	1.976923077
4/14/2018 18:05:53	0.966666667
4/14/2018 18:05:54	0.833333333
4/14/2018 18:05:54	0.944433333
4/14/2018 18:05:55	1.933333333
4/14/2018 18:05:56	1.909090909
4/14/2018 18:05:56	1.866666667
4/14/2018 18:05:57	1.866666667

(a) (b)

Fig. 6.13. Air Quality Sensor (a) Air Quality Sensor (MQ135) line chart (b) MQ135 Sensor Data



Time(X-Axis)	Rain Intensity(Y-axis)
4/14/2018 18:05:51	65.86666667
4/14/2018 18:05:52	66.33333333
4/14/2018 18:05:53	64.5625
4/14/2018 18:05:53	62.86666667
4/14/2018 18:05:54	63.25
4/14/2018 18:05:54	57.75
4/14/2018 18:05:54	46.4
4/14/2018 18:05:55	32.125
4/14/2018 18:05:55	31.84615385
4/14/2018 18:05:56	28.8768899

(a) (b)

Fig. 6.14. Rain Sensor Graph and Data (a) Rain Sensor line chart (b) Rain Sensor Data

CONCLUSION AND FUTURE WORK

After several observations and experimental tests, we come to a conclusion that this model solves the basic problems of the farmer by automating many agricultural and irrigational activities and has a very wide scalability and is extensible after adding the features in the future work by automating most of the farming activities.By using these features, smart farming will help to grow the market for the farmer at the tip of his fingers.

For future developments, the system can be enhanced to work for large area of the farm. Moreover, the raw data that we are collecting in the form of heterogeneous data values can be analysed and processed using big data technologies such as Hadoop, Pig, Hive, MongoDB(No SQL) etc. to understand and predict better crop production in future by artificializing the same condition in a poly-house or by hydroponics.

We can also maintain the pH values of the soil which are distorted because of using several fertilizers and pesticides. The pH data provide by the sensors will help us to identify whether and acidic or alkaline solution is to be added to the fields based on certain calculated values.

With lot of advancements in the digital image processing, we can also be able to identify the disease in the crop and check whether the crop is ripe or not.

Refrences

1. Park,H.,Kim,H.,Hotaek,J.,Song,J.,S.,:"Recent advancements in the Internet-of-Things related standards: A oneM2M perspective", ICT Express 2 (2016).
2. GubbiJayavardhana, et al., Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. (2013)
3. Miorandi,D.,Sicari,S.,Pellegrini,F.,D.,Chlamtac,I.:"Internet of things: Vision, applications and research challenges", Ad Hoc Networks,Elsevier (2012)
4. Jinxing Zhang, ShimonGU, and Chao Zheng (2010), A Summary of Research Progress on Cloud Computing, Application Research of Computers,Vol. 27, No. 2,429-433.
5. Quan Chen, and Qianni Deng (2009), Cloud Computing and Its Key Technologies, Journal of Computer Applications, Vol. 29, No. 9, 256.
6. M. S. Mekala and P. Viswanathan, "A Survey: Smart agriculture IoT with cloud computing," 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, 2017
7. Dr.N.Suma, Sandra Rhea Samson, S.Saranya, G.Shanmugapriya, R.Subhashri, “IOT Based Smart Agriculture Monitoring System”, February 17 Volume 5 Issue 2 , International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 177 – 181
8. M.K.Gayatri, J.Jayasakthi, Dr.G.S.Anandhamala,“Providing Smart Agriculture Solutions to Farmers for Better Yielding Using IoT”, IEEE International Conference on Technological Innovations in ICT forAgriculture and Rural Development (TIAR 2015).
9. ChetanDwarkani M, Ganesh Ram R, Jagannathan S, R.Priyatharshini, “Smart Farming System Using Sensors for Agricultural Task Automation”, IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development (TIAR 2015).
10. S. R. Nandurkar, V. R. Thool, R. C. Thool, “Design and Development of Precision Agriculture System Using Wireless Sensor Network”, IEEE International Conference on Automation, Control, Energy and Systems (ACES), 2014.
11. Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel Porta-Gándara, “Automated Irrigation System Using a Wireless Sensor Network and GPRS Module”, IEEE Transactions on Instrumentation and Measurements, 0018-9456,2013
12. Meonghun Lee, Jeonghwan Hwang, Hyun Yoe, “Agricultural Protection System Based on IoT”, IEEE 16th International Conference on Computational Science and Engineering, 2013.

13. Mohanraj,I.,Kirthika,A.,Naren, J.:"Field Monitoring and Automation using IOT in Agriculture Domain",Proc. 6th International Conference on Advances in Computing and Communications, ICACC 2016, September 2016, Cochin, India
14. http://arduinoarts.com/2011/08/the-arduino-uno-anatomy/
15. https://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html
16. Zhao Liqiang, Yin Shouyi, Liu Leibo, Zhang Zhen, Wei Shaojun,
17. A crop Monitoring System Based on Wireless Sensor Network ELSEVIER,Procedia Environmental Sciences-2011.
18. Keerthi.v , Dr.G.N.Kodandaramaiah, cloud IoT Based greenhouse Monitoring System Int. Journal of Engineering Research and Applications ,ISSN: 2248-9622, Vol. 5, Issue 10, (Part - 3) October 2015, pp.35-41.
19. Monitoring and Irrigation Automation system. ieeexplore.ieee.org/iel7/7589934 /7726872/07726900.
20. Baltej Kaur , Danish Inamdar, Vishal Raut,Akash Patil,Nayan Patil, A Survey On Smart Drip Irrigation System International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 02 | Feb-2016.
21. G. Parameswaran, K.Sivaprasath, Arduino Based Smart Drip Irrigation System Using Internet of Things- DOI 10.4010/2016.1348 ,ISSN 2321 3361 © 2016 IJESC.