UNIVERSITY of GREENWICH

Alliance with FPT Education

BTEC

ASSIGNMENT 1 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 43: Internet of Things**		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Trần Lê Trung Quân	**Student ID**	GCD19894
Class	GCD0803	**Assessor name**	Trần Trọng Minh

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	

Grading grid

P1	P2	P3	P4	M1	M2	M3	M4	D1	D2

❒ Summative Feedback: ❒ Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Internal Verifier's Comments:
Signature & Date:

REPORT LIST

REPORT FIGURES

P1. EXPLORE VARIOUS FORMS OF IOT FUNCTIONALITY

I. Definition about IoT

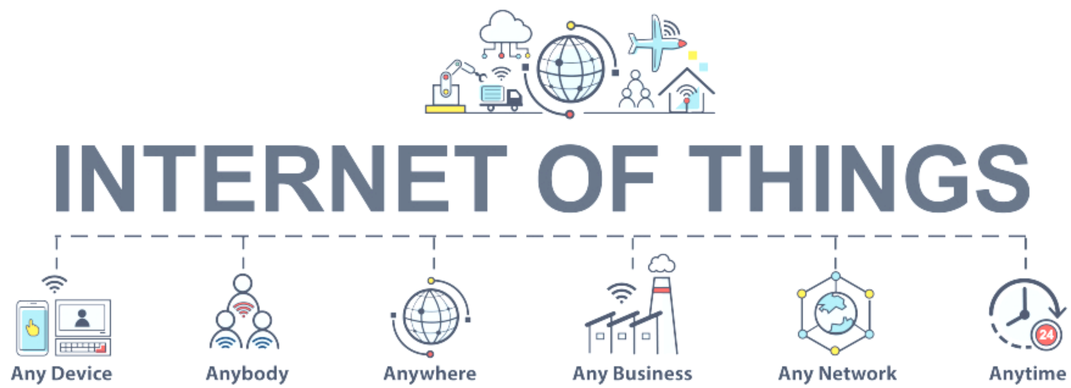


Figure 1: Definition about IoT

The Internet of things (IoT) describes the network of physical objects—“things” or objects—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.

Things have evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT can also be used in healthcare systems.

There are a number of serious concerns about dangers in the growth of IoT, especially in the areas of privacy and security, and consequently industry and governmental moves to address these concerns have begun including the development of international standards.

II. How does IoT work?

Figure 2: How IoT work?

Smartphones do play a large role in the IoT, however, because many IoT devices can be controlled through an app on a smartphone. You can use your smartphone to communicate with your smart thermostat, for example, to deliver the perfect temperature for you by the time you get home from work. Another plus? This can eliminate unneeded heating or cooling while you're away, potentially saving you money on energy costs.

IoT devices contain sensors and mini-computer processors that act on the data collected by the sensors via machine learning. Essentially, IoT devices are mini computers, connected to the internet, and are vulnerable to malware and hacking.

Machine learning is when computers learn in a similar way to humans — by collecting data from their surroundings — and it is what makes IoT devices smart. This data can help the machine learn your preferences and adjust itself accordingly. Machine learning is a type of artificial intelligence that helps computers learns without having to be programmed by someone.

That doesn't mean your smart speaker will discuss the key points of last night's big game with you. But your connected refrigerator may send you an alert on your smartphone that you're low on eggs and milk because it knows you're near a supermarket.

III. Applications of IoT

1. Consumer applications

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology, connected health, and appliances with remote monitoring capabilities.

a. Smart home

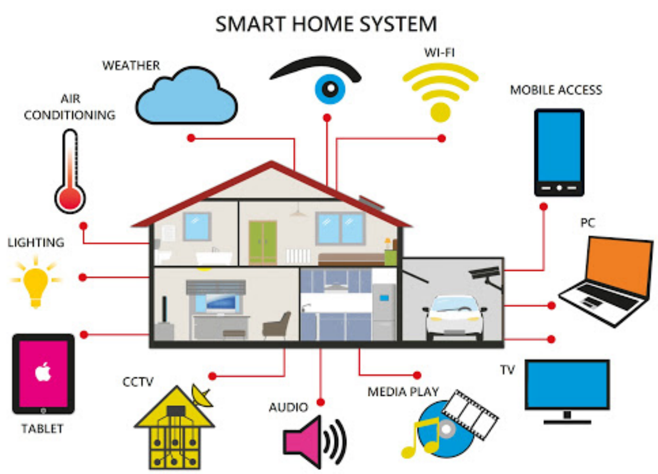


Figure 3: Smart home

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.

A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's HomeKit, manufacturers can have their home products and accessories controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge. There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products and these include the Amazon Echo, Google Home, Apple's HomePod, and Samsung's SmartThings Hub. In addition to the commercial systems, there are many non-proprietary, open source ecosystems; including Home Assistant, OpenHAB and Domoticz.

b. Elder care

One key application of a smart home is to provide assistance for those with disabilities and elderly individuals. These home systems use assistive technology to accommodate an owner's specific disabilities. Voice control can assist users with sight and mobility limitations while alert systems can be connected directly to cochlear implants worn by hearing-impaired users. They can also be equipped with additional safety features. These features can include sensors that monitor for medical emergencies such as falls or seizures. Smart home technology applied in this way can provide users with more freedom and a higher quality of life.

The term "Enterprise IoT" refers to devices used in business and corporate settings. By 2019, it is estimated that the EIoT will account for 9.1 billion devices.

2. Organizational applications

a. Medical and healthcare

Figure 4: Medical and healthcare

The **Internet of Medical Things** (**IoMT**) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. The IoMT has been referenced as "Smart Healthcare", as the technology for creating a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses. A 2015 Goldman Sachs report indicated that healthcare IoT devices "can save the United States more than $300 billion in annual healthcare expenditures by increasing revenue and decreasing cost." Moreover, the use of mobile devices to support medical follow-up led to the creation of 'm-health', used analyzed health statistics."

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. These sensors create a network of intelligent sensors that are able to collect, process, transfer, and analyze valuable information in different environments, such as connecting in-home monitoring devices to hospital-based systems. Other consumer devices to encourage healthy living, such as connected scales or wearable heart monitors, are also a possibility with the IoT. End-to-end health monitoring IoT platforms are also available for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

Advances in plastic and fabric electronics fabrication methods have enabled ultra-low cost, use-and-throw IoMT sensors. These sensors, along with the required RFID electronics, can be fabricated on paper or e-textiles for wireless powered disposable sensing devices. Applications have been established for point-of-care medical diagnostics, where portability and low system-complexity is essential.

As of 2018 IoMT was not only being applied in the clinical laboratory industry, but also in the healthcare and health insurance industries. IoMT in the healthcare industry is now permitting doctors, patients, and others, such as guardians of patients, nurses, families, and similar, to be part of a system, where patient records are saved in a database, allowing doctors and the rest of the medical staff to have access to patient information. Moreover, IoT-based systems are patient-centered, which involves being flexible to the patient's medical conditions. IoMT in the insurance industry provides access to better and new types of dynamic information. This includes sensor-based solutions such as biosensors, wearables, connected health devices, and mobile apps to track customer behaviour. This can lead to more accurate underwriting and new pricing models.

The application of the IoT in healthcare plays a fundamental role in managing chronic diseases and in disease prevention and control. Remote monitoring is made possible through the connection of powerful wireless solutions. The connectivity enables health practitioners to capture patient's data and applying complex algorithms in health data analysis.

b. Transportation

Figure 5: Digital variable speed-limit sign

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistics and fleet management, vehicle control, safety, and road assistance.

c. Building and home automation

IoT devices can be used to monitor and control the mechanical, electrical and electronic systems used in various types of buildings (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this context, three main areas are being covered in literature:

- The integration of the Internet with building energy management systems in order to create energy efficient and IOT-driven "smart buildings".
- The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviors.
- The integration of smart devices in the built environment and how they might be used in future applications.

3. Industrial applications

Also known as IIoT, industrial IoT devices acquire and analyze data from connected equipment, operational technology (OT), locations and people. Combined with operational technology (OT) monitoring devices, IIoT helps regulate and monitor industrial systems. Also, the same implementation can be carried out for automated record updates of asset placement in industrial storage units as the size of the assets can vary from a small screw till the whole motor spare part and misplacement of such assets can cause a percentile loss of manpower time and money.

a. Manufacturing

The IoT can connect various manufacturing devices equipped with sensing, identification, processing, communication, actuation, and networking capabilities. Network control and management of manufacturing equipment, asset and situation management, or manufacturing process control allow IoT to be used for industrial applications and smart manufacturing. IoT intelligent systems enable rapid manufacturing and optimization of new products, and rapid response to product demands.

Figure 6: Manufacturing

Digital control systems to automate process controls, operator tools and service information systems to optimize plant safety and security are within the purview of the IIoT. IoT can also be applied to asset management via predictive maintenance, statistical evaluation, and measurements to maximize reliability. Industrial management systems can be integrated with smart grids, enabling energy optimization. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by networked sensors.

b. Agriculture

There are numerous IoT applications in farming such as collecting data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content. This data can be used to automate farming techniques, take informed decisions to improve quality and quantity, minimise risk and waste, and reduce effort required to manage crops. For example, farmers can now monitor soil temperature and moisture from afar, and even apply IoT-acquired data to precision fertilization programs.

In August 2018, Toyota Tsusho began a partnership with Microsoft to create fish farming tools using the Microsoft Azure application suite for IoT technologies related to water management. Developed in part by researchers from Kindai University, the water pump mechanisms use artificial intelligence to count the number of fish on a conveyor belt, analyze the number of fish, and deduce the effectiveness of water flow from the data the fish provide. The specific computer programs used in the process fall under the Azure Machine Learning and the Azure IoT Hub platforms.

c. Maritime

IoT devices are in use monitoring the environments and systems of boats and yachts. Many pleasure boats are left unattended for days in summer, and months in winter so such devices provide valuable early alert of boat flooding, fire, and deep discharge of batteries. The use of global internet data networks such as Sigfox, combined with long life batteries, and microelectronics allows the engine rooms, bilge, and batteries to be constanted monitored and reported to a connected Android & Apple applications for example.

IV. Some examples for real world application of IoT

1. Environment

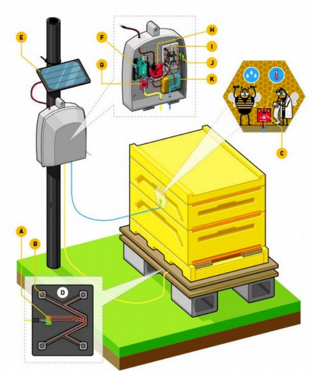


Figure 7: Environment IoT in honey bee

In recent times, we are becoming more and more aware of the environment and the damage that human activities have caused. Now, we are slowly trying to correct our mistakes and tackle environmental issues. We can make use of IoT applications to help us with this goal.

Some of the areas, where IoT are working to preservation of bees. Honey Bees are facing a threat in this polluted world. Climate change is affecting them as well. However, by implanting IoT devices, beekeepers can take better charge of preserving hives. Using IoT with connected sensors, it is possible to keep track of the hive temperature, amount of food present in the hives and also, the pollen collection. IoT can be also used in waste management also, helping us take better care of our environment.

2. Smart Home Applications

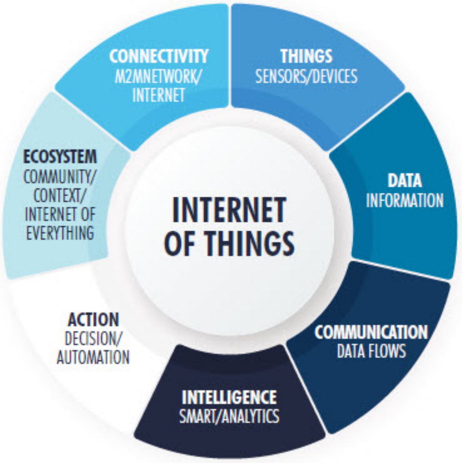
Smart homes are probably the most common of all IoT use cases. The concept of smart homes has revolved around the internet for a long time. When Mark Zuckerberg displayed his smart home Jarvis, many people felt that they could actually fulfill their dream of having a Smart home.

Figure 8: Smart home application

A smart home works with the active use of IoT applications. Smart home devices collect and share information with one another in an integrated platform and automate their actions based on the owner's preference. Hence, it is clear that they engage in a learning process to understand the preferences of their owner.

There are many IoT use cases related to Smart home appliances, but one of them can be Smart Thermostats, which monitor and control home temperatures to the comfort of the owner. Plus, there is smart lighting as well where the lighting adjusts themselves based on the user preference as well as external lighting.

V. IoT characteristics



You can define IoT by looking at the various characteristics in the broader context. We see all of these characteristics coming back in most Internet of Things definitions out there (further below is an overview with some of these IoT definitions).

There are 7 crucial IoT characteristics:

1. Connectivity:

This doesn't need too much further explanation. With everything going on in IoT devices and hardware, with sensors and other electronics and connected hardware and control systems there needs to be a connection between various levels.

Figure 9: IoT characteristics

2. Things:

Anything that can be tagged or connected as such as it's designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items.

3. Data:

Data is the glue of the Internet of Things, the first step towards action and intelligence.

4. Communication:

Devices get connected so they can communicate data and this data can be analyzed. Communication can occur over short distances or over a long range to very long range. Examples: Wi-Fi, LPWA network technologies such as LoRa or NB-IoT.

5. Intelligence:

The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from big data analytics (also artificial intelligence).

6. Action:

The consequence of intelligence. This can be manual action, action based upon debates regarding phenomena (for instance in smart factory decisions) and automation, often the most important piece.

7. Ecosystem:

The place of the Internet of Things from a perspective of other technologies, communities, goals and the picture in which the Internet of Things fits. The Internet of Everything dimension, the platform dimension and the need for solid partnerships.

P2 REVIEW STANDARD ARCHITECTURE, FRAMEWORKS, TOOLS, HARDWARE AND APIS AVAILABLE FOR USE IN IOT DEVELOPMENT

I. Architecture of Internet of Things

Internet of Things (IoT) technology has a wide variety of applications and use of Internet of Things is growing so faster. Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it has not a standard defined architecture of working which is strictly followed universally. The architecture of IoT depends upon its functionality and implementation in different sectors. Still, there is a basic process flow based on which IoT is built.

Here in this article we will discuss basic fundamental architecture of IoT i.e., 4 Stage IoT architecture.

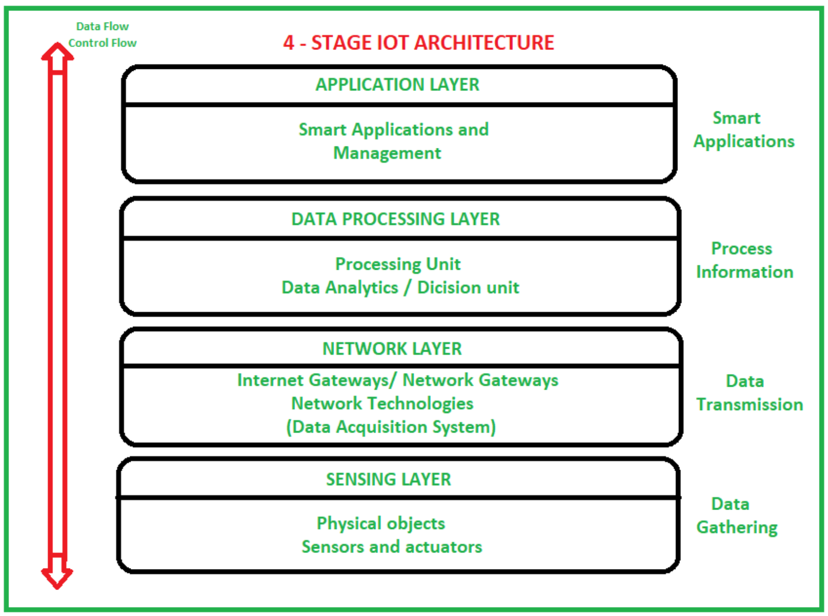


Figure 10: 4 Stage IoT architecture.

So, from the above image it is clear that there is 4 layers are present that can be divided as follows: Sensing Layer, Network Layer, Data processing Layer, and Application Layer.
These are explained as following below.

1. Sensing Layer

Sensors, actuators, devices are present in this Sensing layer. These Sensors or Actuators accepts data(physical/environmental parameters), processes data and emits data over network.

2. Network Layer

Internet/Network gateways, Data Acquisition System (DAS) are present in this layer. DAS performs data aggregation and conversion function (Collecting data and aggregating data then converting analog data of sensors to digital data etc). Advanced gateways which mainly opens up connection between Sensor networks and Internet also performs many basic gateway functionalities like malware protection, and filtering also sometimes decision making based on inputted data and data management services, etc.

<u>3. Data processing Layer</u>

This is processing unit of IoT ecosystem. Here data is analyzed and pre-processed before sending it to data center from where data is accessed by software applications often termed as business applications where data is monitored and managed and further actions are also prepared. So here Edge IT or edge analytics comes into picture.

<u>4. Application Layer</u>

This is last layer of 4 stages of IoT architecture. Data centers or cloud is management stage of data where data is managed and is used by end-user applications like agriculture, health care, aerospace, farming, defense, etc.

II. Framework of Internet of Things

<u>1. What is IoT Framework</u>

IoT is a key part of a large IoT ecosystem, which promotes and links all elements in the scheme. It allows device management, handles communication protocols on software and hardware, collects / analyses information, improves information flow and intelligent apps functionality.

<u>2. List of IoT Framework</u>

a. Cisco IoT Cloud Connect

Cisco IoT Cloud Connect provides robust, automated, and highly secure connectivity for the enterprise. IoT data management is done by the Cisco Kinetic IoT platform to extract, move and compute the data. As Cisco is very famous for its security services, it protects IoT deployment against threats with a secure IoT architecture.

b. DeviceHive IoT

DeviceHive is another rich IoT open-source platform that is distributed under the Apache 2.0 license and can be used and changed free of charge. It provides deployment options for Docker and Kubernetes and can be downloaded and used both by public and personal cloud. You can run batch analysis and machine learning above and beyond your device information. DeviceHive supports several libraries, including Android and iOS.

c. Oracle IoT

We surely include Oracle, a worldwide software company known to offer its top level of solutions in database management, and business software, as we compare the top Internet-of-Things platforms. Oracle offers its flexible environment outstanding company possibilities to create company applications. Oracle supports the processing and builds large-scale IoT networks with very wide data. The use of advanced security systems to protect IoT systems against external threats is another worth mentioning. Since these systems usually have different devices, some of which have no security tool, it is not sufficiently justifiable to implement centralized security measures.

d. Microsoft Azure IoT

Without the Microsoft Azure solution, a cloud service giant with AWS and Google Cloud platform, the comparison of our IoT platform will be not complete. The Microsoft Azure IoT Suite provides preconfigured solutions and the ability to personalize and develop new solutions to meet the project requirements. The strongest safety mechanisms, superb scalability and simple integration with your current or future systems are achieved through Microsoft Azure Internet of thing Suite.

e. Google Cloud Platform – IoT framework

Things can be done by Google. Google Cloud is one of the best IoT systems available today with its end-to-end platform. Google stands out from the others because it can process the large quantity of information using Cloud IoT Core. Due to Google's Cloud Data Studio and Big Query you get advanced analysis. With the help of Google Cloud Platform, you can accelerate your business and with that, you can speed up your device.

f. IBM Watson – IoT framework

We cannot expect the Big Blue to miss the chance to make a difference in the IoT segment. IBM Watson is very popular among the internet of thing platform among developers. The Bluemix hybrid cloud-supported Watson IoT platform allows developers to use IoT-applications easily. IBM Watson manages the secure communication and also data storage. Real-time data exchange also is done by IBM Watson.

g. Amazon Web Services (AWS) IoT framework

Amazon Web Services (AWS) is an IoT platform provided by Amazon. This IoT platform provides cloud computing, database, and security services through the AWS Console. There are so many other services such as Regions, Availability Zones, and Virtual Private Clouds (VPCs). It helps to ease out the improving durability, distribution, availability of the application. It provides Registry for recognizing devices, Secure Device Gateway, Compatible Software Development Kit for devices which AWS partnered with HW manufacturers like Intel, Texas Instruments, Broadcom and Qualcomm.

III. Tools of Internet of Things

IoT Tools stands for the Internet of Things Tools. It is a network or connection of devices, vehicles, and equipment applying embedded electronics, home appliances, buildings and many more. This helps in collecting and exchanging different kinds of data. It also helps the user to control the devices remotely over a network.

Today in the internet-driven world, IoT has engulfed the IT industry and is the latest buzzword. It has opened many new horizons for companies and developers working on IoT. Many exceptional products have been developed due to IoT app development. Companies providing Internet of Things solution are creating hardware and software designs to help the IoT developers to create new and remarkable IoT devices and applications.

List of IoT tools, devices

a. Eclipse IoT

This tool or instrument allows the user to develop, adopt and promote open source IoT technologies. It is best suited to build IoT devices, Cloud platforms, and gateways. Eclipse supports various projects related to IoT. These projects include open-source implementations of IoT protocols, application frameworks and services, and tools for using Lua programming language which is promoted as the best-suited programming language for IoT.

b. Arduino

Arduino is an Italy based IT company that builds interactive objects and microcontroller boards. It is an open-source prototyping platform that offers both IoT hardware and software. Hardware specifications can be applied to interactive electronics and software includes Integrated Development Environment (IDE). It is the most preferable IDEs in all IoT development tools. This platform is easy and simple to use.

c. Kinoma

It is a Marvell semiconductor hardware prototyping platform. It enables three different projects. To support these projects two products are available Kinoma Create and Element Board. Kinoma Create is a hardware kit for prototyping electronic and IoT enabled devices. Kit contains supporting essentials like Bluetooth Low Energy (BLE), integrated Wi-Fi, speaker, microphone and touch screen. Element Board is the smallest JavaScript-powered IoT product platform.

d. Raspbian

This IDE is created for Raspberry Pi board. It has more than 35000 packages and with the help of precompiled software, it allows rapid installation. It was not created by the parent organization but by the IoT tech enthusiasts. For working with Raspberry Pi, this is the most suitable IDE available.

IV. Hardware of Internet of Things

In the Internet of Things, hardware comes in many forms, whether the underlying processors control the phones, the sensors collecting information from the physical world, or the edge machines processing and analyzing the data. At the heart of any wired venture is IoT hardware and the technical capabilities of these boards have only become more important as the Internet of Things has developed. But, because of the sheer number of design boards and modules in the room, choosing the right IoT hardware for a project can be daunting. In this article, we will look at different aspects of IoT hardware and see how these devices communicate data to the internet.

1. IoT Hardware Devices

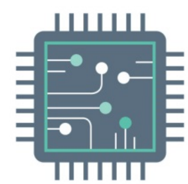
The building blocks of an IoT device are remarkably similar, whether undertaking projects related to the wearable device, an integrated lighting system, or even a jet engine. Wireless sensor node consists of three major hardware components they are sensors, microcontrollers, and communication medium.

a. Sensors

Sensors are the most critical hardware in IoT applications and are used to gather information from the surroundings. These systems are made up of power management modules, RF, energy and sensing modules. Communication from Wi-Fi, Bluetooth, transceiver, BAW, and duplexer is managed by an RF module.

b. Microcontrollers

A microcontroller is a device in a single integrated circuit devoted to executing a single task and running an application. This contains programmable peripherals for contains programmable, memory unit, and a CPU. Microcontrollers are designed primarily for embedded applications and are widely used in remotely operated electronic devices such as mobile phones, washing machines, microwaves, and cameras.



c. Other IoT hardware

Smart wearable devices such as smart memory, glasses, rings, and shoes are examples of IoT hardware. Smart devices allow us to access more of the content and resources that we love and create a new approach to collaboration as part of an IoT network. Desktop, mobile phones, and tablets are standard command center and remains an integral part of IoT application. Other network distribution devices like switches, hubs, and routers act as a key connector in IoT application.

2. IoT Hardware Providers

In today's market, one can find many IoT hardware providers who will be able to provide the required hardware-based up on the project requirement. Let's take a quick overview of a few of the hardware providers.

- Adafruit offers DIY electronic hacking courses online and provides a space to learn. 'Adafruit Feather' is a production line of boards designed for fly prototyping. This production line includes a wide catalog of accessories that speed up IoT application development.
- Arduino is the omnipresent name in the space for electronic development. The company offers a range of open-source development kits, billing software, and microcontrollers.
- Lantronix is a company that offers Infrastructure, modules and gateway kit to support connectivity between IoT applications. This California based company has recently launched advanced gateways based on XP200 industrial standard.
- Espressif is well known for its low energy consuming IoT hardware applications for Wi-Fi and Bluetooth. Espressif is popular for its processors, modules, and production boards series ESP826. Most industry-wide development boards are working on Espressif chipset.

V. IoT Applications or APIs

1. What is an IoT API?

The application program (or programming) interface, or API, is arguably what really ties together the connected "things" of the "internet of things." IoT APIs are the points of interaction between an IoT device and the internet and/or other elements within the network.

As API management company Axway puts it, "APIs are tightly linked with IoT because they allow you to securely expose connected devices to customers, go-to-market channels and other applications in your IT infrastructure."

All IoT apps have certain commonalities or 'building blocks' irrespective of the request. Things, Data Acquisition Module, Data processing module, and communication module together consist of building blocks of IoT application.

1. Thing

The "thing" is fully integrated into the smart device in many IoT products. Think about things such as a smart water pump or an autonomous vehicle. These products can Self-control and manage themselves. In other applications "thing" stands alone as a "stagnant" device, and to make it a smart device, a separate product is connected to it. This is very popular in industrial applications where companies have existing products, and by linking them to the internet they want to make them "smart." jet engines, turbines, conveyor belts, etc. are some examples.

2. Data Acquisition

The data acquisition module focuses on obtaining physical signals from the "thing" that can be manipulated by a computer and converting them into digital signals. This is the component of hardware that includes all sensors that acquire real-world signals such as temperature, light, vibration, motion, etc. However, the module for data acquisition includes more than sensors. It also includes the appropriate hardware for the computer to convert the sensor signal into digital information. This includes the conditioning of signals, conversion from analog to digital and data interpretation.

3. Data Processing

The third essential component of IoT hardware is the data processing module. This unit performs a series of operations on the data, performs spatial analysis, internal data storage, and performing any other edge computing operations. To have a deep discussion about this system with the engineering team, one doesn't require any expertise in computer architecture. Your job should be to consider the product's overarching goal and ask the right questions to direct your team in making the right decisions. Depending on your data retention policy, the local storage requirement can be decided. If your product is being designed for offline operations, connection running time local storage space needs to be predefined.

4. Communication

The communication module is the last but very essential part of the device's hardware. This is a unit that allows communication between devices and storage either locally or in the cloud space. This module can include, to name a few, communication ports like USB, Modbus or Ethernet/IP. It may further consist of wireless communication radio technology such as Wireless fidelity.

2. Why are IoT APIs important?

IBM called IoT APIs one of its top IoT trends for 2016.

"APIs are the market enabler, and 'internet of things' devices would be useless without them. By exposing data that enables multiple devices to be connected, APIs provide an interface between the internet and the things to reveal previously unseen possibilities," said Chris O'Connor, IBM's GM for IoT, in a blog entry. "In the year to come, the power and importance of APIs will be at the forefront of the conversation around enabling—and more important—monetizing the 'internet of things.'"

M1. EVALUATE THE IMPACT OF COMMON IOT ARCHITECTURE, FRAMEWORKS, TOOLS, HARDWARE AND APIS IN THE SOFTWARE DEVELOPMENT LIFECYCLE.

UNIVERSITY of GREENWICH ĐẠ

Alliance with FPT Education Liên k

M2 REVIEW SPECIFIC FORMS OF IOT ARCHITECTURE, FRAMEWORKS, TOOLS, HARDWARE AND APIS FOR DIFFERENT PROBLEM-SOLVING REQUIREMENTS

P3 INVESTIGATE ARCHITECTURE, FRAMEWORKS, TOOLS, HARDWARE AND API TECHNIQUES AVAILABLE TO DEVELOP IOT APPLICATIONS

I. What is Arduino?

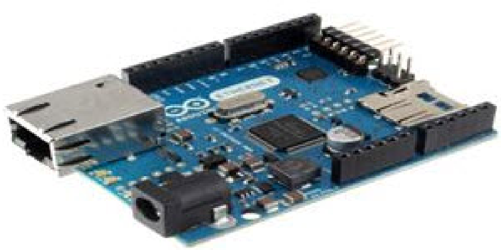


Figure 11: Arduino board

An Arduino board is a one type of microcontroller based kit. The first Arduino technology was developed in the year 2005 by David Cuartielles and Massimo Banzi. The designers thought to provide easy and low cost board for students, hobbyists and professionals to build devices. Arduino board can be purchased from the seller or directly we can make at home using various basic components. The best examples of Arduino for beginners and hobbyists includes motor detectors and thermostats, and simple robots. In the year 2011, Adafruit industries expected that over 3lakhs Arduino boards had been produced. But, 7lakhs boards were in user's hands in the year 2013. Arduino technology is used in many operating devices like communication or controlling.

Arduino Technology

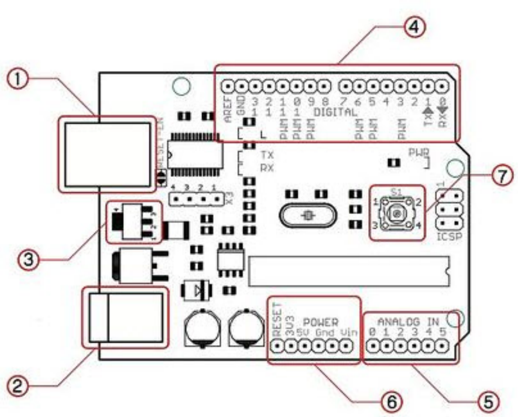


Figure 12: Arduino Pin Diagram

The most important parts on the Arduino board high lighted in red:

1: USB connector
2: Power connector
3: Automatic power switch
4: Digital pins
5: Analog pins
6: Power pins
7: Reset switch

A typical example of the Arduino board is Arduino Uno. It includes an ATmega328 microcontroller and it has 28-pins

The pin configuration of the Arduino Uno board is shown in the above. It consists of 14-digital I/O pins. Where in 6 pins are used as pulse width modulation o/ps and 6 analog i/ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset button, and an ICSP header. Arduino board can be powered either from the personal computer through a USB or external source like a battery or an adaptor. This board can operate with an external supply of 7-12V by giving voltage reference through the IORef pin or through the pin Vin.

Digital I/Ps

It comprises of 14-digital I/O pins, each pin take up and provides 40mA current. Some of the pins have special functions like pins 0 & 1, which acts as a transmitter and receiver respectively. For serial communication, pins-2 & 3 are external interrupts, 3,5,6,9,11 pins delivers PWM o/p and pin-13 is used to connect LED.

Analog i/ps: It has 6-analog I/O pins, each pin provide a 10 bits resolution.

Aref: This pin gives a reference to the analog i/ps.

Reset: When the pin is low, then it resets the microcontroller.

II. Arduino Architecture

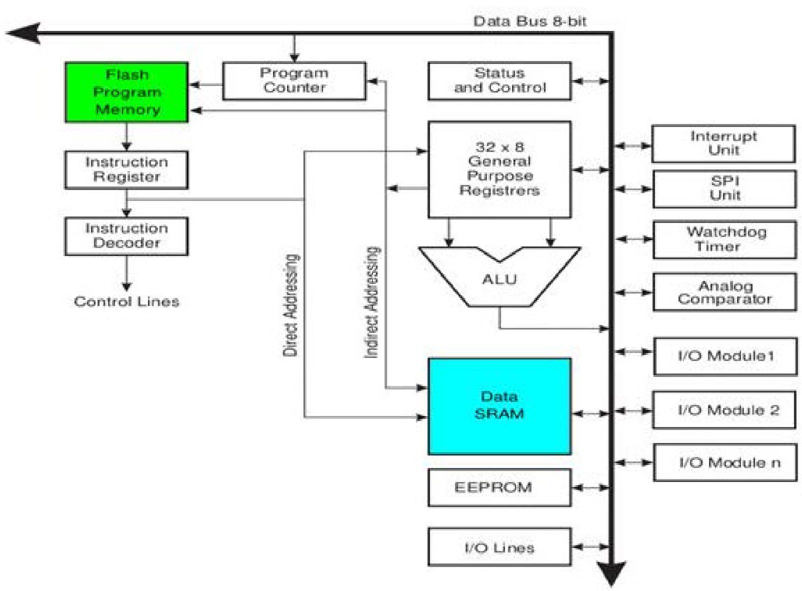


Figure 13: Arduino Architecture

Basically, the processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Wherein the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed.

III. Arduino Hardware

Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in –duino.

Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I^2C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the LilyPad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

Arduino microcontrollers are pre-programmed with a boot loader0 that simplifies uploading of programs to the on-chip flash memory. The default bootloader of the Arduino Uno is the Optiboot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used.

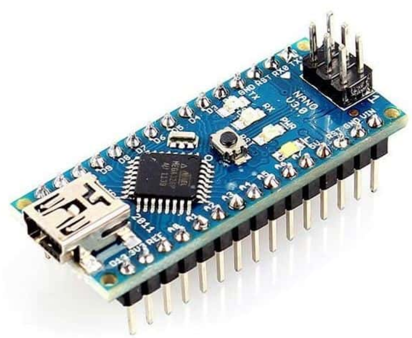


Figure 14: Arduino NANO Microcontroller

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards.

Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility.

Figure 15: Arduino Board

IV. Arduino Tool, Framework and Application programming interface or APIs

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

1. IDE

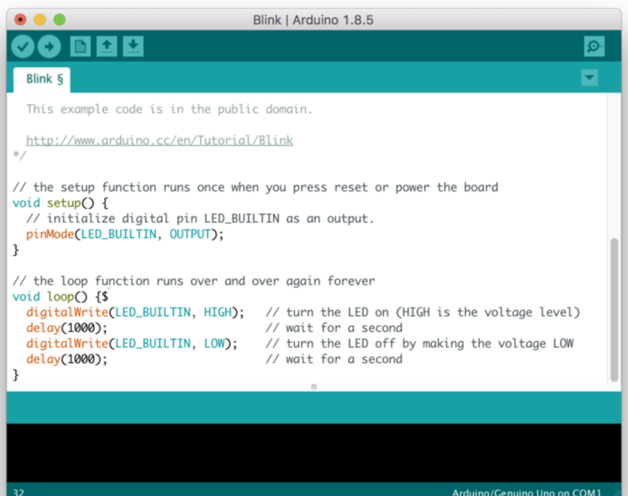


Figure 16: Arduino IDE Blynk

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages *Processing* and *Wiring*. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

2. Pro IDE

On October 18, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, autocompletion support, and Git integration.[62] The application frontend is based on the Eclipse Theia Open Source IDE. The main features available in the alpha release are:[63]

- Modern, fully featured development environment
- Dual Mode, Classic Mode (identical to the Classic Arduino IDE) and Pro Mode (File System view)
- New Board Manager
- New Library Manager
- Board List
- Basic Auto-Completion (Arm targets only)
- Git Integration
- Serial Monitor
- Dark Mode

3. Sketch

A *sketch* is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension **.ino**. Arduino Software (IDE) pre-1.0 saved sketches with the extension **.pde**.

A minimal Arduino C/C++ program consists of only two functions:

- `setup()`: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function `main()`.
- `loop()`: After `setup()` function exits (ends), the `loop()` function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function `while(1)`.

4. Blink example

Most Arduino boards contain a light-emitting diode (LED) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions. A typical program used by beginners, akin to Hello, World!, is "blink", which repeatedly blinks the on-board LED integrated into the Arduino board. This program uses the functions `pinMode()`, `digitalWrite()`, and `delay()`, which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer. Libraries are an open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

```
#define LED_PIN 13                  // Pin number attached to LED.

void setup() {
    pinMode(LED_PIN, OUTPUT);       // Configure pin 13 to be a digital output.
}

void loop() {
    digitalWrite(LED_PIN, HIGH);    // Turn on the LED.
    delay(1000);                    // Wait 1 second (1000 milliseconds).
    digitalWrite(LED_PIN, LOW);     // Turn off the LED.
    delay(1000);                    // Wait 1 second.
}
```

Figure 18: Sample code

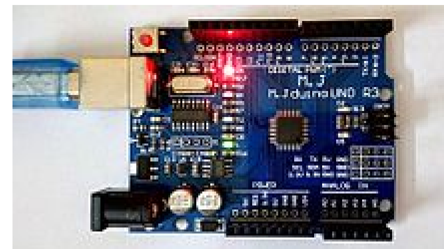


Figure 17: Power LED (red) and User LED (green) attached to pin 13 on an Arduino compatible board

5. Framework

Arduino Wiring-based Framework allows writing cross-platform software to control devices attached to a wide range of Arduino boards to create all kinds of creative coding, interactive objects, spaces or physical experiences

Name	MCU	Debug	Frequency	RAM	Flash
3D Printer control board	STM32F446RET6	External	180 MHz	128 kB	512 kB
3D Printer Controller	STM32F407VET6	External	168 MHz	192 kB	512 kB
3D printer controller	STM32F765VIT6	On-board	216 MHz	512 kB	2048 kB
3DP001V1 Evaluation board for 3D printer	STM32F401VGT6	On-board	84 MHz	96 kB	512 kB
4D Systems gen4 IoD Range	ESP8266		80 MHz	80 kB	512 kB
4DSystems PICadillo 35T	32MX795F512L		80 MHz	128 kB	508 kB
Adafruit BLM Badge	SAMD21E18A	External	48 MHz	32 kB	256 kB
Adafruit Bluefruit Micro	ATMEGA32U4	On-board	8 MHz	2.5 kB	28 kB
Adafruit Bluefruit nRF52832 Feather	NRF52832	External	64 MHz	64 kB	512 kB
Adafruit Circuit Playground Classic	ATMEGA32U4	On-board	8 MHz	2.5 kB	28 kB
Adafruit Circuit Playground Express	SAMD21G18A	External	48 MHz	32 kB	256 kB
Adafruit CLUE nRF52840	NRF52840	External	64 MHz	243 kB	796 kB
Adafruit Crickit M0	SAMD21G18A	External	48 MHz	32 kB	256 kB
Adafruit ESP32 Feather	ESP32	External	240 MHz	320 kB	4096 kB

Figure 19: Arduino Framework

6. Application

- Arduboy, a handheld game console based on Arduino
- Arduinome, a MIDI controller device that mimics the Monome
- Ardupilot, drone software and hardware
- ArduSat, a cubesat based on Arduino.
- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
- Data loggers for scientific research.[72][73][74][75]
- OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars
- OpenEVSE an open-source electric vehicle charger
- XOD, a visual programming language for Arduino

P4 DETERMINE A SPECIFIC PROBLEM TO SOLVE USING IOT.

I. Give out your idea, reason about IoT application used for solving problem in real world

Figure 20: Arduino example board

For choosing the project, I will use 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.

We will use Arduino to build a plant watering system, the system will use a soil humidity sensor to check if the dirt in pot is still wet, if the water too low the system will automatically water the plant. This system can help gardener who want they plants always moist and green or person who want they plant still green even they forgot to water it or too busy to water it.

Figure 21: Project example

II. Why we choose Arduino?

The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.

And over all it’s easy, cheap but very effectively to use.

III. Give out design and explain how system work

In the design I will use Tinkercad, this is just an example design show how basic the system will work. Because there is no Soil Humid sensor so we will replace it to a Terminal sensor to demo the project. First when you open the system it will show welcome message like in Figure 22.

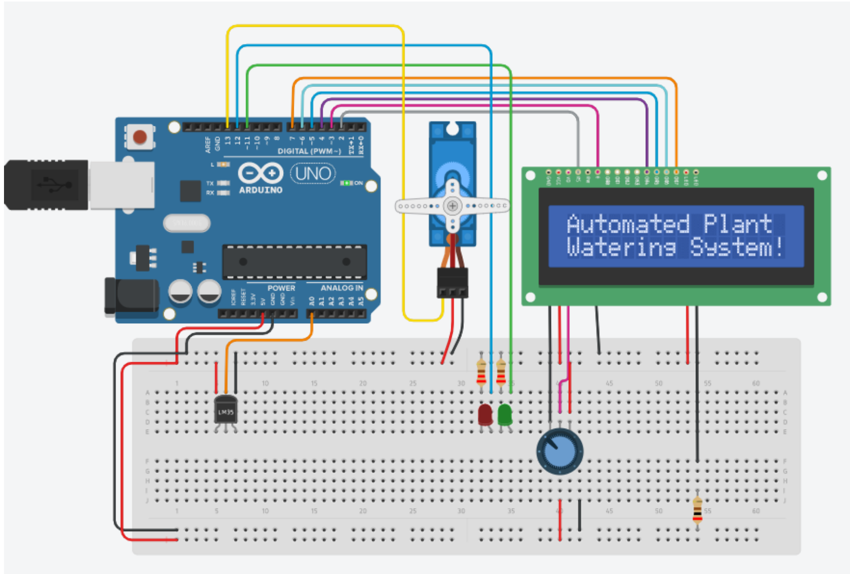
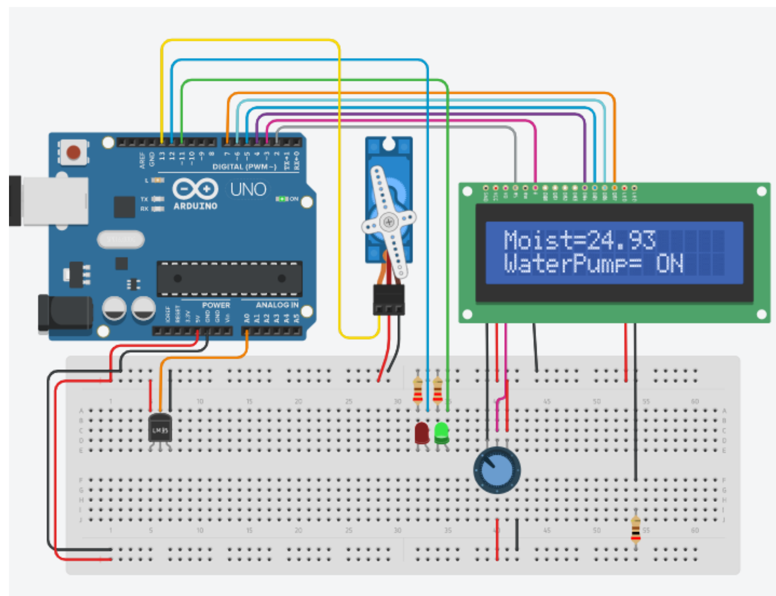


Figure 22: Water system welcome

The system will automatically report the percent of water in your pot, and if the percent of water less than 40% the water pump will active to watering your plant. The green Led will show you if the system working or you can see it through LCD



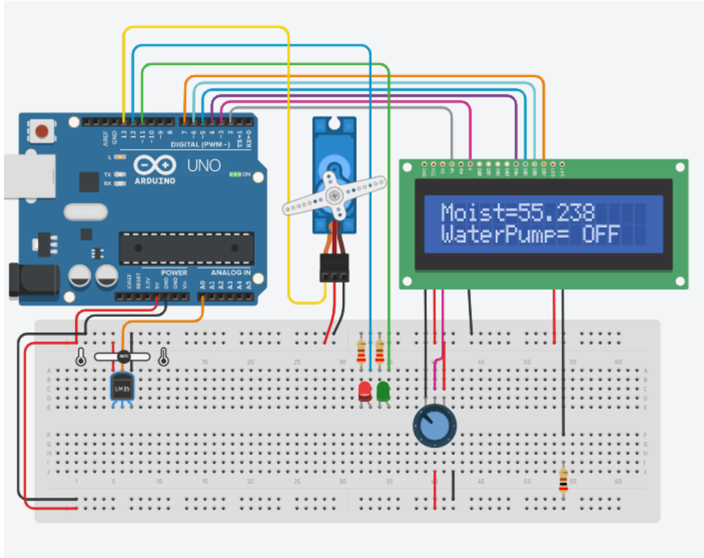


Figure 23: Water pump off

But if the water percent more than 40% the pump will off, but if the water loop at 40% will cause the system on and off repeatedly so even the water percent after the water pump open more than 40% the system will close the pump after 5 or 10 second depend on the setting to avoid the loop or system repeat on/off the pump to raise the life time of pump. The red Led will tell you if the system stopped or working.

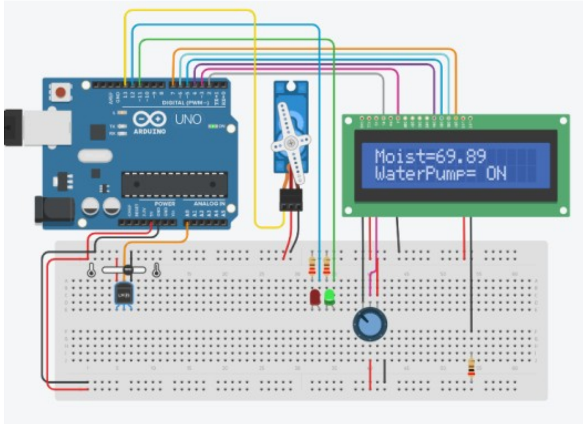


Figure 24: Water pump off delay

Moreover the user can check the water percent on pot from far away and force Water pump to on/off through Blink app.

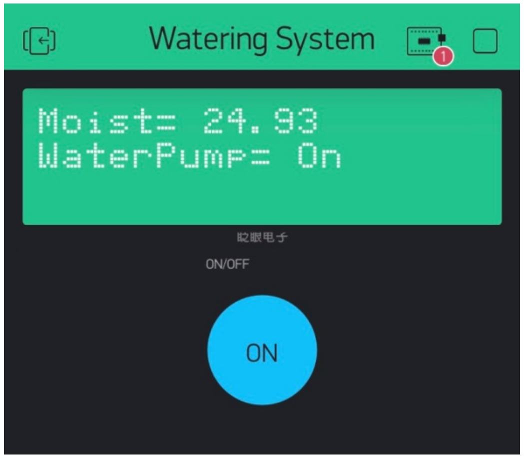


Figure 25: Force pump on

Figure 26: Force pump off

IV. Explain how it works

Basically how system work is if water in pot < 40% the water pump will open, if the water pump raise the water percent in pot more than 40% the pump will continue in few seconds then stop. To make the system work we will use if/else, delay code to program for the system.

```
void loop() {

  int value = analogRead(LM35);
  float Moist = value * 500.0 / 1023.0;
  lcd.setCursor(6,0);
  lcd.print(Moist);
  lcd.setCursor(11,1);


  if (Moist < 40){
    myservo.write(10);
    delay(15);
    digitalWrite(LedRed, LOW);
    digitalWrite(LedGreen, HIGH);
    lcd.print("ON ");
  }
  else {
    delay(3000);
    myservo.write(110);
    delay(15);
    digitalWrite(LedRed, HIGH);
    digitalWrite(LedGreen, LOW);
    lcd.print("OFF");
  }

   delay(1000);
}
```

V. Which devices are used?

In the example design above we will use:

- 1 Arduino Wi-Fi board
- 1 Bread board
- 1 Soil humidity sensor
- 2 Led (Different color)
- 1 Potentiometer (For LCD)
- 3 Resistor (2 for Led and 1 for LCD)
- 1 Micro servo
-

M3 Select the most appropriate IoT architecture, frameworks, tools, hardware and API techniques to include in an application to solve this problem.

M4 Apply your selected techniques to create an IoT application development plan.

D1 Evaluate specific forms of IoT architecture and justify their use when designing software applications.

D2 Make multiple iterations of your IoT application and modify each iteration with enhancements gathered from user feedback and experimentation.

REFERENCE

https://en.wikipedia.org/wiki/Internet_of_things
https://us.norton.com/internetsecurity-iot-what-is-the-internet-of-things.html#:~:text=IoT%20devices%20contain%20sensors%20and,the%20sensors%20via%20machine%20learning.&text=Machine%20learning%20is%20when%20computers,what%20makes%20IoT%20devices%20smart.
https://www.upgrad.com/blog/top-iot-real-world-applications/
https://www.i-scoop.eu/internet-of-things-guide/internet-of-things/
https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/
https://www.educba.com/iot-framework/
https://www.educba.com/iot-tools/
https://www.educba.com/iot-hardware/
https://www.rcrwireless.com/20161018/internet-of-things/what-is-an-iot-api-tag6-tag99
https://www.watelectronics.com/arduino-technology-architecture-and-applications/
https://en.wikipedia.org/wiki/Arduino
https://platformio.org/frameworks/arduino
https://www.arduino.cc/en/guide/introduction