DESIGN AND IMPLEMENTATION INTERNET OF THINGS (IOT) ANDROID REMOTE-CONTROLLED QUADCOPTER

Budi Berlinton Sitorus[1],
[1] Staff pengajar Program Studi Teknik Informatika
Institut Bisnis dan Informatika Kwik Kian Gie
Jl. Yos Sudarso Kav. 87 Sunter Jakarta Utara 14350
http://www.kwikkiangie.ac.id
budi.sitorus@kwikkiangie.ac.id

ABSTRACT

This Internet of Things (IoT) is leading path to the smart world with ubiquitous computing and networking to ease different tasks around users and provide other tasks, such as easy monitoring of different phenomenon surrounding us. In this research , a Wi-Fi controlled remote quadcopter is designed and built, enhanced with computing and communication technologies and join the communication framework, meeting a variety of services based on machine-to-machine interactions using wired and wireless communication.
A quadcopter can be made several ways , one of them is using a simple microcontroller. In this research Remote quadcopter is built by components such 2300KV brushless motor, 2 pairs of propeller, 4 ESCs, 11mAh LippoBaterry, Gyro+accelerometer board and other quadcopter components. As the controller to the quadcopter , Arduino Uno R3 module is utilized.
Results to the research is a remote quadcopter controlled by an Android Smartphone as combination of Arduino UNO and Wi-Fi module. The current results open many possibilities for future development for features of quadcopter.

Keywords: IoT, Remote quadcopter, Arduino

1. Pendahuluan

The Internet of Things (IoT) is becoming more and more integrated into our daily lives, but it seems like few people really be aware what it is, what it can do, or just how ubiquitous it really is. The following stats and facts will describe how IoT as a trend in recent :

- In 2018, the number of IoT-connected devices will surpass mobile devices.
- It's projected that by 2020, 75% of new cars will come with built-in IoT connectivity.
- There were already more things connected to the internet than there were people on Earth as early as 2008. It's predicted that there will be more than 50 billion things connected to the internet by 2020.
- 87% of people don't actually know what "the Internet of Things" means. General Electric believes that the IoT can be used to improve oil and gas exploration. Just a 1% improvement could result in $90 billion

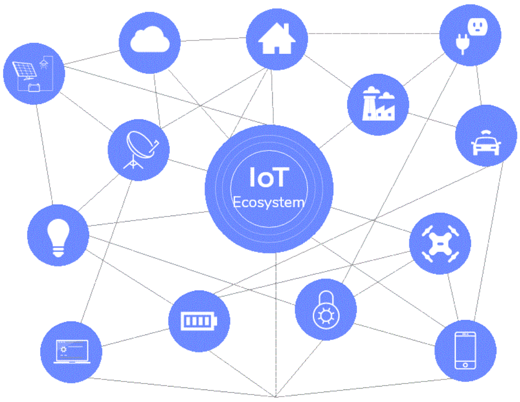

Figure 1. IoT connection illustration

Figure 1 above depicts the links between devices in IoT ecosystem. Because of the IoT, now more and more electronic devices are connected to the internet and are talking to each other behind the scene. According to Atzori [1] , Identification and tracking technologies, wired and wireless sensor and actuator networks, enhanced communication protocols (shared with the Next Generation Internet), and distributed intelligence for smart objects are just the most relevant., any serious contribution to the advance of the Internet of Things must necessarily be the result of synergetic activities conducted in different fields of knowledge, such as telecommunications, informatics, electronics and social science.

To support this, Gubbi [2], Ubiquitous sensing enabled by Wireless Sensor Network (WSN) technologies cuts across many areas of modern day living. It offers the ability to measure, infer and understand environmental indicators, from delicate ecologies and natural resources to urban environments. The proliferation of these devices in a communicating–actuating network creates the Internet of Things (IoT), wherein sensors and actuators blend seamlessly with the environment around us, and the information is shared across platforms in order to develop a common operating picture (COP). The recent adaptation of a variety of enabling wireless technologies such as RFID tags and embedded sensor and actuator nodes supporting the IoT, and it has stepped out of its infancy and is the next revolutionary technology in transforming the Internet into a fully integrated responsive design is a parallel branch of design called adaptive design.

The Internet of Things (IoT) is a new technology paradigm envisioned as a global network of machines and devices capable of interacting with each other. The IoT is recognized as one of the most important areas of future technology and is gaining vast attention from a wide range of industries. Arduino as a microcontroller plays an important role in IoT environment. A remote control quadcopter is available in the market but it costs expensive. There are varies models available and quadcopter commonly used specific frequency to handle movements. The model being built is controlled remotely by Wi-Fi connection from an android smartphone. This new model uses arduino uno as the main controller for the drone.

The aims, objectives and methodology of the research are described briefly as follow :

- This study primarily employs implementation of IoT as one of trending product .
- The study represents Design and Implementation Internet of Things (IoT) Android Remote-Controlled Quadcopter.

From the problems defined above, in order to utilize the benefit of the Arduino as the microcontroller , this research was undertaken for implementing the use of Arduino Uno R3 module combined with Gyro +accelerometer board, motors and their ESC, propeller, and of course a battery. Purpose of the research is to create a quadcopter Wi-Fi controlled through a smartphone as an implementation of IoT

2. Literature Review

Internet of Things is somehow a leading path to the smart world with ubiquitous computing and networking to ease different tasks around users and provide other tasks, such as easy monitoring of different phenomenon surrounding us. In Internet of Things, environmental and daily life items, named also things, objects, or machines, are enhanced with computing and communication technologies and join the communication framework, meeting a variety of services based on person-to-person, person-to-machine, machine-to-person, and machine-to-machine interactions using wired and wireless communication. [5].

Another example of IoT utilization is home automation. It is term when using IoT technology to remotely control tasks in home that normally being performed manually. For example, instead of walking up to living room lamp, reaching out with hand, and turning it on, it can be turned on and off from anywhere with access control, whether that is using a panel somewhere in home or on computer, or from half-way around the globe using smartphone.
According to [6], IoT solutions are complex. The integration of connected devices and IT services poses major challenges in networking, communication, data volume, real-time data analysis, and security. IoT solutions involve many different technologies and require complex development cycles, including significant testing and ongoing monitoring. To overcome these challenges, IT organizations must at least concern about :

- Develop a comprehensive technical strategy to address the complexity.
- Define a reference architecture for their IoT solution.
- Develop required skills to design, develop, and deploy the solution.
- Define your IoT governance processes and policies

IoT solution governance can be viewed as the application of business governance, IT governance, and enterprise architecture (EA) governance to Internet of Things as seen on figure 2

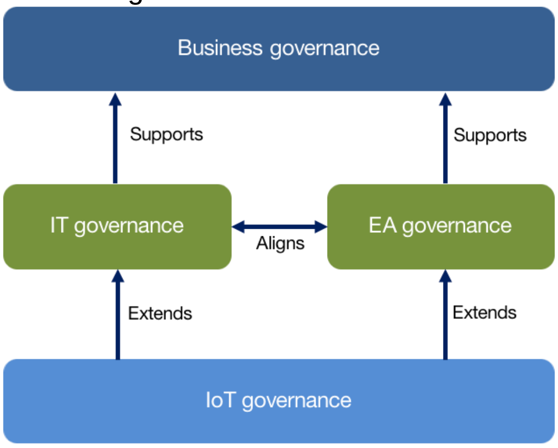

Figure 2 An IoT solution governance model

Successful IoT engagements require that IT organizations define a technical strategy that includes developing a reference architecture, deciding the technology platforms, and developing the processes that are required to design, develop, and operate the IoT solution. Unless teams develop an IoT technical strategy, individual teams across the organization will define their own approaches which often lead to fragmented initiatives that will cost the company more with less chance of success. Figure 3 depicts A technical strategy can include phase-wise activities and clearly defined roles, responsibilities, and deliverables.

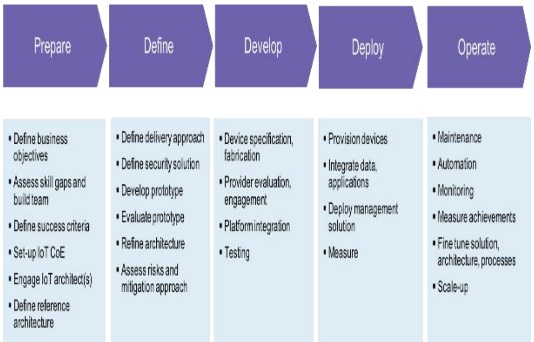

Figure 3. Phases, roles, and deliverables of an IoT technical strategy

To ensure consistency across multiple IoT projects, IoT solutions should adopt a repeatable framework and develop a standard reference architecture that guides individual IoT implementations. Each project must not define their unique way of integrating devices, or communicating with the IoT platform. The IoT reference architecture must meet the needs of different organizational units and define technology standards for all IoT projects to use.

An IoT reference architecture provides a set of architectural patterns, standards, and best practices for use in developing IoT solutions. Use of the approved architectural artifacts from the IoT reference architecture will reduce project risk and lower costs, by reducing the number and complexity of design activities in the project. Your organization's IoT reference architecture can be based on standard IoT reference architectures or industry reference architectures.

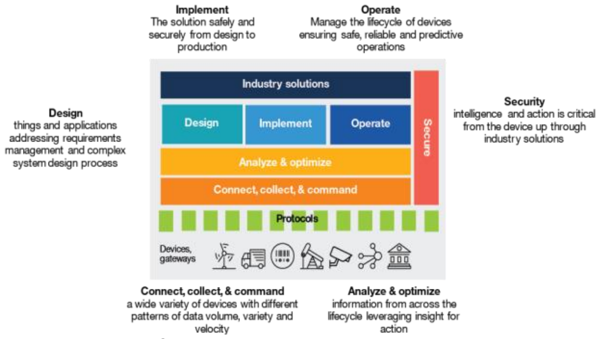

Figure 4. IoT Ecosystem

An IoT ecosystem needs to connect to all types of devices and collect and store data securely. A complete IoT solution needs to include all components of the ecosystem – including devices, network, software, services, security of the complete solution. IoT reference architecture must consider all aspects of the IoT ecosystem as seen in Figure 4 above

Data is generated from devices, and insights from that data are consumed by users or automated operations. Real-time data and near real-time analysis enables timely actions. The type of industry and the nature of the data drives the outcome and selection of a reference architecture. However, collection and storage of data that comes from devices is just the initial step. The value of IoT solutions can be improved by adding more analysis and optimization capabilities. IoT reference architecture must address these more advanced capabilities. To have a great IoT solution would be worthless unless it is secure. All layers of the IoT solution must be protected from vulnerabilities and potential attacks. IoT reference architecture can help ensure that security is not an afterthought.

Things to consider in application layer are 1) Management of the collection, processing, analyzing, and persisting of the large volume of sensor data in near real time, 2) Supporting to the very high data rate, which is much higher than general IT infrastructure, 3) Implementation of predictive analytics capabilities, and also 4)security factors, such as data security, role-based data access, and control functions.

Next to consider in platform layer are 1) Providing for sensor data management, application integration, and device management, 2)Supporting internet-scale messaging, including data collection, publish/subscribe, data mediation, data dispatching, and of course security management, 3) Address security. In communication layer there are 1)Providing a reliable network for capturing and controlling sensor data, 2)Supporting for reliably transporting data from devices to the IoT platform, and 3) Addressing security factor. Lastly, things to considers in physical device layer are 1) Supporting the wide variety of sensors, devices, and gateways2) Supporting remote monitoring and management, 3)Addressing security, such as secured booting, firmware upgrades, intrusion detection, and logging of security events

According to Gartner, Out of that 8.4 billion devices, more than half will be consumer products like smart TVs and smart speakers. The most-used enterprise IoT devices will be smart electric meters and commercial security cameras, figure 5 depicts its details.

Category	2016	2017	2018	2020
Consumer	3,963.0	5,244.3	7,036.3	12,863.0
Business: Cross-Industry	1,102.1	1,501.0	2,132.6	4,381.4
Business: Vertical-Specific	1,316.6	1,635.4	2,027.7	3,171.0
Grand Total	**6,381.8**	**8,380.6**	**11,196.6**	**20,415.4**

Source: Gartner (January 2017)

Figure 5. Significant numbers the increase of IoT use

Interoperability, or the ability of computer systems or software to exchange or make use of information, is a requirement of all devices participating in today's information economy. Traditionally, interoperability has been defined mostly in the context of network communications. But with millions of devices being connected in industries ranging from smart home and building automation to smart energy and retail to healthcare and transportation, a broader definition is now required that considers the cross-domain impact of interoperability on system-to-system performance.

Where Layers 1 through 4 of the OSI model provide a suite of agnostic Internet Protocol-based (IP-based) network infrastructure technologies, syntactic and semantic interoperability often rely on industry-specific formats and protocols optimized based on the type of systems and data at hand. This fact is further complicated by billions of dollars of investment in existing network infrastructure to support M2M communications in these vertical markets. To facilitate broad syntactic interoperability amidst these circumstances, the Industrial Internet Consortium (IIC) recently published the "Industrial Internet Connectivity Framework," or IICF[3]. The IICF redefines the traditional OSI model by combining the Presentation and Session layers (Layers 5 and 6) to provide all of the necessary mechanisms to "facilitate how data is unambiguously structured and parsed by the endpoints" as depicted in figure 2.5. Cross-industry syntactic interoperability is supported by a set of "core connectivity standards" (currently the Data Distribution Service (DDS), OPC-Unified Architecture (OPC-UA), oneM2M, and web services) that communicate through a proposed set of standardized gateways.

Figure 6 depicts Industry use cases benefitting from semantic interoperability

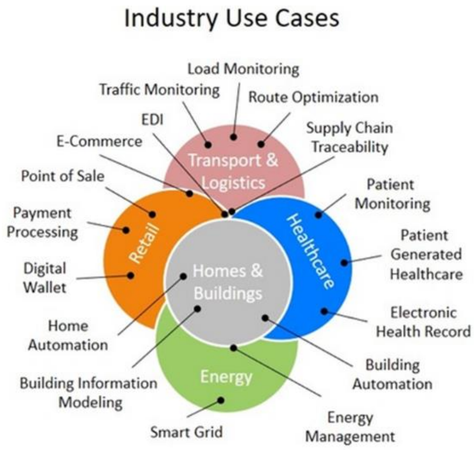

Figure 6. Industry use cases

According to [10] , the choice of a specific sensor is primarily a function of the signal to be measured (for example, position versus motion sensors). There are, however, several generic factors that determine the suitability of a sensor for a specific application. These include, but are not limited to, the following:

- Accuracy: A measure of how precisely a sensor reports the signal. For example, when the water content is 52 percent, a sensor that Reports 52.1 percent is more accurate than one that reports it as 51.5 percent.
- Repeatability: A sensor's performance in consistently reporting the same response when subjected to the same input under constant environmental conditions.
- Range: The band of input signals within which a sensor can perform accurately. Input signals beyond the range lead to inaccurate output signals and potential damage to sensors.
- Noise: The fluctuations in the output signal resulting from the sensor or the external environment.
- Resolution: The smallest incremental change in the input signal that the sensor requires to sense and report a change in the output signal.
- Selectivity: The sensor's ability to selectively sense and report a signal. An example of selectivity is an oxygen sensor's ability to sense only the O component despite the presence of other gases.

3. Conceptual Frameworks and Methods

According to background theories, a conceptual framework for implementing IoT remote quadcopter is based on figure 6 as follow :

Figure 6. Conceptual Framework

The initial phase, prepare, involves defining the business objective, success criteria, and engaging IoT architect(s) . Main step in this phase is to define the objective of the research which is to build a remote quadcopter using microcontroller , in this case it is an Arduino UNO microcontroller. Next is to define the success criteria for the research. Since this research is part of an IoT project , engaging IoT architectures is a must.

The second phase is define, which involves defining delivery approach and security solution, then developing and evaluating prototype, also refining architecture. Delivery approach for components in this equipment is an important aspect. This enable collaboration between all parts involved. To complement the system, it should be capable to offer the fail-safe state for the quadcopter as security solution.

In this research, for limited time availability reason , the next three phases , develop, deploy and operate , are skipped, so only two out of five phases being conducted. Develop phase requires larger scale production for the equipment. The next phase after develop is deploy. This deploy phase can only be done should only the develop phase accomplished. The last phase, operate, may run well after completion of the deploy phase. This research uses life of Internet of Things Methodology (IOTM) approach. IOTM diagram is depicted in figure 7.

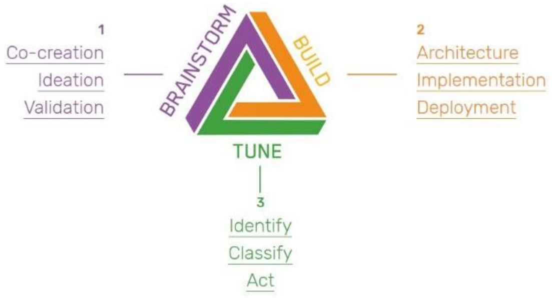

Figure 7,. IoT Methodology

This methodology consists of three processes, 1) Brainstorm, 2)Build and 3) Tune. Brainstorm phase includes co-creation, ideation and validation. In Co-creation there is identification the real problems and finding new ways of working. This also pinpoint some issues in a way can be more understood. Next is Ideate. In a quick and systematic approach, digital and physical mockups are created to show the potential of the best ideas. Using a short list of requirements the goal is to take apart 'fluffy' ideas, examine from different angles, and put them back together with as a clearer, more realistic puzzle-picture/point of view. Last process in Brainstorm is Validate. Validation provides the tools to evaluate your options, avoid the risky routes and put confidence in your team and direction.

Second phase is Build. It consists of another three processes which are 1) building architecture, 2) implementation and 3) deployment. This research includes two out of three processes to be completed, inline with conceptual frameworks the skipping away the deployment process. In the first process, building architecture, proper architectural design in IoT provides a preview of solutions before putting it to work. It is also highlighting risks early and uncovering opportunities to create sustainable development processes to ensure successful ending. The architecture is made simple to avoid over-design because an over-design architecture will sink projects due to such as engineering aesthetics or a complex specification.

The last phase, Tune, involves identifying, classifying and acting. For businesses that rely on the deployed product, a system failure can lead to financial loss. Thus identifying potential problems and preventing failures is an essential step in every project.

4. Results

4.1 Architecture Design

The following is design of quadcopter through architecture design of remote quadcopter and the selection of the suitable components building it. Refering to Internet of Things Methodology (IOTM), it is part of the the second phase, Building. This phase consists of three parts, architecture, implementation and deployment. Architecture is the continuation of the previous step, ideation.

Figure 8. below depicts the design of the quadcopter connection. The quadcopter is built using the Arduino UNO as the main controller of the other parts such as the motors and Electronic Speed Controllers (ESCs) . It may be connected in two ways, either using WiFi connection or through cloud connection. The instructions for the Arduino are stored in microcontroller through its USB port connection.

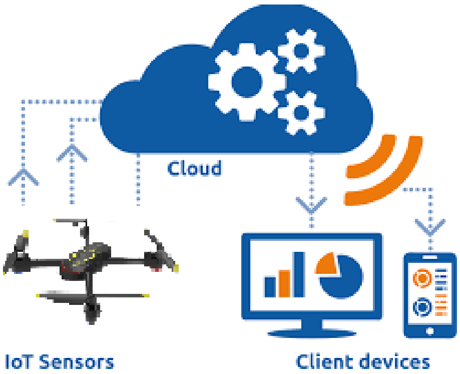

Figure 8. Design of IoT Quadcopter connection

Next, the design of the quadcopter is discussed. A quadcopter is an under actuated aircraft with fixed pitch angle four rotors as shown in Figure 9. The aim is to develop a model of the vehicle as realistically as possible. A typical quadcopter has four rotors with fixed angles and therefore quadcopter has four input forces, which are basically the thrust provided by each propellers as shown in Figure 9.

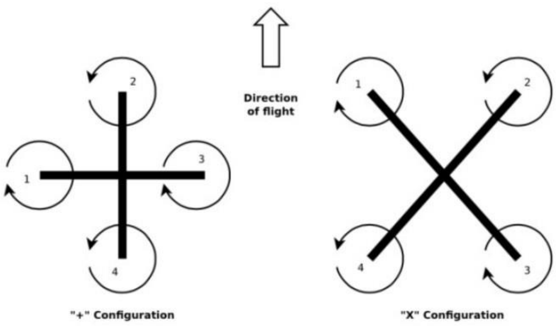

Figure 9. Two main types of quadcopter configuration

There are two possible configurations for most of quadcopter designs “+” and “×”. An X-configuration quadcopter is considered to be more stable compared to + configuration, which is a more acrobatic configuration. Propellers 1 and 3 rotates counter clockwise (CW) , 2 and 4 rotates counter-counter clockwise (CCW). So that, the quadcopter can maintain forward (backward) motion by increasing (decreasing) speed of front (rear) rotors speed while decreasing (increasing) rear (front) rotor speed simultaneously, which means changing the pitch angle. This process is required to compensate the action/reaction effect (Third Newton’s Law). Propeller 1 and 3 have opposite pitch with respect to 2 and 4, so all thrusts have the same direction.

Figure 10. depicts the list of the required materials for building the quadcopter. Frame is the component for placing all other supporting parts. The component for the flight controller is the Arduino UNO. There are four motors and ESCs. Let's begin with the frame of your quadcopter. The frame provides the structure and rigidity and it's where all the components will be mounted. The flight controller manipulates the RPM of the individual motors in response to the user's input. Flight controller will adjust the RPM of the rear motors. The ESCs take the signals from the flight controller and adjust the speed of the motors. The motors and propellers provide the thrust and the lift for your DIY quadcopter. The battery provides the power to all the components. The radio (transmitter and receiver) allows you to control the movements of the quadcopter.

No	Components
1	Frame
2	Flight Controller
3	Motors
4	ESCs (Electronic Speed Controllers)
5	Propellers
6	Battery
7	Transmitter
8	Receiver
9	LiPo Battery charger
10	Low voltage battery alarm

Figure 10. Component listing of the quadcopter materials

Figure 11 depicts NODEMCU_DEVKIT_V1.0_PINMAP. This module has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode. By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, it consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

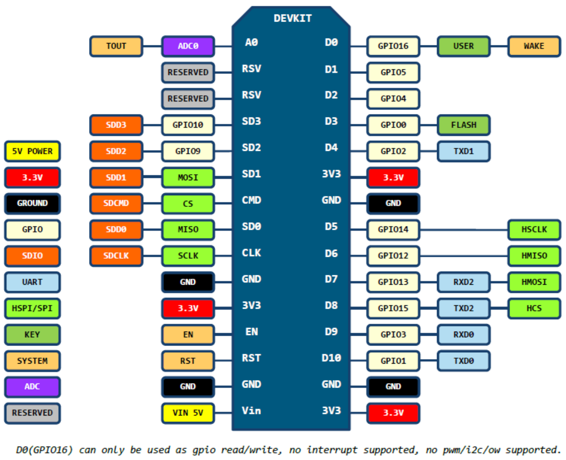

Figure11. NODEMCU_ _PINMAP

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake it up at any required interval. It can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed. In order to satisfy the power demand of mobile and wearable electronics, it also can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption. Figure 12 depicts the illustration of module power management.

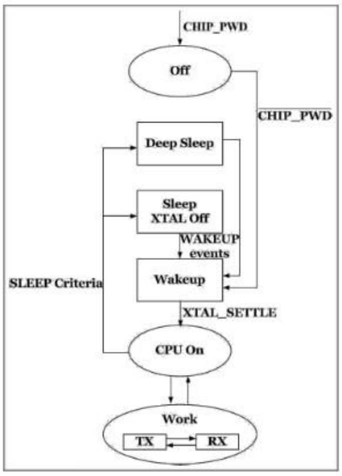

Figure 12. ESP8266 Power Management

The project architecture can be modeled with a top level diagram that shows how all the components of the architecture are connected. This includes the network, USB, motor, and ESC connections. Because this is the general architecture, it does not cover the details of how the applications specifically operates, although the hardware and software sections cover this thoroughly. The diagram shown in Figure 13 displays how everything is connected. The PC application can communicate with the Android phone over TCP and vice versa. The Android phone simultaneously communicates with both the PC application in TCP and the Arduino microcontroller over USB UART for sending and receiving data. Furthermore, the Arduino has one unidirectional communication via PWM with the ESC modules, and those in turn control the motor speed. The ESCs draw power from a lithium ion battery to supply the motors with three-phase AC power.

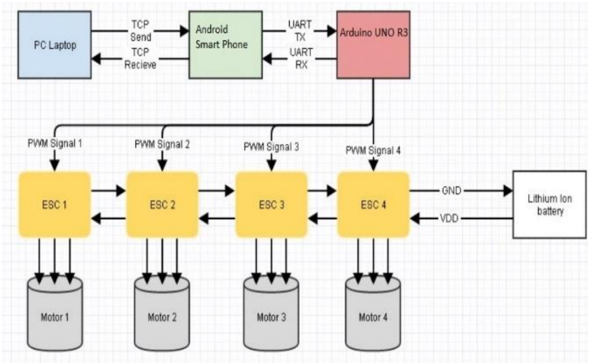

Figure 13 General Architecture Design

Moreover, in order to control the motors from the PC application, a data packet must traverse over WI-FI to the Android phone, then passed to USB UART, and then finally sent as a modification of the PWM signal for a particular motor. This setup requires that a WI-FI network already exist and implies that data, such as a live video signal, can traverse back over the TCP connection to the PC. The hardware diagrams for the project includes a diagram of the Arduino's connects as well as the layout for the ESCs. In Figure 14 below, the hardware connections and pins are labelled specifically.

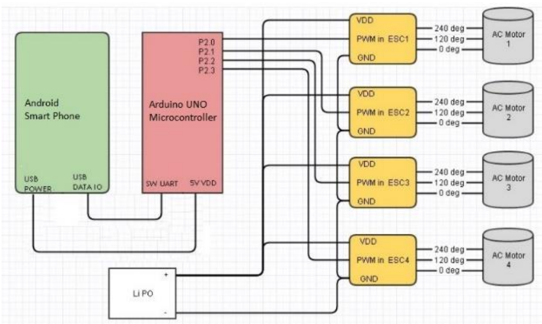

Figure 14 Hardware architecture

The hardware pin diagram resembles the system architecture diagram, but shows the specific pin connections between the Arduino, the Android phone, the ESCs, and the three phase AC motors. The ESCs shown on the pin diagram, however, are much more complicated and encapsulate a significant amount of logic. The key purpose of the ESCs, as was indicated by research, is to convert the PWM signal, in an efficient way, to a three phase power output that can also protect against current spikes and regulate voltages in a safe way; in this case, safety refers to protecting the quadcopter against crashing in flight and thus protecting people. Rapid changes in voltages could cause the quadcopter to lose stability. The spikes could be caused by changes in battery temperature or a fast change in motor speeds. Furthermore, the AC motors spin dangerously fast, and so protecting against spikes, voltage changes, and fluctuations is important for safety purposes, especially if the quadcopter is in flight. There are two types of ESC as shown in figure 15 , Brushed ESC and Brushless ESC.

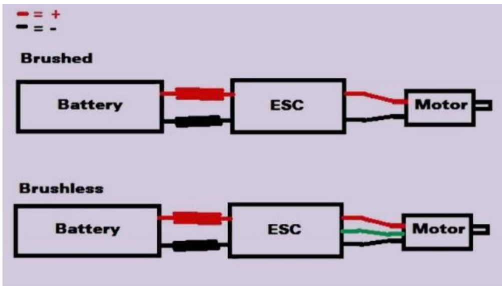

Figure 15. Two types of ESC

Brushed ESC is the first electronic speed controller, It is cheap to use in various RTR electric RC vehicles. Brushless ESC is the modern advancement in technology once it comes to Electronic Speed Controls. It is also a bit more costly. Connected to a brushless motor, it carries more power higher performance as compared to the brushed ones. It can also last a longer period.

4.2 Implementation

Implementation of new and complex IoT concepts should be viewed until it becomes a scientific. Concepts may vary from simple new interactive UI, automated controller, remote monitoring or fully integrated digital twins, the implementation can may concern entirely bespoke or off the shelf solutions. Regarding how the technology is built the challenge comes from the digital and cultural changes of implementing concepts into new environments. In this research , the chosen quadcopter structure is one as "+" configuration. Thus there are two different propellers used in this component building, Figure 16 depicts the complete build of the quadcopter.

Figure 16. Quadcopter with propeller type 1

Figure 17. Quadcopter with propeller type 2

The micro controller ARDUNO UNO is programmed using C language with ARDUINO IDE software. It is a development environment that simply uses an user interface for manipulating coding language. These program are utilized in various calibration steps which includes some calibration. Set up calibration indicates the interconnections of various hardware components used in quadcopter. Initially program is uploaded to Arduino board using IDE Software . Other important calibration is the ESC's calibration.

ESC calibration varies with the brand of ESCs used in quadcopter. The calibration of ESIC is done on priority basis with the help of a radio system for each rotor and corresponding ESC. It includes the following steps as follows:

- Upload the program on controller board, then turn ON the transmitter and put the throttle stick to its maximum.
- Connect the batteries .The auto pilot's red, blue and yellow LED will light up in cyclic pattern that indicates ESCs are ready for calibration mode.
- By keeping transmitter throttle stick high, disconnect and then reconnect the battery.
- Regular no of beeps on transmitter indicates the battery cell count and additional two beeps specify that maximum throttle has been captured.
- Set the transmitter throttle stick down to its minimum position.
- ESCs should now emit a long tone that indicates minimum throttle has been captured and calibration is complete.

ESCs regulate voltage and help to keep the motors running more consistently despite the voltage drop as the batteries become more and more depleted. The other important consideration was whether using a PWM signal hooked up to a BJT would actually work; every source online strongly recommended using an ESC, and never had any examples of anyone using transistors instead of BJTs. Therefore, the best decision was to utilize the ESC modules.

The design decision of choosing the quadcopter kit had several advantages and disadvantages. However, everything was compatible mechanically and the drone chassis was also light weight. Furthermore, the propellers, although somewhat durable, would basically break if they hit anything at high velocity. Consequentially, at least 2 propellers needed to be replaced over research observation because the blades hit something while rotating rapidly. A major improvement would be some kind of circular guard to protect the blades from material coming from the sides. Ideally, it also would have been nice to create a 3D printed chassis for the microcontroller for aesthetic, structural integrity, and aerodynamics purposes. Figure 18 depicts evaluation of the motor operation

Experiment	Procedure	Pass Condition
1	Drive Motor 1 to FULL power	PASS
2	Drive Motor 2 to FULL power	PASS
3	Drive Motor 3 to FULL power	PASS
4	Drive Motor 4 to FULL power	PASS
5	Increase motor 1 and motor 2 speed equally	PASS
6	Increase motor 1 and motor 3 speed equally	PASS
7	Increase motor 2 and motor 3 speed equally	FAIL
8	Increase motor 1 and motor 4 speed equally	PASS

Figure 18. evaluation of the motor operation

5. Conclusions

This research work yielded a development of Quadcopter based-on Arduino Uno as the microcontroller. Quadcopter can be easily made. Design and Implementation Internet of Things (IoT) Android Remote-Controlled Quadcopter has been successful as a low cost alternative to various applications

References

[1] Atzori L. et.al , The Internet of Things: A survey, Computer Network, Volume 54, Issue 15, 28 October 2010, Pages 2787-2805, Elsevier, 2010

[2] Gubbi J. et al. , Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems Volume 29, Issue 7, September 2013, Pages 1645-1660, Elsevier, 2013

[3] Lee, The Internet of Things (IoT): Applications, investments, and challenges for enterprises, Business Horizons, Volume 58, Issue 4, July–August 2015, Pages 431-440, Elsevier, 2015

[4] Smith A.G, 2011, Introduction to Arduino : a piece of cake , SmithPress, 2011

[5] Chaouchi, H.; Bourgeau, T. , Internet of Things: Building the New Digital Society. IoT 2018, 1, 1-4., 2018

[6] Gantait A., et. al., Defining your IoT governance practices: How to manage enterprise-wide IoT initiatives, IBM , 2018

[7] John P. “Describing the Meaning of Data—An Introduction to Data Semantics

and Tagging", Project Haystack Connections, Issue 1, Spring., 2016

[8] Holdowsky J. et.al, Inside the Internet of Things (IoT) : A primer on the technologies building the IoT. Deloitte University Press, 2015

[9] IEEE, "IEEE P1451.6 terms and definitions," http://grouper.ieee.org/ groups/1451/6/TermsDefinitions. htm#Sensor, accessed July 19, 2018.

[10] Fraden J. , Handbook of Modern Sensors:Physics, Designs, and Applications, Fourth Edition (Springer, April 2010), p. 13, 2010

[11] Kalantarzadeh et. al. ,"Sensor characteristics and physical effects," Nanotechnology-Enabled Sensors (Springer,), p. 13., 2008

[12] Thua K.M., Gavrilov A.I. , Designing and modeling of quadcopter control system using L1 adaptive control, Elsevier, 2017