Communication between two Unmanned Aerial Systems

Shivkumar S
Dept. of Electronics and Communication
Amrita School of Engineering
Bangalore,India
shivkumarsankaralingam@gmail.com

Prajwal K.S.
Dept. of Electrical and Electronics
Amrita School of Engineering
Bangalore,India
prajwal99ks@ieee.org

Palanki Amitasree
Dept. of Electrical and Electronics
Amrita School of Engineering
Bangalore,India
amitasreenasa@gmail.com

Palli Ashish
Dept. of Electronics and Communcation
Amrita School of Engineering
Bangalore,India
palliashish2019@gmail.com

K Tejaswini
Dept. of Electronics and Communication
Amrita School of Engineering
Bangalore, India
tejaswinikuree@gmail.com

Nippun Kumar A.A.
Dept. of Computer Science Engineering
Amrita School of Engineering
Bangalore,India
aa_nippunkumaar@blr.amrita.edu

***Abstract* — The use of flying platforms such as unmanned aerial vehicles (UAVs), popularly known as drones, is rapidly growing. In particular, with their inherent attributes such as mobility, flexibility, and adaptive altitude, UAVs admit several key potential applications in wireless systems. Usually, UAV-UAV communication has the potential to create an ad-hoc network and will reduce the tedious hops from source to destination. In this paper, we engineer a working solution for establishing communication between two Tello Drones using ESP8266 microcontrollers with Node MCU as its firmware. The objective is to demonstrate the use of UAV-to-UAV communication and coordination in realistic, operational flight environment. The communication protocols involved in this design are IEEE 802.11 and ESPNOW. The drones will be able to exchange data with each other and this can also be monitored from the base station.**

Keywords—Tello drone, ESP8266, Wi-Fi, Communication

I. Introduction

Unmanned Aerial Vehicles or drones are airborne vehicles which are autonomous or remotely controlled by humans. The drone technology is developing at a very fast pace and are used in various fields such as Disaster Management, Agriculture, Military, Surveillance, Transport etc. The development of communication between the unmanned aerial vehicles has sought utmost significance in the current era, especially in the fields listed above. In this paper two Tello drones will be used to demonstrate UAV-UAV Communication by exchanging their Battery Values, Temperature Values and their Speed while the drones are in motion. The drones will communicate with ESP8266 Microcontrollers. This data will also be received by the base station. This will be helpful for both drones to know about each other very well and coordinate together in various circumstances. The Information or Data such as Height Acceleration and Barometer Values of Tello Drones can also be transmitted.

II. System Architecture

A. Ryze DJI Tello Drone

Ryze DJI Tello drone is a small quadcopter that contains a flight controller, Video downlink system, Vision Positioning System, Propulsion system, and a Flight Battery. It has an onboard camera that captures 5 megapixel photos and streams 720p live video to the Tello app on a mobile device. Its maximum flight time is approximately 13 minutes, and its maximum flight distance is 328 ft (100 m). Failsafe Protection enables Tello to land safely even if you lose connection and its propeller guards can be used to enhance safety.

B. ESP8266 Microcontroller

The ESP8266 is a low-cost, 32-bit Wi-Fi microcontroller. The SDK used to program this microchip is Node MCU which is a Lua-based firmware.

C. Lithium Ion Batteries

The Lithium Ion Batteries of 3.7 V and 650 mah is used to supply voltage. These are light weight batteries.

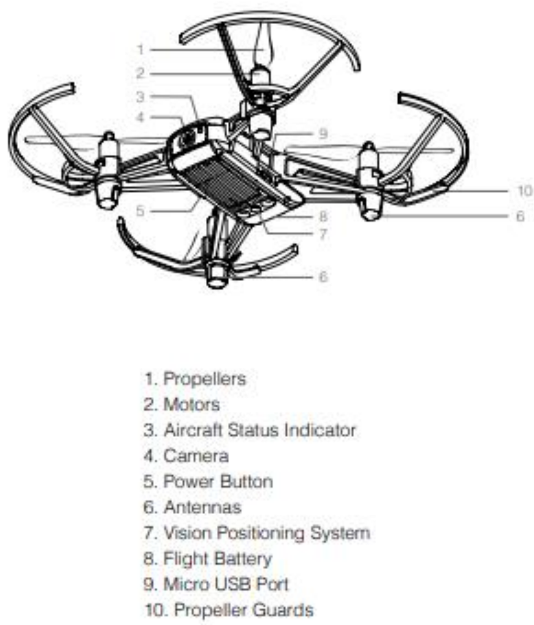


Figure 1: The architecture of the Drone

III. Design Model

The design model is to establish communication between two Tello Drones using ESP8266 microcontrollers with the help of ESPNOW protocol. Two ESP8266 microcontrollers will be mounted on top of the drones. The base station which only receives data will also be an ESP8266 microcontroller which will be connected with the other two microcontrollers using ESPNOW protocol.

Tello Drones will be connected to the Microcontrollers using Wi-Fi. The Drones will receive instructions or commands from the Nodes which are pre-programmed using Arduino IDE. The Drones will also be able to send values to other Nodes which will then be received by the base station. The data received by the base station can be seen through the Serial Monitor.

Two Lithium Ion Batteries of 3.7 V and 650 mah are used to supply voltage to the Microcontrollers which will be mounted on top of the Drones.

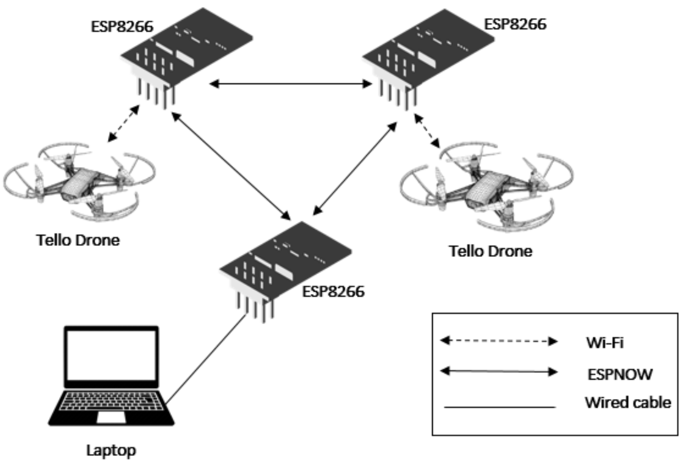


Figure 2: The physical System of the Design

IV. Evaluation and Result

The drones were tested outdoors. The communication was established between the drones and the base station as given in Figure 2. The drones were programmed to send Battery Value, Temperature Value and their Speed to each other and the base station while they are flying.

Figure 3: Drones at rest

Figure 4: Drones at motion

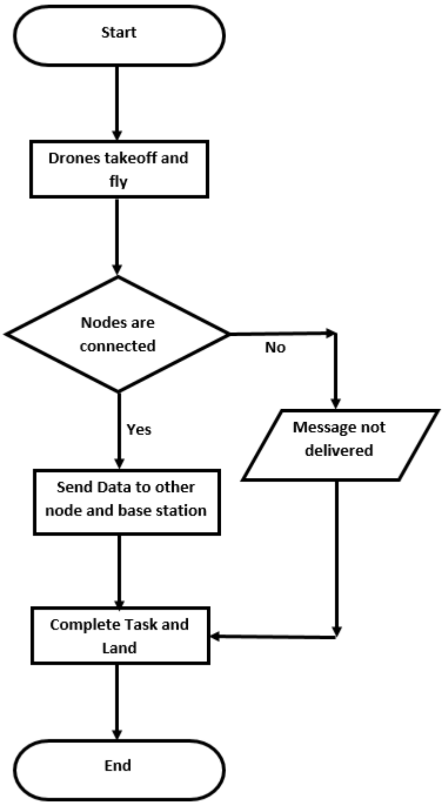


Figure 5: Flowchart on the working algorithm

According to Figure the given flowchart gives the sequential steps in which the tasks and communication are executed. As soon as the Drones are switched on, they takeoff and start flying to perform their tasks. At the same time if the Nodes are connected then the Data that is the Battery Value, Temperature Value and the Drones Speed are sent to the other Node and the Base station. If the Nodes fail to connect the message is not delivered. After the drones complete their tasks. They immediately land. The communication continues till the drones are switched off.

For reference let us say we have Tello Drone-1 and Tello Drone-2 on which ESP8266-1 and ESP8266-2 Microcontrollers are mounted respectively. ESP8266-base station receives values from both the drones.

Before we mount ESP8266-1 and ESP8266-2 Microcontrollers on the drones we first verify whether they are communicating with each other.

We connect ESP8266-1 to the laptop to check whether the microcontroller is sending and receiving data. After we switch on the drones and open the Serial Monitor window we will be shown with the packet delivery status. The result can be verified from Figure 6.

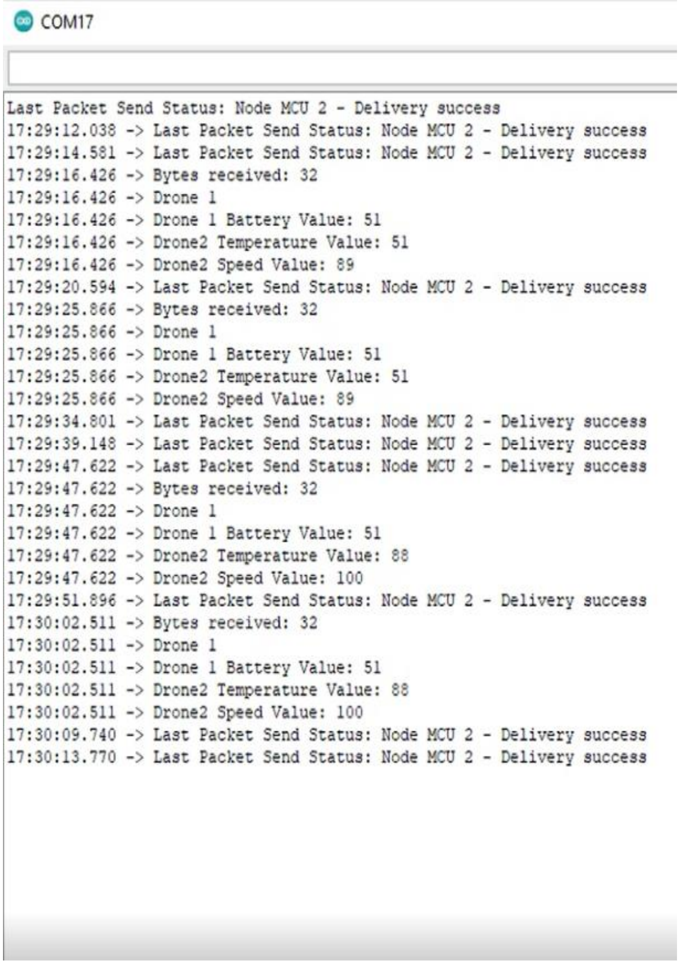
We then connect ESP8266-2 to the laptop to check whether the microcontroller is sending and receiving data. After we switch on the drones and open the Serial Monitor window we will be shown with the packet delivery status. The result can be verified from Figure7.

Now we mount ESP8266-1 and ESP8266-2 Microcontrollers on Tello Drone-1 and Tello Drone-2 respectively.

```
COM17

Last Packet Send Status: Node MCU 1 - Delivery success
17:21:33.464 -> Bytes received: 32
17:21:33.464 -> Drone 2
17:21:33.464 -> Drone2 Battery Value: 73
17:21:33.464 -> Drone2 Temperature Value: 84
17:21:33.464 -> Drone2 Speed Value: 100
17:21:37.212 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:21:37.626 -> Bytes received: 32
17:21:37.626 -> Drone 2
17:21:37.626 -> Drone2 Battery Value: 73
17:21:37.626 -> Drone2 Temperature Value: 84
17:21:37.626 -> Drone2 Speed Value: 100
17:21:46.898 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:21:51.931 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:09.718 -> Bytes received: 32
17:22:09.718 -> Drone 2
17:22:09.718 -> Drone2 Battery Value: 70
17:22:09.718 -> Drone2 Temperature Value: 86
17:22:09.718 -> Drone2 Speed Value: 100
17:22:10.653 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:13.916 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:23.573 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:28.575 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:44.210 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:22:44.559 -> Bytes received: 32
17:22:44.559 -> Drone 2
17:22:44.559 -> Drone2 Battery Value: 66
17:22:44.559 -> Drone2 Temperature Value: 86
17:22:44.559 -> Drone2 Speed Value: 100
17:22:49.880 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:23:00.753 -> Last Packet Send Status: Node MCU 1 - Delivery success
17:23:03.254 -> Bytes received: 32
17:23:03.254 -> Drone 2
17:23:03.254 -> Drone2 Battery Value: 62
17:23:03.254 -> Drone2 Temperature Value: 86
17:23:03.254 -> Drone2 Speed Value: 100
17:23:05.994 -> Last Packet Send Status: Node MCU 1 - Delivery success
```

Figure 6: Serial Monitor of ESP8266-1



```
COM17

Last Packet Send Status: Node MCU 2 - Delivery success
17:29:12.038 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:14.581 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:16.426 -> Bytes received: 32
17:29:16.426 -> Drone 1
17:29:16.426 -> Drone 1 Battery Value: 51
17:29:16.426 -> Drone2 Temperature Value: 51
17:29:16.426 -> Drone2 Speed Value: 89
17:29:20.594 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:25.866 -> Bytes received: 32
17:29:25.866 -> Drone 1
17:29:25.866 -> Drone 1 Battery Value: 51
17:29:25.866 -> Drone2 Temperature Value: 51
17:29:25.866 -> Drone2 Speed Value: 89
17:29:34.801 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:39.148 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:47.622 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:29:47.622 -> Bytes received: 32
17:29:47.622 -> Drone 1
17:29:47.622 -> Drone 1 Battery Value: 51
17:29:47.622 -> Drone2 Temperature Value: 88
17:29:47.622 -> Drone2 Speed Value: 100
17:29:51.896 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:30:02.511 -> Bytes received: 32
17:30:02.511 -> Drone 1
17:30:02.511 -> Drone 1 Battery Value: 51
17:30:02.511 -> Drone2 Temperature Value: 88
17:30:02.511 -> Drone2 Speed Value: 100
17:30:09.740 -> Last Packet Send Status: Node MCU 2 - Delivery success
17:30:13.770 -> Last Packet Send Status: Node MCU 2 - Delivery success
```

Figure 7: Serial Monitor of ESP8266-2

We connect ESP8266-base station to the laptop and switch on the drones. The drones are able to communicate with each other and the base station by sending and receiving Battery Values, Temperature Values, and their Speed. The result can be verified from Figure 8.

```
COM17

20:28:04.614 -> Speed Values : 100
20:28:08.466 -> Bytes received: 32
20:28:08.466 -> Drone 2
20:28:08.466 -> Battery Values : 100
20:28:08.466 -> Temperature Values : 57
20:28:08.466 -> Speed Values : 100
20:28:22.321 -> Bytes received: 32
20:28:22.321 -> Drone 2
20:28:22.321 -> Battery Values : 100
20:28:22.321 -> Temperature Values : 57
20:28:22.321 -> Speed Values : 100
20:28:26.552 -> Bytes received: 32
20:28:26.552 -> Drone 1
20:28:26.552 -> Battery Values : 100
20:28:26.552 -> Temperature Values : 68
20:28:26.552 -> Speed Values : 100
20:28:41.230 -> Bytes received: 32
20:28:41.230 -> Drone 1
20:28:41.230 -> Battery Values : 100
20:28:41.230 -> Temperature Values : 68
20:28:41.230 -> Speed Values : 100
20:28:42.483 -> Bytes received: 32
20:28:42.483 -> Drone 2
20:28:42.483 -> Battery Values : 100
20:28:42.483 -> Temperature Values : 67
20:28:42.483 -> Speed Values : 100
20:28:52.162 -> Bytes received: 32
20:28:52.162 -> Drone 2
20:28:52.162 -> Battery Values : 100
20:28:52.162 -> Temperature Values : 67
20:28:52.162 -> Speed Values : 100
20:28:58.512 -> Bytes received: 32
```

Figure 8: Serial Monitor of ESP8266-base station

V. Scope of Improvement

This design with the help of Tello drones is just a small prototype to demonstrate UAV-UAV Communication. This can be improved by using bigger and developed drones with the same procedure for performing various applications based on disaster management. The number of drones can also be increased. The data or information may not be restricted to only Battery, Temperature, Speed, Height, Acceleration, Barometer and Time of flight of the Drones but also Images and Live Video Stream which will be beneficial to identify Human beings and other living creatures during floods and various other natural calamities,.

VI. Conclusion

We therefore conclude that UAV-UAV communication between the drones has been established but more protocols and better drones can be designed to achieve more responsive and better communication for unmanned aerial vehicles which thereby can assist us in many applications and processes.

REFERENCES

[1] randomnerdtutorials.com/esp-now-esp8266-nodemcu- arduino-ide/
[2] github.com/SweiLz/NodeMCU-TelloDrone