ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 6, No. 2, 1091-1100 (2021)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

Enhanced Data Transportation in Remote Locations Using UAV Aided Edge Computing

Niranjan Ravi*, Mohamed El-Sharkawy

Indiana University Purdue University Indianapolis, Department of Electrical and Computer Engineering, Institute, Indianapolis, IN, 46202, USA

ARTICLE INFO

Article history:
Received: 16 February, 2021
Accepted: 18 March, 2021
Online: 28 April, 2021

Keywords:
Unmanned Aerial Systems (UAV)
IoT
UAVCAN
ECC
IBM Cloud
Thread Mesh Network
Ble Mesh

ABSTRACT

In recent years, the applications in the field of Unmanned Aerial Vehicle (UAV) systems has procured research interests among various communities. One of the primary factors being, thinking beyond the box of what could UAV system bring to the table other than military applications? Evidence to any answer for this question is the current day scenarios. We could see numerous applications of UAV starting from commercial applications of delivering consumer goods to life saving medical applications such as delievery of medical products. Using UAVs in for data transportation in remote locations or locations with no internet is a trivial challenge. In-order to perform the tasks and satisfy the requirement, the UAVs should be equipped with sensors and transmitters. Addition of hardware devices increases the number of connections in hardware design, leading to exposure during flight operation. This research proposes an advanced UAV system enabling wireless data transfer ability and secure data transmission with reduced wiring in comparison to a traditional design of UAV. The applications of this research idea targets using edge computing devices to acquire data in areas where internet connectivity is poor and regions where secured data transmission can be used along with UAV system for secure data transport.

1 Introduction

In current day scenarios, internet is playing a significant role in everyone's daily life. There has been a expeditious technological developments in various fields for betterment of human life. One of such developments is discovering various new modes of data exchange like heterogeneous communication between humans and things compared to a long standing traditional homogeneous communication. Wireless Sensor Networks (WSN) has amplified the applications of embedded systems and Internet of Things (IoT). WSN, a network of randomly dispersed nodes which can have communication with every other node in the network with help of radio signals.

WSN is made up of independent micro-sensor nodes which can sense and communicate the information further to other nodes. A single micro-sensor node can be embedded with a sensor, a microcontroller or a microprocessor, depending upon the target applications [1], [2]. The micro-sensor nodes comes with an advantage of inbuilt power source and a wireless transmitter to enable wireless data transfer facility both online and offline environments. A classic example of WSN is using them to sense atmospheric conditions like temperature and humidity. The usefulness of WSN is tremendous in industrial sectors with hazardous nature where it is difficult to deploy humans for data acquirement.

If the WSN are enhanced with wireless technologies such as ble, thread and zigbee it would enable them to send the data without the need for internet or physical wired connections. These technologies can operate on coin cell battery for years which makes this WSN a low-power architecture. By deploying of edge devices, wireless mesh networks such as ble mesh, thread mesh network would enable a single node to transfer data for longer distance to avoid any failures and ensure no data is lost during transmission [3]. Owing to above factors, the WSN is used in industrial and non industrial applications. At the same time, there has been an increasing demand to use WSN at remote locations to acquire sensitive data but the problem arises how can the data be transmitted for longer distance when it exceeds the limitations of mesh technologies? On the other had, how can we secure the data when it is being transmitted and make it readily available around the globe at the same time? Is it possible to carry out the operation using IoT edge devices?

Edge computing devices helped us to understand and answer

*Corresponding Author: Niranjan Ravi, ECE Department, IUPUI, IN, USA & ravin@iu.edu

the questions. These unanswered questions triggered the motivation to use UAV to transport the acquired data from remote WSN when it is difficult for any human to enter the region. UAVs also has a significant quality of quicker transportation owing to less air traffic.

To ensure a secure data transmission,a ready to use solution, NXP's A71ch, a secure element which can store provision credentials and aid in connecting to private/public clouds [4] was utilized. This process would require new hardware installation in UAV design. The increment of hardware comes with a cost of increased wiring which is addressed by using light-weight protocol called Unmanned Aerial Vehicle Controller Area Network (UAVCAN).

2 Literature Review

In 1999, when term IoT was first coined, it was believed that in future every object would have a globally unique Radio Frequency Identification tag (RFID). IoT provides a new path to explore the data which has been gathered by the embedded systems using its sensors. IoT provides a guiding hand to devices which are resource constrained and would require additional computational and communication capabilities. They help in monitoring the devices at remote locations through the internet in event of any failures. Embedded system devices used in IoT applications are referred as smart objects because they act according to environment changes. But the question of storage arises. How much data can these devices store? Would it be efficient to store the data in the stand-alone device? Is there any alternative?

Cloud platform answers these questions. Cloud computing plays a vital role in enhancing IoT sector. The smart objects are resource-constrained and performing advanced computational would require re-designing the hardware which would result in increased production and operating costs. Also, it would increase the size of the component. Cloud provides alternate approach like data storage, handling and unlimited distributions. This brings up new features like scalability and flexibility for the researchers to deploy the products in various new applications. By uploading the data to cloud complex machine learning algorithms can be deployed to observe patterns in the data on real-time. This cloud infrastructure solves the problem of storage and real-time data visualization. But how secure is the data transmission to the cloud?

Crytographic solutions answers the above questions by offering end-to-end security to minimize data breach. The idea of choosing a right encryption technique in IoT field is widely debated. Because, the data generated by IoT devices are of vast magnitude ranging from a temperature sensor value to medical records which are exchanged with cloud platforms. A pressing need for security measures that should be compatible enough to be deployed in IoT devices is required.

Elliptic Curve Cryptography (ECC), a type of asymmetric cryptography is focused in this research. Asymmetric cryptography knows as public key cryptography algorithm involves a pair of keys: a public key which can be shared/exchanged and a private key, kept as a secret. This provides higher security level at Transport Layer Security (TLS) for devices which are resource constrained, as shown in this research. TLS handshake is for authentication and key exchange in order to establish a secured connection. TLS protocol version and usage of asymmetric encryption algorithm is managed by TLS handshake protocol.

On the other hand, taking advantage of some of the above developments in the field of sensors and electronics, UAV system has evolved its reach into civilian and military applications [5]. They are currently used in surveillance of a location and even launch drone strike on enemy targets. One of the primary reasons for advancement in the field of UAV systems are

1. Environmental safety and protection.
2. Military drone strikes.
3. Drone surveillance during natural disasters.
4. Geological study.

A flight controller module, GPS for navigation purpose, a telemetry device to transmit and receive data from ground control station are used in design and construction of a traditional UAV. It is also equipped with 9 axis sensors to get real time flight dimensions in motion [6]. A lot of new physical interfaces have been introduced in the flight controller design. One such development is introduction of CAN port to reduce the wiring and reduce the noise. A thorough background study on UAVCAN principles was carried out with help of this research [7] where the author has experimented a proof of concept of establishing a one way CAN bus communication between flight controller with an arduino module.

The basic understanding of arduino was helpful but further questions arose like how would this interface work when flight controller is paired with a resource constrained microcontroller? What are the parameters needed to be considered when integrating various operating systems with different clock frequencies and operating speed? How much of the payload(data) can be transmitted for the usage? How many devices can be connected in a single UAVCAN bus? What are the applications it would bring on this successful completion? How to transmit data from a drone system securely to a data centre/cloud networks ? The results of the question are discussed on this research.

3 Background Study

3.1 802.15.4 Standard

This standard defines an effective Medium Access Control (MAC) layer with numerous physical layers (PHY) which can be optional. IEEE 802.15.4 outlines Information elements (IEs) to help in better utilization of the standard. The header IE and payload IE. The header can be useful in authentication and payload can be encrypted. IE is advantageous such as providing backwards-compatibility for future versions of the standard.

3.2 6LowPAN

6LowPAN is abbreviated as IPv6 over Low Power Wireless Personal Networks. As a successor of IPv4, IPv6 has increased the address size of 128 bits as compared to 32 bits. 6LowPAN transmits and receives IPv6 packets over IEEE 802.15.4 link. Enabling the

maximum frame size to be sent transmitted. It is aimed at low power and low cost applications. In places of Ethernet, a packet of Maximum Transmission Unit (MTU) size can be transmitted in a single frame. Between 802.15.4 link layer and IPv6 networking layer, 6LowPAN exists as an intermediate layer. This aids in the process of fragmentation of sender's packet and re-fragmenting it on the receiver node. The transmission load is reduced when fewer bits of data is transmitted. Thread network inherits this feature to transmit data packets. Mesh header supports more efficient methods of compression of messages inside a thread network .

3.3 Constrained Application Protocol

Most of the resources deployed in the field of IoT are resource-constrained in nature. In order to support the data transmission, a specialized web transfer method/protocol called constrained application protocol (CoAP) is deployed. CoAP contains RESTful features to run on any resource constrained embedded devices. CoAP works on User Datagram Protocol (UDP) protocol than Transfer Control Protocol (TCP). A feature of getting response from destination nodes to parent node called Confirmable requests (CON) is also present.

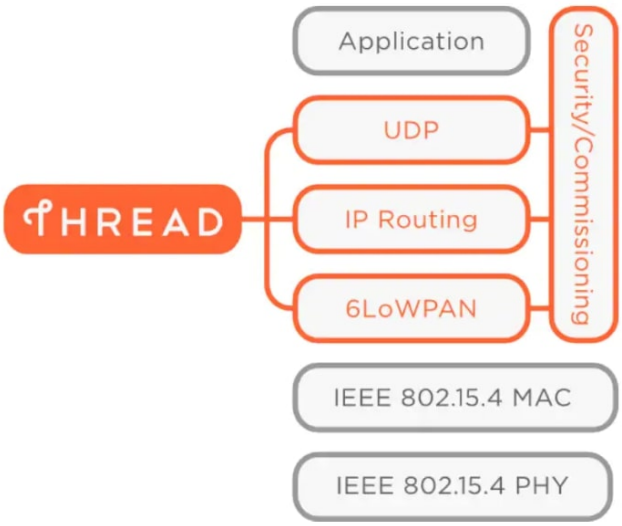

Figure 1: Thread network stack [8]

3.4 Thread Protocol

Thread is an Internet Protocol (IP) based on wireless networking protocol aimed at low powered applications in the field of embedded systems and IoT. Thread is developed based on many standards resulting in providing many advantages like reliability, cost-effective and low power consumption. Thread used UDP on Network layer and uses IEEE 802.15.4 PHY and MAC layers wireless specifications supporting up to 250 devices in a mesh network by operating at 2.4 GHz band as shown in Figure 1. With help of Wi-Fi on their Home Area Network (HAN), users can exchange and communicate with other thread devices within the thread network by using smartphones.

3.4.1 Thread and Border Router

Thread router offer connectivity for the devices inside the network and offers security services to new devices trying to join the network. Border router carries out connectivity between one network and its adjacent. A thread network would consists of at least one border router and it can have many as well depending upon the application. A leader in the thread network would receive requests from end devices to become router. The leader also contains the registry of router IDs in the network. In any instance of the application, if the leader fails, another router is elected to be the leader immediately without any user intervention .

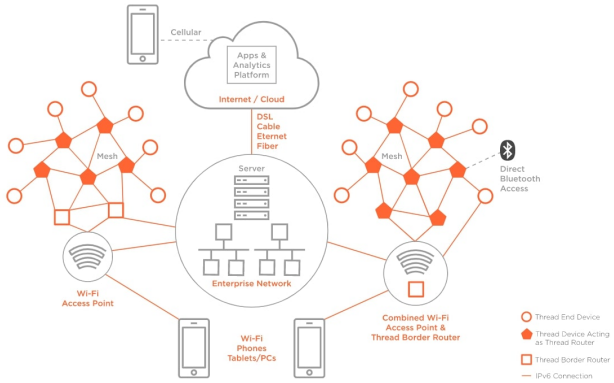

Figure 2: Commercial thread network topology

3.4.2 Router Eligible End Devices (REEDS)

As the name suggest REEDs are devices which primarily acts as end devices. The REEDs are eligible to become active routers. This action is carried out by the network topology without any user intervention. The REEDs can also act as sleepy end devices which has the function to gather data and transmit the information further to the parent node. Sleepy end devices are highly efficient since they are helpful in extending the battery life .

3.4.3 Ensuring No Failure

The thread system has the advantage of working to full efficiency despite network failure or nodes losing connectivity by autonomously re configuring itself in-case of any faults. For instance if a end devices requires a parent node to transmit and the parent node fails due to connectivity problems, then the end device would choose another parent to transmit the data. Thread network topology is shown in Figure 2. A similar procedure is applied in case of failure of border router as well without any user intervention .

3.5 Real-Time Operating Systems (RTOS)

RTOS can be visualized as a software component that allows the system to rapidly switch between various tasks simultaneously on a single processing core and it gives an impression to the user that the system is working on various operations at a same time. Because in reality, the system can run only a single thread and RTOS decides the priority of the task to be handled to execute an efficient operation [9].

3.5.1 FreeRTOS

FreeRTOS is designed to run even on a small microcontroller applications. Microcontroller is an important component of a deeply embedded system and it would require a freeRTOS system to maximize the efficiency during critical applications. The critical features of RTOS are

1. Interrupt latency.
2. Resource-constrained processors.
3. Priority scheduling.
4. Improved efficiency.

3.6 Fight Controller and QGround Control

The flight controller acts like a heart to the drone system. The usage of fibre optics provide considerable advantages to the design. The flight movements are converted into electric signals and are transmitted to ground control station to monitor the flight activity. Pixhawk and Ardupilot are predominantly used flight controllers [10]. QGround control provides complete UI setup and configuration for PX4 software. Real-time screenshot of QGround system is shown Figure 3. Currently, PX4 is predominantly used in developing UAV applications. QGC provides flight map, way points and flight track which are essential to plan a flight mission.

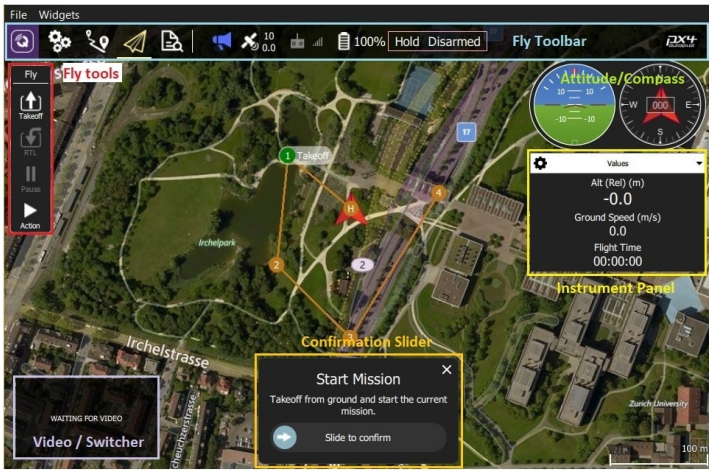

Figure 3: QGroundControl - Ground control station

3.7 Controller Area Network

CAN standard was introduced to harness complex wiring with two wires. It has more immunity to electrical noises and became a popular standard in automobile industries and other sectors. CAN bus works on the principle of multi-master and multi-slave system with message broadcast and maximum signalling rate of about 1 Mbps. The message/payload is Figure 4. ISO specification For CAN Bus broadcasted to all other nodes in the system. Types of CAN,

1. Standard CAN.
2. Extended CAN.

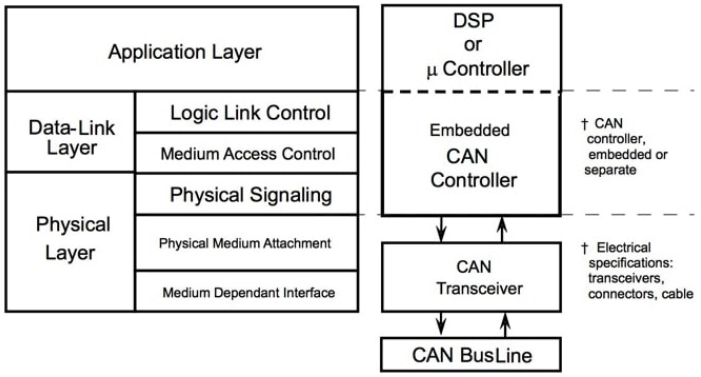

Figure 4: ISO specification for CAN Bus

3.8 UAVCAN

Unmanned Aerial Vehicle Controller Area Network (UAVCAN) protocol was initially designed for robotic and aerospace communications over the robust networks such as CAN bus . The design goals of UAVCAN are

1. Democratic network.
2. Nodes can exchange payloads.
3. Support for redundant interfaces and redundant nodes.
4. High throughput, low latency communication.
5. Simple logic, low computational requirements.
6. Common high-level functions to be defined.

In UAVCAN, each node is equipped with unique numeric identifier known as node ID as shown in Figure 5. 8 bytes of data can be sent in a single frame. In the event of huge payload, payload is divided and transmitted in multi CAN frames. CAN payload structure is displayed in Figure 6.

Message frame

Field name	Priority				Message type ID														Service not message	Source node ID						
CAN ID bits	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Allowed values																						0	1...127						
CAN ID bytes	3					2																1	0						

Figure 5: CAN frame format

Each published message would consists of an unique data type ID and each node in the network would consists of a node ID. This helps in preventing the redundancy in the network. Data Structure Description Language (DSDL) defines data structures format of data which has to be transmitted. Each DSDL file would consists of an unique identifier and a data type name. Few data types in the system are reserved for Vendor specific data types. DSDL helps in optimization which is predominant in deeply embedded systems where dynamic memory allocations may not be permitted. Two methods of data exchange are observed in UAVCAN. They are

- Message broadcasting.
- Service invocation.

Field name	Transfer payload	Start of transfer / End of transfer / Toggle / Transfer ID							
Payload byte	Up to 7 bytes	Tail byte							
Bit position		7	6	5	4	3	2	1	0

Figure 6: CAN payload

Message transfer and service transfer are popularly used modes of transfer. CAN 2.0A has 11 bit identifier and CAN 2.0B has 29 bit identifiers. The payload consists of 7 bytes followed by a single tail byte to indicate start of frame, end of frame, toggle and transfer ID.

3.9 Security Concepts

IoT devices should follow as secured design approach which would enable them for a safe storage of credentials, device authentication and secured data exchange with remote servers. Design criteria of IoT devices is crucial to aid in a safe storage of credentials, device authentication and secured data exchange with remote servers.

RSA (River, Shamir and Adleman) and ECC are recent developments in asymmetric encryption algorithms. ECC is based on algebraic structures of elliptic curves over a finite field which generate a certain public and a private key pair. A digital signature is used in guaranteeing the integrity and authenticity of the device. Digital certificates are like electronic documents which contain details like sender public key, identity of the owner and signature from a trusted entity, Certification Authority (CA).They are used in proving authenticity of the shared public keys.

Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm utilizes ECC to authenticate the devices. The key pair (public and private key generated from ECC) is used in signing/verification a message between sender and receiver. A signed message using the private key and a digital certificate is transmitted from sender side. The receiver device retrieves sender public key from digital certificate to validate the signature of the received message.

4 Hardware And Software Used

1. NXPs rapid IoT prototyping kit.
2. NXP's kinetic platform kits.
3. Pixhawk/PX4.
4. UAV system.
5. NXPs A71ch - Plug and trust for secure IoT applications.
6. Wireless sensor node.
7. IBM Bluemix - IBM cloud platform.
8. MCUXpresso IDE.

4.1 Rapid IoT Prototyping Kit

It is a power optimized IoT solution for rapid prototyping of ideas into POC. It includes pre-programmed applications that enables users to have a quick understanding about hardware and software architectures. It uses FreeRTOS. Web based IDE and GUI tools help in prototyping without any help from programming languages. Figure 7 displays rapid IoT prototyping steps. Wireless protocols such as ble, thread are supported in which thread protocol was used in this application to create a mesh network to transmit data from a remote end to master/parent node. It also provides A1006 authentication which is offered as a turnkey solution for developers to integrate security solutions into the host platforms. The complete authentication time is lesser than 50 ms. It uses FlexCAN architecture [11].

Figure 7: Rapid IoT - Prototyping

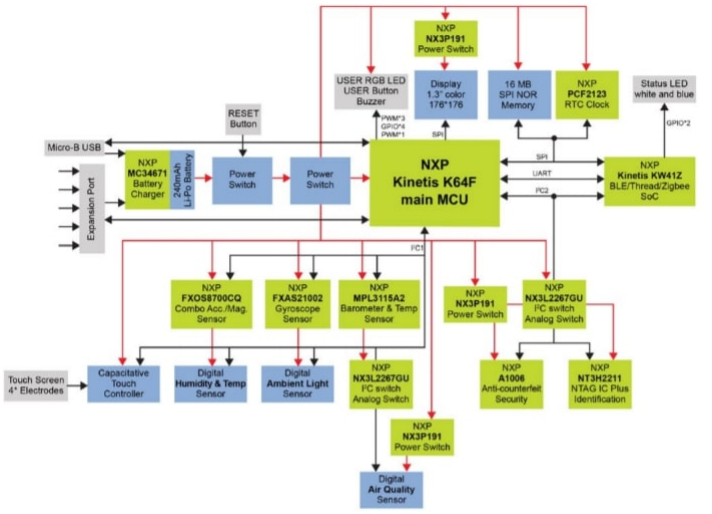

Figure 8: Rapid IoT architecture

The features of rapid IoT utilized in this research are

1. ARM Cortex M4-Core for application processing, Figure 8.
2. Thread connection establishment to a gateway inside the UAV system.
3. A1006 authentication for enhanced security features.

4. Software enablement including RTOS drivers, middleware and cloud platforms.

5. System enablement with advanced sensors like temperature, humidity, light sensors and digital air quality. Temperature sensor was utilized to get real time temperature values to cloud.

Temperature sensors values generated from rapid IoT at predefined intervals are utilized in this research testing.

4.2 Pixhawk/PX4

Pixhawk works on light-weight and energy efficient real-time operating system called NuttX. In order to access Ground Control Station, a command line interface called NuttShell (nsh) is utilized. It also provides micro Object Request Broker (uORB). uORB is used for asynchronous mode of communication to switch between tasks. Data transfer is done by using a simple publish-subscribe pattern. The transmitted and received data is visualized using ground control station. There are multiple feature advancement in recent years and key features utilized in this research are

1. 32bit STM32F427 Cortex M7.

2. 2 MB of memory.

3. 512 KB RAM.

4. ST Micro L3GD20H - 16 bit gyroscope, ST Micro LSM303D - 14 bit accelerometer / magnetometer.

5. Interfaces such as CAN, SPI, UART, I2C, ADC, PWM.

4.3 A71CH - Plug and Trust for secure IoT applications

The data generated by IoT devices is increasing everyday and ways to secure the data is a popular question among research communities nowadays.

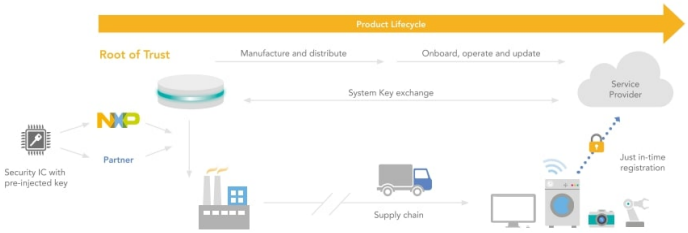

Figure 9: A71ch - Root Of trust

A71ch is a ready-to use solution since it contains necessary private and public keys for the device which enables the device to securely connect to private/public clouds. Figure 9 shows A71ch product features. This security IC can be easily interfaced with various hardware and operating systems. This security IC comes in two product variants:

1. Customer programmable Type.

2. Provisioned and programmable type.

The first type is delivered without any credentials and used in designing new products, to test and evaluate the designs. The second type is ready-to-use IC which is already provisioned by NXP trust provisioning support to ensure a secured TLS connection with IBM Bluemix. A71ch is interfaced to Micro-controller Unit (MCU) using Inter-Integrated Circuit (I2C) protocol. A71ch provides root of trust at the IC level. It delivers proven, chip to the cloud security so connection can be established with cloud platforms such as IBM, AWS and google cloud without the need to write security keys or exposing certificates/credentials. But when the data transmission occurs for the first time, proof of origin of the device and anti-counterfeit protection are required to protect server and device authenticity and avoiding un-trusted servers attacking the system. The complete authentication procedure is composed of below steps:

1. Server certificate verification.

2. Server authenticity verification.

1. IoT device certificate verification.

2. IoT device authenticity verification.

A71ch is used for anti-counterfeit protection of the device against many physical and logical attacks by operating autonomously based on integration of Javacard operating systems and applet firmware. Physical attacks on the system is narrowed down by allowing direct memory access only by fixed functionalities of applet. This isolates the content of memory from host system. The server stores a unique key pair and a digital certificate signed by trusted CA. On the other hand, the IoT device would also have a key pair and a signed digital certificate. The digital certificate's are issued by same trusted CA. These credentials are stored inside the A71ch chip. Server's certificate and authenticity is verified by IoT device using the CA certificate and a random challenge as a sequential process. The challenge involving generating and sending random message to validate the server. The same steps are followed to verify the IoT device's certificate and authenticity.It can be deployed across a broad range of systems to establish a secure connection. Since the communication between A71ch and the Host is based on I2C protocol, it can be carried out through a Secure Channel Protocol (SCP03). This would protect the channel against external attacks such as replay attacks. Also ensuring only the authenticated MCU can exchange information with our security IC by use of session keys. The features of A71ch include

1. Systematic enforced authentication.

2. ECDSA.

3. An unique chip ID.

4. Freezing of credentials.

5. Possibility to lock A71ch as transport lock mechanism.

6. HMAC SHA256 calculation in one shot.

The applications of A71ch include home gateways, home appliances, sensor networks gathering confidential data, connected industrial devices , Figure 10.

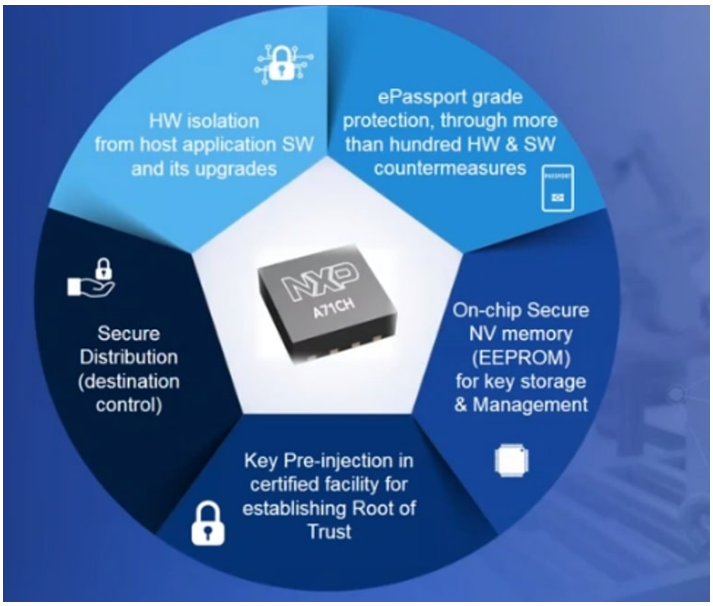

Figure 10: A71ch - Plug and trust for secure IoT [4]

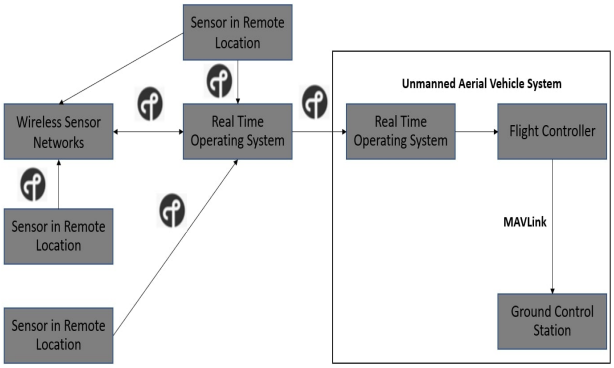

Figure 11: Data transfer using thread protocol from WSN to UAV system

5 Proposed Model and Implementation

This sections explains the initial design phase construction and the modifications in hardware platforms applied for this research work.

5.1 Design Phase

With the help of hardware and software information detailed in section 4, the design phase includes construction of an UAV system using Pixhawk board as flight controller. The interfaces in Pixhawk are helpful in connecting external peripherals like motor, telemetry and GPS module to the UAV system. In order to make the UAV system to adapt for this research, the system was modified by utilizing the CAN bus interface present in flight controller. Two new systems such as Rapid IoT and A71ch are connected to flight controller module.

In order to facilitate the data exchange and visualization aspect, inside the PX4 firmware, two new UAVCAN messages were created each having unique .c and .h files structures. One message to receive data from external input sources and other message to send the information to external devices using a simple uORB pub-sub methodology. In order to encrypt and decode the CAN frame, UAVCAN GUI tool was adopted in testing and result phases.

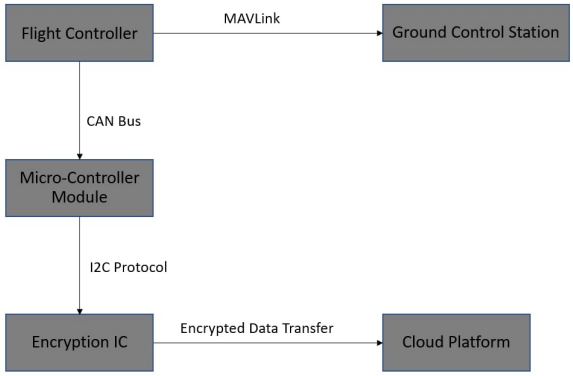

Figure 12: Data transfer to cloud platforms from UAV system

5.2 Integration of Rapid IoT with Pixhawk

Owing to advantages of rapid IoT system, documented in section 4.1, it is embedded with the UAV design using UAVCAN (CAN bus interface) and with help of a docking station. External CAN transreceivers are attached to the docking station. They convert CAN TX and RX signals into CAN High and Low signals which in-turn will be used for transmission. NXP's FRDM KW41Z is assigned to be remote wireless sensor node and it is placed at a distance of 70 - 100 metre. Both the Rapid IoT and remote node are pre-programmed with thread for this research. The remote node would gather atmospheric temperature value for every second. Any sensor data/ raw data can be used here. Once the UAV system is armed, the data from remote node is transmitted wirelessly using thread to rapid IoT in the UAV system, Figure 11.

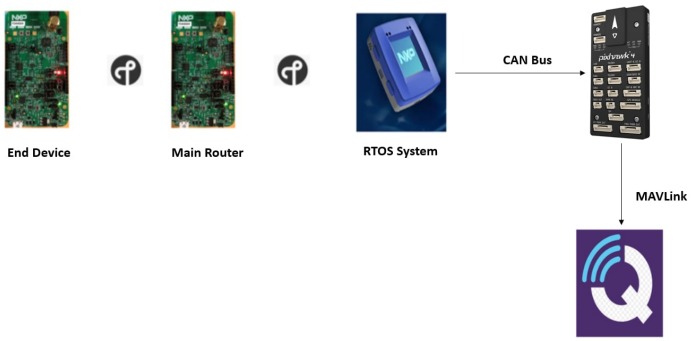

Figure 13: Integration of rapid IoT with pixhawk

The rapid IoT would in-turn send the data as CAN signals to the flight module/controller. The flight controller would re-transmit the data to ground Control station using telemetry and readings are visualized in QGroundControl. This testing phase was carried out at initial step. The design setup was increased by increasing the number of wireless sensor nodes and establishing a mesh network connectivity among them. This enabled UAV system to gather data from the nodes which are about 350 metres. Figure 13 shows the hardware design of UAV system with rapid IoT.

5.3 Establishing a secure data transmission

Using telemetry, the data from flight controller is sent to ground station as radio signals. In order to secure the data, A71ch is connected to Pixhawk using CAN bus protocol. Provisioned and programmable type of A71ch was used in this research. But A71ch is a security IC and it has to be used along with a base board like NXP's LPC54018 which has an inbuilt WiFi module. Establishing a secure connection between A71ch and IBM cloud require three steps. In the first step, A71ch is interfaced with host MCU using I2C protocol. Using the host MCU's debugger port, keys and certificates are inserted into A71ch. Device ID (Identification number - unique number) and generated CA certificate from the device will be utilized in next step. Later, a new user account was created in IBM Bluemix with an organization ID (name of your organization) and registration is completed by entering the device ID and uploading CA certificate from previous step. This portion of initial setup is important to establish a secured connection between cloud and security IC. The final step would be to acquire API key, authentication token from cloud and use it in our code base to initiate the connection from our IC. If the API key is lost or keys are refreshed, the connection setup will not succeed. The above process has to be repeated again in this case. Every provisioned A71ch has to carry-out this one step procedure to function without any failures. The generated data from flight controller is transmitted as CAN frames to LPC module, where security IC receives the data through I2C communication. It would encrypt the data and upload it to cloud as shown in Figure 12. This process would secure the data to prevent data leakage and restrict unwanted access to the data. Hardware design is shown in Figure 14.

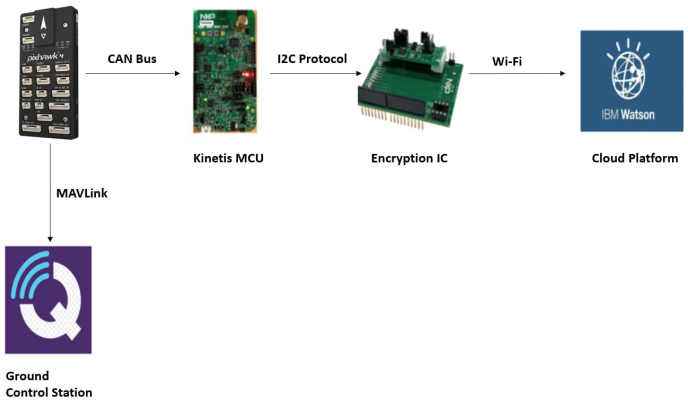

Figure 14: Integration of security IC with pixhawk

6 Testing Phase And Deployment Design

At first, the hardware connections were tested and telemetry was checked if a proper communication channel has been established with the UAV system from the ground control station. The CAN communication was tested between NXP's Kinetis boards using Tera term.

Figure 15: Deployment system of UAV to acquire data from WSN

Later, the interfaced connections were also tested. In various phases of testing, an oscilloscope was used to check for CAN frames. As a next step, UAVCAN functionality in Pixhawk was tested in the lab environment. To ensure whether the CAN frames are being transmitted from the hardware. A publish-subscribe pattern is tested initially to understand the uORB functionality. The status of UAVCAN frames transmitted is witnessed with the help of UAVCAN frame status command in nsh shell.

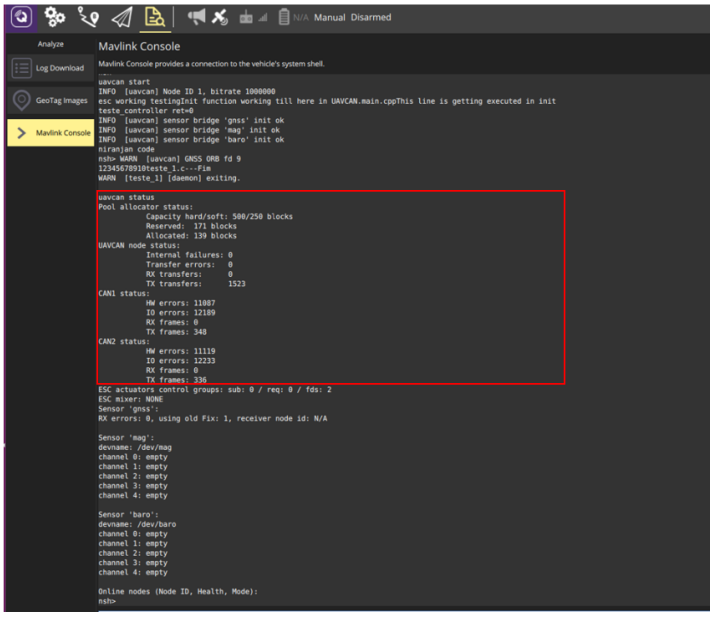

Figure 16: QGroundControl - UAVCAN frame status

An alternative way of testing and visualizing the CAN frames was UAVCAN GUI tool. It shows the sender ID of the CAN frame and the payload. The other alternative was to store the log data and understand the payload and its format during any transmission.

After the testing, two advanced design were implemented using above specified hardware designs.

Figure 17: Deployment design of UAV to transmit data to IBM Bluemix

The designs are shown separate for a better understanding, Figure 15, 17. They can be integrated together and deployed as a single design as well. It had also been tested in this research.

7 Results And Demonstration

Extending the range of data transmission using wireless technologies like thread and incorporating security IC to secure the data transport is tested with the help of NXPhlite,Pixhawk, NXPs rapid IoT and NXP's security IC in real-time environment is tested. Before performing an actual field testing, an initial testing was carried out in two steps. The first step is to ensure the connectivity of hardware to check if there are any wiring issues and second step ensuring all the network devices were working perfectly.

As a first step, Pixhawk module was interfaced with four devices using a CAN network. 8 bytes of data for each CAN frame was transmitted and it was received successfully by other devices as well. This data transmission rate of 8 bytes/second is significantly higher compared to traditional UART (Universal Asynchronous Receiver/Transmitter). The successful completion of this step ensured a fully functional CAN network was established between all the hardware with varied CAN protocols.

In the second step, thread network connectivity was tested between the network devices and connection establishment to IBM cloud was also verified. As mentioned in section 5.3, the unique device ID of A71ch IC is cross-checked with ID displayed in IBM cloud, Figure 19. After successful completion, field testing was carried out.

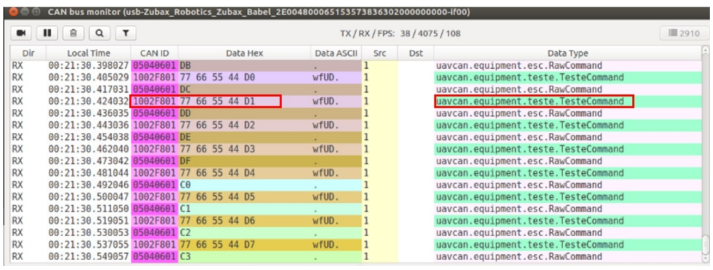

Figure 18: CAN frames - UAVCAN GUI tool

As proposed in section 5.2, with the help of QGround control, Figure 16 shows status of CAN frames both received/transmitted. Standard CAN of payload 8 bytes is tested in this practice. UAVCAN GUI tool displays the values displaying the CAN frames, 8 individual bytes which are being sent from rapid IoT system and which are received by the flight controller, Figure 18.

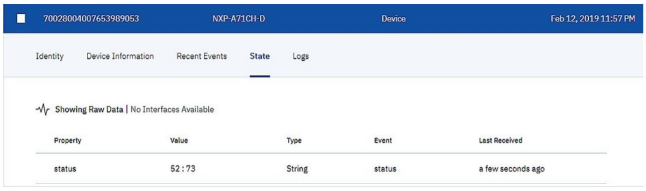

Figure 19: IBM Bluemix - Output data visualization

To establish a secure data transmission, as proposed in section 5.3, the generated values from the flight controller is transmitted to security module which then encrypts and shows the value in IBM Bluemix for each second/whenever the data is published. Figure 19, shows the device ID of the security IC and value column displays the actual value in a string datatype. This is give access across the globe to see the results instantly regarding the flight data. Figure 19 displays instant data transfer into the IBM cloud.

8 Design Challenges

Various design challenges were faced during the research. NuttX operating system had very little documentation with regards to UAVCAN. GitHub repository of PX4 is also being upgraded from time to time but the documentation did not reflect all the changes and new features. UAVCAN is a developing research area and it is difficult for developers to re-design/upgrade the existing design. Only few articles/documentations were available on how to interface flight controller with external hardware devices with UAVCAN [12]. Next challenge was, NXP's rapid IoT used FlexCAN and Pixhawk had UAVCAN. The CAN frame structures were different and without suitable debugging tool, it was difficult to proceed further. To facilitate communication in CAN network, external CAN transmitters were used.

Another challenge was to place the new hardware in UAV system. 3D printed models were very helpful in overcoming above issue. With regards to security IC, understanding the background of security concepts was mandatory. Initial setup and key exchange between IBM cloud and security IC had to be carried out in sequential steps. Any error in the steps will lead to failure and start the process all over again.

9 Conclusion

Technological development and growth in various sectors of life plays a crucial role in making human lives better each day. Internet, a recent trend acts as a stimuli to the process. But there are places in globe, where technological advancements are not fully available and such places remains isolated from the rest of the world. The ideal aim of the research is to provide connectivity/access to areas where

internet connectivity is not good and the places where it is difficult to reach because of terrain. Also, ensuring secure data transport in regions where limited internet connectivity is available.

To achieve that, the design system of an UAV system was proposed. The background knowledge of software technologies and hardware components were detailed in sections 3 and 4. The system was equipped with remote wireless devices which are connected with thread network, the interfacing of flight controller module with other MCUs were carried out using UAVCAN, security of data transmission during over the air transfer (OTA) was addressed with the help of security IC, A71ch. Using the proposed UAV system with wireless data transfer facility, would foster growth quick data gathering and transmission compared to traditional methods. By combining these two ideas, it would result in building an Eco-friendly and a safe environment all over the globe.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgments We would like to thank NXP, UAVCAN forums for guidance and supporting in completion of this research.

References

[1] L. Mainetti, L. Patrono, A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey," 2011 International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2011, **16**(8), 16–21, 2011.

[2] C. Rotariu, R. G. Bozomitu, V. Cehan, A. Pasarica, H. Costin, "A wireless sensor network for remote monitoring of bioimpedance," Proceedings of the International Spring Seminar on Electronics Technology, **2015-Septe**, 487–490, 2015, doi:10.1109/ISSE.2015.7248046.

[3] R. Chitanvis, N. Ravi, T. Zantye, M. El-Sharkawy, "Collision avoidance and Drone surveillance using Thread protocol in V2V and V2I communications," Proceedings of the IEEE National Aerospace Electronics Conference, NAECON, **2019-July**, 406–411, 2019, doi:10.1109/NAECON46414.2019.9058170.

[4] N. Ravi, M. El-Sharkawy, "Integration of UAVs with Real Time Operating Systems using UAVCAN and Establishing a Secure Data Transmission," Ph. D Thesis, IUPUI, 2019.

[5] N. Ravi, R. Chitanvis, M. El-Sharkawy, "Applications of Drones using Wireless Sensor Networks," Proceedings of the IEEE National Aerospace Electronics Conference, NAECON, **2019-July**, 513–518, 2019, doi:10.1109/NAECON46414.2019.9057846.

[6] S. R. Vangimalla, M. El-Sharkawy, "Remote wireless sensor network range extension using UAVs with thread protocol," Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018, 902–906, 2018, doi:10.1109/CSCI46756.2018.00178.

[7] P. Bernardo, "Development of technology and procedures for health monitoring of UAV subsystems Examination Committee," M.S Thesis, Instituto Superior Técnico, (November), 2015.

[8] L. Zimmermann, N. Mars, M. Schappacher, A. Sikora, "Development of Thread-compatible Open Source Stack," Journal of Physics: Conference Series, **870**(1), 2017, doi:10.1088/1742-6596/870/1/012001.

[9] A. Gerstlauer, H. Yu, D. D. Gajski, "RTOS modeling for system level design," Design, Automation, and Test in Europe: The Most Influential Papers of 10 Years Date, 47–58, 2008, doi:10.1007/978-1-4020-6488-3_4.

[10] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," Proceedings - IEEE International Conference on Robotics and Automation, 2992–2997, 2011, doi:10.1109/ICRA.2011.5980229.

[11] J. Pimentel, J. A. Fonseca, "FlexCAN: A Flexible Architecture for highly dependable embedded applications," Int. Workshop on Real-Time Networks, **1**(3), 2004.

[12] N. Ravi, M. El-Sharkawy, "Integration of UAVs with Real Time Operating Systems using UAVCAN," in 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 0600–0605, 2019, doi:10.1109/UEMCON47517.2019.8993011.