

Indoor Positioning System

Ximeng Zhang
zhan3047@purdue.edu

QianLi Ma
ma160@purdue.edu

Follow this and additional works at: <https://docs.lib.purdue.edu/purc>

Recommended Citation

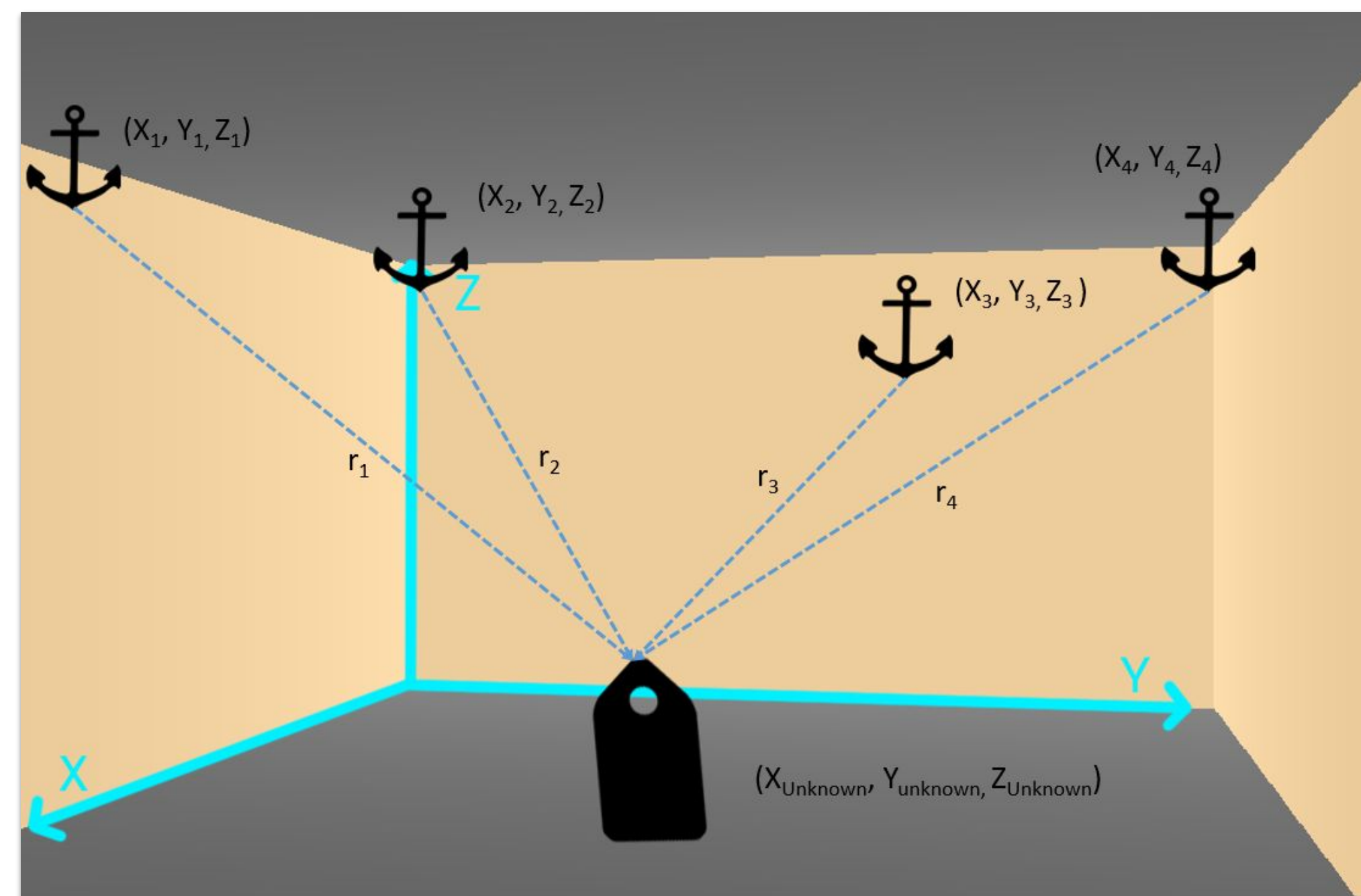
Zhang, Ximeng and Ma, QianLi, "Indoor Positioning System" (2019). *Purdue Undergraduate Research Conference*. 6.
<https://docs.lib.purdue.edu/purc/2019/Posters/6>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

Introduction

The current demand for localization extends beyond what GPS can provide as it is limited to outdoor settings. Radio signals used in the Global Positioning System offer global coverage but are unable to penetrate obstacles and buildings. Our system is designed to solve this problem by implementing an Indoor Localization System using ultra-wideband RF-based localization based on the recent 802.15.4a standard. At a high level, the system is modeled after the architecture of the Global Positioning System by utilizing anchors as the satellites and tags as the receivers. With the use of cloud technology, an end-to-end system is created that achieves security as well as usability. The hardware packaging is encapsulated within a miniature PCB design at a low cost, aimed as a plug-and-play integration within other systems in need of indoor detection. Applications of our IPS design include domains such as navigation (room-to-room assistance in a building), national defense (search and rescue operations, underground tracking, surveillance), commercialized zones (indicators for specific products on shelf, asset tracking in warehouses), and robotics (autonomous vehicles, drone detection). We demonstrate that all of the components mentioned are essential to effectively carry out successful indoor positioning with a focus on user flexibility and efficiency in response.

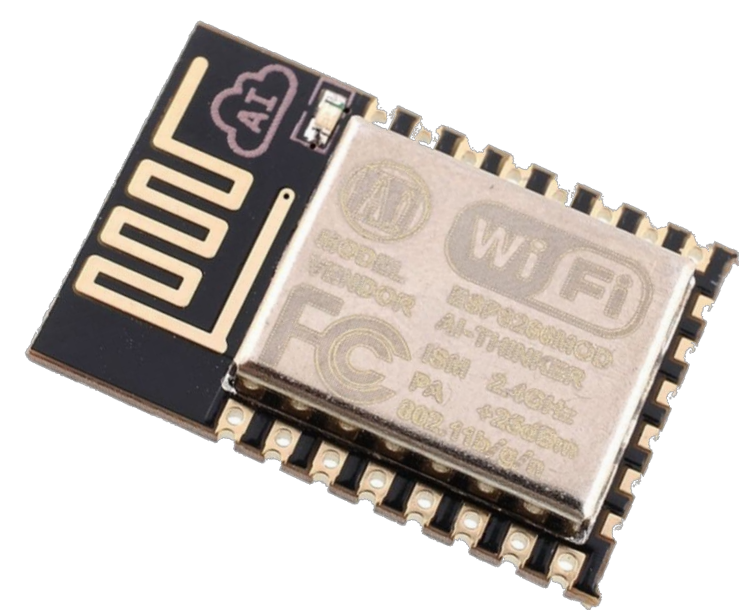
Visual Concept



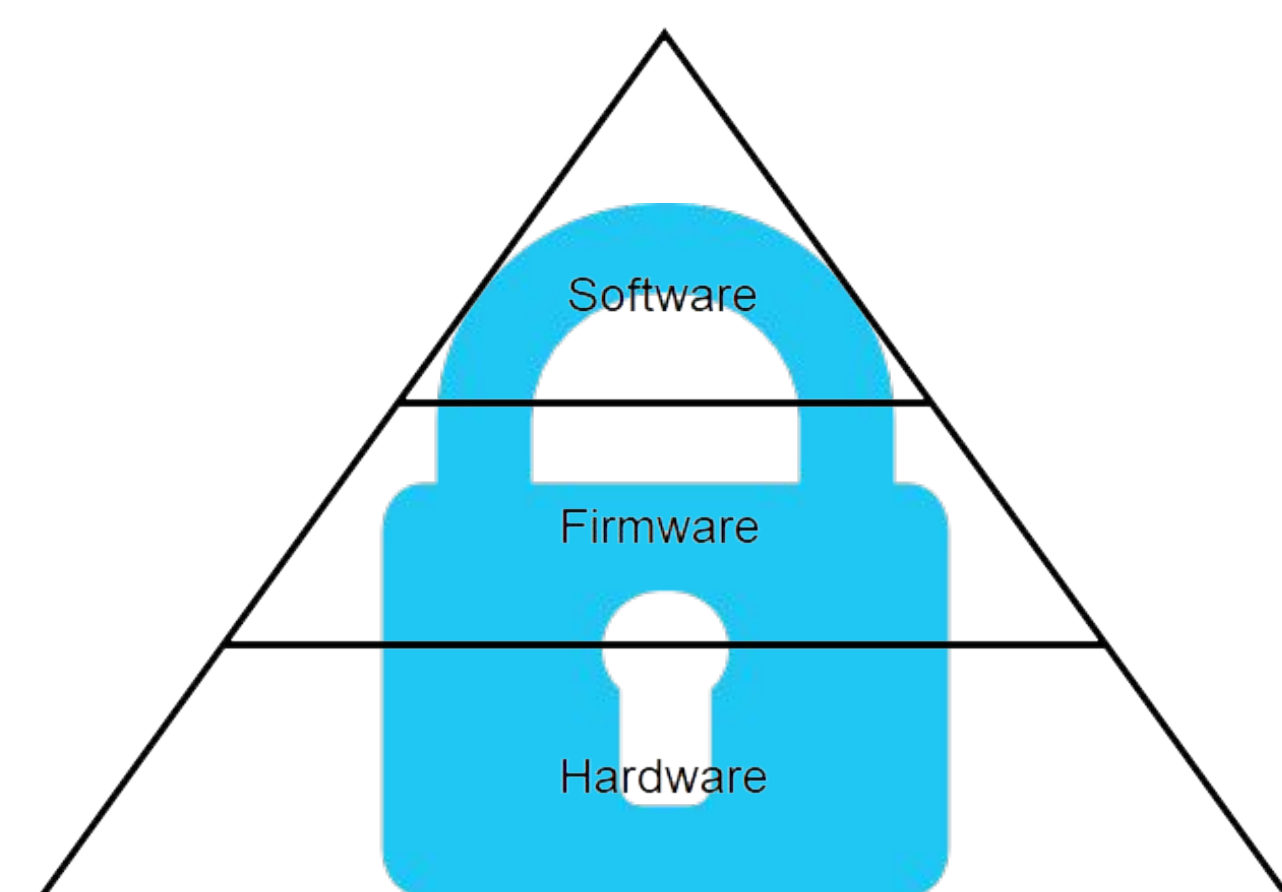
Materials

ESP8266 Microcontroller

DWM1000: 802.15.4a UWB Compliant Transceiver



System Architectural Hierarchy



Method

1. Leveraging the user interface from our mobile application, the user selects Anchor locations (minimum 4 for 3D positioning) on a selected map and building schematic
2. All information is pushed to a cloud broker through Amazon Web Services into associated MQTT topics for M2M communication
3. Tag module remains "subscribed" to topics and receives information as they arrive. Everything received is stored into memory in the firmware.
4. Using the geometric method of Trilateration, the position of the Tag is calculated from coordinate and distance metrics from Anchors
5. Calculated Tag coordinate is pushed onto AWS for the mobile app to continuously display Tag location via a marker
6. Via UART on the firmware, NMEA sentences are reported for plug-in-play integration with other systems dependent on GPS modules
7. Longevity of module life cycle is maintained near 90% efficiency through custom PCB design and low-dropout regulator integration

Determining Tag Location (Trilateration)

Known Information: Number of satellites in 2D space: (n)
Positions of satellites (x_i, y_i)
Distances from transmitters to receiver (r_i)

Unknown: Position of Receiver (X, Y)

Distance Formulas:

$$(X - x_1)^2 + (Y - y_1)^2 = r_1^2 \quad \text{(Satellite 1)} \quad \text{Expansion} \rightarrow X^2 - 2x_1X + x_1^2 + Y^2 - 2y_1Y + y_1^2 = r_1^2 \quad (1)$$

$$(X - x_2)^2 + (Y - y_2)^2 = r_2^2 \quad \text{(Satellite 2)} \rightarrow X^2 - 2x_2X + x_2^2 + Y^2 - 2y_2Y + y_2^2 = r_2^2 \quad (2)$$

$$(X - x_3)^2 + (Y - y_3)^2 = r_3^2 \quad \text{(Satellite 3)} \rightarrow X^2 - 2x_3X + x_3^2 + Y^2 - 2y_3Y + y_3^2 = r_3^2 \quad (3)$$

(1) - (2): $(-2x_1 + 2x_2)X + (-2y_1 + 2y_2)Y = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$

(2) - (3): $(-2x_2 + 2x_3)X + (-2y_2 + 2y_3)Y = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2$

System of Two Equations with Two Unknowns:

$$\begin{cases} AX + BY = C \\ DX + EY = F \end{cases} \quad \begin{cases} X = \frac{CE - FB}{EA - BD} \\ Y = \frac{CD - AF}{BD - AE} \end{cases}$$

Figure 1: 2-Dimensional Space Derivation

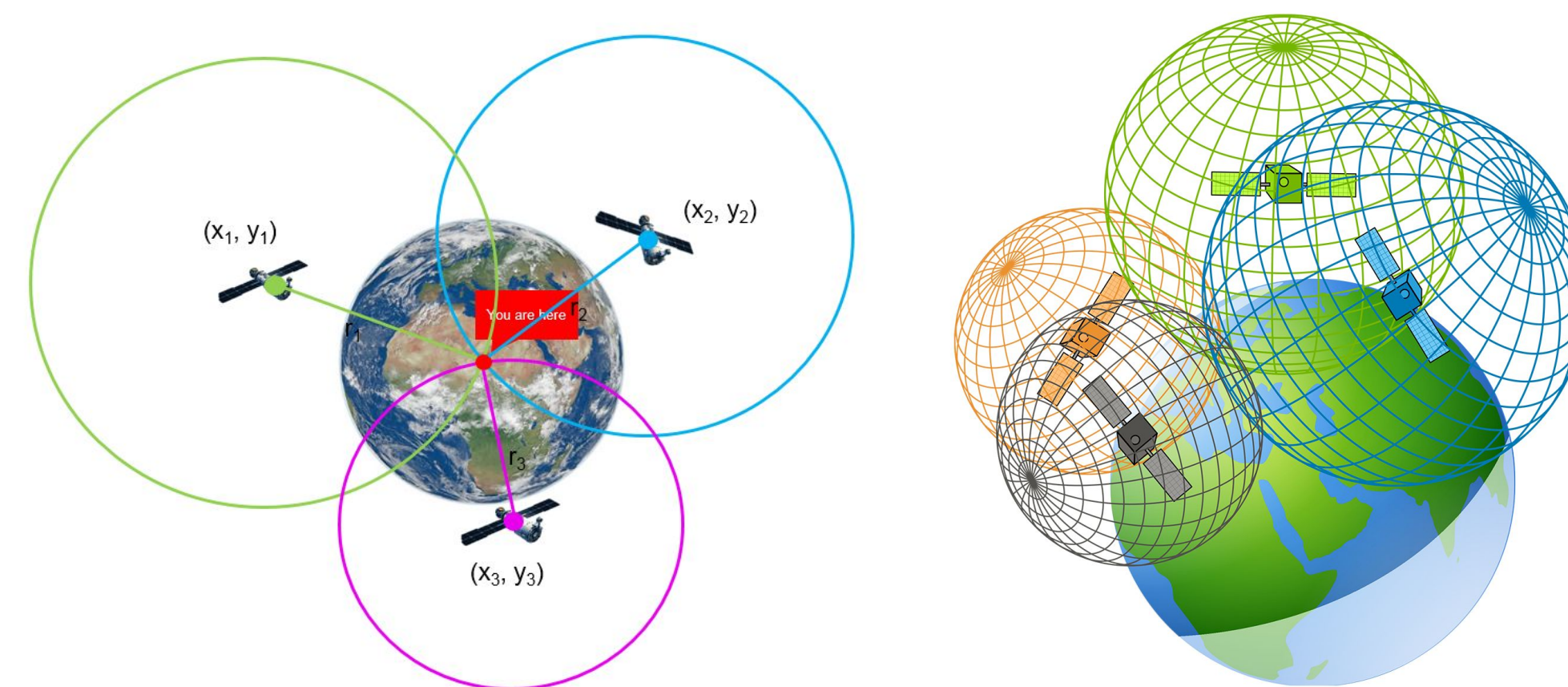


Figure 2: 2D (X, Y) extension into 3D (X, Y, Z)

Machine-to-Machine Communication (Amazon Web Services)

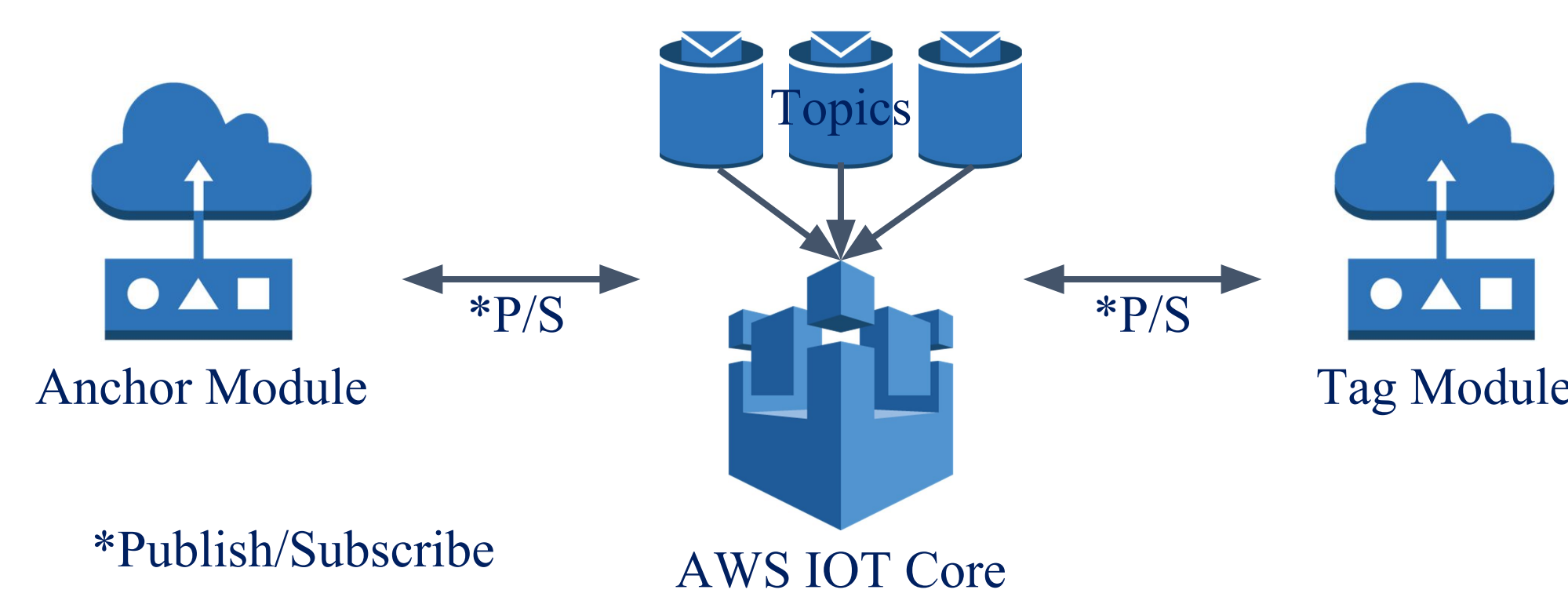


Figure 3: AWS IoT Architecture

AWS provides a secure and comprehensive cloud infrastructure for the use of M2M communication via the IOT Core. A primary module will remain "subscribed" to a topic where a secondary module will then "publish" to that topic. At the time of publishing, the subscribed module will receive anything published onto the topic. This is known as MQTT, a messaging protocol that sits on top of a TCP/IP network layer.

Results

Hardware

For any project, the hardware is the foundation that makes the magic possible. For our concerns, it has been necessary to create a pcb design that encapsulates all hardware components in our system into a sleek and miniature design for the purposes of easy deployment into the user's environment as necessary. What has been achieved includes interfacing with the ESP8266 microcontroller chip and the DWM1000 frequency module chip in addition to integrating a 6-pin Tag-Connect programmer, Low-Dropout regulator for stable power and efficiency, and a switching transistor for auto-programming ability via RTS and DTR signals given from common serial communication bridges.

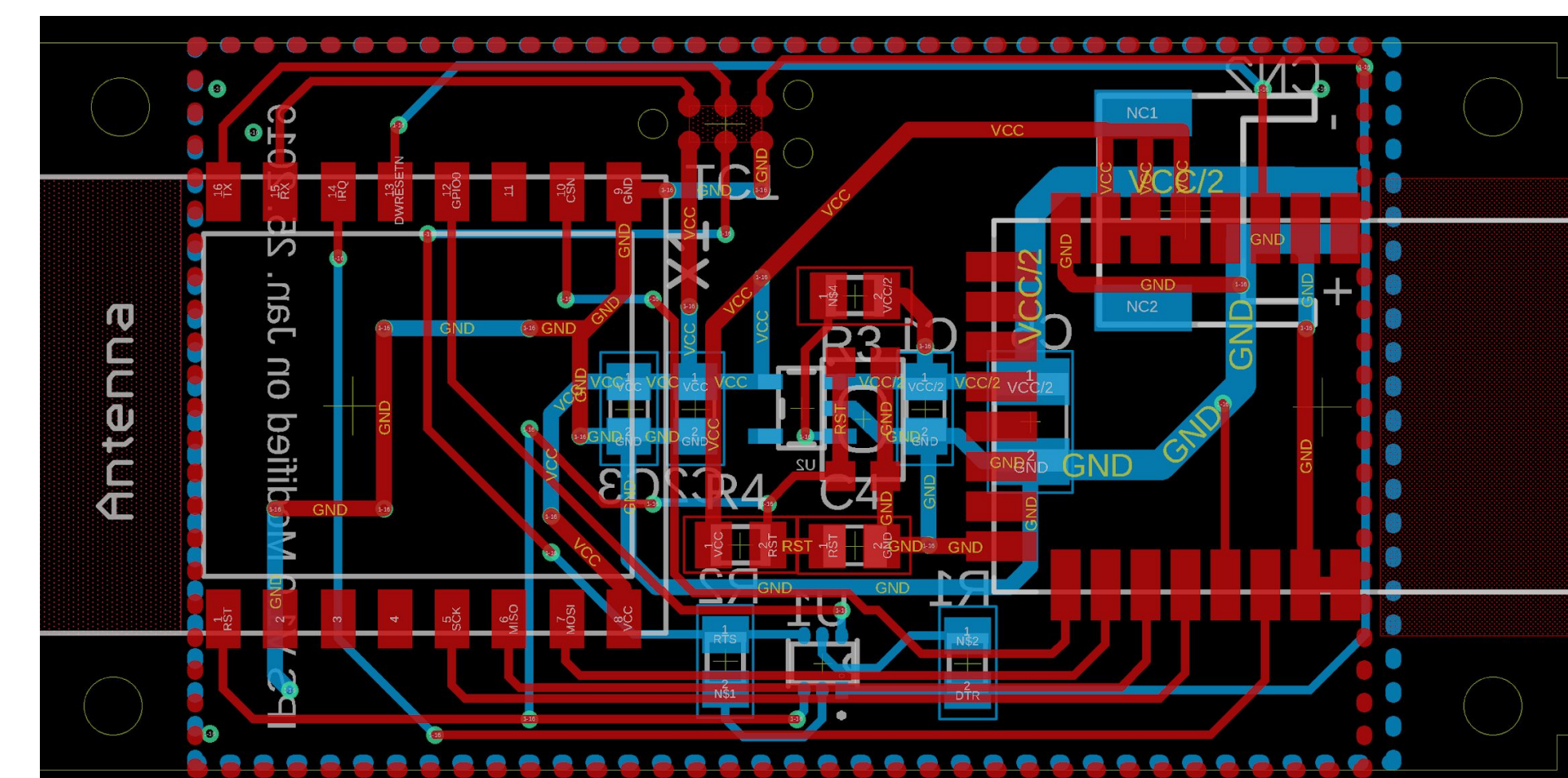


Figure 4: Updated PCB Design

The firmware includes the ability to "publish" and "subscribe" information back and forth with AWS and receive distance metrics from the DWM1000 frequency modules. In addition, trilateration is performed in 3 dimensions. In order to allow seamless integration with other systems, NMEA sentences are reported via UART. For the ease of upgrading the firmware, Over-The-Air programming has been enabled via the inclusion of a WiFi bootloader. This is valuable in situations where Anchors modules may not be physically accessible ie. on the ceiling, etc. Finally, thinking about our system in terms of scalability, the ability to dynamically change the state of the modules (tag/anchor) has been implemented to allow the user to add more anchors for location coverage without adding additional network

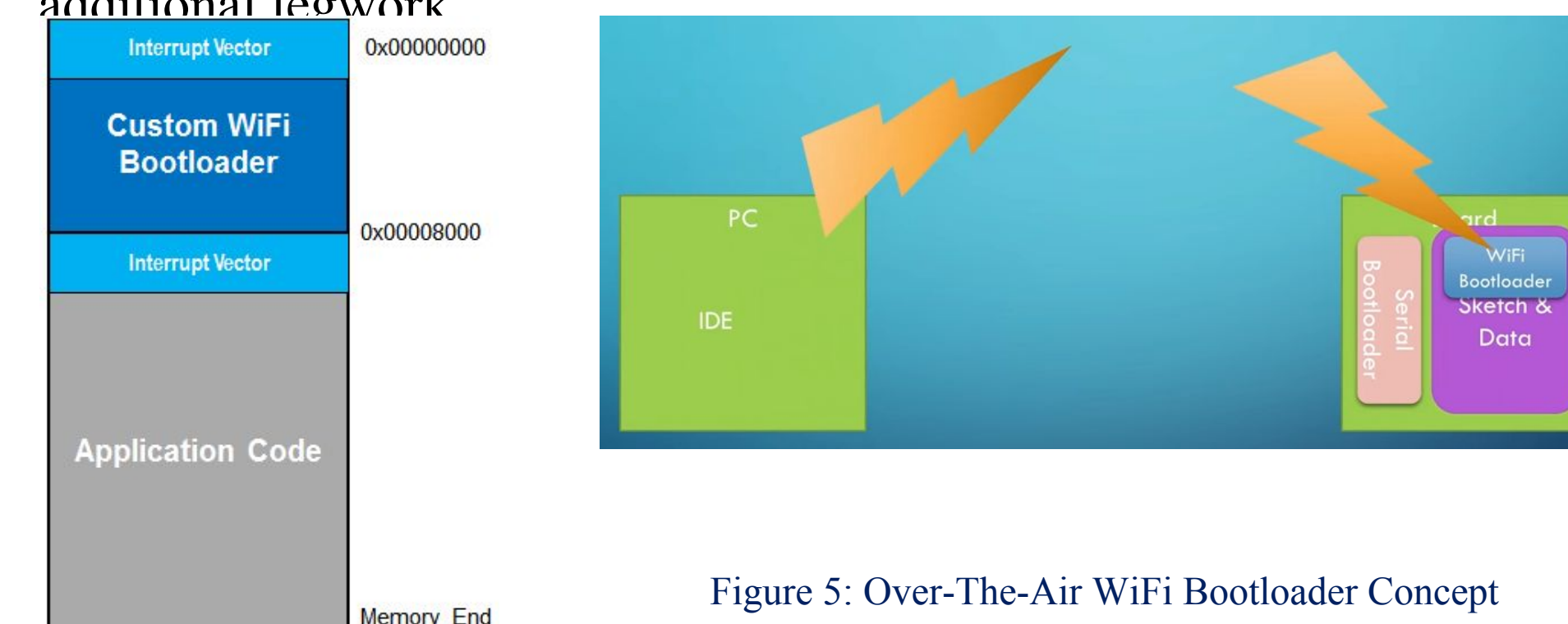


Figure 5: Over-The-Air WiFi Bootloader Concept

Security

Security is a crucial component that ensures safe data management and access for location information. Using the AWS platform, two services are specifically used for our security needs, Amazon Cognito and IAM (Identity and Access Management). Amazon Cognito is able to control user authentication and access for the mobile application. It collects user profiles and saves them into a fully secure directory called "user pools". The service also allows for data synchronization between multiple devices. If the users changes their identity information on one machine, other machines which share the same identity will update the information as well. IAM allows the use of the websocket protocol for HTTP requests made on API calls to the cloud. The security involves client specific keys, secrets, region, port, and endpoint.

Software

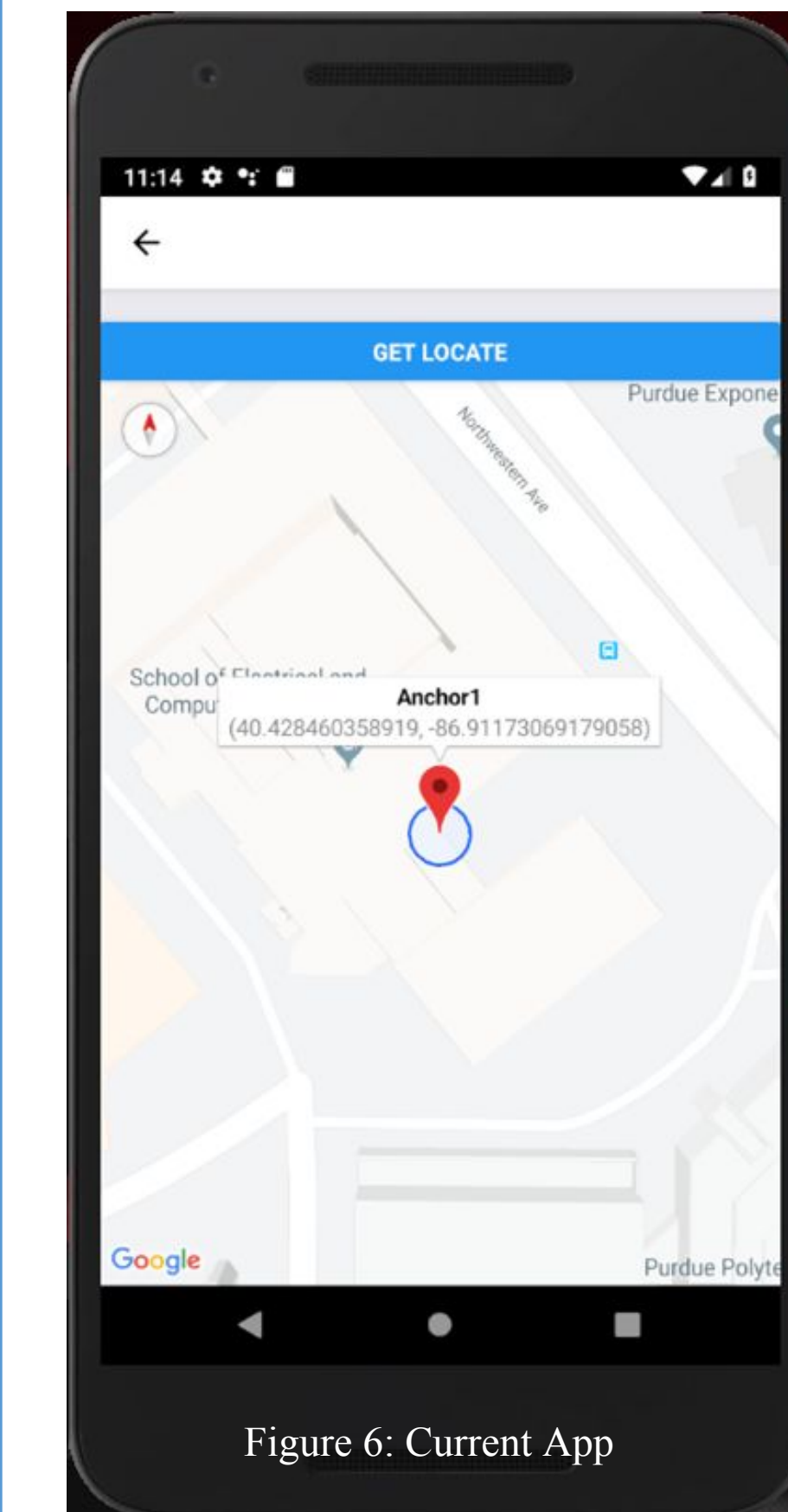


Figure 6: Current App

The top layer of the project is a mobile application developed on React-Native that supports both Android and IOS platforms. Focusing on more of the application of our system, the mobile app enables users to set anchor positions on a given map and building schematic through the "consumer mode". Unlike prior implementations of the system that involved creating a coordinate plane relevant to our system only, by leveraging Google Maps, we are now able to consistently work with GPS coordinates for all modules involved. The goal of this app is to serve as a visual guide for the user to set up all necessary anchors within their environment of interest. Once the system is set up by initializing at least 4 anchor positions, users (Tags) can access their real time location within the building via a marker. In addition to the consumer aspect, the app also serves as a tool to aid in development and debugging for involved developers. By accessing the developer mode, metrics such as module "heartbeat" and "life-cycle" may be viewed to ensure quick deployments and troubleshooting. The developer may also access this page to enable Over-The-Air programming for any firmware updates needed to be pushed.

Drone Demonstration

One application of the indoor positioning system is realizing swarm programming of drones indoors which is not possible without the system as drones not have enough location information indoors due to the lack of GPS information to be aware of their current location. The system enables collision avoidance through a central controller which coordinates each drone in the swarm. Each drone would carry one tag to transmit current location. By entering destinations for each drone through our GUI, paths would be calculated and command would be passed to each drone to execute.



Figure 7: Tello Drone

Conclusions

Indoor Positioning has been a relevant topic for decades as researchers and scientists have been working on a method to extend the usage of the Global Positioning System into an indoor setting. The proposed solution has been verified to work in both the 2D and 3D plane with the required extension of a single anchor in the system as well as with a focus on information security on the cloud. To further the design in the future, we aim to improve the hardware with more power efficient options such as BLE (Bluetooth Low Energy). On the firmware side, we are working on methods for dynamic cellular-handoff of the tag to different sets of anchors to enable easy movement. On the app side, more features are looking to be added such as user registration. Further implementation may include a "recommendations" feature that will remove the need for user-defined anchor placement during the initialization process.