To the Editor: Content highlighted in grey are directives to you.

[Preview]
What do you do when there's no cell service, but you need to contact someone far away? The current market for off-grid communication systems is missing an affordable, user-friendly option, so three Camosun College students created just that. FloraNet is a battery powered and accessible network that uses LoRa radio to keep people in touch in the back country.

[Article Start]
We are Flora Communications, a group of three Electronics and Computer Engineering Technology students at Camosun College. For our capstone project, we've built an alternative off-grid communications network using LoRa Radio that we've dubbed "FloraNet" (Figure 1). The floral name and branding was inspired by the fields of avalanche lilies we saw while we were playing with LoRa in the Pacific Northwest backcountry this summer. FloraNet is a pre-installed network that operates on battery and solar and does not require an internet connection or cell service. FloraNet is designed to be as user-friendly as possible, accessed via web browser through a JavaScript and HTML web app served by an ESP32-S3 microprocessor over a locally hosted Wi-Fi connection. Users can post messages to the public network chat and contact emergency services directly from their smartphone without needing a user account, pre-installed app, or additional hardware.

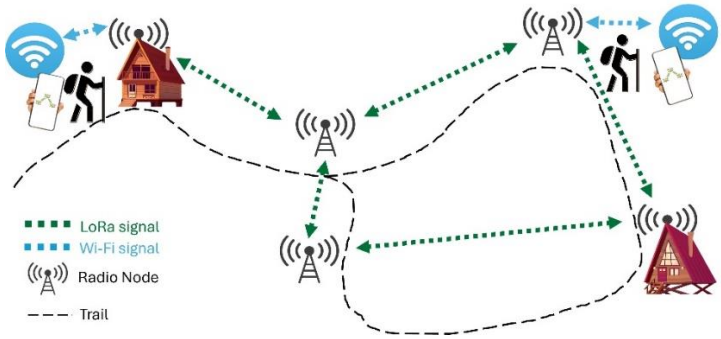

Figure 1
Type: *Image*
Source: *FloraNet.jpg*
Caption: *FloraNet is made up of Petal Radios that send messages using LoRa radio using our firmware and provide a user-friendly interface using a locally hosted Wi-Fi network and custom web app.*

Good Products and Great Products
We believe that a good product is well-designed and useful, but a great product is well-designed and useful *for everyone*. As inventors and enthusiasts, it's easy for us to lose sight of three main barriers to a product being widely adopted and useful for a customer base: complexity, cost, and the customer themselves. The average customer is more prone to these pitfalls than someone who is passionate about the product or technology. If the customer needs a lot of background knowledge or savvy to set up and use the product, the customer base will be limited to the people who have the know-how. If the customer needs to pay a large up-front cost or a recurring subscription to use the product, this limits the customer base to those who can afford it. If the customer feels inconvenienced by the user interface or is constantly needing to troubleshoot the product, they'll quickly move onto something different. The current options for off-grid communication systems are either prohibitively expensive (like the Garmin InReach satellite transceiver) or complex and inconvenient (like LoRa-based Meshtastic). As both technology and outdoor enthusiasts, we saw that the off-grid backcountry communication market is missing a *great* product.

FloraNet targets outdoor recreation areas outside of cell service that see a high volume of casual visitors, such as ski resorts, national parks, or popular hiking trails. Organizations that maintain these places (resort operators, recreation departments, outdoor clubs, etc.) can install a network of the Petal Radios (Figure 2) we designed (also referred to as *nodes*), scattered throughout the desired coverage area every few hundred to thousand metres, depending on the terrain.

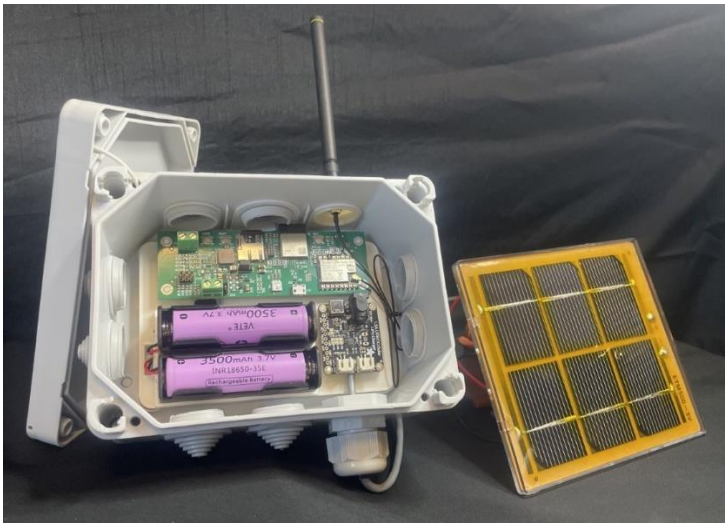

Figure 2
Source: *petal-node.jpg*
Type: *Image*
Caption: *A Petal Radio node with the Petal Radio, enclosure, LoRa antenna, solar panel, charge controller, and battery pack on a custom 3D-printed backing plate. In a real installation, the enclosure would be waterproof, and an external activation button would be wired to the Petal Radio through a cable gland.*

Visitors can go to a Petal Radio node and press the button to activate it. Using the smartphone already in their pocket, they connect to the FloraNet Wi-Fi network and scan the QR code on the node. This brings them to our intuitive messaging web application in their browser where they can send and receive messages with other Petal nodes or emergency services using LoRa ("Long Range") radio. FloraNet is a completely off-grid and user-friendly wide area communication solution that works with any smartphone, installed for a fraction of the cost of a cell network upgrade. The FloraNet design assumes the user knows very little about how the network works and tries to make it as easy as possible to access it. Our design also transfers the cost of backcountry communications to the business or organization that maintains and profits from the wilderness area. The messaging app user interface is simple and the layout is intuitive so people will instantly know how to use it. Reducing the complexity, cost, and inconvenience to the customer makes FloraNet a great product.

Current Options

There are two main categories of backcountry communication products currently available on the market. The most popular option, especially for "hard-core" adventurers, is a satellite transceiver that uses satellite to send messages from anywhere in the world. These are quite expensive, but they are worth the money for people who spend a lot of time outside of cell service. The other option is a "DIY" approach like Meshtastic, which is a LoRa-based mesh network, similar to FloraNet. These networks are usually run by enthusiasts and require every user to have their own device to participate in the network. The devices are cheap compared to a satellite device, but they require a lot of technical know-how to

configure and use. The user interface is also quite unwieldy, even for users who know what they are doing. These products are good, but they are not accessible for everyone.

Satellite Transceivers

Satellite transceivers like the Garmin InReach (Figure 3) are handheld devices that provide full coverage anywhere in the world.

Figure 3
Type: *Image*
Source: *inreach.jpg*
Caption: *Garmin InReach satellite transponder and the associated subscription services.*

Users can send and receive text messages to phone numbers and other InReach devices or press the SOS button to send their GPS location to the Garmin Response Center in Texas where they'll coordinate a rescue with the emergency services closest to your location. These devices are highly effective but terribly expensive. Their upfront cost can be up to $700, plus a monthly subscription to access the satellite network. Because rescues are coordinated by the Garmin Response Center, rescues take a lot longer compared to contacting search and rescue directly with a cellphone. Satellite transceivers are a good product for the hardcore adventurer, but their high cost and the annoyance of bringing another device means they aren't good for everyone or every situation.

DIY Mesh Network

Another option is a mesh network (Figure 4) like Meshtastic, an open-source digital radio enthusiast project that provided us with inspiration for FloraNet. Like FloraNet, a Meshtastic network is made up of Meshtastic devices that communicate using LoRa radio. Each device is a node that users can send text messages from.

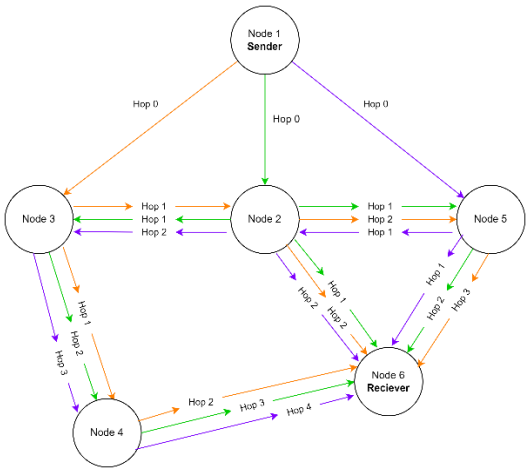

Figure 4
Type: *Image*
Source : *mesh-hopping.jpg*
Caption: *A message from Node 1 to Node 6 "hopping" through a mesh network. Every message re-broadcast is labelled with a hop number. The mesh network structure provides many different paths for the message to take to the receiver, so each colour represents the path a message could take based on which node receives the original transmission.*

The user connects to the device with their smartphone via Bluetooth using the Meshtastic app from their app store. In the app the user can configure the device and exchange text messages with other nodes. When the user presses send, the text message is broadcast using LoRa radio. Any nodes nearby that receive the message will then repeat the message. The message will "hop" from node to node through the mesh network until it reaches its desired destination. Users can purchase a Meshtastic-compatible device for as low as $15 and set it up using the guide on the Meshtastic website.

While Meshtastic is cost-effective, its not a perfect replacement for a satellite transceiver. They only work when all users have a Meshtastic device and the app pre-installed. Moreover, device setup and configuration requires a decent amount of tech savvy and digital radio knowledge. The complex device setup and the inconvenience of needing both a smartphone and a LoRa device has prevented Meshtastic from becoming a viable option for the average consumer.

Please place the following section as a highlighted side bar as a sort of "FYI" explaining how LoRa works

LoRa is a low power, long range digital radio technology initially invented for battery-powered Internet of Things sensors to send data to a master controller. LoRa sends digital 1's and 0's over the airwaves in the 915MHz ISM band in North America using a special way of encoding the digital data (called *modulation*) that uses less power and has a longer range than cell service or Wi-Fi. This is perfect for battery powered devices in outdoor situations. The trade-off for the range and power advantages is a much lower throughput (between 90 bits per second and 20 kilobits per second depending on the desired range), so LoRa only supports text messages up to 255 characters long.

End of LoRa FYI Section

FloraNet Ticks All the Boxes

FloraNet attempts to fill the niche between Garmin InReach and Meshtastic for the casual outdoor adventurer in the places they visit the most. Imagine you and a friend are skiing at a resort with a FloraNet installation. If you were injured outside of cell service, your friend could go to the nearest Petal Radio (a few hundred metres away at most) and send an SOS message to ski patrol using the smartphone already in their pocket. Based on the node they send the messages from, ski patrol will immediately know where you are, and your friend can stay with you until rescue services arrive. Response time will be much faster than if you used a satellite transceiver or if your friend had to ski back to the resort to flag ski patrol down. FloraNet moves the cost of emergency communications from the individual to the community in popular recreation areas, is accessible by anyone who knows how to connect to a Wi-Fi network, and does not require a separate device, user account, or pre-installed third-party app.

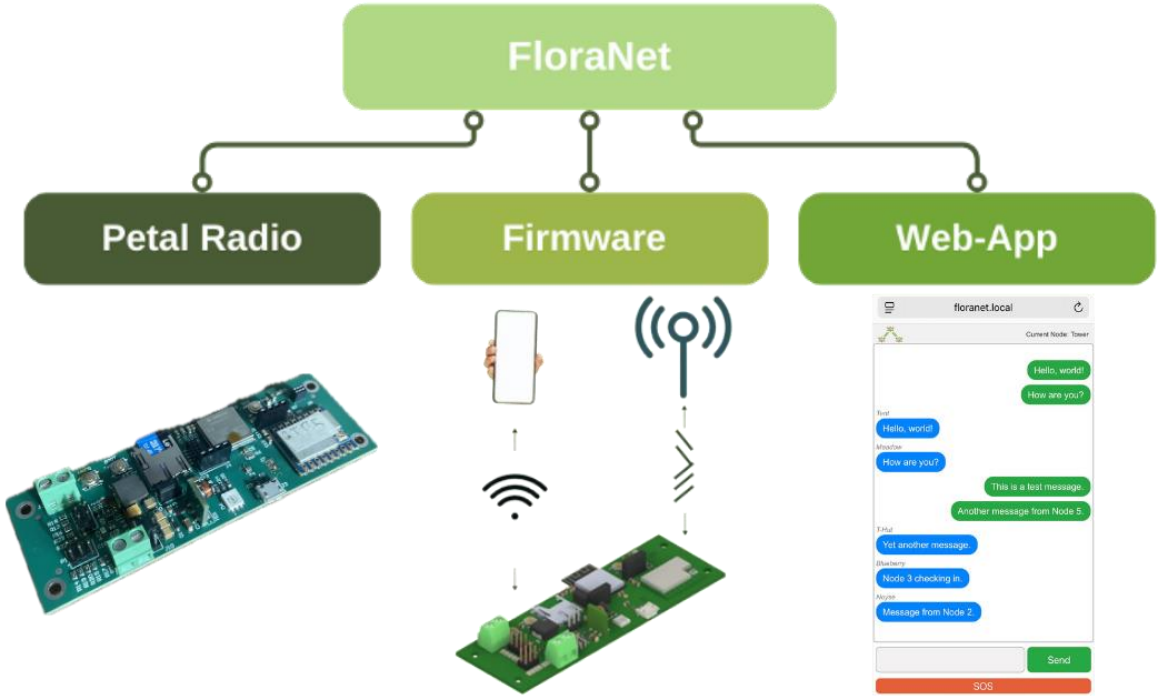

Figure 5
Type: *Image*
Source: *hierarchy.jpg*
Caption: *FloraNet is made up of the hardware we designed called the Petal Radio, the firmware we wrote using FreeRTOS, and the web app we created using JavaScript, HTML, and CSS.*

Bringing FloraNet to Life

Starting in September and finishing in December of 2024, we worked seven days a week to bring FloraNet to life. We started by creating a requirement specification sheet to guide our work. The first of major three major tasks (Figure 5) was designing, assembling, and testing the Petal Radio PCB. We then coded custom firmware for the Petal to interface between the web-based user interface and the rest of the LoRa mesh network. In the meantime, we learned JavaScript, HTML, and CSS web development over the course of a month to build our intuitive messaging web app from scratch. We carefully documented our progress and work over the course of the project, making the hardware files and the code available through our **Text:** GitHub **Hyperlink:** https://github.com/flora-comms/flora-comms along with a **Text:**

Wiki **Hyperlink:** https://github.com/flora-comms/flora-comms/wiki so that users can easily set up their own FloraNet.

Petal Radio

Petal Radio is the PCB prototype for our project (Figure 6). We wanted a board that would work reliably so we could focus on developing firmware and software. It has primary and backup power supply units (PSUs), a microprocessor unit (MPU), and a LoRa transceiver module. We designed it in Altium Designer with feedback from our professors and had it manufactured by JLC PCB in China.

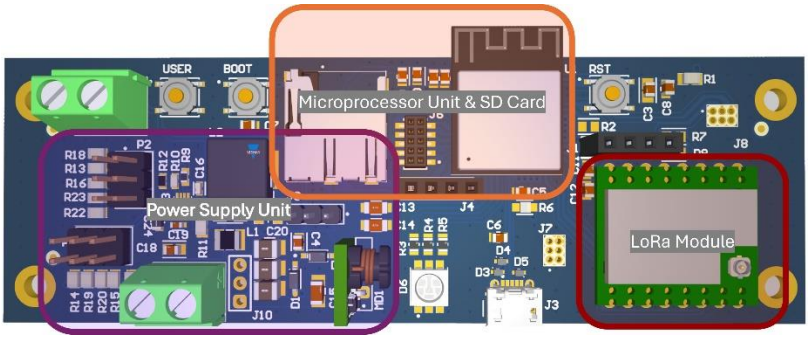

Figure 6
Type: *Image*
Source: *petal-layout.jpg*
Caption: *A 3D rendering of the Petal Radio with important sections highlighted and labelled.*

Power Supply Unit

Since the Petal will run on battery power in the field, we prioritized efficiency when selecting a power supply. We also wanted a low voltage disconnect (LVD) to protect the battery from excessive discharge. We included both a primary and backup PSU, and our testing pitted these two against each other to determine the best option for future versions.

The primary PSU is a Texas Instruments TPS62933P switching power supply IC. Based on our research, this PSU is relatively cheap, offers the highest efficiency, and has a built-in LVD. It can handle anywhere from 3.8V to 30V input with a 3.3V output. The only drawback was that we had to design the complex buck converter and electromagnetic interference filter circuitry around the IC. If our power supply design was faulty, this would render our board useless and prevent us from testing our other circuitry, so we included a backup PSU.

The backup PSU is a Texas Instruments TPSM84203EAB switching power supply module. It is more expensive than the primary and does not have a built-in LVD, but it comes with all the circuitry pre-assembled on the module, so it is plug-and-play with three through-hole pins.

We tested both PSUs for efficiency, ripple voltage, and transient load response. The results showed that the primary PSU circuit outperformed the prefabricated model in all our tests, and the LVD circuit worked better than expected. All of the testing data is available in our **Text:** GitHub **Hyperlink:** tbd 😊.

Microprocessor Unit

The MPU is the heartbeat of the Petal Radio. We chose an ESP32-S3-MINI module, based around a dual core 240MHz ESP32-S3 processor and includes a built-in Wi-Fi modem and antenna that can act as a Wi-Fi access point (AP). The ESP32 family of processors has an active developer community with plenty of

open-source libraries available on GitHub for embedded applications, and its sleep routines and power consumption capabilities lead the industry for battery powered devices. It interfaces with our LoRa module and an SD card through two separate SPI busses and serves the web application using our firmware. The SD card is on the board to hold the web application files and chat history logs that are too large to hold in the processor's internal flash memory. Having a removable SD card also makes it easy to test and modify the web application files without reprogramming the firmware. The MPU is programmable through a micro-USB port, but we also included pin headers for the UART serial pins in case the USB port doesn't work.

LoRa Module

The LoRa module converts the digital messages from our users to radio waves that can be sent long distances. The LoRa module we chose is the WAVE-Core1262-HF module from WaveShare Electronics. It uses the Semtech SX1262 LoRa transceiver for the 915MHz ISM band in North America, which is the current industry standard for IoT LoRa devices. Because RF PCB design requires money and time (two resources we don't have), we used a module that included the necessary RF circuitry despite being more expensive than designing the circuit ourselves. It communicates with the MPU using a SPI bus. The LoRa module interfaces the Petal Radio with the rest of the mesh network.

Firmware

The firmware has three main responsibilities: serving the web application, interfacing with the LoRa mesh network, and managing the device's power consumption. We wrote the firmware in C++ with FreeRTOS, a common embedded operating system, using the PlatformIO Arduino development framework in the VS Code IDE (Figure 8). Each responsibility is a *task,* meaning FreeRTOS can run these processes *asynchronously* (independently) of each other. FreeRTOS also provides *queues* and *event groups*, which are methods for tasks to communicate with each other and cooperatively share memory. We use queues to pass data and an event group to pass flags between the tasks (Figure 7). The FreeRTOS kernel provides all the tools we need for FloraNet.

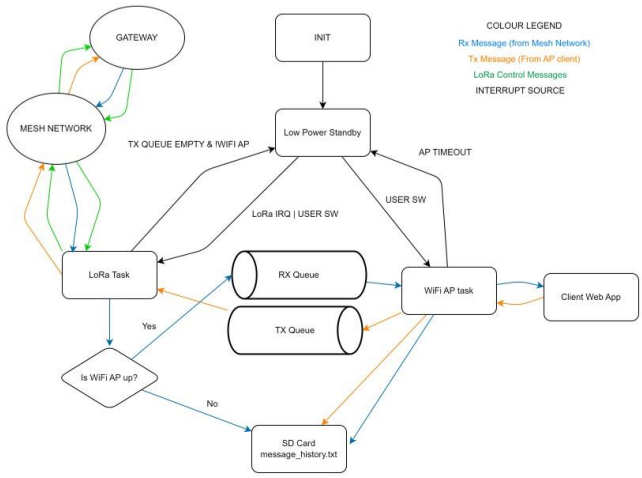

Figure 7

Source: *firmware-block-diagram.jpg*

Caption: *The firmware block diagram showing the LoRa task, web server task, queues, basic logic, and data flow. The rectangular blocks represent firmware tasks or files, the diamonds represent decision points, the circles represent processes external from the firmware, and the arrows represent data.*

The web server task provides the user interface to the client. To serve the web application files from the SD card, we used the popular *ESPAsyncWebServer* library from GitHub user *me-no-dev*. When the user presses the Petal's activation button, the firmware turns on the Wi-Fi network for the user to connect to. When the user connects and scans the QR code or navigates to *floranet.local* in their browser, a DNS server spins up an AsyncWebServer instance and serves the JavaScript, HTML, and CSS from the SD card using HTTP. The client JavaScript upgrades the HTTP connection to a WebSocket, allowing communication in both directions between the client and server. When a client sends a message, the web server task sends it through the "transmit" queue to the LoRa task and sets a flag in the event group to indicate a new message needs to be transmitted to the mesh network. Likewise, when the LoRa task sets a flag in the event group to signal a message has been received from the mesh network, the web server task reads it from the "receive" queue and sends it to the client web app. The web server task is the gateway for the user to the rest of the network.

Figure 8
Type: *C++ code*
Source: *main.cpp*
Caption: *FloraNet initialization code starting the web server, LoRa, and power management, tasks.*

The LoRa task sends and receives messages between the client and the rest of the network. After receiving a message through the queue from web server task, it does some checks to prevent the Petals from talking over each other before creating a 255 byte packet and sending it to the LoRa module for transmission. When not transmitting, the firmware puts the LoRa module into receive mode. Upon receiving a message, the firmware checks that the message hasn't been heard previously and re-broadcasts it if necessary. Every unique message is stored on the SD card so that users can view the chat history in the web app. The LoRa task makes the mesh network robust and reliable.

The power management task monitors the web server and LoRa activity and decides when to put the MPU to sleep to save power. Transmitting a Wi-Fi signal draws a significant amount of power, so we designed the power management task to turn off the web server and Wi-Fi AP after five minutes of inactivity. The firmware puts the Petal into a light sleep mode, cutting power consumption over 90%. When the activation button is pressed or the LoRa module receives a message, the power management task wakes up the processor and signals the appropriate task to deal with the wake-up event. The power management task greatly improves the Petal's efficiency, reducing the necessary battery and solar power requirements which saves money and preventing power outages which increases network reliability.

Web App

Our web app (Figure 9) is a user-friendly messaging app written in vanilla JavaScript, HTML, and CSS where users can post messages or SOS alerts to a public chat box visible to all Petals on the network. The layout is simple and intuitive, inspired by common smartphone messaging platforms. The top right corner shows the user which Petal they are connected to. The sent and received messages are displayed on the screen in the order they are received, with messages sent by the node the user is connected to on the right of the screen in a green bubble, and messages received from other nodes on the left in blue bubbles. Each message is stamped with the node they came from. Along the bottom of the screen, there is a text box to enter messages into, a green Send button, and a large SOS button. The familiar and simple layout means that our web app should be accessible by anyone, regardless of language, in an emergency.

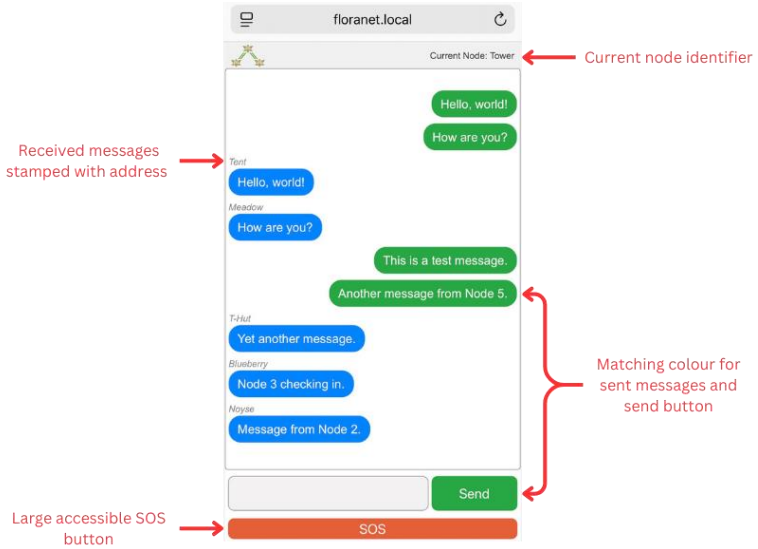

Figure 9
Type: *Image*
Source: *web-app-layout.jpg*
Caption: *The FloraNet messaging web app rendered on a smartphone using Safari browser.*

True Product Value

We designed the Petal Radio, firmware, and web app with the customer in mind, aiming to reallocate cost from the individual to the community, minimize product complexity, and facilitate customer adoption, all while providing a useful service. As inventors, we hope FloraNet inspires you to invent customer-focussed and accessible products. As enthusiasts, we hope FloraNet reminds you what you should expect from the companies that push the boundaries of technology.

Photo sources

https://www.garmin.com/en-CA/p/765374

https://www.garmin.com/en-CA/p/837461

Resources/Keywords
Meshtastic
FreeRTOS
LoRa
ESP32
Altium Designer
JLC PCB
ESPAsyncWebServer
JavaScript
HTML
CSS
Texas Instruments
Switching Power Supply
Buck Converter
HTTP
WebSocket
PlatformIO
C++
VSCode

https://meshtastic.org/docs/overview/radio-settings/#presets

https://github.com/me-no-dev/ESPAsyncWebServer/tree/master/src

https://www.freertos.org/Why-FreeRTOS

https://platformio.org/

https://github.com/espressif/esp-idf/tree/v5.2.3/docs/en/api-reference