# *Integrating Modularity for Mass Customization of IoT Wireless Sensor Systems*

Derek D'Alessandro
*Department of Engineering Systems and Environment*
*University of Virginia*
Charlottesville, USA
djd3ca@virginia.edu

William Gunderson
*Department of Engineering Systems and Environment*
*University of Virginia*
Charlottesville, USA
wpg5yy@virginia.edu

Ethan Staten
*Department of Electrical and Computer Engineering*
*University of Virginia*
Charlottesville, USA
eas5ke@virginia.edu

Yann Kelsen Donastien
*Department of Electrical and Computer Engineering*
*University of Virginia*
Charlottesville, USA
ykd3rd@virginia.edu

Pedro Rodriguez
*Department of Electrical and Computer Engineering*
*University of Virginia*
Charlottesville, USA
per8hx@virginia.edu

Reid Bailey
*Department of Engineering Systems and Environment*
*University of Virginia*
Charlottesville, USA
rrb5b@virginia.edu

*Abstract*—**As data collection and analysis grows in demand across a diverse spectrum of industries, data is collected from many sensors at different ranges with different quantities and types of data. One general approach taken by commercial firms to integrate wireless sensor data is to develop proprietary "ecosystems" of products; home automation companies like NEST, home security companies like SimpliSafe, and agricultural companies like Davis Instruments each require that customers use their hubs with their peripheral sensors. The work in this paper applies a flipped approach where a heterogeneous set of sensors from a range of suppliers connects to a hub over a variety of wireless protocols. The design of the hub, therefore, needs to easily accommodate a wide range of communication and wireless protocols. The focus of this work is on exploring how modularity can be designed into the architecture of a product to facilitate quick and low-cost customization of the hub to a particular need.**

**This particular work focuses on designing such a hub for various low-power wide-area network (LPWAN) applications. LPWANs are technologies and protocols that have longer ranges and lower power usage than higher bandwidth protocols like Wi-Fi. LPWANs, like LoRa, specialize in applications where many sensors are distributed over larger distances and, due to the small amounts of data they intermittently send, require less power. This modular hub needs to be able to recognize the type of radio connected to it and the type of communication (I2C, SPI, UART) used by the radio. Such recognition will enable variable quantities of different radios to be connected to the hub without significant redesign of the electronics or the firmware. Furthermore, the housing for the hub needs to be sufficiently modular so that any radio could be inserted without requiring a new design. Using custom components in only certain interfaces is central to the electronics design, and such modularity depends heavily on the firmware. With respect to the housing, a key trade-off for integrating modularity is accommodating variability in radios while maintaining ergonomic design. A key consideration in both housing and electronic design is incorporating modularity only where needed, and creating components in-house when necessary.**

***Keywords—modularity, wireless, Internet of Things, LPWAN, wireless sensor network***

## I. INTRODUCTION

The Internet of Things (IoT) can be described as the network of every physical device that connects to the internet and can process, store, and communicate data [1]. At the end of 2020, the IoT market had over 35 billion devices connected, and continues to grow largely due to its capabilities for collecting data and gathering insights [2]. Applications for IoT systems now include smart cities, agriculture, banking, and healthcare, each with its own needs.

The traditional approach to developing deployable systems for a particular application has been to create a custom design from scratch. The in-depth knowledge and design work required for such custom work takes a long time and consequently is costly. The upside is that it can deliver a system that exactly meets the needs of a use case. At the other end of the spectrum, recent solutions to the growing demand for IoT systems include integrated proprietary "ecosystems" of products. These products, such as Nest for home automation or Davis Instruments for agriculture, work with a variety of sensor types and are easy to set up for end consumers with minimal technical knowledge. Every part of these systems is centrally designed by the same company so that the parts can all work together seamlessly. Their drawback is the lack of customizability to fit specific use cases. If a use case requires a specific type of sensor, number of sensors, or communication protocol that the proprietary product ecosystem does not accommodate, then the user is out of luck.

The gap between these two extremes (see Fig.1) is the opportunity space where this work focuses. In particular, we explore how modularity can be used to provide more customizable solutions than proprietary IoT product ecosystems while not requiring custom from-scratch design work for each new application.

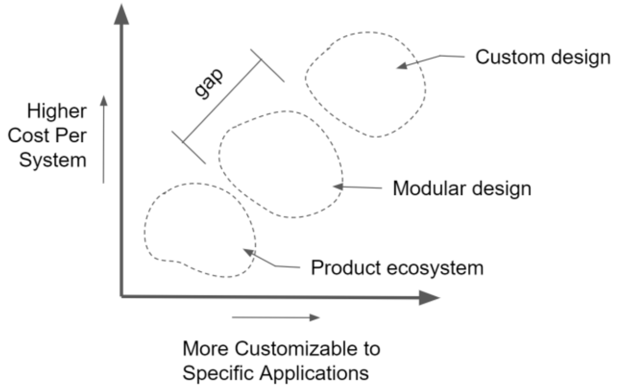


Fig. 1. Role of Modularity in Bridging the Gap

In this paper, we focus on a case study to identify general principles for how the mechanical hardware, electronics hardware, and firmware of IoT wireless systems can use modularity to allow for mass customization and create an optimal solution.

## II. CONCEPT OF OPERATIONS

### A. *General Use Case*

In the use case explored in this paper, multiple sensors communicate using different wireless protocols, gather different types of data at different frequencies, and send that data to the internet. We are focusing on a star architecture where the sensors communicate with a central hub on its way to the internet. This architecture accommodates a large number of sensors by channeling their connection to the internet through a single hub instead of having each sensor connect directly to the internet. The general use case is being explored as a case study within the broader space of IoT because it focuses on the common IoT tasks of gathering disparate data and getting it to the internet.

### B. *Concept of Operations for a Modular Hub*

An ideal hub can be customized to work with different sensors using different radio protocols (e.g., LoRa, BLE, Z-wave, Zigbee) to collect different types of data at minimal cost. One way to accomplish this is for the hub to be built with a large range of radio protocols. Such a hub would be overdesigned for most use cases as only a subset of the radio protocols would be needed. It would be better if only the radio protocols needed for a specific application could be easily installed. The concept of operations for such a hub would be to install specific radios during manufacturing. The radios would be attached to a core board that is common across all variants of the hub. The different types of sensors would be set up in the field (not during manufacturing).

The concept of operations for this hub is in contrast to use-case specific hubs that only work well in one situation (e.g., one type of radio, one type of data). Easy adjustment between use cases requires minimal to no design work; instead relying on modified manufacturing (which radios to include) and installation (setting up different sensors). The hub could be placed in both indoor and outdoor settings, have flexible attachment types, and fit a variable number of radio modules within its housing.

### C. *Current Products for the General Use Case*

Given the centrality of hubs that connect disparate sensors to the internet for this use case, we are focusing our exploration of modularity on sensor hubs. Currently, sensor hubs exist in a variety of forms, but no currently available hub works with a large variety of off-the-shelf sensors and communication protocols. Most work exclusively with proprietary sensors and/or communication protocols that do not always align with consumer needs. For example, the Thingenix SensorHUB solution is only compatible with that company's sensors, which limits the capabilities of that hub to the quality and type of sensors that are offered [3]. Another is the CHESTER Hub, which has an option of many LPWAN protocols, but is limited to one protocol per hub, which can limit the verticality or addition of sensors in its integration [4]. Other companies, like NEST, offer customizable IoT systems, but these systems are not standardized to the point where they can be integrated into current user systems. They work strictly within their own ecosystem of products and cannot expand to utilize others. While these solutions demonstrate modularity to an extent, they do not exhibit the same mass customization that one of these systems could truly benefit from.

## III. LITERATURE REVIEW

On a broad scope, modularity is defined as how replaceable different components of a system are [5]. More specifically, in the context of integrated IoT systems, modularity is the flexibility in design for configuration of IoT technologies[6]. For a company trying to reach new markets, modularity allows variety in products and improves speed of introduction of new products [7]. It also can benefit in terms of testing, as it can be tested in subsystems through their interfaces as opposed to testing a system as its whole [7]. Modularity not only helps the consumer when it comes to customizable solutions, but it helps engineers save development time when they have a referenceable design that can be converted into custom options.

There are a number of different types of modularity that are able to be effectively utilized in an IoT system [8]. These types of modularity are differentiated by the way the component interactions are organized [9]. The first type is bus modularity, which utilizes a bus to which all the other physical components are connected with the same type of interface [9]. This type of modularity is valuable in that it, in theory, allows for variation in both number and type of components in a system, whereas other types of modularity only allow variation in component type [10]. The second type of modularity is slot modularity, which utilizes different types of interfaces between all the components and the base, so they cannot be interchanged [9]. This is useful in an IoT system in that all the components share a universal base, despite having unique interchangeability on an individual part basis [11]. The third type of modularity is sectional, which does not consist of a base module, but consists of unique components connecting to each other via identical interfaces [9].

While useful in many cases, modularity has its weaknesses as well when compared to standalone integral architecture systems [7]. One weakness is that different iterations of modular designs can appear identical. For example, if a design has three Bluetooth modules in it, the common base design of the housing makes it hard to tell it apart from a hub with three LoRa modules. Another weakness is that modular design can reduce product performance. An integral hub for a single use case would usually have better performance since all the development characteristics like size and interfacing can be optimized for one purpose. The goal in this paper, however, is to explore how much can be gained in terms of design flexibility while mitigating potential drawbacks of modularity.

## IV. FROM GENERAL TO SPECIFIC: EXPLORING MODULARITY FOR SPECIFIC USE CASES

For the concept of operations, developing one product that has all of the wireless standards to connect to sensors would be overkill; most use cases would not use all of the wireless protocols. Developing a unique product for each user would be cost prohibitive. In this study, we demonstrate how modularity can serve as the basis for using mass customization to allow each user to only have the wireless protocols they need without requiring a complete redesign effort for each unique instance. Exploring specific versions of the general use case highlights the challenges and opportunity of using modularity to fill the gap identified in Fig. 1. Here, we explore an agricultural use case and a smart city use case.

### *A. Agriculture Case*

A farmer has all her land to maintain and needs to track soil moisture, weather conditions, livestock location, and fence status. Each of these requires a different type of sensor (e.g., temperature, humidity, soil moisture, GPS, contact sensors). These sensors need to connect wirelessly from a distance, and can only communicate with their local protocols. The weather station sensors communicate with BLE, while the soil moisture, livestock and fence sensors communicate with LoRa given how far away they are from the hub. The farmer needs to see all the data and get regular updates on the status of her sensors. The hub that gathers data from all of the sensors is near the weather station and has a LoRa module and a BLE module. Since the hub interacts with battery driven sensors that have low power consumption, it minimizes the cost for the farmer while optimizing efficiency. Within the hub, a LoRa radio module connects to the core board via SPI while the BLE module connects with I2C. This hub is able to receive all the data from the sensors in the field and package the data as needed for a backhaul to a cloud interface the farmer can use to see her status.

### *B. Smart City Case*

The CEO of a medium-sized firm in Munich, Germany wants to test two proprietary sensors it has created for use in smart cities. One sensor aims to detect when sidewalks are being blocked by micro-mobility services like shared bikes and scooters. The second provides real time and predictive information on flooding. Since most of the firm's clients are in Europe, the CEO wants to test all of the sensors with a Sigfox module that communicates using UART. While the initial design from the agricultural example did not include a Sigfox module or UART, the same hub can be modified at the manufacturing stage. In this case, a singular Sigfox module is used instead of using a BLE module and a LoRa module. Each radio module slot can communicate over UART, I2C, or SPI, so the Sigfox module can be put in any slot. The firmware recognizes the type of communication automatically. Even with one less module, the housing can remain unchanged.

The specific use cases highlight the impact of modularity on the hub's housing, electronics hardware, firmware, and their integration. Modularity of radio modules impacts the housing in that the housing needs to enclose a variety of different radios. It impacts electronics hardware through the physical connection of radios to the core board, type of communication protocol, and power requirements of different radios. The firmware must be designed to accommodate different radio hardware and different types and frequencies of data. Findings related to how to design in such modularity are addressed in the next section.

## V. FINDINGS

A mass customizable hub would require flexibility to the type of radio and type of data necessary for a particular use case. This has implications across the entire design, including the core board that the radio modules connect to, the firmware on the core processor, the housing for the hub, and the interfaces between all these subsystems. Figure 2 shows an example system architecture that is referenced throughout this section. In this example, there are three radio module slots.

### *A. Implications for the Core Board*

The core board needs to have enough physical and embedded flexibility to be able to change radios seamlessly.

**Communication Protocols.** The design needs to accommodate radio modules communicating to the main processor with multiple different serial communication protocols, including UART, I2C, and SPI. This can be done with a minimum number of traces on the core board by utilizing bus modularity in sharing which traces could be used for each communication protocol.

**Physical radio connector.** Slot modularity needs to be a part of this by allowing multiple radio modules to connect with the same style of interface into the core board. Since there is no common standardization of physical interface types (e.g., mPCIe, M.2) among off-the-shelf products, modules can benefit from being designed by the same team that designs the hub as opposed to using off-the-shelf radio modules.

**Radio Voltages.** Multiple different radio voltages must be accommodated for (e.g., 1.65V, 3.3V, and 5V) to make the design accessible for different radio modules. There are many solutions for changing the voltages with the simplest perhaps being the inclusions of jumper switches.

**Power management.** Each radio module gets power through its connection to the power supply on the core board. The core board needs to be able to deliver sufficient power for three radio modules. This is an example where overdesigning is used to facilitate modularity; the power architecture will be oversized when fewer than three radio modules are installed.

### *B. Implications for Firmware*

The firmware needs to have enough libraries and functionality to act as an adaptive backbone for the design. The more that the firmware can do without modifications when radio modules or sensors are changed for specific use cases, (data recognition, communication protocol adaptability, etc.), the better the ability to facilitate modularity in the design.

**Radio Expectations.** The firmware should recognize which radio module and serial communication protocol (UART, I2C, SPI) is attached and adjust pin expectations for that communication protocol. This is an example of interactions between the firmware and electronic hardware that need to be modularized together. The wiring needs to be designed to accommodate different communication protocols as does the firmware to run those communication protocols. Because the radio modules will be installed during manufacturing, different firmware could be installed depending on which radio modules are attached to each slot.

**Data Types.** The firmware needs to be able to recognize the various data types that come from the sensors. These can vary from geospatial, to binary, to many other data types. Different data types can vary from user to user; the firmware needs to be written in a way that is robust to differences in data collected and/or allows a user to configure the hub to different data types.

### *C. Implications for Housing*

The housing needs to encase the entire system to protect it from weather and tampering, be user-friendly, and be adaptable to different radio sizes.

**Sizing.** Consistent housing achieves flexibility in the design as all the modules may be different sizes, but they all interface and fit in the hub in any place in the same way. However, the effect of modularity on the housing highlights one of the drawbacks of modularity. The housing needs to be able to accommodate three radios even if only one radio is installed. One way to do this is to manufacture multiple housing sizes and use the smallest one appropriate for a given radio configuration. The other is to make one oversized housing that can accommodate any radio configuration. Minimizing size for hubs like this is not a critical design goal. They are akin to Wi-Fi routers and exterior cable junction boxes on houses: smaller is better but a slightly larger housing is likely preferred over multiple housings to reduce costs associated with manufacturing, design, and assembly.

**Mounting.** The housing needs to be designed for multiple mounting scenarios depending on the use case. The goal is that the hub can be mounted to a surface with a variety of methods of adhesion or attachment. Integrating multiple mounting types (e.g., zip ties, screws, and adhesives such as Command Strips) provides the requisite flexibility. These options represent different types of modularity: screw mounts as slot modularity, zip ties (e.g., around a fence post) as bus modularity, and adhesives as sectional modularity. Together, they achieve flexibility and meet the goal of being able to mount the hub in many different places.

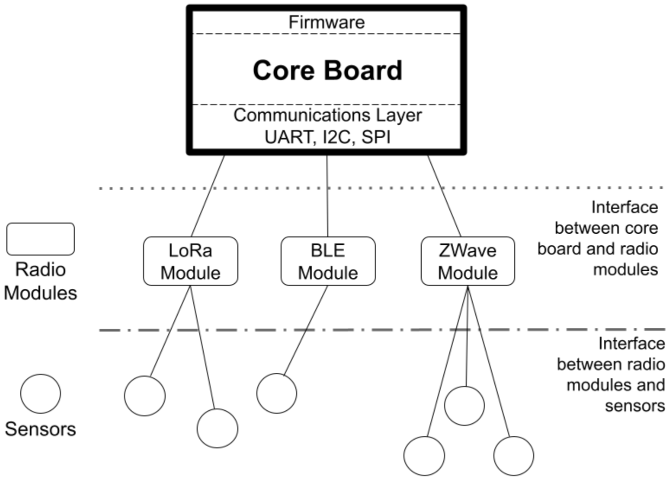
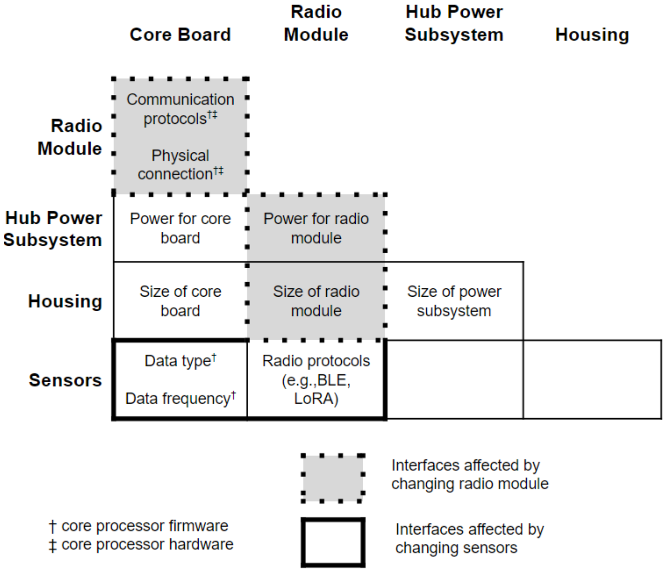


Fig. 2. Architecture Modularity Diagram

### *D. Implications for Interfacing*

In Fig. 2, the types of radio modules shown are just three examples of radios that could be used; they could be any radio type depending on the sensors used. The interface between the processor and radio modules includes the serial communication protocol(s), physical radio connector(s), and power management. As seen in Fig. 3, if changing out a radio module, the processor and associated firmware need to recognize the communication protocol, interface physically with the correct connection type, and provide the correct voltage for the new radio module. These changes are extensive, and require changes to both software and hardware. The interface between the radio modules and sensors predominantly rests in the wireless communication protocol used by both. If a sensor is changed out, there needs to be a radio module with the right protocol, and firmware that can accept the type and frequency of data from that sensor. These changes are limited and feasible in comparison to changing a radio module.



| | Core Board | Radio Module | Hub Power Subsystem | Housing |
|---|---|---|---|---|
| **Radio Module** | Communication protocols†‡<br>Physical connection†‡ | | | |
| **Hub Power Subsystem** | Power for core board | Power for radio module | | |
| **Housing** | Size of core board | Size of radio module | Size of power subsystem | |
| **Sensors** | Data type†<br>Data frequency† | Radio protocols (e.g.,BLE, LoRA) | | |

▨ Interfaces affected by changing radio module

☐ Interfaces affected by changing sensors

† core processor firmware
‡ core processor hardware

Fig. 3. Design Structure Matrix

Modularity needs to be cut off at a point of high standardization. The interface that has the least interactions with the rest of the system and thus has limited cascading design changes is a good candidate to separate parts designed in-house from off-the-shelf parts that might vary between applications. In this case, the interface between the radio modules and sensors is the preferred choice. Few if any design changes would be needed when sensors are changed if the firmware is written to accommodate a wide range of data. One of the reasons this interface makes a cleaner break between what to design in-house is that industry-common standards exist for wireless protocols (e.g., LoRa, BLE). The core board-to-module interface, on the other hand, is more integrated and relies less on standards. Designing the core board in-house while the radio modules are off-the-shelf would limit the range of radio modules that could be attached due to the lack of standards regarding the physical connectors. In addition, the power requirements of different radio modules might limit which radios could be used if the radio module is not designed in-house. This does not forego modularity in the radio modules or the main core board; due to complicated interfaces without common standards, designing both in-house makes it possible to integrate the modularity necessary to allow changing out radio modules.

## VI. CONCLUSION

This paper has highlighted opportunities for modularity in IoT to fill the gap between fully custom designs with infinite possibilities and closed product ecosystems that limit sensor choice. IoT products provide a particularly interesting case for examining modularity due to their integration of hardware and software. More specifically, this work seeks to offer insights through the case study of a sensor hub.

As with prior writings on modularity, one finding is that incorporating and recognizing opportunities for modularity not only within subsystems but also at the product architecture level is vital when designing a customizable system. Another finding that is more IoT-specific focuses on leveraging the decoupling afforded by highly standardized interfaces to decompose what is designed in-house from off-the-shelf components. Deciding across which interface to use fully off-the-shelf parts is key to gaining the most benefits from modularity. All three types of modularity identified by Ulrich – bus, slot, and sectional – have roles to play in integrating modularity into IoT systems. The main trade-offs identified involve overdesigning some aspects, such as the housing size. Using the IoT hub case study where minimizing size is not the most critical objective, such trade-offs may be reasonable to gain the ability to efficiently mass customize products to a range of applications.

As seen in the case study, the broad concepts of modularity can be applied to a variety of specific solution spaces and can be used to create technology that transitions between these use cases efficiently. This can benefit companies that are beginning to utilize IoT with larger audiences such as smart cities, smart farming, and healthcare. The findings were collected through the design and development of a modular hub style solution in combination with external research. Modularity is a means to an end for creating a system that can be adapted to many use cases, and should be considered at every step in the design process if the goal is to reach a broad audience.

## ACKNOWLEDGMENT

The authors all thank Adam El Sheikh and Joseph Carley for their contributions as team members in the Fall of 2020. We would also like to thank Michael Kelly, Colin Ulen, Alexander Young, Ali Slavin, Scott Downen for their steady support and advice over the last eight months. Finally, we want to thank Harry Powell and Todd DeLong for sharing their electrical device design knowledge with us.

## REFERENCES

[1] J. Clark, “What is the Internet of Things, and how does it work?,” *Business Operations*, 28-Aug-2020. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/. [Accessed: 02-Apr-2021].

[2] 2020 Jan 13, “The IoT Rundown For 2020: Stats, Risks, and Solutions,” *Security Today*, 13-Jan-2020. [Online]. Available: https://securitytoday.com/Articles/2020/01/13/The-IoT-Rundown-for-2020.aspx?Page=2. [Accessed: 05-Apr-2021].

[3] Thingenix, “Products and solutions for internet of things (iot).” [Online]. Available: https://www.thingenix.com/en/products. [Accessed: 01-Mar-2021].

[4] “Multipurpose IoT HUB CHESTER: Nb-iot HUB: LORAWAN hub: BY HARDWARIO.” [Online]. Available: https://www.hardwario.com/chester/. [Accessed: 01-Mar-2021].

[5] “What is modularity?,” 02-Aug-2019. [Online]. Available: https://www.computerhope.com/jargon/m/modularity.htm#:~:text=Modularity%20is%20a%20property%20that,upgraded%20without%20affecting%20other%20components. [Accessed: 01-Apr-2021].

[6] E. Ostrosi, S. Sagot, “Modularity and Configuration Applied to Product Integrating the IOT Technology” *27th International Conference on Transdisciplinary Engineering* DOI:10.3233/ATDE200121

[7] M. D. Rosenau and D. A. Cutherell, “Product Architecture,” in The PDMA handbook of new product development, New York, New York: John Wiley & Sons Inc, 1996, pp. 217–233.

[8] C. Heng, “Types of modular interface design,” *TechCoil*, 08-Dec-2017. .

[9] Ulrich, K. The role of product architecture in the manufacturing firm. Research Policy 24, pp. 419-440, 1995.

[10] (2000) BUS MODULARITY. In: Swamidass P.M. (eds) Encyclopedia of Production and Manufacturing Management. Springer, Boston, MA . https://doi.org/10.1007/1-4020-0612-8_110

[11] B. G. Friedman, “What is Modular Design?,” *Medium*, 05-May-2020. [Online]. Available: https://brettgfriedman.medium.com/what-is-modular-design-10d48920dbd4. [Accessed: 04-Apr-2021].