

1	Background	3
1.1	Abstract	3
1.2	Escaping the New Extractivism	4
1.3	Shifting perspective on the Stack	7
1.4	Staying with the Trouble	11
1.5	Network weaving	15

A THESIS ON NETWORK INFRASTRUCTURE
AND SELF-PRODUCTION

MSC INTEGRATED PRODUCT DESIGN

2022

TENDRIL

POLITECNICO DI MILANO

As data-driven decision-making has quickly become the de facto mode of governance of our contemporary, the strive for cost-effective optimization has led the ways in which we interact with our present to become excessively simplified and one-dimensional—deliberately obfuscating user insight for the only sake of vectoring data towards the most profitable solutions, in the fastest way possible.

As data-driven decision-making has quickly become the de facto mode of governance of our contemporary, the strive for cost-effective optimization has led the ways in which we interact with our present to become excessively simplified and one-dimensional—deliberately obfuscating user insight for the only sake of vectoring data towards the most profitable solutions, in the fastest way possible.

Contemporary user interfaces, both outside and within the context of labour, with their biased incline towards efficiency, manage to inscribe economic significance

to everything that attempts to dialogue with them—in this landscape, users function as exploited workers, resources to be extracted and products to be sold . . . If deemed unprofitable, users' faculty to exercise agency over their contemporary gets filtered out and removed by extractivist data funnels, which, in turn, get purposefully designed to do so more and more optimally.

Designed to fuel a global-scale push for capitalogenic automation that, intentionally or not, has come to replace politics, theory and philosophy, and with them, people's say on the way they are building the world of today.

Escaping this seemingly unavoidable trajectory is the core reasoning behind the development of this thesis and its results.

The research has been structured in three sections. The first one, the Background section, inquires into the theory that was required to synthesise a design approach capable of correctly defining all of this thesis' outputs. The second section harbours the outputs themselves—more precisely, The Tendril Repository collects every resource that is needed to design, manufacture and deploy a private meshnet for P2P messaging. Guiding people who decide to take on this project through a comprehensive learn-by-doing process that dives into network-related decision-making, hardware self-production and software configuration.

The exercise described in the Repository has the ultimate goal of allowing its users to reclaim their share of agency over the fruits of their labour, the data they produce and how this data gets used, not only offering them a figurative chance to escape the *new extractivism*, but also a deeply practical alternative to those standard network infrastructures through which the *new extractivism*

itself gets perpetrated. The third and last, the Development section, documents the development process of the Repository, going well into detail about all the decisions that have been made and the reasonings behind them, both in the field of integrated product design and telecommunication engineering—finally, concluding with a retrospective on the chosen approach and on its resulting outcomes.

1.2

Escaping the New Extractivism

New extractivism is a term coined by researcher and artist Vladan Joler in his self-published and homonymous cartographic project over at extractivism.online¹. With this project, Joler illustrates the *new extractivism* as a well-underway variety of world-scale labour exploitation that is indiscriminately involving everyone living in our contemporary, going well into detail into the allegorical concepts that make up its superstructural totality, as well as mapping how these concepts relate with each other.

In this publication, Joler states that in the information age, “everything becomes a potential frontier for expansion and extraction. From the depth of DNA code in every single cell of the human organism, to vast frontiers of human emotions, behavior and social relations, to nature as a whole—everything becomes the territory for the new extractivism. . . . at this moment in the 21st century, we see a new form of extractivism that is well underway: one that reaches into the furthest corners of the biosphere and the deepest layers of human cognitive and affective being.” (Joler, 2020)

In this context “As Shoshana Zuboff points out, surveillance capitalism renders behavior so that it can be parsed as observable, measurable units. Once it is rendered as behavior, it is turned into data. This is what she calls “behavioral surplus”. Since our bodies, minds, and behavior are one of the ultimate resources for the new extractivism, every segment of our existence can be seen as a form of direct or indirect labor producing data as a behavioral surplus. When we breathe, walk, or sleep, every single emotion that we feel, our attention, our body temperature, or diseases that we have—everything can produce a behavioral surplus if being captured by this giant surveillance apparatus. In that sense, even our bare existence can be seen as labor.”²

² Zuboff, 2019, as cited in Joler, 2020

This thesis argues that the newborn conditions of exploitation described in the *new extractivism* are a direct consequence of a world-scale disregard of

This thesis argues that the newborn conditions of exploitation described in the *new extractivism* are a direct consequence of a world-scale disregard of conventional politics in favour of much simpler data-driven forms of governance ushered by informa-



conventional politics in favour of much simpler data-driven forms of governance ushered by information age computation, further moving the hypothesis that these conditions could be ultimately overturned.

tion age computation, further moving the hypothesis that these conditions could be ultimately overturned.

Within companies, data-driven decision-making can be interpreted as a streamlining effort aiming to automate and accelerate management as much as possible, minimising error and narrowing down al-

ternatives in pursuit of the most profitable solutions. At the pace at which specific markets are moving, this process has become essential for their survival and, for all the professionals working in these enterprises, data-driven optimisation has become sovereign—so much so that information as an organising force, as McKenzie Wark states in her 2019 publication “Capital Is Dead: Is This Something Worse?”, is seeping into people’s worldview as a standard mode of technological governance, “What we think of as “technology” these days very often means technologies that instrumentalize information. These are specific kinds of apparatus that gather, sort, manage, and process information so that it can then be used to control other things in the world.”³

3 Wark, 2021

This is especially evident in the field of design, where these apparatuses, especially in the last twenty or so years, have been heavily integrated throughout design processes for converging towards the most cost-effective solutions in the fastest way possible with little to no squander of labour and time.

This striving for design-driven efficiency has come to permeate businesses and state-operated organisations, ushering the line between the private and the public to become increasingly hard to draw, and the relationships between the two more confusing to understand. American architect, urbanist and professor, Keller Easterling, in her homonymous 2014 publication, calls these relationships *extrastatecraft*: “a portmanteau describing the often undisclosed activities outside of, in addition to, and sometimes even in partnership with statecraft.”

As she talks about free economic zones as an example of *extrastatecraft* infrastructure, Easterling states that, “while promoted as relaxed, open, and free from inefficient state bureaucracy, the politics written into the zone’s spaces and activities often diverge from the declared intent. [the zone] is usually an isomorphic exurban enclave that, exempt from the law, can easily banish the circumstances and protections common in richer forms of urbanity. Labour and environmental abuse can proceed unchecked by political processes”; Easterling extends this concept way past urbanism, further saying that, “contemporary infrastructure space is the se-

cret weapon of the most powerful people in the world precisely because it orchestrates activities that can remain unstated but are nevertheless consequential. Some of the most radical changes to the globalising world are being written, not in the language of law and diplomacy, but in these spatial, infrastructural technologies.”⁴

4 Easterling, 2014

In a context freed from the burden of politics, the capital-driven motives that propel automation can persist in disregarding any form of social consciousness, allowing for labour abuse to run rampant. More precisely these motives are permitting organisations to mindlessly pit human labour against machine one, regardless of what that might entail.

In the eyes of capital-driven efficiency, there is no difference between human and machine work, and as one incrementally substitutes the other as a natural consequence of modernity, people lose agency over the decisions these global-scale *automata* take for the sake of profit.

This lack of agency has led these *automata* to become directly responsible for both the analysis of the contemporary and its reshaping, effectively becoming agents of both description and inscription of everything that is in and around them—a global-scale process of machine-learning that more often than not disregards people, ecologies and, generally, life.

This lack of agency has led these *automata* to become directly responsible for both the analysis of the contemporary and its reshaping, effectively becoming agents of both description and inscription of everything that is in and around them—a global-scale process of machine-learning that more often than not disregards people, ecologies and, generally, life.

For these global-scale apparatuses, everything appears as nothing but a resource to be extracted, and, as humankind’s agency over them dwindles, escaping this trajectory becomes increasingly difficult. In his 2017 publication “Futurability: The Age of Impotence and the Horizon of Possibility” Franco “Bifo” Berardi states that, “Future is not prescribed but inscribed, so it must be selected and extracted through a process of interpretation. The process of interpretation of the inscribed possibilities is enabled and shaped by concepts. The dominant code (the gestalt) forbids the vision and makes what is possible inconceivable.”⁵

5 Berardi, 2020

Built to exclude anything that is deemed unprofitable, these biased *automata* purposefully narrow down our capacity to envision alternative futures, not just metaphorically but, as contemporary sovereignty grows increasingly dependent on them, also practically.

“In recent decades, computation has grown to encompass a wide range of phenomena, reducing social life and human language to a determinist strategy based on a format of universal compliance . . .

What is inconceivable today is an approach to techno-power based on social needs rather than on economic realities.”

To not keep spiralling down our dystopias, design and, more broadly, labour is required to regain and protect its agency over the reality it shapes or, at least, retain the ability to question the objectives that propel production and governance—oppose the *dominant code* and turn the inconceivable back to possible.

To not keep spiralling down our dystopias, design and, more broadly, labour is required to regain and protect its agency over the reality it shapes or, at least, retain the ability to question the objectives that propel production and governance—oppose the *dominant code* and turn the inconceivable back to possible.

The outputs of this thesis are going to focus on finding a way to reclaim agency over contemporaneity and, as contemporaneity itself demands everyone to be a resource for data-driven extraction, these outputs, too, need to be just as inclusive.

From here on out, the people this research is aiming to reach will be referred to as *makers*, completely disregarding professional backgrounds, as, once again, they do not matter in the eyes of the *automata*—in this scenario everyone is an exploited creator that is always creating.

Works cited:

Joler, V. (2020). New Extractivism. Retrieved December 2, 2022, from New Extractivism website: <https://extractivism.online/>

Zuboff, S. (2019). *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. New York, NY: PublicAffairs.

Wark, M. (2021). *Capital Is Dead: Is This Something Worse?* London, England: Verso Books.

Easterling, K. (2014). *Extrastatecraft: The Power of Infrastructure Space*. London, England: Verso Books.

Berardi, F. (2020). *Futurability: The age of impotence and the horizon of possibility*. London, England: Verso Books.

1.3

Shifting perspective on the Stack

“Sometime from 1995 to 1997 or so, especially in academic design programs, software seemed to displace theory as a tool for thought. Many students interested in asking essential questions about how things work turned to software, not just to describe those things but also to make them, and not just to make them, but also to think through them. . . . as we quickly learn more precise and higher-resolution processes, it becomes correspondingly harder to see the whole at once. Accomplishments of analysis are paid for with a dissipation of synthesis.”⁶

After too, having well recognized how this loss of agency has been negatively affecting how we think of the world, design theorist Benjamin Bratton further states that “Today we lack adequate vocabularies to properly engage the operations of planetary-scale

6 Bratton, 2016

computation, and we make use of those at hand regardless of how poorly they serve us [further reinforcing the status-quo].”

Benjamin Bratton’s “The Stack: On Software and Sovereignty”, tries to tackle the aforementioned issue by helping us to better picture how exactly the advent of software impacted our contemporary. Citing vocabulary coming from computer science, Bratton provides design professionals with a readable model for visualising these new geopolitical architectures by describing them as part of a planetary-scale computational system—further stating that his publication not only functions as a map to navigate the Here and Now but also as a design brief for building a better one.

The *stack* framework subdivides reality into six layers, crucially disregarding any anthropocentric hierarchies and boundaries between the physical and the virtual, effectively observing our automation-ridden reality as it observes itself—something that Bratton believes to be of fundamental importance for priming any change within it, as it has become “necessary to move beyond a simple analysis of the relationship between an individual human being, their data and any single technology company to tackle the truly planetary scale of extraction.”

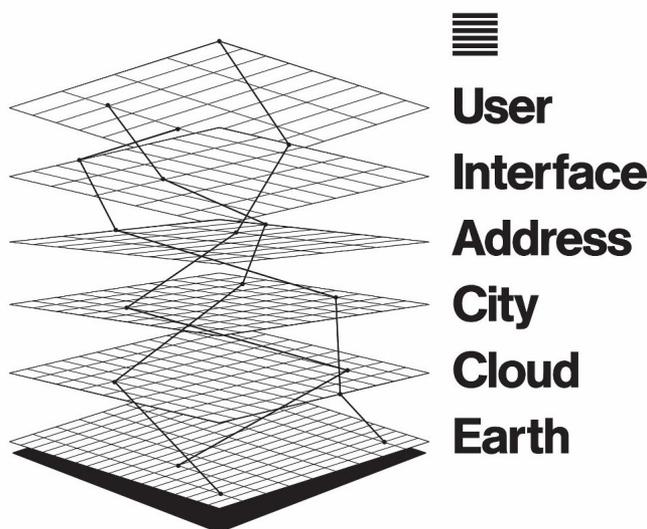


Diagram by Metahaven of the six layers of the *stack*, 2016

This mega-structure is not simply a void in which confines do not exist, but rather the accidental result of many boundaries being layered on top of one another during the course of history, meaning that, as much as the *stack* distorts the traditional logic of physical and political geography, it does not disregard them as non-existent.

The six layers are discretely re-designable and replaceable once they become obsolete and fundamentally comprise all the building blocks that constitute current global-scale sovereignty.

1. From the bottom, the Earth layer is the layer belonging to all the physical matter and energy required for planetary-scale computation including lithium, crude oil, sweat and blood.
2. In the Cloud layer, reside networks, borders and relationships of both state and extra-state apparatuses, one superimposed on top of one another without ever resolving in cohesive cosmopolitanism.
3. In the City layer, live both sensing and sensed entities: regardless of whether or not they are lifeforms, CCTV cameras or website trackers, everything in this layer acts as the eyes through which the *stack* perceives itself.
4. Once that happens, instances can be addressed in the Address layer, where pieces of information like identity documents, internet protocols and civic numbers are getting stored for later access.
5. In the fifth layer, the Interface one, live all the points of contact between two dialoguing complex systems, for example, a GUI between a human and his smart fridge. Interfaces within this layer, by drawing diagrammatic models of the whole structure in varying degrees of friendliness, allow users to interact with the *stack*, and the *stack* to interact with users.
6. Finally, the User layer, the sixth and last one belonging to this framework, hosts everything that is allowed by platforms to have sovereignty over them, regardless if it is a person, a tree in a forest or an algorithm in a self-driving car.

Unsurprisingly, according to Bratton, perspective is a crucial factor for the overall understanding of the *stack*. If observed one-dimensionally, this multilayered behemoth appears as nothing but an overlapping mess of incohesive borders, only once observed in its multi-dimensional entirety, the six layers can be fully unpacked and understood individually.

How we interface with the *stack* determines our degree of familiarity with it. Shifting perspective on the *stack*, revealing all of its six layers, could help *makers* reclaim agency over contemporaneity,

How we interface with the *stack* determines our degree of familiarity with it. Shifting perspective on the *stack*, revealing all of its six layers, could help *makers* reclaim agency over contemporaneity, as well as potentially freeing them from labour exploitation.

as well as potentially freeing them from labour exploitation.

Now, having framed this objective, how do we get there? What does this shift in perspective look like? To point us in the right

direction, we must look at what is preventing *makers* from seeing the wider picture, more precisely what strategies the extractivist *automata* engage to keep people's insight on the *stack* so strictly one-dimensional.

“With many decision systems within the contemporary moving towards automation, data-driven algorithms are assigned a central role within these systems to make decisions based on numbers. This evolving landscape of decision-making is hard to disentangle because many parts—the data sources, the algorithms, the processes—are deliberately kept secret and opaque.”⁷⁷

This purposeful attempt at obfuscating insight happens throughout all the points of interaction between people and the reality they live in, therefore, in all the touchpoints belonging to the Interface layer—the layer in which the *stack* gets exegetically modelled for its users to engage with. By design, extractivist models of the *stack* act as *allogregimes* that forcefully restrict user insight to a one-dimensional image of its much more complex totality. Extractivist user interfaces are often designed and built to be as simple and user-friendly as they can get—something which, for the sake of efficiency, is exceptionally important.

These interfaces act as funnels of value in which data extraction is facilitated to the very extreme, effectively inscribing economic significance to everything that interacts with them. “Your desire has to be parsed into a form a machine can understand; that is the job of this interface layer. The interface also positions you in relation to it, and to the rest of the stack, as a particular kind of subject: you are a user. . . . [a user-expressed desire] becomes a unique vector through a layered space that can fulfil an almost infinite number of desires, so long as they all take the form of a user asking an interface to satisfy a demand with a commodity. It does not really let you want or be much else.”⁷⁸

One-dimensionality is fundamental for reliably vectoring fluxes of data in directions that, for the *automata*, are deemed constructive. These funnels already know what they are looking for, the only thing that matters to them is keeping *makers* engaged enough for the extraction to happen painlessly.

This widespread adoption of user-friendliness, especially in the field of interaction design, has conditioned people to expect it

7 !Mediengruppe Bitnik, Silvio Lorusso, Domenico Quaranta, Luciana Parisi, Felix Stalder, and Janez Jansa, 2020

8 Wark, 2021

One-dimensionality is fundamental for reliably vectoring fluxes of data in directions that, for the *automata*, are deemed constructive. These funnels already know what they are looking for, the only thing that matters to them is keeping *makers* engaged enough for the extraction to happen painlessly.

everywhere, effectively disincentivizing *makers* from wanting to know more about the *stack*, widening the digital divide and removing agency from the objectives that propel it.

As the regime-like violence behind these processes is shrouded in seemingly harmless promises of inclusivity and accessibility, *makers* are being persuaded into giving up control of what they produce, effectively morphing into nothing but resources to be extracted.

If excessive simplicity seems to be the primary design-initiated cause of this loss of agency, perhaps embracing and acquiring a taste for complexity can help us prime a collective shift of perspective on the *stack*.

How could *makers* rediscover complexity in both a fulfilling and *stack*-comprehensive way? And, crucially, where and in what contexts would they be allowed to exercise this approach?

Works cited:

Bratton, B. H. (2016). *The Stack: On Software and Sovereignty*. London, England: MIT Press.

!Mediengruppe Bitnik, Silvio Lorusso, Domenico Quaranta, Luciana Parisi, Felix Stalder, and Janez Jansa. (2020). *Hyperemployment: Post-Work, Online Labour and Automation*. Roma, Italy: Nero Editions.

Wark, M. (2021). *Capital Is Dead: Is This Something Worse?* London, England: Verso Books.

1.4

Staying with the Trouble

On embracing intricacy in the face of apparent chaos, Donna Haraway's writing can guide us through a new and poetic onlook on contemporaneity that focuses on acquiring a Lovecraftian tentacular grip on reality. Haraway, in her 2016 publication "Staying with the Trouble: Making Kin in the Chthulucene" states that, in order to embrace complexity, people are required to remain in direct contact with the source of their concerns, as "staying with the trouble does not require such a relationship to times called the future. In fact, staying with the trouble requires learning to be truly present, not as a vanishing pivot between awful or edenic pasts and apocalyptic or salvific futures, but as mortal critters entwined in myriad unfinished configurations of places, times, matters, meanings."⁹

⁹ Haraway, 2016

Designing for the *chthulucene* to come means treasuring life's heterogeneous needs and perspectives, valuing *oddkin* above all by radically standing in direct contrast with the one-dimension-

Designing for the *chthulucene* to come means treasuring life's heterogeneous needs and perspectives, valuing *oddkin* above all by radically standing in direct contrast with the one-dimensionality of the capitalocene, as "living-with and dying-with each other potently in the *chthulucene* can be a fierce

ality of the capitalocene, as “living-with and dying-with each other potently in the *chthulucene* can be a fierce reply to the dictates of both Anthropos and Capital.”

reply to the dictates of both Anthropos and Capital.”

The radical starkness of this contrast raises the question of where could this tentacular approach to design even be exercised, as, in the eyes of the ever-present *automata*

this practice would be nothing short of frivolous.

Since the aim of this research is a collective reclaiming of agency over contemporaneity, as it is a struggle that indiscriminately involves everyone, the area in which *makers* can partake in this exercise cannot be confined to the realm of design theory—it needs to be inclusive and it needs to be accessible.

On this account, it becomes worth believing that, perhaps, the context in which *makers* might rediscover complexity could be something akin to the landscape of self-production, as the parallels that can be drawn between it and Haraway’s chthonic approach to design are many and quite strong.

Self-production can *stay with the trouble* by employing a bottom-up strategy for responding to a wide variety of needs and wants and, most importantly, doing so iteratively—meaning this kind of process is continuously intertwined with the issues it is trying to tackle and the goals it is trying to meet.

Self-production can *stay with the trouble* by employing a bottom-up strategy for responding to a wide variety of needs and wants and, most importantly, doing so iteratively—meaning this kind of process is continuously intertwined with the issues it is trying to tackle and the goals it is trying to meet.

This ability to stay with the present has been recently displayed during the early days of the Covid-19 pandemic, where digital fabrication filled in the gaps large-scale industrial and market-driven industries left wide open, as top-down governance failed to intervene with quite the same speed.

As designing in this space does not necessarily require market-driven motivation, there is plenty of time for people to engage in complexity at their own pace and for labour to exercise its fair share of agency over its outcomes, regardless of their nature—something that, sadly, has historically led design academia and design professionals to shun self-production in an attempt to delegitimize both its adoption and its adopters.

Regarding the struggle of legitimising this practice, wrote Enzo Mari, who, in his 2002 publication of “Autoprogettazione?” lamented a widespread misunderstanding of its original 1974 “Proposta per un’autoprogettazione” book and exhibition—a fundamental piece of design history that greatly influenced this research into the topic of self-production as a tool for emancipation.

In response to the lack of commercial success of some of his most cost-conscious furniture to date (e.g. the Day and Night Sofa), with “Proposta per un’auto progettazione” Mari envisions labour as a means to decondition people from the concept of “form as aesthetics’, which, according to him, was the reason his efforts were so far being rejected by his public.

The mixed reception that this body of work had in 74’, later led Mari to reprise the fact that his, wasn’t a hidebound attempt to celebrate the easy-go-lucky humbleness of craftsmanship, but rather a commentary on how the virtues of industrial-scale automation were being held back by market-driven trends camouflaged as design culture.

He, too, believed that automation could be an invaluable tool for cutting down on resource consumption, market costs, and, crucially, human labour—ultimately saving the proletariat from the fatigues of work.

He, too, believed that for people to appreciate this, they would require first-hand insight into the design processes he was, as a professional, already quite familiar with. To quote Giulio Carlo Argan, “Mari is right, everyone needs to design: as it is the only way to avoid being designed.”¹⁰

¹⁰ Argan, as cited in Mari, 2002



Pages from Mari's *Auto progettazione?*, 2002

Mari understood that self-production could be a valuable tool for understanding complexities that, otherwise, could never be under most people's control. He offered readers of "Proposta per un'autoprogettazione" a chance to stay intertwined with the troubles of design and production, some things that had never really impacted his mostly bourgeoisie public before, who sadly did not care for the insights they were being offered.

Of course, since this research is, too, attempting a similar approach to Mari's, it incurs the same risk of not reaching its desired public—since 1974, however, many things have changed and self-production has come to adopt a strategy to overcome this issue. By implementing an open-source logic of distribution, self-production in the information age manages to not only pull interest from a global-scale audience but also be persistent and reliable in its attempt to do so.

Platforms and communities surrounding self-production allow these ideas to survive way past their conception and iteratively evolve to stay up-to-date with the troubles they are intertwined with. Outputs become byproducts of commitment and as long as this commitment exists so will the stream of outputs.

Platforms and communities surrounding self-production allow these ideas to survive way past their conception and iteratively evolve to stay up-to-date with the troubles they are intertwined with. Outputs become byproducts of commitment and as long as this commitment exists so will the stream of outputs. Albeit this makes measuring the quantitative success of these solutions quite challenging, self-production, by *staying with the trouble*, does not require the public

to survive, moreover, it is able to wait for one and always be there in the moment of need.

For these reasons, in general, self-production could be a great playground for design research and, in particular to the objectives of this thesis, an ideal space in which *makers* could rediscover complexity, shift perspective on the *stack* and, ultimately, attempt to reclaim agency over contemporaneity.

If Mari's "Proposta per un'autoprogettazione" focused on building wooden furniture to change people's perspective on design and production, what could *makers* build to change perspective on all six layers of the *stack*? And, more precisely, is there something to be designed that would allow *makers* to exercise their agency over these layers?

Works cited:

Haraway, D. J. (2016). *Staying with the Trouble: Making Kin in the Chthulucene*. Durham, NC: Duke University Press.

Mari, E. (2002). *Autoprogettazione*. Mantova, Italy: Corraini.

Mari, E. (1974). *Proposta per un'autoprogettazione*. Milano, Italy: Centro Duchamp/Galleria Milano.

For changing *makers'* perspective on the *stack* and later allowing them to exercise agency over all six of its layers, maximising their amount of insight into them becomes fundamental. Combining cross-*stack* and layer-specific learning would enable people to engage in multiple T-Shaped rounds of development . . . Not to arrogantly strive for totality, but rather prime a tentacular curiosity for the Here and Now.

To figure out what *makers* could design for this exercise to be successful, it is necessary to pinpoint what is the true backbone of current full-*stack* governance and design—a question that can be quickly answered by recognising what historically primed the advent of the information age.

Originally developed for military application, network infrastructure from the '70s like ARPANET and, later in the '80s, NSFNET laid the foundations for concepts and technologies that would later be adopted in Web 1.0, evolving at the turn of the millennium into the full-fledged global scale network of Web 2.0.

“The old order would be swept away and a new day illuminated with the power of networks, iStuff, Twitter revolutions, “Internet freedom”, and smart cities.”¹¹

But, as the size of the network grew bigger, so did the economies that fueled its expansion. In a couple of decades, the internet quickly became one of the most promising frontiers for extraction, production and governance, a process that has eventually led capitalogenic automation to obfuscate people's insight into it and, more generally, remove their agency from the contemporary—morphing these networks from harbingers of utopia to servants of dystopia.

“The sky darkened, and now the Cloud portends instead state surveillance, tax evasion, structural unemployment, troll culture, and flash crashes.”

But, as Bratton also states, “the messianic effervescence of the former and the apocalyptic panic of the latter are part of the problem.” As “the official utopia and the official dystopia are not particularly useful frames of reference, . . . neither provide a robust and

intelligent program for art, design, economics, or engineering. . . . After the cycles of positive and negative hype run their course, [we are required, as professionals, and more broadly as labourers, to recognize] that computation holds both more potential and more risk than we foresaw. Going forward, we really do need new and better models.”

... After the cycles of positive and negative hype run their course, [we are required, as professionals, and more broadly as labourers, to recognize] that computation holds both more potential and more risk than we

11 Bratton, 2016

foresaw. Going forward, we really do need new and better models.”

we are required to not only keep well in mind the dangers of what we are working with but also refrain from losing hope in the virtues of automation as one of planetarity’s most invaluable tools.

On the grounds of this, it becomes worth believing that, perhaps, designing, manufacturing and deploying their very own network infrastructure could allow *makers* to untangle contemporaneity across its many layers and weave it back into something they can understand and govern—not only offering a figurative chance for escaping extractivism but also a practical one.

By focusing on the topic of network infrastructure *makers* can gain *stack*-comprehensive insight into the shape of information age products by intertwining themselves with the complexity that characterises them and, by owning the infrastructure they produce and the data that travels within it, they can exercise agency over their labour.

Just like Mari did with industrial production, to design a better *stack* from within it, and reclaim our agency over how we do so,

By focusing on the topic of network infrastructure *makers* can gain *stack*-comprehensive insight into the shape of information age products by intertwining themselves with the complexity that characterises them and, by owning the infrastructure they produce and the data that travels within it, they can exercise agency over their labour.

Given the scale of today’s internet, size has understandably become a powerful psychological and practical deterrent for wanting to learn more about its dynamics, meaning that to exercise a self-production approach to network infrastructure, *makers* require a *tabula rasa*, the privilege of a fresh start. To allow for this agency to be exercised, the outputs of this thesis will require just enough room for *makers* to experiment without compromising entry-level accessibility, effectively acting as an educative launchpad capable of gently leading them through all the motions of development as well as motivating them to push the project further. Now, what could this launchpad for network weaving look like?

Works cited:

Bratton, B. H. (2016). *The Stack: On Software and Sovereignty*. London, England: MIT Press.

MSC INTEGRATED PRODUCT DESIGN

2022

POLITECNICO DI MILANO

2	Tendrill Repository	21
2.1	What's the Tendril Repository?	21
2.2	What's a wireless mesh network?	21
2.3	What does the network allow me to do?	21
2.3.1	Features	21
2.3.2	Scenarios	22
2.4	How can I make one?	23
2.4.1	Designing the network	23
2.4.2	Manufacturing the network	33
2.4.3	Deploying the network	65

A THESIS ON NETWORK INFRASTRUCTURE
AND SELF-PRODUCTION

TENDRIL

The Tendril Repository is an entirely open-source collection of resources to design, manufacture and deploy your own ad-hoc wireless mesh network for P2P messaging, serving as a *full-stack* learn-by-doing experience aimed at regaining agency over the contemporary.

The repository functions as an easily approachable DIY project with comprehensive documentation on all of its phases, offering an iteratively scalable system for any further development without compromising beginner's accessibility. Go to the [Tendrill Repository GitHub](#)¹² page to download all the above-stated resources by clicking on Code, then Download ZIP.

12



A mesh network, meshnet for short, is a decentralised network topology in which there is no hierarchy between network infrastructure, meaning that *nodes* are allowed to directly transmit information to every other *node* on the network to ultimately cooperate in routing data between clients.

Contrary to standard internet infrastructure, meshnets can dynamically self-organise and self-configure, reducing overhead and improving fault tolerance—to put it simply, as every *node* plays a small part within the larger infrastructure, meshnets are cheap and can be built with incremental complexity while also displaying remarkable resilience in the eventuality one or more *nodes* should fail.

The ad-hoc wireless mesh networks the Tendril Repository allows you to build employ an upper layer that focuses on peer-to-peer messaging. Meaning that thanks to a physical layer composed of a varying number of long-range radio transceivers called *nodes*, it becomes possible for network clients to share text messages. Being the networks ad-hoc, they provide no on-the-grid connectivity,

so, as much as they cannot access the internet, they also do not rely on it. Clients, once they access the network by interfacing with a *node*, can begin sharing low-data packets of information with other clients currently connected to the same network, regardless of internet coverage.

To put it simply, by pairing the *nodes* with a companion app, you can send and receive encrypted messages from your phone, tablet or computer, to and from other networked devices.

2.3.2

Scenarios

An ad-hoc wireless meshnet for P2P messaging has the potential to give coverage wherever internet providers cannot or do not want to, as well as offer an off-the-grid and fully encrypted alternative to standard messaging solutions.

An ad-hoc wireless meshnet for P2P messaging has the potential to give coverage wherever internet providers cannot or do not want to, as well as offer an off-the-grid and fully encrypted alternative to standard messaging solutions.

Additionally, the physical layer of the network can be configured to meet demands that go beyond the topic of P2P telecommunication, as *nodes* can be programmed and customised to suit many other purposes, and even combined with different non-mesh-like network infrastructure to employ alternative upper layers. This flexibility carries throughout all the project's phases, as these networks can be designed, manufactured and deployed with incremental levels of complexity in each and every step.

The application scenarios for these networks are therefore many and entirely up to whoever decides to make one, greatly varying according to how much resources and effort gets put into them. To provide a couple of examples:

Pursuing Privacy—In this scenario, a meshnet gets designed, manufactured and deployed for fully encrypted messaging between two or more clients desiring undisturbed privacy when communicating with each other.

This application would be best suited for both journalists and activist organisations looking for a safe and under-the-radar alternative to normal messaging solutions.



Jakub Geltner's 2015 sculpture *Nest 05*

Filling Coverage Gaps—In this scenario, a meshnet gets designed, manufactured and deployed to curb internet providers' shortcomings in areas that lack any internet or cellular coverage.

This application would be best suited for hikers, skiers and other outdoorsy people looking to communicate with each other, camps, shelters and—in the unfortunate case they were required—emergency service units for rescue.



Picture by Lucia Galotto

Providing Fallback Infrastructure—In this scenario, a meshnet gets designed, manufactured and deployed to quickly intervene in the eventuality main infrastructure should fail.

This application would be best suited for extreme emergency situations in which the rapid deployment of a network could allow for critical communications to run unbroken, or at least until primary infrastructure goes back to working order.



Picture by Sadatsugu Tomizawa during the 2011 Tohoku earthquake

2.4

How can I make one?

2.4.1

Designing the network

In this phase—we understand how *nodes* communicate with each other and apply what we learned about network topology by designing a custom meshnet architecture.

Nodes are radio transceivers that make up the physical layer of the network and together define its topology.

Nodes to interface with client devices like phones, tablets and computers, use Bluetooth Low Energy (BLE), whereas, to transmit data between each other, employ LoRa: a radio transmission technique for LPWANs that operates within licence-free ISM radio bands. LoRa offers long-range connectivity up to a declared

maximum of 5 km in urban areas, and, provided there is a line of sight (LoS) between the *nodes*, up to 15 km in rural areas, achieving data rates between 0.3 kbit/s and 27 kbit/s.¹³

The number, position and class of *nodes* determine how large the network is, how many clients it has, and, ultimately, its purpose. By determining how many, their projected position, and the most suitable combination between their available classes, with *nodes* is possible to cover a wide array of use cases, as well as imagine new and exciting scenarios in which a custom meshnet could be useful. The network designing phase is divided into three steps: Numbering, Positioning and Classifying.

Works cited:

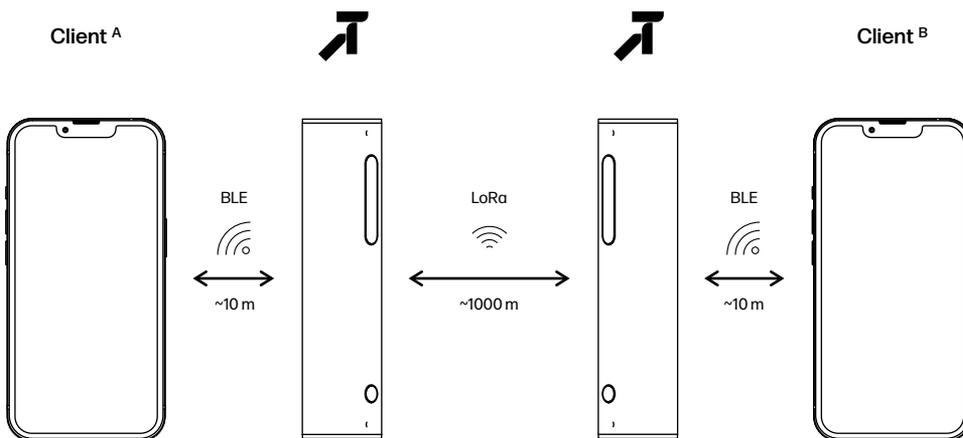
Semtech Corporation. (2022). LoRa and LoRaWAN: Technical overview. Retrieved from <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>

Numbering

In this phase—we determine the number of *nodes* in our custom meshnet architecture.

The number of *nodes* on the network affects two things: the number of active network clients and the overall network coverage.

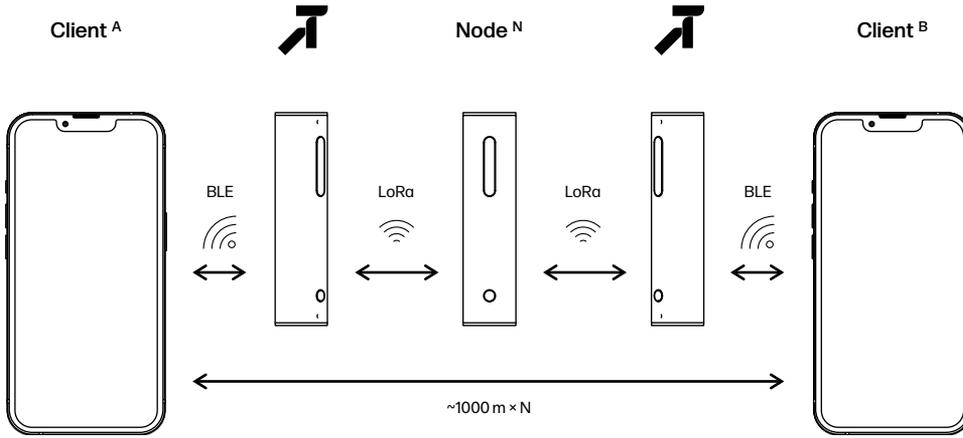
The number of active network clients gets affected by the number of *nodes* as the clients themselves are required to be near one to operate the network (within Bluetooth range), making it safe to assume the number of *nodes* always being greater than or, at least, equal to the number of active clients on the network. The overall network coverage gets affected as meshnet topology extends its range cumulatively—meaning that, given how *nodes* can cooper-



An example of messaging between 2 clients via 2 *nodes*

ate in routing data between clients: the more the *nodes*, the larger the network, and the further the data can travel.

For a network to function, the minimum required number of *nodes* is 2 and the maximum, as of 2022, is 32.

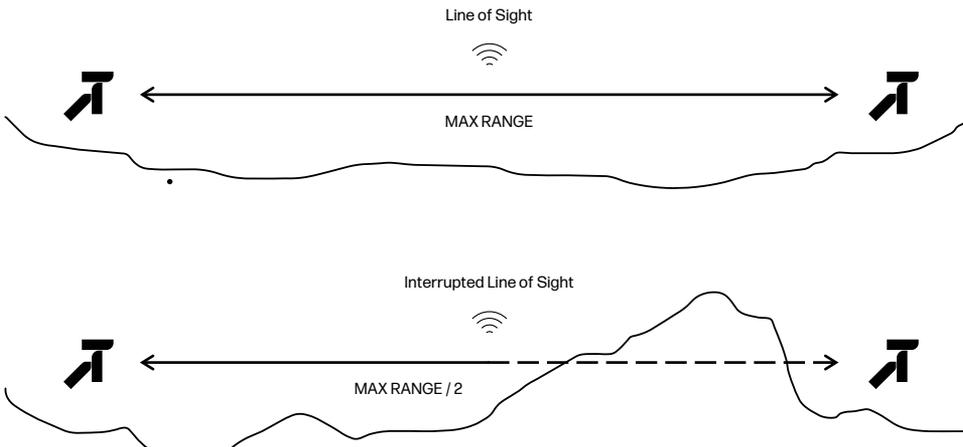


An example of messaging between 2 clients via an N number of *nodes*

Positioning

In this step—we determine how the *nodes* in our custom meshnet architecture could be positioned.

How *nodes* are positioned within the network fundamentally impacts its size, shape and degree of mobility.



An example between a clear LoS transmission and a, less ideal, interrupted one.

Nodes are all equipped with omnidirectional antennas—meaning that, in three-dimensional space, they display a radiation pattern that can be simplified to a sphere varying in size according to their position, as these spherical radiation patterns have a gradient of effectiveness that gets negatively affected by distance and interference (e.g. buildings, trees and terrain).

Cleverly positioning *nodes* in three-dimensional space for both maximising coverage and LoS between them, can therefore greatly affect the performance of a network.

Whenever it is not possible to do so, increasing the number of *nodes* can always help mitigate this problem.

Classifying

In this step—we determine to which combination of classes the *nodes* in our custom meshnet architecture belong.

Nodes are modular and, as of now, come in four classes that can be freely upgraded or downgraded from each other with minimal loss of parts.

Each of them tries to cover a different role within the network’s architecture, but how they will get employed, ultimately comes down to your preference and imagination.

*	Coverage is indicative, omnidirectional and with no LoS unless stated otherwise
**	Cost is indicative and excludes tooling, however, it includes a generic tripod for RES <i>nodes</i> and a generic drone for AIR <i>nodes</i>

****	Coverage with optional 3dBi SubG Antenna Radio Module
	USB-C powered

Coverage*:	~500 m
Cost**:	~25 \$
Board:	Standard, GNSS GPS (optional)
Radio Module:	Standard
Power Module:	N/A****
Cover:	Ø 35 × 72 mm
Attachments:	N/A

**Developer *node* (DEV)—
USB-powered entry-level
node variant meant for be-
ginners and developers.**

**DEV *nodes* offer a cheap intro-
duction to LPWAN mesh network-
ing as well as a reliable starting
point to further build upon.**

**This class of *nodes* is recommend-
ed for all *makers* wanting to dip
their toes into this project without
any substantial overhead costs.**



Coverage*:	~500 m, >1000 m***
Cost**:	~50 \$
Board:	Standard, GNSS GPS (optional)
Radio Module:	Standard, External 3dBi SubG Antenna (optional)
Power Module:	Rechargeable Battery
Cover:	Cover: Ø 35 × 142 mm
Attachments:	Salomon Quicklace Netting (optional)

Nomad *node* (NMD)—Battery-powered *node* variant meant for roaming network clients.

NMD *nodes* offer versatility and performance that make them suitable for most use cases.

This class of *nodes* is recommended for all *makers* wanting a straightforward personal *node*, or those looking to simply increase the number of clients on their network.







Coverage*:	LoS ~3000 m
Cost**:	~150 \$
Board:	GNSS GPS
Radio Module:	Standard
Power Module:	Rechargeable Battery
Cover:	Ø 35 × 142 mm
Attachments:	× 2 Voltaic Systems 1 Watt 6 Volt Solar Panels, Hi-vis <i>node</i> ID, ¼" Mounting Support for Tripods

Resident *node* (RES)—Solar-powered *node* variant meant for permanent stationary deployment.

RES *nodes* can solidify a network's topology by extending coverage for all clients, performing best when positioned with LoS between them.

This class of *nodes* is recommended for all *makers* wanting their network to permanently cover a specific area of land.





Coverage*:	LoS >10000 m
Cost**:	>1000 \$
Board:	Standard
Radio Module:	External 3dBi SubG Antenna
Power Module:	Rechargeable Battery
Cover:	Ø 35 × 142 mm
Attachments:	¼" Mounting Support for Drone Camera Brackets

Airborne *node* (AIR)—
Drone-mounted *node* variant
meant for emergency deployment.

AIR *nodes* can function as
temporary RES *nodes* whenever
LoS is not immediately available.

This highly specialised class of
***nodes* is recommended for all**
***makers* wanting to quickly de-**
ploy a large-sized network over
a specific area of land in the
eventuality the situation called
for extreme levels of urgency.



In this phase—we understand how *nodes* are made and apply what we learned by manufacturing the required amount to satisfy our custom meshnet architecture.

Nodes, to both suit even the most demanding needs as well as keep up with tech advancements in the LoRa ecosystem are designed with modularity and customization in mind.

To reflect this, *nodes* are built around the RAKwireless' RAK WisBlock family of products for IoT applications: an industrial-grade modular system of baseboards, MCUs, modules and accessories, that includes an nRF52 MCU for LoRa radio transmission. The ever-evolving RAK WisBlock ecosystem offers a straightforward and extensively documented *maker* experience that does not require any soldering or electronic engineering knowledge, lowering entry barriers without compromising any later and more advanced experimentation.

To further reinforce this, the FDM printed *node* enclosures are also modular and are designed for fast assembling and disassembling with zero fasteners—as snap fits and Lego components allow for a friendly and tool-free approach to *node* manufacturing. Alongside RAKwireless and Lego, *nodes* also make use of parts coming from a series of other trustworthy suppliers like Voltaic Systems, Adafruit and even Salomon.

Together FDM fabricated parts and components sourced from this combination of suppliers establish a familiar design language that runs across all *node* classes and accessories, encouraging all your further hardware-related endeavours.

The network manufacturing phase is divided into three steps: Sourcing, Fabricating and Assembling. Before starting the manufacturing phase, go to the [Tendrill Repository GitHub](#)¹⁴ page and download all the required materials by clicking on Code, then Download ZIP. Also, make sure you have access to the following equipment:

- Computer with internet access and 3D printing slicing software, as well as a vector graphic editor for making Labels and Node IDs.
- FDM 3D Printer.
- Soldering iron with solder, small heat shrink sleeves and splicing scissors (not needed for DEV *nodes*).
- Generic set of tools that must include a compatible screwdriver for M1.2 RAK WisBlock screws.

14



Sourcing

In this step—we source all the required components for the given number and class of *nodes* in our custom meshnet architecture.

Open the Sourcing/Fabricating Google Sheet listed in the GitHub repository to figure out what components you will need to source. Then, place all the required orders.

Please note that:

Components are sourced internationally, checking local suppliers before placing orders can help mitigate shipping costs.

Lego components are sourced from BrickLink. Follow [BrickLink's New User Tutorial](#)¹⁵ for a quick rundown on how it works.

15



Be careful of ordering the Arduino version of the RAK4631.

RAK4631s and 3dBi SubG Antennas have to be ordered specifying their frequency band, so be mindful of choosing the one that is compliant with your national ISM regulations and sticking to it across all the *nodes* in your network.

Europe	EU433	Europe	EU868	Korea	KR920
China	CN470	Canada	US915	Asia	AS923
India	IN865	Australia	AU915	Russia	RU864

Fabricating

In this step—we fabricate all the required parts for the given number and class of *nodes* in our custom meshnet architecture.

Open the Sourcing/Fabricating Google Sheet listed in the GitHub repository to figure out what parts you will need to fabricate. Then, slice the STLs with a 3D printing slicer software of your choice and begin printing.

Please note that:

All STLs have been designed for FDM 3D printing with Autodesk Inventor, sliced with Prusa Slicer and have been tested on a stock Creality Ender 3 V2 using SUNLU PLA+.

Each part is to be fabricated following the notes stated in the Google Sheet.

All STLs are to be printed with a 0.2 mm layer height, 3 outer perimeters, with no supports, in the orientation they enter the slicer with (unless stated otherwise in the Fabrication Notes). All omitted slicer-related information is entirely up to your own preferences.

Before fabricating the 3D-printed parts, printer calibration is recommended. To do so, refer to [Teaching Tech's "3D Printer Calibration" GitHub page](#)¹⁶.

16



Print `Lego_Test_Pin` and `Lego_Test_Axle` to check your tolerances. If both of them behave like Lego when connected with Lego Pins and Axles proceed with printing all other parts. If the connections are too tight, consider recalibrating your printer and try again until you meet the expected behaviour. If the connections are too loose, consider increasing your extrusion multiplier in 5% (+0,05) increments and try again until you meet the expected behaviour. Printing the outer perimeters first can help with improving tolerances across all components, check to see if your slicer can enable this feature.

Labels and Node IDs will require you to outsource printing & laser-cutting processes. Edit the two `.ai` and export files for printing and cutting. Alternatively, use/reference the two `.pdf` to make your own designs.

Assembling

In this step—we assemble all *nodes* in our custom meshnet architecture.

Refer to the following steps to figure out how to assemble your *nodes* according to their class.

Please note that:

No matter what, avoid powering the boards if they are not fully assembled (RAK19003 Mini Base Board + RAK4631 LPWAN Module with both LoRa and BLE antennas firmly plugged in. If you have a RAK12500 GNSS GPS Location Module installed on your Base Board make sure the GPS antenna is firmly plugged in too.

17



You can refer to the [RAK Documentation Center](#)¹⁷ for any further guidance throughout the board assembly process.

RES *nodes* do not have external USB-C access, meaning that, to flash the boards you will be required to take them apart. For the sake of simplicity is therefore recommended to fully assemble all the boards you have beforehand and jump to the Flashing and Configuring steps of the deployment phase before proceeding any further.

Developer node (DEV)

A Assembling the Board

Before proceeding, read the following documents:

- [WisBlock Quick Start Guide](#)^A
- [RAK19003 Quick Start Guide](#)^B
- [RAK4631 Quick Start Guide](#)^C
- [RAK12500 Quick Start Guide](#)^D (if you plan on installing one)



A



B

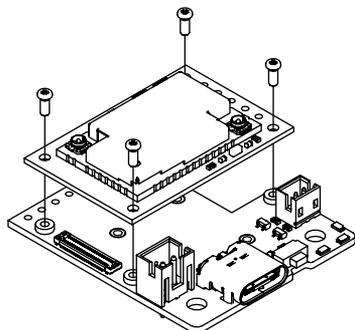


C

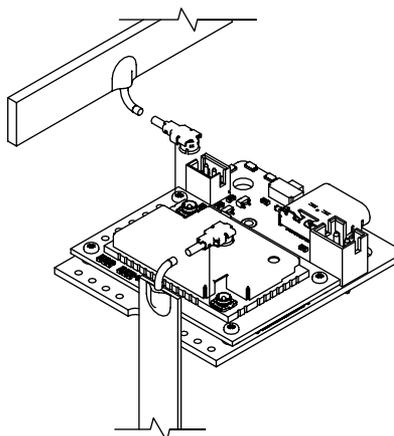


D

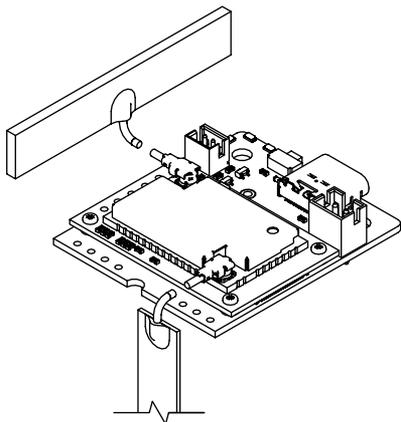
1/4



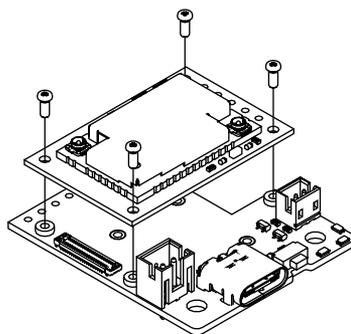
(Standard) 1



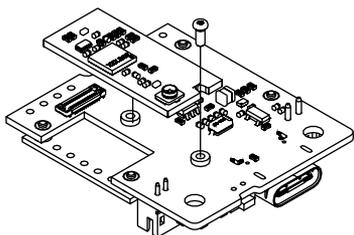
2



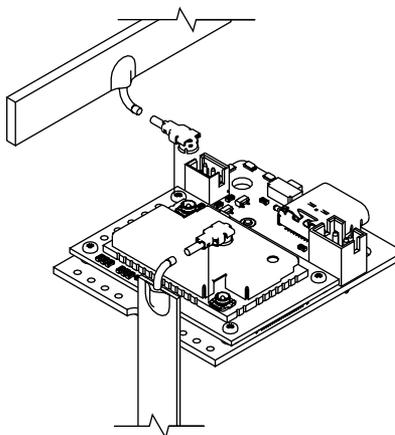
3



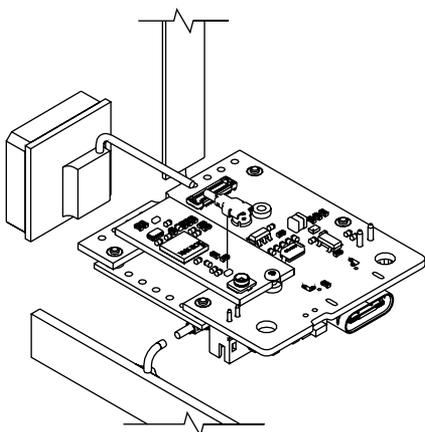
(GNSS)1



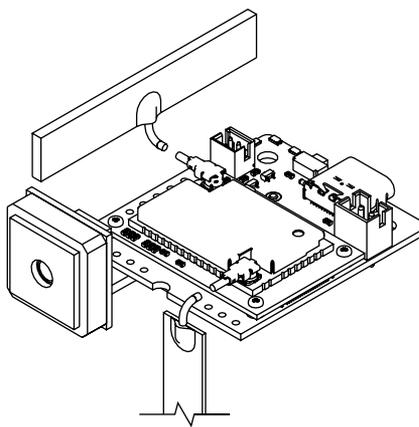
2



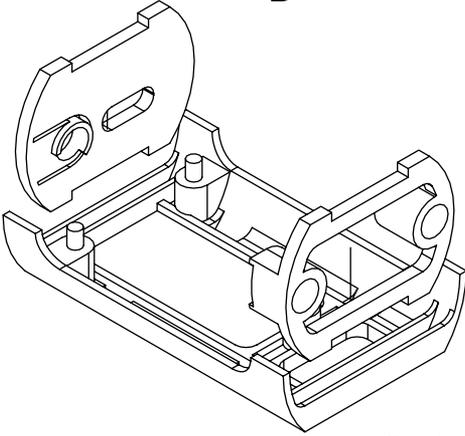
3



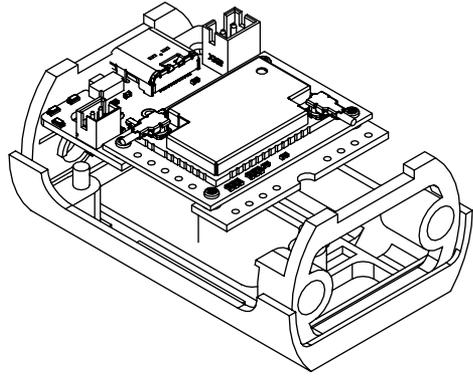
4



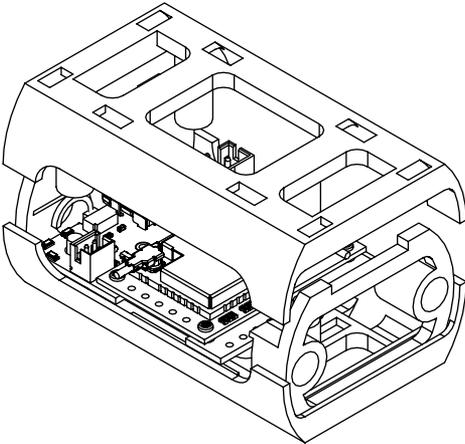
5



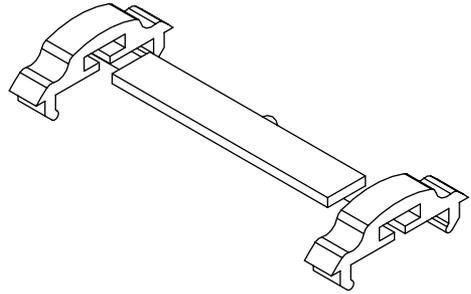
(Standard) **1**



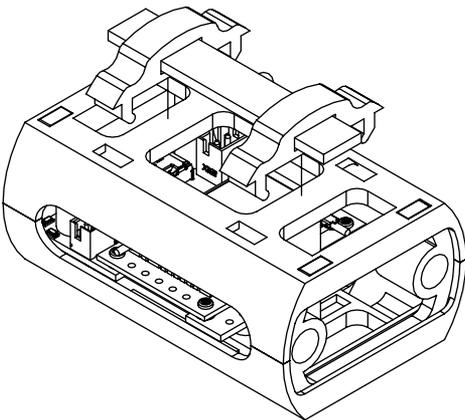
2



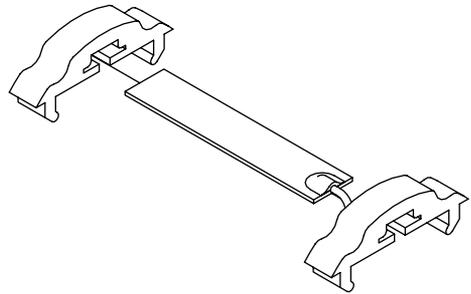
3



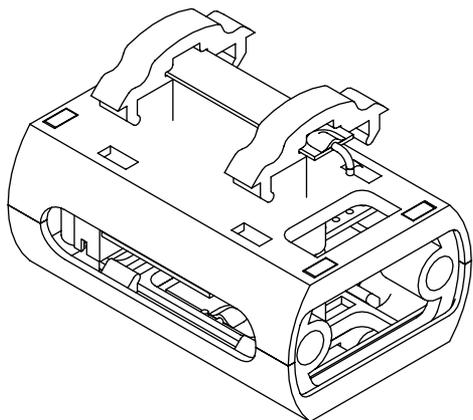
4



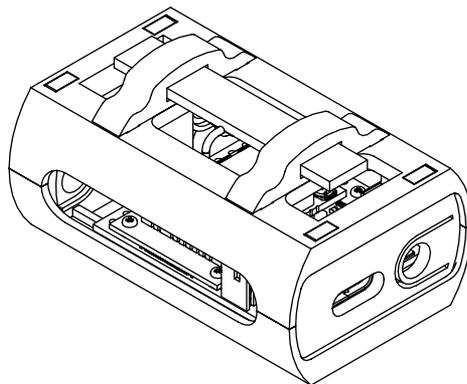
5



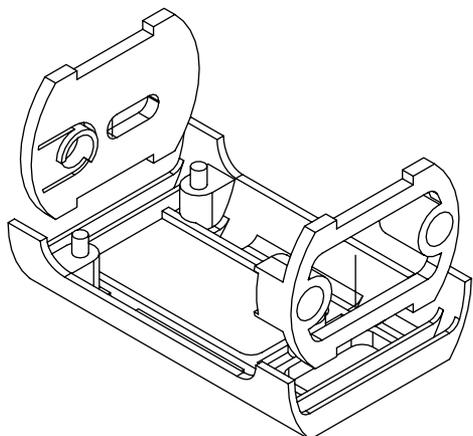
6



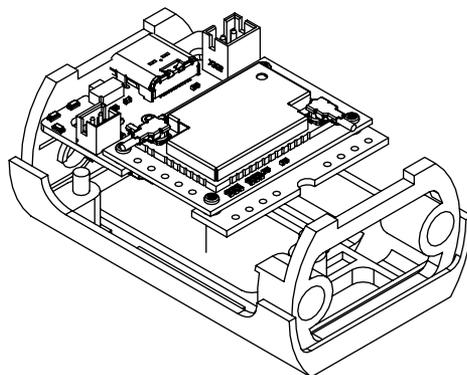
7



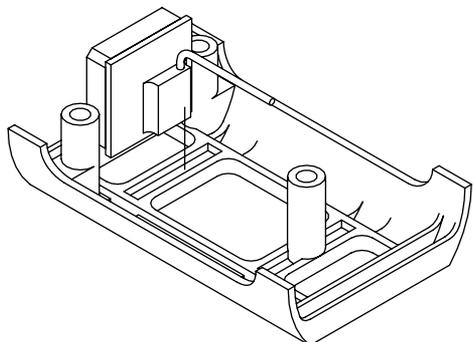
8



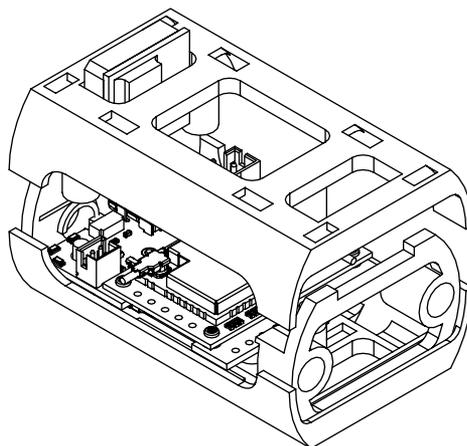
(GNSS) 1



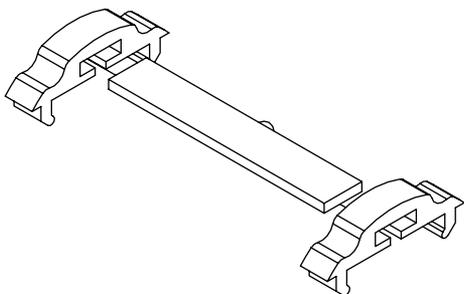
2



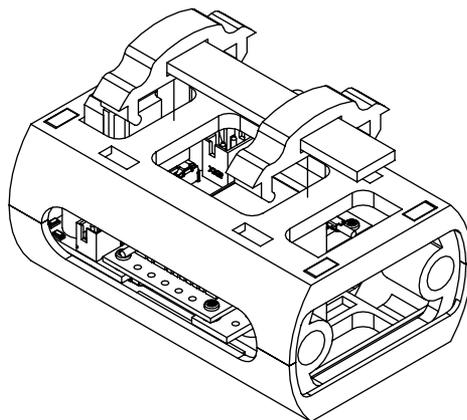
3



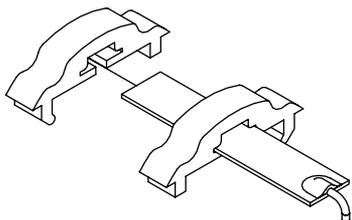
4



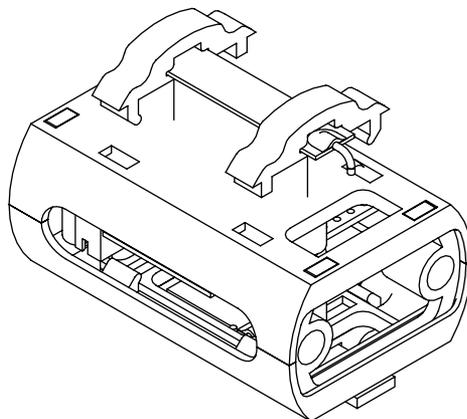
5



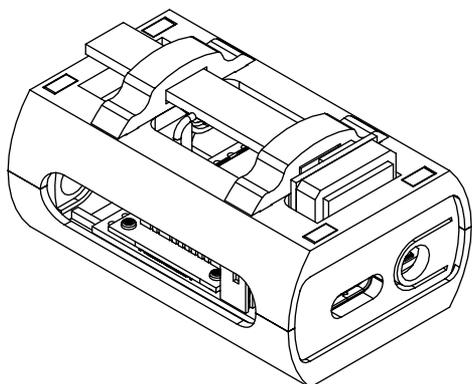
6



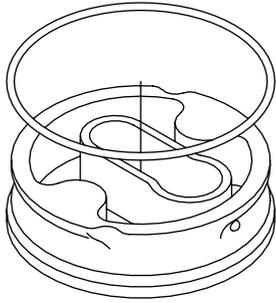
7



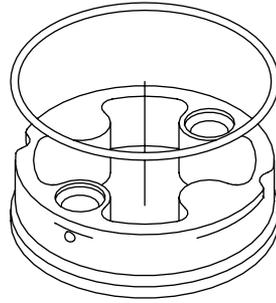
8



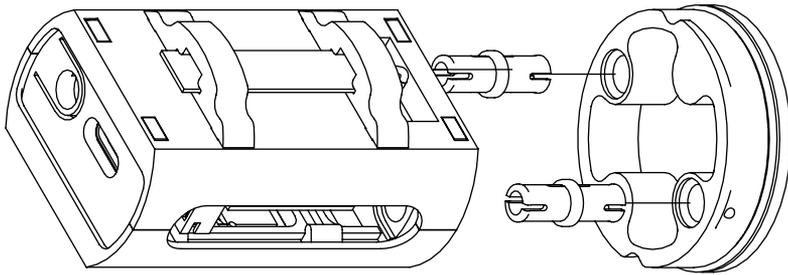
9



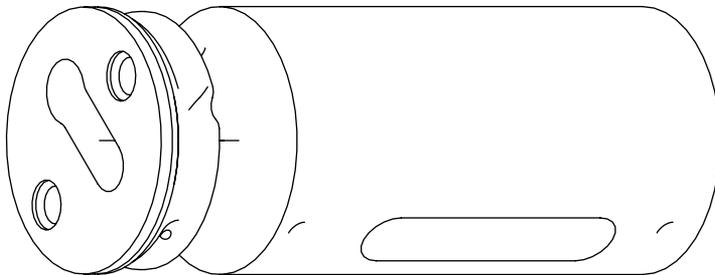
1



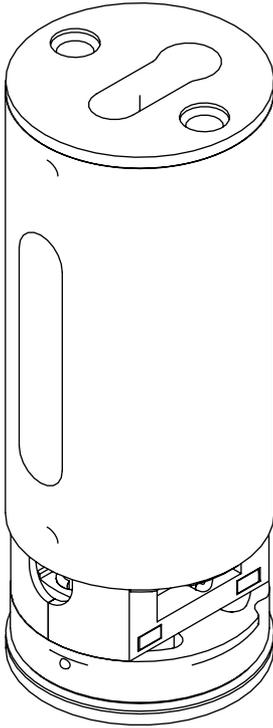
2



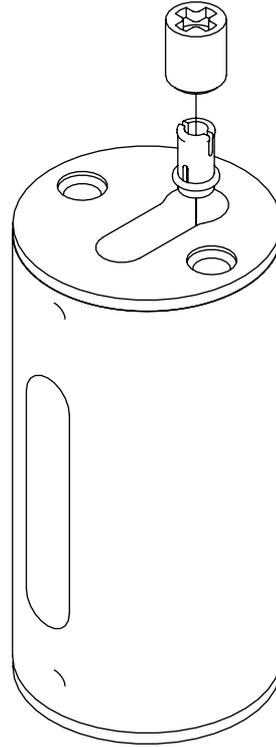
3



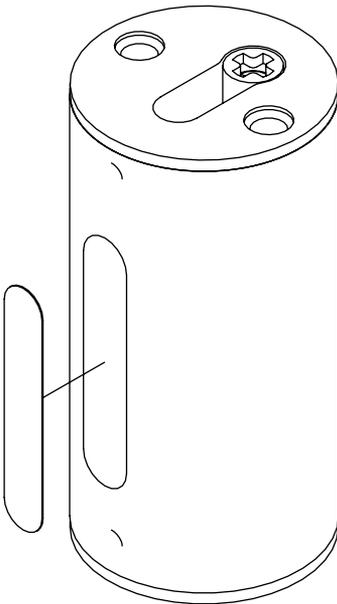
4



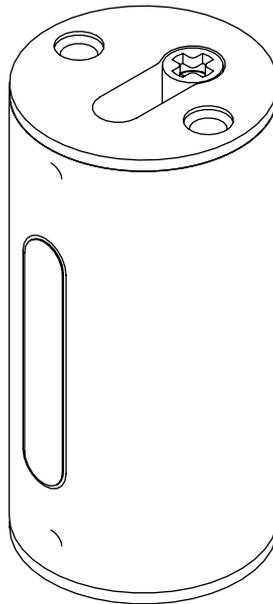
5



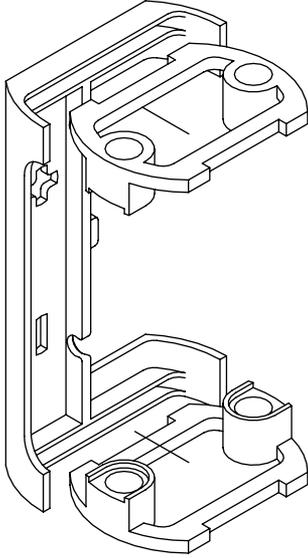
6



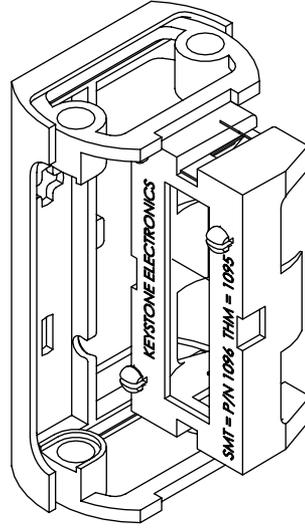
7



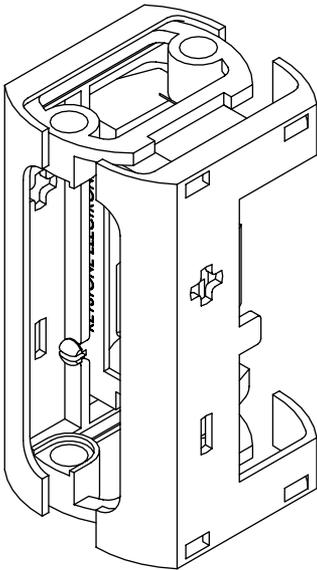
8



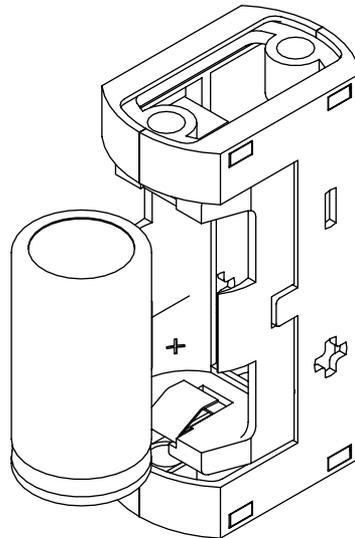
1



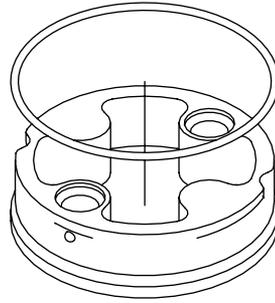
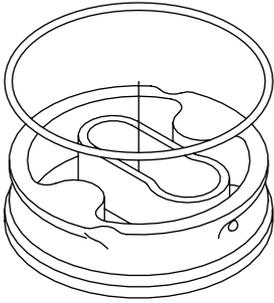
2



3

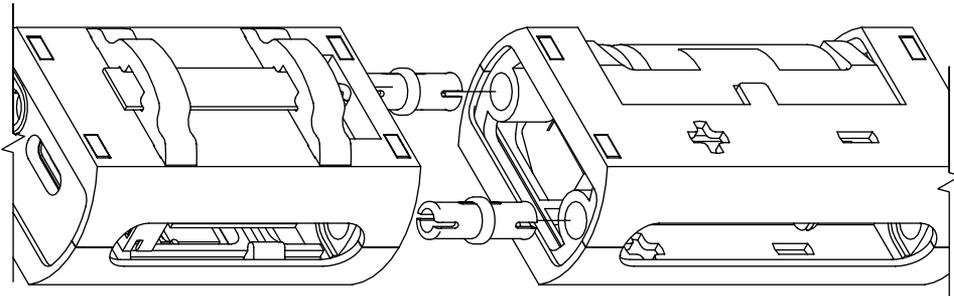


4

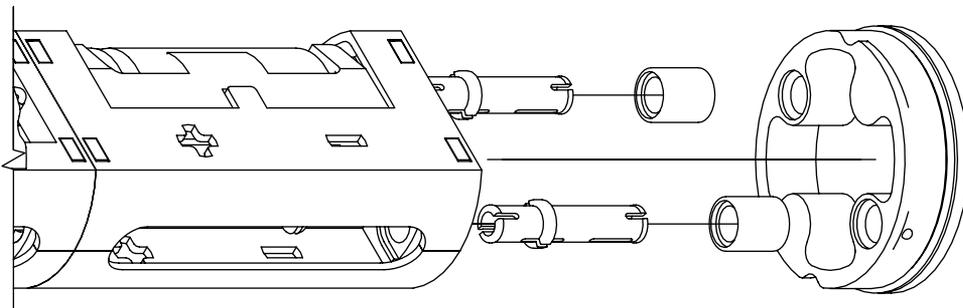


(Standard) 1

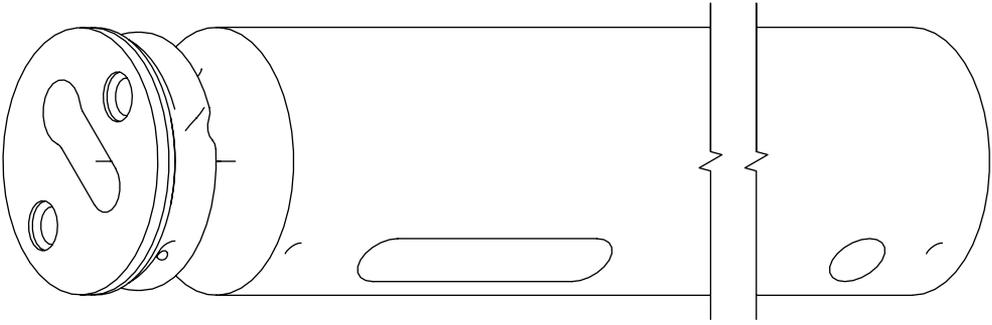
2



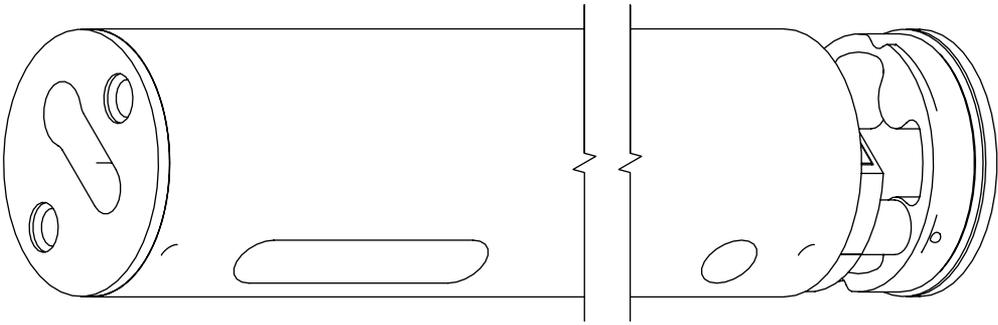
3



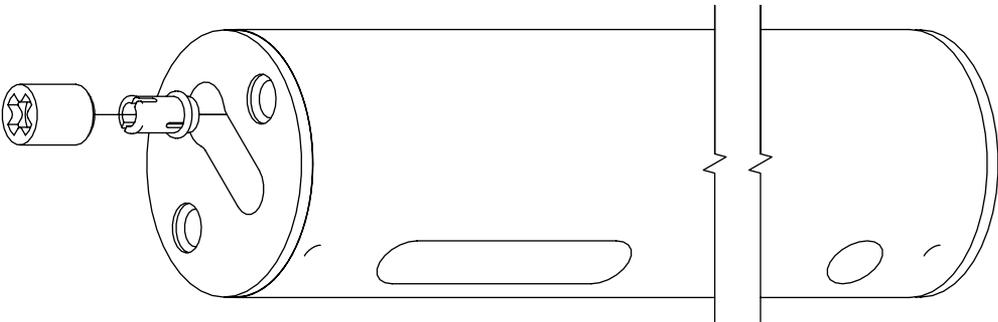
4



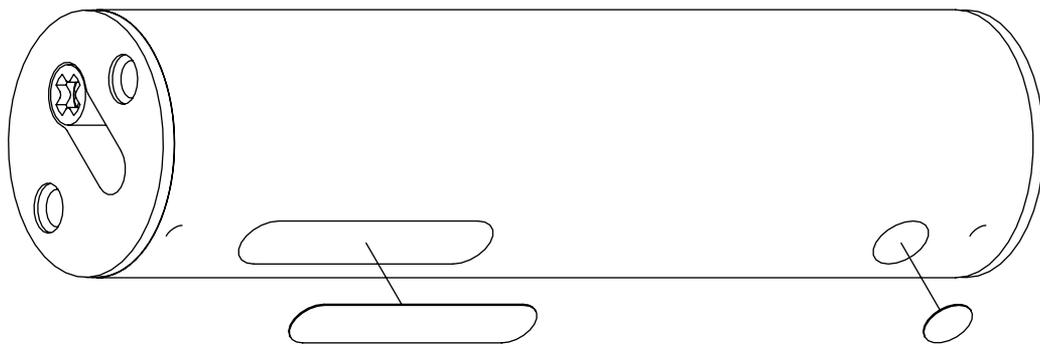
5



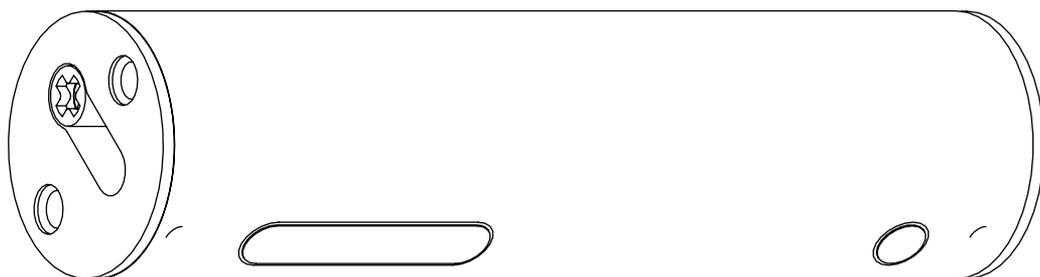
6



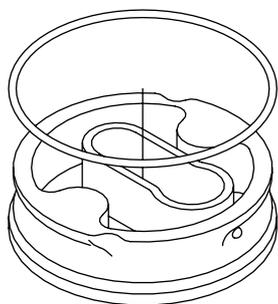
7



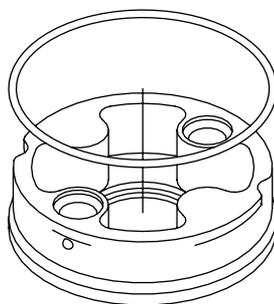
8



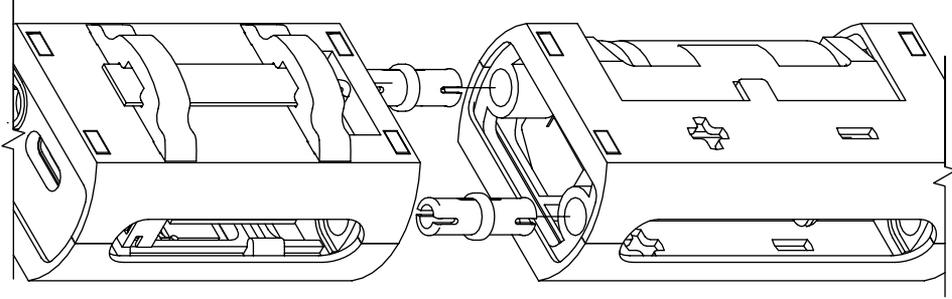
9



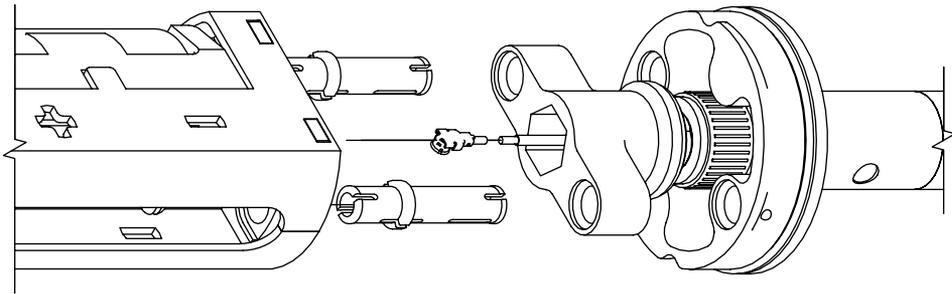
(External Antenna) 1



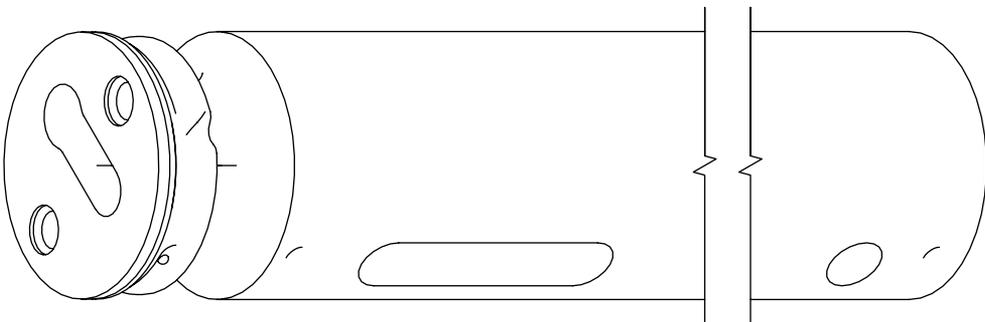
2



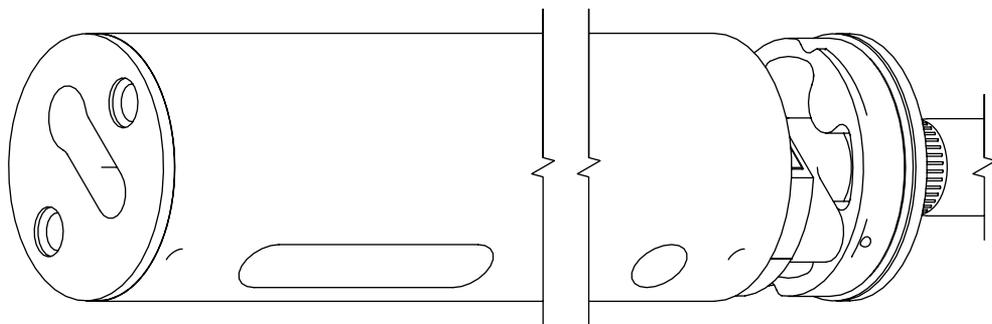
3



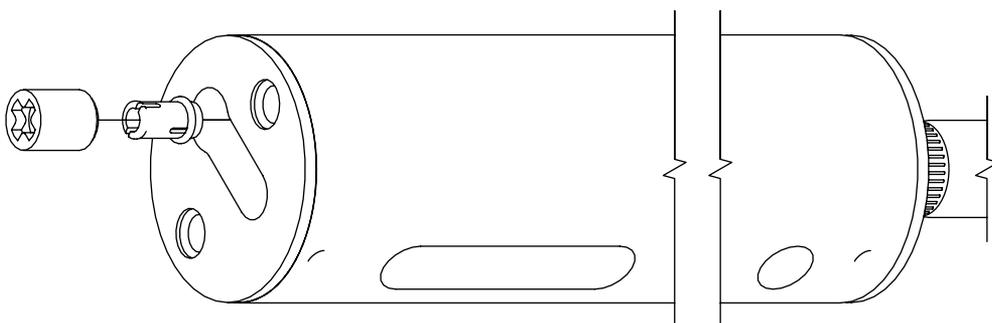
4



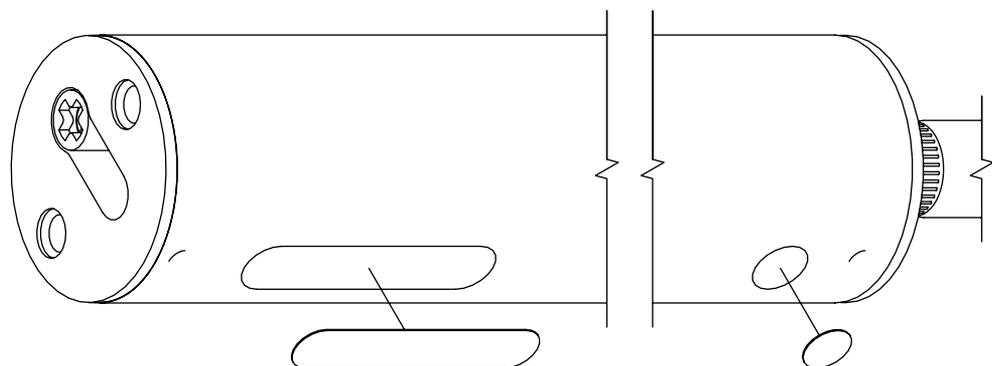
5



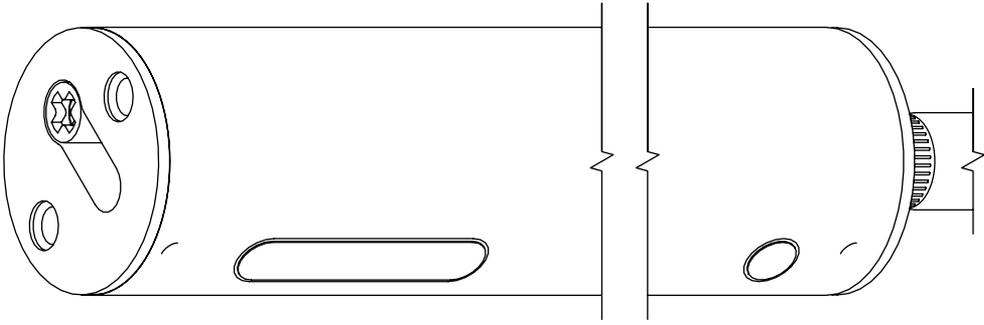
6



7



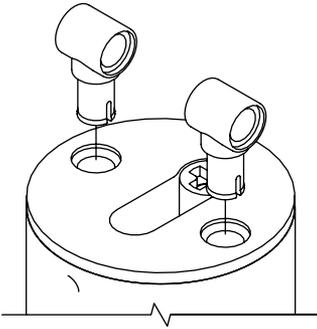
8



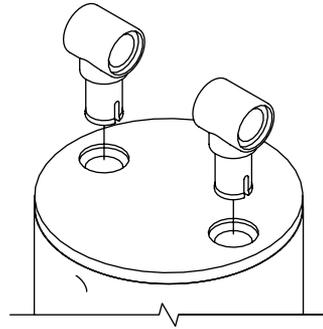
9

1/4

E Assembling the Attachments



1



2



3

Resident *node* (RES)

A Assembling the Board

Read the following documents and refer to point A (GNSS) in the Developer *node* assembly step until the Board is done.

- [WisBlock Quick Start Guide^A](#)
- [RAK19003 Quick Start Guide^B](#)
- [RAK4631 Quick Start Guide^C](#)
- [RAK12500 Quick Start Guide^D](#)



A



B



C



D

After having assembled one board, it is recommended to do the same for all the other *nodes* in your network and jump to the Flashing and Configuring steps of the deployment phase.

B Assembling the Radio Module

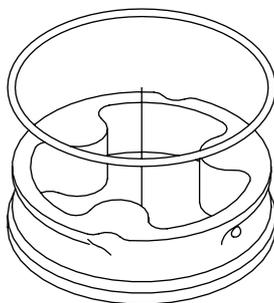
Refer to point B in the Developer *node* assembly step until the Radio Module (GNSS) is done.

C Assembling the Power Module

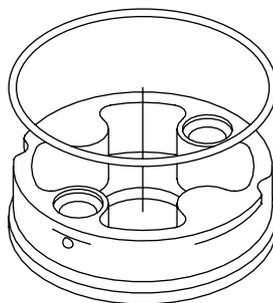
Refer to point C in the Nomad *node* assembly step until the Power Module is done.

1/4

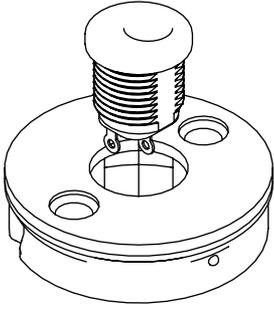
D Assembling the Cover



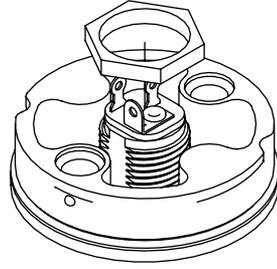
1



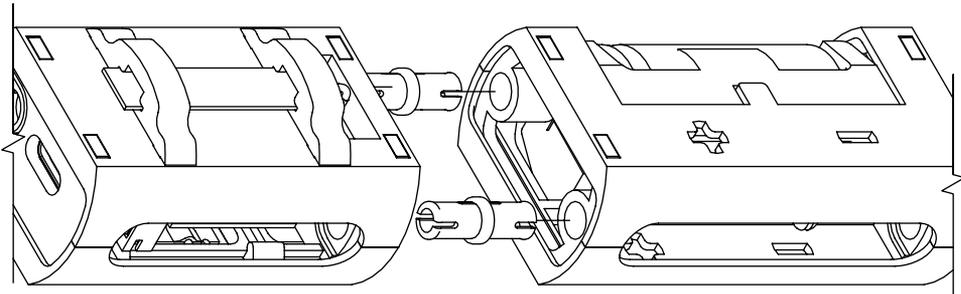
2



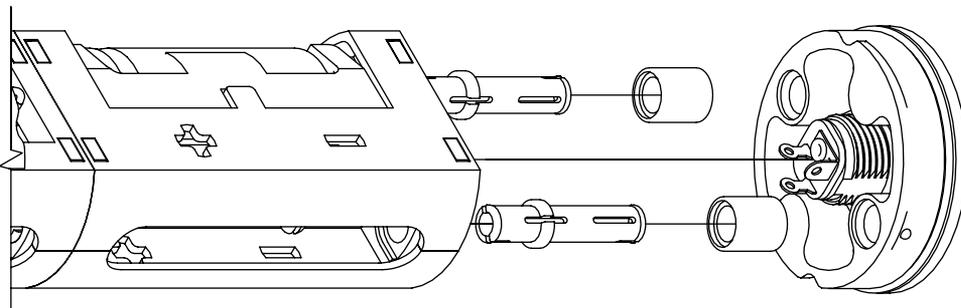
3



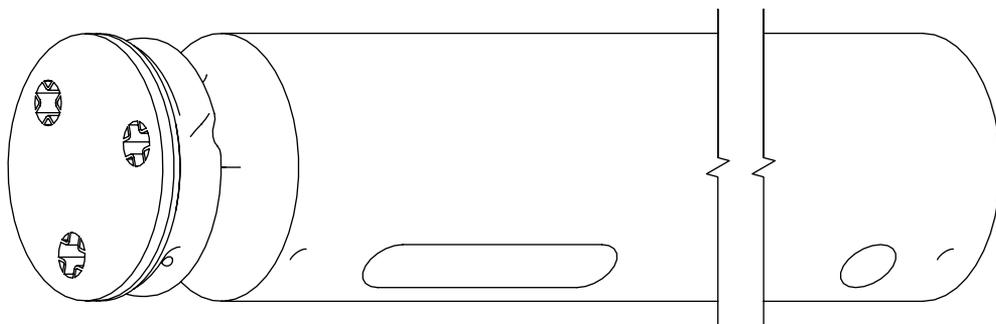
4



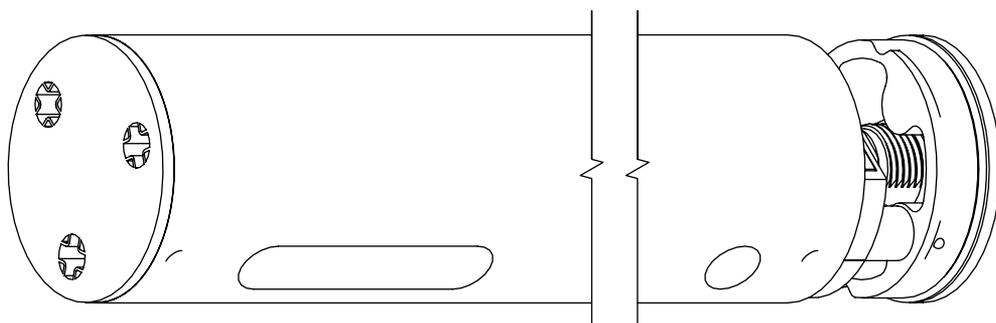
5



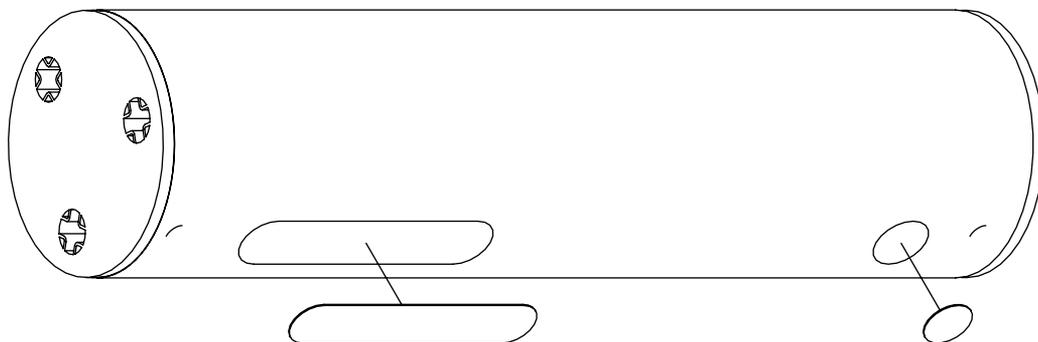
6



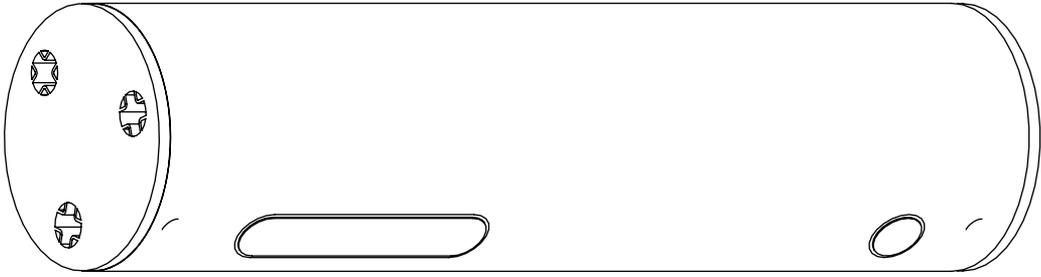
7



8



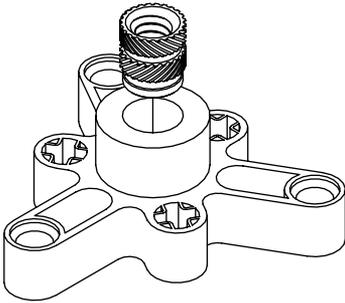
9



10

1/4

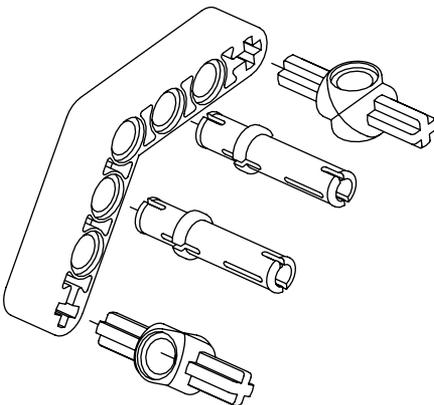
E Assembling the Attachments



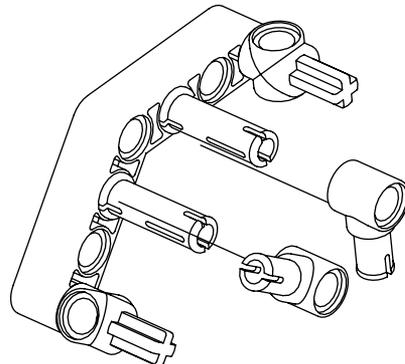
Use the soldering iron to press the Threaded Inserts in. In doing so, keep the soldering iron vertical and the Threaded_Base stable on a flat surface.

When the tip hits the surface (through the bottom hole) the Threaded Insert should be fully seated.

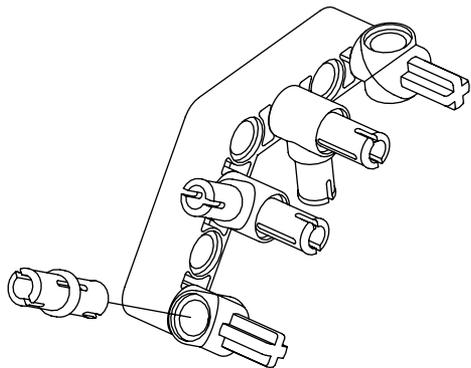
1



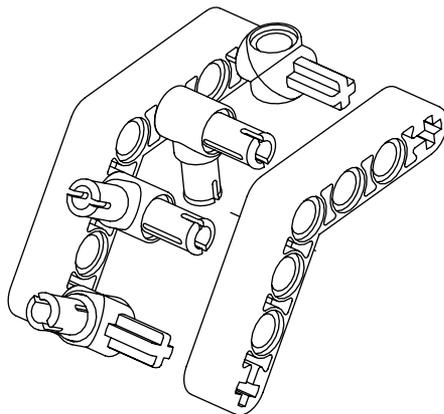
2



3

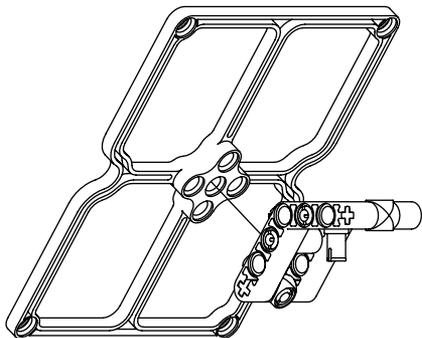


4

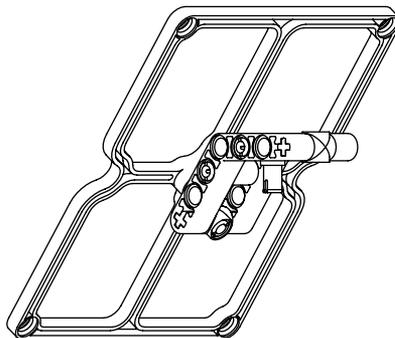


5

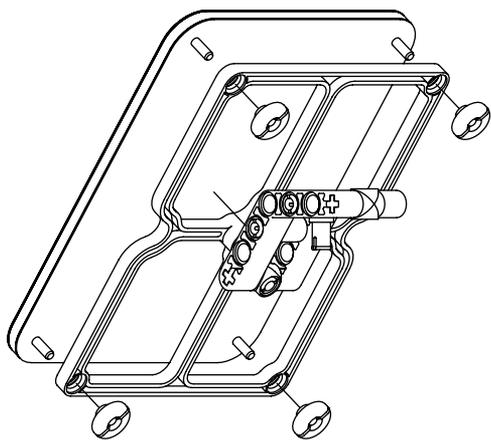
1/2



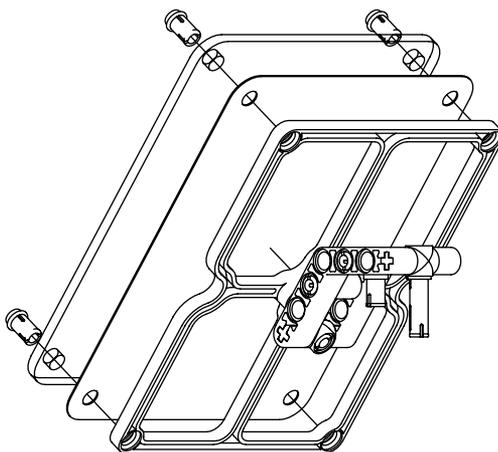
6



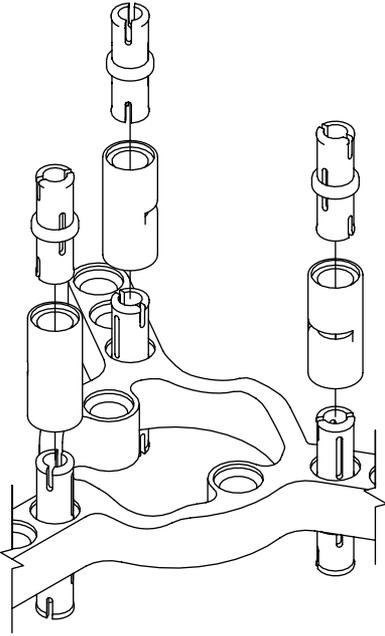
(3x) 7



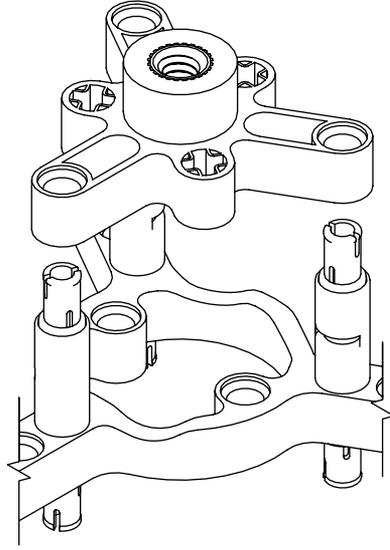
(2x) 8



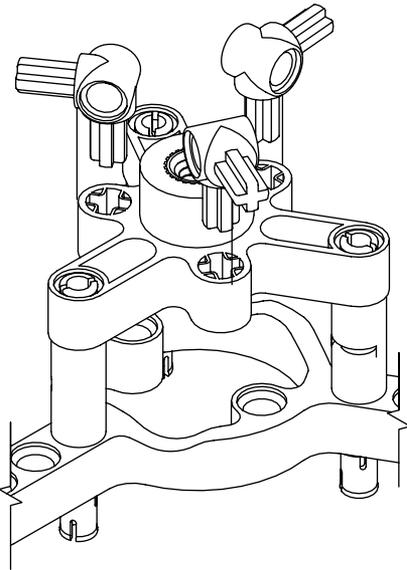
9



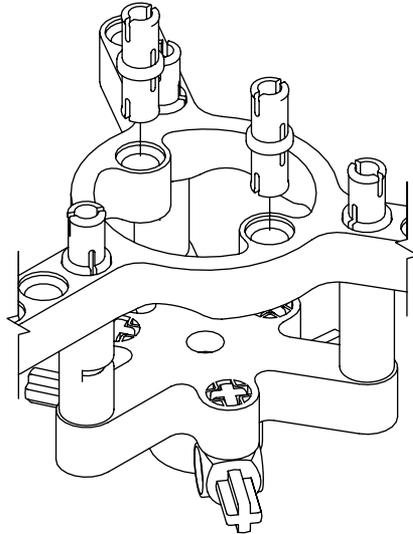
10



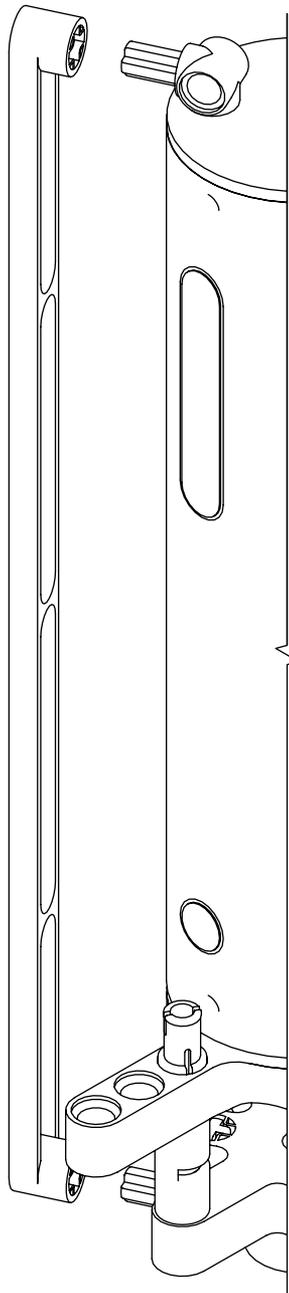
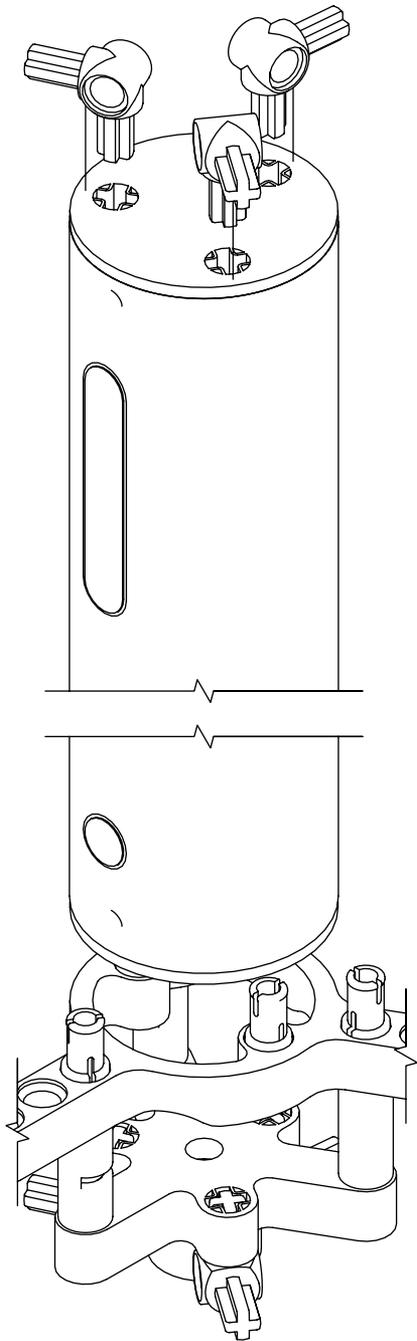
11

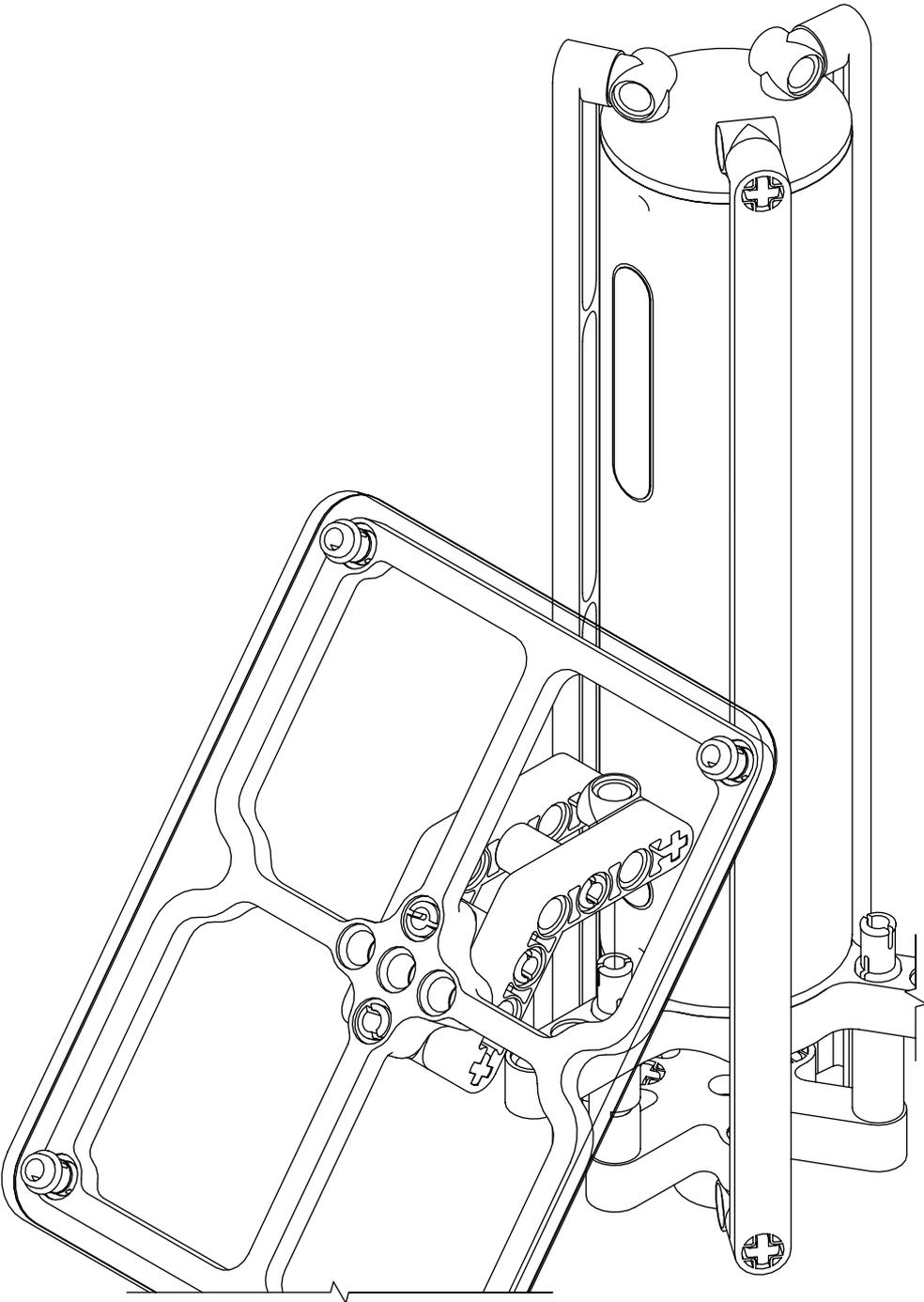


12

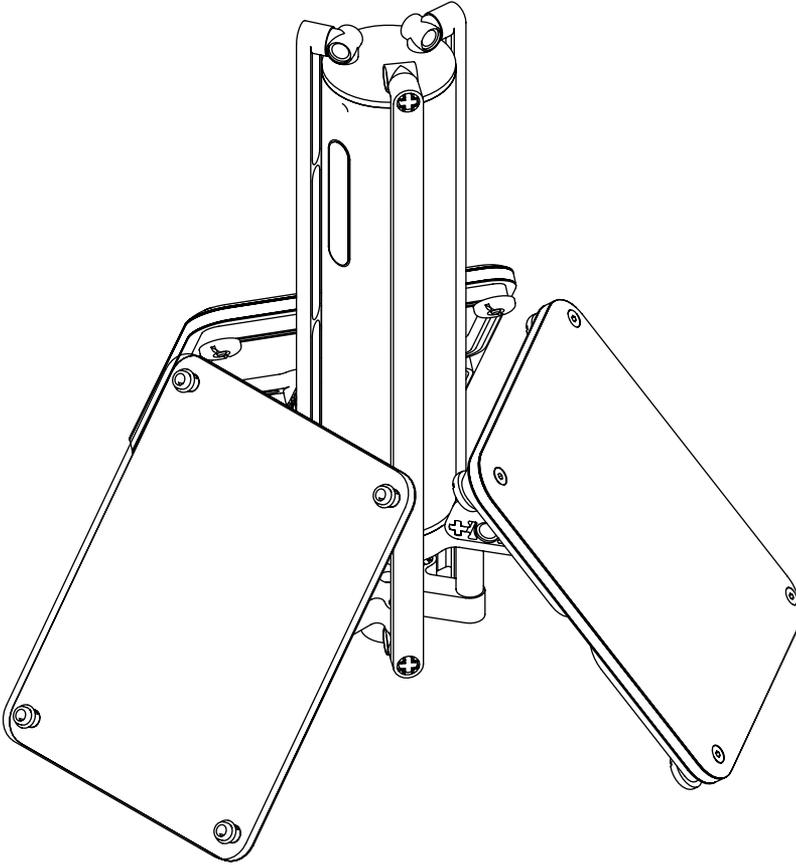


13





(3x) 16



17

Airborne *node* (AIR)

A Assembling the Board

Read the following documents and refer to point A (Standard) in the Developer *node* assembly step until the Board is done.

- [WisBlock Quick Start Guide^A](#)
- [RAK19003 Quick Start Guide^B](#)
- [RAK4631 Quick Start Guide^C](#)



A



B



C

After having assembled one board, it is recommended to do the same for all the other *nodes* in your network and jump to the Flashing and Configuring steps of the deployment phase.

B Assembling the Radio Module

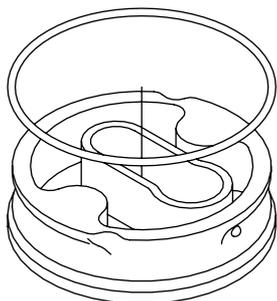
Refer to point B in the Nomad *node* assembly step until the Radio Module (External Antenna) is done. Then, refer to point B in the Developer *node* one (Standard).

C Assembling the Power Module

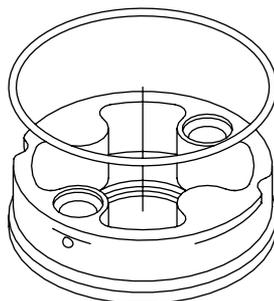
Refer to point C in the Nomad *node* assembly step until the Power Module is done.

1/4

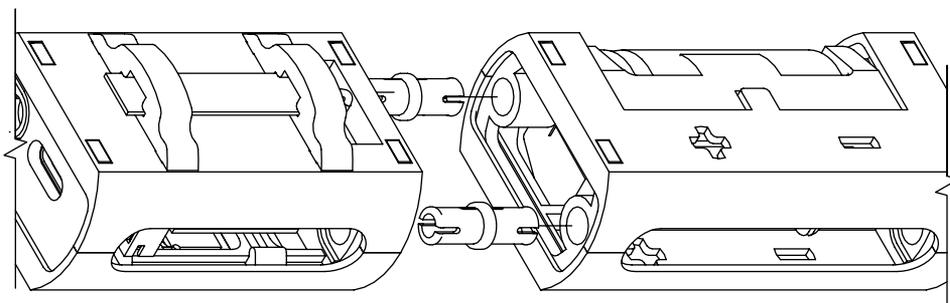
D Assembling the Cover



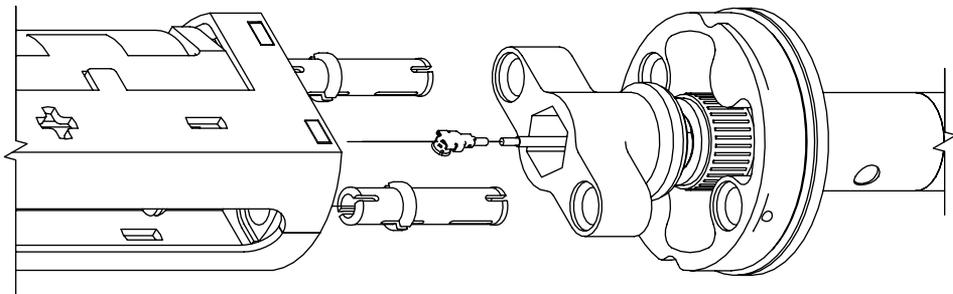
1



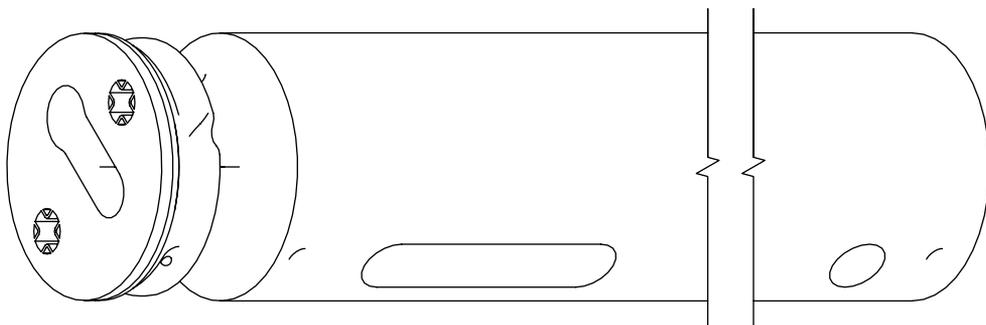
2



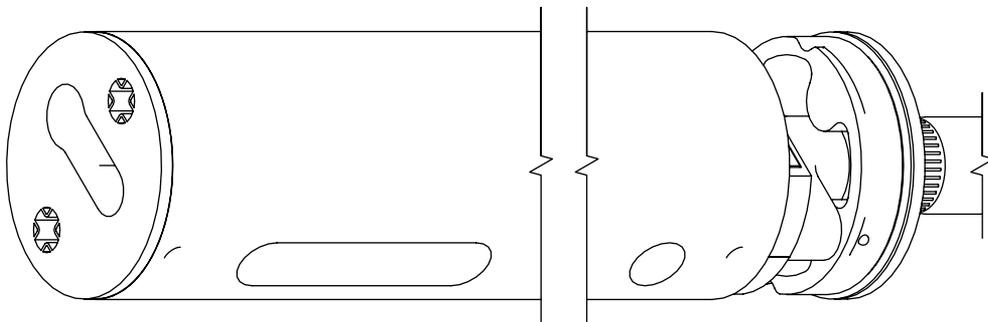
3



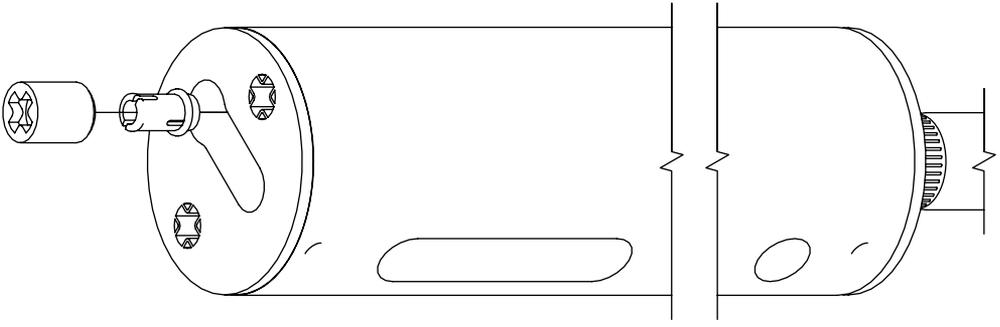
4



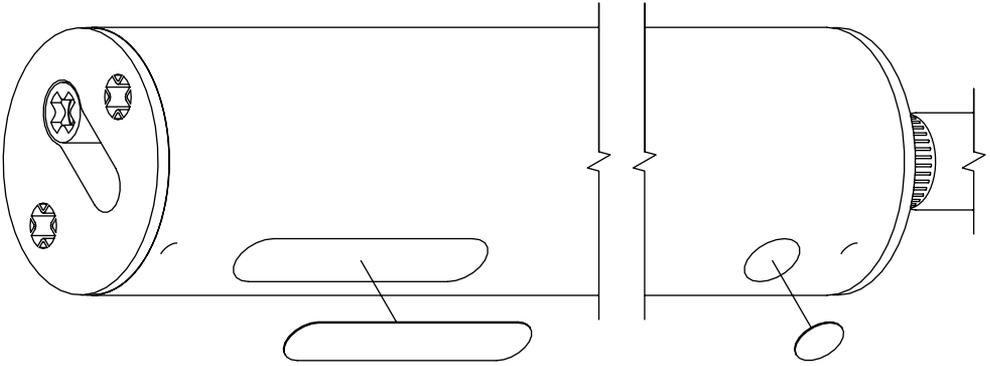
5



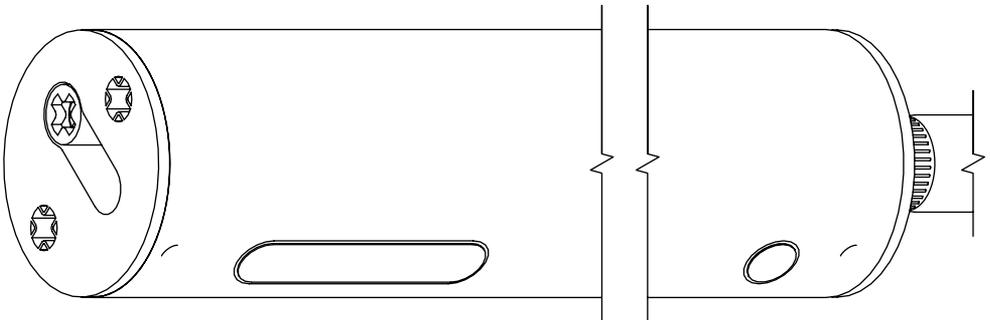
6



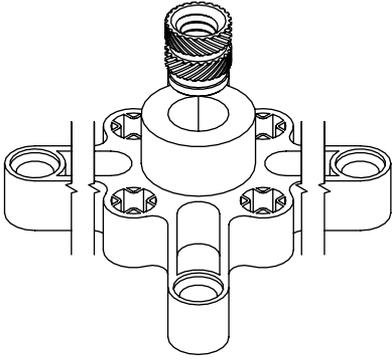
7



8



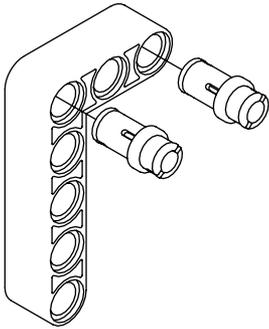
9



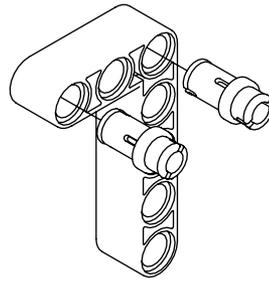
Use the soldering iron to press the Threaded Inserts in. In doing so, keep the soldering iron vertical and the Threaded_Base stable on a flat surface.

When the tip hits the surface (through the bottom hole) the Threaded Insert should be fully seated.

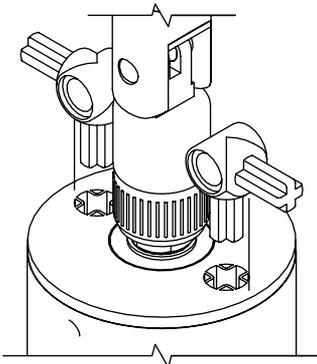
1



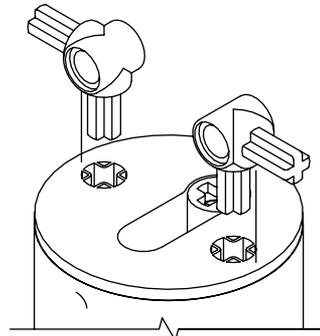
(2×) 2



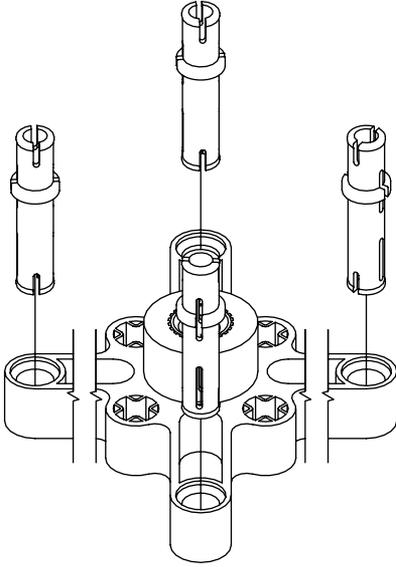
(2×) 3



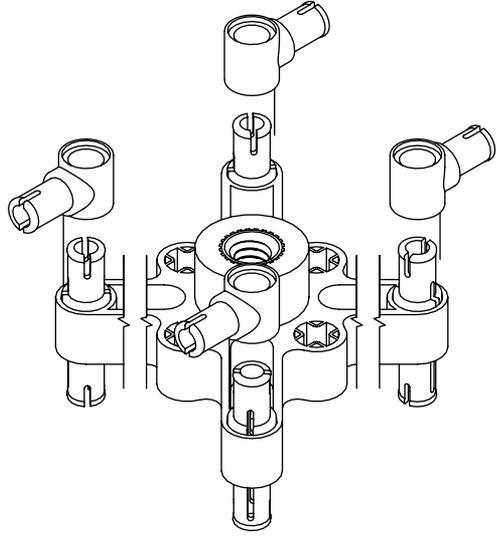
4



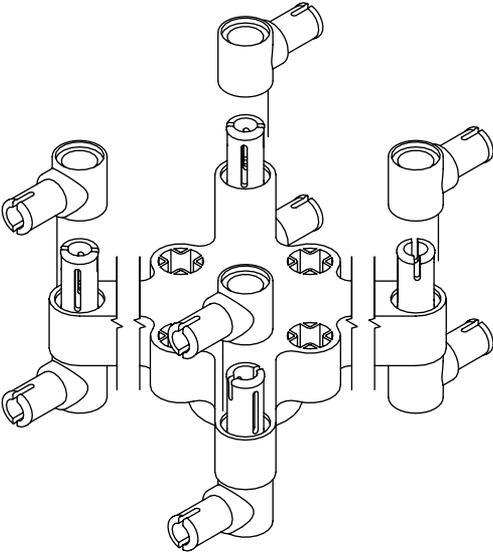
5



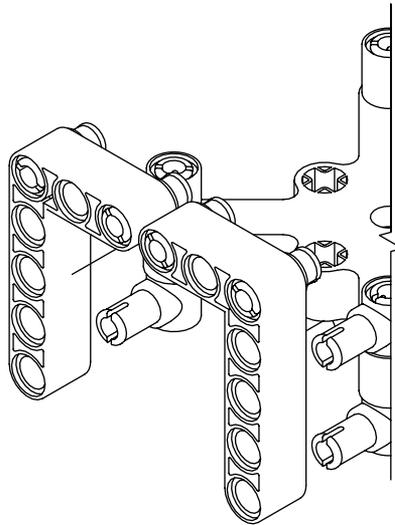
6



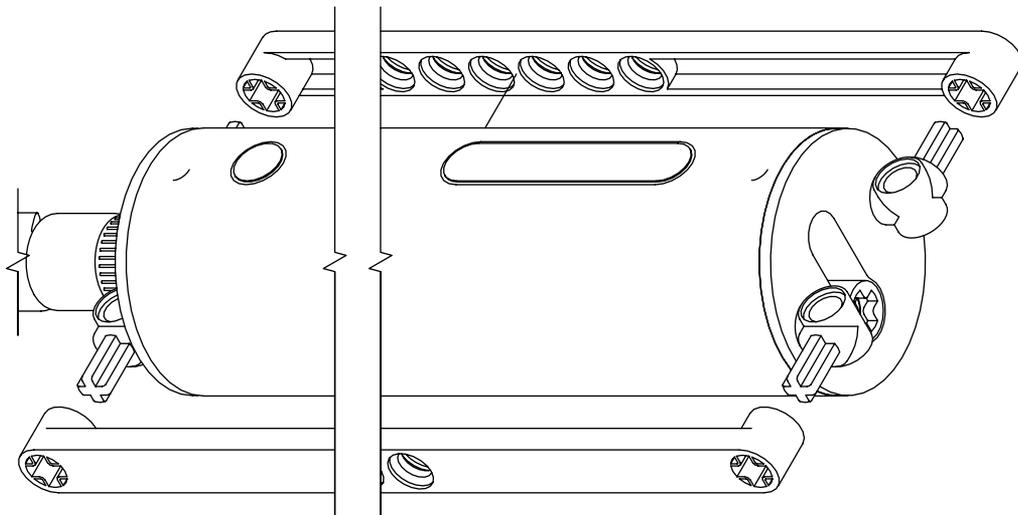
7



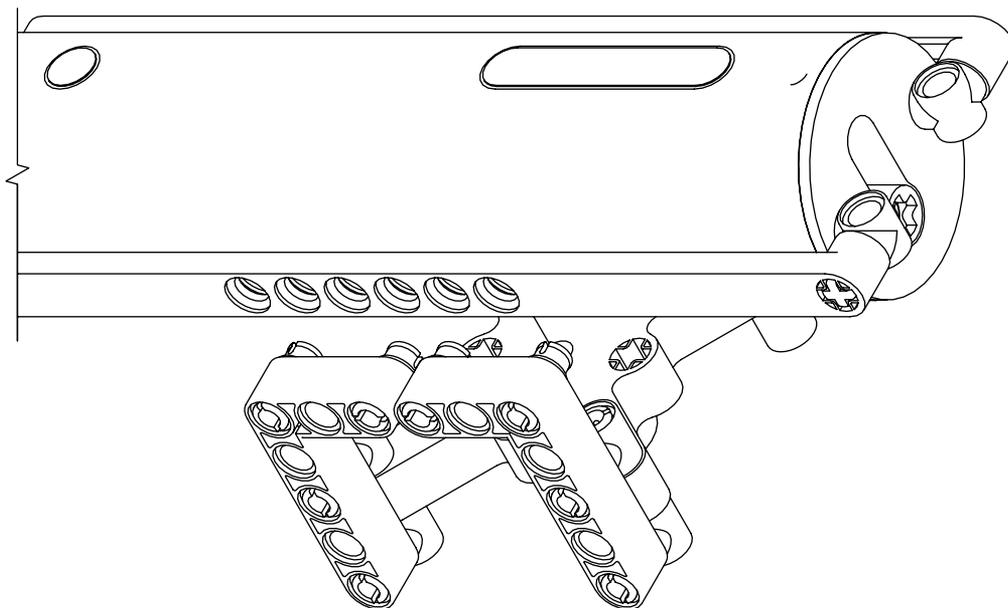
8



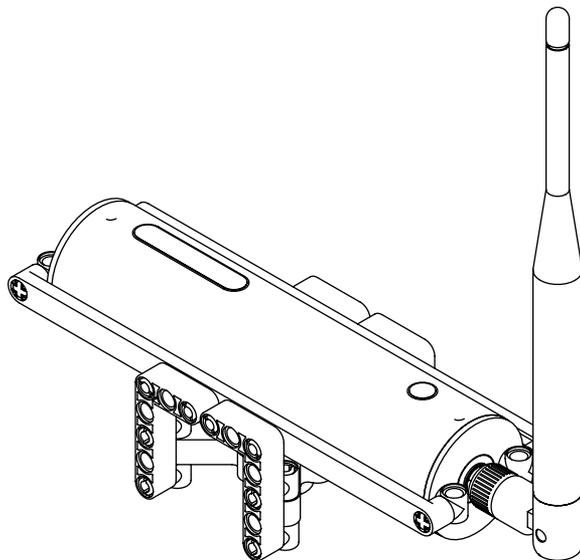
(2x) 9



10



11



12

2.4.3

Deploying the network

In this phase—we understand how *nodes* can be used and apply what we learned by flashing them with the required firmware, interfacing them with your smartphone, tablet or computer and deploying our custom ad-hoc wireless meshnet.

To enable P2P messaging with the LoRa radio transmission technique, the Tendril Repository Networks, rely on [Meshtastic](#)¹⁸, an open-source upper-layer solution that comprises *node* firmware, back-end programming software and even a family of front-end companion apps for iOS & macOS, Android and a web-based one for Chromium browsers.

This ever-evolving ecosystem of resources offers an extensively documented upper-layer implementation process with plenty of community support and recurring updates.

Employing Meshtastic offers a welcoming introduction to online *maker* communities—comprehensive learning environments that function as both nurturing havens for beginners and electrifying launchpads for experts—nests in which "makers" can safely exercise self-production before setting out on their own. So, before starting with the deployment phase, feel free to join the

18





[Meshtastic Discord](#)¹⁹ to meet the community and keep tabs on any development progress coming both from the Meshtastic ecosystem and LoRa's at large.

The network deployment phase is divided into three steps: Flashing, Configuring and Operating.

Flashing

In this step—we flash all *nodes* in our custom meshnet architecture with the Meshtastic firmware.

Please note that:

You can refer to the [Meshtastic Docs](#)²⁰ for any further guidance throughout the flashing process.

20



The devices are going to be flashed following Meshtastic's recommended "drag & drop" method described in the aforementioned docs.

If you have erroneously sourced a RAK4631-R instead of a normal one, refer to [this guide](#)²¹ to convert it to the required Arduino version.

To reset to factory settings, drag and drop the Meshtastic factory reset UFL file from the latest firmware folder into the RAK4631 drive after having put the device in bootloader mode.

21



1. Check to see if the board is assembled properly (RAK19003 Mini Base Board + RAK4631 LPWAN Module with both LoRa and BLE antennas firmly plugged in.

If you have a RAK12500 GNSS GPS Location Module installed on your Base Board make sure the GPS antenna is firmly plugged in too.

2. Download and unzip the latest firmware folder from the [Meshtastic Downloads](#)²² page.
3. Plug the *node* with the provided USB-C cable into your computer.
4. Double-click the button to put the device into bootloader mode.
5. Notice a new drive will be mounted on your computer (Windows, Mac, or Linux).
6. Open this drive and you should see three files: CURRENT.UF2, INDEX.HTM, and INFO_UF2.TXT.
7. Drop the appropriate firmware file (firmware-DEVICE_NAME-vx.x.x-xxxxxxx.uf2) from the release onto this drive.
8. Once the file has finished copying onto the drive, the device will reboot and install the Meshtastic firmware.

22



Configuring

In this step—we configure all *nodes* in our custom meshnet architecture through Meshtastic's command line interface to function together on the same network.

ou can refer to the [Meshtastic Docs](#)²³ for any further guidance throughout the configuring process.



The devices are going to be configured with Meshtastic's Python CLI interface as described in the aforementioned docs.

This step details configuring on a Windows machine using a standalone executable, refer to [this guide](#)²⁴ if you have a different machine and/or wish to proceed differently.



1. Download the latest Meshtastic standalone executable from the [releases GitHub page](#)²⁵ by clicking on “assets” and then “meshtastic_windows”.

2. Rename “meshtastic_windows” to “meshtastic.exe”

3. Open the Command Prompt or another CLI of your choice and navigate to the directory in which the meshtastic.exe executable is located. (e.g. if the executable is on the desktop, run “cd desktop” so that the path changes to “C:\Users\username\Desktop”)

4. Run:

```
meshtastic.exe
```

and check for an affirmative return message.

5. Open the Device Manager and, under Ports (COM & LPT), check for the port your radio is connected to.

6. In the CLI, connect to the radio by running:

```
meshtastic --port COM3
```

(COM3 is an example, check your own in the Device Manager) and check for a “connected to radio” return message.

7. Run:

```
meshtastic --export-config > example_config.yaml
```

and export a configuration file.

8. Duplicate the configuration file for all the *nodes* in your network (so that the “channel_url” stays the same across all nodes) and, if you want, rename all the resulting files to a name of your liking (e.g. nomad1.yaml, nomad2.yaml, resident1.yaml, etc.).

9. One by one open the configuration files with a software of your choice and change the “owner” and “owner_short” of your *node* to a name of your liking and then proceed to save the file.

10. Once you are done, in the CLI, run:

```
meshtastic --configure example_config.yaml
```

to load the configuration file of your choice into the *node* that is currently plugged in and check for confirmation.

```
meshtastic --port COM3 -info
```

(COM3 is an example, check your own in the Device Manager) and check for the returning report.

12. To see all Meshtastic CLI commands, in the CLI, run "meshtastic -help".

Operating

In this step—we operate our custom meshnet architecture through Meshtastic's front-end applications.

Please note that:

You can refer to the [Meshtastic Docs](#)²⁶ for any further guidance throughout the operating process.

This step details operating on an iOS device, some descriptions may vary for other companion apps.



1. Download the appropriate Meshtastic application for the device you wish to interface with the network from the [Meshtastic Downloads](#)²⁷ page.

2. Launch the Meshtastic application.

3. Make sure you are in Bluetooth range from a powered *node*.

4. Navigate to the Bluetooth page to pair with it.

5. If it is your first pairing enter "123456" when asked for a pairing code. Also if it is your first pairing, specify your region-specific frequency band and save (make sure all meshnet *nodes* share the same one), pair again if needed and you are set to go.

- On the Messages screen, you can group-chat with all network clients as well as message them directly by selecting the *node* to which they are connected.
- On the Bluetooth screen, you can pair with all *nodes* within the Bluetooth range of your device.
- On the Nodes screen, you can see all the *nodes* belonging to the meshnet, allowing you to keep tabs on their activities, metrics and battery life.
- On the Mesh Map screen, you can see the latest GPS position of all the *nodes* belonging to the meshnet. This feature, if enabled (Go to Settings > App Settings > Phone GPS and turn on "Provide location to mesh"), allows your device (e.g. phone, tablet, computer, etc.) to regularly transmit its GPS position (also check the App Settings to establish how often) across the meshnet to all other network clients. If the *node* equips a RAK12500 GNSS GPS Location Module, the GPS position will.

FRANCESCO ZURLO

FEDERICO BASSI

ALESSANDRO IANNIELLO

3	Development	73
3.1	Confines & objectives	73
3.1.1	Satisfying the requirements for an IPD thesis	73
3.1.2	Meeting the objectives of the research	73
3.1.3	Working within the confines of the research	75
3.2	Enablers & gatekeepers	75
3.2.1	Working within international radio legislation	75
3.2.2	Employing LoRa mesh networking	78
3.2.3	Designing alongside the Meshtastic community	84
3.2.4	Building for modularity with RAK WisBlocks	86
3.3	Retrospective	94

3.1

Confines & Objectives

3.1.1

Satisfying the requirements for an IPD thesis

According to the programme's listed goals, "[Products evolved] into complex systems that merge both tangible and intangible aspects. [Expanding the discipline to integrate] different facets, such as the relationship between design and technology and the need for new augmented functions; strategic and market-related actions; a focus on social and environmental issues; and a clear understanding of the processes that concur in the definition of the qualities of products."²⁸

To reflect these listed goals, the thesis had to avoid oversimplifying the topics it tried to tackle, especially in the context of understanding what designing for the information age could entail in terms of social responsibility. Attempting this, meant that the thesis had to focus on intangible aspects belonging to design theory just as much as it had to be hands-on with the tangible aspects of production. Thankfully, the core subject of this research has greatly helped in achieving this level of insight into integrated product design, as the thesis itself encourages people to acquire a *full-stack* understanding of these processes.

28 "Degree Programme of Integrated Product Design," 2022

Works cited:

Degree Programme of Integrated Product Design. (2022). Retrieved December 2, 2022, from Polimi.it website: https://www8.ceda.polimi.it/manifesti/manifesti/controller/extra/RegolamentoPublic.do?af_currentWFID=-main&EVN_DEFAULT=evento&aa=2021&k_corso_la=1261&lang=EN

3.1.2

Meeting the objectives of the research

This research has aimed to expand on previous academic work done over the last couple of years at Politecnico, synthesising the preceding explorations into a comprehensive hardware project. Precisely, focusing on ways to allow people to reclaim agency over contemporaneity.

Carrying out this objective with the development of a product has been a challenging endeavour that has been greatly helped by focusing on the topics of network infrastructure and self-production, which have allowed the thesis to easily output something that goes beyond the confines of design theory and truly *stays with the trouble*.

Accessibility has been a key decisive factor for the outputs of the thesis, which had to be designed with incremental complexity to ease *makers* into all the different

Accessibility has been a key decisive factor for the outputs of the thesis, which had to be designed with incremental complexity to ease *makers* into all the different processes—balances had to be struck between the amount of insight these outputs had to offer

processes—balances had to be struck between the amount of insight these outputs had to offer and overall approachability of the topics at hand, keeping people engaged and eager to push the project forward, perhaps even beyond the original scope.

and overall approachability of the topics at hand, keeping people engaged and eager to push the project forward, perhaps even beyond the original scope.

Employing the *stack* framework as a model for contemporaneity has greatly helped to structure the educative objectives this exercise is aiming for, as each of the three phases appearing in the Tendril Re-

pository—Designing the network, Manufacturing the network and Deploying the network—dive into some of the six layers that make up its totality, more precisely:

1. In the designing phase, *makers* could gain insight into and exercise agency over the Cloud, City and Address layers by, respectively, complying with international radio legislation, using the LoRa radio communication technique and designing a suitable network architecture with a mesh-like topology—fundamentally understanding the basics of network infrastructure. In the hope of highlighting the role of governance in the context of infrastructure ownership and legislation.
2. In the manufacturing phase, *makers* could gain insight into and exercise agency over the Earth layer by sourcing and fabricating all the required components needed for the assembling of the *nodes* making up their custom network architecture, using digital fabrication, RAK WisBlocks and Lego—fundamentally understanding how information age products are made, what resources they require to function and where these resources come from. In the hope of highlighting the physicality that allows for the digital to happen, the misleading pointlessness of drawing lines between the two and the sheer size of the global trade that fuels the production of digital products.
3. In the deployment phase, *makers* could gain insight into and exercise agency over the Interface and User layers by configuring the *nodes* to function together, and interface with their devices as well as with themselves, in both the context of back-end development and front-end employment thanks to Meshtastic firmware and software—fundamentally understanding the relationship between users and digital product interfaces from both the perspective of developers and end-users. In the hope of highlighting the ability of these interfaces to dictate and simplify how users relate with the underlying infrastructure, or more specifically, how the Interface layer functions as a governable filter for the layers that exist below it.

3.1.3 Working within the confines of the research

Given that self-production has been a core strategy for achieving this thesis' objectives, this research has, too, been conducted autonomously and from the bottom-up. This has led the research to grow a set of strict boundaries that helped converge toward an output that was true and honest to the nature of the topics at hand.

Working within these confines required a deep dive into state-of-the-art digital fabrication—priming an overall reframing of expectations that strayed away from any conventional logic belonging to industrial production.

Working within these confines required a deep dive into state-of-the-art digital fabrication—priming an overall reframing of expectations that strayed away from any conventional logic belonging to industrial production. This reframing has proven difficult to tackle at times, as designing for this context demanded a thorough involvement

in every single development process, no matter how complex.

Luckily, strategies like open knowledge and open-source design have greatly mitigated the individual workload that was looming above this research, as the further it went, the more “enablers & gatekeepers” it ran into—regulations, technologies and communities that greatly helped to steer the development process towards a successful solution.

3.2 Enablers & gatekeepers

3.2.1 Working within international radio legislation

Some big deciding factors in the thesis' development have been dictated by international regulations for radio communication. These regulations have significantly impacted the overall performance of the networks the Tendril Repository allows you to build, since, for any wireless network to work, the infrastructure that makes up its topology has to communicate using a commonly shared frequency range within the radio spectrum—frequency ranges which, in the interests of both safety and economic interests, are not necessarily all open to the public.

Radio band regulations have ultimately led the Tendril Repository networks to employ the LoRa radio communication technique as a physical layer.

In the seven-layer Open Systems Interconnection model of computer networking, OSI for short, the physical layer is the first and lowest layer—the layer most closely associated with the physical connection between devices. Fundamentally, the physical layer specifies the shapes and properties of the electrical connectors,

the line code and, most importantly, the frequencies to broadcast on.²⁹ Technically, radio frequency, RF for short, is the oscillation rate of an alternating electric current or voltage in the 20 kHz to 300 GHz electromagnetic spectrum, these oscillations in current or voltage fundamentally allow for data to be transmitted and happen within a range that determines the bandwidth size of a given radio signal. Practically, in the context of telecommunications, the combination between frequency and bandwidth determines how far data within the network can travel, how fast it can get there and in what amount.

When it comes to international legislation, this abundance of available frequency ranges can be figuratively imagined as land to be occupied, in which operators need to acquire a licence granting them the right to operate within a given range.

When it comes to international legislation, this abundance of available frequency ranges can be figuratively imagined as land to be occupied, in which operators need to acquire a licence granting them the right to operate within a given range.

These strict regulations allow radio systems to function safely, responsibly and in compliance with their objectives, as each frequency range is only suitable for given applications. To make a couple of examples: low and very low frequencies, LF and VLF for short, have been historically employed for maritime navigation; high and very high (HF and VHF) are used for FM radio and television; ultra-high (UHF) for GPS and cellular; whereas, super and extremely high frequencies (SHF) get primarily employed for satellite communications.

Occupying frequencies without complying with their regulations could lead to the disruption of the services that are already working on them, for this reason, licences have become indisputable requirements that cannot be glanced over, especially in the context of critical infrastructure. However, given how much inclusivity concerns the objectives of this research, a primary decision factor for the frequency ranges on which this thesis' networks could operate has been whether or not a given band required a licence, as it would be nothing short of unreasonable for a *maker* to acquire one just for this exercise.

Most licence-free alternatives are grouped under the international ISM radio bands umbrella, a portion of the radio spectrum reserved for those industrial, scientific and medical purposes that do not threaten the disruption of normal telecommunication infrastructure.³⁰

ISM radio bands are defined in the International Communication Union Radio Regulation document, ITU RR for short, written by the ITU themselves—in the same RR document, since these

bands are many and vary greatly, they also get allocated to specific applications to better coordinate their radio spectrum utilisation.

Funnily enough, despite the original ISM specifications having envisioned these radio bands to be primarily used for non-communication purposes, in the past twenty years there has been a massive surge of radio technology that uses ISM bands for wireless communications.

The appeal of unlicensed use and general radio spectrum cluttering has led the tech industry to focus on ISM radio technologies for IoT applications, ironically making non-ISM applications the largest implementation field within ISM radio bands.

More precisely, the most prolific use of these bands has been for short-range and low-power wireless communications systems—to make a couple of well-known examples: WiFi, Bluetooth (which both share the 2.4 GHz frequency range), RFID and NFC all employ ISM radio bands.

The appeal of unlicensed use and general radio spectrum cluttering has led the tech industry to focus on ISM radio technologies for IoT applications, ironically making non-ISM applications the largest implementation field within ISM radio bands.

More precisely, the most prolific use of these bands has been for short-range and low-power wireless communications systems—to make a couple of well-known examples: WiFi, Bluetooth (which both share the 2.4 GHz frequency range), RFID and NFC all employ ISM radio bands.

This boom in short-range radio technologies for the IoT has even prompted the European Commission to issue new regulations for what they classify as Short Range Devices, SRDs for short. These regulations are de-

scribed by ECC Recommendation 70-03 and are mandated by the European Commission through CEPT and ETSI, which allocate several device bands for SRD purposes, restrict the parameters of their use and provide guidelines for avoiding radio interference.

If it was not for these SRD restrictions, widespread 2.4 GHz communication techniques like WiFi and Bluetooth would have been more than suitable technologies for the Tendril Repository networks, especially considering how much data they can carry . . . However, for the sake of safety, the devices employing these techniques get heavily limited in effective radiated power, ERP for short—to put it bluntly, the effective range between network clients is purposefully restricted to no more than a hundred metres (hence the low-power and short-range device definition), as any more than that would lead the transmitters to exceed a safe ERP.

Unsurprisingly this has led WiFi and Bluetooth to get primarily employed in wireless personal area networking, WPAN for short, a type of networking which would have not contributed to satisfying the objectives of this research. WPANs would have been far too underpowered to allow *makers* to exercise agency

over what they were building and far too ordinary for them to gain any insight into the topic of network infrastructure.

Thankfully, in direct response to these limitations, which not only concern the objectives of my research but also affect the IoT industry, in recent years there is been a substantial push for unlicensed communication techniques for low-power wide-area networks (LPWANs), that can operate within the ISM bands in full compliance with SRD ECC regulations.³¹

These techniques, to meet safe ERPs, sacrifice the volume of data they can carry in favour of sheer range by occupying a lower frequency range with a larger wavelength, to name a couple: DASH7, WiFi HaLow (IEEE 802.11ah) and Narrowband IoT (NB-IoT) are all radio technology standards that have been developed in the last five years to tackle LPWANs. But, as much as all of these standards are promising, Semtech's LoRa, as of 2022, is the most diffused among them, harbouring accessible documentation as well as great manufacturing and community support, something which for all the intents and purposes of this thesis would have been extremely important.

31 European Conference of Postal and Telecommunications Administrations, 1997

Works cited:

International Organization for Standardization. (1994). ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Retrieved from International Organization for Standardization website: <https://www.iso.org/standard/20269.html>

(2020). ITU Radio Regulations, International Telecommunication Union.

European Conference of Postal and Telecommunications Administrations. (1997). ERC/REC 70-03. Retrieved from European Conference of Postal and Telecommunications Administrations website: <https://docdb.cept.org/document/845>

3.2.2

Employing LoRa mesh networking

LoRa is a proprietary radio communication technique developed for LPWANs that, to enable robust long-range communications on unlicensed ISM frequency bands, employs a spread spectrum modulation based on the chirp spectrum modulation technology, CSS for short, and pairs it with integrated forward error connection (FEC). To quantify what this means, the omnidirectional range of generic LoRa radio devices can be up to a declared maximum of 5 km in urban areas, and, provided there is a line of sight (LoS) between the *nodes*, up to 15 km in rural areas, achieving data rates between 0.3 kbit/s and 27 kbit/s.³²

32 Semtech Corporation, 2022

Furthermore, LoRa achieves global compliance with international regulations, as we may already know, using licence-free sub-gigahertz radio frequency bands, precisely EU868 (863–870/873 MHz) in Europe; AU915/AS923-1 (915–928 MHz) in South America; US915 (902–928 MHz) in North America; IN865 (865–867 MHz) in India; and AS923 (915–928 MHz) in Asia.³³

33 LoRa Alliance, 2021

LoRa was developed by the French company Cycleo (Patent No. 9647718B2, 2014), which was subsequently acquired by international semiconductor supplier Semtech, leading LoRa to become the most widely adopted physical layer for LPWANs fairly

... LoRa's remarkable performance, international regulatory compliance and industrial support, make it the perfect candidate for the Tendril Repository networks as well.

quickly. Predictably, LoRa's remarkable performance, international regulatory compliance and industrial support, make it the perfect candidate for the Tendril Repository networks as well.

Having come to a final decision regarding the physical layer and established how *nodes* were going to communicate with one another, attention shifted towards the upper layer. Upper layer is a broad definition that, to stick to the OSI model, would comprise the remaining six layers: Data link, Network, Transport, Session, Presentation and Application. This group of layers determines everything from the physical and logical addressing to the representation and encryption of data, essentially establishing how the networks function and what they function for.

For this reason, choosing what upper layer should govern the Tendril Repository networks, has ultimately come down to isolating what the networks at hand were exactly meant to do, something that further raised the question of what could have been the best way to stay true to the educational purposes of this exercise.

To correctly guide *makers* through the full-stack process of creating network infrastructure, stopping at the physical layer, as much as it would have provided generous room for experimentation, it would have also perplexed most people on what to do next with their freshly built networks, intimidating *makers* from engaging with this project in the first place. Furthermore, refusing to offer guidance through the process of deploying the network would not have allowed this research to meet some of the objectives discussed earlier, greatly compromising insight into both the Interface and User layers of Bratton's *stack*. So, even though *makers* would have always been free to employ a different upper layer from what this exercise is suggesting (something that will have to be reflected in how the *nodes* themselves are designed), a decision on what this upper layer could be had to be taken.

At first glance, the most straightforward option seemed to come from the LoRa Alliance. This non-profit association (that counts over 500 members, including companies like IBM, Orange, Cisco, A2A and, of course, Semtech) takes care of the (entirely open-source) development of the LoRaWAN protocol, a cloud-based medium access control (MAC) layer protocol that man-

ages communications between LPWAN gateways and end-*node* devices. These components are interconnected in a *star-of-stars* topology, where end devices communicate with one or more gateways (using LoRa as the physical layer) and where each gateway dispatches LoRaWAN frames to a network server using a higher-throughput backhaul interface (e.g., WiFi or 5G). Then, applications interfacing with the server can best use the collected data.

As its topology might already suggest, this protocol is best suited for IoT scenarios dependent on collecting, centralising and interpreting information from devices like detectors, sensors and actuators, making it fairly clear that the employment of the LoRaWAN protocol undoubtedly concerns industrial applications more than it does the objectives of this thesis . . . Therefore, as valuable to the LoRa ecosystem as this protocol is, it has been eventually dismissed as the upper layer for this project, since complying with its regulations would not only have been superfluous but also potentially counterproductive.

Specifically, the protocol would not have been suitable for the Tendril Repository networks for three reasons: first of them being that the centralised *star-of-stars* topology of LoRaWAN, given that it requires gateways and network servers to function, greatly contributes to increasing initial overhead costs for no particular grounds that could concern the objectives of this exercise; second of them, the fact that the strict communication frequencies, data rate, and power restrictions required on networked devices needlessly limit their performance to comply with standards that pertain to large-scale projects rather than simple private networks; and, lastly, the IoT-specific nature of the protocol does not necessarily suit the immediate needs of a private user, who would greatly struggle in finding valuable use cases for the infrastructure they are building.

Going forward in the upper-layer decision process, it has been the very need for a use case worthwhile to the eyes of most *makers* that lead the project towards a more suitable solution than LoRaWAN—specifically, one which, through a LoRa physical layer, tackled the topic of peer-to-peer (P2P) messaging, as it could be argued that sending messages to other people is a universally understood feature of information age infrastructure, as well as an essential part of living life in the contemporary.

Venturing into the subject of P2P telecommunications could have offered *makers* an easily comprehensible end goal to their

Venturing into the subject of P2P telecommunications could have offered *makers* an easily comprehensible end goal to their exercise that not only is compliant with the objectives of the research but also highlights

exercise that not only is compliant with the objectives of the research but also highlights its strengths, further reinforcing all the parallels that were drawn between the ownership of infrastructure and the ownership of data that travels within it . . .

its strengths, further reinforcing all the parallels that were drawn between the ownership of infrastructure and the ownership of data that travels within it—making unmistakable that: if Mari believed that building your furniture could contest the furniture industry itself, this research is attempting to do the same with the current internet landscape—moreover raising awareness to the

fact that both critiques are addressing the way “things” are made, not so much the “things” themselves.

For much of the same reasons that have affected this choice, setting up LoRa networks with upper layers that focus on P2P messaging evolved to be a fairly diffused practice in the world of DIY electronics, so much so that resources and documentation on how to build *LoRa messengers* devices have become widely available on platforms in the likes of Hackster³⁴, Instructables³⁵ and Hackaday³⁶.

³⁴ Coward, 2021

³⁵ AkarshA, 2019

³⁶ Cockfield, 2022

A particular community, however, decided to take things much further than a simple article on a blog, going as far as developing an entire ecosystem dedicated to this purpose.

This ecosystem is called Meshtastic and it includes entirely open-source firmware for networked radio devices, platform-agnostic software to interface with them and plenty of resources, documentation and community support to help *makers* figure out how to do so—making it the perfect upper layer candidate for guiding *makers* through the process of deploying their infrastructure.

Meshtastic, contrary to LoRaWAN, as the name might suggest, uses a different topology commonly called LoRa mesh. In a mesh

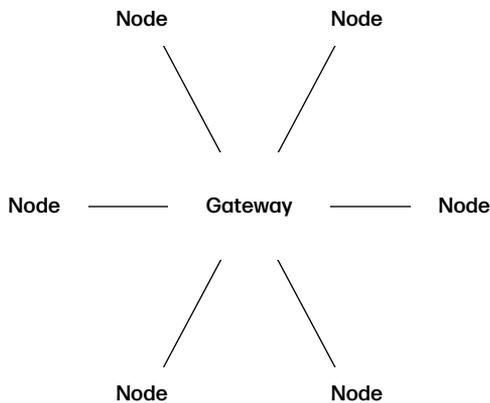
network, meshnet for short, devices act as *nodes* that can connect dynamically and directly to each other without the need for any hierarchy between them, meaning that every *node* contributes to relaying data within the network.

To do so, Meshtastic employs a four-layer broadcasting algorithm that allows the meshnet to relay messages using a routing

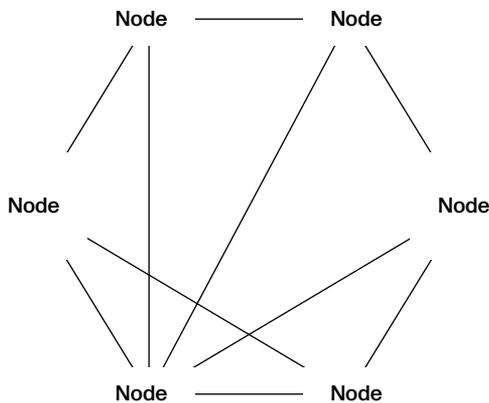
In a mesh network, meshnet for short, devices act as *nodes* that can connect dynamically and directly to each other without the need for any hierarchy between them, meaning that every *node* contributes to relaying data within the network.

technique where the information is propagated along a path by hopping from *node* to *node* until it reaches its destination. Specifically: Layer 0 “LoRa Radio”, adds data conversion into LoRa symbols for radio transmission; Layer 1 “Unreliable Zero Hop Messaging” adds conventional LoRa packet transmission; Layer 2

LoRaWAN's "star-of-stars" topology



Meshtastic's meshnet topology



Topological differences between the LoRaWAN upper layer and Meshtastic's

“Reliable Zero Hop Messaging” adds reliable messaging between the *node* and its immediate neighbours only, meaning that *nodes* keep trying to send a message until they successfully do so; finally, Layer 3 “(Naive) Flooding for Multi-Hop Messaging” adds flooding, meaning that by introducing the concept of HopLimits, in the eventuality in which data needs to reach a non-neighbouring *node*, all the neighbouring *nodes* attempt retransmission on behalf of the original sending *node* in a waterfall effect, allowing for a message to hop within the network until it reaches its destination.³⁷

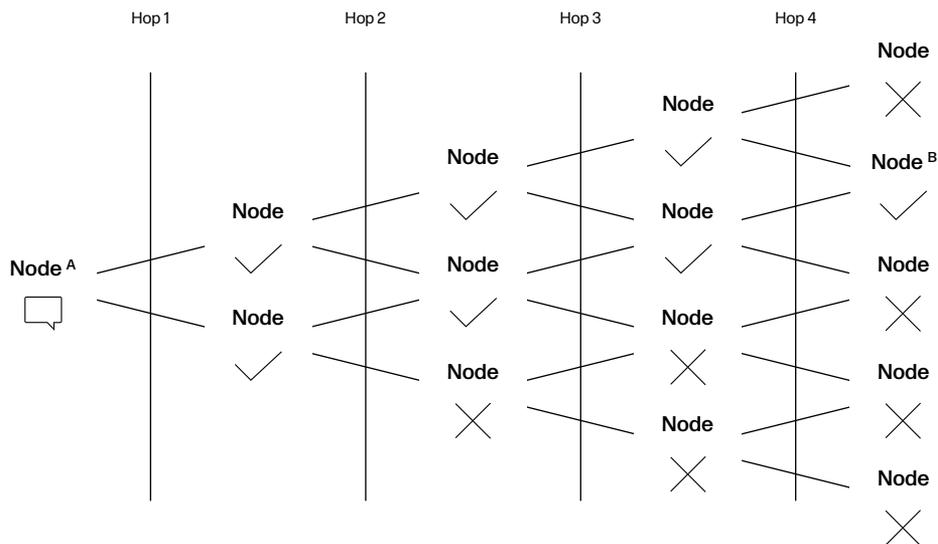
³⁷ Meshtastic, n.d.

Programming the network to employ this broadcasting algorithm requires flashing all the *nodes* within the network with hardware-specific firmware available in the Meshtastic repository and configuring them to function together via Command-line Interface, CLI for short.

Then, to interface with the *nodes*, compile the messages and operate the network, the Meshtastic ecosystem has a family of companion apps, available for iOS & macOS and Android devices, that take care of the front-end side of things, effectively mimicking a normal messaging app.

When you send a message on your Meshtastic companion app, it is relayed to the radio using Bluetooth. The message is then broadcast by the radio three times over a certain interval to create redundancy for lost packets.

When a receiving radio captures a packet, it checks to see if it has heard the message before. If it has it ignores the message. If it has not heard the message, it will rebroadcast it at a certain interval three times. Each message a radio rebroadcasts marks the



HopLimit down by one. When a radio receives a packet with a HopLimit of zero, it will not rebroadcast the message.

Employing Meshtastic not only allowed the Tendril Repository networks to have a solid LoRa upper layer but also enriched the outputs of this research with a full-fledged front-end interface that would have taken years to develop autonomously.

Employing Meshtastic not only allowed the Tendril Repository networks to have a solid LoRa upper layer but also enriched the outputs of this research with a full-fledged front-end interface that would have taken years to develop autonomously.

The contributions this community had in the development, however, did not stop there, as the decision of using their resources has ultimately started an ongoing relationship of mutual collaboration that involves this project, Meshtastic developers and even hardware companies.

Works cited:

Semtech Corporation. (2022). LoRa and LoRaWAN: Technical overview. Retrieved from <https://loro-developers.semtech.com/documentation/tech-papers-and-guides/loro-and-lorawan/>

LoRa Alliance. (2021). RP002-1.0.3 LoRaWAN® Regional Parameters. Retrieved from <https://loro-alliance.org/wp-content/uploads/2021/05/RP-2-1.0.3.pdf>

Seller, O. B. A. (2014). Patent No. 9647718B2. Retrieved from <https://patents.google.com/patent/US9647718B2>

Coward, C. (2021, February 6). This LoRa messenger is perfect for texting in the wilderness. Retrieved December 2, 2022, from Hackster.io website: <https://www.hackster.io/news/this-lora-messenger-is-perfect-for-texting-in-the-wilderness-108ed61aaace>

AkarshA. (2019, March 1). LoRa Messenger for Two Devices for Distances Up to 8km. Retrieved December 2, 2022, from Instructables website: <https://www.instructables.com/LoRa-Messenger-for-Two-Devices-for-Distances-Up-to/>

Cockfield, B. (2022, May 25). Long-distance text communication with LoRa. Retrieved December 2, 2022, from Hackaday website: <https://hackaday.com/2022/05/25/long-distance-text-communication-with-lora/>

Meshtastic. (n.d.). Mesh broadcast algorithm. Retrieved from <https://meshtastic.org/docs/overview/mesh-algo>

Berto, R., Napoletano, P., & Savi, M. (2021). A LoRa-based mesh network for peer-to-peer long-range communication. *Sensors (Basel, Switzerland)*, 21(13), 4314. doi:10.3390/s21134314

3.2.3

Designing alongside the Meshtastic community

After having established how both the lower and upper layers of the network should work, the focus shifted towards hardware. Precisely the development of the LoRa radio devices that would have served as *nodes* capable of relaying messages within the meshnet, interfacing with the Meshtastic front-end companion apps and, as previously hinted, function as a generic LoRa physical layer for alternative upper layer protocols—in the eventuality *makers* required their network infrastructure to undertake tasks different from messaging or simply desired more room to experiment with the LoRa communication technique.

Embarking on this next phase eventually meant taking on the role of Hardware Developer within the Meshtastic community, as working in close contact with multidisciplinary experts, would have significantly sped up and simplified the iterative design process at hand.

The Meshtastic community, other than having its own [web-site](#)³⁸ and [GitHub page](#)³⁹, has a public [Discord server](#)⁴⁰ that, as of 2022, averages over 500 online members per single instance.

In recent years, Discord has been the protagonist of a resurgence in private forum-like platforms, which in both structure and ethos are reminiscing of their Web 1.0 counterparts—the only exception being the lack of a true website address, which is more than reasonable given large social networks having morphed into advertising platforms, internet domains do not exactly offer the exposure they used to at the turn of the millennium.

Early adopters of Discord servers were primarily gamers looking for a place to meet and organise playing together, but, as current circumstances like the Covid pandemic demanded people to develop a broader acquaintance with virtuality, Discord evolved to be the go-to platform for pretty much all forms of online gathering, both in the context of leisure and work—quickly priming its developers to implement features and quality-of-life improvements that even dedicated professional software like Microsoft Teams or Slack could envy.

This plethora of features comprises text channels and direct messages; voice channels with video calls and video sharing; social



networks and streaming platforms integration for listen-along and watch-along sessions; file sharing; and, most importantly, external plug-in integration—something which allows servers to interface with web infrastructure that exists beyond them or, potentially, to bring automation into play. Discord, having already proven itself to be a valid platform for both professionals and non-professionals alike, is unsurprisingly also popular among people that lie somewhere in between these two groups—groups which, given how they share enough overlapping for one single platform to meet both their needs, are hazily merging, further reinforcing the aforementioned hypothesis of contemporaneity blurring the lines between what is labour and what is not, transforming everyone into an exploited creator that is always creating.

Maker communities like Meshtastic have been adopting Discord as an all-comprehensive digital hub for sharing resources, engaging in mutual development help, discussing issues concerning their ecosystem and making server-wide announcements that are not too dissimilar to blog-like news feeds.

Joining the Meshtastic Discord meant starting a development process that, by being intertwined with the needs of the community, stayed ultimately true to the more hands-on objectives of this thesis. Partaking in discussions and observing how members reacted to updates has allowed this project to iteratively evolve towards a collectively endorsed hardware system.

To quote Haraway, making *oddkin* with people coming from different backgrounds has greatly contributed to both diminishing the individual workload and better designing, especially in the context of telecommunication engineering, which, if it were not for some of the people eager to help, would have been a challenging topic to face single handedly.

Joining the Meshtastic Discord meant starting a development process that, by being intertwined with the needs of the community, stayed ultimately true to the more hands-on objectives of this thesis. Partaking in discussions and observing how members reacted to updates has allowed this project to iteratively evolve towards a collectively endorsed hardware system.

To quote Haraway, making *oddkin* with people coming from different backgrounds has greatly contributed to both diminishing the individual workload and better designing, especially in the context of telecommunication engineering, which, if it were not for some of the people eager to help, would have been a challenging topic to face single handedly.

Crucially, being active in the Discord server led to pinpointing a fundamental factor for designing network infrastructure in this ecosystem. Modularity for the sake of future-proofing would have been incredibly important going forward, since it was im-

mediately clear that, given the speed at which the upper layer development was moving, the physical layer had to not only keep up but

also anticipate what could have been brewing on the software and firmware side of things, coming from both Meshtastic and LoRa developers at large—as, once again, the networks the Tendril Repository allows you to build, as much as they try to maximise *maker* insight by implementing a Meshtastic upper layer, should have also offered a solid physical layer capable of functioning for different, broader purposes. If it was not for this involvement with the community, there would have been no way to figure this out, as, as an outsider, it would have been nothing short of impossible to tell how fast things were advancing.

3.2.4

Building for modularity with RAK WisBlocks

Going forward in the process of hardware development, the first step that had to be taken came down to deciding which microcontroller was more suitable for the Tendril Repository *nodes*.

A microcontroller (MCU for microcontroller unit) is a small computer on a single integrated circuit (IC) chip designed for embedded applications where products and devices are controlled automatically.

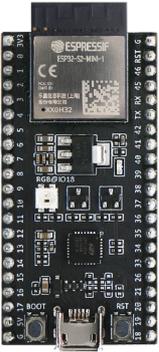
Most embedded systems have minimal requirements for memory and program length, with no operating system, and low software complexity. For this reason, embedded systems usually do not require recognizable input/output (I/O) devices, and may even lack any touchpoints for human interfacing. Typical I/O devices in embedded systems include switches, relays, solenoids, LEDs, small LCDs, sensors for environmental data and—as connectivity plays a key role in enabling these simple devices to coalesce into more complex systems of computation—radio frequency transceivers for Bluetooth, Wi-Fi and, more recently, low-power wide-area networking.

Given this rise in popularity for LPWANs in the IoT market, IC design companies followed suit by starting to offer products that took advantage of these long-range and low-power radio communication techniques—primarily by expanding their catalogue with MCUs integrating transceivers for Semtech’s LoRa, which is by far the most diffused among them.

Despite this was deemed to change shortly thereafter, during the earlier phases of this research into hardware development, the title of most popular MCU among Meshtastic’s community, and LoRa’s at large, belonged to Espressif’s ESP32 family of products, a series of low-power and low-cost chips that combine connectivity components for Wi-Fi, Bluetooth and, in some instances, LoRa.

ESP32s can be bought as both standalone chips or as full-featured development boards that function as alternatives to prod-

41 Espressif's *ESP32-S2-DevKitM-1* board



42 Adafruit's *HUZZAH32 ESP32 Feather* board



43 SparkFun's *ESP32 Thing* board



ucts like Arduino or Raspberry. These highly integrated boards are manufactured by Espressif⁴¹ themselves as well as by a wide variety of DIY-focused suppliers like Adafruit⁴² and SparkFun⁴³, something that has eventually led this MCU to gain remarkable popularity within *maker* communities, particularly within ones wanting to experiment with the LoRa communication technique and, given their vogue within these landscapes, *LoRa messengers*.

So much so that some eastern DIY-focused distribution companies even started selling off-the-shelf *LoRa messengers* employing ESP32s. Most popular among them are LILYGO's TTGOs—a growing series of products available in a wide variety of form factors that go as far as advertising compatibility with Meshtastic firmware and software, as the combination of Bluetooth and LoRa connectivity offered by ESP32 chips allows for the Meshtastic upper layer firmware and front-end interface companion apps to function together in integrating smartphones, tablets, computers and network *nodes* into a seamless user experience.

By combining open-source firmware and software with hardware sourced from their wide variety of suppliers, DIY-focused distribution companies like LILYGO can offer cheap plug-and-play devices capable of entering the fairly untapped off-the-grid messengers' market—a market which has mostly been populated by much more expensive solutions like Garmin's subscription-based inReach or goTenna's Mesh, the second of which, interestingly, also employs LPWAN mesh networking on ISM radio bands.

goTenna is a New York-based startup that has been developing mesh networking technologies for off-grid and decentralised communications since 2013, offering products with a very similar architecture to the LoRa devices that appear in this research, the only crucial exception being their software, firmware and hardware is entirely proprietary.

goTenna, on the shoulders of extensive glorifying press coverage, claims that their products offer unrivalled performance, going as far as dubbing themselves “The World's Leading Mobile Mesh Networking Platform” . . . Yet, as of 2022, it is impossible to buy any of their B2C products, as the company seems to have shifted to an on-request B2B offering mostly aimed at law-enforcement and private military contractors.

Considering goTenna's lacking product supply and generally unflattering reputation among amateur radio communities, it is safe to assume the unreliability of these claims.

It could also be further presumed that, by implementing a completely proprietary development stack, their products struggled to keep up with LPWAN technology and fell victim to obsolescence,

something that, as dangerous as it can get for bygone modes of development like goTenna's, never concerned interdependent open-source ecosystems like LoRa or Meshtastic, which, as of 2022, seem to be thriving.

Currently, in this ever-changing landscape, integrated products are deemed to struggle as IoT and, more precisely, LWPAN tech is moving too quickly for anyone to reasonably anticipate. Innovating in this space, can not be pulled off by simply trumping competitors' performance, instead, it has to be carried through in the long term by displaying a relentless resilience to the winds of change, blowing from both evolving microcontroller tech and IoT markets.

Currently, in this ever-changing landscape, integrated products are deemed to struggle as IoT and, more precisely, LWPAN tech is moving too quickly for anyone to reasonably anticipate. Innovating in this space, can not be pulled off by simply trumping competitors' performance, instead, it has to be carried through in the long term by displaying a relentless resilience to the winds of change, blowing from both evolving microcontroller tech and IoT markets.

The only player in the LPWAN technology field currently abiding by this philosophy is RAKwireless, an IoT IC company based in Shenzhen that has spent the last years focusing on developing a massive ecosystem of industrial-grade modular solutions for LoRa networks that comprises hardware, software and a vast supply of accessories—all of which are supported by extensive [documentation](#)⁴⁴ that goes well into detail on what these products do, how they work and how they can get implemented.

Contrary to ESP32's disparate and often conflicting family of highly-integrated products, RAKwireless' ecosystem is precisely designed for LWPANs, meaning that LoRa connectivity is not just tacked on obsolete IC architecture deemed to forever lag behind, but rather carefully resolved into a flexible system of boards and modules that can adapt to a vast variety of industrial needs as well as keeping up with developments coming from the fast-moving microcontroller tech landscape.

This ecosystem is called RAK WisBlock and, by combining industrial-grade components with *maker*-grade levels of documentation, offers industries and hobbyists alike a trustworthy platform for easily sourcing, understanding and implementing top-of-the-line components for low-power wide-area networking—moreover, doing so without the need of soldering or any electronic engineering knowledge, as WisBlock components click, snap and screw together without requiring any custom fabrication.

The WisBlock system is structured on WisBlock Base boards, platform carriers that allow for the easy plug-in of one WisBlock Core processing board (the MCU), multiple WisBlock Modules

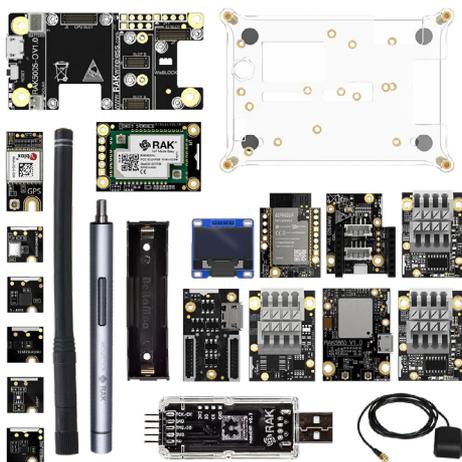
44



and, as of recently, even alternative Power Modules. To quickly have a glance at what is currently available:

- WisBlock Bases come in three sizes that substantially only differ in form factor and number of available Module slots.
- WisBlock Cores offer three alternative MCU architectures: a Raspberry-based one with a Semtech SX1262 LoRa transceiver; an ESP32-based one with Wi-Fi and BLE (Bluetooth Low Energy); and, most popular among the three, a Nordic nRF52-based one with BLE and an SX1262 LoRa transceiver.
- WisBlock Modules include Wireless Modules to further extend the communication capabilities of your board; Sensor Modules for analogue, digital, environmental and geographical data gathering; Interface Modules to expand the I/O capabilities of your board; Display Modules to add OLED and E-Paper screens to your device; and, finally, Storage modules to extend the memory capabilities of your board with different storage options like Flash, EEPROM, or SD-Card slots.
- Finally. WisBlock Power Modules greatly extend the power supply options available for your baseboard, which all come standard with a 3.7V input for Li-Po and Li-ion rechargeable batteries, a 5V one for solar power and a USB-C port.

45 RAKwireless' WisBlock Kit



Implementing the WisBlock⁴⁵ system into Tendril Repository *nodes* would have allowed this project to withstand obsolescence, keep up with Meshtastic's software development and offer *makers* another reliable and nurturing space for fully-supported hardware experimentation.

Unsurprisingly, In the last year or so, Meshtastic developers themselves shifted focus toward RAKwireless products and away from all the more hobbyist alternatives, as the WisBlock design system, not only perfectly covers all that is needed for making *LoRa messengers*, but is also en-

couraging developers to expand software features to implement more products from its line.

After Meshtastic opened the gates to the RAKwireless ecosystem by developing firmware for the RAK4631 (the Nordic nRF52-based WisBlock Core), it immediately became a strong ESP32 contender within the “LoRa messenger” landscape, so much so that the rise in sales of this MCU has eventually led the Shenzhen

IC company to actively collaborate with Meshtastic developers and community members.

Thanks to this ongoing collaboration, it became possible for *makers* to buy a WisBlock Meshtastic Starter Kit³¹ directly from RAKwireless and even speak with some of its employees in the Meshtastic Discord server.

As much as the Tendril Repository benefitted from this collaboration, it also actively contributed to making it happen, as this project was one of the first full-fledged hardware development efforts to take advantage of the WisBlock ecosystem.

To be more specific, Tendril Repository *nodes* employ the RAK4631⁴⁷ with the RAK19003⁴⁸, the smallest two-slotted WisBlock Base available, while also offering the option of installing the only other Meshtastic-compatible WisBlock Module: the RAK12500⁴⁹, a GNSS GPS Location Module for transmitting *node*'s position directly to the network without requiring BLE pairing with another device via the companion apps.

As more Modules become compatible with the Meshtastic upper layer, the list of optional WisBlock Modules for *nodes* is destined to grow, but, as of now, most WisBlock Modules, especially Sensor Modules, are still best suited for applications that go beyond the scope of P2P messaging—further reinforcing the hypothesis that Tendril Repository *nodes*, if *makers* ever wanted to venture down that road, could always offer a solid physical layer for many other LPWAN scenarios and LoRa-based upper layers.

After having established what MCU architecture the *nodes* were going to employ, going forward in the process of hardware development meant shifting attention to how the *nodes* themselves were going to be designed.

Doing so implied focusing back on the core objectives of this research, more precisely on how the full-*stack* learn-by-doing experience offered by the Tendril Repository was structured.

As each phase (Designing the network, Manufacturing the network and Deploying the network) was meant to give insight into and agency over some of the six layers of Bratton's *stack*, *makers* were going to have to engage in layer-specific learning as much as they were going to have to gauge a more general and cross-*stack* understanding of contemporaneity.

Priming this tentacular curiosity for the Here and Now, as previously said in the Background section of this research, was going to be achieved through network weaving. So, as much as building *nodes* was going to play a big part in this process, the focus had to always come back to the networks themselves.

To better reflect how *nodes* were part of larger-than-them LP-

46 RAKwireless Technology Limited, n.d.

47 Nordic nRF52840 BLE Core Module for LoRaWAN with LoRa SX1262



48 WisBlock Mini Base Board



49 GNSS GPS Location Module u-blox ZOE-M8Q



WAN architecture, and their manufacturing only a step in a full-stack development process, *nodes* could not just be represented as all-in-one devices like goTenna's Mesh radios or LILYGO's TTGOs, rather as a family of infrastructure elements that would have enabled *makers* to design their ad-hoc wireless mesh networks.

...*nodes* have been divided into classes designed to cover different roles within the network's architecture, and designed modularly to allow *makers* to freely upgrade or downgrade them between classes with a minimal loss of parts—effectively, bringing the same concept of flexibility for future-proofing that led to the employment of the WisBlock ecosystem beyond the scope of IC design.

For this reason, *nodes* have been divided into classes designed

to cover different roles within the network's architecture, and designed modularly to allow *makers* to freely upgrade or downgrade them between classes with a minimal loss of parts—effectively, bringing the same concept of flexibility for future-proofing that led to the employment of the WisBlock ecosystem beyond the scope of IC design.

Figuring out what these classes could have been, fundamentally, came down to understanding the relationship between *nodes* and meshnet topology, imagining incrementally complex scenarios that took advantage of this topology, and satisfying all the more

technical requirements that these scenarios would have brought up. *Nodes* have therefore been divided into four classes that aim to cover specific roles within meshnet infrastructure, more precisely: the Developer class, the Nomad class, the Resident class and the Airborne class.

1. The Developer class, DEV for short, has been designed to offer a simple and cost-effective point of entry to the Tendril Repository networking ecosystem, only sporting the minimum requirements to function as *nodes*. DEV *nodes* have been envisioned as capable of satisfying technical requirements belonging to meshnet application scenarios indicatively akin to the "Pursuing Privacy" one described in the Repository.

A scenario relying on a network topology in which coverage and mobility do not play any major role.

2. The Nomad class, NMD for short, has been designed to offer a flexible set of *nodes* for on-the-go connectivity, sporting an internal rechargeable battery, an optional 3dBi SubG Antenna and optional Salomon Quicklace Netting for backpacking and mountaineering. NMD *nodes* have been envisioned as capable of satisfying technical requirements belonging to all the meshnet application scenarios described in the Repository, effectively functioning as the most generic *node* class available,

displaying plenty of versatility in all network topologies.

3. The Resident class, RES for short, has been designed to offer a static *node* configuration capable of running autonomously for an indeterminate amount of time, sporting an internal rechargeable battery running off two Voltaic Systems 1 Watt 6 Volt Solar Panels, a Hi-vis *node* ID, and a ¼” Mounting Support for Tripods. RES *nodes* have been envisioned as capable of satisfying technical requirements belonging to meshnet application scenarios indicatively akin to the “Filling Coverage Gaps” one described in the Repository.

A scenario relying on a network topology in which permanent and extensive coverage over a determined area of land is fundamental.

4. The Airborne class, AIR for short, has been designed to offer a dynamic, temporary, and money-no-object alternative to RES *nodes*, sporting an internal rechargeable battery, a 3dBi SubG Antenna as standard and a ¼” Mounting Support for Drone Camera Brackets. AIR *nodes* have been envisioned as capable of satisfying technical requirements belonging to meshnet application scenarios indicatively akin to the “Providing Fallback Infrastructure” one described in the Repository.

A scenario relying on a network topology in which highly mobile and extensive coverage could be of life-saving importance.

Regardless of what class they belong to, all *nodes* have been designed to share a familiar design language consisting of modular FDM 3D printed enclosures and components sourced from a selected variety of trustworthy suppliers, all sharing the same ease of assembly and disassembly philosophy that *makers* would have already found in the WisBlock ecosystem.

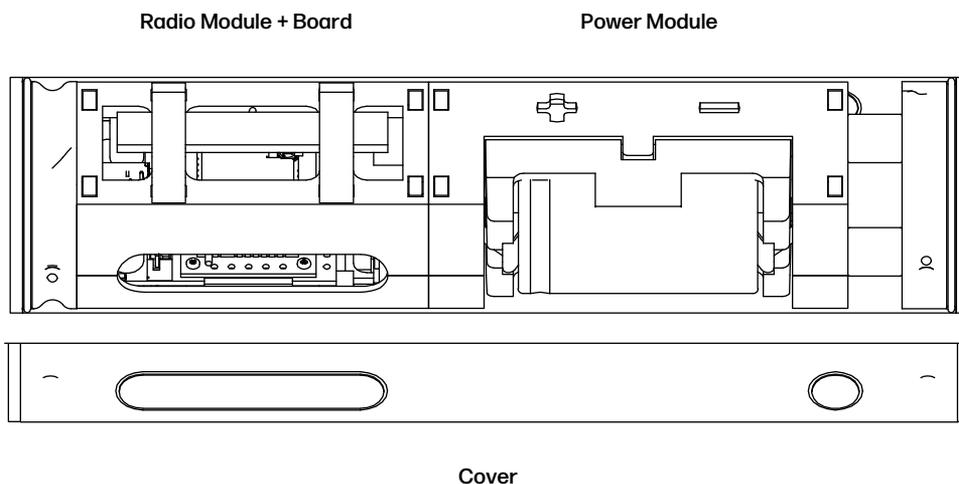
Regardless of what class they belong to, all *nodes* have been designed to share a familiar design language consisting of modular FDM 3D printed enclosures and components sourced from a selected variety of trustworthy suppliers, all sharing the same ease of assembly and disassembly philosophy that *makers* would have already found in the WisBlock ecosystem.

Precisely, *nodes*’ FDM 3D printed enclosures have been designed to assemble without the need for any fasteners, as snap fits and Lego components allow for a friendly and tool-free approach to *node* manufacturing—

whereas, other items sourced from Voltaic Systems, Adafruit and even Salomon would have offered *makers* reliable and long-lasting support throughout the process of network manufacturing and deployment. To grant *makers* the flexibility of easily switching between classes without wasting labour, time and re-

sources, *nodes*, have been structured in a number of cross-class sub-assemblies comprising both out-sourced components and self-fabricated parts:

- A. Board: the sub-assembly with the WisBlock Base, all its Modules and antennas—this sub-assembly is shared by all *node* classes and changes between them if there is a GNSS GPS Location Module installed.
- B. Radio Module: the sub-assembly with the Board and its 3D printed enclosure—this sub-assembly, too, is shared by all *node* classes, and only changes when accommodating a 3dBi SubG Antenna.
- C. Power Module: the sub-assembly with the battery holder, the battery and their 3D printed enclosure—this sub-assembly is shared by NMD, RES and AIR *nodes* without changes.
- D. Cover: the sub-assembly comprising all the parts that make up the outer enclosure in which the Radio and Power Modules slide in—this sub-assembly differs across *node* classes, however, sharing many parts between them.
- E. Attachments: the sub-assembly comprising all 3D printed parts and components outside of the Cover—this sub-assembly greatly differs across *node* classes, sharing close to no parts between them.



The basic modular architecture of an NMD *node*

Carrying modularity throughout *node*'s design could allow *makers* to better understand that in each and every phase, no matter the amount of effort, time and money they are willing to sacri-

fice, the Repository is capable of adapting to their commitment—regardless of how low or how high this commitment might be, the Repository is able to reliably output an exhaustive result.

By employing modularity throughout the full-*stack* designing, manufacturing and deploying exercise, the Repository can effectively become *commitment-dynamic*, as each and every step, with this concept in mind, can be designed for incremental complexity, going as far as not putting a limit on how far things can be taken.

By employing modularity throughout the full-*stack* designing, manufacturing and deploying exercise, the Repository can effectively become *commitment-dynamic*, as each and every step, with this concept in mind, can be designed for incremental complexity, going as far as not putting a limit on how far things can be taken.

able to adapt to their efforts just as much as other people's, evolving hand in hand with their needs and the needs of the ever-changing ecosystems they work for.

To be more specific, as LoRa, Meshtastic and RAKwireless developers are *makers* too, the Repository should have been capable

to adapt to their efforts just as much as other people's, evolving hand in hand with their needs and the needs of the ever-changing ecosystems they work for. Therefore, it becomes worth believing that future-proofing, rather than being a consequence of primacy, comes as a direct result of designing *commitment-dynamic* systems capable of adapting to the winds of change, wherever they might blow from.

Ultimately, thanks to modularity, the outputs of this research manage to stand a chance against obsolescence by, not only embracing the agency people exercise over them, but also nudging them to do so throughout the process.

Even in the two fringe cases in which the impact *makers* exercise on the project is non-existent or, on the other end of the spectrum, comes as a direct result of their job, the Repository can persist unfazed in its original intent.

Works cited:

RAKwireless Technology Limited. (n.d.). WisBlock Meshtastic Starter Kit. Retrieved December 3, 2022, from RAKwireless Store website: <https://store.rakwireless.com/products/wisblock-meshtastic-starter-kit>

3.3

Retrospective

Designing *commitment-dynamic* systems, as this research might have proven, is something that can be done a lot more easily in the landscapes of self-production and open-source design, environments in which the outputs themselves become heterogeneous byproducts of people's will and self-determination.

In this context, design becomes truly democratic, in the sense that, notwithstanding the size of their following, projects can persist free of any rigidity and market-related motivation—the liquefaction of conventional capitalogenic design processes is met with the resurgence of new ideas.

Hypothetically, these spaces could nurture concepts that would have otherwise been incompatible with the demands the extractive “automatas” normally inscribe in our labour and in the way we think of the world—to quote Berardi, ultimately turning the inconceivable back to possible.

To conclude, the Tendril Repository, in order to take on the function of a DIY project that could have offered *makers* a chance to reclaim agency over contemporaneity, had to transform into a *commitment-dynamic* design system capable of correctly responding to it throughout all of its phases and touchpoints.

Doing so required a complete reframing of conventional design processes that has ultimately led the project to become heavily interdependent with its public, as people’s agency over its outputs, not only represents the ultimate aim of the exercise described in the Repository, but also the very propelling force behind its past, present and future development.

To conclude, the Tendril Repository, in order to take on the function of a DIY project that could have offered *makers* a chance to reclaim agency over contemporaneity, had to transform into a *commitment-dynamic* design system capable of correctly responding to it throughout all of its phases and touchpoints.

Doing so required a complete reframing of conventional design processes that has ultimately led the project to become heavily interdependent with its public, as people’s agency over its outputs, not only represents the ultimate aim of the exercise described in the Repository, but also the very propelling force behind its past, present and future development.

For self-production, interdependence embodies a horizon in which *commitment-dynamic* systems can benefit all the parties involved and, potentially, monetize these exchanges of value adequately.

For as long as self-production gets wrongfully interpreted as a struggle for independence from the *stack* in which it happens, there will never be any hope for this prospect to ever happen, as “Under the current regime of so-called independence, dictated by an increasingly small number of platforms, we are divided and . . . all too easily conquered”.⁵⁰

While referring to artist communities on the fringes of the electronic music landscape, artist, professor and researcher Mat Dryhurst also states that “A vision of interdependence acknowledges that individual freedoms thrive in the presence of resilient networks and institutions. . . . Adopting an interdependent logic, artist communities might generate the collective bargaining power to insist on having a say in the technical and economic systems that impact our future”.

This thesis, by being open and transparent about its reliance on people’s agency over its outputs, hopes to highlight the fact that the Tendril Repository does not represent an opportunity to self-

50 Dryhurst, 2019

ishly opt out of contemporaneity, but a deeply intertwined prompt for trying and building a better one. Escaping the *new extractivism* is not an objective that can be pursued individually, rather has to be interpreted as a collective and long-term process that depends on community-wide commitment—something that must be further extended to any other objective propelling self-production.

Works cited:

Dryhurst, M. (2019, April 9). Band together: why musicians must strike a collective chord to survive. *The Guardian*. Retrieved from <https://www.theguardian.com/music/2019/apr/09/experimental-musicians-must-strike-a-collective-chord-red-bull-music-academy-closing>

Full Bibliography:

AkarshA. (2019, March 1). LoRa Messenger for Two Devices for Distances Up to 8km. Retrieved December 2, 2022, from Instructables website: <https://www.instructables.com/LoRa-Messenger-for-Two-Devices-for-Distances-Up-to/>

Berardi, F. (2020). *Futurability: The age of impotence and the horizon of possibility*. London, England: Verso Books.

Berto, R., Napoletano, P., & Savi, M. (2021). A LoRa-based mesh network for peer-to-peer long-range communication. *Sensors* (Basel, Switzerland), 21(13), 4314. doi:10.3390/s21134314

Bratton, B. H. (2016). *The Stack: On Software and Sovereignty*. London, England: MIT Press.

Cockfield, B. (2022, May 25). Long-distance text communication with LoRa. Retrieved December 2, 2022, from Hackaday website: <https://hackaday.com/2022/05/25/long-distance-text-communication-with-lora/>

Coward, C. (2021, February 6). This LoRa messenger is perfect for texting in the wilderness. Retrieved December 2, 2022, from Hackster.io website: <https://www.hackster.io/news/this-lora-messenger-is-perfect-for-texting-in-the-wilderness-108ed61aaace>

Degree Programme of Integrated Product Design. (2022). Retrieved December 2, 2022, from Polimi.it website: https://www8.ceda.polimi.it/manifesti/manifesti/controller/extra/RegolamentoPublic.do?af_currentWFID=main&EVN_DEFAULT=evento&aa=2021&k_corso_la=1261&lang=EN

Dryhurst, M. (2019, April 9). Band together: why musicians must strike a collective chord to survive. *The Guardian*. Retrieved from <https://www.theguardian.com/music/2019/apr/09/experimental-musicians-must-strike-a-collective-chord-red-bull-music-academy-closing>

Easterling, K. (2014). *Extrastatecraft: The Power of Infrastructure Space*. London, England: Verso Books.

European Conference of Postal and Telecommunications Administrations. (1997). ERC/REC 70-03. Retrieved from European Conference of Postal and Telecommunications Administrations website: <https://docdb.cept.org/document/845>

Haraway, D. J. (2016). *Staying with the Trouble: Making Kin in the Chthulucene*. Durham, NC: Duke University Press.

International Organization for Standardization. (1994). ISO/IEC 7498-1:1994 Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. Retrieved from International Organization for Standardization website: <https://www.iso.org/standard/20269.html>

Joler, V. (2020). *New Extractivism*. Retrieved December 2, 2022, from New Extractivism website: <https://extractivism.online/>

LoRa Alliance. (2021). RP002-1.0.3 LoRaWAN® Regional Parameters. Retrieved from <https://lora-alliance.org/wp-content/uploads/2021/05/RP-2-1.0.3.pdf>

Mari, E. (1974). *Proposta per un'autoprogettazione*. Milano, Italy: Centro Duchamp/Galleria Milano.

Mari, E. (2002). Autoprogettazione. Mantova, Italy: Corraini.

!Mediengruppe Bitnik, Silvio Lorusso, Domenico Quaranta, Luciana Parisi, Felix Stalder, and Janez Jansa. (2020). Hyperemployment: Post-Work, Online Labour and Automation. Roma, Italy: Nero Editions.

Meshtastic. (n.d.). Mesh broadcast algorithm. Retrieved from <https://meshtastic.org/docs/overview/mesh-algo>

RAKwireless Technology Limited. (n.d.). WisBlock Meshtastic Starter Kit. Retrieved December 3, 2022, from RAKwireless Store website: <https://store.rakwireless.com/products/wisblock-meshtastic-starter-kit>

Seller, O. B. A. (2014). Patent No. 9647718B2. Retrieved from <https://patents.google.com/patent/US9647718B2>

Semtech Corporation. (2022). LoRa and LoRaWAN: Technical overview. Retrieved from <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>

Wark, M. (2021). Capital Is Dead: Is This Something Worse? London, England: Verso Books.

Zuboff, S. (2019). The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power. New York, NY: PublicAffairs.

(2020). ITU Radio Regulations, International Telecommunication Union.

FRANCESCO ZURLO

FEDERICO BASSI

ALESSANDRO IANNIELLO