

INS
PEDRALBES
2019-2020

IoT MARÍTIMA, LA DETECCIÓN DE
MEDUSES

Autors: Joel Bello Vidal i Marc Soler Pons
Tutor: Ernest Tomàs

1 INTRODUCCIÓ	3
1.1 Objectius	4
2 PROJECTE IoT	6
3 APLICACIÓ: UNA IoT MARINA	10
3.1 Un problema a les costes marines	10
3.2 El fenomen de l'aparició de les meduses	11
3.3 Factors indicatius de l'aparició de meduses.	12
3.4 Enfocament del projecte: projecció ecològica i científica	13
3.5 State of the Art per detecció de meduses	13
3.6 Estimació a partir d'una xarxa de sensors	15
4 ARQUITECTURA DEL SISTEMA	16
4.1 SigFox vs LoRa	16
4.2 Arquitectura final	19
5 CONSTRUCCIÓ DEL NODE	22
5.1 Tecnologia a utilitzar	22
5.2 Models Arduino a utilitzar	24
5.3 Els sensors	28
5.3.1 Temperatura	29
5.3.2 Salinitat	30
5.3.3 Brúixola	31
5.4 Estructura mecànica	32
5.4.1 Caixa de connexions IP68	32
5.4.2 La Carcassa orientadora	33
5.4.3 La antena	34
6 CICLES DE CONSTRUCCIÓ I PROVA	35
6.1 Prova dels sensors amb Arduino UNO	35
6.2 Prova de la xarxa TTN	40
6.3 Exportació dades al cloud Thing Speak	42
6.4 Prova d'estanquitat	45
6.5 Prova d'orientació	46
6.6 Prova final al ICM	47
7 CONCLUSIONS I TREBALLS FUTURS	49
8 ANNEXOS	51
9 REFERÈNCIES	64

1 INTRODUCCIÓ

En un món cada cop més interconnectat, comencen a aparèixer termes més usuals com ara: *Intel·ligència Artificial (AI)*, *BigData*, *CloudComputing*, *SmartCities* o *Industria 4.0*. Intuïm que totes aquestes sigles ens indiquen que el món s'aboca a una nova era tecnològica, que no acabem de comprendre si realment representarà un canvi en les nostres vides o almenys en algunes àrees de coneixement.

Si poguéssim resumir-ho, sembla que les noves tecnologies van encarades a fer prediccions o estimacions per millorar algun tipus de resultat.

Per altra banda, la optimització dels dispositius electrònics ha permès que cada cop sigui més senzill el disseny i implementació de qualsevol sistema intel·ligent que tingui connectivitat amb la xarxa d'internet. D'aquí apareix el terme *IoT* o Internet de les Coses. I aquesta àrea tecnològica la tenim a la nostra disposició, igual com ara la robòtica o com fer aplicacions per mòbil.

Per internet existeixen multitud de projectes orientats a establir prediccions a partir d'un recull de dades. Potser un que exemplifica el que volem aconseguir, perquè barreja també ciència i tecnologia, és el "The Squirrel Cafe Predicting winter weather condition"⁽⁶⁾. El projecte és un prototip connectat a una caixa de fusta a l'aire lliure, on sempre hi ha una quantitat de cacauets i intenta predir si el proper hivern serà molt fred segons la quantitat de cacauets que els esquiroles es mengen

cada dia. La correlació entre aquestes dades i les condicions de fred han estat significatives.

Podem aportar nosaltres, amb pocs recursos, noves solucions tecnològiques a projectes que ens interessin? Bé, doncs aquest treball de recerca intenta ser-ho.

En els propers capítols mirarem de desglossar quines tecnologies hem treballat en el camp de les *IoT* per oferir un sistema alternatiu en l'àmbit de la ciència marina, i amb el temps es podria ampliar l'estudi en el camp de *BigData* i *CloudComputing*.

1.1 Objectius

Ens hem proposat els següents objectius per assolir el nostre projecte:

- + Posar a la pràctica una idea surgida en anys anteriors i que en aquells moments ens semblava un repte insalvable. En el moment de començar aquest TR i reestructurant certs aspectes, creiem que ara està al nostre abast.
- + Gràcies a l'avanç de la tecnologia de petits prototips i de les comunitats que permeten connectar diferents xarxes, aconseguir realitzar un prototipus model de baix cost i que funcioni correctament i de manera eficaç dins d'una xarxa més extensa.
- + Utilitzar tecnologies de codi obert en la mesura del possible.

- + Aportar a la ciència marina una alternativa per fer una predicció de l'aparició de certes espècies tan a les costes del mediterrani com a altres zones.
- + Que el nostre projecte arribi a més gent per millorar-lo, ampliar-lo i fer-lo apte com a possible alternativa a la actual.

2 PROJECTE IoT

Com a context tecnològic el nostre projecte es situa en l'àmbit de la internet de les coses, coneguda per les seves sigles IoT.

L' IoT, "*Internet of things*", és un nou concepte aparegut en els últims en anys relacionat amb la interconnexió d'objectes quotidians amb la xarxa internet. Qualsevol d'aquests objectes poden ser visibles i interaccionar entre ells sense la necessitat de cap activitat humana. Imaginem-nos una nevera connectada a internet que ens comuniqui, per exemple, quan perd temperatura o quan un producte ja ha caducat. Parlem d'una tecnologia o un concepte tecnològic revolucionari ja que fins ara les connexions via internet només eren entre humans.



Figura 2-1. Camp de treball on es pot aplicar les IoT.

En aquest sentit, entrem en una nova era industrial que obre multitud de perspectives de com millorar i optimitzar la nostra vida.

La connexió d'aquests objectes és possible dins d'un entorn on disposem de connectivitat WIFI o Bluetooth, com per exemple a casa o a la empresa. Aquests dos estàndards ja són àmpliament utilitzats i algunes companyies ja estan oferint solucions innovadores en el mercat. Un dels productes més cridaners que es va instal·lant a les llars, és l'assistent de la marca Google o Amazon. A partir d'unes ordres per veu, el dispositiu és capaç d'activar qualsevol aparell de la llar que estigui connectat a la xarxa local. Si el sistema és capaç d'aprendre per ell mateix, és a dir, que a partir d'una ordre no molt clara pugui obtenir el resultat desitjat, estaríem parlant d'intel·ligència artificial *AI*. En aquest sentit, és necessari un emmagatzematge al núvol dels events produïts per tal que l'algoritme actuï correctament. I com més dades, millor a de ser la resposta.

En la Figura 3-2 tenim representades algunes tecnologies que s'utilitzen per realitzar un IoT, en funció de l'aplicació destinada. Potser per a la gran majoria de nosaltres, a part de la Wi-Fi i de la RFID utilitzada en les etiquetes dels productes dels comerços, desconeixem de l'existència de les altres.

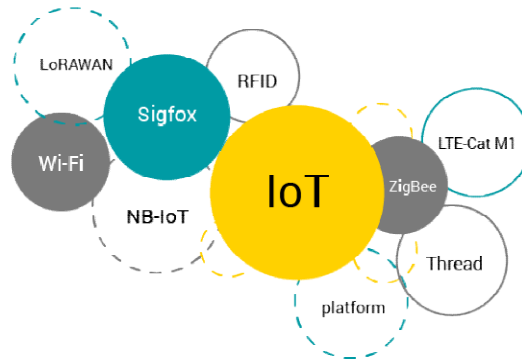


Figura 2-2. Tecnologies utilitzades en IoT

Quines opcions tenim quan sortim de la nostra connexió Wifi o Bluetooth? En un principi, sabem que podem connectar els aparells a la xarxa d'internet amb el nostre mòbil, a través de la xarxa GSM/UMTS mitjançant les operadores locals. També coneixem la possibilitat d'intercanviar dades a través de la cobertura per satèl·lit. Però aquestes són opcions que resulten cares, bàsicament perquè estan pensades per transferir molta quantitat de dades.

Una particularitat de la nova arquitectura IoT és la baixa quantitat de dades que es mouen en un sentit o altre. Per exemple, no és necessari enviar cada segon la temperatura de la nevera perquè és una magnitud que necessita més temps per a que el canvi sigui significatiu. Per tant, és necessari de la aparició de diferents tecnologies de comunicació que optimitzin aquest recurs.

Dins del món de la connectivitat s'ofereixen diferents rangs de connexió en funció de la distància.

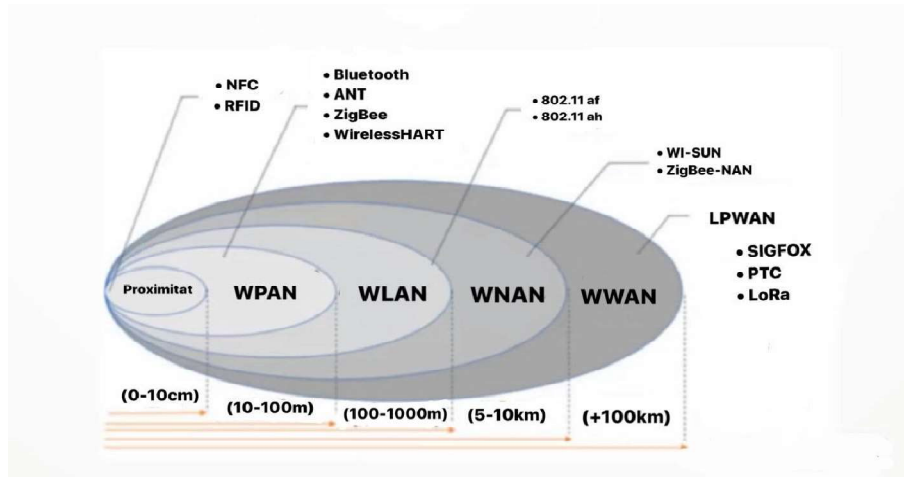


Figura 2-3. Gràfic dels tipus de xarxes inalàmbriques i les seves especificacions.

En aquest esquema podem veure els diferents tipus de rangs de connectivitat en funció de la distància i els tipus de xarxa que s'utilitzen en cada cas.

L'àmbit de treball del nostre projecte es concentra en una xarxa WWAN, "*Wireless Wide Area Network*". Aquest tipus de xarxa inalàmbrica ens dóna la possibilitat d'obtenir una connectivitat a llargues distàncies. Dins de les connexions WWAN funcionen diferents tipus de especificacions de xarxa com SIGFOX, PTC, 3G o LoRa.

En el capítol 5 sobre l'arquitectura del sistema, descriurem dues de les tecnologies que podríem realitzar per la nostra IoT.

3 APLICACIÓ: UNA IoT MARINA

3.1 Un problema a les costes marines

Actualment l'aparició de meduses a les costes catalanes és més freqüent i la seva presència comporta una sèrie de desavantatges. Un dels principals problemes que comporten les meduses a les nostres costes és el turisme ja que la major part dels turistes que visiten la nostra ciutat estiueja a les nostres costes gràcies a la quantitat i qualitat de les platges i les condicions del mar mediterrani que poden gaudir durant gran part de l'estiu. Degut a l'aparició de les meduses el turistes estan atemorits d'entrar a l'aigua i decideixen buscar alternatives fora de les costes que fa una baixada del turisme que té un efecte directe en el consum, sobretot en l'àmbit de la restauració (hotels, restaurants...) tenint així un impacte negatiu a la nostre economia.

Aquest animal marí a més posa en perill la nostra salut, tant a els turistes com als catalans ens posen en perill per les picades que rebem si les toquem que poden donar a varies infeccions o fins i tot la mort⁽¹⁾ si no som conscients del perill d'aquesta espècie.



Figura 3.1-1. Article recent del diari “20 minutos” sobre les plagues de meduses a les costes catalanes.

3.2 El fenomen de l’aparició de les meduses

L’aparició massiva d’aquest invertebrat és deguda a una sèrie de circumstàncies que impliquen un canvi en l’aigua i afecta directament a la vida d’aquests animals.

El primer factor implica un afavoriment a la seva reproducció i és el canvi climàtic⁽²⁾, l’augment de les temperatures afavoreix el procés biològic no tan sols de la seva reproducció si no que també la del plancton, del qual les meduses s’alimenten. El canvi climàtic a més a disminuït les pluges sobretot a les èpoques d’hivern factor que juga a favor de la temperatura ideal per a les meduses.

La contaminació de l’aigua també⁽²⁾ afecta al creixement i alimentació de les meduses ja que l’abocament al riu de certes substàncies augmenten sobretot els nitrats i fosfats en l’aigua, elements essencials en la reproducció del plàncton i el zooplàncton que és font d’alimentació de la medusozoa.

Un tercer factor molt important en l’augment de la presència de les meduses és la desaparició dels seus depredadors trencant d’aquesta manera la cadena tròfica.

L'exploració de la pesca, la contaminació o altres factors fan augmentar la desaparició d'aquests depredadors com les tortugues per exemple.

Un últim factor que afavoreix l'aparició de les meduses són les noves construccions⁽²⁾ com: molls, espigons o zones guanyades al mar ja que es poden convertir en zones d'assentament per a les meduses i llocs d'abocament de residus urbans que al igual que la contaminació al riu afavoreixen el creixement de plàncton i aliment per a les meduses.

3.3 Factors indicatius de l'aparició de meduses.

Algunes de les característiques de l'aigua del mar són indicadores de la possible presència de meduses tals com la quantitat de zooplàncton, temperatura, salinitat o corrents marines. Nosaltres ens centrem en aquests últims tres factors: la salinitat, la temperatura i la direcció i força de les corrents marines.

La salinitat afecta a la densitat de l'aigua; si la salinitat del mar disminueix també ho fa la densitat de l'aigua i d'aquesta manera les meduses pugen del fons marí a la superfície gràcies al canvi de la propietat de l'aigua de l'aigua. Si mesurem aquesta magnitud podem arribar a delimitar uns paràmetres que ens indiquin la possible aparició de meduses a la superfície.

L'altre factor que ens indica la possible aparició d'aquests invertebrats és la temperatura. Aquest factor també està relacionat amb la densitat de l'aigua. Quan la temperatura de l'aigua del mar augmenta fa disminuir la densitat del fluid i al igual que amb la baixa salinitat les meduses surten a la superfície.

L'últim factor que utilitzarem per predir l'aparició de les medusozoos a les costes són les corrents marines. Sabent que les meduses és un animal que es mou gràcies a les corrents si mesurem la direcció i la força de les corrents podem indicar si les meduses es dirigeixen cap a les costes o cap a mar obert, el seu hàbitat natural.

3.4 Enfocament del projecte: projecció ecològica i científica

Tot i que la seva aparició a les zones costaneres és un gran inconvenient el paper de les meduses en els nostres mars i oceans és important⁽³⁾. Les meduses es poden criar, menjar i estudiar i són una de les espècies més antigues del nostre planeta. Aquest senzill animal ens aporta molta informació sobre la regeneració de teixits humans o sobre l'envelliment ja que algunes de les espècies de medusozoa són potencialment immortals.

Per tant, la finalitat del nostre projecte no és la desaparició si no el control i la detecció eficaç d'aquesta espècie i pot ser de moltes altres espècies.

3.5 State of the Art per detecció de meduses

“*The State of the Art*” en la metodologia per la qual es detecten les meduses a les nostres costes són o bé molt cars o molt rudimentaris.

El mètode actual, més econòmic però que alhora resulta molt rudimentari per a la detecció, és simplement un operari que fa una navegació per la costa controlant l'aparició d'aquest animal. Resulta una alternativa ineficient i imprecisa ja que l'operari no pot abarcar tota la costa i tan sols visualitzarà les meduses si es situen a poca profunditat.

Un altre mètode també utilitzat per la detecció no tan sols de meduses si no també d'altres espècies marines és via satèl·lit. Aquesta metodologia implica una inversió de capital molt elevada i només en detecta bancs de peixos, meduses o altres espècies.

Un altre forma de detectar la presència de meduses en diferents punts de les nostres costes és mitjançant la col·laboració ciutadana amb una aplicació, desenvolupada per uns alumnes de la Universitat Politècnica de València, que permet a qualsevol usuari advertir de la presència de meduses en un lloc en concret, el tipus de medusa, la grandària la quantitat d'individus.

L'inconvenient que presenta aquesta alternativa és l'impresició. Els advertiments que els usuaris poden publicar poden ser tan sols d'una única medusa que hagin pogut veure i l'aplicació no mostra amb exactitud el lloc on l'usuari ha determinat l'aparició de l'animal. Aquestes inexactituds poden alarmar o donar informació equivocada de l'existència d'aquests invertebrats.

3.6 Estimació a partir d'una xarxa de sensors

El mètode alternatiu i innovador que nosaltres volem presentar per la detecció d'aquestes meduses es basa en l'obtenció de les dades d'aquests factors de mesura que indiquen la presència de les medusozoes i així poder fer una predicció.

Coneixent els nivells de salinitat, temperatura i la força i direcció de les corrents marines podem arribar a fer una estimació més precisa de la possibilitat de l'aparició de meduses en les costes. Si, per exemple, la salinitat de l'aigua del mar baixa, la temperatura augmenta i les corrents marines es direccionen cap a la costa parlem d'una probabilitat alta de l'aproximació de meduses cap a les costes. Si pel contrari la salinitat és molt alta, la temperatura de l'aigua és baixa i les corrents no es dirigeixen cap a la costa o ni tan sols hi ha grans corrents, ens trobariem en una situació on la probabilitat de l'existència de meduses a les costes seria molt baixa.

Entre aquests tres paràmetres i aquestes dues situacions fem les estimacions per predir o advertir de la presència d'aquest animal marí.

4 ARQUITECTURA DEL SISTEMA

Ara que sabem en quin àmbit està ubicat el nostre sistema, necessitem escollir com enviar les dades a llargues distàncies, de l'ordre de kilòmetres. Les característiques del nostre dispositiu, és que ha de ser autònom i per tant hem de prioritzar el seu baix consum energètic. La necessitat d'enviar dades ha de permetre que tingui un radio-enllaç incorporat i que es connecti a una xarxa WLAN. Per tant, el nostre dispositiu passarà a ser un Node dins d'aquesta xarxa. Tot i que el nostre Node podria enviar i rebre dades, l'utilitzarem tant sols com a emissor de missatges.

Després d'una recerca d'aplicacions realitzades en les comunitats IoT, i amb el criteri d'escollir aquella que ens assegurï de tenir prou informació per l'èxit del nostre sistema, arribem a dues especificacions on haurem de decidir quina ens presenta més avantatges: SigFox o LoRa.

4.1 SigFox vs LoRa

SigFox és una xarxa, privada i de pagament, IoT pensada per treballar en uns baixos consums i al marge del desplegaments de telefonia. És una xarxa alternativa a les solucions de connectivitat com 5G o 3G que serien les més comuns, és totalment aliena a les xarxes existents ja que SigFox compta amb un infraestructura pròpia d'antenes i servidors.

SigFox ofereix una connectivitat de llarg abast (entre el WIFI i la comunicació mòvil). Utilitza una tecnologia UNB que treballa amb baixes velocitats de transferències de dades (10 a 1.000 bits per segon). SigFox transmet a 200Hz d'amplada de banda i per tant requereix poca potència en la transmissió de dades a llargues distàncies. Es compleix aquí la proporció inversa: com menys dades es transmeten, a més distància arribarà el missatge. Aquesta característica de les xarxes WLAN ve determinada sobretot per la taxa d'error: com menys dades ha d'enviar el missatge, menys probabilitat d'error tindrà el receptor de rebre el missatge inicial.

Al igual que LoRa, SigFox és una especificació de xarxes d'àrea àmplia (WAN). No obstant, SigFox no és una xarxa lliure i a disposició de qualsevol usuari. És la millor opció si l'usuari és una empresa privada o si volem connectivitat en punt concrets de la geografia on no tinguem cobertura d'altres xarxes. Actualment, SigFox ja cobreix tota la geografia Europea.

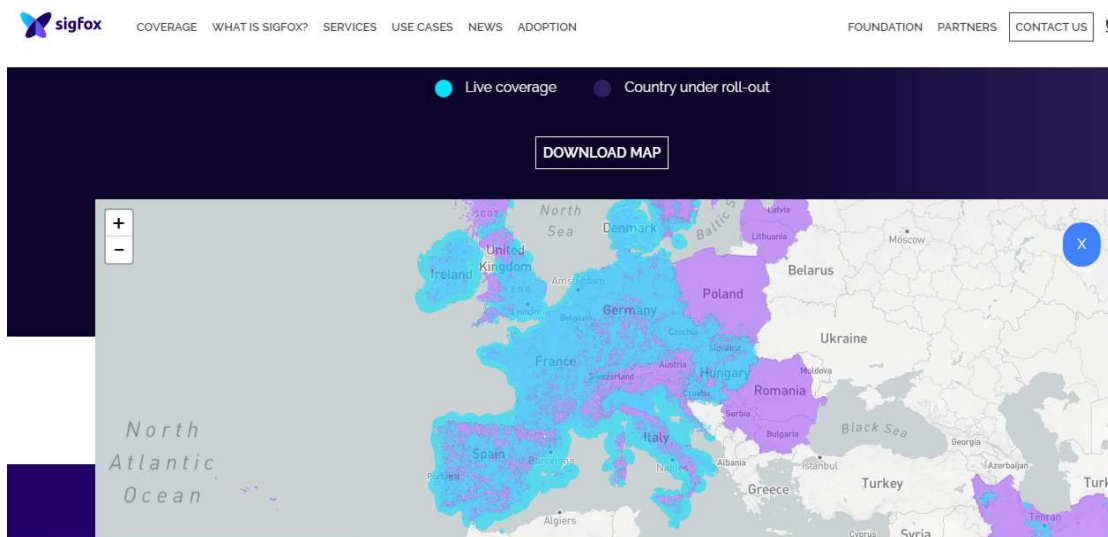


Figura 4.1-1. Zones de cobertura en blau de la xarxa SigFox.

LoRa o LoRaWAN és una especificació de xarxes LPWAN “*Low Power Wide Area Network*”, és a dir, xarxes de baixa potència i àrea àmplia.

LoRa és també una tecnologia de radiofreqüència amb standard obert, creat per moltes companyies amb la intenció de crear una xarxa d'interconnexió formada tant per particulars com organitzacions de qualsevol escala. Si el teu radio-enllaç aconsegueix els protocols LoRa, et podràs connectar a alguna de les seves xarxes.

Entre altres, la organització *TTN*, “The Things Network” ha adoptat standard Lora i la seva filosofia és el creixement de la xarxa per la pròpia comunitat d'usuaris. Aquesta xarxa va començar a créixer a Holanda fa ara un parell d'anys. En aquest cas, és la pròpia comunitat la encarregada d'anar incorporant els punts de connexió o passarel·les (*Gateways*) arreu de la geografia. Aquests *Gateways*, alhora, estan connectats a la xarxa internet; per tant són un pont entre dues xarxes diferents.

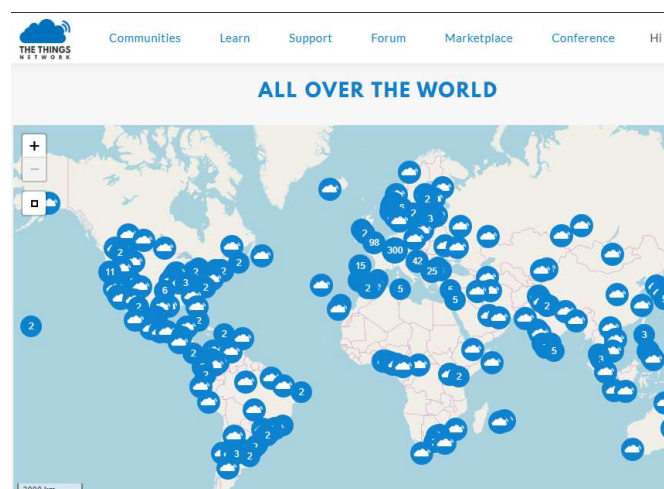


Figura 4.1-2. Gateways TTN disponibles al Desembre de 2019.

Actualment a la zona marítima que ens interessa, ja hi han certs nombres de Gateways en funcionament, ja sigui per particulars a les seves llars, com per organitzacions de qualsevol tipus. En el següent mapa s'observa com a la zona de Barcelona existeixen la major quantitat d'aquestes. És previsible que el nombre es vagi incrementant al llarg dels anys i permetre la connexió de dispositius de manera gratuïta.

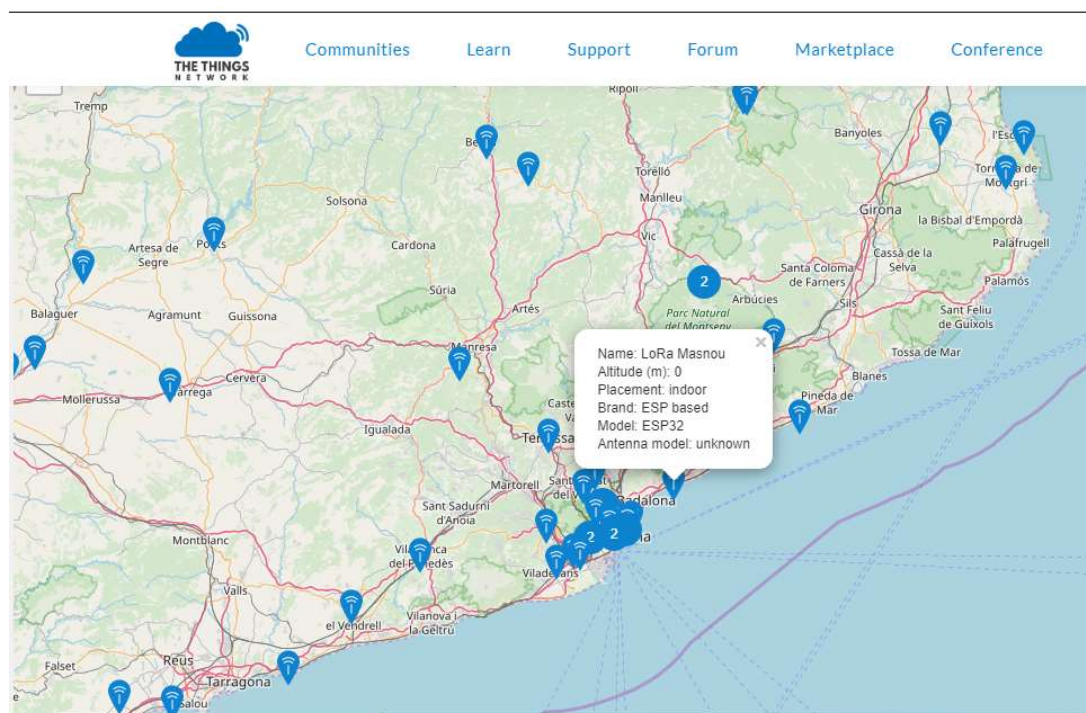


Figura 4.1-3. Mapa de Gateways a la zona del litoral, marcant un de la zona del Masnou.

A Europa, la freqüència de treball LoRa és d'uns 868 Mhz, a diferència del EUA que es troba cap als 925 Mhz.

4.2 Arquitectura final

Pel que fa al sistema de comunicació, l'elecció de la tecnologia LoRa a través de la comunitat TTN és bàsicament per la gratuïtat del sistema.

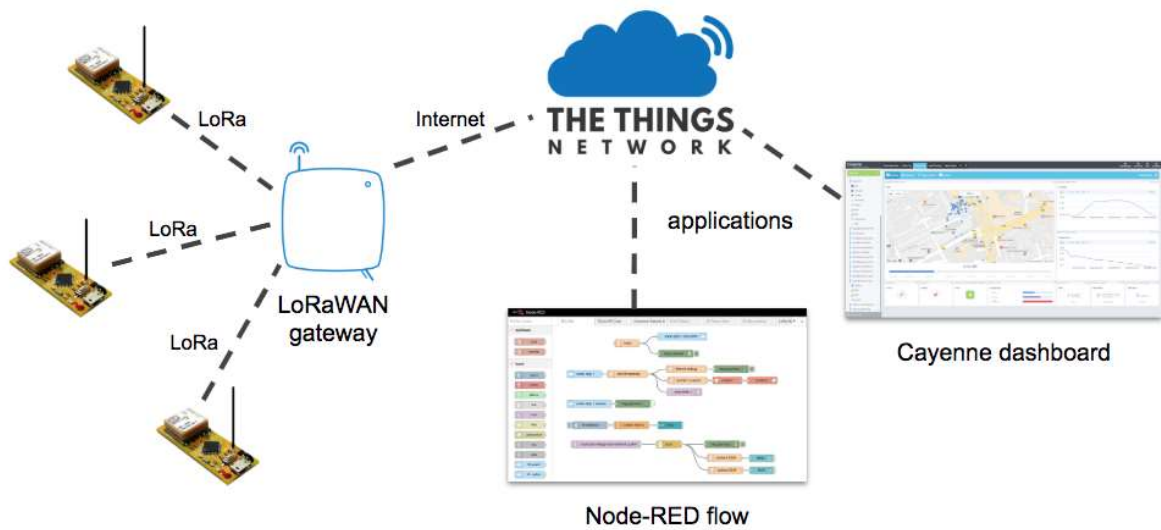


Figura 4.2-1. Esquema de la comunicació a *The Things Of Network*.

Aprofitarem els enllaços ja creats i ens afegirem a la plataforma per crear una aplicació que reculli dades des de un Node que haurem de construir.

Aquests Node, serà un dispositiu que recollirà dades del mar i les enviarà cap a l'aplicació en intervals de temps suficients. Tot i que no hem analitzat quina seria la millor resposta, hem pensat que un interval de 10 minuts seria un compromís entre alguna variació significativa de les dades i l'aprofitament de la bateria que hauria d'incorporar el sensor.

Hi hauria la possibilitat de que el nostre prototipus tingués algun element de recàrrega mitjançant plaques solars i així evitar que cada certs anys s'hagin de canviar les bateries. Però això faria augmentar la complexitat i costos del dispositiu i per tant ho hem descartat.

A grans trets i per tal de reduir la complexitat del Node, el dispositiu estarà enfonsat dins d'una caixa de connexions entre 1 i 3 metres del nivell del mar, amb els sensors incorporats. També, protegit per una carcassa d'un material plàstic com el metacrilat i que ahora realitzarà la funció d'orientació segons el corrent marí. Tot aquest cos estarà subjectat per un cable coaxial d'antena i al extrem superior hi connectarem una antena omnidireccional amb un cert guany.



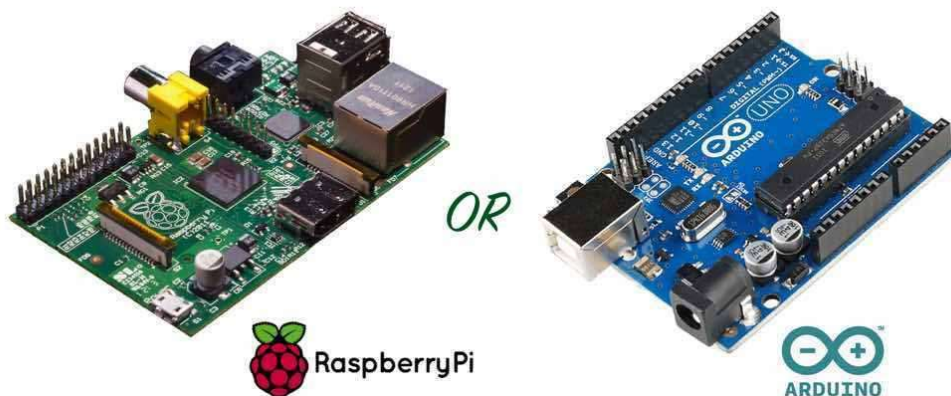
Figura 4.2-2. Prototip final boia.

Aquest extrem de la antena haurà de subjectar-se a alguna boia del litoral de manera que no repercuteixi amb la seva funció indicadora.

5 CONSTRUCCIÓ DEL NODE

5.1 Tecnologia a utilitzar

Arduino i Raspberry són les dues tecnologies més utilitzades per a la creació de projectes o prototips d'electrònica. Ambdues tecnologies utilitzen un sistema de software obert, és a dir, programes informàtics dels quals els seus codis són accessibles per a tothom y tothom els pot copiar, utilitzar i modificar. L'única diferència en el software és el seu funcionament: Arduino executa directament les tares per les quals les hem programat mentre que Raspberry requereix un sistema operatiu complet per poder utilitzar el programa d'una forma o d'una altre.



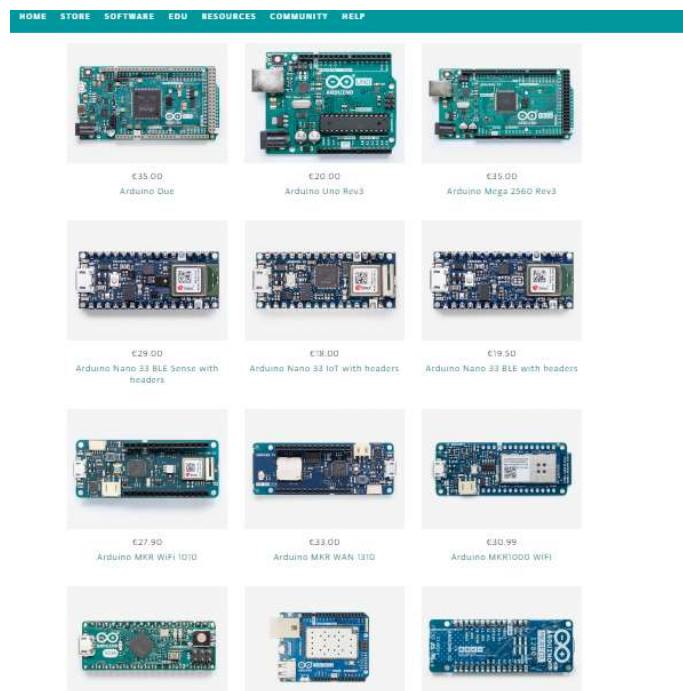
Which one is better for me?

FOSSBYTES

Taula 5.1-1. Les dues plataformes més conegudes per fer prototipat.

Una segona diferència entre aquestes dues tecnologies és el hardware. El hardware d'Arduino també és obert i tothom pot crear la seva pròpia versió de la placa mentre que la companyia Raspberry manté el control de disseny i fabricació de les seves plaques. La connectivitat també diferencien aquests dos conceptes: Raspberry conté a la mateixa placa la connexió ethernet i wifi mentre que Arduino fins ara necessita una expansió de la placa que permet aquestes dues connexions.

En els últims anys, però, la empresa Arduino a estès els seus models de placa de manera significativa, integrant opcions segons les necessitats dels usuaris i oferint una varietat de models a la carta.



Taula 5.1-2. Productes de la botiga Arduino.

La finalitat de la creació d'aquestes dues tecnologies també són diferents. Arduino va ser creat per facilitar l'accés i l'ús de l'electrònica i la programació, té una gran

connectivitat amb el món, gràcies a les seves entrades analògiques i digitals i el seu fàcil ús el que fa que Arduino sigui una tecnologia molt versàtil.

Raspberry va ser creat com a un ordinador en sí, funcional y el qual puguis portar allà on vulguis. Això fa que Raspberry tingui una gran potència de càlcul la qual permet una programació més complexa però fa falta un coneixement més ampli i no és tan versàtil com Arduino.

La nostra primera decisió de la plataforma a escollir fa referència a la corba d'aprenentatge. Arduino ens facilita molt més la part de programació mentre que Raspberry ens obliga a una inversió de temps addicional que s'allunya dels objectius del treball. La facilitat d'aquest aprenentatge i de la programació és gràcies a la gran comunitat Arduino que ens permet reutilitzar codis i llibreries útils per al nostre programa. Una altra característica que ens va decantar per utilitzar Arduino en el nostre prototip és el preu. Arduino s'ajusta més al nostre pressupost inicial i la facilitat d'accés als seus productes és molt més àmplia.

5.2 Models Arduino a utilitzar

Arduino dóna la possibilitat de treballar amb diferents mòduls depenent del que necessitis. Arduino UNO és el mòdul més standard, on tothom que s'inicia l'utilitza per la seva senzillesa i el seu baix cost econòmic. És la placa ideal per fer les proves dels sensors i és la que en un principi varem utilitzar per a les proves de funcionalitat dels sensors a incorporar.



Figura 5.2-1. Placa Arduino UNO

Com a característiques principals i que ens interessa tenir en compte, podem resaltar:

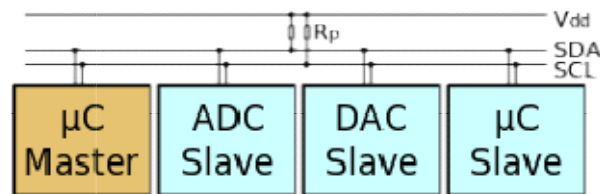
- El seu xip funciona a 5 volts.
- Disposa d'un connector per alimentar-lo amb bateria de 5-9 volts.
- Disposa de 5 canals analògics d'entrada i 13 digitals de entrada/sortida.
- Disposa de 1 canal sèrie.
- Disposa de 1 canal bus I2C

Una gran part de sensors de qualsevol tipus, transformen una magnitud a un valor de voltatge analògic. La disponibilitat de tenir un mínim d'entrades d'aquest tipus és bastant important.

A destacar que els canals digitals són d'entrada o sortida, i que segons com es programi la placa, podem utilitzar-les per llegir dades digitals (com ara un polsador) o per escriure i per tant, dirigides a actuadors (led, motor, relè...).

Que les plaques disposin de canal sèrie, permet als dissenyadors que les mesures que realitza la placa es puguin enviar a un ordinador per cable USB i així visualitzar les dades en pantalla.

També la funcionalitat de disposar de connexió I2C, permet expandir els circuits. El bus I2C és un protocol de comunicació sèrie entre petits mòduls o dispositius. Podríem dir que aquests tenen la capacitat de parlar entre ells i enviar dades. Normalment sempre hi ha un dispositiu que dirigeix la operació de transferència, anomenant Master (que podria ser la placa Arduino). També ha d'existir com a mínim un Esclau (p.ex. un sensor que envia dades d'una magnitud en 3 eixos, i per tant és una dada més complexa). Cada mòdul ha de disposar d'una única adreça en el mateix bus per a que no hi hagin conflictes.



Taula 5.2-2. Model de connexió de diversos dispositius mitjançant bus I2C.

Per adaptar-nos a les dimensions reduïdes del prototip que teníem pensat, Arduino ens proporciona també la placa Arduino NANO, on el funcionament i connexionat d'aquesta placa és exactament el mateix que l'Arduino UNO pero en un format reduït.

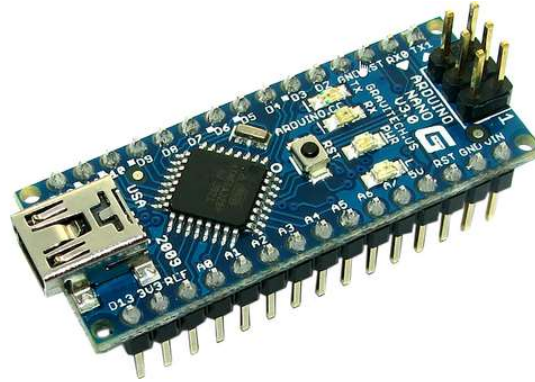


Figura 5.2-3. Placa Arduino Nano

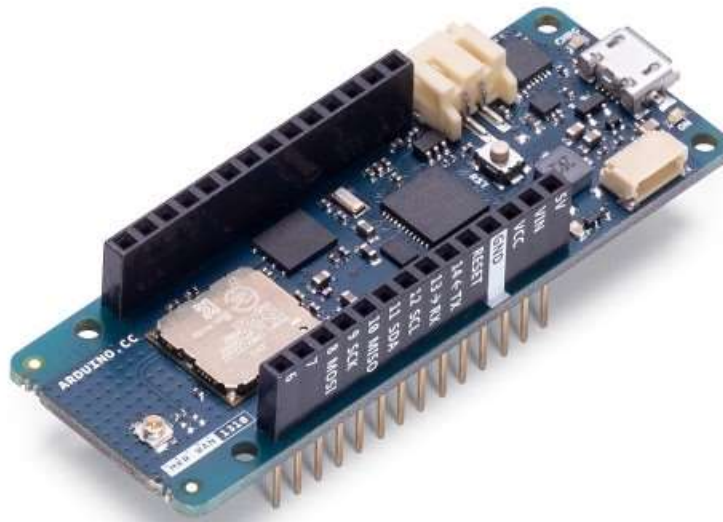
Per fer necessària la comunicació per radiofreqüència necessitem incorporar un mòdul adicional amb tecnologia Lora. El mercat ofereix diversos opcions de baix cost, com ara el: SeeedStudio RFM95 Ultra-Long Range Transceiver Module/Lora Module/Support 868M Frequency.



Figura 5.2-4. Mòdul Lora RFM95

Aquest mòdul faria de pont entre la placa Arduino i la xarxa externa. Tot i que valoràvem les seves reduïdes dimensions i el seu baix cost, la poca informació que teniem de com programar aquest mòdul ens ha fet descartar aquesta opció.

Cercant vam trobar l'opció d'Arduino ideal a les necessitats del projecte: l'Arduino MKR 1310. Aquest Arduino és una placa d'Arduino UNO o NANO amb l'extensió LoRa incorporada a la placa, és a dir, sense la necessitat d'incorporar l'extensió.



Taula 5.2-5. Placa Arduino MKR WAN 1310.

Algunes de les característiques d'aquesta placa són:

- El circuit opera a 3,3 volts, tot i que es pot alimentar externament a 5 volts
- Suporta també 1 canal sèrie i un canal I2C
- Suporta també 7 canals d'entrada analògics i 8 canals d'entrada/sortida digitals.
- Connexió preparada per antena i connector preparat per a bateria.
- Baix consum comparat amb amb les altres plaques comentades.

5.3 Els sensors

Per obtenir les dades necessàries del mar el prototip consta de tres sensors. Un sensor de conductivitat o salinitat, un sensor de temperatura i un sensor magnètic o brúixola.

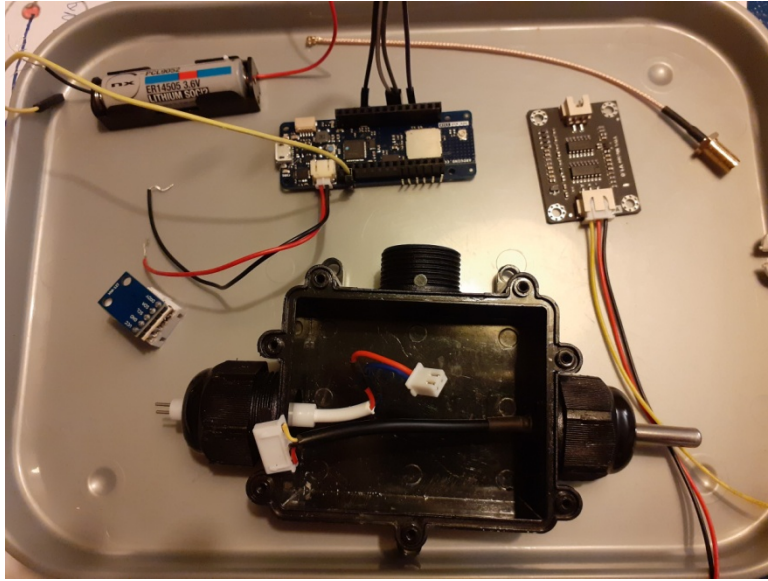


Figura 5.3-1. Sensors a la caixa de connexions.

5.3.1 Temperatura

L'únic requeriment important d'aquest sensor és que sigui hermètic per el seu contacte permanent al fons marí.

El sensor de temperatura que hem utilitzat és el sensor DS18B20. És un sensor de senyal digital que treballa entre 3V i 5V. El sensor presenta una forma encapsulada, escollida per nosaltres ja que és la forma adequada per al nostre prototip. Consta de tres connexions amb la placa Arduino: el GND, el DQ i el Vdd.

Pin Configurations

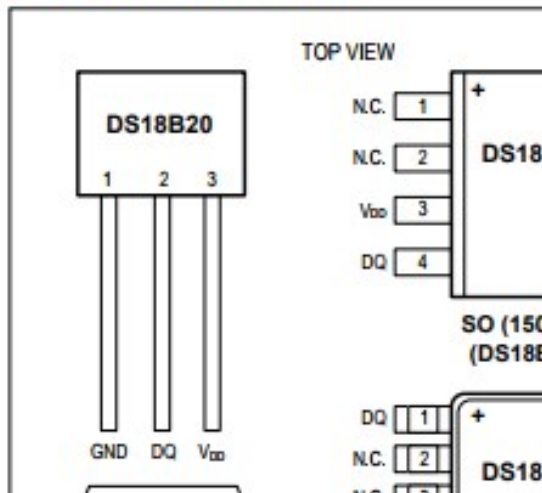


Figura 5.3-2. Integrat encapsulat. sensor.



Figura 5.3-3. Aspecte exterior del sensor.

5.3.2 Salinitat

Per obtenir la salinitat del mar hem utilitzat un sensor de conductivitat TDS de senyal analògic. El sensor funciona entre 3,3V i 5,5V. El sensor es divideix en dues parts.

La primera correspon a les dues barres de grafit que són els electrodes en contacte amb l'aigua per on circula un petit corrent elèctric. En funció de la sal diluïda, aquesta magnitud variarà.

La segona part seria un petit circuit sensor d'aquest corrent amb la particularitat que és un corrent altern. Amb un corrent altern es minimitza l'efecte electròlisi produït en un líquid entre dos electrodes a diferent potencial estàtic. Tot i que les lectures d'aquest sensor serien espaiades temporalment, l'efecte electròlisi permanent tendeix a oxidar un dels dos elèctrodes.

La relació de la conductivitat amb la salinitat és equivalent, si la conductivitat augmenta la salinitat és més alta i si la salinitat baixa la conductivitat també ho farà.



Figura 5.3-4. Sensor de Conductivitat

5.3.3 Brúixola

Per saber la direcció dels corrents marins hem de saber la orientació a la que es sotmetrà la boia en cada moment. La brúixola o magnetometre és un dispositiu que mesura els camps magnètics al seu voltant en els 3 eixos. Si no hi ha cap element magnètic al seu voltant, la mesura serà el camp magnètic terrestre en funció de com el tenim orientat. El sensor mesura els tres eixos x, y i z. Treballa entre 3V i 5V

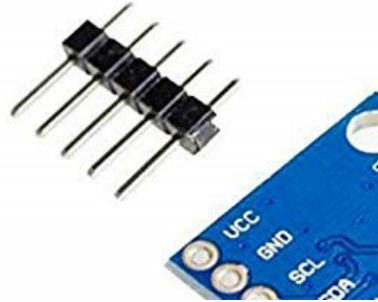


Figura 5.3-5. Magentomentre de tres eixos *tecnoiot 2pcs gy-273 qmc5883l*.

Tan sols mesurarem 2 eixos en un pla. En el programa i per trigonometria calcularem l'angle d'orientació resultant.

5.4 Estructura mecànica

A part de l'antena i el cable, l'estructura mecànica del prototip consta de dues parts. La part interior on es situa la caixa de connexions i la part exterior o carcassa orientadora.

5.4.1 Caixa de connexions IP68

La caixa de connexions on es situen els sensors i la placa arduino MKR 1310 és una caixa estanca de 3 vies amb certificació IP68.



Figura 5.4-1. Caixa de connexions impermeable.

La IP és el grau de protecció de la caixa davant agents exteriors. El primer dígit “6” ens indica el grau de protecció enfront la pols i el segon, el “8”, ens indica el grau de protecció davant líquids. Aquest últim dígit és l’essencial en el nostre projecte ja que la caixa queda submergida al mar. La IP68 ens permet enfonsar de manera segura les nostres connexions fins a un metre i mig de profunditat, el suficient per a que els sensors puguin desenvolupar el seu treball de manera correcta.

5.4.2 La Carcassa orientadora

La boia consta d’una caixa de connexions d’IP68, d’una estructura externa de metacrilat i d’un mosquetó per a la subjecció d’un pes.

La caixa de connexions es situaria, de manera fixa, dins d’una esfera de metacrilat. La esfera consta de dues aletes als extrems, una de les aletes funciona com a orientadora de la boia en funció de les corrents i així orientar la brúixola, que estarà ubicada a l’interior de la caixa, i que envii de manera correcte les dades. L’altre aleta funcionaria com a contrapès de l’altre per equilibrar el pes total

de la boia, aquesta aleta és més curta per deixar així a l'altre funcionar com orientadora de la boia.

El mosquetó es situaria a l'inferior i al centre de l'esfera amb la funció de subjectar un pes per enfonsar de manera total la boia i dona-li rigidesa en front de les corrents.

5.4.3 La antena

Per fer possible la connexió entre el Node i el gateway disposem d'una antena omnidireccional de 2.4Ghz, compatible amb una senyal WIFI de doble banda (2,4G / 5G) i amb una senyal inalàmbrica de llarga distància. Són antenes Wifi que les podem reutilitzar per la transmissió a 868MHz de la connexió LoRa.



Figura 5.4-2. Antena omnidireccional de 2 x 2.4GHz

6 CICLES DE CONSTRUCCIÓ I PROVA

Per tal de complir amb l'objectiu de construir el nostre Node, varem dividir el treball en diferents etapes, de manera que no pasavem a l'altre fins no tenir-ne assegurat el funcionament correcte de l'anterior per poder avançar d'una manera ordenada i amb la informació necessària en cada etapa.

6.1 Prova dels sensors amb Arduino UNO

La primera comprovació fa referència als sensors i el programa necessari. En la figura següent està representat el esquema del connexionat dels 3 sensors utilitzats.

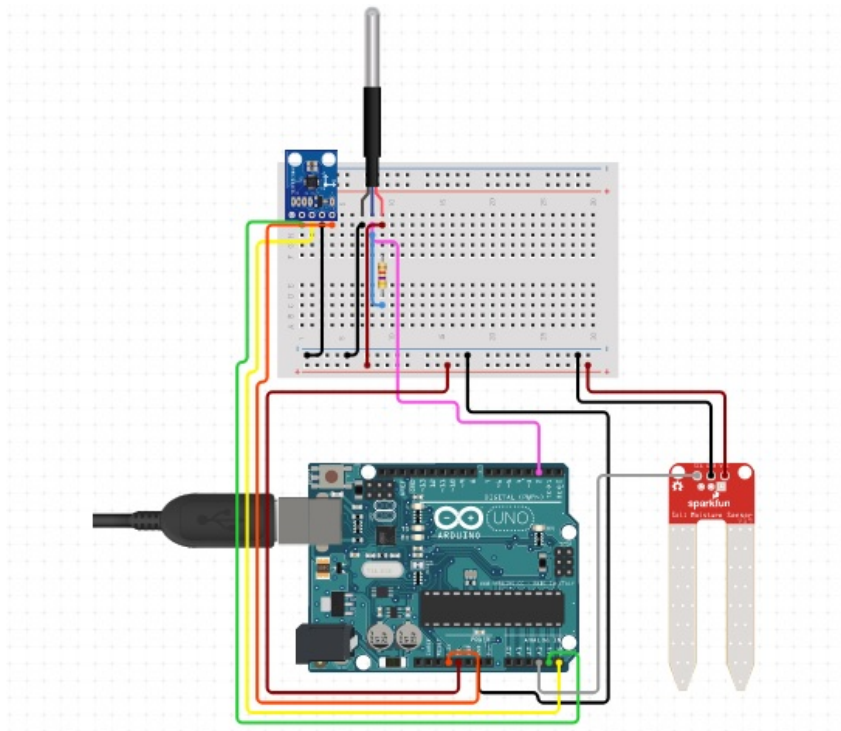


Figura 6.1-1. Esquema connexionat entre l'Arduino i els sensors, realitzat amb circuit.io

- El mòdul blau és la brúixola QMC5883L. A part de la alimentació +V (vermell) i GND (negre), es connecten els dos cables I2C: SDA (groc) i SCL (verd).
- El sensor allargat és el de temperatura DS18B20. A part de +V i GND, es connecta un cable de dades (violeta) amb la necessitat d'intercalar una resistència de 4k7 Ohms entre el cable de dades i +V (pull-up).
- El sensor en forma de forca representaria el nostre sensor de conductivitat TDS (el programa utilitzat no incorpora el nostre sensor en concret, tot i que té les mateixes característiques de connexionat), on a part de +V i GND, és necessari un cable (gris) per connectar el valor analògic a la entrada A0 de la placa.

Per tal de llegir els respectius sensors, hem aprofitat les llibreries fetes per la comunitat Arduino quan calia, i tant sols les hem adequat al nostre codi previ.

Per a la lectura del valor de salinitat, tan sols llegim el valor analògic de l'entrada A0

```
int valor=analogRead(A0);
```

Taula 6.1-2. Lectura del port analògic A0

Per la lectura de temperatura, el sensor es comunica amb el protocol 1Wire. Hem realitzat la connexió del sensor al port digital 10 i hem inserit la funció de lectura que proporciona aquesta llibreria.

```
#include <OneWire.h> //necessari pel sensor temperatura
OneWire ds(10); //a quin pin connectem el sensor

data[i] = ds.read(); //lectura de valors
```

Taula 6.1-3. Instruccions clau per lectura sensor de temperatura.

Per la lectura de la brúixola, hem aprofitat el recurs de Douglas Thain, on comparteix a tota la comunitat en el repositori github[xx] les llibreries que ens permeten llegir les orientacions en 3 eixos del mòdul mitjançant el bus I2C⁽⁴⁾.

Com que el valor d'orientació és un angle que pot estar canviant constantment, hem cregut convenient adaptar el valor a 8 quadrants: el Nord indicaria 0°, el Est 90°, el Sud 180° i el Oest 270° . També incorporat els intermitjos: NE, SE, SO, NO.

```
float angulo = atan2(y, x);
angulo = angulo * (180/3.1416);
angulo = angulo - declinacion;

if(angulo < 0) angulo = angulo + 360;

int orientacion=0;

if((angulo<22.5)||angulo>=337.5) orientacion=0;
if(angulo>=22.5&&(angulo<67.5)) orientacion=45;
if(angulo>=67.5&&(angulo<112.5)) orientacion=90;
if(angulo>=112.5&&(angulo<157.5)) orientacion=135;
if(angulo>=157.5&&(angulo<202.5)) orientacion=180;
if(angulo>=202.5&&(angulo<247.5)) orientacion=225;
if(angulo>=247.5&&(angulo<292.5)) orientacion=270;
if(angulo>=292.5&&(angulo<337.5)) orientacion=315;

return orientacion;
```

Taula 6.1-4. Quantificació del valor real de orientació a 8 valors enters.

L'algoritme del nostre programa inicial segueix la seqüència següent:

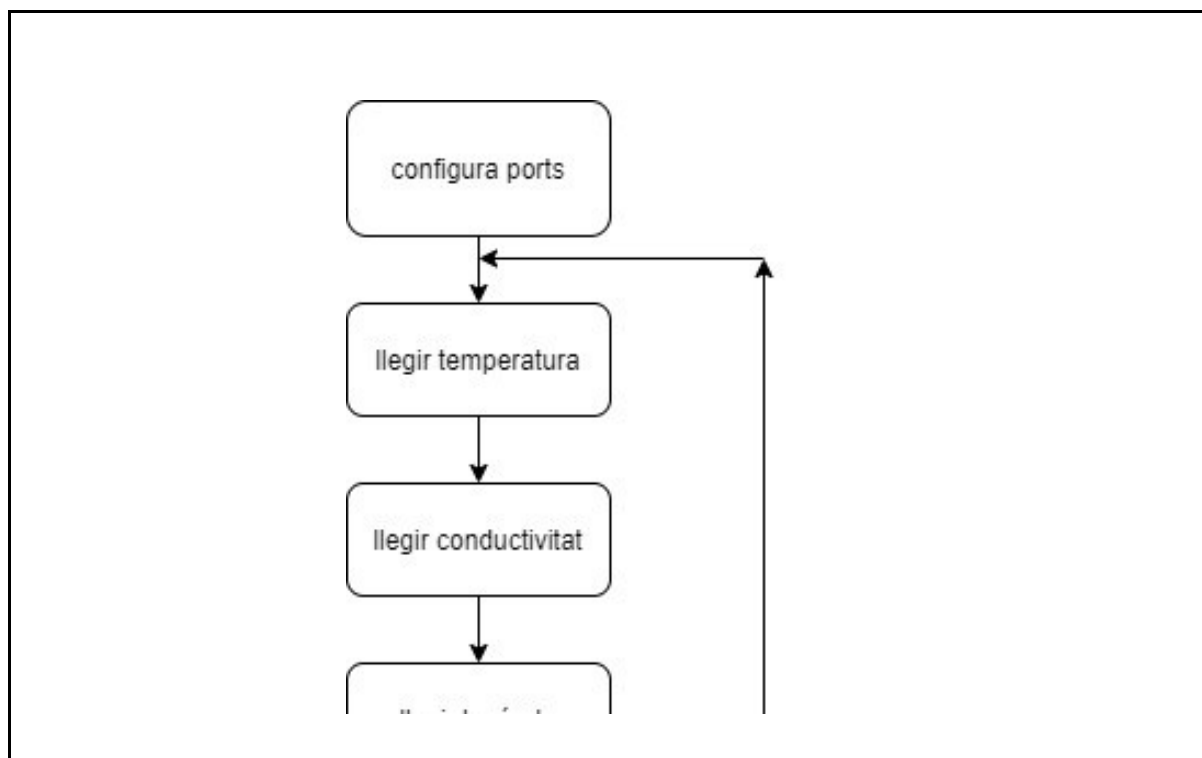


Diagrama 6.1-5. Algoritme inicial de comprovació dels sensors.

El següent pas és implementar-ho amb la placa MKR1310 per a comprovar l'enviament de dades. En aquest cas, utilitzem algun programa d'exemple que el mateix entorn de programació Arduino ofereix amb la inclusió de la llibreria MKRWAN, canviant algunes dades de configuració referents al servidor on s'ha de connectar i afegint el codi anterior de lectura de sensors.

També afegir les llibreries de baix consum, ja que volem optimitzar la despesa energètica de la bateria. Cada 10 minuts, mentre la placa no fa una lectura dels sensors i envia les dades per radiofreqüència, volem que la placa es posi en mode baix consum (deep sleep). Segons les especificacions, quan està en mode dormit,

el consum és de 100µA, enfront dels 100mA que hem mesurat quan està treballant en mode normal.

```
#include <ArduinoLowPower.h>
#include <MKRWAN.h>

LoRaModem modem;

#include "arduino_secrets.h"
String appEui = SECRET_APP_EUI;
String appKey = SECRET_APP_KEY;

.....
int connected = modem.joinOTAA(appEui, appKey); //connexió utilitzant les claus
.....
modem.beginPacket(); //preparació per enviar el paquet de dades
modem.print(payload); //enviament del paquet de dades
.....
LowPower.deepSleep(10*60*1000); //Posar a dormir durant 10min
```

Taula 6.1-6. Llibreries per treballar amb LoRa i mode baix consum.

L'algoritme final correspon a la seqüència que marca el diagrama següent:

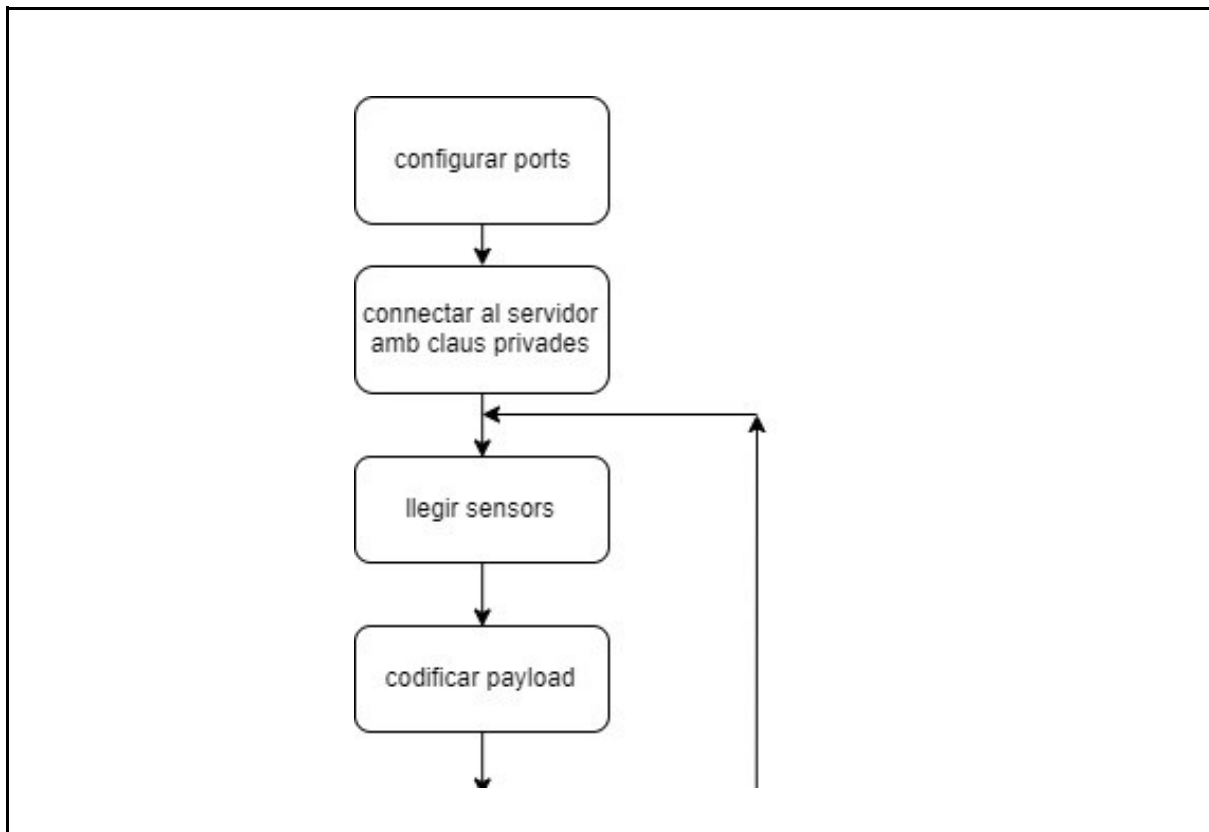


Diagrama 6.1-7. Algoritme general que executa la placa.

6.2 Prova de la xarxa TTN

Per a comprovar l'enviament de dades al servidor The Things Network, hem seguit les instruccions següents que la mateixa plataforma suggereix per fer-ho de manera ràpida i senzilla.⁽⁵⁾

El primer pas és obrir una aplicació que hem anomenat “insti”.

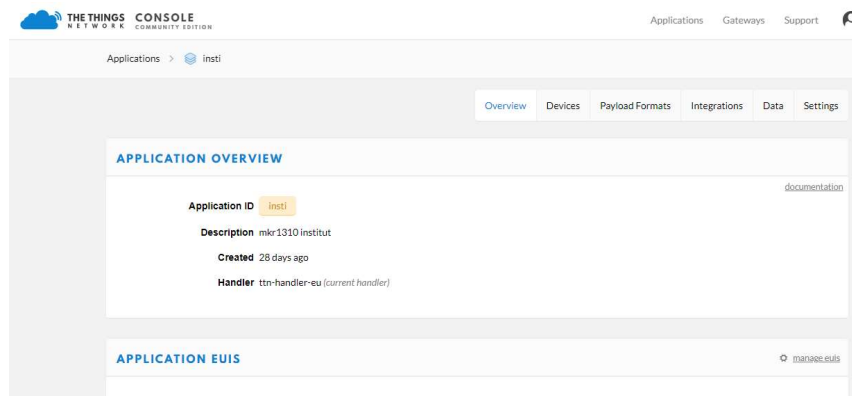


Figura 6.2-1. Configuració de l'aplicació d'obtenció de dades.

Una aplicació ha d'incorporar almenys 1 dispositiu registrat. Si el nostre sistema tingués més nodes o dispositius, l'aplicació ens permet concentrar-los tots i facilitar la gestió de les dades.

La segona part necessària doncs, és registrar el dispositiu. Com que el dispositiu forma part de l'aplicació, aquesta tindrà unes claus d'accés (Application EUI i App Key) que haurem de incorporar al programa de la nostra placa Arduino. D'aquesta manera es protegeix que tan sols els dispositius que coneixen aquestes claus podran enviar dades a l'aplicació.

Per a completar la gestió i codificació de les dades, hem d'escriure un petit "script" o mini funció relacionada amb el "PayLoad". El "PayLoad" és la manera com es desxifren o descodifiquen les dades que rep l'aplicació. Per a reduir la llargada de dades, aquestes s'han d'enviar en format hexadecimal i així optimitzar el pes d'aquestes dades. El que cal aleshores és afegir algunes instruccions que la mateixa plataforma ens ajuda a completar-ho amb èxit.

En el nostre cas enviem 2 dades que volem que tinguin un format enter (la conductivitat i un angle enter de la direcció respecte al nord) i una tercera en format real (la temperatura).

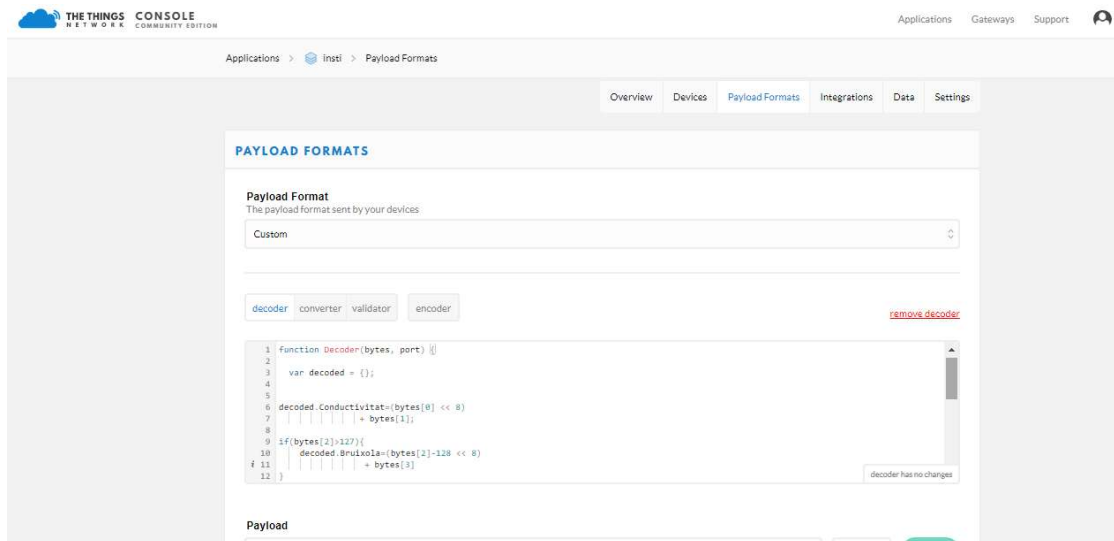


Figura 6.2-2. Finestra on es programa el Payload.

En aquest moment, i si el dispositiu està dins de la zona de cobertura d'algun Gateway, comencem a rebre les dades i codificades amb el format desitjat.

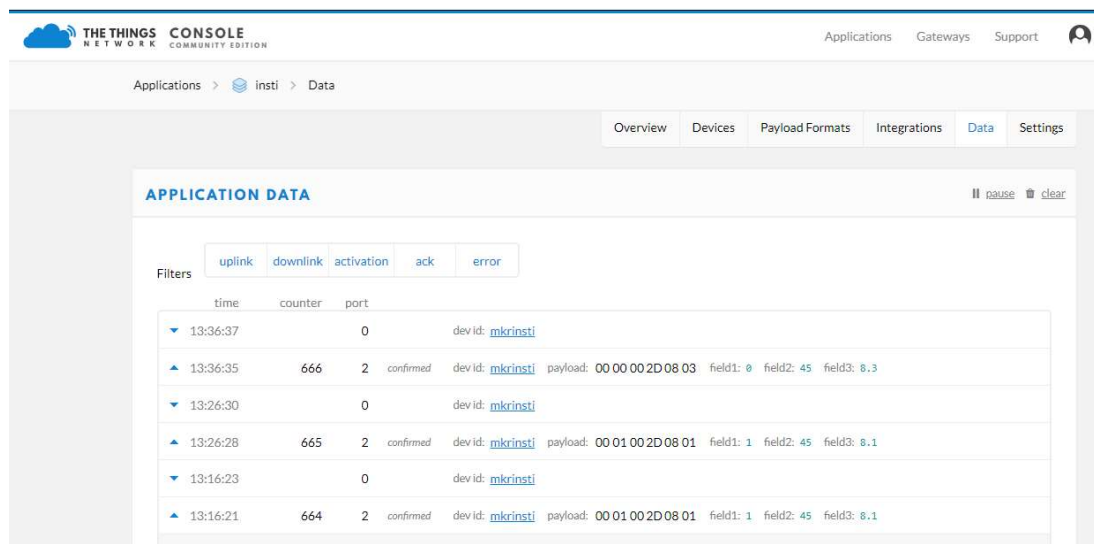


Figura 6.2-3. Resultats obtinguts en la finestra Dades.

Podem resaltar de la figura anterior com, cada 10 minuts, apareixen el comptador de missatges rebuts, l'identificador de dispositiu, el payload en format hexadecimal, i la descodificació d'aquest payload en tres camps: conductivitat, angle respecte nord i temperatura.

6.3 Exportació dades al cloud Thing Speak

Un següent pas és com emmagatzemar aquesta quantitat de dades que aniran acumulant-se dia rere dia. La particularitat del servidor TTN és que podem visualitzar aquestes dades quan obrim la sessió. Però un cop tanquem i tornem a obrir, ja no podem veure les dades anteriors i per tant, en cap moment que no tinguem la sessió oberta. El TTN no és un cloud o base de dades; la seva funció principal és interconnectar dispositius, xarxes i aplicacions.

Tot i així, la plataforma està pensada per a poder exportar les dades a qualsevol servidor destinat a emmagatzemar dades.

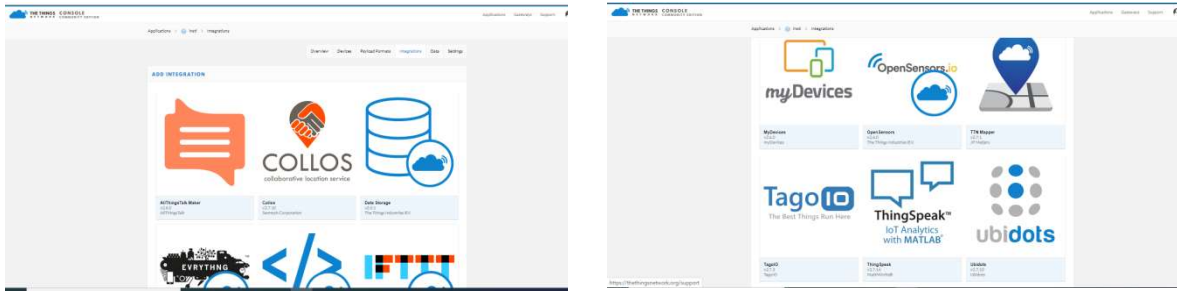


Figura 6.3-1. Algunes plataformes per exportar les dades amb diferents protocols.

Inicialment es volia exportar les dades al Drive del nostre compte corporatiu de l'institut Pedralbes amb la integració HTTP. Es requereix una nova aportació en programació del cantó del nostre compte Drive de l'institut amb les eines GAS (Google App Script), i afegir encara més codi per a representar les dades de manera gràfica en un full de càlcul.

Hem preferit fer l'exportació a la plataforma "The Thing Speak", que ens permet de manera gratuïta emmagatzemar dades si aquestes no superen algun límit en forma de tamany per dia.

A més, la mateixa plataforma ja crea visualitzacions temporals de les dades sense cap esforç de configurar-les o programar-les.

Un cop donats d'alta, podem crear els anomenats canals que podem ser públics o privats. Nosaltres hem creat un canal públic anomenat "insti".

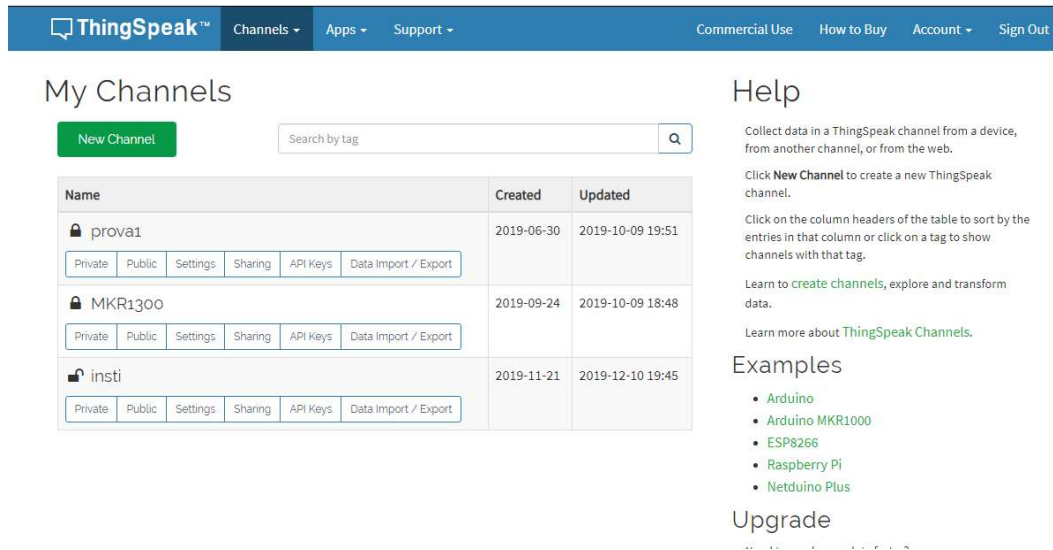


Figura 6.3-2. Web ThingSpeak, diferents canals.

I en aquest canal hem de configurar quins camps voldríem representar a partir d'una tramesa de dades. Aportem els noms dels 3 camps que volem emmagatzemar. I també podem configurar quin tipus de gràfic o visualització li volem donar.

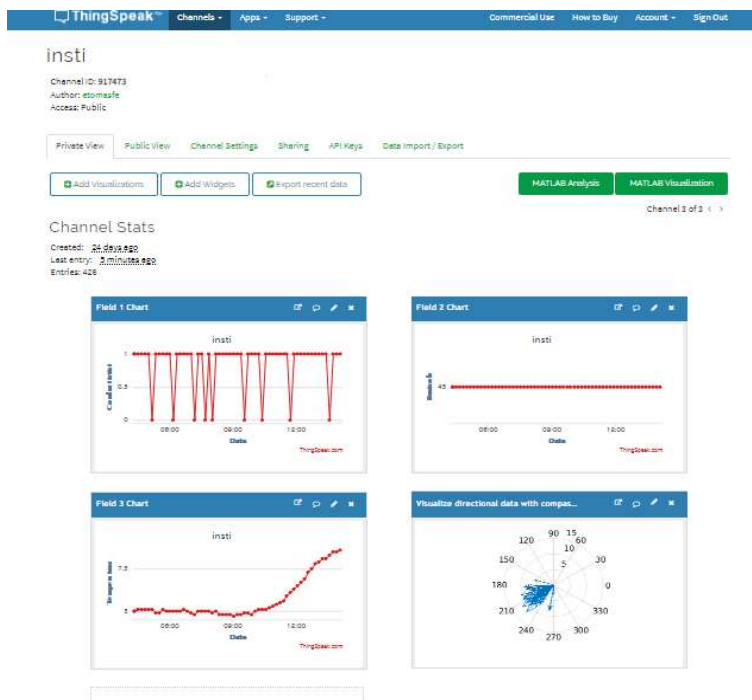


Figura 6.3-3. Visualització de les dades d'un canal a ThingSpeak.

En la Figura 6.3-3 podem observar d'esquerra-dreta i dalt-baix les dades de Conductivitat, Orientació i Temperatura. El quart gràfic està posat per donar visibilitat a les opcions de representar les dades d'una altra manera, com en el cas de la orientació, fent una mitjana dels últims n nombres.

Per acabar, hem de compartir també una clau d'accés entre aquesta plataforma i la de TTN i així assegurar-nos que tan sols rebem dades nostres.

6.4 Prova d'estanquitat

Per comprovar la fiabilitat de la caixa de connexions la varem enfonsar a una piscina a una profunditat d'un metre i mig durant una setmana. En obrir la caixa després de la prova, el paper que anteriorment havíem posat a dins ens va demostrar que no hi havia cap filtració d'aigua ni tan sols d'humitat.



Figura 6.4-1. Caixa de connexions després de la prova d'estanquitat.

6.5 Prova d'orientació

De manera empírica, hem provat diversos pesos per analitzar la reacció de la estructura i hem establert un pes mínim de 1 kg perquè sigui capaç de orientar-se en corrents de baixa força.



Figura 6.5-1. Proves de càrrega amb diferents quantitats de sorra en ampolla de plàstic.

A més hem comprovat l'eficàcia de l'estructura d'orientació i realment la boia sempre queda dirigida cap el sentit de la corrent. Les proves s'han realitzat en una zona del riu Llobregat, amb un corrent d'aigua força petit.

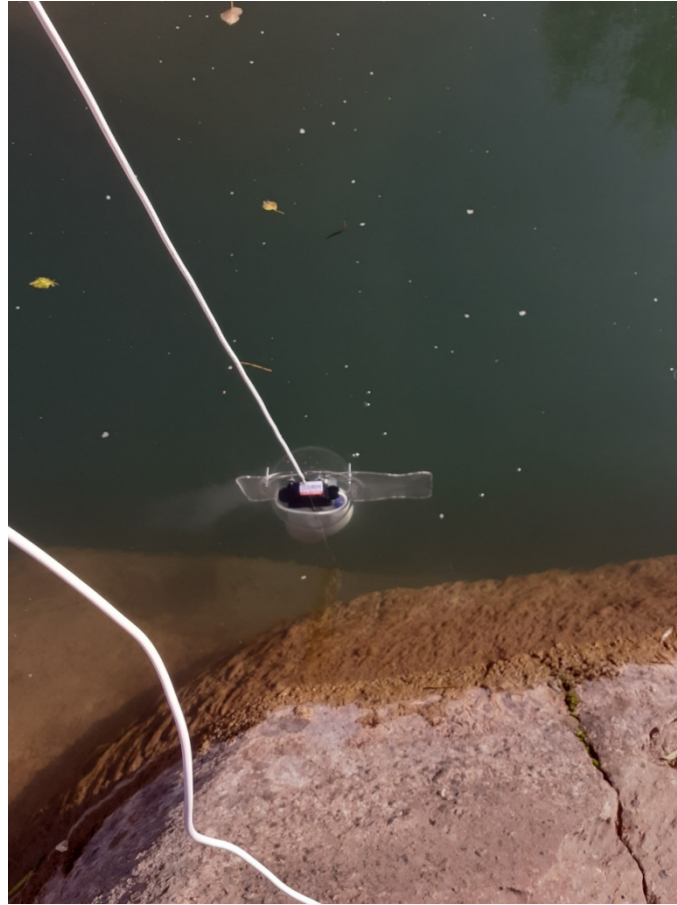


Figura 6.5-2. Estructura submergida i orientada en un corrent del riu Llobregat.

6.6 Prova final al ICM

Per obtenir una opinió d'autoritat i fer conèixer el nostre projecte vam fer una visita al ICM (Institut de les Ciències del Mar). Allà ens van atendre amablement l'enginyer Jaume Piera i el biòleg Carlos Roderó. Varem presentar el projecte, els objectius i el prototip. El nostre objectiu de la visita també era demanar poder deixar al mar durant un temps la boia que ells tinguessin disponible i així comprovar de manera empírica la funcionalitat del prototip. Malauradament no ens podien oferir de penjar el prototip, tot i que ens van oferir de fer una sortida amb

barca per als seus experiments i es comprometien a ficar-la durant unes hores al mar.

Amb aquesta petita prova que ells ens van oferir ja podem fer una visió de les dades recollides i si la carcassa orientadora compleix la seva funció de manera correcte, a més de les lectures de conductivitat i temperatura.

7 CONCLUSIONS I TREBALLS FUTURS

Vam iniciar el projecte amb l'objectiu de solucionar un problema present a les nostres costes, l'aparició masivas de les meduses. Vàrem cercar informació sobre elles i vam trobar que hi havia uns factors mesurables que podem utilitzar per trobar-ne una solució a la seva detecció.

Per començar la cerca a la solució del problema vam decidir fer un projecte IoT, un sistema que recollís totes les dades necessàries per detectar l'aparició d'aquest animal. Comparant aquest tipus de sistema que nosaltres plantegem amb les solucions ja vigents per la solució d'aquest problema ens vam adonar de que era innovador i viable a l'hora de portar-lo a terme.

Fins aquest punt del projecte havíem obtingut molta informació sobre les meduses, els diferents projectes IoT ja vigents⁽⁶⁾, de quins tipus de dades havíem de recollir i utilitzar una plataforma oberta com Arduino, ideal per a fer prototips ràpids. La incorporació al projecte de la plataforma *TTN* amb protocols *LoRa*, ens ha permès interconnectar la recollida d'informació malgrat no tinguem una cobertura WiFi a l'abast, i per tant els projectes *IoT* poden ser de major escala.

Salinitat, temperatura i direcció de les corrents eren les dades essencials que havíem de trobar en l'aigua del mar i per fer-ho necessitàvem una serie de sensors, Aquests sensors han d'estar submergits sota l'aigua i aquí va començar la part mecànica del projecte. En aquesta part el prototip ha patit una evolució per tal de millorar-lo i fer-lo més rendible. A mesura que anavem fent el prototip ens

venien noves idees a la ment que podiem implementar i això va fer que el prototip patís molts canvis fins a la seva forma final. Però gràcies això vam adonar-nos de molts petits errors que podien aparèixer si no feiem les modificacions adequades. En aquest sentit la part mecànica ha estat una de les més difícils d'implementar.

Gràcies a la visita al ICM i a tota la informació que vam recopilar hem aconseguit fer un bon sistema IoT alternatiu per a la detecció de meduses que pot servir d'ajuda, i en un futur si la comunitat augmenta aquesta xarxa podria ser molt més. En aquest sentit, hem decidit fer donació al ICM del nostre prototip per donar-lo a conèixer i potser així ampliar la xarxa amb la contribució d'altres persones.

Amb la visita al ICM vam poder observar altres projectes que allà estaven fent i vàrem concloure que el nostre no només en té com a sortida una única la detecció de meduses si no que la quantitat d'utilitats pot ser inmensa. Desde detecció d'altres animals fins l'estudi de la qualitat de l'aigua o fins i tot enfermetats que s'estan descongelant a causa del canvi climàtic i arriben a partir de l'aigua⁽⁷⁾.

Per acabar, el següent pas lògic del sistema tractaria sobre com processar de manera millor aquestes dades recollides, i trobar una correlació amb la detecció de meduses en els propers temps. Aquí ens trobaríem en una nova disciplina. És per això que aquest treball encara segueix viu i tot depèn de si algú agafa el testimoni.

Per a facilitar la comprensió dels programes, s'han posat comentaris (//) que clarifiquen quina funció realitzen certes instruccions.

Taula 9-1. Programa Test de sensors amb Arduino Uno

```

//#include <MPU6050.h> //proves amb mòdul inclinació

#include <Wire.h>

#include <MechaQMC5883.h> //necessari pel mòdul bruixola
MechaQMC5883 qmc; //necessari pel mòdul bruixola

#include <OneWire.h> //necessari pel sensor temperatura
OneWire ds(10); // on pin 10 (a 4.7K resistor is necessary)

void setup(void) {

  Serial.begin(9600); //inicialitzar canal sèrie per llegir a pantalla
  Wire.begin(); //inicialitzar canal i2c per bruixola
  qmc.init(); //inicialitzar mòdul bruixola
}

void loop(void) {

  delay(2000);
  float temperatura=get_temperature(); //crida a la rutina temperatura
  int conductivitat=get_conductivity(); //crida a la rutina conductivitat
  int norte=get_brujula(); //crida a la rutina bruixola

  //Serial.println("*****");
  //escriu els valors per pantalla
  Serial.println(temperatura);
  Serial.println(conductivitat);
  Serial.println(norte);
  Serial.println("*****");

}

//*****
//*****
int get_brujula(void){

  const float declinacion = 1.4; //valor necessari en el nostre emisferi

  int x,y,z;
  qmc.read(&x,&y,&z);

```

```

//Calcular ángulo el ángulo del eje X respecto al norte
float angulo = atan2(y, x);
angulo = angulo * (180/3.1416);
angulo = angulo - declinacion;

if(angulo < 0) angulo = angulo + 360;

int orientacion=0;

if((angulo<22.5)||angulo>=337.5) orientacion=0;
if((angulo>=22.5)&&(angulo<67.5)) orientacion=45;
if((angulo>=67.5)&&(angulo<112.5)) orientacion=90;
if((angulo>=112.5)&&(angulo<157.5)) orientacion=135;
if((angulo>=157.5)&&(angulo<202.5)) orientacion=180;
if((angulo>=202.5)&&(angulo<247.5)) orientacion=225;
if((angulo>=247.5)&&(angulo<292.5)) orientacion=270;
if((angulo>=292.5)&&(angulo<337.5)) orientacion=315;

return orientacion;
}
//*****
*****+
//*****
*****

int get_conductivity(void) {

int valor=analogRead(A0);
return valor;

}
//*****
*****

float get_temperature(void) {
byte i;
byte present = 0;
byte type_s;
byte data[12];
byte addr[8];
float celsius2, fahrenheit;

ds.reset_search();
delay(250);

if ( !ds.search(addr) ) {
// Serial.println("No more addresses.");
// Serial.println();
ds.reset_search();
delay(250);
// return;
}

// the first ROM byte indicates which chip

```

```

switch (addr[0]) {
  case 0x10:
    // Serial.println(" Chip = DS18S20"); // or old DS1820
    type_s = 1;
    break;
  case 0x28:
    // Serial.println(" Chip = DS18B20");
    type_s = 0;
    break;
  case 0x22:
    // Serial.println(" Chip = DS1822");
    type_s = 0;
    break;
  // default:
  // Serial.println("Device is not a DS18x20 family device.");
  // return;
}

ds.reset();
ds.select(addr);
ds.write(0x44, 1); // start conversion, with parasite power on at the end

delay(1000); // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad

for ( i = 0; i < 9; i++) { // we need 9 bytes
  data[i] = ds.read();
  // Serial.print(data[i], HEX);
  // Serial.print(" ");
}

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
  raw = raw << 3; // 9 bit resolution default
  if (data[7] == 0x10) {
    // "count remain" gives full 12 bit resolution
    raw = (raw & 0xFFF0) + 12 - data[6];
  }
} else {
  byte cfg = (data[4] & 0x60);
  // at lower res, the low bits are undefined, so let's zero them
  if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
  else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
  else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
  //// default is 12 bit resolution, 750 ms conversion time
}

```

```

}
celsius2 = (float)raw / 16.0;
fahrenheit = celsius2 * 1.8 + 32.0;

return celsius2;
}

```

Taula 9-2. Programa complet del Node amb Arduino MKR1310

```

#include <ArduinoLowPower.h>
#include <MKRWAN.h>

LoRaModem modem;

#include "arduino_secrets.h" //les claus per enviar les dades a TTN
String appEui = SECRET_APP_EUI;
String appKey = SECRET_APP_KEY;

#include <Wire.h> //llibreria I2C necessària per el mòdul bruixola

#include <MechaQMC5883.h> //necessari pel mòdul bruixola
MechaQMC5883 qmc; //necessari pel mòdul bruixola

#include <OneWire.h> //necessari pel sensor temperatura
OneWire ds(10); // on pin 10 (a 4.7K resistor is necessary)

void setup() {

  delay(5000);

  //posar sortides 6,7,8,9 a +V per alimentar els mòduls
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);

  delay(2000); //necessari per assegurar que els mòduls ja estan alimentats

  Serial.begin(115200);

  Wire.begin(); //inicialitzar canal i2c per bruixola
  qmc.init(); //inicialitzar mòdul bruixola

  // while (!Serial);
  if (!modem.begin(EU868)) {

```

```

Serial.println("Failed to start module");
while (1) {}
};
Serial.print("Your module version is: ");
Serial.println(modem.version());
Serial.print("Your device EUI is: ");
Serial.println(modem.deviceEUI());

appKey.trim();
appEui.trim();

bool go=false;
while(go==false){
int connected = modem.joinOTAA(appEui, appKey);
go=true;
if (!connected) {
Serial.println("Something went wrong; are you indoor? Move near a window and retry");
//while (1) {}
go=false;
}
}

// Set poll interval to 60 secs.
modem.minPollInterval(120);
// NOTE: independently by this setting the modem will
// not allow to send more than one message every 2 minutes,
// this is enforced by firmware and can not be changed.
}

void loop() {

int conductivitat=get_conductivity();
int norte=get_brujula();
float temperatura=get_temperature();

//Serial.println("*****");
//escriu els valors per pantalla
// Serial.println(temperatura);
// Serial.println(conductivitat);
// Serial.println(norte);
// Serial.println("*****");

String pepe="";
pepe+=(char)(highByte(conductivitat));
pepe+=(char)(lowByte(conductivitat));
pepe+=(char)(highByte(norte));
pepe+=(char)(lowByte(norte));
if(temperatura<0){
pepe+=(char)((char)(abs(floor(temperatura)))+127);
pepe+=(char)(abs(floor(temperatura)-temperatura)*10);
}
}

```

```

else {
    pepe+=(char)(floor(temperatura));
    pepe+=(char)((temperatura-floor(temperatura))*10);
}

int err;
modem.beginPacket(); //preparació per enviar el paquet de dades
modem.print(pepe); //enviament del paquet de dades

err = modem.endPacket(true);
if (err > 0) {
    Serial.println("Message sent correctly!");
} else {
    Serial.println("Error sending message :(");
    Serial.println("(you may send a limited amount of messages per minute, depending on the signal strength");
    Serial.println("it may vary from 1 message every couple of seconds to 1 message every minute)");
}

Serial.println();

digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, LOW);
digitalWrite(9, LOW);
LowPower.deepSleep(10*60*1000); //Posar a dormir durant 10min
digitalWrite(6, HIGH);
digitalWrite(7, HIGH);
digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
delay(2000);

Serial.begin(115200);

Wire.begin(); //inicialitzar canal i2c per bruixola
qmc.init(); //inicialitzar mòdul bruixola

delay(2000);
}

//*****
//*****
//***** RUTINA BRUIXOLA
//*****
//*****
//*****
int get_brujula(void){

const float declinacion = 1.4;

```

```

int x,y,z;
qmc.read(&x,&y,&z);

//Calcular ángulo el ángulo del eje X respecto al norte
float angulo = atan2(y, x);
angulo = angulo * (180/3.1416);
angulo = angulo - declinacion;

if(angulo < 0) angulo = angulo + 360;

int orientacion=0;

if((angulo<22.5)||((angulo>=337.5)) orientacion=0;
if((angulo>=22.5)&&(angulo<67.5)) orientacion=45;
if((angulo>=67.5)&&(angulo<112.5)) orientacion=90;
if((angulo>=112.5)&&(angulo<157.5)) orientacion=135;
if((angulo>=157.5)&&(angulo<202.5)) orientacion=180;
if((angulo>=202.5)&&(angulo<247.5)) orientacion=225;
if((angulo>=247.5)&&(angulo<292.5)) orientacion=270;
if((angulo>=292.5)&&(angulo<337.5)) orientacion=315;

return orientacion;
}
//*****
*****
//***** RUTINA CONDUCTIVITAT *****
*****
//*****
*****
int get_conductivity(void) {

int valor=analogRead(A0);
return valor;

}
//*****
*****
//***** RUTINA TEMPERATURA *****
*****
//*****
*****
float get_temperature(void) {
byte i;
byte present = 0;
byte type_s;
byte data[12];
byte addr[8];
float celsius2, fahrenheit;

ds.reset_search();
delay(250);

```

```

if ( !ds.search(addr) ) {
  // Serial.println("No more addresses.");
  // Serial.println();
  ds.reset_search();
  delay(250);
  // return;
}

// the first ROM byte indicates which chip
switch (addr[0]) {
  case 0x10:
    // Serial.println(" Chip = DS18S20"); // or old DS1820
    type_s = 1;
    break;
  case 0x28:
    // Serial.println(" Chip = DS18B20");
    type_s = 0;
    break;
  case 0x22:
    // Serial.println(" Chip = DS1822");
    type_s = 0;
    break;
  // default:
  // Serial.println("Device is not a DS18x20 family device.");
  // return;
}

ds.reset();
ds.select(addr);
ds.write(0x44, 1);    // start conversion, with parasite power on at the end

delay(1000); // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE);    // Read Scratchpad

for ( i = 0; i < 9; i++) {    // we need 9 bytes
  data[i] = ds.read();
  // Serial.print(data[i], HEX);
  // Serial.print(" ");
}

// Convert the data to actual temperature
// because the result is a 16 bit signed integer, it should
// be stored to an "int16_t" type, which is always 16 bits
// even when compiled on a 32 bit processor.
int16_t raw = (data[1] << 8) | data[0];
if (type_s) {
  raw = raw << 3; // 9 bit resolution default
  if (data[7] == 0x10) {

```

```

    // "count remain" gives full 12 bit resolution
    raw = (raw & 0xFFF0) + 12 - data[6];
}
} else {
    byte cfg = (data[4] & 0x60);
    // at lower res, the low bits are undefined, so let's zero them
    if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms
    else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms
    else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms
    //// default is 12 bit resolution, 750 ms conversion time
}
celsius2 = (float)raw / 16.0;
fahrenheit = celsius2 * 1.8 + 32.0;

return celsius2;
}

```

Taula 9-3. Descodificació PayLoad a la plataforma TTN

```

function Decoder(bytes, port) {

    var decoded = {};

    decoded.Conductivitat=(bytes[0] << 8)
        + bytes[1];

    if(bytes[2]>127){
        decoded.Bruixola=(bytes[2]-128 << 8)
            + bytes[3]
    }
    else {
        decoded.Bruixola=(bytes[2] << 8)
            + bytes[3]
    }
    if(bytes[4]>127){
        decoded.Temperatura=(bytes[4]-128+bytes[5]/10.0)*-1.0;
    }
    else {
        decoded.Temperatura=(bytes[4]+bytes[5]/10.0)*1.0;
    }

    //return decoded;
    return {
        field1:decoded.Conductivitat,
        field2:decoded.Bruixola,
        field3:decoded.Temperatura
    }
}

```



Característiques tècniques dels elements:

Taula 9-3. Sensor de temperatura

El sensor DS18B20 permite medir temperaturas de hasta 125°C de forma fácil y además está sellado en un **envoltorio estanco** que permite sumergirlo en un líquido o protegerlo de la intemperie. Dado que es un sensor digital, la señal leída no se degrada debido a la distancia del cableado. Puede funcionar en modo 1-Wire con una precisión de $\pm 0.5^{\circ}\text{C}$ con una resolución de 12 bits. También pueden utilizarse varios sensores sobre el mismo pin ya que internamente viene programado con un ID único de 64 bits para diferenciarlos. El rango de funcionamiento es de 3 a 5V por lo que se puede utilizar en prácticamente cualquier sistema de que use microcontroladores.

NOTA: Incluye una resistencia de 4.7K Ohm para poder utilizarlo.

Características del cable:

- Tubo de acero inoxidable de 6mm de diámetro por 30mm de largo
- Largo: 91cm
- Diámetro: 4mm
- Contiene un sensor de temperatura DS18B20

Conexión:

En función de la producción, los cables del sensor pueden variar pero seguirán según estas especificaciones:

- Si tienes 4 cables: Rojo es Vcc (positivo), Negro es GND (negativo) y Blanco es el cable de datos. La malla es GND.
- Si tienes 3 cables: Rojo es Vcc (positivo), Azul/Negro es GND (negativo) y Amarillo/Blanco es el cable de datos.

Características del sensor DS18B20:

- Rango de temperatura: -55 a 125°C
- Resolución: de 9 a 12 bits (configurable)
- Interfaz 1-Wire (Puede funcionar con un solo pin)
- Identificador interno único de 64 bits
- Múltiples sensores puede compartir el mismo pin
- Precisión: $\pm 0.5^{\circ}\text{C}$ (de -10°C a +85°C)
- Tiempo de captura inferior a 750ms
- Alimentación: 3.0V a 5.5V

Taula 9-4. Antena

Característica:

- La antena es compatible con la señal wifi de doble banda 2.4G / 5G, 6dbi, puede aumentar su señal inalámbrica incluso a una larga distancia de recepción.
- La antena omnidireccional wifi de alta ganancia se puede ajustar a 90 grados 135 grados 180 grados según sus diferentes necesidades.
- La interfaz adopta el agujero de rosca interno SMA, se puede combinar perfectamente con varios enrutadores, adaptador wifi, tarjeta wifi, módulo de video de alta definición.
- El bulkhead IPEX a RP-SMA tiene una amplia aplicación, se puede utilizar para enrutamiento inalámbrico / AP / tarjeta de red / módulo inalámbrico con interfaz IPEX (Nota: solo se utiliza para la interfaz MINI PCI-E, no se utiliza para M. 2 (interfaz NGFF)).

Presupuesto:

- Rango de frecuencia (MHz) 2300-2500
- Ganancia (dBi)> 5
- V.S.W.R. <1.3
- Polarización Vertical
- No Circularidad (dB) 0.5
- Impedancia (Ω) 50
- Lightning Protection Direct Ground
- Potencia máxima de entrada (W) 50
- Especificaciones mecánicas
- Tipo de conector SMA
- Radiating Element Material Copper
- Material Radome EVA
- Radome Color Gray
- Dimensión (mm) 195Unbend) 173 (curva)
- Longitud del cable de cola (mm) 100/120/200
- Tamaño del embalaje (mm) 295 × 110 × 60
- Wight (Kg) 0.09 / 12
- Condición de aplicación interior
- Temperatura de funcionamiento (\square) -40 ~ + 85
- Reposición de temperatura (\square) -55 ~ + 100

Cable de extensión U.FL / IPX a RP-SMA (F)

- Conecta la red WiFi interna a la antena Booster WiFi externa
- Se puede usar con antenas interiores y exteriores
- Compatible con 802.11a, 802.11b, 802.11g y 802.11n

Taula 9-5. Sensor de salinitat

El equipo de pruebas TDS de uso común es un bolígrafo TDS Aunque es barato y fácil de usar, no puede transmitir datos al sistema de control, hacer un seguimiento en línea a largo plazo y analizar la calidad del agua. Utilizando un instrumento especial, aunque los datos se pueden transmitir, la precisión es alta, pero el precio es muy caro. Con este fin, hemos introducido especialmente este sensor TDS compatible con Arduino, que se puede utilizar para medir el valor TDS del agua después de conectar al controlador Arduino.

Diseñado para arduino, este producto es plug and play y fácil de usar. La amplia fuente de voltaje de 3,3 ~ 5,5 V y la salida de señal analógica de 0 ~ 2,3 V hacen que este producto sea compatible con el sistema de control de 5 V, 3,3 V, que se puede utilizar fácilmente en el sistema de control ya hecho.

Aviso:

Debido a las restricciones de listado del producto, consulta el código fuente después de la compra. Las sondas TDS no se pueden utilizar en agua por encima de 55 °C.

La sonda TDS no debe colocarse demasiado cerca del borde del recipiente, ya que afectará a la pantalla.

La cabeza y el cable de la sonda TDS son impermeables y se pueden sumergir en agua, pero la interfaz de conexión y la placa de adaptador de señal no son impermeables. Por favor, presta atención al uso.

Características

1. Funcionamiento de amplio voltaje: 3,3 ~ 5,5 V.
2. Salida de señal analógica de 0 a 2,3 V, compatible con sistemas de control de 5 V, 3,3 V.

Especificaciones técnicas:

Placa del adaptador de señal:

Voltaje de entrada: 3,3 ~ 5,5 V.

Señal de salida: 0 ~ 2,3 V.

Corriente de funcionamiento: 3 ~ 6 mAh.

Rango de medición TDS: 0 ~ 1000 ppm

Precisión de medición TDS: ±10% F.S. (25 °C)

Interfaz del módulo: XH2.54-3P.

Interfaz de electrodo: XH2.54-2P.

Sonda TDS:

Número de sondas: 2.

Longitud total: 83 cm.

Interfaz de conexión: XH2.54-2P

Otros: sonda impermeable.

Taula 9-6. Brúixola / Magnetometre.

Chips: QMC5883L (HMC5883L Compatible)

Communication: IIC communication protocol

Measuring range: ± 1.3-8 gauss

Power supply :3-5V

Size: 1.8 x 1.3 cm

Net weight:17 g

Pel que fa al pressupost del prototip construït, esmentar que la totalitat del material ha estat comprat a la botiga on-line Amazon, exceptuant la esfera de metacrilat que ha estat comprada a la botiga Servei Estació. En aquest sentit, es podria trobar a altres botigues amb un preu més ajustat. Aquests preus són del 2019.

Taula 9-7. Pressupost

● Cable antena SMA	9,89€
● Bateria Liti recargable	3,43€
● Antena omnidireccional	4,49€
● Sensor TDS salinitat	18,29€
● Magnetometre triple eix	5,91€
● Caixa connexions IP68	10,86€
● Sensor temperatura	2,25€
● Arduino MKRWAN1310	33,0€
● Semiesfera metacrilat(x2)	20,2€
● TOTAL	108,32€

9 REFERÈNCIES

-
- (1):FUERIS, ENRIQUE (2009) “La medusa asesina llega al mediterráneo”. *El Mundo*.
<https://www.elmundo.es/elmundo/2009/04/28/baleares/1240950709.html>
- (2) : LÓPEZ, LAURA [Consultat 2019] “A qué se deben las plagas de medusas?” *El Mundo*.
<http://www.vivirmasymejor.elmundo.es/cuidate/a-que-se-deben-las-plagas-de-medusas>
- (3): VICIOSA, MARIO (2018) “Para qué sirve una medusa” . *El Independiente*.
<https://www.elindependiente.com/futuro/2018/07/14/para-que-sirve-una-medusa-aplicaciones-mares/>
- (4). Dthain [Consultat 2019]. “Driver for QMC5883L chip found in many GY-271 boards”. GitHub.<https://github.com/dthain/QMC5883L>
- (5): The Things Network organization [Consultat 2019]. “Device Registration”.
<https://www.thethingsnetwork.org/docs/devices/registration.html>
- (6): “The Squirrel Cafe Predicting winter weather conditions” [Consultat 2019]
<https://sensorius.jimdo.com/>
- (7): FOX-SKELLY, JASMINE (2017). “Los peligros de las enfermedades ocultas bajo el hielo durante miles de años que estan despertando”. BBC-Hearth. [Consultat 2019].
<https://www.bbc.com/mundo/vert-earth-39851987>

IoT Marítim, la detecció de meduses.