The background of the page is a light blue color with a complex, abstract pattern of darker blue lines and dots. These lines resemble a circuit board layout, with various paths, junctions, and circular nodes. The pattern is dense and fills the entire page, creating a technical and digital atmosphere.

Disseny,
construcció i
programació d'un
datalogger



*El veritable perill no és que els ordinadors
comencin a pensar com els homes, sinó que els homes
comencin a pensar com els ordinadors.*

Sydney Harris



Índex

1. Abstract:	4
2. Introducció:	7
3. Objectiu del treball:.....	8
4. Història d'Arduino	9
5. Què és un datalogger?	10
5.1 Els diferents tipus que existeixen:.....	10
5.2 Quant costa un datalogger convencional?.....	10
6. Què és i els diferents tipus de programació existents.	12
6.1 Software lliure:	13
7. Disseny del prototip.	14
7.1 Disseny de la caixa:.....	14
7.1.2 Grau de protecció IP:.....	15
7.2 Les diferents peces:.....	17
8. Procés de construcció i programació del prototip:.....	20
8.1 Com funciona el sensor del comptador?	20
8.2 Circuit base:.....	20
8.3 Com comptar les pulsacions?.....	21
8.4 L'emmagatzematge de les dades obtingudes:.....	23
8.4.1 Opcions d'emmagatzematge:	24
8.4.2 Els dos recursos per obtenir les dades temporals:.....	25
8.4.3 El mòdul Bluetooth:.....	26
8.4.4 La connexió dels mòduls amb el circuit base:	27
8.4.5 El programari per guardar els valors:	28
9. L'alimentació.	30
10. L'aplicació.	31
10.1 Bluetooth:.....	31
10.1.2 Context Històric:.....	31
10.2 Quines funcions s'han d'incloure dins l'aplicació?.....	32
10.3 Programari de l'aplicació i les seves pantalles.	32
10.3.1 Pantalla d'inici:	33



10.3.2 Pantalla d'elecció d'idioma:	33
10.3.3 Pantalla principal.....	34
11. Càlcul de costos i pressupost.	37
12. Estada a Lab-Ferrer:	38
13. Viabilitat:	39
14. Conclusions:	40
15. Aspectes a millorar:.....	41
16. Webgrafia:.....	42
Annex 1: Diccionari per entendre millor els diagrames de flux.	50
Annex 2: Codi del programa bàsic.....	51
Annex 3: Codi del programa final:.....	53
Annex 4: Fotografies:	57



1. Abstract:

Català:

Des de fa uns anys la conscienciació sobre el medi natural i la contaminació s'ha esdevingut una activitat fonamental per la lluita contra el canvi climàtic i malgrat la gent ser cada vegada més responsables, no n'hi ha prou. S'ha d'aconseguir reduir els residus, reciclar-los, reutilitzar-los i aprofitar-los més.

Una altra revolució d'aquest segle i segurament la més important, és la tecnològica, que ens permet grans avantatges com la de comunicar-nos amb persones d'arreu del món en tan sols uns instants, però també és una de les que més residus genera.

Les noves tecnologies no només han de servir per comunicar-nos entre nosaltres sinó també per millorar el consum de recursos tan preuats com és l'aigua. Així doncs, he fet un projecte que combina les noves tecnologies amb la millora del consum d'aigua.

Per aconseguir-ho, he creat un prototip en base Arduino per a la recollida de dades de reg, les quals després es podran pujar al núvol des d'on es podran analitzar gràficament.

El treball està organitzat en tres parts; la primera, saber què és el que es vol fer i com fer-ho; la segona, disseny del prototip i construcció; i per últim, conclusions i aspectes en que es pot millorar.

Castellano:

Desde hace unos años la concienciación sobre el medio natural y la contaminación se ha convertido en una actividad fundamental para la lucha contra el cambio climático y a pesar de que la gente es cada vez más responsable, no es suficiente. Se debe conseguir reducir los residuos, reciclarlos, reutilizarlos y aprovecharlos más.



Otra revolución de este siglo y seguramente la más importante, es la tecnológica, que nos permite grandes ventajas como la de comunicarse con personas de todo el mundo en tan sólo unos instantes, pero también es una de las que más residuos genera.

Las nuevas tecnologías no sólo deben servir para comunicarnos entre nosotros sino también para mejorar el consumo de recursos tan preciados como es el agua. Así pues, he hecho un proyecto el cual combina las nuevas tecnologías con mejorar el consumo de agua.

Para lograrlo, he creado un prototipo en base Arduino para la recolecta de datos de riego, las que luego se podrán subir a la nube desde donde se podrán analizar gráficamente.

El trabajo está organizado en tres partes; la primera, saber qué es lo que se quiere hacer y cómo hacerlo; la segunda, diseño del prototipo y construcción; y por último, conclusiones y aspectos a mejorar.

English:

In recent years, awareness of the natural environment and pollution has become a key activity in the fight against climate change, and despite people becoming increasingly responsible, it is not enough. We need to reduce waste, recycle it, reuse it and make the most of it.

Another revolution of this century and probably the most important, is the technological one, which allows us great advantages such as communicating with people from all over the world in just a few moments, but it is also one of those that generates the most waste.

New technologies must not only serve to communicate with each other but also to improve the consumption of resources as precious as water. So, I have done a project that combine the new techonolgies with the consumpiton of water.



To achieve this, what I have done is to create a prototype based on Arduino for the collection of irrigation data which can then be uploaded to the cloud from where they can be analyzed graphically.

The work is organized into three parts; the first one, knowing what you want to do and how to do it; the second one, prototype design and construction; and finally, conclusions and with which aspects it can be improved.



2. Introducció:

Des de fa vuit anys estic endinsat en el món de la robòtica i la programació tant d'aplicacions com webs i també de robots. Aquest món hem fascina, sobretot perquè m'encanta la idea de què un objecte inanimat es pugui moure, fins i tot pensar i fer servir la lògica depenent del que tu l'hi hagi programat prèviament. És com ensenyar i controlar el pensament d'un objecte que per si mateix, és incapaç de pensar.

En aquest treball, volia superar-me i fer un projecte diferent de tots els altres, volia que el que escollís fer, tingués una transcendència i funcionalitat dins de la vida d'algunes persones. A més a més des d'un principi desitjava fer un prototip que combinés la connexió Bluetooth entre una aplicació des d'un aparell mòbil i un Arduino. Altres aspectes que volia incloure dins el meu treball, era la recopilació de dades per part del microcontrolador així com el disseny en tres dimensions.

Per tal d'aconseguir tots aquests objectius, vaig decidir fer un microdatalogger, un recopilador de dades en el que poder accedir a llegir els valors mitjançant la transferència de dades per Bluetooth amb una aplicació. Tot i que aquests aparells ja existeixen, el que diferencia alguns d'aquests dataloggers amb el meu és que prioritzo molt la senzillesa en com ha de ser utilitzat el meu aparell, així com el baix preu.

Per saber més informació i acostar-me a persones que saben com es fan aquest tipus d'aparells, vaig decidir aprofitar l'estada a l'empresa que em va oferir l'equip del Laboratori Ferrer, experts en aquests prototips. Treballant amb ells hi vaig estar un total de tres setmanes on vaig poder aprendre molt.



3. Objectiu del treball:

L'objectiu del treball és dissenyar i construir un datalogger que pugui anar a l'exterior sense cap tipus de problema que enregistri quants litres d'aigua s'està utilitzant per regar un camp amb un comptador d'aigua, intentant fer-ho amb els mínims costos possibles ja que el cost d'un aparell d'aquestes característiques pot ser molt elevat.

L'aparell constarà d'un sensor extern al datalogger que serà l'encarregat de que per dins seu hi passi l'aigua. Seguidament les dades s'hauran de guardar cada un cert temps, per tant, serà necessari un dispositiu de rellotge i també un lector de targeta micro SD amb la seva targeta corresponent. Finalment, també hi haurem d'afegir un mòdul Bluetooth ja que el prototip disposarà d'una aplicació recollidora de dades que es connectarà mitjançant aquest tipus de connexió amb l'Arduino i permetrà la transferència i lectura de dades a més a més de poder pujar-les al núvol, d'aquesta manera, es transformarà en un document de text (*.txt*).

Aquest prototip haurà d'estar dins d'una caixa que haurà d'estar completament segellada a prova, tant d'entrada de pols com d'aigua o temperatura, ja que qualsevol d'aquests fenòmens podria escurçar la durabilitat de la placa Arduino i/o de la bateria.



4. Història d'Arduino

Massimo Banzi, David Cuartielles i David Mellis, amb l'objectiu d'apropar l'electrònica a les aules de secundària, van proposar proporcionar als instituts plaques electròniques, promovent d'aquesta manera la creació de projectes digitals per a persones amb pocs coneixements tècnics. Cal esmentar que per aquells temps, el cost d'una d'aquestes plaques de microcontroladors era elevat i, a més a més, no oferien el suport desitjat.

Més tard, el projecte inicial no va seguir endavant i van decidir crear-ne un de nou molt semblant. Aquest projecte, que esdevindria pocs anys després en un líder mundial DIY (Do It Yourselfes), el van anomenar Arduino.

A dia d'avui, Arduino, sense perdre els ideals amb el qual va ser creat el 2005, és un programa totalment lliure i que no cal pagar pels drets, a més a més, s'ha convertit en una empresa de desenvolupament tant de hardware com de software lliure. L'empresa, aquests últims anys també ha augmentat el seu estoc amb dispositius que permeten detectar i controlar objectes del món real.



5. Què és un datalogger?

Un datalogger (en anglès) o un registrador de dades, és un dispositiu electrònic que enregistra valors que poden estar en funció de variables. Per exemple, dades climatològiques, informàtiques, consum d'electricitat en una llar, etc.

Un enregistrator de dades guarda paràmetres de forma continuada (o cada un cert temps o esdeveniment) per mitjà de sensors propis o connectats externament. Un aparell com aquest, pot estar connectat amb un equip informàtic que faci d'intermediari entre el datalogger i la persona que vol llegir els valors, tot i així, hi ha molts dispositius que ja porten incorporat una pantalla (LCD) on hi surten els paràmetres guardats.

Pel que fa a la font d'alimentació, normalment porten una pila o bateria ja que en aquests tipus d'aparells és de gran interès que el dispositiu estigui en un entorn on no hi hagi accés a l'electricitat, com ara al mig d'una muntanya, camp, etc.

5.1 Els diferents tipus que existeixen:

N'existeixen dos tipus:

- Els que actuen com a enregistadors i emmagatzemadors de dades, és a dir, que l'aparell utilitza sensors propis.
- Els que tan sols actuen com a emmagatzemadors de dades, per tant, no tenen sensors propis sinó que cal connectar-hi sensors externs. Dins d'aquest grup, s'hi pot incloure els dataloggers meteorològics digitals.

5.2 Quant costa un datalogger convencional?

El cost d'un datalogger ha anat disminuint amb el temps ja que la tecnologia va millorant. El preu d'un de convencional, pot oscil·lar des dels vint fins a alguns milers d'euros. Els més barats acostumen a ser els que es solen posar dins la llar o en alguns museus per controlar l'humitat o temperatura d'una sala. Tanmateix, el tipus de datalogger que volem crear en aquest projecte no es



troba per unes desenes d'euros sinó que pot arribar perfectament fins als dos-cents euros.



6. Què és i els diferents tipus de programació existents.

Un microcontrolador, igual que un robot, necessita un programa per poder funcionar, tant si la seva funció és poca com molt important. En un programa, el que fas és dir-li a l'aparell que compleixi unes determinades ordres tenint en compte totes les possibilitats que poden succeir. Aquestes diferents possibilitats s'anomenen variables i en el nostre cas poden ser les condicions climatològiques, el pas de l'aigua dins l'interior del sensor i la velocitat, quan la persona vol transferir les dades, etc. Existeixen molts llenguatges de programació, com ara el codi binari, que es utilitza en els microprocessadors dels dispositius informàtics per detectar l'absència o presència de senyal o de bits¹, com també se'ls coneix. Seguidament, un dels llenguatges de programació més utilitzats és el HTML, utilitzat per crear pàgines web i aplicacions. Aquest, va acompanyat del CSS (per definir els estils) i Java Script. Després també existeixen llenguatges de programació per blocs com són l'Scratch, EV3, o Appinventor (l'utilitzarem per crear l'aplicació). Finalment ens trobem amb l'Arduino, un llenguatge de programació lliure (*Software lliure*) que és el que utilitzarem per programar el nostre datalogger.

```

# styles.css  Miret.html  JS main.js
Miret > Miret.html > html > body > div.container > section#samarretes > div.news-cards > div > div.slide-wrap > section.slider
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Miret</title>
7   <script src="http://code.jquery.com/jquery-latest.js"></script>
8   <script src="miret.js"></script>
9   <!--SLIDER-->
10  <link rel="stylesheet" href="https://cdn.jsdelivr.net/bxslider/4.2.12/jquery.bxslider.css">
11  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
12  <script src="https://cdn.jsdelivr.net/bxslider/4.2.12/jquery.bxslider.min.js"></script>
13  <script src="https://cdnjs.cloudflare.com/ajax/libs/bxslider/4.2.15/jquery.bxslider.min.js"></script>
14  <link href="https://cdnjs.cloudflare.com/ajax/libs/bxslider/4.2.15/jquery.bxslider.min.css" rel="stylesheet" /
15  <!--FONT AWESOME-->
16  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css">
17  <!--FONS OSWALD-->
18  <link href="https://fonts.googleapis.com/css2?family=Oswald:wght@200;300;400&display=swap" rel="stylesheet">
19  <!--CUSTOM CSS-->
20  <link rel="stylesheet" href="styles.css">
21  <link rel="shortcut icon" href="https://lh3.googleusercontent.com/proxy/zCR47rBbXBLxRp63D2ms5TV9tEoMuzDe1j
22 </head>
23 <body>
24
25 <!--SHOWCASE-->
26 <header class="showcase">
27   <h1> Cal Miret </h1>
28 </header>
29 <div class="barra"></div>
30 <hr>

```

Figura 1: Programa en HTML d'una pàgina web

¹ **Bit**: En informàtica i altres disciplines, unitat mínima d'informació, que pot tenir només dos valors (zero o u).



6.1 Software lliure:

S'anomena Programari Lliure (*Software Lliure*) quan els usuaris disposen de llibertat per a executar, copiar, distribuir, canviar i millorar el programari. És el que s'anomena com "licències creative commons" que consten de quatre tipus de llibertat per als usuaris del programari:

1. Llibertat absoluta alhora de modificar el programa.
2. La possibilitat de poder modificar el programa en funció de les necessitats que un tingui d'usuari.
3. Qualsevol programa lliure s'ha de poder copiar i distribuir sense cap tipus de restricció.
4. Qualsevol usuari amb coneixements ha de poder modificar el programa i distribuir-lo amb les millores que hi hagi afegit, sempre amb l'intenció de que la resta d'usuaris se'n puguin beneficiar.



7. Disseny del prototip.

7.1 Disseny de la caixa:

A la intempèrie les condicions climatològiques són molt variades i com a conseqüència poden influenciar en gran mesura la durabilitat tant de la placa Arduino com de la font d'alimentació. Això pot comportar la reducció a més de la meitat de la seva vida útil ja que la temperatura òptima per a una bateria de liti es d'uns 25 °C. Tenint en compte que la Segarra té un clima de temperatures extremes tant a l'hivern com a l'estiu, és necessari protegir l'aparell amb una caixa que eviti en certa mesura el canvi de temperatura i evidentment que protegeixi el datalogger de la pluja, calamarsa, etc.

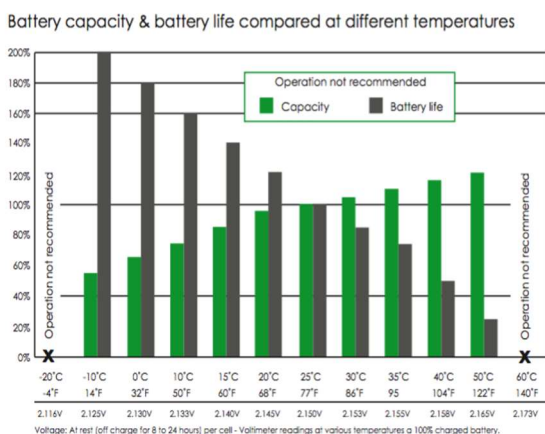


Figura 2: Gràfica de la durabilitat de la bateria i la seva capacitat comparada a diferents temperatures

La caixa la imprimirem amb una impressora en tres dimensions que ens agilitzarà molt la construcció, tot i que a l'hora de dissenyar-la, s'han de tenir alguns aspectes en compte. Primerament, l'error que pot fer la impressora quan imprimeix (normalment és d'entre 1 i 2 mm). Seguidament, dissenyar el model d'una forma que optimitzi material a l'hora d'imprimir perquè si programes que imprimeixi per exemple una esfera buida, com que no pot imprimir "sobre l'aire" el que farà serà construir un suport el qual estarà fet de plàstic ocuparà tot el dins de l'esfera, consumint així una quantitat molt més gran de plàstic que s'acabarà depreciant.



7.1.2 Grau de protecció IP:

Per confirmar en quins tipus de medis pot ser utilitzat el nostre prototip, qualsevol aparell electrònic que hagi d'anar tant a l'exterior com a l'interior, s'examina i es determina un grau de protecció IP (de l'anglès *Ingress Protection*). I s'indica de la següent forma: IP56, IP65, IP67...

IP [**x**][**y**]

Segons quins siguin els números que van seguits de les lletres IP, l'objecte serà adient per uns medis o per uns altres. Seguidament es mostra la taula de la qual els números depenen:

Primera xifra (x)		
Grau	Tamany de l'objecte que ha d'entrar	Efectiu contra:
0	---	Sense protecció
1	<50 mm	Per la prova s'ha d'utilitzar una esfera de 50mm de diàmetre que no ha d'arribar a entrar per complet.
2	<12.5 mm	Per la prova s'ha d'utilitzar una esfera de 12,5 mm de diàmetre que no ha d'arribar ha entrar per complet.
3	<2.5 mm	Per la prova s'ha d'utilitzar una esfera de 2,5 mm de diàmetre que no pot entrar gens ni mica.
4	<1 mm	Per la prova s'ha d'utilitzar una esfera de 1mm de diàmetre que no pot entrar gens ni mica.
5	Protecció contra la pols	L'entrada de pols és inevitable però no pot entrar amb una quantitat tal que interfereixi amb el correcte funcionament de l'aparell.
6	Protecció completa contra la pols	La pols no entra sota cap circumstància.



Segona xifra (y)			
Grau	Efectiu contra:	Descripció:	Detalls:
0	Res	--	--
1	Gotes d'aigua	Gotes d'aigua caigudes verticalment durant 10 minuts	Simula pluja de 1mm
2	Gotes d'aigua amb angle de 15°	Gotes d'aigua caiguda vertical a 15° durant 2,5 minuts	Simula pluja de 3mm
3	Aigua amb esprai	Aigua nebulosa caiguda vertical a 60° durant 10 minuts	Aigua a pressió de 50-150 KPa
4	Aigua esquitxada	Aigua esquitxada caiguda en qualsevol direcció durant 10 minuts	Aigua a pressió de 50-150 KPa
5	Aigua a doll	Aigua projectada en qualsevol direcció durant 3 minuts	Aigua a pressió de 30 KPa a distància de 3m
6	Aigua a doll fort	Aigua projectada en qualsevol direcció durant 3 minuts	Aigua a pressió de 100 KPa a distància de 3m
6K	Aigua a doll molt fort	Aigua projectada en qualsevol direcció durant 3 minuts	Aigua a pressió de 1000 KPa a distància de 3m



7	Immersion en aigua fins a 1m	Immersion fins 1m	Durada de 30 minuts
8	Immersion en aigua a 1m o més	Immersion a partir d'1m a especificar pel fabricant	Durada especificada pel fabricant

Aquestes proves s'han de fer en laboratoris especialitzats per tal que t'ho homologuin. En el meu cas, no dispo de suficients recursos com per a validar completament la caixa, tanmateix, consultant les especificacions de la taula anterior es poden fer una sèrie de proves fàcilment des de casa per fer-nos una idea del seu valor.

El que s'ha fet és col·locar la caixa sobre d'arena i posteriorment agitar-la fent l'efecte de pols. Vaig poder observar que no entrava arena dins seu.

Seguidament, per la segona prova, es va agafar un vaporitzador d'aigua i es va ruixar la caixa. Al veure que no entrava, es va augmentar la quantitat d'aquest líquid i la pressió fins el punt que l'aigua va entrar dins de la caixa.

Amb aquestes dues proves es pot concloure que l'IP resultant, si es fes en una empresa reglamentada, seria d'un IP54 o IP65. Tenint en compte que per a aquests productes s'acostuma a demanar que tinguin un IP54 aquest producte compliria el requisit per posar-se a la venda.

7.2 Les diferents peces:

Amb el programa FreeCAD, crearem el disseny en tres dimensions de la caixa que ha de protegir la part elèctrica del nostre datalogger. Constarà de 4 peces separades per tal d'optimitzar material.

1a peça (la caixa):

És la part més important de l'estructura, consta d'un prisma de base 15 cm per 15 cm i l'alçada màxima (en el punt de la carena) és de 17,25 cm. La part



superior serà triangular amb dents per acoblar-hi la teulada. També incorporarà l'obertura i el marc de la porta.

2a peça (la teulada):

La teulada és la part que estarà més exposada tant a la pluja com a la calamarsa, i per tant, per tal que l'impacte d'aquests fenòmens meteorològics sigui el mínim possible, és necessari fer que l'angle entre els dos vessants de la teulada sigui força petit. En aquest cas el faré de 90 graus. Per a què s'acobli bé a la caixa, hi inclouré unes entrades que coincidiran amb les dents fetes a la part superior de la caixa. Finalment, amb l'objectiu de fer-la més atractiva, arrodoniré les cantonades de la teulada.

3a peça (la biga):

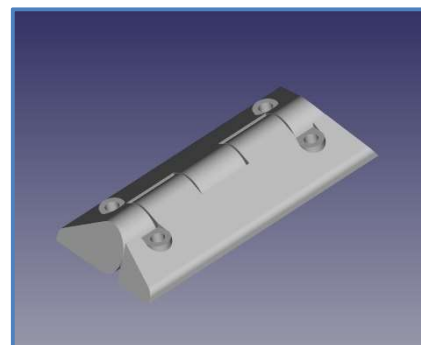
Com ja s'ha dit, en la part de la caixa s'hi farà el marc de la porta que permetrà segellar bé i fer-la hermètica. Tanmateix, aquest marc no es pot fer tot a la vegada ja que a l'hora d'imprimir el consum de plàstic seria molt més elevat perquè hauria de crear un suport per imprimir sobre l'espai. Així doncs, cal fer la part superior d'aquest marc a part.

4a peça (la porta):

La porta ha d'anar encaixada dins l'obertura feta a una de les parets de la caixa. Aquesta porta haurà d'incorporar una maneta que faciliti obrir i tancar la porta. A més a més, al contorn de la part de dins de la peça hi haurà una canaleta on després s'hi enganxarà una junta tòrica², que permetrà que la caixa romangui estanca.

5a peça (la frontissa):

L'última peça és l'encarregada d'ajuntar la porta amb la caixa i que permeti el moviment entre aquestes dues. D'aquesta manera es podrà obrir i tancar la porta sense cap tipus de problema.



² **Junta tòrica:** És una junta d'estanquitat amb forma toroidal, habitualment feta de cautxú, la funció de la qual és assegurar l'estanquitat de fluids.



Un altre aspecte a tenir en compte, encara que no sembli molt important és el color amb el qual s'imprimirà el model. Tenint en compte que el prototip haurà d'estar exposat a la calor de l'estiu, és necessari un color que no absorbeixi la calor, per tant, he optat pel blanc.

Figura 3: Model en tres dimensions de la frontissa.

Un cop totes les peces s'hagin imprès, és necessari acoblar-les. Pel que fa la teulada, la biga i la caixa, les ajuntarem entre elles enganxant-les amb cola especial per a plàstic, i la porta s'ajuntarà amb la caixa a través d'una frontissa també impresa amb la impressora en tres dimensions que permetrà el moviment de la porta.

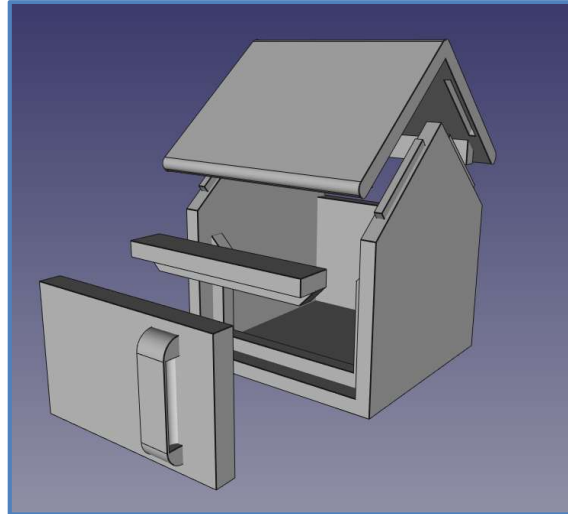


Figura 4: Disseny en tres dimensions de la caseta

Un cop imprès podem observar que la porta tanca perfectament sense ajuda de cap tanca ja que la pressió amb el seu marc ja és suficient.



8. Procés de construcció i programació del prototip:

8.1 Com funciona el sensor del comptador?

Com ja he explicat anteriorment, el prototip disposa d'un sensor extern que és l'encarregat de detectar la quantitat d'aigua que hi passa per dins seu. Aquest comptador funciona per pulsacions. Dins el sensor hi ha el que s'anomena reed-switch que és un interruptor elèctric que s'activa amb un camp magnètic. Dit amb unes altres paraules, quan a prop seu hi ha un imant, el circuit es tanca i deixa passar l'electricitat, en cas contrari el circuit es manté obert.

El sensor el que fa és girar una fletxeta amb un imant a sobre, i damunt la trajectòria d'aquesta fletxa hi ha el reed-switch que està estàtic. Sabent que cada volta que fa la fletxa és un litre, i que per a cada volta passarà per sota el reed-switch una vegada i per tant, es tancarà el circuit, el que farà el programa serà comptar les vegades que es deixa passar l'electricitat, transformant així, les pulsacions a litres.

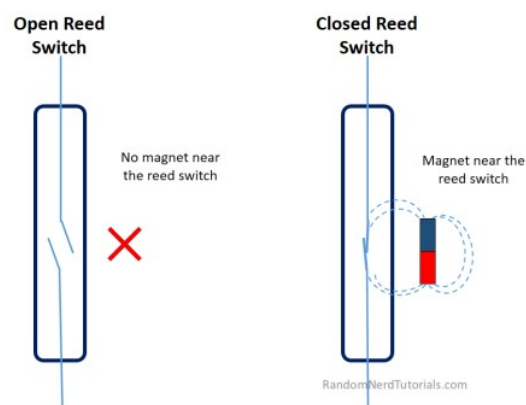


Figura 5: Esquema del funcionament d'un reed-switch

8.2 Circuit base:

En un inici, a l'hora de pensar com ha de ser el prototip, és important tan sols pensar en la funció principal que ha de fer l'aparell. Quan aquest aspecte s'hagi complert, és quan s'hi poden afegir altres funcions complementàries que agilitzaran el procés de recollida de dades.

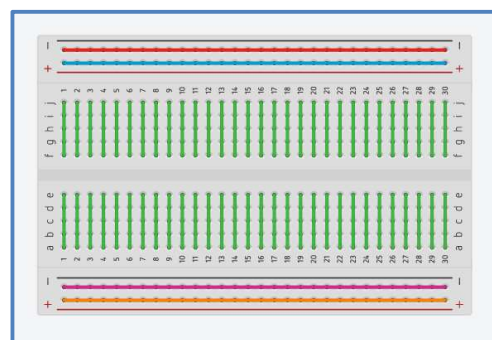


Figura 6: Funcionament de les connexions d'una placa protoboard

Així doncs, el primer pas és detectar les pulsacions que es realitzen amb un polsador (substitueix al sensor real) o, en altres paraules, comptar quantes vegades s'obre i es tanca el circuit. Amb aquest plantejament, es va muntar un



circuit bàsic tal com es mostra a la figura 7, que contenia només un polsador, una resistència de 10.000 ohms i una placa Arduino, tot connectat entre sí amb l'ajuda d'una placa de connexions o, en anglès, protoboard.

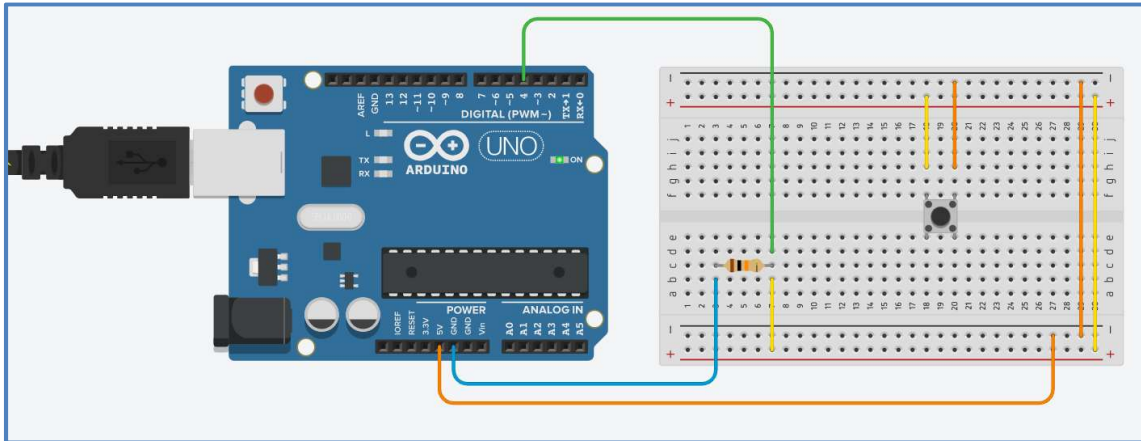


Figura 7: Simulació del circuit bàsic

8.3 Com comptar les pulsacions?

Amb el circuit muntat, és hora d'escriure el programa. Comptar les vegades que s'obra i es tanca el circuit és la funció principal del nostre aparell, per fer-ho podem decantar-nos per dues opcions diferents:

- **Per “polling”:** Fa referència a la lectura constant dels valors cedits generalment per un dispositiu de hardware tant si es produeix algun esdeveniment com si no se'n produeix cap. Aquest mètode té l'inconvenient que, donat que sempre està llegint valors, no s'atura mai i, per tant, consumeix més bateria.
- **Per interrupcions:** Aquest mètode permet que s'activi el programa tan sols quan es produeix l'esdeveniment indicat, de tal manera que no ha d'estar tot el temps llegint els valors. Això fa que sigui molt més eficient i com a conseqüència, que el consum de la bateria disminueixi.

Malgrat els grans avantatges que aporta fer-ho per interrupcions, el software d'Arduino no ho permet, per tant, ens veiem obligats a programar l'aparell pel mètode de polling.



El plantejament inicial és que quan es detecti que el circuit estigui tancat, voldrà dir que el polsador està premut i per tant, haurà de sumar 1 a una variable (variable + 1) tal i com es mostra a la figura 8. Tot i que sembli que amb aquest plantejament hagi de funcionar correctament, no serà així, ja

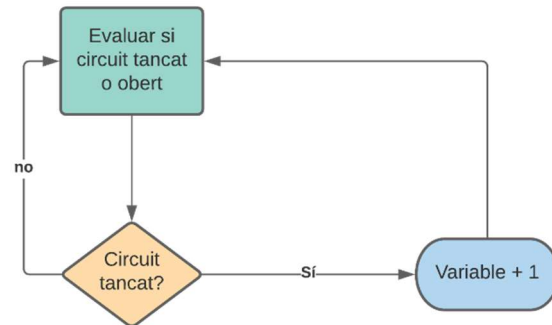


Figura 8: Diagrama de flux del programa

que tenint en compte que el programa es realitza centenes de vegades per segon en forma de bucle³, quan premis el polsador, sumarà 1 a cada volta realitzada pel programa mostrant així un valor incorrecte. És a dir, encara que el temps de la teva pulsació sigui 0,3 segons, el valor de la variable serà d'aproximadament 30 per pulsació. Però no solament volem transformar aquest 30 amb el valor correcte que seria 1, sinó que volem que independentment del temps que duri la pulsació, només incrementi en 1 la variable.

Davant d'aquest repte, vaig buscar per internet però no vaig trobar respostes clares. Al preguntar-ho a un enginyer en telecomunicacions, vaig entendre que el que havia de comptar eren els flancs de pujada i de baixada en lloc de detectar quan estava tancat i quan estava obert el circuit. Tot i així el software d'Arduino no ho permet, fet que va fer que m'hagués d'inventar una manera similar de fer-ho. Finalment, vaig aconseguir inventar un algoritme que al final de cada pulsació, transcrivés el valor erroni per un de correcte. Seguidament es mostra el diagrama de flux i el programa real.

³ **Bucle:** En un circuit elèctric, camí tancat.

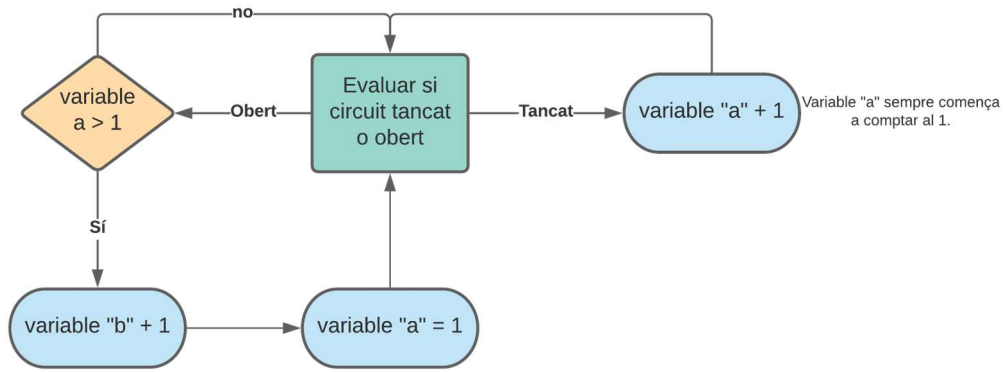


Figura 9: Diagrama de flux del programa

El que fa aquest programa és sumar 1 a variable “a” sempre que detecti el circuit tancat. Fins aquí és exactament el que teníem abans però el que fa corregir l’error, no és mentre està polsat el botó sinó quan ja no ho està. Quan el circuit roman obert, avalua el resultat que ha obtingut durant el període de temps que el circuit ha estat tancat i detecta si hi ha hagut algun error. El primer que fa és comparar el resultat, si aquest és superior a 1 és incorrecte i per tant cal corregir-lo. Per fer-ho entra en joc una nova variable, la variable “b” que serà sempre la correcta. Quan ha detectat l’error incrementa en 1 el valor de “b” i seguidament torna a posar la variable “a” a l’estat inicial 1. Acabat aquest pas, no tornarà a repetir aquesta cadena d’accions fins que la comparació de “a” torni a ser incorrecta, és a dir, quan es torni a deixar de prémer el pulsador una altra vegada. Veure figura 9.

```

#include <TimeLib.h>
#define Pulsador 2
#define Led 13

int inici = 0;
int index = 0;
int contar = 0;
int index2 = 0;

void setup() {
  pinMode (Pulsador, INPUT);
  pinMode (Led, OUTPUT);
  Serial.begin(9600);
  setTime(11,0,0,13,07,2020);
}

void loop() {
  for (index=0; digitalRead (Pulsador) == 1; index+1){
    inici = 1 + inici;
    //Serial.println(inici);
    if (inici > 5000){
      inici = 2;
    }
    delay(2);

    digitalWrite(Led, HIGH);
  }
  for (index2=0; digitalRead (Pulsador) == 0; index2+0){
    digitalWrite(Led, LOW);
    if (inici > 1){
      contar = contar + 1;
    }
  }
}

```

Figura 10: Codi del programa

8.4 L'emmagatzematge de les dades obtingudes:

Un cop el programa sap com interpretar les dades i transformar-les en pulsacions individuals, cal saber què i com emmagatzemar aquestes dades. És evident que el que s’ha de guardar són els valors recollits però també és molt important quan aquestes dades han estat llegides. Per tant, s’ha de saber com



trobar les dades de l'hora actual. El software⁴ o el hardware⁵ són les dos maneres principals tant d'aconseguir la dada temporal com les opcions que hi ha per guardar les dades en el nostre dispositiu.

8.4.1 Opcions d'emmagatzematge:

Per Software:

Per aconseguir memoritzar dades utilitzant programari disposem de tres tipus d'emmagatzematge:

- Memòria **FLASH**: És on s'hi guarda el programa juntament amb el bootloader⁶. Normalment té una capacitat d'entre 1Kb a 4Mb. Es pot iniciar un programa des de la memòria FLASH però no es pot modificar el que hi ha guardat. Aquest tipus d'emmagatzematge té una vida útil d'uns 100.000 cicles d'escriptura i és una memòria no volàtil.
- Memòria **EEPROM** (electrically erasable programmable read-only memory): Normalment es utilitza per a guardar informació després d'haver realitzat un reinici del programa. També s'hi pot guardar des del microcontrolador dades, molt freqüentment és utilitzat per guardar constants de programa. Igual que la memòria FLASH, té una vida limitada d'uns 100.000 cicles d'escriptura. Un altre desavantatge és que tan sols es pot llegir byte per byte la qual cosa es pot fer una mica feixuc. Memòria no volàtil.
- Memòria **SRAM** (static random access memory): És on s'hi guarden els valors de les variables mentre el programa està en funcionament. El desavantatge més gran que té és que la seva memòria és volàtil, és a

⁴ **Software:** (programari en català) Conjunt sistemàtic dels programes d'explotació i dels programes informàtics que serveixen per a aplicacions determinades.

⁵ **Hardware:** (maquinari en català) Conjunt d'elements físics d'un sistema informàtic, que generalment es compon d'una unitat central i de perifèrics.

⁶ **Bootloader:** Petit programa que ha sigut guardat prèviament dins el microcontrolador de la placa i que permet carregar codi sense necessitat de hardware addicional.



dir, a l'apagar el microcontrolador o al reiniciar el programa es perdran tots els valors de les variables.

Per hardware:

La millor manera d'emmagatzemar informació utilitzant hardware, és utilitzant un mòdul de SD o micro SD amb la seva corresponent targeta. Una targeta SD ens permet una memòria molt gran, on s'hi pot escriure quasi indefinidament i a més a més la seva lectura és ràpida, de tot el seu contingut a la vegada i no és volàtil.

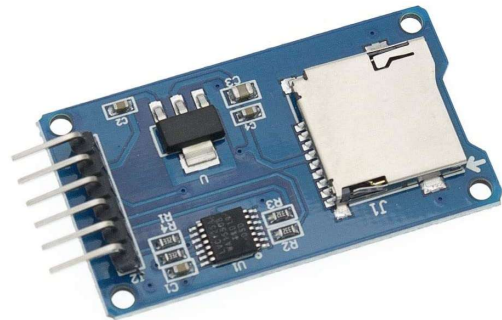


Figura 11: Mòdul lector de targeta micro SD

Així doncs, veient els aspectes a favor i els aspectes en contra de cada mètode d'emmagatzematge arribem a la conclusió que per les funcions que volem aplicar a la memòria, la millor opció és una targeta micro SD.

8.4.2 Els dos recursos per obtenir les dades temporals:

Per Software:

Existeix una llibreria⁷ anomenada *Time.h* que ajuda a fer més amè el procés de programació d'aquesta part de programa. Utilitzant-la, es pot establir una data i hora, llegir-les, a més a més de tenir la possibilitat de comptes enrere, etc. Sens dubte és una bona eina per treballar si vols fer temporitzadors o cronòmetres i pots prescindir de que algunes vegades s'aturi el circuit i al començar de nou a reproduir el codi, el rellotge torni al seu estat preestablert a l'inici del programa.

Per Hardware:

Existeixen dos mòduls capaços de fer la funció d'un rellotge digital anomenats RTC que ve de les paraules en anglès Real Time Clock els quals porten una pila rodona de 3 volts que només es consumeix si la font principal d'alimentació

⁷ **Llibreria** (dins el context de programació): Són parts de codi creades per tercers que podem utilitzar en els nostres projectes i que ens faciliten molt la programació i fa més senzill el fet de programar i de fer-ho entendre.



es troba fora de servei. El fet que incorporin la seva pròpia font d'alimentació dona lloc a que sempre estiguin en funcionament tot i que l'aparell romanguí apagat i que com a conseqüència no es reinicialitzi l'hora cada vegada que encenguem o apaguem el dispositiu tal i com ens passava en el cas anterior.

- RTC DS1307: Aquest mòdul malgrat funcionar suficientment bé, té el problema que les temperatures extremes poden afectar la seva precisió. Estaríem parlant de més de 5 minuts al mes de desfasament.
- RTC DS3231: El fet que aquest mòdul disposi d'un oscil·lador intern que no li afecti tan els canvis de temperatura, permet obtenir una precisió molt més gran, fins al punt de tenir tan sols un desfasament d'uns minuts l'any.



Figura 12: Mòdul RTC DS3231

8.4.3 El mòdul Bluetooth:

Què permet el Bluetooth?

- Facilitar les comunicacions entre equips mòbils.
- Eliminar els cables i connectors entre aquests.
- Oferir la possibilitat de crear petites xarxes inalàmbriques i facilitar la sincronització de dades entre equips personals.

En què consisteix físicament?

Tracta d'un xip de quatre pins:

- VCC: Voltatge positiu d'alimentació. N'hi ha que tan sols poden funcionar a 3.3 volts tot i que la majoria ja venen acondicionats per a que puguin suportar un voltatge d'entre 3.3V a 6V.
- GND: 0 V
- TX: És el pin pel qual es transmeten les dades, aquest pin ha de ser contactat al pin RX (recepció) del Arduino.



- RX: Aquest és el pin encarregat de rebre la informació de la placa Arduino. Aquest pin ha d'anar connectat al pin TX (transmissió) del Arduino.

Gràcies a que tant l'Arduino com el mòdul Bluetooth disposen dels dos pins TX i RX en cadascun, es possible rebre i enviar informació del mòdul a la placa Arduino igual que a la inversa.

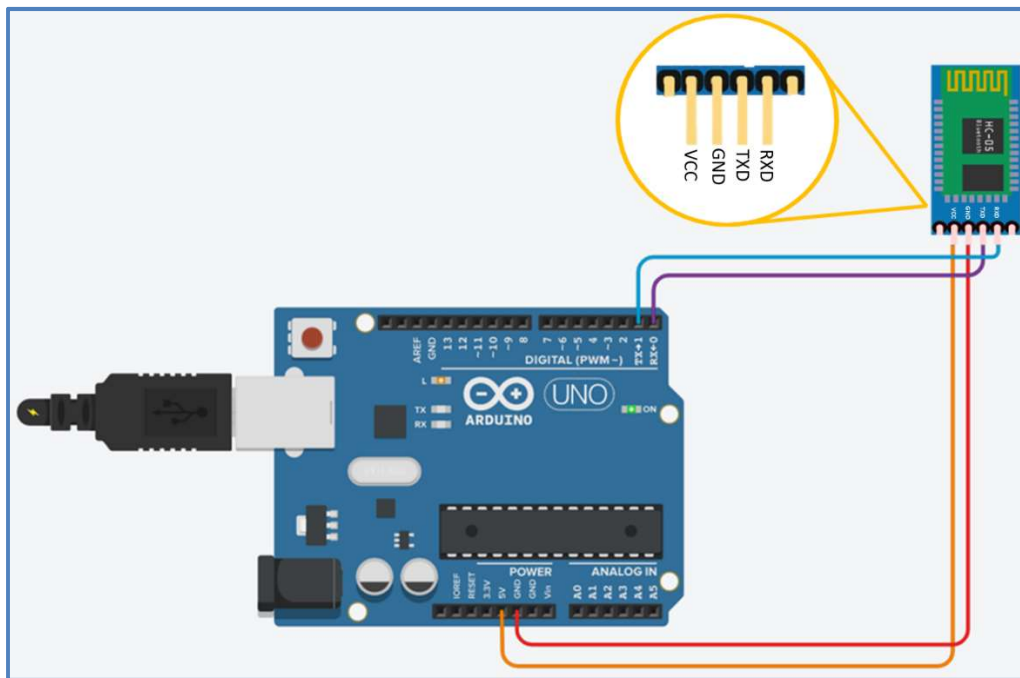


Figura 13: Connexions del mòdul bluetooth HC-05 cap a l'Arduino

8.4.4 La connexió dels mòduls amb el circuit base:

Un cop presa la decisió d'utilitzar mòduls externs per guardar dades i per obtenir el valor temporal, és hora de connectar aquests xips amb el circuit que teníem anteriorment.

Com que hi ha molts pins que han d'anar connectats al mateix port de l'Arduino com ara el port de terra (GND) i a més a més hi ha molts cables, hem de buscar una manera d'optimitzar l'espai per fer-ho d'una manera més ordenada. Així doncs, amb l'ajuda de la placa protoboard, connectem una de les seves files interiors, al GND de l'Arduino, així aconseguim que tota aquella fila faci de port de terra i haurem multiplicat per 5 aquest port. Farem el mateix amb els 5 volts (VCC), però enlloc de connectar-ho en una fila interior, ho connectarem a



una columna exterior, d'aquesta manera també tenim molts més punts per connectar-hi voltatge. En la figura que es mostra seguidament es pot observar com s'han connectat tots els mòduls i el polsador amb la placa Arduino. Cables vermells (GND), cables taronja (VCC).

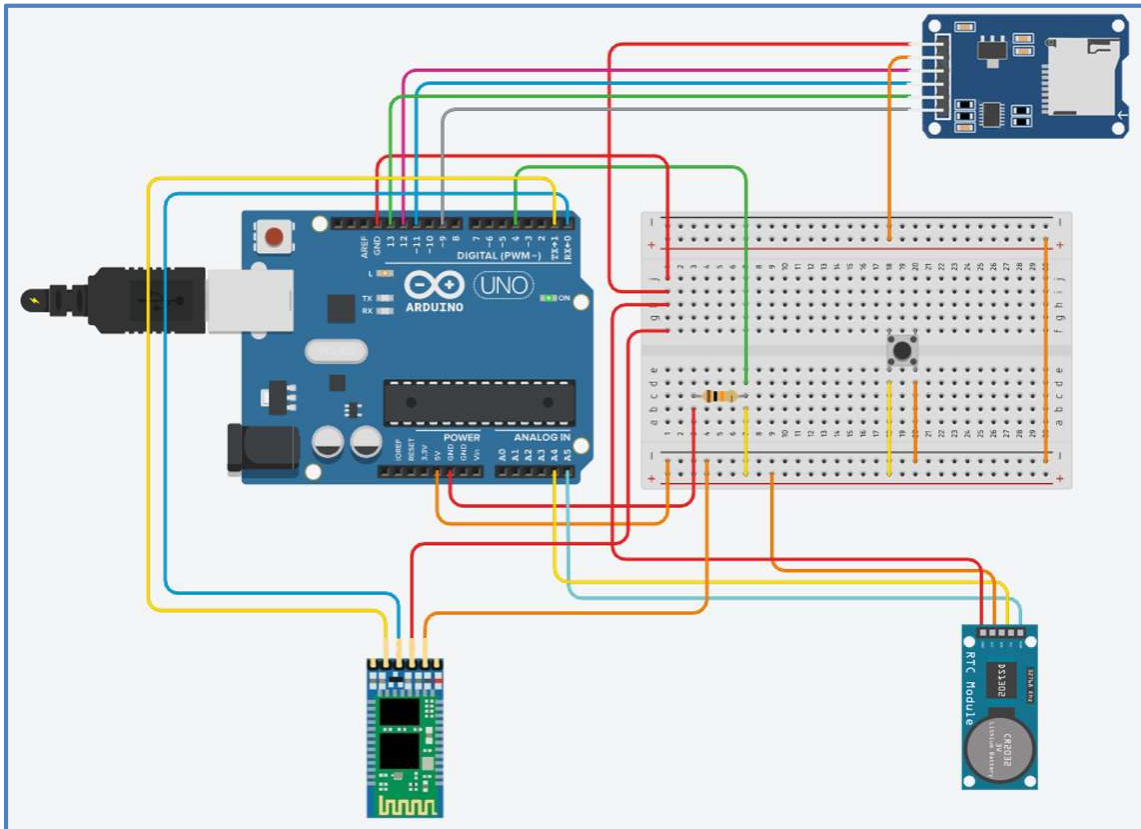


Figura 14: Circuit final

8.4.5 El programari per guardar els valors:

És important establir una freqüència per guardar les dades perquè si hi ha un tall d'alimentació, no es perdin tots els valors obtinguts prèviament. És imprescindible que aquesta funció tan sols es dugui a terme en dos moments. Un d'ells és quan el circuit està obert, que és la majoria de temps i l'altre moment és quan el circuit està tancat. Ja que si no fos així, en el suposat cas que coincidís quan el circuit està tancat i a temps igual a 10 segons, se saltaria el pas d'emmagatzemar els valors i per tant, hi hauria un període de temps en el qual no tindriem dades.

El que se li diu al programa és que tan bon punt hagi acabat de o bé sumar 1 a variable "a", o de comprovar que la funció lògica de "a" > 1 sigui negativa,



compari si els segons que envia el mòdul RTC són igual a 10 segons, esdeveniment que succeirà només una vegada per minut. Seguidament, si la comparació és certa, se li diu que guardi el valor de la variable "b" juntament amb l'hora i la data en que es guarda. Finalment es reinicia "b" posant-se a 0.

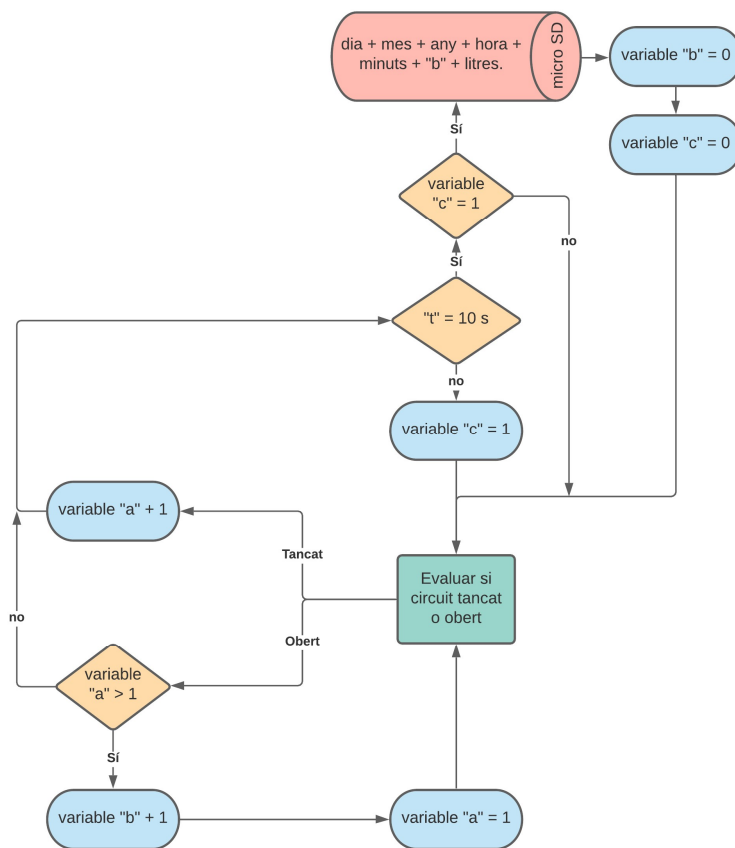


Figura 15: Diagrama de flux del programa final

Tanmateix, ens tornem a trobar amb el mateix problema, la condició de "t = 10 segons" dura un total d'1 segon, fet que farà que es repeteixi moltes vegades el bucle i que per tant, guardi molts més cops les dades dels que necessitem. Tal i com s'ha fet anteriorment, utilitzarem una variable auxiliar que anomenem "c". Aquesta serà la que permetrà guardar els valors una vegada per

segon, deixant introduir d'aquesta manera el valor a la micro SD només quan el valor de la variable "c" sigui igual a 1. Així doncs, quan hagi acabat d'emmagatzemar les dades, "c" canviarà el seu valor en "0" i tan sols tornarà a canviar-lo quan la comparació de "t = 10 segons" sigui falsa. Veure figura número 15.



9. L'alimentació.

Per a que funcioni tot el prototip correctament és necessària una font d'alimentació. Ja que el nostre aparell haurà d'estar en llocs on no hi haurà un endoll a prop, necessitarem piles o una placa solar.

L'opció d'una placa solar col·locada a la teulada que alimenti el circuit, juntament amb una petita bateria que emmagatzemi l'energia és molt bona sinó fos per l'increment de preu que suposaria. Fent referència a un dels objectius inicials, fer l'aparell el més barat possible, aquesta opció de la placa solar queda descartada. Conseqüentment, s'utilitzarà una pila de nou volts que ofereix una llarga durada de funcionament.



Figura 16: Imatge d'una pila de 9V



10. L'aplicació.

La creació d'una aplicació per recollir les dades del datalogger fa que la seva lectura sigui molt més àgil i fluida. La transmissió d'aquests valors es farà mitjançant la connexió anomenada Bluetooth.

10.1 Bluetooth:

El Bluetooth permet connectar dos dispositius per a l'intercanvi d'informació sense fils com poden ser: dades, música, etc. Molt habitualment disposen d'aquesta tecnologia sobre tot, els dispositius mòbils com per exemple: telèfons mòbils, ordinadors portàtils, etc.

10.1.2 Context Històric:

El nom de "Bluetooth" fa referència al rei danès i noruec Harald Blåtand en anglès, Harald Bluetooth (dent blava) possiblement degut a una malaltia anomenada eritroblatosi fetal que feia que algunes dents es tornessin blaves. Aquest rei va ser conegut per unificar les tribus daneses i convertir-les en cristianes. Amb aquesta unificació de Dinamarca i Noruega va posar fi a l'era vikinga.

Jim Kardach, quan va inventar un sistema que permetia als telèfons mòbils comunicar-se amb els ordenadors i unificar la comunicació sense fil, va proposar anomenar a aquesta nova tecnologia com "Bluetooth".

A l'any 2000 el "Bluetooth Special Interest Group (SIG)" compost per diferents empreses molt importants com ara Nokia, Ericsson, 3Com, Lucent, Microsoft, Motorola, Toshiba, IBM i Intel van llençar un logotip que jugava amb la runa⁸ Hagall (ᚷ) i la runa Berkana (ᚱ) que corresponen a les inicials del rei Harald Bluetooth.

⁸ Runa: Signe de l'alfabet emprat antigament pels pobles germànics.

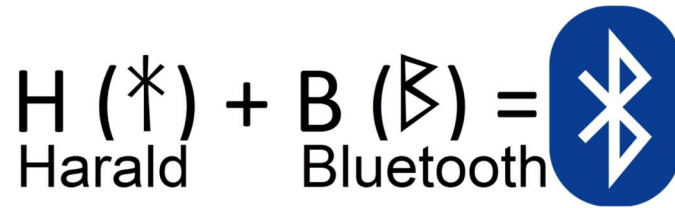


Figura 17: Esquema del naixement del logotip de Bluetooth

10.2 Quines funcions s'han d'incloure dins l'aplicació?

L'aplicació, tal i com s'ha dit anteriorment té com a objectiu principal fer d'intermediari entre les dades obtingudes pel datalogger i l'usuari, facilitant així, la lectura dels valors. Per fer-ho, l'app disposarà de varies funcions.

- Connexió i desconnexió del mòbil cap al datalogger.
- Obtenció de les dades i emmagatzemar-les.
- Decisió de la freqüència la qual l'aparell haurà de recollir els valors.
- Possibilitat de pujar les dades obtingudes al núvol.
- Elecció de l'idioma de la interfície⁹.
- Visualitzar amb quin mòdul Bluetooth està connectat.
- Suprimir les dades.

10.3 Programari de l'aplicació i les seves pantalles.

És molt habitual que una aplicació estigui composta per varies pantalles diferents, per tal d'assignar a cadascuna d'elles una funció específica.

Com estan organitzades les diferents pantalles de la plataforma és molt important perquè si l'estructura és clara i intuïtiva ajudarà l'usuari que l'hagi de fer servir a sentir-se més còmode.

⁹ **Interfície:** Unitat funcional que permet posar en comunicació sistemes amb diferents funcions o característiques.



En el nostre cas, l'app consistirà en quatre pantalles diferents:

10.3.1 Pantalla d'inici:

Disseny i funció:

Aquesta pantalla té la funció de rebre l'usuari dins l'aplicació, per tant, s'hi mostrarà el logotip de l'app, i un símbol el qual representa que l'aplicació s'està carregant.

Programa:

El programari d'aquesta primera pantalla, és molt senzill, tan sols se li diu que quan hagi passat un segon de l'obertura de l'aplicació, canviï de pantalla. En cas de ser el primer cop que s'obre, la mateixa aplicació el redirigirà cap a la pantalla d'elecció d'idioma, en cas contrari, el redirigirà cap a la pantalla l'idioma del qual ja l'haurà triat en un altre moment.

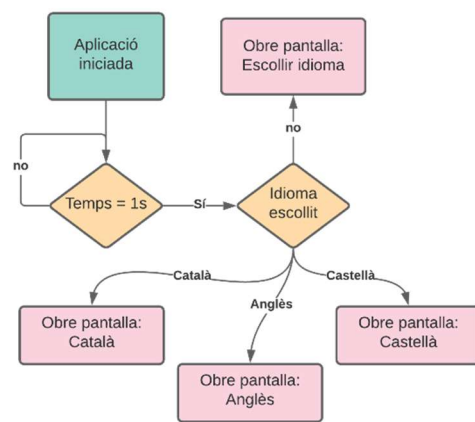


Figura 18: Diagrama de flux del programa de la pantalla inici

10.3.2 Pantalla d'elecció d'idioma:

Disseny i funció:

La funcionalitat d'aquesta pantalla és que l'usuari pugui escollir l'idioma amb el qual vol que es mostri la interfície de l'aplicació. Així doncs, es mostrarà tants botons com idiomes hi ha disponibles, és a dir, tres; català, castellà, i anglès.

Programa:

El programari d'aquesta pantalla es basa en que l'idioma que s'esculli a l'inici no s'hagi de triar cada cop que s'obri l'aplicació, per tant, hem de crear una base de dades on a cada

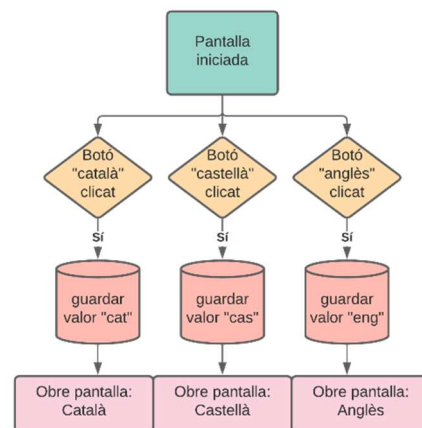


Figura 19: Diagrama de flux del programa de la pantalla d'escollir l'idioma



idioma li correspondrà un valor, per exemple, el valor de l'idioma "català" serà "cat". D'aquesta manera, quan s'entri a l'aplicació, si el valor guardat dins la base de dades és "cat", la pantalla "inici" canviarà automàticament a pantalla "català".

10.3.3 Pantalla principal.

Disseny i funció:

La pantalla està distribuïda en tres parts, la primera fa referència a la connectivitat entre el dispositiu i el mòdul Bluetooth del datalogger juntament amb l'idioma de la interfície; la segona, a la freqüència de recollida de dades; i la tercera, és la que involucra tot el que fa referència a les dades.

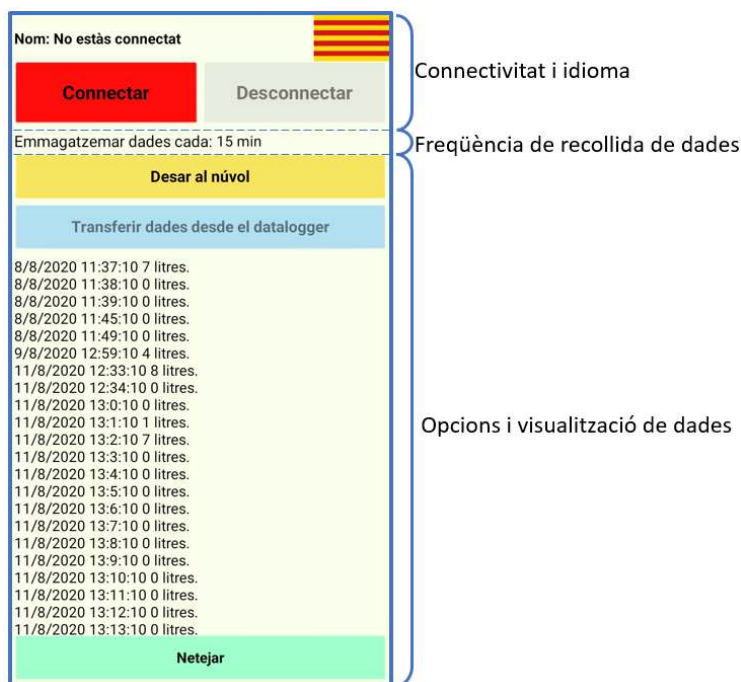


Figura 20: Separació de les tres parts en que està dividida la pantalla principal

Programa:

El programa d'aquesta pantalla també el podríem dividir en tres parts, igual que el seu disseny.

La primera és el feedback des del telèfon mòbil amb el datalogger. Quan l'usuari intenta connectar el mòdul Bluetooth de l'aparell amb el del mòbil, el primer, si ha estat connectat correctament, envia una senyal a l'aplicació la qual la interpretarà com a una connexió exitosa i, per tant, ens ho farà saber



canviant el color del botó “Connectar” de vermell a verd i també ens mostrarà el nom del mòdul Bluetooth al qual estem connectats. En cas contrari, es mostrarà una notificació on hi dirà que la connexió no s’ha pogut dur a terme i el botó “Connectar” seguirà de color vermell.

La segona part, consisteix en indicar al datalogger amb quina freqüència es vol que les dades siguin recollides. Aquesta funció tan sols està disponible quan els dos dispositius romanen connectats. L’usuari haurà de triar entre 1 min, 5 min, 15 min, 60 min, 12 hores o 24 hores. Depenent dels minuts seleccionats, l’aplicació enviarà un senyal o un altre cap al datalogger, el qual executarà parts diferents del seu programa depenent de la freqüència escollida.

Finalment, a l’última part de la pantalla, es visualitzarà les dades recollides i es decidirà què es vol fer amb elles. Tan bon punt s’efectuï la connexió amb el datalogger, les dades seran enviades i visualitzades. També es podran actualitzar clicant al botó “transferir dades des del datalogger”, de la mateixa manera, també es podran suprimir o pujar-les al núvol. Amb l’objectiu d’evitar que les dades es suprimeixin per equivocació, l’aplicació disposarà d’un sistema d’alertes el qual es mostraran quan es vulgui modificar les dades, com ara, suprimir-les o pujar-les al núvol.

Així doncs, el diagrama de flux del programa de la pantalla queda de la manera com es mostra a la figura 22.

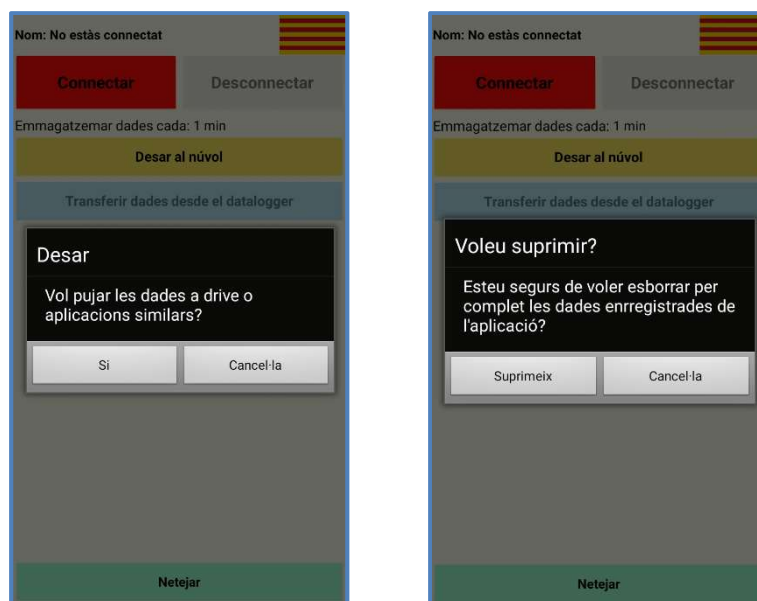


Figura 21: Missatges d'alerta

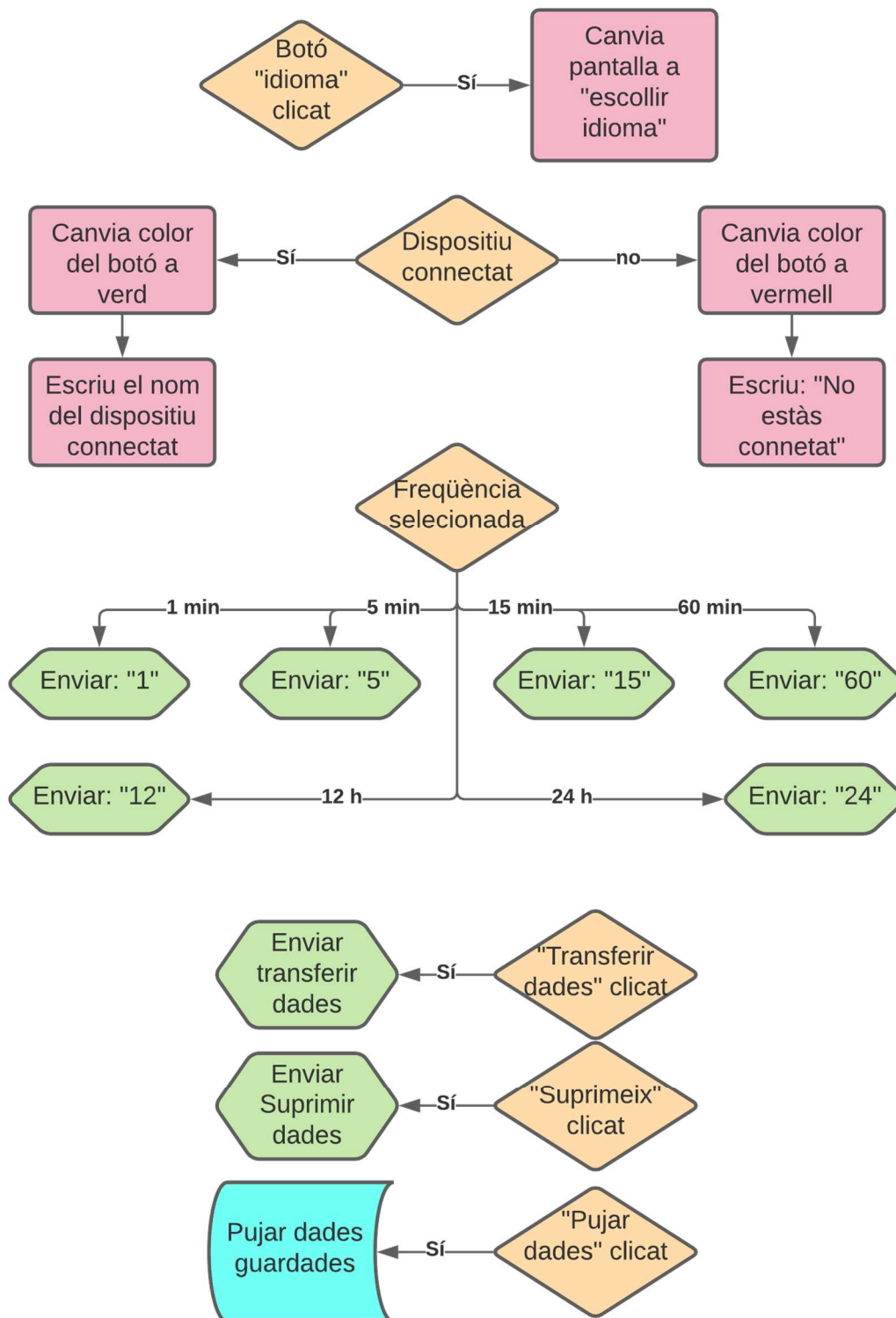


Figura 22: Diagrama de flux del programa de la pantalla principal



11. Càlcul de costos i pressupost.

Per tal que el producte final compleixi un dels objectius inicials com és el de reduir al màxim el preu del producte, és necessari una bona gestió dels diners. Així doncs, escriuré tots els productes que necessito amb un preu orientatiu al costat.

Producte	Preu unitat
Placa Arduino	10,00 €
Targeta micro SD	8,00 €
Femella jack	0,90 €
Placa protoboard	2,98 €
Pila 9v	3,75 €
Placa protoboard per a soldar	2,60 €
Mòdul RTC (rellotge) pila 3v inclosa	6,99 €
Mòdul micro SD	3,50 €
Caixa polsadors (180 pcs)	0,05 €
Caixa resistors (525 resistors)	0,02 €
Cables Arduino (120 cables)	0,05 €
Filament ABS per a impressora 3D 1kg	12,60 € (0,70kg)
Premsaestopa	3,50 €
Cost del producte:	54,94 €



12. Estada a Lab-Ferrer:

Per aprendre més sobre el tema, em va sorgir l'oportunitat de fer pràctiques als Laboratoris Ferrer de Cervera on hi vaig estar un total de tres setmanes.

Lab-Ferrer és una empresa que opera internacionalment, està especialitzada en la producció i venda de sistemes de lectura per optimitzar recursos com l'aigua. Ofereixen un gran ventall de

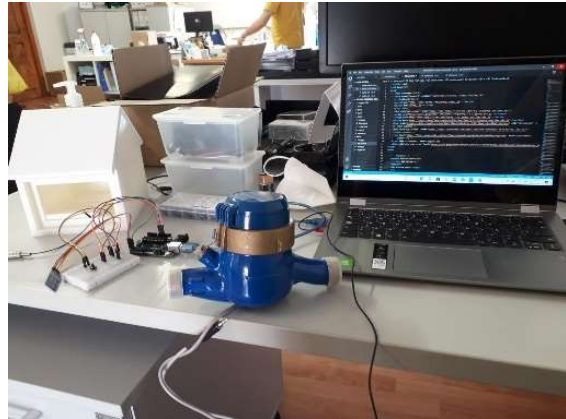


Figura 23: Imatge de la meva estada a Lab-Ferrer

productes com ara: sondes i sensors, sistemes d'adquisició de dades i software (datalogger), a més a més d'una excel·lent assistència tècnica. Alguns dels productes més coneguts són:

- **ECH2OSYSTEM:** Gestió eficient de l'aigua de reg.
- **SOILRHO:** Conductivitat tèrmica i resistivitat tèrmica.
- **AQUALAB:** Activitat d'Aigua i Isotermes de Sorció.
- **EFOOD PRINT:** Empremta Hídrica i empremta de Carboni.

Durant aquest temps vaig poder veure com funcionava l'empresa i vaig poder preguntar tot el que no sabia fer o no m'acabava de funcionar correctament. També em van ajudar a replantejar el programa des d'una altra perspectiva, la qual em va ser de molta utilitat.



13. Viabilitat:

Un cop acabat el prototip s'ha d'avaluar la seva viabilitat amb l'objectiu de que es pugui comercialitzar. Per tal que sigui viable ens hem de fixar en si, quan es posa a la pràctica, és possible realitzar amb eficàcia les funcions que s'han proposat teòricament, així com si pot ser fabricat en una cadena de producció, la facilitat amb la qual els usuaris utilitzen el producte i si es viable obtenir beneficis.

Quan considerem aquests quatre objectius, podem arribar a la conclusió que és viable produir i comercialitzar aquest producte, gràcies a que la protecció del circuit elèctric ha estat dissenyada en tres dimensions, el que facilitaria la reproducció d'aquesta caseta. A més a més, l'aplicació és senzilla d'utilitzar i en relació al circuit elèctric tan sols s'ha d'encendre. Tot i així, si es volen obtenir beneficis, el preu del producte augmentaria i per tant, s'hauria de prescindir d'un dels objectius principals del treball, fer l'aparell econòmic.



14. Conclusions:

Un cop acabat el projecte i a l'analitzar quins eren els objectius inicials i els que hem aconseguit finalment ens adonem que s'han complert tots:

Des d'un principi s'ha construït l'aparell valorant totes les opcions que s'hi podien aplicar, escollint d'aquesta manera les més adequades en cada ocasió.

S'ha aconseguit crear un datalogger que compta les pulsacions, guarda les dades amb els seu temps corresponent.

En relació al programari, durant tot el projecte s'ha dut a terme un procés de prova i error, que ha permès saber on estaven les errades per solucionar-les

També s'ha creat l'aplicació amb la qual podem llegir els valors, pujar-los al núvol, esborrar-los o escollir la freqüència de transmissió de dades. I per si fos poc, està disponible en diferents idiomes.

Pel que fa a la protecció del datalogger, s'ha imprès una caseta en tres dimensions que protegeix el sistema elèctric de dins i es tanca hermèticament gràcies a la junta tòrica posada a la porta. A més a més, un dels objectius principals que era fer un preu econòmic s'ha assolit amb la xifra de 54,94 €.



15. Aspectes a millorar:

Com en tot projecte sempre hi ha marge de millora. En el meu cas, penso que seria bo posar interruptors per encendre i apagar el mòdul bluetooth quan no s'hagi d'utilitzar, d'aquesta manera s'estalviaria energia i per tant, el temps de duració de la bateria augmentaria.

Un altre àmbit en el qual es podria millorar és en l'elecció de la freqüència de recollida de dades, per ara les opcions estan establertes, estaria bé que l'usuari pogués triar l'interval de temps que volgués.

També per reduir energia, i parlant més del programari, estaria bé que ho haguéssim pogut programar per interrupcions i no per polling, que com ja he dit en el treball, seria un fet que milloraria la durabilitat de la bateria.

Per acabar, hagués estat bé fer una simulació d'un cas pràctic de principi a fi.



16. Webgrafia:

- Administrator. (s. f.). Programari lliure. Recuperat 16 de maig de 2020, de Xtec.cat website: <https://ioc.xtec.cat/educacio/recursos-programari-lliure>
- Arduino des de cero en Español - Capítulo 24 - Bluetooth HC-05 Introducción y comandos AT. (2017, agost 6). Recuperat 29 de juliol de 2020, de <https://www.youtube.com/watch?v=5SmKOUHhmWk>
- attachInterrupt(). (s. f.). Recuperat 29 de juliol de 2020, de Arduino.cc website: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>
- Ceupe, E. B. (2020, juliol 22). ¿Qué es la programación? Recuperat 21 de maig de 2020, de Ceupe.com website: <https://www.ceupe.com/blog/que-es-la-programacion.html>
- ¿cómo afecta la temperatura a las baterías? (s. f.). Recuperat 23 de juliol de 2020, de https://www.damiasolar.com/actualidad/blog/articulos-sobre-la-energia-solar-y-sus-componentes/temperatura-baterias_1
- designthemes. (2014, novembre 29). Módulo BlueTooth HC-06. Recuperat 22 de juliol de 2020, de Prometec.net website: <https://www.prometec.net/bt-hc06/>
- Efecto de la temperatura en las baterías. (s. f.). Recuperat 21 de juliol de 2020, de Rebacas.com website: https://www.rebacas.com/blog-baterias/23_Efecto-de-la-temperatura-en-las-bater%C3%ADas.html
- FayerWayer. (2011, setembre 28). La historia del nacimiento de Bluetooth. Recuperat 10 de desembre de 2020, de FayerWayer website: <https://www.fayerwayer.com/2011/09/la-historia-del-nacimiento-de-bluetooth/>



- Fernández, Y. (2018, juliol 21). Qué es Arduino, cómo funciona y qué puedes hacer con uno. Recuperat 23 de maig de 2020, de Xataka.com website: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- FreeCAD. (s. f.). Recuperat 24 de juny de 2020, de Freecadweb.org website: <https://www.freecadweb.org/>
- gnu.org. (s. f.). Recuperat de <https://www.gnu.org/philosophy/free-sw.ca.html>
- Historia. (s. f.). Recuperat 19 de desembre de 2020, de Weebly.com website: <https://arduinohtics.weebly.com/historia.html>
- Historia Del Arduino. (s. f.). Recuperat 19 de desembre de 2020, de Calameo.com website: <https://es.calameo.com/books/00452476670aee7e31f5b>
- Ibáñez. (2012, juny 27). Las altas temperaturas son el peor enemigo de las baterías de iones de litio. Recuperat 24 de juliol de 2020, de Motorpasion.com website: <https://www.motorpasion.com/coches-hibridos-alternativos/las-altas-temperaturas-son-el-peor-enemigo-de-las-baterias-de-iones-de-litio>
- Instrumentación y Sensores para medir actividad del agua e isothermas. (s. f.). Recuperat 15 de juliol de 2020, de Lab-ferrer.com website: <https://www.lab-ferrer.com/>
- Interrupt vs polling. (2017, febrer 24). Recuperat 4 de juny de 2020, de <https://www.youtube.com/watch?v=f7LLKL1g-kk>
- jecrespom. (2016a, novembre 7). Memoria Arduino. Recuperat 26 de juliol de 2020, de Wordpress.com website: <https://aprendiendoarduino.wordpress.com/2016/11/08/memoria-arduino/>
- jecrespom. (2016b, novembre 13). Bluetooth en Arduino. Recuperat 30 de juliol de 2020, de Wordpress.com website:



<https://aprendiendoarduino.wordpress.com/2016/11/13/bluetooth-en-arduino/>

- jecrespom. (2017, juny 21). Memoria Flash, SRAM y EEPROM. Recuperat 21 de maig de 2020, de Wordpress.com website: <https://aprendiendoarduino.wordpress.com/2017/06/21/memoria-flash-sram-y-eeeprom-3/>
- JM Industrial. (s. f.). ¿Dónde se usan los transductores de presión? Recuperat 31 de juliol de 2020, de Com.mx website: <https://www.jmi.com.mx/literatura/blog/item/11-data-logger-que-es.html>
- Juan Gonzalez Gomez. (s. f.). Recuperat 30 d'agost de 2020, de Youtube.com website: <https://www.youtube.com/channel/UCIGE0Dq13T-64rNIMWxnMRQ>
- La historia de Arduino. (2016, març 28). Recuperat 19 de desembre de 2020, de https://www.youtube.com/watch?v=qUBTu_7wcew
- La importancia del Datalogger en una instalación. (2019, juliol 5). Recuperat 30 de maig de 2020, de <https://www.youtube.com/watch?v=w4hJsoblETw>
- La temperatura y el correcto funcionamiento de las baterías. (s. f.). Recuperat 16 de juliol de 2020, de Autosolar.es website: <https://autosolar.es/blog/energia-solar-fotovoltaica/efecto-de-la-temperatura-en-las-baterias>
- Luis. (2016, octubre 18). Reloj y calendario en Arduino con los RTC DS1307 y DS3231. Recuperat 23 de juliol de 2020, de Luisllamas.es website: <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>
- MIT App Inventor. (s. f.). Recuperat 30 de juliol de 2020, de Mit.edu website: <https://appinventor.mit.edu/>
- Módulo SD para Arduino: ¿cómo usar tarjetas de memoria en Arduino? (2019, abril 3). Recuperat 26 de juliol de 2020, de



Tarjetasdememoria.info

website:

<https://www.tarjetasdememoria.info/micro-sd-para-arduino/>

- Programación. (2020, agost 7). Recuperat 31 de novembre de 2020, de Neoattack.com website: <https://neoattack.com/neowiki/programacion/>
- Programación (Informática) - Qué es, información, lenguajes. (2020, agost 16). Recuperat 8 de gener de 2021, de Concepto.de website: <https://concepto.de/programacion/>
- ¿Que es Arduino? (2014, noviembre 20). Recuperat 23 de primavera de 2020, de Arduino.cl website: <https://arduino.cl/que-es-arduino/>
- ¿Qué es Arduino y para qué sirve? (s. f.). Recuperat 16 de gener de 2021, de Mastoner.com website: <https://www.mastoner.com/blog/post/%C2%BFqu%C3%A9-es-arduino-y-para-qu%C3%A9-sirve/>
- ¿Qué es el grado de protección IP? - TELTRONIC. (2018, agost 9). Recuperat 19 de juliol de 2020, de Teltronic.es website: <https://www.teltronic.es/que-es-el-grado-de-proteccion-ip/>
- ¿Qué es un Data Logger y cómo funciona? (s. f.). Recuperat 16 de juníol de 2020, de Dewesoft.com website: <https://dewesoft.com/es/daq/que-es-un-data-logger>
- Que son los Interruptor magnético Reed-switch. (s. f.). Recuperat 21 de desembre de 2020, de Shoptronica.com website: <https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/3981-que-son-los-interruptor-magnetico-reed-switch-0689593949974.html>
- Reloj con Arduino, cómo controlar los tiempos con un RTC. (2017, febrer 21). Recuperat 30 de juliol de 2020, de Programarfacil.com website: <https://programarfacil.com/blog/arduino-blog/reloj-con-arduino-rtc/>
- Signalling: Interrupts and polling. (2013, octubre 16). Recuperat 12 de juny de 2020, de <https://www.youtube.com/watch?v=XVELINDuwfE>



- Tinkercad. (s. f.). Recuperado 3 de gener de 2021, de Tinkercad.com website: <https://www.tinkercad.com/>
- Tipos de memoria de Arduino, qué son y para qué sirven. (2016, febrer 5). Recuperat 6 de septiembre de 2020, de Programarfacil.com website: <https://programarfacil.com/blog/tipos-de-memoria-de-arduino/>
- Wikipedia contributors. (2020, novembre 7). Polling (computer science). Recuperat 31 de juliol de 2020, de Wikipedia, The Free Encyclopedia website: [https://en.wikipedia.org/w/index.php?title=Polling_\(computer_science\)&oldid=987548890](https://en.wikipedia.org/w/index.php?title=Polling_(computer_science)&oldid=987548890)
- Wikipedia contributors. (s. f.-a). Arduino. Recuperat 8 d'agost de 2020, de Wikipedia, The Free Encyclopedia website: <https://es.wikipedia.org/w/index.php?title=Arduino&oldid=131437790>
- Wikipedia contributors. (s. f.-b). Bluetooth. Recuperat 21 de decembre de 2020, de Wikipedia, The Free Encyclopedia website: <https://es.wikipedia.org/w/index.php?title=Bluetooth&oldid=132360295>
- Wikipedia contributors. (s. f.-c). Enregistrador de dades. Recuperat 22 de decembre de 2020, de Wikipedia, The Free Encyclopedia website: https://ca.wikipedia.org/w/index.php?title=Enregistrador_de_dades&oldid=23982836
- Wikipedia contributors. (s. f.-d). Grado de protección IP. Recuperat 31 de juny de 2020, de Wikipedia, The Free Encyclopedia website: https://es.wikipedia.org/w/index.php?title=Grado_de_protecci%C3%B3n_IP&oldid=130427827
- Wikipedia contributors. (s. f.-e). Grau de protecció IP. Recuperat 31 de juny de 2020, de Wikipedia, The Free Encyclopedia website: https://ca.wikipedia.org/w/index.php?title=Grau_de_protecci%C3%B3n_IP&oldid=25957993



- Wikipedia contributors. (s. f.-g). Programación. Recuperat 31 de juliol de 2020, de Wikipedia, The Free Encyclopedia website: <https://es.wikipedia.org/w/index.php?title=Programaci%C3%B3n&oldid=131536511>
- Wikipedia contributors. (s. f.-h). Programari lliure. Recuperat 15 de setembre de 2020, de Wikipedia, The Free Encyclopedia website: https://ca.wikipedia.org/w/index.php?title=Programari_lliure&oldid=25843124
- Wikipedia contributors. (s. f.-i). Reed switch. Recuperat 9 de setembre de 2020, de Wikipedia, The Free Encyclopedia website: https://es.wikipedia.org/w/index.php?title=Reed_switch&oldid=117259268
- Wikipedia contributors. (s. f.-j). Registrador de datos. Recuperat 23 de novembre de 2020, de Wikipedia, The Free Encyclopedia website: https://es.wikipedia.org/w/index.php?title=Registrador_de_datos&oldid=128774156
- Que es un Datalogger? (s. f.). Recuperat 5 de primavera de 2020, de Com.mx website: <https://www.finaltest.com.mx/product-p/art-4.htm>
- [Aplicació creada amb la plataforma AppInventor.](#)
- [Diagrames de flux fets amb el programa Lucidchart.](#)
- [Esquemes dels circuits fets amb el programa TinkerCad.](#)
- Figura 1: Font pròpia
- Figura 2: https://www.researchgate.net/figure/Battery-capacity-battery-life-compared-at-different-temperatures-1_fig2_309681258
- Figura 3: Font pròpia.
- Figura 4: Font pròpia.
- Figura 5: <https://handsontec.com/index.php/product/magnetic-reed-switch-200v-0-5a-molded/>
- Figura 6: Font pròpia.



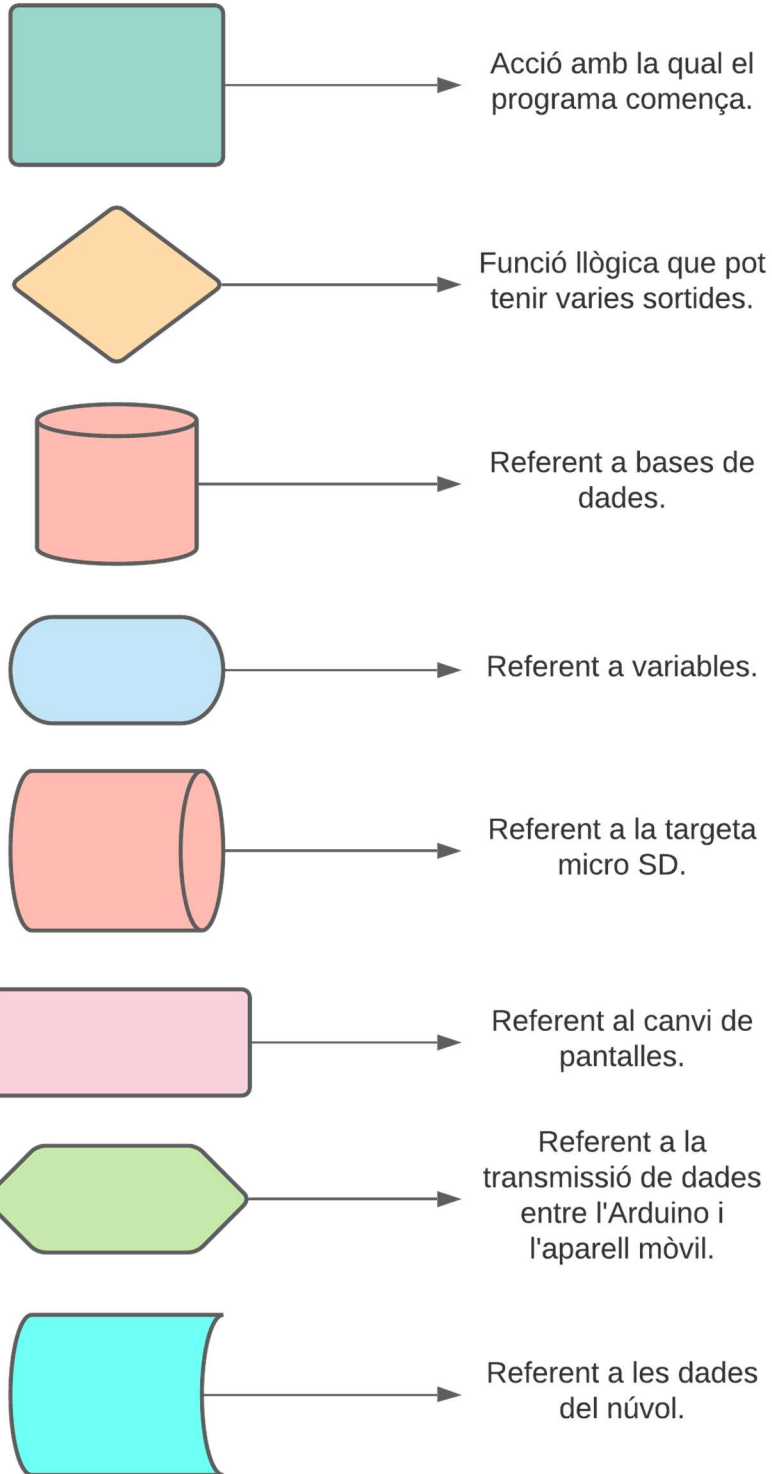
- Figura 7: Font pròpia.
- Figura 8: Font pròpia.
- Figura 9: Font pròpia.
- Figura 10: Font pròpia.
- Figura 11 : https://www.amazon.es/lector-tarjetas-m%C3%B3dulo-interfaces-convertidor/dp/B085XQDWC4/ref=sr_1_36?dchild=1&keywords=micro+s+d+module&qid=1610881798&sr=8-36
- Figura 12 : <http://microelectronicadesign.com/producto/modulo-reloj-ds3231/>
- Figura 13: Font pròpia.
- Figura 14: Font pròpia.
- Figura 15: Font pròpia.
- Figura 16: <https://mexico.newark.com/duracell/mn1604bkd/battery-alkaline-manganese-dioxide/dp/65W8052>
- Figura 17: <https://pc-solucion.es/2018/08/12/bluetooth-la-historia-del-nombre/>
- Figura 18: Font pròpia.
- Figura 19: Font pròpia.
- Figura 20: Font pròpia.
- Figura 21: Font pròpia.
- Figura 22: Font pròpia.
- Figura 23: Font pròpia.
- Figura 24: Font pròpia.
- Figura 25: Font pròpia.
- Figura 26: Font pròpia.
- Figura 27: Font pròpia.
- Figura 28: Font pròpia.
- Figura 29: Font pròpia.



- Figura 30: Font pròpia.



Annex 1: Diccionari per entendre millor els diagrames de flux.





Annex 2: Codi del programa bàsic.

```
#include
<TimeLib.h>

#define Pulsador 2
#define Led 13

int inici = 0;
int index = 0;        //pulses i t'ho mostra//
int contar = 0;
int index2 = 0;

void setup() {
  pinMode (Pulsador, INPUT);
  pinMode (Led, OUTPUT);
  Serial.begin(9600);
  setTime(11,0,0,13,07,2020);
}

void loop() {
  for (index=0; digitalRead (Pulsador) == 1; index+1){
    inici = 1 + inici;
    //Serial.println(inici);
    if (inici > 5000){
      inici = 2;
    }
  }
  delay(2);

  digitalWrite(Led, HIGH);
}
for (index2=0; digitalRead (Pulsador) == 0; index2+0){
  digitalWrite(Led, LOW);
  if (inici > 1){
    contar = contar + 1;

    Serial.print(hour());
    Serial.print(":");
    Serial.print(minute());
    Serial.print(":");
    Serial.print(second());

    Serial.print(" ");
    Serial.print(day());
    Serial.print("/");
```



```
Serial.print(month());  
Serial.print("/");  
Serial.print(year());  
  
Serial.print(" --> ");  
Serial.print (contar);  
    inici = 0;  
Serial.println(" litres.");  
    }  
    else {}  
    }  
  
}
```



Annex 3: Codi del programa final:

```
//Llibreries
#include <SPI.h>

#include <SD.h>

#include <Wire.h>

#include <RTClib.h>

#include <Timelib.h>

#define Pulsador 2
#define Transmisor 7
#define Led 13
// Declarem un RTC DS3231
RTC_DS3231 rtc;

int inicio = 0;
//int inicioAN = 0;
//int estadoAC = 0;
//int estadoAN = 0;
int indice = 0;          //cada 10 segons
int contar = 0;
int indice2 = 0;
int suma = 1;
char valor;
File logFile;
//File dataFile;

//int dataLine;
//int database;

void setup() {
  Serial.print("Começa programa");
  Serial.print("Iniciant SD");
  if (!SD.begin(9))
  {
    Serial.println("Error al iniciar");
    return;
  }
  Serial.println("Iniciat correctament");
```



```
pinMode (Pulsador, INPUT);
pinMode (Transmisor, INPUT);
pinMode (Led, OUTPUT);
Serial.begin(9600);
setTime(11,0,0,13,07,2020);
// delay(3000);
//Comprobem si disposem del RTC connectat
if (! rtc.begin()) {
Serial.println("No hi ha un modul RTC");
while (1);
}

}

void loop() {
DateTime now = rtc.now();
for (indice=0; digitalRead (Pulsador) == 1; indice+1){
    inicio = 1 + inicio;
    digitalWrite(Led, HIGH);
    Serial.print("Hola");
    //Serial.println(inicio);
    if (inicio > 5000){
        inicio = 2;
    }
}
delay(2);
}
for (indice2=0; digitalRead (Pulsador) == 0; indice2+0){
    digitalWrite(Led, LOW);
    if (inicio > 1){
        contar = contar + 1;

        inicio = 0;
    }
    else {}
    if (now.second() == 10){
        if (suma == 1){
            //Serial.println("apunt d'obrir");
            logFile = SD.open("AQUA.txt", FILE_WRITE);
            //Serial.println("obert");
            if(logFile){
                //Serial.println("si_LogFile");
            }
            logFile.print(now.day());
            logFile.print('/');
            logFile.print(now.month());
            logFile.print('/');
            logFile.print(now.year());
```



```
    logFile.print(" ");
    logFile.print(now.hour());
    logFile.print(':');
    logFile.print(now.minute());
    logFile.print(':');
    logFile.print(now.second());

    logFile.print(" ");
    logFile.print (contar);
    logFile.println(" litres.");
    // Serial.println("apunt de tancar");
    logFile.close();
    //Serial.println("tancat");
    // suma = 0;
    // contar = 0;
    }

/* Serial.print(now.day());
Serial.print('/');
Serial.print(now.month());
Serial.print('/');
Serial.print(now.year());
Serial.print(" ");
Serial.print(now.hour());
Serial.print(':');
Serial.print(now.minute());
Serial.print(':');
Serial.print(now.second());

Serial.print(" --> ");
Serial.print (contar);
Serial.println(" litres.");*/
suma = 0;
contar = 0;
break;
} //taquem suma//
else{}
break;
}
else{
    if(suma == 0){
        suma = 1;
        break;
    }
}
```




```
        break;
    }
    break;
}
if (Serial.available()){
    valor = Serial.read();

    if (valor == 'B'){
        logFile = SD.open("AQUA.txt");    // obre els arxius de
prova.txt
        if (logFile) {
            // Serial.println("Contingut de prova:"); // text en el monitor
sèrie
            while (logFile.available()) {    // mentres existeixi contingut
dins l'arxiu
                Serial.write(logFile.read());    // llegeix un caràcter per
vegada
            }
            logFile.close();    // tanca el arxiu
        } else {
            Serial.println("error al obrir la prova.txt");// text per si
l'arxiu no es pot obrir correctament
        }
    }
}
}
```



Annex 4: Fotografies:

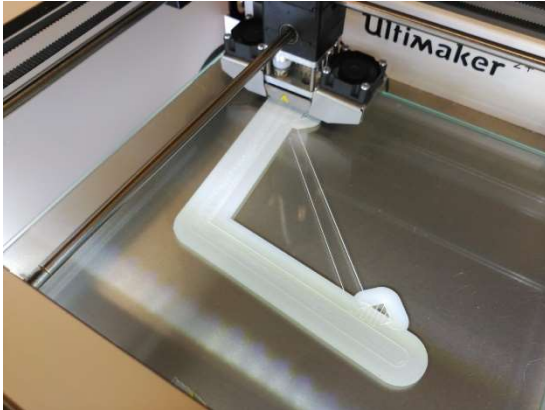


Figura 24: Procés d'impressió de la teulada

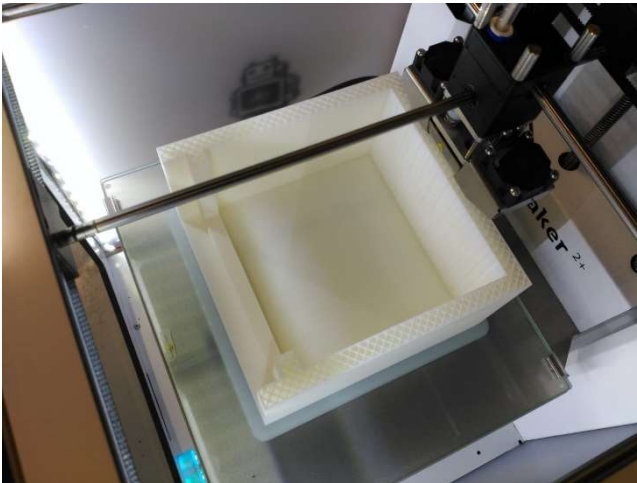


Figura 25: Procés d'impressió de la caixa



Figura 26: Porta acabada d'imprimir



Figura 27: Procés d'acoblament de totes les peces

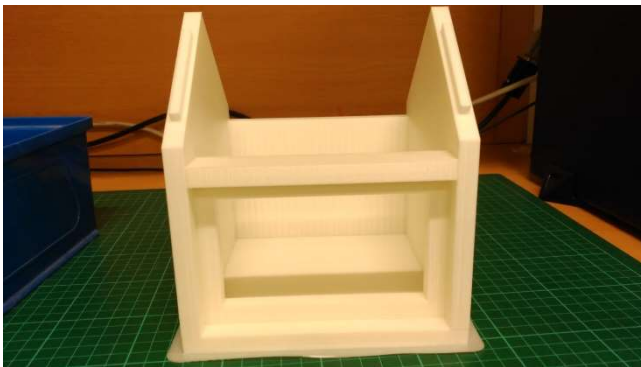


Figura 28: Caixa i biga



Figura 29: La caixa acabada

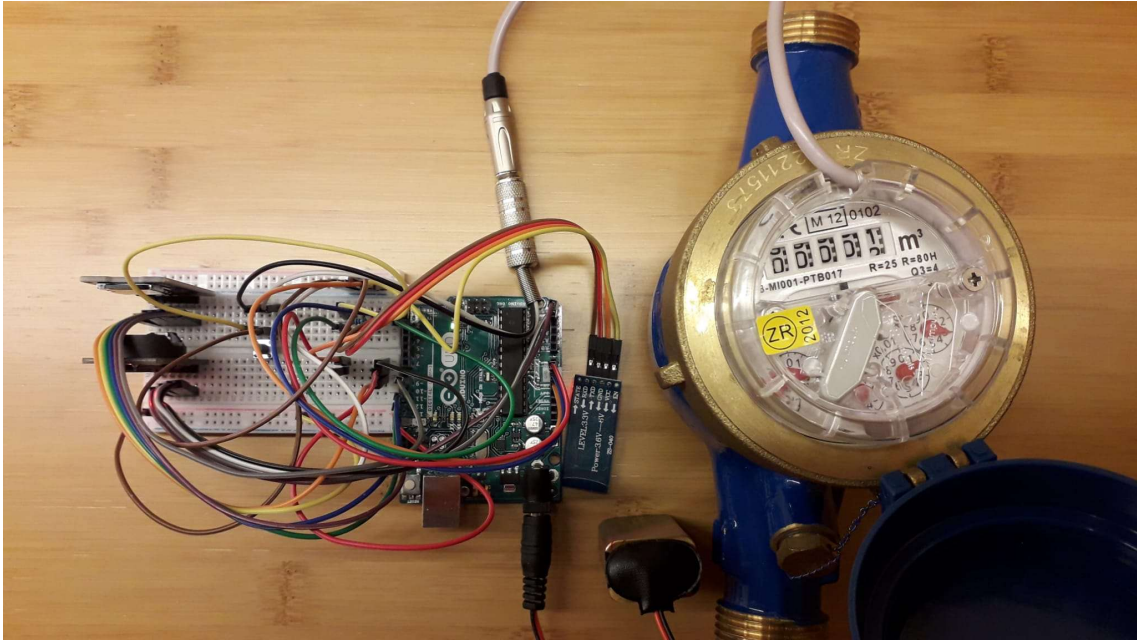


Figura 30: El circuit connectat amb el sensor