Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Xarxa de sensors d'àudio per a la detecció d'accidents urbans

Document: 1. Memòria

Alumne: Daniel Martí Vergé

Tutor: Carles Pous Sabadi
Departament: Enginyeria Elèctrica, Electrònica i Automàtica
Àrea: Enginyeria de sistemes i automàtica

Convocatòria (mes/any) : setembre/2016

ÍNDEX

1. INTRODUCCIÓ

Una xarxa de sensors sense fils permet obtenir de manera remota les dades físiques (temperatura, àudio, irradiació...) en una determinada zona d'interès. En l'actual marc de desplegament de les Smart Cities s'utilitza una gran varietat de sensors amb l'objectiu de millorar la qualitat de vida a les zones urbanes. Aquests dispositius han de tenir un baix consum per tal de funcionar varis mesos de manera autònoma, i alhora disposar de suficients recursos per a l'adquisició de dades i processament de senyals.

Les nostres ciutats es caracteritzen per una densitat elevada de persones i una necessitat d'aquestes persones de desplaçar-se fins a altres punts de la ciutat. Aquest factors resulten en una gran quantitat de moviment de vehicles i conseqüentment es produeixen accidents de trànsit. Existeixen càmeres en certs punts de les ciutats amb les quals s'analitza el trànsit i possibles accidents, però el cost d'instal·lació i manteniment d'una xarxa d'aquest tipus és elevat. Un altre paràmetre amb el qual es pot obtenir anàlisi del trànsit i de possibles accidents és a través del so.

El present projecte explora les característiques del so d'un determinat accident de trànsit i també la metodologia i els dispositius electrònics per captar i identificar si és un accident de trànsit. Es realitza una anàlisi amb Matlab per tal d'extreure primerament les principals característiques del so d'un accident i posteriorment un patró que permeti identificar la similitud amb altres sons. Seguidament s'analitzen els recursos del hardware utilitzat i s'optimitza l'algorisme de captació i comparació de sons.

1.1 Antecedents

Existeixen varis projectes de recerca en l'àmbit de les Smart Cities a la UdG. Els més importants són el ACCUS (Adaptative Cooperative Control in Urban Systems) i el MESC (Plataform for Monitoring and assessing the Efficiency of Distribution System in Smart Cities), que despleguen una plataforma per monitoritzar xarxes de sensors per captar condicions climàtiques, consum d'energia en els edificis, accés de les persones als edificis, entre d'altres.

Figura 1. Logotip del projecte ACCUS

El propi projecte Accus, amb diferents col·laboradors a nivell internacional, té una línia d'aplicació en la monitorització del tràfic urbà mitjançant càmeres, radars microones, etc. Ara bé, la detecció d'accidents de trànsit a través de xarxes de sensors, és un concepte que no s'ha aprofundit en cap projecte ni línia de recerca a la UdG.

A nivell internacional, apareixen diversos estudis que proposen l'anàlisi del so per tal de detectar la velocitat i densitat de trànsit, com el de Prashant Borkar i L.G.Malik en l'article "Acoustic Signal based Traffic Density State Estimation using SVM". Paral·lelament, el projecte SmartSantander, prova pilot dins del programa EAR-IT, neix amb l'objectiu de millorar la seguretat a les ciutats a través de l'anàlisi del so. Disposa d'una xarxa de sensors d'àudio que monitoritza situacions com la densitat de vehicles, accidents amb vehicles i la distribució del soroll a la ciutat. Segons s'indica en el seu portal web, aquesta anàlisi es fa mitjançant processament d'àudio i nivell de dB, i està en fase de disseny.

Un mètode eficaç per a la detecció d'accidents de trànsit, fet servir en alguns països d'Amèrica, consisteix en el processament i anàlisi de les imatges provinents de les càmeres de vigilància instal·lades a les ciutats. Ara bé, aquest sistema té un alt cost de desplegament i manteniment.

1.2 Objecte

La finalitat del present projecte és desenvolupar una xarxa de sensors de baix cost que permetrà fer una prova pilot amb els elements essencials, un node sensor i un coordinador, encara que a nivell general la xarxa podria estar constituïda per diversos nodes sensors. La funcionalitat d'aquesta xarxa és analitzar el so en les ciutats per tal de detectar accidents de trànsit i transmetre la informació a un servidor.

1.3 Especificacions i abast

S'integrarà un sensor d'àudio en un microcontrolador de la família Atmega per tal d'adquirir i processar el so, i per a la transmissió de la informació s'usarà un mòdul ràdio NRF24 en els nodes sensors i un mòdul WiFi ESP8266 al coordinador. S'analitzaran mètodes de detecció de so per a un tipus concret d'accident, el que està composat inicialment per una frenada i posteriorment per un xoc amb trencament de vidres, i únicament amb un únic vehicle implicat. Les proves i simulacions es realitzaran en un espai interior i reproduint sons enregistrats d'accidents.

Es desplegarà una xarxa bàsica composada per un node sensor i un coordinador. El programa del node sensor adquirirà el so a una freqüència de mostreig de 40.000Hz, executarà un algorisme per a determinar l'espectre de freqüències del so adquirit i si cal enviarà informació al dispositiu coordinador. Per altre banda, el programa del coordinador enviarà informació en un servidor quan es detecti un avís d'accident. El nodes sensor i el coordinador s'alimentaran d'una bateria recarregable per energia solar amb una autonomia mínima d'1 any.

2. ORIGEN DEL SO I SENSORITZACIÓ

Les zones urbanes de les ciutats es caracteritzen per un gran densitat de persones i vehicles, conseqüentment s'origina una gran varietat de sons. Per tal de distingir el so d'un accident d'entre tots els sons, són necessaris varis dispositius electrònics i algorismes de processament d'àudio. Un mètode comú d'anàlisi del so es realitza a través de l'espectre freqüències del senyal, que s'obté després de digitalitzar el so analògic captat per el micròfon i transformar aquest senyal del domini temporal al freqüencial.

2.1 Accidents de trànsit

A Espanya durant l'any 2013 es van produir, segons dades d'un estudi elaborat per l'Observatorio Nacional de Seguridad Vial a Espanya, 52.222 accidents de tràfic urbans amb víctimes (de les quals 450 van morir). El 15% d'aquests accidents urbans es van produir en horari de nit.

Quan es produeix un accident, la rapidesa en arribar els serveis d'urgències al lloc de l'accident és vital per a les víctimes que estan en un estat molt greu. Actualment, l'avís als serveis d'emergència el produeix una persona o vianant que està a la via. Segons la conducta general PAS que marca les pautes a seguir en cas de presenciar un accident, el primer pas és senyalitzar la zona per evitar possibles accidents, seguidament es realitza l'avís als serveis d'emergència i finalment es procedeix a auxiliar a la víctima.

Així doncs, en el cas més desfavorable en què no hi hagi ningú presenciant l'accident, el temps que pot tardar a donar-se l'avís és el temps en què apareix una persona a davant l'accident i el temps que es triga en senyalitzar la via amb triangles d'emergència.

Els vehicles que poden intervenir en un accident són cotxes, camions, motocicles i vianants. Les col·lisions simples són els accidents en què intervé solament un vehicle i aquest o bé xoca contra una paret o obstacle (mur o fanal) o bé, bolca el vehicle i fricciona amb el terra. Les col·lisions múltiples impliquen el xoc entre dos o més vehicles. Finalment, també pot esdevenir-se una col·lisió entre un vehicle i un vianant.

El present projecte es focalitza en l'anàlisi d'accidents de cotxes i camions i amb col·lisions simples. Si el cotxe o camió topa frontalment o posteriorment, el tipus de materials que

generalment impacten són els para-xocs de plàstic, una part del bastidor d'acer, la xapa d'acer, el policarbonat dels fars i el vidre de les bombetes.

També pot xocar lateralment el vehicle, impactant les portes d'acer i el vidre de les finestres. Uns materials que poden intervenir també en els accidents en el moment que es frena bruscament són l'acer dels discos de frens i el cautxú dels neumàtics. Els sons importants a analitzar en l'instant que es produeix un accident són el de la frenada i el del xoc, ja que gran part dels accidents tenen aquestes dues particularitats.

2.2 Sons enregistrats d'accidents de trànsit

Per tal de realitzar l'anàlisi de característiques del so d'un accident i per a la simulació i comprovació de l'algorisme de detecció de sons, durant el projecte s'han utilitzat àudios enregistrats que contenen el so de diferents accidents trànsit. Aquests arxius d'àudio s'han descarregat del portal de vídeos i música Youtube. La taula següent recull la informació d'aquests arxius d'àudio d'accidents de trànsit, i un altre arxiu d'àudio que no és d'accident de trànsit, que serviran per a l'anàlisi de freqüència i la simulació.

Nom arxiu	Tipus arxiu	Freqüència mostreig (Hz)	Nombre de punts	Duració (ms)	Tipus de so
Àudio Accident	.wav	44.100	154.369	3.500	Accident de trànsit
Àudio 2 accident	.wav	44.100	137.135	3.110	Accident de trànsit
Àudio 3 accident	.wav	44.100	164.736	3.736	Accident de trànsit
Àudio música	.wav	44.100	927.360	21.029	Cançó "Myth"

Taula 1. Arxius d'àudio utilitzats en el projecte

Els 3 sons d'accidents de trànsit seleccionats es caracteritzen per una fase de frenada i una fase de xoc i trencament de vidres. Els altres 2 arxius corresponen a una cançó de música i al soroll d'una conversa humana.

2.3 Representació general del so d'un accident de trànsit

El fenomen físic del so s'origina degut a la vibració d'una superfície solida o alteracions d'un fluid i es propaga en forma d'ones longitudinals mecàniques a través d'un medi material. Les

ones longitudinals es mouen a través del medi mitjançant la compressió (alta densitat) i refracció (baixa densitat) de molècules en una determinada superfície del medi.

La quantitat de vegades que vibra l'aire que transmet el so és el paràmetre conegut com a freqüència i s'expressa en Hz (vibracions en un segon). En el present projecte s'analitzen les ones que es propaguen per l'aire i amb una velocitat de propagació de 343 m/s (20ºC, 50% humitat i a nivell del mar). Les ones que formen el so són un tipus de moviment oscil·latori i tenen una periodicitat i una amplitud.

En el següent gràfic es representa l'evolució temporal del so d'un accident captat per un micròfon. L'arxiu d'àudio que s'ha utilitzat és l'Àudio Accident i la seva duració és de 3,5 segons.

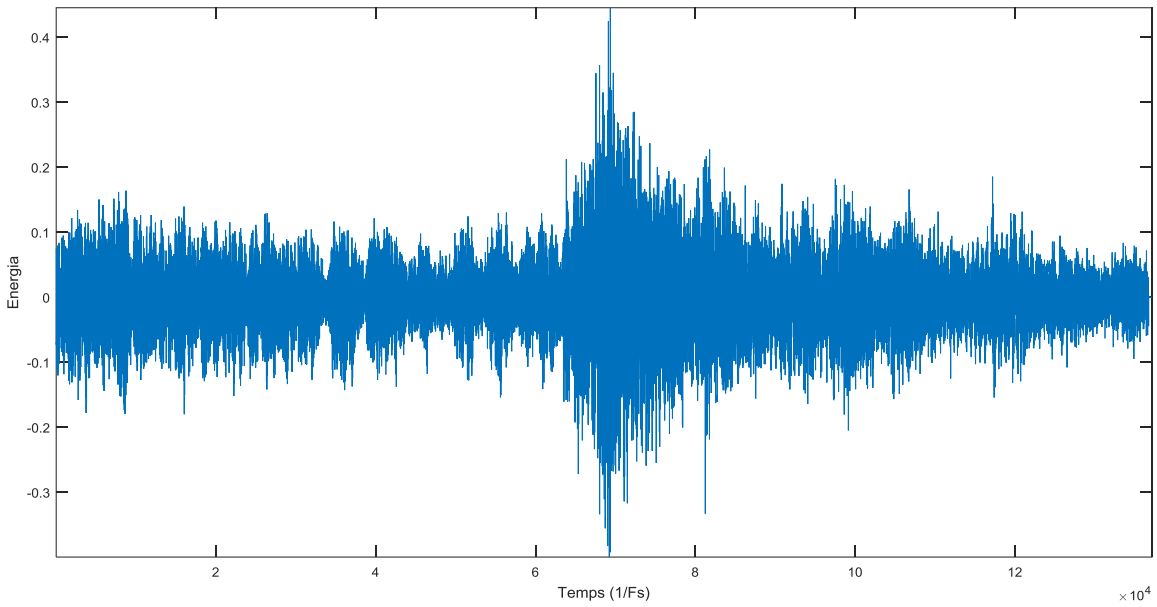

Figura 2. Representació temporal de l'Àudio Accident

Es pot observar que l'ona varia d'amplitud al llarg del temps i que es repeteix periòdicament a una determinada freqüència. Aquesta freqüència depèn del tipus de so que hi ha a cada instant de temps. Quan el micròfon capta la ona, aquesta es pot representar en el domini temporal. Per obtenir el domini freqüencial de l'ona s'ha de realitzar un algorisme de càlcul, com per exemple la transformada de Fourier, que descompon el senyal amb una suma de corbes sinusoïdals.

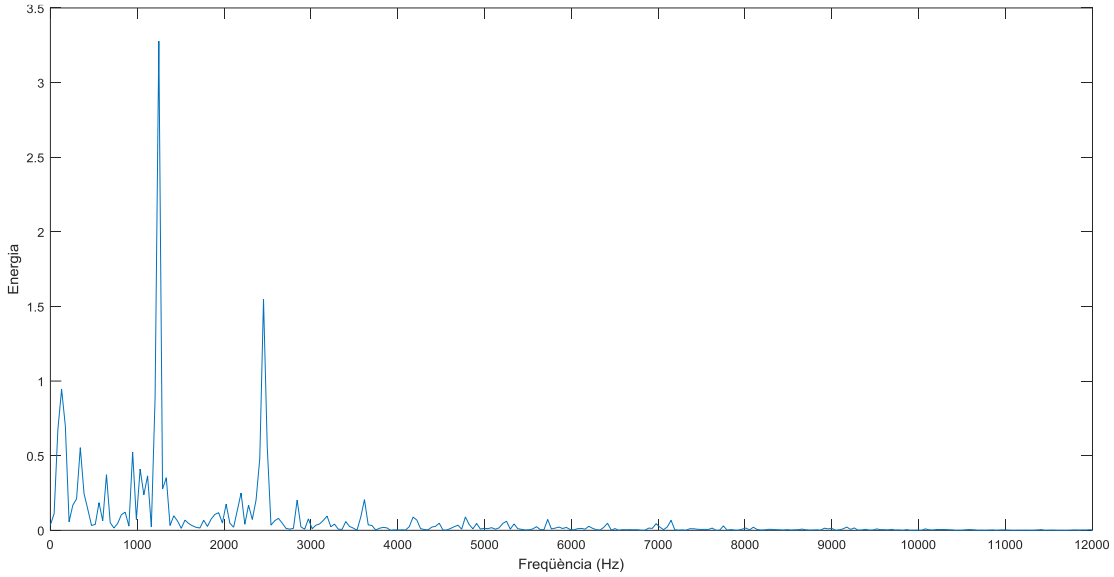

Figura 3. Representació freqüencial d'un fragment de l'Àudio Accident

Es important esbrinar l'espectre de freqüències del soroll que capta el micròfon per tal de comparar-ho amb l'espectre que es produeix en el so d'un accident. El gràfic anterior s'hi pot observar la distribució de freqüències, ara bé, només és d'un petit fragment de la totalitat de l'Àudio Accident. Per a la representació freqüencial existeixen diversos mètodes, en el present projecte s'utilitzen la transformada de Fourier i la transformada de Hartley.

2.4 Adquisició del so

Els sons es poden digitalitzar a través d'un sensor que transforma el canvi de pressió de les ones en energia elèctrica. Aquests sensors són els micròfons. La digitalització es realitza a través d'un convertidor analògic a digital (ADC), que guarda en format digital el senyal del sensor a una freqüència de mostreig determinada.

2.4.1 Tipus de micròfons

Existeixen varis tipus de sensors d'àudio i es poden classificar segons el transductor que utilitzen, la seva directivitat, resposta en freqüència, etc. El tipus de micròfon a escollir depèn de l'ús que se'n farà.

Nom del micròfon	Material transductor	Alimentació	Ample banda (Hz)	Resposta freqüència	Direccionalitat	Sensibilitat	Distorsió externa	Abast
Condensador	Plaques condensador	Externa	20-20.000	Plana	Variable	Alta	Calor, humitat	Mig
Electret	Plaques condensador	Carrega electrostàtica	20-20.000	Regular, accentua aguts	Omnidireccional, bidireccional	Alta	Pols	Baix
Carbó	Carbó	Externa	100-4.000	Irregular	Direccional	Mitja		Baix
Bobina	Iman, bobina	Camp magnètic	40-16.000	Irregular	Omnidireccional, cardioide	Mitja		Baix
Cinta	Iman, cinta metall	Camp magnètic	40-14.000	Plana	Bi-direccional, direccional	Mitja	Vent	Baix
Cristall	Cristall quartz	Deformació piezoelèctrica	100-8.000	Irregular	Omnidireccional	Alta	Calor, humitat	Mig
Ceràmic	Titanita de bario	Deformació piezoelèctrica	100-8.000	Irregular	Omnidireccional	Mig	Calor	Alt

Taula 2. Tipologia de micròfons

La taula anterior resumeix les principals característiques dels micròfons més comercialitzats. La direccionalitat dels micròfons és la distribució de sensibilitat del so provinent de diferents direccions. Els micròfons omnidireccionals presenten una sensibilitat uniforme entorn als seus 360 graus, és a dir, permeten captar so provinent de qualsevol direcció. Referent als micròfons bidireccionals, la seva sensibilitat és present únicament a la part davantera i posterior. Per altre banda, els unidireccionals són micròfons que capten bé el so que els prové del davant i en menor sensibilitat al so que prové de la part posterior.

2.4.2 Conversió analògica a digital

El so es caracteritza per ser un senyal que varia contínuament al llarg del temps. Per al seu enregistrament és necessari un ADC, que mostreja el senyal a una determinada freqüència.

La resolució de l'ADC ve determinada per el nombre de bits en què s'enregistra el senyal analògic. El nombre de bits significa el número de divisions en què es compararà el voltatge d'entrada que detecta l'ADC. Amb més bits resulta en més divisions a comparar el voltatge d'entrada, i per tant significa que si l'entrada varia constantment es detectaran més diferencies entre els valors enregistrats. Més bits també suposa que el valor enregistrat canviarà per una variació més petita del senyal d'entrada. Un altre factor que permet major resolució és aprofitar el rang d'entrada de l'ADC. El voltatge que proporcionen els micròfons sol ser baix, de l'ordre de pocs mV. Una etapa de pre-amplificació serà necessària per tal d'augmentar l'amplitud del senyal i aprofitar millor la resolució de l'ADC.

Seguidament s'adjunta una taula que resumeix la resolució que es pot aconseguir si s'adapta el senyal a tot el rang d'entrada de l'ADC. El nombre de bits dels ADC dels microcontroladors més comuns són de 10 i 12 bits. Els voltatges màxims d'entrada dels ADC més comuns són de 3,3V i de 5V.

Número de bits	Nombre de divisions	Voltatge màxim (V)	Voltatge mínim (V)	Resolució (mV)
10	1.024	5,00	0,00	4,89
10	1.024	3,30	0,00	3,23
12	4.096	5,00	0,00	1,22
12	4.096	3,30	0,00	0,81

Taula 3. Resolució segons tipus d'ADC

La resolució s'ha calculat mitjançant la següent fórmula:

$$r = \frac{V_{max}}{2^n - 1} \quad \text{(Eq. 1)}$$

on,

r = resolució, en mV partit per bit

V_{max} = voltatge màxim d'entrada al ADC, en mV

n = nombre de bits de l'ADC

Podem observar que per un ADC de 10 bits i amb un rang d'entrada analògic de 0 a 3,3V, la resolució serà de 3,23mV. La qualitat del so enregistrat variarà degut a aquesta resolució i també degut a l'amplificació que pugui proporcionar l'etapa de pre-amplificació.

3. ANÀLISI FREQUENCIAL DEL SO D'UN ACCIDENT

Per tal d'extreure un patró del so d'un accident és necessari analitzar el domini temporal i freqüencial del senyal. Les següents tècniques d'anàlisi s'han processat amb el software Matlab. A partir de l'espectre de freqüències es pot observar les components freqüencials i la densitat d'energia. L'espectrograma és útil per identificar les fases de l'accident i extreure les seves freqüències fonamentals. Altres mètodes com el Cross Power Spectrum o el Magnitude Squared Coherence proporcionen un grau de similitud freqüencial entre dos senyals. El mètode de relació de distàncies és adequat per a implementar en un microcontrolador degut a la seva simplicitat.

El tipus de so d'accident que s'analitza és el que està format per una fase de frenada i una de xoc i trencament de vidres. Degut a la baixa capacitat de memòria de dades del microcontrolador Atmega328 s'ha hagut de simplificar el patró que compara les freqüències espectrals de l'Àudio Accident amb les del so que capta el micròfon. Partint d'un espectrograma que representa tota la distribució de freqüències al llarg de la duració del so, s'han seleccionat dos punts importants que serviran per a aplicar els mètodes de similitud.

3.1 Representació freqüencial

El senyal acústic es pot representar com a suma de corbes sinusoïdals de diferents freqüències. Aquest pas és la transformada de Fourier, que permet descompondre el senyal en les seves components freqüencials. En els següents apartats s'analitza el senyal del so d'un accident de trànsit a partir de l'espectre freqüencial.

3.1.1 Transformada de Fourier

La transformada discreta de Fourier (DFT) consisteix en agafar un nombre N de punts del senyal (finestra temporal) i realitzar el sumatori del valor del senyal multiplicat per a la seva component sinusoïdal (expressat en exponencial) per a cada punt:

$$X(k) = \sum_{n=0}^{n=N-1} X(n) \cdot e^{-i \cdot k \cdot n \cdot (\frac{2\pi}{N})} \qquad \text{(Eq. 2)}$$

$$X(k) = \sum_{n=0}^{n=N-1} X(n) \cdot \left(\cos\left(\frac{k \cdot n \cdot 2\pi}{N}\right) - \mathrm{i} \cdot \sin\left(\frac{k \cdot n \cdot 2\pi}{N}\right)\right) \quad \text{(Eq. 3)}$$

on,

X = energia de la component freqüencial k en component real i complexa

N = nombre de punts de la finestra temporal

n = enèsim punt de la finestra

k = índex de la component freqüencial resultant, de 0 fins a N-1

El nombre d'operacions necessari per a realitzar la DFT és de l'ordre de N·N. La magnitud es calcula fent el mòdul del valor real i valor complexa:

$$Magnitud = |X(k)| = \sqrt{X(k,real)^2 + X(k,compl)^2} \quad \text{(Eq. 4)}$$

on,

Magnitud = mòdul de la component freqüencial X(k)

Amb el Matlab, existeix la funció FFT que realitza la transformada rapida discreta de Fourier. La FFT és una versió més eficient de la DFT, on el nombre d'operacions es redueix a N·log(N), i on també s'obtenen les magnituds de cada component freqüencial. Es pot representar un gràfic on a l'eix X estan ubicades N/2 valor de freqüències, i a l'eix Y hi ha la l'amplitud d'energia de cada freqüència. La amplitud d'energia s'obté de la forma següent:

$$Amplitud\ d'energia = \frac{[FFT(A)]^2}{N} \quad \text{(Eq. 5)}$$

on,

FFT(A) = correspon a la transformada rapida de Fourier del senyal A

N = nombre de punts agafat per a la transformada FFT

Cal remarcar que per una freqüència de mostreig Fs, el rang de freqüències que es pot representar és de Fs/2. Si les finestres temporals són de N punts, la dimensió que tindrà l'espectre de freqüències serà de N/2 punts. La distància entre cada component freqüencial s'obté a partir de Fs/N, o bé, (Fs/2)/(N/2). En l'anàlisi amb Matlab s'utilitzaran finestres temporals de 256 punts per a calcular la FFT, resultant cada finestra temporal en vectors de 128 components freqüencials del senyal. La decisió d'agafar les finestres temporals amb 256 punts és amb l'objectiu de traslladar correctament les comparacions al microcontrolador Atmega328, el qual la llibreria per fer el càlcul de l'espectre de freqüències fa servir un màxim de 256 punts.

3.1.2 Transformada ràpida de Hartley

Un altre mètode per a calcular les components freqüencials d'una finestra temporal és la transformada ràpida de Hartley (FHT). Aquest algorisme treballa en nombre reals (a diferència de la FFT que treballa amb reals i complexos) i per tant el nombre de dades usades i operacions és casi la meitat. La fórmula de la DHT és idèntica a la de la DFT, però sense valors complexos:

$$Y(k) = \sum_{n=0}^{n=N-1} Y(n) \cdot \left(\cos\left(\frac{k \cdot n \cdot 2\pi}{N}\right) - \sin\left(\frac{k \cdot n \cdot 2\pi}{N}\right)\right) \qquad \text{(Eq. 6)}$$

on,

Y = energia de la component freqüencial k, en valors reals

N, n i k prenen el mateix significat que en la transformada de Fourier.

3.1.3 Característiques freqüencials i temporals

Per tal de visualitzar el rang de freqüències que caracteritza al so d'un accident de trànsit es procedeix a calcular amb Matlab la FFT de tot el senyal a partir de finestres temporals de 256 punts. La FFT del senyal retornarà un vector de 128 components freqüencials reals

(l'anomenarem FFT_A) i per tal de visualitzar tots els vectors es superposaran tots ells en un mateix gràfic.

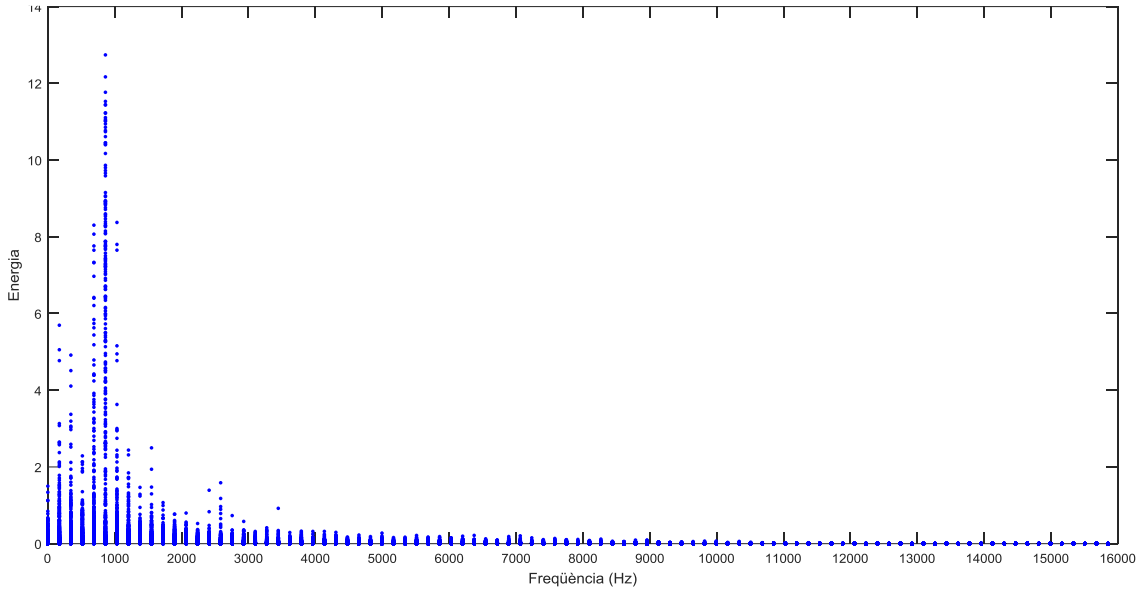

Figura 4. Superposició dels espectres de freqüències de l'Àudio Accident

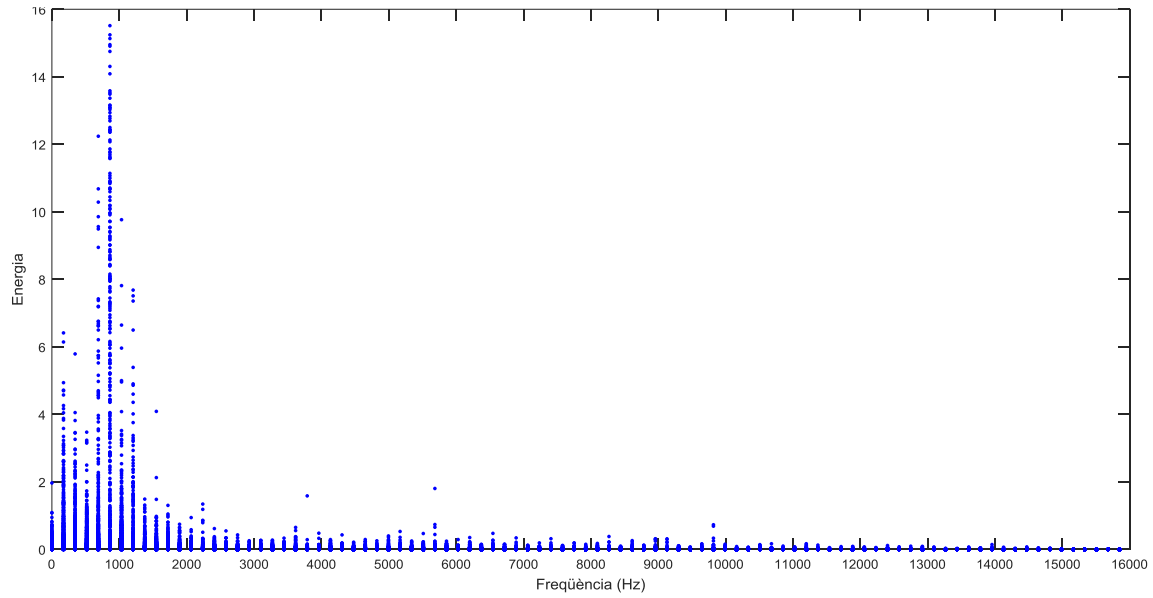

Figura 5. Superposició dels espectres de freqüències de l'Àudio 2 accident

Els gràfics anteriors representen la distribució espectral de dos sons d'accident, que corresponen als arxius Àudio Accident i Àudio 2 accident. La freqüència de mostreig d'aquests arxius d'àudio és de 44.100Hz, aleshores, l'espectre representat amb Matlab estarà format per 128 punts i la distància entre cada punt serà de 172,26Hz (44.100Hz/256).

Es pot observar en els dos senyals que el rang de freqüències que caracteritza als sons d'accidents de tràfic és de 100Hz a 10.000Hz. També s'observa en els gràfics anteriors una gran densitat de freqüències en el rang de 400-1.500Hz, i llavors varis harmònics al voltant de 2.000Hz, 3.000Hz i 5.000Hz.

En la representació espectral dels sons d'accidents realitzada anteriorment s'hi observa la densitat total d'energia freqüencial, però no es distingeix en quin moment apareix cada component freqüencial. A continuació hi ha representat l'espectrograma de l'arxiu Àudio Accident.

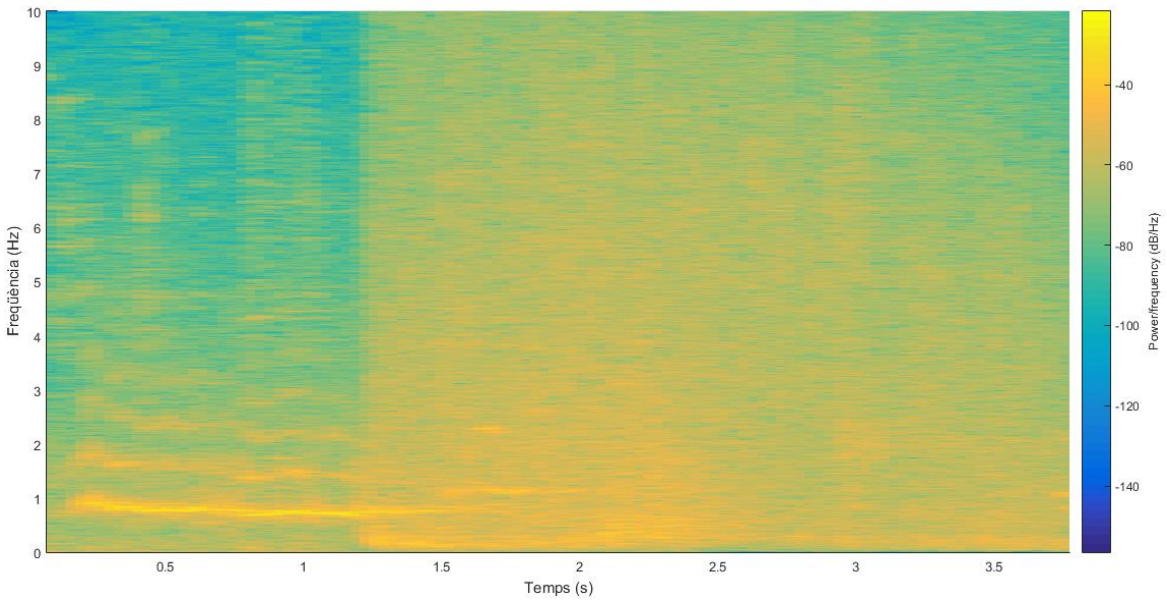

Figura 6. Espectrograma de l'Àudio Accident

L'espectrograma representa en un gràfic les 3 variables freqüència, energia i temps. Aquest tipus de gràfics son útils ja que permeten visualitzar l'aparició de les components freqüencials al llarg de la duració del so, així com la seva amplitud. L'amplitud d'energia es representa mitjançant una escala de colors, de color taronja per als valors més grans d'amplitud i de color blau fosc pels de menys amplitud.

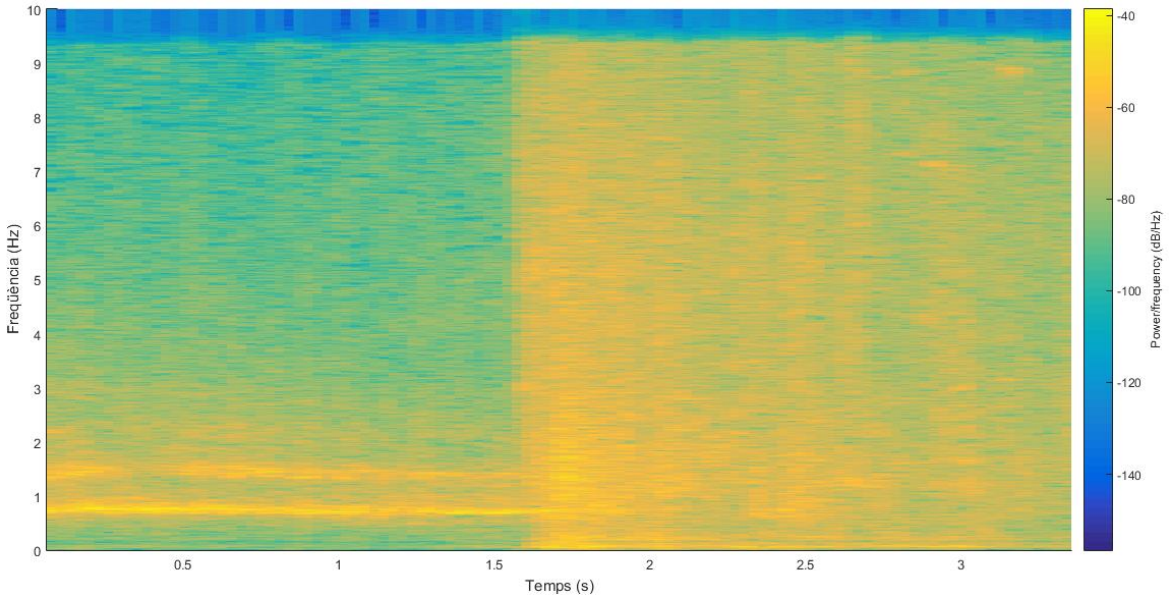

Figura 7. Espectrograma de l'Àudio 2 accident

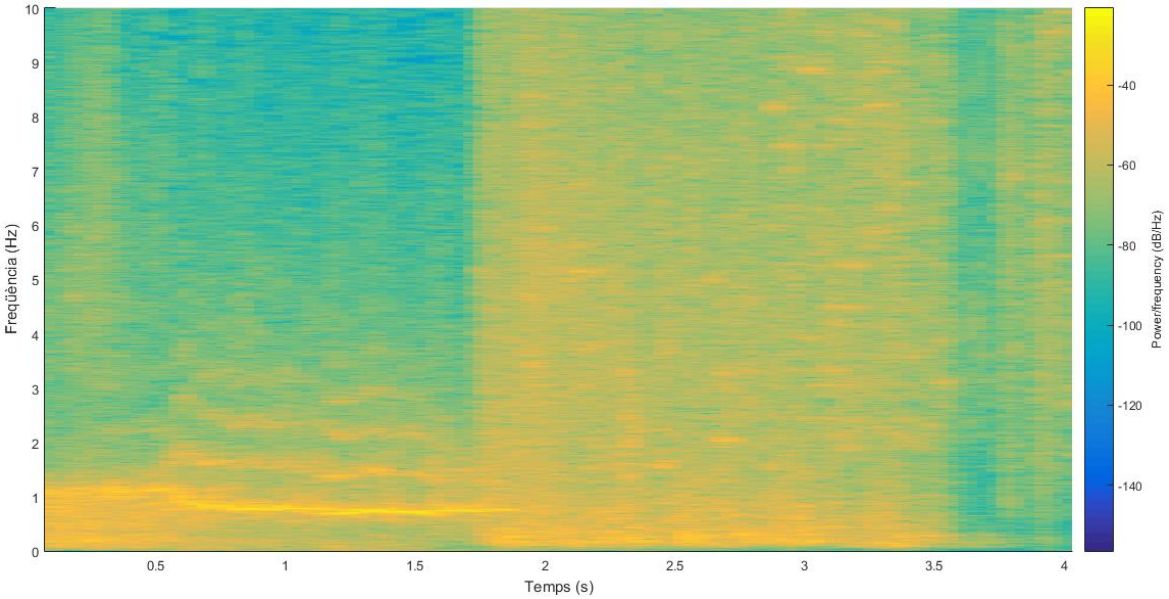

Figura 8. Espectrograma de l'Àudio 3 accident

Els tres espectrogrames anteriors tenen una evolució freqüencial similar. Analitzant el primer gràfic, es pot observar que en un primer tram, des de l'inici fins aproximadament a 1,2s, l'energia freqüencial està repartida principalment en 3 franges: aproximadament a 900Hz, a 1.700Hz i a 2.500Hz. Continuant en el primer gràfic, a partir de 1,2s les components freqüencials es distribueixen més uniformement en tot l'espectre però amb poca energia, i a l'instant 2s apareix una alta densitat d'energia al rang de 0-2.000Hz.

Fase	Duració (ms)	Causa	Freqüència predominant	Harmònics
Frenada	1.000	Fre bloquejat	700-900Hz	1.700Hz, 2.500Hz
Xoc	500	Col·lisió	100-1.000Hz	2.000-10.000Hz

Taula 4. Fases dels sons d'accidents de trànsit

A partir dels espectrogrames dels 3 àudios d'accidents es poden distingir 2 fases principals. La fase de frenada és la que apareix quan el conductor del vehicle s'adona d'un perill imminent i decideix frenar per evitar-ho. Al prémer bruscament el fre es bloqueja la roda i produeix un so que es caracteritza per 3 o 4 bandes de entre 0 i 4.000Hz que es prolonguen durant uns més de 1.000ms. La segona fase es produeix quan el cotxe impacta amb un element de la carretera o amb un altre cotxe. Aleshores la planxa de la carrosseria es doblega i es trenquen vidres dels llums, del parabrises i/o de les finestres. Aquest so es caracteritza per una elevada densitat de freqüències entre 0 i 2.000Hz que tenen molta energia i amb duració de uns 100ms, així com una aparició de components freqüencials entre 2.000Hz i 10.000Hz però amb poca energia i amb duració de més de 500ms. És casi improbable que coexisteixin les dues fases, ja que quan es produeix el xoc el cotxe perd la totalitat de la inèrcia i per tant no hi ha fricció entre frens, neumàtics i l'asfalt.

Cal remarcar que no tots els sons d'accidents de trànsit esdevenen amb les fases de frenada i de xoc. Un accident de trànsit pot succeir només amb la fase de xoc, sense que el conductor hagi polsat el fre i per tant no aparegui la fase de frenada.

3.2 Patrons

A partir d'un espectre de freqüències és possible caracteritzar a un fragment temporal d'un senyal. Aquest espectre, que s'anomenarà patró, ha de ser de reduïdes dimensions ja que la memòria de dades del microcontrolador Arduino és reduïda.

L'arxiu d'on s'obtindran els patrons és l'Àudio Accident, degut a que en l'espectrograma realitzat en l'apartat anterior s'observa clarament les fases de frenada i xoc. El nombre de patrons que s'extreuen és de 2, que representaran a les dues fases. Per tal de fer l'extracció d'un patró es selecciona l'instant on apareix el tipus d'espectre desitjat. Per a l'anàlisi de similitud d'aquest projecte, l'espectre de freqüències del patró serà d'una mida de 128 punts que s'obté d'una finestra temporal de 256 punts.

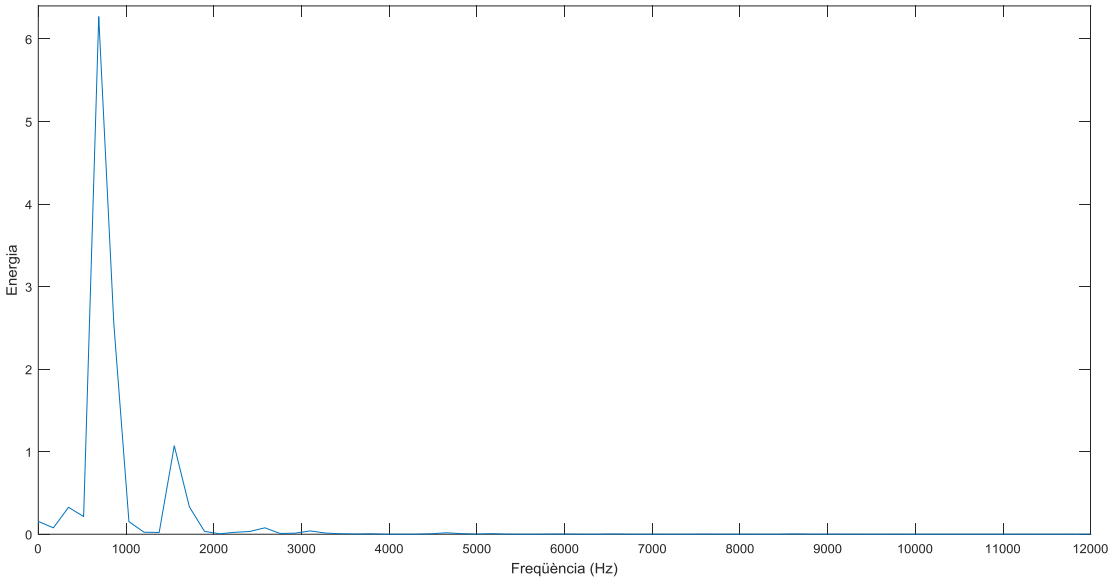

Figura 9. Espectre de freqüències del patró 1

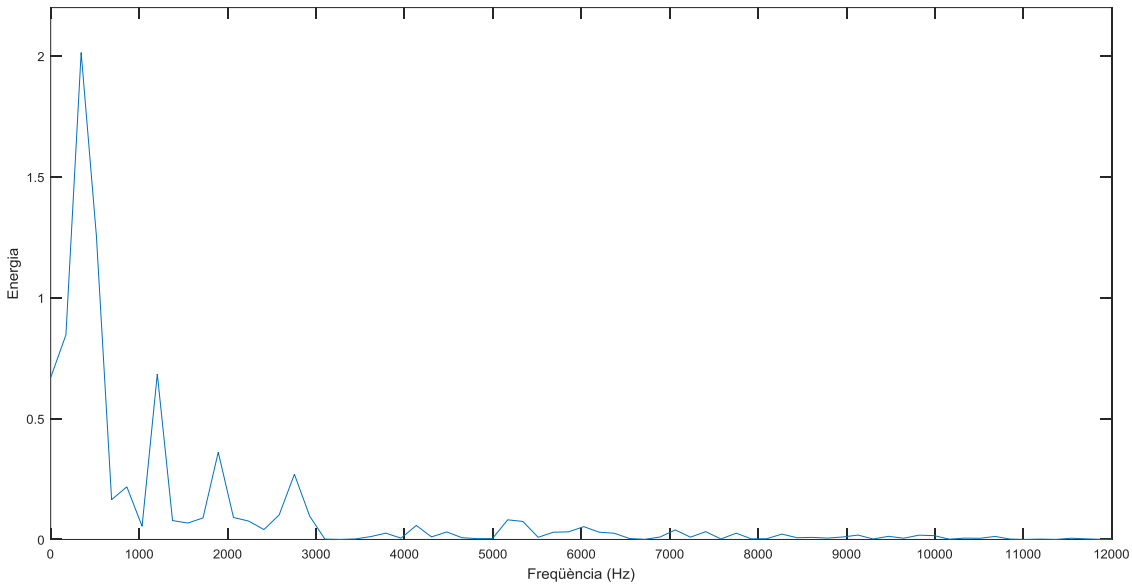

Figura 10. Espectre de freqüències del patró 2

El patró 1 s'ha seleccionat a l'instant 734ms, que pertany a la fase de frenada del senyal analitzat, i es caracteritza per 1 banda de freqüència fonamental 500-1.000Hz i 2 bandes amb menys energia a 1.400-1.900Hz i a 2.500-2.600Hz. El patró 2 l'hem ubicat dins la fase de xoc i trencament de vidres, en l'instant 1.910ms, on es produeix el trencament de vidres i es caracteritza perquè de 0 a 3.000Hz hi ha energia, té un pic fonamental entre a 100-600Hz, i 3 pics amb menys energia a 1.000-1.300Hz,1.800-2.000Hz i 2.600-3.000Hz.

Cada patró el formarà un vector de 128 valors, on cada valor significa l'energia de les diferents components freqüencials. L'espectre de freqüències representat serà d'un rang de 0Hz a 20.000Hz.

Patró	Arxiu origen	Instant senyal (ms)	Finestra temporal (N)	Nombre components	Duració (ms)	Freqüència predominant	Harmònics
patró 1	Àudio Accident	734	256	128	6,4	700Hz	1.600Hz, 2.600Hz
patró 2	Àudio Accident	1916	256	128	6,4	400Hz	1.200Hz, 1.900Hz, 2.700Hz

Taula 5. Patrons extrets dels accidents de trànsit i les seves característiques

3.3 Mètodes de similitud

Per tal d'esbrinar el grau de similitud de l'Àudio Accident amb el so que capta el micròfon s'utilitzaran les funcions de processament de senyal que té Matlab. El Cross Power Spectrum compara la densitat d'energia espectral de les dos senyals i el Magnitude Squared Coherence és una extensió del Cross Power Spectrum i proporciona un coeficient de similitud dels espectres freqüencials. Ha estat necessari afegir una variant al Magnitude Squared Coherence per tal de poder captar l'evolució temporal del grau de similitud. Amb la mateixa finalitat, també s'han dissenyat nous mètodes de similitud, com la relació de distàncies mitja ponderada.

La precisió en els resultats que proporcionen aquests mètodes ve determinada per la grandària de la finestra temporal en què es calcula la FFT (per exemple, de l'ordre de 4.096 punts). Ara bé, en el present projecte els mètodes de similitud es realitzaran amb una finestra temporal de 256 punts, per tal d'adequar l'anàlisi a la capacitat de memòria i a la llibreria de càlcul freqüencial utilitzada en el microcontrolador Atmega328. Paral·lelament, s'ha dissenyat el mètode de relació de distàncies amb l'objectiu de reduir la memòria utilitzada en un microcontrolador.

3.3.1 Cross Power Spectrum

El Cross Power Spectrum és un coeficient que determina la similitud de dos components freqüencials. Es calcula mitjançant:

$$Cross\ Power\ Spectrum = S_{AB}(k) = \frac{FFT_B(k) \times FFT_A(k)}{N} \qquad \text{(Eq. 7)}$$

on,

S_{AB} = valor d'energia obtinguda de la correlació dels dos senyals

FFT_A = transformada ràpida discreta de Fourier del senyal a analitzar

FFT_B = transformada ràpida discreta de Fourier del senyal que es vol comparar

N = número de punts que s'agafa per a la transformada ràpida de Fourier.

El resultat que retorna aquest mètode és el valor d'energia dels dos senyals per a cada component freqüencial. A més nivell d'energia més similitud existeix entre els dos senyals. Es pot usar el mètode Cross Power Spectrum per a la detecció de similitud, ara bé, el mètode Magnitude Squared Coherence dona un resultat més directe.

3.3.2 Magnitude Squared Coherence

El coeficient Magnitude Squared Coherence (MSC) relaciona la correlació que té un senyal amb un altre senyal. Cada coeficient MSC correspon a la similitud de cada component freqüencial de les dos senyals. La fórmula general del MSC és:

$$MSC(k) = \frac{\left[Magnitud\left(S_{AB}(k)\right)\right]^2}{S_{AA}(k) \times S_{BB}(k)} \quad \text{(Eq. 8)}$$

on,

MSC = coeficient de similitud de dos senyals

S_{AB} = Cross Power Spectrum de la component freqüencial del senyal A i el senyal B

S_{AA} ,S_{BB} = auto Cross Power Spectrum de cada senyal.

El càlcul d'aquest coeficient de similitud per a una finestra espectral proporciona un vector de coeficients MSC. Els coeficients són valors normalitzats entre 0 i 1, significant alta similitud els valors que superen a 0,75.

3.3.3 MSC equivalent i mig

El vector MSC és un coeficient de similitud per a una sola finestra freqüencial del senyal. Ara bé, al disposar de varies finestres dels senyals a comparar, s'obtindran varis vectors de N coeficients MSC cadascun.

S'han definit dos càlculs que permetran extreure un valor significatiu de cada vector MSC i representaran la similitud de cada finestra espectral. El primer selecciona els valors més alts del vector MSC i realitza la mitjana aritmètica d'aquests valors, obtenint l'índex MSC equivalent (MSC_{equiv}).

$$MSC_{equiv}(t) = \frac{\sum_1^{nmax} MSC(k)}{n_{max}} \quad \text{(Eq. 9)}$$

on,

MSC_{equiv} = índex de similitud obtingut de la mitjana de nmax coeficients MSC

n_{max} = nombre de coeficients MSC més alts que es busquen al vector de MSC

MSC = coeficient Magnitude Squared Coherence de dos senyals

Per exemple, per a la comparació de dues finestres freqüencials de 128 punts, si s'obtenen 128 coeficients MSC, es seleccionen els 12 valors més alts (n_{max} = 12), es sumen, i es realitza la seva mitjana aritmètica. Si l'índex MSC equivalent sobrepassa a 0,75, significa que els senyals tenen certa similitud espectral.

Pel que fa al segon càlcul, anomenat índex MSC mig (MSC_{mig}) i que també representa al vector MSC, s'ha realitzat la mitjana aritmètica de tots els valors del vector MSC. Aleshores, si el vector MSC està format per molts valors que sobrepassen a 0,75 (similitud entre dues components freqüencials) l'índex MSC mig tendirà a sobrepassar el 0,75.

$$MSC_{mig}(t) = \frac{\sum_1^k MSC(k)}{k} \quad \text{(Eq. 10)}$$

on,

MSC_{mig} = índex de similitud obtingut de la mitjana aritmètica dels MSC

MSC = coeficient de similitud Magnitude Squared Coherence de dos senyals

k = número de components de la finestra espectral d'on s'han calculat els MSC

3.3.4 Relació de distancies

A partir de la relació de distàncies (RD) es pot saber quina similitud tenen dos valors. Aquesta relació divideix la diferencia de distància dels dos valors entre la distància total dels valors. En el cas de l'espectre de freqüències, es mesurarà la distància entre la component freqüencial de dos senyals.

$$Rd(k) = \left(\frac{FFT_B(k) - FFT_A(k)}{FFT_B(k) + FFT_A(k)}\right) \quad \text{(Eq. 11)}$$

on,

Rd = coeficient de similitud de dos senyals

FFT_A = transformada ràpida discreta de Fourier del senyal a analitzar

FFT_B = transformada ràpida discreta de Fourier del senyal a comparar

Per obtenir alta similitud per a valors alts de l'índex RD és necessari normalitzar l'índex Rd:

$$RD(k) = 1 - Rd(k) \quad \text{(Eq. 12)}$$

Quan el valor de RD s'acosta a 1 existeix màxima similitud, i com més proper a 0 no hi ha cap tipus de semblança. Per valors que sobrepassen a 0,50 es considera que existeix similitud.

3.3.5 RD equivalent i mig

A partir dels índexs RD equivalent (RD_{equiv}) i RD mig (RD_{mig}) es tindrà una representació del vector de MCS, de forma similar als índexs MSC equivalent i MSC mig. L'índex RD equivalent selecciona els valors més alts del vector RD i realitza la seva mitjana aritmètica:

$$RD_{equiv}(t) = \frac{\sum_1^{n_{max}} RD(k)}{n_{max}} \quad \text{(Eq. 13)}$$

on,

RD_{equiv} = índex de similitud obtingut a partir de la mitjana de n_{max} coeficients RD

RD = coeficient de similitud de relació de distàncies de dos senyals

n_{max} = nombre de coeficients RD més alts que es busquen al vector de RD(k)

Pel que fa a l'índex RD mig, aquest calcula la mitjana aritmètica d'aquest vector del vector RD.

$$RD_{mig}(t) = \frac{\sum_1^{k} RD(k)}{k} \quad \text{(Eq. 14)}$$

on,

RD_{mig} = índex de similitud obtingut a partir de la mitjana aritmètica dels RD

Valors de l'índex RD equivalent per sobre de 0,5 indiquen similitud. La similitud del RD mig s'identifica de manera igual.

3.3.6 RD mig ponderat

S'ha introduït aquest índex per a donar pes a les components freqüencials del senyal a analitzar que coincideixen amb les components dels patrons de més energia i treure pes a les components que no hi coincideixen. Si definim a Q com el vector de components freqüencials més importants del patró, aleshores, el RD ponderat (RD_p) s'obté de la següent manera:

$$RD_p(k) = \begin{cases} 1 - \left(Rd(k)\right)^2, & si\ Rd(k) < 0.6\ i\ k\ pertany\ a\ Q \\ 1 - \sqrt{Rd(k)}, & si\ Rd(k) > 0.4\ i\ k\ no\ pertany\ a\ Q \end{cases} \quad \text{(Eq. 15)}$$

RD_p = índex de similitud ponderat de dos senyals

Q = vector de freqüències a ponderar

Rd = índex de similitud de relació de distàncies de dos senyals

Seguidament, l'índex RD mig ponderat (RD_{migP}) es calcula:

$$RD_{migP}(t) = \frac{\sum_1^k RD_p(k)}{k} \quad \text{(Eq. 16)}$$

Pel patró 1, el vector Q està definit per les freqüències: 625, 781, 938, 1.094, 1.406, 1.563, 1.719, 1.875, i que corresponen a la posició 4, 5, 6, 7, 8 del vector de valors del patró 1, respectivament. El vector Q del patró 2 està format per les freqüències: 313, 469, 625, 781, 1.094, 2.500, 2.656, 2.813, 2.969Hz que corresponen a les posicions 3, 4, 5, 7, 8, 9, 10, 11, 12, 16, 17, 18, 19 del vector del patró 2, respectivament. Existeix similitud si l'índex RD_{migP} té valors per sobre de 0,5.

3.3.7 RD mig anivellat

El senyal que arribarà al micròfon del node sensor tindrà una amplitud d'energia depenent de la distància en què s'emeti el so. Aleshores, és necessari anivellar a la mateixa amplitud el valor de cada component freqüencial del senyal amb cada component freqüencial del patró. L'anivellació es realitza al vector de freqüències del senyal deixant el patró sempre amb la mateixa amplitud. El mètode per anivellar consisteix en trobar la relació (R_{sp}) que hi ha entre el valor més alt del vector de freqüències del patró (P_m) amb el valor més alt vector de freqüències del senyal (S_m). Aleshores, es multiplica a cada valor del vector de freqüències del senyal per aquesta relació trobada per obtenir el valor anivellat:

$$R_{sp} = \frac{P_m}{S_m} \quad \text{(Eq. 17)}$$

$$FFTA_A(k) = FFT_A(k) \cdot R_{sp} \quad \text{(Eq. 18)}$$

on,

R_{sp} = relació entre el valor més del patró i el valor més alt del senyal

P_m = amplitud d'energia més alta del vector de freqüències del patró

S_m = amplitud d'energia més alta del vector de freqüències del senyal

$FFTA_A$ = transformada rapida discreta de Fourier del senyal A, amb l'amplitud anivellada

Un cop anivellades les components freqüencials del senyal es pot obtenir la relació de distàncies entre les components freqüencials anivellades (R_{dA}) del senyal i les altres components freqüencials del patró.

$$R_{dA}(k) = \left(\frac{FFTA_B(k) - FFTA_A(k)}{FFTA_B(k) + FFTA_A(k)}\right) \quad \text{(Eq. 19)}$$

on,

R_{dA} = coeficient de similitud de dos senyals anivellat

$FFTA_A$ = transformada ràpida de Fourier del senyal a analitzar, amb l'amplitud anivellada

$FFTA_B$ = transformada ràpida de Fourier anivellada del senyal a comparar, amb l'amplitud anivellada

Seguidament es pot calcular el RD mig anivellat (RD_{migA}) entre les components freqüencials anivellades del senyal i les altres components freqüencials del patró.

$$RD_{migA}(t) = \frac{\sum_1^k (1 - R_{dA}(k))}{k} \quad \text{(Eq. 20)}$$

on,

RD_{migA} = índex de similitud obtingut a partir de la mitjana aritmètica del vector de relació de distàncies anivellat

3.4 Proves de similitud

Una vegada extrets els patrons, és el moment de validar si apareix similitud entre aquests i diferents senyals. Les proves es realitzen amb Matlab comparant cada un dels patrons amb diferents senyals. Els mètodes de similitud utilitzats en aquest apartat són els índexs MSC equivalent i mig (MSC_{equiv} i MSC_{mig}), i el RD equivalent i mig ponderat (RD_{equiv} i RD_{migP}). No s'utilitza l'índex RD anivellat (RD_{migA}) ja que el senyal a comparar mantindrà l'amplitud. Les proves també serviran per verificar quin mètode de similitud s'adequa més a l'anàlisi freqüencial.

Referent als índexs MSC equivalent i RD equivalent, s'han calculat a partir d'una mitjana aritmètica dels 15 valors (n_{max}=15) dels vectors MSC i RD. Per altre banda, els índexs MSC mig i RD mig ponderat s'obtenen entre la mitjana aritmètica dels 50 primers coeficients del vector MSC i RD. Aquests 50 primers coeficients són els MSC del rang de freqüències de 0 a 7.812Hz, que es considera un rang que representa la major energia espectral dels sons d'accidents. El resultat per a cada mètode de similitud serà un gràfic temporal amb l'índex de similitud a l'eix y i la duració del senyal a l'eix x.

La comparació de similitud entre els patrons i un senyal qualsevol es realitza a partir d'una finestra lliscant de 256 punts que, a cada iteració, compara la finestra amb el patró i seguidament tornar a lliscar per sobre del senyal per a processar la següent comparació. D'aquesta manera es recórrer tot el senyal i es va comparant cada finestra lliscant amb el patró.

3.4.1 Senyals idèntics

Es comença per comparar la similitud entre els patrons i el mateix senyal d'àudio del qual s'han extret els patrons, l'Àudio Accident. Cal remarcar que quan la finestra lliscant estigui situada en l'instant d'on s'ha extret el patró, aleshores l'espectre freqüencial serà el mateix i el coeficient de similitud tindria que tenir a 1. Es representen el 4 mètodes en una sola imatge, per tal de veure més clar el comportament d'ells. Els índexs MSC estan situats a l'esquerra, i

els índexs RD a la dreta. L'índex RD mig anivellat no es fa servir degut a que els patrons estan extrets d'aquest senyal, per tant l'amplitud ja estarà anivellada.

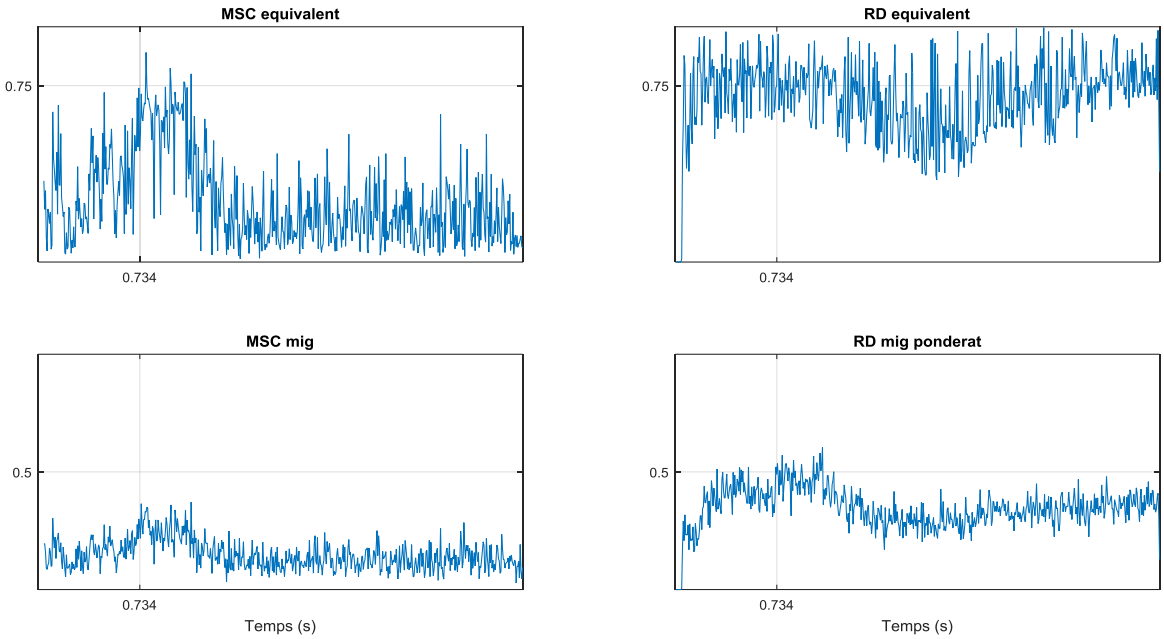

Figura 11. Anàlisi de similitud entre l'Àudio Accident i el patró 1

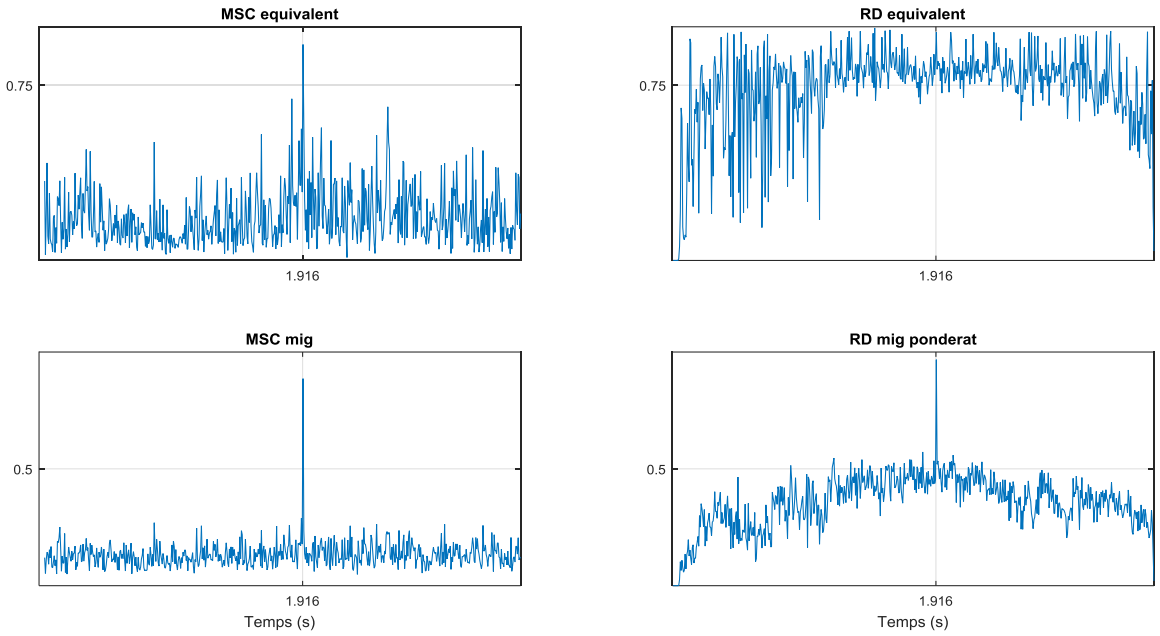

Figura 12. Anàlisi de similitud entre l'Àudio Accident i el patró 2

Podem observar que els patrons coincideixen amb el senyal justament en l'instant del qual s'han extret (0,734s i 1,916s), i per a 3 mètodes de similitud. La similitud entre els patrons i el

senyal és evident ja que aquests patrons s'han extret d'aquest mateix arxiu d'àudio. El mètode de similitud que no mostra semblança entre els patrons i el mateix senyal és el RD equivalent.

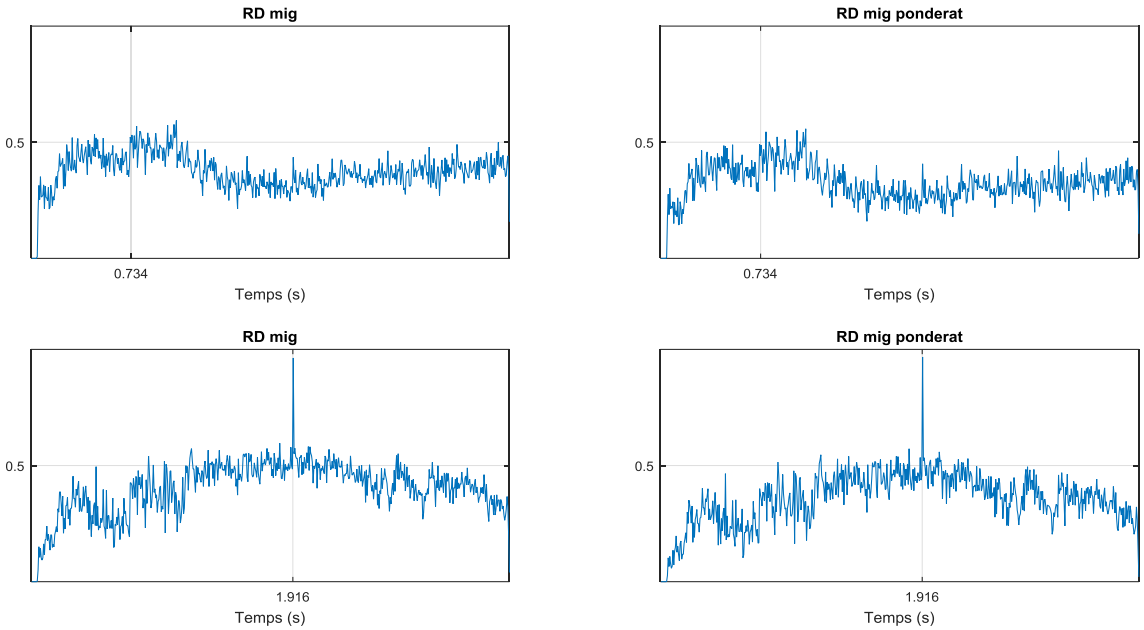

Figura 13. Comparació de l'índex RD mig i amb el RD mig ponderat pel patró 1 (dalt) i pel patró 2 (sota)

El gràfic anterior serveix per comparar l'efecte de ponderació que introdueix l'índex RD_{migP} amb l'índex RD_{mig}. L'efecte que provoca és mantenir el valor de RD_{migP} en els instants on es coincideix amb el patró, i per altre banda reduir el valor RDmigp en els altres instants. Conseqüentment, s'obté un resultat de similitud més clar i precís.

A continuació s'ha optat per realitzar el mateix tipus d'anàlisi però aplicant un salt a la finestra lliscant del senyal, ja que en el microcontrolador és molt probable que apareguin salts entre espectre i espectre degut als recursos limitats i la no possibilitat d'executar operacions en paral·lel. Els salts signifiquen que l'anàlisi de similitud no es realitza de manera continua amb la finestra lliscant, sinó que aquesta es salta un temps del senyal cada vegada. Aquest fenomen suposarà menys vectors per a cada índex de similitud, ja que la finestra lliscant arribarà abans al final del senyal.

S'ha aplicat un salt de 19,2ms entre cada finestra lliscant. Aquest temps de salt s'ha seleccionat pensant que en les proves amb l'Arduino hi apareixerà un salt, entre bucle i bucle, molt proper a 20ms.

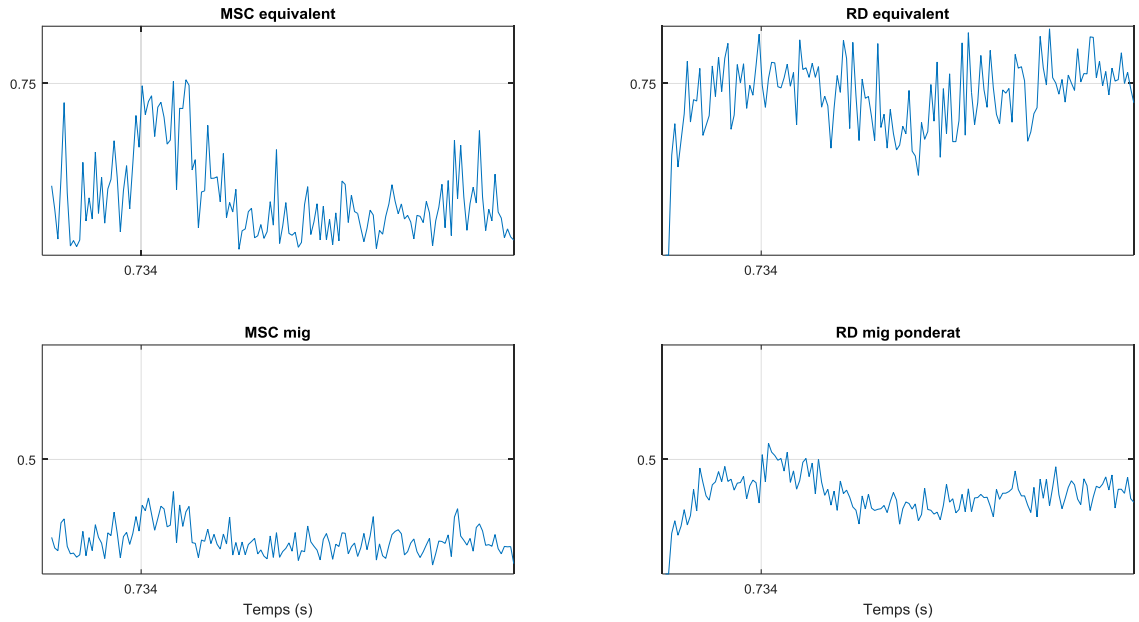

Figura 14. Anàlisi de similitud amb salts de 19,2ms entre l'Àudio Accident i el patró 1

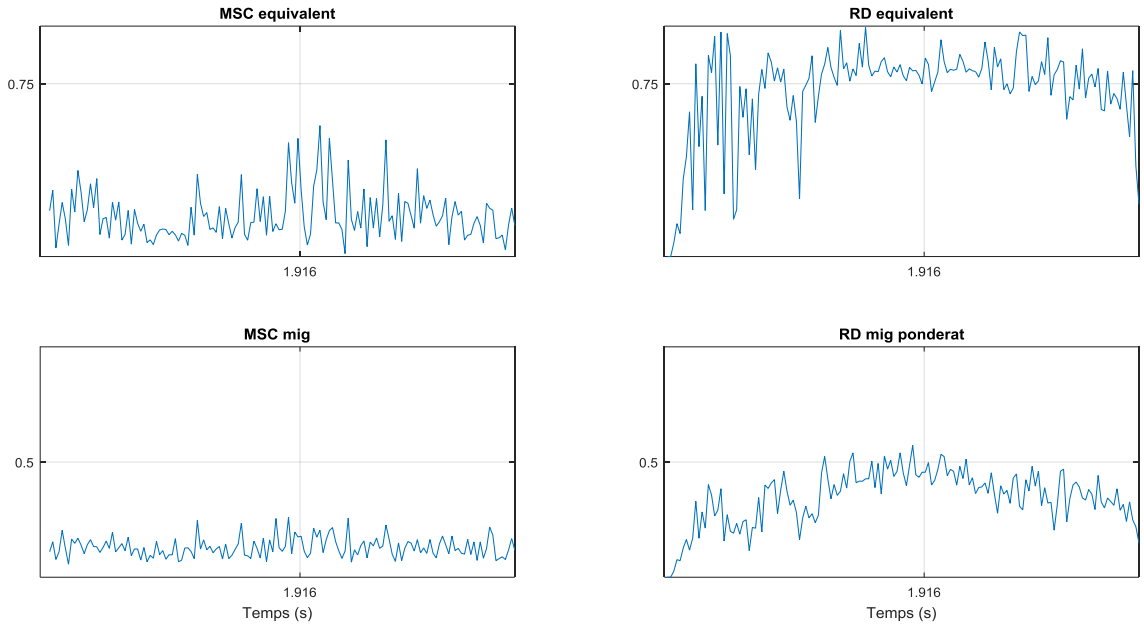

Figura 15. Anàlisi de similitud amb salts de 19,2ms entre l'Àudio Accident i el patró 2

Es pot observar que tot i els salts, apareix una tendència de similitud. Per el patró 1, la similitud apareix d'igual manera que sense salts. D'altre banda, per el patró 2 no hi ha tanta evidencia de similitud amb el mètode de MSC equivalent. En termes generals, amb l'anàlisi de similitud aplicant salts a les finestres, la similitud es torna més dèbil, ja que costa més que coincideixi el patró amb la part del senyal que és similar.

3.4.2 Senyals diferents

Es realitza la similitud entre els patrons i 2 arxius d'àudio d'accidents de trànsit (diferents als del sub-apartat anterior) i un àudio de música.

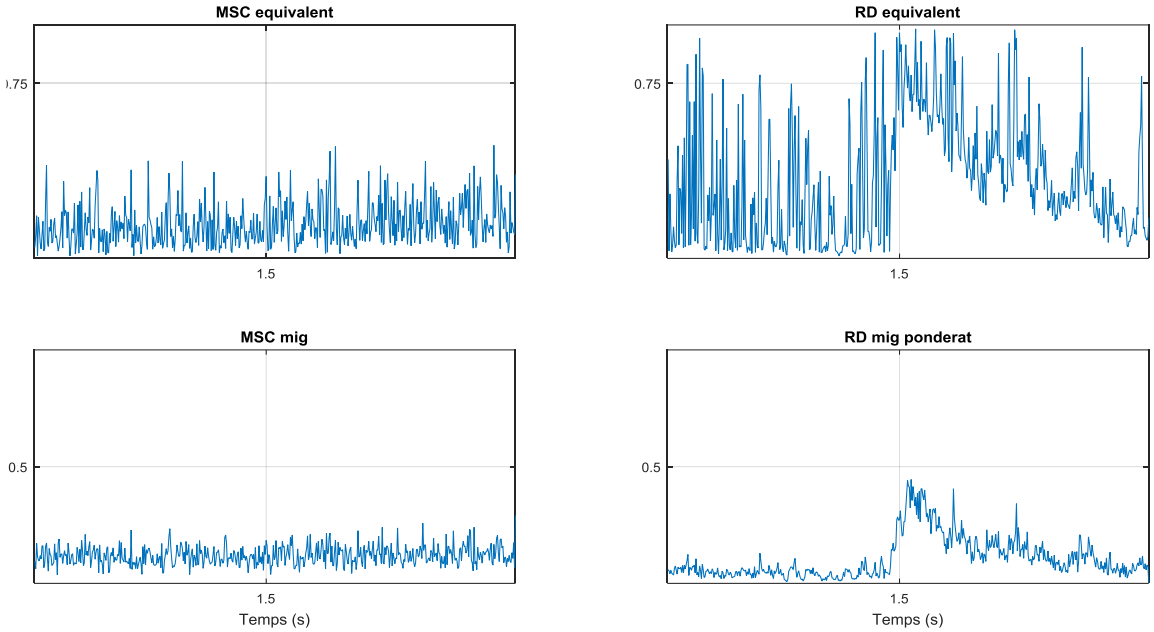

Figura 16. Anàlisi de similitud entre l'Àudio 2 accident i el patró 1

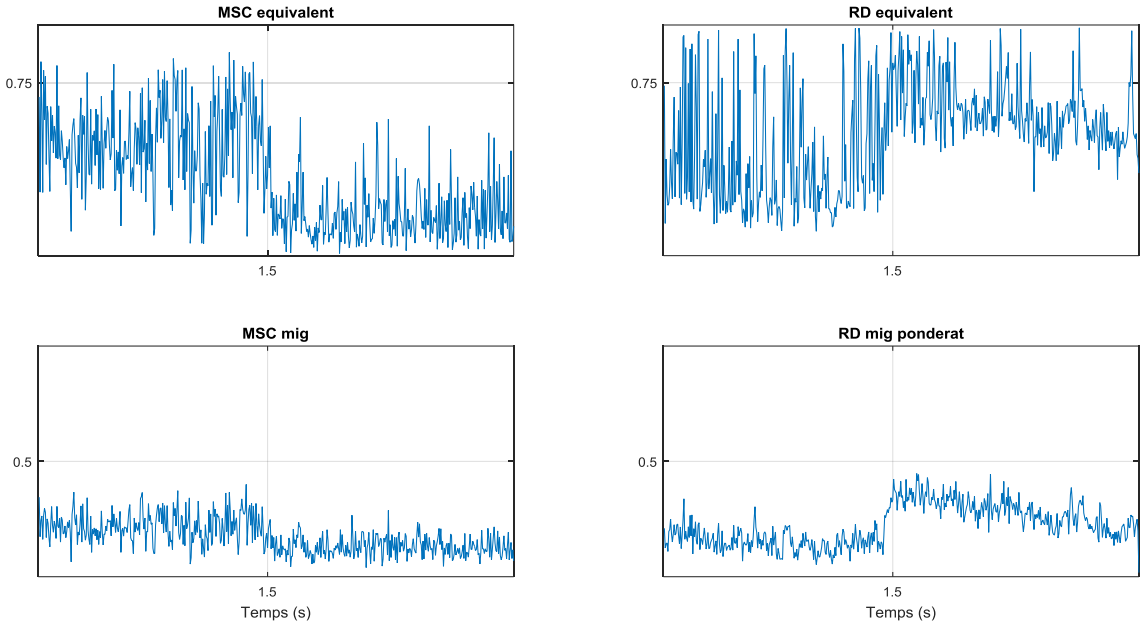

Figura 17. Anàlisi de similitud entre l'Àudio 2 accident i el patró 2

Comparant el senyal Àudio 2 accident amb els patrons, la similitud no és del tot clara per a tots els mètodes. El MSC equivalent mostra solament similitud pel patró 1 i el RD mig ponderat solament pel patró 2.

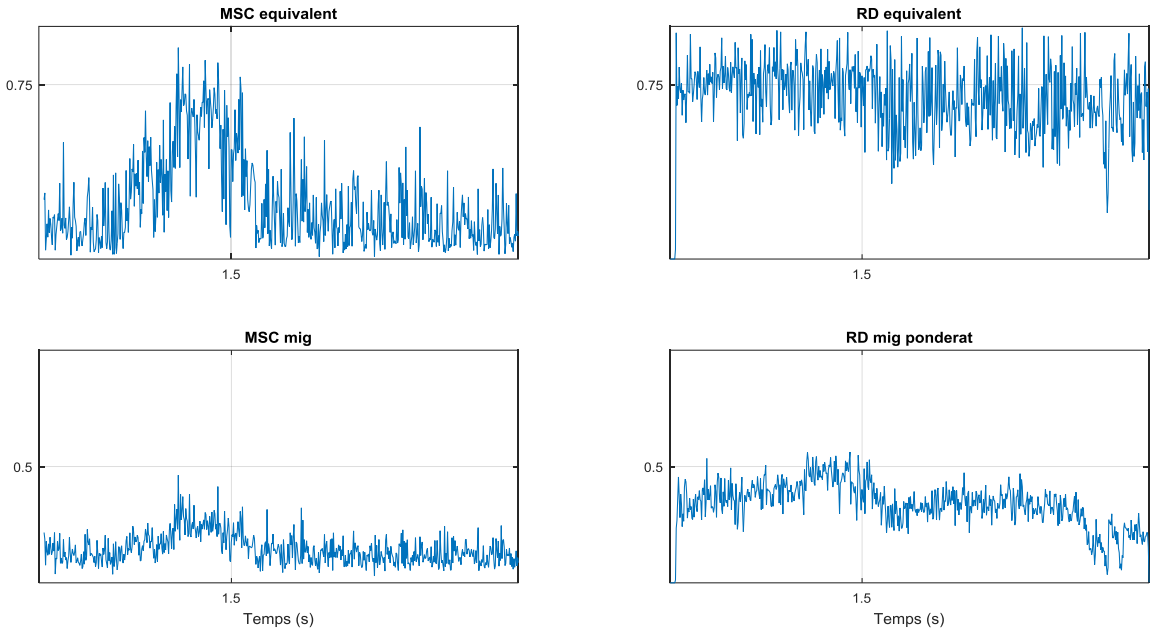

Figura 18. Anàlisi de similitud entre l'Àudio 3 accident i el patró 1

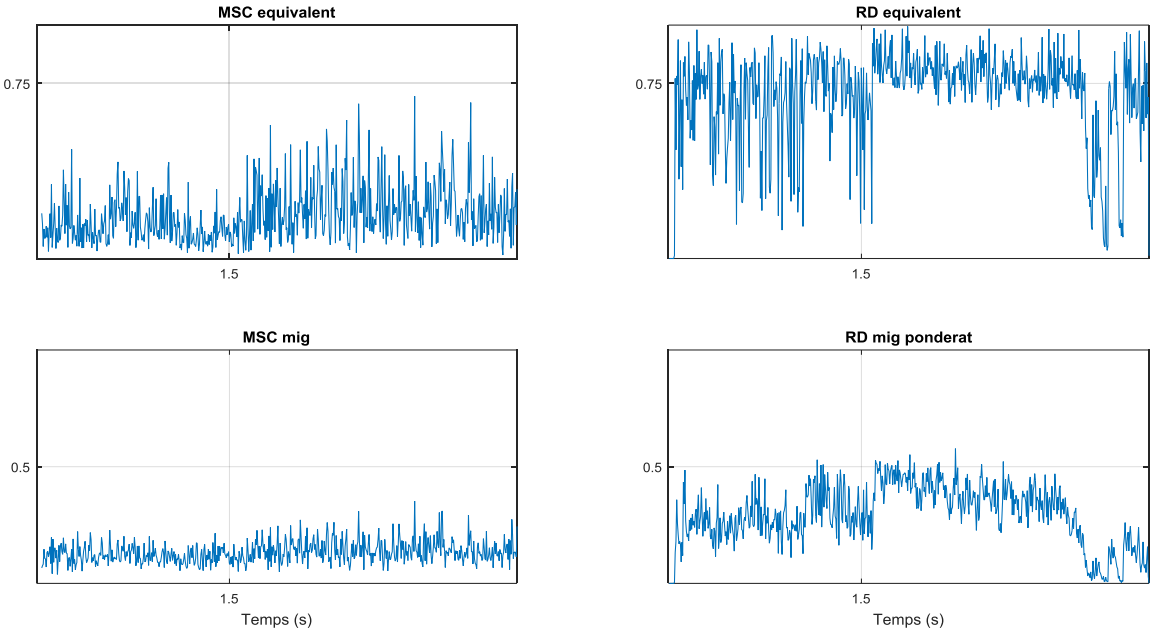

Figura 19. Anàlisi de similitud entre l'Àudio 3 accident i el patró 2

L'Àudio 3 accident presenta similitud amb els patrons. L'índex MSC equivalent i el RD mig ponderat mostren similitud tan per el patró 1 com per el patró 2. També apareix similitud en el mètode MSC mig per el patró 1.

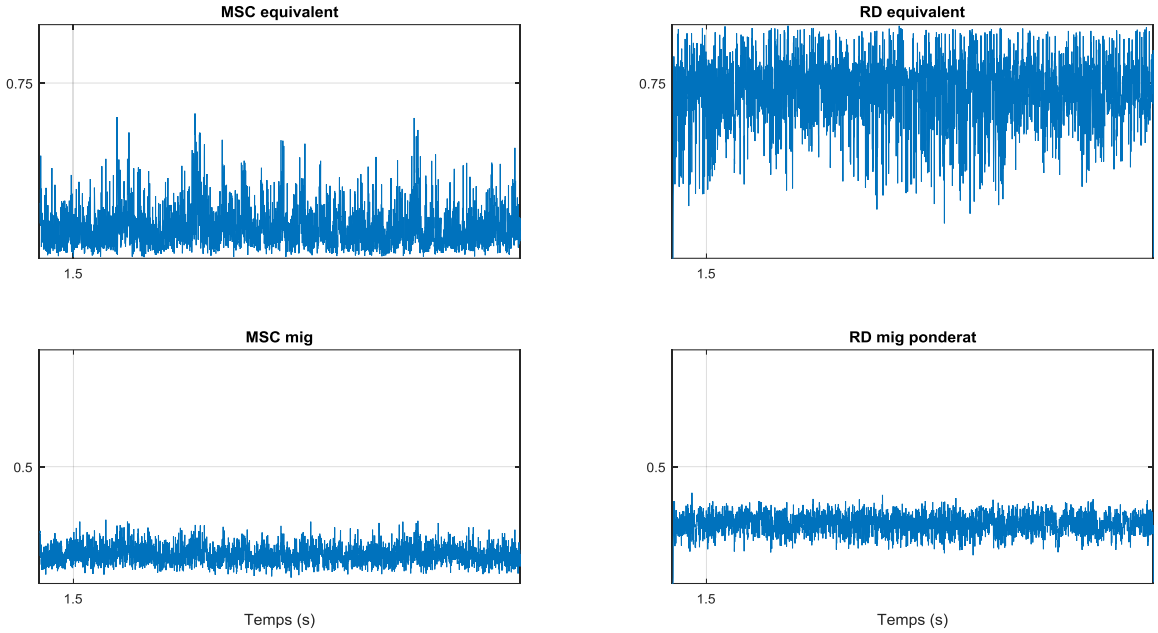

Figura 20. Anàlisi de similitud entre l'àudio música i el patró 1

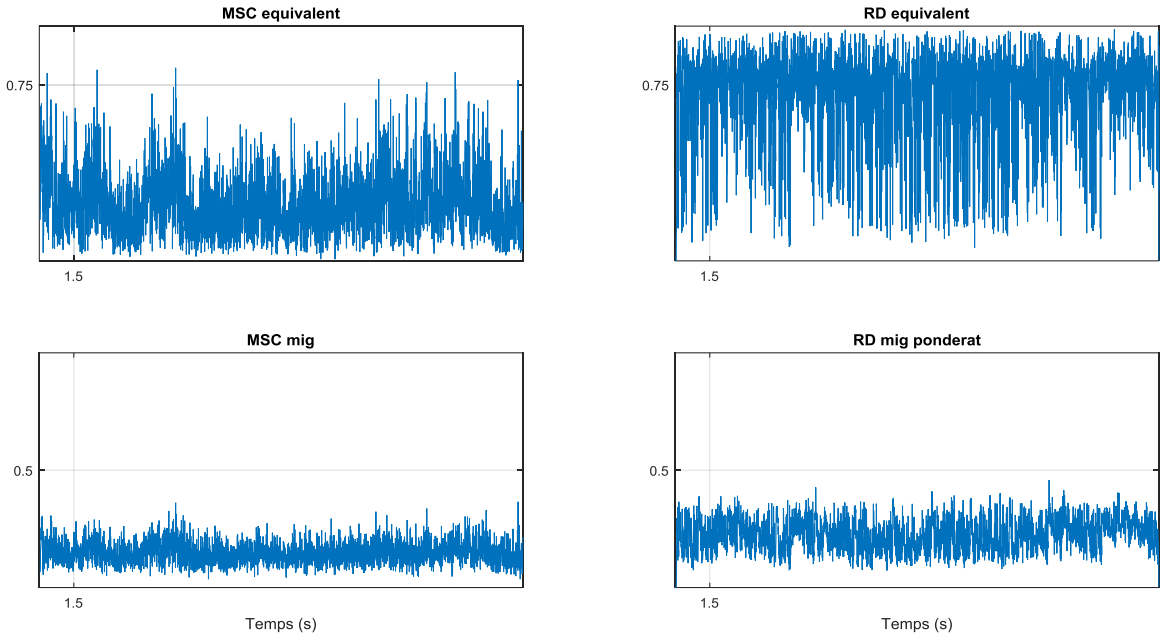

Figura 21. Anàlisi de similitud entre l'àudio música i el patró 2

Aquest darrer anàlisi amb l'arxiu Àudio música és útil per validar les diferents tècniques com els patrons. Pel patró 1 no apareix similitud per a cap índex. Per altre banda, l'índex MSC

equivalent mostra algun pic de similitud amb el patró 2, i algun valor del RD mig ponderat s'acosta a 0,5. Tot i que podria semblar il·lògic la similitud de l'Àudio música amb el patró 2, cal analitzar que l'Àudio música té una duració de 21 segons i per tant és molt probable que coincideixin espectres freqüencials.

3.5 Valoració patrons i mètodes de similitud

Les proves realitzades en aquest capítol ens corroboren que el patró 1 representa significativament la fase de trencament de vidres dels sons d'accidents de trànsit analitzats. Per el patró 2, no apareix una similitud clara comparant-lo amb altres senyals d'accidents de trànsit, però si una tendència a la similitud.

		Similitud			
		MSC_{equiv}	MSC_{mig}	RD_{equiv}	RD_{migP}
Àudio Accident	Patró 1	S	T	-	S
	Patró 2	S	S	-	S
Àudio 2 accident	Patró 1	S	T	-	
	Patró 2			-	T
Àudio 3 accident	Patró 1	S	T	-	S
	Patró 2	T		-	S
Àudio música	Patró 1			-	
	Patró 2	S		-	T

Figura 6. Tècniques de similitud i encerts en les anàlisis

La taula anterior resumeix els resultats de similitud entre els diferents senyals i patrons i per a cada tècnica de similitud utilitzada. La lletra S significa que s'ha trobat similitud, la T que no s'ha arribat al valor que dicta la similitud però la tendència del gràfic mostra clarament similitud.

El mètode MSC mig és el que té més bons resultats de similitud, ja que mostra similitud per als sons d'accidents i, contràriament, no en mostra pel senyal de música. També dona bons resultats el MSC equivalent, encara que el senyal de música ha proporcionat similitud amb el patró 2. La tècnica RD mig ponderat té un comportament bastant igual al de l'índex MSC equivalent, ja que amb els senyals d'accident mostra similitud i amb l'Àudio música també per el patró 2. Finalment el mètode RD equivalent queda descartat d'aquesta valoració ja que en totes les anàlisis ha mostrat similitud i per a casi tota la duració del senyal.

4. NODES I COMPONENTS ELECTRONICS

El sistema electrònic que es desenvolupa en aquest projecte està format per un node sensor i un node coordinador. Els nodes sensors realitzen l'adquisició d'àudio, analitzen l'espectre de freqüències i envien informació al node coordinador si es detecta un accident. El components electrònics que integren el node sensor són un micròfon, una etapa de pre-amplificació, un microcontrolador Arduino Pro Mini, un mòdul de comunicació ràdio NRF24, una bateria de 3,7V, una placa solar de 0,87W per recarregar una bateria de 3,7V i un driver per desconnectar la placa solar quan la bateria està totalment carregada.

El coordinador té la funció d'esperar informació dels nodes sensors i enviar alarmes a un servidor si es detecta un accident. Referent als components que integra el coordinador són un microcontrolador Arduino Pro Mini, un mòdul de comunicació ràdio NRF24, un mòdul de comunicació WiFi ESP8266, una bateria de 3,7V alimentada per una placa solar i un driver per desconnectar la bateria de la placa solar.

Dispositiu	Microcontrolador	Sensors	Comunicació ràdio	Comunicació WiFi	Alimentació	Driver alimentació
Node sensor	Arduino Pro Mini	Micròfon i amplificació	NRF24		Bateria i solar	TP4056
Coordinador	Arduino Pro Mini		NRF24	ESP8266	Bateria i solar	TP4056

Taula 7. Components electrònics del node sensor i del coordinador

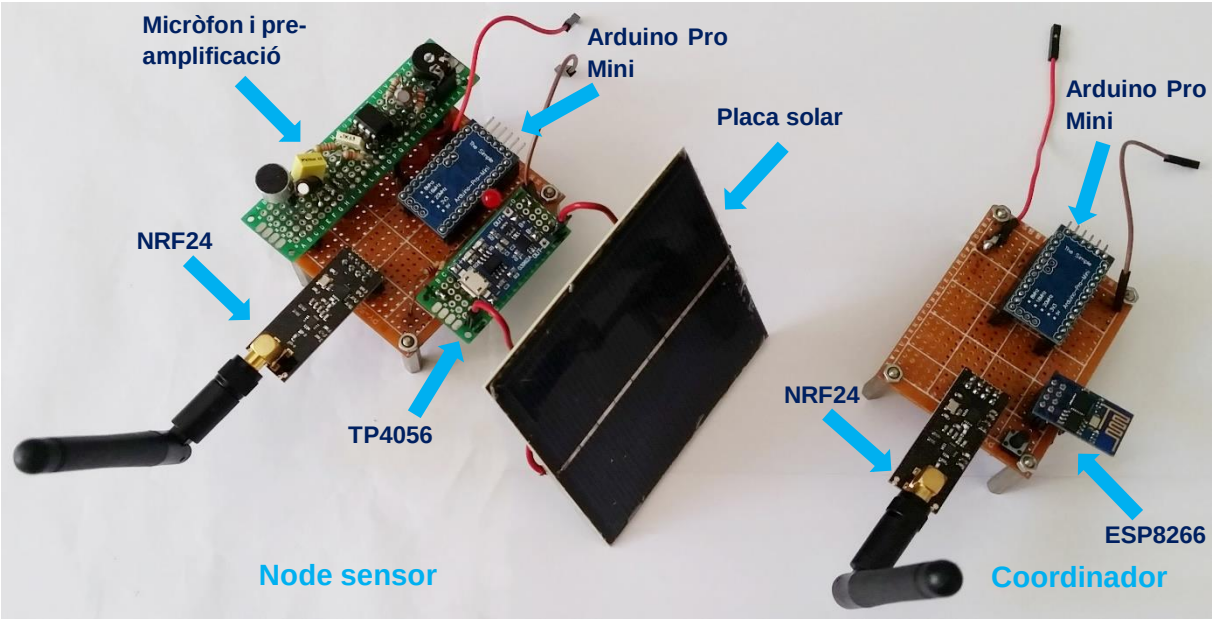

Figura 22. Components electrònics del node sensor i el coordinador

En la imatge anterior es veuen les plaques prototips que s'han dissenyat per poder realitzar les proves. La comprovació del funcionament i el rendiment de la placa solar i el driver TP4056, s'ha realitzat únicament en el node sensor. Les característiques i rendiments que s'especifiquen en el següent capítol sobre la placa solar i el driver del coordinador han estat agafades de les característiques del node sensor.

4.1 Micròfon i pre-amplificació

Per enregistrar el so és necessari un sensor que converteixi les ones en senyal elèctric. Si el senyal elèctric que proporciona el micròfon és baix, llavors caldrà una etapa d'amplificació. La digitalització es realitza a través d'un convertidor analògic a digital, que guarda en format digital el senyal del micròfon a una freqüència de mostreig determinada.

4.1.1 Sensor d'àudio electret

Els micròfons són els dispositius electrònics que permeten captar un so analògic. El sensor escollit és el sensor electret ja que és un micròfon barat, té una bona resposta en freqüència i és molt sensible degut a la seva part interna de pre-amplificació. Un dels seus grans avantatges és la insensibilitat a la temperatura i a la humitat, i un dels desavantatges és el deteriorament que li pot provocar el contacte de la pols.

Figura 23. Micròfon electret

La direccionalitat d'aquest tipus de micròfons és omnidireccional, és a dir, espacialment abraça la captació de sons a 360º. La seva impedància màxima de sortida és de menys d'1kHΩ i té un consum màxim de 0,5mA. Com a inconvenient, es satura per a pressions sonores altes i el transistor genera soroll degut a la alta resistència d'entrada al JFET.

Referent a la relació senyal/soroll (SNR) del micròfon, té un valor aproximat de 58dB. Per tant el soroll que presenta el micròfon és petit en comparació al senyal que capta. La seva sensibilitat és de -42dB.

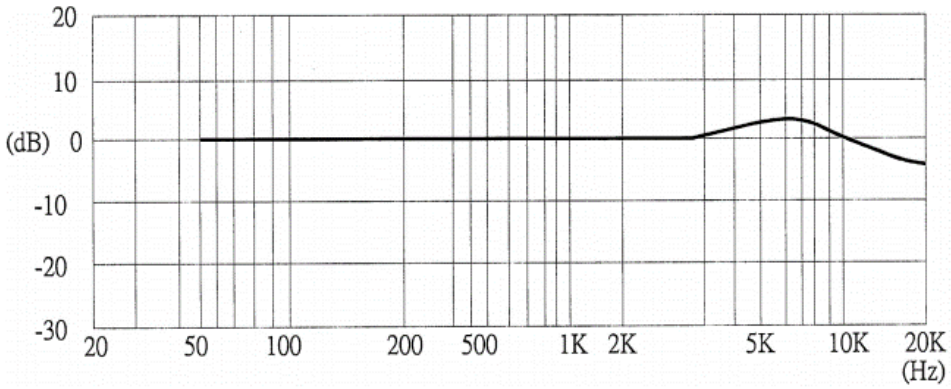

Figura 24. Resposta en freqüència d'un micròfon electret

En el gràfic anterior es pot identificar l'ample de banda que tenen el micròfons electrets. S'ha de tenir en compte que els micròfons electrets accentuen els sons aguts d'entre 4.000 a 9.000Hz. Observant l'espectre de freqüències dels sons d'accidents analitzats, la major part d'energia sonora es concentrava en les components freqüencials per sota de 5.000Hz, per tant, no ha de suposar cap inconvenient la corba de resposta en freqüència dels micròfons electrets.

Internament el micròfon electret està format per unes plaques de condensador, una d'elles és fixa i l'altra mòbil i sensible a l'aire exterior, i un transistor JFET de canal n per amplificar. Quan l'aire provinent d'un so arriba al micròfon aleshores es mou la membrana endavant i endarrere, i aquest canvi provoca una variació en el condensador. El condensador esta connectat a la comporta (G) del transistor JFET. En el pin de drenatge del JFET hi arriba l'alimentació a través d'una resistència, i la sortida del so està connectat al pin de drenatge (D) del JFET. El material dielèctric electret manté una càrrega elèctrica fixa, i per tant un voltatge en el condensador.

4.1.2 Etapa de pre-amplificació

Degut a que el senyal provinent del micròfon electret és molt baix (a l'ordre de pocs mV), és necessari incorporar una etapa d'amplificació a continuació del micròfon. Aquesta amplificació permetrà també ajustar el senyal en el rang del convertidor analògic digital i així aprofitar millor la resolució.

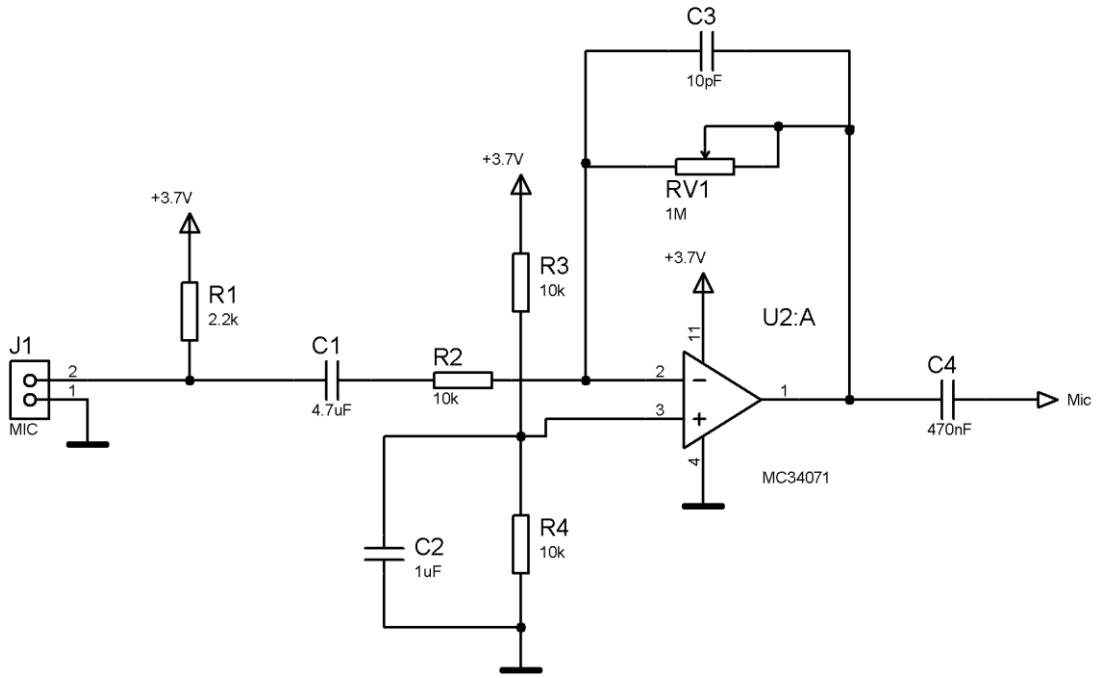

Figura 25. Esquema electrònic etapa de pre-amplificació

El circuit pre-amplificador està format per el senyal del micròfon i una part amplificadora. L'amplificador proporciona un guany al senyal i actua alhora com a filtre passa baix. El guany ve determinat per la resistència variable RV1 i per la resistència R2.

$$Av = -\frac{RV1}{R2} \quad \text{(Eq. 21)}$$

$$Av = -\frac{100k}{10k} = -10 \quad \text{(Eq. 22)}$$

S'inverteix el senyal i la fase, però per a l'anàlisi espectral no té cap conseqüència. La freqüència de tall del filtre passa baix és la següent:

$$Fc = \frac{1}{2 \cdot \pi \cdot RV1 \cdot C3} \quad \text{(Eq. 23)}$$

$$Fc = \frac{1}{2 \cdot \pi \cdot 100k \cdot 10 \cdot 10^{-12}} = 159.155Hz \quad \text{(Eq. 24)}$$

El so d'accidents té freqüències importants entre 100 i 10.000Hz. Per tan, aquesta freqüència de tall de més de 100.000Hz permet l'adquisició correcte del so desitjat.

4.2 Arduino Pro Mini

Tan el node sensor com el coordinador necessiten un microcontrolador que processi les dades i executi les comunicacions. S'ha escollit la marca Arduino, ja que aquesta família de microcontroladors disposa d'una gran quantitat de llibreries i molta informació de suport a internet. El seu llenguatge de programació està basat en el llenguatge C++. Existeixen varies plaques Arduino, amb diferents mides de memòria i nombre de pins.

4.2.1 Microcontrolador Atmega328

Per el present projecte es va optar per l'Arduino Pro Mini ja que ofereix un baix consum, una mida de la placa reduïda de 2x3,5cms i unes prestacions de velocitat i memòria suficients per a l'adquisició i anàlisi del so. La seva velocitat de processament és de 16MHz i es pot alimentar entre 3 i 5V.

Figura 26. Arduino Pro Mini

L'Atmega328 és un microcontrolador amb una CPU de 8 bits, amb 14 pins digital (6 poden utilitzar-se com a PWM), 6 pins analògics i 6 pins per a la connexió TTL a USB (usada per a programar el microcontrolador). Pot donar 20mA per pin. El consum del micròfon és de 2mA, i el consum dels NRF24 i ESP8266 és directe de la bateria.

Disposa de memòria no volàtil de programa (Flash) de 32kB, memòria volàtil de dades (SRAM) de 2kB i memòria no volàtil de paràmetres (EEPROM) de 1kB. Referent a la freqüència de treball, és de 16MHz i permet executar una instrucció cada dos cicles de rellotge, per tant per processar una instrucció es tarden 125ns.

S'utilitzen dos Arduino en el present projecte. Un pel dispositiu node principal i l'altre pel dispositiu coordinador. En el següent requadre hi ha representades les connexions entre els

pins de l'Arduino i els altres components, tan pel node principal com per al coordinador. L'abreviatura D significa pin digital, i pot ser tan de sortida com d'entrada. L'abreviatura A significa entrada analògica. MOSI, MISO, SCK són pins per utilitzar la comunicació sèrie SPI. TX i RX són els pins per usar la comunicació sèrie UART.

Nom	Pin	Tipus	Node micròfon	Coordinador
PB0	12	D8		
PB1	13	D9	NRF24 CS	NRF24 CS
PB2	14	D10		
PB3	15	MOSI	NRF24 MOSI	NRF24 MOSI
PB4	16	MISO	NRF24 MISO	NRF24 MISO
PB5	17	SCK	NRF24 SCK	NRF24 SCK
PB6	7	TOCS1		
PB7	8	TOSC2		
PC0	23	A0	MICRÒFON	
PC1	24	A1	VCC PLACA SOLAR	VCC PLACA SOLAR
PC2	25	A2		
PC3	26	A3		
PC4	27	A4		
PC5	28	A5		
PC6	29	RESET	(FTDI PC)	(FTDI PC)
PD0	30	RX	(FTDI PC)	ESP8266 TX / (FTDI PC)
PD1	31	TX	(FTDI PC)	ESP8266 RX / (FTDI PC)
PD2	32	D2		
PD3	1	D3		ESP8266 CH_PD
PD4	2	D4		
PD5	9	D5		
PD6	10	D6		
PD7	11	D7		
VCC	4	VCC	VCC BATERIA	VCC BATERIA
VCC	6	VCC	(FTDI PC)	(FTDI PC)
AVCC	18	AVCC		
AREF	20	AREF		
GND	3	GND	GND BATERIA	GND BATERIA
GND	21	GND	(FTDI PC)	(FTDI PC)
AGND	5	AGND		
ADC6	19	A6		
ADC7	22	A7		

Taula 8. Distribució de pins entre els Arduino i els mòduls electrònics

Una de les particularitats que té l'Atmega328 és que disposa de varis modes d'operació de baix consum d'energia. Llibreries com "LowPower" i "avr/sleep" permeten adormir el microcontrolador amb un consum de 0,1mA.

4.2.2 Convertidor analògic digital

El soroll d'accidents té freqüències importants entre 100 i 12.000Hz. Per tant, segons el teorema de Nyquist la freqüència de mostreig ha de ser com a mínim el doble de la freqüència del senyal. L'Atmega328P té un convertidor analògic digital de 10 bits i pot mostrejar a 40.000Hz. Així doncs, el convertidor ADC adquireix una mostra delper t senyal a cada 25us.

Per tal de tenir la màxima resolució possible en la conversió d'un senyal analògic a digital s'ha de tenir el senyal adaptat al rang de l'entrada analògica (0-3,,3V). Ara bé, en el cas de l'adquisició de so, l'amplitud màxima té relació amb la distància i amb la potència sonora emesa, així com també del guany que proporciona l'etapa de pre-amplificació. Aquest guany que s'ajusta a partir del potenciòmetre de l'etapa de pre-amplificació, ha de permetre que entre les distàncies desitjades de l'emissió de so i el micròfon, l'ADC treballi des del seu mínim fins al seu màxim voltatge d'entrada. En el cas que el senyal arribi en el màxim rang d'entrada de l'ADC, la resolució que tindrà el senyal convertit a digital serà:

$$r = \frac{3.3V}{2^{10} - 1} = 3{,}2mV \qquad \text{(Eq. 25)}$$

El senyal que li entrem a l'entrada analògica per a ser detectada haurà d'agafar un valor mínim de 3,2mV, i l'ADC canviarà de bit per cada 3,2mV de variació del senyal analògic.

4.3 Comunicacions

Referent al sistema de comunicació sense fils entre nodes i el coordinador, s'ha seleccionat el mòdul NRF24 ja que es pot programar fàcilment amb la IDE Arduino i el seu consum en mode "sleep" és reduït d'uns 0,1mA. Per a la connexió del coordinador amb una xarxa WiFi s'ha optat per el mòdul ESP8266 ja que també es pot integrar la programació en la IDE Arduino i el seu cost és molt baix.

4.3.1 NRF24

La comunicació entre cadascun dels nodes amb el coordinador es realitzarà mitjançant radiofreqüència. L'integrat a escollir ha de ser de baix consum i de reduït cost. Es va descartar el mòdul Xbee degut al seu alt cost. Altres alternatives eren el RFMHW69, que opera a la banda de 915MHz i el NRF24, a la banda ISM de 2.400MHz. L'integrat RFM69HW69 és adequat per a distancies llargues de fins a 500 metres, ara bé el seu consum s'eleva a 150mA. Finalment es va escollir l'integrat NRF24 amb un amplificador de potència i una antena integrada, (comercialment denominat NRF24 PA LNA), ja que té un consum de 0,01mA en mode "sleep" i una distància màxima de comunicació de 900 metres en exterior (segons el fabricant).

El NRF24 és un mòdul que té 125 canals comunicació i una velocitat de transmissió ajustable a 250kbps, 1Mbps i 2Mbps. La potencia de sortida és de 100mW (20dBm) i es pot reduir fins a 1mW, i la sensibilitat varia en funció de la velocitat de transmissió, per 250kbps és de -104dBm i per 2Mbps és de -82dBm. Té un amplificador i una antena externa per tal de millor la distància de comunicació. El mòdul disposa de 8 pins, dels quals 6 corresponen a la comunicació SPI i dos d'alimentació. La seva alimentació d'energia pot anar des de 1,9V fins a 3,7V, i té un consum de pic de 100mA per a la transmissió i de 40mA per a la recepció. El NRF24 disposa del mode "standby" en què el consum d'energia és redueix a menys de 0,1mA. La distància de transmissió entre mòduls NRF24 és de fins a 900 metres a l'aire lliure (segons dades del fabricant), per tant, cada node sensor ha d'estar a una distància màxima de 900 metres del coordinador.

Figura 27. NRF24 PA LNA

El sistema de comunicació que utilitza és la modulació digital GFSK (modulació per desplaçament de freqüència gaussiana) que consisteix en una ona moduladora que conté la

informació (1s o 0s) i una portadora que varia de freqüència. Per representar un 1 lògic la ona portadora augmenta la freqüència i per el 0 lògic es disminueix la freqüència de la portadora.

Porta incorporat el protocol de radiocomunicació Enhanced ShockBurst. Aquest es basa en la comunicació bi-direccional entre el transmissor i el receptor i amb l'enviament de paquets amb acusament de recepció. El transmissor envia el paquet juntament amb la direcció del destinatari, i posteriorment es queda en mode receptor per esperar l'acusament de recepció. El receptor està sempre escoltant i quan detecta un paquet que té la seva adreça envia un acusament de recepció a l'emissor. Si un paquet no ha arribat al destinatari, el transmissor no rebrà el acusament de recepció (ACK) i tornarà a enviar automàticament el paquet. Es pot configurar el nombre de re-enviaments i el temps d'espera entre aquests, en el cas que no arribi el paquet al destí.

Aquest protocol hardware permet la comunicació en xarxa de tipus estrella o arbre, amb fins a sis nodes transmissors penjant i comunicant-se bi-direccionalment a través d'una línia amb un receptor. Alhora, de cada node pot penjar-ne 6 nodes més. Per tal de que un node i un coordinador es comuniquin han de tenir definida la mateixa adreça, i per a cada node és diferent aquesta adreça. Tots aquests nodes i coordinadors que estan relacionats han de tenir configurat el mateix canal de freqüència. També es possible la comunicació directa entre dos nodes definint l'adreça del node receptor al paquet a enviar.

El programa del NRF24 amb les ordres i dades per a realitzar la comunicació amb un altre dispositiu NRF24 s'emmagatzema i s'executa a l'Arduino, i a través de la comunicació sèrie síncrona SPI, l'Arduino envia les comandes i totes les dades d'enviament dels paquets al mòdul NRF24.

4.3.2 ESP8266

Quan el coordinador obté dades i avisos dels nodes, ha de transmetre aquesta informació a un servidor. S'ha optat per a la tecnologia WiFi, ja que està completament desplegada a les ciutats. Es va descartar la tecnologia GPRS degut a que requereix el pagament d'una quota mensual pel servei de transmissió de dades. Tanmateix, en les zones on no hi ha WiFi es podria integrar perfectament la tecnologia GPRS.

L'integrat ESP8266 és un mòdul de comunicació WiFi, que opera a la banda de 2,4GHz. Fa servir el protocol TCP/IP, per tant es pot connectar a qualsevol xarxa WiFi. Disposa d'una

CPU i de unitat de memòria, per tant, aquest mòdul permet emmagatzemar un programa. Es comunica amb l'Arduino per intercanviar dades via els dos pins de l'UART. La velocitat de transmissió de dades a una direcció IP concreta és de fins a 11Mbps (estàndard 802.11 b). El seu consum és de 100mA transmetent i 0,1mA en mode "sleep". Es pot programar que estigui un determinat temps adormit i llavors es desperti, o bé es pot posar en estat baix el pin CH_PD del ESP8266 i llavors entra en mode "power down".

Per introduir el programa de comandes del ESP8266 es pot usar la IDE d'Arduino, on s'hi ha d'instal·lar una extensió addicional per a què es programi correctament el mòdul ESP8266. En el programa del ESP8266 s'ha d'incloure l'identificador SSID i la contrasenya del router, i l'adreça IP i la contrasenya a on enviar les dades. La distància de connexió a un router és d'uns 20 metres.

4.4 Sistema d'alimentació

Per a subministrar l'energia necessària als dispositius node i coordinador s'utilitzen dos bateries de 3,7V de 5.000mAh. A més, s'hi integra una placa solar per tal que siguin totalment autònoms en el consum d'energia. El driver TP4056 detecta que la bateria ha arribat al nivell màxim de carrega i desconnecta el subministrament d'energia de la placa solar a la bateria.

4.4.1 Consum d'energia

Cada dispositiu consumeix una quantitat diferent d'energia, relacionat amb els components que utilitza. El component que consumeix més és el ESP8266, ara bé, no està funcionant el 100% del temps.

L'apartat de comunicació sense fils consumeix molta energia. El mòdul NRF24 quan transmet (node sensor) consumeix a voltant de 100mAh, i quan rep (coordinador) uns 40mAh. Ara bé, tal com està especificat en el capítol 6 (programari), només es transmet i es rep la informació del nivell de bateria i de voltatge solar 1 vegada cada 30 minuts, o si apareix un accident. La resta de temps en què no cal utilitzar (Fu) el NRF24, al voltant del 99% pel node sensor i del 80% pel coordinador, es pot adormir el mòdul NRF24 i per tant el consum és redueix molt. El mateix cas és amb el mòdul ESP8266.

Component	Consum (mAh)	Fu Node s	Consum Node S (mAh)	Fu Coordinador	Consum Coordinador (mAh)
Arduino	7,00	1,00	7,00	1,00	7,00
Micròfon	2,50	1,00	2,50	0,00	0,00
NRF24 - Mode Tx	100,00	0,01	0,50	0,00	0,00
NRF24 - Mode Rx	40,00	0,00	0,00	0,20	8,00
NRF24 - Mode "sleep"	0,10	0,99	0,10	0,80	0,08
ESP8266 - Mode Tx	100,00	0,00	0,00	0,01	1,00
ESP8266 "sleep"	0,10	0,00	0,00	0,01	0,00
TP4056	0,20	1,00	0,20	1,00	0,20
Total hora			10,30		11,03
Total dia			**247,18**		**390,74**

Taula 9. Consum diari mig del node i del coordinador

El dispositiu node sensor té un consum de uns 10mAh, que al dia són al voltant de 250mAh. Com que la placa solar aporta més mA que el que consumeix el node, aquest és autosuficient. Per altre banda el coordinador consumeix més, uns 390mAh, degut a que el mòdul de transmissió no està adormit tant de temps.

4.4.2 Energia solar

El sistema d'alimentació solar està format per una placa solar i un driver. La placa solar té unes dimensions de 8,5cm per 8,5cm. Si el sol brilla a plenitud la placa pot aportar una potencia de pic de 0,87W i una corrent de 175mAh. El driver TP4056 controla que la bateria estigui carregada.

S'ha incorporat un divisor de tensió de la placa solar a massa per tal de mesurar, al punt mig, el nivell de voltatge de la placa solar. L'entrada analògica A1 del microcontrolador llegeix aquest voltatge, ara bé, el voltatge màxim que proporciona la placa solar és de 5V, i el voltatge alt per a l'entrada analògica és de 3,3V. En aquest punt s'obté el voltatge de la placa solar dividit per dos, per tant el màxim voltatge que mesura l'entrada analògica es de 2,5V. Aquestes dades s'enviaran al servidor a través del coordinador. Les resistències són d'un valor alt per reduir-ne el consum de corrent (de l'ordre de uA).

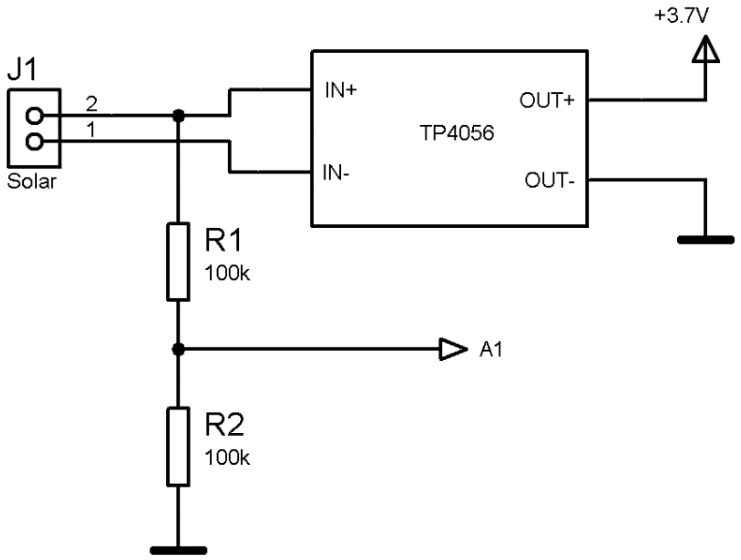

Figura 28. Esquema electrònic de la placa solar i el driver TP4056

L'integrat TP4056 està recomanat per a la recarrega de bateries de 3,7V. A partir d'una resistència detecta la corrent de carrega que arriba a la bateria. Quan la bateria arriba al seu voltatge màxim, aquesta corrent ha disminuït per sota un llindar, i es tanca la connexió entre la placa solar i la bateria. Aleshores la bateria comença a disminuir d'energia fins que arriba al voltant dels 3V, en què el TP4056 torna a connectar la placa solar amb la bateria. El consum d'aquest integrat és de 0,1mA, segons dades del fabricant.

L'energia que aportarà la placa solar depèn de la irradiació solar. Pel càlcul de l'energia diària solar es fa servir una fórmula genèrica que calcula el nombre de panells solars a instal·lar a partir d'una determinada energia a consumir. En el nostre cas, el número de panells solars es fixe (només 1), per tant es pot calcular l'energia que aporta aquest panell amb les dades de irradiació que facilita el servidor de Photovoltaic Geographical Information System.

$$N_{mod} = \frac{C_{ed}}{P_{Mp} \cdot HSP_{crit} \cdot PR} \quad \text{(Eq. 26)}$$

on,

N_{mod} = número de mòduls solars a instal·lar

C_{ed} = consum d'energia de la instal·lació a alimentar, en Wh

P_{Mp} = potencia pic de la placa solar, en Wh

HSP = índex hores solar pic

PR = coeficient d'eficiència de la placa solar.

Agafant de referencia HSP igual a 3,6 del mes de desembre (és el mes amb menys irradiació solar), la quantitat d'energia que aportarà la placa en un dia és:

$$C_{ed} = N_{mod} \cdot P_{Mp} \cdot HSP_{crit} \cdot PR \quad \text{(Eq. 27)}$$

$$C_{ed} = 1 \cdot 0{,}87 \cdot 3{,}55 \cdot 0{,}8 \quad \text{(Eq. 28)}$$

$$C_{ed} = 2{,}47 Wh/dia \quad \text{(Eq. 29)}$$

Aquest càlcul és correcte per el mes de desembre, ara bé, els altres mesos hi ha més irradiació solar i per tant l'energia solar de la placa serà superior. Aquest model és vàlid suposant que la placa solar està ben ubicada i orientada perquè el sol li toqui el màxim d'hores durant el dia.

4.4.3 Autonomia dels dispositius

Una vegada coneguts els consums i el rendiment de la placa solar es pot extreure l'autonomia de cada dispositiu.

Dispositiu	Consum dia (Wh)	Consum dia (mAh)	Energia solar dia (mAh)	Bateria (mAh)	Autonomia (dies)
Node sensor	0,82	247,18	494,16	5.000,00	365,00
Coordinador	1,29	390,74	494,20	5.000,00	365,00

Taula 10. Autonomia del node sensor i del coordinador

La placa solar aporta diàriament més energia de la que es consumeix, per tant la bateria no arriba mai a exhaurir-se en cap dels dos dispositius. Ara bé, una vegada a l'any és necessari revisar l'estat de la bateria, per tant, l'autonomia és de 365 dies tan per el node sensor com el coordinador.

5. PROGRAMARI I DIAGRAMA DE FUNCIONAMENT

L'apartat següent inclou els detalls dels programes per al microcontrolador del node principal, per al coordinador i per el ESP8266. S'explica l'algorisme general dels programes, les tasques que contenen així com els sistemes de comunicació entre dispositius. El llenguatge usat per a la programació dels Arduino i del ESP8266 és el C++.

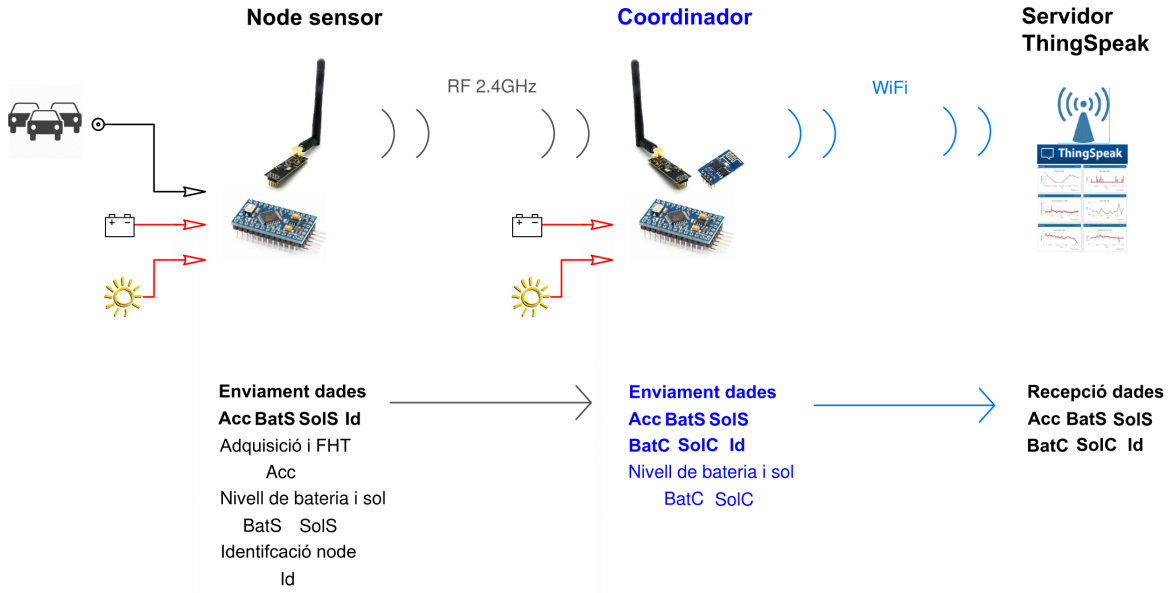

Figura 29. Diagrama de dispositius i d'informació

5.1 Diagrama de flux general

Al moment d'alimentar els dispositius, aquests arranquen i primer configuren els paràmetres inicials que seran necessaris per executar el programa general.

Seguidament el node sensor és l'encarregat d'adquirir el so, determinar el nivell de bateria i el voltatge de la placa solar, calcular la FHT, realitzar l'anàlisi entre el so obtingut i els patrons i finalment, si és necessari, enviar informació al coordinador. El node sensor només envia informació una vegada cada 30 minuts (T1) o si es detecta que s'ha produït un accident.

Referent al coordinador, sempre està en espera de rebre informació del node sensor. Quan la rep, codifica les trames i les traspassa al mòdul ESP8266. Cal remarcar que els dispositius estan funcionant en paral·lel, i es comuniquen mitjançant els mòduls NRF24 sense fils. L'objectiu final és transmetre la informació al servidor ThingSpeak, a través del ESP8266, per tal de posar les dades a disposició de qualsevol organisme.

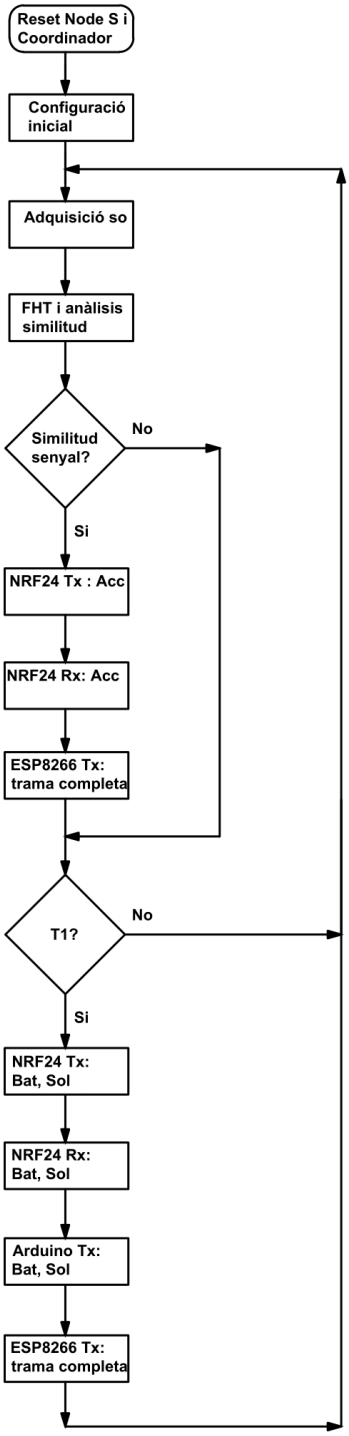

Figura 30. Diagrama de flux del funcionament general dels dispositius

5.2 Node sensor

El següent diagrama mostra el funcionament del programa per al node sensor, que està format per un bucle principal amb varies tasques que realitzen l'adquisició, la transformada FHT, la detecció de similitud i l'enviament de dades al node coordinador.

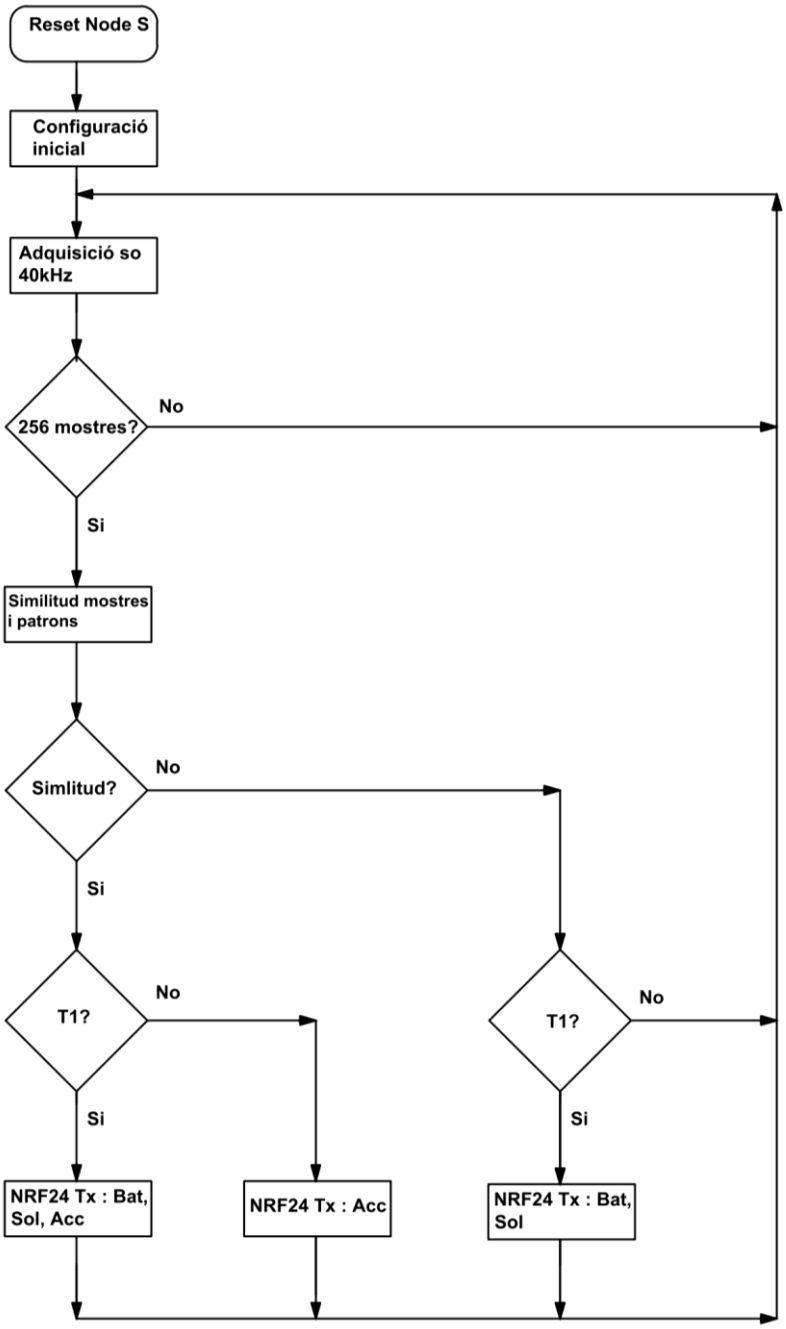

Figura 31. Diagrama de flux del funcionament del node sensor

5.2.1 Detecció del nivell de bateria i d'energia solar

S'ha usat la referencia interna de 1,1V que té l'Atmega328 per a la mesura del voltatge de la bateria. Consisteix en una mesura interna que serveix per comparar qualsevol entrada analògica amb aquesta referencia. Per tant, el valor analògic que llegeix de la bateria el divideix entre 3 i el compara amb 1,1V. Aquest voltatge de la bateria el guardem a la variable BatS de tipus float.

La detecció del nivell de voltatge de la placa solar es realitza a través del pin analògic 1. El nivell màxim de voltatge pot arribar la placa solar és de 5V, i per tal de no passar del rang d'entrada del pin (3,3V) s'ha integrat un divisió de tensió i la mesura analògica es fa en el centre del divisió de tensió. El valor de voltatge de la placa solar es guarda a la variable SolS de tipus float.

Aquesta informació de la bateria i de la placa solar no cal enviar-la al coordinador a cada iteració del bucle, ja que els voltatges no varien amb tanta rapidesa. S'ha establert un temporitzador T1, que contingui 100.000 iteracions del bucle, es procedirà a enviar el paquet de dades al coordinador. La duració mitjana de tot el bucle del node sensor és de un 18ms, aleshores 100.000 iteracions corresponen a 30 minuts.

$$T1 = 100.000 \cdot 18ms = 1.800.000ms = 1.800s = 30min \quad \text{(Eq. 30)}$$

5.2.2 Adquisició i FHT

L'ADC és l'encarregat de capturar el senyal analògic que prové del micròfon i l'etapa de pre-amplificació. L'adquisició de les mostres del senyal es realitza mitjançant interrupcions per Hardware. Cada vegada que el buffer conté una mostra s'interromp el programa principal per tractar i emmagatzemar aquesta mostra. Les mostres són adquirides a una velocitat de 40.000Hz, per tant cada 25us arriba una mostra nova. Cada mostra s'emmagatzema en 10 bits, per tant la variació de l'entrada analògica mínima que fa canviar de bit és de 3,23mV.

Per representar el senyal en el domini freqüencial en un primer moment es va utilitzar una llibreria que executa l'algorisme de la transformada rapida de Fourier (FFT). Ara bé, posteriorment s'ha optat per la transformada ràpida de Hartley (FHT) ja que aquest algorisme realitza les operacions en nombre reals (la FFT ho fa amb reals i complexos) i conseqüentment es redueix el nombre de dades usades i operacions.

S'ha usat la llibrera <FHT.h> d'Arduino per a realitzar la transformada ràpida de Hartley. Per a calcular la transformada es necessiten 256 mostres del senyal i la FHT retorna un vector de 128 magnituds reals de freqüències. Aquestes components freqüencials estan repartides equidistantment en un espectre de 0 a 20.000Hz, per tant entre cada component hi ha una separació de 156Hz.

El càlcul de la FHT es realitza a l'inici del bucle de programa. Per altre banda, l'ADC no esborra les mostres i n'agafa de noves fins que el bucle de programa calcula la FHT. Aleshores, si l'ADC té 256 mostres al buffer però el bucle de programa no està a l'inici, l'ADC estarà deixant d'adquirir mostres del senyal fins que no s'hagi arribat a l'inici de bucle i calculat la FHT. Aquest fet té conseqüència en la representació espectral del senyal a on apareixeran salts.

L'algorisme de detecció inserit a l'Arduino s'explica en el capítol 6. Aquest algorisme fa servir el mètode RD mig anivellat definit en el capítol 3 que consisteix en comparar cada component freqüencial del senyal amb cada patró. Si l'algorisme detecta un accident, llavors el valor de la variable Acc pren el valor 20,2.

5.2.3 Comunicació i enviament de paquets entre node i coordinador

Per a la comunicació entre node i coordinador s'ha usat la llibreria <RF24Network.h> que permet configurar el mòdul NRF24. La configuració que s'ha de definir són les adreces dels nodes, el canal, la velocitat d'enviament de les dades i els paràmetres de reenviament de paquets.

Dispositiu	Mode	Temps sleep (%)	Adreça origen	Adreça destí	Velocitat transmissió	Canal	Reenviament paquets
Node sensor	Transmissor	99%	01	00	250kbps	90	11 repeticions, 2ms
Coordinador	Receptor	80%	00	01	250kbps	90	-

Taula 11. Configuració dels NRF24

Per a la comunicació s'ha dissenyat una xarxa bàsica amb un node sensor i un coordinador. La comunicació directe s'estableix especificant en número octal l'adreça del node que es programa i un número octal de l'adreça del node a on es vol enviar. El canal ha d'estar configurat entre 1 i 125. La velocitat d'enviament de dades la fixem al mínim de 250kbps mitjançant la configuració "radio.setDataRate(RF24_1MBPS)", ja que és suficient per el tipus d'informació que s'ha d'enviar i, segons el fabricant, permet mantenir la distància de transmissió fins als 900 metres (sensibilitat de -104dBm).

Mentre no s'ha d'enviar cap dada al coordinador, és a dir no s'ha detectat cap accident i no s'ha arribat a la iteració 100.000 per a enviar les dades de bateria i placa solar, el mòdul NRF24 es manté en mode "sleep" mitjançant la comanda "radio.powerDown()". En el moment en què s'ha de transmetre la informació, es desperta amb la comanda "radio.powerUp()" i tarda 1,5ms per estar llest per enviar.

Per tal de que el mòdul NRF24 del coordinador estigui un temps adormit, es configura l'opció "radio.setRetries()" al NRF24 del node sensor que permetrà reenviar el paquet si el coordinador no l'ha rebut. S'ha establert un màxim de 11 reenviaments amb un temps d'espera entremig de 2ms. Per tant, durant 22ms s'intentarà enviar el paquet.

Variable	Tipus	Posició en la tupla	Informació	Origen	Destí	Bytes
Acc	float	1	Tipus accident	Node	Coordinador	4
BatS	float	2	Voltatge bateria Node S	Node	Coordinador	4
SolS	float	3	Voltatge placa solar Node S	Node	Coordinador	4
Id	float	4	Número de node S	Node	Coordinador	4

Taula 12. Variables generades al node sensor i a enviar al coordinador

El paquet de dades a enviar al coordinador s'empaqueta en una tupla amb totes les variables. Les variables arribaran amb NRF24 del coordinador amb l'ordre en què estan posades en la tupla. Per identificar quin node és alhora d'enviar l'avís d'accident s'ha creat la variable Id de tipus float i que emmagatzema el nombre de node 10,1. Les altres variables de la tupla són del tipus float també: BatS (voltatge bateria), SolS (voltatge placa solar) i Acc (tipus d'accident). Si apareix un accident es guardarà el valor 20,2 a la variable Acc de tipus float. S'hi apareix un accident, en el paquet només s'enviarà la variable accident, i les altres variables agafaran el valor 0,0. En la iteració que toqui enviar les variables bateria i solar, si no s'ha detectat accident la variable accident tindrà el valor 0,0.

Preàmbul	Adreça destí	Control de paquet	Payload màxim	CRC	Longitud màxima
1 Byte	3-5 Bytes	2 Bytes	32 Bytes	1-2 Bytes	42 Bytes

Taula 13. Longitud de les trames del NRF24

La trama que envia el mòdul transmissor NRF24 consta de varies parts. El preàmbul serveix per avisar als nodes receptors del mateix canal que s'està enviant un nou missatge. La seqüencia del preàmbul pot ser o bé 01010101 o bé 10101010. Seguidament, els nodes receptors que han detectat el preàmbul comproven que l'adreça de destí coincideixi amb la seva. Si va dirigit a ell, procedeix a descodificar el missatge. El "payload" que s'enviarà del node sensor al coordinador contindrà el valor de les variables Acc, BatS, SolS i Id. Aquests són totes de tipus float per tant la mida del "payload" és de 16 Bytes, i la del missatge és de 26 Bytes.

Velocitat transmissió (kbps)	Bytes/s	Bytes/ms	Bytes a enviar	Temps enviament (ms)
250,00	31.250,00	31,25	26,00	0,83
1.000,00	125.000,00	125,00	26,00	0,21

Taula 14. Velocitat de transmissió i temps d'enviament del NRF24

L'anterior taula compara el temps d'enviament a partir de la velocitat de transmissió que es configura en el mòdul NRF24 i per una determinada mida de la trama a enviar. Per a la velocitat configurada de 250kbps es tardaran 0,83ms a enviar el missatge. Si l'integrat tarda 1,5 segons a despertar-se del mode "standby", el temps màxim que trigarà a enviar el missatge serà de 2,33ms.

La programació del mòdul NRF24 es realitza a dins del mateix programa que corre a l'Arduino, és a dir, l'Arduino envia les comandes al mòdul NRF24 cada vegada que es necessari que aquest realitza alguna tasca. Les variables del nivell de bateria, placa solar i situació d'accident es poden enllaçar directament a les comandes d'enviament del mòdul NRF2. S'ha d'incloure la llibreria <SPI> per tal de comunicar l'Arduino amb el mòdul NRF24 mitjançant els pins MOSI, MISO, SCK i CS del dos dispositius.

5.3 Coordinador

La feina d'un node coordinador s'assemblaria a la que fa un router convencional. En el cas que s'instal·lessin més nodes sensors, el coordinador seria l'encarregat de rebre per un mateix canal les dades de fins a 6 nodes diferents. Llavors, el coordinador hauria de decidir a on enviaria aquesta informació, si a altres nodes, a un altre coordinador o a un servidor de la xarxa. En el present projecte només s'ha desenvolupat una prova pilot d'un node i un coordinador, aleshores la funció d'aquest coordinador és enllaçar les dades del node sensor amb un servidor.

Seguidament es mostra el diagrama de flux del programa del coordinador. Aquest programa té l'objectiu principal de mantenir el coordinador els mòduls ESP8266 i NRF24 el màxim temps possible adormits per tal d'estalviar energia. El NRF24 resta adormit 20ms (80%) i es desperta durant 5ms per escoltar si arriba algun nou missatge. Per fer-ho possible s'ha de configurar la funció "radio.setRetries()" al node sensor.

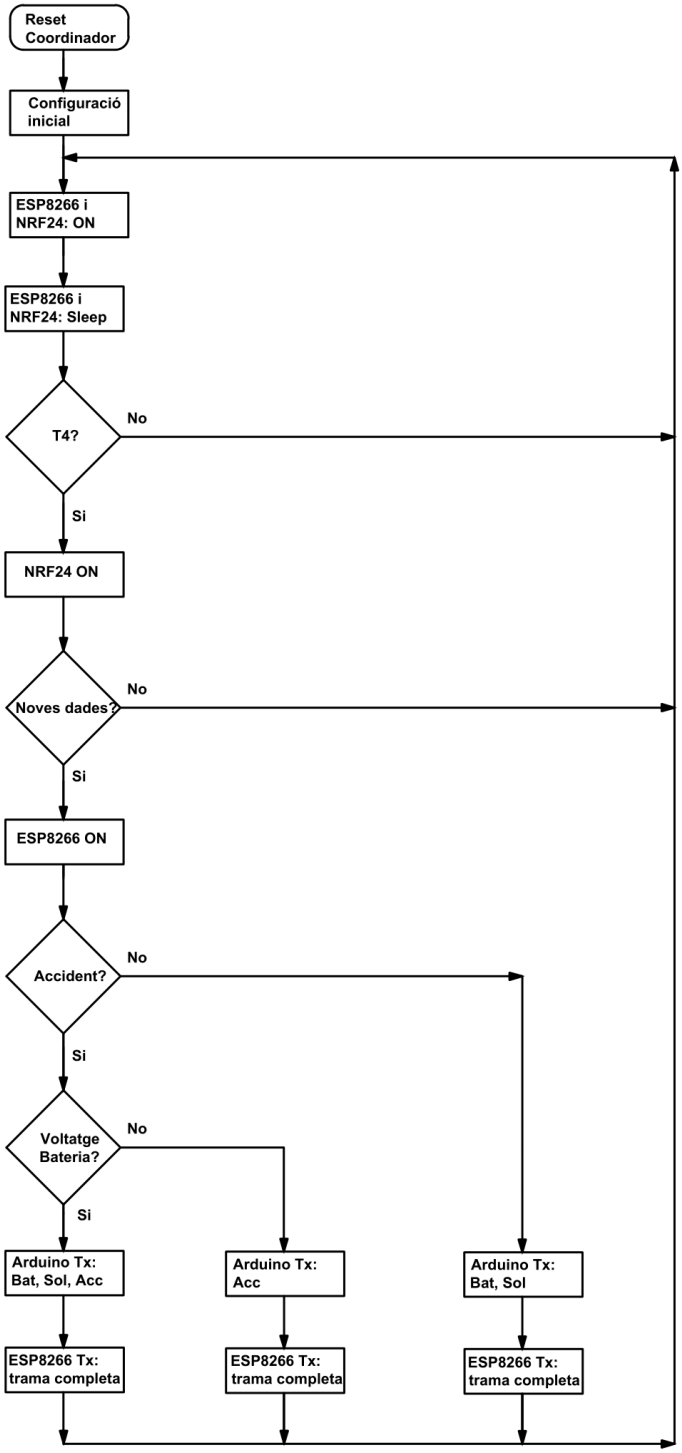

Figura 32. Diagrama de flux del funcionament del coordinador

5.3.1 Detecció del nivell de bateria i d'energia solar

De la mateixa manera que el node sensor, el coordinador també adquireix les dades del voltatge de la bateria i del voltatge que proporciona la placa solar. Per al voltatge de la bateria

s'usa la referencia interna de 1,1V que té l'Atmega328, i es guarda a la variable BatC de tipus float.

Per altre banda, el voltatge de la placa solar es realitza a través del pin analògic 1 de l'Arduino Pro Mini. S'ha integrat un divisió de tensió i la mesura analògica es fa en el centre del divisió de tensió. Aquest voltatge de la placa solar es guarda a la variable SolC de tipus float. Aquestes dues variables s'enviaran juntament amb les variables que es rebin del node sensor i amb una freqüència de 30 minuts.

5.3.2 Comunicacions del coordinador: NRF24 i ESP8266

El coordinador té dues vies de comunicació amb altres dispositius. Una és mitjançant el mòdul NRF24 que porta incorporat, i l'altre és a través del ESP8266. La recepció de les dades al dispositiu coordinador es realitza mitjançant el mòdul NRF24. Per altre banda, l'enviament de dades del coordinador al servidor Thingspeak es realitza a partir del ESP8266.

El coordinador ha d'estar atent a qualsevol missatge d'avís d'accident que pot rebre del node sensor. Ara bé, és necessari adormir el mòdul NRF24 perquè en mode receptor consumeix 40mA. L'opció de reenviament de paquets que té el mode Enhance Sockburst permetrà que el NRF24 del coordinador es pugui adormir durant un temps. Per adormir el NRF24 s'utilitza la instrucció "radio.powerDown()" i el seu consum descendeix a 10uA. El temps que el NRF24 del coordinador resta adormit és de 4 iteracions (20ms) i es desperta durant 1 iteració (5ms). El temporitzador T4 és el que comparà les repeticions de bucle, i que cada una equival a 5ms. Com el que el NRF24 del node sensor té l'ordre de reenviar el paquet durant 22ms en el cas que no l'hagi rebut el coordinador, aleshores el temps de "sleep" del coordinador és inferior a aquest temps de reenviament, conseqüentment el coordinador rebrà el paquet amb total probabilitat.

Dispositiu	Mode	Temps sleep	Tipus comunicació	Velocitat transmissió
Arduino	Transmissor	0%	UART	115.200
ESP8266	Receptor	99%	UART	115.200

Taula 15. Característiques de la comunicació entre l'Arduino i el ESP8266

Una vegada el NRF24 del coordinador ha rebut el paquet del node sensor, seguidament l'Arduino ha de transmetre aquest paquet al mòdul ESP8266 mitjançant la comunicació sèrie (UART). Per fer-ho, abans s'ha de despertar el mòdul ESP8266. Activant el pin digital número

3 de l'Arduino que està connectat al pin CH de l'ESP8266, s'aconsegueix que l'ESP8266 entri en el mode "wake up" i estigui llest per rebre dades pel port sèrie. La velocitat de transmissió de la comunicació es fixa a un valor alt, 115.200bps, perquè el mòdul WiFi pugui adormir-se el més aviat possible.

El ESP8266 només és despertat per l'Arduino del coordinador quan el NRF24 rep dades, per tant cada 30 minuts o bé quan es produeix un accident. L'ESP8266 es desperta i està llest per enviar un paquet en 2ms (segons dades del fabricant), ara bé, necessita establir la connexió a la xarxa WiFi que normalment sol ser de 2 segons com a màxim.

Variable	Tipus	Informació	Origen	Destí	Bytes
Acc_s	string	A + Acc	Coordinador	ESP8266	4
BatS_s	string	B + BatS	Coordinador	ESP8266	4
SolS_s	string	S + SolS	Coordinador	ESP8266	4
Id_s	string	I + Id	Coordinador	ESP8266	4
BatC_s	float to string	C + BatC	Coordinador	ESP8266	4
SolC_s	float to string	T + SolC	Coordinador	ESP8266	4

Taula 16. Variables del coordinador a enviar al ESP8266

Les proves de recepció dels paquets emesos pel NRF24 del node sensor i rebuts pel NRF24 del coordinador han estat satisfactòries. Cada variable és descodificada per el NRF24 del coordinador en l'ordre en què es s'ha guardat en la tupla del paquet que envia el NRF24 del node sensor. Per altre banda, el ESP8266 alhora de rebre les dades via sèrie de l'Arduino no aconsegueix que totes les recepcions siguin correctes ja que algunes vegades capgires els valors de les variables. Per tant, una vegada rebut el paquet al coordinador, s'ha optat per convertir les variables en tipus string i seguidament inserir un identificador a davant dels caràcters de cada variable. L'identificador permetrà reconèixer cada variable que arribi a l'ESP8266 mitjançant el primer caràcter de la variable. Per a la codificació de les variables s'ha afegit el caràcter "A" davant de la variable Acc, "B" a la variable BatS, "S" a la SolS, "I" a la Id, "C" a la BatC i "T" a la SolC.

Start	Dada	Stop	Total bits
1 bit	8 bits	2 bits	11 bits

Taula 17. Longitud de la trama de la comunicació per port sèrie

Velocitat transmissió (bps)	Variables a enviar	Tipus variable	Bits/variable	Bits a enviar	Temps enviament (ms)
115.200	6	string	32	192	1,67

Taula 18. Velocitat de transmissió i temps d'enviament de la comunicació sèrie

Les trames que s'envien via comunicació sèrie estan formades per 11 bits. Per a la velocitat fixada de 115.200bps es tarda 1,67ms per enviar les 6 variables de tipus float. Per tant, quan arriba un nou missatge al coordinador, el temps que es tarda en despertar el mòdul ESP8266 i enviar-li les trames és de 3,67ms. A aquest temps s'hi ha d'afegir el que necessita per connectar-se a la xarxa WiFi, que pot arribar a ser de 2.000ms. Per tant, des que es reben les dades al coordinador fins que estan penjades al servidor poden passar uns 2.000ms.

Per a utilitzar el mòdul l'ESP8266 s'ha fet servir la llibreria <ESP8266Wifi.h>. El programa es es guarda al mòdul ESP8266 (disposa de memòria i processador) a través de l'IDE Arduino. Per a la correcta connexió amb la xarxa WiFi s'indica l'identificador SSID i contrasenya del router, així com l'adreça del servidor Thingspeak i la seva contrasenya per bolcar-hi dades.

5.3.3 Servidor Thingspeak

El servidor ThingSpeak és un servei que ofereix gratuïtament l'empresa MathWorks i que permet afegir dades fàcilment a través de la seva API. Per a emmagatzemar-hi dades solament és necessari indicar el número de canal i contrasenya. A cada canal es pot crear vuit camps i per tal d'inserir-hi informació s'ha de citar el prefix "Field" i el número de variable.

Per tal de transferir les dades del mòdul ESP8266 al servidor Thingspeak s'ha indicat en el programa del ESP8266 el canal número 103215 i la contrasenya del canal. Seguidament, s'ha enllaçat el camp "Field1" a la variable Id, el "Field2" a Acc, el "Field3" a BatS, el "Field4" a SolS i el "Field5" a BatC i el "Field6" a SolC. A partir d'aquí, el ESP8266 enviarà dades al servidor quan l'Arduino del coordinador li ordeni.

5.4 Cronograma de tasques

Mitjançant un cronograma es pot visualitzar les tasques que realitza cada dispositiu així com el temps que ocupa cadascuna d'elles. El primer cronograma correspon a les tasques del node sensor i l'aparició d'un accident. També pertany al node sensor el segon cronograma, amb la particularitat de l'aparició de la necessitat de recollir les dades de bateria i placa solar.

Els dos últims cronogrames representen les tasques del coordinador i apareixen els mateixos esdeveniments que en els cronogrames del node sensor.

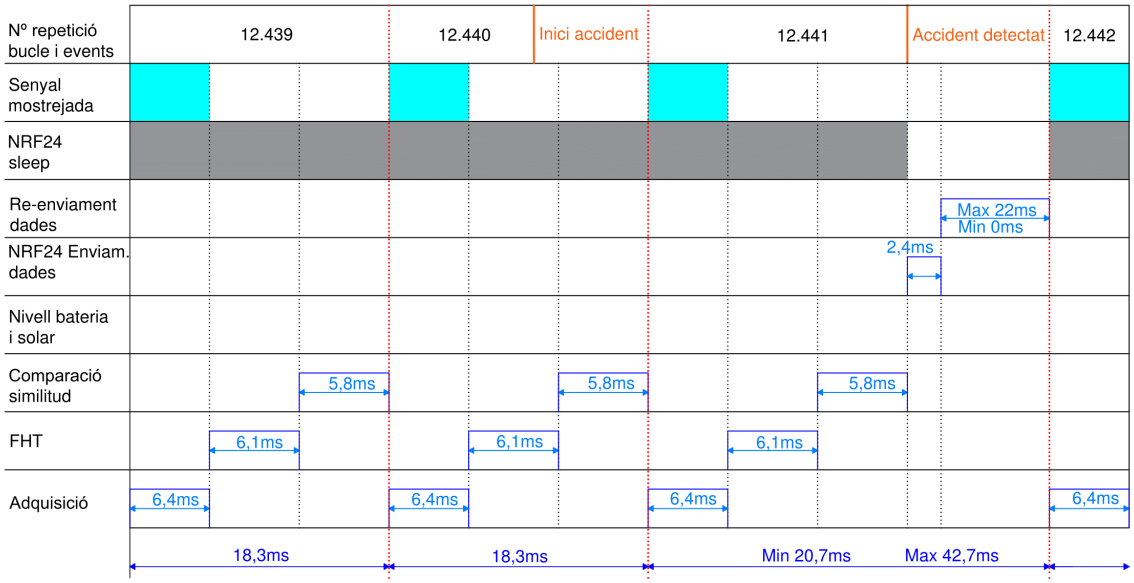

Figura 33. Cronograma de tasques del node sensor amb l'aparició d'un accident

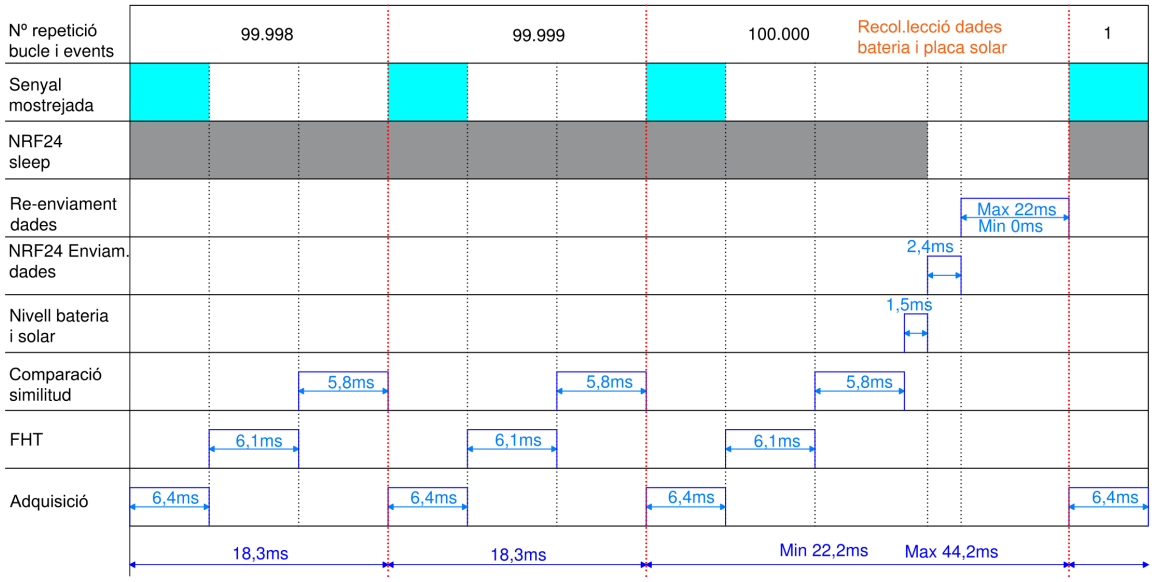

Figura 34. Cronograma de tasques del node sensor amb l'aparició de recollida de dades

En el primer cronograma del node sensor s'observa l'aparició d'un accident i seguidament es desperta el mòdul NRF24 per tal d'enviar les dades al coordinador. El senyal no es mostreja de manera continua, sinó només quan finalitza cada bucle amb la tasca de comparació de similitud. Encara que l'adquisició es realitza de forma paral·lela mitjançant interrupcions, aquesta no pot guardar nous valors fins que FHT processa els que hi ha actualment. Pel que fa al segon cronograma, on es va analitzant el so però no es detecta res, quan s'arriba al bucle

número 100.000 apareix l'esdeveniment de recollir dades de bateria i placa solar. Aleshores es desperta el mòdul NFR24 per enviar aquestes dades i es reinicia el comptador de bucles. Un aspecte a remarcar és quan el NRF24 envia dades, torna a reenviar el paquet fins que no rep l'acusament de recepció del coordinador, amb un límit de 11 repeticions i 2ms entre repetició.

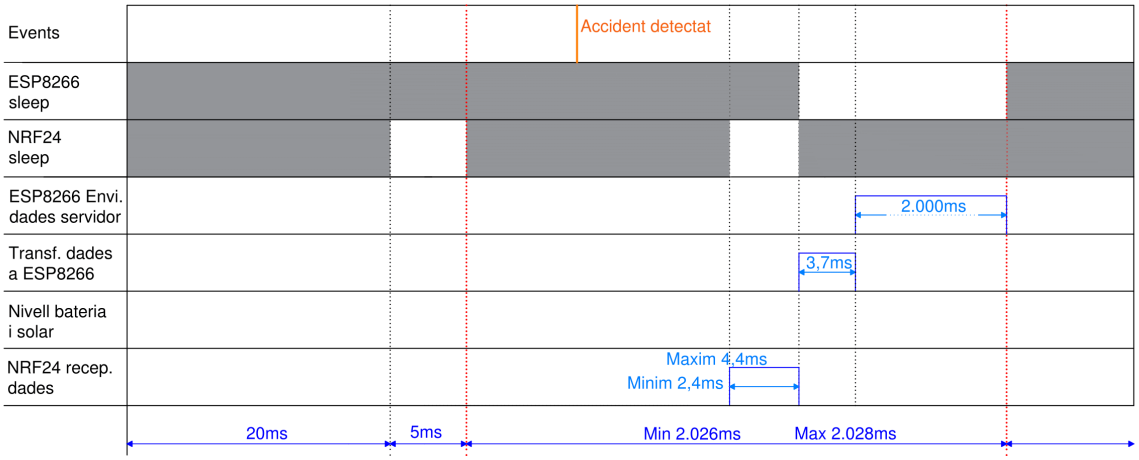

Figura 35. Cronograma de tasques del coordinador amb l'aparició d'un accident

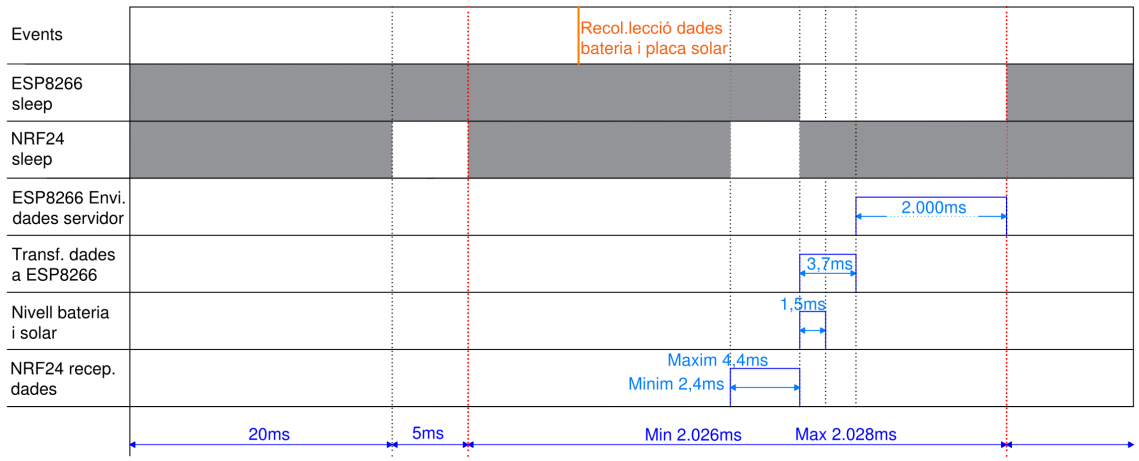

Figura 36. Cronograma de tasques del coordinador amb l'aparició de recollida de dades

En el primer cronograma del coordinador es veu representat el cicle d'adormir-se i despertar-se que realitza el mòdul NRF24. Aquest es desperta per veure si arriben dades del node sensor. Si el node sensor envia dades mentre el coordinador està dormint, el reenviarà fins durant 22ms (11 repeticions), conseqüentment amb 22ms el coordinador s'haurà despertat ja que ho fa cada 20ms. Quan el NRF24 del coordinador ha rebut el paquet, llavors es desperta el mòdul ESP8266 i l'Arduino li envia les dades. Finalment, l'ESP8266 ha de transmetre el paquet al servidor ThingSpeak. Si el paquet de dades que arriba al coordinador és per informar d'un accident, llavors no cal recollir les dades de bateria i solar del coordinador.

6. ALGORISME DE DETECCIÓ AMB ARDUINO

En el següent capítol s'analitzen els aspectes generals de l'Arduino que condicionen l'algorisme així com la metodologia per a detectar si el so és un accident.

6.1 Limitacions Atmega328

Per detectar la similitud dels sons que capta el micròfon és necessari dissenyar un algorisme eficient que s'adapti als recursos limitats de l'Atmega328. Aquest algorisme ha de realitzar les instruccions seqüencialment, ja que l'Arduino no permet executar rutines en paral·lel. El que si que permet l'Arduino són les interrupcions, per Hardware o programades, que consisteixen en interrompre el programa principal per efectuar una petita tasca i seguidament retornar al punt del programa a on s'havia aturat. La part d'adquisició i conversió es realitza per interrupcions, cada cop que arriba un valor nou al buffer de l'ADC s'interromp el programa per tractar el valor i emmagatzemar-lo. Ara bé, la implementació d'interrupcions en el sistema de detecció de similitud sembla inviable. L'objectiu real és realitzar la FHT el màxim ràpid possible per poder passar a calcular la següent finestra espectral, i fer-ho amb interrupcions suposaria retardar aquest càlcul.

La memòria de dades SRAM de l'Arduino és de 2kB, conseqüentment el nombre de variables i vectors serà limitat. Degut a que es farà servir memòria de dades per altres tasques com l'adquisició, l'obtenció de l'espectre freqüencial, monitorització de bateria i per a la comunicació sense fils, la memòria de dades que quedarà per a l'algorisme de detecció serà del 42% (uns 850 Bytes). El patró conté l'espectre freqüencial de l'Àudio Accident que es vol buscar al senyal, per tant ha de disposar de les components freqüencials de 100Hz a 10.000Hz. Ara bé, aquest rang de freqüències es pot limitar més, ja que a partir de 8.000Hz no s'observa energia en els espectres de l'apartat 3.1. Per una matriu 128 punts que representa l'espectre freqüencial del senyal, la component freqüencial número 50 representa a la freqüència 7.900Hz. Aleshores, la matriu de cada patró serà de [49][0] de tipus long, i ocuparà 50x1=50 Bytes.

Una altre limitació de l'Atmega328 és la velocitat de processament de 16MHz de l'Arduino Pro Mini. En el l'apartat 8.1 es pot observar el temps de bucle que tarda l'Arduino Pro Mini a realitzar totes les tasques d'adquisició, càlcul FHT i algorisme de detecció de similitud. Si el temps de bucle és molt llarg, provoca que apareguin salts importants en la representació espectral del senyal. S'intenta no fer servir variables float ja que alenteixen la velocitat al

requerir més cicles de rellotge. Conseqüentment, l'índex RD_{migA} no està en tant per 1 sinó en tant per 100.

6.2 Detecció de similitud

La detecció de similitud comença una vegada l'Arduino ha calculat la FHT de 256 mostres. Els patrons estan inserits a la memòria de l'Arduino i estan representats amb valors freqüencials d'un instant de temps del senyal. La comparació de similitud es realitza mitjançant el mètode de relació de distàncies mig anivellat (RD_{migA}) entre cada punt de l'espectre del vector FHT del senyal i cada punt dels vectors del patró 1 i del patró 2. No s'ha utilitzat cap dels mètodes MSC perquè l'algorisme FHT de la llibreria Arduino retorna els valors freqüencials en magnitud (mòdul) i no en component real i complexa, i a més a més perquè la fórmula de la relació de distancies requereix menys instruccions i espai de dades.

L'algorisme que ha de detectar la similitud s'ha dissenyat amb el mínim d'instruccions possible per tal de minimitzar el temps de bucle. S'identifiquen 3 parts principals de l'algorisme. La primera part realitza l'anivellació de les components freqüencials del senyal amb les components freqüencials dels patrons. Pel que fa a la segona, calcula l'índex RD_{migA} que consisteix en determinar la mitjana aritmètica de relació de distàncies entre les components freqüencials i els patrons. Finalment la tercera compta les vegades que coincideixen el patró 1 i patró 2 i el temps que hi ha entre ells. El resultat final de l'algorisme és una variable booleana que és verdadera si es compleixen les condicions de la tercera part, en cas contrari és falsa. Al final del bucle es reinicien totes les variables utilitzades.

En un primer moment es va dissenyar l'algorisme sense la part d'anivellació, ara bé, després de realitzar proves es va veure necessari d'afegir una part que anivelli el senyal amb els patrons. Degut a que la font emissora del so pot estar a diferents distàncies, l'amplitud de les components freqüencials del senyal variarà. Per altre banda, l'amplitud de les components freqüencials dels patrons és fixe, aleshores és necessari adequar les components freqüencials del senyal per tal que siguin comparables amb les dels patrons. Això permetrà que la detecció de similitud pugui efectuar-se de manera independent a la distància.

Respecte la tercera part, la del compliment de condicions, tampoc estava a l'algorisme en els primers moments, ara bé, es van introduir condicions respecte al nombre de coincidències i respecte a la duració per evitar que en un determinat instant de temps algun possible so aleatori coincideixi amb els patrons.

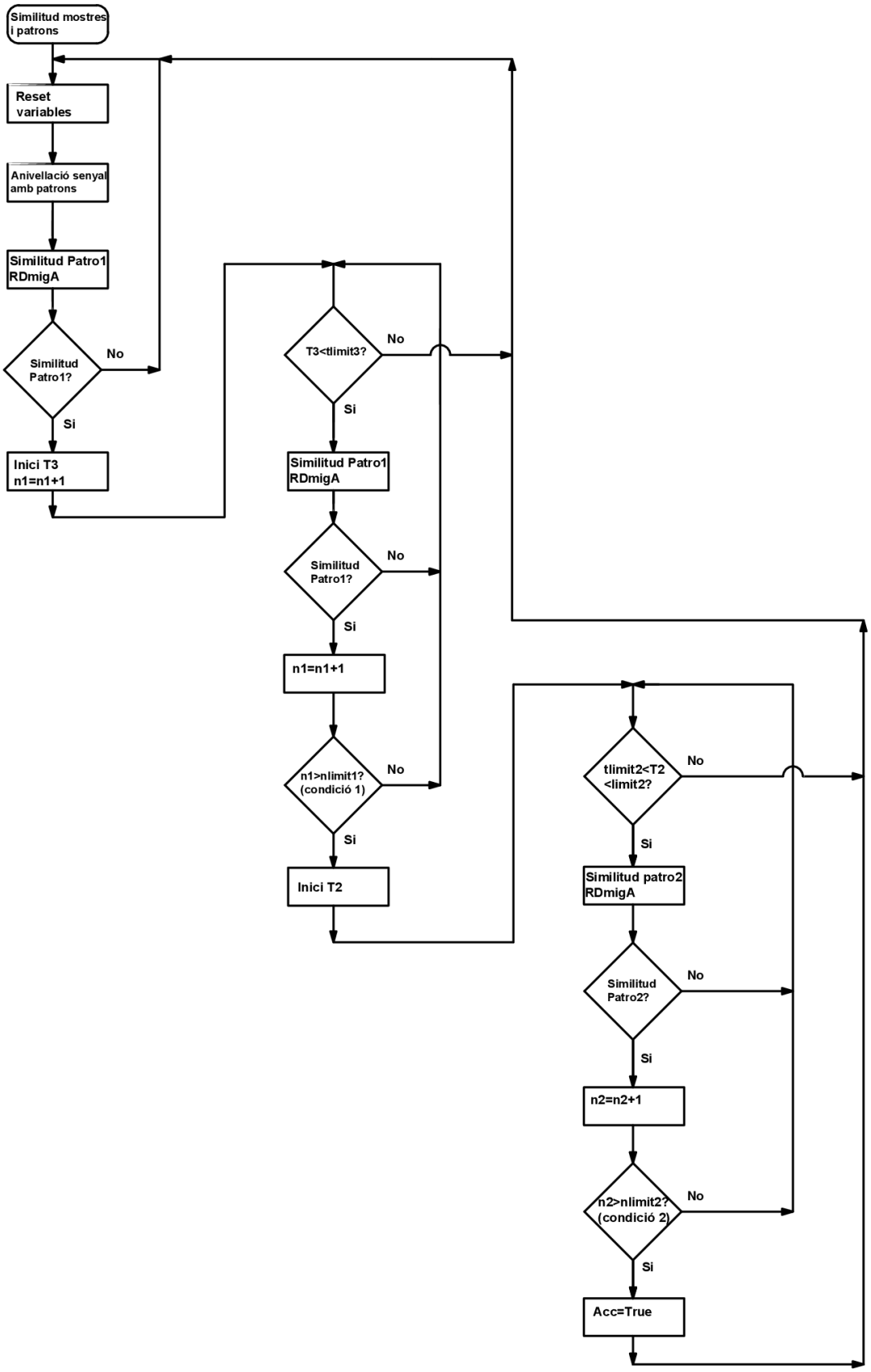

Figura 37. Diagrama de flux de l'algorisme de detecció de similitud

El diagrama de flux anterior representa el funcionament general de l'algorisme de detecció de similitud. S'han utilitzat varies variables per a càlculs d'entremig i temporitzadors.

Variable	Tipus	Valor	Descripció	Memòria ocupada (Bytes
fht_lin_out	int	Vector 128	Espectre freqüencial senyal adquirit	256
patro1	long	Vector 50	Espectre freqüencial patró 1	200
patro2	long	Vector 50	Espectre freqüencial patró 2	200
mx1	long	Variable	Valor màxim espectre	4
mx2	long	Variable	Valor sub-màxim espectre	4
pmx1	long	Constant	Mitjana valors màxims fht_lin_out	4
equiv1	long	Variable	Relació pmx1 i mx1	4
dist1	long	Variable	Diferencia valors p1	4
dist2	long	Variable	Diferencia valors p2	4
dist1total	long	Variable	Suma valors p1	4
dist2total	long	Variable	Suma valors p2	4
rel1	long	Variable	Relació diferencia suma p1	4
rel2	long	Variable	Relació diferencia suma p2	4
rel1acum	long	Variable	Relacions tot l'espectre p1	4
rel2acum	long	Variable	Relacions tot l'espectre p2	4
limit1	int	Constant	Límit per complir condicions	2
limit2	int	Constant	Límit per complir condicions	2
n1	int	Variable	Coincidències p1	2
n2	int	Variable	Coincidències p2	2
nlimit1	int	Constant	Límit per complir condicions	2
nlimit2	int	Constant	Límit per complir condicions	2
t2	int	Variable	Temporitzador 2	2
tlimit1	int	Constant	Límit per complir condicions	2
tlimit2	int	Constant	Límit per complir condicions	2
inicifrenada	bool	Variable	Si es detecta frenada	1
frenada	bool	Variable	Si es detecta la condició de frenada	1
accident	float	Variable	Si es detecta accident	4
enviat	bool	Variable	Si s'ha enviat	1
t3	int	Variable	Inici similitud patró 1	2
tlimit3	int	Variable	Límit per complir condicions	2

Taula 19. Variables utilitzades en l'algorisme de detecció de similitud

Tot seguit s'explica l'algorisme amb la nomenclatura de les variables. L'algorisme de detecció de similitud s'inicia un cop s'obté el vector fht_lint_out de la transformada FHT. La primera part, l'anivellació consisteix en buscar els dos valors màxims (mx1 i mx2) del vector del fht_lint_out. Una vegada obtinguts mx1 i mx2, es realitza la mitjana d'aquests dos valors i es guarda a mx1. Aquest procediment no cal fer-lo per els vectors freqüencials dels patrons, ja

que la mitjana dels valors màxims dels 2 patrons ja estarà calculada des de l'inici i es guardarà a la variable pmx1. Posteriorment es divideix pmx1 entre mx1 i es guarda a equiv1. La variable equiv1 serveix per anivellar, aleshores es multiplica cada component freqüencial del vector del senyal per aquesta variable equiv1. El resultat obtingut és el vector de freqüències del senyal anivellat a la mitjana dels valors màxims dels patrons.

La part de calcular l'índex RD_{migA} comença calculant l'índex RD entre cada component freqüencial del senyal i del patró 1, i posteriorment pel patró 2, per guardar-ho a la variable rel1 i rel2, respectivament. Es realitza els índexs RD per a tots els elements dels vectors freqüències, i s'acumulen a la variable rel1acum i rel2acum. Aleshores es pot realitzar la mitjana dels RD acumulats, obtenint RD_{migA} des 2 patrons i guardant-ho a les variables rel1acum i rel2acum.

La part final consisteix en analitzar si es compleixen unes condicions. Les variables limit1, limit2, tlimit1 i tlimit2 són les que supediten les condicions. Si el valor de rel1acum (que correspon al RD_{migA}) està per sobre del valor que guarda la variable limit1 es considera que el patró 1 coincideix amb el senyal. Cada vegada que hi ha coincidència amb el patró 1 s'incrementa la variable n1. També, cada vegada que el valor de rel2acum està per sobre de limit2 el patró 2 coincideix amb el senyal i s'incrementa la variable n2. Per a detectar si el senyal és un accident s'han de complir dos condicions. La condició 1 és que n1 contingui un nombre de coincidències més gran a nlimit1. La condició 2 que n2 tingui coincidències per sobre de ntlimit2.

A part de complir-se les condicions, és necessari que aquestes es compleixin en un temps determinat. El temporitzador T3 (variable t3) s'inicia en la primera coincidència del patró 1 amb el senyal. Aleshores, la condició 1 ha de complir-se abans que T3 no arribi a tlimit3. Si s'arriba a tlimit3 sense haver-se complert la condició 1, es reinicien totes les variables, impossibilitant el compliment de les condicions. De similar manera s'ha de complir la condició 2. El temporitzador T2 (variable t2) s'inicia quan s'ha complert la condició 1, aleshores la condició 2 s'ha de complir sempre a dins d'un interval de temps, entre tlimit1 i tlimit2. És a dir, si es compleix la condició 1, hi haurà un marge de temps (tlmit2-tlimit1) per a què el patró 2 coincideixi amb el senyal.

El temporitzador T3 serveix per evitar que s'acumulin les similitud del patró 1, així només es poden acumular durant una duració de tlimit3. S'ha introduït T2 perquè la duració d'un accident

no sol ser de més de 4 segons, així s'evita que la condició 2 es compleixi després de molts de temps que s'hagi complert la condició 1.

Les variables limit1, limit2, nlimit1, nlimit2, tlimit1, tlimit2,tlimit3 que fan que es compleixin les condicions s'han obtingut després d'analitzar quan funciona l'algorisme davant de proves amb el so d'un accident de trànsit.

7. INSTALACIO EXTERIOR

La caixa exterior que ha d'albergar a cada dispositiu cal que compleixi les condicions d'impermeabilitat a l'aigua i resistència a temperatures altes. Els graus de protecció IP podria ser un indicador descriptiu de les característiques de seguretat que la caixa exterior ha de tenir. L'IP44, que estableix que el producte ha de impedir l'entrada de partícules amb diàmetre mínim de 1mm i d'aigua llançada a xorro durant 5 minuts com a mínim, podria ser una bona descripció de la caixa exterior dels dispositius.

Per tal que el funcionament del dispositiu node sensor sigui correcte, aquest ha de disposar d'un orifici que permeti l'entrada de les ones sonores. A més ha de tenir una fixació per a la placa solar que possibiliti la rotació a diferents orientacions.

La distància ideal a instal·lar cada node hauria de ser de 300 metres, ja que permetria a cada node detectar un accident so a un màxim de 20 metres. Per tal de reduir el nombre de dispositius, la distribució dels nodes hauria de ser de manera radial i amb el coordinador al centre. Ara bé, si no és possible la distribució dels nodes de forma radial o la distància d'algun node amb el coordinador és de més de 300 metres s'hauria d'instal·lar més d'un coordinador.

La ubicació dels nodes amb micròfons ha d'estar repartida en les zones i carreteres més transitades de les ciutats, per tal de detectar amb eficàcia possibles accidents. Agafant d'exemple la ciutat de Girona, i la carretera principal del Carrer Barcelona amb una distància de 2.800 metres, suposaria una instal·lació d'uns 60 nodes i 9 coordinadors. Els coordinadors haurien d'estar a prop d'una xarxa WiFi pública o s'hauria d'implementar la tecnologia GPRS al dispositius coordinadors.

8. PROVES, SIMULACIONS I RESULTATS

Per comprovar el correcte funcionament del hardware dissenyat així com del programari desenvolupat ha estat necessari realitzar varies proves. Les proves d'adquisició i representació freqüencial mostren la qualitat del so que s'adquireix i la qualitat del càlcul de l'espectre de freqüències, respectivament. A partir de les proves amb l'algorisme de detecció de similitud s'observa la fiabilitat per a detectar sons d'accidents. Les proves i simulacions amb sons d'accidents han estat realitzades a dins d'un edifici i a les distàncies que es mencionen a cada sub-apartat.

8.1 Proves d'adquisició i de representació freqüencial

Durant el desenvolupament del projecte s'han realitzat varies proves per verificar el micròfon, l'etapa de pre-amplificació i l'espectre freqüencial del senyal que calcula l'Arduino. La prova ha consistit en emetre so des d'un telèfon mòbil amb una freqüència fixa de 4.000Hz. Per a generar aquest so s'ha utilitzat una aplicació mòbil de generació de freqüències de so.

8.1.1 Sensibilitat del micròfon

El micròfon i l'etapa de pre-amplificació han de captar el senyal analògic amb la màxima amplitud possible i amb el mínim soroll. En la següent imatge veiem la captació d'un so de emès a 1 metre del micròfon.

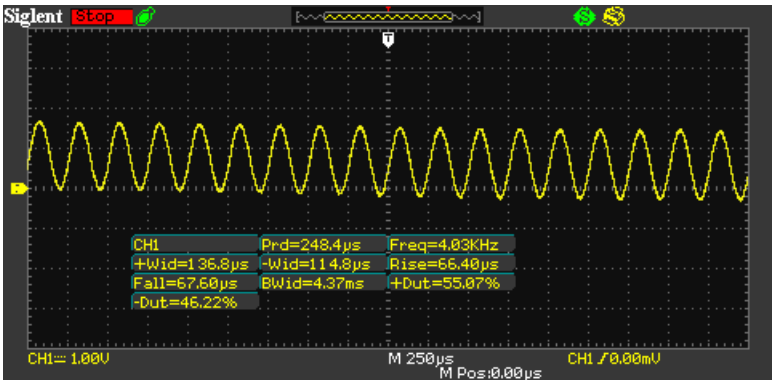

Figura 38. Captació del so de 4.000Hz des d'una distància d'1 metre

Es pot observar que el senyal és sinusoïdal i la seva freqüència és de 4.030Hz, per tant, d'igual freqüència a la que s'ha emès des del telèfon mòbil.

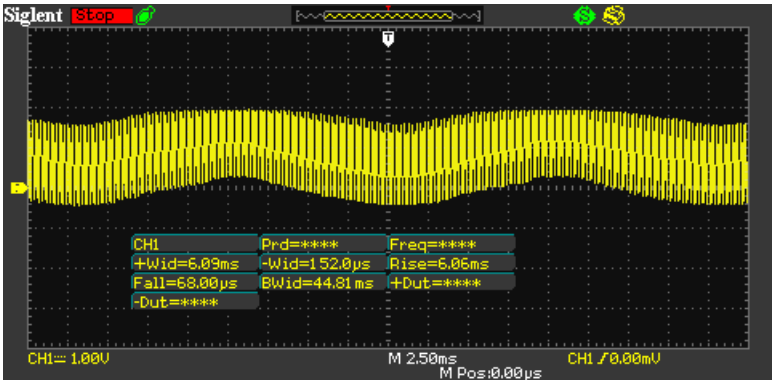

Figura 39. Captació del so de 4.000Hz des d'una distància metre. Envolvent

La primera imatge solament s'hi veia l'harmònic de 4.000Hz. En aquesta darrera imatge es pot veure l'evolvent, que té una freqüència de 50Hz, i seria la freqüència de la xarxa que entra per a la connexió USB que alimenta a l'Arduino.

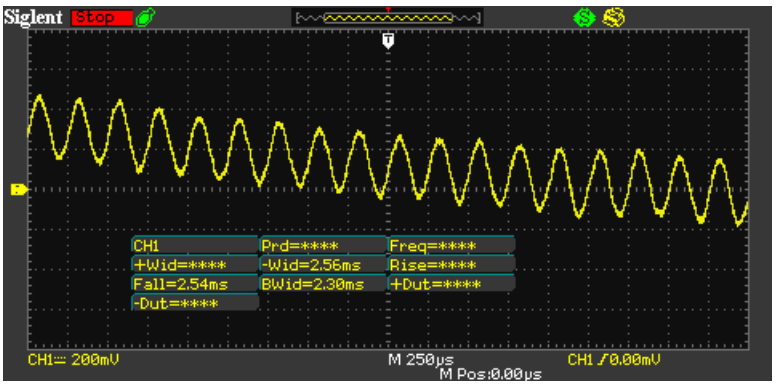

Figura 40. Captació del so de 4.000Hz des d'una distància de 5 metres

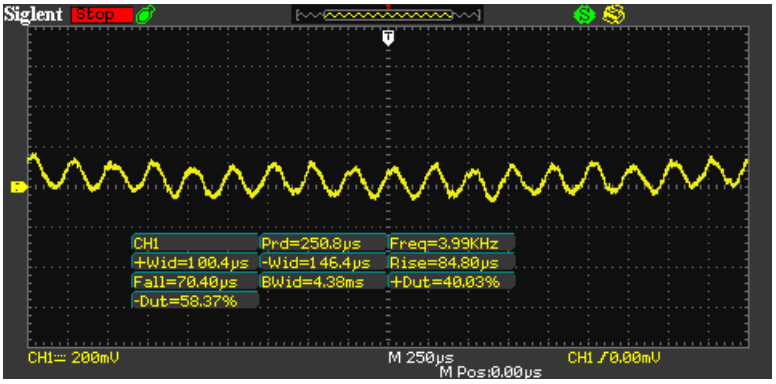

Figura 41. Captació del so de 4.000Hz des d'una distància de 10 metres

Les dues últimes imatges mostren el senyal capturat per una distància de 5 i 10 metres. Als 5 metres l'amplitud és de uns 250mV i per a 10 metres de 180mV.

8.1.2 Salts entre adquisicions

L'adquisició de dades es realitza a una freqüència de 40.000Hz. Quan l'ADC disposa d'un nou valor en el buffer aleshores s'executa una interrupció per Hardware. Cada vegada que l'ADC té un valor seguidament el microprocessador deixa de fer el que estava fent i procedeix a guardar el valor de l'ADC en una matriu. D'aquesta manera, indiferentment del què hi hagi en el bucle principal, l'adquisició amb l'ADC sempre s'està realitzant. Ara bé, degut a que s'ha de realitzar el càlcul de l'espectre de freqüències (FHT) i altres operacions per a la comparació del senyal amb els patrons, el senyal queda representat en l'espectre freqüencial de manera discontinua.

Per tal d'esbrinar la duració del bucle principal s'ha instaurat una marca al final del bucle. Cada vegada que s'arriba al final del bucle s'activa una sortida digital durant 100us i llavors s'apaga. Així doncs si es captura aquesta sortida digital en un oscil·loscopi es pot observar el canvi d'un estat baix a un estat alt, que correspon a la finalització del bucle, i el temps que tarda a repetir-se.

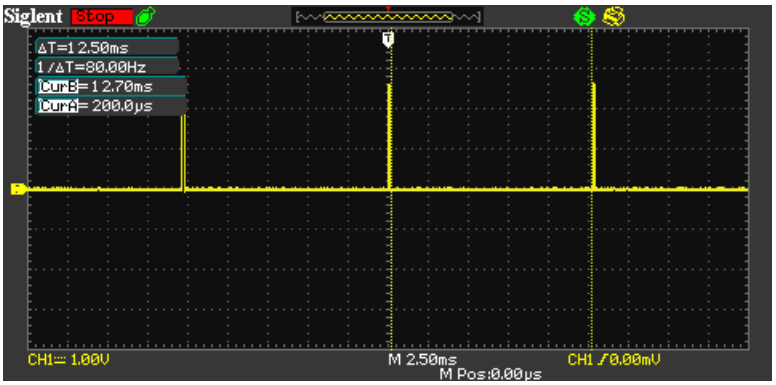

Figura 42. Temps de repetició entre bucles amb adquisició i FHT

L'anterior imatge mostra el bucle amb l'adquisició per interrupció i el càlcul de la FHT. La duració del bucle és de 12,5ms. El programa no inclou l'algorisme de detecció de similitud. El càlcul de la FHT serien casi la totalitat d'aquests 12,5ms, ja que l'adquisició es bastant molt més ràpida.

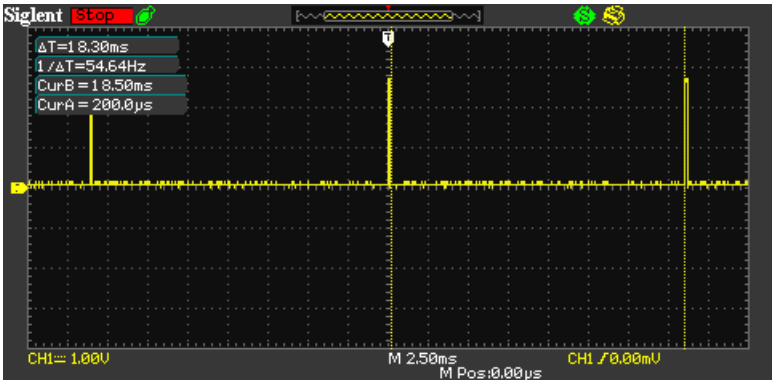

Figura 43. Temps de repetició entre bucles amb adquisició, FHT i detecció de similitud

Aquesta darrera imatge representa el temps de bucle amb l'adquisició per interrupció, el càlcul de la FHT i l'algorisme de detecció de similitud. Si a cada 18,3ms es realitza la transformada FHT dels valors de l'ADC, aleshores el senyal queda representat espectralment per finestres de 6,4ms i llavors té un salt de 11,9ms (18,3ms menys 6,4ms) fins a la pròxima finestra de 6,4ms.

Bucle principal - Atmega328 16MHz	Temps bucle (ms)	Salt adquisicions (ms)
FHT (ADC per interrupció)	12,50	6,10
Algorisme detecció i FHT (ADC per interrupció)	18,30	11,90

Taula 20. Temps entre bucles

En l'apartat 3.3, on s'analitzava la MSC mig entre els patrons i les finestres del senyal, s'ha inserit salts de 19,3ms entres les finestres, i el MSC mig ponderat ha continuat mostrant similitud. Amb els salts es perd informació, ara bé, si aquest salt no és molt elevat el senyal encara queda ben representat espectralment.

8.1.3 Qualitat de les dades de l'ADC

Per visualitzar la qualitat del senyal que captura l'ADC i la precisió dels components electrònics de la placa s'han enregistrat les dades de l'ADC per al so Àudio Accident i després sense cap so (silenci). S'ha comunicat l'Arduino i el PC mitjançant el port sèrie i les dades s'han enviat a partir de la funció "Serial.Print". Cal remarcar que al imprimir pel port sèrie les dades es fa alentir el temps de bucle, que pot ascendir fins a 50ms i provocarà que el senyal estigui capturat en menys mostres. Les dades que captura l'ADC de l'Arduino s'han representat espectralment mitjançant la FFT de Matlab en el gràfics següents.

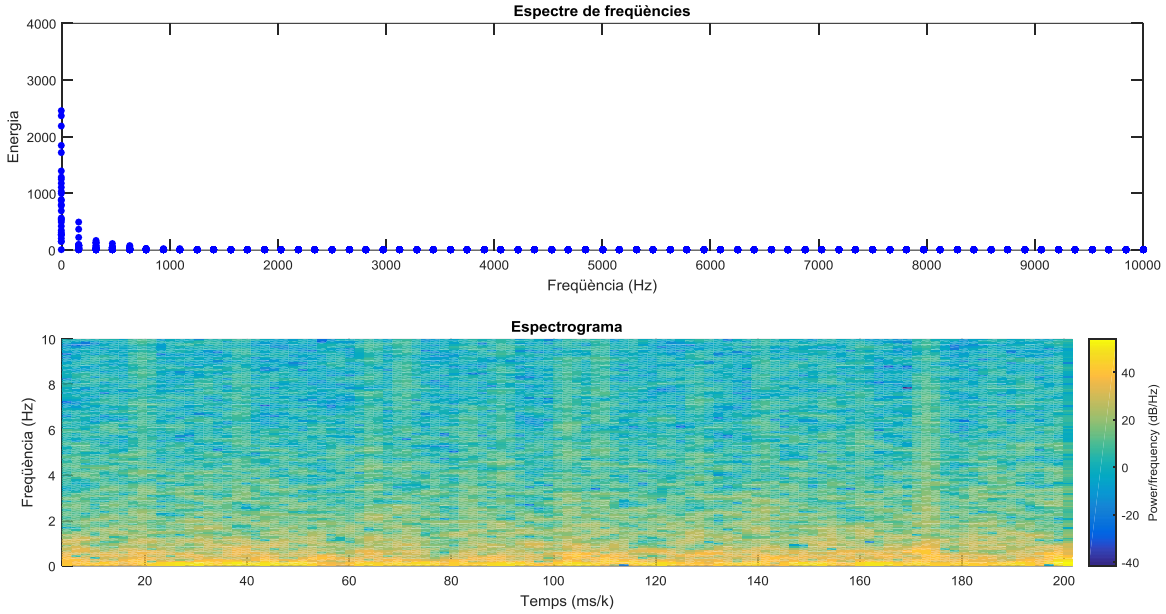

Figura 44. Espectre de freqüències i espectrograma del silenci. Calculat a partir de les dades de l'ADC

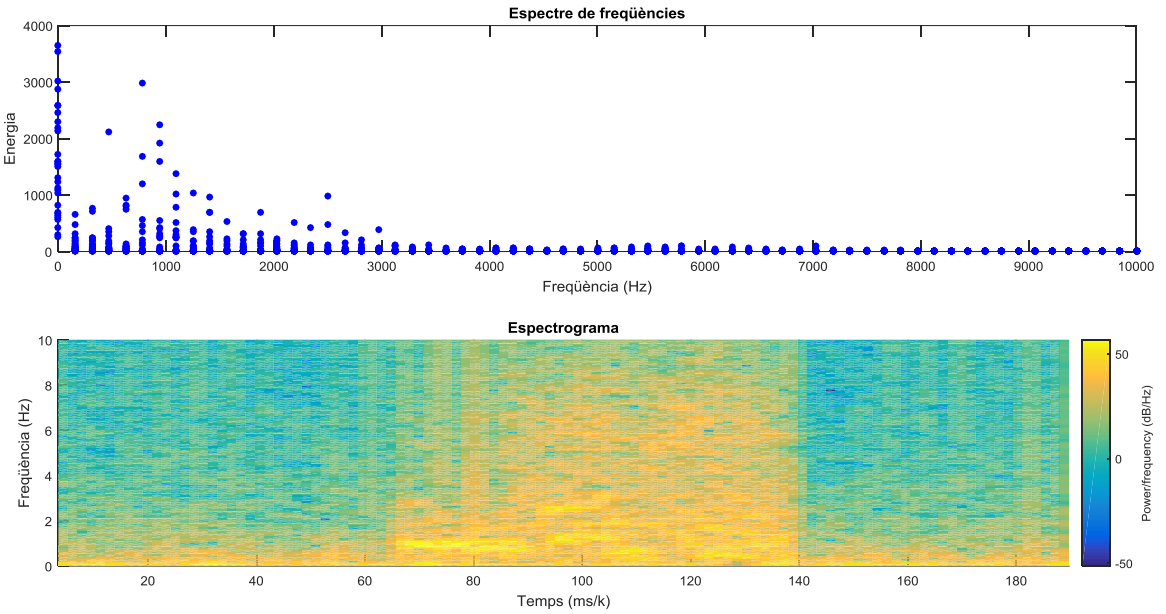

Figura 45. Espectre de freqüències i espectrograma de l'Àudio Accident. Calculat a partir de dades de l'ADC

Observem que per el cas silenci apareix soroll en el rang de freqüències de 0 a 500Hz. Aquest rang de freqüències del soroll no hauria d'afectar als patrons, ja que no tenen components freqüencials en aquest rang. L'espectre freqüencial calculat amb Matlab a partir dels valors adquirits amb l'ADC de l'Arduino és bastant semblant a l'espectre freqüencial calculat amb Matlab, de l'apartat 3.1, a partir dels valors de l'arxiu Àudio Accident. En tots dos espectres apareixen pics d'energia per les freqüències de 0 a 2.000Hz. De manera semblant, si mirem a l'espectrograma de les dades capturades per l'AD de l'Arduino, es pot veure que té una distribució semblant al que s'ha calculat amb Matlab directament a partir de l'arxiu Àudio

Accident. Ara bé, la resolució no és la mateixa, ja que al haver-hi salts de 50ms hi ha menys mostres capturades i per tant es perd informació del senyal.

8.1.4 Qualitat de la transformada FHT

Per tal d'esbrinar la qualitat de l'espectre que retorna la llibreria FHT de l'Arduino i alhora la precisió dels components electrònics de la placa, s'han obtingut dades de l'espectre calculat sense cap so (silenci) i després aplicant el so Àudio Accident. Les dades que obtenim s'han guardat al PC mitjançant la comunicació sèrie que permet l'Arduino.

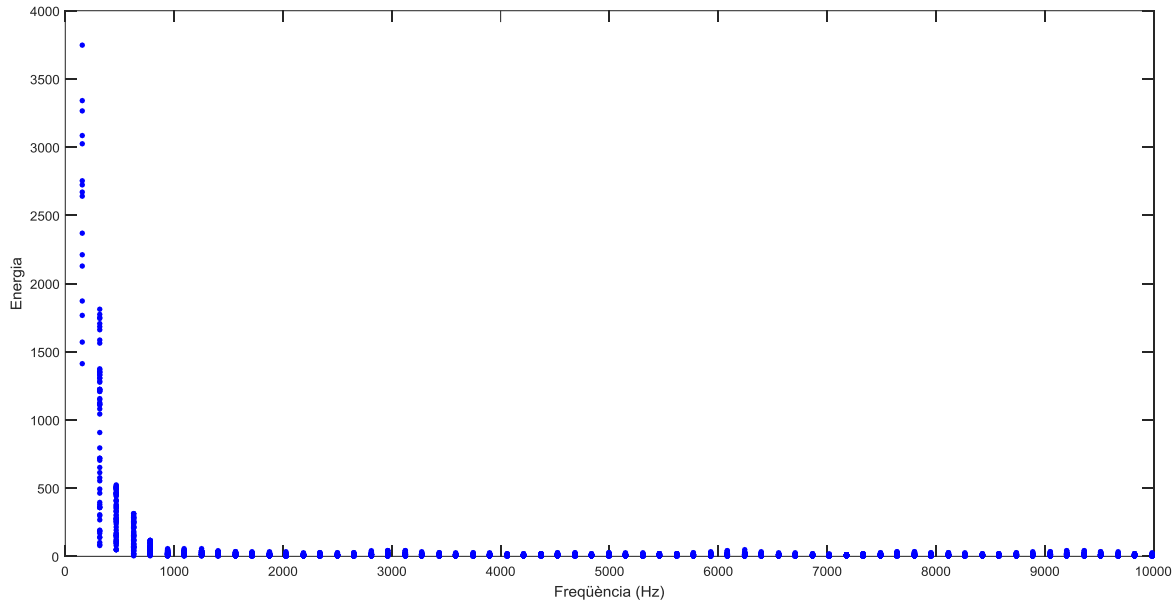

Figura 46. Espectre de freqüències del silenci calculat amb l'algorisme FHT de l'Arduino

El temps del bucle (amb el càlcul de la FHT) ha augmentat dels 18ms als 100ms, ja que per enviar les dades pel port sèrie el microcontrolador ha de realitzar varis passos, a més dels necessaris per calcular la FHT. En aquest primer gràfic veiem representat el soroll que té el micròfon i l'etapa de pre-amplificació en el moment de silenci, corresponent a la franja de 0-500Hz. Aquest soroll pot esser degut a la proximitat del dispositiu node amb al PC per tal de realitzar la transferència de dades a través del port sèrie.

L'espectre freqüencial de l'Àudio Accident que retorna la llibreria FHT està representat en el següent gràfic. S'hi observa de manera similar la distribució de components freqüencials que s'ha analitzat amb el Matlab a l'apartat 3.1, amb l'energia sonora concentrada en el rang de 100-5.000Hz.

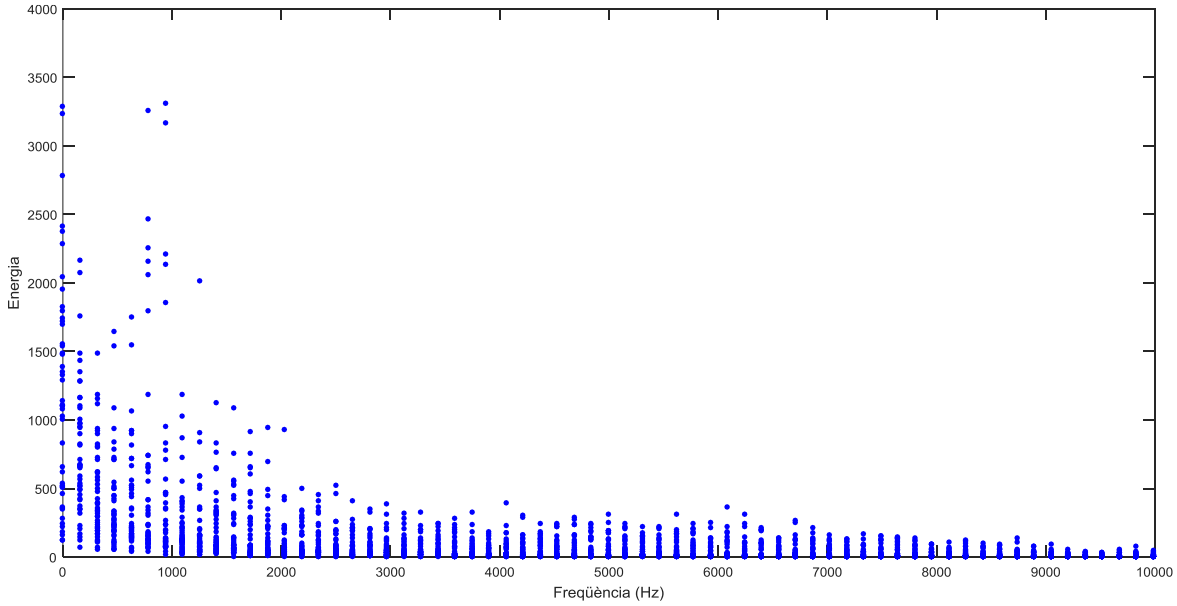

Figura 47. Espectre de freqüències del silenci calculat amb l'algorisme FHT de l'Arduino

8.2 Simulacions de detecció de similitud

Aquest apartat inclou les simulacions de detecció de similitud a partir de sons reproduïts amb un telèfon mòbil convencional. Per tal d'observar els resultats de l'algorisme de detecció d'accidents s'ha valorat quina opció provocava un menor augment del bucle de programa. S'ha descartat visualitzar els resultats a través del port sèrie de l'Arduino, ja que es necessita molts recursos, i per tant molts de cicles de rellotge, per a transferir dades pel port sèrie. Aleshores s'ha optat per la visualització a través de l'oscil·loscopi, configurant en el mateix algorisme l'activació de les sortides digitals de l'Arduino que requereixen de molts pocs cicles de rellotge. La sortida digital 2 s'activa durant 0,1ms si es detecta el patró 1, seguidament si el nombre de vegades que s'ha detectat el patró 1 és més gran al valor de la variable nlimit1 (condició 1) i el temps que ha passat des de la primera detecció del patró 1 és inferior a la variable tlimit3, s'activa durant 1ms la sortida digital 2. Si es detecta el patró 2 s'activa durant 0,1 ms la sortida digital 3, i si el nombre de vegades és més gran que la variable nlimit2 i alhora el temps en què s'ha complert la condició 1 és d'entre tlimit1 i tlimit2 (condició 2) aleshores s'activa la sortida digital 3 durant un pols llarg de 200ms.

S'han realitzat simulacions de la detecció de similitud per a dues distàncies d'emissió. La primera simulació s'ha realitzat a 2 metres del micròfon del node sensor i per al so Àudio Accident i seguidament pel so Àudio música. Posteriorment s'ha realitzat la segona simulació a 5 metres del micròfon pels mateixos tipus de sons.

8.2.1 Simulació a 2 metres

El mòbil s'ha situat a 2 metres del node sensor i amb l'altaveu enfocant el micròfon. El volum de reproducció dels sons ha estat a la meitat de la potència sonora que proporciona el mòbil. S'ha realitzat la simulació a l'interior d'un habitatge i en els moments en què no hi ha cap tipus de soroll extern. Les constants de configuració que actuen com a limitadors de les condicions a complir per a detectar similitud queden resumides a la taula següent:

Variable	limit1	limit2	nlimit1	nlimit2	tlimit1	tlimit2	tlimit3
Definició	Límit RD_{migA} patró 1	Límit RD_{migA} patró 2	Repeticions patró 1	Repeticions patró 2	Temps condició 2	Temps condició 2	Temps condició 1
Valor	22	45	5	4	45	150	50
Condició			Condició 1	Condició 2			
Temps (ms)					824	2.745	915
Visualització	Pols 0,1ms - Groc	Pols 0,1ms - Blau	Pols 200ms - Groc	Pols 200ms - Blau	Supedita a condició 1	Supedita a condició 1	Supedita a condició 2

Taula 21. Constants de configuració per a les condicions de similitud. Distància de 2 metres

En la següent imatge hi ha representats els resultats de l'algorisme de detecció de similitud per a l'Àudio Accident. Es pot veure com el senyal coincideix amb els patrons (pols 0,1ms) i com es compleixen les condicions de similitud (pols de 200ms). Els polsos grocs corresponen al patró 1 i els polos blaus al patró 2.

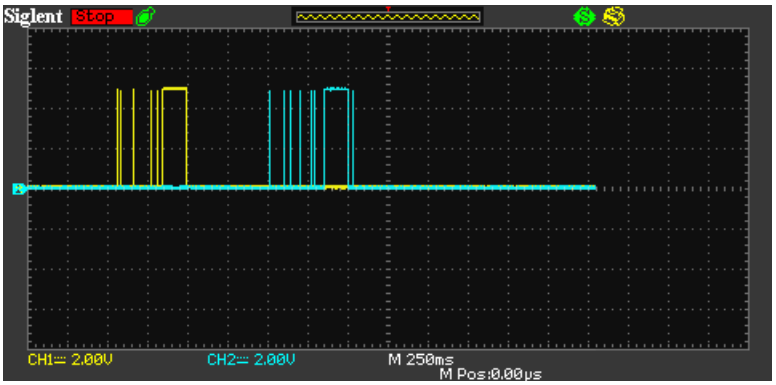

Figura 48. Detecció de similitud del l'Àudio Accident a 2 metres

Seguidament s'augmenta el rang de visualització de l'oscil·loscopi per tal de veure més resultats alhora. El so Àudio Accident s'ha deixat en reproducció repetitiva, és a dir, al finalitzar l'àudio es torna a repetir.

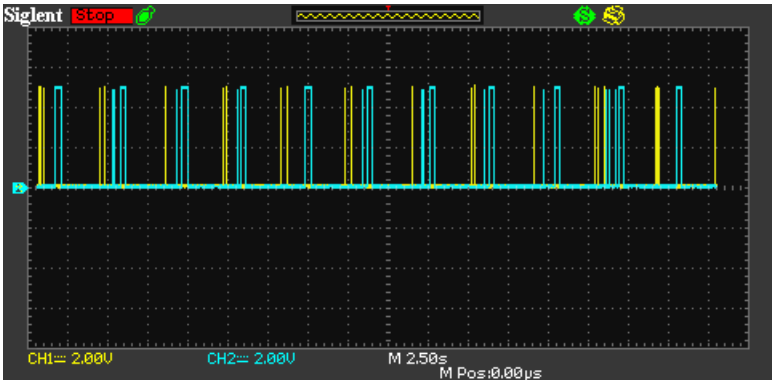

Figura 49. Detecció de similitud del l'Àudio Accident a 2 metres. Mode repetició

A partir d'aquesta simulació, inclosa, s'ha substituït el pols groc de 200ms (condició 1) per un pols de 0,1ms, ja que en aquest instant el so de l'accident està a la meitat i cal evitar alentir el bucle del programa de l'Arduino. En la imatge anterior es veu el resultat de repetir l'Àudio Accident i executar l'algorisme de detecció de similitud és favorable per una distància de 2 metres. De 11 repeticions del so Àudio Accident l'algorisme detecta similitud en totes 11 repeticions (el pols blau més llarg indica la detecció).

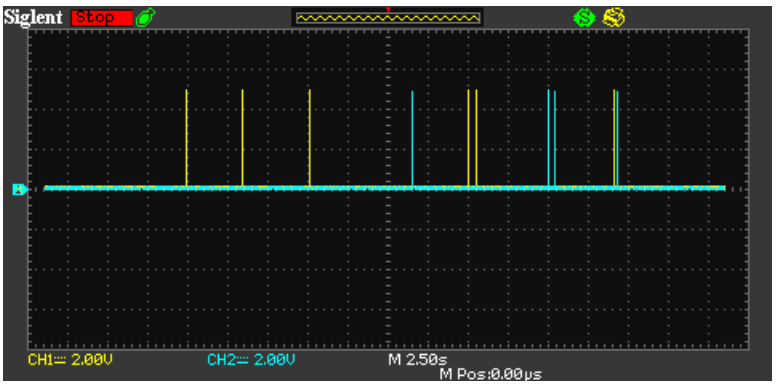

Figura 50. Detecció de similitud del l'Àudio música a 2 metres

Aquesta última imatge mostra la reproducció de l'Àudio música emès a una distància de 2 metres del micròfon. L'algorisme de detecció de similitud troba alguna condicència amb els patrons, ara bé, no es compleix alhora les dues condicions en cap moment.

8.2.2 Simulacions a 5 metres

De la mateixa manera que en l'apartat anterior, s'han realitzat simulacions de detecció de similitud a una distància de 5 metres. Primer ha estat necessari realitzar petites proves per tal de calibrar les constants de configuració. Al canviar de distància respecte a l'apartat anterior,

s'ha de veure l'efecte que provoca la introducció de l'anivellament d'amplitud de les components freqüencials. Aquest funcionalitat ha de permetre que la detecció de similitud no estigui afectada per la distància en què s'emet el so.

Variable	limit1	limit2	nlimit1	nlimit2	tlimit1	tlimit2	tlimit3
Definició	Límit RD_{migA} patró 1	Límit RD_{migA} patró 2	Repeticions patró 1	Repeticions patró 2	Temps condició 2	Temps condició 2	Temps condició 1
Valor	22	44	5	4	50	170	40
Condició			Condició 1	Condició 2			
Temps (ms)					915	3.111	732
Visualització	Pols 0,1ms - Groc	Pols 0,1ms - Blau	Pols 0,1ms - Groc	Pols 200ms - Blau	Supedita a Condició 1	Supedita a Condició 1	Supedita a Condició 2

Taula 22. Constants de configuració per a les condicions de similitud. Distància de 5 metres

S'ha executat dues vegades l'algorisme de detecció de similitud. Una per al so Àudio Accident i l'altre per l'Àudio música.

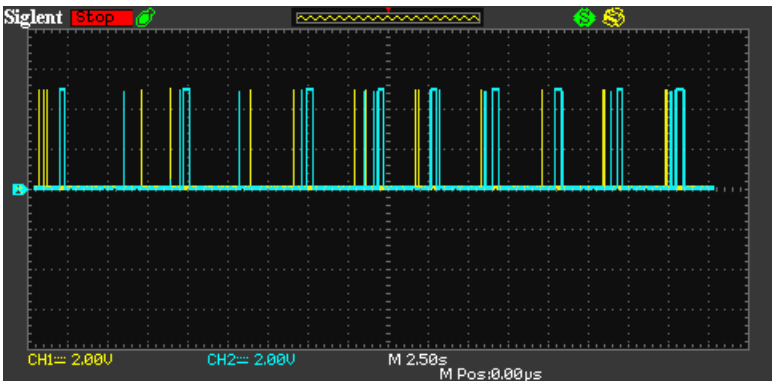

Figura 51. Detecció de similitud del l'Àudio Accident a 5 metres. Mode repetició

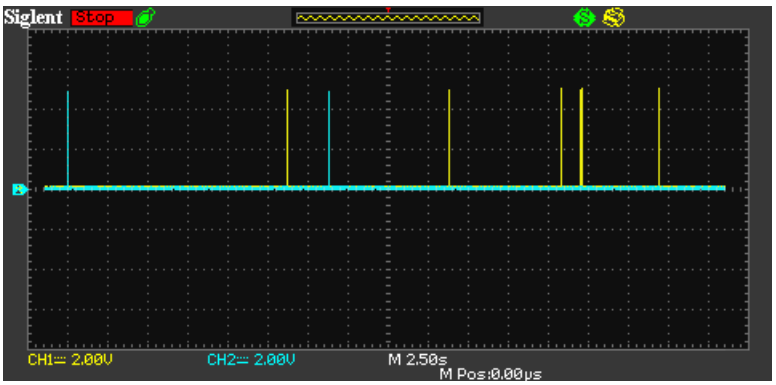

Figura 52. Detecció de similitud del l'Àudio música a 5 metres. Mode repetició

El resultat de la simulació per a l'Àudio Accident a 5 metres ha canviat respecte a la distància anterior. De 11 repeticions de l'Àudio Accident es detecten 9 similituds. Per a la simulació de

l'Àudio música a 5 metres el resultat és semblant que el resultat obtingut a 2 metres. No es detecta cap similitud d'accident, ara bé, si que coincideixen els patrons en diferents instants de temps.

8.2.3 Valoració final de les simulacions de detecció de similitud

Ta com s'ha especificat al principi de l'apartat, les simulacions s'han realitzat amb condicions més favorables de les que hauria de tenir el dispositiu en l'aplicació real. Les simulacions s'han realitzat a dins d'un habitatge i sense sorolls externs. També, l'altaveu emissor del so s'ha enfocat directament al micròfon del node sensor.

Pel que fa a les constants de configuració, ha estat necessari re-ajustar-les quan s'ha canviat de distància d'emissió. El resultat obtingut ha estat favorable per a l'Àudio Accident, ja que s'ha detectat similitud en casi totes les repeticions. Ara bé, aquest canvi de constants no permet valorar amb claredat la importància de l'anivellament d'amplituds al canviar de distàncies.

El tipus de so en les simulacions ha estat el mateix, l'Àudio Accident i l'Àudio música. Caldria realitzar simulacions amb altres arxius de diferents característiques i a diferents volums de potència sonora.

Les respectives simulacions per a les dues distàncies reflecteixen que l'algorisme inserit a l'Arduino efectua un bon tractament del senyal i la comparació de similitud és òptima. Ara bé, és necessari realitzar més simulacions que englobin canvis de distancies, canvis en el volum del so, introducció de soroll de fons, i més condicions que poden afectar al funcionament real de l'algorisme.

8.3 Proves de comunicació

Conèixer els rangs de funcionament dels dispositius és imprescindible per poder preveure el seu comportament. La distància en què els mòduls NRF24 i ESP8266 estableixen comunicació correctament és imprescindible perquè determina la distribució exterior dels nodes i del coordinador.

8.3.1 Prova de comunicació entre node i coordinador

Per tal de validar la comunicació entre el node sensor i el coordinador s'ha programat un algorisme simple. S'ha inserit el programa general de configuració i enviament de dades a l'Arduino del node sensor sense incloure l'algorisme de detecció d'accidents. El paquet de dades que envia aquest algorisme simple és similar al que s'utilitza a l'algorisme general de detecció d'accidents, amb una mida de 26 Bytes, i la velocitat de transmissió també es fixa idèntica a l'algorisme de detecció d'accidents, que correspon a 250kbps. El node sensor enviarà cada 1.000ms el mateix paquet, amb les variables BatS, SolS, Acc i Id de tipus float, ara bé amb els valors assignats directament com a constants. El programa que conté les instruccions d'enviament està inserit en l'Arduino, per tant és necessari 2 mòduls NRF24 i dos Arduino per establir la comunicació.

Per tal de saber si arriba correctament el paquet al coordinador s'ha programat en l'Arduino del coordinador l'activació cada 500ms d'una sortida digital durant 500ms. A la sortida digital s'hi ha connectat momentàniament un LED. Per tant, fins que es comuniqui el node amb el coordinador el LED farà intermitència. Segons el fabricant, la distància de comunicació a l'aire lliure és de 900 metres.

El mètode d'allunyament dels dos dispositius ha consistit en deixar el node sensor estàtic en un punt, i amb el coordinador a la mà començar a caminar en línia recta. La distància s'ha estimat aproximadament a través del comptatge dels passos, amb l'equivalència d'1 pas igual a 1 metre. La taula següent resumeix els resultats d'aquesta prova.

Dispositiu	Distància (m)	Element de confirmació	Intermitència LED
Coordinador	25	LED	Sí
	50	LED	Sí
	100	LED	Sí
	200	LED	Sí
	300	LED	Sí
	350	LED	No

Taula 23. Distàncies de comunicació entre NRF24

A partir dels 300 metres el mòdul NRF24 del coordinador comença a no rebre correctament els paquets enviats pel mòdul NRF24 del node sensor. L'experiment realitzat només comprova que el paquet hagi arribat. Una prova més acurada consistiria en descodificar i comprovar el contingut dels paquets.

8.3.2 Prova de comunicació entre coordinador i servidor Thingspeak

Al llarg de l'execució del projecte s'han efectuat proves de comunicació del node sensor amb al coordinador per tal de monitoritzar la bateria i la placa solar a través del servidor ThingSpeak. No s'ha realitzat un estudi exhaustiu del rendiment de la placa solar, sinó que les proves han servit per veure que el driver de control de la bateria funciona correctament i la placa solar aporta energia a la bateria.

Les dades de nivell de bateria i de nivell d'energia solar del node sensor han estat enviades al coordinador a través dels mòduls NRF24, i el coordinador ha pujat les dades al servidor ThingSpeak. El canal de dades utilitzat és públic i es pot consultar al portal de Thingspeak.com a partir del número d'identificació del canal, que és el 103215.

8.4 Test de consum

El test de consum s'ha realitzat pel node sensor. El mòdul NRF24 resta adormit si no s'ha d'enviar cap missatge o sinó s'ha produït cap accident. L'Arduino, el micròfon, l'etapa de pre-amplificació i el driver TP4056 estan engegats sempre en el node sensor.

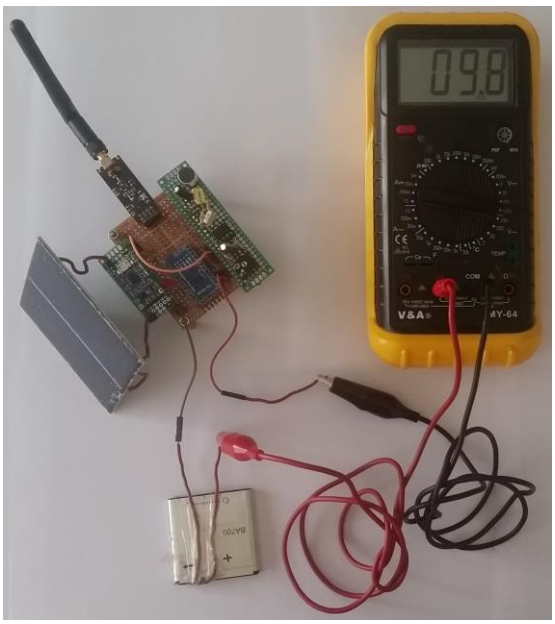

Figura 53. Consum d'energia del node sensor amb el NRF24 dormint

El consum del node sensor és de 9,8mA. Aquesta prova mostra el consum del node sensor per una situació concreta, la que consumeix menys. Les altres situacions, en què s'envien

dades, el consum s'eleva però la seva duració és curta. Tal i com s'indica en l'apartat 4.4, el node sensor esdevé autònom si s'alimenta a partir d'una placa solar i una bateria. Pel coordinador el consum seria una mica més alt, ja que el mòdul NRF24 està adormit menys de temps que el del node sensor. D'igual manera, el coordinador també és autònom amb una placa solar i una bateria.

9. RESUM DEL PRESSUPOST

El projecte dels sensors d'àudio per a la detecció d'accidents urbans ascendeix a un cost de dos mil cent noranta-nou euros amb quaranta-set cèntims, sense IVA.

10. CONCLUSIONS

El desenvolupament del present projecte s'ha basat en aconseguir els objectius especificats a l'inici. L'objectiu principal era desplegar una xarxa pilot de sensors d'àudio formats per un node sensor i un coordinador, i amb la finalitat de detectar accidents de trànsit.

Ha estat necessari orientar el projecte cap a l'estudi d'un tipus de so d'accident concret, el que està format per un únic vehicle i a on es produeixen una fase de frenada i una de xoc. L'anàlisi realitzat amb Matlab ha proporcionat un marc teòric per a poder desenvolupar i integrar l'algorisme en un microcontrolador de recursos limitats, l'Arduino Pro Mini.

El disseny i la programació del hardware s'ha enfocat a dotar d'autonomia energètica als dispositius a través d'una placa solar. Els dispositius es comuniquen remotament i el coordinador és capaç d'enllaçar-se amb la xarxa d'Internet. La transmissió de les dades s'ha reduït al mínim de temps i repeticions per tal d'adormir els dispositius el màxim temps possible.

L'algorisme general d'anàlisi del so ha complert el propòsit principal de comparar i detectar els sons d'accidents de trànsit. Tot i les limitacions del hardware escollit, ha estat possible analitzar en temps real el so per tal d'extreure'n les principals components freqüencials així com la seva evolució per a instants de temps reduïts.

Les simulacions s'han efectuat en condicions favorables i sense tenir en compte tots els factors que poden afectar al dispositiu en l'aplicació real. Com a futurs estudis relacionats en l'anàlisi del so d'accidents de trànsit, caldria validar el model teòric amb simulacions que incloguin soroll de fons, emissió del so des de varies distàncies i proves pilot en entorns reals.

Dani Martí Vergé
Graduat en enginyeria electrònica industrial i automàtica

Girona, 1 de setembre de 2016

11. RELACIÓ DE DOCUMENTS

Aquest projecte està constituït pels següents documents: la memòria, els plànols, el plec de condicions, l'estat d'amidaments i el pressupost.

12. BIBLIOGRAFIA

AirSpayce. NRF24 Class Reference. Australia. (http://www.airspayce.com/mikem/arduino/NRF24/classNRF24.html, 10 de juny de 2016)

Arduino. SPI library. Arduino. (https://www.arduino.cc/en/Reference/SPI, 28 d'abril de 2016)

BIANCHI, A., QUEIROZ, M. Real time digital audio processing using Arduino. Proceedings of the Sound and Music Computing Conference. 2013.

BORKAR, P., MALIK, L. Acoustic Signal Based Traffic Density State Estimation using SVM. I.J. Image, Graphics and Signal Processing. 2013.

CABRERA, J., SALGADO, E. Acustica y fundamentos del sonido (E-Learning). Universidad Nacional Abierta y a Distancia. Bogotá. 2010.

CHU, S., NARAYANAN, S., KUO, J. Environmental Sound Recognition With Time–Frequency Audio Features. IEEE Transactions on Audio, Speech and Language Processing. 2009.

Columbia University. Matlab Audio Processing Examples. USA. (http://www.ee.columbia.edu/ln/rosa/matlab/, 5 d'abril de 2016)

DGT. Estadisticas accidentes de tráfico. Espanya. (http://www.dgt.es/Galerias/seguridad-vial/estadisticas-e-indicadores/publicaciones/accidentes-urban/accidentes_trafico005.pdf, 12 de juny de 2016)

GEORGANTI, E., MAY, T., VAN DE PAR, S., HARMA, A., MOURJOPOULOS, J. Speaker Distance Detection Using a Single Microphone. IEEE Transactions on Audio, Speech and Language Processing. 2011.

Gizmodo. How Shazam Works to identify every song you throw at it. Gizmodo. (http://gizmodo.com/5647458/how-shazam-works-to-identify-nearly-every-song-you-throw-at-it, 28 d'abril del 2016)

Guatewireless. Tabla de relación entre dBm i potencia de tranmissión WLAN. Guatewireless. (http://www.guatewireless.org/internetworking/redes/wireless/tabla-de-relacion-entre-dbm-y-potencia-de-transmision-wlan.html, 15 de juliol de 2016)

KAPLAN, S. A Wireless Sensor Network for Vibration Measurement. Master of Science in the Department of Electrical Engineering. University of Cape Town. 2011.

MathWorks. Cross Spectrum and Magnitude-Squared Coherence. MathWorks. (http://es.mathworks.com/help/signal/ug/cross-spectrum-and-magnitude-squared-coherence.html, 18 de juny de 2016)

Open Music Labs. Arduino FHT library. Chicago. (http://wiki.openmusiclabs.com/wiki/FHTFunctions, 20 d'agost de 2016)

POUS, C. Sistemes d'adquisició de dades, Tractament digital del senyal. Universitat de Girona. 2015.

PVGIS. Performance of Grid-Connected PV. Europa. (http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php#, 19 de maig de 2016)

Sparkfun. Wireless Buying Guide. Sparkfun. (https://www.sparkfun.com/pages/wireless_guide, 10 d'abril de 2016)

SunFields. Manual de cálculo de instalaciones fotovoltaicas Aisladas Autónomas. Santiago de Compostela. (http://www.sfe-solar.com/suministros-fotovoltaica-aislada-autonoma/manual-calculo/, 25 de maig de 2016)

ULPGC. La transformada discreta de Fourier. Las Palmas de Gran Canaria. (http://www2.dis.ulpgc.es/~li-pso/tema4/tema4.htm, 10 d'abril de 2016)

13. GLOSSARI

ADC	Analogic to Digital Converter
CPU	Central Processing Unit
EEPROM	Electrically-Erasable Programmable Read-Only Memory
FFT	Fast Fourier Transform
FHT	Fast Hartley Transform
GPRS	General Packet Radio Service
IDE	Integrated Development Environment
JFET	Junction Field Effect Transistor
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSID	Service Set IDentifier
UART	Universal Asynchronous Receiver/Transmitter
UdG	Universitat de Girona

A. CÀLCULS

La irradiació solar que rep la placa solar no és constant al llarg del dia, ni al llarg de l'any. L'Institute for Energy and Transport (IET) té desenvolupada la plataforma Photovoltaic Geographical Information System que proporciona uns índexs que representen la irradiació mitja diària i mensual per a una àrea geogràfica determinada. La fórmula general del càlcul de l'energia solar a partir dels mòduls instal·lats és:

$$C_{ed} = N_{mod} \cdot P_{Mp} \cdot HSP_{crit} \cdot PR \quad \text{(Eq. 31)}$$

A partir de l'índex hores solar pic (HSP), la potència pic (P_{Mp}) i el coeficient d'eficiència de la placa (PR) es pot calcular l'energia diària mitja que aportarà la placa solar per a cada mes de l'any.

Mesos	Gener	Febrer	Març	Abril	Maig	Juny
Hd	3,7	4,7	5,6	5,5	5,9	6,4
HSP	3,7	4,7	5,6	5,5	5,9	6,4
Energia solar 1 dia (Wh)	2,6	3,3	3,9	3,8	4,1	4,4

Taula 24. Càlcul de l'energia solar a partir de l'índex HSP pel primer semestre de l'any

Mesos	Juliol	Agost	Septembre	Octubre	Novembre	Desembre
Hd	6,4	6,3	5,8	4,9	3,9	3,6
HSP	6,4	6,3	5,8	4,9	3,9	3,6
Energia solar 1 dia (Wh)	4,5	4,4	4,0	3,4	2,7	2,5

Taula 25. Càlcul de l'energia solar a partir de l'índex HSP pel segon semestre de l'any

On N_{mod} es refereix al número de mòduls solars, C_{ed} al consum d'energia, P_{Mp} a la potència pic de la placa solar, HSP a hores solar pic i PR al coeficient d'eficiència de la placa solar.

B. PROGRAMACIÓ

B.1 Node sensor

```
  ////////////////////////////////////////////////////////////////////////
 //                  LLIBRERIES DEL NODE SENSOR 10.1                    //
////////////////////////////////////////////////////////////////////////

#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>

  ////////////////////////////////////////////////////////////////////////
 //                            PARAMETRES I VARIABLES                   //
////////////////////////////////////////////////////////////////////////

//Pins SPI i declaració del NRF24
RF24 radio(7, 8);
RF24Network network(radio);

// Adreça del node actual en octal
const uint16_t aquest_node = 01;

// Adreça node a enviar en octal
const uint16_t altre_node = 00;

// Variables del payload a enviar
struct dades_t
{
  float Acc;
  float BatS;
  float SolS;
  float Id;
};

// Paràmetres i variables del ADC del micròfon
const uint32_t FS = 40000;
const uint32_t FS_INTERVAL = F_CPU / FS;
const uint16_t MIN_ADC_CYCLES = 15;
uint8_t const ADC_REF_AVCC = (1 << REFS0);
int BufPtr = 0;
boolean ErrorCond = false;

// Declaració funcions adquisició
void adcInit(uint32_t ticks, uint8_t pin, uint8_t ref);
void adcStart();
uint16_t acquire();

// Configuració de sortida lineal de la transformada FHT
#define LIN_OUT 1

// Numero de punts de la transformada FHT
#define FHT_N 256


// lliberia de la transformada FHT
#include <FHT.h>
```

```
// Declaració entrades analògiques microfon i placa solar
const uint8_t microfon = 0;
int solar = A1;    // Solar

// Variables anivellació amplitud
long mx1;
long mx2;
long equiv1;

// Variables càlcul similitud
long dist1;
long dist2;
long dist1total;
long dist2total;
long rel1;
long rel2;
long rel1acum;
long rel2acum;

// Variables compliment condicions similitud
int n1 = 0;
int n2 = 0;
int t2 = 0;
byte t3 = 0;
bool frenada = false;
bool inicifrenada = false;

// Constants compliment condicions similitud
const int limit1 = 22;
const int limit2 = 44;
const int nlimit1 = 5;
const int nlimit2 = 4;
const int tlimit1 = 50;
const int tlimit2 = 170;
const int tlimit3 = 40;

// Declaració payload
dades_t dades;

// Comptadors
unsigned long t1 = 0;
int i;

  ////////////////////////////////////////////////////////////////////////
 //                      VECTORS DE FREQUENCIES DELS PATRONS             //
////////////////////////////////////////////////////////////////////////

// Patro 1, fase de frenada
long patro1[] =
{/*0Hz*/155,/*156Hz*/77,/*312Hz*/326,/*468Hz*/214,/*625Hz*/4271,/*781Hz*/21
46,/*937Hz*/1152,/*1093Hz*/150,/*1250Hz*/400,/*1406Hz*/1071,/*1562Hz*/331,/
*1718Hz*/30,/*1875Hz*/0,/*2031Hz*/20,/*2187Hz*/30,/*2343Hz*/77,/*2500Hz*/0,
/*2656Hz*/0,/*2812Hz*/40,/*2968Hz*/0,/*3125Hz*/0,/*3281Hz*/0,/*3437Hz*/0,/*
3593Hz*/0,/*3750Hz*/0,/*3906Hz*/0,/*4062Hz*/0,/*4219Hz*/20, /*4375z*/ 0,
/*4531Hz*/0, /*4688Hz*/ 2, /*4844Hz*/ 0, /*5000Hz*/ 4, /*5156Hz*/0,
/*5313Hz*/0, /*5469Hz*/ 4, /*5625Hz*/0, /*5781Hz*/ 0, /*5938Hz*/ 0,
/*6094Hz*/ 0 , /*6250Hz*/ 0, /*6406Hz*/0, /*6563Hz*/ 0, /*6719Hz*/ 0,
/*6875Hz*/ 8, /*7031Hz*/ 0, /*7188Hz*/ 0, /*7344Hz*/ 0, /*7500Hz*/ 0,
/*7566Hz*/0, /*7813Hz*/0};
```

```
// Patro 2, fase de xoc
long patro2[] =
{/*0Hz*/669,/*156Hz*/846,/*312Hz*/2014,/*468Hz*/1256,/*625Hz*/164,/*781Hz*/
217,/*937Hz*/54,/*1093Hz*/684,/*1250Hz*/78,/*1406Hz*/67,/*1562Hz*/89,/*1718
Hz*/360,/*1875Hz*/90,/*2031Hz*/75,/*2187Hz*/40,/*2343Hz*/101,/*2500Hz*/269,
/*2656Hz*/96,/*2812Hz*/0,/*2968Hz*/0,/*3125Hz*/0,/*3281Hz*/0,/*3437Hz*/30,/
*3593Hz*/0,/*3750Hz*/58,/*3906Hz*/0,/*4062Hz*/30,/*4219Hz*/0, /*4375z*/ 0,
/*4531Hz*/0, /*4688Hz*/ 81, /*4844Hz*/ 74, /*5000Hz*/ 0, /*5156Hz*/30,
/*5313Hz*/30, /*5469Hz*/53, /*5625Hz*/30, /*5781Hz*/30, /*5938Hz*/ 0,
/*6094Hz*/ 0 , /*6250Hz*/ 0, /*6406Hz*/21, /*6563Hz*/ 0, /*6719Hz*/ 0,
/*6875Hz*/ 0, /*7031Hz*/ 11, /*7188Hz*/ 0, /*7344Hz*/ 0, /*7500Hz*/ 0,
/*7566Hz*/0, /*7813Hz*/0};

// Mitjana dels 2 valors maxims del patro 1 i del patro 2
const long pmx1 = 3142;

  ////////////////////////////////////////////////////////////////////////
 //                  CONFIGURACIÓ INICIAL DEL NODE SENSOR                //
////////////////////////////////////////////////////////////////////////

void setup()
{
    // Configuració de l'ADC del microfon
    adcInit(FS_INTERVAL, microfon, ADC_REF_AVCC);

    // Inicialització NRF24
    SPI.begin();
    radio.begin();
    network.begin(90, aquest_node);

    // Habilitacio acusament de recepció NRF24
    radio.setAutoAck(1);

    // Temps d'espera per reenviar paquet (multiple de 250us) i numero
    d'intents
    radio.setRetries(8, 11);

    // Velocitat transmissió radio del NRF24
    radio.setDataRate(RF24_250KBPS);

    // Identificació del node sensor
    dades.Id = 10.1;
}

  ////////////////////////////////////////////////////////////////////////
 //                      BUCLE PRINCIPAL DEL NODE SENSOR                //
////////////////////////////////////////////////////////////////////////

void loop()
{

    // Reset variables
    mx1 = 0;
    mx2 = 0;

    // Comptador
    t1 = t1 + 1;

    // Crida a la funcio que realitza adquisicio
    adquisiciofht();
```

```
//Buscar els 2 màxims del vector capturat
for (i = 0; i < 50; i++) {
     dist1 = (patro1[i] - fht_lin_out[i]);
     if (fht_lin_out[i] > mx1) {
         mx1 = fht_lin_out[i];
     } else if (fht_lin_out[i] > mx2) {
         mx2 = fht_lin_out[i];
     }
}

//Mitjana dels 2 maxims
mx1 = (mx1 + mx2) / 2;

// Anivellació del vector capturat
if (mx1 > 150) {
     equiv1 = pmx1 / mx1;
     for (i = 0; i < 50; i++) {
          if (equiv1 >= 1) {
              fht_lin_out[i] = fht_lin_out[i] * equiv1;
          } else if (equiv1 < 1) {
              fht_lin_out[i] = fht_lin_out[i] * equiv1;
          }
     }
}

// Relació de distàncies entre el vector anivellat i els patrons
rel1acum = 0.0;
rel2acum = 0.0;
for (i = 0; i < 50; i++) {
     dist1 = (patro1[i] - fht_lin_out[i]);
     dist2 = (patro2[i] - fht_lin_out[i]);
     if (dist1 < 0) {
         dist1 = dist1 * (-1);
     }
     if (dist2 < 0) {
         dist2 = dist2 * (-1);
     }
     dist1total = (patro1[i] + fht_lin_out[i]);
     dist2total = (patro2[i] + fht_lin_out[i]);
     rel1 = round((100 * dist1) / dist1total);
     rel2 = round((100 * dist2) / dist2total);
     if (rel1 < 0) {
         rel1 = rel1 * (-1);
     }
     if (rel2 < 0) {
         rel2 = rel2 * (-1);
     }
     rel1 = 100 - rel1;
     rel2 = 100 - rel2;
     rel1acum = rel1 + rel1acum;
     rel2acum = rel2 + rel2acum;
}
rel1acum = round(rel1acum / (i - 1));
rel2acum = round(rel2acum / (i - 1));

// Coincidencia del patro 1 i inici de la frenada
if (rel1acum > limit1) {
    n1 = n1 + 1;
    inicifrenada = true;
}
```

```
// Si s'ha hi ha coincidencia del patro 1, temporitzador t3
if (inicifrenada){
     t3 = t3 + 1;
}

//Condició 1 supeditada a t3
if (t3 > tlimit3 and frenada == false){
     n1 = 0;
     n2 = 0;
     t2 = 0;
     t3 = 0;
     inicifrenada = false;
}

// Compliment condició 1
if (n1 > nlimit1) {
     n1 = 0;
     n2 = 0;
     t2 = 0;
     t3 = 0;
     inicifrenada = false;
     frenada = true;
}

// Si s'ha hi ha compliment condicio 1, temporitzador t2
if (frenada)
{
     t2 = t2 + 1;
}

// Coincidencia del patro 2
if (rel2acum > limit2) {
    n2 = n2 + 1;
}

// Condició 2 supeditada a t2
if (t2 > tlimit2){
     n1 = 0;
     n2 = 0;
     t2 = 0;
     t3 = 0;
     frenada = false;
}

// Compliment condició 2
if (n2 > nlimit2 and t2<tlimit2 and t2>tlimit1 and frenada) {
     n1 = 0;
     n2 = 0;
     t2 = 0;
     t3 = 0;
     frenada = false;
     dades.Acc = 20.2;
}

// Si repetició de bucle és igual a 100.000 (30 minuts) o hi ha un
accident
if (t1 > 99999 or dades.Acc > 10.0) {

     // Si no hi ha accident
     if (dades.Acc < 10.0) {
```

```
            //Crida a la funció llegirVcc() de lectura del voltatge intern
            dades.BatS = 1126400 / (llegirVcc() * 10);
            dades.BatS = round(dades.BatS);
            dades.BatS = dades.BatS / 100;

            // Lectura pin analogic del voltatge de la placa solar
            dades.SolS  = analogRead(solar);
            dades.SolS = dades.SolS * 4.0 / 1023;
            dades.Acc = 0.0;
            t1 = 0;

        // Si hi ha accident
        } else if (t1 < 100000) {
            dades.BatS = 0.0;
            dades.SolS = 0.0;
        }

        // Despertar NRF24
        radio.powerUp();
        radio.begin();
        network.begin(90, aquest_node);
        network.update();
        SPI.begin();
        RF24NetworkHeader header(altre_node);

        // Enviament paquet de dades
        bool ok = network.write(header, &dades, sizeof(dades));
    }

    // Temporitzador t1 (repeticions de bucle)
    t1 = t1 + 1;

    // Adormir el NRF24
    SPI.end();
    radio.powerDown();
}

  ////////////////////////////////////////////////////////////////////////
 //                          FUNCIONS DEL NODE SENSOR                  //
////////////////////////////////////////////////////////////////////////

//Funció d'adquisició i de transformada de la FHT
uint16_t adquisiciofht()
{
    // Inici de l'ADC del microfon
    adcStart();
    BufPtr = 0;

    // Esperar que el buffer de l'ADC tinguui 256 dades
    while (BufPtr < FHT_N) {
        millis();
    }

    // Enfinestrat ordenament dades de l'adc per a la FHT
    fht_window();
    fht_reorder();

    // Processar la FHT
    fht_run();

    // Output de la FHT, vector de 256 punts
```

```
    fht_mag_lin();
}

// Lectura interna del voltatge de l'alimentació
long llegirVcc() {
    long result;
    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1); delay(2);
    ADCSRA |= _BV(ADSC); // Convert
    while (bit_is_set(ADCSRA, ADSC));
    result = ADCL;
    result |= ADCH << 8;
    result = 1126400L / result;
}

// Configuració i inicialització de l'ADC del microfon
void adcInit(uint32_t ticks, uint8_t pin, uint8_t ref)
{
    if (ref & ~((1 << REFS0) | (1 << REFS1))) {
        ErrorCond = true;
        return;
    }
    ADMUX = ref | (pin & 7);
    #if RECORD_EIGHT_BITS
        ADMUX |= (1 << ADLAR);
    #endif
    ADCSRB = (1 << ADTS2) | (1 << ADTS0);
    #if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
        if (pin < 8) {
          ADCSRB &= ~(1 << MUX5);
          DIDR0 |= 1 << pin;
        } else {
          ADCSRB |= (1 << MUX5);
          DIDR2 |= 1 << (7 & pin);
        }
    #else
        if (pin < 6) DIDR0 |= 1 << pin;
    #endif
    #if ADPS0 != 0 || ADPS1 != 1 || ADPS2 != 2
    #error unexpected ADC prescaler bits
    #endif
    uint8_t adps;
    for (adps = 7; adps > 0; adps--) {
        if (ticks >= (MIN_ADC_CYCLES << adps)) break;
    }
    if (adps < 3)
    {
        ErrorCond = true;
        return;
    }
    ADCSRA = adps;
    ticks >>= adps;
    ticks <<= adps;
    TCCR1A = 0;
    uint8_t tshift;
    if (ticks < 0X10000) {
        TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS10);
        tshift = 0;
    } else if (ticks < 0X10000 * 8) {
        TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11);
        tshift = 3;
    } else if (ticks < 0X10000 * 64) {
```

```
        TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11) | (1 << CS10);
        tshift = 6;
    } else if (ticks < 0X10000 * 256) {
        TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS12);
        tshift = 8;
    } else if (ticks < 0X10000 * 1024) {
        TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS12) | (1 << CS10);
        tshift = 10;
    } else {
        ErrorCond = true;
        return;
    }
    ticks >>= tshift;
    ICR1 = ticks - 1;
    OCR1B = 0;
    ticks <<= tshift;
}

// Inici de l'ADC del microfon
void adcStart() {
    ADCSRA |= (1 << ADATE)  | (1 << ADEN) | (1 << ADIE) | (1 << ADSC) ;

    //Habilitació interrupcions del timer1
    TIMSK1 = (1 << OCIE1B);
    TCNT1 = 0;
}

// Interrupció de l'ADC del microfon
ISR(ADC_vect) {
    #if RECORD_EIGHT_BITS
        uint8_t d = ADCH;
    #else
        uint8_t low = ADCL;
        uint8_t high = ADCH;
        uint16_t d = (high << 8) | low;
    #endif
    int k = d - 0x0200;
    k <<= 6;

    // Només guardar en el vector capturat si el buffer té menys de 256
    dades
    if (BufPtr < FHT_N)
    {
        // Vector capturat
        fht_input[BufPtr] = k;
    }
    BufPtr++;
}
ISR(TIMER1_COMPB_vect) {}
```

B.2 Coordinador

```
  ////////////////////////////////////////////////////////////////////////
 //                          LLIBRERIES DEL COORDINADOR                  //
////////////////////////////////////////////////////////////////////////

#include <RF24Network.h>
#include <RF24.h>
#include <SPI.h>
```

```
  ////////////////////////////////////////////////////////////////////////
 //                          PARAMETRES I VARIABLES                      //
////////////////////////////////////////////////////////////////////////

// Pins SPI declaració del NRF24
RF24 radio(7, 8);
RF24Network network(radio);

// Adreça del node actual en octal
const uint16_t aquest_node = 00;

// Adreça node a enviar en octal
const uint16_t altre_node = 01;

// Variables del payload a rebre
struct dades_t {
  float Acc;
  float BatS;
  float SolS;
  float Id;
};

// Variables a adquirir en el coordinador
float BatC;
float SolC;

// Matrius de caràcters per a la concatenació
char charsols[10];
char charbats[10];
char characc[10];
char charbatc[10];
char charsolc[10];
char charid[10];

// Entrada analògica de la placa solar
int sensorsolar = A1;

// Declaració tupla per guardar el payload a rebre
dades_t dades;

// Comptadors
int t4;

  ////////////////////////////////////////////////////////////////////////
 //               CONFIGURACIÓ INICIAL DEL COORDINADOR                   //
////////////////////////////////////////////////////////////////////////
void setup(void) {

    // Inicialització port série
    Serial.begin(115200);

    // Pin digital per a despertar el ESP8266
    pinMode(3, OUTPUT);

    // Inicialització NRF24
    SPI.begin();
    radio.begin();
    network.begin(90, aquest_node);
```

```
    // Velocitat de transmissió radio del NRF24
    radio.setDataRate(RF24_250KBPS);

}

  ////////////////////////////////////////////////////////////////////////
 //                          BUCLE PRINCIPAL COORDINADOR                 //
////////////////////////////////////////////////////////////////////////

void loop(void) {

    // Reset variables
    dades.Id=0.0;

    // Comptador
    t4 = t4 + 1;

    // Si s'ha arribat a la repetició 5
    if (t4 > 4) {

        // Despertar NRF24
        radio.powerUp();
        radio.begin();
        network.begin(90, altre_node);
        network.update();
        SPI.begin();

        // Escoltar si arriba algun paquet
        while ( network.available() ) {
            RF24NetworkHeader header;
            bool ok = network.read(header, &dades, sizeof(dades));

            // Si dades.Id>1.0 ha arribat un nou paquet
            if (dades.Id > 1.0) {

               // Despertar el ESP8266 activant el pin connectat a CH_PD
               digitalWrite(3, HIGH);
               delayMicroseconds(100);

               // Inici port serie per enviar les dades al ESP8266
               Serial.begin(115200);

               // El paquet no es d'accident
               if (dades.Acc < 10.0) {

                   //Crida la funció llegirVcc() llegir voltatge intern
                   BatC = 1126400 / (llegirVcc() * 10);
                   BatC = round(BatC);
                   BatC = BatC / 100;

                   //Lectura pin analogic del voltatge de la placa solar
                   SolC  = analogRead(sensorsolar);
                   SolC = SolC * 4.0 / 1023;

               // El paquet es d'accident
               } else if (dades.Acc > 10.0) {

                   //Assignem 0.0 a les variables SolC i BatC
                   BatC = 0.0;
                   SolC = 0.0;
               }
```

```
// Espais i lletres a concatenar
String stringbats = "";
String stringsols = "";
String stringbatc = "";
String stringsolc = "";
String stringacc = "";
String stringid = "";
String bats = "B";
String sols = "S";
String batc = "C";
String solc = "T";
String acc = "A";
String idd = "I";

// Concatenació variable dades.BatS i el prefix B
dtostrf(dades.BatS, 5, 2, charbats);
for (int i = 0; i < sizeof(charbats); i++)
{
    stringbats += charbats[i];
}
bats += stringbats;
bats.remove(5, 10);

// Impressió del string bats pel port serie
Serial.println(bats);

// Concatenació i impressio dades.SolS i el prefix S
dtostrf(dades.SolS, 5, 2, charsols);
for (int i = 0; i < sizeof(charsols); i++)
{
    stringsols += charsols[i];
}
sols += stringsols;
sols.remove(5, 10);
Serial.println(sols);

// Concatenació i impressió BatC i el prefix C
dtostrf(BatC, 5, 2, charbatc);
for (int i = 0; i < sizeof(charbatc); i++)
{
    stringbatc += charbatc[i];
}
batc += stringbatc;
batc.remove(5, 10);
Serial.println(batc);

// Concatenació i impressió SolC i el prefix T
dtostrf(SolC, 5, 2, charsolc);
for (int i = 0; i < sizeof(charsolc); i++)
{
    stringsolc += charsolc[i];
}
solc += stringsolc;
solc.remove(5, 10);
Serial.println(solc);

// Concatenació i impressió dades.Acc i el prefix A
dtostrf(dades.Acc, 5, 2, characc);
for (int i = 0; i < sizeof(characc); i++)
{
```

```
                    stringacc += characc[i];
                }
                acc += stringacc;
                acc.remove(5, 10);
                Serial.println(acc);

                // Concatenació dades.Id i el prefix I
                dtostrf(dades.Id, 5, 2, charid);
                for (int i = 0; i < sizeof(charid); i++)
                {
                    stringidd += charid[i];
                }
                idd += stringid;
                idd.remove(5, 10);
                Serial.println(idd);

                // Desalimentar el pin en mode power down
                digitalWrite(3, LOW);
                delayMicroseconds(100);
            }
        }

        // Adormir el NRF24 i apagar el port serie
        SPI.end();
        radio.powerDown();
        Serial.end();

        // Resetejar comptador t4
        t4=0;
    }
}

  ////////////////////////////////////////////////////////////////////////
 //                            FUNCIONS DEL COORDINADOR                  //
////////////////////////////////////////////////////////////////////////

// Lectura interna del voltatge d'alimentació
long llegirVcc() {
  long result; // Read 1.1V reference against AVcc
  ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1); delay(2); // Wait
for Vref to settle
  ADCSRA |= _BV(ADSC); // Convert
  while (bit_is_set(ADCSRA, ADSC)); result = ADCL; result |= ADCH << 8;
result = 1126400L / result; // Back-calculate AVcc in mV return result;
}
```

B.3 ESP8266

```
  ////////////////////////////////////////////////////////////////////////
 //                          LLIBRERIES DE L'ESP8266                     //
////////////////////////////////////////////////////////////////////////

#include <ESP8266WiFi.h>

  ////////////////////////////////////////////////////////////////////////
 //                     PARÀMETRES DEL WIFI I DEL SERVIDOR               //
////////////////////////////////////////////////////////////////////////
```

```
// Servidor ThingSpeak
String apiKey = "74YPULIPVKTJTQAU";
const char* server = "api.thingspeak.com";

// SSID WiFi
const char* ssid = "Orange";
const char* password = "A5E1D37D3C";

// Declaració pin GPIO 0 connectat a CH_PD
int pinON = 0;

// Inicialització llibreria
WiFiClient client;


  ////////////////////////////////////////////////////////////////////////
 //                 CONFIGURACIÓ INICIAL DEL ESP8266                   //
////////////////////////////////////////////////////////////////////////

void setup() {

    // Cada vegada que es desperta, es reinicia, i el pinON activa CH_PD
    pinMode(pinON, OUTPUT);
    digitalWrite(pinON, HIGH);
    delayMicroseconds(100);

    // Inici comunicació serie amb l'Arduino
    Serial.begin(115200);

    // Connexió a la xarxa
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delayMicroseconds(100);
    }
}

  ////////////////////////////////////////////////////////////////////////
 //                       BUCLE PRINCIPAL DE L'ESP8266                 //
////////////////////////////////////////////////////////////////////////

void loop() {

     // Connexió al servidor
    String bufferArray[ 4 ];
    if (client.connect(server, 80)) {
        String postStr = apiKey;

        // Emmagatzament dades del port serie
        if ( Serial.available() > 1) {
            for (int i = 0; i < 3; i++) {
                bufferArray[ i ] = Serial.readStringUntil('\n');
            }

             // Descodificació primer caràcter de la primera dada
            String BatS = bufferArray[0];
            if (BatS.charAt(0) == 'B') {
                BatS.remove(0, 1);

                // Guardar al camp que relaciona amb al servidor
                postStr += "&field1=";
```

```
                postStr += String(BatS);
            }

            // Descodificació i guardar al camp
            String SolS = bufferArray[1];
            if (SolS.charAt(0) == 'S') {
                SolS.remove(0, 1);
                postStr += "&field2=";
                postStr += String(SolS);
            }
            String BatC = bufferArray[2];
            if (BatC .charAt(0) == 'C') {
                BatC .remove(0, 1);
                postStr += "&field3=";
                postStr += String(BatC );
            }
            String SolC = bufferArray[2];
            if (SolC.charAt(0) == 'T') {
                SolC.remove(0, 1);
                postStr += "&field4=";
                postStr += String(SolC);
            }
            String Acc = bufferArray[2];
            if (Acc.charAt(0) == 'A') {
                Acc.remove(0, 1);
                postStr += "&field5=";
                postStr += String(Acc);
            }
            String Id = bufferArray[2];
            if (Id.charAt(0) == 'I') {
                Id.remove(0, 1);
                postStr += "&field6=";
                postStr += String(Id);
            }
        }

        // Enviament de les dades al servidor
        postStr += "\r\n\r\n";
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);

    }

    // Finalitzazció conexxió al servidor
    client.stop();

    // Deshabilitació pin Power On
    digitalWrite(pinON, LOW);
    delayMicroseconds(100);
}
```