

Treball de Fi de Grau

Grau en Tecnologies Industrials

Disseny i construcció d'un lector de bus CAN sense contacte

MEMÒRIA

Autor: Albert Valle Vives
Director: Manuel Moreno Eguílaz
Convocatòria: Abril 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

L'objectiu d'aquest projecte és el disseny i construcció d'un dispositiu que pugui llegir, sense contactes, la informació que s'envia a través d'un bus CAN (protocol de comunicació sèrie utilitzat entre les centraletes d'un vehicle). Per tant, ha de ser capaç de captar els senyals del bus mitjançant unes antenes, processar la informació, i enviar-la cap a un ordinador per poder-la veure.

En el present document s'explica què és i com funciona un bus CAN, quina informació passa a través d'ell, i com s'estructura aquesta informació.

Pel que fa al hardware o elements físics que conformen el dispositiu dissenyat, s'han utilitzat unes antenes per captar el senyal, seguit d'un amplificador, un filtre passa alts, un comparador amb histèresis, un biestable RS asíncron, i finalment un transceptor de bus CAN. Per al processament d'informació s'ha utilitzat un microcontrolador PIC18F258 i s'ha soldat tot a una placa de forats.

Per l'altre banda, a nivell de software, s'ha modificat un codi en llenguatge de programació C per a programar el microcontrolador mitjançant l'entorn MPLAB IDE i s'ha creat uns programes amb *python* per a la validació del correcte funcionament del dispositiu hardware.

El resultat ha estat satisfactori; el dispositiu dissenyat funciona correctament i compleix les especificacions exigides que s'han proposat des d'un inici.

Sumari

RESUM	1
SUMARI	3
1. GLOSSARI	5
2. PREFACI	7
2.1. Origen del projecte.....	7
2.2. Motivació.....	7
2.3. Requeriments previs.....	8
3. INTRODUCCIÓ	9
3.1. Antecedents.....	9
3.2. Objectius del projecte	10
3.3. Abast del projecte	10
4. DESCRIPCIÓ DEL BUS CAN	11
4.1. Què és el Bus CAN.....	11
4.2. Característiques principals del protocol CAN.....	12
4.3. Elements d'un sistema Bus CAN.....	13
4.4. Explicació funcional del Bus CAN.....	15
4.5. Formats de trama.....	17
4.5.1. Trama de dades.....	17
5. MATERIAL	19
5.1. Antenes.....	19
5.2. Placa principal.....	20
5.2.1. Amplificador d'instrumentació.....	21
5.2.2. Convertidor de tensió.....	21
5.2.3. Comparador	22
5.2.4. Biestable	22
5.2.5. Transceiver CAN.....	22
5.2.6. Microcontrolador.....	23
5.2.7. Adaptadors	23
5.2.8. Perifèrics	23
5.3. USBTin	24
5.4. Kvaser CanKing.....	24

5.5. Programador.....	25
6. FUNCIONAMENT DEL DISPOSITIU _____	26
6.1. Hardware	26
6.1.1. Antenes	28
6.1.2. Amplificar el senyal.....	30
6.1.3. Convertir el senyal.....	32
6.2. Software	39
7. VALIDACIÓ _____	41
7.1. Simulació	41
7.2. Longitud d'antena	42
7.3. Dispositiu	44
7.3.1. USBtin	44
7.3.2. Kvaser CanKing	46
7.3.3. Python	48
8. COSTOS DEL PROJECTE _____	51
9. IMPACTE AMBIENTAL _____	53
10. CONCLUSIONS _____	54
11. AGRAÏMENTS _____	55
12. BIBLIOGRAFIA _____	56

1. Glossari

CAN	<i>Control Area Network</i> ; bus de comunicacions per a vehicles dissenyat per permetre la comunicació entre els diferents mòduls de l'electrònica sense necessitat d'un ordinador amfitrió.
Hardware	Components físics que constitueixen un sistema electrònic.
Software	Conjunt de programes amb instruccions llegibles per una màquina que necessita per a executar les accions que hi estiguin descrites.
PIC	<i>Programmable Intelligent Computer</i> ; família de microcontroladors fabricats per la empresa Microchip Technology.
CPU	<i>Central Processing Unit</i> ; part d'un ordinador o microcontrolador encarregada de processar totes les funcions d'aquest.
RAM	<i>Random Acces Memory</i> ; memòria volàtil d'un ordinador o microcontrolador. En ella s'hi emmagatzema dades de curt termini ja que no és un memòria permanent.
ROM	<i>Read Only Memory</i> ; memòria no volàtil d'un ordinador o microcontrolador. En ella s'hi emmagatzema programes i dades ja que és una memòria que manté el seu contingut fins i tot sense alimentació.
EEPROM	<i>Electrical Erasable Programmable Read-Only Memory</i> ; és un cas de memòria ROM que pot ser esborrada i reprogramada elèctricament.
Perifèric	Element connectat a un microcontrolador que permet comunicar-lo amb dispositius externs.
USART	<i>Universal Synchronous/Asynchronous Receiver/Transmitter</i> ; perifèric que permet connectar un microcontrolador amb el port sèrie d'un ordinador.
SMD	<i>Surface-Mount Device</i> ; format de petites dimensions d'alguns circuits integrats, resistències i condensadors que no necessiten forats en el PCB.
PCB	<i>Printed Circuit Board</i> ; placa constituïda per camins, pistes o busos de material conductor que s'utilitza per a connectar un conjunt d'elements elèctrics.

Bit	Unitat mínima d'informació que només pot prendre dos valors: 0 i 1.
Bytes	Conjunt de 8 bits que constitueix el mínim conjunt d'informació amb què treballen els ordinadors.
DLC	<i>Data Length Code</i> ; llargada d'un missatge CAN mesurat en bytes.
RTR	<i>Remote Transmission Request</i> ; bit que usa un node receptor per a sol·licitar informació a un altre node dins del protocol CAN.
CMOS	<i>Complementary metal-oxide-semiconductor</i> ; una de les famílies lògiques utilitzades en la fabricació de circuits integrats.
NAND	Porta lògica que produeix una sortida falsa només si totes les seves entrades són verdaderes.
TTL	<i>Transistor-transistor logic</i> ; tecnologia de construcció de circuits digitals.

2. Prefaci

2.1. Origen del projecte

Avui en dia, en el sector d'automoció, les millores tecnològiques introduïdes en els vehicles són cada vegada més abundants (climatitzadors, tancament centralitzat, sostres solars, seients elèctrics, quadre d'instruments, els comandaments al volant, sistemes multimèdia, ADAs, conducció autònoma ...).

El bus CAN és el protocol de comunicació que utilitzen la immensa majoria dels automòbils a Europa, i que ara es comença a introduir en motocicletes, per a la transmissió de missatges entre les unitats electròniques de control. Amb tota aquesta electrònica, és normal que augmenti el risc d'averies d'un automòbil, per això un altre ús del bus CAN és el de la presa de dades del vehicle i la de diagnosticar errors i possibles averies (OBD).

Per tant, es lògic pensar que el bus CAN té un gran paper i és un element de vital importància dins d'un automòbil.

L'origen d'aquest projecte neix amb la curiositat de saber com es podria facilitar la lectura d'aquesta informació que passa a través del bus CAN sense haver de tallar els cables i fer connexions extres, cosa que faria perdre la garantia del vehicle.

2.2. Motivació

A arrel d'aquesta curiositat, va sorgir la idea per part del director d'aquest projecte d'investigar i dissenyar un dispositiu que pogués oferir aquesta funció esmentada anteriorment. Un cop estudiats els productes ja comercialitzats, es va decidir de intentar millorar alguns dels aspectes tècnics que ofereixen certs dispositius.

En definitiva, la motivació de projecte ha estat investigar i experimentar des de zero la manera de construir un dispositiu que pugui llegir la informació que passa a través del bus CAN, millorant característiques d'equips ja comercialitzats.

A nivell personal, aquest projecte em va cridar molt l'atenció, ja que l'objectiu era la construcció d'un dispositiu, és a dir, es centrava sobretot en el hardware i no tant en el software que hi ha darrera de tot aparell electrònic. A part, aquest dispositiu anava destinat al sector automobilístic, que és el sector que més m'atrau del món empresarial.

2.3. Requeriments previs

Per a la realització d'aquest projecte seran necessaris coneixements adquirits en diferents assignatures vistes en el grau *Enginyeria en Tecnologies Industrials*.

Les més significatives serien ***Electrònica*** i ***Informàtica I i II***.

Per una banda, d'Informàtica necessitarem tots els coneixements bàsics de programació en Python impartits a l'assignatura per la creació de programes destinats a la validació final del dispositiu.

Per altre banda, donada la naturalesa d'aquest projecte, tenir coneixements d'electrònica serà de gran importància i ens donaran la base per poder realitzar-lo.

3. Introducció

3.1. Antecedents

Actualment, en el sector de l'automoció existeixen aparells comercials similars al que tractem en aquest treball, aparells d'empreses internacionals dedicades sobretot al disseny i fabricació de solucions telemàtiques i dispositius de telecomunicacions.

Per exemple, **Technoton [1]** és un desenvolupador i fabricant internacional de solucions telemàtiques de combustible per al control del consum de combustible per als vehicles i les instal·lacions fixes. Un dels seus productes és el *CANcrocodile* que permet obtenir dades sobre el vehicle que s'executa des del bus CAN sense cap interrupció en la integritat del bus. La lectura de dades es produeix sense connexió elèctrica als cables. Aquest dispositiu s'alimenta a una tensió nominal de 12V o 24V, i pot llegir missatges transmesos a través del bus CAN sense perdre'n cap a una velocitat màxima de 250 kbits/s.

Un altre exemple prové de **GPS4NET [2]**, que porta més d'una dècada involucrat en la investigació, desenvolupament i producció de diverses solucions telemàtiques. G4N03TAP és un dispositiu genuí creat per recol·lectar informació específica del vehicle, com el consum de combustible o odòmetre de qualsevol vehicle equipat amb bus CAN o JBus. El dispositiu treballa a 12V o 24V també, però aquest permet capturar missatges a una velocitat de fins a 1Mbits/s.

TELTONIKA [3] és un altre exemple d'empresa el qual té com alguns dels seus productes aquesta classe de dispositius. Un d'ells és el SIMPLE-CAN, que treballa a una tensió d'entre 9V i 63V i pot llegir missatges correctament a una velocitat d'entre 33,33 i 500 kbits/s.

En tots els casos, amb petits detalls o diferents versions, aquests aparells tenen la mateixa finalitat, la de llegir la informació transmesa a través del bus CAN sense necessitat de fer una connexió elèctrica al vehicle.

S'ha demanat una oferta a la empresa Technoton del seu producte *CANcrocodile* per tenir una idea del preu d'un dispositiu similar al que es vol dissenyar. Aquest preu és de 32€.

3.2. Objectius del projecte

L'objectiu d'aquest projecte és dissenyar i construir un dispositiu que faci possible la lectura d'informació que passa per un bus CAN sense contacte elèctric amb aquest.

Per aconseguir-ho, aquest dispositiu ha de poder captar els senyals dels missatges del bus CAN i processar aquesta informació per enviar-la a un ordinador on l'usuari sigui capaç de llegir-la correctament.

Vist els equips comercials i a les baixes freqüències que treballen molts d'ells, un altre dels objectius que s'ha proposat aquest projecte és dissenyar el dispositiu de manera que pugui llegir i captar la informació a qualsevol velocitat a la que viatgi, des de 125 kbits/s fins 1Mbits/s, funcionant així, dins d'un gran rang.

També, per a cada velocitat dins d'aquest rang, ha de ser capaç de llegir la informació sense perdre cap missatge, ja sigui quan s'envia un únic missatge com quan s'envien missatges de forma massiva, cosa que farem utilitzant alguns programes escrits en Python.

3.3. Abast del projecte

El projecte s'inicia amb un petit estudi del bus CAN per entendre el funcionament d'aquest dins els automòbils i així poder començar amb una petita idea de com podrà ser el disseny del dispositiu.

El dispositiu que es pretén construir en aquest projecte no està enfocat a la producció en sèrie per a la posterior venda. Per tant, es tractarà d'un prototip funcional, que contempli totes les característiques tècniques proposades. Es buscarà optimitzar les dimensions del dispositiu, utilitzant una placa de forats i els corresponents components però sense enfocar-nos en el disseny físic que hauria de tenir si el volguéssim comercialitzar.

Per altre banda, a nivell de software, ens ajudarem de treballs finals de grau anteriors, relacionats amb el bus CAN, per aconseguir la comunicació entre el dispositiu i un ordinador, agafant els programes utilitzats en aquests treballs com a guia per a crear de nous o simplement modificar-los perquè siguin adients a les nostres necessitats del projecte.

4. Descripció del Bus CAN

4.1. Què és el Bus CAN

El Bus CAN és un protocol de comunicació en sèrie desenvolupat per la firma alemana Bosch en els anys 80 per a l'intercanvi d'informació entre unitats de control electròniques de l'automòbil, tot i que actualment ja ha despertat l'interès d'altres sectors com l'àrea de control i automatització industrial.

Les sigles CAN signifiquen *Controller Area Network* (Xarxa d'àrea de control). Un bus, en informàtica, és un sistema digital que transfereix dades entre els components d'un ordinador o entre ordinadors. Està format per cables o pistes en un circuit imprès, dispositius com per exemple resistències i condensadors a més de circuits integrats.

Aquest sistema permet compartir una gran quantitat d'informació entre els diferents mòduls de control connectats a la xarxa, el que provoca una reducció important tant del nombre de sensors utilitzats així com la quantitat de cables que componen la instal·lació elèctrica. D'aquesta manera augmenten considerablement les funcions actuals en els sistemes de l'automòbil on s'empra el bus CAN sense augmentar els costos.

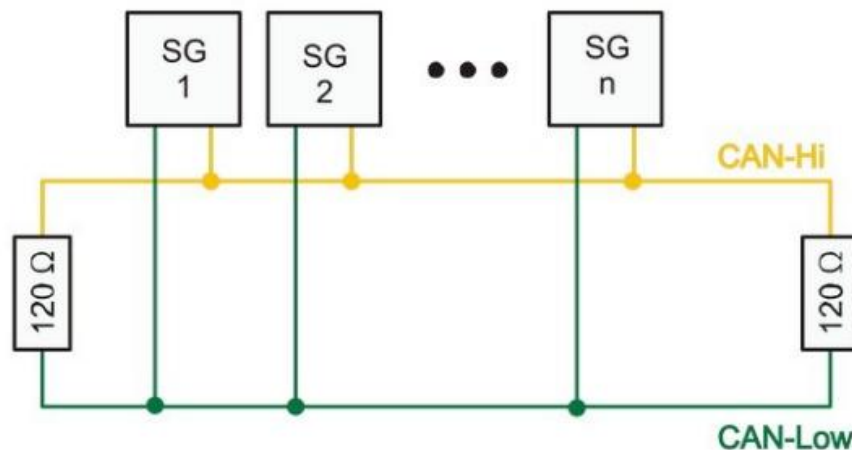


Figura 4.1. Mòduls connectats en una xarxa Bus CAN. Font: www.motorpasion.com

4.2. Característiques principals del protocol CAN

- La informació que circula entre les unitats de comandament a través del bus són paquets de 0 i 1 (bit) amb una longitud limitada i amb una estructura definida de camps que conformen el missatge.
- Un d'aquests camps actua d'identificador del tipus de dada que es transporta, de la unitat de comandament que ho transmet i de la prioritat per transmetre-ho respecte d'altres. El missatge no va adreçat a cap unitat de comandament en concret, cadascuna d'elles reconeixerà mitjançant aquest identificador si el missatge li interessa o no.
- Totes les unitats de la xarxa poden ser transmissores i receptores, i el nombre de mòduls connectats a la xarxa és variable, dins d'uns límits.
- Quan el bus està lliure, qualsevol mòdul pot enviar un missatge, i si un altre ho intenta al mateix temps, el conflicte es resol gràcies a la prioritat que l'identificador del missatge indica.
- El sistema està dotat d'una sèrie de mecanismes que assegurin que el missatge és transmès i rebut correctament. Quan un missatge presenta un error, és anul·lat i tornat a transmetre de forma correcta. De la mateixa manera, un mòdul amb problemes ho avisa als altres mòduls mitjançant el propi missatge, i si la situació no es soluciona, aquest mòdul queda fora de servei però el sistema segueix funcionant.

4.3. Elements d'un sistema Bus CAN

Cables

La informació circula per dos cables trenats que uneixen totes els mòduls que formen el sistema. Aquesta informació es transmet per diferència de tensió entre els dos cables, de manera que un valor alt de tensió representa un 1 i un valor baix de tensió representa un 0. La combinació adequada d'aquests uns i zeros formen el missatge a transmetre.

En un dels cables, els valors de tensió oscil·len entre 0V i 2.25V, per la qual cosa es denomina cable_L (*Low*) i en l'altre, el cable_H (*High*) ho fan entre 2.75V i 5V. Depenen del voltatge que passa pels cables, el bus es pot trobar en estat recessiu, quan els dos cables estan al mateix nivell de tensió, o en estat dominant, quan hi ha una diferència de tensió de 1,5V mínim. Aquest tipus de comunicació té com objectiu proporcionar una major protecció davant a interferències electromagnètiques.

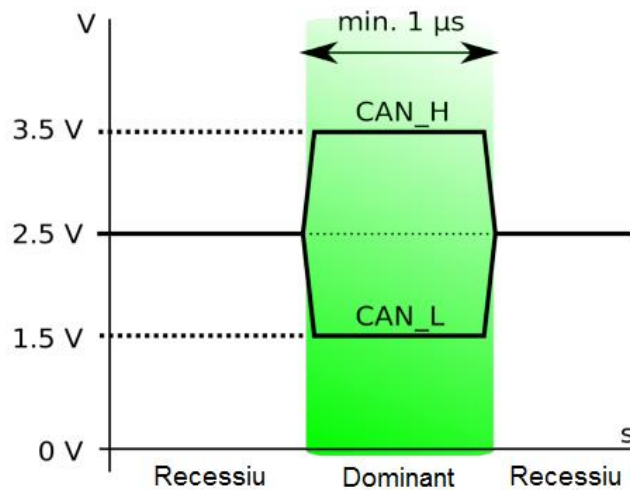


Figura 4.2. Estats Bus CAN. Font: es.wikipedia.org

Elements terminadors

Els elements terminadors del bus CAN són resistències connectades en els extrems dels cables H i L. Habitualment, els seus valors són de 120 Ω (encara que poden variar) i es col·loquen per adequar el funcionament del sistema a diferents longituds de cables i nombre de mòduls connectats, ja que impedeixen possibles efectes paràsits que ens poden trastocar el missatge.

Controlador

És l'element encarregat de la comunicació entre el processador del mòdul i el transmissor-receptor. Treballa preparant la informació que entra i surt entre els dos components.

El controlador està situat en la unitat de control, així doncs, hi haurà tants controladors com mòduls a la nostra xarxa. Aquest element treballa amb nivells de tensió molt baixos i és qui determina la velocitat de transmissió dels missatges, que serà més o menys elevada depenent de la nostra aplicació i del que esperem d'ella. El controlador també intervé en la sincronització entre els diferents mòduls per a la correcta emissió i recepció dels missatges.

Transmissor / Receptor

El transmissor-receptor és l'element que té la missió de rebre i de transmetre les dades, a més de preparar la informació perquè pugui ser utilitzada pels controladors. Aquesta preparació consisteix en situar els nivells de tensió de forma adequada, amplificant el senyal quan la informació es bolca en la línia i reduint-la quan és recollida de la mateixa i subministrada al controlador.

El transmissor-receptor és bàsicament un circuit integrat que està situat en cadascun dels mòduls, treballa amb intensitats pròximes a 0.5 A i en cap cas intervé modificant el contingut del missatge. Per al seu bon funcionament se situa entre els cables que formen la línia bus CAN i el controlador.

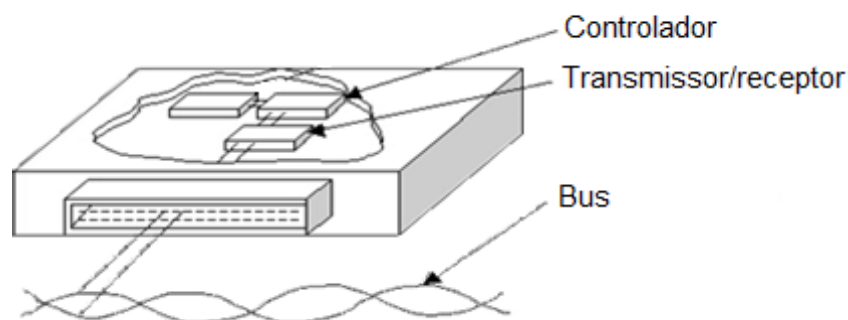


Figura 4.3. Mòdul d'un bus CAN. Font: www.tinet.cat

4.4. Explicació funcional del Bus CAN

Els mòduls que es connecten al sistema bus CAN són els que necessiten compartir informació, pertanyin o no a un mateix sistema. Per exemple, si parlem de cotxes doncs caldria dir que la velocitat de transmissió de les dades de l'ABS haurien de ser molt ràpids per l'important que suposa la informació utilitzada i no obstant això la del climatitzador doncs no caldria tenir una línia súper ràpida ja que el mostreig es pot fer mes pausadament.

El sistema bus CAN està orientat cap al missatge i no al destinatari. La informació és transmesa en forma de missatges estructurats en els quals una part del mateix és un identificador que indica la classe de d'informació que conté. Tots els mòduls reben el missatge, el filtren i només ho fan servir els que necessiten aquesta informació. Tots els mòduls connectats al sistema són capaços tant d'enviar com de rebre missatges del bus. Quan el bus està lliure qualsevol mòdul connectat pot començar a transmetre de nou. En el cas que diversos mòduls vulguin enviar un missatge al mateix temps, ho farà el del missatge que tingui major prioritat.

El procés de transmissió de dades es desenvolupa seguint diversos passos:

Subministrament de dades:

Un mòdul rep informació dels sensors que té associats (r.p.m. d'un motor, velocitat, temperatura del motor, porta oberta, en el nostre cas botons, etc.)

El seu microprocessador passa la informació al controlador on és gestionada i preparada per ser passada al transmissor-receptor on es traduirà en senyals elèctrics.

Transmissió de dades:

El controlador d'aquest mòdul transfereix les dades i l'identificador juntament amb la petició d'inici de transmissió, assumint la responsabilitat que el missatge sigui correctament transmès a totes els mòduls de la xarxa. Per transmetre el missatge ha hagut de trobar el bus lliure. A partir d'aquest moment, la resta de mòduls es converteixen en receptors.

Recepció del missatge:

Quan tots els mòduls reben el missatge, cada mòdul verifica l'identificador per determinar si l'interessa utilitzar la informació del missatge. En cas que un mòdul ho necessiti, aquest processa el missatge, en canvi, si no li interessa, l'ignora.

El sistema bus CAN disposa de mecanismes per detectar errors en la transmissió de missatges, de manera que tots els receptors realitzen una revisió del missatge analitzant una part d'aquest, anomenat camp CRC. Altres mecanismes de control s'apliquen en els emissors que miraran el nivell del bus, la presència de camps de format fix en el missatge (verificació de la trama), anàlisis estadístiques per part dels mòduls dels seus propis errors, etc. Aquestes mesures fan que les probabilitats d'error en l'emissió i recepció de missatges siguin molt baixes.

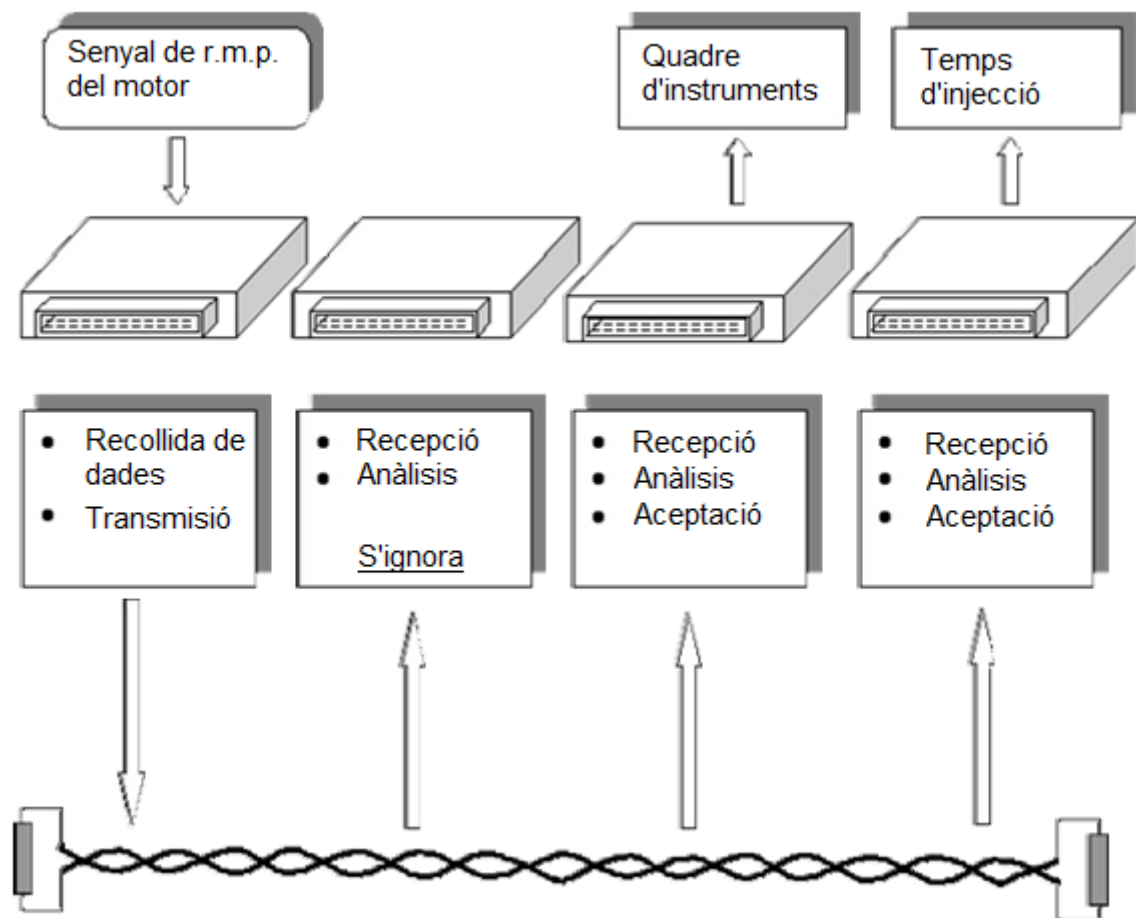


Figura 4.4. Exemple pràctic de CAN. Font: www.tinet.cat

4.5. Formats de trama

El missatge es descompon en camps de diferent grandària (nombre de bits) que ens permetran dur a terme el procés de comunicació entre els mòduls segons el protocol definit per Bosch per al CAN Bus, que faciliten des identificar el mòdul, com indicar el principi i al final del missatge, mostrar les dades, permetre diferents controls, etc.

Els missatges són introduïts en la línia amb una cadència que oscil·la entre els 7 i els 20 mil·lisegons, depenent de la velocitat de l'àrea i del mòdul que els introdueix.

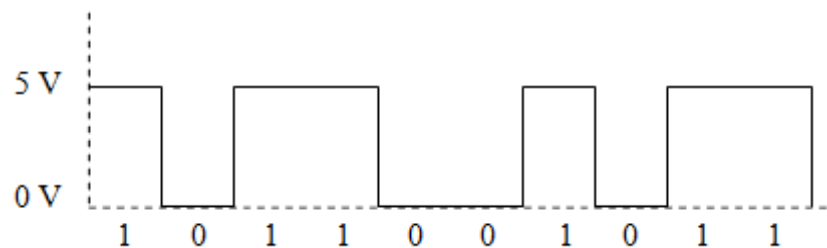


Figura 4.5. Exemple de com s'escriu un missatge. Font: www.tinet.cat

4.5.1. Trama de dades

Estructura del missatge estàndard:

- **Camp d'inici del missatge:** El missatge s'inicia amb un bit dominant i el seu flanc descendent és utilitzat pels mòduls per sincronitzar-se entre si.
- **Camp d'arbitri:** Els 11/29 bits d'aquest camp es fan servir com a identificador que permet reconèixer als mòduls la prioritat del missatge. Com més baix sigui el valor de l'identificador més alta és la prioritat, i per tant determina l'ordre en què van a ser introduïts els missatges en la línia. El bit RTR indica si el missatge conté dades (RTR = 0) o si es tracta d'una trama remota sense dades (RTR = 1). Una trama de dades sempre té una prioritat més alta que una trama remota. La trama remota la fem servir per demanar dades a altres mòduls o bé perquè es necessiten o per fer una revisió.
- **Camp de control:** Aquest camp informa sobre les característiques del camp de dades. El bit IDE indica quan és un "0" que es tracta d'una trama estàndard (11 bits) i quan és un "1" que és una trama estesa (29 bits). El segon bit (RB0) és sempre recessiu. Els quatre bit que componen el camp DLC indiquen el nombre de bytes contingut en el camp de dades (0 a 8).

- Camp de dades: En aquest camp apareix la informació del missatge amb les dades que el mòdul corresponent introdueix en la línia CAN Bus. Pot contenir entre 0 i 8 bytes (de 0 a 64 bits).
- Camp detector d'errors (CRC): Aquest camp té una longitud de 16 bits i s'utilitza per detectar errors pels 15 primers, mentre l'últim sempre serà un bit recessiu (1) per delimitar el camp CRC.
- Camp de confirmació (ACK): El camp ACK es compon de dues bit que són sempre transmesos com recessius (1). Totes els mòduls que reben el mateix CRC modifiquen el primer bit del camp ACK per un dominant (0), de manera que el mòdul que està encara transmetent reconeix que almenys un mòdul ha rebut un missatge escrit correctament. Si no és així, el mòdul transmissor interpreta que el seu missatge té un error.
- Camp de final de missatge (EOF): Aquest camp indica el final del missatge amb una cadena de 7 bits recessius. Ens pot succeir que en determinats missatges es produeixin llargues cadenes de zeros o uns, i que això ens provoqui una pèrdua de sincronització entre els mòduls. El protocol CAN resol aquesta situació ficant un bit de diferent valor cada cinc bits iguals: cada cinc "0" s'insereix un "1" o a l'inrevés. El mòdul que utilitzi el missatge descarta un bit posterior a cinc bits iguals. Aquests bits reben el nom de bit "stuffing".
- Espaiat entre trames (IFS): Consta d'un mínim de 3 bits recessius.

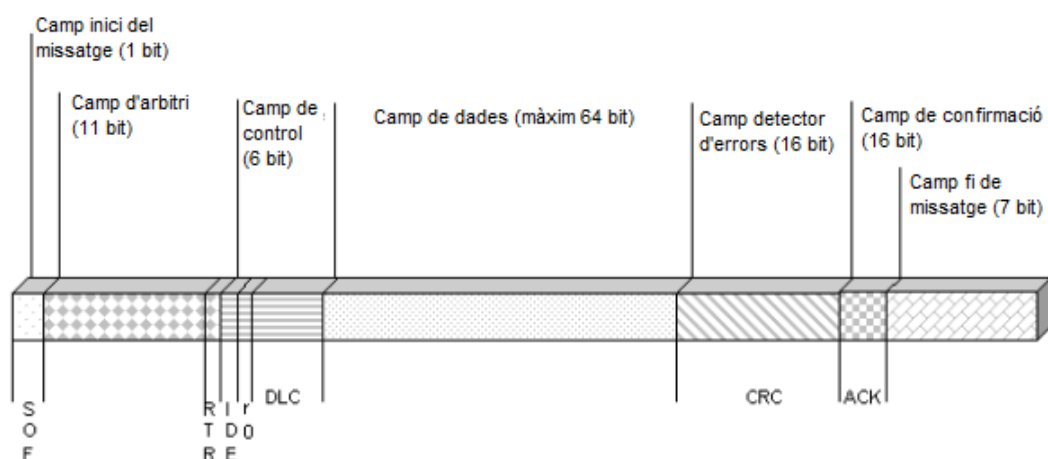


Figura 4.6. Trama de missatge CAN. Font: www.researchgate.net

5. Material

En aquest apartat es comentarà i detallarà tots els elements que componen el dispositiu i material usat per al desenvolupament del projecte.

5.1. Antenes

El primer element del dispositiu són unes antenes que s'han fet soldant una placa de coure de manera que simulés la forma que fariem en una placa de PCB.

Un fenomen que s'ha tingut en compte a l'hora de construir-les és el *crosstalk*[4]. En electrònica, *crosstalk* és la influència o pertorbació que provoca un senyal transmès a través d'un circuit o canal sobre un altre senyal diferent, i sol ser causat per un acoblament capacitiu, inductiu o conductiu no desitjat d'un circuit o canal a un altre.

Per evitar aquests efectes no desitjats, els cables que surten de les antenes, s'han creuat entre ells, formant una trena. Això provocarà que els camps magnètics que genera cada cable influïnt en l'altre quedin neutralitzats.

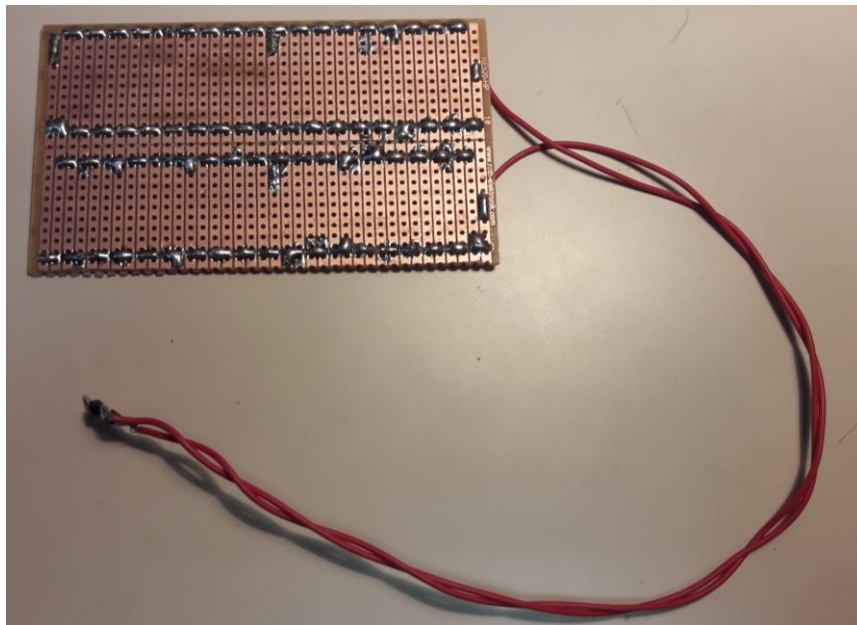


Figura 5.1. Antenes del dispositiu. Font: Pròpia.

5.2. Placa principal

La placa de forats on s'han soldat tots els components és el nucli del dispositiu. En aquesta hi trobem des de l'amplificador que rep el senyal de les antenes fins al microcontrolador que enviarà cap a l'ordinador els missatges. També s'han inclòs tots els elements necessaris per garantir la correcta funcionalitat dels elements, com per exemple resistències i condensadors, cables, un rellotge oscil·lador de quars...

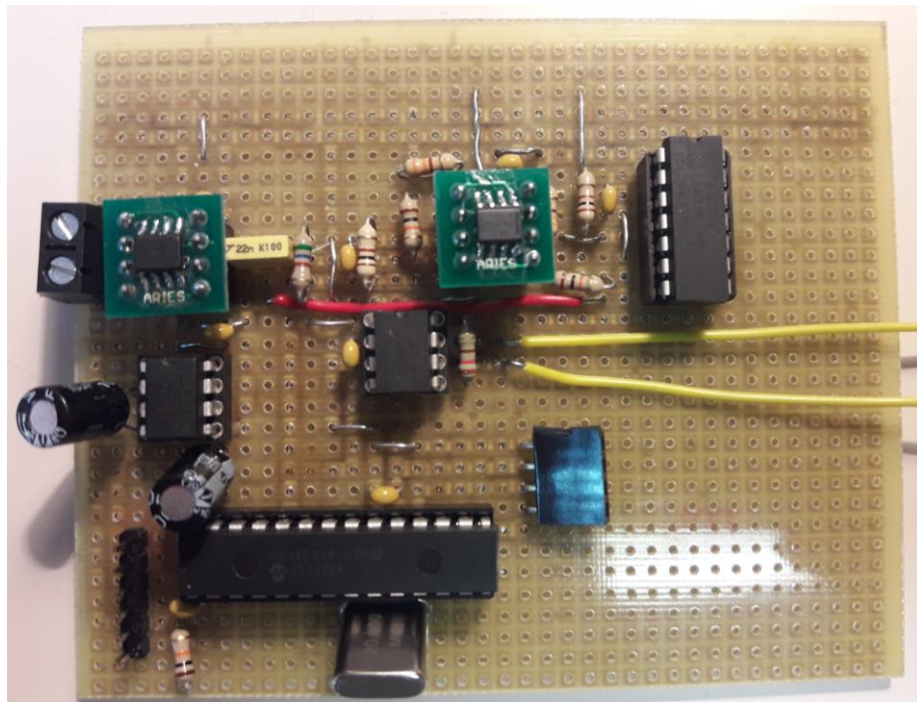


Figura 5.2. Placa principal del dispositiu. Font: Pròpia.

5.2.1. Amplificador d'instrumentació

Un amplificador d'instrumentació és un dispositiu creat a partir de varis amplificadors operacionals. L'amplificador d'instrumentació triat és el AD8421 de *Analog Devices*.

El que s'ha buscat sobretot per part de l'amplificador és que pogués treballar a altes freqüències.



Figura 5.3. AD8421. Font: www.farnell.com.

5.2.2. Convertidor de tensió

El dispositiu ICL7662 de *Renesas Electronic* és un convertidor negatiu de tensió que s'ha utilitzat per aconseguir la tensió negativa que l'amplificador d'instrumentació necessita al treballar en mode *dual supply*.

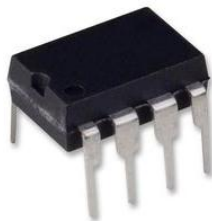


Figura 5.4. ICL7662. Font: www.farnell.com.

5.2.3. Comparador

El comparador utilitzat és el TLC352 de l'empresa *Texas Instruments*. Dins d'aquest dispositiu conté dos comparadors diferents, cosa que, al necessitar fer un comparador amb histèresis, ens ha ajudat a optimitzar espai en la placa.



Figura 5.5. TLC352. Font: www.farnell.com.

5.2.4. Biestable

S'ha utilitzat el dispositiu CMOS d'alta velocitat 74HC00 per construir un biestable SR asíncron. Aquest circuit integrat inclou 4 portes NAND de dues entrades.

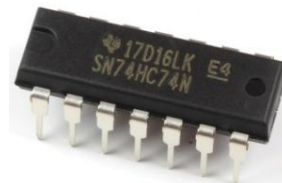


Figura 5.6. 74HC00. Font: www.oddwires.com.

5.2.5. Transceiver CAN

El transceiver o transceptor de CAN és la interfície entre el controlador de bus CAN i el bus. Aquest mòdul és l'encarregat del control dels nivells del senyal o del control d'accés al medi.

El transceiver de CAN que s'ha utilitzat és el PCA82C250. Aquest dispositiu és denominat d'alta velocitat, ja que admet velocitats fins a 1 Mbit/s, una de les característiques per la qual ha estat elegit.



Figura 5.7. PCA82C250. Font: www.farnell.com.

5.2.6. Microcontrolador

Un microcontrolador és un ordinador petit present en circuits integrats que té la finalitat de realitzar una tasca i executar una funció en concret. Aquest inclou dins seu les principals unitats funcionals d'un ordinador: CPU (*Central Processing Unit*), memòria RAM (volàtil), memòria Flash ROM (no volàtil), memòria EEPROM (per guardar dades que no es perdin quan es deixi d'alimentar-lo), ports sèrie, ports d'entrada i sortida, etc.

El microcontrolador que s'ha utilitzat en aquest projecte ha estat el PIC18F258 de l'empresa Microchip Technology. S'ha escollit aquest microcontrolador sobretot pel fet de tenir integrat en ell un controlador de bus CAN.

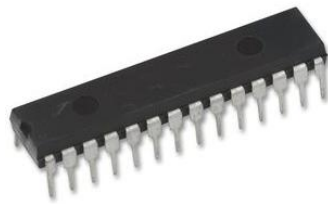


Figura 5.8. PIC18F258. Font: www.farnell.com.

5.2.7. Adaptadors

Degut als diferents encapsulats SMD dels elements, s'han utilitzat adaptadors SOIC-8 que han estat necessaris per fer les proves prèvies en la *Protoboard* i d'altres a l'hora de soldar en la placa final del dispositiu.

5.2.8. Perifèrics

S'han necessitat de diferents connectors per comunicar la placa amb el programador ICD3 per poder provar el programa modificat i deixar programat el microcontrolador, i amb la USART per on es comunica el dispositiu amb l'ordinador personal.

5.3. USBtin

Des de l'inici del projecte, a cada etapa que s'ha fet, s'ha necessitat de missatges CAN generats des d'un ordinador per veure si funcionava correctament. De la mateixa manera, també s'han necessitat per fer les comprovacions finals, un cop construït el dispositiu. Aquests s'han aconseguit mitjançant un dispositiu anomenat USBtin[5]. USBtin és una interfície senzilla d'USB a CAN desenvolupat per Thomas Fischl, de baix cost econòmic i senzilla funcionalitat, per la qual cosa s'ha decidit utilitzar-la. Aquest dispositiu actua com a bus CAN entre l'ordinador i la nostra placa. També, a través del software USBtinViewer permet la visualització dels missatges enviats o rebuts a través de l'ordinador.

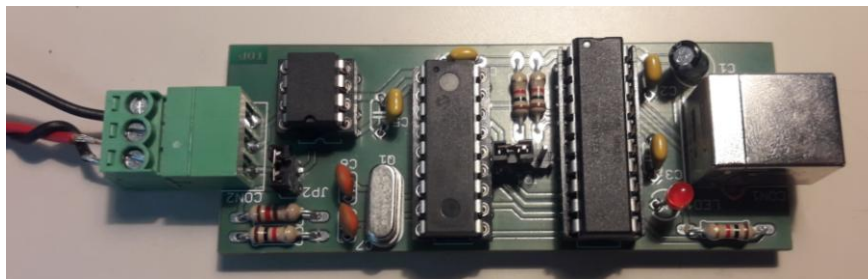


Figura 5.9. USBtin. Font: Pròpia.

5.4. Kvaser CanKing

CanKing és un monitor de bus CAN i una eina de diagnòstic de propòsit general desenvolupat per l'empresa Kvaser[6]. Aquest dispositiu és especialment adequat per al treball de desenvolupament interactiu. És capaç d'enviar i rebre missatges CAN, escriure i guardar dades en un fitxer, visualitzar els missatges transmesos...

En el nostre projecte, s'ha utilitzat el CanKing, juntament amb el hardware Leaf Light de Kvaser (Fig.5.10), per testejar els missatges que surten del nostre dispositiu. Connectant el CAN_H i el CAN_L s'ha observat si els senyals que sortien del transceiver de CAN eren interpretats correctament com a missatges de protocol CAN a les diferents velocitats. També s'ha vist a quina velocitat entraven els missatges, si hi havia errors en algun d'aquests, de qui tipus eren...



Figura 5.10. Kvaser CanKing. Font: www.kvaser.com.

5.5. Programador

Per programar el microcontrolador s'ha utilitzat el MPLAB ICD3, que és un depurador / programador d'alta velocitat per a dispositius de control de senyal digital i microcontroladors de Microchip Technology. El MPLAB ICD3 està preparat per a treballar a nivell de hardware i software amb dispositius de la mateixa empresa, com és el cas del microcontrolador que s'ha utilitzat en el projecte.



Figura 5.11. Programador ICD3. Font: www.microchip.com.

6. Funcionament del dispositiu

Aquest projecte s'ha desenvolupat per etapes. La primera ha estat el disseny i construcció del dispositiu pel que fa al hardware. Aquesta etapa es podria dividir en tres parts: investigar la manera de captar el senyals del bus CAN sense contactes (antenes), la amplificació d'aquest senyal captat i la transformació a senyal digital per a la obtenció dels missatges de protocol CAN.

La segona etapa ha estat més a nivell de software. Ha consistit en l'ampliació d'aquest hardware amb el microcontrolador i la seva programació per aconseguir comunicar el dispositiu dissenyat amb un ordinador per a la visualització dels missatges.

6.1. Hardware

A la figura 6.1 es pot veure l'esquemàtic del dispositiu dissenyat. En aquest apartat s'explica pas per pas com s'ha arribat a dissenyar el dispositiu a nivell de hardware.

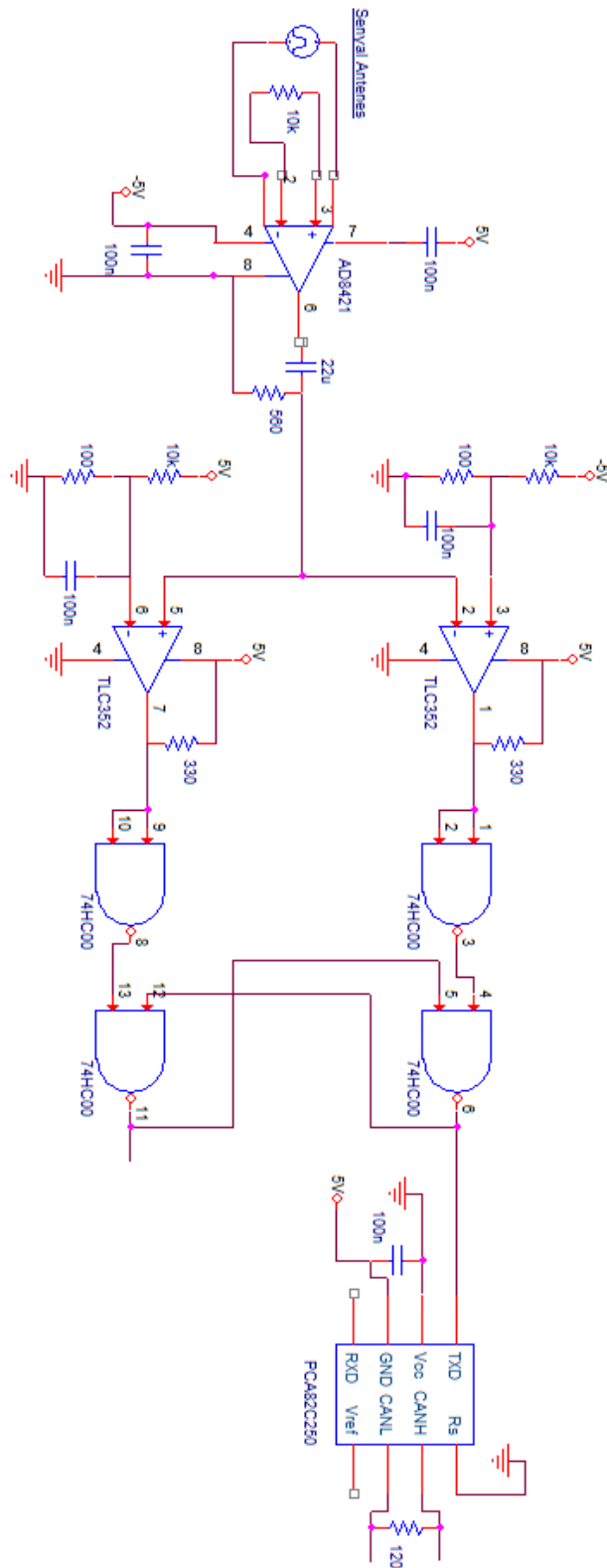


Figura 6.1. Esquema elèctric del dispositiu. Font: Pròpia.

6.1.1. Antenes

Com s'ha comentat abans, el primer pas ha estat aconseguir captar els senyals que passen a través del bus CAN. En aquest cas, amb l'USBtin, s'han generat els missatges CAN, de manera ininterrompuda, aconseguint així un senyal periòdic.

Per obtenir el senyal s'han posant unes antenes a prop dels cables del bus. Aquestes antenes s'han creat a partir d'una placa de coure, fent punts de soldadura als extrems buscant la forma que es veu a la Fig. 6.2.

En la Fig. 6.3 es pot apreciar com es col·loquen les antenes juntament amb els cables del bus per captar els senyals. El cable vermell del USBtin és el CAN High i el blau el CAN Low.

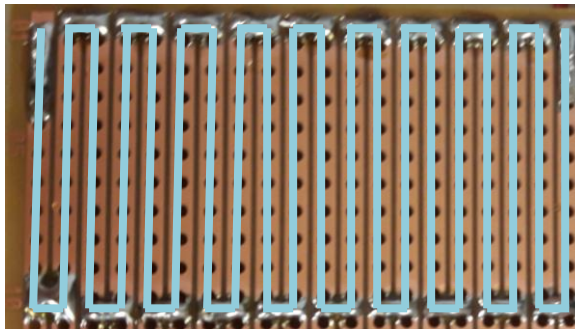


Figura 6.2. Mitja antena. Font: Pròpia.

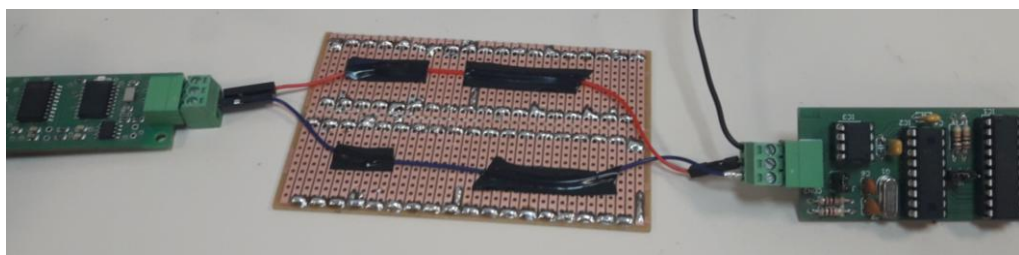


Figura 6.3. Antenes captant senyal del USBtin. Font: Pròpia.

Un cop confeccionades i col·locades les antenes, s'ha procedit a visualitzar com és el senyal que capten amb l'ajuda de l'oscil·loscopi.

En la Fig. 6.4 es pot veure els senyals que es rep de les antenes a una velocitat de 125 Kbit/s, on el canal CH1 és el CAN High i el canal CH2 el CAN Low. En la Fig. 6.5 s'observa el mateix però a màxima velocitat, 1 Mbit/s.

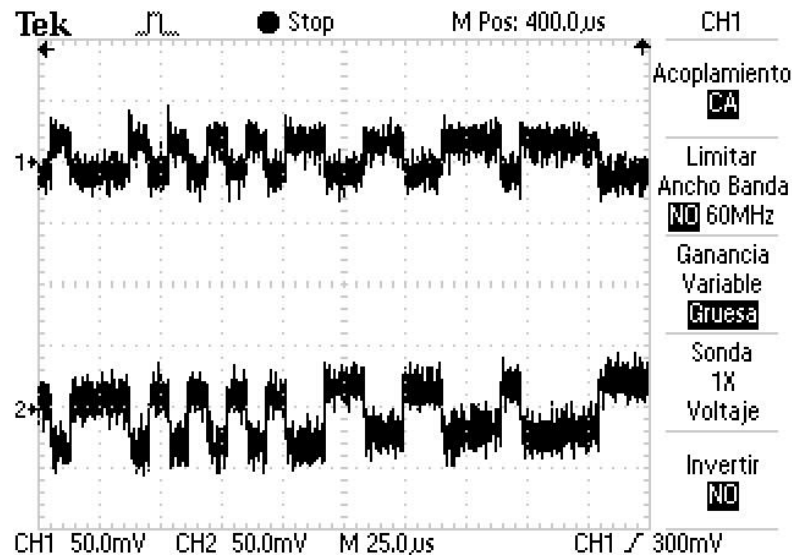


Figura 6.4. Senyal CAN High i Low captat per les antenes a 125 Kbit/s. Font: Pròpia.

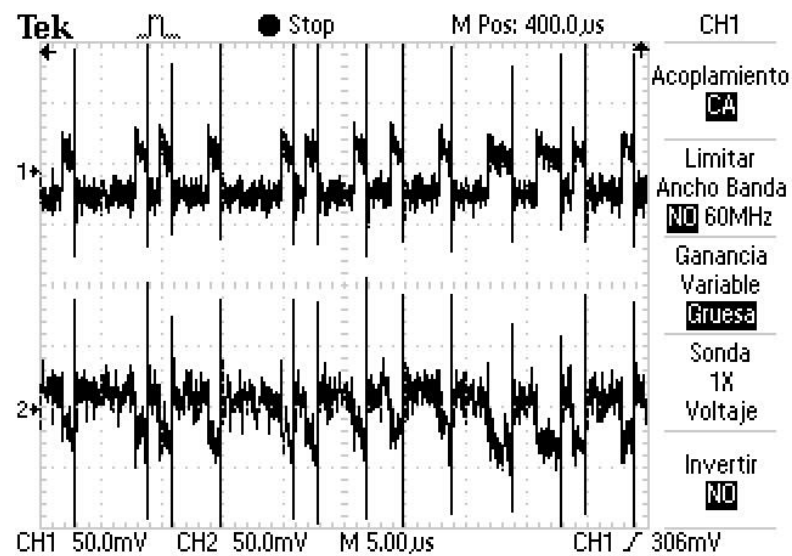


Figura 6.5. Senyal CAN High i Low captat per les antenes a 1 Mbit/s. Font: Pròpia.

Com es pot observar aquests senyals són molt febles, al voltant d'uns 50 mV, però s'aprecia en ells la forma d'un missatge CAN, cosa que indica que les antenes funcionen correctament, tot i el senyal estar distorsionat a causa del soroll.

6.1.2. Amplificar el senyal

Un cop aconseguit els senyals, s'ha seguit amb el disseny de la placa principal del dispositiu. En aquesta part s'ha utilitzat una *protoboard* i una font d'alimentació per dur a terme els diferents assajos.

Per tant, el següent pas, ha estat amplificar el senyal que ens subministra les antenes degut a la seva baixa tensió.

Per fer-ho, s'ha utilitzat l'amplificador d'instrumentació AD8421. Una característica molt bona que té aquest amplificador és que pot treballar a freqüències molt altes, de fins a 10 MHz a guany 1. Això deixa un marge per augmentar-li el guany en cas que sigui necessari si l'amplificació a guany 1 no és suficient i que segueixi funcionant.

En aquest projecte es treballa amb velocitats de missatges CAN de fins a 1 Mbit/s, per tant, es necessita que l'amplificador pugui treballar a 1 MHz còmodament. Observant la Fig. 6.6, es pot veure com l'amplificador pot treballar a aquesta freqüència correctament a guany 10, com a mínim.

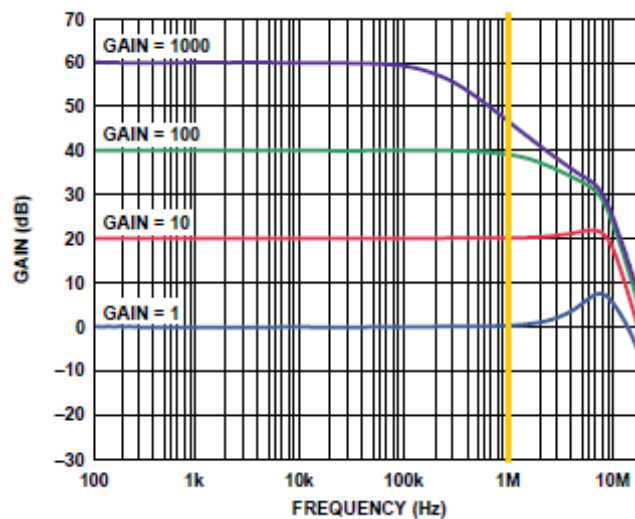


Figura 6.6. Gràfic guany vs. freqüència del AD8421. Font: Datasheet AD8421.

En aquest cas s'ha augmentat el guany fins a 2, utilitzant una resistència de 10K Ω , a partir de la fórmula del guany del AD8421:

$$G = 1 + \frac{9.9k\Omega}{R_G} \quad (1)$$

El senyal que surt de l'amplificador és la diferència de les dues tensions a l'entrada (CAN_H i CAN_L) augmentada pel guany, així doncs a partir d'aquest moment treballarem amb un únic senyal.

$$V_{OUT} = G \times (V_{+IN} - V_{-IN}) + V_{REF} \quad (2)$$

En les següents figures es pot veure el senyal resultant de l'amplificació (canal CH2) respecte els senyals que rebem a l'antena (canal CH1).

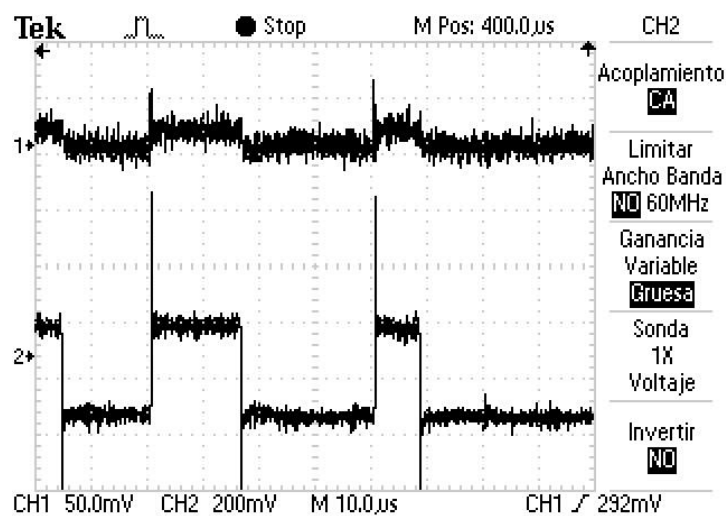


Figura 6.7. Senyal amplificat respecte el CAN Low a 125 Kbit/s. Font: Pròpia.

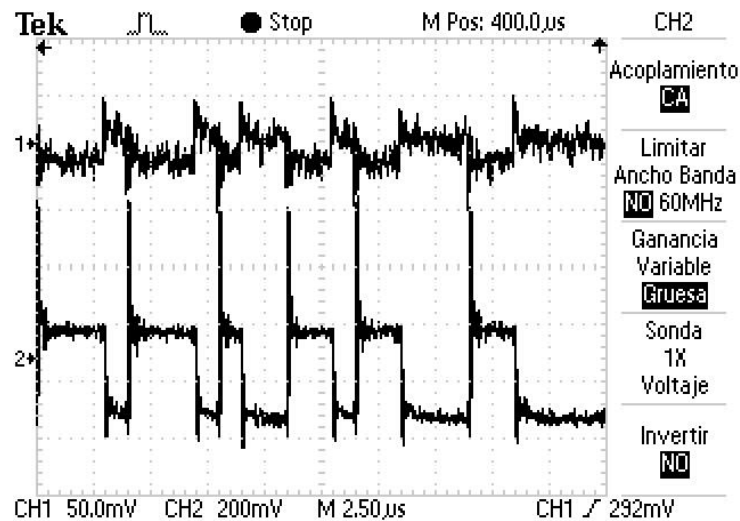


Figura 6.8. Senyal amplificat respecte el CAN High a 1 Mbit/s. Font: Pròpia.

Gràcies a l'amplificació el nou senyal que s'ha obtingut gira al voltant dels 300/400 mV. Com ja hem dit, el nou senyal surt de la diferència entre el CAN High i CAN Low, és per això que, com es pot observar a la Fig. 6.8, respecte al senyal High la sortida queda invertida.

A partir d'aquest nou senyal, s'ha procedit a la conversió a senyals TTL, passant per diferents processos.

6.1.3. Convertir el senyal

Un cop s'ha amplificat el senyal provinent de les antenes, s'ha procedit a convertir-lo en un senyal digital TTL que el transceiver de CAN pugui llegir. Per això s'ha fet servir un comparador amb histèresis. Però primer s'ha filtrat el senyal amb un filtre passiu (Fig. 6.9), concretament un passa alts, eliminant així la possible component continua del senyal adquirit.

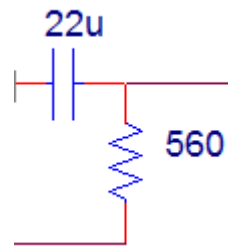


Figura 6.9. Filtre passa alt. Font: Pròpia.

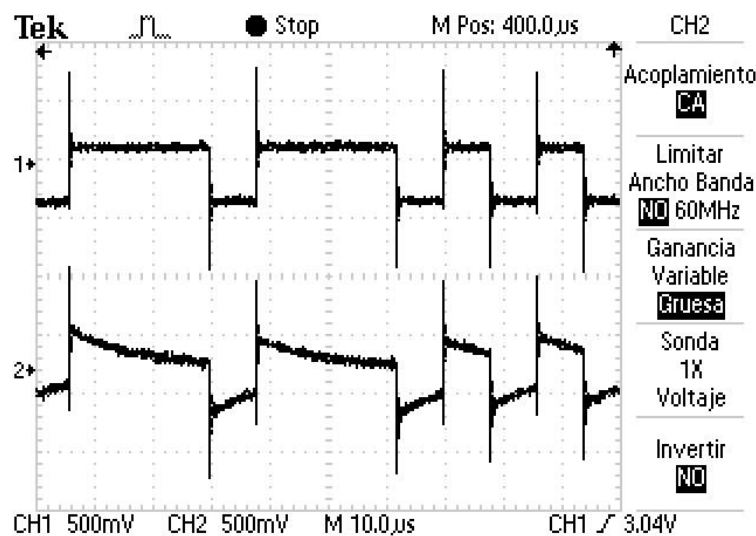


Figura 6.10. Senyal amplificat i filtrat a 125 Kbit/s. Font: Pròpia.

En la Fig. 6.10 es pot observar el senyal amplificat (canal CH1) i el senyal passat pel filtre passa alt (canal CH2). Com es veu, apareixen pendents de tensió entre els flancs de pujada i baixada. Això és degut a que el filtre elimina la component continua del corrent i el condensador comença a descarregar-se.

Inicialment es va fer servir un comparador per passar del senyal analògic al digital, però la comparació amb un únic nivell de tensió resultava en un senyal que es saltava algun flanc o n'introduïa de nous degut a irregularitats i a pics de tensió puntuals en el senyal inicial. Per això, s'ha decidit utilitzar els dos comparadors que conté el dispositiu TLC352, i dur a terme una comparació amb histèresis (Fig. 6.11).

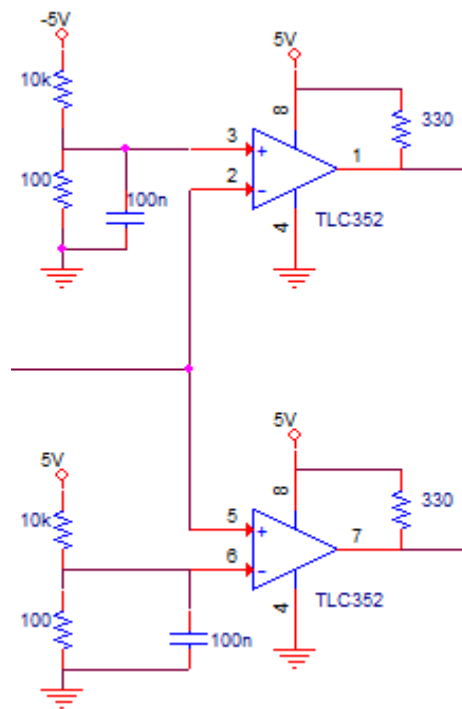


Figura 6.11. Comparadors amb histèresis. Font: Pròpia.

Amb la histèresis és dur a terme una comparació del senyal amb dos nivells de tensió diferents, és a dir, mirar si el senyal està per sobre o per sota d'un rang. El que s'aconsegueix amb això és eliminar transicions de sorolls que ens puguin afectar a fer una bona comparació.

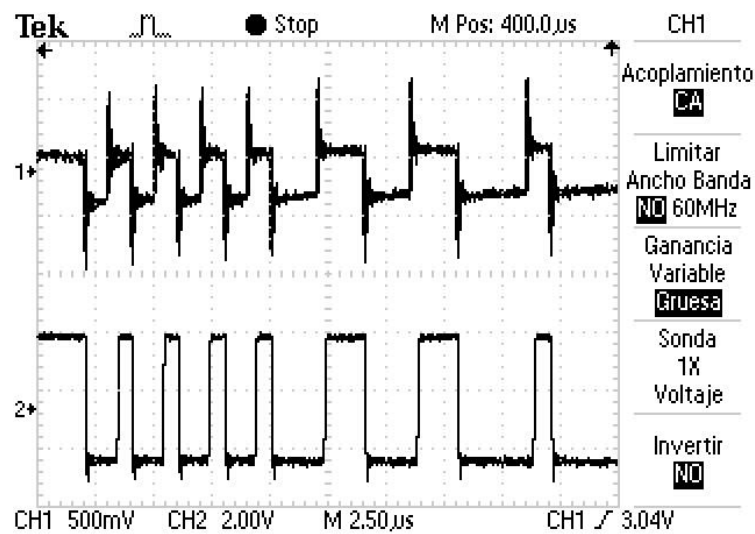


Figura 6.12. Senyal comparador 1 a 1 Mbit/s. Font: Pròpia.

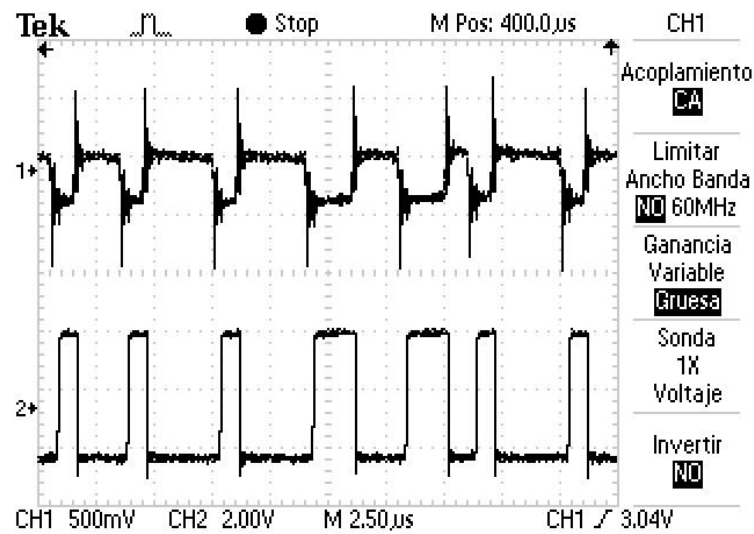


Figura 6.13. Senyal comparador 2 a 1 Mbit/s. Font: Pròpia.

En les figures 6.12 i 6.13. s'observen els senyals dels dos diferents comparadors (canal CH2) a la màxima velocitat.

Després del comparador es disposa de dos senyals digitals diferents. Per unificar aquests senyals en un de sol s'ha utilitzat un biestable SR (Set - Reset) asíncron. En el nostre cas, enlloc de fer servir dos portes lògiques NOR, s'han fet servir quatre portes NAND.

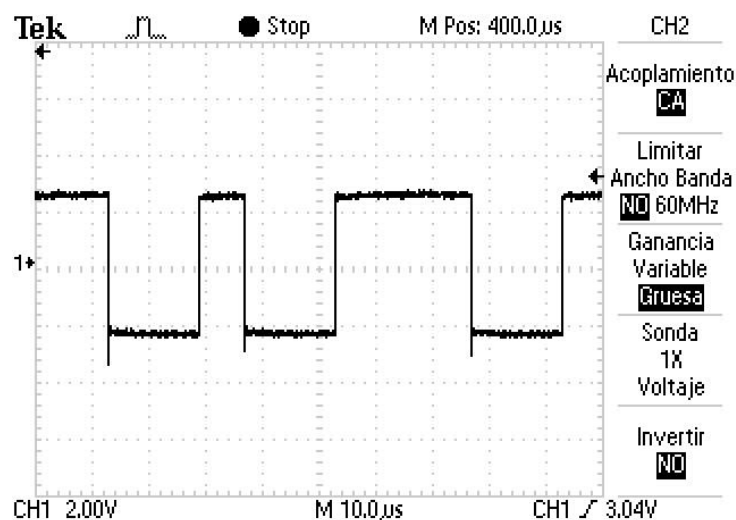


Figura 6.14. Senyal digital final a 125 Kbit/s. Font: Pròpia.

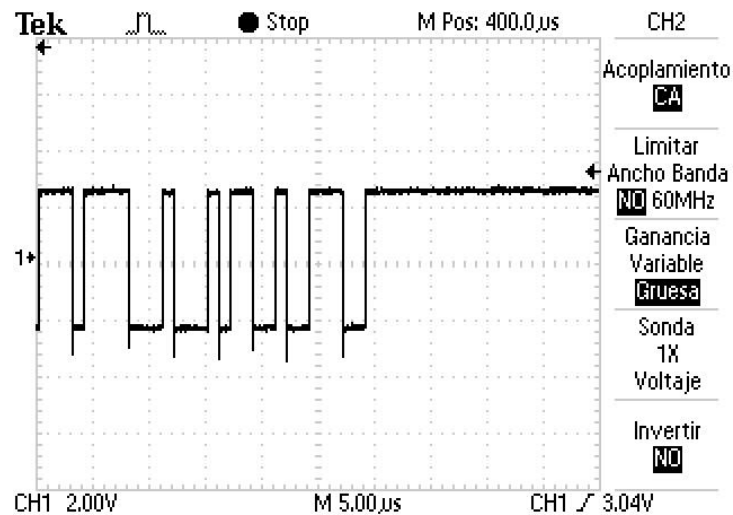


Figura 6.15. Senyal digital final a 1 Mbit/s. Font: Pròpia.

A les captures de l'oscil·loscopi (Fig. 6.14 i 6.15) es mostren el senyal digital un cop unificat el dels dos comparadors a les velocitats mínimes i màximes, veient així que funciona correctament en tot el rang de velocitats que s'ha posat com a objectiu.

Per últim, igual que s'utilitza en els mòduls d'un bus CAN, s'ha fet servir un transceiver CAN per transformar el missatge protocol als nivells de tensió típics del bus CAN a partir d'aquest senyal digital 0-5V.

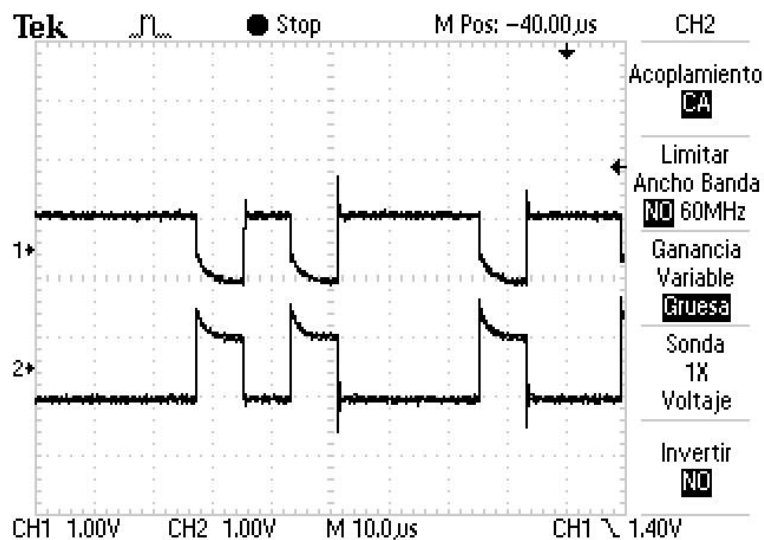


Figura 6.16. Senyals CAN High i Low del dispositiu a 125 Kbit/s. Font: Pròpia.

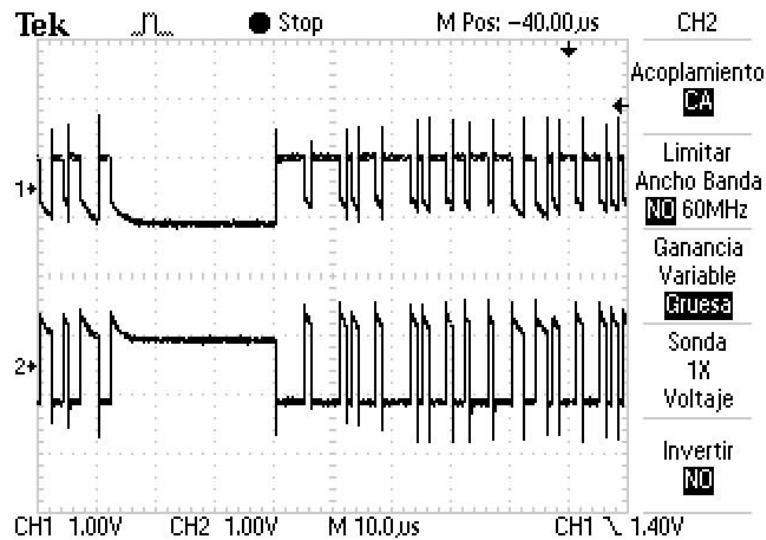


Figura 6.17. Senyals CAN High i Low del dispositiu a 1 Mbit/s. Font: Pròpia.

En les imatges 6.16 i 6.17 es poden veure els senyals CAN High (canal CH1) i CAN Low (canal CH2) finals que componen el missatge CAN que el nostre dispositiu dissenyat capta del bus CAN.

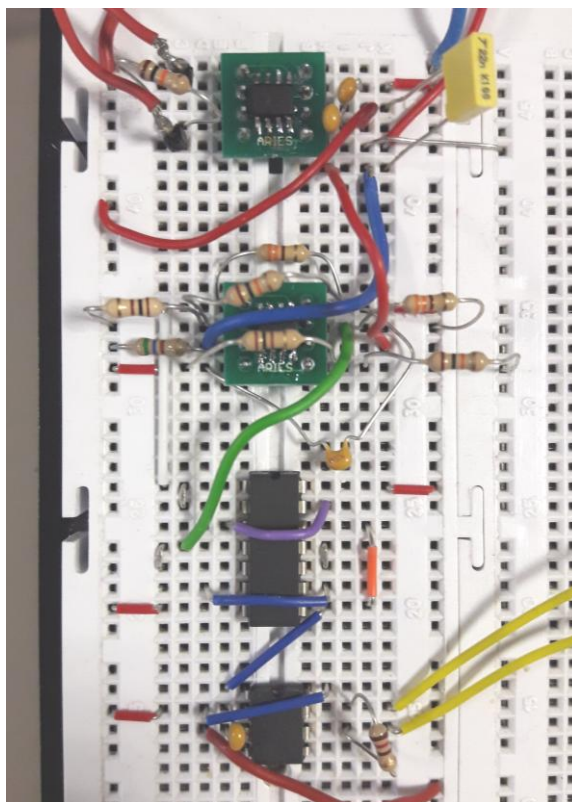


Figura 6.18. Elements del dispositiu en Protoboard . Font: Pròpia.

Un cop dissenyat el dispositiu s'ha soldat tot els elements inicialment provats en la *Protoboard* (Fig. 6.18) en la placa de forats.

Durant els assajos en la *Protoboard* s'ha utilitzat la font d'alimentació per proporcionar el voltatge negatiu que l'amplificador necessita per treballar en mode *dual supply*, cosa que s'ha necessitat ja que en el mode *single supply* l'amplificador no arribava a treballar bé a altes velocitats.

Per arribar a aquesta tensió a la placa de forats, s'ha utilitzat el convertidor de tensió ICL7662, que passarà els 5 V subministrats pel convertidor sèrie USB-UART als -5 V necessaris.

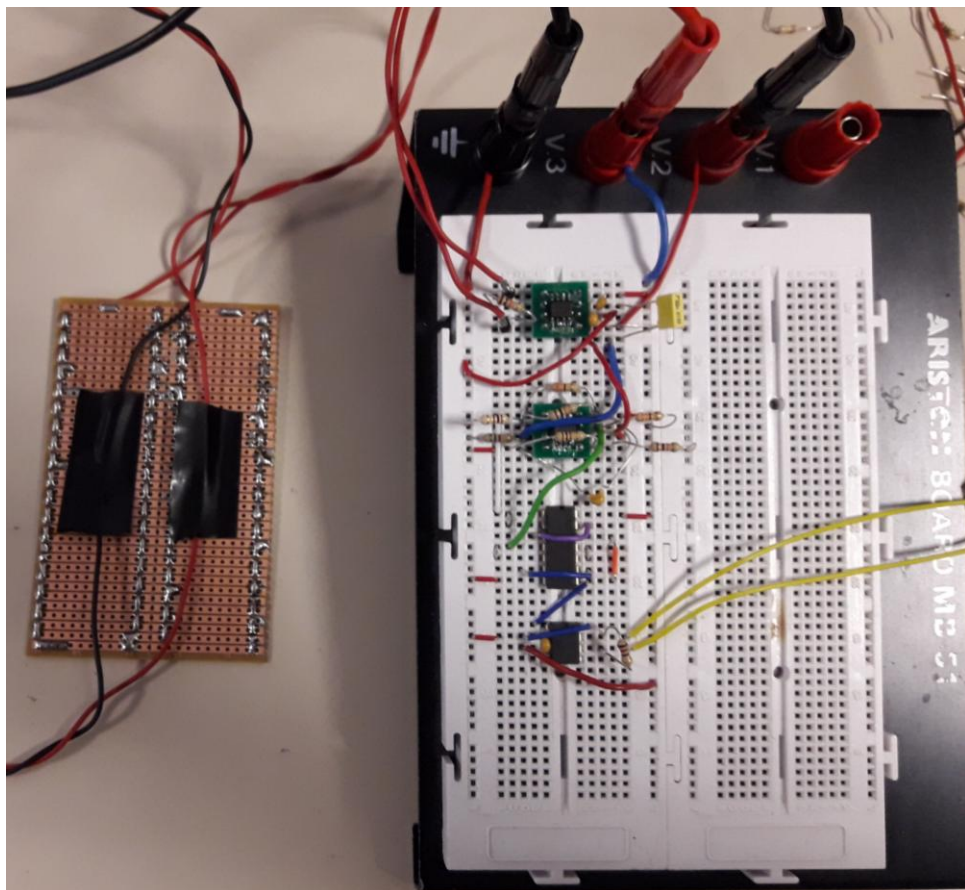


Figura 6.19. Dispositiu dissenyat. Font: Pròpia.

6.2. Software

Un cop s'ha soldat el microcontrolador juntament amb altres elements necessaris com connectors, perifèrics o rellotge oscil·lador de quars en la placa de forats ja definitiva (Fig. 5.2), s'ha passat a la part de programació del microcontrolador.

Per fer-ho, s'ha utilitzat un programa d'ordinador anomenat MPLAB IDE, que ofereix un entorn per al desenvolupament d'aplicacions per aquest tipus de microcontroladors dins d'un sistema més ampli.

Aquest programa utilitza el llenguatge C i s'ha utilitzat un codi fet en un TFG anterior [7] per a la programació, modificant-lo i adaptant-lo a les necessitats d'aquest projecte.

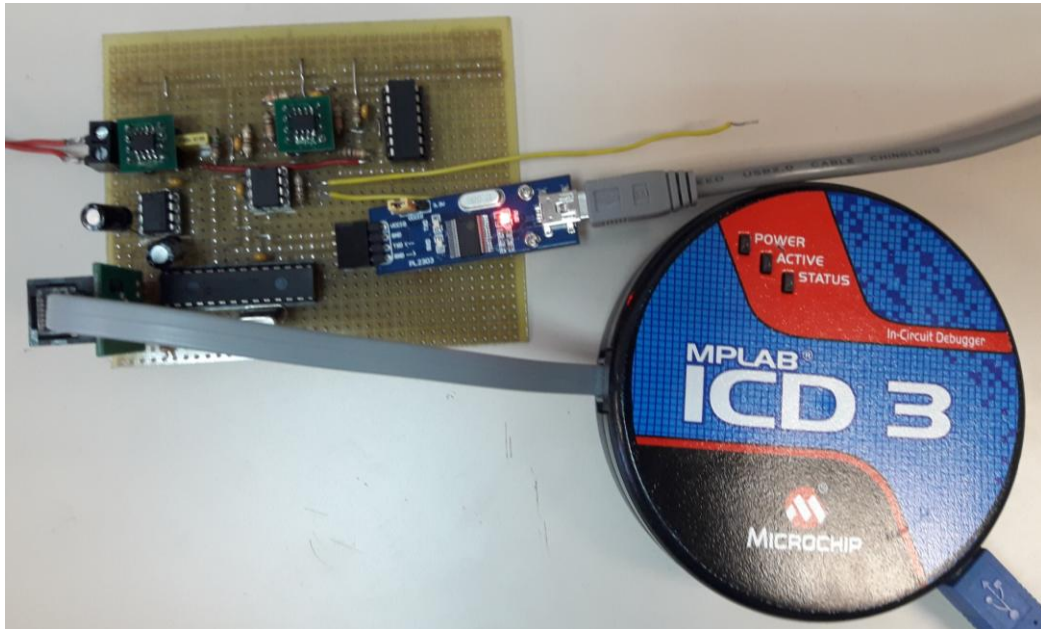


Figura 6.20. Connexions entre els elements durant la programació. Font: Pròpia.

En el projecte d'on s'ha tret el codi l'objectiu era crear un dispositiu que actués com a porta d'enllaç CAN - Bluetooth. Per tant, el microcontrolador estava programat per haver-hi una comunicació bidireccional, és a dir, tant per enviar missatges des del bus CAN a l'ordinador a través del Bluetooth com al revés.

En aquest projecte, en canvi, la comunicació només té una direcció: del dispositiu a l'ordinador. El que es vol es visualitzar a l'ordinador els missatges que el dispositiu capta del bus CAN.

El canvi més important ha estat canviar el mode en el que treballa el controlador de bus CAN, del mode Normal (permetent enviar i rebre missatges) en el que estava treballant per el mode Listen en el qual només deixa rebre i llegir missatges (Fig. 6.21)

```

        CANSetOperationMode(CAN_OP_MODE_LISTEN);
    // CANSetOperationMode(CAN_OP_MODE_NORMAL);

```

Figura 6.21. Canvi de mode en el codi de programació. Font: Pròpia.

Un altre canvi que s'ha fet en el codi és a la funció d'interrupció de l'acció. Per detectar si el missatge que entra es pot processar, ha de ser un missatge vàlid (per tant, si no és vàlid, no passarà res) i si ho és i la llargada del missatge és inferior a 9 bytes, el missatge serà processat (Fig. 6.22).

```

if(CANIsRxReady())                // a CAN message is received
{
    if(CANReceiveMessage(&identificador, MensajeRecibido, &LongMensaje, &RX_FLAGS))
    {
        if(COMSTATbits.RXB0OVFL==1) COMSTATbits.RXB0OVFL;
        if(COMSTATbits.RXB1OVFL==1) COMSTATbits.RXB1OVFL;
        //if((RX_FLAGS&CAN_RX_OVERFLOW)==CAN_RX_OVERFLOW)
        if((RX_FLAGS&CAN_RX_INVALID_MSG)==CAN_RX_INVALID_MSG)
        {
            //COMSTAT=0x00;
            //CANSetOperationMode(CAN_OP_MODE_CONFIG); // entering CAN config mode
            //if(COMSTATbits.RXB0OVFL==1) COMSTATbits.RXB0OVFL;
            //if(COMSTATbits.RXB1OVFL==1) COMSTATbits.RXB1OVFL;
            //CANSetOperationMode(CAN_OP_MODE_LISTEN);
            Nop();
        }
    }
    else
    {
        if(LongMensaje<9)
            CANprocessMessageRX(); // message processing
    }
}

```

Figura 6.22. Canvi en el codi de programació. Font: Pròpia.

7. Validació

La validació del dispositiu s'ha realitzat en diferents etapes de manera progressiva a mesura que s'avançava en la construcció d'aquests. Tant pel hardware com pel software modificat s'han fet diferents comprovacions per assegurar el correcte funcionament del dispositiu.

7.1. Simulació

Un cop confeccionades les antenes, abans de començar el muntatge en la *protoboard*, s'han fet unes simulacions de l'amplificador d'instrumentació amb el programa LTSpice[8], un software gratuït que implementa un simulador de circuits electrònics. Aquestes simulacions s'han utilitzat per veure si l'amplificador escollit era l'adient per dur a terme l'amplificació necessària.

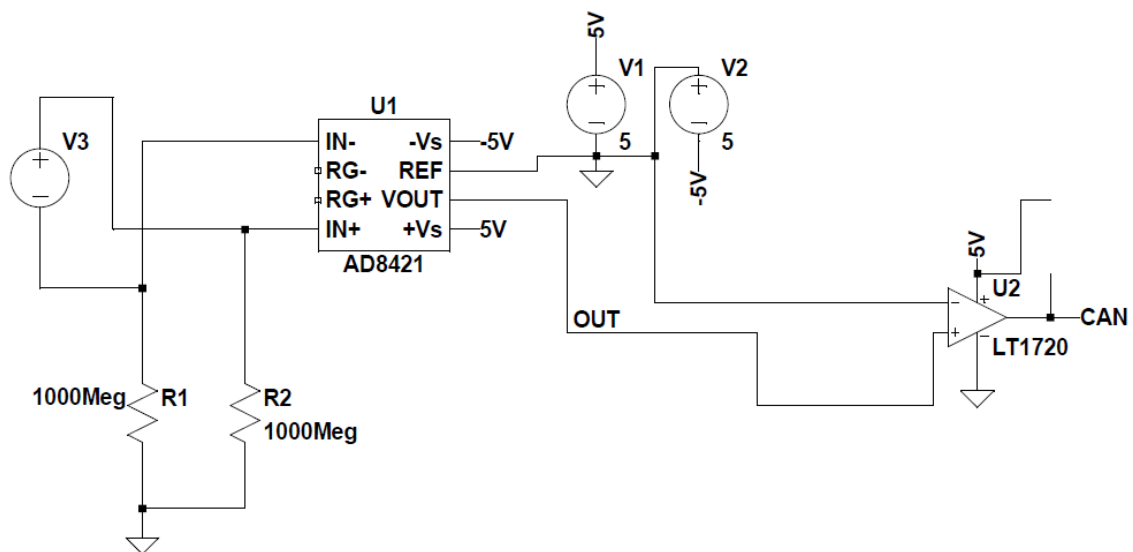


Figura 7.1. Esquemàtic simulació LTSpice. Font: Pròpia.

El software LTSpice és de l'empresa Linear Technology, part de Analog Devices. És per això que l'amplificador simulat és el mateix que s'ha fet servir per el dispositiu, ja que també pertany a Analog Devices, i gràcies a aquesta simulació es va poder assegurar el bon funcionament d'aquest abans del muntatge. En canvi, pel muntatge del comparador, s'ha utilitzat un de diferent al de la simulació.

Com a senyal d'entrada per l'amplificador s'ha utilitzat una font d'ones quadrades per simular la forma dels missatges CAN. La simulació s'ha utilitzat per mirar si l'amplificació suportava les freqüències que s'han necessitat pel disseny del dispositiu i a quins nivell de tensió es podia amplificar jugant amb el guany.

7.2. Longitud d'antena

La validació de les antenes s'ha dut a terme abans de seguir amb el disseny de la placa principal com ja s'ha mostrat en l'apartat anterior. Això s'ha fet a través de l'oscil·loscopi, observant el senyal captat per elles.

Un cop confeccionades i validades les antenes i amb la placa principal dissenyada, s'ha comprovat si la llargada d'aquestes influïen molt en el senyal resultant. Per això, de la mateixa manera que abans, s'ha utilitzat l'oscil·loscopi per observar el senyal a diferents llargades d'antena. Les proves s'han fet soldant 1/4, 1/2, 3/4 i la totalitat de la llargada possible d'antena i amb el dispositiu muntat en la *proto-board*.

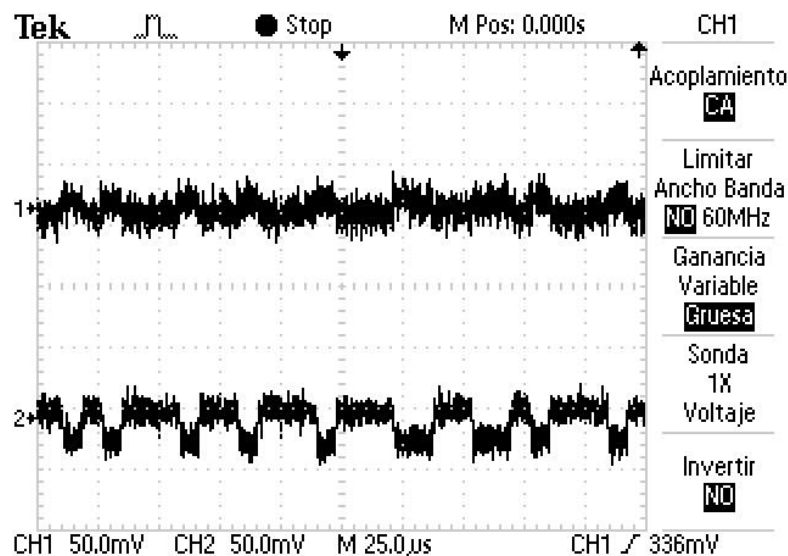


Figura 7.2. Senyals de 1/4 d'antena a 125 Kbit/s. Font: Pròpia.

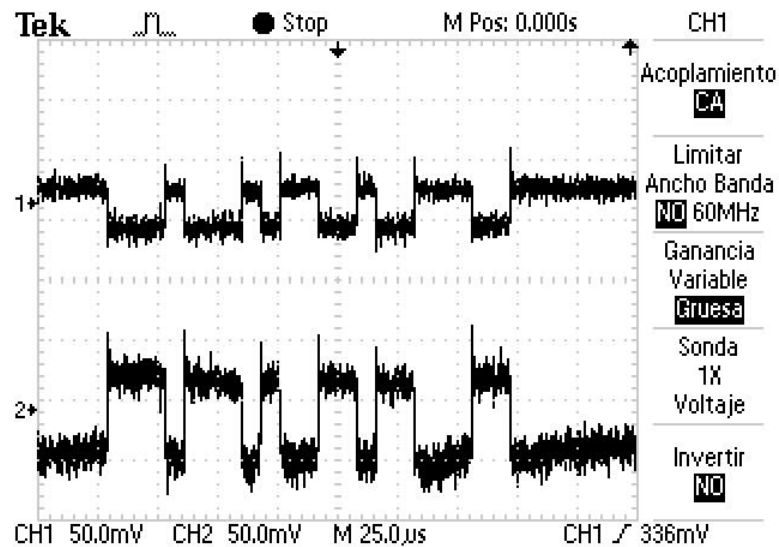


Figura 7.3. Senyals del total d'antena a 125 Kbit/s. Font: Pròpia.

La primera prova ha estat a la velocitat inferior. En la Fig. 7.2 s'observa el senyal CAN High (CH1) i CAN Low (CH2) que l'antena capta del bus CAN quan es fa servir 1/4 d'ella. En la Fig. 7.3 es mostra el mateix quan es fa servir la totalitat de l'antena. Amb l'ajuda d'aquesta prova es pot veure com més llarga sigui l'antena més tensió té el senyal que aquesta capta.

També s'ha provat a màxima velocitat ja que és la més crítica i la que més problemes ha donat durant el projecte.

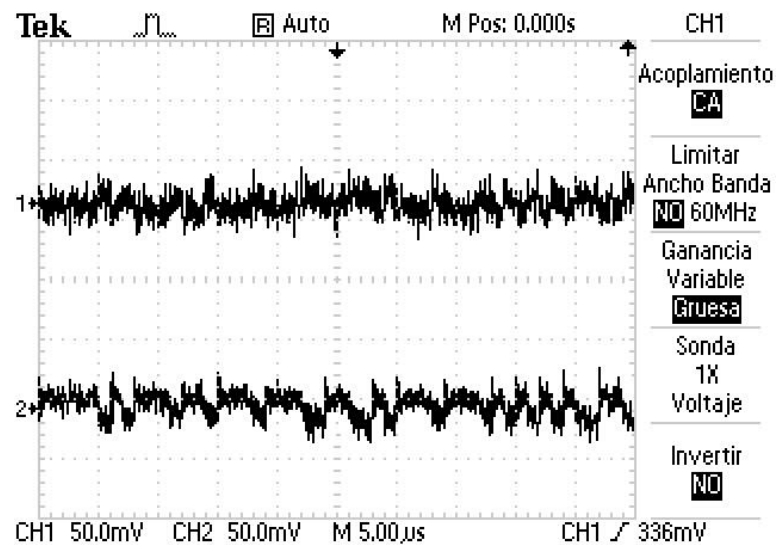


Figura 7.4. Senyals de 1/4 d'antena a 1 Mbit/s. Font: Pròpia.

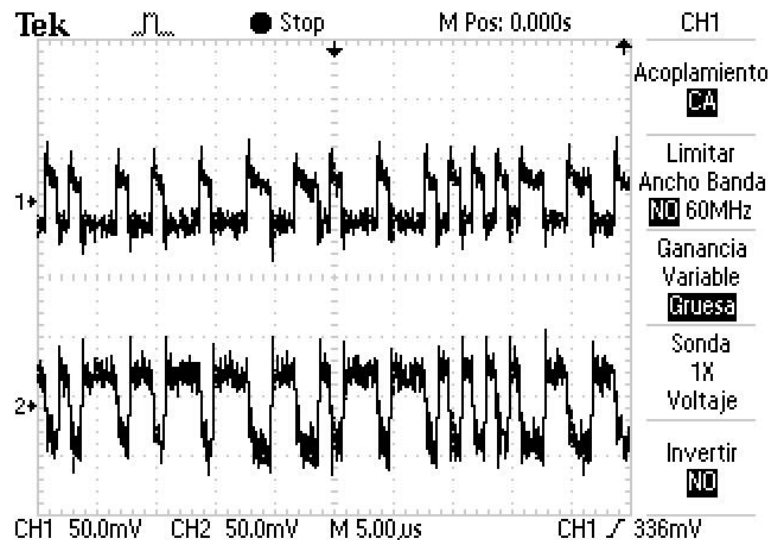


Figura 7.5. Senyals del total d'antena a 1 Mbit/s. Font: Pròpia.

En les Fig. 7.4 i 7.5 es pot observar com, no només augmenta la tensió del senyal, si no que ens ajuda a diferenciar millor la forma del missatge CAN, cosa que ajuda molt quan es treballa a una velocitat tant alta.

A partir d'aquestes proves, i vist el resultat, s'ha treballat amb la totalitat de l'antena.

7.3. Dispositiu

La validació del dispositiu final s'ha dut a terme un cop la placa principal ha estat soldada i el microcontrolador programat. Per fer-ho, s'ha utilitzat diferent material i software. Totes les proves de validació s'han fet a quatre velocitats diferents (125 kbit/s, 250 kbit/s, 500 kbit/s i 1 Mbit/s), assegurant així que el dispositiu funciona dins de tot el rang de velocitats que s'ha posat com a objectiu.

7.3.1. USBtin

Per poder validar el funcionament del dispositiu, és essencial tenir missatges CAN. De la mateixa manera que s'ha fet durant el disseny en la *proto-board*, s'ha utilitzat el hardware USBtin per generar aquests missatges. També s'ha utilitzat el programa USBtinViewer per modificar els missatges com ara canviant les dades, l'identificador, la llargada... o canviant la configuració d'aquests com la velocitat d'enviament, el format en estàndard o estès...

L'enviament de missatges CAN a través de l'USBtin es pot fer de forma ininterrompuda, enviant el mateix missatge un darrere l'altre, o es pot enviar d'un en un.

Durant el disseny del dispositiu, com s'ha comentat abans, els missatges enviats eren ininterromputs, per aconseguir així un senyal constant. En aquesta part del projecte s'ha comprovat de les dues maneres, ja que això era un dels objectius. A la Fig. 7.6 es pot veure la interfície del USBtinViewer quan s'ha enviat un únic missatge a 1 Mbit/s a través del programa. Es pot observar a la part superior com es pot canviar la velocitat dels missatges i a la part inferior com es modifica l'identificador o les dades d'aquests.

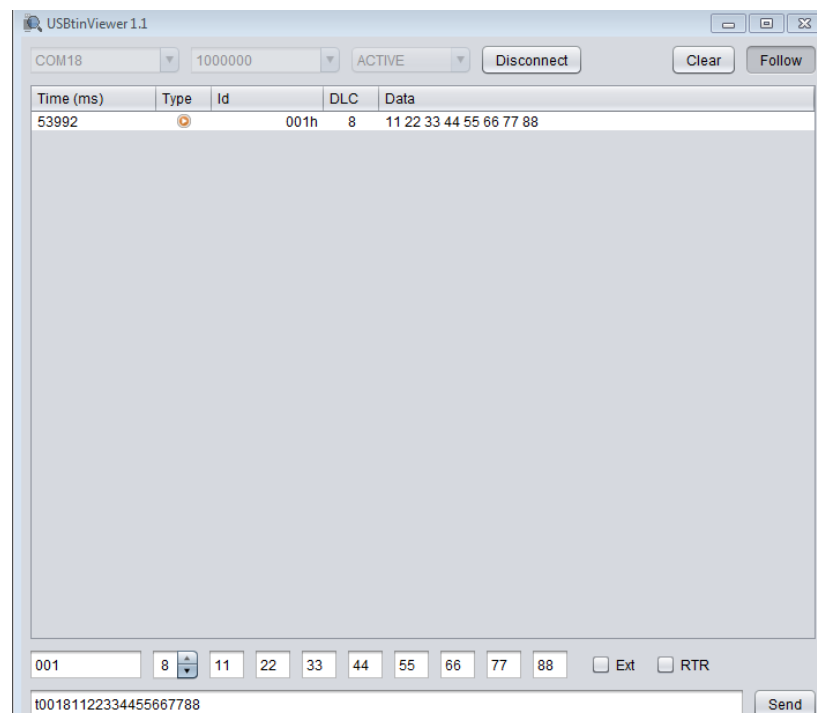


Figura 7.6. USBtinViewer. Font: Pròpia.

7.3.2. Kvaser CanKing

Tot i que la eina CanKing té un gran ventall de funcions com la d'enviar i rebre missatges CAN, filtrar missatges o generar marcs d'error, en aquesta part s'ha fet servir per validar que els missatges que el nostre dispositiu ens capta són missatges de protocol CAN com els que s'envien per l'USBtin o pel programa de *Python*. Connectant els senyals CAN High i CAN Low del dispositiu al Kvaser CanKing, aquest ens mostra pel monitor si els missatges que rep són missatges CAN o si hi ha un error en el senyal i l'interpreta com un missatge erroni.

En la Fig. 7.7, podem veure les connexions que s'han fet per a la validació, on s'observa les antenes del dispositiu dissenyat amb el bus CAN (USBtin) i les sortides CAN (cables grocs) connectades al Kvaser CanKing.

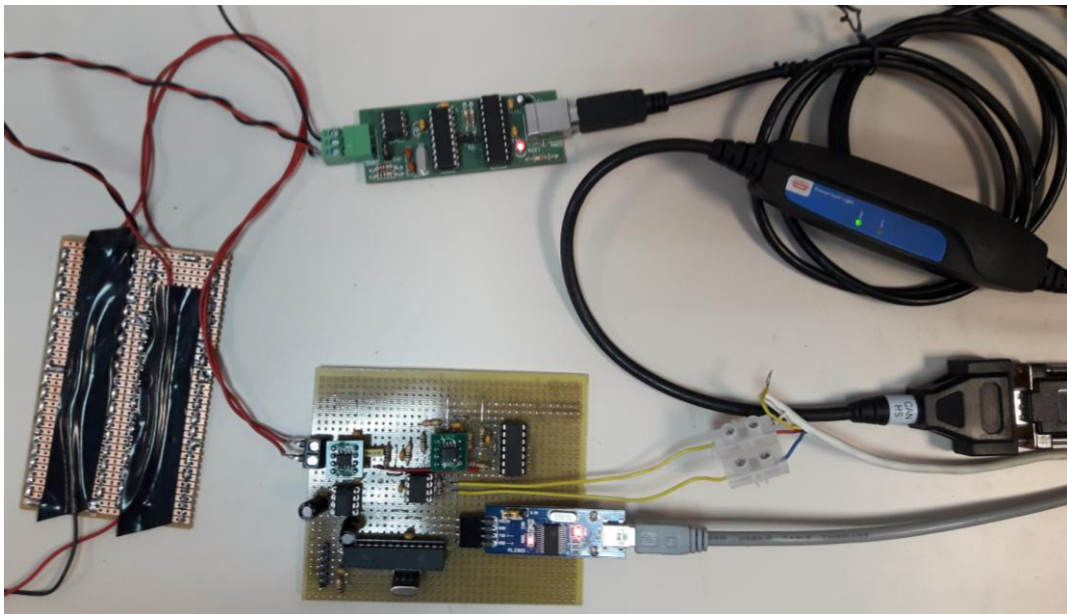


Figura 7.7. Connexions entre el dispositiu i el material necessari per a la validació. Font: Pròpia.

Com es pot veure a la Fig. 7.8, quan s'envia un missatge a 1 Mbit/s, el Kvaser CanKing l'interpreta com a un missatge de protocol CAN rebut correctament (Rx Messages). Si aquest missatge s'hagués detectat com a error hauria d'augmentar el total del comptador *Error Frames*.

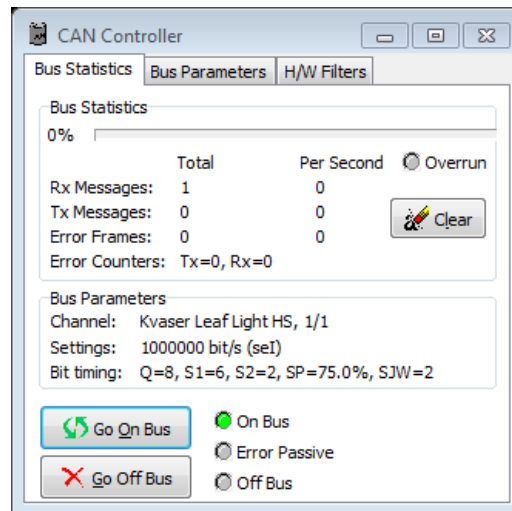


Figura 7.8. Monitor Kvaser CanKing a 1 Mbit/s enviant un missatge. Font: Pròpia.

D'altre manera, si es deixa el missatge en forma bucle, enviant-se contínuament, es pot veure a la Fig. 7.9, que el CanKing interpreta tots els missatges com a correctes, indicant el nombre total de missatges enviats fins que s'interromp el flux i la velocitat a la que es reben (missatges per segon enviats).

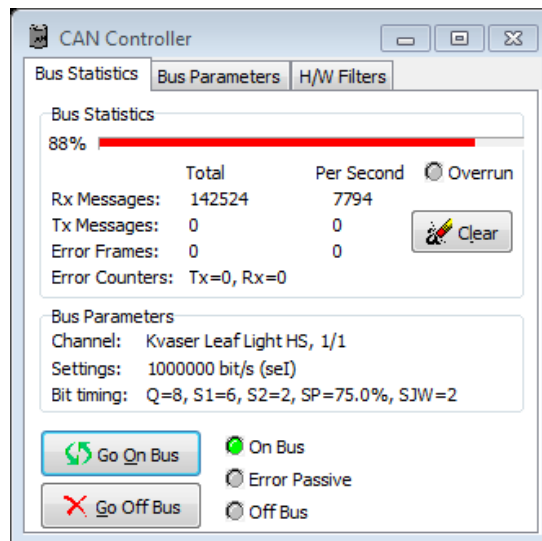


Figura 7.9. Monitor Kvaser CanKing a 1 Mbit/s enviant missatge en bucle. Font: Pròpia.

D'aquesta manera, s'ha assegurat que el senyal que es rep del dispositiu és un missatge CAN i que es rep correctament a qualsevol de les velocitats dins del rang.

7.3.3. Python

El següent pas ha estat veure si els missatges rebuts són els mateixos que els enviats a través de l'USBtin o d'un programa *Python*. Per fer-ho de manera més ràpida, s'han utilitzat uns programes *Python* desenvolupats en TFGs anteriors en els quals s'han fet alguns canvis.

El primer és un petit programa que permet enviar un nombre finit de missatges CAN en un temps concret. També permet modificar parts del missatge com l'identificador o les dades del propi (Fig. 7.10).

```
import usbTinLib as usb
import time

usb1=usb.USBtin('COM19','BAUD_1M','Normal')
usb1.canOpenChannel()

for i in range(100):
    usb1.canWrite('1',5,['10','13','66','77','AA'],0x0002)
    time.sleep(0.1)

usb1.canClose()
```

Figura 7.10. Programa generador de missatges CAN. Font: Pròpia.

L'altre programa és el que permet veure en l'ordinador el missatges que el dispositiu llegeix per comprovar així si coincideixen amb els missatges enviats. En el projecte d'on s'ha tret aquest programa, l'ordinador es comunicava amb el CAN a través de Bluetooth. En canvi, en aquest projecte, el dispositiu es comunica amb l'ordinador a través de l'USART.

Per això, el primer canvi que s'ha fet en aquest codi és configurar la comunicació amb el dispositiu a través del port sèrie de l'ordinador, enlloc del Bluetooth, important el mòdul *serial* (Fig. 7.11). Amb aquest codi s'obre el port sèrie on estigui connectat el convertidor sèrie USB per poder llegir la informació entrant.

```

import serial
import time

class CanNoCont:
    def __init__(self, portin='COM22'):
        self.serialport = serial.Serial(
            port = portin,
            baudrate = 115200,
            parity = serial.PARITY_NONE,
            stopbits = serial.STOPBITS_ONE,
            bytesize = serial.EIGHTBITS
        )
        print("Port Serial is Open")

```

Figura 7.11. Part del codi on es configura el port sèrie. Font: Pròpia.

El següent canvi ha estat modificar la funció *readmessage* per llegir els missatges que entren a través del port sèrie (Fig. 7.12).

```

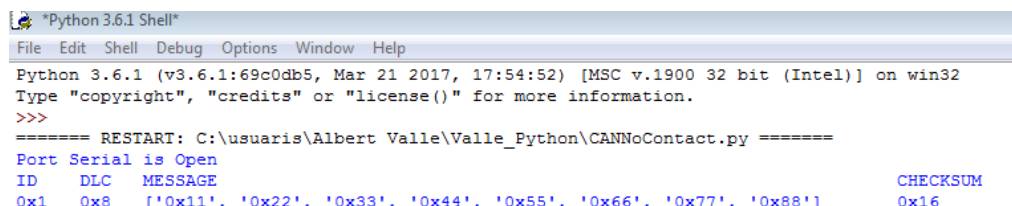
def readmessage(self):
    if self.serialport.inWaiting() > 0:
        rx = ord(self.serialport.read(1))

```

Figura 7.12. Part del codi *Python*. Font: Pròpia.

Per últim, s'ha canviat el que es vol mostrar per pantalla quan s'executa el codi, mostrant informació del missatge que arriba del dispositiu. Es mostra una llista de tots els missatges on es pot observar l'identificador (ID), la llargada que tenen (DLC), les dades dels missatges i el *checksum*.

En la Fig. 7.13 s'observa el visor que mostra els missatges quan s'executa el programa i s'envia un únic missatge a 1 Mbit/s a través del USBtin. El missatge mostrat per ordinador és el mateix que l'enviat (Fig.7.6).



```

Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\usuari\Albert Valle\Valle_Python\CANNoContact.py =====
Port Serial is Open
ID   DLC  MESSAGE                                     CHECKSUM
0x1  0x8  ['0x11', '0x22', '0x33', '0x44', '0x55', '0x66', '0x77', '0x88']  0x16

```

Figura 7.13. Visor dels missatges CAN. Font: Pròpia.

8. Costos del projecte

Per quantificar el cost del projecte, s'ha diferenciat en tres parts. Primer s'ha calculat el cost del dispositiu, des de les antenes fins els elements de la placa de forats. Per una altra banda, s'ha fet un càlcul del cost del material necessari per al desenvolupament d'aquest projecte: soldador, ordinador, softwares... I per últim el cost total de hores, per part de l'enginyer, per dur a terme el projecte. En les següents taules s'hi pot veure el cost de cada part.

Element	Quantitat	Preu/unitat [€]	Preu total [€]
Placa de coure	1	2	2
Placa de forats	1	1,65	1,65
Amplificador d'instrumentació AD8421	1	6,10	6,10
Convertidor de tensió ICL7662	1	2,87	2,87
Comparador TLC352	1	1,27	1,27
Biestable 74HC00N	1	0,35	0,35
Transceptor CAN PCA82C250	1	1,44	1,44
Microcontrolador PIC18F258	1	5,80	5,80
Resistències de pel·lícula de carboni	10	0,25	2,50
Condensadors 100nF ceràmics multicapa	8	0,03	0,24
Condensadors 22nF de polièster	1	0,22	0,22
Condensadors 10µF electrolítics d'alumini	2	0,20	0,40
Condensadors 22pF ceràmics de disc	2	0,05	0,10
Cristall oscil·lador 20MHz	1	0,80	0,80
Bloc terminal PCB	1	0,30	0,30
Adaptadors SOIC-8	2	2,68	5,36
Sòcol (8, 14 i 28 pins)	6	0,20	1,20
Perifèrics	2	0,25	0,50
		TOTAL	33,10

Figura 8.1. Costos del material per al dispositiu. Font: Pròpia.

Element	Preu total [€]
Soldador	150
Oscil·loscopi	500
Font d'alimentació	300
USBtin	34,90
Kvaser Canking	297
Programador ICD3	200
Adaptador USB-UART	4
PC HP 630	400
TOTAL	1.885,9

Figura 8.2. Costos del material utilitzat durant el desenvolupament. Font: Pròpia.

Etapa	Hores	Preu total [€]
Soldadura i validació antenes	5	200
Disseny dispositiu en <i>ProtoBoard</i>	160	6.400
Soldadura de la placa principal	20	800
Modificació del codi	15	600
Validació dispositiu	60	2.400
Memòria	100	4.000
TOTAL	360	14.400

Figura 8.3. Costos relacionats amb la feina de l'enginyer. Font: Pròpia.

En total, el cost del projecte ha estat 16.319 €, sumant els costos dels tres diferents àmbits: cost del dispositiu, cost del material emprat i hores emprades per l'enginyer.

Element	Preu total [€]
Dispositiu	33,10
Material	1.885,90
Treball enginyer	14.400
TOTAL	16.319

Figura 8.4. Cost total del projecte. Font: Pròpia.

9. Impacte ambiental

La implementació del protocol CAN en els automòbils va suposar una gran disminució en cablejat dins d'un cotxe ja que tots els mòduls de l'electrònica que necessitaven comunicar-se entre ells ho feien a través dels cables que els unien. Tot això va passar a fer-se a través de dos únics cables trenats: bus CAN.

Quan es vol accedir al bus CAN per fer un diagnòstic elèctric per detectar errors o per una presa de dades del vehicle, es fa a través d'un connector especial anomenat OBD, que sol trobar-se sota el volant. Amb un adaptador es pot connectar un ordinador, smartphone o similar i així ens assabentarem de tot el que es cou a l'interior del nostre cotxe.

Amb el dispositiu dissenyat aquesta connexió no necessitaria de connectors ni adaptadors més el cablejat que suposa. Només és necessita el cable USB/UART per connectar el dispositiu al ordinador, cosa que comporta un petit estalvi en plàstic i coure dels cables en comparació amb el que va suposar la implementació del bus CAN, o també per part del connector i adaptador.

10. Conclusions

L'objectiu del projecte era el disseny i la construcció d'un dispositiu capaç de captar els missatges que passen per un bus CAN de forma que ho fes sense contactes amb aquest. La idea era fer-ne un de similar als dispositius ja comercialitzats, començant el disseny des de zero, a nivell de funcionalitat però millorant prestacions com la velocitat de missatge a la que pot treballar. Es pot afirmar que això s'ha aconseguit, ja que el dispositiu construït funciona correctament.

Com es pot veure a l'apartat de validació del document, s'han comprovat que els objectius fixats en un inici s'han aconseguit amb èxit. El dispositiu pot treballar correctament fins a una velocitat de 1Mbit/s, velocitat a la que pocs equips comercials treballen. Apart, s'ha comprovat com el sistema incorporat per poder llegir els missatges a través de l'ordinador funciona correctament davant les diferents maneres d'emissió d'aquests, ja sigui enviant un únic missatge com un gran número d'ells.

D'altre banda, el cost del dispositiu es similar al de l'oferta obtinguda, aconseguint així millorar el producte sense aconseguir augments de costos significatius. És cert que el dispositiu no té el disseny final que es necessitaria per comercialitzar-lo, i per tant el cost podria variar una mica.

Per tant, un possible següent pas que seguiria el projecte si es disposés de més temps, seria buscar aquest disseny final construint el dispositiu en una placa de PCB i posant-li un encapsulat, de manera que fos més pràctic a l'hora d'utilitzar-lo.

11. Agraïments

Voldria agrair enormement al director i tutor d'aquest projecte Juan Manuel Moreno Eguílaz per la seva infinita ajuda durant la realització d'aquest. Donar-li les gràcies pels coneixements aportats, sobretot d'electrònica, dels quals he après molt, també per l'ajuda proporcionada en altres àmbits i per tenir les portes obertes tant al laboratori com al despatx sempre que ho he necessitat.

12. Bibliografia

Referències bibliogràfiques

[1] Informació sobre el producte de Technoton. Data de consulta: 14 de març de 2018.

https://www.jv-technoton.com/contactless_reader/cancrocodile

[2] Informació sobre el producte de GPS4NET. Data de consulta: 14 de març de 2018.

<http://gps4net.com/can-obd2-interfaces/g4n02tap-wire-tapping-contactless-reader-can-bus-jbus-fms-j1939-j1708-j1850-j1587-iso11992-iso11783-obd2-telematics-standard.html>

[3] Informació sobre el producte de Teltonika. Data de consulta: 14 de març de 2018.

<http://teltonika.lt/es/product/simple-can/>

[4] Crosstalk. Data de consulta:

BLACKWILL, G.R. *The electronic packaging handbook*, Chapter 8: Interconnects.

<https://en.wikipedia.org/wiki/Crosstalk>

[5] USBtin i USBtinViewer. Data de consulta: 5 d'abril de 2018.

<http://www.fischl.de/usbtin/>

[6] Kvaser CanKing. Data de consulta: 5 d'abril de 2018.

<https://www.kvaser.com/canking/>

[7] ÉRRAEZ, G. *Disseny i implementació d'un convertidor CAN*. Data de consulta: 1 de Març 2018.

<https://upcommons.upc.edu/handle/2117/85261>

[8] LTSpice. Data de consulta: 10 de gener 2018.

<http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>

Bibliografia complementària

Informació del protocol CAN:

- CORRIGAN, S. *Introduction to the Controller Area Network (CAN)*. Texas Instrument, Agost 2002.

<http://www.ti.com/lit/an/sloa101b/sloa101b.pdf>

- DI NATALE, M. *Controller Area Network*. Scuola Superiore S. Anna-Pisa

http://retis.sssup.it/~marco/files/lesson9-controller_area_network.pdf

- COOK, J.A., FREUDENBERG, J.S. *Controller Area Network (CAN)*. EECS 461, Tardor 2008.

https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf

Informació del microcontrolador PIC18F258:

- MICROCHIP TECHNOLOGY INC. *PIC 18FXX8 Datasheet. 28/40-Pin High-Performance, enhanced Flash Microcontrollers with CAN Module.*

<http://ww1.microchip.com/downloads/en/DeviceDoc/41159d.pdf>