

Széchenyi István Egyetem
Gépészmérnöki, Informatikai és
Villamosmérnöki Kar

SZAKDOLGOZAT

Kecskeméti István
Villamosmérnöki BSc szak

2019

[Gerincen:] Kecskeméti István, 2019

SZAKDOLGOZAT

LIN kommunikációs eszköz tervezése és gyártása

Kecskeméti István

**Villamosmérnöki BSc szak
Automatizálási specializáció**

2019

Feladat-kiíró lap szakdolgozathoz

Hallgató adatai

Név: Kecskeméti István

Neptun-kód: FDIP6P

Szak: Villamosmérnöki BSc szak

Specializáció: Automatizálás specializáció

Tagozat: nappali

A szakdolgozat adatai

Kezdő tanév és félév: 2016/17 1

Nyelv: magyar

Típus: nyilvános

LIN kommunikációs eszköz tervezése és gyártása

Feladatok részletezése:

- 1) Az ismertebb autóiipari kommunikációs rendszerek bemutatása. LIN protokoll felépítésének, működésének ismertetése.
- 2) Egy új LIN kommunikációs eszköz megvalósítása, az igények figyelembevételével. Az eszköz megtervezése mind logikai szinten mind áramköri kapcsolás szinten.
- 3) A megtervezett eszköz legyártása, összeszerelése.
- 4) A kész eszközhöz tesztösszeállítás elkészítése, illetve tesztelése.

Belső konzulens adatai

Név: Csizmadia Miklós

Tanszék: Automatizálási tanszék

Beosztás: Egyetemi tanársegéd

Külső konzulens adatai

Név:

Munkahely:

Beosztás:

Győr, 2019.11.12

belső konzulens
Csizmadia Miklós

külső konzulens

Automatizálási Tanszék
Dr. Ballagi Áron

Nyilatkozat

Alulírott, [Kecskeméti István (FDIP6P) Villamosmérnöki BSc szak] szakos hallgató kijelentem, hogy a [*LIN kommunikációs eszköz tervezése és gyártása*] című szakdolgozat feladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Győr, 2019.11.28

Kecskeméti István

Kivonat

LIN kommunikációs eszköz tervezése és gyártása

Az utóbbi években megfigyelhető a járműiparban, hogy a környezettudatosság és a kényelem a fő fejlesztési irány. Gépjárműveink egyre több eszközzel vannak felszerelve, amik az autó optimális működéséért, a biztonságos és kényelmes utazásért felelnek. Ezeket az eszközöket viszont össze kell fogni egy rendszerbe, viszont itt kezdődnek a problémák.

A kezdetben ezek az új funkciók nem igényeltek sokat, ezért egyszerű protokollok használata is elegendő volt, de a megsokszorozódott eszközsám miatt már rengeteg elvárásnak kell megfelelniük ezeknek a kommunikációs rendszereknek. A legnagyobb problémát a rengeteg átláthatatlan vezeték jelentette, ezért hozták létre a mérnökök az ezt leegyszerűsítő buszrendszereket. A leggyakoribbak a Controlled Area Network (CAN), a Local Interconnect Network (LIN), a Flexray és a Media Oriented Transport System (MOST). Mindegyiknek megvannak az előnyei és hátrányai, ezért az autókban a funkciókat fel kellett osztani, hogy milyen igényeik vannak ezektől a rendszerektől. A kábelezés mértéke jelentősen csökkent, a különböző rendszereknek köszönhetően. A teljes autó minden eszközéről lehet tudni az aktuális állapotát, ezért a hibakeresés is egyszerűbb.

Dolgozatomban egy egyedi megoldást mutatok be a standard LIN rendszer helyettesítésére, amellyel még tovább lehet csökkenteni a kábelezés mértékét, megtartva a biztonságos és stabil kommunikációt. Az új LIN rendszer a standardhoz hasonlóan Master-Slave felépítésű, ebből a slave eszköz elkészítését mutatom meg lépésről lépésre, a kapcsolási rajzoktól, a gyártáson át egészen a tesztelésig és beüzemelésig.

Abstract

The designing and development of a LIN communication device

In recent years it has been observed, that the major direction in automotive development is around environmental awareness and comfort. Our vehicles are equipped with more and more devices, that are responsible for the car's optimal operation, for safety and the comfort of travel. These devices need to be linked into a system, and this is where the problems begin.

In the beginning these new functions did not require much, therefore the use of simpler protocols were enough, but as the devices multiplied, these communication systems had to live up to a lot of expectations. The main issue is the number of wiring, that is why engineers developed bus system to simplify this. The most common are the Controlled Area Network (CAN), the Local Interconnect Network (LIN), the Flexray and the Media Oriented Transport System (MOST). Each one has its advantages and disadvantages, for that reason the functions had to be divided by their requirements from the communication system. Because of the different systems the extent of wiring has significantly decreased. Every device in the car is diagnosable, therefore troubleshooting is much easier.

In my thesis, I am going to present a unique solution to replace the standard LIN system, which reduces the cable harness size even more, but keeping the communication safe and stable. The new LIN system similar to the standard consists of a Master and a Slave device. I am going to show step by step how to develop the slave module from the blueprint through manufacturing, assembly and testing.

Tartalomjegyzék

1. Bevezetés	1
1.1. SZEnergy Team.....	1
1.2. Shell Eco-marathon Europe	2
1.3. SZEmission versenyautó.....	3
2. Autóipari buszrendszerek.....	4
2.1 Controlled Area Network.....	5
2.2 Local Interconnect Network.....	8
2.3 Flexray	9
2.4 Media Oriented System Transport	12
2.5 Összehasonlítás	14
3. Áramkör tervezése	14
3.1 Rendszerterv.....	14
3.2 A kiválasztott vezérlők.	16
3.2.1 SIG60	17
3.2.2 SIG61	19
3.3. Kapcsolási rajz	22
3.4. Nyomtatott áramköri tervek.....	26
4. Gyártás, beültetés.....	30
5. Tesztelés.....	35
6 Konklúzió.....	38
7. Fejlesztési lehetőségek.....	39
Irodalomjegyzék.....	I
Ábrajegyzék.....	II
Köszönetnyilvánítás	III
Mellékletek.....	IV

1. Bevezetés

1.1. SZEenergy Team



A SZEenergy Team a Széchenyi István Egyetem oktatóiból és hallgatóiból álló lelkes csapata, melyet a Közúti és Vasúti Járművek Tanszék alternatív energiaforrásokat kutató törekvései hívtak életre. A csapat célja, hogy az alternatív energiával hajtott elektromos járművek hatékonyságát fejlessze és a technológia életképességét bizonyítsa. Tevékenységeik mellett, az általuk fejlesztett járművek a Shell Eco-marathon Europe elnevezésű európai szintű energiatakarékossági versenyen bizonyítanak 2008 óta.

A csapatban átlagosan 25 hallgató tevékenykedik. Közülük legtöbben a versenyautóért felelősek, de menedzsment szakos hallgatók is kiveszik részüket a csapat tevékenységéből, hiszen nélkülük nem működhetne ilyen eredményesen a csapat. A szakmai feladatok elvégzéséért gépész, járműmérnök, mechatronikai mérnök, mérnökinformatikus és villamosmérnök hallgatók felelősek. Évről évre ők dolgozzák ki az új fejlesztéseket, készítik el a gyártási dokumentációkat, gyárttatják le az alkatrészeket és végül szerelik össze a versenyautót.

A csapathoz 2017-ben csatlakoztam, azóta vagyok az elektronika részleg tagja. Részlegünk feladata az autó elektronikáját megtervezni, kivitelezni, és karbantartani. Az első évemben a csapatnál az akkori autónak a rendszerével ismerkedtem meg, mint hardver, mint szoftver oldalról. Ekkor vált világossá számomra, hogy az áramkörök megtervezését, és összeszerelését szeretném feladatként megkapni.

1.2. Shell Eco-marathon Europe



Shell
Eco-marathon

A Shell Eco-marathont a világ legnagyobb energiatakarékossági versenyeként tartják számon.

A verseny alapját egy 1939-ben, Shell tudósok által kötött fogadás adja. A fogadás tárgya szerint az nyert, akinek az autója a legkevesebb tüzelőanyaggal a legnagyobb távolságot tette meg. A nyertes autó nem kevesebb, mint 21,12 km-t tett meg egy liter benzin felhasználásával.

A versenyt évről évre megrendezték, korábban a nagyobb autógyártók részvételével (Fiat, Opel) majd, később kisebb csapatok is összemérhették autóik fogyasztását.

Manapság a verseny minden évben megrendezésre kerül, szinte kizárólag egyetemi csapatok részvételével. A versenyen több mint 200 csapat, több ezer hallgatója vesz részt, akik mind azon dolgoznak, hogy az Ő autójuk érje el a legalacsonyabb fogyasztást a 3 napos verseny során. A versenyen külön kategóriában indulnak a különböző meghajtású versenyjárművek, így külön versenyeznek a benzin, dízel, hidrogén, gáz és elektromos üzemű járművek.

A versenyen két különböző autó kategóriát különböztetünk meg: a Prototype és az Urban Concept kategóriát. E két kategória versenye teljes mértékben külön zajlik és teljesen más szabályrendszer vonatkozik rájuk. A Prototype kategória résztvevőinek autója teljes mértékben a fogyasztáscsökkenés jegyében épül. Ezeket az autókat alacsony építési mód, könnyű szerkezet, mindenekelőtt áramvonalas karosszéria és minél alacsonyabb tömegű pilóta jellemzi.

Ezzel szemben az Urban Concept kategóriában induló autók nagymértékben hasonlítanak a hétköznapi személygépjárművekre. A szigorú szabályrendszer előírja, hogy minden olyan kiegészítő eszközzel fel kell szerelni egy autót, mely szükséges a közúton való közlekedéshez, ezért minden autóban kötelező többek között a biztonsági öv, első és hátsó fényszóró, irányjelző, duka, ablaktörlő, kétkörös hidraulikus fékrendszer megléte és használata.

1.3. SZEmission versenyautó

A csapat 2013 óta versenyzett korábbi autójával a SZElectricityvel, azonban a 2018-as versenyen vált világossá, hogy mennyire vagyunk elmaradva az élmezőnyhöz képest. Sajnos az autó elavult karosszériája miatt már nem tudta felvenni a versenyt a rivális csapatok áramvonalasabb és kedvezőbb aerodinamikájú autóival. A 2018-as verseny után kezdődött meg a munka: egy teljesen új autó elkészítése.



1. ábra. Shell Eco Marathon 2019.

A SZEmission autó önhordó karosszériával rendelkezik, ami azt jelenti, hogy nincs szükség belső merevítő vázra. A karosszéria formája alacsony sebességre lett optimalizálva, ez több szimuláción és szélcsatornában is lett tesztelve. Gépészeti szempontból fontos még, hogy az autó teljesen új futóművet és kormányművet kapott. Az autó elektronikai rendszere a SZElectricity alapján lett elkészítve, kiküszöbölve a régi rendszer problémáit. Az új elektronikában a teljesítmény és kommunikációs vonalak teljesen el vannak különítve, elhárítva így az ebből adódó akadályokat, emellett ki kell tudnia szolgálnia az autonóm funkciókhoz szükséges eszközöket.

Az autón belüli kommunikáció lebonyolításáért alapvetően a CAN (Controlled Area Network) és LIN (Local Interconnect Network) hálózatok felelnek. Az autó központi egysége egy National Instruments által gyártott myRIO programozható adatfeldolgozó egység, ami az autó perifériáinak és telemetria rendszerének működéséért felel. A CAN buszon található a motorvezérlő és a BMS (Battery Management System). Ezeknek a paramétereit folyamatosan felügyeli a myRIO. A LIN buszon található az autó összes perifériája. Mivel a csapat UrbanConcept kategóriában versenyez,

ezért az autót fel kell szerelni azokkal az eszközökkel, amik a városi járművekben is megtalálhatóak, ilyenek például: irányjelző, duka, ablaktörlő, fényszóró, féklámpa. Ezeknek az elemeknek kellett egy olyan eszközt kifejleszteni, amivel a LIN hálózaton keresztül a myRIO-ra tudnak csatlakozni.

2. Autóipari buszrendszerek

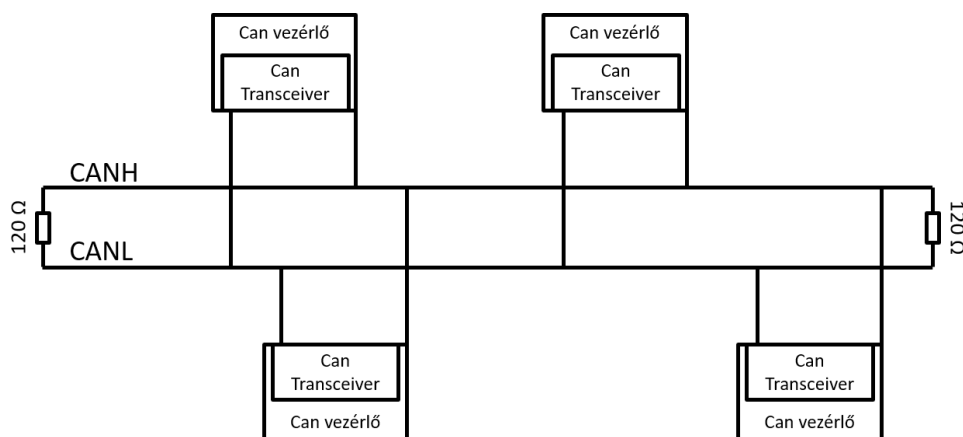
Az autóipar történetében végig követhető, hogy az elektronikai eszközök egyre nagyobb teret nyertek a gépjárművekben. Ma szinte természetesnek tűnik, hogy az autó fel van szerelve vezetéskötő támogatási és szórakoztató rendszerekkel. Azonban nagyjából az 1970-es évek elején ezek az eszközök még ritkaságnak számítottak. Ekkor jelentek meg az első vezérlések, amik a jármű működését segítették elő, ilyenek voltak pl. a motorvezérlések, automata váltó vezérlése stb. Ezt követően jöttek a biztonsági eszközök, amik az utasok épségéért feleltek, később pedig a szórakoztató és kényelmi funkciók. Ahogy egyre több eszköz került az autóba, úgy nőtt a szükséges kábel mennyiség, ez viszont már egyre több problémát jelentett a tervezők számára. A hatalmas kábelkordbácsok mellett, hogy rengeteg helyet foglalnak el, a rendszer átláthatatlanságát, és megbízhatatlanságát is jelentették. Az első fejlesztések ennek a problémának a feloldására, egy központi egység behelyezésével kezdődtek meg. Ez felelt az autó összes rendszeréért, perifériájáért, az autó villamosrendszere csillag topológiában volt összekötve. A kábelezés lényegesen csökkent, ha valamelyik alrendszer meghibásodik, a többi tőle függetlenül tud működni, azonban, ha a fő egységgel van probléma, az a teljes autó leállását eredményezi. [1]

Ezért a fejlesztést új irányba kezdték meg, az autó rendszerei mind egy vezetékre, egy buszra csatlakoztak. Először 3 osztályba sorolták átviteli sebesség szerint az autókban fellelhető buszrendszereket, az A osztály 10kbit/s sebességre képes (ide tartozik a LIN protokoll is), a B osztály 100 kbit/s-ig terjed ki, és a C osztály, az 1 Mbit/s-ig levő tartomány. Későbbiekben az egyre jobban elterjedő multimédiás és valós idejű adatátvitelt igénylő rendszerek miatt jelentek meg a direkt ezekre a felhasználásokra készített protokollok, amik nagy adatátviteli sebességre (akár 50 Mbit/s-re) is képesek, illetve zavarvédettek, így biztosítva a biztonságos kommunikációt. Ezek a nagysebességű protokollok a drága kiépítési költség miatt csak bizonyos alkalmazásokhoz vannak felhasználva. [1]

2.1 Controlled Area Network

A Controlled Area Network (továbbiakban CAN) manapság az egyik legelterjedtebb buszrendszer, amit az autóiparban használnak. Felhasználható olyan területeken, amely megköveteli a valós idejű adatátvitelt, vagy olyan rendszerekben, ahol nagy mennyiségű adat küldésére van szükség. Természetesen ezek mellett az egyszerűbb alkalmazásoknál is használható, mint például a karosszéria és komfort elektronikák. [1]

A CAN Multi-Master elven működik, azaz egyik vezérlő sincs alárendelve a másinak. Egy CAN buszon maximum 32 eszköz lehet. Ezekhez szükséges a két jelvezeték, a CANH, CANL kivezetése, és az árnyékolás. Minden egységnél jelen kell lennie egy vezérlőnek, ami értelmezi a CAN üzeneteket, felügyeli a protokoll betartását, keresi a hibákat és ha szükséges közbe avatkozik, valamint egy illesztőre, aminek feladata, hogy a vezérlő által küldött üzeneteket kiültesse a fizikai buszra, vagy beolvassa onnan, illetve a busz feszültségszintjének fenttartása.



2. ábra. CAN busz felépítése.

A CAN buszon minden egység egyenrangú, ezért bármelyik képes üzenet küldésére. Ha viszont több egység szeretne egyszerre adatot kiküldeni, ezek összeütközhetnek, és az adat rongálódhat vagy akár el is vészhet. Ezért minden üzenet el van látva egy azonosítóval, amely megadja melyik üzenet élvez nagyobb prioritást. Az üzenet tartalma alapján történő priorizálási folyamatot nevezzük arbitrálnak. Az üzeneteknek nincs konkrét címzettjük, ezért a buszon valamennyi egység ugyanazt az üzenetet látja. Mindegyik maga dönti el az üzenet tartalma alapján, hogy szüksége van-e rá, ha igen fogadja és feldolgozza, ha nem, ignorálja. Az egységek a buszhoz CSMA/CD technológiával tudnak hozzáférni. A CSMA/CD (másnéven Carrier Sense Multiple Access with Collision Detection) teszi

lehetővé a rendezett adatátvitelt a buszon. Lényege, hogy minden résztvevő folyamatosan figyeli a busz aktivitását. Ha a buszon nincs üzenet, akkor hozzáférhetnek az egységek, azonban, ha több résztvevő egyszerre akar üzenetet küldeni, ütközés lép fel. Ezt az egységek érzékelik, és az arbitálás alapján eldől, melyik üzenet marad a buszon, a többi pedig elveszik. Arbitálás alatt a legnagyobb prioritású üzenet védve van, nincs bitvesztés. [4]

Indító bit	Arbitációs mező	Vezérlő mező	Adat mező	Hibaellenőrző mező	Jóváhagyó mező	Befejező bit
1 bit	12 - 32 bit	6 bit	0 - 64 bit	16 bit	2 bit	1 bit

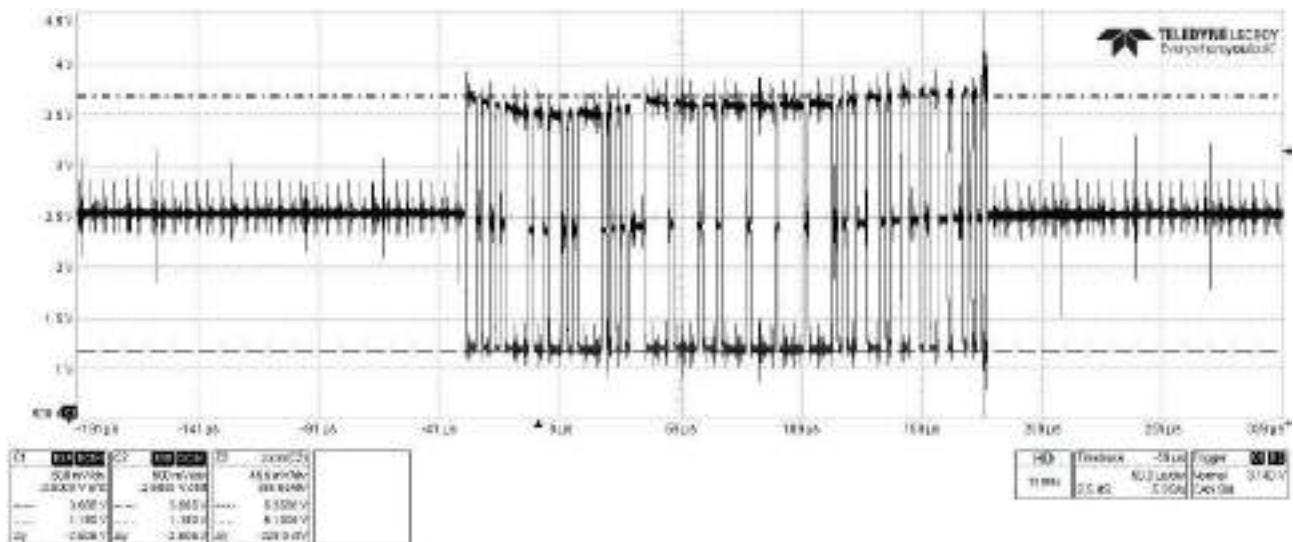
3. ábra. CAN adatkeret.

A CAN protokoll meghatározza az üzenetek formáját, és azt folyamatosan ellenőrzi. Minden adás egy üzenetekkel közti szünettel kezdődik, utána jön a start bit. Ezt követi az azonosító vagy másnéven arbitációs mező. Itt dől el, hogy az egyidejűleg elküldött üzenetek közül melyik rendelkezik több domináns bittel. A 3. ábrán látható, hogy az arbitációs mezőnek két fajtája van. Ennek oka az, hogy a CAN üzenet kereteinek két verziója van. Az „A” vagy standard keret teljes hossza 130 bit, ebből 11 bit az arbitációs mező, a „B” változatban a teljes hossz 150 bit és ebből 29 bit tartozik az arbitálás mezőjébe. A mező utolsó bitje, az adatkérelem bit határozza meg, hogy adattovábbítás vagy adatkérelem van. Ezt követi a vezérlő mező, amely az adatmező összegkódját tartalmazza. Ha az összegkód és az adatmező hossza nem egyezik, akkor a kommunikáció során hiba léphetett fel. Ezután jön maga az adatmező, ennek a hossza maximum 8 byte nagyságú lehet. A CRC mező ellenőrzi a sikeres adatátvitelt. Minden egység a kapott adat alapján képezi le az ellenőrző összeget. Ezt az adó ki is küldi, a vevőknek pedig össze kell ezt a saját eredményükkel hasonlítaniuk. Ha egyezés van, az adatátvitel sikeres volt azonban, ha van eltérés ezt, jelezni kell hiba üzenettel. Az ACK mező az adatátvitel nyugtázására szolgál. Az első bit szolgál jelzésül, az adó kiküldi ezt a domináns bitet, és ha minden rendben volt az adatátvitellel, a vevők recesszív bitre írják át. Ha valahol nem volt tökéletes az üzenet, azt az adó érzékeli, és újból próbálkozik az üzenet küldésével. Az egész keretet egy befejező jel zárja le, ez hét darab recesszív bitből áll. Ez alapján ismerik fel az állomások az adás végét. [4]

Minden egységen belül a vezérlő felel azért, hogy a kapott üzeneteket ellenőrizze, ha hibát észlel, azt jelezze, vagy javítsa ki, szerezzé meg a buszhoz a hozzáférést, küldjön üzeneteket. Manapság már ezek a CAN vezérlők leggyakrabban valamilyen mikrokontroller részei. Egyes vezérlők képesek a CAN „A” és „B” változatát is kezelni. Az üzeneteket azonban ki kell küldeni és

beolvasni a buszról, erre önmagában a vezérlő nem képes, ezért van szükség egy Transceiver-re. Feladata, hogy a bejövő bitsorozatot feszültségszintekké alakítsa, vagy fordítva, a feszültségszintek változását alakítsa vissza bináris kóddá. A Transceiver-ben két tranzisztort kapcsolgat a vezérlő. Ezeknek a gyors egyidejű ki- és bekapcsolásával történik a jelek kiültetése a buszra. a CANH vezetékre a testhez viszonyítva nagyobb, míg a CANL vezetékre kisebb feszültség kerül. [4]

Fontos, hogy az egységek szinkronizálva legyenek egymással, ugyanis, ha nem, az időbeli elcsúszás miatt hamis üzeneteket kaphatnak. Ha a busz inaktív, magas szinten van, viszont, ha elkezd valamelyik egység adni, akkor alacsony szintre váltja. Ekkor kezd az összes résztvevő figyelni, és szinkronizálni magukat az adóhoz. Az üzenetben található felfutó és lefutó élek szolgálnak szinkronizálási pontoknak, ezért, ha sok egymást követő 0 vagy 1 bit van, az szinkronvesztést okozhat. [1]



4. ábra. Oszilloszkóppal elkaptott üzenet a SZEmission CAN hálózaton.

A buszon létrejött feszültségszintek eltérnek a különböző CAN fajtáknál. Alacsony sebességű CAN-nél (CAN A) alaphelyzetben a CANH 0V-on míg, a CANL 5V-on van. Domináns állapotban ez megfordul, a CANH 3,6-4V körül van, míg a CANL 1-1,4V körül. A Transceiver-ek a két vonal közti feszültségkülönbséget dolgozzák fel. Ha a két vonal közötti feszültségkülönbség 2,2-3V között van, az 0 bitnek felel meg, míg, ha a különbség 5V, az 1 bitnek. Ennél a CAN típusnál nem szükséges a buszvezetékek végeit lezárni impedanciával. [4]

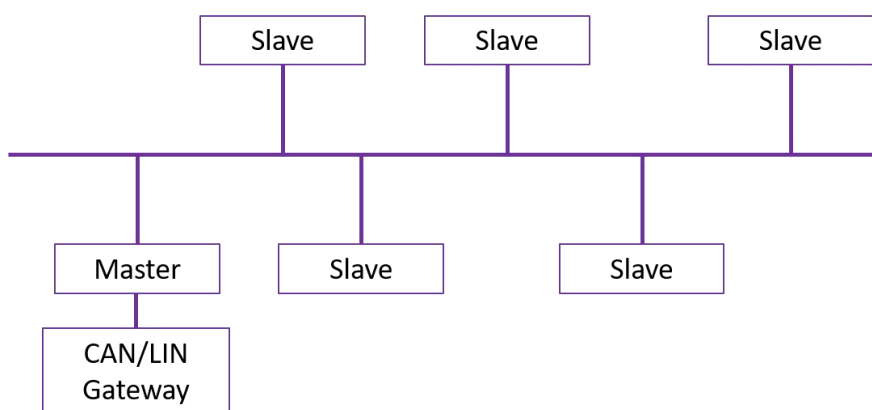
A nagysebességű CAN B hálózatoknál alaphelyzetben a busz 2,5V szinten van, ezt nevezzük a busz recesszív állapotának, ilyenkor a feszültség különbség 0V. A busz domináns állapotában a CANH 1V-al növekszik, míg a CANL csökken, így egy 2V-os különbség jön létre. Itt szükség van

120 ohmos lezáró impedanciák alkalmazására, így a továbbított jel nem verődik vissza a vezeték végéről, nincs reflexió, hanem elnyelődik. [4]

2.2 Local Interconnect Network

Az autóiparban a másik leggyakrabban használt alkalmazás a CAN mellett a Local Interconnect Network (továbbá: LIN) protokoll. Elterjedése annak köszönhető, hogy rengeteg olyan elektronika található a gépjárművekben, ahol nincs szükség valós idejű adatátvitelre és sok jelvezeték, ezért egy olcsóbb hálózat kiépítése is elegendő. A LIN hálózat egy egyvezetékes alacsony adatátviteli sebességű buszrendszer, melynek egyszerűsége az UART-ra épülésből adódik. Gyakori felhasználású a karosszéria és a komfort elektronikában. [2]

A LIN Master-Slave alapon működik, tehát a buszon egy master található, és maximum 15 slave. A LIN általában egy alrendszer egy nagyobb szintű hálózatban, a kettő közötti adatátvitelt a master-nek vagy egy Gateway-nek kell biztosítania. Egy járművön belül akár több LIN hálózat is lehet, amiket egy CAN hálózat fog össze, minden különálló LIN rendszerhez saját master tartozik. A buszvezeték hossza legfeljebb 40 méter lehet. Minden slave-hez tehát három vezeték kell kihúzni, egy táp, egy föld, és egy adat vezeték, jellemzően piros, fekete és ibolyaszínben. Egy Slave lehet akár egy önálló vezérlőegység, akár egy intelligens szenzor, amely képes a buszra adatokat küldeni. Ezeket a szenzorokat el kell látni egy mikroprocesszorral és egy buszillesztő eszközzel. [2]



5. ábra. LIN busz felépítése.

Ellentétben a CAN rendszerekkel, LIN hálózatokon csak a Master kezdeményezhet kommunikációt. A Master dönti el, hogy kinek címezi az üzenetet, és hogy utasítást ad, vagy adatot

kér be. Ha adat lekérdezés van, akkor azonosítás után a címzett válaszol a megfelelő adattal. A Slave-ek önmagukban sosem írnak adatot a buszra, csak felszólításra. Lehetőség van két Slave közötti adatcserére is, viszont ezt is csak a Master utasítására. Ha a buszra kikerül egy üzenet, azt minden eszköz látja, viszont az azonosító mező dönti el, hogy melyik eszköznek kell azt feldolgoznia. [1]

A Master másik fontos funkciója, hogy a LIN hálózaton minden egységet képes alvó üzemmódba helyezni. Ekkor az áramfogyasztás minimális, a Master várakozik. A buszon bármely eszköz felébresztheti a hálózatot. [2]

A LIN protokollja szabja meg, milyen formában történik az adatok küldése. Egy LIN üzenet tartalmaz egy fejléct és egy válasz mezőt. Üzenet küldésekor először minden egységet szinkronizálni kell, ezt egy 13 bites Sync Break és egy 8 bites Sync mező hajtja végre. Sync break alatt csak recesszív, azaz 0 bitek vannak kiküldve, majd a Sync mezőben a 0 és 1 bitek váltják egymást. Ha a szinkronizálás megtörtént, a Master kiküldi az azonosító mezőt, ami tartalmazza a címzettet. Ezt követi az adatmező, amit vagy Master, vagy a Slave küld. Az adat hossza legfeljebb 8 byte lehet. Bezárólag egy ellenőrző összeggel ér véget az üzenet. [2]

Sync Break	Sync	Azonosító	Adatmező	Ellenőrző összeg
------------	------	-----------	----------	------------------

6. ábra. LIN üzenetének formátuma.

A LIN buszhoz hasonló eszközökkel lehet hozzáférni, mint a CAN buszhoz. Itt is szükség van egy Vezérlőre és egy Transceiver-re, egy illesztőre. Utóbbi feladata, hogy az üzenetet a fizikai buszra illessze, a 0 és 1-es bitek a megfelelő feszültséget elérjék. LIN buszon a domináns 0 bitek a 0V szinthez közelítenek, de felső határuk a tápfeszültség 40%-a. A recesszív 1-es bitek felső határa a tápfeszültség, alsó pedig annak 60%-a. Ezeknek a nagy tartományoknak köszönhetően a LIN ellenálló a zavarokkal szemben. [2]

2.3 Flexray

A Flexray rendszer magasabb szintű megoldást nyújt a CAN rendszerek egyes hiányosságaira. Fő alkalmazási területe az X-by Wire rendszerek, lényege, hogy a korábban használt mechanikus összeköttetéseket váltsa fel elektronikus és elektromechanikus összeköttetésekre. Emiatt a Flexray-nek biztosítani kell nagy adatátviteli sebességet, a hibatűrő és rugalmas kommunikációt. [1]

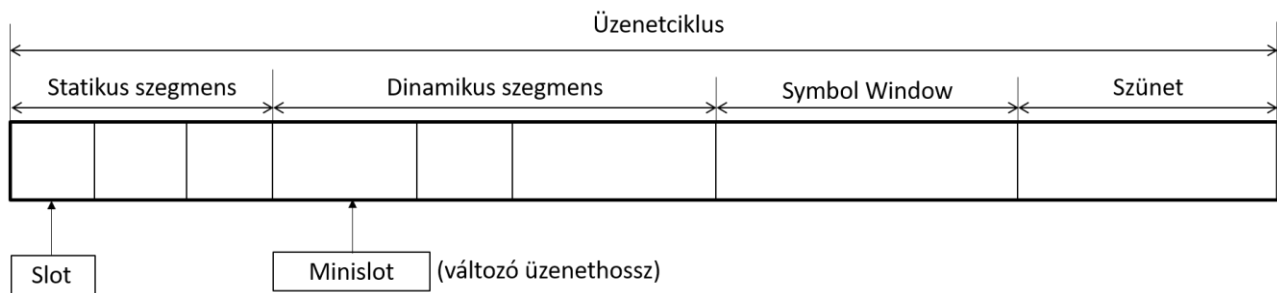
A Flexray adatbusznak többféle topológiája lehet. A legegyszerűbb a vonalas topológia, viszont gyakrabban alkalmazott a csillag topológia. A csillag kapcsolat lehet passzív és aktív. Passzívnál a csillagpont nem végez semmilyen feladatot, míg aktívnál figyeli a buszon az adatfolyamot, az egységek állapotát, és egy esetleges meghibásodásnál képes az elromlott egységet leválasztani a buszról. Emellett a beérkező jeleket felerősíti, így a buszon minden egység megkapja az üzenetet. Az egységek a csillagponttól maximum 24 méter távolságra lehetnek. Aktív csillagpontokat lehet egymás után sorba kötni, így még biztonságosabbá téve a kommunikációt.

A Flexray kommunikáció két vezetéken történik, melynek fő oka a biztonságos és hibamentes adatátvitel. Emellett a két csatornán van lehetőség külön üzeneteket is küldeni. Minden csatornához külön csillagpontok tartoznak. A két vezeték árnyékolása opció, de nem szükséges, csavartérpárban készülnek. A busz végeit aktív csillag topológiánál szükséges mindenhol lezárni ellenállásokkal, passzívnál elég csak a két legtávolabbi pontnál ezt megtenni. A buszon jelenleg a legnagyobb átviteli sebesség 10 Mbit/s, amit aktív csillag topológiával lehet elérni. [1]

Ahhoz, hogy a buszra egy egység rátudjon csatlakozni a következőkre van szükség: egy mikroprocesszorra, egy vezérlőre, két busz illesztőre (Transceiver-re), és két buszvédelemre. A vezérlő és az illesztő feladata nagyon hasonló, mint a korábban tárgyalt CAN és LIN rendszereknél. Az új eszköz a buszvédelem, melynek feladata, hogy a vezérlő csak akkor küldhesse az adatokat ki a buszra, amikor az ő időintervalluma van. A mikroprocesszor közli a védelemmel mely intervallumok tartoznak az hozzá. Mivel csak bizonyos időközönként tudnak az egységek a buszhoz hozzáférni, így az adatütközés minimális. Az adatok fogadása a buszról mindig lehetséges. [1]

A Transceiver-ek állítják elő a megfelelő jelszinteket. Hasonlóan a CAN hálózatokhoz, nyugalmi állapotban a busz 2,5 V-on van, viszont itt mind a 0 és 1 biteket is azonos nagyságú feszültségkülönbség mutatja, tipikusan 750 mV. [1]

A Flexray a TDMA buszhozzáférési technológiát alkalmazza, ami azt jelenti, hogy minden egyes egységnek egy megszabott időpontja van az üzenet küldésére. Minden egységhez saját időablak tartozik, ezeknek van egy meghatározott sorrendje és ciklikusan ismétlődnek. Egy időablak a következőkből épül fel: statikus mező, dinamikus mező, Symbol Window és az üzenetek közti szünet. A statikus szegmensben az adatátvitel szinkron módon történik, ezért fontos, hogy minden egységnek ismernie kell a rendszer globális óra idejét, és a sajátját ehhez kell korrigálnia. Szinte minden egység a hálózaton rendelkezik saját órajellel, viszont ezek az idők eltérnek egymástól kis mértékben. A folyamatos szinkronizáció érdekében minden Flexray hálózaton kell lennie legalább két szinkronizáló egységnek, amik összehangolnak minden eszközt a hálózaton Sync üzenetek küldésével. [1]

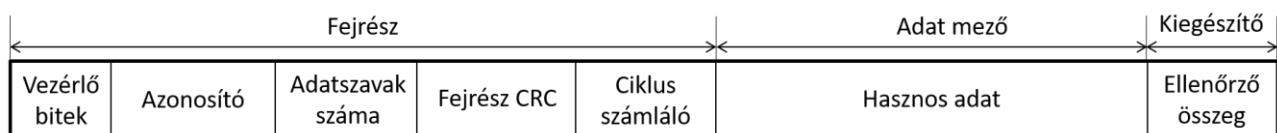


7. ábra. Egy Flexray üzenetciklus felosztása.

A statikus mezőben meghatározott számú slot van, egy slot hossza pont akkora, hogy egy Flexray üzenet elférjen benne. A slot-ok mindkét csatornán szinkronban vannak, minden egységnek van egy sajátja legalább az egyik csatornán. Azok az eszközök, amik mindkét csatornához hozzá férnek használhatják a másodikat redundásként, de akár külön üzenetet is küldhetnek rajta. A fel nem használt slot-ok mind elvesznek. A statikus mezőben mindig csak egy bizonyos mennyiségű adatot lehet kiküldeni, viszont azt hibátlanul teszi valós időben. Ha az egységek szeretnék még további üzeneteket küldeni, azt a dinamikus mezőben tudják megtenni. Ennek a szegmensnek legalább 2 legfeljebb 1023 időablakja lehet. [1]

Hasonlóan a statikus részhez, a dinamikus is időablakokból áll, viszont sokkal rövidebbek ezért minislot a nevük. Az ablakok itt viszont aszinkron módon követik egymást, ezért a hosszuk rugalmas. Minden vezérlőegység, aki akar üzenetet adni, kap adási jogot, és hozzá tud férni a két csatornához. A két csatornán külön üzeneteket adhatnak. A statikus és dinamikus mezők hossza együtt maximum 2047 időablak lehet. A Symbol Window vezérlési információk továbbítására szolgál, nincs információ csere. Ezt követi az üzenetek közti szünet. Itt történik meg a helyi és a globális órajel szinkronizálása. [1]

Az időablakokban küldhetnek az egységek üzenetet. Természetesen a Flexray protokoll megadja ezeknek a formáját.



8. ábra. Flexray üzenet kerete.

Az üzenet a 5 byte hosszú fejrésszel kezdődik. Az elején 5 előre meghatározott vezérlő bit van, ezt

követi a slot azonosítója, majd az adat mező hossza. Ez a hossz a statikus szegmensben minden esetben ugyanakkora. Ezt követi a header ellenőrző összege és a jelenlegi ciklus sorszáma. Az adatmező maximum 254 byte hosszú lehet, statikus üzenetekenél ez a hossz 12 byte. [1]

A Flexray lehetőséget biztosít a buszon levő egységek számára, hogy készenléti vagy alvó üzemmódba váltsanak. Készenléti állapotban a vezérlő kap tápellátást, viszont nem vesz részt a kommunikációban ezért áramfelvétele minimális. Alvó módban a vezérlő teljesen lekapcsol, egyedül a Transceiver figyeli a buszt wake-up üzeneteit. [1]

2.4 Media Oriented System Transport

A Media Oriented System Transport, rövidebb nevén a MOST rendszereket főleg multimédiás alkalmazásokra használják gépjárművekben. Nagy átviteli sebességét köszönheti az optikai szálak használatának, a legújabb verzió akár 50Mbit/s sebességre is képes. A Plug & Play támogatásának köszönhetően a legtöbb multimédiás eszköz MOST hálózaton kommunikál, legyen az navigáció, CD-s rádió, TV, DVD lejátszó vagy tolatókamera. [1]

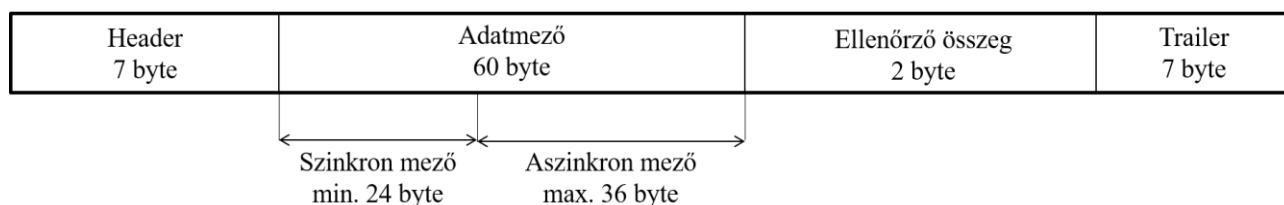
A MOST rendszerek hasonlóan a LIN hálózathoz Master-Slave alapon működnek. A master felel a busz felügyeletéért, tápellátásáért, az órajel adásáért, amihez a slave-ek szinkronizálni tudják magukat, és gateway-ként is funkcionálnia kell. A buszon az egységek gyűrű topológiában vannak összekötve, ezért a kommunikáció csak egy irányba halad. [1]

Az átviteli közeg leggyakrabban optikai szál, de lehet csavartérpár is. Az optikai közegen a jelet fényimpulzusokkal lehet átvinni. Ezt a legtöbbször egy LED állítja elő adó oldalon, vevőnél meg egy fotódióda fogadja. A kibocsátott fény 650 nm hullámhosszúságú, a látható tartományban a vörös színnek felel meg. Az optikai kábel a következő rétegekből áll össze: a mag, ez vezeti a fényt, a mag körüli köpeny, ami a vezetett fény visszaverődését biztosítja, egy fényelnyelő köpeny, ami megakadályozza a külső fénysugaraktól a magot, és egy szigetelő réteg, mely megvédi a kábelt a fizikai sérülésektől. Ezek a kábelek kevésbé hajlékonyak, mint egy közönséges réz vezeték, mivel a magban a fény túlzott elgörbülés esetén már nem verődik teljesen vissza a köpenyről, veszteséges a kommunikáció. Extrém esetben a köpeny megsérül, ilyenkor a fénysugarak kijutnak a kábelből. Az optikai kábelek használatakor körültekintően kell megtervezni a vezetékezés útját, hogy minél kevesebb törés legyen benne. [1]

A buszra való csatlakozáshoz a következőknek kell meglenniük: szükség van egy optikai csatlakozóra, ami két pólussal kell, hogy rendelkezzen. Az egyik pólus egy LED-el van ellátva, ezért ezen csak adat küldése lehetséges, a másik póluson egy fotódióda vagy fotótranszisztor helyezkedik

el, ezért ezen az ágon csak fogadni lehet üzeneteket. A MOST-Transceiver feladata, hogy ezeknek az elemeknek a megfelelő jelszinteket elő állítsa. A MOST vezérlők hasonlítanak a CAN rendszerekben használtakhoz, azonban ezek még nem integrálódtak be más mikroprocesszorokba, jellemzően külön egységként működnek. [1]

A buszhoz való hozzáféréshez az egységek a TDMA technológiát alkalmazzák, a Flexray hálózatokban is ez használatos. A gyűrűn minden egységnek adott ideje van üzenetet adni, aztán tovább adja a jogosultságot a következő résztvevőnek. A MOST szinkron kommunikáció, de vannak aszinkron elemei is. Az üzeneteknek egy meghatározott kerete van, amit el tudnak küldeni az időablakban. A MOST régi verziójánál (MOST25) ezeknek az üzeneteknek a maximális hossza 64 byte, míg a második generációnál (MOST50) a duplája, 128 byte. Az üzenet keretben vannak szinkron és aszinkron mezők is. Aszinkron adatátvitelnél a szabad keret nagysága korlátozott, ezért itt az arbitrálás dönti el melyik üzenet kerül ki a buszra. A MOST adatkerete az 9. ábrán látható.



9. ábra. MOST25 adatkeret.

Az adatmezőben a szinkron adatok legfőbbképpen kép és hanganyagok, míg az aszinkron adatok főleg információk. A szinkron keretnek a minimális hossza 24 byte, azonban ez lehet több is. Ha nincs szükség nagyobb keretre, a maradék részét az adatmezőnek aszinkron tölti ki. [1]

A MOST vezérlők hasonlóan a LIN és FlexRay vezérlőkhöz, három üzemiállapottal rendelkeznek. Normál módban kommunikálnak az eszközök egymással, készenlétkben figyelik a buszt, de nem vesznek részt a kommunikációban, alvó módban lekapcsolnak és csak ébresztésre fognak újra indulni. [1]

2.5 Összehasonlítás

1. táblázat. Autóipari buszrendszerek összehasonlítása.

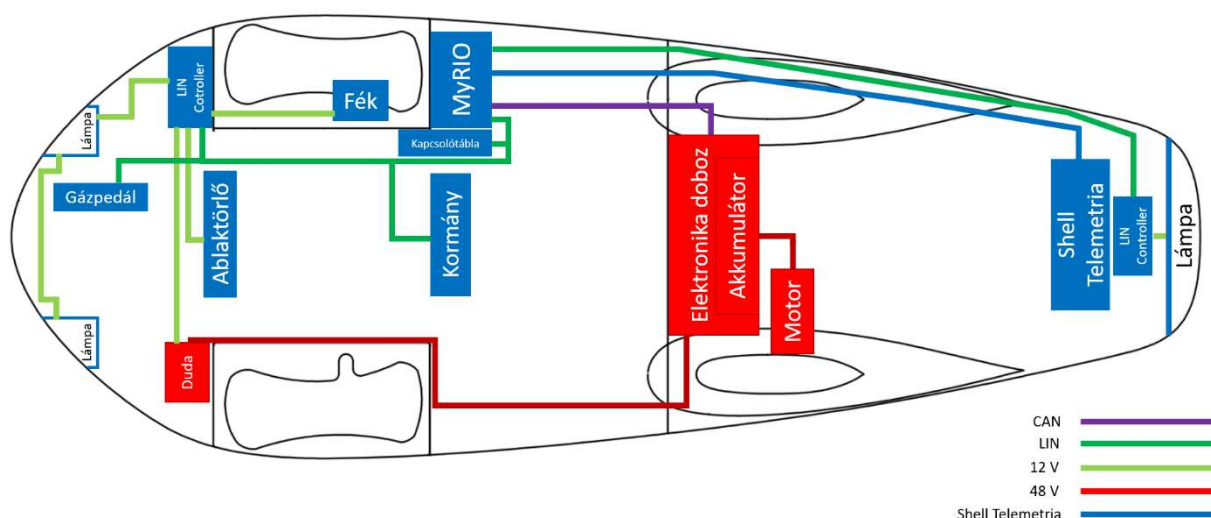
Megnevezés	Local Interconnect Network	Controlled Area Network	Flexray	Media Oriented System Transport
Hierarchia	Single Master	Multi Master	-	Single/Multi Master
Topológia	Busz	Vegyes	Csillag	Gyűrű
Átviteli sebesség	20 kbit/s	1 Mbit/s	10 Mbit/s	50 Mbit/s
Buszhozzáférés	Master vezérelt	CSMA/CD	TDMA	TDMA
Felhasználás	Karosszéria és komfort elektronika	Motorvezérlés, diagnosztika	X by Wire rendszerek	Multimédiás rendszerek
Átviteli közeg	Egy jelvezeték	Két jelvezeték	Két külön csatorna	Optikai kábel
Előnye	Olcsó kiépítés	Leggyakoribb rendszer	Biztonságos adatátvitel	Nagy sávszélességű kommunikáció

3. Áramkör tervezése

3.1 Rendszerterv

Az új autónak egy teljesen új rendszertervet kellett elkészítenünk, amely megfelel az új követelményeknek, illetve a régi rendszer hibáit is kiküszöböli.

A korábbi autóban többször is kommunikációs problémák adódtak a vezetékezésből, ezért a teljesítmény és kommunikációs vonalakat a lehető legtávolabbra helyeztük el egymástól. Ennek köszönhetően a kábelezés is sokkal egyszerűbb és átláthatóbb lett. Az autó főbb elektronikai egységei a következők: akkumulátor doboz, elektronika doboz, myRIO, és a perifériák.



10. ábra. SZEmission elektronikai rendszerterve.

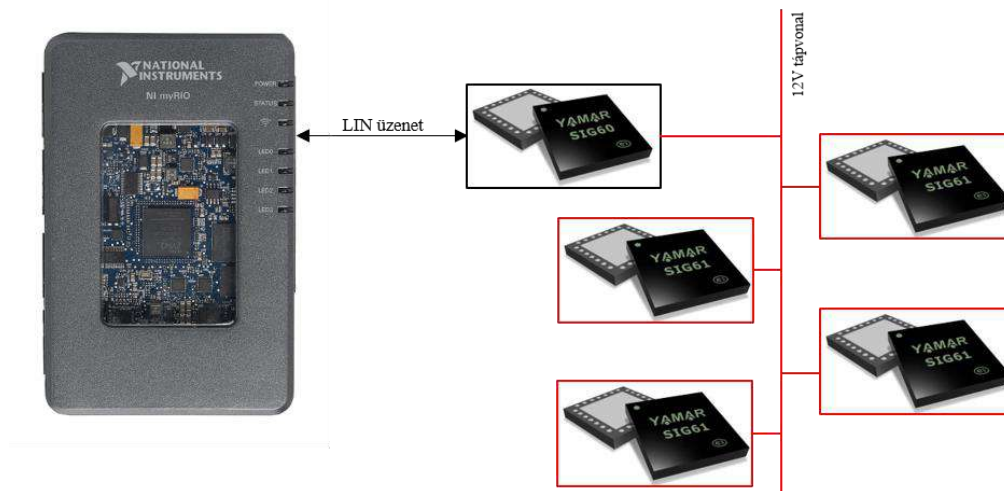
Az akkumulátor dobozban található az autó tápellátásáért felelős két darab 6 cellás akkumulátor, a Battery Management System (továbbiakban BMS) és a központi relé. A BMS feladata, hogy folyamatosan figyelje az akkumulátorok állapotát, és ha kell közbeavatkozzon. A felügyelt paraméterek a következők: a cellák feszültsége, a cellák közötti maximum különbség, a leadott áram, a cellák hőmérséklete, és maga a BMS hőmérséklete. Ha bármelyik paraméter átlépi a küszöbszintet, a relét kikapcsolja a BMS és az autó rendszere leáll. Minden információt kiküld a BMS a CAN hálózatra. Emellett az autó biztonsági köre, a fő indító és a vészkapcsolók is itt kapcsolják a relét le.

Az elektronika dobozban található a motorvezérlő és a hozzá tartozó árammérők. A motorvezérlő feladata, hogy előállítsa a PMSM szinkron motornak a 3 fázisú feszültséget, vezérelje a forgásirányt és a fordulatszámot. Emellett folyamatosan figyeli a motor állapotát, és az információt ki is küldi a CAN hálózatra. A motorvezérlő mellett található egy DCDC konverter, amely az autó rendszer feszültségét 48 V-ot konvertál 12 V-ra. Ezen a 12 V-os hálózaton találhatóak az autó további eszközei.

A myRIO felel az autó teljes működéséért. Ezen a kis számítógépen fut az a rendszer, amely felelős a CAN, LIN hálózat, és a telemetria rendszer működéséért. A CAN hálózatra egy kiegészítő kártya segítségével tud csatlakozni, itt a motorvezérlőtől jött adatokat dolgozza fel, illetve küldi vissza a vezérlőnek szükséges információkat, mint például: gázkar állás, menetirány és engedélyezés.

A LIN hálózaton található az autó legtöbb perifériája. Önmagában a myRIO nem képes a LIN buszon a master szerepét felvenni, a hálózatra az UART portján keresztül tud csatlakozni egy vezérlő

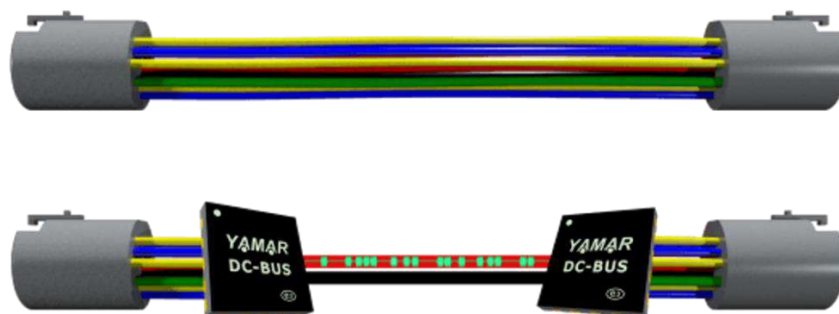
és Transceiver beiktatásával. A végpontoknál már szükséges egy slave megléte is, hogy minden eszköz rá tudjon csatlakozni a LIN hálózatra. Az autó egyes részein nem csak jel szükséges ezeknek a működtetéséhez, hanem teljesítmény is, ezért minden egységhez szükséges lenne 3 ér kivezetése, a táp, a föld és az adat. Ez azonban ismét bonyolult és átláthatatlan kábelezéshez vezetne, ehelyett találtunk egy alternatív megoldást. A LIN adat vezetéket kiváltja két, direkt erre a célra kifejlesztett vezérlő. Két ilyen egyedi áramkör kerül a LIN hálózatra, a master, ami irányítja a teljes kommunikációt, és a slave, ami a master utasításait hajta végre. Saját feladatomnak a slave áramkörének megtervezését és elkészítését kaptam.



11. ábra. LIN kommunikáció a SZEmission-ben.

3.2 A kiválasztott vezérlők.

A YAMAR cég fő profilja a tápvonalon történő kommunikálás biztosítása, legyen az CAN, LIN, SPI vagy UART kommunikáció.



12. ábra. Tápvonalon történő kommunikáció. [A1]

Az egyedi IC-nek köszönhetően le lehet csökkenteni a kábelezések számát, emellett növelik a biztonságos üzenetváltást a különböző kommunikációs egységek között. A cég jelenleg két termékcsaládot gyárt, a DCB család nagysebességű adatátvitelt biztosít, maximálisan 1,4 Mbit/s, így ez alkalmazható CAN, I2C SPI hálózatokban. A másik a SIG család, ezt alkalmazzák olyan rendszerekben, ahol nincs szükség nagy adatátviteli sebességre, a maximum 115 Kbit/s, ezért alkalmas az autóban levő LIN hálózatba.

A tápvonalon való kommunikálásnak több veszélyforrása is van. A tápforrás és a környezetből adódó zajok, zavarok miatt bonyolult jelzésrendszerre és ellenőrzött adatkeretre van szükség. Az üzenetek előre meghatározott frekvenciákon vannak kiküldve, emellett az adatkeret eleje és vége segít elkülöníteni a hasznos adatot a zajoktól. Az eszközök a hálózaton mindig azon a frekvencián figyelik az üzenetet, amire ki lett küldve, és miután felismerték a jel mintákat, kiolvassák az adatot belőlük. Egy tápvonalon akár több SIG hálózat is lehet, ezeket a vivőfrekvenciák választják el egymástól. Fontos, hogy egy hálózaton belül minden egység a megfelelő frekvenciára legyen beállítva. Minden hálózathoz két frekvencia tartozik, az eszközök képesek a kettő közt váltani, ha zavar vagy adat ütközés történik. Egy SIG hálózaton összesen 16 eszköz lehet egyszerre, ebből egynek kell a masternek lennie. Minden eszköz rendelkezik saját azonosítóval, ezért minden üzenet tartalmaz egy azonosító mezőt is.

A SIG család használatával lokálisan lehet a perifériákat vezérelni, a busz huzalozásánál pedig egyszerűbb és átláthatóbb vezetékeztést érhetünk el.

3.2.1 SIG60



13. ábra. SIG60-as IC

A SIG60 a master, feladata, hogy a myRIO és a tápvonalon levő eszközök között az adatáramlást fenntartsa. A tápvonalra ő ülteti ki az üzeneteket, illetve folyamatosan figyeli a zavarokat, ütközéseket. A myRIO-val egy soros porton keresztül küldi és fogadja az adatokat.

A myRIO az adatküldés mellett a SIG60 működését is tudja irányítani. Ahhoz, hogy az IC működési paramétereit módosítani lehessen, „Command” módba kell rakni a HDC pin lehúzásával. Ilyenkor a SIG60 nem küld adatot a tápvonalra, és a két belső regiszterét változtatni lehet. A regiszterekben minden bithez tartozik egy funkció, amit az 1. táblázat mutat.

2. táblázat. SIG60 regisztereinek funkciója.

Regiszter 0	Bit 7	Bit 6	Bit 5	Bit 4
Alapértelmezett	0	0	0	1
	Zavar észlelve	Visszacsatolás kikapcsolása	Ébresztő üzenet kikapcsolása	Automatikus alvó üzemmód
Regiszter 0	Bit 3	Bit 2	Bit 1	Bit 0
Alapértelmezett	1	0	0	1
	Ébresztő üzenet időtartama	Frekvenciaváltás zavar esetén	Visszacsatolás a kimenetről	Alapértelmezett frekvencia választás
Regiszter 1	Bit 7-6	Bit 5-4	Bit 3-0	
Alapértelmezett	1	1	1	
	Előre lefoglalt	Átviteli sebesség	Frekvenciapár választás	

Az IC több üzemmóddal is rendelkezik. Alap esetben a normál mód aktív, ilyenkor képes az üzenetek fogadására és küldésére a SIG60. Alvó üzemmódban az IC nagy része le van kapcsolva, csak a belső alacsony órajel működik. Ebben a módban az áramfelvétel minimális. A busz aktivitása folyamatosan figyelve van, ha nincs aktivitás, folytatódik az alvó üzemmód. Készenléti üzemmódban az IC már észlelte, hogy van aktivitás a buszon, ezt jelezte is a myRIO felé, de még nem képes a kommunikációra. Csak akkor tud készenlétből normál módba váltani, ha azt a myRIO engedélyezte.

3.2.2 SIG61



14. ábra. SIG61-es IC.

A SIG61 feladata, hogy a SIG60 utasításai alapján írja vagy olvassa a nyolc darab ki- és bemeneteit. A SIG61 hasonlóan figyel a tápvonalat, és szűri ki az üzeneteket róla. Ha az üzenetet sikeresen hiba nélkül fogadta, és a címzett azonosító egyezik a sajátjával, akkor feldolgozza az utasítást és végrehajtja.

Hasonlóan a SIG60-as IC-hez a SIG61 paramétereit is lehet változtatni. Viszont itt nem a belső regisztert kell átírni, hanem itt minden módosítható paraméternek megfelelő számú láb van kivezetve. Ezeken a lábakon a magas és alacsony szintekkel lehet beállítani mindent. Ha valamit változtatunk, akkor az IC a következő újra indítástól dolgozik az új beállítással. A kommunikáció frekvencia párját a 2. táblázat alapján lehet beállítani, ahol az F0F1-0 - F0F1-3 lábak a frekvenciapárokat 4 bittel kifejezve, F0 és F1 pedig a kiválasztott frekvenciát jelölik.

3. táblázat SIG61 frekvenciapárjának beállítása.

F0F1-(0-3)	F0	F1
1111	1,75 MHz	4,5 MHz
1110	1,75 MHz	5,5 MHz
1101	1,75 MHz	6 MHz
1100	1,75 MHz	6,5 MHz
1011	4,5 MHz	5,5 MHz
1010	4,5 MHz	6 MHz

1001	4,5 MHz	6,5 MHz
1000	4,5 MHz	10,5 MHz
0111	10,5 MHz	13 MHz
0110	5,5 MHz	10,5 MHz
0101	5,5 MHz	13 MHz
0100	6 MHz	10,5 MHz
0011	6 MHz	13 MHz
0010	6,5 MHz	10,5 MHz
0001	6.5 MHz	13 MHz
0000	5,5 MHz	6,5 MHz

Fontos megjegyezni, hogy e paramétereknek a változtatása csak akkor lép életbe, ha a SIG61 Slave módban van. Az IC képes Masterként is funkcionálni, erről később lesz szó. Az adatátviteli sebességet hasonlóan lehet beállítani a BitRate0-1 lábakkal, ezt a 3. táblázat mutatja.

4. táblázat. SIG61 adatátviteli sebességének beállítása.

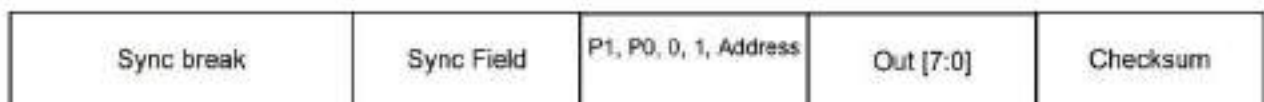
BitRate (0-1)	11	10	01	00
Frekvencia				
1,75 MHz	-	-	9,6K	19,2K
4,5 MHz	38,4K	57,6K	9,6K	19,2K
5,5 MHz	38,4K	57,6K	9,6K	19,2K
6 MHz	38,4K	57,6K	9,6K	19,2K
6,5 MHz	38,4K	57,6K	9,6K	19,2K
10,5 MHz	38,4K	57,6K	115,2K	19,2K
13 MHz	38,4K	57,6K	115,2K	19,2K

Ezeknek a paramétereknek meg kell egyezniük egy hálózaton, ahhoz, hogy minden eszköz képes legyen a kommunikációra. Természetesen minden más is megtalálható a SIG61-en mint, ami a SIG60-on, pl.: interferenciaérzékelés, automatikus frekvenciaváltás, inaktivitás esetén alvó üzemmód.

Minden SIG hálózaton szükség van egy masterre, aki irányítja a kommunikációt, általános esetben ezt a szerepet egy SIG60 fogja betölteni, azonban a SIG61 is képes masterként működni. A Mode3 és Mode4 pinek döntenek el, hogy jelenleg milyen módban van az IC. Ha magasra vanna húzva,

akkor SIG61 módban van, Slaveként viselkedik a hálózaton, és 8 ki- és bemenete van. Ha a két mód pin alacsony szintre van húzva, akkor az IC Masterként működik. Ilyenkor a HDO pin aktív lesz, a kettes és hármas bemenete ugyanúgy viselkedik, mint a SIG60-on a HDI és HDC pin. Ez a háromláb csatlakozna a myRIO-hoz.

A SIG család protokollja hasonló a LIN protokolléhoz, a buszon öt különböző utasítást különböztetünk meg. Az első ilyen a Write, a Master a címzett SIG61 kimeneti pinjeinek az állapotát állítja be. Az üzenet formátuma a 15. ábrán látható.



15. ábra. SIG61 Write utasítás.

Látható, hogy itt is egy egy byte-os Sync break mezőt követ egy szintén egy byte-os szinkronizáló mező, melyeknek feladata, hogy minden eszköz a hálózaton figyelje a buszt. Az azonosító mező első 4 bite előre meghatározott, ezt követi maga a címzés. Az adat mező tükrözi a kimenetek állapotát, amit a SIG61-nek be kell állítani, végül ezt követi az ellenőrző összeg.

Read utasításnál a Master megszólítja azt a Slave-et, aminek az állapotát le akarja kérdezni, majd a Slave válaszol, hogy jelenleg a bemenetein milyen jelek vannak. Az üzenet fejlécét itt is a Master küldi, viszont az adatmezőt és az ellenőrző összeget már a címzett Slave küldi vissza.

A Read-Change utasítás ciklikusan van kiküldve minden Slave-nek. Ez az üzenet ugyanúgy a SIG61 bemeneteit kérdezi le, viszont nem jelenlegi státuszt, hanem a ciklus alatt történt változásokat küldi ilyenkor el a Slave.

Sleep utasítás észlelésekor minden Slave a hálózaton alvó módba kapcsol. Ilyenkor az áramfelvétel elenyésző. A SIG61-ek kimenetei mindig a legutóbbi állapotot tükrözi alvó módban, azonban, ha nem a Master által alszik el az IC, hanem az AutoSleep funkciója miatt, minden kimenetét lekapcsolja. Alvó módban az IC-k minden 32 ms után ellenőrzik van-e a buszon aktivitás, ha van az IC normál módba áll vissza, ha nincs továbbra is alvó módban maradnak. A busz „felébresztését” bármelyik eszköz kezdeményezheti.

Az utolsó utasítás a Change Frequency, ami nevéből adódóan felszólítja az összes eszközt, hogy váltson át a másik előre beállított frekvenciájára. A frekvenciákat a korábban említett F0F1 (0-3) lábakkal lehet beállítani, az alapértelmezést pedig az F1nF0 lábbal.

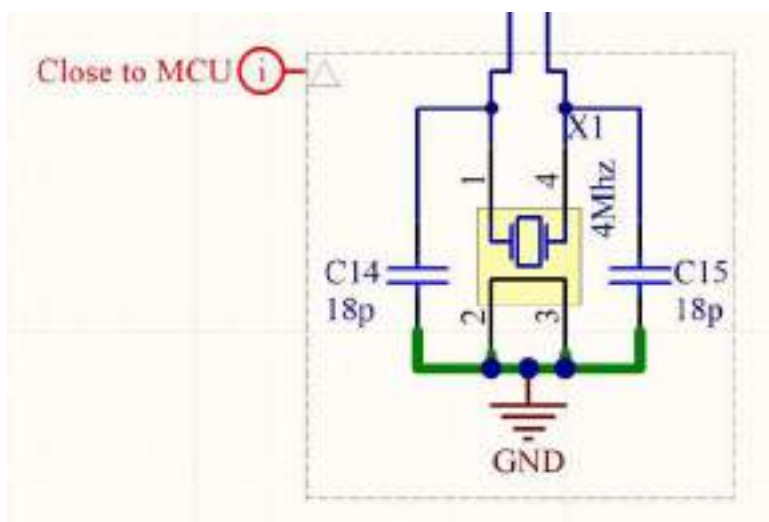
minden lábnak megvan a típusa, illetve a neve. A lábakat a gyártói leírás alapján neveztem el, illetve kiválasztottam, hogy ki vagy bementek vagy tápellátást biztosítanak. Elhelyezésnél törekedtem a valóságos IC lemásolására, így a kapcsolási rajzon is könnyebb eligazodni. A kész szimbólum a 16. ábrán látható.

A következő lépés az IC megfelelő beállítása, és a környezetének felépítése volt. A korábban kifejtett paramétereknek mind a SIG60-on mind itt meg kell egyeznie a helyes működés érdekében. A frekvenciapár választásánál figyelembe vettem a jelenleg készleten elérhető kerámia szűrőket, ezért az 5,5 MHz és 6,5 MHz lett beállítva az F0F1 0-3 lábak földre húzásával. Ebből a frekvenciapárból az F1 azaz 6,5 MHz lett alapértelmezetten beállítva. Az InterHop láb szintén földre húztam, ugyanis az autóban nincs szükség több SIG hálózatra, ezért nem tud ütközés fellépni. A frekvenciák váltására viszont meghagytam a lehetőséget, az MF1nF0 lábat földre és tápfeszültségre is lehet kötni egy jumper segítségével. Az adatátviteli sebesség igény szerint megválasztható a 3. táblázat pin beállításai szerint, itt is jumper-rel lehet a lábakat tápfeszültségre, vagy földre húzni. A Sleep funkciókat teljesen kikapcsoltam, mert a perifériák állapotát folyamatosan le kell tudni kérdezni. Természetesen a Mode 3 és Mode 4 lábak mind tápfeszültségre lettek kötve, ugyanis csak slave módban fognak ezek a SIG61-ek működni.

Mivel a két átviteli frekvencia használata bármikor szükséges lehet, mindkettőhöz egy-egy megfelelő kerámia szűrőt raktam. Sajnos a tervezés idején igen hiányos készlet miatt csak az 5,5 MHz-hez találtam SMD szűrőt, a gyártói adatlap által ajánlott Murata SFSKA5M50CF00-R3-t. A másik frekvenciához egy egyszerű THT-s SFE 6,5 MHz szűrőt rendeltem.

A hálózaton minden egység saját azonosítóval kell, hogy rendelkezzen. A megfelelő ID lábakat biteknek kell tekinteni, 4 láb, 4 bit, összesen 16 kombináció. Mivel minden eszköznek külön azonosítója van, ezeket a lábakat máshogy kell beállítani. Viszont minden kombinációhoz külön áramköri (PCB) tervet kellene készíteni, ez nagymértékben megdrágítaná a gyártást. Itt is alkalmazhattam volna jumper-eket, viszont egy meghibásodás esetén gyorsan kell tudni ID-t váltani a tartalék áramkörökön, ezért egy dip switch-et használtam. A switch egyik fele tápfeszültségre van kötve, másik meg egyenesen az ID lábakra. Ezzel a megoldással gyorsan és egyszerűen lehet cserélgetni az áramköröket.

A SIG61 működéséhez szükséges egy 4 MHz-es oszcillátor. Ezt az OscIn és OscOut lábakra kell kötni, emellett egy-egy kapacitással a földre kell húzni ezeket a lábakat. A kiválasztott oszcillátor: Epson MA-506.



17. ábra. Epson MA-506 4MHz bekötése és modellje.

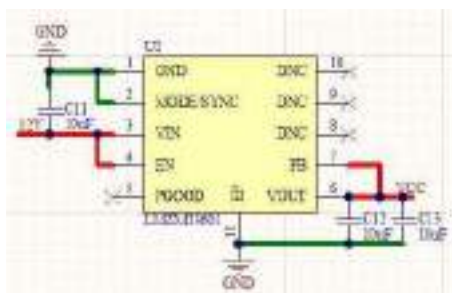
A SIG61 állapotáról visszajelzést lehet kapni az RxOn és TxOn lábokról. Ha van adás, illetve vétel ezek a lábak magas állapotba kerülnek. Visszajelzésként egy-egy LED-et raktam ezekhez a lábakhoz. A lábak kimenetként viselkednek, tehát 3,3V feszültséget és maximum 8 mA áramot adnak le. A kiválasztott LED egy Vishay által gyártott TLMS1000 SMD LED. 1,8 V-os nyitófeszültségével és tipikusan 2 mA-s fogyasztásával tökéletesen illik a kimenet korlátozott tápellátásához. A LED-el sorba kötött ellenállás feladata, hogy a LED-en eső feszültség sose legyen nagyobb a letörési feszültségénél. Az ellenállás értéke egy egyszerű Ohm törvénnyel kiszámolható,

$$R = \frac{3,3V - V_f}{I_f} = \frac{3,3V - 1,8V}{2mA} = 750\Omega, \quad (1)$$

ahol R a LED ellenállását, V_f a nyitófeszültségét, I_f a nyitóirányú áramát jelöli.

Az áramkörön két csatlakozó található. Egyik a tápvonalhoz tartozik, a másik pedig a SIG61 be- és kimeneteihez. A második csatlakozón helyet kapott egy 3,3V-os láb is, így, ha szükséges a SIG61 áramkörét lehet használni tápforrásként.

Végül a tápellátásról gondoskodtam. Az autó teljes kommunikációja 12V-on történik, viszont a SIG61-es IC tápfeszültsége 3,3V. Ehhez egy a Texas Instruments által gyártott LMZM23601 Step-Down konverterét használtam. A konverter adatai: $V_{in} = 4V-36V$, $V_{out} = 3,3V$, $I_{out} = 1A$. A nagy kimeneti áram amiatt szükséges, mert tervezés során az is szempont volt, hogy lehessen a kimeneti oldalról is levenni 3,3V tápforrást egyes perifériák meghajtására. Alap esetben a SIG61 és környezete maximum 50mA-t fogyaszt. A konverter bekötése az 18. ábrán látható.



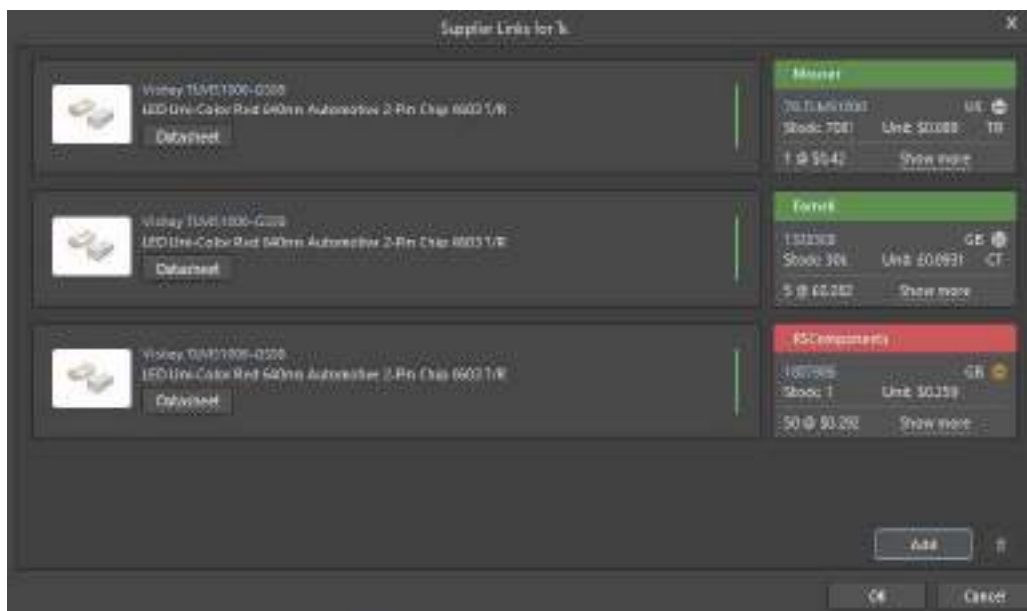
18. ábra. LMZM23601 Stepdown konverter bekötése.

A be és kimeneti kapacitások megválasztása fontos, hogy a konverter stabilan tudjon működni. A megfelelő bemeneti kapacitás megválasztása csökkenti a bemeneti áram hullámosságát, illetve kiegyenlíti a bemeneti feszültséget. Bemeneti kapacitás nélkül nagy EMI sugárzás lenne, ami tönkretesz a környező elektronikai eszközöket. A kimeneten minél kisebb ESR-rel (equivalent series resistance) rendelkező kondenzátort kell választani, mivel ez az ellenállás párhuzamosan kapcsolódik a terhelésre. Jelen esetben a két kimeneti kapacitás ESR-je párhuzamosan van kapcsolva, így eredő értékük csökken. Mind a bemeneti és kimeneti kapacitások a lehető legközelebb lettek elhelyezve az áramkörön. A gyártói ajánlás alapján nagyfeszültségű, X7R dielektrumú kondenzátorok lettek kiválasztva. A konverter elé még egy ferritgyűrűt is elhelyeztem, hogy tápvonalról le legyenek választva az áramkör kapacitásai, és az onnan érkező zavarokat is megsűrje. A konverter által előállított feszültséget bekötöttem a SIG61 IC megfelelő Vdd pinjeihez. Minden pinhez egy 100 nF-os kondenzátort raktam, a lehető legközelebb, hogy az esetleges nagy frekvenciás feszültség tüskéktől megvédje az IC-t, illetve energia tárolóként is szolgáljon a konverter kimenetének hullámszásainál.

A tápvonalról beérkező zavaroktól meg kell védeni a SIG61-et ezért az PowerLine vonal Schottky diódákkal és egy felüláteresztő szűrővel is el lett látva. A plusz szűrőt opcionálisnak hagytam meg, ha a környezet nem követeli meg a használatát, egyszerű jumper-rel áthidalható.

Miután az összes alkatrészt felhelyeztem a kapcsolási rajzra, és mindent megfelelően összekötöttem, kezdtem el hozzárendelni minden alkatrészhez egy cikkszámot. Általánosabb alkatrészekenél, például ellenállásoknál, kondenzátoroknál érdemes előbb valamelyik disztribútor készletét megnézni, és kiválasztani a megfelelő darabokat. Ilyen elemek választásánál törekedtem a lehető legkisebb méretű alkatrészeket kiválasztani, amit még kézzel be lehet szerelni, ezért legfőbbképpen 0402 és 0603 nagyságúakat választottam. Egy-egy elemhez a Supplier Links fülön lehet cikkszámot és forgalmazót hozzárendelni. Ezen a fülön lehet kiválasztani melyik forgalmazótól

rendeljük be az adott alkatrészt. A döntést segíti, hogy a készletről és az egységárakról is információt kapunk. Miután minden alkatrészt kiválasztottam, láttam neki a PCB tervezésének.

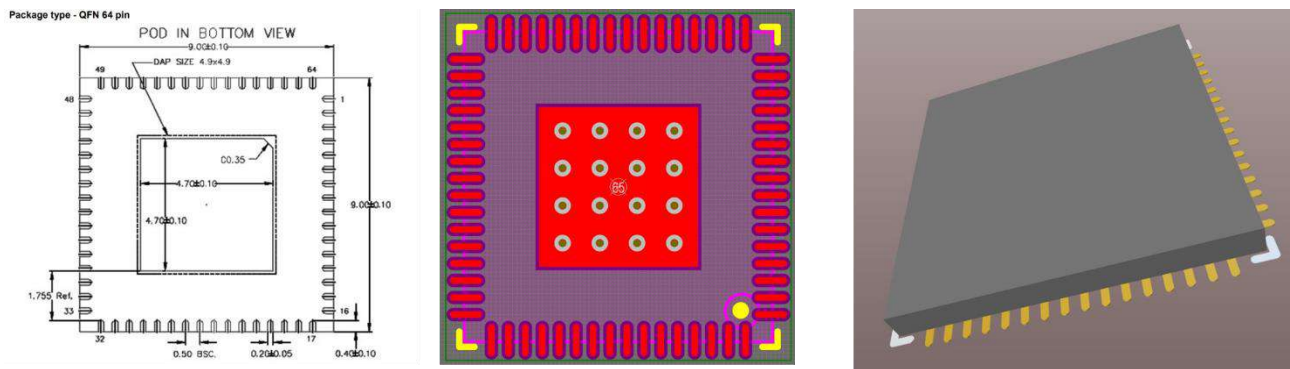


19. ábra. Beszállító választása az alkatrészekhez.

3.4. Nyomtatott áramköri tervek

A PCB tervezése mindig a kapcsolási rajzon alapszik, ezért fontos, hogy az ellenőrizve legyen. Szerencsés esetben minden alkatrészhez tartozik footprint és 3D-s modell is, ilyenkor ezt a program automatikusan betölti a Vault szerverről. Viszont, ha a projektünk tartalmaz egyedileg készített elemeket, azokat szintén nekünk kell berajzolni.

A SIG61 szimbóluma már kész, viszont a footprint és a modell hiányzik. Ezeknek az elkészítéséhez egy PCB Library-t kell a projekthez adni. Az Altium több módot biztosít egyedi modellek készítésére. Az alapvető, hogy mindent a készítő rajzol meg, ez főleg egyedi tokozású alkatrészeknél jellemző. A másik módszer, hogy a program belső footprint generátorát használjuk. A SIG61 egy 64 pin-es QFN tokozású alkatrész, ezért én az utóbbit használtam. A generálás során minden fizikai méretet megadtam az IC-ről, ezt a gyártói leírásban lehet megtalálni. A végeredmény a 20. ábrán látható:



20. ábra. SIG61 footprint-je és 3D-s modellje.

A program megrajzolja az összes lábat, a hozzá tartozó paszta maszkokat, a feliratokat, jelöléseket, és magát a 3D-s kinézetet is. A többi egyedi alkatrésznel már kézzel rajzoltam be ezeket, az egyedi lábméretek és tokozások miatt.

A PCB tervezése előtt be kell állítani a rétegek típusát, és az anyagvastagságukat. A Eurocircuits PCB gyártó cég honlapján található több fajta gyártás technológiából én a standard pool paraméterei alapján dolgoztam, mivel ez a gyártás fajta a legköltséghatékonyabb az összes közül, és a projektemnek teljesen megfelel. A standard pool szolgáltatással akár 8 rétegű áramköröket is lehet készíteni. A SIG61 esetében a rétegek száma: 2, a dielektrikum anyaga FR-4 vastagság 1,15mm. A dielektrikum két oldalára viszik fel a réz rétegeket, ezek 0,018mm. Ezt fedi le egy 0,01mm-es forrasztó maszk majd egy fedő réteg. Altium-ban ezeket a paramétereket a Layer Stack Manager fülön lehet beállítani.

#	Name	Material	Type	Weight	Thickness	Dk	Df
	Top Overlay		Overlay				
	Top Solder	Solder Resist	Solder Mask		0,01mm	3,5	
1	Top Layer		Signal	1/2oz	0,018mm		
	Dielectric 1	FR-4	Dielectric		1,55mm	4,8	
2	Bottom Layer		Signal	1/2oz	0,018mm		
	Bottom Solder	Solder Resist	Solder Mask		0,01mm	3,5	
	Bottom Overlay		Overlay				

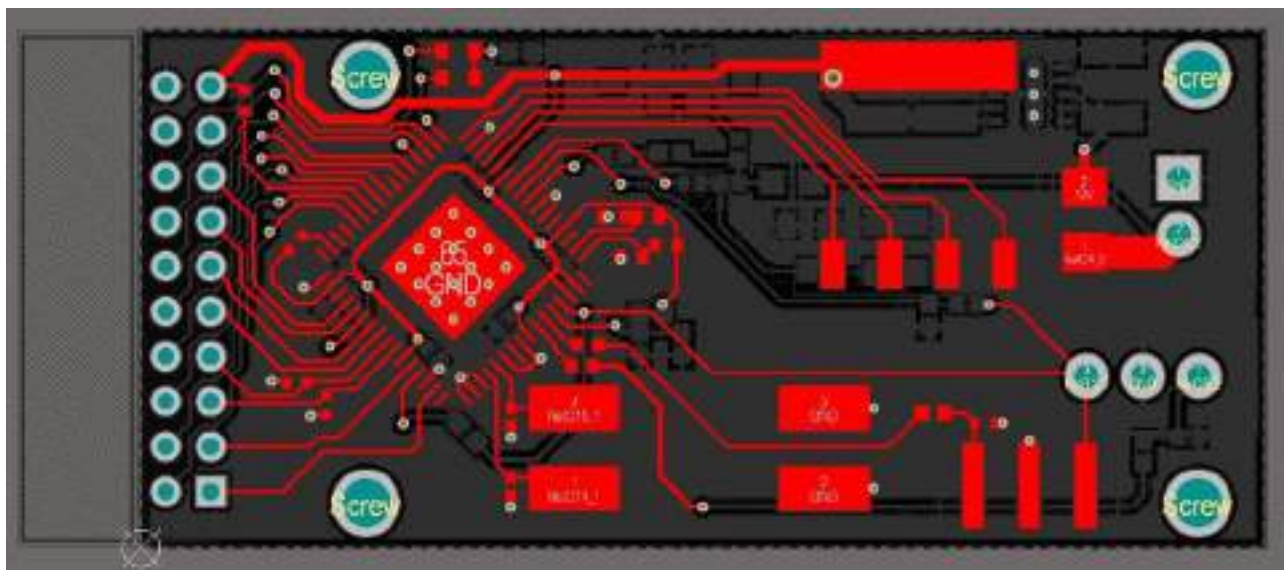
21. ábra. Layer Stack Manager Altiumban.

A standard pool a rétegvastagságok mellett meghatározza a minimum vezetékvastagságot, a furatméreteket, és magának a nyáknak a méreteit.

A tervezési szabályok beállítása a másik fontos lépés a tervezés megkezdése előtt. Itt lehet megadni rengeteg paramétert, amik az áramkör formáját nagymértékben befolyásolják. A legfontosabbak ezek közül: clearance, ez adja meg az egyes objektumok közötti minimum távolságot, legyen az vezeték, furat, alkatrész. Itt lehet beállítani milyen vastag a vezetékek mérete, az átvezetések, via-k, nagysága. A csapaton belül egy egységes szabálykészlettel dolgozunk, ezért én is ezt importáltam.

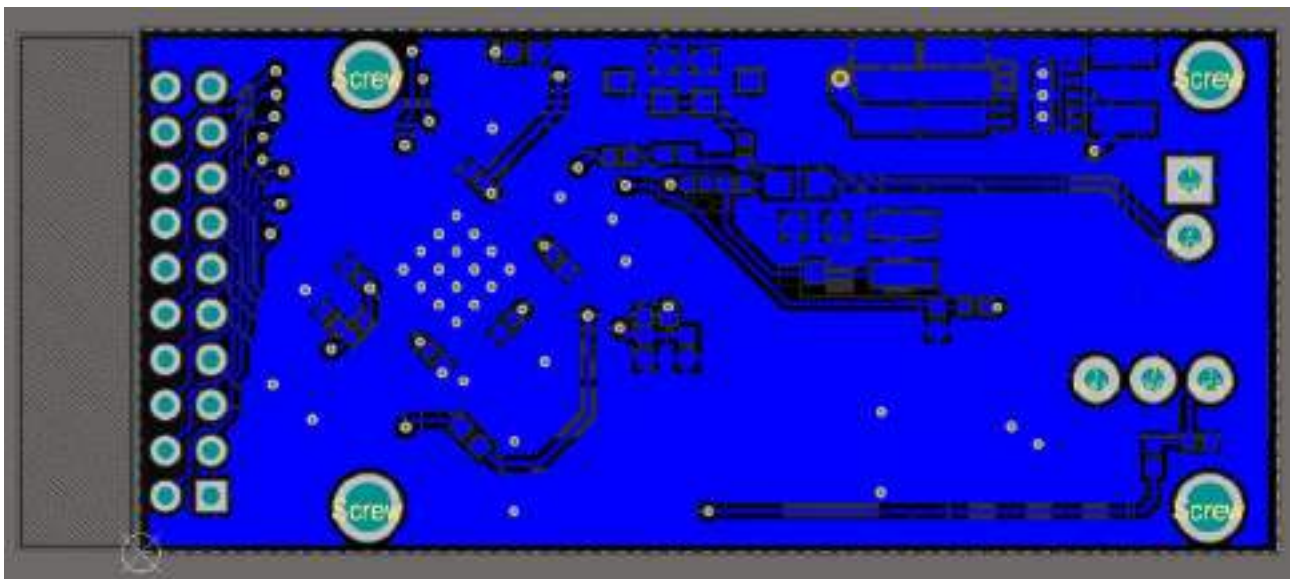
Miután minden alkatrészhez meg van a megfelelő modell és footprint és a megfelelő szabálykészlet is be van állítva, el lehet kezdeni a PCB tervezését. A kapcsolási rajzról importálni kell mindent, ilyenkor a program az alkatrészeket és azok összeköttetéseit berakja a munkatérbe. Innentől kezdve lehet elhelyezni minden elemet. Az elrendezésnél törekedtem a hely kihasználására. Az áramkör kétoldali lesz, viszont figyelembe kell venni, hogy a nagy alkatrészek mind egy oldalra kerüljenek, így könnyebb lesz az összerakás.

Itt is először a SIG61-et helyeztem el és a környező alkatrészeit raktam mellé. A kapcsolási rajzon, ahol jelölve volt, hogy „Close to MCU”, azokat az alkatrészeket prioritizáltam lerakásnál, hogy minél kisebb legyen a távolság. A PCB bal oldalán található a 20 pines csatlakozó amire a perifériákat lehet kötni, mellette van maga a SIG61-es IC. A nagyobb alkatrészek, mint az oszcillátor, a kerámia szűrők és a DIP switch is a felső oldalon kapott helyet. Az alsó oldalon jellemzően a tápellátáshoz tartozó alkatrészek kaptak helyet.



22. ábra. Vezetékezés a PCB felső rétegén.

Az alkatrészek elhelyezése után kezdtem el mindent összekötni. A jelvezetékek 0,2mm míg a tápellátást biztosító vezetékek 0,5mm szélesek. Gyakran fordult elő, hogy az alkatrész nem fért el a felső rétegen, viszont a jel onnan jön, ilyenkor kell átvezetést, via-t használni. A via méretei egységesen: 0,25 mm átmérőjű furat, 0,5 mm átmérőjű réz felület. A hátsó oldalon a kevesebb vezeték és kis alkatrészek miatt a földelés nem vezetékekkel van összekötve, hanem egy úgynevezett polygonnal. A polygon a teljes alsó réteget lefedi, ahol alkatrész vagy más jelvezeték van, azt körbe veszi. Ezzel a megoldással minden föld lábat a legrövidebb úton egy átvezetéssel le lehet kötni.



23. ábra. GND polygon az alsó rétegen.

A polygon-hoz többféle módon csatlakozhatnak a vonalak. A pad-ekre és furatokra úgynevezett relief csatlakozás érvényes, azaz csak pár ponton csatlakozik rá, míg a via-t teljesen körbe veszi. Fontos, hogy mindenhova el tudjon jutni a polygon, ahol GND lekötés van.

Miután minden vezetékezés és alkatrész elhelyezése megtörtént, le kell ellenőrizni az elkészült áramkört. Az ellenőrzés során a korábban beállított szabályokat vizsgálja meg a program. Ha van szabálysértés, azt pontos hibaleírással jelzi a program egy jelentés formájában. Minden hibát ki kell javítani a PCB-n, csak utána lehet legenerálni a gyártási fájlokat. A leggyakoribb hiba a SIG61 áramkörén a nem megfelelő pasztamaszkok közötti távolság volt, ilyenkor minden maszkot egységesen kicsinyítettem.

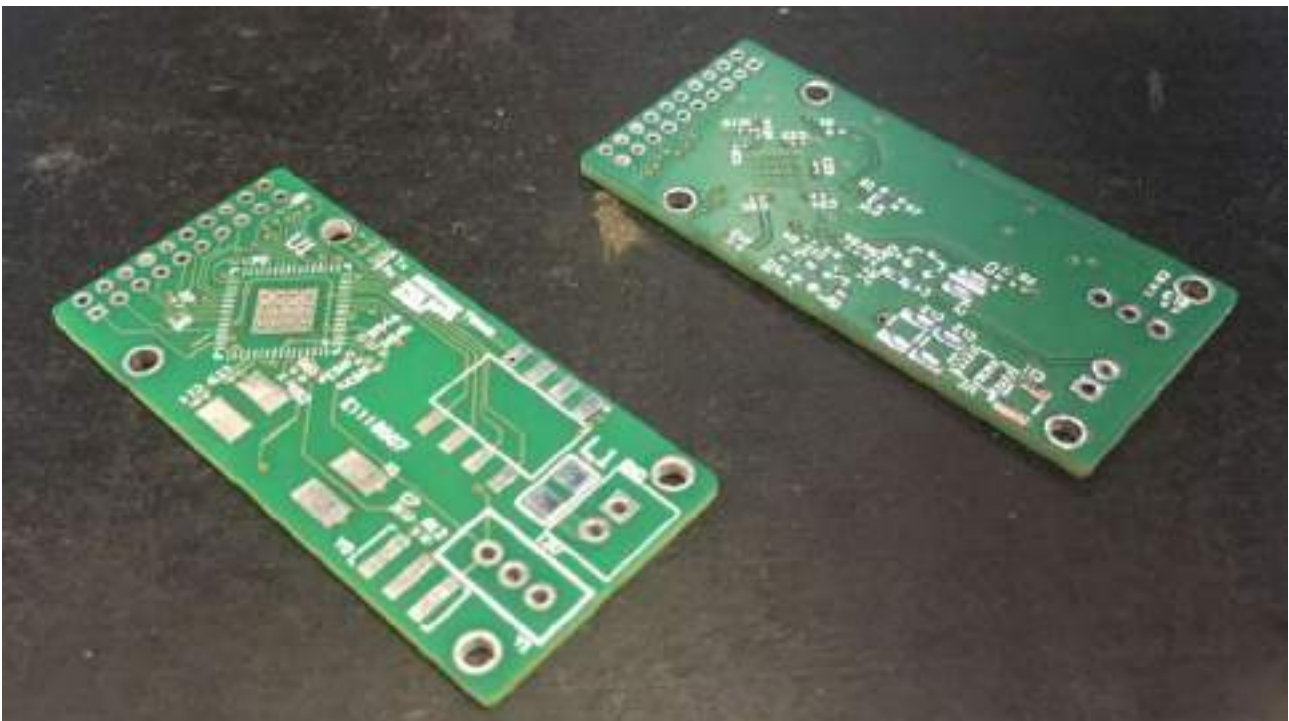
Emellett a PCB formáját is meg kell adni, ezt a Board Planning módban lehet megtenni. Itt a körvonalat szinte bármilyen geometria szerint be lehet állítani. A rendezésnél törekedtem arra, hogy minél kisebb helyen tudjam elhelyezni az összes alkatrészt, a végleges panel mérete 50x23 mm. A

kész körvonalat egy Keep-Out vonallal is jelölni kell, a gyártás során, ez alapján vágják ki a PCB méretét.

Ha az áramkör minden szabálynak megfelel, meg van a köralakja és a jelölések is egyértelműek és láthatók, meg lehet kezdeni a gyártási fájlok generálását. Ehhez egy OutputJob-ot fájl kell létrehozni. A projekthez tartozó dokumentációkat, gyártási fájlokat, képeket, videókat itt lehet megadni. Számomra elsődlegesen a dokumentálás és a gyártáshoz szükséges fájlok elkészítése volt. Utóbbihoz úgynevezett Gerber és Drill fájlok tartoznak. Minden réteghez tartozik külön egy Gerber fájl, ez alapján viszik fel a különböző rétegeket a PCB gyártók. A Drill fájlok adják meg hol és mekkora furatot kell a PCB-be fúrni és kell e réz bevonat rá.

4. Gyártás, beültetés

Az elkészült gyártási fájlokat kiküldtem a Eurocircuits Kft. számára. A cégtől 1 hetes várakozás után kaptam meg az elkészült nyomtatott áramköröket. Párhuzamosan a gyártással a BOM lista alapján megrendeltem az alkatrészeket a különböző forgalmazóktól, majd ezeket beültároztam.



24. ábra. Legyártott nyomtatott áramkör alsó és felső oldala.

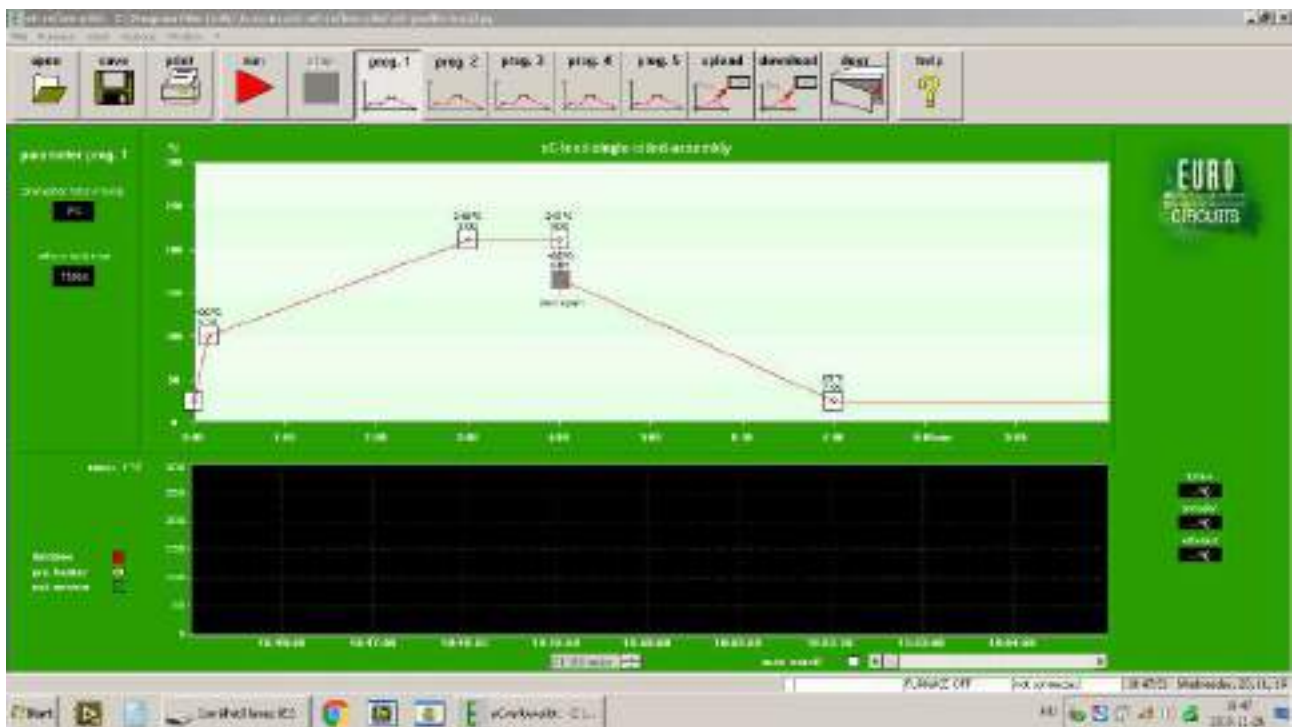
A beültetés első része a pasztázás. Az áramkörön minden egyes lábat be kell vonni forrasztó pasztával, erre lehet később felhelyezni az alkatrészeket. A pasztát egy pneumatikus adagoló segítségével vittem fel, ennek az adagolási sebességét a nyomás változtatásával lehet állítani. Minden SMD pad-re megfelelő mennyiségű pasztát kell felvinni, hogy a kemencében végbemenő folyamatok alatt rendesen létre tudjon jönni a forraszkötés. A QFN tokozású SIG61 IC pasztázásánál kellett a legkörültekintőbben eljárnom, ugyanis az IC lábai a tok alatt helyezkednek el, ezért lényegesen nehezebb az esetleges összefolyásokat kijavítani. A vékony lábméretek miatt itt minden oldalra egy paszta csíkot kell felvinni, ez a csík később teljes mértékben a lábakra fut fel. A paszta, amit használtam tartalmaz ónt, ez a kemencézésnél lesz fontos információ. Természetesen egyszerre csak az áramkör egyik oldalát lehet pasztázni, én az alsó oldallal kezdtem, mert ott a kisebb alkatrészek helyezkednek el.



25. ábra. Különböző csipeszek a beültetéshez.

Miután megtörtént a pasztázás, fel lehet rakni az alkatrészeket. Előjáróban érdemes kinyomtatni a BOM listát, hogy lehessen tudni, pontosan melyik alkatrész hova tartozik. Emellett az alsó és felső oldali alkatrészeket szétválogatni, hogy a beültetés során minél gyorsabban és hatékonyabban lehessen dolgozni. Minden SMD alkatrész szalagban érkezik, ezért a szalagból előre ki kell szedni a darabokat. Felhelyezésükhöz ESD védett műszerész csipeszeket használtam. Ezeknek is több fajta kialakítása van: egyenes, hajlított végű, tompa stb. Az alkatrész formája és mérete határozza meg melyik csipesszel lehet a legegyszerűbben beültetni. A kiszedett alkatrészeket a BOM listában ki kell keresni, pontosan melyik azonosító tartozik hozzájuk, ezt az áramkörön is meg kell

keresni, és elhelyezni. Fokozottan kell figyelni, hogy az alkatrész lábkiosztása szerint legyen az beültetve. A SIG61-en az 1-es lábat egy pont jelzi a tokozáson, és a PCB-n is, ennek megfelelően helyeztem fel. A tokozás oldalán láthatóak a lábkivezetések, viszont kicsiny méretük miatt mikroszkóp alatt kellett beültetnem és helyére igazítanom ezt az alkatrészt. Hasonlóan jártam el a DCDC konverternél és egyéb kényesebb alkatrészeknél is. Miután mindent felhelyeztem az egyik oldalra, mikroszkóp alatt még egyszer átvizsgáltam mindenhol elegendő-e a paszta mennyisége, és az alkatrészek is jó pozícióban vannak. Az esetleges hibákat kijavítva indulhat meg a kemencézés folyamata.



26. ábra. eC-reflow-pilot kezelő felülete. Ónos paszta melegítési profilja.

A Eurocircuits által használt kemencét használtam a gyártás folyamán. Az eC-reflow-mate a cég egyedi gyártású kemencéje, melyet prototípus PCB-k gyártására fejlesztettek ki. A kemencéhez járó eC-reflow-pilot szoftver segítségével lehet beállítani a melegítési folyamatot. A folyamat áll egy felfűtésből, egy homogenizálásból, egy nagy hőmérsékletű melegítésből és a kikapcsolásból. A felfűtés alatt a PCB gyorsan elér egy adott hőmérsékletet, majd innentől lassan emelkedik a PCB hőfoka. Amikor már minden része átmelegedett, a kemence elkezd hirtelen emelni a hőmérsékletet a paszta olvadási pontja fölé. A maximum hőfokon van egy kis tartásidő, majd a fűtőszálak kikapcsolnak, a folyamatnak vége. A szoftverben többféle profilt meg lehet adni, ezekben időbeli és

hőmérsékleti különbségek vannak. Én ónos pasztát használtam, ennek az olvadási pontja 183°C , ezért a maximális kemencézési hőmérséklet 210°C körül van. Ónmentes pasztánál viszont sokkal magasabb az olvadási pont 214°C ezért ott más profilt kell használni. A megfelelő profil beállítását befolyásolja a PCB mérete, és a paszta fajtája.

A gyártás folyamán egyszerre több áramkört készítettem elő a kemenczésre, így időt spórolva, ugyanis a melegítési folyamatok között a kemencének ki kell hűlnie. A kemencében középre kell a PCB-ket elhelyezni, mivel a belső szenzorok is itt helyezkednek el. A belső szenzorok mellett egy külső szenzort a nyák lapba kell beilleszteni, ugyanis ennek az értékei alapján szabályozza a folyamatot a szoftver. A nyákokat vagy a tálcára vagy távtartókra kell elhelyezni. Érdekes a távtartókra helyezni őket, mert a tálca felülete hőt von el, ezért a folyamat lassabb lesz.



27. ábra. Kemenczési folyamat. [A2]

A kemenczési folyamatot szabad szemmel is végig lehet követni a kemencén található ablakon keresztül. Fontos figyelni, hogy az olvadási pont átlépésekor a PCB-n mindenhol megolvadjon a paszta és az alkatrészek a helyükre úszanak. Bármilyen rendellenesség esetén le lehet állítani a folyamatot a kemencén és a szoftveren is. Miután a folyamat véget ért, a tálca automatikusan kinyílik. Ha a nyákok még nem hűltek le, a paszta még olvadt állapotban van, a hirtelen kinyitás hatására azok el tudnak mozdulni, ezért fontos, hogy a fiókot megfogjuk, és lassan engedjük ki.

Az elkészült forrasztást az áramkör lehűlése után át kell ellenőrizni. Az átvizsgáláshoz szintén mikroszkópot használtam. Ilyenkor kell minden alkatrészt megnézni, hogy van-e összefolyás, a

megfelelő helyre úsztak be. Az esetleges hibákat már csak forrasztópákával lehet kijavítani. A forrasztáshoz a Weller WXMP RTM 008 S pákát használtam, nagy teljesítménye és hegyes vége miatt. Leggyakrabban a túlzott pasztázás miatt apró golyók jelennek meg az alkatrészek mellett, ezeket egyszerűen le kell kaparni. Az egymáshoz közeli elemek hajlamosak összeúszni, ezért itt teljesen kellett levennem mindkettőt, és külön visszahelyezni őket. Az olyan alkatrészeknél, ahol a lábak a tokozás alatt vannak, csak hőlégfúvóval lehet leszedni. Először az egész nyákot kell felmelegíteni, majd a problémás alkatrészre kell fókuszálni.



28. ábra. Kemenczési hibák javítása.

Miután minden forrasztás ki lett javítva, fel kell helyezni a THT-s alkatrészeket is. Itt egyszerűen a furatba kell őket helyezni majd beforrasztani. Ezzel a folyamattal elkészült az áramkör legyártása.



29. ábra. Elkészült SIG61-es áramkör.

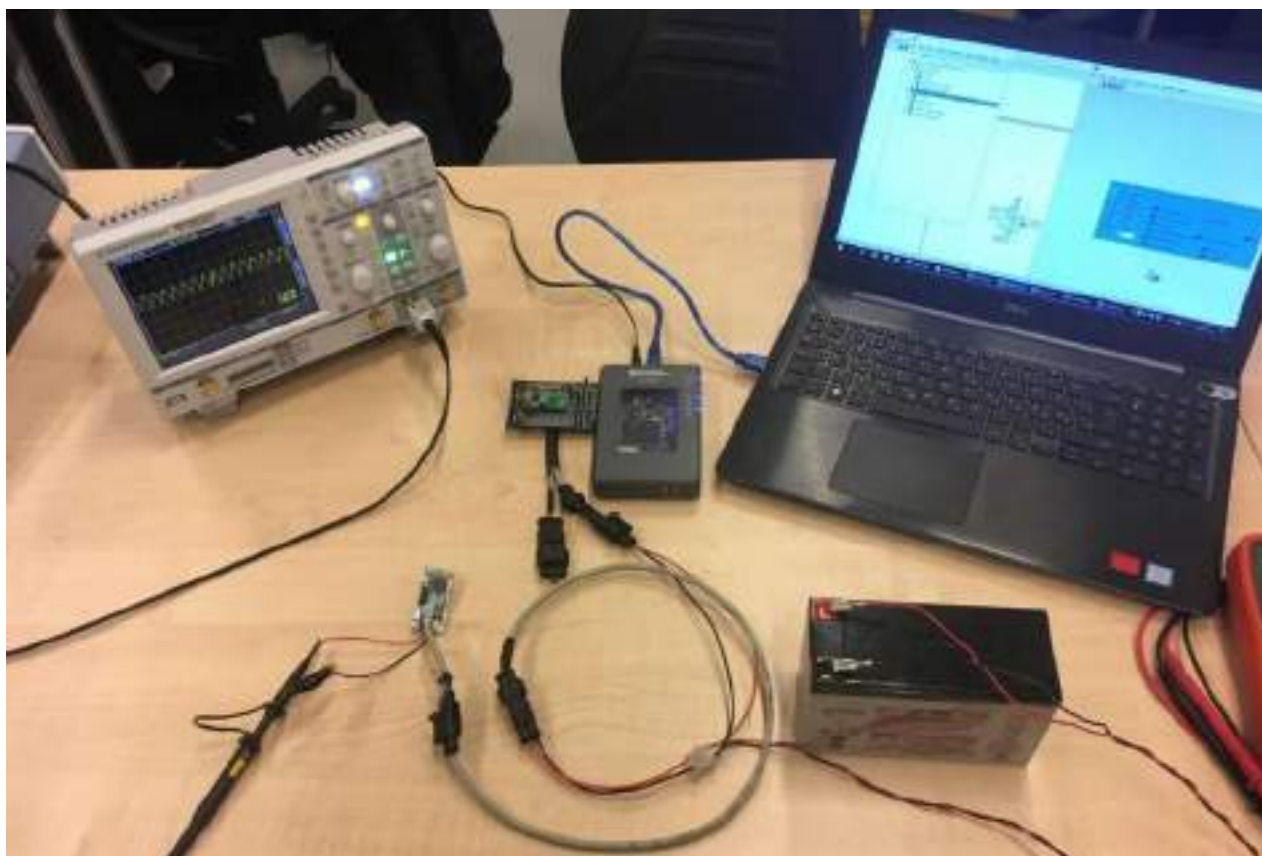
5. Tesztelés

A gyártás után le kell tesztelni az áramköröket, hogy valóban megfelelően működnek-e. Először megmértem multiméterrel az áramkörön van-e valahol zárlat, ahol volt ott ismét forrasztással kijavítottam. Ha mindent rendben találtam, megvizsgáltam, hogyha meg táplálom az áramkört, akkor helyesen működik-e. A 12 V bemeneti feszültséget programozható labortápegység segítségével állítottam elő, és beállítottam 50 mA-es áramkorlátot, majd elsőként az áramfelvételt néztem. Ha az áramkör 10 mA környékén vesz fel teljesítményt, akkor helyesen működik, ha elenyészően kicsit, pár mA-t, akkor valahol szakadás van, ha az áramkorlátot eléri, akkor rövidzár áll fenn. Ezután a konverter helyes működését vizsgáltam, a kimeneti feszültségének 3,3 V-nak kell lennie. Ha ez a feszültség nincs meg, akkor a konverter helytelenül működik, lehet a probléma forrása a nem megfelelő forrasztás, vagy alkatrész hiba.



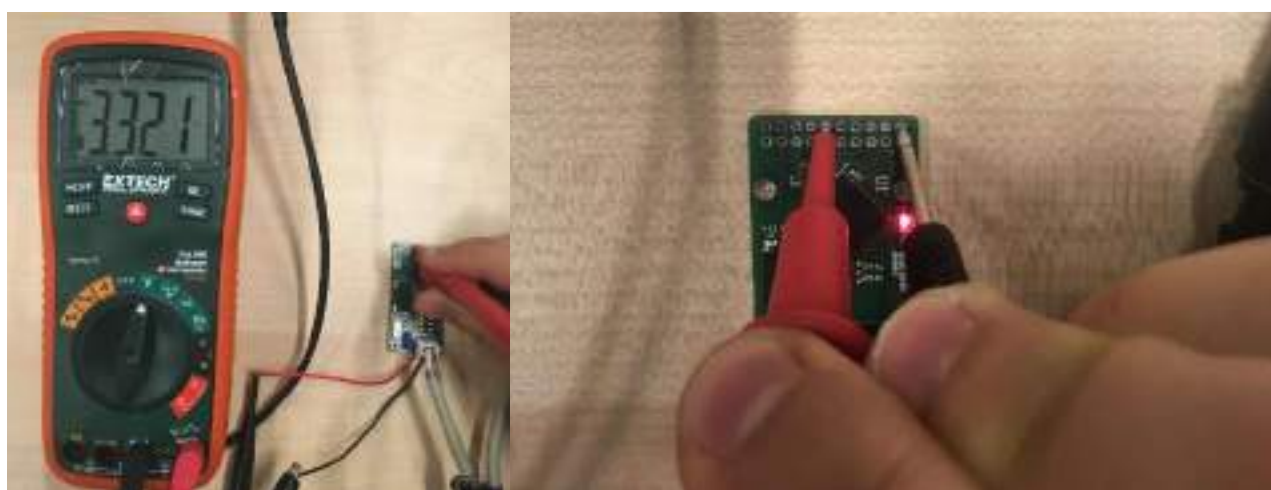
30. ábra. A SIG61 áramkör működés közben.

A méréshez egy összeállítást készítettem elő, ami hasonló az autóban levő rendszerhez. Ehhez szükségem volt egy myRIO-ra, egy üres kiegészítő kártyára, egy SIG60-ra, egy SIG61-re és áramforrásra. A myRIO-ra feltöltöttem a direkt erre a célra írt tesztprogramot. A kiegészítő kártyára ráforrasztottam a SIG60-at és a megfelelő helyre bekötöttem. Megcsináltam a kábelkorbácsot a teszteléshez, mindenhol a megfelelő polaritással kötöttem be a csatlakozókat. A tápellátásért egy 12 V-os akkumulátor felel. Ezután mindent összekötöttem, és elindítottam a rendszert. A myRIO-ra rácsatlakoztam számítógéppel, így real time tudtam követni a ki- és bementek állapotát.



31. ábra. A SIG61-es tesztösszeállítása.

A programmal lehet be lehet állítani, hogy melyik azonosítót tartalmazza a küldött üzenet és bármely kimenetét lehet vezérelni a SIG61-nek. A vezérelt kimeneteken meg kell jelennie a 3,3 V feszültségnek.



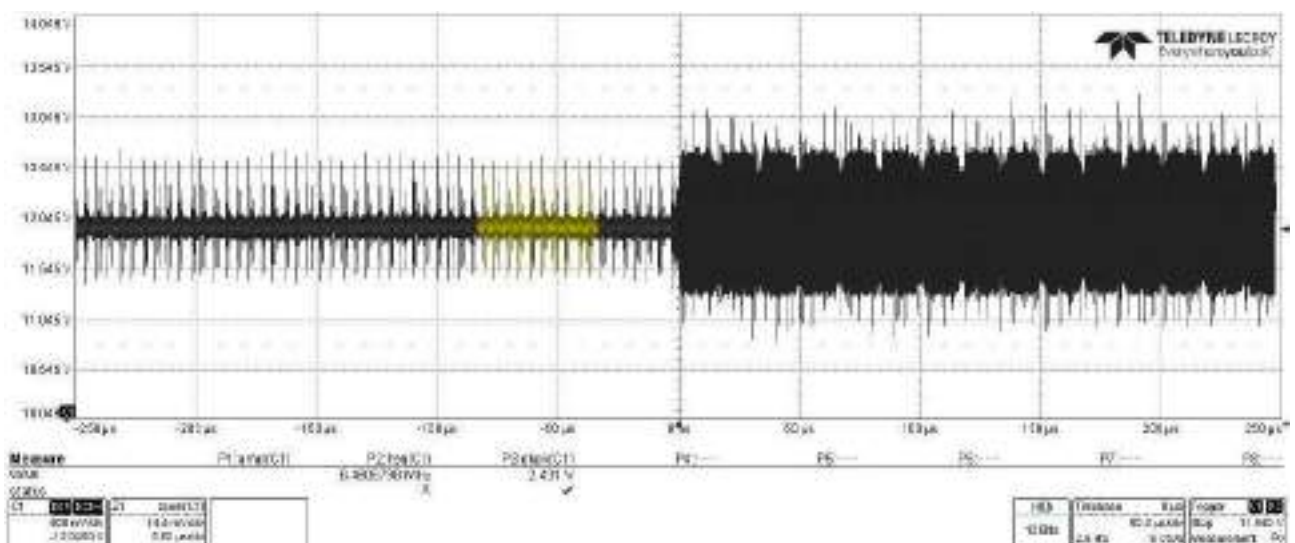
32. ábra. Kimenetek tesztelése.

Ezután jumper kábel segítségével magasra húztam egyes bemeneteket, ezt a tesztprogram érzékelte is.



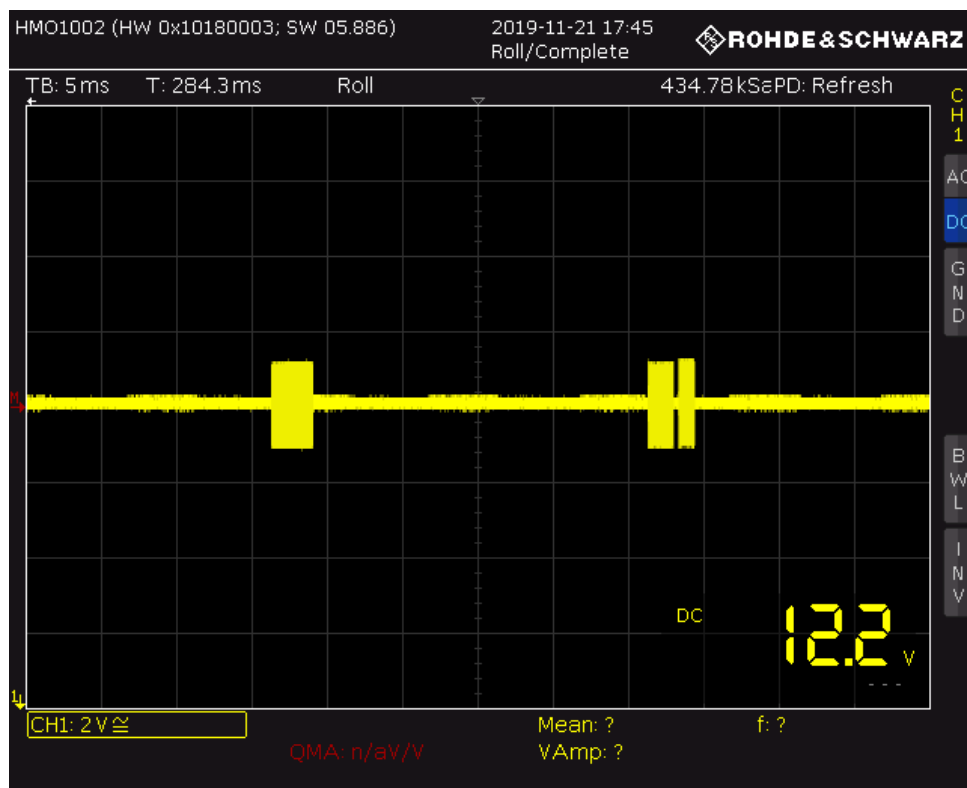
33. ábra. Bemenetek tesztelése. Tesztelő program kezelőfelülete.

A program figyeli a kommunikáció állapotát is, ha esetleg az írással vagy az olvasással gond van, azt jelzi. Miközben a program futott, oszcilloszkóppal megnéztem a tápvonalon, hogy látható a kommunikáció.



34. ábra. A tápvonalra ültetett üzenet.

A 34. ábrán jól látható, hogy az előre beállított 6,5 MHz frekvencián történik a kommunikáció. A Read és Write üzenetek jól megkülönböztethetők egymástól, ezt a 35. ábra mutatja.

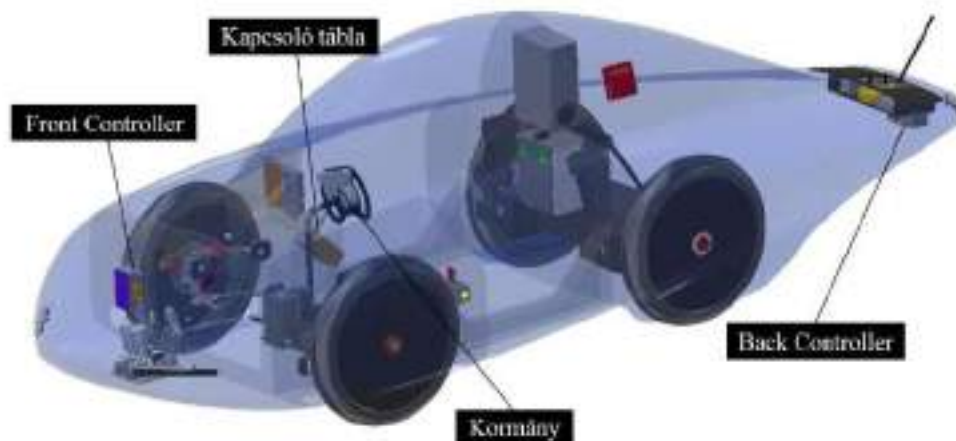


35. ábra. Baloldali Write és jobboldali Read utasítás.

A Read utasítást a master küldi ki a hálózatra a megfelelő azonosítóval, majd kis szünet után a megfelelő slave válaszol a bemeneteinek aktuális állapotáról. Ha minden megfelelően működik, az áramkör beépítésre kerülhet az autóba.

6 Konklúzió

Dolgozatom során egy egyedi LIN kommunikációban résztvevő slave eszköz megtervezését és megvalósítását mutattam be. Betekintést nyertem az áramkör tervezés alapjaiba, és a legyártásukba. Az elkészült áramkörök jelenleg a SZEmission autóban találhatók négy helyen: a kormányon, a kapcsolótáblán, az autó elejében és hátuljában.

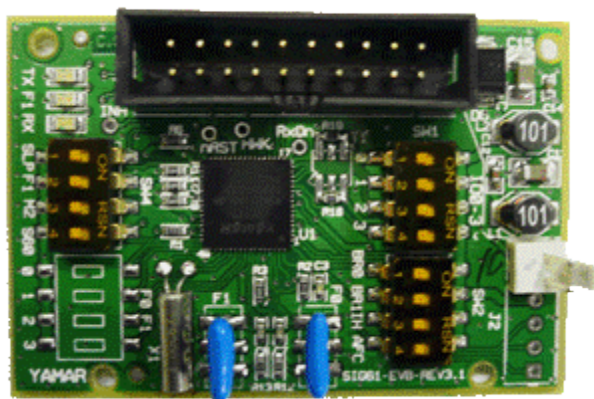


36. ábra. A SZEmission-ben található SIG61-ek helye.

Párjával, a SIG60-al együtt kezelik az autó perifériáit. A 2019-es Shell Eco Marathon-on már ezzel a rendszerrel futott az autó. A verseny területén rengeteg rádió, vezeték nélküli hálózat található, ezért a korábbi években a LIN kommunikációban voltak zavarjelenségek, azonban idén az új eszközökkel felszerelve nem volt ebből adódóan probléma.

7. Fejlesztési lehetőségek

Ezek az áramkörök speciálisan ehhez az autóhoz lettek tervezve, ezért más alkalmazásoknál csak nagyobb módosításokkal lehetne használni őket. Tervben van egy olyan áramkör megvalósítása, ahol mindent a felhasználói igények szerint be lehet állítani. Hasonlóan az azonosítóhoz, a frekvenciát, az átviteli sebességet és az alvó üzemmódot mind kapcsolóval lehetne állítani. A frekvenciaválasztásnál viszont figyelembe kell venni, hogy ahhoz megfelelő kerámia szűrőt is kell használni. A SIG61-k emellett alkalmasak master-ként is viselkedni, ezért lehetséges egy master slave hibrid áramkör megvalósítása is. Egy ilyen áramkört lehet látni a 37. ábrán.



37. ábra. Yamar által gyártott SIG61 áramköre. [A3]

Az egyik legnagyobb hiányosság a stencilfólia volt. Sajnos a szűkös idő miatt nem volt már idő ezt megrendelni, ezért a PCB-re kézzel vittem fel a paszta réteget. A nem elegendő vagy túlzott paszta mennyiség viszont később tesztelés közben okozott gondokat, ezért sokszor kellett utólag javítani az áramkörökön. A legproblémásabb alkatrész ebből a szempontból a DCDC konverter volt, nagyon apró lábkievezetései miatt. A stencil fólia segítségével a paszta réteg felvitele pár másodperces folyamat. A fólia ott van kilyukasztva, ahol a lábak találhatók a PCB-n. A nyákot a megfelelő pozícióba kell felhelyezni, majd ráfeszíteni a fóliát. Ezután egy kés segítségével fel lehet kenni a forrasztópasztát. Ha a fóliát nem használjuk többször, azt alaposan le kell tisztítani, hogy ne tudjon rászáradni a paszta, és lehessen legközelebb is használni. Stencilezés után el lehet kezdeni az alkatrészeket beültetni.

A jövőbeli fejlesztések között tervben van egy olyan hálózat kiépítése, amelyben nincs LIN rendszer, viszont a CAN-hez lehet használni a YAMAR által fejlesztett DCB termékcsaládot. A CAN hálózatra mindenképp szükség, ugyanis a versenyautót a csapat autonóm kategóriában is meg akarja versenyeztetni. Ezekkel az IC-kel ugyanúgy le lehetne csökkenteni a kábelek számát az autóban, mint a jelenlegi LIN hálózathál.

Egy másik irány a fejlesztésben, az autóban levő fogyasztók minimálisra csökkentése. Ez azt jelentené, hogy a BMS-en és a motorvezérlőn kívül minden eszközt amire az autónak nincs szüksége a megfelelő működéshez, kivesszük, vagy helyettesítjük. Jelenleg ezek a tervek, amiken az elektronika részleg dolgozik, hamarosan elkészülnek az első prototípusok és tesztek, hogy tudjuk milyen rendszerrel megyünk ki a 2020-as Shell Eco Marathon-ra.

Irodalomjegyzék

- [1] Dr. Kováts Miklós, Dr. Szalay Zsolt: *Gépjárművek buszhálózatai*
Maróti Könyvkereskedés és Könyvkiadó Kft., Budapest (2013), p. 34-101.
- [2] Lin Specification Package Revision 2.2A
https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf,
2019.10.24.
- [3] SIG60 Datasheet
<https://yamar.com/datasheet/DS-SIG60.pdf>, 2019.08.30.
- [4] SIG61 Datasheet
<https://yamar.com/datasheet/DS-SIG61.pdf>, 2019.08.30.
- [5] Introduction to the Controlled Area Network
<http://www.ti.com/lit/an/sloa101b/sloa101b.pdf>, 2019.09.28

Ábrajegyzék

[A1] <https://yamar.com/technology/>
2019.9.10

[A2] <http://be.eurocircuits.com/shop/offtheshelf/product.aspx?ad=9524&ano=ec-reflow-mate&an=ec-reflow-mate&s=ec-prototype-equipment>
2019.11.10

[A3] <https://yamar.com/product/sig61-evb/>
2019.11.15

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani konzulensemnek Csizmadia Miklósnak, aki a dolgozat megírásával kapcsolatban hasznos tanácsokkal látott el.

Köszönöm a segítséget Szeli Zoltánnak, aki türelemmel adta át tudását az áramkörök tervezéséről és gyártásáról.

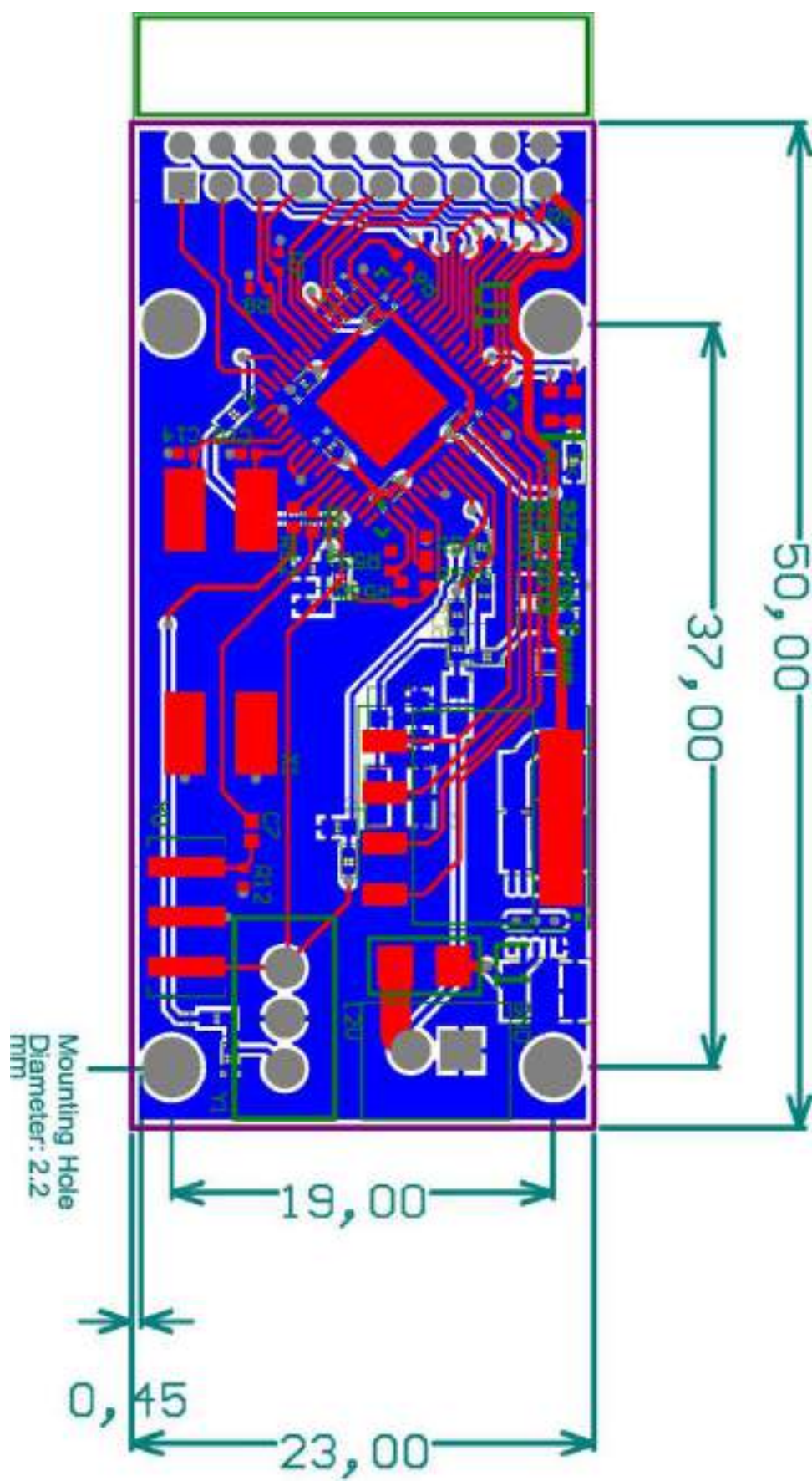
Szeretném megköszönni családomnak, hogy mindig türelemmel mellettem álltak és támogattak.

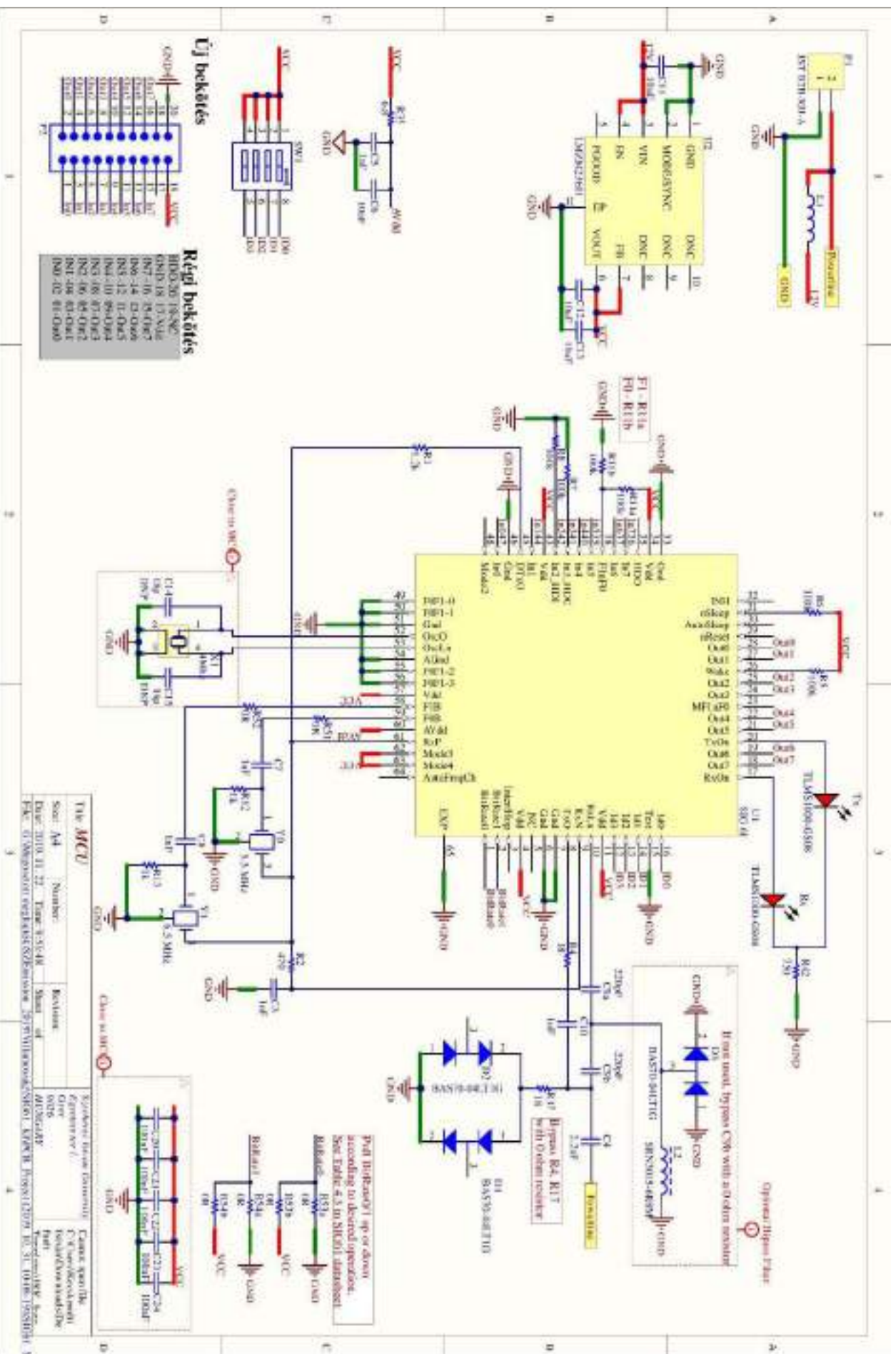
Szeretném megköszönni a barátaimnak, hogy mind tanulásban, mind a dolgozat megírásában segítettek nekem.

És szeretném megköszönni a SZEenergy Team tagjainak, hogy részt vehettem ennek az új autónak a megtervezésében és összeszerelésében.

A szakdolgozat elkészítését a GINOP-2.3.4-15-2016-00003 Felsőoktatási és Ipari Együttműködési Központ a Széchenyi István Egyetemen projekt támogatta.

Mellékletek







Design Rule Verification Report

Date: 2019. 11. 19.
 Time: 17:37:35
 Elapsed Time: 00:00:00
 Filename: [G:\Megosztott\meghajtó\SZEmission_2019\Villaország\SIG61_K\PCB_Project\2019-10-31_10-08-19\SIG61_EB_PcbDoc](#)

Warnings: 0
 Rule Violations: 12

Summary

Warnings	Count
Total	0

Rule Violations	Count
Clearance Constraint (Gap=0.15mm) (All) (All)	0
Short-Circuit Constraint (Allowed=Yes) (InNet('No Net')) (All)	0
Un-Routed Net Constraint (All)	0
Modified Polygon (Allow modified: No) (Allow shelved: No)	0
Width Constraint (Min=0.2mm) (Max=1.5mm) (Preferred=0.254mm) (All)	0
SMD Neck-Down Constraint (Percent=100%) (All)	0
Power Plane Connect Rule(Direct Connect) (Expansion=0.508mm) (Conductor Width=0.254mm) (Air Gap=0.254mm) (Entries=4) (All)	0
Hole Size Constraint (Min=0.25mm) (Max=4mm) (All)	0
Hole To Hole Clearance (Gap=0.254mm) (All) (All)	0
Minimum Solder Mask Silver (Gap=0.1mm) (All) (All)	5
Silk To Solder Mask (Clearance=0.1mm) (IsPad) (All)	3
Silk to Silk (Clearance=0.1mm) (All) (All)	4
Net Antennae (Tolerance=0mm) (Disabled) (All)	0
Height Constraint (Min=0mm) (Max=25.4mm) (Preferred=12.7mm) (All)	0

[Silk primitive without silk layer](#)

0

Total 12**Minimum Solder Mask Sliver [Gap=0.1mm] (All),(All)**

[Minimum Solder Mask Sliver Constraint: \(0.077mm < 0.1mm\) Between Pad C7-1\(34.725mm,5.975mm\) on Top Layer And Pad C7-2\(35.725mm,5.975mm\) on Top Layer \(Top Solder\) Mask Sliver \(0.077mm\)](#)

[Minimum Solder Mask Sliver Constraint: \(0.077mm < 0.1mm\) Between Pad C9b-1\(25.75mm,16.2mm\) on Bottom Layer And Pad C9b-2\(26.55mm,16.2mm\) on Bottom Layer \(Bottom Solder\) Mask Sliver \(0.077mm\)](#)

[Minimum Solder Mask Sliver Constraint: \(0.097mm < 0.1mm\) Between Pad R51-1\(20.375mm,9.025mm\) on Top Layer And Pad R52-1\(20.125mm,8.075mm\) on Top Layer \(Top Solder\) Mask Sliver \(0.097mm\)](#)

[Minimum Solder Mask Sliver Constraint: \(0.097mm < 0.1mm\) Between Pad R51-2\(19.375mm,9.025mm\) on Top Layer And Pad R52-1\(20.125mm,8.075mm\) on Top Layer \(Top Solder\) Mask Sliver \(0.097mm\)](#)

[Minimum Solder Mask Sliver Constraint: \(0.097mm < 0.1mm\) Between Pad R51-2\(19.375mm,9.025mm\) on Top Layer And Pad R52-2\(19.125mm,8.075mm\) on Top Layer \(Top Solder\) Mask Sliver \(0.097mm\)](#)

[Back to top](#)**Silk To Solder Mask (Clearance=0.1mm) (IsPad),(All)**

[Silk To Solder Mask Clearance Constraint: \(0.025mm < 0.1mm\) Between Pad C4-1\(28.05mm,16.25mm\) on Bottom Layer And Text "R17" \(27.1mm,16.875mm\) on Bottom Overlay \(Bottom Overlay\) to \(Bottom Solder\) clearance \(0.025mm\)](#)

[Silk To Solder Mask Clearance Constraint: \(0.096mm < 0.1mm\) Between Pad U2-11\(39.7mm,20.125mm\) on Bottom Layer And Track \(37.494mm,18.531mm\)\(41.558mm,18.531mm\) on Bottom Overlay \(Bottom Overlay\) to \(Bottom Solder\) clearance \(0.096mm\)](#)

[Silk To Solder Mask Clearance Constraint: \(0.097mm < 0.1mm\) Between Pad U2-11\(39.7mm,20.125mm\) on Bottom Layer And Track \(37.494mm,21.706mm\)\(41.558mm,21.706mm\) on Bottom Overlay \(Bottom Overlay\) to \(Bottom Solder\) clearance \(0.097mm\)](#)

[Back to top](#)**Silk to Silk (Clearance=0.1mm) (All),(All)**

[Silk To Silk Clearance Constraint: \(0.1mm < 0.1mm\) Between Text "12V" \(43.8mm,10.625mm\) on Top Overlay And Track \(43.8mm,11.487mm\)\(43.8mm,18.841mm\) on Top Overlay Silk Text to Silk Clearance \(0.1mm\)](#)

[Silk To Silk Clearance Constraint: \(0.062mm < 0.1mm\) Between Text "12V" \(43.8mm,10.625mm\) on Top Overlay And Track \(43.8mm,11.487mm\)\(49.579mm,11.487mm\) on Top Overlay Silk Text to Silk Clearance \(0.062mm\)](#)

[Silk To Silk Clearance Constraint: \(0.009mm < 0.1mm\) Between Text "C9a" \(25.075mm,14.975mm\) on Bottom Overlay And Text "C9b" \(27.05mm,14.975mm\) on Bottom Overlay Silk Text to Silk Clearance \(0.009mm\)](#)

Silk To Silk Clearance Constraint: (0.025mm < 0.1mm) Between Text "C9b" (27.05mm 14.975mm) on Bottom Overlay And Text "D3" (27.25mm 14.85mm) on Bottom Overlay. Silk Text to Silk Clearance (0.025mm)

[Back to top](#)



SIG61 - Smart Slave for Multiplex Powerline Network

This information is preliminary and may be changed without notice

1 GENERAL

The SIG61 is an independent slave in a Powerline communication network controlled by a SIG60 master device. The SIG61 has 4 identification (ID) pins, used to set the device address, 8 input pins and 8 output pins. The master can access any SIG61 device independently by using the proper device address. Data received from a remote SIG60 Master device is reflected to its output pins. The Master device can read the SIG61 input pins remotely. Its small footprint integrates most of the components needed for proper operation allowing small-size control solutions.

The SIG61 is an economical slave device for applications such as controlling motors, reading sensors etc., eliminating the need for dedicated control wires and a host controller for its operation. It helps reducing the harness size and increase reliability. The SIG61 has a sleep mode that enables power saving; special Wakeup messages on the AC or DC line are used to signal the sleeping devices to return to normal operation mode.

The SIG61 is useful for a wide range of Automotive, Avionics and Industrial applications such as sensor reading, actuator activation, doors, seats, mirrors, climate control, lights, Truck-Trailer, etc.



Figure 1.1 - SIG61 Application example

Applications

- Truck-Trailer sub-bus
- Door module
- Climate control network
- Front and back Lights
- Sensors Actuators network
- Entertainment control
- Renewal Energy / Battery management Micro-grid
- Security Monitoring

Features

- 7 selectable Carrier frequencies 1.75MHz - 13MHz
- Selectable bit rate between 9.6 Kbps to 115.2 Kbps.
- 8 output and 8 input pins
- Eliminates data wires and transceiver.
- Operates over wide range of noisy power supply / battery lines.
- Byte oriented communication.
- Sleep Mode for low power consumption.
- Allows Master - Slave multiplex networks
- Several independent networks can operate over the same wire using different carrier frequencies.
- Small footprint QFN 64 pin package

2 OVERVIEW

The SIG61 is an independent slave in a Master-Slaves network operating on a selected narrow band channel. A single SIG60 master controls all the slaves in a network; the slaves may be SIG61 devices as well as SIG60 devices operating as slaves.

Proper Network operation is maintained by employing 5 types of command messages: *Read*, *Read-Change*, *Write*, *Sleep* and *Change-Frequency*. The command format is similar to the standard Universal Asynchronous Receiver Transmitter (UART).

The SIG61 has internal narrow band modem, capable of operating in noisy environments. The receiver listens to the Powerline on its preset frequency. It filters out the signal from noise and interference and tries to recover the original command. If the checksum is correct, the SIG61 extracts the ID, Command and the Data.

If the received ID matches its own ID, the SIG61 proceeds to detect the received command. When a Write command is received, the data part of the command is directed to the corresponding 8 output pins. When a Read command is detected, the SIG61 responds by transmitting a dedicated message towards the master containing an image of its 8 input pins.

Multiple networks can operate concurrently on the same wire using different carrier frequencies.

2.1 Channels and Network

The SIG60-SIG61 network supports 16 combinations of frequency pairs. When set to such a pair, it is easy to switch from one frequency to the other when such need arises. Each channel accommodates a single SIG60 master and up to 15 SIG61/SIG60 slave devices. Additional SIG60-SIG61 networks can coexist on the same power line by employing different frequencies for each network, thus allowing different applications.

Channel frequencies: 1.75MHz, 4.5MHz, 5.5MHz, 6.0MHz, 6.5MHz, 10.5MHz and 13.0MHz.

Data transfer rate: 9.6Kbps up to 115.2Kbps.

Cable length: Dependent on external AC loads connected to the AC/DC Powerline.

2.2 The SIG61 Device

Figure 2.1 outlines the building blocks of the SIG61 device.

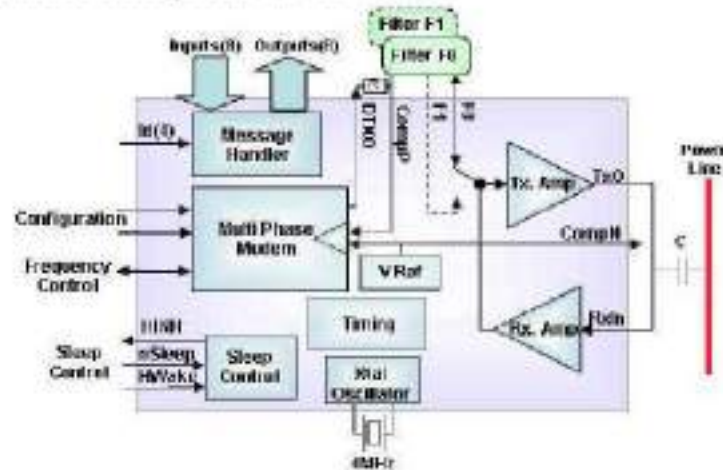


Figure 2.1 - SIG61 Logical Blocks

3 SIG61 SIGNALS

Device signals are described in table 3.1.



Figure 3.1 - SIG61 Pin-out

Pin name	Pin#	Pin type	Description
Control Signals			
HDO	35	Output 8mA	Digital data output signal. Output the received data from the powerline to the host.
INH	32	Output 8mA	Inhibit output for enabling the host or an external voltage regulator powering the host. This signal is HIGH in the normal and standby modes and LOW in sleep mode.
nSleep	31	Input	Sleep control input. Pulling this signal to LOW puts the SIG61 in sleep mode. Should be pulled to Vdd.
Wake	26	Input	Local wakeup input. Negative or positive edge triggered. This pin can be connected to an external switch in the application. When the pin is triggered the device will wake up and send a wake up message to all the devices on the network. When not in use, this pin should be pulled Up or Down.
nReset	29	Input, PU	Reset Input
InterHop	3	Input, PD	Allows automatic frequency hopping whenever an interference signal is detected on the power line. When HIGH, detection of interference switches the operating frequency between F0 and F1. If at the new frequency, no reception occurred for 2 sec, the operating frequency is switched back. For designs with a single channel this pin should be tied to ground.
Test	15	Input, PD	Should be connected to Gnd
MFInF0	23	Output 12mA	Output indicating the operating frequency. F1 when HIGH and F0 when LOW.
Line Interface signals			
OscO	52	Analog Output	Crystal Output

Dsdin	53	Analog Input	Crystal Input
RxN	9	Analog Output	The internal comparator negative pin. Its value is internally pulled to Vdd/2. Bypass RxN to Ground with a 1nF capacitor.
RxP	61	Analog Input	Positive pin input signal. Should be tied to RxN with a 1K Ohm resistor.
DTxO	45	Tristate/Output 2mA	Modulated digital transmit signal output to both ceramic filters.
RxIn	10	Analog Input	Receive input from the power line to the RX operational amplifier. This input is pulled internally to Vdd/2.
TxO	8	Analog Output	Transmit output.
F0B	59	Analog, Bi-directional	F0 External filter I/O. Its value is internally pulled to Vdd/2.
F1B	58	Analog, Bi-directional	F1 External filter I/O. Its value is internally pulled to Vdd/2.
TxOn	20	Output 12mA	HIGH when the device is transmitting a message.
RxOn	17	Output 12mA	HIGH when the device is in receive mode.
I/O Signals			
In0	47	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
In1	44	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
In2_HDI	42	Input	When SIG61 mode is enabled - The pin is read by the Master with a Read or Read-Change command in SIG61 mode. When SIG60 mode is enabled - This pin is HDI input and should be pulled up with 10K resistor. (See section 4.5)
In3_HDC	41	Input	When SIG61 mode is enabled - The pin is read by the Master with a Read or Read-Change command in SIG61 mode. When SIG60 mode is enabled - This pin is HDC input and should be pulled up with 10K resistor. (See section 4.5)
In4	40	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
In5	39	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
In6	37	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
In7	36	Input PD	The pin is read by the Master with a Read or Read-Change command in SIG61 mode only.
Out0	28	Output 8mA	Output of data bit 0 when the Write command received from Master in SIG61 mode only.
Out1	27	Output 8mA	Output of data bit 1 when the Write command received from Master in SIG61 mode only.
Out2	25	Output 8mA	Output of data bit 2 when the Write command received from Master in SIG61 mode only.
Out3	24	Output 8mA	Output of data bit 3 when the Write command received from Master in SIG61 mode only.
Out4	22	Output 8mA	Output of data bit 4 when the Write command received from Master in SIG61 mode only.
Out5	21	Output 8mA	Output of data bit 5 when the Write command received from Master in SIG61 mode only.
Out6	19	Output 8mA	Output of data bit 6 when the Write command received from Master.

Dat7	18	Output 8mA	Output of data bit 7 when the Write command received from Master in SIG61 mode only.
Configuration Signals			
Id0	16	Input PD	Bit 0 ID address in the network in SIG61 mode only.
Id1	14	Input PD	Bit 1 ID address in the network in SIG61 mode only.
Id2	13	Input PD	Bit 2 ID address in the network in SIG61 mode only.
Id3	12	Input PD	Bit 3 ID address in the network in SIG61 mode only.
F0F1-0	49	Input PD	Frequency selection pin in SIG61 mode only. See Table 4.2.
F0F1-1	50	Input PD	Frequency selection pin in SIG61 mode only. See Table 4.2.
F0F1-2	55	Input PD	Frequency selection pin in SIG61 mode only. See Table 4.2.
F0F1-3	56	Input PD	Frequency selection pin in SIG61 mode only. See Table 4.2.
BitRate0	1	Input PD	Bit rate selection pin in SIG61 mode only. See Table 4.3.
BitRate1	2	Input PD	Bit rate selection pin in SIG61 mode only. See Table 4.3.
NC	5	—	Not connected.
Mode2	48	Input PD	Should be left unconnected.
Mode3	62	Input PD	For SIG61 mode this pin should be connected to Vdd. For SIG60 mode should be left unconnected (See section 4.5).
Mode4	63	Input PD	For SIG61 mode this pin should be connected to Vdd. For SIG60 mode should be left unconnected (See section 4.5).
AutoFreqCh	64	Input PD	Automatic Frequency Change in SIG61 mode only. When HIGH, the device automatically switches frequency after about 4 seconds without bus activity.
AutoSleep	30	Input PD	When this pin is set to HIGH, the device will automatically enter sleep mode after about 8 seconds without bus activity.
F1nF0	38	Input PD	Selects between F0 / F1, HIGH – F1, LOW – F0.
Power signals			
Vdd	4,11, 34,43, 57	Power	3.3V power supply.
GND	6,7,33 46,51	Power	Ground
AGnd	54	Power	Analog Ground
AVdd	60	Power	3.3V Analog Power. Separate from Vdd with a 6.8 to 10 Ohm resistor and bypass to Ground with 1nF and 10nF capacitor.
Exp	Exp		May be connected to GND

PD – Pull down resistor 100K ohm $\pm 1\%$

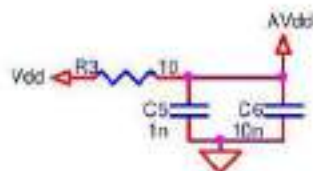
PV – Pull up resistor 100K ohm $\pm 1\%$

Table 3.1 - Device signals

3.1 Power Signals

Vdd and Gnd layout traces should be as wide as possible. It is recommended to connect a 0.1 μ F capacitor between each Vdd and ground pins, as close as possible to the pins.

Analog Vdd pin, AVdd, should be connected to Vdd. AGnd should be connected to ground. The Analog supply has to be sufficiently powerful (capable of current driving), to avoid any fluctuations of supply voltage level. It is recommended to keep the lines connecting the 3.3V power supply to Vdd pins as short as possible with wide PCB traces.



Notes:

- All input the signals should be tied either to Gnd. or Vdd, according to the desired mode of operation.
- F0 and F1 are ceramic filters
- Adjust R1 for maximal output level without distortion on powerline.
- C1 and C2 values depend on crystal used. Usually values are between 0pF (NC) to 1.5pF.
- Optional 3MHz HPF for very noisy channels, replace CR with CRa, C11, OR and L3
- F1 is an optional communication channel

3.2 Ceramic Filter

3.2.1 Ceramic Filter Considerations

The SIG61 is designed to operate with one ceramic filter for transmission and reception. However, if switching between two channels is desired, two ceramic filters are required. The minimum allowable bandwidth of the ceramic filters is ± 60 kHz @ 3dB. Narrow bandwidth limits the maximal bit rate. The SIG61 selectable frequencies meet market available ceramic filters. It is important to select the widest bandwidth available. 1.75MHz, 10.5MHz and 13MHz may use discrete filters. Through hole ceramic filters are available from Oscilent.

Nominal freq.	3 db BW	20db BW	Insertion loss	Stop band attenuation	In/Out imped.	Oscilent part #
MHz	kHz min.	kHz max.	dB max.	dB min.	Ohm	
*1.75	± 70	750	6.0		330-1000	discrete filters
4.50	± 70	750	6.0	30	1000	773-0045
5.50	± 60	750	6.0	30	600	773-0065
6.00	± 60	750	6.0	30	470	773-0060
6.50	± 60	800	6.0	30	470	773-0065
**10.50	± 150	1500	4.5		330	discrete filters
**13.00	280 ± 50	1500	4.5		330	discrete filters

* 1.75MHz can operate only at 9.6Kbps

** 10.5MHz and 13.00MHz operates at 115.2Kbps instead of 9.6Kbps.

SMD ceramic filters are available from Murata.

Nominal freq.	3 db BW	20db BW	Insertion loss	Stop band attenuation	In/Out imped.	Murata part #
MHz	kHz min.	kHz max.	dB max.	dB min.	Ohm	
4.50	± 60	600	6.0	26	1000	SFSK44M50CF00-R3
5.50	± 60	600	6.0	25	600	SFSK45M50CF00-R3
6.00	± 60	600	6.0	25	470	SFSK46M50CF00-R3
6.50	± 60	600	6.0	25	470	SFSK46M50CF00-R3

3.3 Oscillator

The SIG61 is designed to operate with a low cost 4MHz crystal connected between OscIn and OscOut pins. Each of these pins should be connected to the ground via a capacitor. All the corresponding PCB traces should be as short as possible.

Recommended crystals are:

1. NDK AT-51 GW.
2. Epson MA-505.

The values of C1, and C2 in Figure 3.3, pertaining to the oscillator circuitry, should be determined according to the crystal manufacturer recommendations. Values between 0pF and 1.5pF may serve as good starting point.

The overall frequency tolerance should not exceed 200ppm.

3.4 Communication performance

The maximum cable length between two devices depends mainly on the AC impedance of loads connected to that line and number of nodes. The Powerline cable length has less effect on communication. The SIG61 requires a minimum received signal of 20mVpp for proper reception.

4 DEVICE OPERATION

4.1 Protocol

The device operation is controlled via 5 types of commands:

Write command - Upon receiving a *Write* command with the SIG61 specific ID, the device shall output the data byte content as indicated by the command to its Output pins.

Read command - Upon receiving a *Read* message with the SIG61 specific ID, the device shall respond by sending a message containing the status of its Input pins (followed by an appropriate checksum).

Read-Change command - When receiving a *Read-Change* message with the SIG61 specific ID, the device shall respond to the command by indicating if a pulse upon detecting the first change on its input pins. The response message shall contain the new status of the input pins followed by an appropriate checksum.

Sleep command - Upon receiving a *Sleep* command, the device shall enter a low power-consumption (sleep) mode. A wakeup message generated by the master, or by any of the slaves, wakes up all the devices on the network. This is a global message targeting all the slaves in the network.

Change-Frequency command - Upon receiving *Change-Frequency* command, the device shall switch from its current operational frequency to the other. This command is a global command targeting all the slaves in the network.

4.1.1 Command structure

The structure of the five types of commands is detailed below.

Command type 1: Write command.

The *Write* command consists of 5 bytes: sync break, sync field, identifier, data and checksum. Upon receiving a write command, if checksum and protection bits calculations are successful, the data byte content is transferred to the corresponding output pins.

Sync break: Sync break length is at least 13 bit times with compliance to Lin protocol.

Identifier byte: Consist of 2 protection bits (MSB), "01" bits and 4 ID bits - (P1, P0, '0', '1', ID [3:0])

Protection bits calculation:

$$P0 = (\text{Identifier}[0]) \text{ XOR } (\text{Identifier}[1]) \text{ XOR } (\text{Identifier}[2]) \text{ XOR } (\text{Identifier}[4])$$
$$P1 = \sim ((\text{Identifier}[1]) \text{ XOR } (\text{Identifier}[3]) \text{ XOR } (\text{Identifier}[4]) \text{ XOR } (\text{Identifier}[5]))$$

Checksum calculation:

The checksum is an inverted 8 bit sum of the Identifier and Data byte including (own) carry:
 $\text{Checksum} = \sim (\text{Identifier Byte} + \text{Data Byte} + \text{Carry})$

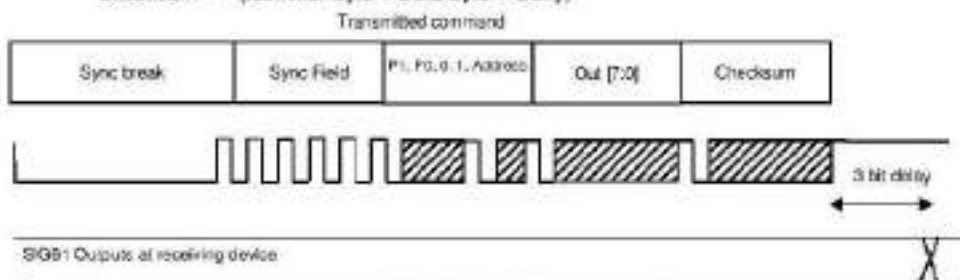


Figure 4.1 - Write command

Sending Write command of Data byte 0xFF to a device with ID '4' will be calculated the following way:

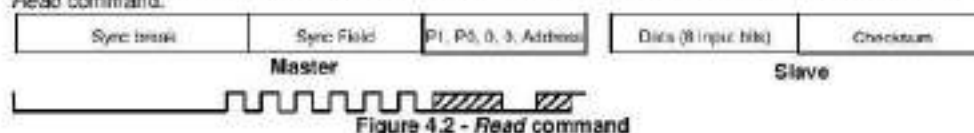
P0 = ('0' XOR '0' XOR '1' XOR '1') = '0'
 P1 = ('0' XOR '0' XOR '1' XOR '0') = '0'
 Identifier Byte = '00'-'01'-'0100' = 0x14
 Checksum = ~ (0x14 + 0xFF + (Carry = '1')) = 0xEB

Command type 2: Read command.

The Read command, initiated by the Master, requests the status of the SIG81 input pins.

The **Read** command from the master consists of 3 bytes: sync break, sync field and identifier. Identifier byte: consist of 2 protection bits (MSB), "00" bits and 4 ID bits - [P1, P0, '0', '0', ID [3:0]]. The two protection bits are calculated as described above.

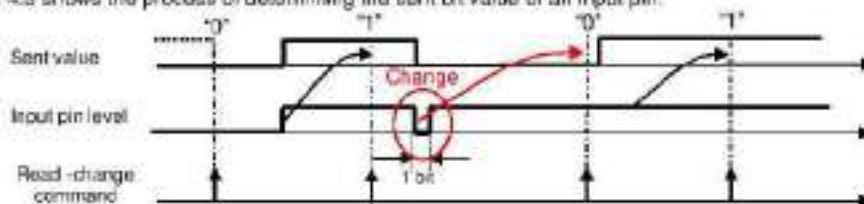
If header detection is correct (including protection bits), the SIG81 device (whose ID matches the one in the command) shall respond by sending two bytes. A data byte containing the status of its eight input signal pins followed by a checksum byte. The checksum calculation is carried out as in the description above while part of it, the identifier byte, was transmitted by the Master. Figure 4.2 shows a generic Read command.



Command type 3: Read-change

The *Read-change* command is similar to the *Read* command. However, it enables to detect any *change* in the input pins (pulse-like behavior) that may have occurred between two consecutive *Read* or *Read-change* commands. The command from the master instructs the SIGBT to send back information on changes of its input pins since the last *Read* or *Read-change* command. The first change on the pin after the *Read* command sets its bit value.

Figure 4.3 shows the process of determining the sent bit value of an input pin.



The Read-change command from the master has 3 bytes: sync break, sync field and identifier. Identifier byte: consist of 2 protection bits (MSB), "10" bits and 4 ID bits - (P1, P0, '1', '0', ID [3:0]). The response for this command is a data byte containing the SIG61's input pins recent changed value, followed by the checksum byte. See checksum and protection bits calculation description above. Figure 4.4 shows a Read-changes command.

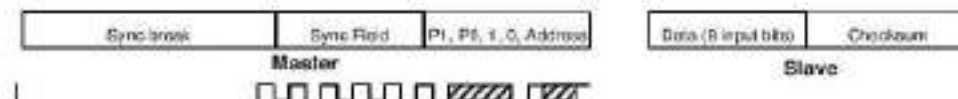


Figure 4.4 - Read-change command

Command type 4: Sleep command.

This type of command consist of 5 bytes: sync break, sync field, "3C" Hex, "00" Hex and checksum. The sleep command identifier is "3C" Hex as in LIN2.0 specifications and the following data byte "00" Hex. Upon reception of sync break, sync field, "3C" Hex and "00" Hex bytes, a device enters sleep mode immediately and as a result it's the following command bytes are ignored.

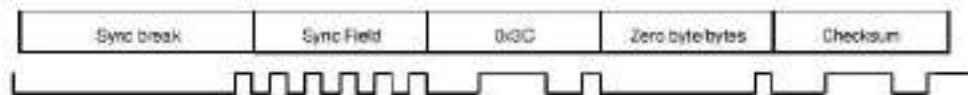


Figure 4.5 - Sleep command

Command type 5: Change Frequency command.

This type of command consists of 5 bytes: sync break, sync field, "FE" Hex, "00" Hex and checksum. The change frequency command identifier is "FE" Hex and the following data byte is "00" Hex. Upon reception of sync break, sync field, "FE" Hex and "00" Hex bytes the frequency changes from F1 to F0, or vice versa. Checksum calculation follows the description above.

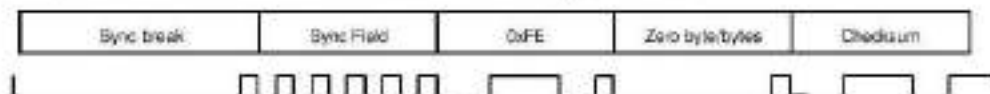


Figure 4.6 - Frequency Change command

4.2 Power Management

The SIG61 device features Sleep mode for power saving. Entering the Sleep mode, as well as waking up, can be initiated locally, by means of dedicated input pins, or remotely through activity (proper messages) over the bus.

4.2.1 Entering Sleep mode

The SIG61 can enter sleep mode by any of the following ways:

1. The device nSleep pin is lowered.
2. Sleep command from a remote master is received.
3. The AutoSleep pin is set HIGH and no reception occurred for about 8 seconds.

4.2.2 Device Outputs during Sleep

During Sleep mode, the SIG61 8 output pins remain unchanged if the device has entered this mode due to a Sleep command from the Master or by lowering the Pin nSleep. However if the device entered the Sleep mode due to the AutoSleep function, SIG61 will lower all the outputs to "0".

4.2.3 Remote wake up process

The SIG61 can be awakened by a remote SIG60 master, or SIG61 slave, device transmitting a wakeup message over the bus. During Sleep Mode, the SIG61 wakes up periodically, every 32mSec, to sense for bus activity. If a wakeup message is detected, the SIG61 device raises pin INH and lowers pin HDO. If nSleep pin is low upon remote waking up, the local host (a device controlling the SIG61) which initially pulled nSleep pin down, must raise the nSleep back high. Figure 4.7 provides the signal description.

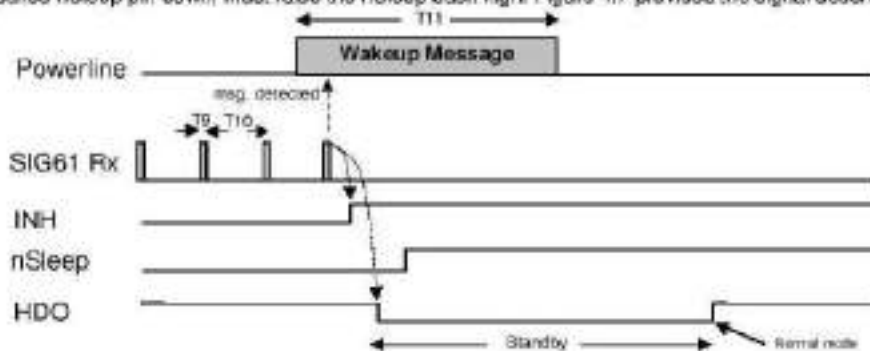


Figure 4.7 - Wakeup from bus message

4.2.4 Wakeup from Wake pin

A transition seen on the Wake pin (caused by an external source) is used to wake up the device. The device then enters Standby mode, it rises pin INH, and transmits a wakeup message to the bus. While transmitting the wakeup message, the device lowers pin HDO. After the transmission is completed the device raises pin HDO. This is depicted in Figure 4.8. After the transmission is completed the device enters Normal mode. If nSleep pin is low upon waking up, the local host which initially pulled nSleep pin down, must raise the nSleep back to high.

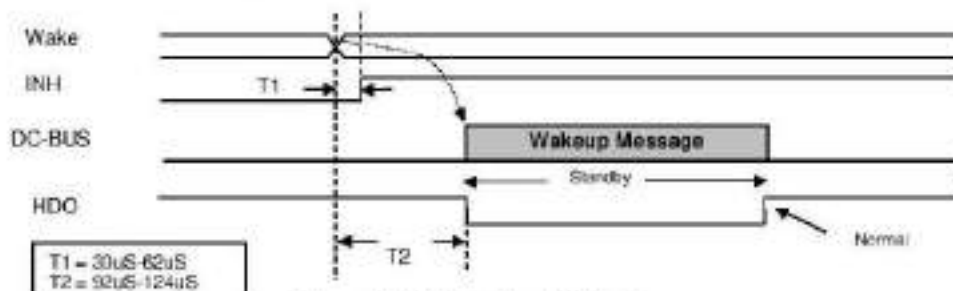


Figure 4.8 - Wakeup from Wake pin

4.3 Timing Characteristic

Symbol	Figure	Characteristics	Min	Max
T ₁	4.3, 4.4	Drop of HDC to drop of HDI	200nS	
T ₂	4.3	Raise of HDC to drop of HDI	200nS	
T ₃	4.3	Command stop bit to raise of HDC	Half bit rate	
T ₄		Drop of nSleep to drop of INH	30uS	78uS
T ₅	4.5	Raise of nSleep to raise of INH	30uS	78uS
T ₆	4.4	Transact on Wake to raise of INH	30uS	78uS

T ₁	4.3	Raise of nSleep to drop of HDO	92uS	3.5mS
T ₂	4.4	Transact on Wake to drop of HDO	92uS	3.5mS
		Power Up or Reset to Normal mode		6mS
T ₉	4.5	Receive duration during sleep mode	1.5mS	
T ₁₀	4.5	Receive period during sleep mode	32mS	+/-20%
T ₁₁	4.5	Wakeup message		150mS

4.4 SIG61 Configuration

The Pins labeled "Mode 4" and "Mode 3" should be tied up to Vdd to enabled SIG61 mode.

The SIG61 operates at default with the following parameters:

Bit rate: 19.2Kbps, F0=5.5MHz, F1=6.5MHz

The following configuration pins set the operating frequencies and bit-rate selection according to tables 4.2 and 4.3.

Changing these pins will only takes affect when the SIG61 device is set to SIG61 mode (i.e. Mode 3 and 4 is connected to Vdd).

SIG61 device configuration will be updated according to current pins status (last change counts), and maintains its configuration in case the SIG61 is switched to SIG60 mode, until next power-cycle/hard reset events.

Table 4.2 – F0, F1 select

F0F1 (3:0) pins	F0	F1
1111	1.75MHz	4.5MHz
1110	1.75MHz	5.5MHz
1101	1.75MHz	6MHz
1100	1.75MHz	6.5MHz
1011	4.5MHz	5.5MHz
1010	4.5MHz	6.0MHz
1001	4.5MHz	6.5MHz
1000	4.5MHz	10.5MHz
0111	10.5MHz	13.0MHz
0110	5.5MHz	10.5MHz
0101	5.5MHz	13.0MHz
0100	6.0MHz	10.5MHz
0011	6.0MHz	13.0MHz
0010	6.5MHz	10.5MHz
0001	6.5MHz	13.0MHz
0000	5.5MHz	6.5MHz

Table 4.3 – bit rates selection

BitRate (1,0) Pins	11	10	01	00
Frequency				
1.75 MHz	-	-	9.6K	19.2K
4.50 MHz	38.4K	57.6K	9.6K	19.2K
5.50 MHz	38.4K	57.6K	9.6K	19.2K
6.00 MHz	38.4K	57.6K	9.6K	19.2K
6.50 MHz	38.4K	57.6K	9.6K	19.2K
10.50 MHz	38.4K	57.6K	115.2	19.2K
13.00 MHz	38.4K	57.6K	115.2	19.2K

Signal	High ("1")	Low ("0")
AutoSleep	Auto Sleep On	Auto Sleep Off
ID[3:0]	Defines device ID	
F1nF0	Select F1	Select F0
InterHop	InterHop On	InterHop Off

4.5 Operating the SIG61 as SIG60

User can use the SIG61 device to be function as a SIG60 device.
The SIG60/61 modes is determined by mode3 and mode 4 pins.

When mode3 and mode4 pins are connected to Vtd, the SIG61 device is set to SIG61 mode, acting as a Slave device with an extended 8 input and 8 outputs.

When mode3 or mode4 pins are left unconnected, the SIG61 device enters SIG60 mode, acting as a SIG60 powerline transceiver.

In SIG61 mode, Pins In2_HDI and In3_HDC acts as inputs pins as part of the total dedicate 8 input pins. Please note that these pins are floated and may be pulled up or down according to user requirement.

In SIG60 mode, Pins In2_HDI and In3_HDC act as SIG60 HDI and HDC pins respectively. In this mode, these pins should be pulled-up with at least 10k ohm resistor.

In SIG61 mode, the SIG61 devices is configured using its dedicated hardware pins (See section 4.4).

In SIG60 mode, the SIG61 device is configured through the data command mode as in SIG60 device.

User may firstly set SIG61 mode, then set configuration using the specific hardware pins, and then switch to SIG60 mode. The configuration will remain until next power-cycle/hard reset events and saves configuration through the data command procedure.

5 ELECTRICAL PARAMETERS

5.1 Absolute Maximum Rating

Ambient Temperature under bias	-40°C to 125°C
Storage Temperature	-55°C to 150°C
Input Voltage	-0.6V to Vdd+0.3V
Vdd Supply voltage	-0.3V to 4V

5.2 Electrical Operating Conditions

Symbol	Characteristics	Min	Typ	Max	Units	Conditions
Vdd	Supply Voltage	3.0	3.3	3.6	V	
Idd	Supply Current		40		mA	5.5MHz
Idd	Supply Current during Tx		50		mA	5.5MHz
Ipd	Supply Current in Sleep mode		260		μA	

5.3 DC Electrical Characteristics

Symbol	Characteristics	Vdd	Typ	Units	Conditions
Vih	Minimum high level input voltage	3.0	2.1	V	
Vil	Maximum low level input voltage	3.0	0.9	V	
VoH	Minimum high level output voltage	3.0	2.4	V	
VOL	Maximum low level output voltage	3.0	0.4	V	
Iout	Maximum output current, other pins				See pins table
Iin	Maximum input current	3.3	+10	μA	

5.4 Operating Temperature

Commercial:	0°C to 70°C
Industrial:	-40°C to 85°C

5.5 Mechanical Information

Package type - QFN 64 pin

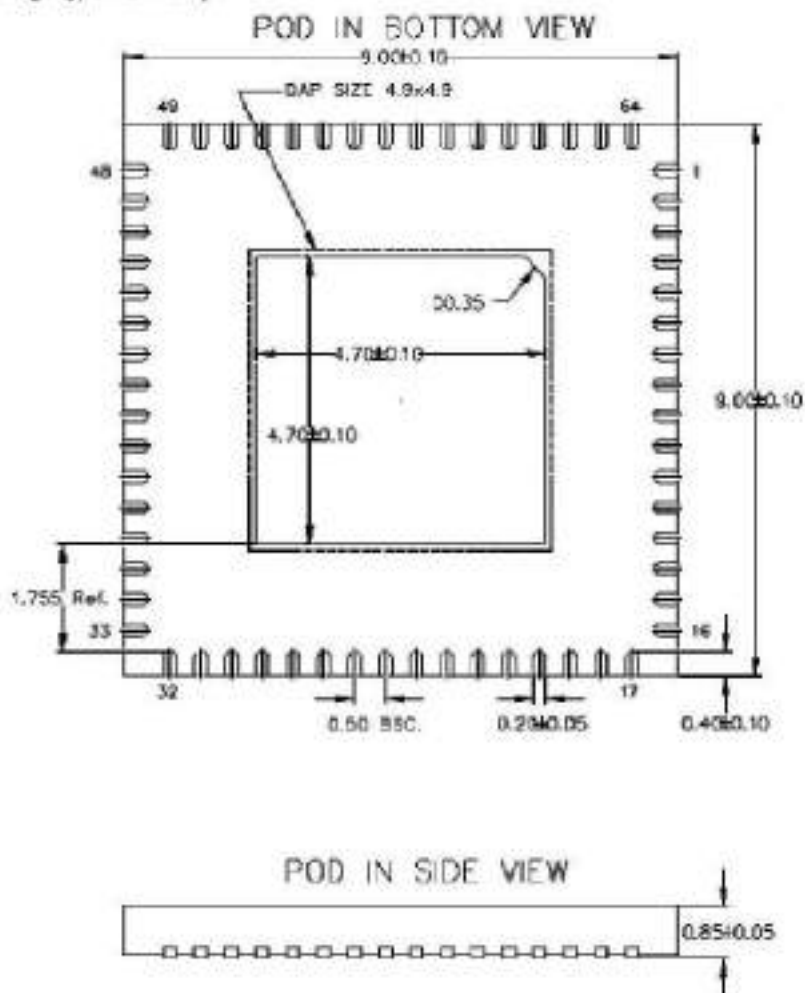


Figure 5.1 – QFN64 mechanical dimensions

Revision changes:

Revision	Date	Comments
0.2	2.3.2008	Detailed description
0.4	22.5.2009	Updated schematic
0.5	25.5.2009	Updated pin information
0.61	22.7.2009	Updated mechanical drawing Added "Outputs during Sleep"
0.7	13.9.2009	Added checksum and parity calculations Renumbering figures
0.8	20.11.2009	Update the Write and Read commands. Update tables 4.2- 4.3 Added 3.4.8 and 3.4.9
0.9	20.12.2009	Updated schematics, protection network, Electrical parameters, pin description table
0.92	28.12.2009	Updated drawings, notations, descriptions.
0.93	7.1.2010	Updated Read-change command explanation
0.932	6.5.2010	Updated drawings, descriptions.
0.94	29.8.2010	Updated ceramic filters, schematics.
0.95	24.10.2010	Updated figures 4.1-4.5
0.96	27.2.2011	Updated figure 3.3 schematic for high noise channels operation
0.961	4.8.2011	Updated figure 3.3 C9 and C11 to 220p
0.97	1.11.2011	Update schema - Connect pins Mode3 and Mode4 to Vdd. Update Table 3.1 - Replace between the GND pins and the Vdd pins, Fix a typo at mode4 pin number.
0.98	30.4.2012	Update figures 4.1 ,descriptions, examples
1.01	30.9.2015	Update Sleep current consumption
1.02	2.1.2017	R2=470 ohm
1.03	6.1.2017	Update section 4.3
1.04	25.4.2017	Updated table 3.1 and added section 4.5
1.05	1.5.2017	Update sections 3, 4.4 and 4.5
1.06	30.12.2018	Formating