

**MISKOLCI EGYETEM**  
**GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR**



**Automatizálási és Infokommunikációs Intézet**

**Villamosmérnöki (BSc) alapszak**

**FEJLESZTŐPANEL KÉSZÍTÉSE STM32  
MIKROVEZÉRLŐHÖZ**

**Szakdolgozat**

**PÖSTÉNYI LEVENTE**

**BQ4L6J**

**Miskolc, 2024**



## **SZAKDOLGOZAT FELADAT**

***Pöstényi Levente***

BSc Villamosmérnök jelölt részére

A tervezés tárgyköre:

A feladat címe: *Fejlesztőpanel készítése STM32 mikrovezérlőhöz*

A feladat részletezése:

1. Mutassa be a CAN kommunikáció működését.
2. Mutassa be az iso SPI kommunikáció működését.
3. Válasszon mikrovezérlőt a fent említett feladatok elvégzéséhez.
4. Végezzen irodalom és piackutatást a mikrovezérlő kiválasztásához.
5. Tervezzen egy áramkört ami alkalmas CAN, CANFD és iso SPI kommunikáció megvalósítására és ezen kommunikációk

Tervezésvezető(k): **Koba Máté**

Konzulens(ek):

A szakdolgozat kiadásának időpontja: 2024.11.23.

A szakdolgozat beadásának határideje: 2024.11.15.

Miskolc, 2024.11.23.

**Dr. Trohák Attila**  
egyetemi docens  
Intézet igazgató

# 1 Tartalomjegyzék

<b>1</b>	<b>TARTALOMJEGYZÉK .....</b>	<b>1</b>
<b>2</b>	<b>BEVEZETÉS .....</b>	<b>3</b>
<b>3</b>	<b>FORMULA STUDENT .....</b>	<b>4</b>
3.1	SZABÁLY RENDSZER .....	6
3.2	ELECTRIC RACING MISKOLC .....	7
3.2.1	<i>erm.e01 autó .....</i>	<i>7</i>
<b>4</b>	<b>MODERN AUTÓELEKTRONIKA .....</b>	<b>10</b>
4.1	CONTROL AREA NETWORK (CAN).....	10
4.1.1	<i>Busz topológia.....</i>	<i>11</i>
4.1.2	<i>Standard CAN.....</i>	<i>11</i>
4.1.3	<i>Extended CAN.....</i>	<i>12</i>
4.1.4	<i>Az arbitráció .....</i>	<i>13</i>
4.1.5	<i>CAN üzenet típusok.....</i>	<i>14</i>
4.1.6	<i>A CAN fizikai felépítése és tulajdonságai.....</i>	<i>15</i>
4.1.7	<i>A CAN FD.....</i>	<i>16</i>
4.2	LOCAL INTERCONNECT NETWORK (LIN).....	17
4.2.1	<i>A LIN üzenetkerete .....</i>	<i>18</i>
4.2.2	<i>A LIN üzenet fajtái .....</i>	<i>19</i>
4.2.3	<i>A LIN fizika felépítése .....</i>	<i>19</i>
<b>5</b>	<b>BEÁGYAZOTT RENDSZEREK MIKROVEZÉRLŐK .....</b>	<b>21</b>
5.1	MIKROVEZÉRLŐK .....	21
5.1.1	<i>Felépítésük.....</i>	<i>21</i>
5.1.2	<i>Működés.....</i>	<i>22</i>
5.2	SERIAL PERIPHERAL INTERFACE (SPI).....	22
5.2.1	<i>SPI adat továbbítás .....</i>	<i>23</i>
5.2.2	<i>SPI fizikai felépítése.....</i>	<i>24</i>
5.2.3	<i>ISO SPI .....</i>	<i>25</i>
5.3	RS-485.....	27
5.3.1	<i>Az RS-485 tulajdonságai:.....</i>	<i>27</i>
5.3.2	<i>Az RS-485 fizikai felépítése .....</i>	<i>27</i>
5.3.3	<i>Az RS-485 összekötési módjai .....</i>	<i>28</i>
5.3.4	<i>UART (Universal Asynchronous Receiver/Transmitter) .....</i>	<i>28</i>
<b>6</b>	<b>A MIKROVEZÉRLŐ KIVÁLASZTÁSA .....</b>	<b>30</b>
6.1	AZ STM32H563RIT MIKROVEZÉRLŐ PARAMÉTEREI: .....	30

6.2	A CSATLAKOZÓK KIOSZTÁSA A MIKROVEZÉRLŐN .....	30
<b>7</b>	<b>AZ ÁRAMKÖR MEGTERVEZÉSE.....</b>	<b>32</b>
7.1	ALTIUM DESIGNER .....	32
7.1.1	<i>Schematic tervezés .....</i>	<i>32</i>
7.1.2	<i>PCB (Nyomtatott áramkör) tervezés .....</i>	<i>32</i>
7.1.3	<i>BOM (Bill of Materials).....</i>	<i>33</i>
7.2	AZ ÁRAMKÖR RÉSZEI ÉS FUNKCIÓI .....	33
7.2.1	<i>Az USB-C csatlakozó .....</i>	<i>34</i>
7.2.2	<i>Az LDO (LD1117) .....</i>	<i>34</i>
7.2.3	<i>A mikrovezérlő és ennek tápellátása .....</i>	<i>35</i>
7.2.4	<i>Az ISO SPI adóvevője .....</i>	<i>36</i>
7.2.5	<i>Az RS-485 adóvevője.....</i>	<i>37</i>
7.2.6	<i>A CANFD adóvevője.....</i>	<i>38</i>
7.3	AZ ÁRAMKÖR FIZIKAI FELÉPÍTÉSE .....	40
<b>8</b>	<b>ÖSSZEGZÉS .....</b>	<b>41</b>
<b>9</b>	<b>IDEGEN NYELVŰ ÖSSZEGZÉS .....</b>	<b>42</b>
<b>10</b>	<b>IRODALOMJEGYZÉK.....</b>	<b>43</b>
<b>11</b>	<b>ÁBRAJEGYZÉK .....</b>	<b>45</b>
<b>12</b>	<b>MELLÉKLETEK.....</b>	<b>47</b>

## 2 Bevezetés

A szakdolgozatom témája egy fejlesztőpanel tervezése, amelyet az Electric Racing Miskolc számára készítek. Az Electric Racing Miskolc egy formula student csapat, amely az évente megrendezendő formula student versenyeken vesz részt. Évente körülbelül 10 verseny van különböző országok szervezésével. A versenyekre a csapatnak egy versenyautót kell építenie, amit utána többféle versenyszámban kell megmértenie. A verseny alatt a pontozás kiter többek között a versenyautó hatékonyságára, a mérnöki tervezésekre és az autó elkészítésének gazdasági részére is. Az autó rengeteg alkatrészből áll és ezeket az alkatrészeket külön-külön kell tesztelni az autóra építés előtt. Ezeknek a tesztelésére szükséges egy fejlesztőkártya tervezése, amely képes az autó legtöbb részének a tesztelésére, továbbá alkalmas a jövőben az új tagok oktatására is. A tesztelés szempontjából a fejlesztőkártyára kerülnie kell Control Area Network (CAN) kommunikációnak, ugyanis az autó legtöbb része ezen a protokollon keresztül kommunikál. A fejlesztő kártya fontos részét képezi, még a Serial Peripheral Interface (SPI), ami egy soros kommunikáció és ezzel kommunikál az autó akkumulátor vezérlője az akkumulátorral. Szükséges továbbá az RS-485 kommunikáció, amely a kormány elektronikájában található. A fejlesztőkártyára fognak kerülni közvetlen csatlakozások a mikrovezérlő lábaihoz, ezekhez később majd tetszőlegesen lehet hozzákötni eszközöket. Ezek alapján ki kell választani a megfelelő mikrovezérlőt és a kommunikációkhoz szükséges áramköröket. Majd ezeket el kell helyezni egy nyomtatott áramkörön és minden csatlakozót hozzákötni a megfelelő lábához. A nyomtatott áramkör tervezésére az Altium Designer nevű szoftvert fogom használni, ami az egyik legelterjedtebb nyomtatott áramkör tervező program rengeteg kiegészítő funkcióval.

Szakdolgozatomban be fogom mutatni, a kommunikációkat, amelyekkel dolgozni fogok, továbbá a fejlesztő kártya megtervezésének folyamatát, ami magában foglalja a kártya áramellátásának kidolgozását, a kommunikációk megfelelő felépítését és bekötését, illetve ezek összehangolását a mikrovezérlővel.

### 3 Formula Student

A verseny célja egy formula stílusú egy üléses autó megépítése, amivel a diákok a világ minden tájáról érkező másik csapatokkal versenyezhetnek. A versenynek rengeteg versenyszáma van, az összes területről, ami egy autó létrehozásához szükséges. A sok különböző versenyszám, miatt nem elég az autónak a leggyorsabbnak lennie, hanem minden területen jól kell teljesítenie. Ilyen területek például: az autó felépítése, annak teljesítménye és a pénzügyi háttere. A formula student egy extra lehetőséget és egyben kihívást ad a diákoknak, hogy az egyetem mellett tapasztalatokat szerezzenek az autóiiparból, ahol nem csak a fejlesztésekre, hanem a fenntarthatóságra is figyelmet kell fordítaniuk. Az elkészült autónak nagyon jó vezetési karakterisztikákkal kell rendelkeznie, mint például: gyorsulás, kezelhetőség és fékezés. Ezek mellett megbízhatónak kell lennie egy reális ár mellett. Az autókat szakmabeli zsűri értékeli, akik az autósport, autóiipar vagy ezeket ellátó ipari környezetből érkeznek.[1]

Az versenyre felkészülés rengeteg tapasztalatot nyújt a diákoknak csapatmunka, idő és projekt menedzsment terén, továbbá betekintést ad abba, hogy az autó iparban hogyan zajlanak az építési, gyártási és pénzügyi folyamatok. A formula student a cégek részére is nagy előnyt jelent hiszen az résztvevő diákok, naprakész tapasztalattal kezdenek az adott cégeknél. A versenyeken robbanómotoros és elektromos hajtással egyaránt részt lehet venni. A verseny kettő nagy részre van osztva, ami közül az egyik a statikus versenyszámok, a másik a dinamikus versenyszámok. Közöttük a különbség az, hogy az autó használva van-e a versenyszámban vagy sem.[1]

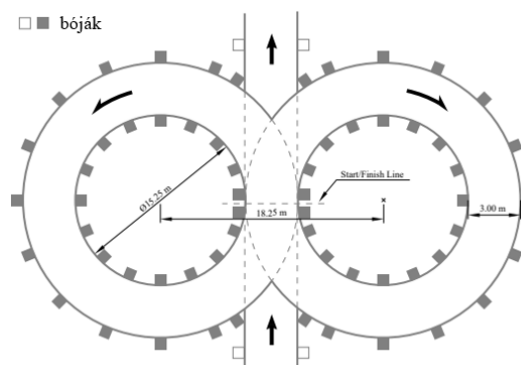
#### Statikus versenyszámok:

- Engineering Design Event: A versenyszám elején a diákoknak egy nyolc oldalas technikai dokumentációt kell benyújtaniuk az autóról. Amennyiben valaki önvezető autót tervezett, akkor az önvezető rendszerről egy maximum öt oldalas beszámolót kell beadnia. A dokumentumoknak tartalmaznia kell a mérnöki tervet és hogy ez a terv, hogyan lett megvalósítva az autóban. A zsűri ellenőrizni fogják utána ezeknek a terveknek a megvalósítását az autón. Ezek után a zsűri kérdéseket fog intézni a csapat tagjaihoz, a terveket és kivitelekkel illetően, ezzel is tesztelve a diákok tudását az autójukról és a mögöttes lévő tudásról. Ezzel a zsűri nem csak a minőségéről győződhet meg a technikai megvalósításnak, hanem arról is, hogy milyen okok állnak a terv mögött.[2]

- Cost and Manufacturing: Az ár egy meghatározó tényező minden termék esetében. Ebben a versenyszámban a diákoknak írniuk kell egy költség beszámolót, ami tartalmazza az autófelépítésének költségeit, beleértve az alkatrészeket és gyártási költségeket is. Ezek után a zsűri még a költségekkel kapcsolatos kérdéseket tesz fel a diákoknak.[2]
- Business Presentation: Ezen a versenyszámon a hallgatóknak egy vállalkozási tervet kell bemutatniuk, egy fiktív cégnek, amit a zsűri képvisel. A tíz perces előadás alatt be kell mutatni miért ez a legjobb megoldás a követelményekre és, hogy ezt hogyan lehetne értékesíteni. A prezentációt egy öt perces beszélgetés követi. A pontozás figyelembe veszi a prezentációt annak felépítését és prezentálását, valamint a kérdésekre, adott válaszokat.[2]

#### Dinamikus versenyszámok:

- Acceleration: A gyorsulási teszt egy 75 méteres egyenes pályán van rendezve, ahonnan az autók állóhelyzetből indulnak. A megfelelő motorterv elengedhetetlen, ahhoz, hogy az autóból a lehető legtöbb nyomatékot tudjuk kihozni. A gyorsuláson kívül a versenyszám teszteli az autó tapadását is, ugyanis ez elengedhetetlen a pályán maradáshoz. A legjobb autók négy másodperc alatt képesek, ezt a távot teljesíteni és a végére a sebességük 100 km/h is nagyobb lehet.[2]
- Skid Pad: Ebben a versenyszámban az 1. ábrán látható nyolcas alakú pályán kell kettő kört menni, minden esetben a második kör ideje van mérve. A mért idő képet ad arról, hogy az autó milyen oldalirányú gyorsulásra képes. Fontos szerepet játszik az egészben az aerodinamika, ami növeli az autó leszorító erejét. A versenyszámban mindenképpen a bóják között kell maradni, ezek felborítása időbeli büntetéssel jár. [2]



1. ábra: Skid Pad versenyszám pályája[3]

- Autocross: Ez a versenyszám egy 1 kilométer hosszú versenypályát foglal magába, amelyen a lehető leggyorsabban kell végig menni. Az autocross egy összesített képet ad az autó versenyképességeiről, és ez dönti el a későbbi endurance számba az indulási pozíciót. Itt szintén idő büntetés jár a felborított bójákért.[2]
- Endurance: A legtöbb pontot érő és egyben a fő versenyszám, ami egy 22 kilométeres pályán zajlik. Az autóknek itt a hosszútávú tűrőképességét tesztelik. A versenyszám tapasztalt pilótákat igényel, ugyanis a pályán egyszerre négy versenyautó tartózkodhat.[2]
- Efficiency: A versenyszámok alatt az üzemanyag és energia fogyasztás pontosan monitorozva van. A hatékonyság azonban nem csak a teljes elfogyasztott üzemanyagtól vagy energiától függ, hanem attól is, hogy mellette, milyen sebességgel haladtak, az autók.[2]

### 3.1 Szabály rendszer

Az autót szigorú szabályok alapján kell megtervezni és építeni, amit minden évben változtatnak. A 2025-ös szabályzat már elérhető, ez jelenleg 133 oldalas. A szabályok kitérnek külön a robbanó motoros és az elektromos autókra is. A szabályok nagyrésze biztonsági előírás, illetve korlátozások. A következőkben az elektromos autókra vonatkozó szabályokra térnék ki:[3]

- Tractive system (TS): Minden olyan alkatrész, ami a motorhoz vagy a magasfeszültségű akkumulátorhoz csatlakozik.
- A megengedhető legnagyobb feszültség két elektromos csatlakozás között az autóban 600 V egyenáram.
- A TS nem kaphat nagyobb teljesítményt az akkumulátorból mint 80 kW.
- Minden autóban kötelezően használni kell egy szigetelés figyelőeszközt.
- A szigetelés figyelő amennyiben 500 ohm/V értéket mutat le kell kapcsolja a biztonsági kört ezzel megszakítva az áramellátást az autóban.
- Biztosítékot kell helyezni az akkumulátor cellákhoz, ami megszakít a cellák maximum áramleadásánál vagy alatta.
- Az akkumulátor cellák legalább 30%-ának a hőmérsékletét monitorozni kell. A hőmérséklet nem mehet a gyártó által meghatározott hőmérséklet felé vagy 60 fok fölé, attól függ melyik alacsonyabb érték. Amennyiben ezt a hőmérsékletet egy másodpercél tovább meghaladja cella le kell csatlakoztatni az akkumulátort.



- Ha bármelyik cella feszültsége kívül esik a gyártó által meghatározott feszültség intervallumon, 500 ms-nél tovább, az akkumulátort le kell csatlakoztatni.
- Az akkumulátor pozitív és negatív végén is kell lennie egy leválasztó relének és mindkettőnek nyitnia kell, mikor az akkumulátort leválasztják.[3]

## 3.2 Electric Racing Miskolc

Az Electric Racing Miskolc egy formula student csapat, melynek célja, egy formula stílusú autó teljes megtervezési és megépítése. A csapat tagjai a Miskolci Egyetem hallgatói, akik a versenyautó fejlesztésével mélyreható ismeretet szerezhhetnek az autóipar és a motorsport terén. A csapatban a hallgatóknak lehetőségük van a legfejlettebb technológiák megismerésére és alkalmazására. A Miskolci Egyetem legtöbb képzéséről vannak hallgatók a csapatban, akik képzésükhöz megfelelő feladatot látnak el. Az autóipar mai alakulását követve, mi is elektromos hajtással tervezünk. Az első autó egy központi elektromos motorral lesz hajtva, amely egy Emrax 228-as vízhűtéssel rendelkező háromfázisú elektromotor. Ezt a motort egy UniTek BAMOCAR D3-as motorvezérlő fogja irányítani. Az autó az összes többi alkatrésze a csapat tagjai által lett tervezve.

### 3.2.1 erm.e01 autó

A csapat első versenyautója az erm.e01 kódnevet viseli. Az autót egy Emrax 228-as motor hajtja. A motor hűthető levegővel és vízzel is. A motor a 2. ábrán látható.



2. ábra: Emrax 228

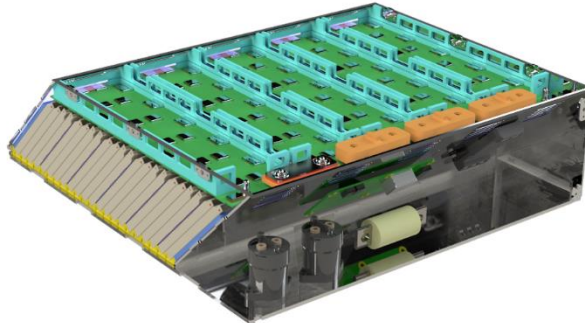
A motor egy permanens mágneses szinkron motor, amelyben 10 póluspár található. A motor maximum 630 voltnyi feszültség különbséget képes elviselni. Nálunk az akkumulátor maximum feszültsége 465 volt lesz, ami nem lépi át ezt a határt. A motor csúcsteljesítménye 124kW 5500-as fordulatszámon, amit maximum feszültségnél képes elérni. A motor hatásfoka 96%. A csúcsáram 360 amper, viszont ez folyamatosan nem

megy keresztül a motor tekercsein. Legtöbbször a csúcsáram, indításkor jelenik meg és csak pár másodpercig tart. A folyamatos áram, amit a motor elvisel az 180 amper.[4] A motor vezérlő az autóban egy UniTek BAMOCAR D3, amit a 3. ábrán láthatunk.



3. ábra: UniTek BAMOCAR D3

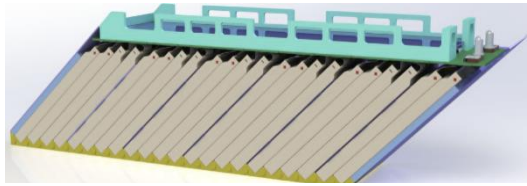
A motorvezérlő bemenetére maximum 700 voltos egyenfeszültséget csatlakoztathatunk, ami teljesen megfelel a mi akkumulátorunknak, ami maximum 465 voltos feszültségkülönbséget tud elérni. Az inverter minden többféle típusú motort is tud vezérelni ezek közé tartozik a szinkron motor is, amelyet mi használunk. Az inverter maximum 85 kVA látszólagos teljesítmény leadására képes. [5]



4. ábra: Az autó akkumulátora

Az autóban használt akkumulátor összesen 465 voltos feszültségre képes, ami már komoly biztonsági intézkedéseket igényel és ezért több magasfeszültséghez szükséges védelmi eszközt és felszerelést is be kellett szerezni. Az akkumulátorban összesen 220 darab cella van, amit a 4. ábra szemléltet. Ezen belül az akkumulátor még 5 kisebb szegmensre van szétszedve, mindegyikben 44 darab akkumulátor cella található. Az akkumulátor cellák kettesével párhuzamosan vannak kötve. A kettes párokból pedig 22 darab sorban van egymáshoz kötve, így adja ki a 44 darab cellát. Egy szegmens felépítését az 5. ábra szemlélteti. A cellák Lítium-polimer típusú, akkumulátorok, egyesével 6600 mAh kapacitással és 3,7 voltos nominális feszültséggel rendelkeznek. Teljesen feltöltött állapotban akár 4,2 voltos feszültséget is le tud adni ezért ezzel számoltunk az

akkumulátor tervezése során.[6] Folyamatosan 99 amper leadására képes az akkumulátor, ami a párhuzamos összeköttetés miatt 198 amper lesz. Egy szegmens maximum



5. ábra: Egy akkumulátor szegmens

feszültsége 93 voltos lehet.

Mivel az autó magas feszültséggel működik ezért szigorú szabályok vonatkoznak ennek a rendszernek a felépítésére. Az egyik legfontosabb eszköz, aminek a használata kötelező az Insulation Monitoring Device (IMD), ami a TS és a karosszéria között nézi az ellenállást, és amennyiben ez egy fix érték alá csökken le kell kapcsolnia a vészkört. A vészkör gondoskodik arról, hogy bármilyen kis hiba esetén az akkumulátor le legyen csatlakoztatva. A vészkörben szerepelnek azokat a kapcsolók is, amelyek csak az autó leállításához szükségesek.[7]

Az autóban majdnem minden CAN vagy CAN FD segítségével kommunikál, az egyik kivétel a kormány elektronika, ami RS-485-ös kommunikációt használ, a másik pedig az AMS (Accumulator Management System), ami az akkumulátort figyeli. Ez utóbbi rendszernek galvanikusan le kell lennie választva tekintettel a magasfeszültségekre és ezt a leválasztást ISO SPI rendszerrel oldottuk meg. Az ISO SPI lényegében egy 1:1 arányú transzformátor, ami így biztosítja, hogy a magasfeszültségű alkatrészek elektronikáival is tudjunk kommunikálni.

## 4 Modern autóelektronika

Az elmúlt húsz évben a beágyazott rendszerek és elektronikák rengeteg új funkciót adtak a modern autókhoz. Csökkentették az üzemanyag felhasználást, növelték az autók biztonságát és kényelmet. Ennek eredményeképpen a mai járművek rengeteg bonyolult hálózatot és nagy mennyiségű mikrovezérlőt tartalmaznak. Ezek a rendszerek több millió kód sort tartalmaznak, amelyek a sok különálló mikrovezérlőkön futnak és olyan belső hálózatokon kommunikálnak, mint amilyen a CAN busz is. Manapság eljutottunk odáig, hogy a CAN busz 1Mbps-os sávszélessége túl lesz terhelve a jövőbeli autók kommunikációjához szükséges adatmennyiséggel és a CAN egy fejlettebb változata kezdett elterjedni a CAN Flexible Data rate (CAN FD). A CAN FD 8Mbps-os sávszélességre képes, ami jelenleg még megfelel az adatok mennyiségének. A gyártók viszont már most fejlesztik a következő kommunikációs protokollt, ami a CAN XL, mely képes lesz 20Mbps-os sávszélességre. A CAN mellett megtalálhatunk még olyan kommunikációkat is mint, amilyen a Local Interface Network (LIN). A LIN viszont a CAN hálózattal ellentétben nem az egész autó rendszerének kommunikációjára van használva, amire nem is lenne megfelelő, ugyanis kis rendszerekhez lett tervezve. Ezért a LIN kommunikációt manapság úgy használják, hogy egyes területeken kisebb hálózatokat hoznak létre, amik saját mikrovezérlővel és LIN kommunikációval rendelkeznek. Ezeket a kisebb hálózatokat pedig CAN segítségével csatlakoztatják az autó egész rendszeréhez.[8]

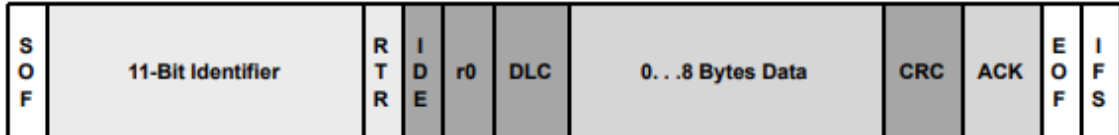
### 4.1 Control Area Network (CAN)

A Control Area Network (CAN) kommunikációs protokoll egy busz topológiát használó kommunikáció. A CAN buszt a Robert Bosch GMBH fejlesztette ki, kifejezetten autók bonyolult vezetékes rendszerének a kiváltására. A CAN egy két vezetékes buszrendszert használ differenciális jelszinttel. A differenciális jel miatt könnyen ellenáll az elektromos interferenciáknak, képes öndiagnosztikára és adat hibákat detektálni és javítani. Ezen tulajdonságok, miatt a CAN egy elterjedt kommunikációs rendszer lett, ami számos ipari területen használnak például, autó ipar, tengerészeti és repülő rendszerekben, valamint az egészségügyben is.

### 4.1.1 Busz topológia

A busz topológiánál az összes eszköz egy vezetékre van kötve, amelyet busznak hívnak. Ez a vezeték egy közös kommunikációs térként szolgál, lehetővé téve az összes eszköz számára, hogy ugyanazt az üzenetet egy időben fogadja. Busz topológiánál, mikor egy eszköz üzenetet küld ez az üzenet, minden buszon lévő eszközhöz eljut és a fogadó eszközök döntenek el, hogy a kapott adattal mit kezdenek. A busz rendszer előnye a könnyű telepíthetőség és a kevés kábelhasználat, továbbá könnyen lehet új eszközöket csatlakoztatni a már meglévő rendszerünkhöz. Főleg kicsit és közepes méretű hálózatoknál használatos. Hátrányai közé tartozik, hogy érzékeny a vezetékek megsérüléséből eredendő hibákra, ugyanis ezek kihatással lehetnek az egész hálózatunkra. Továbbá nagy rendszerekben nem a legjobb megoldás ugyanis a jel gyengül nagy távolságokat megtéve. Az adat átvitel busztopológiában úgynevezett ütközés detektálással működik. Amennyiben két eszköz egyszerre próbál kommunikálni a buszra, ütközés történik és a közölt információ sérül. Ütközés után az adott eszközök, várnak egy ideig majd újra próbálják az információ küldést.[9]

### 4.1.2 Standard CAN



6. ábra: Standard CAN, 11 bites azonosítóval

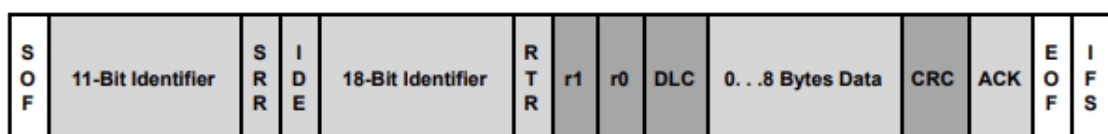
A 6. ábrán láthatjuk egy CAN üzenet felépítését bitekre osztva. A bitek a következőket jelentik:

- SOF: Az úgynevezett start of frame bit az első bit, amit kommunikáció kezdésénél az eszközök küldenek a buszra. Ez mindig domináns értékű és a lényege az eszközök szinkronizálása, tétlen állapot után.
- Identifier: A Standard CAN egy 11 bites azonosítót használ, amely eldönti a küldött üzenet prioritását. Az azonosító minél kisebb bináris érték annál nagyobb lesz a prioritása az üzenetnek.
- RTR: A remote transmission request, tehát távoli üzenetkérő bit lényege, hogy eldöntse az adott üzenetkeretről, hogy ez egy adatot továbbít vagy adatot kér egy másik eszköztől. A bit domináns mikor adattovábbítás van.
- IDE: Egy azonosító kiegészítő bit mely értéke domináns, amennyiben a standard CAN azonosító van használva kiegészítések nélkül.

- r0: Ez a bit a jövőbeli bővítésekre, és a CAN továbbfejlesztésére lett fenntartva. Általában domináns, de lehet recesszív is.
- DLC: Négy bit hosszúságú bináris érték, megmutatja mennyi bájtnyi adat lesz elküldve az üzenetkereten belül.
- Data: Ide kerülnek az adatbitek, amiket továbbítani szeretnénk. Maximum 64 bitnyi adatot vagyunk képesek küldeni egy üzenetkereten belül.
- CRC: A ciklikus redundancia-ellenőrzés egy 16 bit hosszú hibaellenőrzés, amiben az utolsó bit a határolóbit, melynek értéke mindig recesszív. A 16bit hosszúságú mező az elküldött üzenetnek a hibadetektálására szolgál.
- ACK: Minden eszköz, ami pontos üzenetet kapott, ezt a recesszív bitet felülírja egy domináns bittel, ezzel jelezve az üzenet hibamentes megérkezését. Amennyiben valamelyik eszköz ezt a bitet recesszív értéken, hagyja abban az esetben az üzenet újra azonosítás után meg lesz ismételve. Ezzel a módszerrel minden buszon lévő eszköz hitelesíti az adatot. Az ACK 2 bit hosszú, melyből az első bit a nyugtázásra van a második pedig a határolóbit.
- EOF: Az üzenetkeret végét hét recesszív bit jelzi és ez kikapcsolja a bit beillesztést, ezzel együtt ellenőrizve is azt. A bit beillesztés minden öt azonos bit után beszúr egy ellentétes értékű bitet, hogy az üzenet szinkronban maradjon. Amennyiben az üzenet végén az EOF hét recesszív bitjébe a bitbeillesztés beszúr egy bitet, akkor ez a funkció nem működik megfelelően, mert ki kellett volna kapcsoljon.
- IFS: Biztosít két üzenet között, egy kis szünetet, hogy az eszközök a megkapott információt pontosan tudják feldolgozni. Legtöbb esetben három recesszív bit, de bővíthető hét bitre is, amennyiben a rendszer megkívánja ezt.[10]

### 4.1.3 Extended CAN

Az extended CAN ugyanannyi adatbitet tud továbbítani egy üzenettel, mint a Standard CAN viszont az arbitrációs mező ebben az esetben hosszabb. Ez lehetővé teszi, hogy a buszon még több eszközt tudjunk pontosan megkülönböztetni.



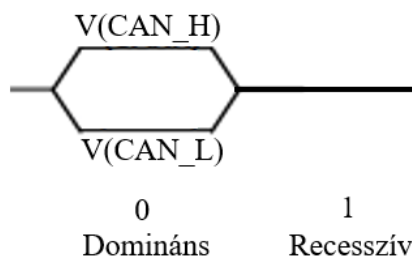
7. ábra: Extended CAN, kiegészítve a 18 bites azonosítóval

A7. ábrán látható, hogy vannak eltérések a Standard és az Extended CAN között, ezeket a következőben fejteném ki:

- SRR: Az extended formátumban az RTR bitet helyettesíti, mint egy helyfoglalóként. Értéke mindig recesszív.
- IDE: Értéke recesszív extended CAN esetében. Jelezve ezzel, hogy az azonosító még folytatódik.
- 18-Bit Identifier: Az IDE-t követő 18 bit, amely az azonosítót egészíti ki ezzel bővítve azt összesen 29 bit hosszúságúra.
- r1: Az r0-hoz hasonlóan egy fenntartott bit, későbbi használatokra, bővítésekre.[10]

#### 4.1.4 Az arbitráció

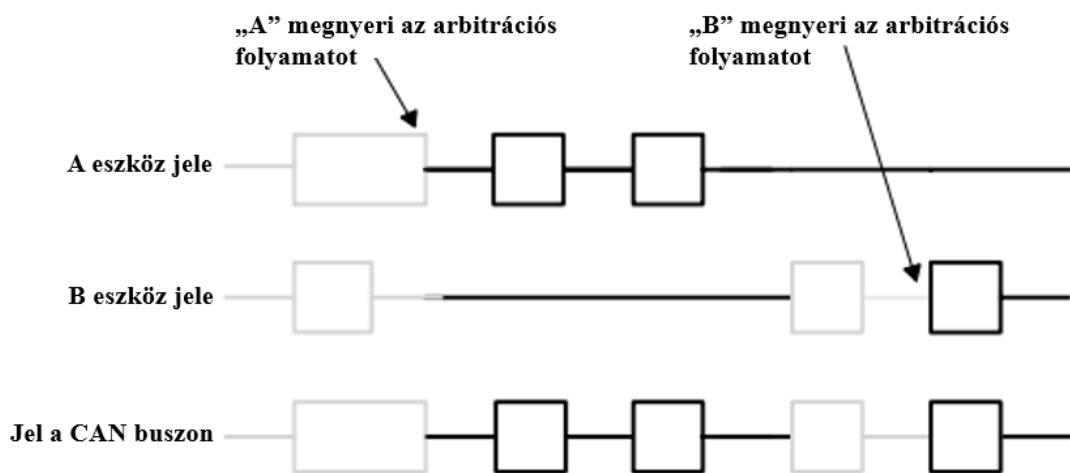
A buszhoz a hozzáférés esemény alapú ezért csak akkor van kommunikáció a buszon, ha az egyik eszköz valamilyen információt szeretne közölni. Abban az esetben, ha két eszköz egyszerre szeretne adatot küldeni a buszra, el kell dönteni melyik eszköz küldheti az információit, ennek az eldöntését hívjuk arbitrációnak. A CAN erre egy olyan módszert dolgozott ki, amelyben az ütköző üzenetek nem sérülnek. Ez annyit tesz, hogy amelyik eszköz megnyeri az arbitrációs folyamatot az folytatja az üzenet küldését anélkül, hogy ez sérülne vagy zavarva lenne másik eszköz által. Az, hogy melyik eszköz használhatja a buszt, az alapján dől el, hogy az azonosító mezőben mekkora a prioritása. Az azonosító részben minél kisebb valakinek a bináris értéke, annál nagyobb a prioritása. Legtöbb esetben a domináns, logikai magas érték az egyhez van rendelve, a logikai alacsony, recesszív érték pedig a nullás bit értékhez, ez a CAN esetében fordítva van itt a domináns bit a nulla és a recesszív az egy. Fizikai felépítés alapján a CAN differenciális jel, ami annyit tesz, hogy a két vezetéken a feszültség különbség alapján dől el az, hogy milyen érték a legyen az üzenetben ezt a 8. ábra szemlélteti.



8. ábra: A bitek és feszültség különbségek kapcsolata

A CAN esetében mikor feszültség különbség van a két vezeték között, akkor nulla a bit értéke. Ezzel könnyen el lehet dönteni a prioritást ugyanis amint egy recesszív bit érkezik, az ezzel egy időben érkező domináns bit a feszültséget fenntartja a buszon. A CAN esetében minden eszköz üzenetküldéssel párhuzamosan olvassa is a buszt, ezért könnyen tudja érzékelni a nála nagyobb prioritású üzeneteket.

Az arbitrációs folyamatot a 9. ábra jól szemlélteti. Látható, hogy a buszra egyszerre „A” és „B” eszköz is szeretne információt küldeni. Viszont az azonosítója az „A” eszköznek



9. ábra: Az üzenet prioritásának eldöntése: Az arbitráció

kisebb, ami nagyobb prioritást jelent, ezért ő küldhet üzenetet a buszra. Kis idő elteltével mikor az „A” eszköz befejezte az üzenetküldést, látható, hogy a buszon nincs éppen információ közölve ezért „B” eszköz elkezd a kommunikációt, és mivel más eszköz nem küldött üzenetet a buszra ezért az arbitrációt megnyeri és folytatja az üzenete továbbítását a buszra.[10]

#### 4.1.5 CAN üzenet típusok

- Adatot továbbító üzenet: A leggyakoribb üzenet típus, részei: azonosító mező, adat mező, CRC mező, nyugtázás mező. Az azonosítási mezőn belül eldől az üzenet prioritása, ami lehet 11bit vagy 29 bit hosszú. A CRC mező 16 bitet tartalmaz hibadetektálásra. A nyugtázási mezőn belüli ACK bit pedig jelzi az üzenet helyes megérkezését.
- Adat kérő üzenet: A lényege, hogy egy másik eszköztől kér információt az eszköz, amely ezt az üzenetet továbbítja. Nagyon hasonló az adat továbbító üzenethez,

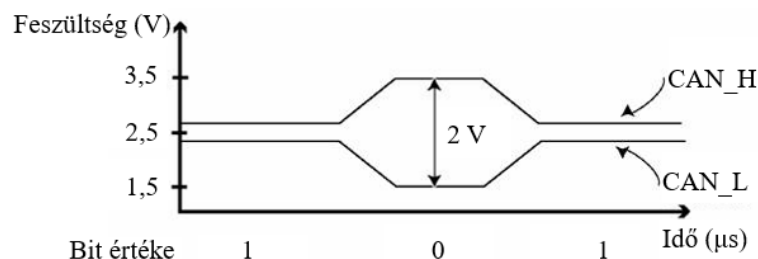


kettő nagy különbség van közöttük. Az egyik különbség az, hogy RTR bit recesszív a másik pedig, hogy az adat mező teljesen üres.

- Hiba üzenet: Egy egyedi üzenet típus, amely megszegi a CAN formai követelmény rendszerét. Akkor kerül kiküldésre, amikor az egyik eszköz hibát érzékel az üzenetben és ekkor az összes többi eszközt hibaüzenet küldésre készíteti. Ezek után az eredeti üzenetküldő megismétli az üzenetét. A CAN vezérlőkbe van építve egy áramkör, ami megakadályozza, hogy egy eszköz folyamatosan hibaüzeneteket küldjön.
- Túlterheltségi üzenet: Hasonló a hiba üzenethez, általában, akkor kerül kiküldésre, ha az adott eszköz túlságosan elfoglalt lesz. Elsődlegesen arra van használva, hogy két üzenet között egy kis extra késleltetést nyújtson.[11]

#### 4.1.6 A CAN fizikai felépítése és tulajdonságai

A CAN egy két vezetékes buszrendszert használ, amelyek közötti feszültség különbség dönti el a bitek értékét. A két vezeték tipikus megnevezése a CAN\_H (CAN high vagyis CAN magas), illetve a CAN\_L (CAN low, vagyis CAN alacsony) ezek a nevek a vezetéken lévő feszültséget reprezentálják. A CAN recesszív, vagyis 1-es bit értéket ad, abban az esetben, ha a vezetékek közötti feszültség különbség 0 V, vagy maximum 0,5 V. A két vezetéken ilyenkor egységesen 2,5 voltos feszültség fog megjelenni. Amennyiben domináns 0 bitértéket szeretnénk előállítani a két vezetéken a feszültség különbségnek 2 voltosnak kell lennie, de minimum 0,9 voltosnak. Ekkor a CAN\_H vezetéken a feszültség 3,5 V míg a CAN\_L vezetéken 1,5V. Ezt a 10. ábra szemlélteti.



10. ábra: A CAN vezetéseinek feszültség értékei, és bitértékei

A busz vezeték két végét egy 120 ohmos lezáró ellenállással szokták összekötni. A CAN két vezetéke egy sodrott érpárt alkot, ami erősen ellenállóvá teszi a kommunikációt, az elektromos és mágneses zavarok ellen. Ez a vezeték lehet árnyékolt vagy árnyékolás mentes.

A CAN multimaster elven működik, aminek lényege, hogy a buszra bármelyik eszköz küldhet üzenetet és nem kell egyiknek se engedély a másiktól. Továbbá a CAN

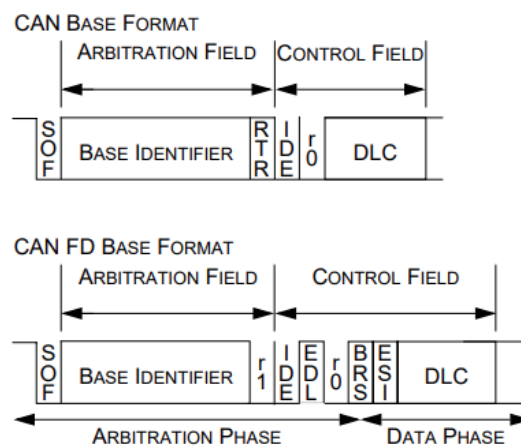
üzenetorientált, amely azt jelenti, hogy az üzenet fontosságától függ annak prioritása nem attól, hogy ki az üzenet küldője.

A CAN egy NRZ (Non Return to Zero) kódolást használ, ami annyit jelent, hogy az egymást követő bitek után a feszültség szintek nem térnek vissza a nullába. Az NRZ azért előnyös, mert egy bit átviteléhez egy bitidő szükséges, viszont a szinkronizációt megnehezíti. Erre találták ki a bit beillesztést, ami az üzenet küldésekor, minden öt azonos bit után beszúr egy ellentétes értékű bitet. Ezeket a beszúrt biteket a vevő eszköz fogja eltávolítani még az üzenet dekódolása előtt.

A CAN átviteli sebessége 5Kbit/s és 1Mbit/s között tudjuk megválasztani. Figyelembe kell venni, hogy a sebesség felső határát tipikusan 40 méternél rövidebb busz vezetékkel lehet elérni.[12]

#### 4.1.7 A CAN FD

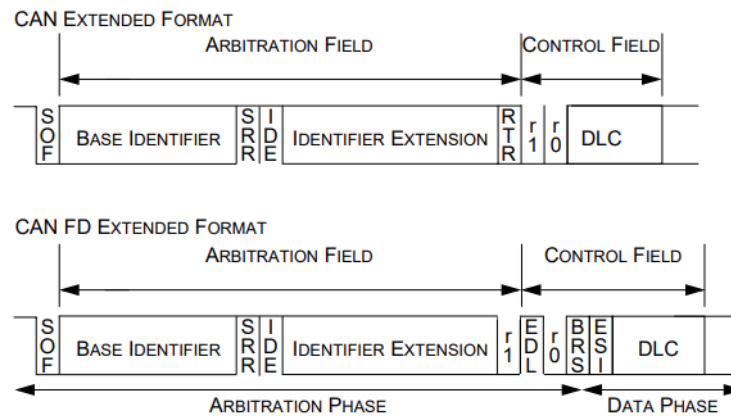
A Control Area Network Flexible Data rate lényege a nevében rejlik, ugyanis üzenet közlés közben képesek vagyunk változtatni az adatátviteli sebességet. Ezt olyan módon tehetjük meg, hogy az üzeneten belül az adatmező sebességét lehetséges felgyorsítani így az több adatot tud továbbítani. Az üzenetek felépítésében a különbségeket Standard CAN esetében a 11. ábra szemlélteti



11. ábra: CAN és CAN FD közötti különbségek standard üzenetformában

Az első látható különbség az RTR bitnél van, ugyanis CAN FD esetében nincsen üzenetkérő bit, hiszen a változtatható sebesség csak az adatmezőre van hatással, ami ebben az esetben nem lenne. Ez a bit CAN FD esetében egy domináns r1 nevű bittel van kiváltva. A következő bit az EDL (Extended Data Length) bit, amely azt jelzi, hogy az adat mező standard maximum 8 bájtot vagy többet fog szállítani. Ez a bit recesszív lesz CAN FD esetében alap esetben pedig domináns és akkor ez a bit az r0. CAN FD esetében

jön egy r0 bit, ami későbbi fejlesztésekre van félretéve. A BRS (Bit Rate Switch) bit dönti el, hogy az adat mezőben a bitráta feljebb lesz-e emelve. Domináns bitként a bitráta nem fog változni és az üzenet végig, azonos sebességgel lesz küldve, míg recesszív értéknél



12. ábra: CAN és CAN FD közötti különbség extended CAN esetében

az adatmező magasabb bitrátaival lesz közölve, mint a többi. Az ESI (Error State Indicator) bit azt mutatja, hogy az üzenet küldő aktív vagy passzív hiba állapotban van. A 12. ábrán láthatjuk a CAN FD Extended változatát. Különbség még az üzenet végén mutatkozik meg, ahol a CRC mező hossza változik az üzenet hosszától függően Egészen 21 bites hosszig.[13]

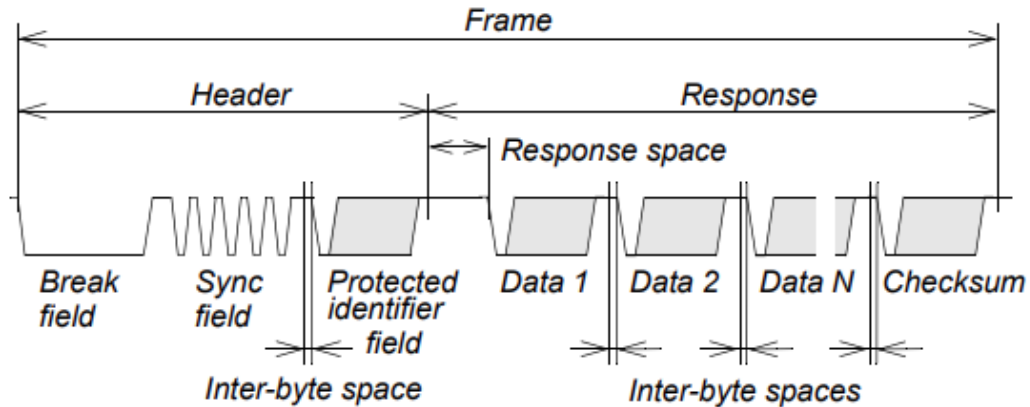
## 4.2 Local Interconnect Network (LIN)

A Local Interconnect Network (LIN) egy alacsony költségű kommunikáció, ami direkt az autóipar számára lett fejlesztve. A LIN nem konkurencia a CAN kommunikációval szemben inkább, annak a kiegészítésére találták ki. Segítségével kisebb, helyi hálózatok hozhatóak létre, ami összefoglal sok, kis elektronikát és ezeket így nem kell egyesével a CAN buszra csatolni, ezzel is csökkentve a kábel használatot. LIN segítségével általában olyan eszközöket szoktak összekötni, amiknél nem számít magas adatátviteli sebesség és az üzenetek meghibásodása nem vezet veszélyes helyzethez. Ilyenek lehetnek a kényelmi elektronikák, vagy az ablaktörlő, tükör állító elektronikák.

A LIN master-slave alapon működik, aminek a lényege, hogy a buszra csatlakoztatott eszközök közül mindig csak az küldhet üzenetet, akinek azt a master eszköz engedélyezte. A kommunikáció sebessége 20 kbit/s-ig mehet fel. A LIN egy vezetékes buszt használ ezzel is csökkentve a költségeket.[14]

#### 4.2.1 A LIN üzenetkerete

A LIN üzenetkerete általában kettő nagy részre bontható az első rész a header nevet viseli, amit a master küld a buszra míg a másik rész a response néven ismert válasz a slave eszköztől. Ezt a két nagy részt még további kisebb részekre lehet osztani, amit a 13. ábra szemléltet.



13. ábra: A LIN üzenetkeret felépítése

A 13. ábrán látható, hogy a header (üzenet fej) része az üzenetnek három kisebb részre oszlik. Az első része az üzenet fejnek a break field (állomásszinkronizálási szünet), aminek a lényege, hogy jelezze az összes többi eszköz felé az üzenet érkezését. Ez 13 domináns bitből és egy recesszívbitből áll. A bitek értéke itt is a megszokotthoz képest fordítva van, tehát a domináns bit a 0 a recesszív pedig az 1. A LIN költségkímélő kialakítása végett gyakran használ RC-oszcillátort, amely egy kondenzátorból és induktivitásból összerakott rezgőkör. Ezek elég pontatlanok szoktak lenni ezért az üzenet kereten belül a következő rész a szinkronizálási mező (Sync field), amely 8 bitből áll. Ez a nyolc bit váltakozva domináns és recesszív értékű. Hexadecimálisan az 55-ös értéket adva (0x55). Az ezt követő rész a Protected identifier field (PID), amely az azonosításra szolgál, összesen 8 bit hosszú. A nyolc bitből 6 darab bit van azonosításra használva a másik kettő paritás bit. Az azonosító 0 és 63 között bármilyen értéket felvehet. A 0 és 59 közötti értékek vannak használva az adatot továbbításakor. Az azonosító mező értéke 60 és 61 között van mikor diagnosztikai vagy konfigurációs adatot közlünk. Illetve a 62-es és 63-as azonosító későbbi változtatásoknak van fenntartva. Ez után kettő paritás bit jön, amelyek értéke az azonosító értékéből van kiszámolva. Miután a master leadta a header üzenetet egy kis várakozási idő van a buszon, hogy a kapott üzenetet mindenki feltudja dolgozni és el tudják dönteni kinek kell és milyen információt közölnie a buszon, továbbá a paritásbiteket ellenőrizni tudják. Az információ feldolgozása után a megfelelő slave a megfelelő adatot a buszra küldi. Az adat hossza maximum 8 bájt lehet tehát összesen 64

bit. Az adatmező után következő bitek (checksum) az üzenetben esetlegesen felbukkanó hibák detektálására szolgálnak. Összesen nyolc bit van hibadetektálásra, amennyiben valamelyik eszköz hibát érzékel az üzenetet figyelmen kívül hagyja. A buszra küldött üzeneteket, minden buszon lévő eszköz látja és az abban lévő adatokat, nyugodtan felhasználhatja.[14]

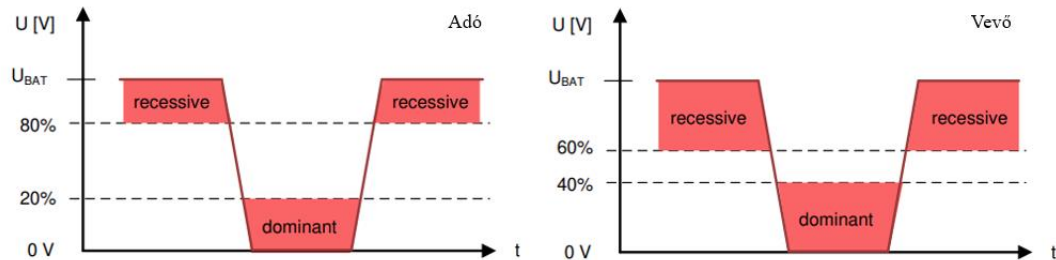
#### 4.2.2 A LIN üzenet fajtái

- Unconditional frame: Ez az alapvető kommunikáció a LIN buszon. Háromféleképpen történhet, de mindegyiknek ugyan az a felépítése. Első esetben a master információt kér az egyik slave eszköztől és ezt ő fel is dolgozza. A második esetben a master maga küldi az adatot is valamelyik slave eszköznek, ilyenkor az adat részét is a master tölti fel az üzenetnek. Harmadik opció pedig mikor a master olyan üzenetet kér, ami egy másik slave számára fontos információt tartalmaz.
- Event triggered frame: Ez az üzenet típus arra lett kitalálva, hogy a buszon a polling (folyamatos kérdések a slave eszközöktől) mennyiségét csökkentse. Az üzenet egy speciális azonosítót tartalmaz, ami annyit kér a slave eszközöktől, hogy azoknál egy bizonyos esemény bekövetkezett-e. Amennyiben az esemény nem következett be a slave nem küld választ, abban az esetben, ha viszont bekövetkezett akkor elküldi a kért információkat. Amennyiben több helyen is bekövetkezett az esemény, akkor az üzenet küldésben ütközés léphet fel. Az ütközés, amennyiben hibás üzenethez vezet a master által javítva lesz, olyan módon, hogy végig kérdezi a slave eszközöket, egyesével, így mindenki közli az általa tárolt információt.
- Sporadic frame: Egy speciális üzenet, ami ritkán van használva és általában nincsen beleépítve a master ciklikusan ismétlődő üzenet rendszerébe. Csak komolyabb adatok megváltozásánál vannak kiküldve.
- Diagnostic frame: Diagnosztikára vagy beállításra szolgáló üzenetkeret. Ebben az üzenetben mindig pontosan nyolc bájtnyi adat van és az azonosító vagy 60-as, vagy 61-es értéket vesz fel.[14]

#### 4.2.3 A LIN fizika felépítése

A LIN vezérlők általában képesek a tápellátásukat közvetlenül az akkumulátorról kapni. A jelszintek ehhez megfelelően igazodni és hasonlóan, mint a CAN esetében a domináns

bit a nulla lesz a recesszív pedig az egyes. A jeleket a LIN vezérlője az akkumulátor feszültségéből állítja elő ezért pontos érték nincs meghatározva. A feszültségszintek az adó és vevő számára eltérően vannak értelmezve. A 14. ábrából jól látszik, hogy az adó számára a jelnek a tápfeszültség 80%-át kell elérnie ahhoz, hogy egyes, vagyis recesszív



14. ábra: A LIN feszültségszintjei és azok logikai értékei

jelként értelmezze, míg vevő számára a jelnek elég a tápfeszültség 60%-át elérnie ahhoz, hogy egyes jelnek érzékelje.

A buszon lévő feszültségnek adó esetében legalább a tápfeszültség 20%-a alá kell csökkennie, hogy az olvasott bit értéke nulla, azaz domináns legyen. A vevő esetében itt is nagyobb intervallummal találkozunk ugyanis a feszültségnek elég a tápfeszültség 40%-a alá csökkennie, hogy domináns, azaz nulla értéként érzékeljük. A tápfeszültség nincsen fixálva a LIN vezérlők általában 9 és 18 volt közötti feszültséget viselnek el tápfeszültségként, de vannak vezérlők, amik egészen 30 voltig kibírják.[15]

## 5 Beágyazott rendszerek mikrovezérlők

A beágyazott rendszer egy kombinációja a számítógépes hardvernek és szoftvernek, amit egy adott feladat elvégzésére terveznek. Manapság rengeteg helyen használják őket: ipari környezetben, felhasználói elektronikákban, agrár szektorban, autóiparban, repülőkön, digitális órákban és telefonokban. A beágyazott rendszerek vagy mikroprocesszorral vagy mikrovezérlővel vannak ellátva. A kettő között a különbség, hogy a mikrovezérlő egy chip, amibe előre meghatározott mennyiségű RAM és ROM van építve, valamint további I/O eszközök. Míg a mikroprocesszor csak maga a processzort tartalmazza és a működéséhez szükséges hardvereket, nekünk kell hozzá beszerezni. A beágyazott rendszerek nagyon különbözőek lehetnek. Vannak, amelyek egész kicsik és csak egy-egy kis funkció ellátását szolgálják, míg vannak olyanok, amik HMI rendszerrel is rendelkeznek és komplex grafikai számításokat is végeznek.[16]

### 5.1 Mikrovezérlők

A mikrovezérlők legtöbb esetben Harvard architektúrára épülnek, és RISC (reduced instruction set computing) típusú utasítás készlettel rendelkeznek. A mikrovezérlők egy hardveren belül tartalmazzák, a központi feldolgozó egységet (CPU), a memóriát (RAM), a háttértárat, ami tipikusan flash memória szokott lenni és a perifériákat, amik lehetnek például: időzítő, analóg digitális átalakítók vagy kommunikációs protokollok. A mikrovezérlők, kicsi és könnyű kialakítások, miatt tökéletesek, olyan elektronikák működtetéséhez, amik elemről vagy akkumulátorról működnek. Ezen felül, ha egy specifikus feladatot kell ellátni, akkor is kiváló választás, sokan hobbyból is foglalkoznak velük. Legtöbbször C nyelven lehet őket programozni és a gyártó szokott biztosítani szoftvert a mikrovezérlők beállításához és programozásához.

#### 5.1.1 Felépítésük

- CPU: A központi feldolgozási egység az a része a mikrovezérlőnek, ahol a számítások, adatfeldolgozások végbe mennek. Ebben található az ALU (aritmetikai és logikai egység) és regiszterek, amik kevés adat tárolására szolgálnak viszont rögtön elérhetőek a processzor számára.
- RAM: (Random Access Memory) Egy ideiglenes tároló rész, ami addig tárolja az adatokat, amíg azok feldolgozásra kerülnek. Csak addig tárol adatot, amíg áramellátást kap.

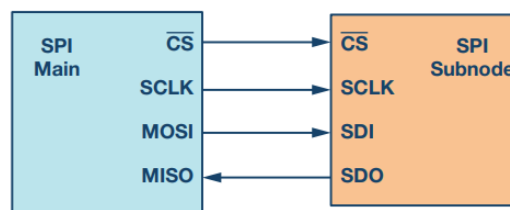
- ROM: (Read Only Memory) Általában egy nagyon alap szoftvert tartalmaz nem mindig része a mikrovezérlőknek.
- Flash: Egy olyan memória típus, ami nem csak olvasható, hanem írható is. Manapság gyakran használják mikrovezérlőkben az alap program tárolására.
- Oszcillátor: A mikrovezérlő beépített oszcillátora biztosítja az órajelet a feladatok végrehajtásához. Előfordul, hogy pontosabb oszcillátor kell, ilyenkor egy kristály oszcillátort szoktunk használni.
- I/O: Kimeneti és bemeneti portok. Ezek szükségesek ahhoz, hogy a mikrovezérlőhöz tudjunk csatlakoztatni további perifériákat.[17]

### 5.1.2 Működés

A mikrovezérlők gyakran csökkentett utasítás készlettel rendelkeznek (risc). Ezek a processzorok egy órajel alatt egy utasítást hajtanak végre. Az éppen következő utasítás adatait a cache memóriába szokták betölteni és az eredmények is ide kerülnek. Ezek után az adatot vagy tároljuk vagy a mikrovezérlő lábán ezzel egy jelet adunk egy perifériának.

## 5.2 Serial Peripheral Interface (SPI)

A Serial Peripheral Interface az egyik legtöbbször használt interfész mikrovezérlők és perifériák között. Ilyen perifériák például: szenzorok, analóg-digitális átalakító, digitális-analóg átalakító vagy akár a RAM. Az SPI egy kétirányú kommunikáció, ami szinkron módon működik és ezt a jelet a master állomás állítja elő. A szükséges vezetékek mennyisége a slave eszközök számától függ, ami minimum három vezetéket eredményez abban az esetben, ha egy slave eszköz van. Bizonyos csatlakoztatási módszerekkel a vezetékek szükséges száma négy vezetéken tartható. Az SPI interfész egyetlen egy master állomást és egy vagy több slave állomást támogat. Ezek összeköttetését szemlélteti a 15. ábra.



15. ábra: A csatlakozások main és subnode között

Látható, hogy a main (master) és a subnode (slave) között négy vezeték fut. Ezek a következők:

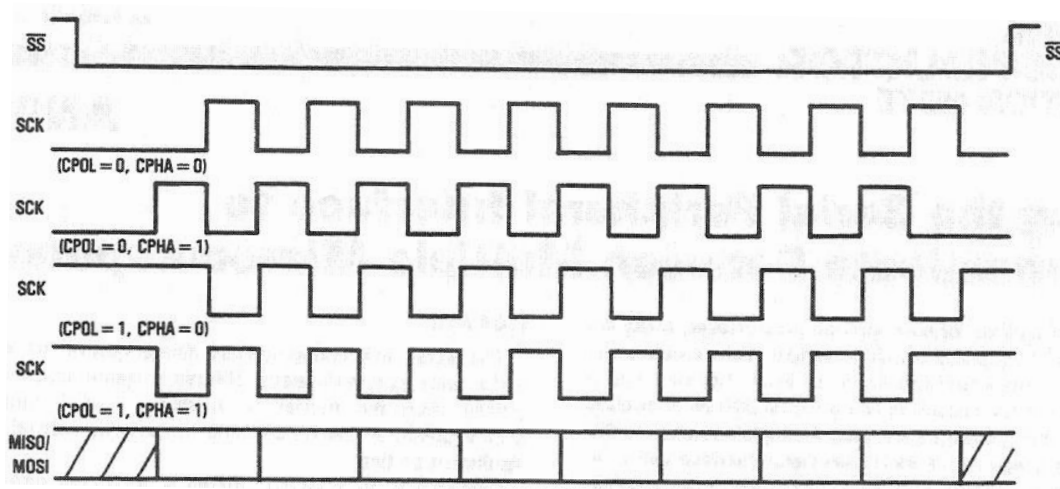


- CS (chip select): A kiválasztó jel, általában nullában aktív jel, és amennyiben magas értékre, azaz egyre változik az adott subnode nem küldheti tovább a jelét. Mindig a main dönti el, éppen ki kommunikálhat, ezt a jelet is ő tudja változtatni.
- SCLK: Az órajel, ami a szinkronizációért felel, a main node küldi.
- MOSI (Master Out Slave In): Ezen a vonalon küldi az információt a master node a subnode számára.
- MISO (Master In Slave Out): Ezen a vonalon kapja a választ a master az éppen kérdezett eszköztől.[18]

### 5.2.1 SPI adat továbbítás

A master egy órajelet állít elő, amelyet minden slave állomás megkap, ezzel biztosítva szinkronizációt, ami fontos az adatok mintavételezési szempontjából. Amennyiben a master engedélyezi a slave kommunikációját, az elkezdheti küldeni az információkat. Az információk megszakítás nélkül folyamatosan jönnek a vezetéken. Az üzenetnek nincs egy fix mérete, ez mindig a perifériáktól függ, egy üzenet lehet akár csak pár bit vagy akár több bájt is. Amennyiben az adatot megkapta a master a Chip Selectet egyre változtatja, és ez letiltja a slave kommunikációját.

Az adatok mintavételezése a buszról négyféleképpen történhet, amit kettő változó átállításával lehet meghatározni. Ez a két változó a CPOL (Clock Polarity) és a CPHA (Clock Phase). A CPOL határozza meg, hogy tétlen állapotba az órajel milyen jelállapotot vegyen fel, ez lehet egyes vagy nulla. A CPHA változó pedig azt adja meg, hogy mikor legyen az információ mintavételezése és mikor jöjjön a következő bit az adatvonalon. Ezek mindig ellentétesen lesznek beállítva, tehát ha az órajel felfutó élére mintavételezés lesz akkor a lefutó élre fogják küldeni a következő bitet. Ha a CPHA nulla értéket kap



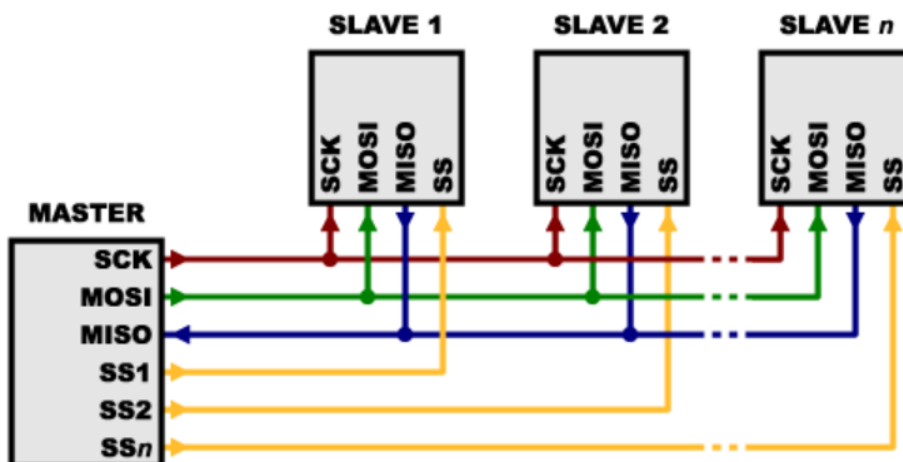
16. ábra: SPI modes

akkor az információ az órajel felfutó élén lesz mintavételezve és a bit a lefutón lesz léptetve, ha pedig egyest akkor a lefutó élen lesz mintavételezve és a felfutó élen lesz léptetve. Ezzel a módszerrel garantáljuk, hogy a jelszintnek lesz ideje beállni mire a mintavételezés történik és így nem kapunk hibás értékeket. A 16. ábra a lehetséges kombinációkat szemlélteti.[19]

### 5.2.2 SPI fizikai felépítése

Az SPI master minimum három vezetéken keresztül csatlakozik a slave eszközhöz, de a csatlakozás fajtájától és a slave állomások számától függően nagyban változhat a vezetékek mennyisége. Az SPI rendszert kettőféleképpen szokták összekötni, ebből az egyiknél simán csatlakoztatjuk a slave eszközöket és mindegyik kap egy Chip Select vezetékét. A másik módszernél pedig egy úgynevezett daisy chainbe kötjük a slave állomásokat. Ebben a módban a slave eszközök MISO vezetéke nem a master eszközhöz van kötve, hanem a következő slave MOSI részébe. Az SPI a vezetékeken a szokásos digitális jelszinteket közli, amely egyes esetekben lehet 5V vagy 3,3V.

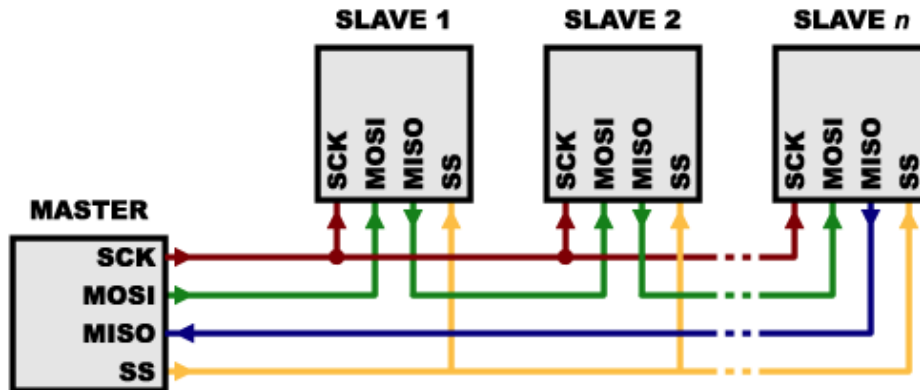
Alapvetően az SPI kommunikációt úgy kötjük össze, hogy a MISO vezeték egy buszként szolgál, amelyre éppen az a slave állomás közöl adatot, amelyiknek a master engedélyezte a jelét. Ebben az esetben a csatlakozások úgy néznek ki, hogy a master CLK (SCK), MISO és MOSI vezetékét összekötjük mindenkivel és a slave eszközök külön-külön kapnak egy engedélyező kimenetet. Ilyenkor az SPI master eszköznek minden slave állomáshoz kell lennie egy külön engedélyező kimenetnek. Ezt a kapcsolási módszert a 17. ábra szemlélteti.



17. ábra: Általános csatlakozás a master és slave eszközök között

A másik összeköttetési módszer a master és slave állomások között az úgynevezett daisy chain módszer. Ebben az esetben az órajel bekötése ugyan úgy történik, mint az előzőnél.

Jelentős különbség a MOSI, MISO és SS (Slave Select/Chip Select) vezeték bekötésénél van. A 18. ábrán látható a bekötési módszer.



18. ábra: SPI daisy chain

Az ábrán jól látható, hogy ennél a módszernél 4 vezetéknél többet sose kell használni, hiszen nem kell minden eszköznek külön CS (Chip Select) vezeték. A nagy különbség abban rejlik, hogy a master állomás MOSI vezetéke nem csatlakozik csak egyetlen slave állomáshoz, és ennek a slave eszköznek a MISO vezetéke nem a master eszközbe csatlakozik, hanem a következő slave állomásba. Ez alapján az összes eszköz egy szép körbe van kötve egymással. Az információ olyan módon áramlik ebben a rendszerben, hogy a slave állomások mindig az előző üzenete mögé kötik a saját üzenetüket.[20]

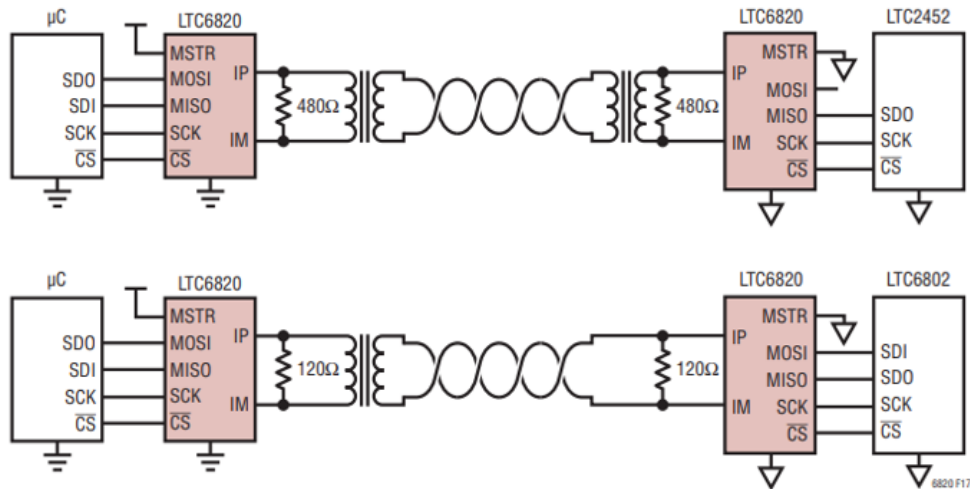
### 5.2.3 ISO SPI

Az Isolated Serial Peripheral Interface (ISO SPI) lényege, hogy a kommunikáció résztvevői galvanikusan le legyenek választva. Ez azt jelenti, hogy a két rendszer között nincsen közvetlen elektronos kapcsolat.

Az ISO SPI fizikai felépítése:

A master és slave között egy transzformátor van, aminek a két oldalán a menetszám megegyezik, ezért a feszültség és az áram nem változik egyik tekercsen sem. A két tekercs mivel fizikailag nincs összekötve ezért a master oldalára nem jöhet át hiba esetén a slave oldalán található általában magas feszültség. Mivel a transzformátor nem tud csak váltakozó árammal üzemelni ezért a master és a slave közé be van építve kettő logikai áramkör, aminek a feladata az SPI kommunikációból származó adatok kódolása valamilyen váltakozó áramú formátumra majd ezt a transzformátor másik oldalán egy hasonló áramkör feladata dekódolni. A master és slave oldalát kétféleképpen is össze lehet kötni. Megoldható az összekötés egy transzformátor használatával, amiből a sodrott érpár rögtön a slave eszközbe csatlakozik. Illetve meg lehet úgy is oldani, hogy kettő

transzformátort használunk mind a két áramkörnél egyet-egyét és közöttük lesz a sodrott érpár. Ezek az összeköttetések a 19. ábrán láthatóak.



19. ábra: ISO SPI felépítése

#### Az ISO SPI jelei:

Az ISO SPI, a transzformátorokon keresztül váltakozó árammal tud kommunikálni, ezek általában valamilyen pulzálások gyártótól függ, hogy az adott impulzusok mit jelentenek. Az LTC6820 négy feszültség impulzust különböztet meg. Összesen kettőféle impulzus van és mindkettőnek van egy rövid és hosszú változata a rövid változatban egy feszültség érték hossza 50 ns míg a hosszú változatban 150 ns egy feszültség érték, így a teljes pulzálás ideje 100 ns vagy 300 ns. A +1 pulzus feszültség értéke + $V_a$  értékből indul és - $V_a$  értékbe esik át. A -1 pulzus, ennek az fordítottja, ugyanis itt - $V_a$  értékkel kezdünk és + $V_a$  értékre vált át. Minden pulzus után a feszültség szint visszaáll nulla értékre. A  $V_a$  értéke nem fix, az általunk használt ellenállásoktól függ. A pulzások némelyikére a slave válaszol, de nem mindegyikre. Ettől függetlenül a nem válaszolásnak is lehet jelentése.

#### A pulzások jelentései:

- Hosszú +1 pulzálás: A chip select értékét magas állapotba állítja, azaz jelzi, hogy kommunikáció fog következni. A slave eszköz nem válaszol semmit.
- Hosszú -1 pulzálás: A chip select értékét alacsony állapotba állítja, azaz jelzi, hogy a kommunikáció véget ér. A slave itt már nem tud reagálni, mert a CS nem aktív.
- Rövid +1 pulzálás: A MOSI értéket magasra állítja, majd az órajel pulzál egyet. A válasz kétféle lehet. Rövid -1 ha a MISO értéke nulla, vagy semmi, ha a MISO értéke egy.
- Rövid -1 pulzálás: A MOSI értéket alacsonyra állítja, majd az órajel pulzál egyet. A válaszok erre ugyan azok lehetnek, mint az előbb.[21]

### 5.3 RS-485

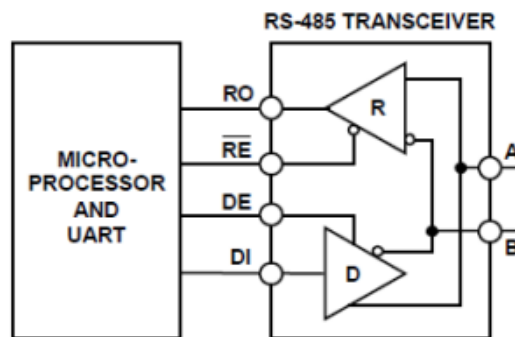
Az RS-485 standard egy fizikai felépítést specifikál, illetve az adók és vevők elektromos karakterisztikáját, és nem foglalja magában a kommunikációs protokollokat. Az RS-485-öt több ipari kommunikáció is alapul használja fizikai közegként mert messzire is pontosan tud adatot továbbítani.

#### 5.3.1 Az RS-485 tulajdonságai:

- Messzire is képes adatot továbbítani egészen 1200 méterig.
- Kétirányú kommunikációt tesz lehetővé egy sodrott érpáron keresztül
- A differenciális adattovábbítás elleállóvá teszi a zavarok ellen és csökkenti a zaj kibocsájtást is.
- A buszra több kommunikációs eszközt is fel lehet csatlakoztatni.
- Képes akár 10Mbit/s sebességre[22]

#### 5.3.2 Az RS-485 fizikai felépítése

Az RS-485 adóvevő egysége négy bemeneti jellel rendelkezik és kettő kimenetivel, ezt a 20. ábra szemlélteti. A bemeneti jelek között van az RO (Receiver output), ami az érkező



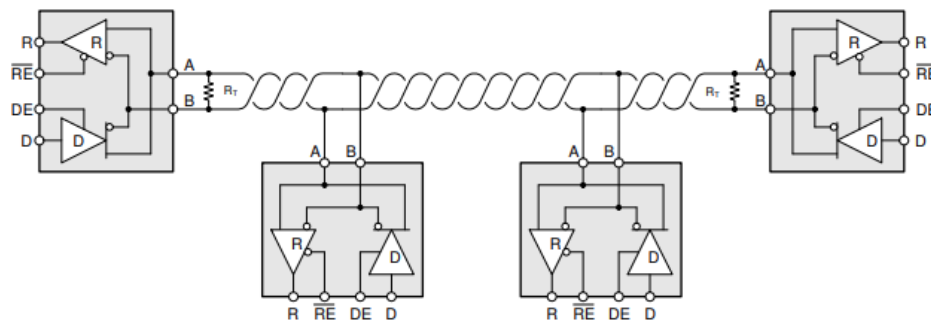
20. ábra: RS-485 adóvevő

üzenetet kiadja, a DI (Driver input), ami a küldendő üzenetet átalakítja RS-485-ös jellé, az RE (RO enable) negáltan és a, DE (DI enable) arra szolgálnak, hogy az adott pillanatban üzenet továbbítás vagy fogadás van. Ezt a két bemenetet össze szokás kötni ugyanis az egyik direkt negáltan van. Az adóvevő két kimenete pedig a sodrott érpár két vezetéke „A” és „B”. A logikai értéket „A” és „B” feszültségének összehasonlításával kapjuk meg. Abban az esetben mikor „A” feszültsége legalább 200 mV értékkel magasabb „B” értékénél, akkor ez logikai egyesként fog megjelenni az információban. Viszont, ha „A” értékéhez hasonlítva „B” eléri legalább a -200mV értéket, az az „B” feszültsége magasabb lesz „A” feszültségénél, akkor ez a logikai nulla értéket fogja

jelenteni a kommunikációban. A két vezeték között a legnagyobb feszültség érték 6 V lehet. Az RS-485 két vezetékét 120 Ohmos lezáró ellenállással kell lezárni.

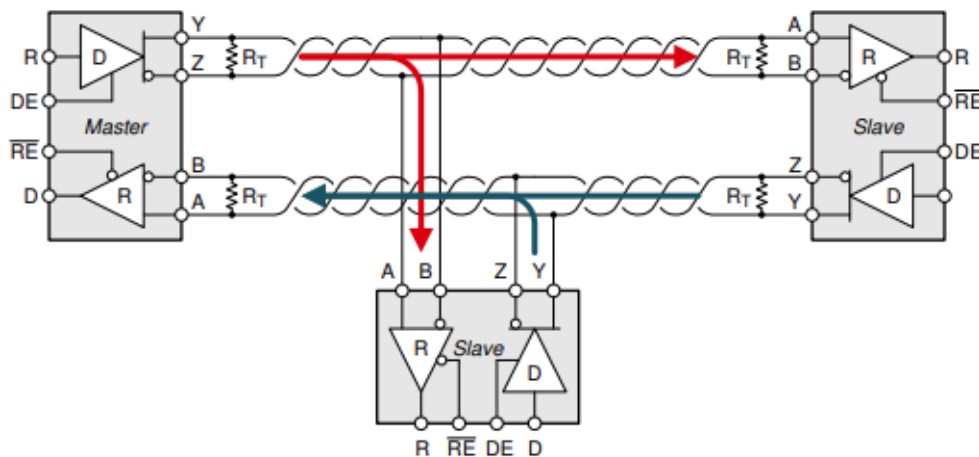
### 5.3.3 Az RS-485 összekötési módjai

Az RS-485-öt kétféleképpen szokták kötni. Half-duplex vagy full-duplex módon. Legtöbbször half duplex módon vannak összekötve az eszközök. Ebben az esetben egy sodrott érpár buszként van használva és ilyenkor ugyan lehetséges a kétirányú kommunikáció, de csak felváltva. Ezt az összekötési módszert a 21. ábra szemlélteti.



21. ábra: RS-485 half-duplex összeköttetés

A másik lehetőség a full-duplex módszer, ahol kettő pár sodrott érpárt használnak és egyszerre kétirányba is történhet kommunikáció. Ezt a módszert a 22. ábra szemlélteti.



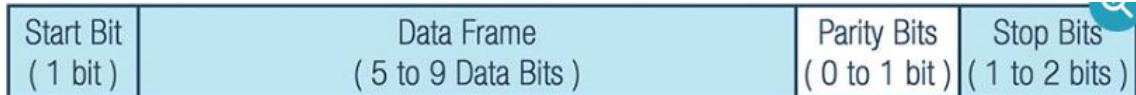
22. ábra: RS-485 full-duplex csatlakoztatása

[23]

### 5.3.4 UART (Universal Asynchronous Receiver/Transmitter)

Az UART az egyik legtöbbet használt kommunikáció eszközök között. Az UART rengetek soros kommunikációs protokollal tud együtt működni, ha jól van összeállítva. A kommunikációs protokollt legtöbb esetben beágyazott rendszerekben, mikrovezérlőkben és bármilyen számítógépben használják. Az UART-nak kettő lényeges csatlakozója van,

az RX és TX, amely az adó és vevő pont. Ezeket csatlakozásnál fordítva kell egymásba kötni, azaz az egyik eszköz RX vezetéke a másik TX csatlakozójába megy. Az UART egyszerű kommunikáció, amely úgynevezett csomagokkal kommunikál. Egy csomag felépítése a következő. Az első bit egy start bit, ezután jön az adat keret, ami 5 és 9 bit között lehet. Aztán jön a paritás bit, ami vagy van egy darab, vagy nincs. Az egész legvégén pedig 1 vagy 2 stop bit áll. Ennek a csomagnak a felépítését a 23. ábra



23. ábra: UART csomag felépítése

szemlélteti.

#### A bitek:

- Start bit: Általában magas értéken van tartva és mikor megkezdődik az adatátvitel egy órajel ciklusig nulla értékre írják.
- Adat keret: 5 és 8 közötti bit, amennyiben van paritás, ha nincs lehet 9 hosszú is. Ez tartalmazza a közölni kívánt információkat.
- Paritás bit: Nem mindig van használva. Programozótól függ az értéke, hogy milyen paritást vizsgálunk, egyest vagy nullást.
- Stop bit: Az üzenet befejezéséhez a küldő fél egy vagy 2 bit idejéig az üzenetet alacsonyból, magas értékűre állítja.[24]

## 6 A mikrovezérlő kiválasztása

A feladat elvégzéséhez a STMicroelectronics 32 bites mikrovezérlői közül, a H5-ös sorozatból az STM32H563RI mikrovezérlőt választottam. A mikrovezérlőn megtalálható minden, ami a feladat elvégzéséhez szükséges. A mikrovezérlőt az alábbi szempontok alapján választottam:

- A mikrovezérlő képes kell, hogy legyen CAN protokoll használatára
- Szükséges legalább egy CAN FD interfész
- Tudnia kell SPI kommunikációt használnia
- A mikrovezérlő 64 lábbal rendelkezzen, ez bőven elég lesz a funkciók betöltésére

Miután a szükséges szempontoknak megfelelően csökkent a mikrovezérlők száma, a következőket kezdtem el figyelembe venni: Legyen rajta legalább kettő CAN csatlakozás, a lehető legnagyobb legyen a működési frekvencia, a gyorsabb adatfeldolgozás érdekében, szintén ilyen indokokból, tartalmazzon minél több RAM-ot és flash memóriát, majd mivel itt már kettő majdnem teljesen ugyanolyan mikrovezérlő volt kiválasztottam azt, amelyik alacsonyabb áron érhető el.

### 6.1 Az STM32H563RIT mikrovezérlő paraméterei:

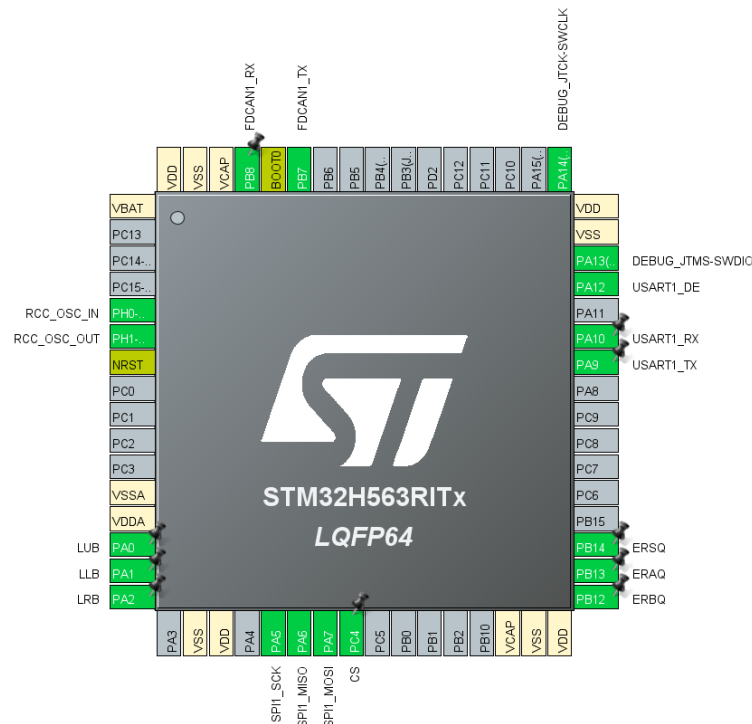
- A mikrovezérlő, egy LQFP64-es 10mm x 10mm-es tokozásban van.
- Egy ARM Cortex M33 CPU van benne, ami maximum 250 MHz-es frekvenciára képes.
- 2 MB-os flash memória van beleépítve
- 640 KB SRAM
- Négy belső oszcillátora van különböző frekvenciákkal, de külső oszcillátorokat is képes használni.
- 1,71V és 3,6V-os feszültség között kaphat tápot.
- 24 darab időzítő van benne
- 2 analóg-digitális és 2 digitális-analóg átalakító, mindkettő 12 bites felbontással[25]

### 6.2 A csatlakozók kiosztása a mikrovezérlőn

Az STMicroelectronics saját szoftvert biztosít mikrovezérlői programozásához, amelyen belül konfigurálhatjuk a hardvert, beállíthatjuk az órajelet is benne, amiben a program



segítségünkre is van, továbbá egy teljes programozási felületként szolgál. A programban C és C++ nyelven programozhatunk, tartalmaz compilert és debug funkciói is vannak. A pinek (csatlakozók/lábak) konfigurációi a 24. ábrán láthatóak.



24. ábra: Mikrovezérlő láb kiosztásai

Vannak lábak, amiket nem lehet módosítani, ilyenek a VSS, VSSA, VBAT, VDD, VDDA, VCAP. Ezeknek a lábaknak a tápellátáshoz van közük, vagy táp bemenet vagy földként. Az FDCAN1\_RX és FDCAN1\_TX a CANFD adóvevő áramkörébe mennek. Az RCC\_OSC\_IN és RCC\_OSC\_OUT pinek a külső kristály használatához kellene, amire a pontosság miatt van szükség. Az LUB, LLB és LRB lábak nyomógomboknak lesznek csatlakozások. Az SPI1\_SCK, SPI1\_MISO és SPI1\_MOSI jelek az ISO SPI adóvevőjébe fognak menni. A Mikrovezérlő programozásához, a DEBUG\_JTCK-SWCLK és a DEBUG\_JTMS-SWDIO pinek lesznek használva, illetve egy külső programozó. Az USART1\_DE, USART1\_RX és USART1\_TX az RS-485-ös adóvevőbe fognak menni. Az utolsó adag bekötött pin pedig egy enkóderbe fog menni.

## **7 Az áramkör megtervezése**

Az áramkör az Altium Designer nevű szoftverrel fogom tervezni, amely egy folyamatosan fejlesztett nyomtatott áramkör tervező program. A programban rengeteg funkció található és folyamatosan újabb és újabb dolgokkal fejlesztik. Az Altiumot széleskörűen használják, az ipar számos területén: fogyasztói elektronikák tervezésénél, automatizálási rendszerek tervezésénél, az autópárhán és telekommunikációs eszközök tervezésénél.

### **7.1 Altium Designer**

Az Altium rengeteg funkcióval rendelkezik, amelyek egy részét én is használtam a tervezés során. Az áramkörnek meg tudjuk tervezni a logikai kinézetét, majd ez alapján az Altium segít a PCB (Nyomtatott áramkör) tervezésében. Kiegészítő funkciói is bőven vannak, ilyen például, az alkatrészek keresése és létrehozása. Kapcsolatban áll más tervező szoftverekkel is, hogy egyszerűen lehessen egy áramkör köré dobozt tervezni. Lehetőség van workspace létrehozására, ami egy felhő alapú rendszer, hogy az adott fájlhoz többen is hozzá férjenek és nagyobb projekteken többen is dolgozhassanak.

#### **7.1.1 Schematic tervezés**

Miután egy projektet létrehozunk ez a következő lépés. Ebben a tervezői felületben meg lehet határozni az elektronikus összeköttetéseket, és az Altium rengeteg alkatrész használatára ad lehetőséget. A legtöbb elektronikai alkatrész megtalálható az Altiumban és akár sajátokat is létre lehet hozni, vagy hiányzókat hozzáadni. A tervezés ebben a fázisában kell összekötni minden olyan alkatrészt, amit az áramkörünkre szeretnénk tenni. Az Altiumban vannak erre segítségül szolgáló eszközök, amik figyelmeztetnek, ha valami, nem megfelelő, ilyen például a rövidzárlat vagy ha az egyik alkatrésznek nincsen lábnyoma. Az alkatrészek lábnyomai szolgálnak arra, hogy gyártásban tudják milyen területet kell rézzel fedni az adott alkatrészeknek.

#### **7.1.2 PCB (Nyomtatott áramkör) tervezés**

Miután a schematic készen van és le lett ellenőrizve, a következő lépés, a PCB megtervezése. Ezen már fizikailag látszódnak az alkatrészek és a PCB fizikai kinézetét is változtathatjuk. Meglehet adni a méreteit, hány rétegű legyen, továbbá feliratokat lehet rá rakni. Az alkatrészek elhelyezésére itt már figyelni kell, ugyanis egy rétegen egy vezetősáv haladhat nem tudják egymást keresztezni. Ilyen esetben egy másik rétegen kell tovább vinni a vezetősávot. Az Altium lehetőséget ad a megtervezett, vagy még tervezés

alatt lévő PCB 3 dimenziós megtekintésére is. Itt le tudjuk ellenőrizni, hogyan fog kinézni az áramkörünk. Ez kifejezetten fontos, hogyha az áramkör később szűkös helyen fog üzemelni. A PCB designer lehetőséget ad szabályok létrehozására és vannak beleépítve alapvető szabályok is. Ilyen szabályok lehetnek például: figyelmeztetést küld nem bekötött lábak után vagy ha túl közel van kettő vezetősáv egymáshoz, esetleg összekötünk két dolgot, amit nem kéne.

### 7.1.3 BOM (Bill of Materials)

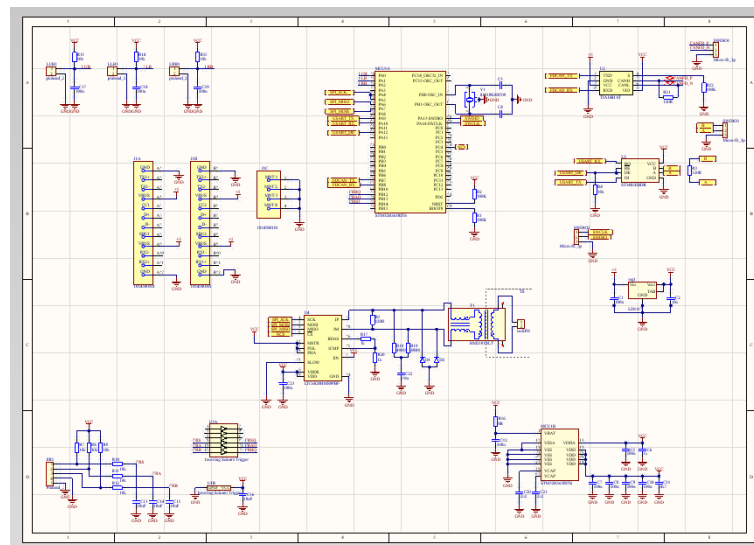
A tervezés után az Altiumban lehetőség BOM-ot létrehozni, amely tartalmazza az összes alkatrészt, azok mennyiségét egy áramkörre, leírásukat, egységárakat, továbbá, meg lehet adni, ha több áramkört gyárunk le miből mennyire lesz szükségünk. A végén az egészből számol nekünk egy teljes költséget.

## 7.2 Az áramkör részei és funkciói

A 25. ábrán látható a teljes schematic kicsinyített kinézete.

Az áramkörnek jelenleg 6 nagyobb része van, amik a következők:

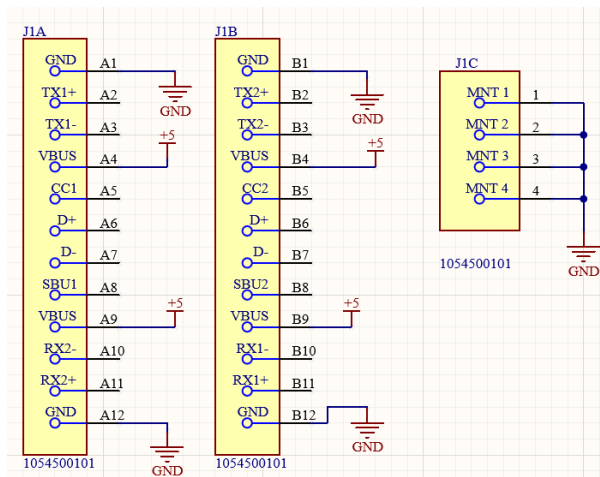
- Egy USB-C csatlakozó, ahonnan a kártya a tápellátását fogja kapni
- Egy LDO (low dropout regulator), ami az USB-C csatlakozóból jövő 5V-os feszültséget 3,3V-ra alakítja
- A mikrovezérlő és ennek a tápellátása
- Az ISO SPI adóvevő és további részei
- Az RS-485 adóvevője és csatlakozója
- A CAN adóvevője és csatlakozója



25. ábra: Az egész schematic

### 7.2.1 Az USB-C csatlakozó

Az 26. ábra jól látható, hogy az adatcsatlakozók közül egyik sincs bekötve a tervrajzon,



26. ábra: USB-C csatlakozó

ugyanis az USB-C csatlakozót csakis áramforrásként szeretnénk használni. Ugyan ebből vezetékek nem mennek, sehova mégis összeköttetésben áll az LDO-val, ugyanis a +5 jelzés egy elnevezést jelent, ami annyit tesz, hogy minden, ami ezt a nevet kapja az adott tervrajzon összeköttetést jelent. Az USB-C csatlakozó tervrajzán továbbá az is jól kivehető, hogy ez a csatlakozó szimmetrikus.

### 7.2.2 Az LDO (LD1117)

Az LDO feladata az USB-C csatlakozóból származó 5V-os feszültséget csökkenteni. Az LDO egy egyszerű és olcsó megoldása a feszültség szabályozásnak, ha a feszültséget csökkenteni szeretnénk. Ezen kívül stabilan tudja tartani a feszültség szinteket és korrigálja magát kisebb változások bekövetkeztében. A felépítésüket tekintve általában, egy NPN vagy PNP tranzisztor, vagy egy FET (field effect transistor) található bennük. A stabilitását egy hibajavító erősítőnek köszönheti, ami a bejövő és kimenő feszültség alapján szabályozza a tranzisztort, vagyis, ha eltérés van a feszültségben, akkor azt tudja javítani.

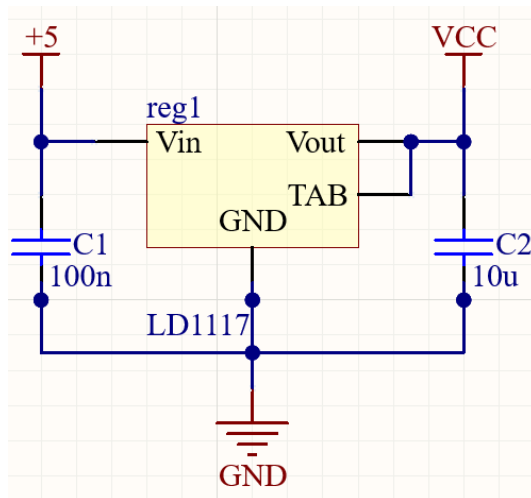
Az általam használt LDO egy STMicroelectronic által gyártott LD1117 cikkszámra hallgató LDO.

Az LD1117 paraméterei:

- A maximum bemeneti feszültsége 12V lehet
- A kimeneti feszültsége 3,3 V

- Áramot pedig 0 és 800 mA között képes leadni.[26]

Az összeköttetéseit a 27. ábra szemlélteti.



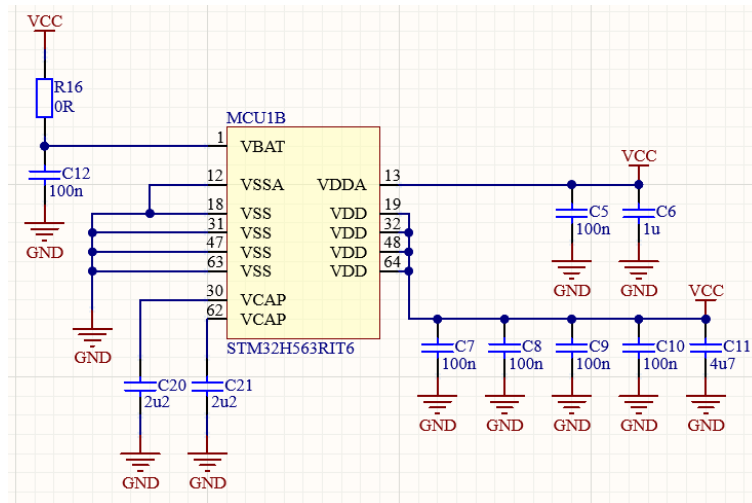
27. ábra: LD1117 LDO

A képen láthatóak az alkatrész lábai, amelyek a következők: Vin ide csatlakozik a szabályozni kívánt feszültség, Vout és TAB ezek a kimeneti lábak a szabályozott feszültség a Vout lábból jön. A TAB feladata a hőelvezetés. Valamint a GND, amelyet földhöz kell kötni. A ki és bemeneti lábaknál kondenzátorok vannak elhelyezve, melyek segítenek a feszültség simításában.

### 7.2.3 A mikrovezérlő és ennek tápellátása

A mikrovezérlő bekötését az 1.Mellékleten lehet megtekinteni. A mikrovezérlőbe a itt látszódnak a bekötött portok, amik így kapcsolatban állnak a hozzájuk tartozó adóvevő eszközzel. Továbbá látható a boot és reset bekötése egy felhúzó és lehúzó ellenálláson keresztül is. Ezekre több okból is szükség van. Segítenek a túlfeszültség ellen, mivel az ellenállások korlátozzák az áramot, így egy túlfeszültség ellen védelmet biztosíthatnak. Segítenek elnyomni a zajt. Amennyiben a tápáram instabil lenne segít a mikrovezérlő védelmében. A mikrovezérlő jobb oldalán található egy kristály, amire azért van szükség mert a CANFD nagyon pontos időzítést kíván meg. Az általam választott kristály egy 40Mhz-en működő 10ppm frekvencia toleranciával bíró kristály. A két oldalára ennek is kondenzátorok vannak kötve, ezek segítenek a frekvencia stabilizálásában, és szűrik az esetleges zajokat. A 28. ábrán látható a mikrovezérlő tápjának bekötése. Az ábrán láthatóak a különböző pinek és ezek különböző bemenetei. Az összes tápfeszültséget szolgáltató lábon találhatóak kondenzátorok, a stabilabb feszültségek érdekében. A Vdda elsődlegesen az analóg rendszereknek szolgáltató feszültséget, illetve az analóg és digitális

átalakítóknak. A Vdd bemenet szolgáltat minden másnak tápellátást. Ilyenek például az input/output csatlakozók, a belső órák és időzítők valamint a mikrovezérlő tápellátása. A

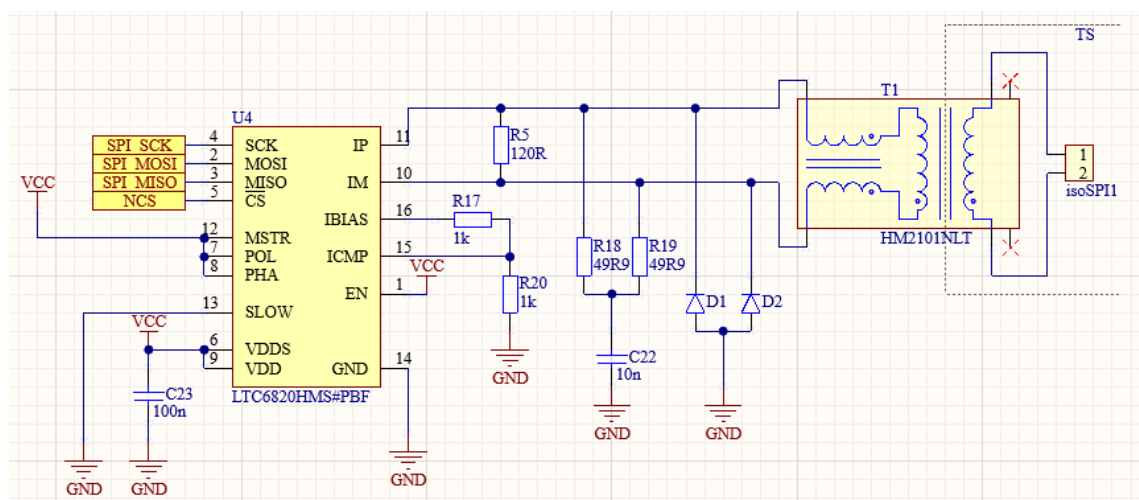


28. ábra: A mikrovezérlő tápja

Vss és Vssa a földpontja a mikrovezérlőnek. A Vbat szerepe, hogy a mikrovezérlőre akár elemet is lehet csatlakoztatni. Erre nekünk nincs szükségünk ezért ez egy 0 ohmos ellenállással lett jelezve melynek lényege, hogy a PCB-n ott lesz a hely az ellenállásnak viszont nem lesz bekötve. A Vcap pedig a mikrovezérlő belső rendszereihez szükséges kondenzátorokat csatlakoztatja.

#### 7.2.4 Az ISO SPI adóvevője

Az ISO SPI adóvevőnek az LTC6820HMS nevű chipet választottam, amely a korábban



29. ábra: ISO SPI bekötése

már leírtak alapján kommunikál és a bekötését a 29. ábrán lehet látni. A bekötésnél hasonló dolgokat kellett figyelembe venni, mint a mikrovezérlőnél. A tápfeszültség egyensúlyozásában sokat segítenek a kondenzátorok. Azokra a lábakra, amelyek adatot

visznek láthatóan be vannak kötve a mikrovezérlőnél látható portok, amelyeken keresztül a mikrovezérlőből származó információk jönnek, hogy azokat továbbítani lehessen az ISO SPI-on keresztül. A HM2101NLT nevű eszköz egy 1:1 arányú transzformátor, aminek a másik végéhez vannak csatlakoztatva az ISO SPI kimeneti csatlakozói.

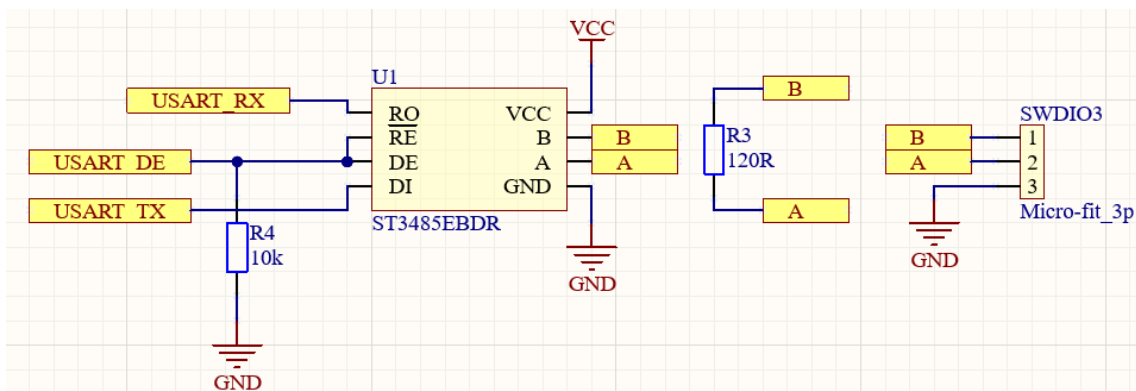
Az LTC6820HMS adatai:

- Maximum bemeneti feszültség: 6 V
- Maximum feszültségek a lábakon: 6 V
- Maximum áram lábakon
- IP IM: 30 mA
- MOSI, MISO, SCK, CS: 20 mA
- Működési hőmérséklet: -40 °C és 125 °C között
- Ultra alacsony áramfogyasztás idle állapotban 2  $\mu$ A[21]

A HM2101NLT gondoskodik a galvanikus leválasztásról, így biztonságban tartva a rendszert esetleges szakadás esetén a másik oldalról érkező magas feszültségtől. Egyenáramból 4300 V-ig képes legalább 60 másodpercig védeni az áramkört míg váltakozó áramtól 2500V-ig.[27]

### 7.2.5 Az RS-485 adóvevője

A fejlesztőkártyán az RS-485-ös kommunikációs közeget UART továbbítására lehet majd használni. A 30. ábrán látható ennek az összeköttetése.



30. ábra: RS-485 összeköttetése

Látható, hogy a mikrovezérlő oldaláról jövő USART\_RX (vevő) a Receiver outputba van csatlakoztatva, ugyanis itt jön ki az adóvevőből az üzenet és megy tovább a mikrovezérlőbe. A másik fontos csatlakozás ennek az ellentétje USART\_TX (adó) a Driver Inputba van csatlakoztatva, ugyanis az adóvevő az ideérkező adatokat küldi tovább a fizikai rétegre. A fizikai réteg az „A” és „B” nevű port amik egy 120 ohmos lezáró

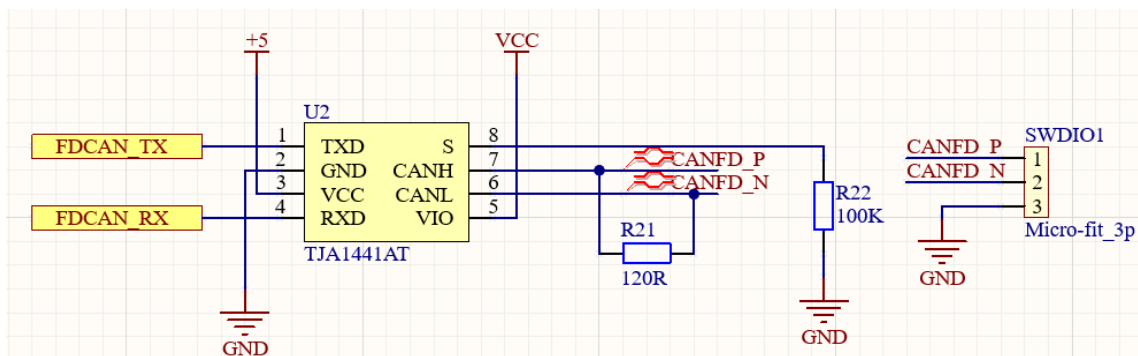
ellenállással vannak zárva és továbbítva vannak egy kimeneti csatlakozóra, amire más eszközöket lehet kötni. Az általam használt RS-485-ös adóvevő az ST3485EBDR.

Ennek az adatai:

- Működési hőmérséklet: -40 és 85 °C között
- Tápfeszültség: 3,3 V, egészen 7 V-ig
- Adatátviteli sebesség: 12Mbps
- Maximum kommunikációs állomások: 64 darab[28]

### 7.2.6 A CANFD adóvevője

A CANFD adóvevője a TJA1441AT cikkszámú chip, amelynek összeköttetését a 31. ábra



31. ábra: CANFD adóvevő bekötése

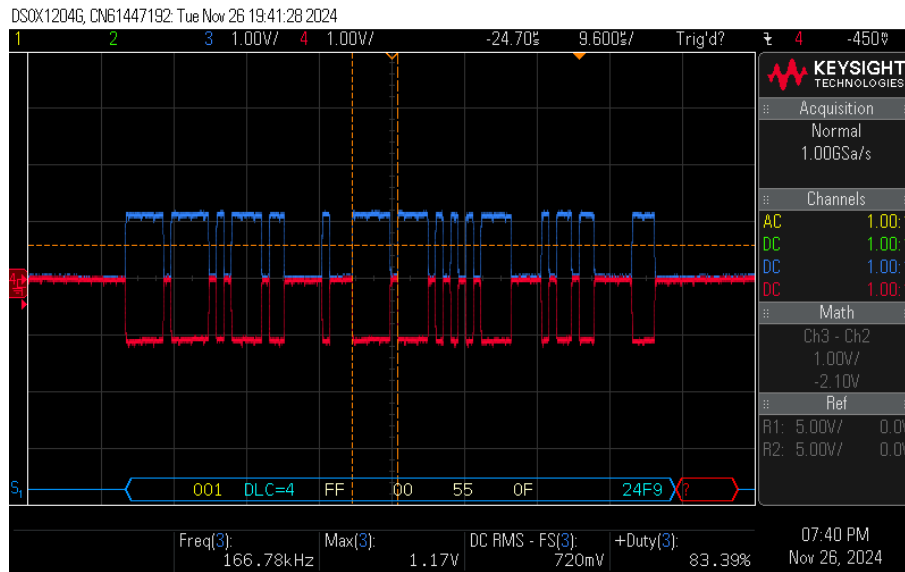
szemlélteti. A mikrovezérlőből érkező jelek csatlakoznak az adóvevő megfelelő bemenetébe vagy kimenetébe, ami azt átalakítva differenciális jellé továbbítódik a csatlakozón keresztül. A differenciális sodrott érpár itt is 120 ohmos ellenállással van lezárva. A TXD és RXD fogadja a jeleket a mikrovezérlőtől, ami már egy CAN frame alapú bitsorozat és ezt alakítja át az áramkör a differenciális jel alakká. Ezt a jelet a CANH és CANL porton keresztül küldi ki a vezetékekre és a csatlakozóra.

A TJA1441AT adatai:

- Bementi feszültség: 4,5-5,5 V
- Bementi áram: 38 mA, maximum: 60 mA
- Maximum ESD védelem: 8 kV
- Működési hőmérséklet: -40 és 150 °C között[29]

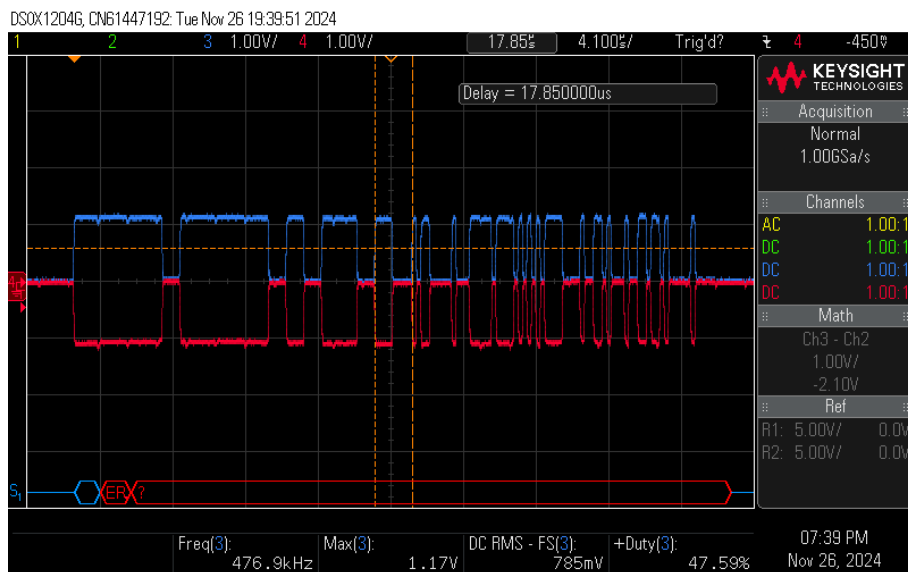
A CANFD kommunikációt egy prototípus kártya segítségével már sikerült tesztelni és oszcilloszkóppal megmérni rajta a CAN fizikai rétegen áthaladó információt. A két oszcilloszkópos mérés során jól látható a különbség a CAN és CANFD között. Az átviteli





32. ábra: A CAN oszcilloszkópos mérése

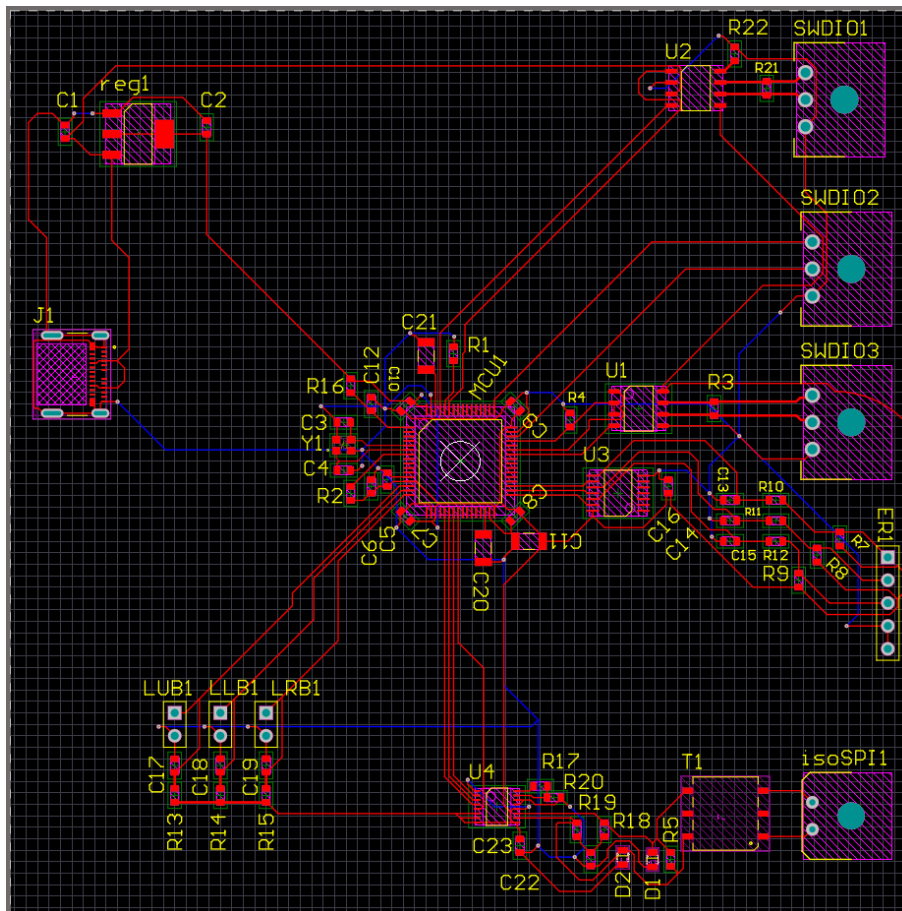
sebesség 1 Mbps -ra volt állítva, illetve CAN FD esetében az adatbitek átviteli sebessége 4Mbps-ra lett állítva. A 32. ábra a CAN átvitelt ábrázolja, míg a 33. ábra a CAN FD átvitelét. A két kép között jól látható, hogy a CAN FD esetében az adatbitek közlő



33. ábra: A CAN FD oszcilloszkópos mérése

résznél a jelek mennyire összesűrűsödnek.

### 7.3 Az áramkör fizikai felépítése



34. ábra. A fejlesztőkártya PCB terve

A 34. ábrán látható a PCB terv, amin látszódik mi merre lett helyezve. A J1 nevű eszköz például az USB-C csatlakozó. Jobboldat találhatóak a kommunikáció adóvevői és a közelükben a hozzájuk tartozó csatlakozók, ellenállások és kondenzátorok. Az áramkör tervezésénél kiemelt figyelmet fordítottam arra, hogy a differenciális vezetékek egymás mellett és ugyan olyan hosszúsággal legyenek meghúzva csatlakozóik irányába. A kondenzátorok, amik a feszültség egyenesítésére szolgálnak közel vannak a mikrovezérlőhöz, és szintúgy a kristály oszcillátor, hogy a jele mihamarabb eljusson, a lehető legkevesebb zavarral találkozva a mikrovezérlőbe. Jobb oldalt fent U2 néven van a CAN FD adóvevője, alatta U1 jelzéssel aRS-485 meghajtója és legalul található U4 néven az ISO SPI meghajtója, aminek a jele a T1 nevű transzformátorba megy, majd innen a csatlakozóba. A tápellátásért felelős LDO a bal felső sarokban található.

## 8 Összegzés

A szakdolgozatomban az Electric Racing Miskolc számára kellett egy fejlesztőkártyát terveznem, amely képes az autó különböző részeit tesztelni. Az autó részei lépésekben külön-külön készülnek el ezért egy fejlesztő kártya, ami ezekkel képes kommunikálni, nagy segítséget nyújt az autó elkészítése során. A fejlesztőkártyát ezen kívül lehet oktatásra is használni, ugyanis az minden részével tud kommunikálni a versenyautónak, amely így lehetőséget nyújt arra, hogy az emberek az adott programokat először erre a teszt kártyára írják meg, és ne az autó valós részein gyakoroljanak. A fejlesztőkártyán összesen négy fajta kommunikáció található, amelyek közül a CAN és a CAN FD a legfontosabb, ugyanis mint az autópárhuzamban is, mi is ezt a kommunikációt használjuk a legtöbb esetben. Azonban az autóban vannak kivételek ilyen például az akkumulátor, aminek biztonsági okokból galvanikusan le kell lennie választva ezért ez ISO SPI-on keresztül kommunikál. Továbbá van egy kormány elektronikánk, ami UART segítségével RS-485-ön keresztül küldi az információkat. Ezeket a kommunikációkat mindet rá kellett szerelnem a kártyára, hogy a célnak megfeleljen az. A kommunikációk mellé még találni kellett egy megfelelő mikrovezérlőt, ami képes volt akár az összeset egyszerre kezelni. A CAN FD miatt egy pontosabb kristály oszcillátort is fel kellett szerelni a nyomtatott áramkörre. Miután a tervek megvoltam elkezdődhetett a tervezés folyamata. Kezdve a mikrovezérlő lábainak kiosztásával, majd azok átrendezésével folytatva. A tervezés során ügyelni kellett arra, hogy legyenek a lábak össze-vissza. A nyomtatott áramkör elemeinek összekötése során figyelni kellett arra, hogy a nyomtatott áramkörön, a vezetősávok mindenhol össze legyenek kötve és ne kanyarodjanak túl éles szögben. Alapos figyelmet igényelt továbbá az alkatrészek elrendezése, és a lábak megfelelő helyre kötése. A tervezés vége felé még előjöttek fejlesztési lehetőségek, amiket a jövőben nagy valószínűséggel meg fogok valósítani. Fejlesztési lehetőségek közül olyanok jutottak eszembe, mint például: gombok, vagy ledek ráhelyezése még a fejlesztőkártyára, ezzel még jobban kihasználva annak lábait és segítve az oktatást, ilyen egyszerűen vezérelhető dolgokkal. Továbbá kerülhetnének rá csak simán kivezetések a mikrovezérlőből, amire tetszés szerint lehetne perifériákat csatlakoztatni.

## 9 Idegen nyelvű összefoglalás

In my thesis, I had to design a development board for Electric Racing Miskolc that can test different parts of the car. The parts of the car are built separately in steps, so a development board that can communicate with them is a great help when building the formula car. The development board can also be used for education, as it can communicate with all parts of the race car, which allows people to first write the given programs on this test board, and they do not have to practice on the real parts of the car. The development board has a total of four types of communication. Out of the four communication CAN and CAN FD are the most important, because, as in the automotive industry, we use this communication in most of our electronic parts. However, there are exceptions in the car, such as the battery, which must be galvanically isolated for safety reasons, so it communicates with ISO SPI. We also have a steering wheel electronics that uses UART to send information with an RS-485 transceiver. I had to mount all of these communications on the board to make it fit the purpose. In addition to the communications, I also had to find a suitable microcontroller that could handle all of them at once. Because of CAN FD, a more accurate crystal oscillator had to be mounted on the printed circuit board. Once I had the plans, the design process could begin. Starting with the allocation of the microcontroller's pins, then continuing with their rearrangement. During the design, I had to make sure that the pins were connected together. When connecting the elements of the printed circuit, I had to make sure that the wires were connected everywhere on the printed circuit board and did not bend at too sharp of an angle. The arrangement of the components and the connection of the pins in the right places also required careful attention. Towards the end of the design, development opportunities emerged, which I will most likely implement in the future. Development options included adding buttons or LEDs to the development board, making better use of its pins and helping with education with such easily controllable things. Additionally, it could simply have pins coming to the side of the board from the microcontroller, to which peripherals could be connected as desired.

## 10 Irodalomjegyzék

- [1] Formula Student Germany, “FSG: Concept,” <https://www.formulastudent.de/about/concept/>.
- [2] Formula Student Germany, “FSG: Disciplines,” <https://www.formulastudent.de/about/disciplines/>.
- [3] “FS-Rules\_2025\_v1.0”.
- [4] “EMRAX\_228\_datasheet\_v1.5”.
- [5] Hans-Paul-Kaysser-Straße, “M A N U A L Digital battery-motor controller BAMOCAR-PG-D3-400/400 BAMOCAR-PG-D3-700/250 BAMOCAR-PG-D3-700/400 for EC servo motor for AC asynchronous servo motor.” [Online]. Available: [www.unitek.eu](http://www.unitek.eu)
- [6] Melasta, “Product Specification Polymer Li-ion Battery 3.7V 6600mAh 15C Melasta reserves the right to alter or amend the design, model and specification without prior notice 2 Content,” 2017.
- [7] “Headline Insulation monitoring device (IMD) for unearthed DC drive systems (IT systems) in electric vehicles.”
- [8] R. de Andrade, M. M. D. Santos, J. F. Justo, L. R. Yoshioka, H.-J. Hof, and J. H. Kleinschmidt, “Security architecture for automotive communication networks with CAN FD,” *Comput Secur*, vol. 129, p. 103203, Jun. 2023, doi: 10.1016/j.cose.2023.103203.
- [9] Lenovo US, “What is a Bus Topology & How Does it Work?,” <https://www.lenovo.com/us/en/glossary/bus-topology/>.
- [10] S. Corrigan HPL, “Application Report Introduction to the Controller Area Network (CAN),” 2002. [Online]. Available: [www.bosch.com](http://www.bosch.com)
- [11] “BOSCH CAN Specification Version 2.0,” 1991.
- [12] Dr. Ajtonyi István, *Ipari Kommunikációs Rendszerek I*.
- [13] “CAN with Flexible Data-Rate Specification Version 1.0,” 2012.
- [14] “Website: [www.lin-subbus.org](http://www.lin-subbus.org) LIN Specification Package Revision 2.2A Specification Package Revision 2.2A,” 2010. [Online]. Available: [www.lin-subbus.org](http://www.lin-subbus.org)
- [15] E. Hackett, “LIN Protocol and Physical Layer Requirements,” 2022. [Online]. Available: [www.ti.com](http://www.ti.com)

- [16] Josh Schneider, “Microcontrollers vs. microprocessors: What’s the difference?,” <https://www.ibm.com/think/topics/microcontroller-vs-microprocessor>.
- [17] RS, “A Complete Guide to Microcontrollers,” <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/microcontrollers-guide>.
- [18] P. Dhaker and A. Devices, “Introduction to SPI Interface.”
- [19] “AN-0991\_Using\_the\_Serial\_Peripheral\_Interface\_to\_Communicate\_between\_Multiple\_Microcomputers (1)”.
- [20] M. Sparkfun, “Serial Peripheral Interface (SPI),” <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all#introduction>.
- [21] A. Devices, “LTC6820 (Rev. C).” [Online]. Available: [www.analog.com](http://www.analog.com)
- [22] “Relazione+RS485”.
- [23] K. Wang and H. Liu, “RS-485 Basics Series,” 2021. [Online]. Available: [www.ti.com](http://www.ti.com)
- [24] E. Peña and M. G. Legaspi, “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,” 2020.
- [25] “This is information on a product in full production. STM32H562xx and STM32H563xx Arm® Cortex®-M33 32-bit MCU +TrustZone® + FPU, 375 DMIPS, 250 MHz, 2-Mbyte flash, 640-Kbyte RAM, math accelerators Core • Arm® Cortex®-M33 CPU with TrustZone®, FPU, frequency up to 250 MHz, MPU, 375 DMIPS (Dhrystone 2.1),” 2024. [Online]. Available: [www.st.com](http://www.st.com)
- [26] “This is information on a product in full production,” 2020. [Online]. Available: [www.st.com](http://www.st.com)
- [27] “Mouser Electronics.”
- [28] “st3485eb-1850066”.
- [29] “TJA1441 High-speed CAN transceiver.”

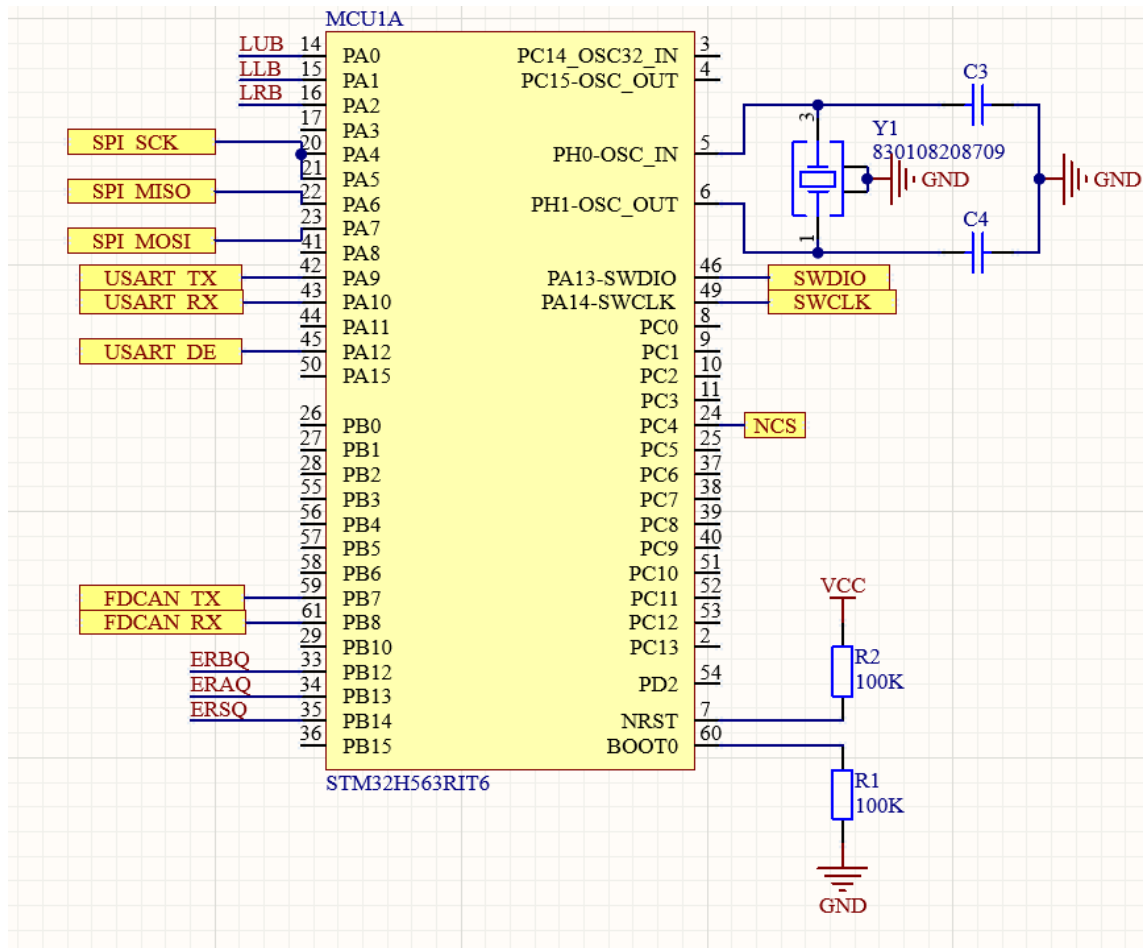
## 11 Ábrajegyzék

1. ábra: Skid Pad versenyszám pályája[3] .....	5
2. ábra: Emrax 228 .....	7
3. ábra: UniTek BAMOCAR D3 .....	8
4. ábra: Az autó akkumulátora .....	8
5. ábra: Egy akkumulátor szegmens .....	9
6. ábra: Standard CAN, 11 bites azonosítóval .....	11
7. ábra: Extended CAN, kiegészítve a 18 bites azonosítóval .....	12
8. ábra: A bitek és feszültség különbségek kapcsolata.....	13
9. ábra: Az üzenet prioritásának eldöntése: Az arbitráció.....	14
10. ábra: A CAN vezetékeinek feszültség értékei, és bitértékei .....	15
11. ábra: CAN és CAN FD közötti különbségek standard üzenetformában.....	16
12. ábra: CAN és CAN FD közötti különbség extended CAN esetében .....	17
13. ábra: A LIN üzenetkeret felépítése.....	18
14. ábra: A LIN feszültségszintjei és azok logikai értékei .....	20
15. ábra: A csatlakozások main és subnode között .....	22
16. ábra: SPI modes .....	23
17. ábra: Általános csatlakozás a master és slave eszközök között .....	24
18. ábra: SPI daisy chain.....	25
19. ábra: ISO SPI felépítése .....	26
20. ábra: RS-485 adóvevő.....	27
21. ábra: RS-485 half-duplex összeköttetés.....	28
22. ábra: RS-485 full-duplex csatlakoztatása .....	28
23. ábra: UART csomag felépítése .....	29
24. ábra: Mikrovezérlő láb kiosztásai .....	31
25. ábra: Az egész schematic .....	33
26. ábra: USB-C csatlakozó.....	34
27. ábra: LD1117 LDO .....	35
28. ábra: A mikrovezérlő tápja.....	36
29. ábra: ISO SPI bekötése .....	36
30. ábra: RS-485 összeköttetése .....	37
31. ábra: CANFD adóvevő bekötése .....	38
32. ábra: A CAN oszcilloszkópos mérése.....	39

33. ábra: A CAN FD oszcilloszkópos mérése.....	39
34. ábra. A fejlesztőkártya PCB terve.....	40



## 12 Mellékletek



1. Melléklet