

Debreceni Egyetem

Természettudományi és Technológiai Kar

Fizikai Intézet

Villamosmérnöki Tanszék

Villamosmérnöki BSc szak

TELJESÍTMÉNYELEKTRONIKAI DEMONSTRÁCIÓS ESZKÖZ MEGTERVEZÉSE ÉS MEGVALÓSÍTÁSA

Elleh Emily Chioma
2024.

Tartalomjegyzék

Bevezetés	2
1. A szinuszos inverterek és a teljesítménykapcsoló elemek általános felépítése	3
1.1 Az inverterek felépítése és működése	3
1.1.1 Az egyfázisú inverterek	4
1.1.2 A háromfázisú inverterek	7
1.2 A teljesítménykapcsoló elemek működése.....	8
2. A szinuszos impulzusszélesség modulációs technikák	11
2.1 A hordozó-alapú szinuszos impulzusszélesség moduláció	12
2.2 A mikrokontroller alapú szinuszosan modulált PWM	14
2.3 A H-híd vezérlése	16
2.4 A háromfázisú rendszerek vezérlése	17
3. Az eszköz fizikai felépítésének és vezérlésének megtervezése	18
3.1 A rendszer blokkvázlata	18
3.2 A rendszer hardveres építőelemeinek kiválasztása	20
3.3 A rendszer működésének teszteléséhez szimulációk készítése	29
4. Az eszköz hardverének megvalósítása	33
4.1 A félhíd modulok nyomtatott áramkörének megtervezése.....	33
4.2 A vezérlőközpont nyomtatott áramkörének megtervezése.....	36
5. A szoftveres vezérlés megtervezése	40
5.1 A kezdőkép elkészítése.....	41
5.2 A szinuszos moduláció létrehozásának lépései	42
5.3 IFB, VFB és TFB megvalósítása.....	50
5.4 A demonstrációs eszköz tesztelése	51
5.4.1 Az SPWM és a vezérlési módok tesztelése	51
5.4.2 A holtidő tesztelése.....	54
Összefoglalás	55
Irodalomjegyzék	56
Mellékletek	59

Bevezetés

Az energia, azon belül is a villamos energia, a mindennapi emberi tevékenységeink alapvető szükséglete. Elengedhetetlen feltétele a háztartási eszközeink működtetésének, ipari folyamataink fenntartásának, és sok más tevékenységnek egyaránt. Az emberiség energiaigénye napról napra nő, így az elektromos energia előállításának, szállításának és átalakításának szerepe is folyamatosan emelkedik. Az energiaátalakítás egyik legfőbb eszköze az inverter, amely lehetővé teszi az egyenáramú energia váltakozó áramú energiává történő átalakítását. Az inverterek szerepe kulcsfontosságú a megújuló energiaforrások (mint pl. a napelemek) alkalmazásában, valamint a motorok, szünetmentes tápegységek és elektromos járművek üzemeltetésében is. Ezeknek a DC/AC konvertáló rendszereknek a működésére több lehetséges topológiát és vezérlési módszert is kidolgoztak, melyeknek mind megvannak a sajátos előnyeik és hátrányaik. Fontos paraméterek a hatékonyság és a harmonikus torzítás, de különbséget találhatunk a létrehozható fázisok számában és azok feszültségfüggvényeinek minőségében is.

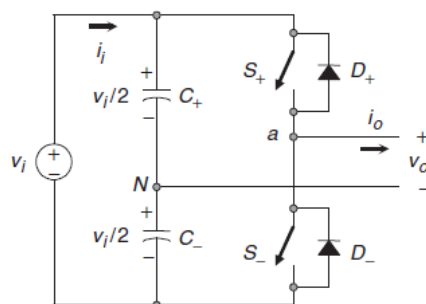
Mivel az inverterek fő célja az energiaátalakítás, nagyáramú és nagyfeszültségű alkalmazásokra vannak tervezve. Felhasználásuktól függően többszáz Volt feszültség és többszáz Amper áram is megfordulhat rajtuk, így nem feltétlenül alkalmasak arra, hogy kontrollált labori körülmények között használják fel őket. A felépítésük általában zárt és komplex, emiatt a belső szerkezeteikkel és vezérlésükkel való megismerkedés sem egyszerű.

A szakdolgozatom célja, hogy egy olyan eszközt hozzak létre, amely számos struktúrát és vezérlési lehetőséget bemutat a különféle inverterekre kifejlesztett technikákból. A rendszert oktatási célra tervezem, emiatt minél több részét és paraméterét szabályozhatóvá szeretném tenni, hogy megfelelően átlátható legyen a felhasználók számára, mely egységek milyen kimenőjelekért felelősek. Az oktatási célt az eszközömmön sok mérőpont elhelyezésével, és egy használati útmutató készítésével is támogatom majd, hogy minél egyértelműbb legyen az eszköz működése. A célom nem egy valós nagyteljesítményű inverter megtervezése, hanem egy gyakorlati órák keretein belül is használható, labortápról is üzemeltethető demonstrációs eszköz létrehozása.

1. A szinuszos inverterek és a teljesítménykapcsoló elemek általános felépítése

1.1 Az inverterek felépítése és működése

Az inverterek legfőbb feladata, hogy a bemenetükre érkező egyenfeszültségből vagy egyenáramból, váltakozó feszültséget vagy váltakozó áramot hozzanak létre. Ezáltal lehetővé téve azt, hogy egyenáramú (DC) tápfeszültségről adjunk át teljesítményt, váltakozó áramot (AC) igénylő terhelésnek. Az inverterek rengeteg felhasználásai között megtaláljuk az ASD-ket (Adjustable Speed Motor Drives), és a szünetmentes tápegységeket is. A járművekben elhelyezett akkumulátorokban tárolt energia megfelelő átalakítását, és a fotovoltaiikus napelemes rendszerek által termelt DC feszültség konvertálását és hálózatra való visszatáplálását is inverterek segítségével valósítják meg. [2] [3] Az invertereket működésük és kimeneteik szerint több csoportba is sorolhatjuk: Bemenőjel típusa szerint megkülönböztetünk feszültségforrás invertereket (VSI, Voltage Source Inverter) és áramforrás invertereket (CSI, Current Source Inverter). A feszültségforrás inverterek esetében a vezérelhető paraméter az inverter kimenő feszültségének függvénye. Ez az elterjedtebb inverteres struktúra, ugyanis a legtöbb ipari alkalmazhatóságnak a feszültségforrásként való felhasználás a feltétele. [2] Kimenőjel típusa szerint a szinuszos kimenőjel a legelterjedtebb, de rendelkezhetnek négyszögjel kimenettel is. [1] A szinuszos inverterek kialakításánál ügyelnek arra, hogy változtatható paraméterként jelenjen meg a szinuszhullám amplitúdója, frekvenciája és fázisa is. [2] Felépítésüket tekintve, az inverterek három nagyobb egységre bonthatóak: DC vonalra, kondenzátorokra, és teljesítménykapcsoló elemekre.



1. ábra. Egyfázisú félhidas VSI felépítése [2]

A DC vonal nem más, mint az egyenfeszültség, amiből a rendszer AC feszültséget hoz létre (v_i). Közvetlenül ezután következnek az egymással sorba kötött, azonos kapacitású, nagy energiatároló képességű kondenzátorok, melyek fő feladata a tápegység feszültségének szimmetrikus megfelelése és a kellő energia tárolása a rendszer számára. A kondenzátorok

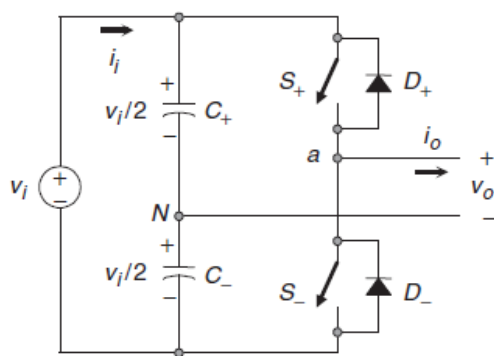
az inverter által generált felharmonikusok szűrésében is részt vesznek. A kapcsolóelemeket félvezető teljesítménykapcsolók alkalmazásával valósítják meg: leggyakrabban MOSFET vagy IGBT tranzisztorokkal. Ezeket a kapcsolóelemeket félhidas felépítésben helyezik el, egy ilyen félhidat az inverter ágának nevezünk. [2] A DC vonal pozitív oldalához közelebb elhelyezkedő kapcsolóelemet a felsőági (high-side) kapcsolóelemnek nevezzük, a földhöz közelebbit pedig alsóáginak (low-side). [16] A kapcsolóelemek vezérlésénél ügyelni kell arra, hogy soha ne legyen egy hídág mindkét tranzisztora egyszerre bekapcsolva, ugyanis ez hídzárlat kialakulásához vezet. Mivel a kapcsolók segítségével csak diszkrét feszültségértékeket tudunk előállítani, az inverterek kimenetein is diszkrét értékek, impulzusok jelennek meg. Ebből következik, hogy az inverterek gyorsan tudják változtatni a kimenetükön lévő jel értékét, azonban nem képesek ezeket a változásokat egyenletesen végrehajtani, így nagy dv/dt -vel rendelkeznek. [2]

1.1.1 Az egyfázisú inverterek

Az egyfázisú inverterek egyetlen fázis létrehozására képesek. Alkalmazási köreik inkább az alacsonyabb teljesítményigényű rendszerekre terjednek ki, mint a tápegységek és a fotovoltaiikus rendszerek. A hídágak számától függően egy fázis létrehozását két konfigurációban is meg tudjuk valósítani: félhidas és H-hidas felépítésben. A két topológia között a legnagyobb különbséget az általuk létrehozott kimenő feszültségjelekben láthatjuk. [2]

Az egyfázisú félhidas inverter

A félhidas felépítés a legegyszerűbb konfiguráció az inverterek struktúráját tekintve, mivel csak két kapcsolóelemet tartalmaz. Ezek a kapcsolóelemek egymással sorba vannak kötve, egyetlen hídágot képezve. Amint az a 2. ábrán is látható, a v_i DC vonal az inverter bemenő DC feszültsége. Ezzel párhuzamosan van elhelyezve két nagykapacitású kondenzátor, amelyek létrehozzák az N nullpontot, ezáltal megfelelve a v_i feszültséget $v_i/2$ -re. A fent említettek alapján a vezérléskor itt is figyelmet kell fordítani arra, hogy a hídágban lévő két kapcsolóelem ne legyen egyszerre bekapcsolt állapotban. Ha ez megtörténik, a DC vonal zárlata miatt nagy áram alakul ki a körben.



2. ábra. Egyfázisú félhidas VSI felépítése [2]

A rendszer vezérlésekor a hídzárlat elkerülésének érdekében előre definiált állapotokat különböztetünk meg. Az inverter a működése során ezen állapotok között lépked. A félhidas topológiában két definiált és egy nemdefiniált állapotot különböztetünk meg, amelyek az alábbiak:

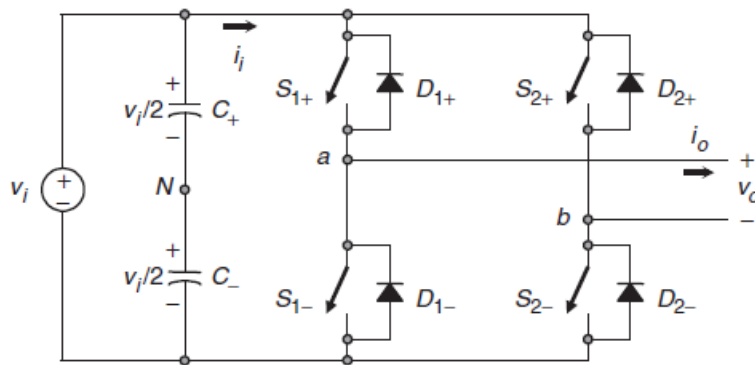
State	State #	v_o	Components conducting
S_+ is on and S_- is off	1	$v_i/2$	S_+ if $i_o > 0$ D_+ if $i_o < 0$
S_- is on and S_+ is off	2	$-v_i/2$	D_- if $i_o > 0$ S_- if $i_o < 0$
S_+ and S_- are all off	3	$-v_i/2$ $v_i/2$	D_- if $i_o > 0$ D_+ if $i_o < 0$

3. ábra. Egyfázisú félhidas VSI állapotai [2]

Ezen állapotok közül az első és a második alkalmazandó, a harmadik pedig elkerülendő (nem definiált). Fontos megfigyelnünk, hogy a félhidas topológiával elérhető maximális kimenő feszültségszint $\pm v_i/2$, tehát a betáplált DC feszültség fele. Az S_+ kapcsoló (2. ábra) bekapcsolt állapotában az a és N pontok között mérhető kimenő feszültség $v_i/2$, és a DC táp polaritásától függően az áram vagy magán az S_+ kapcsolóelemenen folyik keresztül, vagy annak a D_+ -al jelölt diódáján. S_- bekapcsolásakor a kimenő feszültség $-v_i/2$, és az áram vagy magán a kapcsolóelemenen, vagy annak a D_- -al jelölt diódáján halad át. Amikor az egyik kapcsoló be van kapcsolva, ugyanabban az időpillanatban a másik kapcsoló kikapcsolt állapotra van kényszerítve. [2]

Az egyfázisú H-hidas inverterek

Az általam tervezett demonstrációs eszköz egyik legfontosabb feladata a H-hidas szerkezet bemutatása, annak széleskörű alkalmazása miatt. Sok tulajdonságában hasonló a félhidas rokonához, azonban kettő helyett négy kapcsolóelemet tartalmaz, amelyek két hídágat alkotnak a rendszerben. A félhidas felépítéssel ellentétben, ebben a kapcsolásban a második hídág biztosítja a nullpontot a kimenetre csatlakoztatott terhelés számára, nem pedig a DC vonalat kettéosztó kondenzátorok felezőpontja. Ugyanúgy, mint a félhidas felépítésben, az egy ágban lévő kapcsolóelemek itt sem lehetnek egyszerre bekapcsolva, vagy a DC vonal zárlata következhet be a rendszerben. [2]



4. ábra. A H-hidas VSI felépítése [2]

Ebben a felépítésben az inverter működése során öt állapotot különböztetünk meg (5. ábra): négy definiált (1, 2, 3, 4 állapotok) és egy nemdefiniált állapotot (5. állapot).

State	State #	v_{aN}	v_{bN}	v_o	Components conducting
S_{1+} and S_{2-} are on and S_{1-} and S_{2+} are off	1	$v_i/2$	$-v_i/2$	v_i	S_{1+} and S_{2-} if $i_o > 0$ D_{1+} and D_{2-} if $i_o < 0$
S_{1-} and S_{2+} are on and S_{1+} and S_{2-} are off	2	$-v_i/2$	$v_i/2$	$-v_i$	D_{1-} and D_{2+} if $i_o > 0$ S_{1-} and S_{2+} if $i_o < 0$
S_{1+} and S_{2+} are on and S_{1-} and S_{2-} are off	3	$v_i/2$	$v_i/2$	0	S_{1+} and D_{2+} if $i_o > 0$ D_{1+} and S_{2+} if $i_o < 0$
S_{1-} and S_{2-} are on and S_{1+} and S_{2+} are off	4	$-v_i/2$	$-v_i/2$	0	D_{1-} and S_{2-} if $i_o > 0$ S_{1-} and D_{2-} if $i_o < 0$
S_{1-} , S_{2-} , S_{1+} , and S_{2+} are all off	5	$-v_i/2$ $v_i/2$	$v_i/2$ $-v_i/2$	v_i $-v_i$	D_{1-} and D_{2+} if $i_o > 0$ D_{1+} and D_{2-} if $i_o < 0$

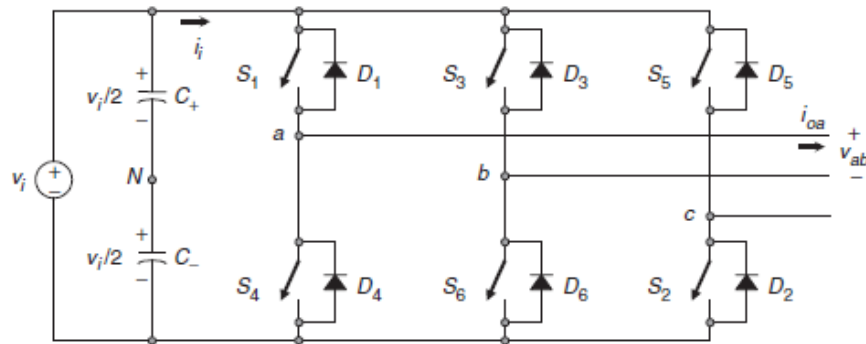
5. ábra. A H-hidas egyfázisú VSI lehetséges állapotai [2]

A nemdefiniált 5. állapot elkerülése fontos, mert így tudjuk biztosítani azt, hogy az inverter kimenetén lévő AC feszültség értéke mindig meghatározható. Fontos előny a félhidas topológiával szemben, hogy H-hidas kapcsolás esetén az elérhető maximális kimenő feszültség szint v_i , tehát a félhíd által létrehozott maximális kimenő feszültség kétszerese. A

H-hidas kapcsolás lehetővé teszi a nulla kimenő feszültség előállítását is (3. és 4. állapot). [2]

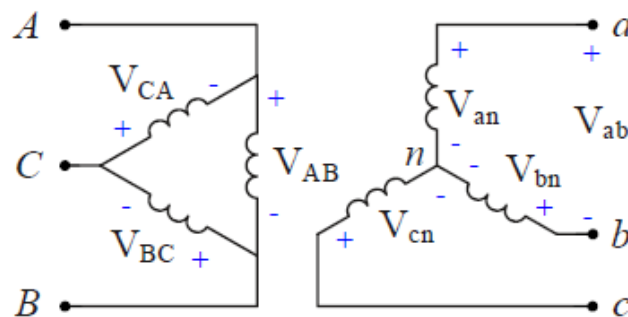
1.1.2 A háromfázisú inverterek

A háromfázisú invertereket főleg a közép- és nagyteljesítményű rendszerekben használják fel. Ezen inverterek fő feladata, hogy háromfázisú feszültségforrást hozzanak létre a rendelkezésre álló DC vonalból. A legtöbb háromfázisú rendszerben a szinuszos feszültségjelek között 120 fokos fáziskülönbségre van szükség, így itt is a cél a 120 fokkal eltolt feszültségfüggvények létrehozása a kimeneten. E jelek megvalósítását úgy érik el, hogy kettő helyett három hídágat alkalmaznak az inverterben. Minden hídág egy fázis létrehozásáért felelős, és a kimenő fázisokat az a , b és c pontokról vehetjük le (6. ábra). [2]



6. ábra. Egy háromfázisú VSI felépítése [2]

Ez a felépítés csillagkapcsolt terhelések esetén lehetővé teszi fázis- és vonali feszültségek megfigyelését is. A fázisfeszültségek a fázis és a csillagkapcsolás n középpontja között mérhetők (V_{an} , V_{bn} , V_{cn}), a vonali feszültségek pedig az egyes fázisok között (V_{ab} , V_{ac} , V_{cb}). [9] A terhelések delta kapcsolása esetén a fázisfeszültségek megegyeznek a vonali feszültségekkel, így itt nem találunk különbséget a két kimenőjel között. [13]



7. ábra. A terhelések delta (bal) és csillagkapcsolása (jobb) [13]

A háromfázisú invertereknek nyolc definiált állapota van, amelyek közül a hetedik és a nyolcadik a 0 Volt kimenő feszültség létrehozását biztosítja (8. ábra). Ilyenkor az áram, vagy csak az alsó, vagy csak a felső komponenseken folyik körbe (freewheeling). Ebben a felépítésben a hídágat alkotó kapcsolóelemek közül nem kapcsolhatunk le egyet sem a többivel egyszerre, mivel az nem meghatározható kimenő feszültségfüggvényekhez vezethet. A vonali feszültségek $+v_i$, $-v_i$ és 0 Volt értékeket vehetnek fel. [2]

State	State #	v_{ab}	v_{bc}	v_{ca}	Space vector
S_1, S_2 , and S_6 are on and S_4, S_5 , and S_3 are off	1	v_i	0	$-v_i$	$\vec{v}_1 = 1 + j0.577$
S_2, S_3 , and S_1 are on and S_5, S_6 , and S_4 are off	2	0	v_i	$-v_i$	$\vec{v}_2 = j1.155$
S_3, S_4 , and S_2 are on and S_6, S_1 , and S_5 are off	3	$-v_i$	v_i	0	$\vec{v}_3 = -1 + j0.577$
S_4, S_5 , and S_3 are on and S_1, S_2 , and S_6 are off	4	$-v_i$	0	v_i	$\vec{v}_4 = -1 - j0.577$
S_5, S_6 , and S_4 are on and S_2, S_3 , and S_1 are off	5	0	$-v_i$	v_i	$\vec{v}_5 = -j1.155$
S_6, S_1 , and S_5 are on and S_3, S_4 , and S_2 are off	6	v_i	$-v_i$	0	$\vec{v}_6 = 1 - j0.577$
S_1, S_3 , and S_5 are on and S_4, S_6 , and S_2 are off	7	0	0	0	$\vec{v}_7 = 0$
S_4, S_6 , and S_2 are on and S_1, S_3 , and S_5 are off	8	0	0	0	$\vec{v}_8 = 0$

8. ábra. Háromfázisú VSI lehetséges állapotai [2]

1.2 A teljesítménycapcsoló elemek működése

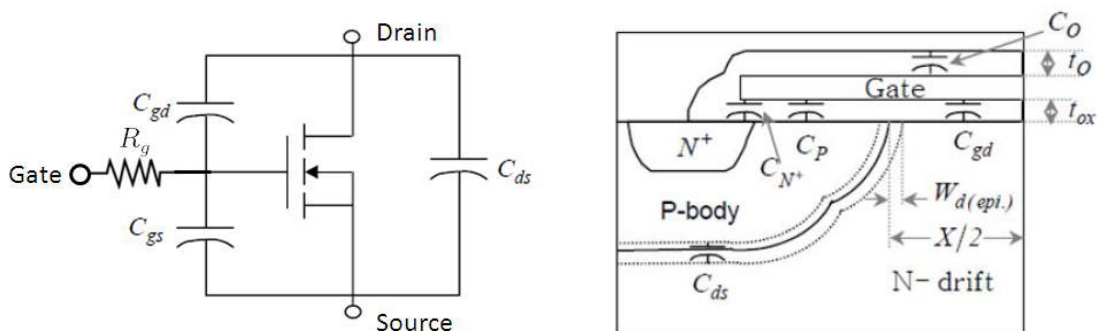
Az inverterek felépítésében a teljesítménycapcsoló elemek a legfontosabb egységek. [11] Ideális esetben, amikor egy kapcsoló bekapcsolt állapotban van, nulla feszültségeséssel rendelkezik, kikapcsolt esetben pedig teljesen megszakítja az áramkört. Emiatt az energia áramlásának irányítását energiaveszteség nélkül képes végrehajtani a rendszerben. Ez azonban valós kapcsolóelemeknél nem teljesül, különféle veszteségekkel jár a működtetésük.

A teljesítményelektronikában alkalmazott kapcsolóelemek félvezetőkből állnak: leggyakrabban MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor), vagy IGBT (Insulated Gate Bipolar Transistor) tranzisztorokból, azok feszültségvezérlésen alapuló működése miatt. Ezeket az elemeket a feszültség, áram- és a teljesítmény határadataik alapján választják ki. [2] Az IGBT tranzisztorokat nagyfeszültségű (>2kV) és nagyáramú (>200 A) rendszerekben alkalmazzák, míg a MOSFET tranzisztorok felhasználása főleg 1kV - 40A alá korlátozódik. [10] Mivel a MOSFET tranzisztorok magasabb frekvenciákon

képesek kapcsolni (még 200 KHz-nél is nagyobb frekvenciákon, ezzel szemben az IGBT-k kapcsolási frekvenciája 25 KHz alatti), és az általam tervezett eszközhöz elegendő az alacsonyabb teljesítménytolerancia, ebben a szakaszban a teljesítmény MOSFET tranzisztorok működésére térek ki. [11]

A diszkrét MOSFET-ek általában három lábkivezetéssel rendelkeznek: drain, gate és source elektródákkal. A kapcsolóelem bekapcsolását a gate elektródára helyezett 5 Volt körüli feszültséggel tudjuk elérni, de ez a feszültségérték változhat a MOSFET kialakításának függvényében. A gate feszültséget mindig a source elektródához képest kell a kapcsolóelemre kapcsolni. A logikai-szintű (logic-level) MOSFET-ek már egy mikrokontroller segítségével is teljesen bekapcsolhatóak, míg egy nem logikai-szintű FET-nek 10 - 15 Voltra is szüksége lehet a megfelelő működéshez. [14] Az áram az n-csatornás MOSFET kapcsolókban a drain elektróda felől halad a source kivezetés felé.

A teljesítmény MOSFET-ek felépítése általában vertikális struktúrát követ: a drain kivezetés nem a gate elektróda mellett helyezkedik el, hanem az alkatrész alján. Emiatt a kapcsolóelemen keresztülfolyó áram is függőlegesen halad, amivel magasabb feszültségtolerancia érhető el. Működtetésük szempontjából, a ki- és bekapcsolási karakterisztikák a legfontosabb függvények. Erre a legnagyobb hatást a parazitakapacitások gyakorolják, amelyeket gate-source (C_{GS}), drain-source (C_{DS}) és gate-drain (C_{GD}) kapacitásokra tudunk lebontani. Ahhoz, hogy a FET bekapcsolt állapotba kerüljön, először ezeknek a parazitakapacitásoknak kell feltöltődniük, majd kikapcsoláskor kisülniük. [16]

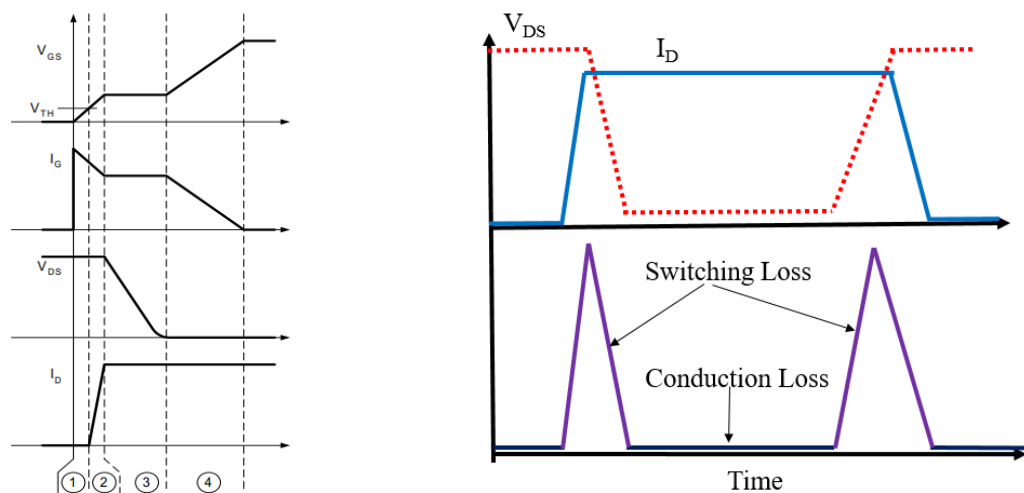


9. ábra. MOSFET parazitakapacitásai és azok elhelyezkedése a rétegekben [16]

A kapcsolási karakterisztikák közül a legfontosabbak a V_{GS} (gate-source feszültség), I_G (gate áram), V_{DS} (drain-source feszültség) és I_D (drain-áram) változásainak függvényei.

A MOSFET bekapcsolását (és kikapcsolását) négy részre lehet bontani (10. ábra):

- **1:** Az első szakaszban a bemeneti kapacitás feltöltődik 0 Voltról V_{th} küszöbfeszültségig. Ilyenkor a legtöbb gate-áram a C_{GS} kapacitást tölti. Ebben a szakaszban megfigyelhetjük a bekapcsolási késleltetést is (turn-on delay), mivel sem a drain-feszültség, sem a drain-áram nem változik.
- **2:** Miután a gate-feszültség meghaladta a küszöbfeszültséget, a MOSFET képes áramot vezetni. Ez a működés lineáris szakasza, ugyanis a drain-áram arányosan nő a gate-feszültséggel. A gate-áram továbbra is C_{GS} és C_{GD} kapacitásokat tölti, de a V_{DS} feszültség itt még változatlan.
- **3:** A harmadik szakaszban a gate-feszültség eléri a $V_{GS_{Miller}}$ feszültség szintet, amely után egy darabig változatlan marad az értéke. Ez a feszültség szint a Miller plateau. A drain-feszültség elkezd csökkenni, azonban a drain-áram változatlan marad, mivel értékét már csak a külső áramkör határolja (szaturáció).
- **4:** Az utolsó szakaszban a kapcsolóelem teljes bekapcsolása történik meg, V_{GS} feszültség további növelésével. V_{GS} végső értéke határozza meg a drain-source ellenállás végértékét ($R_{DS(On)}$), amely a vezetési veszteségre gyakorol hatást (conduction loss, 10. ábra). [19]

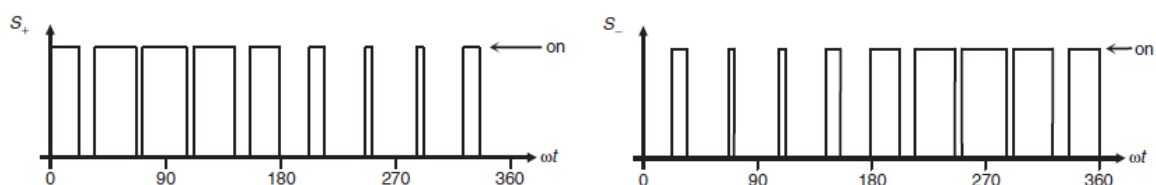


10. ábra. MOSFET bekapcsolási karakterisztikái (bal) [19] és a MOSFET kapcsolási, illetve vezetési vesztesége (jobb) [22]

A kikapcsolási karakterisztika függvényei hasonlóképpen épülnek fel, de a fent leírt függvények inverzeit alkotják. [19] Mivel a teljesítmény általános képlete $U \cdot I$ ($V_{DS} \cdot I_D$ a kapcsolóelem esetében), ha a képletben bármelyik együttható nulla, a veszteség is közel nulla. A MOSFET bekapcsolásánál a kapcsolási veszteséget akkor figyelhetjük meg, amikor a drain-áram már nem nulla, de a drain-feszültség még nem nulla (10. ábra). [20]

2. A szinuszos impulzusszélesség modulációs technikák

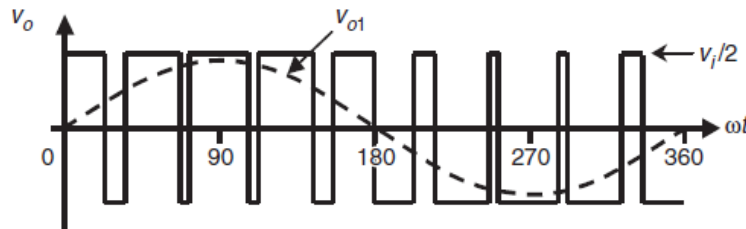
Az inverterek modulációs technikái közül a szinuszos impulzusszélesség moduláció csak egy lehetséges formája a vezérlésnek. Mivel az általam tervezett demonstrációs eszköz a szinuszos kimenőjel előállítására fekteti a hangsúlyt, ez a fejezet e technika bemutatását tartja szem előtt. Az inverterek vezérlése lényegében a kapcsolóelemek vezérlésén alapszik, ezért ezeknek a kapcsolóknak a ki- és bekapcsolási ideje határozza meg az eszköz által létrehozott kimenőjel formáját. A vezérlőjel és a kimenőjel közötti különbség, hogy a vezérlőjelnek logikai szerepe van, emiatt logikai 0 és 1 (a 11. ábrán „on”) között vesz fel értékeket. Ez a valóságban 5 Voltos logika mellett 0 esetén 0 Volt, 1 esetén 5 Volt kiadását jelenti. Gyakori technika az impulzusszélesség moduláció (Pulse Width Modulation) alkalmazása, ahol a kapcsolóelemekre érkező vezérlőjel különböző hosszúságú impulzusokból áll. Ebben a vezérlőjelben a létrehozni kívánt szinuszhullám információja úgy kerül továbbításra, hogy a teljesítménycapcsolók ki- és bekapcsolásának időtartamát folyamatosan változtatjuk, egy szinuszhullám által felvett értékeknek megfelelően. Ezekkel az időzített kapcsolgatásokkal a nagyobb frekvenciás vezérlőjelben létrehozunk egy szinuszos alaphfrekvenciát. Ezt a modulációt SPWM-nek is nevezik, ami a Szinuszos Impulzusszélesség Moduláció (Sinusoidal Pulse Width Modulation) rövidítése. A technika alapján, amikor a szinuszhullám első félperiódusát közvetítjük a kapcsolók felé ($[0, \pi]$), nő az impulzusok kitöltési tényezője. A második félhullámban, a hullám negatív értékeinél ($[\pi, 2\pi]$) csökkentjük a kitöltési tényezőket. A legnagyobb és legkisebb kitöltési tényezőket a szinuszhullám maximumánál (+1) és minimumánál (-1) figyelhetjük meg. A vezérlést a legegyszerűbben egy félhidas egyfázisú inverteren lehet bemutatni, mivel ez a struktúra csak két kapcsolóelemet tartalmaz (2. ábra). Az egy hídágban lévő kapcsolóelemek nem lehetnek egyszerre bekapcsolott állapotban, így a két kapcsolóra érkező vezérlőjel egymás logikai invertáltjainak felelnek meg (11. ábra). [2]



11. ábra.

S+ kapcsolóra érkező vezérlőjel (bal) és S- kapcsolóra érkező vezérlőjel (jobb) [2]

Az inverter kimenetén megjelenő feszültségfüggvény nem más, mint egy szinuszos függvény, amely felharmonikusokat tartalmaz. A kimenőjel (v_o) alakja leköveti S+ kapcsoló feszültségfüggvényét, és félhidas inverter esetén $+v_i/2$ és $-v_i/2$ értékeket tudja felvenni (12. ábra). [2]



12. ábra. Egyfázisú félhidas VSI kimenő feszültsége (v_o) [2]

A tiszta szinuszhullám ebből a szinuszosan modulált jelből egy aluláteresztő szűrő alkalmazásával nyerhető vissza, a nagyfrekvenciás komponens eltávolításával. Ehhez a leggyakrabban LC aluláteresztő szűrőket alkalmaznak. [2]

A felső és alsó hídagi kapcsolóelemek egyidejű bekapcsolása ellen hídzárlatvédelmet is be szoktak építeni a rendszerekbe. Ilyen lehet a holtidő bevezetése, amikor az impulzusok között olyan szakaszokat helyeznek el, amikor az alsóági és a felsőági kapcsolók is kikapcsolt állapotban vannak. [12]

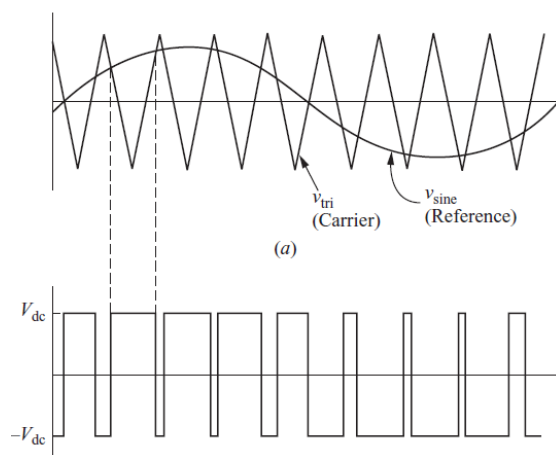
2.1 A hordozó-alapú szinuszos impulzusszélesség moduláció

A fent bemutatott szinuszosan változó vezérlőjel előállítására több módszert is kidolgoztak, de az egyik legegyszerűbb és leggyakoribb eljárás a hordozó-alapú szinuszos impulzusszélesség moduláció (Carrier-based SPWM) alkalmazása. Ebben a modulációs technikában két jelet különböztetünk meg: egy szinuszos moduláló jelet (v_{sine}) és egy háromszöggel alapú hordozó jelet ($v_{\text{triangle}}/v_{\text{carrier}}$). Az SPWM létrehozása a két jel értékeinek folyamatos összehasonlításával történik (akár egy analóg komparátor, vagy egy mikrokontroller segítségével), és amikor a modulálójel értékei nagyobbak, mint a hordozójel értékei, a vezérlőjel logikai 1-et vesz fel, ellenkező esetben pedig nullát. Logikai 1 hatására a hídág felső kapcsolóeleme kapcsol be (S+), logikai nulla hatására pedig az alsó kapcsolóelem (S-).

Összegezve (13. ábra):

Ha $v_{\text{sine}} > v_{\text{tri}}$, a vezérlőjel logikai értéke 1, S+ kapcsoló bekapcsol, S- kapcsoló kikapcsol.

Ha $v_{\text{sine}} < v_{\text{tri}}$, a vezérlőjel logikai értéke 0, S- kapcsoló bekapcsol, S+ kapcsoló kikapcsol.

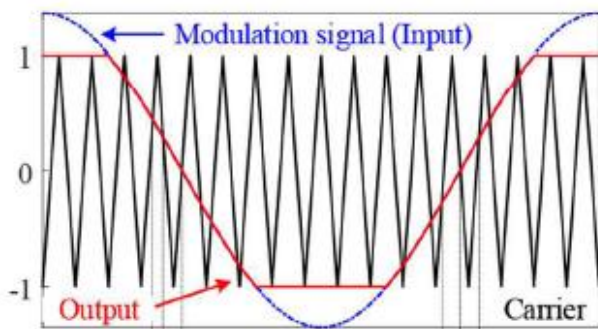


13. ábra. A hordozójel (v_{tri}), modulálójel (v_{sine}) és kimenőjel (v_o) függvényei [1]

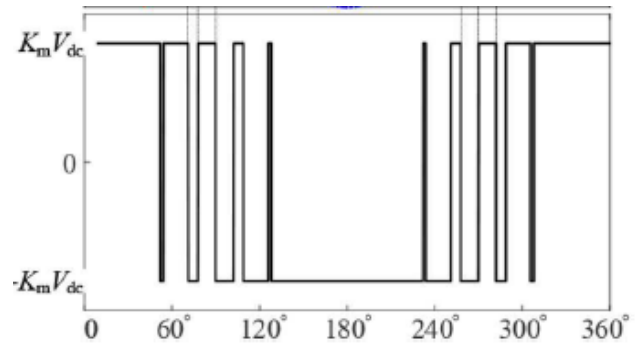
A hordozó-alapú SPWM-ben két paraméterrel jellemezzük a rendszert: az amplitúdó-modulációs aránnyal (m_a) és a frekvencia-modulációs aránnyal (m_f). Ebben a rendszerben m_a a hordozójel amplitúdójának (A_{tri}) és a modulálójel amplitúdójának (A_{sine}) hányadosaként számítható ki:

$$m_a = \frac{A_{sine}}{A_{tri}}.$$

A modulációban m_a értékeiből meg tudjuk határozni, hogy a szinuszos moduláció lineáris, vagy nemlineáris tartományában tartózkodunk-e. Amennyiben $m_a \leq 1$, a lineáris tartományban vagyunk, ellenkező esetben a nemlineáris régióban. Ennek a jelentősége, hogy ha a szinuszhullám amplitúdója túl nagy ($\frac{A_{sine}}{A_{tri}} > 1$), a hordozójellel történő összehasonlításakor kimaradnak metszéspontok, ami a vezérlőjel (és ezáltal a kimenőjel) torzulásához vezet. [2] A 14. ábrán látható, hogy a kimenet (pirossal jelölt, Output) egy vágott szinuszhullám alakját veszi fel. [5] Ez a túlmodulációs vezérlés, mellyel nagyobb amplitúdójú kimenőjel érhető el, de alkalmazása alacsonyrendű felharmonikusok behozatalával is jár. Ha az m_a nagyobb, mint 3.24, négyszögjel moduláció következik be.



14. ábra. A túlmodulációs vezérlés feszültségfüggvényei [5]



15. ábra. A túlmodulációval létrehozott kimenő feszültségjel [5]

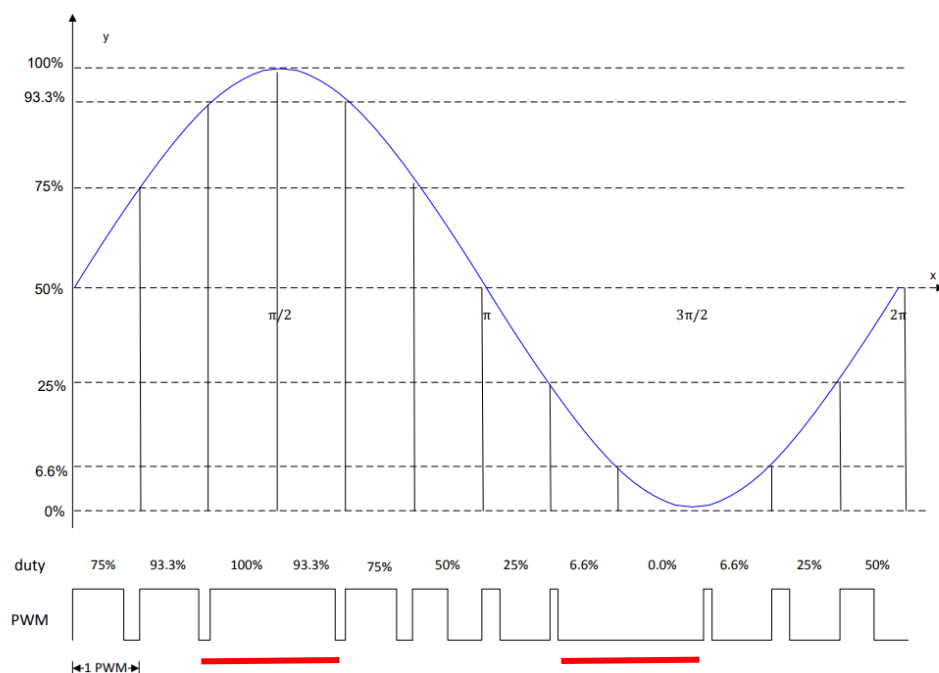
Ezzel szemben m_f a két jel frekvenciáinak hányadosaként van definiálva, így:

$$m_f = \frac{f_{tri}}{f_{sine}} \cdot [2]$$

Mivel a nagyfrekvenciás feszültségharmonikusokat könnyebb a kimenőjelből leszűrni, célszerű a kapcsolási frekvenciát (vagyis a hordozójel frekvenciáját) a lehető legnagyobbra választani. Ebből következik, hogy m_f nagyobb értékeinél a kimenőjel szűrése egyszerűbb, azonban a nagyobb kapcsolási frekvencia nagyobb kapcsolási veszteséggel is jár. [6]

2.2 A mikrokontroller alapú szinuszosan modulált PWM

A szinuszosan modulált impulzusokat analóg háromszög- és szinuszjelek nélkül is létre lehet hozni, mikrokontrollerek segítségével. Ebben a technikában a PWM jel, amit a mikrokontroller generál, állandó frekvenciájú, és csak a PWM impulzusok kitöltési tényezőit változtatjuk szoftveresen (0-100%-ig). A módszer alapja, hogy először digitalizáljuk a szinuszhullámot, mintavételezés segítségével. Mivel a PWM frekvenciája állandó, a mintákat egyenlő időközönként kell venni a hullámból. A szinuszhullám -1 és +1 között vesz fel értékeket, ez azonban nem megfelelő a kitöltési tényezők beállítására, mivel azok csak pozitív egész számok lehetnek. Emiatt a mintákat 0 és 1 közé normalizálják. A kitöltési tényezők a szinuszhullám amplitúdóinformációit hordozzák. A 16. ábrán látható, hogy mivel a hullám amplitúdója 0 és 100% között van értelmezve, kimaradnak impulzusok a maximumnál és a minimumnál, a túlmodulációhoz hasonlóan (pirossal aláhúzott). [7]



16. ábra. Mintavételezett szinuszhullám (12 minta/hullám), és az értékeiből előállított SPWM [7]

Degree	Radian	sine(Radian)	Normalize to 0	%
0	0.0000	0.0000	1.0000	0.50
30	0.5233	0.4998	1.4998	0.75
60	1.0467	0.8658	1.8658	0.93
90	1.5700	1.0000	2.0000	1.00
120	2.0933	0.8666	1.8666	0.93
150	2.6167	0.5011	1.5011	0.75
180	3.1400	0.0016	1.0016	0.50
210	3.6633	-0.4984	0.5016	0.25
240	4.1867	-0.8650	0.1350	0.07
270	4.7100	-1.0000	0.0000	0.00
300	5.2333	-0.8673	0.1327	0.07
330	5.7567	-0.5025	0.4975	0.25

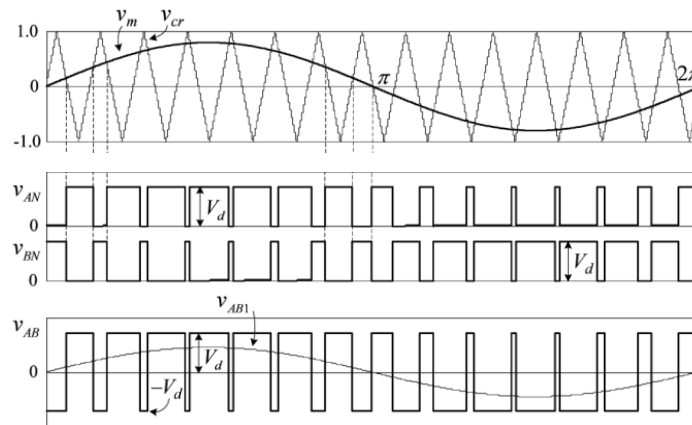
17. ábra. A mintavételezett szinuszhullám szögei (1.-2. oszlop), értékei radiánban (3. oszlop), 0-1 tartományra normalizált értékei (4. oszlop), és a létrehozott kitöltési tényezők (5. oszlop) [7]

Ebben az SPWM generációban nem szokták lehetővé tenni a PWM frekvenciájának változtatását, azonban a szinuszhullám frekvenciájának módosítását igen. [7] A hordozó-alapú PWM és a mikrokontroller alapú PWM között hasonlóságokat lehet felfedezni: a mikrokontroller által előállított PWM jel hordozójelnek tekinthető, és a szinuszhullám nem megfelelő normalizálása túlmodulációhoz vezethet.

2.3 A H-híd vezérlése

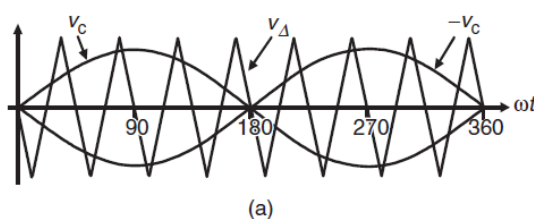
Mivel a félhíd vezérléséhez a fent tárgyalt módszereket alkalmazzák, a következő szakaszban már a H-híd vezérlésére térek ki. A H-híd felépítése több vezérlési módszer alkalmazását teszi lehetővé, amelyeket két csoportra bontunk: bipoláris és unipoláris modulációra.

A bipoláris moduláció során a H-híd egymással átlósan elhelyezkedő kapcsolóelemei ugyanazt a vezérlőjelet kapják meg (4. ábra). Így S1 és S2 kapcsolókra ugyanaz a vezérlőjel érkezik, S3 és S4 kapcsolókra pedig ezek logikai invertáltjai. A bipoláris modulációval csak $+v_i$ és $-v_i$ (a 18. ábrán $+V_d$, $-V_d$) értékeket veheti fel a kimenőjel feszültségfüggvénye, az 5. ábrán látható állapotok közül csak az első kettőt használja. [2]



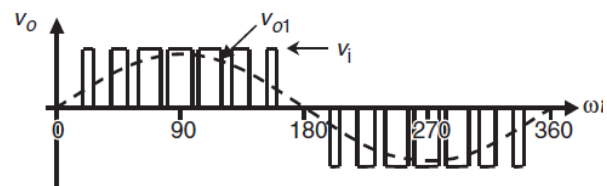
18. ábra. v_{AN} és v_{BN} pontok közötti feszültségek, illetve v_{AB} kimenőjel feszültségfüggvénye [4]

Az unipoláris modulációt többféleképpen is végre tudjuk hajtani. A leggyakoribb módszer két moduláló szinuszejel alkalmazása, amelyek egymáshoz képest 180 fokos fázistolásban vannak. [8] Az unipoláris technika az 5. ábrán szemléltetett állapotokból mind a négy definiáltat alkalmazza, így a kimenő feszültségjel $+v_i$, $-v_i$ és 0 értékeket tudja felvenni. [2] Az így elért unipoláris SPWM frekvenciája kétszerese az eredeti háromszögjel kapcsolási frekvenciájának. [17]



19. ábra.

A két moduláló szinuszejel (v_c és $-v_c$) [2]

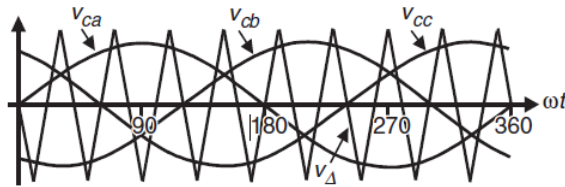


20. ábra.

Az unipoláris moduláció v_o függvénye [2]

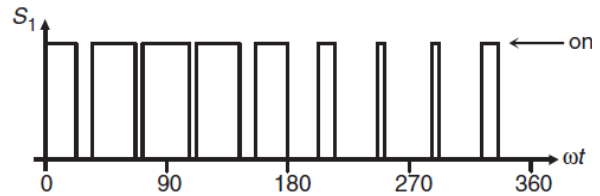
2.4 A háromfázisú rendszerek vezérlése

A háromfázisú inverterek fő feladata az egymáshoz képest 120 fokos fázistolással rendelkező kimenőjelek létrehozása. Ehhez olyan modulálójeleket alkalmaznak, amelyek között 120 fokos a fáziskülönbség. A rendszerben a hídágak felső kapcsolói (S1, S3, S5,) kapják meg a 120 fokos fázistolással rendelkező jeleket, az alsó kapcsolók (S2, S4, S6) pedig a felső jelek logikai invertáltjait (6. ábra).



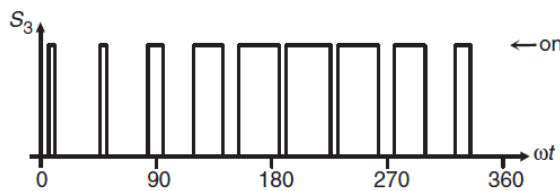
21. ábra.

120 fokkal fázistolott modulálójelek [2]



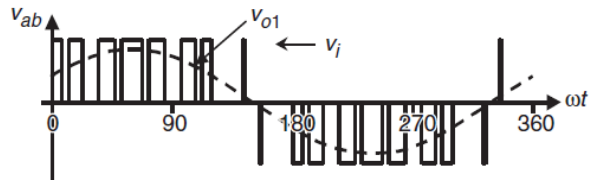
22. ábra.

S1 felső hídági kapcsoló vezérlőjele [2]



23. ábra.

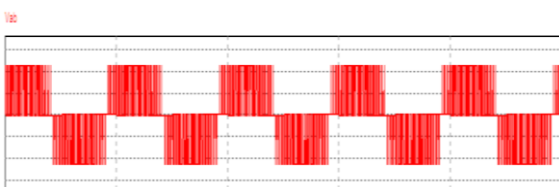
S3 felső hídági kapcsoló vezérlőjele [2]



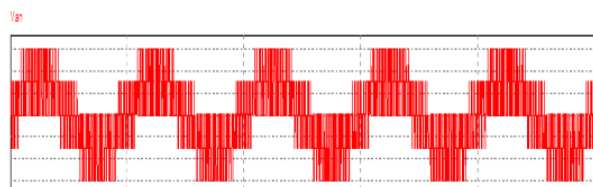
24. ábra.

V_{ab} a és b pontok közötti feszültség [2]

A 8. oldalon leírtak alapján, ebben a vezérlésben megkülönböztetünk csillagkapcsolt terhelések esetén fázis- és vonali feszültségeket is. A fázisfeszültségeket a kimenetek (a , b , c) és az n csillagpont között értelmezzük (V_{an} , V_{bn} , V_{cn}) [9], a vonali feszültségeket pedig a kimenetek között (V_{ab} , V_{ac} , V_{bc}). [2] A vonali feszültségek értékei $+v_i$, $-v_i$ és 0 értékeket vehetnek fel, a fázisfeszültségek azonban $+\frac{2}{3}v_i$, $+\frac{1}{3}v_i$, 0 , $-\frac{1}{3}v_i$ és $-\frac{2}{3}v_i$ értékek között változnak. [9]



25. ábra. V_{ab} vonali feszültség
(rezisztív terhelésen) [9]



26. ábra. V_{an} fázisfeszültség
(rezisztív terhelésen) [9]

3. Az eszköz fizikai felépítésének és vezérlésének megtervezése

3.1 A rendszer blokkvázlata

A demonstrációs eszközöm célja, hogy minél több inverteres struktúrát és vezérlési módszert tudjon bemutatni a fent tárgyaltakból. A tervezés megkezdésénél az összegyűjtött szakirodalmak alapján rendszereztem, milyen építőelemekből kell, hogy álljon az eszközöm. A leírtak alapján három egységre bontottam a felépítést: az inverternek rendelkeznie kell DC vonallal, tehát valamilyen DC tápfeszültséggel, amit szinuszos jellé alakít. Ezután következtek a nagykapacitású kondenzátorok, és a kapcsolóelemek, az inverter legfontosabb egységei, amelyek elvégzik a DC/AC átalakítást. Az inverteres struktúrákból mind a félhíd, H-híd és a háromfázisú inverter felépítését is meg szerettem volna valósítani, amelyhez a rendszeremben összesen három hídág elhelyezése mellett döntöttem. Mivel a félhidas felépítésben csak egy hídágra van szükség, a H-hidasban pedig csak kettőre, a hídágakat a rendszerből kiemelhetővé terveztem. Ezáltal a hídágak önmagukban is működőképes egységekké váltak, mivel a demonstrációs eszköztől függetlenül is lehet vezérelni őket. A „félhíd modulokat” egy nagyobb egységre lehet felhelyezni, mindig annyit, amennyit a rendszer igénybe vesz.

Vezérlési technikák közül a mikrokontroller alapú SPWM alkalmazásával a félhidas, bipoláris, unipoláris és háromfázisú vezérlések megvalósítása volt a cél. E vezérlési módszereknek a szemléltetését oszcilloszkópon keresztül terveztem, ehhez pedig mérőpontokat helyeztem el az eszközöm különböző területein.

A szakirodalmakban olvasottak alapján, a hordozó-alapú SPWM-ben változtatható a modulálójel és a hordozójel frekvenciája is, azonban a jelek létrehozásához komparációra van szükség. A mikrokontroller alapú SPWM modulációban bár nincs szükség komparációra, a hordozójel frekvenciája nem szabályozható. Az eszközöm létrehozása során az egyik célkitűzésem a két vezérlési technika ötvözése volt, mely során mikrokontroller alapú vezérlést alkalmaztam, de az alapjel frekvenciája és a hordozójel frekvenciája egyaránt változtatható paraméterként jelent meg.

Ezen célkitűzések alapján a változtatható paraméterek a rendszerben:

- a generált szinuszhullám frekvenciája
- a hordozójel frekvenciája
- az alkalmazandó inverteres struktúra (félhíd, H-híd, háromfázis)
- az alkalmazandó vezérlési módszer (bipoláris, unipoláris, háromfázisú SPWM)
- az alkalmazott holtidő
- a kiválasztható frekvenciaintervallumok

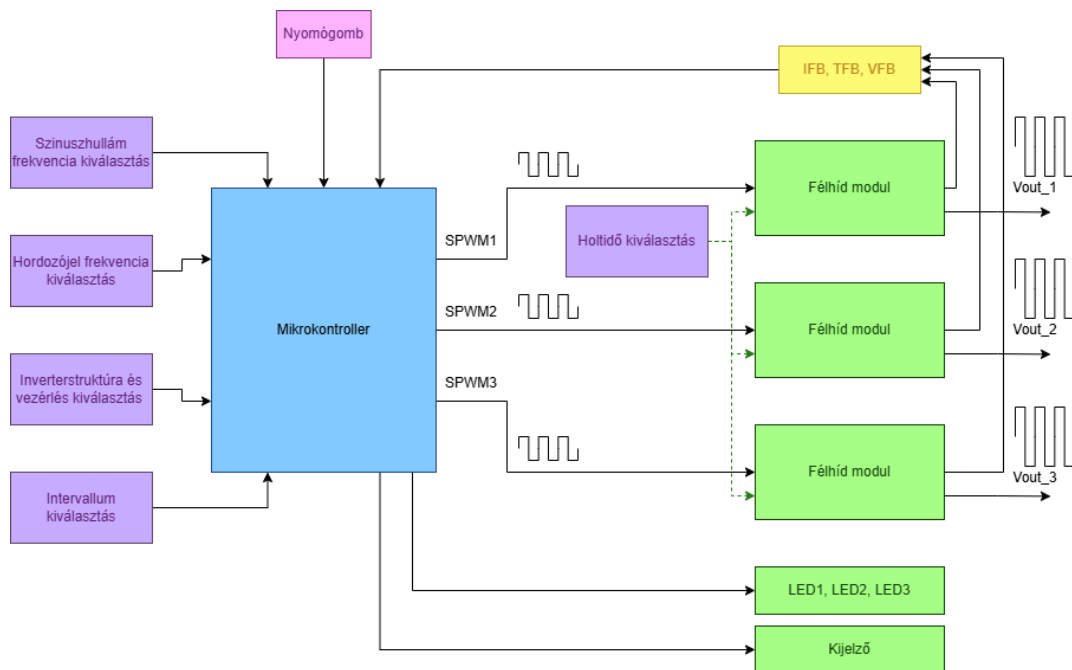
A frekvenciaintervallumok bevezetésére a legfontosabb okokat a későbbiekben fejtem majd ki, azonban érzékelhető, hogy a túl magas és túl alacsony frekvenciák problémát okozhatnak a rendszer működésében, mind a szinuszhullám, mind a hordozó esetében. Emiatt, a megfelelő működés érdekében, bizonyos határokat kell beállítani mindkét jel frekvenciájában. Azért is döntöttem az intervallumok mellett, mert széles spektrumot szerettem volna lefedni a jelek generációja során: akár három nagyságrendnyi különbséget is a kiválasztható frekvenciák között. Ezt azonban csak kontrollált körülmények között lehetett megvalósítani.

A rendszer túlterhelésének elkerüléseért a vezérlőjelek változtatásával a rendszer nem azonnal változtatja meg az inverter paramétereit, hanem egy gombnyomás hatására. Így mentem biztosra abban, hogy a felhasználó csak azokat a feltételeket állítja be az eszközön, amelyeket ténylegesen ki is szeretne választani.

Ahhoz, hogy a demonstráció minél több információt adjon a felhasználónak a működésről, elengedhetetlennek tekintettem különféle visszajelzést adó elemek elhelyezését. Emiatt a fent leírt paramétereket egy kijelző szemlélteti, amely megjeleníti az aktuálisan beállított értékeket és a kiválasztható paramétereket is. Ezen kívül, a rendszer által használni kívánt félhidakat LED-ek is jelzik, annak érdekében, hogy a hozzá nem értők számára is egyértelmű legyen, hogy mely hídágak aktívak (még mérőeszközök nélkül is). A rendszert egy mikrokontroller működteti az egyszerűség és a helytakarékoság miatt, de az előbb említettek alapján, mivel a félhíd modulok kiemelhetőek az eszközből, a vezérlés nem csak mikrokontroller alapú lehet.

Különféle védelmek elhelyezését is megvalósítottam a rendszerben. A hídzárlat elleni védelemért egy áramvisszacsatolást (IFB – Current Feedback) építettem be a rendszerbe, amely monitorozza a hídágakon átfolyó áram nagyságát. Egy feszültségvisszacsatolás (VFB – Voltage Feedback) segítségével egyfajta túlfeszültségvédelem is megvalósul, folyamatos hőmérsékletméréssel (TFB – Temperature Feedback) pedig a kapcsolóelemek túlmelegedése előzhető meg. Ezek a visszacsatolt jelek mind a mikrokontrollerbe vannak visszavezetve, ami elvégzi a szükséges lépéseket a rendszer épségének megőrzéséért (pl.: kapcsolóelemek kinyitása, ezáltal az áramkör megszakítása). A demonstrációs eszközben csak az inverter főbb elemei kerültek elhelyezésre, így a kimenete még nem tiszta szinuszos kimenet. Ez azonban lehetőséget ad a szűrőrendszerek tetszőleges méretezésére. Mivel az eszközöm demonstrációs célra készült, labortápegységről működtethető, így biztonságosabb a használata.

Ezen elképzelések alapján a rendszer blokkdiagramja:



27. ábra. A rendszer blokkvázlata

Az egyik legfontosabb kikötés az eszközöm megtervezésénél a valós idejű szinuszhullámok és hordozójelek generálása volt. Ez lényegében azt jelenti, hogy ha a felhasználó által beállított szinuszhullám 20 Hz-es, a hordozójel pedig 50 Hz-es, akkor a rendszernek generálás közben tényleg meg kell közelítenie valós időben ezeket a frekvenciákat.

3.2 A rendszer hardveres építőelemeinek kiválasztása

A kapcsolóelem kiválasztása

A kapcsolóelemek a legfontosabb egységek az inverterekben, így a megfelelő kiválasztásuk nagyban befolyásolja a rendszer működését. Az eszközümet nem nagyteljesítményre terveztem, azonban széles spektrumú kapcsolási frekvenciára. Mivel az IGBT-k maximum 25-30 KHz-ig alkalmazhatóak, és magasabb költséggel is rendelkeznek, MOSFET-ek alkalmazása mellett döntöttem. Így tudtam biztosra menni abban, hogy a kapcsolási frekvencia nem szab határt a rendszeremnek, amely bár csak demonstrációs célokra készült, kapcsolási frekvencia téren szerettem volna a valódi inverterek kapcsolási frekvenciáit megközelíteni (20 - 30 KHz).

A MOSFET kiválasztásánál számos paramétert figyeltem, de a legfontosabbak:

- Drain-source breakdown voltage (drain-source letörési feszültség): A maximális feszültség, amelyet a kapcsolóelem drain-source elektródáira lehet helyezni annak a kikapcsolt állapotában.
- Gate-source threshold voltage (gate-source küszöbfeszültség): A gate és source elektródák közé helyezendő küszöbfeszültség, amely a kapcsolóelem bekapcsolásához szükséges.
- Drain-source on-state resistance ($R_{DS(on)}$, drain-source bekapcsolt állapotú ellenállás): A drain- és source elektródák közötti ellenállása a kapcsolóelemnek, annak bekapcsolt állapotában. A vezetési veszteség függ tőle, mivel ezen az ellenálláson feszültség esik, amikor áram folyik az alkatrészen keresztül. Az eszköz melegedésére is hatással van, hiszen a nagyobb ellenállású eszköz több hőt termel. Ideális esetben minél kisebbre választandó.
- Total gate-charge (Q_{gate} , teljes gate töltés): Az a töltésmennyiség, amelyet a kapcsolóelem gate elektródájára kell vinnünk ahhoz, hogy az teljesen bekapcsoljon. A parazitakapacitások befolyásolják az értékét. Ideális esetben minél kisebb, hiszen ha nagy mennyiségű töltésre van szükség a bekapcsoláshoz, több idő bekapcsolni a MOSFET-et. Így ez a paraméter a kapcsolási frekvenciának szab határt.
- Continuous drain-current (I_D , folytonos drain áram): Az a maximális áram, amivel a MOSFET csatornáját folyamatosan terhelhetjük. Impulzusos vezérlés mellett (amely nem folytonos áramot eredményez) ezt az értékét átlépheti a drain-áram.

A kapcsolóelemek egyéb jellemzője a Figure of Merit (FOM), melynek kiszámítási módja $R_{DS(on)} * Q_{gate}$. Ez a paraméter a MOSFET-ek összehasonlítását segíti nagyfrekvenciás alkalmazhatóság terén. Minél alacsonyabb a FOM, annál kisebb a kapcsolóelem vesztesége, így magasabb a rendszer hatékonysága. [23]

Alkatrész kiválasztás során csak n-csatornás növekményes FET-eket kerestem, mivel ezek vezérléséhez pozitív gate-feszültségre van szükség, így egyszerűbb a be- és kikapcsolásuk. A fent leírtak alapján két teljesítmény MOSFET-et találtam, amelyek könnyen elérhetőek voltak és megfelelő paraméterekkel is rendelkeztek: Az NCE6020AK teljesítmény MOSFET-et és az NTD5865NL teljesítmény MOSFET-et. Mindkét kapcsolóelem 60 Voltos maximális drain-source feszültséggel rendelkezett, ami ideális, sőt még túlméretezett is a labortápos felhasználásra. Természetesen ezeknek a FET-eknek a kiválasztásával véglegesen eldöntöttem, hogy a rendszerem nem lesz valódi inverterként alkalmazható (230 VAC

létrehozására). Mindkét FET nagyfrekvenciás alkalmazásra lett tervezve, és alacsony $R_{DS(on)}$ -nal is rendelkeztek, de az NTD5865NL egy kicsit jobb tulajdonságokat mutatott. A paraméterek, amelyeket összehasonlítottam az 1. táblázatban találhatóak:

	NTD5865NL	NCE6020AK
IDRAIN (max)	46 A	20 A
IDRAIN (pulse)	203 A	60 A
$R_{DS(on)}$ (max), $V_{GS}=10V$ mellett	16 m Ω	27-35 m Ω
Q_{GATE} (max)	29 nC	47 nC
Maximum Power Dissipation	71 W	45 W
FOM (Figure Of Merit)	464 p Ω *C	1269 p Ω *C

1. táblázat. A két teljesítmény MOSFET paramétereinek összehasonlítása [24] [25]

Ezeknek az adatoknak az összehasonlítása alapján úgy ítéltam meg, hogy az NTD5865NL mellett döntök, a magasabb kapcsolási frekvencia, kisebb FOM és a nagyobb teljesítménytolerancia miatt. Ahhoz, hogy az eszközöm minél jobban hasonlítson egy igazi inverterhez, az alkatrészválasztás során valós teljesítmény MOSFET-eket kerestem. Így, bár 46 Amper valószínűleg soha nem fog keresztülfolyni a rendszeren, az ilyen mértékű túlméretezéssel egyben biztosra mentem abban is, hogy nem fognak könnyen károsulni a kapcsolóelemek. A kiválasztás során azzal is számoltam, hogy bár ideális esetben maximum pár száz mA körüli áram folyik a félhidakon, hídzárlat esetén akár 3-4 Amper is megjelenhet a MOSFET-ek drain-source vonalán. Így minimum 5 Amper áramtoleranciával kialakított kapcsolókat vettem csak figyelembe. A kiválasztott kapcsolóelem DPAK tokozású SMD alkatrész, így kevés helyigénnyel is rendelkezik.



28. ábra. Az NTD5865NL teljesítmény MOSFET

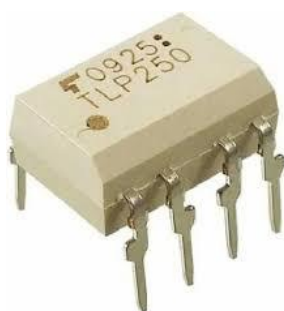
Egyik FET sem logikai szintű, a legalacsonyabb $R_{DS(on)}$ -t a 10 Voltos gate-source feszültségnél lehetett leolvasni az adatlapjaikban. Ez azt jelentette, hogy csak egy mikrokontroller nem elég a megfelelő be- és kikapcsolásukhoz, így valamilyen gate-meghajtó áramkör alkalmazására volt szükség.

A gate-meghajtó áramkör kiválasztása

A fent leírtak alapján is látható, hogy a FET-ek gyors kapcsolásához nagy áramokat kell szolgáltatni a gate elektródájukra, amit egy mikrokontroller nem képes megtenni károsodás nélkül. Az Arduino Uno-t alapul véve, egy mikrokontroller kivezetésén maximum 20 mA áram folyhat keresztül. Ez azonban nem elegendő egy teljesítmény FET működtetéséhez, amelynek többszáz mA áramra is szüksége lehet.

A gate-meghajtó áramkörök lényegében teljesítményerősítő áramkörök, amelyek egy alacsony teljesítményű jelet fogadnak a bemenetükre (pl.: mikrokontroller PWM jelét) és azt felerősítik annyira, hogy az a FET-ek kapcsolásához elegendő legyen. Több típusuk is létezik, és amikor gate-meghajtót kerestem, több fajtát is összehasonlítottam:

A high-side, low-side meghajtók egyetlen FET működtetésére képesek. A high-side (felsőági meghajtó) a hídág tápfeszültséghez közelebbi kapcsolóelemét hajtja meg, a low-side (alsóági) meghajtó a földhöz közelebbi FET-et vezérli. Ezt a két meghajtási módszert fontos megkülönböztetni, hiszen a MOSFET-ek gate elektródájára a feszültséget mindig a source kivezetésükhöz viszonyítva kell adni. Még az alsóági FET-ek meghajtása egyszerű a földpotenciálú source elektródájuk miatt, a felsőági FET-ek source lába nem földpotenciálon van, így a láb „lebegőnek” tekinthető. Emiatt „lebegő” tápfeszültségre is van szükség a vezérlésükhöz. A meghajtók további feladata lehet a galvanikus leválasztás, amellyel elkülönítik az áramkör nagyteljesítményű részét a logikai elemeket tartalmazó területtől. Ilyen meghajtó a TLP250, amely optikai leválasztást alkalmaz:



29. ábra. A TLP250 gate-meghajtó

Ennek a meghajtónak a hátránya, hogy annyi darabra van szükség belőle, ahány FET-et meg kell hajtani, illetve a maximális kapcsolási frekvenciája 25 KHz környékén van. Az én eszközöm esetében 30 kHz kapcsolási frekvenciát szerettem volna elérni, így ez nem volt elegendő. A holtidő kezelését is szoftveresen kellett volna megoldanom ennek az

alkatrésznek az alkalmazásával, ami bár nem lehetetlen, egyszerűbb, ha a meghajtóba beépített funkció.

Mivel nem csak egy-egy önálló kapcsolóelemet kellett meghajtanom, hanem félhidakat, a legegyszerűbb módszer egy félhídmeghajtó kiválasztása volt. A félhídmeghajtókba alpból beépítik a félhidak meghajtásához szükséges legfontosabb tulajdonságokat: felsőági, alsóági meghajtást, szintléptetést (level shifting), bootstrap áramkör alkalmazását (a felsőági FET megfelelő bekapcsolásához), shutdown biztonsági lekapcsolást és meg sok más funkciót. Ezen kívül az egyik legnagyobb előnyük, hogy egy félhid (vagyis 2 darab FET) meghajtásához elegendő egy ilyen meghajtó, tehát feleannyira van szükség belőlük, mint például a TLP250-ből. A tervezésem során több ilyen félhid meghajtót is összehasonlítottam, főleg a beépített lehetőségeket, a meghajtás komplexitását és sebességet figyelve. Ez alapján jutottam el az Infineon IR218x termékcsaládhoz, amelyhez hat integrált áramkör meghajtó tartozik (30. ábra).

Part	Input logic	Cross-conduction prevention logic	Dead-Time	Ground Pins	Ton/Toff
2181	HIN/LIN	no	none	COM	180/220 ns
21814				VSS/COM	
2183	HIN/LIN	yes	Internal 500ns	COM	180/220 ns
21834			Program 0.4 ~ 5 us	VSS/COM	
2184	IN/SD	yes	Internal 500ns	COM	680/270 ns
21844			Program 0.4 ~ 5 us	VSS/COM	

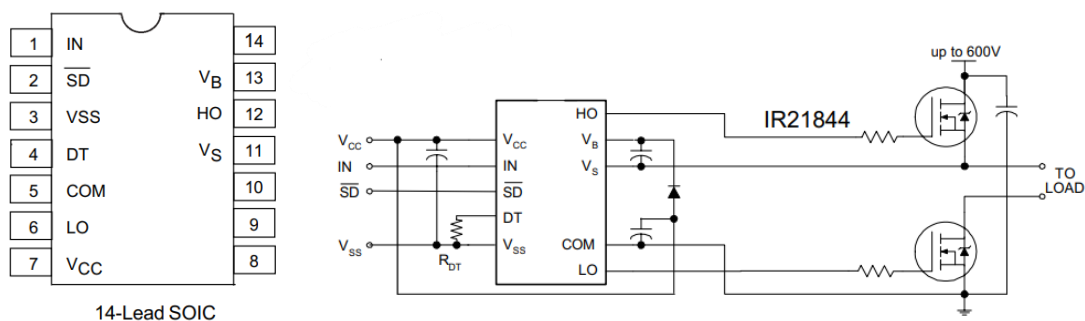
30. ábra. A félhid meghajtók összehasonlítása [20]

Az összehasonlításomat a 2181-es alkatrészsel kezdtem. A bemeneti logikája (Input Logic) HIN/LIN, ami azt jelenti, hogy a felsőági és alsóági FET-ek vezérlésére külön jeleket fogad. Nem rendelkezik kereszt-vezetést megelőző logikával (cross-conduction prevention logic), amely megakadályozná azt, hogy az egy hídágban lévő két kapcsolóelem egyszerre nyisson ki. Beépített holtidővel sem látták el.

Mivel szerettem volna az eszközömben a holtidőt is vezérelhető mennyiségként kezelni, az IR21834 és az IR21844 modulok jöhetek szóba, amelyek programozható holtidőt tettek lehetővé. A legfőbb különbség közöttük, hogy az IR21834 lehetővé teszi a hídág kapcsolóelemeinek egymástól függetlenül történő vezérlését, az IR21844 ezzel szemben csak egy bemenőjelet (IN) fogad. Az IR21844 SD lábbal is rendelkezik, ami a shutdown (leállítás) funkciót vezérli, és lehetővé teszi, hogy a vezérlőjel megszakítása nélkül azonnal kinyissuk a FET-eket. Ennek a viselkedése aktív alacsony, tehát logikai 0 jelre lép életbe a

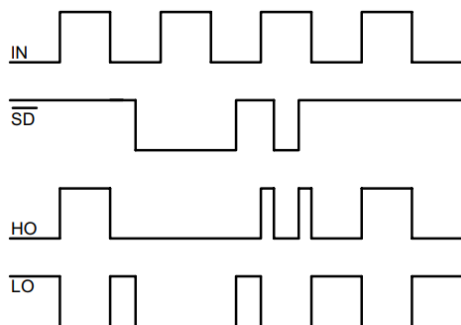
működése. A döntésem végül az IR21844-es meghajtóra esett, mivel a shutdown funkció jelenléte lehetőséget ad az eszközöm működésének biztonságosabbá tételére. Emellett, mivel csak egy IN bemenete van az alkatrésznek, két FET vezérléséhez elegendő volt egy mikrokontroller lábat felhasználnom. Az egyetlen kompromisszum, amit ennek a komponensnek a kiválasztásával meg kellett kötnöm, hogy a két FET egyedi jellel történő kapcsolása lehetetlenné vált. A vezérlési technikákhoz, amiket meg szerettem volna valósítani, ez egy elfogadható feltétel volt, nem volt szükség a külön-külön történő vezérlésre.

Az IR21844 meghajtó egészen 600 Volt DC feszültségig alkalmazható, és rendelkezik UVLO-val (Undervoltage Lockout), bootstrap áramkörrel, szintléptetéssel (level-shifting) és kereszt-vezetést megelőző logikával is. Ideális a nagyfrekvenciás áramkörökhöz, és n-csatornás MOSFET-ek, illetve IGBT-k meghajtására is használható. Az eszköz holtidejét egy ellenállással lehet beállítani (R_{DT}), melyet a DT és V_{SS} lábak közé kell elhelyezni. Ennek az ellenállásnak az értéke $0\ \Omega$ és $200\ K\Omega$ között változhat, ami $0.4 - 5\ \mu s$ -ig változtatja a holtidő értékét. A HO (High Out) vezérlőjelet a meghajtó a felsőági kapcsolóelemnek szolgáltatja, LO-t (Low Out) pedig az alsóáginak.

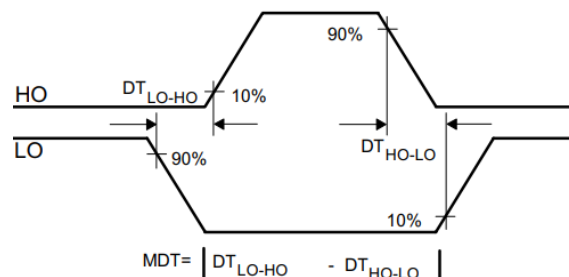


31. ábra. Az IR21844 lábkiosztása és ajánlott kapcsolási rajza [20]

A meghajtó által kiadott jelek (HO/LO):



32. ábra. Az IN láb és a HO/LO lábak feszültségjelei [20]



33. ábra. A holtidő szemléltetése a HO/LO lábakon [20]

A 32. ábrán látható, hogy a HO feszültségjel leköveti az IN bemenetre kapcsolt vezérlőjelet. Az LO lábon pedig mindig a HO láb kimenőjének logikai invertáltja valósul meg. Az SD láb aktiválásakor mind a HO és LO feszültségjelek is 0 értékre csökkennek, így a hídág kapcsolóelemei letiltásra kerülnek. A holtidő működését a 33. ábrán figyelhetjük meg, ahol egy időbeli eltolás jön létre az elemek ki- és bekapcsolása között. A holtidő állapotában mindkét teljesítménykapcsoló kikapcsolt állapotban van (33. ábra, DT).

A mikrokontroller kiválasztása a szoftveres követelmények alapján

Miután eldöntöttem, hogy a demonstrációs eszközümet egy mikrokontroller felhasználásával szeretném megvalósítani, ki kellett választanom, melyik mikrovezérlő rendelkezik azokkal a paraméterekkel, amelyekre szükségem van. A rendszer szoftverével kapcsolatban az egyik legfontosabb feltétel az volt, hogy képes legyen valós idejű szinuszos rezgések generálására, vagy legalább közelítse meg a tényleges frekvenciákat. Mivel a hordozójel frekvenciáját is vezérelhető tényezőként szerettem volna kezelni, a PWM frekvencia szabályozásának lehetőségeivel kezdtem a kutatást. Először az Arduino UNO mikrokontroller képességeit mértem fel.

A PWM jelek széles skálájú manipulációjához regiszterprogramozást alkalmaznak, mely során a regiszterekbe írt megfelelő értékekkel pár tized Hz-től több száz KHz-ig is szabályozható a kiadott jel frekvenciája. Ehhez a beépített időzítőket és megszakításkezelőket használja fel a mikrokontroller.

Az Arduino UNO három időzítővel van felszerelve, melyek nevei: Timer0, Timer1 és Timer2. Ezek közül Timer0 és Timer2 8 bites időzítők (vagyis 255-ig képesek számolni), Timer1 pedig 16 bites időzítő. Az időzítők esetében a PWM frekvenciája az alábbi képlettel számítható ki:

$$f_{\text{PWM}} = \frac{f_{\text{CLOCK}}}{\text{prescaler} \times (\text{timer}_{\text{TOP}} + 1)}$$

Ahol f_{PWM} a PWM jel frekvenciája, f_{CLOCK} a mikrokontroller órajelének frekvenciája (az Arduino UNO mikrokontrollerben 16 MHz) és $\text{timer}_{\text{TOP}}$ az időzítő maximális értéke (Timer0 és Timer2 esetében 255). A prescaler, vagyis előskálázó feladata, hogy a PWM frekvenciáját csökkentse. Értékei általában: 1 (vagyis prescaler nélküli PWM), 8, 64, 256, és 1024. Látható, hogy a képlet alapján Timer0 és Timer2 csak egy pár PWM frekvencia beállítására képesek, mivel f_{CLOCK} és $\text{timer}_{\text{TOP}}$ nem változtatható értékek (körülbelül 5-6 db különféle

frekvencia hozható létre velük a prescaler módosításával). Ettől az én eszközömben sokkal kifinomultabb frekvenciaszabályozásra volt szükség, amelyre csak Timer1 képes:

Timer1 16 bites időzítő, ami azt jelenti, hogy 0 – 65535-ig számol. Ez már önmagában is alacsonyabb frekvenciák beállítását teszi lehetővé:

$$f_{\text{PWM}} = \frac{f_{\text{CLOCK}}}{\text{prescaler} * (\text{timer}_{\text{TOP}} + 1)}$$

$$f_{\text{PWM}} = \frac{f_{\text{CLOCK}}}{\text{prescaler} * (\text{timer}_{\text{TOP}} + 1)}$$

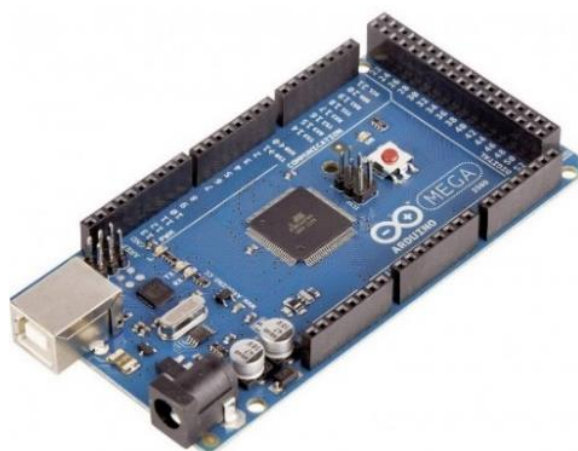
$$f_{\text{timer0MIN}} = \frac{16\,000\,000}{1024 * (255 + 1)} \sim 61 \text{ Hz}$$

$$f_{\text{timer1MIN}} = \frac{16\,000\,000}{1024 * (65535 + 1)} \sim 0,24 \text{ Hz}$$

34. ábra. Timer0 és Timer2 által beállítható legkisebb frekvencia 61 Hz.

35. ábra. Timer1 által beállítható legkisebb frekvencia kevesebb, mint 1 Hz.

Azonban Timer1 legnagyobb előnye, hogy rendelkezik egy kiegészítő regiszterrel (ICR1), amelynek az értéke meghatározza azt, hogy meddig számol az időzítő. ICR1 (ami a timer_{TOP} szerepét tölti be a képletben) bármilyen egész számú értéket felvehet 0 - 65535-ig. Az így beállítható különböző frekvenciák száma lényegesen több, mint a másik két időzítő esetében. Timer1 kiválasztása után a legnagyobb problémát a Timer1 által vezérelt kimenetek száma jelentette. Ez az időzítő az Arduino UNO mikrokontrollerben a kilences és tízes lábakat vezérli (Pin 9, Pin 10). Ez maximálisan két PWM jel létrehozását engedi, így a három fázis megvalósítása nem lett volna lehetséges. Emiatt az Arduino MEGA 2560 mikrokontrollert választottam, amely regisztereiben és felépítésében sokban hasonlít az Arduino UNO-ra, de négy 16 bites időzítője is van. Ezek a 16 bites időzítők név szerint Timer1, Timer3, Timer4 és Timer5. Timer0 és Timer2 a MEGA mikrokontrollerben is 8 bites időzítők.



36. ábra. Az Arduino MEGA 2560 mikrokontroller

Az Arduino MEGA alapja az ATmega2560 integrált áramkör. Az Arduino UNO-hoz hasonlóan 16 MHz-es az órajele, és 8 KB SRAM memóriája is van, ami sokkal több, mint az UNO mikrovezérlőé. A legnagyobb hátrányt az árban találhatjuk, mivel egy Arduino UNO ára 10 000 Ft környékén mozog, egy Arduino MEGA értéke a 20 000 Ft-ot is meghaladja.

A kijelzőelemek és a vezérléshez szükséges alkatrészek kiválasztása

Amikor kijelző eszközt választottam a rendszernek, a legfontosabb szempontok a kijelző mérete és a kijelző által megjeleníthető karakterek maximális száma volt. Először a hagyományos Arduino projektekben is gyakran használt LCD kijelzőket vettem szemügyre. A megjelenítendő adatok mennyisége miatt csak a 20x4 karakteres kijelző jöhetett volna szóba, de a fizikai méretéből kifolyólag nagy területet kellett volna elkülöníteni számára a nyomtatott áramkörön. Végül a legjobb döntés egy OLED kijelző alkalmazása volt, mely lényegesen több karakter megjelenítésére is képes. Egy ilyen OLED kijelző által megjeleníthető maximális karakterszám a karakterek méretétől függ, de általában 5x7 pixel karakter-mérettel számolnak. Az átláthatóság érdekében a karakterek között ki szoktak hagyni egy oszlop és egy sor pixelt, így a karakterek „valódi” mérete 6x8 képpont. Így a 128x64 pixelt tartalmazó kijelzőn a maximális karakterszám:

- Egy sorba helyezhető karakterek száma: $\frac{128 \text{ pixel}}{6 \text{ pixel}} = 21$ karakter.
- Egy oszlopba helyezhető karakterek száma: $\frac{64 \text{ pixel}}{8 \text{ pixel}} = 8$ karakter.

Egy OLED kijelző tehát 21x8 karakter megjelenítésére képes. A kijelző kiválasztásánál az alábbi tulajdonságokat hasonlítottam össze:

	LCD 16x2	LCD 20x4	OLED 128x64
Méret	80 x 36 mm	98 x 60 mm	27.3 x 27.8 mm
Megjeleníthető max. karakterszám	16 x 2 = 32 karakter	20 x 4 = 80 karakter	21 x 8 = 168 karakter
Ár	1670 Ft.	2539 Ft.	2864 Ft.
Kommunikáció	I ² C	I ² C	I ² C

2. táblázat. A kijelzők összehasonlítása

Mivel az OLED kijelző több mint kétszer annyi karakter megjelenítésére képes, mint a 20x4-es LCD kijelző, és sokkal kevesebb helyet igényel, ennek a kijelzőnek az alkalmazása mellett döntöttem. Az OLED-ek előnye az alacsony fogyasztásuk (nincs szükségük

háttérvilágításra) és az is, hogy képek megjelenítésére is képesek. Ezt a képmegjelenítő képességet ki is használtam a későbbiekben.

A rendszer vezérléséhez potenciométerek alkalmazása mellett döntöttem, melyek kezelése és beépítése egyaránt egyszerű. A szinuszjel frekvenciájának változtatásához és az alkalmazandó üzemmód kiválasztásához elegendőnek bizonyult két egyfordulatos potenciométer alkalmazása. Az igazi problémát a hordozójel frekvenciájának szabályozása okozta, ahol több tíz KHz lefedésére volt szükség. Ehhez nem elegendő egy egyfordulatos potenciométer, mivel már kifinomult forgatással is túl nagy különbséget idézne elő a felhasználó a hordozójel frekvenciájában. Természetesen ezt úgy is lehet orvosolni, hogy a kiválasztható értékek nem egyesével, hanem százasaival, akár ezreseivel változnak a potenciométer tekerése közben. A végső rendszerben ezt a módszert kombináltam egy többfordulatos trimmer potenciométer alkalmazásával, ami lehetővé tette a precizitást a vezérlés közben.

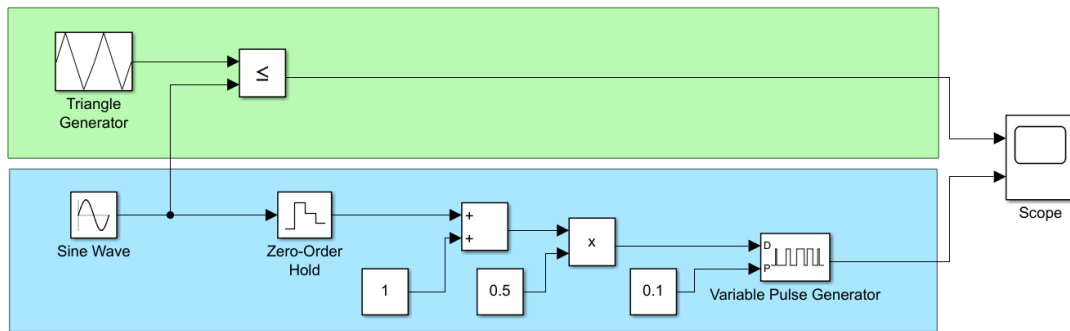
3.3 A rendszer működésének teszteléséhez szimulációk készítése

A rendszerteknikai tervezésben a különféle feszültségjelek összeadása- és kivonása miatt a szimulációs tesztek elvégzése elengedhetetlen volt. Mivel a demonstrációs eszközöm működtetésében hordozó-alapú SPWM technikákat alkalmazok mikrokontroller alapú SPWM vezérlésre, a legnagyobb kérdést az vetette fel, hogy a kívánt feszültségfüggvények kirajzolódnak-e. Ahhoz, hogy jobban értelmezni tudjam a két vezérlési módszer közötti különbséget, MATLAB szoftveres szimulációkat végeztem. Bár a hordozó-alapú SPWM-et sok elektronikai szimulációs szoftver le tudta volna tesztelni (pl.: TINA), a mikrokontroller alapú SPWM kirajzolása MATLAB segítségével volt a legegyszerűbb.

A hordozó-alapú SPWM és a mikrokontroller alapú SPWM közötti különbség

Ahogy azt fent is említettem, a mikrokontroller által periodikusan mintavételezett szinuszhullámból származó vezérlőjel modellezésére a MATLAB környezet a legalkalmasabb, mivel nem csak elektronikai alkatrészek viselkedését kell modellezni.

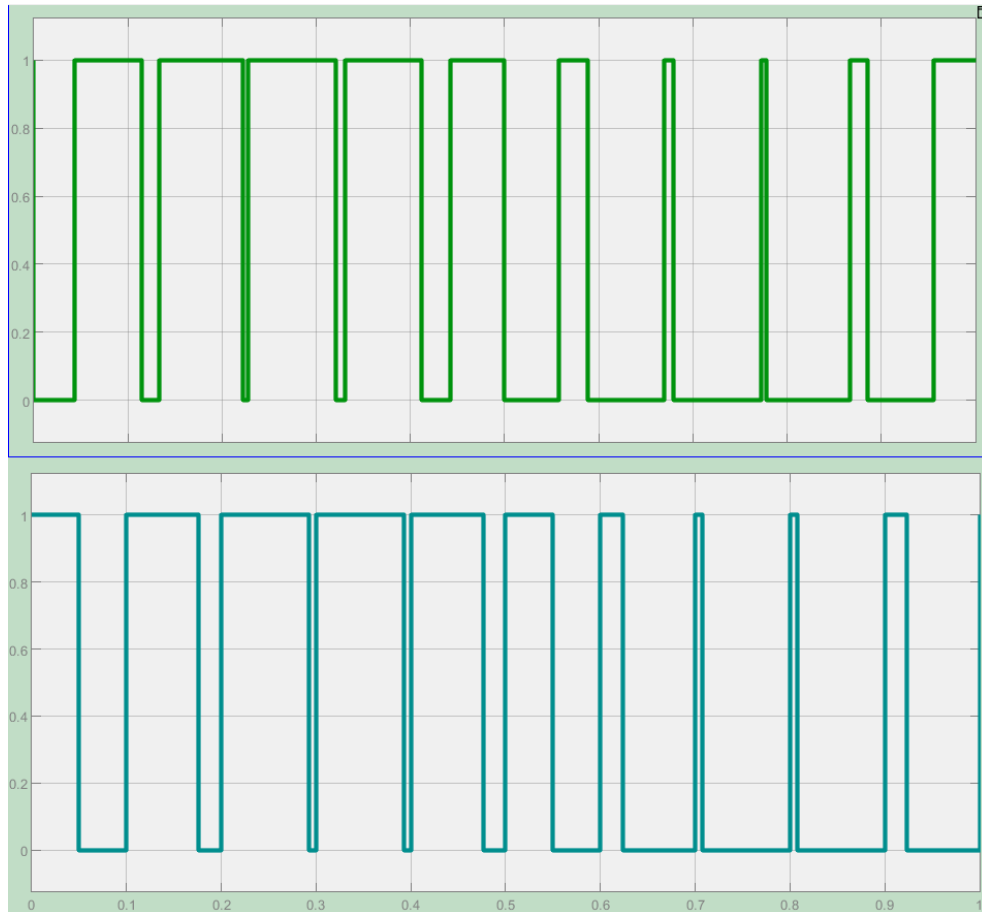
A szimulációt a két jel összehasonlításával kezdtem Simulink segítségével:



37. ábra. A két jel előállítása Simulink segítségével

A hordozó-alapú SPWM (zölddel jelölt) generálásához csak egy háromszögjel generátorra és egy szinuszhullám generátorra van szükség. A „ $\leq$ ” relációs operátor elvégzi a komparálást, és amikor a szinuszhullám értékei nagyobbak, logikai 1-est állít be a kimenetére. A túlmoduláció elkerüléséért a szinuszjel amplitúdóját 0.9-re állítottam, így biztosra mentem abban, hogy nem fognak kimaradni impulzusok. A megfelelő összehasonlítás érdekében mindkét jel egy 1 Hz-es szinuszhullám 10 Hz-es hordozójellel történő átvitelét valósítja meg.

A mintavételezett SPWM előállítása egy kicsit komplexebb megközelítést igényel. A szinuszhullámot először mintavételezni kell, egy nulladrendű tartószerv (zero-order hold) segítségével. A tartószerv 0.1 másodpercenként vesz mintát a modulálójelből (a 10 Hz beállításához). Mivel a szinuszhullám $[-1, 1]$ intervallumon vesz fel értékeket, ezeket normalizálni kell $[0, 1]$ intervallumra (a szakirodalomban is tárgyaltak szerint). Ehhez először hozzá kellett adnom 1-et, majd szoroztam az eredményt 0,5-tel. Így 0-1 közötti értékeket kaptam, melyek alkalmasak 0-100% közötti kitöltési tényezők beállítására. A változó-impulzus generátor (Variable Pulse Generator) két adatot fogad a bemenetére: D (duty cycle) és P (period) bemeneteket. A „D” bemenet egyértelműen az előbb generált 0-1 közötti kitöltési tényezőket kapja meg, a „Period” bemenettel pedig egy impulzus periódusidejét tudjuk szabályozni. Mivel 10 Hz-es hordozójel demonstrációja a cél, egy periódus ideje 0.1 másodpercnek felel meg. Az így kapott szimulációs eredmények:

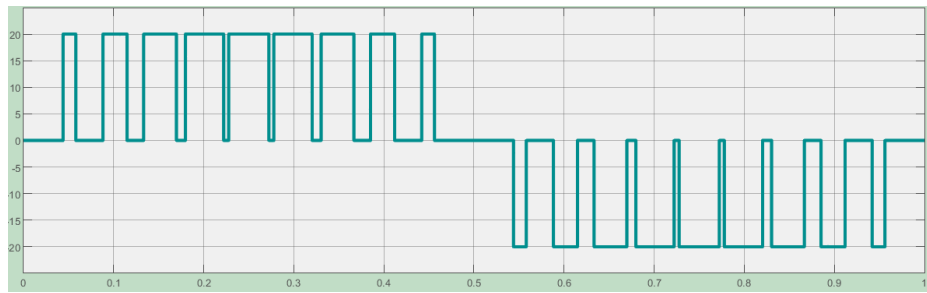


38. ábra. A hordozó-alapú SPWM (zölddel jelölt) és a mintavételezéses SPWM (késsel jelölt) összehasonlítása

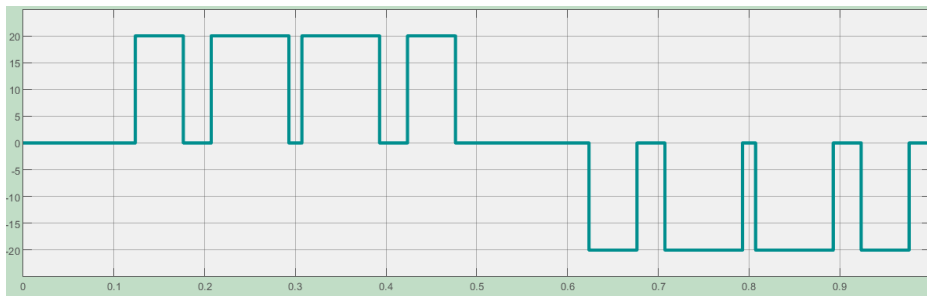
A szimulációk alapján egyértelműen látható, hogy az SPWM osztások kitöltési tényezőiben és időbeli elhelyezkedésében is különbség van. Míg a mintavételezett PWM impulzusainak felfutó élei között mindig 0.1 másodperc a különbség, a hordozójel-alapú PWM impulzusainak kezdete a komparációk eredményeitől függ. Mivel nem ugyanazokba a pozíciókba esnek az impulzusok, és nem is ugyanolyan hosszúságúak, a rendszerem működésének biztosításához további szimulációk elvégzésére volt szükség.

A félhíd, H-híd és a háromfázisú struktúrák szimulációi

A félhíd és a H-híd bipoláris modulációjának szimulációjakor nem lehetett nagy különbséget felfedezni a kimenőjelek között. A szakirodalomnak megfelelően a kimenőjelek lekövétték a vezérlőjeleket. Igazi számottevő különbséget a H-híd unipoláris modulációjában találtam:

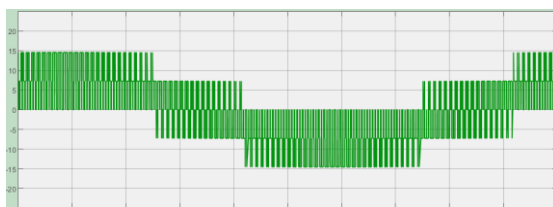


39. ábra. A H-híd unipoláris modulációja (hordozó-alapú)

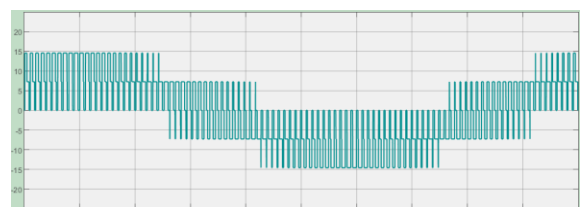


40. ábra. A H-híd unipoláris modulációja (mintavételezéses)

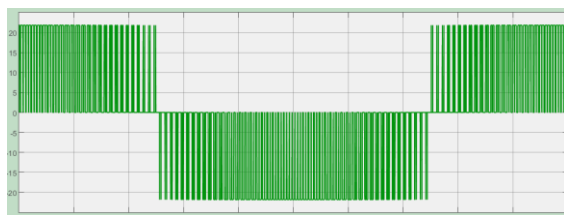
Bár a mintavételezéses módszerrel is megvalósult az unipoláris moduláció, a két kimenőjel kinézetében eltérés fedezhető fel. A szakirodalmi áttekintésben említettem, hogy a hordozó-alapú módszernél a hordozójel frekvenciájának kétszerese az unipoláris jel frekvenciája. Ez teljesül is a 39. ábrán szemléltetett szimulációban. A mintavételezéses megvalósításban azonban, mivel a vezérlőjelek impulzusainak felfutó élei között mindig ugyanannyi időkülönbség van, az unipoláris jel frekvenciája megegyezik az eredeti vezérlőjel frekvenciájával (40. ábra). Ezt a demonstrációs eszközöm működésénél figyelembe kellett vennem. A háromfázisú felépítés tesztjéhez az értelmezhető függvények kirajzolódásának érdekében a hordozójel frekvenciáját 100 Hz-re emeltem:



41. ábra.
A fázisfeszültség tesztelése (hordozó-alapú)

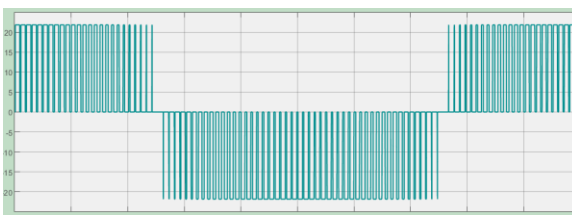


42. ábra.
A fázisfeszültség tesztelése (mintavételezéses)



43. ábra.

A vonali feszültség tesztelése (hordozó-alapú)



44. ábra.

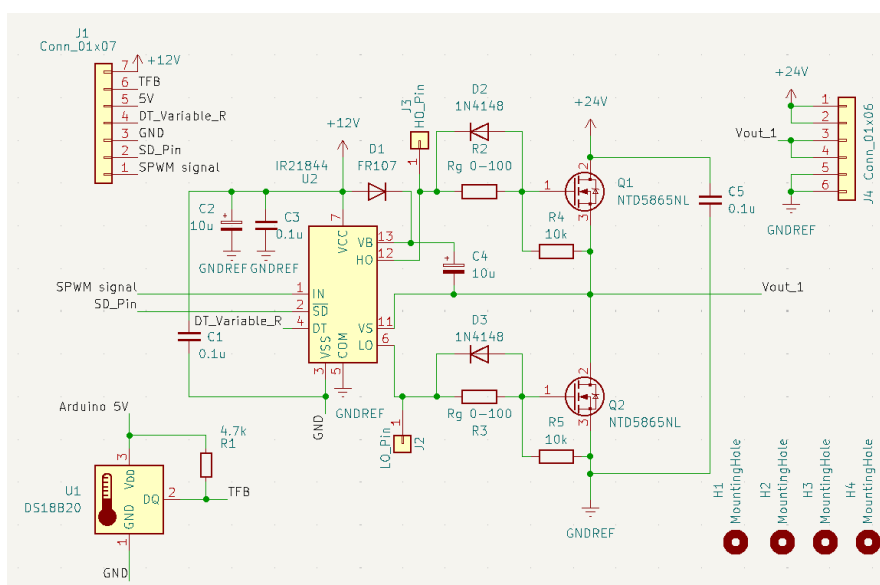
A vonali feszültség tesztelése (mintavételezéses)

A szimulációs tesztek elvégzésével biztosítani tudtam azt, hogy a hordozó-alapú technikákat meg lehet valósítani mikrokontrolleres vezérléssel is. Bár vannak a feszültségjelekben kisebb eltérések, a jelek legfontosabb elemei változatlanok.

4. Az eszköz hardverének megvalósítása

4.1 A félhíd modulok nyomtatott áramkörének megtervezése

A modulok tervezésénél a legfontosabb szempont a méretek minimalizálása volt, mivel három modulnak is el kellett férnie a vezérlőegység felületén. Egy modulon egy félhidat és egy gate-meghajtó IC-t helyeztem el, az azokhoz tartozó passzív alkatrészekkel együtt. Az áramkörtervezésnél a meghajtó áramkör bekötési rajzát vettem alapul (31. ábra) és a MOSFET-ek általános kapcsolásait. Ezek mellett a TFB (hőmérsékletvisszacsatolás) megvalósításához DS18B20 hőmérő IC-eket helyeztem el a MOSFET-ek közelében. Ezek alapján a kapcsolási rajz:



45. ábra. Egy félhíd modul kapcsolási rajza

A félhíd modulok csatlakozói egyszerű 2.54 mm raszterű tűkesorok (J1 és J4). Ez a modulszerű csatlakoztatást teszi lehetővé. A modul bemenőjeleit a bal oldalon láthatjuk J1 csatlakozónál:

1. **SPWM signal:** a mikrokontrollerről érkező SPWM jel, mely a félhíd meghajtó IN bemenetére csatlakozik
2. **SD_Pin:** a mikrokontrollerről érkező shutdown jel, mely a félhíd meghajtó $\overline{SD}$ bemenetére csatlakozik
3. **GND:** a mikrokontroller földje (logikai föld), a meghajtó Vss bemenetére csatlakozik
4. **DT_Variable_R:** a holtidő vezérlésére szolgáló ellenállás bekötésére szolgál
5. **5V:** a mikrokontrollerről érkező 5 Volt, a hőmérő IC tápellátását szolgáltatja
6. **TFB:** a hőmérő IC-ről levett jel, valójában kimenőjel
7. **+12 V:** a meghajtó áramkör működéséhez szükséges tápfeszültség

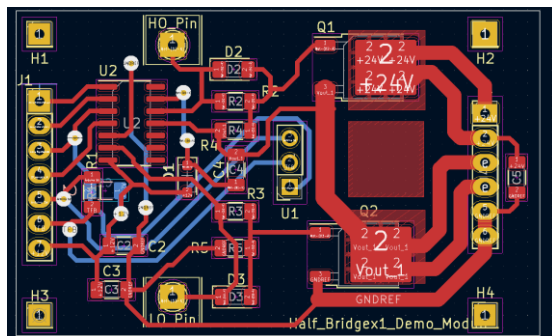
Látható, hogy a TFB lábon kimenőjelet vehetünk le, melyet a rendszer a mikrovezérlőbe vezet vissza elemzésre. Mivel a vezérlőegységen a mikrokontroller a modulok bal oldalához helyezkedik el közelebb, ezt a kimenőjelet a bemenetekhez helyeztem el.

Az áramkörben J4 csatlakozó lábain folyhatnak a legnagyobb áramok (akár 2-3 Amper is hídzárlat esetén), így itt egy vezeték a J4 tűkesor két lábához csatlakozik. Az általam használt tűkesor maximálisan 1.5 Amper áram elvezetésére képes, így két láb felhasználásával az áramtoleranciát 3 Amperre növeltem. Ezen a csatlakozón a tápegység pozitív feszültsége és a föld jelenik meg, illetve a kimenőjelet (V_out) is itt vehetjük le. A DT lábra elhelyezendő ellenállás a modulon kívül kerül bekötésre a helytakarékoság érdekében. Az áramkörben HO_Pin és LO_Pin mérőpontok, melyeket a holtidő szemléltetésére helyeztem el. Mivel HO a felsőági MOSFET vezérlőjele, LO pedig az alsóági kapcsolóelemé, a két vezérlőjel között a holtidőn kívül szintléptetés is megfigyelhető (HO magasabb feszültségű, mint LO). A demonstrációs eszközt használó ezeket mind meg tudja tekinteni a mérőpontokon keresztül.

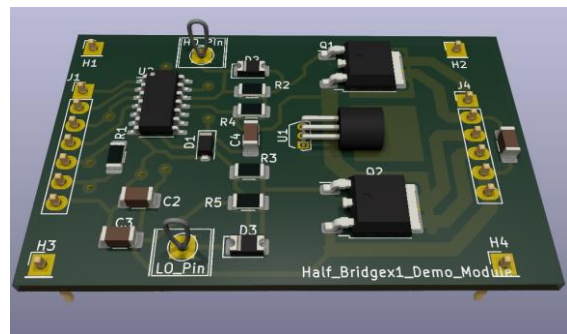
A kapcsolásban R2 és R3 gate ellenállások, melyeknek értéke általában 0-100Ω között szokott változni nagyfrekvenciás FET-eknél. A gate-ellenállás korlátozza a MOSFET-ek vezérlőelektródájára jutó áramot, így a kapcsolási idő is befolyásolható vele. Nagyobb ellenállásoknál nő a kapcsolási idő, azonban csökkenthető a MOSFET-ek parazitakomponenseiből adódó „ringing” jelenség. Ezzel szemben R4 és R5 gate-lehúzó ellenállások, melyek biztosítják a FET-ek kikapcsolt állapotát vezérlőjel hiányában. Mivel a kapcsolóelemek bekapcsolásakor töltéseket viszünk fel azoknak a vezérlőelektródájára, ha eltávolítjuk a vezérlőjelet, de nem távolítjuk el a töltéseket, a FET-ek bekapcsolt állapotban

maradhatnak. Emiatt célszerű a gate lábat egy nagy ellenálláson keresztül a földre kötni, mivel így elkerülhető a FET-ek nem tervezett bekapcsolása. D2 és D3 ellenpárhuzamos diódák a MOSFET-ek kikapcsolásának sebességét növelik, ugyanis a kikapcsolás pillanatában pozitív előfeszítést kapnak. Emiatt rövidzárként viselkednek, és utat nyitnak a gate lábakra leáramló töltéseknek, gyorsítva a kikapcsolást. A legfontosabb kimenőjelet, V_{out} -ot, a félhíd középpontjáról vehetjük le.

A nyomtatott áramkör huzalozásánál a legtöbb helyen 0.5 mm vastagságú vezetéket helyeztem el. A „nagyteljesítményű” oldalon igyekeztem a vezetékeket minél vastagabbra választani, a nagyobb áramtolerancia érdekében, így itt a vezetékek vastagsága 1.5 – 2 mm között változik. Ezáltal a Nyákáruhá weboldalán található táblázat alapján 2 - 3 Amper átfolyó áram csak 10 C° hőmérséklet emelkedéshez vezet a vezetősáv környezetében. [21]

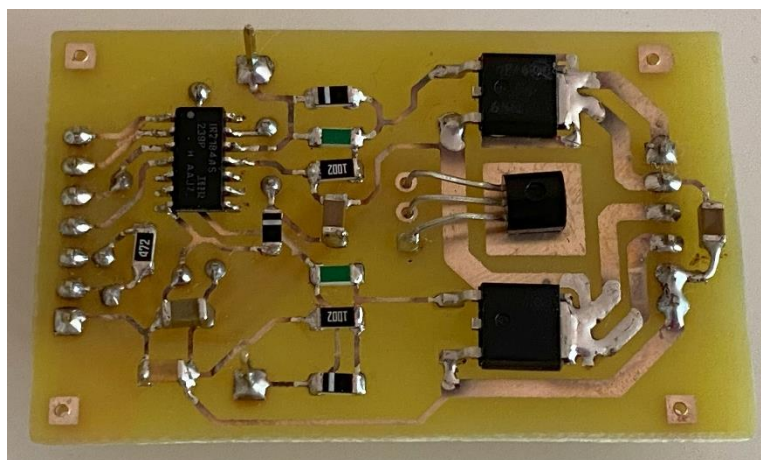


46. ábra. A félhíd modul PCB terve



47. ábra. A félhíd modul 3D modellje

A félhíd modul nyomtatott áramkörének elkészítése után az alkatrészek beforrasztása is megvalósult.



48. ábra. Az elkészített félhíd modul

4.2 A vezérlőközpont nyomtatott áramkörének megtervezése

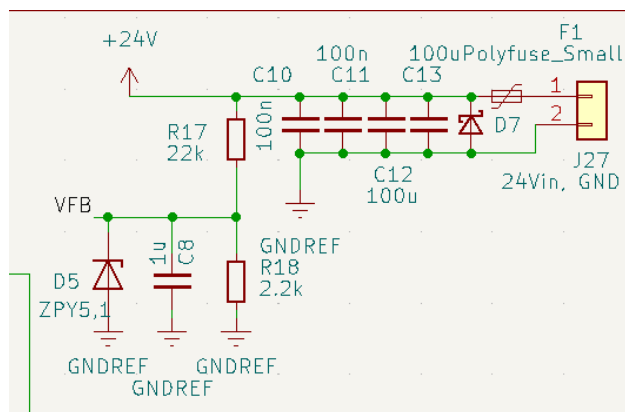
A demonstrációs eszközöm tervezésénél a hardvert további építőelemeire bontottam aszerint, hogy milyen egységeket kell tartalmazzon a vezérlőközpont. Ez alapján vezérlőeszközöket, mikrokontrollert, kijelzőt, három félhíd modul, mérőpontokat, LED-eket és be/kimeneteket különítettem el. Mivel a félhíd modulok végleges mérete 54.5 mm x 33.8 mm lett, a vezérlőközpont méretét 150 mm x 200 mm-ben maximalizáltam. Így egy nyomtatott áramkörre lehetett elhelyezni az összes vezérléshez szükséges alkatrészt és a modulokat is.

Az áramkörben a tápellátást a J27 csatlakozó biztosítja, melyre a tápegységet kell csatlakoztatni (49. ábra). A túláram és a fordított táp ellen kétlépcsős védelemmel is rendelkezik a rendszer: a fordítva bekötött tápellátás kinyitja az ellenpárhuzamosan elhelyezett D7 Schottky diódát. Ebben az esetben a dióda rövidzárként való viselkedése miatt nagy áram jelenik meg a bemenet két lába között. A túláram hatására F1 multifuse biztosíték kiold, és megszakítja az áramkörben folyó áramot a fordított táp megszűnéséig. R18 és R17 egy feszültségosztót hoznak létre a VFB megvalósítására. Mivel VFB a mikrokontrollerbe visszacsatolt jel, az értéke csak 0-5 Volt között mozoghat, hiszen ettől nagyobb feszültségek kárt tehetnek a mikrovezérlőben. A demonstrációs eszköz maximálisan 35 Volt tápfeszültség fogadására képes, ezért a VFB visszacsatolt feszültségjel maximális értéke:

$$V_{FB_{MAX}} = \frac{2.2 \text{ k}\Omega}{2.2 \text{ k}\Omega + 22 \text{ k}\Omega} \times 35 \text{ V} = 3.18 \text{ V}.$$

Annak érdekében, hogy meghibásodás esetén se kerüljön 5 Volttól nagyobb feszültség a mikrokontrollerre, R18 feszültségét D5 zener dióda is korlátozza, melynek letörési feszültsége 5.1 Volt. C8 kondenzátor a VFB visszacsatolt feszültség szűrését valósítja meg, mint aluláteresztő szűrő. C10-C13 kondenzátorok a tápegység áramának szűrését és simítását hajtják végre.

A mikrokontrollernek és a félhíd moduloknak a 12 Voltos tápfeszültséget egy LM7812 stabilizátor biztosítja, melynek maximális bemenő feszültsége 35 Volt (emiat a 35 Voltos tápfeszültség maximalizálás).

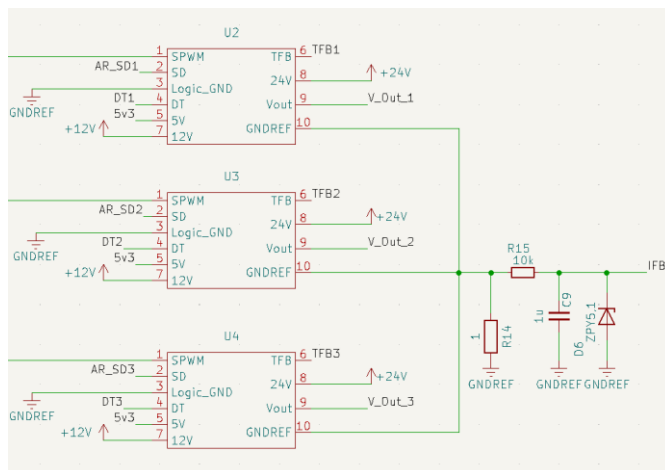


49. ábra. A vezérlőegység bemenete

A félhíd modulokon átfolyó áram visszacsatolása az IFB megvalósításához R14 1 Ω -os huzal ellenálláson keresztül történik meg (50. ábra). Mivel hídzárlat esetén 2-3 Amper áram is megjelenhet a hídágakon, ennek az ellenállásnak bírnia kell ezt az árammennyiséget. Az ellenálláson eldisszipálódó teljesítmény 2 A zárlati áram mellett:

$$P_{MAX} = I^2 * R = 2^2 \text{ A} * 1 \Omega = 4 \text{ W.}$$

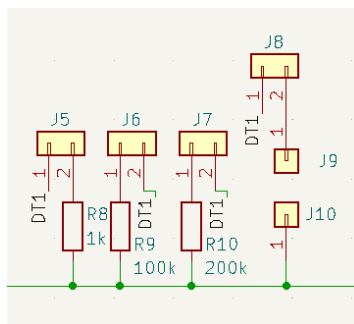
Emiatt egy 5 W teljesítménytoleranciával rendelkező huzal ellenállást választottam R14-nek. A huzal ellenállásról levehető feszültség 2 Amper zárlati áram mellett 2 Volt, melyet IFB-ként kap meg a vezérlő. Az Arduino védelmének érdekében itt is egy zener diódán keresztül csatoltam vissza a feszültségelet (D6). Az áram útjának korlátozását R15 ellenállás biztosítja, mely áramosztót valósít meg R14-el.



50. ábra. A félhíd modulok és IFB visszacsatolása

A holtidő vezérléséért az első félhíd modulban J5, J6, J7, J8, J9 és J10 csatlakozók felelnek, melyek összekötésével különféle ellenállások csatlakozhatnak a modul DT bemenetére (51. ábra). A beépített választható ellenállások 1 K Ω , 100 K Ω és 200 K Ω (a 200K Ω intervallum egyenletesen felosztva). Az ellenállásokat J5, J6 és J7 tűksorok két lábának összekötésével

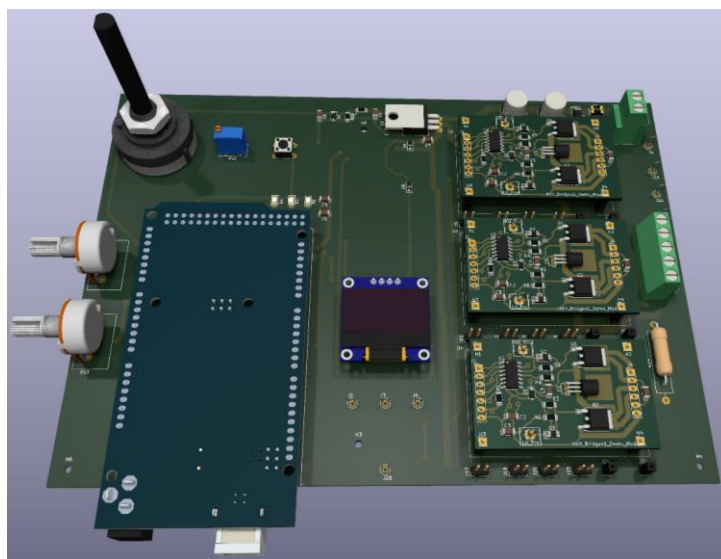
lehet bekötni, azonban J8 bekötésével tetszőleges ellenállás is köthető a félhídra (J9 és J10 közé).



51. ábra. Az első félhíd modul holtidejének beállítása a vezérlőegységen

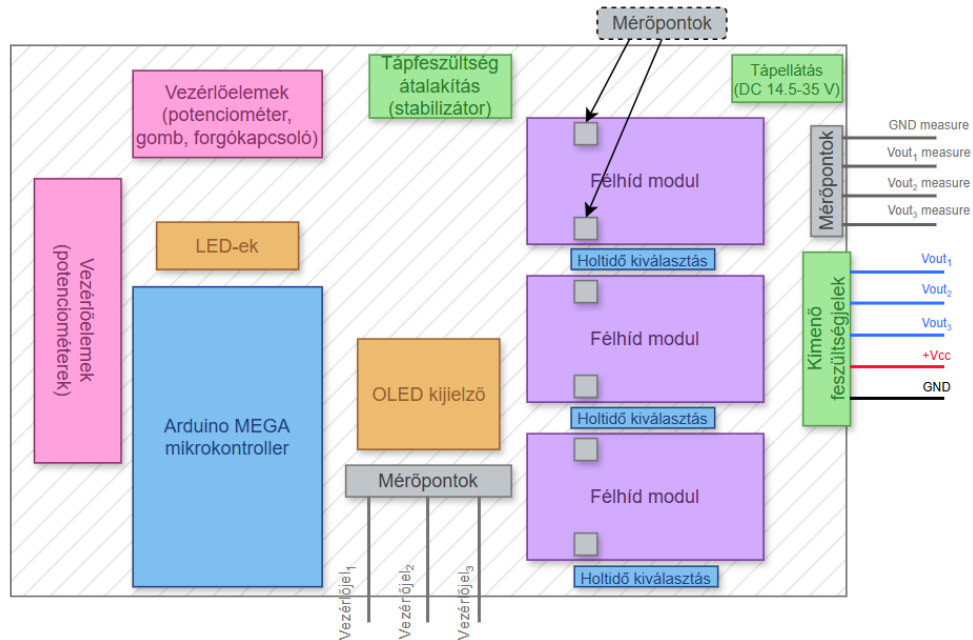
A felhasználó a szinuszhullám frekvenciáját (Hz) és a vezérlési módot (félhíd, unipoláris stb.) RV1 és RV3 egyfordulatos potenciométerek segítségével tudja változtatni (1. melléklet). Mivel a hordozójel szélesebb skálán állítható, ennek a szabályozásához egy többfordulatos trimmer potenciométer (RV2) áll rendelkezésre. A potenciométerek mind a mikrokontroller analóg bemeneteire csatlakoznak. Ezzel szemben az intervallumok kiválasztását egy négyállású forgókapcsoló teszi lehetővé, melynek lábai az Arduino digitális lábaira csatlakoznak.

A vezérlőegységen a kimenő szinuszos feszültségjelek (V_{out1} , V_{out2} , V_{out3}) J26 csatlakozóról vehetők le, melynek rászterávolsága 5.04 mm, a nagyobb áramtűrés érdekében. A PCB megtervezésénél a nagyteljesítményű területet a nyomtatott áramkör jobb oldalán helyeztem el (vastagabb vezetékekkel megerősítve), míg a logikai áramkört elemek a bal oldalra kerültek (2. melléklet).

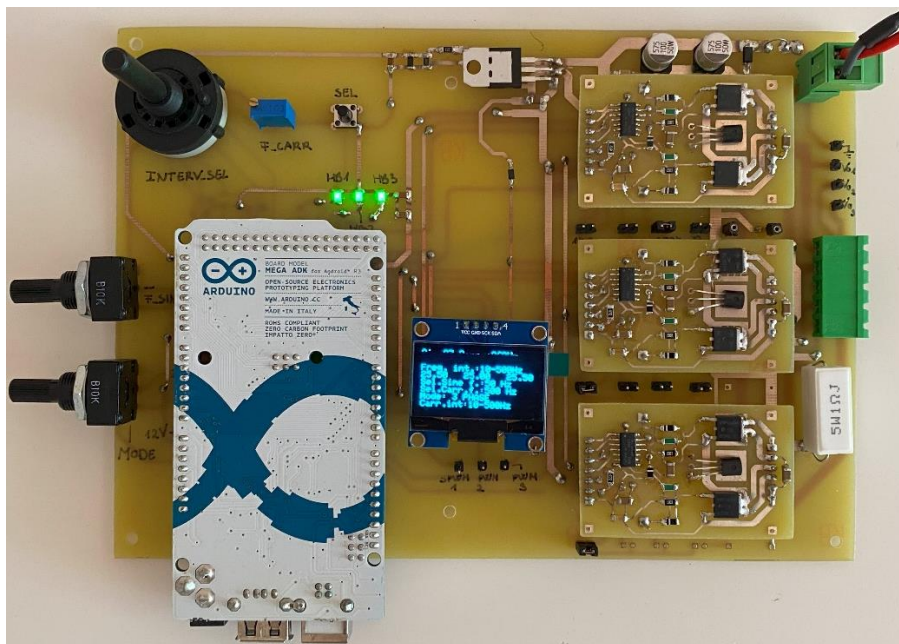


52. ábra. A vezérlőegység 3D modellje

A vezérlőegység felépítésében a félhíd modulok egymás alatt kerültek elhelyezésre. A rendszerben három különböző területen is megtalálhatóak mérőpontok, ahol a vezérlőjelek és kimenőjelek megfigyelése egyaránt lehetséges.



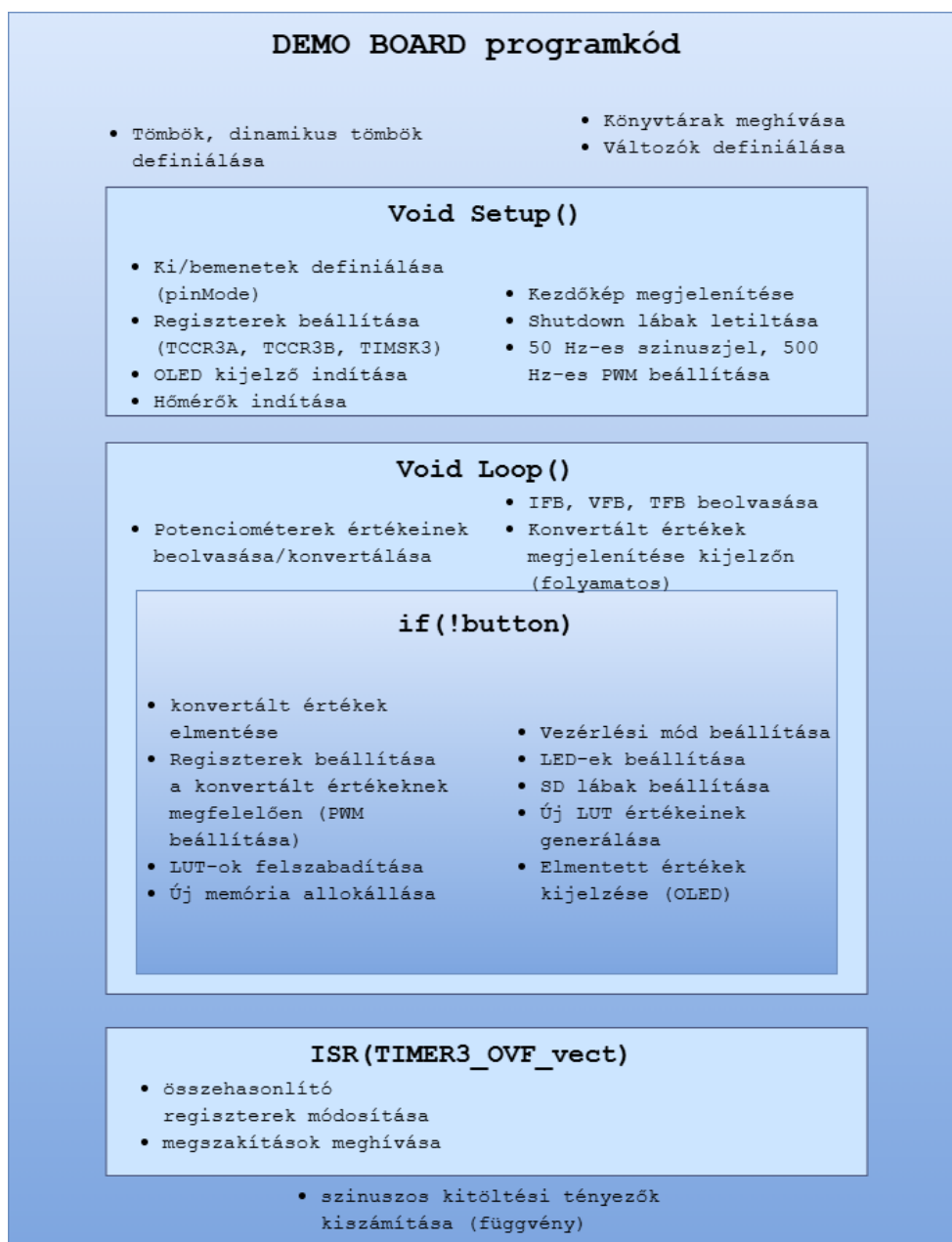
53. ábra. A vezérlőegység felépítésének blokkvázlata



54. ábra. Az elkészített vezérlőegység

5. A szoftveres vezérlés megtervezése

A program blokkvázlata

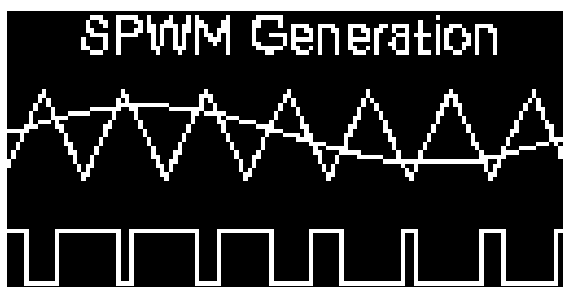


55. ábra. A program felépítése

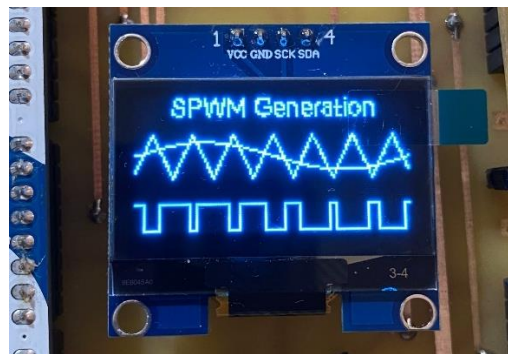
A blokkvázlatból látható, hogy a kód nagy része gombnyomás hatására fut le (55. ábra, `if(!button)`). Ezzel kerül a felhasználó által beállított SPWM a mikrokontroller lábainak kimenetére. A gomb működése aktív alacsony, így akkor fut le ez a kódrészlet, amikor a gombhoz kötött lábon 0 logikai érték jelenik meg (emiat `!button`). Ez a gomb viselkedésének definiálása miatt van, ugyanis „bemeneti felhúzás” (`input_pullup`) valósul meg rajta. Így, a gomb alaphelyzetben logikai 1-es értéket szolgáltat, megnyomás hatására pedig logikai nullát.

5.1 A kezdőkép elkészítése

Az OLED kijelzők képmegjelenítési lehetőségét kihasználva egy kezdőkép létrehozása mellett döntöttem. A kép rögtön azután jelenik meg, hogy feszültség alá helyezzük az eszközt, és az közli a felhasználóval, hogy a rendszer SPWM generáció végrehajtására alkalmas. Az OLED kijelző 128x64 pixel megjelenítésére képes, így a kezdőkép előkészítésénél ezt a méretet állítottam be képméretnek. Ahhoz, hogy az OLED meg tudja jeleníteni a képet, az csak fekete-fehér pixeleket tartalmazhat. Az előkészített képet egy *image2cpp* weboldal segítségével konvertáltam egy bájtokat tartalmazó tömbbé, ahol minden bájt egy bitje a kijelző egy pixelének felel meg. Azok a pixelek, amelyeknek világítaniuk kell, egyesként, a kikapcsolt pixelek pedig nullaként jelennek meg a bájtömbben.



56. ábra. A 128x64 bites fekete-fehér kép



57. ábra Az OLED kijelzőn megjelenített kép

Maguk a bájtok hexadecimális formátumban vannak eltárolva a bájtömbben, 0x azonosítóval. Egy sort 16 bájtömb reprezentál, így a 16 bájtömb 8 bitje alkotja a 128 bites sort. A programkód a kezdőképet a változók inicializálása után hívja meg, még a `setup()` függvény keretein belül, a `display.drawBitmap()` és a `display.display()` parancs segítségével.

A bájtömb megjelenítése a programkódban:

```
static const unsigned char PROGMEM spwm_logo[] =
{0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x1c, 0x7c, 0x84, 0x50, 0x20, 0x7e, 0x00, 0x00, 0x00, 0x00,
0x20, 0x00, 0x00, 0x00, ...}

...

display.drawBitmap(0, 0, spwm_logo, 128, 64, 1);
display.display();
delay(4000);
```

A `display.drawBitmap()` függvény első két paramétere (0, 0) a kép kezdőkoordinátáit adják meg (jelen esetben a képet a kijelző bal felső sarkától kezdi el kirajzolni a program). Ezután a bájtömb neve következik (`spwm_logo`) és a mérete pixelekből (128, 64). A legutolsó paraméter azért felel, hogy a pixelek kirajzolásra kerüljenek a képernyőn (1).

A felhasznált *image2cpp* weboldal: <https://javl.github.io/image2cpp/>

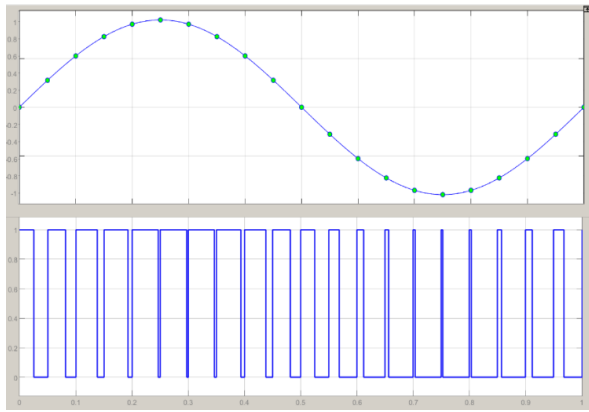
5.2 A szinuszos moduláció létrehozásának lépései

A szoftveres vezérlésnél a legfontosabb feladat a szinuszos impulzusszélesség moduláció megvalósítása volt. A programkódban az SPWM regiszterprogramozással és megszakítások felhasználásával valósul meg. A megszakítások teszik lehetővé, hogy a mikrokontroller által generált frekvenciák megközelítsék valós időben a felhasználó által kiválasztott frekvenciákat. Ezt úgy éri el a program, hogy az ICR3 és TCCR3B regiszterekben tárolt értékek megfelelően beállítja a PWM (vagyis a hordozójel) frekvenciáját, és az OCR3x kimeneti összehasonlító regiszterekben tárolt értékek szerint meghívja a megszakításokat. Mivel az OCR3x regiszterek a kitöltési tényezők beállításáért felelnek, a szinuszhullám előzetes mintavételezéséből származó értékeket kapják meg. Így érhető el az, hogy a kitöltési tényezők szinuszosan változzanak, és a PWM alapjele egy szinuszhullám legyen.

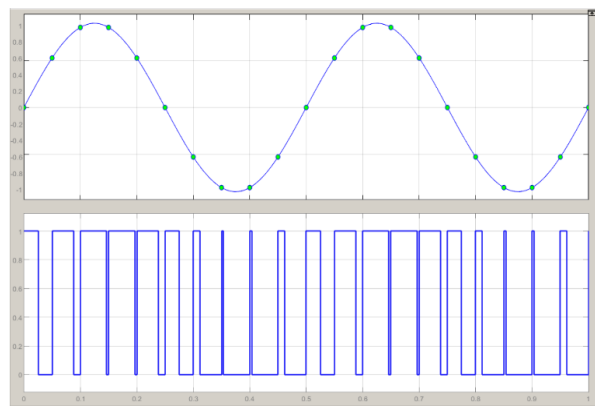
A szinuszhullámból vett minták számának meghatározása

A szinuszhullámból vett minták száma a Look Up Table táblázatok elemeinek számát határozza meg. A mintavételezés legegyszerűbben egy 1 Hz-es és egy 2 Hz-es szinuszhullám 20 Hz-es hordozójellel történő átvitelén szemléltethető:

A 20 Hz-es PWM beállításával azt érjük el, hogy a mikrokontroller egy másodperc alatt 20 impulzust generáljon. Az 1 Hz-es szinuszhullám esetében 20 Hz-es PWM mellett 20 mintát kell vennünk a szinuszhullámból, és e minták függvényében kell változtatnunk a kitöltési tényezőket. A 2 Hz-es szinuszejel esetében azonban nem szükséges 20 minta vétele, mivel a hullám kétszer ismétlődik egy másodpercen belül. Emiatt a hullámból elegendő $20 / 2 = 10$ minta kinyerése. Ezt a legegyszerűbben az 58. és 59. ábra szemlélteti:



58. ábra. Egy 1 Hz-es szinuszjel 20 Hz-es hordozóval történő átvitele



59. ábra. Egy 2 Hz-es szinuszjel 20 Hz-es hordozóval történő átvitele

Természetesen a második esetben a szinuszjel felbontása romlik, mivel csak 10 impulzussal képezzük le 20 helyett. Ilyenkor, ha azonos felbontásban szeretnénk átvinni a két szinuszjelet, a második esetben a kétszeres hullámfrekvenciához kétszeres hordozójel frekvenciát kell beállítanunk. A legfontosabb következtetés a fent leírtakból, hogy a szinuszhullámból vett minták darabszáma megegyezik a hordozójel frekvenciájának és a szinuszhullám frekvenciájának hányadosával. Ez a programkódban az alábbi módon jelenik meg:

```
float AMPLITUDE_SINE = 0.9;
float FREQUENCY_SINE;
int FREQUENCY_CARRIER;
int NUM_SAMPLES;
...
NUM_SAMPLES = (int)(FREQUENCY_CARRIER/FREQUENCY_SINE);
```

Ahol NUM_SAMPLES a minták darabszáma, FREQUENCY_CARRIER a hordozójel frekvenciája és FREQUENCY_SINE a szinuszhullám frekvenciája. Mivel a LUT elemeinek száma csak egész szám lehet, az (int) típuskényszerítés elengedhetetlen az osztás elvégzése után. Így természetesen, ha nem egész szám jön ki hányadosnak, veszítünk valamennyi információt az eredeti jelből, de ez nagy felbontás esetében elhanyagolható. A szinuszhullám amplitúdóját 0.9-re állítottam be, a túlmoduláció elkerülésének érdekében ($m_a < 1$). A LUT leképzési módszerében látható, hogy a LUT tömb elemeinek száma növekszik a szinuszhullám frekvenciájának csökkenésével és a hordozójel frekvenciájának növelésével. Mivel az eszközömön szemléltetni szeretném az 1 Hz-es szinuszjelet és a 30 KHz-es hordozójelet is, intervallumokra osztottam fel a szinuszjel-hordozójel kombinációkat a

programkódomban. Így kerültem el azt, hogy a program esetlegesen 30 000 elemű tömböt hozzon létre LUT-ként. Az intervallumok, amelyeket beállítottam a következők:

	I. intervallum	II. intervallum	III. intervallum	IV. intervallum
A szinuszjel választható frekvenciái	1 Hz – 50 Hz	50 Hz – 150 Hz	150 Hz – 250 Hz	50 Hz – 150 Hz
A hordozójel választható frekvenciái	10 Hz – 500 Hz	500 Hz – 1 KHz	1 KHz – 10 KHz	10 KHz – 30 KHz
A lépték mértéke	10	10	100	500

3. táblázat. A rendszer által beállítható intervallumok

A léptékek beállítását a széles skála lefedése indokolta. A legutolsó intervallumban 20 KHz különbséggel lehet hordozójel frekvenciát változtatni. Itt nem sok különbséget észlelnénk 20000 és 20010 Hz között, emiatt a lépték ötszázásával nő, vagyis 10000 Hz után a következő beállítható hordozójel frekvencia 10500 Hz. Ezeket az intervallumokat a felhasználó a forgókapcsoló segítségével állíthatja be, mely négy pozícióval rendelkezik a négy intervallum kiválasztáshoz. Az első intervallumot felhasználva szemléltetésként, a programkódban az adott módon történik meg az intervallum beállítása:

```
if(digitalRead(interval_1)){
    n = 10;
    BottomValSine = 1;
    TopValSine = 50;
    BottomValCarr = 10;
    TopValCarr = 500;
    carrier_sel=0;
    strcpy(interval_txt, "10-500Hz");
}
```

Amennyiben interval_1 változó, amely a forgókapcsoló első lábáról kapja a jelet, logikai 1-est kap (vagyis a kapcsoló ebbe a pozícióba van állítva) a szinuszjel és a hordozójel alsó és felső határértékei módosításra kerülnek (BottomValSine, TopValSine, BottomValCarr, TopValCarr). A carrier_sel változó nulla értéket kap, melynek később az előosztó beállításánál lesz szerepe, interval_txt char tömbbe pedig a megfelelő intervallum szöveges megnevezése kerül elhelyezésre az strcpy() (string copy) parancs segítségével.

A potenciométerekről az értékek beolvasása az analogRead() függvény segítségével történik meg. A parancs 0-1023 között ad vissza értékeket a lábról leolvasott feszültség

függvényében. Ezután a map() eljárás segítségével a 0-1023-ig leolvasott adatokat a program átskálázza az intervallumok megfelelő értékeire (BottomVal és TopVal közé):

```
float carr = map(analogRead(f_carr_sel), 0, 1010, BottomValCarr,
TopValCarr);
carr = ((int)carr + (n/2)) / n * n;
float f_sine = map(analogRead(f_sine_sel), 0, 1020, BottomValSine,
TopValSine);
```

Az f_carr_sel és f_sine_sel változók a hordozójel és a szinuszjel frekvenciájának beállítására szolgáló potenciométerekre vannak kötve a PCB-n. A carr=(int)carr ... sor szerepe a léptékek bevezetése. Amennyiben tízesével szeretnénk változtatni a hordozójel frekvenciáját n=10. Ilyenkor a potenciométerről beolvasott értéket a program a legközelebbi tízes értékre kerekíti (pl.: 17 esetében 20-ra, de 34 esetében 30-ra).

ICR3 regiszter és a prescaler beállítása TCCR3B regiszter segítségével

A programkódban a hordozójel tényleges beállítása a regiszterek beprogramozásával történik meg. Ehhez az ATmega2560 adatlapját [15] és szakirodalomban is vizsgált képletet használtam fel, mely szerint:

$$f_{\text{PWM}} = \frac{f_{\text{CLOCK}}}{\text{prescaler} \times (\text{timer}_{\text{TOP}} + 1)}$$

Ehhez a PWM jel frekvenciájának beolvasása a potenciométerről történik meg a fent tárgyaltak alapján. Így a feladat ICR3 értékének meghatározása, ami a timer_{TOP} szerepét tölti be a képletben. Ehhez a rendezett egyenlet:

$$\text{timer}_{\text{TOP}} = \frac{f_{\text{CLOCK}}}{\text{prescaler} \times f_{\text{PWM}}} - 1$$

A különböző intervallumok bevezetésével az előosztó (prescaler) értékeit is változtatom, mivel bizonyos frekvenciák csak bizonyos előosztókkal érhetőek el. Előosztó nélkül (prescaler = 1) a legkisebb elérhető frekvencia 244 Hz, de már egy 256-os előosztóval is megvalósítható a 10 Hz-es PWM. A nagyfrekvenciákhoz, mint amilyen a 10-30 KHz-es intervallum, nincsen szükség előosztó használatára. Ez alapján az intervallumok függvényében ICR3 kiszámítása az alábbi módon történik a programban:

```

switch (carrier_sel) {
    case 0:
        calc_icr3 = (16000000.0/(256.0*FREQUENCY_CARRIER))-1.0;
        ICR3      = (int)calc_icr3;
        TCCR3B = 0b00011100;
        strcpy(interval_txt_sel, "10-500Hz");
        break;

    case 1:
        calc_icr3 = (16000000.0/(64.0*FREQUENCY_CARRIER))-1.0;
        ICR3      = (int)calc_icr3;
        TCCR3B = 0b00011011;
        strcpy(interval_txt_sel, "500-1kHz");
        break;

    ...

```

A `carrier_sel` változó a forgókapcsoló állásának függvényében kap értéket 0-3-ig, az intervallumoknak megfelelően. Az első állapotban (case 0) a hordozójel frekvenciái 10-500 Hz-ig vehetnek fel értékeket, amihez a kódban egy 256-os előosztót használtam fel. A `calc_icr3` változó kiszámítása a fent is említett `timerTOP` kiszámításának képletén alapszik, majd ICR3 regiszter `calc_icr3` típuskényszerített verzióját kapja meg. Mivel `calc_icr3` kiszámítása csak egy programon belüli számítás és nincs összeköttetésben a PWM regiszterekkel, a 256-os előosztó a `TCCR3B = 0b00011100` sorban kerül ténylegesen rögzítésre. A többi intervallum beállításakor is hasonlóan jár el a programkód, az utolsó két case() esetében az előosztó 1-esre történő beállításával.

A vezérlési módszer függvényében a LUT tömbök generálása

A kitöltési tényezőket tartalmazó Look Up Table létrehozása az intervallumok és ICR3 regiszter beállítása után történik meg. Miután a felhasználó kiválasztja a frekvenciákat, amelyeket be szeretne állítani, és a program kiszámolja a fent tárgyalt változókat, a mintavételezés és a LUT feltöltése egy saját készítésű függvénnyel valósul meg, melyet `sine_duty_cycle`-nek neveztem el. A függvény visszatérési értéke egy integer, amely egy kitöltési tényezőnek felel meg. Emiatt a függvény `for()` ciklusban futtatandó, hogy az összes kitöltési tényező legenerálásra kerüljön. A függvény felépítése:

```

int sine_duty_cycle(float AMPLITUDE, float PHASE, int inc, int SAMPLES,
int icr3_val){
    float sine_val;

```

```

float samp_time = 1/((float)SAMPLES);
float norm_sin_scaled;
int duty_cycle;
float phase_shift;
phase_shift = (PHASE*PI)/180;
sine_val = AMPLITUDE * sin(2 * PI * samp_time*inc+phase_shift);
norm_sin_scaled = ((sine_val + 1)/2)*icr3_val;
duty_cycle = (int)norm_sin_scaled;
return duty_cycle;
}

```

A függvény a mintavételezendő szinuszjel amplitúdóját (AMPLITUDE), és fázisát (PHASE) fogadja bemenetként. Ezen kívül egy inc változót, ahova a for() ciklus inkrementálandó tényezője kerül (általában i), egy SAMPLES változót, ami a minták darabszámát képviseli és ICR3 értékét, amely a kitöltési tényezők skálázásáért felelős. A program először a mintavételezési időt állítja elő, ami a minták darabszámának reciprokával egyenlő ($1/(\text{float})\text{NUM_SAMPLES}$). A fázistolást a függvény fokban fogadja, de átszámolja radiánba a phase_shift változó meghatározásakor $((\text{PHASE} * \text{PI}) / 180)$. A minta létrehozása a sine_val változó beállításánál történik a sin() függvény segítségével, melyet a math.h könyvtár tartalmaz. A samp_time megszorzása az inc-el lehetővé teszi azt, hogy adott időpontokban vegyünk mintát a jelből (inc = 4 esetén a 4. időpontban, ami 0.1 samp_time mellett a 0.4. másodperc).

A norm_sin_scaled tényező a minták normalizálását végzi el [0, 1] intervallumra. Ezeket az értékeket ezután ICR3 megszorzásával felskálázzuk a megfelelő méretekre. Erre a műveletre a Timer3 működése miatt van szükség, mivel az időzítő maximális (100%-os) kitöltési tényezője maga ICR3. Emiatt az aktuális kitöltési tényező mindig ICR3 és a normalizált értékek szorzatával egyenlő. Ennek az értéknek a típuskényszerített verzióját adja vissza a sine_duty_cycle() függvény.

A LUT-ok tényleges beállítása a kiválasztott vezérlési módtól függ. A program az alábbi vezérlési módok kezelésére képes:

	Félhíd	Bipoláris	Unipoláris	3 fázisú	OFF
A felhasznált félhidak száma	1	2	2	3	0
Aktív LED-ek	LED1	LED1, LED2	LED1, LED2	LED1, LED2, LED3	egyik sem
Aktív shutdown-ok (SD)	SD_2, SD_3	SD_3	SD_3	egyik sem	SD_1, SD_2, SD_3

4. táblázat. A rendszer által kezelt vezérlési módok

Ehhez a program az alábbi módon generálja le a LUT-okat:

```
switch (mode_sel) {
...
case 2: //unipolar pwm
    digitalWrite(SD_1, HIGH);
    digitalWrite(SD_2, HIGH);
    digitalWrite(SD_3, LOW);
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
    for (int i = 0; i < currentSize; i++) {
        LookUp1[i]=sine_duty_cycle(AMPLITUDE_SINE, 0, i, NUM_SAMPLES, ICR3);
        LookUp2[i]=sine_duty_cycle(AMPLITUDE_SINE, 180, i, NUM_SAMPLES, ICR3);
        LookUp3[i] = 0;
    }
    strcpy(mode_sel_txt, "UNIPOLAR");
    break;
...
}
```

Az unipoláris modulációhoz csak két félhíd működtetésére van szükség, így LookUp3 dinamikus tömb értékeit nullákkal tölti fel a program. LookUp2 létrehozásakor a kód 180 fokos fázistolást állít be a sine_duty_cycle függvényen belül. Emellett, az SD lábak a gate-meghajtó áramkör shutdown lábaira csatlakoznak, és ennél a modulációnál a harmadik félhíd FET-jei kinyitásra kerülnek (digitalWrite(SD_3, LOW)). A mode_sel_txt az OLED kijelző által kerül megjelenítésre, hogy tudassa a felhasználóval az aktuális vezérlési módszert.

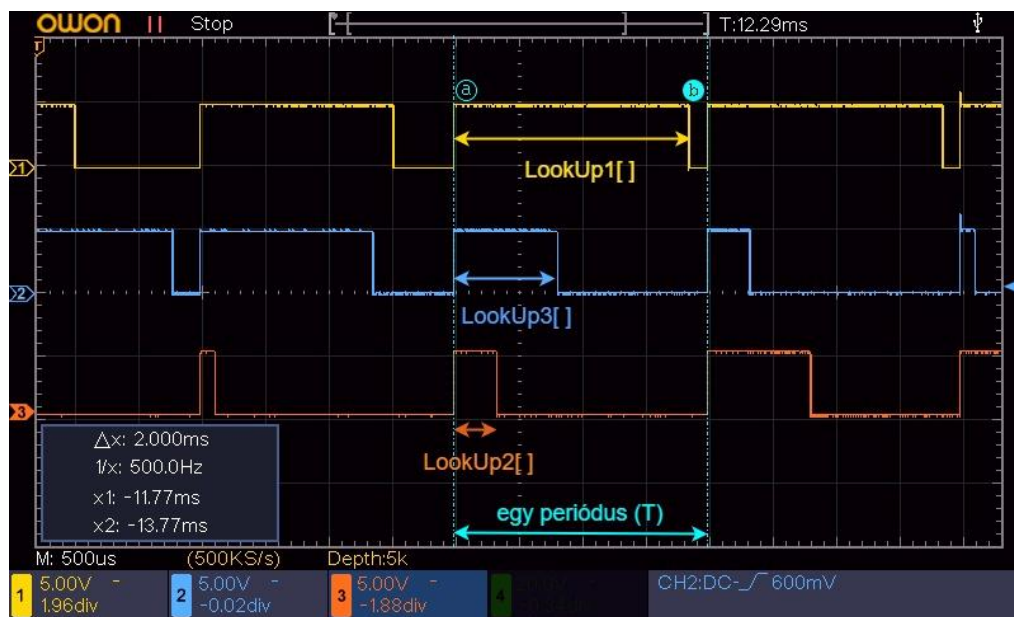
A megszakításkezelő működtetése és a dinamikus tömbök

A megszakításokat az ISR(TIMER3_OVF_vect) megszakításkezelő rutin végzi el, amely egy túlcsordulási „overflow” rutin. A műveletben a szinuszhullám mintavételezéséből származó dinamikus tömbök szerepelnek (LookUp1, LookUp2 és LookUp3). OCR3A, OCR3B és OCR3C „kimeneti összehasonlító” regiszterek (output compare registers). Amikor Timer3 számlálójának értéke megegyezik e regiszterek értékével, az összehasonlító regiszterekhez tartozó lábak kimenőjelei nullázódnak. OCR3A a mikrokontroller 5-ös lábának kimenőjelét vezérli, OCR3B a 2-es lábat, OCR3C pedig a 3-as lábat. A rutin végigfut a LUT tömbök összes elemén, majd a végükre érve előlről kezdi a folyamatot. Ameddig a felhasználó nem változtatja meg a LUT-ok értékeit, az eljárás folytatódik.

```

ISR(TIMER3_OVF_vect) {
    static int num;
    OCR3A = LookUp1[num];
    OCR3B = LookUp3[num];
    OCR3C = LookUp2[num];
    if(++num >= NUM_SAMPLES) {
        num = 0;          //ha végigment a tömb összes elemén, kezdje előlről
    }
}

```



60. ábra. Háromfázisú 50 Hz-es szinuszhullámokat továbbító 500 Hz-es PWM

A 60. ábrán is látható, hogy a három láb különböző kitöltési tényezőket kap ugyanabban az időpillanatban a jelek 120 fokos fázistolása miatt. Ilyenkor LookUp1, LookUp2 és LookUp3 más értékekkel rendelkeznek, amelyek miatt a megszakítások különböző időpillanatokban kerülnek meghívásra.

A dinamikus tömbök használatára a tömbök méretének folyamatos változása miatt volt szükség. Mivel LookUp1, LookUp2 és LookUp3 értékeinek száma a hordozójel és a szinuszhullám frekvenciáinak hányadosával egyenlő, ahogyan ezek az értékek változnak, a hányados is változik. A dinamikus tömbök a programkód elején kerülnek definiálásra, és minden gombnyomáskor felszabadítja őket a program, majd új méretet rendel hozzájuk, és feltölti őket az új frekvenciáknak megfelelően.

```
int* LookUp1=NULL;
```

```

int* LookUp2=NULL;
int* LookUp3=NULL;
... // vezérléstől függően a tömbök feltöltése (unipoláris/bipoláris stb.)
... // gombnyomás hatására új értékek betöltése
int newSize = NUM_SAMPLES;
    if (newSize > 0) {
        free(LookUp1);
        free(LookUp2);
        free(LookUp3);
    }
...
    LookUp1 = (int*) malloc(newSize * sizeof(int));
    LookUp2 = (int*) malloc(newSize * sizeof(int));
    LookUp3 = (int*) malloc(newSize * sizeof(int)); }

```

A programban a `newSize` változó értéke megegyezik a minták darabszámával (`NUM_SAMPLES`), így a dinamikus tömbök mérete is ezzel egyenlő (pl.: 10 minta esetén a LUT-ok 10 számot tartalmaznak). A `malloc()` paranccsal a programkód allokal a minták számának megfelelően az integer változóknak elegendő helyet (`newSize*sizeof(int)`). Gombnyomás hatására, ha az új tömb mérete nem nulla, a program felszabadítja a dinamikus tömböknek allokalált memóriát, és foglal nekik helyet az új méretnek megfelelően.

5.3 IFB, VFB és TFB megvalósítása

A rendszeremet áramvisszacsatolással (IFB), feszültségvisszacsatolással (VFB) és hőmérsékletvisszacsatolással (TFB) is elláttam. E visszacsatolt értékek funkciója az áramkör védelme a szélsőséges esetek ellen. Az áramvisszacsatolás a félhidak csomópontjában elhelyezett huzal ellenállásról levett feszültséget méri. Amennyiben hídzárlat következne be, az 1 Ω -os ellenálláson 2 Amper átfolyó áram esetén 2 Volt feszültség mérhető, Ohm törvénye alapján:

$$U_R = I_R * R, \text{ így } U_R = 2 \text{ A} * 1 \Omega = 2 \text{ Volt.}$$

Ezt a feszültséget a programban egy `analogRead()` függvény méri, melynek a visszatérési értéke 2 Amper esetén:

$$\text{analogRead()} = \frac{2V}{5V} * 1023 \sim 409.$$

Így a program 409 értéknél nyitja ki a félhidakat alkotó MOSFET-eket. A program azt nem tudja megmondani, melyik hídág zárlatos, mivel az összes hídágon egyszerre keresztülfolyó áramot méri. VFB hasonlóképpen működik, azonban 35 Voltra a feszültségosztó miatt az `analogRead()` visszatérési értéke 650. Ez a következő képletből számítható ki:

$\frac{2,2\text{ k}\Omega}{2,2\text{ k}\Omega + 22\text{ k}\Omega} * 35\text{ V} = 3,18\text{ Volt}$. Ez a feszültség esik a feszültségosztó 2,2 K Ω -os ellenállásán.

Ebben az esetben $\text{analogRead}() = \frac{3,18\text{ V}}{5\text{ V}} * 1023 \sim 650$. A programkód ezáltal 650-nél nyitja ki a MOSFET-eket és jelez a felhasználónak a túlfeszültség jelenléte miatt. A hőmérsékletmérő integrált áramkörökhöz készített programkód Celsiusban adja vissza a hőmérsékleteket. Ez a kódrészlet 30 fokos hőmérséklet mérése esetén kapcsolja le a félhidakat. Mivel mindhárom félhíd tartalmaz hőmérőt, a programkód képes megmondani a felhasználónak, hogy melyik félhíd melegegett túl működés közben.

IFB működése	VFB működése	TFB működése (csak T1 példájával)
<pre> if (IFB_val > 409) { digitalWrite (SD_1, LOW); digitalWrite (SD_2, LOW); digitalWrite (SD_3, LOW); display.clearDisplay(); display.print ("IFB ACTIVE"); display.display(); delay (10000); display.clearDisplay(); } </pre>	<pre> if (VFB_val > 650) { digitalWrite (SD_1, LOW); digitalWrite (SD_2, LOW); digitalWrite (SD_3, LOW); display.clearDisplay(); display.print ("VFB ACTIVE"); display.display(); delay (10000); display.clearDisplay(); } </pre>	<pre> if (temperature1 > 30.0 temperature2 > 30.0 temperature3 > 30.0) { digitalWrite (SD_1, LOW); digitalWrite (SD_2, LOW); digitalWrite (SD_3, LOW); display.clearDisplay(); display.println ("TFB ACTIVE"); if (temperature1 > 29.0) { display.println ("Bridge_1 overheat"); ... } </pre>

5. táblázat. A védelmek működése

5.4 A demonstrációs eszköz tesztelése

5.4.1 Az SPWM és a vezérlési módok tesztelése

Az SPWM tesztelésénél a cél különböző frekvenciájú szinuszjelek beállítása különböző frekvenciájú hordozójelek mellett. A rendszer működését először egy 10 Hz-es szinuszhullám 200 Hz-en történő átvitelével teszteltem, amelyen jól látszódnak a hullámformák. A kimenőjeleket J19, J21 és J22 mérőpontokról vettem le (1. melléklet). A PWM meghatározása ezen értékek mellett (elméleti síkon):

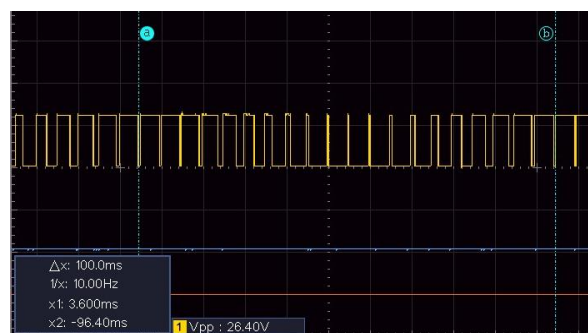
$$200 \text{ Hz} = \frac{16\,000\,000}{256 * (ICR3+1)} \rightarrow ICR3 = 311,5 \rightarrow (\text{int})ICR3 = 311$$

$$f_{\text{PWM}} = \frac{16\,000\,000}{256 * (311+1)} = 200.32 \text{ Hz (hordozójel frekvencia)}$$

Mivel $200 \text{ Hz} / 10 \text{ Hz} = 20$ minta, a szinuszhullámot a jel 20 impulzus alatt továbbítja. E frekvenciák mellett a félhidas, bipoláris és unipoláris vezérlési módokat is teszteltem, és oszcilloszkópon tekintetem meg a generált hullámformákat. A háromfázisú esetben a fázisfeszültségek méréséhez rezisztív csillagkapcsolt terheléseket helyeztem el, és a hordozójel frekvenciáját 500 Hz-re emeltem a megfelelő tesztelés érdekében.



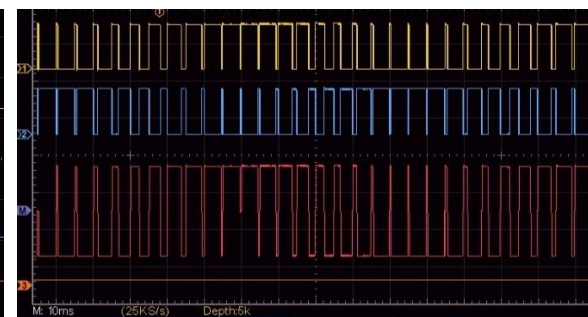
61. ábra. A PWM frekvenciája (200.4 Hz)



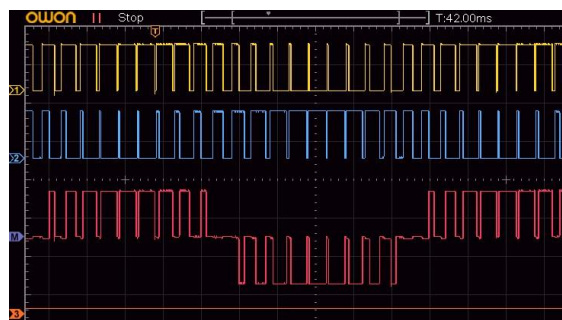
62. ábra. A szinuszhullám továbbítása (20 impulzus/szinuszhullám)



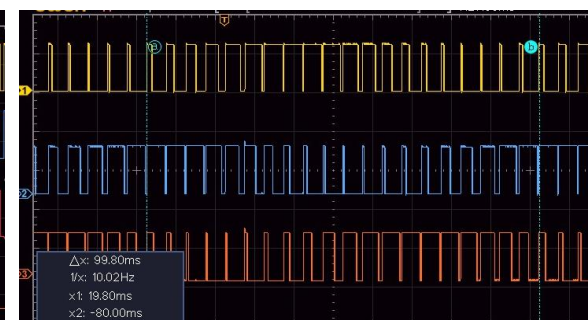
63. ábra. Félhíd vezérlés (CH1)



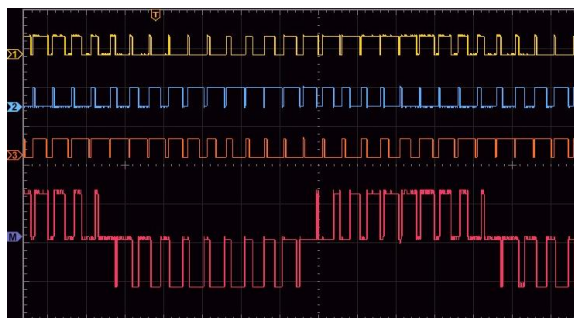
64. ábra. Bipoláris vezérlés (CH1-CH2, rózsaszín jel)



65. ábra. Unipoláris vezérlés (CH1-CH2, rózsaszín jel)



66. ábra. Háromfázisú vezérlés (CH1, CH2, CH3)



67. ábra. Háromfázis vonali feszültség (CH1-CH2)



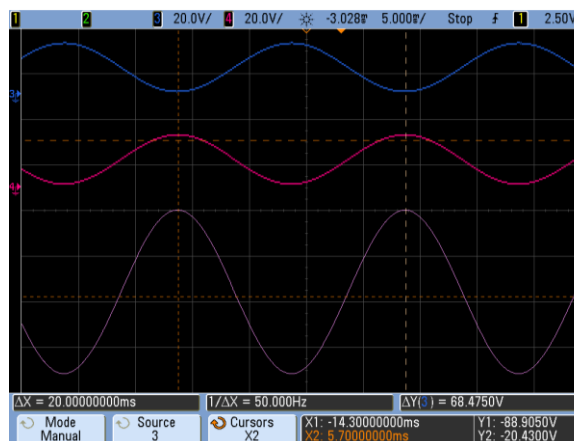
68. ábra. Háromfázis fázisfeszültség (CH1-CH2)

A bipoláris, unipoláris és háromfázisú jelek vizsgálatakor az oszcilloszkópon a csatornák különbségi jeleit állítottam elő (CH1-CH2 a matematikai kivonás funkció segítségével). A kivonás eredménye mindig rózsaszínnel jelent meg a kijelzőn.

Az alaphfrekvencia nagyobb frekvenciákon figyelhető meg a legjobban. A 69. és 70. ábra méréseit 20 KHz-es hordozójel mellett végeztem el, amely egy 50 Hz-es szinuszjelet állított elő. Ennek a létrehozásához a demonstrációs eszköz a 4. intervallumban volt (10KHz-30KHz hordozójel) és unipoláris modulációt valósított meg.



69. ábra. 20 KHz-es PWM, 50 Hz-es szinuszhullám, unipoláris moduláció (CH1-CH2)



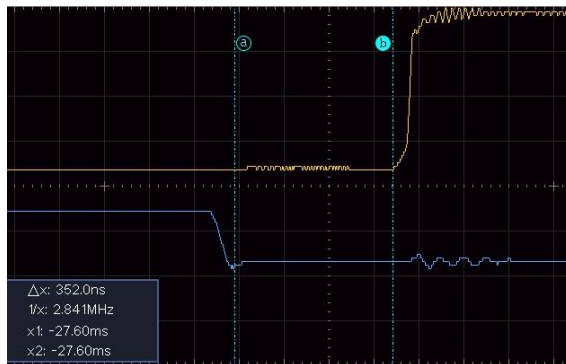
70. ábra. 20 KHz-es PWM, 50 Hz-es szinuszhullám, unipoláris moduláció (CH1-CH2)

Látható, hogy a két felső szinuszjel egymáshoz képest 180 fokos fázistolásban van, a matematikai különbségük pedig kétszintes jel (világos rózsaszín, unipoláris jel).

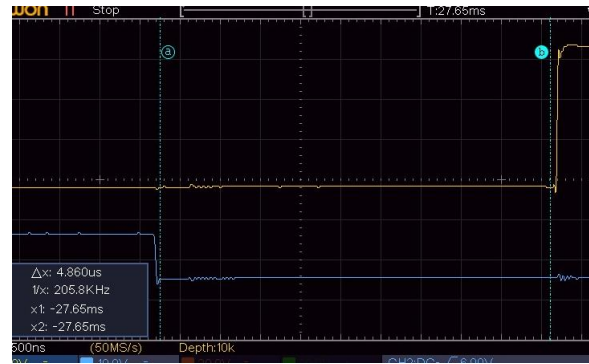
5.4.2 A holtidő tesztelése

A holtidő teszteléséhez a beépített 1 K Ω -os és 200 K Ω -os ellenállásokat használtam fel. Az oszcilloszkóp mérőfejeit HO_Pin és LO_Pin mérőpontokra csatlakoztattam. A beállított vezérlés egy 20 KHz-es hordozójelet és egy 60 Hz-es szinuszhullámot valósított meg.

A számítások alapján a PWM periódusideje $\sim \frac{1}{20\,000} = 50\text{ us}$. Emellett a nagyobb holtidőknél lényeges különbség látható a HO és LO vezérlőjelek között.



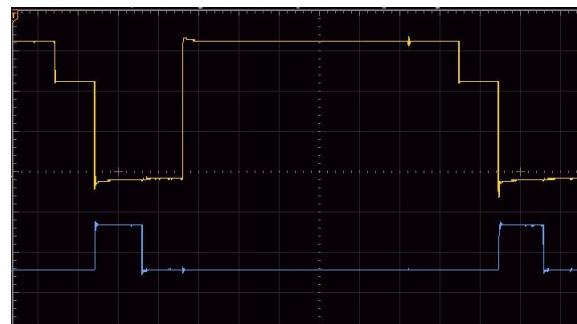
71. ábra. $\sim 350\text{ ns}$ holtidő 1K Ω -os ellenállás mellett



72. ábra. $\sim 4.86\text{ us}$ holtidő 200K Ω -os ellenállás mellett



73. ábra. $\sim 350\text{ ns}$ holtidő 1K Ω -os ellenállás mellett



74. ábra. $\sim 4.86\text{ us}$ holtidő 200K Ω -os ellenállás mellett

Összefoglalás

A szakdolgozatom fő célja egy demonstrációs eszköz létrehozása volt, amely segít a felhasználójának a különféle inverteres technikák elsajátításában. A rendszer szoftverének és hardverének megtervezése, megépítése és tesztelése után kijelenthető, hogy a demonstrációs eszköz képes végrehajtani azokat a feladatokat, amelyeket a szakdolgozatom elején kitűztem célnak. Képes a hordozójel és a szinuszjel frekvenciáinak tetszőleges szabályozására, és ezeket a frekvenciákat valós időben megközelíti a kimenőjelek generációjakor. A különféle vezérlőjelek feszültségfüggvényei is kirajzolódnak a szimulációs eredményeknek és a szakirodalomban tárgyaltaknak megfelelően. Természetesen az eszköz nem tökéletes, és egy olyan komplex témában, mint az inverterek és azok vezérlése, mindig van lehetőség fejlesztések elvégzésére. Az eszközt akár egy igazi inverter teljesítménytoleranciájával is el lehetne látni, vagyis a hardverében nagyfeszültségű és nagyáramú fejlesztéseket bevezetni (nagyobb teljesítménytoleranciájú alkatrészek elhelyezése, nagyteljesítményű védelmek beiktatása). Emellett megfordult a fejemben a tervezés során a soros kommunikációval történő vezérlés is, vagyis valamilyen számítógépes programmal (akár LabView-val) egy vezérlőszoftver létrehozása. Ez valószínűleg precízebb vezérlést tenne lehetővé, mint a potenciométerek, és az eszköz méretét is csökkentené.

A szakdolgozatom írása és tervezése során betekintést nyerhettem napjaink egyik legfontosabb eszközének működésébe és felépítésébe. A kidolgozás során olyan teljesítményelektronikai és szoftveres ismeretekre tehettem szert, melyek nagyban támogatni fogják majd a villamosmérnöki tevékenységeimet a későbbiekben.

Mivel a demonstrációs eszköz oktatási célra készült, a felhasználását egy használati útmutató elkészítésével is támogattam. Az útmutató végighalad az eszköz építőelemein és a lehetséges alkalmazási módszerein is, kapcsolási rajzokkal, magyarázatokkal és oszcilloszkópos képekkel illusztrálva. Remélem, hogy ez az eszköz támogatja majd a villamosmérnöki képzés teljesítményelektronikával kapcsolatos laborgyakorlatait, és legalább annyi tudást fog majd átadni a hallgatóknak, mint amennyit én szereztem a tervezésekor.

Irodalomjegyzék

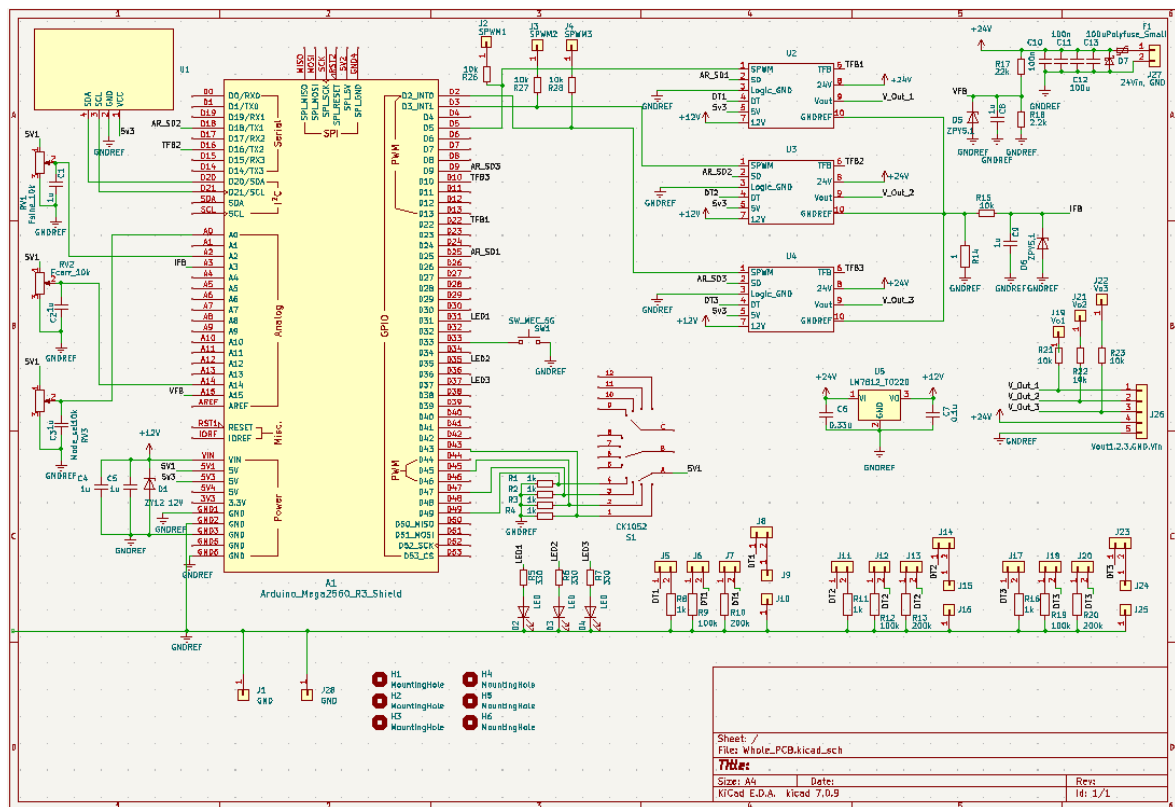
- [1] Daniel W. Hart: *Power Electronics*, McGraw-Hill, 2010
- [2] Muhammad H. Rashid, Ph.D: *Power Electronics Handbook 3rd Edition*, Elsevier Inc., 2011
- [3] Yu, Yan: *Comparison of three-phase inverter modulation techniques: a comprehensive analysis of SPWM and SVPWM*, Highlights in Science, Engineering and Technology. 97, 2024
- [4] Anuja Namboodiri, Harshal S. Wani: *Unipolar and Bipolar PWM Inverter*, IJIRST – International Journal for Innovative Research in Science & Technology, Volume 1, 2014
- [5] Wu, Tianhao & Xie, Xiaorong & Jiang, Qirong & Shen, Zhuoxuan: *Impedance modelling of grid-connected voltage-source converters considering the saturation non-linearity*, IET Generation, Transmission & Distribution, 2020
- [6] Mehmet Emin, Heybet KILIÇ1: *MODULATION INDEX AND SWITCHING FREQUENCY EFFECT ON SYMMETRIC REGULAR SAMPLED SPWM*, Vocational Schools of Technical Sciences, Dicle University, Diyarbakir, Turkey, 2017
- [7] Charles Tsai: *Sine Wave Generation Using PWM With Hercules™ N2HET and HTU* Application Report, Texas Instruments, 2015
https://www.ti.com/lit/an/spna217/spna217.pdf?ts=1732029869694&ref_url=https%253A%252F%252Fwww.google.com%252F
- [8] Aboadla, Ezzidin & Khan, Sheroz & Habaebi, Mohamed & Gunawan, Teddy & Hamida, Belal & Yaacob, Mashkuri: *Effect of Modulation Index of Pulse Width Modulation Inverter on Total Harmonic Distortion for Sinusoidal*, 2016
- [9] Nazmul Islam Raju, Md. Shahinur Islam, Ahmed Ahsan Uddin: *Sinusoidal PWM Signal Generation Technique for Three Phase Voltage Source Inverter with Analog Circuit & Simulation of PWM Inverter for Standalone Load & Micro-grid System* International Journal of Renewable Energy Research, 2013
- [10] Maximilian Nitzsche, Christoph Cheshire, Manuel Fischer, Johannes Ruthardt Jorg Roth-Stielow: *Comprehensive Comparison of a SiC MOSFET and Si IGBT Based Inverter*
<https://d-nb.info/1212856015/34>

- [11] Josh Smith: *MOSFET vs IGBT for Power Electronics: A Comprehensive Comparison*, 2023
<https://digi-electricpro.com/mosfet-vs-igbt-for-power-electronics-a-comprehensive-comparison/>
- [12] Andras Tantos: *H-Bridge Control*, Modular Circuits
<https://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridge-control/>
- [13] Gonzalez-Longatt, Francisco: *Chapter 5. Three-Phase Transformers*, 2019
- [14] International Rectifier, HV Floating MOS-Gate Driver ICs
https://www.infineon.com/dgdl/Infineon-HV_Floating_MOS_Gate_Drivers-ApplicationNotes-v01_00%20EN.pdf?fileId=5546d4626c1f3dc3016c47de609d140a&da=t
- [15] ATmega2560 adatlap:
<https://www.alldatasheet.com/datasheet-pdf/pdf/107092/ATMEL/ATMEGA2560.html>
- [16] Sankalita Saha, Jose R. Celaya, Bhaskar Saha, Phil Wysocki, and Kai F. Goebel: *Towards Modeling the Effects of Lightning Injection on Power MOSFETs*, Annual Conference of the Prognostics and Health Management Society, 2010
- [17] Attia, H., Che, H. S., Suan, F. T., & Elkhateb, A.: *Bipolar and Unipolar Schemes for Confined Band Variable Switching Frequency PWM Based Inverter*, International Journal of Electrical and Computer, 2020
- [18] Shenai, Krishna & Cavallaro, C. & Musumeci, Salvatore & Pagano, Rosario: *Modeling Low-Voltage Power MOSFETs as Synchronous Rectifiers in Buck Converter Applications*, 2003
- [19] Laszlo Balogh: *Fundamentals of MOSFET and IGBT Gate Driver Circuits*, Application Report, TI.com, 2018
https://www.ti.com/lit/ml/slua618a/slua618a.pdf?ts=1732002854842&ref_url=https%253A%252F%252Fwww.google.com%252F
- [20] IR21844 adatlap, International Rectifier:
[https://www.infineon.com/dgdl/Infineon-IR2184\(4\)\(S\)-DataSheet-v01_00-EN.pdf?fileId=5546d462533600a4015355c955e616d4](https://www.infineon.com/dgdl/Infineon-IR2184(4)(S)-DataSheet-v01_00-EN.pdf?fileId=5546d462533600a4015355c955e616d4)
- [21] Nyákáruház Kft., A vezető sávok méretezése:
<https://tudasbazis.nyakaruhaz.hu/a-vezetosavok-meretezese/>
- [22] Haque, Moinul & Mohammad, Mostak & Pries, Jason & Choi, Seungdeog: *Comparison of 22 kHz and 85 kHz 50 kW Wireless Charging System Using Si and SiC Switches for Electric Vehicle*, 2018

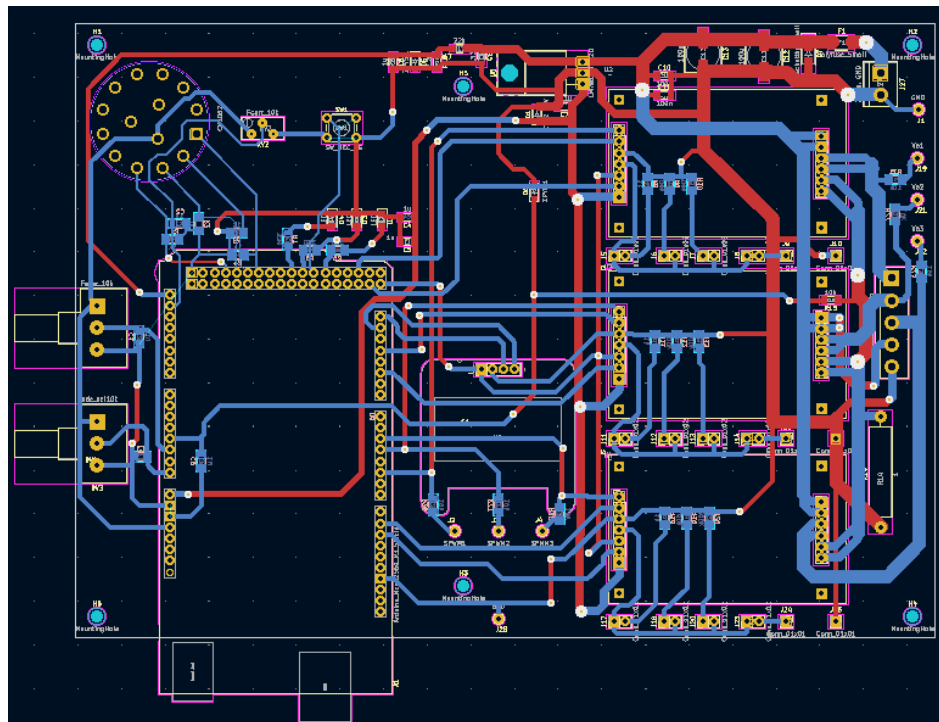
- [23] Siva Uppuluri: *Is today's accepted MOSFET Figure of Merit still relevant*, Nexperia, 2019
<https://efficiencywins.nexperia.com/efficient-products/is-todays-accepted-MOSFET-figure-of-merit-still-relevantn>
- [24] NTD5865NL Teljesítmény MOSFET adatlap:
https://www.hestore.hu/prod_getfile.php?id=9706
- [25] NCE6020AK Teljesítmény MOSFET adatlap:
https://www.hestore.hu/prod_getfile.php?id=11560

Mellékletek

1. A vezérlőegység kapcsolási rajza



2. A vezérlőegység PCB terve



3. A DEMO inverter használati útmutatója (következő oldal)