

FFU ACPS
-
FFU AUTOMATIC CAR PARKING SYSTEM

Szerzők:
Rana Dávid
Tóth Silvester
Németh Balázs

Tartalom

1	Bevezető.....	3
1.1	Cél.....	3
1.2	Eredmény.....	4
2	Ismerkedés a FFU ACPS-el	6
2.1	A kezelőfelület.....	7
2.1.1	Menürendszer.....	8
2.1.2	Eszköztár.....	11
2.1.3	Beállítások Panel.....	12
2.1.4	Szimulációs ablak.....	13
3	A szoftver	14
3.1	Az OpenGL grafika.....	14
3.2	A vezérlőprogram és az OpenGL kapcsolata	15
4	A periféria	16
5	Felhasznált irodalom.....	18

1 Bevezető

Mindannyian voltunk gyerekek. A most felnőtt nők, lánykorukban babáztak, míg a fiúknál, általában az autó volt a menő játék. A gyermekeket a fantáziájuk, messzire röpítette őket a valóságtól, az egyszerű hajasbaba életre kelt, a játékautó pedig varázslatos dolgokra volt képes. Annak idején, mikor mi is ilyen álmokkal szőtt tele kis gyermeki világom, nemegyszer fordult elő az, hogy a kisautó, mely játékom főszereplője volt, megelevenedett, és söfőr nélküli utazásba kezdett, bebarangolva a nagyvilágot, melyet akkor a gyerekszobám piros szőnyege jelentett. Arról álmodtam, hogy egyszer, nekem is lesz egy ilyen szuper autóm, melyben csak hátradőlök s a négykerekű, varázslatos, mesebeli tájakon furikáz majd engem (ahol esetleg meglelem a csodaszép hercegnőt). Azt mondják, az álmok ritkán válnak valóra. De az én gyermekkori álmom úgy érzem, valamilyen szinten teljesülni látszik. Ha nem is mesebeli tájak csodás lényeivel és varázslatos hercegnőkkel, de egy önmagát vezérlő automatizált autóval, igen. Legszébb az egészben, hogy két jóbarátom gyermekkorukban ugyanilyen álmokkal szötte tele kis világát, s ennek köszönhető, hogy együtt megépíthettük a FFU ACPS-t. Ígyhát dokumentációnk következő fejezeteiben, egy álom megvalósulását, és annak terveit vázoljuk majd fel.

1.1 Cél

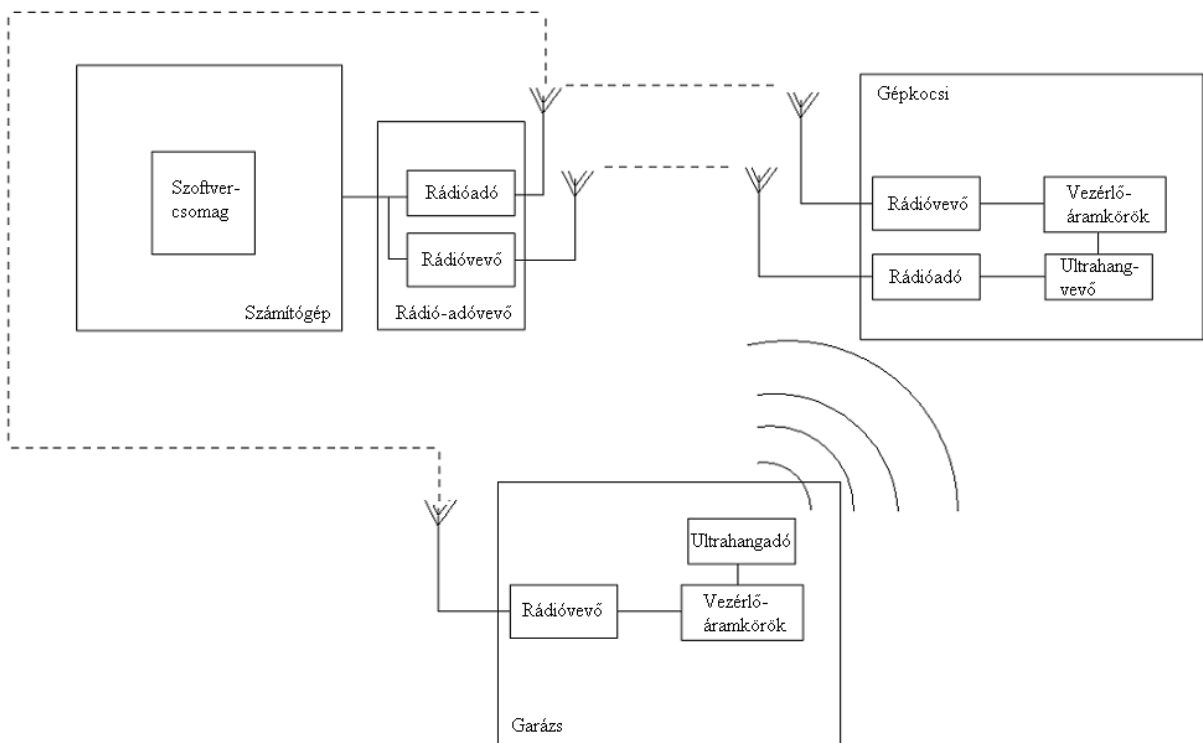
Ahhoz, hogy vázolni tudjuk a terveket, s hogy egyáltalán az automata parkoló rendszer működési elvét bemutathassuk, a bevezetésben használt meseszerű, álombéli kifejezéseket sajnos hanyagolnom kell. Az álom ugyanis valóság lett, amit a hitelesség kedvéért csakis reális, fizikai bizonyítékokkal lehet alátámasztani. A munkát hárman építettük, így mindannyiunknak megvolt a maga feladata. Ahhoz, hogy ennek működését megértsük, vázolnunk szükségeltetik a FFU ACPS teljes struktúráját.

A FFU ACPS egy olyan terv megvalósulásának alapja, ami feltételezi, hogy a jövőben egyre több minden lesz automatizálva. Így például, a garázs előtt hagyott autó felismeri a garázstól való helyzetét, és képes lesz söfőr nélkül beálni. S hogy ez mi módon vihető végbe? A válasz egyszerű, vagyis a modern számítástechnika és elektronika segítségével. Úgy gondoljuk, ezen két tudományág vegyítésével, az emberiség fantasztikus dolgokra lehet képes. A FFU ACPS alapjai is ezen az elven nyugszanak, s az automatizáció segítségével meg is valósulhatott. Az automatizáció, a műszaki tudományok egy olyan ágazata, amely

számítógépes programok és elektronikus áramkörök segítségével önálló munkavégzésre alkalmas robotokat alkot.

1.2 Eredmény

A mi munkánk is automatizáció, ugyanis egy számítógép által futtatott program elektronikus impulzusai hatására, az autóban levő áramkörök a megfelelő módon vezérlik az autót meghajtó motrokat.



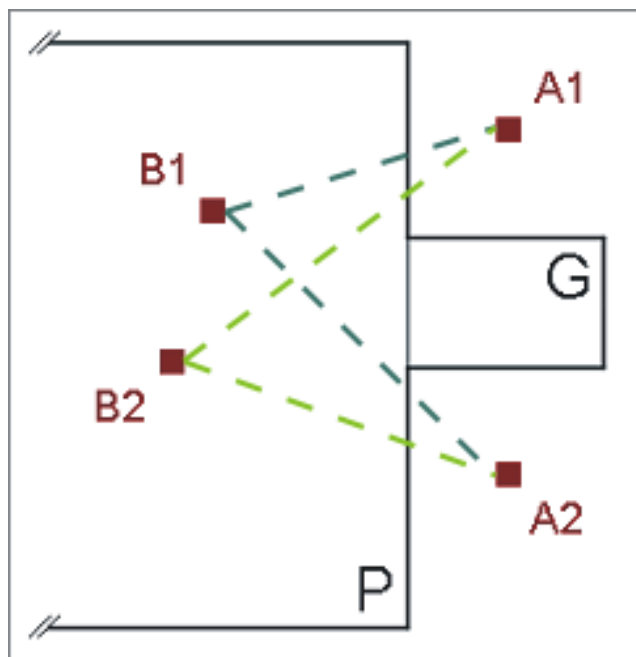
1.ábra: automata parkoló rendszer blokkvázlata

Az 1.ábrán látható az automata parkoló rendszer blokkvázlata, ami dióhéjban vázolja a működési elvet. 4 fontos egységre osztható fel. Első a szoftvercsomag, mely az egész rendszert felügyeli, útvonalat számol, teszteli az útvonalat, szimulálja a beállást, végül LPT porton keresztül, használható jelsorozatot küld a kettős modulnak. Jelsorozat alatt olyan impulzusokat értünk, amik valójában 2,8V alatt és fölött mozognak. Azt a feszültséget, mely 2,8V alatti, 0-nak hívják, 2,8V felett pedig 1-nek. Az 1-es általában, és a mi esetünkben is 3,5V. Ezeket a feszültségtől függő impulzusokat biteknek nevezzük. Minden számítógép szoftver, ezen egyesekkel, és nullákkal kommunikál a külvilággal és a rá csatlakoztatott perifériákkal. Mi esetünkben a periféria a rádió adóvevő, a garázs illetve az autó. Például a

nyomtatót irányító szoftver is ilyen impulzusokat generál a printer vezérléséhez, és ilyen bitsorozatokot küld a billentyűzet is a számítógépnek, mikor lenyomunk egy billentyűt. (Minden billentyűnek más bitekből álló kódja van, amit a számítógép felismer, s ezáltal tud különbséget tenni az között, hogy melyik betűt írassa ki a képernyőre. Ami szintén periféria.) Mivel a szoftver egy nem kézzelfogható dolog ezért szüksége van olyan eszközre ami a valóság számára érzékelhetővé teszi őt. Ez az eszköz a már előbb emlegetett bitsorozat. A bitekre áramkörök reagálnak, előre definiált formában. Azt már a laikusok is tudják, hogy az elektromos áram átalakítható fényvé, hővé, kinetikus energiává. Ez az alapja, az automatizációnak, és így vezérelhető bármilyen eszköz, egy megfoghatatlan, test nélküli szoftver segítségével. Lehet az pár grammos mobiltelefon, vagy akár több tonnás daru!

Az 1.ábrán látható következő egység a rádió adóvevő. Ez a modul biztosítja a kábel nélküli kapcsolatot a szoftver és perifériái között. Ez az egység képes arra, hogy a biteket olyan, térben terjedő elektromágneses hullámmá alakítsa, ami fénysebességgel juttatja el az impulzusokat a célállomáshoz. Ezt az elektromágneses hullámot, rádióhullámnak nevezzük. Erre nagy szükség van, mivel ha elektromos vezetékkel lenne létesítve a kapcsolat, akkor az autó útját kiszámíthatatlanul befolyásolná a kábel súlya, illetve visszatartó ereje. Nem beszélve arról, hogy fel is csavarodhat. A szoftver által küldött impulzusokat rádióhullám formájában megkapja a rádióvevő állomás, ami az elektromágneses hullámot visszaalakítja elektromos árammá. A rádióvevő állomás, az elektromágneses hullámot, már a feldolgozott formában továbbküldi a vezérlőáramköröknek, amik úgy dolgozzák fel az impulzusokat, hogy az a motrok számára használható legyen. Eddig leírtakban vázoltuk, hogy milyen összefüggések alapján irányítja a szoftver az autót. De az, hogy mi módon dönti el a program, hogy az autónak melyik a legoptimálisabb útvonal a garázsig, még kifejtésre vár. Ennek a problémának a kulcsa a helyzetmeghatározó rendszer. Az ő segítségével képes meghatározni a szoftver, hogy a garázshoz képest hol helyezkedik el az autó.

Az egész helyzetmeghatározást az ultrahang adóvevők biztosítják. A hang terjedési sebességéből kiszámítható az ultrahang adó illetve vevő közötti távolság. Az adó a garázs bizonyos pontján van, ugyanúgy a vevő is az autó egy meghatározott pontján. Az 2. ábrán látható, mely távolságok lemérésére van szükség ahhoz, hogy a szoftver megállapíthassa az autó helyzetét, a garázshoz képest. Az A1 és A2 pontokon található az ultrahangadó a B1 és B2 pontokon, pedig az ultrahangvevő.



2.ábra: Helyzetmeghatározási pontok

A kép becsaphat bennünket, ugyanis annak ellenére, hogy két-két pontot látunk, csak egy adó, és csak egy vevő modult használunk. Az adót, egy forgókar segítségével léptetőmotor helyezi át A1 pontból A2-be, illetve a vevőt B1 pontból B2-be.

2 Ismerkedés a FFU ACPS-el

A következőkben a FFU ACPS kezelőfelületét fogjuk ismertetni. Mielőtt a kezelőfelületet ismertetnénk, szeretnénk a program nevével pár gondolatot szólni. Mielőtt a program lényegi funkcióját elemeznénk a program mivoltát kell megértenünk. Ennek legegyszerűbb módja, ha a nevét vesszük alapul. Az „ACPS” egy rövidítés mely a következő szavakból tevődik össze:

A – Automatic

C – Car

P – Parking

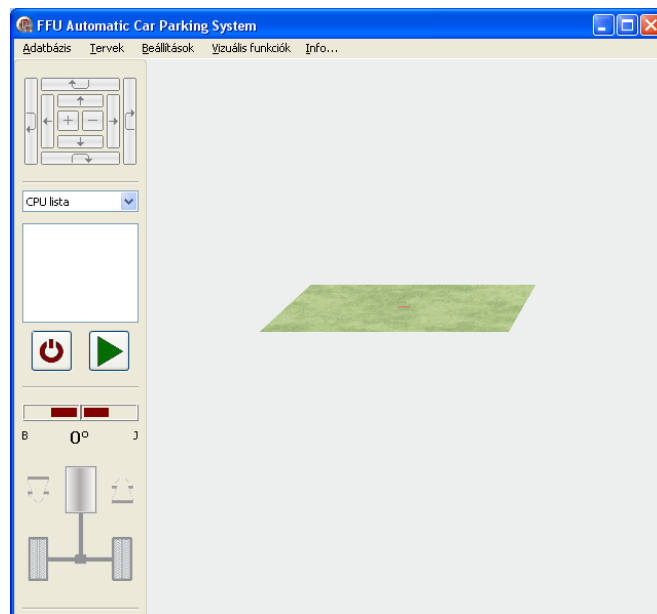
S – System

Lényegében egy olyan szoftverről van szó, mely az egyéb külső hardware-i elemeinek segítségével képes bármilyen típusú gépjárművet leparkoltatni egy garázsba vagy más matematikailag pontosan körbehatarolható helyre. A világon több nagy, neves autógyártó cég is kifejlesztett már hasonló technológiákat, de ezek közül egyiket sem lehetne univerzálisan bevált technológiának nevezni. Az autógyártók technológia mindig csak saját terméküket tudták vezérelni esetlegesen saját termékcsaládjuk összes tagját. Ezzel ellentétben viszont a FFU ACPS rendszere egy univerzálisan minden gépjárműre alkalmazható technológiát kínál.

Ahhoz, hogy mindent átfogó képet kapjunk a rendszer működéséről meg kell ismerkednünk a szoftver kezelőfelületével, szoftver megoldásaival éppúgy, mint a hardware-i megoldásokkal. A következő fejezetekben ezeket fogjuk mélyebben nagyító alá venni.

2.1 A kezelőfelület

A FFU ACPS Borland Delphi 2005 Personal Edition fejlesztői környezetben készült így Microsoft operációs rendszerekből jól ismert grafikai elemeket tartalmazza. A felhasználót első indítás után a következő kép fogadja:



3. ábra: FFU ACPS első indítás után

A kezelőfelületet négy külön álló részre lehet elkülöníteni

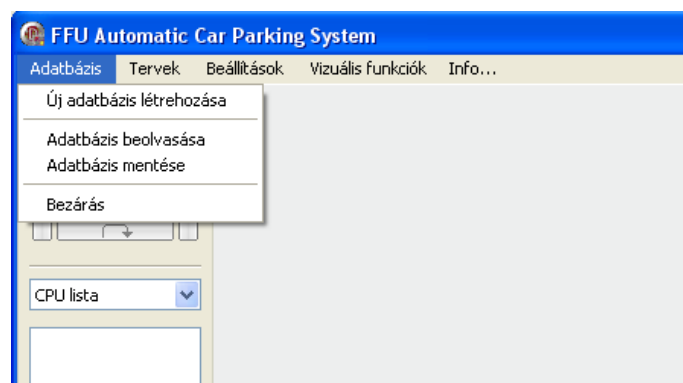
- Menürendszer
- Eszköztár
- Beállítások panel
- Szimulációs ablak

2.1.1 Menürendszer

A Menürendszer 5 részre osztható fel, amelyekkel külön-külön is foglalkozunk:

- Adatbázis
- Tervek
- Beállítások
- Vizuális funkciók
- Info

2.1.1.1 Adatbázis menüpont

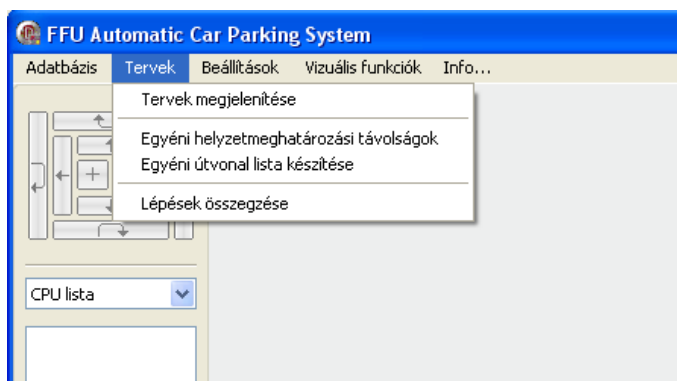


4. ábra: Adatbázis menüpont

Ez a menüpont tartalmazza a szoftver alap funkcióit, olyat mint az új adatbázis készítése, egy már elmentett adatbázis újrahívása esetleges mentése valamint a szoftverből való kilépés. Az adatbázis „ffucaps“ kiterjesztéssel tudunk elmenteni (<fájlnév>.ffuapcs).

Visszaolvasni az elmentett információkat pedig az ugyanilyen kiterjesztésű fájlból tudja. Új adatbázis létrehozása esetén pedig az értékek nullázódnak és új beállításokat adhatunk meg.

2.1.1.2 Tervek menüpont



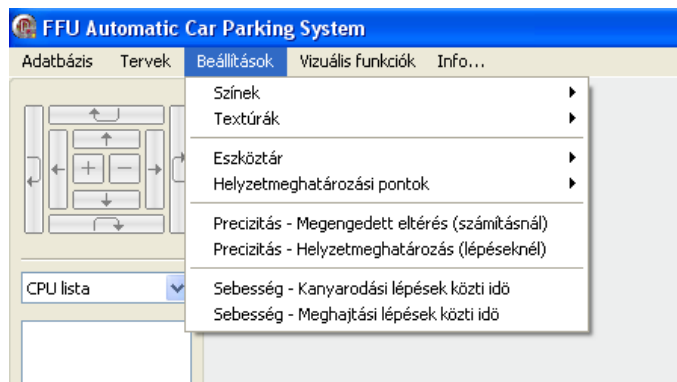
5. ábra: Tervek menüpont

A Tervek megjelenítésre kattintva megjelenik a Beállítások panel, ahol a gépjármű dimenzióit adhatjuk meg. Az „Egyéni helyzet-meghatározási távolságok”-ban megadhatunk saját pontokat, ami alapján a gépjármű helyzete van meghatározva. Ha ezt tesszük, akkor a szoftver nem a tényleges, saját maga által hardware-ileg észlelt adatokkal dolgozik, hanem a mi adatainkkal.

Fontosnak tartottuk azt, hogy a felhasználónak lehetősége nyíljon saját útvonal tervezésére is. Ez azt jelenti, hogy a szoftver nem csak gépjárművek önálló parkolását teszi lehetővé, hanem bármely tetszőleges két pont közti navigációt is.

A Lépések összegzésére kattintva egy új információs ablak jelenik meg, ahol a legfőbb információkat láthatjuk azzal kapcsolatban, hogy milyen adatok alapján számolja a szoftver az egyes manőverezési útvonalat.

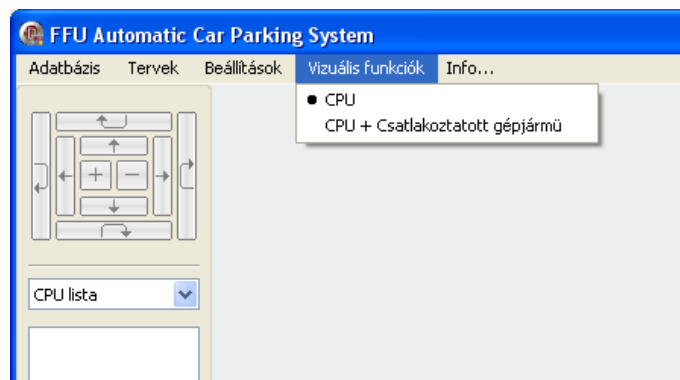
2.1.1.3 Beállítások menüpont



6. ábra: Beállítások menüpont

A Beállítások menüpontban a szoftver saját beállításait találjuk. Olyan beállításokat változtathatunk meg, mint a háttér és a virtuális gépjármű színe, a talaj, a tér, a parkoló és a parkoló-tető textúrájának megváltoztatása, az Eszköztár ki- és bekapcsolása, helyzetmeghatározási pontok megjelenítése illetve letiltása valamint a számolás precizitásának megadása.

2.1.1.4 Vizuális funkciók



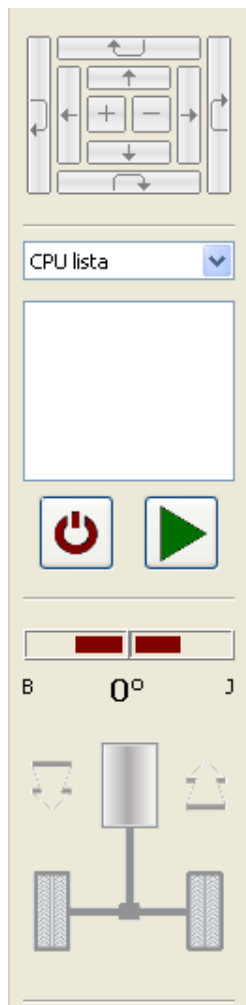
7. ábra: Vizuális funkciók

Ebben a menüpontban azt választhatjuk meg, hogy szimulációt akarunk végezni vagy tényleges fizikai műveletet. Amennyiben a „CPU” beállítást választjuk, abban az esetben szimulációt végzünk. A Program nem kommunikál a külső hardware-eivel, a végtermék pedig csak a számítógép monitorján fog látszani. Viszont ha a „CPU + Csatlakoztatott gépjármű” pontot választjuk a valóságban pontosan az fog megtörténni amit a szoftver grafikusán a monitoron is megjelenít.

2.1.1.5 Info

Ha erre a menüpontra kattintunk, egy új információs ablak ugrik elő, ahol a szoftver készítőinek képeit, neveit, elérhetőségeit, valamint a FFU Team szoftverfejlesztő csoport honlapjának címét és elérhetőségét találjuk.

2.1.2 Eszköztár



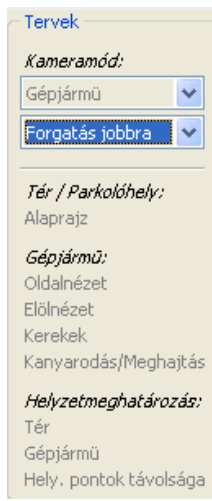
8. ábra: Eszköztár

A Szimulációs ablakban használt kamera pozícióját tudjuk megváltoztatni, így lehetőségünk nyílik arra, hogy más szemszögből is vizsgálhatjuk gépjárművünk parkolási manőverét.

A lenyíló menüben kiválaszthatjuk, hogy a navigációs pontok és precíziós paraméterek segítségével a számítógép által kiszámolt útvonalat akarjuk használni (CPU lista), vagy esetlegesen egy saját útvonalat (Egyéni lista). Amennyiben a saját útvonal mellett döntünk, akkor az „Egyéni útvonal Tervező“ párbeszédablak ugrik elő, ahol ezt könnyedén elvégezhetjük.

Itt nyomon követhetjük, hogy a gépjárművünk kereke mely irányba és hány fokban vannak elfordulva. A gépjármű előre és hátra haladását is nyomon követhetjük ezen az információs panelen.

2.1.3 Beállítások Panel



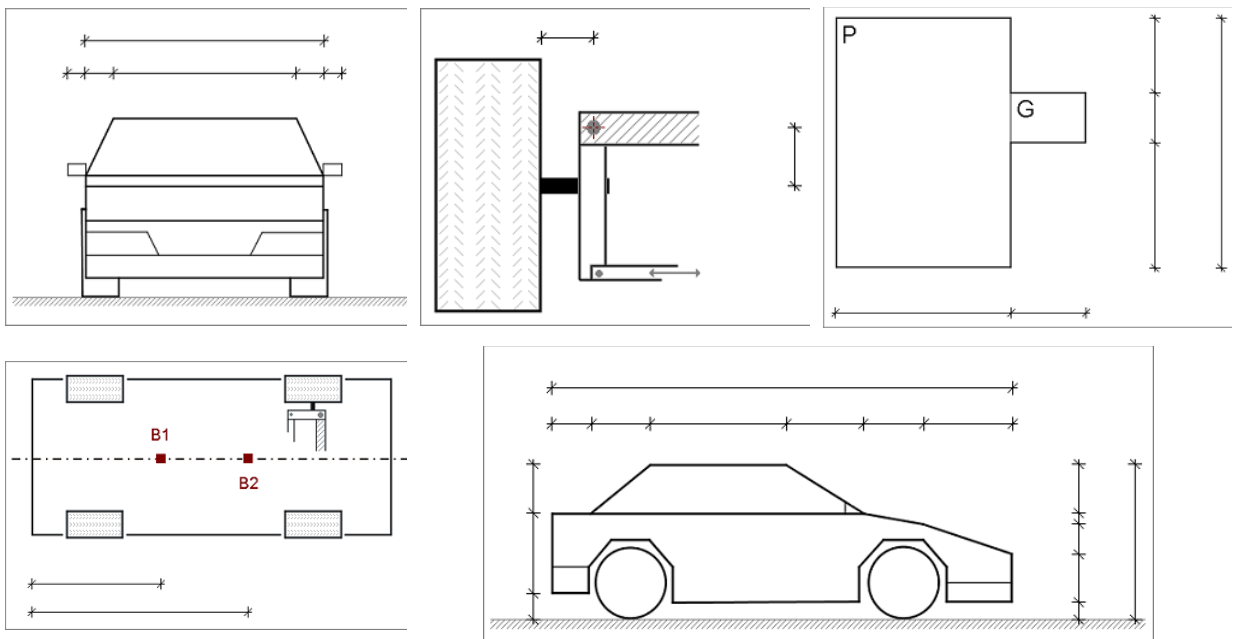
9. ábra: Tervek

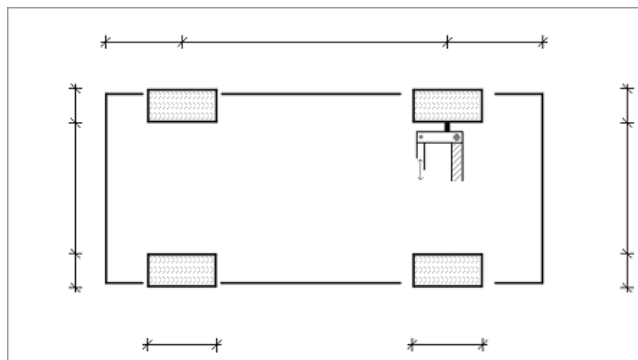
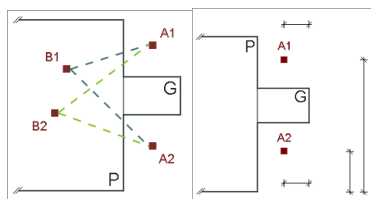
Itt a kameramódot választhatjuk meg. A felső lenyíló menüből, hogy mit lássunk a szimulációs ablakban (Gépjármű, Tér, Mind), a másodikban pedig hogy legyen automatikus forgatás vagy sem valamint azt, hogy a forgatás mely irányba történjen.

A vizuális gépjármű dimenzióinak átszabását teszi lehetővé így segítve azt, hogy a digitálisan elkészített jármű minden tekintetben megfeleljen a valóságban tapasztaltakkal. Sok gondosan elkészített ábra segítségével segíti a szoftver munkánkat és ennek köszönhetően nagyon gyorsan és egyszerűen lehetséges egy új vizuális gépjármű megtervezése.

Akár azt is mondhatnánk, hogy a lehetőségek végtelenek, csak a képzeletünk szabhat határt.

Már említést tettünk a segítő ábrákról. A következő táblázatban lekicsinyítve találhatóak ezek az ábrák.

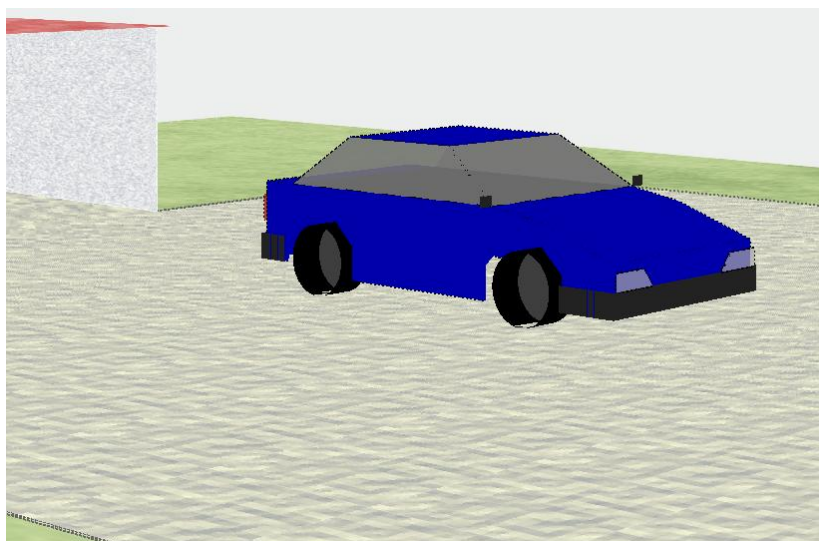




10. ábra: Segédábrák a virtuális gépjármű tervezéséhez

2.1.4 Szimulációs ablak

Szimulációs ablak tartalmazza magát a szimulációt, vagyis azt a grafikai eseménysorozatot, amely azok az adatok alapján készülnek, amelyek aztán a hardware számára feldolgozásra kiküld a program. Magát a szimulációs OpenGL grafika segítségével oldottuk meg.



11. ábra: Egy virtuális gépjármű valamint a tér megjelenítése a Szimulációs ablakban

3 A szoftver

A szoftver "lelke" lényegében az Opengl grafikai rendszer, mely megjeleníti a teljes adatbázisban fellelhető adatokat (erről a következő fejezetekben lesz még szó). A szoftver lényegében szimulálja a valóságban végbement változásokat (gépjármű mozgását). A szoftver célja kiszámolni a legoptimálisabb útvonalat a gépjármű és a garázs között. A szimuláció e útvonal vizuális megjelenítése gépjármű segítségével. A szimuláció valóságban is végbemegy a csatlakoztatott gépjármű bevonásával. A mozgást elősegítő parancssorozat első állomása az LPT port, mellyel a szoftver egy ún. "io.dll" segítségével kommunikál. E dll segítségével a szoftver tökéletes összhangban van a párhuzamos portal, mely egy fontos állomása a parancssorozat a csatlakoztatott gépjárműhöz való eljutásának.

3.1 Az OpenGL grafika

„Az OpenGL hardverfüggetlen programozási felületet biztosító háromdimenziós grafikus alprogramrendszer, melyet a Silicon Graphics Inc. fejlesztett ki. A GL (Graphics Library) több programozási nyelvből elérhető függvénykönyvtárakon keresztül. Az SG (Silicon Graphics) munkaállomások közismert előnye a gyors és igényes grafika, amit hardver oldalról a grafikus kártyába épített egy vagy több geometriai társprocesszor támogat. Ez a koncepció olyan sikeresnek bizonyult, hogy vezető hardver- és szoftvergyártó cégek (többek között a DEC, IBM, Intel, Microsoft és Silicon Graphics) összefogásával létrehoztak egy munkacsoportot, amely ez alapján specifikálta az OpenGL-t.

A legtöbb munkaállomás kategóriájú számítógép támogatja az OpenGL-t, de a PC-s világban használt WIN'95, WIN'98, Windows NT, Windows 2000 és Windows XP operációs rendszerek alatt az IBM PC (és vele kompatibilis) gépeken is futtathatunk ilyen alkalmazásokat. Természetesen a UNIX típusú operációs rendszereknek a PC-ken használatos különböző változatai (pl. Linux) is támogatják az OpenGL alkalmazások futtatását. Az egyes grafikus kártyák eltérő képességűek a beépített geometriai társprocesszorok (grafikai gyorsítók) típusától és számától függően.

Az OpenGL nem geometriai modellezőrendszer, tehát nincsenek benne összetett geometriai objektumok megadására alkalmas parancsok. Bár – mint mondtuk – az OpenGL nem interaktív rajzoló- vagy grafikus rendszer, azonban rendkívül alkalmas a geometriai modellezőrendszerek, CAD rendszerek, grafikus rendszerek, de még az interaktív játékprogramok grafikai igényeinek kielégítésére is. Ennek köszönhetően széles körben

elterjedt. Az OpenGL-t használó alkalmazások több géptípuson, így minden munkaállomás kategóriájú gépen és a nálunk legnépszerűbb PC-ken is futtathatók, még a programok forrásnyelvi szintű portabilitása is megoldható.“ [6]

„Az OpenGL segítségével egyszerű, térben elhelyezkedő 3D-s grafikus primitíveket jeleníthetünk meg a képernyőn. Ennek során megválaszthatjuk a térből síkba való leképezés módját, az objektumok színét, megvilágítását és mintázatát. A síkbeli képen többféle módon figyelembe vehetjük azt is, hogy az elemek a térbeli elhelyezkedésből következően takarják egymást.“ [7]

3.2 A vezérlőprogram és az OpenGL kapcsolata

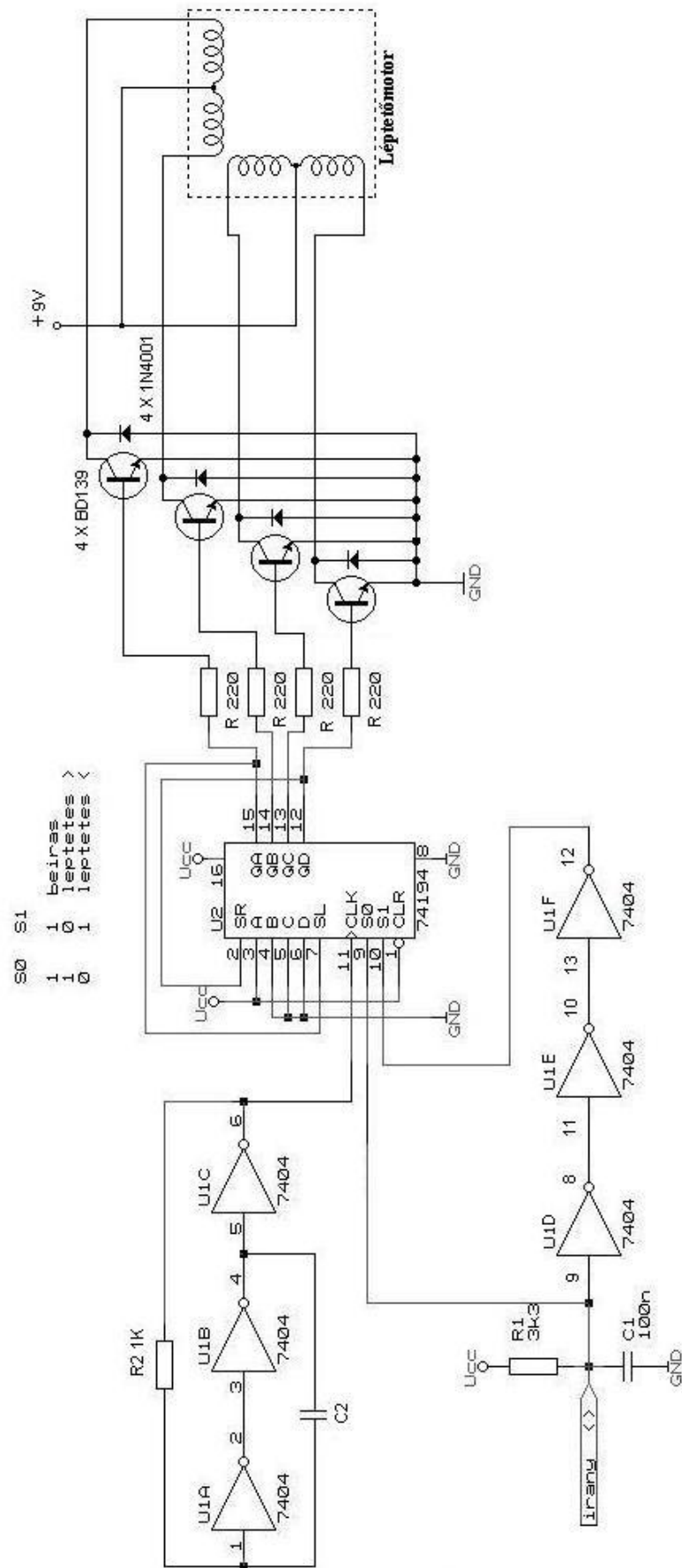
Mivel az OpenGL geometriai primitívekkel dolgozik, ezért a szoftware a legegyszerűbb alakzattal, a poligon segítségével rajzol a leggyakrabban. Természetesen más geometriai elemek is előfordulnak a programban, mint pl. pont, egyenes, szakasz, kör és henger. Mivel az OpenGL koordinátarendszerrel dolgozik, ezért ismernünk kell az objektumok pontos elhelyezkedését a térben. Objektumok alatt olyan alakzatokat értünk, melyekből az egész kirajzolható kép épül fel. Az objektumok pontos elhelyezkedését a kótázás (beállítások panel) után kapjuk, melyeket a program különböző matematikai függvények, tételek segítségével számol ki, mint pl. Pitagorasz-tétel, koszinusz-, szinusztétel, szögfüggvények, vektor hossza, vektorok által közbezárt szög, kör kerülete stb. Az origó ([0,0,0] koordinátájú pont) a tér közepén helyezkedik el, ezért kameraváltásnál e ponton áthaladó tengelyek (x, y, z) körül forgatja illetve csúsztatja a kirajzolható objektumokat.

Az OpenGL szerepe a vezérlőprogramban lényegében az objektumokhoz rendelt kiszámolt adatok megjelenítése vizuális eszközökkel. A gépjármű és környezete objektumokból áll össze, melyek minden egyes koordinátája kiszámolt adat. Ezeket a kiszámolt adatokat a rendszer saját adatbázisba menti (rekordba), melyeket bármikor megtekinthetjük illetve módosíthatjuk.

4 A periféria

Ezen fejezet célja azon áramkörök ismertetése, melyek a számítógép által küldött impulzusokat feldolgozzák, és vezérlik a motrokat. Az első ilyen elektronikai berendezés, a számítógép LPT portjára közvetlenül kapcsolt, rádió adóvevő. Ez egy olyan áramkör, ami vezeték nélküli kapcsolatot biztosít a számítógép és a vezérlendő motrok között. A rádióhullámok frekvenciája 9 KHz-től 3 GHz-ig terjed. Ahhoz, hogy elmagyarázhassam, ennek jelentését, egy csepp fizikával kell megfűszereznem beszámolómat. Az elektromos áram, szabad elektronok mozgását jelenti az elektromos vezetőkben. Ezen elektronok mozgása, elektromágneses hullámokat gerjeszt, amelyek a szabad térben terjednek tovább. Az elektromágneses hullám másodpercenkénti rezgésszáma a frekvencia. Ha megfelelő méretű antennát csatlakoztatunk az áramkörhöz, ezeket a hullámokat úgy kilehet küldeni a nagyvilágba, hogy egy megfelelő vevővel fogni lehessen őket. Minden rádiózással kapcsolatos technika, ezen az elven működik, így a miénk is. Ezeket az elektromágneses hullámokat a rádióvevő, a vezérlőegység számára használható elektromos árammá alakítja át.

A vezérlőegység lelke, a 12. ábrán láthat 74 194-es IC. Ez az integrált áramkör egy másik IC által generált órajel segítségével, ami nevezetesen IC7404-es, hajtja meg a léptetőmotrokat. Az órajel frekvenciája határozza meg azt, hogy a motrok milyen sebességgel végezzék el a lépéseket. Ha a távirányítóval leállítjuk az órajelet, a motor is megáll. Ebből logikusan következik, hogy beindításával, pedig ismét forog. Az órajel frekvenciáját a C2 kondenzátor értéke határozza meg. Az áramkör lelkének 12, 13, 14, 15-ös lábai, egymást követően küldenek feszültség impulzusokat a BD139-es tranzisztorokra, melyek elegendő feszültséget kapcsolnak a léptetőmotor adott tekercsére. A tekercsen mágneses erőtér keletkezik, amely magához vonza a léptetőmotor rotorján levő mágneket. Így fordul egyet a motor tegelye. Ha a tekercseket a tranzisztorok egymás után kapcsolják össze a szükséges feszültséggel, a motor rotorja, forgó mozgást végez. Ez a forgó mozgás, fogaskerekek segítségével az autó kerekeire vihető és ezen a módon teljesül munkánk feltétele, vagyis az, hogy létrehozzuk álmaink önvezérelt autóját. A 74 194-es IC 9-es lábára juttatott impulzus határozza meg a motor forgásának irányát aszerint, hogy az impulzus értéke 1-es vagy 0. Ezen elv alapján működik a perifériában használt összes motor. Legyen az meghajtásért felelős, vagy váltsa az ultrahangos jeladók pozícióját.



12. ábra: Vezérlőáramkör

5 Felhasznált irodalom

- [1] BAKO, M.: Automatizácia knižníc. 2.vyd. 1980. ISBN 80-8046-020-5.
- [2] Telesná výchova a šport : Terminologický výkladový slovník. 2.rev.vyd. Bratislava: SPN, 1980. 35 s. ISBN 80-8046-020-5.
- [3] BIELIKOVÁ, M.: Príspevok k tvorbe konfigurácie softvérového systému s využitím znalostí. Bratislava: FEI STU, 1995. 136 s. Kandidátska dizertačná práca.
- [4] Boutell, Thomas. 1996. What can I do to protect my legal rights on the web?
<http://nswt.tuwien.ac.at:8000/htdocs/boutell/>
- [5] Lisý, L.: Správa o hospodárení podniku. 1998. Osobná komunikácia. 1998-10-22.
E-mail: katka@elf.stuba.sk
- [6] IIT, BME Irányítástechnikai és Informatika Tanszék, Benedek Balázs: *OpenGL*, forrás:
<http://www.iit.bme.hu/~benedek/mirrors/OpenGL/opengl/>, 2006.július
- [7] DELPHI FORFUN HOME, forrás: <http://www.delphiforfun.org/>, 2006.november