

Online irányítású robotkar és alkalmazási lehetősége az oktatásban

LZS-435

Tartalomjegyzék

1. OTDK pályamunka céljai, motivációi:.....	4
2. A Hunveyor program.....	5
3. A nemzetközi oktatási (robot) projektekről.....	7
4. A robotkar szerepe a közoktatásban.....	8
5. A virtuális laboratórium.....	10
6. Gyakorlati megvalósítás.....	11
6.1 A robotkar.....	11
6.2 a szervomotorok és vezérlésük.....	12
6.3 A wireless modul.....	13
6.4 A szerver.....	15
6.5 A kliens oldal.....	16
6.6 A webes felület.....	17
7. A megvalósításhoz használt eszközök, felhasznált technológiák.....	18
7.1 A PHP.....	19
7.2 A MySQL.....	20
7.3 A HTML.....	22
7.4 A TCP/IP hálózati protokoll.....	22
7.5 Az UDP hálózati protokoll.....	23
7.6 A C++.....	24
7.7 Az operációs rendszer és a fejlesztőkörnyezet.....	25
8. Távlati tervek.....	26
Irodalomjegyzék.....	28
Mellékletek.....	30
1. sz. melléklet: Mintatanterv középiskolások számára úrkutatói szimulációhoz.....	30
2. sz. melléklet: ábrák.....	31

A projekt fő célkitűzései, egy online irányítású robotkar szoftverhátterének kidolgozása és az oktatásban való részvételének lehetővé tétele. A robotkar elsősorban űrkutatási szimulációk kivitelezésére szolgál középiskolai –és akár felsőoktatási – intézmények számára. De a természettudományok területén is sokrétűen felhasználható. Egyben remek tanulási lehetőség, ösztönöz akár a csapatmunkára és akár az önálló logikus gondolkodásra is. Mindeközben persze megtapasztalhatók a korlátai is, és adódhatnak problémák, amiket meg kell oldani, ezzel is serkenti a diákokat a gondolkodásra. A lényege, hogy egy távoli virtuális laboratórium falai között van elhelyezve a robotkar, ahol szimulált bolygófelszíni körülmények adóttak. A diákok interneten keresztül tudnak csatlakozni a robotkarhoz és átvenni annak vezérlését, tudományos feladatok szimulációinak végrehajtása céljából. Ez azt hivatott imitálni, mintha egy irányító központból vezérelnék egy bolygófelszíni kutatószondát. A robotkar oktatásban való előnyeinek és lehetőségeinek kívül bemutatásra kerül a szoftverháttér.

The two main goals of this project, are the followings: First of all to create a controlling software for an online controllable robot-arm. In the other hand, it can be useful for educational purposes to encourage team-work or logical thinking. The robotarm's main function is to simulate a space researchment for high schools or universities. But it could be useful for other scientific courses. It is a great chance to learn about the robot technics and space researchment. It also stimulates students to think, or to work as a team. During the simulation, students should solve problems and experience the limits of this technology. The robotarm is placed in a virtual laboratory where the conditons of employment are similar to the ones in space. The students can join the robotarm's control software by Internet and they can do scientific work remotely. The aim of this is to imitate the work of the scientists or researchers who guide a spacecraft from a control center. Both the advantages of the useage in education and the controlling softwares are shown in this study.

1. OTDK pályamunka céljai, motivációi:

Az oktatásba egyre több modern kori jelenséget kell bevonni. A jelenségek megismerése azért fontos, mert a munkaerőpiacon való optimális elhelyezkedéshez idejekorán föl kell hívni a tanulók, hallgatók figyelmét a technológiai fejlesztésekre. Ilyen modern kori jelenségcsoport az, amikor gépi munkával helyettesítünk addig ember végezte tevékenységeket. Röviden a robotika címszó alatt foglalhatjuk össze mindezt.

A pályamunka célja az együttműködésre épülő tanulás megvalósítási lehetőségének megteremtése, aktuális témakörben: a távoli környezet, mint pedagógiai vonzerő kiaknázása, interaktív, modern technológiákat felhasználó oktatási anyagok létrehozása. Szükséges egy szintetizáló tananyag kifejlesztése élethelyzetekre, amely serkenti a kreatív megoldás keresést valós problémákra (akár egyénileg vagy csoportban), ezzel is segítve a diákok ezekhez szükséges képességeinek kialakulását. A digitális kompetenciafejlesztés - nemcsak - informatika órákon fontos, mert ez a technológia és az eszköz sokrétűsége lehetőséget teremt egyéb természettudományi tárgyakon való bemutatásra és használatra is. Fontos továbbá, hogy ez lehetővé teszi a felkészítést a technológiai korszakváltásra, mert napjainkban egyre nélkülözhetetlenebb az internet, mint kapcsolattartási és oktatási felület. A diákok szívesebben használják fel az internetet valamint a számítógépet tudásuk gyarapítására, mint a tankönyveket. Az USA-ban készült egy tanulmány, mely azt bizonyítja, hogy az ottani felsőoktatásban résztvevő diákok egyre szélesebb körben és pozitív véleménnyel használják az internetet tanuláshoz. (*The Internet and Higher Education Volume 11, Issues 3–4, 2008, Pages 165–177*) Ezt a lehetőséget nagyszerűen ki lehet használni hazánkban is, és ezt valósítom meg az online irányítású robotkar projektben. Célja továbbá, hogy interaktív, online vezérléssel a diákok képzeletgazdaggá és részesévé váljanak egy jövőbeli érdekesítő oktatási technológiának. Ezáltal a diákok informatikai, technológiai, matematikai, környezettudományi, fizikai, kémiai, és planetáris geológiai ismeretei is gyarapodjanak.

2. A Hunveyor program

A Hunveyor gyakorló űrszonda modell megépítése 1997-ben merült föl az ELTE Technika Tanszékén. Tanárok és hallgatók együtt építették meg az első modul szerkezetű Hunveyort, mint egy kísérleti oktatási eszközt. A Technika oktatásában sokoldalúan föl lehetett használni a Hunveyor modellt a különféle technológiai, műszaki, informatikai és természettudományos témakörök oktatására.

Az űrszonda modell a gyakorlatban valósítja meg az építést és az oktatást egyszerre. Ez is volt az egyik célja: a diákok idejekorán ismerkedjenek meg összetett elektronikus szerkezettel, ami később sokoldalúan fejleszthető tovább. A Hunveyor azt akarta bemutatni, mit csinál egy egyszerű robot, amikor megérkezik egy idegen égitest felszínére. MÉR. Mit mér? A környezetében lévő állapotokat méri, méri föl. Ehhez mérőműszerek ráépítésére volt szükség. Ezeket a méréseket vezényelte az űrszonda modell fedélzeti számítógépe. Ezzel az érdekes és komplex rendszerrel, amely tehát a diákok által összeépíthető részrendszerekből állt, egy működő mérőrendszer egységet lehetett tanulmányozni. Ilyen jellegű mérések bemutatását célozta meg a robotkar építése is.

A Hunveyor használatánál fontos volt az, hogy a megépített technológiai megoldások oktatási és pedagógiai szempontból is jól áttekinthetők legyenek. Ha sikerült példaértékű megoldásokat találni, akkor az segítette a diákok szemléletének a fejlesztését. Például számos mérő technológia lekicsinyítve és más technológiák társaságában fordul elő a Hunveyoron. Az űrszonda modellt építő diák így össze tudta hasonlítani az űrszondára rákicsinyított űrtechnológiai megoldásokat a hétköznapiak nagyméretű megoldásaival. Nagy méretben ugyanis már ismert technológiák (például manipuláló eszközök, kar, talajfúró, stb.) szerepelhetnek lekicsinyítve a Hunveyoron.

A Hunveyorra építve a hétköznapiakban unalmasnak tartott technológiákat, például az anyagtudományi méréseket, egyes gyártási és mérési technológiákat is szívesebben tanulták a diákok. A Hunveyor használatával az anyagtudományi mérési technológiák de a környezettudományi mérések is jól bemutatathatók. A Hunveyoron nagyszámú, párhuzamosan működő folyamat dolgozik egyszerre. Ezek az együttesen ható eseményszálak a folyamatoknak két különálló csoportját alkotják. A Hunveyorra hatnak a külső környezet folyamatai: áramlások, porlerakódások, fölmelegedés a fény hatására, vagy elektromos

feltöltődés. A Hunveyor belső, mérési technológiai viszont a Hunveyor technológiai folyamatait képviselik. A kettőt egyszerre lehet tanulmányozni: a gyakorló űrszonda műszerparkjának folyamatait és a Hunveyor körül zajló természeti folyamatokat együttesen.

Az együttes vizsgálatban ez a kétféle folyamatcsoport összehasonlítható. Ennek során a természettudományi tantantárgyak (kémia, fizika, biológia, földtudomány) az informatikai témák, részére is gyűlik ismeret. Ezek az ismereti területek nem elválasztva, külön-külön jelennek meg, hanem egységes egészként.

Robotkarunk szempontjából összegezve tehát megtaláljuk a Hunveyoron a komplexitást, a modellezést, a technológiákat, de a környezeti jellemzőket is (pl. az áramlásokat). Ugyanakkor anyagvizsgálati ismeretekre is szert tehetünk.

Míg egyes tantárgyakban a preparátumkészítés az első lépés ahhoz, hogy műszerek segítségével az anyag mélyebb szerkezeti szintjei felé eljussunk. (például a kőzetek szövetének vizsgálatánál), addig Hunveyor gyakorló űrszonda segítségével inkább a mérőműszer együttest fejlesztjük ki. Ez a munkát élményszerűvé teszi, mert a saját készítésű eszközzel lépnek kapcsolatba az anyagokkal a diákok.

Oktatási előny a Hunveyoránál az is, ha az elkészítés folyamatát elemezzük. Mi is ezt tesszük majd a robotkarral. A modellen folytatott építési munka is összetett technológiai folyamat, ezért összefoglalható gyártási folyamatábrán. A gyártási folyamatára a műveletsorok térképe, melyen az idő függvényében láthatjuk a munka fázisait. A munka elemzésének a végeredménye az, hogy a diákok átlátják és megjegyzik az egyes munkafolyamatokat. Fölismerik a részfolyamatok egymáshoz való viszonyát.

A Hunveyor építése során végzett munkák során azt tapasztalhatjuk, hogy a tudás, amit az eszköz megépítése közvetített, összetett formában is megjelenik. Az addig távol eső ismeretek között az elektronikus szerkezet építésével végzett munka teremt kapcsolatot. Ez pedig az új évezred nagy kihívása. Nem szigetként áll egy ismereti tartomány, hanem fokozatosan összképbe rendeződik.

Az ábrákon(1-3. ábra) néhány képpel illusztráljuk a Hunveyor és Husar kísérleti űrszonda modellek építésének eddigi eredményeit.

(Bérczi Hegyi Kovács Fabriczy Földi Keresztesi Cech Drommer Gránicz Hevesi Borbola Tóth Németh Horváth Diósy Kovács Bordás Köllő Roskó (szerkesztette: Bérczi Szaniszló), 2001)(Bérczi Hegyi Hudoba Balogh Bíró Bornemisza Csapó Drommer Geresdi Halász Hargitai Imrek, Keresztesi Kókány Nagy Pápai Samu Sasvári, Senyei Siroki Varga

(szerk. Bérczi Szaniszló), 2006)

A dolgozat tárgyát képező robotkar, a Hunveyor projekt szerves részeként jött létre melynek létjogosultságáról a Fizikai Szemle a 2008/2 számában az alábbi módon ír:

„A Hunveyor – Husar - modellek pedagógiai értéke

A gyakorló űrszondamodellek építése a modern oktatási formák felé mutat: tehát pedagógiai értékű is ez az építési program. Tantárgyintegráló szerepe is elvitathatatlan. Az a hallgató, aki egy működő űrszondát a maga sokszínű fedélzeti technológiáival, elektronikaival, informatikai feladataival végiggyakorolt, meg fogja állni a helyét a polgári életben is, ahol a technológiák ismerete és a szervező-építő tudás is nélkülözhetetlen.

A Hunveyor -rendszer a jövőbeli érdekesítő oktatás egyik ígérete. A Hunveyor gyakorló űrszonda vázának, elektronikájának, s a kezdetben felszerelésre kerülő egyszerűbb mérőrendszereinek megépítésével a hallgatók informatikai, környezettudományi, fizikai, kémiai és planetáris geológiai ismeretei is gyarapodnak. Fontos, hogy mindvégig egységes egészként kezelik a műszeregyüttest mint technológiai rendszert, amely befogadja, méri és továbbítja a környezet folyamatairól érkező adatokat. Az űrszonda mérő és informatikai folyamatait kapcsoltan, szintézisben kell, hogy lássák a diákok a környezetben zajló áramlásokkal (szél, víz, talaj, hő stb.), melyekbe a mérőműszerek csápjaikat belemerítik. A Hunveyor kísérleti gyakorló űrszonda építési munkái így rendszerszemléletet is kialakítanak bennük.”

3. A nemzetközi oktatási (robot) projektekről

Külföldön már több évtizedes múltra nyúlnak vissza az olyan jellegű projektek, versenyek, melyek a résztvevők kreativitásait célozzák meg. A csapatmunkára építő projektek igen fontos szerepet játszanak az egyén képességeinek fejlődéseiben. Megtanulják, hogyan kell együtt, egy csapatként dolgozni, továbbá speciális ismereteket szereznek a világ tudományainak legkülönbözőbb területeiről. Ezzel elősegítik a kreatív gondolkodásmódot, és érdekesítővé teszik a tanulást.

Az ilyen és ehhez hasonló versenyek Magyarországon főként a felsőoktatásban résztvevőknek szólnak, holott az alkotószellemű gondolkodásmód elősegítését már általános és

középiskolákban is érdemes elkezdni. Az egyik ilyen példa a külföldön (Európában és Amerikában) megrendezett *First Lego League*:

Korosztály: 9 -16 év. Témakör: A világ valamely valódi tudományos témája. Résztevők: Tíz fős csapatok tanári irányítással. Két feladat megoldása: robotjáték és a projekt munka.

1.) Robotjáték (programozása az autonóm robotnak)

2.) Valós problémára megoldáskeresés (diákok projekt feladata¹)

(<http://www.firstlegoleague.org> 2012.12.15)

Ebből is látszik, hogy van létjogosultsága egy robottechnikával foglalkozó projektnek, akár a hazai közoktatásban is. Elsősorban középiskolák részére lett kifejlesztve informatika óra, technika óra, természetkutató órák, tanítási órán kívül (szakkör) témakörökön való felhasználás céljából. A 10-17 éves korosztály számára, egy mintatanterv (1.sz melléklet) is készült. Mindazonáltal, a felsőoktatásban is megállja a helyét a megfelelő témakörökben, de itt inkább a működési elv és továbbfejlesztési szempontok, lehetőségek dominálnak.

4. A robotkar szerepe a közoktatásban

A fejlesztéseim célja, hogy az interneten keresztül elérhetővé tegyek különböző köz- és akár magánoktatási intézmények számára olyan technológiákat, amelyek önmagukban költségesek lennének, ha minden intézménynek külön-külön meg kellene vásárolnia. Jelenleg egy virtuális laboratórium részeként működtethető, távolról vezérelhető robotkar kifejlesztésével foglalkozom. Ennek a virtuális laboratóriumnak ezen kívül még szerves részét képezik a Husar eRover keresőrobotjai, amiket szintén lehetséges távolról, akár interneten keresztül vezérelni. Ezen technológia segítségével lehetőségük nyílik az adott intézményeknek, hogy részeseivé váljanak a kutatási munkának, átvegyék az irányítást, anélkül, hogy különösen nagyobb beruházásra lenne szükségük. Ugyanis elég hozzá egy internetkapcsolattal rendelkező számítógép. Ily módon betekintést nyerhetünk különböző területekbe, legyen szó akár biológia vagy kémia óráról vagy orvosi környezetről, ahol azt kell demonstrálni, hogy adott kutatók miként dolgoznak veszélyes anyagokkal, vagy orvosok hogyan műtenek robotkarok felhasználásával. Alkalmazás közben meg tapasztalhatjuk, hogy könnyedén

¹ 2011-ben a verseny témája: innovatív élelmiszer-biztonsági megoldások; steril élelmiszer feldolgozás és szállítás, egészséges feldolgozás és tárolás problémája, stb.

használható ez a technológia. Egy fizika vagy technika óra keretében a tanár nemcsak elmeséli a technikát, hanem a diákok kezébe adja a vezérlést. A gyakorlatban is kipróbálhatják és bemutathatják, hogy milyen lehet amikor egy űrállomást kell robotkarral megjavítani, egy Mars- vagy holdjáró felfedező szonda robotkarját vezérelni. Mindehhez az adott oktatási intézménynek nem kell mást tenni, mint csatlakozni a projekthez. Azaz regisztrál magának egy felhasználói fiókot, ezen a felületen van lehetősége időpontfoglalásra a robotkar használatához, továbbá letöltheti a vezérléshez szükséges programokat, leírásokat, segédanyagot és nem utolsósorban az egész elméleti háttérét bemutató oktatóanyagot.

Felhasználási lehetőségei sokrétűek. Vegyünk a példának okáért egy technika órát egy átlagos gimnáziumban, ahol az űrkatatásról vagy robottechnológiáról van szó. Ez a technológia nagyobb érdeklődést válthat ki, ha nemcsak képen látja az adott diák, hogy mit tud egy űrszonda, kutatórobot, hanem meg is tapasztalhatja. Ellenérvként felhozható, hogy ott az internet, meg lehet nézni felvételeket robottechnológiáról, vannak remek modellek, virtuálisan vezérelhető kutatórobot, még azt is meg lehet nézni, hogy éppen pályájának mely szakaszán tart a Nemzetközi Űrállomás. De ez mégiscsak kevésbé személyes, kevésbé hihető, mint amikor az adott felhasználó perszonális tapasztalatait veheti alapul.

A mai diákok már el vannak halmozva az internet és az elektronika minden kényelmével és lehetőségével. Azonban ez még újdonságnak számít ezen a területen is. A kipróbálásnak is lehet több módja, mert léteznek abszolút virtuális, számítógépen lemodellezett, jobb esetben 3 dimenziós mozgatható kutatóeszközök vagy robotkarok. Erről azonban beláthatjuk, hogy könnyű azt feltételezni: számítógépen bármi lehetséges, bárhogyan tekerheti, nyomkodhatja a gombokat, szinte bármire képes az adott 3 dimenziós modell a virtuális térben. Innentől fogva mutatkozik meg az online vezérlésű robotkar egyik legnagyobb előnye.

Az általam továbbfejlesztett technológia lehetővé teszi, hogy lássák, hogyan reagál a robotkar az ő utasításaira, mi a hatása egy-egy parancsnak vagy kezelőszervi utasításnak. Azonnali visszacsatolást kap a fizikai valóságban történő dolgokról. Egy létező, ténylegesen kézzel fogható eszközt használnak és ezt vezérlik online, valós időben és mindeközben egy online videóképpel kapnak visszajelzést tevékenységükről. Ebben az esetben megtapasztalhatják korlátait, kontrollálási nehézségeit, vagy éppen könnyedségét. Az adott helyzetben sokkal valóság-közelibbnek érezhetik az egész rendszert - mivel az is! Itt a felelősségérzet is könnyebben realizálódik és nem utolsósorban a korosztályra jellemző technikafüggőség szempontjából is értékelhető. Beleélhetik magukat, hogy milyen is lehet egy bolygófelszíni

kutatórobot, mélytengeri robot-tengeralattjáró kapitányának vagy a technikát a valóságban is kezelő specialistának, mérnöknek lenni. Milyen nehézségekkel kell megküzdeni ezeknek a specialistáknak és, hogy mekkora előnyt jelent a robottechnológia.

Mindezeneken felül, az is egy további érdekes témakör, hogy miként működik az online vezérlésű robotkar vezérlőrendszere. Egy tanóra keretein belül lehet beszélni akár a működési elvéről, a felhasznált lényeges fizikai alapelvekről, az informatikai vonatkozásokról, de akár a jövőbeli alkalmazási lehetőségekről. Megismerkedhetnek a rendszer működtetéséhez szükséges technikai elemek sorozatával. Végigkövethető, hogy a kar az egyetemi laborban a vezeték nélküli internet segítségével csatlakozik a szervergéphez, az kábelen/műholdon az adott iskolához, így ott a megfelelően csatlakozó gépekre telepített vezérlőprogram segítségével lehet azt vezérelni. Ezt mindenképpen érdemes és ki is kell terjeszteni az országon, földrészen túlra is, akár az űrtechnológiára. Érdekes lesz áttekinteni technikai szempontból, hogy ott milyen közegeken kell keresztüljutnia a jeleknek. A kezelő távolsága a központtól jelkésedelmet okozhat ezért további fejlesztésekre lesz szükség.

5. A virtuális laboratórium

Az alapötlet a fentebb leírtak alapján alakult ki. A lényege, hogy egy lezárt helyen egy külön szobában az adott szimulációnak megfelelő terepet alakítunk ki, például marsi körülményeket, (leszámítva a légkört és a hőmérsékletet), egy terepasztalt tereptárgyakkal. Ide behelyezzük a Husar eRover egységeket. Ezek egy másik hasonló projekt részeként kerülnek ide, hasonló célokkal, különböző speciális szenzorokkal felszerelve.² Mindezek mellé elhelyezzük a robotkart egy fix stabil helyre, mintha egy űrállomás vagy letelepített kutatószonda tartozéka lenne.

Ezután már csak a fantázia és a tanmenet szab határt, hogy éppen mire használjuk. Alternatív felhasználási lehetőség például, hogy tollat rögzítünk a markoló fejegységbe és rajzolunk vele valamit, de ennél sokkal általánosabb és elfogadottabb felhasználási mód a közetminták begyűjtése. Szintén könnyen imitálható tevékenység, csak kavicsokra van szükség és onnantól kezdve lehet válogatni köztük. A robotkar hatósugarán belül mozgathatjuk a terepasztalon - akár speciális felkérésnek megfelelően - előre elkészített tárgyakat, anyagokat. Össze lehet

² Husar-2 eRover: kisméretű, online irányítható kutatórobot.
<http://planetologia.elte.hu/1/cikkek.phtml?cim=hunveyor.html> (2012.12.12)

gyűjteni például az ott levő közeteket, de akár telepíteni lehet például egy dobozból az oda bekészített műszerimitációkat. Akár kicsomagolással és beindítással együtt. De lehet építeni is az előre elkészített építőmodulokból. A virtuális laboratórium nagy előnye, hogy nem igényel emberi beavatkozást miután berendezték, mert minden benne működő eszköz elérhető interneten keresztül. Innentől kezdve, csak egy lépés, hogy jobban kiterjedjen a robotprogram és időzónákat átívelő távolságból is bármikor el lehessen érni, mivel adott a berendezett laboratórium. Az állandó körülmények megléte miatt nem kell leállítani a rendszert éjszakára, lévén, hogy zárt térben vagyunk teljesen mindegy, hogy éjjel van-e vagy nappal.

Az egész terepet több kamera is figyeli, ezek segítik navigálni a kutatóegységeket. A robotkarhoz külön kamera is tartozik, mivel célszerű közelről látni az elvégzendő feladat tárgyát, olyan szemszögből, ahogy nagymértékű – országok vagy bolygók közti - távirányítás esetén a gép kezelője is látná. Csak így lehet pontosan dolgozni, kényesebb feladatokat is elvégezni.

6. Gyakorlati megvalósítás

6.1 A robotkar

A robotkar az egyetem tervei alapján (4.ábra) a Napport Kft. által legyártott és forgalmazott *robotkar egységcsomag* nevű termék (5.ábra), szervo vezérlő elektronikával. (6. ábra) A robotkar, átlátszó plexiből kivágott lapokból van összeszerelve, melynek a részegységeit („ízületeit”) szervomotorok vezérlik. A rendszer egy talapzaton helyezkedik el, melyen 180°-os szögben tud forogni. A részegységek: alsó kar, felső kar, végcsukló, fogó fejegység, szintén szervomotorokkal vannak ellátva, melyek elméletileg 180°-os mozgási teret engednének meg. A gyakorlatban a szerkezet konstrukciója miatt korlátozódik néhány fokkal, egységenként más-más szögtartományokra. A karnak két fő szegmense van: az egyikkel kapcsolódik a forgatható talapzathoz, a másik körülbelül ugyanakkora és ez az előző folytatása, ez kapcsolja össze a végelemmel (7-8.ábra). Ez a tetején lévő elem csuklószerűen tud fel-le mozogni, és ennek a részeként jelenik meg a fogórész. Magát a fogót két plexi karom és egy szervo képezi. A robotkar méretezése miatt csak egy szervomotor felhasználására volt lehetőség a fogófej működtetéséhez. Ez lekorlátozza azon dolgokat, amiket meg tudunk ragadni vele. Méret és forma szerint körülbelül egy 5x3x3 centiméteres maximum paraméterekkel rendelkező körülbelül 100 gramm össztömegű testet képes

megfogni. Mivel előreláthatólag ez a koncepció csak laboratóriumi körülmények közt lesz használatos, megfelelő környezet kialakításával ellensúlyozni tudjuk ezt a hiányosságot. Ezek közé sorolható a megfelelő méretű és formájú közetminta imitációk felhasználása, vagy a rajzos megvalósítás esetén, a fogófej lecserélése egy speciálisan rajzeszköz rögzítésére szolgáló fajtával. Lévé, hogy a kar anyaga körülbelül 2mm vastag plexi, mely egy eléggé kemény és merev műanyag, teherbírása korlátozott, de statikailag úgy van megtervezve, hogy a talppont lerögzítésével az adott körülményekhez képest maximálisan tud teljesíteni. Ez úgy értendő, hogy amit a szervomotorok elbírnak, azt fel is tudja emelni. Mindezek ellenére valóban fontos dolog, hogy a talpzata fixen le legyen rögzítve, mert ha nem így van, akkor hajlamos lehet a felborulásra. Azonban az adott célra és környezetbe történő implementálása során, úgy vélem, kiküszöbölhetők ezen hiányosságai, és oktatási valamint demonstrációs célokra tökéletesen alkalmazható minden további szükséges emberi beavatkozás nélkül. Ezzel pedig maximálisan képes ellátni a fentebb említett feladatát.

6.2 a szervomotorok és vezérlésük

A szervomotorok vezérléséről egy külön elektronikai modul gondoskodik. Ennek feladata, hogy kezelje a szervomotorokat, kiküldje nekik a megfelelő típusú négyszögjeleket, ezáltal tartva meg az adott pozíciót. A kar összesen 6 darab szervomotort tartalmaz, 5 kicsit és 1 nagyobb, ez felel a talpzat és a kar többi részének összeköttetéséért, és ennek köszönhetően fordul el a tengelye körül.

A többi motor az egyes karmodulok mozgatásáért felelős, és itt kap igazán szerepet a vezérlőelektronika. A szervomotorok működéséhez, be kell állítani egy adott pozícióra, amely pozíció rögzítéséről a benne található visszacsatoló modul gondoskodik. Ugyanis a vezérlőelektronika majd ennek segítségével fogja tudni (ehhez képest fogja értelmezni), hogy milyen pozícióban van éppen az adott motor tengelye. Ha beállítottuk pozícióba, és nem kap újabb utasítást, akkor viszont elenged, ami azt eredményezi, hogy súlyából adódóan összecuslik a kar. Az elektronika tulajdonképpen ennek megakadályozására szolgál, így ha aktiválunk egy szervomotort, akkor az beáll egy alaphelyzetbe, amit a vezérlőmodul biztosít adott frekvenciájú és intenzitású négyszögjelekkel. Ezt a modult egy 8 bites PIC8 processzor vezérli, melynek beégetett programja van. Ezen tehát nem lehet változtatni, ezért a fellépő vezérlési hibákat, hiányosságokat, az ezt vezérlő külső programból kell kijavítani. A

processzor programja a gyártó honlapján elérhető³, assembly nyelven íródott, ami gépközelisége miatt sokkal nehezebben értelmezhető, mint egy magasabb szintű programnyelven íródott szoftver. Az egész vezérlőelektronikai modul egy soros átalakító IC közbeiktatásával, RS232-es szabványú soros port segítségével tud kommunikálni a vezérlési utasításokat adó számítógéppel. Ugyanis az alap program feladata, hogy adott pozícióba mozgat és megtart ott motorkihajtásokat. Ezeket a pozícióra vonatkozó utasításokat a soros vonalon egy külső vezérlőprogrammal tudjuk eljuttatni hozzá. Jelen robotkar esetében - a gyártó céggel együttműködve - annyi szerkezeti módosítást eszközöltem, hogy az IC kicserélésével egy csatlakozóra vezeték nélkül vezérelhetővé válik egy soros porton kommunikáló eszköz közbeiktatásával. A vezeték nélküli internetes összeköttetést használó, szintén PIC processzorral ellátott berendezés a fentebb említett módon csatlakozatható. Az ilyen processzorral rendelkező modul használatával már nem kell közvetlen soros porton összeköttetést biztosítani a vezérlést végző számítógéppel.

6.3 A wireless modul

A wireless modul típusa Wi-Fi Comm Demo Board, a Microchip nevű mikrokontrollerekkel foglalkozó cég kínálatából. A panel a Microchip MRF24WB0MA Wi-Fi moduljával és nagyteljesítményű PIC32 mikrovezérlőjével szerelt. A gyártó így fogalmaz a termékével kapcsolatban:

„A következő generációs, hivatalosan minősített MRF24WB0Mx beágyazott Wi-Fi modulok vezérlőszoftvere egyszerűen használható API interfésszel rendelkezik a Microchip ingyenes TCP/IP szoftverkönyvtára, ill. a 8, 16 és 32 bites PIC mikrovezérlők felé. A Microchip kislevegyszású, beágyazott Wi-Fi moduljainak segítségével egyszerű eszközök is könnyen az Internetre csatlakoztathatók, megspórolva az RF áramkörök fejlesztésének komplex feladatát, magas költségét valamint a hivatalos minősítés beszerzésének munkáját. A modulok további költségeket megtakarítva gyárilag előreprogramozott MAC címmel rendelkeznek. A Microchip továbbfejlesztette az ingyenes TCP/IP szoftverkönyvtárát az EZconfig és ZeroConfig funkciókkal, melyek leegyszerűsítik a Wi-Fi hálózatok üzembe helyezését és konfigurációját.”

³ <http://www.digitalnemesis.com/info/projects/picservo/>

(<http://www.chipcadextra.com/microchip-alkatreszek-es-fejlesztoeszkoezoek/alk-techn/vezetekes-es-vezetek-nelkueli-kapcsolati-megoldasok/wi-fi-zigbee-miwi/wi-fi-comm-demo-board>, 2012.12.12.)

A Wireless modul biztosítja számunkra az összeköttetést mind a vezérlővel, mind az utasításokat adó számítógéppel és ennek közreműködésével az azon lévő vezérlőprogrammal. A felépítése lehetővé teszi továbbá, hogy programrészleteket, bizonyos eljárásokat tároljunk benne, hogy adott esetben autonóm módon végre tudja hajtani azokat. Legalapvetőbb ilyen eljárás, hogy a munka végeztével visszaáll alaphelyzetbe. A szervergéppel UDP internetes protokollon keresztül kommunikál; a szervovezérlővel, vagyis ezáltal a robotkarral pedig a normál RS-232 soros port protokollja által. Az előnye a wireless modul közbe iktatásának, hogy így a szervergépnek nemhogy soros porton való összeköttetésben nem kell lennie a robotkarral, de még csak egy helyiségben sem, csupán azonos vezeték nélküli hálózaton. Nem is feltétlenül indokolt, hogy a szerver erről a hálózatról érje el az internetet. Itt csak annyiban játszik szerepet, hogy az összeköttetést kell biztosítani, egy stabil, kiváló jelerőséget biztosító útválasztó által. Ezt viszont célszerű ténylegesen az adott virtuális laboratórium falain belülre elhelyezni.

Előnyös tulajdonsága egy ilyen Router-nek⁴, hogy több feladatot is el tud látni. Kiemelkedő azon tulajdonsága, hogy egy adott időintervallumban akár több robotkar különálló vezérlésére van lehetőség, vagy ami még lényeges, hogy a Husar eRover felfedezőrobotok vezérlése is megoldható egyidejűleg ezen hálózati eszköz felhasználásával. Ami azért előnyös tulajdonság, mert erőforrás-kímélő, ugyanis annyival kevesebb villamos energiára, hálózati kábelre és router-re van szükség a komplexum üzemeltetéséhez.

Egyazon router -re bízni minden lényeges feladatot elsőre nem tűnhet optimális megoldásnak, de biztonságos módszernek semmiképp. Ennek a megoldásnak a centrális problémája a biztonság, mivel ebben az esetben a router meghibásodása esetén minden ehhez kapcsolódó robotmodul megbénul és irányíthatatlanná válik. Ezt figyelembe véve is még mindig költséghatékonyabb egy eszköz alkalmazása ezekre a célokra. Mivel minimális hálózati adatforgalmat bonyolít le egy robot, főleg mivel nyugtázatlan UDP csomagokkal kommunikál, technikai szempontból egyszerűen nincs szükség több hálózati erőforrásra. Az adott készüléknek ugyanis csak kicsi hálózati csomagok továbbítását kell végezni, valamint a

⁴ útvonalválasztó, hálózati eszköz az OSI modell 3. azaz hálózati rétegből

megfigyelő kamera képének közvetítését. De még ezzel együtt sincs lefoglalva a teljes sávszélesség, amire az itt használt eszköz képes.

6.4 A szerver

A szerverprogram lényege, hogy összeköttetést biztosítson a kliensprogram és a robotkar között. A szerver kapja az utasításokat a kientől, ezeket fogadja, dekódolja és a megfelelő robotkarnak továbbítja. Az egyszerű elmélet mögött komplex gyakorlat rejlik. Ugyanis a szervernek le kell kezelni az összes kientől kapott információt, ki kell választania, hogy melyik kliens az, amelyik valóban jogosult az adott időintervallumban az adott feladat végrehajtására, és biztosítania kell számára a kívánt eszközt. Tehát a szerverprogramnak tulajdonképpen rendelkezni kell valamilyen autentikációt végző protokollal, amit jelen implementációban egy online adatbázist segítségével tud megvalósítani. Elviekben mindenki tudja, hogy mikor jogosult használni az adott robotkart (online regisztrációs rendszer, lásd. 6.6 fejezet). Azonban ha valaki mégis próbálkozna nem regisztrált időpontban belépni ez a módszer megakadályozza. A szerverprogram továbbá figyeli az éppen aktuálisan elérhető robotkarokat, amennyiben több is található IP valamint MAC cím alapján kilistázza/kiszűri őket és ahhoz ad kontrollálási jogot - az előzőleg hitelesített kliens alkalmazásnak - amelyik éppen szabad vagy legelől van a listában. Amint ezzel kapcsolatot hozott létre, a szerver veszi át a tényleges irányítást, persze a wireless modulon keresztül. Ez abban nyilvánul meg, hogy fogadja a kliensprogramtól kapott jeleket, majd dekódolja és annak megfelelően továbbítja a WI-FI modulon keresztül a robotkar szervovezérlő elektronikájának. Ezáltal történik meg a mozgatás vagyis a tényleges irányítás. Mindeközben a szerver fogadja a karhoz tartozó kamera képét és a megfelelő csatornán, és átviteli protokollon keresztül továbbítja a kliens felé, ami majd a megjelenítést végzi. A szerverprogramnak célszerű egy a kliensektől elkülönített gépen futnia, a robotkarok, vagyis a laboratórium közelében. Nevezzük ezt szerver gépnek, persze nem szükségeszerű olyan szintű hardvert biztosítani, mint mondjuk egy egyetemi webszerver. Egy átlagos teljesítményű asztali számítógép (2 magos processzor, 4GB RAM, internet hozzáférés), vagy alsó-középkategóriás laptop (2 magos processzor, 4GB RAM, hálózati tápellátás, vezetékes vagy vezeték nélküli internet hozzáférés) is ugyanúgy el tudja végezni a feladatot.

Annyi kikötés van, hogy ugyanahhoz a hálózathoz kell csatlakoznia, mint a robotkart vezérlő vezeték nélküli modul, és ezen felül szükséges hozzá internetkapcsolat, de minimum

összeköttetés a webes adatbázissal és egy TCP/IP kapcsolat a kliensprogrammal. Mindez lehetséges valós internetkapcsolat nélkül is, csak ebben az esetben azonos hálózaton kell lennie az összes eszköznek, ami általában megköveteli, hogy fizikailag is közel legyenek. Maximum a tesztelési fázisban érdemes így felhasználni. Mivel az oktatási intézmények és ezek szaktanárai vagy diákjai a tanóra, esetleg szakkör keretén belül foglalkoznak a témával az idejük limitált, esetleg nagy a távolság az intézmény és a virtuális laboratórium között, ennek áthidalására javasolt internetes kapcsolattal használni a berendezéseket.

6.5 A kliens oldal

A kliens oldal feladata a robotkar vezérlés reprezentációs megvalósítása, a kontrolláló (vezérlő) felület által. Lényegében ez a modul az, amivel fizikálisan találkozik a felhasználó, amin keresztül irányítani tudja a robotkart. Mindezt itt teszi annak ellenére, hogy a tényleges vezérlés nem itt történik. A kliens oldali program egy ablakos felület nyomógombokkal, üzenetek kiírására alkalmas mezővel, valamint a kamera képének megjelenítésére szolgáló kameraképet közvetítő felülettel. Tervezése során elsődleges szempont a felhasználóbarát kialakítás, de legalábbis, hogy a célközönség számára biztosan érthető és egyértelmű legyen. Ez abban nyilvánul meg, hogy a gombokon jelölve van az adott szervok száma és a mozgítás iránya, valamint a nem mozgató végző gombok funkciója is. Mindenki igényeinek kielégítésére nincsen lehetőség, ezért egy optimumot kell találni.

Jelen helyzetben az egyszerűség, logikus átlátható kezelőfelület a fő irányvonal, ezért alakult ki a gombokkal vezérelhető felület. A szöveges mező azt a célt szolgálja, hogy üzeneteket közvetítsen a rendszerből a felhasználó felé. Ezek az üzenetek lehetnek például a robotkar szervóiról visszakapott jelek vagy akár a szerverrel való összeköttetés létrejöttét vagy éppen létre nem jöttét visszajelző szöveges kiírások. A média mező egyértelműen a kamera képének megjelenítését szolgálja, hangot nem közvetít, mert szükségtelen és ez esetben teljesen feleslegesen terhelnének még ezzel is az összeköttetés sávszélességét. A kliensprogram a szerverrel TCP/IP kapcsolat segítségével van összeköttetésben, általában internet segítségével. A kliens az adott gombok lenyomásának függvényében küldi a szervernek az adatsomagokat, melyeket az feldolgoz és továbbít a vezérlőnek. A TCP nyugtákon kívül a kamera által visszaküldött képen bizonyosodhatunk meg arról, hogy vezérlőparancsunk célba ért-e vagy sem. A vezérlőfelületet design szempontból felesleges nagyon kihangsúlyozni, inkább törekedtem a letisztult formák és egyszerű színek tárházából dolgozni. Nem lényeges,

hogy miként néz ki a felület, persze azért bizonyos alapvető esztétikai és funkcionalitási követelményeknek meg kell felelnie: letisztult átlátható ablakszerkezet és egyértelmű vezérlőgombok.

Úgy gondolom ezt az alap, fejlesztőkörnyezet adta 'basic' sémának köszönhetően teljesíteni lehet. A szín szintén nem hordoz nagy jelentőséget ebben a stádiumban, annyi volt a kikötés, hogy könnyen olvasható átlátható felületet hozzak létre, melyet a diákoktól kezdve a tanárokig bárki könnyen átlát. Ideális esetben bárki elsajátíthatja a használatát könnyen és egyszerűen, ha valami mindezek ellenére nem lenne egyértelmű ott a súgó funkció, tutorial (oktató anyag) esetleg a 'help' menüpont nyújt segítséget.

A kliensoldal nincs szerves összeköttetésben a felhasználó hitelesítését végző webes felülettel, de a szervernél nyilván van tartva, hogy melyiknek milyen azonosítószáma van, ezt a kliensprogram feltelepítése során kapják, és hogy az éppen jogosult-e irányításra vagy nem jogosult. Lehetőséget biztosítottam a folyamatba való betekintésre azok számára is, akik nem jogosultak a vezérlésre. Ehhez hoztam létre a spectator módot ami arra szolgál, hogy megfigyelőként tekintsünk be a mások által irányított folyamatokba. Megfigyelő módban csak annyit tehet az érdeklődő, hogy a közvetített kameraképet nézi, akár egy mások által irányított robotkarról is, ha éppen van ilyen a rendszerben.

6.6 A webes felület

A webes felületen tudnak regisztrálni az adott intézmények a kapott kliensprogram azonosítójának beírásával, valamint egyéb adataik (név, cím, elérhetőség, jelszó) megadásával. Regisztrációt követően be tudnak lépni egy online felületre. Ez a felület tartalmaz egy naptárat, időbeosztást arra vonatkozóan, hogy mikor van szabadon vezérelhető robotkar. Itt tudnak az egyes intézmények helyet - ami valójában inkább időpontot jelent - foglalni maguknak. Ha egy időpontot valaki lefoglal, akkor az megjelenik az adatbázisban is, és nem enged oda másokat a program. Ezen kívül kap egy azonosítót, amit később majd a szerver kezel az autentikációs folyamat során. Ez annyit jelent, hogy abban az időpontban az adott robotkart biztosan a rendelkezésére tudja bocsátani annak, aki előzőleg igényt tartott rá. A felület és a szerver is ugyanazon internetes adatbázissal vannak összeköttetésben, ez garantálja, hogy a szerver ténylegesen meg is kapja a szükséges kódokat. Továbbá egy statisztikát is vezet arról, hogy ki mikor, meddig használta melyik kart. Ezt a funkciót persze feltüntetésre kerül a regisztrációkor, és az így nyert adatokat csak az erre jogosult, az egész

rendszert üzemeltető szakemberek tölthetik le, vagy akiknek ezek a személyek jogosultságot adnak hozzá.

7. A megvalósításhoz használt eszközök, felhasznált technológiák

A kliensoldali és szerveroldali programok C++ nyelven íródtak és a QtCreator nevű fejlesztőkörnyezettel és az ehhez tartozó grafikus megvalósítással interpretálódtak.

A QtCreator egy szabad forráskódú, elsősorban Linux platformra készült, de más platformokon is elérhető integrált fejlesztőkörnyezet. A tervezői úgy alkották meg és fejlesztették ki, hogy Qt fejlesztők igényeihez maximálisan alkalmazkodjon. Mivel ez egy fejlett forráskód-szerkesztő természetesen rendelkezik ezen programok előnyeivel és alapvető funkcióként tartalmaz többek közt szintaktikai kiemelést és kódkiegészítést. További előnyös funkciói közé sorolandó, hogy van benne kódrészlet összecsukási funkció. (Adott kódrészletek, mint például egy függvény vagy ciklus, gombnyomással egy sorossá sűrítethetők össze, persze valójában csak megjelenítési szempontból, mert ugyanúgy megmarad a kódban csak nem látszik.) Amely elősegíti a gyors navigálást és könnyen átláthatóvá tesz egy-egy hosszabb forráskód-részletet. Nem elhanyagolható előny továbbá a zárójelpárok kiemelésének megvalósítása, ami szintén az átláthatóság szempontjából fontos.

A fejlesztő a következőképp nyilatkozik róla:

„A Qt Creator azokra a funkciókra helyezi a hangsúlyt, melyek segítik az új Qt felhasználókat az elindulásban és a tanulásban, ugyanakkor növelik a tapasztalt Qt fejlesztők produktivitását.”

(http://qt-project.org/wiki/CategoryToolsQtCreator_Hungarian , 2012.12.12)

A fejlesztőkörnyezet ezeken kívül tartalmaz még vizuális hibakeresőt, mely sok Qt osztály felépítésével tisztában van, és rögtön jelzi, ha elírtunk valamit vagy megjegyzése van az adott részlethez. Lehetőségünk van továbbá sorról-sorra illetve parancsonkénti végrehajtásra, ami szintén a hibakeresést segíti. Tartozik mindezekon kívül egy grafikus felület (GUI) tervező eszköz is ehhez a környezethez mely az alábbi irányelveket figyelembe véve működik:

„Integrált GUI elrendezés (layout) és űrlap szerkesztő C++ projektek számára, amely lehetővé teszi, hogy gyorsan hozzassunk létre widgeteket és dialogokat a képernyőn

megjelenő reprezentációjukat felhasználva. Az űrlapok teljesen működőképesek és azonnal meggyőződhetünk róluk, hogy ugyanúgy néznek ki, mint ahogy szeretnénk.”

(http://qt-project.org/wiki/CategoryToolsQtCreator_Hungarian, 2012.12.12)

Tehát a beépített grafikusfelület szerkesztő rendszeren keresztül készül a vezérléshez szükséges program. Mivel ez bele van integrálva a fejlesztőkörnyezetbe, használata valóban gyors és praktikus. A gomboktól percek alatt el lehet jutni az általuk vezérelt függvényekhez, funkciókhoz. A forráskódot a szabványos C++ projektfájlokhoz hasonló módon külön állományokban lehet tárolni, úgymint header, osztály és main kategóriák. Az alapprogram telepítése során automatikusan feltelepülő programkönyvtáron és függvénykönyvtárakon kívül elérhető még rengeteg másik is a gyártó honlapján.⁵ Ezekre azért van szükség, mert például a hálózati kapcsolatok felépítéséhez vagy éppen az adatbázissal való kommunikációhoz mind külön-külön függvények szükségesek, amik nem feltétlenül vannak benne egy ugyanazon függvénykönyvtárban.

A webes felület PHP nyelven megírva, MySQL adatbázisban tárolja a regisztrációkat és az időbeosztást, mert ezt el lehet érni PHP-n keresztül a webes oldalról. A szerver program is el tudja érni a megfelelő Qtcreator-ban használatos függvénykönyvtár beillesztésével. A szerveroldali program ily módon tudja ellenőrizni, hogy melyik kliens próbál csatlakozni és meg tudja ítélni, hogy szabad-e neki vagy sem. Ezt az adatbázist a rendszeradminisztrátor kívülről is meg tudja nézni, bele tud írni vagy éppen törölni belőle ha szükséges. Például ha jogosulatlan foglalás történt vagy esetleg rendszerhiba lépett fel. Adminisztrátorra persze nincs állandóan szükség, a rendszer önálló működésre van tervezve. Jóllehet az üzembe helyezés során valamint az adatbázis karbantartásánál megfelelő tudás és rendszerismeret szükséges a feladat tökéletes elvégzéséhez.

7.1 A PHP

A PHP egy kifejezetten az internetre specializálódott szerveroldali script nyelv. A HTML oldalakba az oldal megnyitásakor lefutó PHP kódot ágyazhatunk. Ez esetünkben akkor nagyon hasznos, amikor a weben keresztül történő regisztrációt kell megvalósítani. A PHP első verziója 1994-ben keletkezett egyetlen ember munkájaként (Rasmus Lerdorf) majd az

⁵ <http://qt-project.org/>

évek során rengeteg átalakításon és fejlesztésen ment át. (*Welling Tomson, 2009*) A PHP jelenlegi fő változata az 5-ös, jóllehet én mégis a 4-es verziót használom, mert az már régóta kiforrott, jól használható technológia, míg a legújabb kiadásban vannak olyan újdonságok, amik kevésbé ismertek és használatosak jelenleg. A PHP nyílt forráskódú projekt, tehát bárki ingyen hozzáférhet a forráskódhoz, szabadon használhatja és felhasználhatja. Legfontosabb erősségei versenytársaival szemben többek közt az alacsony költség, a teljesítmény, a skálázhatóság, az objektumorientált programozás széleskörű támogatása, a rengeteg ingyenesen hozzáférhető függvényreferencia és dokumentáció valamint támogatás és végül, de nem utolsó sorban a platformfüggetlenség. Jelen projektben azonban a fő irányelv, ami szerint ezt választottam, az volt, hogy könnyen és hatékonyan lehet egy PHP programot összekapcsolni egy MySQL adatbázissal.

A MySQL egy stabil és gyors relációs adatbázis-kezelő rendszer, mely szintén nyílt forráskódú és könnyedén összekapcsolható a PHP-vel. (*Welling Tomson, 2009*)

7.2 A MySQL

Az adatbázis lehetővé teszi számunkra az adatok hatékony tárolását, keresését, valamint rendszerezését és rendezését különböző szempontok szerint. A MySQL kiszolgáló szabályozza az adatokhoz való hozzáférést ezzel biztosítva, hogy egyidejűleg többen is hozzáférhessenek az adatokhoz, és használhassák azokat, valamint biztosítja, hogy csak jogosult felhasználók férhessen hozzá az adott rekordokhoz. (*Welling Tomson, 2009*)

A MySQL is mint sok más adatbázis rendszer a szabványos adatbázis-lekérdező nyelvet, a strukturált lekérdező nyelvet (Structured Query Language - SQL) használja. A MySQL főbb erősségei, a platformfüggetlenség – ezáltal hordozható – az alacsony költség, a nagy teljesítmény, az egyszerű használhatóság és hogy nyílt forráskódú, aminek következményeként rengeteg ingyenes leírás és támogatási anyag található a használatához.

(*Welling Tomson, 2009*)

Jelen projektben az ingyenessége mellett az egyszerű használhatóságon kívül azért választottam, mert ezt egyaránt el tudom érni PHP és C++ segítségével is. Az időpont foglalásos regisztrációs rendszer miatt vált égetően szükségesé közbeiktatni egy adatbázist, ami rögzíti a foglalásokat és kontrollálja őket. Ugyan a relációs adatbázis adta előnyöket kevésbé tudjuk kihasználni jelen helyzetben, de igazából szükség sincs rá. A lényeg hogy egyszerűen bele tudjuk írni a szükséges adatokat, amiket a másik oldalon rögtön ki is tudunk

olvasni a szerver program közreműködésével. Említésre méltó körülmény továbbá, hogy mivel lényegében platformfüggetlen nyelvről van szó, egyaránt fel tudjuk használni Microsoft Windows alapú webserveren ugyanúgy, mint például UNIX alapúakon. Ennek ott mutatkozik meg legnagyobb előnye, hogy ha az adott intézmény aki a virtuális laboratóriumot kialakítja nem szükséges hogy változtasson az eddigi szoftverjein és másik operációs rendszert telepítsen a szerverére vagy új webszerveret szerezzen be a virtuális laboratórium programban való részvétel miatt. Mindezek mellett, ha nincs rá lehetőség az intézményben, akár külső szerver is bérelhető, és még nagyon nagy tárhelyet sem igényel (15-20 MB) mivel nagyon kevés rekord tárolására van szükség alkalmanként. Amiket ráadásul egy bizonyos idő elteltével kidobhatunk, hogy még az se foglalja a helyet. Összességében azért kellett munkába állítani egy adatbázist, hogy kezelje a regisztrációkat és időpontfoglalásokat. Úgy gondolom, hogy ezt a feladatot a jelen helyzetben használt MySQL adatbázis tökéletesen és alacsony költségek mellett el tudja látni. Mivel a regisztrációs felület szintén az interneten elérhető, ebből következik, hogy valamelyik webes programnyelv segítségével kell megvalósítani. A HTML-be ágyazott PHP véleményem szerint költséghatékonyan végzi a feladatát, mindezt maximális teljesítmény mellett.

A regisztráció webes felületen történik, vagyis böngészőből végezhető, ebből kifolyólag platformfüggetlen, aminek szintén nagy előnyei vannak mind programozói, mind felhasználói szempontból. Programozó szempontból az, hogy nem szükséges többször, több operációs rendszerre specifikálva megírni a programot mégis használható bármelyikről. Nem célszerű csak egy adott platformra megírni a programot mivel jelen körülmények között egyre szélesedik a felhasználható operációs rendszerek tárháza és egyszerűen nem lehetne külön mindenki igényeit maradéktalanul kielégíteni. Felhasználó szempontból pedig azért jó, mert ha valaki használni akarja, nem azt kell nézni, hogy milyen operációs rendszere van, melyik kiadás és hogy ezen vajon működik-e, netán be kell szereznie egy másikat, ha használni akarja. Elég nyitnia egy böngészőt, beírnia a megfelelő webcímet a címsorba, majd a megjelenő oldalon az adatai megadásával regisztrálnia, vagy ha ezt már megtette bejelentkeznie. A bejelentkezési ablak megjelenítését természetesen HTML- nyelven íródott weblap végzi.

7.3 A HTML

A HTML az internet általános megjelenítési programnyelve, a weblapok nagy része ezt használja. A HTML kódot a böngésző értelmezi és fordítja le megjeleníthető formára. Végül a megjelenítést szintén az adott internetböngésző program végzi.

7.4 A TCP/IP hálózati protokoll

A kliens és szerver közti összeköttetést TCP/IP protokollon keresztül oldottam meg. A TCP egy kapcsolat alapú protokoll, amiből kifolyólag megbízhatónak mondható. Ezt úgy éri el, hogy az adatok tényleges küldése előtt kapcsolatot épít ki a küldő- és célállomások között. A kapcsolat megléte teszi lehetővé, hogy ellenőrizze az egyes csomagok állapotát. Itt gondolhatunk arra, hogy átért-e az adatcsomag, és ha a csomag hibásan vagy egyáltalán nem ért át, akkor lehetőséget nyújtson az újraküldésre. Jóllehet ezen beépített lehetőségnek ára van, mely abban mutatkozik meg, hogy jelentősen megnövekszik a fejléc mérete, valamint a feldolgozás ideje. A TCP kapcsolat jelentős elemei a csomagellenőrzés és hibajavítás, a folyamszabályozás és a nyugtázás. Ezen dolgok hatékony megvalósításához a TCP kapukat (port) és csatolókat (socket) használ, valamint kapcsolat alapú adatcserét, sorszámozást és nyugtázást. (*Reynders Wright, 2005*).

A TCP/IP több mint egy protokoll. A TCP (Transmission Control Protocol), azaz átvitelvezrlő protokollt, míg az IP (Internet Protocol) az internetprotokollt jelenti. Tulajdonképpen ez egy protokollkészlet, ami több protokoll együttműködése a feladat elvégzése érdekében. A TCP/IP ma az internet hivatalos nyelve ebből következően használata előfeltétele az internet elérésének. A TCP/IP kombinált protokoll a világ leggyakrabban előforduló protokolljai, mert a hálózati és szállítási rétegben fejtik ki hatásukat, melyet mindenkinek használnia kell, aki az internetre csatlakozik. Az IP azaz az internetprotokoll olyan szabályok együttese, mely arra szolgál hogy egy üzenetet a helyes címre eljuttassanak, nem pedig arra, hogy a fogadó valóban meg is kapja azt. Emiatt az IP kapcsolat nélküli, vagyis nem épít ki kapcsolatot mint a TCP, emiatt megbízhatatlannak tartjuk. Az IP elküldi az információt függetlenül attól, hogy fogadja-e valaki, vagy létezik-e egyáltalán a fogadó fél. Elsődleges célja a címezési séma létrehozása és folyamatos karbantartása, melyet IP-címezésnek nevezünk. Jelenleg minden az internetre csatlakozó eszköznek használnia kell IP címet. Mindegyik IP cím egyedi, az eszközökhöz statikusan és dinamikusan is rendelhetők IP címek. (*A. McMahon, 2004*). A kliens és szerver közti kommunikációnál a fentebb említett okokból

kifolyólag is, a TCP/IP protokollt választottam, mivel egyfelől ha kapcsolatot kiterjesztjük az internetre, akkor kénytelenek vagyunk az ott használt protokollkészletből dolgozni, másfelől ha már megtehetem, úgy gondoltam megbízható kapcsolatot használok ahol van rá mód. A TCP lassúsága jelen esetben elhanyagolható tényező, mert egyrészt számítógépek közt jön létre a kapcsolat, amik nagyobb teljesítményűek, mint egy mikrokontroller, másrészt pedig a megbízhatósága ellensúlyozza a sebességbeli hátrányát. Azonban a robot mikrokontrollerje és a szerver gép programja közötti kapcsolatra alkalmatlan, és túlzó is lenne ezt a protokollkészletet alkalmazni. Abban a helyzetben ugyanis nem lépünk ki az internetre, ezért nem szükségszerű a nagy és komplex TCP/IP protokollösszességet alkalmazni, hanem beérjük egy egyszerűbb fajtával is, mégpedig az UDP -vel.

7.5 Az UDP hálózati protokoll

Az UDP (User Datagram Protocol) az állomás-állomás réteg másik fő protokollja, amely a TCP -hez hasonló módon szintén az IP-t használja az adatcsomagok továbbjuttatására, célba juttatására. Az UDP az IP-hez hasonlóan kapcsolat nélküli protokoll, vagyis nincs szüksége kapcsolat kiépítésére a kommunikációt végző két állomás között. Ezen tulajdonságai miatt megbízhatatlannak nevezzük, ellentétben a már említett megbízható TCP -vel. A TCP –hez hasonló módon a csomagok itt is csatlókhhoz és kapukhoz jutnak el, ellenben itt semmi nem biztosítja, hogy valóban el is jutnak, hibátlanok maradnak és sorrendjük is megfelelő lesz. Ezeket látva jogosan merül fel a kérdés, hogy akkor tulajdonképpen miért is használjuk, de erre is megvannak a válaszok. Az UDP adatcsomagok elküldése igen csekély hálózati terheléssel és erőforrásigénnyel jár, hiszen nincsenek összehangolási paraméterek, elsőbbségi beállítások, újraküldési időzítők, sorszámok és késleltetett nyugtázás valamint csomag újraküldés sem. A fejléc rövid, a protokoll gyors és hatékony működést tesz lehetővé. Az egyetlen lényegi hátránya, hogy a csomagok célba érése nem biztosított. Az UDP -t ezért adatszórásra, valós idejű adatok átvitelére valamint általános hálózati bejelentések közzétételére használják. A protokoll különösen alkalmas videó- és hangátvitelre, ahol az alacsony átviteli költség a követelmény és az elveszett csomagok újraküldése nem szükségszerű. (*Reynders Wright, 2005*)

A mikrokontroller által használt és vezérelt modulokat egyszerűbb egy ilyen protokoll által elérni, és ezzel biztosítani, hogy nincs nagy hálózati erőforrásigénye. Továbbá az adatcsomagok küldésének gyakorisága is növelhető ezáltal. Mindezekon felül pedig elég az

egyirányú hálózati összeköttetés, tehát a robot vezetékek nélküli egységének nem szükségszerű kétirányú kommunikációt használnia. Jóllehet ezt csatlakozásnál megteszi, mert visszajelez ha létrejött a hálózati kapcsolat. Ettől a ponttól kezdve azonban, csak a kapott utasítások végrehajtására és továbbítására van szüksége, erre a célra pedig tökéletesen alkalmas az UDP. Csomagvesztési szempontból sincs nagy probléma, ugyanis amíg a vezérlőfelületen nyomjuk az adott gombot, addig az folyamatosan küldi a csomagokat, amikből ha egy-kettő elveszik útközben nem keletkezik hiba, legfeljebb egy pillanatra szakaszossá válik a kar mozgása. Mivel a szervomotorok kialakításából adódóan, folyamatosan van benne némi szakaszos mozgás, szinte fel sem tűnik, ha csomagvesztés történik. Persze ha nagyon sok csomag elveszik, az már feltűnő, de akkor a hibát nem az UDP protokoll felépítésében, hanem a rendszer valamely másik pontján kell keresni, például a hálózati összeköttetésben.

7.6 A C++

A szerver és kliens alkalmazásokat C++ nyelven írtam. Ennek több oka is van. A mai világban, a legtöbb profi szoftverfejlesztő C++ -ban dolgozik. Ennek a legvalószínűbb oka, hogy ezen a nyelven kis méretű, mégis gyorsan működő programokat írhatunk. Mindehhez a különböző platformok között hordozható robosztus környezet áll rendelkezésünkre. Manapság a rendelkezésre álló eszközökkel, fejlesztőkörnyezetekkel kifejezetten összetett és hatékony, kereskedelmi forgalomban is helytálló szoftvereket gyárthatunk. Mindezt megtehetjük viszonylag egyszerűen, persze ennek megvan az ára, meglehetősen alapos ismertekre van szükségünk mind a nyelvet, mind a fejlesztőkörnyezetet illetően. A többi programozási nyelvhez képest a C++ viszonylag fiatal. Persze maga a programozás, mint diszciplína sem mondható túl öregnek a maga alig több, mint fél évszázados múltjával. A C++ a C nyelv továbbfejlesztése, ami nem sokkal több, mint 30 éves, ezért biztosan állítható, hogy a C++ nem öregebb nála. A C++ úgynevezett magas szintű programnyelv, a gépnek az értelmezéséhez fordítóprogramra van szüksége. A fordítóprogramok (compiler) először végigolvassák a teljes forráskódot, amit a programozó begépett, majd ebből hoznak létre egy úgynevezett tárgykódú állományt (object code). A feldolgozásnak ezen első lépését fordításnak (compiling) nevezik. A fordító tehát először előállítja a tárgykódú állományt, majd később ezt az úgynevezett szerkesztő (linker) alakítja át futtatható programmá. A futtatható program az az állomány, amit az adott operációs rendszer segítségével képesek

vagyunk lefuttatni. (*Liberty B. Horvath, 2008*) Egy adott C++ forráskódot minimális átalakítások árán le tudunk fordítani ugyanúgy Linux mint például Windows alapú operációs rendszeren. Tehát maga kód szinte platformfüggetlen. Azért csak szinte, mert vannak platformspecifikus metódusai, melyeket nem tudunk univerzálisan használni. Persze ez a platformfüggetlenség nem úgy értendő, mint a már előzőekben tárgyalt PHP esetében. Amíg ott ugyanazt a kódot tudtuk felhasználni bármilyen operációs rendszerről – jóllehet ott nem kellett lefordítani speciálisan erre vagy arra a platformra – a C++ esetében ez kicsit bonyolultabb de korántsem lehetetlen. Jelen esetben azonban egy Linux alapú fejlesztőkörnyezetben Linux alatt fordítom a kódot. Több jó okom is van rá, hogy Linuxot válasszak operációs rendszerként. Elsősorban az, hogy nyílt forráskódú, szabad szoftver és ezzel költséghatékony tudok maradni. További szempont, hogy QtCreator könnyen elérhető és telepíthető rá és mivel ezzel készítem a projektben foglalt programok nagy részét, ez is nagyon fontos szempont.

7.7 Az operációs rendszer és a fejlesztőkörnyezet

Jelen projekt megalkotásához Ubuntu 12.04 típusú Linux operációs rendszert használok, KDE grafikus felülettel. C++ fordítóként a QtCreator által is használt gcc -t. A QtCreator – ban a TCP/IP kapcsolat felépítéséhez a QTcpSocket osztályt használom, mely a QAbstractSocket leszármazott osztályaként érhető el. Funkciója, hogy kiépítse a TCP kapcsolatot a két kommunikáló fél között és biztosítsa az adatfolyamot.

(<http://doc.qt.digia.com/qt/qtcpsocket.html>, 2012.12.12)

A szerver és a mikrokontroller vezetékek nélküli egysége közötti UDP kommunikációt a QUdpSocket osztály segítségével valósítom meg, mely szintén a QAbstractSocket egy leszármaztatott osztálya. A QUdpSocket osztály feladata, hogy UDP csomagokat küldjön, és fogadjon, vagyis az UDP protokollon keresztül történő kommunikáció lebonyolítása.

(<http://qt-project.org/doc/qt-4.8/qudpsocket.html>, 2012.12.12)

A szerver és az adatbázis közti kommunikációt a QSqlDatabase osztály teszi lehetővé. A QSqlDatabase osztály biztosítja azt a kapcsolatot, amellyel csatlakozni tudunk egy MySQL adatbázishoz a C++ programon keresztül, és ezáltal ki tudunk olvasni abból rekordokat vagy éppen bele tudunk írni. A kapcsolat kiépítése során meg kell adnunk, hogy milyen drivert kívánunk használni, tehát milyen adatbázishoz csatlakozunk. Utána át kell adni a programnak a szükséges csatlakozási paramétereket, mint felhasználónév, jelszó, elérési útvonal és

adatbázis neve. Ezután már automatikusan csatlakozik és el tudja végezni a szükséges műveleteket. (<http://doc.qt.digia.com/qt/qsqldatabase.html>, 2012.12.12)

8. Távlati tervek

A projekt hasznáról és jövőbeli alkalmazásáról az alábbiakban a projekt egyik atyja, Dr. Hegyi Sándor vélekedése:

„A robotkar fejlesztése tökéletesen betölti azt a szerepkörét, amire készült. A fő szempont egy olyan robot(kar) család megalkotása volt, melyek kis méretekkel rendelkeznek, olcsók, gyorsak, viszont nagy számban vannak jelen a műveleti területen. Így a megoldandó feladatokat, méréseket a robotok közt szét lehet osztani. Továbbfejlesztési cél, hogy ezek a robotok, robotkarok képesek legyenek egymással közösen együttműködni, rendszerbe szerveződni. Az ehhez szükséges vezetékek nélküli kommunikációs eszközök gyártása folyamatban van és még csak néhány példány készült el.

Továbbá célunk még az oktatásban való alkalmazhatóság volt. Olyan robot egységek létrehozása, melyek mind a felső, mind a középiskolai oktatásban megállják a helyüket. Ezzel egy új innovációs irányvonal is elindult az oktatásban, méghozzá az alkalmazott informatika iránya.

A XXI századi oktatásban célunk az is, hogy tantárgyak legyenek érdekesek és aktuálisak. A végzett feladatok tegyék a diákokat képzeletgazdaggá is. A vizuális oktatás bevezetésével, és a modern technológia közvetlen a tantermekbe vitelével, ez mind lehetségessé vált. Középiskolai tanulmányok folytán könnyebb a programozás megértése, ha a diákok közvetlenül látják a lefutó programrészek eredményeit. Az ilyen és ehhez hasonló szakkörök indítása, melyek kreatív robotépítéssel foglalkoznak, nélkülözhetetlenek lesznek az oktatásban. Elősegítik az olyan készségek kifejlődését, melyeket későbbi éveik során is könnyedén tudnak alkalmazni, például kreatív gondolkodás, az önálló probléma megoldás, és a csapatmunka.”

Ez egy remek lehetőség, mind a fiatal fejlesztőmérnököknek és programozóknak mind a jövő és jelen diákjainak. Előbbieknek munkalehetőség, kihívás megoldandó feladat, utóbbiaknak tanulási lehetőség és nem utolsósorban remek szórakozás. Bízom benne, hogy a projekt eléri célját, később esetleg továbbfejlődik, nagyobb, több mindenre képes robotokkal. Mindehhez remek alapot biztosít a jelen projekt célját képező online irányítású robotkar és hozzá tartozó szoftverhátér. Innentől már csak egy lépés, hogy nagyobb méretű és stabilabb anyagú, fém

robotkart építsünk melyet fel lehet majd szerelni különböző szenzorokkal és eszközökkel. A hőmérséklet és páratartalom mérő szenzorok, a fűrő vagy véső fej kialakítás rengeteg további fejlesztési és megvalósítási lehetőséget teremt. Tehát az online irányítású robotkar projekt, nem csupán szimulációs célokra alkalmas, hanem rengeteg továbbfejlesztési lehetőséget is magában hordoz, melyek csak némi kreativitást igényelnek. Így nemcsak a középiskolások számára remek tanulási lehetőség, hanem a felsőoktatásban résztvevő fiatal fejlesztőmérnökök számára is szakmai kihívás, hogy mit lehetne még kihozni belőle.

Irodalomjegyzék

1. BÉRCZI SZANISZLÓ, HEGYI SÁNDOR, KOVÁCS ZSOLT, FABRICZY ANIKÓ, FÖLDI TIVADAR, KERESZTESI MIKLÓS, CECHE VILMOS, DROMMER BÁLINT, GRÁNICZ KATALIN, HEVESI LÁSZLÓ, BORBOLA TAMÁS, TÓTH SZABOLCS, NÉMETH ISTVÁN, HORVÁTH CSABA, DIÓSY TAMÁS, KOVÁCS BARNA, BORDÁS FERENC, KÖLLŐ ZOLTÁN, ROSKÓ FARKAS (szerkesztette: Bérczi Szaniszló) (2001): Kis Atlasz a Naprendszeréről (2): Planetáris felszínek vizsgálata a Surveyor alapján megépített Hunveyor kísérleti gyakorló űrszondával. UNICONSTANT. Budapest-Pécs-Szombathely (ISBN 963 00 6314 XÖ, 963 00 6316 6).
2. BÉRCZI SZANISZLÓ, HEGYI SÁNDOR, HUDOBA GYÖRGY, BALOGH ZOLTÁN, BÍRÓ TAMÁS, BORNEMISZA IMRE, CSAPÓ LÁSZLÓ, DROMMER BÁLINT, GERESDI ATTILA, HALÁSZ ANTAL, HARGITAI HENRIK, IMREK GYULA, KERESZTESI MIKLÓS, KÓKÁNY ANDRÁS, NAGY ATTILA, PÁPAI TIVADAR, SAMU NORBERT, SASVÁRI GÁBOR, SENYEI RITA, SIROKI LÁSZLÓ, VARGA TAMÁS (szerk. Bérczi Szaniszló) (2006): Kis atlasz a Naprendszeréről (10): Fejlesztések a Hunveyor - Husar űrszonda modelleken. ELTE TTK Kozmikus Anyagokat Vizsgáló Űrkutató Csoport, Budapest, (ISBN 963 00 6314 XÖ, ISBN 963 86873 6 3)
3. BÉRCZI SZ. – HEGYI S. – HUDOBA GY.: *A Hunveyor gyakorló űrszondamodell sokoldalú felhasználása a fizika tanításban és a tantárgyi kapcsolatokban*, Fizikai Szemle 2008/2, 55.
4. DEON REYNERS – EDWIN WRIGHT: *TCP/IP és Ethernet hálózatok a gyakorlatban*, Kiskapu Kft., Budapest, 2005.
5. <http://doc.qt.digia.com/qt/qsqldatabase.html>, 2012.12.12
6. <http://doc.qt.digia.com/qt/qtcpsocket.html>, 2012.12.12
7. <http://planetologia.elte.hu/1cikkek.phtml?cim=hunveyor.html> 2012.12.12
8. <http://qt-project.org/doc/qt-4.8/qudpsocket.html>, 2012.12.12
9. http://qt-project.org/wiki/CategoryToolsQtCreator_Hungarian , 2012.12.12
10. http://qt-project.org/wiki/CategoryToolsQtCreator_Hungarian, 2012.12.12

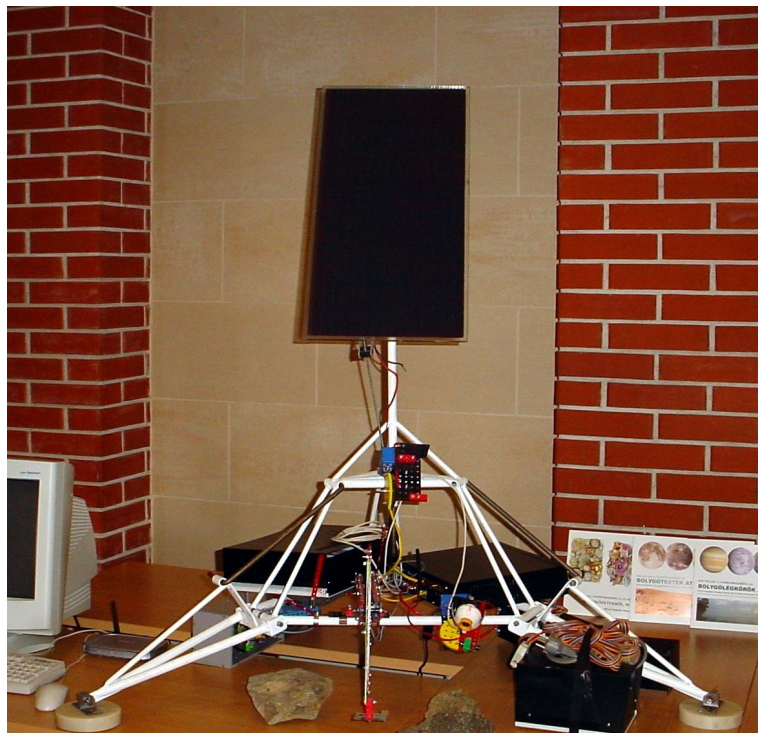
11. <http://www.chipcadextra.com/microchip-alkatreszek-es-fejlesztoeszkoezoek/alk-techn/vezetekes-es-vezetek-nelkueli-kapcsolati-megoldasok/wi-fi-zigbee-miwi/wi-fi-comm-demo-board>, 2012.12.12
12. <http://www.firstlegoleague.org> 2012.12.15
13. JESSE LIBERTY – DAVID B. HORVATH: *Tanuljuk meg a C++ programozási nyelvet 24 óra alatt*, Kiskapu Kft., Budapest, 2008.
14. RICHARD A. MCMAHON: *PC-hálózatok a gyakorlatban*, Panem Könyvkiadó, Budapest, 2004. LUKE WELLING – LAURA THOMSON: *PHP és MySQL webfejlesztőknek*, Perfact-Pro Kft., Budapest, 2010.
15. STEVEN R. TERRELL: *The Internet and Higher Education*, Chicago USA, 2008

Mellékletek

1. sz. melléklet: Mintatanterv középiskolások számára űrkutatási szimulációhoz

1. óra: Tanmenet áttekintése, űrkutatás történet fontosabb események(ember az űrben, ember a Holdon, kutatószondák különböző égitesteken, Curiosity landolása a Mars felszínén)
2. óra: Robottechnológia alapja, fejlődése az informatikáéval párhuzamosan.
3. óra: Hunveyor program ismertetése
4. óra: Dolgozat az eddigiekből
5. óra: Online irányítású robotkar megismerése, kezelőfelület használatának elsajátítása, alapvető funkciók kipróbálása.
6. óra: Online irányításhoz szükséges technológiák megismerése.
7. óra: Dolgozat az új anyagból, csoportok kialakítása.
8. óra: Konkrét projektfeladat elméleti megismerése, tervek a végrehajtására. A feladat: különböző kőzetminták közül az adott színűek kiválogatása, kőzetek vizsgálata, csoportosítása és bepakolása egy robotkar melletti dobozba.
9. óra: Feladat online végrehajtása a robotkar segítségével
10. óra: Egy komplexebb feladat megtervezése. Kutatórobot elhelyezése a bolygó felszínén a robotkar melletti dobozból. Részegységeinek összeszerelése.
11. óra: Feladat online végrehajtása segítségével
12. óra: Témazáró dolgozat

2. sz. melléklet: ábrák



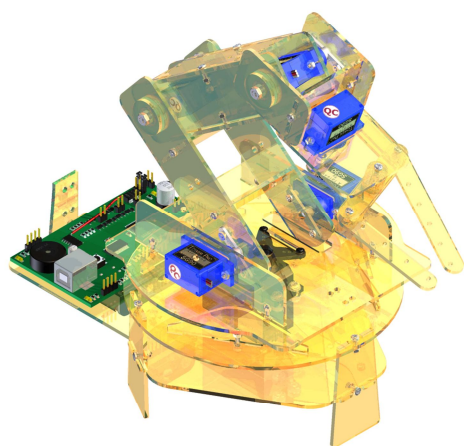
1.ábra . A Hunveyor-1 az ELTE TTK rendezett Űrnapon.



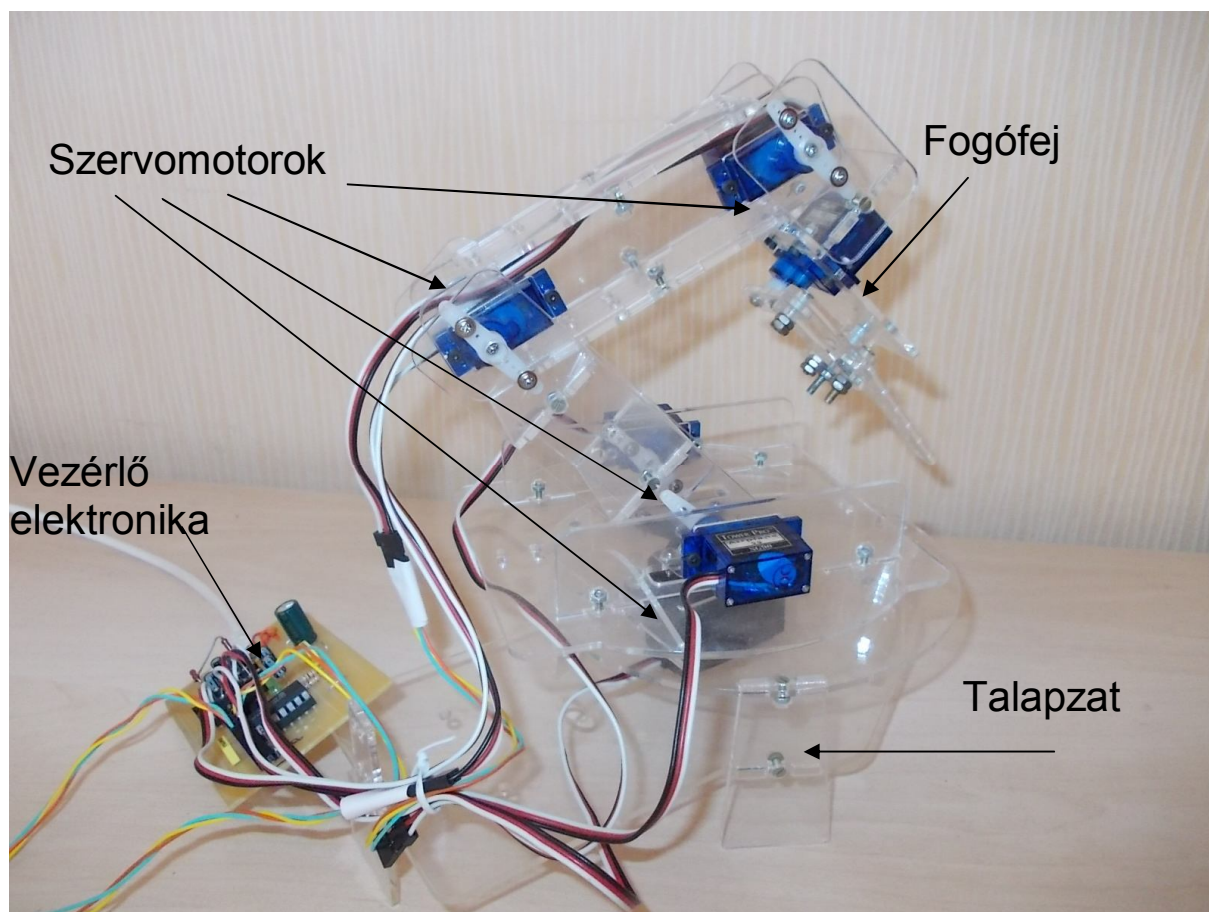
2.ábra A Husar-2d az utahi MDRS terepgyakorlaton



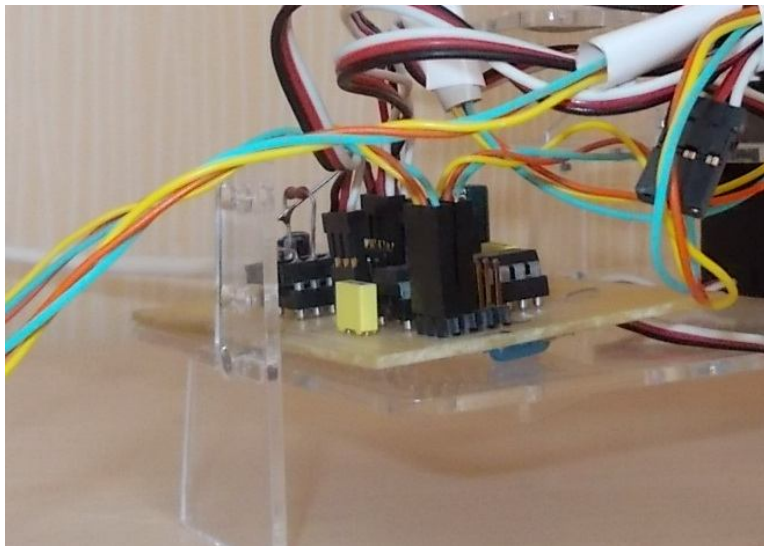
3.ábra Hunveyor-Husar-1 az ELTE TTK-n a terepasztalon közetek között (1999).



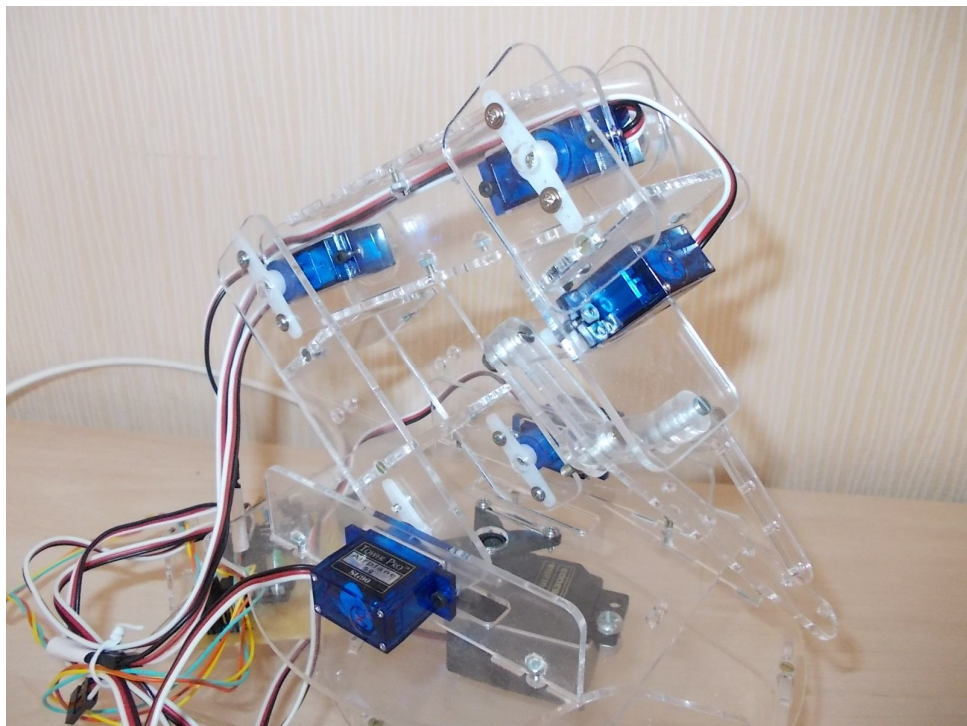
4.ábra a látványterv (Inventor modell)



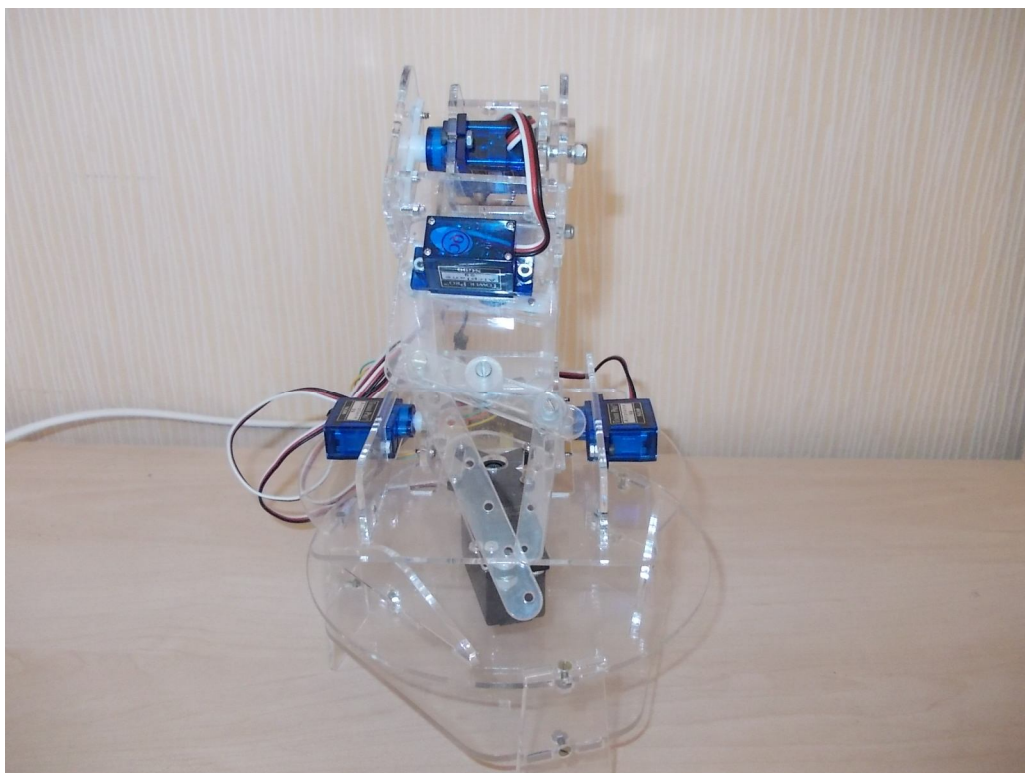
5.ábra a megvalósult robotkar és felépítése



6. ábra a vezérlőelektronika



7. ábra a kar rész



8. ábra a kar rész szemből