

Szakdolgozat

Takács Márton

Debrecen

2009

Debreceni Egyetem
Informatikai Kar
Szilárdtest Fizika Tanszék

Robotvezérlés telekommunikációs eszközökkel

Témavezető:
Dr. Szabó István
Egyetemi Docens
Tanszékvezető

Készítette:
Takács Márton
Mérnök -
informatikus

Debrecen
2009

Tartalomjegyzék

Tartalomjegyzék	3
Bevezetés	4
Robot részei	6
Mechanikai rendszer	6
Szenzoros rendszer	6
Írányítórendszer	6
Ipari robotok	6
Robotok a harcmezőn	9
Robotok a Marson	15
Szenzorok	17
Telekommunikáció	18
Wi-Fi.....	20
A feladat	21
Felhasznált programcsomagok	22
Apache	22
MySQL	23
HTML.....	24
PHP	25
LabVIEW.....	25
A megvalósítás	26
Közös oldal	27
Űrlapok elkészítése, elérése.....	28
Adatbázis / adattáblák.....	34
LabVIEW a roboton	35
Összefoglalás	38
Felhasznált irodalom	39
Köszönetnyilvánítás	40

Bevezetés

Mindig is érdeklődtem a robotok iránt, így nem volt kérdéses, hogy milyen témát válasszak. Amikor elkezdtem anyagot gyűjteni a szakdolgozatomhoz és jobban elmélyültem benne, akkor döbbsentem rá, hogy mennyire felületes ismereteim vannak a robotika terén. Tudni lehet, hogy ez a téma elég széles skálán mozog. Korábbi tanulmányaim során inkább hálózatokkal foglalkoztam, így sikerült egy számomra nagyon érdekes feladatot kitalálni. A dolgozat végére fel szeretnék építeni egy kapcsolatot egy eszköz és kezelője között, hogy egymással mindig tudjanak kommunikálni. Egy információcserét kell elképzelni ahol a robot adatokat, a felhasználó pedig parancsokat küld.

Ha robotról van szó, mindenkinek más-más juthat eszébe, mert rendkívül sokoldalú mechanizmus, szinte bármit el lehet végeztetni vele, különböző feladatokat old meg attól függően, hogy mire programozták. A szláv „robota” szóból ered, jelentése szolgaság, munka. Több definíció leírja mi is valójában, talán a legpontosabb:

„Újraprogramozható, többfunkciós manipulátor anyagok, eszközök, részegységek mozgatóására, megváltoztatására programozott mozdulatsor segítségével különféle feladatok elvégzése érdekében.” (Robot Institute of America, 1980)

Emellett hozzávehetjük még a következőket: önműködő eszköz, mely ember formájú vagy olyan feladatokat hajt végre, melyeket emberinek nevezünk. „Erő és intelligencia együttese, ahol a MI¹ találkozik a valós világgal”. Létezik helyhez kötött és mobil robot, melyeknek a feladat elvégzéséhez helyváltoztatásra van szüksége.

A munkát az ember irányítása alatt, ezzel felügyelve azt, vagy önállóan is végezheti. Általában ott kerülnek használatra, ahol az ember jelenléte veszélyes, de az olyan feladatokat is elvégzik, amit humán erővel nem lehet megvalósítani, például az űrben végzendő munka, de ha a Földön maradunk, akkor a nagy pontosságú feladatokhoz szinte nélkülözhetetlen a robotok jelenléte.

Sokaknál, sőt szinte minden háztartásban van robot, de ezek nem keverhetők össze az automatákkal, amik programozhatóak és rendelkeznek érzékelőkkel, de hiányzik az írható, olvasható memória, az adaptációs és a tanulási képesség.

¹ mesterséges intelligencia

Felhasználási területek [5]:

- Űrkutatás
 - Űrszondák
 - Leszálló egységek
 - Mobil bolygókutató egységek

- Ipari robotok
 - Összeszerelő
 - Hegesztő
 - Szerelő
 - Festő
 - Megmunkáló
 - Karbantartó

- Katasztrófaelhárító robotok
 - Tűzoltó robotok
 - Felderítő robotok (vegyi vagy sugár szennyezet területen)

- Katonai Robotok
 - Akna felkutató, megsemmisítő robotok

Robotok alkalmazási tulajdonságai (az ISO 9283 alapján)

Alkalmazhatók a csuklók száma és típusa, munkatér alakja és nagysága, mechanikai merevség és engedékenység, abszolút és ismétlési pontosság, maximális terhelés, sebesség programozási tulajdonságok, energiaigény és fizikai méret szerint.

A felsoroltak közül a későbbiekben szeretnék egy általános összefoglalást adni az ipari, haditechnikai és az űrben munkálkodó robotokról.

Robot részei

Robot három fő része:

- Mechanikai rendszer
- Szenzoros rendszer
- Irányítórendszer

Mechanikai rendszer, az akciót megvalósító részrendszer

- Helyváltoztató berendezés, mozgás a környezetben
- Manipulációs berendezés, környezeti objektumokkal való operálás (mechanika, anyagtudomány, beavatkozó szervek)

Szenzoros rendszer, az érzékelést megvalósító rendszer

- Belsőállapot, mechanikai rendszer állapota
- Külsőállapot, környezet állapota (anyagtudomány, jelfeldolgozás, adatfeldolgozás, információ-visszakérés)

Irányítórendszer, az akció és az érzékelés közti kapcsolat

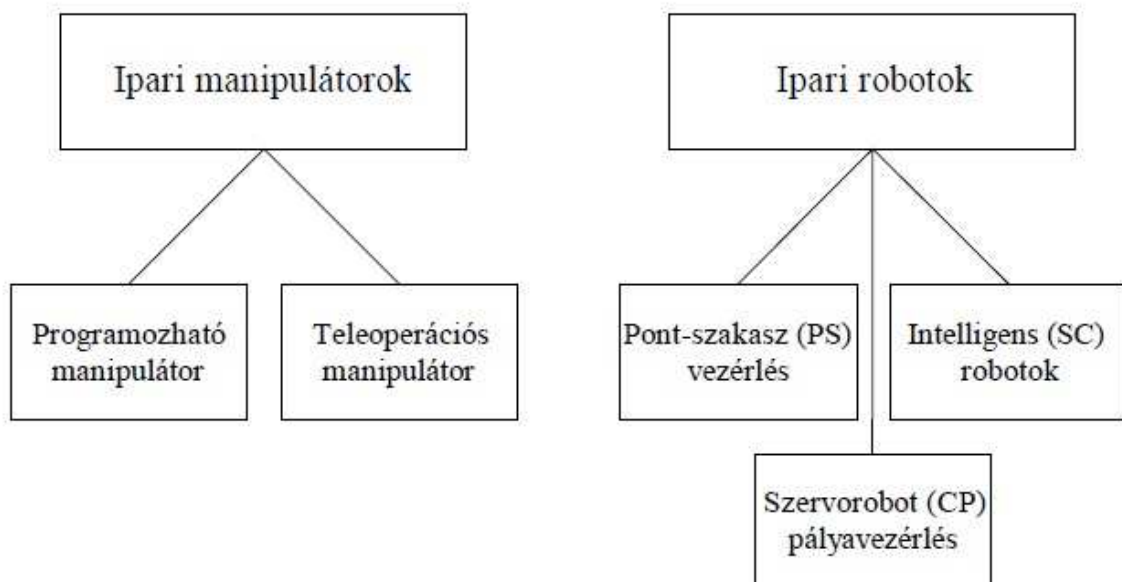
- Döntés egy adott akcióról a mechanikai rendszer és a környezet által jelentett korlátozásokat figyelembe véve (kibernetika, mesterséges intelligencia, szakértői rendszerek, szabályozásmélet)

Ipari robotok

„Az ipari robot univerzálisan állítható többtengelyű mozgó automata, amelynek mozgás egymásutániséga (utak és szögek) szabadon – mechanikus beavatkozás nélkül – programozható és adott esetben szenzorral vezetett, megfogóval, szerszámmal vagy más gyártóeszközzel felszerelhető, anyagkezelési és technológiai feladatra felhasználható” [2].

Ipari robotoknál célszerű a már meglévő ipari komponensek összeintegrálásából építkezni, mivel a robotgyártó és automatizálásra szakosodott cégekkel nem lehet felvenni a versenyt, egyedi hardverekből nem éri meg felépíteni a rendszert kivéve, ha kifejezetten erre van igény. Így fejlesztésre csak kismértékű költséget kell fordítani, az eszköz megbízhatóságát pedig az ipari eszközöket gyártó cég garantálja.

Mobil robot esetében szükséges a nagy sebességű vezeték nélküli hálózati kommunikáció kialakítása. Ilyenkor a robotok rendelkeznek a megfelelő ipari kialakítású WLAN alapú vezeték nélküli kommunikációs eszközökkel. Természetesen ez minden, nem fix helyzetű robotnál érvényes és alkalmazható. Ilyen esetekben sokszor egy WLAN hozzáférési ponttal nem valósítható meg a szükséges terület lefedése, így a WLAN AP-ket cellaszerűen kell elrendezni. Az AP váltásnál egy Rooming műveletet megy végbe és a kommunikáció rövid időre megszakad. Az ipari WLAN eszközök ezt gyorsabban képesek végrehajtani.



1. ábra. Robotok csoportosítása irányításuk szerint

Ipari manipulátorok („buta” vezérlés és kevés szabadság)

Programozható manipulátor:

Megfogó szerkezettel van felszerelve, egyszerű és fix programjával a vezérlés egyetlen mozgásciklus végrehajtására alkalmas, például pneumatikusan működő és a huzalozott logikára épülő vezérlések. Programozásuk kényelmetlen.

Teleoperációs manipulátor:

Itt kézi irányításról beszélünk, melynek távirányítását rudazattal, vagy joystick rendszerű erőátvitellel végzik. Ott vetik be, ahol a munkatér az ember számára veszélyes. A kezelő látja a munkatérrel, egy figyelőablakkal ellátott elválasztott részben dolgozik.

Ipari robotok („okos” vezérlés és szabadon programozható)

Pont-szakaszvezérlésű robot:

A célpontok adva vannak, a robot pályáit, annak felépítése határozza meg. A pozíciót koordináták szerint helyzetérzékeléssel számítják ki. PLC szinten vezérelhető. A robotkéz helyzetét csak kevés pontban definiálja.

Pályavezérlésű, vagy szervorobot:

Pályáját egy program határozza meg, koordinátáinként külön útmérővel rendelkezik. A robotfunkciók és pályák számjegyesen programozhatók (CNC típusú vezérlő).

Intelligens robot:

Szenzorokkal tájékozódik, pályavezérlést használ és magas szinten programozott.

Robotok a harcmezőn

Pilóta nélküli repülő szerkezetek

Egyes szakemberek szerint, a XX. századi háborúkban a pilótáké volt a főszerep, de lehet, hogy a XXI. század már a pilóta nélküli és robotrepülőgépeké lesz.

A jövő légi közelharcai folyhatnak majd ember és ember, esetleg ember és robot, talán robot és robot között.

A pilóta nélküli repülőgép (Unmanned Aerial Vehicle - UAV²), elsősorban katonai feladatokra alkalmazott repülőeszköz, amely valamilyen ön- vagy távirányítással, de leggyakrabban a kettő kombinációjával rendelkezik, ezért fedélzetén nincsen szükség pilótára. „Olyan repülő szerkezetek, amelyeknek a levegőben történő vezetése (irányítása) az ember által közvetve - az irányításra szolgáló berendezés segítségével - történik” [3].

Ennél a fogalomnál valamivel részletesebb az USA Védelmi Minisztériuma által használt meghatározás, mely szerint az UAV hajtóművel rendelkező légi jármű, mely nem hordoz kezelőt a fedélzetén, emelkedéséhez aerodinamikai erőket használ, repülését ön-, vagy távvezérléssel valósítja meg, többször felhasználható, halálos, vagy nem halálos terhet hordozhat.

Az UAV-k koncepciója először az amerikai polgárháború idején merült fel, de ekkor még „csak” légballonokkal szállított robbanóanyagok tömkelegével próbálták megsemmisíteni a másik fél lőszer raktárait. A japánok gyújtóbombákat szállító ballonokat juttattak az ellenséges területek fölé. Könnyű elképzelni, hogy ezek gyakorlati megvalósítása nem volt túl hatékony, az egyre változó légmozgások és egyéb természeti fenyegetésekkel szemben. A mai UAV-n lévő szenzorok már bármilyen időjárási körülménynél alkalmazhatóak, függetlenül a természeti adottságoktól.

² Katonai lexikon 457. oldal

A pilóta jelenléte nélkül 30-40%-os súlycsökkenés érhető el, hiszen nem kell katapult és a hozzá tartozó járulékaik, valamint a pilóta köré épült védelmi rendszerre sincs szükség. Valamennyi vezérlő egysége ott van, ahol a pilóta ülne, így a személyzet hiányának minden előnyét kihasználja.

Az első, rendszerbe állított robotrepülőgép a német V-1 volt, mely a mai robotrepülőgépek szinte minden ismertetőjegyével rendelkezett: sugárhajtómű, hengeres törzs, egyenes szárnyak, a célra vezérlést pedig tehetetlenségi irányítórendszer végezte. Elsősorban szárazföldről indították, de létezett levegőből és tengerről indítható fegyver is.

A robotrepülőgépek és konstrukcióik folyamatos fejlesztésen mennek keresztül, hatótávolságuk növelése, és levegőben tarthatóságuk érdekében. A katonai alkalmazásuk mellett, fontos szerepet játszanak a polgári életben is, mint például:

- Mezőgazdaság
- Környezetvédelem, meteorológia
- Földtani és tengerfenék kutatások
- Határőrség, partiőrség
- Telekommunikáció, híradás
- Tűzszerészeti felderítés
- Légi, földi forgalomirányítás

Olyan feladatokra alkalmazzák ezeket az eszközöket, amelyek túl veszélyesek ahhoz, hogy emberek életét kockáztassák, vagy túl sokáig (esetleg több napig is) eltart egy küldetés, így csak több pilóta lenne képes teljesíteni. Ilyen a felderítés, megfigyelés, tűzérségi tűz helyesbítése, célmegjelölés irányított fegyvereknek, rádiótechnikai átjátszás, rádióelektronikai zavarás és célrepülőgép feladatkörök.

Lehet passzív felderítő vagy felfegyverzett támadó eszköz, ami egy harci gépnél komoly előnyt jelent.



2. ábra. A Kettering Bug pilóta nélküli repülőgép 1930-as évek

A mai UAV-k beprogramozásuk után általában teljesen önirányításúak, tudják mi a feladatuk, képesek önállóan felszállni, a küldetést teljesíteni és leszállás után visszatérni a megadott helyre.

A vadászgépektől átvették a fegyver nélküli felderítő szerepét, mert sokkal kevesebb fegyvert vihetek magukkal, de ez nagyobb sebezhetőséget is jelentett egy ellenséges géppel szemben. Gyorsaságuk miatt, a felderítésnél nem érzékelték az álcázott célpontokat, ezt más műszaki csodák mellett a szintetikus képalkotó radarral küszöbölték ki, melynek kamerái rossz időben, kevés fénynél, de még a tetőkön is átlátnak.

Napjaink egyik korszerű légi járműve a Predator nevet kapta. 2001. szeptember 11-e óta feladata a terroristák felkutatása, megfigyelése. A hegyvidéki területeken megbúvó terrositák barlangokban élnek és így nehezen felderíthető célpontokat jelentenek a hadvezetés számára. A Predator a céltárgy azonosításában és bemérésében precíziós lézeres célmegjelölő berendezést használ, így a taktikai vadászgép lézervezérlésű fegyvert juttathat a céltárgyra nem kell cirkálnia.



3. ábra. Predator

Az új család az UCAV (Unmanned Combat Aerial Vehicle), itt már nincs szükség a távirányító pilótára. Ezeket a kezdetektől nagy túlélő képességű fegyverhordozónak tervezték, így a korszerű komputertechnika révén lényegesen fejlettebb, mint a Predator típusú eszközök. A gép az ellenséges támadást önálló védelmi manőverrel kerüli ki, ezért függetlenebb és jobb túlélő, mint más pilóta nélküli gépek. Egy UCAV teljesen automatikusan működik, harcközben figyeli a hadműveleti területet és a helyzetnek megfelelően módosítja az eredeti feladatkört. Akár 20-30 G-re is képesek, ami egy vadászgépnél a legjobb pilótával is maximum 8-9-szeres gravitációs terhelés lehet, emiatt az UCAV olyan manővereket tud végrehajtani, ami egy ember által vezetett repülőeszközzel lehetetlen lenne.

Olyan esetek is vannak, mikor már nem csak a bázissal kommunikálnak, hanem egymás között is cserélhetnek információt, reagálhatnak a harci környezetre és változtathatnak a célokon, illetve ezt egyenként is megtehetik.

Attól függően, hogy milyen feladatot látnak el kétféle csoportba sorolhatók, nem halálos (non-lethal) és halálos (lethal). Nem halálos UAV-k képesek a terepen elhelyezett különböző

típusú aknák felderítésére, adatok átjátszására, különböző pusztító eszközök számára cél (infra) megjelölésre.

Feladatcsoportok:

- hírszerzés, felderítés, megfigyelés, harcmező kárfelmérés
- elektronikai hadviselés
- az ellenséges légvédelem elnyomása
- álló célok támadása
- mozgó célok támadása
- ballisztikus rakéta és robotrepülőgépek elleni védelem
- légvédelem
- közvetlen légi támogatás
- légi fölény kivívása.

Fontos a megfelelő rejtettség, melynek érdekében az alábbi tényezőket kell figyelembe venni és csökkenteni, ha szükséges:

- az UAV geometriai mérete
- hatásos visszaverő felülete
- zajkibocsátási tényezői
- irányítottá kell tenni a kétoldalú kommunikációs kapcsolatot.

Elérhető, hogy csökkenjen a szemben álló fél vizuális, rádiólokációs, rádióelektronikai és hangfelderítési képessége, így jóval később tud reagálni. A többszöri felhasználhatóság érdekében fontos, hogy a légi jármű biztonságosan visszatérjen a bázisra, erre az eszköz bonyolultságától függően más és más elképzelések és gyakorlati megoldások léteznek, kezdve az egyszerűbbektől (elfogás hálóval, földet érés ejtőernyővel) a műszakilag bonyolultabb kivitelezésekig (automatikus műszeres leszállító rendszer).

Kommunikációs képesség

A szakértők egybehangzóan azt állítják, hogy a jövőben a háborúk a hálózat alapú hadviselésen alapulnak majd, melyet a digitális harcmező és az azon folyó műveletek összetettsége jellemez. Minden résztvevőnek (beosztástól, pozíciótól, feladattól függetlenül)

ismernie kell a körülményeket, rendelkeznie kell a döntését befolyásoló lényeges információkkal, melyek egy védett zárt hálózatba kerülnek, és ebből érhetőek el, megfelelő jogosultságok kiosztásával. Fontos, hogy ezek az adatok a lehető legpontosabbak és legfrissebbek legyenek, valamint valós időben jussanak el a felhasználói oldalra.

Hibák

A harcászati Pilóta nélküli gépek bár rendkívül hatékonyak, nem hibátlanok, feladataikat bizonyos korlátozásokkal képesek végrehajtani. Néhány probléma, melyek limitáló faktorok az UAV-k felhasználásában:

- sebezhetőség az ellenséges tűztől
- időjárási körülmények (borultság, erős szél/turbulencia, jegesedés, zivatar tevékenység, stb.)
- földi irányító állomástól való függőség (korlátozott hatótávolság a láthatóság határain belül)
- korlátozott frekvencia használat
- irányítási és vezetési információk kisugárzása
- korlátozott szenzor képességek (hatótávolság, felbontóképesség)
- csekély túlélőképesség
- extrém hőmérsékleti különbségek (elsősorban az akkumulátorral meghajtott UAV-nál)

A jövő háborúi bizonyára kiterjednek majd az űrre is, de már napjainkban is létezik olyan haditechnikai fejlesztés, melynek eszközrendszere itt, vagy innen fejti ki hatását.

A hadszíntereken megjelennek majd a robotok, majd az intelligens robotok, melyek egyaránt ellátnak támadó és védekező szerepet is.

Robotok a Marson

Jelen pillanatban is folyik a munka a vörös bolygón. Megismerésének nagy kérdése, hogy robotok vizsgálják, amelyek feljuttatása bár olcsóbb, mégis kevesebb ismeretet nyújtanak, vagy költségesebb, emberi expedíciók induljanak. Mindkét oldalnak megvannak a maga előnyei és hátrányai.

A Mars északi sarkvidékén szállt le a Phoenix-űrszonda, de már több probléma jelentkezett, amely emberi jelenléttel könnyen kiküszöbölhető lenne. A Föld körüli térségben jelenleg ideálisan oszlik meg a munka a robotok (azaz a műholdak) és az emberek (a Nemzetközi Űrállomás legénysége) között. A hidegháború sürgette holdutazásoknál született tudományos eredmények többsége jóval kisebb ráfordítással, robotokkal is elérhető lett volna, és alaposabban megtervezett programmal, most sokkal több információval lehetne dolgozni.

Az űrben végzett munka hatékonyságának a kommunikáció időbeli késése szab határt, mivel nem lehet azonnal érzékelni mit csinál és lát a szonda, ezért sok feladatot teljesen önállóan kell végrehajtania [4].

Előnyök, hátrányok

A robotokkal végzett kutatás legfontosabb előnyei

- Viszonylag alacsony költségek: az űrszondákkal végzett Mars kutatás olcsóbb, mint az emberes expedíció, mivel mostohább körülményeket is kibírnak és nem kell visszatérniük a Földre.
- Technológiai tapasztalatok: űrszondás vizsgálatok már évtizedek óta zajlanak. A hibákat, sikereket, kudarokat figyelembe véve fejlesztik az újabbakat a mai szintre.
- Csekély biztonsági igény: egy Mars-szonda elvesztése nem jár egyetlen kutató halálával sem.
- Mesterséges intelligencia: segítségével már a közeljövőben nagymértékben szélesedhetnek a robotok képességei.
- Extrém tűrőképesség: olyan körülmények között (hideg, erős sugárzás, levegőhiány, mérgező kémiai környezet stb.) is dolgozhat, ahol ember már nem tud.

- Látni a láthatatlant: a látható tartomány mellett egyéb hullámhosszakot, lézeres vagy radaros méréseket is használhat.
- Rendkívüli memória: nagy mennyiségű, előre betáplált adatot (terület felépítése, koordináták, földtani képződmények helyzete stb.) képes hibátlanul felidézni, és gyorsan felhasználni a munka keretében.

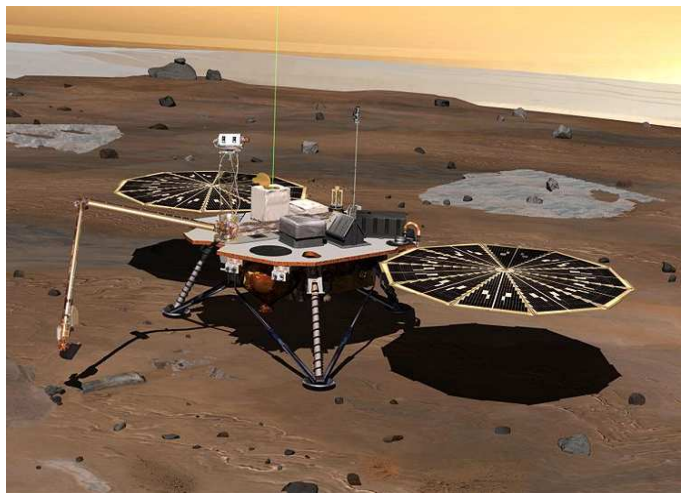
Nyilván ezeket a hátrányokat az emberekkel végzett munkához kell viszonyítani, így meg kell néznünk a másik oldalt is.

A robotokkal végzett kutatás legfontosabb hátrányai

- Önálló ötletek hiánya: nagyrészt előre programozott feladatokat hajtanak végre, ezért az új viszonyokhoz nehezebben alkalmazkodnak, mint az emberek.
- Műszerkészítés hiánya: csak az előre elkészített műszerekkel képes méréseket végezni, nem képes a detektorait helyi igényekhez igazítani.
- Általános rugalmatlanság: ez tartják a legnagyobb hátránynak az emberes expedíciókkal szemben. Éveken át fejlesztett programokat használ, mégis korlátozottan tud a környezethez alkalmazkodni, ellentétben az ember lehet, hogy ugyanazt a problémát másodpercek alatt könnyen megoldja.
- Lassúság: terepen való haladást csak lassan tudják elvégezni.



4. ábra. Spirit



5. ábra. Phoenix

Szenzorok

A korszerű robotok nyílt rendszerek, amelyek kommunikálni tudnak egymással, információt tudnak cserélni állapotukról és fogadni tudják az összetett feladatok koordinált megoldásához szükséges parancsokat, ha a feladatok több közvetítő együttműködésével oldhatók meg. Ezeket az információkat szenzorokkal érzékelik, amik az irányításhoz is szükségesek [5].

A szenzorok alapvető feladata a környezeti információk fogadása és továbbítása a központi rendszer felé.

Típusai:

- Passzív
 - a környezet fizikai jellemzőit érzékelik, pl.: hőmérséklet, nyomás, fény, hang
- Aktív
 - működésük során maguk bocsátanak ki mérő jelet, mely visszaverődését érzékelik, pl.: radarok, megvilágítást igénylő fényérzékelők, kamerák, ultrahangos, lézeres távolságmérők

Telekommunikáció

A telekommunikáció alatt a jelek továbbítását értjük adó és vevő között, valamilyen átviteli közegen, csatornán. A definíció részének gondoljuk azt is, hogy mindez nagy távolságon történik, és a kommunikáció elősegítése a cél [6].

Az adó kódolja, az átviteli közeg viszi, a vevő pedig dekódolja a jelet. A telekommunikáció lehet egyirányú (televízió, rádió) vagy kétirányú (telefon, számítógépes hálózat).

E hálózatok legfontosabb elemei: backbone, hub, router, host, terminal.

Az információt az elektronikus kommunikáció céljából ezekre ki talált eszközökkel elektromos jellé kell alakítani. Az elektromos, elektromágneses jeleket nagy távolságra továbbíthatjuk (Voice-over-IP (VoIP) vagy Session Initiation Protocol (SIP) protokoll).

Információ formailag lehet adat, szöveg, hang, kép. Ezekkel műveletek végezhetők, mint például létrehozás/gyűjtés, kezelés/feldolgozás, tárolás, közlés/átvitel. Az információ csere kapcsolat maga a telekommunikáció.

Hálózat típusok, csoportosítás

A méret vagy távolság meghatározza a hálózatban alkalmazott technikát, amely lehet vezetékes vagy vezeték nélküli. Két vagy több hálózat összekapcsolását internetwork-nek nevezzük, legismertebb internetwork az Internet.

Átviteli technológiától függően két csoportba oszthatók a hálózatok.

Üzenetszórásos hálózat: a kommunikáció úgy történik, hogy a csomópontok a minden más csomópont által küldött csomagokat vagy kereteket is megkapják. A keretben lévő cím határozza meg a címzettet vagy címzetteket, és csak a címzett csomópontok dolgozzák fel a keretet.

Pont-pont kapcsolat: csomópontpárok közötti kapcsolatok sokaságából áll. Itt egy adott küldőtől egy adott címzethez küldött csomagot vagy keretet esetleg más csomópontokon keresztül kell továbbítani. Ezek tárolják és továbbítják azt, amíg el nem éri célját.

Fizikai terület nagysága alapján is meg kell különböztetni a hálózatokat.

Személyes hálózati környezetek (Personal Area Network . PAN) egy személy igényeit elégítik ki.

Helyi hálózatok (Local Area Network . LAN) egy korlátozott méretű területet fednek le, tipikusan magánszemélyek, részlegek, kisebb otthoni szervezetek vagy épületek adott emeletei birtokolják őket. Az átviteli technológiák sebessége 10 Mbps-tól 10 Gbps sebességűig terjed.

Városi hálózatok (Metropolitan Area Network . MAN) nagyobb területet hidalnak át, például egy várost.

Nagy kiterjedésű hálózatok (Wide Area Network . WAN) nagy földrajzi területet fednek le, egy államot, országot vagy akár egy kontinenst is. Egy WAN, alhálózatokban összekapcsolt hosztokból (kliensekből és szerverekből) áll. Üzenetküldésnél a küldő az üzenetet kisebb részekre, csomagokra osztja.

Vezeték nélküli hálózatok

Rövidtávú hálózatokról akkor beszélünk, ha az eszközök 6 és 10 méteren belül vannak összekapcsolva, például a Bluetooth fejhallgatók, számítógépek, szkennerek, monitorok, billentyűzetek.

Vezeték nélküli LAN számítógépekből, rádió-modemmel és antennával felszerelt hozzáférési pontokból áll, melyek küldik és fogadják az adatokat. A fizikai terület kb. 100 méter. Vezeték nélküli WAN kis és nagy sávszélességű hálózat is lehet. Digitális hangot és adatokat is továbbít legfeljebb 2 Mbps sebességgel. Ha épületekben használják, akkor a távolság több kilométer is lehet.

Wi-Fi

Az IEEE³ által kifejlesztett vezeték nélküli mikrohullámú kommunikációt (WLAN) megvalósító, széles körűen elterjedt szabvány (IEEE 802.11), melyet azért találtak ki, hogy kiváltsák a vezetékes hálózatok korlátait, mégis egyenértékű maradjon azzal. Mindegy melyik gyártótól vásárolunk eszközöket, mindegyik működni fog egymással, és kompatibilis lesz. Segítségükkel rádiós adatátviteli összeköttetést tudunk létrehozni, de minden rádiós hálózaton felmerül az adatbiztonság kérdése. A Wi-Fi eszközök tartalmazzák a WEP-et ami a wireless equivalency protocol, egy 40 bites titkosítás, mely a vezetékes hálózatoknál is megszokott biztonságot nyújt. Ezt is lehet növelni, 128 bites vagy nagyobb titkosítást is elérhető.

IEEE szabvány	Megjelenés ideje	Működési frekvencia (GHz)	Sebesség (jellemző) (Mbit/s)	Sebesség (maximális) (Mbit/s)	Hatótávolság beltéren (méter)	Hatótávolság kültéren (méter)
Eredeti 802.11	1997	2,4	0,9	2	~20	~100
802.11a	1999	5	23	54	~35	~120
802.11b	1999	2,4	4,3	11	~38	~140
802.11g	2003	2,4	19	54	~38	~140
802.11n	2008	2,4 / 5	74	248	~70	~250
802.11y	2008 volt	3,7	23	54	~50	~5000

6. ábra. Legelterjedtebb szabványok

- 802.11a: 5 GHz-es frekvenciasávban működő eszközök előnye a nagy távolság és sávszélesség, viszont jellemzően csak pont-pont kapcsolatra használják és az ehhez használható eszközök általában drágábbak. Különösen fontos az optikai rálátás a két pont között.
- 802.11b: 2,4 GHz-es tartományban működő eszközök hatótávolsága a terepviszonyoktól függően széles skálán mozoghat, lényegesen kisebb, mint a 802.11a, pont-multipont kapcsolatoknál 1 km-es sugarú körön belülre szokták tervezni. Átviteli sebessége max. 11 Mbit/s.

³ Institute of Electrical and Engineers, Inc.

- 802.11g: 2,4 GHz-en működő eszközök, a 802.11b-vel sok tekintetben megegyezik, a routerek nagy része mindkettőt támogatja. Előnye, hogy nagyobb sáv szélességet képes átvinni, hátránya pedig, hogy a távolság növekedésével lényegesen romlik a határfoka és érzékenyebb az interferenciára. Átviteli sebessége max. 54 Mbit/s.
- 802.11n: 2,4-5 GHz-en mozog, 248 Mbit/s maximális átviteli sebességű vezeték nélküli kapcsolat.

Kialakítása lehet publikus, nyílt hálózat, ami bármely wi-fi routerrel kialakítható, az így létrehozott hálózathoz bárki csatlakozhat, mindenféle korlátozás nélkül.

A privát hálózat saját felhasználásra lett kialakítva, titkos jelszó védi.

Publikus, zárt hálózatonál egy speciális szoftver gondoskodik arról, hogy a hálózatot csak egy kód ismeretében, korlátozott ideig lehessen használni. Ezt a formát rendszerint éttermek, kávézók használják, ahol az internet elérés fogyasztáshoz van kötve.

Kereskedelmi HotSpot szolgáltatás csak díjfizetés ellenében, korlátozott ideig használható [7].

A feladat

A kezdeti feladatom egy négykerekű robot és a bázis közötti kommunikáció megvalósítása volt, amely vezeték nélküli Wi-Fi csatlakozással van megoldva. Szerencsésnek érzem magam, mert közös megegyezés alapján, ez egy hozzám közelebb álló feladatra módosult amely hasonló, csak a platform változott. Az új projektben fizikailag robot nem szerepel, viszont ezt modellezve, úgy utalok rá, mintha tényleges résztvevője lenne a kapcsolatnak. Fontos dolog, hogy így eszköz független lett az egész művelet, így a későbbiekben, tetszés szerint folytatható a már kiépített ösvény. A projekt lényege maga a kapcsolat megvalósítása, felépítése, hogy egy robot (ez esetben például egy felderítő robotról is beszélhetünk) és egy felhasználó miként tud kommunikálni egymással, és milyen adatokat küldenek / fogadnak. A robot oldalon egy LabVIEW-ban elkészített program fut. Nincsenek konkrétan rögzítve az elküldendő parancsok és a visszaérkező üzenetek. Jelentéseket küld, amelyeket egy rendszer rögzít, onnan bármikor visszaolvasható, illetve a robotnak parancsok küldhetők amit ő sorrendben végrehajt miután a servertől lekérdezte.

Mint ezt látni fogjuk, a feladat hatékonyabb megvalósítása érdekében az egyes részekben más-más programnyelvet, szoftvert, és servert használtam. Ezeket szeretném röviden bemutatni.

Felhasznált programcsomagok

- Xampplite 1.7.2
- Apache
- MySql
- Labview 8.5
- HTML
- PHP
- phpDesigner7

Apache

A robot és a kezelője két weblapon keresztül is kommunikál egymással, így egy servert használtam, mivel böngészőben nyitom meg az űrlapokat, így nem elég csak a PHP-t telepíteni a gépre, mert böngészőben nem lennének képesek közvetlenül megtekinteni a scriptek működését, mert az elsősorban egy web kiszolgálóval szeretne kommunikálni. Ahhoz, hogy használható környezetet építsünk ki, mindenképpen szükség van egy web kiszolgálóra is, így az Apache-ra esett a választás. Nyílt forráskódú, webkiszolgáló alkalmazás, ez van jelen több internetes szerveren is, ingyenes, de a bérelt tárhelyek nagy része is ezt telepíti (a Google keresőmotor, és a Wikipédia elég jó referenciának számít). Kézenfekvő megoldás ha PHP-t, MySQL-t is használunk. Többek között a következő operációs rendszerekhez készítették el az Apache-ot: Unix, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X és Microsoft Windows. Tetszőleges fájlok megosztására is használható. Számunkra elegendő az alap modul, már ezzel is elég sok mindent el lehet érni.

MySQL

Az MySQL az egyik legelterjedtebb adatbázis-kezelő, több felhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő. Ingyenes, és teljesen nyílt forráskódú LAMP (Linux–Apache–MySQL–PHP) összeállítás részeként költséghatékony és egyszerűen beállítható megoldást ad dinamikus web helyek szolgáltatására.

Egyedi illesztő felületekkel az adatbázis-kezelő elérhető C, C++, C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. A MySQL honlapjáról grafikus felületű adminisztráló eszközök is letölthetők: MySQL Administrator és MySQL Query Browser.

A PHP nyelven írt, phpMyAdmin ami a feladatban valósítja meg az adatbáziskezelést nyitott forráskódú és széles körben elterjedt, népszerű adminisztrációs eszköz.

A MySQL 5.x képességei:

- ANSI SQL 99, számos kiegészítéssel
- Keresztplatformos elérhetőség
- Tárolt eljárások
- Adatbázis triggerek
- Kurzor adatbázisok
- "View" adatbázisok
- Valódi VARCHAR támogatás
- INFORMATION_SCHEMA támogatás
- "Strict" (szigorú) mód

SSL támogatás (Secure Socket Layer, Biztonsági Alréteg)

- Lekérdezés gyorsítár (cache)
- Egymásba ágyazott SELECT –ek
- Szöveges indexelés és keresés a MyISAM motorral
- Beágyazott adatbázis-könyvtár
- Részleges UNICODE támogatás
- ACID (Atomicity, Consistency, Isolation, Durability) megfelelés az InnoDB-vel, BDB-vel és Cluster-rel

- Továbbfejlesztett MySQL Cluster
- "Példányosítás"

A feladatban szereplő adatbázisban három tábla készült, mind más feladatokat látnak el. Az adatbázisokról általánosságban elmondható, hogy adatok tárolására alkalmas alkalmazások (számunkra szerverprogramok), amelyek belső algoritmusai, optimalizált funkcióik segítségével hatékonyan és gyorsan tudnak nagyobb adatmennyiséggel is dolgozni. Az adatokat táblázatokban tárolják, melyek logikusan sorokból és oszlopokból állnak. A tábla sora az adatrekord, amely több különböző mezőt (cellát) tartalmazhat, ezek az adattábla oszlopai. Egy elemi mező (cella) több jellemezővel is leírható, például nevével, értékével, típusával. A relációs adatbázisokkal a táblák között relációs kapcsolatokat lehet létrehozni, amelyek alapján a különálló adattáblákat logikailag egymáshoz lehet kapcsolni.

HTML

HTML⁴ oldalt valószínűleg mindenki látott már. Ez egy leíró nyelv, melyet weboldalaknak az elkészítéséhez fejlesztettek ki, és mostanra internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

HTML kódok általában szöveges állományokban találhatóak meg az internethez kapcsolódó számítógépeken. Ezek az állományok tartalmazzák azokat a szimbólumokat, amelyek a megjelenítő programnak leírják, hogyan is kell megjeleníteni illetve feldolgozni az adott állomány tartalmát. Megjelenítő programnak a Mozilla Firefox 3.5.5 verziójú böngészőt választottam.

A HTML állomány felépítése viszonylag egyszerű, három fő részre bontható:

- a Dokumentum Típus Definíció az állomány legelején, ami a használni kívánt DTD-t adja meg, pl:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

⁴ HyperText Markup Language = hiperszöveges jelölőnyelv

- a HTML fejléc <head>, ami technikai és dokumentációs adatokat tartalmaz, melyeket az internet böngésző nem jelenít meg, tehát átlag felhasználó ezeket nem látja
- és a HTML törzs <body>, amely a megjelenítendő információkat tartalmazza.

PHP

A PHP⁵ egy számítógépes szkript nyelv, természetesen ez is mint az előzőek nyílt forráskódú. Legfőbb felhasználási területe a dinamikus weboldalak készítése. A PHP-t főként szerveroldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is készíthetők vele. Esetünkben az Apache serveren futnak a PHP-scriptek, és a táblák, űrlapok közötti kommunikációt valósítják meg.

Szinte minden webserverre, operációs rendszerre és platformra ingyenesen telepíthető. Tehát amikor egy PHP-ben megírt oldalt akarunk elérni, a kiszolgáló először feldolgozza a PHP utasításokat és csak a kész HTML kimenetet küldi el a böngészőnek, így a programkód nem látható kliens oldalról. Ehhez egy értelmezőt (interpretert) használ, amely általában egy külső modulja a webservernek.

A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek (vagy ha igen, korlátozottan). Ilyen például a bejelentkezés, az adatbáziskezelés, filekezelés, e-mail küldése, adatfeldolgozás, stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell végrehajtani (például képek listázása és linkelése, listakészítés stb.), ott ez a programnyelv nagyszerű segítség.

LabVIEW

A National Instruments köztudottan virtuális műszerezéssel foglalkozik, és piacvezető a területen. Alapjaiban megváltoztatta a tudósok és mérnökök megközelítési módját a mérésadatgyűjtéshez és automatizáláshoz. A LabView grafikus fejlesztői környezet tökéletes választás azoknak, akik költségtakarékosan akarnak olyan feladatot elvégezni, melyek

⁵ Hypertext Preprocessor

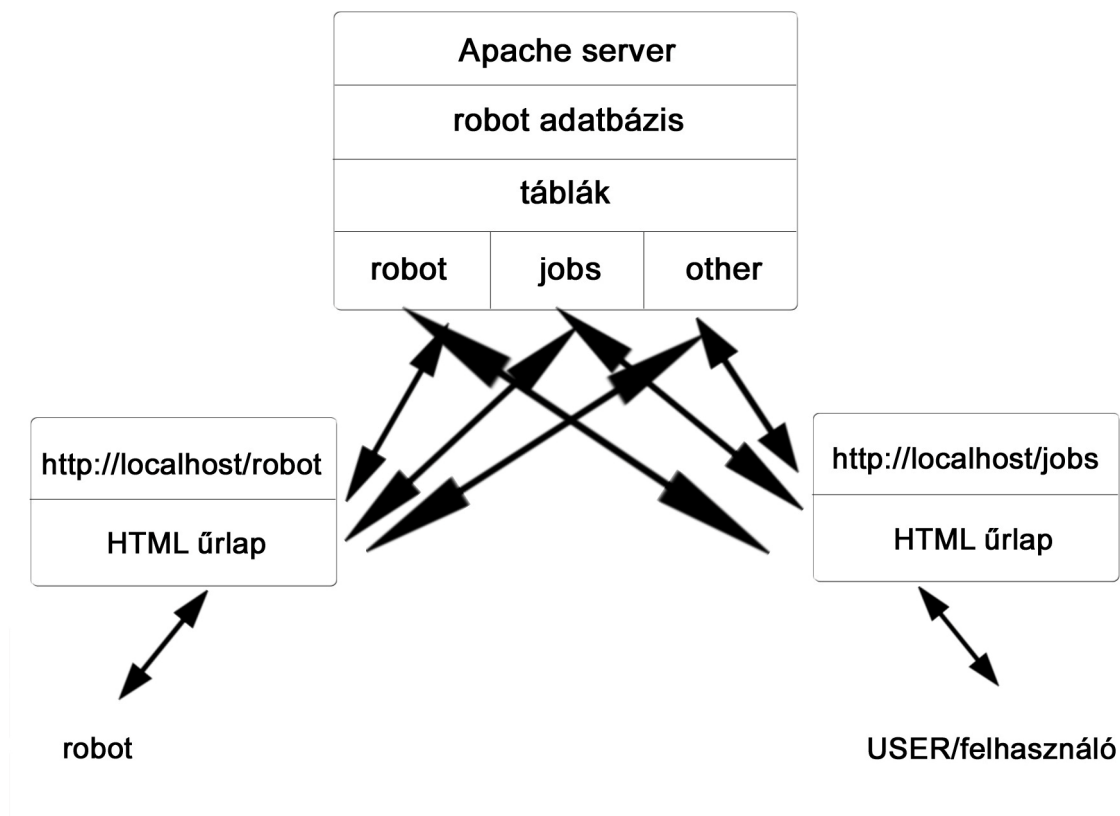
körülményeit a valóságban például nehezen, drágán, vagy egyáltalán nem is lehetne megvalósítani. Ehhez lehet kihasználni a PC-ket és a hozzájuk tartozó technológiák előnyeit.

A LabVIEW programokat virtuális műszereknek, vagy rövidebben VI – oknak nevezzük. Megjelenésükben és működésükben fizikai műszereket utánoznak és nagyon széles eszközkészlettel rendelkeznek az adatgyűjtés, adatelemzés, megjelenítés és adattárolás területén. A VI-ok három részből állnak: az előlapból (front panel), blokk diagramból (block diagram) és az ikon és csatlakozó panelből (icon and connector pane).

LabVIEW-ban kontrolok (control) és indikátorok (indicator) segítségével készíthetjük el a felhasználói interfészt vagy más néven az előlapot. A kontrolok közé tartoznak a forgatógombok (knob), a nyomógombok (push button), a tárcsák (dial) és más input eszközök. Az indikátorok pedig a grafikonok (graph), LED-ek és egyéb megjelenítők. A felhasználói interfész elkészítése után VI-ok és struktúrák felhasználásával ahhoz egy kódot rendelhetünk hogy az előlapi objektumokat vezéreljük. A blokk diagram ezt a kódot tartalmazza. A LabVIEW tartalmaz beépített eszközöket az alkalmazások Web-hez történő kapcsolódására, ami elengedhetetlen a feladatban. Saját Web Szervere is van és támogatja a szoftver szabványokat, például TCP/IP hálózat és az ActiveX vezérlő [8].

A megvalósítás

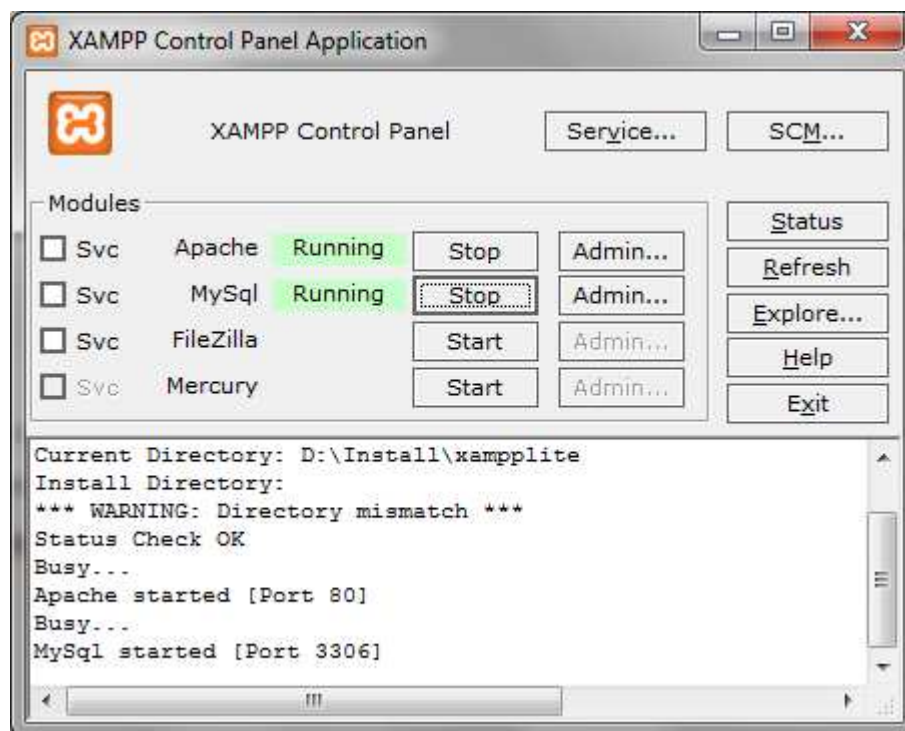
Ahhoz, hogy a robot, és a neki utasításokat adó felhasználó kommunikálni tudjanak, valamilyen kapcsolatot kell létesíteni. Ez úgy van megvalósítva, hogy a roboton egy LabVIEW program fut, amely küldi az információkat a külön neki készített úrlapra, és onnan PHP script továbbítja az adatokat egy adatbázisban létrehozott adattáblába. Ezzel egy időben, ha kapott új parancsokat, akkor azt feldolgozza. Egy egyszerű ábrával tanulmányozható, hogy miről is van szó.



7. ábra. Feladat vázlata

Közös oldal

Az apache webserver és a MySQL adatbáziskezelő kifogástalan működését egy külön ilyen feladatokra kialakított programcsomag segíti. Ez a xampp_1.7.2 verzió. A feltelepítés után futtatva a ..\xampplite\xampp_contol.exe -t, nagyon egyszerűen el lehet indítani a szükséges modulokat.



8. ábra. Xampp

Apache started [Port 80]

MySQL started [Port 3306]

Látható, hogy melyik porton tudjuk elérni az alkalmazásokat, ezekre szükségünk van mindkét oldalon.

Űrlapok elkészítése, elérése

Robot oldal

Böngészőben, nálam a már említett Mozilla Firefox címsorában a localhost-ot, vagy a 127.0.0.1-et kiadva megkapjuk az Apache üdvözlő oldalát és értesítést kapunk a sikeres telepítésről, működésről. (Bármilyen más beállíthatunk indítóoldallá, ha megváltoztatjuk az elérési utat az adott helyen.)

Az űrlapokat HTML kóddal írtam, PhpDesigner7-program segítségével. (Tetszés szerint, akár egy egyszerű szöveges állományban is meg lehet írni a kódokat, nekem fontos volt, hogy lássam az összes php, és HTML fájlt egyszerre.)

Két weblapra van szükség, az egyiket a robot, a másikat pedig a felhasználó fogja közvetlenül elérni. Természetesen, aki a parancsokat adja a robotnak, mindkettőt láthatja.

Az első legyen a robot oldali, „robot” paraméterrel lehet majd elérni. A kódból kiemelek pár fontosabb dolgot, és rövidem magyarázom:

```
..\xampplite\htdocs\robot\index.htm
```

```
<html> ... </html>      közé kerül az egész kód
<head> ... </head>      közé kerül a fejléc
<body> ... </body>      közé kerül a megjelenítendő kód
```

```
//Meg van adva egy action, ami ténylegesen elhelyezi az adatbázisban a bevitt adatokat
```

```
<form action="feldolgoz.php" method="post">
```

```
// <tr> és <td> a sorokat és oszlopokat kezelik, közé kerül a név ami megjelenik, és a változó
// amiben tárolva lesz.
```

```
<tr>
  <td>Idő (hmmss):</td>
  <td><input type="text" name="ido" /></td>
</tr>
<tr>
  <td>GPS szélesség:</td>
  <td><input type="text" name="gps_szel" /></td>
  <td>GPS hosszúság</td>
  <td><input type="text" name="gps_hossz" /></td>
</tr>
```

```
// Látási viszonyokhoz legördülő menüből választható két érték.
```

```
<tr>
  <td>Látási viszonyok:</td>
  <td><select name="latasi_viszonyok">
    <option value="Jó">Jó</option>
    <option value="Rossz">Rossz</option>
```

```

        </select></td>
        <td colspan="2"></td>
    </tr>

```

// Egy gomb lenyomásával lehet feltölteni az adatbázisba a bevitt adatokat

```

<tr>
    <td colspan="4" align="center"><input type="submit" value="Adatok felvitele"
        name="gomb" /></td>
</tr>

```

A feldolgoz.php actionként szerepel az index.htm-ben, nézzük hogyan épül fel

<?php .. ?> között szerepel a php kód

// meg kell adnunk, hogy milyen felhasználó névvel, jelszóval csatlakozzon a serverhez
include 'dbkapcs.php';

A dbkapcs.php mindössze ennyiből áll:

```

<?php
    $host = 'localhost';
    $username = 'root';
    $password = "";
    $database = 'robot';
    $kapcsolat = mysql_connect($host, $username, $password);
    mysql_select_db($database);
?>

```

// A következő parancsok változóban tárolják a szükséges információkat SQL utasítások
 // segítségével. A később felépített táblákat is rendezi. A robot kap egy feladatot, és azt futási
 // időben teljesíti. Arra szolgálnak a sorok, hogy ha a felhasználó közben több feladatot is
 // küld, akkor a robot, munkája végeztével minden el nem végzett sorszámú feladatot
 // kilistázzon, ne csak az utolsót.

```

$get_last_get_query = mysql_fetch_array(mysql_query("SELECT * FROM other"));
$get_last_get = $get_last_get_query['last_get'];
$job_query = mysql_query("SELECT * FROM jobs WHERE feladat_id > $get_last_get");
$job_query = mysql_fetch_assoc(mysql_query("SELECT * FROM jobs ORDER BY id
DESC"));
$last_get = $job_query['feladat_id'];
mysql_query("UPDATE other SET last_get = '$last_get'");

// Megfelelteti a változókat az értékekkel, és feltölti a „robot” adattáblába, ezután lekéri a
// még el nem végzett feladatok listáját, és kiírja.
if (isset($_REQUEST['gomb'])) {
    $ido = $_REQUEST['ido'];
    $gps_szel = $_REQUEST['gps_szel'];
    $gps_hossz = $_REQUEST['gps_hossz'];
    $mitcsinalok = $_REQUEST['mitcsinalok'];
    $kep_kamera = $_REQUEST['kep_kamera'];
    $homerseklet = $_REQUEST['homerseklet'];
    $latasi viszonyok = $_REQUEST['latasi viszonyok'];

    mysql_query("INSERT INTO robot (ido, gps_szel, gps_hossz,
        mitcsinalok, kep_kamera, homerseklet, latasi viszonyok) VALUES ('$ido',
        '$gps_szel', '$gps_hossz', '$mitcsinalok', '$kep_kamera',
        '$homerseklet', '$latasi viszonyok');");

    echo "OK<br />Feladat ID: " . $job_query['feladat_id'] . "<br />" . $job_query['feladat'];
    while ($row = mysql_fetch_array($job_query)) {
        echo "<" . $row['feladat'] . ">";
    }
}
else echo 'ERROR';
?>

```

Böngészőben a <http://localhost/robot/> címen a következő űrlapot hoztam létre a bemutatott kód kiegészítésével. Ez az index.htm-et keresi, és jeleníti meg. A robot használja, adatai feltöltésére.

Idő (hhmmss):	<input type="text"/>	
GPS szélesség:	<input type="text"/>	GPS hosszúság <input type="text"/>
Mit csinálok:	<input type="text"/>	
Kép / kamera:	<input type="text"/>	
Hőmérséklet:	<input type="text"/> C fok	
Látási viszonyok:	<input type="text" value="Jó"/>	
	<input type="button" value="Jó"/> <input type="button" value="Rossz"/>	<input type="button" value="Adatok felvitele"/>

9. ábra. Robot weblap

Felhasználó oldal

A felhasználó meg tudja nézni a robot oldali adatbázist is, de ha ez nem elég, vagy esetleg nem megvalósítható, akkor a robot elküldi neki, hogy éppen mit csinál. Ezt is egy űrlapra tölti fel. Egy képet is küld, amely a weblapon megjelenik, és egy Microsoft Office Excel fájlt is, az oszlopokban pedig az elvégzett feladat. Rendre aktuális idő, GPS szélességi és hosszúsági fok, mit csinál, kép neve, hőmérséklet, látási viszonyok. Az űrlap ugye HTML kód alapján jelenít meg, viszont php is szerepel benne, így htm kiterjesztés helyett most index.php-ban kell dolgozni.

A fontosabb részeket itt is kiemelem:

// a kép nevét bekérve megjeleníti azt

```
<tr>
```

```
    <td>Eredmény</td>
```

```
</td>
```



```
</td>  
</tr>
```

```
// megkapja az Excel fájlt
```

```
<tr>  
  <td>Log</td>  
  <td>  
    </a><a href="../../log/log.xls">log.xls</a></td>  
</tr>
```

A képen szintén egy űrlap látható, amit <http://localhost/jobs/> címen érek el. Megfigyelhető, hogy a robot már elküldte a képet, és a log fájlt is. Feladat száma, és Feladat mezőkben mi tudjuk begépelni a parancsainkat.

Feladat száma:	<input type="text"/>
Feladat:	<input type="text"/>
Előző kész van:	<input type="button" value="True"/> <input type="button" value="False"/>
Eredmény	
Log	log.xls
<input type="button" value="Adatok felvitele"/>	

10. ábra. Jobs weblap

Adatbázis / adattáblák

A phpMyAdmin-t a <http://localhost/phpmyadmin> címen lehet elérni. Egyszerűen, gyorsan és könnyen hozhatunk létre adattáblákat az adatbázisba. Három táblát készítettem, egyet a robot oldalra „robot” névvel, egyet a felhasználónak „jobs” névvel, és egy „other”-t a feladatok sorba rendezése miatt.

A táblák struktúrája:

Mező	Típus	Egybevetés	Tulajdonságok	Null	Alapértelmezett	Extra
<u>id</u>	int(11)			Nem		auto_increment
ido	time			Nem		
gps_szel	double			Nem		
gps_hossz	double			Nem		
mitsinalok	text	latin1_swedish_ci		Nem		
kep_kamera	varchar(255)	latin1_swedish_ci		Nem		
homerseklet	int(11)			Nem		
latasi viszonyok	varchar(255)	latin1_swedish_ci		Nem		

11. ábra. „robot” tábla

Mező	Típus	Egybevetés	Tulajdonságok	Null	Alapértelmezett	Extra
<u>id</u>	int(11)			Nem		auto_increment
feladat_id	varchar(4)	latin1_swedish_ci		Nem		
feladat	text	latin1_swedish_ci		Nem		
keszvan	varchar(255)	latin1_swedish_ci		Nem		
eredmeny	text	latin1_swedish_ci		Nem		

12. ábra. „jobs” tábla

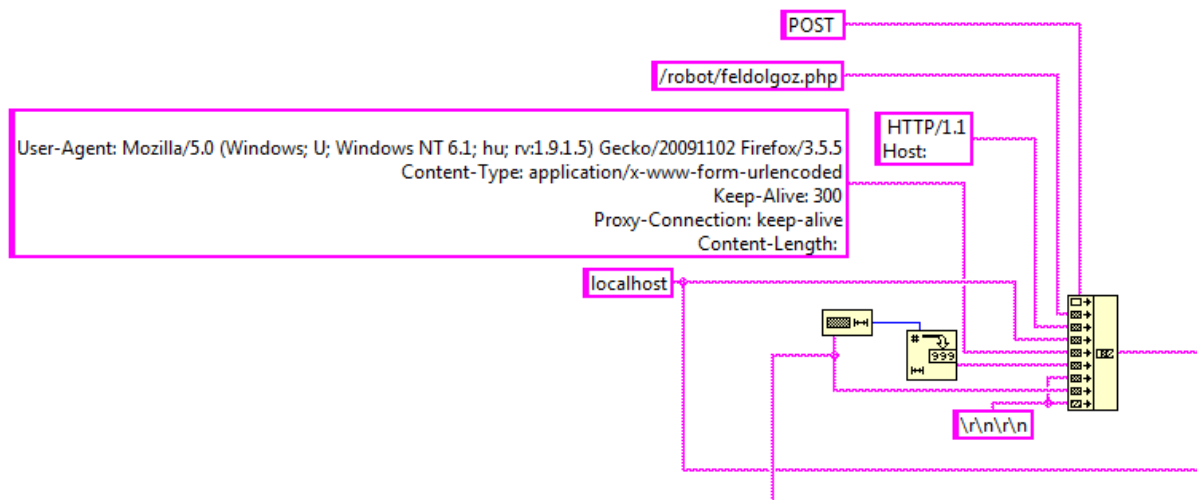
Mező	Típus	Egybevetés	Tulajdonságok	Null	Alapértelmezett	Extra
last_get	int(11)			Nem	0	

13. ábra. „other” tábla

Minden felküldött adat itt tárolódik, a mező nevével hivatkozhatunk rá a programban.

LabVIEW a roboton

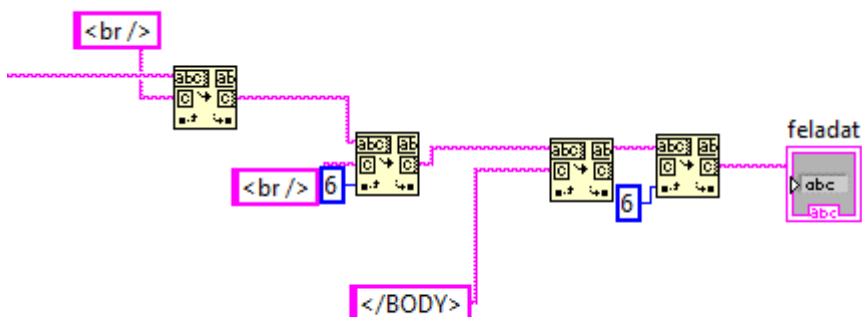
A kapcsolatot TCP Open Connection VI nyitja, ha megadtuk neki a portot és a címet amin csatlakozik. Ez egy TCP Write –hoz kapcsolódik, amibe egyetlen szál van még bevezetve a hiba lekezelésén kívül. A szálon sztringként össze van fűzve az összes információ, ami szükséges ahhoz, hogy az űrlapot kitöltve, az a mögött elhelyezkedő „robot” táblába írja a küldött adatokat. Tehát a Concatenate Strings VI egy nagyon hasznos segítség, mert mindent összefűz amit megkap és egyetlen vonalon továbbítja.



14. ábra. Adatok összefűzése

A TCP Write VI még ezeket kapja robottól, az ábrán látható sztringek mellett: -ido (amit a rendszerórától olvas ki), -gps_szel, -gps_hossz, -mitsinalok, -kep_kamera, -homerseklet, -latasi_viszonyok, -gomb.

Ha ezzel megvagyunk, a robothoz tartozó feldolgoz.php –ból a feltöltés után kapott üzenetből ki kell venni a következő feladatot, esetleg több feladatot, sorszámtól függően és megjeleníteni azt a Front Panelen.



15. ábra. Szövegrész kivágása adott helyről és indicator-ban tárolás

gps_szel	gps_hossz	Következo_feladat(ok) wait_40<do_39><g o_40><wait_40>
47,498970	19,043604	
mit csinállok	homerseklet	
goto	24	
latasi viszonyok	time string	
Jó	22:31:45	
kep_kamera		
r1.jpg		

error in	
status	code
	0
source	

16. ábra. Front Panel

A robot elküldi az elvégzett feladatról a log fájlt a képen látható értékekkel a felhasználónak, és megkapja a még el nem végzett parancsokat.

Összefoglalás

A dolgozatban megpróbáltam lépésről lépésre, érthetően elmagyarázni hogyan is építettem fel a kapcsolatot egy robot és a neki utasításokat adó kezelője között. Mindezek előtt általános információkkal próbáltam szolgálni a korábban összegyűjtött anyagok ismeretében a különböző típusba sorolható robotokról, ezek közül a legnépszerűbbek és a legelterjedtebbek az ipari robotok, mert a mindennapi gyorsan fejlődő életünket talán ezek könnyítik meg a legjobban. Bemutattam a programokat, program nyelveket amelyek segítségemre szolgáltak a feladatom teljesítéséhez.

Harcászati és Mars járó robotok is bővebb kifejtésre kerültek, mert egyrészt nagyon sok lehetőség van a további fejlődésükben, másrészt ezek hordozzák magukban mindazokat a csúcstechnológiai vívmányokat, amiket a mérnökök évtizedeken át fejlesztenek, finomítanak. Lehetőségeimhez mérten figyelemmel fogom kísérni a robotok fejlődését és kíváncsian várom, hogy mire jutunk a mesterséges intelligencia gépekbe integrálásával.

Végezetül a már nagyon régen lefektetett három törvénnyel zárnám a szakdolgozatomat [1].

Robotika 3 alaptörvénye (Isaac Asimov)

1. *A robotnak nem szabad kárt okoznia emberi lényben vagy tétlenül tűrnie, hogy emberi lény bármilyen kárt szenvedjen.*
2. *A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.*
3. *A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első és a második törvény előírásaiba.*

Felhasznált irodalom

1. Robot, robotika
<http://people.inf.elte.hu/kogqaai/robot/tortenelem.html> (Három törvény)
2009.11.05.
2. Ipari robot
<http://www.szgt.uni-miskolc.hu/~mako/eloadasvazlat.pdf>
2009.11.11.
3. Robotok a harcmezőn
Palik Mátyás – A harcászati pilóta nélküli repülő eszközök képességei
(Fogalom, feladatcsoportok)
4. Robotok a Marson
<http://www.origo.hu/tudomany/vilagur/20080619-emberek-vagy-robotok-nagy-kerdesek-a-mars-kutatasaban.html> (Robotok a Mars kutatására)
2009.11.07.
5. Szenzorok
http://mobil.nik.bmf.hu/tantargyak/ir-gyak/IR-Robot_esti.pdf
(Robotok felhasználási példái, szenzorok) 2009.11.07.
6. Telekommunikáció
<http://www.algotech.hu/hu/telekommunikacio>
2009.11.06.
7. Wi-Fi
<http://hu.wikipedia.org/wiki/Wi-Fi>
2009.11.09.
8. LabVIEW
<http://cnx.org/content/m13511/latest/>
2009.11.12.

Köszönetnyilvánítás

Szeretném megköszönni szüleimnek és nővéremnek a békés, nyugodt családi háttérrel és anyagi segítséget a tanulmányaimhoz, továbbá barátnőmnek az erkölcsi támogatást.

Köszönetet mondok dr. Szabó István témavezetőmnek az elméleti és Juhász Tibornak a gyakorlati tanácsaiért.