Dunakavics

A Dunaújvárosi Egyetem online folyóirata 2016. IV. évfolyam II. szám

Műszaki-, Informatikai és Társadalomtudományok

Baráth Artur
Biometrikus azonosítási eljárások bemutatása, elemzése és az RFID-rendszerek vizsgálata
I. rész

Auer Richárd–Tóth Roland–Burkus Ervin
Mobilrobot elektronikai és szoftveres fejlesztése

Tóbel Imre–Kővári Attila
LabVIEW Real Time valósidejű mérő- és irányítórendszer
I. Real Time Target PC-alapú tesztkörnyezet létrehozása

Dunakavics

A Dunaújvárosi Egyetem online folyóirata 2016. IV. évfolyam II. szám

Műszaki-, Informatikai és Társadalomtudományok

Megjelelenik évente 12 alkalommal

Szerkesztőbizottság

András István, Király Zoltán, Kukorelli Katalin, Palotás Béla,
Rajcsányi-Molnár Mónika.

Szerkesztőség

Ladányi Gábor (Műszaki)
Nagy Bálint (Informatika és matematika)
Szakács István (Gazdaság és társadalom)
Klucsik Gábor (technikai szerkesztő)

Felelős szerkesztő Németh István
Tördelés Duma Attila

Szerkesztőség és a kiadó címe 2400 Dunaújváros, Táncsics M. u. 1/a.

Kiadja DUE Press, a Dunaújvárosi Egyetem kiadója
Felelős kiadó Dr. habil András István, rektor

A lap megjelenését támogatta TÁMOP-4.2.3-12/1/KONV-2012-0051
„Tudományos eredmények elismerése és disszeminációja
a Dunaújvárosi Főiskolán".
http://dunakavics.duf.hu

ISSN 2064-5007

Tartalom

BARÁTH ARTUR

***Biometrikus azonosítási eljárások bemutatása, elemzése és az RFID-rendszerek vizsgálata** I. rész* 5

AUER RICHÁRD–TÓTH ROLAND–BURKUS ERVIN

Mobilrobot elektronikai és szoftveres fejlesztése 21

TÓBEL IMRE– KŐVÁRI ATTILA

LabVIEW Real Time valósidejű mérő- és irányítórendszer
I. Real Time Target PC alapú tesztkörnyezet létrehozása 41

Galéria
(Sóti István fotói) 60

BARÁTH ARTUR *

Biometrikus azonosítási eljárások bemutatása, elemzése és az RFID rendszerek vizsgálata (I. rész)

Összefoglalás: A címben szereplő téma aktualitását az adja, hogy napjainkban egyre több olyan esemény történik, melynek során szükségessé válhat a személyazonosság-vizsgálata. A kritikus információs infrastruktúrák védelmének egyik alapvető feladata az információs rendszerben kezelt adatok hozzáférés-védelme, amelynek igen fontos része az azonosítás-hitelesítés. Ennek okán sorra veszem a használatra alkalmasnak vélt technológiákat és az azokat kiszolgáló eszközöket. Elsősorban a modern és hatékony biometrikus azonosítási eljárásokat vizsgálom, de kiegészítő rendszernek a rádiófrekvenciás, birtoklás-alapú RFID-rendszert is tanulmányozom.
Kulcsszavak: Autentikáció, biometria, RFID, emberi tényezők.

Abstract: The actuality of the topic mentioned in the title comes from the more and more frequent daily events, when the checking of identity is necessary. A primary task of the critical information infrastuctures protection is the access control of data used by the informatics system where the identity-authentication is an important part. For this reason I will list all the technologies and supporting systems which seem suitable. I will examine the modern and effective biometric authentication processes in the first case but as for a supporting system I will investigate the Radio Frequency Identification system as well.
Keywords: Authentication, biometry, RFID, human factor.

* *Nemzeti Közszolgálati Egyetem, Doktor Iskola*
E-mail: magicman@mailbox.hu

Bevezetés

Jelen tanulmánnyal megpróbálok rálátást adni a jelenlegi biometrikus azonosítást lehetővé tévő technológiai megoldásokra. Mint az köztudott, a civil szféra adatbiztonsági és hozzáférés-jogosultsági kérdésekben komoly problémákkal küzd, amit minél előbb orvosolni kell. Ehhez olyan technológiára van szükség, ami megfelelő szintű védelmet nyújt, nem jelent kirívóan nagy költségeket a bevezetése és működtetése, továbbá fontos szempont, hogy minél inkább kizárja az ember által okozott autentikációs problémákat.

A munka célja, hogy a rendelkezésre álló technológiák közül kiválassza a fent definiált célnak és feltételeknek leginkább megfelelő rendszert. Ennek elérésére sorra veszem a szóba jöhető eljárásokat, ismertetem tulajdonságaikat, működésüket és ezek vizsgálatával és az eredmények összesítésével kiválasztom a leginkább megfelelőt.

Az adatvédelem jelentősége és a kapcsolódó fogalmak értelmezése

A 21. században a személyes adataink védelmére igen nagy hangsúlyt fektetünk. Nem engedhetjük meg, hogy illetéktelenek eltulajdonítsák adatainkat. A szakemberek, akik a személyes adatok védelmével foglalkoznak, nem hallgathatnak a biometrikus eszközök számának robbanásszerű növekedéséről. Az ilyen autentikációs azonosítást alapul vevő technológiák fejlődése és egyre nagyobb kiterjedtsége különös odafigyelést igényel. Ebben az azonosítási rendszerben a lényeges újdonság, hogy ezeket nem csak az emberek, tudják összevetni, hanem az erre a célra létrehozott számítástechnikai eszközökkel. Ez a szolgáltatás már csak az azonosítandó személy közreműködését igényli, más embereket nem, mivel ez a folyamat automatikusan történik. Az eszközbe beprogramozzák az adott ember tulajdonságait, amely majd a későbbiek során chip vagy arckép alapján képes felismerni és azonosítani.

Véleményem szerint manapság az értékeink védelme elengedhetetlen, legyenek azok a számítógépen lévő adataink vagy a körülöttünk lévő fizikai eszközeink. Ezek védelmére, elhatárolására bizonyos személyektől rengeteg lehetőségünk van. Ezeken belül beszélhetünk különböző védelmi szintekről. Vannak biztonságos, illetve kevésbé biztonságos megoldások, de a védeni kívánt adat vagy tárgy fontossága meghatározza, hogy milyen technikát szükséges használnunk.

Először a fizikai eszközök védelmére szolgáló eszközökről szeretnék értekezni, habár manapság már adatokat is védünk ezekkel a módszerekkel, de arról később.

A lehetőségek sorba vétele előtt szükséges tisztázni az elengedhetetlen definíciókat.

Autentikáció: Az autentikáció egy biztonsági szolgáltatás. Célja az, hogy megbízhatóan meg tudja állapítani egy üzenet, vagy kérés eredetét. A klasszikus módszer, ahogy az emberek azonosítják magukat a számítógép felé (és ahogy a számítógépek is azonosítják egymást) a jelszó megadásával.

Az autentikáció lényegében az egyéni jogosultság meghatározására, a felhasználó azonosítására szolgál. Magyarán megszabja, hogy mely felhasználó (pl. munkahelyi dolgozó) milyen információkhoz juthasson hozzá, meddig terjedhessen a hatásköre. [1]

Biztonsági szint költségarányát tekintve az alábbi kategóriákba sorolható:
– *Low-end* (Alacsony kategóriás, olcsó): Célja arra törekedni, hogy egy helytálló biztonsági szintet alakítsanak ki a lehető legolcsóbb, leggazdaságosabb megoldások felhasználásával.
– *High-end* (Csúcskategóriás): Magas biztonsági szintet igénylő rendszereknél alkalmazzák. Hátránya, hogy drága megoldásnak számít.

Biometrika: A biometrika az emberek azonosítása egy vagy több lényeges testi vagy viselkedési minta alapján. A biometrikát a személyazonosságot kezelő és a hozzáférést vezérlő módszerként alkalmazzák elsősorban az informatika területén. [2]

A biometrika két fő csoportba sorolható:
– *Élettani-biometrika:* Ez számít talán a legelterjedtebbnek a biometrikai eszközök közül. Ezek minden esetben az egyes emberi testrészek alakjával állnak kapcsolatban. Ide tartoznak: az arc (geometria, bőr rajzolat, hőtérkép, mosoly, dinamikus jellemzők), ajak (alakja, lenyomata, mozgása), antropometria (fizikai méret pl. magasság, fejhossz, vállszélesség, stb.), bioelektromos mező, bőr spektroszkópia (képalkotás technikai megoldásának módszere), DNA/DNS, fogsor (harapás, fogsor röntgenfelvétel), fül (geometriája, fülcsatorna geometriája-echo), írisz, retina, kézgeometria (kéz, ujj), köröm (körömágy), pulzus, szag/illat, talp, ujj (fodorszál, ujjperc, ránc, ízület ránc, topográfia), véredény (ujjon, kézfejen, tenyéren, csuklón) vizsgálatai.
– *Viselkedés-biometrika:* Kevésbé elterjedt biometrikai módszer. Ezen biometrika lényege, hogy az emberi test által végrehajtott mozgásokat és az egyén által kifejezett viselkedéseket vizsgálja meg. Ilyen például a billentyűzet használat-, hang- (beszéd, nevetés), járás-, kézírás- (dinamikus aláírás, aláírás hangja), egérhasználat dinamikája.

Az azonosítási rendszereket jogosultság megállapításra használjuk. Ez lehet belépés egy adott számítógépes rendszerbe, vagy hozzáférés valamilyen kritikus adathoz. A belépést vagy hozzáférést mindig valamilyen rendszer kezeli, ami lehet humánerőforrás-alapú vagy elektronikus, még inkább informatikai bázisú. A továbbiakban az informatikai vonalon haladunk tovább. Értelemszerűen több lehetséges megoldás van céljaink megvalósítására, ezeket több szempontrendszer alapján kell sorba vennünk ahhoz, hogy a vizsgálatok végén helyes következtetésre jussunk.

[1] Imre Sándor–Kis Zoltán: *RFID-rendszerek vizsgálata felhasználás és technológia szempontjából.* http://www.rfid.answare.hu:8080/site/kutatasi-erdmenyeink/radios-megoldasok/2006/rfid-rendszerek-vizsgalata-felhasznalas-es-technologia-szempontjabol.pdf/view (2012. 02. 12.)

[2] Varga Domonkos –Oláh András: Biometrikus azonosító rendszerek. *Híradástechnika 2004.* http://www.hiradastechnika.hu/data/upload/file/2004/2004_07/HT0407-8.pdf (2013. 10. 10.)

Szinte minden esetben a védelem feloldására személyazonosítást használunk. Erre három féle módszert alkalmaznak az elenyészően kisméretű kivételektől eltekintve:
– Tudásalapú.
– Birtokolt tárgy alapú.
– Biometrikus tulajdonság alapú.

Ezek biztonsága ebben a sorrendben, fentről lefelé egyre nagyobb. Mint, tudjuk, tökéletes biztonság nincs, de ezen lehetőségek használatával lehet elérni egy olyan biztonsági szintet, ami gazdaságilag és biztonságtechnikailag is elfogadható. Alább részletezem a három módszert:

Tudásalapú. Valamilyen felhasználónév, jelszó, PIN-kód kombinációjával azonosíthatjuk magunkat. Az ismertebb és használtabb megoldás a jelszó, aminek minden esetben tartalmaznia kell az alábbiak közül legalább hármat:
– kis és nagybetű,
– szám,
– speciális karakter.

Ezen felül a hosszúsága is fontos tényező, mert minél több karakterből áll a jelszó, annál több idő és munka visszafejteni. Az általánosan elfogadott minimum hosszúság szakmai körökben 16 karakter. A PIN (nem keverendő a bank- és SIM-kártyák PIN-kódjaival) azonosításnál a titok általában egy négy, hat vagy nyolcjegyű decimális szám, esetleg egy öt karakter hosszú alfanumerikus jelsorozat. A négy karakter hosszúságú PIN 10^2, a hat 10^6, a nyolc 10^8 variációval rendelkezik. Az alfanumerikus karaktersorozat általában az angol ABC 26 betűjét és a tíz számot alapul véve, 5 karakteres jelszót ad, $(26+10)^5$ azaz 60466176 darab variációval. A három azonosítási módszer közül önmagában az utolsó a leggyengébb.

Birtokolt tárgy alapú. Ez esetben egy tárgyat tartunk magunknál, aminek a segítségével azonosíthatjuk magunkat. Ez legtöbb esetben chipkártya, ritkábban RFID- (RFID: Radio Frequency IDentification – rádiófrekvenciás azonosítás) chip, de egyre elterjedtebb dolog egy kis implantátum behelyezése a bőr alá, ami szintén lehet RFID-rendszerű eszköz. [3]

Chipkártya. A kártyát magunknál tartjuk, amikor azonosítani szeretnénk magunkat, akkor odatartjuk a leolvasóhoz és az azonosítja. Rádiófrekvenciás technológiát használ, manapság 112 bites kulcsot alkalmaznak 3-DES (DES: Data Encryption Standard – szimmetrikus kulcsú titkosítási eljárás. A 3-DES vagy tripla DES, a DES egy változata. Az alkalmazott változattól függően kettő vagy három titkosító kulcsot használ a hagyományos DES által használt egy kulccsal szemben.) kódolással. Ezt tartalmazza a chipkártya. A rendszer összeveti az SQL-adatbázisban szereplő kulcsokkal és így azonosítja. Az ilyen rendszerek gyenge pontja a leolvasó és az adatbázis közötti kapcsolati rész, bizonyos vírusokkal eredményesen támadható ugyanis (SQL injection).

[3] Bunyitai Ákos: A ma és holnap beléptető rendszereinek automatikus személyazonosító eljárásai biztonságtechnikai szempontból. *Hadmérnök 2011.* http://www.hadmernok.hu/2011_1_bunyitai.pdf (2013. 08. 11.)

A chipkártyáknak és az ilyen módszert használó rendszereknek számos fajtája van:
– Működési frekvencia szerint (125 kHz, 13,56 MHz, 430 MHz, 862-956 MHz (UHF), 2,45 GHz, stb.
– Olvasási távolság szerint (<0.05 m, 0,05-0,5 m, 0,5-2 m, >2 m).
– Energiaellátás módja szerint (passzív, aktív, részlegesen passzív).
– Végezhető művelet szerint (Gyárilag programozott, Egyszer írható, Átírható, Írható/olvasható).

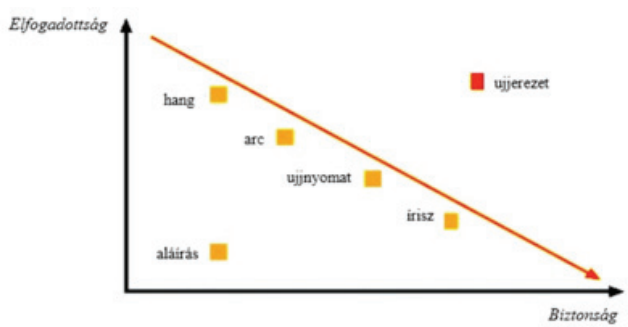
Chipek biztonsága:
– Nyílt (open): egyszerű fix kód, olvasható és értelmezhető (vagy strukturálatlan) adattartalom.
– Kódolt (encoded): olvasható, nem értelmezhető, de másolható.
– Adathozzáférés védett (password protected): jelszóval elérhető, önmagában másolhatatlan.
– Titkosított (encrypted): jelenleg a legbiztonságosabb.

A chipkártyás azonosítás továbbfejlesztése a beültethető implantátum az emberbe, ami szintén egy chipet tartalmaz. Az Amerikai Egyesült Államokban jelenleg a Veri Chip az egyetlen, amit emberi RFID-implantátumként lehet használni. Az azonosítás ugyanazzal a módszerrel történik, mintha kártya lenne nálunk. Ebben egy 16 hosszú kód található, s azzal történik az azonosítás.

Biometrikus rendszerek általános ismeretei

Biztonság szempontjából a biometrikus azonosítóknak is van egyfajta sorrendjük, amit az első ábrán láthatunk.

1. ábra.



Forrás: Szerző.

A legkiemelkedőbbek ezek közül az ujjlenyomat- és az íriszazonosítás. Ebből következően a továbbiakban ezekre koncentrálunk. Elsőként megválaszolok két kérdést, ami szinte minden esetben felmerül a biometrikus módszerek használatakor és igazolást ad arra, hogy a világ igenis ezen irányba tart.

Az alábbi két kérdés válaszai tovább erősítik a biometrikus azonosítás létjogosultságát:

Igaz-e, hogy a jövőben a biometrikus azonosítás lesz majd a személyazonosság bizonyításának alapja?

Igen, mivel a hagyományos azonosító módszerek – például a jelszavak vagy a különböző személyazonosító kártyák – azon alapulnak, „amit tudsz", vagy „amid van". A biometrikus azonosító módszerek viszont azon, „ami vagy": anatómiai sajátosságokon (arc-, ujjlenyomat- vagy íriszfelismerés), vagy viselkedési mintázatokon (aláírás vagy járás). Ezek az eljárások jóval biztonságosabbak a hagyományos módszereknél. A jelszavakkal és azonosító okiratokkal, kártyákkal ellentétben a biometrikus módszereket jóval nehezebb találgatással, megosztással, kölcsönadással, másolással vagy hamisítással kijátszani.

Miért terjednek el már az igen közeli jövőben a biometrikus módszerek?

Mert adott mind a kereslet, mind a kínálat. A keresleti oldalon egyre nagyobb az igény a biztonság iránt, és mind nagyobb a hamisítástól való félelem. Az államok szeretnék tudni, kik lépik át határaikat, kiknek fizetnek szociális juttatásokat. A cégek ellenőrizni akarják, kik lépnek be épületeikbe, kik azok, akik hozzáférnek weboldalaikhoz és adatbázisaikhoz. Ami kulcsfontosságú: a technológia társadalmi elfogadottsága lassacskán szintén nő; a személyazonosság biztosítása, valamint a lopás megelőzése végett egyre többen használnak biometrikus módszereket laptopjuk, telefonjuk védelmére vagy fizetéskor. A kínálati oldalt tekintve a közelmúltban bekövetkezett technológiai robbanás révén kifejleszthettek olyan kisméretű, olcsó és gyors szenzorokat, melyek teljesen automatizálhatók, alkalmasak a „real time" képalkotásra.

A továbbiakban sorra vesszük az egyes eljárások működését.

Ezen azonosító rendszereket úgy határozhatjuk meg, mint eljárások és eszközök összességét, amelyeken a személyek mérhető testi adottságait, tulajdonságait használja fel valamilyen technika alkalmazásával, személyazonosság vagy azonosítás megállapítására. Az azonosítás során az egyén testével történő összehasonlítása valósul meg. Ebben az esetben a rendszertől két választ várhatunk az igent vagy nemet. A személyazonosítás során a rendszer felismeri a személyt olyan módszerrel, hogy megkülönbözteti mások tárolt adataitól. Az első esetben két adat összehasonlítása zajlik, még az utóbbiban egy adat összehasonlítása a többi tárolt adattal.

A biometrikus rendszerek használatának egyre nagyobb szélesedését a biometriai adatok speciális jellemzői teszik indokolttá. Az emberek adatai tekinthetők univerzálisnak, abban az értelmében, hogy ilyen adatokkal mindenki rendelkezik.

Az azonosításra vagy a megkülönböztetésre a legjobb megoldás a DNS – minták, az ujjlenyomatok és a retinaképek, de az arc geometriája a kéz és a hang is használható erre a célra. Ilyen testi jellemzőkkel rendelkezik minden ember. Ezek az adatok egyediek és különbözőek, tehát levonhatjuk azt a következtetést, hogy nincs két egyforma fizikai karakter. Az idő elteltével ezek az adatok nem fognak megváltozni.

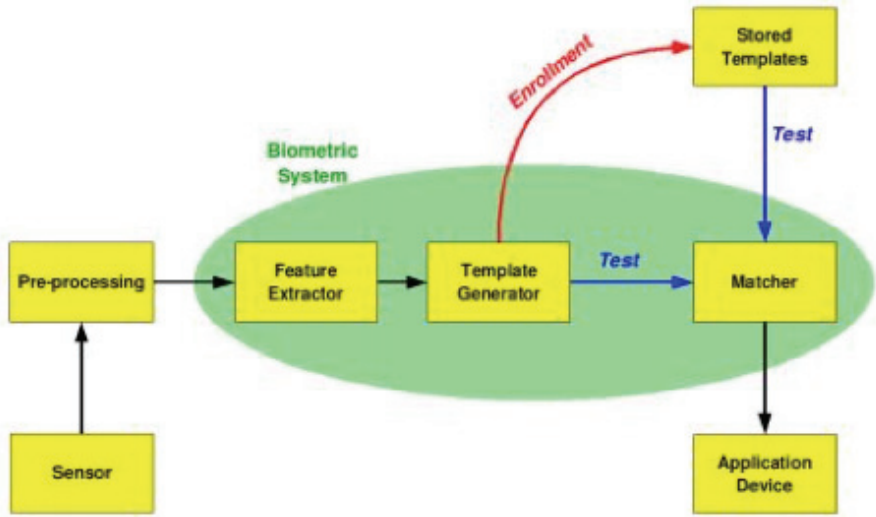
Természetesen ez nem minden biometrikus adatról mondható el. Ha tágabb értelemben vesszük ezeket az azonosító rendszereket, nem csak az egyén fizikai tulajdonságai vehetők figyelembe, hanem a viselkedése is. A viselkedést leíró tényezők közé soroljuk pl. a személy aláírását, mozgását. A biometrikus azonosító rendszerek egy része elegyíti az egyes adatfajták felhasználását, mint például ujjlenyomat- és hangazonosítás, mások pedig olyan módszerekkel, amelyek az egyén ismeretét veszik, figyelembe például tudja-e mi a jelszava, a bankkártyájához tartozó PIN-kód stb. [4]

[4] Szabó Máté Dániel: *Biometrikus azonosítás és adatvédelem.* http://szabomat.hu/tanulmany/biometria.pdf (2013. 08. 16)

Alapvető követelmények e rendszerekkel szemben:
– mindenkire értelmezhető, használható legyen,
– egyediség (ne legyen két azonos, akire igaz),
– időbeni állandóság (hosszú ideig statikus jellemző),
– mérhetőség (lehetőleg nagy teljesítménnyel, gyorsan),
– nehezen hamisítható.

A *2. ábrán* egy átlagos biometrikus vizsgálat általános blokkvázlata látható. Működés szempontjából ezek a minimálisan szükséges lépések, amiknek meg kell történnie.

2. ábra.

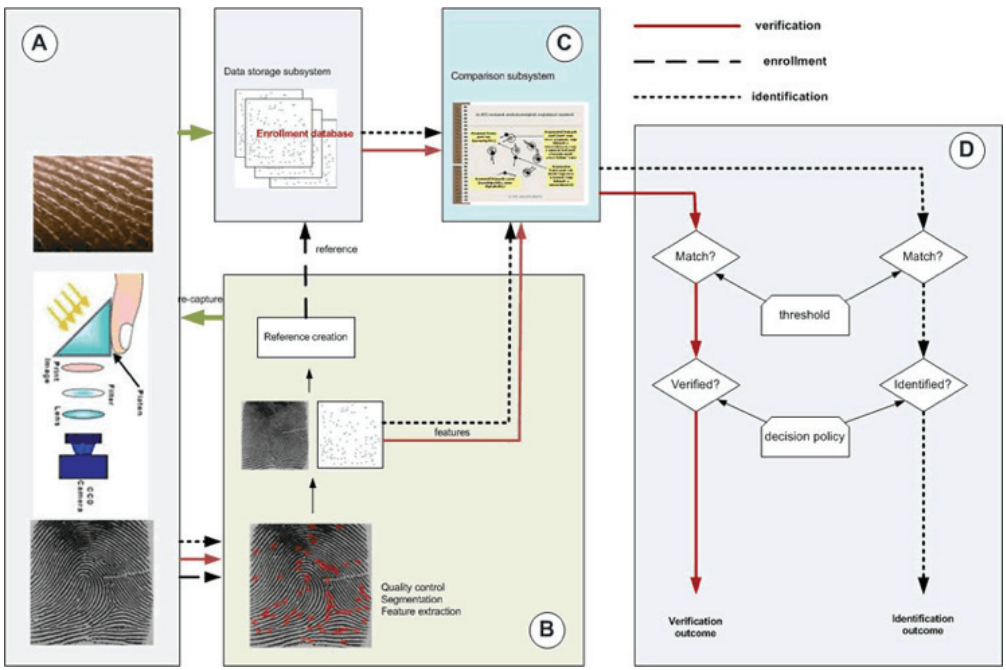


Forrás: http://upload.wikimedia.org/wikipedia/commons/3/3c/Biometric_system_diagram.png

A blokkvázlaton végigkövethetjük a biometrikus azonosítás működését. A szenzoradatokat előfeldolgozásnak vetjük alá (pl. zajszűrés), majd a számunkra kívánatos adatokat szűrjük ki az adathalmazból (feature extraction).

A megmaradt adatok alapján előállítunk egy mintát, amit az adatbázisban eltárolt mintákkal vethetünk össze, majd az eredményeket továbbítjuk további feldolgozásra (riasztás, értesítés, stb.). A harmadik ábrán megtekinthető, milyen elemekből épül fel egy biometrikus azonosító rendszer.

3. ábra.



Forrás: http: www.wikipedia.org

A biometrikus azonosító rendszerek általánosságban négy fő részből állnak. Először is szükség van egy szenzorra, mely rögzíti a biometrikus mintázatot – legyen az ujjlenyomat, írisz-textúra, kézírás, hangminta vagy bármi egyéb.

Ezután ki kell emelni a minta jellegzetes vonásait, melyek az azonosítást megkönnyítik, illetve szükség van egy adatbázisra, mely tartalmazza az egyes biometrikus jellemvonásokra jellemző példákat. Végül kell egy összehasonlító eszköz, mely összeveti az egyes mintát az adatbázisban szereplő mintákkal. [5]

[5] Kovács Tibor (2009): *Biometrikus azonosítás.* Főiskolai digitális jegyzet. BMF.

A megbízhatóságot tekintve számos vizsgálat történt korábban, ezek segítségével a következőket állapíthatjuk meg:

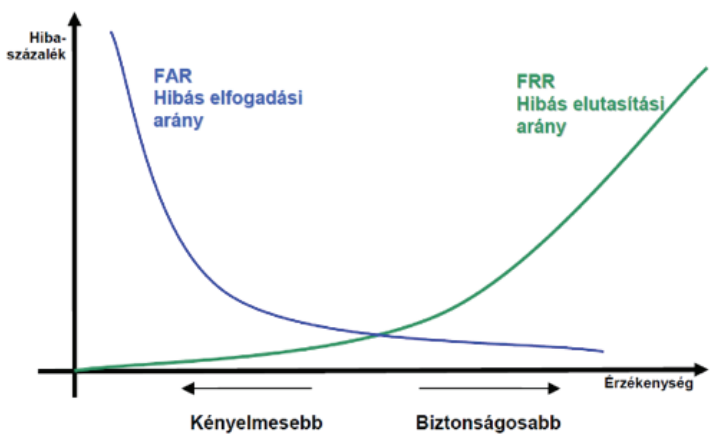
Az azonosítás pontosságát két mutatóval határozzuk meg:
– FAR (**F**alse **A**ccept **R**ate) ez a mérőszám azt mutatja meg, hogy a rendszer milyen gyakorisággal enged be, az arra NEM jogosult felhasználókat.
– FRR (**F**alse **R**eject **R**ate) ez a mérőszám azt mutatja meg, hogy a rendszer milyen gyakorisággal nem enged be, arra jogosult felhasználókat.

4. ábra.

Arcfelismerés	**2000:1**
Hangazonosítás	500:1
Ujjlenyomat azonosítás	1 000 000:1
Íriszvizsgálat	10 000 000:1
Retinaazonosítás	10 000 000:1

A *4. ábrá*n néhány FAR-mutató összehasonlítása (hány helyes azonosításra jut egy téves) látható.

5. ábra.



A FAR-t és az FRR együttesen, egy diagramon ábrázolva két görbét kapunk, amik egy ponton metszik egymást. *5. ábra.* Ezt a pontot nevezzük EER-nak (Equal Error Rate: egyenlőhiba-arány). A *6. ábrán* láthatóak az egyes vizsgálatok relatív pontosságának, a FAR- és a FRR-értékekkel számolt EER-értékeinek összehasonlítása.

6. ábra.

Arcfelismerés	**200:1**
Hangazonosítás	50:1
Ujjlenyomat azonosítás	500:1
Íriszvizsgálat	131000:1
Retinaazonosítás	10 000 000+:1

A mai ujjlenyomat-azonosító rendszerek FRR-értéke 0,01 százalék körüli (tízezerből egy hiba), míg az FAR-értékük 0,1 százalék körüli (ezerből egy hiba). Egy biometrikus rendszer aktuális teljesítménye természetesen több tényező függvénye. A felhasznált biometrikus sajátságokon túl sok függ a szenzor érzékenységétől, az adatbázis méretétől és különböző egyéb tényezőktől (hőmérséklet, páratartalom, kül- vagy beltéri felhasználás és így tovább). A különböző felhasználásoknál persze más és más szempontok fontosak. [4]

[4] Szabó Máté Dániel: *Biometrikus azonosítás és adatvédelem.* http://szabomat.hu/tanulmany/biometria.pdf (2013. 08. 16)

Az ujjlenyomatok azonosítása

Az ujjlenyomatok biztonságos használata az adatvédelemben az egyediségen, illetve az állandóságon alapszik. Nincs két olyan ember még az ikrek sem, akiknél azonos ujjlenyomatot találtak volna. A lenyomatok kétféle besorolását tekintsük át elsősorban, melyek Henry (1900) és Galton (1892) struktúrák.

A Henry-struktúránál a vonulatok száma, míg a Galton struktúránál az ismertető jelek a mérvadók. Bár az ujjlenyomatok egyedisége, vagy minősége nem alapja a Henry-struktúrának, ellentétben a vonulatokkal és egyedi sajátságokkal. Tehát, hogy megállapítható legyen két ujjlenyomatról, hogy azonosak-e vagy sem, megfelelőszámú (kitüntetett) pontnak, illetve sajátságnak kell azonos helyen elhelyezkedniük.

– Kezdődő és végződő vonulat.
– Szétágazás.
– Sziget.
– Bekerítés.

Ez a lista idővel bővült és letisztult. Az American National Standards Institute (Amerikai Szabványügyi Hivatal) javaslata szerint a következő lett:
– Végződő.
– Elágazás.
– Keverék.
– Meghatározatlan.

A ma legelterjedtebb rendszer a Minutiae–Coordinate-modell (az FBI is használja). Ez a rendszer az azonosításhoz két sajátságot, a végződést és elágazást használja fel. A sajátságok minden pontját egy-egy koordinátaként értelmezi, ennek alapján jeleníti meg és ellenőrzi az azonosítást.

A későbbi Sparrous-modell strukturális ábrázolási módot alkalmaz az ujjlenyomatok megrajzolására, ami lehetővé tette azt, hogy a lenyomat még akkor is felismerhető, ha az torzult is.

A teljes ujjlenyomat kb. 100 barázda-elágazást és -végződést, ún. minucia pontot (sajátossági pont) tartalmaz. Az ujjlenyomat-azonosítók 30–60 db minucia pontot hasonlítanak össze az adott mintáról. A kapott kép közvetlenül is felhasználható az azonosításra (globális elemzés), vagy létrehozunk egy ujjlenyomat-kódot, a minucia pontoknak egymáshoz való távolsága, egymással bezárt szöge alapján. Ezek a minták 100 és 1500 Byte közötti terjedelműek lehetnek, az algoritmustól, az azonosított pontok számától függően és természetesen szoros összefüggésben a rendszer megbízhatóságával.

Az érintésalapú biometrikus azonosítás lehetséges módszerei

Tenyérlenyomat azonosítás: nem egy általában használatos biometrikus azonosítási eljárás, elsősorban a bűncselekmény elkövetésének helyszínén lelhető fel. Azonosításukkor a tenyéren található fővonalak rán-colt elhelyezkedését, a fodorszálakat és a szövetmintákat elemzik. A precíz elemzést követően az ujjlenyomat-mintához hasonló információt kapjuk meg.

Kézgeometriai azonosítás: működésének lényege, hogy a kéz formájáról és felületéről vesz mintát, és azt elemezve veszi figyelembe az ujjak hosszúságát és szélességét, a kézfej szélességét, illetve méretarányaiban a tenyeret és az ujjakat. Négy pozicionáló tüske segítségével érik el, mely a tenyeret azonos állásba helyezi a beolvasáshoz. Léteznek pozicionáló tüske nélküli eszközök is, amelyek megkülönböztetett sajátos értékeket vesznek górcső alá. Ilyen rendszereket pl. a munkaidőnyilvántartási-rendszereknél alkalmaznak. Előnye, hogy más rendszerekkel is egybeépíthető.

Tenyér – és ujj – azonosítás: viszonylag új módszer. Működését tekintve az ujjat vagy tenyeret infravörösfény megvilágítja, ami a különböző szövetekről a más szintű elnyelődése miatt más intenzitással verődik vissza. A fényt az erekben folyó vér sokkal jobban elnyeli, így az pontosan kirajzolódik az eszköz számára.

Előnye, hogy a működését nem befolyásolják a bőr felszínén található sérülések és szinte nem lehet meghamisítani.

Ujjlenyomat-azonosítás: az ujjunkat a leolvasóra helyezzük, ami megvizsgálja, hogy élő-e az ujj, ezt hőmérséklet és felületi nedvesség mérésével teszi, majd elvégzi az azonosítást.

A továbbiakban a fenti négy lehetőségből az ujjlenyomat vizsgálati rendszert választom. Ennek oka, hogy az azonosítást nem csak fizikai belépésre tervezem használni, hanem a számítógépes bejelentkezést is ezzel kívánom megoldani. A tenyérleolvasó ekkor igen komoly praktikussági aggályokat vet föl méretei miatt. Gazdaságossági szempontból is megfelelőbb az ujjlenyomat olvasás, ugyanis alacsonyabb költséget jelent a fizikai eszközök beszerzése. Végül, de nem utolsó sorban, higiéniai szempontból is előnyösebb a választott eljárás.

A döntésnek megfelelően tovább mélyülünk az ujjlenyomat-vizsgálat rendszerébe. Ujjlenyomat-olvasó rendszereknél számos technológiai megoldás létezik. Egyes azonosítók képesek megvizsgálni, hogy az ujj, élő ember kezén található-e, ezt úgy elvégezve, hogy az ujj hőmérséklet- és nedvességtartalmának analizálásával állapítják meg. Az eszközök közötti hasonlóságot az adja meg, hogy az ujjakon lévő fodorszálak egyediségét elemzi, tárolja, majd végül összehasonlítással állapítja meg az azonosítandóval. A technikának két változata létezik, az optikai és nem optikai módszer.

Optikai eszközök

Az ujjlenyomatot egy képfelbontó eszköz felületére képezzük le. A képfeldolgozó egység egy CMOS- (Complementary Metal-Oxide Semiconductor – komplementer fém-oxid félvezető) vagy CCD- (Charge-Coupled Device- töltés-csatolt eszköz , analóg jelek továbbítására szolgáló elektronikai alkatrész-lánc) elem. A barázdáltság megvilágítására LED-diódákat használnak. Jelenleg is használt fajtái:

– *Totálreflexiós:* ez esetben az ujjunkat egy prizma felületére helyezzük, majd a megvilágítás során a kapott kép egy képfeldolgozó eszköz felületére képződik le.

– *Diffrakciós:* a totálreflexiós eljáráshoz hasonló működésű, de a prizmát Fresnel lencse (a lencse felületét egy sor koncentrikus gyűrű alakú szakasz alkotja) váltja fel.

– *Chipszenzor:* az ujjat a szenzor felületére helyezzük és a feldolgozandó jeleket pedig optoszálakon keresztülvezeti a képfeldolgozó eszközre.

– *Termikus elemzés:* ebben az esetben nem kell az eszközt megérinteni csak az ujjunkat elhúzni előtte. Ez az eszköz szeletenként olvassa le az információt és alkotja meg a képet. A beépített szenzor pedig a barázdák hőmérsékletbeli különbségét érzékeli. Eddig kevés gyártó próbálkozott e téren, de várhatóan nőni fog a piacon betöltött szerepe és ezzel együtt csökkenni a jelenleg magas ára. Jól használható extrém viszonyok között is.

– Rádiófrekvenciás jelet küldünk az ujjra, mely a felületéről visszasugározza a jelet a vevő szenzor felé. Ez az eljárás képes mélységi képet is alkotni az ujjakról és a barázdákról.
– Kapacitív-elv szerint apró kondenzátorokkal felszerelt érzékelőfelületére helyezzük az ujjat, amely eltérő kapacitást jelez a völgyek és barázdák között. Ez az eltérő jel kerül elektromosan átalakult jellé, amely kiértékelődik. Egy rácsos szerkezetű vezetőrétegből áll, mely egy elektromágneses mezőt alakít ki maga felett. Mikor az ujjunk hozzáér, akkor tőle töltést veszünk el, amelynek a pozícióját a készülék képes érzékelni.
– Ultrahangos jeleket sugároz az érzékelő az ujjra, amelyről a visszaverődő hullámjelek segítségével mélységi képet alkot.
– Nyomásérzékelő a piezo-elektromos mátrix kerül beépítésre a szenzorfelület alatt, mely az ujjfelület egyenlőtlenségeiből alkot képet.
– Az E-mező-technológia a bőr elektromos mezejét méri. Mindennapos használatra alkalmas, az ujjlenyomat minőségére nem érzékeny. Ez az elektronikus leolvasás egy elektromos mezőt alakít ki az ujj és a vele érintkező félvezető körül, amely felveszi az ujjlenyomat barázdáltságát. A kapott kép nagy tisztaságú, háromdimenziós, de kisméretű.

A kutatások, fejlesztések a biometrikus azonosító rendszerek összes elemét érintik, így egyre jobb, megbízhatóbb és gyorsabb szenzorok, algoritmusok születnek. A fejlesztések egyik iránya, hogy egyesítsék a különböző, független biometrikus forrásokból származó adatokat, és így lecsökkentsék a hibaszázalékot. Az US–VISIT stratégiája, a kettős ujjlenyomat-ellenőrzéssel a „multibiometrikus" rendszerek egyik lehetséges formája.

Mindezek tükrében és ismeretében megállapítom, hogy az ujjlenyomat-vizsgálatot be lehet és be kell vezetni a magyar civil informatikai rendszerekben történő autentikálásra, ahol ezt megkívánja az infrastruktúra és az adatbiztonság. Viszont szükséges egy egyszerűbb, alacsonyabb költségeket jelentő rendszer választása is, azon területekre, ahol nem kíván olyan magas szintű biztonságot a rendszer, amit az ujjlenyomat-alapú autentikáció biztosít.

A jövőben az ujjlenyomatok szolgáltatta információ kiegészíthető lesz az arcfelismerő szoftverekkel. Ennek jegyében most a retina-, írisz-, arcfelismerő rendszerek vizsgálata következik.

Az érintésmentes biometrikus azonosítás lehetséges módszerei

Az írisz-felismerést az ember szivárványhártyájának (íriszének) egyedi és időben állandó mintázata teszi lehetővé. Videokamerával készítenek egy képet a szemről, majd az írisz területét eltárolják, miután levágták a képről a pupillát, a szemhéjat és a szempillákat. Az írisz egyedi mintázatában rejlő információt ezt követően digitális formában rögzítik. Az azonosításhoz a bitek sorrendjét kell összehasonlítani.

Az íriszazonosítás, a szem szivárványhártyáján alapul. Ennél az eszköznél a tévedések lehetősége szinte nulla. A szem szivárványhártyáján 400 különböző azonosítási pont van. Az írisz életünk során sosem változik. Annak az esélye, hogy két ember írisze pontosan megegyezzen kizárt. Az eszköz az azonosítás során a szivárványhártya látható és láthatatlan elemeit analizálja. A szabad szemmel látható elemek közé tartozik az írisz sugaras mintázata, körökkel, árkokkal és a koronával. A retinahártya-erezetet pedig infravörös leolvasás után válik láthatóvá. A pupilla-reflexek is megfigyelhetők az azonosításkor, így kizárhatóak a kontaktlencsével való visszaélések. Az íriszazonosító rendszer leolvasója a szivárványhártya képét az összes jellegzetességekkel (gödröcskék, körök, árkok, korona, szövetszálak), melyek a szemet egyedivé teszik, háromdimenziós kontúr-térképpé alakítja. Az így szerzett információk digitalizálás után egy pontosan 2048 számjegyű kódot alkotnak.

Ezt hasonlítják össze a későbbiekben, az adatbázisban tároltakkal. A vizsgálat során két módszert alkalmazhatunk. A felvétel készítése alapján kétféle leolvasást különböztetünk meg: aktívat és passzívat. Az aktív leolvasás a felhasználó aktív közreműködését igényli, mivel a kamerától 0,15–0,35 m távolságra kell tartania a szemét. Ezzel szemben a passzív eljárás a felhasználók szempontjából sokkal kellemesebb, hiszen ez esetben a rendszer először egy nagy látószögű kamera segítségével határozza meg a szemek helyzetét, majd arra fókuszál rá egy másik kamerával, és végzi el a leolvasást akár 0,3–1 m távolságból is. Mindkét esetben a leolvasás körülbelül 1–2 másodpercet vesz igénybe, ez megegyezik az ujjlenyomat azonosításhoz szükséges idővel. Sajnos az aktív eszközök higiéniai problémákat okozhatnak, a passzív eszközök hátránya, hogy rendkívül bonyolult technikai megvalósítást igényelnek, így az áruk is igen magas.

Retina-azonosítás: a szemfenéken található retinát infravörös fénnyel világítják meg. Működése az ujj- és tenyér-azonosító rendszerekhez hasonlít. A visszaverődő infravörös sugarak eltérő mértékben nyelődnek el, így azok ki tudják rajzolni a szemfenék érhálózatát. Az olvasóval közvetlen kapcsolatot kell kialakítani, ezáltal igen nagy a fertőzés veszélye. Ezt a technológiát igen ritkán alkalmazzák, általában nagy biztonságot megkövetelő rendszerek esetében.

Arcfelismerés látható fényben: Az arc felismerése – párosulva a mintaazonosító eljárásokkal – jó azonosítást szolgálhat. Jelenleg különleges esetekben használják. Nem alkalmas tökéletesen egyforma egypetéjű ikrek megkülönböztetésére.

Arcthermogram: felvétel, melyet infrakamerával készítenek, és az arc hőmintáját mutatja. A kép egyedi, és kombinálva nagy bonyolultságú mintaazonosító algoritmussal – amely ellenőrzi a relatív hőmérséklet-különbségeket az arcon – olyan technikát kínál, amely független a kortól, az egészségi állapottól, de még a test hőmérsékletétől is. A módszer tizenkilencezer „adatpont" felvételével kivételesen biztonságos eljárást kínál, amennyiben a technológiai költségek elfogadható szintre csökkennek. Nagy pontosságú, képes megkülönböztetni a teljesen egyformának tűnő ikreket is, akár sötétben is. További előnye a teljes diszkréció. Ennek a technológiának a fejlesztése manapság a költségek csökkentésére irányul annak érdekében, hogy minél szélesebb körben váljék alkalmazhatóvá az azonosítási és hitelesítési eljárásokban. Az arcthermogram a legígéretesebb módszer jelenleg.

A feldolgozás a többi biometriaihoz hasonlóan a zajszűréssel kezdődik. Zajként, zavaró tényezőként szerepelnek: szempilla-, szemhéj-, pupilla-tükröződések. Utána történik meg a struktúra felismerése, majd a kód előállítása. Ezen adat egy polárkoordináta-rendszerben leírt sajátosságok sorozata, melyet a pupillától kifele haladva körkörösen vesznek fel. Mivel kamerával történik a felvétel, ami másodpercenként több tíz képet is készíthet közel azonos beállításból, lehetőség van a „legjobb" és a legkisebb zajt tartalmazó kép kiválasztására (szempilla, szemhéj, reflexiók).

Az írisz-azonosítás rövid idő alatt felzárkózott az ujjnyomat-azonosítás mellé, és ma már egyértelműen az ujj kiegészítőjévé vált. Ennek oka egyrészt az, hogy a multimodal-azonosítás nagyobb biztonságot ad (kisebb téves visszautasítás, nagyobb szelektivitás), továbbá ha az elsődleges az ujjlenyomat-vizsgálat, akkor alternatív azonosítási módszert biztosít.

A szemüveg és a kontaktlencse sem zavarja, a kép fekete-fehér. A template öregedését egy egyszerű újbóli felvétellel kompenzálni lehet (mert csak keveseknél van öregedés, betegség: hályog, cukorbetegség) Fontos szempont, hogy az elkészült kép elég részletes legyen ahhoz, hogy az azonosításhoz használt jellegzetességek jól láthatóak legyenek.

Az íriszdiagnosztika segítségével a mintát adó egészségi állapota is felmérhető. Pupillamérés segítségével pedig kimutatható az emberről, hogy fáradt, fogyasztott-e alkoholt, használt-e drogot.

Az íriszazonosítás legfontosabb előnye a gyorsaság mellett a pontossága – a szivárványhártya stabilitásának köszönhetően –, mivel a szem belső, védett, mégis kívülről látható szervünk.
Számos ország használja az írisz-azonosítást. Sőt szerte a világon kényelmi okokból már több millióan szerepelnek egy íriszazonosítási rendszerben, amely segítségével útlevél nélkül átkelhetnek határokon.

Érdemes keresni az ujjlenyomat-vizsgálat mellé egy kiegészítő rendszert, a megfelelő szintű biztonság elérése végett. Ennek kiválasztásakor fontos szempont a könnyű használhatóság, az elfogadható költségszint és természetesen a megfelelő funkcionalitási szint. Érdemesnek tartom a vizsgálatok irányát az RFID technológia irányába fordítani. Ennek használatával megvalósítható lenne egy többszintű azonosítási rendszer, ami a normál szintű autentikációt RFID-val végezné, a kiemelt szintű beléptetés pedig az ujjlenyomat-vizsgálat és az RFID együttes használatával történne.

AUER RICHÁRD *–TÓTH ROLAND **–BURKUS ERVIN ***

Mobilrobot elektronikai és szoftveres fejlesztése

Összefoglalás: Napjainkban, a robottechnológia gyors fejlődése, magával hozta az azzal kapcsolatban álló szoftverek fejlődését is. Különböző szimulációs szoftverek születtek, több témakörben, különböző algoritmusok készültek térképezésre (mapping) és helymeghatározásra is. Ezáltal inspirálva, építettünk két vonalkövetésére alkalmas robotot. Az ezzel szerzett tapasztalatnak és sikerélménynek köszönhetően, a 2015-ös tavaszi félévében, elkezdtük tovább bővíteni a korábban megszerzett tudást és mélyebben elmerültünk a beágyazott rendszerek témakörében.
Kulcsszavak: Robottechnológia, szoftverek fejlődése, szimulációs szoftverek, beágyazott rendszerek.

Abstract: The quick development of robotics has brought about the development of the software related to it. Different simulation software types have been developed in several fields, and different algorithms have been written for mapping and positioning. Inspired by this, we have built two robots suitable for following lines. As a result of the experience gained through it and thanks to the sense of achievement it has given us, we started to expand the knowledge previously gained in the spring of 2015, and absorbed ourselves more deeply in the world of embedded systems.
Keywords: Robotics, software development, simulation software, embedded systems.

* *Dunaújvárosi Főiskola,*
E-mail: rauer6@gmail.com

** *Dunaújvárosi Főiskola,*
E-mail: halesz89@gmail.com

*** *Dunaújvárosi Főiskola,*
Informatikai Intézet
E-mail: burkus@mail.duf.hu

Bevezető

Eme dolgozat elkészítése két főágon haladt. Az egyik része a robothardverről, illetve a -szimulációról szólt másik fele pedig a számítógépes vezérlőszoftver tervezéséről és implementálásáról. Egy robot megépítése bonyolult feladat, és a hozzá tartozó vezérlőszoftverrel összehangolni, már csak kommunikáció szintjén sem egyszerű. Ugyan a robot már elkészült, a további fejlesztések érdekében fontosnak tartottuk egy szimuláció elkészítését is. Ezt a V-rep nevű szimulációs programban valósítottuk meg.

A robot szerkezeti kialakítását tekintve a legpraktikusabb lehetőséget választottuk, ami egy differenciálisan hajtott robotot eredményezett. Ebben az esetben két hajtott kerék és egy segédkerék található a roboton. A robot vezérlését a sokak által ismert Arduinóval valósítottuk meg, mivel számos kiegészítőt lehet hozzá beszerezni, valamint a programozási nyelve is C-hez közeli, ezáltal ismert volt számunkra. A robotot négy infravörös távolságérzékelővel láttuk el annak érdekében, hogy a fizikai környezetét, megfelelően felmérhesse. A kommunikáció megvalósításához az Arduinóhoz kapható Bluetooth-modult választottuk.

A robot és az ember közötti interfészt megvalósító, számítógépen futó program mellett a robot beágyazott rendszere is tartalmaz egy összetettebb szoftvert, mely a robot önálló működését (barangolását) valósítja meg.

Az Arduino

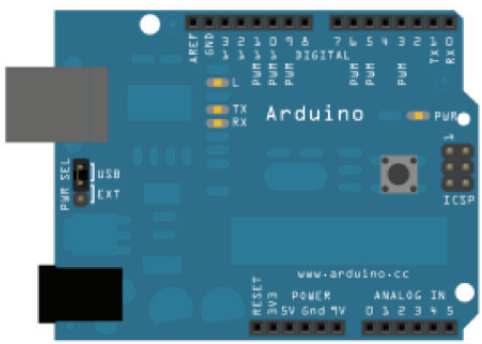
Az Arduino-áramkör egy, a GPL-licenc alá eső termék (szabad hozzáférésű), egy nyílt „physical computing” platform, amely egyszerű mikrovezérlő panelből, és a hozzá szükséges szoftver megírására szolgáló fejlesztői környezetből áll, amelyet elsősorban termékfejlesztők és amatőr programozók használnak. Saját programozási nyelvvel rendelkezik, mely a Processing-ből kifejlesztett Wiring egyik implementációja [1].

[1] Margolis, M. (2011): *Arduino Cookbook*. Massachusetts: O'Reilly Media.

Az Arduino felépítése

Az Arduinót USB-s soros porton keresztül kapcsolhatjuk a számítógéphez. A fejlesztési koncepciója egységes azonban számos kiadása létezik (UNO, DUE, NANO stb.). A „board" fejlesztése során a tervezők gondoltak, az oktatási felhasználásra, ezért ellátták mindenféle védelemmel úgymint, rövidzárlat elleni védelem, fordított bekötés stb.

1. ábra. Arduino felépítése.



Az Arduino tápellátását kaphatja USB-porton keresztül vagy önálló tápegységről esetleg telepről, utóbbi lehet akár egy 9V-os elem is. A megírt programunkat, USB-n keresztül tudjuk feltölteni az Arduinóra, nem felejtő memóriával rendelkezik, így tápellátás megszűnése után a programunk megmarad a „board"-on, mindaddig amíg újra nem írjuk azt [2].

A fenti ábrán is látszik, hogy a „board" egyik oldalán digitális a másik oldalán pedig analóg csatlakozásokat, úgynevezett lábakat találhatunk. Ezeket az egyszerűbb azonosíthatóság érdekében beszámozták, digitális lábak 0–13-ig, az analógok pedig 0–5-ig vannak számozva. Fontos megjegyezni, hogy mindegyik lábat használhatjuk digitális lábként (az analógokat is), azt a programkódban található függvényekkel állíthatjuk be.

A digitális jeleket legegyszerűbben úgy definiálhatjuk, mint egy kapcsoló által előállított elektronikus jel. Ha tehát a kapcsoló bekapcsolva van, átfolyik rajta áram (ezt az állapotot 1-essel jelöljük), ha nincs bekapcsolva, akkor

[2] Banzi, M.–Shiloh, M. (2014): *Make: Getting Started with Arduino*. Gravenstein: Maker Media Inc.

nem folyik áram (ez a 0-ás állapot). Ezeket a folyamatokat egyesek és nullák sorozatával írhatjuk le, melyet szokás bináris jelrendszernek hívni. Az Arduino digitális jeleit könnyen programozhatjuk kifele, illetve befele is. Programozhatjuk kifele, ekkor például jelet küldünk a külvilág felé, ez lehet egy egyszerű LED bekapcsolása, ugyanakkor kiolvashatunk jeleket külső szenzorokról, gombokról.

A „board" másik oldalán található analóg lábak segítségével az Arduino képes analóg jelek fogadásra is. Analóg jeleknek nevezzük az olyan folyamatos jeleket, amelyek időben bármilyen értéket felvehetnek. Az analóg kifejezés arra utal, hogy a végbemenő változás *ok és okozat* összefüggése, tehát például egy hőmérő kijelzőjén az érték függ a környezeti hőmérséklettől és annak függvényében változik.

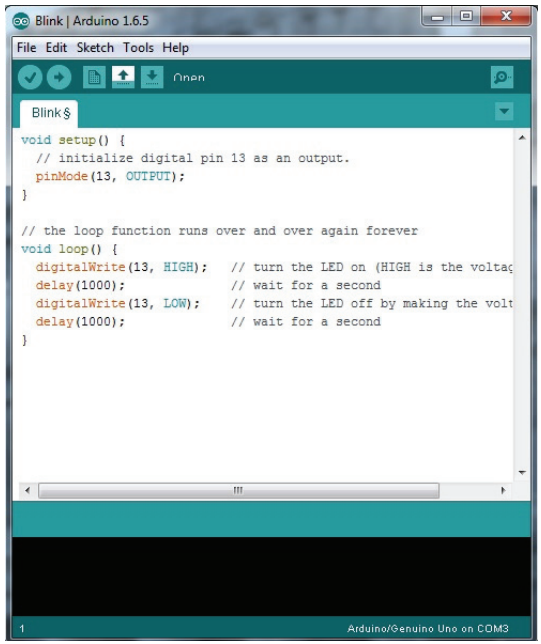
Az analóg jeleknek fontos szerepe van, ugyanis ezek segítségével tudjuk mérni/figyelni a környező világ tulajdonságait, úgymint távolság, mágneses tér, hőmérséklet, fényviszonyok, stb.; ezeket a beérkező jeleket analóg jelsorozatként tudjuk feldolgozni.

Az Arduino hardveres és szoftveres bemutatása

Szoftveres oldal:

Néhány szóban érdemes beszélni az IDE-ről. Angolul Integrated Development Environment, magyarul ez annyit tesz, integrált fejlesztői környezet. Elengedhetetlen a gyors alkalmazás-fejlesztések terén. Ezek az IDE-k tartalmaznak rendszerint egy szövegszerkesztő részt, melyben a programunkat írhatjuk, valamint van bennük egy fordító program, de ezenkívül vannak IDE-k, melyek tartalmaznak fordításautomatizáló eszközöket is. Ilyen IDE-vel rendelkezik az Arduino is, a fejlesztők számos tesztprogramot helyeztek el a fejlesztői környezetben, melyek segítségével könnyen és gyorsan elsajátíthatjuk az alapokat.

2. ábra. Arduino IDE- és a LED-villogtató programkód.



Ezeket a programjainkat C/C++ nyelven írhatjuk, így a nyelvet ismerők könnyen igazodnak ki az Arduinós programok fejlesztésében. Az IDE-hez tartozik egy Wiring-könyvtár, amivel könnyen végezhetjük el az alapvető input/output műveleteket.

Hardveres oldal:
Amint azt a történeti fejezetben is említettük több típusa van az Arduino-családnak, ezek leginkább memóriában, ki- és bemeneti csatlakozásokban, valamint számítási kapacitásban különböznek egymástól. Vannak olyan „board”-ok, melyek beépített Ethernet-csatlakozóval készültek, mások pedig beépített Bluetooth modult kaptak. Mivel az Arduino-családnak számos tagja van, értelmetlen lenne mindent megvizsgálni és elemezni. A robot UNO-val készült, így néhány sorban érdemes a paramétereit megemlíteni. UNO esetében is kétféle mikrokontrollerről beszélhetünk, az egyik a sima DIP IC-vel szerelt változat, a másik pedig egy SMD-s kiadás, ez még nem akkora különbség, tudásilag ugyan azt az ATmega 328-s mikrokontrollert kapta, mely rendelkezik 14 digitális ki- és bemeneti lábbal, valamint 6 analóg lábbal. Ezeken kívül kapott

egy 16 MHz-es kristály oszcillátort egy USB-csatlakozót, illetve egy tápcsatlakoztatási lehetőséget.
Amiben lényegesen eltér az UNO a többi társától, az az, hogy nem használ FTDI USB-drivert, melyet külön kéne telepítenünk, helyette kapott egy programozott USB-ből soros portba konvertáló chipet (mely az Atmega16U2 nevet viseli).

Néhány technikai adat:

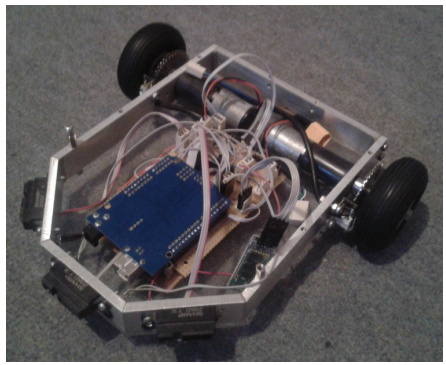
– Mikrokontroller:	ATmega328
– Működési feszültség:	5V
– Bemeneti feszültség:	7-12V
– Bemeneti feszültség limit:	6-20V
– Digitális I/O lábak száma:	14 db
– Analóg lábak száma:	6 db
– Flash memória:	32Kb
– ESRAM:	2Kb
– EEPROM:	1Kb
– Órajel sebesség:	16MHz

Motorvezérlő áramkör

Az Arduino nemcsak, ilyen fejlesztői »board"-okat hanem egyéb, kiegészítő, úgynevezett shieldeket is gyárt/árusít. A robot építése során felmerült két lehetőség: vagy megépítjük saját magunk a robot motorvezérlő paneljét, vagy pedig megvesszük azt készen. Mivel egy késztermék esetén a tapasztalatszerzése kevés, ezért maradtunk annál a lehetőségnél, hogy sajátot tervezünk, majd építünk.

Két prototípus is készült a robotból, eltérő különbségekkel (más mikrovezérlő, különböző alkatrészek stb.). Ezután építettük meg a mostani állapotába.

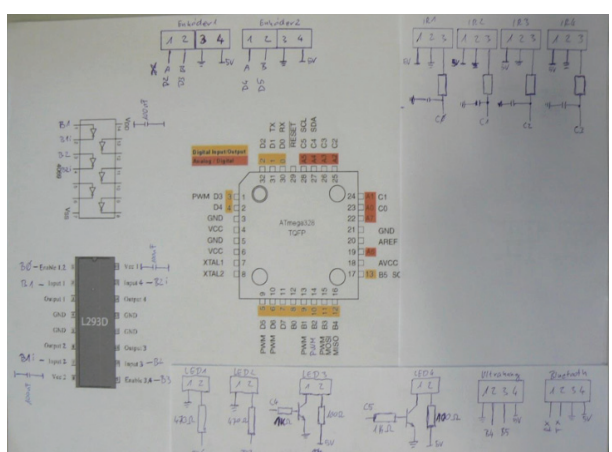
3. ábra. A megvalósított robot.



Az áramkör tervezése

Az áramkör tervezését igen egyszerűen valósítottuk meg, komolyabb programot nem igényelt, ugyanis a nyáklemezünket nem maratással készítettük, hanem csak egyszerűen szigetelt vezetékekkel a lemez hátoldalán.

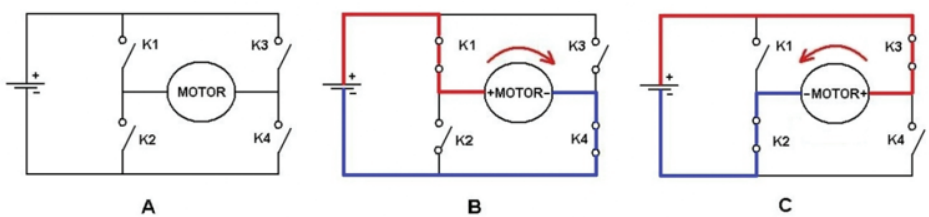
4. ábra. Áramkör sematikus terve.



[3] Szittya Ottó (1999): *Digitális és analóg technika informatikusoknak I–II.* Budapest: LSI Oktatóközpont.

Az ábrán jól látszik, hogy kezdetnek vettük az Arduino-chip lábkiosztását, majd felrajzoltuk a csatlakozókat, (kis téglalapok) és bejelöltük, hogy mi hova kapcsolódik. A tervezés közben felmerült, hogy nem csak a motorvezérlő IC-ket, hanem a csatlakozási pontokat is egy panelre tesszük, ezáltal nem csak egy motorvezérlő shieldet, hanem egy olyan panelt kaptunk, ahova egyszerűen csak csatlakoztatni kell a bluetoothot, infrákat, ledeket, stb. Ez pedig egyszerűen az Arduinóra dugva (egy tüskesor segítségével), „bővítőkártyaként" funkcionál. Mind a motorok, úgy a »board" is egy közös akkumulátorról kapja a tápellátását, ez egy 7.4 V-os, 1800 mA órás lítium polimer akkumulátor.

Az L293D motorvezérlő IC, H-híd kapcsolásnak felel meg, azzal a különbséggel, hogy elektronikus kapcsolókat használ, amelyek az IC megfelelő lábaira adott 0 V-os vagy 5 V-os vezérlőjelekkel nyithatók és zárhatók. Ezáltal a H-híd IC könnyedén vezérelhető a mikrokontrollerrel. Az L293D két H-hidat tartalmaz, ezért egyetlen IC elég a robot két motorjának a vezérléséhez.



Az integrált áramkörökkel történő fejlesztés során felhasználtuk a *Digitális Technika* tantárgy keretein belül, korábban megszerzett tudásunkat [3].

Felhasznált szenzorok bemutatása

A robotra felszereltünk 4 darab infra szenzort, illetve 2 darab enkódert. Az infrákat a barangoló algoritmus megírásakor használtuk fel, az enkóderek pedig a robot sebességét és a megtett távolságát hivatott mérni. A robot képes manuális, illetve automatikus módon működni. Automatikus mód esetén a robot rendelkezik egy barangoló algoritmussal.

6. ábra. A használt szenzor.

A tervezés folyamán felmerült, hogy a robot barangolási folyamatát ultrahangszenzorral valósítjuk meg. Azonban ehhez szinte tökéletes felület kell, mert különben nem érzékel rendesen. A felhasznált infra-szenzor 10-től 80 cm-ig érzékel.

Az enkódereket legtöbb esetben a megtett távolság meghatározásra használják. A tárcsa meghatározza az információt, amit az enkóder ad a vezérlésnek. Kétféle információtovábbítási lehetősége van az enkódereknek. Az egyik az abszolút, ez esetben egy olyan kódot kapunk, mely megmondja a kerék pozícióját, az aktuális szöget. Például ilyenkor maga a tárcsa szolgáltatja az információt. A relatív vagy inkrementális enkóderek pedig azt mondják meg, hogy a kerék milyen mértékben lett elforgatva, itt maga a vezérlés, az elektronika számolja és mondja meg az elfordulást, de ebben az esetben általában alkalmaznak egy referencia pontot, illetve egy plusz érzékelőt, amely a teljes elfordulásnál ad csak jelet.

A robot esetében az egyszerűbbik tárcsát, az inkrementálist használtuk. Az inkrementális működésű enkóderek egyszerűbbek az abszolúthoz képest, kevesebb információt szolgáltatnak szögelfordulás és fordulatszám mérésére használhatóak. A tárcsa rovátkolásának sűrűsége meghatározza a felbontást.

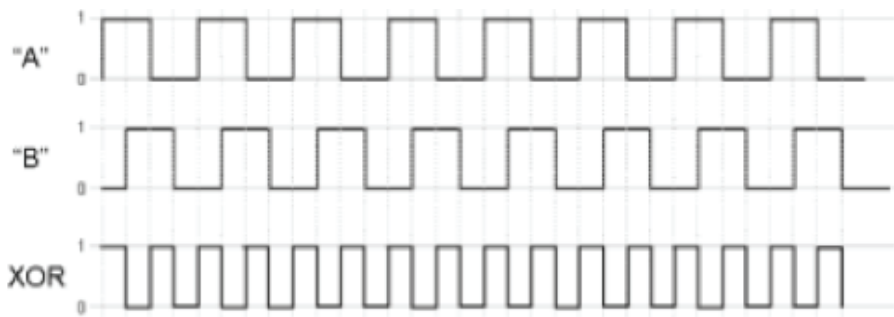
7. ábra. Inkrementális enkóder tárcsa.

4] Bräunl, Th. (2003): *Embedded Robotics*. München: Springer.

[5] Coppelia robotics. *Virtual Robot Experimentation Platform*. USER MANUAL, 2015. http://www.coppeliarobotics.com/helpFiles/

Először is van egy kapukkal felépített logika, ami az „A” és „B” jel fázishelyzetéből megállapítja a forgás irányát. Ezután az „A” vagy a „B” jelet egy le/fel számláló bemenetére vezetik. A számlálás irányát a forgási irányt megállapító logika kimenete vezérli. Ha az enkóder tengelye előre forog, akkor az impulzusok tartalmát növelik. Ha hátra forog, akkor csökkentik. Valójában a számlálót nem az „A” vagy „B” jel lépteti, hanem beiktatnak egy EXOR-kaput, ami összehasonlítja az „A” és „B” jelét [4].

8. ábra. Enkóder működése.



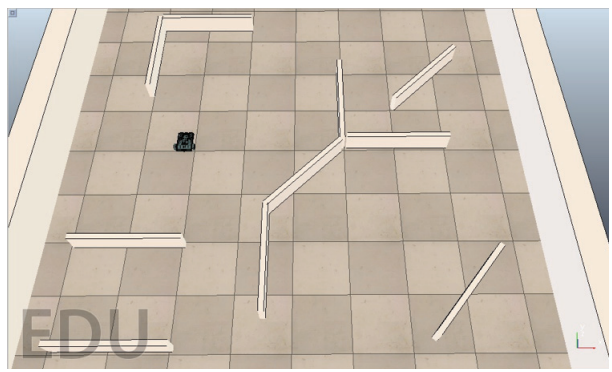
V-rep szimulációs környezet

A V-rep szoftver segítségével komplett robotszimulációkat tudunk megvalósítani számítógépen anélkül, hogy fizikailag megépítettük volna a robotunkat, ezzel pénzt és időt spórolhatunk. Néhány esetben a megtervezett és leszimulált robotunk áttehető egy az egyben a valós robotba, amit megépítettünk, minden módosítás nélkül. Ezeknek a szimulátor-szoftvereknek van egy virtuális robotja, amivel a valós munkában dolgozó robotokat is le tudjuk szimulálni. A szimulátorokkal/szoftverekkel kényelmessé válik a robotok programozása, illetve javítása egészen a végleges verzióig, miközben tesztelni tudjuk őket. A sikeres programozás természetesen függ a szimulált környezet illetve a valós környezettől is, mivel sokszor a valódi környezet, eltér a szimulált környezettől, ezáltal robotunk másképp viselkedhet abban a helyzetben [5].

A robot V-rep szimulációja

A szimulációra egy teljesen egyszerű konstrukciót használtunk, két hajtott kerék valamint egy segédkerék segítségével mozog a robot.

9. ábra. Az akadályok és a robot.



A robot hamarabb elkészült, mint a szimuláció, azonban fontosnak tartottunk elkészíteni egy szimulációt is, az esetleges későbbi tesztelés érdekében. Az Arduino fejlesztői környezetében C/C++ közeli nyelv van, ezáltal ott az említett nyelven könnyebben boldogulnak a programozók. V-repben kicsit más programozási nyelv található, ez pedig a LUA nyelv. Ez utóbbi is hasonló a C/C++-hoz, azonban még is vannak némi eltérések. Többnyire például saját függvényekkel rendelkezik a program, melyek segítségével mozgathatunk egy csuklót, beolvashatunk adatokat különböző szenzorokról, vagy akár „életre" kelthetünk egy egyedileg készített testet. Néhány főbb függvényt érdemes megemlíteni, mind V-rep, mind pedig az Arduinós fejlesztői környezetből és összehasonlítani ezeket.

Barangoló algoritmus bemutatása

Barangoló vagy ismert nevén akadálykikerülő algoritmus, robotok esetében elég hasznos és gyakran használt algoritmus szokott lenni. Mivel csak alapvető funkcióról beszélünk lényege annyi, hogy robot képes legyen akadályokat kikerülni egy algoritmus segítségével.

10. ábra. Robot és a szenzorok.

Ahogy a *10. ábrán* is látszik a legegyszerűbb szenzorelrendezés ha a robotunk elejére teszünk három szenzort három irányba balra, jobbra és előre. Ezek a szenzorok fogják érzékelni az akadályokat. Ezek a szenzorok lehetnek akár ultrahangos, vagy ha jobb minőségben szeretnénk az adatokat megkapni és feldolgozni, akkor komolyabb infrás szenzorokat is használhatunk, erre a célra. Az ábrán jelöltük a motorokat, illetve a szenzorokat. Legegyszerűbben egy igazság-táblán lehet bemutatni, szemléltetni az algoritmus működését.

Bal	Első	Jobb		M1	M4
1	1	0	Jobbra	1	0
0	1	1	Bal	0	1
1	0	1	Előre	1	1
0	0	1	Balra	0	1
1	0	0	Jobbra	1	0
1	0	1	Előre	1	0

Szenzoroknál a jelölés:
0 -> Nincs objektum a szenzor előtt.
1 -> Van objektum a szenzor előtt.

Minden eshetőséget meg kell vizsgálni, az igazság-táblán végighaladva írjuk le az algoritmus működését.

Ennél az algoritmusnál elindítjuk a robotunkat és ha az első szenzor akadályba ütközik, akkor vagy bal, vagy jobb irányba haladunk tovább. Ha a két oldalsó szenzor érzékel falat tehát a hatodik eset áll fenn akkor a robotunkat előre hajtjuk. Ha a szenzorokat minden irányból akadály veszi körül akkor – tehát egy zárt helyre jut – ebben az esetben tolatni kell a robottal [6].

[6] Russel, S.–Norvig, P. (2005): *Mesterséges intelligencia modern megközelítésben.* Budapest: Panem.

Szoftver-specifikáció, feladatelemzés

A szoftver fejlesztésénél az első szempont az volt, hogy megvizsgáltunk két lehetőséget a vezérlést illetően. Az első lehetőség egy olyan PC-s szoftver lett volna, ami a robottól kapott adatok alapján, vezérelte volna a robotot. Ezáltal elkerülhető, hogy a választott beágyazott rendszerünk le legyen terhelve, az csak az adatokat továbbítja a számítógép felé. A második megoldás pedig ennek az ellentéte, vagyis a beágyazott rendszerünk dolgozza fel a beérkező jeleket, és avatkozik be a környezetébe és csak a lényegi információkat küldi tovább a PC oldali vezérlőszoftvernek. A következőkben ennek a két megoldásnak a pro/kontra összehasonlítását fogjuk bemutatni. [7]

[7] Troelsen, A.: *A C# 2008 és a .NET 3.5 1. kötet.* Bicske: Szak kiadó.

PC-s szoftver általi vezérlés

A PC-s szoftveres vezérlés nagy előnye, hogy a számításigényesebb feladatokat vezérlőprogramunk számolja ki, és nem kell azzal a robot mikrovezérlőjét terhelni. További előny hogy ezáltal több szenzort tudunk felrakni a robotra, így még több információ szerzése lehetséges a környezetről. Az egyik szembetűnő veszélye ennek a fajta vezérlésnek, hogy amint a kapcsolat megszakadt a robottal, akkor elvész a kontroll a robot felett.

Természetesen vannak módszerek ennek a problémának az áthidalására, az egyik ilyen egy állandó „küldtem-vettem”-jelentés a robot és a szoftver között. Amint elmarad a visszaigazolás a szoftvertől a robot megáll, ott ahol van.

Szoftver, mint információt közlő program

Ezen módszer lényege, hogy minden, környezetről bejövő információt a beágyazott rendszerünk dolgoz fel, az információt csak elküldi a PC-s vezérlőszoftvernek, de a mikrovezérlő szoftvere, semmilyen algoritmust nem használ a kapott adatok számolására. Csak az információt jeleníti meg: ilyenek a szenzor értékek, futás-naplózása, esetleg a fent említett „vészkapcsoló” funkció

Szoftver specifikáció konklúzió

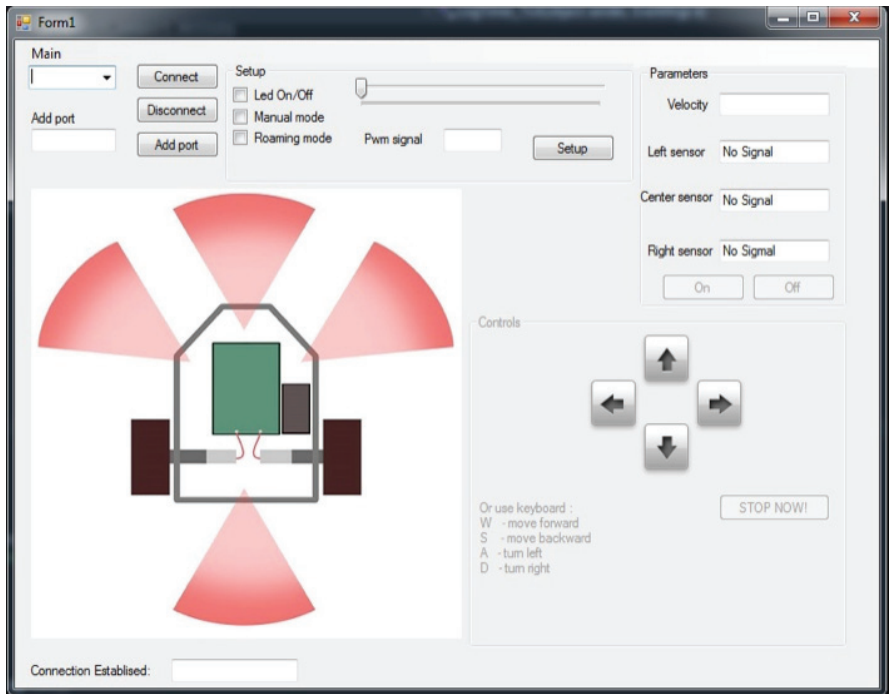
Az általunk választott módszer az utóbbi, nincs annyi szenzor a roboton, ami indokolná a PC-s szoftver általi vezérlést, és a feldolgozandó adatokkal sem kell annyi számítást elvégezni.

A másik indok a választott hardver egy Arduino Uno mikrovezérlő, már meglévő IDE-vel, amivel ugyan gyorsan lehet fejleszteni, nem kell saját vezérlőt tervezni, de sajnos ennek a hátránya a kompromisszumok, a terhelhetőséget illetően.

Felhasználói igények

A szoftvertervezési fázis következő lépése volt, annak a feltárása, hogy a programnak milyen feladatokat kell megvalósítania, milyen igények vannak vele szemben. Ezek az igények listázva a következők voltak:
– felhasználó csatlakozni tudjon az általa kiválasztott soros-porton keresztül a robothoz,
– a program induláskor kérje le az elérhető portokat, és lehetséges legyen a port hozzáadása,
– a programnak felhasználó-biztosnak kell lennie „kivételkezelések”-kor,
– csatlakozáskor a programban lehetséges, a robot beállítása, kétféle mód közül lehet választani „barangolás” és „manuális irányítás” illetve a LED-világítás bekapcsolása,
– az infraszenzor által küldött adatok megjelenítése,
– a program, a lehető legkevesebb mértékben függjön a saját funkciójától, későbbi fejleszthetőséget szem előtt tartva,
– a program folyamatosan figyelje a kapcsolat állapotát, megszakadás esetén értesítse a felhasználót,
A GUI tervezésekor a több részre osztottuk a felületet. A következő részeket hoztuk létre:
– legfelül bal oldalt helyezkedik el a port kiválasztását, csatlakozást, és lecsatlakozást, valamint port hozzáadást megvalósító felület,
– mellette a „setup” felület, mellyel a robot beállítása, illetve a sebesség beállítása lehetséges,
– legfelül jobb oldalt találhatók a robot paraméterei, ezek a szenzorok által küldött jelekre vonatkoznak,
– bal oldalt a csatlakozást megvalósító felület alatt helyezkedik el egy információs kép a roboton található szenzor-pozíciókról,
– mellette a manuális irányításhoz való ablak.

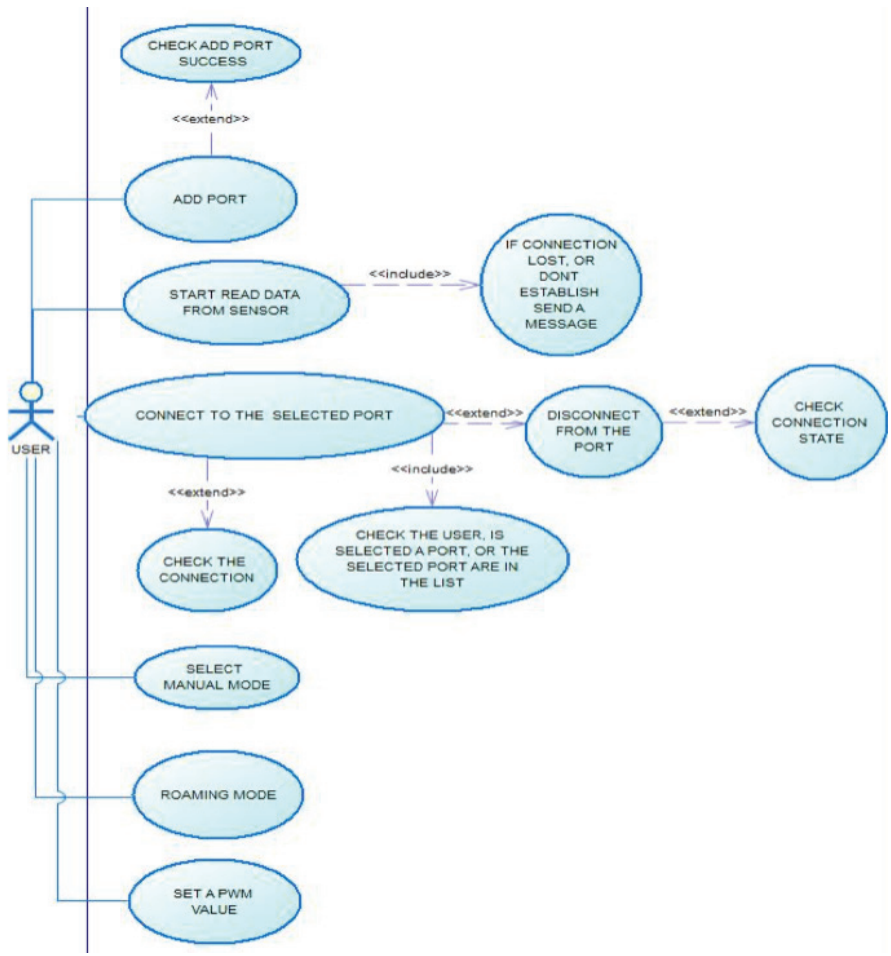
11. ábra. Program kezelőfelülete.



A rendszer use case diagrammja

A program fejlesztése során, a POWER Designer programot használtuk fel.

12. ábra. Program use case diagramja.



Kommunikációs protokoll tervezése

A kommunikációs protokoll megtervezésénél az egyszerűséget tartottuk szem előtt, és fix byteos, kétirányú kommunikációt valósítottunk meg. A kommunikációnak a következő szempontok alapján kellett megfelelnie vezérlőszoftver által küldött parancsoknál [10]:
– robot futási módjának beállítása (akár menet közben is),
– el kell tárolnia a robot aktuális paramétereit és visszaküldeni soros porton,
– fogadnia kell a robothoz érkező üzeneteket.

A csomag felépítése a következőképpen néz ki:
– HEAD (üzenet fejléc) a későbbi változó kommunikációhoz,
– MODE (a robot futási módja, manuális vagy roaming),
– LEDSTATE (LED világítás ki- bekapcsolása),
– M1E, M2E (A motorok állapota, logikai 1-es értéken kap feszültséget, logikai 0-ás értéken nem),
– PWM (a motorsebességének értéke impulzus szélesség modulációval van megvalósítva),
– TAIL (üzenet farok része),
|HEAD |MODE|LEDSTATE|M1E|M2E|PWM|TAIL|.

A robot a szenzorok által mért értékeit, és a robot aktuális beállítását folyamatosan küldi vissza a vezérlőszoftvernek megjelenítésre, illetve naplózás céljából. Ennek a csomagnak a felépítése a következő:
– Bal oldali szenzor értékei.
– Jobb oldali szenzor értékei.
– Középső oldali szenzor értékei.
– PWM érték.
– M1E értéke.
– M2E értéke.

[10] Tannenbaum, A. S.–Wetherall, D. J. (2013): *Számítógép hálózatok*. Budapest: Panem.

Pc-s vezérlőszoftver által kiadott parancsok

A vezérlőprogramba 7 byte-os tömbökben zajlik a kommunikáció a robot és a szoftver között. A fentiekben vázoltak szerint az egyes indexekben a következő értéket veheti fel a tömbünk:

– 0 index Head
  0x10 az üzenet fejléce
– 1 index Mód
  0x2 Manuális mód
  0x3 Roaming mód
  0x00 Azonnali megállás
– 2 index LED Állapot
  0x3 Led bekapcsolva
  0x4 Led kikapcsolva
– 3-4 index M1E, M2E
  0x5 Motor Be
  0x6 Motor Ki
– 5 index PWM
  a tömb az aktuális PWM-et tartalmazza, 0–255-ig vehet fel értéket.
– 6 index Tail
  0x90 az üzenet vége

Osztályok tervezése

A programunk a fentebb említett két fő részből áll. Az első feladat a „RobotParam” nevű osztály megtervezése volt. Az osztályunknak a következőket kell megvalósítania:

– a robot összes aktuális, paraméterének a tárolása, és futás közbeni megváltoztatása,
– a robot felé küldendő csomagok összeállítása.

Az osztály konstruktorának nincs bemenő paramétere. Csak a fontosabb tagváltozókat állítja be.

13. ábra. A RobotParam osztály konstruktora.

```
//Construktor
1 reference
public RobotParam()
{

    mode = 0x00;
    led = 0x00;
    pwm = 0;
    dataS = new byte[7];
    dataI = new byte[7];
}
```

Összefoglalás és továbbfejlesztési lehetőségek

A projekt jelen állapotában még nem végleges, azonban a korábban célul kitűzött mérföldköveket sikeresen megvalósítottunk. Ez meghozta a várt eredményt, és ennek következtében újabb mérföldköveket tűztünk ki. Szeretnénk implementálni például egy SLAM-algoritmust (Simultaneous Localization And Mapping), mellyel a robot képes lesz feltérképezni egy teljesen ismeretlen környezetet. Ezenkívül szeretnénk létrehozni egy belső adatbázist, hogy a később megvalósítandó robotok is kompatibilisek legyenek a PC-s vezérlőszoftverrel. Az egyszerűbb javíthatóság érdekében naplózási funkciókat is szeretnénk megvalósítani. A dolgozatban taglalt témák jó támpontot adnak a további fejlesztésekhez, ezt kiegészítve a megszerzett tapasztalattal megvalósításra kerülhetnek a fent említett elképzelések.

Az általunk létrehozott projekt biztos alapot ad a témával most ismerkedőknek, örömmel vettük észre, hogy van is érdeklődés a téma iránt. Egyszerűsége és átláthatósága miatt könnyen elsajátíthatóak és felhasználhatóak a témában kifejtett és megvalósított módszerek. Az iskolában oktatott elméleti tárgyak (Digitális Technika, Jelek és rendszerek, Számítógép és távközlési hálózatok) hasznosak voltak a projekt készítése során. Az elkövetkező fejlesztésekbe szeretnénk bevonni hallgatótársainkat is, hogy az oktatás során megszerzett elméleti tudást ők is kiegészíthessék a megfelelő gyakorlati tapasztalatokkal.

TÓBEL IMRE *– KŐVÁRI ATTILA **

LabVIEW Real Time valósidejű mérő- és irányítórendszer

I. Real Time Target PC-alapú tesztkörnyezet létrehozása

Összefoglalás: Irányítórendszerek fejlesztésekor nagy segítséget nyújt egy olyan fejlesztői tesztkörnyezet, amely révén egy komplex irányítórendszer prototípusa viszonylag egyszerű módon felépíthető és tesztelhető, méghozzá az irányítóberendezésektől elvárt valósidejű működést figyelembe véve. A LabVIEW és a LabVIEW Real-Time Module segítségével ilyen tesztkörnyezet egyszerűen felépíthető, komplex irányítási feladatok tesztelhetőek, egy ideális környezet a gyors prototípusfejlesztéshez. Jelen cikk egy PC alapú költséghatékony tesztkörnyezetet mutat be, mely segítségével valósidejű irányítások is vizsgálhatóak laboratóriumi körülmények között.
Kulcsszavak: LabVIEW, LabVIEW Real-Time Module, prototípusfejlesztés, PC-alapú költséghatékony tesztkörnyezetet.

Abstract: A development test environment provides a great help in case of development of control systems, which can help to build and test a prototype of complex control system in a relatively simple manner, taking into account the required real-time operation. This test environment can be built simple by the help of LabVIEW and LabVIEW Real-Time Module, complex control tasks can be tested, it is an ideal environment for rapid prototype development. This article presents a cost-effective PC-based test environment that enables to examine real-time controls in a laboratory.
Keywords: LabVIEW, LabVIEW Real-Time Module, prototype development, PC-based test environment.

* *Dunaújvárosi Egyetem, Informatikai Intézet*
E-mail: tobeli@uniduna.hu

** *Dunaújvárosi Egyetem, Informatikai Intézet*
E-mail: kovari@uniduna.hu

[1] Katona József (2015): The examination of the application possibilities of brain wave-based control.
In: András–Rajcsányi–Molnár–Németh (Szerk.): *Szimbolikus közösségek.* Dunaújváros: DUF Press. Pp. 143–152.

[2] Katona József (2015): The Comparison of the non-invasive mobile EEG registration and the signal processing devices. In: Honfi–Király–Nagy (Szerk.): *Informatikai terek.* Dunaújváros: DUF Press. Pp. 97–1110.

[3] Molnár György (2008): Az IKT-val támogatott tanulási környezet követelményei és fejlesztési lehetőségei. *Szakképzési szemle.* Pp. 257–278.

[4] Molnár György (2012): A technológia és hálózatalapú alapú tanulási formák és attitűdök az információs társadalomban, különös tekintettel a felsőoktatás bázisára. *Információs társadalom: társadalomtudományi folyóirat.* Pp. 61–76.

[5] Kálmán Anikó (2006): A felnőttoktatásban és -képzésben alkalmazható kompetenciaelvű módszerek és azok alkalmazhatósága. *Felnőttképzés.* Pp. 60–62.

[6] Kálmán Anikó (2008): A kompetencia alapúság a program és tananyagtervezésben. *A felnőttképzés módszertani kérdései: HEFOP 3.5.1 „Korszerű felnőttképzési módszerek kidolgozása és alkalmazása".* Budapest: NSZFI. Pp. 67–136.

Bevezető

A számítógép alapú mérő- és irányító berendezések felhasználási területe igen széles az ipari alkalmazásoktól akár egészen a speciális agy-számítógép interfészek területéig [1, 2].

Számítógép alapú irányítórendszerek működésétől azt várjuk, hogy azok a bemeneti adatok beolvasása/események észlelése után, a számítógépen futó program segítségével feldolgozzák azokat és a feldolgozás eredménye alapján meghatározzák a szükséges kimeneteket, beavatkozásokat.

Folyamatirányításban a folyamat jeleinek változási sebességéhez, bekövetkezett események gyorsaságához kell illeszteni az irányítórendszer feldolgozási időzítését, hogy a folyamat változásaihoz képest a rendszer rövid időn belül/meghatározott időzítés szerint képes legyen a beavatkozásra.

A korszerű infokommunikációs megoldások a megváltozott munkaformák és munkatevékenységek kihívásaira is kellő választ adnak a megfelelő hardveres és szoftveres támogatások mellett. [3]

Az E-rendszerek kellő gyorsasággal és hatékonysággal alkalmazhatóak a mai kihívásokkal teli információs társadalmunkban az ellenőrzési folyamatok támogatására. [4]

Az olyan számítógép alapú mérő- és irányítórendszereket, amelyek esetén a folyamat jellemzői alapján előírt időzítésnek megfelelően, a bemenet észlelésétől, annak feldolgozásától a kimenet előállításáig, valós idejű rendszereknek nevezzük. A következőkben a National Instruments LabVIEW fejlesztőkörnyezet által támogatott valósidejű irányítások tesztelésére alkalmas személyi számítógép alapú rendszer kerül bemutatásra. A LabVIEW, mint grafikus fejlesztőkörnyezet, a programozási kompetenciák elsajátításában is jól alkalmazható nem csak a fiatalabb, hanem akár a programozási ismertekkel nem rendelkező idősebb korosztály oktatásában is, hozzájárulva az élethosszig tartó tanuláshoz.
[5, 6].

Valósidejű rendszerek

A valós idejű rendszerek működésében a válaszidő kritikus tényező. A valósidejű működés időzítésének feltétele az alábbiak lehetnek [7]:
– *abszolút:* amikor a feldolgozást egy meghatározott időpontokban/időpont körül/időpontig/időpont után kell végrehajtani;
– *relatív:* amikor a feldolgozást valamilyen esemény bekövetkezéséhez viszonyítva meghatározott időpontokban/időpont körül/időpontig/időpont után kell végrehajtani.

Az előzőkben leírtak figyelembevételével a valósidejű irányítórendszerekkel szemben általánosan támasztott követelmények az alábbiak szerint foglalhatóak össze:
– előírt időzítés szerint működjön, vagyis minden feldolgozás, válasz determinisztikus legyen;
– egyidejű eseményeket is fel tudja dolgozni (párhuzamos feldolgozás);
– megbízható, biztonságos.

Számítógép alapú folyamatirányító-rendszerek közül – például az ipari folyamatirányításban alkalmazott – PLC-k működése determinisztikus, ami garantálja, hogy a bemenetek beolvasását, a rajta futó programot, a kimenetek beállítását meghatározott ciklusidővel végre tudja hajtani. Azonban a számítógépek operációs rendszerei, melyek az alkalmazásokat futtatják, általában nem felelnek meg a determinisztikus működésnek. Azon rendszerek esetén, melyeknél elvárt a valósidejű működés, úgynevezett valósidejű operációs rendszereket (RTOS: Real-Time Operating System) alkalmaznak, melyek kernele biztosítja a determinisztikus, precíz időzítés szerinti működést.

A valósidejű rendszerek két részre oszthatóak, megkülönböztetünk [8]:
– „hard real-time";
– „soft real-time" rendszereket.

A hard real-time rendszer esetében feldolgozásiidő-túllépés nem megengedett, az időkritikus algoritmusok adott idő alatti futtatása biztosított. Míg a soft real-time rendszereknél az időkritikus algoritmusok csak magas prioritással futnak, ebben az esetben azonban meghatározott mértékben és gyakorisággal el lehet térni a kívánt időzítéstől (Deadline, Desired Loop Time). (*1. ábra*)

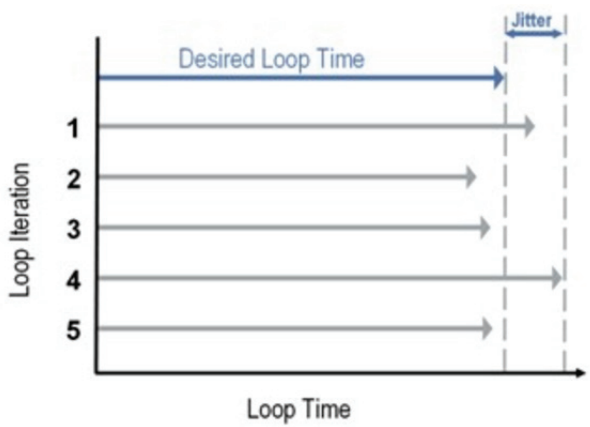
[7] Jancskárné Anweiler Ildikó (2015. 12. 05.): *Számítógépvezérelt irányítások* [Online]. Elérhető: http://e-oktat.pmmf.hu/webgui/www/uploads/1013/szamitogepes_folyamatiranyitas_jancskarne.pdf

[8] *National Instruments* (2015. 12. 05): What is a Real Time Operating System (RTOS)? [Online], Elérhető: http://www.ni.com/white-paper/3938/en/

1. ábra. Hard és Soft Real-Time alkalmazások időzítése (Deadline: feldolgozási határidő). [9]

Az időzítésben bekövetkező hibát „Jitter"-nek nevezik, a valósidejű operációs rendszerek arra vannak optimalizálva, hogy a „Jitter" jelenség minél kisebb legyen. [8] (*2. ábra*)

2. ábra. Egy feladat ciklikus futtatása esetén a futási idő változása. [8]



A valósidejű operációs rendszerek esetén a valósidejű futtatás minél determinisztikusabbá tételéhez kiegészítő egységeket, funkciókat is tartalmaznak, mint például:
– watchdog – válaszidő figyelő, ha a programunk leáll, és így a watchdog számára adott időn belül választ nem tud küldeni, úgy a watchdog a rendszert automatikusan újraindítja;

[8] *National Instruments* (2015. 12. 05.): What is a Real Time Operating System *(RTOS)?* [Online], Elérhető: http://www.ni.com/white-paper/3938/en/

[9] *National Instruments* (2015. 12. 05.): Do I Need a Real-Time System? [Online], Elérhető: http://www.ni.com/white-paper/14238/en/

– determinisztikus adat kommunikációt megvalósító programrészek és a valós idejű program részek között;
– több CPU-maggal rendelkező rendszerek esetén segédprogramok az egyes CPU-magok közötti terheléskiegyenlítés konfigurálásra;
– ciklus végrehajtásához szükséges időzítés szabályozása.

National Instruments LabVIEW Real-Time Module

A LabVIEW Real-Time modul a National Instruments cég által fejlesztett LabVIEW grafikus programozási környezet egy kiegészítése, amely segítségével a LabVIEW környezetben összeállított grafikus programok valósidejű működésű hardverre is letölthetőek, ott futtathatóak. A LabVIEW Real-Time modul többféle valósidejű működésű hardvert támogat, többek között a CompactRIO, CompactDAQ, PXI, képfeldolgozó rendszereket és személyi számítógépeket is.

A LABVIEW Real-Time modul segítségével a LaBVIEW programozási környezetben fejleszthető, tesztelhető, Ethernet kapcsolaton keresztül számos valósidejű futtatást támogató hardverre, az úgynevezett LaBVIEW Real-Time Target-re letölthető. A program fejlesztésére használt számítógépet, amin a LabVIEW fejlesztőkörnyezet fut *Host*-nak, a valósidejű futtatást biztosító hardvert pedig *Target*-nek nevezik. A Targeten futó National Instruments LabVIEW Real-Time operációs rendszer a valósidejű működést az alkalmazás futtatásának pontos időzítésével biztosítja.

A National Instruments LabVIEW Real-Time Module által támogatott hardverek

Az NI LabVIEW Real-Time modul többféle hardver eszközt támogat az alábbiak szerint [10]:
– *PXI (kibővített PCI):* a szabványos ipari PXI-platform (*3. ábra*) egy ipari környezetre tervezett házból áll, valamint szinkronizálás céljából tartalmaz integrált időzítőt és trigger egységet, valamint beágyazott vezérlőt és I/O modulokat is.

[10] *National Instruments* (2015. 12. 05.): Embedded Systems Hardware [Online]. Elérhető: http://www.ni.com/embedded-systems/products/hardware/

3. ábra. Masszív házba helyezett PXI-hardver ipari környezetben történő alkalmazáshoz, valósidejű programok futtatásához.

– *Ipari szabályzó* (*4. ábra*): Az NI ipari szabályzók masszív házban kerülnek elhelyezésre, a ventilátor nélküli házban viszonylag nagyteljesítményű processzor és egy PCI vagy PCIe foglalat is található I/O modulokkal történő bővíthetőség céljából.

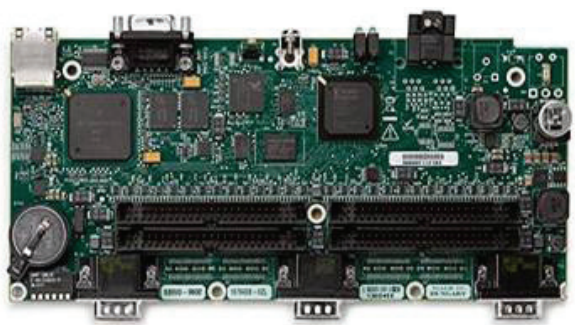
4. ábra. Az NI ipari szabályzó hardver.

– *CompactRIO (kompakt újrakonfigurálható I/O)*: NI CompactRIO (*5. ábra*) lényegében egy valós idejű programok futtatására alkalmas FPGA egységet és I/O modulokat tartalmaz. Az iparban alkalmazott PLC-hez hasonló kialakításúk, nagyobb számítási teljesítmény mellett.

5. ábra. CompactRIO nyújt valós idejű alkalmazásokhoz.

– *Single-Board RIO (Single-Board újrakonfigurálható I/O):* NI Single-Board RIO (*6. ábra*) rendszerek egy áramköri lapon kialakított kompakt vezérlőegységek, melyek architektúrája megegyezik a CompactRIO rendszerrel.

6. ábra. Single-Board RIO hardver.



– *Real-Time Vision (intelligens kompakt kamera)*: NI Smart Cameras (*7. ábra*) ipari nagyteljesítményű képfeldolgozó szenzorokat és a valósidejű képfeldolgozási programok futtatására alkalmas erős processzort tartalmaz, melyek a LabVIEW Real-Time Modul segítségével programozhatóak.

7. ábra. NI intelligens kamera.

– *Ipari, asztali PC* (*8. ábra*): Lehetőségünk van ara, hogy akár általános célú ipari számítógépet, vagy egyéb más single-board számítógépet, vagy akár egy normál asztali számítógépet használhassunk az NI Real-Time környezetében, ha ezek az NI által meghatározott rendszerkövetelményeknek megfelelnek.

8. ábra. Ipari számítógép, asztali számítógép, Single-Board számítógép (SBC) vagy egyéb PC-104 processzor-kártya mint valósidejű hardver.

A személyi számítógépek valósidejű hardverként történő alkalmazása esetén oda kell figyelni arra, hogy a LabVIEW Real-Time Module nem minden hardver eszközt támogat. A támogatott hardver eszközökről részletes információ a National Instruments oldalán található [11].

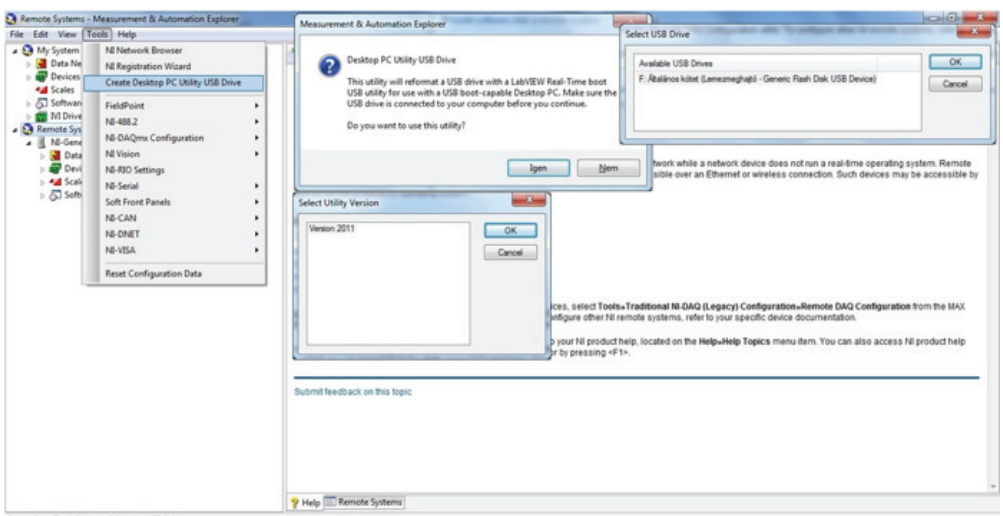
A következőkben a National Instruments LabVIEW Real Time környezete, és az ezen alapuló valósidejű irányítások tesztelésére alkalmas környezet kerül bemutatásra.

[11] *National Instruments* (2015. 12. 05.): Requirements for Desktop PCs as LabVIEW Real-Time Targets [Online]. Elérhető: http://www.ni.com/product-documentation/8239/en/

Valósidejű számítógépes irányítórendszer LaBVIEW Real Time Target alapon

Ahhoz hogy egy személyi számítógép LabVIEW Real-Time Target-ként alkalmazható legyen, a számítógép hardvereinek támogatottnak kell lennie [9]. Egy adott számítógép esetén ez a *Real-Time Desktop PC Evaluator* program segítségével ellenőrizhető. A program a Host számítógépen futó LabVIEW fejlesztőkörnyezet LabVIEW Real-Time Module részeként települ fel, és a Measurement & Automation Explorer (MAX) Tools » RT Disk Utilities » Create Desktop PC Utility USB Drive alkalmazás segítségével egy USB pendrive-ra telepíthető (*9. ábra*). [12]

9. ábra. Desktop PC Utility USD Drive létrehozása.



A Target PC erről indítható, és a megjelenő menürendszerben az *Evaluate system* Amennyiben a számítógép kompatibilis, úgy a képernyő alján az alábbi szöveg jelenik meg: „Tested configuration is compatible with LabVIEW Real Time" [12].

Irányítórendszer-funkció megvalósításához az irányítórendszer jeleit illeszteni kell a Target PC-hez. Erre legalkalmasabb egy multifunkciós adatgyűjtő kártya alkalmazása,

[9] *National Instruments* (2015. 12. 05.): Do I Need a Real-Time System? [Online], Elérhető: http://www.ni.com/white-paper/14238/en/

[12] *National Instruments* (2015. 12. 05.): How Can I Test My PC for Compatibility with LabVIEW Real-Time or LabWindows/CVI Real-Time for Desktop ETS? [Online]. Elérhető: http://digital.ni.com/public.nsf/allkb/9209361E17708D548625744A007FF353

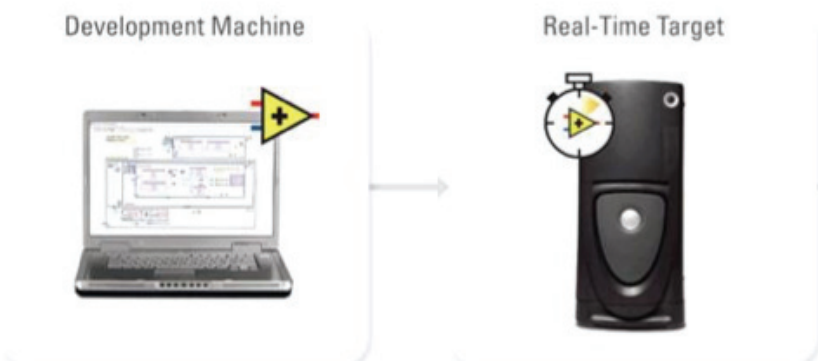
mely az érzékelők/alapjel analóg vagy digitális bemenő jeleit fogadni és konvertálni tudja, valamint a beavatkozó analóg vagy digitális kimenő jeleket is elő tudja állítani. A multifunkciós adatgyűjtő kártyák általában alkalmasak akár inkrementális szöghelyzet jeladó jelének a feldolgozására is.

A következőkben egy multifunkciós adatgyűjtő kártyát tartalmazó, személyi számítógép alapú LabVIEW Real-Time Target kerül bemutatásra, mely általános célú irányítórendszerek fejlesztéséhez, teszteléséhez olcsó, mégis jól alkalmazható rendszert alkot.

Számítógépes irányítórendszer felépítése

A számítógépes irányítórendszer, ahogy korábban említésre került, két fő egységből áll (*10. ábra*):
– Host PC, melyen a szoftver fejlesztés és a futási eredmények megjelenítése történik, egy általános PC, vagy laptop;
– Target PC, melyen a Host PC által Ethernet kapcsolaton keresztül letöltött program valósidejű operációs rendszeren fut, és amely a be- és kimenetek illesztéséhez szükséges A/D, D/A átalakító egységet tartalmazza és hardvere a Real-Time Modul által támogatott kell legyen.

10. ábra. Host fejlesztői PC (Development Machine) és a Target Real-Time PC.



A Target PC-be, az A/D-, D/A-átalakítások, egyéb jelek fogadására, a kompatibilitás miatt egy National Instruments által gyártott multifunkciós adatgyűjtő kártya installálása célszerű, mint például az X vagy M sorozat (*11. ábra*).

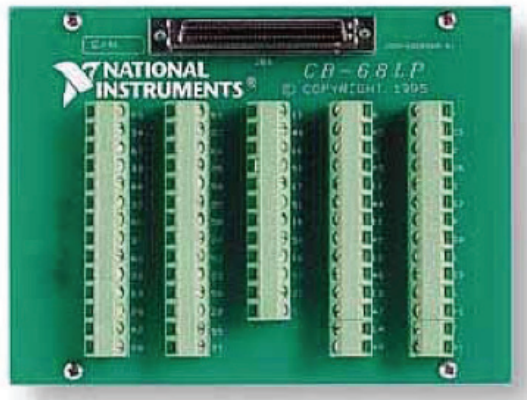
11. ábra NI PCI-6251 M sorozatú multifunkciós adatgyűjtő kártya.

A LabVIEW fejlesztőkörnyezetet futtató Host PC Ethernet hálózaton keresztül kapcsolódik a valósidejű operációs rendszert futtató Target PC-hez. Legegyszerűbb kialakítás esetén a Target PC és a Host PC közvetlenül Ethernet-hálózaton összeköthető UTP-kábel (kereszt kábel) segítségével. Ebben az esetben ahhoz, hogy a két PC kommunikálni tudjon egymással, azonos alhálózatban kell lenniük, fix IP-címmel, mely a hálózati beállításokkal biztosítható.

A Host PC-n a LabVIEW fejlesztőkörnyezetben fejlesztett alkalmazás elindításakor az alkalmazás Etherneten kapcsolaton keresztül automatikusan áttöltődik a valós idejű hardverre, a Target PC-re, és a futás azon elindul. Az alkalmazás futásával összefüggő LabVIEW hagyományos hibakeresési lehetőségei ebben az esetben is használható, mint például a „breakpoint", miközben a program a valós idejű Target PC-n fut.

A kártya használatához az NI-DAQmx illesztő programot telepíteni kell a Host PC-re. A multifunkciós adatgyűjtőhöz egy National Instruments gyártmányú CB-68LP típusú csavarozható kivezetéseket tartalmazó be- kimeneti modul csatlakoztatható (*12. ábra*). Ezen kivezetéseken keresztül köthetőek be az irányítórendszer be- és kimeneti jelei.

12. ábra. NI CB-68LP be-kimeneti csatlakozó modul.



A Target PC hardvere így már alkalmas általános célú irányítási funkciók ellátására, melynek valósidejű működését a LabVIEW Real-Time oprendszere biztosítja.

Számítógépes folyamatirányító egység konfigurálása

A Target PC a korábban létrehozott USB-pendrive segítségével konfigurálható. A LabVIEW Real-Time rendszer betöltésére két lehetőség van:
– *Merevlemez:* LabVIEW Real-Time Format Disk lefuttatása után az USB flash meghajtó segítéségével a rendszert a merevlemezre is telepíthetjük. Miután a telepítés befejeződött, előnyösebb letiltani az USB-t a BIOS-ban.
– *USB meghajtó:* a Real-Time rendszert USB meghajtóról indíthatjuk. Ebben az esetben a LabVIEW-t nem kell telepíteni a merevlemezre, de a merevlemezt le kell formázni.

A Target PC-n a Hyper-Threading-et le kell tiltani a BIOS-ban. Ha mégis bekapcsolva marad, a Real-Time operációs rendszer több processzor magot érzékelhet és instabillá válhat. Hagyományos USB-t engedélyezni kell a Desktop PC Utility USB meghajtó elérése céljából. Merevlemez USB emuláció módot kell beállítani az USB meghajtók felismeréséhez.

[13] *National Instruments* (2015. 12. 05.): Converting a Desktop PC to a LabVIEW Real-Time Target [Online]. Elérhető: http://www.ni.com/tutorial/2733/en/

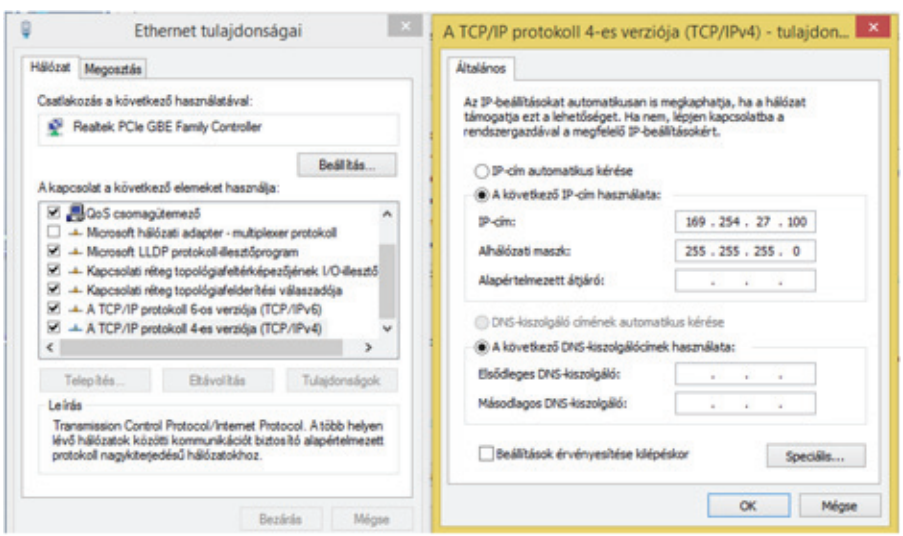
A nem használt egységeket célszerű a BIOS-ban letiltani. [13]

Első lépésben a Windows operációs rendszert futtató Host PC-re történt a LabVIEW programcsomag feltelepítése, mely magában foglalja a Measurement & Automation (MAX) alkalmazást, aminek segítségével telepíthetjük, illetve beállíthatjuk a Target PC-t. A Desktop PC Utility USB Drive segítségével a Target PC elindítható [13].

Ahhoz, hogy a Target PC-hez a Host PC kapcsolódni tudjon, azonos alhálózatban kell lenniük. Mivel a Host és a Target közvetlenül volt Ethernet-kapcsolattal összekötve és DHCP szerver nem üzemelt.

Azért, hogy a Host és aTarget kommunikálni tudjon a Host PC-n a Target PC-vel azonos alhálózatban lévő címet kell megadni. Például ha a Target PC IP-címe 169.254.27.XX, akkor a Host PC-n ugyanebben az alhálózatban lévő fix IP-címet kell beállítani (*13. ábra*).

13. ábra. Host PC hálózati beállításai.

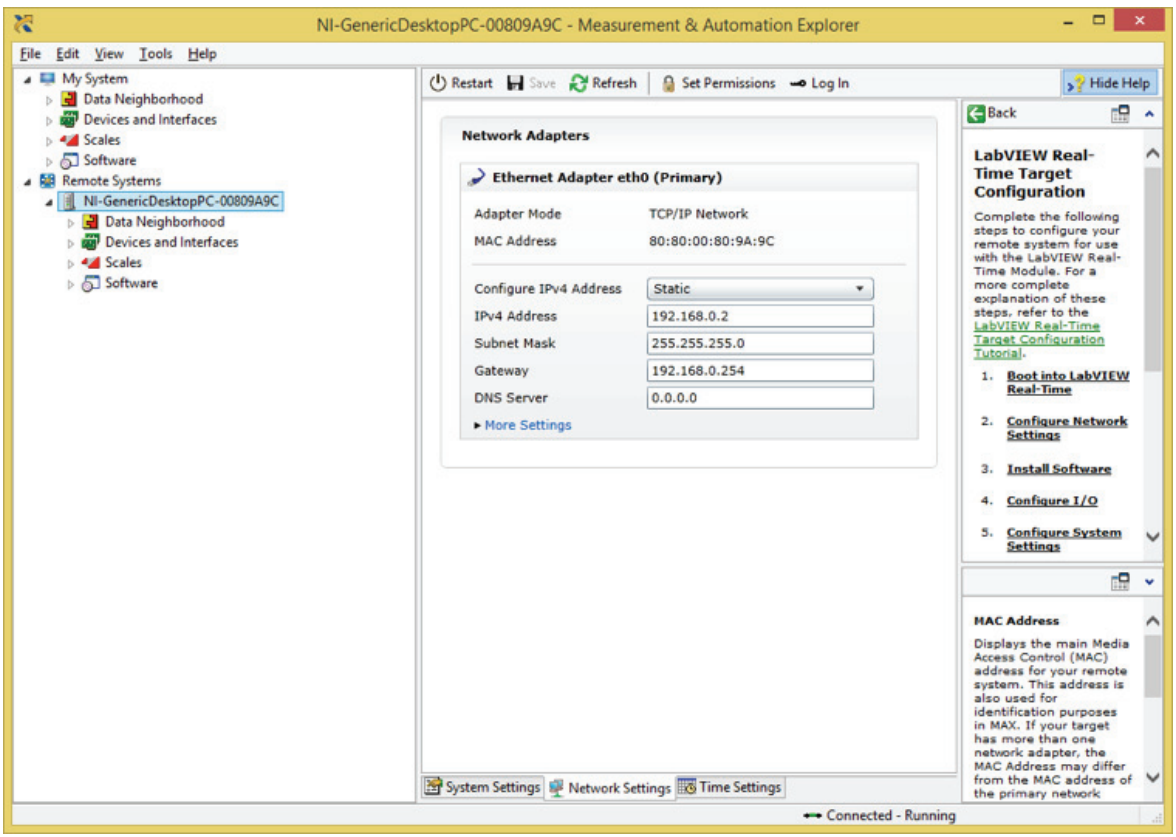


A beállítás után a MAX-ban a Remote Systems listában megjelenik a Target PC, mely paraméterei konfigurálhatóak.

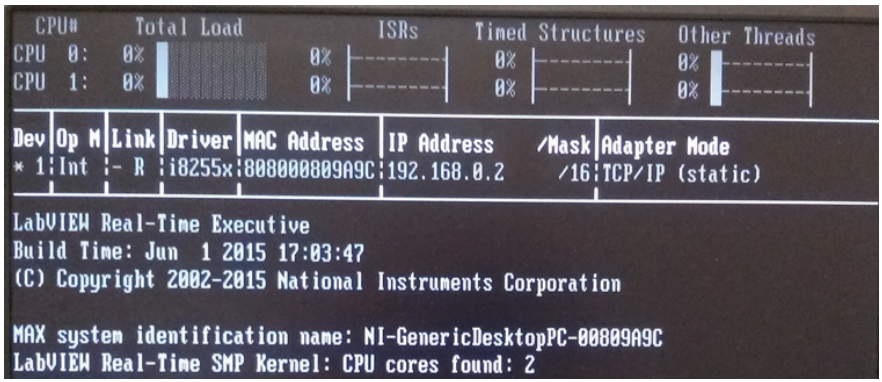
[13] *National Instruments* (2015. 12. 05.): Converting a Desktop PC to a LabVIEW Real-Time Target [Online]. Elérhető: http://www.ni.com/tutorial/2733/en/

A Target PC számára célszerű a belső hálózatokban használt C osztályú statikus IP-cím beállítása, például 192.168.0.2, valamint 255.255.255.0 alhálózati maszk, melyek a Target PC újraindításával lépnek életbe (*14. ábra*). Ha sikerült a beállítás, akkor a Target PC-n a *15. ábrán* láthatóakhoz hasonló jelenik meg.

14. ábra. Target PC IP-címének beállítása.

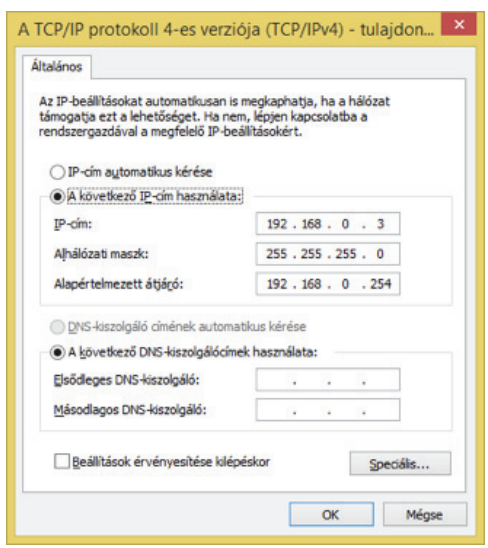


15. ábra. Target PC képernyőképe újraindítás után.



A Host PC hálózati beállításai is ennek megfelelően módosítani kell például 192.168.0.3–ra és 255.255.255.0 alhálózati maszkra, hogy ugyanabban az alhálózatban legyenek, kommunikálni tudjanak.

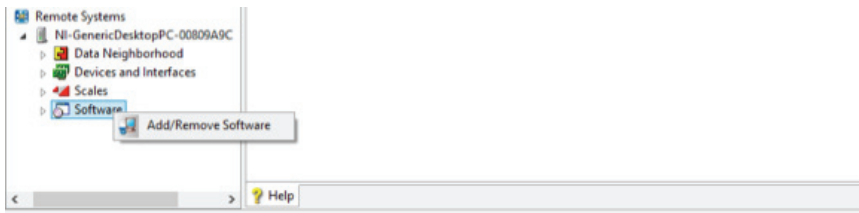
16. ábra. Host PC hálózati beállítása.



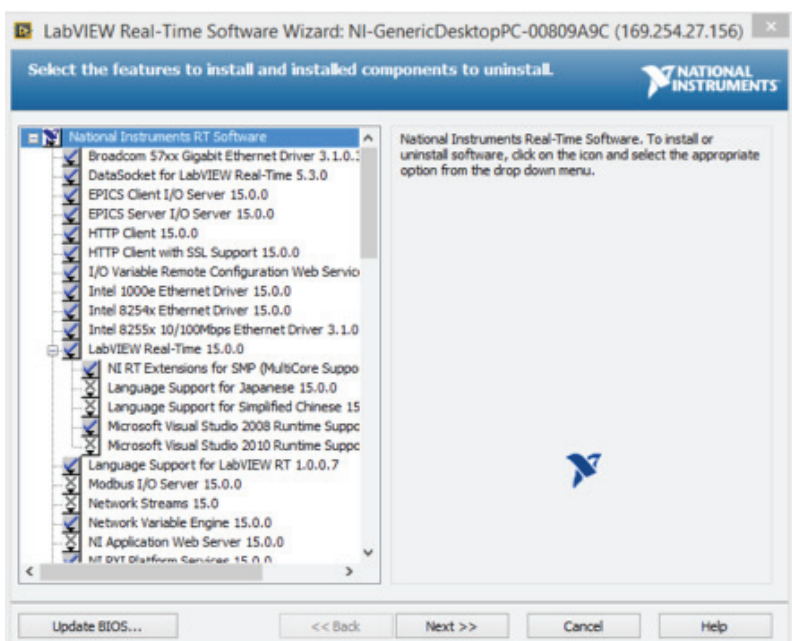
A beállítások érvénybe lépése után a MAX-ban a Target PC állapota a System Settings-ben lekérdezhető.

A Target PC számára a szükséges modulok a Software pontban kiválaszthatóak és telepíthetőek (*17. és 18. ábra*) a multifunkciós adatgyűjtő kártya és a hálózati kártyát tartalmazó illesztő programokkal együtt.

17. ábra. Modulok telepítése a Target PC-re.



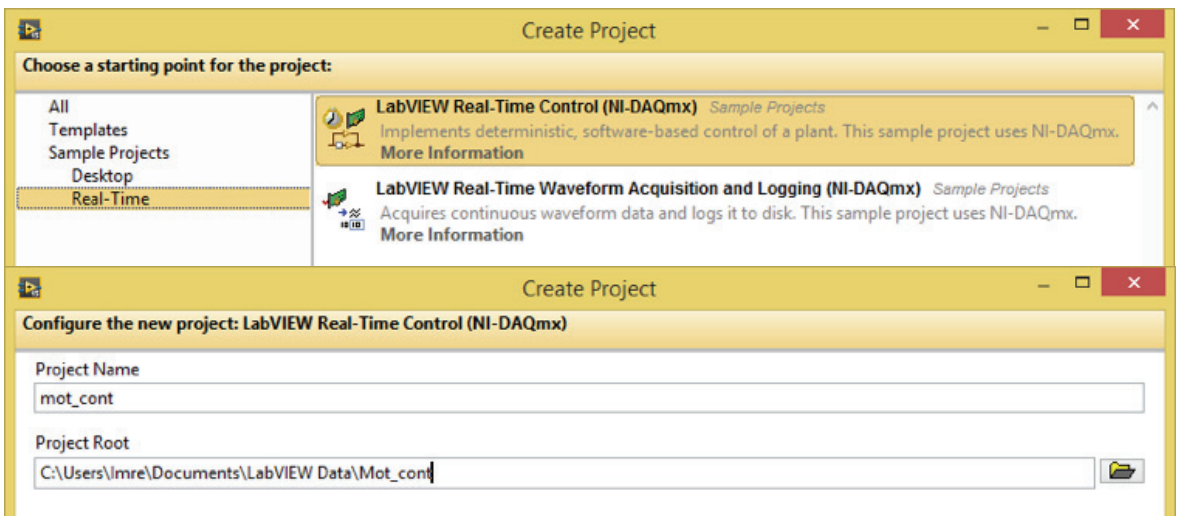
18. ábra. Target PC-re telepítendő modulok.



A modulok telepítése után a Target PC beállításai megtörténtek, a Target PC alkalmas arra, hogy Ethernet kapcsolaton a Host PC-ről programot tölthessünk rá és futtathassuk azt.

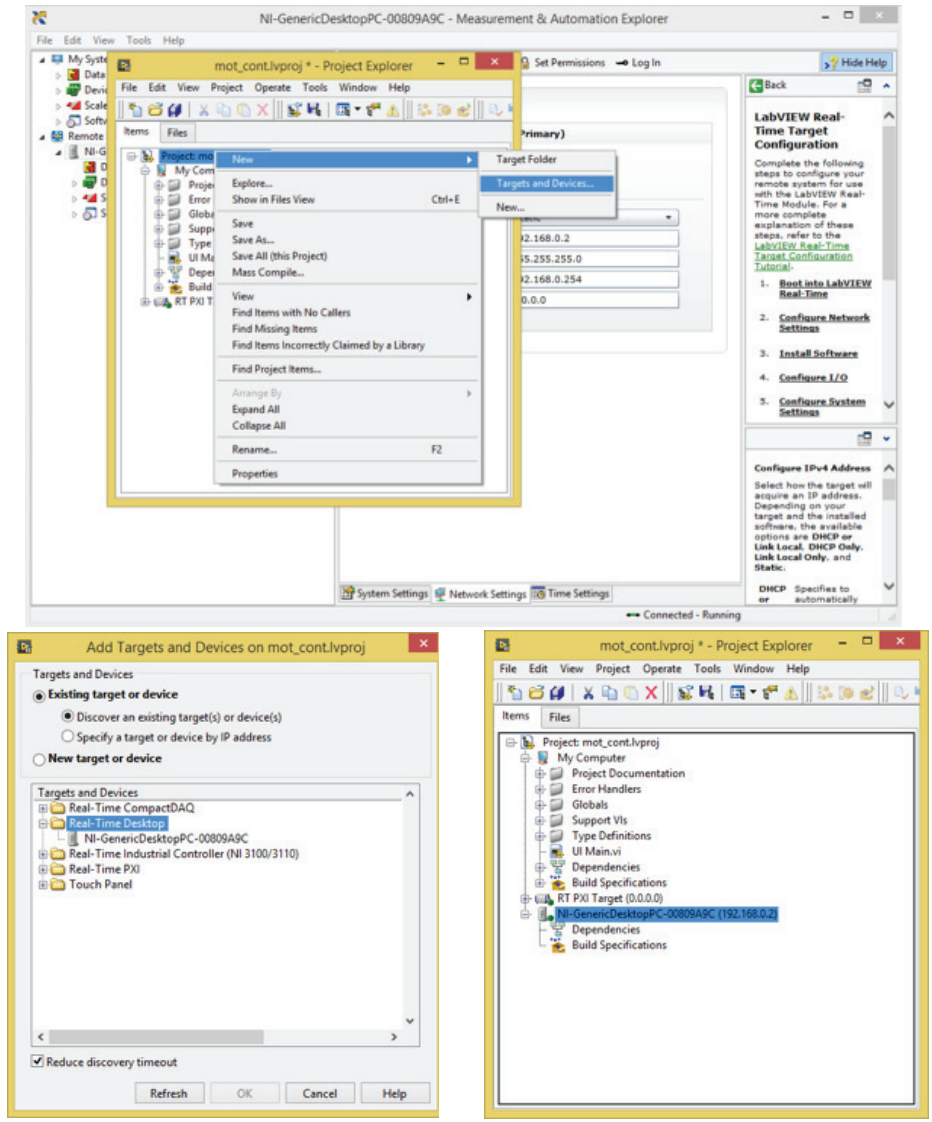
A valósidejű irányítórendszer programozásához a LabVIEW-ban létre kell hozni egy Real-Time projektet, melyet a LabVIEW indítása, majd Create Project választása után lehet megadni a Real-Time projekt választásával, majd a projekt nevének megadásával (*19. ábra*).

19. ábra. Valósidejű irányítás projektjének létrehozása.



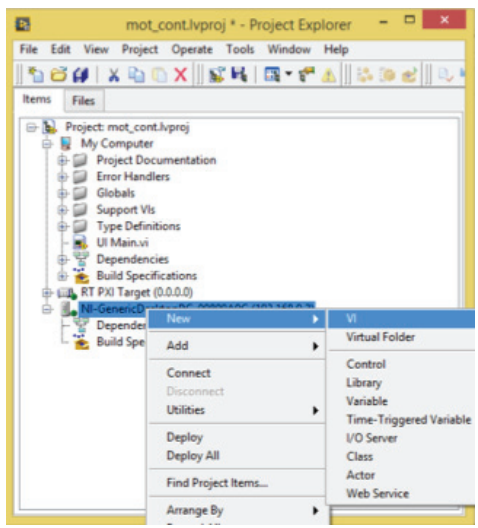
A projekthez hozzá kell adni a Target egységet, jelen esetben az előzőekben konfigurált Target PC-t a Project Explorer-ben (*20. ábra*).

20. ábra. Real-Time projekt – Target megadása.



Ezzel a rendszer konfigurálása megtörtént. A LabVIEW-ban új VI segítségével lehet létrehozni alkalmazást (*21. ábra*).

21. ábra. Új VI létrehozása.



Az alkalmazás Target PC-re történő letöltés után futtatható, a Host PC-n a Target PC-n futó alkalmazás egyes jelei megjeleníthetőek.

Összefoglalás

A cikkben egy olyan számítógép-alapú irányítórendszer került bemutatásra, mely prototípus fejlesztésekhez univerzális, valósidejű platformként alkalmazható és LabVIEW fejlesztőkörnyezet felhasználásával programozható. A fejlesztőrendszer két egységből áll a Host PC-ből, mely a szoftver fejlesztésére, folyamat felügyeletre, információk megjelenítésére használható, és a Target PC, amely valósidejű operációs rendszeren valós időben futtatja a LabVIEW fejlesztőkörnyezetben megírt alkalmazást. A két egység Ethernet-hálózaton kommunikál egymással. A fejlesztőrendszer nagy segítséget nyújthat a gyors prototípus- fejlesztéshez és egyéb tesztelési funkciókhoz is.

Galéria

Sóti István fotói ***(Mongólia)***