



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

Balogh András

**A TÁVVEZÉRLÉS MOBIL ESZKÖZÖKÖN
VALÓ KÖZPONTOSÍTÁSA
WEB-ALKALMAZÁSOK SEGÍTSÉGÉVEL**

TDK Dolgozat

Konzulensek:

Dr. Imre Sándor, egyetemi tanár

Dr. Szabó Sándor, egyetemi adjunktus

2014

Tartalomjegyzék

Összefoglaló	4
Abstract	5
Bevezetés	6
I. Távvezérlési megoldások	8
1. Jellemző vezeték nélküli technológiák	8
1.1. IR távirányítók	9
1.2. RF megoldások	9
2. Távvezérlő eszközök.....	12
2.1. A távirányító, mint fizikai eszköz.....	12
2.2. A távirányító, mint mobil alkalmazás.....	14
3. Alternatív megközelítések	17
II. A távvezérlésre alkalmas mobil platformok	19
1. A mobil készülékek funkcionális felépítése	19
2. Mobil operációs rendszerek	22
2.1. Android	22
2.2. Apple iOS	23
3. A HTML5 technológia.....	24
3.1. Újdonságok	25
3.2. Elérhetőség a mobil operációs rendszerekben	26
III. Tervezési megfontolások	29
1. Követelmények	29
2. Alapmodell.....	30
IV. A Necticon rendszer bemutatása	33
1. A Necticon Mobile.....	33
1.1. Device Interface Manager.....	34
1.2. Necticon Protocol	37
2. A specifikált rendszer előnyei.....	40
V. A kísérleti rendszer megvalósítása	42
1. Felhasznált eszközök	42
2. A rendszer implementációja	43
2.1. A távvezérelendő eszköz megvalósítása.....	43
2.2. A vezérlőfelületek létrehozása.....	44
2.3. A Necticon Mobile implementációja.....	46
2.4. A rendszer értékelése	49

Konklúzió.....	51
Irodalomjegyzék.....	54
Rövidítésjegyzék.....	60

Összefoglaló

A napjainkban elterjedt távvezérlési módszerek alkalmazhatóságát és továbbfejlesztési lehetőségeit számos olyan tényező korlátozza, mely az alkalmazott modellből fakad. A jelenlegi megoldások függetlenül attól, hogy fizikai célhardver, vagy alkalmazás formájában kerülnek megvalósításra, azok decentralizált volta miatt csakis adott eszközökhöz dedikáltan érhetőek el. Noha léteznek univerzális megoldások, ezek egyike sem képes korunknak megfelelő interakciós szintet, egyértelműséget és megfelelő kényelmet biztosítani.

Dolgozatom során éppen ezért a jelenlegi uralkodó modellt általánosítva egy olyan megközelítést alkalmaztam, melynek lényege, hogy a felhasználó egyetlen központi entitáson keresztül lép interakcióba a távvezérelendő eszközökkel. Az alkalmazott modell - annak centrális volta miatt - egyértelműsíti a távvezérlési folyamatot, amely magában hordozza annak bármilyen környezetben történő alkalmazhatóságát. A megalkotott modellt interpretálva a jelenleg elérhető mobilkészülékekre egy olyan rendszert terveztem, amelyben egyetlen központi mobilalkalmazás képes a web-alkalmazások (HTML5) formájában megvalósított vezérlőfelületek megjelenítésére, valamint az azokkal történő kommunikációra. A távvezérelt eszközökkel történő kommunikáció ezt követően bármilyen technológia segítségével megtörténhet.

A rendszer implementálása során Android operációs rendszeren belül valósítottam meg a mobil alkalmazást, valamint kísérleti jelleggel módosítottam olyan céleszközöket, amelyek lehetővé tették a rendszer működésének szemléltetését. A kommunikáció az eszközökkel Bluetooth LE technológia segítségével történt. Ezen felül egyszerű web-alkalmazás formájában implementáltam olyan vezérlőfelületeket, melyek a céleszköz távirányítását tették lehetővé.

Abstract

The methods of remote control available nowadays are limited in applicability and in development by means of factors due to the general model in use. The solutions used today, because of their decentralized approach, are available only on the dedicated devices, independently from the form of implementation (hardware or software). Though there are universal solutions, these are not able to ensure appropriately the expected level of interaction, clarity, and comfort.

In this paper, I made a generalization of the model used by the available solutions to find a better approach to solve the problem. The model I made contains a central entity which creates interaction between the remotely controlled device and the user. The centralized approach in this way clarifies the way of remote control for the user, and implicitly holds the possibility of generalof general application.

Interpreting the model defined I designed a system in which, one central mobile application is able to visualize the controlling interfaces, which are webapplications (HTML5) dedicated to the remotely controlled devices. The message exchange can be achieved with various wireless communication technologies, that are supported by the mobile equipment.

For the implementation of the designed system, I created a mobile application in Android environment, modified the program of some available devices for Bluetooth LE (the communication technology used) development, and implemented a webapplication that is suitable for controlling the modified devices.

Bevezetés

Napjainkban a távvezérlés az ember mindennapi teendői során használt elektronikus eszközök egyik kulcsszolgáltatásává vált, lehetővé téve azt, hogy akár nagyobb távolságokból, pozíciónk megváltoztatása és az eszközök érintése nélkül is kezelni tudjuk azokat egy megfelelő elektronikus szerkezet vagy szoftver segítségével. Ez a fajta kényelem sok esetben előtérbe helyezi a távolról való vezérlést a készülékeken elhelyezett, fizikai felületekkel szemben, éppen ezért a távirányítás alapján befolyásolja mind a felhasználói élményt (kezelhetőség), mind az adott eszköz hasznosságát (használhatóság). A közvetett és közvetlen felhasználók elégedettségét hosszútávon ez a két jellemző határozza meg.

A jelenleg elterjedt távvezérlési modell - ahol a felhasználó az egyes eszközök dedikált távvezérlő készülékeit, vagy szoftvereit alkalmazza - korántsem tekinthető optimális megoldásnak. Ennek egyik oka, hogy hosszú időbe telik és rugalmatlan a kiismerési folyamat, ami dinamikusan változó környezetek esetén problémákhoz vezet. További ok, hogy sok esetben a fizikai felülettel rendelkező távvezérlők nem elegendő méretűek, avagy túl összetettek ahhoz, hogy a felhasználó kényelmesen elnavigálhasson a megfelelő funkciókig, ami jellemzően az intelligensebb, számos komplex szolgáltatással rendelkező eszközök esetén okoz gondot.

Végül azért sem tekinthető optimálisnak a modell, mert ugyan lokálisan nem okoz különösebb kényelmetlenséget, azonban globálisan - tehát az összes környező eszközt tekintve - minden készülék távvezérlési praktikáit ismerni kell és birtokában kell lenni az éppen igényelt távirányító készüléknek vagy alkalmazásnak. Ez utóbbi esetben mind a használt eszközök növekvő száma, mind azok dinamikus cserélődése növeli a távvezérlési folyamat komplexitását, ami összességében negatív hatással van mind a kezelhetőségre, mind használhatóságra.

Léteznek mobilalkalmazás formájában megvalósított távirányítási megoldások is, amelyek egy fokkal könnyebbé és interaktívabbá teszik a folyamatot, azonban a későbbi elemzésem során kiderül, hogy ezek alkalmazhatóságát számos tényező korlátozza. Vizsgálatom során arra is kitérek, hogy az itt alkalmazott megközelítés, annak decentralizált volta miatt több olyan jellemzővel is rendelkezik, amely gátat szab a fejlődésnek, így korunk dinamikusan változó, konvergens és egyre intelligensebb környezetében nem tekinthető időtálló megoldásnak.

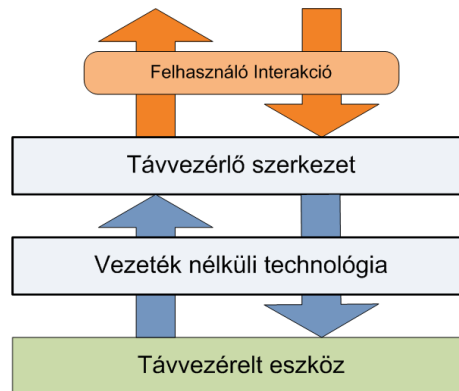
A távirányítók (fizikai és szoftveres) alkalmazása kezdve az otthonunkban megtalálható szórakoztató elektronikai, háztartási és automatizálási eszközökkel, az orvosi, közlekedési, gyártásautomatizálási, építéstechnikai, és informatikai alkalmazásokon keresztül, egészen az űrkutatásig és az egyéb, intelligens környezetet megvalósító megoldásokig számos területen jellemző.

Egy olyan rendszer, amely bármilyen környezetben képes lehet a felhasználók számára egyértelművé és egyszerűbbé tenni annak módját, hogy hogyan tudnak interakcióba lépni az egyes eszközökkel, illetve a kapcsolatba lépés folyamatát is képes gyorsítani, a fentebb említett alkalmazási területeken kiiktathatja a munkavégzésből, az utazásból és az otthoni pihenésből az indokolatlan holtidőket és erőfeszítéseket. Ezzel téve gördülékenyebbé és élvezhetőbbé a mindennapi teendőket. Az általam tervezett megoldás célja ezen felül egy olyan távvezérlési modell alkalmazása (és felállítása), amely annak centralizált volta miatt jobban illeszkedik a jelen és jövő mobilkészülékeihez, a felhasználói szokásokhoz, továbbá az intelligens környezetekhez, azaz időtállóbb megoldást nyújt a távvezérlés megvalósítására.

A következő pontokban röviden bemutatásra kerülnek a jelenleg elérhető távirányítási megoldások (I. fejezet). Azt követve kifejtésre kerülnek a mobil (fizikai és szoftver) platformok jellegzetességei, valamint alkalmassága a távvezérlési funkciók ellátására (II. fejezet). A tervezési megfontolások ismertetése (III. fejezet) után bemutatásra kerül az általam tervezett, központosított távvezérlő rendszer, amely a fentebb taglalt problémákra kínál megoldást, a mobil eszközök és a HTML5 webes technológia alkalmazásával (IV. fejezet). Az elgondolt rendszer verifikáció jelleggel történő megvalósításának ismertetését (V. fejezet) követően dolgozatomat konklúziómmal zárom.

I. Távezérlési megoldások

Egyike az első távirányítási megoldásoknak Nikola Tesla nevéhez fűződik, aki 1898-ban egy olyan hajót mutatott be, amelyet rádióhullámok segítségével volt képes távolról vezérelni egy megfelelő elektromechanikus szerkezet használatával. Az általa bejegyzett szabadalom a „*Method of and Apparatus for Controlling Mechanism of Moving Vessels or Vehicles*” nevet viseli [1].



1. ábra – A jelenleg elterjedt távvezérlési modell

Napjainkban a távirányító berendezések és alkalmazások lényegében Tesla megoldásával azonos modell alapján működnek (1. ábra), azaz jellemzően igényelnek egy vezeték nélküli adatátviteli technológiát - amely elektromágneses hullámokat alkalmaz - és egy megfelelő elektronikus szerkezetet, ahol a vezérlési funkcióknak megfelelő fizikai felület hozza létre a kapcsolatot az ember és a vezérelendő eszköz között.

A jelenleg elérhető megoldások legnagyobb része csak a fentebb említett rétegekben használt technológiákban tér el, ugyanakkor akadnak olyan alternatív megközelítések is, amelyek teljesen elrugaszkodnak a vezeték nélküli technológiáktól és a fizikai felülettől.

1. Jellemző vezeték nélküli technológiák

A következő pontokban röviden ismertetem az egyes távirányítási megoldásokban előforduló vezeték nélküli technológiákat és módszereket, valamint azok jellemző alkalmazási területeit.

1.1. IR távirányítók

A legtöbb távirányító berendezés infravörös (IR) közeli hullámhosszú, tipikusan 940 nm-es fényt kibocsátó diódákat, azaz LED-eket alkalmaz. Ez a fény az emberi látórendszer számára észrevehetetlen, azonban az eszközökön elhelyezett vevők képesek azt érzékelni. Olyan esetekben, amikor a távirányítón csak egyetlen gomb található, pusztán a vivőfrekvencia érzékelése is elegendő az adott esemény vagy funkció triggereléséhez. Többgombos készülékek esetén az előbbinél összetettebb módszerre van szükség. Lehetséges a vivőfrekvencia különböző frekvenciájú jelekkel történő modulálása, majd a vevőoldalon annak demodulálása, azonban manapság a digitális megoldások az elterjedtebbek, ahol általában egyszerű OOK (On-Off Keying) modulációval történik az adott vezérlőszavak átvitele. Annak biztosításához, hogy csak a megfelelő készülék kerüljön vezérlésre, különböző kódolási és dekódolási technikákat alkalmaznak a gyártók [2] [3].

Mivel e technológia a fényt alkalmazza az adatok átvitelére, rálátással kell rendelkezni a vezérelendő objektumra. Ezen a frekvencián ugyanis elektromágneses hullámok elhajlása egyáltalán nem jellemző. A környezetünkben tárgyakat alkotó anyagok fényelnyelő képessége általában igen nagy, így jellemzően csak egy korlátozott, zárt térben működőképes, azaz a megoldás önmagában nem alkalmas egész épületek lefedésére. A probléma áthidalására alkalmazhatók olyan készülékek, amelyek képesek imitálni a távirányítókat, azok vezérlése pedig történhet egy másik alkalmas kommunikációs technológiával és egy további vevő segítségével. Az ilyen rendszerek kiépítése azonban körülményes, és nagyobb épületek esetén igen összetett feladat [3].

1.2. RF megoldások

A második leginkább jellemző megoldás a szabványos, vagy egyedi rádiófrekvenciás technológiák használata, melyek legnagyobb része a különböző ISM (Industrial Scientific and Medical) frekvenciasávokban helyezkedik el. A fentebb leírtakhoz hasonlóan itt is egy vevő és egy adó berendezésből áll általában a rendszer, azonban itt - a reciprocitás értelmében - már lehetőség van a kétirányú kommunikációra ugyanazzal az antennával (adó/vevő), azaz lehetőség van az információ visszacsatolására a vezérelendő eszköz felől [3].

A rádiófrekvenciás hullámok elnyelődése a hétköznapi anyagokban (az alkalmazott frekvenciasávok függvényében) kevésbé jellemző, azonban a különböző szórás, hullámvezetési és visszaverődési jelenségek (általában fémek jelenlétében)

képesek az átvitel minőségét erősen befolyásolni. A különböző jelenségek hatása függ az adott technológia középfrekvenciájától. A frekvenciatranszponáló vevők egyszerű működéséből és felépítéséből fakadóan számos vivőfrekvencia alkalmazható, a moduláció és a demoduláció pedig igen könnyen elvégezhető a jelenlegi áramkörti technológiákkal.

Számos elterjedtebb vezeték nélküli technológia is megtalálható a távirányítási megoldásokban, melyek az egyes felhasználási területeknek megfelelően szegmentálódnak. Az egyes technológiákat a következő (1.) táblázat foglalja össze energiaigényük [4] [5], topológiájuk, felhasználási területeik (távirányítási célból), illetve egyéb tulajdonságaik szerint [6] [7] [8] [9].

1. táblázat – Az elterjedtebb vezeték nélküli technológiák összehasonlítása

	Wi-Fi (802.11n)	Bluetooth (v4.0)	ZigBee	Z-wave
Alkalmazási területek	Távirányító mobil-alkalmazások (CE)	Szórakoztató elektronika, játékkonzolok, perifériák	Otthon- és épületautomatizálás, ipari irányítás, biztonsági rendszerek, orvosi alkalmazások	Otthon- és épületautomatizálás
Frekvenciasáv	2,4 GHz 5 GHz	2,4 GHz	900 MHz 2,4 GHz	900 MHz
Energiaigény	1,28 W (SIMO TX) 2,1 W (MIMO3 TX)	141,12 mW (Enhanced Data Rate, 4 dBm TX) ~60 mW (LE, 1 Mbps)	106,5 mW (3 dBm TX)	75,9 mW (-5 dBm TX)
Adatátviteli sebesség (max.)	600 Mb/s	3 Mb/s	250 kb/s	40 kb/s
Topológia és szerveződés	centralizált és decentralizált (P2P), csillag, fa, szövevényes	decentralizált, csillag (piconet), szövevényes (scatternet)	decentralizált, szövevényes, fa és csillag	decentralizált, fa, csillag és szövevényes
Mobil-készülékekben	Igen	Igen	Nem	Nem

Napjaink két legismertebb vezeték nélküli technológiája a Bluetooth és a Wi-Fi (802.11), melyek részben vagy kizárólag a 2,4 GHz-es ISM sávot alkalmazzák. Láthatóan a Wi-Fi hálózatokat csak a mobil eszközökre tervezett alkalmazások hasznosítják, ennek egyik oka a technológia energiaigénye [10], amely láthatóan sokszorososa a többinek. A technológia alkalmazásának fő szempontjai ilyen esetekben az internetkapcsolat biztosítása, illetve a helyi (otthoni) hálózaton megosztott tartalmakhoz való hozzáférés. Adatátviteli sebessége (600 Mb/s) is ennek megfelelő. A távirányítás lehetősége ilyen esetekben csak másodlagos szolgáltatásként jelentkezik.

Jellemzően a Wi-Fi centralizált szerveződésű változatára építenek, amely minden esetben igényel egy helyi vezeték nélküli hozzáférési pontot vagy átjárót, azaz megfelelően konfigurált hálózati infrastruktúrát. Ugyanakkor előfordulnak olyan készülékek is a piacon, amelyek a decentralizált változatot, a Wi-Fi Direct technológiát alkalmazzák (korábban: ad-hoc Wi-Fi, Wi-Fi P2P), amely már nem igényli a hozzáférési pont jelenlétét a készülékek közötti kommunikációhoz [11].

A Bluetooth (v4.0) technológia távirányításra történő alkalmazása viszonylag kevés területen jellemző. Leggyakrabban a PC-ket vezérlő különböző perifériákban (billentyűzet, egér, prezentációs eszközök) jelenik meg. Tipikusan a játékkonzolok (Xbox, PlayStation) vezeték nélküli felhasználói interfészei alkalmazzák még. Ezen felül a napjainkban egyre népszerűbb okos televíziók (Samsung Smart TV, Google TV) is elláthatók Bluetooth alapú távirányítókkal [7].

A technológia sajátossága az alacsony energiaigény, amely lehetővé teszi a szabványos elemekkel történő tápellátást, emellett viszonylag magas adatátviteli sebességgel rendelkezik (3 Mb/s) [12]. Tömörített hanganyagok átjátszására, kisebb fájlok (képek, dokumentumok) mozgatására kifejezetten alkalmas. A 2010-ben adoptált Bluetooth v4.0 technológia azonban egy olyan új, alacsony energiaigényű változatot vezetett be (Bluetooth LE), amely lényesen alacsonyabb fogyasztási paraméterek mellett képes gyorsabb kapcsolatkiépítést (<1s) és robosztus adatkommunikációt megvalósítani [13].

A kifejezetten alacsony teljesítményű vezeték nélküli technológiák közül az egyik leggyakrabban használt szabványos megoldás a ZigBee, amely akár gombelemekkel is hosszabb ideig működtethető. Mind az alacsony teljesítményéből, mind a decentralizált, szövevényes szerveződéséből fakadóan az elérhető adatátviteli sebesség alacsonyabb (250 kb/s) a korábban említetteknél [14].

Az alacsony sebességéből fakadóan csak olyan területeken jellemző a felhasználása, ahol kisebb adatsomagok (szenzorok által mért értékek, távvezérlő kódszavak) alacsony késleltetéssel való átvitele a cél. Általában otthon- és épületautomatizálási rendszerekben, orvosi alkalmazások esetén, illetve az ipari irányítás területén alkalmazzák [8].

A Z-Wave technológia nem került szabványosításra, azonban jelenléte az otthonautomatizálási rendszerekben meghatározó szereppel bír. Energiaigénye lényegében megegyezik a ZigBee által specifikáltakkal, azonban adatátviteli sebessége mindössze 40 kb/s, így itt kifejezetten jellemző a kis adatsomagok használata. Szerveződése alapvetően megegyezik a ZigBee decentralizált és szövevényes struktúrájával [9]. Sem a ZigBee, sem a Z-Wave technológia nem érhető el napjaink mobilkészülékein.

2. Távvezérlő eszközök

Az ember és a távvezérelt eszköz közötti kapcsolatot a távirányító felület hozza létre, melynek alapvetően két típusa létezik. Az első és egyben legelterjedtebb a dedikált (vagy univerzális) fizikai eszköz formájában megjelenő távirányító (2. ábra). A második az egyre népszerűbb és jelenleg újdonságnak számító, mobil alkalmazás formájában megvalósított virtuális vezérlőfelület, amelyet egy adott mobil operációs rendszeren implementálnak a készülékek fejlesztői. A következő pontokban az egyes típusok tulajdonságai és alapvető problémái kerülnek röviden összefoglalásra.



2. ábra – Különböző távvezérlő készülékek [15] [16] [17]

2.1. A távirányító, mint fizikai eszköz

A fizikai célhardver formájában megvalósított távirányítók kialakítása számos formában történhet meg. Lehetőség van a szintetikus anyagból készült gombok használatára (2. ábra, bal oldalon), amelyek a hagyományos funkcionalitást teszik lehetővé. Az érintésérzékeny felületek és kijelzők alkalmazásával (2. ábra, középen), a

hagyományos funkcionalitás kiegészíthető a visszajelzésekkel, kézi gesztusok felismerésével és látványosabb ábrákkal. Alkalmazhatók ezeken felül mozgásszenzorok is, amelyek a kézmozdulatok felismerését teszik lehetővé (2. ábra, jobb oldalon).

Az alkalmazott technológiák függvényében alakulnak az egyes távvezérlők árai is. Így például egy érintőkijelzővel ellátott darab ára 100–200 \$ intervallumban mozog, míg egy egyszerűbb távvezérlő ára 20–30 \$ körül alakul (kivételtől, céltól függően) [18]. A méretet és a felhasznált gombok mennyiségét pedig alapvetően az alkalmazási terület, azaz a távvezérelendő készülék által nyújtott szolgáltatások száma és összetettsége határozza meg.

Példaképpen egy motorikusan vezérelt kapu nyitásához (vagy zárásához) elegendő egyetlen gomb is, azonban egy otthoni házimozzi rendszer esetén elengedhetetlen egy szofisztikáltabb távvezérlő (2. ábra), amely lehetővé teszi a komplex szolgáltatásokhoz való hozzáférést és a menürendszerben való navigálást, valamint elérhetővé teszi a paraméterek bevitelét is. Ez a fajta megközelítés, bár a felhasználói elégedettséghez szükséges feltételt teljesíti (hiszen az eszközök működésre bírhatók), nem minden esetben elégíti ki az elégségesség feltételét.

A környezetünkben megtalálható elektronikus készülékek egyre több és összetettebb szolgáltatást tartalmaznak integráltan, így a hozzájuk tartozó távirányítók is egyre összetettebbé válnak. Ezen felül számosságuk is nő a használt készülékek számának függvényében, hiszen mindegyikhez dedikált távirányító készülék tartozik, melynek gombkiosztására és a gombokhoz tartozó funkciókra minden használatkor vissza kell emlékeznünk, ami egy bizonyos komplexitási szint után kényelmetlenné vagy akár kezelhetlenné válik.

Természetesen lehetnek olyan szolgáltatások is, amelyekre alkalmasint nincs is szükségünk, azonban akadhat számos olyan, amire viszont lenne, azonban a használatukhoz szükséges lépések száma miatt, a kényelmetlenséget elkerülve inkább nem használjuk őket. Ez a fajta bonyolultság nem az adott szolgáltatást minősíti, hiszen az oda való „eljutás” a kényelmetlen. Egyes esetekben előfordulhat, hogy a távirányító kialakítását korlátozó tényezői miatt (súly, méret) számos dedikált funkciógomb nem is helyezhető el.

A jelenlegi távvezérlési koncepció alapján erre a problémára részben megoldást kínálhatnak az univerzális távirányítók, amelyek kiváltják annak eldöntését, hogy melyik távirányítóra is van éppen szükségünk (3. ábra). Így azonban a készülék-specifikus szolgáltatások eléréséhez rendelt ún. „gyorsgombok” elvesznek, és amennyiben használni kívánjuk ezeket a funkciókat további navigációs lépésekre

kényszerülünk, ami egy olyan eszköznél, ami nem rendelkezik megfelelő visszacsatolási információs felülettel a felhasználó felé, további nehézségeket szül. Ezen felül nem minden szituációban (pl. munkahelyen, vendégségben) érhetők el az univerzális megoldások.



3. ábra – Egy tipikus univerzális távvezérlő [19]

Dinamikusan változó környezetek (új, fejlettebb, olcsóbb készülékek penetrációja) esetén az előbbieken taglalt problémák hatása sokkal erősebb, ugyanis a kiismerési folyamat minden alkalommal újraindul. Bár a távirányítók esetén megfigyelhető egyfajta kialakítási trend - ami a folyamatot gyorsítja - mégis előfordulhat, hogy újabb felületi elemek is megjelennek, melyek feladatát a felhasználói kézikönyv (általában több 10 oldal) tanulmányozásával oldhatjuk csak fel.

A leírtak fényében kijelenthető, hogy a célhardverként megvalósított távirányítók csak minimális mértékben (továbbá drágán) szolgáltatnak a XXI. század intelligens eszközeihez méltó vezérlési és visszajelzési felületet. Használatuk pedig sok esetben indokolatlan és hosszadalmas erőfeszítéseket igényel.

2.2. A távirányító, mint mobil alkalmazás

Korunk mobil eszközei egyértelműen a használhatóság és a kezelhetőség irányába fejlődnek. Számos olyan új „handheld” eszköz jelent meg az utóbbi években a piacon, amely átütő sikert aratott a felhasználók körében. A tabletek, érintőkijelzős okostelefonok, ultrabookok és egyéb hibrid megoldások egyik legfontosabb tulajdonsága, hogy az érintésérzékeny kijelző egy, a mindennapi teendőinkre alkalmas operációs rendszerrel, és egy annak megfelelő hardverrel társul. A Microsoft Windows 8/RT megjelenésével - amely már az érintőkijelzős használatot helyezi előtérbe - a mobilkészülékek és a felhasználók „eszközkezelési” rutinja várhatóan tovább evolválódik az érintőkijelzős használat irányába [20].



4. ábra – Pioneer vezérlőfelület [21]

megvalósítás korlátai, így sokkal ergonomikusabb vezérlőfelületek (interaktív menük, gombok, listák stb.) alakíthatók ki (4. ábra) [3] [21].

A készülékek üzembe helyezése, és az azokkal való megismerkedés is sokkalta egyszerűbbé és felhasználóbarátabbá válhat, hiszen lehetőség van animált és/vagy „TODO” típusú listával segíteni a felhasználót, továbbá ha valahol elakad, bármikor elérheti a világhálón elhelyezett segítségeket, vagy akár bármilyen utánajárás nélkül, egy tapintással hívhatja a lokális ügyfélszolgálatot.

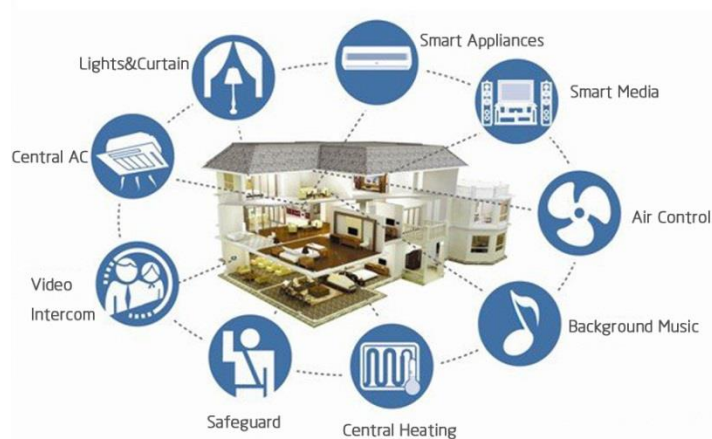
Ahogy a vezeték nélküli technológiák kapcsán is említésre került, ezek a mobil alkalmazások jellemzően olyan készülékek esetén érhetők el, amelyek valamilyen elsődleges szolgáltatás (megosztott médiatartalmakhoz és az internethez való hozzáférés) nyújtásához igénylik a Wi-Fi technológiát. A vezérlőszavak átvitele így megvalósulhat a lokális Wi-Fi hálózaton keresztül, mely jellemzően W-os nagyságrendű energiát igényel. A készülékek stand-by állapotában ebből kifolyólag nem valósítható meg a „bekapcsolás” az IEA (International Energy Agency) 1-Watt iniciatívájának értelmében [22]. Azaz nincs lehetőség a fizikai célhardverektől történő teljes elszakadásra, jóllehet a bekapcsoló-gombot a legtöbb felhasználó könnyen felismeri.

A mobil alkalmazás formájában megvalósított vezérlőfelületek rendszerint az egyes platformok alkalmazásboltjaiban érhetők el, és onnan történhet azok telepítése a mobilkészülékre. A megközelítés egyik hátránya, hogy nem feltétlenül érhetők el az egyes távirányítási megoldások minden platformon, a másik hátrány, hogy, ha el is érhető, annak megtalálása és telepítése számos lépést igényel, azaz hosszabb időbe is beletelhet. Ez utóbbi probléma áthidalható lehet a kétdimenziós QR kódok használatával, amelyek például az eszköz hátlapján, vagy a csomagoláson helyezkedhetnek el. A kód olvasásával egyértelmű hivatkozás kapható, amely egyetlen

lépéssel az alkalmazásboltba irányíthatja a felhasználót, ugyanakkor az ilyen kódok olvasásához erre alkalmas mobil applikáció szükséges.

A távirányító alkalmazások további hátránya azok szolgáltatásként való kezelése, amelyből kifolyólag jellemzően különböző fantázianevekkel illetik őket. Ezen felül számos, a távirányítással többségében azonos funkciót ellátó szolgáltatás (pl. DLNA kliensek, fényképezőgépek vezérlése) a különböző implementációk megkülönböztethetősége érdekében külön elnevezést kap. Ez a fajta divergencia jelentősen rontja a követhetőséget, ugyanis a felhasználóknak ismerniük kell mind a szolgáltatások nevét, mind a hozzájuk tartozó eszközöket, és össze is kell egyeztetniük azokat, továbbá el is kell tudni érniük a mobil operációs rendszerben. A jelenség több hasonlóságot is mutat a fizikai formában megvalósított távirányítóknál taglaltakkal, és amíg a társadalom technikai újdonságok iránt felettebb érdeklődő rétegének ezek nem okoznak különösebb gondot, addig az átlagos, nem műszaki érdeklődésű emberek számára ez összetett feladat is lehet.

A fentebb felsorolt problémáktól eltekintve (hiszen a feladatok igazából megoldhatók) az egyik legnagyobb hátránya az ilyen megoldásoknak, hogy nincs érdemi lehetőség azok integrálására, vagy a különböző alkalmazások közti együttműködés megvalósítására. Ennek egyik korlátozó tényezője a különböző mobil platformokba beépített belső biztonsági mechanizmusok összessége. A másik korlátozó tényező egy olyan a jelenség, melynek során csak egyes azonos márkájú eszközökből álló rendszerek esetén teszik elérhetővé a gyártók az integrált távirányítási lehetőségeket. Ebből következően lehetetlen az olyan készülékekkel való együttműködés, amelyek nem tartoznak az adott gyártó profiljába, noha a környezetünkben mégis megtalálhatók.



5. ábra – Az „okos otthon” koncepciója [23]

Az integrált távvezérlési szolgáltatások alkalmazásának példái lehetnek az 5. ábrán látható, napjainkban egyre inkább előtérbe kerülő otthonautomatizálási rendszerek, amelyek az „okos otthon” koncepcióját hívatottak megvalósítani. Olyan szituációkban, amikor - tegyük fel - megérkezünk a munkahelyünkről, a rendszer képes lehet automatikusan deaktiválni a biztonsági rendszereket, felkapcsolni a lámpákat és a televíziót automatikusan bekapcsolva visszajátszani a híreket, vagy olyan műsorokat, amelyeket a set-top-box rögzített. Hasonlóan alkalmas példa lehet, ha egy előadóban a prezentáció indításakor a világítás elhalványul, a redőnyök/reluxák/függönyök pedig elsötétítik a termet. Mindezen mechanizmusok megvalósításához elengedhetetlen az egyes távirányító alkalmazások közötti kommunikáció.

Az integrált távvezérléshez minden érintett gyártónak meg kellene állapodnia egymással, és kidolgozni minden platformon a megfelelő, szoftvereken belüli kommunikációs interfészeket, ami számos lépést és adott esetben túl sok erőforrást igényelhet. Figyelembe véve azt, hogy hány gyártót érinthet a feladat (jellegéből fakadóan minél többet), a jelenlegi modell gyakorlatilag kivitelezhetetlenné teszi a megoldást.

Összefoglalva a leírtakat, a távirányító mobil alkalmazások számos új, kihasználható grafikai és interaktív erőforrással rendelkeznek (3D grafika, beszédfelismerés, gesztusfelismerés, mozgásszenzorok, kamera stb.). Ezek a megoldások jelenleg a Wi-Fi technológiát alkalmazzák a vezérlőszavak átviteléhez, amely magas energiaigénye miatt nem alkalmazható a bekapcsolási funkciókra, ezen felül az infrastruktúra biztosítása is körülményes. Léteznek Bluetooth alapú megoldások is, azonban a piacon fellelhető eszközökben csak nagyon ritkán fordulnak elő (szinte kizárólag a számítógép vezérlését valósítják meg, így ezekre nem tértem ki külön). Elmondható ezen felül, hogy a távvezérlő alkalmazások telepítése és használata nem feltétlenül egyértelmű az átlagos felhasználó számára, és az integrált távvezérlés megvalósítása is csak számos nehézséggel történhet meg (platformok, megállapodások).

3. Alternatív megközelítések

Léteznek olyan távvezérlési megoldások is, amelyek elrugaszkodnak a vezérlőfelületektől, és más interfészekon keresztül valósítják meg az ember és a készülékek közötti interakciót. A vezérlés szintén távolról, azonban egyedi módszerek alapján történik.

Az egyik legismertebb megoldás az Xbox játékkonzolok kiegészítőjeként megjelent Kinect szenzor, amely 3D-ben képes az (akár több) ember teljes csontvázának

helyzetét követni [24]. Ezen felül alkalmas arc- és hangfelismerésre is. A szerkezetre lehetséges külön alkalmazásokat fejleszteni Windows és egyéb operációs rendszereken is (elszakadva az Xbox platformtól). Segítségével olyan lehetőségek válhatnak elérhetővé (a teljesség igénye nélkül), mint a virtuális billentyűzetek és kezelőfelületek, a különböző arcfelismerésen alapuló biztonsági mechanizmusok, valamint a hang általi vezérlés, továbbá ezek kombinációi. A térbeli képmegjelenítés fejlődésével a kézmozdulatok felismerése egyre fontosabbá válik (napainkban is léteznek ígéretes megoldások [25] [26]).

Hasonló funkcionalitással bír a Samsung által bemutatott interaktív szolgáltatás, amellyel az okostelevisziók vezérlését lehetséges távolról megoldani. A bemutatott technológia lehetővé teszi a kurzor mozdulatokkal történő pozícionálását, a menüben a hang segítségével való navigációt, valamint az arcfelismerést is [27].



6. ábra – EEG szenzor [28]

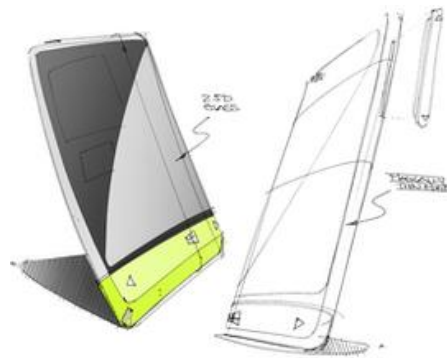
A távvezérlés egy másik úton történő megoldása lehetséges a gondolatok, érzelmek felismerésével, azaz az agyhullámok különböző tulajdonságainak monitorozásával. Az erre alkalmas eszközöket (6. ábra) általában az emberek fején helyezik el, éppen ezért szokás „neuroheadset” néven is emlegetni őket. Több gondolatfelismerésre alapuló megoldás is létezik a piacon jelenleg, melyek akár mobil eszközökön is alkalmazhatók, és szabadon fejleszthetők rájuk különböző alkalmazások [28] [29].

II. A távvezérlésre alkalmas mobil platformok

Mint ahogyan az korábban is szóba került a távirányítók mobil alkalmazás formájában megvalósított megoldásai során, korunk mobil eszközei kifejezetten jó alapot nyújtanak a távvezérlés interaktív módon történő megvalósítására. Dolgozatom jelen szakaszában célom az általam tervezett rendszer alkalmazási környezetének rövid és összefoglaló jellegű leírása. A következő pontokban így a mobilkészülékek különböző típusait, valamint az azokon elérhető és jelen esetben releváns platformokat tekintem át.

1. A mobil készülékek funkcionális felépítése

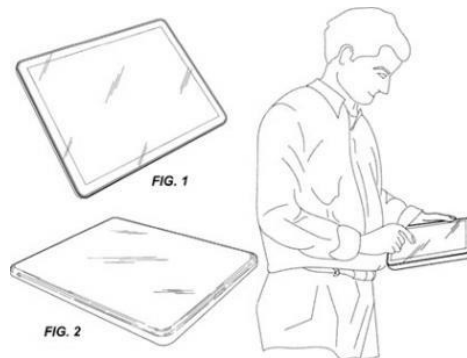
Az egyes mobil eszközök fizikai kivitelüket tekintve jellemzően három kategóriába sorolhatók úgy, mint okostelefonok, tabletek és laptopok [20].



7. ábra – Egy okostelefon tipikus kinézete [30]

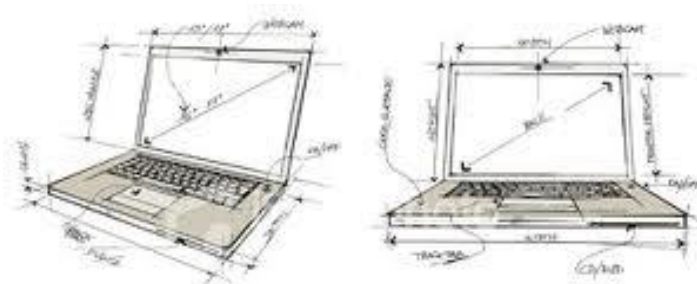
Az okostelefonok előfutárainak a PDA (Personal Digital Assistant) eszközök tekinthetők, melyek támogatták az operációs rendszer és a különböző alkalmazások érintésérzékeny vezérlését. Mivel alapvetően a kézben történő használatra tervezték az okostelefonokat, méretüket tekintve jellemzően 3–5 inch (7,5–12,5 cm) képmérettel rendelkeznek, melyek a készülékek előlapjának legnagyobb részét uralják (7. ábra). Számítási kapacitásukat tekintve akár 1–2 GHz sebességű, többmagos, alacsony fogyasztású, központi számítási egységgel (CPU) vannak ellátva, melyet 1–2 GB RAM memóriával egészítenek ki. Jellemzően rendelkeznek ezen felül grafikai gyorsító egységekkel (GPU) is. Az alkalmazott akkumulátorok jellemzően 0,5–1 nap rendelkezésre állást tesznek lehetővé, azonban ez természetesen függ a számítási- és energiaigényes alkalmazások aktivitásától. Az okostelefonok legfontosabb funkcióinak („killer app”) azonban a telefonálás, az internet böngészése és adott esetben a zenelejátszás tekinthető, minden egyéb lehetőség csak többletszolgáltatásként

jelentkezik. Méretükből és tömegükből (kényelmesen elfér a zsebben, táskában) fakadóan az ilyen eszközök tekinthetők a legmobilabbnak a mobil készülékek között [31].



8. ábra – Egy tablet jellemző kinézete [32]

A tabletek csupán méretükben és egyes funkcióikban térnek el napjainkban az okostelefonoktól, azonban hardveres felépítésük sok esetben megegyezik. Jellemzően 7–11 inch (17,5–27,5 cm) méretű képernyővel vannak ellátva, mely szintén érintésérzékeny és az előlap legnagyobb részét kitölti (8. ábra). Az alkalmazott akkumulátorok által elérhető üzemidő általában 0,5–1 nap, melyet nagymértékben befolyásol a kijelző mérete és technológiája, valamint az elsődleges funkciók teljesítményigénye. A legfontosabb funkció ilyen készülékek esetén a médiatartalmak lejátszása, valamint az internet böngészése és a játék. Ezen felül nagyobb kijelzőjéből fakadóan prezentációk, dokumentumok, kimutatások áttekintésére is lehetőséget ad. Mobilitását tekintve, mind súlya, mind mérete miatt a kevésbé mozgatható eszközök közé tartozik, azonban nagyobb kijelzőjéből fakadóan lehetővé teszi a hosszabb szövegek olvasását álló helyzetben is [33].



9. ábra – Egy laptop tipikus kinézete [34]

A laptopok többsége jelenleg nem tartalmazza az érintésérzékeny kijelzőket. Képméretőjüket tekintve 7–17 inch (17,5–42,5 cm) a jellemző méret. Hardveres felépítésük szerint sokkal szofisztikáltabb megoldások jellemzik, melyeket az elsődleges funkciók szabnak meg. Míg az érintésérzékeny eszközökön a szövegbevitel

jellemzően a képernyőn megjelenő, virtuális billentyűzeteken történik, addig a notebookok esetén ezért kitüntetett fizikai célhardver felel (9. ábra), ami a felhasználók körében jelenleg kényelmesebb megoldásnak minősül. A kurzorok navigációja egy erre alkalmas érintőfelülettel vagy egerrel történik, amely nagyobb fokú precizitást tesz lehetővé. Mindebből és a nagyobb képernyőméretből fakadóan az elsődleges funkciók a szövegbevitel és -szerkesztés, valamint a precízebb kezelést és nagyobb fokú átláthatóságot igénylő alkalmazások köré szerveződnek (szövegszerkesztés, prezentációkészítés, projektkezelés szoftver- és hardverfejlesztés, képszerkesztés és illusztrációkészítés, egyéb tervezési alkalmazások stb.). A laptopok aktív használat mellett 4 órás üzemidőt képesek biztosítani. Mind méretükből, mind súlyukból adódóan az ilyen eszközök a legkevésbé mobilak [35].



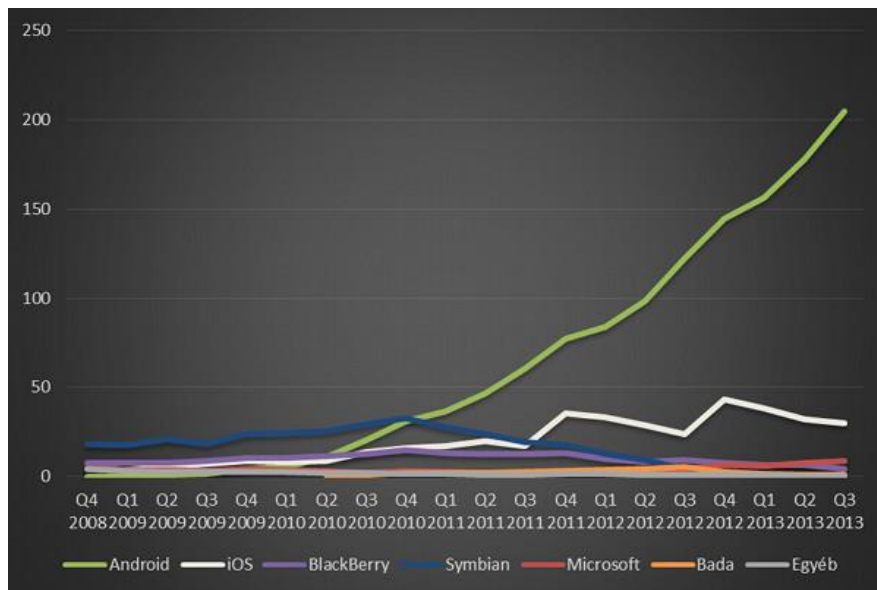
10. ábra –Hibrid mobil készülékek

Léteznek ugyanakkor egyéb, az alapfelépítések előnyeit ötvöző megoldások is (10. ábra - Asus Padfone, Lenovo Convertibles, Microsoft Surface). Ezek az eszközök jellemzően szintén érintésérzékeny kijelzővel vannak ellátva, így csak az erre a feladatra is optimalizált operációs rendszerekkel működnek megfelelően. Jellegükből fakadóan több funkciót képesek ellátni az alapfelépítéssel rendelkező mobil eszközökkel szemben, így míg a laptopok önmagukban kevésbé alkalmasak a távvezérlés feladatára, addig a hibrid megoldásokban lényegesen több lehetőség rejlik [36] [37] [38].

A felsorolt eszköztípusok legnagyobb része tartalmaz beépített Wi-Fi (az internet és a megosztott médiák eléréséhez) és Bluetooth (különböző perifériákhoz, tethering-hez) vezeték nélküli technológiát [7] [6].

2. Mobil operációs rendszerek

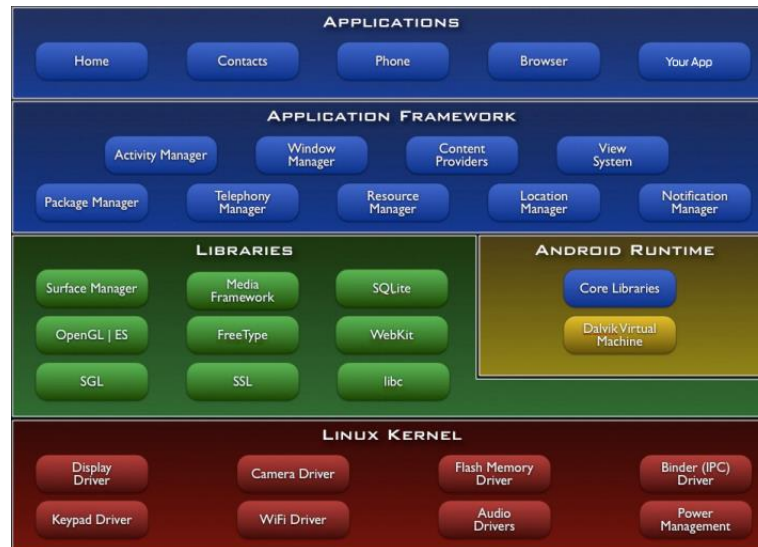
A fentebb taglalt mobil készülékek közül az okostelefonok (és bizonyos mértékben a tabletek) esetén a jelenlegi piacot két mobil operációs rendszer uralja: az Android és az Apple iOS. Bár a hibrid jellegű készülékek esetén gyakori a Microsoft Windows 8/RT rendszer, jelen fejezetben az okostelefonokon elérhető platformokra helyezem a hangsúlyt. Ennek legfőbb oka, hogy az ilyen készülékek, mind áruk, mind kiemelkedő mobilitásuk miatt széles körben elterjedtek. A 11. ábra az okostelefonok eladásait (millió darab) mutatja, platformonként, negyedéves bontásban [39].



11. ábra - Okostelefon-platformok eladásai a Gartner szerint (millió db) [39]

2.1. Android

Az Android operációs rendszer a 12. ábrán látható rétegekből épül fel. A legalsó a „Kernel” réteg, amely egy, a Google által a későbbiekben módosított 2.6-os verziójú Linux Kernel. Ez a réteg felel az operációsrendszer és az elérhető hardver kapcsolatáért, mely feladatot a megfelelő Driverek segítségével éri el (Wi-Fi, Bluetooth, Audio stb.). Az Android ezen felül a Linux rendszert használja minden alapfunkció eléréséhez is. Ilyen alapfunkció lehet a memóriakezelés vagy a folyamatok kezelése is [40].



12. ábra – Az Android operációs rendszer rétegszerkezete [40]

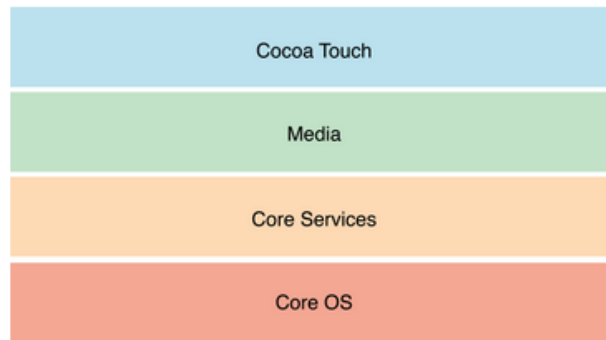
A „Kernel” réteg fölött helyezkedik el a „Libraries” réteg, amely az Android azon natív (C/C++) könyvtárait tartalmazza, melyek az adott hardware-nek megfelelően eltérhetnek. Ez a réteg teszi lehetővé a különböző adattípusok kezelését. A legfontosabb könyvtárak olyan funkciókat töltenek be, mint a 3D grafika, az ablakkezelés, a médiakodekek és -formátumok kezelése, az adatbázisok kezelése, vagy a böngészőmotor. Az Android Runtime tartalmazza a Dalvik virtuális gépet, valamint az alap (Core) JAVA könyvtárakat és osztályokat [40].

Az „Applications Framework” réteg egy olyan alkalmazás keretrendszert definiál, amelynek segítségével a különböző egyedi alkalmazások interakcióba tudnak lépni az operációs rendszerrel (alkalmazások életciklusai, alkalmazások közötti adatmegosztás, híváskezelés, erőforrások kezelése stb.) [40].

A legfelső rétegben helyezkednek el az alkalmazások („Applications”), melyek közül több előre telepítve is elérhető az Android rendszerekben (pl. SMS kliens) [40].

2.2. Apple iOS

Az iOS az előbbiekhöz hasonlóan szintén több különböző feladatot ellátó rétegre (19. ábra) bontható. A legfőbb különbség az Android rendszerhez képest, hogy míg az előbbi általánosabb jelleggel tervezték, azaz számos különböző hardveren való futásra optimalizálták, addig az iOS esetében egyedi tervezésű okostelefonokra történt az illesztés, azaz kevesebb eszközön (lényegében csak a kitüntetett készülékeken) képes futni [41].



13. ábra – Az iOS operációs rétegszerkezete [41]

A legalsó „Core OS” réteg felel a szálkezelésért, a hálózatok kezeléséért, a fájlrendszerhez való hozzáférésért, valamint a memória és egyéb erőforrások menedzsmentjéért. Ezen felül több keretrendszert is definiál, melyek segítségével olyan funkciók érhetők el, amelyek lehetővé teszik a kép- és jelfeldolgozást, a Bluetooth technológia használatát, az iOS készülékeken található 30 pin csatlakozón keresztüli kommunikációt az egyedi tervezésű készülékekkel, vagy a különböző biztonsági szolgáltatások (kulcsok, jogosultságok kezelése) használatát [41].

A következő, „Core Services” rétegben található az olyan alapszolgáltatások keretrendszerei, mint az alacsonyszintű hálózatkezelés, az adatbázis kezelés, a híváskezelés, vagy az alkalmazáson belüli vásárlás [41].

A harmadik, a „Media” réteg a grafika (2D, 3D), az animációk, a szövegrenderelés, vagy a kép- és videószerkesztés. Ebben a rétegben található ezen felül azok a metódusok, amelyek lehetővé teszik a felhasználó médiatartalmainak hozzáféréséhez, lejátszásához és rögzítéséhez [41].

A legfelső, „Cocoa Touch” réteg az alsóbb rétegek egyfajta absztrakciójának tekinthető. Legfontosabb szolgáltatásainak a „multitasking”, a fájlmegosztás, a külső megjelenítő eszközök kezelése, és a dokumentumok kezelése tekintő (a teljesség igénye nélkül) [41].

Mindkét operációs rendszer rendelkezik külön alkalmazásbolttal, ahonnan telepíthetők és ahol megvásárolhatók a kisebb programok. Azonban míg az Android JAVA/C nyelven írt alkalmazásokat, addig az iOS Objective-C nyelvet igényel [41].

3. A HTML5 technológia

A HTML5 az internet egyik legfontosabb szoftveres technológiájának, a HTML-nek (HyperText Markup Language) a legújabb revíziója, mely jelenleg is fejlesztés alatt

áll (World Wide Web Consortium - draft). A HTML feladata alapvetően az interneten elérhető tartalmak strukturálása, valamint azok megjelenítése. A HTML5-ben sincs ez másképp, azonban a cél jelen változatnál az, hogy a különböző webes tartalmak megjelenítéséhez ne legyen szükség minden böngészőn a megfelelő beépülő-modulok telepítésére. Éppen ezért a HTML5 már beépítve tartalmazza, mind a CSS (Cascading Style Sheet) új verzióját, mind a JavaScriptet, valamint számos új szintaktikai elemet és API-t (Application Programming Interface), melyek segítségével optimalizálhatók, könnyen hozzáférhetővé, illetve átláthatóbbá tehetők az egyes weboldalak és azok tartalmi [42].

A HTML jelen változatában több olyan új szolgáltatás is szabványosításra került, melyek lehetővé, illetve könnyebben implementálhatóvá teszik a webes alkalmazásokat. Mivel az általam tervezett rendszer ezeket a lehetőségeket elérhetővé teszi, valamint adott esetben ki is használja, a következőkben az érintett újításokat, valamint az előbbi pontban bemutatott mobil operációs rendszerekben való elérhetőségüket vizsgálom.

3.1. Újdonságok

A HTML5 egyik újítása az offline működés szabványosítása, mely egyrésztől lehetővé teszi a webes alkalmazások (weboldalak) alap logikájának és a felhasználói felületnek a böngészőben, azaz a kliens eszközön való tárolását (application cache), valamint elérhetővé tesz egy offline tárhelyet (Offline storage, szintén a kliens eszközön), amelyen a különböző (jellemzően a felhasználó által bevitt) értékek tárolhatók [42].

Mіндеzen felül létrehozhatók és kezelhetők SQL (Web SQL database) indexelt (IndexedDB) adatbázisok is lokálisan, melyek nagyobb, komplexebb adathalmazok esetén alkalmazhatók. Mindezen szolgáltatások teljesen offline módon, azaz internetkapcsolat nélkül is elérhetők az első futtatást követően, valamint lehetővé válik az egyes adatbázisok (offline és online) szinkronizálása is. Jellemző példa az ilyen adatbázisok esetén a Gmail offline szolgáltatás, mely a leveleket offline esetben is elérhetővé teszi a böngészőn belül [42].

A HTML5-ben szabványosított következő szolgáltatás, az adott böngészőt futtató eszközön levő fájlrendszer elérését megvalósító File API. Segítségével létrehozhatók és olvashatók különböző tartalmak a böngészőn kívül is. A File API segítségével olyan funkciók is megvalósíthatók, mint a „Drag & Drop”, amely az érintőkijelzős készülékeken könnyíti meg a fájlok mozgását az operációs rendszer és a

webalkalmazás között. Tipikus példa erre levelező programok esetén a feltöltendő, vagy letöltendő mellékletek kezelése (Gmail) [42].

A valós idejű működés és a kliens-szerver kapcsolatok kezelése is módosult. A „Web Worker” funkciócsoport segítségével létrehozhatók külön programfuttatási szálak a webalkalmazáson belül, a számításigényesebb feladatok elvégzésére a felhasználói felületek megakasztása nélkül (pl. útvonaltervezés). A „Web Socket” szolgáltatás lehetővé teszi az egyes webalkalmazások közötti egyszerű kapcsolat-felépítést és adatküldést a http fejlécek nélkül, az átküldendő adatok IP csomagba ágyazásával (chat, M2M kommunikáció). Mindezen felül a „Notifications” funkció lehetővé teszi az egyes közösségi, vagy információs oldalak újdonságainak jelzését egy a böngészőn kívüli felületen is [42].

A HTML5 lehetővé teszi az egyes hardverelemekhez történő hozzáférést is, így olyan funkciók is használhatók, mint a beszédfelismerés (mikrofon), a készülék orientációjának kezelése (gyorsulásmérők, giroszkópok) vagy a GPS modulhoz való hozzáféréssel az adott készülék helyének meghatározása (GPS modul hiányában az IP cím alapján) [42].

A grafika és a multimédiás tartalmak kezelése terén is több újítás történt, melyek közül a legfontosabb, hogy a webalkalmazások számára hozzáférhetővé váltak a grafikus gyorsító egységek (GPU), így mind a médiatartalmak dekódolása, mind a 3D-s animációk végrehajtása sokkal gyorsabban történhet meg. Ezen felül a hang- és videótartalmak külön megjelenítő- és kezelőfelületet kaptak, így azok beágyazása az alkalmazásba sokkal egyszerűbbé vált. A „Canvas 2D” és „Canvas 3D (WebGL)” funkciócsoportok pedig mind síkbeli, mind térbeli (akár valós idejű) animációkat is lehetővé tesznek, azaz látványosabb és interaktívabb felhasználói felületek létrehozására kifejezetten alkalmasak [42].

3.2. Elérhetőség a mobil operációs rendszerekben

A legtöbb mobil operációs rendszer böngészőalkalmazása (Android Browser, Apple Safari, Google Chrome, Mozilla Firefox, stb.) a nyílt forráskódú webkit böngészőmotorra épül, így a publikusan elérhető programozási interfészek is e motor egyes implementációinak képességeit teszik elérhetővé. Ezen felül számos asztali platformon (Windows, Mac OS, Linux) is elérhető, azonban míg az asztali platformokon közel teljes mértékű a támogatottság, addig a mobil operációs rendszerekre (Android, iOS) ez már nem teljesen igaz [43] [44].

3.2.1. iOS

Az Apple Inc. által fejlesztett és gyártott mobil készülékekben a webkit egyes szolgáltatásai az UIWebView osztályon keresztül érhetők el, amely lehetővé teszi az egyedi mobil alkalmazásokban a webes tartalmak megjelenítését illetve a JS (JavaScript) kódok futtatását [45].

A HTML5 alapú webalkalmazások offline működését lehetővé tevő szolgáltatások (application cache, web storage, web SQL) alapértelmezés szerint elérhetők az osztályban, egyetlen korlátjuk, hogy az adatbázisok egyenként nem haladhatják meg az 5 MB-os méretet [46].

Az audio- és videotartalmak webalkalmazáson belüli lejátszása csak részben támogatott, a videotartalmak megjelenítése iPhone készülékek esetén egyelőre csak a teljes képernyőben lehetséges. Mindezen felül a tartalmak lejátszását több esetben sem a beágyazott böngésző alkalmazás végzi, hanem az operációs rendszer beépített média lejátszó alkalmazásai, így ez a fajta funkcionalitás jelenleg nem tekinthető teljes értékűnek [45].

Azonban a legtöbb szabványosított megoldás, egyelőre csak a különböző funkcionalitást elősegítő, JS kód injekciókkal, valamint a natív Objective-C kódsorokkal és az operációs rendszerrel történő interakcióval érhetők el. Mivel ezek a megoldások kívül esnek a publikusan elérhetővé tett hivatalos funkcionalitásokon, így dolgozatomban eltekintek ezek bemutatásától.

3.2.2. Android

Az Android operációs rendszerben a webkit egyes szolgáltatásait az android.webkit csomag (package) tartalmazza, melynek számos osztály képezi részét. Ezek közül a legfontosabb a WebView, amely lehetővé teszi az egyes weboldalak megjelenítését, valamint az elérhető alosztályokon keresztül a webalkalmazáson belüli JS kódokat is futtatni képes számos egyéb szolgáltatás mellett [47].

A WebStorage alosztály valósítja meg a HTML5 offline működést lehetővé tevő szolgáltatásainak kezelését (Application Cache, Web SQL, Web Storage). Alapértelmezés szerint az Application cache mérete gyakorlatilag korlátlan, továbbá manuálisan is állítható (WebSettings alosztály). A Web SQL és Web Storage adatbázisok méreteivel kapcsolatban nincs egyértelmű dokumentáció. A W3C (World Wide Web Consortium) ajánlása szerint 5 MB, ugyanakkor az egyes kliensek feladatává teszi a korlátot meghaladó adatbázisok tárolásával kapcsolatos problémák kezelését

[48]. Az egyes lokális fájlok elérése is támogatott a WebView alapú alkalmazásokban (WebSettings).

Az iOS rendszerhez hasonlóan az Android esetén sem támogatott teljes mértékben a video- és audiotartalmak böngészőn belüli lejátszása, melynek legfőbb oka a böngészőkben használt különböző dekóderek [49] [50].

A hardverelemekhez való hozzáférés szempontjából az Android lehetővé teszi a geolokációs adatbázishoz való hozzáférést (GPS), ugyanakkor egyelőre sem a beszédalapú bevitelhez, sem a grafikus gyorsító egységekhez (WebGL) nem lehet jelenleg általános jelleggel hozzáférni [44].

Látható, hogy az Android operációs rendszer által publikussá tett függvényeken és osztályokon keresztül is csak igen korlátozott mértékben lehetséges az egyes HTML5 által szabványosított megoldásokhoz hozzáférni. Ugyanakkor mindkét mobil operációs rendszerre kiadott hivatalos böngészőalkalmazás (Chrome, Safari, Firefox, Android Browser) sokkal szélesebb körben támogatja az újításokat, így a jövőben várhatóan publikusan is elérhetővé válnak az egyes platformokon [44].

III. Tervezési megfontolások

Dolgozatom jelen szakaszában az általam megvalósítani kívánt központosított távvezérlő rendszer egyes tervezési aspektusait (követelmények, alapmodell, alkalmazható vezeték nélküli technológia) fejtem ki.

1. Követelmények

Megoldásom tervezésekor az elsődleges szempontok közé tartozott, hogy az a jelenleg elérhető távvezérlő megoldások egyes korlátozó tényezőit lehetőség szerint kiiktassam, valamint, hogy növeljem a távvezérlés során elérhető felhasználói élményt. Ez utóbbi követelményt több tényező is befolyásolja.

A felhasználó számára alapvetően a „seamless” működés az egyik legfontosabb elvárás, hiszen ez a jellemző foglalja magában a rendszer konzisztensségét, egyértelműségét és egyszerű kezelhetőségét. A rendszer tervezésekor éppen ezért törekedtem arra, hogy minimalizáljam azon lépések és akciók számát, amelyek az adott eszközök eléréséig szükségesek, emellett azt is szem előtt tartottam, hogy azok legyenek egyértelműek és lehetőség szerint minimális műszaki tapasztalatot, tudást igényeljenek.

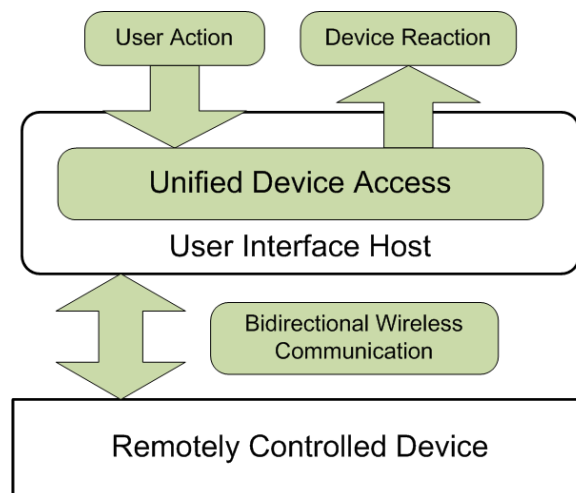
Figyelembe véve azt, hogy a rendszer egyik legfőbb célja az általánosság (s, így a vissz irányba való kompatibilitás is) azaz, hogy annak használata lehetőség szerint bárhol, bármilyen eszköz esetén lehetséges legyen, a rendszer tervezése során nem kötöttem ki az alkalmazandó technológiát, csak azon szükséges funkciókat, amelyek alapvetően szükségesek a rendszerrel való együttműködéhez.

Mіндеzen felül az egyes vezérlőfelületek is készülékenként eltérhetnek, így a lehetséges távvezérelendő készülékek sokszínűsége és az általuk nyújtott szolgáltatások, ill. azok megvalósítási módjainak divergenciája alapján belátható, hogy meglehetősen komplex feladat egy olyan vezérlőfelület-modell megalkotása, amely általánosságban alkalmazható bármely készülék esetén. Éppen ezért az általam tervezett rendszernek ez nem is célja, az egyes vezérlőfelületek tervezése és megvalósítása a készülékek fejlesztőinek feladata marad. A rendszernek ugyanakkor lehetőség szerint transzparensnek kell lennie a fejlesztők szempontjából, valamint minimalizálnia kell azt a „többletmunkát”, amit a rendszer alkalmazása a natív alkalmazások, vagy a fizikai távirányítók fejlesztésének megfelelő „munkán” felül igényelne.

Láthatóan az egyes követelményekben a legfontosabb szempont az általánosság és az egyértelműség (a felhasználók és a fejlesztők szempontjából). A következő pontban a kritériumoknak megfelelően definiálok egy modellt, majd kitérek a különböző vezeték nélküli technológiák irányában támasztott követelményekre.

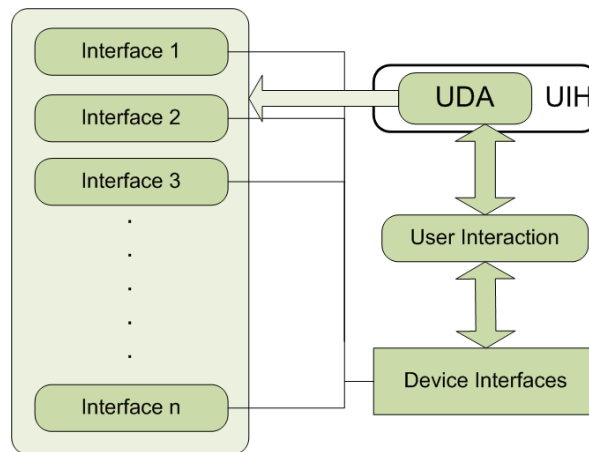
2. Alapmodell

Megoldásom tervezését azon modell felállításával kezdtem, amely megfelelő alapot adhat egy, a jövőben is alkalmazható távvezérlő rendszernek, valamint összhangban van a korábban megfogalmazott kritériumokkal. A definícióhoz a Tesla által használt, valamint a mobil alkalmazások során alkalmazott modellek egyfajta absztrakcióját végeztem el, melynek eredménye leegyszerűsítve a 14. ábrán látható. Elgondolásom alapját a továbbiakban ez a modell szolgálta.



14. ábra – Az alkalmazott modell

A koncepció lényege, hogy az egyes készülékekhez tartozó vezérlőfelületek és logikák (Device Interfaces) központosítva (azaz egyetlen központi entitáson keresztül - Unified Device Access), egy olyan eszközön (UIH - User Interface Host) teszem elérhetővé, amellyel a felhasználók képesek megfelelő szinten interakcióba lépni (16. ábra). Korunk egyik uralkodó jelensége, az „ubiquitous computing” számos olyan új eszköz létrejöttét vetíti előre, melyek túlmutatnak a jelenleg elérhető készülékeken és technológiákon [52].



15. ábra – Vezérlőfelületek (DI) aggregációja az UDA-ban

A jelenség egyfajta indikátorának tekinthető az a funkcionalitásbeli konvergencia, amely a korábban bemutatott „hibrid” mobil készülékek esetén figyelhető meg, így célszerűnek tartottam a modell ehhez való igazítását. Az eszközökkel (Remotely Controlled Device - RCD) történő kommunikáció a korábbi modelleknek megfelelően, egy vezeték nélküli technológiával történik, azonban annak megvalósulása láthatóan az UIH-hoz köthető. Ennek legfőbb oka, hogy míg az UDA (Unified Device Access) értelmezhető absztrakt számítási folyamatként, addig a kommunikáció megvalósítása mindenképpen az adott eszköz (UIH) fizikai kapacitásaitól függ (antenna).

A modell láthatóan teljesíti az egyértelműség feltételét, hiszen egyetlen központi entitás közvetítésével történik az interakció. Mindezen felül magában hordozza az integrált és intelligens távvezérlő rendszerek megvalósítását, hiszen a központi entitás (UDA) alkalmas lehet az egyes vezérlőlogikák közötti interfészek biztosítására.

A modell egyfajta interpretációjának tekinthető a megoldásom, mely a jelenlegi trendnek megfelelően a felhasználónál található, vizuális megjelenítéssel és egyéb felhasználói élményt fokozó interaktív szolgáltatással bíró mobil készülékeket jelöli ki UIH-nak. Az UDA ennek megfelelően egy olyan, adott környezetbe implementálandó (Android, iOS, Windows 8, Bada stb.) mobil alkalmazást takar, amely képes a vezérlőfelületeket (Device Interfaces - DI) párosítani a távvezérelendő eszközzel (RCD), továbbá ezeket a felületeket egy csoportba szedi, és funkció, elhelyezkedés vagy egyéb felhasználói szempontok szerint böngészhetővé teszi.

Az egyes eszközökhöz (RCD) tartozó vezérlőfelületek (DI) kidolgozása és implementálása (a kritériumnak megfelelően) a távvezérelendő készülékek fejlesztőinek a feladata. Ezen DI-k nem egyebek, mint a készülékek (RCD) vizuális leképződései, ha úgy tetszik az RCD-k „arcai”. Adott konvenciókat betartva, a DI-k olyan alkalmazások,

melyek beépülő modulok formájában válhatnak elérhetővé az UDA-n, mint központi alkalmazáson keresztül.

Ugyanakkor a mobil készülékre történő illesztés pusztán egyike a lehetséges interpretációknak. Az egyre nagyobb népszerűségnek örvendő okostelevíziók, vagy a Google által fejlesztett Project Glass keretein belül bemutatott „okos” szemüveg is kijelölhető UIH-nak, azaz az UDA funkcionalitását lehetővé tevő eszköznek [52] [53].

A rendszer további feladata, hogy az UIH által értelmezett és feldolgozott információt eljuttassa egy erre alkalmas vezeték nélküli (vagy akár ilyen szempontból, heterogén) kommunikációs közegen keresztül a távvezérelt eszközökhöz (RCD), és közvetítse a készülék válaszát ugyanezen a csatornán keresztül az UIH felé, ahol az UDA-n keresztül a vezérlőfelület (DI) lefordítja azt a felhasználó számára értelmezhető információvá.

A modell alkalmazhatóságát jelenleg erősen befolyásolja a vezeték nélküli technológiák megléte és azok tulajdonságai. Így amíg nincs egyértelmű szabvány az ilyen eszközök által elérhető vezeték nélküli technológiák tekintetében, addig az alapmodell alkalmazhatósága is ennek megfelelően skálázódik.

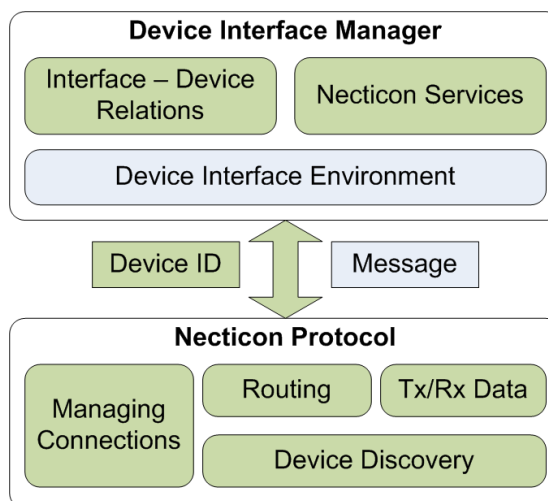
IV. A Necticon rendszer bemutatása

Dolgozatom jelen szakaszában kerül bemutatásra az általam tervezett, mobil eszközökön központosított távvezérlő rendszer, mely kihasználja a webes technológiák és a mobil készülékek nyújtotta lehetőségeket. A rendszert a „Necticon” fantázianévvel illetttem, melyre a továbbiakban ennek megfelelően hivatkozom.

A modellnek megfelelően a rendszer részét képezi egy mobil készüléken (UIH) futó alkalmazás (Necticon Mobile), mely az UDA (Unified Device Access) feladatait látja el, valamint egy vezeték nélküli hálózat (Necticon Network), melyen a kommunikáció zajlik az eszközökkel (RCD).

1. A Necticon Mobile

A Necticon mobil alkalmazás alapvetően két rétegre bontható (16. ábra). Az alsó réteg (Necticon Protocol), a hálózati kapcsolatok létrehozásáért és kezeléséért, az adott vezérelendő eszköz (RCD) legegyszerűbb elérési útvonalának meghatározásáért, illetve az eszköz-specifikus üzenetek biztonságos továbbításáért felel. A felső réteg (Device Interface Manager), a gyártók egyéni vezérlőfelületei számára biztosított alkalmazáskörnyezet, párosítja a DI-eket az eszközökhöz (Device ID), valamint lehetővé tesz egyes saját szolgáltatásokat. A kommunikáció a rétegek között egyszerű függvényhívásokkal zajlik.



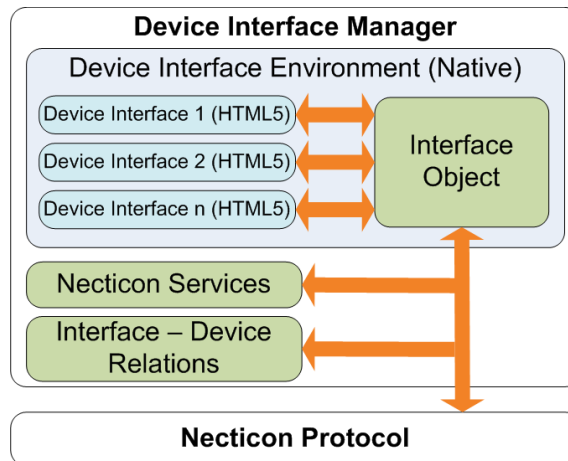
16. ábra – A Necticon Mobile felépítése

1.1. Device Interface Manager

A következő pontokban a Device Interface Manager egyes funkciócsoportjainak (Device Interface Environment, Interface – Device Relations, Necticon Services) működését foglalom össze.

1.1.1. Device Interface Environment

A Device Interface Manager egyik legfontosabb feladata a megfelelő alkalmazáskörnyezet (Device Interface Environment - 17. ábra) biztosítása a vezérlőfelületek számára. Jelen megoldásban ezek a beépülő modulok (DI) a korábban bemutatott HTML5 szabvány szerint implementált webalkalmazások. Az alkalmazáskörnyezet megvalósítása ennek megfelelően egy olyan böngészőfelület létrehozását igényli, amely képes megjeleníteni az ilyen típusú weboldalakat, és amely képes kétirányú kapcsolatot megvalósítani magával az őt futtató környezettel, valamint azon keresztül a Necticon Protocol réteggel. Mivel a natív nyelven írt böngészőalkalmazások (Chrome, Firefoxm Safari stb.) jelenleg nem támogatják a beépülő modulokat (plugin), így a megoldás csak egyedi böngészők (Webkit) létrehozásával történhet, melyek egyelőre csak korlátozott mértékben támogatják a HTML5-ben szabványosított megoldásokat.



17. ábra – A Device Interface Environment viszonya a vezérlőfelületekkel

Ugyanakkor az alkalmazás szempontjából az egyik legfontosabb funkció, az offline működés minden célrendszeren (iOS, Android) elérhető, így elegendő egyszer letölteni a webalkalmazást, hiszen azt követően a lokális adattárból kerülhet kiolvasásra a kód. Ez a folyamat (első hozzáférés) felfogható egyszerű telepítésként is.

A kommunikáció a HTML5 alkalmazások és a környezet között megvalósítható mindkét platformon, egy megfelelő interfész objektum definiálásával, melynek

függvényei a HTML5 alkalmazáson belül elérhetővé válnak. Így az üzenetek küldése és fogadása igen egyszerűen, paraméterátadással történhet meg [47] [45]. A vezérlőfelületek hozzáférhetőségének biztosításához az alkalmazásokat egy webszerveren kell elhelyezni, valamint a környezet (Device Interface Manager) tudtára kell adni, hogy hol is keresse azokat.

Felmerülhetnek adott esetben biztonsági kockázatok is, hiszen a webalkalmazások adott esetben hozzáférhetnek akár operációsrendszer-szintű függvényekhez is (object reflection). Az interfész objektum megfelelő körültekintéssel történő definiálása ezt a lehetőséget kizárhatja.

További kockázat azonban a vezérlőfelületek fejlesztőinek részéről, hogy amennyiben publikusan elérhetővé teszik az egyes felületeket az interneten, úgy annak forrása és működése láthatóvá válik (HTTP). Ez ellen sajnos nincs hatásos védelem, csupán nehezíteni lehet a visszafejtés (reverse engineering) folyamatát (pl. JavaScriptbe ágyazás). Ugyanakkor az ilyen vezérlőfelületek általában nem igényelnek komplex metódusokat, valamint a távvezérelt eszközök funkcionalitását is csak elérhetővé teszik, és nem végeznek ahhoz kapcsolódóan számításokat, azaz különösebb értéket nem képviselnek a vezérlőfelületek.

Így akár elképzelhető lehet a jövőben az is, hogy az egyes gyártók publikusan elérhetővé teszik azokat a kulcsszavakat (üzenetek), amelyek segítségével vezérlik eszközeiket. Ennek eredményeképp egyedi vezérlőfelületek is tervezhetővé válnak bárki által, azaz a nyílt forráskódú szoftverfejlesztéshez hasonló folyamat indulhat el, mely segítheti az ergonomikus és kifinomult vezérlőfelületek fejlődését.

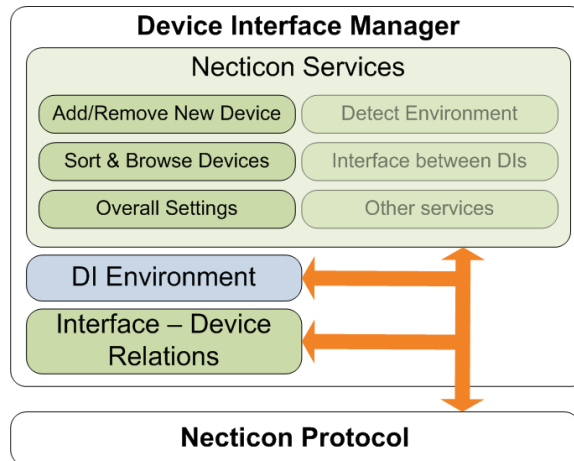
1.1.2. Interface – Device Relations

Ez a funkciócsoport felelős a vezérlőfelületek és a hozzájuk tartozó eszközök összerendelésért, az így alkotott adatbázis karbantartásáért, frissítéséért. Jelen esetben igen egyszerűen megvalósítható, hiszen elegendő eltárolni a Bluetooth MAC címet (Device ID), valamint a hozzátartozó DI webcímet és a böngészőazonosító-karakterláncot (User Agent String, mely az eszközök azonosítását segíti). Az adatbázisnak szintén részét képezi egy alaptáblázat, mely első megközelítésben az egyes gyártók vezérlőfelületeinek gyűjtőhelyeit (webszerverek címei) jelöli ki.

Ugyanakkor előfordulhat, hogy egyetlen eszközhöz több vezérlőfelület is megfelel (pl. szabadon fejlesztett megoldások), valamint több eszköz is tartozhat ugyanahhoz a vezérlőfelülethez (pl. több ugyanolyan eszköz azonos, vagy különböző környezetekben). Ezen szituációk feloldása is a relációkat tartalmazó csoport feladata, hiszen a lekérdezéskor egyértelmű válasszal kell szolgálnia.

1.1.3. Necticon Services

Jelen esetben több metódus és funkció is ide sorolandó (18. ábra). Az új eszközök telepítésének indítása, a már telepített eszközök különböző szempontok (típus, elhelyezkedés, környezet) szerinti böngészése és adott esetben eltávolítása, a rendszerszintű beállítások módosítása és a jövőben esedékes további fejlesztések, lehetőségek is ide sorolandók.



18. ábra – A Necticon Mobile által nyújtott fontosabb szolgáltatások

Az új eszköz telepítése annak érzékelésével kezdődik, melyet a Necticon Protokollon belüli eszközfelderítésért felelős funkciócsoport valósít meg. Miután egy eszköz érzékelésre került, rendelkezésre áll egy megfelelően részletezett terméknev (gyártó és típus, sorozatszám, azonosító, stb.), valamint egy, az adott technológiában értelmezett cím/azonosító. Ezen információk beírásra kerülnek az adatbázisba, majd a gyártóhoz tartozó vezérlőfelület gyűjtőhely kerül megnyitásra, egy a Necticon által generált böngészőazonosító karakterlánc segítségével (melynek része a terméknev). A szerver oldalon ennek megfelelően dönthető el, hogy melyik vezérlőfelület kerüljön letöltésre. Ugyanakkor nem szükséges jelen esetben minden történelmi konvenció betartása, hiszen az RFC1945 egyetlen termékleíró vagy kommentet ír elő kötelező jelleggel [59].

Miután a vezérlőfelület letöltődött, az alapértelmezés szerint eltárolásra kerül lokálisan a HTML5 offline funkcióinak megfelelően, valamint létrejön egy megfelelő bejegyzés az adatbázisban, ami megjeleníthetővé teszi az eszközt a böngészőfelületen. Az egész folyamat alatt a felhasználónak összesen háromszor (worst case) kell közreműködnie. Az első, amikor elindítja az új eszköz telepítését a második, amikor kiválasztja (ha csak egy új eszköz van, akkor ez a lépés átugorható) azt, majd a rendszer a további feladatokat automatikusan végzi. Az eszköz esetleges átnevezése beiktathat

(vagy a technológia kiválasztása) egy harmadik lépést is. Ugyanakkor a leggyakoribb az egy-két lépéses folyamat. Azaz a telepítési nehézségeket sikerült kiiktatni, az eszköz működésének megértése pedig a manapság egyre gyakrabban használt oktatófelületek segítségével, valamint ergonomikus elrendezéssel gyorsan megtörténhet.

A vezérlő felületek indítása még ennél is egyszerűbb. Egyetlen lépésből, az eszköz kiválasztásából áll (böngészés). A megfelelő vezérlőfelület a lokális tárból automatikusan betöltődik, a kapcsolat pedig (a technológia képességeihez mérten) gyorsan létrejön.

Bár a folyamat egyetlen lépést igényel, az eszközök megtalálása több eszköz esetén komplex feladat lehet. Éppen ezért további terveim között szerepel az eszközök környezetfüggő (GPS, cellainformációk, környező eszközök stb.) és gyakoriságon alapuló listázása, mely lerövidítheti a keresés által igényelt időt.

Mivel több felhasználó is hozzáférhet adott esetben ugyanazon eszközhöz, így célszerűnek tartom a későbbiekben egy (akár online kezelhető) profilrendszer (pl. vállalati beosztás szerint) megalkotását, mely a rendszert alkalmazhatóvá teszi nagyobb és több adminisztrációt igénylő területeken.

Az integrált és intelligens távvezérlés koncepciója is láthatóan megvalósítható, azonban számos interfész definiálását igényli, ugyanakkor jelen esetben ezek száma szorozódik a gyártók számával, csak az eszközök típusainak meghatározását igényli (és nem is feltétlenül lenne kötelező ezek támogatása).

1.2. Necticon Protocol

A Necticon Protocol feladata a rendszer által alkalmazott Necticon hálózat karbantartása, az adatküldés és fogadás, valamint a környező eszközökkel kapcsolatos információk biztosítása a felsőbb réteg (Device Interface Manager) számára. A protokoll ennek megfelelően több részlejárásból áll, melyek feladata az eszközben jelenlevő rádiós modul alkalmazásával a környező eszközök felderítése, a kapcsolatok kezelése, valamint az adatok (üzenetek) küldése és fogadása.

A következő két pontban először ismertetem a kommunikációs technológia és a távvezérelendő eszközök által elvárt alapfunkciókat, majd példajellelleggel ismertetem, hogy a Bluetooth Low Energy technológia esetén (melyre a megvalósítás során is építkezem) a fentebb felsorolt feladatok hogyan is valósíthatók meg.

1.2.1. A kommunikációs technológiával szemben támasztott követelmények

A megközelítés jellegéből fakadóan a következő elvárások fogalmazhatók meg egy, a Necticon rendszer adatátviteli alapját képező vezeték nélküli kommunikációs technológiával (és részben a távvezérelendő eszközökkel) szemben:

1. Tegye lehetővé az eszközök globálisan egyértelmű megkülönböztethetőségét.
2. Támogasson legalább egy, a felhasználó által kezdeményezhető, és a mobilkészülékek publikus API-jain keresztül elvégezhető eszköz-felderítési mechanizmust.
3. A kapcsolódás során, vagy azt követően szolgáltatson olyan információkat, melyek segítségével később azonosítani tudja a gyártó az adott eszközt (pl. sorozatszám, típusszám, fantázianév, stb.).
4. Rendelkezzen legalább egy jól definiált entitással (struktúra, csatorna, portszám, stb.), mely előzetesen ismert, vagy a felderítés és kapcsolatkiépítés során kideríthető, és amelynek segítségével a mobil készülékek publikus API-jain keresztül a kétirányú kommunikáció az eszközzel megtörténhet.

Az első kritérium, lényegében önmagyarázó, hiszen ahhoz, hogy ténylegesen az adott eszközzel történjen a kommunikáció, egyértelmű hivatkozási alapra van szükségünk. Fontos kitétel, hogy ezen azonosítók az idő során ne, vagy csak a rendszer számára determinisztikus, vagy követhető módon változzanak. Alkalmazhatók jelen esetben a különböző technológiák (Bluetooth, Wi-Fi) MAC címei, ugyanakkor az IP-címek csak fix kiosztás esetén tekinthetők relevánsnak, hiszen egyéb esetben azok dinamikusan változhatnak.

Az eszközök felderítésére vonatkozó feltétel szintén nyilvánvaló, hiszen, ha máskor nem is, az első telepítés alkalmával mindenképpen fel kell derítenünk az adott készüléket. Az általánosság feltétele révén csak publikusan elérhető API-kat célszerű használni, hiszen az egyes natív függvényhívások támogatottsága eszközönként is eltérhet. Ez utóbbi feltétel implicite azt is jelenti, hogy bármilyen heterogén (pl. Bluetooth – ZigBee, vagy WiFi – Zwave átjárók segítségével) hálózat állhat az API-k mögött.

Míg az első kritérium a hálózatban való elérhetőséget teszi lehetővé, addig a harmadik kritérium egy magasabb szinten, a gyártók vezérlő-felületeinek aggregációs pontjain teszik lehetővé a szükséges vezérlőfelület azonosítását. Ennél fogva a gyártók maguk rendelkezhetnek ezen információk formátumáról és tartalmáról, a Necticon csak

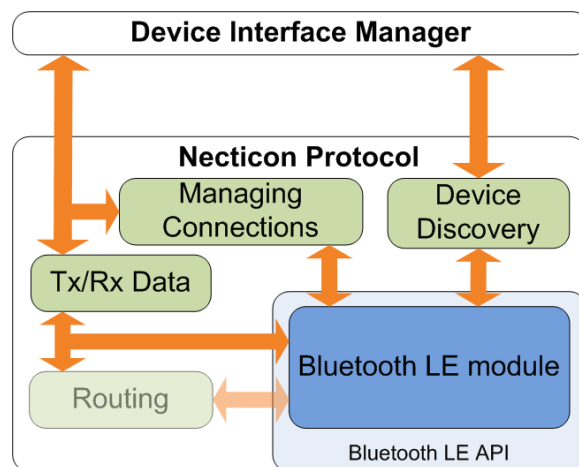
azt képes garantálni, hogy a böngészőazonosító-karakterláncba (user agent string) ezen információk beágyazásra kerülnek.

A negyedik feltétel egy fokkal összetettebb az előzőeknél. Az alapvető célja, hogy felderíthető, vagy ismert legyen a címen felül azon információ-többség, melynek segítségével olyan üzeneteket hozhatunk létre, hogy a céleszköz azokat értelmezni tudja. Jó példa erre a DLNA [55], ahol a felderítés UPNP-n történik, míg a tényleges kommunikáció HTTP-n, vagy RTP-n zajlik.

1.2.1. Bluetooth Low Energy, mint Necticon hálózat

Az alkalmazható technológiák egyike a Bluetooth v4.0 Low Energy változata, melynek segítségével az előbb felsorolt kritériumok egytől-egyig teljesíthetők. A technológia részletes bemutatásától a megoldás szempontjából való partikularitása miatt eltekintek és csak a releváns funkciókról értekezem [13].

A mobil készülékekben található Bluetooth LE modulok, a GAP (Generic Access Profile) rétegen belül Central eszközként vesznek részt a folyamatokban, ennek megfelelően a távvezérelt eszközök pedig (RCD) Peripheral szerepben. A GATT keretrendszerben az UIH kliensként, az RCD pedig szervertként van a jelen megoldás során számon tartva.



19. ábra – A Necticon Protocol feladatai a Bluetooth LE technológia környezetében

Eszközfelderítés:

Az eszközök felderítése a következőképpen történik: egy eszköz (UIH) hallgatózik a hirdetési csatornákon, a többi eszköz (RCD) pedig BLE hirdetéseket küld a megfelelő csatornákon. Ezekben a hirdetésekben helyezhető el az a terméknév, amely a későbbiekben az eszköz alapértelmezés szerinti nevét, valamint a böngészőazonosító alapját adja.

Kapcsolatok kezelése:

A kapcsolatok kiépítése alapján véve két folyamatot igényel. Az első maga a kapcsolat felépítése, míg a második folyamat a párosítás, mely a titkosított adatkommunikációt teszi lehetővé (ám ez természetesen opcionális). A kapcsolatok jelen esetben a direkt UIH-RCD közötti kapcsolatokat jelentik, azaz egyetlen ugrást reprezentálnak.

Üzenetek küldése és fogadása:

Bluetooth LE hálózatokban a kommunikáció alapvetően a GATT keretrendszer által nyújtott funkciók segítségével történik. A kiépült kapcsolatok felett a keretrendszerben, a szerver eszközön (RCD) definiált szolgáltatások (service) egyes változtatható értékeit (characteristics) van lehetőségünk írni és olvasni, azaz felfogható a kommunikáció egyszerű adatbázis kezelési feladatnak (lekérdezés, bevitel). A Necticon Protocol egy vagy több definiált Service-on belül akár több Characteristics értéket is használhat a funkciók betöltésére. Az egyes alkalmazott Service-ek és Characteristic értékeik pedig a GATT felderítés funkciójával kaphatók meg [13].

2. A specifikált rendszer előnyei

A rendszer egyik legfontosabb előnye, hogy lehetőség nyílik a fizikai vezérlőfelületektől való teljes elszakadásra, hiszen a rendszer - a megfelelő követelmények teljesítésével - támogatni képes olyan technológiákat is (pl. Bluetooth Low Energy, ZigBee, Z-wave), melyek - az IEA irányelveinek megfelelően - elegendően alacsony fogyasztással rendelkeznek, valamint elegendő mozgásteret hagynak az egyes elektronikus készülékek egyéb megvalósítási korlátaiból fakadó készlettel teljesítményéhez. A HTML5 megjelenítési képességeit (azaz közvetve a mobil készülékekét) kihasználva a fejlesztőket már nem kötik a fizikai megvalósítás korlátai, így sokkal ergonomikusabb vezérlőfelületek (interaktív menük, gombok, listák, ablakok stb.) alakíthatók ki, ami javítja mind a kezelhetőséget, mind a használhatóságot.

Ezen felül az egyes eszközök tényleges fizikai mérete, súlya csökkenhet, a kezelőfelületeket megjelenítő, kezelő vagy megvalósító elemek hiányában az eszközök energiatakarékossága javulhat, és ezeknek megfelelően az ökológiai lábnyom csökkenhet.

További előny, amely a fizikai vezérlőfelületek elhagyásából fakad, hogy az egyes készülékek gyártási költségei is csökkennek, az eszközök (Remotely Controlled Device) beépíthetősége és integrálhatósága is jóval egyszerűbbé válik. Ráadásként

lehetőség nyílik a nem sztenderd, egyéni készülékformák költséghatékonyabb kialakítására is.

A felsorolt előnyökön felül a rendszer egyik legfontosabb tulajdonsága, hogy teljes mértékben függetlenné teszi a vezérlőfelületek tervezését és azok alkalmazását a mobil platformoktól. A HTML5 jelenléte a szabványosságból fakadóan sokkal időtállóbb, mint a jelenleg elterjedt mobil készülékek által futtatott operációs rendszerek. Ezen felül jelen és jövő mobil operációs rendszereiben a HTML5 támogatottság várhatóan javulni fog, hiszen az interneten elhelyezett tartalmak megjelenítésének módját határozza meg. A platformfüggetlenség mind a készülékek fejlesztői, mind a felhasználók szempontjából növeli az egyértelműséget (bárhol elérhető), valamint csökkenti a használat és a fejlesztés erőforrásigényét, a korábbi megoldásokkal szemben.

Az alkalmazott modelltől fakadóan a Necticon által központosított DI elérés előnye, hogy a felhasználónak nem kell megjegyezni a felületeket megvalósító programok neveit, csak az RCD-ket kell nyilvántartania, amelyekhez a DI-k automatikusan kapcsolódnak. Nincs szükség arra sem, hogy a különböző párosítási folyamatokat fejben tartsa, vagy új eszköz esetén megtanulja, hiszen a Necticon egyetlen, adott módszer szerint rendeli össze az vezérlőfelületeket (DI) az RCD-kel. A centralizációnak további előnye, hogy a DI-k olyan alkalmazási területeken is könnyen konfigurálhatóvá, átláthatóvá és menedzselhetővé válnak, ahol akár százával található különböző RCD-k (egyetemek, áruházak, nagyobb irodaházak, gyárak stb.).

Mind a modell, mind annak interpretációjából fakadóan a rendszer számos továbblépési lehetőséget biztosít, hiszen a rendszer lehetővé teszi az egyes vezérlőfelületekkel, és az azok közötti direkt kommunikációt. Ennek megfelelően a mobil készülékek által nyújtott általános szolgáltatások beágyazhatóvá válnak a rendszerbe és funkcionalitásuk is központilag koordinálhatóvá válik a felhasználó, vagy más intelligens entitás által.

V. A kísérleti rendszer megvalósítása

Dolgozatom jelen szakaszában bemutatom a megvalósítás során használt eszközöket, ismertetem a tervezett rendszer komponenseinek implementálásával kapcsolatos lépéseim, majd értékelem a rendszer működését valós környezetben. Tekintettel arra, hogy a fejlesztés kísérleti jellegű volt, a jelen formában létrehozott rendszer csak a koncepció működését hivatott szemléltetni. Ebből kifolyólag igyekeztem a megvalósítás során az egyszerűsége törekedni.

1. Felhasznált eszközök

A Necticon mobil alkalmazás implementálásához olyan környezetre volt szükség, amely támogatja mind a Bluetooth Low Energy technológiát, mind a HTML5 szabványt, azon belül is az offline működéshez szükséges szolgáltatásokat. Tekintettel arra, hogy a fejlesztőkörnyezetekben elérhető eszköz-emulátorok egyike sem támogatja a Bluetooth LE funkciókat, valódi készülékre volt szükség. Megoldásom során Samsung Galaxy S3 készüléken implementáltam, mely támogatott minden szükséges funkciót, a Bluetooth LE technológiával egyetemben [56].



20. ábra - CC2540 Mini Development Kit [57]

A távvezérelendő eszközök alapjául szolgáló Bluetooth LE modul a Texas Instruments által gyártott, CC2540-es integrált áramköri lapkán alapuló modulokat választottam. A rendszer részét ennek megfelelően két darab mini fejlesztő csomag (CC2540 Mini Development Kit) képezte, melyben egy USB interfésszel ellátott, virtuális soros porton keresztül hozzáférhető modul és egy kulcstartószerű, gombellemmel táplálható kisebb modul (keyfob) található. Ez utóbbi modul két fizikai nyomógombbal, gyorsulás és hőmérsékletszenzorral, piezoelektromos hangszóróval, valamint ezen felül 2 LED-del van ellátva. A CC2540-es chip egy Intel 8051-es típusú 8 bites mikrovezérlőt tartalmaznak, melyen a Texas Instruments által fejlesztett BLE (Bluetooth Low Energy) stack is fut [57].

2. A rendszer implementációja

A rendszer implementálása három különböző fejlesztői környezetet és programozási paradigmát igényelt, az egyes rétegeknek megfelelően. A megvalósítást a legalsó réteggel, azaz a Bluetooth LE technológiát foganatosító távvezérelendő eszközök (RCD) programozásával kezdtem. Azt követve egy egyszerű HTML5 webalkalmazás került kifejlesztésre, végül a Necticon Mobile alkalmazást implementáltam az említett mobil készüléken.

2.1. A távvezérelendő eszköz megvalósítása

A CC2540-es lapkák alapjául szolgáló Intel 8051-es mikrovezérlő egység az IAR Embedded Workbench fejlesztőkörnyezetben szabadon fejleszhető. A Texas Instruments a Bluetooth LE stack alsóbb, fejlesztők által nem hozzáférhető rétegeit úgy valósította meg, hogy a rendszer programozása csakis az általa közzétett, bemutató jellegű kisebb alkalmazásainak módosításával valósítható meg. A fejlesztést megfelelően részletes dokumentáció hiányában a különböző demóalkalmazások „reverse engineering” módszerrel történő visszafejtésével kezdtem, majd a kívánt működésnek megfelelően módosítottam azokat [58].

Bár a koncepció alapvetően a távvezérelt eszközökkel, digitális interfészen keresztüli kommunikációt feltételez, jelen esetben a belső vezérlőegységben (8051) történik az üzenetek dekódolása és a feladatoknak megfelelő válaszok generálása a vezérlőfelületek felé. Tipikusan olyan esetekben lehetséges a rendszer ilyen módon való felhasználása, amikor a távvezérelt eszköz nem végez komplex funkciókat (pl. lámpák, redőnyök). A megoldás ugyanakkor egyedi hardverelem tervezésével kiterjeszhető digitális interfészen keresztüli kommunikációnak megfelelően is.

Egy egyszerűbb eszköz működési elvét követve regisztráltam a modulon (KeyFob) egy új szolgáltatást (Necticon Service) valamint az ahhoz tartozó Characteristics értékeket (RCD_CONTROL_POINT, RCD_RESPONSE). A megoldás alapjául a Texas Instruments KeyFob projektében található SimpleKeys profil szolgált. A profilok a Bluetooth terminológiában alkalmazási rétegbeli előírásokként, valamint külön definiált szolgáltatásokként értelmezendők, melyek egy adott funkciócsoport megvalósítását teszik lehetővé. Az ilyen profilok közül vannak szabványos (Bluetooth SIG által adoptált), valamint nem szabványos megoldások [59].

A SimpleKeys profil ez utóbbi csoportba tartozik, azaz nem szabványos megoldás, így annak módosítása nem ütközik a technológia alkalmazásának

konvencióival. Új profil létrehozására jelen esetben nincs mód, ugyanis saját tapasztalataim szerint lehetetlen a fordító által használt összeállítási fát (build path) módosítani.

Az általam módosított profilban található ezen felül az üzeneteket dekódoló logika, melynek működése alapvetően igen egyszerű. Az RCD_CONTROL_POINT távolról való írási igényét a profil érzékeli, majd, miután feldolgozta a kapott értéket, engedélyezheti az annak megfelelő lokális (nControl) változó írását (az adatbuszra is ebben lépésben helyezhető ki az üzenet). A logika, amely a kapott értéket jelen esetben feldolgozza, a modulon található két LED-et (zöld és piros) kapcsolja külön-külön ki és be, azaz összesen négy érték szerint szűri a bemenetet. A válasz generálása jelen esetben úgy történik, hogy a bemeneti érték visszacsatolásra kerül az RCD_RESPONSE-hoz tartozó lokális változóba (nResponse) majd, ha a GATT réteg jelzési funkciójához a regisztráció megtörtént a távoli eszköz felől (UIH), visszajelzi azt abban a formában. Az adatbuszról érkező válasz beolvasása egy, a hardveres funkciókat megvalósító rétegben (HAL - Hardware Abstraction Layer) elhelyezhető IT rutinban történhet meg, majd az előbbihez hasonló módon küldhető tovább az UIH eszköznek.

A profil működését a TI (Texas Instruments) által elérhetővé tett, Windows operációs rendszeren futtatható BTool (a fejlesztőcsomaghoz járó belső alkalmazás) és a SmartRF PacketSniffer [60] nevű programokkal ellenőriztem. Az első alkalmazás az USB modulok HCI-n (Host Controller Interface) keresztüli vezérlését egy grafikus felülettel segíti. Így sikerült elvégezni a távvezérlendő eszköz felderítését, az ahhoz való kapcsolódást, az RCD_RESPONSE-hoz tartozó jelzésekhez történő regisztrációt, valamint az RCD_CONTROL_POINT Characteristics értékek írását. A PacketSniffer alkalmazás a hirdetési és adatacsatornákon küldött csomagok monitorozását hivatott megvalósítani, így ennek segítségével követtem nyomon a két modul közötti kommunikációt. A profil a vártaknak megfelelően működött. A késleltetés a technológiából fakadóan igen alacsony volt (<0.5 ms).

2.2. A vezérlőfelületek létrehozása

A távvezérelendő eszközben alkalmazott profil funkcióinak megfelelő vezérlőfelület megvalósítása - a koncepciónak megfelelően - egy HTML5 alapon megírt webalkalmazás formájában történt meg. A fejlesztéshez jelen esetben egyszerű szövegszerkesztő programokat, a megjelenítéshez és teszteléshez pedig a Google Chrome (verzió: 37.0.2062.124 m) webböngészőt alkalmaztam. A megoldás jellegéből fakadóan szükség volt a webalkalmazás interneten történő elhelyezésére. A felhasznált

tárhely a HSZK (Hallgatói Számítógép Központ) által biztosított Ural2 szerveren, a felhasználónevemnek megfelelő publikus weboldalon található [56].

Az alkalmazás legfőbb komponensei az „index.html”, amely az alkalmazás belépési pontja a böngésző alkalmazások felől, a „style.css”, amely az egyes HTML5 elemek megjelenését definiálja, valamint a JavaScript könyvtárak, melyek az animációt és a mobil alkalmazásban levő interfész objektummal való kommunikációt teszik lehetővé. További fontos eleme a webalkalmazásnak a „manifest.mf” fájl, amely az offline tárolással kapcsolatos paramétereket tartalmazza.



21. ábra – Az implementált vezérlőfelület

A vezérlőfelület felépítése és működése az alkalmazott profilnak megfelelően igen egyszerű. A kezdeti állapotban a két mező jelzi az egyes LED-ek állapotát, valamint azokat megérintve, egy rövid animációt követően elérhetővé válnak az azokkal kapcsolatos szolgáltatások (ki- és bekapcsolás) gombok formájában.

A gombokra kattintva az azoknak megfelelő függvényekben (Main.js) generálódnak az üzenetek, majd Necticon Mobile interfészobjektumában definiált „Necticon.sendMessage(msg)” függvény kerül hívásra. Az RCD, azaz jelen esetben a kulcstartó modul által generált üzenet, aszinkron módon egy visszahurkolt függvényhívással kerül átadásra. Ennek lényege, hogy a válasz érkezésekor a Necticon Mobile meghív egy JS (JavaScript) függvényt (onRcdReply - Main.js), amelyben pedig egy, az interfész objektumban definiált „Necticon.getMessage()” függvény hívódik meg, amely visszatérési értéként adja át az RCD által generált üzenetet.

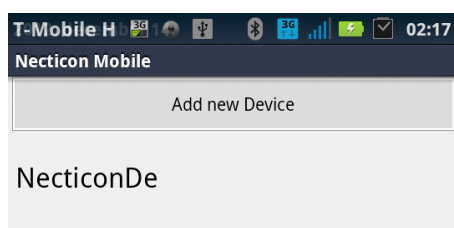
A visszatérési értéknek (azaz jelen esetben a küldött üzenetnek) megfelelően a mezők a nekik megfelelő LED aktuális állapotát (színét) öltik magukra jelezve ezzel, hogy a parancs végrehajtásra került. Az alkalmazás offline betöltődése és az egyes függvények működése a vártak megfelelően működött.

2.3. A Necticon Mobile implementációja

A Samsung Galaxy S3 készüléken futó Android (v4.3) operációs rendszerre történő fejlesztéshez az „Eclipse IDE for Android Developers” (verzió: 23.0.2) alkalmazásfejlesztői környezetet használtam.

A Necticon mobil alkalmazás több kisebb elemből, illetve belső alkalmazásból áll, így a következő pontokban az egyes alkotóelemek működési és implementációs kérdéseit fejtem ki. A korábbiaknak megfelelően itt is a rendszer egyszerűségének szemléltetése volt a cél.

2.3.1. MainActivity



22. ábra – Az alkalmazás kezdőképe

A MainActivity az alkalmazásrendszer belépési pontja. Működése a feladat jellegéből fakadóan igen egyszerű. A felhasználónak két választása van, vagy új eszközt ad hozzá a rendszerhez, vagy kattintással kiválasztja a listából azt az eszközt, amelyhez csatlakozni kíván. A MainActivity indulásakor szinkronizálja az eszközök listáját a már párosított BLE eszközök listájával, mely feladat lényegében megfelel a „Device Interface Relations (DIR)” adatbázis (DI_Database) karbantartásával. Annak okán, hogy jelen esetben ugyanolyan eszközökkel történik a kommunikáció, a relációk felvételekor alapértelmezésként a korábban említett HSZK webtárhely címét adtam meg.

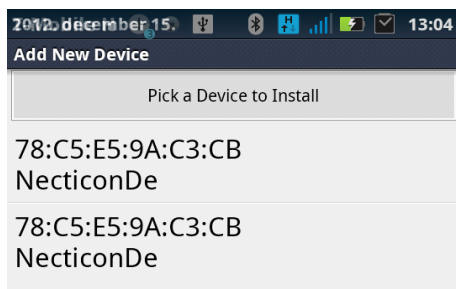
Amennyiben új eszközt szeretnénk hozzáadni, úgy a megfelelő gomb lenyomásával az „AddNewDevice” belső alkalmazás indul el, amely visszatérve az indító alkalmazáshoz, átadja az újonnan párosított eszköz címét és nevét, amely értékek felvevődnek a relációs adatbázisba.

Ha a felhasználó egy a listában már megtalálható eszközre kattint, úgy a „DevIntEnvironment” belső alkalmazás indul el, amely paraméterként megkapja a választott eszköz címét és a már példányosított DIR adatbázist.

2.3.2. DI Database

A „DI_Database” osztály a DIR adatbázist valósítja meg. Három listát tartalmaz, melyek az eszközök fizikai címét, azok nevét, valamint a hozzájuk tartozó vezérlőfelületek tárhelyeinek webcímét tartalmazzák. Az adatbázisba a relációk felvétele összehangolva történik, azaz az egyik listában levő elem indexe egyértelműen azonosítja a hozzá tartozó, másik listában levő információkat. Az egyes relációkat csak az eszközök címei azonosítják egyértelműen, így a későbbi hívások is ezeket veszik alapul. Jelen megoldásban az eszközök nevei nem módosíthatók, azaz megegyeznek azzal, amit a felderítéskor kapnak.

2.3.3. AddNewDevice

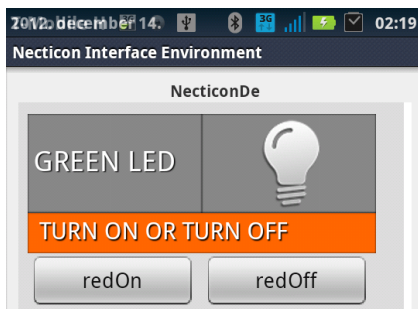


23. ábra – Új eszköz hozzáadása

Az új eszköz hozzáadására hivatott alkalmazás indulásakor az eszközfelderítés automatikusan elindul. A Bluetooth LE jellegéből fakadóan a folyamat igen gyorsan végbemegy (általában < 0,5 másodperc). A felderítést követően az alkalmazás ellenőrzi, hogy a definiált Necticon GATT szolgáltatást tartalmazza-e az adott készülék, és amennyiben igen, úgy listázza azt. A felhasználó ezt követően választhatja ki azon készüléket, melyet telepíteni kíván. A „telepítés” folyamata jelen esetben egy egyszerű listába való másolást jelent (DIR adatbázis), hiszen az első vezérlőfelület-indítás során töltődik le ténylegesen a távvezérlő alkalmazás. A folyamat az eszköz hozzáadása után a főalkalmazáshoz tér vissza.

2.3.4. DevIntEnvironment

Az egyes vezérlőfelületek futtatását lehetővé tevő alkalmazáskörnyezet a „Device Interface Environment” funkcióit valósítja meg. Jelen megoldásban Fragment osztályokat alkalmaztam, amelyek egy adott alkalmazáson (FragmentActivity) belül teszik lehetővé kisebb részalkalmazások (Fragment) futását. Mindezen felül, a váltás az egyes részalkalmazások között egyszerű „lapozással” történhet, így az egyes webalkalmazások (és közvetve a távvezérelt eszközök) közötti váltás gyorsan és kényelmesen történhet meg [62].



24. ábra

A környezetet vizuálisan egyetlen részalkalmazás tölti ki, melynek tetején egy szövegben jelenik meg az adott távoli eszköz neve, alatta pedig a kijelző legnagyobb részét elfoglaló, korábban említett WebView nézet helyezkedik el. A WebView nézet egy olyan böngészőnek tekinthető, amelynek nincs címsávja, és amelynek megjelenése és működése egyedileg befolyásolható [47].

Az alkalmazott WebView osztály példányosítását követően beállítottam az offline működéshez szükséges értékeket, majd hozzáadtam a „WebAppInterface” elnevezésű objektumot („Interface Object”), amely létrehozza a kapcsolatot a webalkalmazás és a Necticon Mobile között. A webalkalmazásban ennek megfelelően elérhetővé váltak az egyes funkciók JavaScript környezetből [47].

Mindezt követően, amennyiben nem került korábban tárolásra a webalkalmazás, úgy az letöltődik a készülékre, azt követően pedig már az alkalmazás lokális adattárából olvasható ki. Ennek működését tesztelendő, kikapcsoltam a Wi-Fi és a celluláris adathálózatokat a mobil készüléken, majd az alkalmazást újonnan indítva figyeltem meg a vezérlőfelület betöltődését, amely a vártnak megfelelően működött. Ráadásul a lokális adattárból történő olvasás lényegesen gyorsabb az alkalmazás letöltésénél, így a felhasználói élmény a már telepített eszközök esetén mindenképpen javul, noha az alkalmazás letöltése sem vett igénybe hosszabb időtartamot (<1 másodperc).

2.3.5. WebAppInterface

Az interfész objektumot, amely a vezérlőfelületek és az operációs rendszer között teremt meg a kapcsolatot, „WebAppInterface” névvel illetttem. Legfontosabb feladata az üzenetváltások közvetítése a webalkalmazás és a mobil operációs rendszer között. A vezérlőfelület leírásánál is említett függvényeken felül az objektum egy, a Necticon protokoll (jelen esetben Bluetooth LE vezérlés) feladatait ellátó „service” osztállyal kommunikált az Android belső üzenetváltási mechanizmusaival. Bár ezen feladat elméletben direkt módon is megoldható lett volna az egyszerűbb implementáció érdekében, a Samsung által implementált BluetoothGatt osztály korántsem működött zökkenőmentesen (csak különleges kontextusban volt képes regisztrálni a GATT specifikus „callback” függvényeket) [63].

Külön érdekesség volt az implementáció során, hogy az interfész objektum működése korántsem volt determinisztikus, holott az implementáció egyszerűségéből fakadóan minimálisra sikerült redukálni a hibalehetőségeket. Az „AndroidManifest.xml” konfigurációs fájlban, ha a „targetSdkVersion” és a „minSdkVersion” a 18-as API szintre volt állítva (ezen szinttől támogatott a Bluetooth Low Energy technológia), akkor nem került meghívásra a „sendMessage()” függvény, míg, ha a „targetSdkVersion” mezőt kihagytam, és a „minSdkVersion” a 10-es API szintre volt állítva, akkor nemhogy hibátlanul működött, de az animációk is érezhetően gyorsabban futottak le. A „hiba” forrása ez idáig ismeretlen.

2.3.6. Necticon Protocol

Az előbbieknek megfelelően a Necticon protokoll egy egyszerű Bluetooth LE funkciókat közvetítő és részben értelmező, háttérben futó szolgáltatásként jelent meg az alkalmazásrendszerben. A megfelelő üzenetek hatására kapcsolódott adott készülékekhez, felderítette a Characteristic értékeket, feliratkozott a notifikációkra (RCD_RESPONSE), valamint írta a megfelelő bejegyzéseket (RCD_CONTROL_POINT).

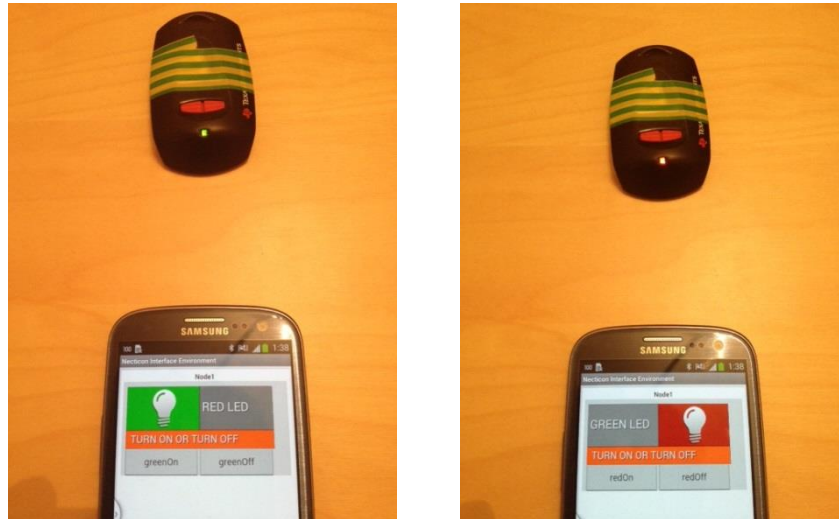
2.4. A rendszer értékelése

A mobil alkalmazás megvalósítása során tapasztalt hibajelenségek ellenére a tervezett rendszer teljes mértékben működőképes volt. A távvezérelt eszközök a vártaknak megfelelően igen kis késleltetéssel voltak képesek a feladatokat végrehajtani, ami a lehetséges alkalmazási területek és a felhasználó élmény szempontjából is kiemelt fontossággal bír. Mindezen felül a megoldás a későbbiekben kiterjeszhető a digitális adatbuszon történő kommunikációra, ami komplexebb feladatokat ellátó eszközök esetén is lehetővé teszi annak alkalmazását.

A webalkalmazások fejlesztése igen egyszerű, és az adott konvenciók, azaz az interfész objektummal történő kommunikációs előírások betartásával az interakció megtörténhet a mobil alkalmazás és a webalkalmazások között. Az előírások ezen felül egyszerűek, hiszen elegendő lehet három függvényt alkalmazni a távvezérlés megvalósításához (sendMessage, getMessage, onRcdReply).

A Necticon Mobile egyes komponensei szintén megfelelően működtek. Az eszközök kiválasztása és elindítása igen egyszerűen történhet meg egyetlen központi alkalmazáson keresztül. A telepítés és indítás nem igényel különösebb felhasználói aktivitást, azaz a felhasználói élmény javul és az erőfeszítések száma csökken. A

webalkalmazások betöltési ideje csak az első alkalommal igényel érzékelhető időt és internetkapcsolatot, a további indítások a lokális adattárból pillanatok alatt megtörténnek. A vezérlőfelületek közötti váltás szintén nem igényel különösebb aktivitást, elegendő egyszerűen lapozni közöttük. Az animációk végrehajtása általában folytonos, csak az első betöltés alkalmával akadoznak, ám akkor is csak igen csekély mértékben. A lokális adattárból történő indítás esetén ez már nem tapasztalható.



25. ábra – A rendszer komponensei működés közben

Összességében az alkalmazott modell és a működő komponensek fényében a megoldás hatékony, használata pedig egyértelmű és egyszerű. Valamint számos olyan szolgáltatást is lehetővé tesz, amely a jelenlegi módszerek esetén elérhetetlen (pl. gyors vezérlőfelület váltás, platform-független fejlesztés, gyors telepítés stb.).

Konklúzió

A napjainkban elérhető távvezérlési megoldások uralkodó módszere továbbra is a dedikált fizikai távvezérlőkkel történő vezérlés, mely a dolgozatban leírtaknak megfelelően számos korlátozó tényezővel rendelkezik, melyek közül a legfontosabb célhardverként vannak megvalósítva, azaz egyéb funkciókra a legtöbb esetben alkalmatlan, ezen felül ahány távvezérelendő eszköz van a környezetünkben, annyi távirányítóra van szükség. Kialakításukat tekintve vannak különböző megoldások, azonban a felhasználói élményt fokozó felületekkel rendelkező eszközök (érintés-érzékeny kijelző, mozgásszenzorok) jellemzően sokkal drágábbak a hagyományos típusoknál, valamint jellegükből adódóan nem feltétlenül alkalmasak minden eszköz vezérlésére.

A szoftveres formában (dedikált mobil alkalmazás) megvalósított távvezérlő alkalmazások ezen korlátozó tényezők közül számosat kiiktatnak, hiszen a fizikai megvalósítás korlátai így nem jelentkeznek, valamint egyetlen eszköz is alkalmas lehet több másik eszköz vezérlésére. Ugyanakkor az ilyen megoldások jellemzően csak a komplexebb szolgáltatásokkal, és nagyobb számítási kapacitással rendelkező eszközök esetén érhetők el, melynek egyik oka a kommunikációt megvalósító vezeték nélküli technológia (jellemzően Wi-Fi) számításiigénye, valamint maga a technológia elérhetősége, hiszen csak a távvezérlés érdekében az esetek legnagyobb többségében nem integrálnak az eszközökbe Wi-Fi modulokat.

Az általam tervezett rendszer egy olyan modellt alkalmaz, amely nem csak az alkalmazásokat központosítja a mobil készülékeken, hanem magát a távvezérlési folyamatot is, azaz a felhasználók és az eszközök közötti interakciós folyamatot. A központi entitás révén gyakorlatilag elhagyhatók a hagyományos értelemben vett alkalmazások és azok telepítési, valamint elérési nehézségei, hiszen a felhasználóknak csak egyetlen entitással szükséges ténylegesen kapcsolatba lépniük, ami az egyértelműséget, egyszersmind a felhasználó-közeliséget növeli. Ezen felül a modell jellegéből fakadóan (központi kezelés) lehetőség nyílik az egyes alkalmazások közötti kommunikációs interfészek biztosítására. További előnye a megközelítésnek, hogy nincs meghatározva annak a központi eszköznek a típusa vagy fajtája, amelyen a modellt alkalmazni lehet. Ami megkötés ilyen szempontból, az értelemszerűen a felhasználói interakciót megvalósító technológiák jelenléte, a megfelelő számítási kapacitás, valamint egy alkalmas vezeték nélküli technológia megléte.

A modell egyik lehetséges interpretációja az általam megvalósított rendszer (Necticon), ahol a központi entitás egy mobil készülék operációs rendszerén belül megvalósított mobil alkalmazás (Necticon Mobile), az egyes vezérlőfelületek pedig HTML5 webalkalmazások formájában megvalósított interaktív tartalmak. Az így megvalósított központi alkalmazás lehetővé teszi az eszközök gyors elérését és telepítését, azaz teljesen eszközalapú a megközelítés, ami a felhasználók számára egyértelmű, a különböző alkalmazások fantáziáneveivel szemben.

A példajelleggel alkalmazott Bluetooth LE technológia lehetővé teszi az eszközökkel történő gyors kapcsolat-felépítést és minimális késleltetéssel az üzenetváltást a vezérlőfelületek és a vezérelt eszközök között. Mindezen felül alacsony energiaigényéből fakadóan lehetőség nyílik a fizikai vezérlőfelületektől történő teljes elszakadásra, amely a jelenlegi fejlődési trendeknek megfelelően számos új lehetőséget hordoz magában (egyedi készülékformák, ökológiai lábnyom csökkentése, nagyfokú integrálhatóság. stb.).

Ugyanakkor a Necticon implementációja során létrehozott egyéb komponensek hibátlanul funkcionáltak (valamint ami hibás volt, az is csak az implementáció módja miatt, hiszen léteznek olyan alkalmazások, amelyek hibátlanul működnek), ami összességében alátámasztja azt a tézisemet, hogy létezik és megvalósítható olyan elgondolás, amely képes nagymértékben egyszerűsíteni, gyorsítani és egyértelművé tenni az eszközökkel történő interakció elérését és megvalósítását, mind a felhasználók, mind a fejlesztők szemszögéből.

A felhasználók szempontjából a szükséges feladatok egyértelműek és nincs szükség a vezérlőfelületek külön telepítésére, valamint a folyamathoz szükséges jóváhagyásokra és lépésekre. Ezen felül a kommunikációt megvalósító technológiához nincs szükség a felhasználó felől semmilyen jellegű beavatkozásra, azaz a hálózat önszerveződőnek tekinthető.

A fejlesztők szempontjából külön könnyebbség, hogy nincs szükség minden alkalmazásplatform mély ismeretére, hiszen egyetlen szabványos (draft) platformon (HTML5) történhet a vezérlőfelületek fejlesztése, amely ráadásul lényegében semmilyen különleges fejlesztői környezetet nem igényel (szövegszerkesztőre és böngészőre van csak szükség). A platform pedig időtálló (hiszen az interneten elérhető tartalmak megjelenítését szabályozza), és a fejlesztési irányelvek szerint kompatibilis a korábbi változatokkal.

Mindezen felül a rendszer számos továbblépési lehetőséget rejt magában, amely több, korunk komplex eszközeivel és szolgáltatásával kapcsolatos problémára is megoldást adhat. Amennyiben a hordozó hálózat véglegesítésre kerül és a többgrásos ad-hoc útvonalak is létrehozhatók, létrejöhet egy olyan kiterjedt vezérlési hálózat, amely alkalmas lehet több eszköz intelligens együttműködésére egy adott környezetben. Ezen felül az így létrejövő hálózat alkalmas lehet egyéb vezetékes és anélküli kommunikációs csatornák vezérlésére is, így akár önszerveződő jelleggel történhet az egyes eszközök közötti médiatartalom továbbítása, a felhasználó bármilyen jellegű beavatkozása nélkül. A fizikai vezérlőfelületektől való elszakadás, a korunkban egyre jellemzőbb webáruházak fejlődését is továbblendítheti, az adott eszközökhöz tartozó, immáron virtuális vezérlőfelületek futtatásával, függetlenül az eszközök jelenlététől. A rendszer okostelevíziókra (UIH) történő kiterjesztésével pedig az „intelligens otthonok” központi intelligenciája állandó jelleggel elérhetővé válik.

Irodalomjegyzék

- [1] N. Tesla, „Method of and Apparatus for Controlling Mechanism of Moving Vessels and Vehicles”. USA Szabadalom száma: 613809, 8 november 1898.
- [2] „Consumer IR,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Consumer_IR. [Hozzáférés dátuma: 18 október 2014].
- [3] Wikipedia, „Remote Control,” [Online]. Available: http://en.wikipedia.org/wiki/Remote_control#Radio_remote_control_systems. [Hozzáférés dátuma: 12 október 2014].
- [4] „cc2541 - 2.4-GHz Bluetooth® low energy and Proprietary System-on-Chip,” Texas Instruments, 2012. [Online]. Available: <http://www.ti.com/product/cc2541>. [Hozzáférés dátuma: 15 október 2014].
- [5] P. Industrial, „Wireless Modules,” 2008. [Online]. Available: http://www.digikey.com/Web%20Export/Supplier%20Content/Panasonic_10/PDF/Panasonic_PIC_WirelessModules.pdf?redirected=1. [Hozzáférés dátuma: 12 október 2014].
- [6] „802.11n,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/802.11n>. [Hozzáférés dátuma: 18 október 2014].
- [7] „Bluetooth,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Bluetooth>. [Hozzáférés dátuma: 18 október 2014].
- [8] „Zigbee,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Zigbee>. [Hozzáférés dátuma: 12 október 2014].
- [9] „Z-Wave,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Zwave>. [Hozzáférés dátuma: 12 október 2014].
- [10] B. G. A. S. D. W. Daniel Halperin, „Demystifying 802.11n Power Consumption,” in *Proceedings of the 2010 international conference on Power aware computing and systems*, USENIX Association Berkeley, CA, USA ©2010, 2010.
- [11] „Wi-Fi Direct,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/WiFi_Direct. [Hozzáférés dátuma: 12 október 2014].

- [12] P. E. D. GmbH, „Class 1 or Class 2 Bluetooth Module Specification,” 2 július 2011. [Online]. Available: <http://panasonic.com/industrial/includes/pdf/PAN1315%20Core%20Specification.pdf>. [Hozzáférés dátuma: 12 október 2012].
- [13] „Core Specification v4.0,” Bluetooth SIG, 30 június 2010. [Online]. Available: <http://www.bluetooth.org/>. [Hozzáférés dátuma: 18 október 2014].
- [14] T. Instruments, „A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications,” [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc2530.pdf>. [Hozzáférés dátuma: 12 október 2014].
- [15] „RMC-QTD1,” Samsung, [Online]. Available: <http://www.samsung.com/nz/consumer/tv-audio-video/accessories/accessories/RMC-QTD1AP2/XY>. [Hozzáférés dátuma: 12 október 2014].
- [16] engadget, „NSZ-GS7,” Sony, [Online]. Available: <http://www.engadget.com/products/sony/nsz-gs7/>. [Hozzáférés dátuma: 12 október 2014].
- [17] „Magic Remote - Mozgásérzékelős távirányító görgővel a 2012/13-as LG Smart Televíziókhoz,” LG, [Online]. Available: <http://www.lg.com/hu/tv-kiegeszitok/lg-AN-MR300-magic-remote>. [Hozzáférés dátuma: 12 október 2014].
- [18] „Amazon,” [Online]. Available: <http://www.amazon.com/>. [Hozzáférés dátuma: 15 november 2014].
- [19] techspot, „Logitech Harmony 900,” Logitech, [Online]. Available: <http://www.techspot.com/products/remotes/logitech-harmony-900.47899/>. [Hozzáférés dátuma: 12 október 2014].
- [20] „Mobile device,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Handheld_device. [Hozzáférés dátuma: 13 október 2014].
- [21] „Slideshow: 14 Android Home Theater Apps,” ElectronicHouse, [Online]. Available: <http://www.electronichouse.com/slideshow/category/11380/1514>. [Hozzáférés dátuma: 13 október 2014].

- [22] „One Watt Initiative,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/One_Watt_Initiative. [Hozzáférés dátuma: 10 október 2014].
- [23] „Smart Home,” Internet of Things China, [Online]. Available: <http://iotchina.net/smart-home/>. [Hozzáférés dátuma: 13 október 2014].
- [24] „Kinect,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Kinect>. [Hozzáférés dátuma: 18 október 2014].
- [25] „Cheoptics360,” Vizoo, [Online]. Available: <http://www.vizoo.com/flash/>. [Hozzáférés dátuma: 18 október 2014].
- [26] „Volumetric display,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Volumetric_display. [Hozzáférés dátuma: 18 október 2014].
- [27] „The Future of Smart TV, Now,” Samsung, 2012. [Online]. Available: <http://www.samsung.com/sg/smarttv/#navigation>. [Hozzáférés dátuma: 18 november 2012].
- [28] „EEG Biosensors,” NeuroSky, [Online]. Available: <http://neurosky.com/products-markets/eeg-biosensors/>. [Hozzáférés dátuma: 18 október 2014].
- [29] emotiv, 2012. [Online]. Available: <http://emotiv.com/>. [Hozzáférés dátuma: 18 október 2014].
- [30] D. Lee, „From sketch to smartphone - how HTC drew inspiration from a shoe,” BBC, 19 szeptember 2012. [Online]. Available: <http://www.bbc.com/news/technology-19646188>. [Hozzáférés dátuma: 12 október 2014].
- [31] „Smartphone,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Smartphone>. [Hozzáférés dátuma: 12 október 2014].
- [32] C. Terrance Gaines, „Tablets are all about Mobile Productivity,” [Online]. Available: <http://www.marioarmstrong.com/2011/03/29/tablets-are-all-about-mobile-productivity/>. [Hozzáférés dátuma: 12 október 2014].
- [33] „Tablet Computer,” Wikipedia, 24 november 2012. [Online]. Available: http://en.wikipedia.org/wiki/Tablet_computer. [Hozzáférés dátuma: 25 november 2012].

- [34] K. Wanzi, „Why We Should Embrace Bring Your Own Device Day,” 20 március 2014. [Online]. Available: <https://sg.news.yahoo.com/why-embrace-bring-own-device-day-075525176.html>. [Hozzáférés dátuma: 12 október 2014].
- [35] „Laptop,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Laptop>. [Hozzáférés dátuma: 18 október 2014].
- [36] „PadFone,” Asus, [Online]. Available: <http://www.asus.com/Mobile/PadFone/>. [Hozzáférés dátuma: 25 november 2014].
- [37] „Laptop és átalakítható ultrabook,” Lenovo, [Online]. Available: <http://shop.lenovo.com/hu/hu/convertibles/>. [Hozzáférés dátuma: 13 október 2014].
- [38] „Surface,” Microsoft, [Online]. Available: <http://www.microsoft.com/Surface/en-US>. [Hozzáférés dátuma: 13 október 2014].
- [39] B. Ádám, „Nyolcvan százalék felett az Android részesedése,” HWSW, 14 november 2013. [Online]. Available: <http://www.hwsz.hu/hirek/51293/gartner-okostelefon-piac-android-ios-windows-phone-blackberry-samsung-apple-lenovo-huawei-lg.html>. [Hozzáférés dátuma: 13 október 2014].
- [40] „Android (operating system),” Wikipedia, [Online]. Available: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)). [Hozzáférés dátuma: 18 október 2014].
- [41] „About the iOS Technologies,” Apple, [Online]. Available: <http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>. [Hozzáférés dátuma: 18 október 2014].
- [42] „HTML5,” Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/HTML5>. [Hozzáférés dátuma: 18 október 2014].
- [43] „WebKit,” Wikipedia, [Online]. Available: <http://hu.wikipedia.org/wiki/WebKit>. [Hozzáférés dátuma: 12 október 2014].
- [44] „Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers,” [Online]. Available: <http://caniuse.com/>. [Hozzáférés dátuma: 18 október 2014].
- „UIWebView Class Reference,” Apple, [Online]. Available:

- [45] http://developer.apple.com/library/ios/#documentation/uikit/reference/UIWebView_Class/Reference/Reference.html. [Hozzáférés dátuma: 12 október 2014].
- [46] „Max size of WebSQL/SQLite database inside UIWebView (phonegap),” Stackoverflow, 21 május 2012. [Online]. Available: <http://stackoverflow.com/questions/9118757/max-size-of-websql-sqlite-database-inside-uiwebview-phonegap>. [Hozzáférés dátuma: 12 október 2014].
- [47] „android.webkit,” Google, 16 november 2012. [Online]. Available: <http://developer.android.com/reference/android/webkit/package-summary.html>. [Hozzáférés dátuma: 18 október 2014].
- [48] „Web Storage,” W3C, 14 május 2014. [Online]. Available: <http://dev.w3.org/html5/webstorage/#disk-space>. [Hozzáférés dátuma: 12 október 2014].
- [49] „Issue 10546: No html5 audio tag support in WebView,” Google, [Online]. Available: <http://code.google.com/p/android/issues/detail?id=10546>. [Hozzáférés dátuma: 13 október 2014].
- [50] „WebView and HTML5 <video>,” Stackoverflow, [Online]. Available: <http://stackoverflow.com/questions/3815090/webview-and-html5-video>. [Hozzáférés dátuma: 14 október 2014].
- [51] „Ubiquitous_computing,” Wikipedia, 24 október 2012. [Online]. Available: http://en.wikipedia.org/wiki/Ubiquitous_computing. [Hozzáférés dátuma: 25 november 2012].
- [52] „Smart TV,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Smart_TV. [Hozzáférés dátuma: 13 október 2014].
- [53] „Project Glass,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Project_Glass. [Hozzáférés dátuma: 13 október 2014].
- [54] „Hypertext Transfer Protocol -- HTTP/1.0,” World Wide Web Consortium, május 1996. [Online]. Available: <http://www.w3.org/Protocols/rfc1945/rfc1945>. [Hozzáférés dátuma: 18 október 2014].
- [55] „Digital Living Network Alliance,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Digital_Living_Network_Alliance. [Hozzáférés

dátuma: 13 október 2014].

- [56] „Galaxy S3,” Samsung, [Online]. Available: <http://www.samsung.com/hu/consumer/mobile-phone/touch-screen/touch-screen/GT-I9300MBDXEH>. [Hozzáférés dátuma: 21 október 2014].
- [57] „CC2540 Mini Development Kit,” Texas Instruments, [Online]. Available: <http://www.ti.com/tool/cc2540dk-mini>. [Hozzáférés dátuma: 15 október 2014].
- [58] „Bluetooth Low Energy Wiki,” Texas Instruments, [Online]. Available: <http://processors.wiki.ti.com/index.php/Category:BluetoothLE?DCMP=blestack&HQS=ble-wiki>. [Hozzáférés dátuma: 14 november 2012].
- [59] „Services,” Bluetooth SIG, [Online]. Available: <https://developer.bluetooth.org/gatt/services/Pages/ServicesHome.aspx>. [Hozzáférés dátuma: 15 október 2014].
- [60] „SmartRF Packet Sniffer,” Texas Instruments, [Online]. Available: <http://www.ti.com/tool/packet-sniffer>. [Hozzáférés dátuma: 22 október 2014].
- [61] „Hallgatói Számítógép Központ,” BME, [Online]. Available: <http://www.hszk.bme.hu/>. [Hozzáférés dátuma: 11 november 2012].
- [62] „FragmentActivity,” [Online]. Available: <http://developer.android.com/reference/android/support/v4/app/FragmentActivity.html>. [Hozzáférés dátuma: 18 október 2014].
- [63] „SOLVED: GATT callback fails to register,” StackOverFlow, [Online]. Available: <http://stackoverflow.com/questions/20069507/solved-gatt-callback-fails-to-register>. [Hozzáférés dátuma: 22 október 2014].

Rövidítésjegyzék

API	Application Programming Interface
BLE	Bluetooth Low Energy
CSS	Cascading Style Sheets
CPU	Central Processing Unit
DB	Database
DI	Device Interface
DIR	Device Interface Relations
DLNA	Digital Living Network Alliance
EEG	Electroencephalography
GATT	Generic Attribute Framework
GPS	Global Positioning System
GPU	Graphics Processing Unit
HSZK	Hallgató Számítógép Központ
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ID	Identifier
ISM	Industrial Scientific and Medical
IR	InfraRed
IEA	International Energy Agency
IP	Internet Protocol
JS	JavaScript
LED	Light Emitting Diode
LE	Low Energy
M2M	Machine to Machine
MAC	Medium Access Control
OOK	On-Off Keying
OS	Operation System
P2P	Peer to Peer
PC	Personal Computer
PDA	Personal Digital Assistant
RTP	Real-time Transport Protocol
RCD	Remotely Controlled Device

SMS	Short Message Service
SIG	Special Interest Group
SQL	Structured Query Language
TV	Television
TI	Texas Instruments
UDA	Unified Device Access
UPNP	Universal Plug and Play
USB	Universal Serial Bus
UI	User Interface
UIH	User Interface Host
WI-FI	Wireless Fidelity
W3C	World Wide Web Consortium