

Óbudai Egyetem
Neumann János Informatikai Kar
Szoftvertchnológia Intézet

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

**BLUETOOTH ALAPÚ
KAPCSOLÓRENDSZER MOBIL
TÁMOGATÁSSAL**

Szerző: **Horváth Zsófia**
mérnök informatikus szak, IV. évf.

Konzulens: **Dr. Turóczy Antal**
adjunktus

1. Tartalomjegyzék

1. Tartalomjegyzék.....	2
2. Rövid tartalmi összefoglaló.....	3
3. Bevezetés.....	4
3.1. Hasonló rendszerek.....	6
3.2. Célkitűzés.....	11
3.3. Beágyazott rendszer.....	12
3.4. Firmware.....	12
3.5. Programozó készülék.....	13
3.6. Kommunikáció.....	14
4. Mikrokontrolleres rendszer tervezése funkcionális szemlélettel.....	16
4.1. Specifikáció.....	16
4.2. Funkcionális blokkvázlat.....	18
4.3. Hardver-szoftver szétválasztás.....	20
4.4. Hardver rendszerterv.....	22
4.5. Megvalósítás.....	23
4.6. Szoftver rendszerterv.....	32
4.7. A rendszer tesztelése.....	35
5. Felhasználói szoftverek fejlesztése objektum orientált szemlélettel.....	37
5.2. Mobileszközre szánt szoftver tervezése.....	38
5.3. Szoftvertesztelés.....	45
6. Továbbfejlesztési lehetőségek.....	46
7. Összegzés.....	47
8. Ábrajegyzék.....	48
9. Irodalomjegyzék.....	49

2. Rövid tartalmi összefoglaló

A TDK munka bemutatja, hogy milyen módon lehetséges megvalósítani egy SmartHome alkalmazást. A dolgozat kitér a különböző rendszerek bemutatására és összehasonlítására.

A tervezési munkafolyamat két fő részre bontott, az egyik a mikrokontrolleres rendszer tervezése funkcionális szemlélettel, a másik pedig a PC, illetve mobil eszköz szoftvertervezése objektum orientált szemlélettel.

A dolgozat részletesen ismerteti a hardver rendszer, firmware és felhasználói szoftver tervezési és megvalósítási lépéseit, valamint a feladat során elkészített felhasználói felületeket.

Dolgozatomban törekedtem arra, hogy a munkám reprodukálható és ezáltal megérthető legyen. Megpróbáltam a tervezést logikusan és következetesen felépíteni, az ilyen rendszerekre vonatkozó általános tervezési és dokumentációs eljárások formális elvárásait megtartva. Célom az elvi elgondolás bemutatása volt, és annak megvilágítása mennyire könnyen kiterjeszthetők az okosotthon rendszerek a mindennapi életre.

3. Bevezetés

Fejlődő világunkban egyre nagyobb szerephez jutnak a mindennapi életünket megkönnyítő alkalmazások, az energiatakarékos és költségkímélő berendezések az otthonainkban. Az ilyen rendszerek az otthon automatizálás témakörébe tartoznak.



1. ábra. SmartHome illusztráció

Az otthon automatizálás a lakossági kiterjesztése az épületautomatizálásnak. Ez magában foglalhatja a központi világítást, a HVAC-t (fűtés, szellőzés és légkondicionálás), a házimozi rendszereket, a kerti locsolót és más rendszereket. A lakásautomatizálás nagy jelentőséggel bír az idősek, fogyatékkal élők és rendszeres ellátásra szorulóknak életében. Tágabb értelemben ide tartoznak azok a telemedicina berendezések, melyek a gyógykezelés helyes végrehajtását és betartását felügyelik.

A jövő otthonai már régóta jelen vannak science fictionként. A komplexitása, a versengés, a nem egységes szabványok és az ebből fakadó költségek miatt korlátozottan jelent meg az otthonokban. Az első „otthon vezérlőnek” nevezhető rendszer, egy kísérleti rendszer volt 1966-ban.

A SmartHome egy szórakoztató elektronikai kiskereskedés, ami a házautomatizálási termékekre, információkra, „Hogyan csináljuk” videókra és ipari információkra szakosodott.

1992-ben alapította III. Joe Dada, aki az igazgatója is. Székhelye Irvine, Kalifornia. Barátoktól és családtagoktól kapott donációkból, megtakarításokból és kis kölcsönökből indították be. Az első Smarthome katalógust 1992-ben adták ki. A SmartLabs Inc. mára 3 vállalatot foglal magában: A Smarthome Directet, ami a Smarthome.com-ot foglalja magába; a SmartLabs Design-t, ami az otthon vezérlésére szolgáló termékek tervezésével foglalkozik; és a SmartLabs Technology-t, ami az úttörő építészeti technológiával, az INSTEON-nal foglalkozik.

Mára a SmartHome több mint 4 millió katalógust nyomtat évente és a szolgáltatásaikat már több mint 130 országban használják.

Mivel az okosotthon tervezésnél problémát jelentett a nem egységes szabvány rendszer, így ők megalkottak egyet, amiben minden általuk árult és forgalmazott termék beletartozik, a villanykörtétől a hűtőszekrényig. A rendszerük egy két frekvenciasávós otthoni mesh hálózat, ami AC-tápvezetékkel és rádió frekvenciás protollokat használ az eszközökkel való kommunikációhoz.

Ezek az eszközök egy számítógépes hálózatba kapcsolódnak, amit egy központi számítógép vezérel, és lehetővé teszi a távoli hozzáférést is internet kapcsolaton keresztül. Az információs technológiák otthoni környezetbe való

integrálása révén, a rendszerek és berendezések képesek kommunikálni integrált módon, ami kényelmet, energiahatékonysági és biztonsági előnyöket nyújt.

Tipikusan egy új otthon automatizálása a ház kivitelezésének a része, hogy a ház átrendezhetősége rugalmasan megtörténhessen. Az elkészült házakat általában vezeték nélküli rendszerekkel rendezik be, ezzel csökkentve az esetleges vezetékváltozásokkal járó bonyodalmakat. Az általam tervezett és kivitelezett rendszer, ez alapján, a szabvány alapján működő rendszerekhez illeszthető.

3.1. Hasonló rendszerek

A magyar kereskedelemben megjelenő lehetőségek között nem nagy a választék. A legtöbb készlet rádiójel vezérlésű dugaszoló aljzat kapcsolót és a hozzá tartozó távirányítót tartalmazza.

A csatlakozó aljzatok betanítására nincs szükség, mivel rögzítetten beállított kódolásúak. Viszont hátrányuk, hogy így csak a saját távirányítójukkal működtethetők, és más rendszerbe nem kapcsolhatók, illetve így a távirányító gombjaitól függ a vezérelhető készülékek száma.

Viszonylag könnyen belátható, hogyha a lakás összes elektromos eszközéhez szeretnék kapcsolót elhelyezni, akkor vagy nagyon sok távirányítót kell használni, vagy egyetlen, de rengeteg gombbal rendelkező távirányítót.

Ráadásul a be- és kikapcsoláshoz külön gombokat kellene, ami megkétszerezi a szükséges gombok számát.



2. ábra. Távirányítós



3. ábra.
Infravörös

Erre megoldást jelenthetnek az olyan infravörös távkapcsolók, melyek akár a TV távirányítójával is működnek. Viszont a beszerelésük nehézkes, mivel nem lehet a konnektorba illeszteni, így kénytelen a felhasználó magam bekötni, vagy fizetni egy szerelőnek, aki megcsinálja. Ezeknek a rendszereknek viszont mindenképp előnyük, hogy már nem kell külön távirányító hozzá. Ennél a megoldásnál van 4, 8 vagy 12 órás kapcsolási funkció is. Ami hasznos funkció, de ilyen kötött idő intervallumokkal nem könnyű használni.

A piacon kapható kényelmesebb időzítéssel rendelkező eszközöknek is vannak hiányosságai. A legnagyobb gondjuk, hogy nem távirányítóval vezérelhetők, hanem nyomógombbal állíthatók. Ez kényelmetlen lehet egy átlagembernek, de a mozgás korlátozottakra gondolva, hamar belátható, hogy számukra a rendszeres lehajlás és aljzatba helyezés gyakorlatilag lehetetlen, főleg sok készülék esetén. Tehát mindenképp érdemes ezeket a kapcsolókat központilag is kezelhetővé tenni.

Elsőnek kerül bemutatásra az alábbi visszaszámlálós kapcsoló. Fixen beállított 60 perces visszaszámlálási idővel rendelkezik, ez azt jelenti, hogy a 60 perces időn nem lehet módosítani. A stop gomb megnyomásával a visszaszámlálás megszakítható, ekkor viszont nem lehet folytatni a megszakítástól, hanem visszaáll a 60 perces alaphelyzetbe és a start gomb megnyomásával előlről kezdődik a visszaszámlálás. Ez elég kevésbé hatékony, korlátozott funkciókkal bír, és a gyakorlatban való alkalmazására is kevés életszerű példát lehet említeni.



4. ábra.
Visszaszámláló

A következő eszköz már percre pontosan állítható, de csak heti rendszerességre. Egyszeri kapcsolási funkcióval nem bír. 4 programcsomag van:

H/Sze/P, H-Sze, Cs-Szo, K/Cs/Szo, ezen kívül rendelkezik visszaszámlálási funkcióval, de már állítható a timer. A maximum visszaszámlálási idő 23 óra 59 perc. Viszont még mindig csak nyomógombokkal közvetlenül programozható, illetve az adatokat csak addig őrzi meg, amíg van tápellátása.



5. ábra. Digitális időkapcsoló

A következő modell intelligens idővezérlésű, ami azt jelenti, hogy nem előre beállított visszaszámlálási funkcióval rendelkezik, hanem tetszés szerint beállítható egy kívánt üzemidő szintén nyomógombok segítségével, majd ez idő lejártá után kikapcsolja az elektromos eszközt.



6. ábra.
Energiatakarékos
kapcsoló



7. ábra. 1-9 óra
közötti
visszaszámláló

Ennek egy másik verziója, ami 1 és 9 óra közötti időpont beállítására képes. Mindkettőn csak elindítani lehet a visszaszámlálást, de megszakítani, szüneteltetni nem.

A feladatomat leginkább megközelítő kapcsoló, már nemcsak heti rendszerességre programozható, hanem napi blokk képzésre is alkalmas, viszont

az egyszeri, dátumhoz kötött kapcsolás még ebben sincs meg, illetve csak napi 20 kapcsolásra képes, és a vezérlés még mindig nyomógombos.



8. ábra. Napi időzítésre alkalmas

A fent felsorolt kapcsolók közös jellemzője, hogy csak egy pólusúak, ami életvédelmi szempontból nem jó döntés, mert célszerű az elektromos eszközöknél mind a nullát, mind a fázist megszakítani. A feladat tervezése közben ez is fontos szempont volt. Ilyen megbízhatósággal rendelkező kapcsolóból csak kettőt találni a piacon.

Az egyik heti programozhatósággal rendelkezik, a következő blokkok képezhetők: Hé-Va, Hé-Szo, Hé-Pé, Hé-Sze-Pé, Ke-Csü-Szo, Szo-V, viszont csak 16 be/kikapcsolási programmal bír. Ennél is nyomógombokkal történik a kezelés.



9. ábra. Digitális időkapcsoló - kétpólusú



10. ábra. Mechanikus időkapcsoló - kétpólusú

A másik csak egyszeri időpont beállítására alkalmas, és bár nem nyomógombos, de a kezelhetősége még mindig körülményes, mivel mechanikus kapcsolóval rendelkezik. Érdeemes áttérni az analóg kapcsolókra, de csak annyiban különböznek, hogy a beállítás mechanikus kapcsolóval történik.

Beállítási lehetőségeiket tekintve, ugyanolyan széles skálán mozognak, mint a digitális kapcsolók.

A fent ismertetett kapcsolók közös jellemzője, hogy a téli/nyári időszámítás átkapcsolása vagy nem lehetséges, vagy nem automatikusan zajlik. Ezen kívül a kezelésükhöz vagy egy saját távirányító szükséges, vagy csak a szerkezeten található nyomógombokkal kezelhető, és a legtöbbjük előre meghatározott blokk képzések szerint használható.

Egy hazai okosház tervezéssel foglalkozó cég kínálatában, nem is találhatók meg ilyen típusú időzített kapcsolók, csak világítás- és fűtésvezérlésre alkalmas és beszerelési munkákat igénylő vezérlőkkel rendelkeznek.

A külföldi piacot átnézve, az itthonihoz hasonló képességű eszközöket találunk. A kapcsolók képességeit megnézve, szembetűnő, hogy, amik képesek napi vagy heti riasztásra, azok egyszeri riasztásra nem képesek, a beállítás pedig nyomógommbal vagy mechanikus kapcsolóval történik, illetve korlátozott a napi be- és kikapcsolások száma.

Mindezen hiányosságokat kívánom kiküszöbölni a saját rendszeremben. Kézenfekvő megoldás, hogy az általam tervezett eszköz számítógéphez csatlakoztatható legyen, és a kezelése ezen keresztül történjen. Ezzel az alkalmazás a fogyatékkal élők számára is elérhetővé válik. A mozgáskorlátozottak ugyanúgy használnak a mindennapjainkban számítógépeket, mint az átlagemberek, illetve az érzékszervi fogyatékkal rendelkezőknek is hozzáférhető, köszönhetően a számítógépes kisegítő programoknak.

Viszont nem szabad elfeledkezni az okostelefonok térhódításáról sem. Ezzel még kényelmesebb hozzáférést lehet biztosítani, és ugyanúgy rendelkeznek kommunikációs képességeikben hátrányos helyzetűeknek szánt kisegítőkkal.

A problémára adott megoldás során a döntéseim egy részénél fontos szerepet fog játszani, hogy ezek a számítástechnikai eszközök milyen lehetőségekkel, illetve korlátokkal rendelkeznek.

3.2. Célkitűzés

Mivel a fentiekben az irodalomkutatás részeként, elsőként tájékozódtam a meglévő alkalmazásokról, és vizsgálataim során mind az előnyös, mind a hátrányos tulajdonságaikat vizsgáltam, ez alapján levonhatom a következtetéseket, hogy a rendszeremnek milyen képességekkel érdemes rendelkeznie.

Ezért választottam a Bluetooth kommunikációt, mivel ezzel a legtöbb telefon rendelkezik, illetve a hordozható számítógépes eszközökben is megtalálható.

Mivel a rendszerem a SmartHome rendszerekbe kívánom integrálhatóvá tenni, így internethez való csatlakozási képességgel nem látom el, hiszen e rendszerek része, hogy a központi modulon keresztül a csatlakozás lehetséges.

Emellett az alkalmazás USB kapcsolaton keresztül is elérhető. Ezt is fontosnak tartom, mivel már a telefonok is rendelkeznek USB porttal, ez újabb csatlakozási lehetőséget jelent a rendszernek. Emellett USB kapcsolaton tervezem a firmware frissítést végezni. Ekkor bootloaderesnek mondható a rendszer, ami azért előnyös, mert így a felhasználó otthon elvégezheti a frissítések telepítését.

Mindezek összefogására és a rendszer vezérlésére egy mikrokontroller szolgál, ami pedig, a fentiekből következően, PC-ről vagy mobil eszközről lesz elérhető.

3.3. Beágyazott rendszer

A célfeladatot végrehajtó számítógépet tartalmazó rendszert divatos szóval beágyazott rendszernek nevezik (embedded system). A beágyazott rendszerek jellegzetessége, a környezetükkel való interakció is, amely felhasználói felületek, kommunikációs csatornák, érzékelők stb. segítségével valósul meg.

Ezen készülékekben rejlő mikroszámítógépet először mikroprocesszor segítségével valósították meg, melyet körbeépítettek az adott feladathoz szükséges perifériákkal, illetve memóriával.

Később egyetlen nyomtatott áramkörön tartalmzták a rendszerek a mikroprocesszor működéséhez szükséges mikroprocesszort, memóriát, járulékos logikát, a tipikusan szükséges perifériákat, így univerzálisan használható egységgé vált.

A technológia fejlődésével lehetővé vált, hogy egy teljes egykártyás mikrogépet egyetlen chipben valósítsanak meg. Ezt a mikroprocesszor magot, memóriát és perifériákat egyetlen chipben tartalmazó eszközt nevezzük mikrokontrollernek (röviden uC). A mai beágyazott rendszerek tekintélyes részének mikroszámítógépét ilyen mikrokontrollerekkel valósítják meg.

Tehát a beágyazott rendszer egy speciális célú számítógép, melyet egy konkrét feladat ellátására terveztek. Az általános célú számítógépekkel szemben előre meghatározott feladatot lát el, így a tervezés során megfelelően tudják optimalizálni a rendszert.^[1]

Ez alapján a rövid ismertető alapján elmondhatjuk, hogy a rendszer egy beágyazott rendszer. Amire az őt vezérlő firmware-t is én készítem.

3.4. Firmware

A firmware alatt azokat a rögzített, többnyire kisméretű programokat és/vagy adatstruktúrákat kell érteni, melyek különböző elektronikai eszközök

vezérlését végzik. A firmware a hardver működtetéséhez szükséges legalapvetőbb és bonyolultabb feladatokat látja el.

Nincs éles határ a firmware és a szoftver között. Mindenesetre a firmware kifejezést a hardverkomponens cseréje nélkül megváltoztatható, magasabb szintű szoftverrel szembeállítva lehet definiálni. A firmware feladata jellemzően az olyan, igen alacsony szintű műveletek elvégzése, melyek nélkül az eszköz teljesen működésképtelen lenne.

A frissítésre hibajavítás vagy funkcióbővítés miatt lehet szükség. A cél az lenne, hogy ez a végfelhasználó által is elvégezhető legyen, mivel a mikrokontrollerhez egy külön programozó kell.

3.5. Programozó készülék

Párhuzamos programozó segítségével lehet beégetni a mikrokontrolleres készülékek programját a chip-be, Flash memóriába vagy EPROM-ba. A programozók egy része univerzális, különféle mikrokontrollerek, FLASH-ek, EPROM-ok, EEPROM-ok, PLD-k programozhatók velük. Más részük egy gyártó eszközeit képesek kezelni.

A programozók többnyire soros vonalon csatlakoznak egy host számítógéphez, ahonnan az adatokat, illetve parancsokat kapják. A párhuzamos programozás során az adatot és többnyire a címet is párhuzamosan kapja meg a programozandó eszköz, a port lábain keresztül. A programozási parancsok többnyire szintén valamely lábakra adott jel vagy kombináció segítségével adhatók meg.

A programozáshoz sok esetben a normál tápfeszültségen kívül egy annál nagyobb feszültség is szükséges. A párhuzamos programozás algoritmusát és időzítését a chipék részletes adatlapja tartalmazza.

Bár a programozók nem drágák, de a felhasználót csak felesleges költségbe vernénk, illetve akkor szükséges lenne a kezelésének ismerete is. Ezért előnyös, hogyha bootloderes a rendszer.



11. ábra. Programozó

3.6. Kommunikáció

Mivel az előzőekben a hardverhez tartozó alapvető fogalmakat bemutatásra kerültek, ezt követően a kommunikációs protokollokra is érdemes kitérni, melyeket már biztosan használatra kerülnek a kivitelezés közben.

3.6.1. Bluetooth^[2]

Rövid hatótávolságú, adatszeréhez használt rádiós kapcsolat. A Bluetooth technológia egy full-duplex, frekvenciaugráltatásos átvitelt használ. Itt, Európában a 2,402 GHz és 2,480 GHz közötti sávban 79 vivőfrekvenciát használ, 1 MHz-es csatornaosztással.

A Bluetooth eszközök ezen a 79 rádiócsatornán kommunikálnak, másodpercenként 1600-szor váltva egyikről a másikra, ráadásul véletlenszerűen. Amikor kettő vagy több eszközt egymáshoz kapcsolunk, akkor azok szinkronban fognak csatornát váltani. Ha két különböző „párbeszéd” egy időben ugyanarra a rádiócsatornára kerül, olyan rövid időre alakul ki interferencia, hogy az nem okoz problémát.

Amikor két, Bluetooth funkciót használó készülék hatósugáron belülre kerül, párbeszéd indul el közöttük. Ha szükség van adatátvitelre, akkor kapcsolódnak egymáshoz. Telefonnal való kommunikációnál a fogadó félnek jóvá kell hagyni az adatátvitelt, ráadásul ehhez jelszó megadására is szükség lehet.

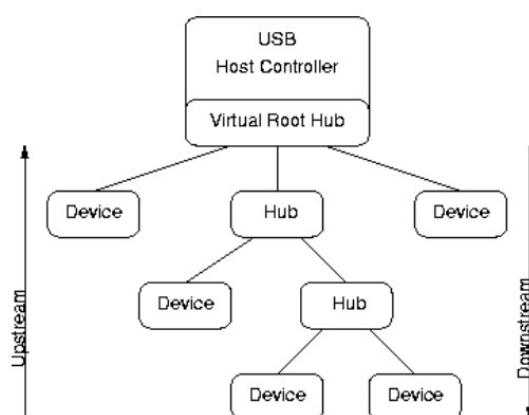
Fontos, hogy két módban lehet használni, single point és multipoint. A single point elnevezés sugallja, hogy egy vevő és egy adó csatlakozik, így használhatjuk hívásokhoz és zenehallgatáshoz is a headsetet. A multipoint módban a készülék több eszközhöz csatlakozhat, de ezekből egyszerre két eszköz aktív.

3.6.2. USB^[3]

Az USB, vagyis az univerzális soros busz egy erősen szabványosított kommunikációs technológia. Az Intel, a Compaq, a Hp, a Microsoft és a NEC cég is közreműködött az USB szabvány megalkotásában. Előnye, hogy teljes körűen Plug and Play, az összes modern operációs rendszer támogatja, és azonos felépítésű, akár PC, akár Mac számítógép része.

Az USB tulajdonsága, hogy osztható, ezt úgy nevezett USB hub-ok hajtják végre. Minden USB bővítőkártyán van egy integrált úgynevezett root hub (gyökerhub). Eerre csatlakoztathatunk USB eszközöket, vagy akár külső hubot.

Jelenleg többféle USB szabványt különböztetünk meg: USB 1.0, USB 1.1, USB 2.0 és USB 3.0, melyek sebességükben és más tulajdonságaikban is eltérnek, viszont visszafelé mindig kompatibilisek.



12. ábra. USB felépítése

4. Mikrokontrolleres rendszer tervezése funkcionális szemlélettel

A tervezés során Benesóczky Zoltán: Mikrokontrollerek alkalmazástechnikája jegyzetét követtem.^[1]

4.1. Specifikáció

Egy rendszer tervezését mindig annak specifikálásával kell kezdeni. Az előzetes specifikáció során megfogalmazásra kerül a megtervezendő feladat. Ez már az előzőekben részben megtörtént.

A feladat egy vezérlő tervezése és elkészítése, mely több elektromos készülék ki/bekapcsolását tudja vezérelni. A vezérlőhöz mobileszközön lehet csatlakozni Bluetooth segítségével, és vagy mobileszközről vagy akár számítógépen keresztül lehet a készülékeket kapcsolni. Az elektromos készülékek ki/bekapcsolásának időzítését is be lehet állítani. Illetve külön hardver nélkül is lehetséges lesz a firmware frissítés megoldása egy bootloader segítségével. A firmware frissítés számítógépen keresztül történik.

Ezután jöhet a részletes specifikáció, melynek során értelmezni kell, kiegészíteni és tovább finomítani az előzetes specifikációt, hogy az alapján már kiderüljenek a tervezéshez szükséges legfontosabb információk. Ekkor már meghozhatók olyan döntések, melyeken a továbbiakban biztosan nem fognak változni.

A kezelőszerv 2 nyomógomb, melyek funkciói:

Reset: A mikrokontroller alaphelyzetbe állítása.

Bootloader: Az USB kapcsolaton keresztül történő firmware frissítés indításhoz.

Az első bekapcsoláskor a rendszer a default beállítások szerint működik. A mikrokontroller működési állapotának visszajelzésére szolgál egy működés indikátor LED.

Két vezérlési mód van. Az egyik a relék ki és bekapcsolt állapotának változtatása, a másik pedig az időzítés beállítása. Természetesen egy elektromos eszköz be- és kikapcsolása időzítés beállítása nélkül is megtörténhet. Az időzítés lehet dátumhoz kötött, tehát egyszeri, vagy rendszeres például heti. Az időzítésért egy real time clock felel.

Az elektromos készülékek be és kikapcsolásáról a relék gondoskodnak, melyek vagy engedélyezik, vagy tiltják a tápellátást. Mindegyik reléhez tartozik egy LED, mely az aktív, illetve passzív állapot jelzésére szolgál.

A mikrokontrollerhez kapcsolódik egy tápellátást figyelő modul, mely detektálja a tápkiesést. Ennek jelzésére megkezdődik a biztonsági mentés a külső EEPROM-ba. Áram kiesésekor kritikus a relék állapotának elmentése, ez kerül az EEPROM-ba, mivel az időzítéseket már beállításukkor el lehet menteni.

A vezérlőhöz kapcsolódó Bluetooth modul inicializálása firmware-rel történik. A Bluetooth modulon található két darab LED, melyek a kapcsolat állapotát jelzik. Ezekről a tervezés során külön nem kell gondoskodni. Ha a zöld LED villog, akkor nincs kapcsolat. Ha folyamatosan világít, akkor detektálta az eszközt és létrejött a csatlakozás. Végül pedig, ha a piros villog, akkor adattranszfer van folyamatban. Ezzel állítható be a kívánt időzítés, és kapcsolgatható az elektromos készülékek.

Mivel az alkalmazás a SmartHome rendszerébe illeszthetőnek tervezett, ezért internet kapcsolatáról nem szükséges gondoskodni, mivel a SmartHome rendszer központi modulja biztosít interneten keresztül távoli elérést. Így a rendszer elérhető, hiszen a központi modul, Bluetoothon keresztül, továbbítja a rendszer számára a szükséges információkat.

4.2. Funkcionális blokkvázlat

A részletes specifikáció alapján elkészíthető a készülék funkcionális blokkvázlata. A feladat értelmezése alapján elképzelt megoldás megértését segítő, azt leíró blokkvázlat készül el, ahol a feladatot megvalósító funkcionális egységek szerepelnek, függetlenül attól, hogy hardver vagy szoftver valósítja meg őket. A blokkvázlat a blokkok adatkapcsolatát is mutatja, ezért szövegesen is célszerű elmagyarázni az elképzelt megoldást.

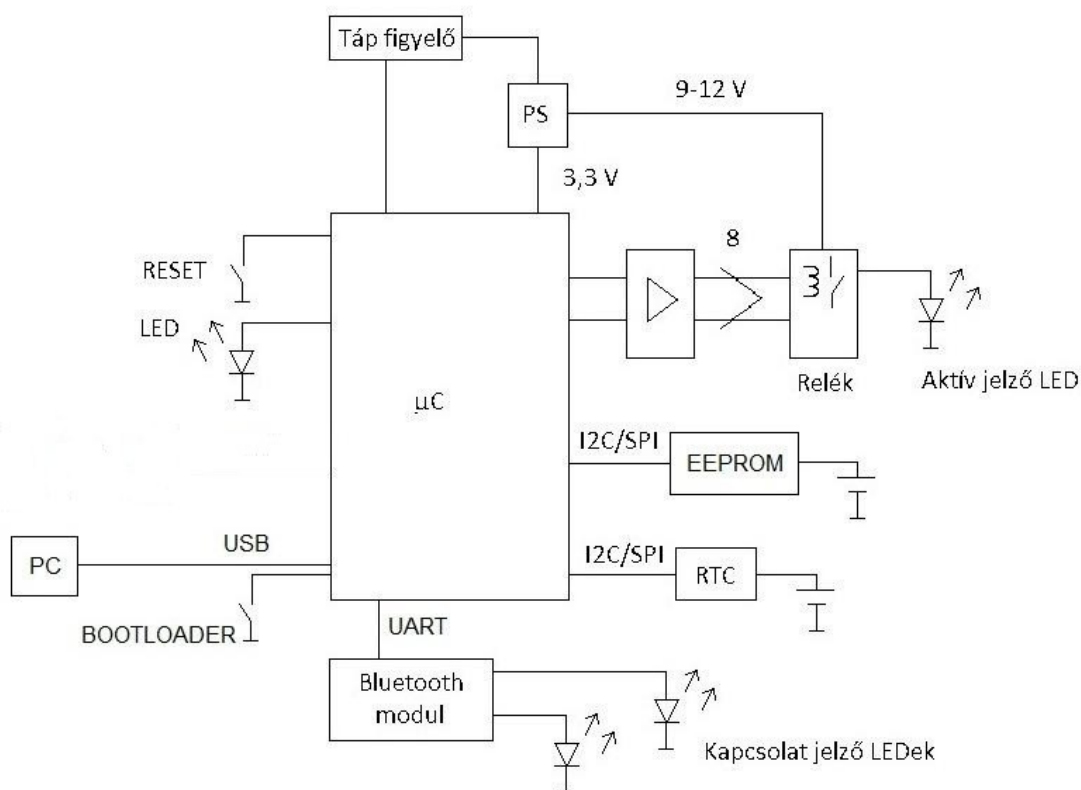
A RESET gomb önálló funkciója a mikrokontroller alaphelyzetbe állítása. Emellett található egy állapot indikátor is, mely jelzi, hogy a mikrokontroller működésben van-e.

A relék felelősek az elektromos készülékek áramtalanításáért, állapotukat külön-külön LED-ek jelzik, így nincs szükség a rendszerhez való csatlakozásra, hogy lekérésre kerüljön a jelenlegi állapotuk. Ha a relé behúzott állapotban van, akkor a LED világít.

A power supply valósítja meg a tápellátást, az esetleges áramszünet miatt egy tápfigyelő áramkörrel kiegészül, hogy időben el lehessen kezdeni a biztonsági mentéseket. Mivel biztonsági szempontból fontos megjegyezni, hogy mely eszközök voltak ki- és bekapcsolva, majd ezekkel az adatokkal újraindítani a rendszert. Ez a mentés a külső EEPROM-ba történik.

A bootloADER-es megvalósításról már volt szó, de annyit hozzá kell fűzni, hogy a kommunikáció megkezdésében egy nyomógomb lesz a segítség. Fontos kiemelni, hogy a firmware frissítés akár futás közben is történhet, a RESET és a BOOTLOADER gomb együttes használatával.

A Bluetooth és real time működésének külön bemutatására nincs szükség, mivel a korábbiakban már tárgyalva voltak.



13. ábra. Funkcionális blokkvázlat

4.2.1. UART

A Bluetooth modul UART kapcsolaton keresztül kommunikál a mikrokontrollerrel. Az UART, univerzális aszinkron adóvevő, egy olyan hardver, amely fordítást végez a soros és párhuzamos interfészek között. Soros átvitelnél az UART a bájtokat aszinkron start-stop bitfolyammá alakítja át, amiket elektromos impulzusokkal továbbít.

Az univerzális jelző azt jelenti, hogy az adatformátum és az átviteli sebesség nem kötött, hanem több, szabványos érték közül kiválasztható. Leggyakrabban integrált áramkörökben lehet vele találkozni, ahol szükséges a számítógéppel vagy más, soros porttal rendelkező eszközzel való kommunikálás.

4.2.2. I2C

A real time clock gondoskodik az időzítésekről, ehhez egy külső kvarc kristály is kapcsolódik. Mint már említésre került mind az EEPROM, mind a real time clock az I2C kommunikációs szabványt használja. Az I2C buszt a Philips cég fejlesztette ki, rövid távolságú elsősorban egy készüléken belüli integrált áramkörök közti kommunikációra. A kommunikáció két jelvezeték igényel: kétirányú adat és órajel. Rendelkezik a buszok jellemzőivel: sok egység kapcsolható a buszra, illetve az egységek lehetnek mesterek, stb.

4.3. Hardver-szoftver szétválasztás

A funkcionális blokkvázlat alapján el kell dönteni, hogy annak mely részeit valósítsa meg hardver és melyeket szoftver. Egyes hardver egységek adottak (pl. mikrokontroller, nyomógombok).

Gazdasági okok miatt általában célszerű a lehető legkevesebb, illetve legolcsóbb hardvert tartalmazó megoldást választani, amellyel a specifikációnak megfelelő, megbízhatóan működő megoldás adható. A hardver ára minden egyes készülékben egy (közel) állandó összeg. A szoftver ára általában eloszlik a készülékek között, azaz mennél több készüléket gyártanak, annál kisebb az egy készülékre eső összeg. Persze vannak kivételek is.

A blokkvázlat elég szemléletesen bemutatja milyen hardver egységekre lesz szükség, így viszonylag triviális a döntés. Konkrét értékű alkatrészeket itt még nem kell választani, a méretezés a tervezés későbbi lépései közé tartozik.

Nyolc darab relére mindenképp szükség van az elektromos készülékek kapcsolásához. Viszont olyan relét érdemes választani, amely két kivezetéssel rendelkezik, mivel életvédelmi szempontok miatt, a rendszer kétpólusú kapcsolónak tervezett. A relék elé egy erősítő mező kerül. Ez csatlakozik a mikrokontrollerhez. Ez alapján elég egyértelmű, hogy a relék meghajtásához mindenképp szükség van nyolc port bitre.

Azért volt célszerű nyolc elektromos készülék vezérlését választani, mivel állapotuk az EEPROM-ba tárolódik, és ez a kevés adat, így pont elfér egy byte-on. Mivel az EEPROM-ba történik a biztonsági mentés, így előnyösebb lenne egy olyan mikrokontroller választása, amely két I2C kommunikációra alkalmas porttal rendelkezik. Hiszen ha a kontroller épp a real time clock-val kommunikál, akkor egy esetleges áramszünet esetén, a kapcsolatbontása, és az EEPROM-mal a kommunikáció kialakítása, majd a biztonsági mentés megkezdése, túl hosszadalmas folyamat lehet. Külső EEPROM esetén három port bit kell a kezeléséhez.

Mivel az EEPROM-ot és a real time clock-ot a feljebb említett okok miatt nem fűzzük egy buszra, ezért a real time clock-nak is szüksége van két port bitre. Illetve mivel több fajta időzítési lehetőséggel kerül kialakításra a rendszer, ezért olyan real time clock-ot célszerű választani, ami két megszakítással rendelkezik. Tehát így a real time clock-nak legfeljebb 4 port bitre van szüksége. A real time clock-hoz tartozó külső kristálynak nincs szüksége port bitre a mikrokontrolleren.

A Bluetooth modul két port bitet igényel, és a bootloader megvalósításához, ami USB-n történik, is két port bit kell. Ezen kívül már csak a két nyomógomb van, ezeknek is szüksége lenne két port bitre. De ezekből a RESET ráköthető a mikrokontroller megfelelő CLEAR lábára, így I/O portot nem igényel.

Mivel további külső hardver nincs, összeszámolhatók a port bitek.

Relé vezérlés:	8 db
EEPROM:	3 db
Real time clock:	4 db
Bluetooth modul:	2 db
USB:	2 db
Nyomógomb:	1 db

Azaz összesen legfeljebb 20 port bit kell. Az összes többi egységet triviálisan szoftver valósítja meg.

A fentiek alapján kiválasztható a feladathoz megfelelő mikrokontroller. Maximálisan 20 port bitre van szükség. Az EEPROM-nak és a real time clock-nak szüksége van 2 MSSP serial portra, illetve, hogy a mikrokontroller képes legyen I2C kommunikációra. A Bluetooth modul miatt egy UART csatlakozásra, a bootloader megvalósításához pedig egy USB csatlakozásra is szükség van. A C-ben írt programhoz, természetesen minimális RAM is szükséges.

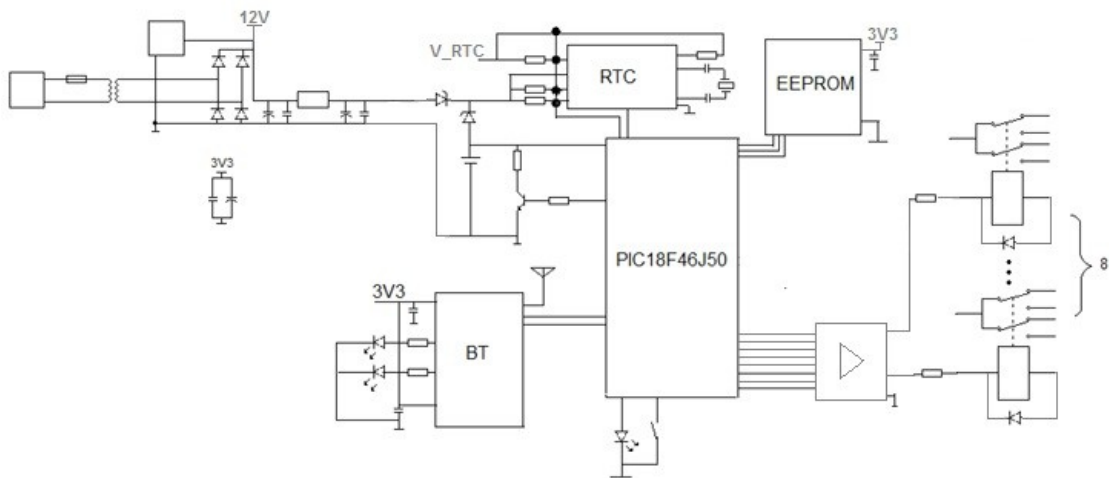
A feltételeknek eleget tesz egy PIC18F46J50 mikrokontroller (32 I/O; 64K memória; 2 UART; 2 MSSP; 1 USB), melynek ára is megfelelően alacsony. Természetesen a kiválasztásnál egyéb szempontok is fontosak (pl. rendelkezésre álló fejlesztői környezet, beszerezhetőség stb.).

4.4. Hardver rendszerterv

Ez már majdnem kapcsolási rajz, csak a lábszámokat és a még nem ismert alkatrész típusokat, illetve értékeket nem tüntetjük fel.

Ezután meg kell határozni a hardver blokkvázlaton szereplő alkatrészek típusát, illetve értékét. Meg kell határozni a készülék fogyasztását és meg kell tervezni a tápellátást.

Végül el kell készíteni a kapcsolási rajzot, az alkatrész listát, a nyomtatott áramkört (későbbiekben NYÁK) tervet. Nagyobb cégeknél a NYÁK tervezés nem a mérnök dolga, de figyelemmel kell kísérnie a kritikus részeket (nagysebességű jelek, analóg részek stb.).



14. ábra. Hardver rendszerterv

4.5 Megvalósítás

4.5.1. Alkatrész kiválasztás

A fentiekben már kiválasztásra került a használandó mikrokontroller, ez a PIC18F-s család 46J50-s tagja volt.



15. ábra. PIC18F46J50

Az alkatrészek kiválasztásánál fontos szempont volt az is, hogy ne kelljen sok helyről beszerezni őket, illetve olyan márkák a választandók, amik termékei megbízhatónak számítanak a piacon.

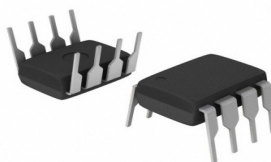
A Bluetooth modul egy BTM-222-s elektronikai alkatrész lett. Ez egyes osztályba tartozik, ami azt jelenti, hogy a hatótávolsága 100 méter, de jó minőségű antennával ez akár több is lehet. UART kommunikációra képes, a kommunikáció protokollja SPP.

Hátránya, hogy felülszerelt (SMD), ami a forrasztást kicsit nehezkesé teszi, de ezek a modulok nem kaphatók furatszerelt (DIP) kivitelezésben. De ha lennének, akkor is a megbízhatóság és a könnyű illeszthetőség részesítendő előnyben, mivel erről a modulról nagyon sok pozitív visszajelzés található. Ez egyben azt is jelenti, hogy a rendszer 3,3 V-s lesz.



16. ábra. Bluetooth modul

Real time clock-nak a Dallas gyártó egy gyártmánya tűnt megfelelőnek, mivel rendelkezik két megszakítással, képes I2C kommunikációra, megfelelőek a működési paraméterei és furatszerelt tokozású. Belső kristállyal nem rendelkezik, így egy külső kristály hozzákapcsolása mindenképp szükségessé vált.



17. ábra. RTC modul

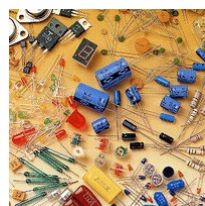
EEPROM-ból a Microchip, aki a mikrokontroller gyártója is, 24LC16B típusú terméke tűnt megfelelőnek. Képes I2C kommunikációra, 16 Kb-s, és

Schmitt triggeresek a bemenetei, ami előnyös a zaj elnyomás szempontjából. A működési elvárásokba is beleillik, valamint könnyen forrasztható.



18. ábra.
EEPROM

A relékhez egy RM48P-12V típusú relé került kiválasztásra. A legfontosabb szempont az volt, hogy két kivezetése legyen, hogy mindkét pólus megszakítható legyen vele, de az egyéb paraméterei is megfelelők a tervezett rendszer szempontjából.



20. ábra. Egyéb
alkatrészek

Az ellenállások, kondenzátorok, diódák és egyéb kisebb elektronikai alkatrészek kiválasztását a méretezés határozza meg.

4.5.2. SMD és DIP

Szükségesnek tűnik rávilágítani, hogy mi a különbség az SMD és a DIP között. Ha például a Bluetooth modul és a real time clock képét megvizsgáljuk szembetűnő a különbség a kettő között.

Az SMD a Surface Mounted Devices rövidítése, ez azt jelenti, hogy az alkatrész felületszerelt. Ez olyan áramkör, amiben az egységeket közvetlenül a felületre szerelik, nem lábakon állnak. Különböző tokozásokat, szám szerint hatot, lehet megkülönböztetni.

A DIP a Dual In-line Package rövidítése, ez azt jelenti, hogy az alkatrész furatszerelt. Talán ez az a tokozás, amely a legismertebb, hiszen ez az egyik

legrégibb technológia. Közvetlenül furatba forrasztva vagy IC foglalatban használják. Mivel igen távol lévő lábai miatt nagyon könnyű forrasztani, ezért eléggé közkedvelt. Ennek is többfajtája megkülönböztethető.

4.5.3. Méretezés

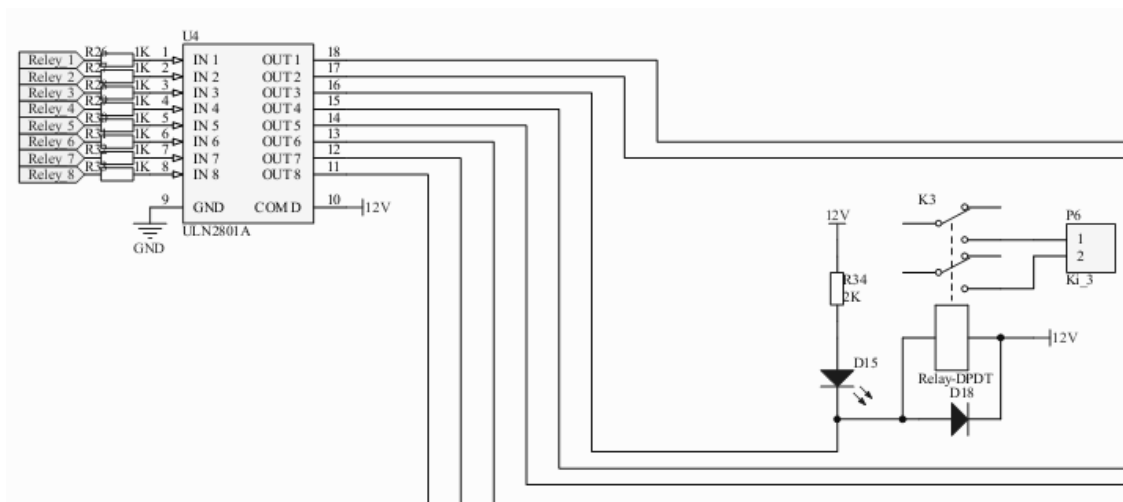
A méretezés kérdését jelentősen megkönnyítő, hogy a legtöbb modulhoz és fontosabb alkatrészhez az adatlapja nyújt ajánlott kapcsolást.

A méretezést egyszerűbb úgy végezni, hogy a rendszer részeire bontottá válik, és mindig csak egy adott részen van a fókusz, ügyelve az összképre. A projekt során az Altium Designer professzionális áramkör tervező szoftvert használtam.

A tápellátás kapcsolásán egy csavaros sorkapocs található, ahova beköthető a vezeték, összekötve egy olvadó biztosítóval, hogy megvédje az esetleges túláramtól az eszközt. Opcionálisan a második csavaros sorkapcon lesz egy DC tápcsatlakozó, ami biztonságosabbá teszi a fejlesztést. Ugyanis így dugasztápról üzemeltethető a berendezés, az érintés védelem miatt.

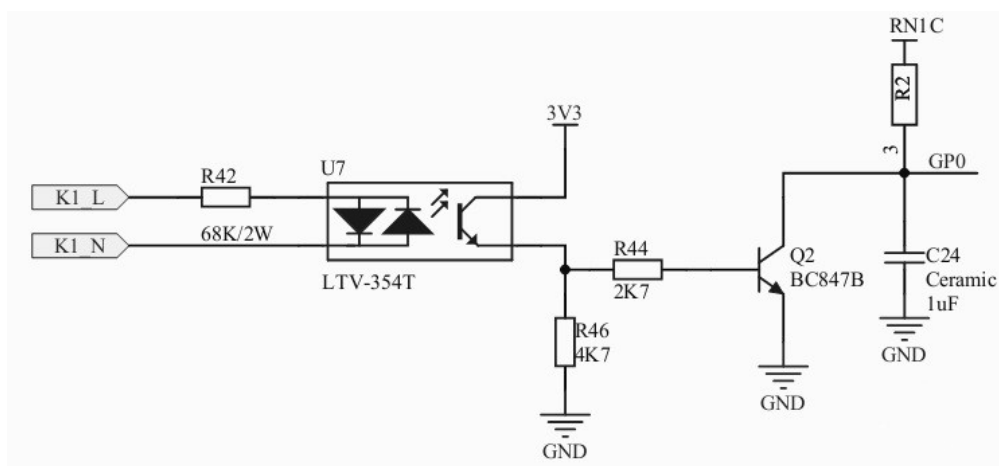
A két Schottky dióda feladat, hogy a real time clock számára a tápfeszültséget leválassza. Abban az esetben, ha használat közben áramszünet van, a real time clock-nak akkor is kell tápot kapnia, ekkor a felső Schottky dióda zárt állapotban van, viszont a másik kinyit, így a battery táplálja a real time clock áramköri IC-t. A tranzisztor segítségével előterhelés adható a battery-nek, ugyanis az elemeknek megvan az a tulajdonságuk, hogy terheletlen állapotban a kapocsfeszültségük magasabb, mint terhelve, a belső ellenállásuk miatt. Hiteles tápmérést akkor lehet biztosítani, ha a táp terhelve van. Erre szolgál a tranzisztor.

A mikrokontrolleren keresztül válik kinyitottá a tranzisztor, ekkor az ellenálláson keresztül megterhelt lesz a battery, és a terhelés alatt végrehajtható egy analóg-digitális átalakítás, a battery feszültsége beolvasásra kerül a mikrokontroller segítségével. Így megtudható az, hogy a batterynek milyen a töltöttsége.



22. ábra. Relé áramkör

Fontos figyelembe venni, hogy esetlegesen a relék tönkre mennek, ennek a detektálásra szolgál egy AC feszültség fogadására alkalmas opto csatoló.

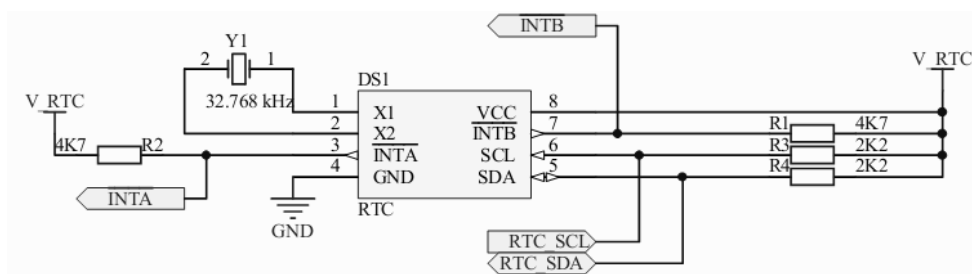


23. ábra. AC-opto

A kiválasztásra került valós idejű óra a rendszer működésének szempontjából elengedhetetlen, hogy a pontos időt számolni és nyilvántartani lehessen, hiszen a reléknek a megfelelő időben kell kapcsolniuk. A real time clock-ok többsége rendelkezik külső oszcillátor bemenettel. Erre a külső órajel bemenetre helyezhető el egy órakvarc, mely 32,768 kHz kvarc, viszonylag jó, 5-10 ppm-s pontossággal.

A real time clocknak két megszakítás kimenete van, melyek open drain jellegű kimenetek, ami annyit jelent, hogy aktívan a föld potenciál felé tudnak húzni. Ahhoz, hogy logikai magas szintet tudjanak adni, felhúzó ellenállásokat kell alkalmazni, melyek 4,7 k Ω -k. Ezek elegendően nagyok ahhoz, hogy ne terheljék túl az IC kimeneti portlábait, és elég kicsik ahhoz, hogy ne okozzon a szivárgó áram gondot.

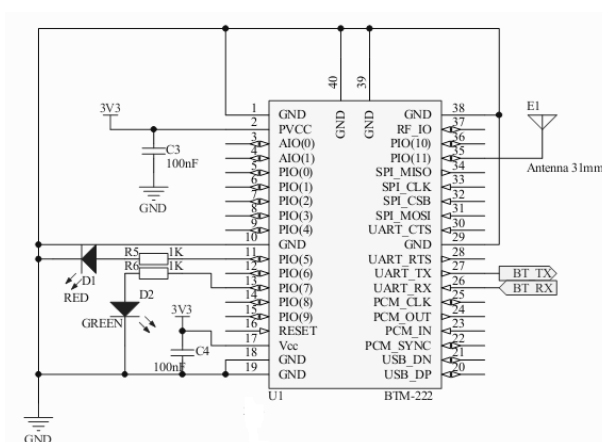
Az SDA, SCL lábon lévő két ellenállás is az adatlap alapján került kiválasztásra, mivel kommunikációs csatornák.



24.

ábra. Real time clock áramkör

A Bluetooth modul kapcsolása teljesen az adatlap ajánlása alapján lett tervezve. A LED-ek a korábbiakban már említett kommunikációs állapotjelzésreszolgálnak. Emellett szükség van a 100nF hidegítő kondenzátorra és az antennára.



25. ábra. Bluetooth áramkör

Egy 100 nF kondenzátor minden tápláláshoz kerül a tápintegritásért. Több szerepe is van, köztük a zajok és zavarok szűrése. Ököl szabály alapján került

kiválasztásra a mérete, komolyabb sebességű hálózatoknál már pontosan ki szokták számolni.

Szükséges kicsit visszatérni a táphoz és annak ellenőrzéséhez. Mivel lényeges, hogy eltároljuk a relék állapotát, így foglalkozni kell azzal, hogy mi van, ha áram kimaradás van. Erről a mikrovezérlőnek is értesülnie kell, és ennek hatására el kell mentenie az adott változókat, amire épp szükség van, egy nem felejtő memóriába. Úgy kell beméretezni a tápellátását az egész kártyának, hogy képes legyen azt az időt áthidalni, mikor elmegy a táp és kimentésre kerülnek az adatok.

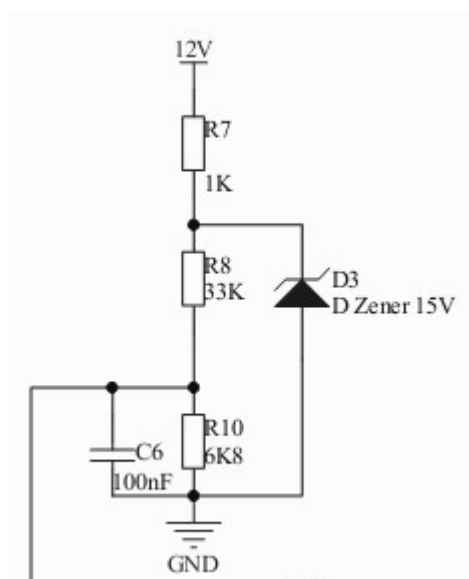
Ez a mentési idő rövid időt vesz igénybe, mivel kevés információ tárolódik. Egy bájtnyi EEPROM megírása az adatlap alapján 5 ms alatt biztonsággal végbe megy. Tehát legalább ennyi időt kell hagyni arra, hogy detektálja a rendszer a tápkiesést és mentsen.

Mivel a fő tápfeszültség 12V, így biztos, hogyha 10V-ra leesik ez a bemenő DC feszültség, akkor valami nincs rendben. Tehát a feladat a 10V feszültség alatti szint detektálása. Áramkör technikailag ez nyilvánvalóan egy komparátort igényel.

Mivel ez a 10-12V érték a mikrokontrollernek már túl magas lenne, ezért ezt le kell osztani egy feszültségosztó segítségével. A mikrovezérlőnek van saját beépített komparátora, ami programozható funkciókkal rendelkezik. Az adatlap tartalmazza, hogy a komparátor bemenetére adható legmagasabb bemeneti impedancia az 10 k Ω . Tehát úgy kell méretezni, hogy a feszültségosztóból származó ellenállás kisebb legyen, mint 10 k Ω . Rendelkezik a mikrokontroller belső precíziós referenciával, amit a komparátor egyik bemenetére lehet kapcsolni. Minimum értéke 0,825V, maximum értéke 2,475V adatlap alapján. Az osztás arányt ennek fényében meg lehet határozni.

Kiegészítendő ez az alapfunkció némi védelemmel is, mivel a villamos hálózatra tűrés tartományok vannak előírva és erre gondolni kell. Ugyanis a

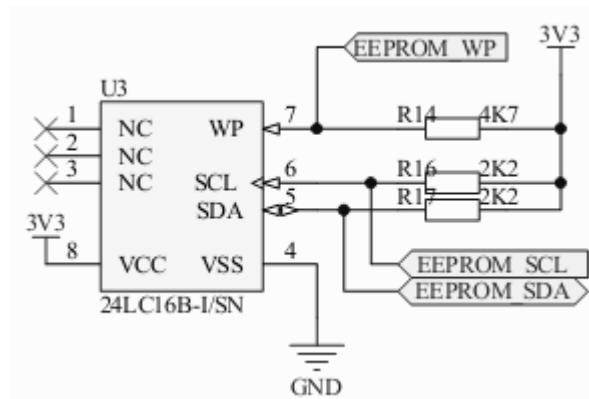
hálózati 230 V feszültség csak névleges és akár plusz 20%-kal is több lehet a konnektorba, így előfordulhatnak túlfeszültségek a hálózatba, ezért védelmet kell beépíteni. Mégpedig úgy, hogy egy Zener dióda a DC táp feszültségét megfogja. A Zener diódából 15 V-t választva azt jelenti, hogyha 15 V fölé emelkedik a DC táp feszültsége, ami már nem üzemszerű, meg fogja a bemenetet 15 V-n, így sem a kontroller, sem az egyéb alkatrészek nem fognak károsodni. A 100 nF kondenzátor pedig némi impedancia csökkentést és szűrést végez.



26. ábra. Tápkiesés detektálás

Az EEPROM is, a real time clock-hoz hasonlóan, I2C kommunikációt használ, és a tervezés eddigi lépései alatt eldöntött, hogy ez egy másik portra kerül. Az SDA, SCL lábakra ugyanaz vonatkozik, mint a real time clock-nál, 2,2 k Ω ellenállással felhúzásra kerül, mivel kommunikációs csatorna, amiből adódik, hogy gyorsnak kell lennie. Van egy write protect bemenete, ami alapvetően tápra van felhúzva, így írásvédetté válik, tehát amikor megjelenik a táp semmi féle zavar miatt nem sérülhet meg az EEPROM tartalma. Nem módosítja a belső celláit semmilyen körülmények között.

Ezt majd le kell húzni logikai egybe, ha írásra kerül sor, és ezután kezdődik meg az I2C kommunikáció.



27. ábra. EEPROM áramkör

4.6. Szoftver rendszerterv

A hardver tervezés befejezése után, folytatható a tervezés a mikorkontrolleren futó szoftver rendszerterv elkészítésével. Itt kell megtervezni és leírni a szoftver működését, eldönteni, hogy mi fut IT szinten, mi a főciklusba ágyazva. A leírás az egyes szoftver egységek specifikációját (mit csinál, melyek a be és kimenő paraméterei, milyen adatstruktúrán végez manipulációkat stb.), és algoritmusának leírását tartalmazza, továbbá azt, hogy az egyes eljárásokból hogyan épül fel az egész rendszer, az eljárások hogyan kapcsolódnak egymáshoz. Az egyes eljárások működésének megértését folyamatábrás leírás segítheti. Az implementáláshoz a MicroChip ezen mikrokontrollerhez szánt programját, az MPLAB-ot használom.

IT forrás alatt azok az egységek értendők, amik működésükkel interruptot, megszakítást válthatnak ki. Ebben a rendszerben több ilyen is előfordul. A Bluetooth modul, a real time clock és az EEPROM mind megszakítás forrása lehet. Valójában nem maguk a perifériák az interrupt kiváltók. Csak jelzik a mának, hogy fontos esemény történt, például a Bluetooth modul adatot küldött, ekkor pedig a mag hardveresen lekezeli az eseményt. Természetesen a megszakításokat lehet engedélyezni vagy tiltatni.

Ennek köszönhetően hatékonyabb firmware-t lehet készíteni, ugyanis nem szükség rendszeres időközönként az interruptot kiváltó egységekre ránézni, hogy

történt-e valami. Elég egyszerűen engedélyezni a megszakítást, így csak akkor kezdődik el a kiértékelés, ha megszakítás érkezett.

A főprogram parancsdekóder része a mobil eszköztől vagy számítógéptől érkező adatkódokat figyeli. Ha volt adatküldés, akkor a relék aktuális állapotától függően, azonnal reagál és a megfelelő állapotba váltja őket, ha szükséges. Az egyes állapotoknál meghívja az aktuális állapothoz tartozó rutint.

Ha volt állapotváltoztatás kérés, akkor a megadott reléhez tartozó lábát, attól függően változtatja, hogy az eszközt be- vagy kikapcsolni szeretnék. Ha bekapcsolni, akkor a megfelelő port lábát logikai magas szintre húzza, ami itt a 3,3 V-t jelenti, ha kikapcsolni, akkor pedig logikai alacsonyba. Itt egyéb esemény nem történhet.

Ha a küldött adatkódban az időpont beállítására is volt vonatkozó rész, akkor a mikrokontroller megkezdí a real time clock-kal való kommunikációt. Az I2C kommunikációnál van egy adat és egy órajel vonala. Aki adja az órajelet az a master, ami ebben a rendszerben a mikrokontroller. A mikrokontroller úgy tud vonalat nyitni, hogy az órajel logikai magas szintje alatt lehúzza a kezdeményező vonalat, ezzel elküldte a START bitet, a kommunikáció befejeztével pedig, ugyanúgy az órajel logikai magas szintje alatt magasba rakja. Minden más típusú bit küldésénél, mindig az órajel logikai alacsony szintjében történik a változás. Ennek megfelelően úgy néz ki az írási kommunikáció, hogy a START bit után egy 7 bites blokkot küld, ami a megfelelő slave, periféria kiválasztására szolgál, illetve egy read/write (logikai alacsony az írás) bitet, hogy tudja, hogy mi következik. A real time clock visszaigazolja a kapcsolatot, és ezután a mikrokontroller megmondhatja a byte-nak a helyét, majd az ezt követő visszaigazolás után, megindulhat az időpont átadása.

Ezen beírt adatok alapján vált ki később interruptot a real time clock a két erre szolgáló lábán, minek hatására a mikrokontroller beolvassa és kiértékeli az adatokat. Az egyik interrupt láb a rendszeres riasztásokra szolgál a másik pedig a

dátumhoz kötöttekhez. Ha egy időzítési megszakítás bekövetkezett, akkor a parancsdekóder a regiszterbe lévő 8 bitet kiértékeli, ezután pedig a megfelelő módon megváltoztatja a relék állapotát.

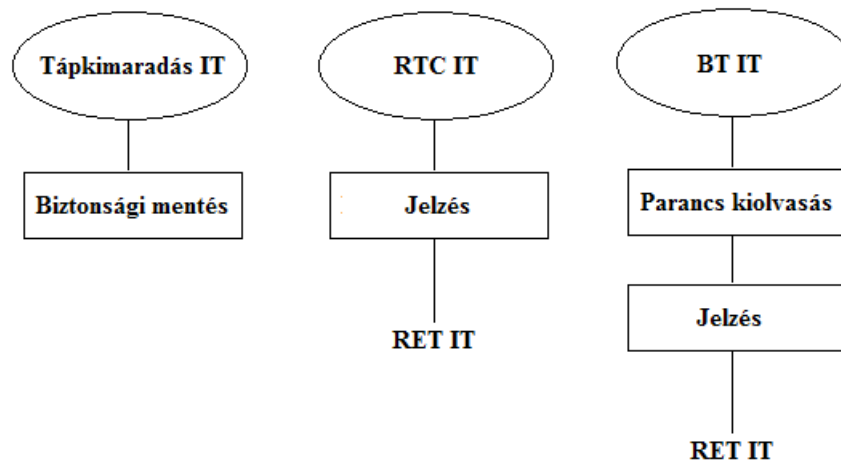
Beállított időzítéseket és a relék állapotát menteni kell és onnan vissza is kell tudni olvasni. Erre szolgál az EEPROM, ami szintén úgy idéz elő megszakítást, hogy elkezdi küldeni az adatait a mikrokontrollernek, ami a real time clock-nál már leírt módon történik.

Az adatok írásánál itt viszont, a 7+1 bites blokk, első 4 bitje címzi az eszközt, nem az első 7. A maradék 3 bit azt mondja meg az EEPROM, mely lapja írandó. Ezután itt is a byte címének kiválasztása történik, és már írhatók is az adatok.

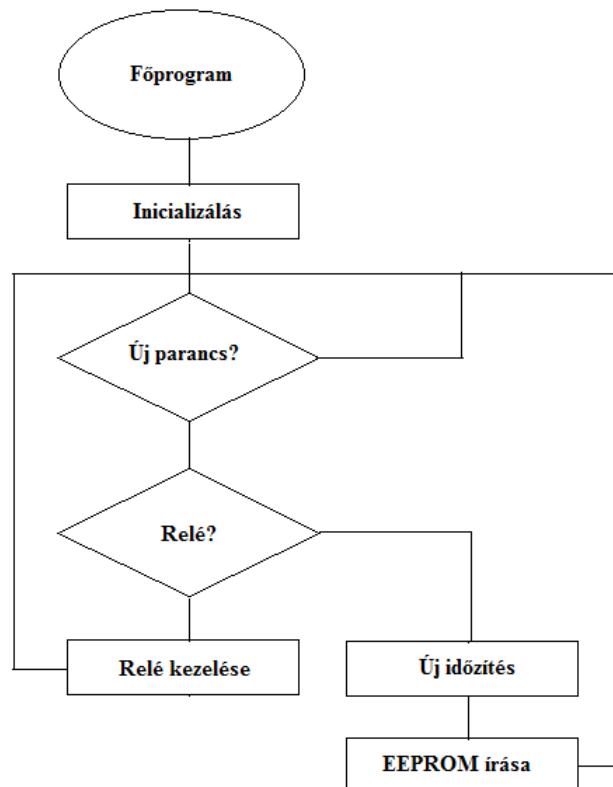
A Bluetooth modul a mobiltól kapott adatokat elkezdi a mikrokontrollernek küldeni bitenként, mivel az UART kommunikáció egy soros kommunikáció és egyszerre csak egy bit lehet a vonalon. Amikor a regiszter megtelik, tehát amikor a Bluetooth modul átküldött 8 bitet, akkor kiváltódik egy interrupt és a mikrokontroller elkezdi kiértékelni az adatokat. Ez alapján megkezdődik az relék állapotának változtatását és beállított időpont frissítését, ha kell.

A hardver tervezés alapján tudott, hogy a mikrokontroller komparálással detektálja a táp kimaradást. Ha a komparátor jelzi, hogy tápkimaradást érzékel, akkor megkezdődik az EEPROM-ba a biztonsági mentés, a fentebb már leírt, I2C szabványnak megfelelően.

Ezután megrajzolhatók a megfelelő folyamatábrák:



28. ábra. Interruptok kezelése



29. ábra. Főprogram

4.7. A rendszer tesztelése

A hardver tesztelése szimulátorral és gyakorlatban is megtörtént, minél több szélsőséges tesztet felállításával. A firmware teszteléséhez segítségre van a fejlesztő környezet, a mikrokontrolleres rendszeren futó firmware

helyességének tesztelése kissé nehézkes, mivel nem sok kezelőszerv található a felületén.

A tápellátás kiesésének produkálására több tesztet is felállításra került. Kipróbálásra került a különböző egységektől kapott adatok olvasása és írása közbeni táp elvétel, és a rendszer teljes mértékben az elvártaknak megfelelően működött.

Ezenkívül a rendszer stressz tesztnek is alá lett vetve, ami alatt a 24 órán keresztül, folyamatos, különböző időintervallumonkénti relé kapcsolgatás értendő.

A Bluetooth kapcsolaton való kommunikáció teszteléséhez, ugyan egy számítógép is megfelelő eszköz lehet, de mindenképp szükséges a mobil eszközön futó program elkészítése az éles teszteléshez.

5. Felhasználói szoftverek fejlesztése objektum orientált szemlélettel

Tervezés során az Objektum orientált szoftverfejlesztés című könyvet vettem alapul.^[4]

5.1 Bevezetés

A könnyen kezelhetőség a legfontosabb egy felhasználói szoftvernél. A legtöbben azt várják az általuk használt szoftvertől, hogy egyszerű, egy-két komponensből álló kezelő felülettel rendelkezzen, és egyértelmű legyen, hova mit kell írni, és melyik gomb mire való.

A felhasználók nem szeretnék sokat tudni a szoftver működéséről, de szeretnék biztonságos szoftvert, ugyanakkor gyorsat is. Bár hasznosnak tartják, hogy a kritikus műveletek esetén visszakérdez a szoftver, de a sok ellenőrző kérdés már terhes. Ekkor hajlamosak a többedik kérdésnél elsiklani a kiírt üzenet felett, és csak minél előbb el akarják érni, hogy a szoftver végrehajtsa, amire utasították.

Emellett szeretnék, ha a karbantartása a szoftvernek, illetve frissítése ne okozzon számukra „egy gombnyomásnál” több fejfájást. Illetve, ha a rendszer erőforrás felhasználása jól átgondolt lenne, és például nem merítené az akkumulátort, vagy foglalna sokat a memóriából.

5.2. Mobileszközre szánt szoftver tervezése

5.2.1. Informális specifikáció

A program célja egy hardver eszköz vezérlése. A felhasználó egy kezelő felület segítségével kapcsolgathatja be vagy ki elektromos készülékeit, illetve a felületen lehetősége van az időzítés beállítására. Az elektromos eszközök kapcsolásáért felelős relék nevükkel azonosíthatók. A felhasználónak lehetősége van rendszeres időzítések beállítására, illetve egyszeri időpont választására. Ettől függően más adatok megadása a szükséges.

A program indításkor automatikusan csatlakozik Bluetoothon keresztül a hardver eszközhöz. A program használat közben folyamatosan lekérdezi az eszközök állapotát, így a felhasználó számára csak az a lehetőség válik aktívvá, a be- és kikapcsolás közül, mely elérhető.

Ezen kívül a felhasználónak lehetősége van megjegyzések elhelyezésére is, hogyha például távoli időpontot állított be, vagy épp valamiért kiemelten fontosnak tartja az adott eszköz kapcsolását, akkor lehetőség szerint erről emlékeztetővel értesítse a program.

Ha a program futása közben bármiféle hiba történik, arról automatikusan és azonnal értesíti a felhasználót.

5.2.2. Fogalmi modell

A szöveg lényeges főneveit kigyűjtve hozzá lehet fogni a fogalmi modellben szereplő alapvető attribútumok, objektumok azonosításához. Ezt a lépést fel lehet használni arra, hogy a kifejezések rövid, ékezetmentes megfelelőt kapjanak.

program:	app
relé azonosító:	relay
időzítés:	timer
Bluetooth:	Bluetooth

állapot: status

megjegyzés: note

Ezután kigyűjthetők a specifikációból a tevékenységek, amelyeket tipikusan igék és igenevek fogalmaznak meg:

elektromos készülékek kapcsolgatása: status changing

időzítés beállítása: timering

Bluetooth csatlakozás: connecting

állapot lekérdezés: status requesting

üzenet létrehozása: message creation

adatküldés: data sending

megjegyzések készítése: make note

emlékeztetés: reminder

A feltérképezett objektumok azonos típusainak osztályokat lehet megfeleltetni, majd az osztályokhoz a tevékenységeket kapcsolni. Ezek alapján első közelítésben össze lehet foglalni a problémater objektumtípusait, objektumait, az objektumok megismert attribútumait és az objektumtípusokhoz rendelhető műveleteket.

Objektum	Objektumtípus	Attribútum	Művelet
app	App		
timer	Timer	note	timering, make note, reminder
Bluetooth	Bluetooth	relay	connecting, status requesting, data sending
message	Message	relay	message creation
status	Status	relay	status changing

Az objektumok közötti asszociációkra a specifikáció szerkezeteinek elemzése alapján lehet következtetni. Arra keressük a választ, hogy „mi, mivel, mit csinál.”

A program:

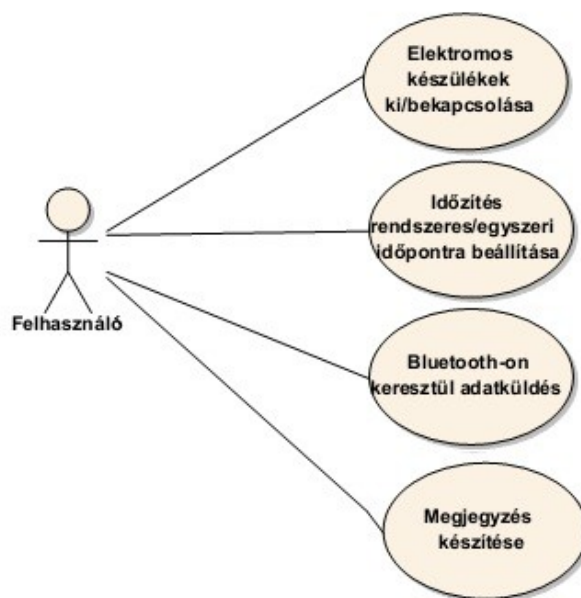
- emlékeztetőt küld,
- megteremti a Bluetooth kapcsolatot,
- szabványos üzenetet hoz létre,
- szabványos üzenetet küld Bluetoothon,
- folyamatosan lekérdezi az elektromos eszköz állapotát

Az öröklési viszonyok feltérképezésére a típusok közötti általánosítási és specializációs viszonyokat kell felismerni.

5.2.3. Használati esetek

A program egyetlen felhasználótípusnak készül. A lehetséges tranzakciók:

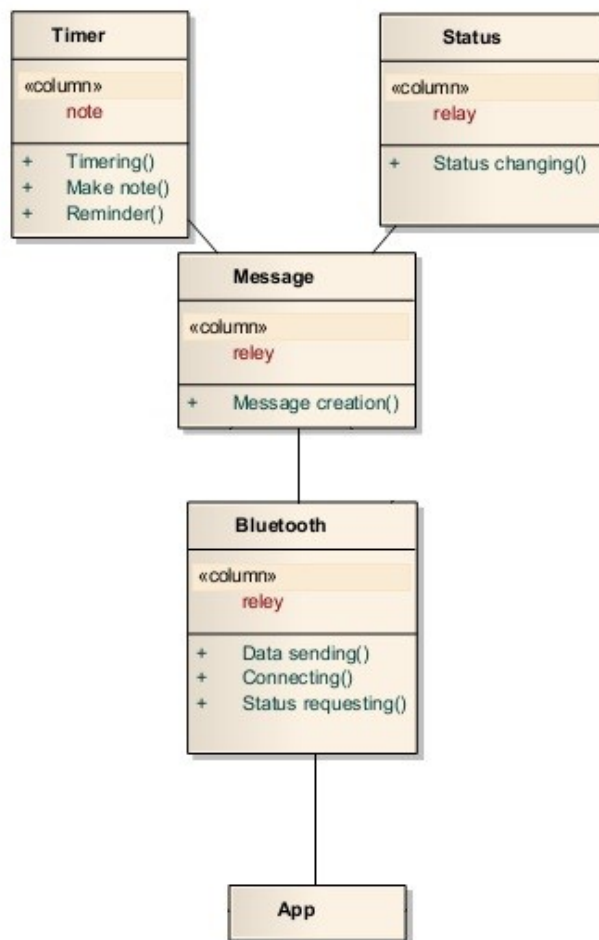
- elektromos készülék ki/bekapcsolása,
- időzítés rendszeres/egyszeri időpontra beállítása,
- Bluetooth-on keresztül adatküldés.



30. ábra. Használati eset modell

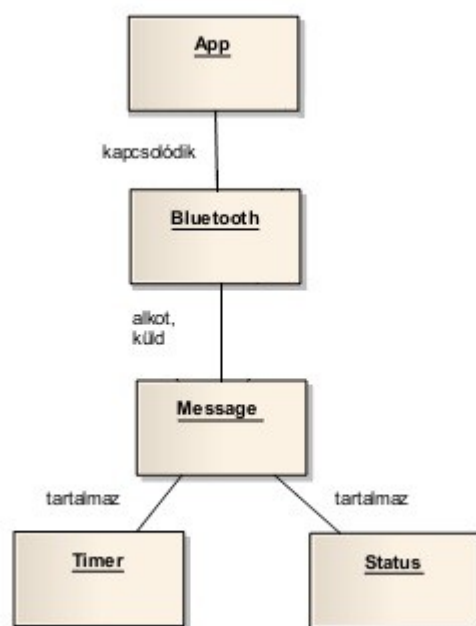
5.2.4. Az objektummodell

Az idáig megszerzett információk alapján felépíthető a feladat objektummodellje. Az osztálydiagram, amely a típusokat tartalmazza, az objektumokat és az objektumok közötti kapcsolatokat jeleníti meg. Valójában az objektumorientált analízis során ritkán kell egyedi objektumokban gondolkozni.



31. ábra. Osztály diagram

Az objektumdiagram, amely a példányok (objektumok) egy lehetséges rendezését mutatja be.



32. ábra. Objektum diagram

5.2.5. Dinamikus modell

A dinamikus modell felvétele során először a használati esetek forgatókönyvei kerülnek összeállításra, amelyek a rendszer működését, mint az egyes objektumok közötti párbeszédet fogják meg. Második lépésben ezen forgatókönyvek kiterjesztésre kerülnek kommunikációs modellekké, megmutatva, hogy a külső párbeszédhez hogyan csatlakoznak a rendszer belső objektumai.

Elektromos készülékek ki/bekapcsolása:

1. Status changing: A relék állapotát állítja át.

Időzítés beállítása:

2. Timering: A relék időponthoz kapcsolásának beállítása.

Bluetooth-on keresztül adatküldés:

3. Connecting: A Bluetooth kapcsolat létrehozása.
4. Status request: A relék állapotának lekérdezése.

5. Message creation: Az eszköznek küldendő szabványos üzenet létrehozása.
6. Data sending: Az elkészített szabványos üzenet elküldése Bluetoothon keresztül.

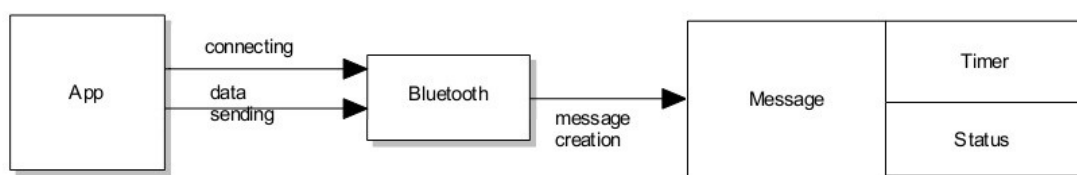
Megjegyzés készítés

7. Make note: Emlékeztető készítése.
8. Reminder: Az emlékeztető alapján riasztás.

A program indításkor automatikusan megpróbál Bluetoothon keresztül csatlakozni az elektromos készülék vezérlőhöz (connecting). Ha nem sikerül, értesíti a felhasználót, hogy a kapcsolódás nem sikerült. Ha sikerült Bluetoothon keresztül csatlakoznia, akkor elkezd lekérdezni az adatokat (status requesting). Ezt a program futása alatt folyamatosan ismétli, hogyha esetleg a program futása közben egy időzítés miatt valamelyik elektromos készülék kikapcsolna, erről azonnal értesüljön a felhasználó. Ezután továbbléphetünk és beállíthatjuk, hogy mely reléket kapcsoljuk ki vagy be, vagy hagyjuk meg az állapotában (status changing), illetve azt, hogy ezt időzítéshez kötjük-e. Ha időzítéshez kötjük, akkor választanunk kell, hogy egyszeri vagy rendszeres időzítést szeretnénk, és hogy a rendszeresség napi vagy heti ütemezésű (timering). Ezen az ágon tovább haladva el kell döntenünk, szeretnénk-e megjegyzést elhelyezni (make note). Tovább nem léphetünk, így visszalépve az időzítés mellett csak az a lehetőségünk van, hogy nem kötjük időzítéshez a kapcsolást, ekkor a relék kapcsolása azonnal megtörténik. A timer és a status objektumokból a message megalkot egy szabványos üzenetet, és message objektumot használva az aktív kommunikációs csatornán keresztül elküldésre kerül az üzenet. A reminder metódus felel azért, hogy megfelelő időben értesítse a felhasználót, az általa elhelyezett megjegyzésekről.

5.2.6. Eseményfolyam-diagram

A kommunikációs modell másik vetülete az úgy nevezett eseményfolyam-diagram, amely az objektumok osztályai között tünteti fel az ünetküldési irányokat.



33. ábra. Esemény-folyam diagram

Igaz ugyan, hogy ténylegesen objektumok kommunikálnak, azonban egy objektum csak akkor képes egy üzenet fogadására, ha az osztálya az üzenetnek megfelelő metódussal rendelkezik. Másrészt, egy objektumnak csak akkor küldhetünk üzenetet, ha a forrásobjektum adott metódusából látható. A láthatóságot leggyakrabban a beágyazott mutatók módszerével biztosítjuk. Ezek miatt, az eseményfolyam-diagram hasznos kiegészítőül szolgálhat az osztályok interfészének és beágyazott mutatóinak tervezése során.

5.2.7. Objektumtervezés

Az objektumtervezés első lépésében az osztályok definiálása történik meg elsősorban az objektum-modell alapján: App, Bluetooth, Message, Timer, Status. Az osztályok közötti öröklési lánc ugyan az objektum-modell alapján felállítható.

Az egyes attribútumok leképzése során megfontolás tárgyát képezi, hogy azok újabb objektumoknak tekintendők-e, amelyekhez ekkor a lehetséges műveletek feltérképezése után osztályokat kell készíteni, vagy pedig befejezve az objektum-orientált finomítást a beépített típusokat felhasználni. A két lehetőség közül aszerint kell választani, hogy az attribútumon milyen, a beépített típusokkal készen kapható műveleteket kell végezni, illetve ezen műveletek

bonyolultsága meghalad-e egy olyan szintet, amelyet már célszerű egyetlen metódusban koncentrálni. A felhasználható beépített típusokat az attribútum értékkészlete határozza meg.

5.2.8. Implementáció

Az analízis és tervezés után, következhet az implementáció. Ehhez az Android platformot választottam, de a későbbiekben szeretném a programot, minden jelentősebb platformon elérhetővé tenni. Az Android operációs rendszer külön előnye, hogy az újabb verziójú telefonjaikban az USB programozás is elérhetővé válik, így a bootloaderes firmware frissítés akár mobil eszközről is elérhető lesz. A fejlesztést az Eclipse fejlesztőkörnyezetben végeztem.

5.3. Szoftvertesztelés

A szoftver verziói folyamatosan tesztelés alá kerültek. A tesztelés lépésről lépésre zajlott. Először csak a Bluetooth kapcsolat létrehozása volt a cél, aztán olyan szabvány üzeneté, ami az eszközöket ki- és bekapcsolja. Egy eszköz, illetve több eszköz egyszerre történő vezérlése is a teszteljárásba tartozott. Eddig még nem került tesztelés alá az összes lehetséges kombináció, azok nagy száma miatt. De minden lehetőségre, tehát egy relé választása, két relé választása, egészen 8 relé választásáig, legalább 3 lehetséges eset tesztelésre került.

Tesztelés alatt áll, az időzítések megadása, illetve annak a vizsgálata, hogyha nem minden reléhez ad meg a felhasználó időzítést, akkor hogyan reagál a rendszer. Ennek a tesztelésnek a körébe tartozik a megadott emlékeztetők megfelelő funkcionálásának vizsgálata is.

6. Továbbfejlesztési lehetőségek

Továbbfejlesztési lehetőségként említhető a rendszer ZigBee alapúvá alakítása, mivel jelenleg helyhez kötöttek az elektromos berendezések, de mint már a rövid tartalmi összefoglalóban is említésre került a cél az elv bemutatása volt, amire ez a megoldás is teljesen alkalmasnak bizonyult.

Mivel a rendszerem a SmartHome szabványhoz illeszthetőnek terveztem, így internet eléréssel nem rendelkezik, de igény szerint akár ilyen irányba is fejleszthető a rendszer és újabb modulok hozzáadása is lehetséges.

A rendszer kiterjesztése teljes lakásautomatizálásra nagy kihívásokat rejtő feladat, hiszen akkor irányítástechnikai és szabályozó berendezéseket kell tervezni, ennél költség hatékonyabb megoldás, olyan eszközök vásárlása, melyek már rendelkeznek valamilyen kommunikációval. Ezek után már könnyen csatlakoztathatók az általam tervezett eszközökhöz.

7. Összegzés

Dolgozatom részeként irodalomkutatást végeztem. Áttekintettem és kipróbáltam az eddig megoldásokat, majd azokat összegezve következtetéseket vontam le. A tapasztalatok alapján kidolgoztam egy saját rendszertervet, melyben meghatároztam az rendszer és az alkalmazások felépítését, és a minimális rendszer követelményeket. A választott alkatrészeket bemutattam és ahol lehetett kitértem a biztonság kérdésére.

Mind a hardver, mind a szoftver fejlesztéshez a szakirodalmat áttanulmányoztam, és az egyes technológiáknak utánajártam és megismertem. A feladatom során törekedtem a minél általánosabb megoldások keresésére a további fejleszthetőség és kompatibilitás miatt. Ha problémába ütköztem, akkor próbáltam alternatívát keresni, és szükség szerint az eddig elkészült terveket is revideálni.

A végső konklúzióm, hogy a mindennapi életből vett, gyakorlatias problémára nyújtottam megoldást és sikerült azt a minőséget elérnem, amivel akár a mindennapi életben is használható.

8. Ábrajegyzék

Ábrajegyzék

1. ábra. SmartHome illusztráció.....	4
2. ábra. Távirányítós kapcsoló.....	6
3. ábra. Infravörös távkapcsoló.....	6
4. ábra. Visszaszámláló.....	7
5. ábra. Digitális időkapcsoló.....	8
6. ábra. Energiatakarékos kapcsoló.....	8
7. ábra. 1-9 óra közötti visszaszámláló.....	8
8. ábra. Napi időzítésre alkalmas.....	9
9. ábra. Digitális időkapcsoló - kétpólusú.....	9
10. ábra. Mechanikus időkapcsoló - kétpólusú.....	9
11. ábra. Programozó.....	14
12. ábra. USB felépítése.....	15
13. ábra. Funkcionális blokkvázlat.....	19
14. ábra. Hardver rendszerterv.....	23
15. ábra. PIC18F46J50.....	23
16. ábra. Bluetooth modul.....	24
17. ábra. RTC modul.....	24
18. ábra. EEPROM.....	25
19. ábra. Relé.....	25
20. ábra. Egyéb alkatrészek.....	25
21. ábra. Tápellátás és tápfelügyelet.....	27
22. ábra. Relé áramkör.....	28
23. ábra. AC-opto.....	28
24. ábra. Real time clock áramkör.....	29
25. ábra. Bluetooth áramkör.....	29
26. ábra. Tápkiesés detektálás.....	31
27. ábra. EEPROM áramkör.....	32
28. ábra. Interruptok kezelése.....	36
29. ábra. Főprogram.....	36
30. ábra. Használati eset modell.....	41

31. ábra. Osztály diagram.....	42
32. ábra. Objektum diagram.....	43
33. ábra. Esemény-folyam diagram.....	45

9. Irodalomjegyzék

1. Benesóczy Zoltán: „Mikrokontrollerek alkalmazástechnikája” (2009)
2. <http://hu.wikipedia.org/wiki/Bluetooth> (utolsó frissítés: 2012.március 21.)
3. <http://www.muszeroldal.hu/assistance/usb.pdf>
(utolsó frissítés: 2006.szeptember 11.)
4. Kondorosi Károly, Szirmay-Kalos László, László Zoltán: „Objektum orientált szoftverfejlesztés” (2007)