

Pásztor Ákos

myPhone project Dokumentáció



Tartalomjegyzék

Feladatkiírás	2
Specifikáció.....	2
Funkcionális leírás	4
Töltés.....	4
Bekapcsolási folyamat	4
Idle-mód.....	4
Tárcsázás	5
Bejövő hívás	5
Hívás közben	5
Kontaktok betöltése az eszközbe.....	5
Hardver.....	6
Alkatrészek, komponensek az alaplapon.....	6
Előlap komponentek.....	6
Tervezés, gyártás, forrasztás.....	6
Szoftver.....	8
Blokkvázlat	8
Inicializálás	9
LCD működése	9
Interrupt rutin	9
Main függvény	10
Felhasználói leírás	11
Bemérési tapasztalatok.....	11
Függelék	12
Képek az eszközről	13
Kapcsolási rajz	14
Blokkvázlat	15

Feladatkiírás

A projekt: egy saját magam által tervezett, összerakott, felprogramozott mobiltelefon, mely képes hívások fogadására, hívások indítására, valamint kontaktok tárolására. A kontaktok létrehozhatóak számítógépen, és letölthetőek a telefonba USB porton keresztül.

Maga a projekt két lépcsős. Ebben a verzióban a telefon áramellátása már akkumulátorokkal történik, valamint a NYÁK egy alumínium dobozba van építve, amely doboz előlapján fóliatasztatúrás billentyűzet, grafika és kivágások találhatóak. Továbbá ebbe a házba van beépítve a mikrofon és a hangszóró is.

Az 1.0-ás verzióról bővebb leírás: <http://akospasztor.com/projects/myphone-v1>

Specifikáció

A telefon vezérlője egy Atmel AtMega128 mikrokontroller, mely 12.288MHz-s frekvencián működik, amit egy külső kvarc-oszcillátor biztosít. A külső (hardveres) reset-ért egy MCP130 reset IC felelős. A programozás és debuggolás JTAG interfészen keresztül lehetséges. Ehhez egy JTAGICE mkII debuggert használtam. Az eszköz minden komponense 5V-ról működik, kivéve a GSM-mikrokontroller közötti UART-kommunikációt, mely 3.3V-ról működik.

Az áramellátásról két darab 2450mAh-s akkumulátor gondoskodik, amelyek USB porton keresztül tölthetőek. Az akkumulátorok töltött állapotban is csak 4V-ot adnak ki magukból, így egy DC-DC step-up konvertert (MP3410) alkalmazok a megfelelő 5V-os szint előállításához. A 3.3V előállításához pedig egy step-down konvertert (LP2985).

A hívásokat egy Huawei EM310 GSM modul kezeli, melyhez külsőleg csatlakoztatható az antenna, illetve a microSIM-kártya tartó. A modul a mikrokontrollerrel UART-on keresztül kommunikál, azonban a GSM modul RX és TX lábán 3.3V felel meg a high-szintnek, így a processzor felől jövő adatvonalon egy ellenállásosztó található, valamint a processzor felé lévő vonalon egy 3.3V-5V szintáttevő (74LVC1G07) található. A modul az RING kimenet-lábán jelzi, ha bejövő hívás van, amelynek a high-szintje szintén 3.3V. Emiatt szintén egy szintáttevőt raktam a GSM RING kimenete és a mikrokontroller megfelelő, bemenetnek konfigurált lába közé.

A kontaktok tárolásáért egy 512Kbit-es SPI buszos EEPROM felelős (25LC512), amely maximum kb. 1000 kontakt tárolását teszi lehetővé. Egy kontakt a következő elemekből épül fel: vezetéknév, keresztnév, telefonszám.

A kommunikáció USB porton biztosított. A mini-USB portot egy FT232 segítségével illesztettem a processzorhoz. Az USB porton keresztül lehetséges a kontaktok letöltése az eszközbe. A letöltés valós időben történik, amikor a telefon idle állapotban van.

A telefon házában található egy 9.7mm-es MCE-100 mikrofon, valamint egy 26W-os, 80hm impedanciájú vékony kivitelű hangszóró is.

A működési információkról három darab LED gondoskodik:

- „busy”: bekapcsoláskor, illetve feljelentkezéskor világít, valamint amikor USB-s kommunikáció során kontaktok kerülnek letöltésre az eszközbe.
- „gsm”: ezt a LED-et a GSM modul vezérli. A GSM modul bekapcsolása során folyamatosan világít, hiba esetén rövid időközönként villog, feljelentkezés után, idle üzemmódban pedig nagy periódusidővel, hosszú időközönként villog. Hívás alatt folyamatosan világít.
- „hb” (heartbeat): ez a LED ad információt arról, hogy az eszköz megfelelően működik-e. Rövid időközönként villan egyet, funkciótól függetlenül. Ha nem villog folyamatosan, akkor ez azt jelzi, hogy az eszköz nem működik helyesen.

A felhasználó számára hasznos információkat egy 4x16-os DEM 16481 LCD kijelzőn jeleníti meg az eszköz. Bekapcsolás során a feljelentkezési információkat írja ki. Feljelentkezés után, idle üzemmódban a következő információk láthatóak rajta: szolgáltató neve, térerő, dátum és idő, akkumulátor feszültsége, valamint töltés közben kiírja, hogy „CHARG”. A kontaktok betöltése során kiírja, hogy éppen aktív egy PC-ről jövő letöltési ciklus.

Az előlapon található egy vékony nyomtatott áramkör, amelyre rá van ragasztva kétoldalas ragasztóval egy fóliatasztatúra. Ezen helyezkednek el a nyomógombok: 0-9, *, #, valamint három darab funkció-gomb: „call”, „menu”, „end”, összesen 15 nyomógomb. A '+' karakter bevitelére nincs külön gomb, ehhez hosszan kell nyomni a 0-s gombot. A felső nyák egy szalagkábel segítségével csatlakozik az alaplapi nyákhoz.

Funkcionális leírás

Töltés

A telefont egy kapcsoló segítségével lehet bekapcsolni, azonban az akkumulátorok kikapcsolt állapotban is tölthetők. A töltéshez nem használtam külön specifikus töltő-áramkört, csak egy 4.7Ohm-os teljesítmény-ellenállást raktam sorba a bejövő tápfesz és az akkumulátorok közé, hogy a megfelelő árammal biztosítsam a töltést. A töltés során két dologra kell odafigyelni: ökölszabály, hogy tölteni a kapacitás tizedrészével szokás, valamint egy PC USB portja maximum 500mA áramot tud kiadni, tehát ennél nagyobb áramfelvételt nem engedélyezhetünk az eszköznek. Másrészt az akkumulátorok összkapacitása kb. 5000mAh, így maximum 500mA-rel tölthetem az eszközt. A tápfesz 5V, az akkumulátorok feszültsége közel 0%-os töltöttségi szint mellett kb. 2.5V, így $R = (5V - 2.5V) / 500mA = 5\Omega$ adódik, tehát egy 4.7Ohm-os, 1W-os ellenállást kötöttem sorba az USB felől jövő tápfesz és az akkumulátorok közé.

Bekapcsolási folyamat

A telefon bekapcsolása után megjelenik egy üdvözlő képernyő. Ez idő alatt történik meg az egyes komponensek inicializálása. Ezután adom rá a tápfeszt a GSM modulra egy FET segítségével. A tápfesz ráadása után várok egy kicsit, míg a kondenzátorok feltöltődnek, és utána egy tranzisztor segítségével szoftveresen bekapcsolom a modult. A GSM modul bekapcsoláskor, feljelentkezéskor és híváskor kb. 1,5A áramot vesz fel, így három darab 330uF-os kondenzátort raktam a VCC lábaihoz a kellő mértékű puffereelés érdekében, azonban sötétben még így is jól látható, hogy feljelentkezés közben annyira megrángatja a tápfeszültséget a modul, hogy ez meglátszik az LCD háttérvilágításán is.

Idle-mód

A sikeres feljelentkezés után lekérdezem a szolgáltató nevét, a térerőt és a dátumot a GSM modultól. A szolgáltató nevét jobbra igazítva írom ki az LCD első sorába. A térerő kijelzését úgy valósítottam meg, hogy a teljes skálát 8 részre osztottam, majd ennek megfelelő darabszámú négyzetet „töltök ki” az LCD második sorában, jobbra igazítva a térerősségnek megfelelően.

A dátumot és az időt a harmadik sorba íratom ki, az aktuális akkumulátor feszültséget pedig a negyedik sorba, balra igazítva. Ha az USB rá van csatlakoztatva az eszközre, akkor ezt egy „CHARG” szöveg kiíratásával jelzem a negyedik sorban, jobbra igazítva.

Ha éppen PC-ről kontaktok letöltése van folyamatban, akkor a folyamat ideje alatt „->MEM” szöveget íratok ki az első sorba, balra igazítva.

A sikeres feljelentkezés után az eszköz idle állapotba kerül. Ebben az állapotban érzékeny mindenféle külső beavatkozásra: ekkor lehet kontaktokat betölteni, az eszköz érzékeny a

gombnyomásokra, végig lehet pörgetni a kontakt-listát, valamint hívást lehet kezdeményezni és fogadni. Ebben az állapotban 20 másodpercenként kérdezem le a térerőt, dátumot és időt, valamint 2 másodpercenként kérdezem le az akkumulátorok feszültségét.

A „menu” gomb megnyomásával lehet belépni a kontaktok menübe. Ekkor kiolvasom az EEPROM-ból a legelső kontakt adatait, majd megjelenítem a kijelzőn. A gomb újbóli megnyomására kiolvasom a soron következő kontakt adatait, majd megjelenítem. Az utolsó kontakt megjelenítése után visszaugrok a memória elejére, és újra az első kontaktot jelenítem meg. Az utolsó sorba kiíratom, hogy összesen hány kontakt van a memóriában, valamint hogy éppen hányadik kontakt szerepel a képernyőn. A „call” gomb megnyomásával lehet hívást kezdeményezni, az „end” gomb megnyomásával lépünk vissza idle-módba.

Tárcsázás

Idle módban lehetőség van tárcsázni is tetszőleges számot, maximum 16 karakterig. A bevitt számjegyek a harmadik sorban jelennek meg, jobbra igazítva. A „call” gombbal lehet hívást kezdeményezni, az „end” gombbal visszalépünk idle módba. A '+' karakternek nincs külön fizikai gombja, így azt a 0-s gomb hosszan nyomásával lehet bevinni.

Másik lehetőség a tárcsázásra, hogy a kívánt kontaktot kiválasztjuk a telefonkönyvből, majd a „call” gomb megnyomásával tárcsázzuk a telefonszámot.

Bejövő hívás

Bejövő hívás során bekapcsolom a végfokot, és PWM segítségével kiadok egy „csengőhangot”, ami annyiból áll, hogy 2 másodperc periódusidővel 1 másodpercig kiadok egy bizonyos frekvenciájú hangot. Eközben lekérdezem a GSM modultól a tárcsázó telefonszámát, majd végigpörgetem az EEPROM-ot, hogy szerepel-e a hívó a telefonkönyvben. Ha igen, akkor nemcsak a hívó telefonszámot, hanem a hozzá tartozó nevet is kiíratom a képernyőre. A „call” gombbal lehet felvenni a hívást, az „end” gombbal lehet elutasítani.

Hívás közben

A hívás közben folyamatosan lekérdezem a GSM modultól, hogy aktív-e még a hívás. Megszakadt vagy a másik fél által befejezett hívás esetén automatikusan befejezi az eszköz a hívási folyamatot és visszatér idle állapotba. Hívás közben az „end” gombbal van lehetőség befejezni a hívást. Foglalt szám esetén automatikusan bontom a kapcsolatot, majd visszatérek idle állapotba.

Kontaktok betöltése az eszközbe

Lehetőség van PC-ről kontaktok letöltésére USB porton keresztül. Ennek meghatározott formátuma van: a kezdő karakter egy '!' felkiáltójel, majd utána következik egy lényegében tetszőleges karakterekből álló header-sor. Ezután soronként következnek a kontaktok. Egy-

egy sor reprezentál egy kontaktot. A sorok <CR><LF> vagy <CR> karakterekkel lehetnek elválasztva egymástól. Egy sor tetszőleges hosszú, de maximum 64 karakter lehet, valamint egy sorban a vezetéknév, keresztnév és telefonszám ’,’ vesszővel van elválasztva. A legutolsó kontakt után nem kell sorvégi karakter, hanem egy ’,’ pontosvessző jelzi, hogy vége a listának.

Hardver

Alkatrészek, komponensek az alaplapon

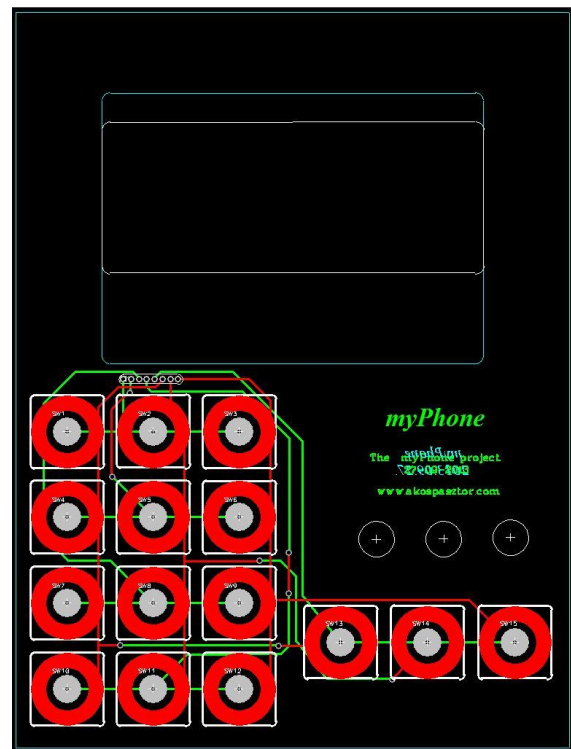
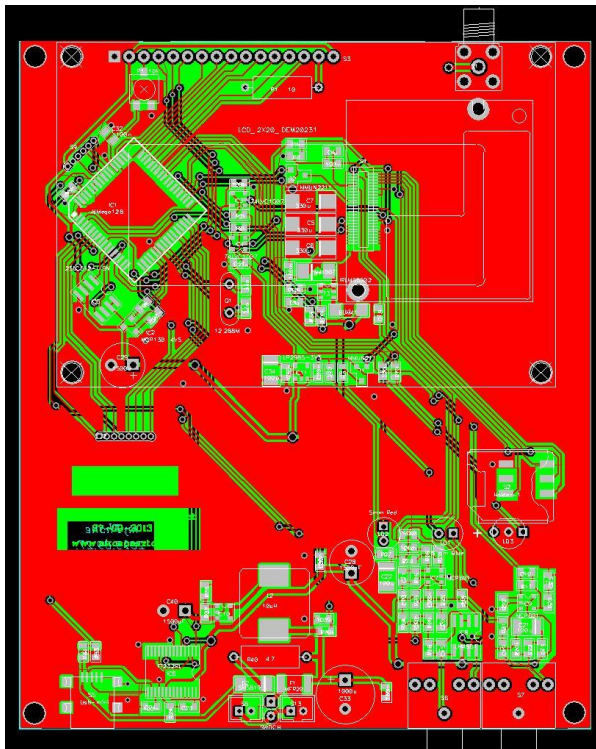
- Atmel AtMega128 mikrokontroller @ 12.288MHz
- JTAG interface programozáshoz és debuggoláshoz
- HUAWEI EM310 GSM modul microSIM-kártya tartóval, és külső antennával
- DEM 16481 SYH-LY 4x16 LCD kijelző (HD44780 kompatibilis vezérlővel) háttérvilágítással és kézzel állítható kontraszttal
- 512Kbit SPI EEPROM a kontaktok tárolására
- MP3410 step-up konverter az 5V előállításához
- LP2985 step-down konverter a 3.3V előállításához (a GSM-UART kommunikációhoz szükséges)
- 74LVC1G07 3.3V – 5V szintáttevő (a GSM-UART kommunikációhoz szükséges)
- FT232RL UART - USB konverter a kommunikációhoz, mini-USB porttal
- MCP601 előerősítő a mikrofonhoz és a hangszóróhoz
- LM4861 1W audió végfok (kikapcsolási lehetőséggel) a hangszóróhoz
- MCP130 reset IC
- LEDek, melyek tükrözik az eszköz állapotát
- Li-polimer akkumulátorok - 2 x 2450mAh
- MCE-100 9.7mm mikrofon
- Hangszóró (8Ohm, 26W)

Előlap komponentek

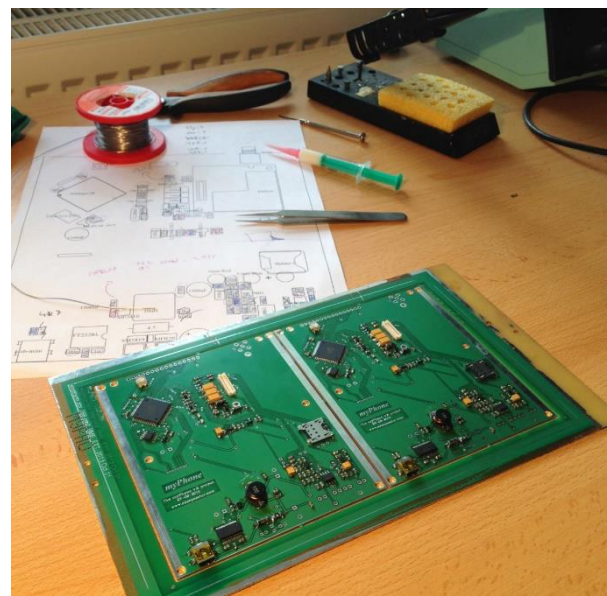
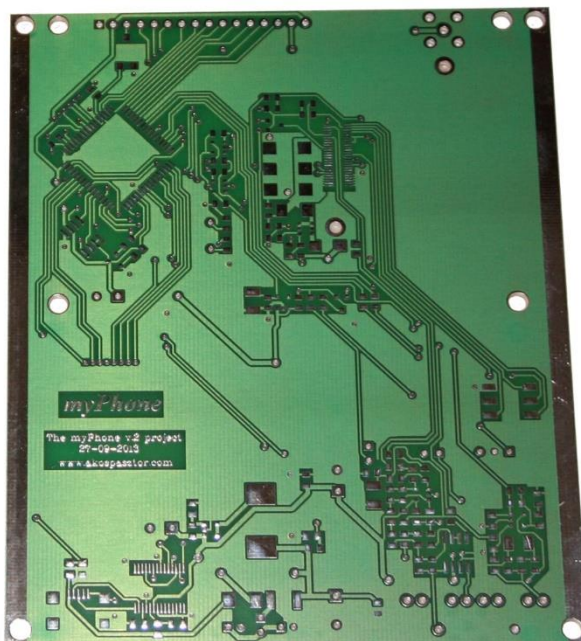
- Egyedileg tervezett, nyákra illesztett fóliatasztatúra kivágásokkal
- 15 alacsony nyomásponjú nyomógomb

Tervezés, gyártás, forrasztás

A kapcsolási rajzot és a layout-ot P-CAD szoftverrel terveztem. A tervezés során ügyeltem a NYÁK pontos méretére annak érdekében, hogy pontosan beleilleszkedjen a szabványos méretű alumínium dobozba.



Az elkészült terveket a BME ETT nyáklaborjában gyártattam le. A kész nyákra az összes alkatrészt kézzel ültettem be. A forrasztáshoz Weller WS-80 forrasztópákát használtam, valamint Stannol Kristall 505 0.5mm-es, 3.0% fluxtartalmú ónnal forrasztottam. A kényesebb alkatrészek esetén Ersz F-SW 32 folyasztószert használtam.



Az alumínium dobozba lyukakat, kivágásokat marattam a hangszóró membránja alá, a mikrofonnak, az USB csatlakozónak, a LED-eknek, a kapcsolónak, valamint a kijelzőnek.

A fóliatasztatúra tervét az előlapi NYÁK-terv alapján készítettem el ügyelve a megfelelő méretre és a kivágások, nyomógombok helyére. A tasztatúrát az Elo-Tec Kft. készítette el díjmentesen, hogy támogasson a projektben. A felhelyezéshez kétoldalas ragasztót használtam, amelyet kivágtam a nyomógombok, LED-ek és a kijelző helyén.

Szoftver

Blokkvázlat



Inicializálás

Az eszköz bekapcsolása után inicializálom a rendszert és perifériákat: System, LCD, UART, SPI, ADC, Timer és interrupt inicializálása. A szoftver felépítése időosztásos interrupt elven alapszik: 1ms időközönként generálok egy interruptot, majd az interruptban lekérdezem az összes periféria állapotát és flagekkel jelzem a main függvénynek, hogy mi az aktuális teendő. 12.288Mhz a mikrokontroller órajele, így 1ms alatt ~ 12 000 műveletet tudok elvégezni. Ebbe kell „beleférnie” az egész interrupt rutinnak, valamint maradnia kell elegendő időnek a műveletek elvégzéséhez a main függvényben a következő interrupt megérkezéséig.

Az inicializálás részben állítom be a watchdog timert, amit majd mindig az interrupt rutinban fogok resetelni.

LCD működése

Külön kitérek az LCD működésére. Az egyszerű kezelhetőség és formázás miatt létrehoztam egy tömböt, amely pontos másolata az LCD modul memóriájának. Ha éppen ki szeretnék valamit írni, akkor ebbe a tömbbe írom be a megfelelő helyre a karaktereket. A tényleges kiíratás menete pedig a következő: az interrupt rutinban van egy pointer, ami kezdetben a tömb első elemére mutat. Az interrupt alatt pontosan 1 karaktert íratok ki az LCD-re ebből a tömbből (amely karakterre a pointer mutat), majd növelem a pointert. Az utolsó karakter kiíratása után a pointert visszaugrasztom a tömb elejére, valamint az LCD-n kiadok egy „home” utasítást, amely következtében az LCD belső kurzora a legelső karakter helyére ugrik. Így 80ms alatt állandóan frissül a teljes képernyő tartalma. (Igaz, hogy az LCD csak 64 karakteres, de a RAM nem „folyamatos”, nem folyamatosan következnek egymás után a sorok. Ez azért van, mert a 4x20-as kivitel memóriája található benne gyárilag, és gondolom az egyszerűség miatt csak megvariálták a memória-kiosztást ahelyett, hogy folytonossá tették volna a kiosztást.)

Interrupt rutin

A beépített Timer2 komponens segítségével generálok 1ms-ként egy interruptot. Az interrupt rutin felépítése:

- Resetelem a watchdog timert
- LED-ek kigyújtása/villogtatása az aktuális státusznak megfelelően
- Kiíratok egy karaktert az LCD-re a display tömbből
- Megvizsgálom, hogy van-e bejövő karakter az USB porton. Ha igen, akkor elveszem a bufferből, és '!' karakter, akkor beállítok egy flag-et, ami jelzi a main függvénynek, hogy az utána következő karaktereket be kell tölteni az EEPROM-ba. Ha ';' karakter jön, akkor jelzem, hogy vége a letöltésnek. Ha a letöltési cikluson kívül érkeznek karakterek (amik nem '!', tehát nem letöltésindító karakterek), akkor azt vissza echo-zom az USB porton keresztül.

- Megvizsgálom, hogy van-e bejövő karakter a GSM felől. Ha igen, akkor elveszem a bufferből, és beállítok egy flag-et, ami jelzi a main függvénynek, hogy le kell kezelni
- Megnézem, hogy van-e bejövő hívás. Ezt a GSM modul jelzi a RING lábán. Ha igen, akkor beállítok egy flag-et, hogy majd a main függvényben le kell kezelni a hívásfogadást
- Megnézem, hogy az USB csatlakoztatva van-e az eszközhöz. Ha igen, akkor kiíratom, hogy „CHARG”. Ha nem, és előzőleg volt csatlakoztatva, akkor törölöm a kiírást.
- A GSM számára be van állítva egy timer célként szolgáló változó, aminek az értéke a 20s lekérdezési idő miatt 20000. Ezt a timert minden interrupt rutinban csökkentem eggyel. Ha eléri a 0-t, akkor beállítok egy flag-et, hogy újból le kell kérdezni a térerőt, időt és dátumot. A lekérdezés után újra beállítom a számláló értékét.
- Az akkumulátor számára is be van állítva egy számláló. Ennek kezdeti értéke a 2s-kénti lekérdezés miatt 2000, és minden interrupt rutinban csökken eggyel. Ha eléri a 0-t, akkor beállítok egy flag-et, ami jelzi, hogy le kell kérdezni az akkumulátor feszültségét. A lekérdezés után újra beállítom a számláló értékét.
- Billentyűzet-kezelés: megvizsgálom, hogy éppen van-e lenyomott gomb. Ha előzőleg nem volt gombnyomás, akkor indítok egy számlálót 0-ról. Minden interruptban növelem a számláló értékét, ha ugyanaz a gomb van éppen lenyomva. Ha eléri az 50-et, akkor azt már érvényes lenyomásnak veszem (50ms), és beállítok egy flag-et megfelelő értékkel, hogy történt gombnyomás. A beállított érték mutatja a main függvény számára, hogy milyen gombot nyomtak meg. Ha a 0 van lenyomva, és a számláló értéke eléri az 1000-et (1s), akkor 0 helyett '+' karaktert íratok ki.

Main függvény

A main függvény az inicializáció mellett tartalmaz egy végtelen ciklust. Ebben a végtelen ciklusban hajtom végre a megfelelő utasításokat. A flag-eket vizsgálva futtatom a megfelelő függvényeket, hogy az eszköz a kívántnak megfelelően működjön. A main függvény feladata továbbá:

- Ha nincs bekapcsolva a GSM, akkor azt be kell kapcsolni, majd feljelentkezni a hálózatra
- Térerő, dátum és idő lekérdezése, kiíratása
- Akkumulátor feszültségének lekérdezése, kiíratása
- Ha van bejövő hívás, akkor a hívó telefonszámát meg kell jeleníteni a kijelzőn, valamint a telefonkönyvben meg kell keresni, hogy tartozik-e hozzá megfelelő név. Az adatokat meg kell jeleníteni a kijelzőn (ha van érvényes találat a telefonkönyvben, akkor a hozzá tartozó nevet is), valamint ez esetben csak a „call” és „end” gombok aktívak. Az interrupt rutin felelős a csengőhangért.
- Aktív híváskor folyamatosan le kell kérdezni a GSM állapotát. Itt is csak az „end” gomb aktív. Ha a másik fél lerakta, vagy megszakadt a hívás, akkor visszatérek idle módba.
- Tárcsázáskor meg kell adni a GSM számára a szükséges parancsokat, és ez esetben is csak az „end” gomb aktív

- USB-ről jövő karakter esetén, ha letöltési ciklusban vagyunk (már volt '!' karakter), akkor az éppen bejövő karaktert az SPI buszon keresztül bele kell tölteni az EEPROM-ba. Ezt addig folytatom, amíg nem kapok ';' karaktert.
- A billentyű lenyomásokat le kell kezelni. Ha számok (vagy *, #, +) kerültek lenyomásra, akkor megjeleníteni a kijelzőn a tárcsázandó számokat, jobbra igazítva. Ekkor aktív a „call” gomb is. Ha „end” gomb került lenyomásra, akkor vissza kell térni idle állapotba. Ha „menu” gombot nyomták meg, akkor meg kell jeleníteni a soron következő kontaktot. Ehhez az EEPROM-ból ki kell olvasni a megfelelő, soron következő sorszámú kontaktot, majd meg kell jeleníteni az adatokat a kijelzőn. Ekkor a „call” gomb is aktív, melyet lenyomva tárcsázható az éppen kiválasztott kontakthoz tartozó telefonszám.

Felhasználói leírás

A telefon bekapcsolásához el kell csúsztatni a bekapcsoló gombot. Ezután az eszköz automatikusan feljelentkezik a hálózatra. A feljelentkezés után megjeleníti az információkat a kijelzőn, ez jelenti azt, hogy idle állapotban van. Innentől kezdve érzékeny a gombnyomásokra.

A számgombok segítségével lehet beütni a kívánt telefonszámot. A '+' karakter beviteléhez legalább 1s-ig a 0-s gombot kell nyomva tartani. Ezután a zöld „call” gomb megnyomásával lehet tárcsázni, a hívást pedig a piros „end” gomb megnyomásával lehet megszakítani.

A sárga „menü” gomb megnyomásával lehet a kontaktokat megjeleníteni. A „menu” gomb újbóli megnyomásával tudunk lépni a következő kontaktra. Az összes kontakt végigpörgetése után a legelső kontaktot fogja az eszköz újból megjeleníteni. A „call” gomb megnyomásával lehet tárcsázni a kiválasztott kontakthoz tartozó telefonszámot. Az „end” gomb megnyomásával lehet visszatérni idle állapotba.

A kontaktok betöltését bármilyen terminal emulátor program segítségével elvégezhetjük. A szükséges formátum a *Funkcionális leírás* szekcióban megtalálható. A szükséges beállítások: 2400bps (baudrate), 8bit data, 1 startbit, 1 stopbit, nincsen paritásbit és handshaking. A kontaktok betöltése bármikor megtehető, ha a telefon idle állapotban van. A betöltés után a „menu” gombbal ellenőrizhetjük, hogy helyesen betöltődtek-e a kontaktok.

Bemérési tapasztalatok

Bekapcsoláskor, feljelentkezéskor és hívás közben 1 - 1.5A áramot, idle állapotban 130-150mA áramot vesz fel. Ez egyébként sötétbe is tapasztalható: például hívás közben a kijelző néha elhalványodik, ahogy rángatja a GSM modul a tápfeszültséget, majd visszatér a teljes fényerőre.

Idle állapotban azért ilyen relatíve nagy az áramfelvétel, mert a kijelző állandóan világít, be van kapcsolva.

Egy interrupt rutin átlagos hossza: 130us. Tehát kb. 870us marad a main függvénynek elvégezni a teendőket a következő interrupt megérkezéséig.

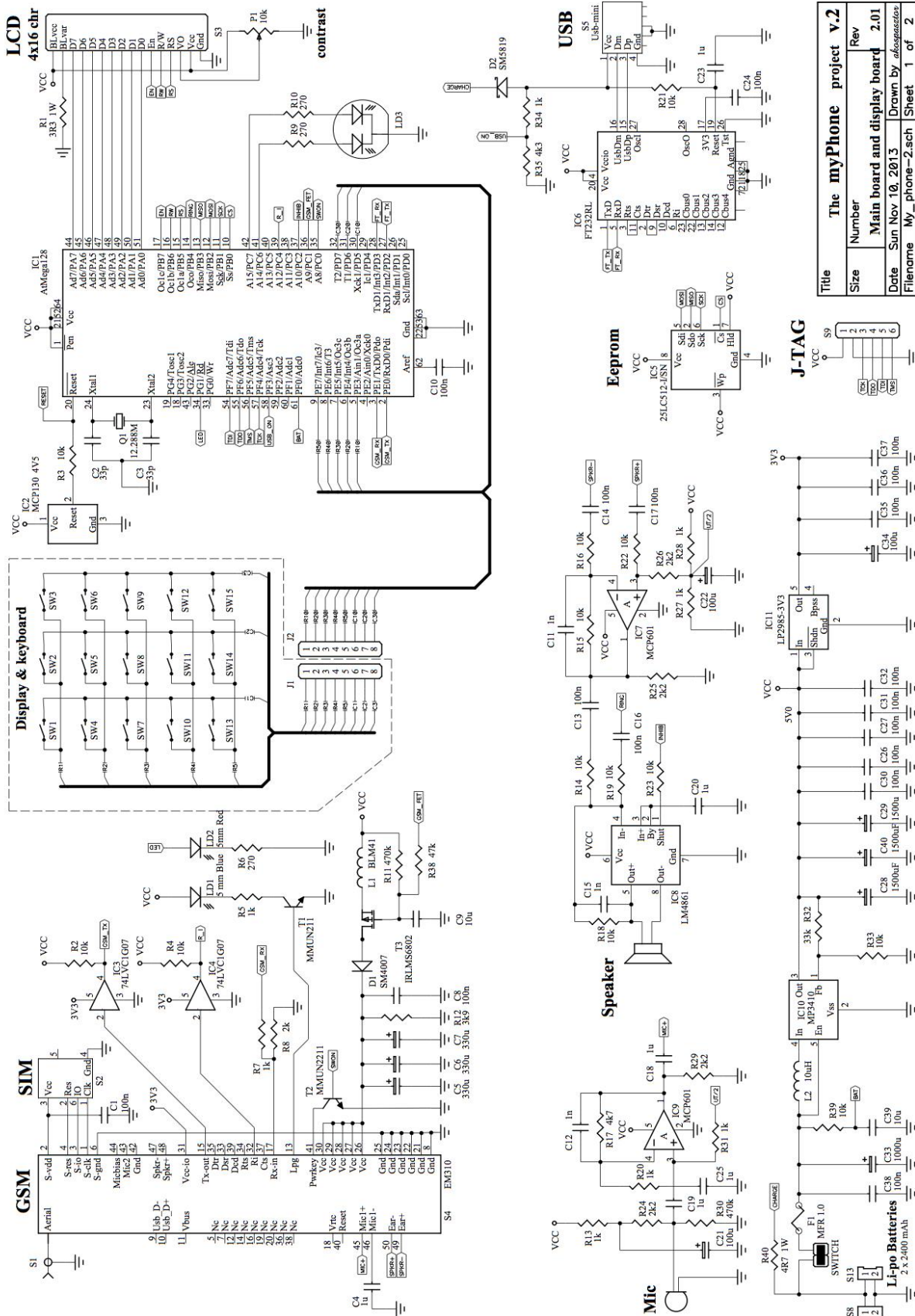
Függelék

1. Képek az eszközeiről
2. Kapcsolási rajz
3. Szoftver blokkvázlata
4. Szoftver forráskódja
 - binary.h
 - interrupt.h
 - main.h
 - interrupt.c
 - main.c

Képek az eszközről



Kapcsolási rajz



Title	The myPhone project v.2
Size	Number
Date	Main board and display board 2.01
Filename	Sun Nov 10, 2013 Drawn by akospasztor
	Sheet 1 of 2

Blokkvázlat

myPhone v2.0 software UML diagram

<http://akospasztor.com>

