

**Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Automatika Intézet**



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

AVR MIKROVEZÉRLŐVEL FELÉPÍTETT GRAFIKUS LCD-S MP3 LEJÁTSZÓ ESZKÖZ

Szerző(k):

Pruck Zsolt

villamosmérnöki szak, III. évfolyam

Konzulens(ek):

Lamár Krisztián

főiskolai adjunktus

Budapest, 2013

Tartalomjegyzék

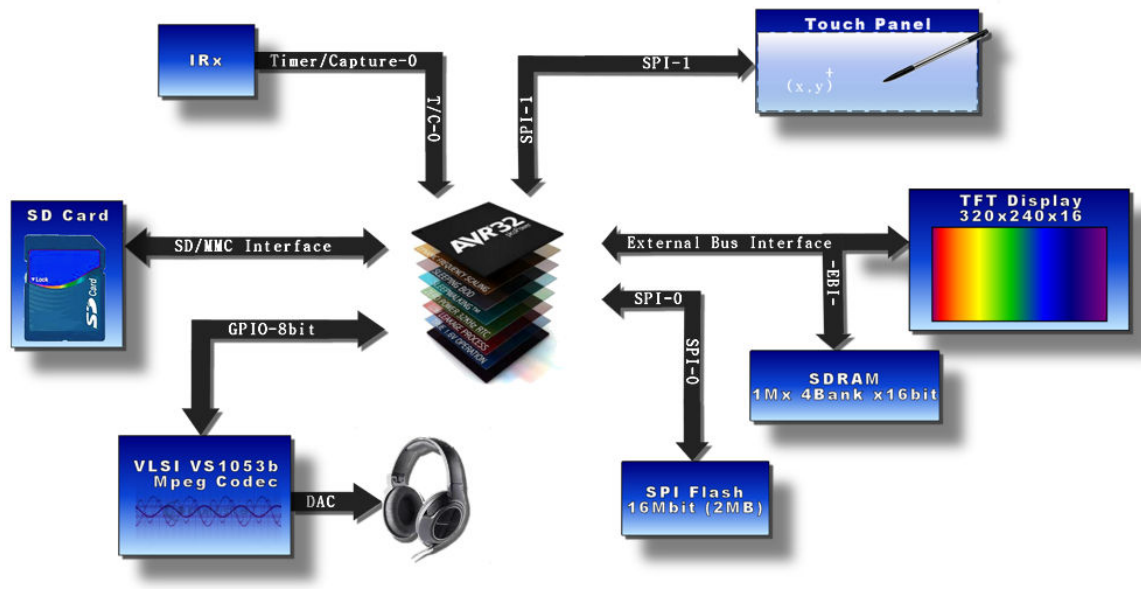
AVR mikrovezérlő felhasznált részegységei.....	3
A felhasznált részegységek rövid ismertetése	4
MPEG layer kodek	4
SD kártya leírása	5
TFT Kijelző tulajdonságai.....	6
SPI flash memória	7
Bővíthetőség, kommunikáció más egységekkel.....	8
Távírányítás	8
Szoftver megtervezése és felépítése	9
FAT32-es fájlrendszer ismertetése, adatfeldolgozása	12
Grafika felépítése és tárolása a flash memóriában	13
Kiegészítő funkciók, az MP3 kodekben	16
Felhasznált irodalom	17

AVR32 mikrovezérlő felhasznált részegységei

A projekt tervezése során felmerült a kérdés, hogy hogyan lehet a legtöbbet kihozni ebből a mikrokontrollerből. A válasz a perifériákban rejlik, ugyanis hardveresen történik az adatkommunikáció a részegységek között, valamint a hasznos processzoridőt nem az átvitel terheli le, hanem az információk feldolgozása. Továbbá a 8 csatornás közvetlen memória hozzáférésű DMA, az adatok mozgását a processzor használata nélkül teszi lehetővé.

Részegységek kapcsolata a mikrokontrollerrel:

- SD kártya: A mikrovezérlő tartalmaz kifejezetten erre a célra szolgáló perifériát, és lehetővé teszi az egyszerű és gyors 4 bites adatátvitelt. Jelen projektben az elért maximális olvasási sebesség körülbelül 1.5MB/sec.
- TFT kijelző: Ez a megjelenítő egy ILI9325 típusú meghajtó IC-t tartalmaz, mely 16-bites adatbusszal rendelkezik, így lehetővé teszi a külső memóriavezérlő perifériához való kapcsolódást. Előnye, hogy nem kell szoftveresen vezérelni az adatátvitelt, csupán meg kell címezni egy pointerrel azt a memória területet, ahova az csatlakoztatva lett. Átviteli sebessége elegendő a 30 fps animációk vetítése 320x240-es felbontás mellett, ugyanis 1 pixelt egy ciklus alatt lehet beírni a kijelzőbe.
- SPI Flash memória: A program és grafikus elemek tárolására szolgáló flash memória az SPI portra van illesztve, az IC maximális órajele 80MHz viszont ezt nem használjuk ki, mivel magasabb órajelnél már az adatátvitel az elektromos zavarjelek miatt már nem stabil, így az eszköz 24MHz-en fut.
- MP3 kodek: A hardveres dekóder/enkóder kommunikációs felülete szintén SPI jellegű, azonban mégsem kötöttem perifériához. Ennek oka a későbbiek során kiderül.



A lejátszó elvi felépítése

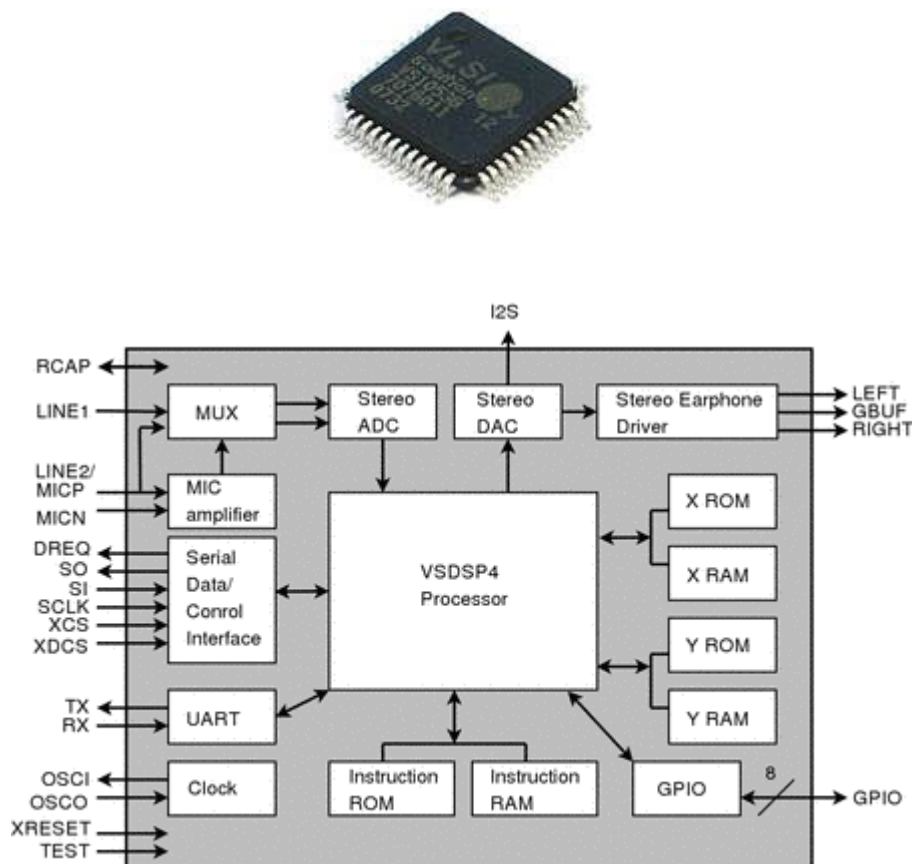
A felhasznált részegységek rövid ismertetése

MPEG I-II-III Layer kodek

Első sorban a VLSI cég által gyártott hardveres sztereo MPEG kodeket ismertetném. Felépítése első ránézésre nem tűnik bonyolultnak, tartalmazza a SPI soros kommunikációs protokollt, ami a vezérlési parancsok kiküldésére, regiszterek írására olvasására, valamint az audio adatfolyam átvitelét biztosítja, ezen felül vezérlő- valamint GPIO lábakat is tartalmaz. Előzőleg említésre került, hogy közvetlenül nem tudjuk a hardveres SPI buszra kötni, mivel a chip deaktiválása után – ChipSelect láb deaktiválásra billentése – az adat- valamint az órajel vezeték nem kerül magas impedanciás jelszintre, így blokkolva a buszra kötött eszközök kommunikációját.

Ezt a jelenséget a kodek beállítása az „SDI Share” funkció okozza, ahol a parancs és az adatfolyam egy buszon történik a ChipSelect lábbal megkülönböztetve. Ennek okán kénytelenek vagyunk szoftveresen megoldani a problémát.

Sajnos ennek hátránya is van, ugyanis a hasznos processzor időből kell gazdálkodnunk, valamint több GPIO, azaz általános felhasználású ki- és bemenetet kell felhasználnunk. Az eszköz tartalmaz továbbá 16 bites DAC átalakítót, ami közvetlenül képes meghajtani egy 30ohm impedanciájú fejhallgatót, kiegészítés céljából csupán egy csatoló kondenzátort kell csatornánként beépíteni. A chip blokkvázlatát az alábbi ábra szemlélteti:



VS1053b típusú MPEG dekóder blokkvázlata

SD kártya leírása

Manapság a legelterjedtebb adathordozónak számít ez a flash memória, ezért ez került felhasználásra. Azonban hogy tudjuk használni, tisztában kell lennünk pár dologgal:

- Hogyan kommunikálhatunk vele?
- Mekkora adattranszferre képes?
- Hogyan illeszthető a controllerhez?

A kommunikációt tanulmányozva különböző specifikációkból kétféle adatátviteli módot támogat:

- SPI soros adatátvitel
- Párhuzamos 4 bites adatátvitel

Mivel számunkra rendelkezésre áll az SD/SDIO/MMC 4 bites periféria és támogatja a nagy sebességű adatátvitelt így ez került felhasználásra. Eddigi méréseim alapján a mikrovezérlő 33MHz-en - PBA buszmátrix sebessége - 1 bájt átvitele 904 nsec, ami körülbelül 4MB/sec adattranszfert biztosít, de mivel fájlrendszert alkalmazunk, ez a sebesség lecsökken 2-2.5MB/sec-ra. Ez bőségesen biztosítja a megfelelő adatsűrűséget a kodek számára is, számítások alapján 320kbit/sec → azaz 40kByte/sec maximum.

Az írási és olvasási műveletek idő diagrammját, parancskészletét a specifikáció tartalmazza.

A kártya belső felépítését az alábbi ábra szemlélteti:

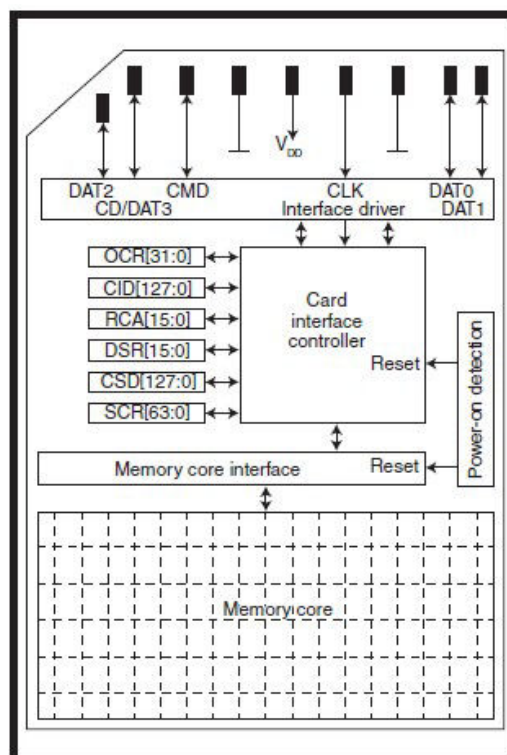


Figure 2—This is the internal architecture of an SD card. (Source: Transcend, www.transcendusa.com)

SD kártya blokkvázlata

TFT Kijelző tulajdonságai

Mivel multimédiás eszközről van szó, szinte elengedhetetlen a grafikus kijelző. Ezen projektben egy 18/16 bites színmélységű TFT kijelző került felhasználásra, amin egy rezisztív érintő panel található meg. Tulajdonságát jellemezve sokoldalú, teljesen személyre szabható beállításokat lehet alkalmazni. Ilyen például a bevitel iránya, képfrissítési frekvencia állíthatósága, kontraszt és gamma változtatása, kijelölhető GRAM írási tartomány valamint a grafika méretezési faktora állítható. Fényerejét a controller egyik PWM csatornáját felhasználva egy végtranzisztor segítségével fokozatosan változtathatjuk.

Amit tudni kell róla:

- Felbontása 320x240xRGB
- Színmélysége 16bit, amennyiben 24 bites grafikával dolgozunk szoftveresen színkódolást kell végrehajtani. Pixel formátuma: RGB:565
- Kommunikációja párhuzamos 16 bites (1 pixel / adattranszfer)
- Állítható beviteli irány: Vízszintes, függőleges



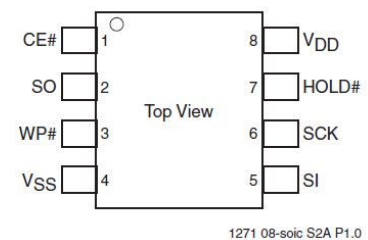
Alkalmazott TFT kijelző képe

SPI flash memória

A projektben szükségünk van egy relatív nagyobb memóriára, ahol a grafikákat – háttér, üdvözlő képernyő, egyéb üzeneti ablak -, beépített hang effekteket – bejelentkező dallam - tárolhatjuk. Választásom egy Silicon Stroeage Technology cég által gyártott 16Mbit-es, vagyis 2MB-os soros flash memóriára esett. Ár és kapacitásméret aránya kedvező, valamint SPI buszra illeszthető. Későbbiek során a rajta levő adatokat automatikusan tudja frissíteni a mikrovezérlő, amennyiben az adathordozó SD kártyára csak ezek a frissítési fájlok kerülnek. 80MHz-es maximális órajel frekvenciája bőségesen elegendő az alkalmazott feladatra. Előnye, hogy külsőleg is írásvédetté tehető elkerülve az esetleges akaratlan frissítést, mondjuk egy kártyán felejtett fájlok miatt.

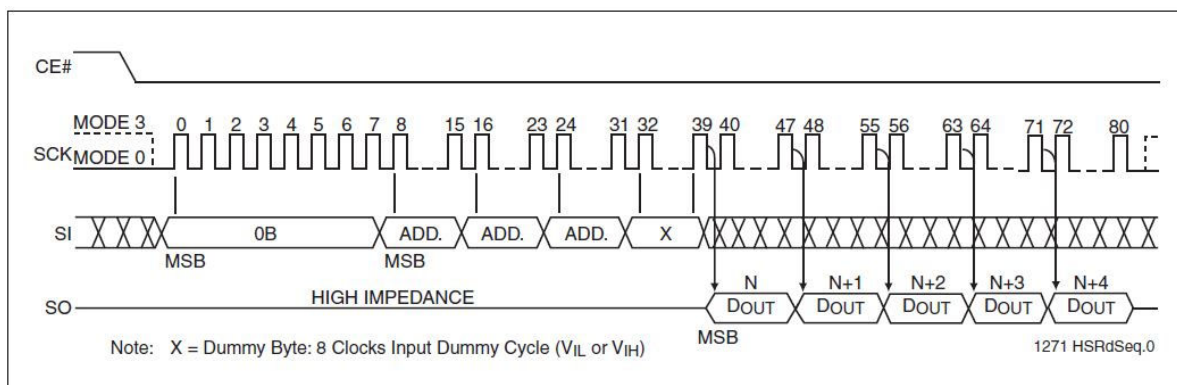
Egyéb hasznos információk:

- 100.000 maximális írás/törlés
- 10mA-es aktív műveletvégzési áramfelvétel
- 35ms-os teljes chip törlés
- AAI – automatikus címnövelés írás és olvasás közben, ezzel meggyorsítva a műveletet, mivel nem kell a memóriarészeket folyamatosan címezni.

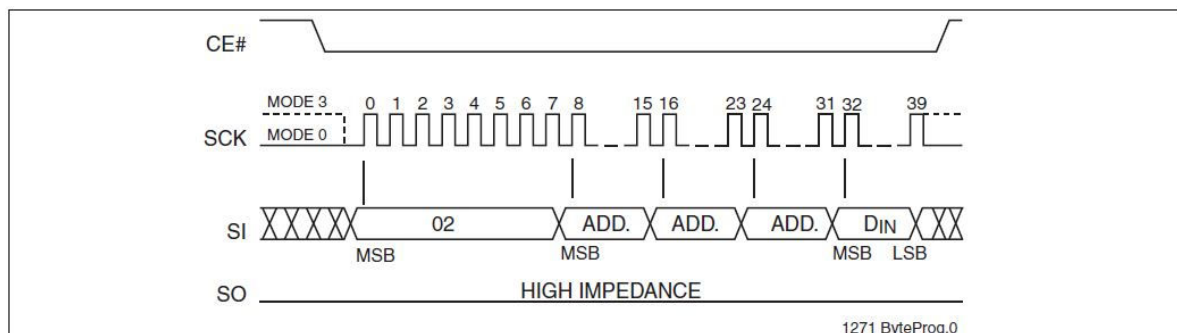


Kommunikációjáról bővebb információt az adatlapjában szereztem.

Az írási valamint a kiolvasási idődiagramot az alábbi két ábra szemlélteti:



Folyamatos olvasási ciklus



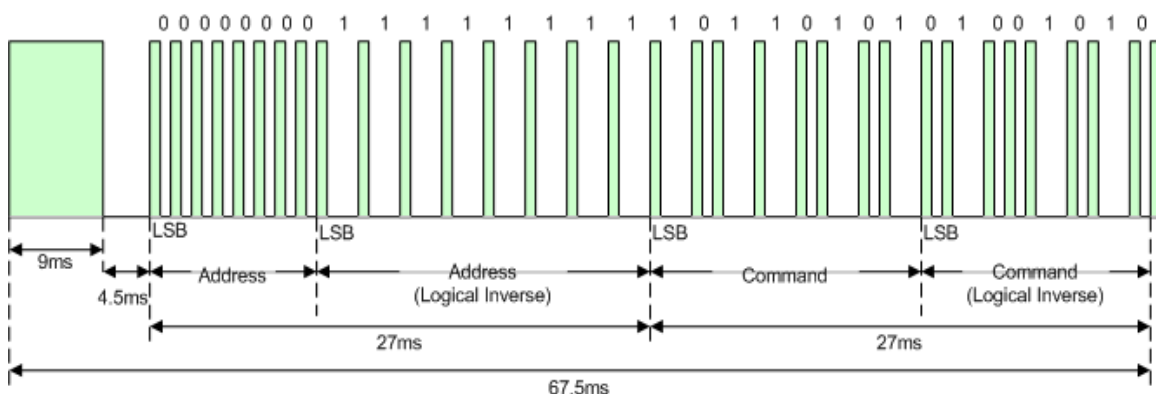
Folyamatos írási ciklus

Bővíthetőség, kommunikáció más egységekkel

Magasabb komfortfokozattal bír egy zenelejátszó, ha annak irányítása, kihangosítása, adatátvittele más eszközökkel is megvalósítható, gondolok itt például egy Bluetooth-os rádióátvitelről. Manapság már egyre elterjedtebb a hordozható számítógépekben, telefonokban ez az egység, így szinte az eszköz fizikai érintése nélkül is frissíthetők a zenefájlok, valamint irányítható a lejátszás is.

Távírányítás

Az infra távirányító jelének dekódolása is helyet kapott a projektben, jelen esetben a **NEC** típusú protokoll. Azért ezt választottam, mert nagy százalékban ezek a távirányító típusok vannak forgalomban.



NEC típusú protokoll oszcilloszkópos képe

A jelfeldolgozáshoz a szabványleírás tanulmányozva, valamint a megfelelő illesztést követően az AVR32 Timer/Capture megszakításos időmérés funkcióját alkalmaztam. Ezzel két megszakítás közti időtartamot lehet megmérni a bázis órajel és az időzítő előosztó ismeretében. Ahogy azt az ábra mutatja, egy startjelet követően a céleszköz címe és a címének egyes komplemente, majd ezután a parancs, valamint parancs egyes komplemente kerül kiküldésre. A logikai „0” szint impulzus ideje oszcilloszkópos mérés alapján 560us, valamint a logikai „1” jelszint impulzus ideje 1.6ms.

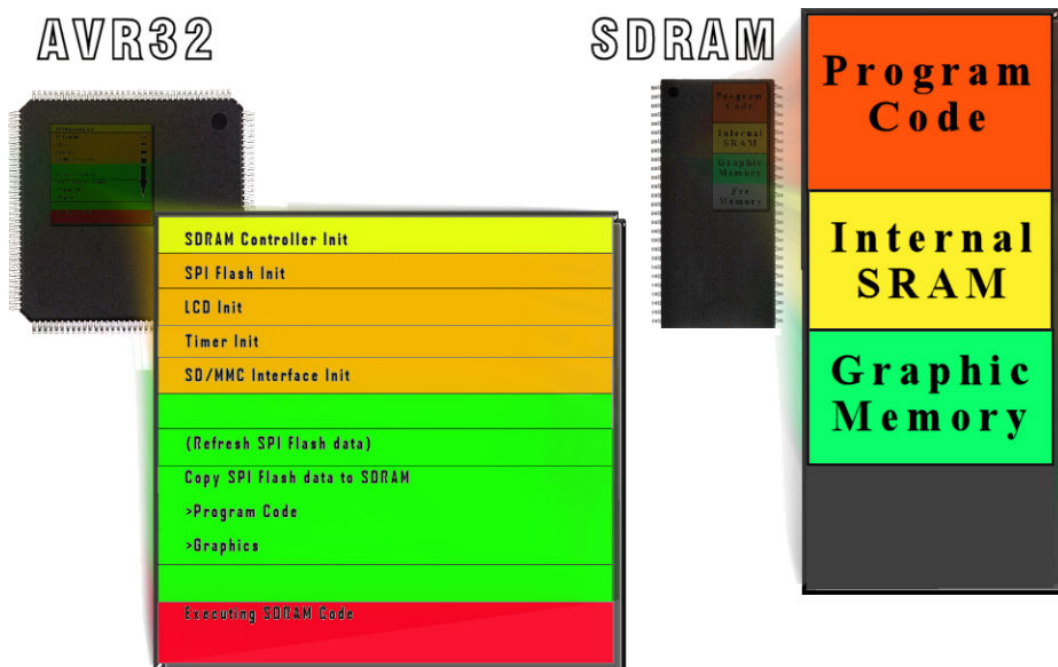
Ezen adatok feldolgozása után már a kapott 8 bites címmel és 8 bites paranccsal már lehet különböző funkciókat hozzárendelni az egyes gombnyomásokhoz. Bővíthető funkció, hogy a fogadott adatot külső Flash memóriában tárolható, ezáltal más távirányító jeleit is meg lehet vele taníttatni.

Szoftver megtervezése és felépítése

A hardveres tervezés után ez a másik legfontosabb része a projektnek, ugyanis itt dől el az a tény, hogy milyen műveletekre mennyi idő fordítható, milyen sorrendet kell betartani a helyes működéshez. Legelső lépésben tanulmányozni kellett, hogy egyes részegységek milyen időigénnyel rendelkeznek, majd ezután meg kellett határozni egy prioritási sorrendet. Továbbá az adott mikrovezérlő belső programmemóriája nem elegendő a lejátszó programjának tárolására és futtatására, így a következő lépések voltak szükségesek:

- Inicializáló programrész valamint a frissítési műveletek a kontroller belső programmemóriájában helyezkedik el
- A felhasználói program egy külső Flash memóriában tároljuk, majd az alap program, ezt bemásolja az SDRAM területére
- Amennyiben külső adathordozón egy megfelelően elnevezett fájlt talál, azt programfrissítésnek tekinti így a Flash memóriában található szoftvert felülírja, így a következő indításnál már egy másik verzió futtatható
- Ezt követően a gyorsabb adatfeldolgozás elérése érdekében a grafikák kerülnek átmásolásra
- Hogy ne ütközzünk további korlátokba, a fordító Linker részében a belső SRAM memóriaterületet is az SDRAM-ban jelöltem ki, ugyanis összesen 8MB-os területtel bátrabban lehet gazdálkodni.

Az adatok elhelyezkedését az alábbi ábra szemlélteti:



Az adatok elhelyezése a kontrollerben és az SDRAM-ban

Az MP3 felhasználói felületű programja az alábbiak szerint épül fel.

Inicializáló rész:

- Üdvözlő képernyő betöltése, folyamatos fényerő fel- és leúsztatásával
- SD kártya elemzése, fájlrendszerének meghatározása, valamint a fájlok felderítése történik meg
- Lejátszó főképernyőjének betöltése, fájlnev és elérési mappa kiírítása
- MP3 kodek inicializálása

Lejátszási rész:

- Zeneszám headerjének betöltése a hardveres kodekbe, majd onnan a feldolgozott audio tulajdonságok bekérése: bitsűrűség, mintavételezési frekvencia, csatornák száma, kódolás típusa.
- Zeneszám lejátszása függvényhívással történik meg, ez alatt a kodek egy külső megszakítással jelzi a mikrokontroller felé, hogy a FIFO puffer hamarosan kiürül, és szükségessé vált a feltöltése.
- Szöveges információk frissítése a képernyőn, animálása Winamp stílusban



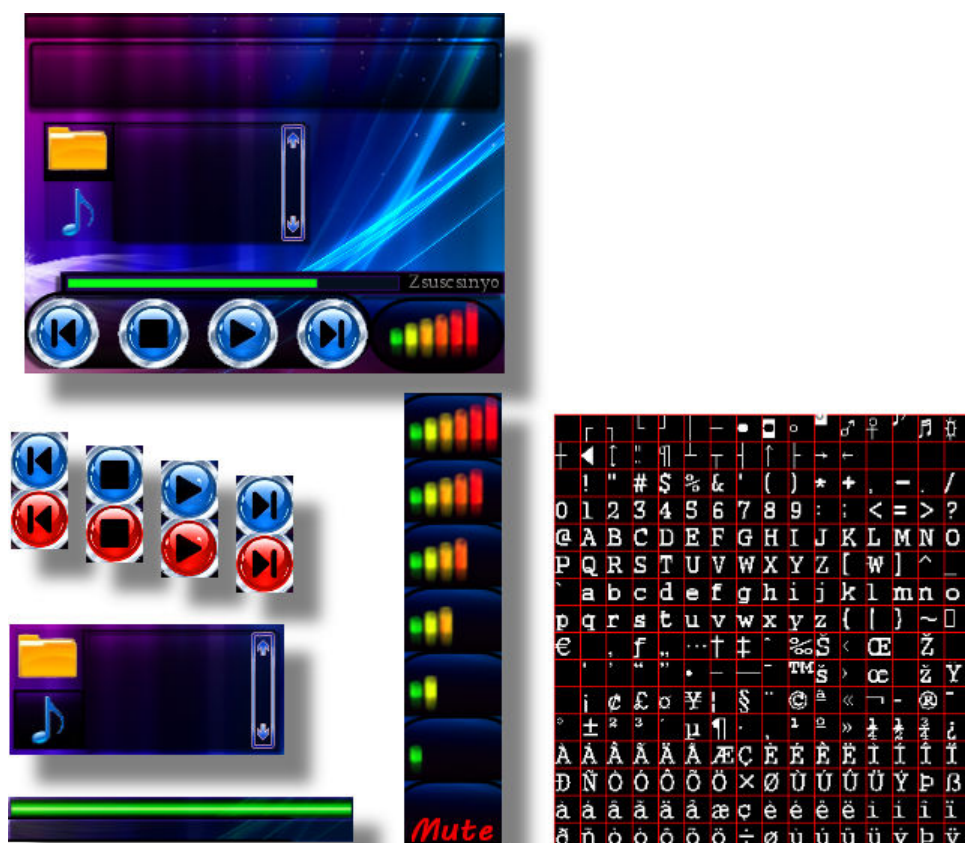
A kijelző felosztása

A lejátszás közbeni grafikus felület frissítése, valamint az érintő képernyő lekérdezése, parancsvégrehajtása a kodek pufferének feltöltését követően történik egészen addig, amíg nincs szükség újabb feltöltésre

A kép bizonyos részein történő grafikai változást, például nyomógomb lenyomást illetve hangerő változtatást animálása szekvenciális képsorozatok egymás utáni vetítésével jön létre.

A karaktertábla szintén egy összefüggő képet alkot, amelyből a megfelelő betű lokalizálása után végigpásztázza annak területét, és a háttér- és előtér színét figyelembe véve egy maszkot kapunk, amit már a kijelző tetszés szerinti pontjára kivetíthetünk.

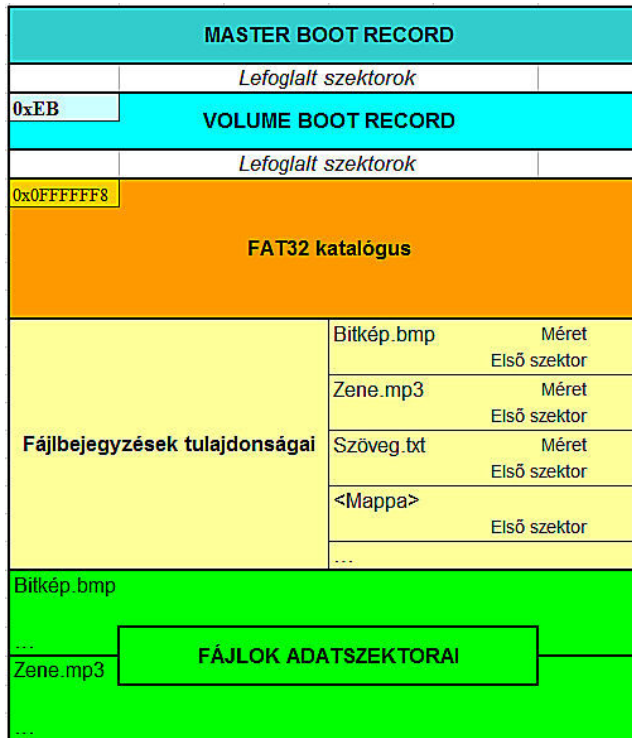
A lejátszó felületét alkotó grafikus szegmensek az alábbi ábrán láthatók:



Grafikus elemek és karaktertábla illusztrálása

FAT32-es fájlrendszer ismertetése, adatfeldolgozása

Az SD kártyás felhasználás, valamint a számítógépről történő fájlkezelés megköveteli a fájlrendszer használatát, amivel a mikrovezérlő képes lesz navigálni a fájlok és mappák közt, részletes információ leírást kapni róluk. Az adott feladathoz a FAT32-es fájlrendszert választottam, mivel szinte minden nagyobb méretű SD kártya ezt használja. Részletes leírást a specifikációban találtam, valamint tanulmányoztam, hogy miként látható a lemezen.



A Master Boot Record (MBR): vagy más néven a partíciós szektor a merevlemez legelső szektorának (azaz az első lemezfelület első sávjának első szektorának) elnevezése. Csak a partícionált merevlemeznek van MBR-jük. A MBR a merevlemez legelején, az első partíció előtt található meg.

Volume Boot Record (VBR): ezen a területen található az az információk, amik az adott partícióra vonatkoznak. Ilyen például a fájlrendszer típusa, mennyi lefoglalt szektor van a katalógus előtt, hány szektort foglal magában egy klaszter, valamint megadja a legelső adatszektor elhelyezkedését.

FAT32 katalógus: ez egy olyan szakasza a fájlrendszernek, ahol az adatszektorok hely-indexei láncszerűen kapcsolódnak

egymáshoz, ha az adat nem folytonos, ezen keresztül találhatóak meg a fájl egyes részei a lemezen.

Fájlbejegyzések tulajdonságai: itt találhatóak meg a fájlok részletes információi, amik az alábbiak:

- DOS(8.3) fájlnev
- Hosszú fájlnev
- Típus: mappa vagy fájl az adott objektum
- Első adat klaszter helye a lemezen és a katalógusban
- Fájl méret
- Fájl létrehozásának ideje
- Fájl attribútumai

Fájlok adatszektorai: ezen a területen helyezkednek el a fájl tartalmi adatai, folytonos mérete a Volume Boot Record információi között a szektor/klaszter határozza meg, folytatását viszont a katalógusból olvassuk ki.

Grafika felépítése és tárolása a flash memóriában

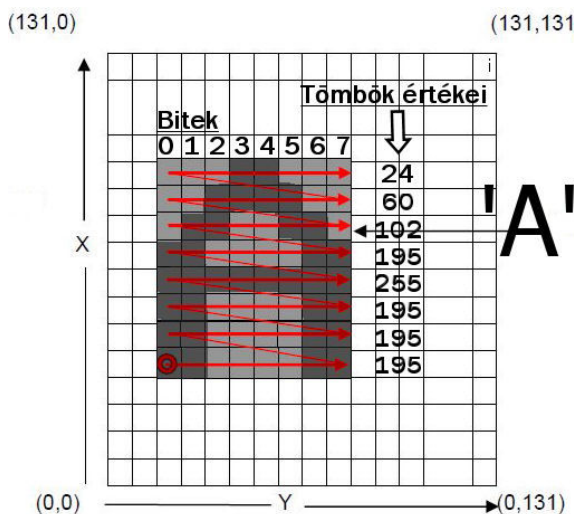
Kivitelezés szempontjából előnyösnek tartottam, hogy tömörítetlen adatokkal dolgozzak, ugyanis a feldolgozás így jóval rövidebb ideig tart. Általánosságban tekintve minden grafikus program tudja kezelni a bitképeket, ezzel tetszés szerinti felületek könnyedén szerkeszthetők. 1 Pixel tartalmát 320x240-es felbontással 24-bites színmélységet alkalmazva 3 byte-on tudjuk eltárolni. Ez összesen 225 KB területet foglal el. A felhasznált 2MB-os memóriát tekintve előre átkonvertált grafikát tárolunk, ugyanis ez 2/3-ára csökkenti a méretet, és így jobban kihasználhatjuk a fennmaradó tárhelyet.

Amikre szükség van:

- Bejelentkező képernyő
- Fő háttér, ami a lejátszás közben látható
- SD kártya behelyezését közlő kép
- Animációs képszezmensek

Ezek összesen 1.3 MB-ot foglalnak el, a fennmaradó memóriaterületet a fejlesztés folyamán használjuk fel, valamint ezen tároljuk a program kódot is.

Betűk alapértelmezett maszkolása, kirajzolásának menete:



A grafikus karakterszimbólumból történő betűmaszk kimásolása mellett még lehetőség van generálásra is. Ez azért hasznos, mert ha későbbiek során nem használunk külső memóriát, akkor is tudunk szöveget megjeleníteni. Első lépésben, egy ideiglenes tömbben generáljuk ki a karaktereket az alábbi ábra szerint. A 7. tömb értékét vizsgáljuk bitek szerint, ha az logikai 1-es akkor az adott címre beteszi a betű színének megfelelő pixel adatokat, azonban ha logikai 0-s szint, akkor viszont a háttér színének megfelelő értéket tesz be a grafikus memóriába.

Ezt a műveletet minden sornál végrehajtja, egészen addig, amíg a tömb legelső eleméhez nem ér. Átlátszó háttérszint nem lehet generálni, azt viszont meg tudjuk tenni, ha logikai 0 szintet talál, akkor megkeresi a legközelebbi 1-es értéket, és arra a címre beállítja a grafikus memóriát.

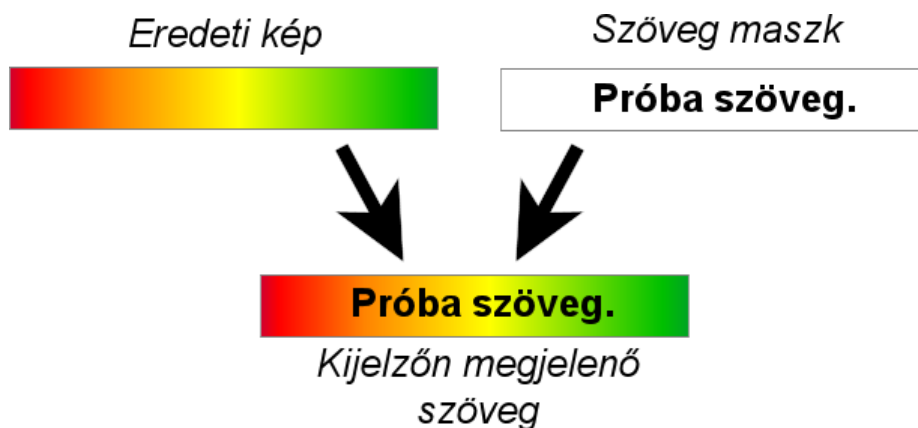
Grafikus betűkészlet kirajzolásának menete:

Az előzőekben már megemlített karaktertáblából egy adott betű lokalizálásához függvényt alkalmazunk, mely megfelelő címtérületre ugrik, majd sorról sorra olvassa ki a teljes betűmaszkot. A grafikus tábla mellett még egy található egy adathalmaz, mely tartalmazza a betűk dimenzionális méretét – karakter szélességét – mellyel esztétikusan jeleníti meg a kiírt szöveget, gondolva itt arra, hogy például az „i” betű nem ugyanakkora helyet foglal, mint a „W”, ezzel lehetőség nyílik arra, hogy akár folyó kézírásos betűtípust is alkalmazhassunk.



Grafikus karaktertábla maszkja

Másik probléma, amit meg kell még oldani, hogy a betű maszkolása közben a háttér ne változzon, azaz átlátszó legyen. Ezt a problémát az alábbi módon sikerült kivitelezni:

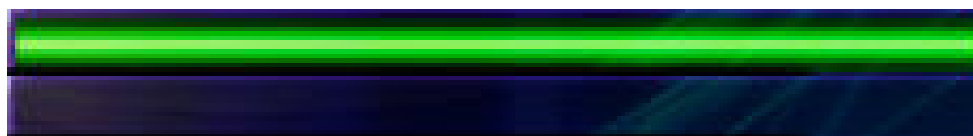


Háttér és betűmaszk rétegeinek mixelése

Az ábrán látszik, hogy több rétegben történik a megjelenítés. Az írási terület kijelölése után a háttér ugyanazon pozíciójától kezdünk kiolvasni, majd azok a képpontok kerülnek átmásolásra, ahol a maszk nem tartalmazza a karakter előtér színét. Amennyiben a betűmaszkban a karakter grafikájához ér a kiolvasás, onnantól már a beállított színnel történik a szöveg kiíratása.

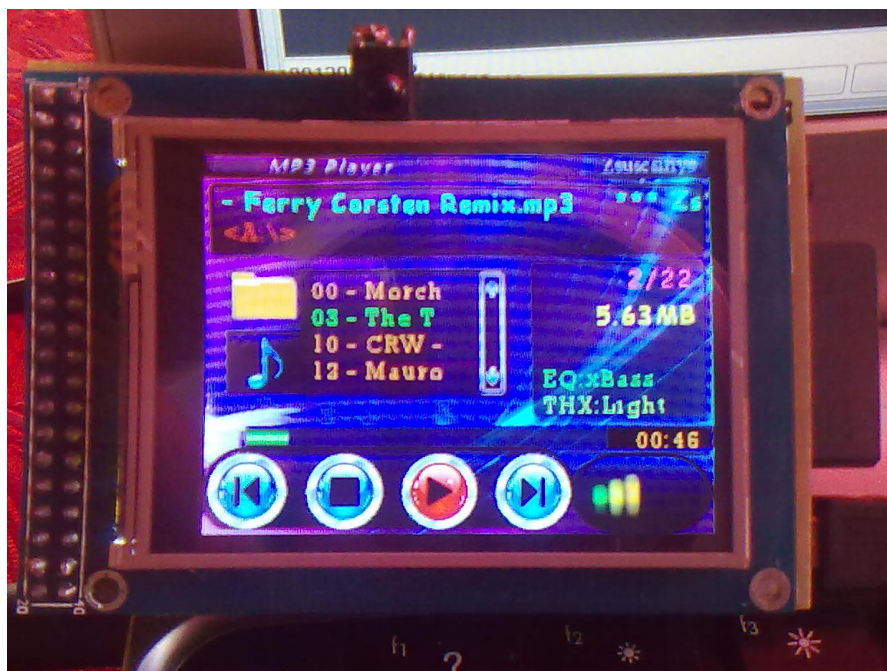
Folyamatjelzők generálása:

A kiválasztott képernyőterület lefoglalása után, beállítjuk a memória pointert az aktuális területre ahol a sáv grafika található, majd a százalékos számítás után soronként kezdi feltölteni a kijelzőt, majd mikor elérte a kívánt értéket, egy teljes képernyővel tovább ugrik a pointer ahol az üres sáv háttére található így a háttérrel kezdi el feltölteni. Ez a folyamat soronként ciklikusan ismétlődik, amíg ki nem rajzolódik a teljes ábra.



Folyamatjelző sáv ábrája

Az alábbi képen látható mindezek összességében elért megjelenés, vizuális hatás.



Futás közbeni vizuális megjelenés

Kiegészítő funkciók, az MP3 kodekben

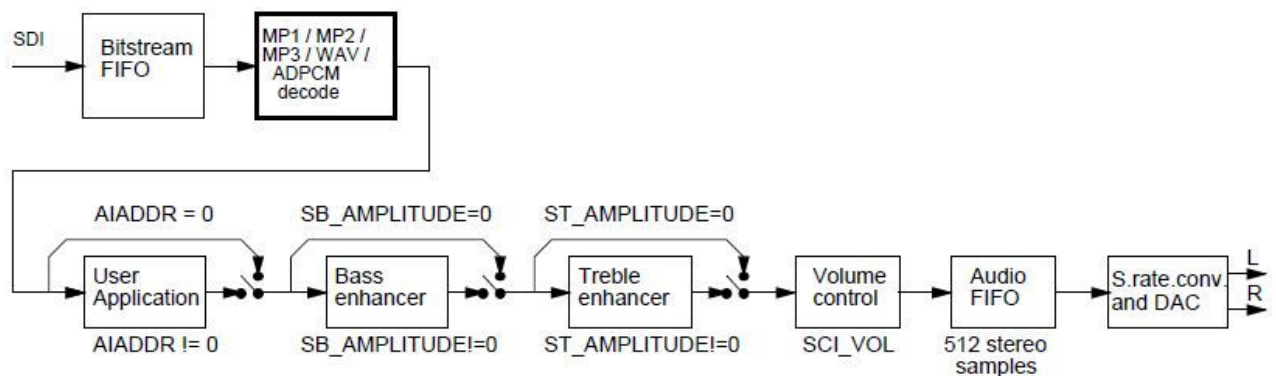
Alapvető zenelejátszási funkciója mellett kiemelt tulajdonságokkal is rendelkezik ez az IC, ilyen például az aktív digitális sávszűrő, mély- és magas hang kiemeléssel. A beállításhoz egy SCI_BASS nevű regisztert kell a táblázatban foglalt módon beállítani.

Name	Bits	Description
ST_AMPLITUDE	15:12	Treble Control in 1.5 dB steps (-8..7, 0 = off)
ST_FREQLIMIT	11:8	Lower limit frequency in 1000 Hz steps (0..15)
SB_AMPLITUDE	7:4	Bass Enhancement in 1 dB steps (0..15, 0 = off)
SB_FREQLIMIT	3:0	Lower limit frequency in 10 Hz steps (2..15)

MP3 kodek SCI_BASS regisztere

A regisztert két részre osztva, az alsó bájtban található meg az alsó vágási frekvencia 1 dB-es lépésenkénti erősítéssel, a felső bájtban viszont a magas vágási frekvencia, valamint a hozzá tartozó dB érték.

Az audio folyamat az alábbi ábrán lehet végigkövetni:



Látható, hogy a streamelés a FIFO pufferen keresztül történik, majd ezután a kodek dolgozza fel a jelet. Lehetőség van saját alkalmazás futtatására a kodeken belül, azonban ezzel nem foglalkozunk. A következő lépésben a mélyhang kiemelése történik, amennyiben a regiszter amplitúdó értéke 0-tól eltérő, majd ezt követően a magas hang szabályozása történik meg, ugyanazon feltétel mellett. Külön regiszter áll rendelkezésre a digitális hangerő szabályozásához, ami csatornánként eltérően is állítható, azaz balanszolható. Végül a kimeneti egységen a sztereo FIFO adattárolón keresztül digitális/analóg konverzió történik, ezzel előállítva a sztereo kimeneti hangot.

Felhasznált irodalom

AVR mikrovezérlő leírása:

Klaus Kühnel - AVR RISC Microcontroller Handbook (könyv)

SD kártya:

Dogan Ibrahim - SD Card Projects Using the PIC Microcontroller (könyv)

MP3 Kodek leírása:

VLSI honlap (<http://www.vlsi.fi/>)

VS1053b kodek adatlap

ILI9325 TFT 320x240xRGB kijelző leírás:

Vezérlő IC adatlap

Infra távirányító protokoll:

Internet (<http://wiki.altium.com/display/ADOH/NEC+Infrared+Transmission+Protocol>)

FAT32 fájlrendszer leírása és specifikációja:

Microsoft EFI FAT32 File System (internet)