Szegedi Tudományegyetem

Informatikai Tanszékcsoport

SZAKDOLGOZAT

Nagy Bence

2015

Szegedi Tudományegyetem

Informatikai Tanszékcsoport

**Átviteli függvény mérési- és HIL szimulációs funkciók megvalósítása oktatási célokra a MADAQ univerzális műszer segítségével**

**Development of modules for transfer function measurement and HIL simulation for educational purposes using the MADAQ universal device**

Szakdolgozat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Készítette: |  | Témavezető: |  |
|  | Nagy Bence |  | Dr. Mingesz Róbert |  |
|  | Mérnök informatikus BSc  szakos hallgató |  | egyetemi adjunktus |  |

Szeged

2015

## Feladatkiírás

A kutatócsoportunk a Mérés és Adatgyűjtés laboratóriumi gyakorlat számára elkészített egy egyedi, ugyanakkor univerzális mérőműszert. A műszer számos analóg bemenettel és kimenettel rendelkezik, e mellett számos digitális I/O vonallal is, melyeken különböző kommunikációs protokollok is megvalósíthatók. Jelenleg azonban sem a mikrovezérlőn futó szoftver, sem a LabVIEW driver messze nem használja ki a hardver képességeit.

A szakdolgozat célja mind a mikrovezérlőn futó szoftver, mind a LabVIEW driver bővítése, hogy ez által jelentősen növelni lehessen a műszer funkcionalitását. E mellett mintaalkalmazásokat is kell készíteni, melyek segítségével számos mérési feladat könnyen megoldható.

A hallgatónak a szakdolgozat keretén belül a következő funkciókat kell megvalósítania:

* Átviteli függvény mérése, lehetőség szerint minél nagyobb frekvenciáig.
* Hardware in the Loop (HIL) rendszer készítése, melyet fel lehet használni a mikrovezérlők kurzus oktatása során.

## Tartalmi összefoglaló

* A téma megnevezése:

Átviteli függvény mérési- és HIL szimulációs funkciók megvalósítása oktatási célokra a MADAQ univerzális műszer segítségével

* A megadott feladat megfogalmazása:

Áramkörök átviteli függvényének mérését lehetővé tevő MADAQ és LabView szoftver fejlesztése. Segéd szűrőáramkör tervezése és megépítése.

Egy ház fűtésének HIL szimulációjának megvalósítása, és egy ezen alapuló vizsgafeladat készítése a „Mikrovezérlők alkalmazástechnikája” gyakorlathoz.

* A megoldási mód:

Specifikáció, átviteli függvény mérési algoritmus megadása. Inkrementális fejlesztés: jelgenerátor funkció készítése, utána 1-, majd 2 csatornás mérés, utólag sine sweep implementálása és futási idő optimalizálása. Ezzel párhuzamosan tesztek, szimulációk futtatása a segédpanel elkészítéséhez. Ezt követően kapcsolási rajz, NYÁK terv készítése, alkatrészek kiválasztása, áramkör megépítése. Hardver- és szoftvertesztelés, hibakeresés, hibák javítása.

* Alkalmazott eszközök, módszerek:

Beágyazott szoftverfejlesztéshez a Silabs Config Wizard2, Silabs IDE és Notepad++ programokat használtam, PC-s szoftverfejlesztéshez LabView-t, áramkörök szimulációjához Tina és MATLAB programokat. Verziókövetéshez github-ot és SVN Tortoise programot. Az ábrák az Adobe Photoshop CS5, Dia, Snipping tool, MSPaint segítségével készültek. A MADAQ-hoz tartozó segédpanelt EAGLE-ben terveztem, és a NYÁK-ot a Mikropan Kft. [21] gyártotta le. Az összeszereléshez a Zaj labor eszközeit (multiméter, forrasztóállomás, fogó, csipesz, oszcilloszkóp, jel analizátor) használtam. Az alkatrészek a RET Kft-től [20] lettek beszerezve.

* Elért eredmények:

A MADAQ eszközt sikerült bővíteni a kitűzött funkciókkal. Elkészült a segédpanel, és sikerült lemérni néhány szűrőkör átviteli függvényét.

A hallgatók élesben megkapták a készített HIL-es feladatot, és levizsgáztak rajta.

* Kulcsszavak:

Átviteli függvény, mérés, szűrő, szoftverfejlesztés, hardverfejlesztés, prototípus, mikrovezérlő, LabView, HIL

## Tartalomjegyzék

[Feladatkiírás 2](#_Toc419579691)

[Tartalmi összefoglaló 3](#_Toc419579692)

[Tartalomjegyzék 4](#_Toc419579693)

[1. Bevezetés 6](#_Toc419579694)

[2. Irodalmi áttekintés 7](#_Toc419579695)

[2.1. AD konverzió 7](#_Toc419579696)

[2.2. Timer-ek 7](#_Toc419579697)

[2.3. Átviteli függvények 8](#_Toc419579698)

[2.4. Szűrőkörök 8](#_Toc419579699)

[2.5. Hardware in the loop 9](#_Toc419579700)

[2.6. Euler módszer 9](#_Toc419579701)

[3. Átviteli függvény mérésének megvalósítása 10](#_Toc419579702)

[3.1. Jelgenerátor működése 10](#_Toc419579703)

[3.1.1. A jelgenerátor működési tartománya 12](#_Toc419579704)

[3.2. Hardverfejlesztés, szűrő segédpanel 14](#_Toc419579705)

[3.2.1. Specifikáció 14](#_Toc419579706)

[3.2.2. Szimulációk, tesztek 14](#_Toc419579707)

[3.2.3. Kapcsolási rajz és NYÁK tervezése 16](#_Toc419579708)

[3.2.4. Eltérések a terv és a megvalósított panel között 18](#_Toc419579709)

[3.3. MADAQ szoftvere 18](#_Toc419579710)

[3.3.1. Bevezetés 18](#_Toc419579711)

[3.3.2. Jelgenerátor és kétcsatornás mérés 19](#_Toc419579712)

[3.3.3. Hardveres konfiguráció 21](#_Toc419579713)

[3.3.4. Kód refaktorálása 21](#_Toc419579714)

[3.4. LabView szoftver 21](#_Toc419579715)

[3.4.1. Mérés pontossága 23](#_Toc419579716)

[4. Ház fűtésének HIL szimulációja 24](#_Toc419579717)

[4.1 Specifikáció 24](#_Toc419579718)

[4.1.1. Felhasznált hardverek 24](#_Toc419579719)

[4.1.2. Feladat leírása 25](#_Toc419579720)

[4.2. Ház termikus modellje 26](#_Toc419579721)

[4.3. MADAQ szoftvere 28](#_Toc419579722)

[4.4. LabView szoftver 29](#_Toc419579723)

[5. Összefoglalás 31](#_Toc419579724)

[5.1. Eredmények 31](#_Toc419579725)

[5.2. Fejlesztési lehetőségek 31](#_Toc419579726)

[Irodalomjegyzék 32](#_Toc419579727)

[Felhasznált irodalom 32](#_Toc419579728)

[Cégek, vállalkozások 33](#_Toc419579729)

[Nyilatkozat 34](#_Toc419579730)

[Köszönetnyilvánítás 35](#_Toc419579731)

[Mellékletek 36](#_Toc419579732)

# 1. Bevezetés

A mérnökinformatikus, fizikus, vagy egyéb műszaki képzés alatt a hallgatók a labor gyakorlatok során sokféle műszerrel és hardveres eszközzel találkoznak. Ilyen például a Zaj labor által készített és fejlesztett MADAQ univerzális mérőműszer. Az eszközzel a hallgatók méréstechnikai ismereteiket bővítik: feladataik között szerepel elektronikus alkatrészek karakterisztikájának mérése, egyszerűbb szenzorok mérése, valamint műszer kalibráció. A fenti alkalmazások az eszköz megvalósítható funkcióinak csak egy kis részét teszik ki. A MADAQ kihasználtsága messze van a fizikai korlátaitól.

A szakdolgozatom első részében a MADAQ-hoz készített átviteli függvény méréshez készített funkciókat mutatom be. Az átviteli függvény egy univerzális és kifinomult módja rendszerek leírásának, amely a bemenet és a kimenet közti kapcsolatot adja meg. A szakdolgozatomban a hangsúly a mérés folyamatán, és a mért adatok feldolgozásán van.

A rendszerelméletben és irányítástechnikában az átviteli függvény fogalma alapvetőnek számít. Az itt lefektetett elméleti tudás kb. az 1800-as évek óta fejlődik, melyet a mérnöki tudományok számos területén használnak. A számítógépek számítási kapacitásának növekedésével egyre hatékonyabb szabályzórendszerek készülhetnek, ahogy az irányítástechnika és az informatika egy úttörő interdiszciplináris mérnöki és ággá fonódik össze.

A szakdolgozatom második felében egy HIL szimulációs rendszer készítéséről fogok írni. A hardware in the loop egy szimulációs technika, amelyet szabályzórendszerek fejlesztéséhez és teszteléséhez használnak. A lényeg, hogy egy vezérelt rendszert (pl. mechanikai rendszer, autó ABS rendszere, emelődaru, motorgyújtás vezérlés, stb.) lecseréljük egy őt utánzó rendszerre, és azon teszteljük a szabályzást. Ezzel a konstrukcióval olyan esetek is előidézhetők, amelyek emberekre veszélyesek lennének vagy a valódi rendszert tönkretennék.

A fent említett technológiával egy vizsgafeladatot kell készíteni hallgatók számára a *Mikrovezérlők alkalmazástechnikája* nevű kurzus laboratóriumi gyakorlatához. Egy a PC-n szimulált ház fűtésének szabályzását kell megoldják a laboron használt fejlesztőkörnyezet segítségével. Újdonság, hogy LED-ek, hétszegmenses kijelző helyett PC-s felhasználói felületen láthatják programjuk futásának eredményét, valamint nyomógombok helyett vezérlőjeleket használnak. Remélhetőleg a hallgatók számára egy valósabb környezetet teremt a feladat.

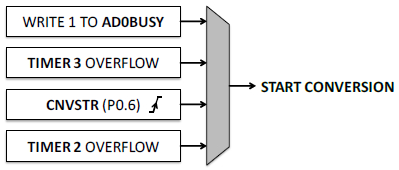
# 2. Irodalmi áttekintés

## 2.1. AD konverzió

Az analóg digitál konverter egy mikrovezérlőben megtalálható periféria, amely analóg feszültséget alakít egy *b*-bites egész számmá, ahol egy fix, stabil referenciafeszültség. A bemeneti feszültség a tartományban mozoghat. [1]

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

Időben változó jelek mérése, mintavételezése AD konverziók sorozatából áll. A feladatomhoz fontos, hogy a mérések szabályos időközönként történjenek, stabil mintavételi frekvenciára van szükség. A 8051F410 mikrovezérlőn az alábbi módon lehet AD konverziót indítani. [1]



**2.1.** ábra – AD konverzió indításának módjai

## 2.2. Timer-ek

A Timer, vagy időzítő egy szintén mikrovezérlőben megtalálható periféria, egy belső oszcillátorral vagy külső jellel léptetett bináris számláló. Események számlálására vagy generálására (pl. interrupt) használható. [1]A programomban a mintavételezés Timer-en alapul. A Timer overflow rate az alábbi képlettel adható meg:

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |

ahol a Timer-t léptető órajel frekvencia, a pedig a számláló kezdőértéke egy overflow-t követően. Egy számláló általában 8 vagy 16 bit-es, tehát értéke 8 vagy 16.

## 2.3. Átviteli függvények

Az átviteli függvény egy kompakt és univerzális reprezentációja lineáris időinvariáns (LTI) rendszereknek. Matematikailag nézve ez egy komplex változós függvény, amely megadja a bemenet-kimenet közötti kapcsolatot. Az átviteli függvény megkapható a rendszer megfigyelése alapján, vagy az őt leíró differenciálegyenletek (ODE*, ordinary differential equation*) manipulálásával. [2]

Egy SISO (single iput, single output) LTI rendszer bemenet-kimenet modellje az alábbi módon adható meg:

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

ahol a rendszer bemenete és a kimenete és konstans együtthatók.

Az átviteli függvény a bemenet-kimenet modellben szereplő kimenő jel Laplace transzformáltja és a bemenő jel Laplace transzformáltjának hányadosa, zérus kezdeti feltételek mellett.

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

ahol a Laplace operátor. [3][4]

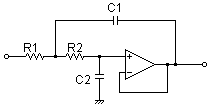
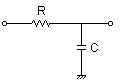
Egy rendszerről látványos információt ad az erősítés és fáziseltolás frekvencia-karakterisztikája, amely a Bode diagramon ábrázolható, és az alábbi alakban adható meg.

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |
|  |  | (2.6) |

ahol a körfrekvencia. [5][6]

## 2.4. Szűrőkörök

A szűrő egy jelek szűrésére használható eszköz vagy eljárás, amely nem kívánt komponenseket távolít el egy jelből. Elektronikában ellenállás, kondenzátor, tekercs, műveleti erősítő segítségével építhetünk szűrőket Egy ilyen áramkör adott frekvenciájú komponenseket csillapít, vagy enged át a kimenetén. Ahogyan a jel fogalma nem csak elektronikus jelekre korlátozódik, így a szűrők sem feltétlen csak elektronikus jelek szűrésére használhatók. [7]



**NULL** ábra – RC és Sallen-key LP szűrő kapcsolási rajza

## 2.5. Hardware in the loop

A *hardware in the loop* egy szimulációs technika, amelyet szabályzórendszerek (*control system*) fejlesztéséhez és teszteléséhez használnak. Az elv, hogy a vezérelt rendszert (*plant*) lecseréljük egy őt utánzó rendszerre. A szabályzórendszer számára ez a csere nem észlelhető. Ezzel a konstrukcióval olyan esetek is előidézhetők, amelyek emberekre veszélyesek lennének vagy a valódi rendszert tönkretennék. [8]

## 2.6. Euler módszer

Az Euler módszer egy numerikus, iteratív eljárás differenciálegyenletek (ODE) megoldásához. Az iteráció egy ismert kezdőértékről indul. A következő érték kiszámításához szükség van az előző értékre, egy lépésközre, valamint magára a differenciálegyenletre. Formális integrálásra azonban nincs szükség, illetve a módszer nem ad analitikus megoldást. Legyen adott egy

|  |  |  |
| --- | --- | --- |
|  |  | (2.7) |

alakú differenciálegyenlet. Ekkor az egyenlet közelítő, numerikus megoldása a

|  |  |  |
| --- | --- | --- |
|  |  | (2.8) |

képletek segítségével számolható ki. [9][10]

# 3. Átviteli függvény mérésének megvalósítása

*„A valóságnak minden térben elhatárolt részét, ahol a különböző anyag- és mozgásformák elemeit kölcsönhatások és kölcsönös összefüggések kapcsolják össze, rendszernek nevezzük.”* [4]

Az átviteli függvény egy univerzális és kifinomult módja rendszerek leírásának. A szakdolgozatomban elektronikus rendszerekkel foglalkozom. A feladat hallgatók számára lehetővé tenni, hogy laboron összeállított áramkörök erősítés és fázis frekvencia-karakterisztikáját vizsgálni és ellenőrizni tudják a MADAQ műszer segítségével. Az átviteli függvény mérés egy gyorsan végrehajtható mérés, amely már ránézés alapján információt ad egy rendszerről. A szakdolgozatomban a hangsúly a mérés folyamatán, és a mért adatok feldolgozásán van.

Fontos megjegyezni, hogy a frekvencia-karakterisztika (*frequency response*) és az átviteli függvény (*transfer function*) nagyon hasonló fogalmak. A különbség, hogy az előbbi Fourier, az utóbbi Laplace tartományban értelmezett. A két tartomány között egy egyszerű helyettesítéssel lehet váltani. A feladat szempontjából a mért átviteli függvény analitikus alakjának meghatározása nem lényeges.

## 3.1. Jelgenerátor működése

*A 3.1.* fejezetben bemutatom milyen megfontolások alapján generálok jeleket.

A szakdolgozatomban a jelgenerálás és a kétcsatornás mérés együtt használandó fogalmak. Átviteli függvény mérésekor a MADAQ egyik csatornája a gerjesztő jelet méri, a másik a vizsgált rendszer kimenetét. A jel egy mintákból álló tömböt jelent. Szakdolgozatomon belül a mérésekhez elegendő volt csak szinusz jelet használni. A jel frekvenciája egy tartományon belül állítható. *(A tömb tetszőleges értékekkel tölthető fel az adott korlátokon belül)*.

A működési elv a következő. A jelgenerálás memóriában tárolt jel-minták alapján történik DA konverter segítségével. Egy időzítő (Timer) adott időközönként a következő mintára vált. Lényegében ilyenkor ez egy gyorsan változó DC jel, amelyet kellően megszűrve analóg AC jelet kaphatunk. Szakdolgozatomban 1 periódusnyi szinusz jel mintáit használtam gerjesztő jelnek.

Bevezetek néhány képletet, változót és fogalmat, hogy paraméterezhetően tudjunk jeleket generálni.

* mintavételi frekvencia, TMR2 interrupt frekvencia -ben, *(sampling frequency)*
* gerjesztő jel frekvenciája -ben, *(signal frequency)*
* kívánt gerjesztő jel frekvencia -ben, *(desired signal frequency)*
* : Timer reload érték, 2 byte-os előjeltelen szám
* MADAQ órajel frekvenciája-ben Timer clock source
* jelet alkotó minták száma, vízszintes felbontás (nem a DAC felbontását jelenti)
* periódusok száma (tömbben található jel periódusainak száma)

A mintavételi frekvencia (3.1) a MADAQ órajelétől és a Timer reload értéktől függ [1]

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

A gerjesztő jel frekvenciája (3.2) a TMR2 interrupt frekvenciától a használt periódusoktól és a minták számától függ:

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

*Megjegyzés:*

Egy frekvenciájú jel többféleképpen is előállítható. Pl. .

Az (3.1) és (3.2) képletet felhasználva adott frekvenciájú jelhez tartozó Timer reload érték (3.3) meghatározása a minták és periódusok számának függvényében:

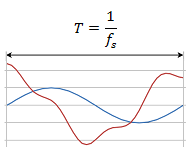
|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |

Ha pl. a tömb 3 periódusnyi szinusz jel mintáit tartalmazza, akkor . Ezt a képletben figyelembe kell venni, különben nem a kívánt frekvenciát kapjuk.

Az (3.1)(3.2)(3.3) képletek segítségével beállíthatjuk a értékét úgy, hogy egy adott frekvenciájú jelet tudjunk generálni, valamint hogy visszaszámoljuk a értékből, hogy milyen frekvenciát lehet a ténylegesen előállítani.

*Megjegyzés:*

Az általam értelmezett gerjesztő jel egy általános mintákból álló tömböt jelent. A a periódusidő, a tömb egyszeri legenerálásához szükséges időt jelenti. Az **3.1.** ábra ezt próbálja szemléltetni. Az ábrán két találomra választott jel látható, az alakjuk lényegtelen.



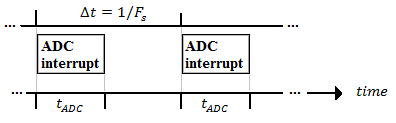
**3.1.** ábra – A periódusidő alternatív értelmezése

### 3.1.1. A jelgenerátor működési tartománya

*Egy kérdés:* mi a maximális mintavételi frekvencia és legfeljebb hány mintából állhat egy jel, hogy ne lépjük túl a hardver korlátait? Szükség lesz még néhány változóra:

* maximum használható minták száma
* legnagyobb stabil mintavételi frekvencia
* ADC interrupt duty cycle, lásd. (3.4) képlet
* ADC interrupt rutin konstans futási ideje (kódtól függ),

A MADAQ eleinte túlterheltség miatt lefagyott, és nem a kívánt jelet generálta. Ahogy a érték a -hez közelít, a jelgenerátor működése egyre megbízhatatlanabb. Adott esetben előfordulhat, hogy a következő megszakítás még az előző befejeződése előtt érkezik. A problémára meg kellett adjak egy korlátot a jelgenerátor számára.



**3.2.** ábra – Az ADC interrupt rutin idődiagramja

Tekintsük a **3.2.** ábrát. A hardveres AD konverzió idejét most nem veszem figyelembe. Ha ismerjük a értékét, akkor beállíthatjuk az mintavételezési frekvenciát úgy, hogy *elegendő* szabad processzoridő legyen az ADC interrupt-ok között. A idő oszcilloszkóp segítségével lett lemérve. Adjunk meg egy arányszámot (3.4), a **3.2.** ábrához:

|  |  |  |
| --- | --- | --- |
|  |  | (3.4) |

Ha, akkor nincs szabad processzoridő a megszakítások között. Ha , akkor pont -nyi szabad processzoridő van. Használjuk fel ezt az arányszámot, hogy a mintavételi frekvenciára, és a jelet alkotó minták számára felső korlátot adhassunk. Legyen egy paraméter, és értéke mondjuk , hogy a kb. processzoridő maradjon egyéb feladatokra. A egy empirikus érték, melyet tesztelgetés alapján határoztam meg, de szükség esetén változtatható. A (3.4) és (3.2) képlet alapján:

|  |  |  |
| --- | --- | --- |
|  |  | (3.5) |
|  |  | (3.6) |

A két képletbe behelyettesítve azt kapjuk, hogy , valamint . Tehát mintavételi frekvencia felett nem mondhatjuk megbízhatónak a rendszert. A minták számát tekintve érdemes legalább 8 pontot használni. Ebben az esetben a használható legnagyobb frekvenciájú jel . A tartományt természetesen túl lehet lépni, de kb. felett több fizikai korlát is akadályozza a megfelelő jelgenerálást, mint például a szűrő nagy csillapítása, vagy a minták alacsony száma, vagy az (3.1) képlet nemlinearitása.

Egyesítsük a (3.3) és (.36) képletet, helyettesítsünk helyére -ot:

|  |  |  |
| --- | --- | --- |
|  |  | (3.7) |

A (3.7) képlet segítségével előállíthatunk frekvenciájú jelet úgy, hogy nem terheljük túl. vagy fagyasztjuk le a hardvert a lehető legtöbb mintát és a lehető legnagyobb mintavételi frekvenciát használva.

## 3.2. Hardverfejlesztés, szűrő segédpanel

### 3.2.1. Specifikáció

A szűrő feladata a MADAQ által kiadott lépcsős gerjesztő jel simítása. Mivel a DAC véges felbontással dolgozik, ezért a kiadott jel sosem lesz teljesen sima.

A szűréshez 2 db másodfokú aktív Sallen-key aluláteresztő szűrőt használok. Az egyik szűrő a tartományban használandó, a másik az ennél magasabb frekvenciájú jelekhez -tól -ig. Az ennél magasabb tartományban a gerjesztő jel pontossága már nem garantált. A két szűrt jel között egy MADAQ-ról vezérelhető analóg kapcsolóval lehet váltani. Erre azért van szükség, mert csak így sikerült a kívánt frekvencia tartományt átfogni. Egy szűrő esetén vagy túl kicsi volt a szűrt jel amplitúdója a nagy csillapítás miatt, vagy lépcsős maradt.

A panelhez BNC csatlakozóval vagy normál anya vezetékkel lehet csatlakozni. A BNC csatlakozó előnye, hogy közvetlenül oszcilloszkópra köthető és kevésbé érzékeny a zajra. Az összes lényeges jel pin-ekre lett kivezetve. A panelen jumper-eket helyeztem el, amelyekkel néhány alternatív összekötés lehetséges.

### 3.2.2. Szimulációk, tesztek

Az áramkör tervezése előtt MATLAB-ban megvizsgáltam néhány szűrőt lépcsős szinusz gerjesztő jel esetén. A választás ekkor esett a Sallen-key szűrőre. Főként arra voltam kíváncsi, hogy mennyire hatékonyan és mekkora frekvenciatartományban képes a jelet simítani.

MATLAB-ban könnyen vizsgálható a LTI rendszerek működése, és tetszőleges gerjesztő jeleket is egyszerű készíteni. A készített MATLAB script-ben a Sallen-key szűrő átviteli függvényét (3.8) [11] használtam, amelyet dekád tartományban vizsgáltam a **3.3.** kódrészletben látható egyedi gerjesztő jellel. A gerjesztő jel néhány valós paramétertől eltekintve megegyezik a MADAQ-on kiadott jellel.

|  |  |  |
| --- | --- | --- |
|  |  | (3.8) |

res = floor(0.8/(4.6\*micro\*fs)); % optimal signal resolution for MADAQ

if (res > smax) res = smax; end

t = 0: (1/Fs) :(1/fs\*n); % n period, sampling rate = 100\*fs

u = 0.5 \* sin(2\*pi \* fs .\* t) + 0.5; % sine wave in [0,1]

u = (A\*2) \* round(u\*res)/res - A; % quantize signal + offset to [-A, A]

**3.3.** kódrészlet – Lépcsőzetes szinusz jel előállítása szimulációhoz

fc = 28480; % cutoff freq in Hz

[R1, C1, R2, C2] = calculate\_filter\_values(fc);

s = zpk('s');

L = (1/(R1\*C1\*R2\*C2)) / (s^2 + s\*( 1/(R2\*C1)+1/(R1\*C1)) + 1/(R1\*C1\*R2\*C2));

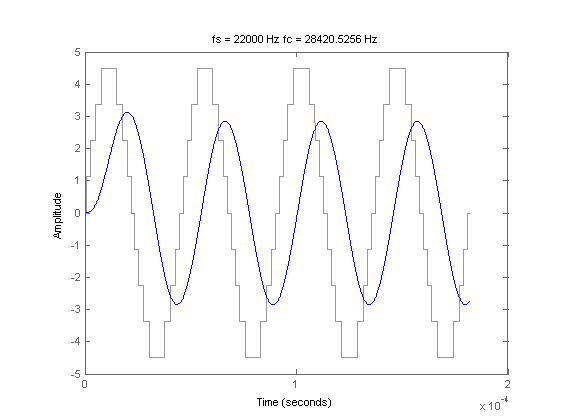
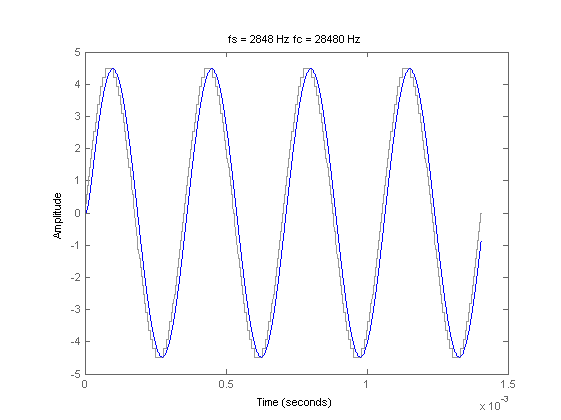
[u, t] = generate\_digital\_sine(22000, 32); % signal freq, max samples

lsim(L, u, t);

**3.4.** kódrészlet – Sallen-key szűrő tesztelése-es jellel

Először a Tina programban akartam szimulálni, de körülményes volt tetszőleges jelalakot készíteni, amelyet paraméterezni is nehezen lehetett. A **3.3.** kódrészlet a MADAQ-ban implementált jelgenerátort hivatott utánozni. A MADAQ fizikai korlátai miatt a jelet alkotó pontok száma egy kritikus adat. Ahogy a gerjesztőjel frekvenciája növekszik, egyre kevesebb pontból lehet szinusz jelet előállítani. Sok pontból álló jelet könnyebb simítani. Sőt, néhány frekvenciánál páratlan számú minták használata szabálytalan jelet eredményezett.

MATLAB-ban a Signal Processing Toolbox segítségével egyszerűen lehet LTI rendszereket szimulálni. Az függvény tetszőleges jelre meghatározza a rendszer válaszát. A **3.4.** kódrészleten ez látható.



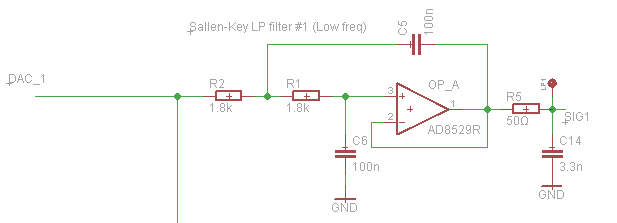
**3.5.** ábra – Szűrt és szűretlen szinusz jel.

Figyeljük meg a **3.5.** ábrán, hogy -nél sima lett a jel, és a szűrő csillapítása sem számottevő, de alatt már látszódtak a lépcsők. Frekvenciában felfelé haladva az amplitúdó -nél már a felére csökken. Kisebb amplitúdójú jel nagyobb hibát eredményez méréseknél az ADC véges felbontása miatt. A tesztek alapján tehát a Sallen-key szűrőt a tartományban érdemes használni a feladathoz.

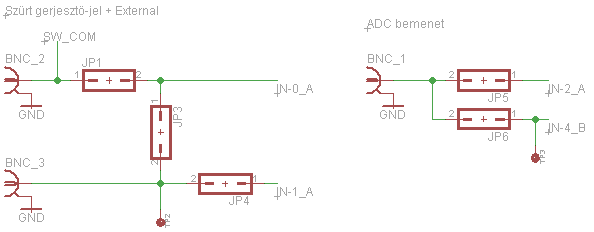
### 3.2.3. Kapcsolási rajz és NYÁK tervezése

A lépcsős gerjesztő jelet 2 db Sallen-key szűrő szűri. A két szűrő paramétereit úgy állítottam be, hogy a tartomány-béli jeleket engedjen át. A vágási frekvenciák Az alkatrészek értékeit a [11] programmal számoltam. A két szűrt jel egy analóg kapcsolóba megy, amely egy vezérlőjelet használva az egyik, vagy másik jelet engedi tovább.

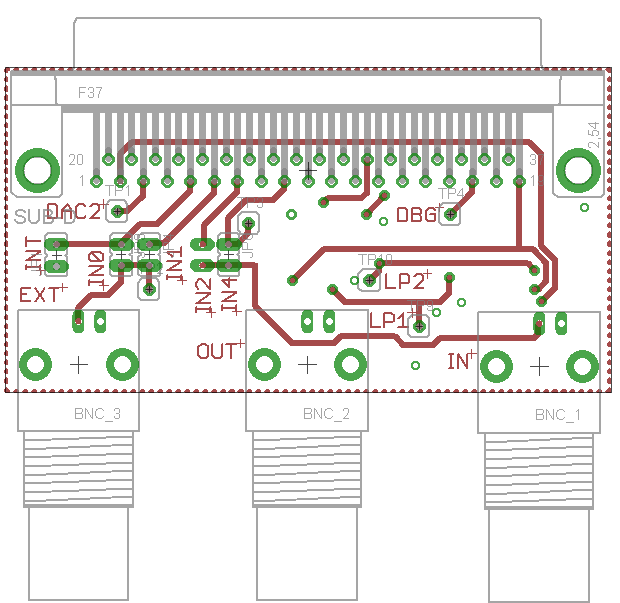
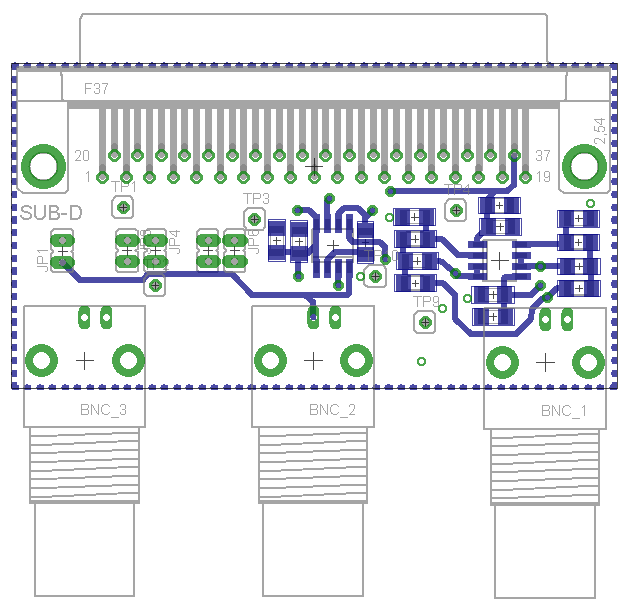
A panelre elhelyeztem néhány jumper-t. Ezekkel megoldható, hogy külső gerjesztő jelet használjunk (MADAQ helyett), valamint megadható, hogy melyik jel melyik ADC-re menjen.



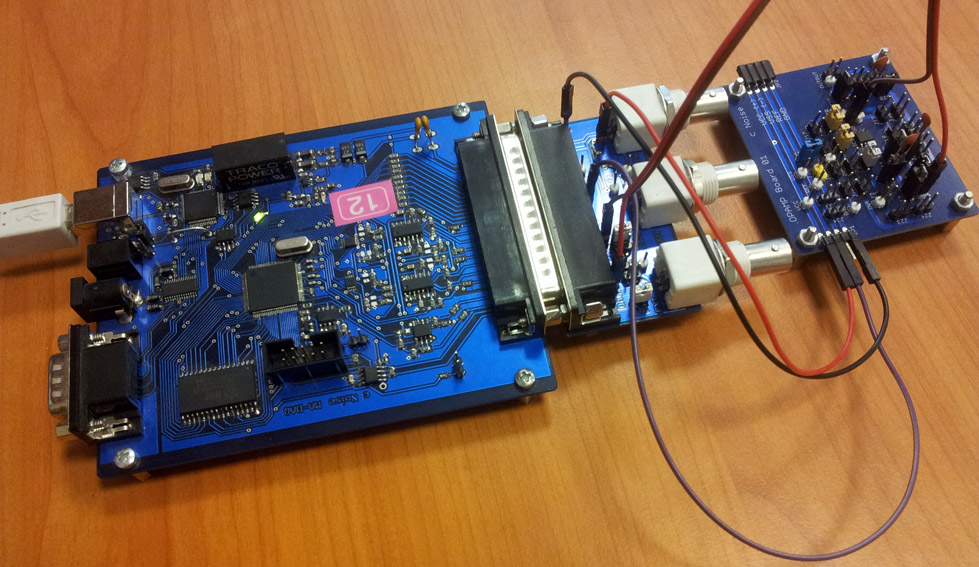
**3.6.** ábra – Sallen-key szűrő



**3.7.** ábra – Jumper opciók



**3.8.** ábra – A szűrő segédpanel áramköri tervének alsó és felső rétege

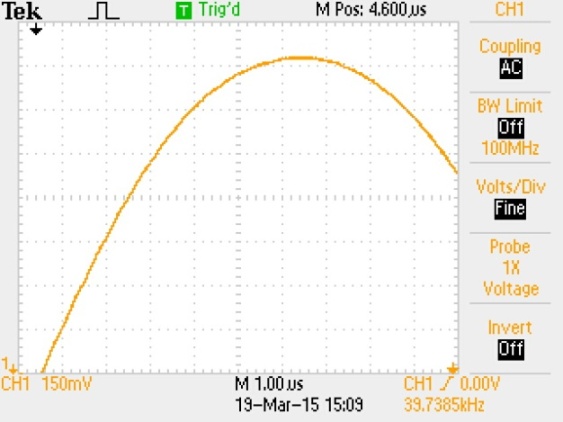
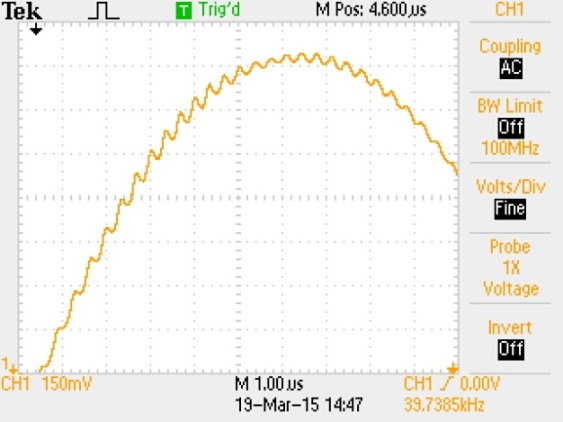


**3.9.** ábra – A MADAQ mérés közben

A **3.9.** ábrán a megépített szűrőpanel látszódik, ahogy a MADAQ-hoz kapcsolódva épp egy mérést végez. A mérendő áramkört témavezetőm által készített panelen állítottam össze.

### 3.2.4. Eltérések a terv és a megvalósított panel között

A tesztelés során 40 kHz-es jelnél nem várt, kb. 1-2 MHz-es gerjedés volt megfigyelhető, e frekvencia alatt nem jelentkezett probléma. Szerencsére a NYÁK-on könnyen lehet pad-eket „vakarni”. Témavezetőm javaslatára a szűrők kimenetére tettem sorosan egy-egy -os ellenállást, de nem segített. Ezt átalakítottam egy kb. 800 kHz-nél vágó RC szűrővé egy -os kondenzátorral. Az eredmény a **3.10.** ábrán látható. A javítás már szerepel a mellékletben található áramköri terven.



**3.10.** ábra – Jelgenerátor gerjedése 40 MHz-nél

## 3.3. MADAQ szoftvere

A *3.3.* fejezetben a MADAQ-on futó, kétcsatornás mérés és jelgenerálás funkcióhoz tartozó kódról és hardveres beállításokról lesz szó.

A MADAQ szoftverének feladata egy PC-ről érkező tetszőleges tömb fogadása, amely a adott frekvenciájú jel generálásához szükséges mintákat tartalmaz. A PC-től konfigurációs beállításokat is fogad, pl. mintavételi frekvencia beállítása, tömb méretének beállítása, szűrő segédpanel vezérlése. A jelgenerálással együtt a MADAQ két csatornán mérést végez, melyet csatornánként egy max. 128 elemű tömbben tárol és PC-re küld.

### 3.3.1. Bevezetés

A MADAQ firmware-ének feladata PC-vel kommunikálni, az onnan érkező utasításoknak megfelelően cselekedni. Az eredeti kódban található utasítások többnyire perifériákkal kapcsolatos konfigurációra szolgálnak, vagy DC mérést végeznek. DC jelek vizsgálatához elég egyszeri méréseket végezni, a mintavételi frekvencia nem kritikus kérdés. Az újdonság a kétcsatornás szinkronmérés, mellyel időben változó AC jelek vizsgálhatók, és amely pontos hardveres időzítést, bonyolultabb konfigurációt relatívan nézve nagyobb memóriát és optimális futási időt kíván. A feladat a fentebb említett módszerrel szűrőkörök erősítés és fáziseltolás frekvencia-karakterisztikáinak mérése.

### 3.3.2. Jelgenerátor és kétcsatornás mérés

Az átviteli függvénnyel kapcsolatos feladatok leginkább a LabView szoftveren belül találhatók, nem a MADAQ-on. A szoftver úgy lett megtervezve, hogy a számítások lehető legtöbb része PC-n történjen, és a MADAQ-nak minél egyszerűbb mérési és konfigurációs feladatokat kelljen végrehajtani. Ez az elv sebesség szempontjából előnyös. Az órajel ugyan 24 MHz, de a kétcsatornás mérési- és jelgenerátor funkcióval túl hamar sikerült a rendszer határát elérni.

A MADAQ utasításai:

* : jelgeneráláshoz használandó minták beolvasása
  + tömb fogadása PC-ről
  + *uint16* típusú adat
  + max. 128 elem
* : TMR2 reload érték beállítása
  + *uint16* típusú adat
  + a mintavételi frekvencia ettől az értéktől függ
* : szűrő segédpanel közvetlen vezérlése
  + analóg kapcsoló beállítása (jel frekvenciájától függően)
  + *bit* típusú adat
* : jelgenerálás és mérés indítása
  + DAC, TMR2, ADC1, ADC2 bekapcsolása
  + feltételezzük, hogy minden megfelelően be lett állítva
* : mérés leállítása és mért adatok küldése PC-nek
  + DAC, TMR2, ADC1, ADC2 bekapcsolása
  + DAC beállítása -ra

A mikrovezérlőknek általában kicsi a gyorsan elérhető memóriájuk. A jelgeneráláshoz használt minták nem férnek a mikrovezérlő belső memóriájában, ezért az External Memory Interface-t használom. A memóriában a minták egymás után vannak tárolva High és Low byte szerint külön. A memóriát a **3.11.** ábrán látható módon használom.

#define SAMPLES(addr) (\*((unsigned char \_\_xdata \*)(addr)))

#define INPUT\_MEASURE(addr) (\*((unsigned char \_\_xdata \*)(256 + addr)))

#define OUTPUT\_MEASURE(addr) (\*((unsigned char \_\_xdata \*)(512 + addr)))

**3.11.** kódrészlet – Minták tárolása az XRAM-ban

A program készítése során az egyik legnehezebb feladat a kellően gyors ADC interrupt függvény megírása volt. Eleinte szerettem volna minél átláthatóbb kódot írni, csak C utasításokat használva. Sajnos nem végeztem pontos összehasonlításokat, de néhány régebbi kódrészlethez képest a mostani azok töredéke végrehajtási időt és kód hosszúságot tekintve. Az első működő ADC interrupt rutin kb. 90 Assembly utasításból állt. Miután néhányszor újragondoltam és újraírtam a kódot, sikerült kb. 35 Assembly utasítást felhasználva megírni az ADC megszakítás rutint. Hogy ez miért olyan fontos, azt a *3.1.* fejezetben fejtettem ki. A kódrészlet a **3.12.** ábrán látható. Ez a függvény tárolja el az ADC1 és ADC2 által mért értékeket, és tárolja el, valamint a rutin végén a következő tömb-béli minta értékére állítja be a DAC-t.

// ------ CH #2 measurement ------

\_\_asm

// SFRPAGE = ADC0\_PAGE;

// OUTPUT\_MEASURE(j) = ADC0H; j++;

// OUTPUT\_MEASURE(j) = ADC0L; j++;

mov \_SFRPAGE**,**#0x00

mov dpl**,**\_n

mov dph**,**#0x02 // dptr: from 0x0200 to 0x02FF

mov a**,**\_ADC0H

movx @dptr**,**a

inc dpl

mov a**,**\_ADC0L

movx @dptr**,**a

\_\_endasm**;**

// ------ signal generation from samples ------

\_\_asm

// DAC0H = SAMPLES(n); n++;

// DAC0L = SAMPLES(n); n++;

mov \_SFRPAGE**,**#0x01 // DAC1\_PAGE

mov dpl**,**\_n

mov dph**,**#0x00

movx a**,**@dptr

mov \_DAC1H**,**a

inc dpl

movx a**,**@dptr

mov \_DAC1L**,**a

inc \_n

inc \_n

\_\_endasm**;**

**3.12.** kódrészlet – ADC megszakítás; 1 csatorna mérése, tárolása és jelgenerálás mintánként

### 3.3.3. Hardveres konfiguráció

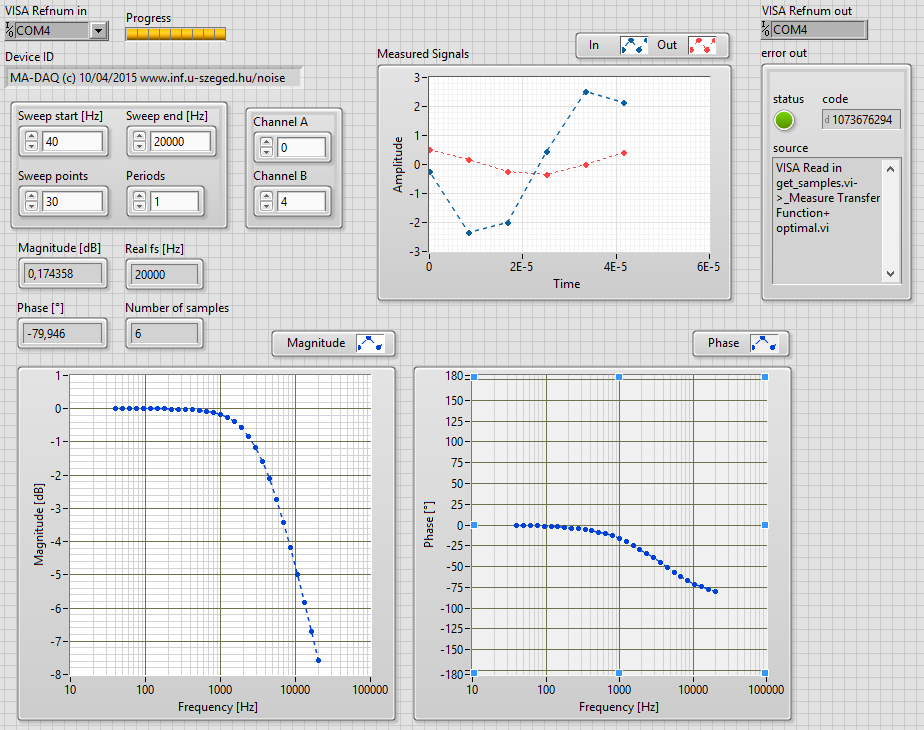
A MADAQ egy univerzális mérőműszer, ami azt jelenti, hogy több feladatra is használható. Alapvető cél volt, hogy az eredeti funkcionalitás megmaradjon. A programozás során törekedtem arra, hogy ez ne változzon. Tipikusan a perifériák konfigurációja okozhat problémát különböző feladatok esetében. Az elvem az volt, hogy egy feladat befejezése után (pl. átviteli függvény mérés) minden az indulási állapotra álljon vissza. Az eszköz indulásakor az ADC és TMR perifériáknak mindenképp kikapcsolva kell lenniük, különben a jelgenerátor értelmetlen értékekkel kezd dolgozni.

### 3.3.4. Kód refaktorálása

A MADAQ eredeti kódja egy forrás fájlba volt másolva, illetve a Config Wizard által generált fájllal néhány utólagos módosítás miatt nem volt szinkronban. A kódban túl gyakran elvesztem, és nehezen láttam át. Illetve a kódban volt néhány nem használt függvény, utasítás, változó. Készítettem egy működésében azonos forráskódot, amelyben a fenti problémák kijavításra kerültek.

## 3.4. LabView szoftver

A *3.4.* fejezetben a PC-on futó programról lesz szó, amely a MADAQ-ról érkező vagy az afelé menő adatokat kezeli, valamint a jelgenerálást és mérést vezérli.



**3.13.** ábra – A LabView program előlapja.

A fejlesztés során törekedtem arra, hogy a különböző feladatok minél jobban elkülönüljenek. Ezt LabView-ban SubVI-ok segítségével lehet megoldani. Az alábbi részben bemutatom, milyen SubVI-okat használtam, és azok feladatát, rövid jellemzését. A bemutatás sorrendje megegyezik a végrehajtás sorrendjével is.

Inicializálás:

* *DeviceInfo.vi*
  + A MADAQ eszközinformációját kéri le
  + Csak egy indikátor, hogy működik-e a kommunikáció
* *ConfigADC.vi*
  + A MADAQ-on található multiplexer beállítása
  + ezzel adható meg, hogy az ADC-k melyik vezetékhez legyen rendelve

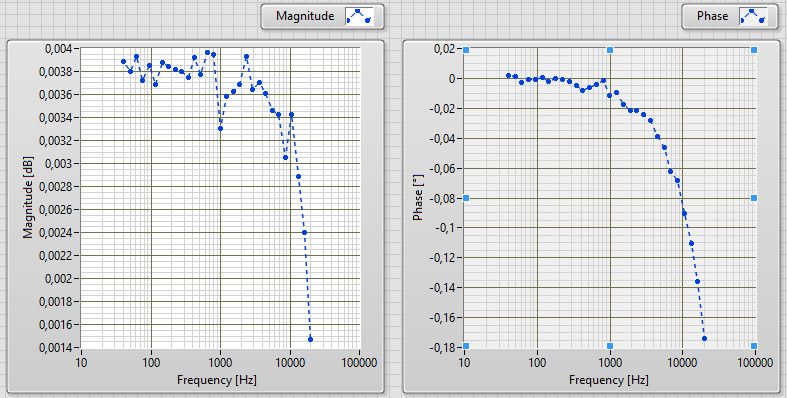
Átviteli függvény mérése egy ciklusban történik. A mérés előtt megadhatjuk, hogy hány frekvencián mérjen a program, és azt milyen tartományban tegye. A beosztás logaritmikus. A ciklusban futó függvények feladataik az alábbi listában láthatóak.

* *SendSamples.vi*
  + Elkészíti a jelgeneráláshoz használt mintákat
  + A gerjesztő jel szinusz (*más jelre nem volt szükség*)
  + A MADAQ eszköznek leküldi a mintákat
  + Az elemek számát *3.1.* fejezetben levő képletek alapján számolja
  + Csak akkor küld mintákat a MADAQ-nak, ha az előző iterációhoz képest azok megváltoztak
* *SendFreq.vi*
  + Beállítja a mintavételi frekvenciát az aktuális iterációban levő értékre
  + A mintavételi frekvencia összefügg a gerjesztőjel frekvenciájával
  + A számítást a *3.1.* fejezet alapján végzi
* *SwitchFilter.vi*
  + Az aktuális jel frekvenciájától függően beállítja, hogy melyik szűrő legyen aktív a segédpanelen
* *GenerateSignal.vi*
  + Elindít egy adott frekvenciájoz tartozó mérést a MADAQ-on
  + A következő utasítás előtt várakozik néhány jel-periódusnyi időt
* *GetSamples.vi*
  + Leállítja a mérést, lekéri a 2 csatorna jelét
* *CalulcateMagPhase.vi*
  + Kiszámolja a tényleges mintavételi frekvenciát a beállított Timer reload értékből
  + A beolvasott a 2 csatorna alapján magnitúdót és fázist számol
  + A mért jeleket kijelzi egy grafikonon
  + A számított értékeket továbbküldi a Bode grafikonra

### 3.4.1. Mérés pontossága

A kíváncsiság kedvéért a gerjesztő jelet közvetlenül BNC csatlakozóval a MADAQ bemenetére kötöttem. Ez a rendszer elméletben konstans -es magnitúdóval és fázistolással rendelkezik. Ha a vezetékből adódó hibáktól eltekintünk, akkor magát a mérés pontosságát figyelhetjük meg. A -hez képest a legrosszabb érték amit kaptam, valamint a legpontatlanabb fázistolás érték helyett . Az eredmény a **3.14.** ábrán látható. Figyeljük meg a függőleges tengelyek beosztását.

Ezen kívül végeztem egy más féle mérést, amellyel a műszer pontosságát a felhasznált minták száma alapján vizsgáltam. Az ehhez használt program a mellékletben elérhető.



**3.14.** ábra – Átviteli függvény mérés pontossága / hibája

# 4. Ház fűtésének HIL szimulációja

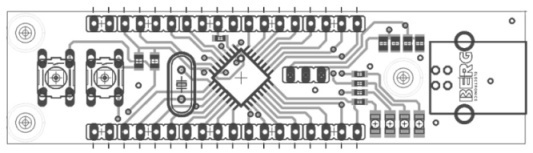
A feladat egy HIL rendszer készítése, amely egy ház fűtését szimulálja. A rendszer az állapotáról mérhető jeleket ad ki, és valós jelek segítségével szabályozható. A hallgatóknak egy vizsga keretében egy szabályzórendszert kell készíteniük. A szimuláció állapotát PC-s program segítségével követhetik.

## 4.1 Specifikáció

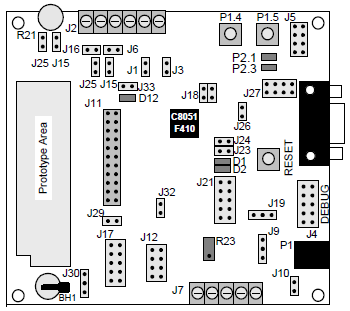
### 4.1.1. Felhasznált hardverek

A feladat bemutatása előtt fontosnak tartom ismertetni a felhasznált hardveres eszközöket és azok szerepét. A fejlesztés során összesen 4 eszközre volt szükség, így a feladat zavaros és megtévesztő lehet.

* PC
  + numerikus szimulációt végez
  + kijelzi annak állapotát indikátorokon, grafikonokon
  + kommunikál a MADAQ-kal
  + a MADAQ által küldött adatokat kijelzi
  + utasítást küld a MADAQ-nak
* MADAQ
  + **3.9.** ábrán látható
  + kommunikál a PC-vel
  + digitális jeleket olvas az F410 kit felől, és továbbítja a PC-nek
  + analóg jelet küld az F410 kit-nek a PC utasítása alapján
* 8051F410 development kit (F410 kit)
  + **4.2.** ábrán látható
  + a MADAQ digitális portjaira küld jeleket adatok PC-s kijelzéséhez és a szimulációba való beavatkozáshoz
  + a PC-s szimulációra vonatkozó analóg jelet olvas
* 8051F410 stick modul (F410 stick)
  + **4.1.** ábrán látható
  + szerepében a MADAQ-kal teljesen megegyező eszköz
  + csak hallgatók nagy létszáma miatt kellett az eszközt felhasználni



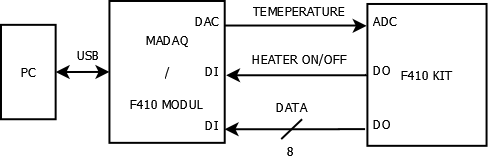
**4.1.** ábra – Az 8051F410 stick modul



**4.2.** ábra – A 8051F410 development kit

### 4.1.2. Feladat leírása

* HIL rendszer megalkotása a MADAQ és PC segítségével
* egy ház belső hőmérsékletét és fűtés rendszerét szimuláló LabView PC szoftver készítése
  + kommunikáció a MADAQ és PC között
  + ház numerikus szimulációja
  + ház állapotának kijelzése PC-n
    - idő
    - külső, belső, fűtővíz hőmérséklet
    - termosztát aktuális értéke
    - fűtés ON/OFF indikátor
    - kazán meghibásodása
* MADAQ forráskódjának bővítése
  + analóg jelek mérése/előállítása
  + digitális portok írása/olvasás
* vizsga feladatsor készítése a „Mikrovezérlők alkalmazástechnikája” nevű laboratóriumi gyakorlathoz a 8051F410 kit felhasználásával
  + a ház hőmérsékletének szabályozása, termosztát implementálása

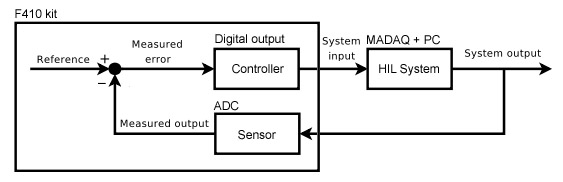


**4.3.** ábra – Szimulációhoz használt eszközök, és a köztük menő jelek

A **4.3.** ábrán látható vezetékek funkciója:

* *TEMPERATURE*: PC-n szimulált ház szobahőmérsékletével arányos analóg feszültség a 0 – 2V tartományban, amely a hőmérséklet tartományt fedi le.
* feszültség-hőmérséklet konverzió:
* *HEATER ON/OFF*: digitális jel a fűtés ki/bekapcsolásához
* *DATA-0 – DATA-7*: digitális jel, PC-s kijelzéshez használható 8 vezeték; a termosztát hőmérséklete

A hallgatóknak kiadott vizsgafeladat a PC-n szimulált ház fűtésének szabályzása a C8051F410 kit segítségével. A megoldás során csak digitális/analóg portok írásával/olvasásával kell foglalkozni, kommunikációval nem. A feladat tehát egy hőmérséklet érték szabályzása ON/OFF vezérlőjellel. Az alábbi ábra ezt szemlélteti:



**4.4.** ábra – Vizsgafeladat szemléltetése egy negatív visszacsatolási hurokkal

## 4.2. Ház termikus modellje

A ház belső fűtés-rendszerének szimulációjához egy MATLAB példaprogram differenciálegyenleteit és fizikai állandóit használtam fel, a megoldáshoz pedig az Euler módszert. A lenti egyenletekben az értékek egy házra jellemző konstansok. Jelentésük és értékük megtalálható a szimulációs programban. az aktuális külső, fűtővíz- és szobahőmérsékletet jelölik ebben a sorrendben. a fűtésből nyert hő, pedig a hő veszteség. A legfontosabb változó a melynek változását az alábbi néhány egyenlet írja le: [12]

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

Az Euler módszer formuláját felhasználva az iterációs képlet:

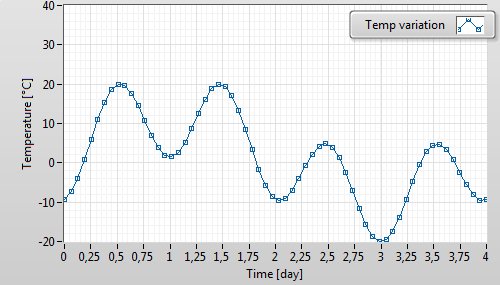
|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

A fenti egyenletekben nincs megadva, hogy a fűtővíz milyen dinamikával melegedjen fel, vagy hűljon le. A vezérlés egy jellel történik. A fűtővíz hőmérsékletének változásához ez RC szűrő áramkör matematikai modelljét használtam. Nem mondható valóságosnak, de a feladatnak megfelel. A fűtés állapotban 50°C-ra melegszik (4.3). állapotban felveszi a szoba hőmérsékletét (4.4). A fűtővíz hőmérsékletének szabályzása nem része a modellnek.

|  |  |  |
| --- | --- | --- |
|  |  | (4.3) |
|  |  | (4.4) |

A külső hőmérséklet egy kitalált napi- és egy 4 napos hő ingást követ, ahol délben a legmagasabb a hőmérséklet, és éjfélkor a legalacsonyabb egy abszolút paraméter-érték között. Ehhez hozzáadódik egy korlátozott Brown mozgás is. Az elképzelést a (4.5) képlet szemlélteti:

|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |



**4.5.** ábra – 4 nap hőingása a [-20°C, 20°C] tartományban

## 4.3. MADAQ szoftvere

A szimulációs LabView programhoz képest a MADAQ-on futó kód sokkal egyszerűbb. Lényegében a MADAQ egyetlen feladata egy fizikai interfészt biztosítani az 8051F410 kit és a PC között. A program futása alatt vagy utasítás érkezésére vár, melyek a PC felől érkeznek UART kommunikáció segítségével. A függvénnyel 1 byte-ot lehet beolvasni a soros port-ról.

**while** **(**1**)** **{**

**while** **(**SInOut**()!** **=** '@'**);**

c **=** SInOut**();**

**...**

// get port state, "p" = port

**else** **if** **(**c **==** 'p'**)** **{**

SOut**(~**P0**);**

SOut**(~**P1**);**

**}**

**...**

**else** **if** **(**c **==** 'd'**)** // set DAC0

**{**

unsigned char a**;**

c**=**SInOut**();** // hi

a**=**SInOut**();** // lo

SFRPAGE **=** DAC0\_PAGE**;**

DAC0CN **=** 0x84**;**

DAC0L**=**a**;**

DAC0H**=**c**;**

**}**

**}**

**4.6.** kódrészlet – DAC beállítása és digitális inputok állapotának továbbítása

## 4.4. LabView szoftver

Az **4.7.** ábrán látható program végzi a két legfontosabb feladatot:

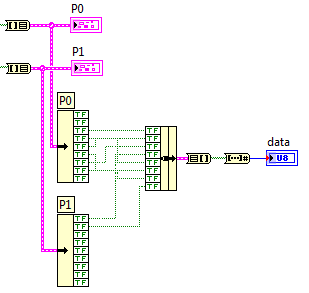
* 4.2. fejezetben leírt ház termikus modelljének szimulálása
* kommunikáció MADAQ-kal.

A differenciálegyenletek megoldása történhetne a MADAQ-on is, de mivel a rendszer időben relatívan lassan változik, ezért a kommunikáció okozta késés nem jelent gondot, nem okoz instabilitást sem. Egy időben gyorsan változó, ráadásul valós idejű rendszert problémásabb lenne a jelenlegi hardver összeállítással kezelni. Valamint egy 8 bites mikrovezérlő kisebb szabadságot ad bonyolultabb számítások implementálásra, mint a LabView.

Egy ház fűtése hosszú ideig tart valós időben, ezért szükség volt egy 3600-szoros időgyorsításra. A házban eltelt egy nap a valóságban A program rendelkezik egy belső órával, amelyet formátumban jelez ki. Az óra pl. a külső hő ingás megfigyeléséhez szükséges.

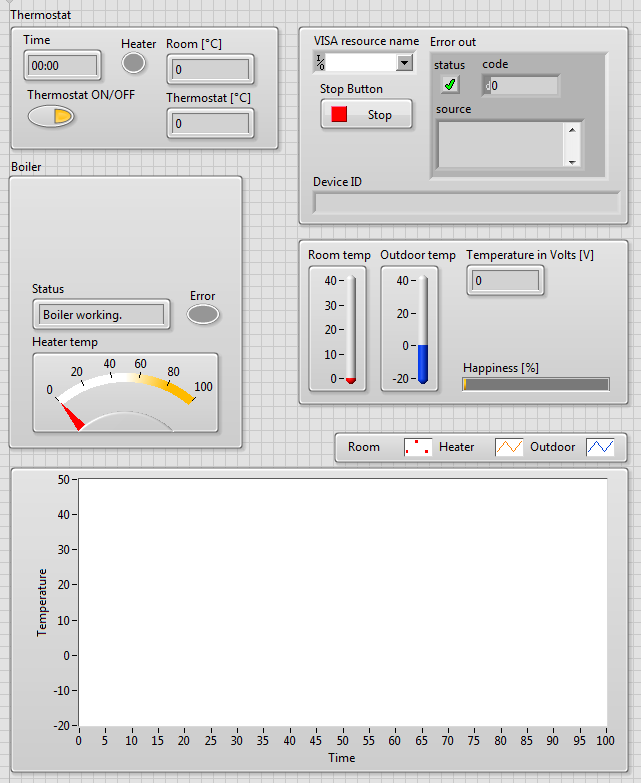
A program első induláskor inicializálja a soros port-ot, konfigurálja a MADAQ digitális bemeneteit a utasítás segítségével, majd lekérdezi az eszközinformációt. Ha nem érkezik válasz, akkor a szimuláció sem indul el. Utána egy végtelen ciklusban frekvenciával lekérdezi a MADAQ-on levő digitális portok állapotát (adat vezeték, fűtés ki/be kapcsolás) és beállítja a DAC feszültségét (hőmérséklettel arányos) a és utasítások segítségével. A numerikus szimuláció szintén frekvenciával végzi az iterációt. A ház állapotát a **4.7.** ábrán látható indikátorok jelzik, amelyek minden iterációban frissülnek. Ha a szoba hőmérséklete alá csökken, akkor a házban levő víz megfagy, amely a fűtőrendszer meghibásodását okozza. Ezt egy státusz indikátor jelzi. Hibás állapotban a fűtés nem lesz hatással a szoba hőmérsékletére, de a szimuláció nem áll meg. A gomb megnyomására a MADAQ visszaáll a kezdőállapotába, hogy más feladatokra használható legyen újraindítás nélkül, valamint lezárja a soros port-ot.

A MADAQ-on 10 szabad láb található, amelyből 8 vezetéket egy 8 bites számként értelmezek. A vezetékezést úgy terveztem, hogy a MADAQ és F410 kit közötti vezetékek helyiérték szerint legyenek egymás mellett és az összeszerelés minél egyszerűbb legyen. A MADAQ D-SUB csatlakozóján azonban a lábak nem helyiérték szerint lettek kiosztva a, sőt a nincs olyan port, amelynek mind a 8 lába szabad volna, így szoftveresen át kellett „rendezni”.



**4.8.** kódrészlet – 8 bites változó készítése bitek átrendezésével

A **4.7.** ábrán a hallgatók számára készített LabView program látható. Az indikátorokat úgy helyeztem el, hogy nagyjából egy valós kezelőfelületre hasonlítsanak. A *Thermostat* doboz egy valódi falón levő termosztátra, a *Boiler* doboz egy kombi kazánra hasonlít. A hőmérsékletek egy hőmérőre hasonlító indikátorról olvashatók, valamint grafikonon láthatók „valós” időben.



**4.7.** ábra – A LabView program előlapja.

# 5. Összefoglalás

## 5.1. Eredmények

A munkám során az alábbi eredményeket értem el:

* A MADAQ szoftverét kétcsatornás szinkronmérési és jelgenerálási funkcióval bővítettem
* LabView driver-eket készítettem átviteli függvény méréséhez (MADAQ segítségével)
* Elkészítettem egy jelgeneráláshoz használt, MADAQ-hoz csatlakozó szűrő panel kapcsolási és NYÁK tervét
* Összeforrasztottam az áramkört
* Készítettem egy módszert, mellyel a jelgenerátor hatékonyan működtethető
* A MADAQ és az F410 stick modul nevű áramkör szoftverét HIL szimulációs funkcióval bővítettem
* Egy ház fűtési rendszerét szimuláló LabView programot készítettem
* Vizsgafeladatot készítettem hallgatók számára a HIL rendszert felhasználva a Mikrovezérlők kurzushoz

## 5.2. Fejlesztési lehetőségek

A jövőben szeretném az átviteli függvény mérési funkciókat az egyetemi órák közé integrálni. Meg lehetne valósítani egyéb rendszerelméletben használt méréseket: pl. impulzus válasz, vagy egységugrásra adott válasz. A LabView programot lehetne bővíteni, hogy a mért rendszerekről több információt adjanak meg, pl. törési frekvencia, erősítés letörési meredeksége, időállandó. A jelgenerátor funkción úgy gondolom, lehetne még gyorsítani, illetve a pontosságán javítani. Az F410 stick modul továbbra is egy kihasználatlan eszköz, a HIL rendszerhez pedig könnyen írhatók új szimulációk. Szívesen készítenék újabb, bonyolultabb szabályzási feladatokat.

# Irodalomjegyzék

## Felhasznált irodalom

[1] Dr. Gingl Zoltán, Dr. Mingesz Róbert - Laboratory practicals with the C8051Fxxx microcontroller family *(megtekintve: 2015.05.14.)*

[2] Richard M. Murray - Analysis and Design of Feedback Systems - Transfer Functions

*(megtekintve: 2015.05.14.)*

<http://www.cds.caltech.edu/~murray/courses/cds101/fa04/caltech/am04_ch6-3nov04.pdf>

[3] Stephen P. Boyd - Sinusoidal steady-state and frequency response *(megtekintve:*

*2015.05.14.)*

<https://web.stanford.edu/~boyd/ee102/freq.pdf>

[4] Gerzson Miklós, Pletl Szilveszter – Irányítástechnika *(megtekintve: 2015.05.14.)*

<http://www.tankonyvtar.hu/hu/tartalom/tamop425/0008_gerzsonpletl/adatok.html>

[5] Differenciáló és integráló áramkörök vizsgálata *(megtekintve: 2015.05.14.)*

<http://titan.physx.u-szeged.hu/~opthome/elabor/A21_GYAK.pdf>

[6] Összetett szűrőkörök vizsgálata *(megtekintve: 2015.05.14.)*

<http://titan.physx.u-szeged.hu/~opthome/elabor/A22_GYAK.pdf>

[7] Wikipedia – RC circuit *(megtekintve: 2015.05.14.)*

<http://en.wikipedia.org/wiki/RC_circuit>

[8] Controllab - HIL *(megtekintve: 2015.05.14.)*

<http://www.hil-simulation.com/home/hil-simulation.html>

[9] Wikipedia - Euler method *(megtekintve: 2015.05.14.)*

<http://en.wikipedia.org/wiki/Euler_method>

[10] Mérés és Adatgyűjtés laboratóriumi gyakorlat – 5. óra *(megtekintve: 2015.05.14.)*

<http://www.inf.u-szeged.hu/~mingesz/Education/MAL/>

[11] OKAWA Electric Design - Filter Design and Analysis *(megtekintve: 2015.05.14.)*

<http://sim.okawa-denshi.jp/en/Fkeisan.htm>

[12] Mathworks documentation - Thermal Model of a House *(megtekintve: 2015.05.14.)*

<http://www.mathworks.com/help/simulink/examples/thermal-model-of-a-house.html>

[13 Wikibooks - Transfer Functions *(megtekintve: 2015.05.14.)*

<http://en.wikibooks.org/wiki/Control_Systems/Transfer_Functions>

[14] Lantos Béla – Irányítási rendszerek elmélete és tervezése I. *(megtekintve: 2015.05.14.)*

[15] Wikipedia – Control Theory *(megtekintve: 2015.05.14.)*

<http://en.wikipedia.org/wiki/Control_theory>

[16 C8051F41x development kit  *(megtekintve: 2015.05.14.)*

<http://nz.apexelex.com/images/microcontroller_kits/C8051F410TB.jpg>

[17] C8051F41x development kit user’s guide *(megtekintve: 2015.05.14.)*

<http://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F41x-DK.pdf>

[18] MADAQ *(megtekintve: 2015.05.14.)*

<http://www.noise.physx.u-szeged.hu/edudev/madaq/doc/MADAQS.jpg>

[19] Brian Douglas – Why learn control theory? *(megtekintve: 2015.05.14.)*

<https://www.youtube.com/watch?v=oBc_BHxw78s>

## Cégek, vállalkozások

[20] ROBTRON ELEKTRONIK TRADE KFT.

<http://www.ret.hu>

[21] MikroPAN KFT.

<http://www.mikropan.hu/>

# Nyilatkozat

Alulírott ………………..………… szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Tanszékcsoport ……………………….. Tanszékén készítettem, …………………….…….…… diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Informatikai Tanszékcsoport könyvtárában, a helyben olvasható könyvek között helyezik el.

Dátum

Aláírás

# Köszönetnyilvánítás

A szakdolgozatom majdnem teljes egészében a Zaj laboron belül készült. Köszönöm nekik, hogy biztosították a helyet, az eszközöket és a légkört. Szeretnék köszönetet mondani témavezetőmnek, Dr. Mingesz Róbertnek, aki mindig hasznos tanácsokkal látott el, aki gondolt arra is, amire én nem. Köszönöm Mellár János segítségét, aki az elektronikával kapcsolatos kérdéseimre válaszolt, és aki az áramköri terveimre áldását adta. Hálás vagyok Makan Gergelynek, aki nagylelkűen segített hirtelen jött kérdéseimben. Hálával tartozom továbbá szüleimnek, mamámnak, testvéremnek, akik nélkül ez a szakdolgozat nem jöhetett volna létre. Köszönöm nekik, hogy tanulmányaim során türelemmel és megértéssel támogattak, és minden helyzetben mellettem álltak.

# Mellékletek

Egy darab CD, amely tartalmazza a szűrő segédpanel kapcsolási rajzát, nyomtatott áramköri tervét, az áramkörről készített képeket, a LabView-s programokat átviteli függvény méréshez, és HIL szimulációhoz, a MADAQ és a 8051F410 stick modul forráskódjait, valamint a 8051F410 kit-hez készített vizsgafeladat megoldását.