

Általános információk, a diplomaterv szerkezete

A diplomaterv szerkezete a BME Villamosmérnöki és Informatikai Karán:

1. Diplomaterv feladatkiírás
2. Címoldal
3. Tartalomjegyzék
4. A diplomatervező nyilatkozata az önálló munkáról és az elektronikus adatok kezeléséről
5. Tartalmi összefoglaló magyarul és angolul
6. Bevezetés: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása
7. A feladatkiírás pontosítása és részletes elemzése
8. Előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések
9. A tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása
10. A megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek
11. Esetleges köszönetnyilvánítások
12. Részletes és pontos irodalomjegyzék
13. Függelék(ek)

Felhasználható a következő oldaltól kezdődő L^AT_EX diplomatervsablon dokumentum tartalma.

A diplomaterv szabványos méretű A4-es lapokra kerüljön. Az oldalak tükörmargóval készüljenek (mindenhol 2,5 cm, baloldalon 1 cm-es kötéssel). Az alapértelmezett betűkészlet a 12 pontos Times New Roman, másfeles sorközzel, de ettől kismértékben el lehet térni, ill. más betűtípus használata is megengedett.

Minden oldalon – az első négy szerkezeti elem kivételével – szerepelnie kell az oldalszámnak.

A fejezeteket decimális beosztással kell ellátni. Az ábrákat a megfelelő helyre be kell illeszteni, fejezetként decimális számmal és kifejező címmel kell ellátni. A fejezeteket decimális alaosztással számozzuk, maximálisan 3 alaosztás mélységben (pl. 2.3.4.1.). Az ábrákat, táblázatokat és képleteket célszerű fejezetként külön számozni (pl. 2.4. ábra, 4.2. táblázat vagy képletnél (3.2)). A fejezetcímeket igazítsuk balra, a normál szövegnél viszont használjunk sorkiegyenlítést. Az ábrákat, táblázatokat és a hozzájuk tartozó címet igazítsuk középre. A cím a jelölt rész alatt helyezkedjen el.

A képeket lehetőleg rajzoló programmal készítsék el, az egyenleteket egyenlet-szerkesztő segítségével írják le (A L^AT_EX ehhez kézenfekvő megoldásokat nyújt).

Az irodalomjegyzék szövegközi hivatkozása történhet sorszámozva (ez a preferált megoldás) vagy a Harvard-rendszerben (a szerző és az évszám megadásával). A teljes lista névsor szerinti sorrendben a szöveg végén szerepeljen (sorszámozott irodalmi hivatkozások esetén hivatkozási sorrendben). A szakirodalmi források címeit azonban mindig az eredeti nyelven kell megadni, esetleg zárójelben a fordítással. A listában szereplő valamennyi publikációra hivatkozni kell a szövegben (a L^AT_EX-sablon a Bib_TE_X segítségével mindent automatikusan kezeli). Minden publikáció a szerzők után a következő adatok szerepelnek: folyóirat cikkeknel a pontos cím, a folyóirat címe, évfolyam, szám, oldalszám tól-ig. A folyóiratok címét csak akkor rövidítsük, ha azok nagyon közismertek vagy nagyon hosszúak. Internetes hivatkozások megadásakor fontos, hogy az elérési út előtt megadjuk az oldal tulajdonosát és tartalmát (mivel a link egy idő után akár elérhetetlenné is válhat), valamint az elérés időpontját.

Fontos:

- A szakdolgozatkészítő / diplomatervező nyilatkozata (a jelen sablonban szereplő szövegtartalommal) kötelező előírás, Karunkon ennek hiányában a szakdolgozat/diplomaterv nem bírálható és nem védhető!
- Mind a dolgozat, mind a melléklet maximálisan 15 MB méretű lehet!

Jó munkát, sikeres szakdolgozatkészítést, ill. diplomatervezést kívánunk!

FELADATKIÍRÁS

A feladatkiírást a tanszéki adminisztrációban lehet átvenni, és a leadott munkába eredeti, tanszéki pecséttel ellátott és a tanszékvezető által aláírt lapot kell belefűzni (ezen oldal *helyett*, ez az oldal csak útmutatás). Az elektronikusan feltöltött dolgozatban már nem kell beleszerkeszteni ezt a feladatkiírást.



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatikai Tanszék

Kvíz generálási lehetőségek vizsgálata egy fullstack megoldás megvalósítása keretében

SZAKDOLGOZAT

Készítette
Muzslai László

Konzulens
dr. Ekler Péter

2024. november 8.

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. Témaválasztás indoklása	1
1.2. Felhasznált technológia jelentősége/elterjedtsége	1
1.3. Dolgozat felépítése	3
2. L^AT_EX-eszközök	4
2.1. A szerkesztéshez használatos eszközök	4
2.2. A dokumentum lefordítása Windows alatt	5
2.3. Eszközök Linuxhoz	6
3. A dolgozat formai kivitele	7
3.1. A dolgozat kimérete	7
3.2. A dolgozat nyelve	7
3.3. A dokumentum nyomdatechnikai kivitele	7
4. A L^AT_EX-sablon használata	9
4.1. Címkék és hivatkozások	9
4.2. Ábrák és táblázatok	9
4.3. Felsorolások és listák	11
4.4. Képletek	12
4.5. Irodalmi hivatkozások	13
4.6. A dolgozat szerkezete és a forrásfájlok	16
4.7. Alapadatok megadása	18
4.8. Új fejezet írása	18
4.9. Definíciók, tételek, példák	18
Köszönetnyilvánítás	19

Irodalomjegyzék	20
Függelék	22
F.1. A TeXstudio felülete	22
F.2. Válasz az „Élet, a világmindenség, meg minden” kérdésére	23

HALLGATÓI NYILATKOZAT

Alulírott *Muzslai László*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2024. november 8.

Muzslai László
hallgató

Kivonat

A közoktatásban és a felsőoktatásban is gyakori probléma mind a tanárok, mind a diákok számára az időhiány a rengeteg munka miatt. Ez az alkalmazás a tanárok munkáját hivatott segíteni az előző félévben elkészített REST API-t felhasználva, és a hozzá írt Android-eszközökre készült program továbbfejlesztése által. Az alkalmazás lehetőséget biztosít egy nagyméretű kérdésbank létrehozására és tárolására. A kérdések bármikor módosíthatók, törölhetők, vagy hozhatók létre újak. Természetesen nem csak egy ember dolgozhat ugyanazon a tárgyon; a kérdésbank és a számonkérések közösen szerkeszthetők.

Az alkalmazás lehetőséget biztosít a kérdések létrehozása mellett egyéni témák létrehozására, amivel a kérdéseket és feladatsorokat lehet csoportosítani. Továbbá el lehet készíteni a saját pontrendszert, akár több félért is, amelyet dinamikusan lehet változtatni a kérdéseknél. Fontos szempont volt az automatizált javítás segítése, így csak egyszerű kérdések vannak: igaz-hamis és feleletválasztós kérdések. Sajnos az AI még nem tart ott, hogy bármilyen kézírást pontosan felismerjen, és ebből a szövegből megállapítsa annak helyességét. Ennek ellenére a szövegfelismerő funkció így is támogatja a javítást, aminek az eredményét megkapja a javító.

Ezekből az elemekből áll össze a számonkérés. Ez a szoftver csak a kérdéssorok összeállításáért és kiértékeléséért felel. Ennek megfelelően elő kell állítani magát a feladatsort. Egy dolgozatot ki lehet exportálni PDF formátumban, erről egy előnézet is lesz, amin nagyjából látszik, hogyan fog kinézni, de a végső változat csak az exportálást követően fog látszani. Ezt követően szabadon nyomtathatóvá válik.

Egy modern szoftver esetén elvárt, hogy könnyen és kényelmesen lehessen kezelni, mindenki számára a neki tetsző módon. Ennek alapján úgy döntöttem, hogy felhasználom az Android fejlesztésben szerzett tapasztalataimat. 2021 augusztusában jelent meg a Compose Multiplatform technológia, amely kedvez az Android-fejlesztőknek, mivel a natív Android-megoldások könnyen átültethetők egy cross-platform alkalmazásba. Jelenleg stabilan működik Android-, asztali- és iOS-alkalmazások készítéséhez, eszköz hiányában az első kettőt készítettem el.

Abstract

In both public and higher education, time constraints are a common issue for both teachers and students due to the large workload. This application is intended to assist teachers by utilizing the REST API developed in the previous semester and enhancing the program created for Android devices. The application allows for the creation and storage of a large question bank. Questions can be modified, deleted, or new ones can be created at any time. Of course, more than one person can work on the same subject; the question bank and the tests can be edited collaboratively.

In addition to creating questions, the application also allows for the creation of custom topics, which can be used to organize questions and assignments. Furthermore, a custom scoring system can be created, even multiple types, which can be dynamically adjusted for different questions. An important aspect was to assist in automated grading, so only simple questions are included: true/false and multiple-choice questions. Unfortunately, AI is not yet at the level where it can accurately recognize any handwriting and determine its correctness from the text. Nonetheless, the text recognition function still supports grading, and the results are provided to the grader.

These elements come together to form the assessment. This software is responsible solely for compiling and evaluating the question sheets. Accordingly, the task sheet itself must be generated. A test can be exported in PDF format, with a preview available that roughly shows how it will look, though the final version will only be visible after exporting. After this, it can be freely printed.

For modern software, it is expected to be easy and convenient to use, allowing everyone to handle it in their preferred way. Based on this, I decided to leverage my experience in Android development. The Compose Multiplatform technology, released in August 2021, is favorable for Android developers, as native Android solutions can easily be adapted into a cross-platform application. It currently works stably for creating Android, desktop, and iOS applications; due to a lack of devices, I have implemented the first two.

1. fejezet

Bevezetés

1.1. Témaválasztás indoklása

2023/2024 őszi félévében ismerkedtem meg a mobil, azon belül is az Android fejlesztéssel. Az ezt követő félévben tovább mélyítettem a tudásomat ebben a témában, az Önálló laboratórium tárgy keretein belül elkezdtem fejleszteni a szakdolgozatom alapjaként szolgáló alkalmazást. Szintén ebben a félévben hallgattam az Android alapú szoftverfejlesztés és a Kotlin alapú szoftverfejlesztés tárgyakat, amik segítettek jobban megérteni ezt a területet. Az így elsajátított tudás és az önálló kutatás és tanulás során előállt egy Kotlin nyelven írt REST API, ami egy PostgreSQL adatbázissal biztosít kommunikációt, és természetesen egy Android alkalmazás, amelyet a szakdolgozat során tovább bővítettem és alakítottam át egy cross-platform szoftverré.

A kérdéssor összeállító alkalmazás ötletét a konzulensem vetette fel, hasznos lenne, ha létezne egy ilyen eszköz, amivel könnyen megoldható ez a feladat. Megtetszett nekem is az ötlet, mivel vannak ismerőseim és családtagjaim, akik szintén tudnának egy ilyen alkalmazást hasznosítani a munkájuk vagy egyéb elfoglaltságaik kapcsán. Egy nagyobb méretű fullstack alkalmazás előállítása túlmutat az Önálló laboratórium keretein, így rengeteg fejlesztési ötlet és lehetőség nem fért bele a félévbe. Továbbgondolva ezt a projektet, folytattam a munkát a szoftveren. Ezen kívül mindig szeretek új és érdekes dolgokat kipróbálni, és ha megtetszik, alaposan tanulmányozni és megtanulni. Pont ezért választottam a Google és a JetBrains legújabb megoldásait a szakdolgozathoz.

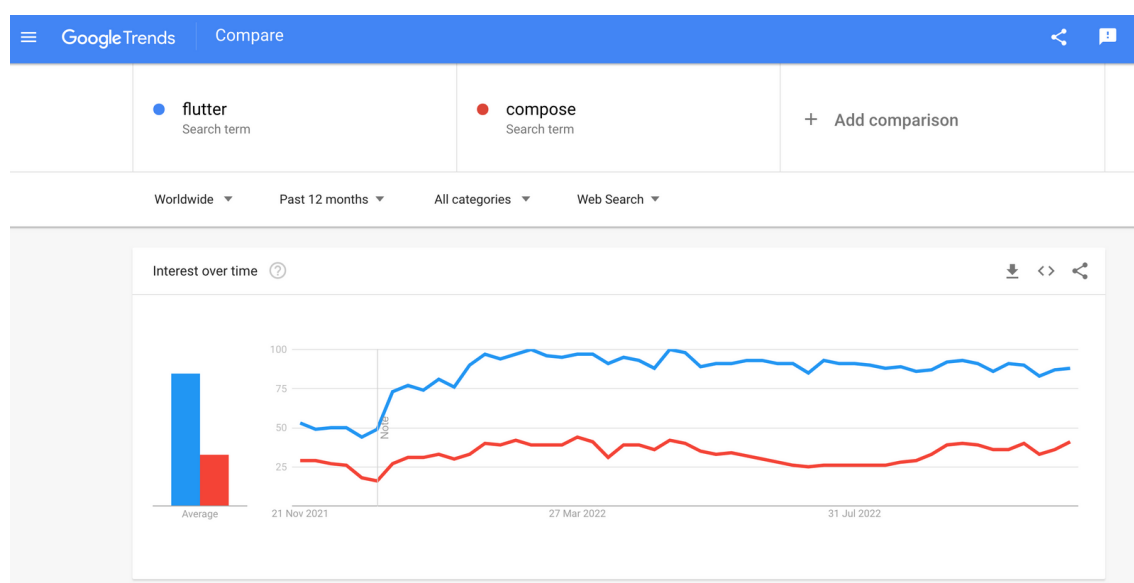
A most elkészített szoftvert jelentősen tovább lehet még fejleszteni, felhasználók, szervezetek regisztrálásával és elkülönítésével, több fajta kérdéstípus megvalósításával. Bevezetni a szervezeteken belül az oktató és diák csoportokat, és egy online kitöltési formát is megtervezni, létrehozni. Az Önálló laboratórium alatt is élveztem ezzel foglalkozni, és még mindig szívesen fejleszteném tovább, és tenném jobba az alkalmazást.

1.2. Felhasznált technológia jelentősége/elterjedtsége

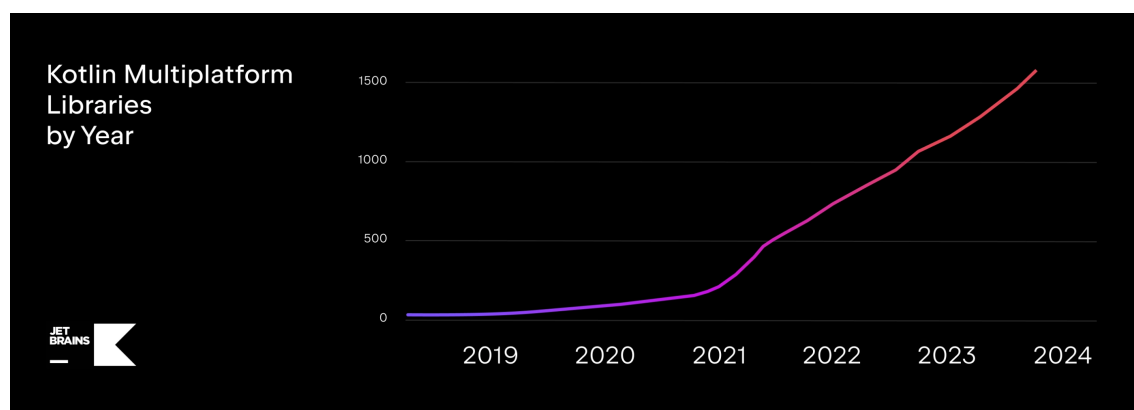
Megfigyelve a mai trendeket, láttam, hogy a multiplatform fejlesztés egyre népszerűbb, mivel gyorsabban és hatékonyabban lehet egyszerűbb alkalmazásokat elkészíteni több fajta felhasználói réteg számára. Úgy döntöttem, hogy kipróbálok egy új megoldást, ami a Compose Multiplatform; 2021 augusztusában jelent meg az 1.0 alpha verziója, és ezt a Kotlin Multiplatform egy évvel korábbi megjelenése tette lehetővé. Az Android fejlesztők körében kifejezetten népszerű lett, annak ellenére is, hogy még mindig van rengeteg

funkció, amit nem támogat, de jelenleg is aktívan fejlesztik, és válik hónapról-hónapra egyre jobbá. A fejlesztések havonta / néhány havonta érkeznek mind a Kotlin nyelvhez és a Kotlin Multiplatformhoz[12], mind a Compose Multiplatformhoz[8]. Jelenleg vannak sokkal jobban elterjedt, használtabb cross-platform keretrendszerek, mint például a Flutter (1.1. ábra) vagy a React Native, ami iOS és Android fejlesztést tesz lehetővé JavaScript/TypeScript nyelven, így a webes fejlesztők gyakran használják natív mobilos appokhoz.

Szerencsére a fejlesztés folyamatos és gyors, amit segít az is, hogy sok nyílt forráskódú könyvtár is készül a felhasználók által.[11] A Compose Multiplatform ennek az ellenpontja lesz[10]; sajnos a webes támogatás még csak alpha verzióban van, így eléggé instabil, és sok olyan eszköz nem használható még, ami a többi területen már stabilan működik, így ezzel egyelőre a szakdolgozat keretein belül nem foglalkoztam részletesebben a webes megoldás megvalósításával.



1.1. ábra. A Flutter és a Compose keresési arányai [3]



1.2. ábra. A könyvtárak növekedésének gyors üteme [11]

A visszajelzések és a mostani szoftverfejlesztési irányokból arra lehet következtetni, hogy még hosszú jövő áll a technológiák előtt. Könnyen, gyorsan és hatékonyan lehet akár egyszerre az összes platformra alkalmazást fejleszteni, nagy méretű közös és kis méretű natív kódbázis írásával és karbantartásával, összesen két nyelv ismeretével. Webre, And-

roidra és asztali alkalmazáshoz elegendő lehet a Kotlin nyelv, esetleg egy kis HTML és JavaScript ismeret; iOS esetén minimális Swift és SwiftUI ismeret jól jöhet, de ez hasonlít a Kotlinra és a Compose-ra. Az a tény, hogy a Compose Multiplatform csupán Kotlin nyelven írt kódbázissal képes natívan megjeleníteni az elkészült alkalmazást minden platformon, nagyon erős eszközzé teszi. Például az Androidon futó alkalmazás az Androidos gombokat, görgetést stb. használja, míg iOS-en az ott megszokott stílust és irányítást kapja a felhasználó.

1.3. Dolgozat felépítése

Az alábbiakban bemutatom a dolgozat felépítését.

1. *Feladat-specifikáció:*

- (a) *Feladat részletes leírása*
- (b) *Diagramok:* Bemutatok néhány diagramot és ábrát, amelyek segítik az alkalmazás felépítésének megértését.

2. *Irodalomkutatás:*

- (a) *Felhasznált technológiák:* az 1.2. szakaszban leírtak és a használt könyvtárak részletes bemutatása
- (b) *Hasonló megoldások:* Rövid összevetés a többi hasonló cross-platform megoldással.

3. *Felsőszintű architektúra:*

- (a) *High level architektúra*
- (b) *Rendszer felépítései, komponensei*

4. *Részletes megvalósítás:*

- (a) *UML class diagramok*
- (b) *Entity-relation diagram*
- (c) *Szekvencia diagram*
- (d) *Kódrészek*

5. *Tesztelés:*

- (a) *Felhasználói leírás*
- (b) *A program bemutatása képekkel*
- (c) *Lehetséges tesztelési megoldások ismertetése*

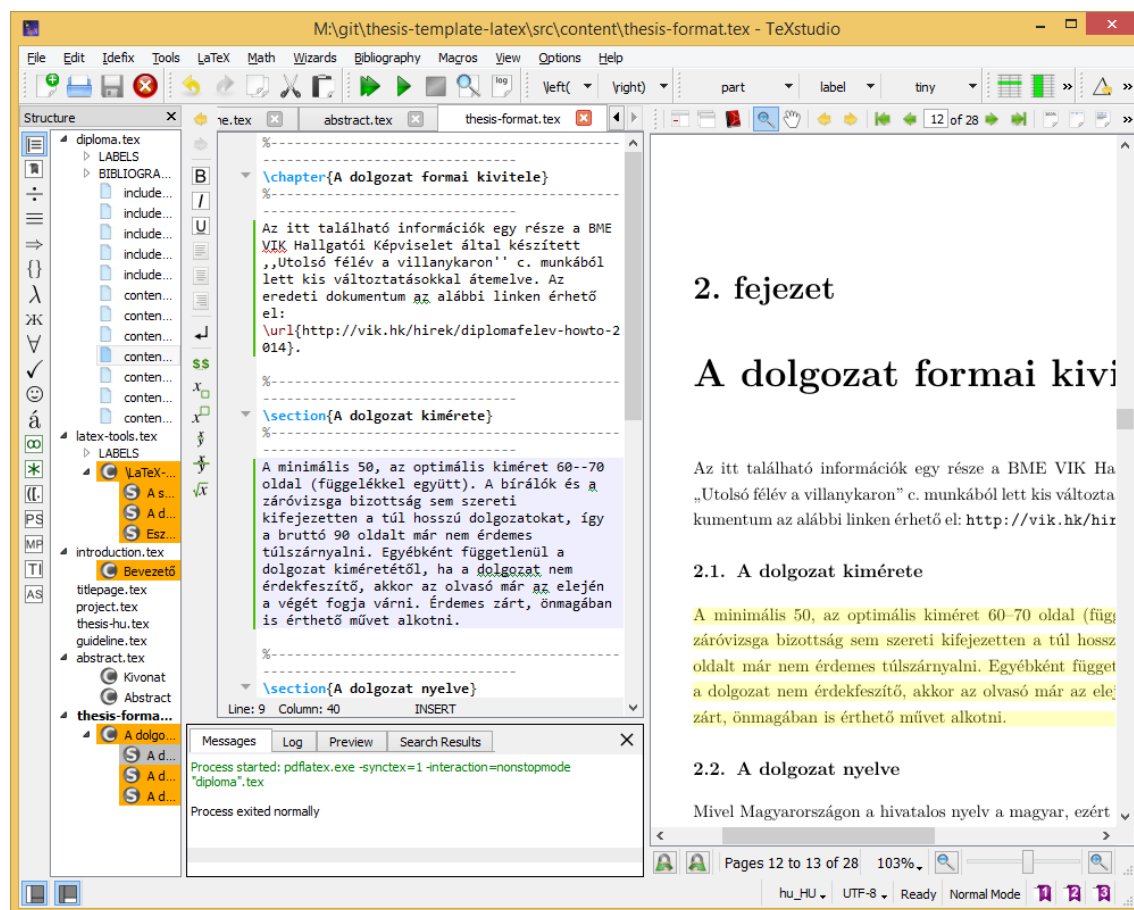
6. *Összefoglalás, továbbfejlesztési lehetőségek*

2. fejezet

L^AT_EX-eszközök

2.1. A szerkesztéshez használatos eszközök

Ez a sablon TeXstudio 2.8.8 szerkesztővel készült. A TeXstudio egy platformfüggetlen, Windows, Linux és Mac OS alatt is elérhető L^AT_EX-szerkesztőprogram számtalan hasznos szolgáltatással (2.1. ábra). A szoftver ingyenesen letölthető¹.



2.1. ábra. A TeXstudio L^AT_EX-szerkesztő.

¹A TeXstudio hivatalos oldala: <http://texstudio.sourceforge.net/>

A TeXstudio telepítése után érdemes még letölteni a magyar nyelvű helyesíráseellenőrző-szótárakat hozzá. A TeXstudio az OpenOffice-hoz használatos formátumot tudja kezelni. A TeXstudio beállításainál a General fülön a Dictionaries résznél tudjuk megadni, hogy melyik szótárt használja.

Egy másik használható Windows alapú szerkesztőprogram a LEd² (LaTeX Editor), a TeXstudio azonban stabilabb, gyorsabb, és jobban használható.

2.2. A dokumentum lefordítása Windows alatt

A TeXstudio és a LEd kizárólag szerkesztőprogram (bár az utóbbiban DVI-nézegető is van), így a dokumentum fordításához szükséges eszközöket nem tartalmazza. Windows alatt alapvetően két lehetőség közül érdemes választani: MiKTeX (<http://miktex.org/>) és TeX Live (<http://www.tug.org/texlive/>) programcsomag. Az utóbbi működik Mac OS X, GNU/Linux alatt és Unix-származékokon is. A MiKTeX egy alapsomag telepítése után mindig letölti a használt funkciókhoz szükséges, de lokálisan hiányzó \TeX -csomagokat, míg a TeX Live DVD ISO verzóban férhető hozzá. Ez a dokumentum TeX Live 2008 programcsomag segítségével fordult, amelynek DVD ISO verziója a megadott oldalról letölthető. A sablon lefordításához a disztribúcióban szereplő magyar .ldf fájlt a <http://www.math.bme.hu/latex/> változatra kell cserélni, vagy az utóbbi változatot be kell másolni a projekt-könyvtárba (ahogy ezt meg is tettük a sablonban) különben anomáliák tapasztalhatók a dokumentumban (pl. az ábra- és táblázat-aláírások formátuma nem a beállított lesz, vagy bizonyos oldalakon megjelenik alapértelmezésben egy fejléc). A TeX Live 2008-at még nem kell külön telepíteni a gépre, elegendő DVD-ről (vagy az ISO fájlból közvetlenül, pl. DaemonTools-szal) használni.

Ha a MiKTeX csomagot használjuk, akkor parancssorból a következő módon tudjuk újrafordítani a teljes dokumentumot:

```
$ texify -p thesis.tex
```

A texify parancs a MiKTeX programcsomag miktex/bin alkönyvtárában található. A parancs gondoskodik arról, hogy a szükséges lépéseket (fordítás, hivatkozások generálása stb.) a megfelelő sorrendben elvégezze. A -p kapcsoló hatására PDF-et generál. A fordítást és az ideiglenes fájlok törlését elvégezhetjük a sablonhoz mellékelt manual_build.bat szkript segítségével is.

A \TeX -eszközöket tartalmazó programcsomag binárisainak elérési útját gyakran be kell állítani a szerkesztőprogramban, például TeXstudio esetén legegyszerűbben az Options / Configure TeXstudio... / Commands menüponttal előhívott dialógusablakban tehetjük ezt meg.

A PDF- \LaTeX használata esetén a generált dokumentum közvetlenül PDF-formátumban áll rendelkezésre. Amennyiben a PDF-fájl egy PDF-nézőben (pl. Adobe Acrobat Reader vagy Foxit PDF Reader) meg van nyitva, akkor a fájlleíró a PDF-néző program tipikusan lefoglalja. Ilyen esetben a dokumentum újrafordítása hibaüzenettel kilep. Ha bezárjuk és újra megnyitjuk a PDF dokumentumot, akkor pedig a PDF-nézők többsége az első oldalon nyitja meg a dokumentumot, nem a legutóbb olvasott oldalon. Ezzel szemben például az egyszerű és ingyenes [Sumatra PDF](#) nevű program képes arra, hogy a megnyitott dokumentum megváltozását detektálja, és frissítse a nézetet az aktuális oldal megtartásával.

²A LEd hivatalos oldala: <http://www.latexeditor.org/>

2.3. Eszközök Linuxhoz

Linux operációs rendszer alatt is rengeteg szerkesztőprogram van, pl. a KDE alapú Kile jól használható. Ez ingyenesen letölthető, vagy éppenséggel az adott Linux-disztribúció eleve tartalmazza, ahogyan a dokumentum fordításához szükséges csomagokat is. Az Ubuntu Linux disztribúciók alatt például legtöbbször a `texlive-*` csomagok telepítésével használhatók a \LaTeX -eszközök. A jelen sablon fordításához szükséges csomagok (kb. 0,5 GB) az alábbi paranccsal telepíthetők:

```
$ sudo apt-get install texlive-latex-extra texlive-fonts-extra texlive-fonts-recommended  
texlive-xetex texlive-science
```

Amennyiben egy újabb csomag hozzáadása után hiányzó fájlra utaló hibát kapunk a fordítótól, telepítenünk kell az azt tartalmazó TeX Live csomagot. Ha pl. a `bibentry` csomagot szeretnénk használni, futtassuk az alábbi parancsot:

```
$ apt-cache search bibentry  
texlive-luatex - TeX Live: LuaTeX packages
```

Majd telepítsük fel a megfelelő TeX Live csomagot, jelen esetben a `texlive-lualatex-et`. (Egy LaTeX csomag több TeX Live csomagban is szerepelhet.)

Ha gyakran szerkesztünk más \LaTeX dokumentumokat is, kényelmes és biztos megoldás a teljes TeX Live disztribúció telepítése, ez azonban kb. 4 GB helyet igényel.

```
sudo apt-get install texlive-full
```

3. fejezet

A dolgozat formai kivitele

Az itt található információk egy része a BME VIK Hallgatói Képviselőlet által készített „Utolsó félév a villanykaron” c. munkából lett kis változtatásokkal átemelve. Az eredeti dokumentum az alábbi linken érhető el: <http://vik.hk/hirek/diplomafelev-howto-2015>.

3.1. A dolgozat kimérete

Szakdolgozat esetében minimum 30, 45 körüli ajánlott oldalszám lehet az iránymutató. De mindenképp érdemes rákérdezni a konzulensnél is az elvárásokra, mert tanszékenként változók lehetnek az elvárások.

Mesterképzésen a Diplomatervezés 1 esetében a beszámoló még inkább az Önálló laboratóriumi beszámolókhöz hasonlít, tanszékenként eltérő formai követelményekkel, – egy legalább 30 oldal körüli dolgozat az elvárt – és az elmúlt fél éves munkáról szól. De egyben célszerű, ha ez a végleges diplomaterv alapja is. (A végleges 60-90 oldal körülbelül a hasznos részre nézve)

3.2. A dolgozat nyelve

Mivel Magyarországon a hivatalos nyelv a magyar, ezért alapértelmezésben magyarul kell megírni a dolgozatot. Aki külföldi posztgraduális képzésben akar részt venni, nemzetközi szintű tudományos kutatást szeretne végezni, vagy multinacionális cégnél akar elhelyezkedni, annak célszerű angolul megírnia diplomadolgozatát. Mielőtt a hallgató az angol nyelvű verzió mellett dönt, erősen ajánlott mérlegelni, hogy ez mennyi többletmunkát fog a hallgatónak jelenteni fogalmazás és nyelvhelyesség terén, valamint – nem utolsósorban – hogy ez mennyi többletmunkát fog jelenteni a konzulens illetve bíráló számára. Egy nehezen olvasható, netalán érthetetlen szöveg teher minden játékos számára.

3.3. A dokumentum nyomdatechnikai kivitele

A dolgozatot A4-es fehér lapra nyomtatva, 2,5 centiméteres margóval (+1 cm kötésbeni), 11–12 pontos betűmérettel, talpas betűtípussal és másfeles sorközzel célszerű elkészíteni.

Annak érdekében, hogy a dolgozat külsőleg is igényes munka benyomását keltse, érdemes figyelni az alapvető tipográfiai szabályok betartására [4].

4. fejezet

A L^AT_EX-sablon használata

Ebben a fejezetben röviden, implicit módon bemutatjuk a sablon használatának módját, ami azt jelenti, hogy sablon használata ennek a dokumentumnak a forráskódját tanulmányozva válik teljesen világossá. Amennyiben a szoftver-keretrendszer telepítve van, a sablon alkalmazása és a dolgozat szerkesztése L^AT_EX-ben a sablon segítségével tapasztalataink szerint jóval hatékonyabb, mint egy WYSWYG (*What You See is What You Get*) típusú szövegszerkesztő esetén (pl. Microsoft Word, OpenOffice).

4.1. Címkék és hivatkozások

A L^AT_EX dokumentumban címkéket (`\label`) rendelhetünk ábrákhoz, táblázatokhoz, fejezetekhez, listákhoz, képletekhez stb. Ezekre a dokumentum bármely részében hivatkozhatunk, a hivatkozások automatikusan feloldásra kerülnek.

A sablonban makrókat definiáltunk a hivatkozások megkönnyítéséhez. Ennek megfelelően minden ábra (*figure*) címkéje `fig:` kulcsszóval kezdődik, míg minden táblázat (*table*), képlet (*equation*), fejezet (*section*) és lista (*listing*) rendre a `tab:`, `eq:`, `sec:` és `lst:` kulcsszóval kezdődik, és a kulcsszavak után tetszőlegesen választott címke használható. Ha ezt a konvenciót betartjuk, akkor az előbbi objektumok számára rendre a `\figref`, `\tabref`, `\eqref`, `\sectref` és `\listref` makrókkal hivatkozhatunk. A makrók paramétere a címke, amelyre hivatkozunk (a kulcsszó nélkül). Az összes említett hivatkozástípus, beleértve az `\url` kulcsszóval bevezetett web-hivatkozásokat is a `hyperref`¹ csomagnak köszönhetően aktívak a legtöbb PDF-nézegetőben, rájuk kattintva a dokumentum megfelelő oldalára ugrik a PDF-néző vagy a megfelelő linket megnyitja az alapértelmezett böngészővel. A `hyperref` csomag a kimeneti PDF-dokumentumba könyvjelzőket is készít a tartalomjegyzékből. Ez egy szintén aktív tartalomjegyzék, amelynek elemeire kattintva a nézegető behozza a kiválasztott fejezetet.

4.2. Ábrák és táblázatok

Használjunk vektorgrafikus ábrákat, ha van rá módunk. PDFLaTeX használata esetén PDF formátumú ábrákat lehet beilleszteni könnyen, az EPS (PostScript) vektorgrafikus képformátum beillesztését a PDFLaTeX közvetlenül nem támogatja (de lehet konvertálni, lásd később). Ha vektorgrafikus formában nem áll rendelkezésünkre az ábra, akkor a

¹Segítségével a dokumentumban megjelenő hivatkozások nem csak dinamikussá válnak, de színezhetőek is, bővebbet erről a csomag dokumentációjában találunk. Ez egyúttal egy példa lábjegyzet írására.

veszteségmentes PNG, valamint a veszteséges JPEG formátumban érdemes elmenteni. Figyeljünk arra, hogy ilyenkor a képek felbontása elég nagy legyen ahhoz, hogy nyomtatásban is megfelelő minőséget nyújtson (legalább 300 dpi javasolt). A dokumentumban felhasznált képfájlokat a dokumentum forrása mellett érdemes tartani, archiválni, mivel ezek hiányában a dokumentum nem fordul újra. Ha lehet, a vektorgrafikus képeket vektorgrafikus formátumban is érdemes elmenteni az újrafelhasználhatóság (az átszerkeszthetőség) érdekében.

Kapcsolási rajzok legtöbbször kimásolhatók egy vektorgrafikus programba (pl. CorelDraw) és onnan nagyobb felbontással raszterizálva kimenthetők PNG formátumban. Ugyanakkor kiváló ábrák készíthetők Microsoft Visio vagy hasonló program használatával is: Visio-ból az ábrák közvetlenül PDF-be is menthetők.

Lehetőségeink Matlab ábrák esetén:

- Képernyőlopás (*screenshot*) is elfogadható minőségű lehet a dokumentumban, de általában jobb felbontást is el lehet érni más módszerrel.
- A Matlab ábrát a File/Save As opcióval lementhetjük PNG formátumban (ugyanaz itt is érvényes, mint korábban, ezért nem javasoljuk).
- A Matlab ábrát az Edit/Copy figure opcióval kimásolhatjuk egy vektorgrafikus programba is és onnan nagyobb felbontással raszterizálva kimenthetjük PNG formátumban (nem javasolt).
- Javasolt megoldás: az ábrát a File/Save As opcióval EPS *vektorgrafikus* formátumban elmentjük, PDF-be konvertálva beillesztjük a dolgozatba.

Az EPS kép az epstopdf programmal² konvertálható PDF formátumba. Célszerű egy batch-fájlt készíteni az összes EPS ábra lefordítására az alábbi módon (ez Windows alatt működik).

```
@echo off
for %%j in (*.eps) do (
    echo converting file "%%j"
    epstopdf "%%j"
)
echo done .
```

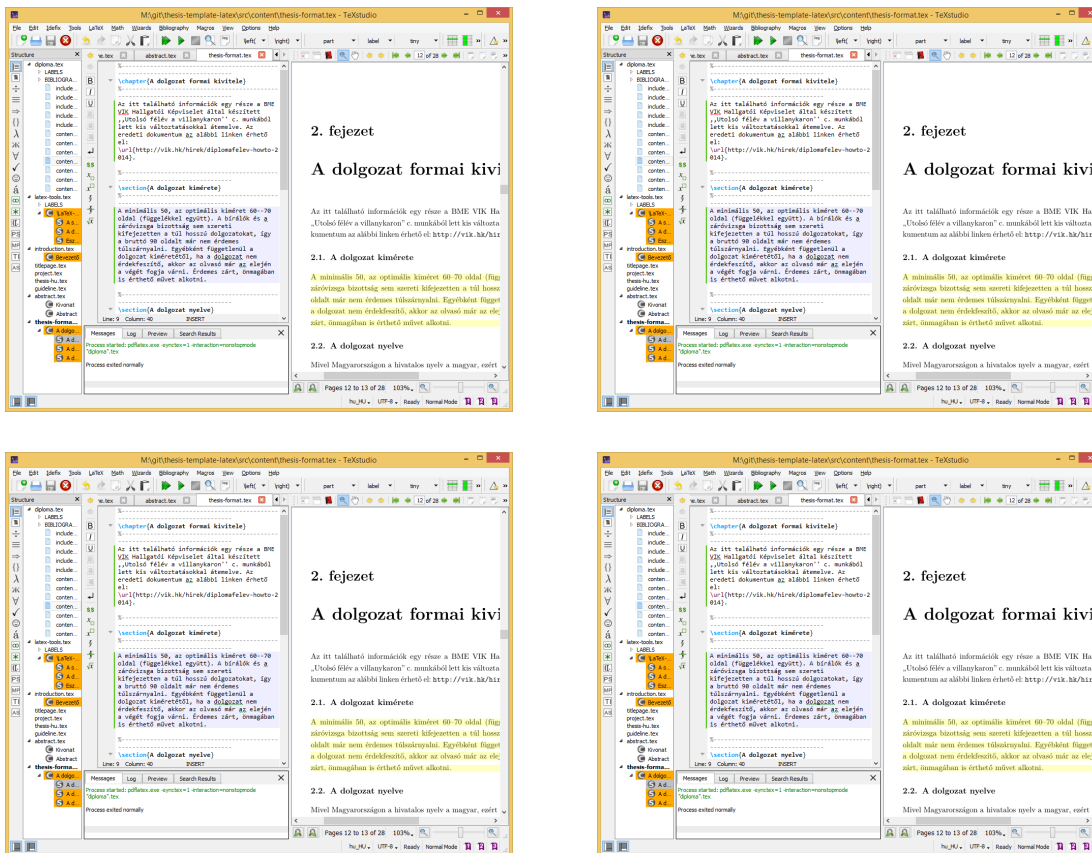
Egy ilyen parancsfájlt (convert.cmd) elhelyeztük a sablon figures\eps könyvtárba, így a felhasználónak csak annyi a dolga, hogy a figures\eps könyvtárba kimenti az EPS formátumú vektorgrafikus képet, majd lefuttatja a convert.cmd parancsfájlt, ami PDF-be konvertálja az EPS fájlt.

Ezek után a PDF-ábrát ugyanúgy lehet a dokumentumba beilleszteni, mint a PNG-t vagy a JPEG-et. A megoldás előnye, hogy a lefordított dokumentumban is vektorgrafikusan tárolódik az ábra, így a mérete jóval kisebb, mintha raszterizáltuk volna beillesztés előtt. Ez a módszer minden – az EPS formátumot ismerő – vektorgrafikus program (pl. CorelDraw) esetén is használható.

A képek beillesztésére a 2. fejezetben mutattunk be példát (2.1. ábra). Az előző mondatban egyúttal az automatikusan feloldódó ábrahivatkozásra is láthatunk példát. Több képfájl is beilleszthetünk egyetlen ábrába. Az egyes képek közötti horizontális és vertikális margót metrikusan szabályozhatjuk (4.1. ábra). Az ábrák elhelyezését számtalan tipográfiai szabály egyidejű teljesítésével a fordító maga végzi, a dokumentum írója csak

²a korábban említett L^AT_EX-disztribúciókban megtalálható

preferenciáit jelezheti a fordító felé (olykor ez bosszúságot is okozhat, ilyenkor pl. a kép méretével lehet játszani).



4.1. ábra. Több képfájl beillesztése esetén térközőket is érdemes használni.

A táblázatok használatára a 4.1 táblázat mutat példát. A táblázatok formázásához hasznos tanácsokat találunk a booktabs csomag dokumentációjában.

Órajel	Frekvencia	Cél pin
CLKA	100 MHz	FPGA CLK0
CLKB	48 MHz	FPGA CLK1
CLKC	20 MHz	Processzor
CLKD	25 MHz	Ethernet chip
CLKE	72 MHz	FPGA CLK2
XBUF	20 MHz	FPGA CLK3

4.1. táblázat. Az órajel-generátor chip órajel-kimenetei.

4.3. Felsorolások és listák

Számozatlan felsorolásra mutat példát a jelenlegi bekezdés:

- *első bajusz*: ide lehetne írni az első elem kifejtését,
- *második bajusz*: ide lehetne írni a második elem kifejtését,

- *ez meg egy szakáll:* ide lehetne írni a harmadik elem kifejtését.

Számozott felsorolást is készíthetünk az alábbi módon:

1. *első bajusz:* ide lehetne írni az első elem kifejtését, és ez a kifejtés így néz ki, ha több sorosra sikeredik,
2. *második bajusz:* ide lehetne írni a második elem kifejtését,
3. *ez meg egy szakáll:* ide lehetne írni a harmadik elem kifejtését.

A felsorolásokban sorok végén vessző, az utolsó sor végén pedig pont a szokásos írásjel. Ez alól kivételt képezhet, ha az egyes elemek több teljes mondatot tartalmaznak.

Listákban a dolgozat szövegétől elkülönítendő kódrészleteket, programsorokat, pszeudo-kódokat jeleníthetünk meg (4.1. kódrészlet).

```
\begin{enumerate}
  \item \emph{első bajusz:} ide lehetne írni az első elem kifejtését,
    és ez a kifejtés így néz ki, ha több sorosra sikeredik,
  \item \emph{második bajusz:} ide lehetne írni a második elem kifejtését,
  \item \emph{ez meg egy szakáll:} ide lehetne írni a harmadik elem kifejtését.
\end{enumerate}
```

4.1. lista. A fenti számozott felsorolás L^AT_EX-forráskódja

A lista keretét, háttérszínét, egész stílusát megválaszthatjuk. Ráadásul különféle programnyelveket és a nyelveken belül kulcsszavakat is definiálhatunk, ha szükséges. Erről bővebbet a listings csomag hivatalos leírásában találhatunk.

4.4. Képletek

Ha egy formula nem túlságosan hosszú, és nem akarjuk hivatkozni a szövegből, mint például a $e^{i\pi} + 1 = 0$ képlet, *szövegközi képletként* szokás leírni. Csak, hogy másik példát is lássunk, az $U_i = -d\Phi/dt$ Faraday-törvény a $\text{rot } E = -\frac{dB}{dt}$ differenciális alakban adott Maxwell-egyenlet felületre vett integráljából vezethető le. Látható, hogy a L^AT_EX-fordító a sorközüket betartja, így a szöveg szedése esztétikus marad szövegközi képletek használata esetén is.

Képletek esetén az általános konvenció, hogy a kisbetűk skalárt, a kis félkövér betűk (**v**) oszlopvektort – és ennek megfelelően \mathbf{v}^T sorvektort – a kapitális félkövér betűk (**V**) mátrixot jelölnek. Ha ettől el szeretnénk térni, akkor az alkalmazni kívánt jelölésmódot célszerű külön alfejezetben definiálni. Ennek megfelelően, amennyiben **y** jelöli a mérések vektorát, **ϑ** a paraméterek vektorát és $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\vartheta}$ a paraméterekben lineáris modellt, akkor a *Least-Squares* értelemben optimális paraméterbecslő $\hat{\boldsymbol{\vartheta}}_{LS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ lesz.

Emellett kiemelt, sorszámozott képleteket is megadhatunk, ennél az equation és a eqnarray környezetek helyett a korszerűbb align környezet alkalmazását javasoljuk (több okból, különféle problémák elkerülése végett, amelyekre most nem térünk ki). Tehát

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (4.1)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \quad (4.2)$$

ahol **x** az állapotvektor, **y** a mérések vektora és **A**, **B** és **C** a rendszert leíró paraméter-mátrixok. Figyeljük meg, hogy a két egyenletben az egyenlőségjelek egymáshoz igazítva

jelennek meg, mivel a mindkettőt az & karakter előzi meg a kódban. Lehetőség van számozatlan kiemelt képlet használatára is, például

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu}, \\ \mathbf{y} &= \mathbf{Cx}.\end{aligned}$$

Mátrixok felírására az $\mathbf{Ax} = \mathbf{b}$ inhomogén lineáris egyenlet részletes kifejtésével mutatunk példát:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}. \quad (4.3)$$

A `\frac` utasítás hatékonyságát egy általános másodfokú tag átviteli függvényén keresztül mutatjuk be, azaz

$$W(s) = \frac{A}{1 + 2T\xi s + s^2T^2}. \quad (4.4)$$

A matematikai mód minden szimbólumának és képességének a bemutatására természetesen itt nincs lehetőség, de gyors referenciaként hatékonyan használhatók a következő linkek:

http://www.artofproblemsolving.com/LaTeX/AoPS_L_GuideSym.php,
<http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>,
<ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>.
 Ez pedig itt egy magyarázat, hogy miért érdemes align környezetet használni:
<http://texblog.net/latex-archive/math/eqnarray-align-environment/>.

4.5. Irodalmi hivatkozások

Egy \LaTeX dokumentumban az irodalmi hivatkozások definíciójának két módja van. Az egyik a `\thebibliography` környezet használata a dokumentum végén, az `\end{document}` lezárás előtt.

```
\begin{thebibliography}{9}

\bibitem{Lamport94} Leslie Lamport, \emph{\LaTeX: A Document Preparation System}.
Addison Wesley, Massachusetts, 2nd Edition, 1994.

\end{thebibliography}
```

Ezek után a dokumentumban a `\cite{Lamport94}` utasítással hivatkozhatunk a forrásra. A fenti megadás viszonylag kötetlen, a szerző maga formázza az irodalomjegyzéket (ami gyakran inkonzisztens eredményhez vezet).

Egy sokkal professzionálisabb módszer a $\text{BiB}\text{\TeX}$ használata, ezért ez a sablon is ezt támogatja. Ebben az esetben egy külön szöveges adatbázisban definiáljuk a forrásmunkákat, és egy külön stílusfájl határozza meg az irodalomjegyzék kinézetét. Ez, összhangban azzal, hogy külön formátumkonvenció határozza meg a folyóirat-, a könyv-, a konferenciacyikk- stb. hivatkozások kinézetét az irodalomjegyzékben (a sablon használata esetén ezzel nem is kell foglalkoznia a hallgatónak, de az eredményt célszerű ellenőrizni). felhasznált hivatkozások adatbázisa egy `.bib` kiterjesztésű szöveges fájl,

amelynek szerkezetét a A 4.2 kódrészlet demonstrálja. A forrásmunkák bevitelekor a sor végi vesszők külön figyelmet igényelnek, mert hiányuk a BiBTeX-fordító hibaüzenetét eredményezi. A forrásmunkákat típus szerinti kulcsszó vezeti be (@book könyv, @inproceedings konferenciakiadványban megjelent cikk, @article folyóiratban megjelent cikk, @techreport valamelyik egyetem gondozásában megjelent műszaki tanulmány, @manual műszaki dokumentáció esetén stb.). Nemcsak a megjelenés stílusa, de a kötelezően megadandó mezők is típusról-típusra változnak. Egy jól használható referencia a <http://en.wikipedia.org/wiki/BibTeX> oldalon található.

```
@book{Wettl04,
  author   = {Ferenc Wettl and Gyula Mayer and Péter Szabó},
  publisher = {Panem Könyvkiadó},
  title    = {\LaTeX-kézikönyv},
  year     = {2004},
}

@article{Candy86,
  author      = {James C. Candy},
  journaltitle = {{IEEE} Trans.\ on Communications},
  month       = {01},
  note        = {\doi{10.1109/TCOM.1986.1096432}},
  number      = {1},
  pages       = {72--76},
  title       = {Decimation for Sigma Delta Modulation},
  volume      = {34},
  year        = {1986},
}

@inproceedings{Lee87,
  author      = {Wai L. Lee and Charles G. Sodini},
  booktitle   = {Proc.\ of the IEEE International Symposium on Circuits and Systems},
  location    = {Philadelphia, PA, USA},
  month       = {05-4--7},
  pages       = {459--462},
  title       = {A Topology for Higher Order Interpolative Coders},
  vol         = {2},
  year        = {1987},
}

@thesis{KissPhD,
  author      = {Peter Kiss},
  institution = {Technical University of Timi\c{s}oara, Romania},
  month       = {04},
  title       = {Adaptive Digital Compensation of Analog Circuit Imperfections for Cascaded Delta-Sigma Analog-to-Digital Converters},
  type        = {phdthesis},
  year        = {2000},
}

@manual{Schreier00,
  author      = {Richard Schreier},
  month       = {01},
  note        = {\url{http://www.mathworks.com/matlabcentral/fileexchange/}},
  organization = {Oregon State University},
  title       = {The Delta-Sigma Toolbox v5.2},
  year        = {2000},
}

@misc{DipPortal,
  author      = {{Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar}},
  howpublished = {\url{http://diplomaterv.vik.bme.hu/}},
  title       = {Diplomaterv portál (2011. február 26.)},
}

@incollection{Mkrtychev:1997,
  author      = {Mkrtychev, Alexey},
  booktitle   = {Logical Foundations of Computer Science},
  doi         = {10.1007/3-540-63045-7_27},
  editor      = {Adian, Sergei and Nerode, Anil},
}
```

```

isbn      = {978-3-540-63045-6},
pages     = {266-275},
publisher = {Springer Berlin Heidelberg},
series    = {Lecture Notes in Computer Science},
title     = {Models for the logic of proofs},
url       = {http://dx.doi.org/10.1007/3-540-63045-7_27},
volume    = {1234},
year      = {1997},
}

```

4.2. lista. Példa szöveges irodalomjegyzék-adatbázisra BibTeX használata esetén.

A stílusfájl egy .sty kiterjesztésű fájl, de ezzel lényegében nem kell foglalkozni, mert vannak beépített stílusok, amelyek jól használhatók. Ez a sablon a BiBTeX-et használja, a hozzá tartozó adatbázisfájl a mybib.bib fájl. Megfigyelhető, hogy az irodalomjegyzéket a dokumentum végére (a `\end{document}` utasítás elé) beillesztett `\bibliography{mybib}` utasítással hozhatjuk létre, a stílusát pedig ugyanitt a `\bibliographystyle{plain}` utasítással adhatjuk meg. Ebben az esetben a plain előre definiált stílust használjuk (a sablonban is ezt állítottuk be). A plain stíluson kívül természetesen számtalan más előre definiált stílus is létezik. Mivel a .bib adatbázisban ezeket megadtuk, a BiBTeX-fordító is meg tudja különböztetni a szerzőt a címtől és a kiadótól, és ez alapján automatikusan generálódik az irodalomjegyzék a stílusfájl által meghatározott stílusban.

Az egyes forrásmunkákra a szövegből továbbra is a `\cite` paranccsal tudunk hivatkozni, így a 4.2. kódrészlet esetén a hivatkozások rendre `\cite{Wettl04}`, `\cite{Candy86}`, `\cite{Lee87}`, `\cite{KissPhD}`, `\cite{Schreirer00}`, `\cite{Mkrtychev:1997}` és `\cite{DipPortal}`. Az egyes forrásmunkák sorszáma az irodalomjegyzék bővítésekor változhat. Amennyiben az aktuális számhoz illeszkedő névelőt szeretnénk használni, használjuk az `\acite{}` parancsot.

Az irodalomjegyzékben alapértelmezésben csak azok a forrásmunkák jelennek meg, amelyekre található hivatkozás a szövegben, és ez így alapvetően helyes is, hiszen olyan forrásmunkákat nem illik az irodalomjegyzékbe írni, amelyekre nincs hivatkozás.

Mivel a fordítási folyamat során több lépésben oldódnak fel a szimbólumok, ezért gyakran többször is le kell fordítani a dokumentumot. Ilyenkor ez első 1-2 fordítás esetleg szimbólum-feloldásra vonatkozó figyelmeztető üzenettel zárul. Ha hibaüzenettel zárul bármelyik fordítás, akkor nincs értelme megismételni, hanem a hibát kell megkeresni. A .bib fájl megváltoztatáskor sokszor nincs hatása a változtatásnak azonnal, mivel nem mindig fut újra a BibTeX fordító. Ezért célszerű a változtatás után azt manuálisan is lefuttatni (TeXstudio esetén Tools/Bibliography).

Hogy a szövegbe ágyazott hivatkozások kinézetét demonstráljuk, itt most sorban meghivatkozunk a [13], [2], [6], [5], [9] és a [7]³ forrásmunkát, valamint az [1] weboldalt.

Megjegyzendő, hogy az ékezetes magyar betűket is tartalmazó .bib fájl az inputenc csomaggal betöltött latin2 betűkészlet miatt fordítható. Ugyanez a .bib fájl hibaüzenettel fordul egy olyan dokumentumban, ami nem tartalmazza a `\usepackage[latin2]{inputenc}` sort. Speciális igény esetén az irodalmi adatbázis általánosabb érvényűvé tehető, ha az ékezetes betűket speciális latex karakterekkel helyettesítjük a .bib fájlban, pl. á helyett `\'a`-t vagy ő helyett `\H{o}`-t írunk.

³Informatikai témában gyakran hivatkozunk cikkeket a Springer LNCS valamely kötetéből, ez a hivatkozás erre mutat egy helyes példát.

Irodalomhivatkozásokat célszerű először olyan szolgáltatásokban keresni, ahol jó minőségű bejegyzések találhatók (pl. ACM Digital Library,⁴ DBLP,⁵ IEEE Xplore,⁶ SpringerLink⁷) és csak ezek után használni kevésbé válogatott forrásokat (pl. Google Scholar⁸). A jó minőségű bejegyzéseket is érdemes megfelelően tisztítani.⁹ A sablon angol nyelvű változatában használt plainnat beállítás egyik sajátossága, hogy a cikkhez generált hivatkozás a cikk DOI-ját és URL-jét is tartalmazza, ami gyakran duplikátumhoz vezet – érdemes tehát a DOI-kat tartalmazó URL mezőket törölni.

4.6. A dolgozat szerkezete és a forrásfájlok

A diplomatersablonban a TeX fájlok két alkönyvtárban helyezkednek el. Az include könyvtárban azok szerepelnek, amiket tipikusan nem kell szerkesztenünk, ezek a sablon részei (pl. címlap). A content alkönyvtárban pedig a saját munkánkat helyezhetjük el. Itt érdemes az egyes fejezeteket külön TeX állományokba rakni.

A diplomatersablon (a kari irányelvek szerint) az alábbi fő fejezetekből áll:

1. 1 oldalas *tájékoztató* a szakdolgozat/diplomaterv szerkezetéről (include/guideline.tex), ami a végső dolgozatból törlendő,
2. *feladatkiírás* (include/project.tex), a dolgozat nyomtatott verziójában ennek a helyére kerül a tanszék által kiadott, a tanszékvezető által aláírt feladatkiírás, a dolgozat elektronikus verziójába pedig a feladatkiírás egyáltalán ne kerüljön bele, azt külön tölti fel a tanszék a diplomaterv-honlapra,
3. *címlap* (include/titlepage.tex),
4. *tartalomjegyzék* (thesis.tex),
5. a diplomatervező *nyilatkozata* az önálló munkáról (include/declaration.tex),
6. 1-2 oldalas tartalmi *összefoglaló* magyarul és angolul, illetve elkészíthető még további nyelveken is (content/abstract.tex),
7. *bevezetés*: a feladat értelmezése, a tervezés célja, a feladat indokoltsága, a diplomaterv felépítésének rövid összefoglalása (content/introduction.tex),
8. sorszámmal ellátott *fejezetek*: a feladatkiírás pontosítása és részletes elemzése, előzmények (irodalomkutatás, hasonló alkotások), az ezekből levonható következtetések, a tervezés részletes leírása, a döntési lehetőségek értékelése és a választott megoldások indoklása, a megtervezett műszaki alkotás értékelése, kritikai elemzése, továbbfejlesztési lehetőségek,
9. esetleges *köszönetnyilvánítások* (content/acknowledgement.tex),
10. részletes és pontos *irodalomjegyzék* (ez a sablon esetében automatikusan generálódik a thesis.tex fájlban elhelyezett \bibliography utasítás hatására, a 4.5. szakaszban leírtak szerint),

⁴<https://dl.acm.org/>

⁵<http://dblp.uni-trier.de/>

⁶<http://ieeexplore.ieee.org/>

⁷<https://link.springer.com/>

⁸<http://scholar.google.com/>

⁹<https://github.com/FTSRG/cheat-sheets/wiki/BibTeX-Fixing-entries-from-common-sources>

11. függelék (content/appendices.tex).

A sablonban a fejezetek a thesis.tex fájlba vannak beillesztve \include utasítások segítségével. Lehetőség van arra, hogy csak az éppen szerkesztés alatt álló .tex fájlt fordítsuk le, ezzel lerövidítve a fordítási folyamatot. Ezt a lehetőséget az alábbi kódrészlet biztosítja a thesis.tex fájlban.

```
\includeonly{
  guideline,%
  project,%
  titlepage,%
  declaration,%
  abstract,%
  introduction,%
  chapter1,%
  chapter2,%
  chapter3,%
  acknowledgement,%
  appendices,%
}
```

Ha az alábbi kódrészletben az egyes sorokat a % szimbólummal kikommentezzük, akkor a megfelelő .tex fájl nem fordul le. Az oldalszámok és a tartalomjegyek természetesen csak akkor billennek helyre, ha a teljes dokumentumot lefordítjuk.

4.7. Alapadatok megadása

A diplomaterv alapadatait (cím, szerző, konzulens, konzulens titulusa) a `thesis.tex` fájlban lehet megadni.

4.8. Új fejezet írása

A főfejezetek külön `content` könyvtárban foglalnak helyet. A sablonhoz 3 fejezet készült. További főfejezeteket úgy hozhatunk létre, ha új `TEX` fájlt készítünk a fejezet számára, és a `thesis.tex` fájlban, a `\include` és `\includeonly` utasítások argumentumába felvesszük az új `.tex` fájl nevét.

4.9. Definíciók, tételek, példák

Definíció 1 (Fluxuskondenzátor térerőssége). Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. ■

Példa 1. *Példa egy példára. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.*

Tétel 1 (Kovács tétele). Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ■

Köszönetnyilvánítás

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

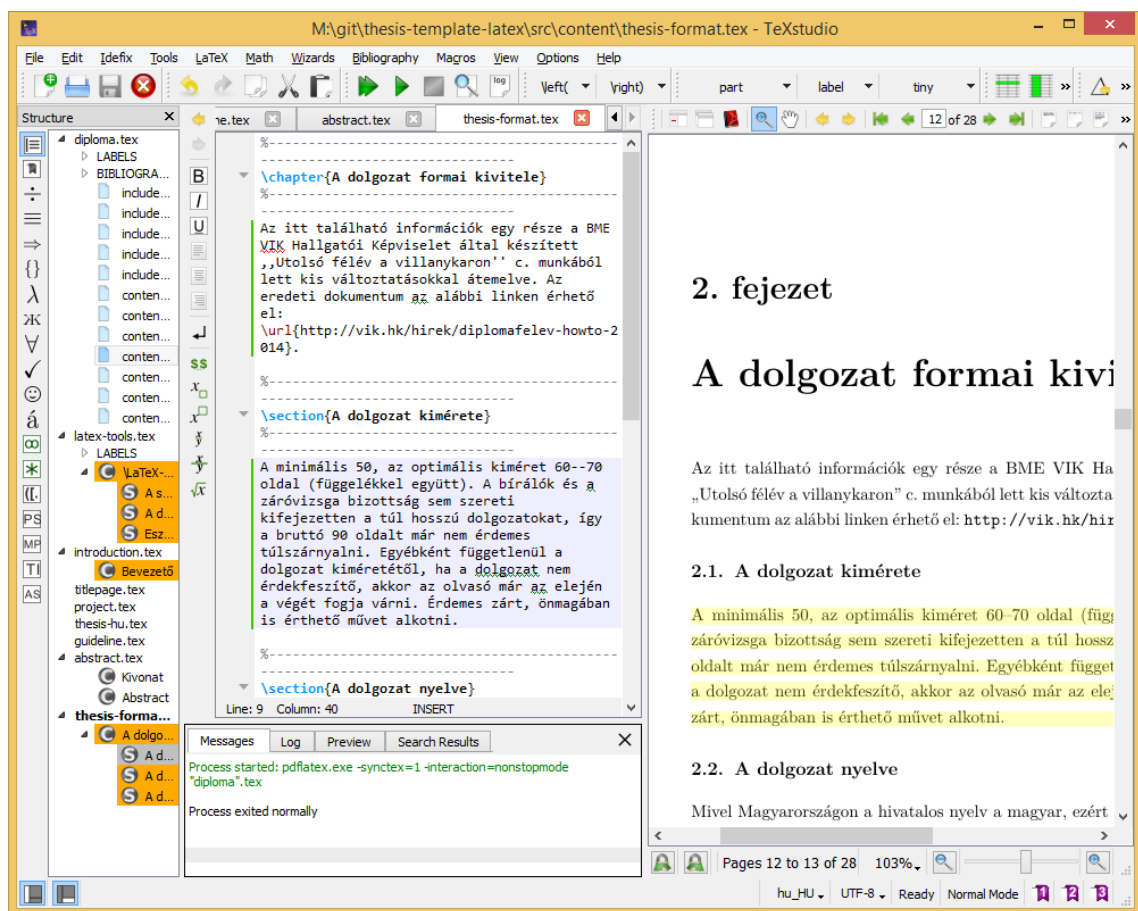
Irodalomjegyzék

- [1] Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar: Diplomaterv portál (2011. február 26.). <http://diplomaterv.vik.bme.hu/>.
- [2] James C. Candy: Decimation for sigma delta modulation. *IEEE Trans. on Communications*, 34. évf. (1986. 01) 1. sz., 72–76. p. DOI: 10.1109/TCOM.1986.1096432.
- [3] Droid Chef: Flutter vs jetpack compose: The battle of the century. *Ishan Khanna*, 2023. 09. <https://blog.droidchef.dev/flutter-vs-jetpack-compose-the-battle-of-the-century/>.
- [4] Gábor Jeney: Hogyan néz ki egy igényes dokumentum? Néhány szóban az alapvető tipográfiai szabályokról, 2014. <http://www.mcl.hu/~jeneyg/kinezet.pdf>.
- [5] Peter Kiss: Adaptive digital compensation of analog circuit imperfections for cascaded delta-sigma analog-to-digital converters, 2000. 04.
- [6] Wai L. Lee–Charles G. Sodini: A topology for higher order interpolative coders. In *Proc. of the IEEE International Symposium on Circuits and Systems* (konferenciaanyag). 1987. 4-7 05., 459–462. p.
- [7] Alexey Mkrtychev: Models for the logic of proofs. In Sergei Adian–Anil Nerode (szerk.): *Logical Foundations of Computer Science*. Lecture Notes in Computer Science sorozat, 1234. köt. 1997, Springer Berlin Heidelberg, 266–275. p. ISBN 978-3-540-63045-6. URL http://dx.doi.org/10.1007/3-540-63045-7_27.
- [8] Elvira Mustafina: Compose multiplatform 1.7.0 released. *JetBrains Blog*, 2024. 10. <https://blog.jetbrains.com/kotlin/2024/10/compose-multiplatform-1-7-0-released/>.
- [9] Richard Schreier: *The Delta-Sigma Toolbox v5.2*. Oregon State University, 2000. 01. <http://www.mathworks.com/matlabcentral/fileexchange/>.
- [10] JetBrains Documentation Team: Popular cross-platform app development frameworks. *Kotlin Multiplatform Documentation*, 2024. 10. <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html#popular-cross-platform-app-development-frameworks>.
- [11] JetBrains Kotlin Team: Kotlin multiplatform goes stable. *Kotlin Blog*, 2023. 11. <https://blog.jetbrains.com/kotlin/2023/11/kotlin-multiplatform-stable/>.
- [12] Kotlin Documentation Team: Update to the new release. *Kotlin Multiplatform Plugin Releases*, 2024. 10. <https://kotlinlang.org/docs/multiplatform-plugin-releases.html#update-to-the-new-release>.

- [13] Ferenc Wettl–Gyula Mayer–Péter Szabó: *L^AT_EX kézikönyv*. 2004, Panem Könyvkiadó.

Függelék

F.1. A TeXstudio felülete



F.1.1. ábra. A TeXstudio L^AT_EX-szerkesztő.

F.2. Válasz az „Élet, a világmindenség, meg minden” kérdésre

A Pitagorasz-tételből levezetve

$$c^2 = a^2 + b^2 = 42. \quad (\text{F.2.1})$$

A Faraday-indukciós törvényből levezetve

$$\text{rot } E = -\frac{dB}{dt} \quad \longrightarrow \quad U_i = \oint_{\mathbf{L}} \mathbf{E} d\mathbf{l} = -\frac{d}{dt} \int_A \mathbf{B} d\mathbf{a} = 42. \quad (\text{F.2.2})$$