



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Pour Benedek

KORMÁNYOSZLOP-MOZGATÓ VEZÉRLŐEGYSÉG HARDVERANALÍZISE ÉS TESZTELÉSE

Szakdolgozat

TANSZÉKI KONZULENS

Dr. Sujbert László

KÜLSŐ KONZULENS

Dr. Balogh András

BUDAPEST, 2013

Tartalomjegyzék

Tartalomjegyzék	2
Összefoglaló	5
Abstract.....	6
1 Bevezetés	7
2 A tesztelő szoftver	10
2.1 Bevezetés	10
2.2 Egy lehetséges megoldás, az XCP protokoll	10
2.2.1 Az XCP protokoll kialakulása	10
2.2.2 Működési elv.....	11
2.2.3 Értékelés.....	11
2.3 Az elkészített megoldás	12
2.3.1 A LIN kommunikációról röviden	12
2.3.2 Tervezés	13
2.3.3 Tesztelő modul.....	15
2.3.3.1 A modul aktiválása, inaktiválása	15
2.3.3.2 Változók felülírása	16
2.3.3.3 Inicializálás	17
3 Hardver perifériák tervezése	18
3.1 Kormányoszlop pozíció szenzor	18
3.1.1 Inkrementális pozíciószenzorok	18
3.1.2 HALL-szenzoros megoldás	18
3.1.2.1 Szenzor, analóg-digitális bemenet	18
3.1.2.2 A kapcsolás	20
3.1.2.3 Áramkör-szimuláció	23
3.2 Feszültségmérés	25
3.2.1 A kapcsolás	26
3.2.2 Áramkör-szimuláció	27
4 Termikus modell meghatározása	30
4.1 Irodalomkutatás	30
4.2 Saját megoldás	32
4.2.1 A mérés elve	32

4.2.2 Mérési elrendezés	33
4.2.3 Mérés, modellezés	35
4.2.4 Hitelesítés.....	38
5 Összegzés.....	41
Irodalomjegyzék.....	43
Függelék.....	44

HALLGATÓI NYILATKOZAT

Alulírott *Pour Benedek*, 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 hitelesített 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.

Kelt: Budapest, 2013. 12. 22.

.....
Pour Benedek
hallgató

Összefoglaló

A mai autókban megtalálható elektronikus vezérlőegységek nemcsak a motorirányítás és járműdinamika területén kapnak nagy szerepet, hanem a felhasználót kiszolgáló kényelmi funkciókat is ellátják. Szakdolgozatomban egy ilyen funkció, egy elektronikus mozgatható kormányoszlop vezérlőegységének néhány szükséges tervezési lépését mutatom be.

Először is megtervezek egy, a berendezés tesztelésére alkalmas szoftvert, amely képes a mikrokontrollerben található néhány működést befolyásoló változónak új értéket adni, illetve néhány fontos változót a felhasználó számára megjeleníteni. A tesztelő programban létrehozok egy felhasználói parancsot, amely képes a tesztelt eszköz inicializáló módját elindítani, majd annak sikeres állapotát visszajelezni.

A következő fejezetben a mikrovezérlőhöz tartozó két külön hardver perifériaegység tervezési, illetve méretezési lépéseit mutatom be. Az áramkörök működésének helyességét szimulációs program segítségével elvégzett analízisek eredménye alapján igazolom.

Végül a mikrovezérlő egységben található meghajtó tranzisztorok termikus modelljének meghatározási lépéseit mutatom be. Az identifikáció helyességét egy hőkamerás mérés segítségével igazolom.

Abstract

In today's cars the electronic control units are playing a key role not only in engine control and vehicle dynamics, but also in comfort functions used by the passengers. In my thesis I introduce some necessary design steps of the control unit one of these comfort functions, the electronically adjustable steering column.

Firstly, I design a software to test the equipment. The software is able to define variables and show the most important ones to the user. In the testing software I create a user command, which is able to start the initialization mode of the tested device, and then to feedback its success status.

In the next chapter I introduce the design and sizing steps of two different hardware peripherals of the microcontroller. To verify the right functionality of the circuits I am using the results of a simulation program's analysis.

In the end I show you the steps of defining the control transistors' thermal model in the microcontroller unit. I verify the right identification with a thermographic measurement.

1 Bevezetés

Az elektronika, illetve az elektronikai technológia fejlődése nagy hatással van az autóiipari megoldások változására, alkalmazások bővülésére. Az 1900-as évek elején használt olcsó autókban szinte csak a belső égésű motor működéséhez elengedhetetlen gyújtásrendszer volt elektronikus. Ahogy egyre inkább elterjedt az autó hétköznapi, közlekedésbeli használata, a Bosch, mint német alkatrészszállító és fejlesztő óriás cég számos újítást vezetett be az autóiiparban. Megjelentek az elektromos fényszórók, indítómotorok, elektromos ablaktörlők, később az 1950-es évekig üzemelő elektromos irányítású, de még karos irányjelzők is. A második világháborúban kialakuló, országok közötti versenyhelyzet számos új fejlesztést szült, ilyen például a vadászgépekben alkalmazott üzemanyag-befecskendezés. A különböző pozícióban közlekedő és óriási gyorsulásokat elérő gépek üzemanyag-ellátása ugyanis a hagyományos porlasztóval nem volt biztonságosan megoldható. A háború után először az autóversenyezésben, később az általános felhasználóknak készült járművekben is megjelent az először csak mechanikus elemeket, később elektronikus vezérlést alkalmazó befecskendezés. [1]

A félvezető technológia megjelenése az autóiiparban is hatalmas hódításokat végzett, ugyanis a kis területen elhelyezkedő komplex elektronika nagymértékben leegyszerűsítette, megkönnyítette a vezérlési feladatokat. Az addigi mechanikus megoldások kezdtek elektronikus vezérlést kapni, majd számos helyen megjelentek az elektromágnesek, villanymotorok, szervomotorok is. Az egyre gyorsabb számítási teljesítményű berendezések lehetővé tették az elektronika megjelenését az autóiipari biztonságtechnikában. 1970. december 12-én bemutatták a Mercedes-Benz első blokkolásgátlóját, amely fogazott jeladókkal figyelte a kerekek megcsúszásának mértékét, és ennek függvényében változtatta a pillanatnyi fékerőt. Később megszületett a kipörgésgátló, illetve manapság már minden autóban megtalálható a fedélzeti számítógép is, ami egyre több információt biztosít a felhasználó számára járművéről. [1]

Az elektronikus megoldások elterjedése az autóiiparban számos helyen hatékony következményeket vont maga után, ugyanis a mechanikus áttételeket kis energiájú elektromos jelek váltották fel, ezzel csökkentve a súrlódást, és egyidejűleg a fogyasztást. Rendszeres kenés ellenére a mechanikus alkatrészek folyamatos kopása folyamatos utánállítást, végül cserét igényel. Számos esetben az elektronikus

berendezések kiküszöbölik ezt a problémát. Tekintsük példának az autó egyik alapvető berendezését, a gyújtást. Az első igazán elterjedt megoldás a megszakító-kalapácsos gyújtás jeladó. Itt egy excentrikus tengely emeli meg a kalapácsot, ezzel megszakítást létrehozva két érintkező között. Csupa mechanikus alkatrész, melyeknek kopása bizonytalanságot okoz, folyamatos ellenőrzést, beállítást igényelt. Sokkal megbízhatóbb azonban a később elterjedt elektronikus gyújtás. Itt egy szenzor ad pontos információt a főtengely pillanatnyi helyzetéről, ez általában mágneses vagy induktív jelforrás, amely a használat során nem kopik, mindig pontos marad. Nem is beszélve arról, hogy a mikrokontrolleres gyújtásvezérlések az egyre növekvő számítási teljesítménynek köszönhetően egyre precízebb gyújtást biztosítanak a pillanatnyi motortulajdonságok figyelembevételével, ezzel is növelve az égés tökéletességét.

A hatékony alkalmazások mellett megjelentek a felhasználók kényelmét szolgáló funkciók is. Ilyenek például az elektromos ablakemelő, elektromos csomagtartófedél, tükör, ülésmozgatás, elektromos vezérlésű klímaberendezés, rádió, hangrendszerek és még számos egyéb alkalmazás. Egy mai autóban ezeknek már nagy része megtalálható az alapfelszereltségben.

Szakedolgozatom kapcsán egy ilyen kényelmi funkcióval, elektronikusan mozgatható kormányoszloppal foglalkoztam. A biztonságos és kényelmes vezetés érdekében szükséges, hogy a sofőr saját testméreteihez tudja állítani környezetét. A mai autókban lehetőség van az ülések, illetve a kormányoszlop helyzetének pozicionálására. Egy luxusautóban már elképzelhetetlen, hogy ezeket a beállításokat kézi erővel kelljen elvégezni, így egyre nagyobb szerepet kapnak az elektronikus vezérlések. A mikroszámítógépes technikának köszönhetően egy ilyen eszköz köré már számos kényelmi funkciót lehet csatolni. Ma már léteznek olyan megoldások, amelyek felismerik, hogy melyik sofőr közeledik az autóhoz, és miután kényelmesen beszállt, a jármű automatikusan aktiválja személyre szabott beállításait, felemeli és pozicionálja az ülést, illetve kezébe adja a kormányt.

Szakedolgozatomban egy ilyen rendszer vezérlőegységének néhány szükséges tervezési lépését mutatom be, melyeket a ThyssenKrupp Presta Hungary Kft-nél végeztem el. Az általam elemzett három különböző típusú feladat által betekintést nyerhetünk egy ilyen kisebb berendezés elkészítésének komplexitásába, ahol a hardveres és szoftveres fejlesztés együttesen történik.

Ahhoz, hogy egy-egy hardveres újítás mellé tökéletes szoftvert írassunk, szükséges, hogy a mikrokontroller néhány belső paraméterének értékét a rendszer működése közben tudjuk átírni. Ez által lehetőség nyílik az egyes változtatások a működő rendszerre tett hatásának valós idejű vizsgálatára, tesztelésére. Az így kapott eredményeket követve lehetőség nyílik a mikrovezérlőn futó program tökéletesítésére. Első lépésben tehát egy olyan számítógépre írt szoftver létrehozását mutatom be, amely képes ezt a valós idejű, vezérlőegység mikrokontrollerének futása közben történő változó-beállítást megvalósítani.

A szoftveres fejlesztés mellett a következő fejezetben bemutatok két hardver perifériaegység tervezési és méretezési lépéseit. A teljes kormányoszlop-mozgató berendezés két fő modulja a mozgatót végző motorok, illetve a vezérlőegység központja, a mikrokontroller. A szakdolgozatomban bemutatott két periféria egység ezen két modul közti kapcsolatot biztosítja.

A vezérlőegység fejlesztéséhez nélkülözhetetlen továbbá a felhasznált alkatrészek fizikai paramétereinek pontos ismerete. Szakdolgozatomban a kormányoszlop-mozgató motorok vezérlését ellátó térvezérlésű tranzisztorok (Field Effect Transistor, FET) termikus modelljének meghatározását mutatom be. Ennek pontos ismerete mind a hardver tervezését, mind a szoftver működését meghatározza.

2 A tesztelő szoftver

2.1 Bevezetés

Egy autóiipari elektronikus vezérlőegység fejlesztése soha nem a gépjárműben kezdődik. Egyrészt azért, mert eszközünk kezdetben még nem felel meg a specifikációknak, másrészt számos olyan üzemállapotot kell kipróbálni, amely egy autóban az egység közvetlen környezete miatt nem lehetséges. Így általában munkánk egy fejlesztői asztalra kerül, ahol azonban sokszor nem áll rendelkezésünkre az összes szükséges periféria, vagy egyszerűen nincs is szükségünk a teljes komplex eszközkészletre, így egy-két rendszert saját magunknak kell létrehozni, modellezni.

Szakedolgozatomat megelőző szakmai gyakorlatom során elkészítettem a kormányoszlop-mozgató berendezés számára szükséges irányítóegység modelljét, amely a járműben is elvégezhető alapfunkciók parancsait képes biztosítani a vezérlőegység számára. Ez kezdetben elegendő volt, de egyes további fejlesztésekhez azonban szükség volt a mikrokontroller néhány belső változójának lekérdezésére, illetve új értékek megadására. Ezen probléma megoldásának bemutatására kerül sor ebben a fejezetben.

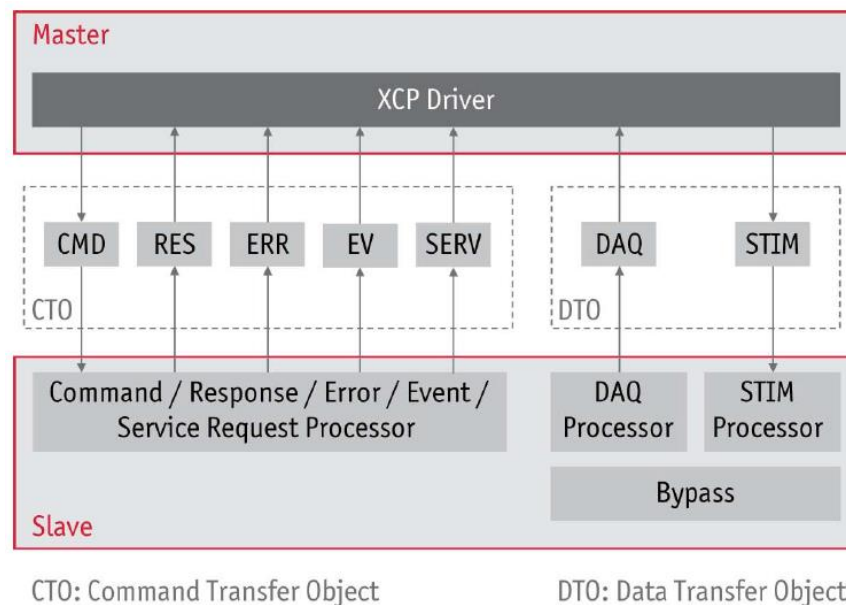
2.2 Egy lehetséges megoldás, az XCP protokoll

2.2.1 Az XCP protokoll kialakulása

A technikusok és fejlesztő mérnökök mindig is nagy hasznát vették azoknak a szabványoknak, amelyek lehetővé tették számukra a mikrokontrollerek belső változóinak, illetve memóriaterületének rugalmas írását és olvasását. Kezdetben az uralkodó autóiipari kommunikációs protokoll a CAN (Controller Area Network) volt, így az 1990-ben megjelent, rugalmas periféria kezelést biztosító szabvány, a CAN Calibration Protocol (CCP) teljes egészében erre a kommunikációra épült. Később a CAN mellett megjelentek egyéb autóiipari kommunikációs szabványok is, mint például a LIN (Local Interconnect Network) vagy a FlexRay. Mivel a CCP a CAN protokollra volt korlátozva, létrejött egy új szabvány, a Universal Measurement and Calibration Protocol (XCP), amely már számos létező kommunikációs protokollt támogat. [2]

2.2.2 Működési elv

Az XCP egy központi egység és akár több perifériaegység közötti kommunikációt képes biztosítani. A kicserélendő információ szabványban meghatározott üzeneteket tartalmaz, melyek vázlatos áttekintését az 2.1. ábra segíti. Témájuk szerint két külön csoportra oszlanak, „Parancsátviteli mezőre” (Command Transfer Object, CTO) és „Adatátviteli mezőre” (Data Transfer Object, DTO). A parancsátviteli mező tartalmazza a kiadott utasításokat (Command, CMD), az ezekre érkező válaszokat (Response, RES), a hibaüzeneteket (Error, ERR), jelentéseket a perifériában lezajló eseményekről (Event, EV), illetve a szükséges kiszolgálási igényeket (Service Request Processor, SERV). Az adatgyűjtés (Data Acquisition, DAQ), illetve az adatszolgáltatás (Stimulation, STIM) az adatátviteli mezőben zajlik. [2]



2.1. ábra Az XCP üzeneteinek csoportosítása

Az autóiari kommunikációról tudni kell, hogy működése általában ciklikus szervezésű. Ez annyit jelent, hogy az egyes különböző azonosítóval rendelkező üzenetek sorozata egy meghatározott ciklusidő elteltével ismétlődik. Egy ciklus felépítésében mindig megtalálható a diagnosztikai üzeneteknek fenntartott időszáv. Az XCP kommunikáció üzenetei is itt kapnak helyet.

2.2.3 Értékelés

Az XCP saját komplexitásával egy nagyon rugalmas, könnyen használható, közismert fejlesztői protokoll. Hatalmas előnye, hogy az autóiari vezérlőegység

hardver szintjén nincs szükség külön perifériára, mivel a kommunikáció az összes autóiipari szabványt támogatja, és teljes egészében az amúgy is használt buszon, periférián zajlik.

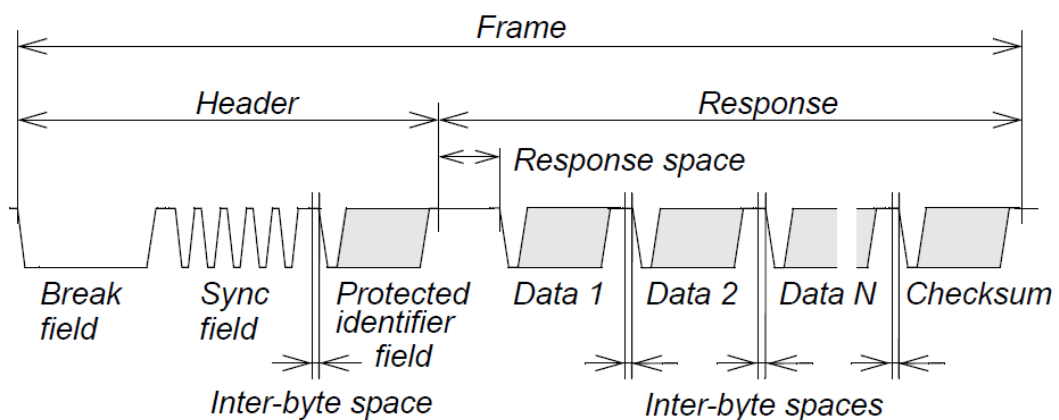
A felhasználói oldala viszont egy komplex szoftvert igényel, mint például a CANape, amelyik első volt az XCP protokollt támogató eszközök között. A fejlesztés során ilyen szoftver nem állt rendelkezésre, így egy saját megoldás létrehozására volt szükség.

2.3 Az elkészített megoldás

2.3.1 A LIN kommunikációról röviden

Mivel a fejlesztendő eszköz egy kényelmi berendezés, ezért a vezérlőegységgel történő kommunikáció a szabványban is megadott, illetve az autóiiparban ilyen célra használatos LIN protokollra épül. Ez egy nagyon egyszerű, kétvezetékes, digitális, soros, adatsomagokból álló kommunikációs szabvány. Az adatok a buszon megjelenő alacsony és magas jelszintekből állnak.

A 2.2. ábrán látható egy adatsomag (Frame), melynek felépítése a következő. A központi egység egy fejléctet (Header) küld ki a buszon, ez egy szünet- és szinkronmezőből áll (Break field, Sync field), majd egy üzenetazonosító (Protected identifier field) következik. Ezután az azonosító által megszólított periféria, vagy akár maga a központi egység, az azonosítóban meghatározott számú adatot (Data), majd egy ellenőrző mezőt (Checksum) küld a buszra. [3]



2.2. ábra A LIN keret felépítése

A szakmai gyakorlatom során elkészített vezérlőszoftvert a VN1611-es típusú Vector autóipari kommunikációs interfészhez tartozó, C# nyelvű, előre megírt felhasználói függvények segítségével valósítottam meg. Ez egy grafikus felülettel rendelkező program, amelynek kezelőfelülete tartalmazza a járműben megtalálható kezelőszerveket. Ezek segítségével a kommunikáció pontos paramétereinek ismerete nélkül, csupán néhány kattintással lehet vezérelni a fejlesztendő eszközünket. A LIN kommunikáció fizikai rétegének pontos működését, időzítését a Vector interfész biztosítja, melynek használata az egyes rétegek pontos megértése és felkonfigurálása után már nagyon kényelmessé teszi a további munkálatokat.

2.3.2 Tervezés

Saját megoldásom az volt, hogy az XCP protokollhoz hasonlóan a teszteléshez szükséges adatok áramlását a már meglévő LIN kommunikáció ciklusában biztosítom. Ez nagyon kézenfekvő, mert nincs szükség új hardver periféria létrehozására, a szükséges bővítéseket szoftveresen lehet elvégezni. A feladat alapja kitalálni, hogy a meglévő adatforgalom mellé hogyan lehet egyszerűen beépíteni a szükséges kicserélendő információt, valamint hogy a szükséges adatok elférnek-e mind a fejlesztendő eszköz specifikációjában megadott ciklusban.

Tehát elsősorban összegezni kell, hogy milyen értékek módosíthatóságára, kiolvashatóságára lesz szükség, illetve azok mekkora területen férnek el. A buszon kétféle üzenet keret van, az egyiket a központi egység, a másikat a perifériaegység küldi ki az üzenetazonosító után. Ez alapján a kicserélendő információt két csoportra osztom.

Az 2.1. táblázatban a tesztelő program, a 2.2. táblázatban a mikrokontroller által kiküldendő adatok láthatóak.

Küldendő adatok				
Változó	Leírás	Maximális érték	Mérték-egység	bit-szám
PWM kitöltési tényező	Az impulzusmodulált jelek maximális kitöltési tényezőjét meghatározó változó	1200	számláló egység	8
Motor sebesség	A mozgató motorok tengelyének maximális frekvenciája	nincs adat	Hz	8
Sebességnövelés	Megadja, hogy a motorok indulásánál milliszekundumonként hány egységet változzon az impulzusmodulált jel kitöltési tényezője	nincs adat	számláló egység / ms	8
Sebességcsökkentés	Megadja, hogy a motorok megállásánál milliszekundumonként hány egységet változzon az impulzusmodulált jel kitöltési tényezője	nincs adat	számláló egység / ms	8
Blokkolási áram	Megadja, hogy egy motor felütközéskor maximálisan mennyi áramot vehet fel	500	ADC egység	9
Lágymegállás poz	Megadja, hogy a kormányoszlop a végérték előtt hol álljon meg	nincs adat	pozíció-számláló	8

2.1. táblázat A küldendő adatok

Összeszámolható tehát, hogy a számítógépes tesztelő program által elküldendő üzenetek összes információjához szükséges adatmező 49 bit. A feladatomban használt LIN keretek adatmezője 8 bájtból áll, így az egy üzenet által küldhető bitek száma 64. Ebből következik, hogy a küldendő adatok elférnek egy keretben.

Fogadandó adatok				
Változó	Leírás	Maximális érték	Mérték-egység	bit-szám
Horizontális pozíció	A kormányoszlop horizontális elmozdulását adja meg	Specifikációban kiküldve		
Vertikális pozíció	A kormányoszlop vertikális elmozdulását adja meg	Specifikációban kiküldve		
Motor sebesség	A kormányoszlopot éppen mozgató motor sebessége (egyszerre csak egy motor működik)	300	Hz	9
Tápfeszültség	A vezérlőegységet ellátó feszültség értéke	4095	ADC egység	12
Motor feszültség	Az éppen működő motor kapcsain eső feszültség	4095	ADC egység	12
Motor áram	Az éppen működő motoron eső áram	2614	ADC egység	12
Processzor üzemmód	A processzor pillanatnyi működési állapota	11	db	4
Processzor hőmérséklet	A processzor hőmérséklete	4095	ADC egység	12
FET1, FET2, FET3 hőmérséklet	A meghajtó, páros tokozású FET hidak hőmérsékletei.	185	°C	3 x 8
Horizontális motor hőmérséklet	A horizontális irányú mozgatást biztosító motor hőmérséklete	180	°C	8
Vertikális motor hőmérséklet	A vertikális irányú mozgatást biztosító motor hőmérséklete	180	°C	8

2.2. táblázat A fogadandó adatok

Összeszámolható tehát, hogy a mikrokontroller által elküldendő üzenetek összes információjához szükséges adatmező 101 bit. Ebből következik, hogy a küldendő adatok elférnek két keretben.

Az összegzés eredménye azt mutatja, hogy bőven van hely a szükséges adatoknak. A további bővítési lehetőségeket szem előtt tartva takarékosan össze kell fűzni őket, és meghatározni, hogy melyik üzenet hol kapjon helyet a ciklusban. Ezt mutatja be a 2.3. táblázat.

VAR_CTR_LIN

7		6		5		4		3		2		1		0	
Softstop pos		Block current time out				Ramp down		Ramp up		Motor speed		Init flag		PWM duty cycle	
												bit 7		bit5	
														bit0	
														0: init req	

DPLY_KEY_LIN

7		6		5		4		3		2		1		0	
init stop		Proc mode		Processor temp		Motor current		Motor voltage		Battery voltage		Version number			
bit 5		bit 4...0		bit7		bit4		bit3		bit0		bit7		bit4	
														bit3	
														bit0	
														0: init end	

DPLY2_KEY_LIN

7		6		5		4		3		2		1		0	
Temp FET2 High		Temp FET2 Low		Desat flags		Error flags		Motor speed							
				bit3...0		bit 5		bit 0							

DPLY3_KEY_LIN

7		6		5		4		3		2		1		0	
Temp Motor Vert		Temp Motor Horiz													

TEST_KEY_LIN

7		6		5		4		3		2		1		0	

2.3. táblázat Az elkészített keretek

2.3.3 Tesztelő modul

Mivel a mikrokontroller specifikációjában nincs leírva, hogy a belső változók módosítását is biztosítani kell, ezért a követelmény alapján elkészített programokat mind a központi irányító egységben, mind a mikrovezérlőben egy új, tesztelő üzemmóddal kell kibővíteni. Fontos, hogy a mikrokontrollerbe írt programot egyaránt lehessen a valós környezetben és tesztelve is használni, ezért szükséges az egyes üzemmódok teljes elkülönítése.

2.3.3.1 A modul aktiválása, inaktiválása

Mivel a feladathoz megkapott, kommunikációt leíró specifikáció kizárólag a kormányoszlop-mozgató berendezés üzeneteit tartalmazza, ezért a tesztelő üzemmódba váltást nagyon speciálisan kell megvalósítani, nehogy valós környezetbe helyezve egy

nem erre a célra szolgáló üzenet aktiválhassa azt, és ezzel zavart okozhasson az eszköz működésében. A probléma elkerülésére a következő megoldást dolgoztam ki.

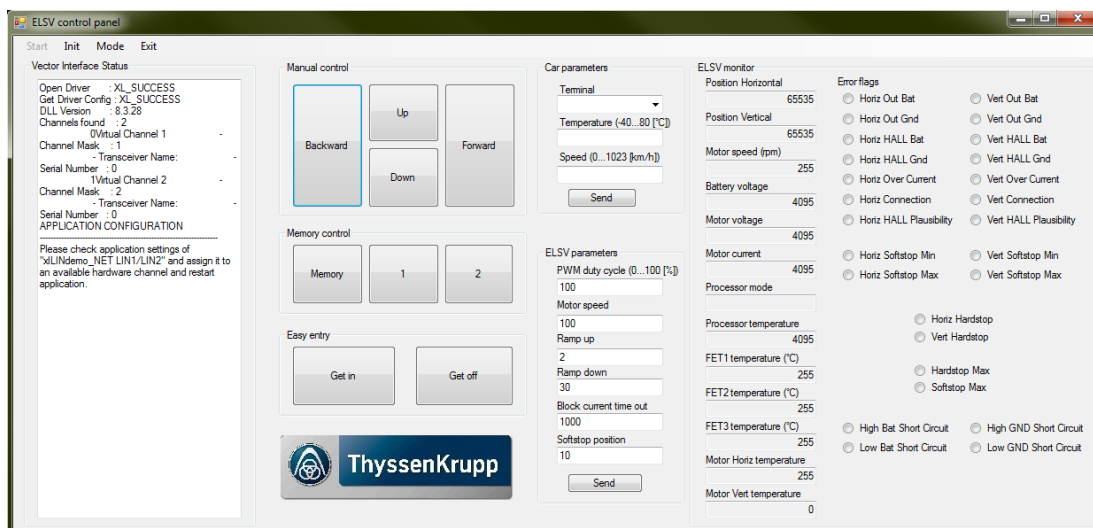
Tesztmódba való váltáskor a központi egység a kommunikációs ciklust megszakítva beékel egy általam megadott azonosítójú üzenetet, amelynek adatmezeje egy kódot tartalmaz. Mivel a kommunikációnak nincs konkrét időzítése, a közbeékelten nem jelent semmiféle problémát. A kormányoszlop vezérlőegységében futó programot fel kell készíteni az ilyen azonosítójú üzenet fogadására. Vétel után ellenőrizni kell a beérkezett kódot. Amennyiben azonosságot talál, elindítja a saját teszt módját, ami képes kezelni az általam létrehozott, már a kommunikációs ciklusba beillesztett speciális üzeneteket.

A tesztmódból való kilépést hasonló módon határoztam meg. Ismét kiadom a tesztmód azonosítójával rendelkező üzenetet, viszont ilyenkor az adatmezőben lévő kód már hibásan kerül kiküldésre. Ekkor a mikrokontroller már nem talál azonosságot, így inaktiválja a teszt üzemmódot.

Ez a megoldás szinte lehetetlené teszi, hogy a vezérlőegység csupán véletlenségből tesztelő üzemmódba váltsa az általam nem ismert egyéb lehetséges, buszon megjelenő üzenetek hatására. Ugyanis az azonosítónak és a teljes nyolcbájtos kódnak is pontosan meg kell egyeznie az általam definiált értékekkel.

2.3.3.2 Változók felülírása

A tesztelő modul lényege a belső változók új értékekkel való feltöltése. Amikor a felhasználó a 2.3. ábrán látható grafikus felület menüsorában a teszt üzemmódot választja, az új mód aktiválása után a kommunikációs ciklus kibővül az új információt hordozó üzenetekkel. Ilyenkor az üzenetek egy előre beállított értéket tartalmaznak. Ezzel egyidejűleg egy újabb elemmel bővül a vezérlőablak, ahol lehetőség van az egyes paraméterek értékének megadására. A lista alatt egy küldés gomb található, amelyre kattintva az új értékekkel feltöltött változók módosítják a saját üzenetüket, így periodikusan elküldésre kerülnek. A mikrokontroller feldolgozza az észlelt különbségeket, majd betölti a megfelelő változókba.

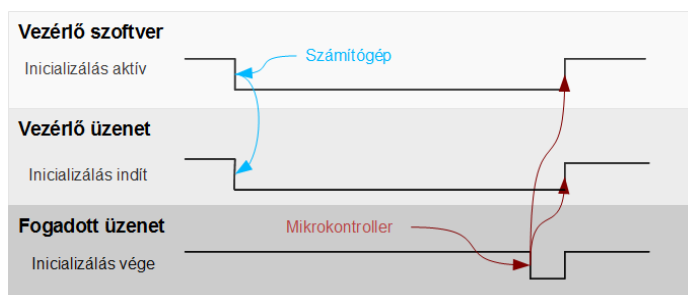


2.3. ábra A grafikus kezelőpanel

2.3.3.3 Inicializálás

Fejlesztés során gyakran előfordul, hogy a kormányoszlop, illetve annak vezérlőegysége nem kívánt állapotokba kerül, így a controller elveszti az éppen aktuális pozíciót, további hibás működéshez vezet. Ezen probléma orvoslására a teszt módban létrehoztam egy inicializálás kérést, amely a mikrovezérlő belső inicializáló módját aktiválja. A mód futtatásához szükség van jelzőbitekre, amelyek lehetővé teszik a vezérlés letiltását az inicializálás idejére.

Mivel alkalmazásomban különvlik a küldendő és a fogadandó üzenet, ezért szükség van két dedikált jelzőbit kijelölésére. Az egyik a vezérlő adatmezőben kap helyet, a programban inicializálás hívásakor nulla értéket kap. Ekkor a szoftver letiltja a grafikus felület vezérlő és kijelző elemeit. A másik egy, a vezérlőegység adatait hordozó fogadott üzenetben kap helyet. Nulla értéket vesz fel, amikor a mikrokontroller sikeresen levezényelte az inicializálást. Ezt a számítógépen futó program detektálja, majd kilép az inicializáló módból, illetve újra engedélyezi a grafikus felület vezérlő és kijelző elemeit. A folyamat vázlatát a 2.4. ábrán látható.



2.4. ábra Inicializálás jelzőrendszer

3 Hardver perifériák tervezése

3.1 Kormányoszlop pozíció szenzor

3.1.1 Inkrementális pozíciószenzorok

Egy kormányoszlop mozgásának elektronikus vezérléséhez elengedhetetlen információ az oszlop pontos helyzete. Ez alapján hoz döntést az irányító eszköz, hogy egy adott irányba lehetséges-e még a mozgítás, illetve ennek segítségével lehet elmenteni, visszaállítani egy adott beállítást. A pozíció meghatározására számos, különböző előnyökkel és hátrányokkal rendelkező lehetőség kínálkozik.

Egyik előforduló alkalmazás az inkrementális szög-szenzor rendszer. Használatuknál a kitérés méréséhez a szenzor segítségével mérhető növekményt kell számlálni, és mindig hozzáadni az előző értékhez. Alkalmazásukkor körültekintően kell eljárni, mert a számlálási állapot a zavaró impulzusok miatt könnyen megváltozhat. Ennek ellenőrzésére rögzített vonatkozási jeleket szokás alkalmazni, de ezek csak korlátozott segítséget nyújtanak. Egy másik alapvető probléma, hogy a berendezés üzemi feszültségének bekapcsolásakor az abszolút értékek elvesznek. Néhány alkalmazásnál ezen nem segít a kikapcsoláskor mérhető pozíció eltárolása, mert az eszköz helyzete ebben a helyzetben manuálisan is megváltoztatható. [4]

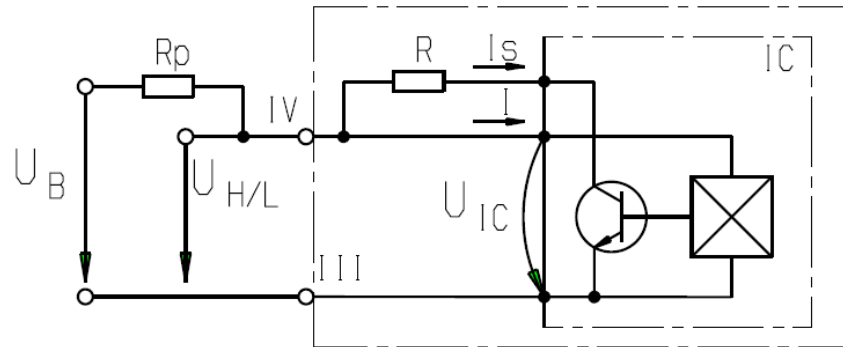
Az autóiparban számos különféle fizikai effektus használható fel inkrementális szenzorrendszeres méréshez, viszont a járművekben uralkodó üzemi körülmények miatt az optikai és kapacitív jellevők kevésbé használatosak. Helyettük főleg az indukciós szenzorok terjedtek el. Korábban többnyire „passzív” elemekből építkeztek, vagyis közvetlenül a szenzornál nem volt elektronika. Az újabb megoldások azonban már kis mérőhatással dolgoznak, emiatt a jelfeldolgozáshoz integrált elektronikára van szükségünk. Ezeket nevezzük „aktív” szenzoroknak. [4]

3.1.2 HALL-szenzoros megoldás

3.1.2.1 Szenzor, analóg-digitális bemenet

A kormányoszlop mozgásához egyenáramú motorokat használunk, amelyek a mechanikus áttételnek köszönhetően csak az elektronikus vezérlés hatására képesek

elmozdulni. Ebből kifolyólag a pozíció mérésére tökéletesen megfelel egy inkrementális szenzorrendszer, mivel a vezérlőegység kikapcsolásakor elmentett pozíció az eszköz újra feszültség alá helyezésekor is ugyanaz lesz. Berendezésünkben Hall-effektusra épülő aktív szenzorokat alkalmazunk a motorok fordulatainak méréséhez, ez adja meg az egyes irányokban mérhető pozíciókat. Ezek aktív szenzorok, helyettesítőképük a 3.1. ábrán látható. A belső ellenállás (R) specifikáció szerinti értéke $392\ \Omega$. Alkalmazásuk áramgenerátoros meghajtást, áram-feszültség átalakítást, illetve zavarűrészt igényel. [5]

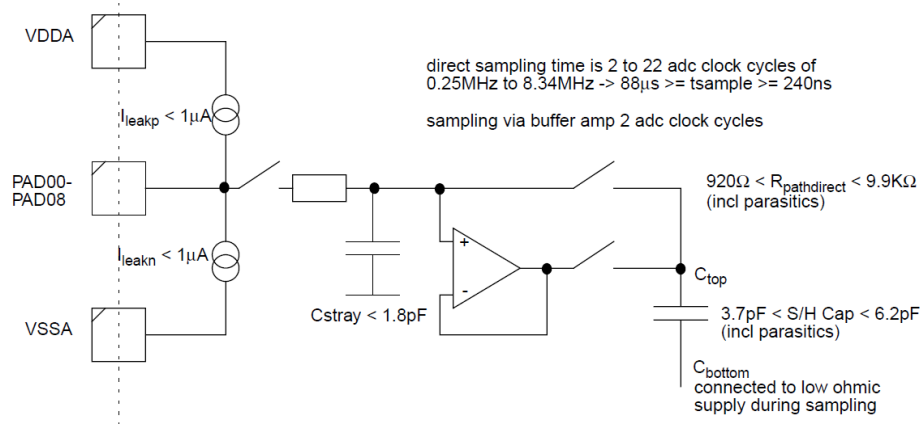


3.1. ábra A HALL-szenzor helyettesítőképe

A HALL-szenzor jele egy kétállapotú, azaz alapvetően digitális jel, mégis a mikrovezérlőben történő feldolgozását analóg-digitális átalakító segítségével valósítottuk meg. Ennek hatalmas előnye, hogy a szenzorbemeneteken mérhető feszültségből meg tudjuk állapítani a mechanikai hibákat, azaz hogy a hozzávezetések zárlatosak-e, vagy netán leszakadtak-e. Amennyiben 0 V feszültséget mérünk, biztonsággal állíthatjuk, hogy a szenzor bemenete földzárlatos. Amennyiben a jelalak alacsony, illetve magas állapota körüli feszültséget mérünk, a szenzor megfelelően működik. Amikor a normál jelalak magas szintjénél magasabb, konstans feszültséget mérünk, tápzárlatra következtethetünk, illetve szakadás esetén egy jellegzetes lebegés figyelhető meg a mikrokontroller bemenetén.

Maga az analóg-digitális bemenet közelítő helyettesítőképe a mikrokontroller-család adatlapjában hozzáférhető. A legfontosabb tudnivalók egy táblázatból kiolvashatóak. Az átalakító közvetlen bemenete egy soros ellenállással modellezhető, melynek tipikus értéke $5\text{ k}\Omega$. A 3.2. ábrán látható különböző kapacitásértékek helyett egy összegző kapacitás van megadva külön mintavételező, illetve nem mintavételező üzemmódra. Mivel számításaimban az eszközt mintavételezési üzemmódban veszem figyelembe, ezért az összkapacitás úgy tekinthető, mint a soros ellenállás után egy kapacitás a föld felé, melynek értéke az adatlap alapján 16 pF . Ezt követően található a

mintavételező bemenet, amelynek modellje egy ideális erősítő. Az adatlapban megadott maximális áram értéke 2,5 mA, ennek túllépése az adott bemenet meghibásodásához vezet. [6]



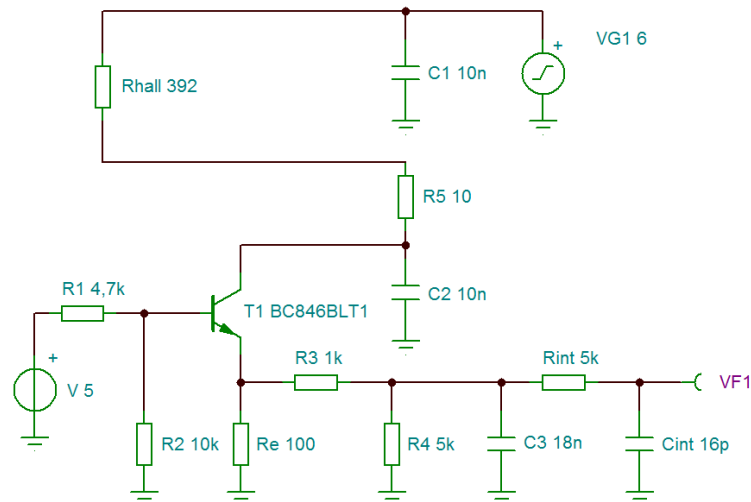
3.2. ábra Az analóg-digitális átalakító helyettesítőképe

3.1.2.2 A kapcsolás

A tervezés során az alapvető számításokat kézzel, papíron végeztem el, viszont a tűrések számításához sokkal gyorsabb és szemléletesebb egy áramkör-szimulációs program használata. Szakdolgozatom elvégzése során a Texas Instruments cég TINA nevű szimulációs programját alkalmaztam. Elsősorban azért, mert használatával kapcsolatban már korábbi tapasztalatokkal rendelkezttem, illetve az egyszerűbb számításokhoz nem volt szükség ennél komplexebb, bonyolultabb szoftverre.

A 3.3. ábrán látható kapcsolást egy, a cégnél végzett korábbi feladatban szereplő kapcsolás alapján terveztem meg. Az áramkör ellátásához a fejlesztendő berendezés saját feszültségforrásai szolgálnak. Az aktív HALL-szenzorhoz (R_{hall}) szükséges áramgenerátort egy földelt bázisú npn tranzisztoros alapkapcsolás alkotja. A tranzisztor egy előre meghatározott munkapontba van beállítva, amelyet a bázisra adott, egy feszültségosztó (R_1 , R_2) segítségével leosztott konstans bázisfeszültség, biztosít. A tranzisztor kollektor-emitter, illetve a HALL-szenzor árama az emitter-ellenállás (R_e) értékével kerül beállításra. A pontos működés tehát a következő. Amikor a HALL szenzor nem vezető, zárt állapotban van, az emitter ellenálláson (R_e) csak a bázisra adott feszültségből származó nagyon kevés áram folyik, amely azon így csak kis feszültséget ejt. Viszont amikor a HALL-szenzor vezető állapotban van, kinyit, a tranzisztor a kollektorán megjelenő feszültség hatására beáll az előre meghatározott munkapontba. A mikrokontroller analóg-digitális átalakítójára vezetett feszültség az emitter ellenálláson

(R_e) eső feszültségből származik, ez van tovább alakítva a zavarok kiszűrése érdekében. A HALL-szenzor védelme érdekében az áramgenerátoros ágba még egy soros ellenállás (R_5), illetve két zavarcsűrő kondenzátor (C_1 , C_2) is került.



3.3. ábra A HALL-meghajtó kapcsolása

Az eszköz specifikációjában megadott, alkatrészek paraméterezéshez szükséges peremfeltételek a következők. A HALL-szenzor által kapcsolható maximális áram értéke 40 mA. A zavarcsűrés biztosításához az emitter-ellenállás utáni aluláteresztő szűrő sávszélessége 10 kHz. Az alkalmazott mikrovezérlő analóg-digitális átalakító bemenetre jutó maximális áram adatlapban meghatározott értéke 1 mA. Az analóg-digitális átalakító referenciafeszültsége 5 V, így a bemenetre adható maximális feszültségszint ezt nem haladhatja meg.

A szükséges, papíron elvégzett számításaim során először a kapcsolási áramot állítottam be. Az emitter-ellenállás (R_e) 100 Ω -os értéke, adott volt a korábbi kapcsolásból, ami alapján dolgoztam. Az általam választott kapcsolási áram értéke körülbelül 30 mA, ami azt jelenti, hogy a tranzisztor emitterén a földponthoz mérhető feszültség 3 V. Egy általános közelítéssel élve, miszerint a tranzisztor bázis-emitter feszültsége 0,7 V, a bázis potenciálja 3,7 V. A bázis táplálására egy 5 V értékű feszültségforrást választottam, így a feszültségosztó által elérendő arány 0,74. Szabványos ellenállásértékeket alkalmazva 4,7 k Ω -ot, illetve 10 k Ω -ot választottam. Így a bázis feszültsége:

$$U_b = 5V \frac{R_2}{R_1 + R_2} = 5V \frac{10k\Omega}{4,7k\Omega + 10k\Omega} = 3,4V \quad [1]$$

Így az emitter potenciálja 2,7 V, ebből adódik, hogy a kapcsolt áram értéke:

$$I_c = I_e = \frac{U_e}{R_e} = \frac{2,7V}{100\Omega} = 27mA \quad [2]$$

A teljes kapcsolás átviteli függvényének kiszámítása helyett egy közelítéssel élek, miszerint az analóg-digitális átalakító bemenetén lévő soros belső ellenállás (R_{int}) és belső kapacitás (C_{int}) hatása a szűrő törésponti frekvenciájára nézve elhanyagolható, mivel a kondenzátor értéke a külső elemekéhez képest nagyon kicsi. A tranzisztor által előállított feszültségre alkalmazott aluláteresztő szűrő átviteli függvénye tehát a következő:

$$A = \frac{R_4 \times \frac{1}{j\omega C}}{R_3 + R_4 \times \frac{1}{j\omega C}} = \frac{\frac{R_4}{R_3 + R_4}}{1 + j\omega \frac{R_3 R_4 C}{R_3 + R_4}} \quad [3]$$

Ebből az alakból könnyen meghatározható a vágási körfrekvencia:

$$\omega = \frac{R_3 + R_4}{R_3 R_4 C} = 2\pi f \quad [4]$$

A szűrő áramkörben a C_3 kondenzátorral párhuzamosan kapcsolt R_4 ellenállásra azért van szükség, hogy a C_3 kondenzátor értéke minél kisebb lehessen. Ennek költségminimalizáló okai vannak, ugyanis nagyobb kapacitású kondenzátornak az ára is nagyobb. Ennek a megoldásnak viszont van egy hátránya, az R_3 és R_4 ellenállás egy feszültségosztót alkot, ami csökkenti az analóg-digitális bemenetre jutó jel szintjét. Ezzel együtt csökkenti a logikailag eltérő jelek feszültségkülönbségét, ezáltal növeli a controllerben végzett komparálás hibáját. Ezt a jelenséget elkerülendő az R_4 ellenállást érdemes jóval nagyobbra választani, mint az R_3 ellenállást. Ezen megfontolások alapján az R_3 és R_4 ellenállásokat 1 k Ω és 10 k Ω értékűre választottam.

A 4. egyenletrendszer átrendezve adott vágási frekvencia és ellenállásértékek mellett a szükséges kondenzátor értéke:

$$C_3 = \frac{R_3 + R_4}{2\pi f R_3 R_4} = 17,51nF \quad [5]$$

A kapott érték alapján a C_3 kondenzátor kapacitásának a szabványos 18 nF értéket választottam.

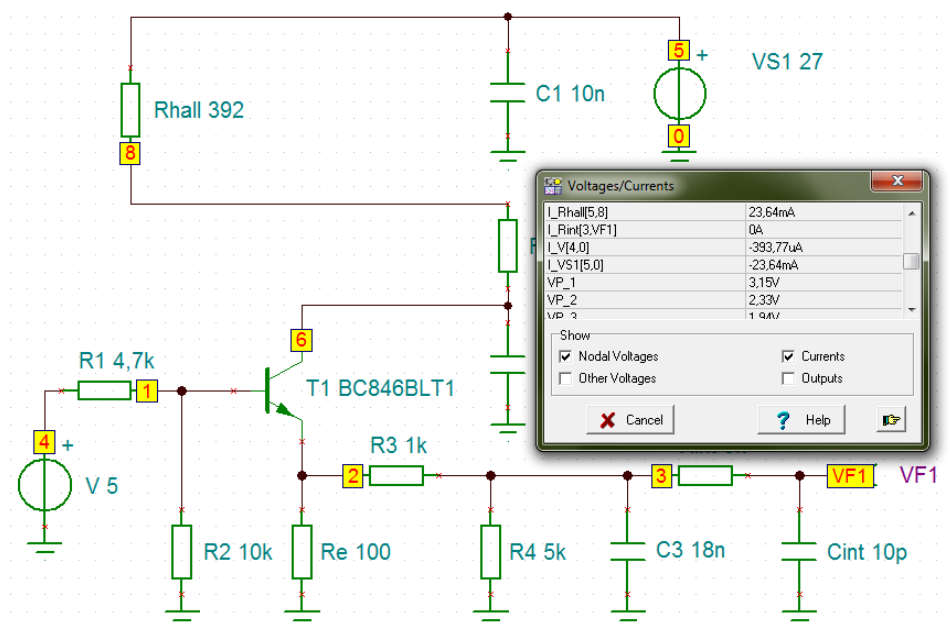
Az áramkörben megtalálható C_1 és C_2 szűrőkondenzátorok értékeit, az R_5 védőellenállás értékét, illetve az npn tranzisztor típusát a korábbi kapcsolás értékeihez képest nem változtattam.

3.1.2.3 Áramkör-szimuláció

A megtervezett, felparaméterezett kapcsolásnak a tesztelése maradt hátra. Ezt a feladatot a TINA áramkörtervező programanalizáló funkcióival valósítottam meg.

Az első szimulációt a maximális kapcsolási áram meghatározására végeztem. Mivel a HALL-szenzort egy egyszerű ellenállással modellezem, ezért a működés során előálló feszültség kapcsolást egy négyszögjel generátorral helyettesítem. A maximális kapcsolási áram a tranziens jelenségektől eltekintve úgy határozható meg, hogy a gerjesztés helyére beillesztek egy 27 V értékű feszültségforrást, amely azt jelenti, hogy a HALL-szenzor nyitott állapotban van, illetve rajta a maximális áram folyik. A 27 V a járművekben elérhető maximális telepfeszültség értéke.

A számításhoz a TINA program egyenáramú (DC) szimulációját használom, mivel a kapcsolást egy állandó értékű feszültségforrással hajtom meg. Az egyenáramú szimuláció egyenáramú munkaponti analízist végez, azaz megkeresi az áramkör állandósult állapotát, majd kiszámolja az egyes csomóponti feszültségeket. Az így kapott eredmények alapján az egyes elemeken átfolyó áram, illetve a transzfer karakterisztika is kiszámítható. [7]

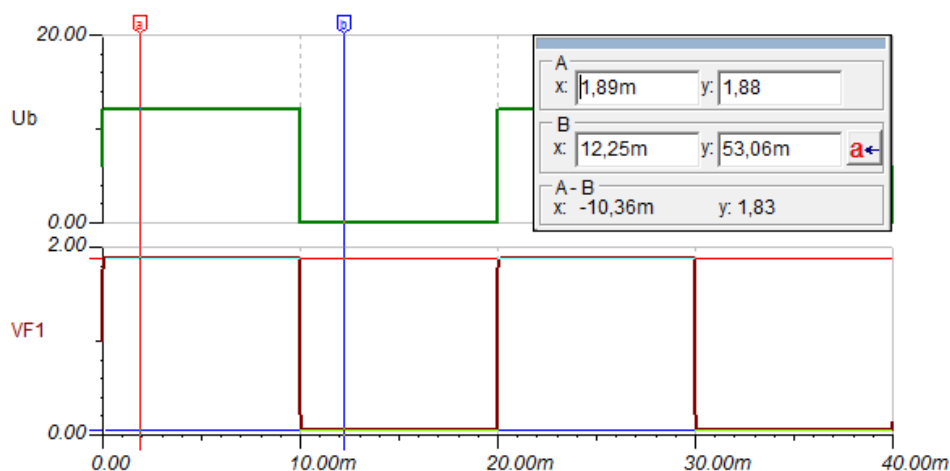


3.4. ábra A HALL meghajtó DC szimulációja

A szimuláció eredménye a 3.4. ábrán látható. Az autóban előforduló maximális bemeneti feszültség, 27 V mellett a HALL-szenzoron átfolyó maximális kapcsolandó áram ($I_{Rhall}[5,8]$) értéke 23,64 mA. Ez a specifikációban megadott maximális érték körülbelül 60%-a, így a jel árama zavarmentes, de elegendő nagyságú jelzésre képes a szenzor meghibásodása nélkül.

A második szimuláció az analóg-digitális bemenetre jutó jel nagyságának számítását szolgálja. A HALL-szenzor most már helyes működésének modellezéséhez egy 6 V amplitúdójú, 6 V középvértékű, 50 Hz frekvenciájú négyszögjel-generátort alkalmazok az áramkör meghajtásakor. Ebben az esetben a kapcsolás kimenetén, azaz a mikrokontroller analóg-digitális átalakítójának bemenetén mérhető feszültség jelalakját vizsgálom az eltelt idő függvényében.

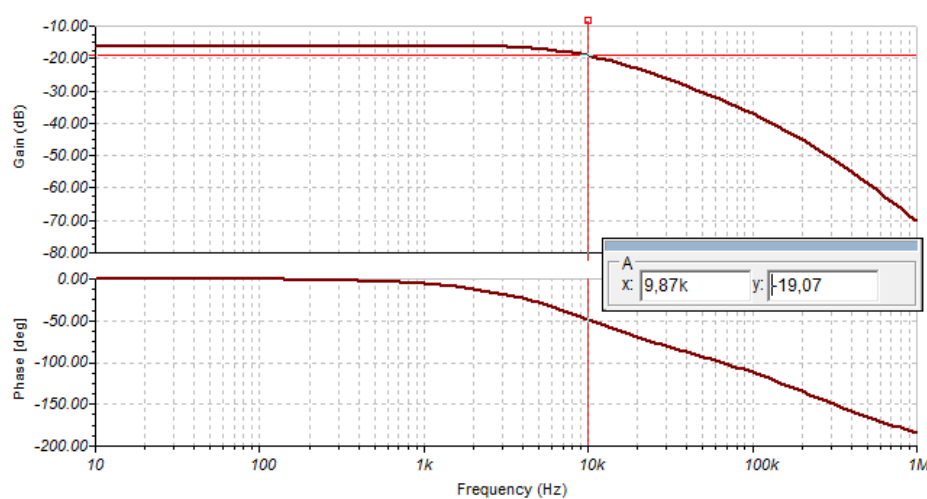
Ehhez a számításhoz a TINA program tranziens analízisét használom. Ez a szimuláció az áramkör meghatározott gerjesztésre adott időtartománybeli válaszát határozza meg. Az analízis paraméterezésénél megadható időtartomány kezdeti értéke csak a kijelzésre vonatkozik, a számítás mindig a 0 időpillanatban kezdődik. A megadható végérték a szimuláció, illetve a kijelzés végértékét is szolgáltatja. [7]



3.5. ábra A HALL meghajtó tranziens szimulációja

A szimuláció eredményeként kapott, 3.5. ábrán látható grafikonról leolvasható, hogy az analóg-digitális bemenetre jutó feszültség a HALL-szenzor nyitása mellett 1,88 V, míg zárása mellett 53,06 mV. Ezek a feszültségértékek 12 bites analóg-digitális átalakító esetén 5 V referenciafeszültség mellett 1539 és 43 számértékeknek felelnek meg, amelyek jól elkülöníthetőek egymástól.

Az utolsó szimuláció az aluláteresztő szűrő vágási frekvenciájának meghatározására szolgál. Ehhez a számításhoz a TINA program váltóáramú (AC) analízisét használom. Ez a szimuláció kisjelű analízist végez az áramkörön, majd megadja a lineáris áramkör frekvenciamenetét. A nemlineáris karakterisztikával rendelkező elemeket az előre kiszámolt egyenáramú munkapont körül linearizálja. [7]



3.6. ábra A HALL meghajtó AC szimulációja

Ebben az esetben az áramkört ismét az előbb meghatározott négyszögjellel gerjesztem, viszont a frekvencia értékének a szimuláció paramétereinek meghatározásakor két szélsőértéket adok meg, illetve egy lépésszámot. Eredményül a kapcsolás átviteli függvényét kapom meg a megadott frekvenciaintervallumban.

A szimuláció eredményeként kapott, 3.6. ábrán látható Bode-diagramról leolvasható a -3 dB-es értéknél található törési frekvencia, 9,87 kHz. Ez a specifikációban előírt 10 kHz körüli érték, így a paraméterezés megfelelő.

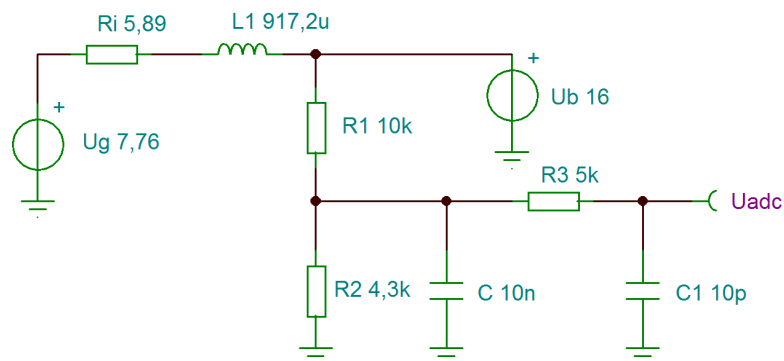
3.2 Feszültségmérés

A fejlesztendő kormányoszlop-mozgató berendezés specifikációjában meghatározott hardver-periféria egy feszültségmérő, amely segítségével a központi mikrovezérlő képes megmérni a motorok kapcsain lévő potenciálokat. Ez visszacsatolást ad a motorok pillanatnyi működéséről, ebből kifolyólag egy hibadetektálási lehetőséget is biztosít. Ugyanis a központi vezérlőegységéhez tartozó perifériák, motorok és HALL-szenzorok, egy autóiipari szabványnak megfelelő csatlakozó segítségével kapcsolódnak. Ez ugyan nem engedi meg, hogy rosszul csatlakoztassuk a berendezést, viszont az autóban kialakuló esetleges mechanikai

sérülés következtében előfordulhat, hogy az egyes vezetékek téves potenciálhoz csatlakoznak. Ez a hiba a vezérlőegység elektronikája szempontjából végzetes következményeket vonhat maga után. Ezt elkerülendő figyelniünk kell, hogy a kivezetéseken nem áll-e fenn zárlati probléma.

3.2.1 A kapcsolás

A 3.7. ábrán látható, alkalmazandó kapcsolás egy egyszerű feszültségosztóból (R_1 , R_2), illetve egy aluláteresztő szűrőből (R_1 , R_2 , C) áll a motorok kapcsain. A kapcsolásba az előzőleg impedancia-analizátorral megmért motor modelljét is belettem, bár a számításokba a közvetlen telepfeszültség miatt nem számít bele.



3.7. ábra A motorfeszültség mérő kapcsolás

A fejlesztendő eszköz specifikációjában meghatározott tulajdonságok a motorkapcsokon lévő potenciál mérésére a következők. Mérési tartománya legyen 0 V és 16 V között. Tartalmazzon 5 kHz törésponti frekvenciájú aluláteresztő szűrőt a zavarok szűrésének érdekében. A mikrokontroller analóg-digitális bemenetén a maximális áram 1 mA értékű lehet.

A szükséges alkatrészek pontos értékét papíron történő számítással határoztam meg. Mivel az analóg-digitális átalakító referencia-feszültsége 5 V, első feladat a mérési határ maximális értékének megfelelő kimeneti értéket egy ennél alacsonyabb feszültségre hangolni. Az első összefüggést a feszültségosztó arányára írtam fel.

$$5V \geq 16V \frac{R_2}{R_1 + R_2} \quad [6]$$

Ezt az egyenlőtlenséget átrendezve megkapjuk az ellenállások arányát.

$$R_2 \leq \frac{5}{11} R_1 \quad [7]$$

R_1 ellenállást 10 k Ω értékűre választva, az R_2 ellenállásra a 4,3 k Ω szabványos érték felel meg.

A teljes kapcsolás átviteli függvényének kiszámításakor a HALL-szenzor meghajtó kapcsolásánál már ismertetett közelítéssel élek. Így tehát a függvény értéke:

$$A = \frac{R_2 \times \frac{1}{j\omega C}}{R_1 + R_2 \times \frac{1}{j\omega C}} = \frac{R_2}{R_1 + R_2 + j\omega R_1 R_2 C} = \frac{\frac{R_2}{R_1 + R_2}}{1 + j\omega \frac{R_1 R_2}{R_1 + R_2} C} \quad [8]$$

Ebből az alakból könnyen meghatározható a vágási körfrekvencia:

$$\omega = \frac{R_1 + R_2}{R_1 R_2 C} = 2\pi f \quad [9]$$

Ezt az egyenletrendszert átrendezve adott vágási frekvencia és ellenállásértékek mellett a szükséges kondenzátor értéke:

$$C = \frac{R_1 + R_2}{2\pi f R_1 R_2} = 10,59 \text{ nF} \quad [10]$$

A kapott eredmény helyett a nála nagyobb, de hozzá legközelebb eső 12 nF szabványos értéket választottam a kondenzátor kapacitásának.

3.2.2 Áramkör-szimuláció

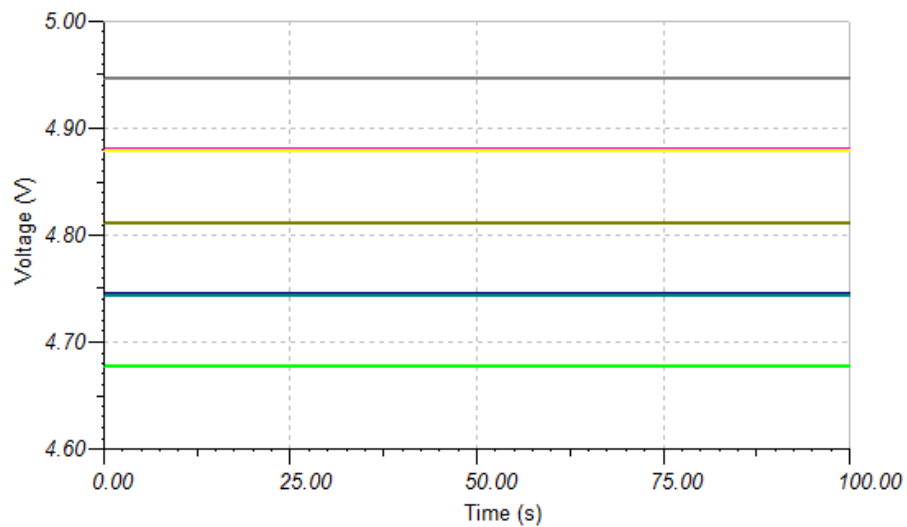
Az áramkör ellenőrzését a TINA szimulációs eszközeivel végeztem. A program képes olyan szimuláció számítására, amelyben lehetőség van az egyes alkatrészek paraméterszórásának beállítására. Az autóiipari alkalmazásokban a diszkrét ellenállások toleranciája 2%. Ennek beállítási módja a két szélsőérték, illetve a köztük lévő lépésszám megadása. A szimulációhoz az áramkör összes diszkrét elemére beállítottam a 2%-os paraméter-eloszlást.

A kimeneti jel szintjének szimulálásához a HALL-szenzor szimulációjánál ismertetett tranziens analízist használtam. Az áramkör gerjesztése ebben az esetben az előírt mérési tartomány maximális értéke, egy 16 V-os egyenáramú feszültségforrás.

A 3.8. ábrán látható görbesereg a beállított különböző paraméterekhez tartozó számítások eredményeit mutatja. Megfigyelhető, hogy az összes görbe az analóg-digitális átalakító bemenetére adható maximális 5 V-os feszültségszint alatt marad.

A szimuláció 16 V-os gerjesztőfeszültsége mellett az általam kiválasztott ellenállásértékek mellett a kimeneten megjelenő feszültség:

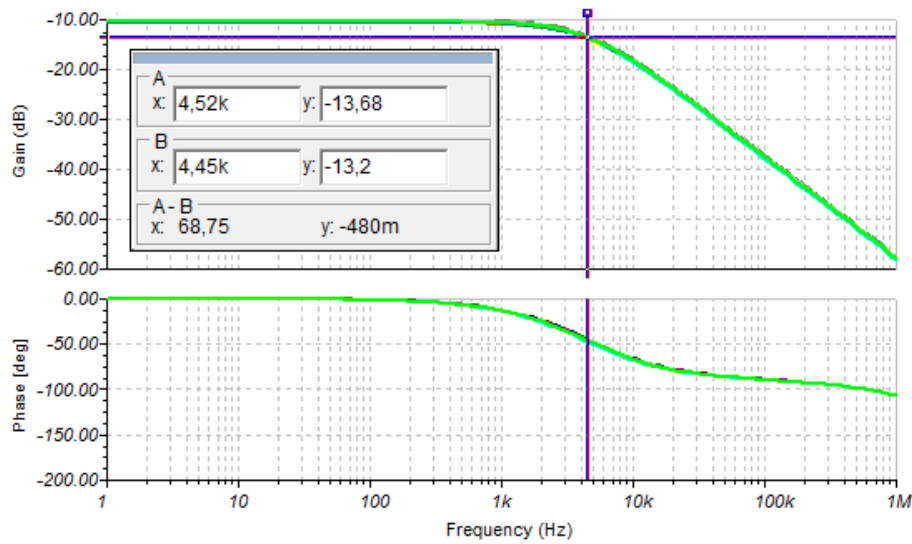
$$U_{adc} = Ub \frac{R_2}{R_1 + R_2} = 4,811 V \quad [11]$$



3.8. ábra A feszültségmérő tranziens szimulációja

A 3.8. ábrán megfigyelhető, hogy a szimuláció eredményén az ezen értéktől legtávolabb eső feszültségszintek távolsága 0,13 V. Ez azt jelenti, hogy a kapcsolás által a mérésbe vitt hiba, $\frac{0,13V}{4,81V} = 0,027$, azaz 2,7%. Mivel ez a hiba az ellenállások értékétől függ, használat során állandó, ezért a mikrokontrollerben egy egyszerű arányossággal kompenzálható.

Az aluláteresztő szűrő törésponti frekvenciáját a TINA program HALL-szenzor szimulációinál ismertetett váltakozó áramú analízisével határoztam meg. Ennél a számításnál is alkalmaztam az elemek értékeinek paraméterszórását.



3.9. ábra A feszültségmérés AC szimulációja

A 3.9. ábrán látható görbesereg két szélsőértékét kiválasztva megállapítható, hogy a beállított törésponti frekvencia kisebb, mint a specifikációban előírt 5 kHz, így a kapcsolás egy nagyobb szűrést alkalmazva teljesíti a követelményeket.

4 Termikus modell meghatározása

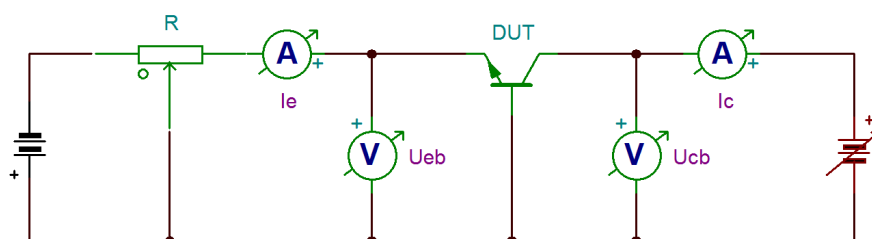
A szakdolgozatom során elvégzett feladatok közé tartozik a fejlesztendő vezérlőegység teljesítménytranzisztorainak termikus mérése. A hibátlan működés érdekében számon kell tartanunk a tranzisztorok hőmérsékletét, mivel az esetleges túlmelegedés a FET tönkremeneteléhez vezet.

4.1 Irodalomkutatás

A feladat elvégzése során megismerkedtem egy korábbi megoldással. Kővári Péter 1982-ben elkészített doktori értekezésében foglalkozott teljesítménytranzisztorok termikus ellenállásának mérésével. [8]

A lehetséges méréseket két csoportba osztja. Folyamatos módszerre, amikor állandó terhelés közben történik a hőfokmérés, valamint impulzusüzemű módszerre, amikor a mérés a terhelő impulzusok szerint van ütemezve. [8] Szakdolgozatomban mindkét mérési módszerre bemutatok egy-egy példát.

A doktori értekezésben megtalálható egyik folyamatos mérési módszerhez tartozó kapcsolási vázlat a 4.1. ábrán látható.



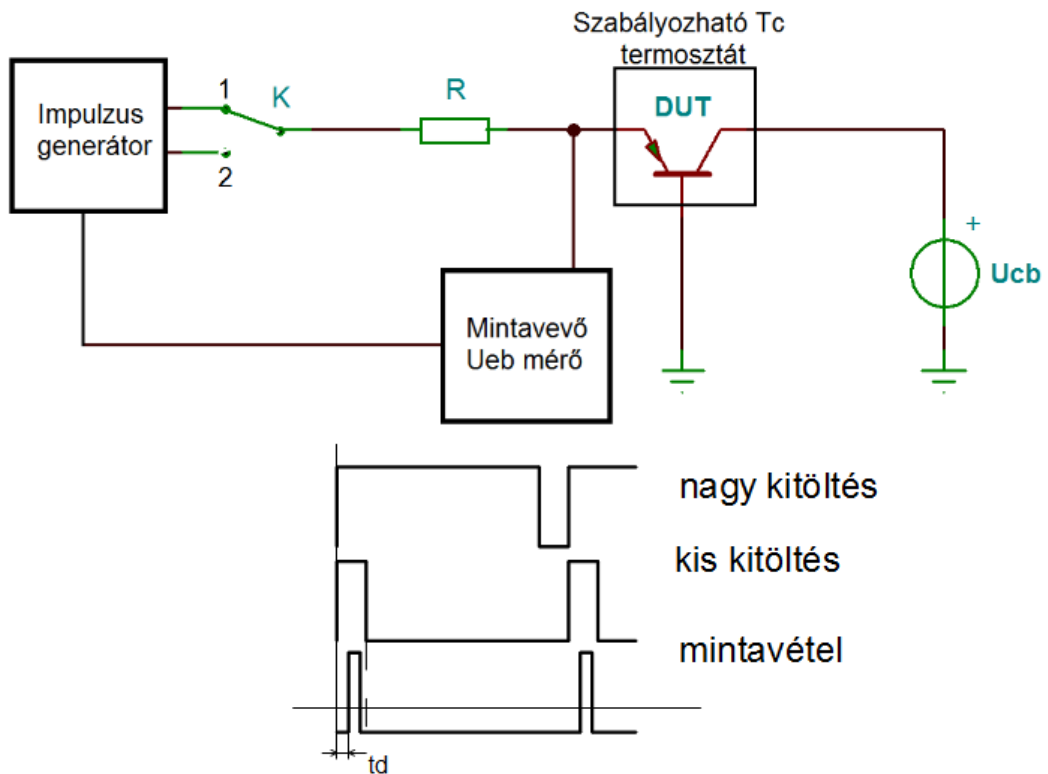
5. ábra A folyamatos módszer mérési elrendezése

A mérendő tranzisztor folyamatos idejű gerjesztését egy földelt bázisú alapkapcsolás segítségével valósítjuk meg. Ebben a mérésben a tranzisztor réteghőmérsékletét szintén folytonosan vizsgáljuk az U_{EB} nyitófeszültség által. Az eljárás során a tranzisztorra egy tokhőmérséklet-stabilizáló, termosztált, vízzel szabályzott réz-tönk kapcsolódik. Ennek segítségével először beállítunk egy T_{c1} hőmérsékletet, majd a $P_{d1} = I_C U_{CB1}$ disszipációt, és megmérjük a réteghőmérsékletet reprezentáló U_{EB} feszültségértéket. Másodszor, azonos I_E áramnál beállítunk egy $T_{c2} >$

T_{c1} tokhőmérsékletet, és az U_{CB} feszültséget, addig csökkentjük, amíg az U_{EB} feszültségi meg nem egyezik az előző mérés során kapott értékkel. A termikus ellenállást a mért adatokból a következő módon számíthatjuk: [8]

$$R_{thjc} = \frac{T_{c2} - T_{c1}}{I_c (U_{CB1} - U_{CB2})} \quad [12]$$

Az impulzusos üzemi módszerek közül egyszerűsége miatt az impulzus kitöltési tényezőjét változtató eljárás bemutatását választottam. A mérési elv a 4.2. ábrán követhető.



6. ábra Az impulzus módszer mérési elrendezése

A mérés során a K kapcsolót először az 1-es állásba kapcsoljuk. A tranzisztorra az előző módszerben ismertetett, de szabályozhatóan termosztált réz-töng van kapcsolva, ami jelen esetben egy T_0 hőmérsékletet biztosít. Ekkor a kapcsolón egy nagy kitöltésű impulzusmodulált feszültség jelenik meg, amely az R ellenálláson létrehozza a gerjesztő emitteráramot. A tranzisztor kollektora és bázisa közé az U_{CB} egyenfeszültség kapcsolódik, így az eszközön disszipált teljesítmény:

$$P_{dl} = \overline{I_{B1}} (U_{CB} + U_{EB}) \quad [13]$$

Ennek hatására a réteghőmérséklet egy T_j értékkel a tokhőmérséklet fölé emelkedik. Ekkor az U_{EB} feszültséget az impulzus felfutó éle után t_d késleltetéssel mérjük az állandósult hőmérséklet kialakulása érdekében.

A mérés második része, amikor a kapcsolót a 2-es állásba kapcsoljuk. Ilyenkor egy kisebb kitöltési tényezőjű impulzusmodulált emitteráram jelenik meg a tranzisztoron, kisebb disszipációt és réteghőmérsékletet eredményezve.

$$P_{d2} = \overline{I_{E2}}(U_{CB} + U_{EB}) < P_{dl} \quad [14]$$

A hőmérséklet kompenzálása érdekében a szabályozható termosztáttal addig emeljük a tokhőmérsékletet, amíg az U_{EB} feszültség a kapcsoló 1-es állásában mért értékkel meg nem egyezik. Az így létrejött ΔT tokhőmérséklet növekedésből a termikus ellenállás meghatározható: [8]

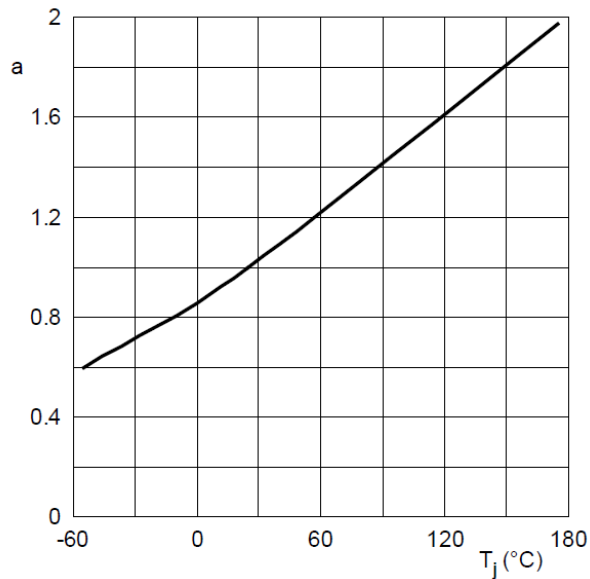
$$R_{thjc} = \frac{\Delta T}{(I_{E1} - I_{E2})(U_{CB} - U_{EB})} \quad [15]$$

4.2 Saját megoldás

4.2.1 A mérés elve

A modellezendő FET tranzisztorok adatlapjában két értéket is találhatunk a termikus ellenállásra. Az egyik eset, amikor a FET tranzisztor egy ideális hűtőfelületre kapcsolva méri meg a termikus ellenállását. Ekkor a megadott adatlapban szereplő 2,36 K/W érték a FET magja és hűtőfelülete között értendő. A másik esetben a tranzisztor egy minimális fémezésen, nyomtatott áramköri kártyára szerelve vizsgálják. Az így megadott, adatlapban szereplő 95 K/W érték a FET magja és a környezet között értendő. Feladatom az elkészült, beültetett vezérlőegységen található meghajtó tranzisztorok teljes környezetükkel együtt alkotott rendszer termikus modelljének meghatározása. [9]

Az FET adatlapjában található a 4.3. ábrán látható grafikon, amely megadja a tranzisztor nyelő (drain) és forrás (source) kivezetései közötti mérhető pillanatnyi ellenállás 25 °C-os értékre vetített arányos értékét a FET magjának hőmérséklete függvényében. Tehát a két kivezetés közötti ellenállás (R_{DS}) méréséből a grafikon alapján meg lehet határozni a mag hőmérsékletét. [9]



$$a = \frac{R_{DSon}}{R_{DSon}(25^{\circ}\text{C})}$$

4.3. ábra Ellenállás - hőmérséklet függvény

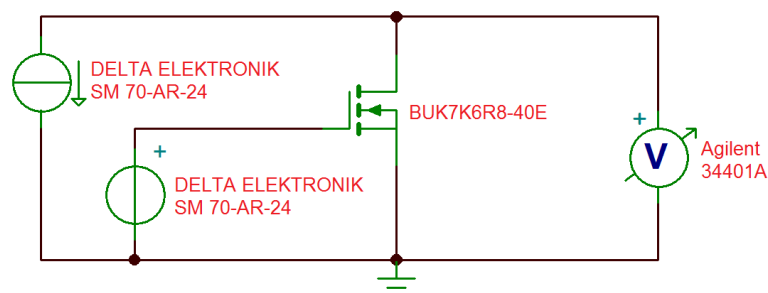
Egy tranzisztor termikus ellenállását úgy tudjuk meghatározni, hogy megmérjük a hőmérséklet változását egy adott, konstans teljesítmény rákapcsolásakor. Ehhez szükség volna egy olyan tápegységre, amely képes teljesítményszabályzásra. Mivel ilyen berendezés nem állt rendelkezésre, méréseim elvégzésére egy közelítő módszert alkalmaztam. Azt állítom, hogy a FET-et egy konstans árammal melegítve hőmérséklete egy állandó értékre áll be. Mivel állandó a hőmérséklet, állandó az R_{DS} ellenállás is, így a rajta folyó áram állandó feszültséget ejt. Ebből következik, hogy tranzisztoron átfolyó teljesítmény is állandó lesz. A termikus ellenállás közelítő módszerrel történő meghatározásánál ezeket a beállási ellenállás- és teljesítményértéket veszem figyelembe.

A mérendő rendszert első közelítésben egy egytárolós taggal közelítem. Ennek a modellnek az identifikációjához a fent említett áram egységugrásra adott ellenálláson eső feszültségből kiszámolt hőmérsékletválaszt veszem figyelembe. Ebből a rendszer termikus ellenállásának ismeretével meghatározható a termikus kapacitás értéke is.

4.2.2 Mérési elrendezés

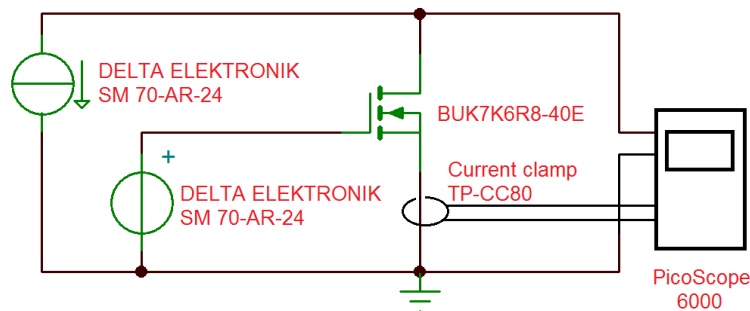
A számítások alapjául szolgáló értékek meghatározásához két különböző mérési elrendezésre van szükségünk.

Először meg kell tudni mérni a FET R_{DS} ellenállását szobahőmérsékleten. Ehhez szükséges tranzisztoron átvezetendő konstans áram beállítása, amelyet egy állítható áramkorláttal rendelkező tápegység segítségével valósítottam meg. A FET adatlapjában olvasható, hogy a tranzisztor teljes nyitáshoz a kapura (gate) adott feszültség (U_{GS}) nagysága 10 V legyen. Ezen feszültség előállítására egy külön tápegységet kell használni, amelynek a földpontja össze van kötve az előző, konstans áramot beállító tápegységével. Ezt az összekötött potenciált kapcsoltam a tranzisztor forrására (source). A szobahőmérsékleten mérhető R_{DS} ellenállás pontos megmérést úgy lehet elvégezni, hogy a tranzisztoron nagyon kevés, melegítésre még nem alkalmas mérőáramot folyatunk át, és a rajta eső feszültséget megmérjük. Ez egy 1 mV körüli érték, melynek megmérésehez egy hat és fél digitális asztali multimétert használtam. Ennek a mérési elrendezésnek a vázlata látható a 4.4. ábrán.



7. ábra Ellenállásmérés szobahőmérsékleten

A termikus mérés egy nagyon nagy érzékenységű, nagyon gyors, viszonylag hosszú ideig mintavételező, mérési adatok elmentésére képes feszültségmérőt igényel. Erre a legjobb eszköz a számítógéphez csatlakoztatható szoftver alapú oszcilloszkóp volt. A FET meghajtásának módja megegyezik az előző mérésben leírtakkal. Illetve a környezeti hőmérséklet mérésére egy, a helyiségben elhelyezett digitális hőmérő szolgált. A tranzisztort fűtő áram értékét egy lakatfogó segítségével szintén az oszcilloszkóppal mértem. Ezen mérőeszköz jelét használtam fel a mérés triggereléséhez. Az termikus modell meghatározásához szükséges elrendezés vázlata tehát a 4.5. ábrán látható.



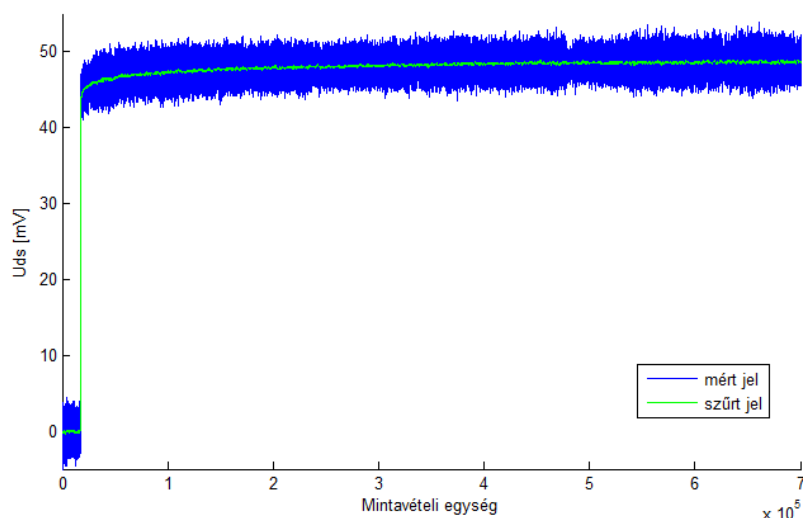
8. ábra Termikus modell mérőkapcsolás

4.2.3 Mérés, modellezés

A mérést a FET R_{DS} szobahőmérsékleten felvett ellenállásának meghatározásával kezdtem, abban az esetben, amikor a tranzisztor fel van forrasztva a központi vezérlőegység nyomtatott áramköri hordozójára, tehát az üzemszerűen használatos hűtőfelületre. A nyitófeszültség bekapcsolása után az áramot 200 mA-ig növeltem. Ennél az áramértéknél kezdtem állandó ellenállásértéket számolni a mért feszültség és az átfolyó áram hányadosaként. Ebből arra következtettem, hogy ezen áramérték körül még nem kezd melegedni a tranzisztor, viszont már elegendő áram folyik át a tényleges ellenállás meghatározásához. Az így kapott szobahőmérsékleti forrás-nyelő ellenállás (R_{DS0}) értékére 5,31 m Ω adódott. Az adatlapban ennek a tipikus értéke 25 °C-on 5,8 m Ω , így a kapott érték 22,3 °C-on hihető.

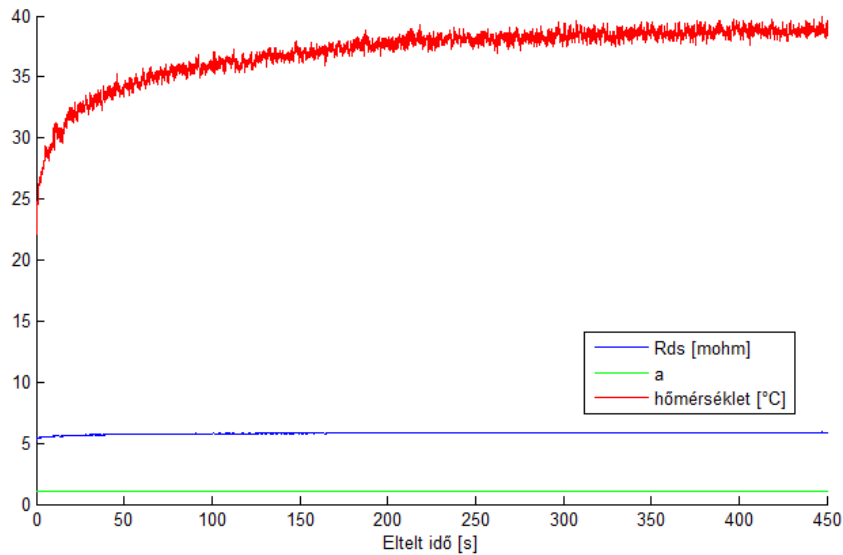
A következő mérés az áram egységugrásra adott, nyelő-forrás kivezetéseken mérhető feszültségválasz (U_{DS}). A modell identifikációját ezen ugrásválasz átszámolása alapján fogom meghatározni. A PicoScope-ot egyszeri pásztázás (single) üzemmódba kapcsoltam, ami azt jelenti, hogy az első triggerjel hatására kezdődik el a mintavételezés. A teljes mérendő időtartam 460 másodperc volt, a mintavételi frekvencia pedig 1996 1/s.

A mért eredményeket a Matlab program segítségével dolgoztam fel. Az oszcilloszkóp által mért feszültségértékeket egy tömbbe importáltam. Mivel a kapott görbe a kis jelértékek mérése miatt nagyon zajos, először egy egydimenziós szűrőt alkalmaztam. Ez úgy működik, hogy az új tömb egy értéke a megadott bemeneti tömb azonos értéke előtti meghatározott számú érték átlaga lesz. A feladathoz egy 1000 értékre átlagoló szűrőt használtam. A bemeneti, illetve a szűrt jel a 4.6. ábrán látható.



4.6. ábra Szűrő alkalmazása mért jelre

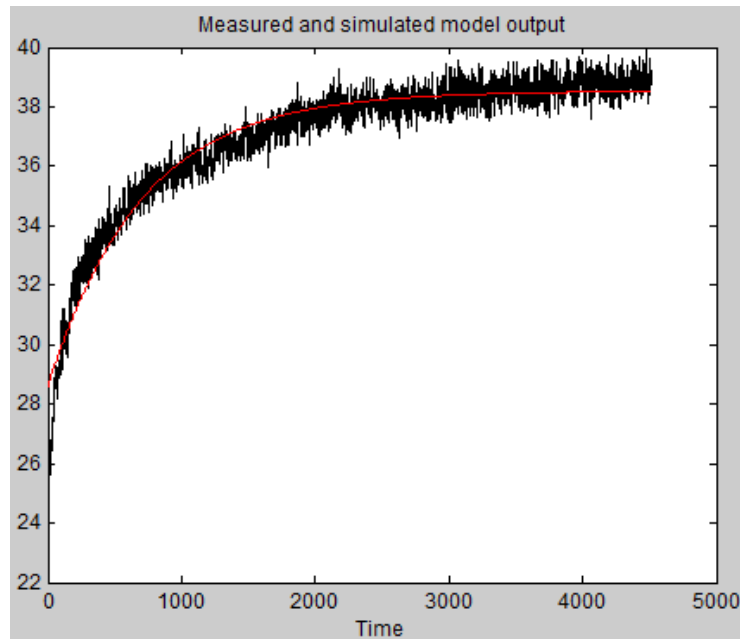
Az így kapott értéket leosztottam a tranzisztoron átfolyó konstans árammal, így megkaptam a R_{DS} pillanatnyi értékeinek pontjait. Mivel a görbe a gerjesztőáram bekapcsolás előtti értékeket is ábrázol, a pontos identifikációhoz meg kell határozni a nulla pontot a fölösleges elemek levágásával. Ehhez ábrázolom a kapott ellenállásgörbét, megnézem, hogy hanyadik értékénél található a megmért R_{DS0} , majd attól az elemtől átmásolom egy új tömbbe. A FET maghőmérsékletének meghatározásához ki kell számolni az adatlapban megadott „a” tényezőt, majd a görbe alapján átszámolni a hőmérsékletet. Az „a” tényező számítása egyszerű, a kezdeti, szobahőmérsékleti R_{DS} értékkel el kell osztani a kiszámolt ellenállásértékeket. Az adatlapban megadott „a” tényező – hőmérséklet görbét egy egyenessel közelítettem, melynek egyenlete alapján a Matlab segítségével számolom át az egyes hőmérsékletértékeket. A pontos megjelenítéshez szükséges továbbá a grafikonok x tengelyének az átskálázása. Mivel két mintavétel között eltelt idő 0,00501 s, ezért létrehozok egy új tömböt, melynek lépései ekkora egységenként vannak. Amennyiben a mért, illetve átszámolt eredményeket ennek a tömbnek függvényében ábrázolom, másodpercben skálázott eredményt kapok. Az így átszámolt R_{DS} ellenállás, „a” tényező és hőmérsékletértékek a 4.7. ábrán láthatók.



4.7. ábra Összesített számítási eredmények

Az áramgerjesztésre adott hőmérsékletválasz meghatározása után már csak az identifikáció van hátra. Ezt a Matlab rendszer identifikációs eszköztárával (System Identification Toolbox) végzem el. Először is meg kell adni a gerjesztést és a választ. Az oszcilloszkóp lakatfogón keresztül mért áramértékeinek lecsonkított ellenállástömbhöz tartozó elemeit tettem bele egy új tömbbe. Ez lett maga a gerjesztés. A válasz pedig maga a kiszámolt hőmérsékletgörbe.

Szükséges továbbá a kezdő pillanat, jelen esetben 0 s, és a mintavételi idő, jelen esetben 0,000501 s megadása. Miután az identifikációs eszköztár importálta a beírt értékeket, következhet a kiszámítandó modell kiválasztása. Itt az egytárolósat választottam, majd becsültem.



4.8. ábra Modell illesztése

Illeszkedését az importált válaszra a 4.8. ábra szemlélteti. Az illesztés pontossága 76,83 %-os, a közelítés tehát egyelőre megfelel. A becült modell átviteli függvénye:

$$A = \frac{4,626}{1 + 69,78s} \quad [16]$$

A hőkapacitás értékét a következőképpen számolom a kezdeti és vég hőmérsékletből, illetve a tranzisztoron átfolyó teljesítményből:

$$R_{th} = \frac{T_{vég} - T_{kezdeti}}{I_{gerj}^2 \cdot R_{DS0}} = \frac{39^\circ C - 22,3^\circ C}{8,304^2 \cdot 0,00531W} = 45,61 \frac{K}{W} \quad [17]$$

A becült modell átviteli függvényéből könnyen leolvasható a periódusidő (T_i), amelyből már meghatározható a termikus kapacitás:

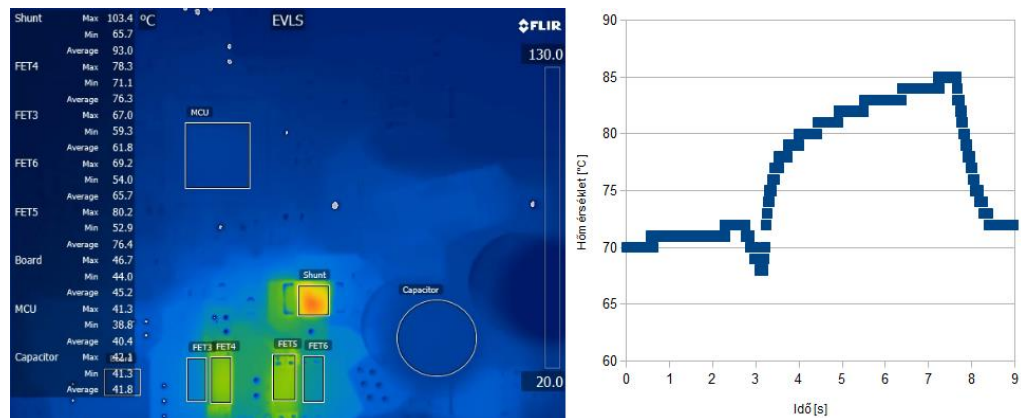
$$C_{th} = \frac{T_i}{R_{th}} = \frac{69,78s}{45,61 \frac{K}{W}} = 1,508 \frac{J}{K} \quad [18]$$

4.2.4 Hitelesítés

A feladatomban meghatározott modell hitelesítésére egy hőkamerás mérés szolgált. A központi vezérlőegység meghajtó fokozatára egy autófényszóróból álló műterhelést kapcsoltam. Az izzóban található két szál, a 60 W-os távolsági, illetve az 55 W-os tompított párhuzamosan volt kötve, így a teljes terhelés 115 W. Ez azt jelenti,

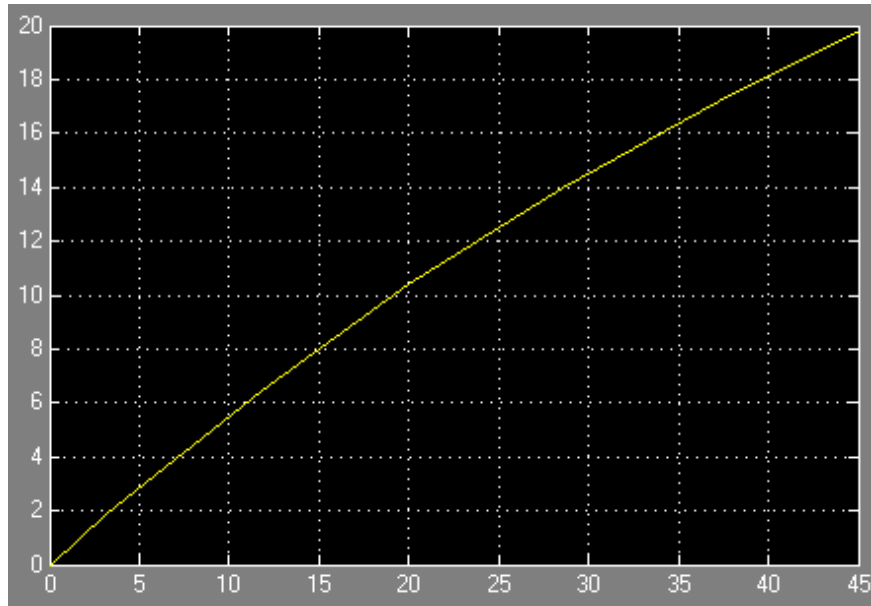
hogy meghajtás közben egy FET-en átfolyó áram, mivel a tápegység 12 V egyenfeszültséget szolgáltat, 9,58 A.

Az eredmény szimulálását a Matlab Simulink eszköztárával végeztem el. A virtuális mérési összeállítás egy egységugrásból, a hőmodell átviteli függvényét tartalmazó elemből, illetve egy oszcilloszkópból áll.



4.9. ábra A hőkamerás mérés eredményei

A 4.9. ábrán az egyik hőkamerás felvétel eredményei láthatóak. A meghatározott modell hitelesítéséhez a grafikonon látható felfutást szimulálom. A Simulink elemiben beállítandó paraméterek a következők. Az egységugrás értéke a műterhelést figyelembe véve 9,58 A, az idő kezdőértéke pedig 0 s. A mintavételi időre 10 s értéket állítottam be, hogy a grafikonról leolvasható 4,5 s-os gerjesztés a szimuláció egész végértékével, jelen esetben 45 s, könnyen megadható legyen. A hőkamera eredményét megjelenítő grafikonról leolvasható, hogy a FET hőmérséklete 68 °C-ról 85 °C-ra emelkedik. Ez 17 K hőmérsékletváltozást jelent.



9. ábra A Simulink szimuláció eredménye

A 4.10. ábrán látható a Simulink szimuláció eredménye. A grafikonról leolvashatjuk, hogy a modell alapján becsült hőmérsékletváltozás nagyjából 20 K. Ez a hőkamerás méréshez képest 3 K eltérést jelent.

Mivel a meghatározott modell a tranzisztorok biztonságát szolgáló közelítés, ezért ez a néhány fokkal történő hőmérséklet-túlbecsülés nem jelent problémát, a meghatározott modell implementálható a mikrokontrollerbe.

5 Összegzés

A szakdolgozatom készítése során egy elektronikusan mozgatható kormányoszlop vezérlőegységének néhány fontos tervezési lépését végeztem el. A fejlesztés komplexitásából adódóan különböző típusú feladatokkal foglalkoztam. A dolgozatomban bemutatott megoldások kitérnek mind a hardveres, mind a szoftveres fejlesztési lépésekre is, illetve egy felhasználandó alkatrész mérési, paraméterezési lépéseire is.

Először is létrehoztam egy tesztelő berendezést, amely képes a vezérlőegységben található mikrokontroller változóinak futásidőben történő megváltoztatására. A feladat megvalósítása során elengedhetetlen volt a fejlesztendő eszköz által használt autóipari kommunikációs protokoll, a LIN szabvány ismerete. Ezen kívül megismerkedtem egy, szintén az autóiparban használatos már létező diagnosztikai megoldással, az XCP protokollal. A tervezés során bemutattam a szükséges átküldendő változókat, paramétereket, illetve azok elhelyezését a kommunikációs ciklusban, valamint az így létrehozott diagnosztikai üzemmód be, illetve kikapcsolását. A megvalósítás során hangsúlyt fektettem a diagnosztikai üzenetek bővíthetőségére is. Dolgozatomban kitértem a szoftveres kezelőpanel és az üzenetek adatkapcsolatának elemzésére is. A tesztelő és diagnosztikai üzemmód mellett az általam megírt programban létrehozta egy kérést, amellyel a mikrokontroller saját inicializáló üzemmódját lehet aktiválni.

A következő fejezetben két hardver periféria egység tervezését, illetve méretezését mutattam be. Ezek a modulok a kormányoszlopot mozgató motorok és a vezérlőegységben található mikrokontroller közötti kapcsolatot biztosítják, így fejlesztésük során ezen berendezések fizikai paramétereinek pontos ismerete nélkülözhetetlen volt. A feladat elvégzése során megismerkedtem a járműiparban használatos inkrementális jeladókkal, különösképpen a jelen alkalmazásban is használatos HALL-szenzorral, illetve a felhasznált mikrovezérlő analóg-digitális bemenetének modelljével. Ezen ismeretek birtokában megterveztem a szenzor jeleit feldolgozó és átalakító áramkört, amely segítségével a jeladó már csatlakoztatható a mikrokontrollerhez. Valamint megterveztem egy feszültségmérő kapcsolást, ami a megadott paraméterek mellett alkalmas a mérendő feszültség és a mikrovezérlő analóg-

digitális bemeneteinek biztonságos összekapcsolására. A hardvertervezés, méretezés során elvégzett számításaim helyességének igazolásához a TINA nevű áramkör-szimulációs programot használtam. Ez különböző analíziseket biztosít a felhasználó számára. A helyes szimuláció elvégzéséhez megismerkedtem az analízisek paraméterezési lehetőségeivel.

Szakedolgozatom utolsó fejezetében a kormányoszlopokat meghajtó motorok vezérléséhez szükséges tervezérlésű tranzisztorok termikus modelljének meghatározását elemzem. Egy doktori disszertáció tanulmányozásával és elemzésével megismertem a feladat néhány lehetséges megoldását. Dolgozatomban a probléma felvázolása után bemutattam a felhasznált tranzisztor méréséhez használható paraméterének, csatorna ellenállásának tulajdonságait, a mérés elvét, illetve a használatos mérési elrendezéseket. A felvett eredmények alapján a Matlab program szimulációs és identifikációs eszköztárainak segítségével meghatároztam a beültetett tranzisztor modelljét, majd egy hőkameras mérés elvégzésével hitelesítettem a kapott paramétereket.

Irodalomjegyzék

- [1] *Íme az autó.* Benedek Attila. A mi világunk sorozat. Szendeczki Nyomda, Nyíregyháza 2008.
- [2] *Overview XCP.* Vector.
http://vector.com/portal/medien/solutions_for/xcp/Vector_XCP_Basics_EN.pdf V1.1 11/2011
- [3] *LIN Specification Package.* LIN Consortium, 2010. www.lin-subbus.org Revision 2.2 A. December 31. 2010.
- [4] *Szenzorok a gépjárművekben.* Robert Bosch GmbH. BOSCH Gépjárműtechnika "Sárga füzet sorozat". Maróti Könyvkiadó Kft., Bp. 2008.
- [5] *Height adjustment hoehenverstellung.* Bosch. Offer drawing anbotszeichn. 01 Release. 2011. 05. 26.
- [6] MC9S12ZVM-Family Reference Manual. Freescale. Preliminary. www.freescale.com Revision 1.1. 18 December, 2012.
- [7] *The SPICE book.* Andrei Vladimirescu. John Wiley & Sons, Inc. Malloy Lithographing, Inc., United States of America 1994.
- [8] *Teljesítmény tranzisztorok termikus ellenállásának mérése.* Kővári Péter. Doktori értekezés. 1982.
- [9] *BUK7K6R8-40E Product data sheet.* NXP Semiconductors.
http://www.nxp.com/documents/data_sheet/BUK7K6R8-40E.pdf
NXP B.V. 2013. 19 March 2013

Függelék

Matlab szkript

```
% termikus modell identifikáció
clc

% paraméterek
Rdso = 5.31; % mohm
Ids = 8.304; % A
Ts = 0.000501; % s

%egy dimenziós szűrő
windowSize = 1000;
szurt = filter(ones(1,windowSize)/windowSize,1,data(:,3)); %Uds [mV]

% számítások
Rds_elozetes = szurt / 8.3043; % ellenállásértékek számítása
%plot(Rds_elozetes)
Rds = Rds_elozetes(18100:918600);
a = Rds / Rdso; % a tényező számítása
T = (a - 0.86) * 157.9; % maghőmérséklet számítása egyenlettel

% x teng. skálázása
x = [0:length(Rds)-1]* Ts;

% gerjesztés beolvasása
gerj = data(18100:918600,2);

% ábrák
figure(1)
hold on
plot(data(:,3),'b')
plot(szurt,'g')
hold off

figure(2)
hold on
plot(x, Rds, 'b')
plot(x, a, 'g')
plot(x, T, 'r')
hold off
```

Inicializáló mód parancsai

```
private void InitToolStripMenuItem_Click(object sender, EventArgs e)
{
    // INIT ON
    if (Test)
    {
        Init = true;           // Init start
        MessData6[1] &= 0x7F; // Set init flags
        MessData7[7] |= 0x10;

        // Graphical feedback
        modeToolStripMenuItem1.ForeColor = Color.DarkGray;

        groupBox_ELSV_parameters.ForeColor = Color.DarkGray;
        label_PWM_duty_cycle.ForeColor = Color.DarkGray;
        label_Motor_speed_control.ForeColor = Color.DarkGray;
        label_Ramp_up.ForeColor = Color.DarkGray;
        label_Ramp_down.ForeColor = Color.DarkGray;
        label_Block_current_time_out.ForeColor = Color.DarkGray;
        label_Softstop_position.ForeColor = Color.DarkGray;
        maskedTextBox_PWM.BackColor = Color.WhiteSmoke;
        maskedTextBox_Motor_Speed.BackColor = Color.WhiteSmoke;
        maskedTextBox_Ramp_up.BackColor = Color.WhiteSmoke;
        maskedTextBox_Ramp_down.BackColor = Color.WhiteSmoke;
        maskedTextBox_Block_current_time_out.BackColor =
Color.WhiteSmoke;
        maskedTextBox_Softstop_pos.BackColor = Color.WhiteSmoke;

        groupBox_Car_parameters.ForeColor = Color.DarkGray;
        label_Terminal.ForeColor = Color.DarkGray;
        comboBox_Terminal.BackColor = Color.WhiteSmoke;
        label_Temperature.ForeColor = Color.DarkGray;
        maskedTextBox_Temperature.BackColor = Color.WhiteSmoke;
        label_Speed.ForeColor = Color.DarkGray;
        maskedTextBox_Speed.BackColor = Color.WhiteSmoke;

        groupBox_Manual_control.ForeColor = Color.DarkGray;
        groupBox_Memory_control.ForeColor = Color.DarkGray;
        groupBox_Easy_entry.ForeColor = Color.DarkGray;
    }
}
```

```
checkboxBox_Backward.Enabled = false;  
checkboxBox_Down.Enabled = false;  
checkboxBox_Forward.Enabled = false;  
checkboxBox_Up.Enabled = false;  
  
button_1.Enabled = false;  
button_2.Enabled = false;  
button_Memory.Enabled = false;  
button_Get_off.Enabled = false;  
button_Get_in.Enabled = false;  
button_ELSV_send.Enabled = false;  
button_Car_send.Enabled = false;  
    }  
}
```

```

// INIT OFF
if (((MessData7[7] & 0x10) == 0) && Init)
{
    Init = false;                // Init finished
    MessData6[1] |= 0x80;        // Set init flags
    MessData7[7] |= 0x10;

    if (OnlyExit) modeToolStripMenuItem1.ForeColor = Color.Black;

    //Graphical feedback
    groupBox_ELSV_parameters.ForeColor = Color.Black;
    label_PWM_duty_cycle.ForeColor = Color.Black;
    label_Motor_speed_control.ForeColor = Color.Black;
    label_Ramp_up.ForeColor = Color.Black;
    label_Ramp_down.ForeColor = Color.Black;
    label_Block_current_time_out.ForeColor = Color.Black;
    label_Softstop_position.ForeColor = Color.Black;
    maskedTextBox_PWM.BackColor = Color.White;
    maskedTextBox_Motor_Speed.BackColor = Color.White;
    maskedTextBox_Ramp_up.BackColor = Color.White;
    maskedTextBox_Ramp_down.BackColor = Color.White;
    maskedTextBox_Block_current_time_out.BackColor = Color.White;
    maskedTextBox_Softstop_pos.BackColor = Color.White;

    groupBox_Car_parameters.ForeColor = Color.Black;
    label_Terminal.ForeColor = Color.Black;
    comboBox_Terminal.BackColor = Color.White;
    label_Temperature.ForeColor = Color.Black;
    maskedTextBox_Temperature.BackColor = Color.White;
    label_Speed.ForeColor = Color.Black;
    maskedTextBox_Speed.BackColor = Color.White;

    groupBox_Manual_control.ForeColor = Color.Black;
    groupBox_Memory_control.ForeColor = Color.Black;
    groupBox_Easy_entry.ForeColor = Color.Black;

    checkBox_Backward.Enabled = true;
    checkBox_Down.Enabled = true;
    checkBox_Forward.Enabled = true;
    checkBox_Up.Enabled = true;

    button_1.Enabled = true;
    button_2.Enabled = true;
    button_Memory.Enabled = true;
    button_Get_off.Enabled = true;
    button_Get_in.Enabled = true;
    button_ELSV_send.Enabled = true;
    button_Car_send.Enabled = true;
}

```