

TMDK-dolgozat

**Egyenáramú motor modellezése Matlab-ban és fordulatszám-
szabályozása mikrokontrollerrel**

Tüű-Szabó Boldizsár

**villamosmérnöki MSC.
automatizálási szakirány**

Konzulens: Prof. Dr. habil. Kuczmann Miklós

Győr, 2013 ősz

Összefoglaló

A TDK-dolgozatom témája egy egyenáramú motor modelljének megalkotása, és egy PWM-szabályozó elkészítése és szimulálása a motor fordulatszámának szabályozására.

Az egyenáramú motorok működését 2 differenciálegyenlet írja le. Ezek felhasználásával megalkottam a motor matematikai modelljét a Matlab-ban. A motor matematikai leírásában szereplő paraméterek (motor ellenállása, induktivitása, motorállandó, és viszkózus súrlódási együttható) nem ismertek, ezért ezeket mérésekkel kellett meghatározni, amit több feszültségen elvégeztem. A mérési eljárásokat és a mérési eredményeket a 2. fejezetben foglaltam össze. A motorállandó és a viszkózus súrlódási együttható számításához szükség volt a motor fordulatszámának mérésére. A motorhoz készítettem egy tárcsát, amelyen 2 mágneset helyeztem el egymással szemben. Készítettem egy Hall-szenzoros kapcsolást, amelynek kimeneti feszültsége mágneses tér hatására 0-ra vált. Ezeket a lefutó éleket számolja meg egy mikrokontroller, amiből a motor fordulatszáma kiszámítható. A várakozásoknak megfelelően a motorállandóra és a viszkózus súrlódási együtthatóra is állandó érték adódott.

A mért paraméter-értékek felhasználásával Matlab-ban ábrázoltam a szögsebesség és a motoráram időfüggvényeit.

A motor fordulatszámára egy PWM-szabályozót készítettem. A PWM-jelet a mikrokontroller állítja elő, a kitöltési tényezőt egy potméterrel lehet beállítani 0,39% és 100% között. A PWM-szabályozó modelljét elkészítettem Matlab-ban, és szimulációkon ábrázoltam a motor szögsebességének és áramának alakulását.

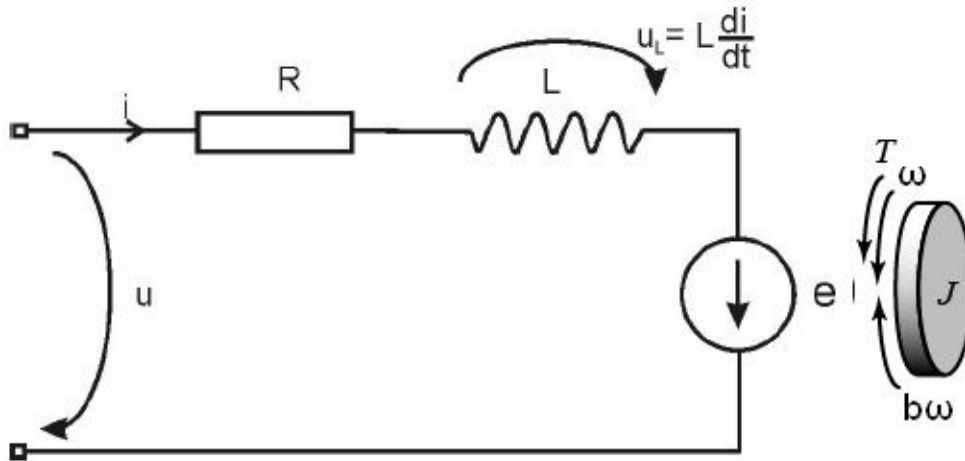
Tartalomjegyzék

1. Az egyenáramú motor modelljének meghatározása.....	1
1.1 Matematikai leírás.....	1
1.2 A modell megalkotása Simulink-ben.....	3
2. Az egyenáramú motor paraméterértékeinek meghatározása mérésekkel.....	6
2.1 Rotor ellenállásának mérése.....	6
2.2 Rotor induktivitásának mérése	7
2.3 A motorállandó és a viszkózus súrlódási együttható meghatározása	9
2.3.1 MCUniversity Toolstick fejlesztőpanel	9
2.3.2 A Hall-szenzoros kapcsolat.....	10
2.3.2 A motor fordulatszámának mérése	11
3, A motor szimulációja Simulink-ban.....	15
4, Az egyenáramú motor PWM sebességszabályozása.....	17
4.1 A PWM szabályozó ismertetése.....	17
4.1.1 A PWM jelet előállító program	18
4.2 A PWM szabályozó szimulációja Matlab-ban	20
4.2.1 A szimulációs eredmények 50%-os kitöltési tényező esetén	22

1. Az egyenáramú motor modelljének meghatározása

1.1 Matematikai leírás [1], [2], [3]

Az egyenáramú motorok soros RL-körrel modellezhetőek.



1. ábra Az egyenáramú motor, mint elektromos áramkör

- $U[V]$: a forgórészre kapcsolt feszültség, amelynek hatására a rotor forogó mozgást végez
- $R[\Omega]$: a rotor ellenállása
- $L[H]$: a rotor induktivitása
- $e[V]$: a rotorban visszaindukált feszültség

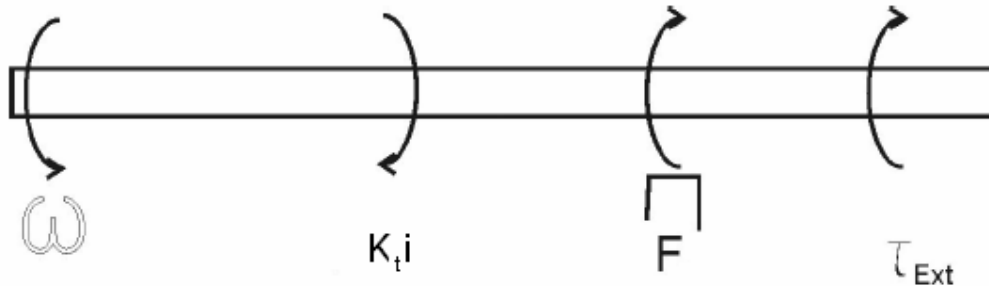
Lenz törvény következményeként, a rotorban visszaindukált feszültség (e) pedig arányos a rotor fordulatszámaival (ω)

$$e = K_e \cdot \omega \quad , \text{ ahol } K_e \text{ a motorra jellemző back EMF konstans, mértékegysége } \left[\frac{V}{\text{rad} \cdot s} \right]$$

Kirchoff II. törvényét felírva a soros RL-körre az egyenáramú motor elektromos viselkedését leíró egyenletet kapjuk:

$$u = R \cdot i + L \frac{di}{dt} + K_e \cdot \omega \quad (1)$$

A motor mechanikai viselkedését leíró egyenlet meghatározásához ismernünk kell a rotorra ható forgatónyomatékokat.



2. ábra A rotorra ható erők

- a motor által kifejtett nyomaték (T), amelyet a Biot-Savart törvény alapján lehet meghatározni

A motor által kifejtett forgatónyomaték (T) arányos a rotoron átfolyó árammal (i).

$$T = K_t \cdot i, \text{ ahol } K_t \text{ a nyomatékállandó, mértékegysége } \left[\frac{Nm}{A} \right]$$

- a motorban fellépő súrlódási erők [F]

$$F = B \cdot \omega, \text{ ahol } B \text{ a viszkózus súrlódási együttható, mértékegysége } \left[\frac{Nm \cdot s}{rad} \right]$$

- a motorra ható külső nyomaték, amit az általa meghajtott gép fejt ki (T_{ext})

Newton II. törvényét felhasználva a motor mechanikai viselkedésére az alábbi egyenletet kapjuk:

$$J \cdot \frac{d\omega}{dt} = K_t \cdot i - F - T_{ext} \quad (2)$$

ahol J a motor forgórészének tehetetlenségi nyomatéka, mértékegysége $[kg \cdot m^2]$

Összegezve a motor dinamikai modelljének meghatározásához az elektromos (1) és a mechanikai (2) viselkedést leíró egyenleteket kell felhasználni.

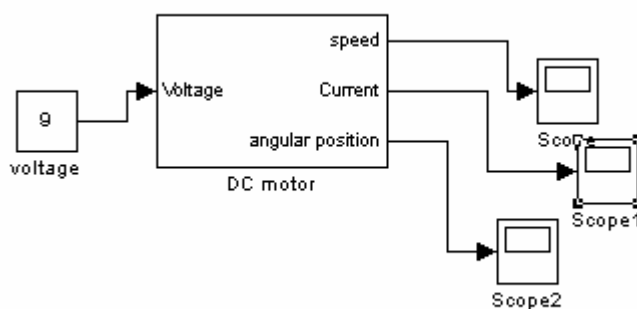
1.2 A modell megalkotása Simulink-ben [4]

Az 1.1 fejezetben ismertetett, a motor dinamikai viselkedését leíró 2 differenciálegyenletet felhasználva elkészítettem a motor modelljét Simulink környezetben.

A Simulink egy grafikus programcsomag, amely blokk diagramon alapuló szimuláció végrehajtására képes. Egy Simulink szimuláció két lépésből áll. Először létrehozuk a modellt, majd szimulációkat végzünk a kívánt paraméterbeállítások mellett.

Az általam elkészített modell 1 bemenettel, és 3 kimenettel rendelkezik. (3. ábra)

A bemenet az egyenáramú motorra kapcsolt feszültség, a kimeneteken a szögsebességet, a motor áramát illetve a pozíciót lehet ábrázolni.

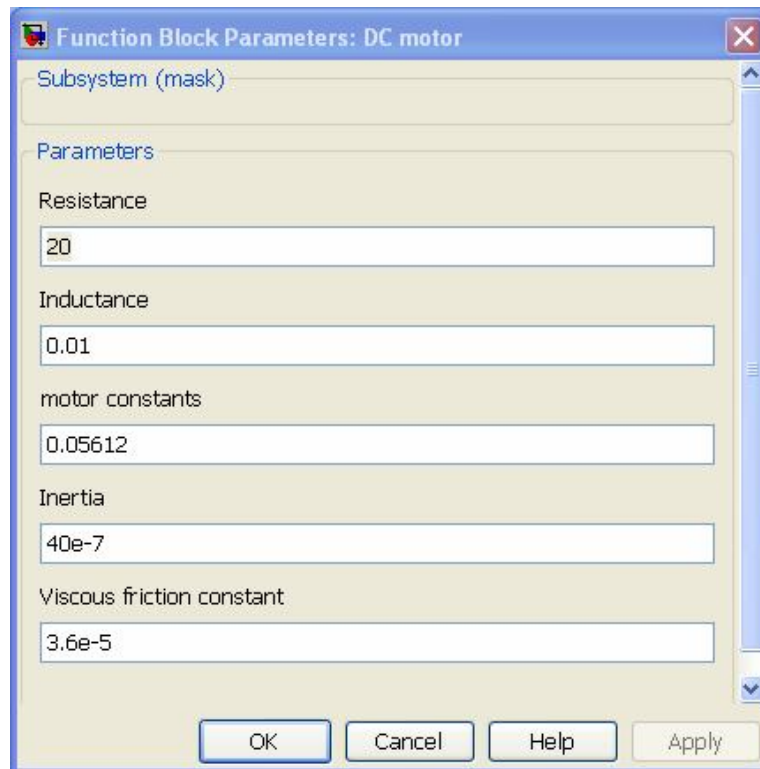


3. ábra Az egyenáramú motor doboza

A motor dobozára kattintva egy ablak jelenik meg, amelyben a motor paramétereit kell megadnunk. (rotor ellenállása, induktivitása, motorállandók, inercia, viszkózus súrlódási együttható). (4. ábra)

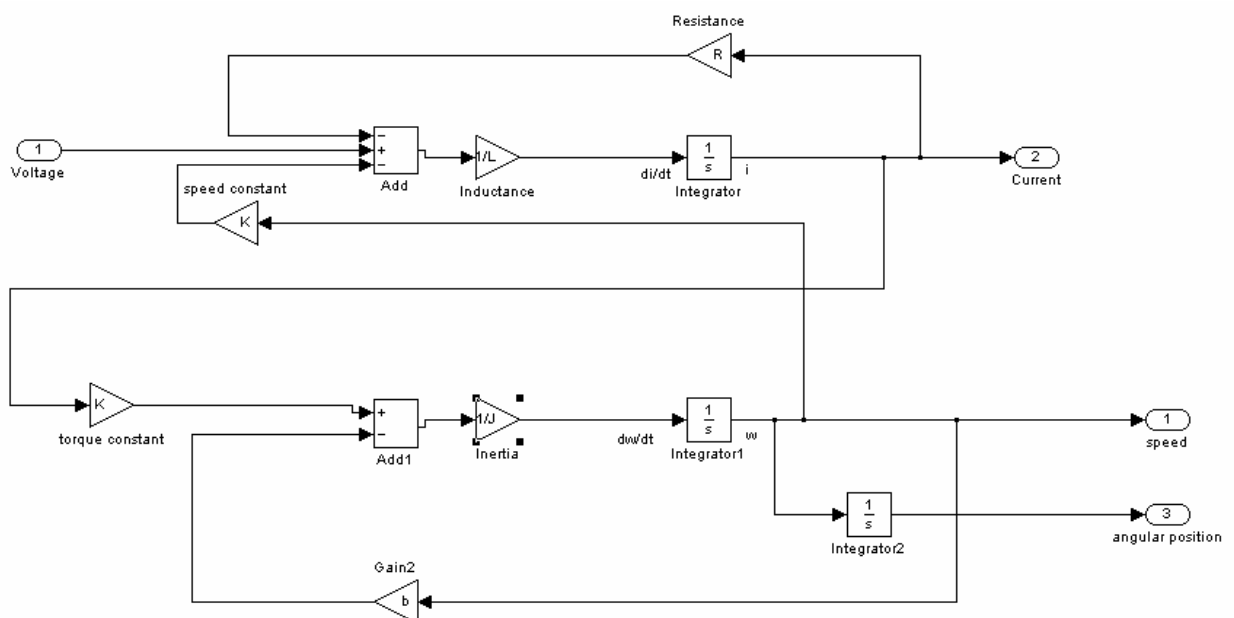
Mivel a back EMF konstans (K_e) és a nyomatékállandó (K_t) értéke az SI-rendszerben megegyezik, ezért itt a két állandót 1 paraméterrel (K : motorállandó) helyettesítettem.

A paramétereket SI-egységekben kell megadni: az ellenállás Ω -ban, az induktivitást H-ban, a motorállandókat $\frac{Nm}{A}$ -ben, a tehetetlenségi nyomatékot $kg \cdot m^2$ -ben, a viszkózus súrlódási együtthatót pedig $\frac{Nm \cdot s}{rad}$ -ban.



4. ábra A motor paramétereinek megadása

A doboz tartalmát megnézve látható, hogy a motor leírásához összegző (add), erősítő (gain) blokkokat (gain) és integrátorokat használtam. (5. ábra)



5. ábra A modell felépítése

A motoráram meghatározásához az (1)-es számú egyenletből indulunk ki. Ezt átrendezve az alábbi egyenletet kapjuk:

$$\frac{di}{dt} = \frac{1}{L}(U - R \cdot i - K \cdot \omega)$$

Mindkét oldalt integrálva, előáll a motoráram.

A másik kimenet, a szögsebesség meghatározásához a motor mechanikai (2) egyenletéből kell kiindulni, amit átrendezve az alábbi egyenletet kapjuk:

$$\frac{d\omega}{dt} = \frac{1}{J}(K \cdot i - b\omega)$$

Ebből szintén integrálással kapjuk meg a szögsebességet.

A 3. kimenet a szögpozíció, a szögsebesség integráljaként adódik.

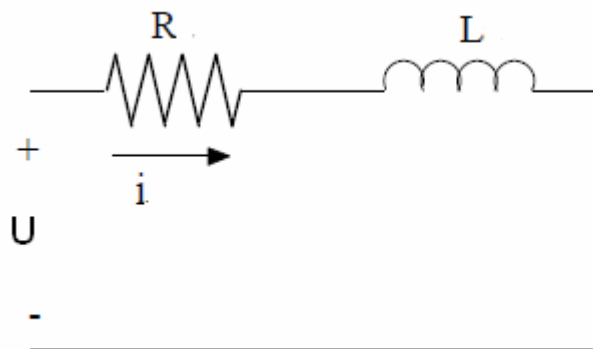
2. Az egyenáramú motor paraméterértékeinek meghatározása mérésekkel [5], [6]

A motorról katalógusadatok nem álltak rendelkezésemre, ezért a motor paramétereinek értékeit nekem kellett mérésekkel meghatároznom. Ebben a fejezetben a mérési módszereket, és a mérés során kapott eredményeket írom le.

2.1 Rotor ellenállásának mérése

A motor elektromos viselkedését leíró (1)-es számú egyenlet a motorra kapcsolt feszültség, a rotor ellenállása, a rotor induktivitása és a visszaindukált feszültség közötti kapcsolatot írja le. A rotor ellenállásának mérését lefogott forgórészszel kell elvégezni. Ekkor mivel a szögsebesség zéró, nem keletkezik visszaindukált feszültség.

Leszorított forgórész esetén az egyenáramú motor a 6. ábrán látható áramkörre egyszerűsödik.



6. ábra Az áramkör lefogott forgórész esetén

Egyenáramú körben a tekercs kivezetései között nincs feszültség, tehát a kapcsok azonos potenciálon vannak.

A motoráramot megmérve, Ohm törvényével számítható a rotor ellenállása.

A mérési eredményeimet az alábbi táblázatban rögzítettem.

A forgórészre kapcsolt feszültség (U)	Motoráram (I)	Rotor ellenállása (R)
2V	0,099A	20,2Ω
3V	0,153A	19,6Ω
4V	0,201A	19,9Ω
5V	0,246A	20,3252Ω

1. táblázat a rotor ellenállásának meghatározása

A mérési eredmények alapján a rotor ellenállása 20Ω.

2.2 Rotor induktivitásának mérése

Az egyenáramú motor egy soros RL-körnek tekinthető. Ha forgórészre egy kisjelű váltakozó feszültséget kapcsolunk, és az áram csúcsértékét mérjük, akkor ebből az induktivitás értéke kiszámolható.

Az eredő impedancia : $Z = R + j\omega L$

Ezt átírva a következő alakba: $Z = R \cdot \left(1 + \frac{j\omega L}{R} \right)$, a törésponti körfrekvenciára az $\omega_0 = \frac{R}{L}$

összefüggést kapjuk.

Feszültség csúcsértéke	Frekvencia	Áram csúcsértéke
1V	10Hz	0,049A
1V	20Hz	0,049A
1V	50Hz	0,049A
1V	100Hz	0,048A
1V	200Hz	0,042A
1V	300Hz	0,036A
1V	320Hz	0,035A

2. táblázat Mérési eredmények 1V csúcsérték esetén

A törésponti frekvencia meghatározásához meg kell keresni azt a frekvenciát, amikor az áram értéke 3dB-el ($\sqrt{2}$ részére) csökken. Ez a frekvencia 320 Hz.

Az $\omega_0 = \frac{R}{L}$ törésponti körfrekvenciára vonatkozó összefüggést felhasználva, az induktivitás értékére 9,95mH kaptam.

Feszültség csúcsértéke	Frekvencia	Áram csúcsértéke
2V	10Hz	0,1A
2V	20Hz	0,099A
2V	50Hz	0,099A
2V	100Hz	0,094A
2V	200Hz	0,088A
2V	300Hz	0,072A
2V	320Hz	0,071A

3. táblázat Mérési eredmények 2V csúcsérték esetén

2V-os feszültség esetén is a törésponti frekvencia 320 Hz körül alakult.

A rotor induktivitásának értéke 10mH=0,01H.

2.3 A motorállandó és a viszkózus súrlódási együttható meghatározása

Ezen motorparaméterek kiszámításához szükséges a motor fordulatszámának mérése. Ehhez egy Hall-szenzoros kapcsolást készítettem. A Hall-szenzor kimeneti jelét egy mikrokontroller dolgozza fel.

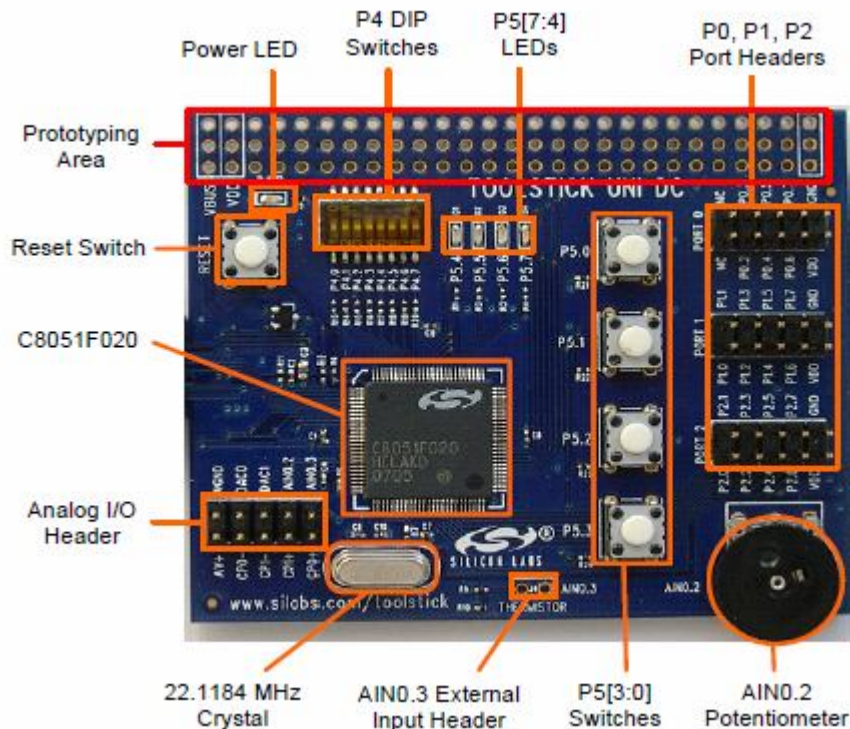
2.3.1 MCUniversity Toolstick fejlesztőpanel [7]

A panel 3 elemből áll:

- ToolStick base adapter: Ez biztosítja az USB debug interfészt és adatkapcsolati útvonalat biztosít a Windows PC és a C8051F020 mikrokontroller között.
- ToolStick University Daughter Card (7. ábra): C8051F020 típusú mikrovezérlő-egységet tartalmaz

A mikrovezérlő jellemzői:

- 64 kB Flash programmemória
 - 4352 B RAM
 - 1 db 12 bites (ADC0) és 1 db 8 bites (ADC1) analóg-digitális átalakító
 - 2 db 12 bites digitális-analóg átalakító
 - 2 db analóg komparátor
 - Feszültségreferencia
 - Beépített hőmérsékletszenzor
 - 5 db 16 bites időzítő (T0-T4)
 - 5 db PCA csatorna
- USB-bővítőkábel



7. ábra ToolStick University Daughter Card

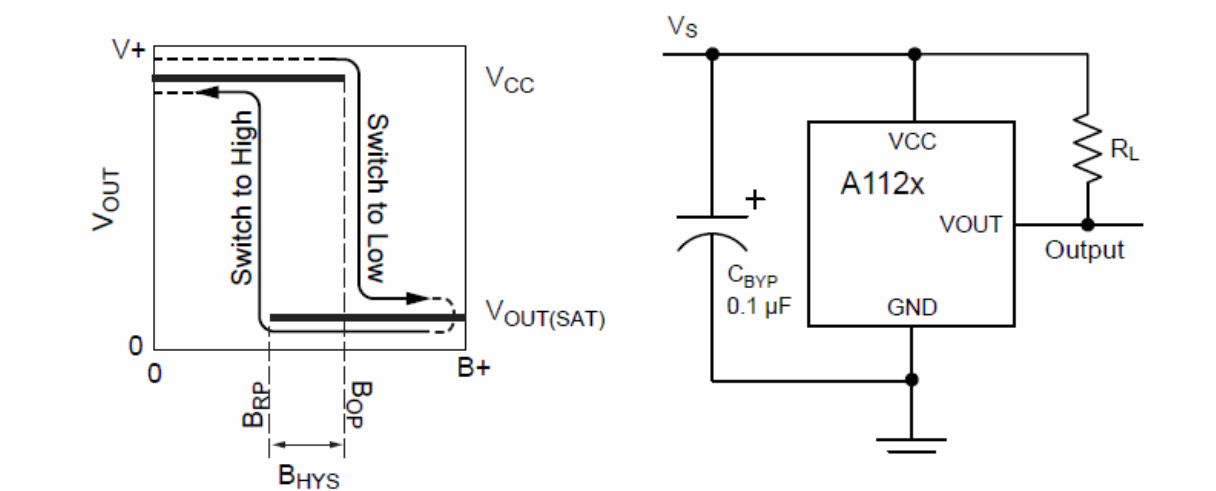
A mikrovezérlő programozását a Silicon Labs Integrated Development Environment (IDE) fejlesztőkörnyezetben végeztem el. A programot C-nyelven írtam meg, a lefordításhoz a Keil C fordítóját használtam.

2.3.2 A Hall-szenzoros kapcsolás

A Hall szenzor működése a felfedezőjéről (E. H. Hall) elnevezett fizikai elven alapul. Ha egy vezetőben vagy félvezetőben áram folyik, és azt mágneses térbe helyezzük, akkor az áramot hordozó részecskékre Lorentz-erő hat, ami azzal jár, hogy a vezető két oldalán feszültségkülönbség lesz. Ezt a feszültséget Hall-feszültségnek nevezik.

A kapcsolásban egy Allegro A1120 típusú unipoláris Hall-szenzor került felhasználásra. Az érzékelő unipoláris és logikai kimenetű. Az unipolaritás azt jelenti, hogy a déli pólus (pozitív mágneses mező) hatására bekapcsol (kimenet alacsony szintű lesz), és akkor kapcsol ki, ha a pozitív mágneses mező megszűnik (kimenet magas szintű lesz). (8. ábra)

A kapcsolásban a 0,1 μ F-os kondenzátor a külső zajok szűrésére szolgál.



8. ábra a kimenet feszültségének alakulása és a kapcsolási rajz

2.3.2 A motor fordulatszámának mérése [8]

Készítettem egy tárcsát, amelyre 2 mágneset ragasztottam egymással szemben. A tárcsával felszerelt egyenáramú motor a 9. ábrán látható.



9. ábra a motor a tárcsával

Az IDE környezetben C-nyelven megírt program a motor forgása során keletkező impulzusokat számolja.

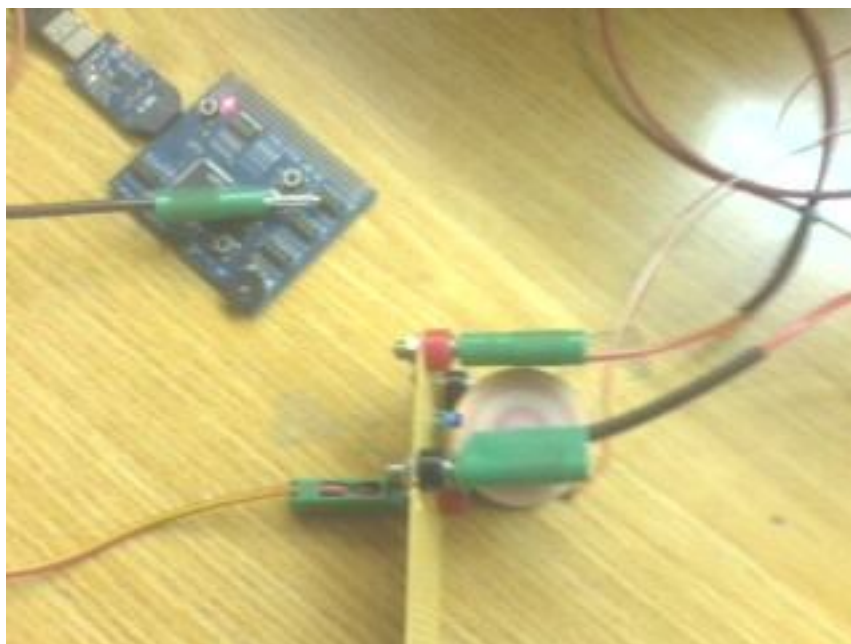
Amikor a Hall-szenzor pozitív mágneses térbe kerül (az egyik mágnes elhalad az érzékelő előtt), akkor az érzékelő kimenete magas szintről alacsonyra vált. A program ezeket a lefutó éleket számolja, ezzel a mikrovezérlő Timer2 16 bites időzítő/számlálóját használtam fel.

A P0.2-es be-/kimenetet a Timer2 külső bemeneteként (T2 external input) konfiguráltam a XBR1 regiszter megfelelő beállításával (XBR1=0x40). A Timer2-at pedig számlálóként használok. (T2CON=0x0F)

Erre a bemenetre kötöttem rá a Hall-szenzor kimeneti jelét. Amikor ez a bemenet magas szintről alacsony szintre vált, a Timer2 egy megszakítást generál. A Timer2-höz az 5-ös megszakítási prioritás tartozik.

Megszakításkor egy megszakítási rutin fut le, amely az impulzusok számát (tl változó) 1-el növeli.

A T változóban tároljuk a 60 másodperc alatt keletkező impulzusok számát, az időtartam letelte után ennek a száma már nem változik.. A 60 másodperc mérésére a Timer0 időzítő/számlálót használtam fel, amit 16 bites időzítő/számlálóként állítottam be (TMOD=0x01). Az időzítő órajele a belső 16 MHz-es oszcillátor órajelének 12-es része, ami azt jelenti, hogy a Timer0 értéke $0,75\mu\text{s}$ -onként nő 1-el. Ahhoz, hogy az időzítő 1 ms-ként csorduljon túl, az időzítő regiszterébe $65536 - \frac{1}{0,75 \cdot 10^{-3}} \approx 64203$ értéket kell tölteni, ami hexadecimálisan 0xFACB (TL0=0xCB; TH0=0xFA). Egy for ciklusban 60000-szer fut le ez a túlsordulás, ami 60 másodpercet jelent.



10. ábra a mérési elrendezés

A mérési eredményeimet az alábbi táblázatban rögzítem.

Motor feszültsége	Motor árama	60 másodperc alatti impulzusszám	Fordulatszám	Szögsebesség
3,3 V	0,031 A	912	456ford/perc	47,75 rad/s
3,8V	0,034 A	1057	528,5ford/perc	55,34 rad/s
9V	0,087 A	2481	1240,5ford/perc	129.9 rad/s

4. táblázat mérési eredmények

A motorállandók kiszámításához a motor elektromos viselkedését leíró egyenletből kell kiindulni. A stabil állapot elérése után a motor árama konstans lesz, ezért annak deriváltja 0 lesz, így az egyenlet az alábbivá egyszerűsödik, ami egyetlen ismeretlent $=K_e$ tartalmaz.

$$u = R \cdot i + K_e \cdot \omega$$

Amiből: $K_e = \frac{u - Ri}{\omega}$ összefüggés adódik

Motor feszültsége	Motorállandó értéke $K = K_e = K_t$
3,3V	$\frac{3,3 - 20 \cdot 0,031}{47,75} = 0,0561 \frac{V}{rad \cdot s}$
3,8V	$\frac{3,8 - 20 \cdot 0,034}{55,34} = 0,0564 \frac{V}{rad \cdot s}$
9V	$\frac{9 - 20 \cdot 0,087}{129,9} = 0,0558 \frac{V}{rad \cdot s}$

5. táblázat motorállandó számítása

A táblázatból kiolvasható, hogy a várakozásoknak megfelelően a motorállandók a motor feszültségétől független, állandó értékek.

A viszkózus súrlódási együttható kiszámításához a mechanikai viselkedést leíró egyenletből kell kiindulni. A stabil állapot elérése után a szöggyorsulás értéke nulla lesz, így a B a

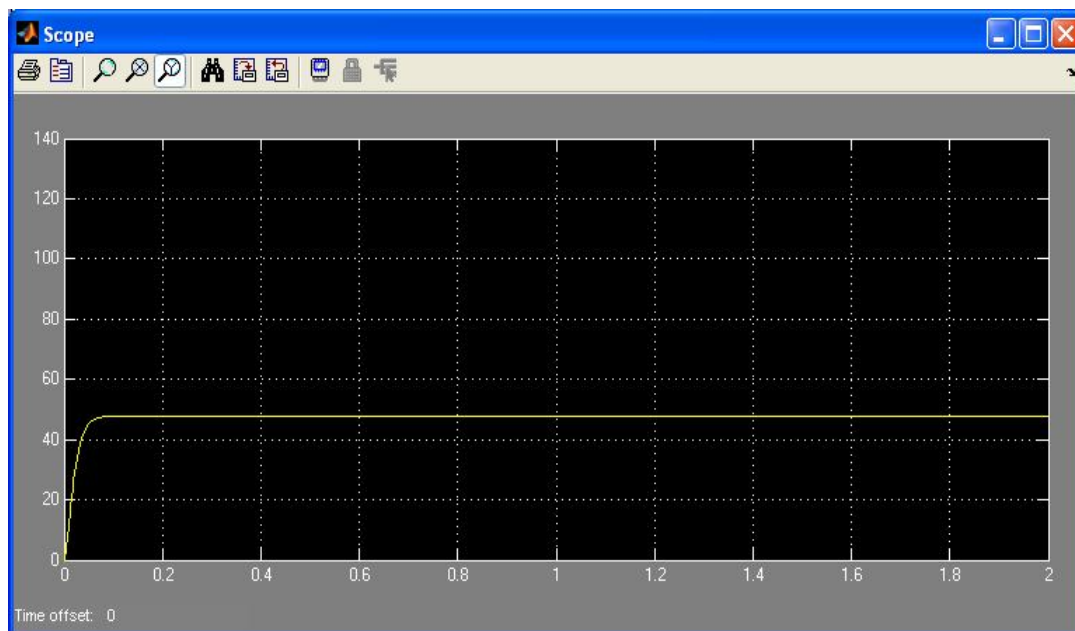
$$B = \frac{K_t \cdot i}{\omega} \text{ összefüggéssel számítható.}$$

Motor feszültsége	Viszkózus súrlódási együttható
3,3V	$\frac{0,0561 \cdot 0,031}{47,75} = 3,64 \cdot 10^{-5} \frac{Nm \cdot s}{rad}$
3,8V	$\frac{0,0564 \cdot 0,034}{55,34} = 3,56 \cdot 10^{-5} \frac{Nm \cdot s}{rad}$
9V	$\frac{0,0558 \cdot 0,087}{129,9} = 3,73 \cdot 10^{-5} \frac{Nm \cdot s}{rad}$

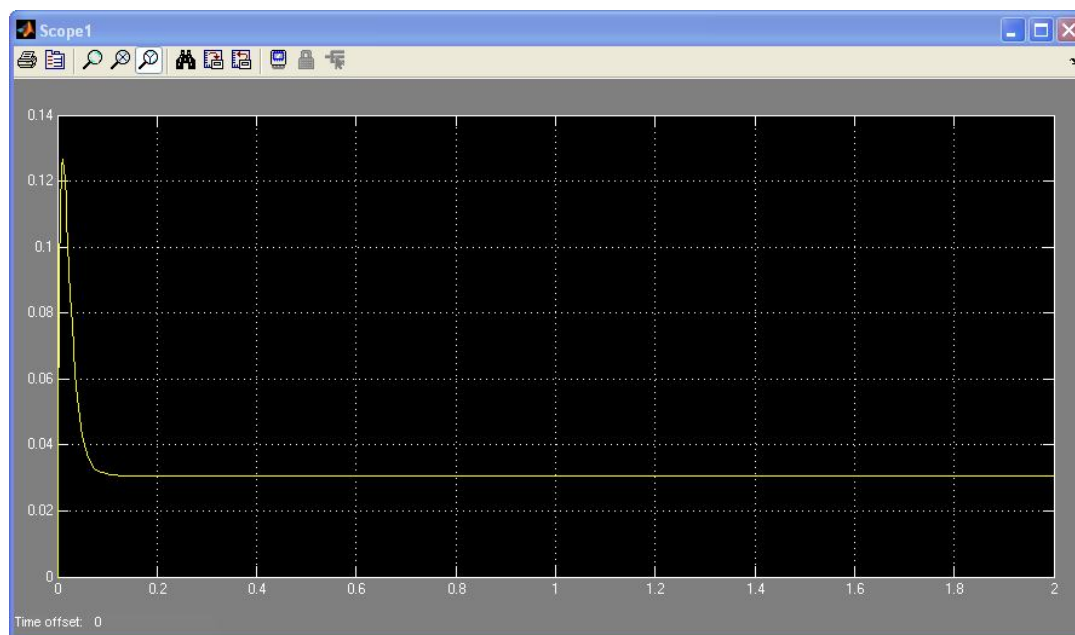
6. táblázat viszkózus súrlódási együttható számítása

3, A motor szimulációja Simulink-ban

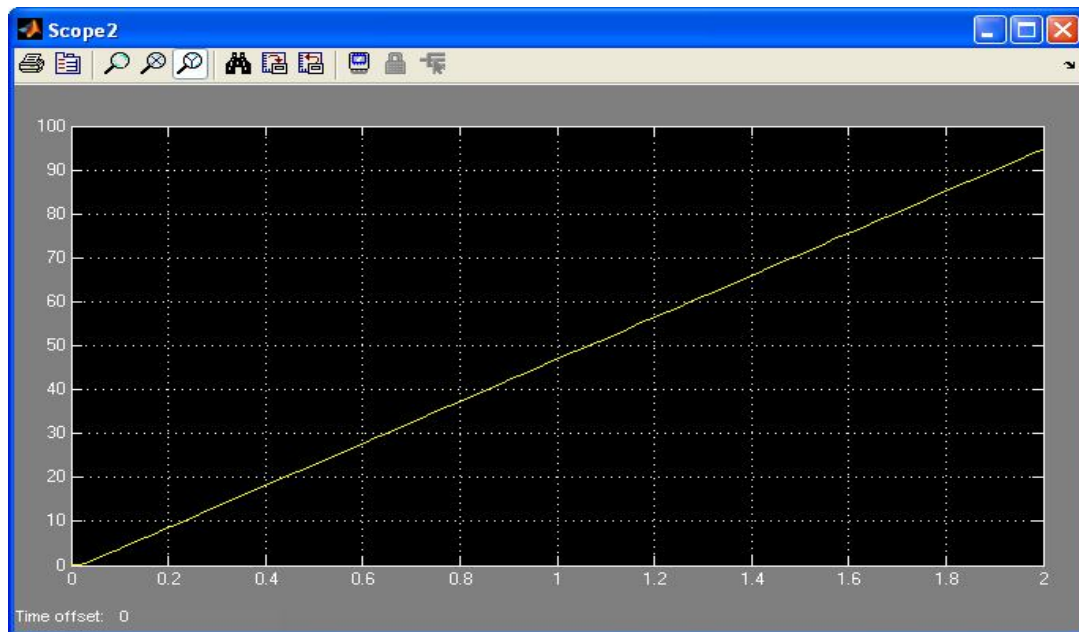
A motormodellben a motor paramétereinek megadása után szimulálni lehet a motor viselkedését, időfüggvényen lehet ábrázolni a szögsebességet, motoráramot és szögpozíciót.



11. ábra a motor szögsebessége $U=3,3V$ esetén



12. ábra a motor árama $U=3,3V$ esetén



13. ábra a motor szögpozíciója $U=3,3V$ esetén

A motor áramánál megfigyelhető, hogy a motor indításakor ($\omega = 0$) kapjuk a legnagyobb meredekséget ($\frac{di}{dt} = \frac{U}{L}$). Egy maximum elérése után az áram értéke csökkenni kezd és beáll egy állandósult értékre ($\frac{U - K \cdot \omega_{all}}{R}$, ahol ω_{all} a szögsebesség állandósult értéke).

A motor szögsebesség a tranziens lecsengése után egy állandó értéket vesz fel (11. ábra). A időfüggvényekből látható, hogy a méréssel meghatározott áram és szögsebesség állandósult értéke megegyezik a szimulációval kapott értékekkel.

4, Az egyenáramú motor PWM sebességszabályozása

A PWM szabályozás (impulzusszélesség moduláció) egy az egyenáramú motoroknál használt eljárás a fordulatszám szabályozására. Ennek lényege, hogy a motorra egy állandó értékű feszültséget kapcsolgatunk ki és be, állandó frekvenciával. A be- és a kikapcsolt állapot arányát a teljes perióduson belül kitöltési tényezőnek nevezzük. A motor fordulatszáma annál nagyobb, minél nagyobb a kitöltési tényező nagysága.

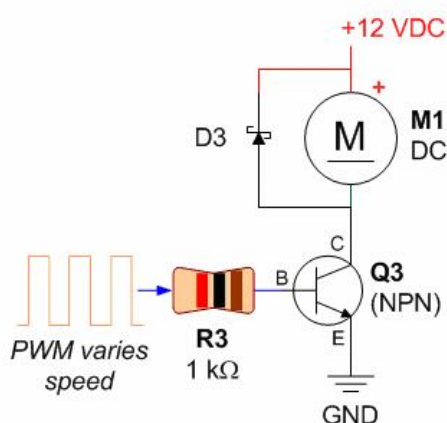
4.1 A PWM szabályozó ismertetése

A PWM jelet a mikrokontroller állítja elő, amelynek kitöltési tényezőjét egy potenciométerrel lehet szabályozni. A kapcsolgathoz egy BC639 npn típusú tranzisztort használtam fel.

Amikor a PWM jel bekapcsolt állapotban van, akkor a kollektor-emitter feszültség közel zérus (kisebb mint 0,2 V), tehát a motoron közel a teljes tápfeszültség megjelenik.

A PWM jel kikapcsolásakor a bázisáram zérusra csökken, a tranzisztor kikapcsolt állapotba kerül. Ekkor a motoron eső feszültség zérus.

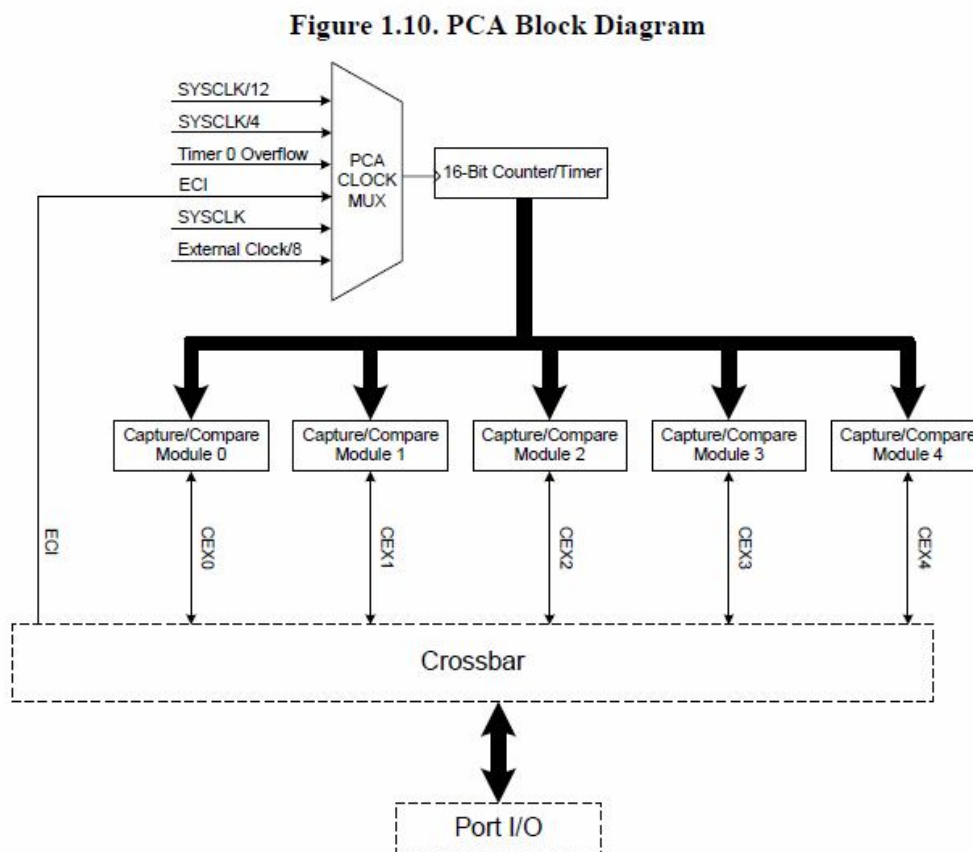
A motorral párhuzamosan kötött D3 dióda szerepe a PWM impulzusok kikapcsolásakor keletkező feszültségtüskék levágása.



14. ábra a PWM szabályozó kapcsolási rajza és a megépített áramkör

4.1.1 A PWM jelet előállító program [7], [8]

A mikrovezérlő tartalmaz egy PCA (Programmable Counter Array) modult, ami egy 16 bites időzítő 5 programozható capture/compare modullal (16. ábra).



15. ábra a PCA blokkdiagramja

A modulokat többféle üzemmódban is lehet alkalmazni. A Modul0-t PWM jel előállítására használok, ezért 8 bites PWM modulként konfiguráltam ($PCA0CPM0 = 0xC2$). A Modul0 kimenetének (CEX0, ezen jelenik meg a PWM jel) a P0.6-es be-/kimenetet állítottam be ($XBR0 = 0x0E$).

A modul időalapja szintén többféle lehet, én a 16 MHz-es oszcillátor frekvenciájának 12-ed részét választottam ($PCA0MD = 0x00$).

A PWM jel kitöltési tényezőjét a $PCA0CPH0$ regiszter értéke határozza meg az alábbi

egyenlet szerint: $D = \frac{256 - PCA0CPH0}{256}$, ahol D a kitöltési tényező

A legnagyobb kitöltési tényező 100% PCA0CPH0=0x00 esetén, a legkisebb kitöltési tényező 0,39% PCA0CPH0=0xFF esetén.

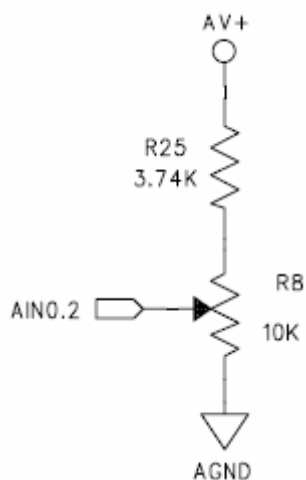
A kitöltési tényezőt a mikrovezérlőn található potméterrel lehet szabályozni, a potméter feszültségét az 12 bites szukcesszív approximációs ADC0 egység digitalizálja. A potméter az AIN0.2 bemenetre kapcsolódik, ezt kellett az ADC0 bemeneteként választani. Az analóg tápfeszültség értéke (AV+) 3,3V, ezért a potenciométer véghelyzetében az AIN0.2 lábon 2,4 V feszültség jelenik meg. Feszültségreferenciaként a belső feszültségreferenciát választottam (2,4V), tehát az A/D átalakító 0V-hoz 0-át, 2,4 V-hoz (potméter véghelyzete) 4095 digitális értéket rendel.

A kitöltési tényező a potméterrel lehet állítani az alábbi módon:

$$PCA0CPH0 = 255 - \frac{Result}{16}$$

Result: az A/D konverzió értéke

A 0 digitális értékhez 0,39%-os kitöltési tényező, a 4095 értékhez 100% kitöltési tényező tartozik.



16. ábra a potenciométer kapcsolási rajza

Az AD0BUSY bit 1-be állítása indítja el az A/D konverziót, amit 10 ms-onként megismétlünk. A 10 ms-os késleltetést a Timer2 egység állítja elő egy szubrutin segítségével. A Timer2 időalapja a 16 MHz-es oszcillátor frekvenciájának 12-es része. A 2.3.2 fejezetben leírtakhoz hasonlóan kell kiszámolni a Timer2-höz tartozó regiszterekbe írandó értékeket

(RCAP2L = 0xCB; RCAP2H = 0xFA), ekkor az időzítő 1 ms-onként csordul túl. Egy int típusú *ms* változóba írjuk be a kívánt késleltetést ms-okban, és minden túlszordulásokra 1-el csökkentjük az értékét. Amikor az értéke 0 lesz, akkor lesz vége a késleltetésnek.

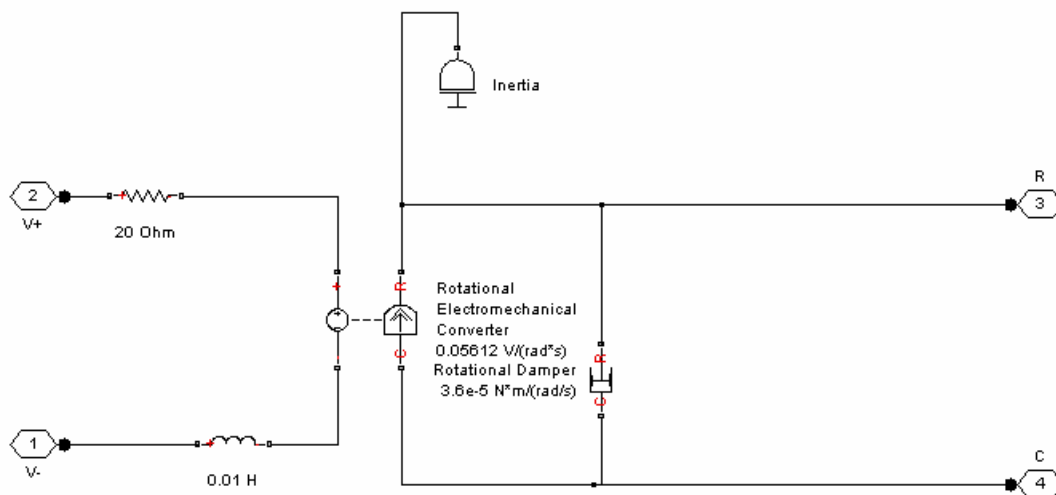
4.2 A PWM szabályozó szimulációja Matlab-ban [9]

Matlab-ban elkészítettem a szabályozó modelljét, és grafikonon ábrázoltam a motor fordulatszámának és áramának időfüggvényeit.

A 17. ábrán a motor modellje látható.

egyenáramú motor, ami az alábbi elemekből tevődik össze:

- ellenállás, amelynek értéke $20\ \Omega$
- induktivitás, értéke $0,01\ \text{H}$
- elektromechanikai átalakító: ez modellezi a visszaindukált feszültséget és a motor által kifejtett nyomatékot
- forgómozgás csillapítása: a viszkózus súrlódási erőt modellezi
- a motor forgórészének tehetetlenségi nyomatéka

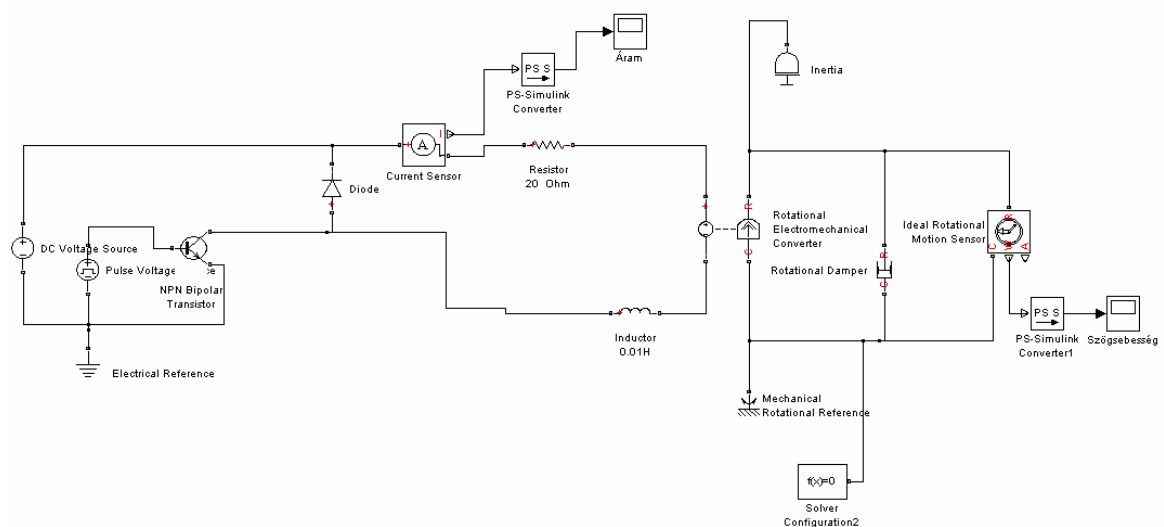


17. ábra a motor modellje

A modell az alábbi elemekből áll:

- 12 VDC egyenáramú feszültségforrás
- Impulzusgenerátor: ez állítja elő a PWM jelet. Mivel a PWM modul időalapja a 16 MHz-es oszcillátor frekvenciájának 12-ed része, ezért a periódusidőt

$$\frac{1}{1333333,33} \cdot 256 = 1,92 \cdot 10^{-4} \text{ s}$$
-ra kell beállítani.
- npn típusú tranzisztor
- dióda
- motor modellje
- áramszenzor
- forgásérzékelő
- 2 oszcilloszkóp: az áram és szögsebesség ábrázolására
- 2 db PS-Simulink konverter: az érzékelő kimeneti jelét mértékegység nélküli Simulink jellé konvertálja

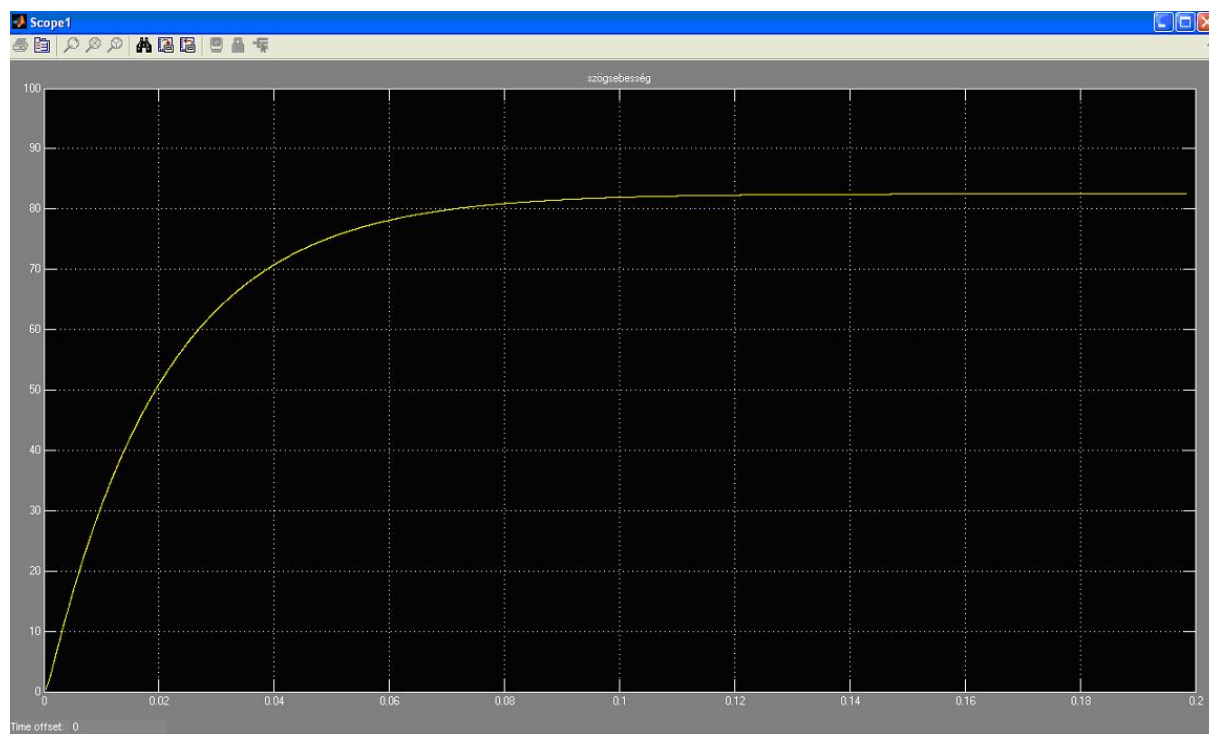


18. ábra a PWM szabályozó modellje

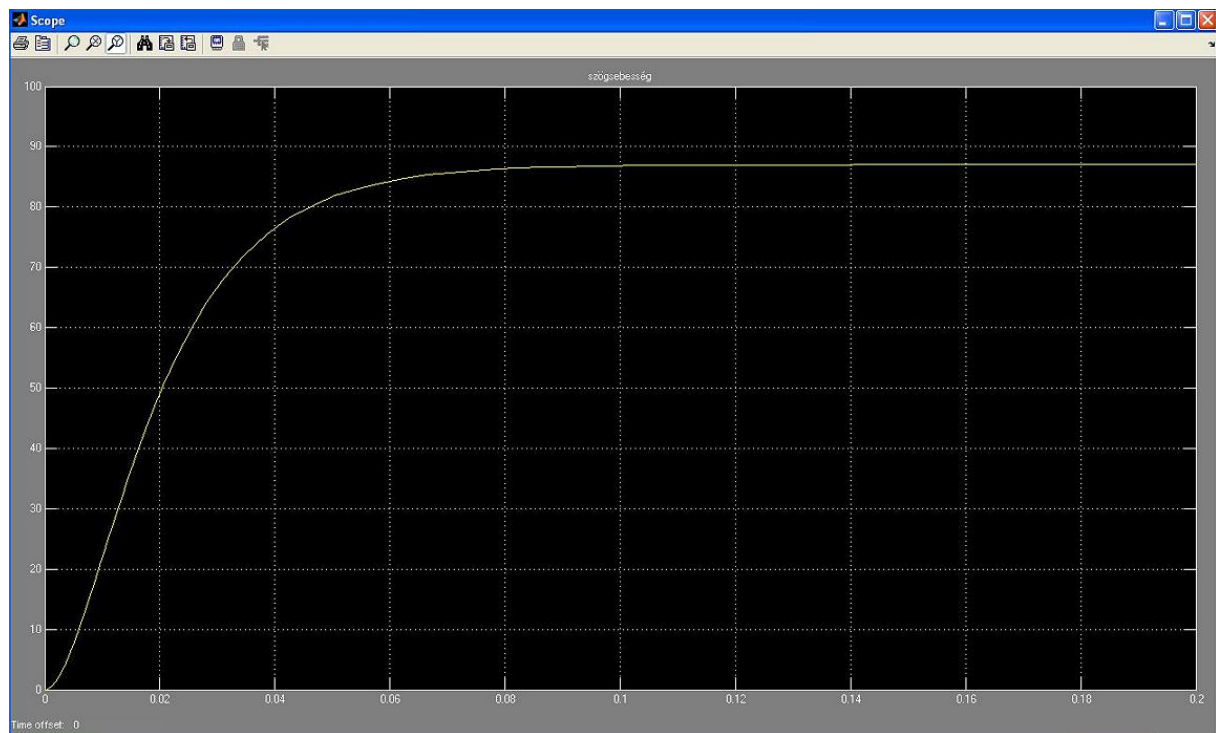
4.2.1 A szimulációs eredmények 50%-os kitöltési tényező esetén

A szimulációnál a tápfeszültséget 12 VDC-re, a kitöltési tényezőt 50%-ra állítottam be.

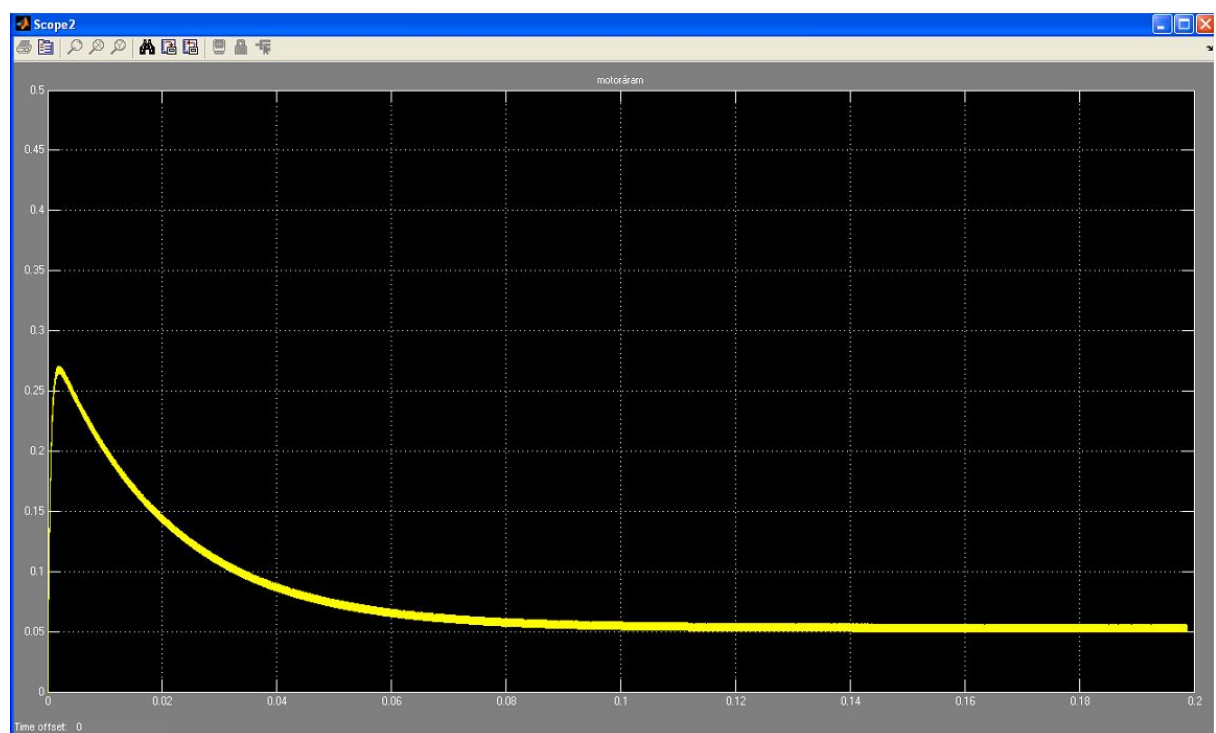
A szimulációs eredményekből jól látható, hogy az eredmények közel azonosak azzal az esettel, amikor 6 V feszültséget kapcsolunk folyamatosan a motorra. Az eltérést az okozza, hogy a PWM szabályozónál a PWM-jel bekapcsolt állapotában a tápfeszültségnél kb. 0,2 V-tal kisebb feszültség (kollektor-emitter feszültség) kapcsolódik a motorra.



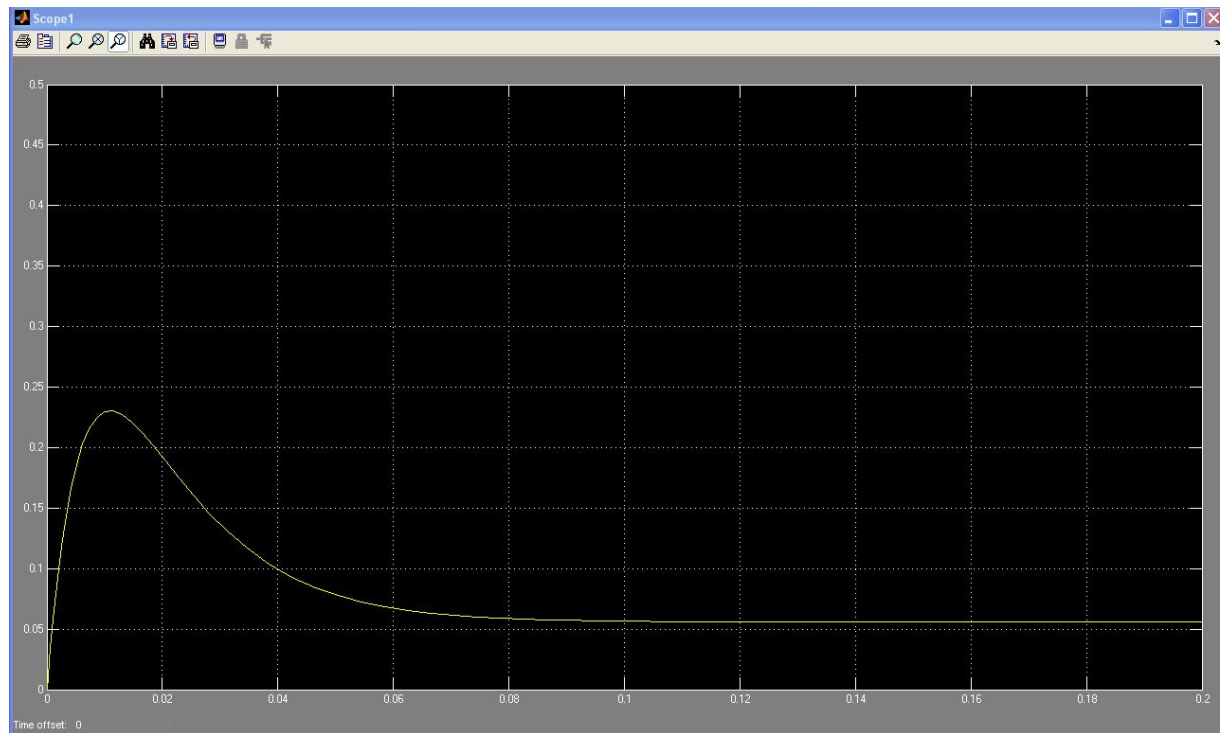
19. ábra a motor szögsebessége 12 VDC tápfeszültség és 50% kitöltési tényező esetén



20. ábra a motor szögsebessége 6 VDC tápfeszültség esetén



21. ábra a motor árama 12 VDC tápfeszültség és 50% kitöltési tényező esetén



22. ábra a motor árama 6 VDC tápfeszültség esetén

A motor árama PWM-szabályozás esetén 2 érték között ingadozik, ez abból adódik, hogy a PWM-jel kikapcsolt állapotában a motor árama csökkenni fog, majd a PWM-jel bekapcsolásakor ismét növekedni. Minél kisebb a PWM-jel periódusideje annál kisebb ez az ingadozás (21. ábra).

Összefoglalás

Munkám során tanulmányoztam az egyenáramú motorok matematikai leírásának módját, és ezt felhasználva megépítettem a motor modelljét. Elvégeztem a motor paramétereinek mérését több feszültségen is. A várakozásoknak megfelelően a mérésekből a paraméterekre állandó értékek adódtak mindegyik feszültségen. A feladat során egy általam korábban nem használt mikrokontrollerrel és fejlesztőkörnyezettel ismerkedtem meg. A mikrokontrollert 2 feladat céljából használtam fel: a motor fordulatszámának méréséhez és a PWM-szabályozó PWM-jelének előállításához.

Nagyon sok tapasztalatot és új ismeretet szereztem a feladat során, a tanulmányaim alatt megszerzett tudásomat sikerült kamatoztatni.

Tervezem egy olyan program elkészítését, amellyel a motor párhuzamosan futtatható a modellel.

Irodalomjegyzék

[1] Keviczky László-Bars Ruth - Hetthéssy Jenő-Barta András - Bányász Csilla. SZABÁLYOZÁSTECHNIKA. UIVERSITAS-GYŐR Kht., Győr, 2006 93-99 old.

[2] Lantos Béla Irányítási rendszerek elmélete és tervezése 1., Egyváltozós szabályozások Akadémiai Kiadó, Budapest, 2005 49-52 old.

[3] DC Motor Speed: System Modeling
<http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SystemModeling#3>

[4] DC Motor Speed: Simulink Modelling
<http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SimulinkModeling>

[5] Measuring motor parameters
<http://support.ctc-control.com/customer/learning/younkin/motorParameters.pdf>

[6] DC Motor Transfer Function Estimation by Explicit Measurement
http://coecsl.ece.illinois.edu/ge320/ge320_Lab2.pdf

[7] Toolstick University Daughter Card User's Guide

[8] C8051F020/1/2/3 User's Manual

[9] DC Motor Model
<http://www.mathworks.com/help/physmod/elec/ug/example--modeling-a-dc-motor.html>

Mellékletek

1, A Hall-szenzor impulzusait számoló progrma

```
#include "compiler_defs.h"
#include "C8051F020_defs.h"

unsigned int tl;

void Timer_Init()
{
    T2CON = 0x0F;           //counter function
    TH2 = 0;
    TL2 = 0;
}

void Port_IO_Init()
{
    XBR1=0x40;              //P0.2 T2EX Input
    P0MDOUT=0x00;
}

void T2_ISR (void) interrupt 5
{
    tl++;                   //impulse count
    T2CON=0x0F;
}

void main(void)
{
    OSCICN = 0x07;          //Internal Oscillator typical
    frequency is 16 MHz
    WDTCN = 0xde;           // Disable watchdog timer
    WDTCN = 0xad;
    XBR1 = 0x4A;
    P0MDOUT=0x00;
```

```

Timer_Init();
ET2 = 1;          /* Enable Timer 2 Interrupts */
TR2 = 1;          /* Start Timer 2 Running */
EA = 1;           /* Global Interrupt Enable */

while (1)
{

    long  time;
    int T;

    while (time<60000)
        {
            for(time=0;time<=60000;time++)
                {
                    TMOD=0x01;
                    TL0=0xCB;
                    TH0=0xFA;
                    TR0=1; // start timer.
                    while(TF0==0); // run until TF0 turns to 1
                    TR0=0; // stop timer
                    TF0=0; // reset the flag
                    T=tl;  // impulse in 60 seconds
                }
        }

    }

}

```

2, A PWM jelet előállító program

```

#include "compiler_defs.h"
#include "C8051F020_defs.h"
//-----
// 16-bit SFR Definitions for 'F02x
//-----

//-----
// Global Constants
//-----

#define SYSCLK      16000000

```

```

#define SAMPLE_RATE 50000          // Sample frequency in Hz
#define INT_DEC 256                // Integrate and decimate ratio
#define SAR_CLK 2500000            // Desired SAR clock speed

#define SAMPLE_DELAY 10            // Delay in ms before taking sample

//-----
// Function Prototypes
//-----

void OSCILLATOR_Init (void);
void PORT_Init (void);
void ADC0_Init (void);
void TIMER3_Init ();
void ADC0_ISR (void);
void Wait_MS (unsigned int ms);

//-----
// Global Variables
//-----

long Result;                      // ADC0 decimated value

//-----
// main() Routine
//-----

void main (void)
{

    WDTCN = 0xde;                  // Disable watchdog timer
    WDTCN = 0xad;

    OSCILLATOR_Init ();            // Initialize oscillator
    PORT_Init ();                  // Initialize Port

    // configure the PCA
    PCA0MD = 0x00; // disable CF interrupt
    // PCA time base = SYSCLK / 12
    PCA0CPL0 = 0; // initialize PCA PWM value
    PCA0CPH0 = 0;
    PCA0CPM0 = 0xC2; // CCM0 in 8-bit PWM mode
    PCA0CN = 0x40; // enable PCA counter

    ADC0_Init ();                  // Init ADC
    AD0EN = 1;                     // Enable ADC
    EA = 1;                        // Enable global interrupts

```



```

    AMX0SL = 0x02;           // Select AIN0.2 pin as ADC mux input

    ADC0CF = (SYSCLK/SAR_CLK) << 3;
    ADC0CF |= 0x00;          // PGA gain = 1 (default)

    EIE2 |= 0x02;            // enable ADC interrupts
}

void ADC0_ISR (void) interrupt 15
{
    AD0INT = 0;
    Result = ADC0;

}

void Wait_MS(unsigned int ms)
{
    CKCON = 0x00;             // use SYSCLK/12 as timebase
    RCAP2L  = 0xCB;
    RCAP2H  = 0xFA;
    ET2 = 0;                  // Disable Timer 2 interrupts
    TR2 = 1;                  // Start Timer 2

    while(ms)
    {
        TF2 = 0;              // Clear flag to initialize
        while(!TF2);          // Wait until timer overflows
        ms--;                  // Decrement ms
    }

    TR2 = 0;                  // Stop Timer 2
    TF2=0;
    AD0BUSY = 1;              // Start AD conversion
}

```