

Miskolci Egyetem
Gépészmérnöki és Informatikai Kar
Automatizálási és Infokommunikációs
Intézeti Tanszék

2019/2020. tanév
I-II. félév

Irányítástechnika c. tantárgy
előadásának és gyakorlatának ütemterve
BSC Járműmérnök szakos hallgatók részére
Tervezés-gyártás szakirány
GEVAU258-B

Tárgynév:	Beágyazott rendszerek			
Rövid név:	Beágy. rendsz.	Kód	GEVAU519B	
Angol név:	Embedded Systems			
Tanszék:	Automatizálási Tanszék			
Tárgyfelelős:	Dr. Vásárhelyi József vajo@uni-miskolc.hu			
Előtanulmányok:	Digitális rendszerek I.,II.,III	Kódja:	GEVAU 505B	
Kredit:	5	Követelmény:	Aláírás, Kollokvium	
Heti óraszámok	Előadás: 2	Gyakorlat:	2	Labor: 2
Oktatási cél:	A digitális rendszerek és a beágyazott rendszerek tervezésében alkalmazott elvek és elméleti ismeretek elsajátítása			
Tárgy tartalom:	Az Irányítástechnika tantárgy célja megismertetni a hallgatókat a rendszerszemlélet alapjaival, a rendszerek osztályozásának kritériumaival. Ezen felül cél a digitális technika funkciók, az azokat megvalósító elemek, valamint a logikai hálózatok és mikrovezérlők tervezési alap módszerek megismerése. Továbbá megismerteti a hallgatókat a rendszer- és irányításmélet, módszertan klasszikus és korszerű elemeivel, hogy ezeket megfelelően tudják kezelni és alkalmazni közlekedési és jármű rendszerek irányításában. A tantárgy tematikája a következő témaköröket öleli fel: Irányításmélet alapfogalmai. A rendszer fogalma. Rendszerek tulajdonságai és osztályozása. A rendszer- és irányításmélet feladatai. Az irányítás fogalma. Nyílt hurkú és zárt hurkú irányítási rendszerek.			
Irodalom:	1. Ajtonyi I, Digitális Rendszerek, Miskolci Egyetem, Miskolc, 2002, ISBN963-661-399-5, pp.322 2. Bokor J., Gáspár P., : Irányítástechnika. Jegyzet, Typotex Kiadó, Budapest, 2008. 3. F. Vahid, T. Giravis, Embedded System Design a Unified Hardware/Software Introduction, John Wiley & Sons, 2002, ISBN 0-471-38678-2, pp. 324. 4. R. Isermann, Digital Control Systems I., Springer-Verlag, 1989, ISBN 3-540-50266-1, pp. 335. 5. R. S. Burns, Advanced Control Engineering, Butterworth-Heinemann, ISBN 0780651008, 2001, pp. 450.			
Mintatantervi elhelyezkedés szakok szerint				
Szak	Szakirány/sáv	Tantervi modul-tantervi kód	Mintatantervi félév	Választhatóság
Villamosmérnöki Szak	Elektronikus tervezés és gyártás		6	kötelező
Jellemző oktatási módok				
Oktatási nyelv:	Magyar, angol			
Előadás:	Minden hallgatónak előadás, számítógépes vetítés és tábla			
Gyakorlat:	Laboratóriumi és tantermi gyakorlatok			
Labor:	Maximum 16 fős csoportokban, Digitális rendszertechnikai laboratóriumban vezetett gyakorlatok, önálló mérések és feladatok teljesítésével.			
Évközi feladatok, zárthelyik:	Kétszer 2x1 órás évközi zárthelyi dolgozat. Egy tervezési feladat megoldása önálló terv-feladat keretében, jegyzőkönyvvel.			
Lezárási feltételek:	Gyakorlatokon aktív részvétel; az előírt tervezési feladatok teljesítése; a két évközi zárthelyi dolgozat eredményes megírása; A mérési jegyzőkönyv beadásának a tanszéki feladatbeadás határideje a következő gyakorlat kezdete, beadási mód: elektronikus; értékelés 1-5.-ig; Az évközi munka értékelése: Zárthelyi dolgozatok eredménye legalább elégséges > 60%, Gyakorlati feladatok önálló teljesítése legalább elégséges > 60%; - 24-28 elégséges, 28-32 közepes 32-36 jó, 36-40 jeles			
Ütemterv				
1.	Ea: Determinisztikus, eseményvezérelt, diszkrét állapotú, statikus rendszerek. Logikai változók, alaplőveletek, kifejezések, függvények. Kanonikus alakok, minimalizálás. Lab: Cypress PSOC lab 1.			
2.	Ea: Kombinációs hálózatok minimalizálása, függvények megvalósítása funkcióblokkok segítségével.			

	Lab: Kombinációs hálózatok megvalósítása Cypress PSOC lab2.
3.	Ea: Szinkron sorrendi hálózatok, tervezése, statikus viselkedése és tranziensei. Diszkrét eseményű rendszerek. Determinisztikus, véges állapotú automaták. Lab: Sorrendi hálózatok megvalósítása Cypress PSOC lab3.
4.	Ea: Moore és Mealy automaták. Determinisztikus, idő vezérelt, diszkrét állapotú, dinamikus rendszerek. Analóg jelek digitalizálása. Az A/D és a D/A átalakítók alaptípusai. Lab: Állapotgép tervezés Cypress PSOC lab4.
5.	Ea: Mikroprocesszorok, mikrovezérlők, számítógép modellek Lab: Analóg és digitális jelek gyakorlati példa Cypress PSOC lab 5.
6.	Ea: Stabilitáselmélet (stabilitás feltételei, zárt és visszacsatolt rendszerek stabilitása). Bevezetés az állapotter-elméletbe (állapotter reprezentációk, transzformációk). Állapotter reprezentációk tulajdonságai, állapotegyenletek megoldása. Állapot visszacsatolás. Állapot megfigyelő. Számítógéppel irányított rendszerek. Diszkrét irányítások. Lab: Egyéni feladat hardver és szoftver fejlesztés.
7.	Ea: Zárt szabályozási körök minőségi jellemzői. Lab: Egyéni feladat készítése, jegyzőkönyvvel. Feladatbeadás a 9. héten.
8.	Ea: Robusztus stabilitás. Lab: Egyéni feladat hardver és szoftver fejlesztés.
9.	Ea: Memória szerepe a beágyazott rendszerekben. Memóriakezelés. Külső és belső memóriák kezelése FPGA illetve SOC rendszerekben.. Lab: Egyéni feladat hardver és szoftver fejlesztés.
10.	Ea: Bevezetés az állapotter-elméletbe (állapotter reprezentációk, transzformációk). Állapotter reprezentációk tulajdonságai, állapotegyenletek megoldása. Lab: Egyéni feladat hardver és szoftver fejlesztés.
11.	Ea: Állapot visszacsatolás. Állapot megfigyelő. Lab: Egyéni feladat hardver és szoftver fejlesztés.
12.	Ea: Számítógéppel irányított rendszerek. Diszkrét irányítások. Lab: Egyéni feladat hardver és szoftver fejlesztés.
13.	Ea: Motorvezérlés, mint irányítási. Léptetőmotorok és váltóáramú motorok vezérlése szabályzása. Tervezési példa: Egyenáramú motor irányítása. Lab: Pótlás
14.	Ea: konzultáció, zárthelyi Lab: Pótlás

Intézetigazgató

Tárgyfelelős:

Dr. Trohák Attila
egyetemi docens

Dr. Vásárhelyi József
egyetemi docens

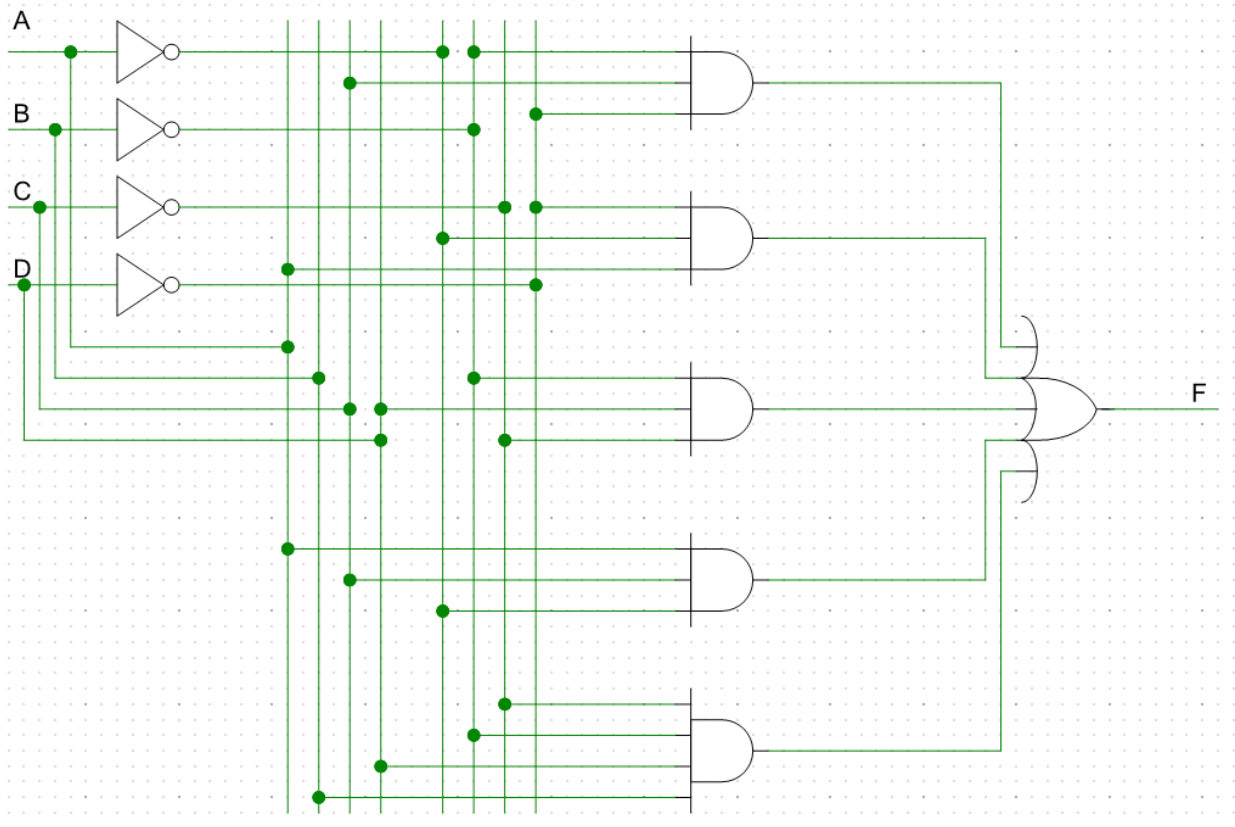
Irányítástechnika 1. Zárthelyi dolgozat minta

GEVAU 258-B

Név:

Neptun:

1. Adott az alábbi logikai kapcsolás. Írja fel a kapcsolás algebrai alakját (2p). Alakítsa át az algebrai alakot a De-Morgan azonosságot használva NAND-NAND függvénné (1p). Ábrázolja a függvényt NAND és Inverter kapuk segítségével (2p).



Megoldási útmutató: A kapcsolási rajz tartalmaz egy OR4 kaput, amihez 5 AND4 kaput kötöttünk. Megvizsgáljuk, hogy az ÉS kapuk bemenete milyen jel csatlakozik. Ily módon megkapjuk a kimeneti függvényt.

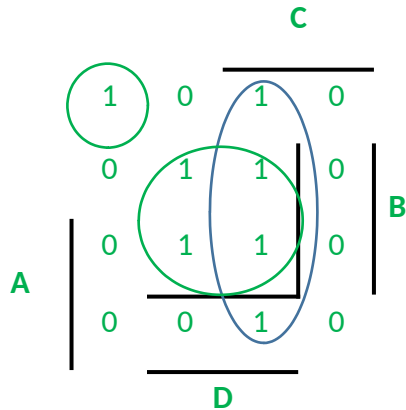
$$F = (\overline{B} \cdot \overline{C} \cdot \overline{D}) + (\overline{D} \cdot \overline{A} \cdot A) + (\overline{B} \cdot \overline{D} \cdot \overline{C}) + (A \cdot C \cdot \overline{A}) + (\overline{C} \cdot \overline{B} \cdot B \cdot D)$$

Útmutató: egyszerűsítés után a feladat De Morgan azonosság alkalmazása és a NAND kapus hálózat rajzolása.

- 2 Adott az alábbi sorszámos diszjunktív alak. Egyszerűsítse és Hazárd mentesítse a függvényt K-V táblában(2+1p).Ábrázolja az egyszerűsített függvényt NAND-NAND rendszerben (2p).

$$F = \sum_{ABCD}^4 (0,3,5,7,11,13,15)$$

Megoldás:



$$F = ABCD + CD + BD = \overline{\overline{ABCD} * \overline{CD} * \overline{BD}}$$

3. Adott az alábbi szinkron ismétlő számláló. Valósítsa meg a számlálót D-flip-flop elemek és multiplexerek használatával. (helyes állapot átmeneti táblázat 1p, függvények helyes felírása 2p, helyes ábrázolás 2p).

6, 0, 9, 15, 5, 12, 11, 10, 14, 3, 2, 4, 7, 1, 13, 8

Felírjuk az állapottáblát. A D flip-flop esetében a a következő állapot az lesz ami a D bemeneten található (gerjesztés/vezérlés). A D flip-flop kimenetei vezérlik a 4 multiplexert, aminek a kimenetei a megfelelő D flip-flop bemenetére kötünk. A multiplexerek bemeneti értéke a megfelelő vezérlési értékre kötjük (0, 1).

Adott az alábbi kód átalakító. Írja fel a kód átalakító kimeneti függvényeit (ábrázolni nem kell). Helyes igazság táblázat 1p, helyes KV egyszerűsítés és felírás 2p. Ügyeljen a Hazárd mentesítésre!(1p)

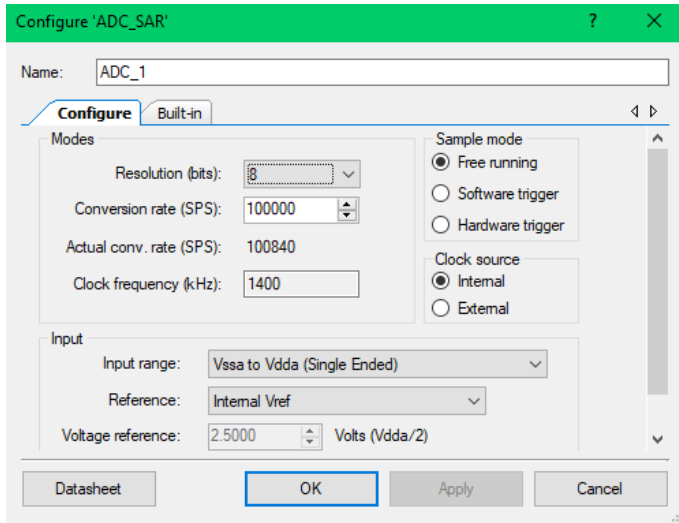
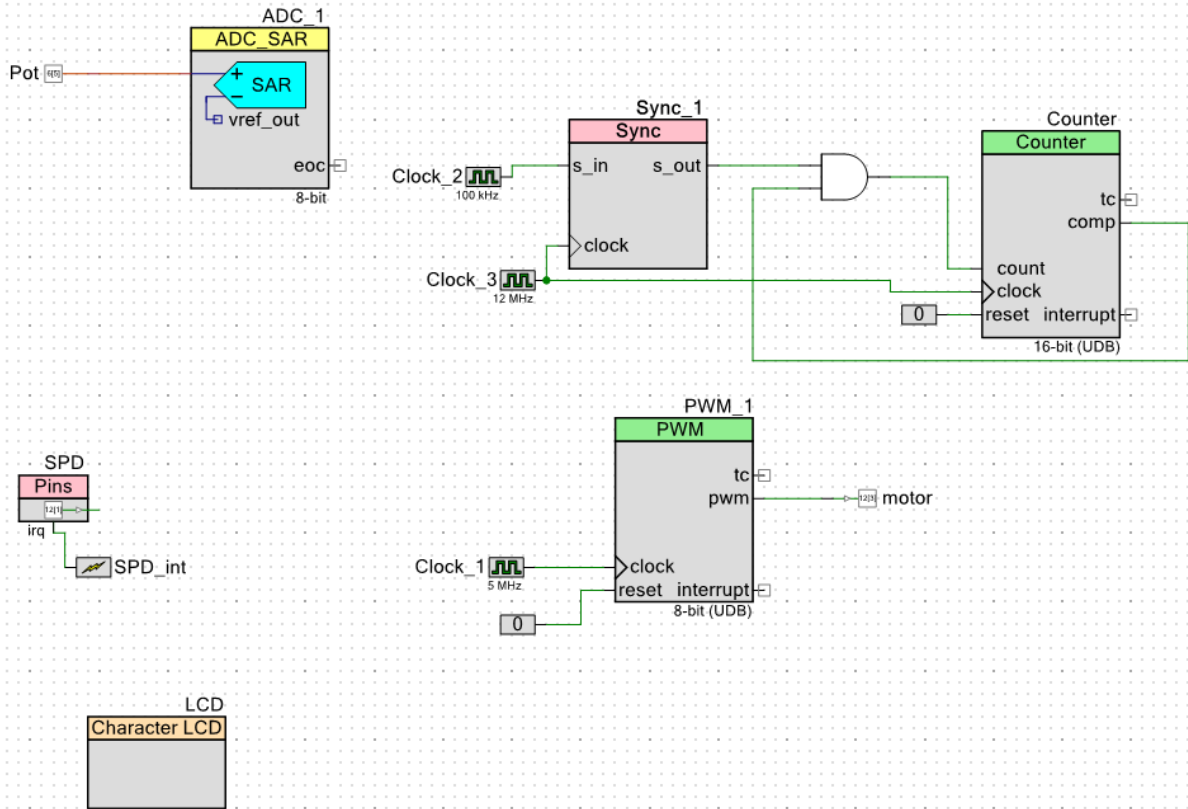
Gray 3-13 -> Bináris 3-13

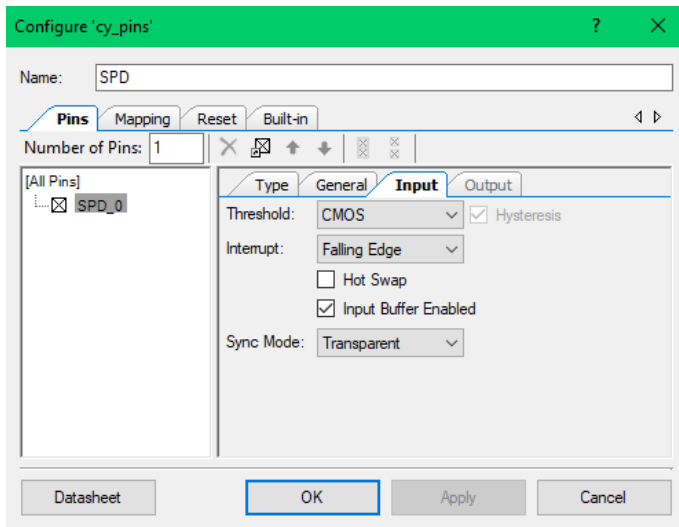
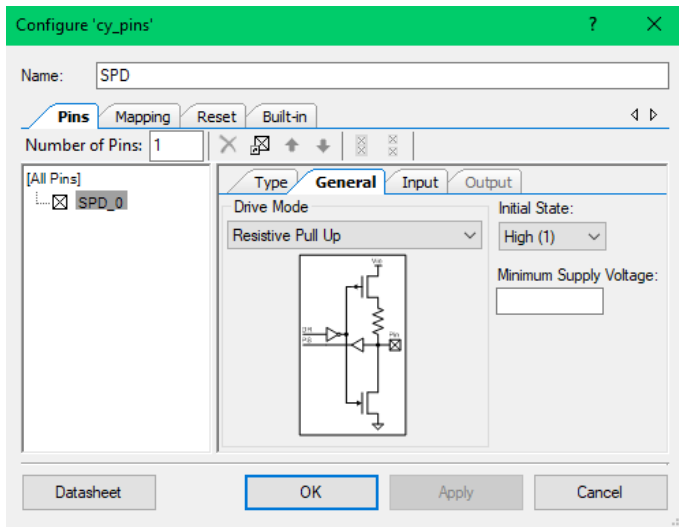
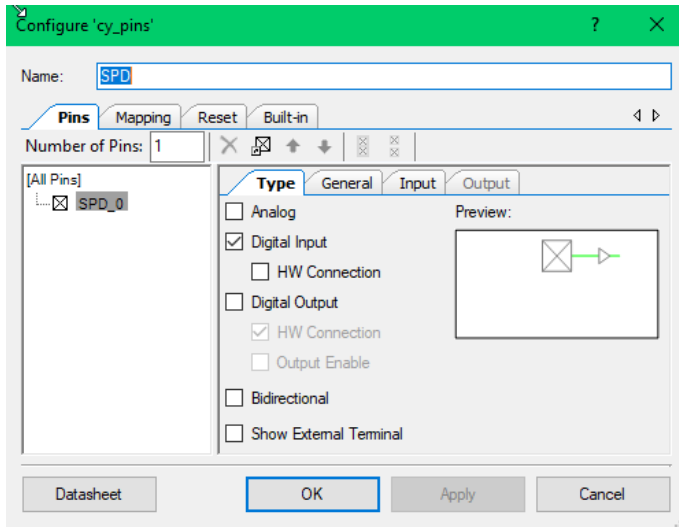
minterm: 3 4 5 6 7 8 9 10 11 12 13

Gray kód: 2, 6, 7, 5, 4, 12 13 15 14 10 11

Bináris kód:3 4 5 6 7 8 9 10 11 12 13

A Fenti táblázat alapján kitöltjük a feladat igazságtáblázatát és a kimeneti függvényeket (B3, B2, B1, B0) ábrázoljuk KV táblán, majd egyszerűsítünk.





Configure 'Counter' ? X

Name: Counter

Configure Advanced Built-in

Resolution: 8-Bit 16-Bit 24-Bit 32-Bit

Implementation: Fixed Function UDB

Period: 65535 Max

Compare Mode: Less Than

Compare Value: 60000 Max

Clock Mode: Up Counter

Datasheet OK Apply Cancel

Configure 'Counter' ? X

Name: Counter

Configure Advanced Built-in

Capture Mode: None

Enable Mode: Software Only

Run Mode: Continuous

Reload Counter: On Capture On Compare
 On Reset On TC

Interrupt: On TC On Capture
 On Compare

Datasheet OK Apply Cancel

Alias	Name	Port	Pin	Lock
	\LCD:LCDPort[6:0]\	P2[6:0]	95..99,1..2	<input checked="" type="checkbox"/>
	motor	P12[3]	68	<input checked="" type="checkbox"/>
	Pot	P6[5]	7	<input checked="" type="checkbox"/>
	SPD	P12[1] I2C1:SDA	54	<input checked="" type="checkbox"/>

```

1  /*
2  PID Szabályozó
3
4  Irányítástechnika beadandó feladat
5  Készítette: Kiss Dávid
6
7  Párhuzamos elvű PID szabályozó megvalósítása
8
9  !Float típusú változónál tizedes pontot alkalmazunk!
10 */
11 #include <project.h>
12
13 // Változók deklarálása
14
15 int16 celertek = 0; // célfordulat értéke
16 int16 period=0; // periódusidő
17
18 unsigned char duty = 0; // kitöltési tényező
19 unsigned char poti = 0; // potméter értéke
20
21 float duty_f = 0; // PID által számított kitöltési tényező
22 float fordulat = 0; // fordulatszám értéke
23 float error = 0; // hiba értéke
24 float pre_error =0; // előző hiba értéke
25 float integral =0; // integráló tag értéke
26 float derivative =0; // differenciáló tag értéke
27
28 float kp= ; // arányos tényező
29 float ki= ; // integráló tényező
30 float kd= ; // differenciáló tényező
31
32 // Megszakítások
33
34 CY_ISR(SPD_Handler) // sebesség számoló megszakítás kezelő
35 { // periódusidő számítása
36     Counter_Stop(); // számláló megáll
37     SPD_ClearInterrupt(); // megszakítás nyugtázása
38     period = Counter_ReadCounter(); // periódusidő kiolvasása
39     Counter_WriteCounter(0); // számláló nullázása
40     Counter_Enable(); // számláló újraindítása
41 }
42
43 int main() // Fő programkód
44 {
45     // Első beállítások
46
47     ADC_1_Start(); // ADC indítása
48     PWM_1_Start(); // PWM indítása
49     LCD_Start(); // LCD indítása
50     Counter_Start(); // Számláló indítása
51     LCD_ClearDisplay(); // LCD törlése
52     CyGlobalIntEnable; // Globális megszakítás kezelők engedélyezése
53     SPD_int_StartEx(SPD_Handler); // sebesség számoló megszakítás kezelő indítása
54

```

```

55 for(;;) // Folyamatosan futó program
56 {
57 // Célfordulat meghatározása
58 ADC_1_StartConvert(); // ADC indítása
59 ADC_1_IsEndConversion(ADC_1_WAIT_FOR_RESULT); // konverziós folyamat végének megvárása
60 poti = ADC_1_GetResult8(); // ADC kiolvasása
61 celertek = (poti * 12); // célfordulat kiszámítása
62
63 // Álló motor detektálása
64 if (Counter_ReadCounter() < 59999) // Ha a számláló(periódusidő) értéke nem túl nagy
65 { fordulat = (400000 / period); } // a fordulatszámot kiszámoljuk
66 else { fordulat = 0; } // Ha az érték túl nagy a fordulatszám 0, a motor áll
67
68 // LCD működtetése
69 LCD_Position(0u, 0u);
70 LCD_PrintString("Celertek:"); // Cél fordulat kiírása
71 LCD_Position(0u, 9u);
72 LCD_PrintNumber(celertek);
73 LCD_PrintString(" "); // Üres helyek feltöltése Space karakterekkel
74 LCD_Position(1u, 0u);
75 LCD_PrintString("Fordulat:"); // Aktuális fordulat kiírása
76 LCD_Position(1u, 9u);
77 LCD_PrintNumber(fordulat);
78 LCD_PrintString(" "); // Üres helyek feltöltése Space karakterekkel
79
80 // PID szabályozó értékek számítása
81 error = celertek - fordulat; // Hiba számítása
82 integral = integral + error; // integráló tag számítása
83 derivative = error - pre_error; // deriváló tag számítása
84 duty_f = (kp * error) + (ki * integral) + (kd * derivative); // PWM kitöltési értékének számítása, párhuzamos PID elve alapján
85 pre_error = error; // hiba értékének eltárolása
86
87 // kimenet limitálása, nem anti-windup
88 if (duty_f > 254.99) // Ha a kitöltés nagyobb lenne, mint 254,99 (túlnövi a maximum értéket)
89 { duty_f = 255.0; } // 255-re állítom
90 else
91 { if (duty_f < 0.1) // Ha a kitöltés kisebb lenne mint 0,1 (negatívba csökkenne)
92 { duty_f = 0.0; } // 0-ra állítom
93
94 // PWM blokk működtetése
95 duty = (int8)duty_f; // kitöltési tényező konvertálása float-ból 8 bites egészé
96 PWM_1_Stop(); // PWM blokk megáll
97 PWM_1_WriteCompare(duty); // kitöltési tényező frissül
98 PWM_1_Start(); // PWM újraindul
99
100 CyDelay(1); // 1 ms késleltetés a futások között
101 }
102 }

```

PID paraméterek:

Ziegler–Nichols „lengetéses” módszerrel megállapítva

$K_{CR}=0,35$

$P_{CR}=0,03$ s

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$