

**Miskolci Egyetem
Gépészmérnöki és Informatikai Kar
Automatizálási és Infokommunikációs
Intézeti Tanszék**

**2019/2020. tanév
I-II. félév**

**Beágyazott rendszerek és Architektúrák c. tantárgy
előadásának és gyakorlatának ütemterve
BSC Mechatronikai hallgatók részére
Tervezés-gyártás szakirány
GEVAU 520-B**

Tárgynév:	Beágyazott rendszerek és Architektúrák			
Rövid név:	Beágy. rendsz.	Kód	GEVAU520B	
Angol név:	Embedded Systems			
Tanszék:	Automatizálási Tanszék			
Tárgyfelelős:	Dr. Vásárhelyi József vajo@uni-miskolc.hu			
Előtanulmányok:	Digitális rendszerek		Kódja:	GEVAU 195-B
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:	<p>Beágyazott rendszerek áttekintése, Mikroprocesszorok, mikrovezérlők felépítése; 8051 alapú fejlett mikrovezérlő család felépítése, programozási környezete; Microblaze mikroprocesszor integrálása FPGA architektúrába; Rendszer a chipen fejlesztési ismeretek; Perifériák csatlakoztatása, illesztése mikroprocesszoros rendszerekben</p> <p><i>Gyakorlat:</i> A gyakorlatok keretében a következő témakörökkel foglalkozunk: Mikrovezérlőkkel megvalósított léptetőmotor-, kefe nélküli motor hajtás megvalósítása, AD/DA átalakítás mikrovezérlős rendszerekben, vezeték nélküli adatátvitel mikrovezérlőkkel (ZigBee), web szerver megvalósítása mikrovezérlővel; Rendszer chipen fejlesztés alapfogalmai.</p>			
Irodalom:	<ol style="list-style-type: none"> Vahid F., Givargis T.: Embedded System Design, a Unified Hardware/Software Introduction, Wiley and Sons, ISBN 0-471-38678-2, 2002, pp. 324. (k) Li Q., Yao C.: Real-Time Concepts for Embedded Systems, CMP Books, ISBN: 1-57820-124-11993 (a) elearning anyag az ekönyvtárban Peter Wilson, Design Recipes for FPGAs using Verilog and VHDL, Newnes, ISBN 978-0-08-097129-2, 2007, pp. 370 C. "Max" Maxfield: The Design Warrior's Guide to FPGAs, Elsevier, ISBN: 0-7506-7604-3, 2004, pp. 560 Li Q., Yao C.: Real-Time Concepts for Embedded Systems, CMP Books, ISBN: 1-57820-124-11993 			
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.			

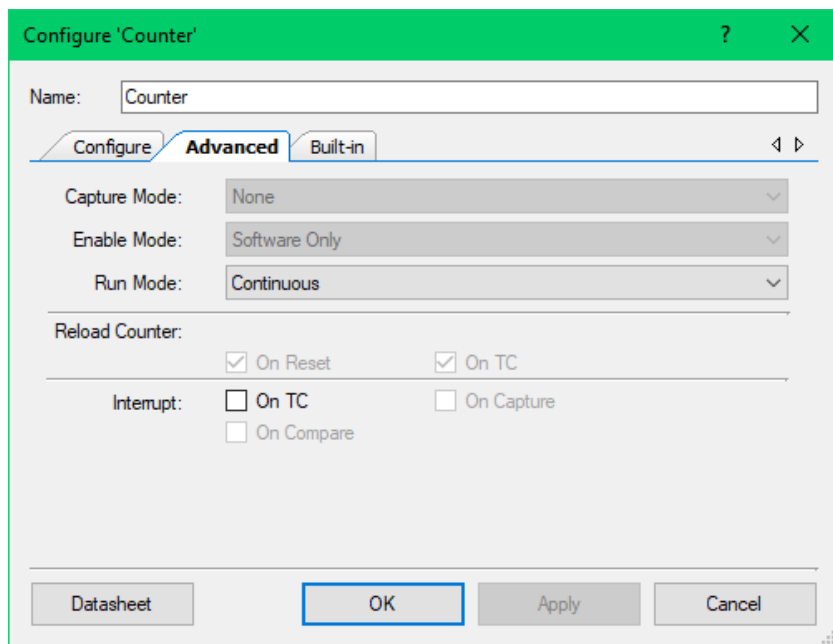
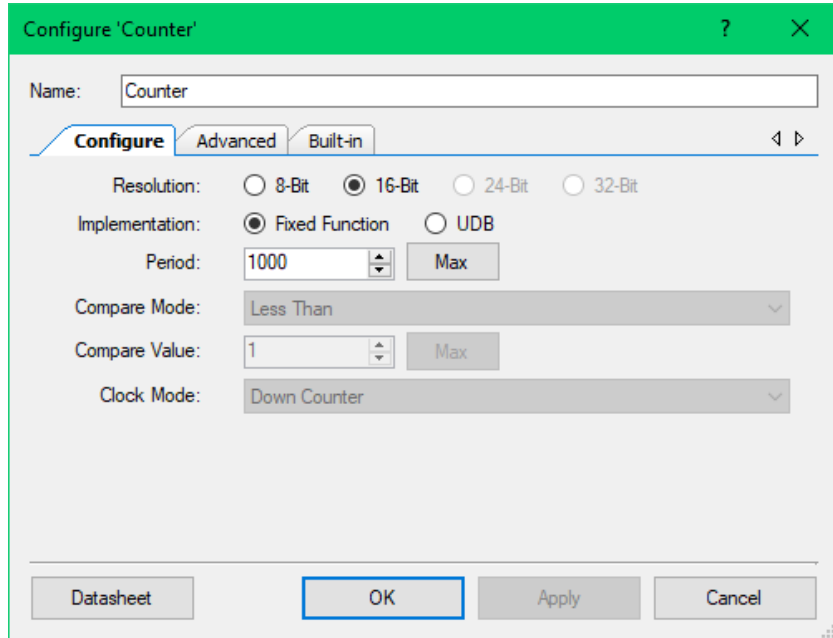
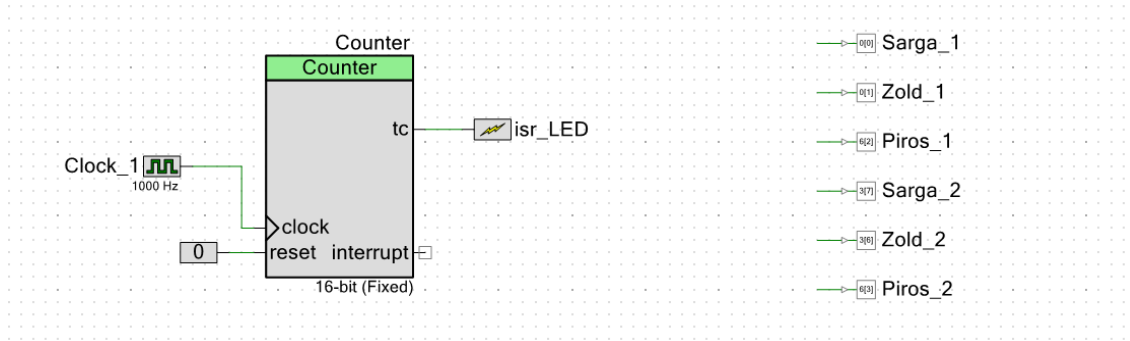
<i>Lezárási feltételek:</i>	Egyéni feladatok elvégzése legalább elégséges; 24-28 elégséges, 28-32 közepes 32-36 jó, 36-40 jeles ZH elért eredmény legalább elégséges > 60%; 24- 28 elégséges, 28-32 közepes, 32-36 jó, 36-40 jeles Gyakorlati jegy: a ZH*0,5+ gyakorlatok átlaga*0,5
<i>Ütemterv</i>	
1.	Ea Beágyazott rendszerek áttekintése, Beágyazott rendszer elemzése tervezési kihívások, követelmények, trendek, Moore törvénye. Lab:
2.	Ea: Hardver elemek, processzor technológiák, processzorok, mikrovezérlők, 8051-es architektúra; Lab:
3.	Ea: Cypress PSOC fejlett mikrovezérlő perifériák I. Lab:
4.	Ea: Cypress PSOC fejlett mikrovezérlő perifériák II. Lab:
5.	Ea.: Cypress PSOC fejlett mikrovezérlő periféria kezelés. Lab:
6.	Ea: Szoftvertervezés, hardvertervezés, hardver-szoftver együttes tervezése és szimulációja. Lab: Egyéni feladat hardver és szoftver fejlesztés.
7.	Ea: Memória szerepe a beágyazott rendszerekben. Interfész technika. Beágyazott rendszerekben használt szabványos interfészek ismertetése, kezelése. Lab: Egyéni feladat készítése, jegyzőkönyvvel. Feladatbeadás a 9. héten.
8.	Ea: Szabványos kommunikációs protokollok. Beágyazott rendszerekben használt szabványos kommunikációs protokollok ismertetése, kezelése. Lab: Egyéni feladat hardver és szoftver fejlesztés.
9.	Ea: Motorvezérlés, mint beágyazott rendszer feladat. Léptetőmotorok és váltóáramú motorok vezérlése szabályozása. Lab: Egyéni feladat hardver és szoftver fejlesztés.
10.	Ea.: Tervezési példa: digitális kamera tervezése. Állapotgépek és konkurens folyamatok kezelése Lab: Egyéni feladat hardver és szoftver fejlesztés.
11.	Ea: Modellek és programozási nyelvek, programozási nyelvek és grafikus tervbeviteli módszerek összehasonlítása. Lab: Egyéni feladat hardver és szoftver fejlesztés.
12.	Ea. Processzek/folyamatok kommunikációja, szinkronizálása, megvalósítása/implementációja. Valós idejű operációs rendszerek. Digitális szabályozási rendszerek tervezése. Lab: Egyéni feladat hardver és szoftver fejlesztés.
13.	Ea: IC technológiák szerepe a beágyazott rendszerekben. Lab: Pótlás
14.	Ea: konzultáció, zárthelyi Lab: Pótlás

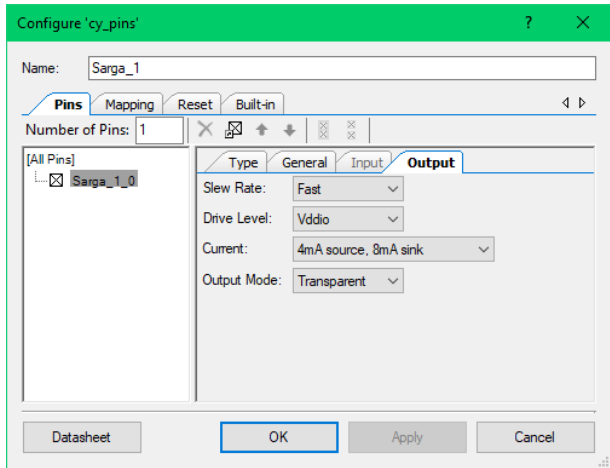
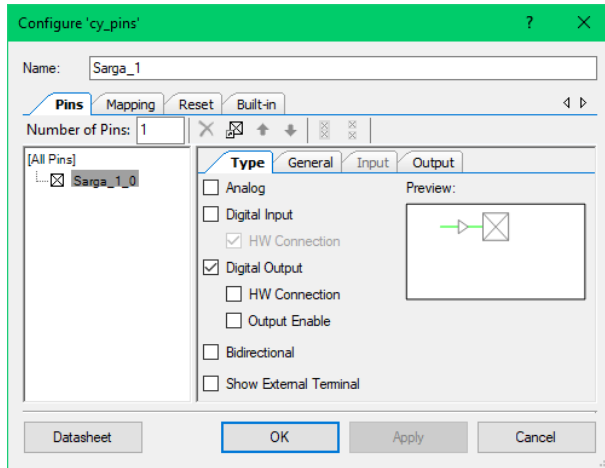
Intézetigazgató

Dr. Trohák Attila
egyetemi docens

Tárgyfelelős:

Dr. Vásárhelyi József
egyetemi docens





```
#include <project.h>
```

```
int a = 0;

CY_ISR(Counter_Handler)
{
    Counter_Stop();
    if (a==3)
        { a=0;}
    else
        {a=a+1;}
    switch (a)
    {
        case 0 :
            Counter_WriteCounter(10000);
            Counter_Enable();
            break;
        case 1 :
            Counter_WriteCounter(2000);
            Counter_Enable();
            break;
        case 2 :
            Counter_WriteCounter(10000);
            Counter_Enable();
            break;
        case 3 :
            Counter_WriteCounter(2000);
            Counter_Enable();
            break;
        default :
            Counter_Enable();
            break;
    }
}
```

```
int main()
{
    CyGlobalIntEnable;
    Counter_Start();
    isr_LED_StartEx(Counter_Handler);

    for(;;)
    {
        switch (a)
        {
            case 0:
                Zold_1_Write(1);
                Sarga_1_Write(0);
                Piros_1_Write(0);
                Zold_2_Write(0);
                Sarga_2_Write(0);
                Piros_2_Write(1);
                break;

            case 1:
                Zold_1_Write(0);
                Sarga_1_Write(1);
                Piros_1_Write(0);
                Zold_2_Write(0);
                Sarga_2_Write(1);
                Piros_2_Write(1);
                break;
        }
    }
}
```

```

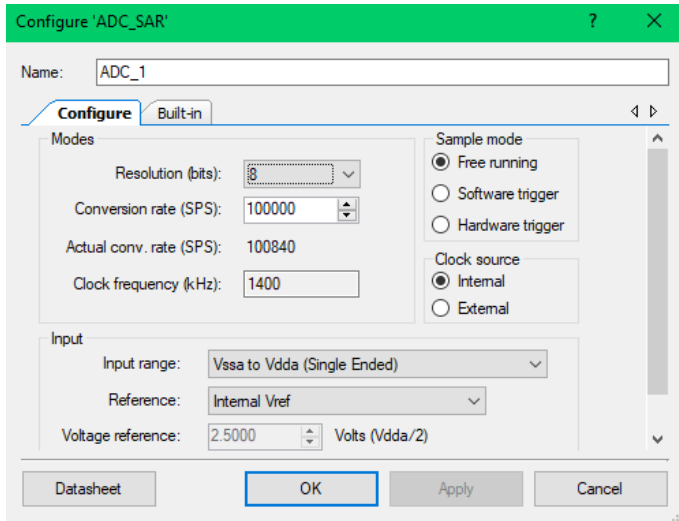
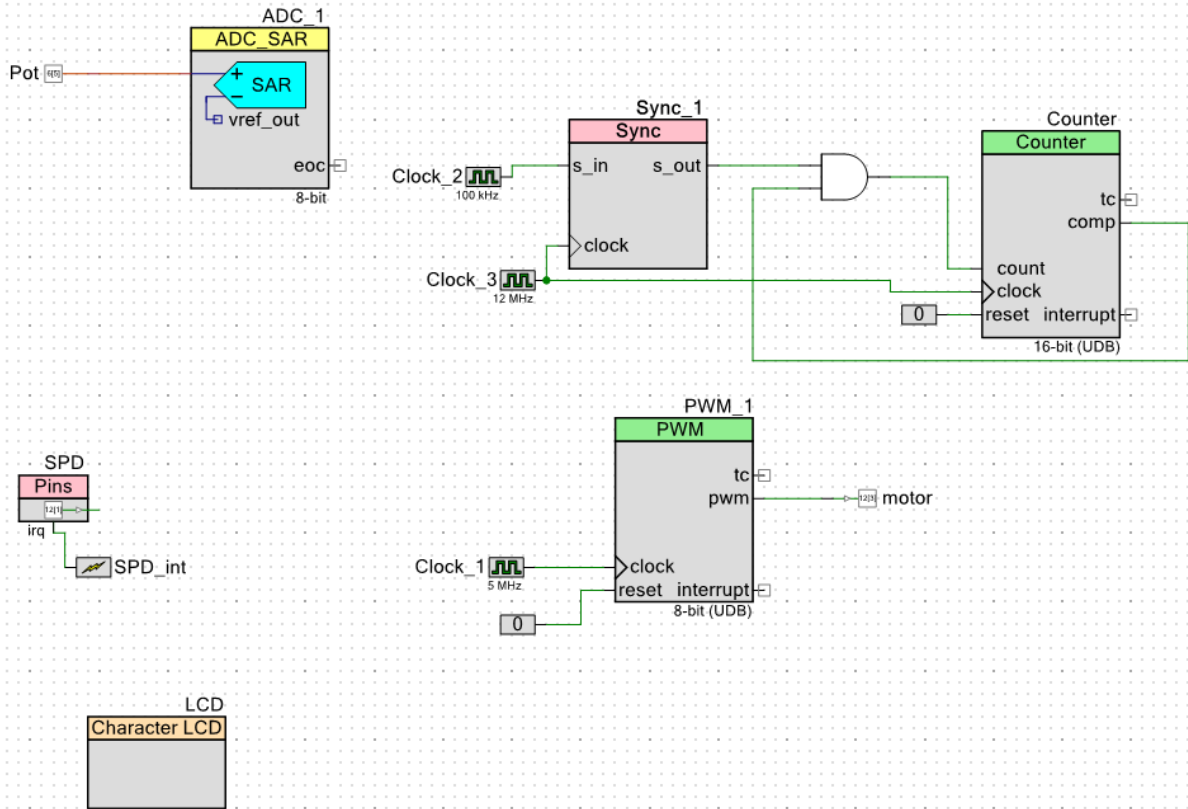
        case 2:
            Zold_1_Write(0);
            Sarga_1_Write(0);
            Piros_1_Write(1);
            Zold_2_Write(1);
            Sarga_2_Write(0);
            Piros_2_Write(0);
        break;

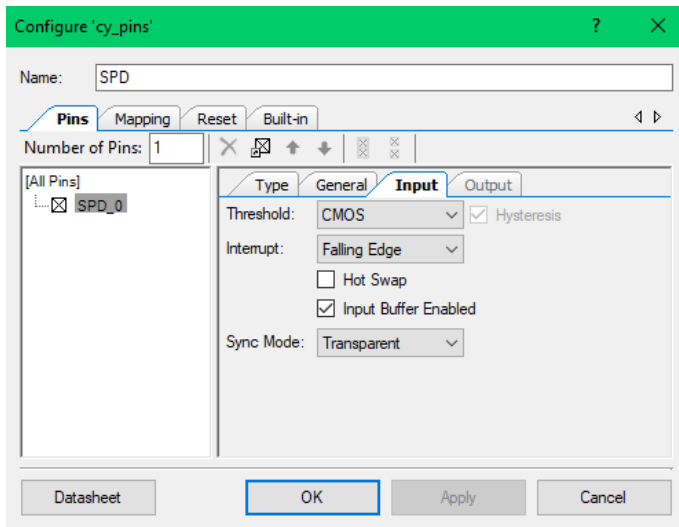
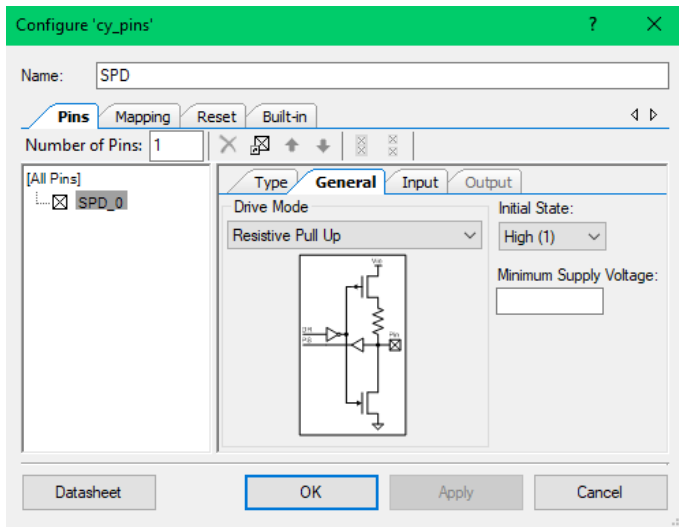
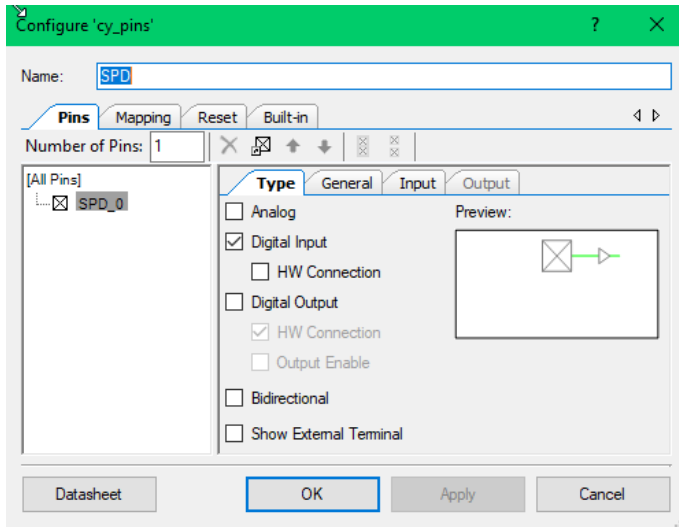
        case 3:
            Zold_1_Write(0);
            Sarga_1_Write(1);
            Piros_1_Write(1);
            Zold_2_Write(0);
            Sarga_2_Write(1);
            Piros_2_Write(0);
        break;

        default:
            Zold_1_Write(0);
            Sarga_1_Write(0);
            Piros_1_Write(0);
            Zold_2_Write(0);
            Sarga_2_Write(0);
            Piros_2_Write(0);
        break;
    }
}
}

```

Alias	Name /	Port	Pin	Lock
	Piros_1	P12[1] I2C1:SDA	54	<input checked="" type="checkbox"/>
	Piros_2	P12[3]	68	<input checked="" type="checkbox"/>
	Sarga_1	P4[3]	81	<input checked="" type="checkbox"/>
	Sarga_2	P4[5]	83	<input checked="" type="checkbox"/>
	Zold_1	P0[7] IDAC:HI	79	<input checked="" type="checkbox"/>
	Zold_2	P0[5] OpAmp-	77	<input checked="" type="checkbox"/>





Configure 'Counter' ? X

Name:

Configure Advanced Built-in

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

Implementation: Fixed Function UDB

Period: Max

Compare Mode:

Compare Value: Max

Clock Mode:

Datasheet OK Apply Cancel

Configure 'Counter' ? X

Name:

Configure Advanced Built-in

Capture Mode:

Enable Mode:

Run Mode:

Reload Counter: On Capture On Compare
 On Reset On TC

Interrupt: On TC On Compare
 On Capture

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	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$