Miskolci Egyetem
Gépészmérnöki és Informatikai Kar
Automatizálási és Kommunikáció-Technológiai Tanszék

2019/2020. tanév
I. félév

Digitális rendszerek komplex tervezése GEVAU 517B c. tantárgy

Előadásának és gyakorlatainak ütemterve

BSc szintű villamosmérnök hallgatók részére.

Tárgynév:	**Digitális rendszerek komplex tervezése**					
Rövid név:			*Kód*	**GEVAU517B**		
Angol név:	Complex Design of Digital Systems					
Tanszék:	Automatizálási és Infokommunikációs Intézet					
Tárgyfelelős:	Bartók Roland, tel: (46) 565 111 /1753 qgeroli5@uni-miskolc.hu					
Előtanulmányok:	Digitális rendszerek III.		*Kódja:*	GEVAU505B		
Kredit:		*Követelmény:*	kollokvium			
Heti óraszámok	*Előadás:*	2	*Gyakorlat:*		*Labor:*	2
Oktatási cél:	A villamosmérnöki ismeretekhez a digitális-technikai alapok elsajátítása.					
Tárgy tartalom:	Digitális rendszerek tervezési módszerei; Tervezési technológiák. Digitális rendszerek általános tervezési módszerei. Tesztelésre és gyártásra tervezés. A tervezésben és gyártásban használt tesztelési eljárások általános ismertetése, különös hangsúllyal a peremfigyeléses tesztelésre. IOT Rendszerek; Vezeték nélküli kommunikáció alapjai; Integrált áramkörök közötti kommunikáció. Mikrovezérlős rendszerek tervezése és tervezési szempontok. Digitális szabályozási rendszerek tervezése, Algoritmusok és architektúrák tervezése szintézise; posztszintézis - terv ellenőrzés. Internetes eszközök tervezése.					
Irodalom:	The Architecture of Computer Hardware, Systems Software & Networking: An Information Technology Approach 4th Edition, Irv Englander John Wiley and Sons C 2010					
Ajánlott Irodalom	1. Labrossse J.J et all: *Embedded Software know it all*, Newnes, ISBN 978-07506-8582-5, 2008, pp.770. 2. Labrosse J.J: *MicroC/OS-II The real-time kernel*, CMP Books, ISBN 1-57820-103-9, 2002, pp. 606. 3. Scott Hauck, Andree Dehon ed. *Reconfigurable Computing The Theory and Practice of FPGA-Based Computation*, Elsvier, ISBN 978-0-12-370522-8, 2008, pp. 945 4. https://www.jtaglive.com/ 5. www.python.org					

Mintatantervi elhelyezkedés szakok szerint				
Szak	*Szakirány/sáv*	*Tantervi modul-tantervi kód*	*Mintatantervi félév*	*Választhatóság*
Villamosmérnöki Szak	BV_E	BV	5	kötelező

Jellemző oktatási módok	
Oktatási nyelv:	Magyar, angol
Előadás:	Tábla + számítógépes vetítés
Gyakorlat:	
Labor:	laboratórium gyakorlat egyéni feladatokkal
Évközi feladatok, zárthelyik:	legalább 1 évközi összetett feladat
Lezárási feltételek:	A Tanulmányi és Vizsgaszabályzat szerint. Az Előadások legalább 60%-ának látogatása, a gyakorlatok legalább 75%-ának teljesítése. Gyakorlatokon aktív részvétel; az előírt feladatok teljesítése; az évközi (házi) feladatok elfogadható szintű elkészítése. A lezáráshoz írásbeli- és szóbeli vizsgát kell tenni. Az évközi teljesítmény 50%-a és a vizsga 50%-ban a tárgyat lezáró jegy. Az évközi feladatok és vizsga legalább elégséges teljesítése (50%)

Ütemterv	
1.	Ea: Digitális rendszerek általános tervezési módszerei. Tervezési szempontok, Tesztelésre és gyártásra tervezés. Gyak: Python programozás alapjai
2.	Ea: Elektronikai tervezés és gyártás során alkalmazott tesztelési módszerek általános ismertetése. Peremfigyelés (Boundary Scan - JTAG) módszer alkalmazása a tervezésben és gyártásban; JTAG alkalmazási területei. Gyak: JTAG gyakorlat + Python
3.	EA: IEEE 1149.1 és IEEE 1532 JTAG szabványok koncepciója, struktúrája. Áramkör szintű és rendszer szintű tesztelés a peremfigyelés módszerével. Feltárható hibák típusai, módszerek összehasonlítása a tervezés és gyártás folyamatában. DFT – Design for Test alapelvei. Gyak: JTAG gyakorlat, tesztelés, scriptek
4.	EA: Integrált áramkörök közötti kommunikációs módok – soros, párhuzamos kommunikáció UART, USART, SPI, I2C, CAN, LIN, USB Gyak: PSoC gyakorlat – I/O kezelés, analóg bemenet
5.	EA: IoT rendszerek; IoT rendszerek vezeték nélküli kommunikációs technológiái Gyak: PSoC gyakorlat – Megszakítások, időzítők
6.	EA: Bluetooth, ZigBee, WiFi kommunikáció alapok Gyak: PSoC gyakorlat – Kommunikációk vizsgálata
7.	EA: Szoftvertervezés beágyazott rendszereken Gyak: PSoC gyakorlat – Egyszerű szoftver váz készítése
8.	EA: Hibakeresés és tesztelés Gyak: PSoC gyakorlat – Hibakeresés és tesztelés
9.	EA: Egyszerű digitális szabályozó tervezése Gyak: PSoC gyakorlat – Fordulatszám szabályozó készítése
10.	EA: Felhasználói felület tervezése Gyak: PSoC gyakorlat – Felhasználói felület tervezése
11.	EA: Szoftvertervezés beágyazott rendszereken - RTOS Gyak: PSoC gyakorlat – RTOS
12.	EA: Egyéni feladatok megoldása Gyak: Egyéni feladatok megoldása
13.	EA: Egyéni feladatok megoldása Gyak: Feladat beadás, Gyakorlat pótlás
14.	EA: Egyéni feladatok megoldása Gyak: Feladat beadás, Gyakorlat pótlás

Miskolc, 2019. szeptember. 1.

Dr. Trohák Attila
tanszékvezető egyetemi docens

Dr. Bartók Roland
egyetemi tanársegéd

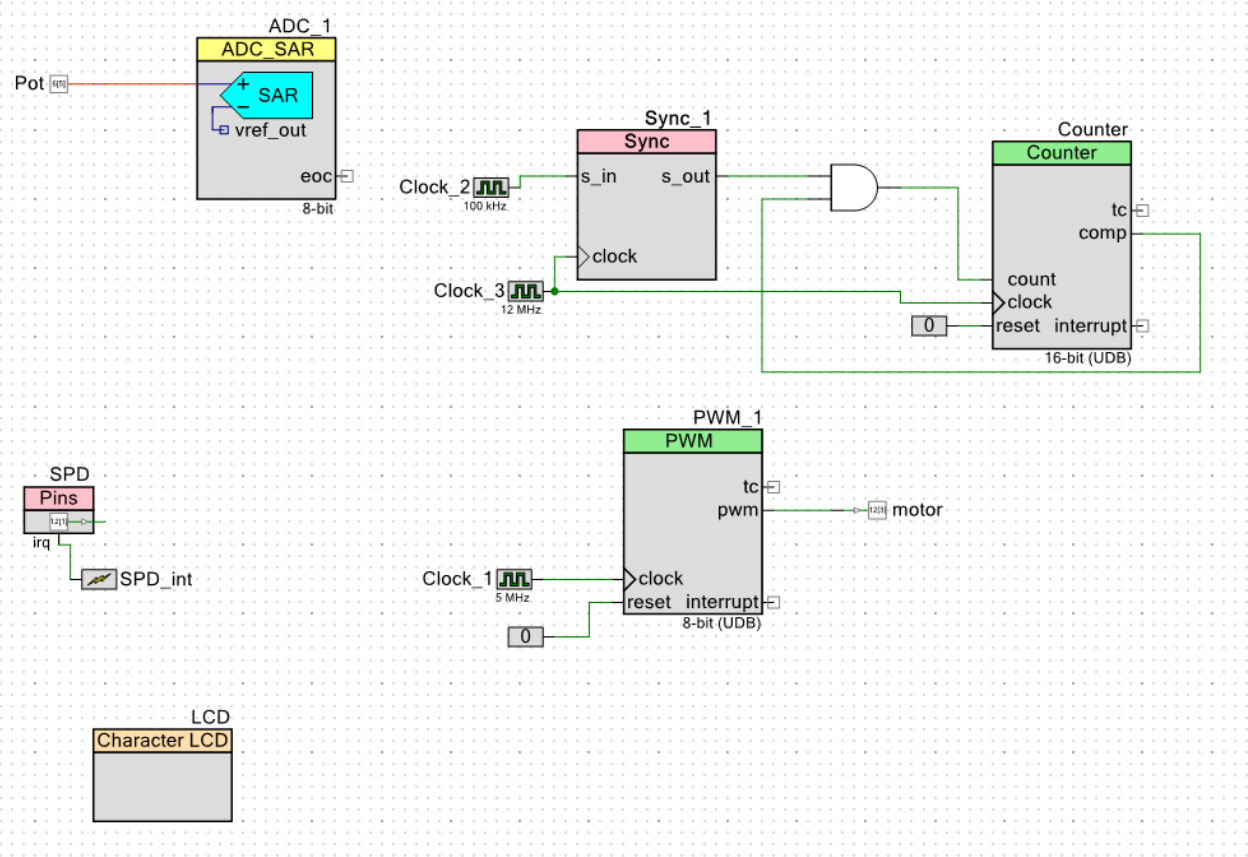
Vizsga minta kérdések

1. Milyen feladat megoldására jött létre a JTAG szabvány?
   - a nagy alkatrészláb sűrűségű integrált áramkör közötti villamos összeköttetések tesztelhetővé váljanak
   - az IC-k belső működését vizsgálják, és programozható eszközöket programozhatnak fel vele
   - a belső Si-lapka a tokozás és µhuzal közötti szakadást kimutassák
2. Mit tartalmaz a BSDL állomány?
   - VHDL-re épült formátum leírja, hogyan van megvalósítva a B.S. az adott eszközön
   - BSD kiterjesztésű fájl, a gyártó biztosítja a BSDL fájlt
   - azt írja le, hogy milyen utasítások vannak definiálva, és hogy azoknak mi az OP kódja
   - ez adja meg, hogy milyen peremfigyelő cellák vannak, és azok melyik lábakhoz vannak hozzárendelve
3. Milyen frekvenciákat használnak az ISM sávok?
   - 868 MHz (915MHz)
   - 2,4 GHz
   - 5,8 GHz
4. Mi a célja a Bluetooth technológiának?
   - Kiselektronikai eszközök vezeték nélküli kommunikációja. A vezeték kiváltása.
5. Mit tesz lehetővé a LORA?
   - A LoRa és LoRaWAN lehetővé tesz olcsó, nagy hatótávolságú összeköttetést IoT-s eszközök számára vidéki, távoli, esetleg offshore ipari területek számára.
6. Ismertesse az LPWAN alkalmazási területeit!
   - Fix helyű, közepes- vagy nagy sűrűségű kapcsolatok esetén: Városokban, vagy épületekben, tökéletes alternatíva az M2M műholdas csatlakozáshoz képest. Néhány példa: okos-világítás, elosztás automatizálás (smart grid), város-központú GPS tulajdon nyomon követése.
   - Hosszú életű, akkumulátor táplálta alkalmazások esetén: Amikor hosszú távolságokról van szó, a hagyományos technológiákkal szemben remek alternatíva lehet, például nagyobb területű vízmérés, gáz detektálás, okos mezőgazdaság, akkumulátor-táplálta zárak és access control pontok.
7. Mennyi vezeték szükséges a TTL-UART használata esetén?
   - A TTL-UART-nál csak 3 vezeték kell a kétoldalú soros kommunikációhoz. Egy Adás (Transmit-TX) egy Vétel (Receive-RX) és egy közös föld (GND) vezeték. A TX vonalon küldjük el az adatot a másik eszköznek, és az RX vonalon a másik eszköz küld adatot nekünk.
8. Hogyan szinkronizálj az adatátvitelt az SPI busz?
   - Az SPI buszon az adatátvitel szinkronizálását a busz vezérlő eszköze (SPI master) által keltett szinkron órajel (SCK) biztosítja
9. Milyen típusú kommunikációt végez a CAN?
   - Üzenetszórásos, azaz minden, a hálózathoz csatlakozó eszköz lát minden üzenetet.
10. Milyen közeg hozzáférési módszert használ a WiFi?
    - CSMA/CA

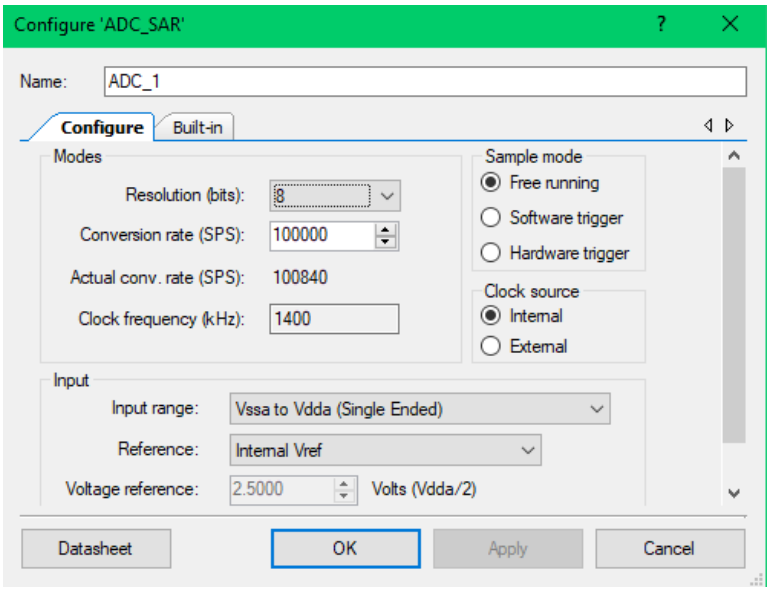
Gyakorlati feladatok minta

PSoC Creator aktuális verziójával felépített egyszerű szabályozó rendszer terve.

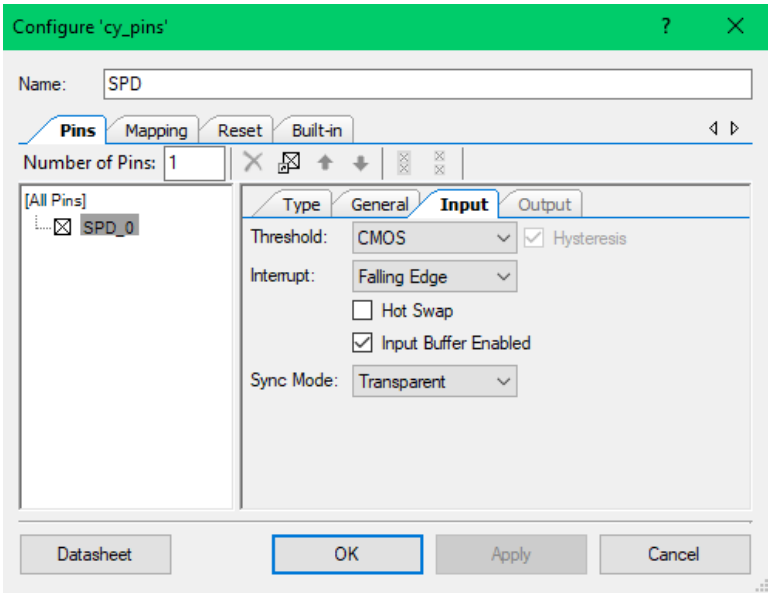
A teljes kapcsolási rajz:

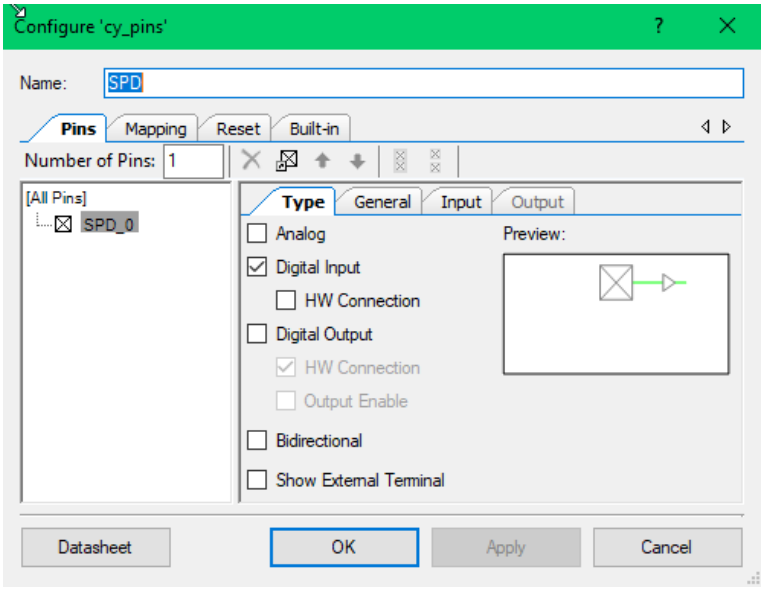


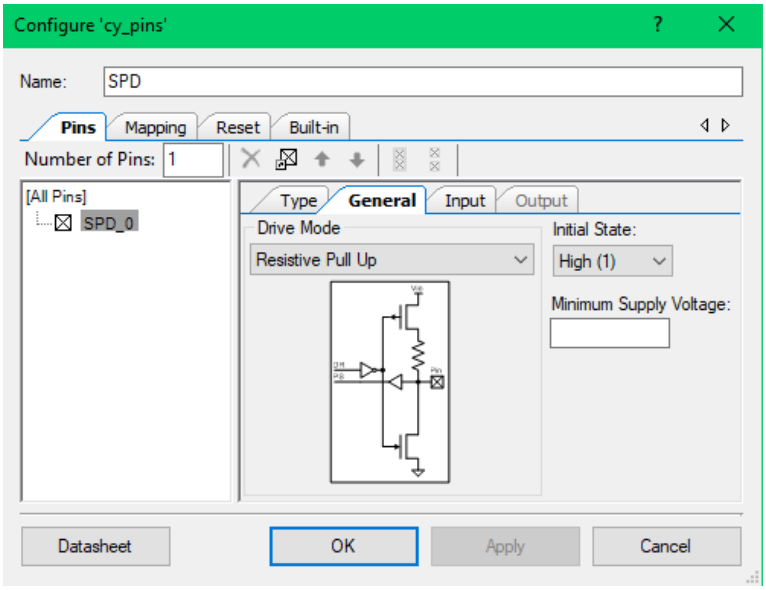
ADC konfigurációja:



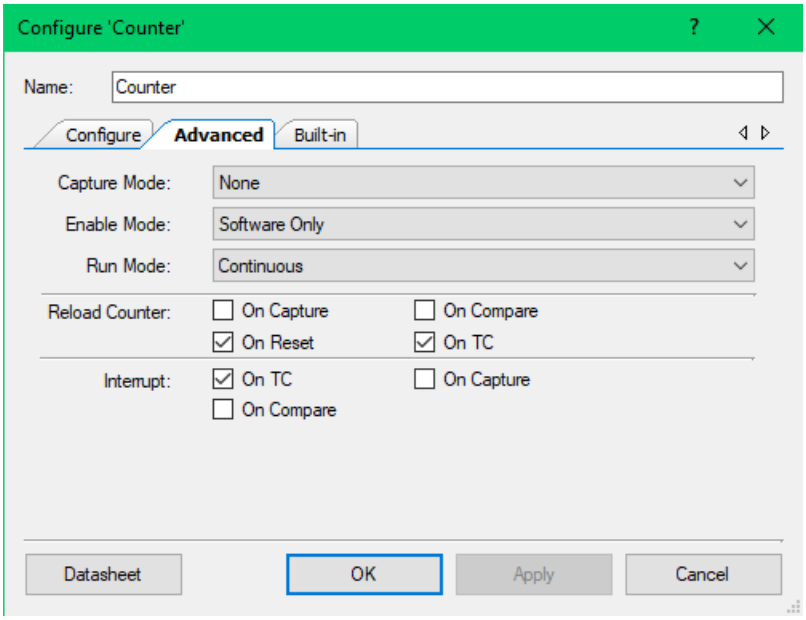
Bemeneti láb konfigurációja:

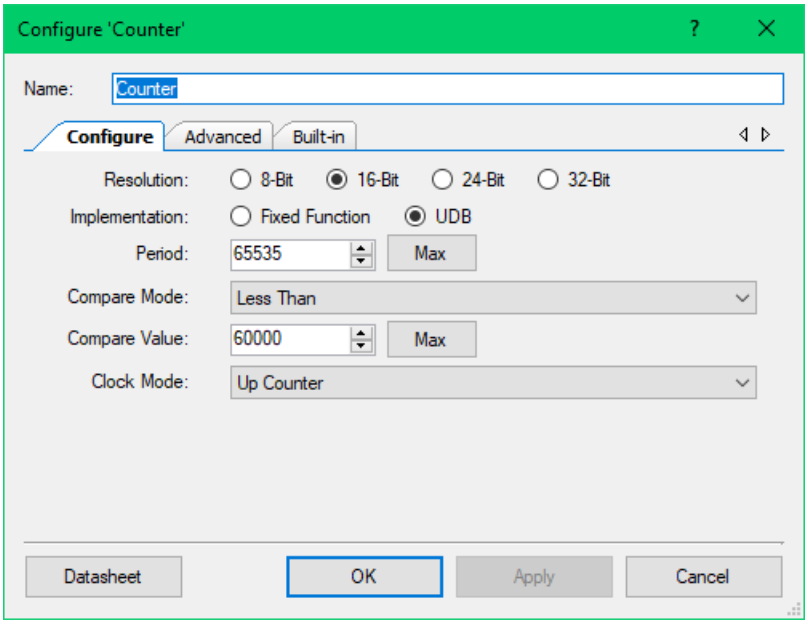






Számláló konfigurációja:





I/O portok konfigurációja:

Alias	Name	Port	Pin	Lock
	\LCD:LCDPort[6:0]\	P2[6:0]	95..99,1..2	☑
	motor	P12[3]	68	☑
	Pot	P6[5]	7	☑
	SPD	P12[1] I2C1:SDA	54	☑

Programkód:

```
/*
    PID Szabályozó

    Irányítástechnika beadandó feladat
    Készitette: Kiss Dávid

    Párhuzamos elvű PID szabályozó megvalósitása

    !Float típusú változónál tizedes pontot alkalmazunk!
*/
#include <project.h>

// Változók deklarálása

int16 celertek = 0;     // célfordulat értéke
int16 period=0;         // periódusidő

unsigned char duty = 0; // kitöltési tényező
unsigned char poti = 0; // potméter értéke

float duty_f = 0;   // PID által számított kitöltési tényező
float fordulat = 0; // fordulatszám értéke
float error = 0;    // hiba értéke
float pre_error =0; // előző hiba értéke
float integral =0;  // integráló tag értéke
float derivative =0;// differenciáló tag értéke

float kp=0.2 ;      // arányos tényező
float ki=0.01;      // integráló tényező
float kd=0.01;      // differenciáló tényező

// Megszakitások

    CY_ISR(SPD_Handler)                     // sebesség számoló megszakítás kezelő
    {                                       // periódusidő számítása
        Counter_Stop();                     // számláló megáll
        SPD_ClearInterrupt();               // megszakítás nyugtázása
        period = Counter_ReadCounter();     // periódusidő kiolvasása
        Counter_WriteCounter(0);            // számláló nullázása
        Counter_Enable();                   // számláló újraindítása
    }


int main()      // Fő programkód
{
// Első beállitások

    ADC_1_Start();                          // ADC indítása
    PWM_1_Start();                          // PWM indítása
    LCD_Start();                            // LCD indítása
    Counter_Start();                        // Számláló indítása
    LCD_ClearDisplay();                     // LCD törlése
    CyGlobalIntEnable;                      // Globális megszakítás kezelők engedélyezése
    SPD_int_StartEx(SPD_Handler);           // sebesség számoló megszakítás kezelő indítása
```

```
    for(;;)     // Folyamatosan futó program
    {
    // Célfordulat meghatározása
        ADC_1_StartConvert();                           // ADC indítása
        ADC_1_IsEndConversion(ADC_1_WAIT_FOR_RESULT);   // konverziós folyamat végének megvárása
        poti = ADC_1_GetResult8();                      // ADC kiolvasása
        celertek = (poti * 12);                         // célfordulat kiszámítása

    // Álló motor detektálása
        if (Counter_ReadCounter()<59999)                // Ha a számláló(periódusidő) értéke nem túl nagy
        { fordulat = (400000 /  period); }              // a fordulatszámot kiszámoljuk
        else { fordulat = 0; }                          // Ha az érték túl nagy a fordulatszám 0, a motor áll

    // LCD működtetése
        LCD_Position(0u, 0u);
        LCD_PrintString("Celertek:");                   // Cél fordulat kiírása
        LCD_Position(0u, 9u);
        LCD_PrintNumber(celertek);
        LCD_PrintString("     ");                       // Üres helyek feltöltése Space karakterekkel
        LCD_Position(1u, 0u);
        LCD_PrintString("Fordulat:");                   // Aktuális fordulat kiírása
        LCD_Position(1u, 9u);
        LCD_PrintNumber(fordulat);
        LCD_PrintString("     ");                       // Üres helyek feltöltése Space karakterekkel

    //  PID szabályozó értékek számítása
        error = celertek - fordulat;                    // Hiba számítása
        integral = integral + error;                    // integráló tag számítása
        derivative = error - pre_error;                 // deriváló tag számítása
        duty_f = (kp * error) + (ki * integral) + (kd * derivative);  // PWM kitöltési értékének számítása, párhuzamos PID elve alapján
        pre_error = error;          // hiba értékének eltárolása

    // kimenet limitálása, nem anti-windup
       if (duty_f > 254.99)         // Ha a kitöltés nagyobb lenne, mint 254,99 (túlnövi a maximum értéket)
            {  duty_f = 255.0;}     // 255-re állítom
       else
        { if (duty_f < 0.1)         // Ha a kitöltés kisebb lenne mint 0,1 (negatívba csökkenne)
            { duty_f= 0.0;}  }      // 0-ra állítom

    // PWM blokk működtetése
        duty = (int8)duty_f;        // kitöltési tényező konvertálása float-ból 8 bites egésszé
        PWM_1_Stop();               // PWM blokk megáll
        PWM_1_WriteCompare(duty);   // kitöltési tényező frissül
        PWM_1_Start();              // PWM újraindul

        CyDelay(1);                 // 1 ms késleltetés a futások között
    }
}
```

PID paraméterek:

Ziegler–Nichols „lengetéses” módszerrel megállapítva

KCR=0,35

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