

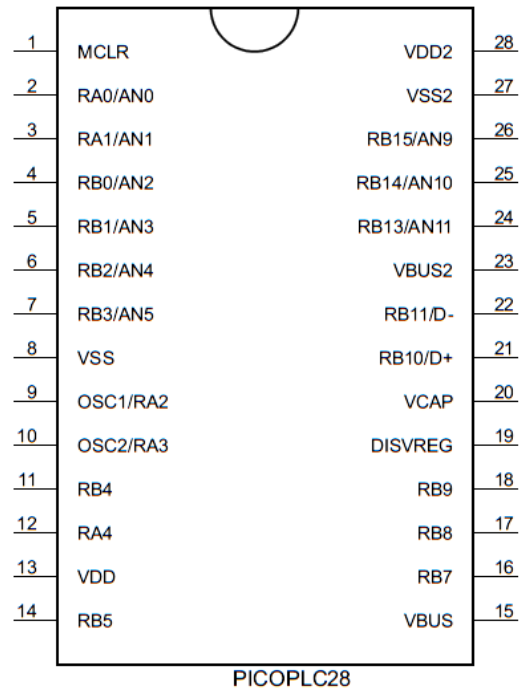
PICoPLC28 PHCxx28 adatlap

rev 1.1

2016.02.10.

PICoPLC

- egyszerű grafikus programozás létra diagramban
- beágyazott környezethez illesztés
- USB-s csatlakozás
- USB-s adatcsere PC-vel
- I/O kezelés
- ADC kezelés
- Timer kezelés
- PWM
- EEPROM
- UART
- Valós idejű szimulátor a PC-n
- Ki- és bemenetek hozzárendelése az ic lábaihoz
- USB adatküldés (mint joystick)



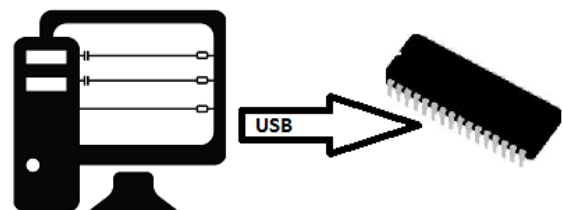
Leírás

A PICoPLC egy mikrokontrolleren futó beágyazott PLC szoftver, amely egyszerűen, grafikus módon (létradiagram) programozható. Ez lehetővé teszi beágyazott rendszerek megvalósítását azok számára is, akik kevésbé jártasak a programozásban. A szükséges szoftvert a PICoPLC létradiagram fejlesztő programjában bárki könnyedén megírhatja, rátöltheti az IC-re, amelyen futtathatja azt. A létradiagram alapvető áramköri kapcsoláshoz hasonlít, így aki kicsit jártas az elektronikában könnyedén elsajátíthatja. A PICoPLC tökéletesen egyesíti a beágyazott és a PLC-s rendszerek előnyeit.

A szoftver tartalmaz minden olyan utasítást, amelyet a hagyományos PLC-k programozásánál ismertek és lehetőség van a beágyazott mikrokontrolleres rendszereknél megszokott funkciók alkalmazására, mint az egyszerű I/O láb kezelésre, timerek használatára, analóg bemenetek olvasására, PWM kimenet megvalósítására,

UART kommunikációra és USB-s adatcserére a PC-vel. A szoftvert a szimulátor segítségével a PC-n tesztelhetjük. Majd mikrokontroller lábakat rendelhetünk a ki- és bemenetekhez és az elkészült szoftvert futtathatjuk a PICoPLC IC-n is.

A PICoPLC rendkívül egyszerűen programozható USB-n keresztül. Nem igényel külön programozót.



Adatok

Tápfeszültség: 3.3V

I/O lábak: 15

Analóg pinek: 9

Lábkiosztás

PICoPLC28 pinout		
Pin	Funkció	Leírás
1	MCLR	Mastre clear
2	RA0/AN0	I/O, analog pin/Vref+
3	RA1/AN1	I/O, analog pin/Vref-
4	RB0/AN2	I/O, analog pin
5	RB1/AN3	I/O, analog pin
6	RB2/AN4	I/O, analog pin
7	RB3/AN5	I/O, analog pin
8	VSS	GND
9	OSC1/RA2	Oscillator
10	OSC2/RA3	Oscillator
11	RB4	I/O
12	RA4	I/O
13	VDD	3.3V
14	RB5	I/O
15	VBUS	USB táp, 5V
16	RB7	I/O
17	RB8	I/O
18	RB9	I/O
19	DISVREG	Conect to GND
20	VCAP	<i>lásd bekötés</i>
21	RB10/D+	USB D+
22	RB11/D-	USB D-
23	VBUS2	3.3V
24	RB13/AN11	I/O, analog pin
25	RB14/AN10	I/O, analog pin
26	RB15/AN9	I/O, analog pin
27	VSS2	GND
28	VDD2	3.3V

PICoPLC utasítások

Az utasításokat kiválaszthatjuk „Utasítások” menüpontból vagy a billentyűk segítségével (a zárójelbe írt karakterek vannak hozzárendelve az adott utasításhoz). A létradiagramban megjelenő utasítások szimbólumaira kattintva további beállítási opciók érhetők el. Többek között nevet adhatunk az utasításban használt változóknak vagy beállíthatjuk a komponensek paramétereit.

- Érintkező(C):
- Tekercs(L): A bemenet állapotát átmásolja egy előre meghatározott helyre, amely lehet a RAM-ban(ekkor a tekercs neve R-rel kezdődik) vagy egy kimeneti láb(ekkor a tekercs neve Y-nal kezdődik). A tekercs leginkább egy ideális relére hasonlít és mindig utolsóként hajtódik végre. Beállítások: Megadhatunk ponált és negált tekercset. Kiválaszthatjuk, hogy belső relé vagy kimeneti láb legyen.
- Késleltetett bekapcsolás(O) - TON: Amikor az utasítás bemenő jele inaktívból aktívá válik, akkor a kimenet a beállított ideig inaktív marad. A késleltetési idő letelte után pedig aktívá válik. A késleltetés ideje beállítható és az inaktív bemenet újraindítja a z időzítést.
- Késleltetett kikapcsolás(F) - TOF: Az utasítás kimenete alapértelmezetten inaktív. Ha a bemenete aktívá válik azonnal aktív lesz a kimenete is, de amikor a bemenet inaktívá válik, akkor a kimenet a beállított ideig még aktív marad, majd az idő leteltével ismét inaktívá válik. A késleltetés ideje beállítható. A kimenet csak akkor válik inaktívá, ha a bemenet megszakítás nélkül inaktív.
- Kumulált késleltetés(T) - RTO: A kumulált késleltetés figyelemmel kíséri, hogy meddig volt a bemenet aktív...
- Reset(E) - RES: Az utasítás olyan időzítő és számláló törlésére szolgál, amelyek nem törlődnek automatikusan (CTU,CTD). Ameddig a RES utasítás bemenete aktív a megadott számláló törlődni fog. A RES utasítás mindig az utolsó utasítás a létrafokon.
- Felfutó él() – OSR: Az utasítás kimenete általában inaktív.

Amennyiben az utasítás bemenete aktív, de az előző ciklusban inaktív volt, akkor a kimenet magas lesz.

Tulajdonképpen ez az utasítás a bemenő jel felfutó élére egy ciklusidejű aktív impulzust generál. Akkor érdemes használni, ha egy esemény sorozatot akarunk indítani felfutó élre.

- Lefutó él() – OSF: Az utasítás kimenete általában inaktív. Amennyiben az utasítás bemenete inaktív, de az előző ciklusban aktív volt, akkor a kimenet magas lesz. Tulajdonképpen ez az utasítás a bemenő jel lefutó élére egy ciklusidejű aktív impulzust generál. Akkor érdemes használni, ha egy esemény sorozatot akarunk indítani lefutó élre.
- Rövidzár, szakadás: A rövidzár nem változtatja meg jelfolyamot, így a bemenete megjelenik kimenetén. A szakadás utasítás kimenete mindig inaktív. Ezek az utasítások hibakeresésre és elhárításra használhatók.
- Főmegszakító – MCR: Alapesetben minden létrafok bemenete aktív vezérléssel bír. Amennyiben egy főmegszakító utasítást inaktív bemenettel hajtunk végre, a megszakítót követő létrafokok bemenete szintén inaktív lesz, egészen a következő főmegszakítóig. (A főmegszakító ágának bemenete mindig aktív) A főmegszakítókat mindig párban érdemes használni. Az első megszakítóval, elkezdjük az áramkör szakaszolását, a másikkal pedig lezárjuk a szakaszolást.
- Értékadás (M) – MOV: Aktív bemenet hatására a célváltozó felveszi a forrás változó vagy konstans értékét. Inaktív vezérlés hatására nem történik semmi, az utasítás hatástalan lesz.

MOV utasítással bármely változónak értéket adhatunk, beleértve a számlálók és időzít állapotváltozóit is. A MOV utasítás mindig az utolsó utasítás a létrafokon.

- Aritmetikai műveletek (+, -, *, /, D) - ADD, SUB, MUL, DIV: Aktív vezérlés hatására a megadott célváltozó értéke az adott művelet eredményének az értékét veszi fel. Az operandusok konstansok vagy tetszőleges változók lehetnek. A műveletek 16 bites előjeles számokkal dolgoznak. A végeredmény minden ciklusban kiértékelődik.
- Összehasonlítás - EQU, NEQ, GTR, GEQ, LES, LEQ: Az utasítás kimenete inaktív, ha a bemenet is inaktív. Aktív bemenettel az utasítás kimenete akkor és csakis akkor aktív, ha az utasításban meghatározott feltétel teljesül. Az utasításokat egyenlőség, kisebb, nagyobb, kisebb vagy egyenlő, nagyobb vagy egyenlő, illetve nem egyenlő típusú vizsgálatokra használhatjuk. A vizsgálati feltétel operandusai változók vagy változó és konstans lehetnek.
- Számlálóok (U, I, J) – CTU, CTD, CTC : A számláló a bementi feltétel felfutó élének hatására egyel növeli (CTU - Count Up) vagy csökkenti (CTD - Count Down) a számláló változójának értékét. A kimeneti aktívvá válik, ha a számláló változó értéke nagyobb, vagy egyenlő a beállított értéknél, egyéb esetekben inaktív. A számláló kimenete lehet aktív akkor is, ha a bemenete inaktív, hiszen a kimenet értéke a változó értékétől függ. Ugyan ahhoz a számláló változóhoz hozzárendelhetünk le- és felfelé számlálókat is. Ebben az esetben az egyik számláló csökkenti, a másik pedig növeli a változót. A RES utasítás

nullázhatja a számlálót, de általános aritmetikai műveleteket is végrehajthatunk a számlálóváltozóján.

- Körkörös számláló (J) – CTC: A körkörös számláló hasonlóan működik, mint a hagyományos CTU számláló, azzal a különbséggel, hogy a megadott határérték elérése után automatikusan nullázza magát. A körkörös számláló a bemeneti feltétel felfutó élére számol. A körkörös számláló utasítás mindig az utolsó utasítás a létrafokon.
- Léptetőregiszter: A léptetőregisztert egy változóhalmazhoz rendeljük hozzá. Például adottak a 'reg0', 'reg1', 'reg2' és 'reg3' változókhoz. A léptetőregiszter kiinduló változója a 'reg0'. A léptetőregiszter a bemeneti feltétel felfutó élére jobbra lépteti a változók értékeit. Tehát 'reg3 := reg2', 'reg2 := reg1' és 'reg1 := reg0'. 'reg0' változatlan. A nagy léptető regiszter sok memóriát fogyaszt a változók eltárolására. A léptetőregiszter utasítás mindig az utolsó utasítás a létrafokon.
- Keresőtáblázat: A kereső táblázat 'n' elemű érték halmaz. Az utasítás aktív vezérlés hatására a táblázat 'i' -edik elemét a 'dest' változóba másolja. Az utasítás nem él vezérelt. A sorszámozás nullától kezdődik, ezért i-nek 0 és (n-1) közé kell esni, ellenkező esetben az utasítás működése nem kiszámítható. A keresőtáblázat utasítás mindig az utolsó utasítás a létrafokon.
- Átviteli karakterisztika: Az átviteli karakterisztika utasítással megoldható összetett függvény vagy jelleggörbe leírása. Jól használható például egy nemlineáris szenzor jelének linearizálására és cél tartományba alakítására.

Tételezzük fel, hogy egy függvénnyel szeretnénk közelítőértékeket számítani és több ponton ismert a függvény x-y összefüggése. Például:

$$f(0) = 2$$

$$f(5) = 10$$

$$f(10) = 50$$

$$f(100) = 100$$

Azaz a

$$(x_0, y_0) = (0, 2)$$

$$(x_1, y_1) = (5, 10)$$

$$(x_2, y_2) = (10, 50)$$

$$(x_3, y_3) = (100, 100)$$

pontok a függvénygörbén vannak. Ez a négy pont felvehető az átviteli karakterisztika utasítás táblázatába. A utasítás beírja az xvar értékhez tartozó y értéket yvar változóba. yvar értéke a táblázat alapján, a táblázat elemeinek sorrendjében lesz kiértékelve. Például ha xvar = 10, akkor az utasítás yvar = 50 hajtja végre. Amennyiben xvar két x érték között van, akkor yvar értéke a két y érték arányos értékével lesz azonos. Például xvar = 55 esetén yvar = 75 lesz. (A figyelembe vett két pont a táblázatból (10, 50) és (100, 100). 55 félúton van 10 és 100 között, és 75 is félúton van 50 és 100 között, tehát (55, 75) a két ponttal meghatározott egyenesen van. Minden x és y pont között egy értelműhöz rendelésnek kell lenni. Minden x pontnak a táblázatban növekvő sorrendben kell követniük egymást. Például 1, 10, 30, 50, 70 megfelelő, de 10, 10, 5, 30, 20 nem szabályos. Bizonyos keresőtáblázatból 16 bites számokkal nem számolhatóak ki az értékek, ekkor a PICoPLC figyelmeztetést ad. Például a lenti táblázat hibaüzenetet ad:

$$(x_0, y_0) = (0, 0)$$

$$(x_1, y_1) = (300, 300)$$

A problémát kiküszöbölhetjük, ha a táblázatot finomítjuk. A lenti és fenti táblázat egyenértékű működési szempontból, de a lenti táblázat nem okoz hibás működést:

$$(x_0, y_0) = (0, 0)$$

$$(x_1, y_1) = (150, 150)$$

$$(x_2, y_2) = (300, 300)$$

Többnyire öt - hat pontnál többre nincs szükség. Ennél több pont feldolgozása jelentősen lassíthatja a szoftver működését. A táblázatban megadott legkisebb x-nél kisebb 'xvar' illetve a legnagyobb x-nél nagyobb 'xvar' értékek beláthatatlan működést eredményeznek. Az átviteli karakterisztika utasítás mindig az utolsó utasítás a létrafokon.

- A/D converter beolvasása: A PICoPLC kihasználhatja a mikrovezérlőn található A/D átalakítót, hogy analóg értékeket olvasson be. Aktív bemenet hatására 'Aname' (vagy amelyet megad a felhasználó) változóba másolódik az A/D átalakító aktuális értéke. 'Aname' változót szabadon módosíthatjuk aritmetikai műveletekkel, felhasználhatjuk átviteli karakterisztika táblázat x értékhez, de használhatjuk értékvizsgáló műveletekhez is (EQU, GRT, stb.) A digitális lábakhoz hasonlóan rendelhetjük hozzá az 'Axxx' változóhoz az analóg lábakat, de figyelembe kell vennünk, nem minden láb működik analóg bemenetként. A szerkesztő ablak alján látható változólistára duplán kattintva felbukkan az analóg láb kiválasztó

táblázat, amelyben csak az analóg lábak szerepelnek. Bizonyos mikrovezérlők esetében meghatározott Analóg/digitális láb kombináció nem lehetséges, még akkor sem, ha a láb egyébként kapcsolódik az AD átalakítóhoz. Amennyiben az utasítás bemenete inaktív, természetesen az 'Aname' változó értéke nem változik. A támogatott mikrovezérlőknél 0V értéknek 0 érték felel meg, míg a tápfeszültséggel azonos feszültség mérési értéke 1023. A mérési érték skálázására aritmetikai műveleteket használhatunk. A A/D converter utasítás utasítás mindig az utolsó utasítás a létrafokon. Alapesetben A/D konverter a tápfeszültséget használja referencia feszültségként. Használhatunk saját referencia feszültséget, amelyeket a Vref+ és/ vagy a Vref- lábakra kell bekötni, valamint az MCU paramétereknél be kell állítani a Vref+ és Vref- használatát.

- PWM kimenet: A PICoPLC kezeli a mikrovezérlőbeépített PWM perifériát. A periféria segítségével PWM jelet állít elő, amely kitöltési tényezője és frekvenciája beállítható. Amikor az utasítás bemenete aktív, a duty_cycle változóban megadott százalékos értékre állítja be a PWM kitöltési tényezőt. A duty_cycle változó 0 értéke állandóan alacsony értéket, míg a duty_cycle = 100 értéke állandóan magas értéket ad ki a mikrovezérlő lábán. (Akik ismerik a PWM periféria működését, feltűnhet, hogy a PICoPLC automatikusan beskalázza a PWM kitöltési tényezőt a periféria beállításaihoz.) Megadhatjuk a PWM célfrekvenciáját Hz-ben. A pontos frekvencia nem

minden esetben érhető el hardverfügőőkokból, de a PICoPLC a legközelebbi lehetséges értéket állítja be. Túl nagy eltérés esetén figyelmeztetést kapunk. Nagy PWM frekvencia eléréséhez szükséges lehet a felbontás csökkentése.

- EEPROM tárolás: Nem elérhető funkció a PICoPLC28-ban.
- UART (soros) vétel: A PICoPLC kezeli a beépített UART perifériát. UART funkció ezért csak olyan mikrovezérlőkön támogatott, melyek tartalmaznak ilyen periféria eszközt. A baud (kommunikáció sebessége) értéket a Beállítások -> MCU paraméterek... menüpontból érhetjük el. A tényleges baud érték a kristály frekvencia függvénye. Túl nagy eltérés esetén figyelmeztetést kapunk. Az utasítás passzív bemenet hatására nem tesz semmit. Aktív bemenet hatására az utasítás megpróbál az UART-ról beolvasni egy bájtot. Ha nem várakozik bájt, akkor az utasítás kimenete passzív, 'var' változó értéke változatlan. Amennyiben sikeresen vettünk egy adatbájtot, egy ciklus időre a kimenet aktív lesz és a vett bájt értéke 'var' változóba íródik.
- UART (soros) küldés: A PICoPLC kezeli a beépített UART perifériát. UART funkció ezért csak olyan mikrovezérlőkön támogatott, melyek tartalmaznak ilyen periféria eszközt. A baud (kommunikáció sebessége) értéket a Beállítások -> MCU paraméterek... menüpontból érhetjük el. A tényleges baud érték a kristály frekvencia függvénye. Túl nagy eltérés esetén figyelmeztetést kapunk. Inaktív bemenet esetén az utasítás nem csinál semmit. Aktív bemenet hatására a 'var' változóban előzetesen megadott karakterértéket az UARTra

küldi. Ebben az utasításban egyszerre 1 byte kiküldésére van lehetőség. A kimeneti feltétel aktív marad mindaddig, míg a karakter küldése folyamatban van, illetve passzív a kimenet, ha a karaktert elküldtük. A soros küldés időt vesz igénybe. Mindenképp ellenőrizzük az utasítás kimenetét, mielőtt újabb karaktert küldenénk. Használhatunk időzítőt is a karakterküldések közt (nem érdemes). Csak akkor aktiválhatjuk az UART (SOROS) KÜLDÉS bemenetét, ha a kimenet már alacsony (azaz az UART nyugalomban van). Mielőtt használnánk ezt az utasítást, érdemes a FORMÁZOTT SZÖVEG UARTON utasítást is átnézni. Sokkal egyszerűbb a használata, és valószínűleg minden feladatot megoldhatunk azzal is.

- Formázott szöveg UART-on: A PICoPLC kezeli a beépített UART perifériát. UART funkció ezért csak olyan mikrovezérlőkön támogatott, melyek tartalmaznak ilyen periféria eszközt. A baud (kommunikáció sebessége) értéket a Beállítások -> MCU paraméterek... menüpontból érhetjük el. A tényleges baud érték a kristály frekvencia függvénye. Túl nagy eltérés esetén figyelmeztetést kapunk. A bementi állapot felfutó élének hatására az utasítás elküldi a teljes formázott szöveget. Leállításra küldés közben nincs mód. Amennyiben a szöveg '\3' helyettesítő szekvenciát tartalmaz, 'var' szöveggé alakított értéke három karakterrel beszűródik a szövegbe. '\3' esetén pontosan három karaktert tartalmazó szöveget szúrunk be. Amennyiben 'var' értéke 35, akkor az elküldött szöveg:

'Nyomas : 35\r\n' (szóközzel kiegészítve) Viszont, ha 'var' értéke pl. 1432 lenne, akkor az utasítás

viselkedése előreláthatatlan, hiszen 1432-ben több mint 3 számjegy található. Ebben az esetben használjuk a '\4' szekvenciát.

Olyan változóknál, melyek negatív értéket is felvehetnek, használjuk a '\-3' (vagy '\-4' stb.) formátumot. Ekkor a PICoPLC beszúr egy szóközt a nem negatív értékek elé és egy mínuszjelet tesz a negatív értékek elé.

Több UART KÜLDÉS utasítás egyidejű aktiválása, előreláthatatlan működést eredményez. Fix szöveg küldésre is használhatjuk ezt az utasítást. Ilyenkor hagyjuk ki a '\3' ('\4', stb.) szekvenciát. Fordított per jel(backslash) küldéshez használjuk a '\\' karakter szekvenciát. További speciális karakter szekvenciák:

- * \r -- kocsni vissza jel
- * \n -- újsor
- * \f -- lapdobás
- * \b -- visszatörlés
- * \xAB -- ASCII karakter 0xAB

(hex) értékkel (pl. '\x20' szóköz karakter)

Az utasítás kimenete a küldés folyamán aktív, ha a küldés véget ért passzív.

Az utasítás sok programmemóriát használ, ezért bánjunk vele takarékosan.

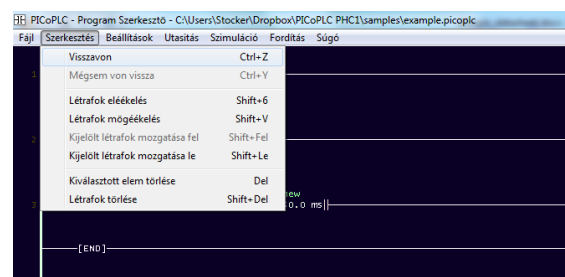
- Felhasználói funkció: PICoASM fájl futtatását teszi lehetővé. A fájl betöltése és fordítása fordítási időben történik. Az utasítás lehelyezésekor semmilyen ellenőrzés nincs. Hibás és nem létező fájlnevet is megadhatunk, ez csak fordítás (szimuláció indításakor is van fordítás!) közben lesz ellenőrizve. A felhasználói funkció (PICoASM) fájl a bementi feltételtől függetlenül futtatásra kerül. Ezzel lehetőségünk nyílik speciális funkcionalitás elérésére is (pl. TOF -

ahol vezérlés nélkül is van kimenet) A funkció kimeneti értéke az egyben az elem kimeneti állapota is. A PICoASM szintaxist illetően részleteket a PICoASM dokumentáció tartalmaz.

- Felhasználói rutin: PICoASM fájl futtatását teszi lehetővé. A fájl betöltése és fordítása fordítási időben történik. Az utasítás lehelyezésekor semmilyen ellenőrzés nincs. Hibás és nem létező fájlnevet is megadhatunk, ez csak fordítás (szimuláció indításakor is van fordítás!) A felhasználói funkció (PICoASM) fájl a bementi feltételtől függetlenül futtatásra kerül. Ezzel lehetőségünk nyílik speciális funkcionalitás elérésére is (pl. TOF - ahol vezérlés nélkül is van kimenet) A funkció kimeneti értéke az egyben az elem kimeneti állapota is. A PICoASM szintaxist illetően részleteket a PICoASM dokumentáció tartalmaz.

Létradiagram szerkesztése

A létradiagram létrafokait a *Szerkesztés* menüpontban található alábbi parancsokkal szerkeszthetjük:



- Visszavon: utasítás visszavonása
- Mégsem von vissza: visszavonás törlése
- Létrafok elékelése: egy új létrafokot helyez azon fok elé, amelyben a kurzor található

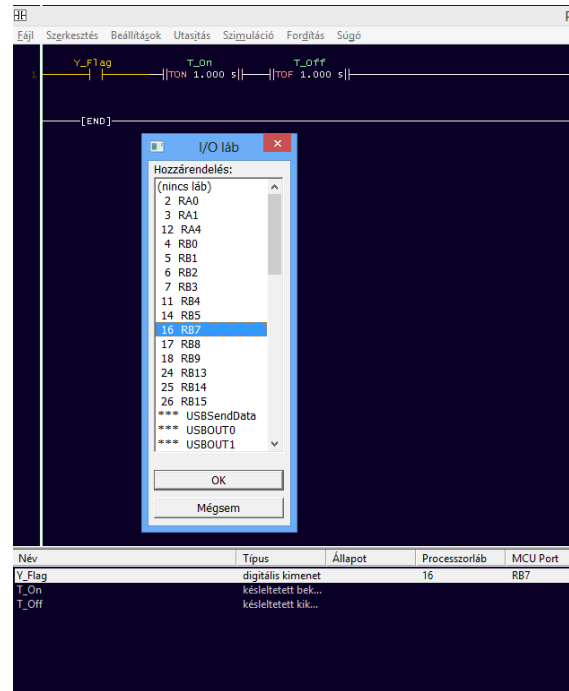
- Létrafok mögé ékelése: egy új létrafokot helyez azon fok mögé, amelyben a kurzor található
- Kijelölt létrafok mozgatása fel: a kurzorral kijelölt létrafok pozícióját kicseréli a felette található fokkal
- Kijelölt létrafok mozgatása le: a kurzorral kijelölt létrafok pozícióját kicseréli az alatta található fokkal
- Kiválasztott elem törlése: a kurzor által kijelölt elemet törli
- Létrafok törlése: a kurzor által kijelölt teljes létrafokot törli

MCU lábak összerendelése a szoftverrel:

A PICoPLC szoftver elemeit hozzá kell rendelni az kiválasztott mikrokontroller lábához vagy egy USB lábhoz (USB-s kezelést lásd a következő pontban). Minden komponens megjelenik a PICoPLC IDE alján található ablakban. Itt az adott sorra kattintva megjelennek a kiválasztható MCU lábak, amelyek tetszőlegesen hozzárendelhetők az adott komponenshez. A hozzárendelést be kell állítani az alábbi esetekben:

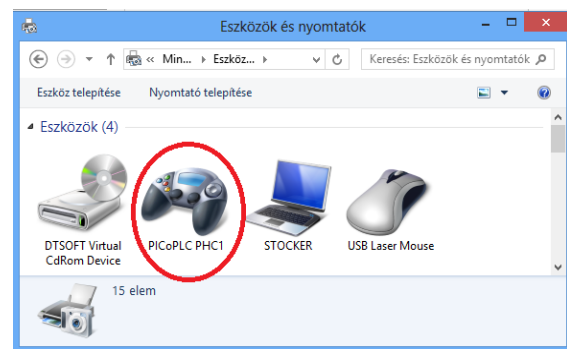
- ha egy tekercset vagy érintkezőt MC lábna lett beállítva
- ADC utasításnál (analóg bemenet lesz)
- PWM utasításnál (PWM kimenet lesz)
- UART adás és vétel esetén

(Az összerendelés elmulasztása esetén a fordító hibát jelez a fordításnál)



USB kommunikáció:

A PICoPLC futás közben USB-n a PC-re kötve egy joystickként jelentkezik be, lehetővé téve adatok továbbítását egy PC-n futó szoftver számára.

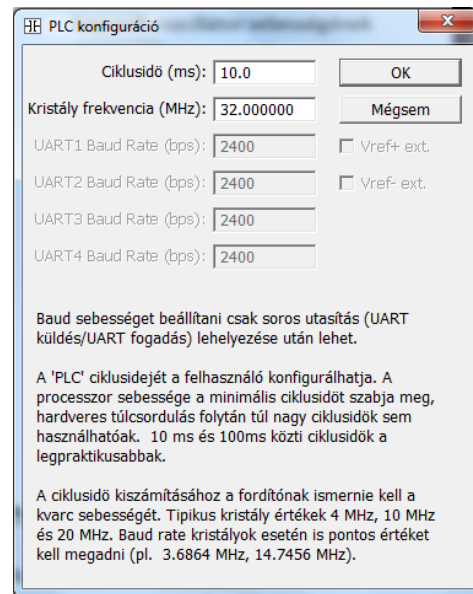


A komponensek hozzárendelhetők a joystick változóhoz. A Tekercs utasítás hozzárendelhető a joystick egy USB bitjéhez, a PWM érék pedig egy USB változóhoz, amelyeket USB-n keresztül felküld a PICoPLC a PC-nek.

Az adatok kezelésénél egy PC szoftverben egy szabványos USB-s joystick kezelést kell megvalósítani.

MCU paraméterek:*Beállítások/MCU Paraméterek* menüpont

- Ciklusidő: a PICoPLC újra és újra lefutatja az utasításokat. A ciklusidő megadja, hogy milyen időközönként fusson le a szoftver
- Kristályfrekvencia: **Mindenképp 20 MHz-es kristályt használjunk az applikációinkban**, különben a PICoPLC nem fog működni. Amennyiben kiválasztottuk a megfelelő mikrokontrollert, a kristályfrekvencia automatikusan mutatni fogja a belső frekvenciát. **Külső kristály frekvencia: 20 MHz**
- Referencia feszültség: Amennyiben a Vref+ és/vagy a Vref- beállítási paraméter be van pipálva az A/D konverter nem a tápfeszültséget használja referenciafeszültségként, hanem a Vref+ és Vref- lábakra bekötött feszültséget.
- Baudrate: UART (soros port) kommunikáció esetén szükséges a kommunikációs sebesség beállítása
Min érték: 9600 bit/s
Max érték: 115200 bit/s
Max 4 UART használható.

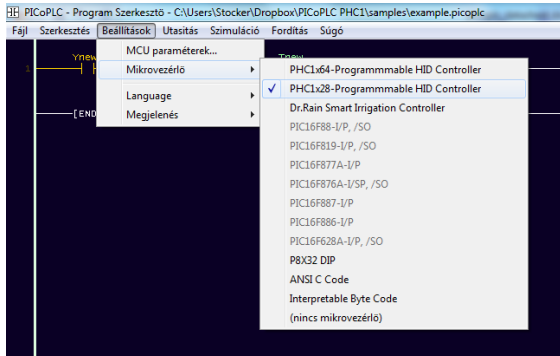
**Programozás**

Az kész szoftvert PICoPLC-re tölthetjük.

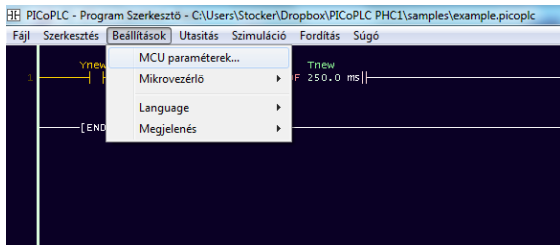
Ekkor a program a tápfeszültség bekapcsolása után futni fog a mikrokontrolleren.

A PICoPLC felprogramozásához az alábbi lépések szükségesek:

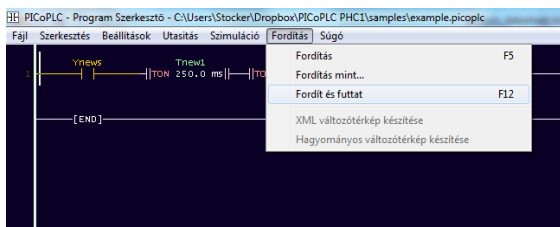
1. Mikrokontroller és PC összekapcsolása USB-n: első lépésként a mikrokontrollert USB-n keresztül kell csatlakoztatni a PC-hez.
2. Mikrokontroller kiválasztása: a *Beállítások/Mikrovezérlő* menüpont alatt ki kell választani az adott mikrokontrollert: *PHC1x28-Programable HID Controller*



3. Mikro vezérlő paramétereinek beállítása: a *Beállítások/MCU Paraméterek* menüpont alatt be kell állítani a mikrokontrollerünk paramétereit (Lásd: MCU Paraméterek, oldal: 9.oldal).



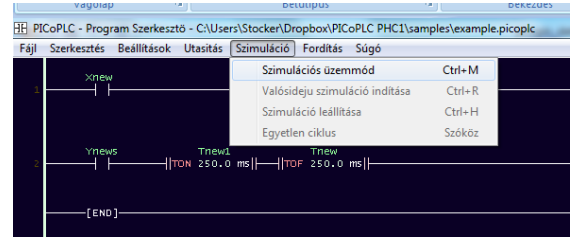
4. Fordítás: a szoftvert le kell fordítani, majd a mikrokontrollerre tölteni. Ezt a *Fordítás/Fordít és futtat* menüpont segítségével kell megtenni. Ekkor a létradiagramban írt program lefordul és a kiválasztott mikrokontroller flash-be íródik.



Szimuláció

A PICoPLC szoftver része ain szimulátor, amelyben a létradiagramban megírt program a hardver jelenléte nélkül tesztelhető. Ebben az esetben egy virtuális

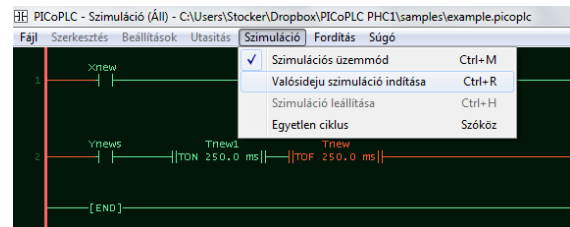
hardveren fut a program. A szimulátorban követhető a programunk futása, láthatóak a változók aktuális értékei, illetve manuálisan kapcsolhatjuk a különböző utasításokat. A szimulátort a *Szimuláció/Szimulációs üzemmód* menüpontban választhatjuk ki (a szimulátor üzemmódo kis pipa jelzi).



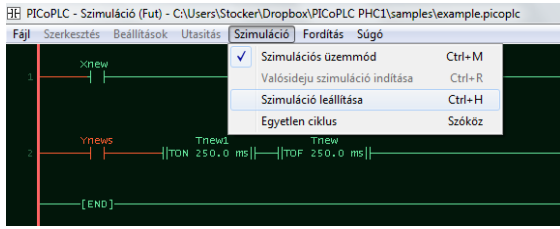
Futtatás során az aktív ágak (rövidre zárt) piros, a megszakítottak (szakadás) pedig zöld színűvé válnak.

Egyes utasítások manuálisa állítható. Például érintkező esetén az egérrel a szimbólumára kattintva zárható rövidre vagy szakítható meg az utasítás.

A szimuláció futtatásánál két lehetőség van. Vagy egyetlen ciklus fut le vagy a szoftver ciklikusan fut. A szimuláció a *Szimuláció/ Valósídejű szimuláció indítása*, illetve a *Szimuláció/Egyetlen ciklus* menüponttal indítható.



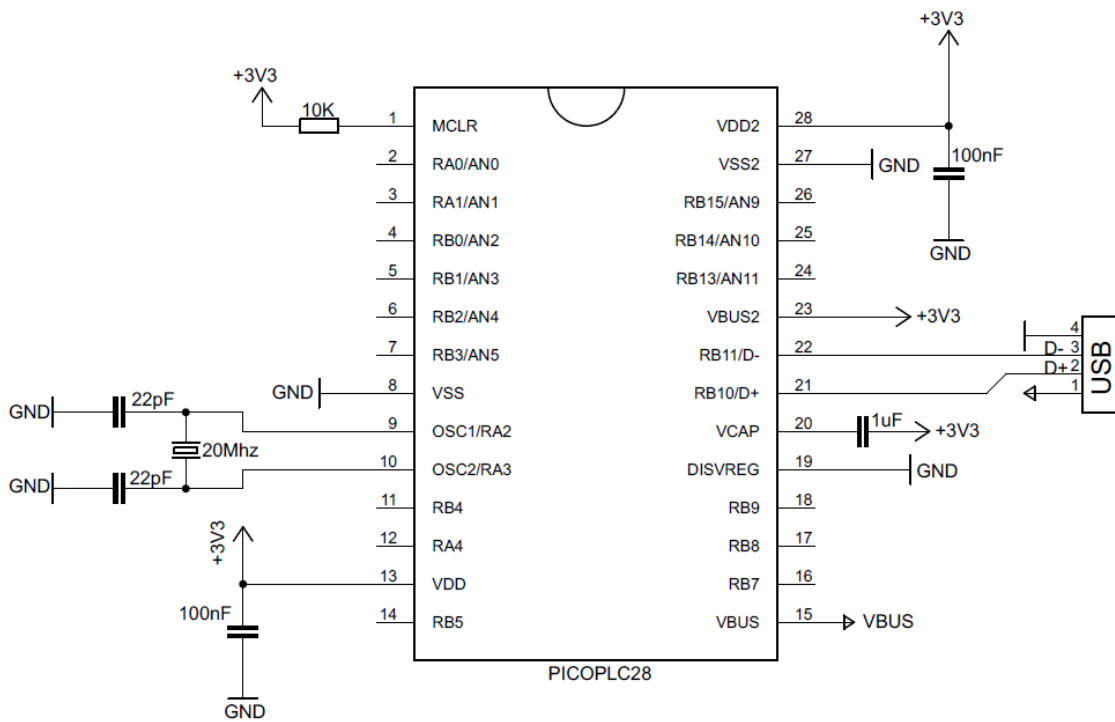
A szimuláció a *Szimuláció/Szimuláció leállítás* menüponttal szakítható meg.



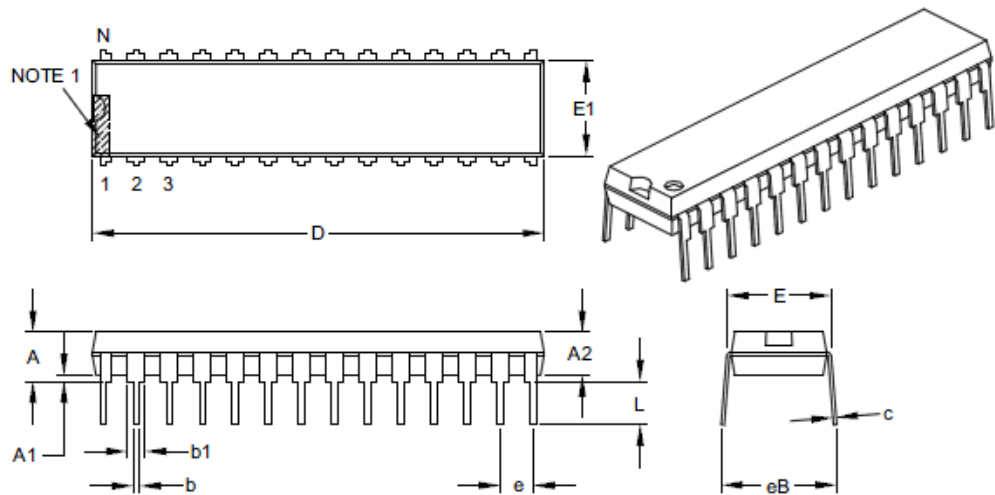
Szimulátor üzemmódban a létradiagram utasításai nem módosíthatóak.

Tipikus bekötés

20MHz-es kristály használata szükséges a helyes működéshez!



Műszaki rajz



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	-	-	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	-	-
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	-	-	.430

Notes:

Szupport, szoftver letöltés és példa programok:

Evotronics Kft

www.evotronics.eu

info@evotronics.eu

Verzió 1.0

Figyelmeztetés:

Ez a fejlesztő/oktató eszköz kutatási és laboratóriumi használatra lett kifejlesztve, ezért ne használja olyan applikációban, amely veszélyeztethet emberi életet, vagy vagyoni kárt okozhat! Nem egy végleges eszköz, amelyet egy kereskedelmi forgalomba kerülő termékbe beépítenek. A gyártó nem vállal felelősséget, ha a termék esetleges beépítése után valamilyen anyagi kár, vagy személyi sérülés történik.