

CompPlc általános célú vezérlő leírása, programozási lehetőségei

A CompPlc vezérlő modul tervezési szempontjainál fontos tényező volt a flexibilitás, moduláris és kompakt felépítés, könnyű programozhatóság. Egy másik meghatározó szempont volt, az általunk megfogalmazott feladatkör kiszolgálásához való illeszkedés. Ennek megfelelően mind hardver, mind szoftver oldalról kompromisszumokkal járt a megvalósítás.

Ennek eredményeként egy jól áttekinthető, funkcionális modulokkal bővíthető felépítést kaptunk. Természetesen a megkötések olyan megoldásokat tettek szükségessé, ami bizonyos tekintetben egyszerűsített felépítést eredményezett az input/output felületeken. Így a kimeneti csatornák open drain felépítésűek, a bemeneti csatornák pedig nem tartalmaznak galvanikus szintelválasztást. Az üzemi feszültségszintet 12V-ban határoztuk meg, így a kimenetek DC 12V reléekkel, a bemenetek pedig DC 0..5V, DC 0...12 V jelszintekkel működnek. Az egyes vezérlő modulok önálló kapcsolóüzemű tápegységgel rendelkeznek. Így egy adott vezérlés kialakításához külön DC 12V-os tápegység alkalmazása szükséges – javasolt a rövidzár védett kapcsolóüzemű tápellátás alkalmazása.

A felépítésből/méretből adódóan korlátozott a be- és kimenetek száma, ez azonban rugalmasan bővíthető RS485 buszon keresztül.

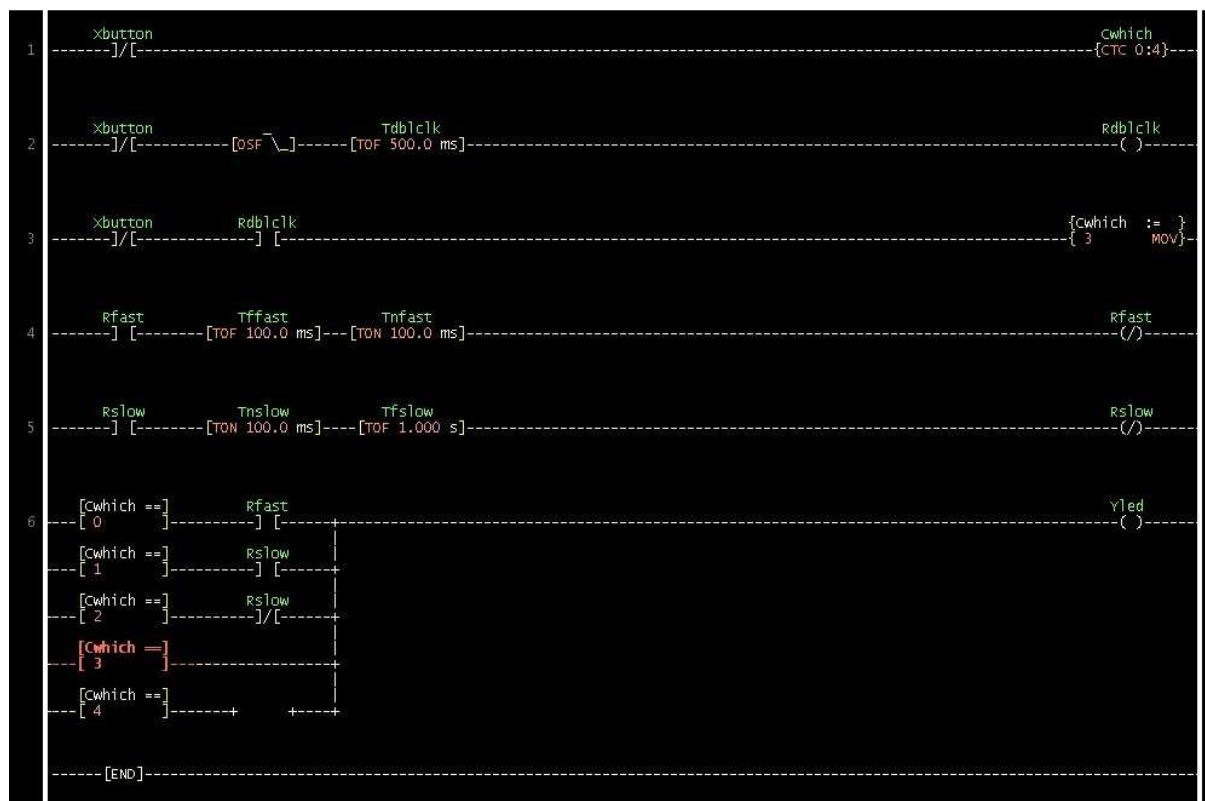
A végleges tervek elkészülése során – amelyek speciális feladat megoldására lettek hangolva – felmerült annak a szükségessége, hogy ne csak professzionális programozói ismeretek birtokában legyen alkalmazható a készülékcsalád, ezzel bővítve az esetleges alkalmazók körét. Ezért vezérlő modulok három különböző MCU alkalmazásával készültek, az alkalmazásnak megfelelő skálázási lehetőség miatt.

A PLC_877 PIC16F877A MCU-t tartalmaz. Elsősorban általános vezérlési, szabályozási feladatokra alkalmas. Ez az a vezérlő modul amely funkció blokkdiagram formában programozható. Az LDMICRO alkalmazásával az egyszerű felhasználást tettük elérhetővé, de természetesen bármilyen más fejlesztői környezet is használható.

Az LDMICRO a teljesség igénye nélkül az alábbi lehetőségeket támogatja:

- digitális ki- bemenetek kezelése
- időzítők (bekapcsolási késleltetés, kikapcsolási késleltetés stb.)
- számlálók
- analóg be- és kimenetek kezelése (ADC felbontás az alkalmazott MCU felépítéséből következik)
- egész számok kezelése, aritmetikai műveletek
- soros kommunikáció
- léptetőregiszterek, lookup táblázatok
- EEPROM változók kezelése (kikapcsolás esetén sem törlődik)

Az alábbi ábra egy változtatható üzemmódú villogó LED logikai funkció diagramját ábrázolja:



Az XBUTTON megnyomásával YLED öt különböző üzemmódban működik.

Az adott fejlesztői környezet natív kódot generál, amely a példa esetében a felhasználói programmemória 5%-át használja fel.

A PLC_252 (PIC18F252) vezérlő modul nagyobb programmemóriával rendelkezik így a kódméret növelhető. Ennél a vezérlő modulnál már nem használható az LDMICRO. A kód létrehozásához egyéb, általános fejlesztői környezetre van szükség, amely tetszőlegesen megválasztható a piacon megtalálhatóak közül. Ez egyben azt is jelenti, hogy a nyelvi lehetőségek bővülése miatt bonyolult, számítógépes feladatok is kiszolgálhatóak.

A PLC_252 és PLC_4013 vezérlők preemtív multitaszking működésmódúak. (A korszerű operációs rendszerek kernelei is ezt a technikát alkalmazzák. Pl.: UNIX, LINUX, Windows)

A PLC_4013 (dsPIC30F4013) vezérlő modullal felépítésénél fogva nagyteljesítményű jelfeldolgozást valósíthatunk meg tovább bővítve az alkalmazási területet.

Egy vezérlő modul és egy port modul szükséges a minimál kiépítéshez. A lehetőségeket opcionálisan megválasztott modulokkal bővíthetjük. A Terminál modul lehetőséget teremt a folyamatos online monitorozásra, beavatkozásra, a vezérlők dinamikus konfigurálására, karakteres és grafikus folyamatkijelzésre. Az Ethernet/Storage modul módot ad egyrészt arra, hogy TCP/IP protokollal alkalmazásával nyilvános hálózaton keresztül elérjük a CompPLC-ket, beágyazott Webszerverükön

CompPlc

keresztül online távoli beavatkozást biztosíthatunk. Lehetőség van FTP szerverekhez való kapcsolódásra is, amely a mérési, vezérlési, szabályozási folyamatok naplózását, adatgyűjtését teszi lehetővé. Másrészt az MMC/SD memóriakártya segítségével lehetőség nyílik lokális, adat és natív kód tárolásra. A modulokon megtalálható RS232 és RS485 illesztők számtalan eszköz, számítógép, végberendezés csatlakoztatását teszik lehetővé.

Az alábbi kódrészlet egy rezgésdiagnosztikai alkalmazás része, amely real-time fourier analízist végez - egyidejűleg több berendezésen - összehasonlítva a mérési adatokat a tárolt „rezgéslenyomattal”, megelőzve a katasztrofális meghibásodást:

```
/*
  This computes an in-place complex-to-complex FFT
  x and y are the real and imaginary arrays of 2^m points.
  dir = 1 gives forward transform
  dir = -1 gives reverse transform
*/
short FFT(short int dir,long m,double *x,double *y)
{
    long n,i,i1,j,k,i2,l,l1,l2;
    double c1,c2,tx,ty,t1,t2,u1,u2,z;

    /* Calculate the number of points */
    n = 1;
    for (i=0;i<m;i++)
        n *= 2;

    /* Do the bit reversal */
    i2 = n >> 1;
    j = 0;
    for (i=0;i<n-1;i++) {
        if (i < j) {
            tx = x[i];
            ty = y[i];
            x[i] = x[j];
            y[i] = y[j];
            x[j] = tx;
            y[j] = ty;
        }
        k = i2;
        while (k <= j) {
            j -= k;
            k >>= 1;
        }
        j += k;
    }

    /* Compute the FFT */
    c1 = -1.0;
    c2 = 0.0;
    l2 = 1;
    for (l=0;l<m;l++) {
        l1 = l2;
        l2 <=< 1;
        u1 = 1.0;
        u2 = 0.0;
        for (j=0;j<l1;j++) {
            for (i=j;i<n;i+=l2) {
                i1 = i + l1;
```

CompPlc

```
        t1 = u1 * x[i1] - u2 * y[i1];
        t2 = u1 * y[i1] + u2 * x[i1];
        x[i1] = x[i] - t1;
        y[i1] = y[i] - t2;
        x[i] += t1;
        y[i] += t2;
    }
    z = u1 * c1 - u2 * c2;
    u2 = u1 * c2 + u2 * c1;
    u1 = z;
}
c2 = sqrt((1.0 - c1) / 2.0);
if (dir == 1)
    c2 = -c2;
c1 = sqrt((1.0 + c1) / 2.0);
}

/* Scaling for forward transform */
if (dir == 1) {
    for (i=0;i<n;i++) {
        x[i] /= n;
        y[i] /= n;
    }
}

return(TRUE);
}
```

A következő kódrészlet egy egyszerű ethernet demonstráció, amely a beérkező pingre választ küld:

```
dim myMacAddr    as byte[6]      ' my MAC address
      myIpAddr    as byte[4]      ' my IP address

main:
    ADPCFG = 0xFBFF                ' ADC converters will be used (channel10)

    PORTB = 0
    TRISB = 0xFFFF                ' set PORTB as input for buttons

    PORTD = 0x00
    TRISD = 0                      ' set PORTD as output

    httpCounter = 0

    myMacAddr[0] = 0x00
    myMacAddr[1] = 0x14
    myMacAddr[2] = 0xA5
    myMacAddr[3] = 0x76
    myMacAddr[4] = 0x19
    myMacAddr[5] = 0x3F

    myIpAddr[0] = 192
    myIpAddr[1] = 168
    myIpAddr[2] = 20
    myIpAddr[3] = 60

    '*
```

CompPlc

```
'* starts with :
'* reset bit on RF0
'* CS bit on RF1
'* my MAC & IP address
'* full duplex
'*

' for faster spi communication use Spi_Init_Advanced routine
Spi2_Init()
Spi_Ethernet_Init(PORTG, 13, PORTG, 12, myMacAddr, myIpAddr,
SPI_Ethernet_FULLDUPLEX)
while true                                ' do forever
    Spi_Ethernet_doPacket()                ' process incoming Ethernet packets

    '*
    '* add your stuff here if needed
    '* _doPacket() must be called as often as possible
    '* otherwise packets could be lost
    '*
wend
```

Az előzőekben vázolt működésmódból és a mellékelt kódrészletekből talán kitűnik a készülék széleskörű és igen flexibilis alkalmazhatósága, arról nem is beszélve, hogy egy-egy újabb feladatmegoldás igényeinek megfelelően tetszőleges port modult hozhatunk létre. Így a tervezésénél maximálisan tekintettel lehetünk az új körülményekre. (speciális zavarvédelemi kialakítás, kibemeneti struktúra módosítása).