



Firmware frissítő alkalmazás fejlesztése

Dajka Attila Norbert

Konzulensek: Molnár Károly, Bilau Zoltán

A feladat

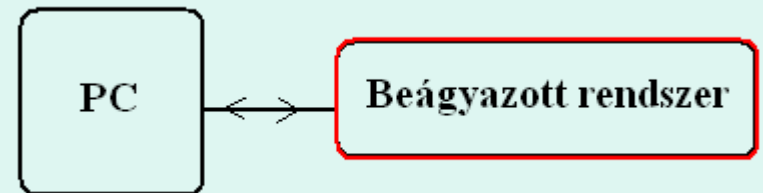
Általános célú firmware frissítő létrehozása

Firmware: beágyazott rendszer alkalmazói programja

- Távoli csere lehetősége
- Frissítés automatikus elvégzése

Konkrét alkalmazás:

áramlásmérő ATmega 128-cal



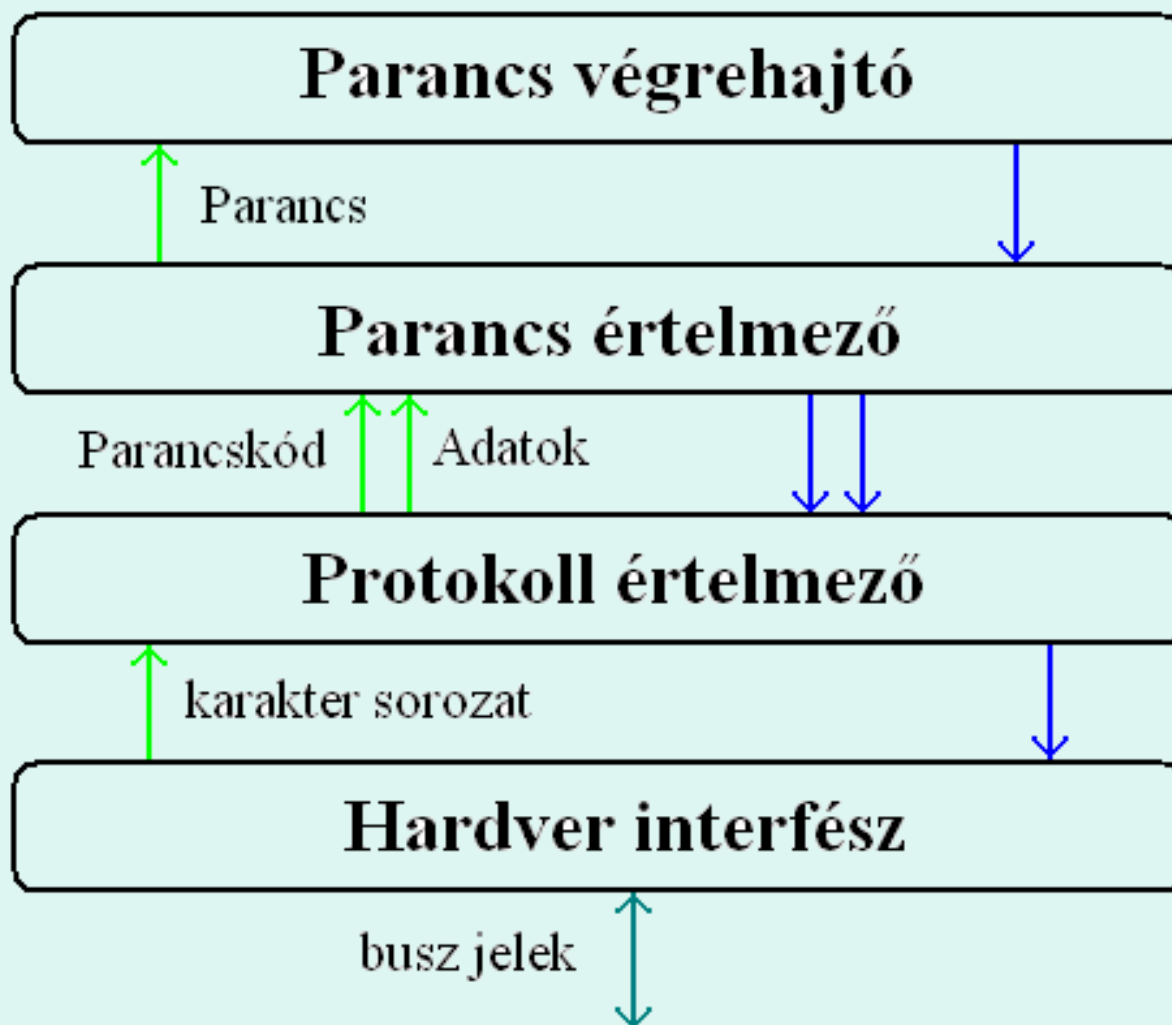
Motiváció I.

- Újraprogramozás az üzem közben használt protokollal
- Nincs szükség kezelőszemélyzet kiküldésére az eszközhöz
- Központilag a busz master képes lesz az újraprogramozást lefolytatni
- Egyszerű verziófrissítés
- Kevésbé körülményes hibajavítás

Motiváció II.

- Kontrollerspecifikus és kontrollerfüggetlen részek
- **Kontrollerspecifikus** részeken változtatni kell új eszközön való használat előtt
- **Kontrollerfüggetlen** részek számos alkalmazásban használhatóak

Firmware frissítő felépítése



Modulok portolhatósága

Parancs végrehajtó függvények
AVR Libc segítségével

Konkrét alkalmazás:
áramlásmérő ATmega 128-cal

Parancs értelmező C kódban
megírva, változtatás nélkül
felhasználható

Parancs végrehajtó	processzorfüggő
Parancs értelmező	portolható
Protokoll értelmező	portolható
Hardver interfész	processzorfüggő

Protokoll értelmező szintén
processzorfüggetlen C kód

Hardver interfész az USART
regiszterekből olvas ill. ír.
Processzoronként más és más

Kommunikáció formája

Fizikai réteg - aszinkron soros átvitel: RS-485

Alkalmazási réteg – Modbus ASCII illetve Escaped protokoll

A Modbus ASCII egy széles körben elterjedt protokoll

Az Escaped egy saját belső használatra kifejlesztett protokoll

Modbus ASCII	Escaped
karakterek	bájtok
LRC 8 biten	CRC 16 biten
maximális adathossz: 253 bájt	maximális adathossz: 1023 bájt

Frissítő funkciói

Flash, EEPROM és RAM:

- Írás
- Olvasás
- Törlés
- Folyamatban levő írás érzékelése

Konzisztencia:

Legyen a futtatni kívánt program 2 dimeziós adattömbként értelmezve

Ellenőrzőösszeg számítás 2 vektort eredményez a 2 dimenzióra

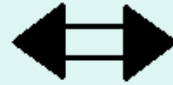
Egyéb funkciók:

- Eszköz leállítása
- Újraindítása
- Verzió információ lekérdezése

...	Bootflash
info	
x, y, *consistencyreg	Application flash
info	
Program with dimensions [x,y]	
consistencyreg	
...	

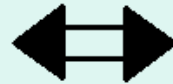
Hardver interfész

Fogadás és küldés megszakítás
kezelő rutinokkal



Fogadás és küldés USART
vezérlő és státusz regiszter
olvasással megszakítás nélkül

Bájt fogadásnál és küldésnél
nincs várakozás

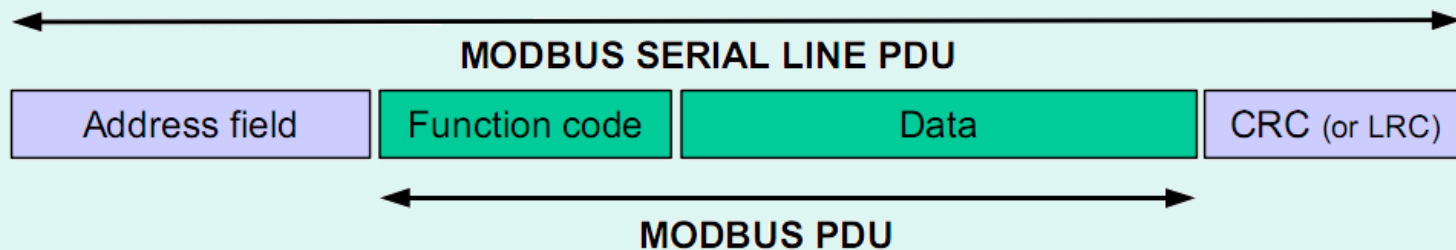


Bájt fogadás
implementációja blokkoló



- USART regiszterek inicializálása megszakításos üzenetkezelésre
- Beérkező és kimenő bájtok tárolása 2 FIFO-ban
- FIFO-k állapotát lekérdező függvények
- Hibadetektálás: Keret-, paritás-, és adattúlfutáshibák érzékelése

Protokoll értelmező



```
Protokoll értelmező(parancskód, *adat, adathossz) {  
    // várakozás beérkező karakterekre  
    // üzenet részek összerakása  
    // hibellenőrzés  
    // parancs értelmező hívása  
    // átvitel inicializálása, ha van kimenő üzenet  
    // átküldendő bájtok karakterekké alakítása  
    // küldés }  
}
```

Összefoglalás, kitekintés

- ✓ Részletes specifikációt elkészítettem
- ✓ A hardver interfész függvényeit és rutinjait megírtam
- ✓ A protokoll értelmező kódja is elkészült
- ❑ A parancs értelmező részben kész
- ❑ A parancs végrehajtó megírása és a hardveren történő debuggolás ezután következik

Köszönöm a figyelmet!