Vezeték nélküli újrakonfigurálhatóság szenzorhálózatban

Készítette: Kalocsai Dávid

Konzulens: Orosz György

BME – MIT

2010.tavasz

Önálló labor 2

BMEVIMIM852

Bevezetés

- Feladat
- Mi is az a Bootloader?
- Mikrokontroller Flash
- Bootloader programozása
- Egy bootloader példa
- Összefoglalás, további célok

Feladat

- **Szenzorhálozatokban**
  - **egyre több vezetéknélküli kommunikációt használó adatgyűjtő**
  - **Intelligens szenzorokat, mikrokontroller alapú egységek**
  - **A fejlesztés során funkciók módosítása**
  - **Nagy területen elosztott**
  - **Nehezen hozzáférhető egységeket**
  - **Fizikailag hozzá kell férni.**

CÉL: Vezetéknélküli kommunikációs csatornán keresztül megvalósított újrakonfiguráció

Bootloader szerepe

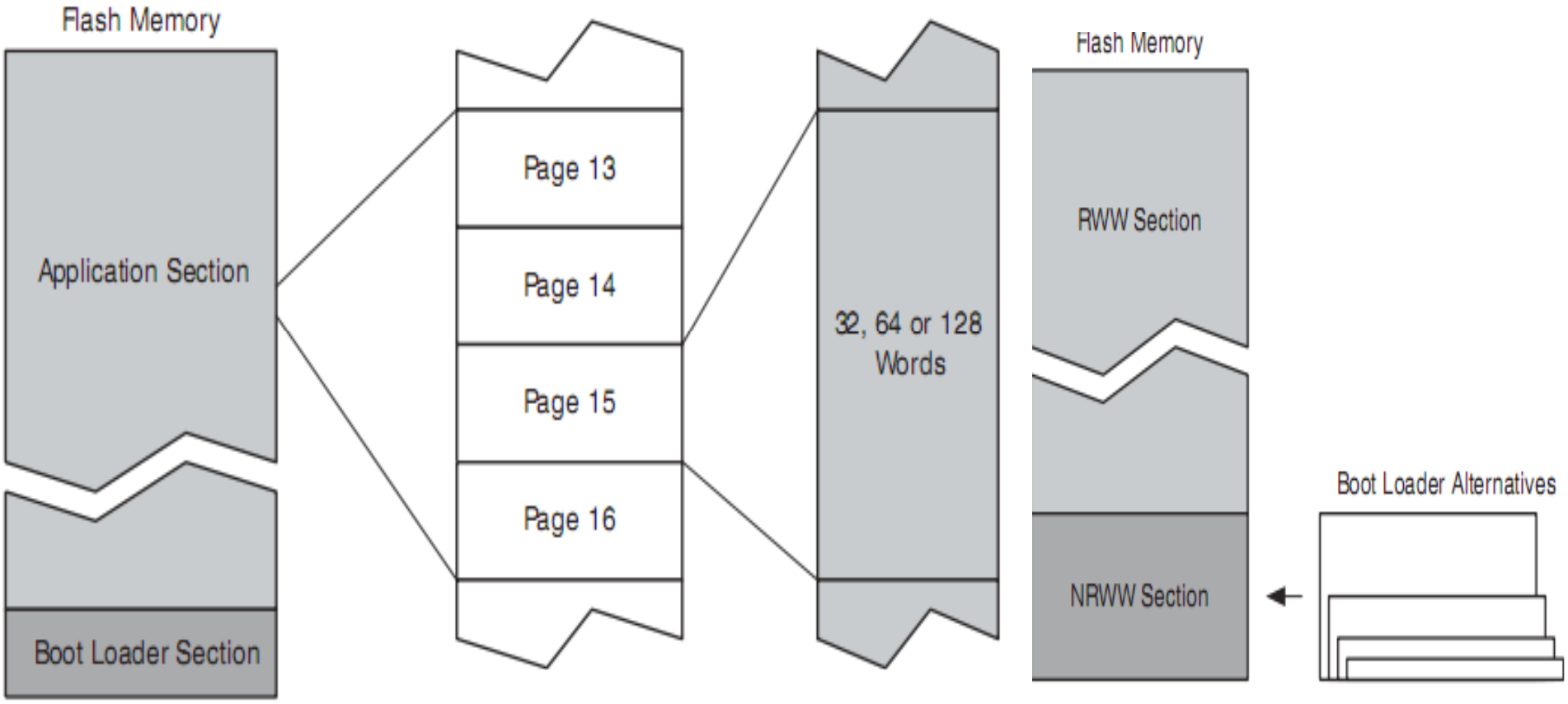
Mi az a bootloader?

- Kis méretű program
- Programmemóriában dedikált helyen
- Perifériákon keresztüli kommunikációval önprogramozás

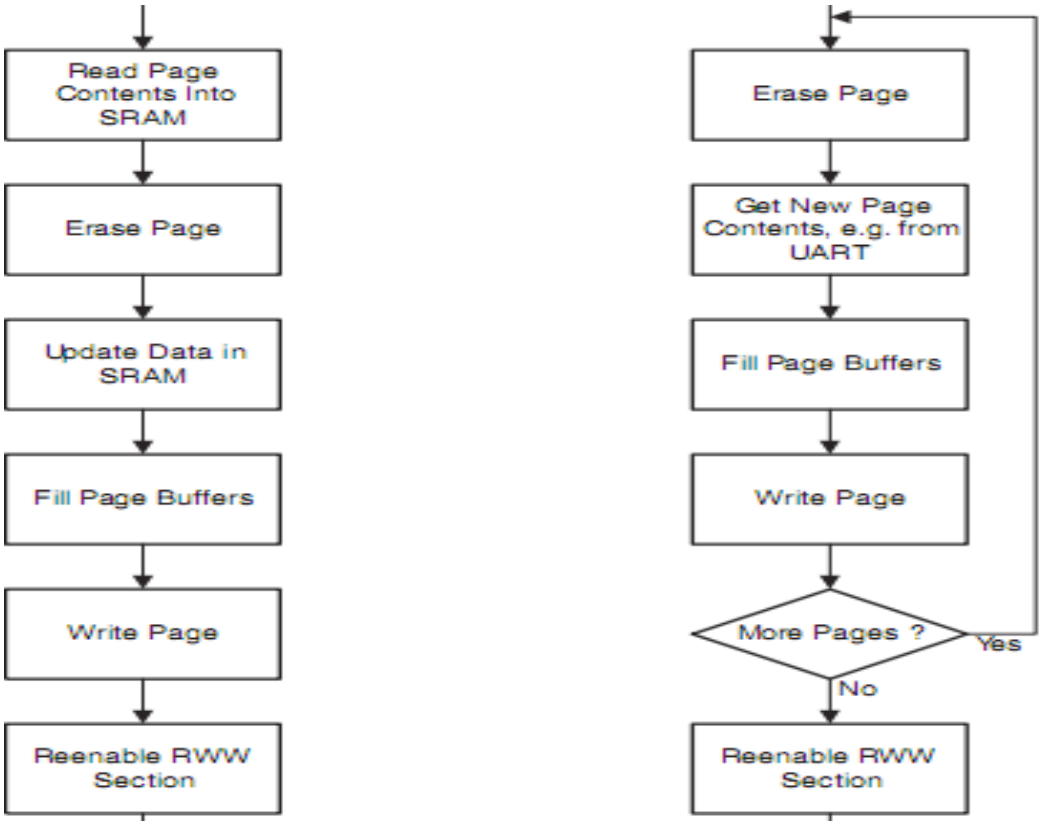
Mire használható a bootloader?

- Programozó-hardver kiküszöblésére
- Firmware frissítés
  - bootloaderbe dekódoló algoritmust is beépíthetünk

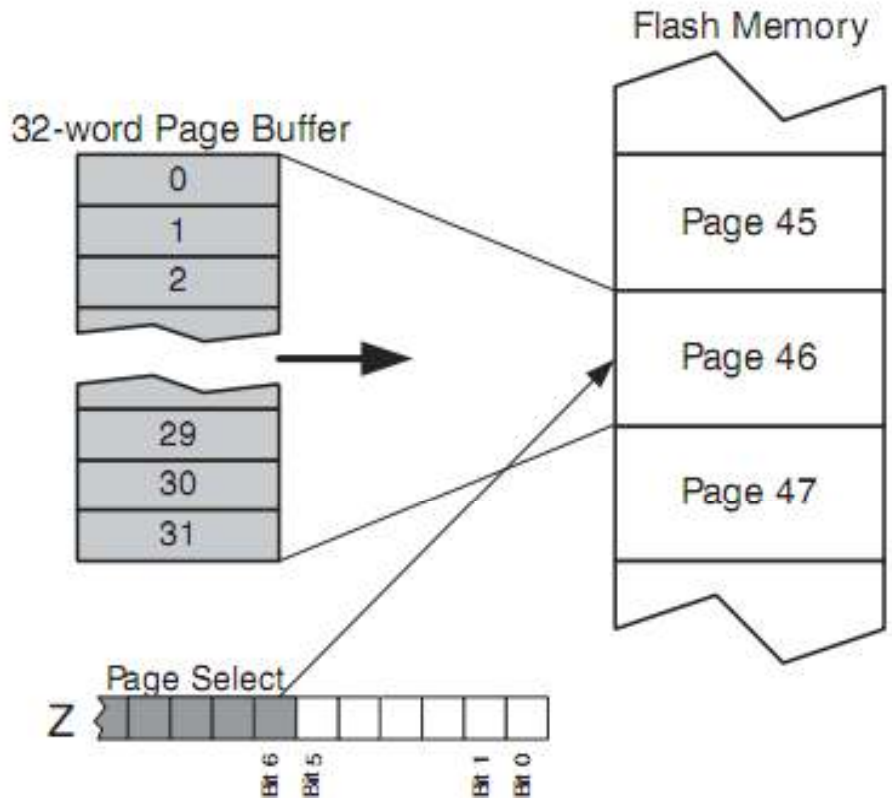
Flash


Flash Memory
Application Section
Boot Loader Section
Page 13
Page 14
Page 15
Page 16
32, 64 or 128 Words
Flash Memory
RWW Section
NRWW Section
Boot Loader Alternatives

Bootloader programozása

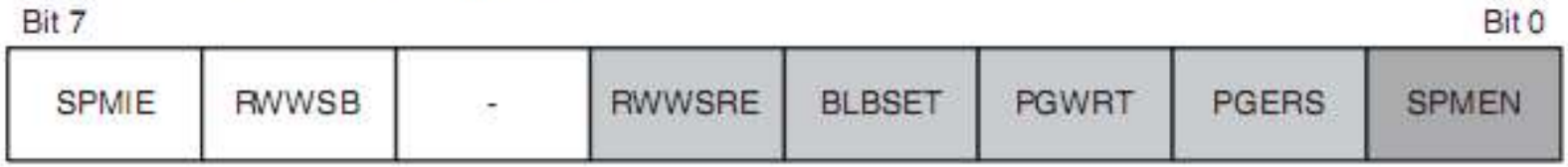


Page Erase,Load and Write



Bootloader programozása

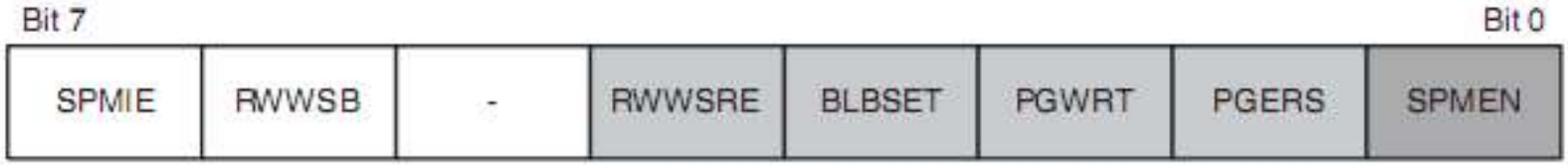
Figure 3. The SPMCR Register



- SPMCR (SPMCSR) regiszter
- SPM (store program memory)
- Z regiszter
- R1:R0 regiszterek

Page Erase

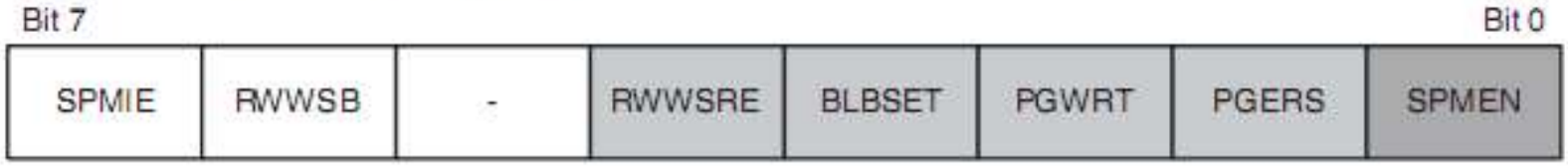
Figure 3. The SPMCR Register



- Page műveletek
- Írás előtt, törlés
- Z – regiszter page címzés (alsó bitek nem fontosak)
- PGRES=1 SPMEN=1, majd SPM (négy órajel)

Loading Page Buffer

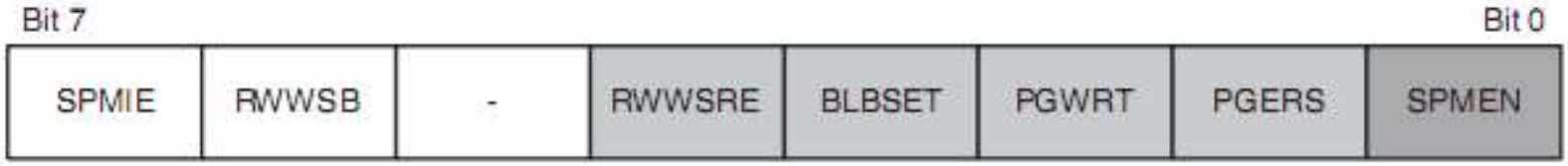
Figure 3. The SPMCR Register



- PageBuffert töltünk fel szavanként. amit a Z regiszterrel címzünk meg
- A word tartalmát az **R1:R0** regiszterbe írjuk
- **Z** regisztert a megfelelő wordre állítjuk (legalsó és felsőbb bitek nem)
- **SPMEN =1**
- **SPM** után négy órajel ciklus

Page Write

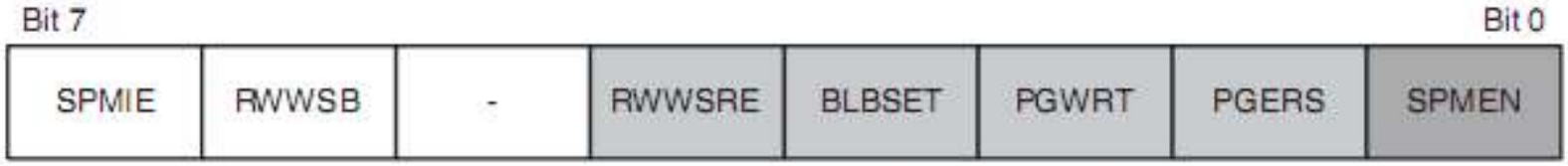
Figure 3. The SPMCR Register



- PageBuffert kész, betöltjük.
- **Z** regiszter beállítása (page címzés)
- **PGWRT=1** és **SPMEN =1**
- **SPM**

SPMCR

Figure 3. The SPMCR Register

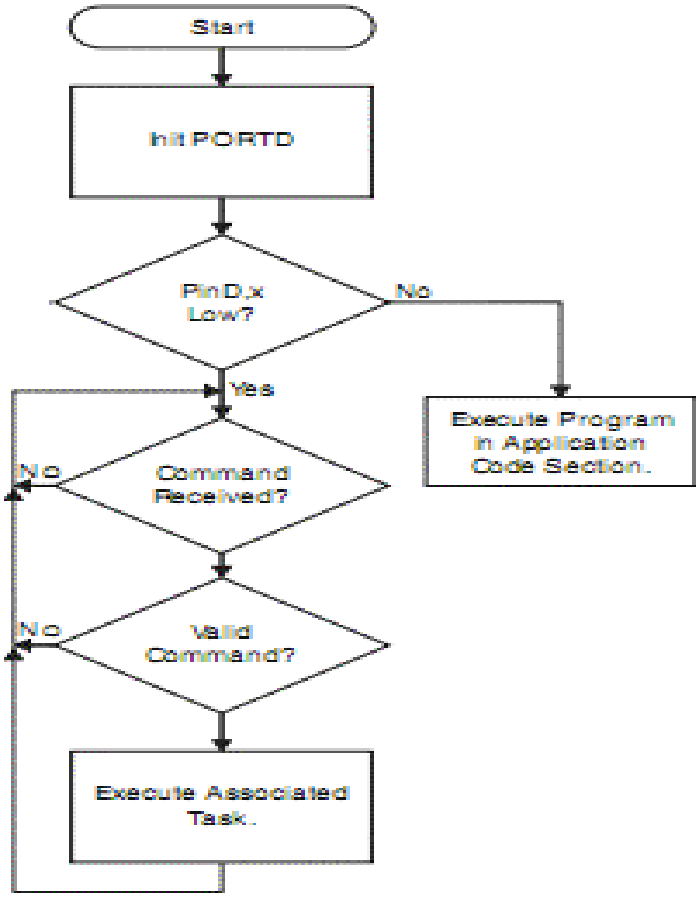


- SPMEN – feladat kész?
- SPMIE – interrupt
  - Interrupt vektorok RWW->NRWW
- RWWSB
  - RWW busy (Hardveres)
  - Nullázás (Szoftveres) – RWWSRE =1, SPMEN=1; SPM
- BLBSET – Lock bits
- EEPROM, Bootloader konfliktus

Bootloader program

- Avr/boot.h
- AVR109 példa
- AVR 911 Open Source Progammer

	Host Writes		Host Reads	
	ID	Data	Data	
Enter Programming Mode	"P"			13d
Auto Increment Address	"a"		dd	
Set Address	"A"	ah al		13d
Write Program Memory, Low Byte	"c"	dd		13d
Write Program Memory, High Byte	"C"	dd		13d
Issue Page Write	"m"			13d
Read Lock Bits	"r"		dd	
Read Program Memory	"R"		2*dd	
Read Data Memory	"d"		dd	
Write Data Memory	"D"	dd		13d
Chip Erase	"e"			13d



Bootloader Kérdések

Bootloader_section? (avr/boot.h)

- Makró, az alkalmazásunk egyes funkciói az NRWW szekcióban tárolódjanak

Bootloader install?

- Makefile: LDFLAGS += -Wl,--section-start=.text=0x3000
- Erase Device,Program FLASH,Verify FLASH,Program Fuses,Verify Fuses,Program lock bits, Verify lock bits

Reset után: bootloader vagy application?

- A BOOTRST fuse

Bootloaderből milyen feltétellel induljon az alkalmazásunk?

- egy gomb, kommunikációs csatornán vett megfelelő üzenet, EEPROM

Hogyan adódik át a vezérlés a bootloadertől?

- asm("jmp 0000")
- Vezérlés átadás: registerek nem üresek

Alkalmazás frissítése sikeres volt –e?

- Beírt program visszaküldése beírás után
- A program kétszer elküldeni egymás után és a második esetben összehasonlítást végezni.
- CRC kód
- EEPROM –ban értékek állítása a programozás elején, végén

Összefoglalás, további célok

- Jelenleg:
  Usart alapú bootloader
  PC oldali vezérlés OpenSourceProgrammer
  Rádiós függvények, RSSI
- Cél:
  Rádiós alapú bootloader
  PC oldali saját szoftver, adatbázissal, MitMót-tal.
  Több különböző programú szenzor

Köszönöm a figyelmet