Interrupts

ATxmega128A1

Networks and Embedded Software

Module 4.3.2

by Wolfgang Neff

Interrupts (1)

- Implementation
  - Enabling Interrupts Globally
    - Set bit in CPU status register
    - Built-in interrupt enabling functions
      - `sei(); // set the global interrupt flag`
      - `cli(); // clear the global interrupt flag`
  - Interrupt Priority Levels
    - Three priority levels
      - High, Medium, Low

Interrupts (2)

- Implementation (continued)
  - Interrupt Priority Levels (continued)
    - Handled by PMIC
      - Programmable Multi-Level Interrupt Controller
    - Each level must be enabled separately
  - Declarations of Interrupt Service Routines
    - ISR(*VECTOR*) { … }

Interrupts (3)

- Register Description
  - SREG: Status Register
    - Bit 7 – I: Global Interrupt Enable
  - PMIC.CTRL: PMIC Control Register
    - Bit 2 – HILVLEN: High-Level Interrupt Enable
      - `PMIC_HILVLEN_bm`
    - Bit 1 – MEDLVLEN: Medium-Level Interrupt Enable
      - `PMIC_MEDLVLEN_bm`
    - Bit 0 – LOLVLEN: Low-Level Interrupt Enable
      - `PMIC_LOLVLEN_bm`

Interrupts (4)

- Example: USART Receive Complete Interrupt
  - Implement interrupt service routine
    - `ISR(USART_RXC_vect) { … }`
  - Enable device interrupt
    - `USART.CTRLA |= USART_RXCINTLVL_LO_gc;`
  - Enable interrupt level
    - `PMIC.CTRL |= PMIC_LOLVLEN_bm;`
  - Enable global interrupts
    - `CPU_SREG |= CPU_I_bm;`

Interrupts (5)

- SREG Register Summary

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SREG	**I**	T	H	S	V	N	Z	C

Interrupts (6)

- PMIC Register Summary

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	NMIEX	-	-	-	-	HILVLEX	MEDLVLEX	LOLVLEX
INTPRI	INTPRI[7:0]							
CTRL	RREN	IVSEL	-	-	-	**HILVLEN**	**MEDLVLEN**	**LOLVLEN**