Interrupts

Please do the following exercises individually.

Timer Control

The timer of an ATxmegaA microcontroller is able to produce the following interrupts:

- OVF_vect: Timer overflow interrupt
- ERR_vect: Timer error interrupt
- CCA_vect: Timer compare or capture channel A interrupt
- CCB_vect: Timer compare or capture channel B interrupt
- CCC_vect: Timer compare or capture channel C interrupt
- CCD_vect: Timer compare or capture channel D interrupt

Its interrupts are configured by two control registers:

Bit	7	6	5	4	3	2	1	0
INTCTRLA	-	-	-	-	ERRINTLVL		OVINTLVL	
INTCTRLB	CCDINTLVL		CCCINTLVL		CCBINTLVL		CCAINTLVL	

Please configure the interrupt as follows[1]:

- A timer overflow produces a low level interrupt
- The timer error interrupt is explicitly turned off
- A capture on channel B produces a high level interrupt

Specify the necessary interrupt service routines and enable these interrupts.

[1] You might need the following constants: `TC0_OVFINTLVL_gm`, `TC0_ERRINTLVL_gm`, `TC0_CCBINTLVL_gm`, `TC_OVFINTLVL_LO_gc`, `TC_ERRINTLVL_OFF_gc`, `TC_CCBINTLVL_MED_gc`, `PMIC_LOLVLEN_bm`, `PMIC_MEDLVLEN_bm`, `CPU_I_bm`.

Interrupts

Please do the following exercises individually.

Timer Control

The timer of an ATxmegaA microcontroller is able to produce the following interrupts:

- OVF_vect: Timer overflow interrupt
- ERR_vect: Timer error interrupt
- CCA_vect: Timer compare or capture channel A interrupt
- CCB_vect: Timer compare or capture channel B interrupt
- CCC_vect: Timer compare or capture channel C interrupt
- CCD_vect: Timer compare or capture channel D interrupt

Its interrupts are configured by two control registers:

Bit	7	6	5	4	3	2	1	0
INTCTRLA	-	-	-	-	ERRINTLVL		OVINTLVL	
INTCTRLB	CCDINTLVL		CCCINTLVL		CCBINTLVL		CCAINTLVL	

Please configure the interrupt as follows[1]:

- A timer overflow produces a low level interrupt
- The timer error interrupt is explicitly turned off
- A capture on channel B produces a high level interrupt

Specify the necessary interrupt service routines and enable these interrupts.

Turn timer overflow interrupt on

```
TIMER.INTCTRLA = (TIMER.INTCTRLA & ~ TC0_OVFINTLVL_gm) | TC_OVFINTLVL_LO_gc;
```

Turn timer error interrupt off

```
TIMER.INTCTRLA = (TIMER.INTCTRLA & ~ TC0_ERRINTLVL_gm) | TC_ERRINTLVL_OFF_gc;
```

Turn timer capture interrupt on

```
TIMER.INTCTRLB = (TIMER.INTCTRLB & ~ TC0_CCBINTLVL_gm) | TC_CCBINTLVL_HI_gc;
```

Interrupt service routines

```
ISR(OVF_vect) { … }
ISR(CCB_vect) { … }
```

Enable interrupts

```
PMIC.CTRL |= (PMIC_HILVLEN_bm | PMIC_LOLVLEN_bm);
CPU_SREG |= CPU_I_bm;
```

[1] You might need the following constants: `TC0_OVFINTLVL_gm`, `TC0_ERRINTLVL_gm`, `TC0_CCBINTLVL_gm`, `TC_OVFINTLVL_LO_gc`, `TC_ERRINTLVL_OFF_gc`, `TC_CCBINTLVL_MED_gc`, `PMIC_LOLVLEN_bm`, `PMIC_MEDLVLEN_bm`, `CPU_I_bm`.