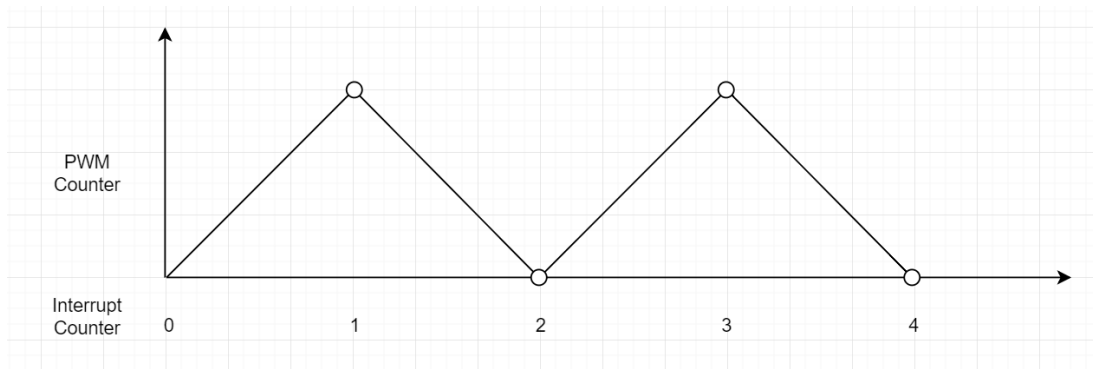


How to distinguish overflow & underflow interrupt in MCB mode of EMIOS?

Principle

Set a count in EMIOS interrupt, count plus one every time. Just keep sure that the first interrupt is overflow interrupt, then when every next overflow interrupt is triggered, the count is odd, every next underflow interrupt is triggered, the count is even.



```
static uint8_t s_interruptCount = 0;

/* Clear flag */
EMIOS_DRV_ClrFlagState(INST_EMIOS_MC0, EMIOS_MC0_CHANNEL23);
++s_interruptCount;
/* overflow */
if (s_interruptCount & 0x01)
{
    SIU->GPDO[130] = 1;
    SIU->GPDO[130] = 0;
    return;
}

/* underflow */
SIU->GPDO[132] = 1;
SIU->GPDO[132] = 0;
```

Caution

1. Before starting EMIOS MC counter, keep sure that EMIOS MC interrupt and global interrupt are enabled. After starting EMIOS MC counter, keep sure that EMIOS MC interrupt and global interrupt are never disabled. Then the first interrupt must be overflow interrupt, the count must be odd, when every underflow interrupt is triggered, the count must be even.

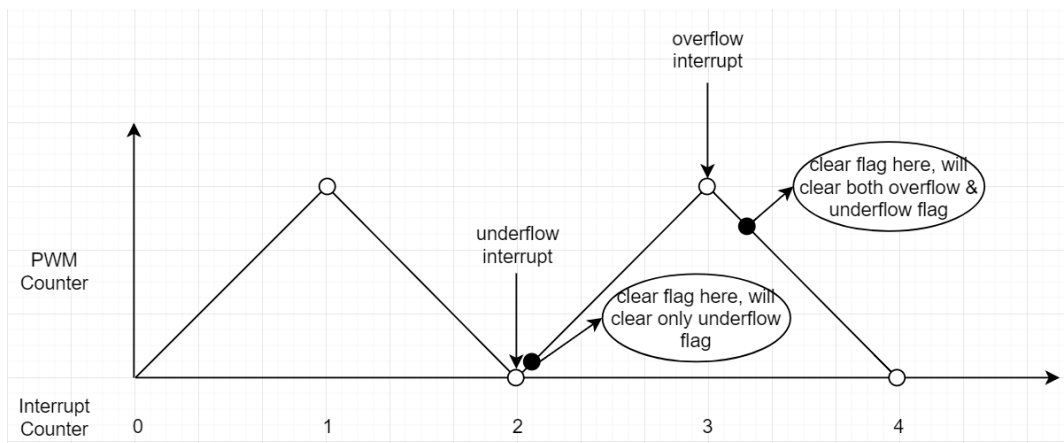
```

/* Disable system global interrupt */
INT_SYS_DisableIRQGlobal();
/* Set interrupt priority. */
EMIOS_DRV_ChannelEnableInterruptRequest(INST_EMIOS_MC0, EMIOS_MC0_CHANNEL23);
INT_SYS_EnableIRQ(EMIOS0_CH23_IRQn);
INT_SYS_SetPriority(EMIOS0_CH23_IRQn, FEATURE_INTERRUPT_PRIO_MAX - 1U);
/* Enable system global interrupt */
INT_SYS_EnableIRQGlobal();

/* Enable eMIOS Global 1 */
EMIOS_DRV_EnableGlobalEmios(INST_EMIOS_MC0);

```

- In EMIOS interrupt service routine, the interrupt flag must be clear at the forefront.



```
static uint8_t s_interruptCount = 0;
```

```

/* Clear flag */
EMIOS_DRV_ClrFlagState(INST_EMIOS_MC0, EMIOS_MC0_CHANNEL23);
++s_interruptCount;
/* overflow */
if (s_interruptCount & 0x01)
{
    SIU->GPDO[130] = 1;
    SIU->GPDO[130] = 0;
    return;
}

/* underflow */
SIU->GPDO[132] = 1;
SIU->GPDO[132] = 0;

```

Test

From the below figure, the overflow & underflow interrupt are distinguished clearly.

