TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – GJ (박현우)
uc820@naver.com

목차

2. Cortex-R5F Hercules Safety MCU – RTI

1) RTI 설정

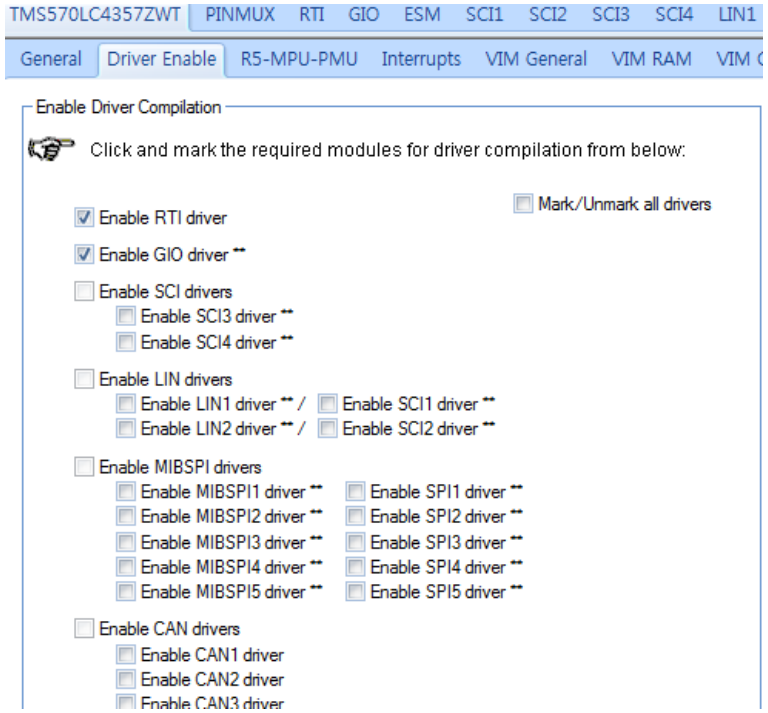
2) rtiInit()

3) MCU 내부 LED 제어

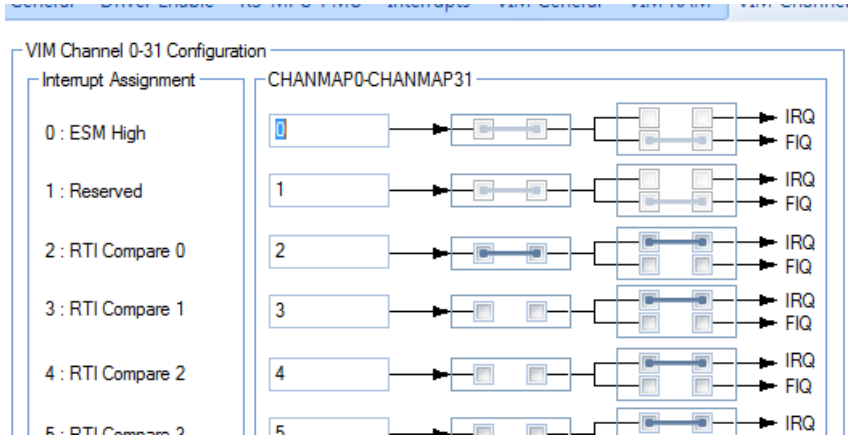
3. 수학 - 미분 / 각속도

2. Cortex-R5F Hercules Safety MCU - RTI (RTI 설정)

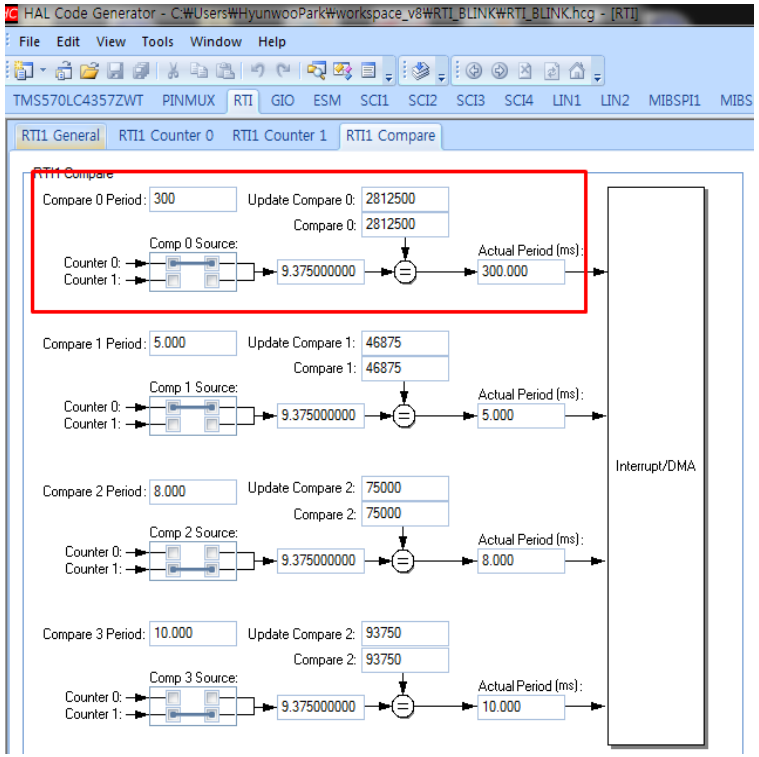
<1>



<2>



<3>



2. Cortex-R5F Hercules Safety MCU – RTI (rtiInit 분석 1)

```
void rtiInit(void)
{
/* USER CODE BEGIN (2) */
/* USER CODE END */
    /** @b Initialize @b RTI1: */

    /** - Setup NTU source, debug options and disable both counter blocks */
    rtiREG1->GCTRL = (uint32)((uint32)0x5U << 16U) | 0x00000000U;

    /** - Setup timebase for free running counter 0 */
    rtiREG1->TBCTRL = 0x00000000U;

    /** - Enable/Disable capture event sources for both counter blocks */
    rtiREG1->CAPCTRL = 0U | 0U;

    /** - Setup input source compare 0-3 */
    rtiREG1->COMPCTRL = 0x00001000U | 0x00000100U | 0x00000000U | 0x00000000U;

    /** - Reset up counter 0 */
    rtiREG1->CNT[0U].UCx = 0x00000000U;

    /** - Reset free running counter 0 */
    rtiREG1->CNT[0U].FRCx = 0x00000000U;

    /** - Setup up counter 0 compare value
    *     - 0x00000000: Divide by 2^32
    *     - 0x00000001-0xFFFFFFFF: Divide by (CPUC0 + 1)
    */
    rtiREG1->CNT[0U].CPUCx = 7U;

    /** - Reset up counter 1 */
    rtiREG1->CNT[1U].UCx = 0x00000000U;

    /** - Reset free running counter 1 */
    rtiREG1->CNT[1U].FRCx  = 0x00000000U;
```

16, 18 bit set 1: 5h NTU1 외부 타임 베이스로 사용될 NTU 입력 신호를 결정하는 비트이다.

나머지 set 0 : Counter block 0, 1 disabled
debug 옵션 끔
0000 0000 0000 0101 0000 0000 0000 0000

17.3.1 RTI Global Control Register (RTIGCTRL)

The global control register starts/stops the counters and selects the signal compared with the timebase control circuit. This register is shown in Figure 17-12 and described in Table 17-2.

Figure 17-12. RTI Global Control Register (RTIGCTRL) [offset = 00]

31 … 20	19 … 16
Reserved	NTUSEL
R-0	R/WP-0

15	14 … 2	1	0
COS	Reserved	CNT1EN	CNT0EN
R/WP-0	R-0	R/WP-0	R/WP-0

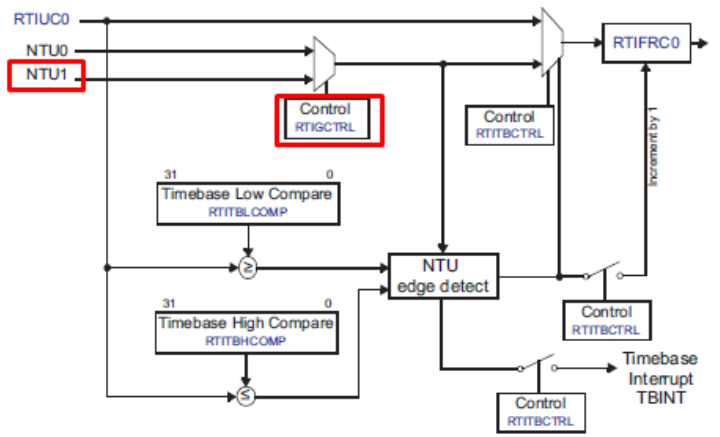
LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 17-2. RTI Global Control Register (RTIGCTRL) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reads return 0. Writes have no effect.
19-16	NTUSEL		Select NTU signal. These bits determine which NTU input signal is used as external timebase
		0h	NTU0
		5h	NTU1
		Ah	NTU2
		Fh	NTU3
		All other values	Tied to 0
15	COS		Continue on suspend. This bit determines if both counters are stopped when the device goes into halting debug mode or if they continue counting.
		0	Counters are stopped while in halting debug mode.
		1	Counters are running while in halting debug mode.
14-2	Reserved	0	Reads return 0. Writes have no effect.
1	CNT1EN		Counter 1 enable. This bit starts and stops counter block 1 (RTIUC1 and RTIFRC1).
		0	Counter block 1 is stopped.
		1	Counter block 1 is running.
0	CNT0EN		Counter 0 enable. This bit starts and stops counter block 0 (RTIUC0 and RTIFRC0).
		0	Counter block 0 is stopped.
		1	Counter block 0 is running.

NOTE: If the application uses the timebase circuit for synchronization between the communications controller and the operating system and the device enters halting debug mode, the synchronization may be lost depending on the COS setting in the RTI module and the halting debug mode behavior of the communications controller.

Figure 17-4. Timebase Control



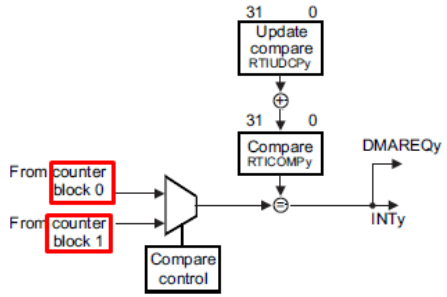


Figure 6-21. Compare Block Diagram

2. Cortex-R5F Hercules Safety MCU – RTI (rtiInit 분석 2)

```
void rtiInit(void)
{
/* USER CODE BEGIN (2) */
/* USER CODE END */
    /** @b Initialize @b RTI1: */

    /** - Setup NTU source, debug options and disable both counter blocks */
    rtiREG1->GCTRL = (uint32)((uint32)0x5U << 16U) | 0x00000000U;

    /** - Setup timebase for free running counter 0 */
    rtiREG1->TBCTRL = 0x00000000U;

    /** - Enable/Disable capture event sources for both counter blocks */
    rtiREG1->CAPCTRL = 0U | 0U;

    /** - Setup input source compare 0-3 */
    rtiREG1->COMPCTRL = 0x00001000U | 0x00000100U | 0x00000000U | 0x00000000U;

    /** - Reset up counter 0 */
    rtiREG1->CNT[0U].UCx = 0x00000000U;

    /** - Reset free running counter 0 */
    rtiREG1->CNT[0U].FRCx = 0x00000000U;

    /** - Setup up counter 0 compare value
    *     - 0x00000000: Divide by 2^32
    *     - 0x00000001-0xFFFFFFFF: Divide by (CPUC0 + 1)
    */
    rtiREG1->CNT[0U].CPUCx = 7U;

    /** - Reset up counter 1 */
    rtiREG1->CNT[1U].UCx = 0x00000000U;

    /** - Reset free running counter 1 */
    rtiREG1->CNT[1U].FRCx  = 0x00000000U;

    /** - Setup up counter 1 compare value
```

1 bit set 0 : failing clock을 가진 NTU1을 탐지하면 자동적으로 free running count 0 값을 증가.

0 bit set 0 : RTIUC0는 RTIFR0를 측정한다.
0000 0000 0000 0101 0000 0000 0000 0000

17.3.2 RTI Timebase Control Register (RTITBCTRL)

The timebase control register selects if the free running counter 0 is incremented by RTICLK or NTU. This register is shown in Figure 17-13 and described in Table 17-3.

Figure 17-13. RTI Timebase Control Register (RTITBCTRL) [offset = 04h]

31 – 8	7 – 2	1	0
Reserved	Reserved	INC	TBEXT
R-0	R-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

Table 17-3. RTI Timebase Control Register (RTITBCTRL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reads return 0. Writes have no effect.
1	INC		Increment free running counter 0. This bit determines whether the free running counter 0 (RTIFRC0) is automatically incremented if a failing clock on the NTU signal is detected.
		0	RTIFRC0 will not be incremented on a failing external clock.
		1	RTIFRC0 will be incremented on a failing external clock.
0	TBEXT		Timebase external. This bit selects whether the free running counter 0 (RTIFRC0) is clocked by the internal up counter 0 (RTIUC0) or from the external signal NTU. Setting the TBEXT bit from 0 to 1 will not increment RTIFRC0, since RTIUC0 is reset. When the timebase supervisor circuit detects a missing clock edge, then the TBEXT bit is reset. Only the software can select whether the external signal should be used.
		0	RTIUC0 clocks RTIFRC0.
		1	NTU clocks RTIFRC0.

- If TBEXT = 0, the compare value is updated.
- If TBEXT = 1, the compare value is unchanged.

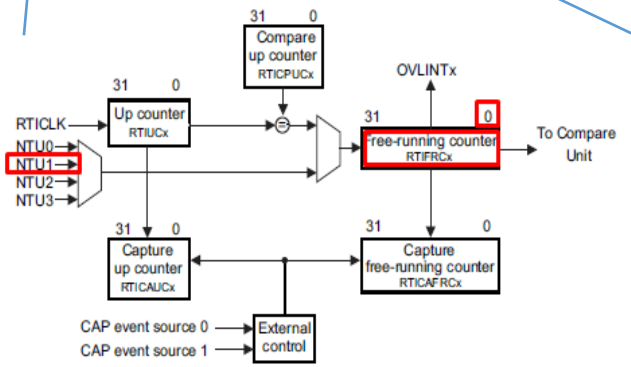


Figure 6-20. Counter Block Diagram

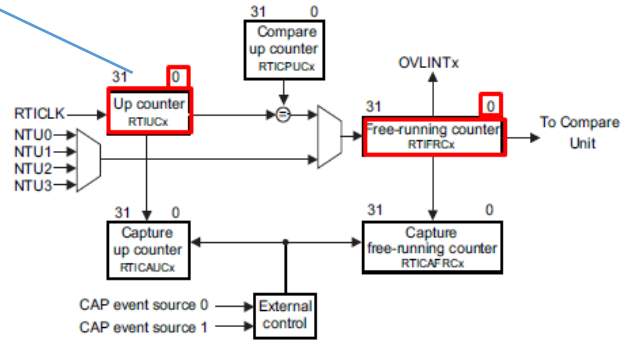


Figure 6-20. Counter Block Diagram

2. Cortex-R5F Hercules Safety MCU – RTI (rtiInit 분석 3)

```
/** - Setup NTU source, debug options and disable both counter blocks */
rtiREG1->GCTRL = (uint32)((uint32)0x5U << 16U) | 0x00000000U;

/** - Setup timebase for free running counter 0 */
rtiREG1->TBCTRL = 0x00000000U;

/** - Enable/Disable capture event sources for both counter blocks */
rtiREG1->CAPCTRL = 0U | 0U;

/** - Setup input source compare 0-3 */
rtiREG1->COMPCTRL = 0x00001000U | 0x00000100U | 0x00000000U | 0x00000000U;

/** - Reset up counter 0 */
rtiREG1->CNT[0U].UCx = 0x00000000U;

/** - Reset free running counter 0 */
rtiREG1->CNT[0U].FRCx = 0x00000000U;

/** - Setup up counter 0 compare value
*     - 0x00000000: Divide by 2^32
*     - 0x00000001-0xFFFFFFFF: Divide by (CPUC0 + 1)
*/
rtiREG1->CNT[0U].CPUCx = 7U;
```

0 bit set 0 : RTICOMP0과 RTIFRC0을 비교
4 bit set 0 : TRICOMP1과 RTIFRC0을 비교
8 bit set 1 : RTICOMP2와 RTIFRC1을 비교
12 bit set 1 : RTICOMP3과 RTIFRC1을 비교

0000 0000 0000 0101 0000 0000 0000 0000

17.3.4 RTI Compare Control Register (RTICOMPCTRL)

The compare control register controls the source for the compare registers. This register is shown in Figure 17-15 and described in Table 17-5.

Figure 17-15. RTI Compare Control Register (RTICOMPCTRL) [offset = 0Ch]

31–16	15–13	12	11–9	8	7–5	4	3–1	0
Reserved	Reserved	COMPSEL3	Reserved	COMPSEL2	Reserved	COMPSEL1	Reserved	COMPSEL0
R-0	R-0	R/WP-0	R-0	R/WP-0	R-0	R/WP-0	R-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -n = value after reset

Table 17-5. RTI Compare Control Register (RTICOMPCTRL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reads return 0. Writes have no effect.
12	COMPSEL3		Compare select 3. This bit determines the counter with which the compare value held in compare register 3 (RTICOMP3) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
11-9	Reserved	0	Reads return 0. Writes have no effect.
8	COMPSEL2		Compare select 2. This bit determines the counter with which the compare value held in compare register 2 (RTICOMP2) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
7-5	Reserved	0	Reads return 0. Writes have no effect.
4	COMPSEL1		Compare select 1. This bit determines the counter with which the compare value held in compare register 1 (RTICOMP1) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
3-1	Reserved	0	Reads return 0. Writes have no effect.
0	COMPSEL0		Compare select 0. This bit determines the counter with which the compare value held in compare register 0 (RTICOMP0) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.

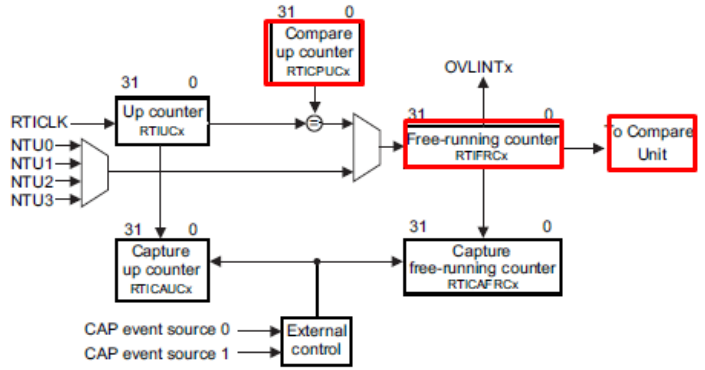
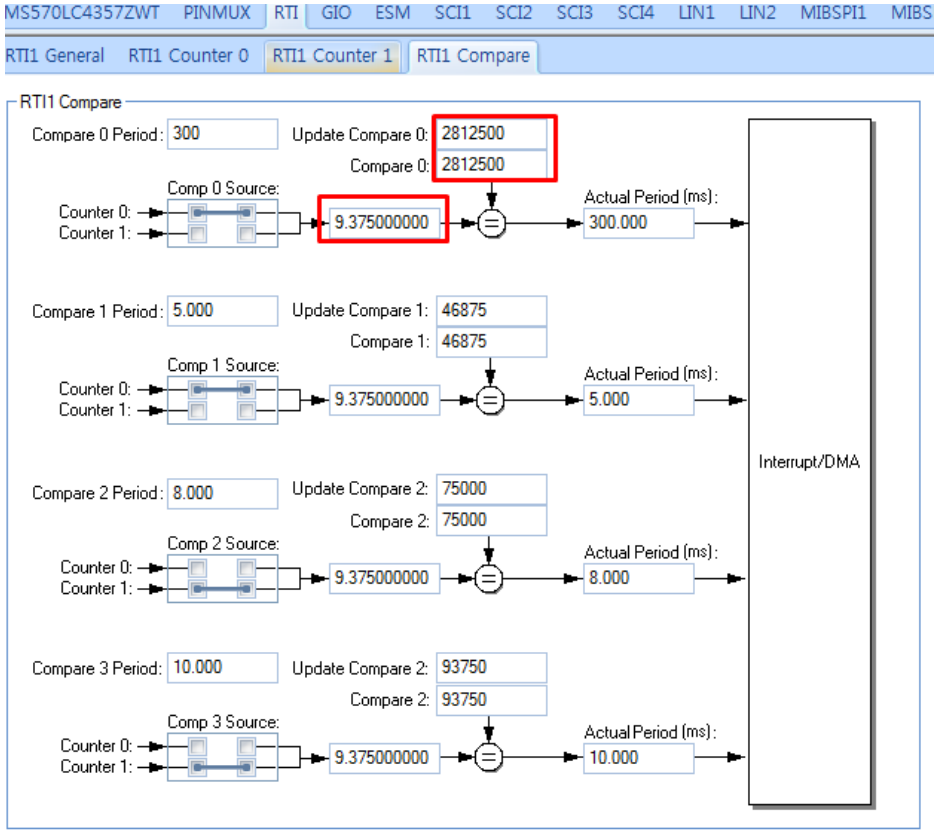


Figure 6-20. Counter Block Diagram

2. Cortex-R5F Hercules Safety MCU – RTI (rtiInit 분석 4)



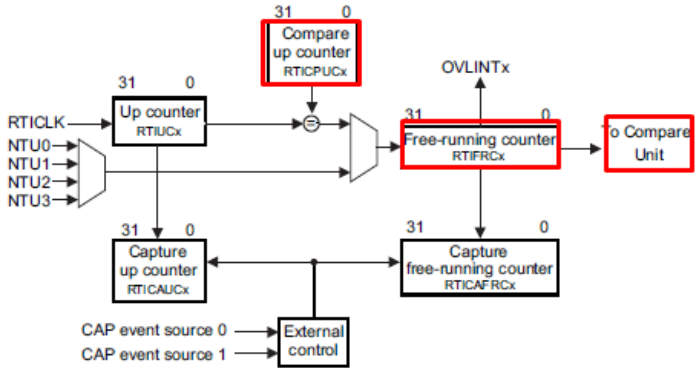
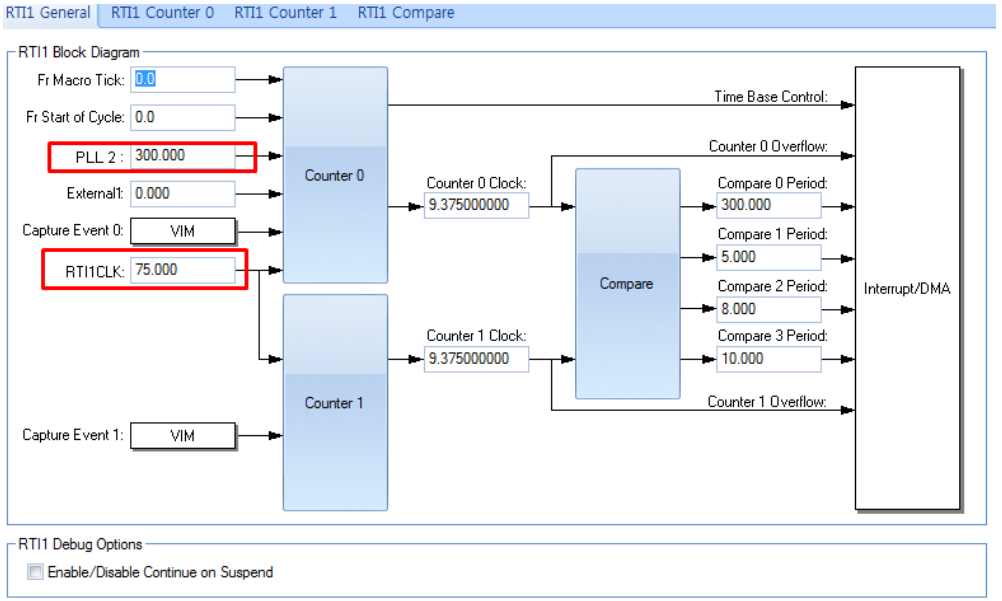


Figure 6-20. Counter Block Diagram

9.375 Mhz = 106 ns

2812500 * 106 ns = 300ms

PLL값에 따라 RTICLK값이 달라지니, 설정을 잘 고려 해야 한다.



2. Cortex-R5F Hercules Safety MCU - RTI (rtiInit 분석 5)

```
rtiREG1->CAPCTRL = 0U | 0U;

/** - Setup input source compare 0-3 */
rtiREG1->COMPCTRL = 0x00001000U | 0x00000100U | 0x00000000U | 0x00000000U;

/** - Reset up counter 0 */
rtiREG1->CNT[0U].UCx = 0x00000000U;

/** - Reset free running counter 0 */
rtiREG1->CNT[0U].FRCx = 0x00000000U;

/** - Setup up counter 0 compare value
*     - 0x00000000: Divide by 2^32
*     - 0x00000001-0xFFFFFFFF: Divide by (CPUC0 + 1)
*/
rtiREG1->CNT[0U].CPUCx = 7U;

/** - Reset up counter 1 */
rtiREG1->CNT[1U].UCx = 0x00000000U;

/** - Reset free running counter 1 */
rtiREG1->CNT[1U].FRCx  = 0x00000000U;

/** - Setup up counter 1 compare value
*     - 0x00000000: Divide by 2^32
*     - 0x00000001-0xFFFFFFFF: Divide by (CPUC1 + 1)
*/
rtiREG1->CNT[1U].CPUCx = 7U;

/** - Setup compare 0 value. This value is compared with selected free running counter. */
rtiREG1->CMP[0U].COMPx = 2812500U;
```

17.3.7 RTI Compare Up Counter 0 Register (RTICPUC0)

The compare up counter 0 register holds the value to be compared with prescale counter 0 (RTIUC0). This register is shown in Figure 17-18 and described in Table 17-8.

Figure 17-18. RTI Compare Up Counter 0 Register (RTICPUC0) [offset = 18h]

31	16
CPUC0	
R/WP-0	

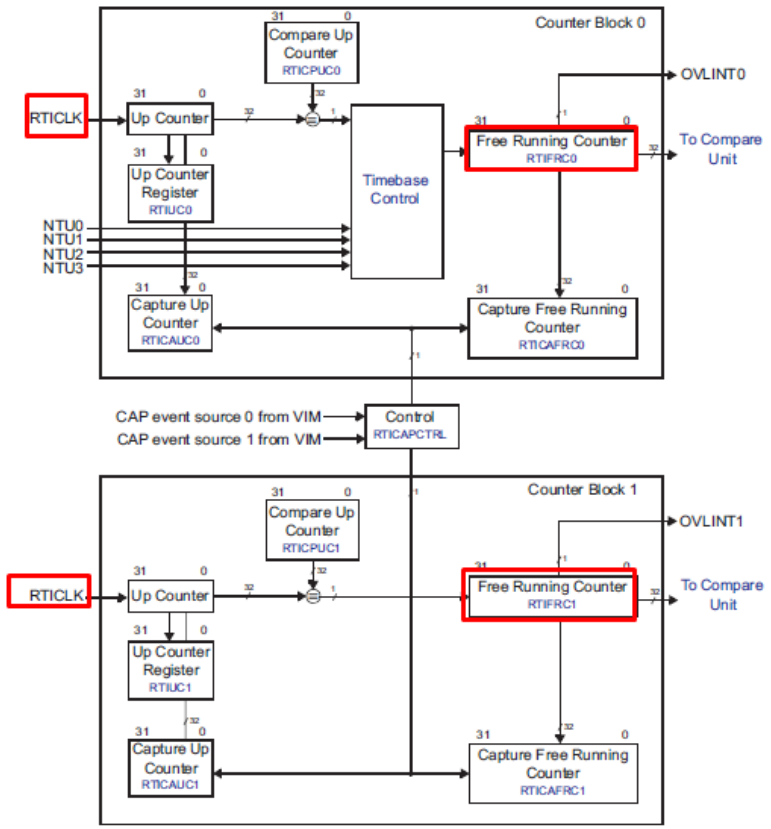
15	0
CPUC0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-8. RTI Compare Up Counter 0 Register (RTICPUC0) Field Descriptions

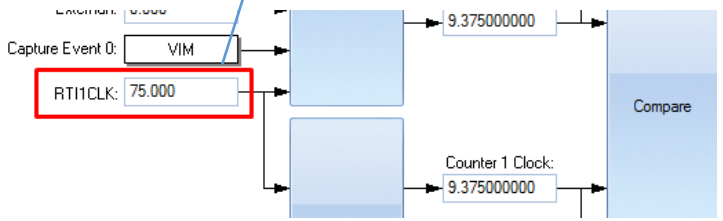
Bit	Field	Value	Description
31-0	CPUC0	0-FFFF FFFFh	Compare up counter 0. This register holds the value that is compared with the up counter 0. When the compare shows a match, the free running counter 0 (RTIFRC0) is incremented. RTIUC0 is set to 0 when the counter value matches the RTICPUC0 value. The value set in this register prescales the RTI clock. If CPUC0 = 0, then f_{FRC0} = RTICLK/(2^{32}+1) (Setting CPUC0 equal to 0 is not recommended. Doing so will hold the Up Counter at 0 for 2 RTICLK cycles after it overflows from FFFF FFFFh to 0.) If CPUC0 ≠ 0, then f_{FRC0} = RTICLK/(RTICPUC0+1) A read of this register returns the current compare value. A write to this register: • If TBEXT = 0, the compare value is updated. • If TBEXT = 1, the compare value is unchanged.

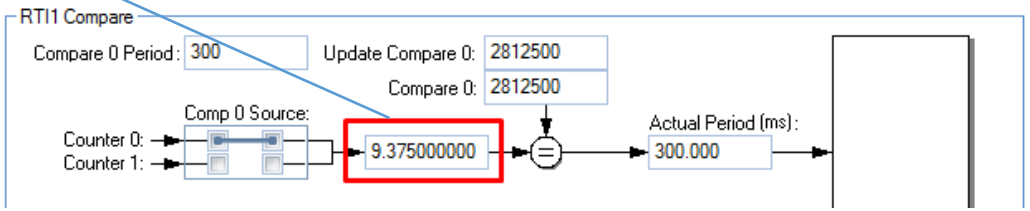
Figure 17-2. Counter Block Diagram



f(FRC0) = RTICLK / 8 , RTICLK = 75Mhz → f = 9.375Mhz

F(FRC1) = RTICLK / 8





2. Cortex-R5F Hercules Safety MCU - RTI (rtiInit 분석 6)

```
32    /** - Setup update compare 3 value. This value is added to the compare 3 value on
33    rtiREG1->CMP[3U].UDCPx = 93750U;
34
35    /** - Clear all pending interrupts */
36    rtiREG1->INTFLAG = 0x0007000FU;
37
38    /** - Disable all interrupts */
39    rtiREG1->CLEARINTENA = 0x00070F0FU;
40
41    /**   @note This function has to be called before the driver can be used.\n
42    *           This function has to be executed in privileged mode.\n
43    *           This function does not start the counters.
44    */
45
46 /* USER CODE BEGIN (3) */
```

16 bit set 1 : timebase 인터럽트 flag 0으로 초기화
17 bit set 1 : FRC0 오버플로우 인터럽트 flag 0
18 bit set 1 : FRC1 오버플로우 인터럽트 flag 1

나머지 bit : 인터럽트 pending bit를 0으로 초기화
0000 0000 0000 0111 0000 0000 0000 0000

17.3.27 RTI Interrupt Flag Register (RTIINTFLAG)

The corresponding flags are set at every compare match of the RTIFRCx and RTICOMPx values, whether the interrupt is enabled or not. This register is shown in Figure 17-38 and described in Table 17-28.

Figure 17-38. RTI Interrupt Flag Register (RTIINTFLAG) [offset = 88h]

31	19	18	17	16
Reserved		OVL1INT	OVL0INT	TBINT
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0

15	4	3	2	1	0
Reserved		INT3	INT2	INT1	INT0
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0	R/W1CP-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -*n* = value after reset

Table 17-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	OVL1INT		Free running counter 1 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
17	OVL0INT		Free running counter 0 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
16	TBINT		Timebase interrupt flag. This flag is set when the TBEXT bit is cleared by detection of a missing external clock edge. It will not be set by clearing TBEXT by software. It determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
15-4	Reserved	0	Reads return 0. Writes have no effect.
3	INT3		Interrupt flag 3. These bits determine if an interrupt due to a Compare 3 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
2	INT2		Interrupt flag 2. These bits determine if an interrupt due to a Compare 2 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
1	INT1		Interrupt flag 1. These bits determine if an interrupt due to a Compare 1 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.

Table 17-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions (continued)

Bit	Field	Value	Description
0	INT0		Interrupt flag 0. These bits determine if an interrupt due to a Compare 0 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.

2. Cortex-R5F Hercules Safety MCU - RTI (rtiInit 분석 7)

```
/** - Setup update compare 3 value. This value is added to the compare 3 \
rtiREG1->CMP[3U].UDCPx = 93750U;

/** - Clear all pending interrupts */
rtiREG1->INTFLAG = 0x0007000FU;

/** - Disable all interrupts */
rtiREG1->CLEARINTENA = 0x00070F0FU;

/**   @note This function has to be called before the driver can be used.\
*           This function has to be executed in privileged mode.\n
*           This function does not start the counters.
*/
```

모든 인터럽트 disable

17.3.26 RTI Clear Interrupt Enable Register (RTICLEARINTENA)

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be disabled. This register is shown in Figure 17-37 and described in Table 17-27.

Figure 17-37. RTI Clear Interrupt Control Register (RTICLEARINTENA) [offset = 84h]

31	24
Reserved	
R-0	

23	19	18	17	16
Reserved		CLEAROVL1INT	CLEAROVL0INT	CLEARTBINT
R-0		R/WP-0	R/WP-0	R/WP-0

15	12	11	10	9	8
Reserved		CLEARDMA3	CLEARDMA2	CLEARDMA1	CLEARDMA0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	4	3	2	1	0
Reserved		CLEARINT3	CLEARINT2	CLEARINT1	CLEARINT0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -n = value after reset

Table 17-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	CLEAROVL1INT		Clear free running counter 1 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
17	CLEAROVL0INT		Clear free running counter 0 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
16	CLEARTBINT		Clear timebase interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	CLEARDMA3		Clear compare DMA request 3.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.
10	CLEARDMA2		Clear compare DMA request 2.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.

Table 17-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions (continued)

Bit	Field	Value	Description
9	CLEARDMA1		Clear compare DMA request 1.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.
8	CLEARDMA0		Clear compare DMA request 0.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.
7-4	Reserved	0	Reads return 0. Writes have no effect.
3	CLEARINT3		Clear compare interrupt 3.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
2	CLEARINT2		Clear compare interrupt 2.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
1	CLEARINT1		Clear compare interrupt 1.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
0	CLEARINT0		Clear compare interrupt 0.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.

2. Cortex-R5F Hercules Safety MCU - RTI (내부 LED 제어 1)

```
74 int main(void)
75 {
76 /* USER CODE BEGIN (3) */
77     rtiInit();
78
79     gioSetDirection(hetPORT1, 0xFFFFFFFF);
80
81     rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
82
83     _enable_IRQ_interrupt_();
84
85     rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);
86 /* USER CODE END */
87
88     return 0;
89 }
90
91 void rtiNotification(rtiBASE_t *rtiREG, uint32 notification){
92     gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ 0x00000001);
93 }
```

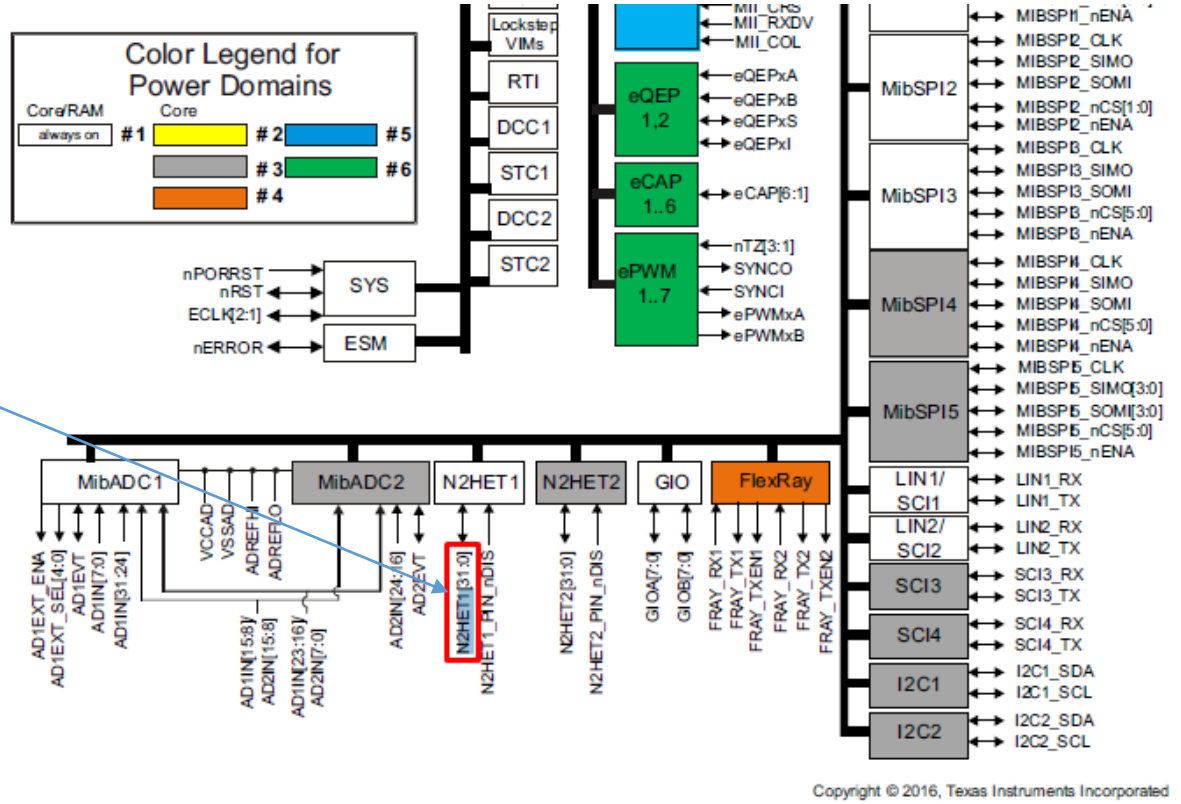
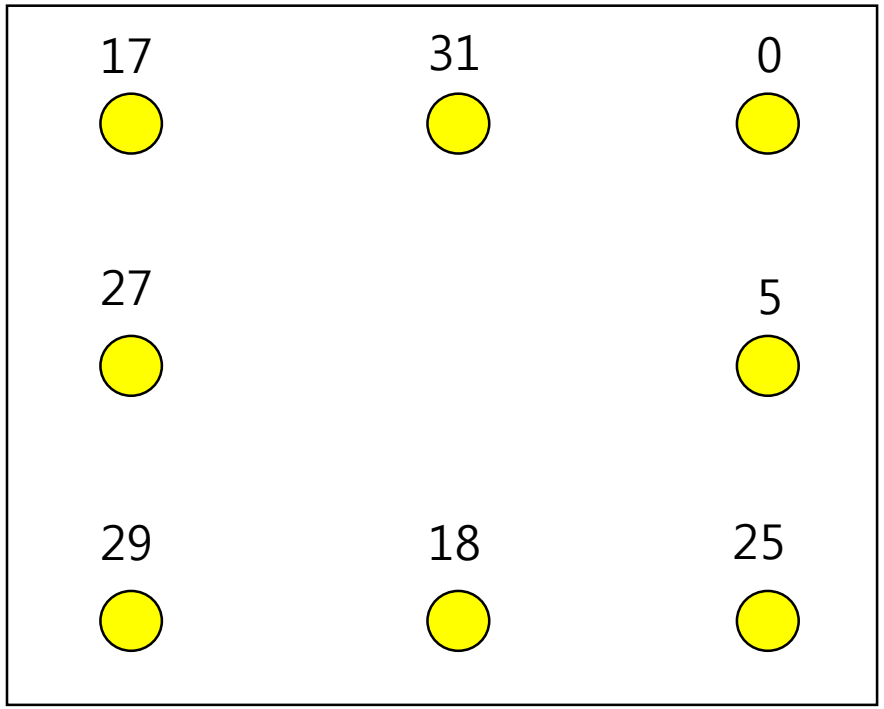


Figure 1-1. Functional Block Diagram

2. Cortex-R5F Hercules Safety MCU - RTI (내부 LED 제어 2)

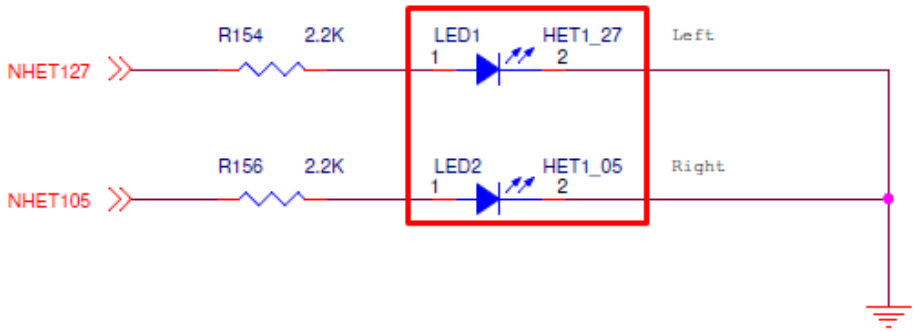


<모두 켜기>

```
gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ 0xAA060011);
```

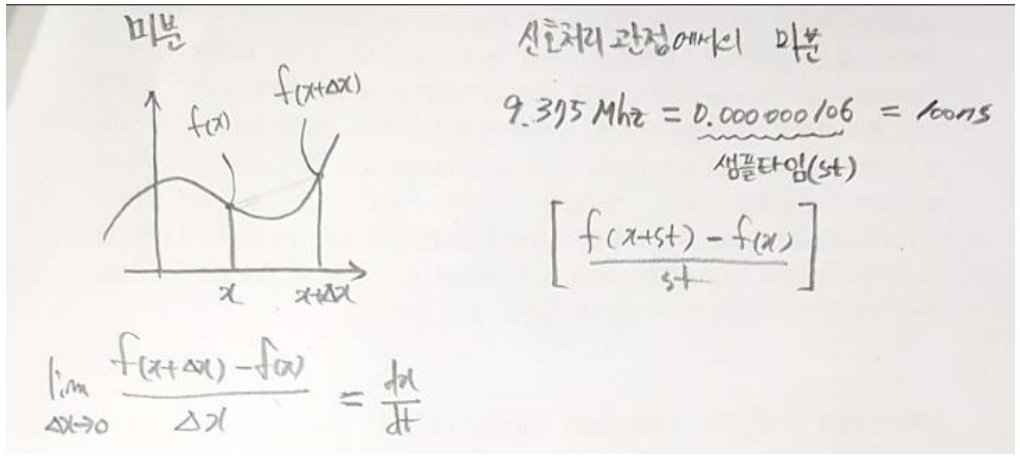
<하나씩 뱀처럼 켜지기>

^0x20000
^0x80000000
^0x00000001
^0x00000010
^0x200000
^0x40000
^0x20000000
^0x8000000
^0xAA060011



3. 수학 – 미분 / 각속도

미분



각속도