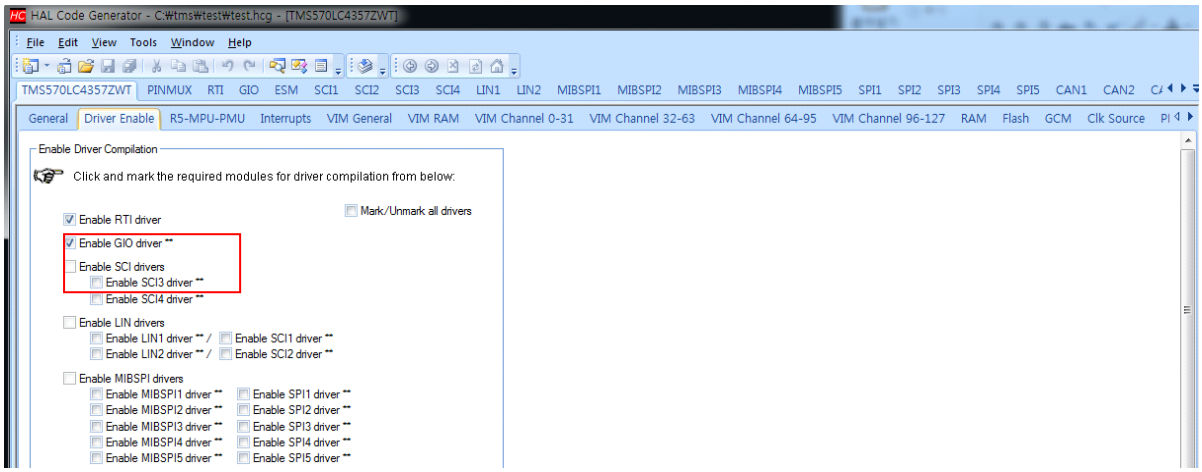
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 : Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 : 황수정
sue100012@naver.com
50일차 (2018. 05. 09)

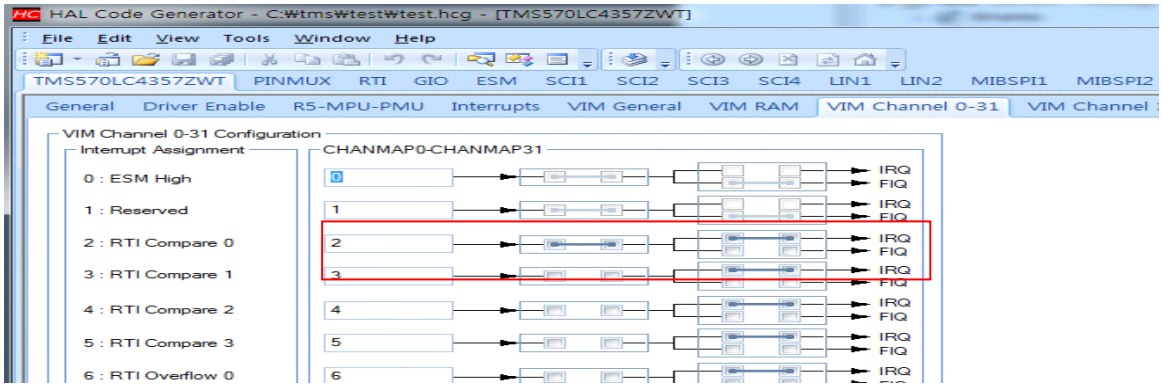
목차

Cortex-R5F – RTI

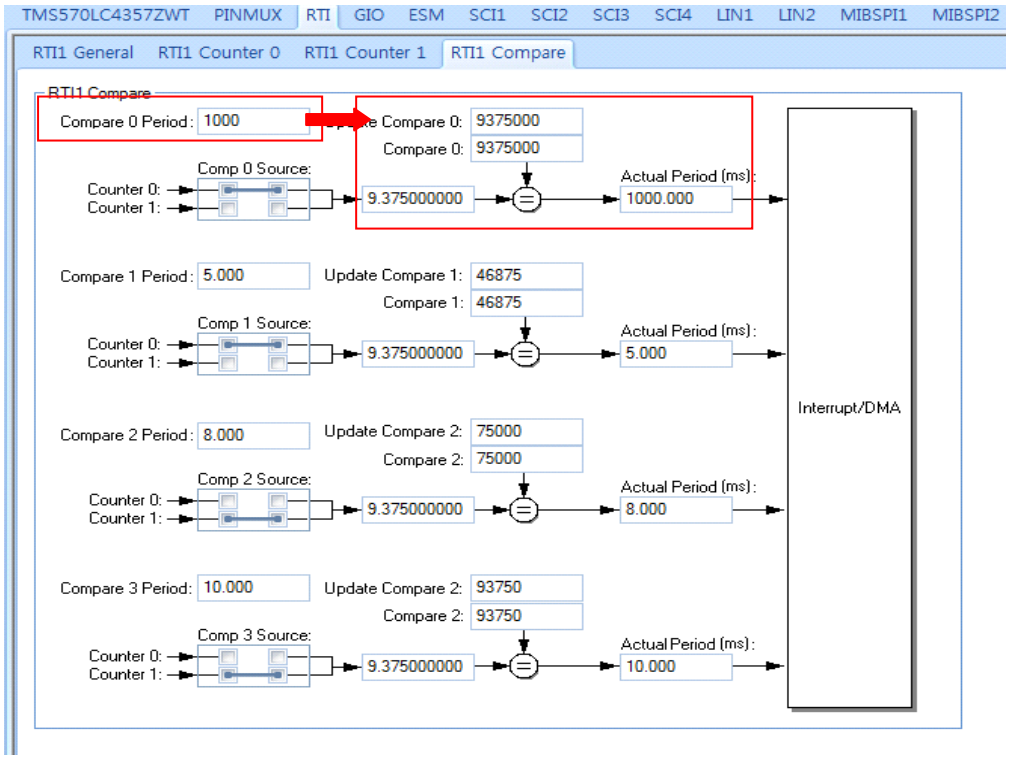
RTI를 사용하기 전에 RTI 환경 설정을 해주어야 한다. HAL에서 RTI, GIO driver를 선택해준다.



VIM channel에서 사용할 곳을 설정해준다.



시간을 설정해준다. 시간을 설정해주면 아래 표시한 값들이 변경되는 것을 확인할 수 있다.



```
#include "HL_sys_common.h"
#include "HL_system.h"

#include "HL_sys_core.h"
#include "HL_mibspi.h"
#include "HL_esm.h"
#include "HL_rti.h"
#include "HL_gio.h"
#include "HL_het.h"

int main(void)
{
      rtiInit();                                          //RTI 리얼 타임 인터럽트를 초기화한다.
      gioSetDirection(hetPORT1, 0xffffffff);
      rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0);
      _enable_IRQ_interrupt_();                           // 인터럽트를 사용 가능하게 해준다.
      rtiStartCounter(rtiREG1, rtiCOUNTER_BLOCK0);

    while(1);

    return 0;
}

void rtiNotification(rtiBASE_t *rtiREG, uint32 notification)
{
    gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ 0x00000001);
}
```

메인 함수의 코드는 위와 같다. 먼저 rtiInit();에 대해 보자.

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

0x5U << 16U는 5를 <<16한 것으로 0x50000한 것과 같다. 18, 16bit를 확인하면 된다. 18, 16 bit가 있는 19 ~ 16을 확인한다.

NTU 신호를

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.

ntu1

17.2.4 Synchronizing Timer Events to Network Time (NTU)

For applications which are participating on a time-triggered communication bus, it is often beneficial to synchronize the application or operating system to the network time. The RTI provides a feature to increment Free Running Counter 0 (RTIFRC0) by a periodic clock provided by the communication module. In this case two different clocks can be chosen. One is the FlexRay module Macrotick (NTU0) and the other is the Start of Cycle (NTU1) information of the same module.

The application has control over which clock (RTICLK, NTU0, NTU1) should be used for clocking RTIFRC0. If NTUx is used, a clock supervision circuit allows to monitor this clock and provides a fallback solution, should the clock be non-functional (missing). A too fast running NTUx cannot be detected.

RTIUC0 is utilized to monitor the NTUx signal. A detection window can be programmed in which a valid NTU clock pulse needs to occur. If no pulse is detected, the RTI automatically switches back to clock the Free Running Counter 0 with RTIUC0. In order to avoid a big jitter in the operating ticks, in case a switch back to RTIUC0 happens, RTICPUC0 should be set to a value so the clock frequency RTIUC0 outputs is approximately the same as the NTUx frequency.

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

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
Reserved	
R-0	

7	2	1	0
Reserved		INC	TBEXT
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.

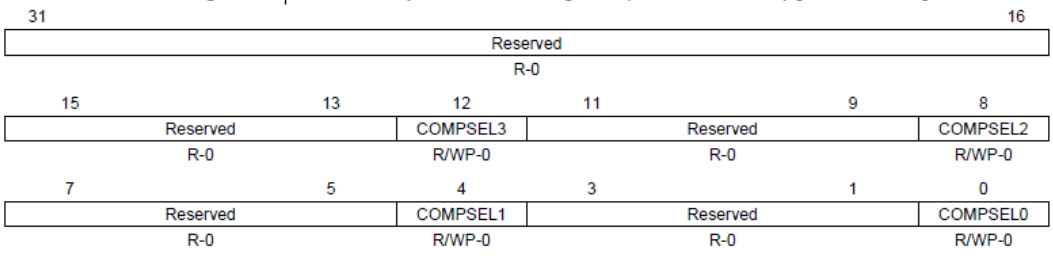
```
/** - Enable/Disable capture event sources for both counter blocks */
rtiREG1->CAPCTRL = 0U | 0U; // 사용하지 않는다.

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

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]



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

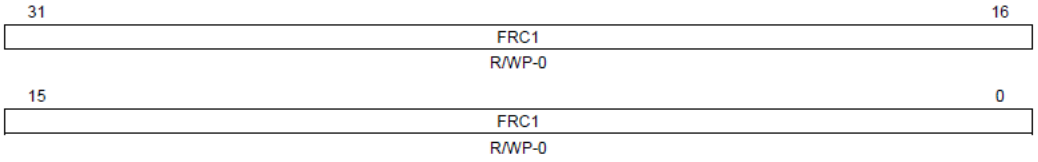
Table 17-5. RTI Compare Control Register (RTICOMPCTRL) Field Descriptions

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.

17.3.10 RTI Free Running Counter 1 Register (RTIFRC1)

The free running counter 1 register holds the current value of the free running counter 1. This register is shown in Figure 17-21 and described in Table 17-11.

Figure 17-21. RTI Free Running Counter 1 Register (RTIFRC1) [offset = 30h]



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

Table 17-11. RTI Free Running Counter 1 Register (RTIFRC1) Field Descriptions

Bit	Field	Value	Description
31-0	FRC1	0-FFFF FFFFh	Free running counter 1. This register holds the current value of the free running counter 1 and will be updated continuously. A read of this register returns the current value of the counter. A write to this register presets the counter. The counter increments then from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC1 and RTIFRC1.**

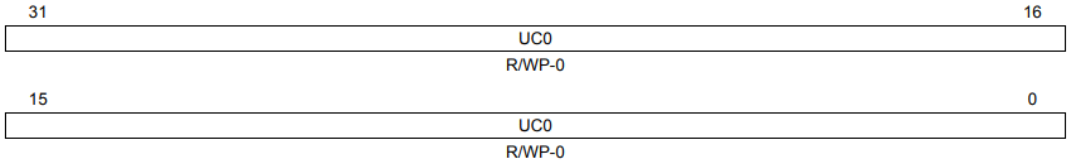
/** - Reset up counter 0 */

```
rtiREG1->CNT[0U].UCx = 0x00000000U;
```

17.3.6 RTI Up Counter 0 Register (RTIUC0)

The up counter 0 register holds the current value of prescale counter. This register is shown in Figure 17-17 and described in Table 17-7.

Figure 17-17. RTI Up Counter 0 Register (RTIUC0) [offset = 14h]



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

Table 17-7. RTI Up Counter 0 Register (RTIUC0) Field Descriptions

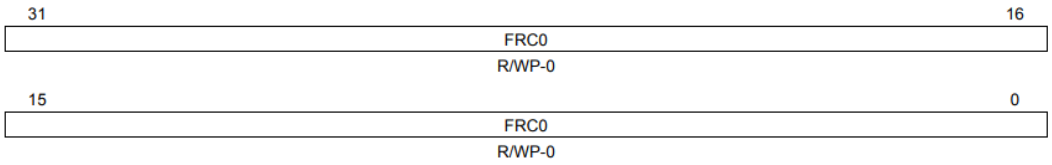
Bit	Field	Value	Description
31-0	UC0	0-FFFF FFFFh	Up counter 0. This register holds the current value of the up counter 0 and prescales the RTI clock. It will be only updated by a previous read of free running counter 0 (RTIFRC0). This method of updating effectively gives a 64-bit read of both counters, without having the problem of a counter being updated between two consecutive reads on up counter 0 (RTIUC0) and free running counter 0 (RTIFRC0). A read of this counter returns the value of the counter at the time RTIFRC0 was read. A write to this counter presets it with a value. The counter then increments from this written value upwards. Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0. **Note: If the preset value is bigger than the compare value stored in register RTICPUC0, then it can take a long time until a compare matches, since RTIUC0 has to count up until it overflows.**

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

17.3.5 RTI Free Running Counter 0 Register (RTIFRC0)

The free running counter 0 register holds the current value of free running counter 0. This register is shown in Figure 17-16 and described in Table 17-6.

Figure 17-16. RTI Free Running Counter 0 Register (RTIFRC0) [offset = 10h]



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

Table 17-6. RTI Free Running Counter 0 Register (RTIFRC0) Field Descriptions

Bit	Field	Value	Description
31-0	FRC0	0-FFFF FFFFh	Free running counter 0. This registers holds the current value of the free running counter 0. A read of this counter returns the current value of the counter. The counter can be preset by writing (in privileged mode only) to this register. The counter increments then from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0.**

```
/** - 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 = 9375000U;

    /** - Setup update compare 0 value. This value is added to the compare 0 value on each compare match. */
    rtiREG1->CMP[0U].UDCPx = 9375000U;

    /** - Setup compare 1 value. This value is compared with selected free running counter. */
    rtiREG1->CMP[1U].COMPx = 46875U;

    /** - Setup update compare 1 value. This value is added to the compare 1 value on each compare match. */
    rtiREG1->CMP[1U].UDCPx = 46875U;

    /** - Setup compare 2 value. This value is compared with selected free running counter. */
    rtiREG1->CMP[2U].COMPx = 75000U;

    /** - Setup update compare 2 value. This value is added to the compare 2 value on each compare match. */
    rtiREG1->CMP[2U].UDCPx = 75000U;

    /** - Setup compare 3 value. This value is compared with selected free running counter. */
    rtiREG1->CMP[3U].COMPx = 93750U;

    /** - Setup update compare 3 value. This value is added to the compare 3 value on each
```

```
compare match. */
    rtiREG1->CMP[3U].UDCPx = 93750U;

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

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.

```
    /** - Disable all interrupts */
    rtiREG1->CLEARINTENA = 0x00070F0FU; //DMA는 현재 사용하지 않는다. 인터럽트를 쓰지 않는다.

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

```
/* USER CODE BEGIN (3) */
/* USER CODE END */
}
//주기가 높은 것을 고성능
void gioSetDirection(gioPORT_t *port, uint32 dir)
{
    port->DIR = dir;
}
```


23.4.18 NHET Direction Register (HETDIR)

N2HET1: offset = FFF7 B84Ch; **N2HET2:** offset = FFF7 B94Ch

Figure 23-73. N2HET Direction Register (HETDIR)

31	16
HETDIR	
R/W-0	

15	0
HETDIR	
R/W-0	

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

Table 23-34. N2HET Direction Register (HETDIR) Field Descriptions

Bit	Field	Value	Description
31-0	HETDIR[n]		Data direction of NHET pins
		0	Pin HET[n] is an input (and its output buffer is tristated).
		1	Pin HET[n] is an output.

NOTE: Table 23-9 shows how the register bits of DIR, PULDIS and PULSEL are affecting the N2HET pins.

핀을 다 OUTPUT으로 설정하겠다. 1로 셋팅한다

```
void rtiEnableNotification(rtiBASE_t *rtiREG, uint32 notification)
{
/* USER CODE BEGIN (38) */
/* USER CODE END */

    rtiREG->INTFLAG = notification;
    rtiREG->SETINTENA    = notification; //컴페어 0번의 인터럽트를 활성화한다.

    /**   @note The function rtiInit has to be called before this function can be used.₩n
    *           This function has to be executed in privileged mode.
    */

/* USER CODE BEGIN (39) */
/* USER CODE END */
}
```