Issue:　HwiHdr is called twice

Reporoduce:

http://software-dl.ti.com/processor-sdk-rtos/esd/docs/latest/rtos/index_how_to_guides.html#rtos-customization-using-an-external-input-to-trigger-an-interrupt-on-am57x

We reproduced with AM574IDK.

I uploaded project that can be reproduced.

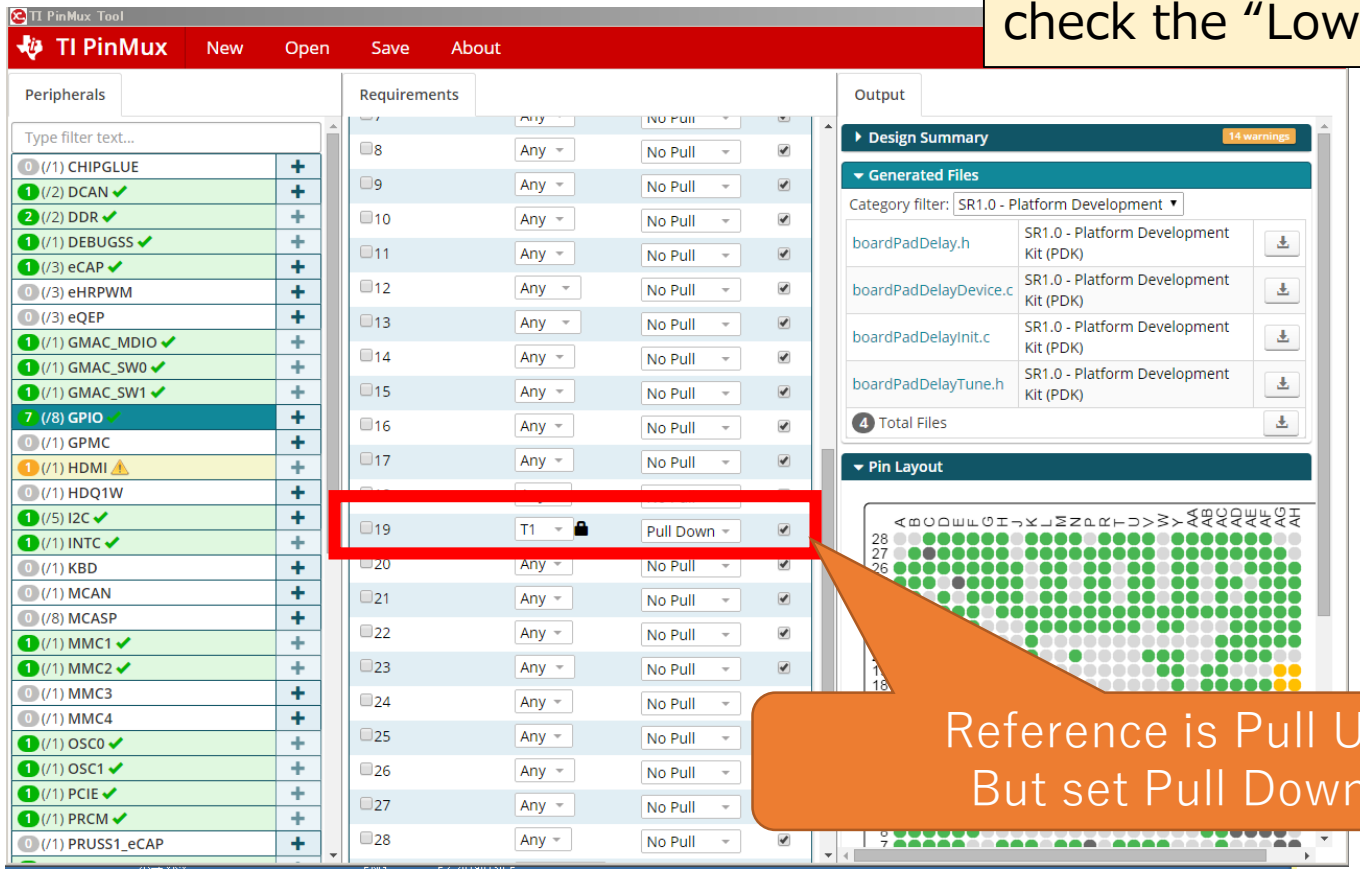
Difference from original test project is “Interrupt trigger”.

Original test	My test
Edge trigger	→ High Level(Low Level)

*Supplement

When checking "High Level", it is necessary to set Pull Down in PinMux Tool and rebuild.

If you want to check with Pull Up, check the “Low Level”.

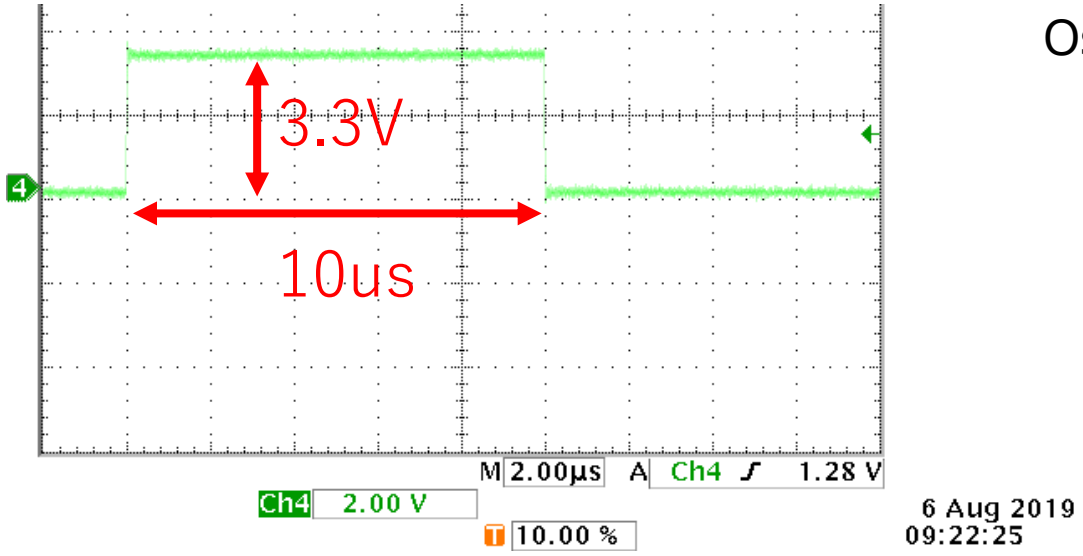


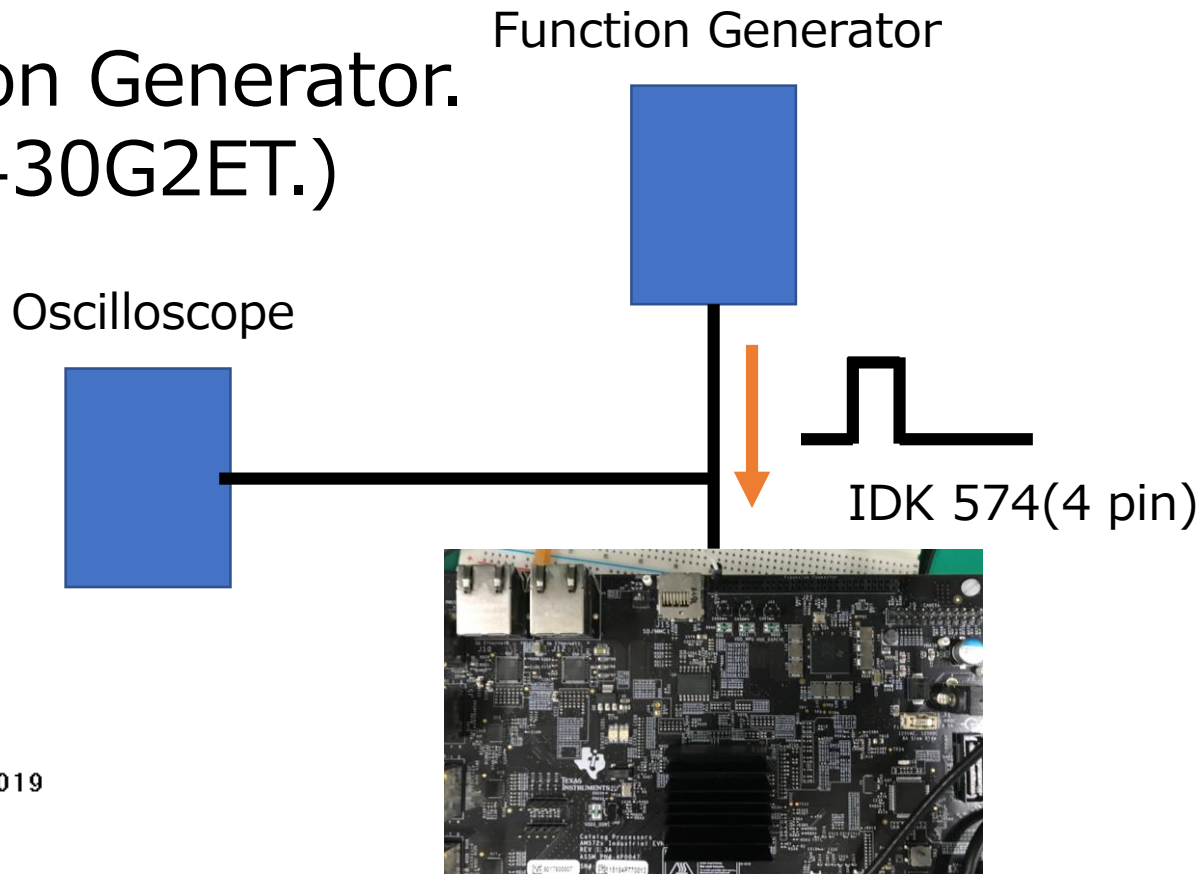
C:¥ti¥pdk_am57xx_1_0_11¥packages¥ti¥board¥src¥idkAM574x¥idkAM574x_pinmux.c

Set up :

Input a pulse to pin4 of the AM574IDK Expansion Connector.
You can check GPIO interrupt.

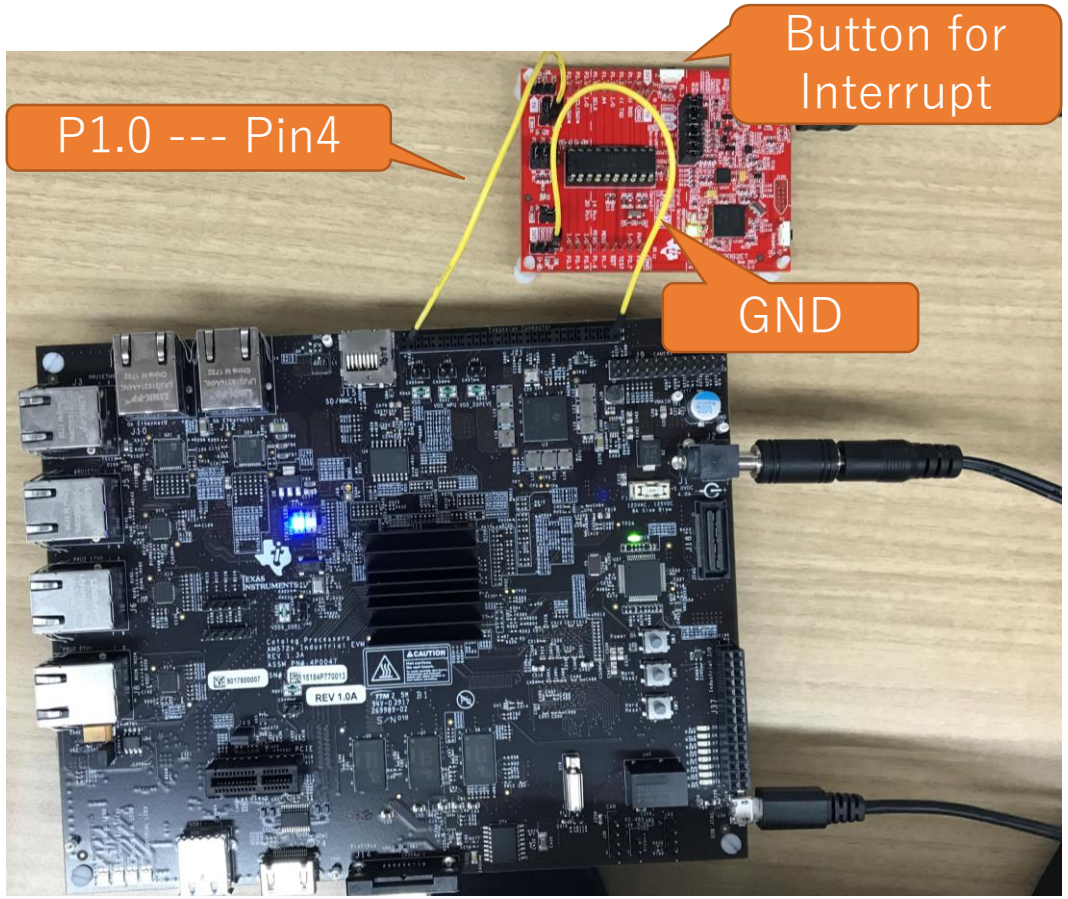
I input a 10us pulse from Function Generator.
(And make it easier, I used MSP430G2ET.)





*Supplement

Example of MSP430G2ET

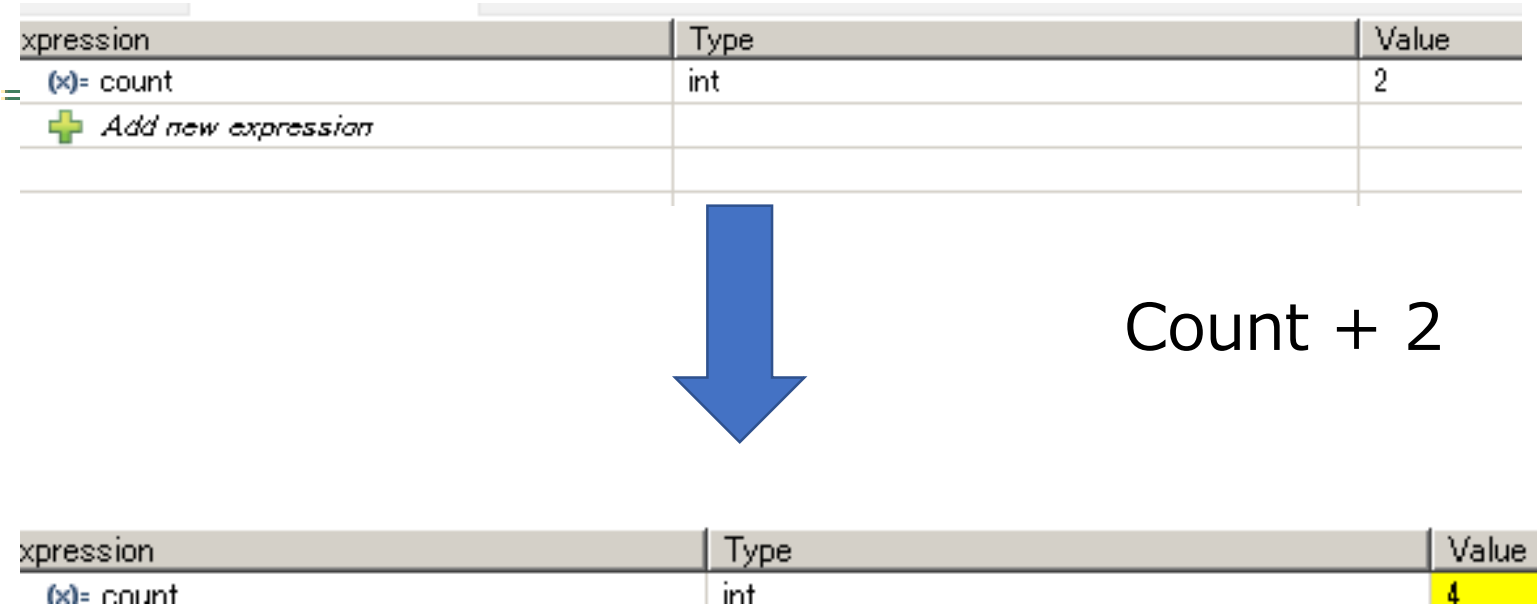



```
#include <msp430.h>

int main(void)
{
  WDTCTL = WDTPW + WDTHOLD;  // Stop watchdog timer
  P1DIR |= 0x01;             // Set P1.0 to output direction

  while (1)                            // Test P1.3
  {
    if (0x08 & ~P1IN){
      P1OUT |= 0x01;       // if P1.3 set, set P1.0
      __delay_cycles(7);  // about 10.2us
      P1OUT &= ~0x01;
      __delay_cycles(1000000); //Chattering prevention
    }
    else
       P1OUT &= ~0x01;       // else reset
  }
}
```

Confirmation :

GPIO CallbackFxn worked twice during one pulse input.
574IDK Blue LED also remains lit by toggling twice.

```
257 /*
258  *  ======== Callback function =
259  */
260 void AppGpioCallbackFxn(void)
261 {
262     /* Toggle LED1 */
263     count++;
264     GPIO_toggle(USER_LED1);
265     AppDelay(DELAY_VALUE);
266     gpio_intr_triggered = 1;
267 }
268
```



* GPIO_clearInt () is not added.
  I added debouncing code, but the situation got worse.

Experiments :

So we did some experiments.

- Add GPIO_clearInt() in CallbackFxn
- Change the trigger HighLevel(LowLevel) -> Rising edge

Results :

■ Add GPIO_clearInt() in CallbackFxn

```
0
1       /* Toggle LED1 */
2       count++;
3       check=1;
4       GPIO_toggle(USER_LED1);
5       AppDelay(DELAY_VALUE);
6       gpio_intr_triggered = 1;
7       GPIO_clearInt(USER_LED0);
8   }
```

CallbackFxn is called only once.
(count is once)

■ Change the trigger in GPIO_idkAM574x_board.c

CallbackFxn is called only once.
(count is once)

```
/* GPIO driver board specific pin configuration structure */
GPIO_PinConfig gpioPinConfigs_1p3[] = {
    /* Input pin with interrupt enabled */
    GPIO_DEVICE_CONFIG(GPIO_GRN_LED_PORT_NUM_1P3, GPIO_GRN_LED_PIN_NUM_1P3) |
//      GPIO_CFG_IN_INT_RISING | GPIO_CFG_INPUT,
//      GPIO_CFG_IN_INT_HIGH | GPIO_CFG_IN_PD | GPIO_CFG_INPUT,
    GPIO_CFG_IN_INT_HIGH | GPIO_CFG_INPUT,
```

Root issue :

We thought this issue was solved. However we confirmed about Hwi. So we noticed a serious anomaly of GPIO Driver (or sysbios).

We used Hwi.addHookSet function to check that Hwi is working properly. (Hwi timing, timestamp, what interrupt)

GpioCallbackFxn was executed once, but Hwi was happening twice.

How to check :

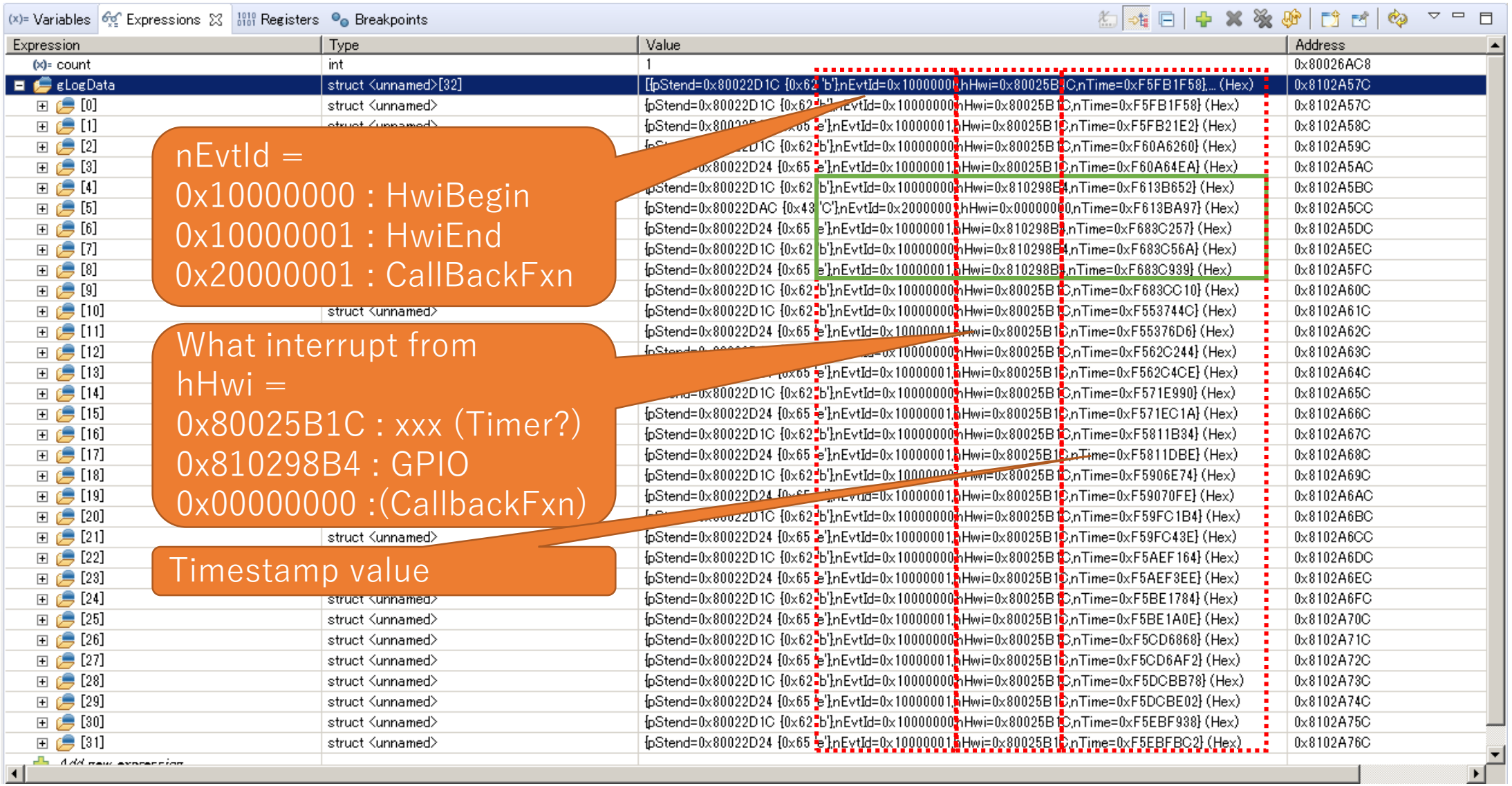
When Hwi Begin/End or CallbackFxn is executed, execution time and contents are stored in “gLogData”.

Stop Logging when a continuous interrupt is detected (when Hwi occurs twice in a short time).

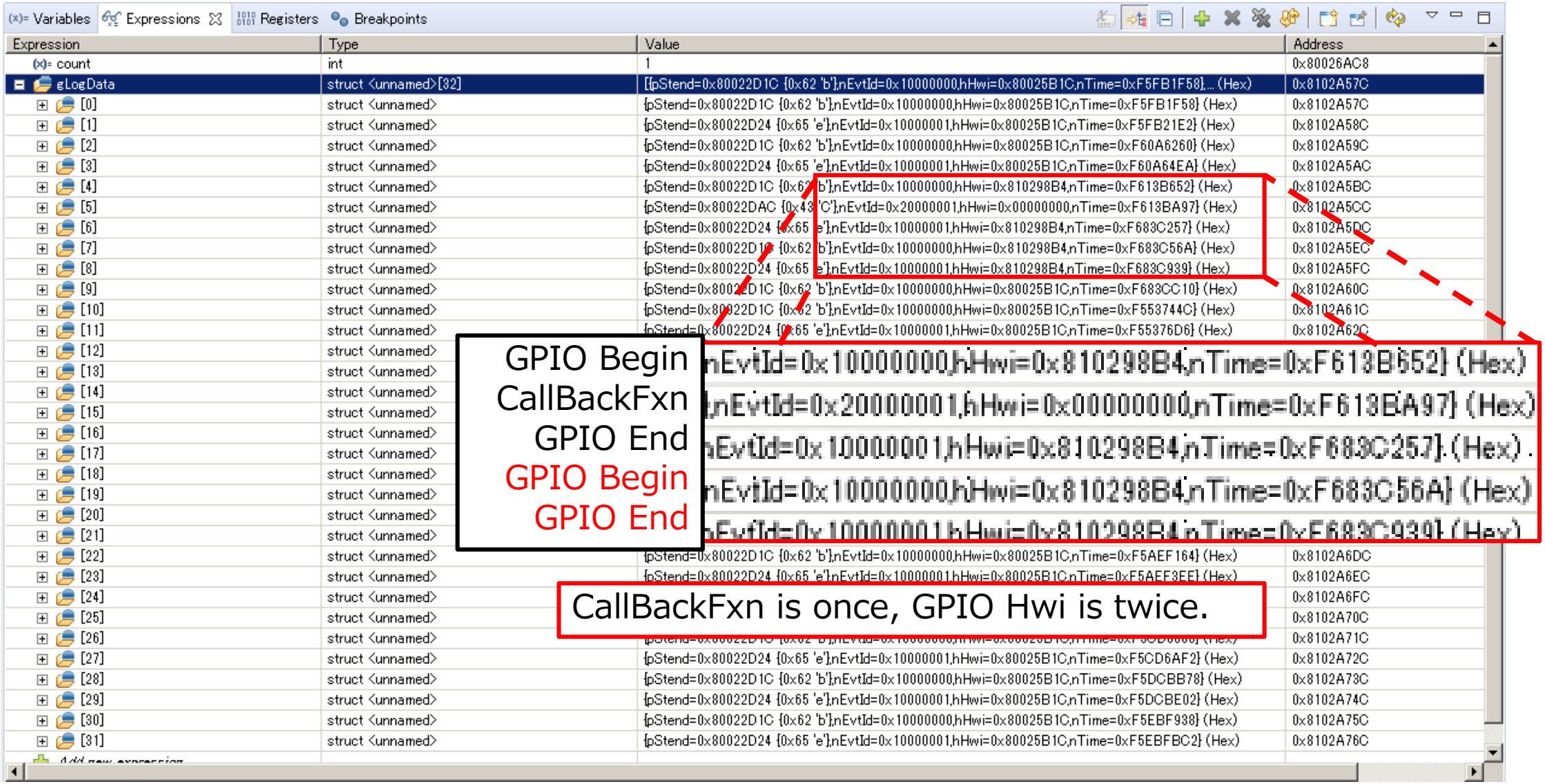
Therefore, after generating an interrupt once, suspend IDK and please check “gLogData”.

*"gLogData" is ring buffer, it needs to be sorted.

How to check : (Number format is Hex)



How to check : (Number format is Hex)



Root issue :

CallBackFxn is once, GPIO Hwi is twice.

When calculating the value of Timestamp （nTime),
the wasteful Hwi occupies about 50us. It’s a serious loss.

We checked in what case this occurred.

Level or Rising	GPIO_clearInt()	CallBackFxn	Hwi begin/end	Total Result
High/Low Level	-	2	2	NG
High/Low Level	Add	1	2	NG
Rising edge	-	1	2	NG
Rising edge	add	1	1	*OK

* Result is “OK”, but my customers want to use High Level.

Requests:

We want to fix the root issue.

Level or Rising	GPIO_clearInt()	CallBackFxn	Hwi begin/end	Total Result
High/Low Level	*	1	1	OK
High/Low Level	Add	1	1	OK
Rising edge	*	1	1	OK
Rising edge	Add	1	1	OK

* If GPIO_clearInt () needs to be added by user, please tell us? (At least not a test project .)

Reference: C:/ti/pdk_am57xx_1_0_11/packages/ti/drv/gpio/docs/doxygen/html/_g_p_i_o_8h.html#a3653b24f2fa808ad1f35d12f7cbbaecf

void GPIO_clearInt()
Note :It is not necessary to call this API within a callback assigned to a pin.