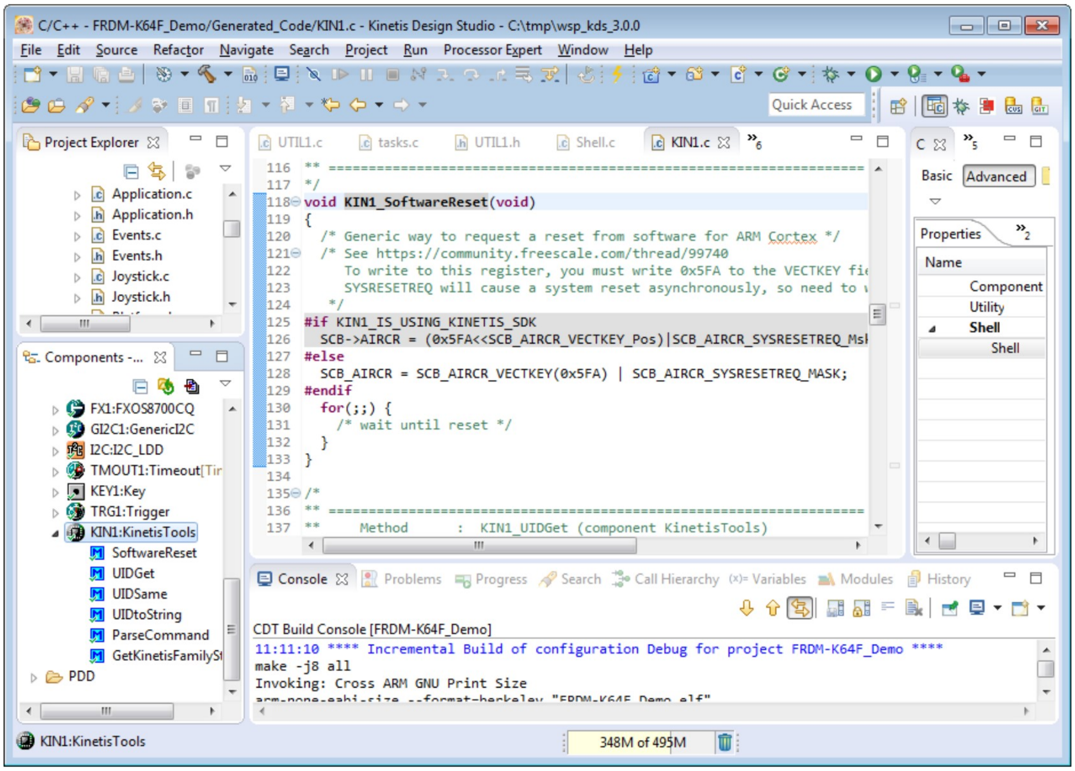
How to Reset an ARM Cortex-M with Software

Posted on July 1, 2015 by Erich Styger

There are cases when I need to do a reset of the device by software. For example I have loaded the application image with the bootloader, and then I need to perform a reset of the microcontroller to do a restart. As a human user I can press the reset button on the board. But how to do this from the software and application running on the board, without user manual intervention? Or if I simply want to reset the system for whatever reason?



```
** ====================================================================
*/
void KIN1_SoftwareReset(void)
{
  /* Generic way to request a reset from software for ARM Cortex */
  /* See https://community.freescale.com/thread/99740
     To write to this register, you must write 0x5FA to the VECTKEY fie
     SYSRESETREQ will cause a system reset asynchronously, so need to w
   */
#if KIN1_IS_USING_KINETIS_SDK
  SCB->AIRCR = (0x5FA<<SCB_AIRCR_VECTKEY_Pos)|SCB_AIRCR_SYSRESETREQ_Msk
#else
  SCB_AIRCR = SCB_AIRCR_VECTKEY(0x5FA) | SCB_AIRCR_SYSRESETREQ_MASK;
#endif
  for(;;) {
    /* wait until reset */
  }
}

/*
** ====================================================================
**     Method      :  KIN1_UIDGet (component KinetisTools)
```

CDT Build Console [FRDM-K64F_Demo]

```
11:11:10 **** Incremental Build of configuration Debug for project FRDM-K64F_Demo ****
make -j8 all
Invoking: Cross ARM GNU Print Size
arm-none-eabi-size --format=berkeley "FRDM-K64F_Demo.elf"
```

— Performing a Software System Reset with Kinetis Design Studio

Using Watchdog Timeout

In the past I have used the following approach for other microcontroller (e.g. the Freescale S08 and S12 devices):

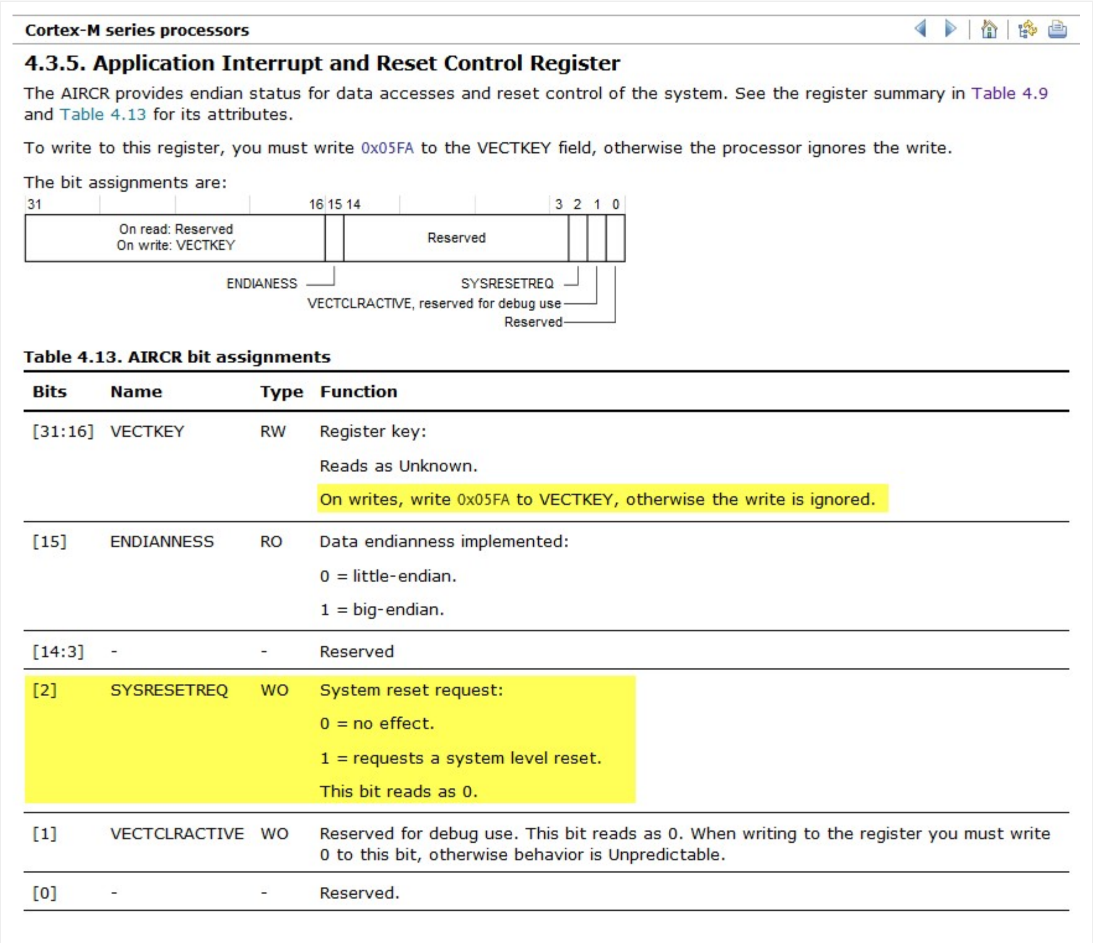
1. Setting up a watchdog timer
2. Then when I want to do a reset, I do *not* kick (serve) the watchdog timer any more

3. As a result, the WDT (watchdog timer) or COP (Computer operating properly) will timeout, and will reset the part

That approach is working, but well is not the easiest way. Especially as on ARM Cortex-M there is a better way :-).

Using ARM System Reset

The ARM Cortex-M which includes the Freescale Kinetis series cores have a System Reset functionality available in the AICR (Application Interrupt and Reset Control Register):



Cortex-M series processors

4.3.5. Application Interrupt and Reset Control Register

The AIRCR provides endian status for data accesses and reset control of the system. See the register summary in Table 4.9 and Table 4.13 for its attributes.

To write to this register, you must write 0x05FA to the VECTKEY field, otherwise the processor ignores the write.

The bit assignments are:

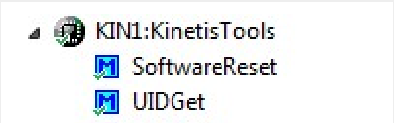
Table 4.13. AIRCR bit assignments

Bits	Name	Type	Function
[31:16]	VECTKEY	RW	Register key: Reads as Unknown. On writes, write 0x05FA to VECTKEY, otherwise the write is ignored.
[15]	ENDIANNESS	RO	Data endianness implemented: 0 = little-endian. 1 = big-endian.
[14:3]	-	-	Reserved
[2]	SYSRESETREQ	WO	System reset request: 0 = no effect. 1 = requests a system level reset. This bit reads as 0.
[1]	VECTCLRACTIVE	WO	Reserved for debug use. This bit reads as 0. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable.
[0]	-	-	Reserved.

— AIRCR Register (Source: ARM Infocenter)

So all I need to write a 0x05FA to VECTKEY with a 1 to SYSRESETREQ :-).

The easiest way is if I used the **KinetisTools** Processor Expert component from SourceForge (see “McuOnEclipse Releases on SourceForge“):



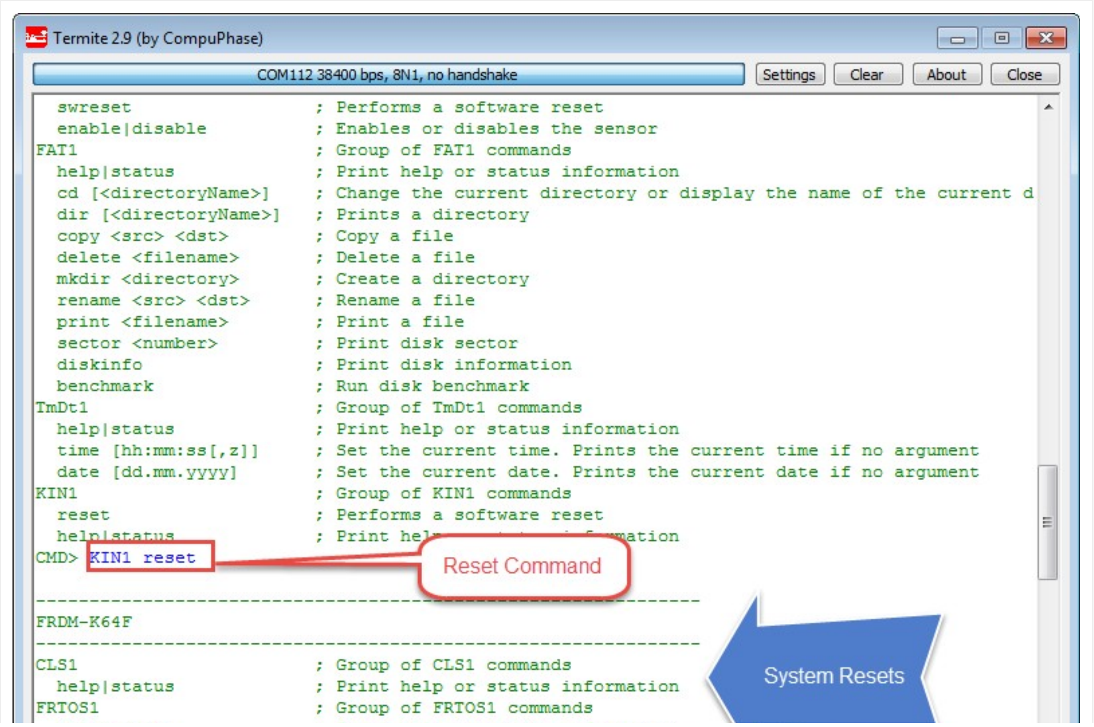
— KinetisTools Processor Expert Component

This componet offers a **SoftwareReset()** function which I can use in my application. It is defined like this in the component:

```
void KIN1_SoftwareReset(void)
{
  /* Generic way to request a reset from software for ARM Cortex */
  /* See https://community.freescale.com/thread/99740
     To write to this register, you must write 0x5FA to the VECTKEY fi
     SYSRESETREQ will cause a system reset asynchronously, so need to 
   */
#if KIN1_IS_USING_KINETIS_SDK
  SCB_AIRCR = (0x5FA<<SCB_AIRCR_VECTKEY_Pos)|SCB_AIRCR_SYSRESETREQ_Msk
#else
  SCB_AIRCR = SCB_AIRCR_VECTKEY(0x5FA) | SCB_AIRCR_SYSRESETREQ_MASK;
#endif
  for(;;) {
    /* wait until reset */
  }
}
```

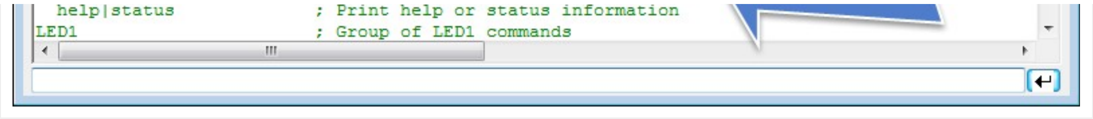
So all what you need is to have such a piece of code in your application to do a system reset.

That component features as well an optional command line interface. That way I can reset the target with a command from the shell 🙂


Termite 2.9 (by CompuPhase)
COM112 38400 bps, 8N1, no handshake
Settings
Clear
About
Close
Reset Command
System Resets

```
  swreset                   ; Performs a software reset
  enable|disable            ; Enables or disables the sensor
FAT1                        ; Group of FAT1 commands
  help|status               ; Print help or status information
  cd [<directoryName>]      ; Change the current directory or display the name of the current d
  dir [<directoryName>]     ; Prints a directory
  copy <src> <dst>          ; Copy a file
  delete <filename>         ; Delete a file
  mkdir <directory>         ; Create a directory
  rename <src> <dst>        ; Rename a file
  print <filename>          ; Print a file
  sector <number>           ; Print disk sector
  diskinfo                  ; Print disk information
  benchmark                 ; Run disk benchmark
TmDt1                       ; Group of TmDt1 commands
  help|status               ; Print help or status information
  time [hh:mm:ss[,z]]       ; Set the current time. Prints the current time if no argument
  date [dd.mm.yyyy]         ; Set the current date. Prints the current date if no argument
KIN1                        ; Group of KIN1 commands
  reset                     ; Performs a software reset
  help|status               ; Print hel                 mation
CMD> KIN1 reset

--------------------------------------------------------------
FRDM-K64F
--------------------------------------------------------------
CLS1                        ; Group of CLS1 commands
  help|status               ; Print help or status information
FRTOS1                      ; Group of FRTOS1 commands
```



```
  help|status          ; Print help or status information
LED1                   ; Group of LED1 commands
```

— Reset System from the Shell

Summary

To reset an ARM Cortex M by software, I can use the AIRCR register. Either I can do this directly, or using my KinetisTools component for Processor Expert :-).

> *As an additional note: there is a sticky 'System Reset Status' register available (I think on most Kinetis devices) which reports the reason for the reset. A good idea is to read that register at startup time and e.g. print the reset reason to the console (Thanks Charles for that cool idea!).*

Happy resetting 🙂

LINKS:

- Cortex-M3 Technical Reference Manual: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0337e/ch06s03s02.html
- McuOnEclipse Releases on SourceForge

SHARE THIS:

Print Email Reddit Twitter Facebook LinkedIn Tumblr Pinterest Pocket WhatsApp Telegram Skype

Like

One blogger likes this.

RELATED

Going to Mars: Building a DIY Eclipse IDE for ARM Embedded Microcontrollers
In "ARM"

Bricking and Recovering FRDM-KL25Z Boards: Reset, SWD Clock and Low Power
In "Boards"

Tutorial: Bits and Pins with Kinetis and the FRDM-KL25Z Board
In "Boards"

This entry was posted in **ARM**, **Boards**, **CPU's**, **Debugging**, **Embedded**, **Embedded Components**, **Freescale**, **HCS08**, **Processor Expert**, **S12**, **SDK**, **SourceForge**, **Tips & Tricks** and tagged **Debugging**, **Embedded Component**, **Freescale**, **Processor Expert**, **technology**, **Tips&Tricks** by **Erich Styger**. Bookmark the **permalink [https://mcuoneclipse.com/2015/07**

/01/how-to-reset-an-arm-cortex-m-with-software/] .

About Erich Styger

Embedded is my passion....
View all posts by Erich Styger →

45 THOUGHTS ON “HOW TO RESET AN ARM CORTEX-M WITH SOFTWARE”

Bill
on **July 2, 2015 at 03:39** said:

Great bit of info. Thanks.

★ Like

Vishal_Girisagar
on **July 2, 2015 at 08:50** said:

Great information Erich. Thank you.

I had a query. Is it applicable to ARM cortex M4 as well?

★ Like

Erich Styger
on **July 2, 2015 at 09:18** said:

Hi Vishal,
yes, it applies to ARM Cortex M4 as well.

★ Like

Naruto Uzumaki
on **July 2, 2015 at 11:06** said:

Thanks a lot, Erich.

Regards,
Naruto Uzumaki 🙂

★ Like

Alberto Spagnolo
on **July 2, 2015 at 11:09** said:

Or, you could use CMSIS with NVIC_SystemReset(). It does the same, but it's already there for you 😉

★ Liked by 2 people

> **Erich Styger**
> on **July 2, 2015 at 11:11** said:
>
> Yes, true. But only if using the CMSIS and if that is provided by the vendor, which is not always the case.
>
> ★ Like

Pingback: McuOnEclipse Components: 05-July-2015 Release | MCU on Eclipse

Gary Lynch
on **November 23, 2015 at 09:05** said:

I am extrapolating from this that KDS provides no command to reset the processor from either the menu system or the toolbar? I have spent most of my career using classical embedded tools and am still sometimes taken aback by the things Eclipse cannot do.

I have used one other Eclipse derivative (Code Composer Studio) and that feature was built in.

★ Like

Erich Styger
on **November 23, 2015 at 09:11** said:

Hi Gary,
of course KDS can reset/restart the microcontroller under debug, so your extrapolation is not correct ;-).
This article is about performing a reset from the software/application running on the target, not about how to do a reset with a debugger.
I hope that makes sense?
Erich

★ Like

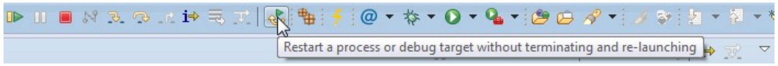
Gary Lynch
on **November 23, 2015 at 09:24** said:

May I ask, then, where it is? I have made several attempt to find it, but so far no luck.

★ Like

Erich Styger
on **November 23, 2015 at 09:33** said:

It is the restart button:


Restart a process or debug target without terminating and re-launching

★ Like

Gary Lynch
on **November 23, 2015 at 09:45** said:

Ok, I did find that, but the description text: (Restart a process or debug target without terminating and re- launching) was so abstruse, I didn't recognize it.

Thanks, Erich!

★ Like

Lior
on **June 16, 2016 at 11:22** said:

This reset usualy works for us, but sometimes it doesn't… I've noticed that the user guide says: SYSRESETREQ -> asserts a proc_reset_signal.
This is intended to force a large system reset of all major components **except for debug**.

What does it mean? Debug means that I am currently debugging using a JTAG, or does it mean that I am running a "debug" configuration compiled image?

Also, I've found that the AIRCR register has two more bits:
VECTRESET & VECTCLRACTIVE, which we should reset whenever we write to this register, "otherwise behavior is Unpredictable". Can this be the case…?

Thanks!
Lior.

★ Like

Erich Styger
on **June 16, 2016 at 11:58** said:

Hi Lior,
it means that it does not reset the debug logic on the chip (has nothing to do with hardware).
I have not seen any issues on my side with the proposed way.

I see this note:
Note

SYSRESETREQ is cleared by a system reset, which means that asserting VECTRESET at the same time may cause SYSRESETREQ to be cleared in the same cycle as it is written to. This may prevent the external system from seeing SYSRESETREQ. It is therefore recommended that VECTRESET and SYSRESETREQ be used exclusively and never both written to 1 at the same time.

★ Like

Lior
on **June 16, 2016 at 12:28** said:

Thanks, Erich!
So, in other words, is it recommended to clear both bits (VECTRESET & VECTCLRACTIVE) at the same time that I set SYSRESETREQ – in order to prevent it to interfere with the reset action?

★ Like

Erich Styger
on **June 16, 2016 at 12:31** said:

Hi Lior,
Only set the SYSRESETREQ bit, like in
SCB_AIRCR = SCB_AIRCR_VECTKEY(0x5FA) |
SCB_AIRCR_SYSRESETREQ_MASK;

★ Like

Lior
on **June 21, 2016 at 14:48** said:

Hey,
I've noticed that maybe there is another option for Cortex-M3, which is to use RSTC_CR.
Have anyone tried this option?

Thanks,
Lior.

★ Like

Pingback: Debugging ARM Cortex-M0+ HardFaults | MCU on Eclipse

Julian
on September 27, 2017 at 15:58 said:

HI! I had a problem with de watchdog. I initialize the watchdog, I do not refresh in the loop and the core doesn t reset. All the registers are initialize ok beacause y see them in the debuger.
¿Does anyone know if there is another register to enable the clock or something similar?

★ Like

Erich Styger
on September 27, 2017 at 16:09 said:

Which device, and which debugger? The watchdog functions depend on the device implementation, and I have seen debuggers which disable the watchdog (so you can debug).
Do you have the same if you run the board without the debugger?

★ Like

Nick
on October 3, 2017 at 18:14 said:

Hi, thanks for your great posts! I’m getting an error when I try to Generate Processor Expert Code:
“Description Resource Path Location Type
Generator: FAILURE: at line 72: “Shell”/Shell has not assigned the component (file: Drivers\sw\KinetisTools.drv) WT4 KIN1 Processor Expert Problem”

If I tick the checkbox to use a shell (and add the required components for that) then it builds just fine. The problem seems to only occur when I try to use the KinetisTools without a shell. Any suggestions?

★ Like

Erich Styger
on **October 3, 2017 at 19:27** said:

Hi Nick
Just checking: Are you using the latest version (https://mcuoneclipse.com/2017/09/25/mcuoneclipse-components-25-sept-2017-release/)?

★ Like

Nick
on **October 3, 2017 at 20:21** said:

Yes, I just installed the mcuoneclipse components (latest version) when I tried this.

★ Like

Erich Styger
on **October 3, 2017 at 21:47** said:

Hi Nick,
hmm, I'm not able to reproduce this. Would you mind to send me your project (I don't need the source files to my email address listed on https://mcuoneclipse.com/about/ so I can try to reproduce this?

★ Like

Nick
on **October 4, 2017 at 21:51** said:

I just sent it over

★ Like

Erich Styger
on **October 5, 2017 at 07:14** said:

Hi Nick,
got it, thank you, looking into it as soon as I can and will respond to you.
Thanks!
Erich

★ Like

Erich Styger
on **October 5, 2017 at 10:25** said:

Hi Nick,
Ok, I see now the problem :-(. I have fixed it and sent you the updated component. It is available on SourceForge as a patch release too: https://sourceforge.net/projects/mcuoneclipse/files/PEx%20Components/
I appologize for the problem it caused,
Erich

★ Like

Shiva
on **July 20, 2018 at 13:51** said:

Hi Erich,

The given information is useful foe me, i wanted to read the reset reason as well. i have tried to read the reset reason using the CPUDBGSCRATCH register, but in the this memory (0xe010200c) always i can see 0xFFFFFFFC values. can you please suggest how exactly i can read the reset reason.

Thanks in advance.

★ Like

Erich Styger
on **July 20, 2018 at 16:26** said:

Hi Shiva,
which microcontroller are you using?

★ Like

Banu
on **July 31, 2018 at 18:14** said:

Hi Erich,
Thank you for your explanation.
I'm working with power modes and using FRDM-KL02z board, I tried to use this method to exit the stop mode, I wrote "KIN1_SoftwareReset();" after calling the enter stop mode function but it is not working. Do you have any idea why it is not working ?as I'm expecting to exit the stop mode into normal run mode.

Regards
Banu

★ Like

Erich Styger
on **July 31, 2018 at 20:19** said:

Hi Banu,
I have not used it in such a sitution. But if you are in stop mode, then no instructions get executed. So you need first to exit stop mode and then do the reset?

★ Like

Banu
on **August 1, 2018 at 16:27** said:

Hi Erich,
Yes, I need to exit the stop mode into the normal run, this can happen by either using reset or interrupt(WFI). I'm not sure how to use the WFI so

I'm thinking if there is software reset code that let the M0+ exit the stop mode?

Regards
Banu

★ Like

Erich Styger
on **August 1, 2018 at 16:47** said:

Hi Banu,
WFI (Wait for Interrupt) is a simple as this:

```
__asm("wfi");
```

★ Liked by 1 person

Banu
on **August 1, 2018 at 17:03** said:

Hi Erich,
Thanks for quick reply.
Yes, I tried " __asm("wfi");" but didn't work, my code is simply as below, I tried to place the "__asm("wfi");" after and before calling the enter stop mode function, do you have any idea why?

```
int main(void)

{
PE_low_level_init();

unsigned int i,j;
PWM1_Enable();
for(i=0;i<5000;i++)
for(;;) {
{
enter_stop();
PWM1_Disable();
for(i=0;i<5000;i++)
PWM1_Enable();
for(i=0;i<50000;i++);
}
}
void enter_stop(void)
{
volatile unsigned int dummyread;

SMC_PMCTRL &= ~SMC_PMCTRL_STOPM_MASK; //mask=0x7u
SMC_PMCTRL |= SMC_PMCTRL_STOPM(0);
/*wait for write to complete to SMC before stopping core */
dummyread = SMC_PMCTRL;
deepsleep();
}
void deepsleep (void)
{
/* Set the SLEEPDEEP bit to enable deep sleep mode (STOP) */
SCB_SCR |= SCB_SCR_SLEEPDEEP_MASK; //0x4u
#ifdef CMSIS
__wfi();
#else
/* WFI instruction will start entry into STOP mode */
__asm("WFI");
#endif

}
```

★ Like

Erich Styger
on **August 1, 2018 at 17:05** said:

your usage of WFI looks ok. All what you need then to continue after that instruction is to raise an interrupt (e.g. with an external pin interrupt).

★ Like

Banu
on **August 1, 2018 at 17:53** said:

I tired the interrupt but did not work. I enabled the interrupt after the WFI but nothing happen it enters the stop mode and stays there -_-

★ Like

Erich Styger
on **August 1, 2018 at 18:32** said:

WFI does exactly what its name is: it waits and sits there until an interrupt occurs. You have to enable interrupts *before* executing wfi.

★ Liked by 1 person

Banu
on **August 2, 2018 at 11:40** said:

Hi Erich,
I added TimerInt components and put the interrupt period 2sec, then wrote in the events.c a wakeup_flag and modify the below code in main.c please check:

```
PWM1_Enable();
for(i=0;i<5000;i++)
for(;;) {
for(i=0;i<1000;i++)
enter_stop();
wake_up_int_Enable();
if (Wakeup_flag==0)
{
deepsleep();
PWM1_Disable();
for(i=0;i<5000;i++)
PWM1_Enable();
for(i=0;i<5000;i++)
}
Wakeup_flag==1;
}
```

I enabled the wake_up_int and then checked the wakeup_flag in events.c and then called the deepsleep function which is the same as in the above post but still has the same issue, the MCU enter the stop mode any stay there 🙁.

Regards

Banu

★ Like

Erich Styger
on **August 2, 2018 at 11:41** said:

The interrupt to wake you up has to be turned on *before* you enter the stop mode.

★ Liked by 1 person

Erich Styger
on **August 2, 2018 at 11:42** said:

And you should chanbe `Wakeup_flag==1;` to `Wakeup_flag=1;`

★ Like

Banu
on **August 2, 2018 at 12:04** said:

Thank you for the quick reply.
I modified it as below:

```
/*****Main.c****/
unsigned char Wakeup_flag=0;
PWM1_Enable();
for(i=0;i<5000;i++)
for(;;) {
for(i=0;i<1000;i++)
wake_up_int_Enable();
if (Wakeup_flag==0)
{
enter_stop();
deepsleep();
PWM1_Disable();
for(i=0;i<5000;i++)
PWM1_Enable();
for(i=0;i<5000;i++)
}
Wakeup_flag=1;
}

/**********Events.c*****/
void wake_up_int_OnInterrupt(void)
{
/* Write your code here … */
Wakeup_flag=0;
LED1_On();
}
```

Still the same issue.

★ Like

Pingback: FreeRTOS: how to End and Restart the Scheduler | MCU on Eclipse

Raz3l
on **January 15, 2020 at 14:23** said:

@Erich,

Could you quickly explain in this comment or in a new series on LPC55S69 how one can reset the processor in both security states?

Thank you!

★ Like

Raz3l
on **January 15, 2020 at 14:23** said:

@Erich,

Could you quickly explain in this comment or in a new series on LPC55S69 how one can reset the processor in both security states?

Thank you!

Could you quickly explain in this comment or in a new series on LPC55S69 how one can reset the processor in both security states?

Thank you!

★ Like

Erich Styger
on **January 16, 2020 at 20:18** said:

I have not explicitly tried, but I would have assumed it would work the same way? Are you saying it is not?

★ Like