

---

## FCCU setup to allow reset from SWT2

### Introduction

This technical note describes how to handle the Fault Collection and Control Unit (FCCU) to trigger with a reset in case the Soft Watchdog Timer (SWT) reaches its timeout.

The SWT prevents that the system locks up in endless situations such as software getting trapped in a loop or if a bus transaction fails to terminate. When enabled, the SWT requires periodic execution of a watchdog servicing operation. The servicing operation resets the timer to a specified time-out period. If this servicing action does not occur before the timer expires, the SWT reports a SWT fault to the FCCU which takes the proper reaction. E.g., a reset request to the Reset Generation Module (RGM).

This document focuses on SWT2 of the SPC58xC devices.

Most of the concepts, however, are valid for all SWT instances of the SPC58xx family.

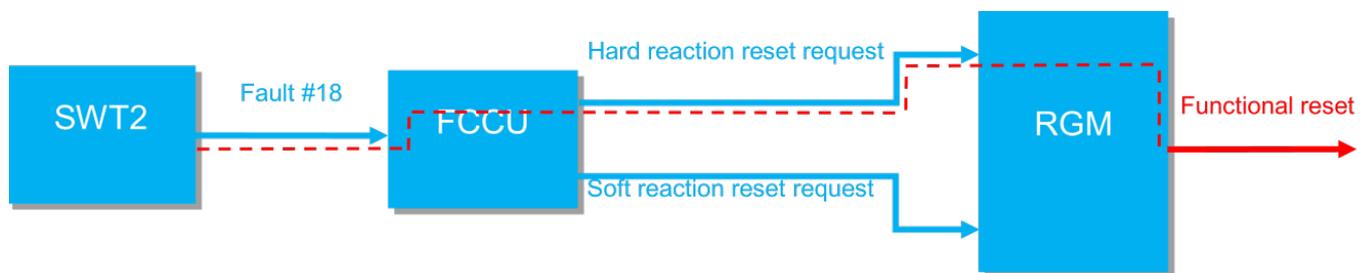
## 1 FCCU setup to allow reset from SWT2

### 1.1 Software Watchdog timer SWT2 reset flow

If the user doesn't refresh the SWT2 timer properly and its timeout expires, the SWT2 triggers the fault 18 to the FCCU.

The FCCU - depending on its configuration - requests a functional reset to the RGM which manages the reset sequence of the device.

Figure 1. SWT2 reset flow



### 1.2 SWT2 setup

By default, the SWT2 after the reset is enabled and locked by a soft lock. Before accessing the registers of the SWT2, the user must remove this lock.

This technical note and the related example code consider the SWT2 in Regular service mode with the Fixed Service sequence. For more information about these working modes refer to the Reference Manual.

```

void disable_SWT_2(void)
{
    SWT_2.SR.R = 0x0000C520U; /* clear soft lock bit 1/2 */
    SWT_2.SR.R = 0x0000D928U; /* clear soft lock bit 2/2 */
    SWT_2.CR.R = 0xFF000000U; /* setup SWT_2 (disable) */
}
  
```

```

void configure_SWT_2(void)
{
    SWT_2.SR.R = 0x0000C520U; /* clear soft lock bit 1/2 */
    SWT_2.SR.R = 0x0000D928U; /* clear soft lock bit 2/2 */
    /* setup Watchdog time-out period in clock cycles
    (The initial value is 0x0003_FDE0) */
  
```

```

    SWT_2.TO.R = 0x00ABCDEFU;
    /* fixed service sequence, regular mode,
    disable timer for debug mode, enable watchdog */
    SWT_2.CR.R = 0xFF00000BU;
}
  
```

```

void refresh_SWT_2(void)
{
    SWT_2.SR.R = 0x0000A602U; /* fixed service sequence 1/2 */
    SWT_2.SR.R = 0x0000B480U; /* fixed service sequence 2/2 */
}
  
```

## 1.3 FCCU setup

The FCCU offers a systematic approach to fault detection and control. The FCCU provides a hardware channel to collect errors and to move the device to a safety state when a failure occurs. The FCCU can manage faults coming from a monitor in two different ways: HW and SW recoverable faults.

If the user configures a fault as hardware recoverable, the FCCU keeps its internal status of the fault asserted until the monitor signals the faulty condition.

If the user configures a fault as software recoverable, the FCCU keeps its internal status of the fault asserted until the software explicitly clears it.

For examples, we can assume that a monitor activates the fault X of the FCCU at time\_1 and de-activates it at time\_2:

1. If the user configures the fault X as hardware recoverable, the FCCU clears its status register of the fault X at time\_2 without any software intervention.
2. If the user configures the fault X as software recoverable, the FCCU keeps the status register of the fault X until the software clears it.

### 1.3.1 SW recoverable fault handling of the SWT2 reset request (Fault #18)

Hereafter the default configuration (reset values) of the FCCU fault related to the SWT2, i.e. FCCU[18]:

- FCCU\_RF\_CFG0[RFC18] = 1
  - software recoverable fault
- FCCU\_RFS\_CFG1[RFSC18] = 2
  - The FCCU reacts to the FCCU[18] with a long functional reset request to the RGM.

By default, both the SWT2 and its related fault in the FCCU are active. As a consequence, after power on, the RGM triggers a functional reset if the user doesn't service the SWT2 properly.

Besides, the FCCU latches the SWT2 timeout fault in the FCCU\_RF\_S0[RFS18]. This flag - as all other FCCU registers - is persistent during a functional reset. Under these circumstances, the FCCU doesn't service any other SWT2 timeout fault until the user doesn't clear the FCCU\_RF\_S0[RFS18]. For proper handling of SWT2 as SW recoverable fault, the user must clear the status flag FCCU\_RF\_S0[RFS18] each time after the reset to allow processing of next reset by the FCCU. See code example.

For later use, The user may log the occurrence of the FCCU[18] before clearing it.

**Note:** *A functional reset is necessary to clear the fault indication of the SWT after its timeout. In the case described above, the FCCU triggers a functional reset to react to the fault.*

```
void clear_RF_S_FCCU_for_SWT_2(void)
{
    FCCU.RFK.R = 0xAB3498FEU; /* write FCCU_RF_Sn unlock key. */
    /* clear the RFS18 , opcode OP12 is automatically set
    into the FCCU_CTRL.OPR field */
    FCCU_RF_S[0].R = 0x00040000U;
    /* wait for operation finalization */
    while (FCCU_CTRL.B.OPS != FCCU_OPS_SUCCESS);
    /* read status flag for SWT2 (RFS18) to verify the deletion */
    FCCU_CTRL.R = 0x000000AU; /* Set the FCCU into the OP10 */
    /* wait for operation finalization */
    while (FCCU_CTRL.B.OPS != FCCU_OPS_SUCCESS);
    if (FCCU_RF_S[0].B.RFS18 != 0)
    {
        /* failure to clear, repeat the clearing sequence above */
    }
}
```

**Note:** *There should not be any other operation in between the above greyed lines. For example command like "FCCU\_RF\_S[0].B.RFS18 = 1;" usually leads to Read-modify-write Asm translation is not allowed here and leads to the Aborted operation FCCU\_CTRL[OPS] = 2. It is also recommended to jump over both line/execute whole assembly block while debugging the code.*

### 1.3.2 HW recoverable fault handling of the SWT2 reset request (Fault #18)

As the second option, the user can configure the FCCU input channel #18 (SWT2 reset request) as HW recoverable by changing the FCCU\_RF\_CFG0[RFC18] from its default value to 0b. See the example code below.

```
void configure_FCCU_for_SWT_2(void)
{
    /* enter the CONFIG mode */
    FCCU.TRANS_LOCK.R = 0x000000BCU; /* write the TRANSKEY */
    /* configure the CONFIG timeout */
    FCCU.CFG_TO.R = 0x00000006U;
    FCCU.CTRLK.R = 0x913756AFU; /* key for OPERATION 1 */
    /* set the FCCU into the CONFIG state OP1 */
    FCCU.CTRL.R = 0x00000001U;
    /* wait for operation finalization */
    while (FCCU.CTRL.B.OPS != FCCU_OPS_SUCCESS);
    /* enable HW recoverable fault to generate Reset on SWT_2 fault */
    FCCU.RF_CFG[0].B.RFC18 = 0x0U;
    /* long Functional Reset as reaction to the SWT_2 refresh fault */
    FCCU.RFS_CFG[1].B.RFSC2 = 0x2U;
    /* move to FAULT state when SWT_2 expires to generate Reset */
    FCCU.RF_E[0].B.RFE18 = 0x1U;

    /* enter back to the NORMAL mode */
    FCCU.CTRLK.R = 0x825A132BU; /* key for OPERATION 2 */
    /* set the FCCU into the NORMAL state OP2 */
    FCCU.CTRL.R = 0x00000002U;
    /* wait for operation finalization */
    while (FCCU.CTRL.B.OPS != FCCU_OPS_SUCCESS);
}
```

**Note:** *There should not be any other operation in between the above greyed lines. It is also recommended to jump over both line/execute whole assembly block while debugging the code.*

Since a functional reset doesn't initialize the FCCU registers, the above change from SW recoverable to HW recoverable fault is persistent until a destructive reset.

## 1.4 RGM setup

The RGM centralizes the different reset sources and manages the reset sequence of the chip. Two FCCU outputs go to the RGM as two reset source (i.s., FCCU hard reaction reset and FCCU soft reaction reset). The FCCU hard reaction reset corresponds to the Long functional reset request (FCCU\_RFS\_CFGn[RFSCn]= 2) and the FCCU soft reaction reset corresponds to the Short functional reset request (FCCU\_RFS\_CFGn[RFSCn]= 1 ). By default, the RGM is configured to handle both FCCU functional reset request.

## Revision history

**Table 1. Document revision history**

Date	Version	Changes
11-Jun-2019	1	Initial release.

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2019 STMicroelectronics – All rights reserved