Freescale Semiconductor
Application Note

AN3831
Rev. 1.0, 3/2009

Getting Started with eXtreme Switch Gen III

Basic Device intialization and configuration examples

By: Paola Ponce, Oscar Camacho, Jaime Orozco

1 Overview

This document serves as a quick reference to provide Embedded engineers with code snippets and initialization examples for getting up and running with the Freescale Family of high side switches known as eXtreme Switch GEN III. It shows control functions, such as Direct Control, Self-PWM generation, current sensing, protections, and diagnostics, among others.

Contents

freescale™ semiconductor

2 Introduction

The eXtreme Switch family of devices is specially designed for low voltage automotive lighting applications. These devices include multiple High Side Drivers with very low RDS_ON. Programming, control, and diagnostics are accomplished using a 16 bit SPI interface. Additionally, each output has its own parallel input or SPI control for pulse-width modulation (PWM) control if desired.

The family includes several part numbers, MC35XS3400, MC10XS3435, MC10XS3412, and MC15XS3400, all of them are quad high side drivers in the PQFN-24 package, with differences in the RDS_ON, power capability, the same pin-out, and SPI configuration registers.

The eXtreme Switch family is compatible with incandescent, halogen, Xenon (HID), and LED lamps, and has programmable over-current and short-circuit protection levels. The devices have a fail-safe mode to provide functionality of the outputs in case the MCU is damaged.

2.1 Protection and diagnostic

The device incorporates the following protections and diagnostics, some of which are programmable and can be reported via the SPI to the microcontroller. For more detailed information please refer to the device data sheet.

- Over-temperature
- Over-current
- Severe short-circuit
- Over/under-voltage
- Reverse battery
- Active Clamp protection
- Loss of ground/V_{BAT}
- Energy discharge protection

2.2 Operational Modes

The eXtreme Switch family has four operating modes: Sleep mode, Normal mode, Fail-safe mode and Fault mode. The device changes its current consumption and capabilities depending on the operational mode. Detailed information for each of the modes can be found on the device's data sheet.

3 Application Example

The example shown on this application note is being developed using CodeWarrior™ for the 9S12XEP100, 16 bit microcontroller.

The driver created for this example was created for the 10XS3412 eXtreme switch. This driver can be used with other members of the device family without any issue.

For user convenience, the code examples have been integrated into functions that can be migrated to different platforms. Such functions have been included into specific ".c" and ".h" files within the project.

The source code project can be downloaded from Freescale Web Page www.freescale.com.

Some macros have been defined in order to easily reassign pins and connections between the eXtreme Switch and the microcontroller. The next table contains the serial peripheral interface (SPI), interrupt (INT), and reset pins assignment summary. All macro declarations are contained in file lle_GPIO.h within the CodeWarrior project.

Serial Peripheral Interface SPI1	Microcontroller Pin/function
SCLK_MCU	SCK1
MOSI_MCU	MOSI1
MISO_MCU	MISO1

Table 1. Pin connections between MC10XS3412 and MC9S12XEP100

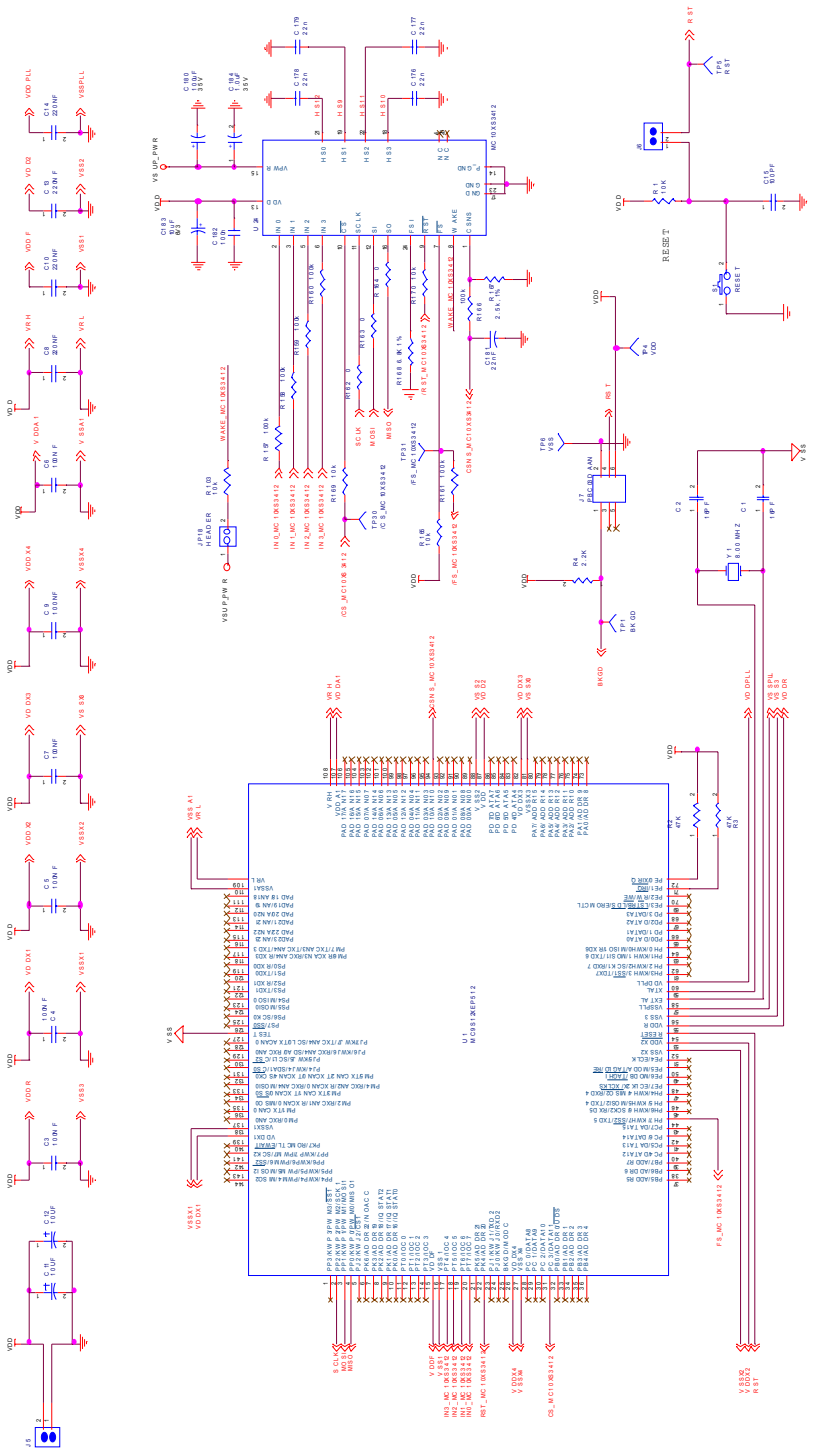
Function	eXtreme Switch PIN	Microcontroller Pin/function
Reset input	RST_MC10XS3412	PORTK_PK4 /* PK4 */
Chip Select	CS_MC10XS3412	PORTC_PC3 /* PC3 */
Direct Input HS0	IN0_MC10XS3412	PTT_PTT7 /* IOC7 */
Direct Input HS1	IN1_MC10XS3412	PTT_PTT6 /* IOC6 */
Direct Input HS2	IN2_MC10XS3412	PTT_PTT5 /* IOC5 */
Direct Input HS3	IN3_MC10XS3412	PTT_PTT4 /* IOC4 */
Fault Status output	FS_MC10XS3412	PTH_PTH7 /* KWH7 */
Output current monitoring	CSNS_MC10XS3412	Not used by the microcontroller in this example

In the following sections the user will find several functions declared in files des_MC10XS3412.h and des_MC10XS3412.c, within the CodeWarrior project that can be used to configure and control the High Side switches. Take into account that they are designed to work with the MC9S12XEP100 microcontroller via 16 bit SPI commands.

Refer to **4 Schematic** to review the electrical connections between the microcontroller and the eXtreme Switch.

4 Schematic

For user convenience, a schematic with the connections between the MC9S12XEP100 microcontroller and MC10X3412 eXtreme Switch has been included in this section.



5 Setting the Serial Peripheral Interface (SPI) Communication

The configuration and control of the eXtreme Switch Gen III is done via one 16 bit daisy chainable SPI command. Figure 1 shows the SPI timing characteristics to ensure proper communication.

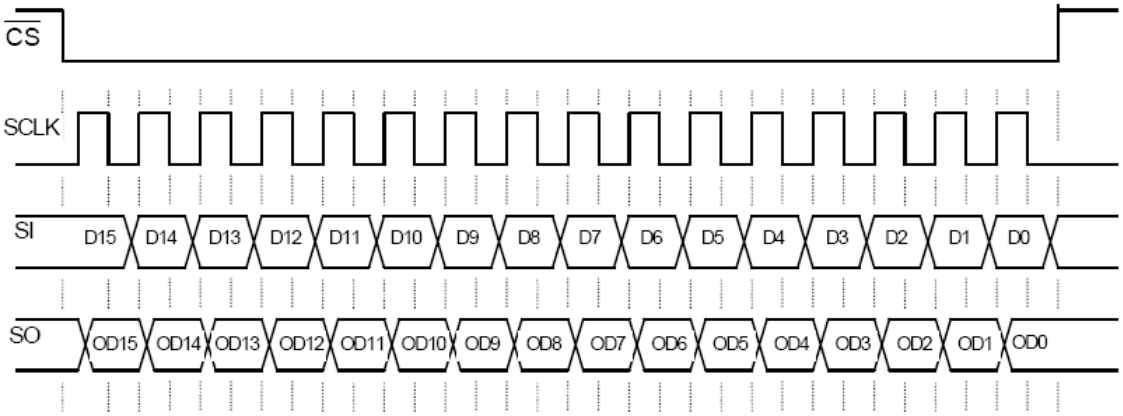


Figure 1. SPI Timing Characteristics

For this example, the SPI module is configured as a master at 500 kHz, with data sampling on the falling edges of the 9S12XEP100. Files lle_SPI.c and lle_SPI.h contain the SPI driver to communicate with the eXtreme Switch. Minor changes must be implemented to use this driver with a different microcontroller.

The function to configure the SPI in this example is shown by the following.

Function	Function Description	Return
lle_SPI_Init()	SPI1 module initializing in Master mode @ 500 kHz Slave select output pin not used by SPI module	Void

Remember the following points for using it:

- Before making the configuration, this function disables the SPI module and clears the status flags.
- To modify the baud rate, calculate the divider as follows:
  - SPI baud rate = bus_clock / baud_rate_divider
  - For SPI baud rate = 500 kHz and bus clock = 40 MHz, baud rate divider = 80
  - Baud rate divider = (SPPR+1)* 2 ^ (SPR+1)
- Modify SPI_SPPR and SPI_SPR on file lle_SPI.h, if a different baud rate is needed.
- The following bits are set in the SPI Control Register 1
  - SPI1CR1_MSTR = 1; (SPI in Master mode)
  - SPI1CR1_CPHA = 1; (Data sampling at falling edges)
  - SPI1CR1_SPE = 1; (SPI module enabled)

In order to send and receive data between the microcontroller and the eXtreme Switch, the following function has been created:

Function	Function Description	Return
des_MC10XS3412_SendCommand()	Write MC10XS3412 register and read Serial Output response	Void

Parameter	Description
UINT8 u8XS10_Command	Configuration Command* (Use register addresses)
UINT8 u8XS10_Data	Configuration Data

Macros have been pre-defined for user convenience. Direct write to the registers can be accomplished by using these labels in commands:

Register Addresses	
XS_STATR	Status Register
XS_PWMR	Output PWM Control Register
XS_CONFR0	Output Configuration Register 0
XS_CONFR1	Output Configuration Register 1
XS_OCR	Output Over Current Register
XS_GCR	Global Configuration Register
XS_CALR	Calibration Register

Depending on the register to be accessed, the user may need to include some output selection bits as described in Table 3, or additional control bits for special configurations (28W_s, PWM register; Xenon_s, Over-current register; V_{DD}_Fail_en, and Global Configuration register).

As an example, the following line uses the des_MC10XS3412_SendCommand() function:

```
des_MC10XS3412_SendCommand(XS_STATR, 0); // Function to write a "0" in the status register
```

6 Watchdog Refreshing

As long as the watchdog bit (D15) of an incoming SPI message is toggled within the minimum watchdog timeout period (WDTO), the device will operate normally.

Anytime a command is sent using the function des_MC10XS3412_SendCommand(), the watchdog bit is toggled. However, the watchdog can be cleared when needed, by using the following function:

Function	Function Description	Return
des_MC10XS3412_ClrWDT()	Clear MC10XS3412 watchdog	Void

The watchdog can be disabled by hardware, and also by grounding the RFSI pin.

7 Device Initialization

In this section, a proposed initial configuration of the MC10XS3412 is explained and how register values are modified accordingly, to get the eXtreme Switch operational. However, they can be changed depending on the specific application requirements. Use the macro declarations to set the corresponding bits on the registers from the file des_MC10XS3412.h, if required.

Table 2. Initial Values

MC10XS3412 default configuration at file des_MC10XS3412.h	
XS10_PWMR_0_INIT	HS0: Output switch OFF; PWM duty cycle=Fully ON
XS10_PWMR_1_INIT	HS1: Output switch OFF; PWM duty cycle =Fully ON
XS10_PWMR_2_INIT	HS2: Output switch OFF; PWM duty cycle =Fully ON
XS10_PWMR_3_INIT	HS3: Output switch OFF; PWM duty cycle =Fully ON
XS10_CONFR0_0_INIT	HS0: Direct Control disabled; Low Slew Rate; No delay
XS10_CONFR0_1_INIT	HS1: Direct Control disabled; Low slew rate; No delay
XS10_CONFR0_2_INIT	HS2: Direct Control disabled; Low slew rate; No delay
XS10_CONFR0_3_INIT	HS3: Direct Control disabled; Low slew rate; No delay
XS10_CONFR1_0_INIT	HS0: Auto-retry disabled; output hard shorted to VPWR disabled; ON output open load detection enabled for LEDs; OFF output open load detection disabled; Low ratio for CSNS
XS10_CONFR1_1_INIT	HS1: Auto-retry disabled; output hard shorted to VPWR disabled; ON output open load detection enabled for LEDs; OFF output open load detection disabled; Low ratio for CSNS
XS10_CONFR1_2_INIT	HS2: Auto-retry disabled; output hard shorted to VPWR disabled; ON output open load detection enabled for LEDs; OFF output open load detection disabled; Low ratio for CSNS
XS10_CONFR1_3_INIT	HS3: Auto-retry disabled; output hard shorted to VPWR disabled; ON output open load detection enabled for LEDs; OFF output open load detection disabled; Low ratio for CSNS
XS10_OCR_0_XENINIT	HS0: Xenon enabled (bulb over-current profile disabled)
XS10_OCR_1_XENINIT	HS1: Xenon enabled (bulb over-current profile disabled)
XS10_OCR_2_XENINIT	HS2: Xenon enabled (bulb over-current profile disabled)
XS10_OCR_3_XENINIT	HS3: Xenon enabled (bulb over-current profile disabled)
XS10_OCR_0_INIT	HS0: Only Inrush current management; Inrush curve fast
XS10_OCR_1_INIT	HS1: Only Inrush current management; Inrush curve fast
XS10_OCR_2_INIT	HS2: Only Inrush current management; Inrush curve fast
XS10_OCR_3_INIT	HS3: Only Inrush current management; Inrush curve fast
XS10_GCR_INIT	PWM module enabled with internal clock; CSNS tri-stated; Over-voltage protection disabled

By simply calling the following function, all commands needed for device initialization are automatically sent to the device:

Function	Function Description	Return
des_MC10XS3412_Config()	MC10XS3412 initial configuration with values from des_MC10XS3412.h	Void

This function writes the initial values over the PWM control (PWMR), Configuration (CONFR0 & CONFR1), and Over-current registers (OCR) for each HS output, as well as the initial value for the Global Configuration register (GCR), as described in Table 2.

For further details on this function and the steps required for the device initialization, open the CodeWarrior project file and examine des_MC10XS3412.h and des_MC10XS3412.c

8 Changing Operation Mode

The following function is used to change between Sleep and Normal mode by simply generating a wake-up event; specifically, the function writes a logic ‘1’ to the RST pin, if the Normal mode is requested. Otherwise, it will go into Sleep mode, the function sets all the INx and RST pins to ‘0’. Fault and Fail Safe modes are accessed automatically when the corresponding conditions occur. Call this function, including the desired operation mode.

Function	Function Description	Return
des_MC10XS3412_SetMode()	Set MC10XS3412 operation mode	Void

Parameter	Description
UINT8 **u8XS10_Mode**	Desired operation mode for MC10XS3412
XS_SLEEP	Sleep mode
XS_NORMAL	Normal mode

9 Output Configuration

This function allows the user to enable direct control and choose the slew rate for the selected output, as well as the delay when using the PWM feature. These settings are selected in the Configuration Register.

Function	Function Description	Return
des_MC10XS3412_HS_Configuration()	Outputs configuration	Void

Table 3. Parameters for HS Outputs configuration

Parameter	Description
UINT8 u8HSout	Selected HS output channel
XS_HS0	Output HS0
XS_HS1	Output HS1
XS_HS2	Output HS2
XS_HS3	Output HS3
UINT8 u8Dir_Control	Direct IN control
XS_DIR_EN	Direct control enabled
XS_DIR_DIS	Direct control disabled (the output is only controlled by ON bit of the PWM register)
UINT8 u8SlewRate	Slew rate speed selection
XS_SLEW_LOW	Low slew rate
XS_SLEW_MED	Medium slew rate
XS_SLEW_HIGH	High slew rate
UINT8 u8SwitchDelay	Output PWM switching delay
XS_NO_DELAY	No delay
XS_DELAY_16	16 PWM clock periods
XS_DELAY_32	32 PWM clock periods
XS_DELAY_48	48 PWM clock periods
XS_DELAY_64	64 PWM clock periods
XS_DELAY_80	80 PWM clock periods
XS_DELAY_96	96 PWM clock periods
XS_DELAY_112	112 PWM clock periods

The following line is an example of its use:

```
des_MC10XS3412_HS_Configuration(XS_HS0,XS_DIR_EN,XS_SLEW_LOW,XS_NO_DELAY);
```

10 Output Control

This function will enable the user to control the state of the selected output by choosing the XS_ON or XS_OFF parameters for the ON bit on the PWM register. The PWM duty cycle can also be selected (independently selected for each output).

Function	Function Description	Return
des_MC10XS3412_HS_Control()	Outputs control	Void

This function uses the following parameters:

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8HS_State	Sets the output state (ON bit)
XS_ON	Enable the corresponding output switch
XS_OFF	Turns the corresponding output switch OFF (if the IN input is also pulled down)
UINT8 u8HS_PWM	Output PWM duty cycle, 0x00 to 0x7F

An example for this function is shown by the following:

```
des_MC10XS3412_HS_Control(XS_HS0, XS_ON, PWM_MAX);
```

11 PWM Module Selection

With this function the user can enable the PWM module and select a clock source.

Function	Function Description	Return
des_MC10XS3412_Select_PWM_Module()	PWM module selection	Void

This function uses the following parameters:

Parameter	Description
UINT8 u8HS_Select	PWM module selection
XS_PWM_DISAB LED	PWM module disabled
XS_PWM_IN0CLK	PWM module enabled with external clock from IN0
XS_PWMINTCLK	PWM module enabled with internal calibrated clock

This function is typically called just once in the application code.

12 Select Open Load Detection

As part of the diagnostic functions, it is possible to detect an open load condition on an HS output, either in ON or OFF state. Select the desired characteristics for the open load detection features with the following function:

Function	Function Description	Return
des_MC10XS3412_OpenLoad_Detect()	Select open load detection	Void

The function uses the following parameters:

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8OLselect	Open load selection
XS_NO_LOAD	ON output open load detection disabled (if used, the ON output open load detection is disabled for both LEDs and bulbs)
XS_LED	Enabled for LEDs
XS_BULB	Enabled for bulbs
XS_OLOFF_EN	OFF output open load enabled
XS_OLOFF_DIS	OFF output open load disabled

This function is typically called just once in the application code.

13 Select Current Sense Ratio

If using the output current monitoring feature*, call the next function to select the current sense ratio for the selected output.

Function	Function Description	Return
des_MC10XS3412_Select_CurrentSense()	Select current sense ratio	Void

This function uses the following parameters:

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8SenseRatio	Current sense ratio selection
XS_CSNS_LOW	Low ratio for the CSNS pin for the corresponding output
XS_CSNS_HIGH	High ratio for the CSNS pin for the corresponding output

*This feature must be enabled in the Global Configuration Register.

14 Select Current Profile

The over-current protection is composed of eight predefined current levels (time dependent) to fit the Xenon-HID inrush characteristics, or 55 or 28 W bulb profiles, selectable separately by the Xenon bit and the 28W bits (last one selectable in the PWM register).

The over-current protection can also be fitted to inrush currents, programmable with the OC[1:0] bits select over-current slope speed.

In Normal mode, using the internal PWM module, the 10XS3412 incorporates a cooling bulb filament management.

To configure the proper over-current profile for each output, use the following function:

Function	Function Description	Return
des_MC10XS3412_Select_CurrentProfile()	Select current profile	Void

Selecting the appropriated parameters for the function from the table below:

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8Xenon	Bulb over-current profile selection
XS_XENON_ENABLED	Bulb over-current profile selection enabled
XS_XENON_DISABLED	Bulb over-current profile selection disabled
UINT8 u8CoolCurve	Bulb cooling curve selection (if the OC_mode and Xenon are set to logic [1])
XS_BULB_COOL_LOW	Curve speed slow
XS_BULB_COOL_MED	Curve speed medium
XS_BULB_COOL_HIGH	Curve speed fast
UINT8 u8InrushCurve	Inrush curve selection (OC[1:0] bits)
XS_INRUSH_LOW	Curve speed slow
XS_INRUSH_MED	Curve speed medium
XS_INRUSH_HIGH	Curve speed fast
XS_INRUSH_VSLOW	Curve speed very slow

The following is an example of this function:

```
des_MC10XS3412_Select_CurrentProfile(XS_HS3 | XS_XENON_ENABLED | XS_BULB_COOL_LOW |
XS_INRUSH_HIGH)
```

15 Select Output Steady State

In steady state, the wire harness will be protected by the OCLO2 current level by default. Three other DC over-current levels are available: OCLO1, or OCLO3, or OCLO4. Select the desired level by using the following function:

Function	Function Description	Return
des_MC10XS3412_Select_SteadyState()	Select output steady state	Void

The function can be configured using the following parameters:

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8SteadyState	Output steady state selection
XS_STEADY_OCL0	Current level 3
XS_STEADY_OCL1	Current level 2
XS_STEADY_OCL2	Current level 4
XS_STEADY_OCL3	Current level 1

16 Select Over-current Mode

This function allows selecting the over-current mode, as described by the following:

Function	Function Description	Return
des_MC10XS3412_Select_OverCurrent()	Select over-current mode	Void

Parameter	Description
UINT8 u8HSout	Selected HS output channel (same as Table 3)
UINT8 u8OCmode	Over-current mode selection
XS_OC_INRUSH_ONLY	Only inrush current management
XS_OC_INRUSH_COOLING	Inrush current and bulb cooling management

17 Read Status Register

This function is available for monitoring any High Side output. When the CS_s pin is pulled low, the output register is loaded. The first sixteen bits of data clocking out of the SO, and following a CS transition, are dependent upon the previously written SPI word. Therefore, this function sends a command to the Status register with the selected channel, so that the content is returned on the SO pin.

Type	Name	Function Description	Return	
UINT 8	des_MC10XS3412_Get_Status()	Read STATUS register for an output channel	U8XS10_Frame[1]	Value containing the fault register flags

Parameter	Description
UINT8 u8HSout	Selected HS channel for reading
XS_HS0_RD	Channel HS0 selected
XS_HS1_RD	Channel HS1 selected
XS_HS2_RD	Channel HS2 selected
XS_HS3_RD	Channel HS3 selected

An example of calling this function is below:

```
STATUS_MC10X3412 = des_MC10XS3412_Get_Status(XS_HS1_RD);
```

18 "C" Code Usage Examples

This section provides examples of common tasks in automotive body lighting applications that can be accomplished using the eXtreme Switches. The following examples show the steps and SPI commands required to accomplish each task.

18.1 Direct Output Control

In this example, power outputs follow the microcontroller pins' logic level. In other words, the status of the lamp is given by the logic level of INx of the eXtreme Switch. If lamp dimming is required, a PWM timer must be used as a resource from the microcontroller.

To configure an HS output as Direct Control, use the Output Configuration function (See Output Configuration) including 'XS_DIR_EN'.

An example can be found below:

```
// Configure HS0 as direct control
des_MC10XS3412_HS_Configuration(XS_HS0,XS_DIR_EN,XS_SLEW_LOW,XS_NO_DELAY);
```

From the previous code example, the HS outputs are directly controlled by the INx pins, which are connected to the microcontroller as described in Table 1. Using those macros the user can directly turn ON and OFF the corresponding output:

```
//Control variable
INT ctrl;
if (ctrl)
{
IN1_MC10XS3412 = 1;
}
else
{
IN1_MC10XS3412 = 0;
}
```

An external PWM signal generated by the microcontroller can be applied directly at the INx pin for dimming purposes.

18.2 Using Self-PWM

The eXtreme Switch includes a Self-PWM feature that alleviates resources from the microcontroller. An internal oscillator with a programmable duty cycle generates the pulse width modulation. The following is an example of how the internal PWM can be used:

a) Select a source clock to enable the PWM module

```
/* Internal CLK selected*/
des_MC10XS3412_Select_PWM_Module(PWM_INTCLK)
```

b) Use a control variable for the duty cycle

```
/* PWM period (0x00-0x7F) to be sent to eXtreme Switch */
UINT8 ES_PWM = 0;
```

c) Configure HS0 as Self-PWM control

```
/* Configure HS0 as Self-PWM control */
des_MC10XS3412_HS_Configuration(XS_HS0,XS_DIR_DIS,XS_SLEW_LOW,XS_NO_DELAY);
```

d) Use the control function with the PWM duty cycle

```
/* Program PWM duty cycle into eXtreme Switch */
des_MC10XS3412_HS_Control(XS_HS0, XS_ON, ES_PWM);
```

For this purpose no PWM timer resources are required from the microcontroller.

18.3 Protections and Diagnostics

The eXtreme Switch family can protect itself and diagnose several failure modes in the output stage, power supply, and communication with the microcontroller. In this specific example, the open load in the ON state and open load in the OFF state are diagnosed.

The user can find the following steps for the basic open load detection:

a) If there's any fault, the FS pin will be on Low state,

```
if (FS_MC10XS3412 == 0)
{
     /* A fault has occurred */
}
```

b) It's possible to monitor any HS output by reading the Status Register when using the Get_Status functions as is shown by the following:

```
/* eXtreme Switches Status Variable */
UINT8 stat;
stat = des_MC10XS3412_Get_Status(XS_HS0_RD);
```

c) For this example, the variable "stat" has the Flags' state of the status register corresponding to the selected HS output. For this example the Open load faults have been identified:

```
/** isolating the desired fault flags**/
stat = stat & (XS_FAULT_OLON|XS_FAULT_OLOFF);
/* checking for set flags: */
switch (stat)
{
case XS_FAULT_OLON:
{
     /* An Open load when ON fault has occurred */
     break;
}

case XS_FAULT_OLOFF:
{
     /* An Open load when OFF fault has occurred */
     break;
}

default:
{
     /* Other fault */
     break;
}
}//f-switch
```

18.4 Using the Current Sense feature

For this example, HS0 has been configured as Self-PWM. The CSNS pin will report a current proportional to light intensity on HS0.

Find the development in the following example:

a) Set HS0 as self-PWM control

```
UINT8 ES_PWM = 0; /* Variable containing duty-cycle */
des_MC10XS3412_HS_Configuration(XS_HS0,XS_DIR_DIS,XS_SLEW_LOW,XS_NO_DELAY);
```

b) Set current sense feature ONLY ONCE!

```
des_MC10XS3412_Select_CurrentSense(XS_HS0,XS_CSNS_LOW);
```

c) Program PWM duty cycle into eXtreme Switch

```
des_MC10XS3412_HS_Control(XS_HS0, XS_ON, ES_PWM);
```

Measuring the voltage at the CSNS pin, a mirror of the instantaneous current flowing through HS0 is visible.

From the previous example, measuring the voltage at the CSNS pin, a mirror of the instantaneous current flowing through HS0 is visible.

> Note: In many Medium and High End microcontrollers, as the MC9S12XE used in this example, the analog to digital converter (ADC) includes features of automatic compare interrupts and low power modes schemes. Either external trigger control signals (ETRIGx), or setting ADC input channels to give an interrupt after certain value, is recommended to synchronize the CSNS function coming from eXtreme Switch. By using this feature, it is not necessary to poll the analog value at the CSNS pin, alleviating microcontroller processing and reducing average power consumption. This feature, along with the ADC internal clock source, can also be used in low power modes to set Current Sensing as a wake up source for the microcontroller.

19 Conclusions

This document has presented examples for configuring the eXtreme Switch Gen III. The examples were developed as drivers that send the proper SPI commands to the eXtreme Switch Registers, in order to configure different functions, change the operation mode, and implement diagnostics, all using an MC9S12XEP100 16-bit microcontroller.

The examples shown can be migrated to different platforms easily by applying minor changes to the device drivers. Different configurations of exposed functions can be implemented by selecting predefined macros.

For further information about eXtreme Switch applications and data, follow the links in the References section.

20 References

1. MC10XS3412 DATASHEET MC10XS3412
2. MC15XS3400 DATASHEET MC15XS3400
3. MC35XS3400 DATASHEET MC35XS3400
4. MC10XS3435 DATASHEET MC10XS3435
5. MC9S12XEP100 DATASHEET MC9S12XEP100RMV1
6. AN3569 EMC, ESD and Fast Transient Pulses Performances AN3569
7. AN3740 Evaluation of Power Dissipation for the eXtreme Switch Devices AN3740
8. AN2467 Power Quad Flat No-Lead (PQFN) Package AN2467

21 Source Code

All code snippets and examples shown in this application note are available as a CodeWarrior project. To get the latest version please visit Freescale.com or contact Technical Support.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

AN3831
Rev. 1.0
3/2009

freescale™ semiconductor