

Using TRGMUX on KL28 Based on SDK2.0

1. Introduction

The TRGMUX, a new module of Kinetis L series MCU, is introduced in this application note. The related API, based on SDK2.0, is also introduced. Finally, two examples of using TRGMUX to connect TPM to ADC, and to connect to DMA and LPSPI, are explained.

Contents

1.	Introduction	1
2.	TRGMUX module.....	2
3.	Related API in SDK2.0 introduction	3
3.1.	API from TRGMUX.....	3
3.2.	API from target module.....	3
4.	Example – Connecting TPM to ADC via TRGMUX.....	5
5.	Example – Connecting TPM to DMA + LPSPI via TRGMUX.....	6
6.	Revision history	8



2. TRGMUX module

The Trigger MUX module (TRGMUX) enables software to configure the trigger inputs for various peripherals. The structure of TRGMUX is shown in the following figure. As the figure shows, TPM0 is the trigger source, after moving through TRGMUX, it can be routed to output for DMA and ADC. TRGMUX must be configured correctly by setting the registers in TRGMUX. TRGMUX features 1 to 4 outputs for a peripheral. For different peripherals, the meaning of the output is not the same. TRGMUX provides 4 trigger output for DMA, which are connected separately to DMA channels 0 to 3. It provides two trigger outputs for ADC, which go separately to the ADC hardware trigger select signal ADHWTSa and ADHWTSb. The trigger source of each output can be set in the TRGMUX_Peripheral [SELn] register.

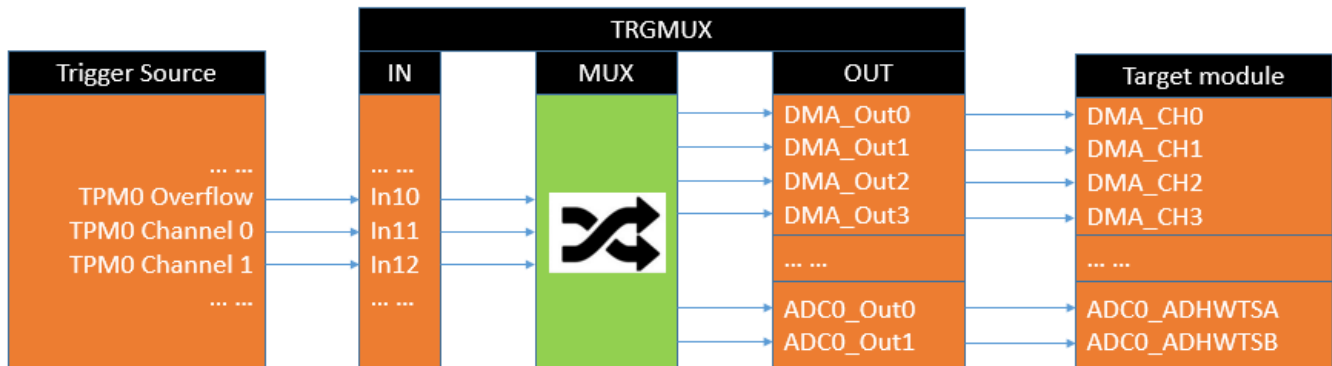


Figure 1. TRGMUX structure

3. Related API in SDK2.0 introduction

To implement an application on the TRGMUX based on SDK2.0, the API must be called from the TRGMUX driver. The related API for the target module, which is to be triggered to enable the trigger function, typically also needs to be called.

3.1. API from TRGMUX

The API from TRGMUX is defined as:

```
status_t TRGMUX_SetTriggerSource(TRGMUX_Type *base,
                                trgmux_device_t index,
                                trgmux_trigger_input_t input,
                                trgmux_source_t trigger_src);
```

The first parameter specifies the TRGMUX module currently being used. The second parameter specifies the target module which is to be triggered. The third parameter specifies the trigger index, as for some modules there are more than one TRGMUX output. The trigger source for each TRGMUX output can be set with this parameter. The final parameter specifies the trigger source for the specified TRGMUX output channel.

A typical usage example is:

```
TRGMUX_SetTriggerSource(TRGMUX0,
                        kTRGMUX_Trgmux0Adc0,
                        kTRGMUX_TriggerInput0,
                        kTRGMUX_SourceTpm0Ch1);
```

In this example, TRGMUX0 is used and ADC is the target module which is to be triggered. It is triggered by the first TRGMUX output channel for ADC which goes to ADHWTSA and the trigger source is TMP0 CH1.

3.2. API from target module

The related API is not the same with a different target module. The examples of ADC and DMA as the target module are used here.

For ADC, the API to enable the trigger function is:

```
static inline void ADC16_EnableHardwareTrigger(ADC_Type *base, bool enable);
```

Example – Connecting TPM to ADC via TRGMUX

It is typically called as follows:

```
ADC16_EnableHardwareTrigger(ADC0, true;
```

For DMA, there are two related APIs to configure DMAMUX:

```
static inline void DMAMUX_EnablePeriodTrigger(DMAMUX_Type *base, uint32_t channel);  
static inline void DMAMUX_SetSource(DMAMUX_Type *base, uint32_t channel, uint8_t source);
```

It typically needs to be called as follows:

```
DMAMUX_EnablePeriodTrigger(DMAMUX0, 0);  
DMAMUX_SetSource(DMAMUX0, 0, REQUEST_SOURCE_ALWAYS_ENABLE);
```

4. Example – Connecting TPM to ADC via TRGMUX

This example is based on the driver example `tpm_simple_pwm` and `adc16_polling` for FRDM-KL28. For ADC to work correctly, the trigger source which can provide both trigger and pre-trigger must be selected. Otherwise, ADC will not work.

For example, when using TPM as the trigger source, the TPM overflow does not provide a pre-trigger, therefore it cannot be the trigger source for ADC. In this example TPM CH1 is selected as the trigger source.

The key code is:

```
ADC16_EnableHardwareTrigger(DEMO_ADC16_BASE, true);

TRGMUX_SetTriggerSource(TRGMUX0,
                        kTRGMUX_Trgmux0Adc0,
                        kTRGMUX_TriggerInput0,
                        kTRGMUX_SourceTpm0Ch1);
```

There are some changes in this example. For example, it reads the light sensor output and shows it in the console. After the demo code is running, press 1 to go into the TPM→ADC example, as shown in the following figure:

```
TRGMUX demo on FRDM-KL28.
1. Run example TPM -> ADC
2. Run example TPM -> DMA+LPSPI
1
ADC16_DoAutoCalibration() Done.
Press any key to continue.
ADC Value: 57581
Press any key to continue.
ADC Value: 58604
Press any key to continue.
ADC Value: 3695
Press any key to continue.
ADC Value: 3074
Press any key to continue.
ADC Value: 2993
Press any key to continue.
ADC Value: 58018
Press any key to continue.
ADC Value: 57751
Press any key to continue.
```

Figure 2. TRGMUX demo – Trigger ADC by TPM

In the test, when shining a light source, such as the flashlight from a mobile phone, onto the FRDM-K28F board, then switching the light back off, the ADC output result changes accordingly. See the attached code for more details.

5. Example – Connecting TPM to DMA + LPSPI via TRGMUX

In this example, LPSPI is working together with DMA, and DMA is triggered by TPM CH1. It is based on the driver example `tpm_simple_pwm` and `lpspi_edma_transfer` for FRDM-KL28.

The LPSPI pin assignment is from PTD0 to PTD3, and the signal can be measured on PIN6 to PIN12 on J2, as shown in the following figure. The logic analyzer or scope can be used to capture the signals for analysis.

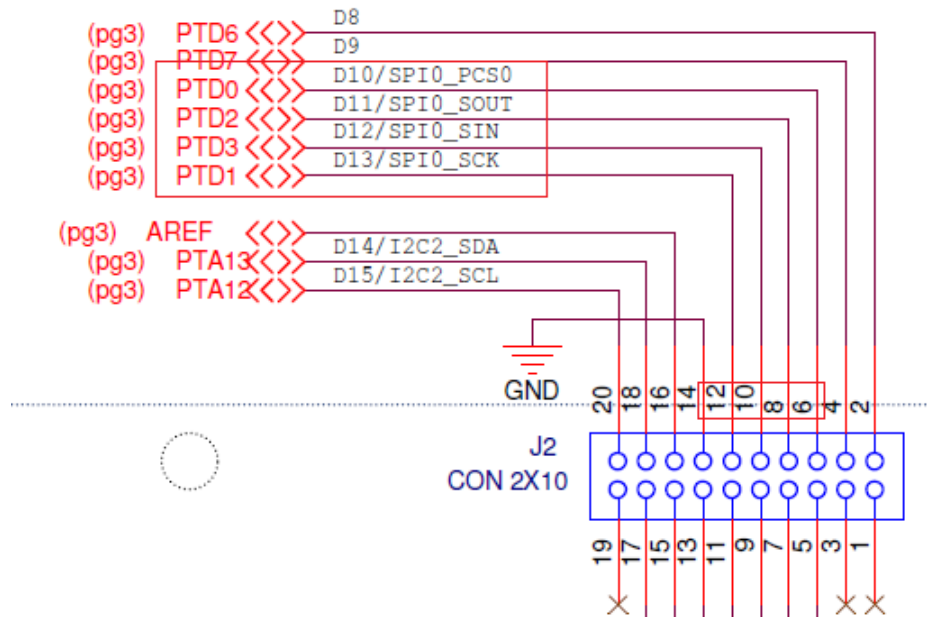


Figure 3. LPSPI pin assignment

When implementing this example, follow these four key points:

1. Only DMA channel 0 to channel 3 can be connected to TRGMUX
2. Enable the period trigger for the DMA channel which was selected to connect to TRGMUX
3. Set the DMAMUX source to always be enabled for the channel used
4. Set the trigger source to the required source, in this example it is TPM0 CH1

The corresponding key code is:

```
DMAMUX_EnablePeriodTrigger(EXAMPLE_LPSPI_MASTER_DMAMUX_BASEADDR, masterTxChannel);
DMAMUX_SetSource(EXAMPLE_LPSPI_MASTER_DMAMUX_BASEADDR,
                 masterTxChannel,
                 (uint8_t)kDmaRequestMux0AlwaysOn63);
TRGMUX_SetTriggerSource(TRGMUX0,
                       kTRGMUX_Trgmux0Dmamux0,
                       kTRGMUX_TriggerInput1,
                       kTRGMUX_SourceTpm0Ch1);
```

After the menu is displayed, press 2 to run the example. The console output is shown in the following figure:

```
TRGMUX demo on FRDM-KL28.
1. Run example TPM -> ADC
2. Run example TPM -> DMA+LPSPI
2
TPM -> DMA+LPSPI start.
PCSO --> J2.6
SOUT --> J2.8
SIN  --> J2.10
SCK  --> J2.12
```

Figure 4. Console output for example of Connect TPM to DMA + LPSPI by TRGMUX

The following figure shows the resulting LPSPI waveform after 2 has been pressed. For more details, refer to the attached code.

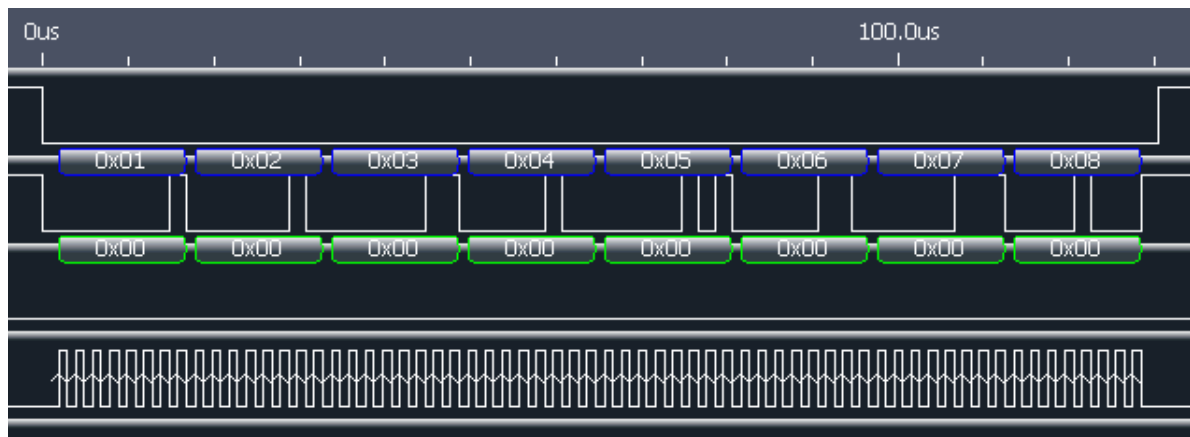


Figure 5. Logic analyzer output for example of Connect TPM to DMA + LPSPI by TRGMUX

6. Revision history

Table 1. Revision history

Revision number	Date	Substantive changes
0	01/2017	Initial release

How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, and Tower, are trademarks of NXP B.V. All other product or service names are the property of their respective owners.

ARM, the ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2017 NXP B.V.

Document Number: AN5399

Rev. 0

01/2017

