S32K3 RTD TRAINING - LCU

AUTOMOTIVE APPLICATIONS TEAM

Created:	Mar 2021
Last Update:	Mar 2021

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- LCU Overview
- RTD configuration
- Main API functions
- Example Code
- LUT Configuration Tips

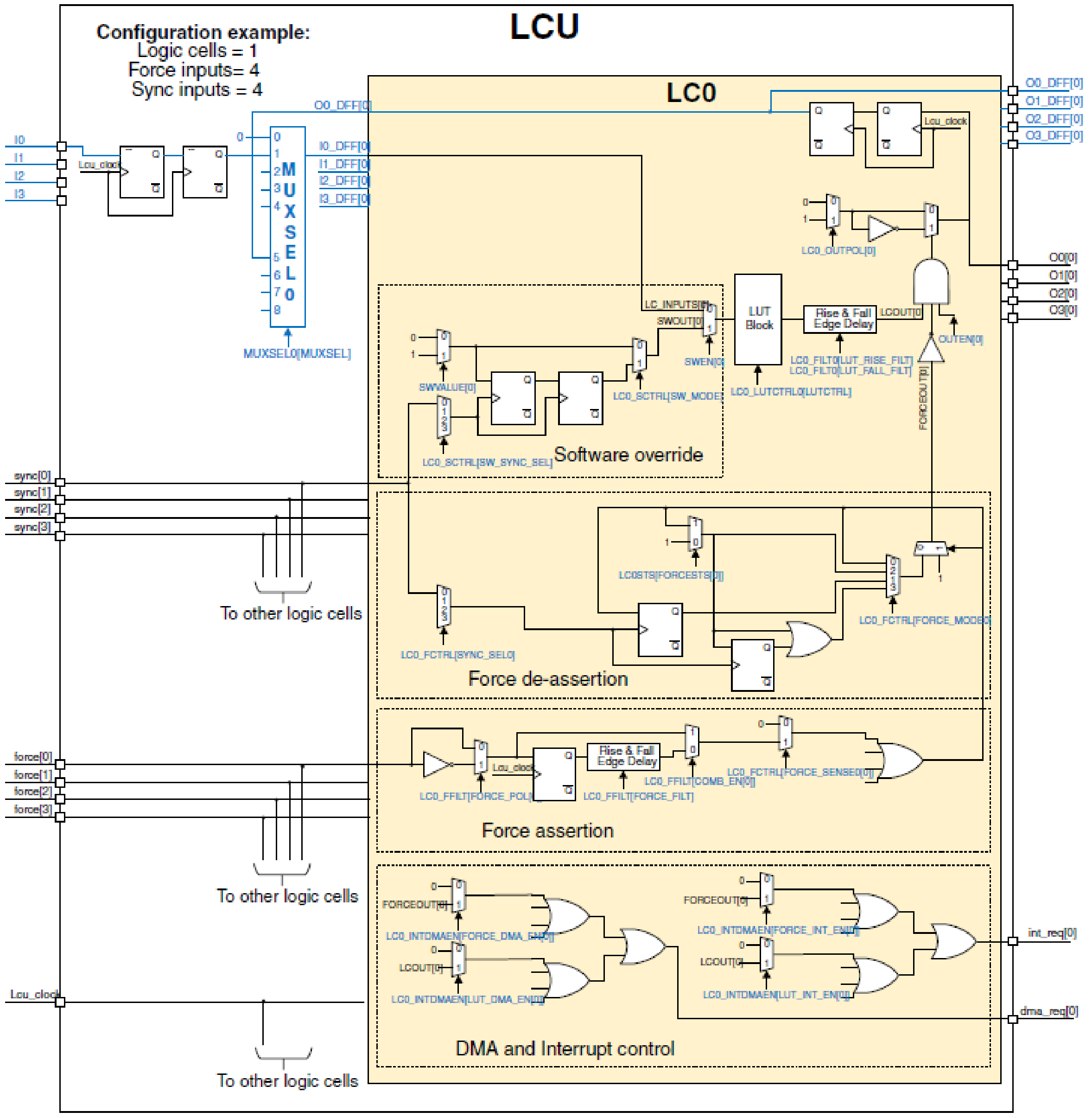
LCU OVERVIEW

S32K3xx LCU Overview

- Logic Control Unit (LCU) comprises logic cells with programmable logic that operates outside the speed limitations of software execution.
- Logic cell takes up to 4 input signals and through the Look Up Table (LUT) generates 4 output signals
- Input sources are a combination of the following:
  - I/O pins
  - Peripheral outputs
  - Register bits
- Output can be directed internally to peripherals and to an output pin.
- Combinatorial Logic:
  - Any 16 x 4 Truth Table such AND, NAND, AND-OR, AND-OR-INVERT, OR-XOR, OR-XNOR and their combinations
- Latches:
  - S-R, D-flip flop, JK-flip flop
- Advanced:
  - Incremental encoder, ACIM, PMSM & BLDC motor controllers

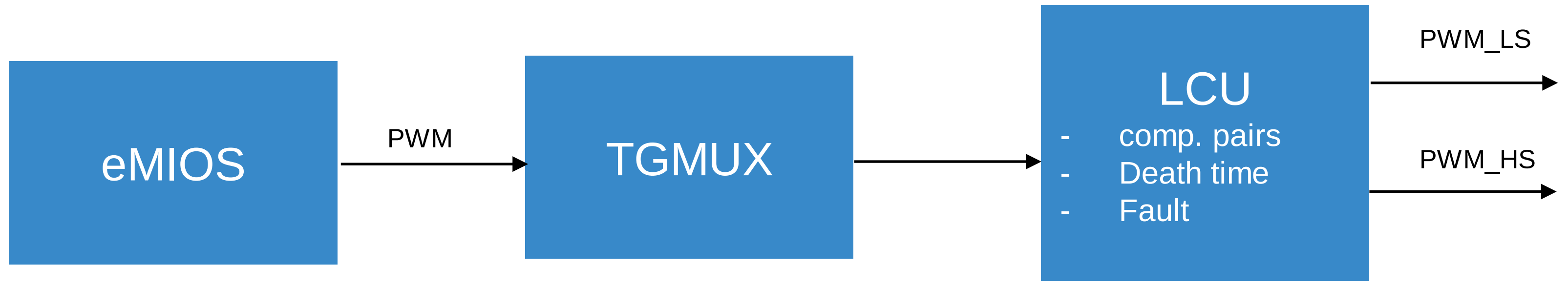
S32K3xx LCU Overview

- The LCU is used to create small combinatorial sequential logic circuits.
  - The logic operation can be programmed by software.
- **Hardware resource:**
  - Up to 6 LCs (Logic Cells)
  - Each LC contains 4 LTUs (Look-up Table) attached with 4 inputs and outputs.
  - Software override option included to allow software input to logic functions
  - Force control support for motor control / power conversion applications



LCU Example Project

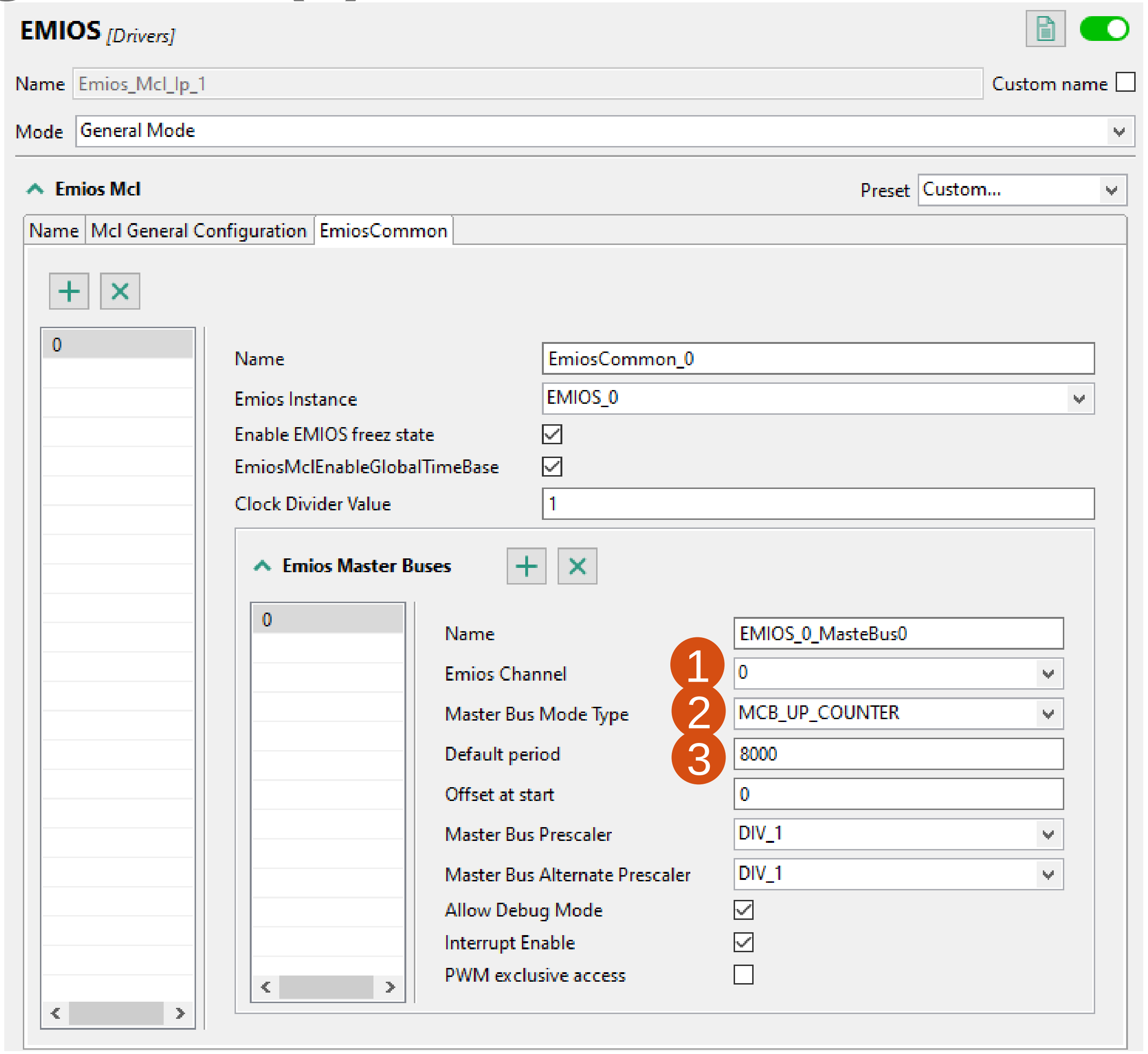
How to use the LCU module to generate a complementary pair PWM signal?



RTD CONFIGURATION

eMIOS_Mcl component Configuration(1)-Counter Bus

- eMIOS Channel: Select 0 to generate the counter bus B
- Master Bus Mode Type: Select the MCB_UP_COUNTER to generate counter bus B
- Default period: Specified the counter bus B period



EMIOS [Drivers]

Name: Emios_Mcl_Ip_1 | Custom name ☐

Mode: General Mode

Emios Mcl | Preset: Custom...

Tabs: Name | Mcl General Configuration | EmiosCommon

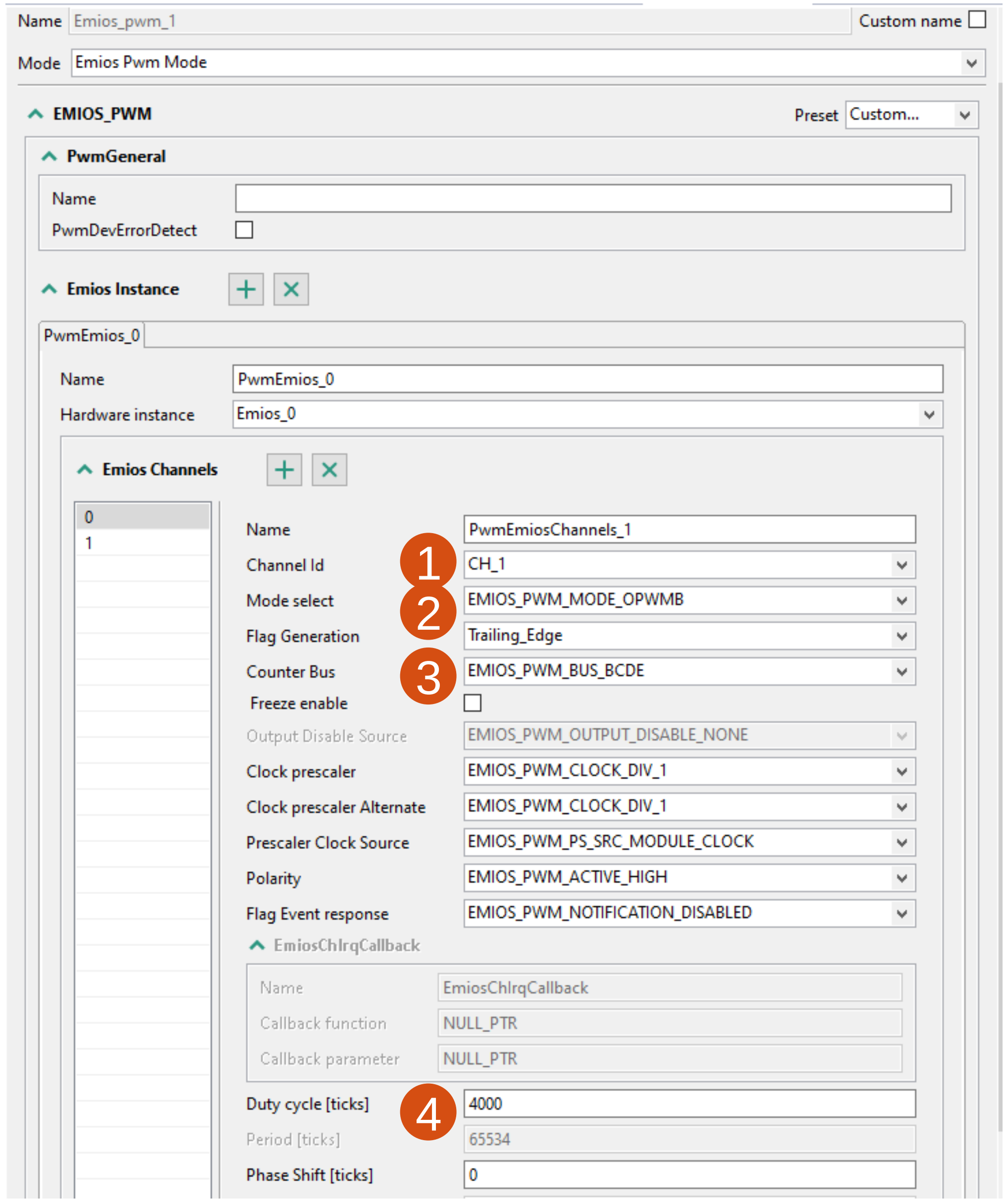
Name	EmiosCommon_0
Emios Instance	EMIOS_0
Enable EMIOS freez state	✓
EmiosMclEnableGlobalTimeBase	✓
Clock Divider Value	1

Emios Master Buses

Name	EMIOS_0_MasteBus0
Emios Channel	1 0
Master Bus Mode Type	2 MCB_UP_COUNTER
Default period	3 8000
Offset at start	0
Master Bus Prescaler	DIV_1
Master Bus Alternate Prescaler	DIV_1
Allow Debug Mode	✓
Interrupt Enable	✓
PWM exclusive access	☐

eMIOS_pwm component Configuration(1)-PWM generation

- Channel Id: Specified which channel used to PWM generation
- Mode Select: Select the OPWMB mode to generate a PWM signal
- Counter Bus: Specified the reference counter bus
- Duty Cycle: Specified the duty cycle of generated PWM signal



Name	Emios_pwm_1
Custom name	☐
Mode	Emios Pwm Mode

EMIOS_PWM — Preset: Custom...

PwmGeneral

Name	
PwmDevErrorDetect	☐

Emios Instance

PwmEmios_0

Name	PwmEmios_0
Hardware instance	Emios_0

Emios Channels

0
1

Name	PwmEmiosChannels_1
Channel Id	CH_1
Mode select	EMIOS_PWM_MODE_OPWMB
Flag Generation	Trailing_Edge
Counter Bus	EMIOS_PWM_BUS_BCDE
Freeze enable	☐
Output Disable Source	EMIOS_PWM_OUTPUT_DISABLE_NONE
Clock prescaler	EMIOS_PWM_CLOCK_DIV_1
Clock prescaler Alternate	EMIOS_PWM_CLOCK_DIV_1
Prescaler Clock Source	EMIOS_PWM_PS_SRC_MODULE_CLOCK
Polarity	EMIOS_PWM_ACTIVE_HIGH
Flag Event response	EMIOS_PWM_NOTIFICATION_DISABLED

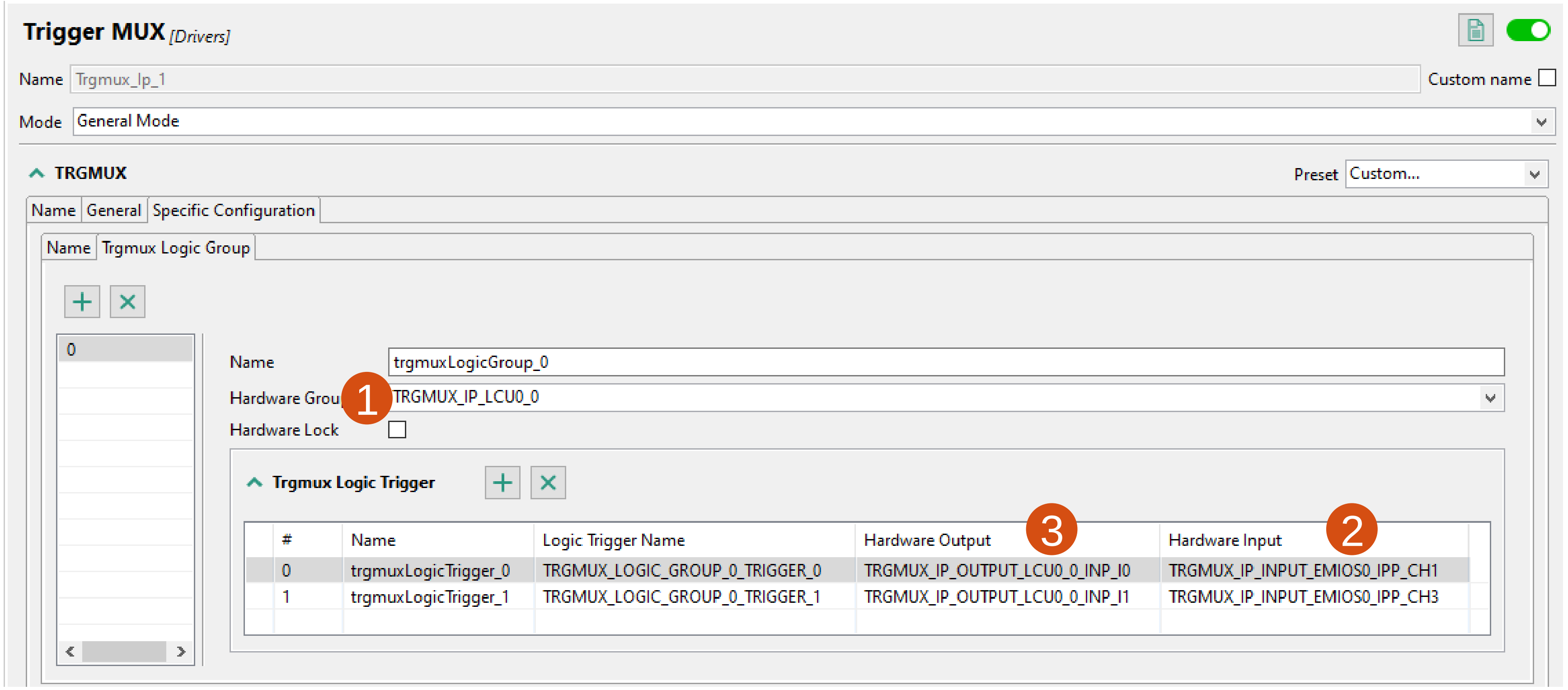
EmiosChIrqCallback

Name	EmiosChIrqCallback
Callback function	NULL_PTR
Callback parameter	NULL_PTR

Duty cycle [ticks]	4000
Period [ticks]	65534
Phase Shift [ticks]	0

Trgmux component Configuration(1)-Routing PWM to LCU

- Hardware Group: Specified the LCU group
- Hardware Input: Specified which eMIOS channel will be routed to LCU
- Hardware Output: Specified which logic cell and input connected to the eMIOS channel



Trigger MUX *[Drivers]*

Name: Trgmux_Ip_1 Custom name ☐

Mode: General Mode

TRGMUX — Preset: Custom...

Name | General | Specific Configuration

Name | Trgmux Logic Group

0

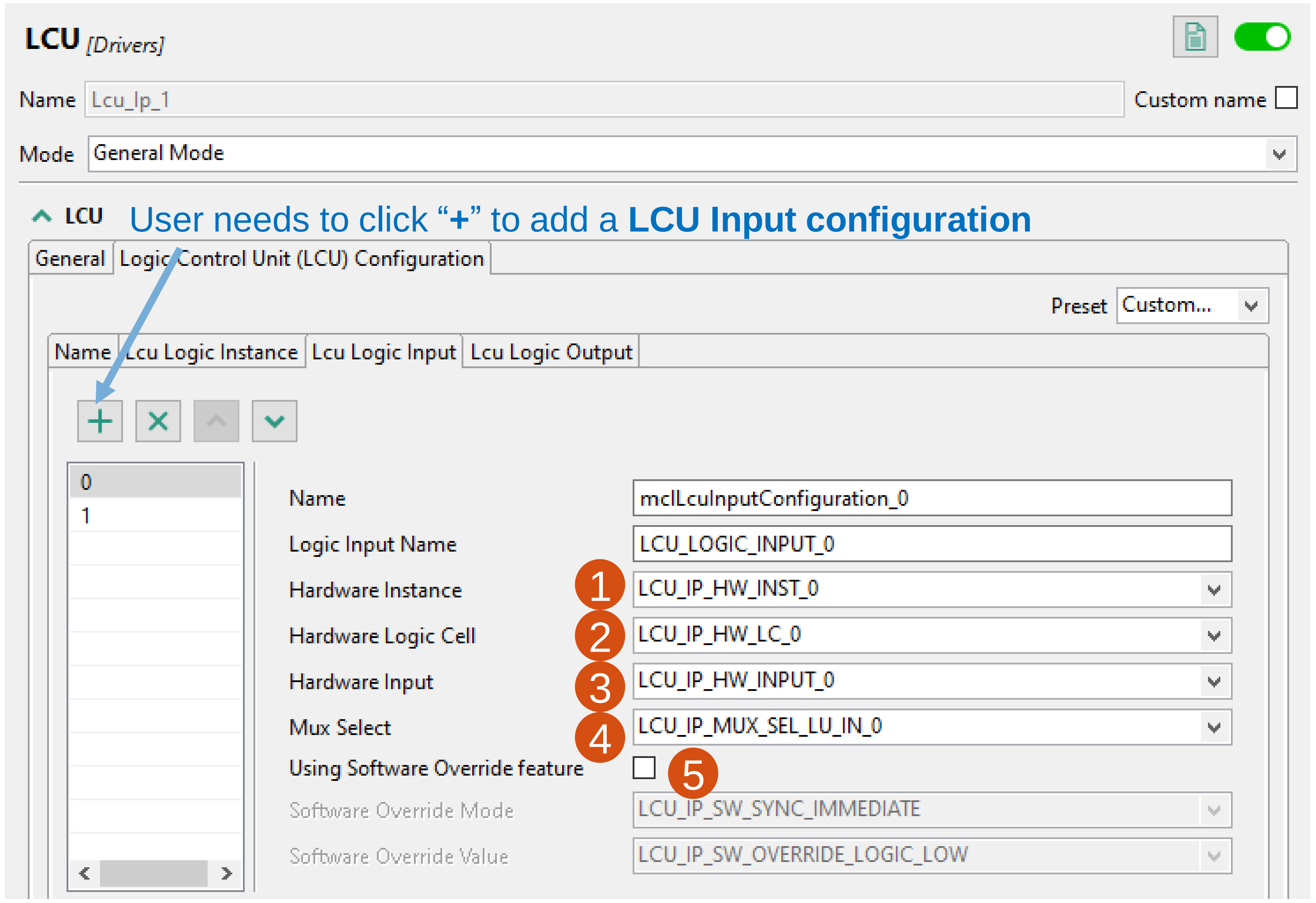
Name	trgmuxLogicGroup_0
Hardware Group (1)	TRGMUX_IP_LCU0_0
Hardware Lock	☐

Trgmux Logic Trigger

#	Name	Logic Trigger Name	Hardware Output (3)	Hardware Input (2)
0	trgmuxLogicTrigger_0	TRGMUX_LOGIC_GROUP_0_TRIGGER_0	TRGMUX_IP_OUTPUT_LCU0_0_INP_I0	TRGMUX_IP_INPUT_EMIOS0_IPP_CH1
1	trgmuxLogicTrigger_1	TRGMUX_LOGIC_GROUP_0_TRIGGER_1	TRGMUX_IP_OUTPUT_LCU0_0_INP_I1	TRGMUX_IP_INPUT_EMIOS0_IPP_CH3

LCU Component Configuration(1)-Lcu Logic Input

- LCU Hardware Module: select the LCU instance to use
  - LCU_0
  - LCU_1
- Logic Cell: select the logic cell of each LCU instance to use
  - LC_0/1/2
- Hardware Input: select the input of each logic cell
  - INPUT_0/1/2/3
- Mux Select: selects the sources for inputs to the LCs
  - SEL_LOGIC_0
  - SEL_LU_IN_0/1/2/3/4/5/6/7/8/9/10/11
- Software Override: check to enable SW override feature
  - Input software override logic to override external inputs



Name Lcu_Ip_1 Custom name ☐

Mode General Mode

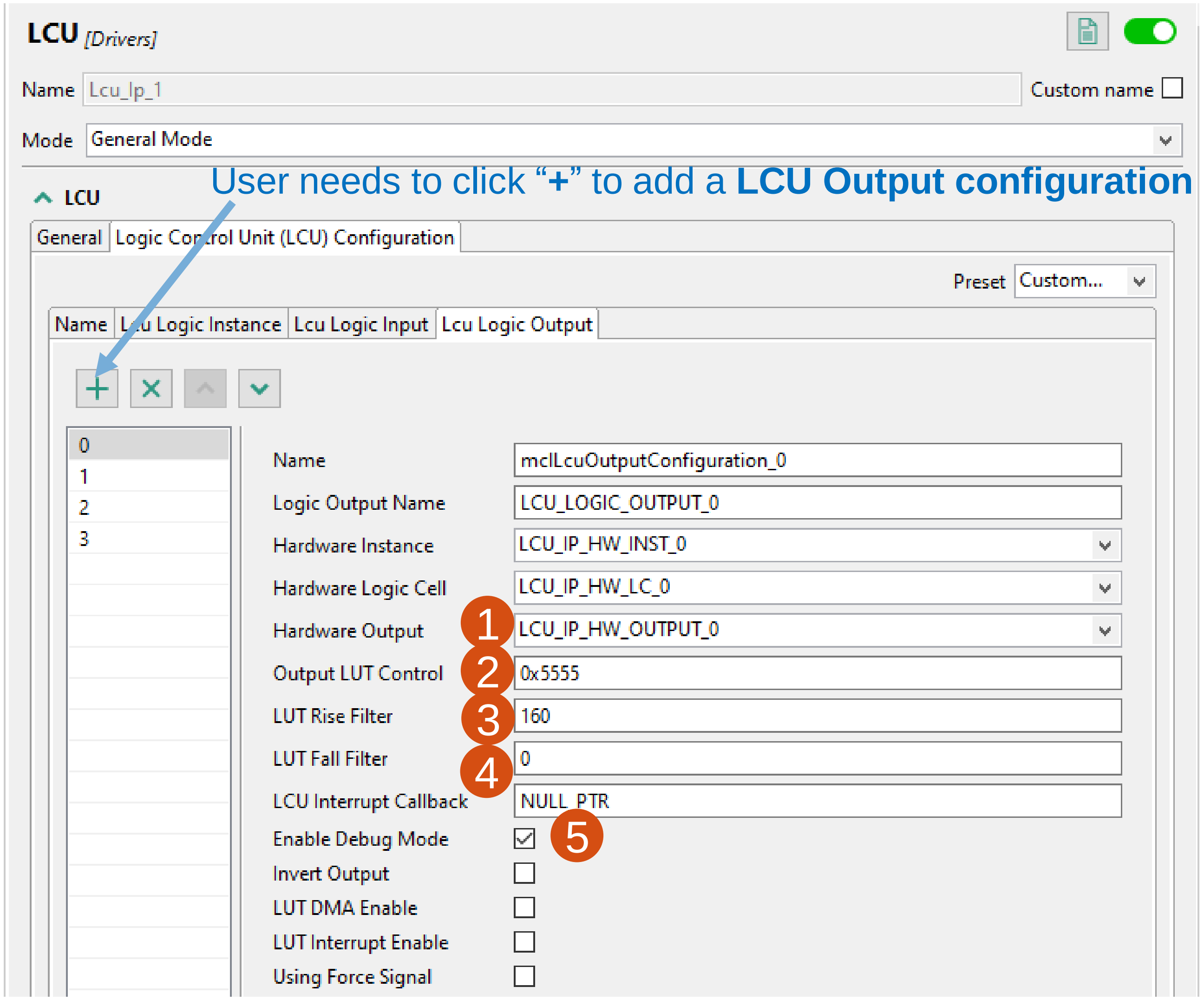
LCU User needs to click "+" to add a **LCU Input configuration**

Preset Custom...

Name	mclLcuInputConfiguration_0
Logic Input Name	LCU_LOGIC_INPUT_0
Hardware Instance	LCU_IP_HW_INST_0
Hardware Logic Cell	LCU_IP_HW_LC_0
Hardware Input	LCU_IP_HW_INPUT_0
Mux Select	LCU_IP_MUX_SEL_LU_IN_0
Using Software Override feature	☐
Software Override Mode	LCU_IP_SW_SYNC_IMMEDIATE
Software Override Value	LCU_IP_SW_OVERRIDE_LOGIC_LOW

LCU Component Configuration(2)-Lcu Logic Output

- Hardware Output: select the output of each logic cell
  - OUTPUT_0/1/2/3
- LUT Control:
  - Specifies the LUT positions, based on the combined LC input value, that result in assertion of this output.
- LUT Rise Filter: Specifies the rising edge thresholds for LC output filters
  - 0 – 65535 LCU clocks
  - Used to configure dead-time
- LUT Fall Filter: Specifies the falling edge thresholds for LC output filters
  - 0 – 65535 LCU clocks
- Enable Debug Mode: check to enable outputs to continue operation in Debug mode
  - Unchecked: Inactive
  - Checked: Continue normal operation



User needs to click "+" to add a **LCU Output configuration**

Name	Lcu_Ip_1
Mode	General Mode

Name	mclLcuOutputConfiguration_0
Logic Output Name	LCU_LOGIC_OUTPUT_0
Hardware Instance	LCU_IP_HW_INST_0
Hardware Logic Cell	LCU_IP_HW_LC_0
Hardware Output	LCU_IP_HW_OUTPUT_0
Output LUT Control	0x5555
LUT Rise Filter	160
LUT Fall Filter	0
LCU Interrupt Callback	NULL_PTR
Enable Debug Mode	☑
Invert Output	☐
LUT DMA Enable	☐
LUT Interrupt Enable	☐
Using Force Signal	☐

MAIN API FUNCTIONS

Main RTD API Functions & Usage

The LCU component provides the following main API functions

- Lcu_Ip_ReturnType **Lcu_Ip_Init**(const Lcu_Ip_InitType * const pxLcuInit)
  - It initialize the input/output configuration and should be called before invoking all other API functions.
- Lcu_Ip_ReturnType **Lcu_Ip_Deinit**(void)
  - It resets all logic cells to default.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncInputSwOverrideEnable**(const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)
  - This function is called for enable software override function.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncInputSwOverrideValue**(const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)
  - Specifies the software override value for each logic cell input.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncInputMuxSelect**(const Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)
  - Selects the source of the logic cell input.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncOutputEnable**(const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)
  - Enables logic cell outputs.

Main RTD API Functions & Usage

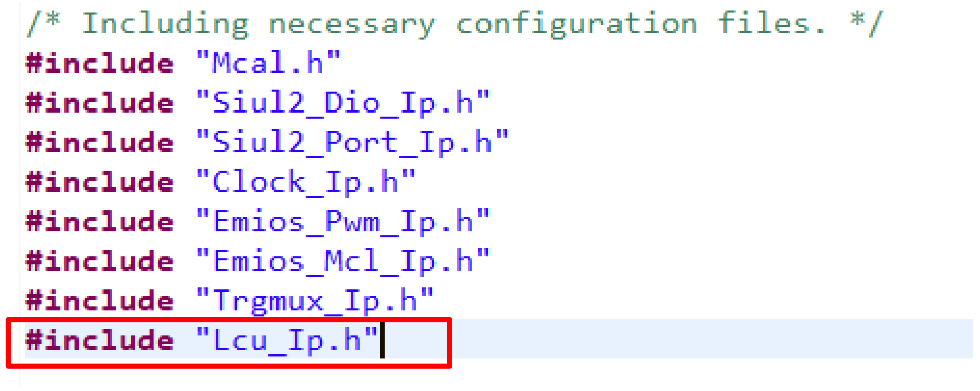
The LCU component provides the following main API functions

- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncOutputPolarity**(const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)
  - Specifies the polarity of the outputs.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncOutputFallFilter**(const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)
  - Specifies the number of consecutive clock cycles the filter output must be logic 0 before the output signal goes low.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncOutputRiseFilter**(const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)
  - Specifies the number of consecutive clock cycles the filter output must be logic 1 before the output signal goes high.
- Lcu_Ip_ReturnType **Lcu_Ip_SetSyncOutputLutControl**(const Lcu_Ip_SyncOutputValueType List[], const uint8 Dimension)
  - Specifies the LUT positions, based on the combined LC input value, that result in assertion of this output.
- Lcu_Ip_ReturnType **Lcu_Ip_GetSyncLogicInput**(Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)
  - Indicates states of logic cell inputs.
- Lcu_Ip_ReturnType **Lcu_Ip_GetSyncSwOverrideInput**(Lcu_Ip_SyncInputValueType List[], const uint8 Dimension)
  - Indicates states of logic cell inputs or software-overridden inputs, depending upon the state of the SW override function.

EXAMPLE CODE

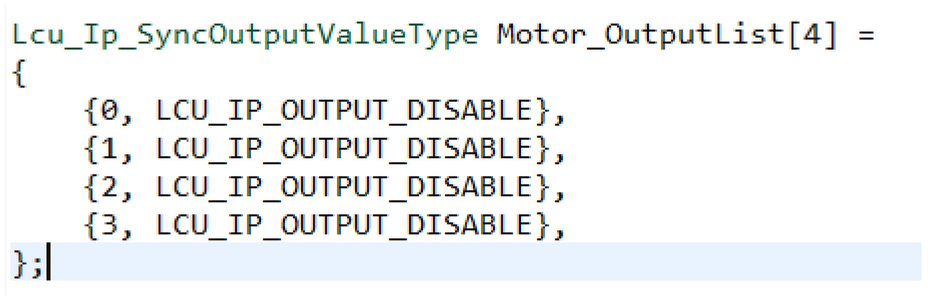
Example Project—Application Codes

- ① Include "Lcu_Ip.h"—user config data and API function declare



```c
/* Including necessary configuration files. */
#include "Mcal.h"
#include "Siul2_Dio_Ip.h"
#include "Siul2_Port_Ip.h"
#include "Clock_Ip.h"
#include "Emios_Pwm_Ip.h"
#include "Emios_Mcl_Ip.h"
#include "Trgmux_Ip.h"
#include "Lcu_Ip.h"
```

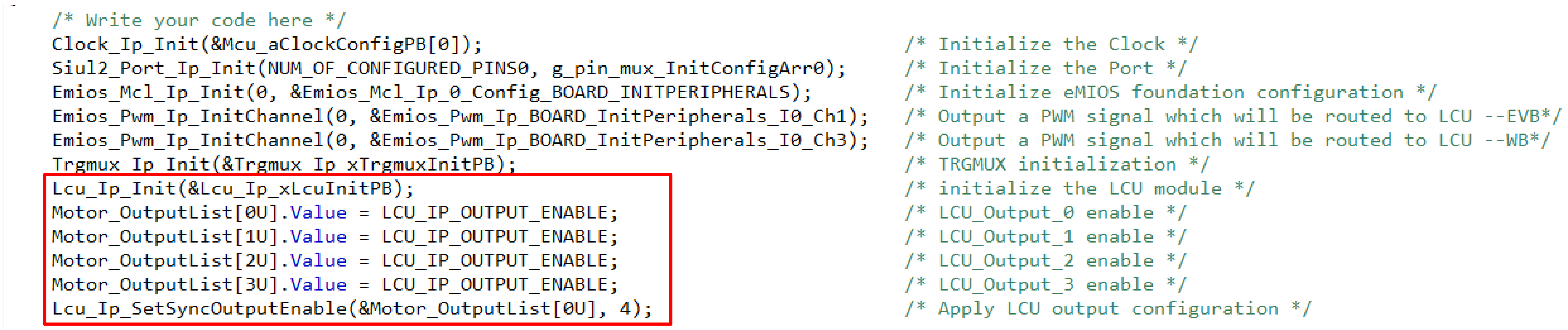
- ② Declare a **Lcu_Ip_SyncOutputValueType** variable representing the output state of the LCU



```c
Lcu_Ip_SyncOutputValueType Motor_OutputList[4] =
{
    {0, LCU_IP_OUTPUT_DISABLE},
    {1, LCU_IP_OUTPUT_DISABLE},
    {2, LCU_IP_OUTPUT_DISABLE},
    {3, LCU_IP_OUTPUT_DISABLE},
};
```

Example Project—Application Codes

- ③ Initialize the LCU module and enable the output



```
/* Write your code here */
Clock_Ip_Init(&Mcu_aClockConfigPB[0]);                                     /* Initialize the Clock */
Siul2_Port_Ip_Init(NUM_OF_CONFIGURED_PINS0, g_pin_mux_InitConfigArr0);     /* Initialize the Port */
Emios_Mcl_Ip_Init(0, &Emios_Mcl_Ip_0_Config_BOARD_INITPERIPHERALS);        /* Initialize eMIOS foundation configuration */
Emios_Pwm_Ip_InitChannel(0, &Emios_Pwm_Ip_BOARD_InitPeripherals_I0_Ch1);   /* Output a PWM signal which will be routed to LCU --EVB*/
Emios_Pwm_Ip_InitChannel(0, &Emios_Pwm_Ip_BOARD_InitPeripherals_I0_Ch3);   /* Output a PWM signal which will be routed to LCU --WB*/
Trgmux_Ip_Init(&Trgmux_Ip_xTrgmuxInitPB);                                  /* TRGMUX initialization */
Lcu_Ip_Init(&Lcu_Ip_xLcuInitPB);                                           /* initialize the LCU module */
Motor_OutputList[0U].Value = LCU_IP_OUTPUT_ENABLE;                         /* LCU_Output_0 enable */
Motor_OutputList[1U].Value = LCU_IP_OUTPUT_ENABLE;                         /* LCU_Output_1 enable */
Motor_OutputList[2U].Value = LCU_IP_OUTPUT_ENABLE;                         /* LCU_Output_2 enable */
Motor_OutputList[3U].Value = LCU_IP_OUTPUT_ENABLE;                         /* LCU_Output_3 enable */
Lcu_Ip_SetSyncOutputEnable(&Motor_OutputList[0U], 4);                      /* Apply LCU output configuration */
```

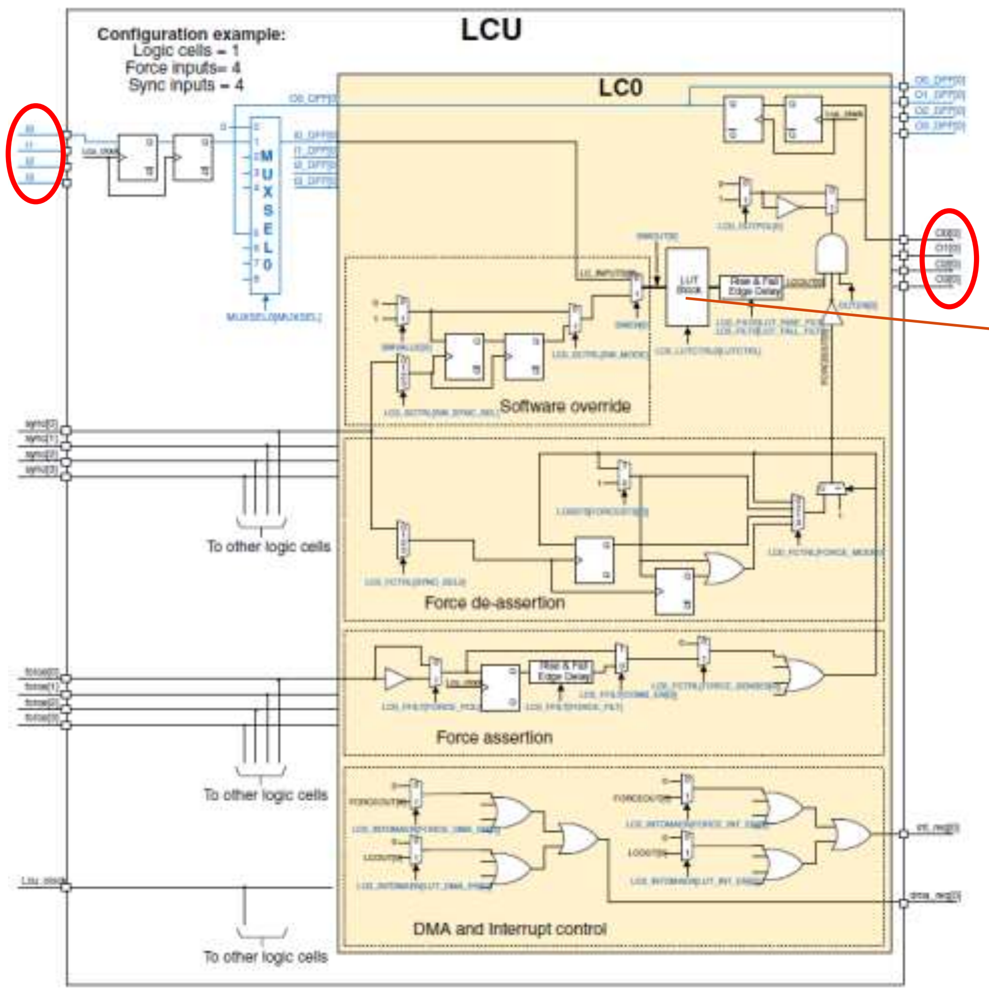
Example Project—Dependency and HW requirements

- **Dependent RTD components**
  - **Siul2_Dio**: provide the GPIO pin operation API functions;
  - **Siul2_Port**: provide the GPIO PORT configuration API functions;
  - **Clock**: configure system clocks;
  - **Emios_Pwm**: provide the PWM output configuration API function;
  - **Emios_Mcl**: provide the eMIOS basic configuration API function;
  - **Trgmux**: provide the Trgmux configuration API function.

- **HW requirements**
  - S32K344-WB(schematic: SPF-47478 Rev.A)
  - S32K3X4EVB-Q257(schematic: SPF-47651 Rev.B)
  - Support by two different **.mex** file, need to open in **S32_CT** and re-generate the configuration codes.

LUT CONFIGURATION TIPS

LUT Configuration Tips



Logic cell 0 O0= not I0_DFF
O1= I0_DFF

Inputs					Outputs			
I3	I2	I1	I0		O0	O1	O2	O3
Not related	Not related	Not related	PWM		PWM_LS	PWM_HS	Not related	Not related
0	0	0	0		1	0	0	0
0	0	0	1		0	1	0	0
0	0	1	0		1	0	0	0
0	0	1	1		0	1	0	0
0	1	0	0		1	0	0	0
0	1	0	1		0	1	0	0
0	1	1	0		1	0	0	0
0	1	1	1		0	1	0	0
1	0	0	0		1	0	0	0
1	0	0	1		0	1	0	0
1	0	1	0		1	0	0	0
1	0	1	1		0	1	0	0
1	1	0	0		1	0	0	0
1	1	0	1		0	1	0	0
1	1	1	0		1	0	0	0
1	1	1	1		0	1	0	0
LUT register value					0x5555	0xAAAA	0x0000	0x0000

SECURE CONNECTIONS
FOR A SMARTER WORLD