How to use printf() to print string to UART in KDS3.0 + KSDK1.3

By: Alice Yang

This document includes two chapters , chapter 1 is about how to use printf() to print string in KSDK1.3 project , the usage in project without MQX has been introduced on another document, you can find it here :

https://community.freescale.com/docs/DOC-104349, so in this DOC I only introduce how to use the printf() in KSDK MQX- Lite and KSDK MQX-Standard project.

Chapter 2 introduces how to check which UART port is used when use printf() to print string on the FRDM board and TOWER board.

1. Use printf() in KSDK MQX project 1.1 Use printf() in KSDK MQX-Lite project

1.2 Use printf() in KSDK MQX-Standard project

- 2. How to check which UART port is used when use printf()
- 2.1 How to check which UART port is used on FRDM board when use printf()
 2.2 How to check which UART port is used on TOWER board when use printf()
 2.2.1 Use the OpenSDA port on

"TWR-K65F180M" board 2.2.2 Use the UART port on "TWR – SER" board.

1. Use printf() in KSDK MQX project

It includes two cases: KSDK MQX-Lite project and KSDK MQX-Standard project.

- 1.1 Use printf() in KSDK MQX-Lite project
 - Create a KSDK MQX-Lite project: after create the KSDK1.3 project with Processor Expert , Select the "MQX KSDK" in the OS option :

S Component Inspector - osal 🕺 S Components Library	Basic Advanced	🖰 🗘 🔿	▽ □	
Properties Methods				
Component name osal				*
Component version 1.3.0			_	
OS BareMetal				
BareMetal				
Vs Witz_KSUK pd components				
DeluCOSIII ick				
Counter SYST_CVR				
Timer period 1 ms in 1.000 ms				
Interrupt settings				
Interrupt INT_SysTick				
Interrupt priority redium priority r 112				
Install interrupt				
ISR name SysTick_Handler				
>>				
				Ŧ
			P.	

then click this:

📎 *Component Inspector - osal 🙁 🚫 Components Libra	ry
Properties Methods	
type filter text All OS Inherited Component	Component name psal Component version 1.3.0 OS MQX_KSDK Inherited Component MQX KSDK 1.3.0/MQX_KSDK

Select "MQX Lite":

🏷 (Component Inspector - mqx_ksdk 🕱 🗞 Components Library
Prop	erties Methods Events
Prop	erties Methods Events Component name mqx_ksdk Component version 1.3.0 Settings Configuration parameters Inherited components Shared components Configuration set MQX Lite Components MQX Lite Components MQX Standard Lightweight events Lightweight message queues Message Mutexes Timer Use idle task Floating point context saving Task destruction
	Lowest task priority 32
	User settings
	Definitions 1 line(s)
	4

- In the Components window , we can see the component "fsl_debug_console", double click this component , then the UART configuration view on show at the right , configure the UART: Device, Baud rate and pins. (For example for FRDM-K64 board)



- Configure the PuTTY: 1152000 baud, No parity, 8 data bits, 1 stop bit.

🕵 PuTTY Configuration	
Category:	
Session Logging Terminal Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH SSH SSH Serial	Basic options for your PuTTY session Specify the destination you want to connect to Serial line COM2 Connection type: Raw Telnet Rlogin SSH Serial Load, save or delete a stored session Saved Sessions Default Settings Load Save Default Settings Load Save Delete
About	Open Cancel

- Run the project, we can see the result on the PuTTY.

Debug - printf_sdk_mqx-lite_pe_k64/Sources/os_tasks.c - Kinetis Design Studio - C\Users\b51432\workspace_ha File_Edit_Source_Ref.stor_Nuigate_Source_Design_Dup_MOX_Descares Exact_Window_Halp	andson.kds	
ine cuit source veración vangate search project num mor processi cipert vintovi resp : • • • • • • • • • • • • • • • • • • •	∋ ∦ ▼ [] ≎ !!! ▼ ? ▼ != (= + → -	
* Debug 🛙	<u>``</u> # # ▼ = □	🗱 🗣 Variables 😫 💁 Breakpoints 🔠 Registers 🕮 EmbSys Registers
 ▲ [e] printf_sdk_mqx-lite_pe_t64_Debug_Segger [GDB SEGGER I-Link Debugging] ▲ @printf_sdk_mqx-lite_pe_t64.elf ↓ DinkGDBSeverCL ↓ LinkGDBSeverCL ↓ Barn-none-eabi-gdb ↓ Semihosting and SWV 		Name Type
Desirateformet Dessing Destations M	Party COM3 - PuTTY	
<pre>*/ Void Taski_task(os_task_param_t task_init_data) {</pre>	Thank you for your interest in NKP S Thank you for your interest in NKP S	emiconductor project o emiconductor project o emiconductor project emiconductor project
<pre>sidef PEX_USE_RTOS } sendif } /* ElD os_tasks */ #idefcolumnum</pre>	Thank you for your interest in NRP S Thank you for your interest in NRP S	emiconductor project emiconductor project emiconductor project emiconductor project emiconductor project emiconductor project
} /* #endif ⊖ /*! e)	Thank you for your interest in NAP S	

1.2 Use printf() in KSDK MQX-Standard project

- Create a KSDK MQX-Standard project: after create the KSDK1.3 project with Processor Expert , select the "MQX_KSDK" in the OS option, then select "MQX Standard"

🗞 Component Inspector - osal 🙁 🗞 Components Library	Basic Advanced	💾 🔶 🔿	~ -	
Properties Methods				
Component name osal				*
Component version 1.3.0				
Os BareMetal A				
BareNetal				
Us MQX.KSDK d components EncePTIOS.ture.mov				
Counter Syst_CVN				
Interpreted and the two many according to th				
Interrupt INT_SysTick				
Interrupt priority medium priority + 112				
Install interrupt				
ISR name SysTick_Handler				
>>				
			1	*

Properties Methods Events
Component name mqx_ksdk
Component version 1.3.0
Settings
Configuration parameters Inherited components Shared components
Configuration set MQX Standard
Verification options MQX Lite
Components MQX Standard
Lightweight events
Lightweight message queues 🔽
Message V
Timer V
Use idle task
>> Floating point context saving
Task destruction
Lowest task priority 32
User settings
Definitions 1 line(s)

- Please delete the component of "fsl_debug_console" under the project, for in the MQX-Standard library, the project do not need this component to print, it have own driver for printf(), we only need configure the UART.



-In the debug console , please select "KSDK1.3.0/fsl_uart"

🏠 Project Explorer 💥 📄 📮 🗖 🗖	🗞 *Component Inspector - mqx_ksdk 🛛 💊 Comp	onents Library							
printf_sdk_mqx-standard_pe_k64	Properties Methods Events								
	type filter text	ſ		C	ha and ha alle				
	⊿ All			Component versio	mqx_ksak				
	⊿ Settings			Settings	1 100				
	Configuration parameters Inheritad components								
	innented components			Configuration pa	rameters Inherited com	oonents			
				debug console	KSDK 1.3.0/fsl_uart	- >			
				fsl_mpu	KSDK 1.3.0/fsl_mpu	- >			
				fsl_hwtimer	KSDK 1.3.0/fsl_hwtimer	- >			
				OS_Task	KSDK 1.3.0/OS_Task	- >			
	L			•					
😪 Components - printf sdk max-standard pe k 💥 😑 🗖									
Generator_Configurations									
🖓 RAM									
FLASH FLASH	Problems Property Console Property	ior.							
▲ Uss b Ma osalifs os abstraction	7 errors 0 warnings 0 others	103							
Processors	Description	Resource Pat	h	Locat	ion Type				
Cpu:MK64FN1M0VLQ12	Errors (7 items)				21				
Gpu:MK64FN1M0VLQ12									
Components A Components									
b Image: bin_mux:PinSettings									

- Double click the component of "fsl_uart", then configure it :

😪 Components - df 🔀	🖻 🧐 🔒	▽ □ □	-	*Cor	mponent	Inspector - uart	1 🕺 🚫 Compone	nts Library						Bas
osa1:fsl_os_abstraction		^	ER D	rone	rtier 1	Aethods Events)							
max_ksdk:MOX_KSPK				Tope		ethous Evenus								
b b uart1:fsl_uart				Co	omponen	t name uart1								
j sl_mpu1:fsl_mpu						0	- 0.	and a set of the set of the set						
fsl_hwtimer1:fsl_hwtimer					evice		• Ur	lassigned periprieral						
MainTask:OS_Task					omponen	t version 1.3.0								
Interrupt handling				Co	omponer	t mode Inter	rupt mode 👻							
Memory with variable-size	blocks													
Mutexes		Configurations Pins/Signals Initialization Shared components Inherited components												
Miscellaneous					Config	urations								
kernel log					V LIART	configurations								
b b Lightweight semaphores					- OAIG	connigurations								
b b Lightweight events					Config	rations list	- 1 +							
b b Lightweight logs					#	Configuration	Name	Type	Read only	Raud rate	Parity mode	Stop hits	Bits per char	
b b Lightweight message queu	le				-	configuration	Nume	type	Read only	O	runty mouc	Stop bits	bits per entir	
b b Lightweight timer					⊠ 0	V	uart1_InitConfig0	uart_user_config_t	v	Ű	Disabled	1	8	
b B Scheduling														
Description of the second s														
Task queues			>>											
D Eming														
b b specific														
OSA_SemaCreate														

- Configure the PuTTY as I mentioned in the 1.1, run the project, we can see the result on the PuTTY.

🖻 rtos_main_task.c 🛛 💽 0x0 🔒 main.c 🔒 idletask.c		
⊖ void main_task(os_task_param_t task_init_data)		
<pre>{ /* Write your local variable definition here */</pre>		
/* Initialization of Processor Expert components (when some RTOS is active).	DON'T REMOVE THIS CODE!!! */	
<pre>#ifdef MainTask_PEX_RTOS_COMPONENTS_INIT PEX_components_init();</pre>	B COM3 - PuTTY	
<pre>#end1+ /* End of Processor Expert components initialization. */</pre>	Thank you fom your interest in NXP Semiconductor project	*
<pre>#ifdef PEX_USE_RTOS while (1) { tendif </pre>	Thank you fom your interest in NXP Semiconductor project	
/* Write your code here */	Thank you fom your interest in NXP Semiconductor project Thank you for yomr interest in NXP Semiconductor project	
<pre>printf("Thank you for your interest in NXP Semiconductor project \r\n"); OSA_TimeDelay(10); /* Example code (for task release) */</pre>	Thank you for yomr interest in NXP Semiconductor project	
	Thank you for yomr interest in NXP Semiconductor project	
#Ifdaf DEV LICE DTOC	Thank you for yomr interest i	-

In sum, only the project of "MQX standard" use it own driver for printf(), so we need delete the driver "fsl_debug_console". In other SDK+PE projects , they all use the driver of "fsl_debug_console" for printf().

2. How to check which UART port be used when use printf()

When use kinetis demo boards, some customers don't know which UART port should be used when use the printf() function, so I'd like to focus on the FRDM board and TOWER board about how to cofirm which UART port.

- 2.1 How to check which UART port be used on FRDM board when use printf() For example the FRDM-K64F board:
 - On the FRDM-K64F board , we use the OpenSDA port, that is J26 pin, then we search the "J26" on FRDM-K6 4F sch, we can see J26 connect to the chip of "MK20D128VFM5"



- Then continue to check which place the MK20D128VFM5 be connected, we can see it have two signals "UART1_RX_TGTMCU" and "UART1_TX_TGTMCU", they are connected to "PTB16" and "PTB17".



- Then continue to search the two pins "PTB16" and "PTB17" be connected to which UART port on the k64 chip ,the result is they are connected to UART0 of k64.

ng (0, 4)			SPIL POOVUARIS RAFE ADIS SPIL SCK/UARTS_TX/FB_ADIS	50 59	PTB10/ADC1_SE14/SPI1_PCS0/UART3_RX/FB_AD19/FTM0_FLT1
pg (3,4)	PTB[1023]	PTB16	SPI1_SOUT/FB_AD17	62	
		N PTB17	SPI1 SIN/FB AD16	63	
		PIB18	CAN0 TX/I2S0 TX BCLK/FB AD15	64	
		DTDIA	OTHER DIVISION THE FOURD OF D	05	T PIBI8/GANU IA/ELMZ GHU/ZOU IA BUIN/EB AUID/ELMZ UU

So on FRDM-K64F board, when we use printf() function, configure the UARTO, PTB16 pin as RX, PTB17 as TX.

2.2 How to check which UART port be used on TOWER board when use printf()

The Tower System includes four modules, the main board "TWR-xxx(chip name)", two elevator modules "TWR-ELEV Primary" and "TWR-SELE Secondary", and the "TWR-SER" module. In generally, there are two ports to print use UART on the TOWER demo board. One is use the OpenSDA on the main board, the other one is use the UART port on "TWR-SER" board.

For example the TWR-K65F18M board:

2.2.1 Use the OpenSDA port on "TWR-K65F180M" board



In this situation , it is the same with on FRDM board to check which UART port is used.

- Search J7 pin on "TWR-K65F180M" board, it is connected to "MK20DX128VFM5" chip, then we can find two signal on it "OpenSDA_URX" and "Open_UTX".



- Then continue to check the two signals , we can see they can be connected to "UART2_TX" and "UART2_RX " throgh J33 and J34, please set jumper to connect the "2" and "3".



- Then search the PTE16 and PTE17 on K65 chip , they are connected to UART2 of the chip.

PIETI/203_301/230_1X_F3/LPUARTU_RT3/FTM3_0H0	H4	12SO TX BOLK	PTF12
	H3	UART2_TX/SPI0_PCS0	PTE16
	F5	UART2 RX/SPI0 SCK	PTE17
	E6		PTE18
PTE19/ADC0_SE7A/SPI0_SIN/LIABT2_RTS//2C0_SCI_/CMP3_OUT	F/	I2C0 SCL/SPI0 SIN	PTE19
PTE24/ADC0 SE17/CAN1 TX/UART4 TX/I2C0 SCI/EWM OUT	L/ K7	CANT IX/ADC0 SE17	PTE24
PTE25/ADC0 SE18/LLWU P21/CAN1 RX/UART4 RX/I2C0 SDA/EWM IN	KQ KQ		PTE25
PTE26/ENET_1588_CLKIN/UART4_CTS/RTC_CLKOUT/USB0_CLKIN	1.8	ENET 1300 OEMIN	PTF27
PTE27/UART4_RTS	M7		PTE28
PTE28			

So use the UART2 port, PTE16 pin as TX, PTE17 pin as RX.



2.2.2 use the UART port on "TWR-SER" board.

- Search the J8 on "TWR-SER", we can see it have two signals "ELE_RXD" and "ELE_TXD"



They are connected to the "A43" and "A44" of J1 .

RXD0	A41 ×	
	A43 ^	ELE_RXD
	A44	ELE_TXD
BKGD	A45 ×	

- Then search "A43" and "A44" on TWR-K65F180M board, we can find they are connected to

"ELEV_UART_RX" and "ELEV_UART_TX".

UARTO_HX	A42	RA42 0 R104	UARTO TX PTA2		
UARTO_TA	A43				
UARTI_RA	A44				
UARTI_TX	A45	VSSAR 0 A R100	DND CELEV_OART_IX (9		
VSSA	A46	VDDAR 0	DNP WORK		

- Then search "ELEV_UART_RX" and "ELEV_UART_TX" to continue find which UART port, we can see the two signals become signals "ELEV_URX" and "ELEV_UTX" through electric converter, the two signals can be connected to "PTE16" pin and "PTE17" pin through J33(1-2) and J34(1-2).



- So, connect the 1 and 2 of J33 J34, we will use the PTE16 pin and PTE17 pin, we have find the two pins are the UART2 on the above section. That is to say the UART2. In fact, the connection between J7 on TWR-SER board and UART2 on K65F180M chip is the two elevator modules, while the label of the connection is the same on the four board(for example the "A43" and "A44"), so we can directly search it on TWR-K65F180M board, ignore searching it on the two elevator boards.