ARM Cortex Core Microcontrollers

5th Laboratory: UART DMA

Scherer Balázs

Budapest University of Technology and Economics
Department of Measurement and Information Systems

USART1 Communication with DMA

- USART1 is already configured

- Modify UART1 configuration to request DMA interaction
- Enable DMA2-s clock
- Configure DMA
- Start DMA
- Modify Printf to use DMA

Modify UART1 configuration to request DMA interaction

- USART needs to signal the DMA that it is able to send or receive the next byte

- Firmware library:
  - LL_USART_EnableDMAReq_TX(USART1);

Enable DMA2-s clock

- We are using DMA2 (AHB1 bus), Channel 4, Stream 7

Table 43. DMA2 request mapping

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1	SAI1_A(1)	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A(1)	ADC1	SAI1_B(1)	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
Channel 1	-	DCMI	ADC2	ADC2	SAI1_B(1)	SPI6_TX(1)	SPI6_RX(1)	DCMI
Channel 2	ADC3	ADC3	-	SPI5_RX(1)	SPI5_TX(1)	CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX	-	SPI1_RX	SPI1_TX	-	SPI1_TX	-	-
Channel 4	SPI4_RX(1)	SPI4_TX(1)	USART1_RX	SDIO	-	USART1_RX	SDIO	USART1_TX
Channel 5	-	USART6_RX	USART6_RX	SPI4_RX(1)	SPI4_TX(1)	-	USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
Channel 7	-	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX(1)	SPI5_TX(1)	TIM8_CH4 TIM8_TRIG TIM8_COM

- LL_AHB1_GRP1_EnableClock (LL_AHB1_GRP1_PERIPH_DMA2);

Configure DMA

- We are using DMA2 (AHB1 bus), Channel 4, Stream 7 and UART1->DR is the target

```
LL_DMA_InitTypeDef  DMA_InitStruct;

DMA_InitStruct.Channel = LL_DMA_CHANNEL_4;
DMA_InitStruct.Direction = LL_DMA_DIRECTION_MEMORY_TO_PERIPH;
DMA_InitStruct.FIFOMode = LL_DMA_FIFOMODE_DISABLE;
DMA_InitStruct.MemBurst = LL_DMA_MBURST_SINGLE;
DMA_InitStruct.MemoryOrM2MDstAddress = (uint32_t)buffer;
DMA_InitStruct.MemoryOrM2MDstDataSize = LL_DMA_MDATAALIGN_BYTE;
DMA_InitStruct.MemoryOrM2MDstIncMode = LL_DMA_MEMORY_INCREMENT;
DMA_InitStruct.Mode = LL_DMA_MODE_NORMAL;
DMA_InitStruct.NbData = strlen(buffer);
DMA_InitStruct.PeriphBurst = LL_DMA_PBURST_SINGLE;
DMA_InitStruct.PeriphOrM2MSrcAddress = (uint32_t)&(USART1->DR);
DMA_InitStruct.PeriphOrM2MSrcDataSize = LL_DMA_PDATAALIGN_BYTE;
DMA_InitStruct.PeriphOrM2MSrcIncMode = LL_DMA_PERIPH_NOINCREMENT;
DMA_InitStruct.Priority = LL_DMA_PRIORITY_LOW;

LL_DMA_Init  (DMA2,LL_DMA_STREAM_7, &DMA_InitStruct);
```

Start DMA

- Initialised stren needs to be started

```
LL_DMA_EnableStream  (DMA2, LL_DMA_STREAM_7);
```

DMA Printf

- Create a new string based I/O redirection with the byte based putch it offer no benefit

```
int __io_putstr(char *ptr, int len)
{
	LL_DMA_InitTypeDef  DMA_InitStruct;
	LL_DMA_DisableStream (DMA2, LL_DMA_STREAM_7);

	 LL_DMA_ClearFlag_TC7 (DMA2);
	DMA_InitStruct.Channel = LL_DMA_CHANNEL_4;
	DMA_InitStruct.Direction = LL_DMA_DIRECTION_MEMORY_TO_PERIPH;
	DMA_InitStruct.FIFOMode = LL_DMA_FIFOMODE_DISABLE;
	DMA_InitStruct.MemBurst = LL_DMA_MBURST_SINGLE;
	DMA_InitStruct.MemoryOrM2MDstAddress = (uint32_t)ptr;
	DMA_InitStruct.MemoryOrM2MDstDataSize = LL_DMA_MDATAALIGN_BYTE;
	DMA_InitStruct.MemoryOrM2MDstIncMode = LL_DMA_MEMORY_INCREMENT;
	DMA_InitStruct.Mode = LL_DMA_MODE_NORMAL;
	DMA_InitStruct.NbData = len;
	DMA_InitStruct.PeriphBurst = LL_DMA_PBURST_SINGLE;
	DMA_InitStruct.PeriphOrM2MSrcAddress = (uint32_t)&(USART1->DR);
	DMA_InitStruct.PeriphOrM2MSrcDataSize = LL_DMA_PDATAALIGN_BYTE;
	DMA_InitStruct.PeriphOrM2MSrcIncMode = LL_DMA_PERIPH_NOINCREMENT;
	DMA_InitStruct.Priority = LL_DMA_PRIORITY_LOW;

	LL_DMA_Init  (DMA2,LL_DMA_STREAM_7, &DMA_InitStruct);
	LL_DMA_EnableStream  (DMA2, LL_DMA_STREAM_7);
}
```