Driver Design

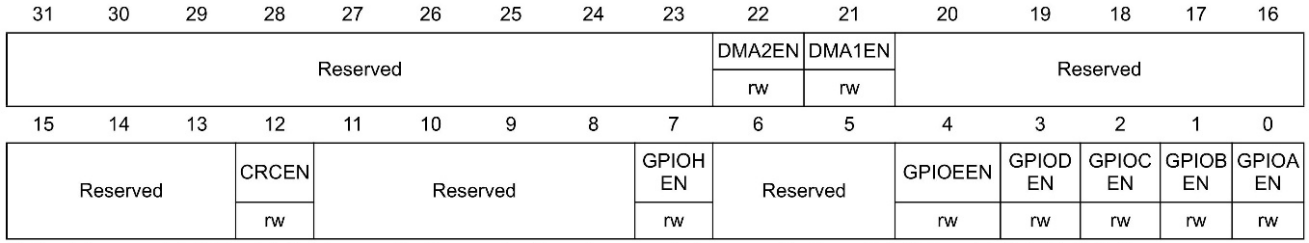
RCC, EXTI, TIMER

6.3.9 RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Address offset: 0x30

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

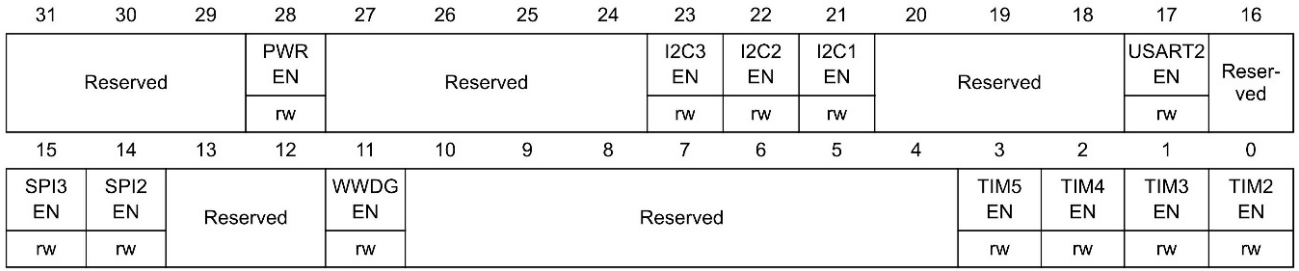


6.3.11 RCC APB1 peripheral clock enable register (RCC_APB1ENR)

Address offset: 0x40

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

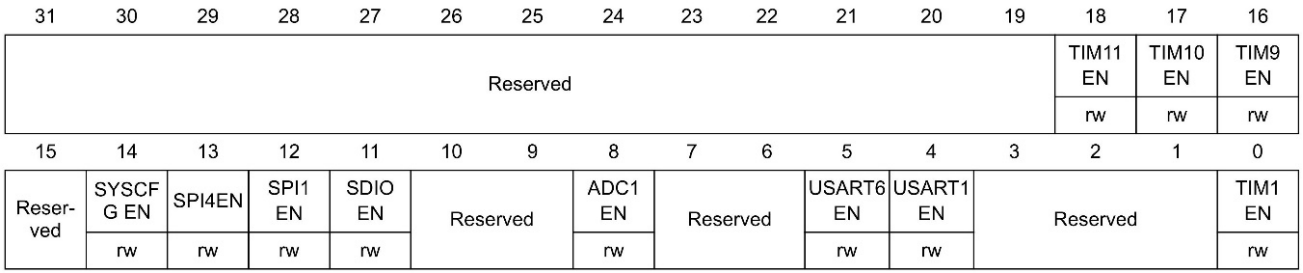


6.3.12 RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Address offset: 0x44

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.



Define RCC_Peripherals macros

- #define RCC_AHB1Periph_GPIOA ((uint32_t)0x00000001)
- #define RCC_AHB1Periph_GPIOB ((uint32_t)0x00000002)
- #define RCC_AHB1Periph_GPIOC ((uint32_t)0x00000004)
- …
- #define RCC_APB1Periph_TIM2 ((uint32_t)0x00000001)
- #define RCC_APB1Periph_TIM3 ((uint32_t)0x00000002)
- #define RCC_APB1Periph_TIM4 ((uint32_t)0x00000004)
- …
- #define RCC_APB2Periph_USART1 ((uint32_t)0x00000010)
- #define RCC_APB2Periph_ADC1 ((uint32_t)0x00000100)
- #define RCC_APB2Periph_SYSCFG ((uint32_t)0x00004000)
- …

RCC

- APB2 Peripheral clocks configuration.
  - void RCC_APB2PeriphClockCmd(uint32_t RCC_Periph, FunctionalState NewState);

10.3.1 Interrupt mask register (EXTI_IMR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	Reserved		MR18	MR17	MR16
									rw	rw			rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

10.3.3 Rising trigger selection register (EXTI_RTSR)

Address offset: 0x08
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	Reserved		TR18	TR17	TR16
									rw	rw			rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

10.3.4 Falling trigger selection register (EXTI_FTSR)

Address offset: 0x0C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	Reserved		TR18	TR17	TR16
									rw	rw			rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Define EXTI_Lines macros

- #define EXTI_Line0 ((uint32_t)0x00001)
- #define EXTI_Line1 ((uint32_t)0x00002)
- #define EXTI_Line2 ((uint32_t)0x00004)
- #define EXTI_Line3 ((uint32_t)0x00008)
- …

EXTI

- Initializes the EXTI peripheral.
  - EXTI line configuration.
  - Rising Falling edge configuration.
  - void EXTI_Init(uint32_t EXTI_Line, uint32_t Rising_Mask, uint32_t Falling_Mask, FunctionalState NewState);
- Checks whether the specified EXTI line is asserted or not.
  - ITStatus EXTI_GetITStatus(uint32_t EXTI_Line);
- Clears the EXTI's line pending bits.
  - void EXTI_ClearITPendingBit(uint32_t EXTI_Line);

TIMER

- Sets the TIMx Capture Compare Register value
  - void TIM_SetCompare1(TIM_TypeDef* TIMx, uint32_t Compare1);
  - void TIM_SetCompare2(TIM_TypeDef* TIMx, uint32_t Compare2);
  - void TIM_SetCompare3(TIM_TypeDef* TIMx, uint32_t Compare3);
  - void TIM_SetCompare4(TIM_TypeDef* TIMx, uint32_t Compare4);

13.4.4 TIMx DMA/Interrupt enable register (TIMx_DIER)

Address offset: 0x0C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

13.4.5 TIMx status register (TIMx_SR)

Address offset: 0x10

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Define TIM_interrupt_sources macros

- #define TIM_IT_Update ((uint16_t)0x0001)
- #define TIM_IT_CC1 ((uint16_t)0x0002)
- #define TIM_IT_CC2 ((uint16_t)0x0004)
- #define TIM_IT_CC3 ((uint16_t)0x0008)
- #define TIM_IT_CC4 ((uint16_t)0x0010)
- #define TIM_IT_Trigger ((uint16_t)0x0040)

TIMER

- Enables or disables the specified TIM interrupts.
  - void TIM_ITConfig(TIM_TypeDef* TIMx, uint16_t TIM_IT, FunctionalState NewState);
- Checks whether the TIM interrupt has occurred or not.
  - ITStatus TIM_GetITStatus(TIM_TypeDef* TIMx, uint16_t TIM_IT);
- Clears the TIMx's interrupt pending bits.
  - void TIM_ClearITPendingBit(TIM_TypeDef* TIMx, uint16_t TIM_IT);