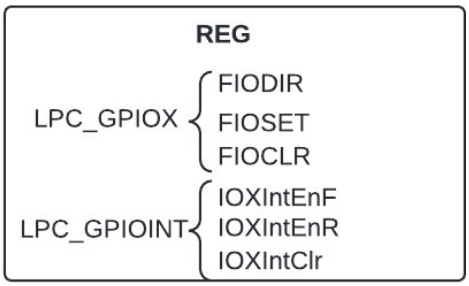
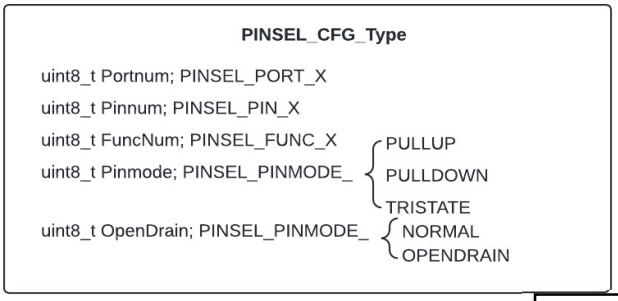
“lpc17xx_gpio.h”

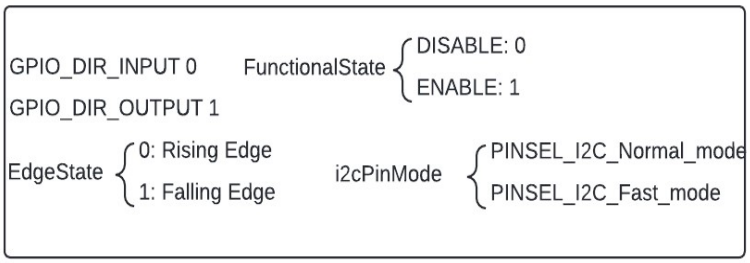
- void **GPIO_SetDir**(uint8_t portNum, uint32_t bitValue, uint8_t dir);
- void **GPIO_SetValue**(uint8_t portNum, uint32_t bitValue);
- void **GPIO_ ClearValue** (uint8_t portNum, uint32_t bitValue);
- void **GPIO_ ReadValue** (uint8_t portNum);
- void **GPIO_IntCmd**(uint8_t portNum, uint32_t bitValue, uint8_t edgeState);
- void **GPIO_ClearInt**(uint8_t portNum, uint32_t bitValue);
- FunctionalState GPIO_GetIntStatus(uint8_t portNum, uint32_t pinNum, uint8_t edgeState);



“lpc17xx_pinsel.h”

- void **PINSEL_ConfigPin**(PINSEL_CFG_Type* PinCfg);
- void **PINSEL _ConfigTraceFunc**(FunctionalState NewState);
- void **PINSEL _ SetI2C0Pins**(uint8_t i2cPinMode, FunctionalState filterSlewRateEnable);





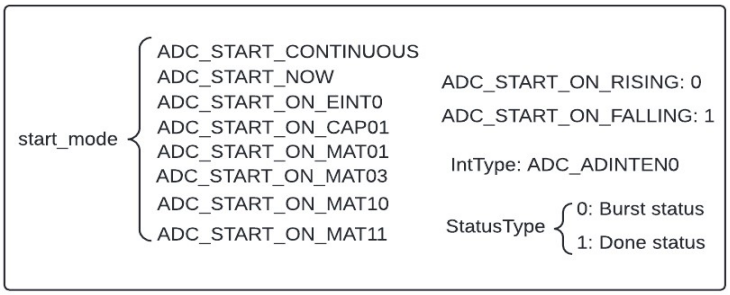
“lpc17xx_systick.h”

- void **SYSTICK_InternalInit**(uint32_t time);
- void **SYSTICK_ExternalInit**(uint32_t freq, uint32_t time);
- void **SYSTICK_Cmd**(FunctionalState NewState);
- void **SYSTICK_IntCmd**(FunctionalState NewState);
- uint32_t **SYSTICK_GetCurrent_Value**(void);
- void **SYSTICK_ClearCounterFlag**(void);

maximum time can be set:
* 1/SystemCoreClock * (2^24) * 1000 (ms)

“lpc17xx_adc.h”

- void **ADC_Init**(LPC_ADC_TypeDef* ADCx, uint32_t rate); // ADCx: LPC_ADC rate:[Hz]
- void **ADC_DeInit**(LPC_ADC_TypeDef* ADCx);
- void **ADC_BurstCmd**(LPC_ADC_TypeDef* ADCx, FunctionalState NewState);
- void **ADC_PowerdownCmd**(LPC_ADC_TypeDef* ADCx, FunctionalState NewState);
- void **ADC_StartCmd**(LPC_ADC_TypeDef* ADCx, uint8_t start_mode);
- void **ADC_ChannelCmd**(LPC_ADC_TypeDef* ADCx, uint8_t Channel, FunctionalState NewState);
- void **ADC_EdgeStartConfig**(LPC_ADC_TypeDef* ADCx, uint8_t EdgeOption);
- void **ADC_IntConfig**(LPC_ADC_TypeDef* ADCx, ADC_TYPE_INT_OPT IntType, FunctionalState NewState);
- uint16_t **ADC_ChannelGetData**(LPC_ADC_TypeDef* ADCx, uint8_t channel);
- FlagStatus **ADC_ChannelGetStatus**(LPC_ADC_TypeDef* ADCx, uint8_t channel, uint32_t StatusType);
- uint32_t **ADC_GlobalGetData**(LPC_ADC_TypeDef* ADCx);
- FlagStatus **ADC_GlobalGetStatus**(LPC_ADC_TypeDef* ADCx, uint32_t StatusType);



“lpc17xx_timer.h”

- void **TIM_Init**(LPC_TIM_TypeDef* TIMx, TIM_MODE_OPT TimerCounterMode, void* TIM_ConfigStruct);
- void **TIM_DeInit**(LPC_TIM_TypeDef* TIMx);
- void **TIM_ClearIntPending**(LPC_TIM_TypeDef* TIMx, TIM_INT_TYPE IntFlag);
- void **TIM_ClearIntCapturePending**(LPC_TIM_TypeDef* TIMx, TIM_INT_TYPE IntFlag);
- FlagStatus **TIM_GetIntStatus**(LPC_TIM_TypeDef* TIMx, TIM_INT_TYPE IntFlag);
- FlagStatus **TIM_GetIntCaptureStatus**(LPC_TIM_TypeDef* TIMx, TIM_INT_TYPE IntFlag);
- void **TIM_ConfigStructInit**(TIM_MODE_OPT TimerCounterMode, void* TIM_ConfigStruct);
- void **TIM_ConfigMatch**(LPC_TIM_TypeDef* TIMx, TIM_MATCHCFG_Type* TIM_MatchConfigStruct);
- void **TIM_UpdateMatchValue**(LPC_TIM_TypeDef* TIMx, uint8_t MatchChannel, uint32_t MatchValue);
- void **TIM_SetMatchExt**(LPC_TIM_TypeDef* TIMx, TIM_EXTMATCH_OPT ext_match);
- void **TIM_ConfigCapture**(LPC_TIM_TypeDef* TIMx, TIM_CAPTURECFG_Type* TIM_CaptureConfigStruct);
- void **TIM_Cmd**(LPC_TIM_TypeDef* TIMx, FunctionalState NewState);
- uint32_t **TIM_GetCaptureValue**(LPC_TIM_TypeDef* TIMx, TIM_COUNTER_INPUT_OPT CaptureChannel);
- void **TIM_ResetCounter**(LPC_TIM_TypeDef* TIMx);

TIM_TIMERCFG_Type

uint8_t PrescaleOption; { TIM_PRESCALE_TICKVAL, TIM_PRESCALE_USVAL }
uint8_t Reserved[3];
uint32_t RescaleValue;

TIM_MATCHCFG_Type

uint8_t MatchChannel;
uint8_t IntOnMatch;
uint8_t ResetOnMatch;
uint8_t ExtMatchOutputType; { TIM_EXMATCH_NOTHING, TIM_EXMATCH_LOW, TIM_EXMATCH_HIGH, TIM_EXMATCH_TOGGLE }
uint8_t Reserved[3];
uint8_t MatchValue;

TIMx → "lpc17xx.h" LPC_TIMX

IntFlag { TIM_MR0_INT, TIM_MR1_INT, TIM_MR2_INT, TIM_MR3_INT, TIM_CR0_INT, TIM_CR1_INT }

TimerCounterMode { TIM_TIMER_MODE, TIM_COUNTER_RISING_MODE, TIM_COUNTER_FALLING_MODE, TIM_COUNTER_ANY_MODE }

“lpc17xx_exti.h”

- void **EXTI_Init**(void);
- void **EXTI_DeInit**(void);
- void **EXTI_Config**(EXTI_InitTypeDef* EXTICfg);
- void **EXTI_SetMode**(EXTI_LINE_ENUM EXTILine, EXTI_MODE_ENUM mode);
- void **EXTI_SetPolarity**(EXTI_LINE_ENUM EXTILine, EXTI_POLARITY_ENUM polarity);
- void **EXTI_ClearEXTIFlag**(EXTI_LINE_ENUM EXTILine);

EXTI_InitTypeDef

EXTI_LINE_ENUM EXTI_Line; { EXTI_EINTX

EXTI_MODE_ENUM EXTI_Mode; { EXTI_MODE_LEVEL_SENSITIVE
EXTI_MODE_EDGE_SENSITIVE

EXTI_POLARITY_ENUM EXTI_Mode; { EXTI_POLARITY_LOW_ACTIVE_OR_FALLING_EDGE
EXTI_POLARITY_HIGH_ACTIVE_OR_RISING_EDGE

Exception Number	IRQNumber	Vector Offset	Exception	Priority	CMSIS_Handler
1	-	0x4	Reset	-3, Highest	Reset_Handler
2	-14	0x8	NMI	-2	NMI_Handler
3	-13	0xC	Hard fault	-1	HardFault_Handler
4	-13	0x10	Memory Management fault	config	MemManage_Handler
5	-11	0x14	Bus fault	config	BusFault_Handler
6	-10	0x18	Usage fault	config	UsageFault_Handler
7-10	-	-	-	-	-
11	-5	0x2C	SVCall	config	SVC_Handler
12-13	-	-	-	-	-
14	-2	0x38	PendSV	config	PendSV_Handler
15	-1	0x3C	Systick	config	Systick_Handler

Exception Number	IRQNumber	Vector Offset	Interruption	Priority	CMSIS_IRQ
16	0	0x40	WDT	-	WDT_IRQHandler
17	1	0x44	Timer 0	-	TIMER0_IRQHandler
18	2	0x48	Timer 1	-	TIMER1_IRQHandler
19	3	0x4C	Timer 2	-	TIMER2_IRQHandler
20	4	0x50	Timer 3	-	TIMER3_IRQHandler
21	5	0x54	UART0	-	UART0_IRQHandler
22	6	0x58	UART1	-	UART1_IRQHandler
23	7	0x5C	UART2	-	UART2_IRQHandler
24	8	0x60	UART3	-	UART3_IRQHandler
25	9	0x64	PWM1	-	PWM1_IRQHandler
26	10	0x68	I2C0	-	I2C0_IRQHandler
27	11	0x6C	I2C1	-	I2C1_IRQHandler
28	12	0x70	I2C2	-	I2C2_IRQHandler
29	13	0x74	SPI	-	SPI_IRQHandler
30	14	0x78	SSP0	-	SSP0_IRQHandler
31	15	0x7C	SSP1	-	SSP1_IRQHandler
32	16	0x80	PLL0	-	PLL0_IRQHandler
33	17	0x84	RTC	-	RTC_IRQHandler
34	18	0x88	EINT0	-	EINT0_IRQHandler
35	19	0x8C	EINT1	-	EINT1_IRQHandler
36	20	0x90	EINT2	-	EINT2_IRQHandler
37	21	0x94	EINT3	-	EINT3_IRQHandler
38	22	0x98	ADC	-	ADC_IRQHandler
39	23	0x9C	BOD	-	BOD_IRQHandler
40	24	0xA0	USB	-	USB_IRQHandler
41	25	0xA4	CAN	-	CAN_IRQHandler
42	26	0xA8	GPDMA	-	DMA_IRQHandler
43	27	0xAC	I2S	-	I2S_IRQHandler
44	28	0xB0	Ethernet	-	ENET_IRQHandler
45	29	0xB4	RITINT	-	RIT_IRQHandler
46	30	0xB8	Motor PWM	-	MCPWM_IRQHandler
47	31	0xBC	Quad Encoder	-	QEI_IRQHandler
48	32	0xC0	PLL1	-	PLL1_IRQHandler
49	33	0xC4	USB Activity	-	-
50	34	0xC8	CAN Activity	-	-

void **NVIC_SetPriority**(IRQn_t IRQn, uint32_t priority);
void **NVIC_GetPriority**(IRQn_t IRQn);
void **NVIC_EnableIRQ**(IRQn_t IRQn);
void **NVIC_DisableIRQ**(IRQn_t IRQn);