"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 _ SetI2COPins(uint8_t i2cPinMode, FunctionalState filterSlewRateEnable);

```
PINSEL_CFG_Type

uint8_t Portnum; PINSEL_PORT_X

uint8_t Pinnum; PINSEL_PIN_X

uint8_t FuncNum; PINSEL_FUNC_X

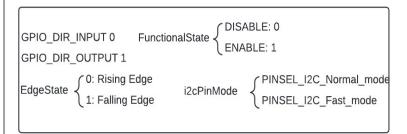
uint8_t Pinmode; PINSEL_PINMODE_

uint8_t OpenDrain; PINSEL_PINMODE_

VIRISTATE

VORMAL

OPENDRAIN
```



maximum time can be set:

* 1/SystemCoreClock * (2^24) * 1000 (ms)

REG

LPC GPIOX

LPC GPIOINT

FIODIR

FIOSET

FIOCLR

IOXIntEnF

IOXIntEnR IOXIntClr

"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);

"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);

```
start_mode 

ADC_START_CONTINUOUS

ADC_START_NOW
ADC_START_ON_EINT0
ADC_START_ON_CAP01
ADC_START_ON_MAT01
ADC_START_ON_MAT03
ADC_START_ON_MAT03
ADC_START_ON_MAT10
ADC_START_ON_MAT11

ADC_START_ON_MAT11

StatusType

O: Burst status
1: Done status
```

"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);

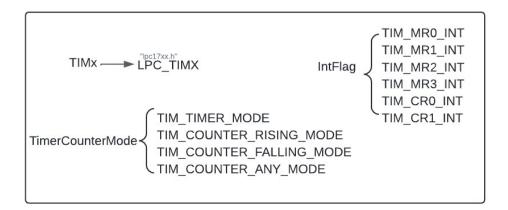
```
TIM_MATCHCFG_Type

uint8_t MatchChannel;
uint8_t IntOnMatch;
uint8_t ResetOnMatch;
uint8_t ExtMatchOutputType;

Uint8_t ExtMatchOutputType;

TIM_EXMATCH_NOTHING
TIM_EXMATCH_LOW
TIM_EXMATCH_HIGH
TIM_EXMATCH_HIGH
TIM_EXMATCH_TOGGLE

uint8_t MatchValue;
```



																		<pre>void NVIC_DisableIRQ(IRQn_t IRQn);</pre>	<pre>void NVIC_EnableIRQ(IRQn_t IRQn);</pre>	<pre>void NVIC_GetPriority(IRQn_t IRQn);</pre>	void NVIC_SetPriority			15	14	12-13	11	7-10	6	5	4	3	2	1	Exception Number
																		Q(IRQn_t IRC	(IRQn_t IRQ	y(IRQn_t IR	SetPriority(IRQn_t IRQn, uint32			-1	-2	-	-5		-10	-11	-13	-13	-14		IRQNumber
														շո);	-			0x3C	0x38	•	0x2C		0x18	0x14	0x10	0xC	0x8	0x4	Vector Offset						
25 1 25 1 25 1 25 1 25 1 25 1 25 1 25 1	MW9 100 100 100 100 100 100 100 100 100 10												TO T	priority);			Systick	PendSV	•	SVCall		Usage fault	Bus fault	Memory Management fault	Hard fault	NMI	Reset	t Exception							
	191	net				-•	UCXpresso LPC1769 REV B (6) 2818					0000	C Prif. Exceeded Artists S C C		ISC	-	Iq:	esi sosi sosi						config	config		config		config	config	config	-1	-2	-3, Highest	Priority
		SO O JAG TING SWOX JAG TING SWOX JAG TOS SWOX JAG TOS SWOX VIO JAG TOS SWOX VIO JAG TOS SWOX O JAG TOS S																		Systick_Handler	PendSV_Handler		SVC_Handler		UsageFault_Handler	BusFault_Handler	MemManage_Handler	HardFault_Handler	NMI_Handler	Reset_Handler	CMSIS_Handler				
50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Exception Number IRQNumber Vector Offset Interruption
34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IRQNumber
0xC8	0xC4	0xC0	0xBC	0xB8	0xB4	0xB0	0xAC	0xA8	0xA4	0xA0	0x9C	0x98	0x94	0x90	0x8C	0x88	0x84	0x80	0x7C	0x78	0x74	0x70	0x6C	0x68	0x64	0x60	0x5C	0x58	0x54	0x50	0x4C	0x48	0x44	0x40	Vector Offse
CAN Activity	USB Activity	PLL1	Quad Encoder	Motor PWM	RITINT	Ethernet	125	GPDMA	CAN	USB	BOD	ADC	EINT3	EINT2	EINT1	EINTO	RTC	PLLO	SSP1	SSP0	SPI	12C2	12C1	12C0	PWM1	UART3	UART2	UART1	UART0	Timer 3	Timer 2	Timer 1	Timer 0	WDT	Interruption
	•	,	,		,	,		,		,			,	,		,	,			,								,			•				Priority
	•	PLL1_IRQHandler	QEI_IRQHandler	MCPWM_IRQHandler	RIT_IRQHandler	ENET_IRQHandler	I2S_IRQHandler	DMA_IRQHandler	CAN_IRQHandler	USB_IRQHandler	BOD_IRQHandler	ADC_IRQHandler	EINT3_IRQHandler	EINT2_IRQHandler	EINT1_IRQHandler	EINTO_IRQHandler	RTC_IRQHandler	PLL0_IRQHandler	SSP1_IRQHandler	SSP0_IRQHandler	SPI_IRQHandler	I2C2_IRQHandler	I2C1_IRQHandler	I2C0_IRQHandler	PWM1_IRQHandler	UART3_IRQHandler	UART2_IRQHandler	UART1_IRQHandler	UARTO_IRQHandler	TIMER3_IRQHandler	TIMER2_IRQHandler	TIMER1_IRQHandler	TIMER0_IRQHandler	WDT_IRQHandler	CMSIS_IRQ