```
/****************************************************************************//**
 * @file     LPC17xx.h
 * @brief    CMSIS Cortex-M3 Core Peripheral Access Layer Header File for
 *           NXP LPC17xx Device Series
 * @version: V1.09
 * @date:    17. March 2010
 *
 * @note
 * Copyright (C) 2009 ARM Limited. All rights reserved.
 *
 * @par
 * ARM Limited (ARM) is supplying this software for use with Cortex-M
 * processor based microcontrollers.  This file can be freely distributed
 * within development tools that are supporting such ARM based processors.
 *
 * @par
 * THIS SOFTWARE IS PROVIDED "AS IS".  NO WARRANTIES, WHETHER EXPRESS,
IMPLIED
 * OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS
SOFTWARE.
 * ARM SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL,
OR
 * CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
 *
 ******************************************************************************/

#ifndef __LPC17xx_H__
#define __LPC17xx_H__

/*
 *
==============================================================================
=
 * ---------- Interrupt Number Definition -----------------------------------
 *
==============================================================================
=
 */

typedef enum IRQn
{
/******  Cortex-M3 Processor Exceptions Numbers
***********************************************************/
  NonMaskableInt_IRQn        = -14,    /*!< 2 Non Maskable Interrupt                */
  MemoryManagement_IRQn      = -12,    /*!< 4 Cortex-M3 Memory Management Interrupt */
  BusFault_IRQn              = -11,    /*!< 5 Cortex-M3 Bus Fault Interrupt         */
  UsageFault_IRQn            = -10,    /*!< 6 Cortex-M3 Usage Fault Interrupt       */
  SVCall_IRQn                = -5,     /*!< 11 Cortex-M3 SV Call Interrupt          */
```

```
  DebugMonitor_IRQn           = -4,      /*!< 12 Cortex-M3 Debug Monitor Interrupt          */
  PendSV_IRQn                 = -2,      /*!< 14 Cortex-M3 Pend SV Interrupt                */
  SysTick_IRQn                = -1,      /*!< 15 Cortex-M3 System Tick Interrupt            */

/******  LPC17xx Specific Interrupt Numbers
*******************************************************/
  WDT_IRQn                    = 0,       /*!< Watchdog Timer Interrupt                      */
  TIMER0_IRQn                 = 1,       /*!< Timer0 Interrupt                              */
  TIMER1_IRQn                 = 2,       /*!< Timer1 Interrupt                              */
  TIMER2_IRQn                 = 3,       /*!< Timer2 Interrupt                              */
  TIMER3_IRQn                 = 4,       /*!< Timer3 Interrupt                              */
  UART0_IRQn                  = 5,       /*!< UART0 Interrupt                               */
  UART1_IRQn                  = 6,       /*!< UART1 Interrupt                               */
  UART2_IRQn                  = 7,       /*!< UART2 Interrupt                               */
  UART3_IRQn                  = 8,       /*!< UART3 Interrupt                               */
  PWM1_IRQn                   = 9,       /*!< PWM1 Interrupt                                */
  I2C0_IRQn                   = 10,      /*!< I2C0 Interrupt                                */
  I2C1_IRQn                   = 11,      /*!< I2C1 Interrupt                                */
  I2C2_IRQn                   = 12,      /*!< I2C2 Interrupt                                */
  SPI_IRQn                    = 13,      /*!< SPI Interrupt                                 */
  SSP0_IRQn                   = 14,      /*!< SSP0 Interrupt                                */
  SSP1_IRQn                   = 15,      /*!< SSP1 Interrupt                                */
  PLL0_IRQn                   = 16,      /*!< PLL0 Lock (Main PLL) Interrupt                */
  RTC_IRQn                    = 17,      /*!< Real Time Clock Interrupt                     */
  EINT0_IRQn                  = 18,      /*!< External Interrupt 0 Interrupt                */
  EINT1_IRQn                  = 19,      /*!< External Interrupt 1 Interrupt                */
  EINT2_IRQn                  = 20,      /*!< External Interrupt 2 Interrupt                */
  EINT3_IRQn                  = 21,      /*!< External Interrupt 3 Interrupt                */
  ADC_IRQn                    = 22,      /*!< A/D Converter Interrupt                       */
  BOD_IRQn                    = 23,      /*!< Brown-Out Detect Interrupt                    */
  USB_IRQn                    = 24,      /*!< USB Interrupt                                 */
  CAN_IRQn                    = 25,      /*!< CAN Interrupt                                 */
  DMA_IRQn                    = 26,      /*!< General Purpose DMA Interrupt                 */
  I2S_IRQn                    = 27,      /*!< I2S Interrupt                                 */
  ENET_IRQn                   = 28,      /*!< Ethernet Interrupt                            */
  RIT_IRQn                    = 29,      /*!< Repetitive Interrupt Timer Interrupt          */
  MCPWM_IRQn                  = 30,      /*!< Motor Control PWM Interrupt                   */
  QEI_IRQn                    = 31,      /*!< Quadrature Encoder Interface Interrupt        */
  PLL1_IRQn                   = 32,      /*!< PLL1 Lock (USB PLL) Interrupt                 */
  USBActivity_IRQn            = 33,      /* USB Activity interrupt                         */
  CANActivity_IRQn            = 34,      /* CAN Activity interrupt                         */
} IRQn_Type;


/*
 *
==============================================================================
=
 * ----------- Processor and Core Peripheral Section ----------------------
```

```
 *
============================================================================
=
 */

/* Configuration of the Cortex-M3 Processor and Core Peripherals */
#define __MPU_PRESENT             1       /*!< MPU present or not                */
#define __NVIC_PRIO_BITS          5       /*!< Number of Bits used for Priority Levels */
#define __Vendor_SysTickConfig    0       /*!< Set to 1 if different SysTick Config is used */


#include "core_cm3.h"                     /* Cortex-M3 processor and core peripherals */
#include "system_LPC17xx.h"               /* System Header                           */


/******************************************************************************/
/*                Device Specific Peripheral registers structures             */
/******************************************************************************/

#if defined ( __CC_ARM   )
#pragma anon_unions
#endif

/*------------- System Control (SC) ------------------------------------------*/
typedef struct
{
  __IO uint32_t FLASHCFG;               /* Flash Accelerator Module          */
       uint32_t RESERVED0[31];
  __IO uint32_t PLL0CON;                /* Clocking and Power Control        */
  __IO uint32_t PLL0CFG;
  __I  uint32_t PLL0STAT;
  __O  uint32_t PLL0FEED;
       uint32_t RESERVED1[4];
  __IO uint32_t PLL1CON;
  __IO uint32_t PLL1CFG;
  __I  uint32_t PLL1STAT;
  __O  uint32_t PLL1FEED;
       uint32_t RESERVED2[4];
  __IO uint32_t PCON;
  __IO uint32_t PCONP;
       uint32_t RESERVED3[15];
  __IO uint32_t CCLKCFG;
  __IO uint32_t USBCLKCFG;
  __IO uint32_t CLKSRCSEL;
  __IO uint32_t         CANSLEEPCLR;
  __IO uint32_t         CANWAKEFLAGS;
       uint32_t RESERVED4[10];
  __IO uint32_t EXTINT;                 /* External Interrupts               */
       uint32_t RESERVED5;
```

```
  __IO uint32_t EXTMODE;
  __IO uint32_t EXTPOLAR;
       uint32_t RESERVED6[12];
  __IO uint32_t RSID;                /* Reset                          */
       uint32_t RESERVED7[7];
  __IO uint32_t SCS;                 /* Syscon Miscellaneous Registers     */
  __IO uint32_t IRCTRIM;               /* Clock Dividers                 */
  __IO uint32_t PCLKSEL0;
  __IO uint32_t PCLKSEL1;
       uint32_t RESERVED8[4];
  __IO uint32_t USBIntSt;              /* USB Device/OTG Interrupt Register  */
  __IO uint32_t DMAREQSEL;
  __IO uint32_t CLKOUTCFG;             /* Clock Output Configuration         */
 } LPC_SC_TypeDef;

/*------------- Pin Connect Block (PINCON) ----------------------------------*/
typedef struct
{
  __IO uint32_t PINSEL0;
  __IO uint32_t PINSEL1;
  __IO uint32_t PINSEL2;
  __IO uint32_t PINSEL3;
  __IO uint32_t PINSEL4;
  __IO uint32_t PINSEL5;
  __IO uint32_t PINSEL6;
  __IO uint32_t PINSEL7;
  __IO uint32_t PINSEL8;
  __IO uint32_t PINSEL9;
  __IO uint32_t PINSEL10;
       uint32_t RESERVED0[5];
  __IO uint32_t PINMODE0;
  __IO uint32_t PINMODE1;
  __IO uint32_t PINMODE2;
  __IO uint32_t PINMODE3;
  __IO uint32_t PINMODE4;
  __IO uint32_t PINMODE5;
  __IO uint32_t PINMODE6;
  __IO uint32_t PINMODE7;
  __IO uint32_t PINMODE8;
  __IO uint32_t PINMODE9;
  __IO uint32_t PINMODE_OD0;
  __IO uint32_t PINMODE_OD1;
  __IO uint32_t PINMODE_OD2;
  __IO uint32_t PINMODE_OD3;
  __IO uint32_t PINMODE_OD4;
  __IO uint32_t I2CPADCFG;
} LPC_PINCON_TypeDef;

/*------------- General Purpose Input/Output (GPIO) -------------------------*/
```

```
typedef struct
{
  union {
    __IO uint32_t FIODIR;
    struct {
      __IO uint16_t FIODIRL;
      __IO uint16_t FIODIRH;
    };
    struct {
      __IO uint8_t  FIODIR0;
      __IO uint8_t  FIODIR1;
      __IO uint8_t  FIODIR2;
      __IO uint8_t  FIODIR3;
    };
  };
  uint32_t RESERVED0[3];
  union {
    __IO uint32_t FIOMASK;
    struct {
      __IO uint16_t FIOMASKL;
      __IO uint16_t FIOMASKH;
    };
    struct {
      __IO uint8_t  FIOMASK0;
      __IO uint8_t  FIOMASK1;
      __IO uint8_t  FIOMASK2;
      __IO uint8_t  FIOMASK3;
    };
  };
  union {
    __IO uint32_t FIOPIN;
    struct {
      __IO uint16_t FIOPINL;
      __IO uint16_t FIOPINH;
    };
    struct {
      __IO uint8_t  FIOPIN0;
      __IO uint8_t  FIOPIN1;
      __IO uint8_t  FIOPIN2;
      __IO uint8_t  FIOPIN3;
    };
  };
  union {
    __IO uint32_t FIOSET;
    struct {
      __IO uint16_t FIOSETL;
      __IO uint16_t FIOSETH;
    };
    struct {
```

```
      __IO uint8_t  FIOSET0;
      __IO uint8_t  FIOSET1;
      __IO uint8_t  FIOSET2;
      __IO uint8_t  FIOSET3;
    };
  };
  union {
    __O  uint32_t FIOCLR;
    struct {
      __O  uint16_t FIOCLRL;
      __O  uint16_t FIOCLRH;
    };
    struct {
      __O  uint8_t  FIOCLR0;
      __O  uint8_t  FIOCLR1;
      __O  uint8_t  FIOCLR2;
      __O  uint8_t  FIOCLR3;
    };
  };
} LPC_GPIO_TypeDef;

typedef struct
{
  __I  uint32_t IntStatus;
  __I  uint32_t IO0IntStatR;
  __I  uint32_t IO0IntStatF;
  __O  uint32_t IO0IntClr;
  __IO uint32_t IO0IntEnR;
  __IO uint32_t IO0IntEnF;
       uint32_t RESERVED0[3];
  __I  uint32_t IO2IntStatR;
  __I  uint32_t IO2IntStatF;
  __O  uint32_t IO2IntClr;
  __IO uint32_t IO2IntEnR;
  __IO uint32_t IO2IntEnF;
} LPC_GPIOINT_TypeDef;

/*------------- Timer (TIM) ----------------------------------------------------*/
typedef struct
{
  __IO uint32_t IR;
  __IO uint32_t TCR;
  __IO uint32_t TC;
  __IO uint32_t PR;
  __IO uint32_t PC;
  __IO uint32_t MCR;
  __IO uint32_t MR0;
  __IO uint32_t MR1;
  __IO uint32_t MR2;
```

```
  __IO uint32_t MR3;
  __IO uint32_t CCR;
  __I  uint32_t CR0;
  __I  uint32_t CR1;
       uint32_t RESERVED0[2];
  __IO uint32_t EMR;
       uint32_t RESERVED1[12];
  __IO uint32_t CTCR;
} LPC_TIM_TypeDef;

/*------------- Pulse-Width Modulation (PWM) ---------------------------------*/
typedef struct
{
  __IO uint32_t IR;
  __IO uint32_t TCR;
  __IO uint32_t TC;
  __IO uint32_t PR;
  __IO uint32_t PC;
  __IO uint32_t MCR;
  __IO uint32_t MR0;
  __IO uint32_t MR1;
  __IO uint32_t MR2;
  __IO uint32_t MR3;
  __IO uint32_t CCR;
  __I  uint32_t CR0;
  __I  uint32_t CR1;
  __I  uint32_t CR2;
  __I  uint32_t CR3;
       uint32_t RESERVED0;
  __IO uint32_t MR4;
  __IO uint32_t MR5;
  __IO uint32_t MR6;
  __IO uint32_t PCR;
  __IO uint32_t LER;
       uint32_t RESERVED1[7];
  __IO uint32_t CTCR;
} LPC_PWM_TypeDef;

/*------------- Universal Asynchronous Receiver Transmitter (UART) -----------*/
typedef struct
{
  union {
  __I  uint8_t  RBR;
  __O  uint8_t  THR;
  __IO uint8_t  DLL;
       uint32_t RESERVED0;
  };
  union {
  __IO uint8_t  DLM;
```

```
  __IO uint32_t IER;
  };
  union {
  __I  uint32_t IIR;
  __O  uint8_t  FCR;
  };
  __IO uint8_t  LCR;
       uint8_t  RESERVED1[7];
  __I  uint8_t  LSR;
       uint8_t  RESERVED2[7];
  __IO uint8_t  SCR;
       uint8_t  RESERVED3[3];
  __IO uint32_t ACR;
  __IO uint8_t  ICR;
       uint8_t  RESERVED4[3];
  __IO uint8_t  FDR;
       uint8_t  RESERVED5[7];
  __IO uint8_t  TER;
       uint8_t  RESERVED6[39];
  __IO uint32_t FIFOLVL;
} LPC_UART_TypeDef;

typedef struct
{
  union {
  __I  uint8_t  RBR;
  __O  uint8_t  THR;
  __IO uint8_t  DLL;
       uint32_t RESERVED0;
  };
  union {
  __IO uint8_t  DLM;
  __IO uint32_t IER;
  };
  union {
  __I  uint32_t IIR;
  __O  uint8_t  FCR;
  };
  __IO uint8_t  LCR;
       uint8_t  RESERVED1[7];
  __I  uint8_t  LSR;
       uint8_t  RESERVED2[7];
  __IO uint8_t  SCR;
       uint8_t  RESERVED3[3];
  __IO uint32_t ACR;
  __IO uint8_t  ICR;
       uint8_t  RESERVED4[3];
  __IO uint8_t  FDR;
       uint8_t  RESERVED5[7];
```

```
  __IO uint8_t  TER;
       uint8_t  RESERVED6[39];
  __IO uint32_t FIFOLVL;
} LPC_UART0_TypeDef;

typedef struct
{
  union {
  __I  uint8_t  RBR;
  __O  uint8_t  THR;
  __IO uint8_t  DLL;
       uint32_t RESERVED0;
  };
  union {
  __IO uint8_t  DLM;
  __IO uint32_t IER;
  };
  union {
  __I  uint32_t IIR;
  __O  uint8_t  FCR;
  };
  __IO uint8_t  LCR;
       uint8_t  RESERVED1[3];
  __IO uint8_t  MCR;
       uint8_t  RESERVED2[3];
  __I  uint8_t  LSR;
       uint8_t  RESERVED3[3];
  __I  uint8_t  MSR;
       uint8_t  RESERVED4[3];
  __IO uint8_t  SCR;
       uint8_t  RESERVED5[3];
  __IO uint32_t ACR;
       uint32_t RESERVED6;
  __IO uint32_t FDR;
       uint32_t RESERVED7;
  __IO uint8_t  TER;
       uint8_t  RESERVED8[27];
  __IO uint8_t  RS485CTRL;
       uint8_t  RESERVED9[3];
  __IO uint8_t  ADRMATCH;
       uint8_t  RESERVED10[3];
  __IO uint8_t  RS485DLY;
       uint8_t  RESERVED11[3];
  __IO uint32_t FIFOLVL;
} LPC_UART1_TypeDef;

/*---------- Serial Peripheral Interface (SPI) ----------------------------*/
typedef struct
{
```

```
  __IO uint32_t SPCR;
  __I  uint32_t SPSR;
  __IO uint32_t SPDR;
  __IO uint32_t SPCCR;
       uint32_t RESERVED0[3];
  __IO uint32_t SPINT;
} LPC_SPI_TypeDef;

/*------------- Synchronous Serial Communication (SSP) -----------------------*/
typedef struct
{
  __IO uint32_t CR0;
  __IO uint32_t CR1;
  __IO uint32_t DR;
  __I  uint32_t SR;
  __IO uint32_t CPSR;
  __IO uint32_t IMSC;
  __IO uint32_t RIS;
  __IO uint32_t MIS;
  __IO uint32_t ICR;
  __IO uint32_t DMACR;
} LPC_SSP_TypeDef;

/*------------- Inter-Integrated Circuit (I2C) -------------------------------*/
typedef struct
{
  __IO uint32_t I2CONSET;
  __I  uint32_t I2STAT;
  __IO uint32_t I2DAT;
  __IO uint32_t I2ADR0;
  __IO uint32_t I2SCLH;
  __IO uint32_t I2SCLL;
  __O  uint32_t I2CONCLR;
  __IO uint32_t MMCTRL;
  __IO uint32_t I2ADR1;
  __IO uint32_t I2ADR2;
  __IO uint32_t I2ADR3;
  __I  uint32_t I2DATA_BUFFER;
  __IO uint32_t I2MASK0;
  __IO uint32_t I2MASK1;
  __IO uint32_t I2MASK2;
  __IO uint32_t I2MASK3;
} LPC_I2C_TypeDef;

/*------------- Inter IC Sound (I2S) -----------------------------------------*/
typedef struct
{
  __IO uint32_t I2SDAO;
  __IO uint32_t I2SDAI;
```

```
  __O  uint32_t I2STXFIFO;
  __I  uint32_t I2SRXFIFO;
  __I  uint32_t I2SSTATE;
  __IO uint32_t I2SDMA1;
  __IO uint32_t I2SDMA2;
  __IO uint32_t I2SIRQ;
  __IO uint32_t I2STXRATE;
  __IO uint32_t I2SRXRATE;
  __IO uint32_t I2STXBITRATE;
  __IO uint32_t I2SRXBITRATE;
  __IO uint32_t I2STXMODE;
  __IO uint32_t I2SRXMODE;
} LPC_I2S_TypeDef;

/*------------- Repetitive Interrupt Timer (RIT) -----------------------------*/
typedef struct
{
  __IO uint32_t RICOMPVAL;
  __IO uint32_t RIMASK;
  __IO uint8_t  RICTRL;
       uint8_t  RESERVED0[3];
  __IO uint32_t RICOUNTER;
} LPC_RIT_TypeDef;

/*------------- Real-Time Clock (RTC) ----------------------------------------*/
typedef struct
{
  __IO uint8_t  ILR;
       uint8_t  RESERVED0[7];
  __IO uint8_t  CCR;
       uint8_t  RESERVED1[3];
  __IO uint8_t  CIIR;
       uint8_t  RESERVED2[3];
  __IO uint8_t  AMR;
       uint8_t  RESERVED3[3];
  __I  uint32_t CTIME0;
  __I  uint32_t CTIME1;
  __I  uint32_t CTIME2;
  __IO uint8_t  SEC;
       uint8_t  RESERVED4[3];
  __IO uint8_t  MIN;
       uint8_t  RESERVED5[3];
  __IO uint8_t  HOUR;
       uint8_t  RESERVED6[3];
  __IO uint8_t  DOM;
       uint8_t  RESERVED7[3];
  __IO uint8_t  DOW;
       uint8_t  RESERVED8[3];
  __IO uint16_t DOY;
```

```
      uint16_t RESERVED9;
  __IO uint8_t  MONTH;
      uint8_t  RESERVED10[3];
  __IO uint16_t YEAR;
      uint16_t RESERVED11;
  __IO uint32_t CALIBRATION;
  __IO uint32_t GPREG0;
  __IO uint32_t GPREG1;
  __IO uint32_t GPREG2;
  __IO uint32_t GPREG3;
  __IO uint32_t GPREG4;
  __IO uint8_t  RTC_AUXEN;
      uint8_t  RESERVED12[3];
  __IO uint8_t  RTC_AUX;
      uint8_t  RESERVED13[3];
  __IO uint8_t  ALSEC;
      uint8_t  RESERVED14[3];
  __IO uint8_t  ALMIN;
      uint8_t  RESERVED15[3];
  __IO uint8_t  ALHOUR;
      uint8_t  RESERVED16[3];
  __IO uint8_t  ALDOM;
      uint8_t  RESERVED17[3];
  __IO uint8_t  ALDOW;
      uint8_t  RESERVED18[3];
  __IO uint16_t ALDOY;
      uint16_t RESERVED19;
  __IO uint8_t  ALMON;
      uint8_t  RESERVED20[3];
  __IO uint16_t ALYEAR;
      uint16_t RESERVED21;
} LPC_RTC_TypeDef;

/*------------- Watchdog Timer (WDT) -----------------------------------------*/
typedef struct
{
  __IO uint8_t  WDMOD;
      uint8_t  RESERVED0[3];
  __IO uint32_t WDTC;
  __O  uint8_t  WDFEED;
      uint8_t  RESERVED1[3];
  __I  uint32_t WDTV;
  __IO uint32_t WDCLKSEL;
} LPC_WDT_TypeDef;

/*------------- Analog-to-Digital Converter (ADC) ----------------------------*/
typedef struct
{
  __IO uint32_t ADCR;
```

```
  __IO uint32_t ADGDR;
       uint32_t RESERVED0;
  __IO uint32_t ADINTEN;
  __I  uint32_t ADDR0;
  __I  uint32_t ADDR1;
  __I  uint32_t ADDR2;
  __I  uint32_t ADDR3;
  __I  uint32_t ADDR4;
  __I  uint32_t ADDR5;
  __I  uint32_t ADDR6;
  __I  uint32_t ADDR7;
  __I  uint32_t ADSTAT;
  __IO uint32_t ADTRM;
} LPC_ADC_TypeDef;

/*---------- Digital-to-Analog Converter (DAC) ----------*/
typedef struct
{
  __IO uint32_t DACR;
  __IO uint32_t DACCTRL;
  __IO uint16_t DACCNTVAL;
} LPC_DAC_TypeDef;

/*---------- Motor Control Pulse-Width Modulation (MCPWM) ----------*/
typedef struct
{
  __I  uint32_t MCCON;
  __O  uint32_t MCCON_SET;
  __O  uint32_t MCCON_CLR;
  __I  uint32_t MCCAPCON;
  __O  uint32_t MCCAPCON_SET;
  __O  uint32_t MCCAPCON_CLR;
  __IO uint32_t MCTIM0;
  __IO uint32_t MCTIM1;
  __IO uint32_t MCTIM2;
  __IO uint32_t MCPER0;
  __IO uint32_t MCPER1;
  __IO uint32_t MCPER2;
  __IO uint32_t MCPW0;
  __IO uint32_t MCPW1;
  __IO uint32_t MCPW2;
  __IO uint32_t MCDEADTIME;
  __IO uint32_t MCCCP;
  __IO uint32_t MCCR0;
  __IO uint32_t MCCR1;
  __IO uint32_t MCCR2;
  __I  uint32_t MCINTEN;
  __O  uint32_t MCINTEN_SET;
  __O  uint32_t MCINTEN_CLR;
```

```
  __I  uint32_t MCCNTCON;
  __O  uint32_t MCCNTCON_SET;
  __O  uint32_t MCCNTCON_CLR;
  __I  uint32_t MCINTFLAG;
  __O  uint32_t MCINTFLAG_SET;
  __O  uint32_t MCINTFLAG_CLR;
  __O  uint32_t MCCAP_CLR;
} LPC_MCPWM_TypeDef;

/*------------- Quadrature Encoder Interface (QEI) ---------------------------*/
typedef struct
{
  __O  uint32_t QEICON;
  __I  uint32_t QEISTAT;
  __IO uint32_t QEICONF;
  __I  uint32_t QEIPOS;
  __IO uint32_t QEIMAXPOS;
  __IO uint32_t CMPOS0;
  __IO uint32_t CMPOS1;
  __IO uint32_t CMPOS2;
  __I  uint32_t INXCNT;
  __IO uint32_t INXCMP;
  __IO uint32_t QEILOAD;
  __I  uint32_t QEITIME;
  __I  uint32_t QEIVEL;
  __I  uint32_t QEICAP;
  __IO uint32_t VELCOMP;
  __IO uint32_t FILTER;
       uint32_t RESERVED0[998];
  __O  uint32_t QEIIEC;
  __O  uint32_t QEIIES;
  __I  uint32_t QEIINTSTAT;
  __I  uint32_t QEIIE;
  __O  uint32_t QEICLR;
  __O  uint32_t QEISET;
} LPC_QEI_TypeDef;

/*------------- Controller Area Network (CAN) --------------------------------*/
typedef struct
{
  __IO uint32_t mask[512];             /* ID Masks                      */
} LPC_CANAF_RAM_TypeDef;

typedef struct                         /* Acceptance Filter Registers   */
{
  __IO uint32_t AFMR;
  __IO uint32_t SFF_sa;
  __IO uint32_t SFF_GRP_sa;
  __IO uint32_t EFF_sa;
```

```
  __IO uint32_t EFF_GRP_sa;
  __IO uint32_t ENDofTable;
  __I  uint32_t LUTerrAd;
  __I  uint32_t LUTerr;
  __IO uint32_t FCANIE;
  __IO uint32_t FCANIC0;
  __IO uint32_t FCANIC1;
} LPC_CANAF_TypeDef;

typedef struct                  /* Central Registers            */
{
  __I  uint32_t CANTxSR;
  __I  uint32_t CANRxSR;
  __I  uint32_t CANMSR;
} LPC_CANCR_TypeDef;

typedef struct                  /* Controller Registers         */
{
  __IO uint32_t MOD;
  __O  uint32_t CMR;
  __IO uint32_t GSR;
  __I  uint32_t ICR;
  __IO uint32_t IER;
  __IO uint32_t BTR;
  __IO uint32_t EWL;
  __I  uint32_t SR;
  __IO uint32_t RFS;
  __IO uint32_t RID;
  __IO uint32_t RDA;
  __IO uint32_t RDB;
  __IO uint32_t TFI1;
  __IO uint32_t TID1;
  __IO uint32_t TDA1;
  __IO uint32_t TDB1;
  __IO uint32_t TFI2;
  __IO uint32_t TID2;
  __IO uint32_t TDA2;
  __IO uint32_t TDB2;
  __IO uint32_t TFI3;
  __IO uint32_t TID3;
  __IO uint32_t TDA3;
  __IO uint32_t TDB3;
} LPC_CAN_TypeDef;

/*------------- General Purpose Direct Memory Access (GPDMA) ----------------*/
typedef struct                  /* Common Registers             */
{
  __I  uint32_t DMACIntStat;
  __I  uint32_t DMACIntTCStat;
```

```
  __O  uint32_t DMACIntTCClear;
  __I  uint32_t DMACIntErrStat;
  __O  uint32_t DMACIntErrClr;
  __I  uint32_t DMACRawIntTCStat;
  __I  uint32_t DMACRawIntErrStat;
  __I  uint32_t DMACEnbldChns;
  __IO uint32_t DMACSoftBReq;
  __IO uint32_t DMACSoftSReq;
  __IO uint32_t DMACSoftLBReq;
  __IO uint32_t DMACSoftLSReq;
  __IO uint32_t DMACConfig;
  __IO uint32_t DMACSync;
} LPC_GPDMA_TypeDef;

typedef struct                  /* Channel Registers              */
{
  __IO uint32_t DMACCSrcAddr;
  __IO uint32_t DMACCDestAddr;
  __IO uint32_t DMACCLLI;
  __IO uint32_t DMACCControl;
  __IO uint32_t DMACCConfig;
} LPC_GPDMACH_TypeDef;

/*------------- Universal Serial Bus (USB) -----------------------------------*/
typedef struct
{
  __I  uint32_t HcRevision;         /* USB Host Registers             */
  __IO uint32_t HcControl;
  __IO uint32_t HcCommandStatus;
  __IO uint32_t HcInterruptStatus;
  __IO uint32_t HcInterruptEnable;
  __IO uint32_t HcInterruptDisable;
  __IO uint32_t HcHCCA;
  __I  uint32_t HcPeriodCurrentED;
  __IO uint32_t HcControlHeadED;
  __IO uint32_t HcControlCurrentED;
  __IO uint32_t HcBulkHeadED;
  __IO uint32_t HcBulkCurrentED;
  __I  uint32_t HcDoneHead;
  __IO uint32_t HcFmInterval;
  __I  uint32_t HcFmRemaining;
  __I  uint32_t HcFmNumber;
  __IO uint32_t HcPeriodicStart;
  __IO uint32_t HcLSTreshold;
  __IO uint32_t HcRhDescriptorA;
  __IO uint32_t HcRhDescriptorB;
  __IO uint32_t HcRhStatus;
  __IO uint32_t HcRhPortStatus1;
  __IO uint32_t HcRhPortStatus2;
```

```
    uint32_t RESERVED0[40];
__I  uint32_t Module_ID;

__I  uint32_t OTGIntSt;             /* USB On-The-Go Registers          */
__IO uint32_t OTGIntEn;
__O  uint32_t OTGIntSet;
__O  uint32_t OTGIntClr;
__IO uint32_t OTGStCtrl;
__IO uint32_t OTGTmr;
    uint32_t RESERVED1[58];

__I  uint32_t USBDevIntSt;          /* USB Device Interrupt Registers     */
__IO uint32_t USBDevIntEn;
__O  uint32_t USBDevIntClr;
__O  uint32_t USBDevIntSet;

__O  uint32_t USBCmdCode;           /* USB Device SIE Command Registers   */
__I  uint32_t USBCmdData;

__I  uint32_t USBRxData;            /* USB Device Transfer Registers      */
__O  uint32_t USBTxData;
__I  uint32_t USBRxPLen;
__O  uint32_t USBTxPLen;
__IO uint32_t USBCtrl;
__O  uint32_t USBDevIntPri;

__I  uint32_t USBEpIntSt;           /* USB Device Endpoint Interrupt Regs */
__IO uint32_t USBEpIntEn;
__O  uint32_t USBEpIntClr;
__O  uint32_t USBEpIntSet;
__O  uint32_t USBEpIntPri;

__IO uint32_t USBReEp;              /* USB Device Endpoint Realization Reg*/
__O  uint32_t USBEpInd;
__IO uint32_t USBMaxPSize;

__I  uint32_t USBDMARSt;            /* USB Device DMA Registers          */
__O  uint32_t USBDMARClr;
__O  uint32_t USBDMARSet;
    uint32_t RESERVED2[9];
__IO uint32_t USBUDCAH;
__I  uint32_t USBEpDMASt;
__O  uint32_t USBEpDMAEn;
__O  uint32_t USBEpDMADis;
__I  uint32_t USBDMAIntSt;
__IO uint32_t USBDMAIntEn;
    uint32_t RESERVED3[2];
__I  uint32_t USBEoTIntSt;
__O  uint32_t USBEoTIntClr;
```

```
  __O  uint32_t USBEoTIntSet;
  __I  uint32_t USBNDDRIntSt;
  __O  uint32_t USBNDDRIntClr;
  __O  uint32_t USBNDDRIntSet;
  __I  uint32_t USBSysErrIntSt;
  __O  uint32_t USBSysErrIntClr;
  __O  uint32_t USBSysErrIntSet;
       uint32_t RESERVED4[15];

  union {
  __I  uint32_t I2C_RX;              /* USB OTG I2C Registers            */
  __O  uint32_t I2C_TX;
  };
  __I  uint32_t I2C_STS;
  __IO uint32_t I2C_CTL;
  __IO uint32_t I2C_CLKHI;
  __O  uint32_t I2C_CLKLO;
       uint32_t RESERVED5[824];

  union {
  __IO uint32_t USBClkCtrl;          /* USB Clock Control Registers      */
  __IO uint32_t OTGClkCtrl;
  };
  union {
  __I  uint32_t USBClkSt;
  __I  uint32_t OTGClkSt;
  };
} LPC_USB_TypeDef;

/*------------- Ethernet Media Access Controller (EMAC) ----------------------*/
typedef struct
{
  __IO uint32_t MAC1;                /* MAC Registers                    */
  __IO uint32_t MAC2;
  __IO uint32_t IPGT;
  __IO uint32_t IPGR;
  __IO uint32_t CLRT;
  __IO uint32_t MAXF;
  __IO uint32_t SUPP;
  __IO uint32_t TEST;
  __IO uint32_t MCFG;
  __IO uint32_t MCMD;
  __IO uint32_t MADR;
  __O  uint32_t MWTD;
  __I  uint32_t MRDD;
  __I  uint32_t MIND;
       uint32_t RESERVED0[2];
  __IO uint32_t SA0;
  __IO uint32_t SA1;
```

```
  __IO uint32_t SA2;
       uint32_t RESERVED1[45];
  __IO uint32_t Command;              /* Control Registers                */
  __I  uint32_t Status;
  __IO uint32_t RxDescriptor;
  __IO uint32_t RxStatus;
  __IO uint32_t RxDescriptorNumber;
  __I  uint32_t RxProduceIndex;
  __IO uint32_t RxConsumeIndex;
  __IO uint32_t TxDescriptor;
  __IO uint32_t TxStatus;
  __IO uint32_t TxDescriptorNumber;
  __IO uint32_t TxProduceIndex;
  __I  uint32_t TxConsumeIndex;
       uint32_t RESERVED2[10];
  __I  uint32_t TSV0;
  __I  uint32_t TSV1;
  __I  uint32_t RSV;
       uint32_t RESERVED3[3];
  __IO uint32_t FlowControlCounter;
  __I  uint32_t FlowControlStatus;
       uint32_t RESERVED4[34];
  __IO uint32_t RxFilterCtrl;         /* Rx Filter Registers              */
  __IO uint32_t RxFilterWoLStatus;
  __IO uint32_t RxFilterWoLClear;
       uint32_t RESERVED5;
  __IO uint32_t HashFilterL;
  __IO uint32_t HashFilterH;
       uint32_t RESERVED6[882];
  __I  uint32_t IntStatus;            /* Module Control Registers         */
  __IO uint32_t IntEnable;
  __O  uint32_t IntClear;
  __O  uint32_t IntSet;
       uint32_t RESERVED7;
  __IO uint32_t PowerDown;
       uint32_t RESERVED8;
  __IO uint32_t Module_ID;
} LPC_EMAC_TypeDef;

#if defined ( __CC_ARM   )
#pragma no_anon_unions
#endif


/*****************************************************************************/
/*                       Peripheral memory map                               */
/*****************************************************************************/
/* Base addresses                                                            */
#define LPC_FLASH_BASE        (0x00000000UL)
```

```
#define LPC_RAM_BASE        (0x10000000UL)
#define LPC_GPIO_BASE       (0x2009C000UL)
#define LPC_APB0_BASE       (0x40000000UL)
#define LPC_APB1_BASE       (0x40080000UL)
#define LPC_AHB_BASE        (0x50000000UL)
#define LPC_CM3_BASE        (0xE0000000UL)

/* APB0 peripherals                                     */
#define LPC_WDT_BASE        (LPC_APB0_BASE + 0x00000)
#define LPC_TIM0_BASE       (LPC_APB0_BASE + 0x04000)
#define LPC_TIM1_BASE       (LPC_APB0_BASE + 0x08000)
#define LPC_UART0_BASE      (LPC_APB0_BASE + 0x0C000)
#define LPC_UART1_BASE      (LPC_APB0_BASE + 0x10000)
#define LPC_PWM1_BASE       (LPC_APB0_BASE + 0x18000)
#define LPC_I2C0_BASE       (LPC_APB0_BASE + 0x1C000)
#define LPC_SPI_BASE        (LPC_APB0_BASE + 0x20000)
#define LPC_RTC_BASE        (LPC_APB0_BASE + 0x24000)
#define LPC_GPIOINT_BASE    (LPC_APB0_BASE + 0x28080)
#define LPC_PINCON_BASE     (LPC_APB0_BASE + 0x2C000)
#define LPC_SSP1_BASE       (LPC_APB0_BASE + 0x30000)
#define LPC_ADC_BASE        (LPC_APB0_BASE + 0x34000)
#define LPC_CANAF_RAM_BASE  (LPC_APB0_BASE + 0x38000)
#define LPC_CANAF_BASE      (LPC_APB0_BASE + 0x3C000)
#define LPC_CANCR_BASE      (LPC_APB0_BASE + 0x40000)
#define LPC_CAN1_BASE       (LPC_APB0_BASE + 0x44000)
#define LPC_CAN2_BASE       (LPC_APB0_BASE + 0x48000)
#define LPC_I2C1_BASE       (LPC_APB0_BASE + 0x5C000)

/* APB1 peripherals                                     */
#define LPC_SSP0_BASE       (LPC_APB1_BASE + 0x08000)
#define LPC_DAC_BASE        (LPC_APB1_BASE + 0x0C000)
#define LPC_TIM2_BASE       (LPC_APB1_BASE + 0x10000)
#define LPC_TIM3_BASE       (LPC_APB1_BASE + 0x14000)
#define LPC_UART2_BASE      (LPC_APB1_BASE + 0x18000)
#define LPC_UART3_BASE      (LPC_APB1_BASE + 0x1C000)
#define LPC_I2C2_BASE       (LPC_APB1_BASE + 0x20000)
#define LPC_I2S_BASE        (LPC_APB1_BASE + 0x28000)
#define LPC_RIT_BASE        (LPC_APB1_BASE + 0x30000)
#define LPC_MCPWM_BASE      (LPC_APB1_BASE + 0x38000)
#define LPC_QEI_BASE        (LPC_APB1_BASE + 0x3C000)
#define LPC_SC_BASE         (LPC_APB1_BASE + 0x7C000)

/* AHB peripherals                                      */
#define LPC_EMAC_BASE       (LPC_AHB_BASE  + 0x00000)
#define LPC_GPDMA_BASE      (LPC_AHB_BASE  + 0x04000)
#define LPC_GPDMACH0_BASE   (LPC_AHB_BASE  + 0x04100)
#define LPC_GPDMACH1_BASE   (LPC_AHB_BASE  + 0x04120)
#define LPC_GPDMACH2_BASE   (LPC_AHB_BASE  + 0x04140)
#define LPC_GPDMACH3_BASE   (LPC_AHB_BASE  + 0x04160)
```

```
#define LPC_GPDMACH4_BASE     (LPC_AHB_BASE  + 0x04180)
#define LPC_GPDMACH5_BASE     (LPC_AHB_BASE  + 0x041A0)
#define LPC_GPDMACH6_BASE     (LPC_AHB_BASE  + 0x041C0)
#define LPC_GPDMACH7_BASE     (LPC_AHB_BASE  + 0x041E0)
#define LPC_USB_BASE          (LPC_AHB_BASE  + 0x0C000)

/* GPIOs                                                */
#define LPC_GPIO0_BASE        (LPC_GPIO_BASE + 0x00000)
#define LPC_GPIO1_BASE        (LPC_GPIO_BASE + 0x00020)
#define LPC_GPIO2_BASE        (LPC_GPIO_BASE + 0x00040)
#define LPC_GPIO3_BASE        (LPC_GPIO_BASE + 0x00060)
#define LPC_GPIO4_BASE        (LPC_GPIO_BASE + 0x00080)


/*****************************************************************************/
/*                       Peripheral declaration                              */
/*****************************************************************************/
#define LPC_SC              ((LPC_SC_TypeDef       *) LPC_SC_BASE      )
#define LPC_GPIO0             ((LPC_GPIO_TypeDef     *) LPC_GPIO0_BASE    )
#define LPC_GPIO1             ((LPC_GPIO_TypeDef     *) LPC_GPIO1_BASE    )
#define LPC_GPIO2             ((LPC_GPIO_TypeDef     *) LPC_GPIO2_BASE    )
#define LPC_GPIO3             ((LPC_GPIO_TypeDef     *) LPC_GPIO3_BASE    )
#define LPC_GPIO4             ((LPC_GPIO_TypeDef     *) LPC_GPIO4_BASE    )
#define LPC_WDT               ((LPC_WDT_TypeDef      *) LPC_WDT_BASE      )
#define LPC_TIM0              ((LPC_TIM_TypeDef      *) LPC_TIM0_BASE     )
#define LPC_TIM1              ((LPC_TIM_TypeDef      *) LPC_TIM1_BASE     )
#define LPC_TIM2              ((LPC_TIM_TypeDef      *) LPC_TIM2_BASE     )
#define LPC_TIM3              ((LPC_TIM_TypeDef      *) LPC_TIM3_BASE     )
#define LPC_RIT             ((LPC_RIT_TypeDef      *) LPC_RIT_BASE      )
#define LPC_UART0             ((LPC_UART0_TypeDef    *) LPC_UART0_BASE    )
#define LPC_UART1             ((LPC_UART1_TypeDef    *) LPC_UART1_BASE    )
#define LPC_UART2             ((LPC_UART_TypeDef     *) LPC_UART2_BASE    )
#define LPC_UART3             ((LPC_UART_TypeDef     *) LPC_UART3_BASE    )
#define LPC_PWM1              ((LPC_PWM_TypeDef      *) LPC_PWM1_BASE     )
#define LPC_I2C0            ((LPC_I2C_TypeDef      *) LPC_I2C0_BASE     )
#define LPC_I2C1            ((LPC_I2C_TypeDef      *) LPC_I2C1_BASE     )
#define LPC_I2C2            ((LPC_I2C_TypeDef      *) LPC_I2C2_BASE     )
#define LPC_I2S             ((LPC_I2S_TypeDef      *) LPC_I2S_BASE      )
#define LPC_SPI             ((LPC_SPI_TypeDef      *) LPC_SPI_BASE      )
#define LPC_RTC             ((LPC_RTC_TypeDef      *) LPC_RTC_BASE      )
#define LPC_GPIOINT           ((LPC_GPIOINT_TypeDef  *) LPC_GPIOINT_BASE  )
#define LPC_PINCON            ((LPC_PINCON_TypeDef   *) LPC_PINCON_BASE   )
#define LPC_SSP0            ((LPC_SSP_TypeDef      *) LPC_SSP0_BASE     )
#define LPC_SSP1            ((LPC_SSP_TypeDef      *) LPC_SSP1_BASE     )
#define LPC_ADC             ((LPC_ADC_TypeDef      *) LPC_ADC_BASE      )
#define LPC_DAC             ((LPC_DAC_TypeDef      *) LPC_DAC_BASE      )
#define LPC_CANAF_RAM       ((LPC_CANAF_RAM_TypeDef *) LPC_CANAF_RAM_BASE)
#define LPC_CANAF             ((LPC_CANAF_TypeDef    *) LPC_CANAF_BASE    )
#define LPC_CANCR             ((LPC_CANCR_TypeDef    *) LPC_CANCR_BASE    )
```

```
#define LPC_CAN1            ((LPC_CAN_TypeDef       *) LPC_CAN1_BASE     )
#define LPC_CAN2            ((LPC_CAN_TypeDef       *) LPC_CAN2_BASE     )
#define LPC_MCPWM             ((LPC_MCPWM_TypeDef     *) LPC_MCPWM_BASE    )
#define LPC_QEI           ((LPC_QEI_TypeDef       *) LPC_QEI_BASE      )
#define LPC_EMAC            ((LPC_EMAC_TypeDef      *) LPC_EMAC_BASE     )
#define LPC_GPDMA             ((LPC_GPDMA_TypeDef     *) LPC_GPDMA_BASE    )
#define LPC_GPDMACH0          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH0_BASE )
#define LPC_GPDMACH1          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH1_BASE )
#define LPC_GPDMACH2          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH2_BASE )
#define LPC_GPDMACH3          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH3_BASE )
#define LPC_GPDMACH4          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH4_BASE )
#define LPC_GPDMACH5          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH5_BASE )
#define LPC_GPDMACH6          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH6_BASE )
#define LPC_GPDMACH7          ((LPC_GPDMACH_TypeDef   *) LPC_GPDMACH7_BASE )
#define LPC_USB             ((LPC_USB_TypeDef       *) LPC_USB_BASE      )

#endif  // __LPC17xx_H__
```