KL25Z Learning Note

## Freescale ARM Cortex-M Embedded Programming

- ISO C99 data types are defined in stdint.h, which is used to prevent ambiguity and make the code portable between different compliers.

- XOR can be used to toggle a bit like var1=var1^0x10

- Each port has up to 32 pins and they are designated as PTA0-PTA31. It must be noted that not all 32 pins of each port are implemented.

- ARM chips have two buses: APB and AHB. AHB allows one clock cycle access to the peripherals while the APB is slower with the minimum of 2 clock cycles. AHB is referred as FGPIO in Freescale datasheet.

- The SIM\_SCGC5(system clock gating control register) is used to enable the clock source for the I/O port.

- The physical address of registers are already defined in the header file MKL25Z4.H.

- Keil MDK-ARM uses a different syntax to define the registers to be in compliant with CMSIS (Cortex microcontrollers software interface standard). Each port is defined as a pointer to a struct with the registers as the member of the struct. This defined data format is different from that used in Freescale KDS. But both header files share the same name, MKL25Z4.H.

|  |  |
| --- | --- |
| **CMSIS** | **KDS** |
| Define pointer to a GPIO struct  \_\_I and \_\_O are both volatile  \_\_I: read only; \_\_O: write only  typedef struct {    \_\_IO uint32\_t PDOR;     \_\_O  uint32\_t PSOR;      \_\_O  uint32\_t PCOR;     \_\_O  uint32\_t PTOR;     \_\_I  uint32\_t PDIR;    \_\_IO uint32\_t PDDR;  } GPIO\_Type; | Define pointer to a GPIO struct  typedef struct GPIO\_MemMap {    uint32\_t PDOR; /\*\*< Port Data Output Register, offset: 0x0 \*/    uint32\_t PSOR; /\*\*< Port Set Output Register, offset: 0x4 \*/    uint32\_t PCOR; /\*\*< Port Clear Output Register, offset: 0x8 \*/    uint32\_t PTOR; /\*\*< Port Toggle Output Register, offset: 0xC \*/    uint32\_t PDIR;  /\*\*< Port Data Input Register, offset: 0x10 \*/    uint32\_t PDDR; /\*\*< Port Data Direction Register, offset: 0x14 \*/  } volatile \*GPIO\_MemMapPtr; |
| Define GPIOB base address  #define PTB\_BASE     (0x400FF040u) | Define GPIO struct pointer to B base address  #define GPIOB\_BASE\_PTR     ((GPIO\_MemMapPtr)0x400FF040u) |
| Define GPIO struct pointer to B base address  #define PTB     ((GPIO\_Type \*)PTB\_BASE) | Transform expression of access to the element of the struct  #define GPIO\_PDOR\_REG(base)     ((base)->PDOR) |
| Call PDOR  PTB->PDOR | Define GPIOB PDOR, which is equal to GPIOB\_BASE\_PTR->PDOR  #define GPIOB\_PDOR     GPIO\_PDOR\_REG(GPIOB\_BASE\_PTR) |

## Code

The content of the port output register is represented by a symbol.

“volatile unsigned int\*” is used to declare a pointer to a volatile unsigned 8-bit integer.

/\* Port B Data Output Register \*/   
#define GPIOB\_PDOR  (\*((volatile unsigned int\*)0x400FF040))

Delay n ms, the number of 7000 is actually specified by measurement.

/\* Delay n milliseconds   
\* The CPU core clock is set to MCGFLLCLK at 41.94 MHz in SystemInit().   
\*/   
void delayMs(int n) {   
    int i;   
    int j;   
    for(i = 0 ; i < n; i++)   
        for (j = 0; j < 7000; j++) {}   
}

Set bit and clear bit, doesn’t affect any other bit

GPIOB\_PDOR &= ~0x80000; // clear bit D19, KDS

PTB->PDOR &= ~0x80000;  /\* turn on green LED, CMSIS \*/

GPIOB\_PDOR |= 0x80000; // set bit D19, KDS

PTB->PDOR |= 0x80000;   /\* turn off green LED, CMSIS \*/

## PORT

PDOR: data output register, 0x0000

PSOR: set output register, 0x0004

PCOR: clear register, 0x0008

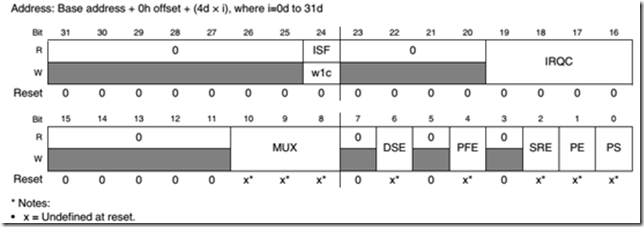
PTOR: toggle output register, 0x000C

PDIR: input register, 0x0010

PDDR: data direction register, 0x0014

pin multiplexing: a single pin is used for multiple functions.

PORTx\_PCR: select alternate I/O function, drive strength, internal resistor (pull-up or pull-down)

[](file:///C:\Users\LZJ\AppData\Local\Temp\WindowsLiveWriter1286139640\supfiles554BBF2\image4.png)

PS: pull select

PE: pull enable

SRE: slew rate

PFE: passive input filter enable

DSE: drive strength

MUX: alternative function selection (001-GPIO)

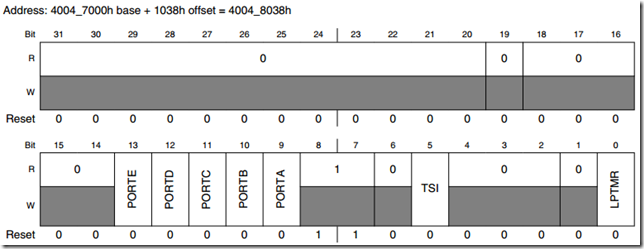
IRQC: interrupt configuration when pin is triggered

ISF: interrupt status flag

#### [KL25 Signal Multiplexing and Pin Assignments](KL25%20Signal%20Multiplexing%20and%20Pin%20Assignments.htm" \t "_blank)

## SIM

SIM\_SCGC5: system clock gating control 5

[](file:///C:\Users\LZJ\AppData\Local\Temp\WindowsLiveWriter1286139640\supfiles554BBF2\image5.png)

LPTMR: lower power timer access control

TSI: TSI access control

PORTx: clock gate control